MCAL User Manual for Pwm_17_GtmCcu6

32-bit TriCore^{TM} AURIX^{TM} TC3xx microcontroller

About this document

Scope and purpose

This User Manual is intended to enable users to integrate the Microcontroller Abstraction Layer (MCAL) software for the TriCore^{TM} AURIX^{TM} family of 32-bit microcontrollers.

This document describes responsibilities of integrator in-charge of integrating MCAL software with the basic software (BSW) stack. This document also provides detailed information on safety, configuration and functions along with examples of usage of significant features.

Note: Detailed information about package installation, safety and other generic information that are common across all modules are provided in MCAL User Manual General.

Intended audience

This document is intended for anyone using the Pwm_17_GtmCcu6 module of the TC3xx MCAL software.

Document conventions

Table 1   Conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, table and figure captions, screen names, windows, dialog boxes, menus, sub-menus
Italics	Denotes variable(s) and reference(s)
`Courier New`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
>	Indicates that a cascading sub-menu opens when you select a menu item
[cover parentID=<alpha numeric value>]	Used for traceability completeness. Reader should ignore these.

Reference documents

This User Manual should be read in conjunction with the following documents:
- AURIX^{TM} TC3xx MCAL User Manual General
- Specification of PWM Driver, AUTOSAR_SWS_PWM_Driver, AUTOSAR Release 4.2.2
- Specification of PWM Driver, AUTOSAR_SWS_PWM_Driver, AUTOSAR Release 4.4.0

Table of contents

Table of contents

Table of contents

Table of contents

1 Pwm_17_GtmCcu6 driver

1.1 User information

1.1.1 Description

The pulse width modulation (PWM) driver is responsible for providing standard services related to the PWM signal generation specified by AUTOSAR. The underlying timer engine of a PWM channel is a GTM (TOM or ATOM slice) or CCU6 (T12 or T13 slice) timer channel. The PWM driver provides UI options to configure the driver parameters as described in the AUTOSAR PWM specification and additional parameters to configure the timer engine. The parameters of the GTM or CCU6 timer slice that must be configured are described in the MCU driver chapter.

1.1.2 Hardware-software mapping

This section describes the system view of the PWM driver and peripherals administered by it.

1 Pwm_17_GtmCcu6 driver
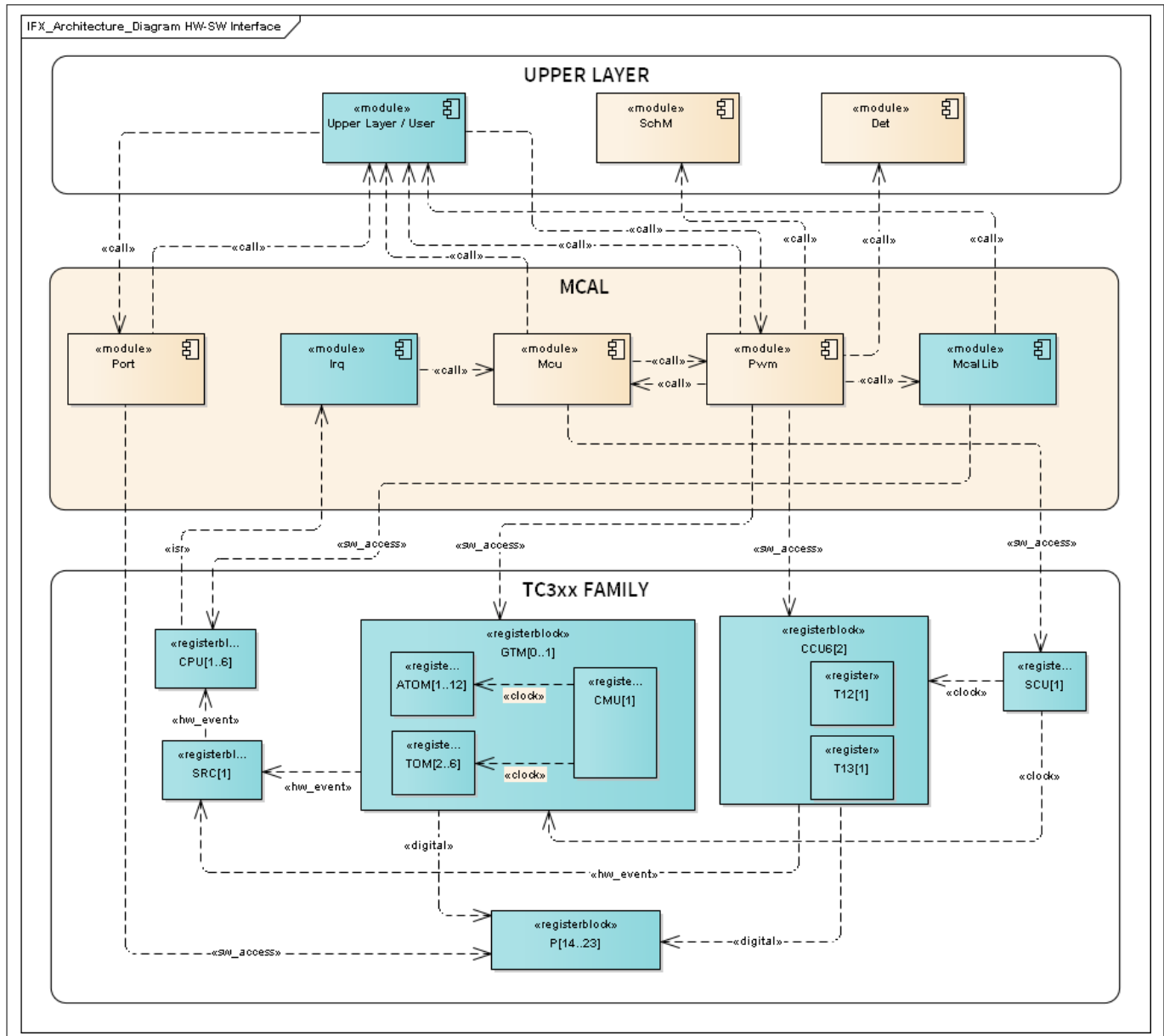
Figure 1   Mapping of hardware-software interfaces

1.1.2.1   Port: dependent hardware peripheral

Hardware functional features

The output signals are routed to the GTM and CCU6 through the port pads. The port pads are configured and enabled through the PORT driver.

Users of the hardware

The port pads are configured by the PORT driver.

Hardware diagnostic features

Not applicable.

Hardware events

Not applicable.

1.1.2.2 GTM: primary hardware peripheral

Hardware functional features

The PWM driver uses the TOM/ATOM sub-module of GTM for generating output signal for a configured duty cycle and period.

The PWM driver uses the TOM/ATOM sub-module for period, duty cycle and polarity updates.

The key GTM features used by the PWM driver are:

- Synchronous and asynchronous update of duty cycles and period
- Host trigger feature is used if a synchronous start is desired
- Signaling mechanisms: Global enable/disable mechanism, output enable mechanism
- Trigger mechanisms: Host CPU, the internal trigger signal (from current channel)
- Continuous counting up mode
- ATOM operation mode: Signal output mode PWM (SOMP)
- ATOM CPU controlled update

The unsupported features of the GTM are:

- Global force update signaling mechanism
- The TBU time stamp trigger mechanism
- Continuous counting up-down mode, one shot up mode, one shot down mode
- ATOM operational mode: SOMI/SOMC/SOMS/SOMB
- ATOM ARU controlled update
- Pulse count modulation mode
- Trigger generation
- TOM BLDC support
- TOM gated counter mode

Users of the hardware

The TOM/ATOM channel of the GTM is exclusively used by the PWM driver. The GTM TOM or ATOM channel can be used by the PWM, GPT, ADC, OCU, DSADC or WDG driver.

The MCU driver initializes the GTM clocks, timer to port pin connections and provides APIs to program the GTM SFRs. The PWM driver uses these APIs to write the GTM SFRs. Additionally, updates to channel-specific SFRs are performed by the PWM driver. Since these channels are exclusively reserved for the PWM driver, access to the channel-specific SFRs from other drivers or user software is not allowed. The MCU driver is responsible to route the GTM interrupt to the PWM driver.

Hardware diagnostic features

Not applicable.

Hardware events

The PWM driver uses the following hardware events from the TOM/ATOM sub-module of the GTM IP:

- Period match events generated by the timer channel
- Compare match events generated by the timer channel

1.1.2.3 SCU: dependent hardware peripheral

Hardware functional features

1 Pwm_17_GtmCcu6 driver

The PWM driver depends on the SCU IP for the clock functionality. The driver requires the fSPB, fGTM and fCCU6 clock signals for functioning.

Users of the hardware

The SCU IP supplies clock for all the peripherals and the MCU driver, and is responsible for configuring the clock tree. To avoid conflicts due to simultaneous writes, update to all the ENDINIT protected registers is performed using the MCALLIB APIs.

Hardware diagnostic features

The SMU alarms configured for the SCU IP are not monitored by the PWM driver.

Hardware events

Hardware events from the SCU are not used by the PWM driver.

1.1.2.4   CCU6: primary hardware peripheral

Hardware functional features

The PWM driver uses the CCU6 for generating output signal for a configured duty cycle and period.

The main features of the CCU6 functional block configured and accessed by the PWM driver are period, duty cycle and polarity update.

The key CCU6 features used by the PWM driver are:

- Edge-aligned operating mode

- Compare mode

- Interrupt generation at period-match and compare-match events

- Synchronous start: Host trigger feature is used if a synchronous start is required a. for T12 channels b. from T12 to T13 channel

The unsupported features of the CCU6 are:

- Hall sensor mode
- Trap handling of CCU6
- Capture mode for T12 and T13
- Multi-channel mode
- Center-aligned operating mode
- Generation of three-phase PWM
- Dead-time control
- Single-shot mode
- Start control by external events
- Counting of external events
- Hysteresis-like control mode
- Block commutation for Brushless DC-drives
- Noise filter for position input signals
- Automatic rotational speed measurement and commutation control for block commutation
- Integrated error handling

Users of the hardware

1 Pwm_17_GtmCcu6 driver

The T12/T13 channel of the CCU6 is exclusively used by the PWM driver. The CCU6 T12 or T13 channel can be used by the PWM, ICU driver.

The MCU driver provides APIs to program the CCU6 SFRs. The PWM driver uses these APIs to write the CCU6 SFRs. Additionally, updates to channel-specific SFRs are performed by the PWM driver. Since these channels are exclusively reserved for the PWM driver, access to the channel-specific SFRs from other drivers or user software is not allowed. The MCU driver is responsible to route the CCU6 interrupt to the PWM driver.

Hardware diagnostic features

Not applicable.

Hardware events

The PWM driver uses the following hardware events from the CCU6 IP:

- Period match events generated by the timer channel

- Compare match events generated by the timer channel

1.1.3 File structure

1.1.3.1 C file structure

This section provides details of the C files of the PWM driver.

1 Pwm_17_GtmCcu6 driver
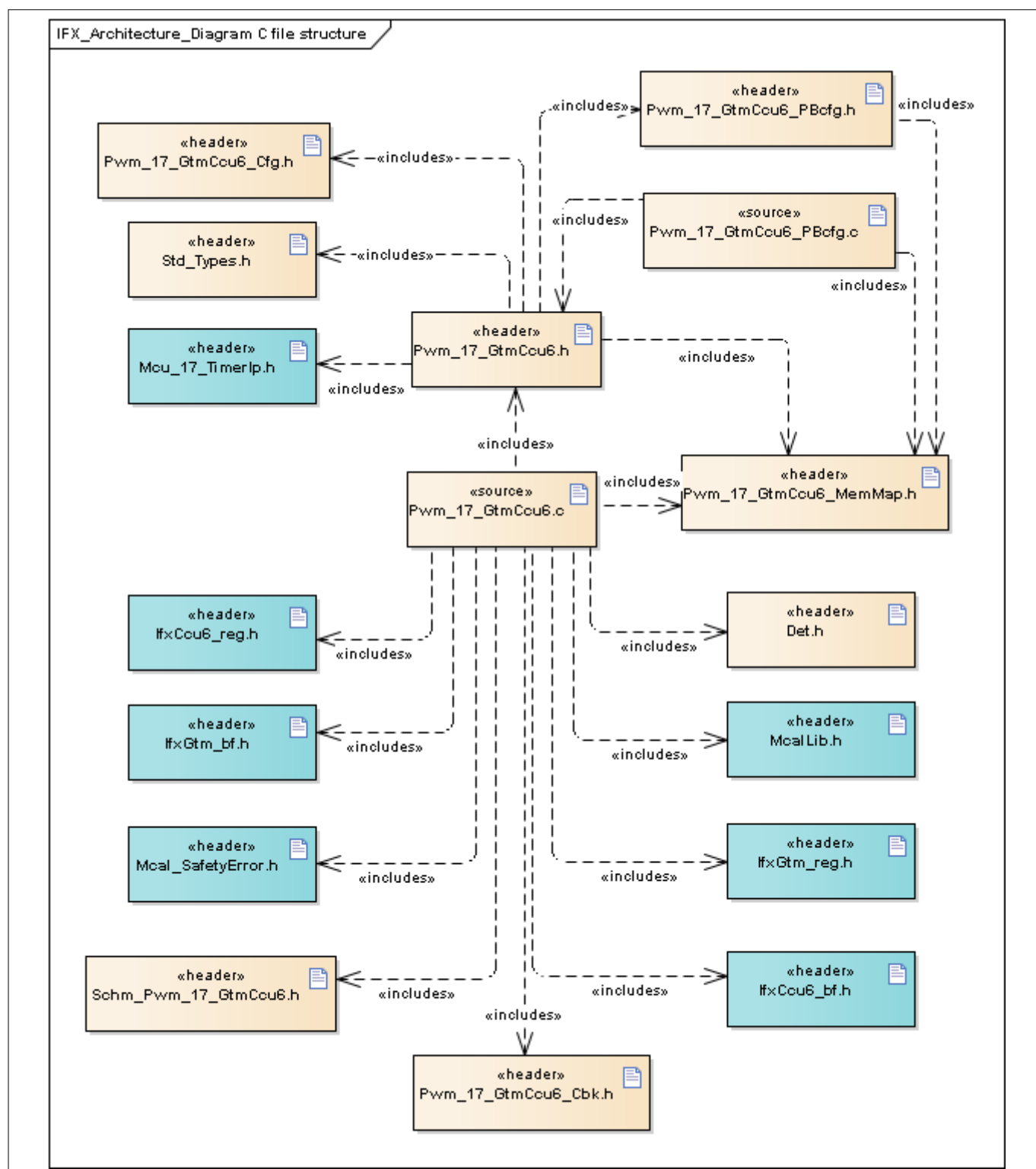
Figure 2   Pwm_C_file_structure-1.png

Table 2   C file structure

File name	Description
Det.h	Provides the exported interfaces of Development Error Tracer
IfxCcu6_bf.h	SFR header file for CCU6

RESTRICTED

Table 2   C file structure (continued)

File name	Description
IfxCcu6_reg.h	SFR header file for CCU6
IfxGtm_bf.h	SFR header file for GTM
IfxGtm_reg.h	SFR header file for GTM
McalLib.h	Static header file defining prototypes of data structure and APIs exported by the MCALLIB.
Mcal_SafetyError.h	Header file containing the prototype of the API for reporting safety-related errors
Mcu_17_TimerIp.h	Header file defining prototypes of data structures and APIs of Timer IPs (GTM, CCU6 and GPT12), containing functions such as initialization, enable, interrupt handlers and other services and is included by Mcu_17_TimerIp.c source file
Pwm_17_GtmCcu6.c	Static source code containing API definition
Pwm_17_GtmCcu6.h	Header file (Static) defining prototypes of data structures and APIs. Header file implements all pre-processor directives.
Pwm_17_GtmCcu6_Cbk.h	Includes callback header definition
Pwm_17_GtmCcu6_Cfg.h	Header file (Generated) containing constants and pre-processor macros
Pwm_17_GtmCcu6_MemMap.h	Mapping of code and data (variables, constant variables) to specific memory sections
Pwm_17_GtmCcu6_PBcfg.c	File (Generated) containing objects to data structures
Pwm_17_GtmCcu6_PBcfg.h	Code template header file for the PWM driver that should be included by the application
Schm_Pwm_17_GtmCcu6.h	PWM critical sections are declared in this file
Std_Types.h	Standard type declaration file as defined by AUTOSAR. It is independent of compiler or platform.

1.1.3.2   Code generator plugin files

This section provides details of the code generator plugin files of the PWM driver.

1 Pwm_17_GtmCcu6 driver
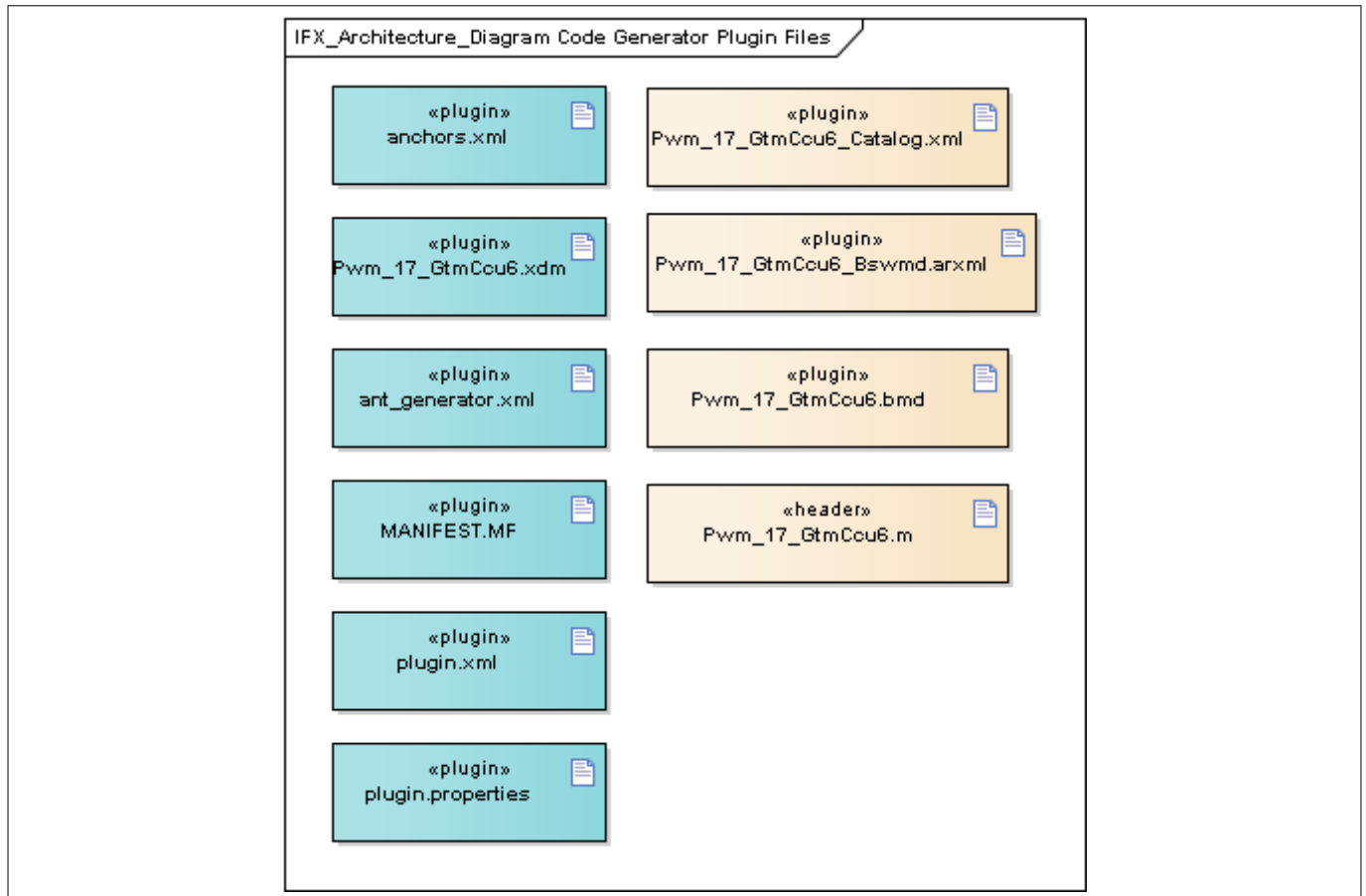
Figure 3   Pwm_Code_Generator_Plugin_Files-1.png

Table 3   Code generator plugin files

File name	Description
MANIFEST.MF	Tresos plugin support file containing the meta data for the PWM driver
Pwm_17_GtmCcu6.bmd	AUTOSAR format XML data model schema file (for each device)
Pwm_17_GtmCcu6.m	File contains macros for the PWM code generation
Pwm_17_GtmCcu6.xdm	Tresos format XML data model schema file
Pwm_17_GtmCcu6_Bswmd.arxml	AUTOSAR format module description file
Pwm_17_GtmCcu6_Catalog.xml	AUTOSAR format catalog file
anchors.xml	Tresos anchors support file for the PWM driver
ant_generator.xml	Tresos support file to generate and rename multiple post-build configuration when using variation point
plugin.properties	Tresos plugin support file for the PWM driver
plugin.xml	Tresos plugin support file for the PWM driver

1 Pwm_17_GtmCcu6 driver

1.1.4 Integration hints

This section lists the key points that an integrator or user of the PWM driver must consider.

1.1.4.1 Intergration with AUTOSAR stack

This section lists the modules, which are not part of MCAL, but are required to integrate the PWM driver.

- **EcuM**

  The ECU Manager module is a part of the AUTOSAR stack that manages common aspects of the ECU. Specifically, in the context of MCAL, the EcuM is used for initialization and de-initialization of the software drivers. The EcuM module provided in the MCAL package is a stub code and needs to be replaced with a complete EcuM module during the integration phase.

  Initialization of PWM:

  The user of PWM driver may use APIs of EcuM to initialize the driver. The initialization of the PWM driver should be invoked from each CPU core, which intends to use the services of the PWM driver. All cores can execute initialization simultaneously.

  De-initialization of PWM:

  The user of PWM driver may use APIs of EcuM to de-initialize the driver. The de-initialization of the PWM driver should be invoked from each CPU core that uses the services of the PWM driver. All cores can execute de-initialization simultaneously.

- **Memory mapping**

  Memory mapping is a concept from AUTOSAR that allows relocation of text, variables, constants and configuration data to user-specific memory regions. To achieve this, all the relocatable elements of the driver are encapsulated in different memory-section macros. These macros are defined in the `Pwm_17_GtmCcu6_MemMap.h` file.

  The `Pwm_17_GtmCcu6_MemMap.h` file is provided in the MCAL package as a stub code. The integrator must place appropriate compiler pragmas within the memory-section macros. The pragmas ensure that the elements

1 Pwm_17_GtmCcu6 driver

are relocated to the correct memory region. A sample implementation listing the memory-section macros is shown as follows:

```
/**** GLOBAL RAM DATA -- NON-CACHED LMU ****/

#if defined PWM_17_GTMCCU6_START_SEC_VAR_CLEARED_ASIL_B_GLOBAL_UNSPECIFIED
/*****User pragmas here for Non-cached LMU*****/
#undef PWM_17_GTMCCU6_START_SEC_VAR_CLEARED_ASIL_B_GLOBAL_UNSPECIFIED
#undef MEMMAP_ERROR
#elif defined PWM_17_GTMCCU6_STOP_SEC_VAR_CLEARED_ASIL_B_GLOBAL_UNSPECIFIED
#ifdef _TASKING_C_TRICORE_
/*****User pragmas here for Non-cached LMU*****/
#undef PWM_17_GTMCCU6_STOP_SEC_VAR_CLEARED_ASIL_B_GLOBAL_UNSPECIFIED
#undef MEMMAP_ERROR



/**** CORE[x] CONFIG DATA -- PF[x] ****/ /*[x]=0..5*/

#elif defined PWM_17_GTMCCU6_START_SEC_CONFIG_DATA_ASIL_B_CORE[x]_UNSPECIFIED
/*****User pragmas here for PF[x]*****/
#undef PWM_17_GTMCCU6_START_SEC_CONFIG_DATA_ASIL_B_CORE0_UNSPECIFIED
#undef MEMMAP_ERROR
#elif defined PWM_17_GTMCCU6_STOP_SEC_CONFIG_DATA_ASIL_B_CORE0_UNSPECIFIED
/*****User pragmas here for PF[x]*****/
#undef PWM_17_GTMCCU6_STOP_SEC_CONFIG_DATA_ASIL_B_CORE0_UNSPECIFIED
#undef MEMMAP_ERROR
/**** CODE -- PF[x] ****/
#elif defined PWM_17_GTMCCU6_START_SEC_CODE_ASIL_B_GLOBAL



/*****User pragmas here for PF[x]*****/

#undef PWM_17_GTMCCU6_START_SEC_CODE_ASIL_B_GLOBAL
#undef MEMMAP_ERROR
#elif defined PWM_17_GTMCCU6_STOP_SEC_CODE_ASIL_B_GLOBAL
/*****User pragmas here for PF[x]*****/
#undef PWM_17_GTMCCU6_STOP_SEC_CODE_ASIL_B_GLOBAL
#undef MEMMAP_ERROR
#endif
#if defined MEMMAP_ERROR
#error "Pwm_17_GtmCcu6_MemMap.h, wrong pragma command"
#endif
```

- **DET**

  The DET module is a part of the AUTOSAR stack that handles all the development and runtime errors reported by the BSW modules. The PWM driver reports all the development errors to the DET module through the `Det_ReportError()` API. The user of the PWM driver must process all the errors reported to the DET module through the `Det_ReportError()` API.

  The `Det.h` and `Det.c` files are provided in the MCAL package as a stub code and needs to be replaced with a complete DET module during the integration phase.

- **DEM**

1 Pwm_17_GtmCcu6 driver

The DEM module is not required for the integration of the PWM driver.

- **SchM**

  The SchM module is a part of the RTE that manages the BSW scheduler. The PWM driver uses the exclusive areas defined in the `SchM_Pwm_17_GtmCcu6.h` file to protect the SFRs and variables from concurrent accesses from different threads. The SchMs identified for the PWM driver are:

  HandleNotification critical section to protect update of notification related global variable and SFRs

  PeriodAndDutyUpdate critical section to protect update of period and duty related shadow SFRs

  The `SchM_Pwm_17_GtmCcu6.h` and `SchM_Pwm_17_GtmCcu6.c` files are provided in the MCAL package as an example code and needs to updated by the integrator. The user must implement the SchM functions defined by the PWM driver as **suspend / resume** of interrupts for the CPU on which the API is invoked. A sample implementation of the SchM functions is shown as follows:

```
/**** Sample implementation of SchM_Pwm_17_GtmCcu6.c ****/
#include "Os.h"
void SchM_Enter_Pwm_17_GtmCcu6_HandleNotification(void)
{
 /* Start of Critical Section */
 SuspendAllInterrupts();/* Suspend CPU core interrupt */
}
void SchM_Exit_Pwm_17_GtmCcu6_HandleNotification(void)
{
 /* End of Critical Section */
 ResumeAllInterrupts();/* Resume CPU core interrupt */
}
void SchM_Enter_Pwm_17_GtmCcu6_PeriodAndDutyUpdate(void)
{
 /* Start of Critical Section */
 SuspendAllInterrupts();/* Suspend CPU core interrupt */
}
void SchM_Exit_Pwm_17_GtmCcu6_PeriodAndDutyUpdate(void)
{
 /* End of Critical Section */
 ResumeAllInterrupts();/* Resume CPU core interrupt */
}
```

- **Safety error**

  The PWM driver reports all the detected safety errors through the `Mcal_ReportSafetyError()` API.

  The driver performs only detection and reporting of the safety errors. The handling of the reported errors shall be done by the user. The `Mcal_ReportSafetyError()` API is provided in the `Mcal_SafetyError.c` and `Mcal_SafetyError.h` files as a stub code, and must be updated by the integrator to handle the reported errors.

  Note: All DET errors are also reported as safety errors (error code used is same as DET).

- **Notifications and callbacks**

  The PWM driver does not implement any notifications. However, the PWM driver reports the rising/falling/both edges through notification functions. These notification functions can be configured by the user in the EB Tresos for each PWM channel separately.

  The PWM driver does not expect any callbacks from application but it needs a callback from the Mcu module for ISR handling.

- **Operating system(OS)**

  The OS or application must ensure correct type of service and interrupt priority is configured in the SR register. Enabling and disabling of interrupts must also be managed by the OS or application.

  The OS files provided by the MCAL package is only an example code and must be updated by the integrator with the actual OS files for the desired function.

1.1.4.2   Multicore and Resource Manager

The PWM driver supports execution of its APIs simultaneously from all CPU cores. The user should allocate logical PWM channels to the CPU cores at pre-compile time using the Resource manager module. The following are the key points to be considered with respect to multicore in the driver:

- Logical PWM channels are allocated to CPU cores at pre-compile time. For example, Pwm_Channel0, Pwm_Channel1

- It must be ensured that PWM channel id passed as parameter while invoking an API belongs to the same core

- DETs are raised in case APIs are invoked with mismatch of core and channel id

- Interrupts raised by a PWM channel must be serviced by the CPU core to which the channel has been allocated to

- PWM channels using GTM-ATOM, channel GTM-ATOM[i]_CH[X] and ATOM[i]_CH[x+1] must be allocated to the same core as these two channels share the same interrupt line

- PWM channels using GTM-TOM, channel GTM-TOM[i]_CH[X] and TOM[i]_CH[x+1] must be allocated to the same core as these two channels share the same interrupt line

- Locating constants, variables and configuration data to the correct memory space should be done by the user. Memory sections are marked GLOBAL (common to all cores) and CORE[x](specific to a CPU core). The following should be considered by the user to ensure better performance of the driver:

  Code section:

  The executable code of the PWM driver is placed under single MemMap section. It can be relocated to any PFlash.

  Data section:

  The RAM variable memory sections marked as specific to core should be relocated to the DSPR/DLMU of the same core. The sections marked as global should be relocated to the non-cached LMU region.

  Configuration data and constants:

  The configuration data section sections marked as specific to core should be relocated to the PFlash of the same core. The sections marked as global should be relocated to the PFlash of the master core.

  Note: Relocating code, data and constants to a distant memory space would impact execution timings.

  Note: If the driver operates from single (master) core, all the sections may be relocated to the PFlash/DSPR/ DLMU of the same CPU core.

1.1.4.3   MCU support

The PWM driver is dependent on the MCU driver for clock configuration, GTM timer to port pin connections and timer IP-related services. The initialization of the PWM driver must be started only after completing the MCU initialization. The following must be considered while configuring the MCU driver in tresos:

- The GTM/CCU6 hardware timers used by the PWM driver must be reserved in the MCU configuration for exclusive use by the PWM

- For signal output, the GTM timer to port pin connections should be configured in the MCU configuration for each TOM/ATOM channel allocated for PWM

1 Pwm_17_GtmCcu6 driver

Access of shared GTM SFRs

If channels of same the TOM/ATOM module are shared between the application and the PWM driver, the user shall ensure that shared register of TOM/ATOM modules are accessed using the MCU timer IP library APIs.

1.1.4.4　　　　Port support

The PORT driver configures the port pins of the entire microcontroller. The user must configure the port pins used by the PWM driver through the PORT configuration and initialize the port pins prior to invoking the PWM initialization.

1.1.4.5　　　　DMA support

The PWM driver does not use any services provided by the DMA driver.

1.1.4.6　　　　Interrupt connections

The interrupt connections of the PWM driver are described in this section.

If the user wants notifications, the user should enable interrupts in the Interrupt configuration register. The interrupt configuration registers of different hardware used by PWM channels are as follows.

Table 4　　　　GTM interrupts

GTM hardware used	SRC register
GTM-TOM	SRC_GTMTOMwx (w= TOM module; x= TOM channel)
GTM-ATOM	SRC_GTMATOMwx (w= ATOM module; x= ATOM channel)

Table 5　　　　CCU6 interrupts

CCU6 kernel used	CCU6 comparator used	SRC register
CCU60	CCU60	SRC_CCU60SR0
CCU60	CCU61	SRC_CCU60SR1
CCU60	CCU62	SRC_CCU60SR2
CCU60	CCU63	SRC_CCU60SR3
CCU61	CCU60	SRC_CCU61SR0
CCU61	CCU61	SRC_CCU61SR1
CCU61	CCU62	SRC_CCU61SR2
CCU61	CCU63	SRC_CCU61SR3

The MCU driver through its timer-related services provides interrupt handler for GTM and CCU6 timers.

All the ISRs to the GTM must be routed to the `Mcu_17_Gtm_TomChannelIsr` or `Mcu_17_Gtm_AtomChannelIsr` API, which further invokes `Pwm_17_GtmCcu6_Timer_Isr`.

All the ISRs to the CCU6 must be routed to the `Mcu_17_Ccu6_ChannelIsr` API, which further invokes `Pwm_17_GtmCcu6_Timer_Isr`. An example ISR handling is shown as follows.

1 Pwm_17_GtmCcu6 driver

```
/* include MCU timer header file */
#include "Mcu_17_TimerIp.h"
/*********SRC_ GTMTOM0SR0*********/
ISR(GTMTOM0SR0_ISR)
{
 /* Enable Global Interrupts */
 ENABLE();

 /* Parameter is Channel Number */
 Mcu_17_Gtm_TomChannelIsr (TOM_MODULE_0 , TOM_CHANNEL_0);
 /* TOM_MODULE_0 = 0, TOM_CHANNEL_0 = 0 */
}


/*********SRC_ CCU60SR0*********/
ISR(CCU60SR0_ISR){
 /* Enable Global Interrupts */
 ENABLE();

 /* Parameter is Channel Number */
 Mcu_17_Ccu6_ChannelIsr(CCU6_KERNEL_0,CCU6_CHANNEL_0);
}
```

1.1.4.7 Example usage

Driver initialization

The following code listing shows the steps involved in the initialization of the PWM driver.

```
#include "Mcu.h"
#include "Irq.h"
#include "Port.h"
#include "Pwm_17_GtmCcu6.h"

Mcu_Init(&Mcu_Config);
Mcu_InitClock(0);
while(Mcu_GetPllStatus() != MCU_PLL_LOCKED);
Mcu_DistributePllClock();
/* Configure Interrupt priority */
IrqGtm_Init();
/* Port Initialization */
Port_Init(&Port_ConfigRoot);
/* Pwm Initialization */
Pwm_17_GtmCcu6_Init(&Pwm_17_GtmCcu6_Config);
```

After invoking the `Pwm_17_GtmCcu6_Init` API, the PWM output signals are generated for the configured duty and period.

Note: User must ensure that the `Pwm_17_GtmCcu6_Init` API is called before using any other runtime APIs provided by the PWM driver.

Driver de-initialization

The following code listing depicts the steps involved in the de-initialization of the PWM driver.

Note: De-initializing the PWM driver will put all the channels to the idle state.

```
/* De-initialize PWM driver */
Pwm_17_GtmCcu6_DeInit();
/* Pwm Channel output is set to Idle */
```

Set duty cycle

The following code snippet shows invoking of the `Pwm_17_GtmCcu6_SetDutyCycle` API.

```
/* Change Duty to 25% when PwmDutyShiftInTicks is OFF*/
Pwm_17_GtmCcu6_SetDutyCycle(0,0x2000);
/* Change Duty to 25% when PwmDutyShiftInTicks is ON*/
/* For channel 0, the period is 48, 25% would result in 12 ticks */
Pwm_17_GtmCcu6_SetDutyCycle(0,12);
```

Set period and duty cycle

1 Pwm_17_GtmCcu6 driver

The following code snippet shows invoking of `Pwm_17_GtmCcu6_SetDutyCycle` API.

```
/* Change Duty to 25% and period 60 ticks for channel 4 when PwmDutyShiftInTicks is OFF */
Pwm_17_GtmCcu6_SetPeriodAndDuty(4,60,0x2000);
/* Change Duty to 25% and period 60 ticks for channel 4 when PwmDutyShiftInTicks is ON */
/* For Period 60, 25% would result in 15 ticks */
Pwm_17_GtmCcu6_SetPeriodAndDuty(4,60,15);
```

Notification, set idle and get output state

The following code listing shows invoking of the notification APIs, `Pwm_17_GtmCcu6_SetOutputToIdle` and `Pwm_17_GtmCcu6_GetOutputState` API.

```
/* Pwm Initialization */
Pwm_17_GtmCcu6_Init(&Pwm_Config);
/* Enable Notification for channel 3 */
Pwm_17_GtmCcu6_EnableNotification(3,PWM_RISING_EDGE);
/* Disable Notification for channel 3 */
Pwm_17_GtmCcu6_DisableNotification(3);
/* Set Output of channel 3 to Idle */
Pwm_17_GtmCcu6_SetOutputToIdle(3);
/* Get Output State of channel 3 */
/* Here variable 'State' holds the output state of channel 3 */
State = Pwm_17_GtmCcu6_GetOutputState(3);
```

1.1.5   Key architectural considerations

1.1.5.1   User mode support

The PWM driver operates in both User-1 and Supervisor modes without the need of any configuration parameter to configure the behavior.

[cover parentID PWM={1A65EADD-AFD0-4845-B2D2-8257E086DD67}]

[cover parentID PWM={ED41EDC1-CB6C-4821-BED8-1735365FE93D}]

1.2 Assumptions of Use (AoU)

The AoU for the PWM driver are as follows.

- **Channel allocation to cores**

Integrator shall ensure the following points while allocating channels to cores:

- If the PwmHandleShiftByOffset configuration parameter is set to ON, all GTM channels of TGC/AGC should be allocated to the same core. This will enable synchronous start of all channels in same TGC/AGC

- For PWM channels of type GTM, Channel x and Channel x+1 of same TGC/AGC shall be allocated to same core, as they share same interrupt node

- To enable synchronous start of CCU6 channels, all CCU6 channels of the kernel shall be allocated to same core

[cover parentID PWM={A1C61D00-C7D6-49cc-B0AE-CEF01CC900D4}]

- **Config pointer to initialization**

When the PWM driver is used in the multicore environment, user has to ensure that the same configuration pointer is passed to the initialization function from different cores.

[cover parentID PWM={170B536A-4745-4890-87FD-6155B7B45F0E}]

- **Notification when duty is 0% or 100%**

Decision: When notification for duty 0% or 100% is enabled by the user for fixed and variable period channels then the following notification combination shall be used.

- Duty is 0%:

- Polarity - high, notify value - rising edge

- Polarity - low, notify value - falling edge

- Duty is 100%:

- Polarity - high, notify value - falling edge

- Polarity - low, notify value - rising edge

The notify value is the input parameter for the Pwm_17_GtmCcu6_EnableNotification function.

Rational: If both edges are enabled for 0% or 100 % duty, the differentiation of edges is not possible.

[cover parentID PWM={B1EE9B66-F309-4a94-BBCC-8487D15184C3}]

- **ShiftOffset = ON for shifted channels**

If the shifted PWM channels are used then it is recommended to set the PwmHandleShiftByOffset configuration parameter to ON. If PwmHandleShiftByOffset is set to OFF, the shifted period channels start late compared to the reference channel due to which the first period of shifted period channels may not match the period of the reference channel.

[cover parentID PWM={05182FFE-74CA-4bf4-AE87-BD22DE99E628}]

1.3 Reference information

1.3.1 Configuration interfaces
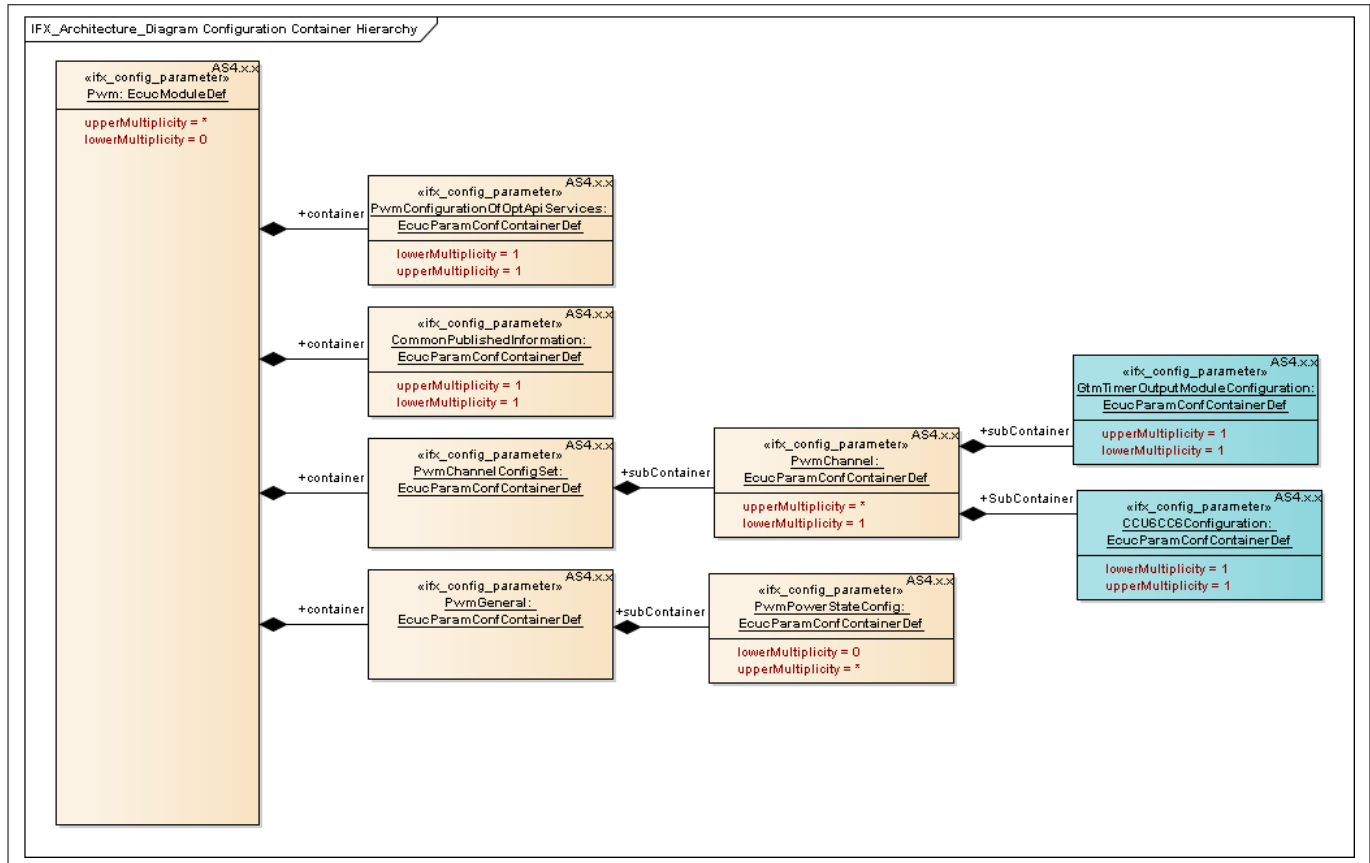
Figure 4   Container hierarchy along with their configuration parameters

1.3.1.1 Container: CCU6CC6Configuration

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.1.1 CCU6KernelUsed

Table 6   Specification for CCU6KernelUsed

Name	CCU6KernelUsed		
Description	Lists all the CCU6 kernels *Note: CCU6 resource allocation is done at the kernel level for each module.* *Once the kernel is allocated to the module, the T12 or T13 timer of the kernel can only be assigned to one of the channels within the module.* *Note: Default value is set to blank as user has to select the appropriate reference value from the MCU driver.*		
Multiplicity	1..1	Type	EcucReferenceDef

Table 6   **Specification for CCU6KernelUsed (continued)**

Range	Reference to Node: McuCcu6ModuleAllocationConf		
Default value	NULL		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.2   CCU6TimerClockSelect

Table 7   **Specification for CCU6TimerClockSelect**

Name	CCU6TimerClockSelect		
Description	Selects the clock source for T12 or T13 timer. *Note: The default value is CCU6_CONFIGURABLE_CLOCK_0 as it is the lowest configurable divider value.*		
Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	CCU6_CONFIGURABLE_CLOCK_0: Selected clock is fcc6 or fcc6/256 (pre scalar enabled). *Note: The fcc6 is the module clock for the CCU6 kernel. For more information, refer to the Hardware Target Specification.* CCU6_CONFIGURABLE_CLOCK_1: Selected clock is fcc6/2 or fcc6/512 (prescalar enabled) CCU6_CONFIGURABLE_CLOCK_2: Selected clock is fcc6/4 or fcc6/1024 (prescalar enabled) CCU6_CONFIGURABLE_CLOCK_3: Selected clock is fcc6/8 or fcc6/2048 (prescalar enabled) CCU6_CONFIGURABLE_CLOCK_4: Selected clock is fcc6/16 or fcc6/4096 (prescalar enabled) CCU6_CONFIGURABLE_CLOCK_5: Selected clock is fcc6/64 or fcc6/16348 (prescalar enabled) CCU6_CONFIGURABLE_CLOCK_6: Selected clock is fcc6/32 or fcc6/8192 (prescalar enabled) CCU6_CONFIGURABLE_CLOCK_7: Selected clock is fcc6/128 or fcc6/32768 (prescalar enabled)		
Default value	CCU6_CONFIGURABLE_CLOCK_0		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	-		

Table 7   **Specification for CCU6TimerClockSelect (continued)**

Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.1.3   CCU6TimerPrescalarEnabled

Table 8   **Specification for CCU6TimerPrescalarEnabled**

Name	CCU6TimerPrescalarEnabled		
Description	If CCU6TimerPrescalarEnabled is FALSE then T12 or T13 clock = fcc6 divided by (2 power CCU6TimerClockSelect)		
	If CCU6TimerPrescalarEnabled is TRUE then T12 or T13 clock = fcc6 divided by (2 power (CCU6TimerClockSelect+8))		
	Note: The default value is set to FALSE as the hardware default value is FALSE.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE		
	FALSE		
Default value	FALSE		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.4   CCU6TimerUsed

Table 9   **Specification for CCU6TimerUsed**

Name	CCU6TimerUsed		
Description	Selects the T12 or T13 timer.		
	Note: The default value is T12 as it is the lowest configurable timer for the kernel.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	T12: T12: selected T12 timer in the Ccu6x kernel		
	T13: T13: selected T13 timer in the CCU6x kernel		
Default value	T12		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-

Table 9   **Specification for CCU6TimerUsed (continued)**

Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	CCU6KernelUsed		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.1.5   Cc6xChannel

Table 10   **Specification for Cc6xChannel**

Name	Cc6xChannel		
Description	Selects a CC6x channel. *Note: The default value is chosen as Cc60 as it is the CCU6 lowest comparator.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	Cc60: CC60 comparator selected Cc61: CC61 comparator selected Cc62: CC62 comparator selected		
Default value	Cc60		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2   Container: CommonPublishedInformation

Contains published information about vendor and versions

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.2.1   ArMajorVersion

Table 11   **Specification for ArMajorVersion**

Name	ArMajorVersion
Description	Provides the major version of the AUTOSAR specification

Table 11   **Specification for ArMajorVersion (continued)**

Multiplicity	1..1	Type	EcucIntegerParamDef
Range	0 - 255		
Default value	4		
Post-build variant value	FALSE	Post-build variant multiplicity	-
Value configuration class	Published-Information	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.2   ArMinorVersion

Table 12   **Specification for ArMinorVersion**

Name	`ArMinorVersion`		
Description	Provides the minor version of the AUTOSAR specification		
Multiplicity	1..1	Type	EcucIntegerParamDef
Range	0 - 255		
Default value	As per Autosar Version		
Post-build variant value	FALSE	Post-build variant multiplicity	-
Value configuration class	Published-Information	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.3   ArPatchVersion

Table 13   **Specification for ArPatchVersion**

Name	`ArPatchVersion`		
Description	Provides the patch version of the AUTOSAR specification		
Multiplicity	1..1	Type	EcucIntegerParamDef
Range	0 - 255		
Default value	As per Autosar Version		

Table 13	Specification for ArPatchVersion (continued)		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.4   ModuleId

Table 14	Specification for ModuleId		
Name	`ModuleId`		
Description	Provides the module Id		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 65535		
Default value	121		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.5   Release

Table 15	Specification for Release		
Name	`Release`		
Description	Indicates the TC3xx device derivative used for the implementation		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	As per the hardware derivate		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

Table 15	**Specification for Release (continued)**		
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.6   SwMajorVersion

Table 16	**Specification for SwMajorVersion**		
Name	SwMajorVersion		
Description	Module Major version		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	1 - 255		
Default value	As per Driver Version		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.7   SwMinorVersion

Table 17	**Specification for SwMinorVersion**		
Name	SwMinorVersion		
Description	Provides the minor version of the software		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per the driver Version		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL

Table 17	Specification for SwMinorVersion (continued)
Dependency	-
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.2.8   SwPatchVersion

Table 18	Specification for SwPatchVersion		
Name	SwPatchVersion		
Description	Provides the patch version of the software		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 255		
Default value	As per the driver Version		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.9   VendorApiInfix

Table 19	Specification for VendorApiInfix		
Name	VendorApiInfix		
Description	Provides the vendor-specific name		
Multiplicity	1..1	**Type**	EcucStringParamDef
Range	String		
Default value	GtmCcu6		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.2.10 VendorId

Table 20 **Specification for VendorId**

Name	VendorId		
Description	Provides the Infineon vendor ID in the HIS software specification		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 65535		
Default value	17		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Published-Information	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.3 Container: GtmTimerOutputModuleConfiguration

This container contains the elements for configuring the GTM timer hardware (TOM or ATOM). The settings in this container are used to configure the timing needs of the TOM or ATOM timer.

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.3.1 GtmTimerClockSelect

Table 21 **Specification for GtmTimerClockSelect**

Name	GtmTimerClockSelect		
Description	Selects the clock source for the TOM or ATOM timer. The default value is GTM_FIXED_CLOCK_0 as it is the lowest configurable divider value.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	GTM_CONFIGURABLE_CLOCK_0: Configurable Clock 0 is selected for the ATOM module. GTM_CONFIGURABLE_CLOCK_1: Configurable Clock 1 is selected for the ATOM module. GTM_CONFIGURABLE_CLOCK_2: Configurable Clock 2 is selected for the ATOM module. GTM_CONFIGURABLE_CLOCK_3: Configurable Clock 3 is selected for the ATOM module. GTM_CONFIGURABLE_CLOCK_4: Configurable Clock 4 is selected for the ATOM module. GTM_CONFIGURABLE_CLOCK_5: Configurable Clock 5 is selected for the ATOM module. GTM_CONFIGURABLE_CLOCK_6: Configurable Clock 6 is selected for the ATOM module. GTM_CONFIGURABLE_CLOCK_7: Configurable Clock 7 is selected for the ATOM module. GTM_FIXED_CLOCK_0: Fixed Clock 0 is selected for the TOM module. GTM_FIXED_CLOCK_1: Fixed Clock 1 is selected for the TOM module.		

1 Pwm_17_GtmCcu6 driver

Table 21 **Specification for GtmTimerClockSelect (continued)**

	GTM_FIXED_CLOCK_2: Fixed Clock 2 is selected for the TOM module. GTM_FIXED_CLOCK_3: Fixed Clock 3 is selected for the TOM module. GTM_FIXED_CLOCK_4: Fixed Clock 4 is selected for the TOM module.		
Default value	GTM_FIXED_CLOCK_0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	PwmAssignedHwUnit		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.3.2 GtmTimerUsed

Table 22 **Specification for GtmTimerUsed**

Name	GtmTimerUsed		
Description	The TOM or ATOM channel resource assigned to the PWM channel. This parameter lists all the GTM timer channels (TOM or ATOM). The referred timer channel in the MCU should have TomChannelUsage/AtomChannelUsage as USED_BY_PWM_DRIVER. *Note: The default value is set to BLANK as user has to select the appropriate reference value from the MCU driver.*		
Multiplicity	1..1	**Type**	EcucChoiceReference Def
Range	Reference to Node: McuGtmAtomChannelAllocationConf, McuGtmTomChannelAllocationConf		
Default value	NULL		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	PwmAssignedHwUnit		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.4 Container: Pwm

This container contains the configurations of the PWM driver

1 Pwm_17_GtmCcu6 driver

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.4.1   Config Variant

Table 23   Specification for Config Variant

Name	Config Variant		
Description	Selects the config variant for the PWM driver. The default value of this parameter is set to VariantPostBuild as per AUTOSAR.		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	VariantPostBuild: Post Build Support		
Default value	VariantPostBuild		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5   Container: PwmChannel

This container contains the configuration of an individual PWM channel.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: -

1.3.1.5.1   PwmAssignedHwUnit

Table 24   Specification for PwmAssignedHwUnit

Name	PwmAssignedHwUnit		
Description	Hardware resource used for PWM generation: GTM or CCU6. *Note: This parameter will hold only CCU6 option for GTM-less devices. In this case default value will be CCU6.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	CCU6: CCU6 timer selected. GTM: GTM timer selected.		
Default value	GTM		

Table 24	Specification for PwmAssignedHwUnit (continued)		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.2   PwmChannelClass

Table 25	Specification for PwmChannelClass		
Name	PwmChannelClass		
Description	Class of PWM channel		
Multiplicity	0..1	**Type**	EcucEnumerationParamDef
Range	PWM_FIXED_PERIOD: The PWM channel has a fixed period. Only duty cycle can be changed. PWM_FIXED_PERIOD_CENTER_ALIGNED: This class of PWM channel refers to a PWM_FIXED_PERIOD channel for the period and the duty cycle will fall exactly on the middle of the period. Only duty cycle can be changed for this channel class. *Note: PWM_FIXED_PERIOD_CENTER_ALIGNED channel is only applicable for GTM channels.* PWM_FIXED_PERIOD_SHIFTED: This class of PWM channel refers to a PWM_FIXED_PERIOD channel for the period and the duty cycle is shifted by a required fixed percentage. Only duty cycle can be changed. PWM_VARIABLE_PERIOD: This class of PWM channel has a variable period and duty cycle. Both can be changed during run time. This channel should not be referred by any other channel and does not refer to any channel either. *Note: PWM_VARIABLE_PERIOD is not applicable if PwmSetPeriodAndDuty is set to false*		
Default value	PWM_FIXED_PERIOD		
Post-build variant value	TRUE	**Post-build variant multiplicity**	TRUE
Value configuration class	Post-Build	**Multiplicity configuration class**	Post-Build
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	PwmSetPeriodAndDuty		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.3 PwmChannelEcucPartitionRef

Table 26 Specification for PwmChannelEcucPartitionRef

Name	PwmChannelEcucPartitionRef		
Description	This parameter maps a PWM channel to zero or multiple ECUC partitions to limit the access to this channel. The ECUC partitions referenced are a subset of the ECUC partitions where the PWM driver is mapped. *Note: Parameter support is added only for AUTOSAR schema compliance, this parameter is not used in code generation logic, hence this parameter is made editable false.*		
Multiplicity	0..*	**Type**	EcucReferenceDef
Range	Reference to Node: EcucPartition		
Default value	NULL		
Post-build variant value	TRUE	**Post-build variant multiplicity**	TRUE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	None
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.5.4 PwmChannelId

Table 27 Specification for PwmChannelId

Name	PwmChannelId		
Description	Logical channel Id of the PWM channel. This value is assigned to the symbolic name derived from the PwmChannel container short name. The Id value must be consecutive. *Note: A consecutive value is calculated for each new PWM channel. The minimum channel ID is selected as the default value.* *Note: The value of PwmChannelId should be unique in a configuration set.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - Total number of channels configured - 1		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		

Table 27	Specification for PwmChannelId (continued)
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.5.5 PwmCoherentUpdate

Table 28 **Specification for PwmCoherentUpdate**

Name	PwmCoherentUpdate		
Description	Switch for enabling the update of the duty/period parameter at the end of the current period.		
	TRUE: update of period/duty cycle is done at the end of period of the currently generated waveform (current waveform is finished).		
	FALSE: update of period/duty cycle is done immediately (just after the service call, the current waveform is cut).		
	Depending on the PwmChannelClass the update end period will either work for duty cycle or duty and period. Both duty and period are updated for the variable period channel and only duty cycle is updated for other PwmChannelClass channels.		
	Note: This parameter is only applicable for GTM channels.		
Multiplicity	1..1	Type	EcucBooleanParamDef
Range	TRUE		
	FALSE		
Default value	FALSE		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	PwmAssignedHwUnit, PwmChannelClass, PwmChannelCoherentSelection		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.6 PwmDutycycleDefault

Table 29 **Specification for PwmDutycycleDefault**

Name	PwmDutycycleDefault
Description	Default value of the duty cycle for a channel. It can be entered as absolute ticks or as relative percentage based on the PwmDutyShiftInTicks parameter.
	When the PwmDutyShiftInTicks is set to OFF, the value is relative to period.
	0 is 0%,
	0x8000 is 100%

Table 29	Specification for PwmDutycycleDefault (continued)		
	When the PwmDutyShiftInTicks is set to ON, the value is in absolute ticks. 0 is 0%, If the value same is greater than or equal to PwmPeriodDefault then duty cycle is 100% Range: 0 to 0x8000, if PwmDutyShiftInTicks is STD_OFF for TOM/ATOM/CCU6 0 to 0xFFFF, if PwmDutyShiftInTicks is STD_ON and the module is TOM,CCU6 0 to 0xFFFFFF, if PwmDutyShiftInTicks is STD_ON and the module is ATOM *Note: The default is set to FALSE as the hardware reset is FALSE.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 16777215		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	PwmDutyShiftInTicks		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.7 PwmIdleState

Table 30	Specification for PwmIdleState		
Name	PwmIdleState		
Description	Represents the output state of the PWM after the signal is stopped (that is call to Pwm_17_GtmCcu6_SetOutputToIdle). *Note: Default is hardware reset value.*		
Multiplicity	1..1	**Type**	EcucEnumerationParamDef
Range	PWM_HIGH: The PWM channel output will be set to high (3 or 5 V) in idle state. PWM_LOW: The PWM channel output will be set to low (0 V) in idle state.		
Default value	PWM_LOW		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL

Table 30   Specification for PwmIdleState (continued)

Dependency	-
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.5.8   PwmMcuClockReferencePoint

Table 31   Specification for PwmMcuClockReferencePoint

Name	PwmMcuClockReferencePoint		
Description	Contains reference to the McuClockReferencePoint. Since this parameter is not used, it is, hence disabled.		
Multiplicity	1..1	**Type**	EcucReferenceDef
Range	Reference to Node: McuClockReferencePointConfig		
Default value	NULL		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	ECU
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.9   PwmNotification

Table 32   Specification for PwmNotification

Name	PwmNotification		
Description	The PwmNotification is used by the PWM driver to invoke the user-defined function for edge generation of the respective channel. The parameter can be a name or the address (numeric value) of the notification function. *Note: Since the name of the function is configurable, the default value is kept as NULL.* *Note: The PWM driver does not validate the configured function name or address for correctness and the responsibility falls on the user.* If McuTomChannelEventHandledByDsadc/McuAtomChannelEventHandledByDsadc is TRUE for the channel, notification callback will be handled by DSADC and not by PWM, when an event occurs for the TOM/ATOM channel respectively. Hence, PwmNotification parameter will not be editable for that channel.		
Multiplicity	0..1	**Type**	EcucFunctionNameDef
Range	String		
Default value	NULL_PTR		

Table 32	Specification for PwmNotification (continued)		
Post-build variant value	TRUE	**Post-build variant multiplicity**	TRUE
Value configuration class	Post-Build	**Multiplicity configuration class**	Post-Build
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	McuAtomChannelEventHandledByDsadc, McuTomChannelEventHandledByDsadc, PwmNotificationSupported		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.10 PwmPeriodDefault

Table 33	Specification for PwmPeriodDefault		
Name	PwmPeriodDefault		
Description	Value of period used for initialization. Significant if PwmChannelClass is PWM_FIXED_PERIOD or PWM_VARIABLE_PERIOD Range: 0 to 0xFFFF, when module is TOM or CCU6 0 to 0xFFFFFF, when module is ATOM *Note: The default value is set to match the hardware reset value*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 16777215		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	PwmChannelClass		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.11 PwmPolarity

Table 34	Specification for PwmPolarity
Name	PwmPolarity
Description	Defines the starting polarity of each PWM channel. *Note: The default is set to PWM_LOW as the hardware reset value is FALSE.*

Table 34   **Specification for PwmPolarity (continued)**

Multiplicity	1..1	Type	EcucEnumerationParamDef
Range	PWM_HIGH: PWM channel output is high at the beginning of the cycle and then goes low when the duty count is reached PWM_LOW: PWM channel output is low at the beginning of the cycle and then goes high when the duty count is reached		
Default value	PWM_LOW		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	AUTOSAR_ECUC	Scope	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.12   PwmReferenceChannel

Table 35   **Specification for PwmReferenceChannel**

Name	`PwmReferenceChannel`		
Description	Significant if PwmChannelClass is PWM_FIXED_PERIOD_SHIFTED or PWM_FIXED_PERIOD_CENTER_ALIGNED Reference channel must be provided if PwmChannelClass is PWM_FIXED_PERIOD_SHIFTED or PWM_FIXED_PERIOD_CENTER_ALIGNED. Reference channel can only be of PWM_FIXED_PERIOD. *Note: The default value is blank as user has to select the appropriate reference value from the MCU driver.*		
Multiplicity	1..1	Type	EcucReferenceDef
Range	Reference to Node: PwmChannel		
Default value	NULL		
Post-build variant value	TRUE	Post-build variant multiplicity	-
Value configuration class	Post-Build	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	PwmChannelClass		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.5.13 PwmShiftValue

Table 36 **Specification for PwmShiftValue**

Name	PwmShiftValue		
Description	Configures the initial shift value with respect to the PWM_FIXED_PERIOD class channel. It can be entered as absolute ticks or as relative percentage based on the PwmDutyShiftInTicks parameter. When PwmDutyShiftInTicks is set to OFF, the value is relative to period. 0 is 0%, 0x8000 is 100% When PwmDutyShiftInTicks is set to ON, the value is in absolute ticks. Value 0 is 0%, Value same as PwmPeriodDefault of reference channel is 100% Significant if PwmChannelClass is PWM_FIXED_PERIOD_SHIFTED *Note: The default is set to match the hardware reset value.*		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 16777215		
Default value	0		
Post-build variant value	TRUE	**Post-build variant multiplicity**	-
Value configuration class	Post-Build	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	PwmChannelClass, PwmDutyShiftInTicks		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.6 Container: PwmChannelConfigSet

This container contains the configuration parameters and sub containers of the AUTOSAR PWM driver

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.7 Container: PwmConfigurationOfOptApiServices

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.7.1 PwmDeInitApi

Table 37 **Specification for PwmDeInitApi**

Name	PwmDeInitApi

Table 37   **Specification for PwmDeInitApi (continued)**

Description	Adds/removes the Pwm_17_GtmCcu6_DeInit() API from the code.		
	TRUE Pwm_17_GtmCcu6_DeInit() API is available to the user.		
	FALSE Pwm_17_GtmCcu6_DeInit() API is not available to the user.		
	Note: The Pwm_17_GtmCcu6_DeInit() API is disabled by default to minimize the executable code size.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE		
	FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.2   PwmGetOutputState

Table 38   **Specification for PwmGetOutputState**

Name	PwmGetOutputState		
Description	Adds/removes the Pwm_17_GtmCcu6_GetOutputState() API from the code.		
	TRUE Pwm_17_GtmCcu6_GetOutputState() API is available to the user.		
	FALSE Pwm_17_GtmCcu6_GetOutputState() API is not available to the user.		
	Note: The Pwm_17_GtmCcu6_GetOutputState() API is disabled by default to minimize the executable code size.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE		
	FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL

Table 38	Specification for PwmGetOutputState (continued)
Dependency	-
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.7.3   PwmSetDutyCycle

Table 39   Specification for PwmSetDutyCycle

Name	PwmSetDutyCycle		
Description	Adds/removes the service Pwm_17_GtmCcu6_SetDutyCycle() API from the code. TRUE: Pwm_17_GtmCcu6_SetDutyCycle() API is available to the user FALSE: Pwm_17_GtmCcu6_SetDutyCycle() API is not available to the user *Note: Pwm_17_GtmCcu6_SetDutyCycle() API is disabled by default to minimize the executable code size.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.4   PwmSetOutputToIdle

Table 40   Specification for PwmSetOutputToIdle

Name	PwmSetOutputToIdle		
Description	Adds/removes the service Pwm_17_GtmCcu6_SetOutputToIdle() from the code. TRUE: Pwm_17_GtmCcu6_SetOutputToIdle() API is available to the user FALSE: Pwm_17_GtmCcu6_SetOutputToIdle() API is not available to the user *Note: Pwm_17_GtmCcu6_SetOutputToIdle() API is disabled by default to minimize the executable code size.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE		

Table 40   **Specification for PwmSetOutputToIdle (continued)**

	FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.5   PwmSetPeriodAndDuty

Table 41   **Specification for PwmSetPeriodAndDuty**

Name	PwmSetPeriodAndDuty		
Description	Adds/removes the service Pwm_17_GtmCcu6_SetPeriodAndDuty() from the code.		
	TRUE: Pwm_17_GtmCcu6_SetPeriodAndDuty() API is available to the user		
	FALSE: Pwm_17_GtmCcu6_SetPeriodAndDuty() API is not available to the user		
	Note: Pwm_17_GtmCcu6_SetPeriodAndDuty() API is disabled by default to minimize the executable code size.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE		
	FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.7.6   PwmVersionInfoApi

Table 42   **Specification for PwmVersionInfoApi**

Name	PwmVersionInfoApi

Table 42   **Specification for PwmVersionInfoApi (continued)**

Description	Adds/removes the service Pwm_17_GtmCcu6_GetVersionInfo from the code.		
	TRUE: Pwm_17_GtmCcu6_GetVersionInfo() API is available to the user.		
	FALSE: Pwm_17_GtmCcu6_GetVersionInfo() API is not available to the user.		
	Note: The Pwm_17_GtmCcu6_GetVersionInfo() API is disabled by default to minimize the executable code size.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE		
	FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8   Container: PwmGeneral

Post-Build Variant Multiplicity: -

Multiplicity Configuration Class: -

1.3.1.8.1   PwmChannelCoherentSelection

Table 43   **Specification for PwmChannelCoherentSelection**

Name	`PwmChannelCoherentSelection`		
Description	Switch for enabling the channel-wise update of the duty/period parameter at the end of the current period.		
	TRUE: Update of period/duty cycle is done based on the PWM channel-specific configuration PwmCoherentUpdate configuration parameter.		
	FALSE: Update of period/duty cycle is done based on the PwmDutycycleUpdatedEndPeriod and PwmPeriodUpdatedEndPeriod global parameters.		
	Note: This parameter is only applicable for GTM channels.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE		
	FALSE		
Default value	FALSE		

Table 43	Specification for PwmChannelCoherentSelection (continued)		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	PwmAssignedHwUnit		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.2   PwmDevErrorDetect

Table 44	Specification for PwmDevErrorDetect		
Name	PwmDevErrorDetect		
Description	Switches the DET detection and notification to ON or OFF. TRUE: enabled FALSE: disabled *Note: The default value is set to FALSE to minimize the executable code size.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.3   PwmDutyShiftInTicks

Table 45	Specification for PwmDutyShiftInTicks	
Name	PwmDutyShiftInTicks	
Description	Determines whether duty cycle and shift values are absolute or relative to period. TRUE: duty cycle and shift value to be entered as absolute ticks FALSE: duty cycle and shift values are entered relative to period	

Table 45 **Specification for PwmDutyShiftInTicks (continued)**

	0 is 0%, 0x8000 is 100%. *Note: The default value is set to FALSE as it is a non-AUTOSAR feature*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.4   PwmDutycycleUpdatedEndperiod

Table 46 **Specification for PwmDutycycleUpdatedEndperiod**

Name	PwmDutycycleUpdatedEndperiod		
Description	Global switch for enabling the update of the duty cycle parameter at the end of the current period. TRUE: update of duty cycle is done at the end of period of the currently generated waveform (current waveform is finished). FALSE: update of duty cycle is done immediately (just after the service call, the current waveform is updated with the new duty cycle value). Default value is set to Hardware default value. *Note: If the global switch is TRUE then for PWM_VARIABLE_PERIOD channel both duty and period will get updated at the end of period of the currently generated waveform.* *Note: This parameter is only applicable for GTM channels.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

Table 46	Specification for PwmDutycycleUpdatedEndperiod (continued)		
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	PwmAssignedHwUnit, PwmChannelCoherentSelection		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.5 PwmEcucPartitionRef

Table 47	Specification for PwmEcucPartitionRef		
Name	PwmEcucPartitionRef		
Description	This parameter maps the PWM driver to zero or multiple ECUC partitions to make the driver API available in the according partition. *Note: Parameter support is added only for AUTOSAR schema compliance, this parameter is not used in code generation logic, hence this parameter is made editable false.*		
Multiplicity	0..*	**Type**	EcucReferenceDef
Range	Reference to Node: EcucPartition		
Default value	NULL		
Post-build variant value	TRUE	**Post-build variant multiplicity**	TRUE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	None
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.8.6 PwmEnable0Or100DutyNotification

Table 48	Specification for PwmEnable0Or100DutyNotification		
Name	PwmEnable0Or100DutyNotification		
Description	If set to ON, notifications are supported for 0% and 100% for the fixed and variable period channels. If set to OFF, notifications are not supported for 0% and 100% for the fixed and variable period channels. *Note: The default value is set to FALSE to minimize the executable code size.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE		

1 Pwm_17_GtmCcu6 driver

Table 48   Specification for PwmEnable0Or100DutyNotification (continued)

	FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	PwmNotificationSupported		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.7   PwmHandleShiftByOffset

Table 49   Specification for PwmHandleShiftByOffset

Name	PwmHandleShiftByOffset		
Description	When PwmHandleShiftByOffset is enabled then the shifted channels start with an offset and will work similar to the fixed period channel. When PwmHandleShiftByOffset is not selected the shifted channels are triggered from the referenced fixed period channel. *Note: This parameter is only applicable for GTM channels.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	PwmAssignedHwUnit		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.8   PwmIndex

Table 50   Specification for PwmIndex

Name	PwmIndex

Table 50   **Specification for PwmIndex (continued)**

Description	Instance ID of the PWM module instance		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 4294967295		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.9   PwmInitCheckApi

Table 51   **Specification for PwmInitCheckApi**

Name	PwmInitCheckApi		
Description	The parameter adds or removes the Pwm_17_GtmCcu6_InitCheck() API from the code.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	TRUE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.10   PwmKernelEcucPartitionRef

Table 52   **Specification for PwmKernelEcucPartitionRef**

Name	PwmKernelEcucPartitionRef

Table 52 **Specification for PwmKernelEcucPartitionRef (continued)**

Description	This parameter maps the PWM kernel to zero or one ECUC partitions to assign the driver kernel to a certain core. The ECUC partition referenced is a subset of the ECUC partitions where the PWM driver is mapped to.		
	Note: Parameter support is added only for AUTOSAR schema compliance, this parameter is not used in code generation logic, hence this parameter is made editable false.		
Multiplicity	0..1	**Type**	EcucReferenceDef
Range	Reference to Node: EcucPartition		
Default value	NULL		
Post-build variant value	TRUE	**Post-build variant multiplicity**	TRUE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	None
Dependency	-		
Autosar Version	Applicable for Autosar version 4.4.0.		

1.3.1.8.11   PwmLowPowerStatesSupport

Table 53 **Specification for PwmLowPowerStatesSupport**

Name	PwmLowPowerStatesSupport		
Description	This Parameter is disabled as power modes are not supported.		
	Adds/removes all power state management-related APIs (PWM_SetPowerState, PWM_GetCurrentPowerState, PWM_GetTargetPowerState, PWM_PreparePowerState, PWM_Main_PowerTransitionManager) indicating that the hardware offers low power state management.		
Multiplicity	0..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.12   PwmMultiCoreErrorDetect

Table 54   Specification for PwmMultiCoreErrorDetect

Name	PwmMultiCoreErrorDetect		
Description	Enables or disables the multicore-related DET detection and reporting. It is applicable only when DET is enabled and also an error message will be raised if this parameter is set to TRUE when CPU is a single-core. *Note: The default value is set to FALSE to minimize the executable code size.*		
Multiplicity	1..1	Type	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	Post-build variant multiplicity	-
Value configuration class	Pre-Compile	Multiplicity configuration class	-
Origin	IFX	Scope	LOCAL
Dependency	PwmDevErrorDetect		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.13   PwmNotificationSupported

Table 55   Specification for PwmNotificationSupported

Name	PwmNotificationSupported		
Description	Switch to indicate that the notifications are supported. TRUE: notifications are supported FALSE: notifications are not supported *Note: The default value is set to FALSE to minimize the executable code size.*		
Multiplicity	1..1	Type	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	Post-build variant multiplicity	-
Value configuration class	Pre-Compile	Multiplicity configuration class	-
Origin	AUTOSAR_ECUC	Scope	LOCAL

Table 55 **Specification for PwmNotificationSupported (continued)**

Dependency	-
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.1.8.14 PwmPeriodUpdatedEndperiod

Table 56 **Specification for PwmPeriodUpdatedEndperiod**

Name	PwmPeriodUpdatedEndperiod		
Description	Global switch for enabling the update of the period parameter at the end of the current period.		
	TRUE: update of period and duty cycle for variable period channel is done at the end of period of currently generated waveform (current waveform is finished)		
	FALSE: update of period and duty cycle for variable period channel is done immediately (just after the service call, the current waveform is updated with new period value)		
	The PWM_FIXED_PERIOD, PWM_FIXED_PERIOD_SHIFTED and PWM_FIXED_PERIOD_CENTER_ALIGNED class channels do not get affected due to this parameter setting.		
	Note: Default value is set to Hardware default value.		
	Note: This parameter is only applicable for GTM channels.		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	PwmAssignedHwUnit, PwmChannelCoherentSelection		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.15 PwmPowerStateAsynchTransitionMode

Table 57 **Specification for PwmPowerStateAsynchTransitionMode**

Name	PwmPowerStateAsynchTransitionMode
Description	This parameter is disabled as power modes are not supported.
	Enables/disables support of the PWM driver to the asynchronous power state transition.

Table 57	**Specification for PwmPowerStateAsynchTransitionMode (continued)**		
Multiplicity	0..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	FALSE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	FALSE
Value configuration class	Pre-Compile	**Multiplicity configuration class**	Pre-Compile
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.8.16   PwmSafetyEnable

Table 58	**Specification for PwmSafetyEnable**		
Name	PwmSafetyEnable		
Description	Switch to enable reporting of safety errors (range and plausibility checks). *Note: When this switch is enabled, AUTOSAR DETs are enabled by default. The detection of safety-related errors is enabled by default to ensure that safety issues are addressed during the product lifecycle.*		
Multiplicity	1..1	**Type**	EcucBooleanParamDef
Range	TRUE FALSE		
Default value	TRUE		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	IFX	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9   Container: PwmPowerStateConfig

This container is disabled as power modes are not supported.

1 Pwm_17_GtmCcu6 driver

Each instance of this parameter defines a power state and the callback to be invoked when this power state is reached.

Post-Build Variant Multiplicity: FALSE

Multiplicity Configuration Class: -

1.3.1.9.1 PwmPowerState

Table 59   Specification for PwmPowerState

Name	PwmPowerState		
Description	This Parameter is disabled as power modes are not supported. Each instance of this parameter describes a different power state supported by the PWM hardware. It should be defined by the hardware supplier and used by the PWM driver to reference specific hardware configurations which set the PWM hardware module in the referenced power state.		
Multiplicity	1..1	**Type**	EcucIntegerParamDef
Range	0 - 4294967295		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-
Value configuration class	Pre-Compile	**Multiplicity configuration class**	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.1.9.2 PwmPowerStateReadyCbkRef

Table 60   Specification for PwmPowerStateReadyCbkRef

Name	PwmPowerStateReadyCbkRef		
Description	This parameter is disabled as power modes are not supported. Each instance of this parameter contains a reference to a power mode callback defined in a CDD or IoHwAbs component.		
Multiplicity	1..1	**Type**	EcucFunctionNameDef
Range	String		
Default value	0		
Post-build variant value	FALSE	**Post-build variant multiplicity**	-

Table 60 **Specification for PwmPowerStateReadyCbkRef (continued)**

Value configuration class	Pre-Compile	Multiplicity configuration class	-
Origin	AUTOSAR_ECUC	**Scope**	LOCAL
Dependency	-		
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.		

1.3.2 Functions - Type definitions

This section lists the data types of the PWM driver.

1.3.2.1 Pwm_17_GtmCcu6_ChannelType

Table 61 **Specification for Pwm_17_GtmCcu6_ChannelType**

Syntax	`Pwm_17_GtmCcu6_ChannelType`	
Type	uint8	
File	`Pwm_17_GtmCcu6.h`	
Range	0-199	The range is for the number of TOM channels, ATOM channels and CCU6 channels for the device variant. The maximum number of channels will vary depending on the device variant. 200 is for the superset device variant.
Description	Specifies the identification (ID) for a channel.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.2 Pwm_17_GtmCcu6_NotifiPtrType

Table 62 **Specification for Pwm_17_GtmCcu6_NotifiPtrType**

Syntax	`Pwm_17_GtmCcu6_NotifiPtrType`
Type	Pointer to a function of type void Function_Name (void)
File	`Pwm_17_GtmCcu6.h`
Description	Channel notification function pointer
Source	AUTOSAR
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.2.3   Pwm_17_GtmCcu6_PeriodType

Table 63   Specification for Pwm_17_GtmCcu6_PeriodType

Syntax	`Pwm_17_GtmCcu6_PeriodType`	
Type	uint32	
File	`Pwm_17_GtmCcu6.h`	
Range	0-16777215	Range: 0 to 0xFFFF for TOM/CCU6 0 to 0xFFFFFF for ATOM
Description	Definition of the period of a PWM channel	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.4   Pwm_17_GtmCcu6_OutputStateType

Table 64   Specification for Pwm_17_GtmCcu6_OutputStateType

Syntax	`Pwm_17_GtmCcu6_OutputStateType`	
Type	uint8	
File	`Pwm_17_GtmCcu6.h`	
Range	PWM_17_GTMCCU6_HIGH	The PWM channel is in the high state. The PWM channel output will be in the high state (3 or 5 V).
	PWM_17_GTMCCU6_LOW	The PWM channel is in low state. The PWM channel output will be in the low state (0 V).
Description	Output state of a PWM channel. Note that this will be read from the output state on the TOM or ATOM channel connected to the port pin. This will not be read directly from the port pin. However, both will be the same. This type is used to read PwmPolarity and PwmIdleState.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.5   Pwm_17_GtmCcu6_EdgeNotificationType

Table 65   Specification for Pwm_17_GtmCcu6_EdgeNotificationType

Syntax	`Pwm_17_GtmCcu6_EdgeNotificationType`	
Type	uint8	
File	`Pwm_17_GtmCcu6.h`	
Range	PWM_17_GTMCCU6_RISING_EDGE	Notification is called when the rising edge occurs on the PWM output signal.

Table 65   Specification for Pwm_17_GtmCcu6_EdgeNotificationType (continued)

	PWM_17_GTMCCU6_FALLING_EDGE	Notification is called when the falling edge occurs on the PWM output signal.
	PWM_17_GTMCCU6_BOTH_EDGES	Notification is called when both the rising and falling edge occur on the PWM output signal.
Description	Definition of the type of edge notification of a PWM channel. The edges are passed in the Pwm_17_GtmCcu6_EnableNotification() API.	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.6   Pwm_17_GtmCcu6_ChannelClassType

Table 66   Specification for Pwm_17_GtmCcu6_ChannelClassType

Syntax	`Pwm_17_GtmCcu6_ChannelClassType`	
Type	uint8	
File	`Pwm_17_GtmCcu6.h`	
Range	PWM_17_GTMCCU6_FIXED_PERIOD	
	PWM_17_GTMCCU6_FIXED_PERIOD_SHIFTED	
	PWM_17_GTMCCU6_FIXED_PERIOD_CENTER_ALIGNED	
	PWM_17_GTMCCU6_VARIABLE_PERIOD	
Description	Defines the class of a PWM channel	
Source	AUTOSAR	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.2.7   Pwm_17_GtmCcu6_ConfigType

Table 67   Specification for Pwm_17_GtmCcu6_ConfigType

Syntax	`Pwm_17_GtmCcu6_ConfigType`	
Type	Structure	
File	`Pwm_17_GtmCcu6.h`	
Range	Hardware dependent structure[]	The contents of the initialization data structure are hardware specific.
Description	This type definition is used to configure the overall PWM configuration. The pointer to the object of this data type is used in the Pwm_17_GtmCcu6_Init() API to initialize the PWM driver.	
Source	AUTOSAR	

Table 67	Specification for Pwm_17_GtmCcu6_ConfigType (continued)
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3 Functions - APIs

This section lists all the APIs of the PWM driver.

1.3.3.1 Pwm_17_GtmCcu6_DeInit

Table 68   Specification for Pwm_17_GtmCcu6_DeInit API

Syntax	`void  Pwm_17_GtmCcu6_DeInit`
	`(`
	`  void`
	`)`	

Service ID	0x01	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Non Reentrant	
Parameters (in)	-	-
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	De-initializes the PWM module and signals. This function de-initializes the PWM driver in the context of core from where this API is invoked, sets the state of the PWM signals to idle state, disables interrupts and edge notifications	
Source	AUTOSAR	
Error handling	PWM_17_GTMCCU6_E_UNINIT	
Configuration dependencies	PwmDeInitApi	
User hints	None	
SFR accessed	CCU6_CC63SR(w), CCU6_CC6SR(w), CCU6_IEN(rw), CCU6_ISR(rw), CCU6_TCTR4(rw), CPU_CORE_ID(r), GTM_ATOM_AGC_ENDIS_STAT(w), GTM_ATOM_AGC_GLB_CTRL(w), GTM_ATOM_AGC_OUTEN_STAT(w), GTM_ATOM_CH_CTRL(rw), GTM_ATOM_CH_IRQ_EN(w), GTM_ATOM_CH_IRQ_NOTIFY(w), GTM_TOM_CH_CTRL(rw), GTM_TOM_CH_IRQ_EN(w), GTM_TOM_CH_IRQ_NOTIFY(w), GTM_TOM_TGC0_ENDIS_STAT(w), GTM_TOM_TGC0_GLB_CTRL(w), GTM_TOM_TGC0_OUTEN_STAT(w), GTM_TOM_TGC1_ENDIS_STAT(w),	

1 Pwm_17_GtmCcu6 driver

Table 68	Specification for `Pwm_17_GtmCcu6_DeInit` API (continued)
	GTM_TOM_TGC1_GLB_CTRL(w), GTM_TOM_TGC1_OUTEN_STAT(w), SCU_CCUCON0(r), SCU_OSCCON(r), SCU_SYSPLLCON0(r), SCU_SYSPLLCON1(r), STM_TIM0(r)
	Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.2 Pwm_17_GtmCcu6_DisableNotification

Table 69	Specification for `Pwm_17_GtmCcu6_DisableNotification` API	

Syntax	`void  Pwm_17_GtmCcu6_DisableNotification`
	`(`
	`    const Pwm_17_GtmCcu6_ChannelType ChannelNumber`
	`)`	

Service ID	0x06	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Reentrant for different channels	
Parameters (in)	ChannelNumber	Numeric identifier of the PWM
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	Service to disable the PWM signal edge notification	
Source	AUTOSAR	
Error handling	PWM_17_GTMCCU6_E_PARAM_CHANNEL, PWM_17_GTMCCU6_E_UNINIT, PWM_17_GTMCCU6_E_CORE_CHANNEL_MISMATCH	
Configuration dependencies	PwmNotificationSupported	
User hints	None	
SFR accessed	CCU6_IEN(rw), CCU6_ISR(w), CPU_CORE_ID(r), GTM_ATOM_CH_IRQ_EN(w), GTM_ATOM_CH_IRQ_NOTIFY(w), GTM_TOM_CH_IRQ_EN(w), GTM_TOM_CH_IRQ_NOTIFY(w)	
	Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.3 Pwm_17_GtmCcu6_EnableNotification

Table 70 **Specification for** `Pwm_17_GtmCcu6_EnableNotification` **API**

Syntax	`void  Pwm_17_GtmCcu6_EnableNotification`
	`(`
	`    const Pwm_17_GtmCcu6_ChannelType ChannelNumber,`
	`    const Pwm_17_GtmCcu6_EdgeNotificationType Notification`
	`)`	

Service ID	0x07	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Reentrant for different channels	
Parameters (in)	ChannelNumber	Numeric identifier of the PWM
	Notification	Type of the notification PWM_17_GTMCCU6_RISING_EDGE or PWM_17_GTMCCU6_FALLING_EDGE or PWM_17_GTMCCU6_BOTH_EDGES
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	Service to enable the PWM signal edge notification according to the notification parameter	
Source	AUTOSAR	
Error handling	PWM_17_GTMCCU6_E_UNINIT, PWM_17_GTMCCU6_E_PARAM_CHANNEL, PWM_17_GTMCCU6_E_PARAM_NOTIFICATION, PWM_17_GTMCCU6_E_CORE_CHANNEL_MISMATCH, PWM_17_GTMCCU6_E_NO_NOTIF_CONFIGURED, PWM_17_GTMCCU6_E_INVALID_EDGE_NOTIF	
Configuration dependencies	PwmNotificationSupported	
User hints	None	
SFR accessed	CCU6_IEN(rw), CCU6_ISR(w), CPU_CORE_ID(r), GTM_ATOM_CH_IRQ_EN(w), GTM_ATOM_CH_IRQ_NOTIFY(w), GTM_TOM_CH_IRQ_EN(w), GTM_TOM_CH_IRQ_NOTIFY(w) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.4   Pwm_17_GtmCcu6_GetOutputState

Table 71   Specification for Pwm_17_GtmCcu6_GetOutputState API

Syntax	Pwm_17_GtmCcu6_OutputStateType Pwm_17_GtmCcu6_GetOutputState (const Pwm_17_GtmCcu6_ChannelType ChannelNumber)	
Service ID	0x05	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Reentrant for different channels	
Parameters (in)	ChannelNumber	Numeric identifier of the PWM
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Pwm_17_GtmCcu6_OutputStateType	-
Description	Service to read the internal state of the PWM output signal	
Source	AUTOSAR	
Error handling	PWM_17_GTMCCU6_E_UNINIT, PWM_17_GTMCCU6_E_PARAM_CHANNEL, PWM_17_GTMCCU6_E_CORE_CHANNEL_MISMATCH	
Configuration dependencies	PwmGetOutputState	
User hints	None	
SFR accessed	CCU6_CMPSTAT(r), CPU_CORE_ID(r), GTM_ATOM_CH_STAT(r), GTM_TOM_CH_STAT(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.5   Pwm_17_GtmCcu6_GetVersionInfo

Table 72   Specification for Pwm_17_GtmCcu6_GetVersionInfo API

Syntax	void Pwm_17_GtmCcu6_GetVersionInfo (Std_VersionInfoType * const VersionInfoPtr)
Service ID	0x08
Sync/Async	Synchronous

1 Pwm_17_GtmCcu6 driver

Table 72	Specification for `Pwm_17_GtmCcu6_GetVersionInfo` API (continued)	
ASIL Level	B	
Re-entrancy	Reentrant	
Parameters (in)	-	-
Parameters (out)	VersionInfoPtr	Pointer to store the version information of this module
Parameters (in - out)	-	-
Return	void	-
Description	Service returns the version information of the PWM module	
Source	AUTOSAR	
Error handling	PWM_17_GTMCCU6_E_PARAM_POINTER	
Configuration dependencies	PwmVersionInfoApi	
User hints	None	
SFR accessed	-	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.3.6 Pwm_17_GtmCcu6_Init

Table 73	Specification for `Pwm_17_GtmCcu6_Init` API	
Syntax	```void Pwm_17_GtmCcu6_Init(const Pwm_17_GtmCcu6_ConfigType * const ConfigPtr)```	
Service ID	0x00	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Non Reentrant	
Parameters (in)	ConfigPtr	Pointer to configuration set
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	This function initializes the PWM driver in the context of the core from where the Init is invoked.	

1 Pwm_17_GtmCcu6 driver

Table 73	Specification for `Pwm_17_GtmCcu6_Init` API (continued)
	Init function initializes the resource allocated to current core.
	Note: PWM driver initialization shall be called from any of the cores after the MCU driver initialization.
	Note: The users of the PWM driver shall not call API Pwm_17_GtmCcu6_Init() during a running operation i.e. All channels that are configured after Pwm_17_GtmCcu6_Init() will be treated as running channels.
	Note: The PWM driver's environment shall not call any API of the PWM driver before Pwm_17_GtmCcu6_Init() except the API Pwm_17_GtmCcu6_GetVersionInfo().
Source	AUTOSAR
Error handling	PWM_17_GTMCCU6_E_INIT_FAILED, PWM_17_GTMCCU6_E_ALREADY_INITIALIZED, PWM_17_GTMCCU6_E_CORE_NOT_CONFIGURED
Configuration dependencies	-
User hints	None
SFR accessed	CCU6_CC63SR(w), CCU6_CC6SR(w), CCU6_IEN(rw), CCU6_INP(rw), CCU6_ISR(rw), CCU6_MODCTR(rw), CCU6_PISEL0(rw), CCU6_PSLR(rw), CCU6_T12(w), CCU6_T12MSEL(rw), CCU6_T12PR(w), CCU6_T13(w), CCU6_T13PR(w), CCU6_TCTR0(rw), CCU6_TCTR2(rw), CCU6_TCTR4(rw), CPU_CORE_ID(r), GTM_ATOM_AGC_ENDIS_CTRL(rw), GTM_ATOM_AGC_ENDIS_STAT(w), GTM_ATOM_AGC_FUPD_CTRL(rw), GTM_ATOM_AGC_GLB_CTRL(w), GTM_ATOM_AGC_OUTEN_CTRL(rw), GTM_ATOM_AGC_OUTEN_STAT(w), GTM_ATOM_CH_CM0(w), GTM_ATOM_CH_CM1(w), GTM_ATOM_CH_CN0(w), GTM_ATOM_CH_CTRL(w), GTM_ATOM_CH_IRQ_EN(w), GTM_ATOM_CH_IRQ_MODE(w), GTM_ATOM_CH_IRQ_NOTIFY(w), GTM_ATOM_CH_SR0(w), GTM_ATOM_CH_SR1(w), GTM_TOM_CH_CM0(w), GTM_TOM_CH_CM1(w), GTM_TOM_CH_CN0(w), GTM_TOM_CH_CTRL(w), GTM_TOM_CH_IRQ_EN(w), GTM_TOM_CH_IRQ_MODE(w), GTM_TOM_CH_IRQ_NOTIFY(w), GTM_TOM_CH_SR0(w), GTM_TOM_CH_SR1(w), GTM_TOM_TGC0_ENDIS_CTRL(rw), GTM_TOM_TGC0_ENDIS_STAT(w), GTM_TOM_TGC0_FUPD_CTRL(rw), GTM_TOM_TGC0_GLB_CTRL(w), GTM_TOM_TGC0_OUTEN_CTRL(rw), GTM_TOM_TGC0_OUTEN_STAT(w), GTM_TOM_TGC1_ENDIS_CTRL(rw), GTM_TOM_TGC1_ENDIS_STAT(w), GTM_TOM_TGC1_FUPD_CTRL(rw), GTM_TOM_TGC1_GLB_CTRL(w), GTM_TOM_TGC1_OUTEN_CTRL(rw), GTM_TOM_TGC1_OUTEN_STAT(w), SCU_CCUCON0(r), SCU_OSCCON(r), SCU_SYSPLLCON0(r), SCU_SYSPLLCON1(r), STM_TIM0(r)
	Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.7 Pwm_17_GtmCcu6_InitCheck

Table 74	Specification for `Pwm_17_GtmCcu6_InitCheck` API
Syntax	`Std_ReturnType  Pwm_17_GtmCcu6_InitCheck` `(`

1 Pwm_17_GtmCcu6 driver

Table 74   **Specification for** `Pwm_17_GtmCcu6_InitCheck` **API (continued)**

	`const Pwm_17_GtmCcu6_ConfigType * const ConfigPtr` `)`	
Service ID	0x10	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Non Reentrant	
Parameters (in)	ConfigPtr	Valid address pointing to config
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	Std_ReturnType	E_OK : If initialization comparison is succeeds E_NOT_OK : If initialization comparison fails
Description	This routine verifies the initialization of the PWM driver. *Note: Init check should be call before invoking any runtime APIs.* *Sequence:* *1. Invoke Pwm_17_GtmCcu6_Init from a core.* *2. Invoke Pwm_17_GtmCcu6_InitCheck from the same core.*	
Source	IFX	
Error handling	-	
Configuration dependencies	PwmInitCheckApi	
User hints	None	
SFR accessed	CCU6_CC63SR(r), CCU6_CC6SR(r), CCU6_CLC(r), CCU6_IEN(r), CCU6_INP(r), CCU6_MODCTR(r), CCU6_PISEL0(r), CCU6_PSLR(r), CCU6_T12MSEL(r), CCU6_T12PR(r), CCU6_T13PR(r), CCU6_TCTR0(r), CCU6_TCTR2(r), CPU_CORE_ID(r), GTM_ATOM_AGC_ENDIS_CTRL(r), GTM_ATOM_AGC_ENDIS_STAT(r), GTM_ATOM_AGC_GLB_CTRL(r), GTM_ATOM_AGC_OUTEN_CTRL(r), GTM_ATOM_AGC_OUTEN_STAT(r), GTM_ATOM_CH_CM0(r), GTM_ATOM_CH_CM1(r), GTM_ATOM_CH_CN0(r), GTM_ATOM_CH_CTRL(r), GTM_ATOM_CH_IRQ_EN(r), GTM_ATOM_CH_IRQ_MODE(r), GTM_ATOM_CH_SR0(r), GTM_ATOM_CH_SR1(r), GTM_TOM_CH_CM0(r), GTM_TOM_CH_CM1(r), GTM_TOM_CH_CN0(r), GTM_TOM_CH_CTRL(r), GTM_TOM_CH_IRQ_EN(r), GTM_TOM_CH_IRQ_MODE(r), GTM_TOM_CH_SR0(r), GTM_TOM_CH_SR1(r), GTM_TOM_TGC0_ENDIS_CTRL(r), GTM_TOM_TGC0_ENDIS_STAT(r), GTM_TOM_TGC0_GLB_CTRL(r), GTM_TOM_TGC0_OUTEN_CTRL(r), GTM_TOM_TGC0_OUTEN_STAT(r), GTM_TOM_TGC1_ENDIS_CTRL(r), GTM_TOM_TGC1_ENDIS_STAT(r), GTM_TOM_TGC1_GLB_CTRL(r), GTM_TOM_TGC1_OUTEN_CTRL(r), GTM_TOM_TGC1_OUTEN_STAT(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	

Table 74	Specification for `Pwm_17_GtmCcu6_InitCheck` API (continued)
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.8 Pwm_17_GtmCcu6_SetDutyCycle

Table 75	Specification for `Pwm_17_GtmCcu6_SetDutyCycle` API	

Syntax	`void  Pwm_17_GtmCcu6_SetDutyCycle`
	`(`
	`    const Pwm_17_GtmCcu6_ChannelType ChannelNumber,`
	`    const uint32 DutyCycle`
	`)`	

Service ID	0x02	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Reentrant for different channels	
Parameters (in)	ChannelNumber DutyCycle	Numeric identifier of the PWM Duration of ON time of PWM
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	Service sets the duty cycle of the PWM channel. Since ATOM timer channel could support maximum of 24 bits, the DutyCycle parameter is uint32 when PwmDutyShiftInTicks parameter is set to true. When PwmDutyShiftInTicks is set to false, then the DutyCycle parameter is uint16. Function prototype will change accordingly based on the configuration of PwmDutyShiftInTicks parameter. The range of DutyCycle parameter is based on configuration of PwmDutyShiftInTicks parameter. When PwmDutyShiftInTicks parameter is true: 1. 0x0000 to 0xFFFF for TOM and CCU6 2. 0x000000 to 0xFFFFFF for ATOM When PwmDutyShiftInTicks parameter is false: 1. 0x0000 to 0x8000	
Source	AUTOSAR	
Error handling	PWM_17_GTMCCU6_E_PARAM_CHANNEL, PWM_17_GTMCCU6_E_UNINIT, PWM_17_GTMCCU6_E_CORE_CHANNEL_MISMATCH, PWM_17_GTMCCU6_E_PARAM_DUTY	
Configuration dependencies	PwmSetDutyCycle	

Table 75 **Specification for** `Pwm_17_GtmCcu6_SetDutyCycle` **API (continued)**

User hints	None
SFR accessed	CCU6_CC63SR(w), CCU6_CC6SR(w), CCU6_IEN(rw), CCU6_ISR(w), CCU6_T12PR(w), CCU6_T13PR(w), CCU6_TCTR4(rw), CPU_CORE_ID(r), GTM_ATOM_AGC_GLB_CTRL(w), GTM_ATOM_CH_CM0(w), GTM_ATOM_CH_CM1(w), GTM_ATOM_CH_IRQ_EN(w), GTM_ATOM_CH_IRQ_NOTIFY(w), GTM_ATOM_CH_SR0(w), GTM_ATOM_CH_SR1(w), GTM_TOM_CH_CM0(w), GTM_TOM_CH_CM1(w), GTM_TOM_CH_IRQ_EN(w), GTM_TOM_CH_IRQ_NOTIFY(w), GTM_TOM_CH_SR0(w), GTM_TOM_CH_SR1(w), GTM_TOM_TGC0_GLB_CTRL(w), GTM_TOM_TGC1_GLB_CTRL(w), SCU_CCUCON0(r), SCU_OSCCON(r), SCU_SYSPLLCON0(r), SCU_SYSPLLCON1(r), STM_TIM0(r)
	Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.9 Pwm_17_GtmCcu6_SetOutputToIdle

Table 76 **Specification for** `Pwm_17_GtmCcu6_SetOutputToIdle` **API**

Syntax	`void  Pwm_17_GtmCcu6_SetOutputToIdle`
	`(`
	`    const Pwm_17_GtmCcu6_ChannelType ChannelNumber`
	`)`	

Service ID	0x04	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Reentrant for different channels	
Parameters (in)	ChannelNumber	Input Channel ID
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	Service sets the PWM output to the configured Idle state.	
	Note: After the call of API Pwm_17_GtmCcu6_SetOutputToIdle(), variable period type channels shall be reactivated by Application using the API Pwm_17_GtmCcu6_SetPeriodAndDuty() to activate the PWM channel with the newly passed period and duty.	
	Note: After the call of API Pwm_17_GtmCcu6_SetOutputToIdle(), fixed period type channels shall be reactivated by Application using the API Pwm_17_GtmCcu6_SetDutyCycle() to activate the PWM channel with the duty and old period.	
Source	AUTOSAR	

Table 76   **Specification for** `Pwm_17_GtmCcu6_SetOutputToIdle` **API (continued)**

Error handling	PWM_17_GTMCCU6_E_UNINIT, PWM_17_GTMCCU6_E_PARAM_CHANNEL, PWM_17_GTMCCU6_E_CORE_CHANNEL_MISMATCH
Configuration dependencies	PwmSetOutputToIdle
User hints	None
SFR accessed	CCU6_CC63SR(w), CCU6_CC6SR(w), CCU6_IEN(rw), CCU6_ISR(rw), CCU6_TCTR4(rw), CPU_CORE_ID(r), GTM_ATOM_CH_CM0(w), GTM_ATOM_CH_CM1(w), GTM_ATOM_CH_IRQ_EN(w), GTM_ATOM_CH_IRQ_NOTIFY(w), GTM_ATOM_CH_SR0(w), GTM_ATOM_CH_SR1(w), GTM_TOM_CH_CM0(w), GTM_TOM_CH_CM1(w), GTM_TOM_CH_IRQ_EN(w), GTM_TOM_CH_IRQ_NOTIFY(w), GTM_TOM_CH_SR0(w), GTM_TOM_CH_SR1(w), GTM_TOM_TGC0_GLB_CTRL(w), GTM_TOM_TGC1_GLB_CTRL(w), SCU_CCUCON0(r), SCU_OSCCON(r), SCU_SYSPLLCON0(r), SCU_SYSPLLCON1(r), STM_TIM0(r)
	Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.3.10   Pwm_17_GtmCcu6_SetPeriodAndDuty

Table 77   **Specification for** `Pwm_17_GtmCcu6_SetPeriodAndDuty` **API**

Syntax	void Pwm_17_GtmCcu6_SetPeriodAndDuty
	(
	    const Pwm_17_GtmCcu6_ChannelType ChannelNumber,
	    const Pwm_17_GtmCcu6_PeriodType Period,
	    const uint16 DutyCycle
	)	

Service ID	0x03	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Reentrant for different channels	
Parameters (in)	ChannelNumber Period DutyCycle	Numeric identifier of the PWM New Period of PWM signal Duration of ON time of PWM
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	Service sets the Duty and Period of the PWM signal.	

Table 77	Specification for `Pwm_17_GtmCcu6_SetPeriodAndDuty` API (continued)
	Since ATOM timer channel could support maximum of 24 bits, the DutyCycle parameter is uint32 when PwmDutyShiftInTicks is set to true.
	If PwmDutyShiftInTicks is set to false, then the DutyCycle parameter is uint16. Function prototype will change accordingly based on the configuration of PwmDutyShiftInTicks parameter.
	The range of DutyCycle and Period parameter is based on configuration of PwmDutyShiftInTicks parameter.
	When PwmDutyShiftInTicks parameter is true:
	1. 0x0000 to 0xFFFF for TOM and CCU6
	2. 0x000000 to 0xFFFFFF for ATOM
	When PwmDutyShiftInTicks parameter is false:
	1. 0x0000 to 0x8000 for DutyCycle
	2. 0x0000 to 0xFFFF for TOM and CCU6 for Period
	3. 0x000000 to 0xFFFFFF for ATOM for Period
Source	AUTOSAR
Error handling	PWM_17_GTMCCU6_E_PERIOD_UNCHANGEABLE, PWM_17_GTMCCU6_E_PARAM_CHANNEL, PWM_17_GTMCCU6_E_UNINIT, PWM_17_GTMCCU6_E_CORE_CHANNEL_MISMATCH, PWM_17_GTMCCU6_E_PARAM_PERIOD, PWM_17_GTMCCU6_E_PARAM_DUTY
Configuration dependencies	PwmSetPeriodAndDuty
User hints	-
SFR accessed	CCU6_CC63SR(w), CCU6_CC6SR(w), CCU6_IEN(rw), CCU6_ISR(w), CCU6_T12PR(w), CCU6_T13PR(w), CCU6_TCTR4(rw), CPU_CORE_ID(r), GTM_ATOM_AGC_GLB_CTRL(w), GTM_ATOM_CH_CM0(w), GTM_ATOM_CH_CM1(w), GTM_ATOM_CH_IRQ_EN(w), GTM_ATOM_CH_IRQ_NOTIFY(w), GTM_ATOM_CH_SR0(w), GTM_ATOM_CH_SR1(w), GTM_TOM_CH_CM0(w), GTM_TOM_CH_CM1(w), GTM_TOM_CH_IRQ_EN(w), GTM_TOM_CH_IRQ_NOTIFY(w), GTM_TOM_CH_SR0(w), GTM_TOM_CH_SR1(w), GTM_TOM_TGC0_GLB_CTRL(w), GTM_TOM_TGC1_GLB_CTRL(w), SCU_CCUCON0(r), SCU_OSCCON(r), SCU_SYSPLLCON0(r), SCU_SYSPLLCON1(r), STM_TIM0(r)
	Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.

1.3.4 Notifications and Callbacks

This section lists all the notification and callbacks of the PWM driver.

1.3.4.1 Pwm_17_GtmCcu6_Isr

Table 78 **Specification for** `Pwm_17_GtmCcu6_Isr` **API**

Syntax	`void  Pwm_17_GtmCcu6_Isr`
	`(`
	`    const uint32 ChannelNumber,`
	`    const uint32 IsrStatus`
	`)`	

Service ID	None	
Sync/Async	Synchronous	
ASIL Level	B	
Re-entrancy	Non Reentrant	
Parameters (in)	ChannelNumber	PWM channel number which caused the interrupt.
	IsrStatus	This parameter gives the information about the comparator which caused the interrupt.
Parameters (out)	-	-
Parameters (in - out)	-	-
Return	void	-
Description	Handler to call the configured notification function	
Source	IFX	
Error handling	PWM_17_GTMCCU6_E_INVALID_ISR, PWM_17_GTMCCU6_E_UNINIT, PWM_17_GTMCCU6_E_PARAM_CHANNEL	
Configuration dependencies	-	
User hints	None	
SFR accessed	CPU_CORE_ID(r) *Note : The list includes all the SFRs accessed in the context of the API. It lists the SFRs accessed by the driver and called interfaces from other drivers. During runtime, the SFRs accessed from this list may vary based on configuration and execution context.*	
Autosar Version	Applicable for Autosar versions 4.2.2 and 4.4.0.	

1.3.5 Scheduled functions

The PWM driver does not provide any scheduled functions.

1.3.6 Interrupt service routines

The PWM driver does not provide any interrupt handlers.

1.3.7 Callout

The PWM driver does not provide any callout.

1.3.8 Errors Handling

This section describes the various error types reported by the PWM driver.

Error Name: Description	Source	Error ID (AS422)	Type (AS422)	Error ID (AS440)	Type (AS440)
PWM_17_GTMCCU6_E_ALREADY_INITIALIZED: Error is reported when Pwm_17_GtmCcu6_Init() API service is called while the PWM driver has already been initialized.	AUTOSAR	0x14	DET_SAFETY	0x14	DET_SAFETY
PWM_17_GTMCCU6_E_CORE_CHANNEL_MISMATCH: Error is reported when ChannelId is not allocated to the core from which the API is called.	IFX	0x65	DET_SAFETY	0x65	DET_SAFETY
PWM_17_GTMCCU6_E_CORE_NOT_CONFIGURED: Error is reported when PWM module is not configured for the core from which it was called.	IFX	0x64	DET_SAFETY	0x64	DET_SAFETY
PWM_17_GTMCCU6_E_INIT_FAILED: Error is reported when Pwm_17_GtmCcu6_Init() API is called with a wrong parameter.	AUTOSAR	0x10	DET_SAFETY	0x10	DET_SAFETY
PWM_17_GTMCCU6_E_INVALID_EDGE_NOTIF: Error is reported when Pwm_17_GtmCcu6_EnableNotification() API is called with an invalid notification type for PWM channel whose interrupt is routed to DSADC.	IFX	0xCD	SAFETY	0xCD	SAFETY
PWM_17_GTMCCU6_E_INVALID_ISR: Error is reported when PWM ISR is called with an incorrect compare match interrupt.	IFX	0xC8	SAFETY	0xC8	SAFETY
PWM_17_GTMCCU6_E_NO_NOTIF_CONFIGURED: Error is reported by Pwm_17_GtmCcu6_EnableNotification() API, when invoked on a non-DSADC triggering PWM	IFX	0xCC	SAFETY	0xCC	SAFETY

1 Pwm_17_GtmCcu6 driver

Error Name: Description	Source	Error ID (AS422)	Type (AS422)	Error ID (AS440)	Type (AS440)
channel with no notification configured.					
PWM_17_GTMCCU6_E_PARAM_CHANNEL: Error is reported when the API service is used with an invalid channel identifier.	AUTOSAR	0x12	DET_SAFETY	0x12	DET_SAFETY
PWM_17_GTMCCU6_E_PARAM_DUTY: Error is reported when API is called with an incorrect duty. When PwmDutyShiftInTicks = OFF Valid ranges are from 0 to 0x8000 When PwmDutyShiftInTicks = ON Valid ranges are from 0 to Period (16 bit / 24 bit)	IFX	0xC9	SAFETY	0xC9	SAFETY
PWM_17_GTMCCU6_E_PARAM_NOTIFICATION: Error is reported when Pwm_17_GtmCcu6_EnableNotification() API is called with an invalid notification type. This is reported when safety is enabled.	IFX	0xCA	SAFETY	0xCA	SAFETY
PWM_17_GTMCCU6_E_PARAM_PERIOD: Error is reported when API is called with an incorrect period. TOM/CCU6: valid values are 0-0xFFFF ATOM: valid values are 0-0xFFFFFF	IFX	0xCB	SAFETY	0xCB	SAFETY
PWM_17_GTMCCU6_E_PARAM_POINTER: Error is reported if the API is invoked with NULL pointer as a parameter.	AUTOSAR	0x15	DET_SAFETY	0x15	DET_SAFETY
PWM_17_GTMCCU6_E_PERIOD_UNCHANGEABLE: Error is reported when usage of unauthorized PWM service on PWM channel configured with a fixed period.	AUTOSAR	0x13	DET_SAFETY	0x13	DET_SAFETY

1 Pwm_17_GtmCcu6 driver

Error Name: Description	Source	Error ID (AS422)	Type (AS422)	Error ID (AS440)	Type (AS440)
PWM_17_GTMCCU6_E_UNINIT: Error is reported when the API service is used without module initialization.	AUTOSAR	0x11	DET_SAFETY	0x11	DET_SAFETY

1.3.9 Deviations and limitations

This section describes the deviations and limitations of the PWM driver.

1.3.9.1 Deviations

This section describes the deviations of the PWM driver.

1.3.9.1.1 Software specification deviations

This section describes the deviations from software specification.

Table 79   Known deviations

Reference	Deviation
ECUC_Pwm_00124 :PwmPeriodDefault	According to AUTOSAR specification, period value is entered in seconds, instead PWM driver expects value to be entered in ticks.
PWM_FIXED_PERIOD_CENTER_ALIGNED	PWM_FIXED_PERIOD_CENTER_ALIGNED range is added along with the other AUTOSAR ranges as microcontroller specific feature.
Safety error for unintended service request	Refer to Reporting of unintended service requests.

1.3.9.1.2 AMDC Violations

This section describes the violations reported by the vector AMDC checker tool with respect to AUTOSAR.

Table 80   Violations reported by AMDC checker tool for A202

AMDC Rule	A202
Description	AUTOSAR insists period value to be given in terms of seconds. But in current PWM implementation due to support of Non-Autosar requirement i.e PwmDutyShiftInTicks parameter. The period value is given in terms of ticks, hence PwmPeriodDefault type is changed to Integer.

Table 81   Violations reported by AMDC checker tool for A207

AMDC Rule	A207
Description	To support Non-Autosar requirement i.e PwmDutyShiftInTicks is ON range of

Table 81   Violations reported by AMDC checker tool for A207 (continued)

	PwmDutycycleDefault parameter should be 24 bit. Therefore for this parameter max value for range attribute is deviated from AUTOSAR. Refer below for additional details.
	- When PwmDutyShiftInTicks is OFF, the value is relative to period. (Autosar range 16 bit) 0 to 0x8000
	- When PwmDutyShiftInTicks is ON, the value is in absolute ticks. (Non Autosar range 24 bit)
	- 0 to 0x8000 if PwmDutyShiftInTicks is STD_OFF for TOM/ATOM/CCU6
	- 0 to 0xFFFF if PwmDutyShiftInTicks is STD_ON and the module is TOM/CCU6
	- 0 to 0xFFFFFF if PwmDutyShiftInTicks is STD_ON and the module is ATOM

1.3.9.1.3   VSMD Violations

This section describes the violations reported by the EB VSMD checker tool with respect to AUTOSAR.

Table 82   Violations reported by VSMD checker tool for EcucSws_1007

Rule ID:	EcucSws_1007
VSMD Node(s):	/AURIX2G/EcucDefs/Pwm/PwmChannelConfigSet/ PwmChannel/PwmDutycycleDefault
Description:	For Integer and Float Parameters the MIN values must be >= and the MAX values <= as in the StMD.
Additional Information:	To support Non-Autosar requirement i.e PwmDutyShiftInTicks is ON range of PwmDutycycleDefault parameter should be 24 bit. Therefore for this parameter max value for range attribute is deviated from AUTOSAR. Refer below for additional details.

- When PwmDutyShiftInTicks is OFF, the value is relative to period. (Autosar range 16 bit) 0 to 0x8000

- When PwmDutyShiftInTicks is ON, the value is in absolute ticks. (Non Autosar range 24 bit)

- 0 to 0x8000 if PwmDutyShiftInTicks is STD_OFF for TOM/ATOM/CCU6

- 0 to 0xFFFF if PwmDutyShiftInTicks is STD_ON and the module is TOM/CCU6

- 0 to 0xFFFFFF if PwmDutyShiftInTicks is STD_ON and the module is ATOM |

Table 83   Violations reported by VSMD checker tool for EcucSws_1008

Rule ID:	EcucSws_1008

Table 83   Violations reported by VSMD checker tool for EcucSws_1008 (continued)

VSMD Node(s):	/AURIX2G/EcucDefs/Pwm/PwmChannelConfigSet/ PwmChannel/PwmPeriodDefault
Description:	For Containers, Parameters and References the parameter type may stay the same or may be changed to the corresponding derived-type in the VSMD.
Additional Information:	AUTOSAR insists period value to be given in terms of seconds. But in current PWM implementation due to support of Non-Autosar requirement i.e PwmDutyShiftInTicks parameter. The period value is given in terms of ticks, hence PwmPeriodDefault type is changed to Integer.

1.3.9.2   Limitations

This section describes the limitations of PWM driver.

PWM driver limitation:

- Fixed period center-aligned channels should be present in the same TGC/AGC of the same TOM/ATOM module where the referenced fixed period channel is present.

- The transition from 100% to other percentages with the shift value will not be the same as the ideal case: The transition from 100% to other percentages with a shift will not happen as per expected signal in case of coherent update. This is because three signal level changes are needed to produce the expected waveform. The registers CM0 and CM1 are updated from shadow registers at the end of the cycle. The signal will remain in the same level till the CM0 value is reached, as shown in the following diagram.

  Note: This limitation is also applicable for cases where call is made from the `Pwm_17_GtmCcu6_SetOutputToIdle` *API to some other duty.*

  Note:

  CM0 - Period Match register

  CM1- Duty Match register

  SR0 - Period Match Shadow register

  SR1 - Duty Match Shadow register

1 Pwm_17_GtmCcu6 driver
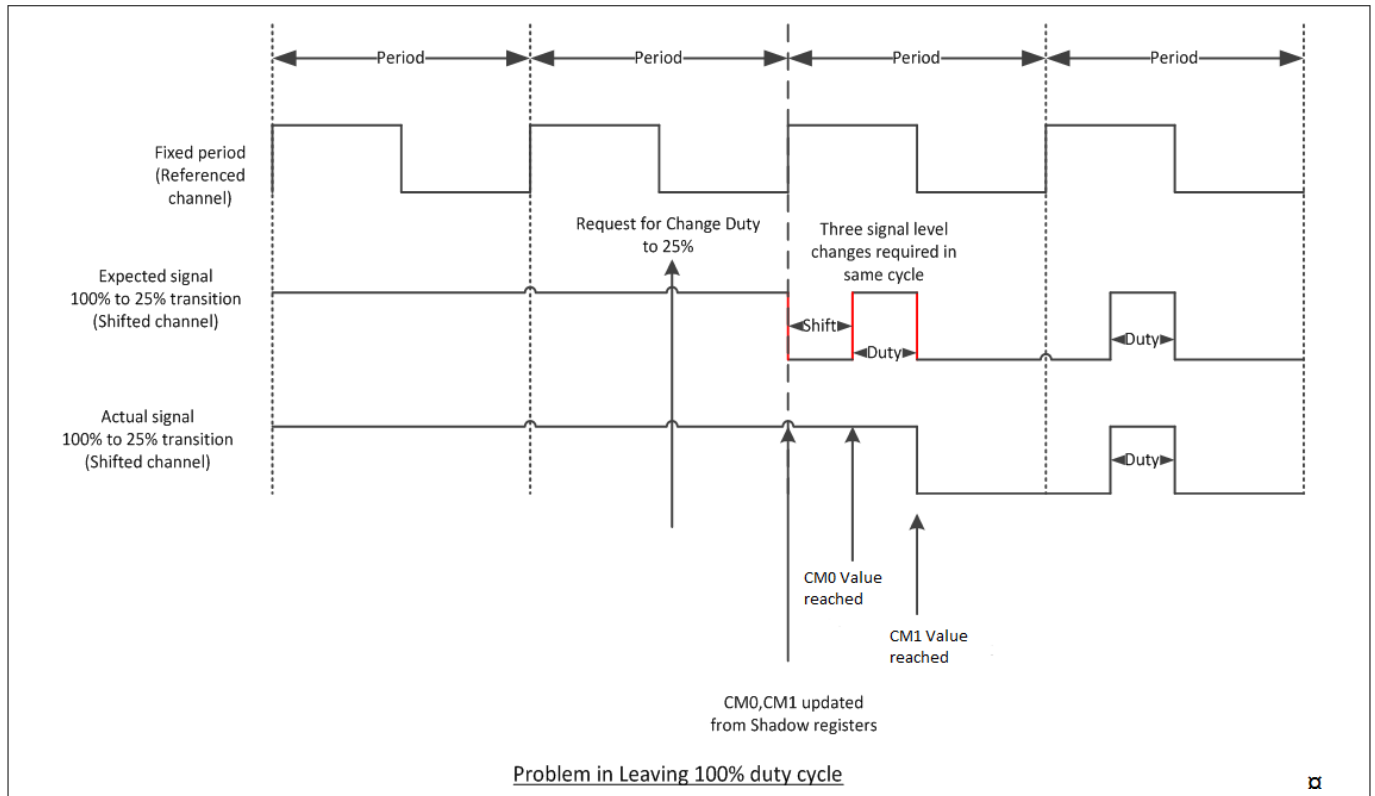
Figure 5   Transition from 100% to other percentages with a shift value

- In case of non-coherent update the register CMx (CM0 period and CM1 duty) are updated directly (applicable for `Pwm_17_GtmCcu6_SetDutyCycle, Pwm_17_GtmCcu6_SetPeriodAndDuty` and `Pwm_17_GtmCcu6_SetOutputToIdle` APIs). Depending on the point of time of the update of CMx registers in relation to the actual values of CN0 and CMx, the new duty cycle is applied in the current period or the following period. The new duty cycle may jitter from update to update by a maximum of one period. Refer the following diagram. For shifted and center aligned channels update will be done with respect to period of reference fixed period channel.
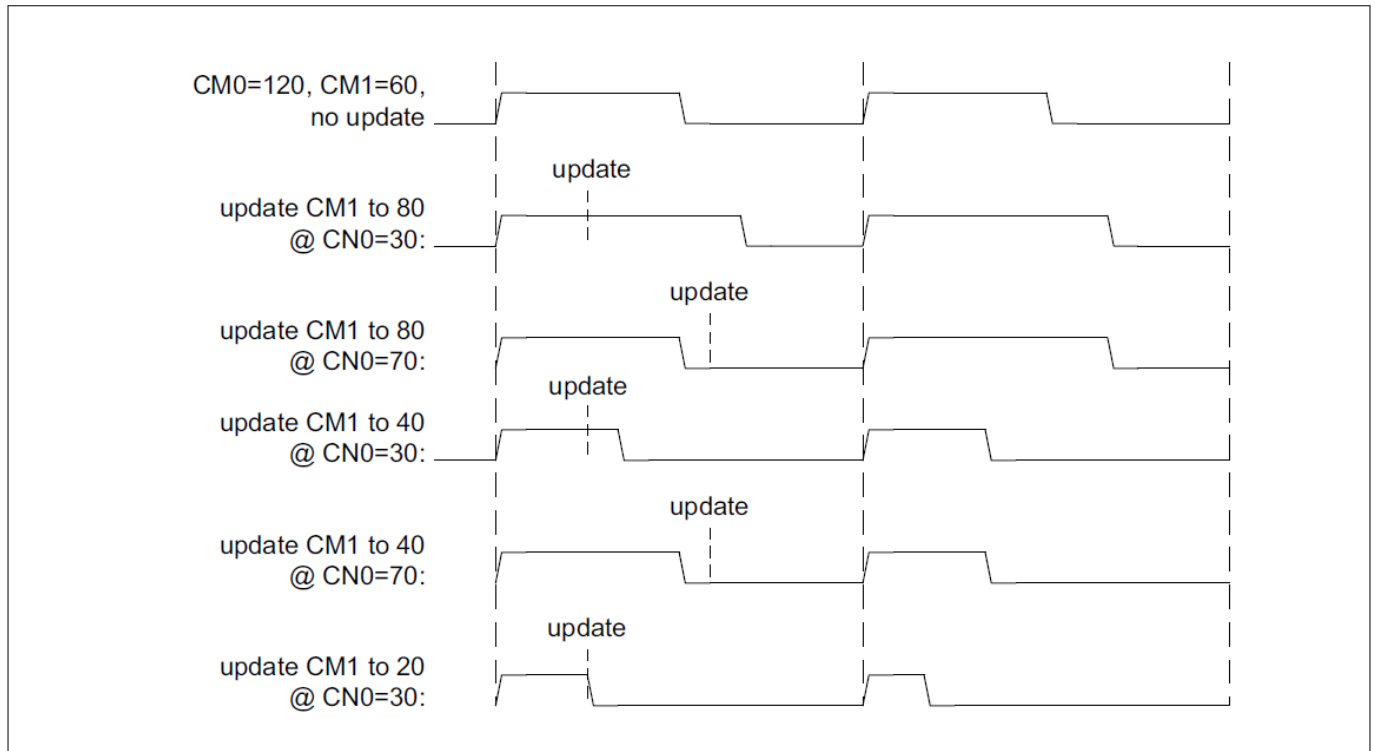
Figure 6   **Non-coherent update of duty-cycle**

- In case of coherent updated shifted channels where shift is not equal to zero, when duty change is requested from a condition where shift + duty is greater than period to shift + duty less than period then the signal will remain on duty for more than a period value. Refer figure below. If a request is made to change the duty from the condition shift and duty-cycle value together is greater than period (Shift+Duty > Period) to less than period (Shift+Duty < Period) in the case for a coherent updated shifted channel where shift value is not equal to zero, the following behavior will be observed.
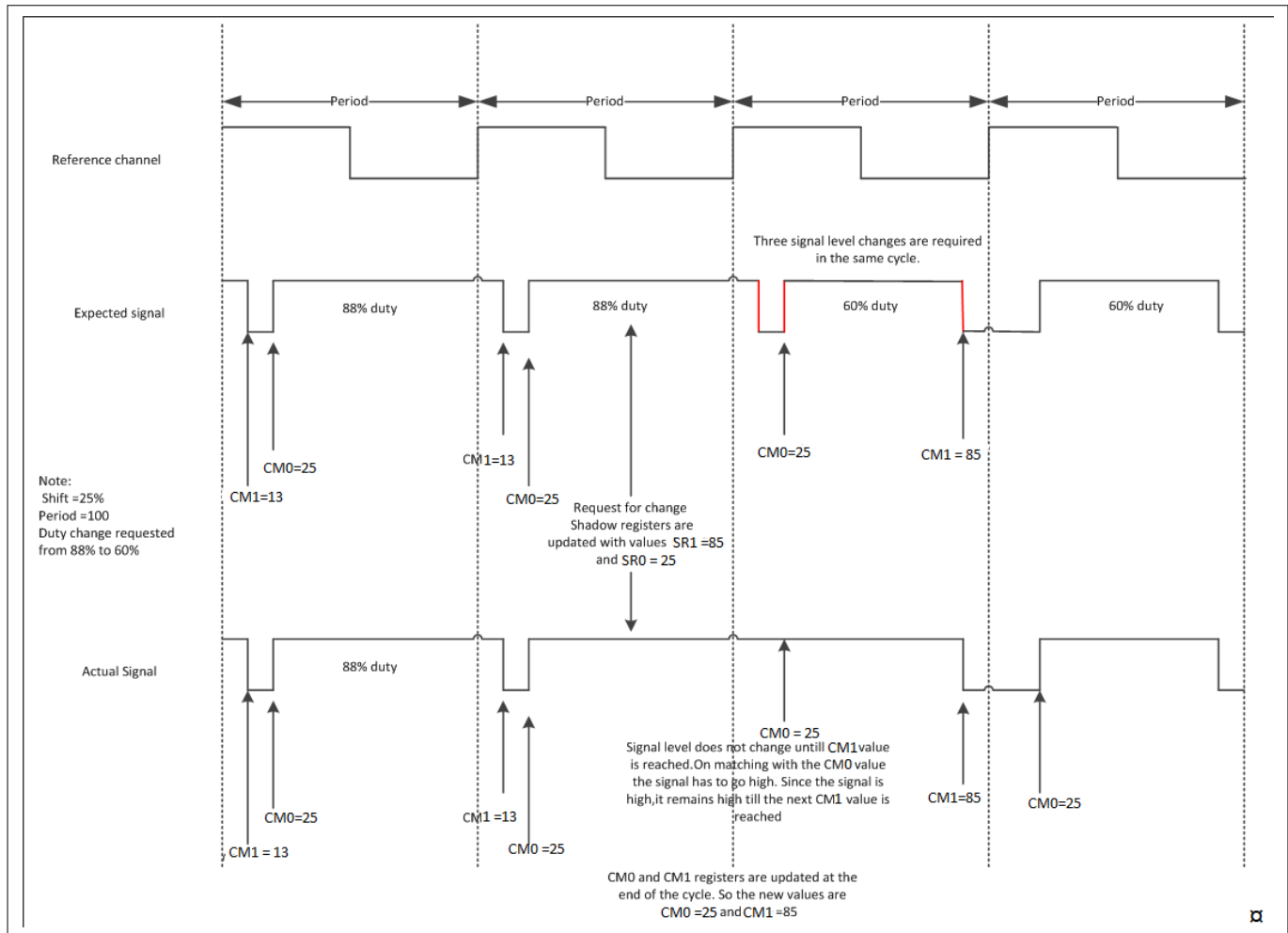
Figure 7   Duty update with a shift value behavior

In this diagram, on compare match with CM0 value the signal will go high and on compare match with CM1 value the signal will go low. The values are calculated as CM0/SR0= Shift and CM1/SR1= (Shift+Duty % period). Here in the figure initially 88% duty was requested and shift was 25%. So if the period is 100 ticks then, CM0 = 25 and CM1 = 13 ((25+88) %100).Later when the duty of 60% is requested the values calculated are SR0=25 (Shift) and SR1 = 85 using ((25+60) %100).

The CM1 and CM0 values are updated from Shadow registers SR1 and SR0 respectively (in the case of coherent update). When the duty change is requested then the function Pwm_17_Gtm_SetDutyCycle will update the corresponding shadow registers and return immediately after completion of the function. The updating of shadow registers to CM0 and CM1 registers will happen only after the end of the period.

Since the new values are updated to CM1 register from shadow register the signal will remain in the same state till the CM1 value is reached as shown in the figure. The signal will remain high for a time greater than period as shown in the figure above because there are three signal level changes required in the same cycle which could not be handled with two values CM0 and CM1 available at our disposal.

- In case shifted channels (PwmHandleShiftByOffset = true) the changes will be happened by the end of the period of the same channel. It will not depend on the reference channel. The limitations in table below are also applicable. Background: A global configuration parameter has been added PwmHandleShiftByOffset in the PwmGeneral container and once this parameter is selected then the shifted channels are configured similar to fixed period channels and but they are started by an offset. The offset is calculated by offset = period – shift. This offset acts as the required shift. All the channels in the TGC where these kind of shifted channels are present are triggered by a global host trigger. This host trigger will help in enabling all the channels in the TGC at the same time, and since all the channels are started the same time the

corresponding shift is achieved accordingly. Please refer to respective API's to know how these kind of shifted channels are handled.

Table 84   Difference between the two types of handling for shifted channels

Subject	FPS(PwmHandleShiftByOffset=false)	FPS(PwmHandleShiftByOffset=true)
Handling of shift	Compare register is used	Counter register is preloaded with period - shift
Handling of period	Fixed period reference channel's compare register is used	No reference channel, same channels compare register is used, however in configuration reference channel is provided for AUTOSAR compatibility
Host trigger	Host trigger is not used	Host trigger is used.
TGC/AGC restriction due to host trigger	Channels in the same TGC can be shared across other drivers.	Channels in the same TGC cannot be shared across other drivers.
Handling of shift + duty crossing period	For more than one cycle the output line is either high or low during change of duty	No limitation in this case.
Leaving 100% duty cycle	The signal will remain in the 100% duty till the next compare value is reached; The shift will not be visible.	The signal will remain in the 100% duty till the next compare value is reached; The shift will not be visible.
Update of new duty cycle	In case of coherent update the new duty cycle always happens at the end of reference channel period	In case of coherent update the new duty cycle happens at the end of its own period. This means in corner cases the new duty cycle might happen in the same cycle of the configured reference channel.
Pwm_17_GtmCcu6_SetOutputToIdle function	Always happens in the next cycle.	Always happens in the next cycle.
Update of 0% and 100% duty cycle	Happens at the start of the next cycle of reference channel	Happens at the start of the next cycle after the shift value is elapsed.
Handling of shift + duty exactly equal to period	For more than one cycle an incorrect duty cycle could be observed when a transition is asked from API Pwm_17_GtmCcu6_SetOutputToIdle or during change of duty to 100%	No limitation in this case

- Channels of TGC or AGC can be shared across other drivers but users of the other drivers or within the PWM driver should not introduce the sequence of referenced fixed period and fixed period shifted channels when PwmHandleShiftByOffset parameter is FALSE or referenced fixed period and fixed period center-aligned channels in-between the other channels sequence. This is because, introducing channel sequence in-between will break the sequence of other channels.

- User shall consider the following points when PwmHandleShiftByOffset is TRUE:

1 Pwm_17_GtmCcu6 driver

- To ensure synchronous start of all the shifted channels at the same time, configure all the shifted channels including the fixed period channel in the same TGC or AGC module (Recommended).

- If maximum shifted channels and fixed period channel are configured across the TGC module numbers or AGC module numbers, there could be a delay on the shifted channels based on the execution time of the PWM initialization sequence.

- User can measure the actual offset during the PWM initialization and compensate the delay of the shifted channels accordingly in the configuration.

- Execution time depends on the initialization sequence and number of Host-Triggers.

PWM Initialization Sequence:

- Driver initializes all the relevant registers of the configured hardware with the values passed in the configuration pointer structure before generating the PWM signal.

- Driver performs the Host-Trigger when PwmHandleShiftByOffset is TRUE to generate the PWM signal. i.e,

- Loop from lowest configured TOM module to the highest configured TOM module

- Host-trigger of TGC0 for current TOM

- Host-trigger of TGC1 for current TOM

- Loop from lowest configured ATOM module to the highest configured ATOM module

- Host-trigger of AGC for current ATOM

- When notification for 0% or 100% is enabled by the user for fixed and variable period channels the notification for falling edge does not work for 0% duty for polarity high, similarly rising edge notification will not work 0% duty for polarity low, however the user can configure notification as both edges to get notification without worrying about the polarity.

- For fixed and variable period channels when notification is asked for falling edge for a channel whose polarity is HIGH and idle state LOW after calling SetOutputToIdle function, notifications are not generated. Similarly if a channel with polarity LOW and idle state HIGH and rising edge notification is asked, notification is not generated. This is because internally SetOutputToIdle function moves to 0% or 100% duty based on IDLE state and polarity.

- In case of non-coherent PWM channel with polarity PWM_LOW and idle state PWM_HIGH, when transition from idle state to non-zero duty cycle state, with rising edge notification enabled, a notification is provided even when there is no rising edge in the first cycle only, this due to CM1 compare match.

- In case of coherent PWM channel with polarity PWM_LOW and idle state PWM_LOW, when transition from idle state to non-zero duty cycle state, with rising edge notification enabled, a notification is provided even when there is no rising edge in the first cycle only, this due to CM0 compare match.

- Set output to idle for CCU6 channels

  Applicable APIs: `Pwm_17_GtmCcu6_SetOutputToIdle` and `Pwm_17_GtmCcu6_DeInit`.

  Behavior1:

  Precondition: Polarity is HIGH and idle state is LOW

  If set to idle is called when current state is HIGH, then switch to idle occurs after a delay due to hardware limitation.

  Behavior2:

  Precondition: Polarity is LOW and Idle State is HIGH.

  If set to idle is called when current state is LOW, then switch to idle occurs after a delay due to hardware limitation.

  Hardware Limitation: Duty and period values cannot be updated directly to running register.

  Workaround in design is to stop the timer, update new values and perform shadow transfer, which leads to delay.

  [cover parentID PWM={A0182969-A76A-4eee-AEF8-0FEE0170C4CE}]

1 Pwm_17_GtmCcu6 driver

- CCU6 Configuration:

  Non-coherent update of duty cycle and period is not supported for PWM channels of type CCU6.

  Centre aligned channels is not supported for PWM channel of type CCu6.

- For PWM channels of type CCU6, the first cycle will have period value as period+1 instead of period.

- Callback notification for a TOM or ATOM channel will be invoked by DSADC and not by PWM, when for the channel McuTomChannelEventHandledByDsadc = ON or McuAtomChannelEventHandledByDsadc = ON respectively.

Revision History

Table 85 *Revision History*

Date	Version	Description
2021-03-04	5.0	Document is released.
2021-03-04	4.1	Updated limitations section.
2020-12-02	4.0	Document is released.
2020-12-02	3.1	Updated description and default value of PwmInitCheckApi parameter.
2020-11-26	3.0	Document is released.
2020-11-26	2.1	Updated SFR access information for Notifications and Callbacks.
2020-11-10	2.0	Document is released.
2020-11-02	1.1	- Updated limitations section - Updated SFR access information for APIs
2020-08-13	1.0	Document is released.
2020-08-06	0.1	- Initial Version - Pwm_17_GtmCcu6 driver chapter moved from MCISAR_TC3xx_UM_Basic to this document - Added AMDC and VSMD violation tables - Updated Limitations section

Trademarks

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