User Manual

for MPC574XG GPT Driver

Document Number: UM35GPTASR4.2 Rev0002 R1.0.0
Rev. 5.0.0

Contents

Section number **Title** **Page**

Chapter 1
Revision History

Chapter 2
Introduction

Chapter 3
Driver

Chapter 4
Tresos Configuration Plug-in

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0.0	22/08/2014	Son Nguyen	User Manual for MPC574XG - 0.9.0 Release
2.0.0	24/04/2015	Phap Nguyen	User Manual for MPC574XG - 1.0.0 Release
3.0.0	10/07/2015	Phap Nguyen	User Manual for CALYPSO - 1.0.1 Release
4.0.0	12/08/2016	Nghia LE	User Manual for CALYPSO - 1.0.2 Release
5.0.0	17/02/2017	Nghia Tran Thinh	User Manual for CALYPSO AUTOSAR4.2.2 - 1.0.0 Release

Chapter 2
Introduction

This User Manual describes NXP Semiconductor AUTOSAR General Purpose Timer (GPT) for MPC574XG .

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 intented to be used with the following microcontroller devices of NXP Semiconductor .

Table 2-1. MPC574XG Derivatives

NXP Semiconductor	MPC5748G_LQFP176, MPC5748G_MAPBGA256, MPC5748G_MAPBGA324, MPC5747G_LQFP176, MPC5747G_MAPBGA256, MPC5747G_MAPBGA324, MPC5746G_LQFP176, MPC5746G_MAPBGA256, MPC5746G_MAPBGA324, MPC5748C_LQFP176, MPC5748C_MAPBGA256, MPC5748C_MAPBGA324, MPC5747C_LQFP176, MPC5747C_MAPBGA256, MPC5747C_MAPBGA324, MPC5746C_LQFP176, MPC5746C_MAPBGA256, MPC5746C_MAPBGA324, MPC5746C_MAPBGA100, MPC5745C_LQFP176, MPC5745C_MAPBGA256, MPC5745C_MAPBGA100, MPC5744C_LQFP176, MPC5744C_MAPBGA256,

Table 2-1. MPC574XG Derivatives

	MPC5744C_MAPBGA100, MPC5746B_LQFP176, MPC5746B_MAPBGA256, MPC5746B_MAPBGA100, MPC5744B_LQFP176, MPC5744B_MAPBGA256, MPC5744B_MAPBGA100, MPC5745B_LQFP176, MPC5745B_MAPBGA256, MPC5745B_MAPBGA100

All of the above microcontroller devices are collectively named as MPC574XG .

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 Make file Interface
C/CPP	C and C++ Source Code
DET	Diagnostic Event Traceability
EMIOS	Configurable Enhanced Modular IO Subsystem
GPT	General Purpose Timer
N/A	Not Applicable
MCU	Micro Controller Unit
PIT	Periodic Interrupt Timer
RTI	Real Time Interrupt
RTC	Real Time Clock
STM	System Timer Module
VLE	Variable Length Encoding

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 4.2 Rev0002GPT Driver Software Specification Document.	V3.2.0
2	MPC5748G Reference Manual	Rev. 5, 12/2016
3	MPC5746C Reference Manual	Rev. 4, 12/2016
4	MPC5748G_1N81M_Rev.2 (official document) (1N81M)	Jun-16
5	MPC5748G_1N81M_0N78S_Comparison_Summary_v2_0 (internal document) (1N81M, 0N78S)	31.10.2016
6	MPC5746C_1N06M_Rev.4 (official document) (1N06M)	Jul-16
7	MPC5746C_cut1.1_cut2.0_cut2.1_comparison_v0 (internal document) (1N06M, 0N84S, 1N84S)	14-Sep-16

Table continues on the next page...

Table 2-3. Reference List (continued)

#	Title	Version
8	C3M_cut2.1_new_errata_20170113 (internal document) (1N84S)	13-Jan-17

Chapter 3
Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 4.2 Rev0002GPT Driver Software Specification document (See Table Reference List).

3.2 Driver Design Sumary

The GPT Driver implements maximum 55 channels on 3 types of MPC574XG peripherals.

25 channels are implemented on 3 Enhanced Modular IO Subsystem (eMIOS) modules.

17 channels are implemented on single Periodic Interrupt Timer (PIT) modules - 16 PIT channels, 1 PITRTI channel.

12 channels are implemented on 3 System Timer Module (STM) modules.

1 channel is implemented on 1 Real Time Clock (RTC) modules.

eMIOS channels

- 32 x 16-bit counters per module

PIT timer channels

- Independent timeout periods for each timer

STM timer channels

- One 32-bit up counter with 8-bit prescaler (1 to 256).
- Four 32-bit compare channels
- Independent interrupt source for each channel

RTC channels

- Interrupt at a pre-defined interval independent of the mode of operation

NOTE

The table provides information regarding the Timer channels available for the various derivatives across different packages in MPC574XG family. This table lists only the supported packages by GPT driver.

Table 3-1. GPT Hardware channels availability for MPC574XG family

Device	Total eMIOS channels	Total PIT channels	Total STM channels	Total RTC channels	Total RTI channels
mpc5745b_lqfp176	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5745b_mapbga100	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5745b_mapbga256	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5745c_lqfp176	19 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5745c_mapbga100	19 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5745c_mapbga256	19 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5745d_lqfp176	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5745d_mapbga100	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5745d_mapbga256	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5746b_lqfp176	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5746b_mapbga100	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5746b_mapbga256	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5746c_lqfp176	19 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5746c_mapbga100	19 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5746c_mapbga256	19 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5746c_mapbga324	19 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5746d_lqfp176	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5746d_mapbga100	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5746d_mapbga256	19 ch, 16-bit	16 ch	4 ch	1 ch	1 ch
mpc5746g_lqfp176	25 ch, 16-bit	16 ch	12 ch	1 ch	1 ch
mpc5746g_mapbga256	25 ch, 16-bit	16 ch	12 ch	1 ch	1 ch
mpc5746g_mapbga324	25 ch, 16-bit	16 ch	12 ch	1 ch	1 ch
mpc5747c_lqfp176	25 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5747c_mapbga256	25 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5747c_mapbga324	25 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5747g_lqfp176	25 ch, 16-bit	16 ch	12 ch	1 ch	1 ch
mpc5747g_mapbga256	25 ch, 16-bit	16 ch	12 ch	1 ch	1 ch
mpc5747g_mapbga324	25 ch, 16-bit	16 ch	12 ch	1 ch	1 ch
mpc5748c_lqfp176	25 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5748c_mapbga256	25 ch, 16-bit	16 ch	8 ch	1 ch	1 ch
mpc5748c_mapbga324	25 ch, 16-bit	16 ch	8 ch	1 ch	1 ch

Table continues on the next page...

Table 3-1. GPT Hardware channels availability for MPC574XG family (continued)

Device	Total eMIOS channels	Total PIT channels	Total STM channels	Total RTC channels	Total RTI channels
mpc5748g_lqfp176	25 ch, 16-bit	16 ch	12 ch	1 ch	1 ch
mpc5748g_mapbga256	25 ch, 16-bit	16 ch	12 ch	1 ch	1 ch
mpc5748g_mapbga324	25 ch, 16-bit	16 ch	12 ch	1 ch	1 ch

3.3 Deviation from Requirements

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

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 3-2. 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 3-3. Driver Deviations Table

Requirement	Status	Description	Notes
GPT172	N/I	The module shall optionally include the Dem.h file if any production error will be issued by the implementation. Comment: By this inclusion the APIs to report errors as well as the required Event Id symbols are included. This specification defines the name of the Event Id symbols, which are provided by XML to the DEM configuration tool. The DEM configuration tool assigns ECU dependent values to the Event Id symbols and publishes the symbols in Dem_IntErrId.h.	The production errors will be reported via SERR

Table continues on the next page...

Table 3-3. Driver Deviations Table (continued)

Requirement	Status	Description	Notes
GPT179	N/I	If production errors are specified for GPT module: Production errors shall be reported to Diagnostic Event Manager[7].	The production errors will be reported via SERR
GPT261	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.
GPT278	N/I	GPT import the following types: Module: Dem Imported Type: Dem_EventIdType, Dem_EventStatusType Module: EcuM Imported Type:EcuM_WakeupSourceType Module: Std_Types Imported Type: Std_ReturnType, Std_VersionInfoType	Production errors are reported through SERR so the following types are not imported: Dem_EventIdType, Dem_EventStatusType,Dem
GPT333	N/I	Each variable that shall be accessible by AUTOSAR Debugging, shall be defined as global variable.	Part6 -Table8- 1e: Avoid global variables or else justify their usage.ebugging concept not supported.
GPT334	N/I	All type definitions of variables which shall be debugged, shall be accessible by the header file Gpt.h.	Part6 -Table8- 1e: Avoid global variables or else justify their usage.ebugging concept not supported.
GPT335	N/I	The declaration of variables in the header file shall allow to calculate the size of the variables by C-"sizeof".	Debugging Support not covered by sMCAL
GPT336	N/I	Variables available for debugging shall be described in the respective Basic Software Module Description	Debugging Support not covered by sMCAL
GPT337	N/I	The operation mode of the GPT driver and the state of each timer channel shall be available for debugging.	Debugging Support not covered by sMCAL
GPT354	N/S	If the register can affect several hardware modules and if it is not an I/O register it shall be initialized by the MCU driver	Not a GPT requirement.
GPT355	N/S	One-time writable registers that require initialization directly after reset shall be initialized by the startup code	Not a GPT requirement.
GPT356	N/S	All other registers shall be initialized by the startup code	Not a GPT requirement.
GPT372	N/I	File name: Gpt_Irq.c Description/Condition: File for implementation of interrupt frame (not interrupt service routine)	Gpt_Irq.c is not needed because the generic interrupt handler(not the interrupt service routine) is defined in the Gpt.c. This approach shall simplify the GPT file structure and implementation.

3.4 Runtime Errors

None

3.5 Software specification

The following sections contains driver software specifications.

3.5.1 Define Reference

Constants supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.2 Rev0002 .

3.5.1.1 Define GPT_VENDOR_ID_C

Table 3-4. Define GPT_VENDOR_ID_C Description

Name	GPT_VENDOR_ID_C
Initializer	43

3.5.1.2 Define GPT_AR_RELEASE_MAJOR_VERSION_C

Violates: MISRA 2004 Rule 1.4, The compiler/linker shall be checked to ensure that 31 character significance and case sensitivity are supported for external identifiers.

Table 3-5. Define GPT_AR_RELEASE_MAJOR_VERSION_C Description

Name	GPT_AR_RELEASE_MAJOR_VERSION_C
Initializer	4

3.5.1.3 Define GPT_AR_RELEASE_MINOR_VERSION_C

Violates: MISRA 2004 Rule 1.4, The compiler/linker shall be checked to ensure that 31 character significance and case sensitivity are supported for external identifiers.

Table 3-6. Define GPT_AR_RELEASE_MINOR_VERSION_C Description

Name	GPT_AR_RELEASE_MINOR_VERSION_C
Initializer	2

3.5.1.4 Define GPT_AR_RELEASE_REVISION_VERSION_C

Violates: MISRA 2004 Rule 1.4, The compiler/linker shall be checked to ensure that 31 character significance and case sensitivity are supported for external identifiers.

Table 3-7. Define GPT_AR_RELEASE_REVISION_VERSION_C Description

Name	GPT_AR_RELEASE_REVISION_VERSION_C
Initializer	2

3.5.1.5 Define GPT_SW_MAJOR_VERSION_C

Table 3-8. Define GPT_SW_MAJOR_VERSION_C Description

Name	GPT_SW_MAJOR_VERSION_C
Initializer	1

3.5.1.6 Define GPT_SW_MINOR_VERSION_C

Table 3-9. Define GPT_SW_MINOR_VERSION_C Description

Name	GPT_SW_MINOR_VERSION_C
Initializer	0

3.5.1.7 Define GPT_SW_PATCH_VERSION_C

Table 3-10. Define GPT_SW_PATCH_VERSION_C Description

Name	GPT_SW_PATCH_VERSION_C
Initializer	2

3.5.1.8 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 3-11. Define GPT_E_ALREADY_INITIALIZED Description

Name	GPT_E_ALREADY_INITIALIZED
Initializer	(uint8)0x0DU

3.5.1.9 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 3-12. Define GPT_E_BUSY Description

Name	GPT_E_BUSY
Initializer	(uint8)0x0BU

3.5.1.10 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 3-13. Define GPT_E_INVALID_CALL Description

Name	GPT_E_INVALID_CALL
Initializer	(uint8)0xA0U

3.5.1.11 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

Name	GPT_E_PARAM_CHANNEL
Initializer	(uint8)0x14U

3.5.1.12 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

Name	GPT_E_PARAM_MODE
Initializer	(uint8)0x1FU

3.5.1.13 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

Name	GPT_E_PARAM_POINTER
Initializer	(uint8)0x16U

3.5.1.14 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

Name	GPT_E_PARAM_VALUE
Initializer	(uint8)0x15U

3.5.1.15 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 3-18. Define GPT_E_UNINIT Description

Name	GPT_E_UNINIT
Initializer	(uint8)0x0AU

3.5.1.16 Define GPT_CHECKWAKEUP_ID

API service ID for Gpt_CheckWakeup function.

Details:

Parameters used when raising an error/exception

Table 3-19. Define GPT_CHECKWAKEUP_ID Description

Name	GPT_CHECKWAKEUP_ID
Initializer	(uint8)0x0CU

3.5.1.17 Define GPT_DEINIT_ID

API service ID for Gpt_DeInit function.

Details:

Parameters used when raising an error/exception

Table 3-20. Define GPT_DEINIT_ID Description

Name	GPT_DEINIT_ID
Initializer	(uint8)0x02U

3.5.1.18 Define GPT_DISABLENOTIFICATION_ID

API service ID for Gpt_DisableNotification function.

Details:

Parameters used when raising an error/exception

Table 3-21. Define GPT_DISABLENOTIFICATION_ID Description

Name	GPT_DISABLENOTIFICATION_ID
Initializer	(uint8)0x08U

3.5.1.19 Define GPT_DISABLEWAKEUP_ID

API service ID for Gpt_DisableWakeup function.

Details:

Parameters used when raising an error/exception

Table 3-22. Define GPT_DISABLEWAKEUP_ID Description

Name	GPT_DISABLEWAKEUP_ID
Initializer	(uint8)0x0AU

3.5.1.20 Define GPT_INSTANCE_ID

Instance ID of this gpt driver.

Table 3-23. Define GPT_INSTANCE_ID Description

Name	GPT_INSTANCE_ID
Initializer	(uint8)0U

3.5.1.21 Define GPT_MODULE_ID

Table 3-24. Define GPT_MODULE_ID Description

Name	GPT_MODULE_ID
Initializer	100

3.5.1.22 Define GPT_ENABLENOTIFICATION_ID

API service ID for Gpt_EnableNotification function.

Details:

Parameters used when raising an error/exception

Table 3-25. Define GPT_ENABLENOTIFICATION_ID Description

Name	GPT_ENABLENOTIFICATION_ID
Initializer	(uint8)0x07U

3.5.1.23 Define GPT_ENABLEWAKEUP_ID

API service ID for Gpt_EnableWakeup function.

Details:

Parameters used when raising an error/exception

Table 3-26. Define GPT_ENABLEWAKEUP_ID Description

Name	GPT_ENABLEWAKEUP_ID
Initializer	(uint8)0x0BU

3.5.1.24 Define GPT_GETVERSIONINFO_ID

API service ID for Gpt_GetVersionInfo function.

Details:

Parameters used when raising an error/exception

Table 3-27. Define GPT_GETVERSIONINFO_ID Description

Name	GPT_GETVERSIONINFO_ID
Initializer	(uint8)0x00U

3.5.1.25 Define GPT_INIT_ID

API service ID for Gpt_Init function.

Details:

Parameters used when raising an error/exception

Table 3-28. Define GPT_INIT_ID Description

Name	GPT_INIT_ID
Initializer	(uint8)0x01U

3.5.1.26 Define GPT_PROCESSCOMMONINTERRUPT_ID

API service ID for Gpt_ConfigurationLock function.

Details:

Parameters used when raising an error/exception

Table 3-29. Define GPT_PROCESSCOMMONINTERRUPT_ID Description

Name	GPT_PROCESSCOMMONINTERRUPT_ID
Initializer	(uint8)0x0DU

3.5.1.27 Define GPT_SETMODE_ID

API service ID for Gpt_SetMode function.

Details:

Parameters used when raising an error/exception

Table 3-30. Define GPT_SETMODE_ID Description

Name	GPT_SETMODE_ID
Initializer	(uint8)0x09U

3.5.1.28 Define GPT_STARTTIMER_ID

API service ID for Gpt_StartTimer function.

Details:

Parameters used when raising an error/exception

Table 3-31. Define GPT_STARTTIMER_ID Description

Name	GPT_STARTTIMER_ID
Initializer	(uint8)0x05U

3.5.1.29 Define GPT_STOPTIMER_ID

API service ID for Gpt_StopTimer function.

Details:

Parameters used when raising an error/exception

Table 3-32. Define GPT_STOPTIMER_ID Description

Name	GPT_STOPTIMER_ID
Initializer	(uint8)0x06U

3.5.1.30 Define GPT_TIMEELAPSED_ID

API service ID for Gpt_GetTimeElapsed function.

Details:

Parameters used when raising an error/exception

Table 3-33. Define GPT_TIMEELAPSED_ID Description

Name	GPT_TIMEELAPSED_ID
Initializer	(uint8)0x03U

3.5.1.31 Define GPT_TIMEREMAINING_ID

API service ID for Gpt_GetTimeRemaining function.

Details:

Parameters used when raising an error/exception

Table 3-34. Define GPT_TIMEREMAINING_ID Description

Name	GPT_TIMEREMAINING_ID
Initializer	(uint8)0x04U

3.5.1.32 Define GPT_CHANGE_NEXT_TIMEOUT_VALUE_ID

API service ID for Gpt_ChangeNextTimeoutValue function.

Details:

Parameters used when raising an error/exception

Table 3-35. Define GPT_CHANGE_NEXT_TIMEOUT_VALUE_ID Description

Name	GPT_CHANGE_NEXT_TIMEOUT_VALUE_ID
Initializer	(uint8)0x0fU

3.5.1.33 Define GPT_DEV_ERROR_DETECT

Gpt Dev error detect switch.

Table 3-36. Define GPT_DEV_ERROR_DETECT Description

Name	GPT_DEV_ERROR_DETECT
Initializer	(STD_ON)

3.5.1.34 Define GPT_PRECOMPILE_SUPPORT

Table 3-37. Define GPT_PRECOMPILE_SUPPORT Description

Name	GPT_PRECOMPILE_SUPPORT
Initializer	(STD_OFF)

3.5.1.35 Define GPT_REPORT_WAKEUP_SOURCE

Report Wakeup Source switch.

Table 3-38. Define GPT_REPORT_WAKEUP_SOURCE Description

Name	GPT_REPORT_WAKEUP_SOURCE
Initializer	(STD_ON)

3.5.1.36 Define GPT_DEINIT_API

GPT_DEINIT_API switch.

Table 3-39. Define GPT_DEINIT_API Description

Name	GPT_DEINIT_API
Initializer	(STD_ON)

3.5.1.37 Define GPT_ENABLE_DISABLE_NOTIFICATION_API

GPT_ENABLE_DISABLE_NOTIFICATION_API switch.

Table 3-40. Define GPT_ENABLE_DISABLE_NOTIFICATION_API Description

Name	GPT_ENABLE_DISABLE_NOTIFICATION_API
Initializer	(STD_ON)

3.5.1.38 Define GPT_TIME_ELAPSED_API

GPT_TIME_ELAPSED_API switch.

Table 3-41. Define GPT_TIME_ELAPSED_API Description

Name	GPT_TIME_ELAPSED_API
Initializer	(STD_ON)

3.5.1.39 Define GPT_TIME_REMAINING_API

GPT_TIME_REMAINING_API switch.

Table 3-42. Define GPT_TIME_REMAINING_API Description

Name	GPT_TIME_REMAINING_API
Initializer	(STD_ON)

3.5.1.40 Define GPT_VERSION_INFO_API

GPT_VERSION_INFO_API switch.

Table 3-43. Define GPT_VERSION_INFO_API Description

Name	GPT_VERSION_INFO_API
Initializer	(STD_ON)

3.5.1.41 Define GPT_WAKEUP_FUNCTIONALITY_API

GPT_WAKEUP_FUNCTIONALITY_API switch.

Table 3-44. Define GPT_WAKEUP_FUNCTIONALITY_API Description

Name	GPT_WAKEUP_FUNCTIONALITY_API
Initializer	(STD_ON)

3.5.1.42 Define GPT_CHANGE_NEXT_TIMEOUT_VALUE

GPT_CHANGE_NEXT_TIMEOUT_VALUE switch.

Table 3-45. Define GPT_CHANGE_NEXT_TIMEOUT_VALUE Description

Name	GPT_CHANGE_NEXT_TIMEOUT_VALUE
Initializer	(STD_OFF)

3.5.1.43 Define GPT_SET_CLOCK_MODE

GPT_SET_CLOCK_MODE switch.

Table 3-46. Define GPT_SET_CLOCK_MODE Description

Name	GPT_SET_CLOCK_MODE
Initializer	(STD_ON)

3.5.1.44 Define GPT_USER_MODE_SOFT_LOCKING

GPT_USER_MODE_SOFT_LOCKING switch.

Table 3-47. Define GPT_USER_MODE_SOFT_LOCKING Description

Name	GPT_USER_MODE_SOFT_LOCKING
Initializer	(STD_OFF)

3.5.1.45 Define GPT_CHN_NOT_USED

Table 3-48. Define GPT_CHN_NOT_USED Description

Name	GPT_CHN_NOT_USED
Initializer	255U

3.5.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.2 Rev0002 .

3.5.2.1 Enumeration Gpt_ChannelModeType

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

Table 3-49. Enumeration Gpt_ChannelModeType Values

Name	Initializer	Description
GPT_CH_MODE_CONTINUOUS	0U	GPT channel mode - continuous mode.
GPT_CH_MODE_ONESHOT	1U	GPT channel mode - one-shot mode.

3.5.2.2 Enumeration Gpt_ModeType

This enumerated type allows the selection of different power modes.

Implements: Gpt_ModeType_enumeration

Table 3-50. Enumeration Gpt_ModeType Values

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

3.5.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.2 Rev0002 .

3.5.3.1 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 3-51. Gpt_CheckWakeup Arguments

Type	Name	Direction	Description
EcuM_WakeupSourceType	wakeupSource	**input**	Wakeup source.

3.5.3.2 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);`

3.5.3.3 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 3-52. Gpt_DisableNotification Arguments

Type	Name	Direction	Description
`Gpt_ChannelType`	channel	**input**	Channel id.

3.5.3.4 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 3-53. Gpt_DisableWakeup Arguments

Type	Name	Direction	Description
`Gpt_ChannelType`	channel	**input**	Channel id.

3.5.3.5 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 3-54. Gpt_EnableNotification Arguments

Type	Name	Direction	Description
`Gpt_ChannelType`	channel	**input**	Channel id.

3.5.3.6 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 3-55. Gpt_EnableWakeup Arguments

Type	Name	Direction	Description
`Gpt_ChannelType`	channel	**input**	Channel id.

3.5.3.7 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 3-56. 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.

3.5.3.8 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 3-57. Gpt_GetTimeElapsed Arguments

Type	Name	Direction	Description
`Gpt_ChannelType`	channel	**input**	- channel id.

3.5.3.9 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 3-58. Gpt_GetTimeRemaining Arguments

Type	Name	Direction	Description
`Gpt_ChannelType`	channel	**input**	- channel id.

3.5.3.10 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 3-59. Gpt_GetVersionInfo Arguments

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

3.5.3.11 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 3-60. Gpt_Init Arguments

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

3.5.3.12 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 3-61. Gpt_SetMode Arguments

Type	Name	Direction	Description
`Gpt_ModeType`	mode	**input**	Operation mode.

3.5.3.13 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 GPT292) 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 3-62. Gpt_StartTimer Arguments

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

3.5.3.14 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 3-63. Gpt_StopTimer Arguments

Type	Name	Direction	Description
`Gpt_ChannelType`	channel	**input**	Channel id.

3.5.3.15 Function Gpt_SetClockMode

GPT driver function for 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 channel);`

Table 3-64. Gpt_StopTimer Arguments

Type	Name	Direction	Description
Gpt_ClockModeType	clkMode	**input**	Mode clock is selected

3.5.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.2 Rev0002 .

3.5.4.1 Structure Gpt_ChannelConfigType

Gpt channel configuration type.

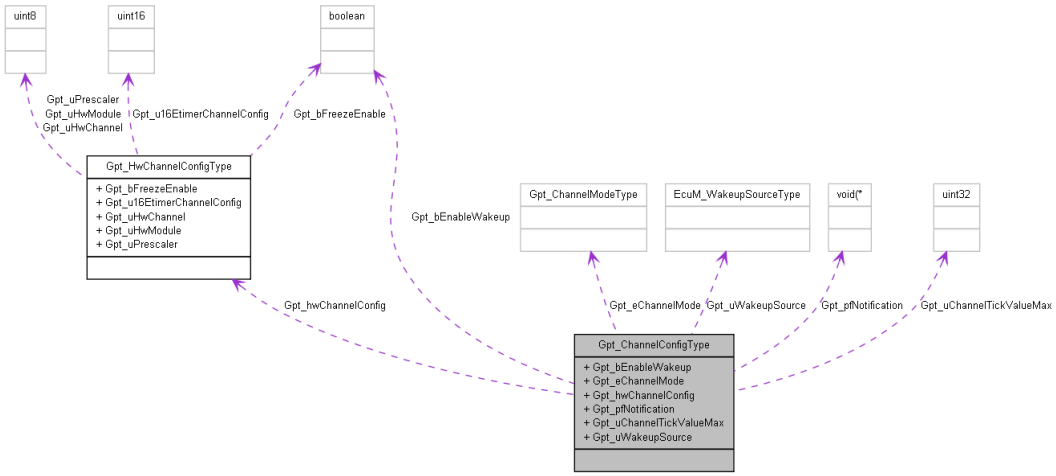

Figure 3-1. Struct Gpt_ChannelConfigType

Declaration:

```
typedef struct
    {
        VAR(boolean, GPT_VAR) Gpt_bEnableWakeup;
        VAR(Gpt_NotificationType, GPT_VAR) Gpt_pfNotification;
        #if ((GPT_WAKEUP_FUNCTIONALITY_API == STD_ON)
&&                                          (GPT_REPORT_WAKEUP_SOURCE == STD_ON))
        VAR(EcuM_WakeupSourceType, GPT_VAR) Gpt_uWakeupSource;
        #endif
        VAR(Gpt_ValueType, GPT_VAR) Gpt_uChannelTickValueMax;
        VAR(Gpt_ChannelModeType, GPT_VAR) Gpt_eChannelMode;
        VAR(Gpt_HwChannelConfigType,GPT_VAR) Gpt_HwChannelConfig;
    } Gpt_ChannelConfigType;
```

Table 3-65. Structure Gpt_ChannelConfigType member description

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

3.5.4.2 Structure Gpt_ConfigType

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

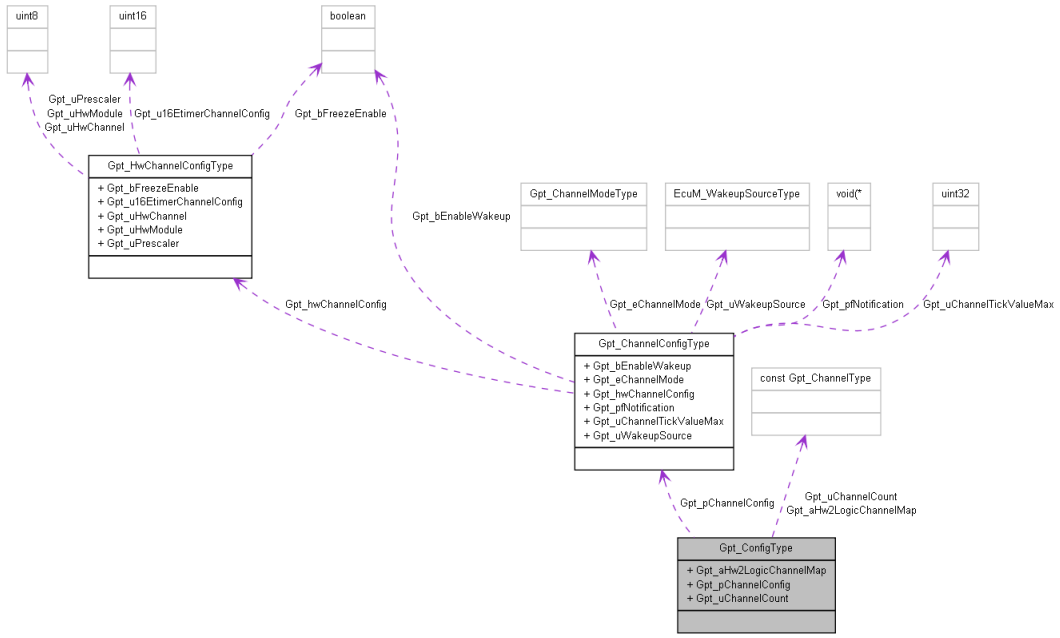

Figure 3-2. Struct Gpt_ConfigType

Implements: Gpt_ConfigType_structure

Declaration:

```
typedef struct
{
    CONST(Gpt_ChannelType, GPT_CONST) Gpt_uChannelCount;
    CONST(Gpt_ChannelConfigType, GPT_CONST) (*Gpt_pChannelConfig)[];
    CONST(Gpt_ChannelType, GPT_CONST) Gpt_aHw2LogicChannelMap[GPT_CHANNEL_IDX_NUM];
} Gpt_ConfigType;
```

Table 3-66. 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_aHw2LogicChannelMap	Hardware to logic channel map table.

3.5.4.3 Structure Gpt_HwChannelConfigType

Low level configuration for channel specific parameters.

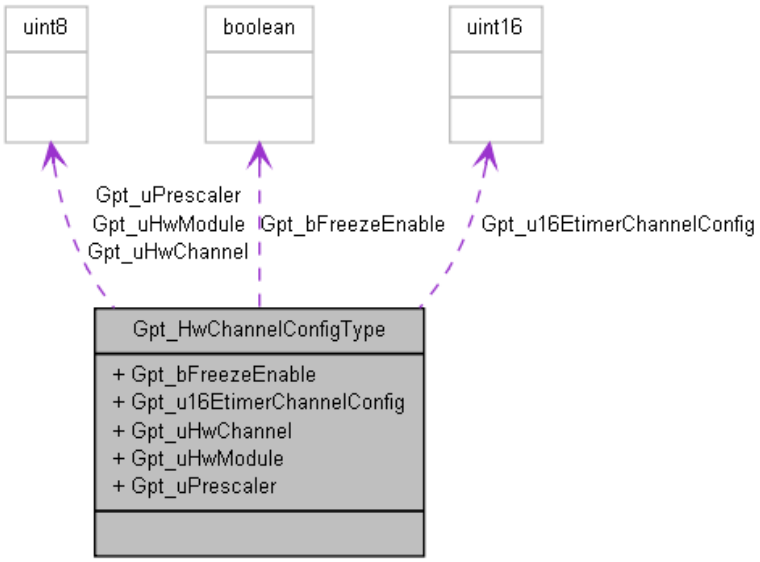

Figure 3-3. Struct Gpt_HwChannelConfigType

Declaration:

```
typedef struct
{
    VAR(uint8, GPT_VAR) Gpt_u8HwChannel;
    VAR(uint8, GPT_VAR) Gpt_u8HwModule;
    VAR(boolean, GPT_VAR) Gpt_bFreezeEnable;
    VAR(Gpt_PrescalerType, GPT_VAR) Gpt_uPrescaler;
    VAR(Gpt_PrescalerType, GPT_VAR) Gpt_uRtcClockSource;
#if(GPT_SET_CLOCK_MODE == STD_ON)
    VAR(Gpt_PrescalerType, GPT_VAR) Gpt_uStmClockSource;
#endif /* GPT_DUAL_CLOCK_MODE */
#if(GPT_SET_CLOCK_MODE == STD_ON)
    VAR(Gpt_PrescalerType, GPT_VAR) Gpt_uPrescaler_Alternate;
#endif /* GPT_DUAL_CLOCK_MODE */
} Gpt_HwChannelConfigType;
```

Table 3-67. Structure Gpt_HwChannelConfigType member description

Member	Description
Gpt_u8HwChannel	GPT hw channel ID
Gpt_u8HwModule	GPT hw module used
Gpt_bFreezeEnable	Channel freeze enable
Gpt_uPrescaler	Channel prescaler value
Gpt_uRtcClockSource	Channel RTC Clock Source selection
Gpt_uStmClockSource	Channel STM Clock Source selection
Gpt_uPrescaler_Alternate	Channel prescaler Alternate value

3.5.4.4 Structure Gpt_ChannelInfoType

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

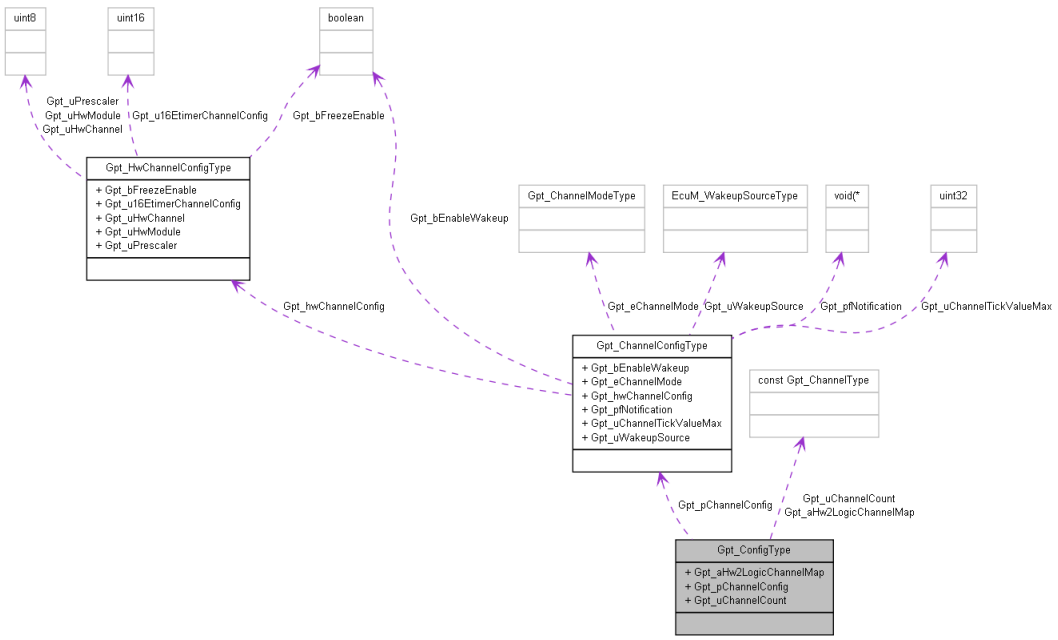

Figure 3-4. Struct Gpt_ChannelInfoType

Implements: Gpt_ChannelInfoType_structure

Declaration:

```
typedef struct
{
    VAR(Gpt_ChannelStatusType,GPT_VAR) eChannelStatus;
    #if (GPT_ENABLE_DISABLE_NOTIFICATION_API == STD_ON);
       VAR(boolean,GPT_VAR) bNotificationEnabled;
    #endif
    #if((GPT_WAKEUP_FUNCTIONALITY_API == STD_ON) &&
        (GPT_REPORT_WAKEUP_SOURCE == STD_ON))
      VAR(boolean,GPT_VAR) bWakeupEnabled;
      VAR(boolean,GPT_VAR) bWakeupGenerated;
    #endif
} Gpt_ChannelInfoType;
```

Table 3-68. Structure Gpt_ChannelInfoType member description

Member	Description
eChannelStatus	GPT channel runtime status.
bNotificationEnabled	GPT channel has notification enabled at runtime.
bWakeupEnabled	GPT channel has wakeup enabled at runtime.
bWakeupGenerated	GPT channel executed wakeup.

3.5.4.5 Structure Gpt_HwChannelInfoType

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

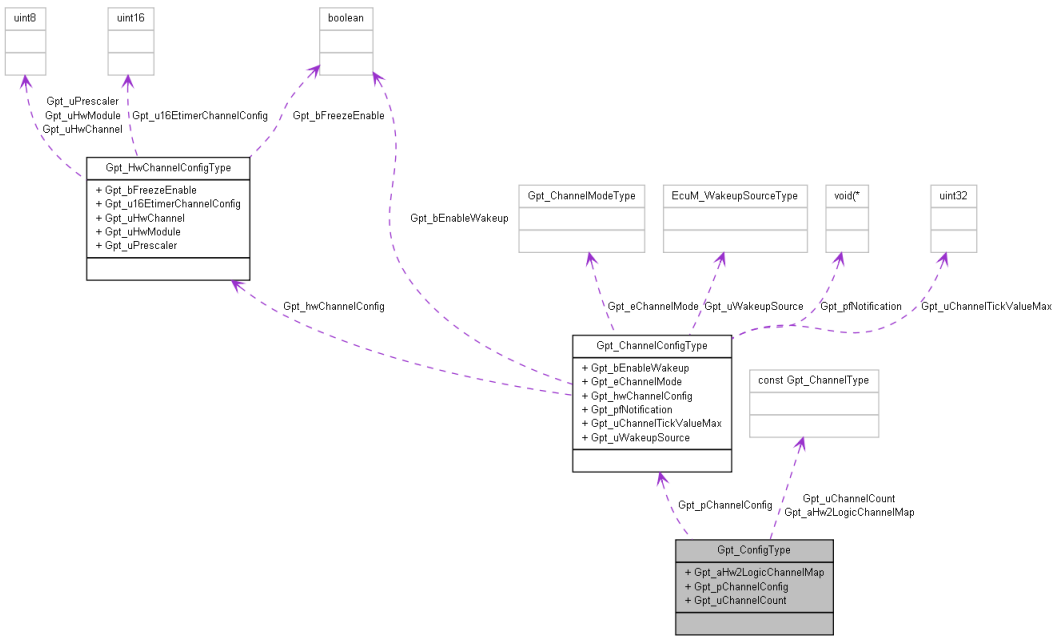

Figure 3-5. Struct Gpt_HwChannelInfoType

Implements: Gpt_HwChannelInfoType

Declaration:

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

Table 3-69. Structure Gpt_HwChannelInfoType member description

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

3.5.5 Types Reference

Types supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.2 Rev0002 .

3.5.5.1 Typedef Gpt_ChannelType

Gpt channel ID data type.

Implements: Gpt_ChannelType_typedef

Type: uint8

3.5.5.2 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: void (*);

3.5.5.3 Typedef Gpt_ValueType

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

Implements: Gpt_ValueType_typedef

Type: uint32

3.5.5.4 Typedef Gpt_PrescalerType

Gpt prescaler data type.

Type: uint8

3.5.6 Variables Reference

Variables supported by the driver are as per AUTOSAR GPT Driver software specification Version 4.2 Rev0002 .

3.6 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 :

- IMPLEMENTATION_CONFIG_VARIANT
- GptConfigurationOfOptApiServices
- GptNonAUTOSAR
- GptDriverConfiguration
- CommonPublishedInformation
- GptChannelConfigSet

Table 4-1. Revision table

Revision	Date
4.2.0	2011-11-02

4.2 Form IMPLEMENTATION_CONFIG_VARIANT

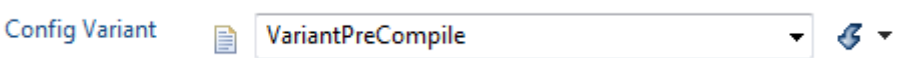

Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT form.

Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description

Property	Value
Label	Config Variant
Default	VariantPostBuild

Table continues on the next page...

Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description (continued)

Property	Value
Range	VariantPostBuild VariantPreCompile

4.3 Form GptDemEventParameterRefs

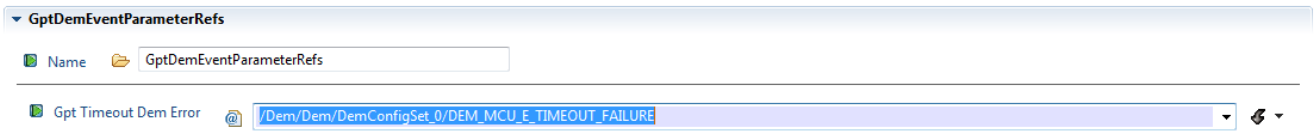

Figure 4-2. Tresos Plugin snapshot for GptDemEventParameterRefs form.

Table 4-3. Attribute GptDemEventParameterRefs detailed description

Property	Value
Name	GptDemEventParameterRefs
GPT_E_TIMEOUT	/Dem/Dem/DemConfigSet_0

4.4 Form GptConfigurationOfOptApiServices

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

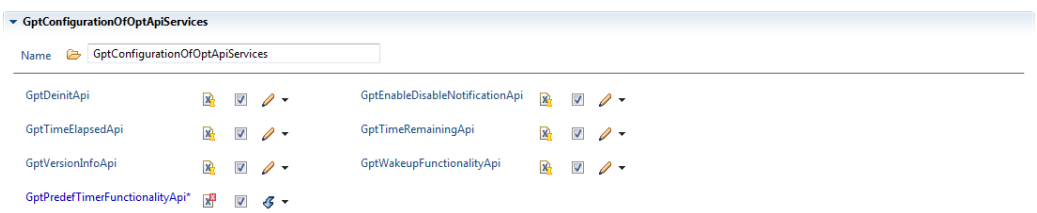

Figure 4-3. Tresos Plugin snapshot for GptConfigurationOfOptApiServices form.

4.4.1 GptDeinitApi (GptConfigurationOfOptApiServices)

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

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.4.2 GptEnableDisableNotificationApi (GptConfigurationOfOptApiServices)

Adds / removes the services Gpt_EnableNotification() and Gpt_DisableNotification from the code.

Table 4-5. Attribute GptEnableDisableNotificationApi (GptConfigurationOfOptApiServices) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.4.3 GptTimeElapsedApi (GptConfigurationOfOptApiServices)

Adds / removes the service Gpt_GetTimeElapsed() from the code.

Table 4-6. Attribute GptTimeElapsedApi (GptConfigurationOfOptApiServices) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.4.4 GptTimeRemainingApi (GptConfigurationOfOptApiServices)

Adds / removes the service Gpt_GetTimeRemaining() from the code.

Table 4-7. Attribute GptTimeRemainingApi (GptConfigurationOfOptApiServices) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.4.5 GptVersionInfoApi (GptConfigurationOfOptApiServices)

Adds / removes the service Gpt_GetVersionInfo() from the code.

Table 4-8. Attribute GptVersionInfoApi (GptConfigurationOfOptApiServices) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.4.6 GptWakeupFunctionalityApi (GptConfigurationOfOptApiServices)

Adds / removes the services Gpt_SetMode(), Gpt_EnableWakeup() Gpt_DisableWakeup() and Gpt_Cbk_CheckWakeup() from the code.

Table 4-9. Attribute GptWakeupFunctionalityApi (GptConfigurationOfOptApiServices) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.4.7 GptPredefTimerFunctionalityApi (GptConfigurationOfOptApiServices)

Adds / removes the services GptPredefTimer100us32bitEnable(), GptPredefTimer1usEnablingGrade() from the code.

Table 4-10. Attribute GptPredefTimerFunctionalityApi (GptConfigurationOfOptApiServices) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.5 Form GptNonAUTOSAR

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

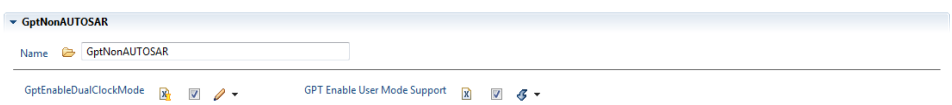

Figure 4-4. Tresos Plugin snapshot for GptNonAUTOSAR form.

4.5.1 GptEnableDualClockMode (GptNonAUTOSAR)

Table 4-11. Attribute GptEnableDualClockMode (GptNonAUTOSAR) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.5.2 GptEnableUserModeSupport (GptNonAUTOSAR)

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

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.6 Form GptDriverConfiguration

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

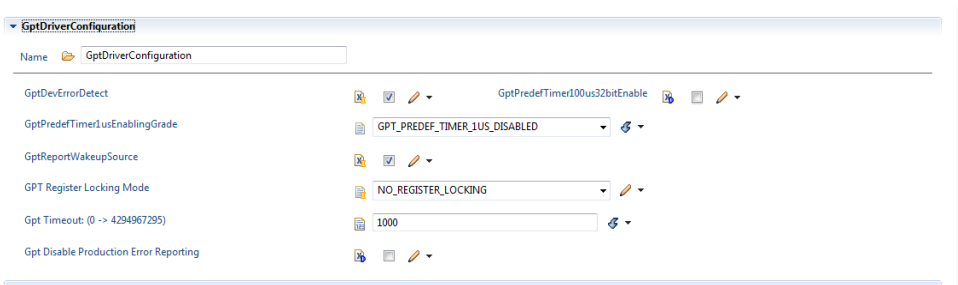

Figure 4-5. Tresos Plugin snapshot for GptDriverConfiguration form.

4.6.1 GptDevErrorDetect (GptDriverConfiguration)

Enables/Disables development error detection.

Table 4-13. Attribute GptDevErrorDetect (GptDriverConfiguration) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.6.2 GptReportWakeupSource (GptDriverConfiguration)

Enables/Disables wakeup source reporting.

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

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Default	true

4.6.3 GptRegisterLockingMode (GptDriverConfiguration)

Enables/Disables register locking from user/supervisor mode

Table 4-15. Attribute GptRegisterLockingMode (GptDriverConfiguration) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Default	NO_REGISTER_LOCKING

4.6.4 GptPredefTimer100us32bitEnable (GptDriverConfiguration)

Enables/Disables register locking from user/supervisor mode

Table 4-16. Attribute GptPredefTimer100us32bitEnable (GptDriverConfiguration) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.6.5 GptPredefTimer1usEnablingGrade (GptDriverConfiguration)

Enables/Disables register locking from user/supervisor mode

Table 4-17. Attribute GptPredefTimer1usEnablingGrade (GptDriverConfiguration) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Default	NO_REGISTER_LOCKING

4.6.6 GptTimeout (GptDriverConfiguration)

Enables/Disables register locking from user/supervisor mode

Table 4-18. Attribute GptTimeout (GptDriverConfiguration) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Default	NO_REGISTER_LOCKING

4.6.7 GptDisableDemReportErrorStatus (GptDriverConfiguration)

Enables/Disables register locking from user/supervisor mode

Table 4-19. Attribute GptDisableDemReportErrorStatus (GptDriverConfiguration) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.7 Form GptHwConfiguration

This container contains list of available interrupt sources supported by hardware. This tab also enables or disables interrupts for channels which are used in application.

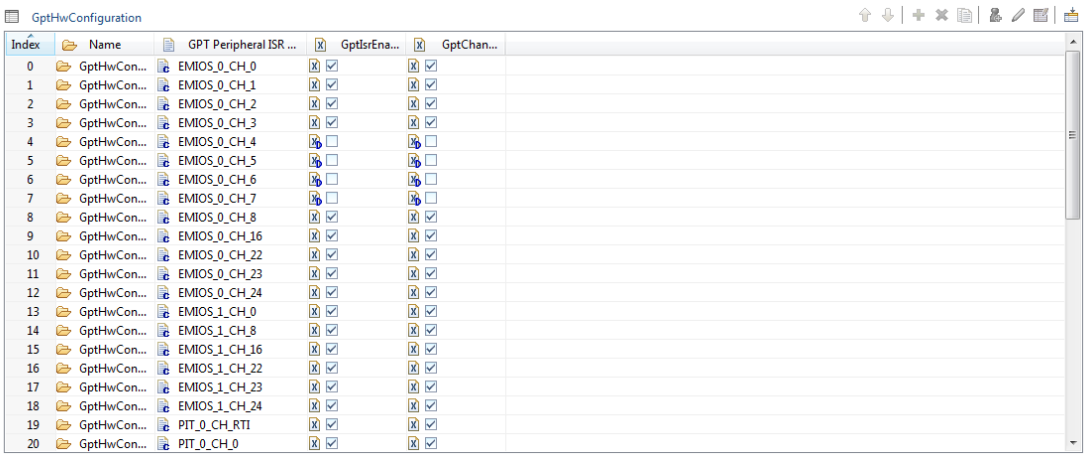

Figure 4-6. Tresos Plugin snapshot for GptHwConfiguration form.

4.7.1 Form GptHwInterruptChannel

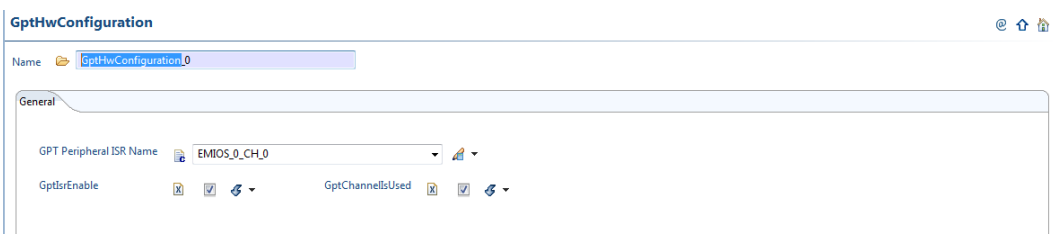

Figure 4-7. Tresos Plugin snapshot for GptHwInterruptChannel form.

4.7.1.1 GptIsrHwId (GptHwInterruptChannel)

Vendor specific: Selects the physical GPT Channel.

Table 4-20. Attribute GptHwChannel (GptHwInterruptChannel) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Range	STM_x_CH_y EMIOS_x_CH_y PIT_x_CH_y RTC_0_CH_0

4.7.1.2 GptIsrEnable (GptHwInterruptChannel)

Vendor specific: check ISr is enable or not.

Table 4-21. Attribute GptIsrEnable (GptHwInterruptChannel) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.7.1.3 GptChannelIsUsed (GptHwInterruptChannel)

Vendor specific: Check channel is used or not.

Table 4-22. Attribute GptChannelIsUsed (GptHwInterruptChannel) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.8 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.

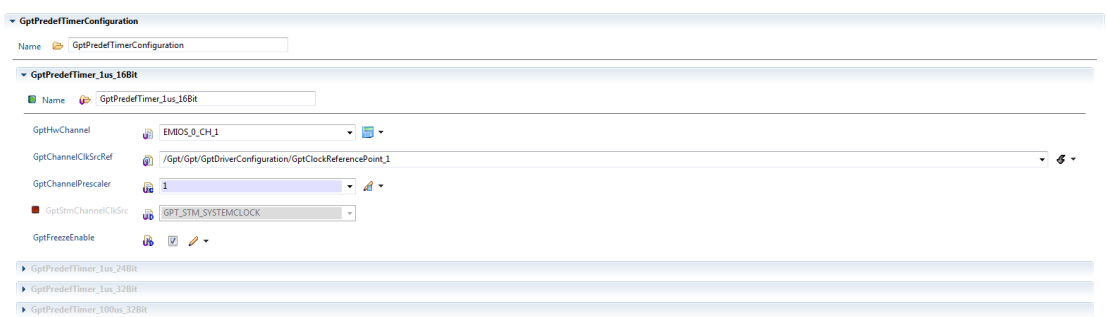

Figure 4-8. Tresos Plugin snapshot for GptPredefTimerConfiguration form.

4.9 Form GptClockReferencePoint

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

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

4.9.1 GptClockReference (GptClockReferencePoint)

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

Table 4-23. Attribute GptClockReference (GptClockReferencePoint) detailed description

Property	Value
Type	REFERENCE
Origin	AUTOSAR_ECUC

4.10 Form CommonPublishedInformation

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

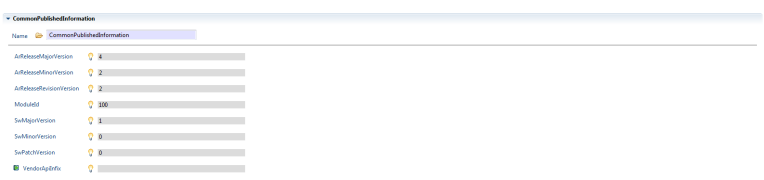

Figure 4-10. Tresos Plugin snapshot for CommonPublishedInformation form.

4.10.1 ArReleaseMajorVersion (CommonPublishedInformation)

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

Table 4-24. 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

4.10.2 ArReleaseMinorVersion (CommonPublishedInformation)

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

Table 4-25. Attribute ArReleaseMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

4.10.3 ArReleaseRevisionVersion (CommonPublishedInformation)

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

Table 4-26. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Release Revision Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

4.10.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

Table 4-27. Attribute ModuleId (CommonPublishedInformation) detailed description

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

4.10.5 SwMajorVersion (CommonPublishedInformation)

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

Table 4-28. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false

Table continues on the next page...

Table 4-28. Attribute SwMajorVersion (CommonPublishedInformation) detailed description (continued)

Property	Value
Default	1
Invalid	Range >=1 <=1

4.10.6 SwMinorVersion (CommonPublishedInformation)

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

Table 4-29. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >=0 <=0

4.10.7 SwPatchVersion (CommonPublishedInformation)

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

Table 4-30. Attribute SwPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Patch Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >=0 <=0

4.10.8 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 4-31. Attribute VendorApiInfix (CommonPublishedInformation) detailed description

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

4.10.9 VendorId (CommonPublishedInformation)

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

Table 4-32. Attribute VendorId (CommonPublishedInformation) detailed description

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

4.11 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.

4.11.1 Form GptChannelConfiguration

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

Is included by form : Form GptChannelConfigSet

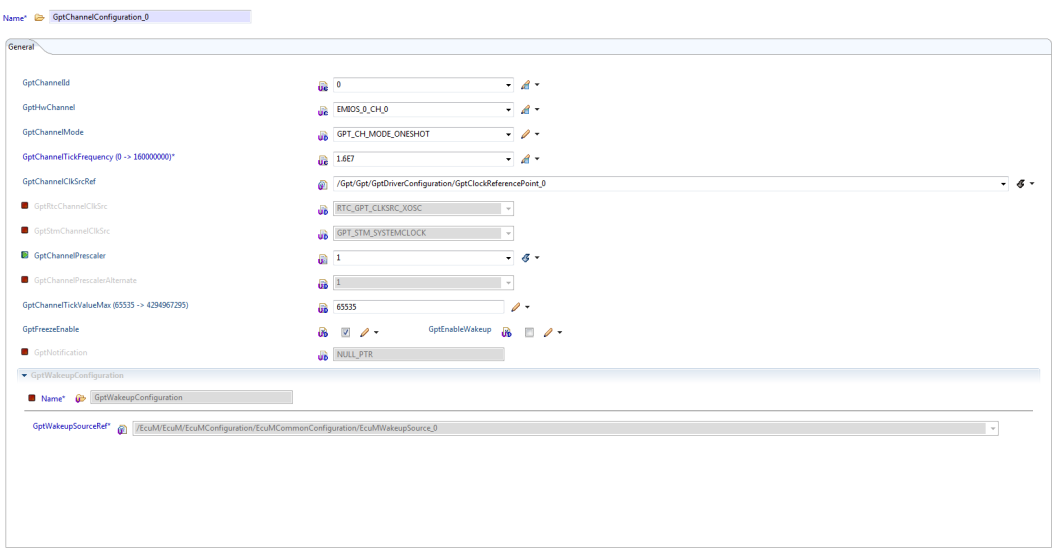

Figure 4-11. Tresos Plugin snapshot for GptChannelConfiguration form.

4.11.1.1 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 4-33. Attribute GptChannelId (GptChannelConfiguration) detailed description

Property	Value
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	true
Invalid	Range <=4294967295 >=0

4.11.1.2 GptHwChannel (GptChannelConfiguration)

Vendor specific: Selects the physical GPT Channel.

Table 4-34. Attribute GptHwChannel (GptChannelConfiguration) detailed description

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.11.1.3 GptChannelMode (GptChannelConfiguration)

Specifies the behaviour of the timerchannel after the timeout has expired

Table 4-35. 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

4.11.1.4 GptChannelTickFrequency (GptChannelConfiguration)

Specifies the tick frequency of the timer channel in Hz.

Table 4-36. Attribute GptChannelTickFrequency (GptChannelConfiguration) detailed description

Property	Value
Type	FLOAT
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	1.0
Invalid	Range >0 <=160000000

4.11.1.5 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 4294967295 for the RTC, PIT and STM channels (coresponding to the 32 bits counter resolution) and 65535 for the EMIOS channels (corresponding to the 16 bits counter resolution)!

Table 4-37. Attribute GptChannelTickValueMax (GptChannelConfiguration) detailed description

Property	Value
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	65535

4.11.1.6 GptChannelClkSrcRef (GptChannelConfiguration)

Reference to the GptClockReferencePoint from which the channel clock is derived.

Table 4-38. Attribute GptChannelClkSrcRef (GptChannelConfiguration) detailed description

Property	Value
Type	REFERENCE
Origin	AUTOSAR_ECUC

4.11.1.7 GptRtcChannelClkSrc (GptChannelConfiguration)

Select clock source for RTC module

Table 4-39. Attribute GptRtcChannelClkSrc (GptChannelConfiguration) detailed description

Property	Value
Type	ENUMERATION
Symbolic Name	false
Default	RTC_GPT_CLKSRC_XOSC
Range	RTC_GPT_CLKSRC_XOSC RTC_GPT_CLKSRC_FIRC RTC_GPT_CLKSRC_SIRC RTC_GPT_CLKSRC_SXOSC

4.11.1.8 GptStmChannelClkSrc (GptChannelConfiguration)

Select clock source for RTC module

Table 4-40. Attribute GptStmChannelClkSrc (GptChannelConfiguration) detailed description

Property	Value
Type	ENUMERATION
Symbolic Name	false
Default	GPT_STM_SYSTEMCLOCK
Range	GPT_STM_SYSTEMCLOCK GPT_STM_FXOSCCLOCK

4.11.1.9 GptChannelPrescaler (GptChannelConfiguration)

Vendor specific: The GPT module specific clock prescaler value.

Table 4-41. Attribute GptChannelPrescaler (GptChannelConfiguration) detailed description

Property	Value
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range >=1 <=256

4.11.1.10 GptChannelPrescalerAlternate (GptChannelConfiguration)

Vendor specific: The GPT module specific clock prescaler value.

Table 4-42. Attribute GptChannelPrescalerAlternate (GptChannelConfiguration) detailed description

Property	Value
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1

Table continues on the next page...

Table 4-42. Attribute GptChannelPrescalerAlternate (GptChannelConfiguration) detailed description (continued)

Property	Value
Invalid	Range >=1 <=256

4.11.1.11 GptFreezeEnable (GptChannelConfiguration)

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

NOTE

All channels in a same module timer must be either disabled or enabled simultaneously.

Table 4-43. Attribute GptFreezeEnable (GptChannelConfiguration) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.11.1.12 GptEnableWakeup (GptChannelConfiguration)

Enables wakeup capability of CPU for a channel.

Table 4-44. Attribute GptEnableWakeup (GptChannelConfiguration) detailed description

Property	Value
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

4.11.1.13 GptNotification (GptChannelConfiguration)

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

Table 4-45. Attribute GptNotification (GptChannelConfiguration) detailed description

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

4.11.1.14 GptWakeupSourceRef(GptChannelConfiguration)

In case the wakeup-capability is true this value is transmitted to the Ecu State Manager. Implementation Type: reference to EcuM_WakeupSourceType

Table 4-46. Attribute GptWakeupSourceRef(GptChannelConfiguration) detailed description

Property	Value
Type	Reference
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number UM35GPTASR4.2 Rev0002 R1.0.0
Revision 5.0.0