User Manual

for S32K1_S32M24X GPT Driver

Document Number: UM2GPTASRR21-11 Rev0000R2.0.0 Rev. 1.0

Chapter 1

Revision History

Revision	Date	Author	Description
1.0	04.08.2023	NXP RTD Team	S32K1_S32M24X Real-Time Drivers AUTOSAR 4.4 & R21-11 Version 2.0.0

Chapter 2

Introduction

This User Manual describes NXP Semiconductor AUTOSAR GPT. 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:

- s32k116_qfn32
- s32k116_lqfp48
- s32k118_lqfp48
- s32k118_lqfp64
- s32k142_lqfp48
- s32k142_lqfp64
- s32k142_lqfp100
- s32k142w_lqfp48
- s32k142w_lqfp64
- s32k144_lqfp48

- s32k144_lqfp64 / MWCT1014S_lqfp64
- s32k144_lqfp100 / MWCT1014S_lqfp100
- s32k144_mapbga100
- s32k144w_lqfp48
- s32k144w_lqfp64
- s32k146_lqfp64
- s32k146_lqfp100 / MWCT1015S_lqfp100
- s32k146_mapbga100 / MWCT1015S_mapbga100
- s32k146_lqfp144
- s32k148_lqfp100
- s32k148_mapbga100 / MWCT1016S_mapbga100
- s32k148_lqfp144
- s32k148_lqfp176
- s32m241_lqfp64
- s32m242_lqfp64
- s32m243_lqfp64
- s32m244_lqfp64

All of the above microcontroller devices are collectively named as S32K1_S32M24X. Note: MWCT part numbers contain NXP confidential IP for Qi Wireless Power

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** style: Used for important terms, notes and warnings.
- *Italic* style: 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.

Warning

This is a warning

2.4 Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler
BSMI	Basic Software Make file Interface
GPT	General Purpose Timer
C/CPP	C and C++ Source Code
CS	Chip Select
CTU	Cross Trigger Unit
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DMA	Direct Memory Access
ECU	Electronic Control Unit
FIFO	First In First Out
LSB	Least Significant Bit
MCU	Micro Controller Unit
MIDE	Multi Integrated Development Environment
MSB	Most Significant Bit
N/A	Not Applicable
RAM	Random Access Memory
SIU	Systems Integration Unit
SWS	Software Specification
VLE	Variable Length Encoding
XML	Extensible Markup Language

2.5 Reference List

#	Title	Version
1	Specification of GPT Driver	AUTOSAR Release R21-11
2	S32K1xx Series Reference Manual	Rev. 14, 09/2021
3	S32M24x Reference Manual	Rev. 2 Draft A, 05/2023
4	S32K116 Mask Set Errata for Mask 0N96V	Rev. 22/OCT/2021
5	S32K118 Mask Set Errata for Mask 0N97V	Rev. 22/OCT/2021
6	S32K142 Mask Set Errata for Mask 0N33V	Rev. 22/OCT/2021
7	S32K144 Mask Set Errata for Mask 0N57U	Rev. 22/OCT/2021
8	S32K144W Mask Set Errata for Mask 0P64A	Rev. 22/OCT/2021
9	S32K146 Mask Set Errata for Mask 0N73V	Rev. 22/OCT/2021
10	S32K148 Mask Set Errata for Mask 0N20V	Rev. 22/OCT/2021
11	S32M244 Mask Set Errata for Mask P64A+P73G	Rev. 0
12	S32M242 Mask Set Errata for Mask N33V+P73G	Rev. 0, 6/2023
13	S32K1xx Data Sheet	Rev. 14, 08/2021
14	S32M2xx Data Sheet	Rev. 3 DraftA, 05/2023

Chapter 3

Driver

3.1 Requirements

- Requirements for this driver are detailed in the RTD GPT AUTOSAR Release 4.4.0.
- Driver Software Specification document (See S32K1xx Series Reference Manual, Rev. 14, 09/2021).

3.2 Driver Design Summary

The RTD driver assures reentrancy (single core execution) for the APIs based on the following assumptions:

- The "called-again" API is for a different resource (hardware/logic channel);
- Common variables/registers accessed with "rmw" are guarded by Exclusive Areas which need to be correctly implemented in RTE on user side;

The GPT Driver implements the following channels on S32K1xx peripherals.

The table provides information regarding the Timer channels available for the various derivatives across different packages in S32K1xx family. This table lists only the supported packages by GPT driver.

RTC module features :

- 32-bit counter
- SRTC interrupt with interrupt enable.
- Selectable counter clock sources
- Counter runs in all modes of operation.

LPIT timer module features :

- 32-bit counters per module
- Four channel.
- external triggers (triggers from outside the LPIT module)
- or internal triggers (triggers from other timer channels inside the LPIT module).
- Independent timeout periods for each timer
- Independent interrupt source.

LPTMR timer module features :

- One channel 16-bit time counter.
- Configurable clock source for prescaler/glitch filter
- Configurable input source for pulse counter

FTM timer module features :

- One 16-bit up counter with 16-bit
- Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128
- Independent interrupt source for each channel.

3.3 Hardware Resources

#	Hardware IP	Description
1	LPTMR	Low Power Timer
2	LPIT	Low Power Interrupt Timer
3	SRTC	Real Time Clock
4	FTM	FlexTimer

3.4 Deviations 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.

Deviations Status Column Description

Term	Definition
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.

Requirement	Status	Description	Notes
SWS_Gpt_00261	N/S	Gpt_Irq.c shall include Gpt.h for the prototype declaration of the notification functions.	Rejection reason: Gpt_Irq.c is not needed. Autosar specific interrupt behaviour is implemented using a normal function placed in the Gpt.c file.
SWS_Gpt_00278	N/S	Module - Header File - Imported Type - EcuM_flex - EcuM.h - EcuM_WakeupSourceType - Std_Types - StandardTypes.h - Std_ReturnType - StandardTypes.h - Std_VersionInfoType -	Rejection reason: No production errors needed for current development.
SWS_Gpt_00381	N/S	These requirements are not applicable to this specification.	Not a requirement
ECUC_Gpt_00235	N/S	Container Name - GptWakeupConfiguration - Description - Function pointer to callback function (for wakeup notification). - Configuration Parameters -	Rejection reason: Wrong Description: Function pointer to callback function (for non-wakeup notification).It shall relate to wakeup configuration.
SWS_Gpt_CONSTR_00001	N/S	DRAFT: The ECUC partitions referenced by GptKernelEcucPartitionRef shall be a subset of the ECUC partitions referenced by GptEcucPartitionRef.()	Type IV Autosar multicore not implemented for current module. AAI-445; Agree that each module can reject the Autosar Standard requirement

Requirement	Status	Description	Notes
ECUC_Gpt_00338	N/S	Name - GptKernelEcuc↩ PartitionRef - Parent Container - GptDriverConfiguration - Description - Maps the GPT kernel to zero or one ECUC partitions to assign the driver kernel to a certain core. The ECUC partition referenced is a subset of the ECUC partitions where the GPT driver is mapped to. Note: The kernel reference shall not be set in case the GPT driver is implemented without a kernel (refer to definition of GptEcucPartitionRef).Tags: atp.Status=draft - Multiplicity - 0..1 - Type - Reference to [EcucPartition] - Post-Build Variant Multiplicity - true - Post-Build Variant Value - true - Multiplicity Configuration Class - Pre-compile time - X - All Variants - Link time - – - - Post-build time - – - - Value Configuration Class - Pre-compile time - X - All Variants - Link time - – - - Post-build time - – - - Scope / Dependency - scope: ECU -	Type IV Autosar multicore not implemented for current module. AAI-445; Agree that each module can reject the Autosar Standard requirement.

3.5 Driver Limitations

The GPT driver software have some following limitations for RTD S32K1xx :

- Multicore support is not available.
- Predefined Timer functionality is supported only with FTM Ip and because the FTM time reference is a 16-bit counter only GptPredefTimer_1us_16Bit is available.

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.

3.7 Runtime errors

The driver generates the following DEM errors at runtime.

Function	Error Code	Condition triggering the error
Gpt_ValidateChannelStatus()	GPT_E_BUSY	API service called when timer channel is still busy (running)
Gpt_ValidateMode()	GPT_E_MODE	API service called when driver is in wrong mode

3.8 Symbolic Names Disclaimer

All containers having symbolicNameValue set to TRUE in the AUTOSAR schema will generate defines like:

#define <Mip>Conf_<Container_ShortName>_<Container_ID>

For this reason it is forbidden to duplicate the names of such containers across the RTD configurations 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 driver. All the parameters are described below.

- Module Gpt
  - Container GptChannelConfigSet
    - Container GptChannelConfiguration
      - Parameter GptChannelId
      - Parameter GptHwIp
      - Parameter GptChannelMode
      - Parameter GptChannelTickFrequency
      - Parameter GptChannelTickValueMax
      - Parameter GptEnableWakeup
      - Parameter GptNotification
      - Reference GptChannelEcucPartitionRef
      - Reference GptModuleRef
      - Reference GptChannelClkSrcRef
      - Container GptWakeupConfiguration
      - Reference GptWakeupSourceRef
    - Container GptFtm
      - Parameter GptFtmModule
      - Parameter GptFtmPrescaler
      - Parameter GptFtmAlternatePrescaler
      - Parameter GptFtmChannelClkSrc
      - Parameter FtmFreezeEnable
      - Parameter GptFtmCountingMode
      - Container GptFtmChannels
      - Parameter GptFtmChannel
      - Parameter FtmAbsoluteCounting
    - Container GptSRtc
      - Parameter GptSRtcModule
      - Parameter GptSRtcChannelClkSrc
    - Container GptLptmr
      - Parameter GptLptmrModule

- Parameter GptLptmrChannelClkSrc
- Parameter GptLptmrPrescaler
- Parameter GptLptmrAlternatePrescaler
- Parameter GptLptmrPrescalerEnable

* Container GptLpit
  - Parameter GptLpitModule
  - Parameter LpitFreezeEnable
  - Parameter LpitDozeEnable
  - Container GptLpitChannels
  - Parameter GptLpitChannel
  - Parameter LPitExternalTrigger
  - Parameter LPitReloadOnTrigger
  - Parameter LPitStopOnInterrupt
  - Parameter LPitStartOnTrigger
  - Parameter ChainMode
  - Parameter LPitTriggerChannels

– Container GptHwConfiguration
  * Parameter GptIsrHwId
  * Parameter GptIsrEnable
  * Parameter GptChannelIsUsed

– Container GptConfigurationOfOptApiServices
  * Parameter GptDeinitApi
  * Parameter GptEnableDisableNotificationApi
  * Parameter GptTimeElapsedApi
  * Parameter GptTimeRemainingApi
  * Parameter GptVersionInfoApi
  * Parameter GptWakeupFunctionalityApi
  * Parameter GptPredefTimerFunctionalityApi

– Container GptAutosarExt
  * Parameter GptEnableDualClockMode
  * Parameter GptChangeNextTimeoutValueApi
  * Parameter GptEnableUserModeSupport
  * Parameter ChainModeApi
  * Parameter GptStandbyWakeupSupport
  * Parameter GptEnableTriggers

– Container GptDriverConfiguration
  * Parameter GptDevErrorDetect
  * Parameter GptPredefTimer100us32bitEnable
  * Parameter GptMulticoreSupport
  * Parameter GptPredefTimer1usEnablingGrade

    - Parameter GptTimeoutMethod
    - Parameter GptTimeoutDuration
    - Parameter GptReportWakeupSource
    - Reference GptEcucPartitionRef
    - Reference GptKernelEcucPartitionRef
    - Container GptClockReferencePoint
        - Reference GptClockReference
- Container GptPredefTimerConfiguration
    - Container GptPredefTimer_1us_16Bit
        - Parameter GptHwChannel
        - Parameter GptFtmChannelClkSrc
        - Parameter GptChannelPrescaler
        - Parameter GptFreezeEnable
        - Reference GptChannelClkSrcRef
    - Container GptPredefTimer_1us_24Bit
        - Parameter GptHwChannel
        - Parameter GptChannelPrescaler
        - Parameter GptFreezeEnable
        - Reference GptChannelClkSrcRef
    - Container GptPredefTimer_1us_32Bit
        - Parameter GptHwChannel
        - Parameter GptChannelPrescaler
        - Parameter GptFreezeEnable
        - Reference GptChannelClkSrcRef
    - Container GptPredefTimer_100us_32Bit
        - Parameter GptHwChannel
        - Parameter GptChannelPrescaler
        - Parameter GptFreezeEnable
        - Reference GptChannelClkSrcRef
- Container CommonPublishedInformation
    - Parameter ArReleaseMajorVersion
    - Parameter ArReleaseMinorVersion
    - Parameter ArReleaseRevisionVersion
    - Parameter ModuleId
    - Parameter SwMajorVersion
    - Parameter SwMinorVersion
    - Parameter SwPatchVersion
    - Parameter VendorApiInfix
    - Parameter VendorId

4.1 Module Gpt

Configuration of the Gpt (General Purpose Timer) module.

Included containers:

- GptChannelConfigSet
- GptHwConfiguration
- GptConfigurationOfOptApiServices
- GptAutosarExt
- GptDriverConfiguration
- GptPredefTimerConfiguration
- CommonPublishedInformation

Property	Value
type	ECUC-MODULE-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantSupport	true
supportedConfigVariants	VARIANT-POST-BUILD, VARIANT-PRE-COMPILE

4.2 Container 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 subcontainers:

- GptChannelConfiguration
- GptFtm
- GptSRtc
- GptLptmr
- GptLpit

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.3 Container GptChannelConfiguration

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

Included subcontainers:

- GptWakeupConfiguration

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.4 Parameter GptChannelId

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	true
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	0
max	4294967295
min	0

4.5 Parameter GptHwIp

Vendor specific: Selects the physical GPT Channel.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	FTM
literals	['FTM', 'LPIT', 'LPTMR', 'SRTC']

4.6 Parameter GptChannelMode

Specifies the behaviour of the timerchannel after the timeout has expired

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	GPT_CH_MODE_ONESHOT
literals	['GPT_CH_MODE_CONTINUOUS', 'GPT_CH_MODE_ONESHOT']

4.7 Parameter GptChannelTickFrequency

EN: Specifies the tick frequency of the timer channel in Hz.

Property	Value
type	ECUC-FLOAT-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0.0
max	1.6E8
min	0.0

4.8 Parameter GptChannelTickValueMax

Maximum value in ticks, the timer channel is able to count. With the next tick, the timer rolls over to zero.

It is mandatory to set 65535 for the FTM channels (corresponding to the 16 bits counter resolution)!

It is mandatory to set 65535 for the LPtimer channels (corresponding to the 16 bits counter resolution)!

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	65535
max	4294967295
min	65535

4.9 Parameter GptEnableWakeup

Enables wakeup capability of CPU for a channel.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.10 Parameter GptNotification

Function pointer to callback function(for non-wakeup notification).

The field is editable only if the switch GptEnableDisableNotificationApi is true.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NULL_PTR

4.11 Reference GptChannelEcucPartitionRef

Maps a GPT channel to zero or one ECUC partition to limit the access to this channel group. The ECUC partitions referenced are a subset of the ECUC partitions where the GPT driver is mapped to.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0

Property	Value
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.12 Reference GptModuleRef

Maps a GPT channel to zero or one ECUC partition to limit the access to this channel group. The ECUC partitions referenced are a subset of the ECUC partitions where the GPT driver is mapped to.

Property	Value
type	ECUC-CHOICE-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destinations	['/TS_T40D2M20I0R0/Gpt/GptChannelConfigSet/GptFtm/GptFtmChannels', '/TS_T40D2M20I0R0/Gpt/GptChannelConfigSet/GptSRtc', '/TS_T40D2↩ M20I0R0/Gpt/GptChannelConfigSet/GptLptmr', '/TS_T40D2M20I0R0/↩ Gpt/GptChannelConfigSet/GptLpit/GptLpitChannels']

4.13 Reference GptChannelClkSrcRef

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

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Gpt/GptDriverConfiguration/GptClockReferencePoint

4.14 Container GptWakeupConfiguration

This container defines the wakeup source codes reported to Ecu State Manager.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.15 Reference GptWakeupSourceRef

In case the wakeup-capability is true this value is transmitted to the Ecu State Manager.

Implementation Type: reference to EcuM_WakeupSourceType

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE

Property	Value
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	true
destination	/AUTOSAR/EcucDefs/EcuM/EcuMConfiguration/EcuMCommon↩ Configuration/EcuMWakeupSource

4.16 Container GptFtm

Configuration of a Ftm module available on the platfom.

Included subcontainers:

- GptFtmChannels

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	4
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.17 Parameter GptFtmModule

Select the physical Ftm Module.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	FTM_0
literals	['FTM_0', 'FTM_1', 'FTM_2', 'FTM_3']

4.18 Parameter GptFtmPrescaler

Vendor specific: The GPT module specific clock prescaler value.

Note with FTM:

- FTM prescaler should be 1, 2, 4, 8, 16, 32, 64 or 128.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	1
max	128
min	1

4.19 Parameter GptFtmAlternatePrescaler

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 written in register: 0

2 - Value written in register: 1

4 - Value written in register: 2

8 - Value written in register: 3

16 - Value written in register: 4

32 - Value written in register: 5

64 - Value written in register: 6

128 - Value written in register: 7

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	1
max	128
min	1

4.20 Parameter GptFtmChannelClkSrc

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.

FTM_GPT_IP_CLOCK_SOURCE_NONENone use clock for FTM?

FTM_GPT_IP_CLOCK_SOURCE_SYSTEMCLKSystem clock

FTM_GPT_IP_CLOCK_SOURCE_FIXED_FREQUENCYFixed frequency clock

FTM_GPT_IP_CLOCK_SOURCE_EXTERNALCLKExternal clock

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	FTM_GPT_IP_CLOCK_SOURCE_NONE
literals	['FTM_GPT_IP_CLOCK_SOURCE_NONE', 'FTM_GPT_IP_CLOCK_S↩ OURCE_SYSTEMCLK', 'FTM_GPT_IP_CLOCK_SOURCE_FIXED_FR↩ EQUENCY', 'FTM_GPT_IP_CLOCK_SOURCE_EXTERNALCLK']

4.21 Parameter FtmFreezeEnable

Enables/Disables freeze bit.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.22 Parameter GptFtmCountingMode

This parameter indicates the count direction for the whole GPT driver.

NoteThis feature is not supported in current implementation. Up-counting is always used as default

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: POST-BUILD
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	FTM_GPT_IP_MODE_UP_TIMER
literals	['FTM_GPT_IP_MODE_DOWN_TIMER', 'FTM_GPT_IP_MODE_UP↩ _TIMER']

4.23 Container GptFtmChannels

Ftm hw channels

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	8
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.24 Parameter GptFtmChannel

Selects one of the Ftm channels available on the platform.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	CH_0
literals	['CH_0', 'CH_1', 'CH_2', 'CH_3', 'CH_4', 'CH_5', 'CH_6', 'CH_7']

4.25 Parameter FtmAbsoluteCounting

Enables/Disables absolute compare value.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.26 Container GptSRtc

Configuration of a sRtc module available on the platfom.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.27 Parameter GptSRtcModule

Select the physical sRtc Module.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	SRTC_0_CH_0
literals	['SRTC_0_CH_0']

4.28 Parameter GptSRtcChannelClkSrc

Selectable counter clock sources

? CLK_SRC_OSC_32KHZ

? CLK_SRC_LPO_1KHZ

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	SRTC_IP_CLK_SRC_OSC_32KHZ
literals	['SRTC_IP_CLK_SRC_LPO_1KHZ', 'SRTC_IP_CLK_SRC_OSC_32KHZ']

4.29 Container GptLptmr

Configuration of a LPtimer module available on the platform.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.30 Parameter GptLptmrModule

Select the physical LPtimer Module.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LPTMR_0_CH_0
literals	['LPTMR_0_CH_0']

4.31 Parameter GptLptmrChannelClkSrc

Select the clock source for the Low Power Timer module for this platform.

? SIRCDIV2_CLK

? LPO1K

? RTC_CLK

? PCC_LPTMR0

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LPTMR_GPT_IP_CLOCK_SIRCDIV2
literals	['LPTMR_GPT_IP_CLOCK_SIRCDIV2', 'LPTMR_GPT_IP_CLOCK_L↩ PO1K', 'LPTMR_GPT_IP_CLOCK_RTC_CLK', 'LPTMR_GPT_IP_CL↩ OCK_PCC_LPTMR0']

4.32 Parameter GptLptmrPrescaler

Vendor specific: The GPT module specific clock prescaler value.

Configures the Prescaler in Time Counter mode, from one of 16 divisions.

2 - Value written in register is 0

4 - Value written in register is 1

8 - Value written in register is 2

16 - Value written in register is 3

32 - Value written in register is 4

64 - Value written in register is 5

128 - Value written in register is 6

256 - Value written in register is 7

512 - Value written in register is 8

1024 - Value written in register is 9

2048 - Value written in register is 10

4096 - Value written in register is 11

8192 - Value written in register is 12

16384 - Value written in register is 13

32768 - Value written in register is 14

65536 - Value written in register is 15

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	2
max	65536
min	2

4.33 Parameter GptLptmrAlternatePrescaler

Vendor specific: The GPT module specific clock prescaler value.

Configures the Prescaler in Time Counter mode, from one of 16 divisions.

2 - Value written in register is 0

4 - Value written in register is 1

8 - Value written in register is 2

16 - Value written in register is 3

32 - Value written in register is 4

64 - Value written in register is 5

128 - Value written in register is 6

256 - Value written in register is 7

512 - Value written in register is 8

1024 - Value written in register is 9

2048 - Value written in register is 10

4096 - Value written in register is 11

8192 - Value written in register is 12

16384 - Value written in register is 13

32768 - Value written in register is 14

65536 - Value written in register is 15

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	2
max	65536
min	2

4.34 Parameter GptLptmrPrescalerEnable

When PBYP is set, the selected prescaler clock in Time Counter mode or selected input source in Pulse Counter mode directly clocks the CNR.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.35 Container GptLpit

Configuration of a Lpit module available on the platfom.

Included subcontainers:

- GptLpitChannels

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	2
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.36 Parameter GptLpitModule

Select the physical Lpit Module.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	LPIT_0
literals	['LPIT_0']

4.37 Parameter LpitFreezeEnable

Enables/Disables freeze bit.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.38 Parameter LpitDozeEnable

Enables/Disables DOZE Mode.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1

Property	Value
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.39 Container GptLpitChannels

Lpit hw channels.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	4
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.40 Parameter GptLpitChannel

Selects one of the Lpit channels available on the platform.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

Property	Value
defaultValue	CH_0
literals	['CH_0', 'CH_1', 'CH_2', 'CH_3']

4.41 Parameter LPitExternalTrigger

Select between external/internal trigger sources.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.42 Parameter LPitReloadOnTrigger

LPIT TImer channel will Reload Timer at each rising edge of trigger source.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.43 Parameter LPitStopOnInterrupt

LPIT Timer Channel will stop when its interrupt occurs.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.44 Parameter LPitStartOnTrigger

LPIT Timer Channel will Start at each rising edge of Trigger Signal.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.45 Parameter ChainMode

Enables/Disables chain mode

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	false

4.46 Parameter LPitTriggerChannels

Vendor specific: LPIT External Trigger Channels Select.

Trigger Channels can be selected from 16 external sources.

0 Timer channel 0 trigger source is selectd.

1 Timer channel 1 trigger source is selectd.

.. ..

14 Timer channel 14 trigger source is selectd.

15 Timer channel 15 trigger source is selectd.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	Channel_0_Trigger_Source
literals	['Channel_0_Trigger_Source', 'Channel_1_Trigger_Source', 'Channel_2_↩ Trigger_Source', 'Channel_3_Trigger_Source', 'Channel_4_Trigger_Source', 'Channel_5_Trigger_Source', 'Channel_6_Trigger_Source', 'Channel_7_↩ Trigger_Source', 'Channel_8_Trigger_Source', 'Channel_9_Trigger_Source', 'Channel_10_Trigger_Source', 'Channel_11_Trigger_Source', 'Channel_12_↩ Trigger_Source', 'Channel_13_Trigger_Source', 'Channel_14_Trigger_Source', 'Channel_15_Trigger_Source']

4.47 Container GptHwConfiguration

List of all HW channel resources for GPT module.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	42
upperMultiplicity	42
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.48 Parameter GptIsrHwId

ID of HW interrupt resources.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	true
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	LPIT_0_CH_1
literals	['LPIT_0_CH_0', 'LPIT_0_CH_1', 'LPIT_0_CH_2', 'LPIT_0_CH_3', 'FTM_0_CH_0', 'FTM_0_CH_1', 'FTM_0_CH_2', 'FTM_0_CH_3', 'FTM_0_CH_4', 'FTM_0_CH_5', 'FTM_0_CH_6', 'FTM_0_CH_7', 'FTM_1_CH_0', 'FTM_1_CH_1', 'FTM_1_CH_2', 'FTM_1_CH_3', 'FTM_1_CH_4', 'FTM_1_CH_5', 'FTM_1_CH_6', 'FTM_1_CH_7', 'FTM_2_CH_0', 'FTM_2_CH_1', 'FTM_2_CH_2', 'FTM_2_CH_3', 'FTM_2_CH_4', 'FTM_2_CH_5', 'FTM_2_CH_6', 'FTM_2_CH_7', 'FTM_3_CH_0', 'FTM_3_CH_1', 'FTM_3_CH_2', 'FTM_3_CH_3', 'FTM_3_CH_4', 'FTM_3_CH_5', 'FTM_3_CH_6', 'FTM_3_CH_7', 'LPTMR_0_CH_0', 'SRTC_0_CH_0', 'FTM_0_PREDEF', 'FTM_1_PREDEF', 'FTM_2_PREDEF', 'FTM_3_PREDEF']

4.49 Parameter GptIsrEnable

Enable/Disable HW channels' Interrupt Sources.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.50 Parameter GptChannelIsUsed

This column configures HW channels which are going to be used.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.51 Container GptConfigurationOfOptApiServices

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

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.52 Parameter GptDeinitApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.53 Parameter GptEnableDisableNotificationApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.54 Parameter GptTimeElapsedApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.55 Parameter GptTimeRemainingApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.56 Parameter GptVersionInfoApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.57 Parameter GptWakeupFunctionalityApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.58 Parameter GptPredefTimerFunctionalityApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.59 Container GptAutosarExt

Enabling the settings of this section will configure the driver in a mode not compliant with AUTOSAR requirements.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.60 Parameter GptEnableDualClockMode

Enables prescaler settings at mode transition.true: Enabled.false: Disabled.

Note This feature is not required by Autosar.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.61 Parameter GptChangeNextTimeoutValueApi

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.62 Parameter GptEnableUserModeSupport

When this parameter is enabled, the GPT module will adapt to run from User Mode. There is no difference between User mode and Privileged mode in GPT module.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.63 Parameter ChainModeApi

Vendor specific: Enable/disable API for Chain Mode support.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.64 Parameter GptStandbyWakeupSupport

The driver shall NOT CLEAR the interrupt flag, the interrupt enable bit and also should not disable the counter, during init (Gpt_SRtc_Init()) the flag is already set.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.65 Parameter GptEnableTriggers

Enable Trigger Mode.true: Enabled.false: Disabled.

Note This feature is not required by Autosar.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.66 Container GptDriverConfiguration

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

Included subcontainers:

- GptClockReferencePoint

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.67 Parameter GptDevErrorDetect

Enables/Disables development error detection.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.68 Parameter GptPredefTimer100us32bitEnable

Enables/Disables the feature 100us/ tick

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	false

4.69 Parameter GptMulticoreSupport

Enables/Disables Multicore Support.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.70 Parameter GptPredefTimer1usEnablingGrade

Specifies the grade of enabling the GPT Predef Timers with 1us tick duration.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF

Property	Value
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	GPT_PREDEF_TIMER_1US_16BIT_ENABLED
literals	['GPT_PREDEF_TIMER_1US_16BIT_ENABLED', 'GPT_PREDEF_TIMER_1US_DISABLED']

4.71 Parameter GptTimeoutMethod

GptTimeoutMethod: Configures the timeout method.

Based on this selection a certain timeout method from OsIf will be used in the driver.

Note: If SystemTimer or CustomTimer are selected make sure the corresponding timer is enabled in OsIf General configuration.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	OSIF_COUNTER_DUMMY
literals	['OSIF_COUNTER_SYSTEM', 'OSIF_COUNTER_CUSTOM', 'OSIF_COUNTER_DUMMY']

4.72 Parameter GptTimeoutDuration

The unit of measurement is given in number of microseconds. This is a timeout value which is used to wait till
- PIT_RTI_LDVAL is synchronized into the RTI clock domain
If the Status is not updated then after this timeout a runtime error will be reported.
This parameter is used for PitRti only

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	800
max	65535
min	1

4.73 Parameter GptReportWakeupSource

Enables/Disables wakeup source reporting.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.74 Reference GptEcucPartitionRef

Maps the GPT driver to zero or multiple ECUC partitions to make the driver API available in the according

partition. Depending on the addressed timer resource the interfaces operate as follows:

In case of partition local timer resources (n:1 mapping) the API operates as an independent instance in the according ECUC partition.

In case of global timer resources (1:m mapping) the API operates on the global timer resource either by protected access to the resource or by implementing an according kernel.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.75 Reference GptKernelEcucPartitionRef

Maps the GPT kernel to zero or one ECUC partitions to assign the driver kernel to a certain core. The ECUC partition referenced is a subset of the ECUC partitions where the GPT driver is mapped to.

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	true
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	true
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.76 Container GptClockReferencePoint

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

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.77 Reference GptClockReference

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

Property	Value
type	ECUC-REFERENCE-DEF
origin	AUTOSAR_ECUC
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Mcu/McuModuleConfiguration/McuClockSetting↩ Config/McuClockReferencePoint

4.78 Container GptPredefTimerConfiguration

Container for configuring the Predefined Timer functionality.

Included subcontainers:

- GptPredefTimer_1us_16Bit
- GptPredefTimer_1us_24Bit
- GptPredefTimer_1us_32Bit
- GptPredefTimer_100us_32Bit

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.79 Container GptPredefTimer_1us_16Bit

This container contains the 1U_16BIT predef timer configuration (parameters) of the GPT Driver

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.80 Parameter GptHwChannel

Vendor specific: Selects the physical GPT Channel.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	FTM_0_PREDEF
literals	['FTM_0_PREDEF', 'FTM_1_PREDEF', 'FTM_2_PREDEF', 'FTM_3_P↩ REDEF']

4.81 Parameter GptFtmChannelClkSrc

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.

FTM_GPT_IP_CLOCK_SOURCE_NONENone use clock for FTM?

FTM_GPT_IP_CLOCK_SOURCE_SYSTEMCLKSystem clock

FTM_GPT_IP_CLOCK_SOURCE_FIXED_FREQUENCYFixed frequency clock

FTM_GPT_IP_CLOCK_SOURCE_EXTERNALCLKExternal clock

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	FTM_GPT_IP_CLOCK_SOURCE_NONE
literals	['FTM_GPT_IP_CLOCK_SOURCE_NONE', 'FTM_GPT_IP_CLOCK_S↩ OURCE_SYSTEMCLK', 'FTM_GPT_IP_CLOCK_SOURCE_FIXED_FR↩ EQUENCY', 'FTM_GPT_IP_CLOCK_SOURCE_EXTERNALCLK']

4.82 Parameter GptChannelPrescaler

Vendor specific: The GPT module specific clock prescaler value.

Note with FTM:

- FTM prescaler should be 1, 2, 4, 8, 16, 32, 64 or 128.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	1
max	256
min	1

4.83 Parameter GptFreezeEnable

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.84 Reference GptChannelClkSrcRef

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

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Gpt/GptDriverConfiguration/GptClockReferencePoint

4.85 Container GptPredefTimer_1us_24Bit

This container contains the 1U_24BIT predef timer configuration (parameters) of the GPT Driver

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.86 Parameter GptHwChannel

Vendor specific: Selects the physical GPT Channel.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NOT_APPLICABLE
literals	['NOT_APPLICABLE']

4.87 Parameter GptChannelPrescaler

Vendor specific: The GPT module specific clock prescaler value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	1
max	256
min	1

4.88 Parameter GptFreezeEnable

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.89 Reference GptChannelClkSrcRef

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

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Gpt/GptDriverConfiguration/GptClockReferencePoint

4.90 Container GptPredefTimer_1us_32Bit

This container contains the predef timer configuration (parameters) of the GPT Driver

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.91 Parameter GptHwChannel

Vendor specific: Selects the physical GPT Channel.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NOT_APPLICABLE
literals	['NOT_APPLICABLE']

4.92 Parameter GptChannelPrescaler

Vendor specific: The GPT module specific clock prescaler value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	1
max	256
min	1

4.93 Parameter GptFreezeEnable

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.94 Reference GptChannelClkSrcRef

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

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Gpt/GptDriverConfiguration/GptClockReferencePoint

4.95 Container GptPredefTimer_100us_32Bit

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

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF

Property	Value
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.96 Parameter GptHwChannel

Vendor specific: Selects the physical GPT Channel.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	NOT_APPLICABLE
literals	['NOT_APPLICABLE']

4.97 Parameter GptChannelPrescaler

Vendor specific: The GPT module specific clock prescaler value.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD

Property	Value
defaultValue	1
max	256
min	1

4.98 Parameter GptFreezeEnable

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

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	true

4.99 Reference GptChannelClkSrcRef

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

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Gpt/GptDriverConfiguration/GptClockReferencePoint

4.100 Container CommonPublishedInformation

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

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.101 Parameter ArReleaseMajorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.102 Parameter ArReleaseMinorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	7
max	7
min	7

4.103 Parameter ArReleaseRevisionVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.104 Parameter ModuleId

Module ID of this module from Module List.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	100
max	100
min	100

4.105 Parameter SwMajorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	2
max	2
min	2

4.106 Parameter SwMinorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.107 Parameter SwPatchVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.108 Parameter VendorApiInfix

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>.

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.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	

4.109 Parameter VendorId

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	43
max	43
min	43

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 FTM IPL

6.1.1 Detailed Description

Data Structures

- struct Ftm_Gpt_Ip_InstanceConfigType
  Structure to configure the FTM instance. More...
- struct Ftm_Gpt_Ip_ChannelConfigType
  Structure to configure the FTM channels. More...
- struct Ftm_Gpt_Ip_ChState
  internal context structure More...
- struct Ftm_Gpt_Ip_InstancePrescalerType
  internal context structure More...

Types Reference

- typedef void(* Ftm_Gpt_Ip_CallbackType) (uint8 callbackParam)
  Callback type for each channel.

Enum Reference

- enum Ftm_Gpt_Ip_CountingMode
  Unit options for counting mode.
- enum Ftm_Gpt_Ip_ClockSource
  Enum containing the FTM module clock sources.
- enum Ftm_Gpt_Ip_ClockModeType
  Prescaler type. Indicates of whether the clock channel mode is "NORMAL" or "ALTERNATE".
- enum Ftm_Gpt_Ip_ChannelModeType
  Channel mode type. Indicates of whether the channel mode is "CONTINUOUS" or "ONE SHOT".

Function Reference

- void Ftm_Gpt_Ip_Init (uint8 instance, const Ftm_Gpt_Ip_InstanceConfigType *configPtr)
  Function Name : Ftm_Gpt_Ip_Init.
- void Ftm_Gpt_Ip_InitChannel (uint8 instance, const Ftm_Gpt_Ip_ChannelConfigType *configPtr)
  Function Name : Ftm_Gpt_Ip_InitChannel.
- void Ftm_Gpt_Ip_Deinit (uint8 instance)
  Function Name : Ftm_Gpt_Ip_Deinit.
- void Ftm_Gpt_Ip_StartCounting (uint8 instance, uint8 channel, uint16 compareValue)
  Function Name : Ftm_Gpt_Ip_StartCounting.
- void Ftm_Gpt_Ip_StartTimer (uint8 instance, uint16 counterValue)
  Function Name : Ftm_Gpt_Ip_StartTimer.
- void Ftm_Gpt_Ip_StopTimer (uint8 instance)
  Function Name : Ftm_Gpt_Ip_StopTimer.
- void Ftm_Gpt_Ip_EnableChannelInterrupt (uint8 instance, uint8 channel)
  Function Name : Ftm_Gpt_Ip_EnableChannelInterrupt.
- void Ftm_Gpt_Ip_DisableChannelInterrupt (uint8 instance, uint8 channel)
  Function Name : Ftm_Gpt_Ip_DisableChannelInterrupt.
- void Ftm_Gpt_Ip_SetHalfCycleReloadPoint (uint8 instance, uint16 reloadPoint)
  Function Name : Ftm_Gpt_Ip_SetHalfCycleReloadPoint.
- void Ftm_Gpt_Ip_ChangeNextTimeoutValue (uint8 instance, uint8 channel, uint16 value)
  The function changes the Ftm compare register value.
- void Ftm_Gpt_Ip_SetClockMode (uint8 instance, Ftm_Gpt_Ip_ClockModeType prescalerMode)
  The function changes the FTM prescaler value.
- void Ftm_Gpt_Ip_StartPredefTimer (uint8 instance, uint8 channel, uint8 uPrescaler, uint8 clocksource, boolean bFreezeEnable)
  The function start the FTM channel.
- void Ftm_Gpt_Ip_StopPredefTimer (uint8 instance, uint8 channel)
  The function stop the FTM channel.

Variables

- FTM_Type *const ftmGptBase [FTM_INSTANCE_COUNT]
  Table of base addresses for FTM instances.

6.1.2 Data Structure Documentation

6.1.2.1 struct Ftm_Gpt_Ip_InstanceConfigType

Structure to configure the FTM instance.

This structure holds the configuration settings for InstanceConfigType

Implements Ftm_Gpt_Ip_InstanceConfigType

Definition at line 166 of file Ftm_Gpt_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	freezeBits	Enable/Disable freeze bits.
Ftm_Gpt_Ip_ClockSource	clocksource	Select FTM clocksource.
uint8	clockAlternatePrescaler	Select AlternatePrescaler.
uint8	clockPrescaler	Select prescalerValue.
Ftm_Gpt_Ip_CountingMode	mode	Select mode.

6.1.2.2 struct Ftm_Gpt_Ip_ChannelConfigType

Structure to configure the FTM channels.

This structure holds the configuration settings for the ChannelConfigType

Implements Ftm_Gpt_Ip_ChannelConfigType

Definition at line 183 of file Ftm_Gpt_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	hwChannel	hwChannel
Ftm_Gpt_Ip_CallbackType	callback	callback
uint8	callbackParam	callbackParam
Ftm_Gpt_Ip_ChannelModeType	channelMode	channelMode

6.1.2.3 struct Ftm_Gpt_Ip_ChState

internal context structure

This structure is used by the IPL driver for internal logic. The content is populated on InitChannel

Definition at line 197 of file Ftm_Gpt_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	chInit	chInit
Ftm_Gpt_Ip_CallbackType	callback	callback
uint8	callbackParam	callbackParam
Ftm_Gpt_Ip_ChannelModeType	channelMode	channelMode

6.1.2.4 struct Ftm_Gpt_Ip_InstancePrescalerType

internal context structure

This structure is used by the IPL driver for internal logic. The content is populated on Init

Definition at line 211 of file Ftm_Gpt_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	clockPrescaler	Clock divide value for the NormalPrescaler.
uint8	clockAlternatePrescaler	Clock divide value for the AlternatePrescaler.

6.1.3 Types Reference

6.1.3.1 Ftm_Gpt_Ip_CallbackType

```
typedef void(* Ftm_Gpt_Ip_CallbackType) (uint8 callbackParam)
```

Callback type for each channel.

Ftm_Gpt_Ip_CallbackType

Definition at line 158 of file Ftm_Gpt_Ip_Types.h.

6.1.4 Enum Reference

6.1.4.1 Ftm_Gpt_Ip_CountingMode

```
enum Ftm_Gpt_Ip_CountingMode
```

Unit options for counting mode.

This is used to choose how the timer is counting

Enumerator

FTM_GPT_IP_MODE_UP_TIMER	Timer with up counter.
FTM_GPT_IP_MODE_UP_DOWN_TIMER	timer with up-down counter

Definition at line 109 of file Ftm_Gpt_Ip_Types.h.

6.1.4.2 Ftm_Gpt_Ip_ClockSource

enum Ftm_Gpt_Ip_ClockSource

Enum containing the FTM module clock sources.

This is used to choose the FTM clock sources.

Enumerator

FTM_GPT_IP_CLOCK_SOURCE_NONE	None use clock for FTM
FTM_GPT_IP_CLOCK_SOURCE_SYSTEMCLK	System clock
FTM_GPT_IP_CLOCK_SOURCE_FIXED_FREQUENCY	Fixed frequency
FTM_GPT_IP_CLOCK_SOURCE_EXTERNALCLK	External clock

Definition at line 120 of file Ftm_Gpt_Ip_Types.h.

6.1.4.3 Ftm_Gpt_Ip_ClockModeType

enum Ftm_Gpt_Ip_ClockModeType

Prescaler type. Indicates of whether the clock channel mode is "NORMAL" or "ALTERNATE".

This enumeration specifies the possible types of prescalers used to configure base-clock timers

Enumerator

FTM_GPT_IP_CLOCKMODE_NORMAL	Selected value is the NORMAL configured prescaler.
FTM_GPT_IP_CLOCKMODE_ALTERNATE	Selected value is the ALTERNATE configured prescaler.

Definition at line 134 of file Ftm_Gpt_Ip_Types.h.

6.1.4.4 Ftm_Gpt_Ip_ChannelModeType

enum Ftm_Gpt_Ip_ChannelModeType

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

ChannelModeType of channel.

Enumerator

FTM_GPT_IP_CH_MODE_CONTINUOUS	channel mode - continuous mode
FTM_GPT_IP_CH_MODE_ONESHOT	channel mode - one-shot mode.

Definition at line 145 of file Ftm_Gpt_Ip_Types.h.

6.1.5 Function Reference

6.1.5.1 Ftm_Gpt_Ip_Init()

```
void Ftm_Gpt_Ip_Init (
            uint8 instance,
            const Ftm_Gpt_Ip_InstanceConfigType * configPtr )
```

Function Name : Ftm_Gpt_Ip_Init.

Initializes the FTM instance. This functions is called for each FTM hardware Instance.

Parameters

in	*instance*	FTM hardware instance number
in	*configPtr*	Pointer to a selected configuration structure

Returns

void

Precondition

The data structure including the configuration set required for initializing the GPT driver

6.1.5.2 Ftm_Gpt_Ip_InitChannel()

```
void Ftm_Gpt_Ip_InitChannel (
            uint8 instance,
            const Ftm_Gpt_Ip_ChannelConfigType * configPtr )
```

Function Name : Ftm_Gpt_Ip_InitChannel.

Initializes the FTM channels. This functions is called for each FTM hardware channel and:

Parameters

in	*instance*	FTM hardware instance number
in	*configPtr*	Pointer to a selected configuration structure

Returns

void

Precondition

The data structure including the configuration set required for initializing the GPT driver

6.1.5.3 Ftm_Gpt_Ip_Deinit()

```
void Ftm_Gpt_Ip_Deinit (
             uint8 instance )
```

Function Name : Ftm_Gpt_Ip_Deinit.

De-Initializes the FTM module. This functions is called for each FTM hardware instance and:

- resets all channels to default
- disables the timer compare interrupts corresponding to Ftm channel
- clears the timer compare interrupt flags corresponding to Ftm channel
- resets the counter register and the counter initial value register.
- resets the channel value register and the modulo register
- disables the freeze mode

Parameters

in	*instance*	FTM hardware instance number

Returns

void

Precondition

The data structure including the configuration set required for initializing the GPT driver.

6.1.5.4 Ftm_Gpt_Ip_StartCounting()

```
void Ftm_Gpt_Ip_StartCounting (
             uint8 instance,
             uint8 channel,
             uint16 compareValue )
```

Function Name : Ftm_Gpt_Ip_StartCounting.

This function is called for starting the Ftm timer channel

Parameters

in	*instance*	FTM hardware instance number
in	*channel*	Ftm channel
in	*compareValue*	Compare value

Returns

void

Precondition

The driver needs to be initialized. This function is called for starting the FTM timer channel.

6.1.5.5 Ftm_Gpt_Ip_StartTimer()

```
void Ftm_Gpt_Ip_StartTimer (
             uint8 instance,
             uint16 counterValue )
```

Function Name : Ftm_Gpt_Ip_StartTimer.

This function is called for setting a new start counter value and enables the FTM counter and

- sets the new counter value
- enables the FTM counter

6.1.5.6 Ftm_Gpt_Ip_StopTimer()

```
void Ftm_Gpt_Ip_StopTimer (
             uint8 instance )
```

Function Name : Ftm_Gpt_Ip_StopTimer.

This function is callded for stopping the Ftm counter.

- disables the FTM counter

Parameters

in	*instance*	FTM hardware instance

Returns

void

Precondition

The driver needs to be initialized. This function is called for stoping the FTM timer channel.

6.1.5.7 Ftm_Gpt_Ip_EnableChannelInterrupt()

```
void Ftm_Gpt_Ip_EnableChannelInterrupt (
            uint8 instance,
            uint8 channel )
```

Function Name : Ftm_Gpt_Ip_EnableChannelInterrupt.

This function allows enabling interrupt generation of timer channel when timeout occurs

Parameters

in	*instance*	FTM hardware instance
in	*channel*	FTM hardware channel

Returns

void

Precondition

The driver needs to be initialized.

6.1.5.8 Ftm_Gpt_Ip_DisableChannelInterrupt()

```
void Ftm_Gpt_Ip_DisableChannelInterrupt (
            uint8 instance,
            uint8 channel )
```

Function Name : Ftm_Gpt_Ip_DisableChannelInterrupt.

This function allows disabling interrupt generation of timer channel when timeout occurs

Parameters

in	*instance*	FTM hardware instance
in	*channel*	FTM hardware channel

Returns

void

Precondition

The driver needs to be initialized.

6.1.5.9 Ftm_Gpt_Ip_SetHalfCycleReloadPoint()

```
void Ftm_Gpt_Ip_SetHalfCycleReloadPoint (
            uint8 instance,
            uint16 reloadPoint )
```

Function Name : Ftm_Gpt_Ip_SetHalfCycleReloadPoint.

Configures the value of the counter with half cycle of reload point.

Parameters

in	*instance*	FTM hardware instance
in	*reloadPoint*	Reload value

Returns

Precondition

The driver needs to be initialized.

6.1.5.10 Ftm_Gpt_Ip_ChangeNextTimeoutValue()

```
void Ftm_Gpt_Ip_ChangeNextTimeoutValue (
            uint8 instance,
```

```
            uint8 channel,
            uint16 value )
```

The function changes the Ftm compare register value.

This function:

- Write next timeout to local variable

Parameters

in	*instance*	FTM hardware instance
in	*channel*	Channel
in	*value*	Channel timeout value

Returns

void

Precondition

The driver needs to be initialized.

6.1.5.11 Ftm_Gpt_Ip_SetClockMode()

```
void Ftm_Gpt_Ip_SetClockMode (
            uint8 instance,
            Ftm_Gpt_Ip_ClockModeType prescalerMode )
```

The function changes the FTM prescaler value.

This function sets the FTM prescaler based on the input mode.

Parameters

in	*instance*	FTM hardware instance
in	*prescalerMode*	FTM_GPT_IP_CLOCKMODE_NORMAL or FTM_GPT_IP_CLOCKMODE_ALTERNATE

Returns

void

Precondition

The driver needs to be initialized.On/Off by the configuration parameter: GPT_DUAL_CLOCK_MODE

6.1.5.12 Ftm_Gpt_Ip_StartPredefTimer()

```
void Ftm_Gpt_Ip_StartPredefTimer (
             uint8 instance,
             uint8 channel,
             uint8 uPrescaler,
             uint8 clocksource,
             boolean bFreezeEnable )
```

The function start the FTM channel.

This function start the FTM channel the input mode.

Parameters

in	*instance*	FTM hardware instance
in	*uPrescaler*	Prescaler
in	*bFreezeEnable*	enables/disables freeze bits

Returns

void

Precondition

The driver needs to be initialized

6.1.5.13 Ftm_Gpt_Ip_StopPredefTimer()

```
void Ftm_Gpt_Ip_StopPredefTimer (
             uint8 instance,
             uint8 channel )
```

The function stop the FTM channel.

This function stop the FTM channel.

Parameters

in	*channel*	FTM hardware channel
in	*instance*	FTM hardware instance

Returns

void

Precondition

The driver needs to be initialized

6.1.6 Variable Documentation

6.1.6.1 ftmGptBase

```
FTM_Type* const ftmGptBase[FTM_INSTANCE_COUNT]  [extern]
```

Table of base addresses for FTM instances.

6.2 Gpt Driver

6.2.1 Detailed Description

Macros

- #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.
- #define GPT_E_UNINIT

  Function called without module initialization.
- #define GPT_E_ALREADY_INITIALIZED

  Initialization called when already initialized.
- #define GPT_E_PARAM_CHANNEL

  Function called for invalid channel.
- #define GPT_E_PARAM_VALUE

  Function called with parameter value out of range.
- #define GPT_E_PARAM_POINTER

  Function called with NULL pointer.
- #define GPT_E_PARAM_PREDEF_TIMER

  Function called with invalid the parameter in function Gpt_GetPredefTimerValue.
- #define GPT_E_BUSY

  Function called when timer channel is still running.
- #define GPT_E_MODE

  Function called with invalid the parameter in function Gpt_GetPredefTimerValue.
- #define GPT_E_TIMEOUT

  Function called when a timeout is occurred.
- #define GPT_E_INIT_FAILED

  Function called with invalid the parameter in function Gpt_Init.
- #define GPT_E_PARAM_CLOCK_MODE

  API Gpt_SetClockMode service called with wrong parameter.
- #define GPT_E_PARAM_MODE

  Function called with invalid mode param.
- #define GPT_E_PARAM_CONFIG

  function called for invalid channel on the current core
- #define GPT_GETVERSIONINFO_ID

  API service ID for Gpt_GetVersionInfo function.
- #define GPT_INIT_ID

  API service ID for Gpt_Init function.
- #define GPT_DEINIT_ID

  API service ID for Gpt_DeInit function.
- #define GPT_TIMEELAPSED_ID

  API service ID for Gpt_GetTimeElapsed function.
- #define GPT_TIMEREMAINING_ID

  API service ID for Gpt_GetTimeRemaining function.
- #define GPT_STARTTIMER_ID

API service ID for Gpt_StartTimer function.

- #define GPT_STOPTIMER_ID

  API service ID for Gpt_StopTimer function.
- #define GPT_ENABLENOTIFICATION_ID

  API service ID for Gpt_EnableNotification function.
- #define GPT_DISABLENOTIFICATION_ID

  API service ID for Gpt_DisableNotification function.
- #define GPT_SETMODE_ID

  API service ID for Gpt_SetMode function.
- #define GPT_DISABLEWAKEUP_ID

  API service ID for Gpt_DisableWakeup function.
- #define GPT_ENABLEWAKEUP_ID

  API service ID for Gpt_EnableWakeup function.
- #define GPT_CHECKWAKEUP_ID

  API service ID for Gpt_CheckWakeup function.
- #define GPT_PROCESSCOMMONINTERRUPT_ID

  API service ID for Gpt_ProcessCommonInterrupt generic ISR handler.
- #define GPT_CHANGE_NEXT_TIMEOUT_VALUE_ID

  API service ID for Gpt_ChangeNextTimeoutValue function.
- #define GPT_SET_CLOCK_MODE_ID

  API service ID for Gpt_SetClockMode function.
- #define GPT_GET_PREDEF_TIMERVALUE_ID

  API service ID for Gpt_GetPredefTimerValue function.
- #define GPT_ENABLE_CHAIN_MODE_ID

  API service ID for Gpt_Channel_EnableChainMode function.
- #define GPT_INSTANCE_ID

  Instance ID of this GPT driver.
- #define GPT_VALIDATE_GLOBAL_CALL

  GPT_VALIDATE_GLOBAL_CALL.
- #define GPT_VALIDATE_CHANNEL_CALL

  GPT_VALIDATE_CHANNEL_CALL.
- #define GPT_VALIDATE_STATE

  GPT_VALIDATE_STATE.
- #define GPT_VALIDATE_PARAM

  GPT_VALIDATE_PARAM.

Enum Reference

- enum Gpt_ModeType

  This enumerated type allows the selection of different power modes.
- enum Gpt_ChannelModeType

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

  Prescaler type. Indicates of whether the clock channel mode is "GPT_NORMAL" or "GPT_ALTERNATE".

Function Reference

- void Gpt_GetVersionInfo (Std_VersionInfoType *VersionInfoPtr)

  This function returns the version information of this module.
- void Gpt_Init (const Gpt_ConfigType *configPtr)

  GPT driver initialization function.
- void Gpt_DeInit (void)

  GPT driver de-initialization function.
- Gpt_ValueType Gpt_GetTimeElapsed (Gpt_ChannelType channel)

  GPT driver function for fetching the elapsed timer value.
- Gpt_ValueType Gpt_GetTimeRemaining (Gpt_ChannelType channel)

  GPT driver function for fetching the remaining timer value.
- void Gpt_StartTimer (Gpt_ChannelType channel, Gpt_ValueType value)

  GPT driver function for starting a timer channel.
- void Gpt_StopTimer (Gpt_ChannelType channel)

  GPT driver function for stopping a timer channel.
- void Gpt_EnableNotification (Gpt_ChannelType channel)

  GPT driver function for enabling the notification for a timer channel.
- void Gpt_DisableNotification (Gpt_ChannelType channel)

  GPT driver function for disabling the notification for a timer channel.
- void Gpt_SetMode (Gpt_ModeType Mode)

  GPT driver function for setting the operation mode.
- void Gpt_DisableWakeup (Gpt_ChannelType channel)

  GPT driver function for disabling the wakeup interrupt invocation for a timer channel.
- void Gpt_EnableWakeup (Gpt_ChannelType channel)

  GPT driver function for enabling the wakeup interrupt invocation for a timer channel.
- void Gpt_CheckWakeup (EcuM_WakeupSourceType wakeupSource)

  GPT driver function for checking if a wakeup capable GPT channel is the source for a wakeup event.
- void Gpt_ChangeNextTimeoutValue (Gpt_ChannelType channel, Gpt_ValueType value)

  The function changes the time out period value of the requested running channel.
- void Gpt_SetClockMode (Gpt_ClockModeType eClkMode)

  This function changes the channel pre scaler.
- void Gpt_Channel_EnableChainMode (Gpt_ChannelType channel)

  The function enables the chain functionality for timer.
- void Gpt_Channel_DisableChainMode (Gpt_ChannelType channel)

  The function disables the chain functionality for timer.
- Std_ReturnType Gpt_GetPredefTimerValue (Gpt_PredefTimerType PredefTimer, uint32 *TimeValuePtr)

  Provides the current value of the given predefined free-running timer.
- void Gpt_ProcessCommonInterrupt (uint8 channel)

  Gpt common handler to implements generic part of the ISR.

6.2.2 Macro Definition Documentation

6.2.2.1 GPT_E_INVALID_CALL

```
#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.

Errors and exceptions that will be detected by the GPT driver.

Gpt_Det_ErrorCodes_define

Definition at line 142 of file Gpt.h.

6.2.2.2 GPT_E_UNINIT

```
#define GPT_E_UNINIT
```

Function called without module initialization.

Errors and exceptions that will be detected by the GPT driver.

Gpt_Det_ErrorCodes_define

Definition at line 153 of file Gpt.h.

6.2.2.3 GPT_E_ALREADY_INITIALIZED

```
#define GPT_E_ALREADY_INITIALIZED
```

Initialization called when already initialized.

Errors and exceptions that will be detected by the GPT driver.

Gpt_Det_ErrorCodes_define

Definition at line 163 of file Gpt.h.

6.2.2.4 GPT_E_PARAM_CHANNEL

```
#define GPT_E_PARAM_CHANNEL
```

Function called for invalid channel.

Errors and exceptions that will be detected by the GPT driver.

Gpt_Det_ErrorCodes_define

Definition at line 172 of file Gpt.h.

6.2.2.5 GPT_E_PARAM_VALUE

```
#define GPT_E_PARAM_VALUE
```

Function called with parameter value out of range.

Errors and exceptions that will be detected by the GPT driver

Gpt_Det_ErrorCodes_define

Definition at line 181 of file Gpt.h.

6.2.2.6 GPT_E_PARAM_POINTER

```
#define GPT_E_PARAM_POINTER
```

Function called with NULL pointer.

Errors and exceptions that will be detected by the GPT driver

Gpt_Det_ErrorCodes_define

Definition at line 192 of file Gpt.h.

6.2.2.7 GPT_E_PARAM_PREDEF_TIMER

```
#define GPT_E_PARAM_PREDEF_TIMER
```

Function called with invalid the parameter in function Gpt_GetPredefTimerValue.

Errors and exceptions that will be detected by the GPT driver

Gpt_Det_ErrorCodes_define

Definition at line 202 of file Gpt.h.

6.2.2.8 GPT_E_BUSY

```
#define GPT_E_BUSY
```

Function called when timer channel is still running.

Errors and exceptions that will be detected by the GPT driver.

Gpt_Det_ErrorCodes_define

Definition at line 211 of file Gpt.h.

6.2.2.9 GPT_E_MODE

```
#define GPT_E_MODE
```

Function called with invalid the parameter in function Gpt_GetPredefTimerValue.

Errors and exceptions that will be detected by the GPT driver

Gpt_Det_ErrorCodes_define

Definition at line 219 of file Gpt.h.

6.2.2.10 GPT_E_TIMEOUT

```
#define GPT_E_TIMEOUT
```

Function called when a timeout is occurred.

Errors and exceptions that will be detected by the GPT driver.

Gpt_Det_ErrorCodes_define

Definition at line 228 of file Gpt.h.

6.2.2.11 GPT_E_INIT_FAILED

```
#define GPT_E_INIT_FAILED
```

Function called with invalid the parameter in function Gpt_Init.

Errors and exceptions that will be detected by the GPT driver

Gpt_Det_ErrorCodes_define

Definition at line 236 of file Gpt.h.

6.2.2.12 GPT_E_PARAM_CLOCK_MODE

```
#define GPT_E_PARAM_CLOCK_MODE
```

API Gpt_SetClockMode service called with wrong parameter.

Parameters used when raising an error/exception

Definition at line 245 of file Gpt.h.

6.2.2.13 GPT_E_PARAM_MODE

```
#define GPT_E_PARAM_MODE
```

Function called with invalid mode param.

Errors and exceptions that will be detected by the GPT driver

Gpt_Det_ErrorCodes_define

Definition at line 255 of file Gpt.h.

6.2.2.14 GPT_E_PARAM_CONFIG

```
#define GPT_E_PARAM_CONFIG
```

function called for invalid channel on the current core

Errors and exceptions that will be detected by the GPT driver

Gpt_Det_ErrorCodes_define

Definition at line 264 of file Gpt.h.

6.2.2.15 GPT_GETVERSIONINFO_ID

```
#define GPT_GETVERSIONINFO_ID
```

API service ID for Gpt_GetVersionInfo function.

API SERVICE IDs

Parameters used when raising an error/exception

Definition at line 273 of file Gpt.h.

6.2.2.16 GPT_INIT_ID

```
#define GPT_INIT_ID
```

API service ID for Gpt_Init function.

Parameters used when raising an error/exception

Definition at line 280 of file Gpt.h.

6.2.2.17 GPT_DEINIT_ID

```
#define GPT_DEINIT_ID
```

API service ID for Gpt_DeInit function.

Parameters used when raising an error/exception

Definition at line 288 of file Gpt.h.

6.2.2.18 GPT_TIMEELAPSED_ID

```
#define GPT_TIMEELAPSED_ID
```

API service ID for Gpt_GetTimeElapsed function.

Parameters used when raising an error/exception

Definition at line 296 of file Gpt.h.

6.2.2.19 GPT_TIMEREMAINING_ID

```
#define GPT_TIMEREMAINING_ID
```

API service ID for Gpt_GetTimeRemaining function.

Parameters used when raising an error/exception

Definition at line 304 of file Gpt.h.

6.2.2.20 GPT_STARTTIMER_ID

```
#define GPT_STARTTIMER_ID
```

API service ID for Gpt_StartTimer function.

Parameters used when raising an error/exception

Definition at line 311 of file Gpt.h.

6.2.2.21 GPT_STOPTIMER_ID

```
#define GPT_STOPTIMER_ID
```

API service ID for Gpt_StopTimer function.

Parameters used when raising an error/exception

Definition at line 318 of file Gpt.h.

6.2.2.22 GPT_ENABLENOTIFICATION_ID

```
#define GPT_ENABLENOTIFICATION_ID
```

API service ID for Gpt_EnableNotification function.

Parameters used when raising an error/exception

Definition at line 326 of file Gpt.h.

6.2.2.23 GPT_DISABLENOTIFICATION_ID

```
#define GPT_DISABLENOTIFICATION_ID
```

API service ID for Gpt_DisableNotification function.

Parameters used when raising an error/exception

Definition at line 332 of file Gpt.h.

6.2.2.24 GPT_SETMODE_ID

```
#define GPT_SETMODE_ID
```

API service ID for Gpt_SetMode function.

Parameters used when raising an error/exception

Definition at line 341 of file Gpt.h.

6.2.2.25 GPT_DISABLEWAKEUP_ID

```
#define GPT_DISABLEWAKEUP_ID
```

API service ID for Gpt_DisableWakeup function.

Parameters used when raising an error/exception

Definition at line 351 of file Gpt.h.

6.2.2.26 GPT_ENABLEWAKEUP_ID

```
#define GPT_ENABLEWAKEUP_ID
```

API service ID for Gpt_EnableWakeup function.

Parameters used when raising an error/exception

Definition at line 357 of file Gpt.h.

6.2.2.27 GPT_CHECKWAKEUP_ID

```
#define GPT_CHECKWAKEUP_ID
```

API service ID for Gpt_CheckWakeup function.

Parameters used when raising an error/exception

Definition at line 365 of file Gpt.h.

6.2.2.28 GPT_PROCESSCOMMONINTERRUPT_ID

```
#define GPT_PROCESSCOMMONINTERRUPT_ID
```

API service ID for Gpt_ProcessCommonInterrupt generic ISR handler.

Parameters used when raising an error/exception

Definition at line 372 of file Gpt.h.

6.2.2.29 GPT_CHANGE_NEXT_TIMEOUT_VALUE_ID

```
#define GPT_CHANGE_NEXT_TIMEOUT_VALUE_ID
```

API service ID for Gpt_ChangeNextTimeoutValue function.

Parameters used when raising an error/exception

Definition at line 379 of file Gpt.h.

6.2.2.30 GPT_SET_CLOCK_MODE_ID

```
#define GPT_SET_CLOCK_MODE_ID
```

API service ID for Gpt_SetClockMode function.

Parameters used when raising an error/exception

Definition at line 387 of file Gpt.h.

6.2.2.31 GPT_GET_PREDEF_TIMERVALUE_ID

```
#define GPT_GET_PREDEF_TIMERVALUE_ID
```

API service ID for Gpt_GetPredefTimerValue function.

Parameters used when raising an error/exception

Definition at line 395 of file Gpt.h.

6.2.2.32 GPT_ENABLE_CHAIN_MODE_ID

```
#define GPT_ENABLE_CHAIN_MODE_ID
```

API service ID for Gpt_Channel_EnableChainMode function.

Parameters used when raising an error/exception

Definition at line 403 of file Gpt.h.

6.2.2.33 GPT_INSTANCE_ID

```
#define GPT_INSTANCE_ID
```

Instance ID of this GPT driver.

Definition at line 411 of file Gpt.h.

6.2.2.34 GPT_VALIDATE_GLOBAL_CALL

```
#define GPT_VALIDATE_GLOBAL_CALL
```

GPT_VALIDATE_GLOBAL_CALL.

Validates the global call uses all the channels - Gpt_Init, Gpt_DeInit, Gpt_SetMode.

Definition at line 72 of file Gpt_EnvCfg.h.

6.2.2.35 GPT_VALIDATE_CHANNEL_CALL

```
#define GPT_VALIDATE_CHANNEL_CALL
```

GPT_VALIDATE_CHANNEL_CALL.

Validates the call for a specific channel.

Definition at line 78 of file Gpt_EnvCfg.h.

6.2.2.36 GPT_VALIDATE_STATE

```
#define GPT_VALIDATE_STATE
```

GPT_VALIDATE_STATE.

Validates the channel status.

Definition at line 84 of file Gpt_EnvCfg.h.

6.2.2.37 GPT_VALIDATE_PARAM

```
#define GPT_VALIDATE_PARAM
```

GPT_VALIDATE_PARAM.

Validates the time value parameter.

Definition at line 90 of file Gpt_EnvCfg.h.

6.2.3 Enum Reference

6.2.3.1 Gpt_ModeType

```
enum Gpt_ModeType
```

This enumerated type allows the selection of different power modes.

Modes of the GPT driver.

Enumerator

GPT_MODE_NORMAL	GPT Normal operation mode of the GPT.
GPT_MODE_SLEEP	GPT Sleep mode.

Definition at line 422 of file Gpt.h.

6.2.3.2 Gpt_ChannelModeType

`enum Gpt_ChannelModeType`

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

ChannelModeType of channel.

Enumerator

GPT_CH_MODE_CONTINUOUS	GPT channel mode - continuous mode.
GPT_CH_MODE_ONESHOT	GPT channel mode - one-shot mode.

Definition at line 432 of file Gpt.h.

6.2.3.3 Gpt_ClockModeType

`enum Gpt_ClockModeType`

Prescaler type. Indicates of whether the clock channel mode is "GPT_NORMAL" or "GPT_ALTERNATE".

This enumeration specifies the possible types of prescalers used to configure base-clock timers

Enumerator

GPT_CLOCKMODE_NORMAL	Selected value is the NORMAL configured prescaler.
GPT_CLOCKMODE_ALTERNATE	Selected value is the ALTERNATE configured prescaler.

Definition at line 443 of file Gpt.h.

6.2.4 Function Reference

6.2.4.1 Gpt_GetVersionInfo()

```
void Gpt_GetVersionInfo (
            Std_VersionInfoType * VersionInfoPtr )
```

This function returns the version information of this module.

This service returns the version information of this module. The version information includes:

- Module Id
- Vendor Id
- Vendor specific version numbers If source code for caller and callee of this function is available this function should be realized as a macro. The macro should be defined in the modules header file.

Parameters

out	*versioninfo*	- pointer to location to store version info

Returns

void

Precondition

Gpt_Init must be called before.

6.2.4.2 Gpt_Init()

```
void Gpt_Init (
            const Gpt_ConfigType * configPtr )
```

GPT driver initialization function.

This service is a non reentrant function used 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

Parameters

in	*configPtr*	Pointer to a selected configuration structure

Returns

void

Precondition

The data structure including the configuration set required for initializing the GPT driver..

6.2.4.3 Gpt_DeInit()

```
void Gpt_DeInit (
            void  )
```

GPT driver de-initialization function.

Service for de initializing 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 de initialization functionality without any action.

Returns

void

Precondition

The driver needs to be initialized.

6.2.4.4 Gpt_GetTimeElapsed()

```
Gpt_ValueType Gpt_GetTimeElapsed (
            Gpt_ChannelType channel )
```

GPT driver function for fetching the elapsed timer value.

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.

Parameters

in	*channel*	- channel id

Returns

Gpt_ValueType - Elapsed Time in number of ticks

Precondition

The driver needs to be initialized.

6.2.4.5 Gpt_GetTimeRemaining()

```
Gpt_ValueType Gpt_GetTimeRemaining (
            Gpt_ChannelType channel )
```

GPT driver function for fetching the remaining timer value.

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, pre-scalers 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.

Parameters

in	*channel*	- channel id

Returns

Gpt_ValueType - Returns the time remaining until the target time is reached in number of ticks.

Precondition

The driver needs to be initialized.

6.2.4.6 Gpt_StartTimer()

```
void Gpt_StartTimer (
            Gpt_ChannelType channel,
            Gpt_ValueType value )
```

GPT driver function for starting a timer channel.

The function Gpt_StartTimer shall start the selected timer channel with a defined time-out period. The function Gpt_StartTimer shall invoke the configured notification for that channel (see also GPT292) after the time-out 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 time-out 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, pre-scalers 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.

Parameters

in	*channel*	channel id
in	*value*	time-out period (in number of ticks) after a notification or a wakeup event shall occur.

Returns

void

Precondition

The driver needs to be initialized.

6.2.4.7 Gpt_StopTimer()

```
void Gpt_StopTimer (
            Gpt_ChannelType channel )
```

GPT driver function for stopping a timer channel.

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 time-out 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.

Parameters

in	*channel*	channel id

Returns

void

Precondition

The driver needs to be initialized. Gpt_StartTimer must be called before.

6.2.4.8 Gpt_EnableNotification()

```
void Gpt_EnableNotification (
            Gpt_ChannelType channel )
```

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

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.

Parameters

in	*channel*	channel id

Returns

void

Precondition

The driver needs to be initialized.

6.2.4.9 Gpt_DisableNotification()

```
void Gpt_DisableNotification (
            Gpt_ChannelType channel )
```

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

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.

Parameters

in	*channel*	channel id

Returns

void

Precondition

The driver needs to be initialized.

6.2.4.10 Gpt_SetMode()

```
void Gpt_SetMode (
            Gpt_ModeType Mode )
```

GPT driver function for setting the operation mode.

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.

Parameters

in	*eMode*	operation mode : • GPT_MODE_NORMAL: Normal operation mode of the GPT driver. • GPT_MODE_SLEEP: Sleep mode of the GPT driver (wakeup capable)

Returns

void

Precondition

The driver needs to be initialized.

6.2.4.11 Gpt_DisableWakeup()

```
void Gpt_DisableWakeup (
            Gpt_ChannelType channel )
```

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

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.

Parameters

in	*channel*	channel id

Returns

void

Precondition

The driver needs to be initialized.

6.2.4.12 Gpt_EnableWakeup()

```
void Gpt_EnableWakeup (
            Gpt_ChannelType channel )
```

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

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.

Parameters

in	*channel*	channel id

Returns

void

Precondition

The driver needs to be initialized. The channel must be configured as wakeup capable.

6.2.4.13 Gpt_CheckWakeup()

```
void Gpt_CheckWakeup (
            EcuM_WakeupSourceType wakeupSource )
```

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

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.

Parameters

in	*wakeupSource*	wakeup source

Returns

void

Precondition

The driver needs to be initialized. The channel must be configured as wakeup capable.

6.2.4.14 Gpt_ChangeNextTimeoutValue()

```
void Gpt_ChangeNextTimeoutValue (
            Gpt_ChannelType channel,
            Gpt_ValueType value )
```

The function changes the time out period value of the requested running channel.

The function changes the time out period (in number of ticks) of the channel is running which will be used after the first compare matching. This is a non-autosar function.

Parameters

in	*channel*	channel id
in	*value*	time out period (in number of ticks) after a notification shall occur

Returns

void

Precondition

Gpt_Init and Gpt_StartTimer must be called before.

6.2.4.15 Gpt_SetClockMode()

```
void Gpt_SetClockMode (
            Gpt_ClockModeType eClkMode )
```

This function changes the channel pre scaler.

This function sets all channels pre scalers based on the input mode.

Parameters

in	*eClkMode*	pre scaler setting (NORMAL or ALTERNATE)

Returns

void

Precondition

Gpt_Init must be called before.

6.2.4.16 Gpt_Channel_EnableChainMode()

```
void Gpt_Channel_EnableChainMode (
            Gpt_ChannelType channel )
```

The function enables the chain functionality for timer.

The function enables the chain functionality for timer. Timer will be chained with timer n-1. Channel 0 cannot be chained. This is a non-autosar function.

Parameters

in	*channel*	channel id

Returns

void

Precondition

Gpt_Init must be called before.

6.2.4.17 Gpt_Channel_DisableChainMode()

```
void Gpt_Channel_DisableChainMode (
            Gpt_ChannelType channel )
```

The function disables the chain functionality for timer.

The function disables the chain functionality for timer. Timer will not be chained with timer n-1. Channel 0 cannot be chained or unchained. This is a non-autosar function.

Parameters

in	*channel*	channel id

Returns

void

Precondition

Gpt_Init must be called before.

6.2.4.18 Gpt_GetPredefTimerValue()

```
Std_ReturnType Gpt_GetPredefTimerValue (
            Gpt_PredefTimerType PredefTimer,
            uint32 * TimeValuePtr )
```

Provides the current value of the given predefined free-running timer.

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

Parameters

in	*PredefTimer*	Gpt_PredefTimerType (GPT_PREDEF_TIMER_1US_16BIT, GPT_PREDEF_TIMER_1US_24BIT, GPT_PREDEF_TIMER_1US_32BIT GPT_PREDEF_TIMER_100US_32BIT)
out	*TimeValuePtr*	Pointer to time value destination data in RAM

Precondition

Gpt_Init must be called before.

Returns

returnValue - E_OK: no error has been detected.

- E_NOT_OK: aborted due to errors.

6.2.4.19 Gpt_ProcessCommonInterrupt()

```
void Gpt_ProcessCommonInterrupt (
             uint8 channel )
```

Gpt common handler to implements generic part of the ISR.

Generic function used by all interrupt service routines to call notification functions and wakeup the EcuM

Parameters

in	*channel*	logic channel number

Returns

void

Precondition

The driver needs to be initialized.

6.3 LPit IPL

6.3.1 Detailed Description

Data Structures

- struct Lpit_Gpt_Ip_InstanceConfigType
  Structure to configure the LPIT. More...
- struct Lpit_Gpt_Ip_ChannelConfigType
  Structure to configure the LPIT timer channel. More...
- struct Lpit_Gpt_Ip_State
  internal context structure More...

Types Reference

- typedef void(* Lpit_Gpt_Ip_CallbackType) (uint8 callbackParam)
  Callback type for each channel.

Enum Reference

- enum Lpit_Gpt_Ip_StatusType
  LPit Status error.
- enum Lpit_Gpt_Ip_ChannelModeType
  Channel mode type. Indicates of whether the channel mode is "CONTINUOUS" or "ONE SHOT".

Function Reference

- void Lpit_Gpt_Ip_Init (uint8 instance, const Lpit_Gpt_Ip_InstanceConfigType *config)
  Function Name : Lpit_Gpt_Ip_Init.
- void Lpit_Gpt_Ip_InitChannel (uint8 instance, const Lpit_Gpt_Ip_ChannelConfigType *configChannel)
  Function Name : Lpit_Gpt_Ip_InitChannel.
- void Lpit_Gpt_Ip_Deinit (uint8 instance)
  Function Name: Lpit_Gpt_Ip_Deinit.
- void Lpit_Gpt_Ip_StartTimer (uint8 instance, uint8 channel, uint32 countValue)
  Function Name : Lpit_Gpt_Ip_StartTimer.
- void Lpit_Gpt_Ip_StopTimer (uint8 instance, uint8 channel)
  Function Name : Lpit_Gpt_Ip_StopTimer.
- void Lpit_Gpt_Ip_EnableChInterrupt (uint8 instance, uint8 channel)
  Function Name : Lpit_Gpt_Ip_EnableChInterrupt.
- void Lpit_Gpt_Ip_DisableChInterrupt (uint8 instance, uint8 channel)
  Function Name : Lpit_Gpt_Ip_DisableChInterrupt.
- Lpit_Gpt_Ip_StatusType Lpit_Gpt_Ip_ChainMode (uint8 instance, uint8 channel, boolean enable)
  Function Name : Lpit_Gpt_Ip_ChainMode.
- void Lpit_Gpt_Ip_ChangeNextTimeoutValue (uint8 instance, uint8 channel, uint32 value)
  The function changes Lpit Timer Load Value Register value.

Variables

- LPIT_Type *const LPitGptBase [LPIT_INSTANCE_COUNT]
  Table of base addresses for PIT instances.

6.3.2 Data Structure Documentation

6.3.2.1 struct Lpit_Gpt_Ip_InstanceConfigType

Structure to configure the LPIT.

This structure holds the configuration settings for the LPIT Implements :

Definition at line 126 of file LPit_Gpt_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	runInDozeMode	DOZE Mode Enable Bit
boolean	stopRunInDebug	Stop timer running in debug mode.

6.3.2.2 struct Lpit_Gpt_Ip_ChannelConfigType

Structure to configure the LPIT timer channel.

This structure holds the configuration settings for the LPIT timer channel Implements :

Definition at line 140 of file LPit_Gpt_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	hwChannel	Timer channel number
Lpit_Gpt_Ip_CallbackType	callback	callback
uint8	callbackParam	callbackParam
uint32	triggerConfig	Trigger source configure for LPIT Timer.
Lpit_Gpt_Ip_ChannelModeType	channelMode	channelMode

6.3.2.3 struct Lpit_Gpt_Ip_State

internal context structure

This structure is used by the IPL driver for internal logic. The content is populated on Init

Definition at line 157 of file LPit_Gpt_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	chInit	chInit
Lpit_Gpt_Ip_CallbackType	callback	callback
uint8	callbackParam	callbackParam
Lpit_Gpt_Ip_ChannelModeType	channelMode	channelMode

6.3.3 Types Reference

6.3.3.1 Lpit_Gpt_Ip_CallbackType

```
typedef void(* Lpit_Gpt_Ip_CallbackType) (uint8 callbackParam)
```

Callback type for each channel.

Pit_Ip_CallbackType

Definition at line 118 of file LPit_Gpt_Ip_Types.h.

6.3.4 Enum Reference

6.3.4.1 Lpit_Gpt_Ip_StatusType

```
enum Lpit_Gpt_Ip_StatusType
```

LPit Status error.

Status error

Enumerator

LPIT_GPT_IP_SUCCESS	Status value is SUCCESS.
LPIT_GPT_IP_ERROR	Status value is ERROR.

Definition at line 95 of file LPit_Gpt_Ip_Types.h.

6.3.4.2 Lpit_Gpt_Ip_ChannelModeType

```
enum Lpit_Gpt_Ip_ChannelModeType
```

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

ChannelModeType of channel.

Enumerator

LPIT_GPT_IP_CH_MODE_CONTINUOUS	channel mode - continuous mode
LPIT_GPT_IP_CH_MODE_ONESHOT	channel mode - one-shot mode.

Definition at line 105 of file LPit_Gpt_Ip_Types.h.

6.3.5 Function Reference

6.3.5.1 Lpit_Gpt_Ip_Init()

```
void Lpit_Gpt_Ip_Init (
            uint8 instance,
            const Lpit_Gpt_Ip_InstanceConfigType * config )
```

Function Name : Lpit_Gpt_Ip_Init.

Driver initialization function for LPit instance.

Parameters

in	*instance*	LPIT hw instance number
in	*config*	Pointer to a selected configuration structure

Returns

void

Precondition

The data structure including the configuration set required for initializing the driver

6.3.5.2 Lpit_Gpt_Ip_InitChannel()

```
void Lpit_Gpt_Ip_InitChannel (
            uint8 instance,
            const Lpit_Gpt_Ip_ChannelConfigType * configChannel )
```

Function Name : Lpit_Gpt_Ip_InitChannel.

Initializes the LPIT channels. This functions is called for each PIT hw channel and:

Parameters

in	*instance*	PIT hw instance number
in	*configChannel*	Pointer to a selected configuration structure.

Returns

void

Precondition

The data structure including the configuration set required for initializing the driver.

6.3.5.3 Lpit_Gpt_Ip_Deinit()

```
void Lpit_Gpt_Ip_Deinit (
            uint8 instance )
```

Function Name: Lpit_Gpt_Ip_Deinit.

De-Initializes the LPIT instances. This functions is called and

Parameters

in	*instance*	LPit hw instance

Returns

void

Precondition

The data structure including the configuration set required for initializing the GPT driver.

6.3.5.4 Lpit_Gpt_Ip_StartTimer()

```
void Lpit_Gpt_Ip_StartTimer (
             uint8 instance,
             uint8 channel,
             uint32 countValue )
```

Function Name : Lpit_Gpt_Ip_StartTimer.

This function is called for starting the LPit timer channel.

- sets the timeout value into the LPIT timer channel register
- enables the LPIT channel

Parameters

in	*instance*	LPit hw instance
in	*channel*	LPit hw channel
in	*countValue*	channel timeout value

Returns

void

Precondition

The driver needs to be initialized. This function is called for starting the Pit timer channel.

6.3.5.5 Lpit_Gpt_Ip_StopTimer()

```
void Lpit_Gpt_Ip_StopTimer (
             uint8 instance,
             uint8 channel )
```

Function Name : Lpit_Gpt_Ip_StopTimer.

This function is called for stopping the Pit counter. This function disables the LPIT channel and:

Parameters

in	*instance*	LPit hw instance
in	*channel*	LPit hw channel

Returns

void

Precondition

The driver needs to be initialized. LPit_Gpt_Ip_StartTimer must be call before.

6.3.5.6 Lpit_Gpt_Ip_EnableChInterrupt()

```
void Lpit_Gpt_Ip_EnableChInterrupt (
            uint8 instance,
            uint8 channel )
```

Function Name : Lpit_Gpt_Ip_EnableChInterrupt.

This function allows enabling interrupt generation of timer channel when timeout occurs

Parameters

in	*instance*	LPit hw instance
in	*channel*	LPit hw channel

Returns

void

Precondition

The driver needs to be initialized.

6.3.5.7 Lpit_Gpt_Ip_DisableChInterrupt()

```
void Lpit_Gpt_Ip_DisableChInterrupt (
            uint8 instance,
            uint8 channel )
```

Function Name : Lpit_Gpt_Ip_DisableChInterrupt.

This function allows disabling interrupt of a timer channel

Parameters

in	*instance*	LPit hw instance
in	*channel*	LPit hw channel

Returns

void

Precondition

The driver needs to be initialized.

6.3.5.8 Lpit_Gpt_Ip_ChainMode()

```
Lpit_Gpt_Ip_StatusType Lpit_Gpt_Ip_ChainMode (
            uint8 instance,
            uint8 channel,
            boolean enable )
```

Function Name : Lpit_Gpt_Ip_ChainMode.

This function:

- Chain/Unchain LPit channels.

Parameters

in	*instance*	LPit hw channel ID
in	*channel*	channel timeout value
in	*enable*	enable/disable chain mode

Returns

returnValue

Precondition

The driver needs to be initialized. LPIT_GPT_IP_CHAIN_MODE == STD_ON

6.3.5.9 Lpit_Gpt_Ip_ChangeNextTimeoutValue()

```
void Lpit_Gpt_Ip_ChangeNextTimeoutValue (
             uint8 instance,
             uint8 channel,
             uint32 value )
```

The function changes Lpit Timer Load Value Register value.

This function:

- sets the next timeout value.

Parameters

in	*instance*	Lpit hw instance ID
in	*channel*	Lpit hw channel ID
in	*value*	channel timeout value

Returns

void

Precondition

The driver needs to be initialized. LPIT_GPT_IP_CHANGE_NEXT_TIMEOUT_VALUE == STD_ON

6.3.6 Variable Documentation

6.3.6.1 LPitGptBase

```
LPIT_Type* const LPitGptBase[LPIT_INSTANCE_COUNT]  [extern]
```

Table of base addresses for PIT instances.

6.4 Lptmr IPL

6.4.1 Detailed Description

Data Structures

- struct Lptmr_Gpt_Ip_State

  Internal context structure Lptmr_Gpt_Ip_State. More...
- struct Lptmr_Gpt_Ip_InstanceState

  internal context structure More...

Macros

- #define LPTMR_GPT_IP_TM_MODE

  Define Mode.
- #define E_TIMEOUT

  LPTMR E_TIMEOUT.

Types Reference

- typedef void(* Lptmr_Gpt_Ip_CallbackType) (uint8 callbackParam)

  Callback type for each channel.

Enum Reference

- enum Lptmr_Gpt_Ip_ClockSelectType

  Enum containing the LPTMR module clock sources.
- enum Lptmr_Gpt_Ip_StatusType

  LPTMR Status error.
- enum Lptmr_Gpt_Ip_ChannelModeType

  Channel mode type. Indicates of whether the channel mode is "CONTINUOUS" or "ONE SHOT".
- enum Lptmr_Gpt_Ip_ClockModeType

  Prescaler type. Indicates of whether the clock channel mode is "NORMAL" or "ALTERNATE".

Function Reference

- void Lptmr_Gpt_Ip_Init (uint8 instance, const Lptmr_Gpt_Ip_ConfigType *config)
  LPTMR Driver initialization function.
- void Lptmr_Gpt_Ip_DeInit (uint8 instance)
  de-initialization function for Lptmr module.
- void Lptmr_Gpt_Ip_StartTimer (uint8 instance, uint16 value)
  Function for starting the Lptmr timer channel.
- void Lptmr_Gpt_Ip_StopTimer (uint8 instance)
  Function for stopping the Lptmr timer channel.
- void Lptmr_Gpt_Ip_EnableInterrupt (uint8 instance)
  Driver function for Enable Interrupt for LPTMR channel.
- Lptmr_Gpt_Ip_StatusType Lptmr_Gpt_Ip_SetCompareValue (uint8 instance, uint16 compareValue)
  Lptmr_Gpt_Ip_SetCompareValue.
- void Lptmr_Gpt_Ip_DisableInterrupt (uint8 instance)
  Gpt driver function for Disable Interrupt for LPTMR channel.
- void Lptmr_Gpt_Ip_SetClockMode (uint8 instance, Lptmr_Gpt_Ip_ClockModeType clockMode)
  The function changes the LPtimer prescaler value.

6.4.2 Data Structure Documentation

6.4.2.1 struct Lptmr_Gpt_Ip_State

Internal context structure Lptmr_Gpt_Ip_State.

This structure is used by the IPL driver for internal logic. The content is populated on Init.

Definition at line 160 of file Lptmr_Gpt_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	chInit	chInit
Lptmr_Gpt_Ip_CallbackType	callback	callback
uint8	callbackParam	callbackParam
Lptmr_Gpt_Ip_ChannelModeType	channelMode	channelMode

6.4.2.2 struct Lptmr_Gpt_Ip_InstanceState

internal context structure

This structure is used by the IPL driver for internal logic. The content is populated on Init

Definition at line 187 of file Lptmr_Gpt_Ip_Types.h.

Data Fields

Type	Name	Description
uint8	clockPrescaler	Clock divide value for the NormalPrescaler.
uint8	clockAlternatePrescaler	Clock divide value for the AlternatePrescaler.

6.4.3 Macro Definition Documentation

6.4.3.1 LPTMR_GPT_IP_TM_MODE

```
#define LPTMR_GPT_IP_TM_MODE
```

Define Mode.

Mode type (TMS)

Time Counter mode

Definition at line 165 of file Lptmr_Gpt_Ip.h.

6.4.3.2 E_TIMEOUT

```
#define E_TIMEOUT
```

LPTMR E_TIMEOUT.

Definition at line 88 of file Lptmr_Gpt_Ip_Types.h.

6.4.4 Types Reference

6.4.4.1 Lptmr_Gpt_Ip_CallbackType

```
typedef void(* Lptmr_Gpt_Ip_CallbackType) (uint8 callbackParam)
```

Callback type for each channel.

Lptmr_Gpt_Ip_CallbackType

Definition at line 135 of file Lptmr_Gpt_Ip_Types.h.

6.4.5 Enum Reference

6.4.5.1 Lptmr_Gpt_Ip_ClockSelectType

```
enum Lptmr_Gpt_Ip_ClockSelectType
```

Enum containing the LPTMR module clock sources.

Lptmr_Gpt_Ip_ClockSelectType

Enumerator

LPTMR_GPT_IP_CLOCK_SIRCDIV2	LPTMR clock source SIRCDIV2_CLK.
LPTMR_GPT_IP_CLOCK_LPO1K	LPTMR clock source LPO1K.
LPTMR_GPT_IP_CLOCK_RTC_CLK	LPTMR clock source RTC_CLK.
LPTMR_GPT_IP_CLOCK_PCC_LPTMR0	LPTMR clock source PCC_LPTMR0.

Definition at line 99 of file Lptmr_Gpt_Ip_Types.h.

6.4.5.2 Lptmr_Gpt_Ip_StatusType

enum Lptmr_Gpt_Ip_StatusType

LPTMR Status error.

Status error

Enumerator

LPTMR_GPT_IP_SUCCESS	Status value is SUCCESS.
LPTMR_GPT_IP_ERROR	Status value is ERROR.
LPTMR_GPT_IP_TIMEOUT	Status value is TIMEOUT.

Definition at line 111 of file Lptmr_Gpt_Ip_Types.h.

6.4.5.3 Lptmr_Gpt_Ip_ChannelModeType

enum Lptmr_Gpt_Ip_ChannelModeType

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

ChannelModeType of channel.

Enumerator

LPTMR_GPT_IP_CH_MODE_CONTINUOUS	channel mode - continuous mode
LPTMR_GPT_IP_CH_MODE_ONESHOT	channel mode - one-shot mode.

Definition at line 122 of file Lptmr_Gpt_Ip_Types.h.

6.4.5.4 Lptmr_Gpt_Ip_ClockModeType

```
enum Lptmr_Gpt_Ip_ClockModeType
```

Prescaler type. Indicates of whether the clock channel mode is "NORMAL" or "ALTERNATE".

This enumeration specifies the possible types of prescalers used to configure base-clock timers

Enumerator

LPTMR_GPT_IP_CLOCKMODE_NORMAL	Selected value is the NORMAL configured prescaler
LPTMR_GPT_IP_CLOCKMODE_ALTERNATE	Selected value is the ALTERNATE configured prescaler.

Definition at line 173 of file Lptmr_Gpt_Ip_Types.h.

6.4.6 Function Reference

6.4.6.1 Lptmr_Gpt_Ip_Init()

```
void Lptmr_Gpt_Ip_Init (
            uint8 instance,
            const Lptmr_Gpt_Ip_ConfigType * config )
```

LPTMR Driver initialization function.

This function is called separately for each LPTMR hw channel corresponding to the configured

```
- enables the LPTMR module
- configures the freeze mode (enabled or disabled)
- disables the IRQ corresponding to the LPTMR channel
- clears the (pending) interrupt flag corresponding to Lptmr channel
- disables the LPTMR timer channel
- clears the Load Value register corresponding to the Lptmr channel.
```

Parameters

in	*instance*	hw instance
in	*∗config*	pointer to configuration

Returns

void

Precondition

The data structure including the configuration set required for initializing the GPT driver

6.4.6.2 Lptmr_Gpt_Ip_DeInit()

```
void Lptmr_Gpt_Ip_DeInit (
             uint8 instance )
```

de-initialization function for Lptmr module.

This function is called separately for each LPTMR hw channel corresponding to the configured timer channels, and:

- disables the LPTMR channel
- disables the freeze mode
- disables IRQ corresponding to Lptmr channel
- clears the (pending) interrupt flag corresponding to Lptmr channel

Parameters

in	*instance*	LPTMR hw instance

Returns

void

Precondition

The data structure including the configuration set required for initializing the GPT driver

6.4.6.3 Lptmr_Gpt_Ip_StartTimer()

```
void Lptmr_Gpt_Ip_StartTimer (
             uint8 instance,
             uint16 value )
```

Function for starting the Lptmr timer channel.

This function:

- clears the (pending) interrupt flag corresponding to Lptmr channel
- disables the LPTMR timer channel
- sets the timeout value into the LPTMR timer channel register
- enables the LPTMR timer channel
- enables the IRQ corresponding to the LPTMR timer channel,if channel configured in One Shot mode.

Parameters

in	*value*	channel timeout value

Returns

void

Precondition

The data structure including the configuration set required for initializing the GPT driver

6.4.6.4 Lptmr_Gpt_Ip_StopTimer()

```
void Lptmr_Gpt_Ip_StopTimer (
             uint8 instance )
```

Function for stopping the Lptmr timer channel.

This function disables the LPTMR channel

Parameters

in	*instance*	LPTMR hw instance

Returns

void

Precondition

The driver needs to be initialized.

6.4.6.5 Lptmr_Gpt_Ip_EnableInterrupt()

```
void Lptmr_Gpt_Ip_EnableInterrupt (
             uint8 instance )
```

Driver function for Enable Interrupt for LPTMR channel.

This function:

- Enable Interrupt for LPTMR channel

Parameters

in	*instance*	LPTMR hw instance

Returns

void

Precondition

The driver needs to be initialized.

6.4.6.6 Lptmr_Gpt_Ip_SetCompareValue()

```
Lptmr_Gpt_Ip_StatusType Lptmr_Gpt_Ip_SetCompareValue (
            uint8 instance,
            uint16 compareValue )
```

Lptmr_Gpt_Ip_SetCompareValue.

This function:

- Set the compare value in counter tick units, for a LPTMR instance. Possible return values:

STATUS_SUCCESS: completed successfully

- STATUS_ERROR: cannot reconfigure compare value (TCF not set)
- STATUS_TIMEOUT: compare value is smaller than current counter value

Parameters

in	*instance*	LPTMR hw instance
in	*compareValue*	Compare value in counter tick units

Returns

void

Precondition

The driver needs to be initialized.

6.4.6.7 Lptmr_Gpt_Ip_DisableInterrupt()

```
void Lptmr_Gpt_Ip_DisableInterrupt (
            uint8 instance )
```

Gpt driver function for Disable Interrupt for LPTMR channel.

This function:

- Disable Interrupt for LPTMR channel

Parameters

in	*instance*	LPTMR hw instance

Returns

void

Precondition

The driver needs to be initialized.

6.4.6.8 Lptmr_Gpt_Ip_SetClockMode()

```
void Lptmr_Gpt_Ip_SetClockMode (
            uint8 instance,
            Lptmr_Gpt_Ip_ClockModeType clockMode )
```

The function changes the LPtimer prescaler value.

This function sets the LPtimer prescaler based on the input mode.

Parameters

in	*instance*	LPtimer hardware instance
in	*clockMode*	LPTMR_GPT_IP_CLOCKMODE_NORMAL or LPTMR_GPT_IP_CLOCKMODE_ALTERNATE

Returns

void

Precondition

The driver needs to be initialized.On/Off by the configuration parameter: GPT_DUAL_CLOCK_MODE

6.5 Rtc IPL

6.5.1 Detailed Description

Data Structures

- struct Srtc_Ip_ConfigType
  Structure to configure the SRTC. More...
- struct Srtc_Ip_TimedateType
  SRTC Time Date structure. More...
- struct Srtc_Ip_AlarmConfigType
  SRTC alarm configuration. More...
- struct Srtc_Ip_State
  Internal context structure Srtc_Ip_State. More...

Macros

- #define SECONDS_IN_A_DAY
  SRTC Channels defines.

Types Reference

- typedef void(* Srtc_Ip_CallbackType) (uint8 callbackParam)
  Callback type for each channel.

Enum Reference

- enum Srtc_Ip_ClockSelectType
  Enum containing the SRTC module clock sources.
- enum Srtc_Ip_ClockOutType
  SRTC CLKOUT pin configuration.
- enum Srtc_Ip_InterruptType
  Enum containing SRTC interrupt flags.
- enum Srtc_Ip_StatusType
  SRtc Status error.
- enum Srtc_Ip_SecIntFreqType
  SRTC Seconds interrupt configuration.
- enum Srtc_Ip_ChannelModeType
  Channel mode type. Indicates of whether the channel mode is "CONTINUOUS" or "ONE SHOT".

Function Reference

- boolean Srtc_Ip_GetInterruptStatusFlag (uint8 instance, Srtc_Ip_InterruptType interruptMode)

  Function Name : Srtc_Ip_GetInterruptStatusFlag Description : Get the status of Time Invalid Flag (TIF), Time Alarm Flag (TAF) and Time Overflow Flag (TOF): RTC Status Register (SR)
- void Srtc_Ip_ConvertSecondsToTimeDate (const uint32 *const seconds, Srtc_Ip_TimedateType *const timeDate)

  SRtc Driver function.
- void Srtc_Ip_ConvertTimeDateToSeconds (const Srtc_Ip_TimedateType *const timeDate, uint32 *const seconds)

  SRtc Driver function.
- void Srtc_Ip_Init (uint8 instance, const Srtc_Ip_ConfigType *config)

  SRtc driver initialization function for SRtc module.
- void Srtc_Ip_DeInit (uint8 instance)

  SRtc driver de-initialization function.
- Srtc_Ip_StatusType Srtc_Ip_StartCounter (uint8 instance)

  SRtc Driver function for starting the Rtc counter.
- Srtc_Ip_StatusType Srtc_Ip_StopCounter (uint8 instance)

  SRtc Driver function for stopping the Rtc counter.
- void Srtc_Ip_StartTimer (uint8 instance, uint32 value)

  SRtc Driver function for starting the Rtc timer channel with a timeout value.
- void Srtc_Ip_EnableInterrupt (uint8 instance, Srtc_Ip_InterruptType interruptMode)

  SRtc Driver function for enabling interrupt for RTC channel.
- void Srtc_Ip_DisableInterrupt (uint8 instance, Srtc_Ip_InterruptType interruptMode)

  SRtc Driver function for Disable Interrupt for RTC channel.
- Srtc_Ip_StatusType Srtc_Ip_SetTimeDate (uint8 instance, const Srtc_Ip_TimedateType *timeDate)

  SRtc Driver function.
- Srtc_Ip_StatusType Srtc_Ip_GetTimeDate (uint8 instance, Srtc_Ip_TimedateType *const timeDate)

  SRtc Driver function.
- Srtc_Ip_StatusType Srtc_Ip_ConfigureAlarm (uint8 instance, const Srtc_Ip_AlarmConfigType *alarm↩ Config)

  SRtc Driver function.
- Srtc_Ip_StatusType Srtc_Ip_ConfigureSecondsInterrupt (uint8 instance, Srtc_Ip_SecIntFreqType occurs↩ Frequency)

  SRtc Driver function.

Variables

- uint32 Srtc_Ip_u32TargetValue

  array variable used to store the runtime target time value.

6.5.2 Data Structure Documentation

6.5.2.1 struct Srtc_Ip_ConfigType

Structure to configure the SRTC.

This structure holds the configuration settings for the SRTC

Definition at line 196 of file SRtc_Ip_Types.h.

Data Fields

Type	Name	Description
Srtc_Ip_ClockSelectType	clockSelect	SRTC Clock Select
Srtc_Ip_ClockOutType	clockOutSelect	SRTC Clock Pin select - RTC_CLKOUT
Srtc_Ip_CallbackType	callback	Periodic interrupt callback
uint8	callbackParam	Pointer to callback parameters
Srtc_Ip_ChannelModeType	channelMode	channelMode

6.5.2.2 struct Srtc_Ip_TimedateType

SRTC Time Date structure.

Srtc_Ip_TimedateType

Definition at line 213 of file SRtc_Ip_Types.h.

Data Fields

Type	Name	Description
uint16	year	Year
uint16	month	Month
uint16	day	Day
uint16	hour	Hour
uint16	minutes	Minutes
uint8	seconds	Seconds

6.5.2.3 struct Srtc_Ip_AlarmConfigType

SRTC alarm configuration.

Srtc_Ip_AlarmConfigType

Definition at line 227 of file SRtc_Ip_Types.h.

Data Fields

Type	Name	Description
Srtc_Ip_TimedateType	alarmTime	Alarm time
uint32	repetitionInterval	Interval of repetition in seconds
uint32	numberOfRepeats	Number of alarm repeats
boolean	repeatForever	Repeat forever if set, discard number of repeats.
boolean	alarmIntEnable	Enable alarm interrupt
Srtc_Ip_CallbackType	alarmCallback	Pointer to the user callback method.
uint8	callbackParams	Pointer to the callback parameters.

6.5.2.4 struct Srtc_Ip_State

Internal context structure Srtc_Ip_State.

This structure is used by the IPL driver for internal logic. The content is populated on Init.

Definition at line 255 of file SRtc_Ip_Types.h.

Data Fields

Type	Name	Description
boolean	alarmInit	alarmInit
uint32	repetitionInterval	Interval of repetition in seconds
volatile uint32	numberOfRepeats	Number of alarm repeats
boolean	repeatForever	Repeat forever if set, discard number of repeats.
boolean	alarmIntEnable	Enable alarm interrupt
Srtc_Ip_CallbackType	alarmCallback	Pointer to the user callback method.
uint8	callbackParams	Pointer to the callback parameters.
volatile boolean	isAlarmTimeNew	Check if there is a new alarm
boolean	chInit	chInit
Srtc_Ip_CallbackType	callback	callback
uint8	callbackParam	callbackParam
Srtc_Ip_ChannelModeType	channelMode	channelMode

6.5.3 Macro Definition Documentation

6.5.3.1 SECONDS_IN_A_DAY

```
#define SECONDS_IN_A_DAY
```

SRTC Channels defines.

Definition at line 88 of file SRtc_Ip_Types.h.

6.5.4 Types Reference

6.5.4.1 Srtc_Ip_CallbackType

```
typedef void(* Srtc_Ip_CallbackType) (uint8 callbackParam)
```

Callback type for each channel.

Srtc_Ip_CallbackType

Definition at line 190 of file SRtc_Ip_Types.h.

6.5.5 Enum Reference

6.5.5.1 Srtc_Ip_ClockSelectType

```
enum Srtc_Ip_ClockSelectType
```

Enum containing the SRTC module clock sources.

Srtc_Ip_ClockSelectType

Enumerator

SRTC_IP_CLK_SRC_OSC_32KHZ	SRTC Prescaler increments using 32 KHz crystal
SRTC_IP_CLK_SRC_LPO_1KHZ	SRTC Prescaler increments using 1KHz LPO

Definition at line 107 of file SRtc_Ip_Types.h.

6.5.5.2 Srtc_Ip_ClockOutType

enum Srtc_Ip_ClockOutType

SRTC CLKOUT pin configuration.

Enumerator

SRTC_IP_CLKOUT_DISABLED	Clock out pin is disabled
SRTC_IP_CLKOUT_SRC_TSIC	Output on RTC_CLKOUT as configured on Time seconds interrupt
SRTC_IP_CLKOUT_SRC_32KHZ	Output on RTC_CLKOUT of the 32KHz clock

Definition at line 116 of file SRtc_Ip_Types.h.

6.5.5.3 Srtc_Ip_InterruptType

enum Srtc_Ip_InterruptType

Enum containing SRTC interrupt flags.

SRtc_Ip_InterruptType

Enumerator

SRTC_IP_INVALID_INTERRUPT	RTC_TIME_INVALID_INTERRUPT
SRTC_IP_OVERFLOW_INTERRUPT	RTC_TIME_OVERFLOW_INTERRUPT.
SRTC_IP_ALARM_INTERRUPT	RTC_TIME_ALARM_INTERRUPT
SRTC_IP_SECONDS_INTERRUPT	RTC_TIME_SECONDS_INTERRUPT

Definition at line 127 of file SRtc_Ip_Types.h.

6.5.5.4 Srtc_Ip_StatusType

enum Srtc_Ip_StatusType

SRtc Status error.

Status error

Enumerator

SRTC_IP_SUCCESS	Status value is SUCCESS.
SRTC_IP_ERROR	Status value is ERROR

Definition at line 139 of file SRtc_Ip_Types.h.

6.5.5.5 Srtc_Ip_SecIntFreqType

enum Srtc_Ip_SecIntFreqType

SRTC Seconds interrupt configuration.

Enumerator

SRTC_IP_INT_1HZ	SRTC seconds interrupt occurs at 1 Hz
SRTC_IP_INT_2HZ	SRTC seconds interrupt occurs at 2 Hz
SRTC_IP_INT_4HZ	SRTC seconds interrupt occurs at 4 Hz
SRTC_IP_INT_8HZ	SRTC seconds interrupt occurs at 8 Hz
SRTC_IP_INT_16HZ	SRTC seconds interrupt occurs at 16 Hz
SRTC_IP_INT_32HZ	SRTC seconds interrupt occurs at 32 Hz
SRTC_IP_INT_64HZ	SRTC seconds interrupt occurs at 64 Hz
SRTC_IP_INT_128HZ	SRTC seconds interrupt occurs at 128 Hz

Definition at line 148 of file SRtc_Ip_Types.h.

6.5.5.6 Srtc_Ip_ChannelModeType

enum Srtc_Ip_ChannelModeType

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

ChannelModeType of channel.

Enumerator

SRTC_IP_CH_MODE_CONTINUOUS	channel mode - continuous mode
SRTC_IP_CH_MODE_ONESHOT	channel mode - one-shot mode.

Definition at line 177 of file SRtc_Ip_Types.h.

6.5.6 Function Reference

6.5.6.1 Srtc_Ip_GetInterruptStatusFlag()

```
boolean Srtc_Ip_GetInterruptStatusFlag (
            uint8 instance,
            Srtc_Ip_InterruptType interruptMode )
```

Function Name : Srtc_Ip_GetInterruptStatusFlag Description : Get the status of Time Invalid Flag (TIF), Time Alarm Flag (TAF) and Time Overflow Flag (TOF): RTC Status Register (SR)

Parameters

in	*instance*	RTC hw instance number
in	*interruptMode*	Enum containing RTC interrupt type

Returns

TRUE if an interrupt has occurred, FALSE otherwise

Precondition

The driver needs to be initialized.

6.5.6.2 Srtc_Ip_ConvertSecondsToTimeDate()

```
void Srtc_Ip_ConvertSecondsToTimeDate (
            const uint32 *const seconds,
            Srtc_Ip_TimedateType *const timeDate )
```

SRtc Driver function.

This function:

- will convert seconds into time-date format.

Parameters

in	*seconds*	number of seconds
in	*∗timeDate*	pointer to configuration

Returns

void

Precondition

The driver needs to be initialized.

6.5.6.3 Srtc_Ip_ConvertTimeDateToSeconds()

```
void Srtc_Ip_ConvertTimeDateToSeconds (
             const Srtc_Ip_TimedateType *const timeDate,
             uint32 *const seconds )
```

SRtc Driver function.

This function:

- will convert time-date into seconds.

Parameters

in	*seconds*	number of seconds
in	*∗timeDate*	pointer to configuration

Returns

void

Precondition

The driver needs to be initialized.

6.5.6.4 Srtc_Ip_Init()

```
void Srtc_Ip_Init (
            uint8 instance,
            const Srtc_Ip_ConfigType * config )
```

SRtc driver initialization function for SRtc module.

This function:

- Disables the time counter
- Disables all interupt modes
- Clears Time Invalid Flag
- Selects source Clock
- If compensation support is enabled sets the configured values
- Sets channel state variables

Parameters

in	*instance*	hw instance
in	*∗config*	pointer to configuration

Returns

void

Precondition

The data structure including the configuration set required for initializing the driver

6.5.6.5 Srtc_Ip_DeInit()

```
void Srtc_Ip_DeInit (
            uint8 instance )
```

SRtc driver de-initialization function.

This function:

- Performs a software reset
- Clears global variables

Parameters

in	*instance*	Rtc hw instance

Returns

void

Precondition

The data structure including the configuration set required for initializing the GPT driver

6.5.6.6 Srtc_Ip_StartCounter()

```
Srtc_Ip_StatusType Srtc_Ip_StartCounter (
            uint8 instance )
```

SRtc Driver function for starting the Rtc counter.

This function:

- Enables the counter

Parameters

in	*instance*	Rtc hw instance

Returns

Srtc_Ip_StatusType status error

Precondition

The driver needs to be initialized.

6.5.6.7 Srtc_Ip_StopCounter()

```
Srtc_Ip_StatusType Srtc_Ip_StopCounter (
            uint8 instance )
```

SRtc Driver function for stopping the Rtc counter.

This function:

- Disables Time Alaram Interrupt
- Clears interrupt flag
- Disables the counter

Parameters

in	*instance*	Rtc hw instance

Returns

Srtc_Ip_StatusType status error

Precondition

The driver needs to be initialized.

6.5.6.8 Srtc_Ip_StartTimer()

```
void Srtc_Ip_StartTimer (
            uint8 instance,
            uint32 value )
```

SRtc Driver function for starting the Rtc timer channel with a timeout value.

This function:

- Disables counter
- Disables Time Alaram Interrupt
- Gets Time Seconds Register value
- Calcutates new compare value
- Sets Time Alarm Register
- Enables Time Alaram Interrupt
- Enables counter

Parameters

in	*instance*	Rtc hw instance value channel timeout value

Returns

void

Precondition

The data structure including the configuration set required for initializing the GPT driver

6.5.6.9 Srtc_Ip_EnableInterrupt()

```
void Srtc_Ip_EnableInterrupt (
            uint8 instance,
            Srtc_Ip_InterruptType interruptMode )
```

SRtc Driver function for enabling interrupt for RTC channel.

This function:

- Enables Interrupt for RTC channel based on selected mode

Parameters

in	*instance*	Rtc hw instance interruptMode Rtc hw interrupt mode

Returns

void

Precondition

The driver needs to be initialized.

6.5.6.10 Srtc_Ip_DisableInterrupt()

```
void Srtc_Ip_DisableInterrupt (
            uint8 instance,
            Srtc_Ip_InterruptType interruptMode )
```

SRtc Driver function for Disable Interrupt for RTC channel.

This function:

- Disable Interrupt for RTC channel based on selected mode

Parameters

in	*instance*	Rtc hw instance

Returns

void

Precondition

The driver needs to be initialized.

6.5.6.11 Srtc_Ip_SetTimeDate()

```
Srtc_Ip_StatusType Srtc_Ip_SetTimeDate (
            uint8 instance,
            const Srtc_Ip_TimedateType * timeDate )
```

SRtc Driver function.

This function:

- Sets the date passed by the user.

Parameters

in	*instance*	SRtc hw instance
in	**timeDate*	pointer to configuration

Returns

Srtc_Ip_StatusType status error

Precondition

The driver needs to be initialized.

6.5.6.12 Srtc_Ip_GetTimeDate()

```
Srtc_Ip_StatusType Srtc_Ip_GetTimeDate (
            uint8 instance,
            Srtc_Ip_TimedateType *const timeDate )
```

SRtc Driver function.

This function:

- Gets the current time and date and it will store in the state structure.

Parameters

in	*instance*	sRtc hw instance
in	*∗timeDate*	pointer to configuration

Returns

Srtc_Ip_StatusType status error

Precondition

The driver needs to be initialized.

6.5.6.13 Srtc_Ip_ConfigureAlarm()

```
Srtc_Ip_StatusType Srtc_Ip_ConfigureAlarm (
            uint8 instance,
            const Srtc_Ip_AlarmConfigType * alarmConfig )
```

SRtc Driver function.

This function:

- Configures the alarm based on the configuration structure passed by the user. When using alarm that are configured to be repetitive, enable the interrupt. Otherwise the repeat function will not work.

Parameters

in	*instance*	Rtc hw instance
in	*∗alarmConfig*	pointer to configuration

Returns

Srtc_Ip_StatusType status error

Precondition

The driver needs to be initialized.

6.5.6.14 Srtc_Ip_ConfigureSecondsInterrupt()

```
Srtc_Ip_StatusType Srtc_Ip_ConfigureSecondsInterrupt (
            uint8 instance,
            Srtc_Ip_SecIntFreqType occursFrequency )
```

SRtc Driver function.

This function:

- Configures the Time Seconds Interrupt with the Seconds interrupt Configuration Type.

Parameters

in	*instance*	Rtc hw instance
in	*occursFrequency*	how often seconds interrupt occurs

Returns

Srtc_Ip_StatusType status error

Precondition

The driver needs to be initialized.

6.5.7 Variable Documentation

6.5.7.1 Srtc_Ip_u32TargetValue

```
uint32 Srtc_Ip_u32TargetValue  [extern]
```

array variable used to store the runtime target time value.

How to Reach Us:

Home Page:
nxp.com

Web Support:
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.