

AC784xx_DFP RTC

3.1.0

Generated by Doxygen 1.8.13

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	2
2.1	AC784xx_API_Reference_Manual_RTC.pdf File Reference	2
2.2	AC784xx_Rtc_Reg.h File Reference	2
2.2.1	Detailed Description	3
2.2.2	Function Documentation	3
2.2.2.1	Rtc_Reg_Enable()	3
2.2.2.2	Rtc_Reg_ReadAlarm()	3
2.2.2.3	Rtc_Reg_ReadCTRL()	4
2.2.2.4	Rtc_Reg_ReadLR()	4
2.2.2.5	Rtc_Reg_ReadPSC()	4
2.2.2.6	Rtc_Reg_ReadPSR()	5
2.2.2.7	Rtc_Reg_ReadSR()	5
2.2.2.8	Rtc_Reg_ReadTC()	5
2.2.2.9	Rtc_Reg_Reset()	5
2.2.2.10	Rtc_Reg_WriteAlarm()	6
2.2.2.11	Rtc_Reg_WriteCTRL()	6
2.2.2.12	Rtc_Reg_WriteLR()	7
2.2.2.13	Rtc_Reg_WritePSR()	7
2.2.2.14	Rtc_Reg_WriteSR()	7
2.2.2.15	Rtc_Reg_WriteTC()	8
2.3	Rtc_Hal.c File Reference	8
2.3.1	Detailed Description	9

2.3.2	Macro Definition Documentation	9
2.3.2.1	CLEAR_TIF_VALUE	9
2.3.3	Enumeration Type Documentation	9
2.3.3.1	Rtc_StateType	9
2.3.4	Function Documentation	10
2.3.4.1	Rtc_Hal_DeInit()	10
2.3.4.2	Rtc_Hal_EnableInterrupt()	10
2.3.4.3	Rtc_Hal_GetCurrentValue()	11
2.3.4.4	Rtc_Hal_GetPrescalerValue()	11
2.3.4.5	Rtc_Hal_GetWorkFreq()	12
2.3.4.6	Rtc_Hal_Init()	12
2.3.4.7	Rtc_Hal_InstallCallback()	12
2.3.4.8	Rtc_Hal_Pause()	13
2.3.4.9	Rtc_Hal_SetAlarm()	13
2.3.4.10	Rtc_Hal_SetConfig()	14
2.3.4.11	Rtc_Hal_SetCurrentValue()	14
2.3.4.12	Rtc_Hal_Start()	15
2.3.4.13	Rtc_Hal_Stop()	15
2.3.4.14	RTC_IRQHandler()	16
2.4	Rtc_Hal.h File Reference	16
2.4.1	Detailed Description	17
2.4.2	Macro Definition Documentation	17
2.4.2.1	RTC_EVENT_ALARM	17
2.4.2.2	RTC_EVENT_INVALID	18
2.4.2.3	RTC_EVENT_PRESCALER	18
2.4.2.4	RTC_EVENT_TIMER	18
2.4.2.5	RTC_INT_ALARM_EN	18
2.4.2.6	RTC_INT_ALL_EN	18
2.4.2.7	RTC_INT_INVALID_EN	19
2.4.2.8	RTC_INT_PRESCALER_EN	19
2.4.2.9	RTC_INT_TIMER_EN	19

2.4.3	Typedef Documentation	19
2.4.3.1	Rtc_Hal_CallbackType	19
2.4.4	Enumeration Type Documentation	20
2.4.4.1	Rtc_ClockOutputType	20
2.4.4.2	Rtc_ClockSourceType	20
2.4.5	Function Documentation	20
2.4.5.1	Rtc_Hal_DeInit()	20
2.4.5.2	Rtc_Hal_EnableInterrupt()	21
2.4.5.3	Rtc_Hal_GetCurrentValue()	21
2.4.5.4	Rtc_Hal_GetPrescalerValue()	22
2.4.5.5	Rtc_Hal_GetWorkFreq()	22
2.4.5.6	Rtc_Hal_Init()	22
2.4.5.7	Rtc_Hal_InstallCallback()	23
2.4.5.8	Rtc_Hal_Pause()	23
2.4.5.9	Rtc_Hal_SetAlarm()	24
2.4.5.10	Rtc_Hal_SetConfig()	24
2.4.5.11	Rtc_Hal_SetCurrentValue()	24
2.4.5.12	Rtc_Hal_Start()	25
2.4.5.13	Rtc_Hal_Stop()	25

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

AC784xx_API_Reference_Manual_RTC.pdf	2
AC784xx_Rtc_Reg.h	
Rtc access register inline function definition	2
Rtc_Hal.c	
Rtc hal source file	8
Rtc_Hal.h	
Rtc hal define	16

Chapter 2

File Documentation

2.1 AC784xx_API_Reference_Manual_RTC.pdf File Reference

2.2 AC784xx_Rtc_Reg.h File Reference

rtc access register inline function definition.

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

Functions

- LOCAL_INLINE void [Rtc_Reg_WriteCTRL](#) (uint32 Value)
Write RTC's CTRL register.
- LOCAL_INLINE uint32 [Rtc_Reg_ReadCTRL](#) (void)
Read RTC's CTRL register.
- LOCAL_INLINE void [Rtc_Reg_WriteAlarm](#) (uint32 Value)
Write RTC's alarm register.
- LOCAL_INLINE uint32 [Rtc_Reg_ReadAlarm](#) (void)
Read RTC's alarm register.
- LOCAL_INLINE void [Rtc_Reg_WriteTC](#) (uint32 Value)
Write RTC's TC register.
- LOCAL_INLINE uint32 [Rtc_Reg_ReadTC](#) (void)
Read RTC's TC register.
- LOCAL_INLINE void [Rtc_Reg_WritePSR](#) (uint32 Value)
Write RTC's PSR register.
- LOCAL_INLINE uint32 [Rtc_Reg_ReadPSR](#) (void)
Read RTC's PSR register.
- LOCAL_INLINE uint32 [Rtc_Reg_ReadPSC](#) (void)
Read RTC's PSC register.
- LOCAL_INLINE void [Rtc_Reg_WriteSR](#) (uint32 Value)
Write RTC's alarm register.
- LOCAL_INLINE uint32 [Rtc_Reg_ReadSR](#) (void)
Read RTC's SR register.
- LOCAL_INLINE void [Rtc_Reg_WriteLR](#) (uint32 Value)
Write RTC's alarm register.
- LOCAL_INLINE uint32 [Rtc_Reg_ReadLR](#) (void)
Read RTC's LR register.
- LOCAL_INLINE void [Rtc_Reg_Reset](#) (void)
Set rtc software reset.
- LOCAL_INLINE void [Rtc_Reg_Enable](#) (boolean IsEnable)
SEnable or disable rtc module.

2.2.1 Detailed Description

rtc access register inline function definition.

2.2.2 Function Documentation

2.2.2.1 Rtc_Reg_Enable()

```
LOCAL_INLINE void Rtc_Reg_Enable (
    boolean IsEnable )
```

SEnable or disable rtc module.

Note

Function ID : DES_GPT_API_663

Parameters

in	<i>IsEnable</i>	true enable rtc module; false disable rtc module
in, out	<i>None</i>	
out	<i>None</i>	

Returns

None

Definition at line 224 of file AC784xx_Rtc_Reg.h.

2.2.2.2 Rtc_Reg_ReadAlarm()

```
LOCAL_INLINE uint32 Rtc_Reg_ReadAlarm (
    void )
```

Read RTC's alarm regiser.

Note

Function ID : DES_GPT_API_653

Returns

uint32: value of alarm register

Definition at line 105 of file AC784xx_Rtc_Reg.h.

2.2.2.3 Rtc_Reg_ReadCTRL()

```
LOCAL_INLINE uint32 Rtc_Reg_ReadCTRL (
    void )
```

Read RTC's CTRL register.

Note

Function ID : DES_GPT_API_651

Returns

uint32: value of CTRL register

Definition at line 85 of file AC784xx_Rtc_Reg.h.

2.2.2.4 Rtc_Reg_ReadLR()

```
LOCAL_INLINE uint32 Rtc_Reg_ReadLR (
    void )
```

Read RTC's LR register.

Note

Function ID : DES_GPT_API_661

Returns

uint32: value of LR register

Definition at line 197 of file AC784xx_Rtc_Reg.h.

2.2.2.5 Rtc_Reg_ReadPSC()

```
LOCAL_INLINE uint32 Rtc_Reg_ReadPSC (
    void )
```

Read RTC's PSC register.

Note

Function ID : DES_GPT_API_658

Returns

uint32: value of PSC register

Definition at line 156 of file AC784xx_Rtc_Reg.h.

2.2.2.6 Rtc_Reg_ReadPSR()

```
LOCAL_INLINE uint32 Rtc_Reg_ReadPSR (  
    void )
```

Read RTC's PSR regiser.

Note

Function ID : DES_GPT_API_657

Returns

uint32: value of PSR register

Definition at line 145 of file AC784xx_Rtc_Reg.h.

2.2.2.7 Rtc_Reg_ReadSR()

```
LOCAL_INLINE uint32 Rtc_Reg_ReadSR (  
    void )
```

Read RTC's SR regiser.

Note

Function ID:

Returns

uint32: value of SR register

Definition at line 176 of file AC784xx_Rtc_Reg.h.

2.2.2.8 Rtc_Reg_ReadTC()

```
LOCAL_INLINE uint32 Rtc_Reg_ReadTC (  
    void )
```

Read RTC's TC regiser.

Note

Function ID : DES_GPT_API_655

Returns

uint32: value of TC register

Definition at line 125 of file AC784xx_Rtc_Reg.h.

2.2.2.9 Rtc_Reg_Reset()

```
LOCAL_INLINE void Rtc_Reg_Reset (  
    void )
```

Set rtc software reset.

Note

Function ID : DES_GPT_API_662

Parameters

in	<i>None</i>	
in, out	<i>None</i>	
out	<i>None</i>	

Returns

None

Definition at line 210 of file AC784xx_Rtc_Reg.h.

2.2.2.10 Rtc_Reg_WriteAlarm()

```
LOCAL_INLINE void Rtc_Reg_WriteAlarm (
    uint32 Value )
```

Write RTC's alarm register.

Note

Function ID : DES_GPT_API_652

Parameters

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

Definition at line 95 of file AC784xx_Rtc_Reg.h.

2.2.2.11 Rtc_Reg_WriteCTRL()

```
LOCAL_INLINE void Rtc_Reg_WriteCTRL (
    uint32 Value )
```

Write RTC's CTRL register.

Note

Function ID : DES_GPT_API_650

Parameters

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

Definition at line 75 of file AC784xx_Rtc_Reg.h.

2.2.2.12 Rtc_Reg_WriteLR()

```
LOCAL_INLINE void Rtc_Reg_WriteLR (
    uint32 Value )
```

Write RTC's alarm register.

Note

Function ID : DES_GPT_API_660

Parameters

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

Definition at line 186 of file AC784xx_Rtc_Reg.h.

2.2.2.13 Rtc_Reg_WritePSR()

```
LOCAL_INLINE void Rtc_Reg_WritePSR (
    uint32 Value )
```

Write RTC's PSR register.

Note

Function ID : DES_GPT_API_656

Parameters

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

Definition at line 135 of file AC784xx_Rtc_Reg.h.

2.2.2.14 Rtc_Reg_WriteSR()

```
LOCAL_INLINE void Rtc_Reg_WriteSR (
    uint32 Value )
```

Write RTC's alarm register.

Note

Function ID : DES_GPT_API_659

Parameters

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

Definition at line 166 of file AC784xx_Rtc_Reg.h.

2.2.2.15 Rtc_Reg_WriteTC()

```
LOCAL_INLINE void Rtc_Reg_WriteTC (
    uint32 Value )
```

Write RTC's TC register.

Note

Function ID : DES_GPT_API_654

Parameters

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

Definition at line 115 of file AC784xx_Rtc_Reg.h.

2.3 Rtc_Hal.c File Reference

rtc hal source file.

```
#include "Rtc_Hal.h"
#include "Ckgen_Hal.h"
#include "Core_Hal.h"
#include "AC784xx_Rtc_Reg.h"
#include "OsIf_Time.h"
```

Macros

- #define [CLEAR_TIF_VALUE](#) (0x01U)
The value(none zero) in order to clear TIF flag.

Enumerations

- enum [Rtc_StateType](#) { [RTC_UNINITED](#) = 0x0U, [RTC_IDLE](#), [RTC_PAUSED](#), [RTC_RUNNING](#) }
RTC state type.

Functions

- Hal_StatusType [Rtc_Hal_Init](#) ([Rtc_ClockSourceType](#) Clk)
Initialize rtc module.
- Hal_StatusType [Rtc_Hal_DeInit](#) (void)
Deinitialize rtc module.
- Hal_StatusType [Rtc_Hal_Start](#) (uint32 Unit)
Start RTC.
- Hal_StatusType [Rtc_Hal_Stop](#) (void)
Stop RTC.
- Hal_StatusType [Rtc_Hal_Pause](#) (void)
Pause RTC.
- uint32 [Rtc_Hal_GetCurrentValue](#) (void)
Get RTC real time value.
- void [Rtc_Hal_SetCurrentValue](#) (uint32 Value)
Set RTC real time counter.
- Hal_StatusType [Rtc_Hal_SetAlarm](#) (uint32 Value)
Set alarm count.
- void [Rtc_Hal_InstallCallback](#) (const [Rtc_Hal_CallbackType](#) Func)
Install counter overflow interrupt callback.
- void [Rtc_Hal_EnableInterrupt](#) (uint32 InterruptBits)
Enable or disable rtc interrupt.
- Hal_StatusType [Rtc_Hal_SetConfig](#) ([Rtc_ClockOutputType](#) OutputCfg)
Configure RTC output.
- uint32 [Rtc_Hal_GetWorkFreq](#) (void)
Get rtc work frequency.
- uint32 [Rtc_Hal_GetPrescalerValue](#) (void)
Get prescaler value.
- void [RTC_IRQHandler](#) (void)

2.3.1 Detailed Description

rtc hal source file.

2.3.2 Macro Definition Documentation

2.3.2.1 CLEAR_TIF_VALUE

```
#define CLEAR_TIF_VALUE (0x01U)
```

The value(none zero) in order to clear TIF flag.

Definition at line 54 of file Rtc_Hal.c.

2.3.3 Enumeration Type Documentation

2.3.3.1 Rtc_StateType

```
enum Rtc\_StateType
```

RTC state type.

Enumerator

RTC_UNINITED	Un initialization status.
RTC_IDLE	RTC has been initialized and is idle
RTC_PAUSED	RTC is paused.
RTC_RUNNING	RTC is running.

Definition at line 69 of file Rtc_Hal.c.

2.3.4 Function Documentation**2.3.4.1 Rtc_Hal_DeInit()**

```
Hal_StatusType Rtc_Hal_DeInit (  
    void )
```

Deinitialize rtc module.

Note

Function ID: DES_GPT_API_602

Returns

Hal_StatusType: STATUS_SUCCESS: success; STATUS_ERROR: RTC isn't initialized or is started.

Definition at line 168 of file Rtc_Hal.c.

2.3.4.2 Rtc_Hal_EnableInterrupt()

```
void Rtc_Hal_EnableInterrupt (  
    uint32 InterruptBits )
```

Enable or disable rtc interrupt.

Note

Function ID: DES_GPT_API_608

Parameters

in	<i>InterruptBits</i>	RTC interrupt config. Bit 0: RTC Timer overflow interrupt enable; Bit 1: Alarm interrupt enable; Bit 2: Prescaler interrupt enable.
----	----------------------	---

Returns

void

Definition at line 296 of file Rtc_Hal.c.

2.3.4.3 Rtc_Hal_GetCurrentValue()

```
uint32 Rtc_Hal_GetCurrentValue (
    void )
```

Get RTC real time value.

Note

Function ID: DES_GPT_API_605

Returns

uint32: Value of RTC timer register.

Definition at line 256 of file Rtc_Hal.c.

2.3.4.4 Rtc_Hal_GetPrescalerValue()

```
uint32 Rtc_Hal_GetPrescalerValue (
    void )
```

Get prescaler value.

Note

Function ID: DES_GPT_API_611

Returns

uint32: Value of RTC prescaler register.

Definition at line 352 of file Rtc_Hal.c.

2.3.4.5 Rtc_Hal_GetWorkFreq()

```
uint32 Rtc_Hal_GetWorkFreq (
    void )
```

Get rtc work frequency.

Note

Function ID: DES_GPT_API_610

Returns

uint32: work frequency

Definition at line 327 of file Rtc_Hal.c.

2.3.4.6 Rtc_Hal_Init()

```
Hal_StatusType Rtc_Hal_Init (
    Rtc_ClockSourceType Clk )
```

Initialize rtc module.

Note

Function ID: DES_GPT_API_601

Parameters

in	Clk	Specify RTC work clock source. RTC_CLOCK_INVALID: doesn't set clock source.
----	-----	---

Returns

Hal_StatusType: Initialize success or not, the range is the STATUS_SUCCESS STATUS_ERROR

Definition at line 121 of file Rtc_Hal.c.

2.3.4.7 Rtc_Hal_InstallCallback()

```
void Rtc_Hal_InstallCallback (
    const Rtc_Hal_CallbackType Func )
```

Install counter overflow interrupt callback.

Note

Function ID: DES_GPT_API_607

Parameters

in	<i>Func</i>	the callback function
----	-------------	-----------------------

Returns

void

Definition at line 291 of file Rtc_Hal.c.

2.3.4.8 Rtc_Hal_Pause()

```
Hal_StatusType Rtc_Hal_Pause (  
    void )
```

Pause RTC.

Precondition

Pre-condition: RTC is started.

Note

Function ID: DES_GPT_API_613

Returns

Hal_StatusType: STATUS_SUCCESS: success; STATUS_ERROR: RTC isn't started.

Definition at line 240 of file Rtc_Hal.c.

2.3.4.9 Rtc_Hal_SetAlarm()

```
Hal_StatusType Rtc_Hal_SetAlarm (  
    uint32 Value )
```

Set alarm count.

Note

Function ID: DES_GPT_API_606

Parameters

in	<i>Value</i>	Count for alarm.
----	--------------	------------------

Returns

Hal_StatusType: Set success or not, the range is the STATUS_SUCCESS STATUS_ERROR

Definition at line 276 of file Rtc_Hal.c.

2.3.4.10 Rtc_Hal_SetConfig()

```
Hal_StatusType Rtc_Hal_SetConfig (
    Rtc_ClockOutputType OutputCfg )
```

Configure RTC output.

Note

Function ID: DES_GPT_API_609

Parameters

in	<i>OutputCfg</i>	: RTC output configuration. 0 : Disabled, 1: RTC clock output, 2: RTC prescaler output.
----	------------------	---

Returns

Hal_StatusType: STATUS_SUCCESS: success;

Definition at line 312 of file Rtc_Hal.c.

2.3.4.11 Rtc_Hal_SetCurrentValue()

```
void Rtc_Hal_SetCurrentValue (
    uint32 Value )
```

Set RTC real time counter.

Precondition

Pre-condition: RTC isn't started.

Note

Function ID: DES_GPT_API_612

Parameters

in	<i>Value</i>	Value of RTC real timer.
----	--------------	--------------------------

Returns

void

Definition at line 267 of file Rtc_Hal.c.

2.3.4.12 Rtc_Hal_Start()

```
Hal_StatusType Rtc_Hal_Start (
    uint32 Unit )
```

Start RTC.

Note

Function ID: DES_GPT_API_603

Parameters

in	Unit	Value for RTC prescaler. And it is the unit of RTC real time register.
----	------	--

Returns

Hal_StatusType: STATUS_SUCCESS: success; STATUS_ERROR: RTC isn't initialized or has been started.

Definition at line 181 of file Rtc_Hal.c.

2.3.4.13 Rtc_Hal_Stop()

```
Hal_StatusType Rtc_Hal_Stop (
    void )
```

Stop RTC.

Note

Function ID: DES_GPT_API_604

Returns

Hal_StatusType: STATUS_SUCCESS: success; STATUS_ERROR: RTC isn't initialized or isn't started.

Definition at line 216 of file Rtc_Hal.c.

2.3.4.14 RTC_IRQHandler()

```
void RTC_IRQHandler (
    void )
```

Definition at line 363 of file Rtc_Hal.c.

2.4 Rtc_Hal.h File Reference

rtc hal define.

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

Macros

- `#define RTC_INT_TIMER_EN 0x01U`
Enable flag of RTC timer overflow interrupt.
- `#define RTC_INT_ALARM_EN 0x02U`
Enable flag(bit) of RTC alarm interrupt.
- `#define RTC_INT_PRESCALER_EN 0x04U`
Enable flag of RTC prescaler overflow interrupt.
- `#define RTC_INT_INVALID_EN 0x08U`
Enable flag of invalid interrupt.
- `#define RTC_INT_ALL_EN 0x0FU`
Enable flags of RTC all interrupt.
- `#define RTC_EVENT_TIMER RTC_INT_TIMER_EN`
Event of timer overflow for RTC callback.
- `#define RTC_EVENT_ALARM RTC_INT_ALARM_EN`
Event of alarm for RTC callback.
- `#define RTC_EVENT_PRESCALER RTC_INT_PRESCALER_EN`
Event of RTC prescaler overflow for RTC callback.
- `#define RTC_EVENT_INVALID RTC_INT_INVALID_EN`
Event of RTC invalid interrupt for RTC callback.

Typedefs

- `typedef void(* Rtc_Hal_CallbackType) (uint32 Event)`
rtc callback function prototype.

Enumerations

- `enum Rtc_ClockSourceType {`
`RTC_CLOCK_HSE = 0U, RTC_CLOCK_VHSI, RTC_CLOCK_LSI32K, RTC_CLOCK_CLKIN,`
`RTC_CLOCK_INVALID }`
RTC Clock source type.
- `enum Rtc_ClockOutputType {` `RTC_OUTPUT_DISABLED = 0U, RTC_OUTPUT_CLOCK, RTC_OUTPUT_PRESCALER }`
RTC output type.

Functions

- Hal_StatusType [Rtc_Hal_Init](#) ([Rtc_ClockSourceType](#) Clk)
Initialize rtc module.
- Hal_StatusType [Rtc_Hal_DeInit](#) (void)
Deinitialize rtc module.
- Hal_StatusType [Rtc_Hal_Start](#) (uint32 Unit)
Start RTC.
- Hal_StatusType [Rtc_Hal_Stop](#) (void)
Stop RTC.
- Hal_StatusType [Rtc_Hal_Pause](#) (void)
Pause RTC.
- uint32 [Rtc_Hal_GetCurrentValue](#) (void)
Get RTC real time value.
- Hal_StatusType [Rtc_Hal_SetAlarm](#) (uint32 Value)
Set alarm count.
- void [Rtc_Hal_SetCurrentValue](#) (uint32 Value)
Set RTC real time counter.
- void [Rtc_Hal_InstallCallback](#) (const [Rtc_Hal_CallbackType](#) Func)
Install counter overflow interrupt callback.
- void [Rtc_Hal_EnableInterrupt](#) (uint32 InterruptBits)
Enable or disable rtc interrupt.
- Hal_StatusType [Rtc_Hal_SetConfig](#) ([Rtc_ClockOutputType](#) OutputCfg)
Configure RTC output.
- uint32 [Rtc_Hal_GetWorkFreq](#) (void)
Get rtc work frequency.
- uint32 [Rtc_Hal_GetPrescalerValue](#) (void)
Get prescaler value.

2.4.1 Detailed Description

rtc hal define.

2.4.2 Macro Definition Documentation

2.4.2.1 RTC_EVENT_ALARM

```
#define RTC_EVENT_ALARM RTC\_INT\_ALARM\_EN
```

Event of alarm for RTC callback.

Definition at line 74 of file Rtc_Hal.h.

2.4.2.2 RTC_EVENT_INVALID

```
#define RTC_EVENT_INVALID RTC_INT_INVALID_EN
```

Event of RTC invalid interrupt for RTC callback.

Definition at line 80 of file Rtc_Hal.h.

2.4.2.3 RTC_EVENT_PRESCALER

```
#define RTC_EVENT_PRESCALER RTC_INT_PRESCALER_EN
```

Event of RTC prescaler overflow for RTC callback.

Definition at line 77 of file Rtc_Hal.h.

2.4.2.4 RTC_EVENT_TIMER

```
#define RTC_EVENT_TIMER RTC_INT_TIMER_EN
```

Event of timer overflow for RTC callback.

Definition at line 71 of file Rtc_Hal.h.

2.4.2.5 RTC_INT_ALARM_EN

```
#define RTC_INT_ALARM_EN 0x02U
```

Enable flag(bit) of RTC alarm interrupt.

Definition at line 59 of file Rtc_Hal.h.

2.4.2.6 RTC_INT_ALL_EN

```
#define RTC_INT_ALL_EN 0x0FU
```

Enable flags of RTC all interrupt.

Definition at line 68 of file Rtc_Hal.h.

2.4.2.7 RTC_INT_INVALID_EN

```
#define RTC_INT_INVALID_EN 0x08U
```

Enable flag of invalid interrupt.

Definition at line 65 of file Rtc_Hal.h.

2.4.2.8 RTC_INT_PRESCALER_EN

```
#define RTC_INT_PRESCALER_EN 0x04U
```

Enable flag of RTC prescaler overflow interrupt.

Definition at line 62 of file Rtc_Hal.h.

2.4.2.9 RTC_INT_TIMER_EN

```
#define RTC_INT_TIMER_EN 0x01U
```

Enable flag of RTC timer overflow interrupt.

Definition at line 56 of file Rtc_Hal.h.

2.4.3 Typedef Documentation

2.4.3.1 Rtc_Hal_CallbackType

```
typedef void(* Rtc_Hal_CallbackType) (uint32 Event)
```

rtc callback function prototype.

Parameters

in	<i>Event</i>	Event combinations (bitwise OR) of RTC_EVENT_TIMER/RTC_EVENT_ALARM/RTC_EVENT_PRESCALER.
----	--------------	--

Returns

void

Definition at line 116 of file Rtc_Hal.h.

2.4.4 Enumeration Type Documentation

2.4.4.1 Rtc_ClockOutputType

enum [Rtc_ClockOutputType](#)

RTC output type.

Enumerator

RTC_OUTPUT_DISABLED	RTC clock output is enabled.
RTC_OUTPUT_CLOCK	RTC clock output is enabled.
RTC_OUTPUT_PRESCALER	RTC prescaler output is enabled

Definition at line 102 of file Rtc_Hal.h.

2.4.4.2 Rtc_ClockSourceType

enum [Rtc_ClockSourceType](#)

RTC Clock source type.

Enumerator

RTC_CLOCK_HSE	RTC uses HSE DIV1 as clock source.
RTC_CLOCK_VHSI	RTC uses VHSI DIV1 as clock source.
RTC_CLOCK_LSI32K	RTC uses LSI32K as clock source.
RTC_CLOCK_CLKIN	RTC uses RTC_CLKIN as clock source.
RTC_CLOCK_INVALID	Invalid clock source.

Definition at line 86 of file Rtc_Hal.h.

2.4.5 Function Documentation

2.4.5.1 Rtc_Hal_DeInit()

```
Hal_StatusType Rtc_Hal_DeInit (  
    void )
```

Deinitialize rtc module.

Note

Function ID: DES_GPT_API_602

Returns

Hal_StatusType: STATUS_SUCCESS: success; STATUS_ERROR: RTC isn't initialized or is started.

Definition at line 168 of file Rtc_Hal.c.

2.4.5.2 Rtc_Hal_EnableInterrupt()

```
void Rtc_Hal_EnableInterrupt (
    uint32 InterruptBits )
```

Enable or disable rtc interrupt.

Note

Function ID: DES_GPT_API_608

Parameters

in	InterruptBits	RTC interrupt config. Bit 0: RTC Timer overflow interrupt enable; Bit 1: Alarm interrupt enable; Bit 2: Prescaler interrupt enable.
----	---------------	---

Returns

void

Definition at line 296 of file Rtc_Hal.c.

2.4.5.3 Rtc_Hal_GetCurrentValue()

```
uint32 Rtc_Hal_GetCurrentValue (
    void )
```

Get RTC real time value.

Note

Function ID: DES_GPT_API_605

Returns

uint32: Value of RTC timer register.

Definition at line 256 of file Rtc_Hal.c.

2.4.5.4 Rtc_Hal_GetPrescalerValue()

```
uint32 Rtc_Hal_GetPrescalerValue (
    void )
```

Get prescaler value.

Note

Function ID: DES_GPT_API_611

Returns

uint32: Value of RTC prescaler register.

Definition at line 352 of file Rtc_Hal.c.

2.4.5.5 Rtc_Hal_GetWorkFreq()

```
uint32 Rtc_Hal_GetWorkFreq (
    void )
```

Get rtc work frequency.

Note

Function ID: DES_GPT_API_610

Returns

uint32: work frequency

Definition at line 327 of file Rtc_Hal.c.

2.4.5.6 Rtc_Hal_Init()

```
Hal_StatusType Rtc_Hal_Init (
    Rtc_ClockSourceType Clk )
```

Initialize rtc module.

Note

Function ID: DES_GPT_API_601

Parameters

<i>in</i>	<i>Clk</i>	Specify RTC work clock source. RTC_CLOCK_INVALID: doesn't set clock source.
-----------	------------	---

Returns

Hal_StatusType: Initialize success or not, the range is the STATUS_SUCCESS STATUS_ERROR

Definition at line 121 of file Rtc_Hal.c.

2.4.5.7 Rtc_Hal_InstallCallback()

```
void Rtc_Hal_InstallCallback (
    const Rtc_Hal_CallbackType Func )
```

Install counter overflow interrupt callback.

Note

Function ID: DES_GPT_API_607

Parameters

<i>in</i>	<i>Func</i>	the callback function
-----------	-------------	-----------------------

Returns

void

Definition at line 291 of file Rtc_Hal.c.

2.4.5.8 Rtc_Hal_Pause()

```
Hal_StatusType Rtc_Hal_Pause (
    void )
```

Pause RTC.

Precondition

Pre-condition: RTC is started.

Note

Function ID: DES_GPT_API_613

Returns

Hal_StatusType: STATUS_SUCCESS: success; STATUS_ERROR: RTC isn't started.

Definition at line 240 of file Rtc_Hal.c.

2.4.5.9 Rtc_Hal_SetAlarm()

```
Hal_StatusType Rtc_Hal_SetAlarm (
    uint32 Value )
```

Set alarm count.

Note

Function ID: DES_GPT_API_606

Parameters

in	Value	Count for alarm.
----	-------	------------------

Returns

Hal_StatusType: Set success or not, the range is the STATUS_SUCCESS STATUS_ERROR

Definition at line 276 of file Rtc_Hal.c.

2.4.5.10 Rtc_Hal_SetConfig()

```
Hal_StatusType Rtc_Hal_SetConfig (
    Rtc_ClockOutputType OutputCfg )
```

Configure RTC output.

Note

Function ID: DES_GPT_API_609

Parameters

in	OutputCfg	: RTC output configuration. 0 : Disabled, 1: RTC clock output, 2: RTC prescaler output.
----	-----------	---

Returns

Hal_StatusType: STATUS_SUCCESS: success;

Definition at line 312 of file Rtc_Hal.c.

2.4.5.11 Rtc_Hal_SetCurrentValue()

```
void Rtc_Hal_SetCurrentValue (
    uint32 Value )
```

Set RTC real time counter.

Precondition

Pre-condition: RTC isn't started.

Note

Function ID: DES_GPT_API_612

Parameters

<i>in</i>	<i>Value</i>	Value of RTC real timer.
-----------	--------------	--------------------------

Returns

void

Definition at line 267 of file Rtc_Hal.c.

2.4.5.12 Rtc_Hal_Start()

```
Hal_StatusType Rtc_Hal_Start (  
    uint32 Unit )
```

Start RTC.

Note

Function ID: DES_GPT_API_603

Parameters

<i>in</i>	<i>Unit</i>	Value for RTC prescaler. And it is the unit of RTC real time register.
-----------	-------------	--

Returns

Hal_StatusType: STATUS_SUCCESS: success; STATUS_ERROR: RTC isn't initialized or has been started.

Definition at line 181 of file Rtc_Hal.c.

2.4.5.13 Rtc_Hal_Stop()

```
Hal_StatusType Rtc_Hal_Stop (  
    void )
```

Stop RTC.

Note

Function ID: DES_GPT_API_604

Returns

Hal_StatusType: STATUS_SUCCESS: success; STATUS_ERROR: RTC isn't initialized or isn't started.

Definition at line 216 of file Rtc_Hal.c.

Index

AC784xx_API_Reference_Manual_RTC.pdf, [2](#)

AC784xx_Rtc_Reg.h, [2](#)

Rtc_Reg_Enable, [3](#)

Rtc_Reg_ReadAlarm, [3](#)

Rtc_Reg_ReadCTRL, [3](#)

Rtc_Reg_ReadLR, [4](#)

Rtc_Reg_ReadPSC, [4](#)

Rtc_Reg_ReadPSR, [4](#)

Rtc_Reg_ReadSR, [5](#)

Rtc_Reg_ReadTC, [5](#)

Rtc_Reg_Reset, [5](#)

Rtc_Reg_WriteAlarm, [6](#)

Rtc_Reg_WriteCTRL, [6](#)

Rtc_Reg_WriteLR, [6](#)

Rtc_Reg_WritePSR, [7](#)

Rtc_Reg_WriteSR, [7](#)

Rtc_Reg_WriteTC, [8](#)

CLEAR_TIF_VALUE

Rtc_Hal.c, [9](#)

RTC_EVENT_ALARM

Rtc_Hal.h, [17](#)

RTC_EVENT_INVALID

Rtc_Hal.h, [17](#)

RTC_EVENT_PRESCALER

Rtc_Hal.h, [18](#)

RTC_EVENT_TIMER

Rtc_Hal.h, [18](#)

RTC_INT_ALARM_EN

Rtc_Hal.h, [18](#)

RTC_INT_ALL_EN

Rtc_Hal.h, [18](#)

RTC_INT_INVALID_EN

Rtc_Hal.h, [18](#)

RTC_INT_PRESCALER_EN

Rtc_Hal.h, [19](#)

RTC_INT_TIMER_EN

Rtc_Hal.h, [19](#)

RTC_IRQHandler

Rtc_Hal.c, [15](#)

Rtc_ClockOutputType

Rtc_Hal.h, [20](#)

Rtc_ClockSourceType

Rtc_Hal.h, [20](#)

Rtc_Hal.c, [8](#)

CLEAR_TIF_VALUE, [9](#)

RTC_IRQHandler, [15](#)

Rtc_Hal_DeInit, [10](#)

Rtc_Hal_EnableInterrupt, [10](#)

Rtc_Hal_GetCurrentValue, [11](#)

Rtc_Hal_GetPrescalerValue, [11](#)

Rtc_Hal_GetWorkFreq, [11](#)

Rtc_Hal_Init, [12](#)

Rtc_Hal_InstallCallback, [12](#)

Rtc_Hal_Pause, [13](#)

Rtc_Hal_SetAlarm, [13](#)

Rtc_Hal_SetConfig, [14](#)

Rtc_Hal_SetCurrentValue, [14](#)

Rtc_Hal_Start, [15](#)

Rtc_Hal_Stop, [15](#)

Rtc_StateType, [9](#)

Rtc_Hal.h, [16](#)

RTC_EVENT_ALARM, [17](#)

RTC_EVENT_INVALID, [17](#)

RTC_EVENT_PRESCALER, [18](#)

RTC_EVENT_TIMER, [18](#)

RTC_INT_ALARM_EN, [18](#)

RTC_INT_ALL_EN, [18](#)

RTC_INT_INVALID_EN, [18](#)

RTC_INT_PRESCALER_EN, [19](#)

RTC_INT_TIMER_EN, [19](#)

Rtc_ClockOutputType, [20](#)

Rtc_ClockSourceType, [20](#)

Rtc_Hal_CallbackType, [19](#)

Rtc_Hal_DeInit, [20](#)

Rtc_Hal_EnableInterrupt, [21](#)

Rtc_Hal_GetCurrentValue, [21](#)

Rtc_Hal_GetPrescalerValue, [21](#)

Rtc_Hal_GetWorkFreq, [22](#)

Rtc_Hal_Init, [22](#)

Rtc_Hal_InstallCallback, [23](#)

Rtc_Hal_Pause, [23](#)

Rtc_Hal_SetAlarm, [23](#)

Rtc_Hal_SetConfig, [24](#)

Rtc_Hal_SetCurrentValue, [24](#)

Rtc_Hal_Start, [25](#)

Rtc_Hal_Stop, [25](#)

Rtc_Hal_CallbackType

Rtc_Hal.h, [19](#)

Rtc_Hal_DeInit

Rtc_Hal.c, [10](#)

Rtc_Hal.h, [20](#)

Rtc_Hal_EnableInterrupt

Rtc_Hal.c, [10](#)

Rtc_Hal.h, [21](#)

Rtc_Hal_GetCurrentValue

Rtc_Hal.c, [11](#)

Rtc_Hal.h, [21](#)

Rtc_Hal_GetPrescalerValue

Rtc_Hal.c, [11](#)

Rtc_Hal.h, [21](#)

Rtc_Hal_GetWorkFreq

Rtc_Hal.c, [11](#)

- Rtc_Hal.h, [22](#)
- Rtc_Hal_Init
 - Rtc_Hal.c, [12](#)
 - Rtc_Hal.h, [22](#)
- Rtc_Hal_InstallCallback
 - Rtc_Hal.c, [12](#)
 - Rtc_Hal.h, [23](#)
- Rtc_Hal_Pause
 - Rtc_Hal.c, [13](#)
 - Rtc_Hal.h, [23](#)
- Rtc_Hal_SetAlarm
 - Rtc_Hal.c, [13](#)
 - Rtc_Hal.h, [23](#)
- Rtc_Hal_SetConfig
 - Rtc_Hal.c, [14](#)
 - Rtc_Hal.h, [24](#)
- Rtc_Hal_SetCurrentValue
 - Rtc_Hal.c, [14](#)
 - Rtc_Hal.h, [24](#)
- Rtc_Hal_Start
 - Rtc_Hal.c, [15](#)
 - Rtc_Hal.h, [25](#)
- Rtc_Hal_Stop
 - Rtc_Hal.c, [15](#)
 - Rtc_Hal.h, [25](#)
- Rtc_Reg_Enable
 - AC784xx_Rtc_Reg.h, [3](#)
- Rtc_Reg_ReadAlarm
 - AC784xx_Rtc_Reg.h, [3](#)
- Rtc_Reg_ReadCTRL
 - AC784xx_Rtc_Reg.h, [3](#)
- Rtc_Reg_ReadLR
 - AC784xx_Rtc_Reg.h, [4](#)
- Rtc_Reg_ReadPSC
 - AC784xx_Rtc_Reg.h, [4](#)
- Rtc_Reg_ReadPSR
 - AC784xx_Rtc_Reg.h, [4](#)
- Rtc_Reg_ReadSR
 - AC784xx_Rtc_Reg.h, [5](#)
- Rtc_Reg_ReadTC
 - AC784xx_Rtc_Reg.h, [5](#)
- Rtc_Reg_Reset
 - AC784xx_Rtc_Reg.h, [5](#)
- Rtc_Reg_WriteAlarm
 - AC784xx_Rtc_Reg.h, [6](#)
- Rtc_Reg_WriteCTRL
 - AC784xx_Rtc_Reg.h, [6](#)
- Rtc_Reg_WriteLR
 - AC784xx_Rtc_Reg.h, [6](#)
- Rtc_Reg_WritePSR
 - AC784xx_Rtc_Reg.h, [7](#)
- Rtc_Reg_WriteSR
 - AC784xx_Rtc_Reg.h, [7](#)
- Rtc_Reg_WriteTC
 - AC784xx_Rtc_Reg.h, [8](#)
- Rtc_StateType
 - Rtc_Hal.c, [9](#)