



# MCAL Architecture Specification

Document Version : 73

Document Owner : Texas Instruments

Document Status : Published

Last Approval Date : Oct 10, 2022

**TI Confidential - NDA Restrictions**

**Copyright ©2022 Texas Instruments Incorporated**

- [Revision History](#)
- [Terms and Abbreviations](#)



- Introduction
- Overview
- Purpose and Scope
- Stakeholders
- References
- Software Architecture Overview
- MCAL Software Architecture
- Design Considerations
- Variants, Configuration and System family concepts
- Reusability, modularity and expandability
- Assumptions
- Constraints
- Dependencies
- Software High Level Design
- Naming conventions used/followed in source code files
- Configurator and Integration into MCAL
- Directory Description
- Data structures and resources
- Event Overview
- Operating System Details
- Task Details
- Data Flow Diagram
- Control Flow Diagram
- Commercial off the shelf/reusable components used
  
- Data storage and protection
- Resource Partition



- Interrupt Map
- Memory Assignment
- Add other Resources
- Debuggability/traceability/Visibility
- Dynamic Behavior
- SW Modules
- DIO Driver
- ADC Driver
- WDG Driver
- CAN Driver
- GPT Driver
- SPI Driver
- PWM Driver
- PWM (ePWM) Driver
- ICU Driver
- CDD IPC Driver
- FLS Driver
- MCU Driver
- ETH Driver
- BSW Stubs
- Performance Objectives
- Resource Consumption Objectives
- Critical timing and Performance
- Decision Analysis & Resolution (DAR)
  
- Testing Guidelines
- Template Revision History



## 1 Revision History

Version	Date	Author	Document Status	Comments
0.1	04 Jun 2021	Nikki Shah	<span style="background-color: green; color: white; padding: 2px 5px;">DONE</span>	Initial Version
0.2	15 Jul 2021	Nikki Shah	<span style="background-color: green; color: white; padding: 2px 5px;">DONE</span>	Peer Review Comments Addressed
0.3	15 Jul 2021	Nikki Shah	<span style="background-color: green; color: white; padding: 2px 5px;">DONE</span>	Inspection Review in progress
0.4	30 Jul 2021	Nikki Shah	<span style="background-color: green; color: white; padding: 2px 5px;">DONE</span>	Review comments Addressed.



Version	Date	Author	Document Status	Comments
0.5	19 Jan 2022	Nikki Shah	<span style="background-color: #2e7131; color: white; padding: 2px 5px; border-radius: 3px;">DONE</span>	IP-ReArch - JACINTOREQ-1870. This document is made SoC Agnostic. HW IP details added.
v61	04 Mar 2022	Nikki Shah	<span style="background-color: #2e7131; color: white; padding: 2px 5px; border-radius: 3px;">DONE</span>	Review Comments Addressed.
v.67	03 Aug 2022	Rohit Tiwari	<span style="background-color: #2e7131; color: white; padding: 2px 5px; border-radius: 3px;">DONE</span>	Added MCU Specification
v.68	04 Aug 2022	Rohit Tiwari	<span style="background-color: #2e7131; color: white; padding: 2px 5px; border-radius: 3px;">DONE</span>	Added MCU in the HW IP mapping table

## 2 Terms and Abbreviations

<b>Abbreviation /Term</b>	<b>Meaning / Explanation</b>
MCAL	Microcontroller Abstraction Layer
ADC	Analog to Digital Converter
WDG	Watchdog
DIO	Digital Input Output
ETH	Ethernet
SPI	Serial Peripheral Interface
CAN	Controller Area Network
PWM	Pulse Width Modulator



Abbreviation /Term	Meaning / Explanation
ICU	Input Capture Unit
GPT	General Purpose Timer
MCU	Microcontroller Unit
CDD	Complex Device Drivers
IPC	Inter-Processor Communication
FLS	Flash
AUTOSAR	Automotive Open System Architecture Standard
DET	Default Error Tracer
DEM	Diagnostic Event Manager



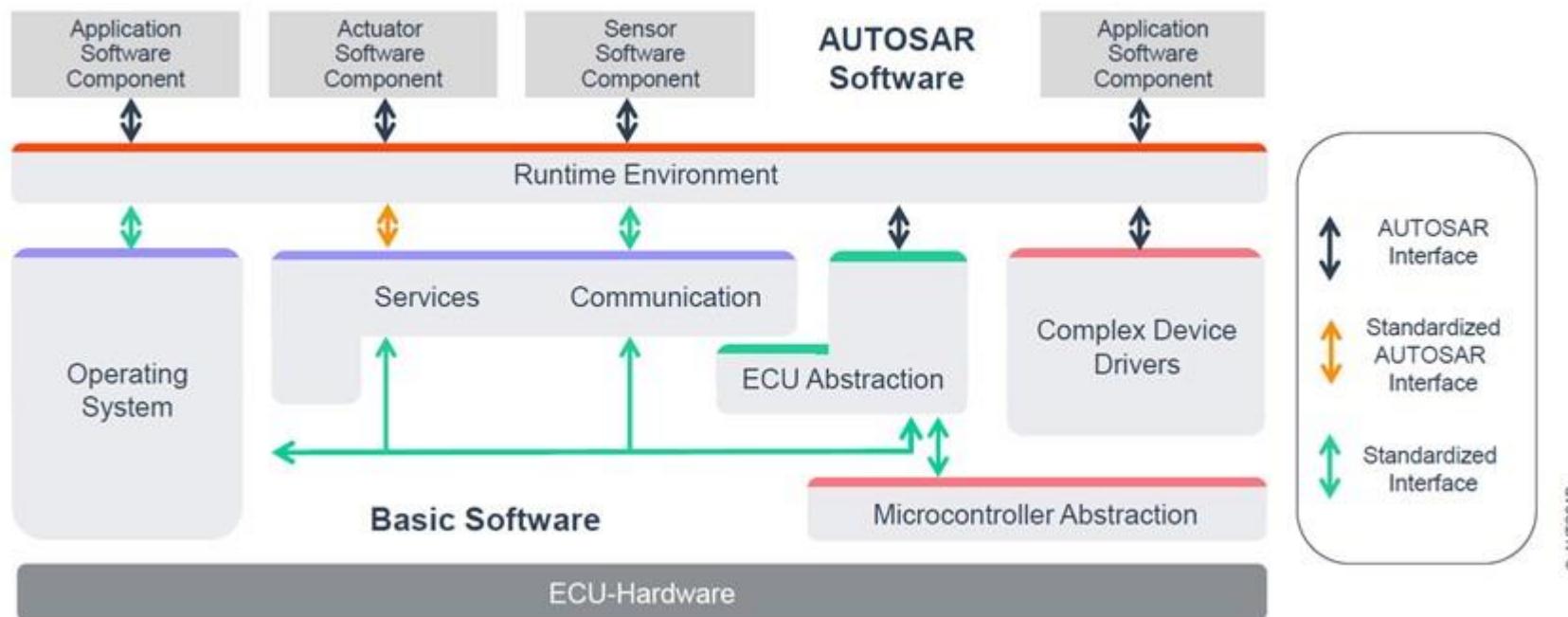
## 3 Introduction

### 3.1 Overview

This document describes the architecture of the Microcontroller Abstraction Layer (MCAL) developed TI SoCs according to AUTOSAR architecture.

AUTOSAR (AUTomotive Open System ARchitecture) is an open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers and tool developers.

The AUTOSAR Architecture distinguishes on the highest abstraction level between three software layers: Application, Runtime Environment and Basic software.



© AUTOSAR

Figure 1: AUTOSAR Architecture

The AUTOSAR Basic Software is further divided in the layers: Services, ECU Abstraction, Microcontroller Abstraction and Complex Drivers.

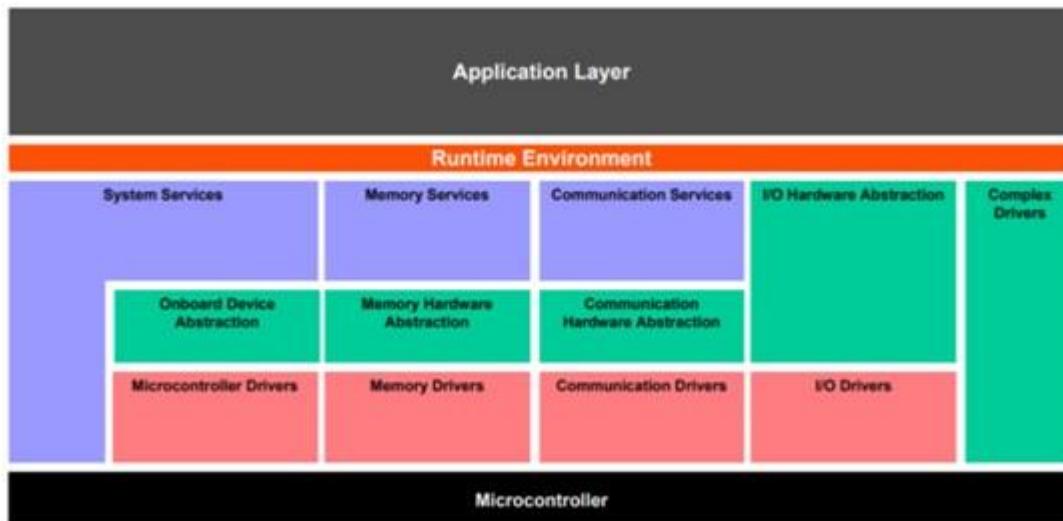


Figure 2: AUTOSAR Architecture Layers

The **Microcontroller Abstraction Layer (MCAL)** is the lowest software layer of the Basic Software. It contains internal drivers, which are software modules with direct access to the µC and internal peripherals. The main task is to make higher software layers independent of the µC.

The **Complex Device Drivers Layer (CDD)** spans from the hardware to the RTE. The main task is to provide the possibility to integrate special purpose functionality, e.g. drivers for devices:



- Not specified within AUTOSAR •      With very high timing constraints or
- For migration purposes.

## 3.2 Purpose and Scope

The purpose of this project is to provide a set of configurable **Microcontroller Abstraction Layer** (MCAL) drivers for **TI SoCs** in accordance to the **Automotive Open System Architecture standard (AUTOSAR) release 4.3 revision 1**.

MCAL includes implementation for below module groups:

1. Communication Drivers: CAN, SPI, ETH\*
2. I/O Drivers: ADC, DIO, PWM, ICU
3. Memory Drivers: FLS
4. Microcontroller Drivers: MCU, WDG, GPT
5. CDD: IPC

This document captures the overall software architecture and principles to be used for MCAL development. This document is to be used as reference, along with the AUTOSAR Specifications, for designing and implementing the software drivers.

Note: More detailed design and implementation details will be captured as part of design documents.

\*Note: ETH driver detailed requirements are not included as part of this document as it is yet to be analyzed.



### 3.3 Stakeholders

1. TI Software Developers who are designing and developing the MCAL software (includes developers, test engineers, software architect and stake holders).
2. Consumers
  - a. SDK Team members (including developers, integrators, test engineers, software architect and stake holders).
  - b. AUTOSAR Stack vendors/customers who will be integrating the MCAL.

### 3.4 References

1. MCAL Functional Requirements (Bitbucket project: EP ADAS MCAL) 2.  
AUTOSAR references for each driver are linked in SW Modules  
Section.
3. Refer to SoC User Manual for SoC TRM.

## 4 Software Architecture Overview

### 4.1 MCAL Software Architecture

TI as SoC vendor is expected to provide MCAL layer for these devices and AUTOSAR stack vendor would integrate provided MCAL driver to complete the AUTOSAR stack on TI SoCs. Typically provided MCAL drivers / modules would be used in AUTOSAR stack with no modifications (the MCAL drivers/modules could be re-configured to suite ECU/ System needs, but no source code changes).

This mandates that provided MCAL modules provides best-possible performance, features and quality, the requirements listed in sections below highlight required MCAL module, it's functional needs and collateral's that demonstrate quality of drivers/modules.

As part of AUTOSAR specification, each of the MCAL driver should be configurable using an GUI (EB Tresos). Each MCAL module will be accompanied with GUI based configurator. Derivation of architecture requirements for each driver module is done based on AUTOSAR specification. Each individual MCAL module provides standardized interfaces to upper software layers.

## 5 Design Considerations

Primary objective of the product is to provide drivers / modules & configurators required for AUTOSAR stack to be functional on TI SoCs.

- Function
- MCAL SW Provides MCU HW abstraction for customer SW development
- MCAL SW delivery from TI is integrated directly into customer SW, hence MCAL module safety collateral can be leveraged by Customers for their SW stack safety process



- MCAL SW component supports the safety diagnostics for the various supported IP, per the definition in the safety manual
- Development and Target Environment • MCAL to be delivered as own product.
- MCAL executed in context of no Operating system
- Integration
- MCAL integrates CSL Libraries into its driver code for low level hardware interaction.
- MCU Demos interface MCAL drivers and Complex Device Drivers to perform use-case demonstrations.
- FreeRTOS and BootLoader is used during testing of MCAL Drivers on TI EVMs.

## 5.1 Variants, Configuration and System family concepts

MCAL provides the driver implementation for the various TI device. Note, that care is taken to abstract the device-specific aspects to folders in order to support multiple devices.

MCAL follows AUTOSAR specification 4.3.1.

## 5.2 Reusability, modularity and expandability

MCAL is abstracted to separate the SoC specific integration and core MCAL driver implementation. In this way the core MCAL modules can be re-used and/or expanded for future devices.

The existing CSL-R and CSL-FL for the hardware IP modules that will be used by the MCAL drivers will be reused from SDK as applicable for the software to interface with the hardware. It will be included in the MCAL software package itself and will be renamed in order to avoid conflicts with the existing CSL code and usage.

MCAL is architected such that new SoC variant support can be added without disturbing the existing support. Some point to note about MCAL architecture:



1. The core driver code (along with the HW IP source code) has been implemented in way that is SoC agnostic. This means that this code would not need any modification when new SoC support is added (given that HW IP stays the same).
2. All the SoC specific code is moved to the configuration plugins for each driver modules. When a new SoC is added, the new plugin source created will accommodate the SoC level changed required (ex. instance numbers).
3. Each SoC will have its individual configuration plugin per driver module. User will have to choose based on SoC of interest.

## 5.3 Assumptions

Some key assumptions are as listed below

- Version of AUTOSAR: MCAL module specification are based on AUTOSAR version 4.3.1
- Compiler: MCAL drivers/modules provided shall be compiled using TI compiler (i.e. no third party compilers shall be employed, any exceptions shall be adequately documented).

## 5.4 Constraints

MCAL is component in AUTOSAR stack and AUTOSAR requirements detail the required interfaces for each module. These interfaces, including ECUC parameters shall be adhered to expedite integration of MCAL modules into third party AUTOSAR stack.

## 5.5 Dependencies

Some key **dependencies** are as listed below



- **SDK Utilities/Libraries:** Some of the utility (such as interrupt hooks, routines to display text on console, take inputs from console, start-up routines) shall be re-used from SDK. These utilities shall only be used to demonstrate functionality of MCAL drivers/modules, and used only by MCAL example applications.
- **Configurator:** Shall rely on a third party tool to configure MCAL modules/drivers, such as Elektrobit's Studio (EB)



## 6 Software High Level Design

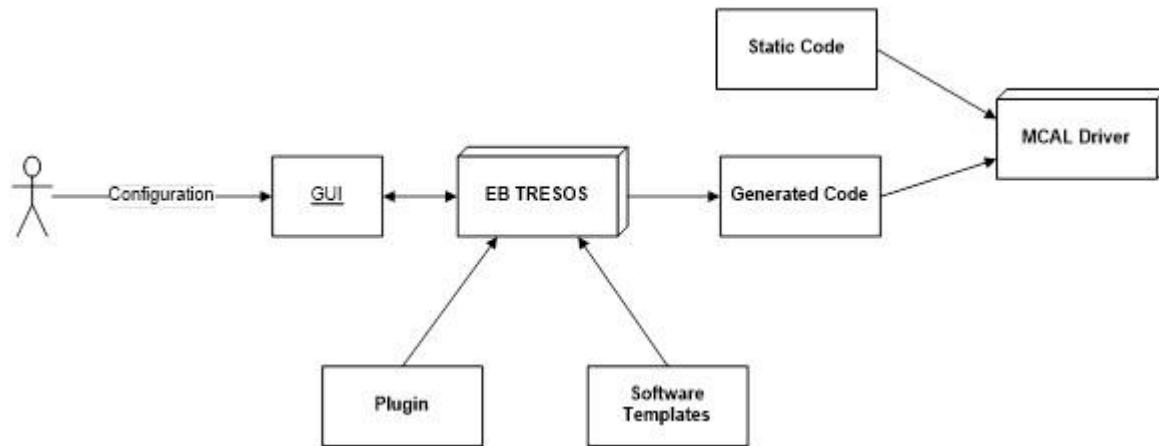
### 6.1 Naming conventions used/followed in source code files

General naming convention shall follow [AUTOSAR\\_SWS\\_BSWGeneral.pdf](#)

SW Module specific naming conventions shall follow driver specific AUTOSAR specification here: <https://www.autosar.org/standards/classic-platform/classic-platform-431/>

### 6.2 Configurator and Integration into MCAL

MCAL configuration, generation and integration follow this process:



- Plugin files are integrated into the AUTOSAR configuration tool (EB Tresos). These include the integration files and the XML configuration files (parameters and recommended configuration).
- User selects the desired configuration using a GUI provided by EB Tresos.
- Code generator takes the user selected configurations and the provided Software Templates and provides C code and header files based on the configuration.
- Generated code is integrated with static code containing the API definitions. This package is further integrated in the final system.



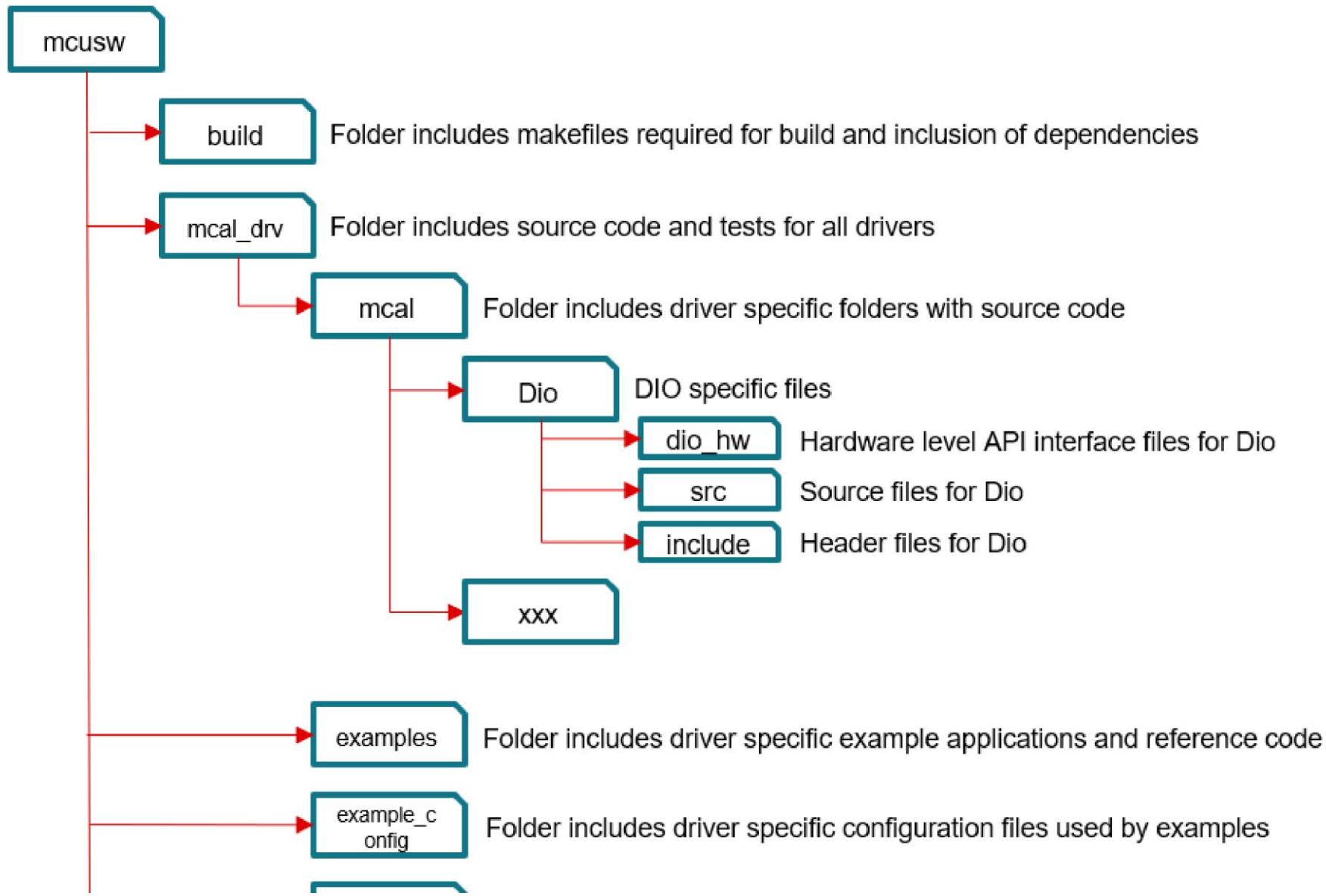
## 6.3 Directory Description

MCAL has the following directory structure. MCAL will be delivered as part MCUSW component in Processor SDK package.

### **MCUSW**

Includes below components:

```
|-- build  
|-- mcal_drv  
|-- mcuss_demos  
|-- mcusw_manifest.html  
|-- mcusw_release_notes.html  
|-- relnotes_archive
```

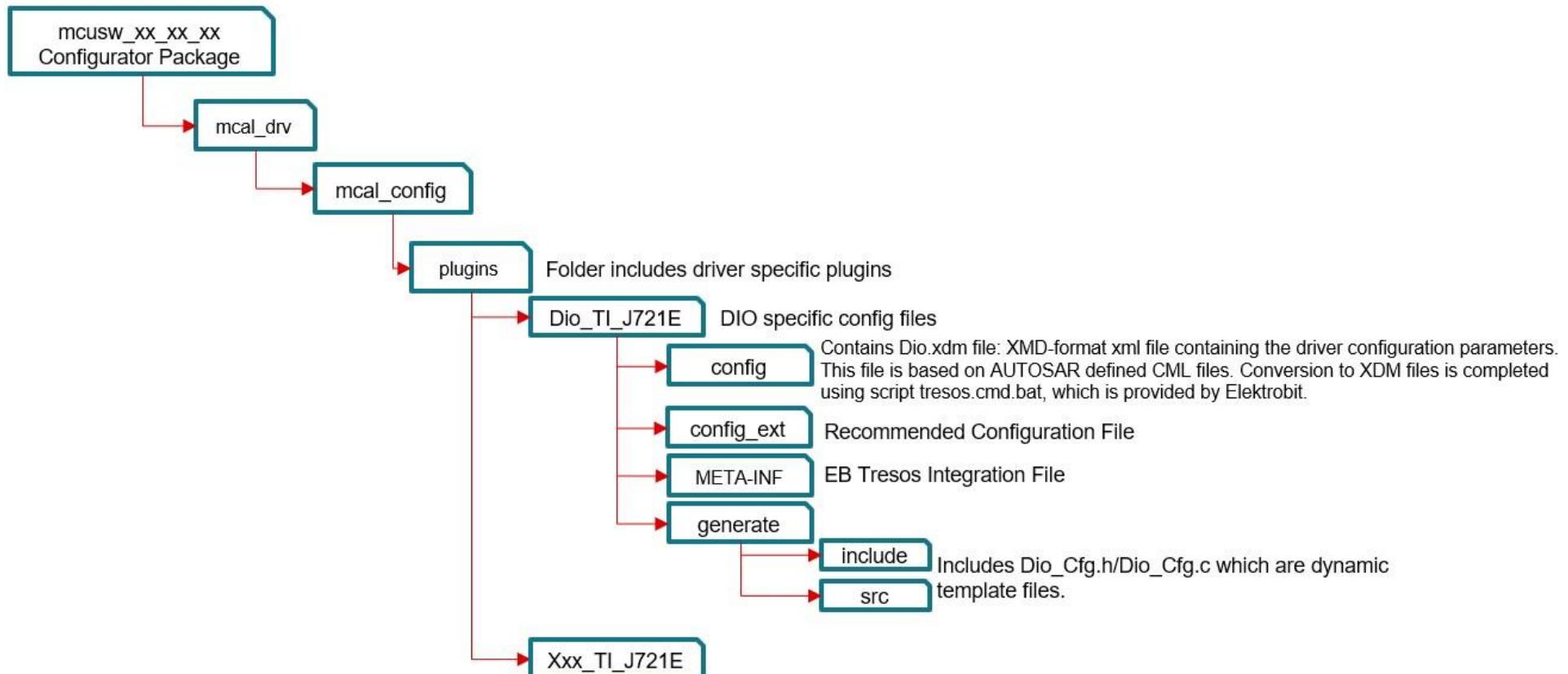




MCAL also includes folders such as scripts, and tool, which are not shown in above diagram. These folders are used to archive scripts and tools needed for builds and test. The top level MCAL also includes release notes, and build support files.

#### **Configurator Package**

The configurator package is licensed and will be accessible through secure software. The configurator structure is as follows:





## Documents

Documents such as user guides, API guides, validation reports are part of mcusw\_docs component in Processor SDK.

Additionally, a CSP package that can be used for requalification efforts will be provided through mySecureSW. It will include the following documents:

Ref: [Compliance Support Package \(CSP\)](#)

- Software FMEA
- Design Documents
- Test plan and reports
- Tractability data
- Dynamic code coverage analysis report
- Static code analysis report

The CSP will have individual drivers folder within, which will include the content mentioned above for the particular driver.

There will also be a folder named "Common" within the CSP. This will include below document which are common between the MCAL drivers.

- Architecture document
- Requirements

## 6.4 Data structures and resources

Architecture Identifier	Description
 MCAL-7678 - Resource <span style="background-color: #c8f7e4; border: 1px solid #2e6b2e; padding: 2px;">PUBLISHED</span>	<p>The MCAL modules shall not access any memory or any common peripheral/registers other than the module memory space. With the following exception</p> <p>TimeStamp Call : This function might be required to implement time bound waits. The modules shall use abstraction to determine time</p>

## 6.5 Event Overview

MCAL will have SW Module specific Event Overview and shall follow driver specific AUTOSAR specification.

## 6.6 Operating System Details

MCAL SW Modules will be bare-metal and will not depend on any OS.

## 6.7 Task Details

MCAL will not create any tasks.

## 6.8 Data Flow Diagram

MCAL will have SW Module specific Data Flow Diagram and shall follow AUTOSAR specification.



## 6.9 Control Flow Diagram

MCAL will have SW Module specific Control Flow Diagram and shall follow AUTOSAR specification.

## 6.10 Commercial off the shelf/reusable components used

None

## 6.11 Data storage and protection

The MCAL drivers' text segments could be placed in any memory region (such as Internal RAM (MCU RAM, MSMC), TCM, etc...)).

The driver implementation shall not impose any restrictions on placement of text segments (FLS driver is the only exception).

## 6.12 Resource Partition

### 6.12.1 Interrupt Map

MCAL driver demo applications require to register ISR to interrupts. MCUSW shall depend on SDK utilities for interrupt registration and display result/status on console via print utilities

### 6.12.2 Memory Assignment

MCAL is a library and memory placement of code and data is done by application. It is expected that the application places MCAL in RAM.

Architecture Identifier	Description
 MCAL-7696 - Memory Management: No Dynamic Alloc  <span data-bbox="186 684 309 708">PUBLISHED</span>	The MCAL shall not do any dynamic allocation of memory. All memory requirement should be statically allocated at compile time

### 6.12.3 Add other Resources

None

### 6.13 Debuggability/traceability/Visibility

The code shall be buildable in debug mode so that it may be stepped through using a debugger and a program such as Code Composer Studio (CCS).

For testing, display of results of the diagnostics test code will need access to UART port.

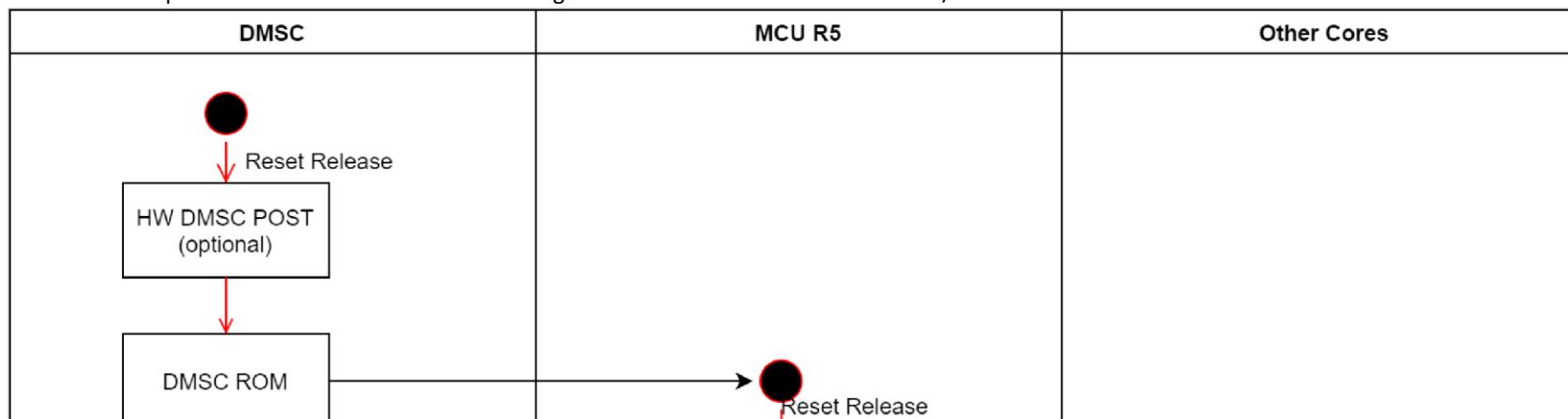
## 7 Dynamic Behavior

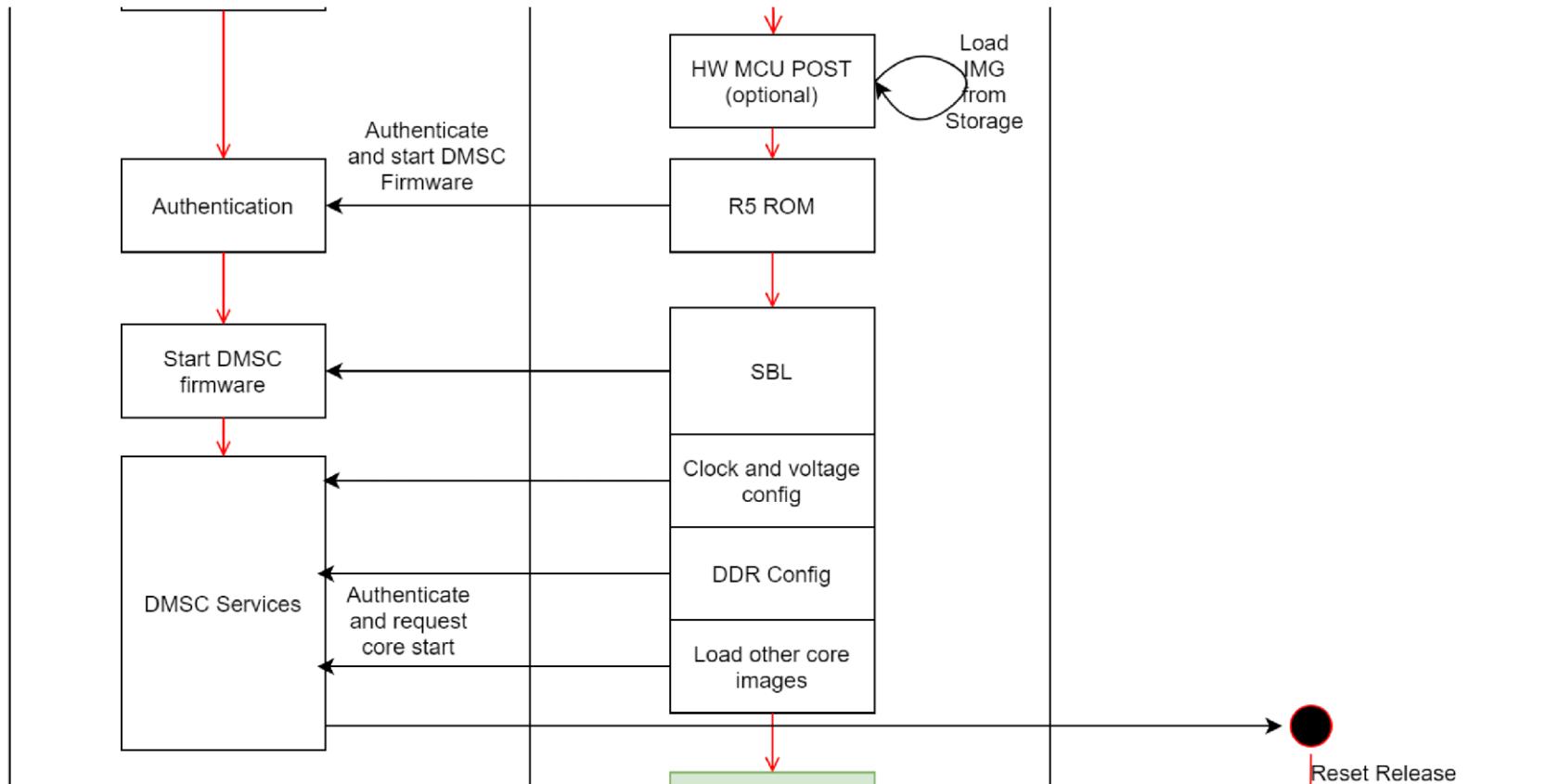
The SoC architecture has a separate processor that is the Device Management Security Controller (DMSC). The DMSC processor is the first to come up on boot-up and expected to bring the MCU core out of reset. As part of the power-up, the HW POST (Power-on Self Test) is executed for DMSC and/or MCU depending on device settings and version. Once the R5F is booted, it executes the Boot ROM and loads the Secondary Boot Loader (SBL). The SBL then performs device configurations, and then loads the applications to the cores (including loading the MCU R5F application to the same core).

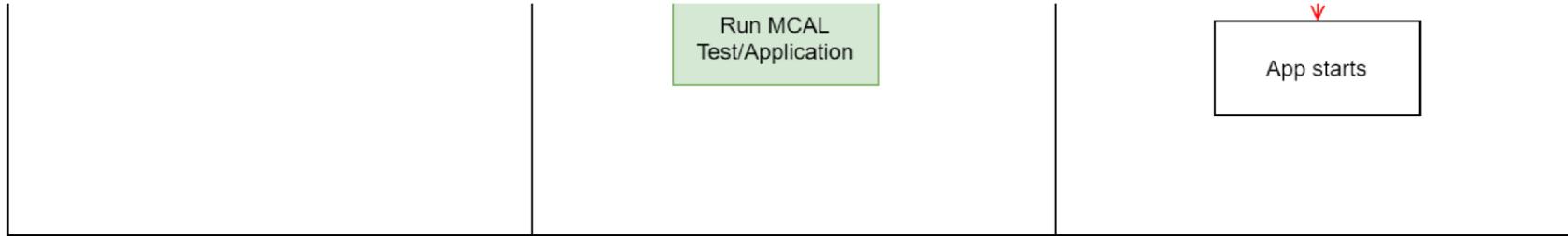
DMSC firmware → Boots up R5F → R5F runs the Boot ROM → R5F loads Secondary boot loader → Loads the actual application.

The following diagram shows the boot sequence. This sequence is the same for any application, including the application using the MCAL drivers. The green box shows the point where the Application which will execute the MCAL APIs begins. The rest of the diagram is to illustrate the sequence which is used to start the application.

Note: This sequence is used to demonstrate the usage of MCAL drivers. AUTOSAR vendors/customer will use it in AUTOSAR stack context.



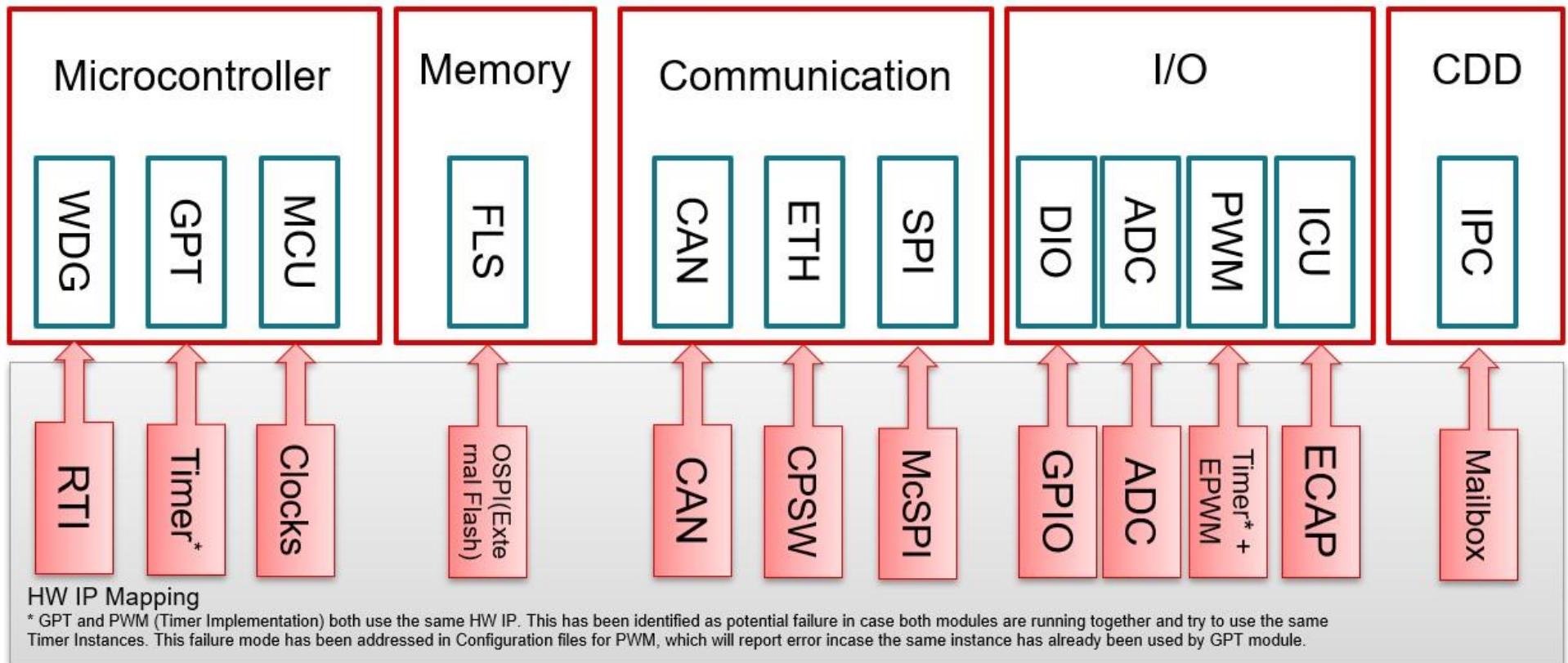






## 8 SW Modules

MCAL is comprised of several modules which use hardware IPs from TI SoC to accomplish required functionality (Refer to diagram below).





Note: The following sections are module specific. The "Specification Category" is derived from the AUTOSAR Spec for the particular module. It will specify the module dependencies, file structure, general behavior, error handling, type definitions, top-level API specification, and configuration variant requirement. An additional category is also added to address the safety diagnostic features.

Below table outlines specific HW IP usage per module:

	<b>SW Modules</b>	<b>HW IP</b>
1	ADC	adc12_16ffc_10_rel.1.0.x
2	DIO	gpio_144_10_rel.1.5.x
3	WDG	rti_10_rel.0.0.x
4	CAN	mcanss_10_rel.1.1.x
5	SPI	spi_10_rel.2.11.x
6	GPT	dmtimer_dmc1ms_10_rel.1.0.x
7	PWM	dmtimer_dmc1ms_10_rel.1.0.x

	<b>SW Modules</b>	<b>HW IP</b>
8	EPWM	ehrpwm_10_rel.1.3.x
9	ICU	ecap_10_rel.1.0.x
10	FLS	ospi_10_rel.1.0.x
11	CDD IPC	ksipc_mailbox_rel.1.0.x
12	MCU	NA

## 8.1 DIO Driver

The DIO module provides interfaces to external peripherals by abstracting the input and output pins on the microcontroller device as detailed in the (1) [AUTOSAR BSW DIO Driver Specification](#), and will make use of the GPIO internal peripheral.

The DIO Driver provides services for reading and writing to/from DIO Channels (Pins), DIO Ports and DIO Channel Groups. The behavior of those services is synchronous. This module works on pins and ports which should be configured by the application for this purpose. For this reason, there is no configuration and initialization of this port structure in the DIO Driver.

The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW DIO Driver Specification](#).



	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	Safety Process/ Diagnostic	MCAL-5634 - DIO Module shall include the Safety Diagnostic Features  <b>PUBLISHED</b>	Following the Safety manual GPIO IP and FMA review, the required safety diagnostic features should be added.
2	Dependencies on other modules	MCAL-5530 - [SWS_Dio_00 061] : Dio Initialization  <b>PUBLISHED</b>	[SWS_Dio_00061] The Dio module shall not provide APIs for overall configuration and initialization of the port structure which is used in the Dio module. These actions are done by the PORT Driver Module.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
3	Dependencies on other modules	 <a href="#">MCAL-5498 - [SWS_Dio_00102] : Dio Initialization</a> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	[SWS_Dio_00102] The Dio module's user shall only use the Dio functions after the Port Driver has been initialized. Otherwise the Dio module will exhibit undefined behavior.
4	File Structure	 <a href="#">MCAL-5491 - [SWS_Dio_00117] : Dio File structure</a> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	SWS_DIO_00117 Dio module shall follow the file structure defined by SWS_DIO_00117.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
5	General Behavior	 <a href="#">MCAL-5439 - [SWS_Dio_00_128] : Dio Channel Definition</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	[SWS_Dio_00128] A general-purpose digital IO pin represents a DIO channel.
6	General Behavior	 <a href="#">MCAL-5523 - [SWS_Dio_00_127] : Port Channel Configuration</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	[SWS_Dio_00127] The Port module shall configure a DIO channel as input or output [SWS_Dio_00001 and SWS_Dio_00002].

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
7	General Behavior	 <a href="#">MCAL-5494 - [SWS_Dio_00053] : Dio Port Definition</a> <span>PUBLISHED</span>	[SWS_Dio_00053] In the DIO Driver, it shall be possible to group several DIO channels by hardware (typically controlled by one hardware register) to represent a DIO port.
8	General Behavior	 <a href="#">MCAL-5488 - [SWS_Dio_00056] : Dio Channel Group Definition</a> <span>PUBLISHED</span>	[SWS_Dio_00056] A channel group is a formal logical combination of several adjoining DIO channels within a DIO port.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
9	General Behavior	 <a href="#">MCAL-5520 - [SWS_Dio_00060] : Dio Re-entrant Functions</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	[SWS_Dio_00060] All read and write functions of the Dio module shall be reentrant. Reason: The DIO Driver may be accessed by different upper layer handlers or drivers. These upper layer modules may access the driver concurrently.
10	General Behavior	 <a href="#">MCAL-5451 - [SWS_Dio_00026] : Dio Configuration Symbolic names</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	[SWS_Dio_00026] The configuration process for Dio module shall provide symbolic names for each configured DIO channel, port and group.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1 1	General Behavior	 <b>MCAL-5479 - [SWS_Dio_00113] : Dio Configuration Symbolic names</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	[SWS_Dio_00113] The Dio module shall publish the symbolic names which have been created during the configuration process in the file “Dio_Cfg.h”.
1 2	Initialization	 <b>MCAL-5536 - DIO Initialization</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	Initialization of the hardware will not be done by the DIO driver. MCAL does not support a PORT driver currently. Therefore, initialization of the hardware will need to be done by the application using SDK modules such as pinmux.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1 3	Dio Transfer Service	 <b>MCAL-5537 -</b> DIO Transfer Service <span style="background-color: #c8f7e4; padding: 2px;">PUBLISHED</span>	The DIO Driver provides services to transfer data to the microcontroller's pins, and shall follow the channel, port and channel group transfer requirements as specified in(1) <a href="#">AUTOSAR BSW DIO Driver Specification</a>
1 4	Error Classification and Detection	 <b>MCAL-5538 -</b> DIO shall provide development error check. <span style="background-color: #c8f7e4; padding: 2px;">PUBLISHED</span>	The Dio Driver shall provide development error checks and API parameter checking.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1 5	Error Classification and Detection	 <a href="#">MCAL-5485 - [SWS_Dio_00140] : Dio Error Handling</a> <span>PUBLISHED</span>	SWS_Dio_00140 The Dio Driver shall use Det_ReportError service to report development errors.
1 6	Type Definitions	 <a href="#">MCAL-5539 - DIO shall provide required data types.</a> <span>PUBLISHED</span>	The Dio Driver shall define required data types as specified in(1) <a href="#">AUTOSAR BSW DIO Driver Specification</a> . Dio_ChannelType, Dio_PortType, Dio_ChannelGroupType, Dio_LevelType, Dio_PortLevelType

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																														
1 7	Function Definitions	 <b>MCAL-5445 - [SWS_Dio_00133] : Dio Read Channel Service</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Dio_00133] Dio_ReadChannel API shall be implemented and follow <a href="#">AUTOSAR DIO 4.3.1 Spec.</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Dio_ReadChannel</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">Dio_LevelType Dio_ReadChannel(     Dio_ChannelType ChannelId )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x00</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ChannelId</td> <td>ID of DIO channel</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Dio_LevelType</td> <td>STD_HIGH The physical level of the corresponding Pin is STD_HIGH STD_LOW The physical level of the corresponding Pin is STD_LOW</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Returns the value of the specified DIO channel.</td> </tr> </table>	<b>Service name:</b>	Dio_ReadChannel		<b>Syntax:</b>	Dio_LevelType Dio_ReadChannel( Dio_ChannelType ChannelId )		<b>Service ID[hex]:</b>	0x00		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant		<b>Parameters (in):</b>	ChannelId	ID of DIO channel	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Dio_LevelType	STD_HIGH The physical level of the corresponding Pin is STD_HIGH STD_LOW The physical level of the corresponding Pin is STD_LOW	<b>Description:</b>	Returns the value of the specified DIO channel.	
<b>Service name:</b>	Dio_ReadChannel																																
<b>Syntax:</b>	Dio_LevelType Dio_ReadChannel( Dio_ChannelType ChannelId )																																
<b>Service ID[hex]:</b>	0x00																																
<b>Sync/Async:</b>	Synchronous																																
<b>Reentrancy:</b>	Reentrant																																
<b>Parameters (in):</b>	ChannelId	ID of DIO channel																															
<b>Parameters (inout):</b>	None																																
<b>Parameters (out):</b>	None																																
<b>Return value:</b>	Dio_LevelType	STD_HIGH The physical level of the corresponding Pin is STD_HIGH STD_LOW The physical level of the corresponding Pin is STD_LOW																															
<b>Description:</b>	Returns the value of the specified DIO channel.																																

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																																	
1 8	Function Definitions	 <b>MCAL-5463 - [SWS_Dio_00134] : Dio Write Channel Service</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Dio_00134] Dio_WriteChannel API shall be implemented and follow <a href="#">AUTOSAR DIO 4.3.1 Spec.</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Dio_WriteChannel</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">void Dio_WriteChannel(     Dio_ChannelType ChannelId,     Dio_LevelType Level )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x01</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ChannelId</td> <td>ID of DIO channel</td> </tr> <tr> <td></td> <td>Level</td> <td>Value to be written</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service to set a level of a channel.</td> </tr> </table>	<b>Service name:</b>	Dio_WriteChannel		<b>Syntax:</b>	void Dio_WriteChannel( Dio_ChannelType ChannelId, Dio_LevelType Level )		<b>Service ID[hex]:</b>	0x01		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant		<b>Parameters (in):</b>	ChannelId	ID of DIO channel		Level	Value to be written	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Service to set a level of a channel.	
<b>Service name:</b>	Dio_WriteChannel																																			
<b>Syntax:</b>	void Dio_WriteChannel( Dio_ChannelType ChannelId, Dio_LevelType Level )																																			
<b>Service ID[hex]:</b>	0x01																																			
<b>Sync/Async:</b>	Synchronous																																			
<b>Reentrancy:</b>	Reentrant																																			
<b>Parameters (in):</b>	ChannelId	ID of DIO channel																																		
	Level	Value to be written																																		
<b>Parameters (inout):</b>	None																																			
<b>Parameters (out):</b>	None																																			
<b>Return value:</b>	None																																			
<b>Description:</b>	Service to set a level of a channel.																																			

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																									
1 9	Function Definitions	 <b>MCAL-5506 - [SWS_Dio_00135] : Dio Read Port Service</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Dio_00135] Dio_ReadPort API shall be implemented and follow <a href="#">AUTOSAR DIO 4.3.1 Spec.</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Dio_ReadPort</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Dio_PortLevelType Dio_ReadPort(     Dio_PortType PortId )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x02</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>PortId</td> <td>ID of DIO Port</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Dio_PortLevelType</td> <td>Level of all channels of that port</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Returns the level of all channels of that port.</td> </tr> </table>	<b>Service name:</b>	Dio_ReadPort	<b>Syntax:</b>	Dio_PortLevelType Dio_ReadPort( Dio_PortType PortId )	<b>Service ID[hex]:</b>	0x02	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	PortId	ID of DIO Port	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Dio_PortLevelType	Level of all channels of that port	<b>Description:</b>	Returns the level of all channels of that port.	
<b>Service name:</b>	Dio_ReadPort																											
<b>Syntax:</b>	Dio_PortLevelType Dio_ReadPort( Dio_PortType PortId )																											
<b>Service ID[hex]:</b>	0x02																											
<b>Sync/Async:</b>	Synchronous																											
<b>Reentrancy:</b>	Reentrant																											
<b>Parameters (in):</b>	PortId	ID of DIO Port																										
<b>Parameters (inout):</b>	None																											
<b>Parameters (out):</b>	None																											
<b>Return value:</b>	Dio_PortLevelType	Level of all channels of that port																										
<b>Description:</b>	Returns the level of all channels of that port.																											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																																	
2 0	Function Definitions	 <b>MCAL-5466 - [SWS_Dio_00136] : Dio Write Port Service</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Dio_00136] Dio_WritePort API shall be implemented and follow <a href="#">AUTOSAR DIO 4.3.1 Spec.</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Dio_WritePort</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">void Dio_WritePort(     Dio_PortType PortId,     Dio_PortLevelType Level )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x03</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>PortId</td> <td>ID of DIO Port</td> </tr> <tr> <td></td> <td>Level</td> <td>Value to be written</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service to set a value of the port.</td> </tr> </table>	<b>Service name:</b>	Dio_WritePort		<b>Syntax:</b>	void Dio_WritePort( Dio_PortType PortId, Dio_PortLevelType Level )		<b>Service ID[hex]:</b>	0x03		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant		<b>Parameters (in):</b>	PortId	ID of DIO Port		Level	Value to be written	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Service to set a value of the port.	
<b>Service name:</b>	Dio_WritePort																																			
<b>Syntax:</b>	void Dio_WritePort( Dio_PortType PortId, Dio_PortLevelType Level )																																			
<b>Service ID[hex]:</b>	0x03																																			
<b>Sync/Async:</b>	Synchronous																																			
<b>Reentrancy:</b>	Reentrant																																			
<b>Parameters (in):</b>	PortId	ID of DIO Port																																		
	Level	Value to be written																																		
<b>Parameters (inout):</b>	None																																			
<b>Parameters (out):</b>	None																																			
<b>Return value:</b>	None																																			
<b>Description:</b>	Service to set a value of the port.																																			

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
2 1	Function Definitions	 <b>MCAL-5508 - [SWS_Dio_00137] : Dio Read Channel Group Service</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Dio_00137] Dio_ReadChannelGroup API shall be implemented and follow <a href="#">AUTOSAR DIO 4.3.1 Spec.</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Dio_ReadChannelGroup</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Dio_PortLevelType Dio_ReadChannelGroup(     const Dio_ChannelGroupType* ChannelGroupIdPtr )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x04</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ChannelGroupIdPtr      Pointer to ChannelGroup</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Dio_PortLevelType      Level of a subset of the adjoining bits of a port</td> </tr> <tr> <td><b>Description:</b></td> <td>This Service reads a subset of the adjoining bits of a port.</td> </tr> </table>	<b>Service name:</b>	Dio_ReadChannelGroup	<b>Syntax:</b>	Dio_PortLevelType Dio_ReadChannelGroup( const Dio_ChannelGroupType* ChannelGroupIdPtr )	<b>Service ID[hex]:</b>	0x04	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	ChannelGroupIdPtr      Pointer to ChannelGroup	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Dio_PortLevelType      Level of a subset of the adjoining bits of a port	<b>Description:</b>	This Service reads a subset of the adjoining bits of a port.
<b>Service name:</b>	Dio_ReadChannelGroup																						
<b>Syntax:</b>	Dio_PortLevelType Dio_ReadChannelGroup( const Dio_ChannelGroupType* ChannelGroupIdPtr )																						
<b>Service ID[hex]:</b>	0x04																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	ChannelGroupIdPtr      Pointer to ChannelGroup																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	Dio_PortLevelType      Level of a subset of the adjoining bits of a port																						
<b>Description:</b>	This Service reads a subset of the adjoining bits of a port.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																																	
2	Function Definitions	 <b>MCAL-5462 - [SWS_Dio_00138] : Dio Write Channel Group Service</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Dio_00138] Dio_WriteChannelGroup API shall be implemented and follow <a href="#">AUTOSAR DIO 4.3.1 Spec.</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Dio_WriteChannelGroup</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2"> <pre>void Dio_WriteChannelGroup(     const Dio_ChannelGroupType* ChannelGroupIdPtr,     Dio_PortLevelType Level )</pre> </td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x05</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ChannelGroupIdPtr</td> <td>Pointer to ChannelGroup</td> </tr> <tr> <td></td> <td>Level</td> <td>Value to be written</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service to set a subset of the adjoining bits of a port to a specified level.</td> </tr> </table>	<b>Service name:</b>	Dio_WriteChannelGroup		<b>Syntax:</b>	<pre>void Dio_WriteChannelGroup(     const Dio_ChannelGroupType* ChannelGroupIdPtr,     Dio_PortLevelType Level )</pre>		<b>Service ID[hex]:</b>	0x05		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant		<b>Parameters (in):</b>	ChannelGroupIdPtr	Pointer to ChannelGroup		Level	Value to be written	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Service to set a subset of the adjoining bits of a port to a specified level.	
<b>Service name:</b>	Dio_WriteChannelGroup																																			
<b>Syntax:</b>	<pre>void Dio_WriteChannelGroup(     const Dio_ChannelGroupType* ChannelGroupIdPtr,     Dio_PortLevelType Level )</pre>																																			
<b>Service ID[hex]:</b>	0x05																																			
<b>Sync/Async:</b>	Synchronous																																			
<b>Reentrancy:</b>	Reentrant																																			
<b>Parameters (in):</b>	ChannelGroupIdPtr	Pointer to ChannelGroup																																		
	Level	Value to be written																																		
<b>Parameters (inout):</b>	None																																			
<b>Parameters (out):</b>	None																																			
<b>Return value:</b>	None																																			
<b>Description:</b>	Service to set a subset of the adjoining bits of a port to a specified level.																																			

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
2 3	Function Definitions	 <b>MCAL-5477 - [SWS_Dio_00139] : Dio Get Version Info Service</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Dio_00139] Dio_GetVersionInfo API shall be implemented and follow <a href="#">AUTOSAR DIO 4.3.1 Spec.</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Dio_GetVersionInfo</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Dio_GetVersionInfo(     Std_VersionInfoType* VersionInfo )</td> </tr> <tr> <td><b>Service ID/Hex:</b></td> <td>0x12</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>VersionInfo   Pointer to where to store the version information of this module.</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Service to get the version information of this module.</td> </tr> </table>	<b>Service name:</b>	Dio_GetVersionInfo	<b>Syntax:</b>	void Dio_GetVersionInfo( Std_VersionInfoType* VersionInfo )	<b>Service ID/Hex:</b>	0x12	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	VersionInfo   Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	Service to get the version information of this module.
<b>Service name:</b>	Dio_GetVersionInfo																						
<b>Syntax:</b>	void Dio_GetVersionInfo( Std_VersionInfoType* VersionInfo )																						
<b>Service ID/Hex:</b>	0x12																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	VersionInfo   Pointer to where to store the version information of this module.																						
<b>Return value:</b>	None																						
<b>Description:</b>	Service to get the version information of this module.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																						
2 4	Function Definitions	 <b>MCAL-5450 - [SWS_Dio_00190] : Dio Flip Channel Service</b> <span style="background-color: #90EE90; color: black; padding: 2px;">PUBLISHED</span>	<p>[SWS_Dio_00190] Dio_FlipChannel API shall be implemented and follow <a href="#">AUTOSAR DIO 4.3.1 Spec.</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Dio_FlipChannel</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Dio_LevelType Dio_FlipChannel(     Dio_ChannelType ChannelId</td> </tr> <tr> <td></td> <td>)</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x11</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ChannelId      ID of DIO channel</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Dio_LevelType      STD_HIGH: The physical level of the corresponding Pin is STD_HIGH.                          STD_LOW: The physical level of the corresponding Pin is STD_LOW.</td> </tr> <tr> <td><b>Description:</b></td> <td>Service to flip (change from 1 to 0 or from 0 to 1) the level of a channel and return the level of the channel after flip.</td> </tr> </table>	<b>Service name:</b>	Dio_FlipChannel	<b>Syntax:</b>	Dio_LevelType Dio_FlipChannel( Dio_ChannelType ChannelId		)	<b>Service ID[hex]:</b>	0x11	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	ChannelId      ID of DIO channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Dio_LevelType      STD_HIGH: The physical level of the corresponding Pin is STD_HIGH. STD_LOW: The physical level of the corresponding Pin is STD_LOW.	<b>Description:</b>	Service to flip (change from 1 to 0 or from 0 to 1) the level of a channel and return the level of the channel after flip.
<b>Service name:</b>	Dio_FlipChannel																								
<b>Syntax:</b>	Dio_LevelType Dio_FlipChannel( Dio_ChannelType ChannelId																								
	)																								
<b>Service ID[hex]:</b>	0x11																								
<b>Sync/Async:</b>	Synchronous																								
<b>Reentrancy:</b>	Reentrant																								
<b>Parameters (in):</b>	ChannelId      ID of DIO channel																								
<b>Parameters (inout):</b>	None																								
<b>Parameters (out):</b>	None																								
<b>Return value:</b>	Dio_LevelType      STD_HIGH: The physical level of the corresponding Pin is STD_HIGH. STD_LOW: The physical level of the corresponding Pin is STD_LOW.																								
<b>Description:</b>	Service to flip (change from 1 to 0 or from 0 to 1) the level of a channel and return the level of the channel after flip.																								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
2 5	Configuration Specification	 <a href="#">MCAL-5453 - [SWS_Dio_00129] : Dio Configuration specification</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	[SWS_Dio_00129] At least one of the following configuration variants has to be supported by implementation: - VARIANT-PRE-COMPIL - VARIANT-LINK-TIME

## 8.2 ADC Driver

The ADC module initializes and controls the internal Analogue Digital Converter Unit(s) of the microcontroller. It provides services to start and stop a conversion respectively to enable and disable the trigger source for a conversion. Furthermore it provides services to enable and disable a notification mechanism and routines to query the status and result of a conversion. The ADC module works on so called ADC Channel Groups, which are build from so called ADC Channels. An ADC Channel Group combines an analogue input pin (ADC Channel), the needed ADC circuitry itself and conversion result register into an entity that can be individually controlled and accessed via the ADC module.

The ADC module will be developed as detailed in the (2) [AUTOSAR BSW ADC Driver Specification](#), and will make use of the ADC internal peripheral.

The SWS numbers in below table are in reference to the specification described in (2) [AUTOSAR BSW ADC Driver Specification](#).



	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	Safety Process/ Diagnostic	 <a href="#">MCAL-5895</a> - ADC Module shall include the Safety Diagnostic Features <span>PUBLISHED</span>	Following the Safety manual ADC IP and FMA review, the required safety diagnostic features should be added.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
2	File Structure	 <a href="#">MCAL-5820 - SWS_Adc_00</a> 280 : ADC Module Concurrent Conversion Support <span>PUBLISHED</span>	[SWS_Adc_00267] ADC module shall follow the file structure defined by [SWS_Adc_00267].
3	General Behavior	 <a href="#">MCAL-5798 - SWS_Adc_00</a> 090 : ADC Channel group <span>PUBLISHED</span>	[SWS_Adc_00090] The ADC module shall allow grouping of one or more ADC channels into so called ADC Channel groups.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
4	General Behavior	 <a href="#">MCAL-5832 - SWS_Adc_00</a> 091 : ADC Channel group atleast one channel <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	[SWS_Adc_00091] The ADC module's configuration shall be such that an ADC Channel group contains at least one ADC Channel.
5	General Behavior	 <a href="#">MCAL-5744 - SWS_Adc_00</a> 092 : ADC Channel assignment <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	[SWS_Adc_00092] The ADC module shall allow the assignment of an ADC channel to more than one group.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
6	General Behavior	 <a href="#">MCAL-5706 - SWS_Adc_00</a> 277 : ADC Channel group to ADC HW unit assignment <span>PUBLISHED</span>	[SWS_Adc_00277] The ADC module's configuration shall be such that all channels contained in one ADC Channel group shall belong to the same ADC HW Unit.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
7	General Behavior	 <a href="#">MCAL-5734 - SWS_Adc_00</a> 279 : ADC Channel group trigger source configuration <b>PUBLISHED</b>	[SWS_Adc_00279] The ADC module shall allow configuring exactly one trigger source for each ADC Channel group.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
8	General Behavior	 <a href="#">MCAL-5699 - SWS_Adc_00</a> 140 : ADC Conversion output consistency <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	[SWS_Adc_00140] The ADC module shall guarantee the consistency of the returned result value for each completed conversion.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
9	General Behavior	 <a href="#">MCAL-5820 - SWS_Adc_00</a> 280 : ADC Module Concurrent Conversion Support <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	[SWS_Adc_00280] The ADC module shall convert only one ADC Channel group per ADC HW Unit at a time. The ADC module shall not support the concurrent conversion of different (even exclusive) ADC Channel groups on the same ADC HW Unit.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1 0	Error Classification and Detection	 <a href="#">MCAL-5885 - SWS_Adc_00377 : ADC</a> Optional Interfaces <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>   <a href="#">MCAL-5724 - SWS_Adc_00530: Adc</a> Det_ReportRuntimeError <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>The ADC Driver shall provide runtime and development error checks and API parameter checking.</p> <p>SWS_Adc_00530 - ADC Driver shall use the Det_ReportRuntimeError service to report runtime errors.</p> <p>SWS_Adc_00377- ADC Driver shall use the Det_ReportError service to report development errors.</p>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1 1	Type Definitions	 <b>MCAL-5896 -</b> ADC Module shall define required data types as per Autosar Spec <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	The ADC Driver shall define required data types as specified in(2) <a href="#">AUTOSAR BSW ADC Driver Specification</a> section 8.2.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																														
1 2	Function Definitions	 <b>MCAL-5821 - SWS_Adc_00365 : Adc_Init API</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Adc_00365] Adc_Init API shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"><b>Service name:</b></td><td colspan="2">Adc_Init</td></tr> <tr> <td><b>Syntax:</b></td><td colspan="2">void Adc_Init(                   const Adc_ConfigType* ConfigPtr                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td colspan="2">0x00</td></tr> <tr> <td><b>Sync/Async:</b></td><td colspan="2">Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td colspan="2">Non Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>ConfigPtr</td><td>Pointer to configuration set in Variant PB (Variant PC requires a NULL_PTR).</td></tr> <tr> <td><b>Parameters (inout):</b></td><td colspan="2">None</td></tr> <tr> <td><b>Parameters (out):</b></td><td colspan="2">None</td></tr> <tr> <td><b>Return value:</b></td><td colspan="2">None</td></tr> <tr> <td><b>Description:</b></td><td colspan="2">Initializes the ADC hardware units and driver.</td></tr> </table>	<b>Service name:</b>	Adc_Init		<b>Syntax:</b>	void Adc_Init( const Adc_ConfigType* ConfigPtr )		<b>Service ID[hex]:</b>	0x00		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Non Reentrant		<b>Parameters (in):</b>	ConfigPtr	Pointer to configuration set in Variant PB (Variant PC requires a NULL_PTR).	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Initializes the ADC hardware units and driver.	
<b>Service name:</b>	Adc_Init																																
<b>Syntax:</b>	void Adc_Init( const Adc_ConfigType* ConfigPtr )																																
<b>Service ID[hex]:</b>	0x00																																
<b>Sync/Async:</b>	Synchronous																																
<b>Reentrancy:</b>	Non Reentrant																																
<b>Parameters (in):</b>	ConfigPtr	Pointer to configuration set in Variant PB (Variant PC requires a NULL_PTR).																															
<b>Parameters (inout):</b>	None																																
<b>Parameters (out):</b>	None																																
<b>Return value:</b>	None																																
<b>Description:</b>	Initializes the ADC hardware units and driver.																																

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1 3	Function Definitions	 <a href="#">MCAL-5779 - SWS_Adc_00419 : Adc_SetupResultsBuffer API</a> <span>PUBLISHED</span>	[SWS_Adc_00419] Adc_SetupResultsBuffer API shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a> .

	Specification Category	Architecture Identifier	Description								
			<p><b>Service name:</b> Adc_SetupResultBuffer</p> <p><b>Syntax:</b></p> <pre>Std_ReturnType Adc_SetupResultBuffer(     Adc_GroupType Group,     const Adc_ValueGroupType* DataBufferPtr )</pre> <p><b>Service ID/hex:</b> 0x0c</p> <p><b>Sync/Async:</b> Synchronous</p> <p><b>Reentrancy:</b> Reentrant</p> <table border="1"> <tr> <td><b>Parameters (in):</b></td> <td>Group</td> <td>Numeric ID of requested ADC channel group.</td> </tr> <tr> <td></td> <td>DataBufferPtr</td> <td>pointer to result data buffer</td> </tr> </table> <p><b>Parameters (inout):</b> None</p> <p><b>Parameters (out):</b> None</p> <p><b>Return value:</b></p> <table border="1"> <tr> <td>Std_ReturnType</td> <td>E_OK: result buffer pointer initialized correctly E_NOT_OK: operation failed or development error occurred</td> </tr> </table> <p><b>Description:</b> Initializes ADC driver with the group specific result buffer start address where the conversion results will be stored. The application has to ensure that the application buffer, where DataBufferPtr points to, can hold all the conversion results of the specified group. The initialization with Adc_SetupResultBuffer is required after reset, before a group conversion can be started.</p>	<b>Parameters (in):</b>	Group	Numeric ID of requested ADC channel group.		DataBufferPtr	pointer to result data buffer	Std_ReturnType	E_OK: result buffer pointer initialized correctly E_NOT_OK: operation failed or development error occurred
<b>Parameters (in):</b>	Group	Numeric ID of requested ADC channel group.									
	DataBufferPtr	pointer to result data buffer									
Std_ReturnType	E_OK: result buffer pointer initialized correctly E_NOT_OK: operation failed or development error occurred										

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
1 4	Function Definitions	 <b>MCAL-5770 - SWS_Adc_00366 :</b> Adc_Delnit API <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Adc_00366] shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Adc_Delnit</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Adc_DeInit(             void             )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x01</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Returns all ADC HW Units to a state comparable to their power on reset state.</td> </tr> </table>	<b>Service name:</b>	Adc_Delnit	<b>Syntax:</b>	void Adc_DeInit( void )	<b>Service ID[hex]:</b>	0x01	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Returns all ADC HW Units to a state comparable to their power on reset state.
<b>Service name:</b>	Adc_Delnit																						
<b>Syntax:</b>	void Adc_DeInit( void )																						
<b>Service ID[hex]:</b>	0x01																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Returns all ADC HW Units to a state comparable to their power on reset state.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
1 5	Function Definitions	 <b>MCAL-5839 - SWS_Adc_00367 : Adc_StartGroupConversion API</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Adc_00367] shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"><b>Service name:</b></td><td>Adc_StartGroupConversion</td></tr> <tr> <td><b>Syntax:</b></td><td>void Adc_StartGroupConversion(                           Adc_GroupType Group                           )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x02</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Asynchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Group      Numeric ID of requested ADC Channel group.</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Starts the conversion of all channels of the requested ADC Channel group.</td></tr> </table>	<b>Service name:</b>	Adc_StartGroupConversion	<b>Syntax:</b>	void Adc_StartGroupConversion( Adc_GroupType Group )	<b>Service ID[hex]:</b>	0x02	<b>Sync/Async:</b>	Asynchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Group      Numeric ID of requested ADC Channel group.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Starts the conversion of all channels of the requested ADC Channel group.
<b>Service name:</b>	Adc_StartGroupConversion																						
<b>Syntax:</b>	void Adc_StartGroupConversion( Adc_GroupType Group )																						
<b>Service ID[hex]:</b>	0x02																						
<b>Sync/Async:</b>	Asynchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	Group      Numeric ID of requested ADC Channel group.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Starts the conversion of all channels of the requested ADC Channel group.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
1 6	Function Definitions	 MCAL-5782 - SWS_Adc_00 368 : Adc_StopGroupConversion API  <b>PUBLISHED</b>	<p>[[SWS_Adc_00368]] shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td> <td>Adc_StopGroupConversion</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Adc_StopGroupConversion(                   Adc_GroupType Group                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x03</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Group      Numeric ID of requested ADC Channel group.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Stops the conversion of the requested ADC Channel group.</td> </tr> </table>	<b>Service name:</b>	Adc_StopGroupConversion	<b>Syntax:</b>	void Adc_StopGroupConversion( Adc_GroupType Group )	<b>Service ID[hex]:</b>	0x03	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Group      Numeric ID of requested ADC Channel group.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Stops the conversion of the requested ADC Channel group.
<b>Service name:</b>	Adc_StopGroupConversion																						
<b>Syntax:</b>	void Adc_StopGroupConversion( Adc_GroupType Group )																						
<b>Service ID[hex]:</b>	0x03																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	Group      Numeric ID of requested ADC Channel group.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Stops the conversion of the requested ADC Channel group.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
1 7	Function Definitions	 <b>MCAL-5799 - SWS_Adc_00369 : Adc_ReadGroup API</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[[SWS_Adc_00369]] shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"><b>Service name:</b></td><td colspan="2">Adc_ReadGroup</td></tr> <tr> <td><b>Syntax:</b></td><td colspan="2">Std_ReturnType Adc_ReadGroup(     Adc_GroupType Group,     Adc_ValueGroupType* DataBufferPtr )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td colspan="2">0x04</td></tr> <tr> <td><b>Sync/Async:</b></td><td colspan="2">Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td colspan="2">Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Group</td><td>Numeric ID of requested ADC channel group.</td></tr> </table>	<b>Service name:</b>	Adc_ReadGroup		<b>Syntax:</b>	Std_ReturnType Adc_ReadGroup( Adc_GroupType Group, Adc_ValueGroupType* DataBufferPtr )		<b>Service ID[hex]:</b>	0x04		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant		<b>Parameters (in):</b>	Group	Numeric ID of requested ADC channel group.
<b>Service name:</b>	Adc_ReadGroup																				
<b>Syntax:</b>	Std_ReturnType Adc_ReadGroup( Adc_GroupType Group, Adc_ValueGroupType* DataBufferPtr )																				
<b>Service ID[hex]:</b>	0x04																				
<b>Sync/Async:</b>	Synchronous																				
<b>Reentrancy:</b>	Reentrant																				
<b>Parameters (in):</b>	Group	Numeric ID of requested ADC channel group.																			

	Specification Category	Architecture Identifier	Description			
			<b>Parameters (inout):</b>	None		
			<b>Parameters (out):</b>	DataBufferPtr	ADC results of all channels of the selected group are stored in the data buffer addressed with the pointer.	
			<b>Return value:</b>	Std_ReturnType	E_OK: results are available and written to the data buffer E_NOT_OK: no results are available or development error occurred	
			<b>Description:</b>	Reads the group conversion result of the last completed conversion round of the requested group and stores the channel values starting at the DataBufferPtr address. The group channel values are stored in ascending channel number order ( in contrast to the storage layout of the result buffer if streaming access is configured).		

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
1 8	Function Definitions	 <b>MCAL-5876 - SWS_Adc_00372 : Adc_EnableGroupNotification API</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[[SWS_Adc_00372]] shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td><td style="padding: 2px;">Adc_EnableGroupNotification</td></tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td><td style="padding: 2px;">void Adc_EnableGroupNotification(                           Adc_GroupType Group                           )</td></tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td><td style="padding: 2px;">0x07</td></tr> <tr> <td style="padding: 2px;"><b>Sync/Async:</b></td><td style="padding: 2px;">Synchronous</td></tr> <tr> <td style="padding: 2px;"><b>Reentrancy:</b></td><td style="padding: 2px;">Reentrant</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (in):</b></td><td style="padding: 2px;">Group      Numeric ID of requested ADC Channel group.</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (inout):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (out):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Return value:</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Description:</b></td><td style="padding: 2px;">Enables the notification mechanism for the requested ADC Channel group.</td></tr> </table>	<b>Service name:</b>	Adc_EnableGroupNotification	<b>Syntax:</b>	void Adc_EnableGroupNotification( Adc_GroupType Group )	<b>Service ID[hex]:</b>	0x07	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Group      Numeric ID of requested ADC Channel group.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Enables the notification mechanism for the requested ADC Channel group.
<b>Service name:</b>	Adc_EnableGroupNotification																						
<b>Syntax:</b>	void Adc_EnableGroupNotification( Adc_GroupType Group )																						
<b>Service ID[hex]:</b>	0x07																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	Group      Numeric ID of requested ADC Channel group.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Enables the notification mechanism for the requested ADC Channel group.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																														
1 9	Function Definitions	 <b>MCAL-5740 - SWS_Adc_00373 : Adc_DisableGroupNotification API</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[[SWS_Adc_00373] shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Adc_DisableGroupNotification</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">void Adc_DisableGroupNotification(                           Adc_GroupType Group                           )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x08</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Group</td> <td>Numeric ID of requested ADC Channel group.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Disables the notification mechanism for the requested ADC Channel group.</td> </tr> </table>	<b>Service name:</b>	Adc_DisableGroupNotification		<b>Syntax:</b>	void Adc_DisableGroupNotification( Adc_GroupType Group )		<b>Service ID[hex]:</b>	0x08		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant		<b>Parameters (in):</b>	Group	Numeric ID of requested ADC Channel group.	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Disables the notification mechanism for the requested ADC Channel group.	
<b>Service name:</b>	Adc_DisableGroupNotification																																
<b>Syntax:</b>	void Adc_DisableGroupNotification( Adc_GroupType Group )																																
<b>Service ID[hex]:</b>	0x08																																
<b>Sync/Async:</b>	Synchronous																																
<b>Reentrancy:</b>	Reentrant																																
<b>Parameters (in):</b>	Group	Numeric ID of requested ADC Channel group.																															
<b>Parameters (inout):</b>	None																																
<b>Parameters (out):</b>	None																																
<b>Return value:</b>	None																																
<b>Description:</b>	Disables the notification mechanism for the requested ADC Channel group.																																

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																														
2 0	Function Definitions	 <b>MCAL-5849 - SWS_Adc_00374 : Adc_GroupStatus API</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[[SWS_Adc_00374]] shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Adc_GetGroupStatus</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">Adc_StatusType Adc_GetGroupStatus(                           Adc_GroupType Group                           )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x09</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Group</td> <td>Numeric ID of requested ADC Channel group.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Adc_StatusType</td> <td>Conversion status for the requested group.</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Returns the conversion status of the requested ADC Channel group.</td> </tr> </table>	<b>Service name:</b>	Adc_GetGroupStatus		<b>Syntax:</b>	Adc_StatusType Adc_GetGroupStatus( Adc_GroupType Group )		<b>Service ID[hex]:</b>	0x09		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant		<b>Parameters (in):</b>	Group	Numeric ID of requested ADC Channel group.	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Adc_StatusType	Conversion status for the requested group.	<b>Description:</b>	Returns the conversion status of the requested ADC Channel group.	
<b>Service name:</b>	Adc_GetGroupStatus																																
<b>Syntax:</b>	Adc_StatusType Adc_GetGroupStatus( Adc_GroupType Group )																																
<b>Service ID[hex]:</b>	0x09																																
<b>Sync/Async:</b>	Synchronous																																
<b>Reentrancy:</b>	Reentrant																																
<b>Parameters (in):</b>	Group	Numeric ID of requested ADC Channel group.																															
<b>Parameters (inout):</b>	None																																
<b>Parameters (out):</b>	None																																
<b>Return value:</b>	Adc_StatusType	Conversion status for the requested group.																															
<b>Description:</b>	Returns the conversion status of the requested ADC Channel group.																																

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
2 1	Function Definitions	 <b>MCAL-5844 - SWS_Adc_00</b> 375 : <b>Adc_GetStreamLastPointer API</b> <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	[[SWS_Adc_00375] shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a> .

	Specification Category	Architecture Identifier	Description
			<p><b>Service name:</b> Adc_GetStreamLastPointer</p> <p><b>Syntax:</b> Adc_StreamNumSampleType Adc_GetStreamLastPointer(                     Adc_GroupType Group,                     Adc_ValueGroupType** PtrToSamplePtr                     )</p> <p><b>Service ID[hex]:</b> 0x0b</p> <p><b>Sync/Async:</b> Synchronous</p> <p><b>Reentrancy:</b> Reentrant</p> <p><b>Parameters (in):</b> Group    Numeric ID of requested ADC Channel group.</p> <p><b>Parameters (inout):</b> None</p> <p><b>Parameters (out):</b> PtrToSamplePtr                                  Pointer to result buffer pointer.</p> <p><b>Return value:</b> Adc_StreamNumSampleType                          Number of valid samples per channel.</p> <p><b>Description:</b> Returns the number of valid samples per channel, stored in the result buffer. Reads a pointer, pointing to a position in the group result buffer. With the pointer position, the results of all group channels of the last completed conversion round can be accessed. With the pointer and the return value, all valid group conversion results can be accessed (the user has to take the layout of the result buffer into account).</p>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																						
2	Function Definitions	 <b>MCAL-5880 - SWS_Adc_00376 : Adc_GetVersionInfo API</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[[SWS_Adc_00376] shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"><b>Service name:</b></td><td>Adc_GetVersionInfo</td></tr> <tr> <td><b>Syntax:</b></td><td>void Adc_GetVersionInfo(                   Std_VersionInfoType* versioninfo                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x0a</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>None</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>versioninfo</td><td>Pointer to where to store the version information of this module.</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td colspan="2">Returns the version information of this module.</td></tr> </table>	<b>Service name:</b>	Adc_GetVersionInfo	<b>Syntax:</b>	void Adc_GetVersionInfo( Std_VersionInfoType* versioninfo )	<b>Service ID[hex]:</b>	0x0a	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	versioninfo	Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	Returns the version information of this module.	
<b>Service name:</b>	Adc_GetVersionInfo																								
<b>Syntax:</b>	void Adc_GetVersionInfo( Std_VersionInfoType* versioninfo )																								
<b>Service ID[hex]:</b>	0x0a																								
<b>Sync/Async:</b>	Synchronous																								
<b>Reentrancy:</b>	Reentrant																								
<b>Parameters (in):</b>	None																								
<b>Parameters (inout):</b>	None																								
<b>Parameters (out):</b>	versioninfo	Pointer to where to store the version information of this module.																							
<b>Return value:</b>	None																								
<b>Description:</b>	Returns the version information of this module.																								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
2 3	Configurable interfaces	 <b>MCAL-5762 - SWS_Adc_00078 : ADC Configurable interfaces</b> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Adc_00078] shall be implemented and follow <a href="#">AUTOSAR BSW ADC Driver Specification</a>.</p> <p>The ADC module's ISR's, providing the "conversion completed events", shall be responsible for resetting the interrupt flags (if needed by hardware) and calling the associated notification function.</p>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
2 4	Configurable interfaces	 <b>MCAL-5774 - SWS_Adc_00 082 : IoHwAb_Adc Notification</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Adc_00082 shall be implemented and follow AUTOSAR BSW ADC Driver Specification.]</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td><td style="padding: 2px;">IoHwAb_AdcNotification&lt;#groupID&gt;</td></tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td><td style="padding: 2px;">void IoHwAb_AdcNotification&lt;#groupID&gt;(                   void                   )                   </td></tr> <tr> <td style="padding: 2px;"><b>Sync/Async:</b></td><td style="padding: 2px;">Synchronous</td></tr> <tr> <td style="padding: 2px;"><b>Reentrancy:</b></td><td style="padding: 2px;">Non Reentrant</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (in):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (inout):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (out):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Return value:</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Description:</b></td><td style="padding: 2px;">Will be called by the ADC Driver when a group conversion is completed for group &lt;#groupID&gt;.</td></tr> </table>	<b>Service name:</b>	IoHwAb_AdcNotification<#groupID>	<b>Syntax:</b>	void IoHwAb_AdcNotification<#groupID>( void ) 	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Will be called by the ADC Driver when a group conversion is completed for group <#groupID>.
<b>Service name:</b>	IoHwAb_AdcNotification<#groupID>																				
<b>Syntax:</b>	void IoHwAb_AdcNotification<#groupID>( void ) 																				
<b>Sync/Async:</b>	Synchronous																				
<b>Reentrancy:</b>	Non Reentrant																				
<b>Parameters (in):</b>	None																				
<b>Parameters (inout):</b>	None																				
<b>Parameters (out):</b>	None																				
<b>Return value:</b>	None																				
<b>Description:</b>	Will be called by the ADC Driver when a group conversion is completed for group <#groupID>.																				

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
2 5	Sequence Diagrams	 <a href="#">MCAL-5897</a> - ADC Module shall follow the sequence diagram as described in Autosar Spec <b>PUBLISHED</b>	ADC module implementation shall follow that sequence diagram as described in <a href="#">AUTOSAR BSW ADC Driver Specification</a> section 9.
2 6	Configuration Specification	 <a href="#">MCAL-5662</a> - ECUC_Adc_00462: Adc Configuration Container <b>PUBLISHED</b>	[SWS_Dio_00129] At least one of the following configuration variants has to be supported by implementation: - VARIANT-PRE-COMPIL - VARIANT-POST-BUILD

## 8.3 WDG Driver

WDG employs built in hardware module called "RTI", RTI is a down-counter that count from non-zero value to zero and generates an interrupt to CPU or ESM which in-turn could be programmed to reset the SoC as detailed in the (1) [AUTOSAR BSW WDG Driver Specification](#), and will make use of the GPIO internal peripheral.

RTI(Real Time Interrupt) module supports three functional modes Counter mode, Capture mode & Windowed watchdog timer mode. Only Windowed watchdog timer mode is used to meet AUTOSAR WDG requirements.

The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW WDG Driver Specification](#).

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	Safety Process/ Diagnostic	 <a href="#">MCAL-5893</a> - WDG Module shall include the Safety Diagnostic Features <span style="background-color: #d9f7d9; padding: 2px;">PUBLISHED</span>	Following the Safety manual RTI IP and FMA review, the required safety diagnostic features should be added.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
2	Dependencies on other modules	 <a href="#">MCAL-5554 - [External Dependency]ESM</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	Wdg module depends on the external module ESM to generate pulse after expiry which could be used to trigger SoC reset.
3	File Structure	 <a href="#">MCAL-5573 - SWS_Wdg_00061 : File Structure</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	Wdg shall comply to file structure as described in <a href="#">AUTOSAR BSW WDG Driver Specification</a> .



	Specification Category	Architecture Identifier	Description
4	Error Classification and Detection	 <a href="#">MCAL-5632</a> - WDG Module: Shall implement development errors and extended production errors <span>PUBLISHED</span>	Shall allow development errors, and extended production errors.
5	Error Classification and Detection	 <a href="#">MCAL-5583</a> - SWS_Wdg_00111 : Optional Interfaces <span>PUBLISHED</span>	[SWS_Wdg_00111] Dem_SetEventStatus and Set_ReportError services shall be used.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
6	Internal WDG Driver	 <a href="#">MCAL-5607 - SWS_Wdg_00161 : Internal Wdg Req : Access PUBLISHED</a>	To access the internal watchdog hardware, the corresponding Wdg module instance shall access the hardware for watchdog servicing directly
7	Functionality	 <a href="#">MCAL-5595 - WDG : Trigger Routine PUBLISHED</a>	Triggering concept shall be implemented to support windowed watchdogs. This is to decouple the timing from servicing the watchdog hardware from the logical control.
8	Type Definitions	 <a href="#">MCAL-5631 - WDG Module: Imported Types and Defined Types PUBLISHED</a>	The WDG Driver shall have Imported types from following modules: Dem, Std_Types and WdgIf. Refer to SWS_Wdg_00105. It shall also define a new type for "Wdg_COnfigType", refer to SWS_Wdg_00171.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																												
9	Function Definitions	 <a href="#">MCAL-5584 - SWS_Wdg_00106 : API Wdg_Init</a> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Adc_00106] shall be implemented and follow <a href="#">AUTOSAR BSW WDG Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Wdg_Init</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Wdg_Init(                   const Wdg_ConfigType* ConfigPtr                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x00</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ConfigPtr</td> <td>Pointer to configuration set.</td> </tr> <tr> <td><b>Parameters</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>(inout):</b></td> <td colspan="2"></td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Initializes the module.</td> </tr> </table>	<b>Service name:</b>	Wdg_Init	<b>Syntax:</b>	void Wdg_Init( const Wdg_ConfigType* ConfigPtr )	<b>Service ID[hex]:</b>	0x00	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	ConfigPtr	Pointer to configuration set.	<b>Parameters</b>	None		<b>(inout):</b>			<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Initializes the module.	
<b>Service name:</b>	Wdg_Init																														
<b>Syntax:</b>	void Wdg_Init( const Wdg_ConfigType* ConfigPtr )																														
<b>Service ID[hex]:</b>	0x00																														
<b>Sync/Async:</b>	Synchronous																														
<b>Reentrancy:</b>	Non Reentrant																														
<b>Parameters (in):</b>	ConfigPtr	Pointer to configuration set.																													
<b>Parameters</b>	None																														
<b>(inout):</b>																															
<b>Parameters (out):</b>	None																														
<b>Return value:</b>	None																														
<b>Description:</b>	Initializes the module.																														

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
10	Function Definitions	 <a href="#">MCAL-5567</a> - SWS_Wdg_00155 : API Wdg_SetTriggerCondition <span style="background-color: #e0f2e0; color: green;">PUBLISHED</span>	<p>[SWS_AdC_00115] shall be implemented and follow <a href="#">AUTOSAR BSW WDG Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Wdg_SetTriggerCondition</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Wdg_SetTriggerCondition(                   uint16 timeout                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x03</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>timeout      Timeout value (milliseconds) for setting the trigger counter.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Sets the timeout value for the trigger counter.</td> </tr> </table>	<b>Service name:</b>	Wdg_SetTriggerCondition	<b>Syntax:</b>	void Wdg_SetTriggerCondition( uint16 timeout )	<b>Service ID[hex]:</b>	0x03	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	timeout      Timeout value (milliseconds) for setting the trigger counter.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Sets the timeout value for the trigger counter.
<b>Service name:</b>	Wdg_SetTriggerCondition																						
<b>Syntax:</b>	void Wdg_SetTriggerCondition( uint16 timeout )																						
<b>Service ID[hex]:</b>	0x03																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	timeout      Timeout value (milliseconds) for setting the trigger counter.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Sets the timeout value for the trigger counter.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																						
11	Function Definitions	 <a href="#">MCAL-5576 - SWS_Wdg_00109 : API</a> Wdg_GetVersionInfo <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Adc_00109] shall be implemented and follow <a href="#">AUTOSAR BSW WDG Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Wdg_GetVersionInfo</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Wdg_GetVersionInfo(     Std_VersionInfoType* versioninfo )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x04</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>versioninfo</td> <td>Pointer to where to store the version information of this module.</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Returns the version information of the module.</td> </tr> </table>	<b>Service name:</b>	Wdg_GetVersionInfo	<b>Syntax:</b>	void Wdg_GetVersionInfo( Std_VersionInfoType* versioninfo )	<b>Service ID[hex]:</b>	0x04	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	versioninfo	Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	Returns the version information of the module.	
<b>Service name:</b>	Wdg_GetVersionInfo																								
<b>Syntax:</b>	void Wdg_GetVersionInfo( Std_VersionInfoType* versioninfo )																								
<b>Service ID[hex]:</b>	0x04																								
<b>Sync/Async:</b>	Synchronous																								
<b>Reentrancy:</b>	Reentrant																								
<b>Parameters (in):</b>	None																								
<b>Parameters (inout):</b>	None																								
<b>Parameters (out):</b>	versioninfo	Pointer to where to store the version information of this module.																							
<b>Return value:</b>	None																								
<b>Description:</b>	Returns the version information of the module.																								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
12	Sequence Diagrams	 <a href="#">MCAL-5630 - WDG Module implementation shall follow sequence diagram as per AUTOSAR Spec.</a> <b>PUBLISHED</b>	WDG module implementation shall follow that sequence diagram as described in <a href="#">AUTOSAR BSW WDG Driver Specification</a> section 9.
13	Configuration Specification	 <a href="#">MCAL-5582 - ECUC_Wdg_00073 : Module Name Wdg</a> <b>PUBLISHED</b>	[SWS_Wdg_00073] At least one of the following configuration variants has to be supported by implementation: - VARIANT-PRE-COMPIL , VARIANT-LINK-TIME, VARAINT-POST-BUILD

## 8.4 CAN Driver

CAN module offers following services.

- On L-PDU transmission, the Can module writes the L-PDU in an appropriate buffer inside the CAN controller hardware.
- On L-PDU reception, the Can module calls the RX indication callback function with ID, DLC and pointer to L-SDU as parameter.
- The Can module provides an interface that serves as periodical processing function, and which must be called by the Basic Software Scheduler module periodically.



- The Can module provides services to control the state of the CAN controllers. Bus-off and Wake-up events are notified by means of callback functions

The Controller Area Network (CAN) is a serial communications protocol which efficiently supports distributed real-time control. CAN has high immunity to electrical interference. In a CAN network, many short messages are broadcast to the entire network, which provides for data consistency in every node of the system.

The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW CAN Driver Specification](#).

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	Safety Process/ Diagnostic	 <a href="#">MCAL-6100</a> - CAN Module shall include the Safety Diagnostic Features <span style="background-color: #e0f2e0; border: 1px solid #4CAF50; color: #4CAF50; padding: 2px 5px;">PUBLISHED</span>	Following the Safety manual MCAN IP and FMA review, the required safety diagnostic features should be added.
2	Dependency on other modules	 <a href="#">MCAL-6171</a> - Dependency on other module : CanTrcv <span style="background-color: #e0f2e0; border: 1px solid #4CAF50; color: #4CAF50; padding: 2px 5px;">PUBLISHED</span>	The Can module's environment shall ensure Can Transceiver is up and running before any Can transmission starts.



	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
3	File Structure	 <a href="#">MCAL-5967 - SWS_Can_00034 : File Structure</a> <span>PUBLISHED</span>	
4	Functional Specification	 <a href="#">MCAL-6182 - CAN Module shall follow the functional specification for BSW.</a> <span>PUBLISHED</span>	CAN Module shall follow the functional specification for BSW.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
5	Functional Specification	 <a href="#">MCAL-6183</a> - CAN Module provides access to one CAN Hardware Unit that may consist of several CAN controllers. <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	Can module provides access to one CAN Hardware Unit that may consist of several CAN controllers.
6	State Machine	 <a href="#">MCAL-6184</a> - CAN Module has a very simple state machine, with the two states CAN_UNINIT and CAN_READY <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	The Can module has a very simple state machine, with the two states CAN_UNINIT and CAN_READY

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
7	Initialization	 <a href="#">MCAL-6185 - CAN module initialization</a> <span>PUBLISHED</span>	The ECU State Manager module shall initialize the Can module during startup phase by calling the function Can_Init before using any other functions of the Can module.
8	Transmission	 <a href="#">MCAL-6186 - CAN module L-PDU transmission</a> <span>PUBLISHED</span>	On L-PDU transmission, the Can module converts the L-PDU contents ID and Data Length to a hardware specific format (if necessary) and triggers the transmission
9	Notification Concept	 <a href="#">MCAL-6187 - CAN module offers only an even triggered notification</a> <span>PUBLISHED</span>	CAN module offers only an even triggered notification

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
10	Error Classification and Detection	 <a href="#">MCAL-6031 - SWS_Can_00104 : Error Classification</a> <span style="background-color: #e0f7fa; padding: 2px 5px;">PUBLISHED</span>	[SWS_Can_00104] <sup>†</sup> The Can module shall be able to detect the following errors and exceptions depending on its configuration
11	Type Definitions	 <a href="#">MCAL-6188 - CAN module shall define the types as defined in AUTOSAR</a> <span style="background-color: #e0f7fa; padding: 2px 5px;">PUBLISHED</span>	Imported types and type definitions.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
12	Function Definitions	 <a href="#">MCAL-5969 - SWS_Can_00223 : Can_Init API</a> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Can_00223] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td><td style="padding: 2px;">Can_Init</td></tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td><td style="padding: 2px;">void Can_Init(                   const Can_ConfigType* Config                   )</td></tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td><td style="padding: 2px;">0x00</td></tr> <tr> <td style="padding: 2px;"><b>Sync/Async:</b></td><td style="padding: 2px;">Synchronous</td></tr> <tr> <td style="padding: 2px;"><b>Reentrancy:</b></td><td style="padding: 2px;">Non Reentrant</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (in):</b></td><td style="padding: 2px;">Config      Pointer to driver configuration.</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (inout):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (out):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Return value:</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Description:</b></td><td style="padding: 2px;">This function initializes the module.</td></tr> </table>	<b>Service name:</b>	Can_Init	<b>Syntax:</b>	void Can_Init( const Can_ConfigType* Config )	<b>Service ID[hex]:</b>	0x00	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	Config      Pointer to driver configuration.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function initializes the module.
<b>Service name:</b>	Can_Init																						
<b>Syntax:</b>	void Can_Init( const Can_ConfigType* Config )																						
<b>Service ID[hex]:</b>	0x00																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	Config      Pointer to driver configuration.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function initializes the module.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
13	Function Definitions	 <a href="#">MCAL-6165</a> - <b>SWS_Can_00224 :</b> <b>Can_GetVersionInfo</b> API <b>PUBLISHED</b>	<p>[SWS_Can_00224] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1"> <tr> <td><b>Service name:</b></td><td><code>Can_GetVersionInfo</code></td></tr> <tr> <td><b>Syntax:</b></td><td><code>void Can_GetVersionInfo(                   Std_VersionInfoType* versioninfo )</code></td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x07</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>None</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td><code>versioninfo</code>   Pointer to where to store the version information of this module.</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This function returns the version information of this module.</td></tr> </table>	<b>Service name:</b>	<code>Can_GetVersionInfo</code>	<b>Syntax:</b>	<code>void Can_GetVersionInfo(                   Std_VersionInfoType* versioninfo )</code>	<b>Service ID[hex]:</b>	0x07	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	<code>versioninfo</code>   Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	This function returns the version information of this module.
<b>Service name:</b>	<code>Can_GetVersionInfo</code>																						
<b>Syntax:</b>	<code>void Can_GetVersionInfo(                   Std_VersionInfoType* versioninfo )</code>																						
<b>Service ID[hex]:</b>	0x07																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	<code>versioninfo</code>   Pointer to where to store the version information of this module.																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function returns the version information of this module.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
14	Function Definitions	 <a href="#">MCAL-5990 - SWS_Can_91002 : Can_DeInit API</a> <span style="background-color: #e0f7fa; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Can_91002] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td><td>Can_DeInit</td></tr> <tr> <td><b>Syntax:</b></td><td>void Can_DeInit(                   void                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x10</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Non Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>None</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This function de-initializes the module.</td></tr> </table>	<b>Service name:</b>	Can_DeInit	<b>Syntax:</b>	void Can_DeInit( void )	<b>Service ID[hex]:</b>	0x10	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function de-initializes the module.
<b>Service name:</b>	Can_DeInit																						
<b>Syntax:</b>	void Can_DeInit( void )																						
<b>Service ID[hex]:</b>	0x10																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function de-initializes the module.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
15	Function Definitions	 MCAL-6065 - SWS_CAN_00491 : Can_SetBaudrate API <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	[SWS_CAN_00491] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a>

Specification Category	Architecture Identifier	Description											
		<p><b>Service name:</b> Can_SetBaudrate</p> <p><b>Syntax:</b></p> <pre>Std_ReturnType Can_SetBaudrate(     uint8 Controller,     uint16 BaudRateConfigID )</pre> <p><b>Service ID[hex]:</b> 0x0f</p> <p><b>Sync/Async:</b> Synchronous</p> <p><b>Reentrancy:</b> Reentrant for different Controllers. Non reentrant for the same Controller.</p> <table border="1" data-bbox="799 790 2061 886"> <tr> <td data-bbox="799 790 1096 822"><b>Parameters (in):</b></td> <td data-bbox="1096 790 1320 822">Controller</td> <td data-bbox="1320 790 2061 822">CAN controller, whose baud rate shall be set</td> </tr> <tr> <td></td> <td data-bbox="1096 822 1320 886">BaudRateConfigID</td> <td data-bbox="1320 822 2061 886">references a baud rate configuration by ID (see CanControllerBaudRateConfigID)</td> </tr> </table> <p><b>Parameters (inout):</b> None</p> <p><b>Parameters (out):</b> None</p> <p><b>Return value:</b></p> <table border="1" data-bbox="799 997 2061 1092"> <tr> <td data-bbox="799 997 1096 1029">Std_ReturnType</td> <td data-bbox="1096 997 2061 1029">E_OK: Service request accepted, setting of (new) baud rate started</td> </tr> <tr> <td></td> <td data-bbox="1096 1029 2061 1092">E_NOT_OK: Service request not accepted</td> </tr> </table> <p><b>Description:</b> This service shall set the baud rate configuration of the CAN controller. Depending on necessary baud rate modifications the controller might have to reset.</p>	<b>Parameters (in):</b>	Controller	CAN controller, whose baud rate shall be set		BaudRateConfigID	references a baud rate configuration by ID (see CanControllerBaudRateConfigID)	Std_ReturnType	E_OK: Service request accepted, setting of (new) baud rate started		E_NOT_OK: Service request not accepted	
<b>Parameters (in):</b>	Controller	CAN controller, whose baud rate shall be set											
	BaudRateConfigID	references a baud rate configuration by ID (see CanControllerBaudRateConfigID)											
Std_ReturnType	E_OK: Service request accepted, setting of (new) baud rate started												
	E_NOT_OK: Service request not accepted												

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
16	Function Definitions	 <a href="#">MCAL-6093 - SWS_Can_00230: Can_SetControllerMode API</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	[SWS_Can_00230] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a>

Specification Category	Architecture Identifier	Description										
		<p><b>Service name:</b> Can_SetControllerMode</p> <p><b>Syntax:</b></p> <pre>Std_ReturnType Can_SetControllerMode(     uint8 Controller,     Can_ControllerStateType Transition )</pre> <p><b>Service ID[hex]:</b> 0x03</p> <p><b>Sync/Async:</b> Asynchronous</p> <p><b>Reentrancy:</b> Non Reentrant</p> <table border="1" data-bbox="810 790 2061 859"> <tr> <td data-bbox="810 790 1260 827"><b>Parameters (in):</b></td><td data-bbox="1260 790 1327 827">Controller</td><td data-bbox="1327 790 2061 827">CAN controller for which the status shall be changed</td></tr> <tr> <td data-bbox="810 827 1260 859"></td><td data-bbox="1260 827 1327 859">Transition</td><td data-bbox="1327 827 2061 859">Transition value to request new CAN controller state</td></tr> </table> <p><b>Parameters (inout):</b> None</p> <p><b>Parameters (out):</b> None</p> <p><b>Return value:</b></p> <table border="1" data-bbox="810 970 2061 1092"> <tr> <td data-bbox="810 970 1260 1092"></td><td data-bbox="1260 970 1327 1092">Std_ReturnType</td><td data-bbox="1327 970 2061 1092">           E_OK: request accepted            E_NOT_OK: request not accepted, a development error occurred         </td></tr> </table> <p><b>Description:</b> This function performs software triggered state transitions of the CAN controller State machine.</p>	<b>Parameters (in):</b>	Controller	CAN controller for which the status shall be changed		Transition	Transition value to request new CAN controller state		Std_ReturnType	E_OK: request accepted E_NOT_OK: request not accepted, a development error occurred	
<b>Parameters (in):</b>	Controller	CAN controller for which the status shall be changed										
	Transition	Transition value to request new CAN controller state										
	Std_ReturnType	E_OK: request accepted E_NOT_OK: request not accepted, a development error occurred										

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
17	Function Definitions	 <a href="#">MCAL-6056 - SWS_Can_00231: Can_DisableControllerInterruptions API</a> <span style="background-color: #d9ffd9; border: 1px solid #2e6b2e; padding: 2px 5px; border-radius: 5px;">PUBLISHED</span>	<p>[SWS_Can_00231] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Can_DisableControllerInterruptions</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Can_DisableControllerInterruptions(                   uint8 Controller                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x04</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Controller   CAN controller for which interrupts shall be disabled.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>This function disables all interrupts for this CAN controller.</td> </tr> </table>	<b>Service name:</b>	Can_DisableControllerInterruptions	<b>Syntax:</b>	void Can_DisableControllerInterruptions( uint8 Controller )	<b>Service ID[hex]:</b>	0x04	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Controller   CAN controller for which interrupts shall be disabled.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function disables all interrupts for this CAN controller.
<b>Service name:</b>	Can_DisableControllerInterruptions																						
<b>Syntax:</b>	void Can_DisableControllerInterruptions( uint8 Controller )																						
<b>Service ID[hex]:</b>	0x04																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	Controller   CAN controller for which interrupts shall be disabled.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function disables all interrupts for this CAN controller.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
18	Function Definitions	 MCAL-6045 - SWS_Can_00232 : Can_EnableControllerInterrupts API <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Can_00232] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Can_EnableControllerInterrupts</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Can_EnableControllerInterrupts(                   uint8 Controller                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x05</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Controller   CAN controller for which interrupts shall be re-enabled</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>This function enables all allowed interrupts.</td> </tr> </table>	<b>Service name:</b>	Can_EnableControllerInterrupts	<b>Syntax:</b>	void Can_EnableControllerInterrupts( uint8 Controller )	<b>Service ID[hex]:</b>	0x05	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Controller   CAN controller for which interrupts shall be re-enabled	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function enables all allowed interrupts.
<b>Service name:</b>	Can_EnableControllerInterrupts																						
<b>Syntax:</b>	void Can_EnableControllerInterrupts( uint8 Controller )																						
<b>Service ID[hex]:</b>	0x05																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	Controller   CAN controller for which interrupts shall be re-enabled																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function enables all allowed interrupts.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
19	Function Definitions	 <a href="#">MCAL-6167</a> - <b>SWS_Can_91004 :</b> Can_GetControllerErrorState API <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Can_91004] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td> <td style="padding: 2px;">Can_GetControllerErrorState</td> </tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td> <td style="padding: 2px;">Std_ReturnType Can_GetControllerErrorState(     uint8 ControllerId,     Can_ErrorStateType* ErrorStatePtr )</td> </tr> <tr> <td colspan="2" style="padding: 2px;"><b>Service ID[hex]:</b> 0x11</td> </tr> <tr> <td colspan="2" style="padding: 2px;"><b>Sync/Async:</b> Synchronous</td> </tr> <tr> <td colspan="2" style="padding: 2px;"><b>Reentrancy:</b> Non Reentrant for the same ControllerId</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (in):</b></td> <td style="padding: 2px;">ControllerId      Abstracted CanIf ControllerId which is assigned to a CAN controller, which is requested for ErrorState.</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (inout):</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (out):</b></td> <td style="padding: 2px;">ErrorStatePtr      Pointer to a memory location, where the error state of the CAN controller will be stored.</td> </tr> <tr> <td style="padding: 2px;"><b>Return value:</b></td> <td style="padding: 2px;">Std_ReturnType     E_OK: Error state request has been accepted.     E_NOT_OK: Error state request has not been accepted.</td> </tr> <tr> <td colspan="2" style="padding: 2px;"><b>Description:</b> This service obtains the error state of the CAN controller.</td> </tr> </table>	<b>Service name:</b>	Can_GetControllerErrorState	<b>Syntax:</b>	Std_ReturnType Can_GetControllerErrorState( uint8 ControllerId, Can_ErrorStateType* ErrorStatePtr )	<b>Service ID[hex]:</b> 0x11		<b>Sync/Async:</b> Synchronous		<b>Reentrancy:</b> Non Reentrant for the same ControllerId		<b>Parameters (in):</b>	ControllerId      Abstracted CanIf ControllerId which is assigned to a CAN controller, which is requested for ErrorState.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	ErrorStatePtr      Pointer to a memory location, where the error state of the CAN controller will be stored.	<b>Return value:</b>	Std_ReturnType E_OK: Error state request has been accepted. E_NOT_OK: Error state request has not been accepted.	<b>Description:</b> This service obtains the error state of the CAN controller.	
<b>Service name:</b>	Can_GetControllerErrorState																						
<b>Syntax:</b>	Std_ReturnType Can_GetControllerErrorState( uint8 ControllerId, Can_ErrorStateType* ErrorStatePtr )																						
<b>Service ID[hex]:</b> 0x11																							
<b>Sync/Async:</b> Synchronous																							
<b>Reentrancy:</b> Non Reentrant for the same ControllerId																							
<b>Parameters (in):</b>	ControllerId      Abstracted CanIf ControllerId which is assigned to a CAN controller, which is requested for ErrorState.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	ErrorStatePtr      Pointer to a memory location, where the error state of the CAN controller will be stored.																						
<b>Return value:</b>	Std_ReturnType E_OK: Error state request has been accepted. E_NOT_OK: Error state request has not been accepted.																						
<b>Description:</b> This service obtains the error state of the CAN controller.																							



	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																														
20	Function Definitions	 <a href="#">MCAL-6007 - SWS_Can_91014 : Can_GetControllerMode API</a> <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Can_91014] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Can_GetControllerMode</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">Std_ReturnType Can_GetControllerMode(     uint8 Controller,     Can_ControllerStateType* ControllerModePtr )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x12</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Controller</td> <td>CAN controller for which the status shall be requested.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>ControllerModePtr</td> <td>Pointer to a memory location, where the current mode of the CAN controller will be stored.</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: Controller mode request has been accepted. E_NOT_OK: Controller mode request has not been accepted.</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">This service reports about the current status of the requested CAN controller.</td> </tr> </table>	<b>Service name:</b>	Can_GetControllerMode		<b>Syntax:</b>	Std_ReturnType Can_GetControllerMode( uint8 Controller, Can_ControllerStateType* ControllerModePtr )		<b>Service ID[hex]:</b>	0x12		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Non Reentrant		<b>Parameters (in):</b>	Controller	CAN controller for which the status shall be requested.	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	ControllerModePtr	Pointer to a memory location, where the current mode of the CAN controller will be stored.	<b>Return value:</b>	Std_ReturnType	E_OK: Controller mode request has been accepted. E_NOT_OK: Controller mode request has not been accepted.	<b>Description:</b>	This service reports about the current status of the requested CAN controller.	
<b>Service name:</b>	Can_GetControllerMode																																
<b>Syntax:</b>	Std_ReturnType Can_GetControllerMode( uint8 Controller, Can_ControllerStateType* ControllerModePtr )																																
<b>Service ID[hex]:</b>	0x12																																
<b>Sync/Async:</b>	Synchronous																																
<b>Reentrancy:</b>	Non Reentrant																																
<b>Parameters (in):</b>	Controller	CAN controller for which the status shall be requested.																															
<b>Parameters (inout):</b>	None																																
<b>Parameters (out):</b>	ControllerModePtr	Pointer to a memory location, where the current mode of the CAN controller will be stored.																															
<b>Return value:</b>	Std_ReturnType	E_OK: Controller mode request has been accepted. E_NOT_OK: Controller mode request has not been accepted.																															
<b>Description:</b>	This service reports about the current status of the requested CAN controller.																																

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
21	Function Definitions	 MCAL-6038 - SWS_Can_00233 : Can_Write API <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	[SWS_Can_00233] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>															
			<p><b>Service name:</b> Can_Write</p> <p><b>Syntax:</b> Std_ReturnType Can_Write(     Can_HwHandleType Hth,     const Can_PduType* PduInfo )</p> <p><b>Service ID[hex]:</b> 0x06</p> <p><b>Sync/Async:</b> Synchronous</p> <p><b>Reentrancy:</b> Reentrant (thread-safe)</p> <table border="1"> <tr> <td><b>Parameters (in):</b></td> <td>Hth</td> <td>information which HW-transmit handle shall be used for transmit. Implicitly this is also the information about the controller to use because the Hth numbers are unique inside one hardware unit.</td> </tr> <tr> <td></td> <td>PduInfo</td> <td>Pointer to SDU user memory, Data Length and Identifier.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: Write command has been accepted  E_NOT_OK: development error occurred  CAN_BUSY: No TX hardware buffer available or pre-emptive call of Can_Write that can't be implemented re-entrant (see</td> </tr> </table>	<b>Parameters (in):</b>	Hth	information which HW-transmit handle shall be used for transmit. Implicitly this is also the information about the controller to use because the Hth numbers are unique inside one hardware unit.		PduInfo	Pointer to SDU user memory, Data Length and Identifier.	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Std_ReturnType	E_OK: Write command has been accepted  E_NOT_OK: development error occurred  CAN_BUSY: No TX hardware buffer available or pre-emptive call of Can_Write that can't be implemented re-entrant (see
<b>Parameters (in):</b>	Hth	information which HW-transmit handle shall be used for transmit. Implicitly this is also the information about the controller to use because the Hth numbers are unique inside one hardware unit.																
	PduInfo	Pointer to SDU user memory, Data Length and Identifier.																
<b>Parameters (inout):</b>	None																	
<b>Parameters (out):</b>	None																	
<b>Return value:</b>	Std_ReturnType	E_OK: Write command has been accepted  E_NOT_OK: development error occurred  CAN_BUSY: No TX hardware buffer available or pre-emptive call of Can_Write that can't be implemented re-entrant (see																

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>								
			<table border="1"> <tr> <td></td><td>Can_ReturnType)</td></tr> <tr> <td><b>Description:</b></td><td>This function is called by CanIf to pass a CAN message to CanDrv for transmission.</td></tr> </table>		Can_ReturnType)	<b>Description:</b>	This function is called by CanIf to pass a CAN message to CanDrv for transmission.				
	Can_ReturnType)										
<b>Description:</b>	This function is called by CanIf to pass a CAN message to CanDrv for transmission.										
22	Function Definitions	 <a href="#">MCAL-6136</a> - <b>SWS_Can_00225 :</b> Can_MainFunction_Write API <span style="color: green;">PUBLISHED</span>	<p>[SWS_Can_00225] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Can_MainFunction_Write</td></tr> <tr> <td><b>Syntax:</b></td><td>void Can_MainFunction_Write(     void )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x01</td></tr> <tr> <td><b>Description:</b></td><td>This function performs the polling of TX confirmation when CAN_TX_PROCESSING is set to POLLING.</td></tr> </table>	<b>Service name:</b>	Can_MainFunction_Write	<b>Syntax:</b>	void Can_MainFunction_Write( void )	<b>Service ID[hex]:</b>	0x01	<b>Description:</b>	This function performs the polling of TX confirmation when CAN_TX_PROCESSING is set to POLLING.
<b>Service name:</b>	Can_MainFunction_Write										
<b>Syntax:</b>	void Can_MainFunction_Write( void )										
<b>Service ID[hex]:</b>	0x01										
<b>Description:</b>	This function performs the polling of TX confirmation when CAN_TX_PROCESSING is set to POLLING.										

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>								
23	Function Definitions	 <a href="#">MCAL-6157 - SWS_Can_00226 : Can_MainFunction_Read API</a> <span style="background-color: #e0f2e0; border: 1px solid #80c080; padding: 2px;">PUBLISHED</span>	<p>[SWS_Can_00226] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td><td style="padding: 2px;">Can_MainFunction_Read</td></tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td><td style="padding: 2px;">void Can_MainFunction_Read(     void )</td></tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td><td style="padding: 2px;">0x08</td></tr> <tr> <td style="padding: 2px;"><b>Description:</b></td><td style="padding: 2px;">This function performs the polling of RX indications when CAN_RX_PROCESSING is set to POLLING.</td></tr> </table>	<b>Service name:</b>	Can_MainFunction_Read	<b>Syntax:</b>	void Can_MainFunction_Read( void )	<b>Service ID[hex]:</b>	0x08	<b>Description:</b>	This function performs the polling of RX indications when CAN_RX_PROCESSING is set to POLLING.
<b>Service name:</b>	Can_MainFunction_Read										
<b>Syntax:</b>	void Can_MainFunction_Read( void )										
<b>Service ID[hex]:</b>	0x08										
<b>Description:</b>	This function performs the polling of RX indications when CAN_RX_PROCESSING is set to POLLING.										
24	Function Definitions	 <a href="#">MCAL-5953 - SWS_Can_00227 : Can_MainFunction_BusOff API</a> <span style="background-color: #e0f2e0; border: 1px solid #80c080; padding: 2px;">PUBLISHED</span>	<p>[SWS_Can_00227] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td><td style="padding: 2px;">Can_MainFunction_BusOff</td></tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td><td style="padding: 2px;">void Can_MainFunction_BusOff(     void )</td></tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td><td style="padding: 2px;">0x09</td></tr> <tr> <td style="padding: 2px;"><b>Description:</b></td><td style="padding: 2px;">This function performs the polling of bus-off events that are configured statically as 'to be polled'.</td></tr> </table>	<b>Service name:</b>	Can_MainFunction_BusOff	<b>Syntax:</b>	void Can_MainFunction_BusOff( void )	<b>Service ID[hex]:</b>	0x09	<b>Description:</b>	This function performs the polling of bus-off events that are configured statically as 'to be polled'.
<b>Service name:</b>	Can_MainFunction_BusOff										
<b>Syntax:</b>	void Can_MainFunction_BusOff( void )										
<b>Service ID[hex]:</b>	0x09										
<b>Description:</b>	This function performs the polling of bus-off events that are configured statically as 'to be polled'.										

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>								
25	Function Definitions	 <a href="#">MCAL-6178 - SWS_Can_00228 : Can_MainFunction_Wakeup API</a> <span style="background-color: #90EE90; border: 1px solid black; padding: 2px;">PUBLISHED</span>	<p>[SWS_Can_00228] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td><td style="padding: 2px;">Can_MainFunction_Wakeup</td></tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td><td style="padding: 2px;">void Can_MainFunction_Wakeup(     void )</td></tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td><td style="padding: 2px;">0x0a</td></tr> <tr> <td style="padding: 2px;"><b>Description:</b></td><td style="padding: 2px;">This function performs the polling of wake-up events that are configured statically as 'to be polled'.</td></tr> </table>	<b>Service name:</b>	Can_MainFunction_Wakeup	<b>Syntax:</b>	void Can_MainFunction_Wakeup( void )	<b>Service ID[hex]:</b>	0x0a	<b>Description:</b>	This function performs the polling of wake-up events that are configured statically as 'to be polled'.
<b>Service name:</b>	Can_MainFunction_Wakeup										
<b>Syntax:</b>	void Can_MainFunction_Wakeup( void )										
<b>Service ID[hex]:</b>	0x0a										
<b>Description:</b>	This function performs the polling of wake-up events that are configured statically as 'to be polled'.										
26	Function Definitions	 <a href="#">MCAL-6123 - SWS_Can_00368 : Can_MainFunction_Mode API</a> <span style="background-color: #90EE90; border: 1px solid black; padding: 2px;">PUBLISHED</span>	<p>[SWS_Can_00368] shall be implemented and follow <a href="#">AUTOSAR BSW CAN Driver</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td><td style="padding: 2px;">Can_MainFunction_Mode</td></tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td><td style="padding: 2px;">void Can_MainFunction_Mode(     void )</td></tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td><td style="padding: 2px;">0x0c</td></tr> <tr> <td style="padding: 2px;"><b>Description:</b></td><td style="padding: 2px;">This function performs the polling of CAN controller mode transitions.</td></tr> </table>	<b>Service name:</b>	Can_MainFunction_Mode	<b>Syntax:</b>	void Can_MainFunction_Mode( void )	<b>Service ID[hex]:</b>	0x0c	<b>Description:</b>	This function performs the polling of CAN controller mode transitions.
<b>Service name:</b>	Can_MainFunction_Mode										
<b>Syntax:</b>	void Can_MainFunction_Mode( void )										
<b>Service ID[hex]:</b>	0x0c										
<b>Description:</b>	This function performs the polling of CAN controller mode transitions.										

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
27	Interfaces	 <a href="#">MCAL-7470 - CAN</a> Module shall implement option and mandatory interfaces <span style="background-color: #d9e1f2; padding: 2px 5px;">PUBLISHED</span>	CAN module shall implemented some optional and as well as some mandatory interfaces. Module will use the CanIf, Det module for callback functions.
28	Sequence Diagrams	 <a href="#">MCAL-7471 - CAN</a> Module shall follow the sequence diagram in AUTOSAR spec. <span style="background-color: #d9e1f2; padding: 2px 5px;">PUBLISHED</span>	CAN module implementation shall follow that sequence diagram as described in <a href="#">AUTOSAR BSW CAN Driver</a> section 9.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
29	Configuration Specification	 <a href="#">MCAL-7472 - CAN</a> Module shall have configuration with either Pre-Compile Variant, or Post-Build Variant, <a href="#">PUBLISHED</a>	CAN module shall follow the containers and configuration parameters as described in <a href="#">AUTOSAR BSW CAN Driver</a> section 10. - VARIANT-PRE-COMPILe , VARAINT-POST-BUILD

## 8.5 GPT Driver

The GPT module initializes, configures and controls the internal Timer(s) hardware to realize GPT driver as detailed in AUTOSAR BSW GPT Driver Specification. GPT primarily used to generate different time bases that other modules of AUTOSAR could depend on. Below listed are some of the key features provided.

- Free running 32 bit up counter
- Auto reload mode (can be used for continuous counter operation)
- Support dynamic Start / Stop counter operation
- Programmable clock dividers ( $2^n$ , where  $n = [0-8]$ )
- 2 timers modules could be operated in cascaded mode to provide 64bit counter
- Programmable interrupt generation on overflow, compare and capture • Programmable clock source



Supports 3 basic functional modes Timer mode, Capture mode & Compare mode.

The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW GPT Driver Specification](#).

	Specification Category	Architecture Identifier	Description
1	Safety Process/ Diagnostic	 <a href="#">MCAL-6253</a> - MCAL Module : GPT : Safety Diagnostic <span>PUBLISHED</span>	Following the Safety manual Timer IP and FMA review, the required safety diagnostic features should be added.
2	File Structure	 <a href="#">MCAL-7474</a> - GPT Module: Shall follow the file structure in AUTOSAR Spec. <span>PUBLISHED</span>	GPT shall follow the file structure as defined in <a href="#">AUTOSAR BSW GPT Driver Specification</a> section 5.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
3	General Behavior and State Machine	 <a href="#">MCAL-7475 - GPT</a> Module: Shall have services to start and stop timer channels. <span style="background-color: #c8f7e4; padding: 2px;">PUBLISHED</span>	The GPT driver provides services for starting and stopping timer channels (logical timer instances assigned to a timer hardware), individual for each channel by calling of: <b>Gpt_StartTimer</b> <b>Gpt_StopTimer</b>
4	Error Classification and Detection	 <a href="#">MCAL-7476 - GPT</a> Module: Shall have development Errors and Runtime Errors. <span style="background-color: #c8f7e4; padding: 2px;">PUBLISHED</span>	GPT module shall have development Errors and Runtime Errors as defined in section 7.4 of <a href="#">AUTOSAR BSW GPT Driver Specification</a> .
5	Imported Types	 <a href="#">MCAL-6277 - SWS_Gpt_00278 : Imported Types</a> <span style="background-color: #c8f7e4; padding: 2px;">PUBLISHED</span>	[SWS_Gpt_00278] GPT module shall use imported types EcuM and Std_Types.

	Specification Category	Architecture Identifier	Description
6	Type Definitions	 MCAL-7479 - GPT Module: Type Definitions <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	GPT module shall define types as described in section 8.2 of <a href="#">AUTOSAR BSW GPT Driver Specification</a> .

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																						
7	Function Definitions	 MCAL-6297 - SWS_Gpt_00279 : API : VersionInfo <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Gpt_00279] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Gpt_GetVersionInfo</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Gpt_GetVersionInfo(                   Std_VersionInfoType* VersionInfoPtr                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x00</td> </tr> <tr> <td> </td> <td></td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>VersionInfoPtr   Pointer to where to store the version information of this module.</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Returns the version information of this module.</td> </tr> </table>	<b>Service name:</b>	Gpt_GetVersionInfo	<b>Syntax:</b>	void Gpt_GetVersionInfo( Std_VersionInfoType* VersionInfoPtr )	<b>Service ID[hex]:</b>	0x00	 		<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	VersionInfoPtr   Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	Returns the version information of this module.
<b>Service name:</b>	Gpt_GetVersionInfo																								
<b>Syntax:</b>	void Gpt_GetVersionInfo( Std_VersionInfoType* VersionInfoPtr )																								
<b>Service ID[hex]:</b>	0x00																								
<b>Sync/Async:</b>	Synchronous																								
<b>Reentrancy:</b>	Reentrant																								
<b>Parameters (in):</b>	None																								
<b>Parameters (inout):</b>	None																								
<b>Parameters (out):</b>	VersionInfoPtr   Pointer to where to store the version information of this module.																								
<b>Return value:</b>	None																								
<b>Description:</b>	Returns the version information of this module.																								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
8	Function Definitions	 MCAL-6263 - <b>SWS_Gpt_00280:</b> API : Init <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Gpt_00280] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Gpt_Init</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Gpt_Init(                   const Gpt_ConfigType* ConfigPtr )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x01</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ConfigPtr      Pointer to a selected configuration structure</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Initializes the GPT driver.</td> </tr> </table>	<b>Service name:</b>	Gpt_Init	<b>Syntax:</b>	void Gpt_Init( const Gpt_ConfigType* ConfigPtr )	<b>Service ID[hex]:</b>	0x01	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	ConfigPtr      Pointer to a selected configuration structure	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Initializes the GPT driver.
<b>Service name:</b>	Gpt_Init																						
<b>Syntax:</b>	void Gpt_Init( const Gpt_ConfigType* ConfigPtr )																						
<b>Service ID[hex]:</b>	0x01																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	ConfigPtr      Pointer to a selected configuration structure																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Initializes the GPT driver.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
9	Function Definitions	 <a href="#">MCAL-6233 - SWS_Gpt_00281 : Delnit : Prototype</a> <span style="background-color: #e0f7fa; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Gpt_00281] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td> <td style="padding: 2px;">Gpt_Delnit</td> </tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td> <td style="padding: 2px;"> <pre>void Gpt_DeInit(     void )</pre> </td> </tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td> <td style="padding: 2px;">0x02</td> </tr> <tr> <td style="padding: 2px;"><b>Sync/Async:</b></td> <td style="padding: 2px;">Synchronous</td> </tr> <tr> <td style="padding: 2px;"><b>Reentrancy:</b></td> <td style="padding: 2px;">Non Reentrant</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (in):</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (inout):</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (out):</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Return value:</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Description:</b></td> <td style="padding: 2px;">Deinitializes the GPT driver.</td> </tr> </table>	<b>Service name:</b>	Gpt_Delnit	<b>Syntax:</b>	<pre>void Gpt_DeInit(     void )</pre>	<b>Service ID[hex]:</b>	0x02	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Deinitializes the GPT driver.
<b>Service name:</b>	Gpt_Delnit																						
<b>Syntax:</b>	<pre>void Gpt_DeInit(     void )</pre>																						
<b>Service ID[hex]:</b>	0x02																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Deinitializes the GPT driver.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																									
10	Function Definitions	 <b>MCAL-6218 - SWS_Gpt_00282 :</b> GTE : Service API : prototype <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Gpt_00282] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Gpt_GetTimeElapsed</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Gpt_ValueType Gpt_GetTimeElapsed(                   Gpt_ChannelType Channel                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x03</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel</td> <td>Numeric identifier of the GPT channel.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Gpt_ValueType</td> <td>Elapsed timer value (in number of ticks)</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Returns the time already elapsed.</td> </tr> </table>	<b>Service name:</b>	Gpt_GetTimeElapsed	<b>Syntax:</b>	Gpt_ValueType Gpt_GetTimeElapsed( Gpt_ChannelType Channel )	<b>Service ID[hex]:</b>	0x03	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Channel	Numeric identifier of the GPT channel.	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Gpt_ValueType	Elapsed timer value (in number of ticks)	<b>Description:</b>	Returns the time already elapsed.	
<b>Service name:</b>	Gpt_GetTimeElapsed																											
<b>Syntax:</b>	Gpt_ValueType Gpt_GetTimeElapsed( Gpt_ChannelType Channel )																											
<b>Service ID[hex]:</b>	0x03																											
<b>Sync/Async:</b>	Synchronous																											
<b>Reentrancy:</b>	Reentrant																											
<b>Parameters (in):</b>	Channel	Numeric identifier of the GPT channel.																										
<b>Parameters (inout):</b>	None																											
<b>Parameters (out):</b>	None																											
<b>Return value:</b>	Gpt_ValueType	Elapsed timer value (in number of ticks)																										
<b>Description:</b>	Returns the time already elapsed.																											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																														
11	Function Definitions	 <b>MCAL-6240 - SWS_Gpt_00283 : GTR : Service API : Prototype</b> <span style="background-color: #d9e1f2; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Gpt_00283] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Gpt_GetTimeRemaining</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">Gpt_ValueType Gpt_GetTimeRemaining(     Gpt_ChannelType Channel )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x04</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel</td> <td>Numeric identifier of the GPT channel.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Gpt_ValueType</td> <td>Remaining timer value (in number of ticks)</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Returns the time remaining until the target time is reached.</td> </tr> </table>	<b>Service name:</b>	Gpt_GetTimeRemaining		<b>Syntax:</b>	Gpt_ValueType Gpt_GetTimeRemaining( Gpt_ChannelType Channel )		<b>Service ID[hex]:</b>	0x04		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant		<b>Parameters (in):</b>	Channel	Numeric identifier of the GPT channel.	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Gpt_ValueType	Remaining timer value (in number of ticks)	<b>Description:</b>	Returns the time remaining until the target time is reached.	
<b>Service name:</b>	Gpt_GetTimeRemaining																																
<b>Syntax:</b>	Gpt_ValueType Gpt_GetTimeRemaining( Gpt_ChannelType Channel )																																
<b>Service ID[hex]:</b>	0x04																																
<b>Sync/Async:</b>	Synchronous																																
<b>Reentrancy:</b>	Reentrant																																
<b>Parameters (in):</b>	Channel	Numeric identifier of the GPT channel.																															
<b>Parameters (inout):</b>	None																																
<b>Parameters (out):</b>	None																																
<b>Return value:</b>	Gpt_ValueType	Remaining timer value (in number of ticks)																															
<b>Description:</b>	Returns the time remaining until the target time is reached.																																

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
12	Function Definitions	 <b>MCAL-6296 - SWS_Gpt_00284 : StartT : Service API</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Gpt_00284] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"><b>Service name:</b></td><td>Gpt_StartTimer</td></tr> <tr> <td><b>Syntax:</b></td><td>void Gpt_StartTimer(                   Gpt_ChannelType Channel,                   Gpt_ValueType Value                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x05</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (but not for the same timer channel)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the GPT channel. Value        Target time in number of ticks.</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Starts a timer channel.</td></tr> </table>	<b>Service name:</b>	Gpt_StartTimer	<b>Syntax:</b>	void Gpt_StartTimer( Gpt_ChannelType Channel, Gpt_ValueType Value )	<b>Service ID[hex]:</b>	0x05	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)	<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel. Value        Target time in number of ticks.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Starts a timer channel.
<b>Service name:</b>	Gpt_StartTimer																						
<b>Syntax:</b>	void Gpt_StartTimer( Gpt_ChannelType Channel, Gpt_ValueType Value )																						
<b>Service ID[hex]:</b>	0x05																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel. Value        Target time in number of ticks.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Starts a timer channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
13	Function Definitions	 <b>MCAL-6208 - SWS_Gpt_00285 : StopT : Service API : prototype</b> <span style="background-color: #d9f7d9; border: 1px solid black; padding: 2px;">PUBLISHED</span>	<p>[SWS_Gpt_00285] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;"><b>Service name:</b></td> <td>Gpt_StopTimer</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Gpt_StopTimer(                   Gpt_ChannelType Channel                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x06</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant (but not for the same timer channel)</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel      Numeric identifier of the GPT channel.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Stops a timer channel.</td> </tr> </table>	<b>Service name:</b>	Gpt_StopTimer	<b>Syntax:</b>	void Gpt_StopTimer( Gpt_ChannelType Channel )	<b>Service ID[hex]:</b>	0x06	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)	<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Stops a timer channel.
<b>Service name:</b>	Gpt_StopTimer																						
<b>Syntax:</b>	void Gpt_StopTimer( Gpt_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x06																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Stops a timer channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
14	Function Definitions	 <b>MCAL-6209 - SWS_Gpt_00286 : EnNotify : Service API</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Gpt_00286] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"><b>Service name:</b></td><td>Gpt_EnableNotification</td></tr> <tr> <td><b>Syntax:</b></td><td>void Gpt_EnableNotification(                   Gpt_ChannelType Channel                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x07</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (but not for the same timer channel)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the GPT channel.</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Enables the interrupt notification for a channel (relevant in normal mode).</td></tr> </table>	<b>Service name:</b>	Gpt_EnableNotification	<b>Syntax:</b>	void Gpt_EnableNotification( Gpt_ChannelType Channel )	<b>Service ID[hex]:</b>	0x07	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)	<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Enables the interrupt notification for a channel (relevant in normal mode).
<b>Service name:</b>	Gpt_EnableNotification																						
<b>Syntax:</b>	void Gpt_EnableNotification( Gpt_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x07																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Enables the interrupt notification for a channel (relevant in normal mode).																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
15	Function Definitions	 <a href="#">MCAL-6200 - SWS_Gpt_00287 : DisNotify : Service API</a> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Gpt_00287] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Gpt_DisableNotification</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Gpt_DisableNotification(                   Gpt_ChannelType Channel                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x08</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant (but not for the same timer channel)</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel      Numeric identifier of the GPT channel.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Disables the interrupt notification for a channel (relevant in normal mode).</td> </tr> </table>	<b>Service name:</b>	Gpt_DisableNotification	<b>Syntax:</b>	void Gpt_DisableNotification( Gpt_ChannelType Channel )	<b>Service ID[hex]:</b>	0x08	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)	<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Disables the interrupt notification for a channel (relevant in normal mode).
<b>Service name:</b>	Gpt_DisableNotification																						
<b>Syntax:</b>	void Gpt_DisableNotification( Gpt_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x08																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Disables the interrupt notification for a channel (relevant in normal mode).																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
16	Function Definitions	 <b>MCAL-6226 - SWS_Gpt_00288 : SetMode : Service API</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Gpt_00288] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Syntax:</b></td><td style="padding: 2px;">void Gpt_SetMode(     Gpt_ModeType Mode )</td></tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td><td style="padding: 2px;">0x09</td></tr> <tr> <td style="padding: 2px;"><b>Sync/Async:</b></td><td style="padding: 2px;">Synchronous</td></tr> <tr> <td style="padding: 2px;"><b>Reentrancy:</b></td><td style="padding: 2px;">Non Reentrant</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (in):</b></td><td style="padding: 2px; vertical-align: top;">Mode GPT_MODE_NORMAL: Normal operation mode of the GPT driver.  GPT_MODE_SLEEP: Sleep mode of the GPT driver (wakeup capable).  See also <a href="#">Gpt_ModeType</a>.</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (inout):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (out):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Return value:</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Description:</b></td><td style="padding: 2px;">Sets the operation mode of the GPT.</td></tr> </table>	<b>Syntax:</b>	void Gpt_SetMode( Gpt_ModeType Mode )	<b>Service ID[hex]:</b>	0x09	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	Mode GPT_MODE_NORMAL: Normal operation mode of the GPT driver.  GPT_MODE_SLEEP: Sleep mode of the GPT driver (wakeup capable).  See also <a href="#">Gpt_ModeType</a> .	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Sets the operation mode of the GPT.
<b>Syntax:</b>	void Gpt_SetMode( Gpt_ModeType Mode )																				
<b>Service ID[hex]:</b>	0x09																				
<b>Sync/Async:</b>	Synchronous																				
<b>Reentrancy:</b>	Non Reentrant																				
<b>Parameters (in):</b>	Mode GPT_MODE_NORMAL: Normal operation mode of the GPT driver.  GPT_MODE_SLEEP: Sleep mode of the GPT driver (wakeup capable).  See also <a href="#">Gpt_ModeType</a> .																				
<b>Parameters (inout):</b>	None																				
<b>Parameters (out):</b>	None																				
<b>Return value:</b>	None																				
<b>Description:</b>	Sets the operation mode of the GPT.																				

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
17	Function Definitions	 <a href="#">MCAL-6249</a> - <b>SWS_Gpt_00289 :</b> DisWakeups : Service API <b>PUBLISHED</b>	<p>[SWS_Gpt_00289] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Gpt_DisableWakeup</td></tr> <tr> <td><b>Syntax:</b></td><td>void Gpt_DisableWakeup(                           Gpt_ChannelType Channel                           )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x0a</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (but not for the same timer channel)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the GPT channel.</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Disables the wakeup interrupt of a channel (relevant in sleep mode).</td></tr> </table>	<b>Service name:</b>	Gpt_DisableWakeup	<b>Syntax:</b>	void Gpt_DisableWakeup( Gpt_ChannelType Channel )	<b>Service ID[hex]:</b>	0x0a	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)	<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Disables the wakeup interrupt of a channel (relevant in sleep mode).
<b>Service name:</b>	Gpt_DisableWakeup																						
<b>Syntax:</b>	void Gpt_DisableWakeup( Gpt_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x0a																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Disables the wakeup interrupt of a channel (relevant in sleep mode).																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
18	Function Definitions	 <a href="#">MCAL-6203</a> - <b>SWS_Gpt_00290</b> : EnWakeup : Service API <b>PUBLISHED</b>	<p>[SWS_Gpt_00290] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Gpt_EnableWakeup</td></tr> <tr> <td><b>Syntax:</b></td><td>void Gpt_EnableWakeup(                   Gpt_ChannelType Channel                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x0b</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (but not for the same timer channel)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the GPT channel.</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Enables the wakeup interrupt of a channel (relevant in sleep mode).</td></tr> </table>	<b>Service name:</b>	Gpt_EnableWakeup	<b>Syntax:</b>	void Gpt_EnableWakeup( Gpt_ChannelType Channel )	<b>Service ID[hex]:</b>	0x0b	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)	<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Enables the wakeup interrupt of a channel (relevant in sleep mode).
<b>Service name:</b>	Gpt_EnableWakeup																						
<b>Syntax:</b>	void Gpt_EnableWakeup( Gpt_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x0b																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (but not for the same timer channel)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the GPT channel.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Enables the wakeup interrupt of a channel (relevant in sleep mode).																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
19	Function Definitions	 <a href="#">MCAL-6207</a> - <b>SWS_Gpt_00328 :</b> CheckWakeup : Service API <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Gpt_00328] shall be implemented and follow <a href="#">AUTOSAR BSW GPT Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"><b>Service name:</b></td><td>Gpt_CheckWakeup</td></tr> <tr> <td><b>Syntax:</b></td><td>void Gpt_CheckWakeup(     EcuM_WakeupSourceType WakeupSource )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x0c</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>WakeupSource   Information on wakeup source to be checked. The associated GPT channel can be determined from configuration data.</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Checks if a wakeup capable GPT channel is the source for a wakeup event and calls the ECU state manager service EcuM_SetWakeupEvent in case of a valid GPT channel wakeup event.</td></tr> </table>	<b>Service name:</b>	Gpt_CheckWakeup	<b>Syntax:</b>	void Gpt_CheckWakeup( EcuM_WakeupSourceType WakeupSource )	<b>Service ID[hex]:</b>	0x0c	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	WakeupSource   Information on wakeup source to be checked. The associated GPT channel can be determined from configuration data.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Checks if a wakeup capable GPT channel is the source for a wakeup event and calls the ECU state manager service EcuM_SetWakeupEvent in case of a valid GPT channel wakeup event.
<b>Service name:</b>	Gpt_CheckWakeup																						
<b>Syntax:</b>	void Gpt_CheckWakeup( EcuM_WakeupSourceType WakeupSource )																						
<b>Service ID[hex]:</b>	0x0c																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	WakeupSource   Information on wakeup source to be checked. The associated GPT channel can be determined from configuration data.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Checks if a wakeup capable GPT channel is the source for a wakeup event and calls the ECU state manager service EcuM_SetWakeupEvent in case of a valid GPT channel wakeup event.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
20	Interfaces	 <a href="#">MCAL-7477 - GPT</a> Module: Interfaces <span style="background-color: #c8f7e4; border: 1px solid #2e6b2e; padding: 2px 5px; color: #2e6b2e;">PUBLISHED</span>	GPT module shall follow the interface definitions as described in section 8.7 of <a href="#">AUTOSAR BSW GPT Driver Specification</a> .
21	Sequence Diagrams	 <a href="#">MCAL-7478 - GPT</a> Module: Sequence Diagram <span style="background-color: #c8f7e4; border: 1px solid #2e6b2e; padding: 2px 5px; color: #2e6b2e;">PUBLISHED</span>	GPT module implementation shall follow that sequence diagram as described in <a href="#">AUTOSAR BSW GPT Driver Specification</a> section 9.
22	Configuration Specification	 <a href="#">MCAL-6216 - ECUC_Gpt_00336</a> : Gpt Configuration Container <span style="background-color: #c8f7e4; border: 1px solid #2e6b2e; padding: 2px 5px; color: #2e6b2e;">PUBLISHED</span>	GPT module shall follow the containers and configuration parameters as described in <a href="#">AUTOSAR BSW GPT Driver Specification</a> section 10.  ECUC_Gpt_00336 - VARIANT-PRE-COMPIL , VARAINT-POST-BUILD are the variants it can use.

## 8.6 SPI Driver

The The Multi-channel Serial Peripheral Interface (MCSPI) is a master/slave synchronous serial bus. It allows a master device to initiate serial data transfers to a slave device.

The MCSPI is a master synchronous serial bus module. This MCAL SPI driver supports McSPI interface.

The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW SPI Driver Specification](#).

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	Safety Process/ Diagnostic	 <a href="#">MCAL-6396 - MCAL Module : SPI : Safety Diagnostic</a> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	Following the Safety manual SPI IP and FMA review, the required safety diagnostic features should be added.
2	Constraints	 <a href="#">MCAL-7482 - MCAL Module: SPI implementation will have constraints as defined in AUTOSAR spec.</a> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00040] The SPI Handler/Driver handles only the Master mode. ()</p> <p>[SWS_Spi_00050] The SPI Handler/Driver only supports full-duplex mode. ()</p> <p>[SWS_Spi_00108] The LEVEL 2 SPI Handler/Driver is specified for microcontrollers that have to provide, at least, two SPI busses using separated hardware units.</p> <p>Otherwise, using this level of functionality does not make sense. ()</p>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
3	File Structure	 <a href="#">MCAL-6539 - SWS_Spi_00092 : File structure</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	SPI shall follow the file structure as defined in <a href="#">AUTOSAR BSW SPI Driver Specification</a> section 5.1 [SWS_Spi_00092]
4	Overall functionality and features	 <a href="#">MCAL-7483 - MCAL Module: SPI implementation shall support levels to have scalability.</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	<p>SPI Module shall support Levels to have scalability.</p> <p>Scalability of this monolithic SPI Handler/Driver allows getting a simple software module that fits simple needs up to a smart software module that fits enhanced needs.</p>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
5	General Behavior	 <a href="#">MCAL-7493</a> - MCAL Module: SPI shall follow the general behavior rules per AUTOSAR Spec. <a href="#">PUBLISHED</a>	<p>SPI shall follow the general behavior as defined in <a href="#">AUTOSAR BSW SPI Driver Specification</a> section 7.2.</p> <p>It shall have defined units such as Channels, Jobs, and sequences which shall be configurable.</p>
6	Channel Buffers	 <a href="#">MCAL-7484</a> - MCAL Module: SPI implementation shall support two types of channel buffers <a href="#">PUBLISHED</a>	<p>In order to allow taking advantages of all microcontroller capabilities but also to allow sending/receiving of data to/from a dedicated memory location, all levels have an optional feature with respect to the location of Channel Buffers.</p> <p>Hence, two main kinds of channel buffering can be used by configuration:</p> <ul style="list-style-type: none"> <li>Internally buffered Channels (IB): The buffer to transmit/receive data is provided by the Handler/Driver.</li> <li>Externally buffered Channels (EB): The buffer to transmit/receive is provided by the user (statically and/or dynamically).</li> </ul> <p>Both channel buffering methods may be used depending on the 3 use cases described below:</p> <ul style="list-style-type: none"> <li>Usage 0: the SPI Handler/Driver manages only Internal Buffers.</li> <li>Usage 1: the SPI Handler/Driver manages only External Buffers.</li> <li>Usage 2: the SPI Handler/Driver manages both buffers types.</li> </ul>

	<b>Specification</b>	<b>Architecture</b>	<b>Description</b>
	<b>Category</b>	<b>Identifier</b>	
7	Level 0	<a href="#">MCAL-7485</a> - Simple Synchronous Behavior	<p>MCAL Module: SPI The intention of this functionality level is to provide a Handler/Driver with a reduced implementation set of services to handle only simple synchronous transmissions. This is often the case for ECU including simple SPI networks but also for ECU using high speed external devices.</p> <p>Level 0 shall be simple synchronous behavior <b>PUBLISHED</b></p> <p>A simple synchronous transmission means that the function calling the transmission service is blocked during the ongoing transmission until the transmission is finished.</p>
8	Level 1	<a href="#">MCAL-7486</a> - Basic Asynchronous behavior	<p>MCAL Module: SPI The intention of this functionality level is to provide a Handler/Driver with a reduced implementation set of services to handle asynchronous transmissions only. This is often the case for ECU with functions related to Level 1 shall be Basic SPI networks having different priorities but also for</p> <p>Asynchronous behavior <b>PUBLISHED</b></p> <p>ECU using low speed external devices.</p> <p>An asynchronous transmission means that the user calling the transmission service is not blocked when the transmission is on-going. Furthermore, the user can be notified at the end of transmission1 .</p>

<b>Specification</b>	<b>Architecture</b>	<b>Description</b>
----------------------	---------------------	--------------------

	<b>Category</b>	<b>Identifier</b>	
9	Level 2	<p><a href="#">MCAL-7487</a> - Enhanced Behavior</p> <p>A MCAL Module: SPI shall be Enhanced Behavior</p> <p>PUBLISHED</p>	<p>The intention of this functionality level is to provide a Handler/Driver with a complete set of services to handle synchronous and asynchronous transmissions. This could Level 2 be the case for ECU with a lot of functions related to SPI networks having different priorities but also for ECU using external devices with different speeds.</p> <p>Handling asynchronous and synchronous transmissions means that the microcontroller for which this software module is dedicated has to provide more than one SPI bus (see [SWS_Spi_00108]). In fact, the goal is to support SPI buses using a so-called synchronous driver and to support other SPI buses using a so-called asynchronous driver.</p>
10	Error	<p><a href="#">MCAL-7488</a> - A SPI module shall have Development Errors, Extended Production Errors and Runtime Errors as defined in section 7.4 Classification</p> <p>Module: SPI of <a href="#">AUTOSAR BSW SPI Driver Specification</a>. It shall use Det_ReportRuntimeError, Det_ReportError and Dem_SetEventStatus interfaces to report errors. (SWS_Spi_00389, SWS_Spi_00339) implementation</p> <p>shall have Error Classification</p> <p>PUBLISHED</p>	MCAL

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
11	Error Detection	 <a href="#">MCAL-7489</a> - MCAL Module: SPI implementation shall have Error Detection <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	SPI module shall have API Parameter checking and SPI state checking, as defined in section 7.4 of <a href="#">AUTOSAR BSW SPI Driver Specification</a> .
12	Imported Types	 <a href="#">MCAL-6646</a> - <a href="#">SWS_Spi_00174</a> : Imported types <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	[SWS_Spi_00174] GPT module shall use imported types Dem and Std_Types.
13	Type Definitions	 <a href="#">MCAL-7490</a> - MCAL Module: SPI Type Definitions <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	SPI module shall define types as described in section 8.2 of <a href="#">AUTOSAR BSW SPI Driver Specification</a> .

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																					
14	Function Definitions	 <b>MCAL-6495 - SWS_Spi_00175 : Spi_Init Functional</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00175] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_Init</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Spi_Init(                   const Spi_ConfigType* ConfigPtr                   )</td> </tr> <tr> <td><b>Service ID/hex:</b></td> <td>0x00</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ConfigPtr</td> <td>Pointer to configuration set</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Service for SPI initialization.</td> </tr> </table>	<b>Service name:</b>	Spi_Init	<b>Syntax:</b>	void Spi_Init( const Spi_ConfigType* ConfigPtr )	<b>Service ID/hex:</b>	0x00	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	ConfigPtr	Pointer to configuration set	<b>Parameters (in-out):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Service for SPI initialization.
<b>Service name:</b>	Spi_Init																							
<b>Syntax:</b>	void Spi_Init( const Spi_ConfigType* ConfigPtr )																							
<b>Service ID/hex:</b>	0x00																							
<b>Sync/Async:</b>	Synchronous																							
<b>Reentrancy:</b>	Non Reentrant																							
<b>Parameters (in):</b>	ConfigPtr	Pointer to configuration set																						
<b>Parameters (in-out):</b>	None																							
<b>Parameters (out):</b>	None																							
<b>Return value:</b>	None																							
<b>Description:</b>	Service for SPI initialization.																							

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																						
15	Function Definitions	 <a href="#">MCAL-6513 - SWS_Spi_00176 : Spi_Delnit Functional</a> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00176] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_Delnit</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Std_ReturnType Spi_DeInit(     void )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x01</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: de-initialisation command has been accepted E_NOT_OK: de-initialisation command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service for SPI de-initialization.</td> </tr> </table>	<b>Service name:</b>	Spi_Delnit	<b>Syntax:</b>	Std_ReturnType Spi_DeInit( void )	<b>Service ID[hex]:</b>	0x01	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (in-out):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Std_ReturnType	E_OK: de-initialisation command has been accepted E_NOT_OK: de-initialisation command has not been accepted	<b>Description:</b>	Service for SPI de-initialization.	
<b>Service name:</b>	Spi_Delnit																								
<b>Syntax:</b>	Std_ReturnType Spi_DeInit( void )																								
<b>Service ID[hex]:</b>	0x01																								
<b>Sync/Async:</b>	Synchronous																								
<b>Reentrancy:</b>	Non Reentrant																								
<b>Parameters (in):</b>	None																								
<b>Parameters (in-out):</b>	None																								
<b>Parameters (out):</b>	None																								
<b>Return value:</b>	Std_ReturnType	E_OK: de-initialisation command has been accepted E_NOT_OK: de-initialisation command has not been accepted																							
<b>Description:</b>	Service for SPI de-initialization.																								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																																	
16	Function Definitions	 <b>MCAL-6547 - SWS_Spi_00177 :</b> <b>Spi_WriteIB</b> Functional <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00177] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Spi_WriteIB</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">Std_ReturnType Spi_WriteIB(     Spi_ChannelType Channel,     const Spi_DataBufferType* DataBufferPtr )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x02</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel</td> <td>Channel ID.</td> </tr> <tr> <td></td> <td>DataBufferPtr</td> <td>Pointer to source data buffer. If this pointer is null, it is assumed that the data to be transmitted is not relevant and the default transmit value of this channel will be used instead.</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: write command has been accepted E_NOT_OK: write command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service for writing one or more data to an IB SPI Handler/Driver Channel specified by parameter.</td> </tr> </table>	<b>Service name:</b>	Spi_WriteIB		<b>Syntax:</b>	Std_ReturnType Spi_WriteIB( Spi_ChannelType Channel, const Spi_DataBufferType* DataBufferPtr )		<b>Service ID[hex]:</b>	0x02		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant		<b>Parameters (in):</b>	Channel	Channel ID.		DataBufferPtr	Pointer to source data buffer. If this pointer is null, it is assumed that the data to be transmitted is not relevant and the default transmit value of this channel will be used instead.	<b>Parameters (in-out):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Std_ReturnType	E_OK: write command has been accepted E_NOT_OK: write command has not been accepted	<b>Description:</b>	Service for writing one or more data to an IB SPI Handler/Driver Channel specified by parameter.	
<b>Service name:</b>	Spi_WriteIB																																			
<b>Syntax:</b>	Std_ReturnType Spi_WriteIB( Spi_ChannelType Channel, const Spi_DataBufferType* DataBufferPtr )																																			
<b>Service ID[hex]:</b>	0x02																																			
<b>Sync/Async:</b>	Synchronous																																			
<b>Reentrancy:</b>	Reentrant																																			
<b>Parameters (in):</b>	Channel	Channel ID.																																		
	DataBufferPtr	Pointer to source data buffer. If this pointer is null, it is assumed that the data to be transmitted is not relevant and the default transmit value of this channel will be used instead.																																		
<b>Parameters (in-out):</b>	None																																			
<b>Parameters (out):</b>	None																																			
<b>Return value:</b>	Std_ReturnType	E_OK: write command has been accepted E_NOT_OK: write command has not been accepted																																		
<b>Description:</b>	Service for writing one or more data to an IB SPI Handler/Driver Channel specified by parameter.																																			

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																									
17	Function Definitions	 <b>MCAL-6622 - SWS_Spi_00178 : Spi_AsyncTransmit Functional</b> <span style="background-color: #d9ffd9; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00178] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_AsyncTransmit</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Std_ReturnType Spi_AsyncTransmit(                           Spi_SequenceType Sequence                           )</td> </tr> <tr> <td><b>Service ID/hex:</b></td> <td>0x03</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Asynchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Sequence</td> <td>Sequence ID.</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: Transmission command has been accepted E_NOT_OK: Transmission command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service to transmit data on the SPI bus.</td> </tr> </table>	<b>Service name:</b>	Spi_AsyncTransmit	<b>Syntax:</b>	Std_ReturnType Spi_AsyncTransmit( Spi_SequenceType Sequence )	<b>Service ID/hex:</b>	0x03	<b>Sync/Async:</b>	Asynchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Sequence	Sequence ID.	<b>Parameters (in-out):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Std_ReturnType	E_OK: Transmission command has been accepted E_NOT_OK: Transmission command has not been accepted	<b>Description:</b>	Service to transmit data on the SPI bus.	
<b>Service name:</b>	Spi_AsyncTransmit																											
<b>Syntax:</b>	Std_ReturnType Spi_AsyncTransmit( Spi_SequenceType Sequence )																											
<b>Service ID/hex:</b>	0x03																											
<b>Sync/Async:</b>	Asynchronous																											
<b>Reentrancy:</b>	Reentrant																											
<b>Parameters (in):</b>	Sequence	Sequence ID.																										
<b>Parameters (in-out):</b>	None																											
<b>Parameters (out):</b>	None																											
<b>Return value:</b>	Std_ReturnType	E_OK: Transmission command has been accepted E_NOT_OK: Transmission command has not been accepted																										
<b>Description:</b>	Service to transmit data on the SPI bus.																											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																									
18	Function Definitions	 <b>MCAL-6736 - SWS_Spi_00179 :</b> <b>Spi_ReadIB</b> Functional <span style="background-color: #d9ffd9; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00179] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_ReadIB</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Std_ReturnType Spi_ReadIB(                   Spi_ChannelType Channel,                   Spi_DataBufferType* DataBufferPointer                 )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x04</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel</td> <td>Channel ID.</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>DataBufferPointer</td> <td>Pointer to destination data buffer in RAM</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: read command has been accepted E_NOT_OK: read command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service for reading synchronously one or more data from an IB SPI Handler/Driver Channel specified by parameter.</td> </tr> </table>	<b>Service name:</b>	Spi_ReadIB	<b>Syntax:</b>	Std_ReturnType Spi_ReadIB( Spi_ChannelType Channel, Spi_DataBufferType* DataBufferPointer )	<b>Service ID[hex]:</b>	0x04	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Channel	Channel ID.	<b>Parameters (in-out):</b>	None		<b>Parameters (out):</b>	DataBufferPointer	Pointer to destination data buffer in RAM	<b>Return value:</b>	Std_ReturnType	E_OK: read command has been accepted E_NOT_OK: read command has not been accepted	<b>Description:</b>	Service for reading synchronously one or more data from an IB SPI Handler/Driver Channel specified by parameter.	
<b>Service name:</b>	Spi_ReadIB																											
<b>Syntax:</b>	Std_ReturnType Spi_ReadIB( Spi_ChannelType Channel, Spi_DataBufferType* DataBufferPointer )																											
<b>Service ID[hex]:</b>	0x04																											
<b>Sync/Async:</b>	Synchronous																											
<b>Reentrancy:</b>	Reentrant																											
<b>Parameters (in):</b>	Channel	Channel ID.																										
<b>Parameters (in-out):</b>	None																											
<b>Parameters (out):</b>	DataBufferPointer	Pointer to destination data buffer in RAM																										
<b>Return value:</b>	Std_ReturnType	E_OK: read command has been accepted E_NOT_OK: read command has not been accepted																										
<b>Description:</b>	Service for reading synchronously one or more data from an IB SPI Handler/Driver Channel specified by parameter.																											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																											
19	Function Definitions	 <b>MCAL-6418 - SWS_Spi_00180 : Spi_SetupEB</b> Functional <span style="background-color: #d9e1f2; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Spi_00180] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_SetupEB</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Std_ReturnType Spi_SetupEB(     Spi_ChannelType Channel,     const Spi_DataBufferType* SrcDataBufferPtr,     Spi_DataBufferType* DesDataBufferPtr,     Spi_NumberOfDataType Length )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x05</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Channel</td> <td>Channel ID.</td> </tr> <tr> <td>SrcDataBufferPtr</td> <td>Pointer to source data buffer.</td> </tr> <tr> <td>Length</td> <td>Length (number of data elements) of the data to be transmitted from SrcDataBufferPtr and/or received from DesDataBufferPtr Min.: 1 Max.: Max of data specified at configuration for this channel</td> </tr> </table> </td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td>DesDataBufferPtr</td> <td>Pointer to destination data buffer in RAM.</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: Setup command has been accepted E_NOT_OK: Setup command has not been accepted</td> </tr> </table>	<b>Service name:</b>	Spi_SetupEB	<b>Syntax:</b>	Std_ReturnType Spi_SetupEB( Spi_ChannelType Channel, const Spi_DataBufferType* SrcDataBufferPtr, Spi_DataBufferType* DesDataBufferPtr, Spi_NumberOfDataType Length )	<b>Service ID[hex]:</b>	0x05	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Channel</td> <td>Channel ID.</td> </tr> <tr> <td>SrcDataBufferPtr</td> <td>Pointer to source data buffer.</td> </tr> <tr> <td>Length</td> <td>Length (number of data elements) of the data to be transmitted from SrcDataBufferPtr and/or received from DesDataBufferPtr Min.: 1 Max.: Max of data specified at configuration for this channel</td> </tr> </table>	Channel	Channel ID.	SrcDataBufferPtr	Pointer to source data buffer.	Length	Length (number of data elements) of the data to be transmitted from SrcDataBufferPtr and/or received from DesDataBufferPtr Min.: 1 Max.: Max of data specified at configuration for this channel	<b>Parameters (in-out):</b>	DesDataBufferPtr	Pointer to destination data buffer in RAM.	<b>Parameters (out):</b>	None		<b>Return value:</b>	Std_ReturnType	E_OK: Setup command has been accepted E_NOT_OK: Setup command has not been accepted
<b>Service name:</b>	Spi_SetupEB																													
<b>Syntax:</b>	Std_ReturnType Spi_SetupEB( Spi_ChannelType Channel, const Spi_DataBufferType* SrcDataBufferPtr, Spi_DataBufferType* DesDataBufferPtr, Spi_NumberOfDataType Length )																													
<b>Service ID[hex]:</b>	0x05																													
<b>Sync/Async:</b>	Synchronous																													
<b>Reentrancy:</b>	Reentrant																													
<b>Parameters (in):</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Channel</td> <td>Channel ID.</td> </tr> <tr> <td>SrcDataBufferPtr</td> <td>Pointer to source data buffer.</td> </tr> <tr> <td>Length</td> <td>Length (number of data elements) of the data to be transmitted from SrcDataBufferPtr and/or received from DesDataBufferPtr Min.: 1 Max.: Max of data specified at configuration for this channel</td> </tr> </table>	Channel	Channel ID.	SrcDataBufferPtr	Pointer to source data buffer.	Length	Length (number of data elements) of the data to be transmitted from SrcDataBufferPtr and/or received from DesDataBufferPtr Min.: 1 Max.: Max of data specified at configuration for this channel																							
Channel	Channel ID.																													
SrcDataBufferPtr	Pointer to source data buffer.																													
Length	Length (number of data elements) of the data to be transmitted from SrcDataBufferPtr and/or received from DesDataBufferPtr Min.: 1 Max.: Max of data specified at configuration for this channel																													
<b>Parameters (in-out):</b>	DesDataBufferPtr	Pointer to destination data buffer in RAM.																												
<b>Parameters (out):</b>	None																													
<b>Return value:</b>	Std_ReturnType	E_OK: Setup command has been accepted E_NOT_OK: Setup command has not been accepted																												

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																						
			<p><b>Description:</b> Service to setup the buffers and the length of data for the EB SPI Handler/Driver Channel specified.</p>																						
20	Function Definitions	 MCAL-6393 - <b>SWS_Spi_00181 :</b> Spi_GetStatus Functional <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00181] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_GetStatus</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Spi_StatusType Spi_GetStatus ( void )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x06</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Spi_StatusType</td> <td>Spi_StatusType</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service returns the SPI Handler/Driver software module status.</td></tr> </table>	<b>Service name:</b>	Spi_GetStatus	<b>Syntax:</b>	Spi_StatusType Spi_GetStatus ( void )	<b>Service ID[hex]:</b>	0x06	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (in-out):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Spi_StatusType	Spi_StatusType	<b>Description:</b>	Service returns the SPI Handler/Driver software module status.	
<b>Service name:</b>	Spi_GetStatus																								
<b>Syntax:</b>	Spi_StatusType Spi_GetStatus ( void )																								
<b>Service ID[hex]:</b>	0x06																								
<b>Sync/Async:</b>	Synchronous																								
<b>Reentrancy:</b>	Reentrant																								
<b>Parameters (in):</b>	None																								
<b>Parameters (in-out):</b>	None																								
<b>Parameters (out):</b>	None																								
<b>Return value:</b>	Spi_StatusType	Spi_StatusType																							
<b>Description:</b>	Service returns the SPI Handler/Driver software module status.																								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																									
21	Function Definitions	 <b>MCAL-6679 - SWS_Spi_00182 : Spi_GetJobResult Functional</b> <span style="background-color: #d9e1f2; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Spi_00182] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_GetJobResult</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Spi_JobResultType Spi_GetJobResult(     Spi_JobType Job )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x07</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Job</td> <td>Job ID. An invalid job ID will return an undefined result.</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td>None</td> <td></td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> <td></td> </tr> <tr> <td><b>Return value:</b></td> <td>Spi_JobResultType</td> <td>Spi_JobResultType</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">This service returns the last transmission result of the specified Job.</td></tr> </table>	<b>Service name:</b>	Spi_GetJobResult	<b>Syntax:</b>	Spi_JobResultType Spi_GetJobResult( Spi_JobType Job )	<b>Service ID[hex]:</b>	0x07	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Job	Job ID. An invalid job ID will return an undefined result.	<b>Parameters (in-out):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Spi_JobResultType	Spi_JobResultType	<b>Description:</b>	This service returns the last transmission result of the specified Job.	
<b>Service name:</b>	Spi_GetJobResult																											
<b>Syntax:</b>	Spi_JobResultType Spi_GetJobResult( Spi_JobType Job )																											
<b>Service ID[hex]:</b>	0x07																											
<b>Sync/Async:</b>	Synchronous																											
<b>Reentrancy:</b>	Reentrant																											
<b>Parameters (in):</b>	Job	Job ID. An invalid job ID will return an undefined result.																										
<b>Parameters (in-out):</b>	None																											
<b>Parameters (out):</b>	None																											
<b>Return value:</b>	Spi_JobResultType	Spi_JobResultType																										
<b>Description:</b>	This service returns the last transmission result of the specified Job.																											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																														
22	Function Definitions	 <b>MCAL-6463 - SWS_Spi_00183 : Spi_GetSequenceResult Functional</b> <span style="background-color: #d9e1f2; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Spi_00183] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Spi_GetSequenceResult</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">Spi_SeqResultType Spi_GetSequenceResult(                           Spi_SequenceType Sequence                           )</td> </tr> <tr> <td><b>Service ID(hex):</b></td> <td colspan="2">0x08</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Sequence</td> <td>Sequence ID. An invalid sequence ID will return an undefined result.</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Spi_SeqResultType</td> <td>Spi_SeqResultType</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">This service returns the last transmission result of the specified Sequence.</td> </tr> </table>	<b>Service name:</b>	Spi_GetSequenceResult		<b>Syntax:</b>	Spi_SeqResultType Spi_GetSequenceResult( Spi_SequenceType Sequence )		<b>Service ID(hex):</b>	0x08		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant		<b>Parameters (in):</b>	Sequence	Sequence ID. An invalid sequence ID will return an undefined result.	<b>Parameters (in-out):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Spi_SeqResultType	Spi_SeqResultType	<b>Description:</b>	This service returns the last transmission result of the specified Sequence.	
<b>Service name:</b>	Spi_GetSequenceResult																																
<b>Syntax:</b>	Spi_SeqResultType Spi_GetSequenceResult( Spi_SequenceType Sequence )																																
<b>Service ID(hex):</b>	0x08																																
<b>Sync/Async:</b>	Synchronous																																
<b>Reentrancy:</b>	Reentrant																																
<b>Parameters (in):</b>	Sequence	Sequence ID. An invalid sequence ID will return an undefined result.																															
<b>Parameters (in-out):</b>	None																																
<b>Parameters (out):</b>	None																																
<b>Return value:</b>	Spi_SeqResultType	Spi_SeqResultType																															
<b>Description:</b>	This service returns the last transmission result of the specified Sequence.																																

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
23	Function Definitions	 <b>MCAL-6661 - SWS_Spi_00184 : Spi_GetVersionInfo</b> Functional <span style="background-color: #d9e1f2; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00184] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_GetVersionInfo</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Spi_GetVersionInfo(                   Std_VersionInfoType* versioninfo                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x09</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>versioninfo   Pointer to where to store the version information of this module.</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>This service returns the version information of this module.</td> </tr> </table>	<b>Service name:</b>	Spi_GetVersionInfo	<b>Syntax:</b>	void Spi_GetVersionInfo( Std_VersionInfoType* versioninfo )	<b>Service ID[hex]:</b>	0x09	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (in-out):</b>	None	<b>Parameters (out):</b>	versioninfo   Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	This service returns the version information of this module.
<b>Service name:</b>	Spi_GetVersionInfo																						
<b>Syntax:</b>	void Spi_GetVersionInfo( Std_VersionInfoType* versioninfo )																						
<b>Service ID[hex]:</b>	0x09																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (in-out):</b>	None																						
<b>Parameters (out):</b>	versioninfo   Pointer to where to store the version information of this module.																						
<b>Return value:</b>	None																						
<b>Description:</b>	This service returns the version information of this module.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																									
24	Function Definitions	 <b>MCAL-6565 - SWS_Spi_00185 : Spi_SyncTransmit Functional</b> <span style="background-color: #d9ffd9; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00185] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_SyncTransmit</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Std_ReturnType Spi_SyncTransmit(     Spi_SequenceType Sequence )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x0a</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Asynchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Sequence</td> <td>Sequence ID.</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: Transmission command has been accepted E_NOT_OK: Transmission command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service to transmit data on the SPI bus</td> </tr> </table>	<b>Service name:</b>	Spi_SyncTransmit	<b>Syntax:</b>	Std_ReturnType Spi_SyncTransmit( Spi_SequenceType Sequence )	<b>Service ID[hex]:</b>	0x0a	<b>Sync/Async:</b>	Asynchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Sequence	Sequence ID.	<b>Parameters (in-out):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Std_ReturnType	E_OK: Transmission command has been accepted E_NOT_OK: Transmission command has not been accepted	<b>Description:</b>	Service to transmit data on the SPI bus	
<b>Service name:</b>	Spi_SyncTransmit																											
<b>Syntax:</b>	Std_ReturnType Spi_SyncTransmit( Spi_SequenceType Sequence )																											
<b>Service ID[hex]:</b>	0x0a																											
<b>Sync/Async:</b>	Asynchronous																											
<b>Reentrancy:</b>	Reentrant																											
<b>Parameters (in):</b>	Sequence	Sequence ID.																										
<b>Parameters (in-out):</b>	None																											
<b>Parameters (out):</b>	None																											
<b>Return value:</b>	Std_ReturnType	E_OK: Transmission command has been accepted E_NOT_OK: Transmission command has not been accepted																										
<b>Description:</b>	Service to transmit data on the SPI bus																											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																									
25	Function Definitions	 <a href="#">MCAL-6540 - SWS_Spi_00186 : Spi_GetHWUnitStatus Functional</a> <span style="background-color: #d9e1f2; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Spi_00186] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_GetHWUnitStatus</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Spi_StatusType Spi_GetHWUnitStatus(     Spi_HWUnitType HWUnit )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x0b</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>HWUnit</td> <td>SPI Hardware microcontroller peripheral (unit) ID.</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Spi_StatusType</td> <td>Spi_StatusType</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">This service returns the status of the specified SPI Hardware microcontroller peripheral.</td> </tr> </table>	<b>Service name:</b>	Spi_GetHWUnitStatus	<b>Syntax:</b>	Spi_StatusType Spi_GetHWUnitStatus( Spi_HWUnitType HWUnit )	<b>Service ID[hex]:</b>	0x0b	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	HWUnit	SPI Hardware microcontroller peripheral (unit) ID.	<b>Parameters (in-out):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Spi_StatusType	Spi_StatusType	<b>Description:</b>	This service returns the status of the specified SPI Hardware microcontroller peripheral.	
<b>Service name:</b>	Spi_GetHWUnitStatus																											
<b>Syntax:</b>	Spi_StatusType Spi_GetHWUnitStatus( Spi_HWUnitType HWUnit )																											
<b>Service ID[hex]:</b>	0x0b																											
<b>Sync/Async:</b>	Synchronous																											
<b>Reentrancy:</b>	Reentrant																											
<b>Parameters (in):</b>	HWUnit	SPI Hardware microcontroller peripheral (unit) ID.																										
<b>Parameters (in-out):</b>	None																											
<b>Parameters (out):</b>	None																											
<b>Return value:</b>	Spi_StatusType	Spi_StatusType																										
<b>Description:</b>	This service returns the status of the specified SPI Hardware microcontroller peripheral.																											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																									
26	Function Definitions	 <b>MCAL-6497 - SWS_Spi_00187 :</b> Spi_Cancel Functional <span style="background-color: #d9ffd9; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00187] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Spi_Cancel</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Spi_Cancel(                   Spi_SequenceType Sequence                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x0c</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Asynchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Sequence</td> <td>Sequence ID.</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td>None</td> <td></td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> <td></td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> <td></td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service cancels the specified on-going sequence transmission.</td> </tr> </table>	<b>Service name:</b>	Spi_Cancel	<b>Syntax:</b>	void Spi_Cancel( Spi_SequenceType Sequence )	<b>Service ID[hex]:</b>	0x0c	<b>Sync/Async:</b>	Asynchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	Sequence	Sequence ID.	<b>Parameters (in-out):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Service cancels the specified on-going sequence transmission.	
<b>Service name:</b>	Spi_Cancel																											
<b>Syntax:</b>	void Spi_Cancel( Spi_SequenceType Sequence )																											
<b>Service ID[hex]:</b>	0x0c																											
<b>Sync/Async:</b>	Asynchronous																											
<b>Reentrancy:</b>	Reentrant																											
<b>Parameters (in):</b>	Sequence	Sequence ID.																										
<b>Parameters (in-out):</b>	None																											
<b>Parameters (out):</b>	None																											
<b>Return value:</b>	None																											
<b>Description:</b>	Service cancels the specified on-going sequence transmission.																											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																														
27	Function Definitions	 <b>MCAL-6499 - SWS_Spi_00188 : Spi_SetAsyncMode</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00188] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Spi_SetAsyncMode</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">Std_ReturnType Spi_SetAsyncMode ( Spi_AsyncModeType Mode )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x0d</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Mode</td> <td>New mode required.</td> </tr> <tr> <td><b>Parameters (in-out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: Setting command has been done E_NOT_OK: setting command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service to set the asynchronous mechanism mode for SPI busses handled asynchronously.</td> </tr> </table>	<b>Service name:</b>	Spi_SetAsyncMode		<b>Syntax:</b>	Std_ReturnType Spi_SetAsyncMode ( Spi_AsyncModeType Mode )		<b>Service ID[hex]:</b>	0x0d		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Non Reentrant		<b>Parameters (in):</b>	Mode	New mode required.	<b>Parameters (in-out):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Std_ReturnType	E_OK: Setting command has been done E_NOT_OK: setting command has not been accepted	<b>Description:</b>	Service to set the asynchronous mechanism mode for SPI busses handled asynchronously.	
<b>Service name:</b>	Spi_SetAsyncMode																																
<b>Syntax:</b>	Std_ReturnType Spi_SetAsyncMode ( Spi_AsyncModeType Mode )																																
<b>Service ID[hex]:</b>	0x0d																																
<b>Sync/Async:</b>	Synchronous																																
<b>Reentrancy:</b>	Non Reentrant																																
<b>Parameters (in):</b>	Mode	New mode required.																															
<b>Parameters (in-out):</b>	None																																
<b>Parameters (out):</b>	None																																
<b>Return value:</b>	Std_ReturnType	E_OK: Setting command has been done E_NOT_OK: setting command has not been accepted																															
<b>Description:</b>	Service to set the asynchronous mechanism mode for SPI busses handled asynchronously.																																

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>												
28	Function Definitions	 <b>MCAL-6578 - SWS_Spi_00189 : Spi_MainFunction_H andling Functional</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Spi_00189] shall be implemented and follow <a href="#">AUTOSAR BSW SPI Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;"><b>Service name:</b></td><td>Spi_MainFunction_Handling</td></tr> <tr> <td><b>Syntax:</b></td><td>void Spi_MainFunction_Handling(</td></tr> <tr> <td></td><td>                    void</td></tr> <tr> <td></td><td>                    )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x10</td></tr> <tr> <td><b>Description:</b></td><td>--</td></tr> </table>	<b>Service name:</b>	Spi_MainFunction_Handling	<b>Syntax:</b>	void Spi_MainFunction_Handling(		void		)	<b>Service ID[hex]:</b>	0x10	<b>Description:</b>	--
<b>Service name:</b>	Spi_MainFunction_Handling														
<b>Syntax:</b>	void Spi_MainFunction_Handling(														
	void														
	)														
<b>Service ID[hex]:</b>	0x10														
<b>Description:</b>	--														
29	Configurable Interfaces	 <b>MCAL-7491 - MCAL Module: SPI Configuration Interfaces</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>SPI Module will use target functions which can be configured. The target function is usually a call-back function. The name of these interfaces is not fixed because they are configurable. Callback prototypes are provided by callback header files included by Spi_Cfg.h files. SPI module shall make sure of Spi_JobDoneNotification and Spi_SeqEndNotification functions.</p>												

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
30	Sequence Diagrams	 <a href="#">MCAL-7492 - MCAL Module: SPI Sequence Diagram</a> <span style="background-color: #e0f2e0; border: 1px solid #d0e0d0; padding: 2px;">PUBLISHED</span>	SPI module implementation shall follow that sequence diagram as described in <a href="#">AUTOSAR BSW SPI Driver Specification</a> section 9.
31	Configuration Specification	 <a href="#">MCAL-6744 - ECUC_Spi_00103 : Module Spi</a> <span style="background-color: #e0f2e0; border: 1px solid #d0e0d0; padding: 2px;">PUBLISHED</span>	SPI module shall follow the containers and configuration parameters as described in <a href="#">AUTOSAR BSW SPI Driver Specification</a> section 10.  ECUC_Spi_00103- VARIANT-PRE-COMPIL, VARAINT-POST-BUILD, VARAINT-LINK-TIME are the variants it can use.

## 8.7 PWM Driver

The PWM module implements an interface in C programming language for handling the PWM functionality of the device. This PWM driver takes care of initializing and deinitializing the PWM unit and offers services to:

- Generate pulses with variable pulse width(Duty Cycle - can range from 0% to 100%)
- Set parameters of a PWM channels waveform(Duty Cycle and Period)
- Enable/Disable notifications

The PWM module initializes, configures and controls the internal hardware to realize PWM driver as detailed in AUTOSAR BSW PWM Driver Specification. The PWM functionality is realized through the DM Timers.

The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW PWM Driver Specification](#).

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	Safety Process/ Diagnostic	 <a href="#">MCAL-6881</a> - MCAL Module : PWM : Safety Diagnostic <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	Following the Safety manual Timer IP and FMA review, the required safety diagnostic features should be added.
2	File Structure	 <a href="#">MCAL-7494</a> - Module PWM: File Structure <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	PWM shall follow the file structure as defined in <a href="#">AUTOSAR BSW PWM Driver Specification</a> section 5.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
3	General Behavior	 <b>MCAL-7495</b> - Module PWM: General Behavior <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Pwm_00088] All functions from the PWM module except Pwm_Init, Pwm_DeInit and Pwm_GetVersionInfo shall be re-entrant for different PWM channel numbers.</p> <p>In order to keep a simple module implementation, no check of SWS_Pwm_00088 must be performed by the module. ()</p> <p>[SWS_Pwm_00089] The Pwm module's user shall ensure the integrity if several function calls are made during run time in different tasks or ISRs for the same PWM channel.</p> <p>[SWS_Pwm_00070] All time units used within the API services of the PWM module shall be of the unit ticks. (SRS_BSW_00343)</p>
4	Error Classification and Detection	 <b>MCAL-7496</b> - Module PWM: Error Classification and Detection <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>PWM module shall have development Errors and Runtime Errors as defined in section 7.4 and 8.7 of <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <p>SWS_Pwm_00104 - Det_ReportError shall be used to report development errors.</p> <p>API Parameter checking shall also be employed.</p>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
5	Duty Cycle Resolution and Scaling	 <a href="#">MCAL-7497</a> - Module PWM: Duty Cycle Resolution and Scaling <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>PWM module shall follow the Duty Cycle Resolution and Scaling guidelines as defined in section 7.7 of <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <p>As an implementation guide, the following source code example is given:</p> <pre>AbsoluteDutyCycle = ((uint32)AbsolutePeriodTime * RelativeDutyCycle) &gt;&gt; 15;</pre>
6	Imported Types	 <a href="#">MCAL-6753</a> - SWS_Pwm_000 94 : PWM Imported Types Dem <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	[SWS_Pwm_00094] PWM module shall use imported types Std_Types.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
7	Type Definitions	 <a href="#">MCAL-7498</a> - Module PWM: Type Definitions <span>PUBLISHED</span>	PWM module shall define types as described in section 8.2 of <a href="#">AUTOSAR BSW PWM Driver Specification</a> .

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
8	Function Definitions	 MCAL-6867 - SWS_Pwm_000 95 : Pwm_Init API <span style="background-color: #e0f2e0; padding: 2px 5px;">PUBLISHED</span>	<p>[SWS_Pwm_00095] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td> <td>Pwm_Init</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Pwm_Init(                   const Pwm_ConfigType* ConfigPtr                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x00</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ConfigPtr      Pointer to configuration set</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Service for PWM initialization.</td> </tr> </table>	<b>Service name:</b>	Pwm_Init	<b>Syntax:</b>	void Pwm_Init( const Pwm_ConfigType* ConfigPtr )	<b>Service ID[hex]:</b>	0x00	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	ConfigPtr      Pointer to configuration set	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Service for PWM initialization.
<b>Service name:</b>	Pwm_Init																						
<b>Syntax:</b>	void Pwm_Init( const Pwm_ConfigType* ConfigPtr )																						
<b>Service ID[hex]:</b>	0x00																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	ConfigPtr      Pointer to configuration set																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Service for PWM initialization.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
9	Function Definitions	 MCAL-6816 SWS_Pwm_000 96 : Pwm_Delnit API Service Description <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Pwm_00096] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Pwm_Delnit</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Pwm_DeInit(                 void               )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x01</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Service for PWM De-Initialization.</td> </tr> </table>	<b>Service name:</b>	Pwm_Delnit	<b>Syntax:</b>	void Pwm_DeInit( void )	<b>Service ID[hex]:</b>	0x01	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Service for PWM De-Initialization.
<b>Service name:</b>	Pwm_Delnit																						
<b>Syntax:</b>	void Pwm_DeInit( void )																						
<b>Service ID[hex]:</b>	0x01																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Service for PWM De-Initialization.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																								
10	Function Definitions	 MCAL-6803 SWS_Pwm_000 97: Pwm_SetDutyCycle API Service Description <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Pwm_00097] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Pwm_SetDutyCycle</td> </tr> <tr> <td><b>Syntax:</b></td> <td> <pre>void Pwm_SetDutyCycle(     Pwm_ChannelType ChannelNumber,     uint16 DutyCycle )</pre> </td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x02</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant for different channel numbers</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>ChannelNumber</td> <td>Numeric identifier of the PWM</td> </tr> <tr> <td>DutyCycle</td> <td>Min=0x0000 Max=0x8000</td> </tr> </table> </td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Service sets the duty cycle of the PWM channel.</td> </tr> </table>	<b>Service name:</b>	Pwm_SetDutyCycle	<b>Syntax:</b>	<pre>void Pwm_SetDutyCycle(     Pwm_ChannelType ChannelNumber,     uint16 DutyCycle )</pre>	<b>Service ID[hex]:</b>	0x02	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant for different channel numbers	<b>Parameters (in):</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>ChannelNumber</td> <td>Numeric identifier of the PWM</td> </tr> <tr> <td>DutyCycle</td> <td>Min=0x0000 Max=0x8000</td> </tr> </table>	ChannelNumber	Numeric identifier of the PWM	DutyCycle	Min=0x0000 Max=0x8000	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Service sets the duty cycle of the PWM channel.
<b>Service name:</b>	Pwm_SetDutyCycle																										
<b>Syntax:</b>	<pre>void Pwm_SetDutyCycle(     Pwm_ChannelType ChannelNumber,     uint16 DutyCycle )</pre>																										
<b>Service ID[hex]:</b>	0x02																										
<b>Sync/Async:</b>	Synchronous																										
<b>Reentrancy:</b>	Reentrant for different channel numbers																										
<b>Parameters (in):</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>ChannelNumber</td> <td>Numeric identifier of the PWM</td> </tr> <tr> <td>DutyCycle</td> <td>Min=0x0000 Max=0x8000</td> </tr> </table>	ChannelNumber	Numeric identifier of the PWM	DutyCycle	Min=0x0000 Max=0x8000																						
ChannelNumber	Numeric identifier of the PWM																										
DutyCycle	Min=0x0000 Max=0x8000																										
<b>Parameters (inout):</b>	None																										
<b>Parameters (out):</b>	None																										
<b>Return value:</b>	None																										
<b>Description:</b>	Service sets the duty cycle of the PWM channel.																										

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																																				
11	Function Definitions	 <b>MCAL-6795</b> SWS_Pwm_000 98: Pwm_SetPeriodAndDuty API Service Description <b>PUBLISHED</b>	<p>[SWS_Pwm_00098] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td> <td colspan="2">Pwm_SetPeriodAndDuty</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2"> <pre>void Pwm_SetPeriodAndDuty(     Pwm_ChannelType ChannelNumber,     Pwm_PeriodType Period,     uint16 DutyCycle )</pre> </td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td colspan="2">0x03</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant for different channel numbers</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ChannelNumber</td> <td>Numeric identifier of the PWM</td> </tr> <tr> <td></td> <td>Period</td> <td>Period of the PWM signal</td> </tr> <tr> <td></td> <td>DutyCycle</td> <td>Min=0x0000 Max=0x8000</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service sets the period and the duty cycle of a PWM channel</td> </tr> </table>	<b>Service name:</b>	Pwm_SetPeriodAndDuty		<b>Syntax:</b>	<pre>void Pwm_SetPeriodAndDuty(     Pwm_ChannelType ChannelNumber,     Pwm_PeriodType Period,     uint16 DutyCycle )</pre>		<b>Service ID[hex]:</b>	0x03		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant for different channel numbers		<b>Parameters (in):</b>	ChannelNumber	Numeric identifier of the PWM		Period	Period of the PWM signal		DutyCycle	Min=0x0000 Max=0x8000	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Service sets the period and the duty cycle of a PWM channel	
<b>Service name:</b>	Pwm_SetPeriodAndDuty																																						
<b>Syntax:</b>	<pre>void Pwm_SetPeriodAndDuty(     Pwm_ChannelType ChannelNumber,     Pwm_PeriodType Period,     uint16 DutyCycle )</pre>																																						
<b>Service ID[hex]:</b>	0x03																																						
<b>Sync/Async:</b>	Synchronous																																						
<b>Reentrancy:</b>	Reentrant for different channel numbers																																						
<b>Parameters (in):</b>	ChannelNumber	Numeric identifier of the PWM																																					
	Period	Period of the PWM signal																																					
	DutyCycle	Min=0x0000 Max=0x8000																																					
<b>Parameters (inout):</b>	None																																						
<b>Parameters (out):</b>	None																																						
<b>Return value:</b>	None																																						
<b>Description:</b>	Service sets the period and the duty cycle of a PWM channel																																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
12	Function Definitions	 <b>MCAL-6771</b> SWS_Pwm_000 99 : Pwm_SetOutputToIdle API Service Description <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Pwm_00099] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Pwm_SetOutputToIdle</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Pwm_SetOutputToIdle(                   Pwm_ChannelType ChannelNumber                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x04</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant for different channel numbers</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ChannelNumber      Numeric identifier of the PWM</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Service sets the PWM output to the configured Idle state.</td> </tr> </table>	<b>Service name:</b>	Pwm_SetOutputToIdle	<b>Syntax:</b>	void Pwm_SetOutputToIdle( Pwm_ChannelType ChannelNumber )	<b>Service ID[hex]:</b>	0x04	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant for different channel numbers	<b>Parameters (in):</b>	ChannelNumber      Numeric identifier of the PWM	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Service sets the PWM output to the configured Idle state.
<b>Service name:</b>	Pwm_SetOutputToIdle																						
<b>Syntax:</b>	void Pwm_SetOutputToIdle( Pwm_ChannelType ChannelNumber )																						
<b>Service ID[hex]:</b>	0x04																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant for different channel numbers																						
<b>Parameters (in):</b>	ChannelNumber      Numeric identifier of the PWM																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Service sets the PWM output to the configured Idle state.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																									
13	Function Definitions	 MCAL-6842 SWS_Pwm_001 01 : Pwm_DisableNotification API Service Description <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Pwm_00101] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Pwm_DisableNotification</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Pwm_DisableNotification(                   Pwm_ChannelType ChannelNumber                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x06</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant for different channel numbers</td> </tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Parameters (in):</b></td> <td>ChannelNumber</td> <td>Numeric identifier of the PWM</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Service to disable the PWM signal edge notification.</td> </tr> </table>	<b>Service name:</b>	Pwm_DisableNotification	<b>Syntax:</b>	void Pwm_DisableNotification( Pwm_ChannelType ChannelNumber )	<b>Service ID[hex]:</b>	0x06	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant for different channel numbers	<b>Parameters (in):</b>	ChannelNumber	Numeric identifier of the PWM	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Service to disable the PWM signal edge notification.	
<b>Service name:</b>	Pwm_DisableNotification																											
<b>Syntax:</b>	void Pwm_DisableNotification( Pwm_ChannelType ChannelNumber )																											
<b>Service ID[hex]:</b>	0x06																											
<b>Sync/Async:</b>	Synchronous																											
<b>Reentrancy:</b>	Reentrant for different channel numbers																											
<b>Parameters (in):</b>	ChannelNumber	Numeric identifier of the PWM																										
<b>Parameters (inout):</b>	None																											
<b>Parameters (out):</b>	None																											
<b>Return value:</b>	None																											
<b>Description:</b>	Service to disable the PWM signal edge notification.																											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
14	Function Definitions	 MCAL-6764 SWS_Pwm_001 02: Pwm_EnableN otification API Service Description <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	[SWS_Pwm_00102] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a> .

	Specification Category	Architecture Identifier	Description																							
			<table border="1"> <tr> <td data-bbox="702 477 945 509"><b>Service name:</b></td><td data-bbox="945 477 2084 509">Pwm_EnableNotification</td></tr> <tr> <td data-bbox="702 509 945 636"><b>Syntax:</b></td><td data-bbox="945 509 2084 636"> <pre>void Pwm_EnableNotification(     Pwm_ChannelType ChannelNumber,     Pwm_EdgeNotificationType Notification )</pre> </td></tr> <tr> <td data-bbox="702 636 945 668"><b>Service ID[hex]:</b></td><td data-bbox="945 636 2084 668">0x07</td></tr> <tr> <td data-bbox="702 668 945 700"><b>Sync/Async:</b></td><td data-bbox="945 668 2084 700">Synchronous</td></tr> <tr> <td data-bbox="702 700 945 732"><b>Reentrancy:</b></td><td data-bbox="945 700 2084 732">Reentrant for different channel numbers</td></tr> <tr> <td data-bbox="702 732 945 890" rowspan="2"><b>Parameters (in):</b></td><td data-bbox="945 732 1320 763">ChannelNumber</td><td data-bbox="1320 732 2084 763">Numeric identifier of the PWM</td></tr> <tr> <td data-bbox="945 763 1320 890">Notification</td><td data-bbox="1320 763 2084 890">Type of notification PWM_RISING_EDGE or PWM_FALLING_EDGE or PWM_BOTH_EDGES</td></tr> <tr> <td data-bbox="702 890 945 954"><b>Parameters (inout):</b></td><td data-bbox="945 890 2084 954">None</td></tr> <tr> <td data-bbox="702 954 945 986"><b>Parameters (out):</b></td><td data-bbox="945 954 2084 986">None</td></tr> <tr> <td data-bbox="702 986 945 1017"><b>Return value:</b></td><td data-bbox="945 986 2084 1017">None</td></tr> <tr> <td data-bbox="702 1017 945 1081"><b>Description:</b></td><td data-bbox="945 1017 2084 1081">Service to enable the PWM signal edge notification according to notification parameter.</td></tr> </table>	<b>Service name:</b>	Pwm_EnableNotification	<b>Syntax:</b>	<pre>void Pwm_EnableNotification(     Pwm_ChannelType ChannelNumber,     Pwm_EdgeNotificationType Notification )</pre>	<b>Service ID[hex]:</b>	0x07	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant for different channel numbers	<b>Parameters (in):</b>	ChannelNumber	Numeric identifier of the PWM	Notification	Type of notification PWM_RISING_EDGE or PWM_FALLING_EDGE or PWM_BOTH_EDGES	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Service to enable the PWM signal edge notification according to notification parameter.
<b>Service name:</b>	Pwm_EnableNotification																									
<b>Syntax:</b>	<pre>void Pwm_EnableNotification(     Pwm_ChannelType ChannelNumber,     Pwm_EdgeNotificationType Notification )</pre>																									
<b>Service ID[hex]:</b>	0x07																									
<b>Sync/Async:</b>	Synchronous																									
<b>Reentrancy:</b>	Reentrant for different channel numbers																									
<b>Parameters (in):</b>	ChannelNumber	Numeric identifier of the PWM																								
	Notification	Type of notification PWM_RISING_EDGE or PWM_FALLING_EDGE or PWM_BOTH_EDGES																								
<b>Parameters (inout):</b>	None																									
<b>Parameters (out):</b>	None																									
<b>Return value:</b>	None																									
<b>Description:</b>	Service to enable the PWM signal edge notification according to notification parameter.																									

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
15	Function Definitions	 MCAL-6843 SWS_Pwm_001 03 : Pwm_GetVersionInfo API Service Description <span style="background-color: #e0ffe0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Pwm_00103] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Pwm_GetVersionInfo</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Pwm_GetVersionInfo(                   Std_VersionInfoType* versioninfo                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x08</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>versioninfo   Pointer to where to store the version information of this module.</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Service returns the version information of this module.</td> </tr> </table>	<b>Service name:</b>	Pwm_GetVersionInfo	<b>Syntax:</b>	void Pwm_GetVersionInfo( Std_VersionInfoType* versioninfo )	<b>Service ID[hex]:</b>	0x08	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	versioninfo   Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	Service returns the version information of this module.
<b>Service name:</b>	Pwm_GetVersionInfo																						
<b>Syntax:</b>	void Pwm_GetVersionInfo( Std_VersionInfoType* versioninfo )																						
<b>Service ID[hex]:</b>	0x08																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	versioninfo   Pointer to where to store the version information of this module.																						
<b>Return value:</b>	None																						
<b>Description:</b>	Service returns the version information of this module.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
16	Configurable Interfaces	 MCAL-6783 SWS_Pwm_001 05 : Pwm_Notification_<#Channel> API Service Description <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Pwm_00105] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Pwm_Notification_&lt;#Channel&gt;</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Pwm_Notification_&lt;#Channel&gt;(     void )</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>PWM user implementation dependant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>The Pwm module shall call the function Pwm_Notification_&lt;#Channel&gt; accordingly to the last call of Pwm_EnableNotification for channel &lt;#Channel&gt;.</td> </tr> </table>	<b>Service name:</b>	Pwm_Notification_<#Channel>	<b>Syntax:</b>	void Pwm_Notification_<#Channel>( void )	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	PWM user implementation dependant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	The Pwm module shall call the function Pwm_Notification_<#Channel> accordingly to the last call of Pwm_EnableNotification for channel <#Channel>.
<b>Service name:</b>	Pwm_Notification_<#Channel>																				
<b>Syntax:</b>	void Pwm_Notification_<#Channel>( void )																				
<b>Sync/Async:</b>	Synchronous																				
<b>Reentrancy:</b>	PWM user implementation dependant																				
<b>Parameters (in):</b>	None																				
<b>Parameters (inout):</b>	None																				
<b>Parameters (out):</b>	None																				
<b>Return value:</b>	None																				
<b>Description:</b>	The Pwm module shall call the function Pwm_Notification_<#Channel> accordingly to the last call of Pwm_EnableNotification for channel <#Channel>.																				

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
17	Sequence Diagrams	 <a href="#">MCAL-7499</a> - Module PWM: Sequence Diagram <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	PWM module shall follow the sequence diagrams as described in section 8.7 of <a href="#">AUTOSAR BSW PWM Driver Specification</a> .
18	Configuration Specification	 <a href="#">MCAL-6850</a> - ECUC_Pwm_00 148 : Pwm Configuration Container <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	PWM module shall follow the containers and configuration parameters as described in <a href="#">AUTOSAR BSW PWM Driver Specification</a> section 10.  ECUC_Pwm_00148 - VARIANT-PRE-COMPIL , VARAINT-POST-BUILD are the variants it can use.

## 8.8 PWM (ePWM) Driver

The PWM module implements an interface in C programming language for handling the PWM functionality of the device. This PWM driver takes care of initializing and deinitializing the PWM unit and offers services to:

- Generate pulses with variable pulse width(Duty Cycle - can range from 0% to 100%)
- Set parameters of a PWM channels waveform(Duty Cycle and Period)
- Enable/Disable notifications

The PWM module initializes, configures and controls the internal hardware to realize PWM driver as detailed in AUTOSAR BSW PWM Driver Specification. The PWM functionality is realized through the ePWM.



The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW PWM Driver Specification](#).

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	Safety Process/ Diagnostic	 MCAL-7592 - MCAL Module : ePWM : Safety Diagnostic <span>PUBLISHED</span>	Following the Safety manual EPWM PIP and FMA review, the required safety diagnostic features should be added.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
2	File Structure	 <a href="#">MCAL-7504</a> - Module PWM: File Structure <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	PWM shall follow the file structure as defined in <a href="#">AUTOSAR BSW PWM Driver Specification</a> section 5.
3	General Behavior	 <a href="#">MCAL-7507</a> - Module PWM: General Behavior <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Pwm_00088] All functions from the PWM module except Pwm_Init, Pwm_Delnit and Pwm_GetVersionInfo shall be re-entrant for different PWM channel numbers.</p> <p>In order to keep a simple module implementation, no check of SWS_Pwm_00088 must be performed by the module. ()</p> <p>[SWS_Pwm_00089] The Pwm module's user shall ensure the integrity if several function calls are made during run time in different tasks or ISRs for the same PWM channel.</p> <p>[SWS_Pwm_00070] All time units used within the API services of the PWM module shall be of the unit ticks. () (SRS_BSW_00343)</p>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
4	Error Classification and Detection	 <a href="#">MCAL-7516</a> - Module PWM: Error Classification and Detection <b>PUBLISHED</b>	<p>PWM module shall have development Errors and Runtime Errors as defined in section 7.4 and 8.7 of <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <p>SWS_Pwm_00104 - Det_ReportError shall be used to report development errors.</p> <p>API Parameter checking shall also be employed.</p>
5	Duty Cycle Resolution and Scaling	 <a href="#">MCAL-7503</a> - Module PWM: Duty Cycle Resolution and Scaling <b>PUBLISHED</b>	<p>PWM module shall follow the Duty Cycle Resolution and Scaling guidelines as defined in section 7.7 of <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <p>As an implementation guide, the following source code example is given:</p> <pre>AbsoluteDutyCycle = ((uint32)AbsolutePeriodTime * RelativeDutyCycle) &gt;&gt; 15;</pre>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
6	Imported Types	 <a href="#">MCAL-7603 - SWS_Pwm_00094 : PWM Imported Types Dem</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	[SWS_Pwm_00094] PWM module shall use imported types Std_Types.
7	Type Definitions	 <a href="#">MCAL-7538 - Module PWM: Type Definitions</a> <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	PWM module shall define types as described in section 8.2 of <a href="#">AUTOSAR BSW PWM Driver Specification</a> .

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
8	Function Definitions	 <a href="#">MCAL-7534 - SWS_Pwm_00095 : Pwm_Init API</a> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Pwm_00095] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td> <td style="padding: 2px;">Pwm_Init</td> </tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td> <td style="padding: 2px;"> <pre>void Pwm_Init(     const Pwm_ConfigType* ConfigPtr )</pre> </td> </tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td> <td style="padding: 2px;">0x00</td> </tr> <tr> <td style="padding: 2px;"><b>Sync/Async:</b></td> <td style="padding: 2px;">Synchronous</td> </tr> <tr> <td style="padding: 2px;"><b>Reentrancy:</b></td> <td style="padding: 2px;">Non Reentrant</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (in):</b></td> <td style="padding: 2px;">           ConfigPtr      Pointer to configuration set         </td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (inout):</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (out):</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Return value:</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Description:</b></td> <td style="padding: 2px;">Service for PWM initialization.</td> </tr> </table>	<b>Service name:</b>	Pwm_Init	<b>Syntax:</b>	<pre>void Pwm_Init(     const Pwm_ConfigType* ConfigPtr )</pre>	<b>Service ID[hex]:</b>	0x00	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	ConfigPtr      Pointer to configuration set	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Service for PWM initialization.
<b>Service name:</b>	Pwm_Init																						
<b>Syntax:</b>	<pre>void Pwm_Init(     const Pwm_ConfigType* ConfigPtr )</pre>																						
<b>Service ID[hex]:</b>	0x00																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	ConfigPtr      Pointer to configuration set																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Service for PWM initialization.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
9	Function Definitions	<p> <a href="#">MCAL-7577</a> - SWS_Pwm_00096 : Pwm_DeInit API Service Description <span style="background-color: #e0f2e0; color: green;">PUBLISHED</span></p>	<p>[SWS_Pwm_00096] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Pwm_DeInit</td></tr> <tr> <td><b>Syntax:</b></td><td>void Pwm_DeInit(                   void                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x01</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Non Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>None</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td> <b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Service for PWM De-Initialization.</td></tr> </table>	<b>Service name:</b>	Pwm_DeInit	<b>Syntax:</b>	void Pwm_DeInit( void )	<b>Service ID[hex]:</b>	0x01	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	 <b>Return value:</b>	None	<b>Description:</b>	Service for PWM De-Initialization.
<b>Service name:</b>	Pwm_DeInit																						
<b>Syntax:</b>	void Pwm_DeInit( void )																						
<b>Service ID[hex]:</b>	0x01																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
 <b>Return value:</b>	None																						
<b>Description:</b>	Service for PWM De-Initialization.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																								
10	Function Definitions	<p> <a href="#">MCAL-7618</a> - SWS_Pwm_00097: Pwm_SetDutyCycle API Service</p> <p>Description</p> <p>PUBLISHED</p>	<p>[SWS_Pwm_00097] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Pwm_SetDutyCycle</td></tr> <tr> <td><b>Syntax:</b></td><td>void Pwm_SetDutyCycle(     Pwm_ChannelType ChannelNumber,     uint16 DutyCycle )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x02</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant for different channel numbers</td></tr> <tr> <td><b>Parameters (in):</b></td><td> <table border="1"> <tr> <td>ChannelNumber</td><td>Numeric identifier of the PWM</td></tr> <tr> <td>DutyCycle</td><td>Min=0x0000 Max=0x8000</td></tr> </table> </td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Service sets the duty cycle of the PWM channel.</td></tr> </table>	<b>Service name:</b>	Pwm_SetDutyCycle	<b>Syntax:</b>	void Pwm_SetDutyCycle( Pwm_ChannelType ChannelNumber, uint16 DutyCycle )	<b>Service ID[hex]:</b>	0x02	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant for different channel numbers	<b>Parameters (in):</b>	<table border="1"> <tr> <td>ChannelNumber</td><td>Numeric identifier of the PWM</td></tr> <tr> <td>DutyCycle</td><td>Min=0x0000 Max=0x8000</td></tr> </table>	ChannelNumber	Numeric identifier of the PWM	DutyCycle	Min=0x0000 Max=0x8000	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Service sets the duty cycle of the PWM channel.
<b>Service name:</b>	Pwm_SetDutyCycle																										
<b>Syntax:</b>	void Pwm_SetDutyCycle( Pwm_ChannelType ChannelNumber, uint16 DutyCycle )																										
<b>Service ID[hex]:</b>	0x02																										
<b>Sync/Async:</b>	Synchronous																										
<b>Reentrancy:</b>	Reentrant for different channel numbers																										
<b>Parameters (in):</b>	<table border="1"> <tr> <td>ChannelNumber</td><td>Numeric identifier of the PWM</td></tr> <tr> <td>DutyCycle</td><td>Min=0x0000 Max=0x8000</td></tr> </table>	ChannelNumber	Numeric identifier of the PWM	DutyCycle	Min=0x0000 Max=0x8000																						
ChannelNumber	Numeric identifier of the PWM																										
DutyCycle	Min=0x0000 Max=0x8000																										
<b>Parameters (inout):</b>	None																										
<b>Parameters (out):</b>	None																										
<b>Return value:</b>	None																										
<b>Description:</b>	Service sets the duty cycle of the PWM channel.																										

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																																				
11	Function Definitions	<p> <a href="#">MCAL-7508</a> - SWS_Pwm_00098: Pwm_SetPeriodAndDuty API Service Description</p> <p><b>PUBLISHED</b></p>	<p>[SWS_Pwm_00098] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td colspan="2">Pwm_SetPeriodAndDuty</td></tr> <tr> <td><b>Syntax:</b></td><td colspan="2">void Pwm_SetPeriodAndDuty(     Pwm_ChannelType ChannelNumber,     Pwm_PeriodType Period,     uint16 DutyCycle )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td colspan="2">0x03</td></tr> <tr> <td><b>Sync/Async:</b></td><td colspan="2">Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td colspan="2">Reentrant for different channel numbers</td></tr> <tr> <td><b>Parameters (in):</b></td><td>ChannelNumber</td><td>Numeric identifier of the PWM</td></tr> <tr> <td></td><td>Period</td><td>Period of the PWM signal</td></tr> <tr> <td></td><td>DutyCycle</td><td>Min=0x0000 Max=0x8000</td></tr> <tr> <td><b>Parameters (inout):</b></td><td colspan="2">None</td></tr> <tr> <td><b>Parameters (out):</b></td><td colspan="2">None</td></tr> <tr> <td><b>Return value:</b></td><td colspan="2">None</td></tr> <tr> <td><b>Description:</b></td><td colspan="2">Service sets the period and the duty cycle of a PWM channel</td></tr> </table>	<b>Service name:</b>	Pwm_SetPeriodAndDuty		<b>Syntax:</b>	void Pwm_SetPeriodAndDuty( Pwm_ChannelType ChannelNumber, Pwm_PeriodType Period, uint16 DutyCycle )		<b>Service ID[hex]:</b>	0x03		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant for different channel numbers		<b>Parameters (in):</b>	ChannelNumber	Numeric identifier of the PWM		Period	Period of the PWM signal		DutyCycle	Min=0x0000 Max=0x8000	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Service sets the period and the duty cycle of a PWM channel	
<b>Service name:</b>	Pwm_SetPeriodAndDuty																																						
<b>Syntax:</b>	void Pwm_SetPeriodAndDuty( Pwm_ChannelType ChannelNumber, Pwm_PeriodType Period, uint16 DutyCycle )																																						
<b>Service ID[hex]:</b>	0x03																																						
<b>Sync/Async:</b>	Synchronous																																						
<b>Reentrancy:</b>	Reentrant for different channel numbers																																						
<b>Parameters (in):</b>	ChannelNumber	Numeric identifier of the PWM																																					
	Period	Period of the PWM signal																																					
	DutyCycle	Min=0x0000 Max=0x8000																																					
<b>Parameters (inout):</b>	None																																						
<b>Parameters (out):</b>	None																																						
<b>Return value:</b>	None																																						
<b>Description:</b>	Service sets the period and the duty cycle of a PWM channel																																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
12	Function Definitions	<p> MCAL-7550 - SWS_Pwm_00099 : Pwm_SetOutputToIdle API Service Description</p> <p><b>PUBLISHED</b></p>	<p>[SWS_Pwm_00099] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Pwm_SetOutputToIdle</td></tr> <tr> <td><b>Syntax:</b></td><td>void Pwm_SetOutputToIdle(                   Pwm_ChannelType ChannelNumber                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x04</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant for different channel numbers</td></tr> <tr> <td><b>Parameters (in):</b></td><td>ChannelNumber      Numeric identifier of the PWM</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Service sets the PWM output to the configured Idle state.</td></tr> </table>	<b>Service name:</b>	Pwm_SetOutputToIdle	<b>Syntax:</b>	void Pwm_SetOutputToIdle( Pwm_ChannelType ChannelNumber )	<b>Service ID[hex]:</b>	0x04	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant for different channel numbers	<b>Parameters (in):</b>	ChannelNumber      Numeric identifier of the PWM	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Service sets the PWM output to the configured Idle state.
<b>Service name:</b>	Pwm_SetOutputToIdle																						
<b>Syntax:</b>	void Pwm_SetOutputToIdle( Pwm_ChannelType ChannelNumber )																						
<b>Service ID[hex]:</b>	0x04																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant for different channel numbers																						
<b>Parameters (in):</b>	ChannelNumber      Numeric identifier of the PWM																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Service sets the PWM output to the configured Idle state.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																														
13	Function Definitions	<p> <a href="#">MCAL-7558 - SWS_Pwm_00101</a> :</p> <p>Pwm_DisableNotification API Service Description</p> <p><b>PUBLISHED</b></p>	<p>[SWS_Pwm_00101] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td colspan="2">Pwm_DisableNotification</td></tr> <tr> <td><b>Syntax:</b></td><td colspan="2">void Pwm_DisableNotification(                   Pwm_ChannelType ChannelNumber                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td colspan="2">0x06</td></tr> <tr> <td><b>Sync/Async:</b></td><td colspan="2">Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td colspan="2">Reentrant for different channel numbers</td></tr> <tr> <td><b>Parameters (in):</b></td><td>ChannelNumber</td><td>Numeric identifier of the PWM</td></tr> <tr> <td><b>Parameters (inout):</b></td><td colspan="2">None</td></tr> <tr> <td><b>Parameters (out):</b></td><td colspan="2">None</td></tr> <tr> <td><b>Return value:</b></td><td colspan="2">None</td></tr> <tr> <td><b>Description:</b></td><td colspan="2">Service to disable the PWM signal edge notification.</td></tr> </table>	<b>Service name:</b>	Pwm_DisableNotification		<b>Syntax:</b>	void Pwm_DisableNotification( Pwm_ChannelType ChannelNumber )		<b>Service ID[hex]:</b>	0x06		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant for different channel numbers		<b>Parameters (in):</b>	ChannelNumber	Numeric identifier of the PWM	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	Service to disable the PWM signal edge notification.	
<b>Service name:</b>	Pwm_DisableNotification																																
<b>Syntax:</b>	void Pwm_DisableNotification( Pwm_ChannelType ChannelNumber )																																
<b>Service ID[hex]:</b>	0x06																																
<b>Sync/Async:</b>	Synchronous																																
<b>Reentrancy:</b>	Reentrant for different channel numbers																																
<b>Parameters (in):</b>	ChannelNumber	Numeric identifier of the PWM																															
<b>Parameters (inout):</b>	None																																
<b>Parameters (out):</b>	None																																
<b>Return value:</b>	None																																
<b>Description:</b>	Service to disable the PWM signal edge notification.																																

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
14	Function Definitions	 <a href="#">MCAL-7549</a> - SWS_Pwm_00102: Pwm_EnableNotification API Service Description <span style="background-color: #2e6b2e; color: white; padding: 2px;">PUBLISHED</span>	[SWS_Pwm_00102] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a> .

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>				
			<p><b>Service name:</b> Pwm_EnableNotification</p> <p><b>Syntax:</b></p> <pre>void Pwm_EnableNotification(     Pwm_ChannelType ChannelNumber,     Pwm_EdgeNotificationType Notification )</pre> <p><b>Service ID[hex]:</b> 0x07</p> <p><b>Sync/Async:</b> Synchronous</p> <p><b>Reentrancy:</b> Reentrant for different channel numbers</p> <table border="1"> <tr> <td>ChannelNumber</td> <td>Numeric identifier of the PWM</td> </tr> <tr> <td>Notification</td> <td>Type of notification PWM_RISING_EDGE or PWM_FALLING_EDGE or PWM_BOTH_EDGES</td> </tr> </table> <p><b>Parameters (in):</b></p> <p><b>Parameters (inout):</b> None</p> <p><b>Parameters (out):</b> None</p> <p><b>Return value:</b> None</p> <p><b>Description:</b> Service to enable the PWM signal edge notification according to notification parameter.</p>	ChannelNumber	Numeric identifier of the PWM	Notification	Type of notification PWM_RISING_EDGE or PWM_FALLING_EDGE or PWM_BOTH_EDGES
ChannelNumber	Numeric identifier of the PWM						
Notification	Type of notification PWM_RISING_EDGE or PWM_FALLING_EDGE or PWM_BOTH_EDGES						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
15	Function Definitions	<p> MCAL-7612 - SWS_Pwm_00103</p> <p>:</p> <p>Pwm_GetVersionInfo API Service Description</p> <p><b>PUBLISHED</b></p>	<p>[SWS_Pwm_00103] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Pwm_GetVersionInfo</td></tr> <tr> <td><b>Syntax:</b></td><td>void Pwm_GetVersionInfo(     Std_VersionInfoType* versioninfo )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x08</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>None</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>versioninfo   Pointer to where to store the version information of this module.</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Service returns the version information of this module.</td></tr> </table>	<b>Service name:</b>	Pwm_GetVersionInfo	<b>Syntax:</b>	void Pwm_GetVersionInfo( Std_VersionInfoType* versioninfo )	<b>Service ID[hex]:</b>	0x08	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	versioninfo   Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	Service returns the version information of this module.
<b>Service name:</b>	Pwm_GetVersionInfo																						
<b>Syntax:</b>	void Pwm_GetVersionInfo( Std_VersionInfoType* versioninfo )																						
<b>Service ID[hex]:</b>	0x08																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	versioninfo   Pointer to where to store the version information of this module.																						
<b>Return value:</b>	None																						
<b>Description:</b>	Service returns the version information of this module.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
16	Configurable Interfaces	<p> MCAL-7562 - SWS_Pwm_00105 :</p> <p>Pwm_Notification_&lt;#Channel&gt; API Service Description</p> <p><b>PUBLISHED</b></p>	<p>[SWS_Pwm_00105] shall be implemented and follow <a href="#">AUTOSAR BSW PWM Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Pwm_Notification_&lt;#Channel&gt;</td></tr> <tr> <td><b>Syntax:</b></td><td>void Pwm_Notification_&lt;#Channel&gt;(     void )</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>PWM user implementation dependant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>None</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>The Pwm module shall call the function Pwm_Notification_&lt;#Channel&gt; accordingly to the last call of Pwm_EnableNotification for channel &lt;#Channel&gt;.</td></tr> </table>	<b>Service name:</b>	Pwm_Notification_<#Channel>	<b>Syntax:</b>	void Pwm_Notification_<#Channel>( void )	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	PWM user implementation dependant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	The Pwm module shall call the function Pwm_Notification_<#Channel> accordingly to the last call of Pwm_EnableNotification for channel <#Channel>.
<b>Service name:</b>	Pwm_Notification_<#Channel>																				
<b>Syntax:</b>	void Pwm_Notification_<#Channel>( void )																				
<b>Sync/Async:</b>	Synchronous																				
<b>Reentrancy:</b>	PWM user implementation dependant																				
<b>Parameters (in):</b>	None																				
<b>Parameters (inout):</b>	None																				
<b>Parameters (out):</b>	None																				
<b>Return value:</b>	None																				
<b>Description:</b>	The Pwm module shall call the function Pwm_Notification_<#Channel> accordingly to the last call of Pwm_EnableNotification for channel <#Channel>.																				

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
17	Sequence Diagrams	 MCAL-7524 - Module PWM: Sequence Diagram <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	PWM module shall follow the sequence diagrams as described in section 8.7 of <a href="#">AUTOSAR BSW PWM Driver Specification</a> .
18	Configuration Specification	 MCAL-7580 - ECUC_Pwm_0014 8 : Pwm Configuration Container <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	PWM module shall follow the containers and configuration parameters as described in <a href="#">AUTOSAR BSW PWM Driver Specification</a> section 10.  ECUC_Pwm_00148 - VARIANT-PRE-COMPIL , VARAINT-POST-BUILD are the variants it can use.

## 8.9 ICU Driver

The Icu driver uses ECAP module to capture events. The Icu driver provides the following features:

- Signal Measurements - High time, Low time, Period time, Duty cycle
- Edge Detection - Provide notification for each edge detected
- Edge Counting - Measure edge counts
- Edge Timestamping - Measure the absolute time when edges occur

The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW ICU Driver Specification](#).



	Specification Category	Architecture Identifier	Description
1	Safety Process/ Diagnostic	 <b>MCAL-6956 -</b> MCAL Module : ICU: Safety Diagnostic <b>PUBLISHED</b>	Following the Safety manual Timer IP and FMA review, the required safety diagnostic features should be added.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
2	File Structure	 <b>MCAL-7645 -</b> MCAL Module: ICU: File Structure <b>PUBLISHED</b>	ICU shall follow the file structure as defined in <a href="#">AUTOSAR BSW ICU Driver Specification</a> section 5.
3	General Behavior	 <b>MCAL-7646 -</b> MCAL Module: ICU: General Behavior <b>PUBLISHED</b>	ICU shall follow the file structure as defined in <a href="#">AUTOSAR BSW ICU Driver Specification</a> section 7.1. To ensure data consistency re-entrant code shall be provided.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
4	Error Classification and Detection	 <a href="#">MCAL-7647 - MCAL Module: ICU: Error Classification and Detection</a> <span>PUBLISHED</span>	<p>ICU module shall support Development Error Types as defined in <a href="#">AUTOSAR BSW ICU Driver Specification</a> section 7.2 and 7.3.</p> <p>Det_ReportRuntimeError, DetReportError services shall be used.</p> <p>SWS_Icu_91001, SWS_Icu_00213, SWS_Icu_00382 and SWS_Icu_00022</p>
5	Type Definitions	 <a href="#">MCAL-7648 - MCAL Module: ICU: Type Definitions</a> <span>PUBLISHED</span>	ICU module shall define types as described in section 8.2 of <a href="#">AUTOSAR BSW ICU Driver Specification</a> .

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
6	Function Definitions	 <b>MCAL-7139 - SWS_Icu_0019</b> 1 : Service name: Icu_Init <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00191] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Icu_Init</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Icu_Init(           const Icu_ConfigType* ConfigPtr           )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x00</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ConfigPtr      Pointer to a selected configuration structure</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>This function initializes the driver.</td> </tr> </table>	<b>Service name:</b>	Icu_Init	<b>Syntax:</b>	void Icu_Init( const Icu_ConfigType* ConfigPtr )	<b>Service ID[hex]:</b>	0x00	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	ConfigPtr      Pointer to a selected configuration structure	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function initializes the driver.
<b>Service name:</b>	Icu_Init																						
<b>Syntax:</b>	void Icu_Init( const Icu_ConfigType* ConfigPtr )																						
<b>Service ID[hex]:</b>	0x00																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	ConfigPtr      Pointer to a selected configuration structure																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function initializes the driver.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
7	Function Definitions	 <b>MCAL-6984 - SWS_Icu_0019</b> 3 : Service name: <b>Icu_DelInit</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00193] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td><td style="padding: 2px;">Icu_DelInit</td></tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td><td style="padding: 2px;">void Icu_DeInit(                   void                   )                   </td></tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td><td style="padding: 2px;">0x01</td></tr> <tr> <td style="padding: 2px;"><b>Sync/Async:</b></td><td style="padding: 2px;">Synchronous</td></tr> <tr> <td style="padding: 2px;"><b>Reentrancy:</b></td><td style="padding: 2px;">Non Reentrant</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (in):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (inout):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Parameters (out):</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Return value:</b></td><td style="padding: 2px;">None</td></tr> <tr> <td style="padding: 2px;"><b>Description:</b></td><td style="padding: 2px;">This function de-initializes the ICU module.</td></tr> </table>	<b>Service name:</b>	Icu_DelInit	<b>Syntax:</b>	void Icu_DeInit( void ) 	<b>Service ID[hex]:</b>	0x01	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function de-initializes the ICU module.
<b>Service name:</b>	Icu_DelInit																						
<b>Syntax:</b>	void Icu_DeInit( void ) 																						
<b>Service ID[hex]:</b>	0x01																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function de-initializes the ICU module.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																							
8	Function Definitions	 <b>MCAL-6932 - SWS_Icu_00197 : Service name:</b> <b>Icu_SetActivationCondition</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00197] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td colspan="2">Icu_SetActivationCondition</td> </tr> <tr> <td><b>Syntax:</b></td> <td colspan="2">void Icu_SetActivationCondition(     Icu_ChannelType Channel,     Icu_ActivationType Activation )</td> </tr> <tr> <td><b>Service ID/hex:</b></td> <td colspan="2">0x05</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td colspan="2">Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td colspan="2">Reentrant (limited according to ICU050)</td> </tr> <tr> <td rowspan="2"><b>Parameters (in):</b></td> <td>Channel</td> <td>Numeric identifier of the ICU channel</td> </tr> <tr> <td>Activation</td> <td>Type of activation (if supported by hardware) - ICU_RISING_EDGE - ICU_FALLING_EDGE - ICU_BOTH_EDGES</td> </tr> <tr> <td><b>Parameters</b></td> <td colspan="2">None</td> </tr> </table>	<b>Service name:</b>	Icu_SetActivationCondition		<b>Syntax:</b>	void Icu_SetActivationCondition( Icu_ChannelType Channel, Icu_ActivationType Activation )		<b>Service ID/hex:</b>	0x05		<b>Sync/Async:</b>	Synchronous		<b>Reentrancy:</b>	Reentrant (limited according to ICU050)		<b>Parameters (in):</b>	Channel	Numeric identifier of the ICU channel	Activation	Type of activation (if supported by hardware) - ICU_RISING_EDGE - ICU_FALLING_EDGE - ICU_BOTH_EDGES	<b>Parameters</b>	None	
<b>Service name:</b>	Icu_SetActivationCondition																									
<b>Syntax:</b>	void Icu_SetActivationCondition( Icu_ChannelType Channel, Icu_ActivationType Activation )																									
<b>Service ID/hex:</b>	0x05																									
<b>Sync/Async:</b>	Synchronous																									
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																									
<b>Parameters (in):</b>	Channel	Numeric identifier of the ICU channel																								
	Activation	Type of activation (if supported by hardware) - ICU_RISING_EDGE - ICU_FALLING_EDGE - ICU_BOTH_EDGES																								
<b>Parameters</b>	None																									

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>								
			<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>(inout):</b></td> <td></td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>This function sets the activation-edge for the given channel.</td> </tr> </table>	<b>(inout):</b>		<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function sets the activation-edge for the given channel.
<b>(inout):</b>											
<b>Parameters (out):</b>	None										
<b>Return value:</b>	None										
<b>Description:</b>	This function sets the activation-edge for the given channel.										

	Specification Category	Architecture Identifier	Description																									
9	Function Definitions	 <b>MCAL-7049 - SWS_Icu_0019</b> 8 : Service name: Icu_DisableNotification <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00198] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" data-bbox="691 568 1913 1044"> <tr> <td data-bbox="691 568 961 608"><b>Service name:</b></td> <td data-bbox="961 568 1913 608">Icu_DisableNotification</td> </tr> <tr> <td data-bbox="691 608 961 703"><b>Syntax:</b></td> <td data-bbox="961 608 1913 703">void Icu_DisableNotification(     Icu_ChannelType Channel )</td> </tr> <tr> <td data-bbox="691 703 961 743"><b>Service ID[hex]:</b></td> <td data-bbox="961 703 1913 743">0x06</td> </tr> <tr> <td data-bbox="691 743 961 782"><b>Sync/Async:</b></td> <td data-bbox="961 743 1913 782">Synchronous</td> </tr> <tr> <td data-bbox="691 782 961 822"><b>Reentrancy:</b></td> <td data-bbox="961 782 1913 822">Reentrant (limited according to ICU050)</td> </tr> <tr> <td data-bbox="691 846 961 886"><b>Parameters (in):</b></td> <td data-bbox="961 846 1140 886">Channel</td> <td data-bbox="1140 846 1913 886">Numeric identifier of the ICU channel</td> </tr> <tr> <td data-bbox="691 886 961 941"><b>Parameters (inout):</b></td> <td data-bbox="961 886 1140 941">None</td> <td data-bbox="1140 886 1913 941"></td> </tr> <tr> <td data-bbox="691 941 961 981"><b>Parameters (out):</b></td> <td data-bbox="961 941 1140 981">None</td> <td data-bbox="1140 941 1913 981"></td> </tr> <tr> <td data-bbox="691 981 961 1021"><b>Return value:</b></td> <td data-bbox="961 981 1140 1021">None</td> <td data-bbox="1140 981 1913 1021"></td> </tr> <tr> <td data-bbox="691 1021 961 1044"><b>Description:</b></td> <td colspan="2" data-bbox="961 1021 1913 1044">This function disables the notification of a channel.</td></tr> </table>	<b>Service name:</b>	Icu_DisableNotification	<b>Syntax:</b>	void Icu_DisableNotification( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x06	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel	Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	This function disables the notification of a channel.	
<b>Service name:</b>	Icu_DisableNotification																											
<b>Syntax:</b>	void Icu_DisableNotification( Icu_ChannelType Channel )																											
<b>Service ID[hex]:</b>	0x06																											
<b>Sync/Async:</b>	Synchronous																											
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																											
<b>Parameters (in):</b>	Channel	Numeric identifier of the ICU channel																										
<b>Parameters (inout):</b>	None																											
<b>Parameters (out):</b>	None																											
<b>Return value:</b>	None																											
<b>Description:</b>	This function disables the notification of a channel.																											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
10	Function Definitions	 <b>MCAL-7093 - SWS_Icu_0019</b> : Service name: <b>Icu_EnableNotification</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00199] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><b>Service name:</b></td><td>Icu_EnableNotification</td></tr> <tr> <td><b>Syntax:</b></td><td>void Icu_EnableNotification(                   Icu_ChannelType Channel                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x07</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (limited according to ICU050)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the ICU channel</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This function enables the notification on the given channel.</td></tr> </table>	<b>Service name:</b>	Icu_EnableNotification	<b>Syntax:</b>	void Icu_EnableNotification( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x07	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function enables the notification on the given channel.
<b>Service name:</b>	Icu_EnableNotification																						
<b>Syntax:</b>	void Icu_EnableNotification( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x07																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function enables the notification on the given channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
11	Function Definitions	 <b>MCAL-7080 - SWS_Icu_00200 : Service name: Icu_GetInputState</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00200] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Icu_GetInputState</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Icu_InputStateType Icu_GetInputState(                           Icu_ChannelType Channel                          )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x08</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant (limited according to ICU050)</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel      Numeric identifier of the ICU channel</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Icu_InputStateType ICU_ACTIVE: An activation edge has been detected ICU_IDLE: No activation edge has been detected since the last call of Icu_GetInputState() or Icu_Init().</td> </tr> <tr> <td><b>Description:</b></td> <td>This function returns the status of the ICU input.</td> </tr> </table>	<b>Service name:</b>	Icu_GetInputState	<b>Syntax:</b>	Icu_InputStateType Icu_GetInputState( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x08	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Icu_InputStateType ICU_ACTIVE: An activation edge has been detected ICU_IDLE: No activation edge has been detected since the last call of Icu_GetInputState() or Icu_Init().	<b>Description:</b>	This function returns the status of the ICU input.
<b>Service name:</b>	Icu_GetInputState																						
<b>Syntax:</b>	Icu_InputStateType Icu_GetInputState( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x08																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	Icu_InputStateType ICU_ACTIVE: An activation edge has been detected ICU_IDLE: No activation edge has been detected since the last call of Icu_GetInputState() or Icu_Init().																						
<b>Description:</b>	This function returns the status of the ICU input.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
12	Function Definitions	 <b>MCAL-7105 - SWS_Icu_0020</b> 1 : Service name: Icu_StartTime stamp <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	[SWS_Icu_00201] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a> .

Specification Category	Architecture Identifier	Description							
		<p><b>Service name:</b> Icu_StartTimestamp</p> <p><b>Syntax:</b></p> <pre>void Icu_StartTimestamp(     Icu_ChannelType Channel,     Icu_ValueType* BufferPtr,     uint16 BufferSize,     uint16 NotifyInterval )</pre> <p><b>Service ID[hex]:</b> 0x09</p> <p><b>Sync/Async:</b> Asynchronous</p> <p><b>Reentrancy:</b> Reentrant (limited according to ICU050)</p> <table border="1" data-bbox="705 806 1918 933"> <tr> <td>Channel</td> <td>Numeric identifier of the ICU channel</td> </tr> <tr> <td>BufferSize</td> <td>Size of the external buffer (number of entries)</td> </tr> <tr> <td>NotifyInterval</td> <td>Notification interval (number of events). This parameter can not be checked in a reasonable way.</td> </tr> </table> <p><b>Parameters (in):</b></p> <p><b>Parameters (inout):</b> None</p> <p><b>Parameters (out):</b> BufferPtr    Pointer to the buffer-array where the timestamp values shall be placed.</p> <p><b>Return value:</b> None</p> <p><b>Description:</b> This function starts the capturing of timer values on the edges.</p>	Channel	Numeric identifier of the ICU channel	BufferSize	Size of the external buffer (number of entries)	NotifyInterval	Notification interval (number of events). This parameter can not be checked in a reasonable way.	
Channel	Numeric identifier of the ICU channel								
BufferSize	Size of the external buffer (number of entries)								
NotifyInterval	Notification interval (number of events). This parameter can not be checked in a reasonable way.								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
13	Function Definitions	 <b>MCAL-6978 - SWS_Icu_00202 : Service name: Icu_StopTimestamp</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00202] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><b>Service name:</b></td><td>Icu_StopTimestamp</td></tr> <tr> <td><b>Syntax:</b></td><td>void Icu_StopTimestamp(                   Icu_ChannelType Channel                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x0a</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (limited according to ICU050)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the ICU channel</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This function stops the timestamp measurement of the given channel.</td></tr> </table>	<b>Service name:</b>	Icu_StopTimestamp	<b>Syntax:</b>	void Icu_StopTimestamp( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x0a	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function stops the timestamp measurement of the given channel.
<b>Service name:</b>	Icu_StopTimestamp																						
<b>Syntax:</b>	void Icu_StopTimestamp( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x0a																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function stops the timestamp measurement of the given channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
14	Function Definitions	 <b>MCAL-6925 - SWS_Icu_00203 : Service name:</b> <b>Icu_GetTimestampIndex</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00203] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><b>Service name:</b></td><td>Icu_GetTimestampIndex</td></tr> <tr> <td><b>Syntax:</b></td><td>Icu_IndexType Icu_GetTimestampIndex(                   Icu_ChannelType Channel                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x0b</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (limited according to ICU050)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the ICU channel</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>Icu_IndexType      Abstract return type to cover different microcontrollers.</td></tr> <tr> <td><b>Description:</b></td><td>This function reads the timestamp index of the given channel.</td></tr> </table>	<b>Service name:</b>	Icu_GetTimestampIndex	<b>Syntax:</b>	Icu_IndexType Icu_GetTimestampIndex( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x0b	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Icu_IndexType      Abstract return type to cover different microcontrollers.	<b>Description:</b>	This function reads the timestamp index of the given channel.
<b>Service name:</b>	Icu_GetTimestampIndex																						
<b>Syntax:</b>	Icu_IndexType Icu_GetTimestampIndex( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x0b																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	Icu_IndexType      Abstract return type to cover different microcontrollers.																						
<b>Description:</b>	This function reads the timestamp index of the given channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
15	Function Definitions	 <b>MCAL-7125 - SWS_Icu_00204 : Service name:</b> <b>Icu_ResetEdgeCount</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00204] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td> <td style="padding: 2px;">Icu_ResetEdgeCount</td> </tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td> <td style="padding: 2px;">void Icu_ResetEdgeCount(     Icu_ChannelType Channel )</td> </tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td> <td style="padding: 2px;">0x0c</td> </tr> <tr> <td style="padding: 2px;"><b>Sync/Async:</b></td> <td style="padding: 2px;">Synchronous</td> </tr> <tr> <td style="padding: 2px;"><b>Reentrancy:</b></td> <td style="padding: 2px;">Reentrant (limited according to ICU050)</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (in):</b></td> <td style="padding: 2px;">Channel      Numeric identifier of the ICU channel</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (inout):</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Parameters (out):</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Return value:</b></td> <td style="padding: 2px;">None</td> </tr> <tr> <td style="padding: 2px;"><b>Description:</b></td> <td style="padding: 2px;">This function resets the value of the counted edges to zero.</td> </tr> </table>	<b>Service name:</b>	Icu_ResetEdgeCount	<b>Syntax:</b>	void Icu_ResetEdgeCount( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x0c	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function resets the value of the counted edges to zero.
<b>Service name:</b>	Icu_ResetEdgeCount																						
<b>Syntax:</b>	void Icu_ResetEdgeCount( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x0c																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function resets the value of the counted edges to zero.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
16	Function Definitions	 <b>MCAL-7060 - SWS_Icu_00205 : Service name:</b> <b>Icu_EnableEdgeCount</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00205] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Icu_EnableEdgeCount</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Icu_EnableEdgeCount(                   Icu_ChannelType Channel                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x0d</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant (limited according to ICU050)</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel      Numeric identifier of the ICU channel</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>This function enables the counting of edges of the given channel.</td> </tr> </table>	<b>Service name:</b>	Icu_EnableEdgeCount	<b>Syntax:</b>	void Icu_EnableEdgeCount( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x0d	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function enables the counting of edges of the given channel.
<b>Service name:</b>	Icu_EnableEdgeCount																						
<b>Syntax:</b>	void Icu_EnableEdgeCount( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x0d																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function enables the counting of edges of the given channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
17	Function Definitions	 <b>MCAL-6913 - SWS_Icu_00364 : Service name:</b> <b>Icu_EnableEdgeDetection</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00364] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><b>Service name:</b></td><td>Icu_EnableEdgeDetection</td></tr> <tr> <td><b>Syntax:</b></td><td>void Icu_EnableEdgeDetection(     Icu_ChannelType Channel )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x16</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (limited according to ICU050)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the ICU channel</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This function enables / re-enables the detection of edges of the given channel.</td></tr> </table>	<b>Service name:</b>	Icu_EnableEdgeDetection	<b>Syntax:</b>	void Icu_EnableEdgeDetection( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x16	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function enables / re-enables the detection of edges of the given channel.
<b>Service name:</b>	Icu_EnableEdgeDetection																						
<b>Syntax:</b>	void Icu_EnableEdgeDetection( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x16																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function enables / re-enables the detection of edges of the given channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
18	Function Definitions	 <b>MCAL-7121 - SWS_Icu_00377 : Service name:</b> <b>Icu_DisableEdgeDetection</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00377] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><b>Service name:</b></td><td>Icu_DisableEdgeDetection</td></tr> <tr> <td><b>Syntax:</b></td><td>void Icu_DisableEdgeDetection(     Icu_ChannelType Channel )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x17</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (limited according to ICU050)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the ICU channel</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This function disables the detection of edges of the given channel.</td></tr> </table>	<b>Service name:</b>	Icu_DisableEdgeDetection	<b>Syntax:</b>	void Icu_DisableEdgeDetection( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x17	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function disables the detection of edges of the given channel.
<b>Service name:</b>	Icu_DisableEdgeDetection																						
<b>Syntax:</b>	void Icu_DisableEdgeDetection( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x17																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function disables the detection of edges of the given channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
19	Function Definitions	 <b>MCAL-7016 - SWS_Icu_00206 : Service name:</b> <b>Icu_DisableEdgeCount</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00206] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Icu_DisableEdgeCount</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Icu_DisableEdgeCount(     Icu_ChannelType Channel )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x0e</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant (limited according to ICU050)</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel      Numeric identifier of the ICU channel</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>This function disables the counting of edges of the given channel.</td> </tr> </table>	<b>Service name:</b>	Icu_DisableEdgeCount	<b>Syntax:</b>	void Icu_DisableEdgeCount( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x0e	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function disables the counting of edges of the given channel.
<b>Service name:</b>	Icu_DisableEdgeCount																						
<b>Syntax:</b>	void Icu_DisableEdgeCount( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x0e																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function disables the counting of edges of the given channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
20	Function Definitions	 <b>MCAL-7005 - SWS_Icu_00207 : Service name:</b> <b>Icu_GetEdgeNumbers</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00207] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Icu_GetEdgeNumbers</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Icu_EdgeNumberType Icu_GetEdgeNumbers(     Icu_ChannelType Channel )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x0f</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant (limited according to ICU050)</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel                  Numeric identifier of the ICU channel</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Icu_EdgeNumberType    Abstract return type to cover different microcontrollers.</td> </tr> <tr> <td><b>Description:</b></td> <td>This function reads the number of counted edges.</td> </tr> </table>	<b>Service name:</b>	Icu_GetEdgeNumbers	<b>Syntax:</b>	Icu_EdgeNumberType Icu_GetEdgeNumbers( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x0f	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel                  Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Icu_EdgeNumberType    Abstract return type to cover different microcontrollers.	<b>Description:</b>	This function reads the number of counted edges.
<b>Service name:</b>	Icu_GetEdgeNumbers																						
<b>Syntax:</b>	Icu_EdgeNumberType Icu_GetEdgeNumbers( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x0f																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel                  Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	Icu_EdgeNumberType    Abstract return type to cover different microcontrollers.																						
<b>Description:</b>	This function reads the number of counted edges.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
21	Function Definitions	 <b>MCAL-6958 - SWS_Icu_00208 : Service name:</b> <b>Icu_StartSignalMeasurement</b> <span style="background-color: #d9ead3; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00208] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td><td>Icu_StartSignalMeasurement</td></tr> <tr> <td><b>Syntax:</b></td><td>void Icu_StartSignalMeasurement(     Icu_ChannelType Channel )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x13</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Asynchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (limited according to ICU050)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the ICU channel</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This function starts the measurement of signals.</td></tr> </table>	<b>Service name:</b>	Icu_StartSignalMeasurement	<b>Syntax:</b>	void Icu_StartSignalMeasurement( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x13	<b>Sync/Async:</b>	Asynchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function starts the measurement of signals.
<b>Service name:</b>	Icu_StartSignalMeasurement																						
<b>Syntax:</b>	void Icu_StartSignalMeasurement( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x13																						
<b>Sync/Async:</b>	Asynchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function starts the measurement of signals.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
22	Function Definitions	 <b>MCAL-7114 - SWS_Icu_00209 : Service name:</b> <b>Icu_StopSignalMeasurement</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00209] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><b>Service name:</b></td><td>Icu_StopSignalMeasurement</td></tr> <tr> <td><b>Syntax:</b></td><td>void Icu_StopSignalMeasurement(                   Icu_ChannelType Channel                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x14</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (limited according to ICU050)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel      Numeric identifier of the ICU channel</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This function stops the measurement of signals of the given channel.</td></tr> </table>	<b>Service name:</b>	Icu_StopSignalMeasurement	<b>Syntax:</b>	void Icu_StopSignalMeasurement( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x14	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This function stops the measurement of signals of the given channel.
<b>Service name:</b>	Icu_StopSignalMeasurement																						
<b>Syntax:</b>	void Icu_StopSignalMeasurement( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x14																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel      Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function stops the measurement of signals of the given channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
23	Function Definitions	 <b>MCAL-7136 - SWS_Icu_00210 : Service name:</b> <b>Icu_GetTimeElapsed</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00210] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Icu_GetTimeElapsed</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Icu_ValueType Icu_GetTimeElapsed(     Icu_ChannelType Channel )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x10</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant (limited according to ICU050)</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>Channel                  Numeric identifier of the ICU channel</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Icu_ValueType            see Description</td> </tr> <tr> <td><b>Description:</b></td> <td>This function reads the elapsed Signal Low Time for the given channel.</td> </tr> </table>	<b>Service name:</b>	Icu_GetTimeElapsed	<b>Syntax:</b>	Icu_ValueType Icu_GetTimeElapsed( Icu_ChannelType Channel )	<b>Service ID[hex]:</b>	0x10	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel                  Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Icu_ValueType            see Description	<b>Description:</b>	This function reads the elapsed Signal Low Time for the given channel.
<b>Service name:</b>	Icu_GetTimeElapsed																						
<b>Syntax:</b>	Icu_ValueType Icu_GetTimeElapsed( Icu_ChannelType Channel )																						
<b>Service ID[hex]:</b>	0x10																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel                  Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	Icu_ValueType            see Description																						
<b>Description:</b>	This function reads the elapsed Signal Low Time for the given channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
24	Function Definitions	 <b>MCAL-6985 - SWS_Icu_0021</b> 1 : Service name: <b>Icu_GetDutyCycleValues</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00211] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><b>Service name:</b></td><td>Icu_GetDutyCycleValues</td></tr> <tr> <td><b>Syntax:</b></td><td>void Icu_GetDutyCycleValues(     Icu_ChannelType Channel,     Icu_DutyCycleType* DutyCycleValues )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x11</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant (limited according to ICU050)</td></tr> <tr> <td><b>Parameters (in):</b></td><td>Channel                  Numeric identifier of the ICU channel</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>DutyCycleValues          Pointer to a buffer where the results (high time and period time) shall be placed.</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This function reads the coherent active time and period time for the given ICU Channel.</td></tr> </table>	<b>Service name:</b>	Icu_GetDutyCycleValues	<b>Syntax:</b>	void Icu_GetDutyCycleValues( Icu_ChannelType Channel, Icu_DutyCycleType* DutyCycleValues )	<b>Service ID[hex]:</b>	0x11	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant (limited according to ICU050)	<b>Parameters (in):</b>	Channel                  Numeric identifier of the ICU channel	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	DutyCycleValues          Pointer to a buffer where the results (high time and period time) shall be placed.	<b>Return value:</b>	None	<b>Description:</b>	This function reads the coherent active time and period time for the given ICU Channel.
<b>Service name:</b>	Icu_GetDutyCycleValues																						
<b>Syntax:</b>	void Icu_GetDutyCycleValues( Icu_ChannelType Channel, Icu_DutyCycleType* DutyCycleValues )																						
<b>Service ID[hex]:</b>	0x11																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant (limited according to ICU050)																						
<b>Parameters (in):</b>	Channel                  Numeric identifier of the ICU channel																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	DutyCycleValues          Pointer to a buffer where the results (high time and period time) shall be placed.																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function reads the coherent active time and period time for the given ICU Channel.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
25	Function Definitions	 <b>MCAL-7078 - SWS_Icu_00212 : Service name:</b> <b>Icu_GetVersionInfo</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Icu_00212] shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><b>Service name:</b></td><td>Icu_GetVersionInfo</td></tr> <tr> <td><b>Syntax:</b></td><td>void Icu_GetVersionInfo(                   Std_VersionInfoType* versioninfo                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x12</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>None</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>versioninfo   Pointer to where to store the version information of this module.</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This function returns the version information of this module.</td></tr> </table>	<b>Service name:</b>	Icu_GetVersionInfo	<b>Syntax:</b>	void Icu_GetVersionInfo( Std_VersionInfoType* versioninfo )	<b>Service ID[hex]:</b>	0x12	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	versioninfo   Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	This function returns the version information of this module.
<b>Service name:</b>	Icu_GetVersionInfo																						
<b>Syntax:</b>	void Icu_GetVersionInfo( Std_VersionInfoType* versioninfo )																						
<b>Service ID[hex]:</b>	0x12																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	versioninfo   Pointer to where to store the version information of this module.																						
<b>Return value:</b>	None																						
<b>Description:</b>	This function returns the version information of this module.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
26	Configurable Interfaces	 <b>MCAL-7649 -</b> MCAL Module: ICU: Configurable Interfaces <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	<p>SWS_Icu_00214 and SWS_Icu_00215 shall be implemented and follow <a href="#">AUTOSAR BSW ICU Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Icu_SignalNotification_&lt;Channel&gt;</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Icu_SignalNotification_&lt;Channel&gt;(     void )</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrancy of interface not relevant for this module. (in general it is in this case not reentrant).</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>According to the last call of Icu_EnableNotification, this notification function to be called if the requested signal edge (rising / falling / both edges) occurs (once per edge).</td> </tr> </table>	<b>Service name:</b>	Icu_SignalNotification_<Channel>	<b>Syntax:</b>	void Icu_SignalNotification_<Channel>( void )	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrancy of interface not relevant for this module. (in general it is in this case not reentrant).	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	According to the last call of Icu_EnableNotification, this notification function to be called if the requested signal edge (rising / falling / both edges) occurs (once per edge).
<b>Service name:</b>	Icu_SignalNotification_<Channel>																				
<b>Syntax:</b>	void Icu_SignalNotification_<Channel>( void )																				
<b>Sync/Async:</b>	Synchronous																				
<b>Reentrancy:</b>	Reentrancy of interface not relevant for this module. (in general it is in this case not reentrant).																				
<b>Parameters (in):</b>	None																				
<b>Parameters (inout):</b>	None																				
<b>Parameters (out):</b>	None																				
<b>Return value:</b>	None																				
<b>Description:</b>	According to the last call of Icu_EnableNotification, this notification function to be called if the requested signal edge (rising / falling / both edges) occurs (once per edge).																				

Specification Category	Architecture Identifier	Description
		<p><b>Service name:</b> Icu_TimestampNotification_&lt;Channel&gt;</p> <p><b>Syntax:</b> void Icu_TimestampNotification_&lt;Channel&gt;(                   void                   )</p> <p><b>Sync/Async:</b> Synchronous</p> <p><b>Reentrancy:</b> Reentrancy of interface not relevant for this module. (in general it is in this case not reentrant).</p> <p><b>Parameters (in):</b> None</p> <p><b>Parameters (inout):</b> None</p> <p><b>Parameters (out):</b> None</p> <p><b>Return value:</b> None</p> <p><b>Description:</b> This notification to be called if the number of requested timestamps (Notification interval &gt; 0) are acquired and if the notification has been enabled by the call of Icu_EnableNotification().</p>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
27	Sequence Diagrams	 <b>MCAL-7650 -</b> MCAL Module: ICU: Sequence Diagram <b>PUBLISHED</b>	ICU module shall follow the sequence diagrams as described in section 9 of <a href="#">AUTOSAR BSW ICU Driver Specification</a> .
28	Configuration Specification	 <b>MCAL-7031 -</b> ECUC_Icu_003 57 : Icu Module Container <b>PUBLISHED</b>	ICU module shall follow the containers and configuration parameters as described in section 10 of <a href="#">AUTOSAR BSW ICU Driver Specification</a> .  ECUC_Icu_00357- VARIANT-PRE-COMPIL , VARAINT-POST-BUILD are the variants it can use.

## 8.10 CDD IPC Driver

CDD IPC primarily used for communication with other cores on the SoC. This implementation relies on mailbox and shared memory to establish communication channel.

Shared memory holds the messages that requires to be transported and mailbox is used to notify the remote core on availability of a message.

CDD IPC allows core hosting MCAL/AUTOSAR to communicate with other cores (processing entities, with-in SoC) hosting CDD IPC driver as well as HLOS Linux IPC driver. This driver could be used to transmit and receive variable length messages between cores, via logical communication channel ID's. Can be mapped to Sender-Receiver AUTOSAR interface, for data oriented communication between core that host AUTOSAR / NON AUTOSAR processing entities



The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW CDD Specification](#).

	<b>Specification Category</b>		<b>Description</b>
1	Safety Process/ Diagnostic	 <a href="#">MCAL-7172</a> - MCAL Module : CDD IPC : Safety Diagnostic <span>PUBLISHED</span>	Following the Safety manual IPC and FMA review, the required safety diagnostic features should be added.

#### Architecture Identifier

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
2	Dependency on other modules	 <a href="#">MCAL-7181</a> - MCAL Module: CDD IPC: PDK Dependency <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	CDD IPC shall not have external dependency on PDK CSL or PDK IPC LLD. IPC driver shall be integrated inside CDD IPC MCAL.
3	General Behavior	 <a href="#">MCAL-7653</a> - MCAL Module: CDD IPC: State Machine <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	CDD IPC shall maintain two distinct states <b>Initialized &amp; Un Initialized</b>
4	Error Classification and Detection	 <a href="#">MCAL-7654</a> - MCAL Module: CDD IPC: Error Classification and Detection <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	CDD IPC shall report development errors and runtime/production errors and use Det-ReportError service or Dem_ReportErrorStatus service.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>												
5	Function Definitions	 <a href="#">MCAL-7655</a> - MCAL Module: CDD IPC: Cdd_IpcNewMessageNotify <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px 5px;">PUBLISHED</span>	<p>Cdd_IpcNewMessageNotify</p> <p>Is a function implemented by the application, with prototype <code>void Cdd_IpcNewMessageNotify ( uint32 commId )</code>, where commId is the value specified by integrator while creating the communication channelid</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th><i>Description</i></th> <th><i>Comment</i></th> </tr> </thead> <tbody> <tr> <td><b>Function Name</b></td> <td>Cdd_IpcNewMessageNotify</td> <td>Is a symbolic name, integrators can specify desired name</td> </tr> <tr> <td><b>Syntax</b></td> <td> <code>void</code>  <code>Cdd_IpcNewMessageNotify( uint32</code>  <code>commId )</code> </td> <td>Shall be implemented by the MCAL consumer</td> </tr> <tr> <td><b>Called Context</b></td> <td>Interrupt</td> <td>This function would be invoked by driver in interrupt context. Also refer <a href="#">Flow Chart</a> for implementation of the ISR.</td> </tr> </tbody> </table>		<i>Description</i>	<i>Comment</i>	<b>Function Name</b>	Cdd_IpcNewMessageNotify	Is a symbolic name, integrators can specify desired name	<b>Syntax</b>	<code>void</code> <code>Cdd_IpcNewMessageNotify( uint32</code> <code>commId )</code>	Shall be implemented by the MCAL consumer	<b>Called Context</b>	Interrupt	This function would be invoked by driver in interrupt context. Also refer <a href="#">Flow Chart</a> for implementation of the ISR.
	<i>Description</i>	<i>Comment</i>													
<b>Function Name</b>	Cdd_IpcNewMessageNotify	Is a symbolic name, integrators can specify desired name													
<b>Syntax</b>	<code>void</code> <code>Cdd_IpcNewMessageNotify( uint32</code> <code>commId )</code>	Shall be implemented by the MCAL consumer													
<b>Called Context</b>	Interrupt	This function would be invoked by driver in interrupt context. Also refer <a href="#">Flow Chart</a> for implementation of the ISR.													



	Specification Category	Architecture Identifier	Description	
			<i>Description</i>	<i>Comment</i>
			<b>Reentrancy</b>	Non Reentrant
			<b>Parameter in</b>	uint32 commId  commId is the value specified by integrator while creating the communication channel <a href="#">id</a>
			<b>Return Value</b>	None
				NA

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>												
6	Function Definitions	 <a href="#">MCAL-7656</a> - MCAL Module: CDD IPC: <code>Cdd_IpcNewCtrlMessageNotify</code> <span style="background-color: #e0f2e0; border: 1px solid #e0f2e0; padding: 2px 5px;">PUBLISHED</span>	<p><code>Cdd_IpcNewCtrlMessageNotify</code></p> <p>Is a function implemented by the application, with prototype <code>void Cdd_IpcNewCtrlMessageNotify ( uint32 remoteProId )</code>, where <code>remoteProId</code> is remote processor ID <a href="#">Specified During Initialization</a></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th><i>Description</i></th> <th><i>Comment</i></th> </tr> </thead> <tbody> <tr> <td><b>Function Name</b></td> <td><code>Cdd_IpcNewCtrlMessageNotify</code></td> <td>Is a symbolic name, integrators can specify desired name</td> </tr> <tr> <td><b>Syntax</b></td> <td> <code>void</code>  <code>Cdd_IpcNewCtrlMessageNotify( uint32 remoteProId )</code> </td> <td>Shall be implemented by the MCAL consumer</td> </tr> <tr> <td><b>Called Context</b></td> <td>Interrupt</td> <td>This function would be invoked by driver in interrupt context. Also refer <a href="#">Control End Point</a></td> </tr> </tbody> </table>		<i>Description</i>	<i>Comment</i>	<b>Function Name</b>	<code>Cdd_IpcNewCtrlMessageNotify</code>	Is a symbolic name, integrators can specify desired name	<b>Syntax</b>	<code>void</code> <code>Cdd_IpcNewCtrlMessageNotify( uint32 remoteProId )</code>	Shall be implemented by the MCAL consumer	<b>Called Context</b>	Interrupt	This function would be invoked by driver in interrupt context. Also refer <a href="#">Control End Point</a>
	<i>Description</i>	<i>Comment</i>													
<b>Function Name</b>	<code>Cdd_IpcNewCtrlMessageNotify</code>	Is a symbolic name, integrators can specify desired name													
<b>Syntax</b>	<code>void</code> <code>Cdd_IpcNewCtrlMessageNotify( uint32 remoteProId )</code>	Shall be implemented by the MCAL consumer													
<b>Called Context</b>	Interrupt	This function would be invoked by driver in interrupt context. Also refer <a href="#">Control End Point</a>													



	Specification Category	Architecture Identifier	Description		
				Description	Comment
			<b>Reentrancy</b>	Non Reentrant	
			<b>Parameter in</b>	uint32 remoteProclId	One of the Remote processor ID Specified During Initialization
			<b>Return Value</b>	None	NA

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
7	Function Definitions	 <a href="#">MCAL-7657</a> - MCAL Module: CDD IPC: <code>Cdd_IpcInit</code> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>Cdd_IpcInit</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th><i>Description</i></th> <th><i>Comment</i></th> </tr> </thead> <tbody> <tr> <td><b>Service Name</b></td> <td>Cdd_IpcInit</td> <td>First API to be invoked to initialize the module</td> </tr> <tr> <td><b>Syntax</b></td> <td>Std_ReturnType  <code>Cdd_IpcInit(void)</code></td> <td>Service for CDD Initialization</td> </tr> <tr> <td><b>Service ID</b></td> <td>0x02</td> <td></td> </tr> <tr> <td><b>Sync / Async</b></td> <td>Sync</td> <td></td> </tr> <tr> <td><b>Reentrancy</b></td> <td>Non Reentrant</td> <td></td> </tr> </tbody> </table>		<i>Description</i>	<i>Comment</i>	<b>Service Name</b>	Cdd_IpcInit	First API to be invoked to initialize the module	<b>Syntax</b>	Std_ReturnType <code>Cdd_IpcInit(void)</code>	Service for CDD Initialization	<b>Service ID</b>	0x02		<b>Sync / Async</b>	Sync		<b>Reentrancy</b>	Non Reentrant	
	<i>Description</i>	<i>Comment</i>																			
<b>Service Name</b>	Cdd_IpcInit	First API to be invoked to initialize the module																			
<b>Syntax</b>	Std_ReturnType <code>Cdd_IpcInit(void)</code>	Service for CDD Initialization																			
<b>Service ID</b>	0x02																				
<b>Sync / Async</b>	Sync																				
<b>Reentrancy</b>	Non Reentrant																				



	Specification Category	Architecture Identifier	Description												
			<table border="1"><thead><tr><th></th><th><i>Description</i></th><th><i>Comment</i></th></tr></thead><tbody><tr><td><i>Parameter in</i></td><td>none</td><td>NA</td></tr><tr><td><i>Parameters out</i></td><td>none</td><td>NA</td></tr><tr><td><i>Return Value</i></td><td>Standard return type</td><td>E_OK or CDD_IPC_E_INIT_FAILED in case of initialization failure id, or CDD_IPC_E_ALREADY_INITIALIZED in case of reinitialization</td></tr></tbody></table>		<i>Description</i>	<i>Comment</i>	<i>Parameter in</i>	none	NA	<i>Parameters out</i>	none	NA	<i>Return Value</i>	Standard return type	E_OK or CDD_IPC_E_INIT_FAILED in case of initialization failure id, or CDD_IPC_E_ALREADY_INITIALIZED in case of reinitialization
	<i>Description</i>	<i>Comment</i>													
<i>Parameter in</i>	none	NA													
<i>Parameters out</i>	none	NA													
<i>Return Value</i>	Standard return type	E_OK or CDD_IPC_E_INIT_FAILED in case of initialization failure id, or CDD_IPC_E_ALREADY_INITIALIZED in case of reinitialization													

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
8	Function Definitions	 <a href="#">MCAL-7658</a> - MCAL Module: CDD IPC: <b>Cdd_IpcDeinit</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>Cdd_IpcDeinit</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th><i>Description</i></th> <th><i>Comment</i></th> </tr> </thead> <tbody> <tr> <td><b>Service Name</b></td> <td>Cdd_IpcDeinit</td> <td>Last API to be invoked to de initialize the module, can be turned OFF <a href="#">CddDeinitApi</a></td> </tr> <tr> <td><b>Syntax</b></td> <td>Std_ReturnType Cdd_IpcDeinit (void)</td> <td>Service for CDD Initialization</td> </tr> <tr> <td><b>Service ID</b></td> <td>0x08</td> <td></td> </tr> <tr> <td><b>Sync / Async</b></td> <td>Sync</td> <td></td> </tr> <tr> <td><b>Reentrancy</b></td> <td>Non Reentrant</td> <td></td> </tr> </tbody> </table>		<i>Description</i>	<i>Comment</i>	<b>Service Name</b>	Cdd_IpcDeinit	Last API to be invoked to de initialize the module, can be turned OFF <a href="#">CddDeinitApi</a>	<b>Syntax</b>	Std_ReturnType Cdd_IpcDeinit (void)	Service for CDD Initialization	<b>Service ID</b>	0x08		<b>Sync / Async</b>	Sync		<b>Reentrancy</b>	Non Reentrant	
	<i>Description</i>	<i>Comment</i>																			
<b>Service Name</b>	Cdd_IpcDeinit	Last API to be invoked to de initialize the module, can be turned OFF <a href="#">CddDeinitApi</a>																			
<b>Syntax</b>	Std_ReturnType Cdd_IpcDeinit (void)	Service for CDD Initialization																			
<b>Service ID</b>	0x08																				
<b>Sync / Async</b>	Sync																				
<b>Reentrancy</b>	Non Reentrant																				



	Specification Category	Architecture Identifier	Description		
				Description	Comment
			<b>Parameter in</b>	none	NA
			<b>Parameters out</b>	none	NA
			<b>Return Value</b>	Standard return type	E_OK

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
9	Function Definitions	 <a href="#">MCAL-7659</a> - MCAL Module: CDD IPC: <code>Cdd_IpcSendMsg</code> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p><code>Cdd_IpcSendMsg</code></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th><i>Description</i></th><th><i>Comment</i></th></tr> </thead> <tbody> <tr> <td><b>Service Name</b></td><td><code>Cdd_IpcSendMsg</code></td><td>Service for sending an message to remote cores</td></tr> <tr> <td><b>Syntax</b></td><td> <code>Std_ReturnType Cdd_IpcSendMsg(uint32 chId, void *pBuf, uint32 bufLen)</code> </td><td></td></tr> <tr> <td><b>Service ID</b></td><td>0x03</td><td></td></tr> <tr> <td><b>Sync / Async</b></td><td>Sync</td><td></td></tr> <tr> <td><b>Reentrancy</b></td><td>Non Reentrant</td><td></td></tr> </tbody> </table>		<i>Description</i>	<i>Comment</i>	<b>Service Name</b>	<code>Cdd_IpcSendMsg</code>	Service for sending an message to remote cores	<b>Syntax</b>	<code>Std_ReturnType Cdd_IpcSendMsg(uint32 chId, void *pBuf, uint32 bufLen)</code>		<b>Service ID</b>	0x03		<b>Sync / Async</b>	Sync		<b>Reentrancy</b>	Non Reentrant	
	<i>Description</i>	<i>Comment</i>																			
<b>Service Name</b>	<code>Cdd_IpcSendMsg</code>	Service for sending an message to remote cores																			
<b>Syntax</b>	<code>Std_ReturnType Cdd_IpcSendMsg(uint32 chId, void *pBuf, uint32 bufLen)</code>																				
<b>Service ID</b>	0x03																				
<b>Sync / Async</b>	Sync																				
<b>Reentrancy</b>	Non Reentrant																				



Specification Category	Architecture Identifier	Description		
			Description	Comment
		<b>Parameter in</b>	chId	chId Refers to communication ID specified while configuring this module, refer <a href="#">chId</a>
		<b>Parameter in</b>	pBuf	Non NULL_PTR that describes the message that has to sent
		<b>Parameter in</b>	bufLen	Message length in bytes
		<b>Return Value</b>	Standard return type	E_OK on successful transmission, CDD_IPC_E_SEND on error and CDD_IPC_E_UNINIT when initialized

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
10	Function Definitions	 <a href="#">MCAL-7660 - MCAL</a> Module: CDD IPC: <b>Cdd_IpcReceiveMsg</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>Cdd_IpcReceiveMsg</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th><i>Description</i></th><th><i>Comments</i></th></tr> </thead> <tbody> <tr> <td><b>Service Name</b></td><td>Cdd_IpcReceiveMsg</td><td>Service for reception of N bytes of data from remote cores</td></tr> <tr> <td><b>Syntax</b></td><td>Std_ReturnType Cdd_IpcReceiveMsg(uint32 chId, void *pBuf, uint32 bufLen)</td><td></td></tr> <tr> <td><b>Service ID</b></td><td>0x04</td><td></td></tr> <tr> <td><b>Sync / Async</b></td><td>Sync</td><td></td></tr> <tr> <td><b>Reentrancy</b></td><td>Non Reentrant</td><td></td></tr> </tbody> </table>		<i>Description</i>	<i>Comments</i>	<b>Service Name</b>	Cdd_IpcReceiveMsg	Service for reception of N bytes of data from remote cores	<b>Syntax</b>	Std_ReturnType Cdd_IpcReceiveMsg(uint32 chId, void *pBuf, uint32 bufLen)		<b>Service ID</b>	0x04		<b>Sync / Async</b>	Sync		<b>Reentrancy</b>	Non Reentrant	
	<i>Description</i>	<i>Comments</i>																			
<b>Service Name</b>	Cdd_IpcReceiveMsg	Service for reception of N bytes of data from remote cores																			
<b>Syntax</b>	Std_ReturnType Cdd_IpcReceiveMsg(uint32 chId, void *pBuf, uint32 bufLen)																				
<b>Service ID</b>	0x04																				
<b>Sync / Async</b>	Sync																				
<b>Reentrancy</b>	Non Reentrant																				

	Specification Category	Architecture Identifier	Description		
				<i>Description</i>	<i>Comments</i>
			<b>Parameter in</b>	chId	chId Refers to communication ID specified while configuring this module, refer <a href="#">chId</a>
			<b>Parameter in out</b>	pBuf	Non NULL_PTR that can hold the received message. Call shall ensure sufficient memory is available, shall be greater than or equal to maximum size specified in configuration. Refer <a href="#">maxMsgSize</a>
			<b>Parameter in</b>	bufLen	Message length in bytes
			<b>Return Value</b>	Standard return type	E_OK on successful reception, CDD_IPC_E_RECEIVE_RETRY on no messages, CDD_IPC_E_UNINIT, when uninitialized.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>															
11	Function Definitions	 <a href="#">MCAL-7661</a> - MCAL Module: CDD IPC: <b>Cdd_IpcAnnounce</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>Cdd_IpcAnnounce</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th><i>Description</i></th><th><i>Comments</i></th></tr> </thead> <tbody> <tr> <td><b>Service Name</b></td><td>Cdd_IpcAnnounce</td><td>Used to broadcast capabilities of this core to all other cores, can be turned OFF <a href="#">CddIpcAnnounceApi</a></td></tr> <tr> <td><b>Syntax</b></td><td>Std_ReturnType Cdd_IpcAnnounce(void *pBuf, uint32 chId)</td><td>Service broadcast of message to all cores for a given communication channel</td></tr> <tr> <td><b>Service ID</b></td><td>0x05</td><td></td></tr> <tr> <td><b>Sync / Async</b></td><td>Sync</td><td></td></tr> </tbody> </table>		<i>Description</i>	<i>Comments</i>	<b>Service Name</b>	Cdd_IpcAnnounce	Used to broadcast capabilities of this core to all other cores, can be turned OFF <a href="#">CddIpcAnnounceApi</a>	<b>Syntax</b>	Std_ReturnType Cdd_IpcAnnounce(void *pBuf, uint32 chId)	Service broadcast of message to all cores for a given communication channel	<b>Service ID</b>	0x05		<b>Sync / Async</b>	Sync	
	<i>Description</i>	<i>Comments</i>																
<b>Service Name</b>	Cdd_IpcAnnounce	Used to broadcast capabilities of this core to all other cores, can be turned OFF <a href="#">CddIpcAnnounceApi</a>																
<b>Syntax</b>	Std_ReturnType Cdd_IpcAnnounce(void *pBuf, uint32 chId)	Service broadcast of message to all cores for a given communication channel																
<b>Service ID</b>	0x05																	
<b>Sync / Async</b>	Sync																	



	Specification Category	Architecture Identifier	Description	
			<i>Description</i>	<i>Comments</i>
			<b>Reentrancy</b>	Non Reentrant
			<b>Parameter in</b>	pBuf Non NULL_PTR that describes the message that has to sent
			<b>Parameter in</b>	chld A Valid Communication Channel Identifier
			<b>Return Value</b>	Standard return type E_OK on successful transmission and CDD_IPC_E_SEND on error and CDD_IPC_E_UNINIT when initialized. Also check <a href="#">Constraint</a>

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>												
12	Function Definitions	 <a href="#">MCAL-7662</a> - MCAL Module: CDD IPC: <code>Cdd_IpcGetVersionInfo</code> <ul style="list-style-type: none"> <li>o <span style="color: green;">PUBLISHED</span></li> </ul>	<p><code>Cdd_IpcGetVersionInfo</code></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th><i>Description</i></th><th><i>Comments</i></th></tr> </thead> <tbody> <tr> <td style="text-align: center;"><i>Service Name</i></td><td><code>Cdd_IpcGetVersionInfo</code></td><td>Can potentially be turned OFF, via configuration parameter <a href="#">CddVersionInfoApi</a></td></tr> <tr> <td style="text-align: center;"><i>Syntax</i></td><td> <pre>void Cdd_IpcGetVersionInfo(Std_VersionInfo oType VersionInfoPtr)</pre> </td><td></td></tr> <tr> <td style="text-align: center;"><i>Service ID</i></td><td>0x01</td><td></td></tr> </tbody> </table>		<i>Description</i>	<i>Comments</i>	<i>Service Name</i>	<code>Cdd_IpcGetVersionInfo</code>	Can potentially be turned OFF, via configuration parameter <a href="#">CddVersionInfoApi</a>	<i>Syntax</i>	<pre>void Cdd_IpcGetVersionInfo(Std_VersionInfo oType VersionInfoPtr)</pre>		<i>Service ID</i>	0x01	
	<i>Description</i>	<i>Comments</i>													
<i>Service Name</i>	<code>Cdd_IpcGetVersionInfo</code>	Can potentially be turned OFF, via configuration parameter <a href="#">CddVersionInfoApi</a>													
<i>Syntax</i>	<pre>void Cdd_IpcGetVersionInfo(Std_VersionInfo oType VersionInfoPtr)</pre>														
<i>Service ID</i>	0x01														



	Specification Category	Architecture Identifier	Description		
				Description	Comments
			<b>Sync / Async</b>	Sync	
			<b>Reentrancy</b>	Reentrant	
			<b>Parameters out</b>	VersionInfoPtr	A pointer of type Std_VersionInfoType, which holds the read back values
			<b>Return Value</b>	None	

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
13	Function Definitions	 <a href="#">MCAL-7663</a> - MCAL Module: CDD IPC: <b>Cdd_IpcRegisterReadBack</b> <span style="background-color: #e0f2e0; border: 1px solid #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>Cdd_IpcRegisterReadBack</p> <table border="1"> <thead> <tr> <th></th><th><i>Description</i></th><th><i>Comments</i></th></tr> </thead> <tbody> <tr> <td><b>Service Name</b></td><td>Cdd_IpcRegisterReadBack</td><td>Can potentially be turned OFF</td></tr> <tr> <td><b>Syntax</b></td><td>Std_ReturnType  <code>Cdd_IpcRegisterReadBack(Cdd_IpcRegRbValues *RegRbPtr)</code></td><td><a href="#">Cdd_IpcRegRbValues</a> defines the type, that holds critical values. This service can be turned OFF <a href="#">CddRegisterReadBackApi</a></td></tr> <tr> <td><b>Service ID</b></td><td>0x07</td><td></td></tr> <tr> <td><b>Sync / Async</b></td><td>Sync</td><td></td></tr> <tr> <td><b>Reentrancy</b></td><td>Non Reentrant</td><td></td></tr> </tbody> </table>		<i>Description</i>	<i>Comments</i>	<b>Service Name</b>	Cdd_IpcRegisterReadBack	Can potentially be turned OFF	<b>Syntax</b>	Std_ReturnType <code>Cdd_IpcRegisterReadBack(Cdd_IpcRegRbValues *RegRbPtr)</code>	<a href="#">Cdd_IpcRegRbValues</a> defines the type, that holds critical values. This service can be turned OFF <a href="#">CddRegisterReadBackApi</a>	<b>Service ID</b>	0x07		<b>Sync / Async</b>	Sync		<b>Reentrancy</b>	Non Reentrant	
	<i>Description</i>	<i>Comments</i>																			
<b>Service Name</b>	Cdd_IpcRegisterReadBack	Can potentially be turned OFF																			
<b>Syntax</b>	Std_ReturnType <code>Cdd_IpcRegisterReadBack(Cdd_IpcRegRbValues *RegRbPtr)</code>	<a href="#">Cdd_IpcRegRbValues</a> defines the type, that holds critical values. This service can be turned OFF <a href="#">CddRegisterReadBackApi</a>																			
<b>Service ID</b>	0x07																				
<b>Sync / Async</b>	Sync																				
<b>Reentrancy</b>	Non Reentrant																				



Specification Category	Architecture Identifier	Description		
			<i>Description</i>	<i>Comments</i>
		<b>Parameters out</b>	RegRbPtr	A pointer of type <a href="#">Cdd_IpcRegRbValues</a> , which holds the read back values
		<b>Return Value</b>	Standard return type	E_OK or E_NOT_OK in case of error

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>															
14	Function Definitions	 <b>MCAL-7664</b> - MCAL Module: CDD IPC: Cdd_IpcReceiveCtrlMsg <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px 5px;">PUBLISHED</span>	<p>Cdd_IpcReceiveCtrlMsg</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th><i>Description</i></th> <th><i>Comments</i></th> </tr> </thead> <tbody> <tr> <td><b>Service Name</b></td> <td>Cdd_IpcReceiveCtrlMsg</td> <td>Service for reception of N bytes of control data from remote cores</td> </tr> <tr> <td><b>Syntax</b></td> <td> <pre>Std_ReturnType Cdd_IpcReceiveCtrlMsg(uint32 *pRemoteProcl, uint32 *pRemoteEndPt, void *pBuf, uint32 bufLen)</pre> </td> <td></td> </tr> <tr> <td><b>Service ID</b></td> <td>0x09</td> <td></td> </tr> <tr> <td><b>Sync / Async</b></td> <td>Sync</td> <td></td> </tr> </tbody> </table>		<i>Description</i>	<i>Comments</i>	<b>Service Name</b>	Cdd_IpcReceiveCtrlMsg	Service for reception of N bytes of control data from remote cores	<b>Syntax</b>	<pre>Std_ReturnType Cdd_IpcReceiveCtrlMsg(uint32 *pRemoteProcl, uint32 *pRemoteEndPt, void *pBuf, uint32 bufLen)</pre>		<b>Service ID</b>	0x09		<b>Sync / Async</b>	Sync	
	<i>Description</i>	<i>Comments</i>																
<b>Service Name</b>	Cdd_IpcReceiveCtrlMsg	Service for reception of N bytes of control data from remote cores																
<b>Syntax</b>	<pre>Std_ReturnType Cdd_IpcReceiveCtrlMsg(uint32 *pRemoteProcl, uint32 *pRemoteEndPt, void *pBuf, uint32 bufLen)</pre>																	
<b>Service ID</b>	0x09																	
<b>Sync / Async</b>	Sync																	

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>															
			<table border="1"> <thead> <tr> <th></th><th><i>Description</i></th><th><i>Comments</i></th></tr> </thead> <tbody> <tr> <td><b>Reentrancy</b></td><td>Non Reentrant</td><td></td></tr> <tr> <td><b>Parameter in out</b></td><td>pRemoteProclD</td><td>Holds the remote processor identifier, one of valid remote processors specified while initializing, refer <a href="#">RemoteProclD</a></td></tr> <tr> <td><b>Parameter in out</b></td><td>pRemoteEndPt</td><td>Holds the remote processor end point, that is the originator of this control message</td></tr> <tr> <td><b>Parameter in out</b></td><td>pBuf</td><td>Non NULL_PTR that can hold the received message. Call shall ensure sufficient memory is available, shall be greater than or equal to maximum size specified in configuration. Refer <a href="#">maxMsgSize</a></td></tr> </tbody> </table>		<i>Description</i>	<i>Comments</i>	<b>Reentrancy</b>	Non Reentrant		<b>Parameter in out</b>	pRemoteProclD	Holds the remote processor identifier, one of valid remote processors specified while initializing, refer <a href="#">RemoteProclD</a>	<b>Parameter in out</b>	pRemoteEndPt	Holds the remote processor end point, that is the originator of this control message	<b>Parameter in out</b>	pBuf	Non NULL_PTR that can hold the received message. Call shall ensure sufficient memory is available, shall be greater than or equal to maximum size specified in configuration. Refer <a href="#">maxMsgSize</a>
	<i>Description</i>	<i>Comments</i>																
<b>Reentrancy</b>	Non Reentrant																	
<b>Parameter in out</b>	pRemoteProclD	Holds the remote processor identifier, one of valid remote processors specified while initializing, refer <a href="#">RemoteProclD</a>																
<b>Parameter in out</b>	pRemoteEndPt	Holds the remote processor end point, that is the originator of this control message																
<b>Parameter in out</b>	pBuf	Non NULL_PTR that can hold the received message. Call shall ensure sufficient memory is available, shall be greater than or equal to maximum size specified in configuration. Refer <a href="#">maxMsgSize</a>																



	Specification Category	Architecture Identifier	Description		
				<i>Description</i>	<i>Comments</i>
			<b>Parameter in</b>	bufLen	Received message length in bytes
			<b>Return Value</b>	Standard return type	E_OK on successful reception, E_NOT_OK on no messages, CDD_IPC_E_UNINIT, when uninitialized and CDD_IPC_E_PARAM_POINTER when any one of the pointer is NULL

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
15	Function Definitions	 <a href="#">MCAL-7665</a> - MCAL Module: CDD IPC: <code>Cdd_IpcIsInitDone</code> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>Cdd_IpcIsInitDone</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th><th><i>Description</i></th><th><i>Comments</i></th></tr> </thead> <tbody> <tr> <td><b>Service Name</b></td><td><code>Cdd_IpcIsInitDone</code></td><td>Returns TRUE if <code>Cdd_ipcInit</code> is completed else FALSE</td></tr> <tr> <td><b>Syntax</b></td><td><code>boolean Cdd_IpcIsInitDone(void)</code></td><td></td></tr> <tr> <td><b>Service ID</b></td><td>0x0A</td><td></td></tr> <tr> <td><b>Sync / Async</b></td><td>Sync</td><td></td></tr> <tr> <td><b>Reentrancy</b></td><td>Reentrant</td><td></td></tr> </tbody> </table>		<i>Description</i>	<i>Comments</i>	<b>Service Name</b>	<code>Cdd_IpcIsInitDone</code>	Returns TRUE if <code>Cdd_ipcInit</code> is completed else FALSE	<b>Syntax</b>	<code>boolean Cdd_IpcIsInitDone(void)</code>		<b>Service ID</b>	0x0A		<b>Sync / Async</b>	Sync		<b>Reentrancy</b>	Reentrant	
	<i>Description</i>	<i>Comments</i>																			
<b>Service Name</b>	<code>Cdd_IpcIsInitDone</code>	Returns TRUE if <code>Cdd_ipcInit</code> is completed else FALSE																			
<b>Syntax</b>	<code>boolean Cdd_IpcIsInitDone(void)</code>																				
<b>Service ID</b>	0x0A																				
<b>Sync / Async</b>	Sync																				
<b>Reentrancy</b>	Reentrant																				



Specification Category	Architecture Identifier	Description		
		<i>Description</i>	<i>Comments</i>	
		<b>Return Value</b>	Boolean	Returns TRUE if Cdd_IpcInit was completed else FALSE

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																		
16	Function Definitions	 <b>MCAL-7666</b> - MCAL Module: CDD IPC: Cdd_IpcGetMaxMsgSize <span style="background-color: #e0f2e0; border-radius: 5px; padding: 2px;">PUBLISHED</span>	<p>Cdd_IpcGetMaxMsgSize</p> <table border="1"> <thead> <tr> <th></th> <th><b>Description</b></th> <th><b>Comments</b></th> </tr> </thead> <tbody> <tr> <td><b>Service Name</b></td> <td>Cdd_IpcGetMaxMsgSize</td> <td>Returns max msg size for the specified channel</td> </tr> <tr> <td><b>Syntax</b></td> <td>uint32 Cdd_IpcGetMaxMsgSize(uint32 chId)</td> <td></td> </tr> <tr> <td><b>Service ID</b></td> <td>0x0B</td> <td></td> </tr> <tr> <td><b>Sync / Async</b></td> <td>Sync</td> <td></td> </tr> <tr> <td><b>Reentrancy</b></td> <td>Reentrant</td> <td></td> </tr> </tbody> </table>		<b>Description</b>	<b>Comments</b>	<b>Service Name</b>	Cdd_IpcGetMaxMsgSize	Returns max msg size for the specified channel	<b>Syntax</b>	uint32 Cdd_IpcGetMaxMsgSize(uint32 chId)		<b>Service ID</b>	0x0B		<b>Sync / Async</b>	Sync		<b>Reentrancy</b>	Reentrant	
	<b>Description</b>	<b>Comments</b>																			
<b>Service Name</b>	Cdd_IpcGetMaxMsgSize	Returns max msg size for the specified channel																			
<b>Syntax</b>	uint32 Cdd_IpcGetMaxMsgSize(uint32 chId)																				
<b>Service ID</b>	0x0B																				
<b>Sync / Async</b>	Sync																				
<b>Reentrancy</b>	Reentrant																				

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>		
				<i>Description</i>	<i>Comments</i>
			<b>Return Value</b>	uint32	Returns max msg size for the specified channel
17	Configuration Specification	 <a href="#">MCAL-7227</a> - Cdd_IpcConfigType : Variant <span style="background-color: #d9e1f2; border: 1px solid #d9e1f2; padding: 2px 5px;">PUBLISHED</span>	CDD IPC shall use Pre-Compile Time variant.		

## 8.11 FLS Driver

This Flash driver uses OSPI module to transfer data between the system and flash device. Below diagram shows the integration between the two hardware IPs that are used by the Flash Driver: FSS and OSPI.

The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW FLS Driver Specification](#).



	Specification Category	Architecture Identifier	Description
1	Safety Process/ Diagnostic	 <a href="#">MCAL-7283 - MCAL</a> Module : FLS : Safety Diagnostic <span>PUBLISHED</span>	Following the Safety manual OSPI IP and FMA review, the required safety diagnostic features should be added.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
2	File Structure	 <b>MCAL-7370 -</b> MCAL Module : Flash Driver : Complies to the file structure specified in Autosar Reqs <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	[SWS_Fls_00107] The Fls module shall comply with section 5.1 in <a href="#">AUTOSAR BSW FLS Driver Specification</a> .

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
3	General Behavior	 MCAL-7669 - MCAL Module : Flash Driver : General Behavior <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	The FIs module shall follow the general design rules as described in section 7.1 of <a href="#">AUTOSAR BSW FLS Driver Specification</a>
4	Error Classification and Detection	 MCAL-7670 - MCAL Module : Flash Driver : Error Classification <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	FLS Module shall report Runtime Errors and Transient Faults.  DetReportRuntimeError and Det_ReportError services shall be used.

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
5	Imported Types	 MCAL-7277 - MCAL Module : Flash Driver : Imported Types <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	SWS_Fls_00248] Fls module shall use imported types Std_Types and MemIF.
6	Type Definitions	 MCAL-7671 - MCAL Module : Flash Driver : Type Definitions <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	FLS module shall define types as described in section 8.2 of <a href="#">AUTOSAR BSW FLS Driver Specification</a> .

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
7	Function Definitions	 <b>MCAL-7354 -</b> MCAL Module : Flash Driver : Func Defs : Fls_Init 	<p>[SWS_Fls_00249] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Fls_Init</td></tr> <tr> <td><b>Syntax:</b></td><td>void Fls_Init(                   const Fls_ConfigType* ConfigPtr                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x00</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Non Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>ConfigPtr      Pointer to flash driver configuration set.</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>Initializes the Flash Driver.</td></tr> </table>	<b>Service name:</b>	Fls_Init	<b>Syntax:</b>	void Fls_Init( const Fls_ConfigType* ConfigPtr )	<b>Service ID[hex]:</b>	0x00	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	ConfigPtr      Pointer to flash driver configuration set.	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Initializes the Flash Driver.
<b>Service name:</b>	Fls_Init																						
<b>Syntax:</b>	void Fls_Init( const Fls_ConfigType* ConfigPtr )																						
<b>Service ID[hex]:</b>	0x00																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	ConfigPtr      Pointer to flash driver configuration set.																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Initializes the Flash Driver.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
8	Function Definitions	 MCAL-7314 - MCAL Module : Flash Driver : Func Defs : FLs_Erase <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	[SWS_Fls_00250] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a> .

Specification Category	Architecture Identifier	Description																		
		<p><b>Service name:</b> Fls_Erase</p> <p><b>Syntax:</b> Std_ReturnType Fls_Erase(                   Fls_AddressType TargetAddress,                   Fls_LengthType Length                   )</p> <p><b>Service ID[hex]:</b> 0x01</p> <p><b>Sync/Async:</b> Asynchronous</p> <p><b>Reentrancy:</b> Non Reentrant</p> <table> <tr> <td rowspan="2"><b>Parameters (in):</b></td> <td>TargetAddress</td> <td>Target address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1</td> </tr> <tr> <td>Length</td> <td>Number of bytes to erase Min.: 1 Max.: FLS_SIZE - TargetAddress</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: erase command has been accepted E_NOT_OK: erase command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Erases flash sector(s).</td></tr> </table>	<b>Parameters (in):</b>	TargetAddress	Target address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1	Length	Number of bytes to erase Min.: 1 Max.: FLS_SIZE - TargetAddress	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Std_ReturnType	E_OK: erase command has been accepted E_NOT_OK: erase command has not been accepted	<b>Description:</b>	Erases flash sector(s).		
<b>Parameters (in):</b>	TargetAddress	Target address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1																		
	Length	Number of bytes to erase Min.: 1 Max.: FLS_SIZE - TargetAddress																		
<b>Parameters (inout):</b>	None																			
<b>Parameters (out):</b>	None																			
<b>Return value:</b>	Std_ReturnType	E_OK: erase command has been accepted E_NOT_OK: erase command has not been accepted																		
<b>Description:</b>	Erases flash sector(s).																			

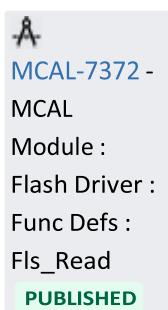
	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
9	Function Definitions	 MCAL-7333 - MCAL Module : Flash Driver : Func Defs : Fls_Write <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	[SWS_Fls_00251] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a> .

Specification Category	Architecture Identifier	Description																						
		<p><b>Service name:</b> Fls_Write</p> <p><b>Syntax:</b></p> <pre>Std_ReturnType Fls_Write(     Fls_AddressType TargetAddress,     const uint8* SourceAddressPtr,     Fls_LengthType Length )</pre> <p><b>Service ID[hex]:</b> 0x02</p> <p><b>Sync/Async:</b> Asynchronous</p> <p><b>Reentrancy:</b> Non Reentrant</p> <table border="1"> <tr> <td rowspan="3"><b>Parameters (in):</b></td> <td>TargetAddress</td> <td>Target address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1</td> </tr> <tr> <td>SourceAddressPtr</td> <td>Pointer to source data buffer</td> </tr> <tr> <td>Length</td> <td>Number of bytes to write Min.: 1 Max.: FLS_SIZE - TargetAddress</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> <td></td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> <td></td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: write command has been accepted E_NOT_OK: write command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="3">Writes one or more complete flash pages.</td></tr> </table>	<b>Parameters (in):</b>	TargetAddress	Target address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1	SourceAddressPtr	Pointer to source data buffer	Length	Number of bytes to write Min.: 1 Max.: FLS_SIZE - TargetAddress	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Std_ReturnType	E_OK: write command has been accepted E_NOT_OK: write command has not been accepted	<b>Description:</b>	Writes one or more complete flash pages.				
<b>Parameters (in):</b>	TargetAddress	Target address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1																						
	SourceAddressPtr	Pointer to source data buffer																						
	Length	Number of bytes to write Min.: 1 Max.: FLS_SIZE - TargetAddress																						
<b>Parameters (inout):</b>	None																							
<b>Parameters (out):</b>	None																							
<b>Return value:</b>	Std_ReturnType	E_OK: write command has been accepted E_NOT_OK: write command has not been accepted																						
<b>Description:</b>	Writes one or more complete flash pages.																							

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
10	Function Definitions	 <b>MCAL-7344 -</b> MCAL Module : Flash Driver : Func Defs : Fls_Cancel <b>PUBLISHED</b>	<p>[SWS_Fls_00252] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td> <td>Fls_Cancel</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Fls_Cancel(                   void                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x03</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Cancels an ongoing job.</td> </tr> </table>	<b>Service name:</b>	Fls_Cancel	<b>Syntax:</b>	void Fls_Cancel( void )	<b>Service ID[hex]:</b>	0x03	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	Cancels an ongoing job.
<b>Service name:</b>	Fls_Cancel																						
<b>Syntax:</b>	void Fls_Cancel( void )																						
<b>Service ID[hex]:</b>	0x03																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	Cancels an ongoing job.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
11	Function Definitions	 <a href="#">MCAL-7424 - MCAL</a> Module : Flash Driver : Func Defs : Fls_GetStatus <span style="background-color: #e0f2e0; border: 1px solid black; padding: 2px;">PUBLISHED</span>	<p>[SWS_Fls_00253] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Fls_GetStatus</td> </tr> <tr> <td><b>Syntax:</b></td> <td>MemIf_StatusType Fls_GetStatus (     void )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x04</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>MemIf_StatusType</td> </tr> <tr> <td><b>Description:</b></td> <td>Returns the driver state.</td> </tr> </table>	<b>Service name:</b>	Fls_GetStatus	<b>Syntax:</b>	MemIf_StatusType Fls_GetStatus ( void )	<b>Service ID[hex]:</b>	0x04	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	MemIf_StatusType	<b>Description:</b>	Returns the driver state.
<b>Service name:</b>	Fls_GetStatus																						
<b>Syntax:</b>	MemIf_StatusType Fls_GetStatus ( void )																						
<b>Service ID[hex]:</b>	0x04																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	MemIf_StatusType																						
<b>Description:</b>	Returns the driver state.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
12	Function Definitions	 <b>MCAL-7377 -</b> MCAL Module : Flash Driver : Func Defs : Fls_GetJobRe sult <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Fls_00254] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Fls_GetJobResult</td> </tr> <tr> <td><b>Syntax:</b></td> <td>MemIf_JobResultType Fls_GetJobResult(                   void                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x05</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>MemIf_JobResultType</td> </tr> <tr> <td><b>Description:</b></td> <td>Returns the result of the last job.</td> </tr> </table>	<b>Service name:</b>	Fls_GetJobResult	<b>Syntax:</b>	MemIf_JobResultType Fls_GetJobResult( void )	<b>Service ID[hex]:</b>	0x05	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	MemIf_JobResultType	<b>Description:</b>	Returns the result of the last job.
<b>Service name:</b>	Fls_GetJobResult																						
<b>Syntax:</b>	MemIf_JobResultType Fls_GetJobResult( void )																						
<b>Service ID[hex]:</b>	0x05																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	MemIf_JobResultType																						
<b>Description:</b>	Returns the result of the last job.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																
13	Function Definitions	 <b>MCAL-7372 -</b> MCAL Module : Flash Driver : Func Defs : Fls_Read <b>PUBLISHED</b>	<p>[SWS_Fls_00256] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td> <td>Fls_Read</td> </tr> <tr> <td><b>Syntax:</b></td> <td> <pre>Std_ReturnType Fls_Read(     Fls_AddressType SourceAddress,     uint8* TargetAddressPtr,     Fls_LengthType Length )</pre> </td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x07</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Asynchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td> <table border="1"> <tr> <td>SourceAddress</td> <td>Source address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1</td> </tr> <tr> <td>Length</td> <td>Number of bytes to read Min.: 1</td> </tr> </table> </td> </tr> </table>	<b>Service name:</b>	Fls_Read	<b>Syntax:</b>	<pre>Std_ReturnType Fls_Read(     Fls_AddressType SourceAddress,     uint8* TargetAddressPtr,     Fls_LengthType Length )</pre>	<b>Service ID[hex]:</b>	0x07	<b>Sync/Async:</b>	Asynchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	<table border="1"> <tr> <td>SourceAddress</td> <td>Source address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1</td> </tr> <tr> <td>Length</td> <td>Number of bytes to read Min.: 1</td> </tr> </table>	SourceAddress	Source address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1	Length	Number of bytes to read Min.: 1
<b>Service name:</b>	Fls_Read																		
<b>Syntax:</b>	<pre>Std_ReturnType Fls_Read(     Fls_AddressType SourceAddress,     uint8* TargetAddressPtr,     Fls_LengthType Length )</pre>																		
<b>Service ID[hex]:</b>	0x07																		
<b>Sync/Async:</b>	Asynchronous																		
<b>Reentrancy:</b>	Non Reentrant																		
<b>Parameters (in):</b>	<table border="1"> <tr> <td>SourceAddress</td> <td>Source address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1</td> </tr> <tr> <td>Length</td> <td>Number of bytes to read Min.: 1</td> </tr> </table>	SourceAddress	Source address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1	Length	Number of bytes to read Min.: 1														
SourceAddress	Source address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1																		
Length	Number of bytes to read Min.: 1																		

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>															
			<table border="1"> <tr> <td></td><td></td><td>Max.: FLS_SIZE - SourceAddress</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td><td></td></tr> <tr> <td><b>Parameters (out):</b></td><td>TargetAddressPtr</td><td>Pointer to target data buffer</td></tr> <tr> <td><b>Return value:</b></td><td>Std_ReturnType</td><td>E_OK: read command has been accepted E_NOT_OK: read command has not been accepted</td></tr> <tr> <td><b>Description:</b></td><td colspan="2">Reads from flash memory.</td></tr> </table>			Max.: FLS_SIZE - SourceAddress	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	TargetAddressPtr	Pointer to target data buffer	<b>Return value:</b>	Std_ReturnType	E_OK: read command has been accepted E_NOT_OK: read command has not been accepted	<b>Description:</b>	Reads from flash memory.	
		Max.: FLS_SIZE - SourceAddress																
<b>Parameters (inout):</b>	None																	
<b>Parameters (out):</b>	TargetAddressPtr	Pointer to target data buffer																
<b>Return value:</b>	Std_ReturnType	E_OK: read command has been accepted E_NOT_OK: read command has not been accepted																
<b>Description:</b>	Reads from flash memory.																	
14	Function Definitions	 <a href="#">MCAL-7331</a> - MCAL Module : Flash Driver : Func Defs : Fls_Compare <b>PUBLISHED</b>	[SWS_Fls_00257] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a> .															

Specification Category	Architecture Identifier	Description																				
		<p><b>Service name:</b> Fls_Compare</p> <p><b>Syntax:</b></p> <pre>Std_ReturnType Fls_Compare(     Fls_AddressType SourceAddress,     const uint8* TargetAddressPtr,     Fls_LengthType Length )</pre> <p><b>Service ID[hex]:</b> 0x08</p> <p><b>Sync/Async:</b> Asynchronous</p> <p><b>Reentrancy:</b> Non Reentrant</p> <table border="1"> <tr> <td rowspan="3"><b>Parameters (in):</b></td> <td>SourceAddress</td> <td>Source address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1</td> </tr> <tr> <td>TargetAddressPtr</td> <td>Pointer to target data buffer</td> </tr> <tr> <td>Length</td> <td>Number of bytes to compare Min.: 1 Max.: FLS_SIZE - SourceAddress</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> <td></td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> <td></td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: compare command has been accepted E_NOT_OK: compare command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">Compares the contents of an area of flash memory with that of an application data buffer.</td></tr> </table>	<b>Parameters (in):</b>	SourceAddress	Source address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1	TargetAddressPtr	Pointer to target data buffer	Length	Number of bytes to compare Min.: 1 Max.: FLS_SIZE - SourceAddress	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Std_ReturnType	E_OK: compare command has been accepted E_NOT_OK: compare command has not been accepted	<b>Description:</b>	Compares the contents of an area of flash memory with that of an application data buffer.		
<b>Parameters (in):</b>	SourceAddress	Source address in flash memory. This address offset will be added to the flash memory base address. Min.: 0 Max.: FLS_SIZE - 1																				
	TargetAddressPtr	Pointer to target data buffer																				
	Length	Number of bytes to compare Min.: 1 Max.: FLS_SIZE - SourceAddress																				
<b>Parameters (inout):</b>	None																					
<b>Parameters (out):</b>	None																					
<b>Return value:</b>	Std_ReturnType	E_OK: compare command has been accepted E_NOT_OK: compare command has not been accepted																				
<b>Description:</b>	Compares the contents of an area of flash memory with that of an application data buffer.																					

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																						
15	Function Definitions	 <b>MCAL-7397 -</b> MCAL Module : Flash Driver : Func Defs : Fls_GetVersionInfo <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Fls_00259] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Fls_GetVersionInfo</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Fls_GetVersionInfo(</td> </tr> <tr> <td></td> <td>                  Std_VersionInfoType* VersioninfoPtr                 )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x10</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>VersioninfoPtr Pointer to where to store the version information of this module.</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>Returns the version information of this module.</td> </tr> </table>	<b>Service name:</b>	Fls_GetVersionInfo	<b>Syntax:</b>	void Fls_GetVersionInfo(		Std_VersionInfoType* VersioninfoPtr )	<b>Service ID[hex]:</b>	0x10	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	VersioninfoPtr Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	Returns the version information of this module.
<b>Service name:</b>	Fls_GetVersionInfo																								
<b>Syntax:</b>	void Fls_GetVersionInfo(																								
	Std_VersionInfoType* VersioninfoPtr )																								
<b>Service ID[hex]:</b>	0x10																								
<b>Sync/Async:</b>	Synchronous																								
<b>Reentrancy:</b>	Reentrant																								
<b>Parameters (in):</b>	None																								
<b>Parameters (inout):</b>	None																								
<b>Parameters (out):</b>	VersioninfoPtr Pointer to where to store the version information of this module.																								
<b>Return value:</b>	None																								
<b>Description:</b>	Returns the version information of this module.																								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
16	Function Definitions	 <b>MCAL-7250 - MCAL</b> Module : Flash Driver : Func_Defs : Fls_BlkCk <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	[SWS_Fls_00371] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a> .

Specification Category	Architecture Identifier	Description																					
		<p><b>Service name:</b> Fls_BankCheck</p> <p><b>Syntax:</b></p> <pre>Std_ReturnType Fls_BankCheck(     Fls_AddressType TargetAddress,     Fls_LengthType Length )</pre> <p><b>Service ID[hex]:</b> 0x0a</p> <p><b>Sync/Async:</b> Asynchronous</p> <p><b>Reentrancy:</b> Non Reentrant</p> <table border="1"> <tr> <td data-bbox="660 779 893 922"><b>Parameters (in):</b></td><td data-bbox="893 779 1185 922">TargetAddress</td><td data-bbox="1185 779 2061 922">Address in flash memory from which the blank check should be started. Min.: 0 Max.: FLS_SIZE - 1</td></tr> <tr> <td data-bbox="660 922 893 1029"></td><td data-bbox="893 922 1185 1029">Length</td><td data-bbox="1185 922 2061 1029">Number of bytes to be checked for erase pattern. Min.: 1 Max.: FLS_SIZE - TargetAddress</td></tr> <tr> <td data-bbox="660 1029 893 1108"><b>Parameters (inout):</b></td><td colspan="2" data-bbox="893 1029 1185 1108">None</td></tr> <tr> <td data-bbox="660 1108 893 1156"><b>Parameters (out):</b></td><td colspan="2" data-bbox="893 1108 1185 1156">None</td></tr> <tr> <td data-bbox="660 1156 893 1298"><b>Return value:</b></td><td data-bbox="893 1156 1185 1298">Std_ReturnType</td><td data-bbox="1185 1156 2061 1298">E_OK: request for blank checking has been accepted by the module E_NOT_OK: request for blank checking has not been accepted by the module</td></tr> <tr> <td data-bbox="660 1298 893 1457"><b>Description:</b></td><td colspan="3" data-bbox="893 1298 2061 1457">The function Fls_BankCheck shall verify, whether a given memory area has been erased but not (yet) programmed. The function shall limit the maximum number of checked flash cells per main function cycle to the configured value FlsMaxReadNormalMode or FlsMaxReadFastMode respectively.</td></tr> </table>	<b>Parameters (in):</b>	TargetAddress	Address in flash memory from which the blank check should be started. Min.: 0 Max.: FLS_SIZE - 1		Length	Number of bytes to be checked for erase pattern. Min.: 1 Max.: FLS_SIZE - TargetAddress	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	Std_ReturnType	E_OK: request for blank checking has been accepted by the module E_NOT_OK: request for blank checking has not been accepted by the module	<b>Description:</b>	The function Fls_BankCheck shall verify, whether a given memory area has been erased but not (yet) programmed. The function shall limit the maximum number of checked flash cells per main function cycle to the configured value FlsMaxReadNormalMode or FlsMaxReadFastMode respectively.				
<b>Parameters (in):</b>	TargetAddress	Address in flash memory from which the blank check should be started. Min.: 0 Max.: FLS_SIZE - 1																					
	Length	Number of bytes to be checked for erase pattern. Min.: 1 Max.: FLS_SIZE - TargetAddress																					
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	Std_ReturnType	E_OK: request for blank checking has been accepted by the module E_NOT_OK: request for blank checking has not been accepted by the module																					
<b>Description:</b>	The function Fls_BankCheck shall verify, whether a given memory area has been erased but not (yet) programmed. The function shall limit the maximum number of checked flash cells per main function cycle to the configured value FlsMaxReadNormalMode or FlsMaxReadFastMode respectively.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>								
17	Function Definitions	 <b>MCAL-7309 -</b> MCAL Module : Flash Driver : Fls_MainFunc tion <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Fls_00255] shall be implemented and follow <a href="#">AUTOSAR BSW FLS Driver Specification</a>. This shall be a scheduled function.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Service name:</b></td> <td style="padding: 2px;">Fls_MainFunction</td> </tr> <tr> <td style="padding: 2px;"><b>Syntax:</b></td> <td style="padding: 2px;">void Fls_MainFunction(                   void                   )                   </td> </tr> <tr> <td style="padding: 2px;"><b>Service ID[hex]:</b></td> <td style="padding: 2px;">0x06</td> </tr> <tr> <td style="padding: 2px;"><b>Description:</b></td> <td style="padding: 2px;">Performs the processing of jobs.</td> </tr> </table>	<b>Service name:</b>	Fls_MainFunction	<b>Syntax:</b>	void Fls_MainFunction( void ) 	<b>Service ID[hex]:</b>	0x06	<b>Description:</b>	Performs the processing of jobs.
<b>Service name:</b>	Fls_MainFunction										
<b>Syntax:</b>	void Fls_MainFunction( void ) 										
<b>Service ID[hex]:</b>	0x06										
<b>Description:</b>	Performs the processing of jobs.										
18	Configurable Interfaces	 <b>MCAL-7672 -</b> MCAL Module : Flash Driver : Configurable Interfaces <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	Fls module shall use the Fee Module Stubs to target notification functions.								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
19	Sequence Diagrams	 <a href="#">MCAL-7673 - MCAL</a> Module : Flash Driver : Sequence Diagram <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	FLS module shall follow the sequence diagrams as described in section 9 of <a href="#">AUTOSAR BSW FLS Driver Specification</a> .

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
20	Configuration Specification	 <a href="#">MCAL-7382 - MCAL</a> Module : Flash Driver : Config : Fls : Configuration of Fls <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	ECUC_Fls_00001 - Fls Module shall support VARIANT-POSTBUILD or VARIANT-PRE-COMPIL

## 8.12 MCU Driver

The MCU module provides services for basic microcontroller initialization, power down functionality, reset and microcontroller specific functions required by other MCAL modules.

The SWS numbers in below table are in reference to the specification described in (1) [AUTOSAR BSW MCU Driver Specification](#).

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	Safety Process/ Diagnostic	MCAL-8526	Following the Safety manual IP and FMA review, the required safety diagnostic features should be added.
2	File Structure	 <b>MCAL-8527 -</b> MCAL Module : MCU Driver : Complies to the file structure specified in Autosar Reqs <b>PUBLISHED</b>	The Mcu module shall comply with section 5.1 in <a href="#">AUTOSAR BSW MCU Driver Specification</a> .

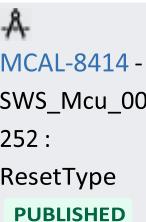
	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
3	General Behavior	 <b>MCAL-8528 - MCAL</b> Module : MCU Driver : General Behavior <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	The Mcu module shall follow the general design rules as described in section 7.1 of <a href="#">AUTOSAR BSW MCU Driver Specification</a>
4	Error Classification and Detection	 <b>MCAL-8529 - MCAL</b> Module : MCU Driver : Error Classification <span style="background-color: #00AEEF; color: white; padding: 2px;">PUBLISHED</span>	MCU Module shall report Runtime Errors and Transient Faults.  DetReportRuntimeError and Det_ReportError services shall be used.

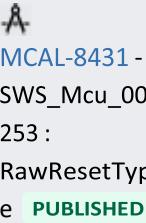
	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
5	Imported Types	 MCAL-8530 - MCAL Module : MCU Driver : Imported Types <b>PUBLISHED</b>	[SWS_Mcu_00152] Module Imported Type: Dem_and_Std_Types

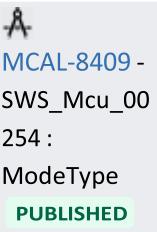
	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>												
6	Type Definitions	 MCAL-8415 - SWS_Mcu_00249 : ConfigType <b>PUBLISHED</b>	<p>[SWS_Mcu_00249] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1"> <tr> <td><b>Name:</b></td> <td colspan="2">Mcu_ConfigType</td> </tr> <tr> <td><b>Type:</b></td> <td colspan="2">Structure</td> </tr> <tr> <td><b>Range:</b></td> <td>Hardware dependent structure</td> <td>A structure to hold the MCU driver configuration.</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">A pointer to such a structure is provided to the MCU initialization routines for configuration.</td> </tr> </table>	<b>Name:</b>	Mcu_ConfigType		<b>Type:</b>	Structure		<b>Range:</b>	Hardware dependent structure	A structure to hold the MCU driver configuration.	<b>Description:</b>	A pointer to such a structure is provided to the MCU initialization routines for configuration.	
<b>Name:</b>	Mcu_ConfigType														
<b>Type:</b>	Structure														
<b>Range:</b>	Hardware dependent structure	A structure to hold the MCU driver configuration.													
<b>Description:</b>	A pointer to such a structure is provided to the MCU initialization routines for configuration.														

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>								
7	Type Definitions	 MCAL-8460 - SWS_Mcu_00 250 : PllStatusType <span style="background-color: #e0f2e0; border: 1px solid black; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00250] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Name:</b></td> <td style="padding: 2px;">Mcu_PllStatusType</td> </tr> <tr> <td style="padding: 2px;"><b>Type:</b></td> <td style="padding: 2px;">Enumeration</td> </tr> <tr> <td style="padding: 2px;"><b>Range:</b></td> <td style="padding: 2px;">           MCU_PLL_LOCKED -- PLL is locked            MCU_PLL_UNLOCKED -- PLL is unlocked            MCU_PLL_STATUS_UNDEFINED -- PLL Status is unknown         </td> </tr> <tr> <td style="padding: 2px;"><b>Description:</b></td> <td style="padding: 2px;">This is a status value returned by the function Mcu_GetPllStatus of the MCU module.</td> </tr> </table>	<b>Name:</b>	Mcu_PllStatusType	<b>Type:</b>	Enumeration	<b>Range:</b>	MCU_PLL_LOCKED -- PLL is locked MCU_PLL_UNLOCKED -- PLL is unlocked MCU_PLL_STATUS_UNDEFINED -- PLL Status is unknown	<b>Description:</b>	This is a status value returned by the function Mcu_GetPllStatus of the MCU module.
<b>Name:</b>	Mcu_PllStatusType										
<b>Type:</b>	Enumeration										
<b>Range:</b>	MCU_PLL_LOCKED -- PLL is locked MCU_PLL_UNLOCKED -- PLL is unlocked MCU_PLL_STATUS_UNDEFINED -- PLL Status is unknown										
<b>Description:</b>	This is a status value returned by the function Mcu_GetPllStatus of the MCU module.										

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>									
8	Type Definitions	 <a href="#">MCAL-8464 - SWS_Mcu_00251 : ClockType</a> <span>PUBLISHED</span>	<p>[SWS_Mcu_00251] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1"> <tr> <td><b>Name:</b></td><td>Mcu_ClockType</td></tr> <tr> <td><b>Type:</b></td><td>uint</td></tr> <tr> <td><b>Range:</b></td><td>0 .. &lt;number of clock settings&gt;- 1</td><td>-- The range is dependent on the number of different clock settings provided in the configuration structure. The type shall be chosen depending on MCU platform for best performance.</td></tr> <tr> <td><b>Description:</b></td><td>Specifies the identification (ID) for a clock setting, which is configured in the configuration structure</td></tr> </table>	<b>Name:</b>	Mcu_ClockType	<b>Type:</b>	uint	<b>Range:</b>	0 .. <number of clock settings>- 1	-- The range is dependent on the number of different clock settings provided in the configuration structure. The type shall be chosen depending on MCU platform for best performance.	<b>Description:</b>	Specifies the identification (ID) for a clock setting, which is configured in the configuration structure
<b>Name:</b>	Mcu_ClockType											
<b>Type:</b>	uint											
<b>Range:</b>	0 .. <number of clock settings>- 1	-- The range is dependent on the number of different clock settings provided in the configuration structure. The type shall be chosen depending on MCU platform for best performance.										
<b>Description:</b>	Specifies the identification (ID) for a clock setting, which is configured in the configuration structure											

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																												
9	Type Definitions	 <b>MCAL-8414 - SWS_Mcu_00 252 : ResetType</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00252] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Name:</b></td> <td colspan="3">Mcu_ResetType</td> </tr> <tr> <td><b>Type:</b></td> <td colspan="3">Enumeration</td> </tr> <tr> <td><b>Range:</b></td> <td>MCU_POWER_ON_RESET</td> <td>--</td> <td>Power On Reset (default)</td> </tr> <tr> <td></td> <td>MCU_WATCHDOG_RESET</td> <td>--</td> <td>Internal Watchdog Timer Reset</td> </tr> <tr> <td></td> <td>MCU_SW_RESET</td> <td>--</td> <td>Software Reset</td> </tr> <tr> <td></td> <td>MCU_RESET_UNDEFINED</td> <td>--</td> <td>Reset is undefined</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="3">This is the type of the reset enumerator containing the subset of reset types. It is not required that all reset types are supported by hardware.</td> </tr> </table>	<b>Name:</b>	Mcu_ResetType			<b>Type:</b>	Enumeration			<b>Range:</b>	MCU_POWER_ON_RESET	--	Power On Reset (default)		MCU_WATCHDOG_RESET	--	Internal Watchdog Timer Reset		MCU_SW_RESET	--	Software Reset		MCU_RESET_UNDEFINED	--	Reset is undefined	<b>Description:</b>	This is the type of the reset enumerator containing the subset of reset types. It is not required that all reset types are supported by hardware.		
<b>Name:</b>	Mcu_ResetType																														
<b>Type:</b>	Enumeration																														
<b>Range:</b>	MCU_POWER_ON_RESET	--	Power On Reset (default)																												
	MCU_WATCHDOG_RESET	--	Internal Watchdog Timer Reset																												
	MCU_SW_RESET	--	Software Reset																												
	MCU_RESET_UNDEFINED	--	Reset is undefined																												
<b>Description:</b>	This is the type of the reset enumerator containing the subset of reset types. It is not required that all reset types are supported by hardware.																														

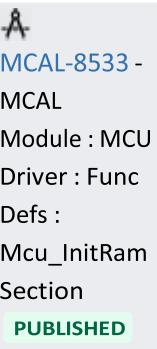
	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																
10	Type Definitions	 <b>MCAL-8431 - SWS_Mcu_00 253 : RawResetType</b> <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00253] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Name:</b></td> <td colspan="3">Mcu_RawResetType</td> </tr> <tr> <td><b>Type:</b></td> <td colspan="3">uint</td> </tr> <tr> <td><b>Range:</b></td> <td>MCU dependent register value</td> <td>--</td> <td>The type shall be chosen depending on MCU platform for best performance.</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="3">This type specifies the reset reason in raw register format read from a reset status register.</td> </tr> </table>	<b>Name:</b>	Mcu_RawResetType			<b>Type:</b>	uint			<b>Range:</b>	MCU dependent register value	--	The type shall be chosen depending on MCU platform for best performance.	<b>Description:</b>	This type specifies the reset reason in raw register format read from a reset status register.		
<b>Name:</b>	Mcu_RawResetType																		
<b>Type:</b>	uint																		
<b>Range:</b>	MCU dependent register value	--	The type shall be chosen depending on MCU platform for best performance.																
<b>Description:</b>	This type specifies the reset reason in raw register format read from a reset status register.																		

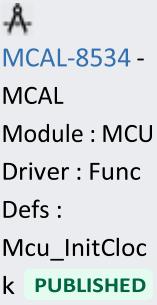
	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>										
11	Type Definitions	 <b>MCAL-8409 - SWS_Mcu_00 254 :</b> ModeType <b>PUBLISHED</b>	<p>[SWS_Mcu_00254] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1"> <tr> <td><b>Name:</b></td> <td>Mcu_ModeType</td> </tr> <tr> <td><b>Type:</b></td> <td>uint</td> </tr> <tr> <td><b>Range:</b></td> <td>0 .. &lt;number of MCU modes&gt;-1</td> <td>-- The range is dependent on the number of MCU modes provided in the configuration structure. The type shall be chosen depending on MCU platform for best performance.</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">This type specifies the identification (ID) for a MCU mode, which is configured in the configuration structure.</td> </tr> </table>	<b>Name:</b>	Mcu_ModeType	<b>Type:</b>	uint	<b>Range:</b>	0 .. <number of MCU modes>-1	-- The range is dependent on the number of MCU modes provided in the configuration structure. The type shall be chosen depending on MCU platform for best performance.	<b>Description:</b>	This type specifies the identification (ID) for a MCU mode, which is configured in the configuration structure.	
<b>Name:</b>	Mcu_ModeType												
<b>Type:</b>	uint												
<b>Range:</b>	0 .. <number of MCU modes>-1	-- The range is dependent on the number of MCU modes provided in the configuration structure. The type shall be chosen depending on MCU platform for best performance.											
<b>Description:</b>	This type specifies the identification (ID) for a MCU mode, which is configured in the configuration structure.												

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																
12	Type Definitions	 MCAL-8404 - SWS_Mcu_00 255 : Mcu_RamSectionType <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00255] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;"><b>Name:</b></td><td colspan="3" style="padding: 2px;">Mcu_RamSectionType</td></tr> <tr> <td style="padding: 2px;"><b>Type:</b></td><td colspan="3" style="padding: 2px;">uint</td></tr> <tr> <td style="padding: 2px;"><b>Range:</b></td><td style="padding: 2px;">0 .. &lt; number of RAM sections &gt; - 1</td><td style="padding: 2px;">--</td><td style="padding: 2px;">The range is dependent on the number of RAM sections provided in the configuration structure. The type shall be chosen depending on MCU platform for best performance.</td></tr> <tr> <td style="padding: 2px;"><b>Description:</b></td><td colspan="3" style="padding: 2px;">This type specifies the identification (ID) for a RAM section, which is configured in the configuration structure.</td></tr> </table>	<b>Name:</b>	Mcu_RamSectionType			<b>Type:</b>	uint			<b>Range:</b>	0 .. < number of RAM sections > - 1	--	The range is dependent on the number of RAM sections provided in the configuration structure. The type shall be chosen depending on MCU platform for best performance.	<b>Description:</b>	This type specifies the identification (ID) for a RAM section, which is configured in the configuration structure.		
<b>Name:</b>	Mcu_RamSectionType																		
<b>Type:</b>	uint																		
<b>Range:</b>	0 .. < number of RAM sections > - 1	--	The range is dependent on the number of RAM sections provided in the configuration structure. The type shall be chosen depending on MCU platform for best performance.																
<b>Description:</b>	This type specifies the identification (ID) for a RAM section, which is configured in the configuration structure.																		

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>								
13	Type Definitions	 MCAL-8437 - SWS_Mcu_00 256 : RamStateType 	<p>[SWS_Mcu_00256] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1"> <tr> <td><b>Name:</b></td><td>Mcu_RamStateType</td></tr> <tr> <td><b>Type:</b></td><td>Enumeration</td></tr> <tr> <td><b>Range:</b></td><td>           MCU_RAMSTATE_INVALID -- Ram content is not valid or unknown (default).            MCU_RAMSTATE_VALID -- Ram content is valid:         </td></tr> <tr> <td><b>Description:</b></td><td>This is the Ram State data type returned by the function Mcu_GetRamState of the Mcu module. It is not required that all RAM state types are supported by the hardware.</td></tr> </table>	<b>Name:</b>	Mcu_RamStateType	<b>Type:</b>	Enumeration	<b>Range:</b>	MCU_RAMSTATE_INVALID -- Ram content is not valid or unknown (default). MCU_RAMSTATE_VALID -- Ram content is valid:	<b>Description:</b>	This is the Ram State data type returned by the function Mcu_GetRamState of the Mcu module. It is not required that all RAM state types are supported by the hardware.
<b>Name:</b>	Mcu_RamStateType										
<b>Type:</b>	Enumeration										
<b>Range:</b>	MCU_RAMSTATE_INVALID -- Ram content is not valid or unknown (default). MCU_RAMSTATE_VALID -- Ram content is valid:										
<b>Description:</b>	This is the Ram State data type returned by the function Mcu_GetRamState of the Mcu module. It is not required that all RAM state types are supported by the hardware.										

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																									
14	Function Definitions	 <b>MCAL-8532 - MCAL</b> Module : MCU Driver : Func Defs : Mcu_Init <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00153] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Mcu_Init</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Mcu_Init(                   const Mcu_ConfigType* ConfigPtr                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x00</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ConfigPtr</td> <td>Pointer to MCU driver configuration set.</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Return value:</b></td> <td colspan="2">None</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">This service initializes the MCU driver.</td> </tr> </table>	<b>Service name:</b>	Mcu_Init	<b>Syntax:</b>	void Mcu_Init( const Mcu_ConfigType* ConfigPtr )	<b>Service ID[hex]:</b>	0x00	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	ConfigPtr	Pointer to MCU driver configuration set.	<b>Parameters (inout):</b>	None		<b>Parameters (out):</b>	None		<b>Return value:</b>	None		<b>Description:</b>	This service initializes the MCU driver.	
<b>Service name:</b>	Mcu_Init																											
<b>Syntax:</b>	void Mcu_Init( const Mcu_ConfigType* ConfigPtr )																											
<b>Service ID[hex]:</b>	0x00																											
<b>Sync/Async:</b>	Synchronous																											
<b>Reentrancy:</b>	Non Reentrant																											
<b>Parameters (in):</b>	ConfigPtr	Pointer to MCU driver configuration set.																										
<b>Parameters (inout):</b>	None																											
<b>Parameters (out):</b>	None																											
<b>Return value:</b>	None																											
<b>Description:</b>	This service initializes the MCU driver.																											

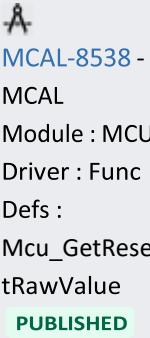
	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
15	Function Definitions	 <b>MCAL-8533 -</b> MCAL Module : MCU Driver : Func Defs : Mcu_InitRam Section <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00154] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Mcu_InitRamSection</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Std_ReturnType Mcu_InitRamSection(     Mcu_RamSectionType RamSection )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x01</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>RamSection     Selects RAM memory section provided in configuration set</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType E_OK: command has been accepted E_NOT_OK: command has not been accepted e.g. due to parameter error</td> </tr> <tr> <td><b>Description:</b></td> <td>This service initializes the RAM section wise.</td> </tr> </table>	<b>Service name:</b>	Mcu_InitRamSection	<b>Syntax:</b>	Std_ReturnType Mcu_InitRamSection( Mcu_RamSectionType RamSection )	<b>Service ID[hex]:</b>	0x01	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	RamSection     Selects RAM memory section provided in configuration set	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Std_ReturnType E_OK: command has been accepted E_NOT_OK: command has not been accepted e.g. due to parameter error	<b>Description:</b>	This service initializes the RAM section wise.
<b>Service name:</b>	Mcu_InitRamSection																						
<b>Syntax:</b>	Std_ReturnType Mcu_InitRamSection( Mcu_RamSectionType RamSection )																						
<b>Service ID[hex]:</b>	0x01																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	RamSection     Selects RAM memory section provided in configuration set																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	Std_ReturnType E_OK: command has been accepted E_NOT_OK: command has not been accepted e.g. due to parameter error																						
<b>Description:</b>	This service initializes the RAM section wise.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
16	Function Definitions	 <b>MCAL-8534 -</b> MCAL Module : MCU Driver : Func Defs : Mcu_InitClock <b>PUBLISHED</b>	<p>[SWS_Mcu_00155] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1"> <tr> <td><b>Service name:</b></td> <td>Mcu_InitClock</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Std_ReturnType Mcu_InitClock(                   Mcu_ClockType ClockSetting                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x02</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>ClockSetting Clock setting</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType E_OK: Command has been accepted E_NOT_OK: Command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td>This service initializes the PLL and other MCU specific clock options.</td> </tr> </table>	<b>Service name:</b>	Mcu_InitClock	<b>Syntax:</b>	Std_ReturnType Mcu_InitClock( Mcu_ClockType ClockSetting )	<b>Service ID[hex]:</b>	0x02	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	ClockSetting Clock setting	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Std_ReturnType E_OK: Command has been accepted E_NOT_OK: Command has not been accepted	<b>Description:</b>	This service initializes the PLL and other MCU specific clock options.
<b>Service name:</b>	Mcu_InitClock																						
<b>Syntax:</b>	Std_ReturnType Mcu_InitClock( Mcu_ClockType ClockSetting )																						
<b>Service ID[hex]:</b>	0x02																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	ClockSetting Clock setting																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	Std_ReturnType E_OK: Command has been accepted E_NOT_OK: Command has not been accepted																						
<b>Description:</b>	This service initializes the PLL and other MCU specific clock options.																						

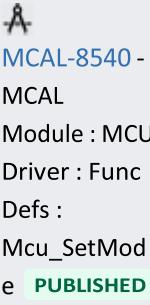
	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																							
17	Function Definitions	 <b>MCAL-8535 -</b> MCAL Module : MCU Driver : Func Defs : Mcu_Distribu tePllClock <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00156] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Mcu_DistributePllClock</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Std_ReturnType Mcu_DistributePllClock(     void )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x03</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Non Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Std_ReturnType</td> <td>E_OK: Command has been accepted E_NOT_OK: Command has not been accepted</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="3">This service activates the PLL clock to the MCU clock distribution.</td></tr> </table>	<b>Service name:</b>	Mcu_DistributePllClock	<b>Syntax:</b>	Std_ReturnType Mcu_DistributePllClock( void )	<b>Service ID[hex]:</b>	0x03	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Std_ReturnType	E_OK: Command has been accepted E_NOT_OK: Command has not been accepted	<b>Description:</b>	This service activates the PLL clock to the MCU clock distribution.		
<b>Service name:</b>	Mcu_DistributePllClock																									
<b>Syntax:</b>	Std_ReturnType Mcu_DistributePllClock( void )																									
<b>Service ID[hex]:</b>	0x03																									
<b>Sync/Async:</b>	Synchronous																									
<b>Reentrancy:</b>	Non Reentrant																									
<b>Parameters (in):</b>	None																									
<b>Parameters (inout):</b>	None																									
<b>Parameters (out):</b>	None																									
<b>Return value:</b>	Std_ReturnType	E_OK: Command has been accepted E_NOT_OK: Command has not been accepted																								
<b>Description:</b>	This service activates the PLL clock to the MCU clock distribution.																									

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																						
18	Function Definitions	 <b>MCAL-8536 -</b> MCAL Module : MCU Driver : Func Defs : Mcu_GetPllStatus <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00157] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1"> <tr> <td><b>Service name:</b></td> <td>Mcu_GetPllStatus</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Mcu_PllStatusType Mcu_GetPllStatus(                           void                           )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x04</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Mcu_PllStatusType</td> <td>PLL Status</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">This service provides the lock status of the PLL.</td> </tr> </table>	<b>Service name:</b>	Mcu_GetPllStatus	<b>Syntax:</b>	Mcu_PllStatusType Mcu_GetPllStatus( void )	<b>Service ID[hex]:</b>	0x04	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Mcu_PllStatusType	PLL Status	<b>Description:</b>	This service provides the lock status of the PLL.	
<b>Service name:</b>	Mcu_GetPllStatus																								
<b>Syntax:</b>	Mcu_PllStatusType Mcu_GetPllStatus( void )																								
<b>Service ID[hex]:</b>	0x04																								
<b>Sync/Async:</b>	Synchronous																								
<b>Reentrancy:</b>	Reentrant																								
<b>Parameters (in):</b>	None																								
<b>Parameters (inout):</b>	None																								
<b>Parameters (out):</b>	None																								
<b>Return value:</b>	Mcu_PllStatusType	PLL Status																							
<b>Description:</b>	This service provides the lock status of the PLL.																								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
19	Function Definitions	 <b>MCAL-8537 -</b> MCAL Module : MCU Driver : Func Defs : Mcu_GetResetReason <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00158] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Mcu_GetResetReason</td> </tr> <tr> <td><b>Syntax:</b></td> <td> <pre>Mcu_ResetType Mcu_GetResetReason(     void )</pre> </td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x05</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Mcu_ResetType</td> </tr> <tr> <td><b>Description:</b></td> <td>The service reads the reset type from the hardware, if supported.</td> </tr> </table>	<b>Service name:</b>	Mcu_GetResetReason	<b>Syntax:</b>	<pre>Mcu_ResetType Mcu_GetResetReason(     void )</pre>	<b>Service ID[hex]:</b>	0x05	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Mcu_ResetType	<b>Description:</b>	The service reads the reset type from the hardware, if supported.
<b>Service name:</b>	Mcu_GetResetReason																						
<b>Syntax:</b>	<pre>Mcu_ResetType Mcu_GetResetReason(     void )</pre>																						
<b>Service ID[hex]:</b>	0x05																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	Mcu_ResetType																						
<b>Description:</b>	The service reads the reset type from the hardware, if supported.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																						
20	Function Definitions	 <b>MCAL-8538 -</b> MCAL Module : MCU Driver : Func Defs : Mcu_GetResetRawValue <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00159] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Mcu_GetResetRawValue</td> </tr> <tr> <td><b>Syntax:</b></td> <td>Mcu_RawResetType Mcu_GetResetRawValue (                   void                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x06</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Mcu_RawResetType</td> <td>Reset raw value</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">The service reads the reset type from the hardware register, if supported.</td></tr> </table>	<b>Service name:</b>	Mcu_GetResetRawValue	<b>Syntax:</b>	Mcu_RawResetType Mcu_GetResetRawValue ( void )	<b>Service ID[hex]:</b>	0x06	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Mcu_RawResetType	Reset raw value	<b>Description:</b>	The service reads the reset type from the hardware register, if supported.	
<b>Service name:</b>	Mcu_GetResetRawValue																								
<b>Syntax:</b>	Mcu_RawResetType Mcu_GetResetRawValue ( void )																								
<b>Service ID[hex]:</b>	0x06																								
<b>Sync/Async:</b>	Synchronous																								
<b>Reentrancy:</b>	Reentrant																								
<b>Parameters (in):</b>	None																								
<b>Parameters (inout):</b>	None																								
<b>Parameters (out):</b>	None																								
<b>Return value:</b>	Mcu_RawResetType	Reset raw value																							
<b>Description:</b>	The service reads the reset type from the hardware register, if supported.																								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
21	Function Definitions	 <a href="#">MCAL-8539 -</a> MCAL Module : MCU Driver : Func Defs : Mcu_Perform Reset 	<p>[SWS_Mcu_00160] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Mcu_PerformReset</td></tr> <tr> <td><b>Syntax:</b></td><td>void Mcu_PerformReset(                         void                         )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x07</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Non Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>None</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>The service performs a microcontroller reset.</td></tr> </table>	<b>Service name:</b>	Mcu_PerformReset	<b>Syntax:</b>	void Mcu_PerformReset( void )	<b>Service ID[hex]:</b>	0x07	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Non Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	The service performs a microcontroller reset.
<b>Service name:</b>	Mcu_PerformReset																						
<b>Syntax:</b>	void Mcu_PerformReset( void )																						
<b>Service ID[hex]:</b>	0x07																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Non Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	The service performs a microcontroller reset.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
22	Function Definitions	 <b>MCAL-8540 - MCAL</b> Module : MCU Driver : Func Defs : Mcu_SetMode <span style="background-color: #e0ffe0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00161] shall be implemented and follow <a href="#">AUTOSAR BSW MCU Driver Specification</a>.</p> <table border="1"> <tr> <td><b>Service name:</b></td><td>Mcu_SetMode</td></tr> <tr> <td><b>Syntax:</b></td><td>void Mcu_SetMode(                   Mcu_ModeType McuMode                   )</td></tr> <tr> <td><b>Service ID[hex]:</b></td><td>0x08</td></tr> <tr> <td><b>Sync/Async:</b></td><td>Synchronous</td></tr> <tr> <td><b>Reentrancy:</b></td><td>Reentrant</td></tr> <tr> <td><b>Parameters (in):</b></td><td>McuMode Set different MCU power modes configured in the configuration set</td></tr> <tr> <td><b>Parameters (inout):</b></td><td>None</td></tr> <tr> <td><b>Parameters (out):</b></td><td>None</td></tr> <tr> <td><b>Return value:</b></td><td>None</td></tr> <tr> <td><b>Description:</b></td><td>This service activates the MCU power modes.</td></tr> </table>	<b>Service name:</b>	Mcu_SetMode	<b>Syntax:</b>	void Mcu_SetMode( Mcu_ModeType McuMode )	<b>Service ID[hex]:</b>	0x08	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	McuMode Set different MCU power modes configured in the configuration set	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	None	<b>Description:</b>	This service activates the MCU power modes.
<b>Service name:</b>	Mcu_SetMode																						
<b>Syntax:</b>	void Mcu_SetMode( Mcu_ModeType McuMode )																						
<b>Service ID[hex]:</b>	0x08																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	McuMode Set different MCU power modes configured in the configuration set																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	None																						
<b>Return value:</b>	None																						
<b>Description:</b>	This service activates the MCU power modes.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																				
23	Function Definitions	 <a href="#">MCAL-8541 - MCAL</a> Module : MCU Driver : Func Defs : Mcu_GetVersionInfo <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00162] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td><b>Service name:</b></td> <td>Mcu_GetVersionInfo</td> </tr> <tr> <td><b>Syntax:</b></td> <td>void Mcu_GetVersionInfo(                   Std_VersionInfoType* versioninfo                   )</td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x09</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>versioninfo   Pointer to where to store the version information of this module.</td> </tr> <tr> <td><b>Return value:</b></td> <td>None</td> </tr> <tr> <td><b>Description:</b></td> <td>This service returns the version information of this module.</td> </tr> </table>	<b>Service name:</b>	Mcu_GetVersionInfo	<b>Syntax:</b>	void Mcu_GetVersionInfo( Std_VersionInfoType* versioninfo )	<b>Service ID[hex]:</b>	0x09	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	versioninfo   Pointer to where to store the version information of this module.	<b>Return value:</b>	None	<b>Description:</b>	This service returns the version information of this module.
<b>Service name:</b>	Mcu_GetVersionInfo																						
<b>Syntax:</b>	void Mcu_GetVersionInfo( Std_VersionInfoType* versioninfo )																						
<b>Service ID[hex]:</b>	0x09																						
<b>Sync/Async:</b>	Synchronous																						
<b>Reentrancy:</b>	Reentrant																						
<b>Parameters (in):</b>	None																						
<b>Parameters (inout):</b>	None																						
<b>Parameters (out):</b>	versioninfo   Pointer to where to store the version information of this module.																						
<b>Return value:</b>	None																						
<b>Description:</b>	This service returns the version information of this module.																						

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>																						
24	Function Definitions	 <b>MCAL-8542 -</b> MCAL Module : MCU Driver : Func Defs : Mcu_GetRamState <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	<p>[SWS_Mcu_00207] shall be implemented and follow AUTOSAR BSW MCU Driver Specification.</p> <table border="1"> <tr> <td><b>Service name:</b></td> <td>Mcu_GetRamState</td> </tr> <tr> <td><b>Syntax:</b></td> <td><code>Mcu_RamStateType Mcu_GetRamState (</code>  <code>                    void</code>  <code>                    )</code></td> </tr> <tr> <td><b>Service ID[hex]:</b></td> <td>0x0a</td> </tr> <tr> <td><b>Sync/Async:</b></td> <td>Synchronous</td> </tr> <tr> <td><b>Reentrancy:</b></td> <td>Reentrant</td> </tr> <tr> <td><b>Parameters (in):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (inout):</b></td> <td>None</td> </tr> <tr> <td><b>Parameters (out):</b></td> <td>None</td> </tr> <tr> <td><b>Return value:</b></td> <td>Mcu_RamStateType</td> <td>Status of the Ram Content</td> </tr> <tr> <td><b>Description:</b></td> <td colspan="2">This service provides the actual status of the microcontroller Ram. (if supported)</td> </tr> </table>	<b>Service name:</b>	Mcu_GetRamState	<b>Syntax:</b>	<code>Mcu_RamStateType Mcu_GetRamState (</code> <code>                    void</code> <code>                    )</code>	<b>Service ID[hex]:</b>	0x0a	<b>Sync/Async:</b>	Synchronous	<b>Reentrancy:</b>	Reentrant	<b>Parameters (in):</b>	None	<b>Parameters (inout):</b>	None	<b>Parameters (out):</b>	None	<b>Return value:</b>	Mcu_RamStateType	Status of the Ram Content	<b>Description:</b>	This service provides the actual status of the microcontroller Ram. (if supported)	
<b>Service name:</b>	Mcu_GetRamState																								
<b>Syntax:</b>	<code>Mcu_RamStateType Mcu_GetRamState (</code> <code>                    void</code> <code>                    )</code>																								
<b>Service ID[hex]:</b>	0x0a																								
<b>Sync/Async:</b>	Synchronous																								
<b>Reentrancy:</b>	Reentrant																								
<b>Parameters (in):</b>	None																								
<b>Parameters (inout):</b>	None																								
<b>Parameters (out):</b>	None																								
<b>Return value:</b>	Mcu_RamStateType	Status of the Ram Content																							
<b>Description:</b>	This service provides the actual status of the microcontroller Ram. (if supported)																								

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
25	Configurable Interfaces	 <b>MCAL-8543 - MCAL</b> Module : Mcu Driver : Configurable Interfaces <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	Mcu module shall use the Fee Module Stubs to target notification functions.
26	Sequence Diagrams	 <b>MCAL-8544 - MCAL</b> Module : MCU Driver : Sequence Diagram <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	MCU module shall follow the sequence diagrams as described in section 9 of <a href="#">AUTOSAR BSW MCU Driver Specification</a> .

	<b>Specification Category</b>	<b>Architecture Identifier</b>	<b>Description</b>
27	Configuration Specification	 <b>MCAL-8545 -</b> MCAL Module : MCU Driver : Config : Mcu : Configuration of Mcu <b>PUBLISHED</b>	ECUC_Mcu_00189 - Mcu Module shall support VARIANT-POSTBUILD or VARIANT-PRE-COMPIL

## 8.13 ETH Driver

ETH Driver Requirements will be added in future release.

## 8.14 BSW Stubs

All MCAL modules shall use BSW Stubs in demo/example application in lieu of the AUTOSAR Stack BSW. The details of BSW Stubs per module shall be included in the module specific design guide.

	<b>BSW Stub</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	CanIf	 <a href="#">MCAL-7699 - MCAL</a> BSWStubs: CanIf <span style="background-color: #e0f2e0; border: 1px solid #d0e0d0; padding: 2px;">PUBLISHED</span>	MCAL shall provide a stub like implementation of Can Interface module.
2	Dem	 <a href="#">MCAL-7700 - MCAL</a> BSWStubs: Dem <span style="background-color: #e0f2e0; border: 1px solid #d0e0d0; padding: 2px;">PUBLISHED</span>	MCAL shall provide a stub like implementation of Diagnostic Event Manager.
3	Det	 <a href="#">MCAL-7701 - MCAL</a> BSWStubs: Det <span style="background-color: #e0f2e0; border: 1px solid #d0e0d0; padding: 2px;">PUBLISHED</span>	MCAL shall provide a stub like implementation of Development Error Tracer.
4	EcuM	 <a href="#">MCAL-7702 - MCAL</a> BSWStubs: EcuM <span style="background-color: #e0f2e0; border: 1px solid #d0e0d0; padding: 2px;">PUBLISHED</span>	MCAL shall provide a stub like implementation of ECU Manager module.
5	Fee	 <a href="#">MCAL-7703 - MCAL</a> BSWStubs: Fee <span style="background-color: #e0f2e0; border: 1px solid #d0e0d0; padding: 2px;">PUBLISHED</span>	MCAL shall provide a stub like implementation of Fee (Flash EEPROM Emulation) module.

	<b>BSW Stub</b>	<b>Architecture Identifier</b>	<b>Description</b>
6	MemIf	 <a href="#">MCAL-7704 - MCAL</a> BSWStubs: MemIf <span style="background-color: #90EE90; color: black; padding: 2px;">PUBLISHED</span>	MCAL shall provide a stub like implementation of MemIf module.
7	Os	 <a href="#">MCAL-7705 - MCAL</a> BSWStubs: Os <span style="background-color: #90EE90; color: black; padding: 2px;">PUBLISHED</span>	MCAL shall provide a stub like implementation of Os module.
8	Rte	 <a href="#">MCAL-7706 - MCAL</a> BSWStubs: Rte <span style="background-color: #90EE90; color: black; padding: 2px;">PUBLISHED</span>	MCAL shall provide a stub like implementation of AUTOSAR Rte.
9	Tm	 <a href="#">MCAL-7707 - MCAL</a> BSWStubs: Tm <span style="background-color: #90EE90; color: black; padding: 2px;">PUBLISHED</span>	MCAL shall provide a stub like implementation of AUTOSAR Tm Module (Time).
10	WdgIf	 <a href="#">MCAL-7708 - MCAL</a> BSWStubs: WdgIf <span style="background-color: #90EE90; color: black; padding: 2px;">PUBLISHED</span>	MCAL shall provide a stub like implementation of AUTOSAR WdgIf (WDG Interface) Module.

## 9 Performance Objectives

### 9.1 Resource Consumption Objectives

The MCAL module design shall estimate the Code and Data segment sizes in bytes and keep module sizes less than or equal to below numbers.

**Code Size Per Module**

	<b>SW Module</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	DIO	 <a href="#">MCAL-5437</a> - DIO Code Size <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	Implementation of the DIO driver shall not exceed 5 Kilo bytes of code.
2	PWM	 <a href="#">MCAL-6794</a> - PWM Code Size <span style="background-color: #e0f2e0; padding: 2px;">PUBLISHED</span>	Implementation of the PWM driver shall not exceed 40 Kilo bytes of code size.

	<b>SW Module</b>	<b>Architecture Identifier</b>	<b>Description</b>
3	EPWM	 <a href="#">MCAL-7556 - PWM Code Size</a> <b>PUBLISHED</b>	Implementation of the PWM driver shall not exceed 40 Kilo bytes of code size.
4	ADC	 <a href="#">MCAL-5823 - ADC Code Size</a> <b>PUBLISHED</b>	Implementation of the ADC driver shall not exceed 30 Kilo bytes of code size.
5	CAN	 <a href="#">MCAL-5920 - CAN Code Size</a> <b>PUBLISHED</b>	Implementation of the CAN driver shall not exceed 30 Kilo bytes of code size
6	SPI	 <a href="#">MCAL-6489 - SPI Code Size</a> <b>PUBLISHED</b>	Implementation of the SPI driver shall not exceed 30 Kilo bytes of code size.

	<b>SW Module</b>	<b>Architecture Identifier</b>	<b>Description</b>
7	GPT	 <a href="#">MCAL-6301</a> - Code Size <b>PUBLISHED</b>	The implementation of the GPT driver (excluding the configurator) shall not occupy more than 5 KB of text section and 1 KB of data section
8	WDG	 <a href="#">MCAL-5568</a> - WDG : Code size <b>PUBLISHED</b>	WDG Implementation shall not consume more than 30 Kilobytes for .text section and 2Kilobytes for .const section.
9	CDD IPC	 <a href="#">MCAL-7156</a> - IPC Code Size <b>PUBLISHED</b>	The implementation of the CDD IPC driver (excluding the configurator) shall not occupy more than 5 KB of text section
10	FLS	 <a href="#">MCAL-7674</a> - MCAL Module : Flash Driver : Code Size <b>PUBLISHED</b>	Implementation of the PWM driver shall not exceed 50 Kilo bytes of code size.

	<b>SW Module</b>	<b>Architecture Identifier</b>	<b>Description</b>
11	ICU	 <a href="#">MCAL-7651</a> - MCAL Module: ICU: Code Size <b>PUBLISHED</b>	Implementation of the PWM driver shall not exceed 40 Kilo bytes of code size.
12	MCU	 <a href="#">MCAL-8546</a> - MCAL Module: MCU: Code Size <b>PUBLISHED</b>	Implementation of the MCU driver shall not exceed 3 Kilo bytes of code size.

**Stack Size Per Module:**

	<b>SW Module</b>	<b>Architecture Identifier</b>	<b>Description</b>
1	DIO	 <a href="#">MCAL-5431</a> - DIO Stack Size <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	Worst case stack utilization of DIO driver shall not exceed 2 kilo bytes.
2	PWM	 <a href="#">MCAL-6862</a> - PWM Stack Size <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	Worst case stack utilization of PWM driver shall not exceed 2 kilo bytes
3	EPWM	 <a href="#">MCAL-7514</a> - PWM Stack Size <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	Worst case stack utilization of PWM driver shall not exceed 2 kilo bytes
4	ADC	 <a href="#">MCAL-5879</a> - ADC Stack Size <span style="background-color: #00AEEF; color: white; padding: 2px 5px;">PUBLISHED</span>	Worst case stack utilization of ADC driver shall not exceed 4 kilo bytes

	<b>SW Module</b>	<b>Architecture Identifier</b>	<b>Description</b>
5	CAN	 <a href="#">MCAL-5974 - CAN Stack Size</a> <span>PUBLISHED</span>	Worst case stack utilization of CAN driver shall not exceed 4 kilo bytes.
6	SPI	 <a href="#">MCAL-6636 - SPI Stack Size</a> <span>PUBLISHED</span>	Worst case stack utilization of SPI driver shall not exceed 4 kilo bytes.
7	GPT	 <a href="#">MCAL-6228 - Stack Size</a> <span>PUBLISHED</span>	The Implementation of service API's shall not consume more than 2 kilobytes of stack space
8	WDG	 <a href="#">MCAL-5564 - WDG : Stack Size</a> <span>PUBLISHED</span>	WDG Service API's shall not consume more than 2 kilo-bytes of stack memory

	<b>SW Module</b>	<b>Architecture Identifier</b>	<b>Description</b>
9	CDD IPC	 <a href="#">MCAL-7239</a> - Stack : Size <b>PUBLISHED</b>	The Implementation of service API's shall not consume more than 2 kilobytes of stack space.
10	FLS	 <a href="#">MCAL-7675</a> - MCAL Module : Flash Driver : Stack Size <b>PUBLISHED</b>	Worst case stack utilization of PWM driver shall not exceed 4 kilo bytes
11	ICU	 <a href="#">MCAL-7652</a> - MCAL Module: ICU: Stack Size <b>PUBLISHED</b>	Worst case stack utilization of PWM driver shall not exceed 2 kilo bytes

	<b>SW Module</b>	<b>Architecture Identifier</b>	<b>Description</b>
12	MCU	 <a href="#">MCAL-8547</a> - MCAL Module: MCU: Stack Size <b>PUBLISHED</b>	Worst case stack utilization of MCU driver shall not exceed 3 kilo bytes



## 9.2 Critical timing and Performance

None

## 10 Decision Analysis & Resolution (DAR)

No.	Decision Criteria	Alternatives	Selected	Rationale alternative	
1	Decision on 1. CSL code consists of prepared for safety Chip Select and definitions. MCAL recommended that into MCAL space.	Option 2. API's directly from dependencies library calls and become selfheaders files. The MCAL drivers shall use PDK CSL for register access.	To become best whether to port MCAL removes any PDK through CSL contained.	register level access certification, it is Library(CSL) API will make use of CSL external	
<b>Trade-offs</b>					

CSL code that is ported into MCAL will need to be manually maintained along with driver source code.



No.	Decision Criteria	Alternatives	Selected alternative	Rationale	Trade-offs
		2. CSL code will be brought into the MCAL code base. Each driver will bring in the IP specific CSL code as part of its driver source code. Shared header files (such as header files) will go in a common MCAL include directory.			



No.	Decision Criteria	Alternatives	Selected alternative	Rationale	Trade-offs
2	Decision on how to package MCAL and deliver to customers.	<ol style="list-style-type: none"><li>MCAL shall be delivered as a stand alone package, and not have dependency on SDK releases.</li><li>MCAL shall be delivered along with SDK release packages. MCAL examples have dependency on SDK components and MCAL drivers have dependency on compiler provided as part of SDK release.</li></ol>	Option 1.	<p>MCAL will be delivered as a standalone package as the core MCAL drivers do not have dependency on SDK.</p> <p>This is required as MCAL will not support multiple families of SoCs and needs to be integrated into multiple SDKs.</p>	Customer who require to use MCAL example applications will need to download the required SDK separately.



## 11 Testing Guidelines

Testing will be performed for Unit (Design), Integration (Architecture) and Qualification (Requirements). Unit and Integration tests will consider positive, negative, and boundary testing.

Static and Dynamic analysis will also be employed. Testing will be performed on EVM. Tests will cover bare-metal tests. For more information on the test strategy, please see the Test Strategy document.

MCAL tests should adhere to guidelines to enable test automation to the extent possible.

Examples for showing the usage of the safety diagnostics in the application context will also be provided to ease customer integration and understanding of the diagnostics.



## 12 Template Revision History

Author Name	Description	Version	Date
Yaniv Machani	Initial version	0.1	03 Oct 2018
Yaniv Machani	Updated to include EP views	0.4	02 Nov 2018
Yaniv Weizman	Restructuring and editing to further meet the A-SPICE and EP requirements	0.5	27 Dec 2018
Yaniv Weizman	Adding link to Architecture review template	0.6	22 Oct 2019



Author Name	Description	Version	Date
Yaniv Weizman	Adding requirement type column for requirements table (Functional/Non-Functional). Adding DAR table	0.65	13 Nov 2019
Yaniv Weizman	Adding tables for Testing guidelines	0.7	18 Nov 2019
Krishna	Updated the template to meet ASPICE requirements	1.0	11 Aug 2020