

---

# User Manual

for S32K14X GPT Driver

Document Number: UM2GPTASR4.3 Rev0001R1.0.1  
Rev. 1.0





# Contents

Section number	Title	Page
<b>Chapter 1</b>		
<b>Revision History</b>		
<b>Chapter 2</b>		
<b>Introduction</b>		
2.1	Supported Derivatives.....	11
2.2	Overview.....	11
2.3	About this Manual.....	12
2.4	Acronyms and Definitions.....	12
2.5	Reference List.....	13
<b>Chapter 3</b>		
<b>Driver</b>		
3.1	Requirements.....	15
3.2	Driver Design Sumary.....	15
3.3	Hardware Resources.....	16
3.4	Deviation from Requirements.....	16
3.5	Driver limitations.....	17
3.6	Driver usage and configuration tips.....	17
3.6.1	Dual Clock Feature.....	18
3.6.2	Input trigger for channel.....	18
3.6.3	GptStandbyWakeupSupport.....	18
3.6.4	GptEnableUserModeSupport.....	19
3.6.5	GptChangeNextTimeoutValueApi.....	19
3.7	Runtime Errors.....	19
3.8	Software specification.....	19
3.8.1	Define Reference.....	20
3.8.1.1	Define GPT_VENDOR_ID_C.....	20
3.8.1.2	Define GPT_AR_RELEASE_MAJOR_VERSION_C.....	20
3.8.1.3	Define GPT_AR_RELEASE_MINOR_VERSION_C.....	20

Section number	Title	Page
3.8.1.4	Define GPT_AR_RELEASE_REVISION_VERSION_C.....	20
3.8.1.5	Define GPT_SW_MAJOR_VERSION_C.....	20
3.8.1.6	Define GPT_SW_MINOR_VERSION_C.....	21
3.8.1.7	Define GPT_SW_PATCH_VERSION_C.....	21
3.8.1.8	Define GPT_E_ALREADY_INITIALIZED.....	21
3.8.1.9	Define GPT_E_BUSY.....	22
3.8.1.10	Define GPT_E_INVALID_CALL.....	22
3.8.1.11	Define GPT_E_MODE.....	22
3.8.1.12	Define GPT_E_PARAM_CHANNEL.....	23
3.8.1.13	Define GPT_E_PARAM_MODE.....	23
3.8.1.14	Define GPT_E_PARAM_POINTER.....	23
3.8.1.15	Define GPT_E_PARAM_VALUE.....	24
3.8.1.16	Define GPT_E_PARAM_PREDEF_TIMER.....	24
3.8.1.17	Define GPT_E_UNINIT.....	25
3.8.1.18	Define GPT_CHECKWAKEUP_ID.....	25
3.8.1.19	Define GPT_DEINIT_ID.....	25
3.8.1.20	Define GPT_DISABLENOTIFICATION_ID.....	26
3.8.1.21	Define GPT_DISABLEWAKEUP_ID.....	26
3.8.1.22	Define GPT_ENABLENOTIFICATION_ID.....	26
3.8.1.23	Define GPT_ENABLEWAKEUP_ID.....	27
3.8.1.24	Define GPT_GETVERSIONINFO_ID.....	27
3.8.1.25	Define GPT_INIT_ID.....	27
3.8.1.26	Define GPT_INSTANCE_ID.....	28
3.8.1.27	Define GPT_MODULE_ID.....	28
3.8.1.28	Define GPT_PROCESSCOMMONINTERRUPT_ID.....	28
3.8.1.29	Define GPT_SETMODE_ID.....	29
3.8.1.30	Define GPT_SET_CLOCK_MODE_ID.....	29
3.8.1.31	Define GPT_STARTTIMER_ID.....	29
3.8.1.32	Define GPT_STOPTIMER_ID.....	30

Section number	Title	Page
3.8.1.33	Define GPT_TIMEELAPSED_ID.....	30
3.8.1.34	Define GPT_TIMEREMAINING_ID.....	30
3.8.1.35	Define GPT_CHANGE_NEXT_TIMEOUT_VALUE_ID.....	31
3.8.1.36	Define GPT_DEV_ERROR_DETECT.....	31
3.8.1.37	Define GPT_PRECOMPILE_SUPPORT.....	31
3.8.1.38	Define GPT_REPORT_WAKEUP_SOURCE.....	31
3.8.1.39	Define GPT_DEINIT_API.....	32
3.8.1.40	Define GPT_ENABLE_DISABLE_NOTIFICATION_API.....	32
3.8.1.41	Define GPT_PREDEFTIMER_FUNCTIONALITY_API.....	32
3.8.1.42	Define GPT_TIME_ELAPSED_API.....	33
3.8.1.43	Define GPT_TIME_REMAINING_API.....	33
3.8.1.44	Define GPT_USER_MODE_SOFT_LOCKING.....	33
3.8.1.45	Define GPT_VERSION_INFO_API.....	33
3.8.1.46	Define GPT_WAKEUP_FUNCTIONALITY_API.....	34
3.8.1.47	Define GPT_CHANGE_NEXT_TIMEOUT_VALUE.....	34
3.8.2	Enum Reference.....	34
3.8.2.1	Enumeration Gpt_ChannelModeType.....	34
3.8.2.2	Enumeration Gpt_ChannelStatusType.....	34
3.8.2.3	Enumeration Gpt_ClockModeType.....	35
3.8.2.4	Enumeration Gpt_ModeType.....	35
3.8.2.5	Enumeration Gpt_PredDefTimerType.....	35
3.8.3	Function Reference.....	36
3.8.3.1	Function Gpt_CheckWakeup.....	36
3.8.3.2	Function Gpt_DeInit.....	36
3.8.3.3	Function Gpt_DisableNotification.....	37
3.8.3.4	Function Gpt_DisableWakeup.....	38
3.8.3.5	Function Gpt_EnableNotification.....	38
3.8.3.6	Function Gpt_EnableWakeup.....	39
3.8.3.7	Function Gpt_GetPredDefTimerValue.....	39

Section number	Title	Page
3.8.3.8	Function Gpt_GetTimeElapsed.....	40
3.8.3.9	Function Gpt_GetTimeRemaining.....	40
3.8.3.10	Function Gpt_GetVersionInfo.....	41
3.8.3.11	Function Gpt_Init.....	41
3.8.3.12	Function Gpt_SetClockMode.....	42
3.8.3.13	Function Gpt_SetMode.....	43
3.8.3.14	Function Gpt_StartTimer.....	43
3.8.3.15	Function Gpt_StopTimer.....	44
3.8.3.16	Function Gpt_ChangeNextTimeoutValue.....	44
3.8.4	Structs Reference.....	45
3.8.4.1	Structure Gpt_ChannelConfigType.....	45
3.8.4.2	Structure Gpt_ConfigType.....	46
3.8.4.3	Structure Gpt_HwChannelConfigType.....	47
3.8.4.4	Structure Gpt_HwChannelInfoType.....	48
3.8.5	Types Reference.....	49
3.8.5.1	Typedef Gpt_ChannelType.....	49
3.8.5.2	Typedef Gpt_NotificationType.....	50
3.8.5.3	Typedef Gpt_ValueType.....	50
3.8.5.4	Typedef Gpt_PrescalerType.....	50
3.8.5.5	Typedef Gpt_ClockSourceType.....	50
3.9	Symbolic Names Disclaimer.....	50

## Chapter 4 Tresos Configuration Plug-in

4.1	Configuration elements of Gpt.....	53
4.2	Form IMPLEMENTATION_CONFIG_VARIANT.....	53
4.3	Form GptConfigurationOfOptApiServices.....	54
4.3.1	GptDeinitApi (GptConfigurationOfOptApiServices).....	54
4.3.2	GptEnableDisableNotificationApi (GptConfigurationOfOptApiServices).....	54
4.3.3	GptTimeElapsedApi (GptConfigurationOfOptApiServices).....	55

Section number	Title	Page
4.3.4	GptTimeRemainingApi (GptConfigurationOfOptApiServices).....	55
4.3.5	GptVersionInfoApi (GptConfigurationOfOptApiServices).....	55
4.3.6	GptWakeupFunctionalityApi (GptConfigurationOfOptApiServices).....	56
4.3.7	GptPredefTimerFunctionalityApi (GptConfigurationOfOptApiServices).....	56
4.4	Form GptNonAUTOSAR.....	56
4.4.1	GptChangeNextTimeoutValueApi (GptNonAUTOSAR).....	57
4.4.2	GptEnableDualClockMode (GptNonAUTOSAR).....	57
4.4.3	GptEnableTriggers (GptNonAUTOSAR).....	57
4.4.4	GptEnableUserModeSupport (GptNonAUTOSAR).....	58
4.4.5	GptStandbyWakeupSupport (GptNonAUTOSAR).....	58
4.5	Form GptDriverConfiguration.....	59
4.5.1	GptDevErrorDetect (GptDriverConfiguration).....	59
4.5.2	GptPredefTimer100us32bitEnable (GptDriverConfiguration).....	59
4.5.3	GptPredefTimer1usEnablingGrade (GptDriverConfiguration).....	60
4.5.4	GptReportWakeupSource (GptDriverConfiguration).....	60
4.5.5	GptRegisterLocking (GptDriverConfiguration).....	61
4.5.6	Form GptClockReferencePoint.....	61
4.5.6.1	GptClockReference (GptClockReferencePoint).....	61
4.6	Form GptPredefTimerConfiguration.....	62
4.7	Form GptChannelConfigSet.....	62
4.7.1	Form GptChannelConfiguration.....	63
4.7.1.1	GptChannelId (GptChannelConfiguration).....	63
4.7.1.2	GptHwChannel (GptChannelConfiguration).....	63
4.7.1.3	GptChannelMode (GptChannelConfiguration).....	64
4.7.1.4	GptChannelTickFrequency (GptChannelConfiguration).....	64
4.7.1.5	GptFtmPrescaler and GptFtmPrescaler_Alternate (GptChannelConfiguration).....	65
4.7.1.6	GptLptmrPrescaler and GptLptmrPrescaler_Alternative(GptChannelConfiguration).....	65
4.7.1.7	GptFtmChannelClkSrc (GptChannelConfiguration).....	66
4.7.1.8	GptLptmrChannelClkSrc (GptChannelConfiguration).....	67

Section number	Title	Page
4.7.1.9	GptSrcChannelClkSrc (GptChannelConfiguration).....	67
4.7.1.10	GptLPitIsExternalTrigger (GptChannelConfiguration).....	68
4.7.1.11	GptLPitEnReloadOnTrigger (GptChannelConfiguration).....	68
4.7.1.12	GptLPitEnStopOnInterrupt (GptChannelConfiguration).....	69
4.7.1.13	GptLPitEnStartOnTrigger (GptChannelConfiguration).....	69
4.7.1.14	GptLPitTriggerChannels (GptChannelConfiguration).....	69
4.7.1.15	GptChannelTickValueMax (GptChannelConfiguration).....	70
4.7.1.16	GptFreezeEnable (GptChannelConfiguration).....	70
4.7.1.17	GptEnableWakeup (GptChannelConfiguration).....	71
4.7.1.18	GptNotification (GptChannelConfiguration).....	71
4.7.2	Form GptHwInterruptChannel.....	71
4.7.2.1	GptIsrHwId (GptHwInterruptChannel).....	72
4.7.2.2	GptIsrEnable (GptHwInterruptChannel).....	72
4.7.2.3	GptChannelIsUsed (GptHwInterruptChannel).....	72
4.8	Form GptHwConfiguration.....	73
4.9	Form CommonPublishedInformation.....	73
4.9.1	ArReleaseMajorVersion (CommonPublishedInformation).....	74
4.9.2	ArReleaseMinorVersion (CommonPublishedInformation).....	74
4.9.3	ArReleaseRevisionVersion (CommonPublishedInformation).....	75
4.9.4	ModuleId (CommonPublishedInformation).....	75
4.9.5	SwMajorVersion (CommonPublishedInformation).....	76
4.9.6	SwMinorVersion (CommonPublishedInformation).....	76
4.9.7	SwPatchVersion (CommonPublishedInformation).....	77
4.9.8	VendorApiInfix (CommonPublishedInformation).....	77
4.9.9	VendorId (CommonPublishedInformation).....	77



# Chapter 1

## Revision History

**Table 1-1. Revision History**

Revision	Date	Author	Description
1.0	21/06/2019	NXP MCAL Team	Updated version for ASR 4.3.1S32K14XR1.0.1



# Chapter 2

## Introduction

This User Manual describes NXP Semiconductors AUTOSAR General Purpose Timer ( GPT ) for S32K14X .

AUTOSAR GPT driver configuration parameters and deviations from the specification are described in GPT Driver chapter of this document. AUTOSAR GPT driver requirements and APIs are described in the AUTOSAR GPT driver software specification document.

### 2.1 Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices of NXP Semiconductors .

**Table 2-1. S32K14X Derivatives**

NXP Semiconductors	s32k148_lqfp144, s32k148_lqfp176, s32k148_mapbga100, s32k146_lqfp144, s32k146_lqfp100, s32k146_lqfp64, s32k146_mapbga100, s32k144_lqfp100, s32k144_lqfp64, s32k144_mapbga100, s32k142_lqfp100, s32k142_lqfp64, s32k118_lqfp48, s32k118_lqfp64, s32k142_lqfp48, s32k144_lqfp48, s32k148_lqfp100
--------------------	--

All of the above microcontroller devices are collectively named as S32K14X .

### 2.2 Overview

**AUTOSAR (AUTomotive Open System ARchitecture)** is an industry partnership working to establish standards for software interfaces and software modules for automobile electronic control systems.

## AUTOSAR

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.
- is a strong global partnership that creates one common standard: "Cooperate on standards, compete on implementation".
- is a key enabling technology to manage the growing electrics/electronics complexity. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.
- facilitates the exchange and update of software and hardware over the service life of the vehicle.

## 2.3 About this Manual

This Technical Reference employs the following typographical conventions:

**Boldface type:** Bold is used for important terms, notes and warnings.

*Italic font:* Italic typeface is used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

### Note

This is a note.

## 2.4 Acronyms and Definitions

**Table 2-2. Acronyms and Definitions**

Term	Definition
API	Application Programming Interface
AUTOSAR	AUTomotive Open System ARchitecture
ASM	Assembler
BSMI	Basic Software Makefile Interface
C/CPP	C and C++ Source Code
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ETIMER	Enhanced Motor Control Timer
GPT	General Purpose Timer

*Table continues on the next page...*

**Table 2-2. Acronyms and Definitions (continued)**

Term	Definition
ISR	Interrupt Service Routine
MCU	Micro Controller Unit
N/A	Not Applicable
LPIT	Low Power Interrupt Timer
LPTMR	Low Power Timer
RTC	Real Time Clock
FTM	FlexTimer Module

## 2.5 Reference List

**Table 2-3. Reference List**

#	Title	Version
1	Specification of GPT Driver	AUTOSAR Release 4.3.1
2	S32K14X Reference Manual	Reference Manual, Rev. 9, 9/2018
3	S32K142 Mask Set Errata for Mask 0N33V (0N33V)	30/11/2017
4	S32K144 Mask Set Errata for Mask 0N57U (0N57U)	30/11/2017
5	S32K146 Mask Set Errata for Mask 0N73V (0N73V)	30/11/2017
6	S32K148 Mask Set Errata for Mask 0N20V (0N20V)	25/10/2018
7	S32K118 Mask Set Errata for Mask 0N97V (0N97V)	07/01/2019



# Chapter 3

## Driver

### 3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 4.3 Rev0001GPT Driver Software Specification document (See Table [Reference List](#) ).

### 3.2 Driver Design Summary

The GPT driver provides services for starting and stopping timer channels (logical timer instances assigned to a timer hardware).

A timer channel can be configured and run in "one-shot mode" or in "continuous mode", and the elapsed time and time remaining are also supported.

The GPT Driver implements maximum 38 channels on 4 types of S32K14X peripherals.

- 32 channels are implemented on 8 FlexTimer (FTM) modules.
- 4 channels are implemented on Low Power Interrupt Timer (LPIT).
- 1 channels are implemented on Low Power Timer (LPTMR).
- 1 channels are implemented on Real Time Clock (SRTC).

#### FTM channels

- 8 x 16-bit counters on each module FTM\_0, FTM\_1, FTM\_2, FTM\_3, FTM\_4, FTM\_5, FTM\_6, FTM\_7

#### LPIT timer channel

- Independent timeout periods for each 32-bit timer

#### LPTMR timer channel

- Independent timeout periods for each 16-bit timer

#### SRTC timer channel

- Independent timeout periods for each 32-bit timer

### 3.3 Hardware Resources

The hardware configured by the Gpt driver are FTM, LPIT, LPTMR and SRTC.

### 3.4 Deviation from Requirements

The driver deviates from the AUTOSAR GPT Driver software specification in some places.

Gpt\_PBcfg<VariantNo>.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB)

Gpt\_Cfg.c file will contain the definition for all parameters that are not variant aware

There are also some additional requirements (on top of requirements detailed in AUTOSAR GPT Driver software specification) which need to be satisfied for correct operation.

**Table 3-1. Deviations Status Column Description**

Term	Definition
N/A	Not available
N/T	Not testable
N/S	Out of scope
N/I	Not implemented
N/F	Not fully implemented

Below table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the driver.

**Table 3-2. Driver Deviations Table**

Requirement	Status	Description	Notes
SWS_Gpt_00235	N/I	SWS Item: ECUC_Gpt_00235   Container Name: GptWakeupConfiguration   Description: Function pointer to callback function (for wakeup notification).	Wrong Description: Function pointer to callback function (for non-wakeup notification). It shall relate to wakeup configuration.
SWS_Gpt_00261	N/I	Gpt_Irq.c shall include Gpt.h for the prototype declaration of the notification functions.	Gpt_Irq.c is not needed. Autosar specific interrupt behaviour is implemented using a normal function placed in the Gpt.c file.

*Table continues on the next page...*



**Table 3-2. Driver Deviations Table (continued)**

Requirement	Status	Description	Notes
SWS_Gpt_00278	N/I	Module Imported Type I Dem Dem_EventIdType Dem_EventStatusType I EcuM EcuM_WakeupSourceType I Std_Types Std_ReturnType Std_VersionInfoType	No production errors needed for current development.
SWS_Gpt_00381	N/S	These requirements are not applicable to this specification. (SRS_BSW_00344, SRS_BSW_00159, SRS_BSW_00167, SRS_BSW_00170, SRS_BSW_00398, SRS_BSW_00416, SRS_BSW_00437, SRS_BSW_00168, SRS_BSW_00168, SRS_BSW_00423, SRS_BSW_00424, SRS_BSW_00425, SRS_BSW_00426, SRS_BSW_00427, SRS_BSW_00428, SRS_BSW_00429, SRS_BSW_00432, SRS_BSW_00433, SRS_BSW_00422, SRS_BSW_00417, SRS_BSW_00161, SRS_BSW_00162, SRS_BSW_00005, SRS_BSW_00415, SRS_BSW_00325, SRS_BSW_00326, SRS_BSW_00342, SRS_BSW_00160, SRS_BSW_00007, SRS_BSW_00413, SRS_BSW_00347, SRS_BSW_00307, SRS_BSW_00373, SRS_BSW_00335, SRS_BSW_00348, SRS_BSW_00353, SRS_BSW_00361, SRS_BSW_00328, SRS_BSW_00006, SRS_BSW_00439, SRS_BSW_00357, SRS_BSW_00377, SRS_BSW_00378, SRS_BSW_00306, SRS_BSW_00308, SRS_BSW_00309, SRS_BSW_00376, SRS_BSW_00359, SRS_BSW_00360, SRS_BSW_00440, SRS_BSW_00330, SRS_BSW_00331, SRS_BSW_00009, SRS_BSW_00172, SRS_BSW_00010, SRS_BSW_00333, SRS_BSW_00321, SRS_BSW_00341, SRS_BSW_00334, SRS_SPAL_12462, SRS_SPAL_12463, SRS_SPAL_12068, SRS_SPAL_12075, SRS_SPAL_12064, SRS_SPAL_12077, SRS_SPAL_12078, SRS_SPAL_12092, SRS_SPAL_12265)	Not a requirement.

## 3.5 Driver limitations

None

## 3.6 Driver usage and configuration tips

In this chapter, the extra features from our drivers that are not described in the AutoSAR standard are detailed.

### 3.6.1 Dual Clock Feature

In order to allow dynamic change of the driver working frequency, the GPT driver has the Dual Clock Feature

The `GptEnableDualClockMode` from `GptNonAUTOSAR` should be enabled in order to have this feature active. Afterwards, the `Prescaler_Alternate` parameter allows setting a different prescaler for each channel. These parameters will be changed when calling the function call `Gpt_SetClockMode`.

`Gpt_SetClockMode` may be called only after `Gpt_Init` is called and when `GptEnableDualClockMode` is checked. Recommended usage of this API is to call it when the driver is in a lower power state but still in active use.

### 3.6.2 Input trigger for channel

The `GptEnableTriggers` from `GptNonAUTOSAR` should be enabled in order to have this feature active.

LPIT need to be configured to enable `GptLPitIsExternalTrigger` for selecting `GptLPitTriggerChannels` and other trigger feature: `GptLPitEnReloadOnTrigger`, `GptLPitEnStopOnInterrupt`, `GptLPitEnStartOnTrigger`.

In order to use this feature, MCL should be configured to select input sources and output sources for LPIT.

The feature configures the input trigger for LPIT from FTM channel of PWM module and LPIT can be triggered from the outside (TRGMUX)

Setup a PWM channel which will be used as external trigger for GPT. LPIT channel will need to start counting on a compare-match event from PWM. LPIT will trigger to PDB and PDB trigger ADC whenever compare-match from LPIT

### 3.6.3 GptStandbyWakeupSupport

The `GptStandbyWakeupSupport` from `GptNonAUTOSAR` should be enabled in order to have this feature active.

SRTC and LPTMR need to be configured to enable GptStandbyWakeupSupport for selecting wakeup feature: ON, OFF.

In order to use this feature, The driver shall NOT CLEAR the interrupt flag, the interrupt enable bit and also should not disable the counter, during init , the flag is already set..

Setup a SRTC channel and LPTMR which will be used for GPT.

### 3.6.4 GptEnableUserModeSupport

The GptEnableUserModeSupport from GptNonAUTOSAR should be enabled in order to have this feature active.

The GptEnableUserModeSupport from GptNonAUTOSAR does not support for S32K118 Derivative.

When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG\_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG\_PROT\_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM

### 3.6.5 GptChangeNextTimeoutValueApi

The GptChangeNextTimeoutValueApi from GptNonAUTOSAR should be enabled in order to have this feature active.

In order to use this feature, Enables settings for changing the channel counter compare value of a running counter

## 3.7 Runtime Errors

This driver doesn't generate any runtime error.

## 3.8 Software specification

The following sections contains driver software specifications.

### 3.8.1 Define Reference

Constants supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.3 Rev0001 .

#### 3.8.1.1 Define GPT\_VENDOR\_ID\_C

**Table 3-3. Define GPT\_VENDOR\_ID\_C Description**

<b>Name</b>	GPT_VENDOR_ID_C
<b>Initializer</b>	43

#### 3.8.1.2 Define GPT\_AR\_RELEASE\_MAJOR\_VERSION\_C

**Table 3-4. Define GPT\_AR\_RELEASE\_MAJOR\_VERSION\_C Description**

<b>Name</b>	GPT_AR_RELEASE_MAJOR_VERSION_C
<b>Initializer</b>	4

#### 3.8.1.3 Define GPT\_AR\_RELEASE\_MINOR\_VERSION\_C

**Table 3-5. Define GPT\_AR\_RELEASE\_MINOR\_VERSION\_C Description**

<b>Name</b>	GPT_AR_RELEASE_MINOR_VERSION_C
<b>Initializer</b>	3

#### 3.8.1.4 Define GPT\_AR\_RELEASE\_REVISION\_VERSION\_C

**Table 3-6. Define GPT\_AR\_RELEASE\_REVISION\_VERSION\_C Description**

<b>Name</b>	GPT_AR_RELEASE_REVISION_VERSION_C
<b>Initializer</b>	1

### 3.8.1.5 Define GPT\_SW\_MAJOR\_VERSION\_C

**Table 3-7. Define GPT\_SW\_MAJOR\_VERSION\_C  
Description**

<b>Name</b>	GPT_SW_MAJOR_VERSION_C
<b>Initializer</b>	1

### 3.8.1.6 Define GPT\_SW\_MINOR\_VERSION\_C

**Table 3-8. Define GPT\_SW\_MINOR\_VERSION\_C  
Description**

<b>Name</b>	GPT_SW_MINOR_VERSION_C
<b>Initializer</b>	0

### 3.8.1.7 Define GPT\_SW\_PATCH\_VERSION\_C

**Table 3-9. Define GPT\_SW\_PATCH\_VERSION\_C  
Description**

<b>Name</b>	GPT_SW_PATCH_VERSION_C
<b>Initializer</b>	1

### 3.8.1.8 Define GPT\_E\_ALREADY\_INITIALIZED

initialization called when already initialized

#### Details:

Errors and exceptions that will be detected by the GPT driver

**Implements:** Gpt\_Det\_ErrorCodes\_define

**Table 3-10. Define GPT\_E\_ALREADY\_INITIALIZED  
Description**

<b>Name</b>	GPT_E_ALREADY_INITIALIZED
<b>Initializer</b>	(uint8)0x0DU

### 3.8.1.9 Define GPT\_E\_BUSY

function called when timer channel is still running

#### Details:

Errors and exceptions that will be detected by the GPT driver

**Implements:** Gpt\_Det\_ErrorCodes\_define

**Table 3-11. Define GPT\_E\_BUSY Description**

<b>Name</b>	GPT_E_BUSY
<b>Initializer</b>	(uint8)0x0BU

### 3.8.1.10 Define GPT\_E\_INVALID\_CALL

function Gpt\_StartTimer is called when the driver is in sleep mode for a channel which is not wakeup enabled

#### Details:

Errors and exceptions that will be detected by the GPT driver

**Implements:** Gpt\_Det\_ErrorCodes\_define

**Table 3-12. Define GPT\_E\_INVALID\_CALL Description**

<b>Name</b>	GPT_E_INVALID_CALL
<b>Initializer</b>	(uint8)0xA0U

### 3.8.1.11 Define GPT\_E\_MODE

function called with invalid the parameter in function Gpt\_GetPredefTimerValue

#### Details:

Errors and exceptions that will be detected by the GPT driver

**Implements:** Gpt\_Det\_ErrorCodes\_define

**Table 3-13. Define GPT\_E\_MODE Description**

<b>Name</b>	GPT_E_MODE
<b>Initializer</b>	((uint8)0x0C)

### 3.8.1.12 Define GPT\_E\_PARAM\_CHANNEL

function called for invalid channel

**Details:**

Errors and exceptions that will be detected by the GPT driver

**Implements:** Gpt\_Det\_ErrorCodes\_define

**Table 3-14. Define GPT\_E\_PARAM\_CHANNEL Description**

<b>Name</b>	GPT_E_PARAM_CHANNEL
<b>Initializer</b>	(uint8)0x14U

### 3.8.1.13 Define GPT\_E\_PARAM\_MODE

function called with invalid mode param.

**Details:**

Errors and exceptions that will be detected by the GPT driver

**Implements:** Gpt\_Det\_ErrorCodes\_define

**Table 3-15. Define GPT\_E\_PARAM\_MODE Description**

<b>Name</b>	GPT_E_PARAM_MODE
<b>Initializer</b>	(uint8)0x1FU

### 3.8.1.14 Define GPT\_E\_PARAM\_POINTER

function called with NULL pointer

#### Details:

Errors and exceptions that will be detected by the GPT driver

**Implements:** Gpt\_Det\_ErrorCodes\_define

**Table 3-16. Define GPT\_E\_PARAM\_POINTER Description**

<b>Name</b>	GPT_E_PARAM_POINTER
<b>Initializer</b>	(uint8)0x16U

### 3.8.1.15 Define GPT\_E\_PARAM\_VALUE

function called with parameter value out of range

#### Details:

Errors and exceptions that will be detected by the GPT driver

**Implements:** Gpt\_Det\_ErrorCodes\_define

**Table 3-17. Define GPT\_E\_PARAM\_VALUE Description**

<b>Name</b>	GPT_E_PARAM_VALUE
<b>Initializer</b>	(uint8)0x15U

### 3.8.1.16 Define GPT\_E\_PARAM\_PREDEF\_TIMER

function called with invalid the parameter in function Gpt\_GetPredefTimerValue

#### Details:

Errors and exceptions that will be detected by the GPT driver

**Implements:** Gpt\_Det\_ErrorCodes\_define



**Table 3-18. Define GPT\_E\_PARAM\_PREDEF\_TIMER**  
Description

<b>Name</b>	GPT_E_PARAM_PREDEF_TIMER
<b>Initializer</b>	((uint8)0x17)

### 3.8.1.17 Define GPT\_E\_UNINIT

function called without module initialization

#### Details:

Errors and exceptions that will be detected by the GPT driver

**Implements:** Gpt\_Det\_ErrorCodes\_define

**Table 3-19. Define GPT\_E\_UNINIT Description**

<b>Name</b>	GPT_E_UNINIT
<b>Initializer</b>	(uint8)0x0AU

### 3.8.1.18 Define GPT\_CHECKWAKEUP\_ID

API service ID for Gpt\_CheckWakeup function.

#### Details:

Parameters used when raising an error/exception

**Table 3-20. Define GPT\_CHECKWAKEUP\_ID Description**

<b>Name</b>	GPT_CHECKWAKEUP_ID
<b>Initializer</b>	(uint8)0x0CU

### 3.8.1.19 Define GPT\_DEINIT\_ID

API service ID for Gpt\_DeInit function.

#### Details:

Parameters used when raising an error/exception

**Table 3-21. Define GPT\_DEINIT\_ID Description**

<b>Name</b>	GPT_DEINIT_ID
<b>Initializer</b>	(uint8)0x02U

### 3.8.1.20 Define GPT\_DISABLENOTIFICATION\_ID

API service ID for Gpt\_DisableNotification function.

#### Details:

Parameters used when raising an error/exception

**Table 3-22. Define GPT\_DISABLENOTIFICATION\_ID Description**

<b>Name</b>	GPT_DISABLENOTIFICATION_ID
<b>Initializer</b>	(uint8)0x08U

### 3.8.1.21 Define GPT\_DISABLEWAKEUP\_ID

API service ID for Gpt\_DisableWakeup function.

#### Details:

Parameters used when raising an error/exception

**Table 3-23. Define GPT\_DISABLEWAKEUP\_ID Description**

<b>Name</b>	GPT_DISABLEWAKEUP_ID
<b>Initializer</b>	(uint8)0x0AU

### 3.8.1.22 Define GPT\_ENABLENOTIFICATION\_ID

API service ID for Gpt\_EnableNotification function.

**Details:**

Parameters used when raising an error/exception

**Table 3-24. Define GPT\_ENABLENOTIFICATION\_ID Description**

<b>Name</b>	GPT_ENABLENOTIFICATION_ID
<b>Initializer</b>	(uint8)0x07U

### 3.8.1.23 Define GPT\_ENABLEWAKEUP\_ID

API service ID for Gpt\_EnableWakeup function.

**Details:**

Parameters used when raising an error/exception

**Table 3-25. Define GPT\_ENABLEWAKEUP\_ID Description**

<b>Name</b>	GPT_ENABLEWAKEUP_ID
<b>Initializer</b>	(uint8)0x0BU

### 3.8.1.24 Define GPT\_GETVERSIONINFO\_ID

API service ID for Gpt\_GetVersionInfo function.

**Details:**

Parameters used when raising an error/exception

**Table 3-26. Define GPT\_GETVERSIONINFO\_ID Description**

<b>Name</b>	GPT_GETVERSIONINFO_ID
<b>Initializer</b>	(uint8)0x00U

### 3.8.1.25 Define GPT\_INIT\_ID

API service ID for Gpt\_Init function.

#### Details:

Parameters used when raising an error/exception

**Table 3-27. Define GPT\_INIT\_ID Description**

<b>Name</b>	GPT_INIT_ID
<b>Initializer</b>	(uint8)0x01U

### 3.8.1.26 Define GPT\_INSTANCE\_ID

Instance ID of this gpt driver.

**Table 3-28. Define GPT\_INSTANCE\_ID Description**

<b>Name</b>	GPT_INSTANCE_ID
<b>Initializer</b>	(uint8)0U

### 3.8.1.27 Define GPT\_MODULE\_ID

**Table 3-29. Define GPT\_MODULE\_ID Description**

<b>Name</b>	GPT_MODULE_ID
<b>Initializer</b>	100

### 3.8.1.28 Define GPT\_PROCESSCOMMONINTERRUPT\_ID

API service ID for Gpt\_ConfigurationLock function.

#### Details:

Parameters used when raising an error/exception

**Table 3-30. Define GPT\_PROCESSCOMMONINTERRUPT\_ID Description**

<b>Name</b>	GPT_PROCESSCOMMONINTERRUPT_ID
<b>Initializer</b>	(uint8)0x0DU

### 3.8.1.29 Define GPT\_SETMODE\_ID

API service ID for Gpt\_SetMode function.

#### Details:

Parameters used when raising an error/exception

**Table 3-31. Define GPT\_SETMODE\_ID Description**

<b>Name</b>	GPT_SETMODE_ID
<b>Initializer</b>	(uint8)0x09U

### 3.8.1.30 Define GPT\_SET\_CLOCK\_MODE\_ID

GPT\_SET\_CLOCK\_MODE\_ID switch.

**Table 3-32. Define GPT\_SET\_CLOCK\_MODE\_ID Description**

<b>Name</b>	GPT_SET_CLOCK_MODE_ID
<b>Initializer</b>	(uint8)0x10U

### 3.8.1.31 Define GPT\_STARTTIMER\_ID

API service ID for Gpt\_StartTimer function.

#### Details:

Parameters used when raising an error/exception

**Table 3-33. Define GPT\_STARTTIMER\_ID Description**

<b>Name</b>	GPT_STARTTIMER_ID
<b>Initializer</b>	(uint8)0x05U

### 3.8.1.32 Define GPT\_STOPTIMER\_ID

API service ID for Gpt\_StopTimer function.

#### Details:

Parameters used when raising an error/exception

**Table 3-34. Define GPT\_STOPTIMER\_ID Description**

<b>Name</b>	GPT_STOPTIMER_ID
<b>Initializer</b>	(uint8)0x06U

### 3.8.1.33 Define GPT\_TIMEELAPSED\_ID

API service ID for Gpt\_GetTimeElapsed function.

#### Details:

Parameters used when raising an error/exception

**Table 3-35. Define GPT\_TIMEELAPSED\_ID Description**

<b>Name</b>	GPT_TIMEELAPSED_ID
<b>Initializer</b>	(uint8)0x03U

### 3.8.1.34 Define GPT\_TIMEREMAINING\_ID

API service ID for Gpt\_GetTimeRemaining function.

#### Details:

Parameters used when raising an error/exception

**Table 3-36. Define GPT\_TIMEREMAINING\_ID Description**

<b>Name</b>	GPT_TIMEREMAINING_ID
<b>Initializer</b>	(uint8)0x04U

### 3.8.1.35 Define GPT\_CHANGE\_NEXT\_TIMEOUT\_VALUE\_ID

API service ID for GPT\_CHANGE\_NEXT\_TIMEOUT\_VALUE\_ID function.

#### Details:

Parameters used when raising an error/exception

**Table 3-37. Define GPT\_CHANGE\_NEXT\_TIMEOUT\_VALUE\_ID Description**

<b>Name</b>	GPT_CHANGE_NEXT_TIMEOUT_VALUE_ID
<b>Initializer</b>	(uint8)0x0FU

### 3.8.1.36 Define GPT\_DEV\_ERROR\_DETECT

Gpt Dev error detect switch.

**Table 3-38. Define GPT\_DEV\_ERROR\_DETECT Description**

<b>Name</b>	GPT_DEV_ERROR_DETECT
<b>Initializer</b>	(STD_ON)

### 3.8.1.37 Define GPT\_PRECOMPILE\_SUPPORT

**Table 3-39. Define GPT\_PRECOMPILE\_SUPPORT Description**

<b>Name</b>	GPT_PRECOMPILE_SUPPORT
<b>Initializer</b>	(STD_OFF)

### 3.8.1.38 Define GPT\_REPORT\_WAKEUP\_SOURCE

Report Wakeup Source switch.

**Table 3-40. Define GPT\_REPORT\_WAKEUP\_SOURCE Description**

<b>Name</b>	GPT_REPORT_WAKEUP_SOURCE
<b>Initializer</b>	(STD_ON)

### 3.8.1.39 Define GPT\_DEINIT\_API

GPT\_DEINIT\_API switch.

**Table 3-41. Define GPT\_DEINIT\_API Description**

<b>Name</b>	GPT_DEINIT_API
<b>Initializer</b>	(STD_ON)

### 3.8.1.40 Define GPT\_ENABLE\_DISABLE\_NOTIFICATION\_API

GPT\_ENABLE\_DISABLE\_NOTIFICATION\_API switch.

**Table 3-42. Define GPT\_ENABLE\_DISABLE\_NOTIFICATION\_API Description**

<b>Name</b>	GPT_ENABLE_DISABLE_NOTIFICATION_API
<b>Initializer</b>	(STD_ON)

### 3.8.1.41 Define GPT\_PREDEFTIMER\_FUNCTIONALITY\_API

GPT\_PREDEFTIMER\_FUNCTIONALITY\_API switch.

**Table 3-43. Define GPT\_PREDEFTIMER\_FUNCTIONALITY\_API Description**

<b>Name</b>	GPT_PREDEFTIMER_FUNCTIONALITY_API
<b>Initializer</b>	([!IF "GptConfigurationOfOptApiServices/ GptPredefTimerFunctionalityApi"!])STD_ON[!ELSE!]STD_OFF[!ENDIF!])



### 3.8.1.42 Define GPT\_TIME\_ELAPSED\_API

GPT\_TIME\_ELAPSED\_API switch.

**Table 3-44. Define GPT\_TIME\_ELAPSED\_API Description**

<b>Name</b>	GPT_TIME_ELAPSED_API
<b>Initializer</b>	(STD_ON)

### 3.8.1.43 Define GPT\_TIME\_REMAINING\_API

GPT\_TIME\_REMAINING\_API switch.

**Table 3-45. Define GPT\_TIME\_REMAINING\_API Description**

<b>Name</b>	GPT_TIME_REMAINING_API
<b>Initializer</b>	(STD_ON)

### 3.8.1.44 Define GPT\_USER\_MODE\_SOFT\_LOCKING

GPT\_USER\_MODE\_SOFT\_LOCKING switch.

**Table 3-46. Define GPT\_USER\_MODE\_SOFT\_LOCKING Description**

<b>Name</b>	GPT_USER_MODE_SOFT_LOCKING
<b>Initializer</b>	(STD_OFF)

### 3.8.1.45 Define GPT\_VERSION\_INFO\_API

GPT\_VERSION\_INFO\_API switch.

**Table 3-47. Define GPT\_VERSION\_INFO\_API Description**

<b>Name</b>	GPT_VERSION_INFO_API
<b>Initializer</b>	(STD_ON)

### 3.8.1.46 Define GPT\_WAKEUP\_FUNCTIONALITY\_API

GPT\_WAKEUP\_FUNCTIONALITY\_API switch.

**Table 3-48. Define GPT\_WAKEUP\_FUNCTIONALITY\_API Description**

<b>Name</b>	GPT_WAKEUP_FUNCTIONALITY_API
<b>Initializer</b>	(STD_ON)

### 3.8.1.47 Define GPT\_CHANGE\_NEXT\_TIMEOUT\_VALUE

GPT\_CHANGE\_NEXT\_TIMEOUT\_VALUE switch.

**Table 3-49. Define GPT\_CHANGE\_NEXT\_TIMEOUT\_VALUE Description**

<b>Name</b>	GPT_CHANGE_NEXT_TIMEOUT_VALUE
<b>Initializer</b>	(STD_ON)

## 3.8.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.3 Rev0001 .

### 3.8.2.1 Enumeration Gpt\_ChannelModeType

Gpt channel mode type. Indicates of whether the channel mode is "CONTINUOUS" or "ONE SHOT".

**Table 3-50. Enumeration Gpt\_ChannelModeType Values**

<b>Name</b>	<b>Initializer</b>	<b>Description</b>
GPT_CH_MODE_CONTINUOUS	0U	GPT channel mode - continous mode.
GPT_CH_MODE_ONESHOT	1U	GPT channel mode - one-shot mode.

### 3.8.2.2 Enumeration Gpt\_ChannelStatusType

Gpt channel status type. This enumerated type allows the selection of channel status type.

**Table 3-51. Enumeration Gpt\_ChannelStatusType Values**

Name	Initializer	Description
GPT_STATUS_UNINITIALIZED	0U	GPT channel was uninitialized.
GPT_STATUS_INITIALIZED	1U	GPT channel was initialized.
GPT_STATUS_STOPPED	2U	GPT channel was stopped.
GPT_STATUS_EXPIRED	3U	GPT channel was expired.
GPT_STATUS_RUNNING	4U	GPT channel is running.

### 3.8.2.3 Enumeration Gpt\_ClockModeType

This enumerated type allows the selection of different select clock modes.

**Table 3-52. Enumeration Gpt\_ClockModeType Values**

Name	Initializer	Description
GPT_CLOCKMODE_NORMAL	0	GPT Normal clock mode of the GPT.
GPT_CLOCKMODE_ALTERNATE	1	GPT Alternate clock mode.

### 3.8.2.4 Enumeration Gpt\_ModeType

This enumerated type allows the selection of different power modes.

**Implements:** Gpt\_ModeType\_enumeration

**Table 3-53. Enumeration Gpt\_ModeType Values**

Name	Initializer	Description
GPT_MODE_NORMAL	0	GPT Normal operation mode of the GPT.
GPT_MODE_SLEEP	1	GPT sleep mode.

### 3.8.2.5 Enumeration Gpt\_PreddefTimerType

Predf Timer type. Indicates the type of preddef timer.

**Implements:** Gpt\_PredefTimerType\_enumeration**Table 3-54. Enumeration Gpt\_PredefTimerType Values**

Name	Initializer	Description
GPT_PREDEF_TIMER_1US_16BIT	0U	16bit timer: To support 16bit hardware timers.
GPT_PREDEF_TIMER_1US_24BIT	1U	24bit timer: To support 24bit hardware timers.
GPT_PREDEF_TIMER_1US_32BIT	2U	32bit timer: To support 32bit hardware timers.
GPT_PREDEF_TIMER_100US_32BIT	3U	100µs32bit timer: covers automotive use cases (time span 4.9 days).

### 3.8.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.3 Rev0001 .

#### 3.8.3.1 Function Gpt\_CheckWakeup

GPT driver function for checking if a wakeup capable GPT channel is the source for a wakeup event.

**Details:**

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. The driver needs to be initialized before calling `Gpt_CheckWakeup()`. Otherwise, the function Gpt\_CheckWakeup shall raise the development error GPT\_E\_UNINIT.

**Implements:** Gpt\_CheckWakeup\_Activity

**Prototype:** void Gpt\_CheckWakeup(EcuM\_WakeupSourceType wakeupSource);

**Table 3-55. Gpt\_CheckWakeup Arguments**

Type	Name	Direction	Description
EcuM_WakeupSourceType	wakeupSource	input	Wakeup source.

### 3.8.3.2 Function Gpt\_DeInit

GPT driver de-initialization function.

**Details:**

Service for deinitializing all hardware timer channels to their power on reset state. The state of the peripheral after DeInit shall be the same as after power on reset. The service influences only the peripherals, which are allocated by static configuration and the runtime configuration set passed by the previous call of `Gpt_Init()`. The driver needs to be initialized before calling `Gpt_DeInit()`. Otherwise, the function `Gpt_DeInit` shall raise the development error `GPT_E_UNINIT` and leave the desired deinitialization functionality without any action.

**Implements:** `Gpt_DeInit_Activity`

**Prototype:** `void Gpt_DeInit(void);`

### 3.8.3.3 Function Gpt\_DisableNotification

GPT driver function for disabling the notification for a timer channel.

**Details:**

Service for disabling the notification for a channel during runtime. This function can be called, while the timer is already running. When disabled, no notification will be sent. When re-enabled again, the user will not be notified of events, occurred while notifications have been disabled. Usage of re-entrant capability is only allowed if the callers take care that there is no simultaneous usage of the same channel. The driver needs to be initialized before calling `Gpt_DisableNotification()`. Otherwise, the function shall raise the development error `GPT_E_UNINIT`.

**Implements:** `Gpt_DisableNotification_Activity`

**Prototype:** `void Gpt_DisableNotification(Gpt_ChannelType channel);`

**Table 3-56. Gpt\_DisableNotification Arguments**

Type	Name	Direction	Description
<code>Gpt_ChannelType</code>	<code>channel</code>	input	Channel id.

### 3.8.3.4 Function Gpt\_DisableWakeup

GPT driver function for disabling the wakeup interrupt invocation for a timer channel.

#### Details:

This service shall disable the wakeup interrupt invocation of a single GPT channel. Usage of re-entrant capability is only allowed if the callers take care that there is no simultaneous usage of the same channel. The driver needs to be initialized before calling `Gpt_DisableWakeup()`. Otherwise, the function `Gpt_DisableWakeup` shall raise the development error `GPT_E_UNINIT`.

**Implements:** `Gpt_DisableWakeup_Activity`

**Prototype:** `void Gpt_DisableWakeup(Gpt_ChannelType channel);`

**Table 3-57. Gpt\_DisableWakeup Arguments**

Type	Name	Direction	Description
Gpt_ChannelType	channel	input	Channel id.

### 3.8.3.5 Function Gpt\_EnableNotification

GPT driver function for enabling the notification for a timer channel.

#### Details:

Service for enabling the notification for a channel during runtime. This function can be called, while the timer is already running. Usage of re-entrant capability is only allowed if the callers take care that there is no simultaneous usage of the same channel. The driver needs to be initialized before calling `Gpt_EnableNotification()`. Otherwise, the function `Gpt_EnableNotification` shall raise the development error `GPT_E_UNINIT`.

**Implements:** `Gpt_EnableNotification_Activity`

**Prototype:** `void Gpt_EnableNotification(Gpt_ChannelType channel);`

**Table 3-58. Gpt\_EnableNotification Arguments**

Type	Name	Direction	Description
Gpt_ChannelType	channel	input	Channel id.

### 3.8.3.6 Function Gpt\_EnableWakeup

GPT driver function for enabling the wakeup interrupt invocation for a timer channel.

#### Details:

This service shall re-enable the wakeup interrupt invocation of a single GPT channel. If supported by hardware and enabled, an internal hardware timer can serve as a wakeup source. Usage of re-entrant capability is only allowed if the callers take care that there is no simultaneous usage of the same channel.

**Implements:** Gpt\_EnableWakeup\_Activity

**Prototype:** void Gpt\_EnableWakeup(Gpt\_ChannelType channel);

**Table 3-59. Gpt\_EnableWakeup Arguments**

Type	Name	Direction	Description
Gpt_ChannelType	channel	input	Channel id.

### 3.8.3.7 Function Gpt\_GetPredefTimerValue

This function provides the current value of the given predefined free-running timer.

#### Details:

This function provides the current value of the given predefined free-running timer.

**Implements:** Gpt\_GetPredefTimerValue\_Activity

**Prototype:** Std\_ReturnType Gpt\_GetPredefTimerValue(Gpt\_PredefTimerType PredefTimer, uint32\* TimeValuePtr);

**Table 3-60. Gpt\_GetPredefTimerValue Arguments**

Type	Name	Direction	Description
Gpt_PredefTimerType	PredefTimer	input	Predefined Timer Type
uint32*	TimeValuePtr	output	Pointer to the output time value.
Std_ReturnType	Return value	output	Return E_OK if no errors. Otherwise return E_NOT_OK.

### 3.8.3.8 Function Gpt\_GetTimeElapsed

GPT driver function for fetching the elapsed timer value.

#### Details:

Service for querying the time already elapsed. In one shot mode, this is the value relative to the point in time, the channel has been started with Gpt\_StartTimer (calculated by the normal operation function by subtracting the current minus the initial timer value and returning the absolute value). In continuous mode, the function returns the timer value relative to the last timeout or the start of the channel. All time units used within the API services of the GPT driver shall be of the unit ticks. Usage of re-entrant capability is only allowed if the callers take care that there is no simultaneous usage of the same channel. To get times out of register values it is necessary to know the oscillator frequency, pre prescalers and so on. Since these settings are made in MCU and(or) in other modules it is not possible to calculate such times. Hence the conversions between time and ticks shall be part of an upper layer. The driver needs to be initialized before calling Gpt\_GetTimeElapsed(). Otherwise, the function shall raise the development error GPT\_E\_UNINIT and return 0.

**Return:** Gpt\_ValueType - Elapsed Time in number of ticks.

**Implements:** Gpt\_GetTimeElapsed\_Activity

**Prototype:** Gpt\_ValueType Gpt\_GetTimeElapsed(Gpt\_ChannelType channel);

**Table 3-61. Gpt\_GetTimeElapsed Arguments**

Type	Name	Direction	Description
Gpt_ChannelType	channel	input	- channel id.

### 3.8.3.9 Function Gpt\_GetTimeRemaining

GPT driver function for fetching the remaining timer value.

#### Details:

This function returns the timer value remaining until the next timeout period will expire (calculated by the normal operation function by subtracting the timeout minus the current timer value and returning the absolute value) All time units used within the API services



of the GPT driver shall be of the unit ticks. Usage of re-entrant capability is only allowed if the callers take care that there is no simultaneous usage of the same channel. To get times out of register values it is necessary to know the oscillator frequency, prescalers and so on. Since these settings are made in MCU and(or) in other modules it is not possible to calculate such times. Hence the conversions between time and ticks shall be part of an upper layer. The driver needs to be initialized before calling `Gpt_GetTimeRemaining()`. Otherwise, the function shall raise the development error `GPT_E_UNINIT` and return 0.

**Return:** `Gpt_ValueType` - Remaining Time in number of ticks.

**Implements:** `Gpt_GetTimeRemaining_Activity`

**Prototype:** `Gpt_ValueType Gpt_GetTimeRemaining(Gpt_ChannelType channel);`

**Table 3-62. Gpt\_GetTimeRemaining Arguments**

Type	Name	Direction	Description
<code>Gpt_ChannelType</code>	<code>channel</code>	input	- channel id.

### 3.8.3.10 Function Gpt\_GetVersionInfo

This function returns the version information of this module.

**Details:**

This service returns the version information of this module. The version information includes: Module Id Vendor Id Vendor specific version numbers

**Implements:** `Gpt_GetVersionInfo_Activity`

**Prototype:** `void Gpt_GetVersionInfo(Std_VersionInfoType *versioninfo);`

**Table 3-63. Gpt\_GetVersionInfo Arguments**

Type	Name	Direction	Description
<code>Std_VersionInfoType *</code>	<code>versioninfo</code>	output	- pointer to location to store version info.

### 3.8.3.11 Function Gpt\_Init

GPT driver initialization function.

**Details:**

Service for driver initialization. The Initialization function shall initialize all relevant registers of the configured hardware with the values of the structure referenced by the parameter ConfigPtr. All time units used within the API services of the GPT driver shall be of the unit ticks. This function shall only initialize the configured resources. Resources that are not configured in the configuration file shall not be touched. The following rules regarding initialization of controller registers shall apply to the GPT Driver implementation: [1] If the hardware allows for only one usage of the register, the driver module implementing that functionality is responsible for initializing the register [2] If the register can affect several hardware modules and if it is an IO register it shall be initialized by the PORT driver [3] If the register can affect several hardware modules and if it is not an IO register it shall be initialized by the MCU driver [4] One-time writable registers that require initialization directly after reset shall be initialized by the startup code [5] All other registers shall be initialized by the startup code

**Implements:** Gpt\_Init\_Activity

**Prototype:** void Gpt\_Init(const Gpt\_ConfigType \*configPtr);

**Table 3-64. Gpt\_Init Arguments**

Type	Name	Direction	Description
const Gpt_ConfigType*	configPtr	input	Pointer to a selected configuration structure.

### 3.8.3.12 Function Gpt\_SetClockMode

This function changes the channel prescaler.

**Details:**

This function sets all channels prescalers based on the input mode.

**Implements:** Gpt\_SetClockMode\_Activity

**Prototype:** void Gpt\_SetClockMode(Gpt\_ClockModeType clkMode);

**Table 3-65. Gpt\_SetClockMode Arguments**

Type	Name	Direction	Description
Gpt_ClockModeType	clkMode	input	- prescaler

### 3.8.3.13 Function Gpt\_SetMode

GPT driver function for setting the operation mode.

#### Details:

Service for GPT mode selection. This service shall set the operation mode to the given mode parameter . When sleep mode is requested, the ECU State Manager calls Gpt\_SetMode with mode parameter "GPT\_MODE\_SLEEP" and prepares the GPT for sleep mode. The MCU Driver is then putting the controller into SLEEP mode The driver needs to be initialized before calling `Gpt_SetMode()`. Otherwise, the function Gpt\_SetMode shall raise the development error GPT\_E\_UNINIT.

**Implements:** Gpt\_SetMode\_Activity

**Prototype:** `void Gpt_SetMode(Gpt_ModeType mode);`

**Table 3-66. Gpt\_SetMode Arguments**

Type	Name	Direction	Description
Gpt_ModeType	mode	input	Operation mode.

### 3.8.3.14 Function Gpt\_StartTimer

GPT driver function for starting a timer channel.

#### Details:

The function Gpt\_StartTimer shall start the selected timer channel with a defined timeout period. The function Gpt\_StartTimer shall invoke the configured notification for that channel (see also SWS\_Gpt\_00292) after the timeout period referenced via the parameter value (if enabled). All time units used within the API services of the GPT driver shall be of the unit ticks. In production mode no error is generated. The rational is that it adds no additional functionality to the driver. In this case the timer will be restarted with the timeout value, given as a parameter to the service. Usage of re-entrant capability is only allowed if the callers take care that there is no simultaneous usage of the same channel. To get times out of register values it is necessary to know the oscillator frequency, prescalers and so on. Since these settings are made in MCU and(or) in other modules it is

not possible to calculate such times. Hence the conversions between time and ticks shall be part of an upper layer. The driver needs to be initialized before calling `Gpt_StartTimer()`. Otherwise, the function `Gpt_StartTimer` shall raise the development error `GPT_E_UNINIT`.

**Implements:** `Gpt_StartTimer_Activity`

**Prototype:** `void Gpt_StartTimer(Gpt_ChannelType channel, Gpt_ValueType value);`

**Table 3-67. Gpt\_StartTimer Arguments**

Type	Name	Direction	Description
<code>Gpt_ChannelType</code>	<code>channel</code>	input	Channel id.
<code>Gpt_ValueType</code>	<code>value</code>	input	Timeout period (in number of ticks) after a notification or a wakeup event shall occur.

### 3.8.3.15 Function Gpt\_StopTimer

GPT driver function for stopping a timer channel.

**Details:**

Service for stopping the selected timer channel Stopping a timer channel, not been started before will not return a development error Timer channels configured in one shot mode are stopped automatically, when the timeout period has expired. Usage of re-entrant capability is only allowed if the callers take care that there is no simultaneous usage of the same channel. The driver needs to be initialized before calling `Gpt_StopTimer()`. Otherwise, the function shall raise the development error `GPT_E_UNINIT`.

**Implements:** `Gpt_StopTimer_Activity`

**Prototype:** `void Gpt_StopTimer(Gpt_ChannelType channel);`

**Table 3-68. Gpt\_StopTimer Arguments**

Type	Name	Direction	Description
<code>Gpt_ChannelType</code>	<code>channel</code>	input	Channel id.

### 3.8.3.16 Function Gpt\_ChangeNextTimeoutValue

The function changes the timeout period value of the requested running channel.

## Details:

The function changes the timeout period (in number of ticks) of a running channel. The new target value will be used after the current compare matching. The user shall call this function only for eTimer channels because other modules does not offer hardware support. This is a non-Autosar function.

**Implements:** Gpt\_ChangeNextTimeoutValue\_Activity

**Prototype:** void Gpt\_ChangeNextTimeoutValue(Gpt\_ChannelType channel, Gpt\_ValueType value);

**Table 3-69. Gpt\_ChangeNextTimeoutValue Arguments**

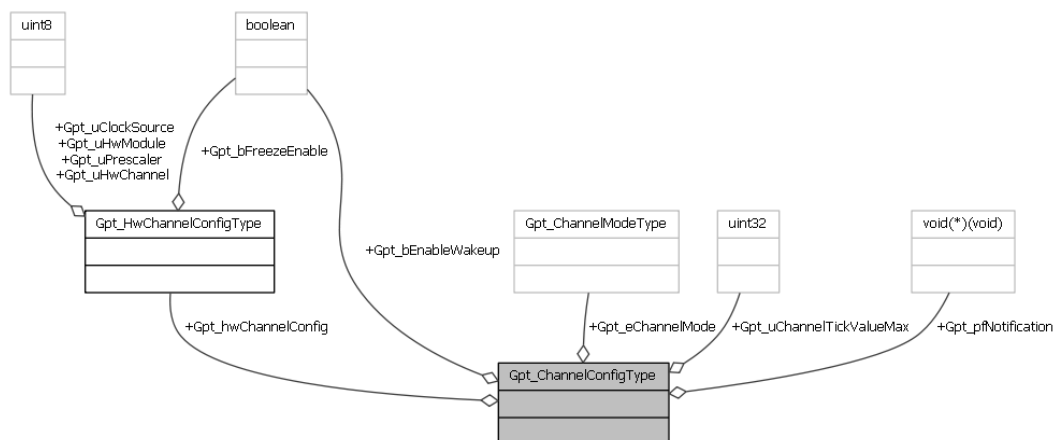
Type	Name	Direction	Description
Gpt_ChannelType	channel	input	Channel id.
Gpt_ValueType	value	input	New timeout period (in number of ticks) after a notification or a wakeup event shall occur.

## 3.8.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.3 Rev0001 .

### 3.8.4.1 Structure Gpt\_ChannelConfigType

Gpt channel configuration type.



**Figure 3-1. Struct Gpt\_ChannelConfigType**

**Declaration:**

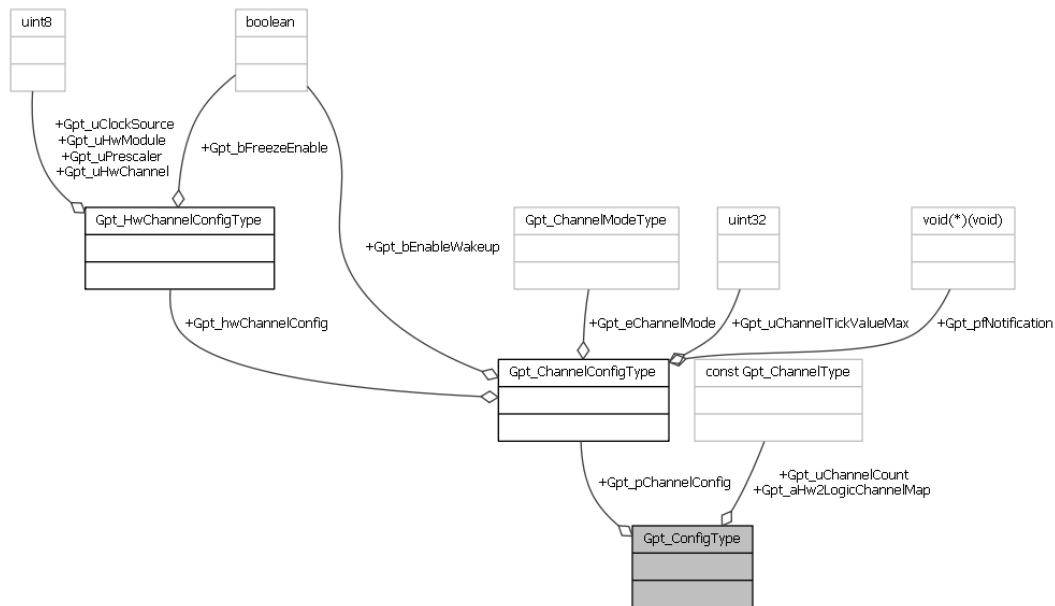
```
typedef struct
{
    boolean Gpt_bEnableWakeup;
    Gpt_NotificationType Gpt_pfNotification;
    EcuM_WakeupSourceType Gpt_uWakeupSource;
    Gpt_ValueType Gpt_uChannelTickValueMax;
    Gpt_ChannelModeType Gpt_eChannelMode;
    Gpt_HwChannelConfigType Gpt_hwChannelConfig;
} Gpt_ChannelConfigType;
```

**Table 3-70. Structure Gpt\_ChannelConfigType member description**

Member	Description
Gpt_bEnableWakeup	GPT ch WakeUp enable.
Gpt_eChannelMode	GPT channel mode.
Gpt_hwChannelConfig	Hardware dependent channel configuration.
Gpt_pfNotification	Pointer to external callback.
Gpt_uChannelTickValueMax	Channel max tick value.
Gpt_uWakeupSource	EcuM wake up source Id.

**3.8.4.2 Structure Gpt\_ConfigType**

Gpt configuration type. This is the type of the data structure including the configuration set required for initializing the GPT driver.

**Figure 3-2. Struct Gpt\_ConfigType**

**Implements:** Gpt\_ConfigType\_structure**Declaration:**

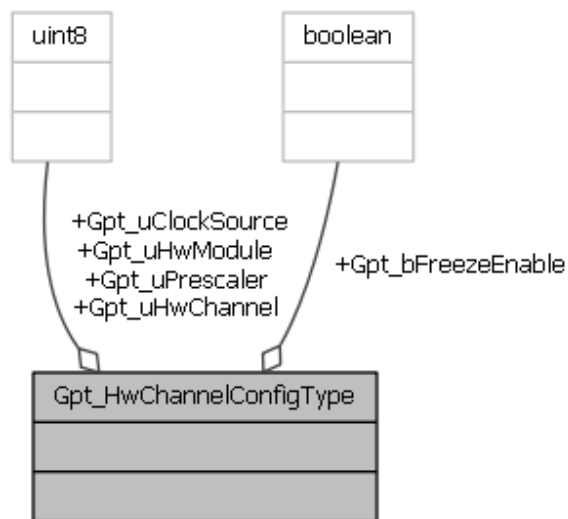
```
typedef struct
{
    const Gpt_ChannelType Gpt_uChannelCount;
    const Gpt_ChannelConfigType(* Gpt_pChannelConfig) [];
    #if(GPT_PREDEFTIMER_FUNCTIONALITY_API == STD_ON)
        const Gpt_HwPredefChannelConfigType (**Gpt_pChannelPredefConfigType);
    #endif
    const Gpt_ChannelType    Gpt_aHw2LogicChannelMap[GPT_CHANNEL_IDX_NUM];
} Gpt_ConfigType;
```

**Table 3-71. Structure Gpt\_ConfigType member description**

Member	Description
Gpt_uChannelCount	Number of GPT channels (configured in tresos plugin builder).
Gpt_pChannelConfig	Pointer to the GPT channel configuration.
Gpt_pChannelPredefConfigType	Pointer to the GPT channel predef timer configuration.
Gpt_aHw2LogicChannelMap	Hardware to logic channel map table.

**3.8.4.3 Structure Gpt\_HwChannelConfigType**

Low level configuration for channel specific parameters.

**Figure 3-3. Struct Gpt\_HwChannelConfigType****Declaration:**

```
typedef struct
{
    uint8 Gpt_uHwChannel,
    uint8 Gpt_uHwModule,
    boolean Gpt_bFreezeEnable,
    Gpt_ClockSourceType Gpt_uClockSource,
```

```

    Gpt_PrescalerType Gpt_uFtmPrescaler,
    Gpt_PrescalerType Gpt_uLptmrPrescaler,
    Gpt_ClockSourceType Gpt_uLptmrClock,
    #if (GPT_SET_CLOCK_MODE == STD_ON)
    Gpt_PrescalerType Gpt_uFtmPrescaler_Alternate,
    Gpt_PrescalerType Gpt_uLptmrPrescaler_Alternate
    #endif /* GPT_DUAL_CLOCK_MODE */
    Gpt_ClockSourceType Gpt_uSRtcClock;
    #if (GPT_LPIT_ENABLE_EXT_TRIGGERS==STD_ON)
    uint32 Gpt_uTimerTriggerConfig;
    #endif
} Gpt_HwChannelConfigType;

```

**Table 3-72. Structure Gpt\_HwChannelConfigType member description**

Member	Description
Gpt_uHwChannel	GPT hw channel ID.
Gpt_uHwModule	GPT hw module used.
Gpt_bFreezeEnable	Channel freeze enable.
Gpt_uClockSource	FTM channel clock source.
Gpt_uPrescaler	Ftm channel prescaler value.
Gpt_uLptmrPrescaler	Lptmr channel prescaler value.
Gpt_uLptmrClock	Lptmr channel clock source.
Gpt_uFtmPrescaler_Alternate	FTM channel alternate prescaler value.
Gpt_uSRtcClock	SRtc channel clock source.
Gpt_uTimerTriggerConfig	Trigger source configure for LPIT Timer.

### 3.8.4.4 Structure Gpt\_HwChannelInfoType

Gpt configuration type. This is the type of the data structure including the hardware channel info type.



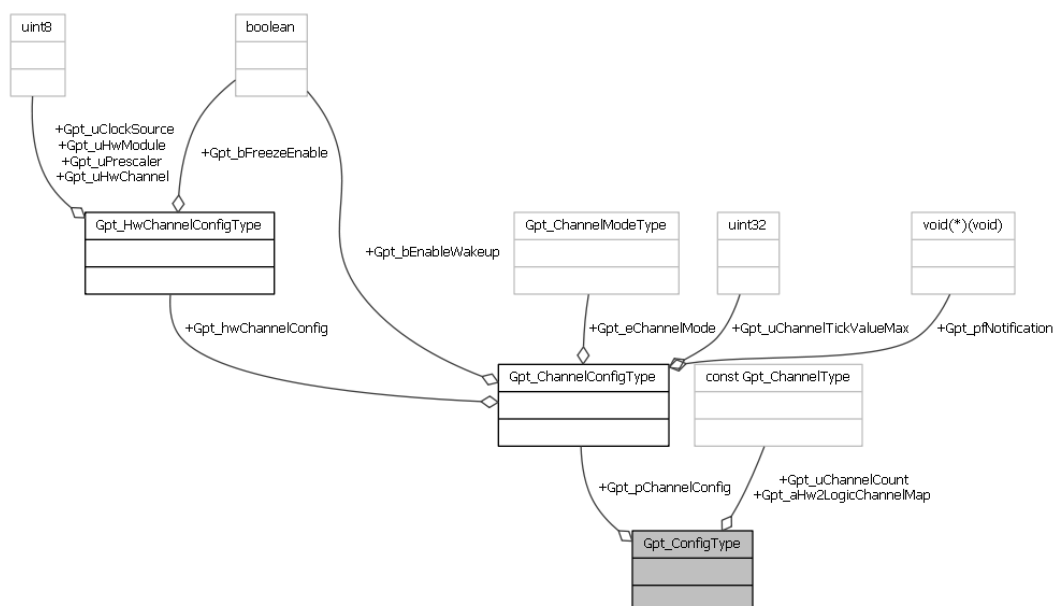


Figure 3-4. Struct Gpt\_HwChannelInfoType

**Implements:** Gpt\_HwChannelInfoType

**Declaration:**

```

typedef struct
{
    VAR(boolean, GPT_VAR) bChannelRollover;
    VAR(Gpt_ValueType, GPT_VAR) uTargetTime;
} Gpt_HwChannelInfoType;
  
```

Table 3-73. Structure Gpt\_HwChannelInfoType member description

Member	Description
bChannelRollover	GPT channel rollover information.
uTargetTime	GPT channel target value.

## 3.8.5 Types Reference

Types supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.3 Rev0001 .

### 3.8.5.1 Typedef Gpt\_ChannelType

Gpt channel ID data type.

**Implements:** Gpt\_ChannelType\_ttypedef

**Type:** uint8

### 3.8.5.2 Typedef Gpt\_NotificationType

Gpt channel notification type. The callback notifications shall be configurable as pointers to user defined functions within the configuration structure.

**Implements:** Gpt\_Notification\_interface

**Type:** typedef void (\* Gpt\_NotificationType)(void);

### 3.8.5.3 Typedef Gpt\_ValueType

Gpt timeout value type.Used for reading and setting the timer value in number of ticks.

**Implements:** Gpt\_ValueType\_ttypedef

**Type:** uint32

### 3.8.5.4 Typedef Gpt\_PrescalerType

Gpt prescaler data type.

**Type:** uint8

### 3.8.5.5 Typedef Gpt\_ClockSourceType

Gpt clock source data type.

**Type:** uint8

## 3.9 Symbolic Names Disclaimer

All containers having the symbolic name tag set as true in the Autosar schema will generate defines like:

```
#define <Container_Short_Name> <Container_ID>
```

For this reason it is forbidden to duplicate the name of such containers across the MCAL configuration, or to use names that may trigger other compile issues (e.g. match existing #ifdefs arguments).



## Chapter 4

# Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the GPT Driver. The most of the parameters are described below.

### 4.1 Configuration elements of Gpt

Included forms :

- [Form IMPLEMENTATION\\_CONFIG\\_VARIANT](#)
- [Form GptConfigurationOfOptApiServices](#)
- [Form GptNonAUTOSAR](#)
- [Form GptDriverConfiguration](#)
- [Form GptPredefTimerConfiguration](#)
- [Form GptChannelConfigSet](#)
- [Form GptHwConfiguration](#)
- [Form CommonPublishedInformation](#)

### 4.2 Form IMPLEMENTATION\_CONFIG\_VARIANT



Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION\_CONFIG\_VARIANT form.

Table 4-1. Attribute IMPLEMENTATION\_CONFIG\_VARIANT detailed description

Property	Value
Label	Config Variant
Type	ENUMERATION
Default	VariantPostBuild
Range	VariantPostBuild VariantPreCompile

### 4.3 Form GptConfigurationOfOptApiServices

This container contains all configuration switches for configuring optional API services of the GPT driver.

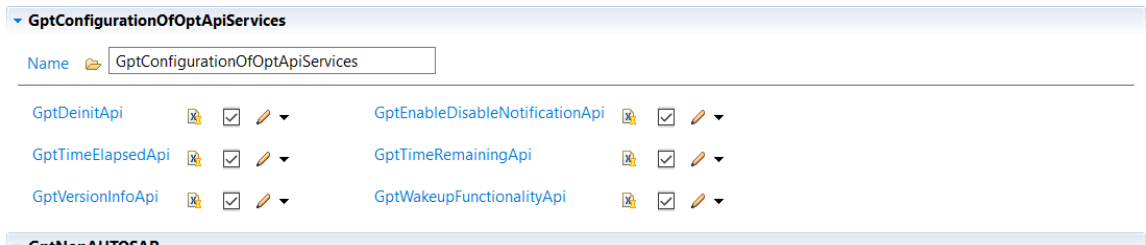


Figure 4-2. Tresos Plugin snapshot for GptConfigurationOfOptApiServices form.

#### 4.3.1 GptDeinitApi (GptConfigurationOfOptApiServices)

Adds / removes the service Gpt\_DeInit() from the code.

Table 4-2. Attribute GptDeinitApi (GptConfigurationOfOptApiServices) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

#### 4.3.2 GptEnableDisableNotificationApi (GptConfigurationOfOptApiServices)

Adds / removes the services Gpt\_EnableNotification() and Gpt\_DisableNotification from the code.

Table 4-3. Attribute GptEnableDisableNotificationApi (GptConfigurationOfOptApiServices) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

### 4.3.3 GptTimeElapsedApi (GptConfigurationOfOptApiServices)

Adds / removes the service Gpt\_GetTimeElapsed() from the code.

**Table 4-4. Attribute GptTimeElapsedApi (GptConfigurationOfOptApiServices) detailed description**

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

### 4.3.4 GptTimeRemainingApi (GptConfigurationOfOptApiServices)

Adds / removes the service Gpt\_GetTimeRemaining() from the code.

**Table 4-5. Attribute GptTimeRemainingApi (GptConfigurationOfOptApiServices) detailed description**

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

### 4.3.5 GptVersionInfoApi (GptConfigurationOfOptApiServices)

Adds / removes the service Gpt\_GetVersionInfo() from the code.

**Table 4-6. Attribute GptVersionInfoApi (GptConfigurationOfOptApiServices) detailed description**

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

### 4.3.6 GptWakeupFunctionalityApi (GptConfigurationOfOptApiServices)

Adds / removes the services Gpt\_SetMode(), Gpt\_EnableWakeup() Gpt\_DisableWakeup() and Gpt\_Cbk\_CheckWakeup() from the code.

**Table 4-7. Attribute GptWakeupFunctionalityApi (GptConfigurationOfOptApiServices) detailed description**

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

### 4.3.7 GptPredefTimerFunctionalityApi (GptConfigurationOfOptApiServices)

Enables/Disables predefined timer functionalities from GPT driver.

Add/Remove Gpt\_GetPredefTimerValue service from driver code.

**Table 4-8. Attribute GptPredefTimerFunctionalityApi (GptConfigurationOfOptApiServices) detailed description**

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

## 4.4 Form GptNonAUTOSAR

This container contains all configuration switches for configuring optional Non AUTOSAR API services of the GPT driver.



**GptNonAUTOSAR**

Name GptNonAUTOSAR

GptChangeNextTimeoutValueApi ☐ GPT Enable User Mode Support ☐

Gpt Standby Wakeup Support ☐ GptEnableDualClockMode ☐

GptEnableTriggers ☐

**GptDriverConfiguration**

**Figure 4-3. Treso Plugin snapshot for GptNonAUTOSAR form.**

#### 4.4.1 GptChangeNextTimeoutValueApi (GptNonAUTOSAR)

Vendor specific: Enables settings for changing the channel counter compare value of a running counter.

**Table 4-9. Attribute GptChangeNextTimeoutValueApi (GptNonAUTOSAR) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

#### 4.4.2 GptEnableDualClockMode (GptNonAUTOSAR)

Vendor specific: Enables settings for using an alternative prescaler.

**Table 4-10. Attribute GptEnableDualClockMode (GptNonAUTOSAR) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

### 4.4.3 GptEnableTriggers (GptNonAUTOSAR)

Vendor specific: Enables/disables LPIT channels' external triggers functionality.

**Table 4-11. Attribute GptEnableTriggers (GptNonAUTOSAR) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

### 4.4.4 GptEnableUserModeSupport (GptNonAUTOSAR)

Vendor specific: Enables/Disable settings for running user mode.

**Table 4-12. Attribute GptEnableUserModeSupport (GptNonAUTOSAR) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

### 4.4.5 GptStandbyWakeupSupport (GptNonAUTOSAR)

Vendor specific: Enables/Disable settings for support Standbywakeup.

**Table 4-13. Attribute GptStandbyWakeupSupport (GptNonAUTOSAR) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false
Enable	true

## 4.5 Form GptDriverConfiguration

This container contains the module-wide configuration (parameters) of the GPT Driver.

Included forms :

- [Form GptClockReferencePoint](#)

Figure 4-4. Tresos Plugin snapshot for GptDriverConfiguration form.

### 4.5.1 GptDevErrorDetect (GptDriverConfiguration)

Enables/Disables development error detection.

Table 4-14. Attribute GptDevErrorDetect (GptDriverConfiguration) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

### 4.5.2 GptPredefTimer100us32bitEnable (GptDriverConfiguration)

Enables/Disables PredefTimer type of 100us32bit.

#### NOTE

*Only 16-bit predefined timer is supported by S32K14X Hardware. Therefore, this node is always unchecked and not editable.*

**Table 4-15. Attribute GptPredefTimer100us32bitEnable (GptDriverConfiguration) detailed description**

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

### 4.5.3 GptPredefTimer1usEnablingGrade (GptDriverConfiguration)

Enables/Disables PredefTimer type of 1us.

GPT\_PREDEF\_TIMER\_1US\_16BIT\_ENABLED: Enables predefined timers with 16-bit resolution and 1MHz frequency.

GPT\_PREDEF\_TIMER\_1US\_16\_24BIT\_ENABLED: Enables 1MHz predefined timers with 16-bit and 24-bit resolution.

GPT\_PREDEF\_TIMER\_1US\_16\_24\_32BIT\_ENABLED: Enables 1MHz predefined timers with 16-bit 24-bit and 32-bit resolution.

GPT\_PREDEF\_TIMER\_1US\_DISABLED: Disable 1MHz predefined timers type

#### NOTE

*Only 16-bit predefined timer is supported by S32K14X Hardware.*

**Table 4-16. Attribute GptPredefTimer1usEnablingGrade (GptDriverConfiguration) detailed description**

Property	Value
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Range	GPT_PREDEF_TIMER_1US_16BIT_ENABLED GPT_PREDEF_TIMER_1US_16_24BIT_ENABLED GPT_PREDEF_TIMER_1US_16_24_32BIT_ENABLED GPT_PREDEF_TIMER_1US_DISABLED
Default	GPT_PREDEF_TIMER_1US_16BIT_ENABLED

### 4.5.4 GptReportWakeupSource (GptDriverConfiguration)

Enables/Disables wakeup source reporting.

**Table 4-17. Attribute GptReportWakeupSource (GptDriverConfiguration) detailed description**

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Default	true

### 4.5.5 GptRegisterLocking (GptDriverConfiguration)

Enables/Disables register locking from user/supervisor mode

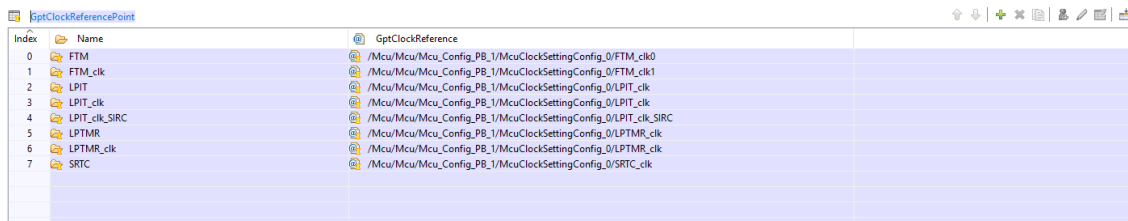
**Table 4-18. Attribute GptRegisterLocking (GptDriverConfiguration) detailed description**

Property	Value
Type	ENUMERATION
Origin	Freescall
Default	NO_REGISTER_LOCKING

### 4.5.6 Form GptClockReferencePoint

This container contains a parameter, which represents a reference to a container of the type McuClockReferencePoint (defined in module MCU).

Is included by form : [Form GptDriverConfiguration](#)

**Figure 4-5. Tresos Plugin snapshot for GptClockReferencePoint form.**

#### 4.5.6.1 GptClockReference (GptClockReferencePoint)

Reference to a container of the type McuClockReferencePoint, to select an input clock.

Table 4-19. Attribute GptClockReference (GptClockReferencePoint) detailed description

Property	Value
Type	REFERENCE
Origin	AUTOSAR_ECUC

## 4.6 Form GptPredefTimerConfiguration

This container is the base of an Configuration Set which contains the configured GPT channels. This way, different configuration sets can be defined for post-build process.

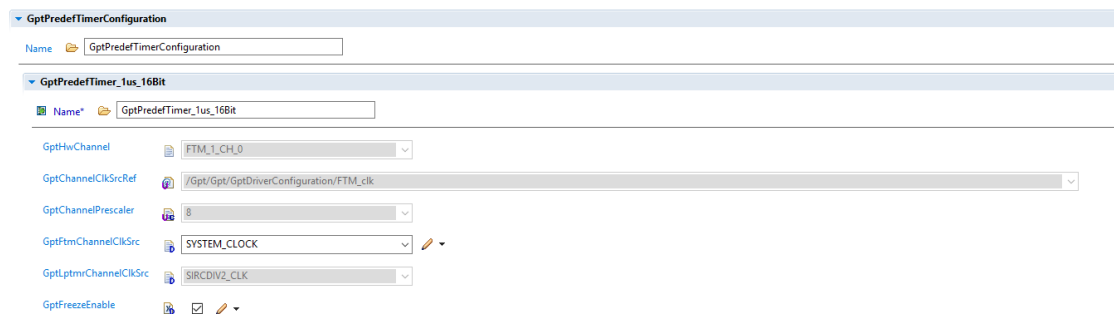


Figure 4-6. Tresos Plugin snapshot for GptPredefTimerConfiguration form.

## 4.7 Form GptChannelConfigSet

This container is the base of an Configuration Set which contains the configured GPT channels. This way, different configuration sets can be defined for post-build process.

**Included forms :**

- [Form GptChannelConfiguration](#)

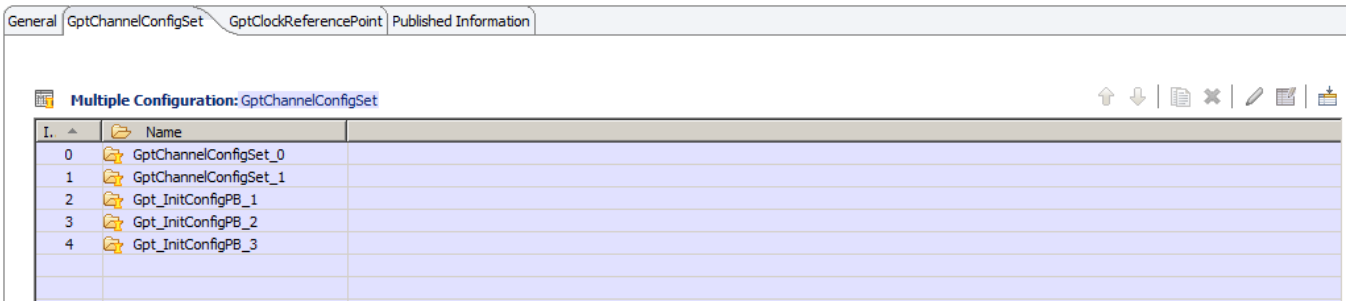


Figure 4-7. Tresos Plugin snapshot for GptChannelConfigSet form.

## 4.7.1 Form GptChannelConfiguration

This container contains the channel-wide configuration (parameters) of the GPT Driver

The screenshot shows the 'General' tab of the GptChannelConfiguration form. The left pane lists the configuration properties, and the right pane shows their corresponding values or controls:

- GptChannelId**: 0
- GptHwChannel**: LPTMR\_0\_CH\_0
- GptChannelMode**: GPT\_CH\_MODE\_CONTINUOUS
- GptChannelTickFrequency** (0 -> 16000000): 2000000.0
- GptFtmPrescaler** (0 -> 7): 0
- GptFtmPrescaler\_Alternate** (0 -> 7): 0
- GptLptmrPrescaler** (0 -> 15): 0
- GptLptmrPrescaler\_Alternate** (0 -> 15): 0
- GptChannelClkSrcRef**: /Gpt/GptDriverConfiguration/LPTMR\_clk
- GptFtmChannelClkSrc**: SYSTEM\_CLOCK
- GptLptmrChannelClkSrc**: SIRC DIV2\_CLK
- GptSRtcChannelClkSrc**: EXTERNAL\_CLOCK
- GptLptmrExternalTrigger**: ☐ GptLptmrReloadOnTrigger
- GptLptmrStopOnInterrupt**: ☐ GptLptmrStartOnTrigger
- GptLptmrTriggerChannels**: Channel\_0\_Trigger\_Source
- GptChannelTickValueMax** (65535 -> 4294967295): 65535
- GptFreezeEnable**: ☒ GptEnableWakeup
- GptNotification**: ☐ GptTestCallback

**Figure 4-8. Tresos Plugin snapshot for GptChannelConfiguration form.**

### 4.7.1.1 GptChannelId (GptChannelConfiguration)

Channel Id of the GPT channel. This value will be assigned to the symbolic name derived of the GptChannelConfiguration container short name.

**Table 4-20. Attribute GptChannelId (GptChannelConfiguration) detailed description**

Property	Value
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Invalid	Range <div>&lt;=4294967295</div> <div>&gt;=0</div>

### 4.7.1.2 GptHwChannel (GptChannelConfiguration)

Vendor specific: Selects the physical GPT Channel.

**Table 4-21. Attribute GptHwChannel (GptChannelConfiguration) detailed description**

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Valid	Range FTM_x_CH_y, where x=0..7, y=0..7 (belong to derivative) LPIT_x_CH_y, where x=0, y=0..3 LPTMR_0_CH_0 SRTC_0_CH_0

### 4.7.1.3 GptChannelMode (GptChannelConfiguration)

Specifies the behaviour of the timerchannel after the timeout has expired

**Table 4-22. Attribute GptChannelMode (GptChannelConfiguration) detailed description**

Property	Value
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	GPT_CH_MODE_ONESHOT
Range	GPT_CH_MODE_CONTINUOUS GPT_CH_MODE_ONESHOT

### 4.7.1.4 GptChannelTickFrequency (GptChannelConfiguration)

Specifies the tick frequency of the timer channel in Hz.

**Table 4-23. Attribute GptChannelTickFrequency (GptChannelConfiguration) detailed description**

Property	Value
Type	FLOAT
Origin	AUTOSAR_ECUC
Symbolic Name	false
Invalid	Range



**Table 4-23. Attribute GptChannelTickFrequency (GptChannelConfiguration) detailed description**

Property	Value
	$\geq 0.0$ $\leq 160000000.0$

#### 4.7.1.5 GptFtmPrescaler and GptFtmPrescaler\_Alternate (GptChannelConfiguration)

Vendor specific: The GPT module specific clock prescaler value.

Selects one of 8 division factors for the clock source selected by GptFtmChannelClkSrcRef. The new prescaler factor affects the clock source on the next system clock cycle after the new value is updated into the register bits.

- 1 - Value in register 1
- 2 - Value in register 2
- 4 - Value in register 4
- 8 - Value in register 8
- 16 - Value in register 16
- 32 - Value in register 32
- 64 - Value in register 64
- 128 - Value in register 128

**Table 4-24. Attribute GptFtmPrescaler and GptFtmPrescaler\_Alternate(GptChannelConfiguration) detailed description**

Property	Value
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range $\geq 1$ $\leq 128$

#### 4.7.1.6 GptLptmrPrescaler and GptLptmrPrescaler\_Alternative(GptChannelConfiguration)

Vendor specific: The GPT module specific clock prescaler value.

Selects one of 16 division factors for the clock source selected by GptLptmrChannelClkSrcRef. The new prescaler factor affects the clock source on the next system clock cycle after the new value is updated into the register bits.

2 - Value in register 2

4 - Value in register 4

8 - Value in register 8

.....

16384 - Value in register 16384

32768 - Value in register 32768

65536 - Value in register 65536

**Table 4-25. Attribute GptLptmrPrescaler and GptLptmrPrescaler\_Alternative (GptChannelConfiguration) detailed description**

Property	Value
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >=2 <=65536

#### 4.7.1.7 GptFtmChannelClkSrc (GptChannelConfiguration)

Vendor specific: The GPT module specific clock input for the timer unit can statically be configured and allows to select different clock sources per module.

Select the clock source for the FlexTimer module for this platform.

SYSTEM\_CLOCK

FIXED\_FREQUENCY\_CLOCK

## EXTERNAL\_CLOCK

**Table 4-26. Attribute GptFtmChannelClkSrc (GptChannelConfiguration) detailed description**

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	SYSTEM_CLOCK
Range	SYSTEM_CLOCK FIXED_FREQUENCY_CLOCK EXTERNAL_CLOCK

**4.7.1.8 GptLptmrChannelClkSrc (GptChannelConfiguration)**

Vendor specific: The GPT module specific clock input for the timer unit can statically be configured and allows to select different clock sources per module.

Select the clock source for the Low Power Timer module for this platform. There are 4 types:

SICR\_CLOCK

LPO\_CLOCK

SIM\_LPO\_CLOCK

PCC\_CLOCK

**Table 4-27. Attribute GptLptmrChannelClkSrc (GptChannelConfiguration) detailed description**

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	SICR_CLOCK
Range	SICR_CLOCK LPO_CLOCK SIM_LPO_CLOCK PCC_CLOCK

#### 4.7.1.9 GptSrtcChannelClkSrc (GptChannelConfiguration)

Vendor specific: The GPT module specific clock input for the timer unit can statically be configured and allows to select different clock sources per module.

Select the clock source for the *Safety Real Time Clock(SRTC)* module for this platform.

EXTERNAL\_CLOCK

LPO\_CLOCK

**Table 4-28. Attribute GptSrtcChannelClkSrc (GptChannelConfiguration) detailed description**

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	EXTERNAL_CLOCK
Range	EXTERNAL_CLOCK LPO_CLOCK

#### 4.7.1.10 GptLPitIsExternalTrigger (GptChannelConfiguration)

Vendor specific: Enables/Disables Trigger Mode for LPIT timers.

**Table 4-29. Attribute GptLPitIsExternalTrigger (GptChannelConfiguration) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

#### 4.7.1.11 GptLPitEnReloadOnTrigger (GptChannelConfiguration)

Vendor specific: LPIT timers will reload on trigger or not.

**Table 4-30. Attribute GptLPitEnReloadOnTrigger (GptChannelConfiguration) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

#### 4.7.1.12 GptLPitEnStopOnInterrupt (GptChannelConfiguration)

Vendor specific: LPIT timers will stop on Interrupt or not

**Table 4-31. Attribute GptLPitEnStopOnInterrupt (GptChannelConfiguration) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

#### 4.7.1.13 GptLPitEnStartOnTrigger (GptChannelConfiguration)

Vendor specific: LPIT timers will start On trigger or not.

**Table 4-32. Attribute GptLPitEnStartOnTrigger (GptChannelConfiguration) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

#### 4.7.1.14 GptLPitTriggerChannels (GptChannelConfiguration)

Vendor specific: Configuring Trigger channels for LPIT timers.

Channel\_0\_Trigger\_Source

Channel\_1\_Trigger\_Source

Channel\_2\_Trigger\_Source

...

Channel\_14\_Trigger\_Source

Channel\_15\_Trigger\_Source

**Table 4-33. Attribute GptLPitTriggerChannels (GptChannelConfiguration) detailed description**

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	Channel_0_Trigger_Source
Range	Channel_0_Trigger_Source Channel_1_Trigger_Source Channel_2_Trigger_Source ... Channel_15_Trigger_Source

#### 4.7.1.15 GptChannelTickValueMax (GptChannelConfiguration)

Maximum value in ticks, the timer channel is able to count. With the next tick, the timer rolls over to zero. There is mandatory to set 4294967296 for the LPIT and SRTC channels (corresponding to the 32 bits counter resolution)

**Table 4-34. Attribute GptChannelTickValueMax (GptChannelConfiguration) detailed description**

Property	Value
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false

#### 4.7.1.16 GptFreezeEnable (GptChannelConfiguration)

Vendor specific: Select to set Freeze enable for the hw resources.

**Table 4-35. Attribute GptFreezeEnable (GptChannelConfiguration) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

#### 4.7.1.17 GptEnableWakeup (GptChannelConfiguration)

Enables wakeup capability of CPU for a channel.

**Table 4-36. Attribute GptEnableWakeup (GptChannelConfiguration) detailed description**

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

#### 4.7.1.18 GptNotification (GptChannelConfiguration)

Function pointer to callback function(for non-wakeup notification). The field is editable only if the switch GptEnableDisableNotificationApi is true.

**Table 4-37. Attribute GptNotification (GptChannelConfiguration) detailed description**

Property	Value
Type	FUNCTION-NAME
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	NULL_PTR

## 4.7.2 Form GptHwInterruptChannel

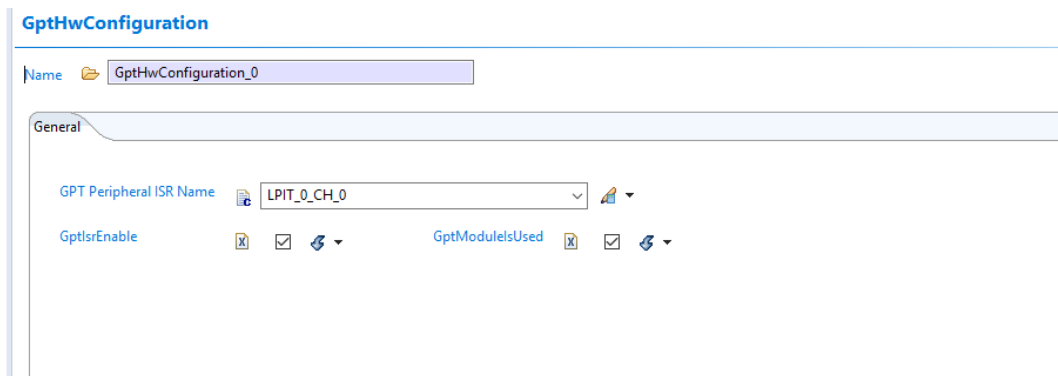


Figure 4-9. Tresos Plugin snapshot for GptHwInterruptChannel form.

### 4.7.2.1 GptIsrHwId (GptHwInterruptChannel)

Vendor specific: Selects the physical GPT Channel.

Table 4-38. Attribute GptHwChannel (GptHwInterruptChannel) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Range	FTM_x_CH_y LPTMR_x_CH_y LPIT_x_CH_y SRTC_0_CH_0

### 4.7.2.2 GptIsrEnable (GptHwInterruptChannel)

Vendor specific: check ISr is enable or not.

Table 4-39. Attribute GptIsrEnable (GptHwInterruptChannel) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false



### 4.7.2.3 GptChannellsUsed (GptHwInterruptChannel)

Vendor specific: Check channel is used or not.

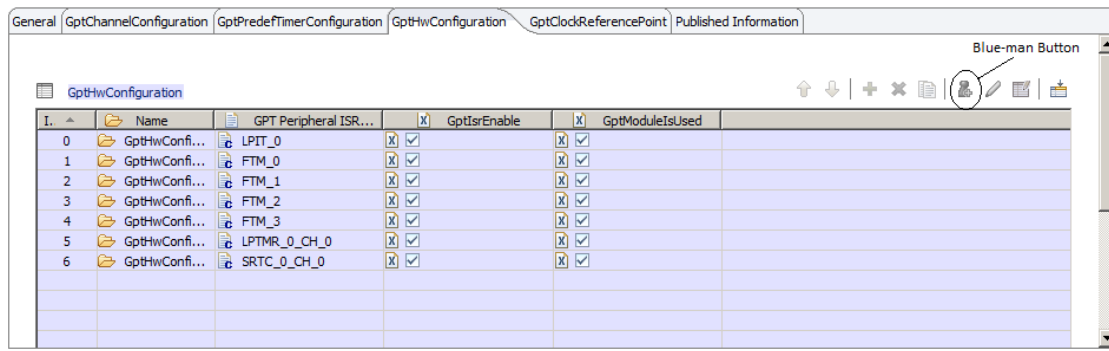
**Table 4-40. Attribute GptChannellsUsed (GptHwInterruptChannel) detailed description**

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

## 4.8 Form GptHwConfiguration

This container contains list of available IPs and their interrupt sources supported by hardware. This tab also enables or disables modules and their ISRs which are used in application.

When the first time configuring for GPT module, please click on BLUE MAN button at the right conner of this tab for configuring automatically.



**Figure 4-10. Tresos Plugin snapshot for GptHwConfiguration form.**

## 4.9 Form CommonPublishedInformation

Common container, aggregated by all modules. It contains published information about vendor and versions.

CommonPublishedInformation

Name

CommonPublishedInformation

ArReleaseMajorVersion

4

ArReleaseMinorVersion

3

ArReleaseRevisionVersion

1

ModuleId

100

SwMajorVersion

1

SwMinorVersion

0

SwPatchVersion

1

VendorApilnfix

VendorId

43

Figure 4-11. Tresos Plugin snapshot for CommonPublishedInformation form.

4.9.1 ArReleaseMajorVersion (CommonPublishedInformation)

Major version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-41. Attribute ArReleaseMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	4
Invalid	Range >=4 <=4

4.9.2 ArReleaseMinorVersion (CommonPublishedInformation)

Minor version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-42. Attribute ArReleaseMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Type	INTEGER_LABEL

Table continues on the next page...

**Table 4-42. Attribute ArReleaseMinorVersion (CommonPublishedInformation) detailed description (continued)**

Property	Value
Origin	Custom
Symbolic Name	false
Default	3
Invalid	Range >=3 <=3

### 4.9.3 ArReleaseRevisionVersion (CommonPublishedInformation)

Revision version number of AUTOSAR specification on which the appropriate implementation is based on.

**Table 4-43. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description**

Property	Value
Label	AUTOSAR Release Revision Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range >=1 <=1

### 4.9.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

**Table 4-44. Attribute ModuleId (CommonPublishedInformation) detailed description**

Property	Value
Label	Module Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	100

*Table continues on the next page...*

**Table 4-44. Attribute ModuleId (CommonPublishedInformation) detailed description (continued)**

Property	Value
Invalid	Range $\geq 100$ $\leq 100$

### 4.9.5 SwMajorVersion (CommonPublishedInformation)

Major version number of the vendor specific implementation of the module. The numbering is vendor specific.

**Table 4-45. Attribute SwMajorVersion (CommonPublishedInformation) detailed description**

Property	Value
Label	Software Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range $\geq 1$ $\leq 1$

### 4.9.6 SwMinorVersion (CommonPublishedInformation)

Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.

**Table 4-46. Attribute SwMinorVersion (CommonPublishedInformation) detailed description**

Property	Value
Label	Software Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range $\geq 0$ $\leq 0$

### 4.9.7 SwPatchVersion (CommonPublishedInformation)

Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.

**Table 4-47. Attribute SwPatchVersion (CommonPublishedInformation) detailed description**

Property	Value
Label	Software Patch Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range >=1 <=1

### 4.9.8 VendorApiInfix (CommonPublishedInformation)

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows:

<ModuleName>\_<VendorId>\_<VendorApiInfix><Api name from SWS>. E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can\_Write defined in the SWS will translate to Can\_123\_v11r456Write. This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

**Table 4-48. Attribute VendorApiInfix (CommonPublishedInformation) detailed description**

Property	Value
Label	Vendor Api Infix
Type	STRING_LABEL
Origin	Custom
Symbolic Name	false
Default	
Enable	false

### 4.9.9 VendorId (CommonPublishedInformation)

Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list.

**Table 4-49. Attribute VendorId (CommonPublishedInformation) detailed description**

Property	Value
Label	Vendor Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	43
Invalid	Range >=43 <=43

**How to Reach Us:****Home Page:**[nxp.com](http://nxp.com)**Web Support:**[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro,  $\mu$ Vision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2019 NXP B.V.

Document Number UM2GPTASR4.3 Rev0001R1.0.1  
Revision 1.0