

<b>Document Title</b>	Specification of GPT Driver
<b>Document Owner</b>	AUTOSAR GbR
<b>Document Responsibility</b>	AUTOSAR GbR
<b>Document Identification No</b>	030
<b>Document Classification</b>	Standard

<b>Document Version</b>	2.2.0
<b>Document Status</b>	Final
<b>Part of Release</b>	3.0
<b>Revision</b>	0001

<b>Document Change History</b>			
<b>Date</b>	<b>Version</b>	<b>Changed by</b>	<b>Change Description</b>
11.12.2007	2.2.0	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Introduction of consistent description of wakeup concept (as evaluated in Startup/Wakeup Taskforce). This includes modifications and extensions of textual descriptions as well as the modification of sequence charts related to wakeup.</li> <li>• SWS Improvement: improvement of wording, alignment of API description</li> <li>• Introduction of additional development error in case of already initialized module</li> <li>• Document meta information extended</li> <li>• Small layout adaptations made</li> </ul>
31.01.2007	2.1.0	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Header file structure changed significantly</li> <li>• Return values and development errors for Gpt_GetTimeRemaining() and</li> <li>• Gpt_GetTimeElapsed() changed</li> <li>• Development error checking of ConfigPtr in Gpt_Init() changed</li> <li>• Configuration container structure and configuration parameters</li> <li>• changed</li> <li>• Interface Dem_ReportErrorEvent() removed</li> <li>• Legal disclaimer revised</li> <li>• Release Notes added</li> <li>• "Advice for users" revised</li> <li>• "Revision Information" added</li> </ul>
28.04.2006	2.0.0	AUTOSAR Administration	<p>Document structure adapted to common Release 2.0 SWS Template.</p> <ul style="list-style-type: none"> <li>• Added wake-up functionality</li> </ul> <p>For more details see chapter 11</p>

**Document Change History**

<b>Date</b>	<b>Version</b>	<b>Changed by</b>	<b>Change Description</b>
03.06.2005	1.0.0	AUTOSAR Administration	1.0.0 Initial release

Page left intentionally blank

## Disclaimer

**Any use** of these specifications requires membership within the AUTOSAR Development Partnership or an agreement with the AUTOSAR Development Partnership. The AUTOSAR Development Partnership will not be liable for any use of these specifications.

Following the completion of the development of the AUTOSAR specifications commercial exploitation licenses will be made available to end users by way of written License Agreement only.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Copyright © 2004-2007 AUTOSAR Development Partnership. All rights reserved.

## Advice to users of AUTOSAR Specification Documents:

AUTOSAR Specification Documents may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

## Table of Contents

1	Introduction and functional overview .....	8
2	Acronyms and abbreviations .....	9
3	Related documentation.....	10
3.1	Input documents.....	10
3.2	Related standards and norms .....	11
4	Constraints and assumptions .....	12
4.1	Limitations .....	12
4.2	Applicability to car domains .....	12
5	Dependencies to other modules.....	13
5.1	File structure.....	13
5.1.1	Code file structure.....	13
5.1.2	Header file structure .....	14
6	Requirements traceability .....	16
7	Functional specification .....	23
7.1	General behavior.....	23
7.1.1	Functional overview.....	23
7.1.2	State transitions .....	25
7.1.3	Version checking .....	26
7.2	Error classification .....	26
7.3	Error detection.....	27
7.4	Error notification .....	28
8	API specification.....	29
8.1	Imported types.....	29
8.2	Type Definitions.....	29
8.2.1	Gpt_ConfigType.....	29
8.2.2	Gpt_ChannelType.....	29
8.2.3	Gpt_ValueType.....	29
8.2.4	Gpt_ModeType.....	30
8.3	Function definitions.....	30
8.3.1	Gpt_GetVersionInfo .....	30
8.3.2	Gpt_Init .....	31
8.3.3	Gpt_DeInit .....	32
8.3.4	Gpt_GetTimeElapsed .....	33
8.3.5	Gpt_GetTimeRemaining .....	35
8.3.6	Gpt_StartTimer .....	37
8.3.7	Gpt_StopTimer .....	38
8.3.8	Gpt_EnableNotification .....	39
8.3.9	Gpt_DisableNotification .....	40
8.3.10	Gpt_SetMode.....	41
8.3.11	Gpt_DisableWakeup.....	42
8.3.12	Gpt_EnableWakeup.....	43
8.3.13	Gpt_Cbk_CheckWakeup .....	44

8.4	Call-back Notifications .....	45
8.5	Scheduled functions .....	45
8.6	Expected Interfaces .....	45
8.6.1	Mandatory Interfaces .....	45
8.6.2	Optional Interfaces .....	45
8.6.3	Configurable Interfaces .....	45
8.6.3.1	GptNotification .....	46
9	Sequence diagrams .....	47
9.1	Gpt_Init .....	47
9.2	GPT continuous mode .....	48
9.3	GPT one shot mode .....	49
9.4	Disable/Re-enable Notifications .....	50
9.5	Wakeup .....	51
10	Configuration specification .....	52
10.1	How to read this chapter .....	52
10.1.1	Configuration and configuration parameters .....	52
10.1.2	Variants .....	52
10.1.3	Containers .....	52
10.1.4	Specification template for configuration parameters .....	53
10.2	Containers and configuration parameters .....	54
10.2.1	Variants .....	54
10.2.2	Gpt .....	55
10.2.3	GptDriverConfiguration .....	55
10.2.4	GptChannelConfiguration .....	56
10.2.5	GptChannelConfigSet .....	58
10.2.6	GptWakeupConfiguration .....	59
10.2.7	GptConfigurationOfOptApiServices .....	59
10.3	Published Information .....	60
11	Changes to Release 1 .....	62
11.1	Deleted SWS Items .....	62
11.2	Replaced SWS Items .....	62
11.3	Changed SWS Items .....	62
11.4	Added SWS Items .....	63
12	Changes to Release 2.0 .....	64
12.1	Deleted SWS Items .....	64
12.2	Replaced SWS Items .....	64
12.3	Changed SWS Items .....	64
12.4	Added SWS Items .....	65
13	Changes to Release 2.1 .....	66
13.1	Deleted SWS Items .....	66
13.2	Replaced SWS Items .....	66
13.3	Changed SWS Items .....	66
13.4	Added SWS Items .....	67
14	Changes during SWS Improvements by Technical Office .....	68
14.1	Deleted SWS Items .....	68

14.2	Replaced SWS Items .....	68
14.3	Changed SWS Items .....	68
14.4	Added SWS Items .....	68

# 1 Introduction and functional overview

This specification specifies the functionality, API and the configuration of the AUTOSAR Basic Software module GPT Driver.

There are 2 categories of timers within AUTOSAR:

1. Operating system timers (AUTOSAR OS Alarms)
2. Hardware timers,

but only hardware timers are within the scope of this document.

<i>Type of Timer</i>	<i>Specification/ Implementation within Autosar</i>	<i>Usage within Autosar</i>	<i>Use case</i>
Operating system timers	Operating system	<i>BSW: directly Application: via RTE</i>	Activation of periodic tasks. Interior light time-out after door close
<b>Hardware timers</b>	<b>GPT Driver</b>	<b>BSW only</b>	<b>Valve timing Stepper motor control</b>

**Table 1: Scope of the GPT Driver specification**

The module only uses the hardware timer channels of the general-purpose timer unit and thus provides exact and short-term timings for use in the Operating system or within other basic software modules where an OS Alarm service has too much overhead.

An example of a typical time period range is 50µs ... 5ms.



## 2 Acronyms and abbreviations

<b>Acronym:</b>	<b>Description:</b>
Timer channel	Each channel represents one instance of GPT hardware. It identifies a resource that provides a timing value and/or a notification.
Timeout period	Number of ticks, after the timer will expire.
One shot mode	Timer channel stops after reaching its timeout period value
Continuous mode	Timer channel is restarted automatically after reaching timeout period value
Timer tick	Defines the timer resolution

**Table 2: Acronyms**

<b>Abbreviation:</b>	<b>Description:</b>
DEM	Diagnostic Event Manager
DET	Development Error Tracer
GPT	General Purpose Timer
ICU	Input Capture Unit
ECU	Electronic Control Unit
MCU	Micro Controller Unit
BSW	Basic SoftWare
OS	Operating System

**Table 3: Abbreviations**

### 3 Related documentation

#### 3.1 Input documents

- [1] List of Basic Software Modules,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_BasicSoftwareModules.pdf
- [2] Layered Software Architecture,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_LayeredSoftwareArchitecture.pdf
- [3] General Requirements on Basic Software Modules,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_SRS\_General.pdf
- [4] Specification of Development Error Tracer,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_SWS\_DevelopmentErrorTracer.pdf
- [5] Specification of ECU Configuration,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_ECU\_Configuration.pdf
- [6] Specification of Diagnostic Event Manager,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_SWS\_DEM.pdf
- [7] Specification of ECU State Manager,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_SWS\_ECU\_StateManager.pdf
- [8] General Requirements on SPAL,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_SRS\_SPAL\_General.pdf
- [9] Requirements on GPT Driver,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_SRS\_GPT\_Driver.pdf
- [10] Specification of ICU Driver,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_SWS\_ICU\_Driver.pdf
- [11] Specification of MCU Driver,  
[https://svn2.autosar.org/repos2/22\\_Releases](https://svn2.autosar.org/repos2/22_Releases)  
AUTOSAR\_SWS\_MCU\_Driver.doc

- [12] Specification of I/O Hardware Abstraction,  
[https://svn2.autosar.org/repos2/22\\_Releases/AUTOSAR\\_SWS\\_IOHW\\_Abstraction.pdf](https://svn2.autosar.org/repos2/22_Releases/AUTOSAR_SWS_IOHW_Abstraction.pdf)
- [13] Glossary,  
[https://svn2.autosar.org/repos2/22\\_Releases/AUTOSAR\\_Glossary.pdf](https://svn2.autosar.org/repos2/22_Releases/AUTOSAR_Glossary.pdf)
- [14] AUTOSAR Basic Software Module Description Template,  
[https://svn2.autosar.org/repos2/22\\_Releases/AUTOSAR\\_BSW\\_Module\\_Description.pdf](https://svn2.autosar.org/repos2/22_Releases/AUTOSAR_BSW_Module_Description.pdf)

### **3.2 Related standards and norms**

- [14] IEC 7498-1 The Basic Model, IEC Norm, 1994

## **4 Constraints and assumptions**

### **4.1 Limitations**

No limitations.

### **4.2 Applicability to car domains**

No restrictions.

## 5 Dependencies to other modules

### Module DET [4]

In development mode the Error hook-function of module DET [4] will be called.

### Module DEM [6]

Production errors will be reported to the Diagnostic Event Manager

### Module MCU [11]

The GPT depends on the system clock, prescaler(s) and PLL. Thus, changes of the system clock (e.g. PLL on → PLL off) also affect the clock settings of the GPT hardware. Module GPT will not take care of setting the registers, which configure the clock, prescaler(s) and PLL in its init function. This has to be done by the MCU module [11].

### Module EcuM [7]

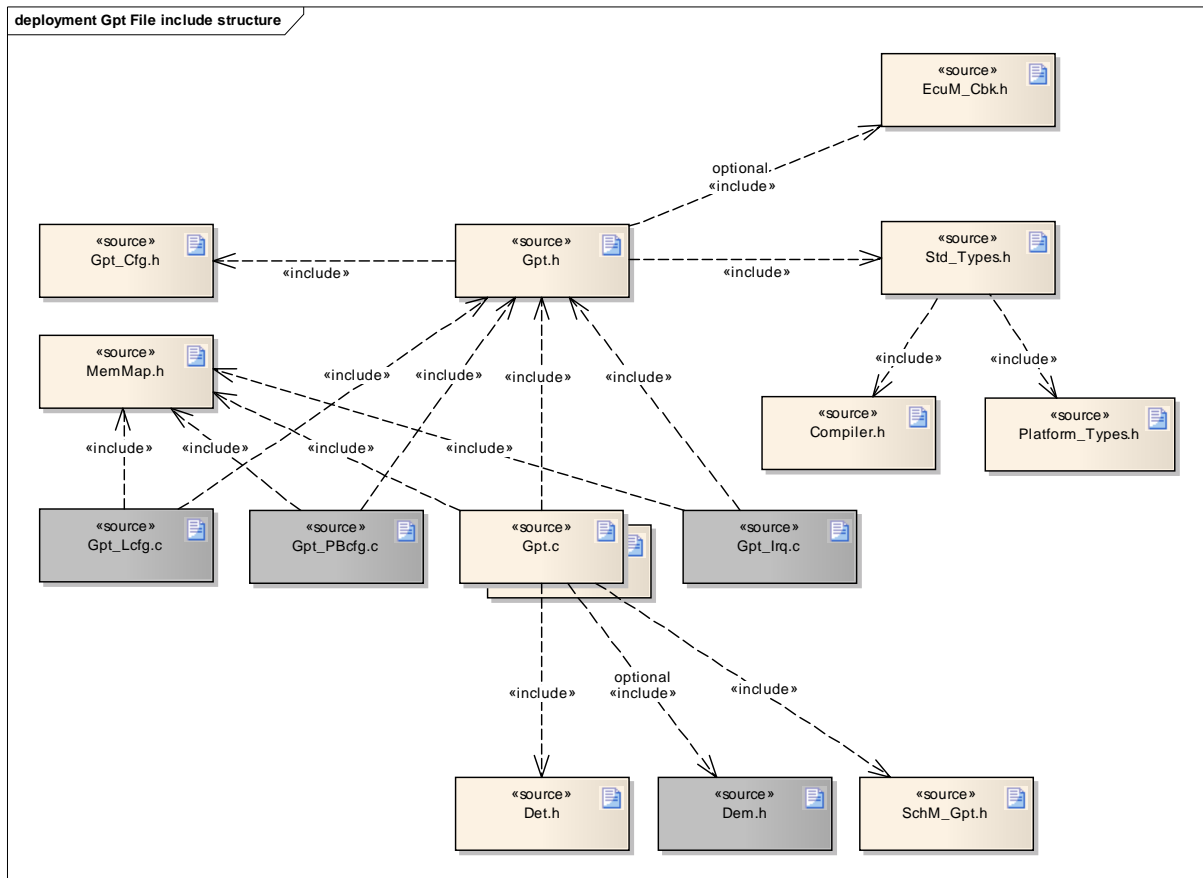
This module processes the wakeup notifications of the GPT.

## 5.1 File structure

### 5.1.1 Code file structure

**GPT171:** The code file structure shall not be defined within this specification completely. At this point it shall be pointed out that the code-file structure shall include the file named "Gpt\_PBcfg.c" for post build time configurable parameters. This file shall contain all post-build time configurable parameters.

**5.1.2 Header file structure**



**Figure 1: Header file structure**

**GPT172:** The module shall optionally include the Dem.h file if any production error will be issued by the implementation. By this inclusion the APIs to report errors as well as the required Event Id symbols are included. This specification defines the name of the Event Id symbols, which are provided by XML to the DEM configuration tool. The DEM configuration tool assigns ECU dependent values to the Event Id symbols and publishes the symbols in Dem\_IntErrId.h.

The gray boxes are optional:

**GPT259:** Gpt.h shall include Gpt\_Cfg.h for the API pre-compiler switches

**GPT293:** Gpt.c shall include Gpt.h

Gpt.c has implicit access to the Gpt\_Cfg.h through the Gpt.h file.

**GPT261:** Gpt\_Irq.c shall include Gpt.h for the prototype declaration of the notification functions.

**GPT262:** The file Gpt.h shall contain the type definitions for Gpt\_Lcfc.c and Gpt\_PBcfc.c.

**GPT271:** `Gpt.h` shall include `EcuM_Cbk.h`, if wakeup functionality is configured.

## 6 Requirements traceability

This chapter refers to input requirements specified in the SRS documents (Software Requirements Specifications) that are applicable for this software module.

The table below lists links to specification items of the GPT driver SWS document, which satisfy the input requirements. Only functional requirements are referenced.

Document: General Requirements on Basic Software Modules [3]

<b>Requirement</b>	<b>Satisfied by</b>
[BSW00344] Reference to link-time configuration	<a href="#">GPT184</a>
[BSW00404] Reference to post build-time configuration	Not applicable (no post-build time configurable parameters specified)
[BSW00405] Reference to multiple configuration sets	Not applicable (no post-build time configurable parameters specified)
[BSW00345] Pre-compile-time configuration	<a href="#">GPT183</a>
[BSW159] Tool-based configuration	See Figure 1
[BSW167] Static configuration checking	Not applicable (requirement on configuration tool)
[BSW171] Configurability of optional functionality	<a href="#">GPT182</a> , <a href="#">GPT193</a> , <a href="#">GPT194</a> , <a href="#">GPT195</a> , <a href="#">GPT196</a> , <a href="#">GPT199</a> , <a href="#">GPT200</a> , <a href="#">GPT201</a> , <a href="#">GPT202</a> , <a href="#">GPT203</a>
[BSW170] Data for reconfiguration of AUTOSAR SW-components	Not applicable (requirement on SW component)
[BSW00380] Separate C-File for configuration parameters	See Figure 1
[BSW00419] Separate C-Files for pre-compile time configuration parameters	See Figure 1
[BSW00381] Separate configuration header file for pre-compile time parameters	See Figure 1
[BSW412] Separate H-File for configuration parameters	See Figure 1
[BSW00383] List dependencies of configuration files	See section 10.2
[BSW00384] List dependencies to other modules	See section 10.2
[BSW00387] Specify the configuration class of callback function	See section 10.2
[BSW00388] Introduce containers	<a href="#">GPT183</a> , <a href="#">GPT184</a> , <a href="#">GPT193</a> , <a href="#">GPT235</a> , <a href="#">GPT269</a>
[BSW00389] Containers shall have names	<a href="#">GPT183</a> , <a href="#">GPT184</a> , <a href="#">GPT193</a> , <a href="#">GPT235</a> , <a href="#">GPT269</a>
[BSW00390] Parameter content shall be unique within the module	<a href="#">GPT183</a> , <a href="#">GPT184</a> , <a href="#">GPT193</a> , <a href="#">GPT235</a> , <a href="#">GPT189</a> , <a href="#">GPT269</a>
[BSW00391] Parameter shall have unique names	<a href="#">GPT183</a> , <a href="#">GPT184</a> , <a href="#">GPT193</a> , <a href="#">GPT235</a> , <a href="#">GPT189</a>
[BSW00392] Parameters shall have a type	<a href="#">GPT183</a> , <a href="#">GPT184</a> , <a href="#">GPT193</a> , <a href="#">GPT235</a> , <a href="#">GPT189</a>
[BSW00393] Parameters shall have a range	<a href="#">GPT183</a> , <a href="#">GPT184</a> , <a href="#">GPT193</a> , <a href="#">GPT235</a>
[BSW00394] Specify the scope of the parameters	<a href="#">GPT183</a> , <a href="#">GPT184</a> , <a href="#">GPT193</a> , <a href="#">GPT235</a>
[BSW00395] List the required parameters (per parameter)	<a href="#">GPT240</a> , <a href="#">GPT241</a> , <a href="#">GPT242</a> , see also Figure 10



<b>Requirement</b>	<b>Satisfied by</b>
[BSW00396] Configuration classes	<a href="#">GPT183</a> , <a href="#">GPT184</a> , <a href="#">GPT193</a> , <a href="#">GPT235</a>
[BSW00397] Pre-compile-time parameters	<a href="#">GPT183</a>
[BSW00398] Link-time parameters	<a href="#">GPT184</a>
[BSW00399] Loadable Post-build time parameters	Not applicable (no post-build time configurable parameters specified)
[BSW00400] Selectable Post-build time parameters	Not applicable (no post-build time configurable parameters specified)
[BSW00438] Post Build Configuration Data Structure	Gpt_Init function is called by Ecu State Manager directly
[BSW00402] Published information	<a href="#">GPT189</a>
[BSW00375] Notification of wake-up reason	<a href="#">GPT188</a> , <a href="#">GPT235</a>
[BSW101] Initialization interface	<a href="#">GPT006</a>
[BSW00416] Sequence of initialization	Nor applicable (GPT is not responsible for overall Autosar modules initialization)
[BSW00406] check module initialization	<a href="#">GPT220</a> , <a href="#">GPT221</a> , <a href="#">GPT222</a> , <a href="#">GPT223</a> , <a href="#">GPT224</a> , <a href="#">GPT225</a> , <a href="#">GPT226</a> , <a href="#">GPT227</a> , <a href="#">GPT228</a> , <a href="#">GPT229</a> , <a href="#">GPT230</a>
[BSW00437] Nolnit--Area in RAM	Not applicable (requirement on implementation)
[BSW168] Diagnostic Interface of SW components	Not applicable (requirement on SW components)
[BSW00407] Function to read out published parameters	<a href="#">GPT189</a> , <a href="#">GPT181</a>
[BSW00423] Usage of SW-C template to describe BSW modules with AUTOSAR Interfaces	Not applicable (GPT driver has no Autosar Interface)
[BSW00424] BSW main processing function task allocation	Not applicable (no main processing function specified)
[BSW00425] Trigger conditions for schedulable objects	Not applicable (requirement for the implementer)
[BSW00426] Exclusive areas in BSW modules	Not applicable (applies only for the module description template)
[BSW00427] ISR description for BSW modules	Not applicable (applies only for the module description template)
[BSW00428] Execution order dependencies of main processing functions	Not applicable (no main processing function specified)
[BSW00429] Restricted BSW OS functionality access	Not applicable (requirement for the implementer)
[BSW00431]The BSW Scheduler module implements task bodies	Not applicable (no scheduling functionality in the GPT module)
[BSW00432] Modules should have separate main processing functions for read/receive and write/transmit data path	Not applicable (no main processing function specified)
[BSW00433] Calling of main processing functions	Not applicable (no main processing function specified)
[BSW00434] The Schedule Module shall provide an API for exclusive areas	Not applicable (no scheduling functionality in the GPT module)

<b>Requirement</b>	<b>Satisfied by</b>
[BSW00336] Shutdown interface	<a href="#">GPT008</a>
[BSW00337] Classification of errors	<a href="#">GPT001</a> , <a href="#">GPT004</a>
[BSW00338] Detection and Reporting of development errors	<a href="#">GPT178</a> , <a href="#">GPT204</a>
[BSW00369] Do not return development error codes via API	<a href="#">GPT178</a> , <a href="#">GPT179</a> , <a href="#">GPT204</a>
[BSW00339] Reporting of production relevant error status	<a href="#">GPT179</a>
[BSW00422] Pre-de-bouncing of production relevant error status	Not applicable (requirement on module "Diagnostic Event Manager")
[BSW00417] Reporting of Error Events by Non-Basic Software	Not applicable (applies only for non BSW modules)
[BSW00323] API Parameter checking	<a href="#">GPT001</a> , <a href="#">GPT204</a> , <a href="#">GPT210</a> , <a href="#">GPT211</a> , <a href="#">GPT212</a> , <a href="#">GPT213</a> , <a href="#">GPT214</a> , <a href="#">GPT215</a> , <a href="#">GPT216</a> , <a href="#">GPT217</a> , <a href="#">GPT218</a>
[BSW004] Version check	<a href="#">GPT256</a>
[BSW00409] Header files for production code error IDs	<a href="#">GPT172</a>
[BSW00385] List possible error notifications	<a href="#">GPT001</a> , <a href="#">GPT004</a> , see also Table 4
[BSW00386] Configuration for detecting an error	<a href="#">GPT204</a> , see also Table 5
[BSW161] Microcontroller abstraction	Not applicable (architectural AUTOSAR concept is the basis for this driver)
[BSW162] ECU layout abstraction	Not applicable (architectural AUTOSAR concept is the basis for this driver)
[BSW005] No hard coded horizontal interfaces within MCAL	Not applicable (architectural AUTOSAR concept is the basis for this driver)
[BSW00415] User dependent include files	See Figure 1
[BSW164] Implementation of interrupt service routines	Not applicable (GPT is part of the MCAL and thus is allowed to implement ISRs).
[BSW00325] Runtime of interrupt service routines	Not applicable (requirement on implementation)
[BSW00326] Transition from ISRs to OS tasks	Not applicable (requirement on implementation)
[BSW00342] Usage of source code and object code	Not applicable (requirement on implementation)
[BSW00343] Specification and configuration of time	<a href="#">GPT055</a> , <a href="#">GPT192</a>
[BSW160] Human-readable configuration data	Not applicable (requirement on implementation)
[BSW007] HIS MISRA C	Not applicable (requirement on implementation)
[BSW00300] Module naming convention	See Figure 1
[BSW00413] Accessing instances of BSW modules	Not applicable (requirement on implementation)
[BSW00347] Naming separation of different instances of BSW drivers	Not applicable (requirement on implementation)
[BSW00441] Enumeration literals and #define naming convention	<a href="#">GPT189</a> , <a href="#">GPT001</a> , <a href="#">GPT178</a> , section 8.2.4, <a href="#">GPT185</a> , <a href="#">GPT186</a>
[BSW00305] Self-defined data types naming convention	See section 8.2

<b>Requirement</b>	<b>Satisfied by</b>
[BSW00307] Global variables naming convention	Not applicable (requirement on implementation)
[BSW00310] API naming convention	See section 8.3
[BSW00373] Main processing function naming convention	Not applicable (no main processing function specified)
[BSW00327] Error values naming convention	See Table 4
[BSW00335] Status values naming convention	Not applicable (no status values specified within this SWS)
[BSW00350] Development error detection keyword	<a href="#">GPT183</a>
[BSW00408] Configuration parameter naming convention	See section 10.2
[BSW00410] Compiler switches shall have defined values	See section 10.2
[BSW00411] Get version info keyword	See sections 8.3.1 and 10.2.5
[BSW00346] Basic set of module files	See Figure 1
[BSW158] Separation of configuration from implementation	See Figure 1, <a href="#">GPT183</a> , <a href="#">GPT184</a>
[BSW00314] Separation of interrupt frames and service routines	See Figure 1
[BSW00370] Separation of callback interface from API	See Figure 1
[BSW00435] Module Header File Structure for the Basic Software Scheduler	See Figure 1
[BSW00436] Module Header File Structure for the Basic Software Memory Mapping	See Figure 1
[BSW00348] Standard type header	See Figure 1
[BSW00353] Platform specific type header	See Figure 1
[BSW00361] Compiler specific language extension header	See Figure 1
[BSW00301] Limit imported information	See Figure 1
[BSW00302] Limit exported information	See Figure 1
[BSW00328] Avoid duplication of code	Not applicable (requirement for the implementer)
[BSW00312] Shared code shall be reentrant	See sections 0, 0, 8.3.6, 8.3.7, 0, 0, 0, 0
[BSW006] Platform independency	Not applicable (module is not above MCAL)
[BSW00439] Declaration of interrupt handlers and ISRs	Not applicable (requirement on implementation)
[BSW00357] Standard API return type	Not applicable (this type is not used within this SWS)
[BSW00377] Module specific API return types	Not applicable (this type is not used within this SWS)
[BSW00304] AUTOSAR integer data types	<a href="#">GPT174</a> , sections 8.2 and 10.3
[BSW00355] Do not redefine AUTOSAR integer data types	Not applicable (no integer data types redefined in this specification)
[BSW00378] AUTOSAR boolean type	See section 10.2.2
[BSW00306] Avoid direct use of compiler and platform specific keywords	See Figure 1

<b>Requirement</b>	<b>Satisfied by</b>
[BSW00308] Definition of global data	Not applicable (requirement for the implementer)
[BSW00309] Global data with read-only constraint	See section 0
[BSW00371] Do not pass function pointers via API	Not applicable (no function pointers are passed via API in this SWS.)
[BSW00358] Return type of <code>init()</code> functions	See section 0
[BSW00414] Parameter of <code>init</code> function	<a href="#">GPT257</a>
[BSW00376] Return type and parameters of main processing functions	Not applicable (no main processing function specified)
[BSW00359] Return type of callback functions	See section 8.6
[BSW00360] Parameters of callback functions	See section 8.6
[BSW00440] Function prototype for callback functions of AUTOSAR Services	Not applicable (no AUTOSAR RTE callback services provided)
[BSW00329] Avoidance of generic interfaces	Not applicable (no generic interfaces specified within this SWS)
[BSW00330] Usage of macros / inline functions instead of functions	Not applicable (requirement for the implementer)
[BSW00331] Separation of error and status values	Not applicable (no status values specified within this SWS)
[BSW009] Module User Documentation	See this SWS
[BSW00401] Documentation of multiple instances of configuration parameters	See section 10.2
[BSW172] Compatibility and documentation of scheduling strategy	See section 8.3
[BSW010] Memory resource documentation	Not applicable (requirement for the implementer)
[BSW00333] Documentation of callback function context	See section 0
[BSW00374] Module vendor identification	<a href="#">GPT189</a>
[BSW00379] Module identification	<a href="#">GPT189</a>
[BSW003] Version identification	<a href="#">GPT189</a>
[BSW00318] Format of module version numbers	<a href="#">GPT189</a>
[BSW00321] Enumeration of module version numbers	Not applicable (requirement for the implementer)
[BSW00341] Microcontroller compatibility documentation	Not applicable (requirement for the implementer)
[BSW00334] Provision of XML file	Not applicable (specified by WP4.1.1.2)

Document: General Requirements on SPAL [8]

<b>Requirement</b>	<b>Satisfied by</b>
[BSW12263] Object code compatible configuration concept	<a href="#">GPT184</a>
[BSW12056] Configuration of notification mechanism	<a href="#">GPT208</a>
[BSW12267] Configuration of wakeup sources	<a href="#">GPT188</a> , <a href="#">GPT235</a>
[BSW12057] Driver module initialization	<a href="#">GPT006</a>
[BSW12125] Initialization of hardware resources	<a href="#">GPT068</a>
[BSW12163] Driver module deinitialization	<a href="#">GPT008</a>
[BSW12461] Responsibility for register initialization	<a href="#">GPT205</a> , <a href="#">GPT006</a>
[BSW12462] Provide settings for register initialization	See section 10.3
[BSW12463] Combine and forward settings for register initialization	Not applicable (applies only for configurator)
[BSW12062] Selection of static configuration sets	<a href="#">GPT006</a>
[BSW12068] MCAL initialization sequence	Not applicable (overall module initialization is not triggered by this module)
[BSW12069] Wake-up notification of ECU State Manager	<a href="#">GPT188</a>
[BSW157] Notification mechanisms of drivers and handlers	<a href="#">GPT014</a> , <a href="#">GPT015</a> , <a href="#">GPT232</a>
[BSW12155] Prototypes of callback functions	<a href="#">GPT232</a>
[BSW12169] Control of operation mode	<a href="#">GPT151</a>
[BSW12063] Raw value mode	GPT167, GPT168
[BSW12075] Use of application buffers	Not applicable (no random streaming capability)
[BSW12129] Resetting of interrupt flags	<a href="#">GPT206</a>
[BSW12064] Change of operation mode during running operation	Not applicable see section 0
[BSW12448] Behavior after development error detection	<a href="#">GPT178</a> , <a href="#">GPT234</a> , <a href="#">GPT296</a> , <a href="#">GPT302</a> , <a href="#">GPT084</a> , <a href="#">GPT204</a>
[BSW12067] Setting of wake-up conditions	The GPT HW module has no external interrupt source. The only wakeup condition is the internal timeout. The appropriate notification can be enabled/disabled.  <a href="#">GPT014</a> , <a href="#">GPT015</a> , <a href="#">GPT232</a> , <a href="#">GPT233</a>
[BSW12077] Non-blocking implementation	Not applicable (requirement for the implementer)
[BSW12078] Runtime and memory efficiency	Not applicable (requirement for the implementer)
[BSW12092] Access to drivers	Not applicable (no handler or manager above)
[BSW12265] Configuration data shall be kept constant	Not applicable (requirement for the implementer)
[BSW12264] Specification of configuration items	<a href="#">GPT183</a> , <a href="#">GPT184</a> , <a href="#">GPT193</a> , <a href="#">GPT235</a>

Document: Requirements on GPT Driver [9]

Requirements (module specific)	Satisfied by
[BSW12328] GPT driver time unit	<a href="#">GPT055</a> , GPT192
[BSW12404] Configuration of one-shot/continuous mode	<a href="#">GPT185</a> , <a href="#">GPT186</a>
[BSW12114] Configuration of timer clock source	<a href="#">GPT187</a>
[BSW12460] Configuration of symbolic names for time values	Not applicable (Requirement for configuration tool)
[BSW12116] GPT Deinitialization	<a href="#">GPT008</a> , <a href="#">GPT161</a> , <a href="#">GPT162</a> , <a href="#">GPT308</a>
[BSW12117] Read timer value	<a href="#">GPT083</a> , <a href="#">GPT010</a>
[BSW12128] Start timer	<a href="#">GPT274</a> , <a href="#">GPT275</a> , GPT060
[BSW12119] Stop timer	<a href="#">GPT013</a>
[BSW12120] Provide notification	<a href="#">GPT232</a> , <a href="#">GPT233</a>
[BSW12121] Enable notification	<a href="#">GPT014</a>
[BSW12122] Disable notification	<a href="#">GPT015</a>
[BSW13601] Wakeup functionality	<a href="#">GPT184</a> , <a href="#">GPT151</a> , <a href="#">GPT159</a> , <a href="#">GPT160</a>
[BSW13602] Enable/Disable Wakeup	<a href="#">GPT159</a> , <a href="#">GPT160</a>
[BSW13603] Wake-up mode selection service	<a href="#">GPT151</a> , <a href="#">GPT152</a> , <a href="#">GPT153</a>

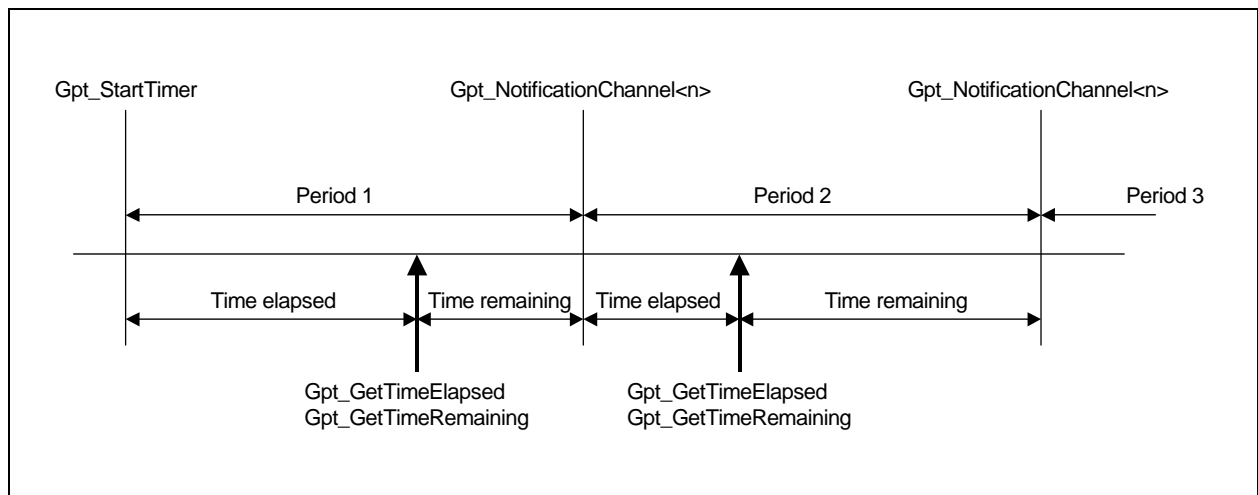
## 7 Functional specification

### 7.1 General behavior

#### 7.1.1 Functional overview

The GPT driver provides services for starting and stopping a functional timer instance (channel) within the hardware timer module. Individual timeout periods (one shot mode) as well as repeating timeout periods (continuous mode) can be generated. The user can configure, if a notification shall be invoked, when the requested timeout period has expired. Notifications can be enabled and disabled at runtime.

Both, the relative time elapsed since the last notification occurred (respectively the channel has been started) and the time remaining until the next notification will occur, can be queried.



**Figure 2: Querying the elapsed/remaining time**

Note: The GPT driver only generates time bases, and does not serve as an event counter. This functionality is provided by another driver module (→ICU driver, see [10]).

The GPT Driver can be used to wakeup the ECU, whenever a predefined timeout period has expired. A mode switching service is provided to switch the GPT Driver between normal operation and sleep mode.

**GPT127:** If supported by hardware and enabled, an internal hardware timer can serve as a wakeup source.

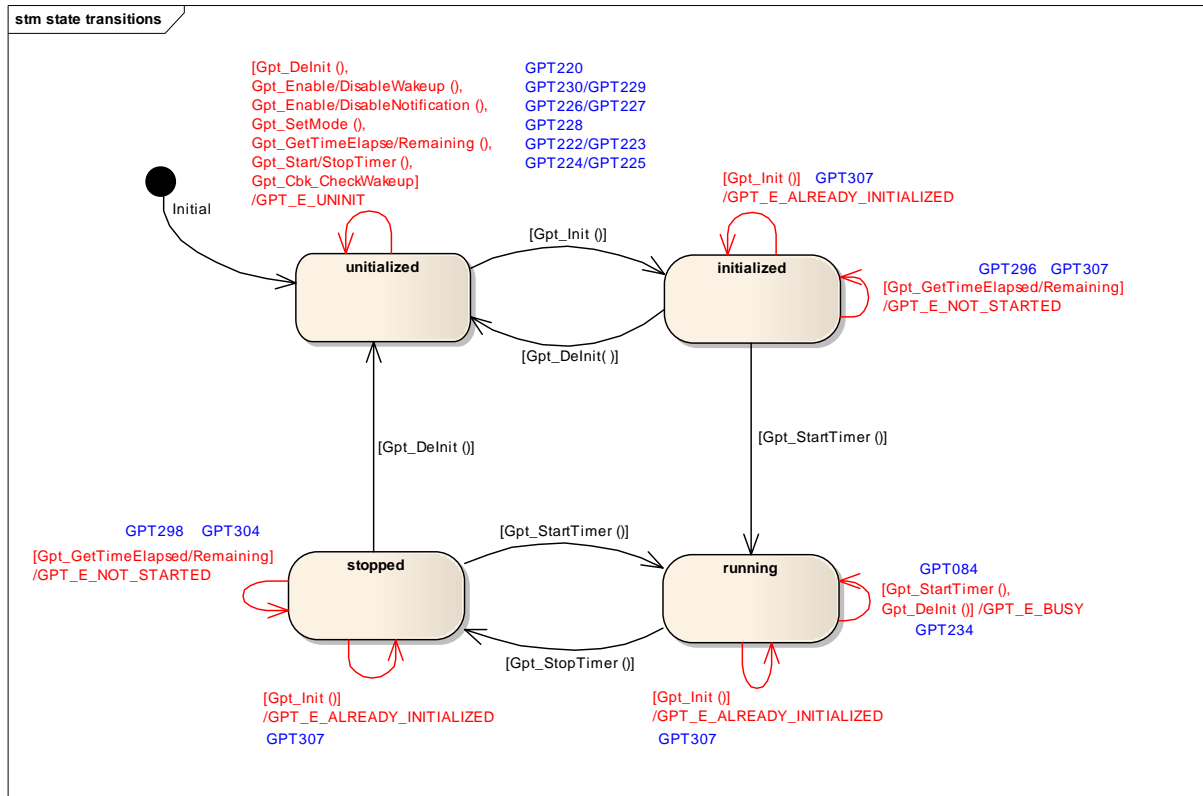
For a detailed description on wakeup handling please refer to the ECU State Manager specification [7].

The driver does not support timeout periods, which exceed the maximum value restricted by the clock source, prescaler and width of the timer register. The user must handle this.



### 7.1.2 State transitions

The state chart below (Figure 3) shows the behavior when calling GPT services on channels in different operational states.



**Figure 3: Channel state transitions**

**Notes:**

- notation: [function call]/error code
- transitions in red font show the behavior with development (DET) error detection turned on.
- the figure only shows the development error detection for functions called in the wrong sequence. It is not reflected in this figure, if i.e. `Gpt_StartTimer()` is called with an invalid parameter.
- if a function call is not explicitly shown in a state, it is assumed that this function call neither leads to a DET report nor to a state transition of the specific channel

### 7.1.3 Version checking

**GPT256:** The Gpt module shall implement compile-time mechanisms that check the following:

For included header files:

- GPT\_AR\_MAJOR\_VERSION
- GPT\_AR\_MINOR\_VERSION

shall be identical.

For the module internal c and h files:

- GPT\_SW\_MAJOR\_VERSION
- GPT\_SW\_MINOR\_VERSION
- GPT\_AR\_MAJOR\_VERSION
- GPT\_AR\_MINOR\_VERSION
- GPT\_AR\_PATCH\_VERSION

shall be identical.

## 7.2 Error classification

**GPT173:** Values for production code Event Ids are assigned externally by the configuration of the Dem. They are published in the file `Dem_IntErrId.h` and included via `Dem.h`.

**GPT174:** Development error values are of type `uint8`.

**GPT001:** The following development errors shall be detectable by the GPT driver depending on its build version (development/production mode). This checking shall be statically configurable (on/off) for those errors that only can occur during development .

<i>Type of error</i>	<i>Relevance</i>	<i>Related error code</i>	<i>Value [hex]</i>
Operational function (getTimeElapsed/ GetTimeRemaining, start/stop timer, enable/disable notification, enable/disable wakeup, set mode, check wakeup) or Gpt_Delnit called prior to init function	Development	GPT_E_UNINIT	0x0A
Function startTimer or delnit called while timer is already running	Development	GPT_E_BUSY	0x0B
Operational function (getTimeElapsed/ getTimeRemaining) called prior to start timer function or after timer has been stopped.	Development	GPT_E_NOT_STARTED	0x0C

Type of error	Relevance	Related error code	Value [hex]
API Gpt_Init service called while the GPT driver has already been initialized	Development	GPT_E_ALREADY_INITIALIZED	0x0D
Operational function (getTimeElapsed/ GetTimeRemaining, start/stop timer, enable/disable notification, enable/disable wakeup) called with invalid channel ID.  Enable/disable wakeup called on a non-wakeup capable channel.	Development	GPT_E_PARAM_CHANNEL	0x14
Function startTimer called with invalid value	Development	GPT_E_PARAM_VALUE	0x15
Gpt_setMode called with invalid mode parameter	Development	GPT_E_PARAM_MODE	0x1F
No production errors assigned	Production	-	

**Table 4: Error classification**

**GPT004:** Additional errors that are detected because of specific implementation and/or specific hardware properties shall be added in the GPT device specific implementation specification. The classification and enumeration shall be compatible to the errors listed above.

### 7.3 Error detection

**GPT175:** The detection of development errors is configurable (STD\_ON/STD\_OFF) at pre-compile time. The switch `GptDevErrorDetect` (see chapter 10) shall activate or deactivate the detection of all development errors.

**GPT176:** If the `GptDevErrorDetect` switch is enabled, API parameter checking is enabled. The detailed description of the detected errors can be found in chapter 7.2 and chapter 8.

**GPT177:** The detection of production code errors cannot be switched off.

**GPT204:** If development error detection is enabled for the GPT Driver, the following API parameter checking shall be performed according to the respective functions (see table below). The error shall be reported to the Development Error Tracer.

Function Gpt_	Criteria of detection	Related error code
GetVersionInfo	None	None
Init	ConfigPtr = NULL called again	GPT_E_PARAM_CONFIG GPT_E_ALREADY_INITIALIZED
Delnit	called prior to initialization called while timer is running	GPT_E_UNINIT GPT_E_BUSY
GetTimeElapsed	called prior to initialization called prior to starting the timer channel	GPT_E_UNINIT GPT_E_NOT_STARTED

Function Gpt_	Criteria of detection	Related error code
	called on a stopped channel channel out of range	GPT_E_NOT_STARTED GPT_E_PARAM_CHANNEL
GetTimeRemaining	called prior to initialization called prior to start. timer channel called on a stopped channel channel out of range	GPT_E_UNINIT GPT_E_NOT_STARTED GPT_E_NOT_STARTED GPT_E_PARAM_CHANNEL
StartTimer	called prior to initialization called while channel is already running Channel out of range Passed timer value out of range	GPT_E_UNINIT GPT_E_BUSY GPT_E_PARAM_CHANNEL GPT_E_PARAM_VALUE
StopTimer	called prior to initialization channel out of range	GPT_E_UNINIT GPT_E_PARAM_CHANNEL
EnableNotification	called prior to initialization channel out of range	GPT_E_UNINIT GPT_E_PARAM_CHANNEL
DisableNotification	called prior to initialization channel out of range	GPT_E_UNINIT GPT_E_PARAM_CHANNEL
SetMode	called prior to initialization called with invalid mode param.	GPT_E_UNINIT GPT_E_PARAM_MODE
EnableWakeup	called prior to initialization channel out of range called on a non-wakeup capable channel	GPT_E_UNINIT GPT_E_PARAM_CHANNEL GPT_E_PARAM_CHANNEL
DisableWakeup	called prior to initialization channel out of range called on a non-wakeup capable channel	GPT_E_UNINIT GPT_E_PARAM_CHANNEL GPT_E_PARAM_CHANNEL
Cbk_CheckWakeup	called prior to initialization	GPT_E_UNINIT

**Table 5: Error detection**

## 7.4 Error notification

**GPT178:** Detected development errors shall be reported to the `Det_ReportError` service of the Development Error Tracer (DET[4]) if the preprocessor switch `GptDevErrorDetect` is set (see chapter 10).

**GPT179:** Production errors shall be reported to Diagnostic Event Manager[6].

## 8 API specification

### 8.1 Imported types

In this chapter all types included from the following files are listed:

#### GPT278:

Header file	Imported Type
Dem_Types.h	Dem_EventIdType
EcuM_Types.h	EcuM_WakeupSourceType
Std_Types.h	Std_VersionInfoType

### 8.2 Type Definitions

#### 8.2.1 Gpt\_ConfigType

<b>Name:</b>	Gpt_ConfigType	
<b>Type:</b>	Structure	
<b>Range:</b>	--	Implementation specific configuration data structure, see 10 for configurable parameters.
<b>Description:</b>	This is the type of the data structure including the configuration set required for initializing the GPT timer unit.	

#### 8.2.2 Gpt\_ChannelType

<b>Name:</b>	Gpt_ChannelType	
<b>Type:</b>	uint8, uint16, uint32	
<b>Range:</b>	--	Implementation specific. But not all values may be valid within this type. This type shall be chosen in order to have the most efficient implementation on a specific micro controller platform.
<b>Description:</b>	Numeric ID of a GPT channel.	

#### 8.2.3 Gpt\_ValueType

<b>Name:</b>	Gpt_ValueType	
<b>Type:</b>	uint8, uint16, uint32	
<b>Range:</b>	--	The range of this type is $\mu$ C dependent (width of the timer register) and has to be described by the supplier.
<b>Description:</b>	Used for reading the current timer value/setting periodic timer values (in number of	

	ticks) up to hours.
--	---------------------

## 8.2.4 Gpt\_ModeType

<b>Name:</b>	Gpt_ModeType	
<b>Type:</b>	Enumeration	
<b>Range:</b>	GPT_MODE_NORMAL	Normal operation mode of the GPT
	GPT_MODE_SLEEP	Operation for reduced power operation mode. In Wakeup mode only wakeup capable channels are available.
<b>Description:</b>	Allows the selection of different power modes.	

## 8.3 Function definitions

This is a list of functions provided for upper layer modules.

**GPT055:** All time units used within the API services of the GPT driver shall be of the unit ticks.

To get times out of register values it is necessary to know the oscillator frequency, prescalers and so on. Since these settings are made in MCU and/or in other modules it is not possible to calculate such times.

Hence the conversions between time and ticks shall be part of an upper layer.

### 8.3.1 Gpt\_GetVersionInfo

#### GPT279:

<b>Service name:</b>	Gpt_GetVersionInfo	
<b>Syntax:</b>	<pre>void Gpt_GetVersionInfo(     Std_VersionInfoType* versioninfo )</pre>	
<b>Service ID[hex]:</b>	0x00	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	None	
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	versioninfo	Pointer to where to store the version information of this module.
<b>Return value:</b>	None	
<b>Description:</b>	Returns the version information of this module.	

**GPT181:** The function `Gpt_GetVersionInfo` shall return the version information of this module. The version information includes:

- Module Id
- Vendor Id
- Vendor specific version numbers (BSW00407).

**GPT273:** If source code for caller and callee of `Gpt_GetVersionInfo` is available, the GPT module should realize `Gpt_GetVersionInfo` as a macro, defined in the module's header file.

**GPT182:** The function `Gpt_GetVersionInfo` shall be pre compile time configurable On/Off by the configuration parameter: `GptVersionInfoApi`

### 8.3.2 Gpt\_Init

#### GPT280:

<b>Service name:</b>	Gpt_Init	
<b>Syntax:</b>	<pre>void Gpt_Init(     const Gpt_ConfigType* configPtr )</pre>	
<b>Service ID[hex]:</b>	0x01	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Non Reentrant	
<b>Parameters (in):</b>	configPtr	Pointer to a selected configuration structure
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Initializes the hardware timer module.	

**GPT006:** The function `Gpt_Init` shall initialize the hardware timer module according to a configuration set referenced by `ConfigPtr`.

**GPT272:** For variants with no postbuild multiple selectable configuration parameters (Variant PC), the GPT module's environment shall pass a `NULL` pointer to the function `Gpt_Init` (see also [GPT257](#)).

**GPT107:** The function `Gpt_Init` shall disable all notifications.

**GPT068:** The function `Gpt_Init` shall only initialize the configured resources. Resources that are not configured in the configuration file shall not be touched.

**GPT205:** The GPT Driver shall apply the following rules regarding initialization of controller registers:

- [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 I/O register it shall be initialized by the PORT driver
- [3] If the register can affect several hardware modules and if it is not an I/O register it shall be initialized by the MCU driver
- [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

**GPT307:** If development error detection for the GPT module is enabled: if called when the GPT driver and hardware are already initialized, the function `Gpt_Init` shall raise development error `GPT_E_ALREADY_INITIALIZED` and return without any action.

**GPT258:** The function `Gpt_Init` shall disable the wakeup interrupt invocation of all channels after the function call.

**GPT294:** If development error detection for the GPT module is enabled: if the function `Gpt_Init` is called with a `NULL configPtr` and if a variant containing postbuild multiple selectable configuration parameters is used (Variant PB), the function `Gpt_Init` shall raise the development error `GPT_E_PARAM_CONFIG` and return without any action.

**GPT309:** A re-initialization of the GPT driver by executing the `Gpt_Init()` function requires a de-initialization before by executing a `Gpt_DeInit()`.

### 8.3.3 Gpt\_DeInit

#### GPT281:

<b>Service name:</b>	Gpt_DeInit
<b>Syntax:</b>	void Gpt_DeInit(  )
<b>Service ID[hex]:</b>	0x02
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Non Reentrant
<b>Parameters (in):</b>	None
<b>Parameters (inout):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	Deinitializes all hardware timer channels.

**GPT008:** The function `Gpt_DeInit` shall deinitialize all hardware timer channels used by the configuration to their power on reset state. It's the responsibility of the hardware design that the state does not lead to undefined activities in the  $\mu$ C.

**GPT161:** Values of registers which are not writable shall be excluded by the function `Gpt_DeInit`.

**GPT105:** The function `Gpt_DeInit` shall disable all notifications.

**GPT162:** The function `Gpt_DeInit` shall influence only the peripherals, which are allocated by the static configuration.

**GPT308:** If a postbuild multiple selectable configuration variant was used, the function `Gpt_DeInit` shall further influence only the peripherals, which are allocated by the runtime configuration set passed by the previous call of the function `Gpt_Init()`.



The function `Gpt_DeInit` shall influence only the peripherals, which are allocated by the static configuration.

**GPT194:** The function `Gpt_DeInit` shall be pre compile time configurable On/Off by the configuration parameter: `GptDeinitApi`.

**GPT234:** If development error detection for the GPT module is enabled: if the function `Gpt_DeInit` is called and if any channel is in state running, the function `Gpt_DeInit` shall raise the development error `GPT_E_BUSY` and leave the desired deinitialization functionality without any action.

**GPT220:** If development error detection for the GPT module is enabled: if the function `Gpt_DeInit` is called before the GPT module was initialized, the function `Gpt_DeInit` shall raise the development error `GPT_E_UNINIT` and leave the desired deinitialization functionality without any action.

### 8.3.4 Gpt\_GetTimeElapsed

#### GPT282:

<b>Service name:</b>	Gpt_GetTimeElapsed	
<b>Syntax:</b>	Gpt_ValueType Gpt_GetTimeElapsed( Gpt_ChannelType channel )	
<b>Service ID[hex]:</b>	0x03	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	channel	Numeric identifier of the GPT channel.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Gpt_ValueType	Elapsed timer value (in number of ticks)
<b>Description:</b>	Gets the time already elapsed.	

**GPT010:** The function `Gpt_GetTimeElapsed` shall query the time already elapsed. When the channel is in the mode “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). When the channel is in the mode “continuous mode”, the function `Gpt_GetTimeElapsed` shall return the timer value relative to the last timeout / the start of the channel .

**GPT295:** If the function `Gpt_GetTimeElapsed` is called prior to starting the specified timer channel, the function `Gpt_GetTimeElapsed` shall return the value “0”.

**GPT297:** If the function `Gpt_GetTimeElapsed` is called on a timer channel in stopped state (channel has been initialized, started and stopped by `Gpt_StopTimer`), the function `Gpt_GetTimeElapsed` shall return the value “0”.

The rationale of GPT295 and GPT297 is to have the same behaviour for a stopped channel like a never started channel.

**GPT299:** If the function `Gpt_GetTimeElapsed` is called on a channel configured for one shot mode after the timeout period has already expired, the function `Gpt_GetTimeElapsed` shall return the value "0".

**GPT113:** The GPT module's environment shall only use the re-entrant capability of the function `Gpt_GetTimeElapsed` if the GPT module's environment takes care that there is no simultaneous usage of the same channel.

**GPT195:** The function `Gpt_GetTimeElapsed` shall be pre compile time configurable On/Off by the configuration parameter: `GptTimeElapsedApi`.

**GPT222:** If development error detection for the GPT module is enabled: if the function `Gpt_GetTimeElapsed` is called before the GPT module was initialized, the function `Gpt_GetTimeElapsed` shall raise the development error `GPT_E_UNINIT` and return with the value "0".

**GPT210:** If development error detection for the GPT module is enabled: if the parameter "channel" is invalid, the function `Gpt_GetTimeElapsed` shall raise the development error `GPT_E_PARAM_CHANNEL` and return with the value "0".

**GPT296:** If development error detection for the GPT module is enabled: if the function `Gpt_GetTimeElapsed` is called prior to starting the specified timer channel, the function `Gpt_GetTimeElapsed` shall raise the development error `GPT_E_NOT_STARTED` and behave according to GPT295.

**GPT298:** If development error detection for the GPT module is enabled: if the function `Gpt_GetTimeElapsed` is called on a stopped timer channel (channel has been initialized and started before), the function `Gpt_GetTimeElapsed` shall raise the development error `GPT_E_NOT_STARTED` and behave according to GPT297.

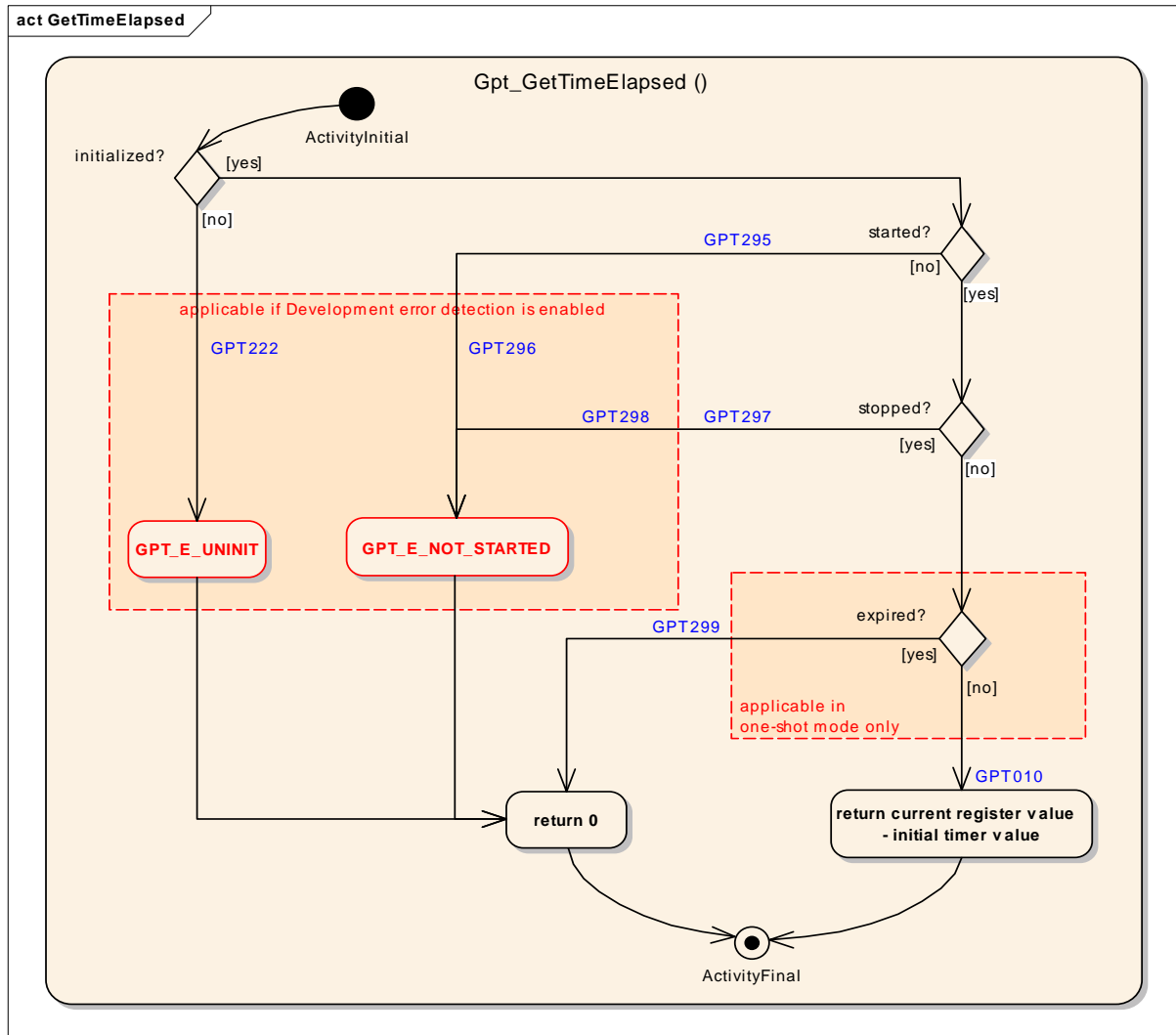


Figure 4: Overview on return and error values of `Gpt_GetTimeElapsed`

Note that a continuous channel never expires and thus no return/error values are defined for that special case.

### 8.3.5 Gpt\_GetTimeRemaining

#### GPT283:

<b>Service name:</b>	Gpt_GetTimeRemaining	
<b>Syntax:</b>	Gpt_ValueType Gpt_GetTimeRemaining( Gpt_ChannelType channel )	
<b>Service ID[hex]:</b>	0x04	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	channel	Numeric identifier of the GPT channel.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	Gpt_ValueType	Remaining timer value (in number of ticks)

<b>Description:</b>	Gets the time remaining until the next timeout period will expire.
---------------------	--

**GPT083:** The function `Gpt_GetTimeRemaining` shall return 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).

**GPT301:** If the function `Gpt_GetTimeRemaining` is called prior to starting the specified timer channel, the function `Gpt_GetTimeRemaining` shall return the value "0".

**GPT303:** If the function `Gpt_GetTimeRemaining` is called on a stopped timer channel (channel has been initialized and started before), the function `Gpt_GetTimeRemaining` shall return the value "0".

The rationale of [GPT301](#) and [GPT303](#) is to have the same behaviour for a stopped channel like a never started channel.

**GPT305:** If the function `Gpt_GetTimeRemaining` is called on a channel configured for one shot mode, after the timeout period has already expired, the function `Gpt_GetTimeRemaining` shall return the value "0".

**GPT114:** The GPT module's environment shall only use the re-entrant capability of the function `Gpt_GetTimeRemaining` if the GPT module's environment takes care that there is no simultaneous usage of the same channel.

**GPT196:** The function `Gpt_GetTimeRemaining` shall be pre compile time configurable On/Off by the configuration parameter: `GptTimeRemainingApi`.

**GPT223:** If development error detection for the GPT module is enabled: if the function `Gpt_GetTimeRemaining` is called before the GPT module was initialized, the function `Gpt_GetTimeRemaining` shall raise the development error `GPT_E_UNINIT` and return with the value "0".

**GPT211:** If development error detection for the GPT module is enabled: if the parameter `channel` is not within the allowed range (as specified by configuration), the function `Gpt_GetTimeRemaining` shall raise the development error `GPT_E_PARAM_CHANNEL` and return with the value "0".

**GPT302:** If development error detection for the GPT module is enabled: if the function `Gpt_GetTimeRemaining` is called prior to starting the specified timer channel, the function `Gpt_GetTimeRemaining` shall raise the development error `GPT_E_NOT_STARTED` and behave according to [GPT301](#).

**GPT304:** If development error detection for the GPT module is enabled: if the function `Gpt_GetTimeRemaining` is called on a stopped timer channel (channel has been initialized and started before), the function `Gpt_GetTimeRemaining` shall

raise the development error GPT\_E\_NOT\_STARTED and behave according to GPT303.

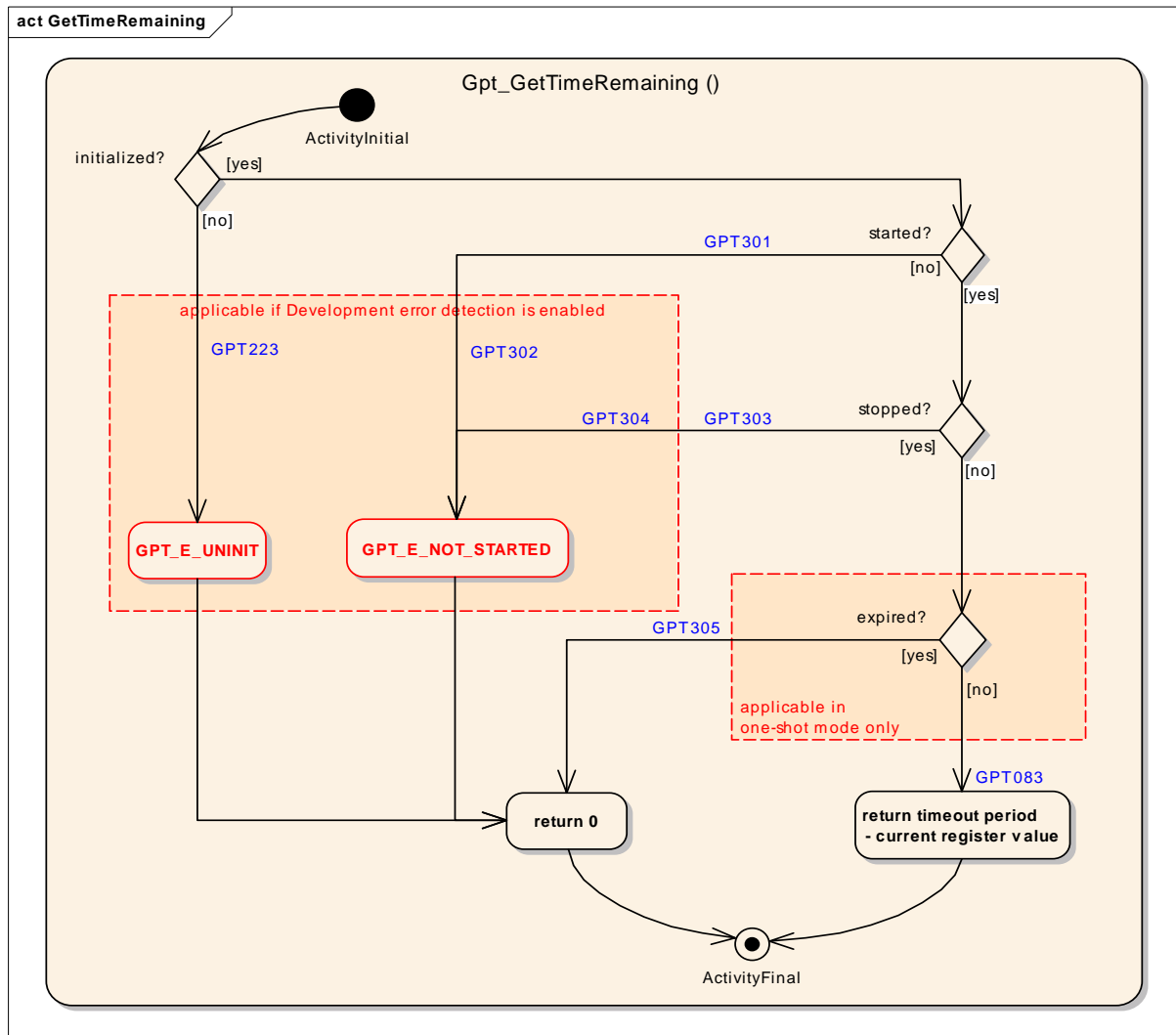


Figure 5: Overview on return and error values of Gpt\_GetTimeRemaining

Note that a continuous channel never expires and thus no return/error values are defined for that special case.

### 8.3.6 Gpt\_StartTimer

**GPT284:**

<b>Service name:</b>	Gpt_StartTimer
<b>Syntax:</b>	void Gpt_StartTimer( Gpt_ChannelType channel, Gpt_ValueType value )
<b>Service ID[hex]:</b>	0x05
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Reentrant
<b>Parameters (in):</b>	channel   Numeric identifier of the GPT channel.

	value	Timeout period (in number of ticks) after a notification shall occur.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Starts a timer channel.	

**GPT274:** The function `Gpt_StartTimer` shall start the selected timer channel with a defined timeout period.

**GPT275:** The function `Gpt_StartTimer` shall invoke the configured notification for that channel (see also GPT292) after the timeout period referenced via the parameter `value` (if enabled).

**GPT115:** The GPT module's environment shall only use the re-entrant capability of the function `Gpt_StartTimer` if the GPT module's environment takes care that there is no simultaneous usage of the same channel.

**GPT212:** If development error detection for the GPT module is enabled: the function `Gpt_StartTimer` shall raise the development error `GPT_E_PARAM_CHANNEL` if the parameter `channel` is not within the allowed range (as specified by configuration)..

**GPT218:** If development error detection for the GPT module is enabled: the function `Gpt_StartTimer` shall raise the development error `GPT_E_PARAM_VALUE` if the parameter `value` is not within the allowed range (exceeding the maximum timer resolution).

**GPT224:** If development error detection for the GPT module is enabled: if the function `Gpt_StartTimer` is called before the GPT module was initialized, the function `Gpt_StartTimer` shall raise the development error `GPT_E_UNINIT`.

**GPT084:** If development error detection for the GPT module is enabled: if the function `Gpt_StartTimer` is called on a channel, which is already started and still running, the function `Gpt_StartTimer` shall raise the development error `GPT_E_BUSY` and return without any action.

### 8.3.7 Gpt\_StopTimer

**GPT285:**

<b>Service name:</b>	Gpt_StopTimer	
<b>Syntax:</b>	<pre>void Gpt_StopTimer(     Gpt_ChannelType channel )</pre>	
<b>Service ID[hex]:</b>	0x06	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	channel	Numeric identifier of the GPT channel.
<b>Parameters (inout):</b>	None	

<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	Stops a timer channel.

**GPT013:** The function `Gpt_StopTimer` shall stop the selected timer channel .

**GPT099:** The function `Gpt_StopTimer` shall not raise a development error when the function `Gpt_StopTimer` stops a timer channel, which has not been started before.

**GPT103:** Timer channels configured in one shot mode have to be stopped explicitly by the user, when the timeout period has expired.

**GPT116:** The GPT module's environment shall only use the re-entrant capability of the function `Gpt_StopTimer` if the GPT module's environment takes care that there is no simultaneous usage of the same channel.

**GPT213:** If development error detection for the GPT module is enabled: if the parameter `channel` is not within the allowed range (as specified by configuration) , the function `Gpt_StopTimer` shall raise the development error `GPT_E_PARAM_CHANNEL`.

**GPT225:** If development error detection for the GPT module is enabled: if the function `Gpt_StopTimer` is called before the GPT module was initialized, the function `Gpt_StopTimer` shall raise the development error `GPT_E_UNINIT`.

### 8.3.8 Gpt\_EnableNotification

#### GPT286:

<b>Service name:</b>	Gpt_EnableNotification	
<b>Syntax:</b>	<pre>void Gpt_EnableNotification(     Gpt_ChannelType channel )</pre>	
<b>Service ID[hex]:</b>	0x07	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	channel	Numeric identifier of the GPT channel.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Enables the notification for a channel.	

**GPT014:** The function `Gpt_EnableNotification` shall enable the invocation of the configured notification function (see also [GPT233](#)) for a channel.

**GPT117:** The GPT module's environment shall only use the re-entrant capability of the function `Gpt_EnableNotification` if the GPT module's environment takes care that there is no simultaneous usage of the same channel.

**GPT199:** The function `Gpt_EnableNotification` shall be pre compile time configurable On/Off by the configuration parameter: `GptEnableDisableNotificationApi`

**GPT226:** If development error detection for the GPT module is enabled: if the function `Gpt_EnableNotification` is called before the GPT module was initialized, the function `Gpt_EnableNotification` shall raise the development error `GPT_E_UNINIT`.

**GPT214:** If development error detection for the GPT module is enabled: if the parameter `channel` is not within the allowed range (as specified by configuration), the function `Gpt_EnableNotification` shall raise the development error `GPT_E_PARAM_CHANNEL`.

### 8.3.9 Gpt\_DisableNotification

#### GPT287:

<b>Service name:</b>	Gpt_DisableNotification	
<b>Syntax:</b>	<pre>void Gpt_DisableNotification(     Gpt_ChannelType channel )</pre>	
<b>Service ID[hex]:</b>	0x08	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	channel	Numeric identifier of the GPT channel.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Disables the notification for a channel.	

**GPT015:** The function `Gpt_DisableNotification` shall disable the invocation of the configured notification function (see also [GPT233](#)) for a channel.

**GPT118:** The GPT module's environment shall only use the re-entrant capability of the function `Gpt_DisableNotification` if the GPT module's environment takes care that there is no simultaneous usage of the same channel.

**GPT200:** The function `Gpt_DisableNotification` shall be pre compile time configurable On/Off by the configuration parameter: `GptEnableDisableNotificationApi`.

**GPT227:** If development error detection for the GPT module is enabled: if the function `Gpt_DisableNotification` is called before the GPT module was initialized, the function `Gpt_DisableNotification` shall raise the development error `GPT_E_UNINIT`.



**GPT217:** If development error detection for the GPT module is enabled: if the parameter `channel` is not within the allowed range (as specified by configuration), the function `Gpt_DisableNotification` shall raise the development error `GPT_E_PARAM_CHANNEL`.

### 8.3.10 Gpt\_SetMode

#### GPT288:

<b>Service name:</b>	Gpt_SetMode
<b>Syntax:</b>	void Gpt_SetMode( Gpt_ModeType mode )
<b>Service ID[hex]:</b>	0x09
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	Non Reentrant
<b>Parameters (in):</b>	mode GPT_MODE_NORMAL: Normal operation mode of the GPT enabled. GPT_MODE_SLEEP: Operation for reduced power operation mode. In Wakeup mode only wakeup capable channels are capable of generating interrupts. See also Gpt_ModeType.
<b>Parameters (inout):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	Sets the operation mode of the GPT.

**GPT151:** The function `Gpt_SetMode` shall set the operation mode to the given mode parameter .

**GPT255:** The function `Gpt_SetMode` is only feasible if `GptReportWakeupSource` is statically configured available.

**GPT152:** If the parameter `mode` has the value `GPT_MODE_NORMAL`, the function `Gpt_SetMode` shall not affect the notifications as configured and selected by the `Gpt_DisableNotification` and `Gpt_EnableNotification`.

**GPT153:** If the parameter `mode` has the value `GPT_MODE_SLEEP`, the function `Gpt_SetMode` shall only enable the interrupts for those channels which are configured as wakeup capable and which are not disabled via the function `Gpt_DisableWakeup`. The function `Gpt_SetMode` shall disable all other interrupts and must not lead to an exit from the reduced power mode state (e.g. idle, halt) of the MCU if the wakeup timer expires.

**GPT164:** If the parameter `mode` has the value `GPT_MODE_SLEEP`, the function `Gpt_SetMode` shall stop all non-wakeup capable timer channels. Only those channels, which can serve as a wakeup source are running.

**GPT165:** If the parameter `mode` has the value `GPT_MODE_NORMAL` and the current mode is `GPT_MODE_SLEEP`, the function `Gpt_SetMode` shall not restart automatically the timer channels which have been stopped by entering the sleep mode.

**GPT228:** If development error detection for the GPT module is enabled: if the function `Gpt_SetMode` is called before the GPT module was initialized, the function `Gpt_SetMode` shall raise the development error `GPT_E_UNINIT`.

**GPT231:** If development error detection for the GPT module is enabled: the function `Gpt_SetMode` shall raise the development error `GPT_E_PARAM_MODE` if the parameter `mode` is invalid.

**GPT201:** The function `Gpt_SetMode` shall be pre compile time configurable On/Off by the configuration parameter: `GPT_WAKEUP_FUNCTIONALITY_API` (see 10.2.5)

The function `Gpt_SetMode` influences the functionality of the GPT channels. Therefore the mode switching of the module shall be compatible to the overall state of the ECU.

The function `Gpt_SetMode` is affected by the configuration parameter `GptReportWakeupSource`.

### 8.3.11 Gpt\_DisableWakeup

#### GPT289:

<b>Service name:</b>	Gpt_DisableWakeup	
<b>Syntax:</b>	<pre>void Gpt_DisableWakeup(     Gpt_ChannelType channel )</pre>	
<b>Service ID[hex]:</b>	0x0a	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	channel	Numeric identifier of the GPT channel.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Disables the wakeup interrupt invocation of a channel.	

**GPT159:** The function `Gpt_DisableWakeup` shall disable the wakeup interrupt invocation of a single GPT channel, referenced by the parameter `channel`.

**GPT157:** The function `Gpt_DisableWakeup` is only feasible, if `GptReportWakeupSource` is statically configured available.

**GPT155:** The GPT module's environment shall only use the re-entrant capability of the function `Gpt_DisableNotification` if the GPT module's environment takes care that there is no simultaneous usage of the same channel.

**GPT202:** The function `Gpt_DisableWakeup` shall be pre compile time configurable On/Off by the configuration parameter: `GptWakeupFunctionalityApi`

**GPT215:** If development error detection for the GPT module is enabled: the function `Gpt_DisableWakeup` shall raise the development error `GPT_E_PARAM_CHANNEL` if the parameter `channel` is not within the allowed range (as specified by configuration) or on a non-wakeup capable channel.

**GPT229:** If development error detection for the GPT module is enabled: if the function `Gpt_DisableWakeup` is called before the GPT module was initialized, the function `Gpt_DisableWakeup` shall raise the development error `GPT_E_UNINIT`.

### 8.3.12 Gpt\_EnableWakeup

#### GPT290:

<b>Service name:</b>	Gpt_EnableWakeup	
<b>Syntax:</b>	<pre>void Gpt_EnableWakeup(     Gpt_ChannelType channel )</pre>	
<b>Service ID[hex]:</b>	0x0b	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	channel	Numeric identifier of the GPT channel.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Enables the wakeup interrupt invocation of a channel.	

**GPT160:** The function `Gpt_EnableWakeup` shall re-enable the wakeup interrupt invocation of a single GPT channel, referenced by the parameter `channel`.

**GPT158:** The function `Gpt_EnableWakeup` is only feasible, if `GptReportWakeupSource` is statically configured available.

**GPT156:** The GPT module's environment shall only use the re-entrant capability of the function `Gpt_EnableWakeup` if the GPT module's environment takes care that there is no simultaneous usage of the same channel.

**GPT203:** The function `Gpt_EnableWakeup` shall be pre compile time configurable On/Off by the configuration parameter: `GptWakeupFunctionalityApi` (see 10.2.5)

**GPT230:** If development error detection for the GPT module is enabled: if the function `Gpt_EnableWakeup` is called before the GPT module was initialized, the function `Gpt_EnableWakeup` shall raise the development error `GPT_E_UNINIT`.

**GPT216:** If development error detection for the GPT module is enabled: the function `Gpt_EnableWakeup` shall raise the development error `GPT_E_PARAM_CHANNEL` if the parameter `channel` is not within the allowed range (as specified by configuration) or on a non-wakeup capable channel.

### 8.3.13 Gpt\_Cbk\_CheckWakeup

#### GPT328:

<b>Service name:</b>	Gpt_Cbk_CheckWakeup	
<b>Syntax:</b>	<pre>void Gpt_Cbk_CheckWakeup(     EcuM_WakeupSourceType wakeupSource )</pre>	
<b>Service ID[hex]:</b>	0x0c	
<b>Sync/Async:</b>	Synchronous	
<b>Reentrancy:</b>	Reentrant	
<b>Parameters (in):</b>	wakeupSource	Information on wakeup source to be checked. The associated GPT channel can be determined from configuration data.
<b>Parameters (inout):</b>	None	
<b>Parameters (out):</b>	None	
<b>Return value:</b>	None	
<b>Description:</b>	Checks if a wakeup capable GPT channel is the source for a wakeup event and calls the ECU state manager service <code>EcuM_SetWakeupEvent</code> in case of a valid GPT channel wakeup event.	

**GPT321:** The function `Gpt_Cbk_CheckWakeup` shall check if a wakeup capable GPT channel is the source for a wakeup event and call `EcuM_SetWakeupEvent` to indicate a valid timer wakeup event to the ECU State Manager [7].

**GPT322:** The function `Gpt_Cbk_CheckWakeup` is only feasible, if `GptReportWakeupSource` is statically configured available.

**GPT323:** The GPT module's environment shall only use the re-entrant capability of the function `Gpt_Cbk_CheckWakeup` if the GPT module's environment takes care that there is no simultaneous usage of the same channel.

**GPT324:** The function `Gpt_Cbk_CheckWakeup` shall be pre compile time configurable On/Off by the configuration parameter: `GptWakeupFunctionalityApi` (see 10.2.5)

**GPT325:** If development error detection for the GPT module is enabled: if the function `Gpt_Cbk_CheckWakeup` is called before the GPT module was initialized, the function `Gpt_Cbk_CheckWakeup` shall raise the development error `GPT_E_UNINIT`.

## 8.4 Call-back Notifications

Since the GPT is a driver module it doesn't provide any callback functions for lower layer modules.

## 8.5 Scheduled functions

None.

## 8.6 Expected Interfaces

In this chapter all interfaces required from other modules are listed.

### 8.6.1 Mandatory Interfaces

This chapter defines all interfaces, which are required to fulfill the core functionality of the module.

None.

### 8.6.2 Optional Interfaces

This chapter defines all interfaces, which are required to fulfill an optional functionality of the module.

GPT291:

<i>API function</i>	<i>Description</i>
EcuM_SetWakeupEvent	Sets the wakeup event.
Dem_ReportErrorStatus	Reports errors to the DEM.
Det_ReportError	Service to report development errors.
EcuM_CheckWakeup	This callout is called by the EcuM to poll a wakeup source. It shall also be called by the ISR of a wakeup source to set up the PLL and check other wakeup sources that may be connected to the same interrupt.

**GPT326:** EcuM\_CheckWakeup shall be called within the Interrupt Service Routine, servicing the GPT channel wakeup event on wakeup-capable channels.

**GPT327:** The ISR's, providing the wakeup events, shall be responsible for resetting the interrupt flags (if needed by hardware).

### 8.6.3 Configurable Interfaces

In this chapter all interfaces are listed where the target function could be configured. The target function is usually a call-back function. The names of these kinds of interfaces is not fixed because they are configurable.

### 8.6.3.1 GptNotification

#### GPT292:

<b>Service name:</b>	Gpt_Notification_<channel>
<b>Syntax:</b>	void Gpt_Notification_<channel>(  )
<b>Sync/Async:</b>	Synchronous
<b>Reentrancy:</b>	GPT user implementation dependant.
<b>Parameters (in):</b>	None
<b>Parameters (inout):</b>	None
<b>Parameters (out):</b>	None
<b>Return value:</b>	None
<b>Description:</b>	--

**GPT232** notification prototype `GptNotification_<channel>` is for the notification callback function and shall be implemented by the user.

**GPT207:** The callback notifications shall be configurable as function pointers within the initialization data structure (`Gpt_ConfigType`).

**GPT086:** The callback notifications `GptNotification_<channel>` shall be configurable as pointers to user defined functions within the configuration structure .

**GPT209:** Each channel shall provide its own notification if configured.

**GPT087:** The GPT module's environment shall declare a separate notification for each channel to avoid parameter values and to improve runtime efficiency.

**GPT208:** If a callback notification is configured as null pointer, no callback shall be executed.

**GPT093:** When disabled, the GPT Driver will send no notification. When re-enabled again, the user will not be notified of events, occurred while notifications have been disabled.

**GPT233** The GPT Driver shall invoke a notification whenever the defined time period of the channel has expired.

**GPT206:** The ISR's, providing the timeout period events, shall be responsible for resetting the interrupt flags (if needed by hardware) and calling the according notification function.

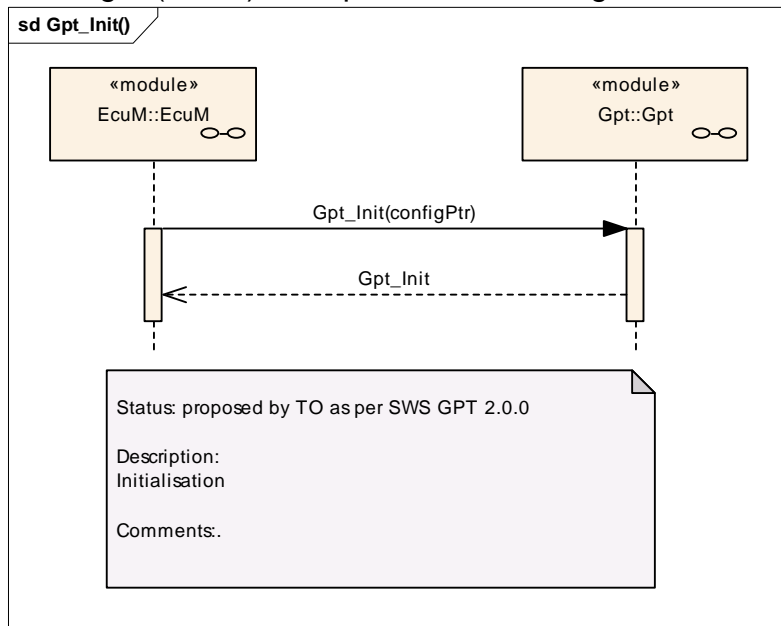
For all available channels, callback functions have to be declared by the configuration tool (see chapter 10).

## 9 Sequence diagrams

All functions except `Gpt_Init`, `Gpt_DeInit`, `Gpt_GetVersionInfo` and `Gpt_SetMode` are synchronous and re-entrant.

### 9.1 Gpt\_Init

The ECU State Manager (EcuM) is responsible for calling the init function.



**Figure 6: Sequence Diagram - Gpt\_Init**

## 9.2 GPT continuous mode

Channel 2 is configured as “Continuous Mode”

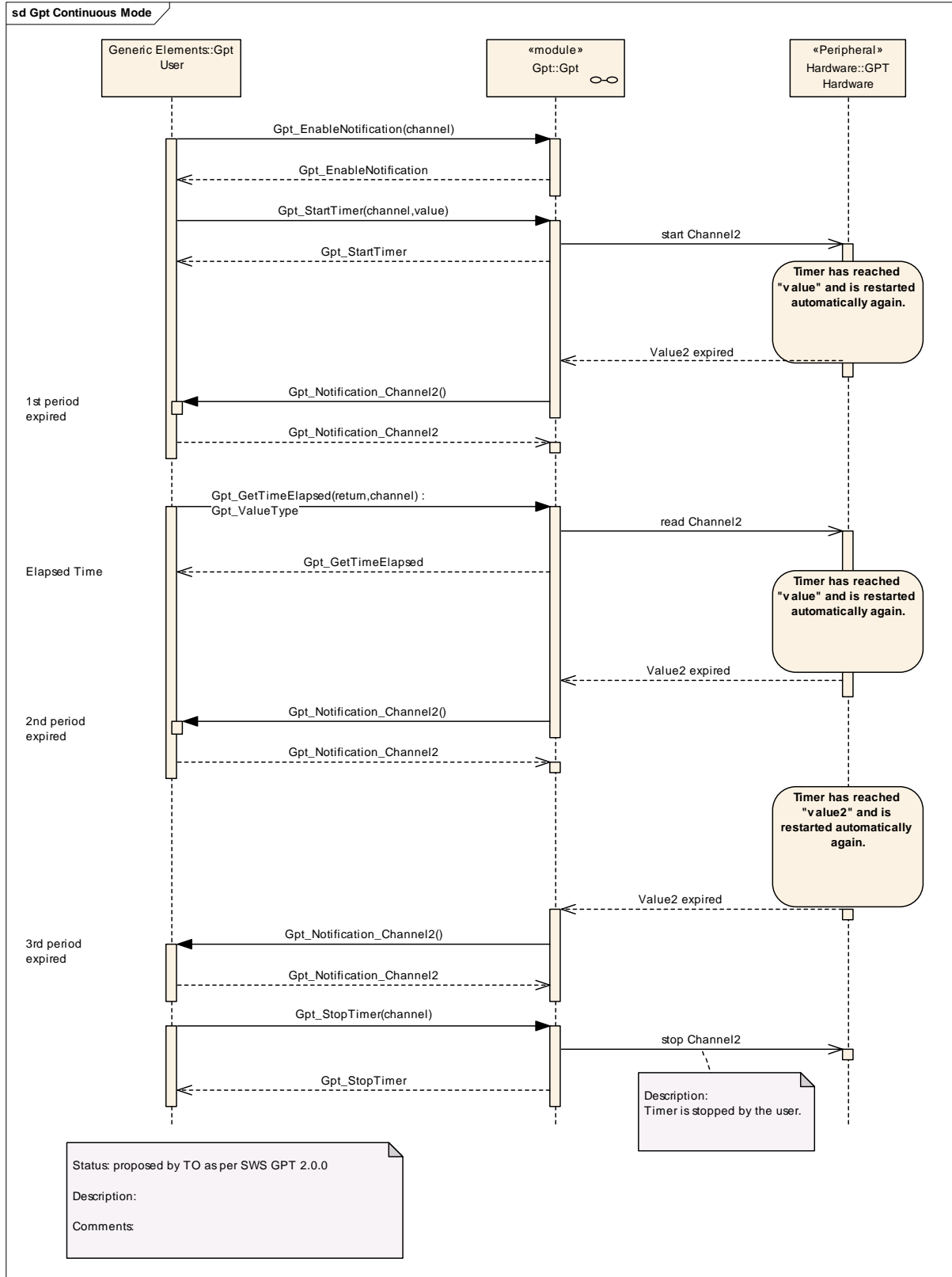
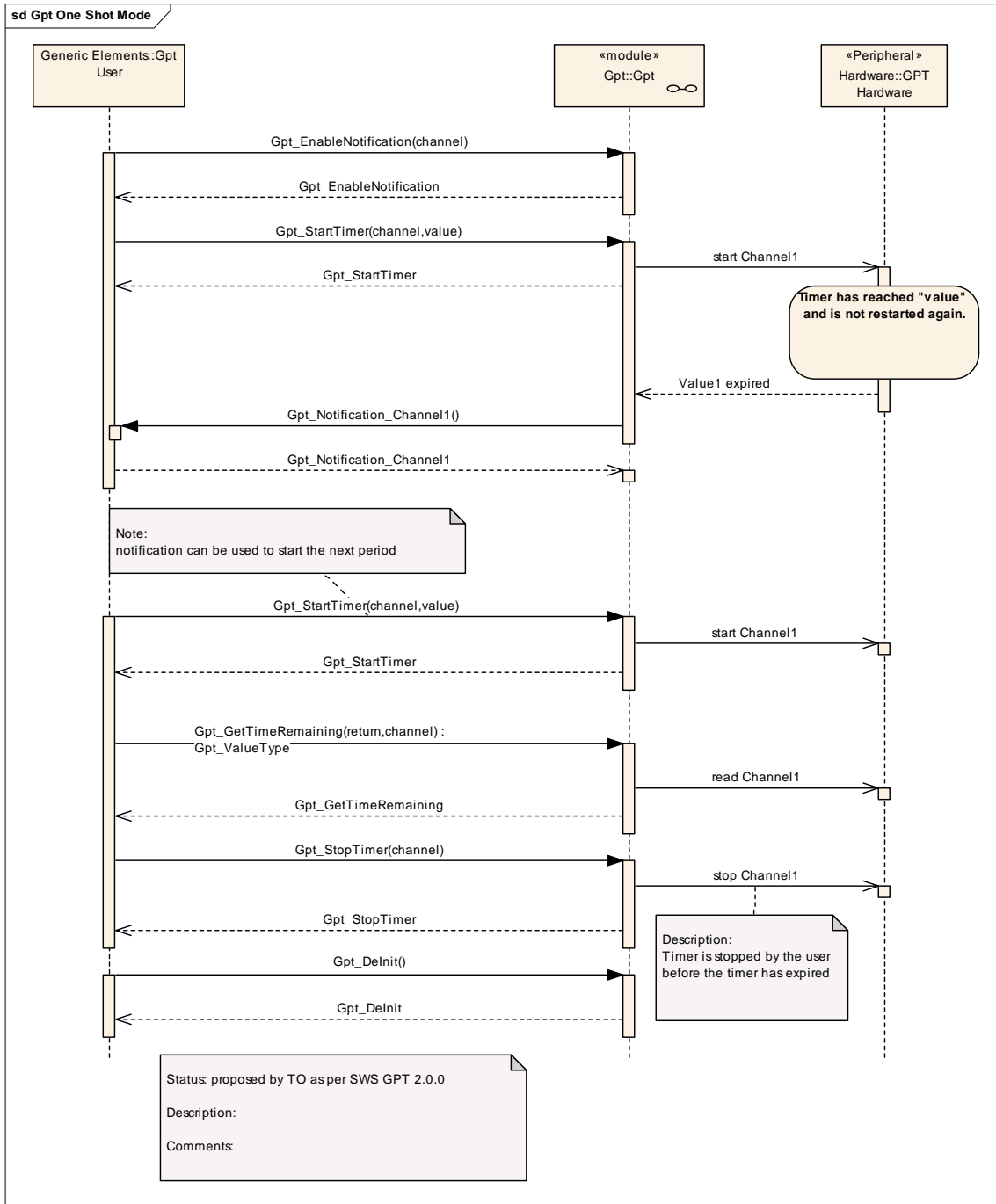


Figure 7: Sequence Diagram - GPT continuous mode



### 9.3 GPT one shot mode

Channel 1 is configured for “One shot Mode”

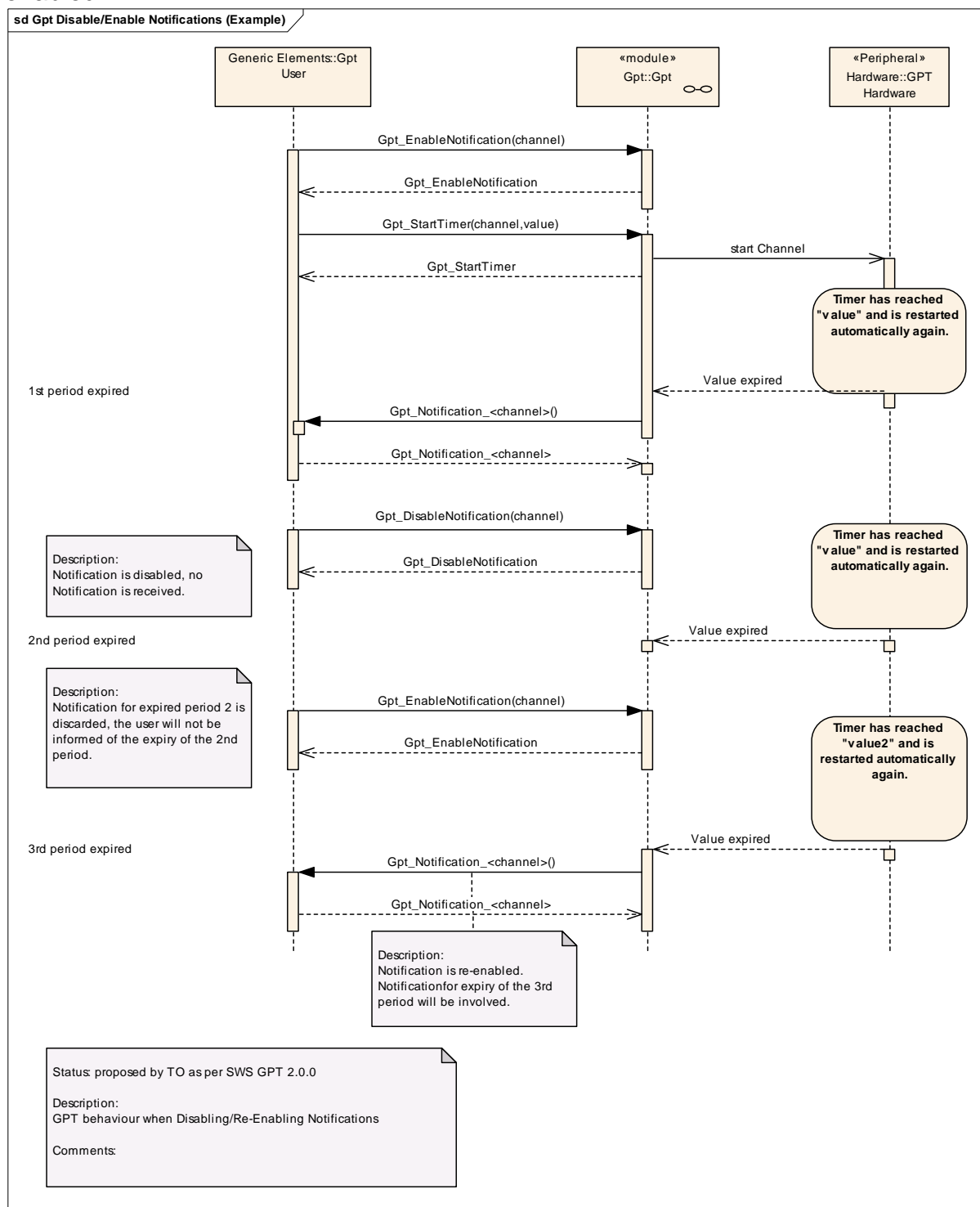


**Figure 8: Sequence Diagram - GPT one shot mode**

### 9.4 Disable/Re-enable Notifications

The sequence diagram shown in this chapter explains the behavior of the driver, when the timeout notification is disabled, while the timer is still running.

When disabled the user will not be informed, when timeout period 2 has expired. This notification is discarded and not made up again, when the timeout notification is re-enabled.



**Figure 9: Sequence Diagram - Disable/Re-enable Notifications**

## 9.5 Wakeup

Note: Sequence charts on timer wakeup can be found in the ECU state manager specification [7].

## 10 Configuration specification

In general, this chapter defines configuration parameters and their clustering into containers. In order to support the specification Chapter 10.1 describes fundamentals. It also specifies a template (table) you shall use for the parameter specification. We intend to leave Chapter 10.1 in the specification to guarantee comprehension.

Chapter 10.2 specifies the structure (containers) and the parameters of the module GPT

Chapter 10.3 specifies published information of the module GPT

### 10.1 How to read this chapter

In addition to this section, it is highly recommended to read the documents:

- AUTOSAR Layered Software Architecture [2].
- AUTOSAR ECU Configuration Specification [5]  
This document describes the AUTOSAR configuration methodology and the AUTOSAR configuration metamodel in detail.

The following is only a short survey of the topic and it will not replace the ECU Configuration Specification document.

#### 10.1.1 Configuration and configuration parameters

Configuration parameters define the variability of the generic part(s) of an implementation of a module. This means that only generic or configurable module implementation can be adapted to the environment (software/hardware) in use during system and/or ECU configuration.

The configuration of parameters can be achieved at different times during the software process: before compile time, before link time or after build time. In the following, the term “configuration class” (of a parameter) shall be used in order to refer to a specific configuration point in time.

#### 10.1.2 Variants

#### 10.1.3 Containers

Containers structure the set of configuration parameters. This means:

- *all* configuration parameters are kept in containers.
- (sub-) containers can reference (sub-) containers. It is possible to assign a multiplicity to these references. The multiplicity then defines the possible number of instances of the contained parameters.

#### 10.1.4 Specification template for configuration parameters

Pre-compile time - specifies whether the configuration parameter shall be of configuration class *Pre-compile time* or not

<b>Label</b>	<b>Description</b>
x	The configuration parameter shall be of configuration class <i>Pre-compile time</i> .
--	The configuration parameter shall never be of configuration class <i>Pre-compile time</i> .

Link time - specifies whether the configuration parameter shall be of configuration class *Link time* or not

<b>Label</b>	<b>Description</b>
x	The configuration parameter shall be of configuration class <i>Link time</i> .
--	The configuration parameter shall never be of configuration class <i>Link time</i> .

Post Build - specifies whether the configuration parameter shall be of configuration class *Post Build* or not

<b>Label</b>	<b>Description</b>
x	The configuration parameter shall be of configuration class <i>Post Build</i> and no specific implementation is required.
L	<i>Loadable</i> - the configuration parameter shall be of configuration class <i>Post Build</i> and only one configuration parameter set resides in the ECU.
M	<i>Multiple</i> - the configuration parameter shall be of configuration class <i>Post Build</i> and is selected out of a set of multiple parameters by passing a dedicated pointer to the init function of the module.
--	The configuration parameter shall never be of configuration class <i>Post Build</i> .

## 10.2 Containers and configuration parameters

The following chapters summarize all configuration parameters. The detailed meanings of the parameters describe Chapters 7 and Chapter 8.

### 10.2.1 Variants

**GPT276: Variant PC:** This variant is limited to pre-compile-configuration parameters only.

**GPT277: Variant PB:** This variant allows a mix of pre-compile time and post-build multiple selectable configurable parameters.

**GPT257:** The initialization function of this module shall always have a pointer as a parameter, even though for Variant PC no configuration set shall be given. Instead a NULL pointer shall be passed to the initialization function.

**GPT270:** Within one container it shall not be possible to mix parameters assigned to different configuration classes.

### 10.2.2 Gpt

<b>Module Name</b>	<b>Gpt</b>
<b>Module Description</b>	Configuration of the Gpt (General Purpose Timer) module.

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
GptChannelConfigSet	1..*	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.
GptConfigurationOfOptApiServices	1	This container contains all configuration switches for configuring optional API services of the GPT driver
GptDriverConfiguration	1	This container contains the module-wide configuration (parameters) of the GPT Driver

### 10.2.3 GptDriverConfiguration

<b>SWS Item</b>	<b>GPT183 :</b>
<b>Container Name</b>	GptDriverConfiguration
<b>Description</b>	This container contains the module-wide configuration (parameters) of the GPT Driver
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>GPT321 :</b>		
<b>Name</b>	GptDevErrorDetect {GPT_DEV_ERROR_DETECT}		
<b>Description</b>	Enables/Disables development error detection		
<b>Multiplicity</b>	1		
<b>Type</b>	BooleanParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: Module		

<b>SWS Item</b>	<b>GPT322 :</b>		
<b>Name</b>	GptReportWakeupSource {GPT_REPORT_WAKEUP_SOURCE}		
<b>Description</b>	Enables/Disables wakeup source reporting		
<b>Multiplicity</b>	1		
<b>Type</b>	BooleanParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: Module		

<b>No Included Containers</b>
-------------------------------

### 10.2.4 GptChannelConfiguration

<b>SWS Item</b>	<b>GPT184 :</b>		
<b>Container Name</b>	GptChannelConfiguration		
<b>Description</b>	This container contains the channel-wide configuration (parameters) of the GPT Driver		
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>GPT307 :</b>		
<b>Name</b>	GptChannelClkSrc {GPT_CHANNEL_CLKSRC}		
<b>Description</b>	GPT187: The GPT module specific clock input for the timer unit can statically be configured and allows to select different clock sources (external clock, internal GPT specific clock) per channel		
<b>Multiplicity</b>	0..1		
<b>Type</b>	IntegerParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	--	
	<b>Post-build time</b>	M	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: Instance		

<b>SWS Item</b>	<b>GPT308 :</b>		
<b>Name</b>	GptChannelId {GPT_CHANNEL_ID}		
<b>Description</b>	Channel Id of the GPT channel. This value will be assigned to the symbolic name derived of the GptChannelConfiguration container short name.		
<b>Multiplicity</b>	1		
<b>Type</b>	IntegerParamDef (Symbolic Name generated for this parameter)		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	--	
	<b>Post-build time</b>	M	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: Instance		

<b>SWS Item</b>	<b>GPT309 :</b>		
<b>Name</b>	GptChannelMode {GPT_CHANNEL_MODE}		
<b>Description</b>	Specifies the behaviour of the timerchannel after the timeout has expired		
<b>Multiplicity</b>	1		
<b>Type</b>	EnumerationParamDef		
<b>Range</b>	GPT_MODE_CONTINOUS	GPT186: Timerchannel is restarted automatically after reaching its end value	
	GPT_MODE_ONESHOT	GPT185: Timerchannel stops after reaching its end value	
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	--	
	<b>Post-build time</b>	M	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: Instance		

<b>SWS Item</b>	<b>GPT310 :</b>		
<b>Name</b>	GptChannelPrescale {GPT_CHANNEL_PRESCALE}		
<b>Description</b>	GPT module specific prescaler factor per channel		
<b>Multiplicity</b>	0..1		
<b>Type</b>	IntegerParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE



	<b>Link time</b>	--	
	<b>Post-build time</b>	M	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: Instance		

<b>SWS Item</b>	<b>GPT311 :</b>		
<b>Name</b>	GptEnableWakeup {GPT_ENABLE_WAKEUP}		
<b>Description</b>	GPT188: Enables wakeup capability of CPU for a channel when timeout period expires. This might be different to enabling the notification depending on hardware capabilities		
<b>Multiplicity</b>	1		
<b>Type</b>	BooleanParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	--	
	<b>Post-build time</b>	M	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: Instance		

<b>SWS Item</b>	<b>GPT312 :</b>		
<b>Name</b>	GptNotification {Gpt_Notification}		
<b>Description</b>	Function pointer to callback function		
<b>Multiplicity</b>	1		
<b>Type</b>	FunctionNameDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	--	
	<b>Post-build time</b>	M	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: Instance		

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
GptWakeupConfiguration	0..1	--

**GPT236:** It shall not be possible to add or remove GPT channels dynamically at runtime.

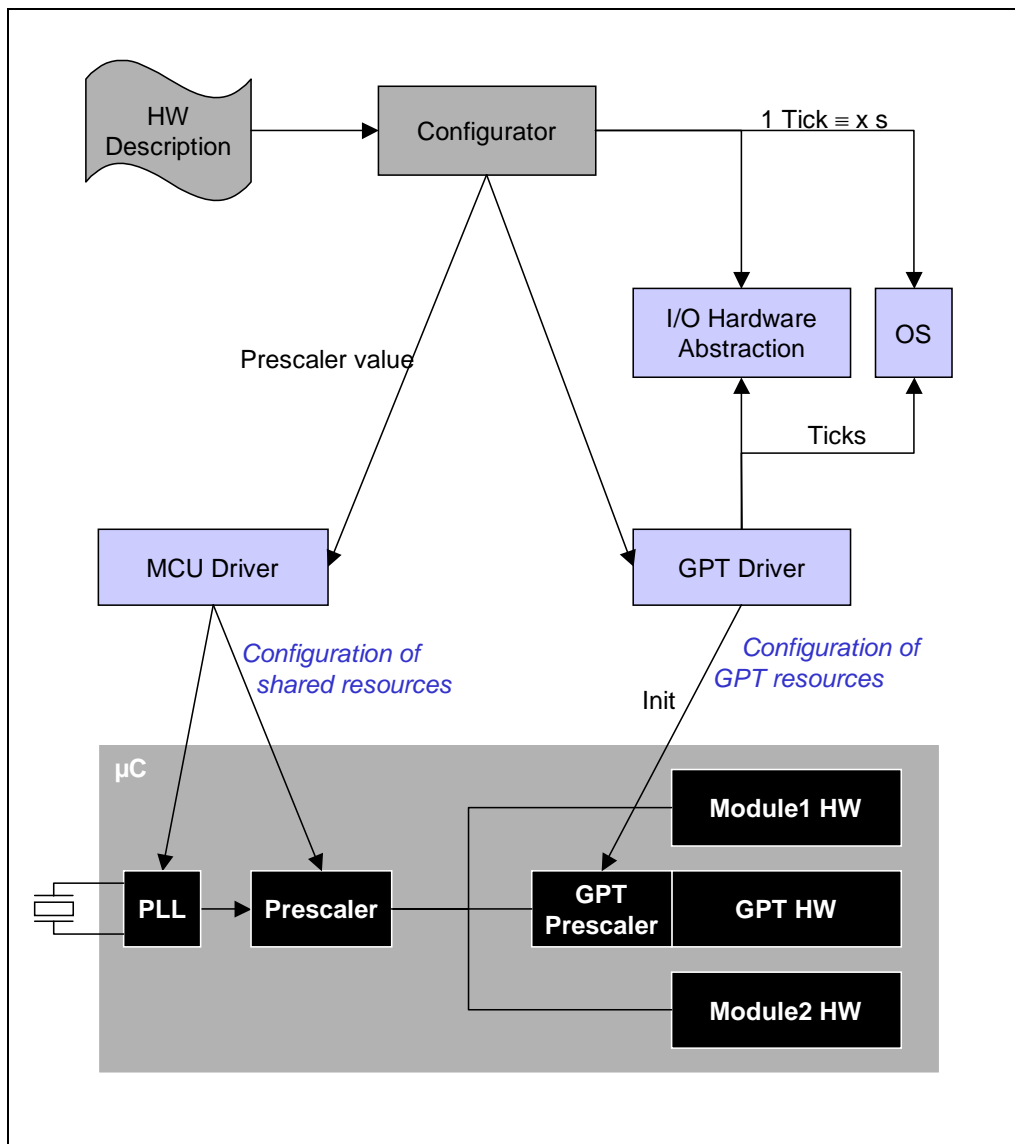


Figure 10: Scope of the GPT Driver configuration

### 10.2.5 GptChannelConfigSet

<b>SWS Item</b>	<b>GPT269 :</b>
<b>Container Name</b>	GptChannelConfigSet [Multi Config Container]
<b>Description</b>	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.
<b>Configuration Parameters</b>	

<b>Included Containers</b>		
<b>Container Name</b>	<b>Multiplicity</b>	<b>Scope / Dependency</b>
GptChannelConfiguration	1..*	This container contains the channel-wide configuration (parameters) of the GPT Driver

### 10.2.6 GptWakeupConfiguration

<b>SWS Item</b>	<b>GPT235 :</b>
<b>Container Name</b>	GptWakeupConfiguration{GPT_WAKEUP_CONFIGURATION}
<b>Description</b>	--
<b>Configuration Parameters</b>	

<b>SWS Item</b>	<b>GPT313 :</b>		
<b>Name</b>	GptWakeupSourceRef {Gpt_WakeupSourceRef}		
<b>Description</b>	In case the wakeup-capability is true this value is transmitted to the Ecu State Manager. Implementation Type: reference to EcuM_WakeupSourceType		
<b>Multiplicity</b>	1		
<b>Type</b>	Reference to EcuMWakeupSource		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	VARIANT-PRE-COMPILE
	<b>Link time</b>	--	
	<b>Post-build time</b>	M	VARIANT-POST-BUILD
<b>Scope / Dependency</b>	scope: instance		

**No Included Containers**

### 10.2.7 GptConfigurationOfOptApiServices

<b>SWS Item</b>	<b>GPT193 :</b>		
<b>Container Name</b>	GptConfigurationOfOptApiServices{Configuration of optional API services}		
<b>Description</b>	This container contains all configuration switches for configuring optional API services of the GPT driver		
<b>Configuration Parameters</b>			

<b>SWS Item</b>	<b>GPT314 :</b>		
<b>Name</b>	GptDeinitApi {GPT_DEINIT_API}		
<b>Description</b>	Adds / removes the service Gpt_DeInit() from the code.		
<b>Multiplicity</b>	1		
<b>Type</b>	BooleanParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: Module		

<b>SWS Item</b>	<b>GPT315 :</b>		
<b>Name</b>	GptEnableDisableNotificationApi {GPT_ENABLE_DISABLE_NOTIFICATION_API}		
<b>Description</b>	Adds / removes the services Gpt_EnableNotification() and Gpt_DisableNotification from the code.		
<b>Multiplicity</b>	1		
<b>Type</b>	BooleanParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: Module		

<b>SWS Item</b>	<b>GPT317 :</b>		
<b>Name</b>	GptTimeElapsedApi {GPT_TIME_ELAPSED_API}		
<b>Description</b>	Adds / removes the service Gpt_GetTimeElapsed() from the code		
<b>Multiplicity</b>	1		
<b>Type</b>	BooleanParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: Module		

<b>SWS Item</b>	<b>GPT318 :</b>		
<b>Name</b>	GptTimeRemainingApi {GPT_TIME_REMAINING_API}		
<b>Description</b>	Adds / removes the service Gpt_GetTimeRemaining() from the code.		
<b>Multiplicity</b>	1		
<b>Type</b>	BooleanParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: Module		

<b>SWS Item</b>	<b>GPT319 :</b>		
<b>Name</b>	GptVersionInfoApi {GPT_VERSION_INFO_API}		
<b>Description</b>	Adds / removes the service Gpt_GetVersionInfo() from the code.		
<b>Multiplicity</b>	1		
<b>Type</b>	BooleanParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: Module		

<b>SWS Item</b>	<b>GPT320 :</b>		
<b>Name</b>	GptWakeupFunctionalityApi {GPT_WAKEUP_FUNCTIONALITY_API}		
<b>Description</b>	Adds / removes the services Gpt_SetMode(), Gpt_EnableWakeup() Gpt_DisableWakeup() and Gpt_Cbk_CheckWakeup() from the code.		
<b>Multiplicity</b>	1		
<b>Type</b>	BooleanParamDef		
<b>Default value</b>	--		
<b>ConfigurationClass</b>	<b>Pre-compile time</b>	X	All Variants
	<b>Link time</b>	--	
	<b>Post-build time</b>	--	
<b>Scope / Dependency</b>	scope: Module		

**No Included Containers**

### 10.3 Published Information

Published information contains data defined by the implementer of the SW module that does not change when the module is adapted (i.e. configured) to the actual HW/SW environment. It thus contains version and manufacturer information.

The standard common published information like

vendorId (<Module>\_VENDOR\_ID),  
moduleId (<Module>\_MODULE\_ID),  
arMajorVersion (<Module>\_AR\_MAJOR\_VERSION),  
arMinorVersion (<Module>\_AR\_MINOR\_VERSION),  
arPatchVersion (<Module>\_AR\_PATCH\_VERSION),  
swMajorVersion (<Module>\_SW\_MAJOR\_VERSION),  
swMinorVersion (<Module>\_SW\_MINOR\_VERSION),  
swPatchVersion (<Module>\_SW\_PATCH\_VERSION),  
vendorApiInfix (<Module>\_VENDOR\_API\_INFIX)

is provided in the BSW Module Description Template (see [13] Figure 4.1 and Figure 7.1).

Additional published parameters are listed below if applicable for this module.

## 11 Changes to Release 1

### 11.1 Deleted SWS Items

<i>SWS Item</i>	<i>Rationale</i>
GPT027, GPT046, GPT063, GPT064	Not required for new configuration concept
GPT119, GPT120, GPT121	Redundant to <a href="#">GPT205</a>
GPT124	Requirement applicable for configuration tool
GPT169	Not required for new version checking concept
GPT190	Requirement applicable for configuration tool
GPT191	Requirement applicable for configuration tool

### 11.2 Replaced SWS Items

<i>SWS Item of Release 1</i>	<i>replaced SWS Item</i>	<i>by</i>	<i>Rationale</i>
GPT002	<a href="#">GPT178</a>		Redundancy
GPT003	<a href="#">GPT179</a>		Redundancy
GPT016	<a href="#">GPT232</a>		Moved to another chapter
GPT017	<a href="#">GPT233</a>		Moved to another chapter
GPT024	<a href="#">GPT187</a>		Moved to another chapter
GPT049	<a href="#">GPT204</a>		Moved to another chapter
GPT069	<a href="#">GPT206</a>		Moved to another chapter
GPT071	<a href="#">GPT185</a>		Moved to another chapter
GPT072	<a href="#">GPT186</a>		Moved to another chapter
GPT078	<a href="#">GPT207</a>		Moved to another chapter
GPT080	<a href="#">GPT208</a>		Moved to another chapter
GPT096	<a href="#">GPT234</a>		Moved to another chapter
GPT104	<a href="#">GPT188</a>		Moved to another chapter
GPT106	<a href="#">GPT209</a>		Moved to another chapter
GPT122	GPT190		Moved to another chapter
GPT123	GPT191		Moved to another chapter
GPT125	GPT119, GPT120, GPT121		Redundancy
GPT166	GPT192		Reformulation of SWS ID to achieve consistency among SPAL SWS.

### 11.3 Changed SWS Items

<i>SWS Item</i>	<i>Rationale</i>
<a href="#">GPT010</a>	Bug 5541
<a href="#">GPT084</a> , <a href="#">GPT295</a> , <a href="#">GPT296</a> , <a href="#">GPT301</a> , <a href="#">GPT302</a>	Improved description.
GPT085	Incorporated review comment
<a href="#">GPT105</a>	Bug 4555
GPT119	Bug 5463
GPT128, GPT129,	Bug #10088

GPT150, <a href="#">GPT235</a>	
<a href="#">GPT184</a>	Replaced types with “implementation specific” for parameters within GptChannelConfiguration
<a href="#">GPT148</a>	Bug 4549
GPT149	Bug 4550

## 11.4 Added SWS Items

<b>SWS Item</b>	<b>Rationale</b>
<a href="#">GPT127</a> , <a href="#">GPT128</a> , <a href="#">GPT129</a> , <a href="#">GPT132</a> , <a href="#">GPT150</a>	Wakeup feature first introduced in Release 2.0
<a href="#">GPT171</a> , <a href="#">GPT172</a> , <a href="#">GPT173</a> , <a href="#">GPT174</a> , <a href="#">GPT175</a> , <a href="#">GPT176</a> , <a href="#">GPT177</a> , <a href="#">GPT178</a> , <a href="#">GPT179</a> , <a href="#">GPT181</a> , <a href="#">GPT182</a> ,	Required for new SWS template.
<a href="#">GPT183</a>	Required field for specifying Container “GptDriverConfiguration”
<a href="#">GPT184</a>	Required field for specifying Container “GptChannelConfiguration”
<a href="#">GPT189</a>	Required field for specifying “Published Information”
<a href="#">GPT193</a>	Required field for specifying “Configuration of optional API services”
<a href="#">GPT194</a>	Configuration of optional API service “Gpt_DeInit()”
<a href="#">GPT195</a>	Configuration of optional API service “Gpt_GetTimeElapsed() “
<a href="#">GPT196</a>	Configuration of optional API service “Gpt_GetTimeRemaining() “
<a href="#">GPT197</a>	Configuration of optional API service “Gpt_StartTimer() “
<a href="#">GPT198</a>	Configuration of optional API service “Gpt_StopTimer() “
<a href="#">GPT199</a>	Configuration of optional API service “Gpt_EnableNotification() “
<a href="#">GPT200</a>	Configuration of optional API service “Gpt_DisableNotification() “
<a href="#">GPT201</a>	Configuration of optional API service “Gpt_SetMode() “
<a href="#">GPT202</a>	Configuration of optional API service “Gpt_DisableWakeup() “
<a href="#">GPT203</a>	Configuration of optional API service “Gpt_EnableWakeup() “
<a href="#">GPT205</a>	Rules concerning register initialization defined
<a href="#">GPT210</a> , <a href="#">GPT222</a>	API parameter checking: “Gpt_GetTimeElapsed”
<a href="#">GPT211</a> , <a href="#">GPT223</a>	API parameter checking: “Gpt_GetTimeRemaining”
<a href="#">GPT212</a> , <a href="#">GPT218</a> , <a href="#">GPT224</a>	API parameter checking: “Gpt_StartTimer”
<a href="#">GPT213</a> , <a href="#">GPT225</a>	API parameter checking: “Gpt_StopTimer”
<a href="#">GPT214</a> , <a href="#">GPT226</a>	API parameter checking: “Gpt_EnableNotification”
<a href="#">GPT215</a> , <a href="#">GPT230</a>	API parameter checking: “Gpt_EnableWakeup”
<a href="#">GPT216</a> , <a href="#">GPT229</a>	API parameter checking: “Gpt_DisableWakeup”
<a href="#">GPT217</a> , <a href="#">GPT227</a>	API parameter checking: “Gpt_DisableNotification”
<a href="#">GPT220</a>	API parameter checking: “Gpt_DeInit”
<a href="#">GPT221</a>	API parameter checking: “Gpt_GetVersionInfo”
<a href="#">GPT228</a> , <a href="#">GPT231</a>	API parameter checking: “Gpt_SetMode”
<a href="#">GPT235</a>	Configuration of wakeup
<a href="#">GPT236</a>	Bug 4940
<a href="#">GPT240</a> , <a href="#">GPT241</a> , <a href="#">GPT242</a> , <a href="#">GPT243</a> ,	Dependency to other modules
<a href="#">GPT245</a> , <a href="#">GPT246</a>	Mandatory Interfaces: DEM
<a href="#">GPT255</a>	Review comment #71
<a href="#">GPT256</a>	Review comment #75
<a href="#">GPT257</a>	added (SPAL decision, 42 <sup>nd</sup> meeting, minutes day2, issue 5)
Release 2.1	
<a href="#">GPT258</a>	Bugzilla #11592

## 12 Changes to Release 2.0

### 12.1 Deleted SWS Items

<i>SWS Item</i>	<i>Rationale</i>
GPT149	Covered by reformulated GPT097
GPT170	Covered by reformulated GPT101
GPT059	Bug <b>13096</b>
GPT219	Bug <b>13902</b>
GPT246	Bug <b>15795</b>
GPT243	Bug <b>14502</b>
GPT244	Bug <b>12379</b>
GPT263	Bug <b>17614</b>
GPT197	Bug <b>17617</b>
GPT198	Bug <b>17617</b>

### 12.2 Replaced SWS Items

<i>SWS Item of Release 1</i>	<i>replaced SWS Item</i>	<i>by</i>	<i>Rationale</i>
--	--	--	--

### 12.3 Changed SWS Items

<i>SWS Item</i>	<i>Rationale</i>
<a href="#">GPT216</a>	<b>Rfc #11978:</b> Chapter 8.3.12, GPT216, Formal Error Description: SWS requirement GPT216 is for Gpt_EnableWakeUp(). In this requirement Function name is mentioned as Gpt_EnableNotification instead of Gpt_EnableWakeUp.
<a href="#">GPT221</a>	<b>Rfc #12318:</b> Chapter 8.3.1, GPT221 Description: 1) There is a wrong mention on Gpt_Deinit.
<a href="#">GPT218</a>	<b>Rfc #12321:</b> Chapter 8.3.6, GPT218 Description: It should be "GPT_E_PARAM_VALUE" and not "GPT_E_PARAM_CHANNEL"
<a href="#">GPT210</a>	<b>Rfc #12327:</b> Chapter 8.3.4 and 8.3.5 Description: Return value not mentioned when development error detection is enabled and Gpt_GetTimeElapsed\Gpt_GetTimeRemaining is called prior to initializing the driver.
<a href="#">GPT211</a>	
<a href="#">GPT222</a>	
<a href="#">GPT223</a>	
GPT097	<b>Rfc #14814</b>
GPT100	
GPT101	
GPT102	
<a href="#">GPT184</a>	<b>Bug 12661</b>
<a href="#">GPT153</a>	<b>Rfc #13708</b>
<a href="#">GPT159</a>	
<a href="#">GPT160</a>	
<a href="#">GPT258</a>	
<a href="#">GPT188</a>	<b>Rfc #13743</b>



<a href="#">GPT183</a>	<b>Rfc #15740:</b> container structure changed, "GptDriverConfiguration" includes no subcontainer anymore
<a href="#">GPT184</a>	<b>Rfc #15740:</b> Changed multiplicity of "wakeup configuration" from (0..*) → (0..1)
<a href="#">GPT175</a>	<b>Bug 17616</b>

## 12.4 Added SWS Items

<b>SWS Item</b>	<b>Rationale</b>
<a href="#">GPT259</a>	<b>Rfc #12317:</b> GPT: File Structure not correct Gpt_xCfg.c doesn't include Gpt.h, then how can it define the configuration structure(Gpt_ConfigType) declared in Gpt.h?
<a href="#">GPT260</a>	
<a href="#">GPT261</a>	
<a href="#">GPT262</a>	
<a href="#">GPT263</a>	
<a href="#">GPT264</a>	
<a href="#">GPT265</a>	
<a href="#">GPT266</a>	<b>Rfc #14814</b>
<a href="#">GPT267</a>	
<a href="#">GPT268</a>	<b>Rfc #15740:</b> introduced new container "GptChannelConfigSet"
<a href="#">GPT269</a>	
<a href="#">GPT270</a>	<b>Rfc #15470:</b> SPAL decision (Meeting 51): it shall not be allowed to mix parameters assigned to different configuration classes within one container
<a href="#">GPT271</a>	<b>Rfc #12487</b> File Include Structure

## 13 Changes to Release 2.1

### 13.1 Deleted SWS Items

<b>SWS Item</b>	<b>Rationale</b>
GPT128	Bug10356: this item was describing the obsolete wakeup concept as depicted in deleted sequence charts (Figure 7 and Figure 8). Since this concept has been rated invalid, the corresponding SWS Items can be removed as well.
GPT129	Bug10356: this item was describing the obsolete wakeup concept as depicted in deleted sequence charts (Figure 7 and Figure 8). Since this concept has been rated invalid, the corresponding SWS Items can be removed as well.
GPT150	Bug10356: this item was describing the obsolete wakeup concept as depicted in deleted sequence charts (Figure 7 and Figure 8). Since this concept has been rated invalid, the corresponding SWS Items can be removed as well.
GPT088	[RfC#19058] Initiation of SWS Improvement, Issues 14, 15
GPT300	[RfC#19058] Initiation of SWS Improvement, Issue 22
GPT306	[RfC#19058] Initiation of SWS Improvement, Issue 24
GPT085	[RfC#19058] Initiation of SWS Improvement, Issue 27
GPT091	[RfC#19058] Initiation of SWS Improvement, Issue 35
GPT092	[RfC#19058] Initiation of SWS Improvement, Issue 39
GPT264	GPT provides no Callback function
GPT265	GPT provides no Callback function
GPT266	GPT provides no Callback function
GPT221	BSW00406 changed: Gpt_GetVersionInfo can be called without module initialization
GPT132	Removed, since wakeup sequences shall only be depicted in the ECU state manager

### 13.2 Replaced SWS Items

<b>SWS Item of Release 2.1</b>	<b>replaced by SWS Item in R3.0</b>	<b>Rationale</b>
GPT148	<a href="#">GPT307</a>	[RfC#17500] ADC269: Criteria of detection of ADC_E_UNINIT

### 13.3 Changed SWS Items

<b>SWS Item</b>	<b>Rationale</b>
<a href="#">GPT172</a>	Bug17223: sentence has been agreed in SPAL meeting 2006-12-12
<a href="#">GPT257</a>	Bug 18755
<a href="#">GPT204</a>	[RfC#17500] ADC269: Criteria of detection of ADC_E_UNINIT Added description of GPT_E_ALREADY_INITIALIZED
<a href="#">GPT001</a>	[RfC#17500] ADC269: Criteria of detection of ADC_E_UNINIT Added error code of GPT_E_ALREADY_INITIALIZED
<a href="#">GPT171</a>	[RfC#19058] Initiation of SWS Improvement, Issue 3
<a href="#">GPT261</a>	[RfC#19058] Initiation of SWS Improvement, Issue 5
<a href="#">GPT262</a>	[RfC#19058] Initiation of SWS Improvement, Issue 6
<a href="#">GPT272</a>	[RfC#19058] Initiation of SWS Improvement, Issue 10

<a href="#">GPT107</a>	[RfC#19058] Initiation of SWS Improvement, Issue 11
<a href="#">GPT258</a>	[RfC#19058] Initiation of SWS Improvement, Issue 12
<a href="#">GPT294</a>	[RfC#19058] Initiation of SWS Improvement, Issue 13
<a href="#">GPT008</a>	[RfC#19058] Initiation of SWS Improvement, Issues 14, 15, 16
<a href="#">GPT161</a>	[RfC#19058] Initiation of SWS Improvement, Issues 14, 15, 16
<a href="#">GPT105</a>	[RfC#19058] Initiation of SWS Improvement, Issue 17
<a href="#">GPT162</a>	[RfC#19058] Initiation of SWS Improvement, Issue 18
<a href="#">GPT234</a>	[RfC#19058] Initiation of SWS Improvement, Issue 19
<a href="#">GPT297</a>	[RfC#19058] Initiation of SWS Improvement, Issue 20
<a href="#">GPT103</a>	[RfC#19058] Initiation of SWS Improvement, Issue 30
<a href="#">GPT211</a>	[RfC#19058] Initiation of SWS Improvement, Issue 23
<a href="#">GPT275</a>	[RfC#19058] Initiation of SWS Improvement, Issue 26
<a href="#">GPT212</a>	[RfC#19058] Initiation of SWS Improvement, Issue 28
<a href="#">GPT218</a>	[RfC#19058] Initiation of SWS Improvement, Issue 29
<a href="#">GPT103</a>	[RfC#19058] Initiation of SWS Improvement, Issue 31
<a href="#">GPT213</a>	[RfC#19058] Initiation of SWS Improvement, Issue 32
<a href="#">GPT014</a>	[RfC#19058] Initiation of SWS Improvement, Issues 33, 34
<a href="#">GPT214</a>	[RfC#19058] Initiation of SWS Improvement, Issue 36
<a href="#">GPT015</a>	[RfC#19058] Initiation of SWS Improvement, Issues 37, 38
<a href="#">GPT217</a>	[RfC#19058] Initiation of SWS Improvement, Issue 40
<a href="#">GPT159</a>	[RfC#19058] Initiation of SWS Improvement, Issue 42
<a href="#">GPT215</a>	[RfC#19058] Initiation of SWS Improvement, Issue 45
<a href="#">GPT160</a>	[RfC#19058] Initiation of SWS Improvement, Issue 46
<a href="#">GPT216</a>	[RfC#19058] Initiation of SWS Improvement, Issue 49
<a href="#">GPT233</a>	[RfC#19058] Initiation of SWS Improvement, Issue 50
<a href="#">GPT271</a>	[Bug 19685] Recursive inclusion with EcuM.h

### 13.4 Added SWS Items

<i>SWS Item</i>	<i>Rationale</i>
<a href="#">GPT272</a>	Bug 18755: sentence agreed within SPAL, 2006-12-12
<a href="#">GPT308</a>	[RfC#19058] Initiation of SWS Improvement, Issue 18
<a href="#">GPT309</a>	[RfC#19605] Re-initialization of MCAL modules (all drivers)

## 14 Changes during SWS Improvements by Technical Office

### 14.1 Deleted SWS Items

<i>SWS Item</i>	<i>Rationale</i>
GPT192	No requirement on GPT driver
GPT167	No requirement
GPT168	No requirement
GPT060	No requirement
GPT260	No requirement, just an explanation (also see new req GPT293)

### 14.2 Replaced SWS Items

<i>SWS Item of Release 1</i>	<i>replaced SWS Item</i>	<i>by</i>	<i>Rationale</i>
GPT012	<a href="#">GPT274</a> , <a href="#">GPT275</a>		Made requirement atomic
GPT097	<a href="#">GPT295</a> , <a href="#">GPT296</a>		Made requirement atomic
GPT267	<a href="#">GPT297</a> , <a href="#">GPT298</a>		Made requirement atomic
GPT100	<a href="#">GPT299</a> , <a href="#">GPT300</a>		Made requirement atomic
GPT101	<a href="#">GPT301</a> , <a href="#">GPT302</a>		Made requirement atomic
GPT268	<a href="#">GPT303</a> , <a href="#">GPT304</a>		Made requirement atomic
GPT102	<a href="#">GPT305</a> , <a href="#">GPT306</a>		Made requirement atomic

### 14.3 Changed SWS Items

Many requirements have been changed to improve understandability without changing the technical contents.

### 14.4 Added SWS Items

<i>SWS Item</i>	<i>Rationale</i>
<a href="#">GPT273</a>	Hint Gpt_GetVersionInfo
<a href="#">GPT276</a>	Definition of configuration variant needs an id
<a href="#">GPT277</a>	Definition of configuration variant needs an id
<a href="#">GPT278</a>	UML Model linking of imported types
<a href="#">GPT279</a>	UML Model linking of Gpt_GetVersionInfo
<a href="#">GPT280</a>	UML Model linking of Gpt_Init
<a href="#">GPT281</a>	UML Model linking of Gpt_DeInit
<a href="#">GPT282</a>	UML Model linking of Gpt_GetTimeElapsed
<a href="#">GPT283</a>	UML Model linking of Gpt_GetTimeRemaining
<a href="#">GPT284</a>	UML Model linking of Gpt_StartTimer
<a href="#">GPT285</a>	UML Model linking of Gpt_StopTimer
<a href="#">GPT286</a>	UML Model linking of Gpt_EnableNotification
<a href="#">GPT287</a>	UML Model linking of Gpt_DisableNotification
<a href="#">GPT288</a>	UML Model linking of Gpt_SetMode
<a href="#">GPT289</a>	UML Model linking of Gpt_DisableWakeup
<a href="#">GPT290</a>	UML Model linking of Gpt_EnableWakeup

<a href="#">GPT291</a>	UML Model linking of optional interfaces
<a href="#">GPT292</a>	UML Model linking of GptNotification_<channel>
<a href="#">GPT293</a>	New requirement according to the figure 1
<a href="#">GPT294</a>	Added explicit development error for Gpt_Init
<a href="#">GPT321</a>	Introd. new service Gpt_Cbk_CheckWakeup to harmonize wakeup concept
<a href="#">GPT322</a>	Introd. new service Gpt_Cbk_CheckWakeup to harmonize wakeup concept
<a href="#">GPT323</a>	Introd. new service Gpt_Cbk_CheckWakeup to harmonize wakeup concept
<a href="#">GPT324</a>	Introd. new service Gpt_Cbk_CheckWakeup to harmonize wakeup concept
<a href="#">GPT325</a>	Introd. new service Gpt_Cbk_CheckWakeup to harmonize wakeup concept
<a href="#">GPT326</a>	Introd. new service Gpt_Cbk_CheckWakeup to harmonize wakeup concept
<a href="#">GPT327</a>	Introd. new service Gpt_Cbk_CheckWakeup to harmonize wakeup concept
<a href="#">GPT328</a>	Introd. new service Gpt_Cbk_CheckWakeup to harmonize wakeup concept