# RTC

## 概述

RTC是一个实时日历模块，可以用来实现时钟、闹表、定时功能。时钟源可以选择IMOSC/4， ISOSC，ESOSC， EMOSC/4。选择的时候，请考虑振荡器的容差会对时间的准确程度产生直接的影响。

RTC的很多寄存器不会受到复位的影响。

## API列表

Table20‑1RTC CSI接口函数

|  |  |  |
| --- | --- | --- |
| API | 说明 | 函数位置 |
| csi\_rtc\_init | 初始化RTC，主要是时钟源和RTC计时方式（24H/12H）。 | rtc.c |
| csi\_rtc\_rb\_enable | 使能/禁止回读功能。 |
| csi\_rtc\_change\_fmt | 改变RTC的计时方式（24H/12H）。 |
| csi\_rtc\_set\_time | 设置时间。 |
| csi\_rtc\_get\_time | 读取当前时间。 |
| csi\_rtc\_set\_alarm | 设置闹表时间。 |
| csi\_rtc\_get\_alarm\_remaining\_time | 获得距离闹表的倒计时时间。单位:s |
| csi\_rtc\_cancel\_alarm | 取消闹表。 |
| csi\_rtc\_start\_as\_timer | 将RTC当做一个定时器使用。 |
| csi\_rtc\_int\_enable | 使能RTC的某个中断。 |
| csi\_rtc\_start | 开始运行。 |
| csi\_rtc\_stop | 停止运行。 |
| csi\_rtc\_set\_evtrg | 设置RTC向外输出的触发事件。 |

## API详细描述

### csi\_rtc\_init

void csi\_rtc\_init(csp\_rtc\_t \*ptRtc, csi\_rtc\_config\_t \*tConfig)

#### 功能描述

用于对RTC进行初始化，包括时钟源、计时模式（12小时制/24小时制）。

#### 参数说明

|  |  |  |
| --- | --- | --- |
| 参数 | 说明 | 位置 |
| ptRtc | 指向RTC控制寄存器结构体的指针，用于进行寄存器操作。 | csp\_rtc.h |
| tConfig | 结构体变量。  typedef struct  {      uint8\_t     byClkSrc;   /// RTC\_ISOSC/RTC\_EMOSC/RTC\_IMOSC\_DIV4/RTC\_EMOSC\_DIV4      uint8\_t     byFmt;      //  RTC\_24FMT/RTC\_12FMT  } csi\_rtc\_config\_t; | rtc.h |

### csi\_rtc\_rb\_enable

void csi\_rtc\_rb\_enable(csp\_rtc\_t \*ptRtc, bool bEnable)

#### 功能描述

RTC的回读功能是可以开关的。打开后，可以通过RTC\_TIMR和RTC\_DATR读取当前的实时时间。关闭就不支持回读，对功耗敏感的应用或时间段可以尝试关闭回读功能。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

bEnable：bool型参数，可选值ENABLE/DISABLE，或1/0。

1. 返回值：无

### csi\_rtc\_change\_fmt

void csi\_rtc\_change\_fmt(csp\_rtc\_t \*ptRtc, rtc\_fmt\_e eFmt)

#### 功能描述

改变RTC时间模式（12H <-> 24H）。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

eFmt：枚举型参数，可选值RTC\_24FMT / RTC\_12FMT。

1. 返回值：无

### csi\_rtc\_set\_time

csi\_error\_t csi\_rtc\_set\_time(csp\_rtc\_t \*ptRtc, csi\_rtc\_time\_t \*rtctime)

#### 功能描述

设置改变RTC的时间。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

csi\_rtc\_time\_t：时间结构体参数。

1. 返回值：csi\_error\_t型数据。当写入的时间超出允许的范围时，会返回CSI\_ERROR。
2. 参数/返回值说明表

|  |  |  |
| --- | --- | --- |
| 参数/返回值 | 说明 | 位置 |
| ptRtc | RTC寄存器结构体指针，用于进行寄存器操作。 | csp\_rtc.h |
| rtctime | 时间结构体参数。  typedef struct {      int iSec; ///< Second. [0-59]  int iMin; ///< Minute. [0-59]  int iHour; ///< Hour. [0-23]  int iMday; ///< Day. [1-31]  int iMon; ///< Month. [1-12]  int iYear; ///< Year-1900. [70- ] !NOTE:100=2000  int iWday; /// weekday [1-7]  int iYday; ///< Days in year.[0-365] !NOTE:January 1st = 0  int iIsdst; /// Non-0 if daylight savings time is in effect  int iPm; ///< PM. [0/1]  } csi\_rtc\_time\_t; |  |

### csi\_rtc\_get\_time

void csi\_rtc\_get\_time(csp\_rtc\_t \*ptRtc, csi\_rtc\_time\_t \*rtctime)

#### 功能描述

读取RTC当前的时间。如果回读功能没有开启，则一直读回设置值。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

csi\_rtc\_time\_t：时间结构体参数。同[参数说明](#_参数说明)。

1. 返回值：无。

### csi\_rtc\_set\_alarm

csi\_error\_t csi\_rtc\_set\_alarm(csp\_rtc\_t \*ptRtc, csi\_rtc\_alarm\_e eAlm, csi\_rtc\_time\_t \*ptAlmTime)

#### 功能描述

用于设置闹表的时间和闹表模式。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

eAlm：APT32F110x可以使用闹表AlarmA、AlarmB，所以可以传入RTC\_ALMA、RTC\_ALMB。

ptAlmTime：时间结构体参数。同[参数说明](#_参数说明)。

1. 返回值：csi\_error\_t型枚举变量。
2. 参数/返回值说明表

|  |  |  |
| --- | --- | --- |
| 参数/返回值 | 说明 | 位置 |
| ptRtc | RTC寄存器结构体指针，用于进行寄存器操作。 | csp\_rtc.h |
| eAlm | csi\_rtc\_alarm\_e枚举类型变量，APT32F110x可以使用闹表AlarmA、AlarmB，所以可以传入RTC\_ALMA、RTC\_ALMB。 | rtc.h |
| ptAlmTime | csi\_rtc\_time\_t结构变量。  如果设置结构体成员，如tm\_sec= 0xff，表示屏蔽该位的比较。 | rtc.h |

### csi\_rtc\_get\_alarm\_remaining\_time

uint32\_t csi\_rtc\_get\_alarm\_remaining\_time(csp\_rtc\_t \*ptRtc, csi\_rtc\_alarm\_e eAlm)

#### 功能描述

获得距离闹表的倒计时时间。单位:s。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

eAlm：APT32F110x可以使用闹表AlarmA、AlarmB，所以可以传入RTC\_ALMA、RTC\_ALMB。

1. 返回值：uint32\_t型变量。如果闹表时间距离当前时间超过一年，当前驱动会返回CSI\_UNSUPPORTED。

### csi\_rtc\_cancel\_alarm

void csi\_rtc\_cancel\_alarm(csp\_rtc\_t \*ptRtc, csi\_rtc\_alarm\_e eAlm)

#### 功能描述

关闭闹表，关闭闹表中断。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

eAlm：APT32F110x可以使用闹表AlarmA、AlarmB，所以可以传入RTC\_ALMA、RTC\_ALMB。

1. 返回值：无。

### csi\_rtc\_start\_as\_timer

void csi\_rtc\_start\_as\_timer(csp\_rtc\_t \*ptRtc, csi\_rtc\_timer\_e ePrd)

#### 功能描述

将RTC当做定时器来使用，但是定时间隔只能在几个选项中选择。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

ePrd：csi\_rtc\_timer\_e枚举变量，定时间隔。

1. 返回值：无。
2. 参数/返回值说明表

|  |  |  |
| --- | --- | --- |
| 参数/返回值 | 说明 | 位置 |
| ptRtc | RTC寄存器结构体指针，用于进行寄存器操作。 | csp\_rtc.h |
| ePrd | csi\_rtc\_timer\_e枚举变量。  typedef enum {      RTC\_TIMER\_DIS = 0,      RTC\_TIMER\_0\_5S,      RTC\_TIMER\_1S,      RTC\_TIMER\_1MIN,      RTC\_TIMER\_1H,      RTC\_TIMER\_1DAY,      RTC\_TIMER\_1MON  }csi\_rtc\_timer\_e; | rtc.h |

### csi\_rtc\_int\_enable

void csi\_rtc\_int\_enable(csp\_rtc\_t \*ptRtc, csi\_rtc\_intsrc\_e eIntSrc, bool bEnable)

#### 功能描述

中断使能。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

eIntSrc：csi\_rtc\_intsrc\_e枚举变量。

bEnable：bool型变量

1. 返回值：无。
2. 参数/返回值说明表

|  |  |  |
| --- | --- | --- |
| 参数/返回值 | 说明 | 位置 |
| ptRtc | RTC寄存器结构体指针，用于进行寄存器操作。 | rtc.h |
| eIntSrc | rtc\_int\_e枚举变量。  typedef enum{  RTC\_INTSRC\_NONE = 0x00, //无中断      RTC\_INTSRC\_ALMA = 0x01, //AlarmA中断      RTC\_INTSRC\_ALMB = 0x02, //AlarmB中断      RTC\_INTSRC\_CPRD = 0x04,  //CPRD定时中断      RTC\_INTSRC\_TRGEV0 = 0x08, //输出触发事件1中断      RTC\_INTSRC\_TRGEV1= 0x10  //输出触发事件2中断  }csi\_rtc\_intsrc\_e; |
| bEnable | bool型变量，ENABLE/DISABLE | common.h |

### csi\_rtc\_start

void csi\_rtc\_start(csp\_rtc\_t \*ptRtc)

#### 功能描述

RTC开始工作。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

1. 返回值：无。

### csi\_rtc\_stop

void csi\_rtc\_stop(csp\_rtc\_t \*ptRtc)

#### 功能描述

RTC停止工作。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

1. 返回值：无。

### csi\_rtc\_set\_evtrg

csi\_error\_t csi\_rtc\_set\_evtrg(csp\_rtc\_t \*ptRtc, csi\_rtc\_trgsel\_e eEvtrg, csi\_rtc\_trgsrc\_e eTrgSrc, uint8\_t byTrgPrd)

#### 功能描述

设置RTC输出触发事件，包括事件通道、事件源、几次事件后生成对外触发事件。

#### 参数说明

1. 参数

ptRtc：RTC寄存器结构体指针，用于进行寄存器操作。

eEvtrg：指定事件通道号，RTC\_TRGSEL0或RTC\_TRGSEL1。

eTrgSrc：指定事件源。

byTrgPrd：指定几次事件后生成对外触发事件，

1. 返回值：

CSI\_OK：配置成功。

CSI\_ERROR：配置失败。

1. 参数/返回值说明表

|  |  |  |
| --- | --- | --- |
| 参数/返回值 | 说明 | 位置 |
| ptRtc | RTC寄存器结构体指针，用于进行寄存器操作。 | csp\_rtc.h |
| eEvtrg | csi\_rtc\_trgsel\_e枚举型变量。  typedef enum{  RTC\_TRGSEL0 = 0,  RTC\_TRGSEL1  }csi\_rtc\_trgsel\_e; | rtc.h |
| eTrgSrc | csi\_rtc\_trgsrc\_e枚举型变量。  typedef enum{      RTC\_TRGOUT\_DIS = 0, //禁止触发输出      RTC\_TRGOUT\_ALRMA,   //AlarmA时间到触发      RTC\_TRGOUT\_ALRMB,   //AlarmB时间到触发      RTC\_TRGOUT\_ALRMA\_ALRMB, //AlarmA或B时间到触发      RTC\_TRGOUT\_CPRD    //CPRD定时事件到触发  }csi\_rtc\_trgsrc\_e; | rtc.h |