TRAVEO™ T2G family AUTOSAR MCAL PWM release notes

SRN223361 version 1.15

About this document

Scope and purpose

Thank you for your interest in the TRAVEO™ T2G family AUTOSAR MCAL PWM driver version 1.15. This document lists the installation requirements, software changes, limitations, and known issues.

Intended audience

This document is intended for anyone who uses the pulse width modulation (PWM) driver of the TRAVEO™ T2G family.

Table of contents

Table of contents

1 System requirements and recommendations

Software prerequisites	Supported version
EB tresos Studio package for INFINEON	26.2.0

1.1 Supported compilers

Green Hills Software, compiler v2017.1.4

IAR Embedded Workbench 8.0, EWARM FS 8.22.3

1.2 Compiler options

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

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

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

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

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

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 (Pwm_lib) section	Size (in bytes)
.text	7578
.bss	1
Combined	7579

GHS (Pwm_src) section	Size (in bytes)
.text	2712
.bss	116
.rodata	548
Combined	3376

IAR (Pwm_lib) section	Size (in bytes)
.text	13692
.bss	1
Combined	13693

IAR (Pwm_src) section	Size (in bytes)
.text	2454
.bss	116
.rodata	516
Combined	3086

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)
`Pwm_Init`	96
`Pwm_DeInit`	36
`Pwm_SetDutyCycle`	104
`Pwm_SetPeriodAndDuty`	116
`Pwm_SetOutputToIdle`	28
`Pwm_GetOutputState`	32
`Pwm_DisableNotification`	28
`Pwm_EnableNotification`	36
`Pwm_GetVersionInfo`	12
`Pwm_StartGroupTrigger`	100
`Pwm_GetChannelStatus`	96
`Pwm_StopGroupTrigger`	44
`Pwm_SetChannelOutput`	80

Function	Max stack usage (in bytes)
Pwm_DisableTrigger	84
Pwm_EnableTrigger	84
Pwm_SetOutputOffset	104
Pwm_SetTriggerDelay	100
Pwm_SetPrescaler	92
Pwm_SetDutyCycleBuffer	88
Pwm_SetDutyAndChannelOutputBuffer	96
Pwm_SetChannelOutputBuffer	92
Pwm_Isr_Vector_290_Cat1	32
Pwm_Isr_Vector_290_Cat2	32
Pwm_Isr_Vector_297_Cat1	32
Pwm_Isr_Vector_297_Cat2	32
Pwm_Isr_Vector_342_Cat1	32
Pwm_Isr_Vector_342_Cat2	32
Pwm_Isr_Vector_351_Cat1	32
Pwm_Isr_Vector_351_Cat2	32

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

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

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

1.5.2 IAR Embedded Workbench

Function	Max stack usage (in bytes)
Pwm_Init	56
Pwm_DeInit	40
Pwm_SetDutyCycle	40
Pwm_SetPeriodAndDuty	72
Pwm_SetOutputToIdle	24
Pwm_GetOutputState	40
Pwm_DisableNotification	16
Pwm_EnableNotification	24
Pwm_GetVersionInfo	16

Function	Max stack usage (in bytes)
Pwm_StartGroupTrigger	48
Pwm_GetChannelStatus	76
Pwm_StopGroupTrigger	40
Pwm_SetChannelOutput	40
Pwm_DisableTrigger	32
Pwm_EnableTrigger	32
Pwm_SetOutputOffset	40
Pwm_SetTriggerDelay	40
Pwm_SetPrescaler	64
Pwm_SetDutyCycleBuffer	40
Pwm_SetDutyAndChannelOutputBuffer	48
Pwm_SetChannelOutputBuffer	32
Pwm_Isr_Vector_290_Cat1	24
Pwm_Isr_Vector_290_Cat2	24
Pwm_Isr_Vector_297_Cat1	24
Pwm_Isr_Vector_297_Cat2	24
Pwm_Isr_Vector_342_Cat1	24
Pwm_Isr_Vector_342_Cat2	24
Pwm_Isr_Vector_351_Cat1	24
Pwm_Isr_Vector_351_Cat2	24

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

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

1.6 Note on "*_Bswmd.arxml"

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

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

Named files are not tested.

1.7 Release details

Module software version
1.15.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 Pwm_MemMap.h stub file version
1.0.1

2 Installation

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

3 Deviations from AUTOSAR

T2MC-14405 - [SWS_Pwm_00079] VARIANT-PRE-COMPILE (pre compile) is limited to pre-compile configuration parameters only

Title: [SWS_Pwm_00079] VARIANT-PRE-COMPILE (pre compile) is limited to pre-compile configuration parameters only

Description: [SWS_Pwm_00079] [VARIANT-PRE-COMPILE (Pre Compile) is limited to pre-compile configuration parameters only.] ()

Reason for rejection: Only post-build time is supported.

T2MC-14443 - [SWS_Pwm_00153] These requirements are not applicable to this specification

Title: [SWS_Pwm_00153] These requirements are not applicable to this specification

Description: [SWS_Pwm_00153] [These requirements are not applicable to this specification.] (SRS_BSW_00159, SRS_BSW_00167, SRS_BSW_00170, SRS_BSW_00419, SRS_BSW_00383, 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_00415, SRS_BSW_00164, SRS_BSW_00325, SRS_BSW_00326, SRS_BSW_00342, SRS_BSW_00160, SRS_BSW_00007, SRS_BSW_00300, SRS_BSW_00413, SRS_BSW_00347, SRS_BSW_00305, SRS_BSW_00307, SRS_BSW_00310, SRS_BSW_00373, SRS_BSW_00327, SRS_BSW_00335, SRS_BSW_00350, SRS_BSW_00408, SRS_BSW_00410, SRS_BSW_00348, SRS_BSW_00353, SRS_BSW_00361, SRS_BSW_00301, SRS_BSW_00302, SRS_BSW_00328, SRS_BSW_00312, SRS_BSW_00006, SRS_BSW_00357, SRS_BSW_00377, SRS_BSW_00304, SRS_BSW_00355, SRS_BSW_00378, SRS_BSW_00306, SRS_BSW_00308, SRS_BSW_00309, SRS_BSW_00371, SRS_BSW_00358, SRS_BSW_00414, SRS_BSW_00376, SRS_BSW_00359, SRS_BSW_00360, SRS_BSW_00329, SRS_BSW_00330, SRS_BSW_00331, SRS_BSW_00009, SRS_BSW_00401, SRS_BSW_00172, SRS_BSW_00010, SRS_BSW_00333, SRS_BSW_00003, SRS_BSW_00341, SRS_BSW_00334, SRS_SPAL_12267, SRS_SPAL_12461, SRS_SPAL_12462, SRS_SPAL_12463, SRS_SPAL_12068, SRS_SPAL_12069, SRS_SPAL_12169, SRS_SPAL_12075, SRS_SPAL_12064, SRS_SPAL_12067, SRS_SPAL_12077, SRS_SPAL_12078, SRS_SPAL_12092, SRS_SPAL_12265, SRS_Pwm_12379)

Reason for rejection: Named RQMs are not applicable.

T2MC-14218 - [SWS_Pwm_00154] The PwmDriver shall support power state changes and its APIs

Title: [SWS_Pwm_00154] The PwmDriver shall support power state changes and its APIs

Description: [SWS_Pwm_00154] [The PwmDriver shall support power state changes and its APIs when the corresponding configuration parameter `PwmLowPowerStatesSupport` is set to TRUE.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE.

T2MC-14219 - [SWS_Pwm_00155] The `Pwm_PreparePowerState`, `Pwm_SetPowerState`, `Pwm_GetCurrentPowerState` and `Pwm_GetTargetPowerState` APIs shall be generated and used to manage and get information on power state transitions

Title: [SWS_Pwm_00155] The `Pwm_PreparePowerState`, `Pwm_SetPowerState`, `Pwm_GetCurrentPowerState` and `Pwm_GetTargetPowerState` APIs shall be generated and used to manage and get information on power state transitions

Description: [SWS_Pwm_00155] [If the parameter `PwmLowPowerStatesSupport` is enabled then the APIs `Pwm_PreparePowerState`, `Pwm_SetPowerState`, `Pwm_GetCurrentPowerState`, `Pwm_GetTargetPowerState` shall be generated and shall be used to manage and get information on power state transitions.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE.

T2MC-14220 - [SWS_Pwm_00156] The `Pwm_GetTargetPowerState` and `Pwm_GetCurrentPowerState` APIs shall be used to gather information on the requested and target Pwm power states

Title: [SWS_Pwm_00156] The `Pwm_GetTargetPowerState` and `Pwm_GetCurrentPowerState` APIs shall be used to gather information on the requested and target Pwm power states

Description: [SWS_Pwm_00156] [The APIs `Pwm_GetTargetPowerState` and `Pwm_GetCurrentPowerState` shall be respectively used to gather information on the requested and the target Pwm power states.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; therefore, the `Pwm_GetTargetPowerState` and `Pwm_GetCurrentPowerState` APIs will not be used.

T2MC-14221 - [SWS_Pwm_00157] The `Pwm_PreparePowerState` API shall be used to start a power state transition

Title: [SWS_Pwm_00157] The `Pwm_PreparePowerState` API shall be used to start a power state transition

Description: [SWS_Pwm_00157] [The API `Pwm_PreparePowerState` shall be used to start a power state transition.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; therefore, the `Pwm_PreparePowerState` API will not be used.

T2MC-14222 - [SWS_Pwm_00158] The `Pwm_SetPowerState` API shall be used to achieve the requested power state of the Pwm module

Title: [SWS_Pwm_00158] The `Pwm_SetPowerState` API shall be used to achieve the requested power state of the Pwm module

Description: [SWS_Pwm_00158] [After preparation for a power state is achieved by (**[SWS_Pwm_00157]**) then the API `Pwm_SetPowerState` shall be used to achieve the requested power state of the Pwm module.
In order to avoid incoherent power state conditions, some APIs (`Pwm_SetPowerState`,

Pwm_PreparePowerState) have to be called in a given sequence, otherwise an error (if DET tracing is enabled) is stored and the action is interrupted. The Pwm Driver keeps track of the call sequence.] ()

Reason for rejection: Because power state control is not supported by the hardware, PwmLowPowerStatesSupport is always set to FALSE; therefore, the Pwm_SetPowerState and Pwm_PreparePowerState APIs will not be used.

T2MC-14223 - [SWS_Pwm_00159] The Pwm driver shall keep track of the call order of the Pwm_SetPowerState and Pwm_PreparePowerState APIs

Title: [SWS_Pwm_00159] The Pwm driver shall keep track of the call order of the Pwm_SetPowerState and Pwm_PreparePowerState APIs

Description: [SWS_Pwm_00159] [The Pwm Driver shall keep track of the call order of the APIs Pwm_SetPowerState and Pwm_PreparePowerState. In case the first one is called before the second one is called, a DET entry shall be stored and the action shall not be executed.] ()

Reason for rejection: Because power state control is not supported by the hardware, PwmLowPowerStatesSupport is always set to FALSE; therefore, the Pwm_SetPowerState and Pwm_PreparePowerState APIs will not be used.

T2MC-14224 - [SWS_Pwm_00160] The Pwm module shall keep track of the current and target power states

Title: [SWS_Pwm_00160] The Pwm module shall keep track of the current and target power states

Description: [SWS_Pwm_00160] [The Pwm Module shall keep track of the current and of the target power state if the parameter PwmLowPowerStatesSupport is set to TRUE] ().

Reason for rejection: Because power state control is not supported by the hardware, PwmLowPowerStatesSupport is always set to FALSE.

T2MC-14225 - [SWS_Pwm_00161] After initialization, the power state of the module shall be always FULL POWER

Title: [SWS_Pwm_00161] After initialization, the power state of the module shall be always FULL POWER

Description: [SWS_Pwm_00161] [After the Initialization the power state of the module shall be always FULL POWER if the PwmLowPowerStatesSupport is set to TRUE.] ()

Reason for rejection: Because power state control is not supported by the hardware, PwmLowPowerStatesSupport is always set to FALSE.

T2MC-14226 - [SWS_Pwm_00162] Pwm driver shall support synchronous and asynchronous power state transitions

Title: [SWS_Pwm_00162] Pwm driver shall support synchronous and asynchronous power state transitions

Description: [SWS_Pwm_00162] [The Pwm Driver shall support synchronous and asynchronous power state transitions, depending on the value of the configuration parameter PwmPowerStateAsynchTransitionMode.] ()

Reason for rejection: Because power state control is not supported by the hardware, PwmLowPowerStatesSupport is always set to FALSE; PwmPowerStateAsynchTransitionMode will not be used.

T2MC-14227 - [SWS_Pwm_00163] The preparation and setting processes shall be considered concluded as soon as the respective APIs return

Title: [SWS_Pwm_00163] The preparation and setting processes shall be considered concluded as soon as the respective APIs return

Description: [SWS_Pwm_00163] [In case the configuration parameter `PwmPowerStateAsynchTransitionMode` is set to FALSE, the preparation process and the setting process shall be considered concluded as soon as the respective APIs return.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; `PwmPowerStateAsynchTransitionMode` will not be used.

T2MC-14228 - [SWS_Pwm_00164] The preparation process shall continue in the background after the related API returns

Title: [SWS_Pwm_00164] The preparation process shall continue in the background after the related API returns

Description: [SWS_Pwm_00164] [In case the configuration parameter `PwmPowerStateAsynchTransitionMode` 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: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; `PwmPowerStateAsynchTransitionMode` will not be used.

T2MC-14336 - [SWS_Pwm_00166] Function definition: `Pwm_SetPowerState`

Title: [SWS_Pwm_00166] Function definition: `Pwm_SetPowerState`

Description: [SWS_Pwm_00166]
[

Service name:	`Pwm_SetPowerState`
Syntax:	`Std_ReturnType Pwm_SetPowerState( Pwm_PowerStateRequestResultType* Result )`
Service ID[hex]:	0x09
Sync/async:	Synchronous
Reentrancy:	Non Reentrant
Parameters (in):	None
Parameters (inout):	None

Parameters (out):	Result	If the API returns `E_OK`: `PWM_SERVICE_ACCEPTED`: Power state change executed. If the API return `E_NOT_OK`: `PWM_NOT_INIT`: PWM Module not initialized. `PWM_SEQUENCE_ERROR`: wrong API call sequence. `PWM_HW_FAILURE`: the HW module has a failure which prevents it to enter the required power state.
Return value:	Std_ReturnType	`E_OK`: Power Mode changed `E_NOT_OK`: request rejected
Description:	This API configures the Pwm module so that it enters the already prepared power state, chosen between a predefined set of configured ones.	

] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14337 - [SWS_Pwm_00167] The `Pwm_SetPowerState` API configures the HW to enter the given power state

Title: [SWS_Pwm_00167] The `Pwm_SetPowerState` API configures the HW to enter the given power state

Description: [SWS_Pwm_00167]
[The API configures the HW in order to enter the given Power State. All preliminary actions to enable this transition (e.g. setting all channels in IDLE status, de-registering of all notifications and so on) must already have been taken by the responsible SWCs (e.g. IoHwAbs).
The API shall not execute preliminary, implicit power state changes (i.e. if a requested power state is not reachable starting from the current one, no intermediate power state change shall be executed and the request shall be rejected)] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14338 - [SWS_Pwm_00168] If the target power state is the same as the current one, no action is executed and the API returns immediately with an `E_OK` result

Title: [SWS_Pwm_00168] If the target power state is the same as the current one, no action is executed and the API returns immediately with an `E_OK` result

Description: [SWS_Pwm_00168]
[In case the target power state is the same as the current one, no action is executed and the API returns immediately with an `E_OK` result.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14339 - [SWS_Pwm_00169] If the normal power state is requested, the API shall refer to the necessary parameters

Title: [SWS_Pwm_00169] If the normal power state is requested, the API shall refer to the necessary parameters

Description: [SWS_Pwm_00169]
[In case the normal Power State is requested, the API shall refer to the necessary parameters contained in the same containers used by `Pwm_Init`.
No separate container or hard coded data shall be used for the normal (i.e. full) power mode, in order to avoid misalignments between initialization parameters used during the init phase and during a power state change.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14340 - [SWS_Pwm_00170] Only power state transition specific reconfigurations shall be executed in the context of the `Pwm_SetPowerState` API

Title: [SWS_Pwm_00170] Only power state transition specific reconfigurations shall be executed in the context of the `Pwm_SetPowerState` API

Description: [SWS_Pwm_00170]
[For the other power states, only power state transition specific reconfigurations shall be executed in the context of this API (i.e. the API cannot be used to apply a completely new configuration to the Pwm module). Any other re-configuration not strictly related to the power state transition shall not take place.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14341 - [SWS_Pwm_00171] The `Pwm_SetPowerState` API shall refer to the configuration container related to the required power state

Title: [SWS_Pwm_00171] The `Pwm_SetPowerState` API shall refer to the configuration container related to the required power state

Description: [SWS_Pwm_00171]
[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).] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14342 - [SWS_Pwm_00172] The `Pwm_SetPowerState` API shall report the `PWM_E_UNINIT` DET error

Title: [SWS_Pwm_00172] The `Pwm_SetPowerState` API shall report the `PWM_E_UNINIT` DET error

Description: [SWS_Pwm_00172]
[The API shall report the DET error `PWM_E_UNINIT` in case this API is called before having initialized the HW unit.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14343 - [SWS_Pwm_00173] The `Pwm_SetPowerState` API shall report the `PWM_E_NOT_DISENGAGED` DET error

Title: [SWS_Pwm_00173] The `Pwm_SetPowerState` API shall report the `PWM_E_NOT_DISENGAGED` DET error

Description: [SWS_Pwm_00173]
[The API shall report the DET error `PWM_E_NOT_DISENGAGED` in case this API is called when one or more HW channels (where applicable) are in a state different than IDLE (or similar non-operational states) and/or there are still notification registered for the HW module channels.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14235 - [SWS_Pwm_00174] Pwm driver shall report the `PWM_E_POWER_STATE_NOT_SUPPORTED` DET error

Title: [SWS_Pwm_00174] Pwm driver shall report the `PWM_E_POWER_STATE_NOT_SUPPORTED` DET error

Description: [SWS_Pwm_00174] [The API shall report the DET error `PWM_E_POWER_STATE_NOT_SUPPORTED` in case this API is called with an unsupported power state or the peripheral does not support low power states at all.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this error code will not be used.

T2MC-14236 - [SWS_Pwm_00175] Pwm driver shall report the `PWM_E_TRANSITION_NOT POSSIBLE` DET error

Title: [SWS_Pwm_00175] Pwm driver shall report the `PWM_E_TRANSITION_NOT POSSIBLE` DET error

Description: [SWS_Pwm_00175] [The API shall report the DET error `PWM_E_TRANSITION_NOT POSSIBLE` in case the requested power state cannot be directly reached from the current power state.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this error code will not be used.

T2MC-14237 - [SWS_Pwm_00176] Pwm driver shall report the `PWM_E_PERIPHERAL_NOT_PREPARED` DET error

Title: [SWS_Pwm_00176] Pwm driver shall report the `PWM_E_PERIPHERAL_NOT_PREPARED` DET error

Description: [SWS_Pwm_00176] [The API shall report the DET error `PWM_E_PERIPHERAL_NOT_PREPARED` in case the HW unit has not been previously prepared for the target power state by use of the API `Pwm_PreparePowerState`().] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this error code will not be used.

T2MC-14348 - [SWS_Pwm_00177] Function definition: `Pwm_GetCurrentPowerState`

Title: [SWS_Pwm_00177] Function definition: `Pwm_GetCurrentPowerState`

Description: [SWS_Pwm_00177]
[

Service name:	`Pwm_GetCurrentPowerState`	

Syntax:	`Std_ReturnType Pwm_GetCurrentPowerState(`
	`  Pwm_PowerStateType* CurrentPowerState,`
	`  Pwm_PowerStateRequestResultType* Result`
	`)`	

Service ID[hex]:	0x0a	
Sync/async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	CurrentPowerState	The current power mode of the PWM HW Unit is returned in this parameter
	Result	If the API returns `E_OK`: `PWM_SERVICE_ACCEPTED`: Current power mode was returned. If the API returns `E_NOT_OK`: `PWM_NOT_INIT`: PWM Module not initialized.
Return value:	Std_ReturnType	`E_OK`: Mode could be read `E_NOT_OK`: Service is rejected
Description:	This API returns the current power state of the PWM HW unit.	

] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14349 - [SWS_Pwm_00178] The `Pwm_GetCurrentPowerState` API returns the power state of the HW unit

Title: [SWS_Pwm_00178] The `Pwm_GetCurrentPowerState` API returns the power state of the HW unit

Description: [SWS_Pwm_00178]
[The API returns the power state of the HW unit. In case default error reporting is activated:] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14350 - [SWS_Pwm_00179] The `Pwm_GetCurrentPowerState` API shall report the `PWM_E_UNINIT` DET error

Title: [SWS_Pwm_00179] The `Pwm_GetCurrentPowerState` API shall report the `PWM_E_UNINIT` DET error

Description: [SWS_Pwm_00179]
[The API shall report the DET error `PWM_E_UNINIT` in case this API is called before having initialized the HW unit.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14352 - [SWS_Pwm_00180] Function definition: `Pwm_GetTargetPowerState`

Title: [SWS_Pwm_00180] Function definition: `Pwm_GetTargetPowerState`

Description: [SWS_Pwm_00180]
[

Service name:	`Pwm_GetTargetPowerState`	
Syntax:	`Std_ReturnType Pwm_GetTargetPowerState( Pwm_PowerStateType* TargetPowerState, Pwm_PowerStateRequestResultType* Result )`	
Service ID[hex]:	0x0b	
Sync/async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	None	
Parameters (inout):	None	
Parameters (out):	TargetPowerState	The Target power mode of the PWM HW Unit is returned in this parameter
	Result	If the API returns `E_OK`: `PWM_SERVICE_ACCEPTED`: Target power mode was returned. If the API returns `E_NOT_OK`: `PWM_NOT_INIT`: PWM Module not initialized.
Return value:	Std_ReturnType	`E_OK`: Mode could be read `E_NOT_OK`: Service is rejected
Description:	This API returns the Target power state of the PWM HW unit.	

] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14353 - [SWS_Pwm_00181] The `Pwm_GetTargetPowerState` API returns the requested power state of the HW unit

Title: [SWS_Pwm_00181] The `Pwm_GetTargetPowerState` API returns the requested power state of the HW unit

Description: [SWS_Pwm_00181]
[The API returns the requested power state of the HW unit. This shall coincide with the current power state if no transition is ongoing.
The API is considered to always succeed except in case of HW failures.
In case default error reporting is activated:] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14354 - [SWS_Pwm_00182] The `Pwm_GetTargetPowerState` API shall report the `PWM_E_UNINIT` DET error

Title: [SWS_Pwm_00182] The `Pwm_GetTargetPowerState` API shall report the `PWM_E_UNINIT` DET error

Description: [SWS_Pwm_00182]
[The API shall report the DET error `PWM_E_UNINIT` in case this API is called before having initialized the HW unit.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14356 - [SWS_Pwm_00183] Function definition: `Pwm_PreparePowerState`

Title: [SWS_Pwm_00183] Function definition: `Pwm_PreparePowerState`

Description: [SWS_Pwm_00183]
[

Service name:	`Pwm_PreparePowerState`	

Syntax:	`Std_ReturnType Pwm_PreparePowerState(`
	`  Pwm_PowerStateType PowerState,`
	`  Pwm_PowerStateRequestResultType* Result`
	`)`	

Service ID[hex]:	0x0c	
Sync/async:	Synchronous	
Reentrancy:	Non Reentrant	
Parameters (in):	PowerState	The target power state intended to be attained
Parameters (inout):	None	

Parameters (out):	Result	If the API returns `E_OK`: `PWM_SERVICE_ACCEPTED`: PWM Module power state preparation was started. If the API returns `E_NOT_OK`: `PWM_NOT_INIT`: PWM Module not initialized. `PWM_SEQUENCE_ERROR`: wrong API call sequence (Current Power State = Target Power State). `PWM_POWER_STATE_NOT_SUPP`: PWM Module does not support the requested power state. `PWM_TRANS_NOT_POSSIBLE`: PWM Module cannot transition directly from the current power state to the requested power state or the HW peripheral is still busy.
Return value:	Std_ReturnType	`E_OK`: Preparation process started `E_NOT_OK`: Service is rejected
Description:	This API starts the needed process to allow the PWM HW module to enter the requested power state.	

] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14357 - [SWS_Pwm_00184] This `Pwm_PreparePowerState` API initiates all actions needed to enable a HW module to enter the target power state

Title: [SWS_Pwm_00184] This `Pwm_PreparePowerState` API initiates all actions needed to enable a HW module to enter the target power state

Description: [SWS_Pwm_00184]
[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.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14358 - [SWS_Pwm_00185] If the target power state is the same as the current one, no action is executed and the API returns immediately with an `E_OK` result

Title: [SWS_Pwm_00185] If the target power state is the same as the current one, no action is executed and the API returns immediately with an `E_OK` result

Description: [SWS_Pwm_00185]
[In case the target power state is the same as the current one, no action is executed and the API returns immediately with an `E_OK` result.
The responsibility of the preconditions is left to the environment.
In case default error reporting is activated.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14359 - [SWS_Pwm_00186] The `Pwm_PreparePowerState` API shall report the `PWM_E_UNINIT` DET error

Title: [SWS_Pwm_00186] The `Pwm_PreparePowerState` API shall report the `PWM_E_UNINIT` DET error

Description: [SWS_Pwm_00186]
[The API shall report the DET error `PWM_E_UNINIT` in case this API is called before having initialized the HW unit.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14360 - [SWS_Pwm_00187] The `Pwm_PreparePowerState` API shall report the `PWM_E_POWER_STATE_NOT_SUPPORTED` DET error

Title: [SWS_Pwm_00187] The `Pwm_PreparePowerState` API shall report the `PWM_E_POWER_STATE_NOT_SUPPORTED` DET error

Description: [SWS_Pwm_00187]
[The API shall report the DET error `PWM_E_POWER_STATE_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.]()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14361 - [SWS_Pwm_00188] The `Pwm_PreparePowerState` API shall report the `PWM_E_TRANSITION_NOT POSSIBLE` DET error

Title: [SWS_Pwm_00188] The `Pwm_PreparePowerState` API shall report the `PWM_E_TRANSITION_NOT POSSIBLE` DET error

Description: [SWS_Pwm_00188]
[The API shall report the DET error `PWM_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 `Pwm_Main_PowerTransitionManager`.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14368 - [SWS_Pwm_00189] Function definition: `Pwm_Main_PowerTransitionManager`

Title: [SWS_Pwm_00189] Function definition: `Pwm_Main_PowerTransitionManager`

Description: [SWS_Pwm_00189]
[

Service name:	`Pwm_Main_PowerTransitionManager`
Syntax:	`void Pwm_Main_PowerTransitionManager( void )`
Service ID[hex]:	0x0d

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

] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14369 - [SWS_Pwm_00190] The `Pwm_Main_PowerTransitionManager` API executes any non-immediate action needed to finalize a power state transition

Title: [SWS_Pwm_00190] The `Pwm_Main_PowerTransitionManager` API executes any non-immediate action needed to finalize a power state transition

Description: [SWS_Pwm_00190]
[This API executes any non-immediate action needed to finalize a power state transition requested by `Pwm_PreparePowerState`().] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14370 - [SWS_Pwm_00191] The rate of scheduling shall be defined for the `Pwm_Main_PowerTransitionManager` API

Title: [SWS_Pwm_00191] The rate of scheduling shall be defined for the `Pwm_Main_PowerTransitionManager` API

Description: [SWS_Pwm_00191]
[The rate of scheduling shall be defined by Pwm MainSchedulePeriod and shall be variable, as the function only needs to be called if a transition has been requested.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14371 - [SWS_Pwm_00192] The `Pwm_Main_PowerTransitionManager` API shall also issue callback notifications to the eventually registered users

Title: [SWS_Pwm_00192] The `Pwm_Main_PowerTransitionManager` API shall also issue callback notifications to the eventually registered users

Description: [SWS_Pwm_00192]
[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 power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14372 - [SWS_Pwm_00193] If the PWM module is not initialized, this function shall simply return without any further elaboration

Title: [SWS_Pwm_00193] If the PWM module is not initialized, this function shall simply return without any further elaboration

Description: [SWS_Pwm_00193]
[In case the PWM 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 initialized 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 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 power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14344 - [SWS_Pwm_00194] The `Pwm_SetPowerState` API shall report the `PWM_E_POWER_STATE_NOT_SUPPORTED` DET error

Title: [SWS_Pwm_00194] The `Pwm_SetPowerState` API shall report the `PWM_E_POWER_STATE_NOT_SUPPORTED` DET error

Description: [SWS_Pwm_00194]
[The API shall report the DET error `PWM_E_POWER_STATE_NOT_SUPPORTED` in case this API is called with an unsupported power state or the peripheral does not support low power states at all.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14345 - [SWS_Pwm_00195] The `Pwm_SetPowerState` API shall report the `PWM_E_TRANSITION_NOT POSSIBLE` DET error

Title: [SWS_Pwm_00195] The `Pwm_SetPowerState` API shall report the `PWM_E_TRANSITION_NOT POSSIBLE` DET error

Description: [SWS_Pwm_00195]
[The API shall report the DET error `PWM_E_TRANSITION_NOT POSSIBLE` in case the requested power state cannot be directly reached from the current power state.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14346 - [SWS_Pwm_00196] The `Pwm_SetPowerState` API shall report the `PWM_E_PERIPHERAL_NOT_PREPARED` DET error

Title: [SWS_Pwm_00196] The `Pwm_SetPowerState` API shall report the `PWM_E_PERIPHERAL_NOT_PREPARED` DET error

Description: [SWS_Pwm_00196]
[The API shall report the DET error `PWM_E_PERIPHERAL_NOT_PREPARED` in case the HW unit has not been previously prepared for the target power state by use of the API `Pwm_PreparePowerState`().] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14384 - [SWS_Pwm_00198] Function definition: `IoHwAb_Pwm_NotifyReadyForPowerState`<#Mode>

Title: [SWS_Pwm_00198] Function definition: `IoHwAb_Pwm_NotifyReadyForPowerState`<#Mode>

Description: [SWS_Pwm_00198]
[

Service name:	`IoHwAb_Pwm_NotifyReadyForPowerState`**<#Mode>**
Syntax:	`void IoHwAb_Pwm_NotifyReadyForPowerState<#Mode>( void )`
Service ID[hex]:	0x60
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 PWM Driver when the requested power state preparation for mode <#Mode> is completed.

] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14385 - [SWS_Pwm_00199] The `IoHwAb_Pwm_NotifyReadyForPowerState`<#Mode>API shall be called to indicate the completion of the power transition preparation phase to the IoHwAbs module

Title: [SWS_Pwm_00199] The `IoHwAb_Pwm_NotifyReadyForPowerState`<#Mode>API shall be called to indicate the completion of the power transition preparation phase to the IoHwAbs module

Description: [SWS_Pwm_00199]
[In case the PWM Driver is configured to support power state management with asynchronous transitions, this API shall be called to signal completion of the power transition preparation phase to the IoHwAbs

module.
This is a callback, this API is to be implemented in the IoHwAbs component.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this function will not be used.

T2MC-14234 - [SWS_Pwm_00200] Pwm driver shall report the `PWM_E_NOT_DISENGAGED` DET error

Title: [SWS_Pwm_00200] Pwm driver shall report the `PWM_E_NOT_DISENGAGED` DET error

Description: [SWS_Pwm_00200] [The API shall report the DET error `PWM_E_NOT_DISENGAGED` in case this API is called when one or more HW channels (where applicable) are in a state different than IDLE (or similar non-operational states) and/or there are still notification registered for the HW module channels.] ()

Reason for rejection: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this error code will not be used.

T2MC-14283 - [SWS_Pwm_10120] For pre-compile and link time config, NULL pointer shall be passed to `Pwm_Init`

Title: [SWS_Pwm_10120] For pre-compile and link time config, NULL pointer shall be passed to `Pwm_Init`

Description: [SWS_Pwm_10120] [For pre-compile and link time configuration variants, a NULL pointer shall be passed to the initialization routine.] ()

Reason for rejection: Because only post-build is supported, a NULL pointer will be reported as error.

T2MC-14284 - [SWS_Pwm_20120] For pre-compile and link time configuration variants, the check for this NULL pointer has to be omitted

Title: [SWS_Pwm_20120] For pre-compile and link time configuration variants, the check for this NULL pointer has to be omitted

Description: [SWS_Pwm_20120] [In this case the check for this NULL pointer has to be omitted.] ()

Reason for rejection: Because only post-build is supported, a NULL pointer will be reported as error.

T2MC-14203 - [SWS_Pwm_40075] *Pwm_Lcfg.c* shall include *Pwm.h* and *Pwm_Memmap.h*

Title: [SWS_Pwm_40075] *Pwm_Lcfg.c* shall include *Pwm.h* and *Pwm_Memmap.h*

Description: [SWS_Pwm_40075] [*Pwm_Lcfg.c* shall include *Pwm.h* and *Pwm_Memmap.h*.] ()

Reason for rejection: Only post-build time is supported. Therefore, *Pwm_Lcfg.c* file is not required.

---

4 Limitations

T2MC-14410 - [ECUC_Pwm_00131] Config parameter: PwmDevErrorDetect

Title: [ECUC_Pwm_00131] Config parameter: `PwmDevErrorDetect`

Description:

SWS Item	**ECUC_Pwm_00131 :**		
Name	`PwmDevErrorDetect`		
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 the AUTOSAR specification, detection of development errors is still enabled and errors will be reported via PwmErrorCalloutFunction.

T2MC-14412 - [ECUC_Pwm_00139] Config parameter: `PwmIndex`

Title: [ECUC_Pwm_00139] Config parameter: `PwmIndex`

Description:

SWS Item	**ECUC_Pwm_00139 :**		
Name	`PwmIndex`		
Description	Specifies the InstanceId of this module instance. If only one instance is present it shall have the Id 0.		
Multiplicity	1		
Type	`EcucIntegerParamDef`		
Range	0..4294967295		
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: This parameter is always set to 0.

T2MC-14413 - [ECUC_Pwm_00142] Config parameter: `PwmLowPowerStatesSupport`

Title: [ECUC_Pwm_00142] Config parameter: `PwmLowPowerStatesSupport`

Description:

SWS Item	**ECUC_Pwm_00142 :**		
Name	`PwmLowPowerStatesSupport`		
Description	Adds / removes all power state management related APIs (`PWM_SetPowerState, PWM_GetCurrentPowerState, PWM_GetTargetPowerState, PWM_PreparePowerState, PWM_Main_PowerTransitionManager`), indicating if the HW offers low power state management.		
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		

Limitation: Because power state control is not supported by the hardware, this parameter is always set to FALSE.

T2MC-14416 - [ECUC_Pwm_00143] Config parameter: `PwmPowerStateAsynchTransitionMode`

Title: [ECUC_Pwm_00143] Config parameter: `PwmPowerStateAsynchTransitionMode`

Description:

SWS Item	**ECUC_Pwm_00143 :**
Name	PwmPowerStateAsynchTransitionMode
Description	Enables / disables support of the PWM Driver to the asynchronous power state transition.
Multiplicity	0..1
Type	`EcucBooleanParamDef`
Default value	false
Post-build variant multiplicity	false

Limitations

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 `PwmLowPowerStatesSupport` is set to true.		

Limitation: Because power state control is not supported by the hardware, `PwmPowerStateAsynchTransitionMode` is always set to FALSE.

T2MC-14418 - [ECUC_Pwm_00144] Container name: `PwmPowerStateConfig`

Title: [ECUC_Pwm_00144] Container name: `PwmPowerStateConfig`

Description:

SWS Item	**ECUC_Pwm_00144 :**
Container name	`PwmPowerStateConfig`
Description	Each instance of this parameter defines a power state and the callback to be called when this power state is reached.
Configuration parameters	

Limitation: Because power state control is not supported by the hardware, `PwmPowerStateConfig` shall not be configured.

T2MC-14420 - [ECUC_Pwm_00145] Config parameter: `PwmPowerStateReadyCbkRef`

Title: [ECUC_Pwm_00145] Config parameter: `PwmPowerStateReadyCbkRef`

Description:

SWS Item	**ECUC_Pwm_00145 :**		
Name	`PwmPowerStateReadyCbkRef`		
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 `PwmLowPowerStatesSupport` is set to true.		

Limitation: Because power state control is not supported by the hardware, `PwmPowerStateReadyCbkRef` shall not be configured.

T2MC-14419 - [ECUC_Pwm_00146] Config parameter: `PwmPowerState`

Title: [ECUC_Pwm_00146] Config parameter: `PwmPowerState`

Description:

SWS Item	**ECUC_Pwm_00146 :**		
Name	`PwmPowerState`		
Description	Each instance of this parameter describes a different power state supported by the PWM HW. It should be defined by the HW supplier and used by the PWM Driver to reference specific HW configurations which set the PWM 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 `PwmLowPowerStatesSupport` is set to true.		

Limitation: Because power state control is not supported by the hardware, `PwmPowerState` is always set to 0.

T2MC-15251 - [PWM] AUTOSAR C implementation rules

Title: [PWM] 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.

T2MC-90332 - [PWM] Config parameter: `PwmMultipleUpdateInPeriod`

Title: [PWM] Config parameter: `PwmMultipleUpdateInPeriod`

Description:

SWS Item	-		
Name	`PwmMultipleUpdateInPeriod`		
Description	Switch for enabling multiple updates in a period. `PwmMultipleUpdateInPeriod` specifies the behavior when the following functions are called while the channel is waiting for an update by the next period (`PWM_CH_COND_WAIT_UPDATE`). - `Pwm_SetPeriodAndDuty` - `Pwm_SetDutyCycle` - `Pwm_SetTriggerDelay` - `Pwm_SetOutputOffset` If `PwmMultipleUpdateInPeriod` is true, the function overwrites the register with the new value. PWM channel continues the output waveform. There is a risk of abnormal waveform output if API is continuously called at intervals shorter than one period. If false, the PWM channel stops and restarts with the new value. **Note**: In the following cases, the channel transitions to the wait for update state (`PWM_CH_COND_WAIT_UPDATE`) until the next period. - Calling `Pwm_SetTriggerDelay`, `Pwm_SetOutputOffset` and `Pwm_SetChannelOutput` - Calling `Pwm_Init` or `Pwm_StartGroupTrigger` with any start delay configured channel - Calling `Pwm_SetPeriodAndDuty` or `Pwm_SetDutyCycle` when `PwmPeriodUpdatedEndperiod` is true		
Multiplicity	1		
Type	`EcucBooleanParamDef`		
Default value	False		
Post-build variant value	False		
Value configuration class	***Pre-compile time***	X	All Variants
	Link time	--	
	Post-build time	--	
Scope / dependency	scope: local		

Limitation: When `PwmMultipleUpdateInPeriod` is TRUE and one of `Pwm_SetPeriodAndDuty/Pwm_SetDutyCycle/Pwm_SetTriggerDelay/Pwm_SetOutputOffset` API is called two or more times for the channel during one cycle, the channel output waveform may be corrupted. There is no notification, report, or status for this.

T2MC-17896 - [PWM] Function definition: `Pwm_SetPrescaler`

Title: [PWM] Function definition: `Pwm_SetPrescaler`

Description: The `Pwm_SetPrescaler` service selects the prescaler used when switching from one clock source to another.

- Service name: `Pwm_SetPrescaler`
- Return value: None
- Parameters (in): `Pwm_ChannelType ChannelNumber`, `Pwm_ClkFrequencyType ClockFrequency` (For type definition, see SWASPwm)
- Service ID[hex]: 0x44
- Sync/Async: Synchronous
- Reentrancy: Reentrant (but not for the same timer channel)

The API changes a pre-scaling value of the selected channel by the specified input clock source frequency for TCPWM.
The `Pwm_SetPrescaler` function can be configured as On/Off with the configuration parameter: `PwmSetPrescaler`.

The API shall report the DET error `PWM_E_UNINIT` if this API is called before having initialized the HW unit.
If `Pwm_SetPrescaler` is called with the `ChannelNumber` parameter that does not correspond to a configured channel, the `PWM_E_PARAM_CHANNEL` error shall be reported.
If `Pwm_SetPrescaler` is called with the `ClockFrequency` parameter that is 0 or an incorrect value, the `PWM_E_PARAM_CLOCK` error shall be reported.

If `Pwm_SetPrescaler` is called while waiting for the trigger for synchronous start/stop, the `PWM_E_WAITING_TRIGGER` error shall be reported.

This API is used to change the prescaler of TCPWM in runtime when the input clock frequency changes according to the MCU module and is necessary to reflect on the TCPWM.
When the setting TCPWM prescaler is incorrect and is different from the `ClockFrequency` API parameter and the real clock frequency value, the expected action of timer cannot execute.

This API assumes that it will be called on a stopped channel. If the API is called on a channel that has not been stopped, the channel will be restarted after it is stopped.

Limitation: This API stops the channel once to change the pre-scaling value. This is because of the following:

- There is no specific API to stop the channel
- The pre-scaling register is static; it can be set only when the counter is not running.

T2MC-90327 - [PWM] The `Pwm_Init` function shall start with group start delay settings

Title: [PWM] The `Pwm_Init` function shall start with group start delay settings

Description: If `PwmGroupStartDelay` is enabled, the `Pwm_Init` function shall start with group start delay settings. If this API does not start the channel, the output level is set to `PwmIdelState` until the channel starts. The output level during group start delay is the inverse of the `PwmPolarity` level.

Note: *The API does not start the channel if `PwmStartAtInit` is set to FALSE or `PwmStartAtInit` is set to TRUE and the channel is waiting for `Port_ActTrigger()`.*

Limitation: The notification will be generated immediately after starting the channel for the following conditions:

- Enabled the notification function that is configured by the group start delay
- Configured value of `PwmAlignment` is `PWM_LEFT_ALIGND`
- Disabled `PwmOutputOffset` configuration

The group start delay may be repeated, when the group start delay channel was started without any input trigger.

T2MC-90328 - [PWM] The `Pwm_StartGroupTrigger` function shall start with group start delay settings

Title: [PWM] The `Pwm_StartGroupTrigger` function shall start with group start delay settings

Description: If `PwmGroupStartDelay` is enabled, the `Pwm_StartGroupTrigger` function shall start with group start delay settings. If the API does not start the channel, the output level is set to `PwmIdelState` until the channel starts. The output level during group start delay is the inverse of the `PwmPolarity` level.

Note: The API does not start the channel if `PwmChannelGroupStartTrigger` is enabled.

Limitation: The notification will be generated immediately after starting the channel for the following conditions:

- Enabled the notification function that is configured by the group start delay
- Configured value of `PwmAlignment` is `PWM_LEFT_ALIGND`
- Disabled `PwmOutputOffset` configuration

The group start delay may be repeated, when the group start delay channel was started without any input trigger.

T2MC-14391 - [SWS_Pwm_00045] If the PWM channel is not of type `PWM_VARIABLE_PERIOD`, the `Pwm_SetPeriodAndDuty`() function shall call `PWM_E_PERIOD_UNCHANGEABLE` development error

Title: [SWS_Pwm_00045] If the PWM channel is not of type `PWM_VARIABLE_PERIOD`, the `Pwm_SetPeriodAndDuty`() function shall call `PWM_E_PERIOD_UNCHANGEABLE` development error

Description: [SWS_Pwm_00045] [If default error detection for the Pwm module is enabled: The API `Pwm_SetPeriodAndDuty`() shall check if the given PWM channel is of the channel class type `PWM_VARIABLE_PERIOD`. If this is not the case the development error `PWM_E_PERIOD_UNCHANGEABLE` shall be called.] (SRS_BSW_00323, SRS_BSW_00386)

Limitation: Because DET error detection mechanism is used as a safety mechanism (fault detection), the detection of the development error is always executed.

T2MC-14392 - [SWS_Pwm_00047] If the `ChannelNumber` parameter is invalid, PWM functions shall raise the `PWM_E_PARAM_CHANNEL` development error

Title: [SWS_Pwm_00047] If the `ChannelNumber` parameter is invalid, PWM functions shall raise the `PWM_E_PARAM_CHANNEL` development error

Description: [SWS_Pwm_00047] [If default error detection for the Pwm module is enabled: The PWM functions shall check the parameter `ChannelNumber` and raise development

---

error `PWM_E_PARAM_CHANNEL` if the parameter `ChannelNumber` is invalid.] (SRS_BSW_00323, SRS_BSW_00386)

Limitation: Because DET error detection mechanism is used as a safety mechanism (fault detection), the detection of development error is always executed.

T2MC-14252 - [SWS_Pwm_00094] Imported type

Title: [SWS_Pwm_00094] Imported type

Description: [SWS_Pwm_00094]
[

Module	***Imported type***
Dem	Dem_EventIdType
	Dem_EventStatusType
Std_Types	Std_ReturnType
	Std_VersionInfoType

] ()

Limitation: Dem is not used for PWM since there are no production errors.

T2MC-14376 - [SWS_Pwm_00104] Optional interface: `Dem_ReportErrorStatus` API function

Title: [SWS_Pwm_00104] Optional interface: `Dem_ReportErrorStatus` API function

Description: [SWS_Pwm_00104]
[

API function	***Description***
Dem_ReportErrorStatus	Queues the reported events from the BSW modules (API is only used by BSW modules). The interface has an asynchronous behavior, because the processing of the event is done within the Dem main function. OBD Events Suppression shall be ignored for this computation.
Det_ReportError	Service to report development errors.

] ()

Limitation: Dem is not used for PWM because there are no production errors.

T2MC-14281 - [SWS_Pwm_00116] No function call of the Pwm module before having called `Pwm_Init`

Title: [SWS_Pwm_00116] No function call of the Pwm module before having called `Pwm_Init`

Description: [SWS_Pwm_00116] [The Pwm module's environment shall not call any function of the Pwm module before having called `Pwm_Init`.] ()

Limitation: The `Pwm_GetChannelStatus` safety function and the `Pwm_GetVersionInfo` function are the only functions which can be called before `Pwm_Init` has been executed.

T2MC-14390 - [SWS_Pwm_00117] If PWM module is not initialized, the called function shall raise the `PWM_E_UNINIT` development error

Title: [SWS_Pwm_00117] If PWM module is not initialized, the called function shall raise the `PWM_E_UNINIT` development error

Description: [SWS_Pwm_00117] [If default error detection for the Pwm module is enabled: If any function (except `Pwm_Init`) is called before `Pwm_Init` has been called, the called function shall raise development error `PWM_E_UNINIT`.] (SRS_BSW_00406, SRS_BSW_00323, SRS_BSW_00386)

Limitation: Because DET error detection mechanism is used as a safety mechanism (fault detection), the detection of development error is always executed.

T2MC-14268 - [SWS_Pwm_00165] Type definition: `Pwm_PowerStateRequestResultType`

Title: [SWS_Pwm_00165] Type definition: `Pwm_PowerStateRequestResultType`

Description: [SWS_Pwm_00165]
[

Name:	`Pwm_PowerStateRequestResultType`	
Type:	Enumeration	
Range:	`PWM_SERVICE_ACCEPTED`	Power state change executed.
	`PWM_NOT_INIT`	PWM Module not initialized.
	`PWM_SEQUENCE_ERROR`	Wrong API call sequence.
	`PWM_HW_FAILURE`	The HW module has a failure which prevents it to enter the required power state.
	`PWM_POWER_STATE_NOT_SUPP`	PWM Module does not support the requested power state.
	`PWM_TRANS_NOT_POSSIBLE`	PWM Module cannot transition directly from the current power state to the requested power state or the HW peripheral is still busy.
Description:	Result of the requests related to power state transitions.	

] ()

Limitation: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this type definition shall not be used.

T2MC-14270 - [SWS_Pwm_00197] Type definition: `Pwm_PowerStateType`

Title: [SWS_Pwm_00197] Type definition: `Pwm_PowerStateType`

Description: [SWS_Pwm_00197]
[

Name:	`Pwm_PowerStateType`	
Type:	Enumeration	
Range:	PWM_FULL_POWER	Full Power (0)
	1..255	Power modes with decreasing power consumptions.
Description:	Power state currently active or set as target power state.	

Mandatory parameters:

- Assigned HW channel
- Default value for period
- Default value for duty cycle
- Polarity (high or low)
- Idle state high or low
- Channel class:
- Fixed period
- Fixed period, shifted (if supported by hardware)
- Variable period

Optional parameters (if supported by hardware):

- Channel phase shift
- Reference channel for phase shift
- Microcontroller specific channel properties

] (SRS_Pwm_12293, SRS_Pwm_12378)

Limitation: Because power state control is not supported by the hardware, `PwmLowPowerStatesSupport` is always set to FALSE; this type definition shall not be used.

T2MC-14276 - [SWS_Pwm_10009] The `Pwm_Init` function shall start all PWM channels

Title: [SWS_Pwm_10009] The `Pwm_Init` function shall start all PWM channels

Description: [SWS_Pwm_10009] [The function `Pwm_Init` shall start all PWM channels with the configured default values.] (SRS_SPAL_12057)

Limitation: If the `PwmStartAtInit` of the channel is disabled, it will not be started by `Pwm_Init`.

T2MC-14387 - [SWS_Pwm_10051] The PWM function shall report the error to the development error tracer if a development error occurs

Title: [SWS_Pwm_10051] The PWM function shall report the error to the development error tracer if a development error occurs

Description: [SWS_Pwm_10051] [If default error detection for the Pwm module is enabled, and a development error occurs, then the corresponding PWM function shall report the error to the Default Error Tracer.] (SRS_BSW_00323, SRS_BSW_00386)

Limitation: Because DET error detection mechanism is used as a safety mechanism (fault detection), the detection of development error is always executed.

T2MC-14388 - [SWS_Pwm_20051] The PWM function shall skip the desired functionality if a development error occurs

Title: [SWS_Pwm_20051] The PWM function shall skip the desired functionality if a development error occurs

Description: [SWS_Pwm_20051] [If default error detection for the Pwm module is enabled, and a development error occurs, then the corresponding PWM function shall skip the desired functionality in order

to avoid any corruptions of data or hardware registers leaving the function without any actions.] (SRS_BSW_00323, SRS_BSW_00386)

Limitation: Because DET error detection mechanism is used as a safety mechanism (fault detection), the detection of development error is always executed.

T2MC-14389 - [SWS_Pwm_30051] The corresponding PWM function shall return `PWM_LOW` for the `Pwm_GetOutputState` function if a development error occurs

Title: [SWS_Pwm_30051] The corresponding PWM function shall return `PWM_LOW` for the `Pwm_GetOutputState` function if a development error occurs

Description: [SWS_Pwm_30051] [If default error detection for the Pwm module is enabled, and a development error occurs, then the corresponding PWM function shall return `PWM_LOW` for the function `Pwm_GetOutputState`.] (SRS_BSW_00323, SRS_BSW_00386)

Limitation: Because DET error detection mechanism is used as a safety mechanism (fault detection), the detection of development error is always executed.

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.

8 Version history

8.1 Module SW-Version 1.3

Initial module setup.

8.2 Module SW-Version 1.4

T2MC-38077 - [GPT, ICU, PWM] Wrong config data is generated when ordering of configuration is changed

Title: [GPT, ICU, PWM] Wrong config data is generated when ordering of configuration is changed

Description: In Tresos GUI, when the order of configuration is changed intentionally the following configuration lists, the configuration data is generated by wrong order. In such cases, the module API cannot operate correctly. It should be generated not by the index list but by the channel ID.

GPT: `GptChannelConfiguration`
ICU: `IcuChannelGroup`
PWM: `PwmChannelGroup`

T2MC-39159 - [PWM] PWM output might be not restarted after calling `Pwm_SetChannelOutput`

Title: [PWM] PWM output might be not restarted after calling `Pwm_SetChannelOutput`

Description: When there is following sequence, PWM output is not restarted after calling `Pwm_SetChannelOutput`.

1. Pwm_SetOutputToIdle()
2. Pwm_SetChannelOutput(pwm_count, PWM_LOW)
3. Pwm_StartGroupTrigger(PwmConf_PwmChannelGroup_PwmChannelGroup_0)

The output is LOW in all time after 3.

T2MC-38120 - [PWM] Wrong output might occur when the period tick is 0 or 1

Title: [PWM] Wrong output might occur when the period tick is 0 or 1

Description: Wrong output might occur in PWM. The defect occurrence condition is either one of the following.

API `Pwm_SetPeriodAndDuty`:

- The argument "Period" is specified by 0.
- The argument "Period" is specified by 1 with center alignment.

API Pwm_Init:

- The channel configuration is specified by the following all conditions.
  "PwmPeriodDefault" is equivalent to 1-tick time.
  "PwmStartAtInit" is true.
  "PwmAlignment" is PWM_CENTER_ALIGNED.

API `Pwm_SetDutyCycle`/`Pwm_SetPeriodAndDuty`/`Pwm_Init` ("PwmStartAtInit" is true):
Case1:

- "PwmPeriodDefault" is equivalent to 0x10000-tick time or the argument "Period" is specified by 0x10000.

- "PwmDutycycleDefault" is configured as 0x8000 or the argument "DutyCyle" is specified by 0x8000.
- The timer counter width is 16 bits.
- "PwmAlignment" is PWM_LEFT_ALIGNED.

Case2:

- "PwmPeriodDefault" is equivalent to 0x10000-tick time or the argument "Period" is specified by 0x10000.
- "PwmDutycycleDefault" is configured as 0x0 or the argument "DutyCyle" is specified by 0x0.
- "PwmAlignment" is PWM_RIGHT_ALIGNED.

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.

8.3 Module SW-Version 1.5

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

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

Description: Checking for valid C function name:
Check all configuration parameters related to the function name to see if it is a valid C function name.
A part of parameters are not checked.
If an invalid function name is set, a compile error will occur during the build process, which is inconvenient for users.
Therefore, it is better to check whether the configured function names are valid C function names in advance (i.e. during configuration phase).

Checking for valid filename:
Check all configuration parameters related to the file name to see if it is valid.
A part of parameters cannot check the fact that empty file name (i.e. ".h") is wrong.
If an invalid file name is set, a compile error will occur during the build process, which is inconvenient for users.
Therefore, it is better, to check in advance, whether the configured file names are valid.

This CR is intended to solve the inconvenience.

T2MC-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-43302 - [GPT, ICU, OCU, PWM] Improve trigger configuration check

Title: [GPT, ICU, OCU, PWM] Improve trigger configuration check

Description: If the input trigger resource for TCPWM is shared by other modules, a warning message is output. To improve usability and prevent misuse, make conflict processing clear when it is shared in related config parameters.

The related configurations are as follows. For more details, please see the attached file.

- GPT
  - GptPredefTimerStartTriggerSelect
  - GptInputTriggerSelection
- ICU
  - IcuInputTriggerSelection
  - IcuChannelGroupStartTrigger
  - IcuChannelGroupStopTrigger
- OCU
  - OcuStartTriggerSelect0
  - OcuStartTriggerSelect1
- PWM
  - PwmChannelGroupStartTrigger
  - PwmChannelGroupStopTrigger
  - PwmStartTriggerSelect0
  - PwmStartTriggerSelect1
  - PwmStartDelayTrigger

T2MC-39475 - [ICU, OCU, PWM] Wrong behavior occurred due to API input while waiting for synchronous start/stop trigger

Title: [ICU, OCU, PWM] Wrong behavior occurred due to API input while waiting for synchronous start/stop trigger

Description: Malfunction occurred due to other API input while waiting for synchronous start trigger.

T2MC-39474 - [PMW] StartDelay and OutputOffset do not work in some cases

Title: [PMW] StartDelay and OutputOffset do not work in some cases

Description: Pwm driver might output wrong PWM signal in Start delay and Offset delay functions.

1. When the value of `PwmChannelStartDelay` is other than 0, PWM output might be wrong.
   E.g. Left alignment NG case: StartDelay tick >= Period - Duty tick
2. When `Pwm_SetOutputOffset` API is called during the running status, PWM output might be wrong.
3. It does not meet the following requirement.
   MCAL_PWMDIAG_04: When changing the period during runtime, it shall be configurable if the ratio between the rising edge offset and the period shall be kept.

Besides, some of Start delay and Offset delay functions are restricted as follows. These restrictions can be removed based on the results of the review.

4. Offset delay function (`PwmOutputOffset` and `Pwm_SetOutputOffset` API) is not allowed to use in PwmChannelClass PWM_FIXED_PERIOD.
5. `PwmChannelStartDelay` and `PwmOutputOffset` are not allowed to use at the same time.

T2MC-38072 - [PWM, OCU] Outputs a momentary level change under certain condition

Title: [PWM, OCU] Outputs a momentary level change under certain condition

Description: PWM/OCU driver may output a momentary level change under certain conditions.

Momentary output level change may occur when the following APIs are called under certain conditions.
PWM API:
Pwm_Init, Pwm_SetDutyCycle, Pwm_SetPeriodAndDuty, Pwm_StartGroupTrigger, Pwm_SetOutputToIdle, Pwm_StopGroupTrigger, Pwm_EnableTrigger, Pwm_DisableTrigger, Pwm_SetPrescaler, Pwm_DeInit

OCU API:
Ocu_Init, Ocu_SetPinState, Ocu_SetPinAction, Ocu_StartChannel, Ocu_StopChannel, Ocu_SetPrescaler

---

T2MC-48070 - [PWM] Output is not zero when the period tick is 0

Title: [PWM] Output is not zero when the period tick is 0

Description: The Pwm driver does not match the following requirement when the period tick is 0.
T2MC-14310 - [SWS_Pwm_00150] If the period is set to zero the output shall be zero (zero percent duty-cycle)

T2MC-39458 - [PWM] Support TRAVEO™ T2G-B-H-8M

Title: [PWM] Support TRAVEO™ T2G-B-H-8M
Description: AUTOSAR MCAL supports the TRAVEO™ T2G-B-H-8M.

TCPWM resource data has instances of TCPWM. Therefore, Pwm driver must identify TCPWM resource instances in the following cases.

- TCPWM resources for all instances need to be obtained from the resource properties file and made selectable in the module configuration.
- Instance support is required for processing associated with TCPWM resources (for example, configuration, generated code, etc.).

In addition to the above, users guide needs update.

T2MC-39854 - [PWM] When 2 APIs are called in same period, PWM driver might be wrong output.

Title: [PWM] When 2 APIs are called in same period, PWM driver might be wrong output.

Description: When 2 APIs in following are called in a same period, PWM driver might be wrong output.

```
Pwm_SetDutyCycle()
Pwm_SetPeriodAndDuty()
Pwm_SetOutputOffset()
Pwm_SetTriggerDelay()
Pwm_SetChannelOutput()
```

The second API is called at the boundary of the period.

8.4 Module SW-Version 1.6

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

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

Description: Any device-dependent information should not be included in the user guide.
Therefore, delete the datasheet name from the related documentation in the user guide.

T2MC-50756 - [PWM] Limit of OutputOffset and AdcTrigger is not correct

Title: [PWM] Limit of OutputOffset and AdcTrigger is not correct

Description: Limit of PwmOutputOffset, PwmAdcTriggerDefault and PwmAdcTriggerDefaultTick is not correct.

T2MC-51795 - [PWM] When PwmChannelStartDelay is blank, error of tresos is occurred at other resource check

Title: [PWM] When PwmChannelStartDelay is blank, error of tresos is occurred at other resource check

Description: When PwmChannelStartDelay is blank, error of tresos is occurred at other resource check.

T2MC-51852 - [PWM] User guide missing/wrong description

Title: [PWM] User guide missing/wrong description

Description: PWM user guide missing/wrong description

4.2.2 PwmPeriodDefault: [wrong]

Pwm_Cfg.h is wrong. Pwm_PBcfg.h is right.

"equal to or less than" and "65535" are wrong. "less than" and "65536" are right.

5.16 Reentrancy: [missing]

Note

The PWM module's user shall establish a mutual exclusion mechanism in case several function calls are made during run time in different tasks or ISRs targeting the same PWM channel.

8.5 Module SW-Version 1.7

T2MC-59531 - [GPT] Correction of prescale and input trigger selection of external input clock

Title: [GPT] Correction of prescale and input trigger selection of external input clock

Description: In case of external clock input (CLI_EXT), the following configuration parameter should be corrected:

#1. GptChannelPrescale
Prescale value should be fixed to 1 since the prescaler cannot be used in the case of the external clock.

#2. GptInputTriggerSelection
Allow to select trigger multiplexer as clock sources of external input clock for external clock input functional enhancement.

Note: *Modification of #2 affects resource conflict check processing in ICU, OCU and PWM modules (xdm files, user guides).*

T2MC-59564 - [PWM] Channel state inconsistency may occur due to HW state transition delay

Title: [PWM] Channel state inconsistency may occur due to HW state transition delay

Description: By the rate of Pclk and prescale value per MCU clock, the channel state may be inconsistency.

When using following APIs is, the channel state may be inconsistency:

- Pwm_SetPeriodAndDuty
- Pwm_SetDutyCycle
- Pwm_SetTriggerDelay
- Pwm_SetOutputOffset

When using following APIs is, the channel state may be inconsistency at next API:

- Pwm_SetOutputToIdle
- Pwm_StopGroupTrigger

T2MC-59565 - [PWM] Channels of different ConfigSet might be listed in PwmChannelRef

Title: [PWM] Channels of different ConfigSet might be listed in PwmChannelRef

Description: When configset name is partly same as other configset name, other configset channel is selectable.

T2MC-59645 - [PWM] If period is max, Pwm output signal might be incorrectly

Title: [PWM] If period is max, Pwm output signal might be incorrectly

Description: When the following conditions are satisfied, Pwm output signal will be incorrectly:

(1) Period is max. (65535ticks) and

(2) PwmChannelStartDelay is enable. and

(3) DutyCycle is 0 or 32768. and

---

(4) PwmOutputOffset is enable. and

(5) The channel is running.

(6) Either of following factors are satisfied:

(6-a) Pwm_GetChannelStatus() is called.

Then, Pwm_GetChannelStatus() will return PWM_CH_COND_WAIT_UPDATE.

(6-b) Pwm_SetChannelOutput is called.

Then, the channel will ignore this call.

(6-c) Pwm_SetDutyCycle(), Pwm_SetPeriodAndDuty() or Pwm_SetOutputOffset() is called.

Then, the channel will restart.

The followings are the specification of Pwm configurations and APIs after corrected the defect:

Configuration:

- Max of PwmPeriodDefault value is decrease. (The concretely value depends to McuClockReferencePointFrequency.)
- Max of PwmHwTriggerOutputDefaultTime is decrease. (The concretely value depends to McuClockReferencePointFrequency.)
- Max of PwmHwTriggerOutputDefaultTick is one decrease. (Max is 65534)
- Max of PwmOutputOffset is one decrease. (Max is 65533)

API:

- Pwm_SetOutputOffset(): Max of OffsetTick parameter is one decrease. (Max is 65533)
- Pwm_SetTriggerDelay(): Max of TriggerTicks parameter is one decrease. (Max is 65534)
- Pwm_SetPeriodAndDuty(): Max of Period parameter is one decrease. (Max is 65534)

T2MC-59640 - [PWM] When Pwm configurations are same ICU and GPT input trigger, Pwm don't output error

Title: [PWM] When Pwm configurations are same ICU and GPT input trigger, Pwm don't output error

Description: When listed-(1) configurations are same listed-(2), Pwm driver don't output error.

(1) PwmStartTriggerSelect0, PwmStartTriggerSelect1, PwmStartDelayTrigger, PwmChannelGroupStartTrigger, PwmChannelGroupStopTrigger

(2) IcuInputTriggerSelection, IcuChannelGroupStartTrigger, IcuChannelGroupStopTrigger, GptPredefTimerStartTriggerSelect

---

T2MC-57013 - [PWM] Improve hardware trigger output

Title: [PWM] Improve hardware trigger output

Description: Trigger output shall be improved for flexibility: All trigger destinations shall be supported, not just ADC.
Therefore, following changes shall be made:

1. Configuration parameter PwmHardwareTriggeredAdc shall be renamed to PwmHwTriggerOutputLine.

---

2. PwmHwTriggerOutputLine shall allow all trigger outputs that are connectable to any HW (not just ADC) on the particular chip.
3. PwmAdcTriggerFactor shall be renamed to PwmHwTriggerOutputFactor.
4. PwmAdcTriggerDefault shall be renamed to PwmHwTriggerOutputDefaultTime.
5. PwmAdcTriggerDefaultTick shall be renamed to PwmHwTriggerOutputDefaultTick.
6. PwmAdcTriggerDefaultType shall be renamed to PwmHwTriggerOutputUnitType.

8.6 Module SW-Version 1.8

T2MC-65849 - [PWM] The center aligned channel with output trigger can be wrongly enabled

Title: [PWM] The center aligned channel with output trigger can be wrongly enabled

Description: When PWM is configured, the center aligned channel with disabled PwmOutputOffset must have GRP_AMC_PRESENT resource for PwmHwTriggerOutputLine.

However, the channel without GRP_AMC_PRESENT resource can be wrongly enabled for PwmHwTriggerOutputLine.

When PwmHwTriggerOutputLine is enabled in its channel, output trigger does not occur.

Also, the interrupt functions (Pwm_Isr_[IRQ Number]_Cat1 and Pwm_Isr_[IRQ Number]_Cat2) are generated to source file even when PwmNotification is disabled.

T2MC-77594 – Support IAR compiler

Title: Support IAR compiler

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

T2MC-82019 - Notification is not called when output signal becomes constant

Title: Notification is not called when output signal becomes constant

Description: When duty cycle is changed to 0% or 100% by Pwm_SetPeriodAndDuty, output signal becomes constant. The correct behavior is that notification is not called when signal is constant. For more details, see Rev. *C of Document revision history in TRAVEO™ T2G Family PWM driver user guide.

8.7 Module SW-Version 1.9

T2MC-87886 - [PWM] Addition of group start delay

Title: [PWM] Addition of group start delay

Description: There is a start delay in the channel settings. The resource of start delay needs more than 1 timer and 1 trigger per channel.

Add start delay to group config. The resource of group start delay decreases to 1 timer and 1 trigger per group.

Add PwmGroupStartDelay to the PWM channel group config.

PwmGroupStartDelay has sub-configs. (PwmGroupStartDelayTimer, PwmGroupStartDelayMcuClockReferencePoint, PwmGroupStartDelayTrigger)

Pwm_Init and Pwm_StartGroupTrigger start with group start delay settings. Pwm_SetPeriodAndDuty and Pwm_SetDutyCycle start without group start delay settings.

Each channel during group start delay, outputs PwmIdleState and no output trigger. Each channel is switched to normal period and duty output trigger settings by PwmGroupStartDelayTrigger after the group start delay time.

PwmGroupStartDelayTimer resource for switching is shared in the PWM channel group.

(PwmStartDelayTimer, PwmStartDelayMcuClockReferencePoint, PwmStartDelayTrigger) and (PwmGroupStartDelayTimer, PwmGroupStartDelayMcuClockReferencePoint, PwmGroupStartDelayTrigger) are an exclusive function.

When PwmChannelClass is PWM_FIXED_PERIOD_SHIFTED or PWM_VARIABLE_PERIOD, start delay and group start delay can be set to enable.

[Impact on the configuration interface]

If an existing project does not use PwmStartDelayTimer in the PWM configuration, errors may occur with importing. In that case, Sets PwmStartDelayTimer to disable.

T2MC-87888 - [PWM] Addition of multiple updates for channel in waiting update state

Title: [PWM] Addition of multiple updates for channel in waiting update state

Description: For the following waiting channel to update conditions, a new parameter PwmMultipleUpdateInPeriod is added to the PWM general configuration:

- Call Pwm_Init API including any of channels with start delay
- Call Pwm_StartGroupTrigger API including any of channels with start delay
- Call Pwm_SetPeriodAndDuty API when PwmPeriodUpdatedEndperiod is true
- Call Pwm_SetDutyCycle API when PwmDutycycleUpdatedEndperiod is true
- Call Pwm_SetTriggerDelay API
- Call Pwm_SetOutputOffset API
- Call Pwm_SetChannelOutput API

If the added parameter PwmMultipleUpdateInPeriod is false (default), the waiting channel is stopped immediately and restarted as per previous specifications, when the related APIs (that were on hold to be updated) are called.

If PwmMultipleUpdateInPeriod is true, the output waveform and the output trigger of the waiting channel is kept stable (not stopped and not restarted), when the related APIs (that were on hold to be updated) are called.

The following are the related APIs: Pwm_SetPeriodAndDuty, Pwm_SetDutyCycle, Pwm_SetTriggerDelay, and Pwm_SetOutputOffset.

T2MC-87885 - [PWM] Addition of trigger delay scaling config

Title: [PWM] Addition of trigger delay scaling config

Description: A new parameter PwmHwTriggerOutputScaled is added to the PWM channel configuration.

If PwmHwTriggerOutputScaled is true (default), the trigger delay is changed by the Pwm_SetPeriodAndDuty API as per previous specifications

---

If PwmHwTriggerOutputScaled is false, the trigger delay is not changed by the Pwm_SetPeriodAndDuty API.

8.8 Module SW-Version 1.10

T2MC-91796 - [PWM] Pwm.xdm is inconsistent with Pwm.arxml

Title: [PWM] Pwm.xdm is inconsistent with Pwm.arxml

Description: There are some inconsistent description in *Pwm.xdm* and *Pwm.arxml* files.

Pwm.xdm

- The description of PwmStartTriggerSelect0 and PwmStartTriggerSelect1 is different in *Pwm.xdm* and *Pdm.arxml*.
- IMPLEMENTATIONCONFIGCLASS is incorrect in some parameters.
- POSTBUILDVARIANTMULTIPLICITY is incorrect in some parameters.
- REFINED_MODULE_DEF is incorrect.

Pwm.arxml

- The description of PwmDutycycleDefault is different in *Pwm.xdm* and *Pwm.arxml*.
- MULTIPLE-CONFIGURATION-CONTAINER and MULTIPLICITY-CONFIG-CLASSES and POST-BUILD-VARIANT-MULTIPLICITY are incorrect in some parameters.
- LOWER-MULTIPLICITY and UPPER-MULTIPLICITY are incorrect in some parameters.

8.9 Module SW-Version 1.11

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

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

Description: Some modules differ in the macro names defined and the macro names used in the processing. For example, when the macro set to TRUE is judged as STD_ON, the definition value is 1 for both, but the same macro must be used.

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

In Platform_Types.h of the base module

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

In Std_Types.h of the base module

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

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

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

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

According to AUTOSAR specification:
[SWS_COMPILER_00026]
#define VAR(vartype, memclass)

True:
volatile P2VAR(Spi_DmaChannelRegsType, AUTOMATIC, REGSPACE) retPtr;

False:
volatile Spi_DmaChannelRegsType * retPtr;

This issue is present in the following cases:

- All types of pointer declaration/definition are defined without macros.

These contain the function parameter/global variable/local variable/structure field/union field.

- All types of function declaration/definition are defined without macros.
- When there is nested macro usage in function macros.
- Raw pointer is used in the function macro:

e.g., FUNC(int *, memclass) function(void);

- Global variable or static variable in the function is not defined with macros.

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

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

T2MC-39519 - Support EB tresos V26.2.0

Title: Support EB tresos V26.2.0

Description: Support EB tresos V26.2.0

[Impact]
Strict AUTOSAR specification and check for parameter configuration errors are implemented in EB tresos V26.2.0.
In addition, handling of reference paths (relative paths) such as system description file (ARXML) is changed in EB tresos V26.2.0.

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

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

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

8.10 Module SW-Version 1.12

T2MC-164408 - Improvement of interrupt register clear processing

Title: Improvement of interrupt register clear processing

Description: Some modules clear the interrupt register by read modify write (RMW). However, there is a possibility that unintended bits might also be cleared, if some bits are already set before clearing, because the attribute of the interrupt register is RW1C (every bit is cleared upon writing 1).
Also, unnecessary read access to the register reduces performance.
Therefore, change the clearing process to write intended bit only.

T2MC-164778 - Support MISRA C:2012 coding rule

Title: Support MISRA C:2012 coding rule

Description: Support MISRA C:2012 coding rule.

The MISRA C:2012 coding rule checks the source code.
If a deviation from the rules is required, add the deviation comment to the code and report the result.
If a deviation is for MISRA-C:2004 only, remove the deviation comment.

8.11 Module SW-Version 1.13

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.

8.12 Module SW-Version 1.14

T2MC-170535 - [PWM] Interrupt status flag cleared at the end of interrupt processing

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

Description: The PWM interrupt function clears the interrupt flag after performing the interrupt processing. So, if an interrupt occurs during interrupt processing, it is not processed. It should be improved to clear the factor at the beginning of the interrupt function to allow handling of factors that occur during PWM interrupts.

If an interrupt occurs during interrupt processing whatever the factor it is, it will be missed and will not be called user-specified notification function.

Workaround:
Make the system premised on interrupts being lost if the period or duty is too short, or disable the notification.

T2MC-170796 - [PWM] Guarantee the order of register settings between relevant peripherals for robustness

Title: [PWM] 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 instructions and peripheral operations.
The PWM driver before Port_ActTrigger is called must guarantee the order of register settings that might be the disorder by the write cash.
Therefore, it was added to the register read back process in order to avoid this issue.

T2MC-170546 - [PWM] Unused structure member in MCAL code

Title: [PWM] Unused structure member in MCAL code

Description: An unused structure member has been found inside MCAL code. A structure member GroupId exists in Pwm_ChannelGroupConfigType, which is not used. This issue would not affect any functions and their behavior. However, the unused structure member should be removed because it is redundant.

T2MC-170845 - Add note on Pwm_SetChannelOutput API to user guide

Title: Add note on Pwm_SetChannelOutput API to user guide

Description: Add the following note on Pwm_SetChannelOutput API to user guide.

If the specified channel is called by the Pwm_SetChannelOutput API at the end of period timing in waiting update state, the output waveform might be disturbed.

In the following cases, the channel is set to waiting update state (PWM_CH_COND_WAIT_UPDATE) until the next period.

- Calling Pwm_SetTriggerDelay, Pwm_SetOutputOffset or Pwm_SetChannelOutput
- Calling Pwm_Init or Pwm_StartGroupTrigger with any start delay configured channel
- Calling Pwm_SetPeriodAndDuty (PwmPeriodUpdatedEndperiod is true)
- Calling Pwm_SetDutyCycle (PwmDutycycleUpdatedEndperiod is true)

Workaround:
Call the Pwm_SetChannelOutput API after the next period.

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 the 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.

8.13 Module SW-Version 1.15

T2MC-172343 - Support PWM stepper motor control

Title: Support PWM stepper motor control

Description: The following API and configs are added as PWM stepper motor control.

The line state and complementary line state are controlled by these APIs and configurations.

Added APIs:

- Pwm_SetDutyAndChannelOutputBuffer
  Prepares to change the duty cycle, line state, and complementary line state of the PWM output waveform.
- Pwm_SetDutyCycleBuffer
  Prepares to change the duty cycle of the PWM output waveform.
- Pwm_SetChannelOutputBuffer
  Prepares to change the line state and complementary line state of the PWM output waveform.

Added Config in PwmChannelGroup:

- PwmChannelGroupSwitchEventTrigger config
  If the Port_ActTrigger API is called with this trigger, the duty cycle and/or line state and complementary line state of the PWM output waveform with Pwm_SetDutyAndChannelOutputBuffer, Pwm_SetDutyCycleBuffer, or Pwm_SetChannelOutputBuffer APIs are automatically changed from the next period.

Added Config in PwmConfigurationOfOptApiServices:

- PwmSetDutyAndChannelOutputBufferApi
  Adds/removes Pwm_SetDutyAndChannelOutputBuffer
- PwmSetDutyCycleBufferApi
  Adds/removes Pwm_SetDutyCycleBuffer
- PwmSetChannelOutputBufferApi
  Adds/removes Pwm_SetChannelOutputBuffer

Changed API:

- Pwm_Init API
  If PwmUpdateOutputAtInitEnable config is TRUE, the first line state and complementary line state of the PWM output waveform are controlled by the PwmOutputAtInitSelect and PwmCompOutputAtInitSelect configurations.

Added Config in PwmChannel:

- PwmUpdateOutputAtInitEnable config
  Disables or enables the PwmOutputAtInitSelect and PwmCompOutputAtInitSelect configurations. The default is disable.

- PwmOutputAtInitSelect config
  Specifies the line state (LOW, HIGH, PWM, PWM_INV, HIGHZ) for Pwm_Init if PwmUpdateOutputAtInitEnable is enabled.
- PwmCompOutputAtInitSelect config
  Specifies the complementary line state (LOW, HIGH, PWM, PWM_INV, HIGHZ) for Pwm_Init if PwmUpdateOutputAtInitEnable is enabled.

The following are supported in release V1.12.0.

T2MC-178684 - Addition of necessary steps before entering DeepSleep mode in the user guide

Title: Addition of necessary steps before entering DeepSleep mode in the user guide

Description: Add the information in the user guide on the API that needs to be called to stop the TCPWM counter before entering DeepSleep mode.

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.

The following are supported in release V1.15.0.

T2MC-183983 - Update copyright notice and disclaimer statement

Title: Update copyright notice and disclaimer statement

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

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

Edition 2022-06-24

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2022 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Go to www.infineon.com/support

Document reference

002-23361 Rev. *O

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.