目录

第13章 Cortex-M4-实时时钟

13.1 实时时钟概述

13.1.1 实时时钟介绍

英文缩写：RTC。显示年、月、日、时、分、秒、星期,自动计算闰年，能够区分每个月的天数。

RTC 特点：能从 RTC 获取到具体的日期时间，断掉后再开机时间仍然准确。

RTC 模块分为两种，一种集成在芯片内部，另外一种是外接 RTC 芯片。

13.1.2 常用的实时时钟芯片

常见实时时钟芯片：DS1302、DS1307、PCF8563 等。
显示年、月、日、时、分、秒、星期，自动计算闰年，能够区分每个月的天数。

13.2 stm32 内部实时时钟介绍

13.2.1 STM32 内部实时时钟特点

BCD 码 → 二进制码十进制数 Bin Change Dec

实时时钟 (RTC) 是一个独立的 BCD 定时器/计数器。 RTC 提供一个日历时钟、两个可编程闹钟中断，以及一个具有中断功能的周期性可编程唤醒标志。 RTC 还包含用于管理低功耗模式的自动唤醒单元。

两个 32 位寄存器包含二进码十进数格式 (BCD) 的秒、分钟、小时（12 或 24 小时制）、星期几、日期、月份和年份。此外，还可提供二进制格式的亚秒值。

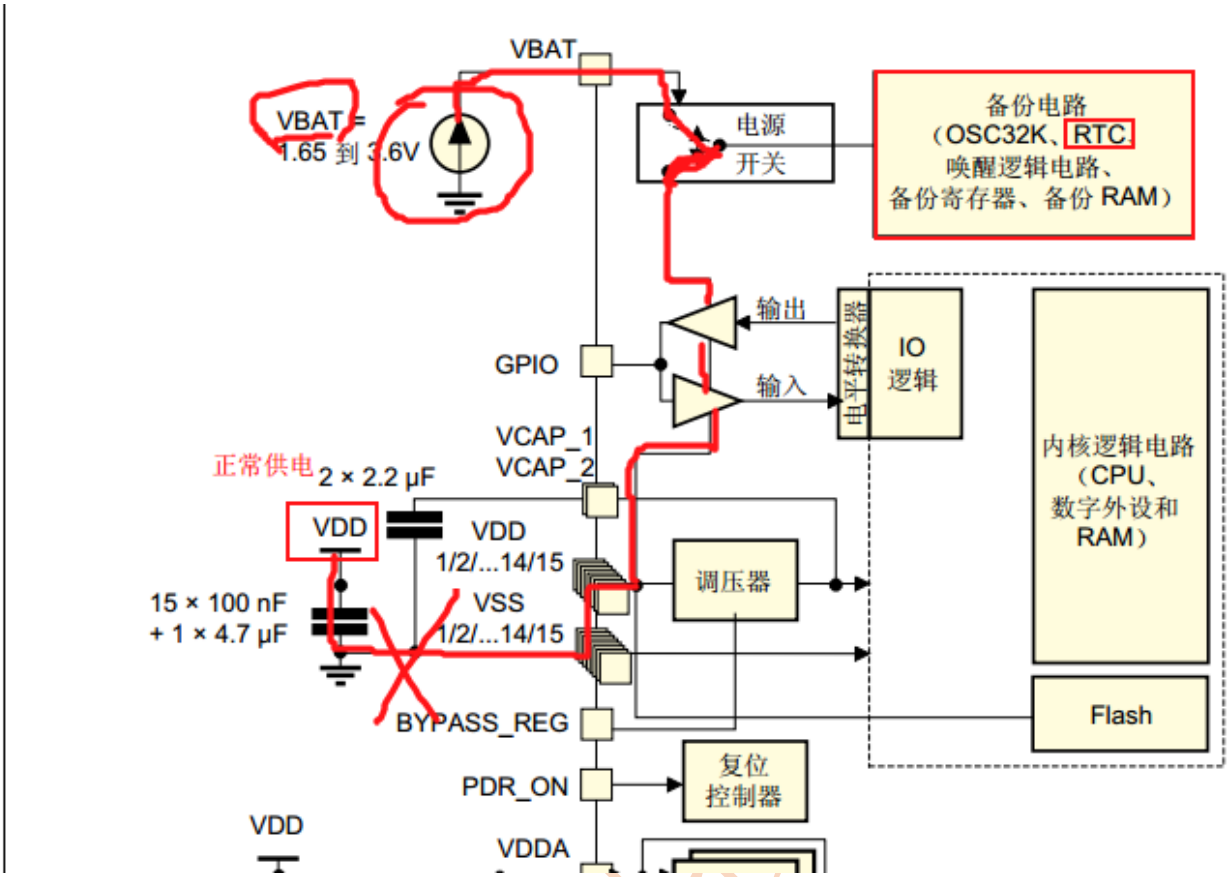
系统可以自动将月份的天数补偿为 28、 29（闰年）、 30 和 31 天。并且还可以进行夏令时补偿。

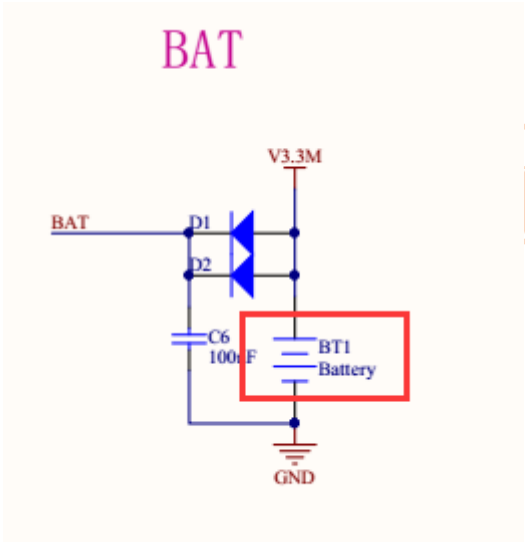
其它 32 位寄存器还包含可编程的闹钟亚秒、 秒、分钟、小时、星期几和日期。

此外，还可以使用数字校准功能对晶振精度的偏差进行补偿。

上电复位后，所有 RTC 寄存器都会受到保护，以防止可能的非正常写访问。无论器件状态如何（运行模式、低功耗模式或处于复位状态），只要电源电压保持在工作范围内， RTC 便不会停止工作。

13.2.2 RTC 的电源部分





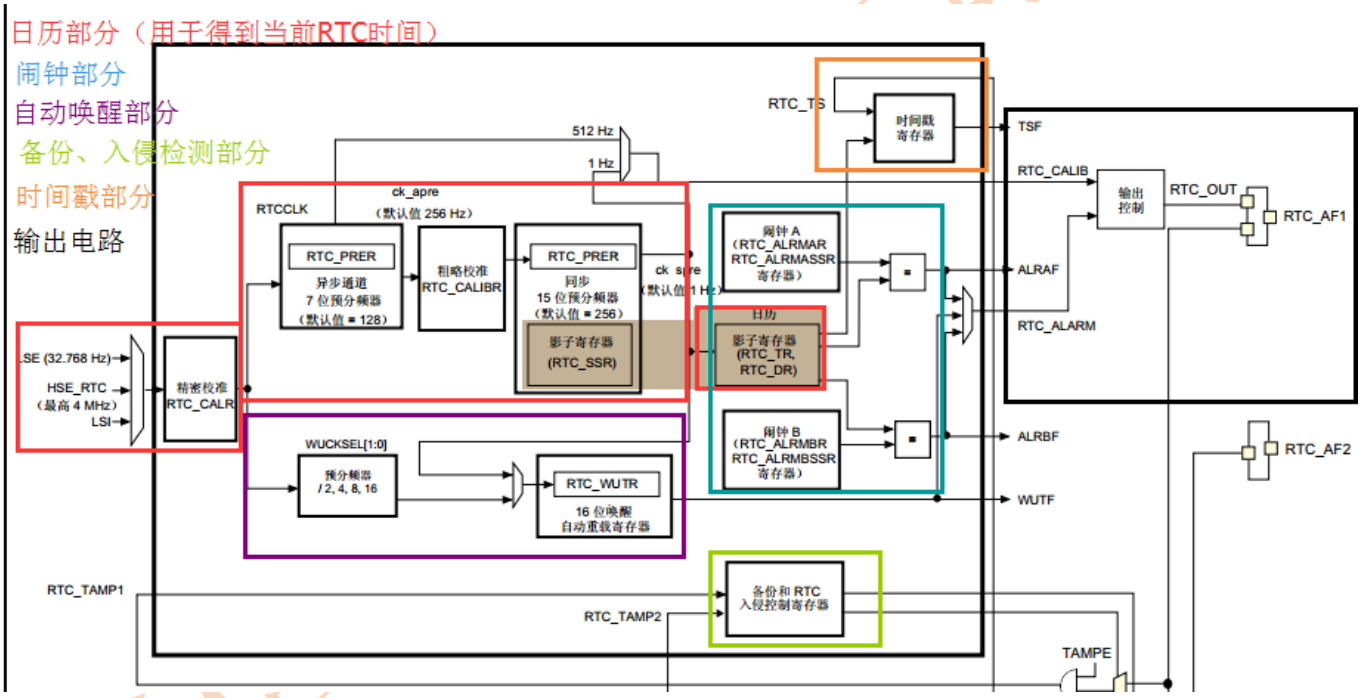
13.2.3 STM32 内部实时时钟的功能介绍

RTC 单元的主要特性如下

- 包含亚秒、秒、分钟、小时（12/24 小时制）、星期几、日期、月份和年份的日历。
- 软件可编程的夏令时补偿。
- 两个具有中断功能的可编程闹钟。可通过任意日历字段的组合驱动闹钟。
- 自动唤醒单元，可周期性地生成标志以触发自动唤醒中断。
- 参考时钟检测：可使用更加精确的第二时钟源（50 Hz 或 60 Hz）来提高日历的精确度。
- 利用亚秒级移位特性与外部时钟实现精确同步。

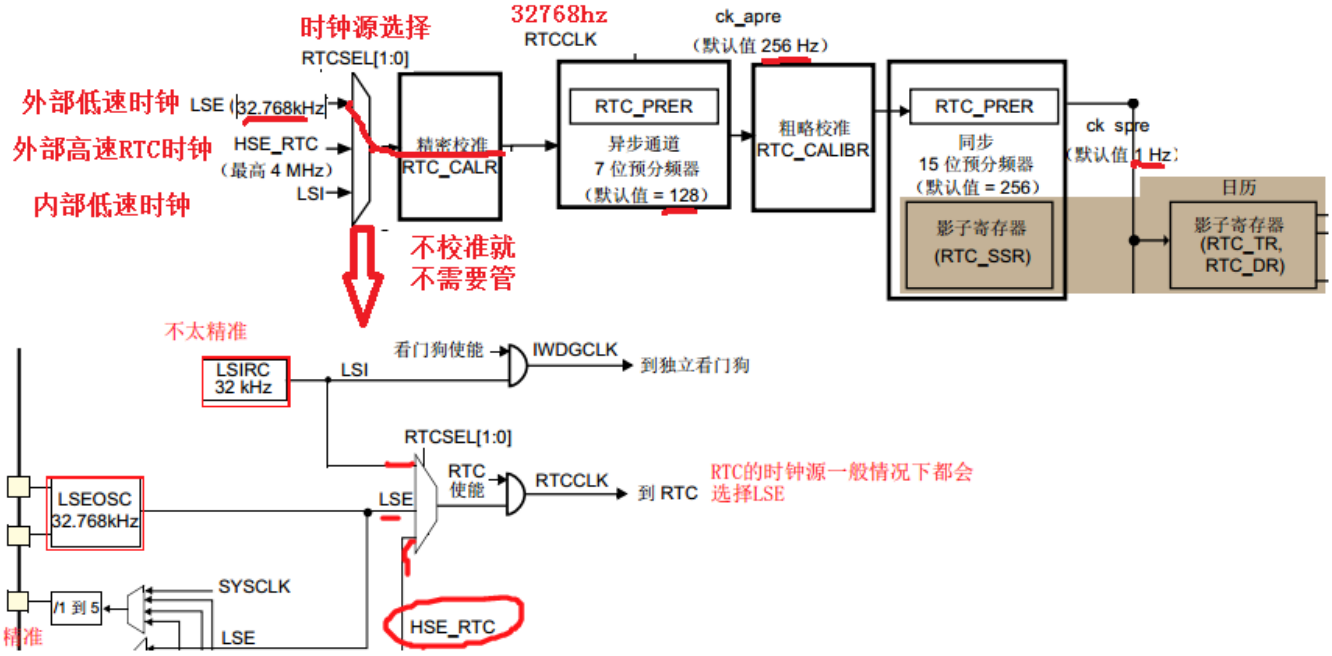
● 可屏蔽中断/事件：
— 闹钟 A
— 闹钟 B
— 唤醒中断
— 时间戳
— 入侵检测
● 数字校准电路（周期性计数器调整）
— 精度为 5 ppm
— 精度为 0.95 ppm，在数秒钟的校准窗口中获得
● 用于事件保存的时间戳功能（1 个事件）
● 入侵检测：
— 2 个带可配置过滤器和内部上拉的入侵事件
● 20 个备份寄存器（80 字节）。发生入侵检测事件时，将复位备份寄存器。

13.3 stm32 内部实时时钟框架



13.4 RTC 基本日历功能框架分析

基本日历功能主要就是让日历模块正常工作（1 秒 1 秒地计数），我们从日历时间读取出当前实时时间日期。也就是说，日历的工作频率就是 1HZ。



LSE(32.768Khz)-->RTCCLK(32.768khz)-->异步预分频器(128 分频)-->ck_apre(256hz)---->同步预分频器(256 分频)-->ck_spre(1hz)-->给到日历（1 秒 1 秒地计数）

13.4.1 RTC 初始化注意事项

13.4.1.1 RTC 寄存器写保护

系统复位后，可通过 PWR 电源控制寄存器 (PWR_CR) 的 DBP 位保护 RTC 寄存器以防止非正常的写访问。必须将 DBP 位置 1 才能使能 RTC 寄存器的写访问。

- 访问 RTC 和 RTC 备份寄存器

1. 将 RCC_APB1ENR 寄存器中的 PWREN 位置 1，使能电源接口时钟（分别参见*第 6.3.15 节*和*第 6.3.16 节*了解 STM32F405xx/07xx 和 STM32F415xx/17xx 和 STM32F42xxx 和 STM32F43xxx）
2. 将*用于 STM32F405xx/07xx 和 STM32F415xx/17xx 的 PWR 电源控制寄存器 (PWR_*和*用于STM32F42xxx 和 STM32F43xxx 的 PWR 电源控制寄存器 (PWR_CR)* 中的 位置 1，使能对备份域的访问
3. 选择 RTC 时钟源：参见*第 6.2.8 节：RTC/AWU 时钟*
4. 通过对 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 RTCEN [15] 位进行编程，RTC 时钟

公共聊天
助教聊天

位 8 **DBP**：禁止备份域写保护 (Disable backup domain write protection)

在复位状态下，RCC_BDCR 寄存器、RTC 寄存器（包括备份寄存器）以及 PWR_CSR 寄存器的 BRE 位均受到写访问保护。必须将此位置 1 才能使能对这些寄存器的写访问。

0：禁止对 RTC、RTC 备份寄存器和备份 SRAM 的访问

1：使能对 RTC、RTC 备份寄存器和备份 SRAM 的访问

上电复位后，所有 RTC 寄存器均受到写保护。通过向写保护寄存器 (RTC_WPR) 写入一个密钥来使能对 RTC 寄存器的写操作。

要解锁所有 RTC 寄存器（RTC_ISR[13:8]、 RTC_TAFCR 和 RTC_BKPxR 除外）的写保护，需要执行以下步骤：

1. 将“0xCA”写入 RTC_WPR 寄存器。
2. 将“0x53”写入 RTC_WPR 寄存器。

写入一个错误的关键字会再次激活写保护。

保护机制不受系统复位影响。

取消 RTC 寄存器写保护：
RTC->WPR=0xCA;
RTC->WPR=0x53;

后面就可以对 RTC 所有的寄存器进行写操作

激活 RTC 寄存器写保护:RTC->WPR=0xFF;

13.4.1.2 RTC 进入初始化模式（设置日历寄存器要注意）

要编程包括时间格式和预分频器配置在内的初始时间和日期日历值，需按照以下顺序操作：

1. 将 RTC_ISR 寄存器中的 INIT 位置 1 以进入初始化模式。在此模式下，日历计数器将停止工作并且其值可更新。
   (初始化模式，用于编程时间和日期寄存器（RTC_TR 和 RTC_DR）以及预分频器寄存器(RTC_PRER)。计数器停止计数，当 INIT 被复位后，计数器从新值开始计数。)
2. 轮询 RTC_ISR 寄存器中的 INITF 位。当 INITF 置 1 时进入初始化阶段模式。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）。
3. 要为日历计数器生成 1 Hz 时钟，应首先编程 RTC_PRER 寄存器中的同步预分频系数，然后编程异步预分频系数。即使只需要更改这两个字段中之一，也必须对 RTC_PRER 寄存器执行两次单独的写访问。
4. 在影子寄存器（RTC_TR 和 RTC_DR）中加载初始时间和日期值，然后通过 RTC_CR 寄存器中的 FMT 位配置时间格式（12 或 24 小时制）。(RTC_TR 和 RTC_DR 是影子寄存器)
5. 通过清零 INIT 位退出初始化模式。随后，自动加载实际日历计数器值，在 4 个 RTCCLK 时钟周期后重新开始计数。

   当初始化序列完成之后，日历开始计数。

13.4.1.3 RTC 同步（读取日历值要注意）

每次将日历寄存器中的值复制到 RTC_SSR、 RTC_TR 和 RTC_DR 影子寄存器时， RTC_ISR 寄存器中的 RSF 位都会置 1（日历影子寄存器已同步 ）。每两个 RTCCLK 周期执行一次复制。为确保这 3 个值来自同一时刻点，读取 RTC_SSR 或 RTC_TR 时会锁定高阶日历影子寄存器中的值，直到读取 RTC_DR。为避免软件对日历执行读访问的时间间隔小于 2 个 RTCCLK 周期：第一次读取日历之后必须通过软件将 RSF 清零，并且软件必须等待到 RSF 置 1 之后才可再次读取 RTC_SSR、 RTC_TR 和 RTC_DR 寄存器。

如何读取日历值
清除 RSF
等待 RSF 由硬件置 1（日历影子寄存器已同步）
读取日历值（RTC_TR、RTC_DR）

13.4.1.4 STM32 内部实时时钟寄存器说明

23.6.10 RTC 写保护寄存器 (RTC_WPR)

补充：

BCD 编码格式：二进码十进数格式

用 4 位二进制数表示一个十进制数 0000 - 1001 0-9

用 4 位二进制数表示一个十六进制数

1010 a

1100 c

14:53:20

小时：00010100

分钟：01010011

秒：00100000

RTC 时间寄存器 (RTC_TR)

RTC_TR 是日历时间影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见 *第 576 页的日历初始化和配置*和 *第 577 页的读取日历*。

偏移地址：0x00

上电复位值：0x0000 0000

系统复位：当 BYPSHAD = 0 时为 0x0000 0000；当 BYPSHAD = 1 时不受影响。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PM	HT[1:0]		HU[3:0]			
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MNT[2:0]			MNU[3:0]				Reserved	ST[2:0]			SU[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 22 PM： AM/PM 符号 (AM/PM notation)

0： AM 或 24 小时制

1： PM

位 21:20 HT[1:0]：小时的十位（BCD 格式） (Hour tens in BCD format)

位 16:16 HU[3:0]：小时的个位（BCD 格式） (Hour units in BCD format)

位 15 保留，必须保持复位值。

位 14:12 MNT[2:0]：分钟的十位（BCD 格式） (Minute tens in BCD format)

位 11:8 MNU[3:0]：分钟的个位（BCD 格式） (Minute units in BCD format)

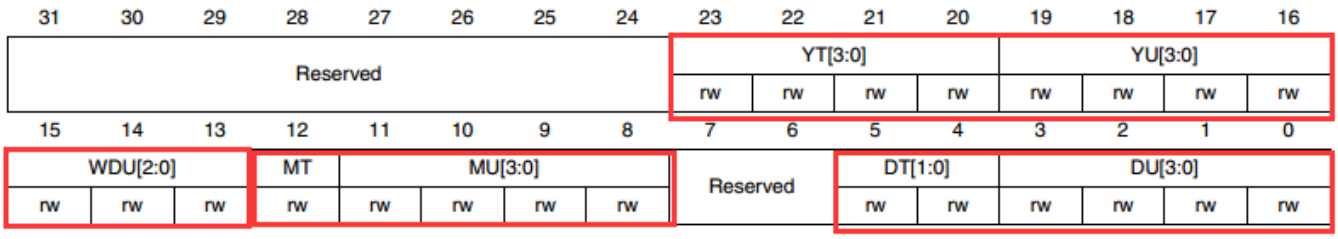
位 7 保留，必须保持复位值。

位 6:4 ST[2:0]：秒的十位（BCD 格式） (Second tens in BCD format)

位 3:0 SU[3:0]：秒的个位（BCD 格式） (Second units in BCD format)

RTC 日期寄存器 (RTC_DR)

RTC_DR 是日历日期影子寄存器。只能在初始化模式下对该寄存器执行写操作



位 23:20 YT[3:0]：年份的十位（BCD 格式） (Year tens in BCD format)

位 19:16 YU[3:0]：年份的个位（BCD 格式） (Year units in BCD format)

位 15:13 WDU[2:0]：星期几的个位 (Week day units)

000：禁止

001：星期一

...

111：星期日

位 12 MT：月份的十位（BCD 格式） (Month tens in BCD format)

位 11:8 MU：月份的个位（BCD 格式） (Month units in BCD format)

位 7:6 保留，必须保持复位值。

位 5:4 DT[1:0]：日期的十位（BCD 格式） (Date tens in BCD format)

位 3:0 DU[3:0]：日期的个位（BCD 格式） (Date units in BCD format)

RTC 控制寄存器 (RTC_CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
								rw	rw	rw	rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	ALRBE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	DCE	FMT	BYPS HAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 6 FMT：小时格式 (Hour format)

0： 24 小时/天格式

1： AM/PM 小时格式

位 5 BYPSHAD：旁路影子寄存器 (Bypass the shadow registers)

0：日历值（从 RTC_SSR、 RTC_TR 和 RTC_DR 读取时）取自影子寄存器，该影子寄存器每两个 RTCCLK 周期更新一次。

1：日历值（从 RTC_SSR、 RTC_TR 和 RTC_DR 读取时）直接取自日历计数器。

注意：如果 APB1 时钟的频率低于 7 倍的 RTCCLK 频率，则必须将 BYPSHAD 置“1”。

RTC 初始化和状态寄存器 (RTC_ISR)

系统复位值：不受影响（INIT、INITF 和 RSF 除外，它们在复位时被清零）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															RECALPF
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTWF	ALRBWF	ALRAWF
	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	rc_w0	r	r	r

位 7 INIT：初始化模式 (Initialization mode)

0：自由运行模式。

1：初始化模式，用于编程时间和日期寄存器（RTC_TR 和 RTC_DR）以及预分频器寄存器(RTC_PRER)。计数器停止计数，当 INIT 被复位后，计数器从新值开始计数。

位 6 INITF：初始化标志 (Initialization flag)

当此位置 1 时， RTC 处于初始化状态，此时可更新事件、日期和预分频器寄存器。

0：不允许更新日历寄存器。

1：允许更新日历寄存器。

位 5 RSF：寄存器同步标志 (Registers synchronization flag)

每次将日历寄存器的值复制到影子寄存器（RTC_SSRx、 RTC_TRx 和 RTC_DRx）时，都会由硬件将此位置 1。在初始化模式下、平移操作挂起时 (SHPF=1) 或在旁路影子寄存器模式 (BYPSHAD=1) 下，该位由硬件清零。该位还可由软件清零。

0：日历影子寄存器尚未同步

1：日历影子寄存器已同步

RTC 预分频器寄存器 (RTC_PRER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PREDIV_A[6:0]						
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PREDIV_S[14:0]														
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 22:16 PREDIV_A[6:0]：异步预分频系数 (Asynchronous prescaler factor)

下面是异步分频系数的公式：

ck_apre 频率 = RTCCLK 频率/(PREDIV_A+1)

注意： PREDIV_A [6:0]= 000000 为禁用值。

位 15 保留，必须保持复位值。

位 14:0 PREDIV_S[14:0]：同步预分频系数 (Synchronous prescaler factor)

下面是同步分频系数的公式：

ck_spre 频率 = ck_apre 频率/(PREDIV_S+1)

RTC 写保护寄存器 (RTC_WPR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								KEY							
								w	w	w	w	w	w	w	w

位 7:0 KEY：写保护关键字 (Write protection key)

可通过软件对该字节执行写操作。

读取该字节时，始终返回 0x00。

有关如何解锁 RTC 寄存器写保护的介绍，请参见 RTC 寄存器写保护。

13.4.2 RTC 基本日历功能实验

13.4.2.1 软件设计

选择 LSE 作为 RTC 时钟源（打开 LSE，等待 LSE 就绪，选择选择 LSE 作为 RTC 时钟源）

使能对 RTC、 RTC 备份寄存器和备份 SRAM 的访问

取消 RTC 寄存器写保护

进入初始化模式（关闭日历计数器）

设置分频值（异步分频和同步分频）

设置 RTC_DR 和 RTC_TR

退出初始化模式（日历计数器重新开始计数）

激活写保护

13.5 RTC 自动唤醒功能

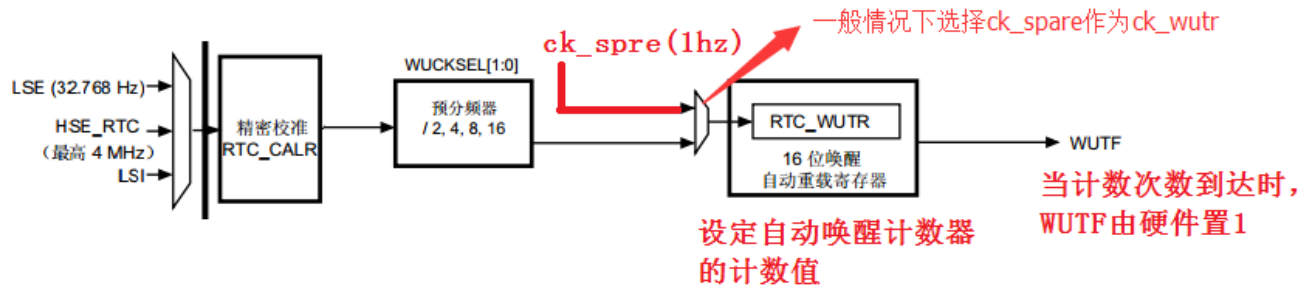
通过设定一个时间周期，当时间到了的时候，就会产生一些标志/中断，通过 IO 口将当前标志输出出去，产生外部中断。

一般自动唤醒都是设定一秒产生一次中断，在中断中获取 RTC 时间。

要使能 RTC 唤醒中断，需按照以下顺序操作：

1. 将 EXTI 线 22 配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_WKUP IRQ 通道并将其使能。
3. 配置 RTC 以生成 RTC 唤醒定时器事件。

13.5.1 自动唤醒功能框图分析



13.5.2 自动唤醒功能相关寄存器

13.5.2.1 RTC 控制寄存器 (RTC_CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
								rw	rw	rw	rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	ALRBE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	DCE	FMT	BYPS HAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 14 WUTIE：使能唤醒定时器使能 (Wakeup timer interrupt enable)

0：禁止唤醒定时器中断

1：使能唤醒定时器中断

位 10 WUTE：唤醒定时器使能 (Wakeup timer enable)

0：禁止唤醒定时器（禁止唤醒定时器才能对唤醒寄存器写操作）

1：使能唤醒定时器

位 2:0 WUCKSEL[2:0]：唤醒时钟选择 (Wakeup clock selection)

000：选择 RTC/16 时钟

001：选择 RTC/8 时钟

010：选择 RTC/4 时钟

011：选择 RTC/2 时钟

10x：选择 ck_spre 时钟（通常为 1 Hz）

11x：选择 ck_spre 时钟（通常为 1 Hz）并将 WUT 计数器值增加 216（见下面的注释）

13.5.2.2 RTC 初始化和状态寄存器 (RTC_ISR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															RECALPF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TAMP 2F	TAMP 1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUT WF	ALRB WF	ALRA WF
	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	rc_w0	r	r	r

位 10 WUTF：唤醒定时器标志 (Wakeup timer flag)

当唤醒自动重载计数器计数到 0 时，由硬件将此标志置 1。

该标志由软件写零清除。

软件必须在 WUTF 再次置 1 的 1.5 个 RTCCLK 周期之前将该标志清零。

位 2 WUTWF：唤醒定时器写标志 (Wakeup timer write flag)

在 RTC_CR 寄存器中的 WUTE 位置 0 后，当唤醒定时器值可更改时，由硬件将该位置 1。

0：不允许更新唤醒定时器配置

1：允许更新唤醒定时器配置

13.5.2.3 RTC 唤醒定时器寄存器 (RTC_WUTR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **WUT[15:0]**：唤醒自动重载值位 (Wakeup auto-reload value bit)
当使能唤醒定时器时（WUTE 置 1），每 (WUT[15:0] + 1) 个 ck_wut 周期将 WUTF 标志置 1 一次。ck_wut 周期通过 RTC_CR 寄存器的 WUCKSEL[2:0] 位进行选择。
当 WUCKSEL[2] = 1 时，唤醒定时器变为 17 位，WUCKSEL[1] 等效为 WUT[16]，即要重载到定时器的最高有效位。
注意：WUTF 第一次置 1 发生在 WUTE 置 1 之后 (WUT+1) 个 ck_wut 周期。禁止在 WUCKSEL[2:0]=011(RTCCLK/2) 时将 WUT[15:0] 设置为 0x0000。

13.5.3 自动唤醒中断配置方法

要使用自动唤醒中断，单单使能 RTC 的自动唤醒中断功能是不行的，除了使能 RTC 的自动唤醒中断功能外还需要配置外部中断线 22 中断

另外七根 EXTI 线连接方式如下：
- EXTI 线 16 连接到 PVD 输出
- EXTI 线 17 连接到 RTC 闹钟事件
- EXTI 线 18 连接到 USB OTG FS 唤醒事件
- EXTI 线 19 连接到以太网唤醒事件
- EXTI 线 20 连接到 USB OTG HS（在 FS 中配置）唤醒事件
- EXTI 线 21 连接到 RTC 入侵和时间戳事件
- EXTI 线 22 连接到 RTC 唤醒事件

23.5 RTC 中断

所有 RTC 中断均与 EXTI 控制器相连。

要使能 RTC 闹钟中断，需按照以下顺序操作：
1. 将 EXTI 线 17 配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_Alarm IRQ 通道并将其使能。
3. 配置 RTC 以生成 RTC 闹钟（闹钟 A 或闹钟 B）。

要使能 RTC 唤醒中断，需按照以下顺序操作：
1. 将 EXTI 线 22 配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_WKUP IRQ 通道并将其使能。
3. 配置 RTC 以生成 RTC 唤醒定时器事件。

要使能 RTC 入侵中断，需按照以下顺序操作：
1. 将 EXTI 线 21 配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 TAMP_STAMP IRQ 通道并将其使能。
3. 配置 RTC 以检测 RTC 入侵事件。

要使能 RTC 时间戳中断，需按照以下顺序操作：
1. 将 EXTI 线 21 配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 TAMP_STAMP IRQ 通道并将其使能。
3. 配置 RTC 以检测 RTC 时间戳事件。

所以，要使用 RTC 自动唤醒中断就需要开启当前 RTC 唤醒中断和外部中断线 22 中断。

13.5.4 自动唤醒配置流程

前提：得到 ck_spre 频率为 1hz

取消 RTC 写保护
禁止唤醒定时器，等待允许更新唤醒定时器配置

选择 ck_spre 作为唤醒定时器时钟源

设置唤醒定时器重载值

使能唤醒中断（模块级中断）

配置外部中断线 22（选择外部中断线 22 作为输入线，上升沿触发，屏蔽软件、事件，使能外部中断线 22 中断，配置 NVIC）

开启唤醒定时器

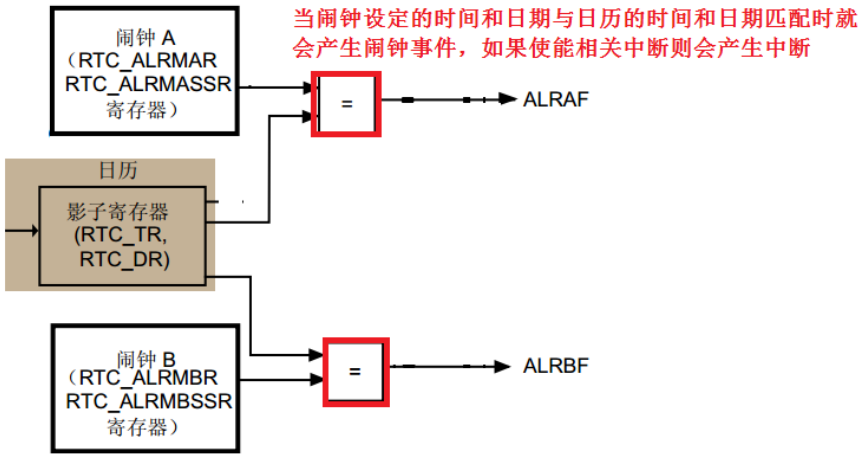
编写中断服务函数（清两处标志）

13.6 RTC 闹钟功能

RTC 单元提供两个可编程闹钟，即闹钟 A 和闹钟 B。可通过将 RTC_CR 寄存器中的 ALRAE 和 ALRBE 位置 1 来使能可编程闹钟功能。如果日历亚秒、秒、分钟、小时、日期或日分别与闹钟寄存器 RTC_ALRMASSR/RTC_ALRMAR 和 RTC_ALRMBSSR/RTC_ALRMBR 中编程的值相匹配，则 ALRAF 和 ALRBF 标志会被置为 1。可通过 RTC_ALRMAR 和 RTC_ALRMBR 寄存器的 MSKx 位以及 RTC_ALRMASSR 和 RTC_ALRMBSSR 寄存器的 MASKSSx 位单独选择各日历字段。可通过 RTC_CR 寄存器中的 ALRAIE 和 ALRBIE 位使能闹钟中断。

闹钟 A 和闹钟 B（如果已通过 RTC_CR 寄存器中的位 OSEL[0:1] 使能）可连接到 RTC_ALARM 输出。可通过 RTC_CR 寄存器的 POL 位配置 RTC_ALARM 极性。

13.6.1 RTC 闹钟功能框图分析



13.6.2 RTC 闹钟功能相关寄存器

13.6.2.1 RTC 控制寄存器 (RTC_CR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
								rw	rw	rw	rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	ALRBE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	DCE	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 13 ALRBIE： 闹钟 B 中断使能 (Alarm B interrupt enable)

0：闹钟 B 中断禁止

1：闹钟 B 中断使能

位 12 ALRAIE：闹钟 A 中断使能 (Alarm A interrupt enable)

0：禁止闹钟 A 中断

1：使能闹钟 A 中断

位 9 ALRBE： 闹钟 B 使能 (Alarm B enable)

0：禁止闹钟 B

1：使能闹钟 B

位 8 ALRAE： 闹钟 A 使能 (Alarm A enable)

0：禁止闹钟 A

1：使能闹钟 A

13.6.2.2 RTC 初始化和状态寄存器 (RTC_ISR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															RECALPF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTWF	ALRBWF	ALRAWF
	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	rc_w0	r	r	r

位 9 ALRBF：闹钟 B 标志 (Alarm B flag)

当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 B 寄存器 (RTC_ALRMBR) 匹配时，由硬件将该标志置 1。

该标志由软件写零清除。

位 8 ALRAF：闹钟 A 标志 (Alarm A flag)

当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 A 寄存器 (RTC_ALRMAR) 匹配时，由硬件将该标志置 1。

该标志由软件写零清除。

位 1 ALRBWF：闹钟 B 写标志 (Alarm B write flag)

在 RTC_CR 寄存器中的 ALRBIE 位置 0 之后，当闹钟 B 的值可更改时，由硬件将该位置 1。

该位在初始化模式下由硬件清零。

0：不允许更新闹钟 B

1：允许更新闹钟 B

位 0 ALRAWF：闹钟 A 写标志 (Alarm A write flag)

在 RTC_CR 寄存器中的 ALRAE 位置 0 后，当闹钟 A 的值可更改时，由硬件将该位置 1。

该位在初始化模式下由硬件清零。

0：不允许更新闹钟 A

1：允许更新闹钟 A

13.6.2.3 RTC 闹钟 A 寄存器 (RTC_ALRMAR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

13.6.2.4 RTC 闹钟 B 寄存器 (RTC_ALRMBR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

13.6.3 RTC 闹钟中断配置方法

要使用自动唤醒中断，单单使能 RTC 的闹钟中断功能是不行的，除了使能 RTC 的闹钟中断功能外还需要配置外部中断线 17 中断

另外七根 EXTI 线连接方式如下：

- EXTI 线 16 连接到 PVD 输出
- EXTI 线 17 连接到 RTC 闹钟事件
- EXTI 线 18 连接到 USB OTG FS 唤醒事件
- EXTI 线 19 连接到以太网唤醒事件
- EXTI 线 20 连接到 USB OTG HS（在 FS 中配置）唤醒事件
- EXTI 线 21 连接到 RTC 入侵和时间戳事件
- EXTI 线 22 连接到 RTC 唤醒事件

23.5 RTC 中断

所有 RTC 中断均与 EXTI 控制器相连。

要使能 RTC 闹钟中断，需按照以下顺序操作：
1. 将 EXTI 线 17 配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_Alarm IRQ 通道并将其使能。
3. 配置 RTC 以生成 RTC 闹钟（闹钟 A 或闹钟 B）。

要使能 RTC 唤醒中断，需按照以下顺序操作：
1. 将 EXTI 线 22 配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_WKUP IRQ 通道并将其使能。
3. 配置 RTC 以生成 RTC 唤醒定时器事件。

要使能 RTC 入侵中断，需按照以下顺序操作：
1. 将 EXTI 线 21 配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 TAMP_STAMP IRQ 通道并将其使能。
3. 配置 RTC 以检测 RTC 入侵事件。

要使能 RTC 时间戳中断，需按照以下顺序操作：
1. 将 EXTI 线 21 配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 TAMP_STAMP IRQ 通道并将其使能。
3. 配置 RTC 以检测 RTC 时间戳事件。

13.6.4 配置流程

闹钟 A 每分钟第 3 秒产生闹钟中断---匹配项只有秒

前提：日历功能要配置完成

取消 RTC 写保护
禁止闹钟 A，等待允许更新闹钟 A
设定闹钟 A 的时间/日期（只匹配秒）
清除一次闹钟 A 标志（防止初始化时标志为 1）
使能闹钟 A 中断
配置外部中断线 17
使能闹钟 A
激活写保护

编写中断服务函数（清两处标志）