

AC784xx_DFP TIMER

4.1.0

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	2
2.1	File List	2
3	Class Documentation	3
3.1	Timer_Channel_ConfigType Struct Reference	3
3.1.1	Detailed Description	3
3.1.2	Member Data Documentation	3
3.1.2.1	Config	3
3.1.2.2	Mode	4
3.1.2.3	TriggerSrc	4
4	File Documentation	5
4.1	AC784xx_API_Reference_Manual_TIMER.pdf File Reference	5
4.2	AC784xx_Timer_Reg.h File Reference	5
4.2.1	Detailed Description	6
4.2.2	Function Documentation	6
4.2.2.1	Timer_Reg_GetChBase()	6
4.2.2.2	Timer_Reg_ReadCR()	6
4.2.2.3	Timer_Reg_ReadCTRL()	7
4.2.2.4	Timer_Reg_ReadCVAL()	7
4.2.2.5	Timer_Reg_ReadENR()	7
4.2.2.6	Timer_Reg_ReadIER()	8
4.2.2.7	Timer_Reg_ReadSR()	8

4.2.2.8	Timer_Reg_ReadTVAL()	8
4.2.2.9	Timer_Reg_WriteCR()	9
4.2.2.10	Timer_Reg_WriteCTRL()	9
4.2.2.11	Timer_Reg_WriteCVAL()	10
4.2.2.12	Timer_Reg_WriteENR()	10
4.2.2.13	Timer_Reg_WriteIER()	11
4.2.2.14	Timer_Reg_WriteSR()	11
4.2.2.15	Timer_Reg_WriteTVAL()	12
4.3	Timer_Hal.c File Reference	12
4.3.1	Detailed Description	13
4.3.2	Macro Definition Documentation	13
4.3.2.1	TIMER_CHANNEL_CTRL_MODE_0	13
4.3.2.2	TIMER_CHANNEL_CTRL_MODE_1	14
4.3.2.3	TIMER_CHANNEL_CTRL_MODE_2	14
4.3.2.4	TIMER_CHANNEL_CTRL_MODE_3	14
4.3.2.5	TIMER_STATUS_CH0_STARTED	14
4.3.2.6	TIMER_STATUS_INITED	14
4.3.3	Function Documentation	14
4.3.3.1	TIMER_Channel0_IRQHandler()	15
4.3.3.2	TIMER_Channel1_IRQHandler()	15
4.3.3.3	TIMER_Channel2_IRQHandler()	15
4.3.3.4	TIMER_Channel3_IRQHandler()	15
4.3.3.5	Timer_Hal_DeInit()	15
4.3.3.6	Timer_Hal_EnableInterrupt()	16
4.3.3.7	Timer_Hal_GetConfig()	16
4.3.3.8	Timer_Hal_GetCurrentValue()	17
4.3.3.9	Timer_Hal_GetRemainingValue()	17
4.3.3.10	Timer_Hal_Init()	17
4.3.3.11	Timer_Hal_InstallCallback()	18
4.3.3.12	Timer_Hal_MicrosToTicks()	19
4.3.3.13	Timer_Hal_ResetValue()	19

4.3.3.14	Timer_Hal_SetConfig()	20
4.3.3.15	Timer_Hal_Start()	20
4.3.3.16	Timer_Hal_Stop()	21
4.4	Timer_Hal.h File Reference	21
4.4.1	Detailed Description	22
4.4.2	Macro Definition Documentation	23
4.4.2.1	STATUS_TIMER_WRONG_PARAM	23
4.4.2.2	STATUS_TIMER_WRONG_STATE	23
4.4.2.3	TIMER_CHN_EN	23
4.4.2.4	TIMER_DBG_EN	23
4.4.2.5	TIMER_INT_EN	23
4.4.2.6	TIMER_IRQ_EN	24
4.4.2.7	TIMER_ONESHOT_EN	24
4.4.2.8	TIMER_TROT	24
4.4.2.9	TIMER_TSOT	24
4.4.3	Enumeration Type Documentation	24
4.4.3.1	Timer_ClockSourceType	24
4.4.3.2	Timer_ModeType	25
4.4.4	Function Documentation	25
4.4.4.1	Timer_Hal_DeInit()	25
4.4.4.2	Timer_Hal_EnableInterrupt()	26
4.4.4.3	Timer_Hal_GetConfig()	26
4.4.4.4	Timer_Hal_GetCurrentValue()	27
4.4.4.5	Timer_Hal_GetRemainingValue()	27
4.4.4.6	Timer_Hal_Init()	27
4.4.4.7	Timer_Hal_InstallCallback()	28
4.4.4.8	Timer_Hal_MicrosToTicks()	29
4.4.4.9	Timer_Hal_ResetValue()	29
4.4.4.10	Timer_Hal_SetConfig()	30
4.4.4.11	Timer_Hal_Start()	30
4.4.4.12	Timer_Hal_Stop()	31

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Timer_Channel_ConfigType	
Timer channel configure type	3

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_TIMER.pdf	5
AC784xx_Timer_Reg.h	
Timer access register inline function definition	5
Timer_Hal.c	
Timer HAL source file	12
Timer_Hal.h	
Timer HAL header file	21

Chapter 3

Class Documentation

3.1 Timer_Channel_ConfigType Struct Reference

Timer channel configure type.

```
#include <Timer_Hal.h>
```

Public Attributes

- [Timer_ModeType](#) Mode
- uint32 [Config](#)
- uint32 [TriggerSrc](#)

3.1.1 Detailed Description

Timer channel configure type.

Definition at line 106 of file Timer_Hal.h.

3.1.2 Member Data Documentation

3.1.2.1 Config

```
uint32 Timer_Channel_ConfigType::Config
```

Timer configuraiton. It is combinations(bitwise OR) of TIMER_ONESHOT_EN/TIMER_DBG_EN/TIMER_CHN_EN/TIMER_TROT/TIMER_TSOT.

Definition at line 109 of file Timer_Hal.h.

3.1.2.2 Mode

`Timer_ModeType` `Timer_Channel_ConfigType::Mode`

Work mode of timer.

Definition at line 108 of file `Timer_Hal.h`.

3.1.2.3 TriggerSrc

`uint32` `Timer_Channel_ConfigType::TriggerSrc`

Trigger source of timer.

Definition at line 111 of file `Timer_Hal.h`.

The documentation for this struct was generated from the following file:

- [Timer_Hal.h](#)

Chapter 4

File Documentation

4.1 AC784xx_API_Reference_Manual_TIMER.pdf File Reference

4.2 AC784xx_Timer_Reg.h File Reference

Timer access register inline function definition.

```
#include "Device_Register.h"
```

Functions

- LOCAL_INLINE uint32 [Timer_Reg_ReadCR](#) (void)
Read timer module control register.
- LOCAL_INLINE void [Timer_Reg_WriteCR](#) (uint32 Value)
Set timer module control register.
- LOCAL_INLINE uint32 [Timer_Reg_ReadSR](#) (void)
Read interrupt status register.
- LOCAL_INLINE void [Timer_Reg_WriteSR](#) (uint32 Value)
Set interrupt status register.
- LOCAL_INLINE uint32 [Timer_Reg_ReadIER](#) (void)
Read IER register.
- LOCAL_INLINE void [Timer_Reg_WriteIER](#) (uint32 Value)
Set IER register.
- LOCAL_INLINE uint32 [Timer_Reg_ReadENR](#) (void)
Read ENR register.
- LOCAL_INLINE void [Timer_Reg_WriteENR](#) (uint32 Value)
Set ENR register.
- LOCAL_INLINE uint32 [Timer_Reg_ReadTVAL](#) (const TIMER_CHANNEL_Type *Base)
Read TVAL register.
- LOCAL_INLINE void [Timer_Reg_WriteTVAL](#) (TIMER_CHANNEL_Type *Base, uint32 Value)
Set TVAL register.
- LOCAL_INLINE uint32 [Timer_Reg_ReadCVAL](#) (const TIMER_CHANNEL_Type *Base)
Read CVAL register.
- LOCAL_INLINE void [Timer_Reg_WriteCVAL](#) (TIMER_CHANNEL_Type *Base, uint32 Value)
Set CVAL register.
- LOCAL_INLINE uint32 [Timer_Reg_ReadCTRL](#) (const TIMER_CHANNEL_Type *Base)
Read CTRL register.
- LOCAL_INLINE void [Timer_Reg_WriteCTRL](#) (TIMER_CHANNEL_Type *Base, uint32 Value)
Set CTRL register.
- LOCAL_INLINE TIMER_CHANNEL_Type * [Timer_Reg_GetChBase](#) (uint32 Channel)
Get base address of a timer channel.

4.2.1 Detailed Description

Timer access register inline function definition.

4.2.2 Function Documentation

4.2.2.1 Timer_Reg_GetChBase()

```
LOCAL_INLINE TIMER_CHANNEL_Type* Timer_Reg_GetChBase (
    uint32 Channel )
```

Get base address of a timer channel.

Note

Function ID : DES_GPT_API_563

Parameters

in	<i>Channel</i>	Channel index.
----	----------------	----------------

Returns

Pointer to structure of TIMER_CHANNEL_Type.

Definition at line 220 of file AC784xx_Timer_Reg.h.

4.2.2.2 Timer_Reg_ReadCR()

```
LOCAL_INLINE uint32 Timer_Reg_ReadCR (
    void )
```

Read timer module control register.

Note

Function ID : DES_GPT_API_550

Returns

value of control register.

Definition at line 64 of file AC784xx_Timer_Reg.h.

4.2.2.3 Timer_Reg_ReadCTRL()

```
LOCAL_INLINE uint32 Timer_Reg_ReadCTRL (
    const TIMER_CHANNEL_Type * Base )
```

Read CTRL register.

Note

Function ID : DES_GPT_API_561

Parameters

in	<i>Base</i>	Base address to TIMER_CHANNEL_Type.
----	-------------	-------------------------------------

Returns

value of CTRL register

Definition at line 197 of file AC784xx_Timer_Reg.h.

4.2.2.4 Timer_Reg_ReadCVAL()

```
LOCAL_INLINE uint32 Timer_Reg_ReadCVAL (
    const TIMER_CHANNEL_Type * Base )
```

Read CVAL register.

Note

Function ID : DES_GPT_API_559

Parameters

in	<i>Base</i>	Base address to TIMER_CHANNEL_Type.
----	-------------	-------------------------------------

Returns

Value of CVAL register

Definition at line 173 of file AC784xx_Timer_Reg.h.

4.2.2.5 Timer_Reg_ReadENR()

```
LOCAL_INLINE uint32 Timer_Reg_ReadENR (
    void )
```

Read ENR register.

Note

Function ID : DES_GPT_API_556

Returns

value of ENR register

Definition at line 128 of file AC784xx_Timer_Reg.h.

4.2.2.6 Timer_Reg_ReadIER()

```
LOCAL_INLINE uint32 Timer_Reg_ReadIER (
    void )
```

Read IER register.

Note

Function ID : DES_GPT_API_554

Returns

value of IER register

Definition at line 107 of file AC784xx_Timer_Reg.h.

4.2.2.7 Timer_Reg_ReadSR()

```
LOCAL_INLINE uint32 Timer_Reg_ReadSR (
    void )
```

Read interrupt status register.

Note

Function ID : DES_GPT_API_552

Returns

value of interrupt status register

Definition at line 85 of file AC784xx_Timer_Reg.h.

4.2.2.8 Timer_Reg_ReadTVAL()

```
LOCAL_INLINE uint32 Timer_Reg_ReadTVAL (
    const TIMER_CHANNEL_Type * Base )
```

Read TVAL register.

Note

Function ID:

Parameters

in	<i>Base</i>	Base address to TIMER_CHANNEL_Type.
----	-------------	-------------------------------------

Returns

value of TVAL register

Definition at line 150 of file AC784xx_Timer_Reg.h.

4.2.2.9 Timer_Reg_WriteCR()

```
LOCAL_INLINE void Timer_Reg_WriteCR (  
    uint32 Value )
```

Settimer module control register.

Note

Function ID : DES_GPT_API_551

Parameters

in	<i>Value</i>	Value set to control register
----	--------------	-------------------------------

Returns

none

Definition at line 75 of file AC784xx_Timer_Reg.h.

4.2.2.10 Timer_Reg_WriteCTRL()

```
LOCAL_INLINE void Timer_Reg_WriteCTRL (  
    TIMER_CHANNEL_Type * Base,  
    uint32 Value )
```

Set CTRL register.

Note

Function ID : DES_GPT_API_562

Parameters

in	<i>Base</i>	Base address to TIMER_CHANNEL_Type.
in	<i>Value</i>	Value set to register

Returns

none

Definition at line 209 of file AC784xx_Timer_Reg.h.

4.2.2.11 Timer_Reg_WriteCVAL()

```
LOCAL_INLINE void Timer_Reg_WriteCVAL (
    TIMER_CHANNEL_Type * Base,
    uint32 Value )
```

Set CVAL register.

Note

Function ID : DES_GPT_API_560

Parameters

in	<i>Base</i>	Base address to TIMER_CHANNEL_Type.
in	<i>Value</i>	Value set to register

Returns

none

Definition at line 185 of file AC784xx_Timer_Reg.h.

4.2.2.12 Timer_Reg_WriteENR()

```
LOCAL_INLINE void Timer_Reg_WriteENR (
    uint32 Value )
```

Set ENR register.

Note

Function ID : DES_GPT_API_557

Parameters

in	<i>Value</i>	Value set to register
----	--------------	-----------------------

Returns

none

Definition at line 139 of file AC784xx_Timer_Reg.h.

4.2.2.13 Timer_Reg_WriteIER()

```
LOCAL_INLINE void Timer_Reg_WriteIER (
    uint32 Value )
```

Set IER register.

Note

Function ID : DES_GPT_API_555

Parameters

in	Value	Value set to register
----	-------	-----------------------

Returns

none

Definition at line 118 of file AC784xx_Timer_Reg.h.

4.2.2.14 Timer_Reg_WriteSR()

```
LOCAL_INLINE void Timer_Reg_WriteSR (
    uint32 Value )
```

Set interrupt status register.

Note

Function ID : DES_GPT_API_553

Parameters

in	Value	Value set to interrupt status register
----	-------	--

Returns

none

Definition at line 96 of file AC784xx_Timer_Reg.h.

4.2.2.15 Timer_Reg_WriteTVAL()

```
LOCAL_INLINE void Timer_Reg_WriteTVAL (
    TIMER_CHANNEL_Type * Base,
    uint32 Value )
```

Set TVAL register.

Note

Function ID : DES_GPT_API_558

Parameters

in	<i>Base</i>	Base address to TIMER_CHANNEL_Type.
in	<i>Value</i>	Value set to register

Returns

none

Definition at line 162 of file AC784xx_Timer_Reg.h.

4.3 Timer_Hal.c File Reference

Timer HAL source file.

```
#include "Timer_Hal.h"
#include "AC784xx_Timer_Reg.h"
#include "Ckgen_Hal.h"
#include "Rcm_Hal.h"
#include "Ctu_Hal.h"
#include "Core_Hal.h"
```

Macros

- #define [TIMER_CHANNEL_CTRL_MODE_0](#) (0x00UL)
Register value for timer mode 0.
- #define [TIMER_CHANNEL_CTRL_MODE_1](#) (0x10UL)
Register value for timer mode 1.
- #define [TIMER_CHANNEL_CTRL_MODE_2](#) (0x20UL)
Register value for timer mode 2.
- #define [TIMER_CHANNEL_CTRL_MODE_3](#) (0x30UL)
Register value for timer mode 3.
- #define [TIMER_STATUS_INITED](#) 0x100U
Initialized flag of Timer HAL.
- #define [TIMER_STATUS_CH0_STARTED](#) 0x01UL
Started flag of timer 0.

Functions

- Hal_StatusType [Timer_Hal_Init](#) (Timer_ClockSourceType Clk)
Initialize timer module.
- Hal_StatusType [Timer_Hal_DeInit](#) (void)
Deinit the timer driver.
- Hal_StatusType [Timer_Hal_SetConfig](#) (uint32 Channel, const [Timer_Channel_ConfigType](#) *ConfigPtr)
Set configuration information for a timer channel.
- Hal_StatusType [Timer_Hal_GetConfig](#) (uint32 Channel, [Timer_Channel_ConfigType](#) *ConfigPtr)
Get configuration information of a timer channel.
- Hal_StatusType [Timer_Hal_Start](#) (uint32 Channel, uint32 Timeout)
Start a channel.
- Hal_StatusType [Timer_Hal_Stop](#) (uint32 Channel)
Stop a channel.
- uint32 [Timer_Hal_GetCurrentValue](#) (uint32 Channel)
Returns the time already elapsed.
- uint32 [Timer_Hal_GetRemainingValue](#) (uint32 Channel)
Returns the time remaining until the target time is reached.
- void [Timer_Hal_EnableInterrupt](#) (uint32 Channel, uint32 InterruptBits)
Enable channel interrupt.
- void [Timer_Hal_InstallCallback](#) (uint32 Channel, Hal_CallbackType Func, void *Args)
Install timer callback.
- Hal_StatusType [Timer_Hal_ResetValue](#) (uint32 Channel)
Reset current value of timer.
- uint32 [Timer_Hal_MicrosToTicks](#) (uint32 Micros)
Translate microseconds to tick of timer base on clock source.
- void [TIMER_Channel0_IRQHandler](#) (void)
- void [TIMER_Channel1_IRQHandler](#) (void)
- void [TIMER_Channel2_IRQHandler](#) (void)
- void [TIMER_Channel3_IRQHandler](#) (void)

4.3.1 Detailed Description

Timer HAL source file.

4.3.2 Macro Definition Documentation

4.3.2.1 TIMER_CHANNEL_CTRL_MODE_0

```
#define TIMER_CHANNEL_CTRL_MODE_0 (0x00UL)
```

Register value for timer mode 0.

Definition at line 55 of file Timer_Hal.c.

4.3.2.2 TIMER_CHANNEL_CTRL_MODE_1

```
#define TIMER_CHANNEL_CTRL_MODE_1 (0x10UL)
```

Register value for timer mode 1.

Definition at line 57 of file Timer_Hal.c.

4.3.2.3 TIMER_CHANNEL_CTRL_MODE_2

```
#define TIMER_CHANNEL_CTRL_MODE_2 (0x20UL)
```

Register value for timer mode 2.

Definition at line 59 of file Timer_Hal.c.

4.3.2.4 TIMER_CHANNEL_CTRL_MODE_3

```
#define TIMER_CHANNEL_CTRL_MODE_3 (0x30UL)
```

Register value for timer mode 3.

Definition at line 61 of file Timer_Hal.c.

4.3.2.5 TIMER_STATUS_CH0_STARTED

```
#define TIMER_STATUS_CH0_STARTED 0x01UL
```

Started flag of timer 0.

Definition at line 67 of file Timer_Hal.c.

4.3.2.6 TIMER_STATUS_INITED

```
#define TIMER_STATUS_INITED 0x100U
```

Initialized flag of Timer HAL.

Definition at line 64 of file Timer_Hal.c.

4.3.3 Function Documentation

4.3.3.1 TIMER_Channel0_IRQHandler()

```
void TIMER_Channel0_IRQHandler (  
    void )
```

Definition at line 444 of file Timer_Hal.c.

4.3.3.2 TIMER_Channel1_IRQHandler()

```
void TIMER_Channel1_IRQHandler (  
    void )
```

Definition at line 449 of file Timer_Hal.c.

4.3.3.3 TIMER_Channel2_IRQHandler()

```
void TIMER_Channel2_IRQHandler (  
    void )
```

Definition at line 454 of file Timer_Hal.c.

4.3.3.4 TIMER_Channel3_IRQHandler()

```
void TIMER_Channel3_IRQHandler (  
    void )
```

Definition at line 459 of file Timer_Hal.c.

4.3.3.5 Timer_Hal_DeInit()

```
Hal_StatusType Timer_Hal_DeInit (  
    void )
```

Deinit the timer driver.

Note

Function ID: DES_GPT_API_502
Service ID: none

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_STATE.

Precondition

Timer has been initialized and no timer channel is started.

Definition at line 154 of file Timer_Hal.c.

4.3.3.6 Timer_Hal_EnableInterrupt()

```
void Timer_Hal_EnableInterrupt (
    uint32 Channel,
    uint32 InterruptBits )
```

Enable channel interrupt.

Note

Function ID: DES_GPT_API_509
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
in	<i>InterruptBits</i>	0: Disabled, 1: Enabled.

Returns

void

Definition at line 382 of file Timer_Hal.c.

4.3.3.7 Timer_Hal_GetConfig()

```
Hal_StatusType Timer_Hal_GetConfig (
    uint32 Channel,
    Timer_Channel_ConfigType * ConfigPtr )
```

Get configuration information of a timer channel.

Note

Function ID: DES_GPT_API_504
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
out	<i>ConfigPtr</i>	Channel configuration.

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_PARAM.

Definition at line 186 of file Timer_Hal.c.

4.3.3.8 Timer_Hal_GetCurrentValue()

```
uint32 Timer_Hal_GetCurrentValue (
    uint32 Channel )
```

Returns the time already elapsed.

Note

Function ID: DES_GPT_API_507
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
----	----------------	----------------------

Returns

uint32: Elapsed timer value (in number of ticks).

Definition at line 355 of file Timer_Hal.c.

4.3.3.9 Timer_Hal_GetRemainingValue()

```
uint32 Timer_Hal_GetRemainingValue (
    uint32 Channel )
```

Returns the time remaining until the target time is reached.

Note

Function ID: DES_GPT_API_508
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
----	----------------	----------------------

Returns

Remaining value of timer.

Definition at line 368 of file Timer_Hal.c.

4.3.3.10 Timer_Hal_Init()

```
Hal_StatusType Timer_Hal_Init (
    Timer_ClockSourceType Clk )
```

Initialize timer module.

Note

Function ID: DES_GPT_API_501
Service ID: none

Parameters

<i>in</i>	<i>Clk</i>	Specify clock source of timer. TIMER_CLOCK_INVALID: doesn't set clock source.
-----------	------------	---

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_STATE.

See also

[Timer_ClockSourceType](#)

Precondition

Timer isn't initialized.

Definition at line 126 of file Timer_Hal.c.

4.3.3.11 Timer_Hal_InstallCallback()

```
void Timer_Hal_InstallCallback (
    uint32 Channel,
    Hal_CallbackType Func,
    void * Args )
```

Install timer callback.

Note

Function ID: DES_GPT_API_510
Service ID: none

Parameters

<i>in</i>	<i>Channel</i>	Timer channel ID.
<i>in</i>	<i>Func</i>	Pointer to a function.
<i>in</i>	<i>Args</i>	Pointer to void pointer parameter for callback function.

Returns

void

Definition at line 409 of file Timer_Hal.c.

4.3.3.12 Timer_Hal_MicrosToTicks()

```
uint32 Timer_Hal_MicrosToTicks (
    uint32 Micros )
```

Translate microseconds to tick of timer base on clock source.

Note

Function ID: DES_GPT_API_512
Service ID: none

Parameters

in	<i>Micros</i>	Microseconds
----	---------------	--------------

Returns

Tick value for the specified microseconds

Definition at line 432 of file Timer_Hal.c.

4.3.3.13 Timer_Hal_ResetValue()

```
Hal_StatusType Timer_Hal_ResetValue (
    uint32 Channel )
```

Reset current value of timer.

Note

Function ID: DES_GPT_API_511
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
----	----------------	----------------------

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_PARAM/STATUS_TIMER_WRONG_STATE.

Definition at line 419 of file Timer_Hal.c.

4.3.3.14 Timer_Hal_SetConfig()

```
Hal_StatusType Timer_Hal_SetConfig (
    uint32 Channel,
    const Timer_Channel_ConfigType * ConfigPtr )
```

Set configuration information for a timer channel.

Note

Function ID: DES_GPT_API_503
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
in	<i>ConfigPtr</i>	Channel configuration.

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_PARAM.

See also

[Timer_Channel_ConfigType](#)

Definition at line 172 of file Timer_Hal.c.

4.3.3.15 Timer_Hal_Start()

```
Hal_StatusType Timer_Hal_Start (
    uint32 Channel,
    uint32 Timeout )
```

Start a channel.

Note

Function ID: DES_GPT_API_505
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
in	<i>Timeout</i>	Timeout value to be set.

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_PARAM/STATUS_TIMER_WRONG_STATE.

Definition at line 201 of file Timer_Hal.c.

4.3.3.16 Timer_Hal_Stop()

```
Hal_StatusType Timer_Hal_Stop (
    uint32 Channel )
```

Stop a channel.

Note

Function ID: DES_GPT_API_506
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
----	----------------	----------------------

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_PARAM/STATUS_TIMER_WRONG_STATE.

Definition at line 323 of file Timer_Hal.c.

4.4 Timer_Hal.h File Reference

Timer HAL header file.

```
#include "Device_Register.h"
```

Classes

- struct [Timer_Channel_ConfigType](#)
Timer channel configure type.

Macros

- #define [TIMER_ONESHOT_EN](#) 0x01U
Timer oneshot enable.
- #define [TIMER_DBG_EN](#) 0x02U
Timer debug enable.
- #define [TIMER_CHN_EN](#) 0x04U
Link enable.
- #define [TIMER_TROT](#) 0x08U
TROT enable.

- #define [TIMER_TSOT](#) 0x10U
TSOT enable.
- #define [TIMER_IRQ_EN](#) 0x20U
Timer interrupt enable config.
- #define [TIMER_INT_EN](#) 0x01U
Timer interrupt flag.
- #define [STATUS_TIMER_WRONG_STATE](#) 0x91
- #define [STATUS_TIMER_WRONG_PARAM](#) 0x92

Enumerations

- enum [Timer_ClockSourceType](#) {
[TIMER_CLOCK_SPLL](#) = 0U, [TIMER_CLOCK_VHSI](#), [TIMER_CLOCK_HSI](#), [TIMER_CLOCK_HSE](#),
[TIMER_CLOCK_INVALID](#) }
Timer Clock source type.
- enum [Timer_ModeType](#) {
[TIMER_MODE_0](#) = 0U, [TIMER_MODE_1](#), [TIMER_MODE_2](#), [TIMER_MODE_3](#),
[TIMER_MODE_NUM](#) }
Timer work mode.

Functions

- Hal_StatusType [Timer_Hal_Init](#) ([Timer_ClockSourceType](#) Clk)
Initialize timer module.
- Hal_StatusType [Timer_Hal_DeInit](#) (void)
Deinit the timer driver.
- Hal_StatusType [Timer_Hal_SetConfig](#) (uint32 Channel, const [Timer_Channel_ConfigType](#) *ConfigPtr)
Set configuration information for a timer channel.
- Hal_StatusType [Timer_Hal_GetConfig](#) (uint32 Channel, [Timer_Channel_ConfigType](#) *ConfigPtr)
Get configuration information of a timer channel.
- Hal_StatusType [Timer_Hal_Start](#) (uint32 Channel, uint32 Timeout)
Start a channel.
- Hal_StatusType [Timer_Hal_Stop](#) (uint32 Channel)
Stop a channel.
- uint32 [Timer_Hal_GetCurrentValue](#) (uint32 Channel)
Returns the time already elapsed.
- uint32 [Timer_Hal_GetRemainingValue](#) (uint32 Channel)
Returns the time remaining until the target time is reached.
- void [Timer_Hal_EnableInterrupt](#) (uint32 Channel, uint32 InterruptBits)
Enable channel interrupt.
- void [Timer_Hal_InstallCallback](#) (uint32 Channel, Hal_CallbackType Func, void *Args)
Install timer callback.
- Hal_StatusType [Timer_Hal_ResetValue](#) (uint32 Channel)
Reset current value of timer.
- uint32 [Timer_Hal_MicrosToTicks](#) (uint32 Micros)
Translate microseconds to tick of timer base on clock source.

4.4.1 Detailed Description

Timer HAL header file.

4.4.2 Macro Definition Documentation

4.4.2.1 STATUS_TIMER_WRONG_PARAM

```
#define STATUS_TIMER_WRONG_PARAM 0x92
```

Definition at line 74 of file Timer_Hal.h.

4.4.2.2 STATUS_TIMER_WRONG_STATE

```
#define STATUS_TIMER_WRONG_STATE 0x91
```

Definition at line 73 of file Timer_Hal.h.

4.4.2.3 TIMER_CHN_EN

```
#define TIMER_CHN_EN 0x04U
```

Link enable.

Definition at line 63 of file Timer_Hal.h.

4.4.2.4 TIMER_DBG_EN

```
#define TIMER_DBG_EN 0x02U
```

Timer debug enable.

Definition at line 61 of file Timer_Hal.h.

4.4.2.5 TIMER_INT_EN

```
#define TIMER_INT_EN 0x01U
```

Timer interrupt flag.

Definition at line 71 of file Timer_Hal.h.

4.4.2.6 TIMER_IRQ_EN

```
#define TIMER_IRQ_EN 0x20U
```

Timer interrupt enable config.

Definition at line 69 of file Timer_Hal.h.

4.4.2.7 TIMER_ONESHOT_EN

```
#define TIMER_ONESHOT_EN 0x01U
```

Timer oneshot enable.

Definition at line 59 of file Timer_Hal.h.

4.4.2.8 TIMER_TROT

```
#define TIMER_TROT 0x08U
```

TROT enable.

Definition at line 65 of file Timer_Hal.h.

4.4.2.9 TIMER_TSOT

```
#define TIMER_TSOT 0x10U
```

TSOT enable.

Definition at line 67 of file Timer_Hal.h.

4.4.3 Enumeration Type Documentation

4.4.3.1 Timer_ClockSourceType

```
enum Timer\_ClockSourceType
```

Timer Clock source type.

Enumerator

TIMER_CLOCK_SPLL	Timer uses SPLL DIV2 as clock source.
TIMER_CLOCK_VHSI	Timer uses VHSI DIV2 as clock source.
TIMER_CLOCK_HSI	Timer uses HSI as clock source.
TIMER_CLOCK_HSE	Timer uses HSE DIV2 clock source.
TIMER_CLOCK_INVALID	Invalid clock source.

Definition at line 80 of file Timer_Hal.h.

4.4.3.2 Timer_ModeType

```
enum Timer_ModeType
```

Timer work mode.

Enumerator

TIMER_MODE_0	Timer work mode : 32 bit period timer.
TIMER_MODE_1	Timer work mode : dual 16 bit period timer.
TIMER_MODE_2	Timer work mode : 32 bit pulse accumulator.
TIMER_MODE_3	Timer work mode : 32 bit pulse capturer.
TIMER_MODE_NUM	Number of timer work modes.

Definition at line 92 of file Timer_Hal.h.

4.4.4 Function Documentation**4.4.4.1 Timer_Hal_DeInit()**

```
Hal_StatusType Timer_Hal_DeInit (  
    void )
```

Deinit the timer driver.

Note

Function ID: DES_GPT_API_502
Service ID: none

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_STATE.

Precondition

Timer has been initialized and no timer channel is started.

Definition at line 154 of file Timer_Hal.c.

4.4.4.2 Timer_Hal_EnableInterrupt()

```
void Timer_Hal_EnableInterrupt (
    uint32 Channel,
    uint32 InterruptBits )
```

Enable channel interrupt.

Note

Function ID: DES_GPT_API_509
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
in	<i>InterruptBits</i>	0: Disabled, 1: Enabled.

Returns

void

Definition at line 382 of file Timer_Hal.c.

4.4.4.3 Timer_Hal_GetConfig()

```
Hal_StatusType Timer_Hal_GetConfig (
    uint32 Channel,
    Timer_Channel_ConfigType * ConfigPtr )
```

Get configuration information of a timer channel.

Note

Function ID: DES_GPT_API_504
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
out	<i>ConfigPtr</i>	Channel configuration.

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_PARAM.

Definition at line 186 of file Timer_Hal.c.

4.4.4.4 Timer_Hal_GetCurrentValue()

```
uint32 Timer_Hal_GetCurrentValue (
    uint32 Channel )
```

Returns the time already elapsed.

Note

Function ID: DES_GPT_API_507
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
----	----------------	----------------------

Returns

uint32: Elapsed timer value (in number of ticks).

Definition at line 355 of file Timer_Hal.c.

4.4.4.5 Timer_Hal_GetRemainingValue()

```
uint32 Timer_Hal_GetRemainingValue (
    uint32 Channel )
```

Returns the time remaining until the target time is reached.

Note

Function ID: DES_GPT_API_508
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
----	----------------	----------------------

Returns

Remaining value of timer.

Definition at line 368 of file Timer_Hal.c.

4.4.4.6 Timer_Hal_Init()

```
Hal_StatusType Timer_Hal_Init (
    Timer_ClockSourceType Clk )
```

Initialize timer module.

Note

Function ID: DES_GPT_API_501
Service ID: none

Parameters

<i>in</i>	<i>Clk</i>	Specify clock source of timer. TIMER_CLOCK_INVALID: doesn't set clock source.
-----------	------------	---

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_STATE.

See also

[Timer_ClockSourceType](#)

Precondition

Timer isn't initialized.

Definition at line 126 of file Timer_Hal.c.

4.4.4.7 Timer_Hal_InstallCallback()

```
void Timer_Hal_InstallCallback (
    uint32 Channel,
    Hal_CallbackType Func,
    void * Args )
```

Install timer callback.

Note

Function ID: DES_GPT_API_510
Service ID: none

Parameters

<i>in</i>	<i>Channel</i>	Timer channel ID.
<i>in</i>	<i>Func</i>	Pointer to a function.
<i>in</i>	<i>Args</i>	Pointer to void pointer parameter for callback function.

Returns

void

Definition at line 409 of file Timer_Hal.c.

4.4.4.8 Timer_Hal_MicrosToTicks()

```
uint32 Timer_Hal_MicrosToTicks (
    uint32 Micros )
```

Translate microseconds to tick of timer base on clock source.

Note

Function ID: DES_GPT_API_512
Service ID: none

Parameters

in	<i>Micros</i>	Microseconds
----	---------------	--------------

Returns

Tick value for the specified microseconds

Definition at line 432 of file Timer_Hal.c.

4.4.4.9 Timer_Hal_ResetValue()

```
Hal_StatusType Timer_Hal_ResetValue (
    uint32 Channel )
```

Reset current value of timer.

Note

Function ID: DES_GPT_API_511
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
----	----------------	----------------------

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_PARAM/STATUS_TIMER_WRONG_STATE.

Definition at line 419 of file Timer_Hal.c.

4.4.4.10 Timer_Hal_SetConfig()

```
Hal_StatusType Timer_Hal_SetConfig (
    uint32 Channel,
    const Timer_Channel_ConfigType * ConfigPtr )
```

Set configuration information for a timer channel.

Note

Function ID: DES_GPT_API_503
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
in	<i>ConfigPtr</i>	Channel configuration.

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_PARAM.

See also

[Timer_Channel_ConfigType](#)

Definition at line 172 of file Timer_Hal.c.

4.4.4.11 Timer_Hal_Start()

```
Hal_StatusType Timer_Hal_Start (
    uint32 Channel,
    uint32 Timeout )
```

Start a channel.

Note

Function ID: DES_GPT_API_505
Service ID: none

Parameters

in	<i>Channel</i>	Timer channel index.
in	<i>Timeout</i>	Timeout value to be set.

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_PARAM/STATUS_TIMER_WRONG_STATE.

Definition at line 201 of file Timer_Hal.c.

4.4.4.12 Timer_Hal_Stop()

```
Hal_StatusType Timer_Hal_Stop (
    uint32 Channel )
```

Stop a channel.

Note

Function ID: DES_GPT_API_506
Service ID: none

Parameters

in	Channel	Timer channel index.
----	---------	----------------------

Returns

Hal_StatusType. if successful return STATUS_SUCCESS or return STATUS_TIMER_WRONG_PARAM/STATUS←
_TIMER_WRONG_STATE.

Definition at line 323 of file Timer_Hal.c.

Index

AC784xx_API_Reference_Manual_TIMER.pdf, [5](#)

AC784xx_Timer_Reg.h, [5](#)

Timer_Reg_GetChBase, [6](#)

Timer_Reg_ReadCTRL, [6](#)

Timer_Reg_ReadCVAL, [7](#)

Timer_Reg_ReadCR, [6](#)

Timer_Reg_ReadENR, [7](#)

Timer_Reg_ReadIER, [8](#)

Timer_Reg_ReadSR, [8](#)

Timer_Reg_ReadTVAL, [8](#)

Timer_Reg_WriteCTRL, [9](#)

Timer_Reg_WriteCVAL, [10](#)

Timer_Reg_WriteCR, [9](#)

Timer_Reg_WriteENR, [10](#)

Timer_Reg_WriteIER, [11](#)

Timer_Reg_WriteSR, [11](#)

Timer_Reg_WriteTVAL, [11](#)

Config

Timer_Channel_ConfigType, [3](#)

Mode

Timer_Channel_ConfigType, [3](#)

STATUS_TIMER_WRONG_PARAM

Timer_Hal.h, [23](#)

STATUS_TIMER_WRONG_STATE

Timer_Hal.h, [23](#)

TIMER_CHANNEL_CTRL_MODE_0

Timer_Hal.c, [13](#)

TIMER_CHANNEL_CTRL_MODE_1

Timer_Hal.c, [13](#)

TIMER_CHANNEL_CTRL_MODE_2

Timer_Hal.c, [14](#)

TIMER_CHANNEL_CTRL_MODE_3

Timer_Hal.c, [14](#)

TIMER_CHN_EN

Timer_Hal.h, [23](#)

TIMER_Channel0_IRQHandler

Timer_Hal.c, [14](#)

TIMER_Channel1_IRQHandler

Timer_Hal.c, [15](#)

TIMER_Channel2_IRQHandler

Timer_Hal.c, [15](#)

TIMER_Channel3_IRQHandler

Timer_Hal.c, [15](#)

TIMER_DBG_EN

Timer_Hal.h, [23](#)

TIMER_INT_EN

Timer_Hal.h, [23](#)

TIMER_IRQ_EN

Timer_Hal.h, [23](#)

TIMER_ONESHOT_EN

Timer_Hal.h, [24](#)

TIMER_STATUS_CH0_STARTED

Timer_Hal.c, [14](#)

TIMER_STATUS_INITED

Timer_Hal.c, [14](#)

TIMER_TROT

Timer_Hal.h, [24](#)

TIMER_TSOT

Timer_Hal.h, [24](#)

Timer_Channel_ConfigType, [3](#)

Config, [3](#)

Mode, [3](#)

TriggerSrc, [4](#)

Timer_ClockSourceType

Timer_Hal.h, [24](#)

Timer_Hal.c, [12](#)

TIMER_CHANNEL_CTRL_MODE_0, [13](#)

TIMER_CHANNEL_CTRL_MODE_1, [13](#)

TIMER_CHANNEL_CTRL_MODE_2, [14](#)

TIMER_CHANNEL_CTRL_MODE_3, [14](#)

TIMER_Channel0_IRQHandler, [14](#)

TIMER_Channel1_IRQHandler, [15](#)

TIMER_Channel2_IRQHandler, [15](#)

TIMER_Channel3_IRQHandler, [15](#)

TIMER_STATUS_CH0_STARTED, [14](#)

TIMER_STATUS_INITED, [14](#)

Timer_Hal_DeInit, [15](#)

Timer_Hal_EnableInterrupt, [15](#)

Timer_Hal_GetConfig, [16](#)

Timer_Hal_GetCurrentValue, [16](#)

Timer_Hal_GetRemainingValue, [17](#)

Timer_Hal_Init, [17](#)

Timer_Hal_InstallCallback, [18](#)

Timer_Hal_MicrosToTicks, [18](#)

Timer_Hal_ResetValue, [19](#)

Timer_Hal_SetConfig, [19](#)

Timer_Hal_Start, [20](#)

Timer_Hal_Stop, [21](#)

Timer_Hal.h, [21](#)

STATUS_TIMER_WRONG_PARAM, [23](#)

STATUS_TIMER_WRONG_STATE, [23](#)

TIMER_CHN_EN, [23](#)

TIMER_DBG_EN, [23](#)

TIMER_INT_EN, [23](#)

TIMER_IRQ_EN, [23](#)

TIMER_ONESHOT_EN, [24](#)

TIMER_TROT, [24](#)

TIMER_TSOT, [24](#)

Timer_ClockSourceType, [24](#)

Timer_Hal_DeInit, [25](#)

Timer_Hal_EnableInterrupt, [25](#)

- Timer_Hal_GetConfig, [26](#)
- Timer_Hal_GetCurrentValue, [26](#)
- Timer_Hal_GetRemainingValue, [27](#)
- Timer_Hal_Init, [27](#)
- Timer_Hal_InstallCallback, [28](#)
- Timer_Hal_MicrosToTicks, [28](#)
- Timer_Hal_ResetValue, [29](#)
- Timer_Hal_SetConfig, [29](#)
- Timer_Hal_Start, [30](#)
- Timer_Hal_Stop, [31](#)
- Timer_ModeType, [25](#)
- Timer_Hal_DeInit
 - Timer_Hal.c, [15](#)
 - Timer_Hal.h, [25](#)
- Timer_Hal_EnableInterrupt
 - Timer_Hal.c, [15](#)
 - Timer_Hal.h, [25](#)
- Timer_Hal_GetConfig
 - Timer_Hal.c, [16](#)
 - Timer_Hal.h, [26](#)
- Timer_Hal_GetCurrentValue
 - Timer_Hal.c, [16](#)
 - Timer_Hal.h, [26](#)
- Timer_Hal_GetRemainingValue
 - Timer_Hal.c, [17](#)
 - Timer_Hal.h, [27](#)
- Timer_Hal_Init
 - Timer_Hal.c, [17](#)
 - Timer_Hal.h, [27](#)
- Timer_Hal_InstallCallback
 - Timer_Hal.c, [18](#)
 - Timer_Hal.h, [28](#)
- Timer_Hal_MicrosToTicks
 - Timer_Hal.c, [18](#)
 - Timer_Hal.h, [28](#)
- Timer_Hal_ResetValue
 - Timer_Hal.c, [19](#)
 - Timer_Hal.h, [29](#)
- Timer_Hal_SetConfig
 - Timer_Hal.c, [19](#)
 - Timer_Hal.h, [29](#)
- Timer_Hal_Start
 - Timer_Hal.c, [20](#)
 - Timer_Hal.h, [30](#)
- Timer_Hal_Stop
 - Timer_Hal.c, [21](#)
 - Timer_Hal.h, [31](#)
- Timer_ModeType
 - Timer_Hal.h, [25](#)
- Timer_Reg_GetChBase
 - AC784xx_Timer_Reg.h, [6](#)
- Timer_Reg_ReadCTRL
 - AC784xx_Timer_Reg.h, [6](#)
- Timer_Reg_ReadCVAL
 - AC784xx_Timer_Reg.h, [7](#)
- Timer_Reg_ReadCR
 - AC784xx_Timer_Reg.h, [6](#)
- Timer_Reg_ReadENR
 - AC784xx_Timer_Reg.h, [7](#)
- Timer_Reg_ReadIER
 - AC784xx_Timer_Reg.h, [8](#)
- Timer_Reg_ReadSR
 - AC784xx_Timer_Reg.h, [8](#)
- Timer_Reg_ReadTVAL
 - AC784xx_Timer_Reg.h, [8](#)
- Timer_Reg_WriteCTRL
 - AC784xx_Timer_Reg.h, [9](#)
- Timer_Reg_WriteCVAL
 - AC784xx_Timer_Reg.h, [10](#)
- Timer_Reg_WriteCR
 - AC784xx_Timer_Reg.h, [9](#)
- Timer_Reg_WriteENR
 - AC784xx_Timer_Reg.h, [10](#)
- Timer_Reg_WriteIER
 - AC784xx_Timer_Reg.h, [11](#)
- Timer_Reg_WriteSR
 - AC784xx_Timer_Reg.h, [11](#)
- Timer_Reg_WriteTVAL
 - AC784xx_Timer_Reg.h, [11](#)
- TriggerSrc
 - Timer_Channel_ConfigType, [4](#)