MCAL GPT Module Software Design Document

Document Version : 48
Document Owner : Texas Instruments
Document Status : Published
Last Approval Date : Mar 11, 2022

1 Revision History

Version	Date	Author	Document Status	Comments
0.1	25 Jul 2018	Sujith S	DONE	First Version
0.2	25 Jul 2018	Sujith S	DONE	Format conversion and review completed
0.3	25 Jul 2018	Sujith S	DONE	Updated section "Development Errors" to include requirement mapping
0.4	19 Jan 2020	Sunil M S	DONE	Updates w.r.o porting AUTOSAR 4.3.1 Version
0.5	06 Oct 2021	Nishit Dhas	DONE	Added Safety Diagnostic Features and changed design document format as per ASPICE
0.6	24 Jan 2022	Nikki S	IN PROGRESS	JACINTOREQ-1870
v48	04 Mar 2022	Nikki S	DONE	Review Comments Addressed

2 Terms and Abbreviations

Abbreviation / Term	Meaning / Explanation
GPT	General Purpose Timer
AUTOSAR	AUTomotive Open System ARchitecture
RTE	Runtime Environment
BSW	Basic Software
MCAL	MicroController Abstraction Layer
SBL	Serial Bootloader
API	Application Programming Interface
DET	Default Error Tracer
DEM	Diagnostic Event Manager – module to handle diagnostic relevant events.
ECU	Electronic Control Unit

Abbreviation / Term	Meaning / Explanation
MCU	Micro Controller Unit
OS	Operating System
SoC	System on a Chip
DAR	Decision Analysis and Resolution

3 Introduction

3.1 Overview

The figure below depicts the AUTOSAR layered architecture as 3 distinct layers, Application, Runtime Environment (RTE) and Basic Software (BSW). The BSW is further divided into 4 layers, Services, Electronic Control Unit Abstraction, MicroController Abstraction (MCAL) and Complex Drivers.

AUTOSAR Architecture

MCAL is the lowest abstraction layer of the Basic Software. It contains software modules that interact with the Microcontroller and its internal peripherals directly. Gpt driver is part of the Microcontroller Drivers (block, show above). Below shows the position of the Gpt driver in the AUTOSAR Architecture.

3.2 Purpose and Scope

The Detailed Design document provides the design details of GPT driver and aims to provide a guide to a design that could be implemented by a software developer.

The scope of this document is to describe the software design procedure of GPT module.

3.3 Module Overview

GPT primarily used to generate different time bases that other modules of AUTOSAR could depend on. It uses hardware IP "dmtimer_dmc1ms_10_rel.1.0.x". Below listed are some of the key features provided. Refer to SoC User Manual for specific details.

- 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 (2n, 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. Refer section (References) for more details on timer operation.

3.4 Requirements

The Gpt driver shall implement as per requirements detailed in Reference 1 - AUTOSAR 4.3.1

3.4.1 Features Supported

Below listed are some of the key features that are expected to be supported

- Starting and stopping of hardware timers
- Getting the timer values
- Setting one shot mode or continuous mode
- Controlling time triggered interrupt notifications
- Controlling time triggered wakeup interrupts
- Supports additional configuration parameters, refer section (Implementation specific parameters (computed)) & (Gpt_RegisterReadback)

Design Identifier	**Description**
MCAL-6324 - SWS_Gpt_00380 : Published info in header **PUBLISHED**	SWS_Gpt_00380 : Published info in header

3.4.2 Features Not Supported / NON Compliance

- **[NON Compliance]** Gpt PreDef Timers is not supported
- **[NON Compliance]** GptClockReference doesn’t refer to McuClockReferencePoint. Refer section (Constraints) for details.
- Standard AUTOSAR GPT specification Reference 1 - AUTOSAR 4.3.1, categorizes few BSW General Requirements as non-requirements.

3.5 Assumptions

Below listed are assumed to valid for this design/implementation, exceptions and other deviations are listed for each explicitly. Care should be taken to ensure these assumptions are addressed.

1. The functional clock to the GPT module is expected to be on before calling any GPT module API.
2. The GPT driver as such doesn't perform any PRCM programming to get the functional clock.
3. **The clock-source selection for GPT is not performed by the GPT driver, other entities such as SBL, Sciclient shall perform the same.**
4. **The GTC hardware present in the SOC shall not be supported as GPT module.**

Note that assumption 1 & 2 are specified by AUTOSAR GPT specification and 3 & 4 are device specific assumption.

3.6 Constraints

Design Identifier	Description
MCAL-6225 - ECUC_Gpt_00330 : GptClockReference PUBLISHED	ECUC_Gpt_00330 : GptClockReference
MCAL-6229 - ECUC_Gpt_00333 : GptChannelClkSrcRef PUBLISHED	ECUC_Gpt_00333 : GptChannelClkSrcRef
MCAL-6283 - ECUC_Gpt_00329 : GptClockReferencePoint : Container PUBLISHED	ECUC_Gpt_00329 : GptClockReferencePoint : Container

Some of the critical constraints of this design are listed below

- Is cases where MCU module is not employed (supported) to configure the clock source for GPT module (refer Assumptions sub-item 3), the SBL/GEL selects the clock sources.

3.7 Hardware and SW platforms

Hardware Platforms

- Refer to specified SoC User Manual to check if ADC module is supported.

Software Platforms

- Bare-Metal

3.8 Dependencies

In addition to dependencies listed in section 5 of Reference 1 - AUTOSAR 4.3.1 , GPT driver shall depend on these modules to realize the required functionality. GPT uses Timer hardware present in the device to realize the functionality, this peripheral requires 2 different clock to be operational, namely ICLK and FCLK.

3.8.1 SBL

- **ICLK**: Is interface clock required for internal operation of the peripheral. This is not expected to change and typically programmed by SBL, please refer the device specific manual for details and valid value.
- **FCLK**: Is functional clock, used to drive the counter of the timer module. SBL/GEL selects the right clock source for the peripheral.

3.9 Stakeholders

- Developers
- Test Engineers
- Customer Integrator

3.10 References

	Specification	Comment/Link
1	AUTOSAR 4.3.1	AUTOSAR Specification for GPT Driver.
2	BSW General Requirements / Coding guidelines	Autosar and Coding guidelines for the Mcal drivers.
3	Software Product Specification (SPS)	Product Functional requirements.
4	Software Architecture	Mcal Software Architecture.

4 Design Description

4.1 Fundamental Operation

As detailed in the TRM, the timer module generates an interrupt when the counter reaches its maximum value (i.e. 0xFFFFFFFF, for a 32 bit counter). The basic idea is use “Auto Reload” mode of the timer and initial count that could be set (TCRR register). Consider an example where timer is configured to expire after reaching a count of 0x0E000000

Timer Not Yet Started

Following sequence of steps shall be performed, before the timer could be started

- The initial count of the counter is set to 0xF1FFFFFF (i.e. 0xFFFFFFFF - 0x0E000000)
- The reload register (TLDR) is set with 0xF1FFFFFF as depicted above

Timer Started

- Timer is programmed as configured (one shot or continuous mode)
- The timer is started and the counter (register TCRR), starts counting on every pulse
    - As depicted in above figure, TCRR has moved w.r.t to TLDR

Timer Expired

- When the timer expires, the TCRR is loaded with value present in TLDR as show above
- An interrupt can be triggered at this point
- The timer would default to as show in figure "Timer not yet started", in continuous mode. Also note that no explicit start would be required
- In One Shot mode, timer is halted. i.e. TCRR stop counting the count of 0xFFFFFFFF is retained by TCRR

4.2 Modes of Timer

Refer Reference 1 - AUTOSAR 4.3.1 specifically section 7.1 of the specification for more details

4.2.1 Continuous Mode

Mode: Continuous: Sourced from AUTOSAR GPT Driver spec

Continuous Mode: Expected time value Sourced from AUTOSAR GPT Driver spec

Design Identifier	Description
MCAL-6194 - SWS_Gpt_00330 : Mode : C : Free Running PUBLISHED	SWS_Gpt_00330 : Mode : C : Free Running
MCAL-6346 - SWS_Gpt_00186 : Mode : Continuous PUBLISHED	SWS_Gpt_00186 : Mode : Continuous
MCAL-6231 - SWS_Gpt_00329 : Timer Start value PUBLISHED	SWS_Gpt_00329 : Timer Start value
MCAL-6262 - SWS_Gpt_00337 : State : Debug Info PUBLISHED	SWS_Gpt_00337 : State : Debug Info

4.2.2 **One Shot Mode**

Mode: One Shot: Sourced from AUTOSAR GPT Driver spec

Design Identifier	Description
MCAL-6303 - SWS_Gpt_00185 : Mode : One Shot PUBLISHED	SWS_Gpt_00185 : Mode : One Shot
MCAL-6231 - SWS_Gpt_00329 : Timer Start value PUBLISHED	SWS_Gpt_00329 : Timer Start value

4.3 Determination of Time Elapsed

The elapsed time could be computed under following conditions, also Refer Reference 1 - AUTOSAR 4.3.1 specifically “Table 5: Summary: Return values and DET errors of Gpt_GetTimeElapsed”

- In continuous mode
    - Can be obtained by subtracting Timer Reload Register value (TLDR) from current counter value (TCRR)
- In One Shot mode
    - Timer is counting
        - Same as “In Continuous mode”
    - Timer expired
        - Can be obtained by subtracting Timer Reload Register value from max value (i.e. 0xFFFFFFFF for 32 bit counter)

4.4 Determination of time remaining

The elapsed time could be computed under following conditions, also Refer Reference 1 - AUTOSAR 4.3.1 section 1.5 specifically “Table 6: Summary: Return values and DET errors of Gpt_GetTimeRemaining”

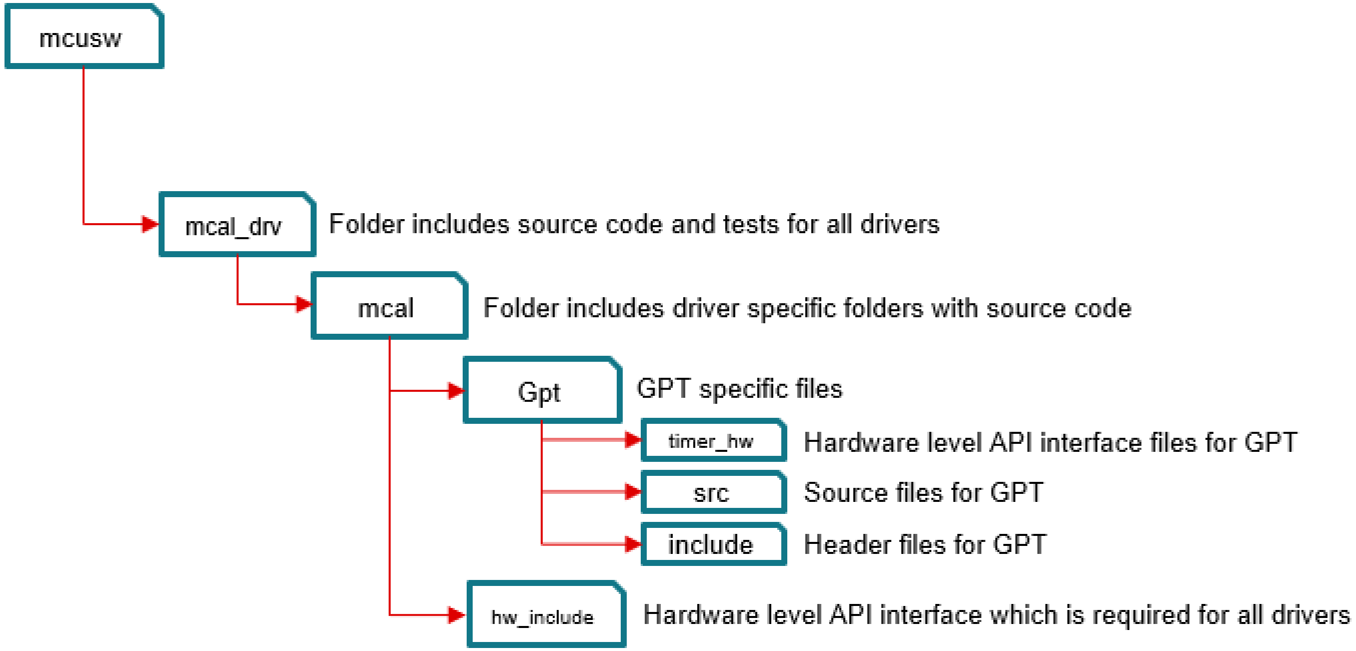
- In continuous mode
    - Can be obtained by subtracting current counter value (TCRR) from max value (i.e. 0xFFFFFFFF for 32 bit counter)
- In One Shot mode
    - Timer is counting
        - Same as “In Continuous mode”
    - Timer expired
        - Value is always 0

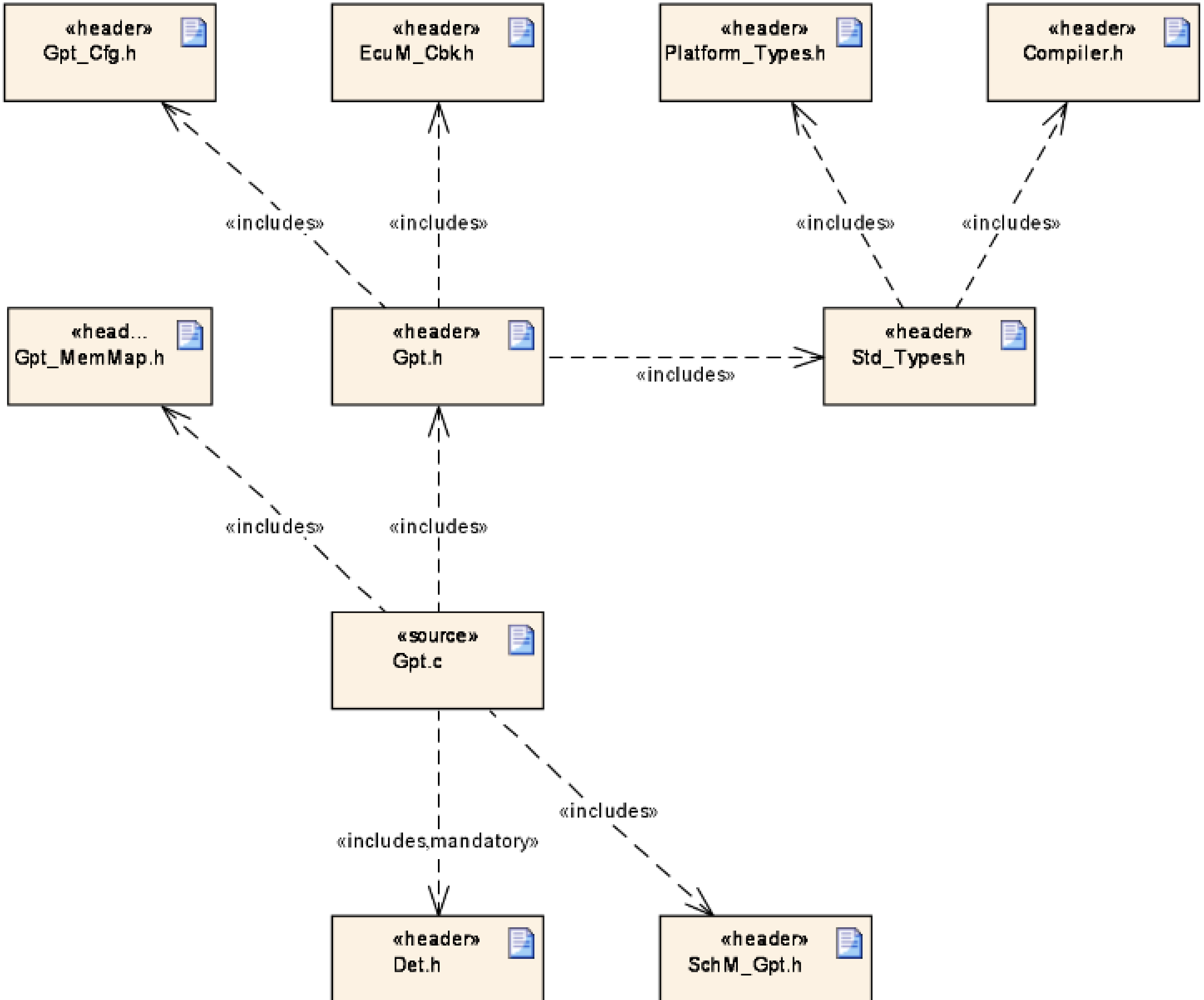
4.5 Directory Structure

The directory structure is as depicted in figures below, the source files can be categorized under “Driver Implementation” and “Example Application”

Driver Implemented by

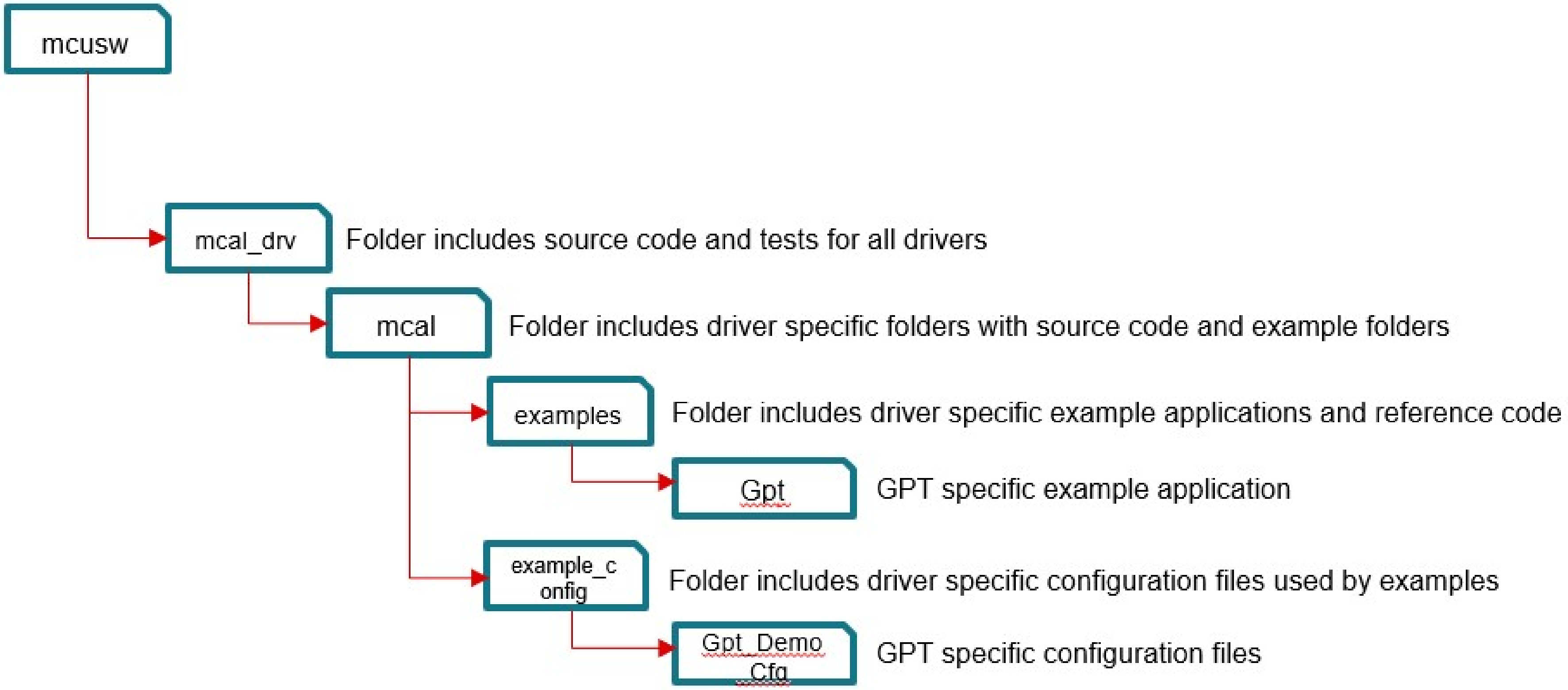
- Gpt.h and Gpt_Irq.h: Shall implement the interface provided by the driver
- Gpt.c, Gpt_Gptimer.c, Gpt_Irq.c and Gpt_Priv.h: Shall implement the driver functionality


mcusw
mcal_drv
Folder includes source code and tests for all drivers
mcal
Folder includes driver specific folders with source code
Gpt
GPT specific files
timer_hw
Hardware level API interface files for GPT
src
Source files for GPT
include
Header files for GPT
hw_include
Hardware level API interface which is required for all drivers


«header»
Gpt_Cfg.h
«header»
EcuM_Cbk.h
«header»
Platform_Types.h
«header»
Compiler.h
«includes»
«includes»
«includes»
«includes»
«head...
Gpt_MemMap.h
«header»
Gpt.h
«header»
Std_Types.h
«includes»
«includes»
«includes»
«source»
Gpt.c
«includes»
«includes,mandatory»
«header»
Det.h
«header»
SchM_Gpt.h

Example Application

- Gpt_Cfg.h and Gpt_Cfg.c: Shall implement the generated configuration for pre-compile variant
- Gpt_PBcfg.c: Shall implement the generated configuration for post-build variant
- GptApp.c and GptApp.h: Shall implement the example application that demonstrates the use of the driver



mcusw

mcal_drv Folder includes source code and tests for all drivers

mcal Folder includes driver specific folders with source code and example folders

examples Folder includes driver specific example applications and reference code

Gpt GPT specific example application

example_config Folder includes driver specific configuration files used by examples

Gpt_Demo_Cfg GPT specific configuration files

Design Identifier	Description
MCAL-6299 - SWS_Gpt_00375 : Development Error : Include file PUBLISHED	SWS_Gpt_00375 : Development Error : Include file
MCAL-6265 - SWS_Gpt_00261 : IRQ : Include file PUBLISHED	SWS_Gpt_00261 : IRQ : Include file
MCAL-6356 - SWS_Gpt_00293 : Include file PUBLISHED	SWS_Gpt_00293 : Include file

4.6 Configurator

The AUTOSAR GPT Driver Specification details mandatory parameters that shall be configurable via the configurator. Please refer section 10 of Reference 1 - AUTOSAR 4.3.1.

Design Identifier	Description
MCAL-6352 - ECUC_Gpt_00183 : GptDriverConfiguration : Container PUBLISHED	ECUC_Gpt_00183 : GptDriverConfiguration : Container

Design Identifier	Description
MCAL-6352 - ECUC_Gpt_00183 : GptDriverConfiguration : Container PUBLISHED	ECUC_Gpt_00183 : GptDriverConfiguration : Container
MCAL-6314 - ECUC_Gpt_00321 : GptDevErrorDetect PUBLISHED	ECUC_Gpt_00321 : GptDevErrorDetect
MCAL-6336 - ECUC_Gpt_00322 : GptReportWakeupSource PUBLISHED	ECUC_Gpt_00322 : GptReportWakeupSource
MCAL-6283 - ECUC_Gpt_00329 : GptClockReferencePoint : Container PUBLISHED	ECUC_Gpt_00329 : GptClockReferencePoint : Container
MCAL-6225 - ECUC_Gpt_00330 : GptClockReference PUBLISHED	ECUC_Gpt_00330 : GptClockReference
MCAL-6214 - ECUC_Gpt_00269 : GptChannelConfigSet PUBLISHED	ECUC_Gpt_00269 : GptChannelConfigSet
MCAL-6328 - ECUC_Gpt_00309 : GptChannelMode PUBLISHED	ECUC_Gpt_00309 : GptChannelMode

Design Identifier	Description
MCAL-6198 - ECUC_Gpt_00331 : GptChannelTickFrequency PUBLISHED	ECUC_Gpt_00331 : GptChannelTickFrequency
MCAL-6270 - ECUC_Gpt_00332 : GptChannelTickValueMax PUBLISHED	ECUC_Gpt_00332 : GptChannelTickValueMax
MCAL-6245 - ECUC_Gpt_00311 : GptEnableWakeup PUBLISHED	ECUC_Gpt_00311 : GptEnableWakeup
MCAL-6305 - ECUC_Gpt_00312 : GptNotification PUBLISHED	ECUC_Gpt_00312 : GptNotification
MCAL-6229 - ECUC_Gpt_00333 : GptChannelClkSrcRef PUBLISHED	ECUC_Gpt_00333 : GptChannelClkSrcRef
MCAL-6322 - ECUC_Gpt_00235 : GptWakeupConfiguration PUBLISHED	ECUC_Gpt_00235 : GptWakeupConfiguration
MCAL-6317 - ECUC_Gpt_00313 : GptWakeupSourceRef PUBLISHED	ECUC_Gpt_00313 : GptWakeupSourceRef

Design Identifier	Description
MCAL-6278 - ECUC_Gpt_00193 : GptConfigurationOfOptApiServices : Container PUBLISHED	ECUC_Gpt_00193 : GptConfigurationOfOptApiServices : Container
MCAL-6213 - ECUC_Gpt_00314 : GptDeinitApi PUBLISHED	ECUC_Gpt_00314 : GptDeinitApi
MCAL-6244 - ECUC_Gpt_00315 : GptEnableDisableNotificationApi PUBLISHED	ECUC_Gpt_00315 : GptEnableDisableNotificationApi
MCAL-6361 - ECUC_Gpt_00317 : GptTimeElapsedApi PUBLISHED	ECUC_Gpt_00317 : GptTimeElapsedApi
MCAL-6289 - ECUC_Gpt_00318 : GptTimeRemainingApi PUBLISHED	ECUC_Gpt_00318 : GptTimeRemainingApi
MCAL-6298 - ECUC_Gpt_00319 : GptVersionInfoApi PUBLISHED	ECUC_Gpt_00319 : GptVersionInfoApi
MCAL-6242 - ECUC_Gpt_00320 : GptWakeupFunctionalityApi PUBLISHED	ECUC_Gpt_00320 : GptWakeupFunctionalityApi

Design Identifier	Description
MCAL-6331 - ECUC_Gpt_00308 : GptChannelId PUBLISHED	ECUC_Gpt_00308 : GptChannelId

4.6.1 NON Standard configurable parameters

Following lists this design’s specific configurable parameters

Parameter	Usage Comment
GptDefaultOSCounterId	This shall allow integrators to specify the OS counter instance to be used in OS API GetCounterValue () The driver shall implement timed-wait for all waits (e.g. waiting for reset to complete). This timed wait shall use OS API GetCounterValue ()
GptDeviceVariant	This shall allow integrators to select the device variant for which integration is being performed. This parameter shall be used by driver to impose device specific constraints. The user guide shall detail the device specific constraints
GptChannelPrescale	This parameter per channel and as part of container “GptChannelConfigSet” will allow user to scale the FCLK

4.6.2 Implementation specific parameters (computed)

The configurator shall determine the maximum number of channels that are configured and generate a macro to define the same. This shall be used to perform range checks on channel configurations and channel ID provided at driver initialization time. Refer section (MACROS, Data Types & Structures)

4.6.3 Variant Support

The driver shall support both VARIANT-POST-BUILD & VARIANT-PRE-COMPILE

Design Identifier	Description
MCAL-6352 - ECUC_Gpt_00183 : GptDriverConfiguration : Container PUBLISHED	ECUC_Gpt_00183 : GptDriverConfiguration : Container
MCAL-6222 - SWS_Gpt_00342 : Support PB or PC PUBLISHED	SWS_Gpt_00342 : Support PB or PC

4.7 Error Classification

Errors are classified in two categories, development error and runtime / production error.

4.7.1 Development Errors

Type of Error	Related Error code	Value (Hex)
API service called without module initialization	GPT_E_UINIT	0x0A
API service for initialization is called when already initialized	GPT_E_ALREADY_INITIALIZED	0x0D
API error return code: Init function failed	GPT_E_INIT_FAILED	0x0E
API parameter checking: invalid channel	GPT_E_PARAM_CHANNEL	0x14
API parameter checking: invalid value	GPT_E_PARAM_VALUE	0x15
API parameter checking: invalid pointer	GPT_E_PARAM_POINTER	0x16
API parameter checking: invalid mode	GPT_E_PARAM_MODE	0x1F

4.7.2 Error Detection

The detection of development errors is configurable (ON / OFF) at pre-compile time. The switch GptDevErrorDetect will activate or deactivate the detection of all development errors.

Design Identifier	Description
MCAL-6304 - MCAL Module : Gpt enhancements PUBLISHED	MCAL Module : Gpt enhancements

4.7.3 Error notification (DET)

All detected development errors are reported to Det_ReportError service of the Development Error Tracer (DET).

4.7.4 Runtime Errors

The following runtime/production errors shall be detectable by Gpt driver.

Type of Error	Related Error code	Value (Hex)
API service is called when timer channel is still busy	GPT_E_BUSY	0x0B

4.7.5 Error notification (DEM)

All detected run time errors shall be reported to Det_ReportRuntimeError () service.

Design Identifier	Description
MCAL-6312 - SWS_Gpt_00325 : CheckWakeup : Error : If uninitialized PUBLISHED	SWS_Gpt_00325 : CheckWakeup : Error : If uninitialized
MCAL-6367 - SWS_Gpt_00230 : EnWakeup : Error : Uninitialized PUBLISHED	SWS_Gpt_00230 : EnWakeup : Error : Uninitialized
MCAL-6340 - SWS_Gpt_00229 : DisWakeup : Error : Un Initialized PUBLISHED	SWS_Gpt_00229 : DisWakeup : Error : Un Initialized
MCAL-6310 - SWS_Gpt_00215 : DisWakeup : Error : Invalid ch PUBLISHED	SWS_Gpt_00215 : DisWakeup : Error : Invalid ch
MCAL-6302 - SWS_Gpt_00228 : SetMode : Error :Uninitialized PUBLISHED	SWS_Gpt_00228 : SetMode : Error :Uninitialized
MCAL-6264 - SWS_Gpt_00379 : DisNotify : Error : no function configured PUBLISHED	SWS_Gpt_00379 : DisNotify : Error : no function configured
MCAL-6274 - SWS_Gpt_00231 : SetMode : Error : Invalid mode PUBLISHED	SWS_Gpt_00231 : SetMode : Error : Invalid mode

Design Identifier	Description
MCAL-6206 - SWS_Gpt_00227 : DisNotify : Error : Uninitialized PUBLISHED	SWS_Gpt_00227 : DisNotify : Error : Uninitialized
MCAL-6250 - SWS_Gpt_00377 : EnNotify : Error : No function PUBLISHED	SWS_Gpt_00377 : EnNotify : Error : No function
MCAL-6236 - SWS_Gpt_00214 : EnNotify : Error : Invalid channel PUBLISHED	SWS_Gpt_00214 : EnNotify : Error : Invalid channel
MCAL-6365 - SWS_Gpt_00226 : EnNotify : Error : Un Initialized PUBLISHED	SWS_Gpt_00226 : EnNotify : Error : Un Initialized
MCAL-6247 - SWS_Gpt_00225 : StopT : Error : Uninitialized PUBLISHED	SWS_Gpt_00225 : StopT : Error : Uninitialized
MCAL-6223 - SWS_Gpt_00213 : StopT : Error : Invalid channel ID PUBLISHED	SWS_Gpt_00213 : StopT : Error : Invalid channel ID
MCAL-6295 - SWS_Gpt_00099 : StopT : Error : No error if not running PUBLISHED	SWS_Gpt_00099 : StopT : Error : No error if not running

Design Identifier	Description
MCAL-6321 - SWS_Gpt_00084 : StartT : Error : Start on running timer PUBLISHED	SWS_Gpt_00084 : StartT : Error : Start on running timer
MCAL-6238 - SWS_Gpt_00224 : StartT : Development error : Config PUBLISHED	SWS_Gpt_00224 : StartT : Development error : Config
MCAL-6358 - SWS_Gpt_00223 : GTR : Error : UnInitialized PUBLISHED	SWS_Gpt_00223 : GTR : Error : UnInitialized
MCAL-6360 - SWS_Gpt_00212 : StartT : Error : Invalid channel PUBLISHED	SWS_Gpt_00212 : StartT : Error : Invalid channel
MCAL-6197 - [SWS_Gpt_00405] : Gpt Mandatory Interface PUBLISHED	[SWS_Gpt_00405] : Gpt Mandatory Interface
MCAL-6193 - [SWS_Gpt_00406] : Gpt Optional Interface PUBLISHED	[SWS_Gpt_00406] : Gpt Optional Interface
MCAL-6318 - SWS_Gpt_00346 : EC : GPT_E_BUSY PUBLISHED	SWS_Gpt_00346 : EC : GPT_E_BUSY

Design Identifier	Description
MCAL-6235 - SWS_Gpt_00332 : Functionality on Dev Error PUBLISHED	SWS_Gpt_00332 : Functionality on Dev Error
MCAL-6359 - SWS_Gpt_00338 : API : Dev Error PUBLISHED	SWS_Gpt_00338 : API : Dev Error
MCAL-6286 - SWS_Gpt_00307 : Init : State error detection PUBLISHED	SWS_Gpt_00307 : Init : State error detection
MCAL-6334 - SWS_Gpt_00234 : DeInit : Error if any timer running PUBLISHED	SWS_Gpt_00234 : DeInit : Error if any timer running
MCAL-6329 - SWS_Gpt_00217 : DisNotify : Error : Invalid channel PUBLISHED	SWS_Gpt_00217 : DisNotify : Error : Invalid channel
MCAL-6260 - SWS_Gpt_00220 : DeInit : Error if uninitialized PUBLISHED	SWS_Gpt_00220 : DeInit : Error if uninitialized
MCAL-6292 - SWS_Gpt_00211 : GTR : Error : Invalid Channel PUBLISHED	SWS_Gpt_00211 : GTR : Error : Invalid Channel
MCAL-6204 - SWS_Gpt_00218 : StartT : Error : Parameter check PUBLISHED	SWS_Gpt_00218 : StartT : Error : Parameter check

Design Identifier	Description
MCAL-6205 - SWS_Gpt_00210 : GTE : Error : Channel PUBLISHED	SWS_Gpt_00210 : GTE : Error : Channel
MCAL-6316 - SWS_Gpt_00222 : GTE : Error : Uninitialized PUBLISHED	SWS_Gpt_00222 : GTE : Error : Uninitialized

5 Implementation Details

5.1 Data structures and resources

MACROS, Data Types & Structures

The sections below lists some of key data structures that shall be implemented and used in driver implementation.

Maximum number of channels

Type	Identifier	Comments
uint32	GPT_MAX_CHANNELS	Defines the maximum number of channels that are configured. Its required that configurations for all channel specified is valid.

Gpt_ValueType

Used to specify the timer ticks, please refer section 8.3.2 of Reference 1 - AUTOSAR 4.3.1.

Gpt_ModeType

Enumeration, refer section 8.3.4 of Reference 1 - AUTOSAR 4.3.1.

Gpt_PredefTimerType

Enumeration, refer section 8.3.5 of Reference 1 - AUTOSAR 4.3.1.

Gpt_NotifyType

Is a function pointer with prototype as **void Gpt_Notification_<channel> (void)"** . The "_channel" postfix is provided by integrators via the configurator. "_channel" is used by applications to uniquely identify the GPT channel. Refer section 8.7.3.1 of Reference 1 - AUTOSAR 4.3.1.

Design Identifier	Description
MCAL-6271 - SWS_Gpt_00086 : NotifyCb : Provided as function ptr PUBLISHED	SWS_Gpt_00086 : NotifyCb : Provided as function ptr

Gpt_ChannelMode

Refer section 10.6.2 of Reference 1 - AUTOSAR 4.3.1.

Gpt_RegisterReadbackType

Name	Type	Range	Comments
gptRev	uint32	0 to 0xFFFFFFFF	H/W version identifier, will not change for a given SoC
gptTtgr	uint32	0 to 0xFFFFFFFF	Shall always read 0xFFFFFFFF
gptTimerSynCtrl	uint32	0 to 0xFFFFFFFF	Interface control register, will read 0x00000000

Gpt_ChannelConfigType

Used to define channel specific parameters for, one channel and the values of these are expected to be populated by configurator.

Type	Variable Name	Comments
uint32	channelId	Used to identify the channel (instance of timer (h/w) in implementation
Gpt_ChannelMode	channelMode	Used to specify the mode of channel
uint32	tickValueMax	Maximum value in ticks, the timer channel is able to count. With the next tick, the timer rolls over to zero
uint8	enableWakeupFlag	Flag indicating, if wakeup functionality is to be supported on this channel
Gpt_NotifyType	fnPtrNotifyFunction	Pointer to function, which shall be called to notify. Expected to be populated by the configurator
uint8	wakeupSourceRef	Value that shall be passed as function argument, when wakeup event occurs
uint32	prescale	The peripheral provide ability to scale function clock (FCLK), this parameter shall define scaling factor for this clock. The value/range would be device specific, check with device TRM for details.

Design Identifier	Description
MCAL-6251 - SWS_Gpt_00362 : NotifyCb : CB defined by CFG tool PUBLISHED	SWS_Gpt_00362 : NotifyCb : CB defined by CFG tool
MCAL-6349 - SWS_Gpt_00360 : Type : Mode PUBLISHED	SWS_Gpt_00360 : Type : Mode

Design Identifier	Description
MCAL-6282 - SWS_Gpt_00359 : Type : Value PUBLISHED	SWS_Gpt_00359 : Type : Value
MCAL-6368 - SWS_Gpt_00358 : Type : Channel PUBLISHED	SWS_Gpt_00358 : Type : Channel
MCAL-6322 - ECUC_Gpt_00235 : GptWakeupConfiguration PUBLISHED	ECUC_Gpt_00235 : GptWakeupConfiguration
MCAL-6305 - ECUC_Gpt_00312 : GptNotification PUBLISHED	ECUC_Gpt_00312 : GptNotification
MCAL-6245 - ECUC_Gpt_00311 : GptEnableWakeup PUBLISHED	ECUC_Gpt_00311 : GptEnableWakeup
MCAL-6328 - ECUC_Gpt_00309 : GptChannelMode PUBLISHED	ECUC_Gpt_00309 : GptChannelMode
MCAL-6290 - ECUC_Gpt_00184 : GptChannelConfiguration PUBLISHED	ECUC_Gpt_00184 : GptChannelConfiguration
MCAL-6214 - ECUC_Gpt_00269 : GptChannelConfigSet PUBLISHED	ECUC_Gpt_00269 : GptChannelConfigSet

Gpt_ConfigType

Used to define all channels specific parameters, shall be supplied to Gpt_Init () function. Values of these are expected to be populated by configurator.

Type	Variable Name	Comments
Gpt_ChannelConfigType *	channelCfgPtr	Is a constant pointer, pointing to all channel specific parameters. For “VARIANT-PRE-COMPILE” an instance of this structure shall instantiated in Gpt_Cfg.c by the configurator and driver implementation shall refer directly. Consider naming this variable as GptChannelConfigSet_PC For “VARIANT-POST-BUILD” an pointer of this type shall be provided to Gpt_Init () function.

Gpt_ConfigChannelType_PC

Used to define all channels identifiers, values of these are expected to be populated by configurator.

Type	Variable Name	Comments
Gpt_ChannelType	channelId[GPT_MAX_CHANNELS]	Shall contain channel identifiers for all channels. For “VARIANT-PRE-COMPILE” an instance of this structure shall instantiated in Gpt_Cfg.c by the configurator and driver implementation shall refer directly. Consider naming this variable as GptChannelIdConfig_PC. For “VARIANT-POST-BUILD”, this shall not be used, i.e. Gpt_Init (NULL_PTR)

Global Variables

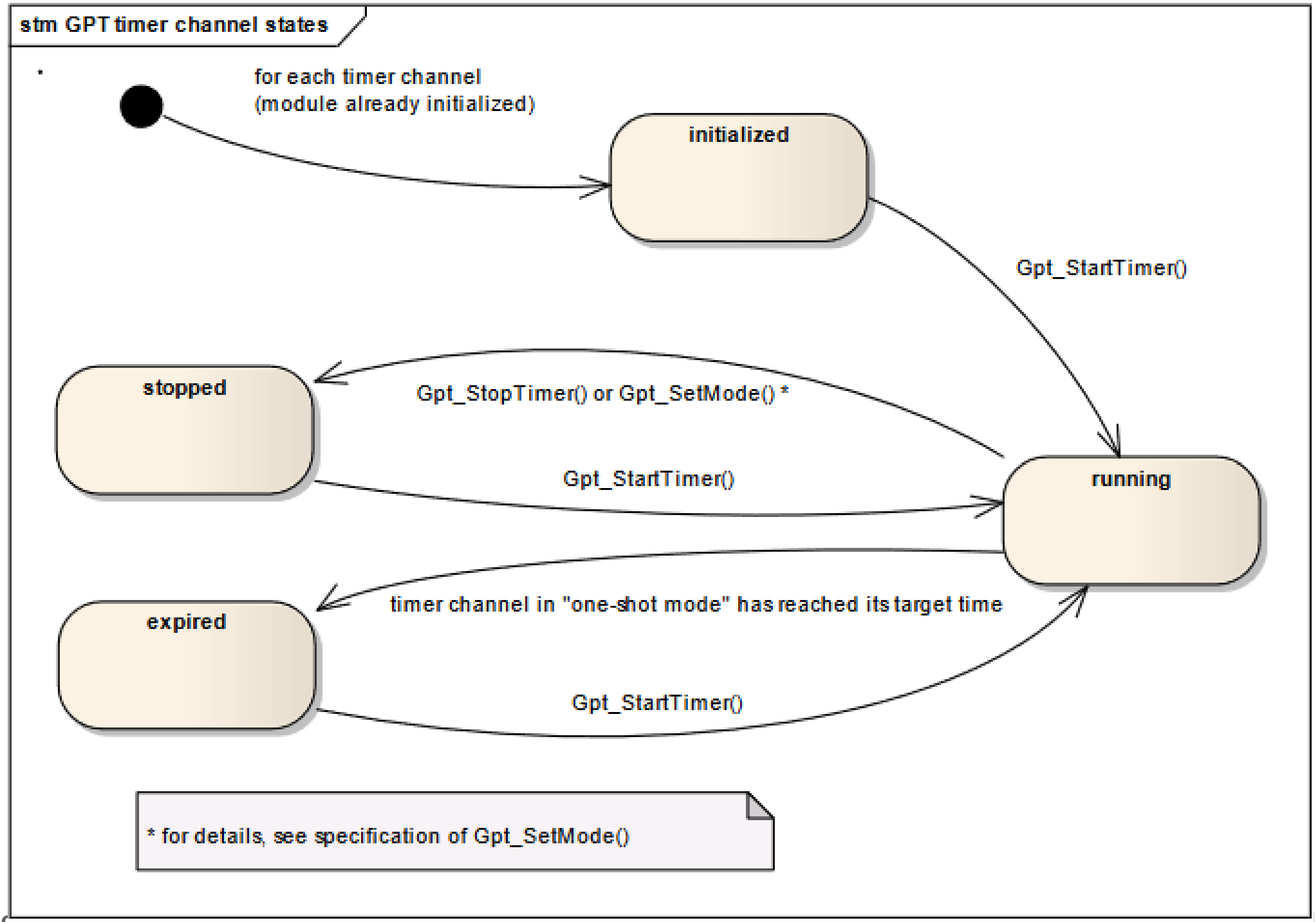
This design expects that implementation will require to use following global variables.

Variable	Type	Description	Default Value
Gpt_DrvStatus	uint32	Initialization status of the driver is maintained	FALSE
Gpt_DrvObj	Gpt_DriverObjType	GPT driver object, local to the implementation and scope shall NOT be limited to Gpt.c	Undefined
Gpt_WakeupSourceType	EcuM_WakeupSourceType	Used to store the WakeupSource Ref value for all configured channels. Local to the driver implementation and scope shall be limited to Gpt.c	Undefined
Gpt_IsrFxn	Gpt_IsrRefType	Array of pointers to the call notification and wakeup Functions	Undefined

5.2 Dynamic Behavior - Control Flow Diagram

States

As detailed in section 7.1 of Reference 1 - AUTOSAR 4.3.1, a timer would be in one of the following states. Initialized, running, stopped, expired. A variable shall be maintained on per channel basis to track and maintain the state. The diagram below shows transitions of states and it’s associated service API’s.


stm GPT timer channel states
for each timer channel
(module already initialized)
initialized
Gpt_StartTimer()
stopped
Gpt_StopTimer() or Gpt_SetMode() *
Gpt_StartTimer()
running
expired
timer channel in "one-shot mode" has reached its target time
Gpt_StartTimer()
* for details, see specification of Gpt_SetMode()

Timer States : Sourced from AUTOSAR Spec

5.3 Dynamic Behavior - Data Flow Diagram

Not Applicable

5.4 Application Parameters

Design Identifier	Description
MCAL-6344 - SWS_Gpt_00357 : Type : Config PUBLISHED	SWS_Gpt_00357 : Type : Config
MCAL-6313 - SWS_Gpt_00350 : EC : GPT_E_PARAM_POINTER PUBLISHED	SWS_Gpt_00350 : EC : GPT_E_PARAM_POINTER
MCAL-6285 **- SWS_Gpt_00349 : EC : GPT_E_PARAM_VALUE** PUBLISHED	SWS_Gpt_00349 : EC : GPT_E_PARAM_VALUE

Design Identifier	Description
MCAL-6269 - SWS_Gpt_00347 : EC : GPT_E_ALREADY_INITIALIZED PUBLISHED	SWS_Gpt_00347 : EC : GPT_E_ALREADY_INITIALIZED
MCAL-6258 - SWS_Gpt_00351 : EC : GPT_E_PARAM_MODE PUBLISHED	SWS_Gpt_00351 : EC : GPT_E_PARAM_MODE
MCAL-6232 - SWS_Gpt_00348 : EC : GPT_E_PARAM_CHANNEL PUBLISHED	SWS_Gpt_00348 : EC : GPT_E_PARAM_CHANNEL
MCAL-6196 - SWS_Gpt_00015 : DisNotify : Disable notification PUBLISHED	SWS_Gpt_00015 : DisNotify : Disable notification

Gpt_Init

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
CfgPtr	Pointer to configuration set	0xFFFFFFFF	-	-	NA

Gpt_GetTimeElapsed

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
Channel	Channel number should be less than GPT_MAX_CHANNEL	0-30	-	-	N.A

Gpt_GetTimeRemaining

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
Channel	Channel number should be less than GPT_MAX_CHANNEL	0-30	-	-	N.A

Gpt_StartTimer

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
Channel	Channel number should be less than GPT_MAX_CHANNEL	0-30	-	-	N.A
Value	Value should be less than GPT_TIMER_MAX_RESOLUTION	0xFFFFFFFF	-	-	N.A

Gpt_StopTimer

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
Channel	Channel number should be less than GPT_MAX_CHANNEL	0-30	-	-	N.A

Gpt_EnableNotification

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
Channel	Channel number should be less than GPT_MAX_CHANNEL	0-30	-	-	N.A

Gpt_DisableNotification

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
Channel	Channel number should be less than GPT_MAX_CHANNEL	0-30	-	-	N.A

Gpt_SetMode

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
Mode	Sleepmode	1	-	-	N.A

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
	Normalmode	0	-	-	N.A

Gpt_DisableWakeup

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
Channel	Channel number should be less than GPT_MAX_CHANNEL	0-30	-	-	N.A

Gpt_EnableWakeup

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
Channel	Channel number should be less than GPT_MAX_CHANNEL	0-30	-	-	N.A

Gpt_CheckWakeup

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
WakeupSource	Information on wakeup source on corresponding GPT channel	0-1	-	-	N.A

Gpt_GetVersionInfo

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
VersionInfoPtr	Pointer to where to store the version information of this module	0xFFFFFFFF	-	-	N.A

Gpt_RegisterReadback

Parameter	Description	Possible Value ranges	Unit of Value	Default Value	Variant
GptChannel	No of channels should be less than GPT_MAX_CHANNEL	0-30	-	-	N.A
RegRbPtr	Pointer to store the readback values	0xFFFFFFFF	-	-	N.A

5.5 Safety Diagnostic Features

TIM3 - Software readback of written configuration / TIM4 - Periodic Software Readback of Static Configuration Registers

Software Readback of Written Configuration ensures that the configuration register are written with the expected value. Periodic readback of configuration registers can provide a diagnostic for inadvertent writes to these registers.

The GPT MCAL driver provides the API - **Gpt_RegisterReadback** to readback static and written configuration registers to implement this diagnostic feature.

Design Identifier	Description
MCAL-6354 - GPT: Safety Diagnostics: TIM4: Periodic Software Readback of static configuraiton registers PUBLISHED	GPT: Safety Diagnostics: TIM4: Periodic Software Readback of static configuration registers
MCAL-6339 - GPT: Safety Diagnostics: TIM3: Software Readback of written configuration PUBLISHED	GPT: Safety Diagnostics: TIM3: Software Readback of written configuration

6 Low Level Definitions

6.1 Driver API's

For the standard API's please refer 8.3 of Reference 1 - AUTOSAR 4.3.1. Sections below highlight other design considerations for the implementation.

6.1.1 Gpt_Init

Refer section 8.3.2 of Reference 1 - AUTOSAR 4.3.1.

Design identifier	Description
MCAL-6370 - SWS_Gpt_00309 : Init : Re init post De Init PUBLISHED	SWS_Gpt_00309 : Init : Re init post De Init
MCAL-6248 - SWS_Gpt_00339 : Init : Mode normal PUBLISHED	SWS_Gpt_00339 : Init : Mode normal
MCAL-6326 - SWS_Gpt_00258 : Init : Disable int PUBLISHED	SWS_Gpt_00258 : Init : Disable int
MCAL-6286 - SWS_Gpt_00307 : Init : State error detection PUBLISHED	SWS_Gpt_00307 : Init : State error detection

Design identifier	Description
MCAL-6212 - SWS_Gpt_00356 : Init : All other registers PUBLISHED	SWS_Gpt_00356 : Init : All other registers
MCAL-6287 - SWS_Gpt_00354 : Init : IP : Mcu PUBLISHED	SWS_Gpt_00354 : Init : IP : Mcu
MCAL-6227 - SWS_Gpt_00353 : Init : Pins : Port PUBLISHED	SWS_Gpt_00353 : Init : Pins : Port
MCAL-6261 - SWS_Gpt_00352 : Init : IP registers PUBLISHED	SWS_Gpt_00352 : Init : IP registers
MCAL-6338 - SWS_Gpt_00068 : Init : Only configured resources PUBLISHED	SWS_Gpt_00068 : Init : Only configured resources
MCAL-6350 - SWS_Gpt_00107 : Init : Disable Int PUBLISHED	SWS_Gpt_00107 : Init : Disable Int
MCAL-6335 - SWS_Gpt_00006 : Init : Init GPT hw PUBLISHED	SWS_Gpt_00006 : Init : Init GPT hw

Design identifier	Description
MCAL-6288 - SWS_Gpt_00257 : PC : Params Ptr is NULL for Init Service API PUBLISHED	SWS_Gpt_00257 : PC : Params Ptr is NULL for Init Service API
MCAL-6284 - SWS_Gpt_00355 : Init : Onetime write : Startup code PUBLISHED	SWS_Gpt_00355 : Init : Onetime write : Startup code
MCAL-6243 - SWS_Gpt_00404 : EC : GPT_E_INIT_FAILED PUBLISHED	SWS_Gpt_00404 : EC : GPT_E_INIT_FAILED

6.1.2 Gpt_DeInit

Refer section 8.3.3 of Reference 1 - AUTOSAR 4.3.1.

Design identifier	Description
MCAL-6370 - SWS_Gpt_00309 : Init : Re init post De Init PUBLISHED	SWS_Gpt_00309 : Init : Re init post De Init
MCAL-6332 - SWS_Gpt_00363 : DeInit : state to uninitialized PUBLISHED	SWS_Gpt_00363 : DeInit : state to uninitialized

Design identifier	Description
MCAL-6260 - SWS_Gpt_00220 : DeInit : Error if uninitialized PUBLISHED	SWS_Gpt_00220 : DeInit : Error if uninitialized
MCAL-6211 - SWS_Gpt_00105 : DeInit : Disable notify and wakeup ints PUBLISHED	SWS_Gpt_00105 : DeInit : Disable notify and wakeup ints
MCAL-6334 - SWS_Gpt_00234 : DeInit : Error if any timer running PUBLISHED	SWS_Gpt_00234 : DeInit : Error if any timer running
MCAL-6213 - ECUC_Gpt_00314 : GptDeinitApi PUBLISHED	ECUC_Gpt_00314 : GptDeinitApi
MCAL-6353 - SWS_Gpt_00308 : DeInit : Only PB configured resources PUBLISHED	SWS_Gpt_00308 : DeInit : Only PB configured resources
MCAL-6219 - SWS_Gpt_00162 : DeInit : Only configured resources PUBLISHED	SWS_Gpt_00162 : DeInit : Only configured resources
MCAL-6221 - SWS_Gpt_00194 : DeInit : Only PC configured resources PUBLISHED	SWS_Gpt_00194 : DeInit : Only PC configured resources

Design identifier	Description
MCAL-6224 - SWS_Gpt_00008 : DeInit : Reset PUBLISHED	SWS_Gpt_00008 : DeInit : Reset
MCAL-6255 - SWS_Gpt_00345 : EC : GPT_E_UNINIT PUBLISHED	SWS_Gpt_00345 : EC : GPT_E_UNINIT

6.1.3 Gpt_GetTimeElapsed

Refer section 8.3.4 of Reference 1 - AUTOSAR 4.3.1.

Design identifier	Description
MCAL-6205 - SWS_Gpt_00210 : GTE : Error : Channel PUBLISHED	SWS_Gpt_00210 : GTE : Error : Channel
MCAL-6316 - SWS_Gpt_00222 : GTE : Error : Uninitialized PUBLISHED	SWS_Gpt_00222 : GTE : Error : Uninitialized
MCAL-6330 - SWS_Gpt_00361 : GTE : Continuous mode PUBLISHED	SWS_Gpt_00361 : GTE : Continuous mode

Design identifier	Description
MCAL-6257 - SWS_Gpt_00113 : GTE : reentrant PUBLISHED	SWS_Gpt_00113 : GTE : reentrant
MCAL-6357 - SWS_Gpt_00195 : GTE : cfg compile time PUBLISHED	SWS_Gpt_00195 : GTE : cfg compile time
MCAL-6343 - SWS_Gpt_00299 : GTE : oneshot : expired PUBLISHED	SWS_Gpt_00299 : GTE : oneshot : expired
MCAL-6266 - SWS_Gpt_00297 : GTE : Stop state PUBLISHED	SWS_Gpt_00297 : GTE : Stop state
MCAL-6252 - SWS_Gpt_00295 : GTE : Init state PUBLISHED	SWS_Gpt_00295 : GTE : Init state
MCAL-6361 - ECUC_Gpt_00317 : GptTimeElapsedApi PUBLISHED	ECUC_Gpt_00317 : GptTimeElapsedApi
MCAL-6362 - SWS_Gpt_00010 : GTE : oneshot mode PUBLISHED	SWS_Gpt_00010 : GTE : oneshot mode

6.1.4 Gpt_GetTimeRemaining

Refer section 8.3.5 of Reference 1 - AUTOSAR 4.3.1.

Design identifier	Description
MCAL-6325 - SWS_Gpt_00083 : GTR : remaining value PUBLISHED	SWS_Gpt_00083 : GTR : remaining value
MCAL-6234 - SWS_Gpt_00301 : GTR : in init state PUBLISHED	SWS_Gpt_00301 : GTR : in init state
MCAL-6342 - SWS_Gpt_00303 : GTR : in stopped state PUBLISHED	SWS_Gpt_00303 : GTR : in stopped state
MCAL-6345 - SWS_Gpt_00305 : GTR : One shot mode PUBLISHED	SWS_Gpt_00305 : GTR : One shot mode
MCAL-6363 - SWS_Gpt_00114 : GTR : Re Entrant PUBLISHED	SWS_Gpt_00114 : GTR : Re Entrant
MCAL-6272 - SWS_Gpt_00196 : GTR : Cfg compile time PUBLISHED	SWS_Gpt_00196 : GTR : Cfg compile time
MCAL-6358 - SWS_Gpt_00223 : GTR : Error : UnInitialized PUBLISHED	SWS_Gpt_00223 : GTR : Error : UnInitialized

Design identifier	Description
MCAL-6292 - SWS_Gpt_00211 : GTR : Error : Invalid Channel PUBLISHED	SWS_Gpt_00211 : GTR : Error : Invalid Channel
MCAL-6289 - ECUC_Gpt_00318 : GptTimeRemainingApi PUBLISHED	ECUC_Gpt_00318 : GptTimeRemainingApi

6.1.5 Gpt_StartTimer

Refer section In addition to details provided in 8.3.6 of Reference 1 - AUTOSAR 4.3.1 following design shall be implemented.

- Interrupt shall be enabled for channels, for which the notification or wake-enable is enabled.
- The timer operating in ONE-SHOT mode the interrupt shall be enabled, irrespective of the enabled/disabled notification. As ISR would be right place to mark the channel as EXPIRED, otherwise the API's like Gpt_GetTimeRemaining () and Gpt_ElapsedTime () would require special handling for EXPIRED timers.

Design Identifier	Description
MCAL-6204 - SWS_Gpt_00218 : StartT : Error : Parameter check PUBLISHED	SWS_Gpt_00218 : StartT : Error : Parameter check
MCAL-6309 - SWS_Gpt_00274 : StartT : Selected timer PUBLISHED	SWS_Gpt_00274 : StartT : Selected timer

Design Identifier	Description
MCAL-6254 - SWS_Gpt_00275 : StartT : wakeup or interrupt PUBLISHED	SWS_Gpt_00275 : StartT : wakeup or interrupt
MCAL-6210 - SWS_Gpt_00115 : StartT : Re Entrant PUBLISHED	SWS_Gpt_00115 : StartT : Re Entrant
MCAL-6321 - SWS_Gpt_00084 : StartT : Error : Start on running timer PUBLISHED	SWS_Gpt_00084 : StartT : Error : Start on running timer
MCAL-6230 - SWS_Gpt_00364 : StartT : state to running PUBLISHED	SWS_Gpt_00364 : StartT : state to running
MCAL-6360 - SWS_Gpt_00212 : StartT : Error : Invalid channel PUBLISHED	SWS_Gpt_00212 : StartT : Error : Invalid channel
MCAL-6238 - SWS_Gpt_00224 : StartT : Development error : Config PUBLISHED	SWS_Gpt_00224 : StartT : Development error : Config

6.1.6 **Gpt_StopTimer**

Refer section 8.3.7 of Reference 1 - AUTOSAR 4.3.1.

Design Identifier	Description
MCAL-6247 - SWS_Gpt_00225 : StopT : Error : Uninitialized PUBLISHED	SWS_Gpt_00225 : StopT : Error : Uninitialized
MCAL-6280 - SWS_Gpt_00343 : StopT : State changed to stopped PUBLISHED	SWS_Gpt_00343 : StopT : State changed to stopped
MCAL-6237 - SWS_Gpt_00013 : StopT : Selected timer channel PUBLISHED	SWS_Gpt_00013 : StopT : Selected timer channel
MCAL-6295 - SWS_Gpt_00099 : StopT : Error : No error if not running PUBLISHED	SWS_Gpt_00099 : StopT : Error : No error if not running
MCAL-6239 - SWS_Gpt_00344 : StopT : No action if not running PUBLISHED	SWS_Gpt_00344 : StopT : No action if not running
MCAL-6273 - SWS_Gpt_00116 : StopT : Re Entrant PUBLISHED	SWS_Gpt_00116 : StopT : Re Entrant
MCAL-6223 - SWS_Gpt_00213 : StopT : Error : Invalid channel ID PUBLISHED	SWS_Gpt_00213 : StopT : Error : Invalid channel ID

6.1.7 Gpt_EnableNotification

Refer section 8.3.8 of Reference 1 - AUTOSAR 4.3.1.

Design Identifier	Description
MCAL-6241 - SWS_Gpt_00199 : EnNotify : Pre Compile time CFG PUBLISHED	SWS_Gpt_00199 : EnNotify : Pre Compile time CFG
MCAL-6192 - SWS_Gpt_00331 : Notify Enable PUBLISHED	SWS_Gpt_00331 : Notify Enable
MCAL-6236 - SWS_Gpt_00214 : EnNotify : Error : Invalid channel PUBLISHED	SWS_Gpt_00214 : EnNotify : Error : Invalid channel
MCAL-6250 - SWS_Gpt_00377 : EnNotify : Error : No function PUBLISHED	SWS_Gpt_00377 : EnNotify : Error : No function
MCAL-6306 - SWS_Gpt_00014 : EnNotify : Enable interrupt PUBLISHED	SWS_Gpt_00014 : EnNotify : Enable interrupt
MCAL-6308 - SWS_Gpt_00117 : EnNotify : Re Entrant PUBLISHED	SWS_Gpt_00117 : EnNotify : Re Entrant
MCAL-6365 - SWS_Gpt_00226 : EnNotify : Error : Un Initialized PUBLISHED	SWS_Gpt_00226 : EnNotify : Error : Un Initialized

6.1.8 Gpt_DisableNotification

In addition to details provided in 8.3.9 of Reference 1 - AUTOSAR 4.3.1 following design shall be implemented.

- Interrupt shall be enabled for provided channel, when the channel state is in RUNNING or STOPPED or EXPIRED or INITIALIZED

Design Identifier	Description
MCAL-6264 - SWS_Gpt_00379 : DisNotify : Error : no function configured PUBLISHED	SWS_Gpt_00379 : DisNotify : Error : no function configured
MCAL-6281 - SWS_Gpt_00118 : DisNotify : Re Entrant PUBLISHED	SWS_Gpt_00118 : DisNotify : Re Entrant
MCAL-6244 - ECUC_Gpt_00315 : GptEnableDisableNotificationApi PUBLISHED	ECUC_Gpt_00315 : GptEnableDisableNotificationApi
MCAL-6329 - SWS_Gpt_00217 : DisNotify : Error : Invalid channel PUBLISHED	SWS_Gpt_00217 : DisNotify : Error : Invalid channel
MCAL-6206 - SWS_Gpt_00227 : DisNotify : Error : Uninitialized PUBLISHED	SWS_Gpt_00227 : DisNotify : Error : Uninitialized
MCAL-6217 - SWS_Gpt_00200 : DisNotify : Compile time cfg PUBLISHED	SWS_Gpt_00200 : DisNotify : Compile time cfg

6.1.9 Gpt_SetMode

Refer section 8.3.10 of Reference 1 - AUTOSAR 4.3.1

Design Identifier	Description
MCAL-6327 - SWS_Gpt_00341 : SetMode : No automatic wakeup of sleep ch PUBLISHED	SWS_Gpt_00341 : SetMode : No automatic wakeup of sleep ch
MCAL-6319 - SWS_Gpt_00201 : SetMode : Compile time cfg PUBLISHED	SWS_Gpt_00201 : SetMode : Compile time cfg
MCAL-6259 - SWS_Gpt_00151 : SetMode : Cfg timer for mode PUBLISHED	SWS_Gpt_00151 : SetMode : Cfg timer for mode
MCAL-6276 - SWS_Gpt_00255 : SetMode : Available only if cfg PUBLISHED	SWS_Gpt_00255 : SetMode : Available only if cfg
MCAL-6333 - SWS_Gpt_00152 : SetMode : Normal PUBLISHED	SWS_Gpt_00152 : SetMode : Normal
MCAL-6202 - SWS_Gpt_00153 : SetMode : Sleep PUBLISHED	SWS_Gpt_00153 : SetMode : Sleep

Design Identifier	Description
MCAL-6366 - SWS_Gpt_00164 : SetMode : Normal : Stop non sleep ch PUBLISHED	SWS_Gpt_00164 : SetMode : Normal : Stop non sleep ch
MCAL-6351 - SWS_Gpt_00165 : SetMode : Normal : No Re Start of sleep ch PUBLISHED	SWS_Gpt_00165 : SetMode : Normal : No Re Start of sleep ch
MCAL-6302 - SWS_Gpt_00228 : SetMode : Error :Uninitialized PUBLISHED	SWS_Gpt_00228 : SetMode : Error :Uninitialized
MCAL-6274 - SWS_Gpt_00231 : SetMode : Error : Invalid mode PUBLISHED	SWS_Gpt_00231 : SetMode : Error : Invalid mode

6.1.10 Gpt_DisableWakeup

Refer section 8.3.11 of Reference 1 - AUTOSAR 4.3.1

Design Identifier	Description
MCAL-6340 - SWS_Gpt_00229 : DisWakeup : Error : Un Initialized PUBLISHED	SWS_Gpt_00229 : DisWakeup : Error : Un Initialized

Design Identifier	Description
MCAL-6242 - ECUC_Gpt_00320 : GptWakeupFunctionalityApi PUBLISHED	ECUC_Gpt_00320 : GptWakeupFunctionalityApi
MCAL-6293 - SWS_Gpt_00159 : DisWakeup : wakeup disabled PUBLISHED	SWS_Gpt_00159 : DisWakeup : wakeup disabled
MCAL-6267 - SWS_Gpt_00157 : DisWakeup : Only if wakeup src cfg PUBLISHED	SWS_Gpt_00157 : DisWakeup : Only if wakeup src cfg
MCAL-6320 - SWS_Gpt_00155 : DisWakeup : re entrant PUBLISHED	SWS_Gpt_00155 : DisWakeup : re entrant
MCAL-6246 - SWS_Gpt_00202 : DisWakeup : PreCompile time cfg PUBLISHED	SWS_Gpt_00202 : DisWakeup : PreCompile time cfg
MCAL-6310 - SWS_Gpt_00215 : DisWakeup : Error : Invalid ch PUBLISHED	SWS_Gpt_00215 : DisWakeup : Error : Invalid ch

6.1.11 **Gpt_EnableWakeup**

Refer section 8.3.12 of Reference 1 - AUTOSAR 4.3.1

Design Identifier	Description
MCAL-6242 - ECUC_Gpt_00320 : GptWakeupFunctionalityApi PUBLISHED	ECUC_Gpt_00320 : GptWakeupFunctionalityApi
MCAL-6367 - SWS_Gpt_00230 : EnWakeup : Error : Uninitialized PUBLISHED	SWS_Gpt_00230 : EnWakeup : Error : Uninitialized
MCAL-6291 - SWS_Gpt_00158 : EnWakeup : Only if wakeup source cfg PUBLISHED	SWS_Gpt_00158 : EnWakeup : Only if wakeup source cfg
MCAL-6323 - SWS_Gpt_00216 : EnWakeup : Error : Invalid ch PUBLISHED	SWS_Gpt_00216 : EnWakeup : Error : Invalid ch
MCAL-6311 - SWS_Gpt_00156 : EnWakeup : Re Entrant PUBLISHED	SWS_Gpt_00156 : EnWakeup : Re Entrant
MCAL-6366 - SWS_Gpt_00164 : SetMode : Normal : Stop non sleep ch PUBLISHED	SWS_Gpt_00164 : SetMode : Normal : Stop non sleep ch
MCAL-6355 - SWS_Gpt_00203 : EnWakeup : Compile time cfg PUBLISHED	SWS_Gpt_00203 : EnWakeup : Compile time cfg
MCAL-6202 - SWS_Gpt_00153 : SetMode : Sleep PUBLISHED	SWS_Gpt_00153 : SetMode : Sleep

Design Identifier	Description
MCAL-6337 - SWS_Gpt_00160 : EnWakeup : Enable int PUBLISHED	SWS_Gpt_00160 : EnWakeup : Enable int
MCAL-6307 - SWS_Gpt_00127 : Wakeup Enable PUBLISHED	SWS_Gpt_00127 : Wakeup Enable

6.1.12 Gpt_CheckWakeup

Refer section 8.3.13 of Reference 1 - AUTOSAR 4.3.1

Design Identifier	Description
MCAL-6242 - ECUC_Gpt_00320 : GptWakeupFunctionalityApi PUBLISHED	ECUC_Gpt_00320 : GptWakeupFunctionalityApi
MCAL-6312 - SWS_Gpt_00325 : CheckWakeup : Error : If uninitialized PUBLISHED	SWS_Gpt_00325 : CheckWakeup : Error : If uninitialized
MCAL-6195 - SWS_Gpt_00322 : CheckWakeup : if wakeup source cfg PUBLISHED	SWS_Gpt_00322 : CheckWakeup : if wakeup source cfg

Design Identifier	Description
MCAL-6348 - SWS_Gpt_00321 : CheckWakeup : check if ch is source for wakeup PUBLISHED	SWS_Gpt_00321 : CheckWakeup : check if ch is source for wakeup
MCAL-6256 - SWS_Gpt_00323 : CheckWakeup : Re Entrant PUBLISHED	SWS_Gpt_00323 : CheckWakeup : Re Entrant
MCAL-6279 - SWS_Gpt_00324 : CheckWakeup : Pre Compile time cfg PUBLISHED	SWS_Gpt_00324 : CheckWakeup : Pre Compile time cfg
MCAL-6199 - SWS_Gpt_00326 : CheckWakeup CB called in interrupt context PUBLISHED	SWS_Gpt_00326 : CheckWakeup CB called in interrupt context

6.1.13 Gpt_GetVersionInfo

Refer section 8.3.1 of Reference 1 - AUTOSAR 4.3.1

Design Identifier	Description
MCAL-6242 - ECUC_Gpt_00320 : GptWakeupFunctionalityApi PUBLISHED	ECUC_Gpt_00320 : GptWakeupFunctionalityApi

Design Identifier	Description
MCAL-6359 - SWS_Gpt_00338 : API : Dev Error PUBLISHED	SWS_Gpt_00338 : API : Dev Error

6.1.14 **Gpt_RegisterReadback**

As noted from previous implementation, the timer configuration registers could potentially be corrupted by other entities (s/w or h/w). One of the recommended detection methods would be to periodically read-back the configuration and confirm configuration is consistent. The service API defined below shall be implemented to enable this detection.

	Description	Comments
Service Name	Gpt_RegisterReadback	Can potentially be turned OFF
Syntax	Std_ReturnType Gpt_RegisterReadback(Gpt_ChannelType GptChannel, Gpt_RegisterReadbackType *RegRbPtr)	Gpt_RegisterReadbackType defines the type, that holds critical values, refer below
Service ID	0x0F	
Sync / Async	Sync	

Reentrancy	Non Reentrant	
Parameter in	GptChannel	Identifies a unique valid channel
Parameters out	RegRbPtr	A pointer of type Gpt_RegisterReadbackType, which holds the read back values
Return Value	Standard return type	E_OK or E_NOT_OK in case of GPT not initialized or NULL buffer pointer

This service could potentially be turned OFF in the configurator.

Notify ISR

On elapse of configured count, the timer peripheral generates an interrupt. The implementation shall provide an ISR with prototype as “void Gpt_<ChannelNum>Isr (void)” The control flow shall be as depicted in flow chart figure below. Since, the function prototype dosen’t take any arguments to uniquely identify the timer channel that caused this interrupt, a separate ISR shall be implemented for each configured / enabled channel.

Notify ISR Flow chart

Design Identifier	Description
MCAL-6300 - SWS_Gpt_00206 : NotifyCb : ISR calls CB and clear Ints PUBLISHED	SWS_Gpt_00206 : NotifyCb : ISR calls CB and clear Ints
MCAL-6220 - SWS_Gpt_00233 : NotifyCb : CB when target time reached PUBLISHED	SWS_Gpt_00233 : NotifyCb : CB when target time reached
MCAL-6341 - SWS_Gpt_00093 : NotifyCb : No CB when disabled PUBLISHED	SWS_Gpt_00093 : NotifyCb : No CB when disabled
MCAL-6369 - SWS_Gpt_00209 : NotifyCb : separate CB for each ch PUBLISHED	SWS_Gpt_00209 : NotifyCb : separate CB for each ch
MCAL-6201 - SWS_Gpt_00292 : NotifyCb : Service API PUBLISHED	SWS_Gpt_00292 : NotifyCb : Service API
MCAL-6364 - SWS_Gpt_00327 : ISR Shall reset int flags PUBLISHED	SWS_Gpt_00327 : ISR Shall reset int flags

Wakeup ISR

On elapse of configured count the timer peripheral generates an interrupt. The implementation shall provide an ISR with prototype as “void Gpt_<ChannelNum>Isr (void)” The control flow shall be as depicted in flow chart above, with following exceptions

1. Check for notify & call notify shall not be implemented, instead EcuM_CheckWakeup () shall be called with configured wakeup source.

2. Check for mode and its associated action for true condition shall not be implemented.

Design Identifier	Description
MCAL-6199 - SWS_Gpt_00326 : CheckWakeup CB called in interrupt context PUBLISHED	SWS_Gpt_00326 : CheckWakeup CB called in interrupt context
MCAL-6347 - SWS_Gpt_00270 : PC & PB in exclusive CFG containers PUBLISHED	SWS_Gpt_00270 : PC & PB in exclusive CFG containers

7 Performance Objectives

7.1 Resource Consumption Objectives

ROM - Program(KB)	ROM - Data(KB)	RAM - Program(KB)	RAM - Data(KB)	Stack Size (KB)	EEPROM (KB)	% CPU Utilization
5	NA	NA	1	2	NA	NA

7.2 Critical timing and Performance

Not Applicable

8 Decision Analysis & Resolution (DAR)

Sections below list some of the important design decisions and rational behind those decision.

8.1 Use of DM Timer Auto-Reload mode for GPT continuous mode

The timer hardware doesn’t support continuous mode if configured timer count is less than max count of timer (0xFFFFFFFF in 32 bit timer). To implement GPT continuous mode we have to use timer interrupt to trigger timer start or tweak timer configuration during timer start to work in Continuous mode.

No.	Decision Criteria	Alternatives	Selected alternative	Rationale	Trade-offs
1	Implementation of GPT ‘Continuous’ mode, without timing constraints (or programming registers in ISR)	**Use GPT count completion interrupt to restart timer:** For implementing continuous mode with this count provided to start timer function will be programmed as to match value for interrupt generation and timer will start counting from zero. Once it reaches count value it will generate match interrupt. Same will be used for notification and wake up event. Here driver will check if timer is configured in continuous mode and if yes it will trigger timer enable again. **Advantages:** • Simple to implement. No overhead configuration. **Disadvantages:**	Use of Timer auto reload mode	To avoid dependency on processor and interrupt service routine, recommended to use timer hardware auto-reload feature. Software will make sure to return correct values for time elapsed and time remaining functions.	Complex software implementation as software has to make sure correct value is written and also keep track of count value given by application for timer elapsed and time remaining functions.

No.	Decision Criteria	Alternatives	Selected alternative	Rationale	Trade-offs
		• Latency between timer interrupt generation and timer restart. This will be major issue as it will vary with Processor speed. Also will not be constant between two timer count completion cycles. • Dependency on ISR for timer restart			

No.	Decision Criteria	Alternatives	Selected alternative	Rationale	Trade-offs
		Use of Timer auto reload mode: Device timer HW supports auto-reload mode when timer overflow (max count possible for timer HW reached) occurs. In this case timer HW restarts timer with value loaded in Counter register (CR). To use this feature for GPT continuous mode we need to calculate count value with reference to max count. For example if count value is 0x10 then counter register should be programmed with (0xFFFF_FFFF – 0x10) value. This will ensure timer restarts after counting till overflow. **Advantages:** • No dependency on ISR for timer restart. This will be taken care by hardware.			

No.	Decision Criteria	Alternatives	Selected alternative	Rationale	Trade-offs
		- Independent of processor speed. **Disadvantages:** - Complex software implementation as software has to make sure correct value is written and also keep track of count value given by application for timer elapsed and time remaining functions.			

Design Identifier	Description
MCAL-6194 - SWS_Gpt_00330 : Mode : C : Free Running PUBLISHED	SWS_Gpt_00330 : Mode : C : Free Running
MCAL-6346 - SWS_Gpt_00186 : Mode : Continuous PUBLISHED	SWS_Gpt_00186 : Mode : Continuous
MCAL-6231 - SWS_Gpt_00329 : Timer Start value PUBLISHED	SWS_Gpt_00329 : Timer Start value

9 Testing Guidelines

The sections below identify some of the aspects of design that would require emphasis during testing of this design implementation

- **State Transitions**
    - Test cases shall exercise all state transitions as detailed in section (States)
    - Ensure non supported API's in a given state, returns valid error code
- **Wake functionality**
    - Test cases shall ensure, wake up functionality is exercised on one channel at least
- **Mode**
    - Test cases shall ensure, a timer shall be operable in all supported modes (but not concurrently, for a single channel)
- **Concurrency**
    - Test cases shall ensure, multiple channels can be operated concurrently
- **Timeout**
    - Test cases exercising Gpt_Start API, as perform equivalence class test on Gpt_ValueType Gpt_Start (). As a large Gpt_ValueType increase test cycle time
    - Large Gpt_ValueType shall be performed only for "Full Test Cycle"
- **Elapsed / Remaining time**
    - Test cases shall ensure API (when available) GptTimeElapsedApi () is invoked on elapsed timer (one shot mode) and value shall not change
    - Test cases shall ensure API (when available) GptTimeRemainingApi () is invoked on elapsed timer (one shot mode) and value shall not change

10 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
Yaniv Weizman	Adding requirement type column for requirements table (Functional/Non-Functional). Adding DAR table	0.65	13 Nov 2019

Author Name	Description	Version	Date
Yaniv Weizman	Adding tables for Testing guidelines	0.7	18 Nov 2019
Krishna	Updated based on ASPICE requirements	0.8	20 Aug 2020
Krishna	Updated based on the feedback from Jon N	0.9	09 Oct 2020
Krishna	Updated the traceability scheme	1.0	17 Dec 2020