晟矽微电 应用笔记

MS32F031

RTC SS 中断使用

AN22001

V1.0

目 录

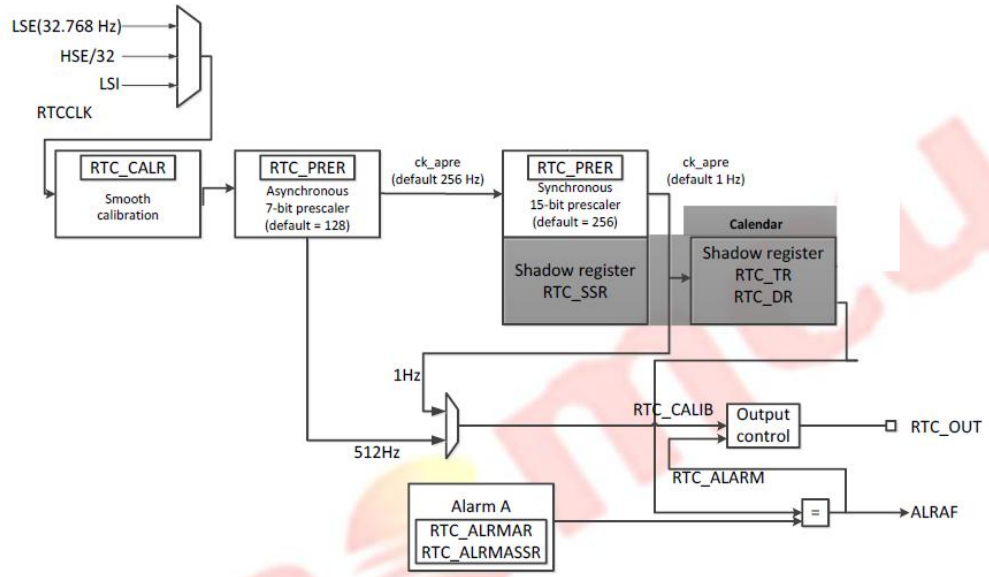
1 适用范围

本文档适用于 MS32F031A6 使用 RTC ms 中断应用。

2 RTS SS 中断解析

RTC ms 中断，需要通过 SS（子秒，亚秒）匹配实现。

2.1 理论基础



（RTC 时钟）/ （PREDIV_A 7bit 异步分频）/ （PREDIV_S 15bit 同步分频）→1Hz（1S）；

RTC 亚秒寄存器(RTC_SSR)，可以理解为一个初值 PREDIV_S（同步预分频器系数）的递减计数器，(PREDIV_S 对应 1000ms)。

RTC_PRER - PREDIV_S:

BIT[14:0]	PREDIV_S[14:0]	同步预分频器系数 ck_spre 频率 = ck_apre 频率 /(PREDIV_S+1)

RTC_SSR - SS:

BIT[15:0]	SS	亚秒值 SS[15:0] 是同步预分频器计数器中的值。“秒”的小数部分由下列公式给出: Second fraction = (PREDIV_S - SS) / (PREDIV_S + 1) 注: 仅当执行完一个移位操作后，SS 可能大于 PREDIV_S。此时，正确的时间/日期比 RTC_TR/RTC_DR 少一秒。

2.2 通用实现方式

15bit MASKSS 均参与匹配，中断程序中更新 RTC_ALRMASSR SS 值（减所需时间增量值；

增量值=$\frac{\text{所需时间}}{\text{增量单位}}$，增量单位（$\frac{1000}{(PREDIV\_S+1)}$）ms。

注：设置 RTC_ALRMASSR SS 时需要 Disable Alarm，DisableWriteProtection；

可参与附件例程“RTC_Alarm_SS_universal”。

2.3 另一种实现方式

代码参考“MS32F0x1_Periph_Lib_Example\proj\MS32F031_EV\RTC\RTC_Alarm_SS”。
例程以使用 LSE 为例，进行说明；同步分频值 PREDIV_S 为 0xFF；

RTC_CFG.c | RTC_CFG.h

```
37  #define RTC_INIT_SECONDS          0x45
38  // 2021-12-10
39  #define RTC_INIT_WEEKDAY          MS32_RTC_WEEKDAY_FRIDAY
40  #define RTC_INIT_DAY              0x10
41  #define RTC_INIT_MONTH            MS32_RTC_MONTH_DECEMBER
42  #define RTC_INIT_YEAR             0x21
43
44  // 23:58:50
45  #define RTC_ALARM_HOURS           0x23
46  #define RTC_ALARM_MINUTES         0x58
47  #define RTC_ALARM_SECONDS         0x50
48
49  // ck_apre=LSxFreq/(ASYNC prediv + 1) ,LSI:40K,LSE:32.768K
50  // ck_spre=ck_apre/(SYNC prediv + 1)
51  #if(defined(RTC_USER_LSE) && RTC_USER_LSE)
52  #define RTC_ASYNCH_PREDIV         0x7F
53  #define RTC_SYNCH_PREDIV          0xFF
54  #else
55  #define RTC_ASYNCH_PREDIV         39
56  // RTC_SYNCH_PREDIV 1023,RTC calendar clk 0.97Hz
57  // RTC_SYNCH_PREDIV 999,alatm add once 1s,RTC calendar clk 1Hz
58  #define RTC_SYNCH_PREDIV          0x3FF
59  #endif
```

RTC Alarm A 亚秒寄存器(RTC_ALRMASSR) 中的 SS 使用默认值 0；

BIT[27:24]	MASKSS[3:0]	从该位起品比最明显的位。 0: Alarm A 不匹配亚秒值。报警在秒单元增加(假设其余字段是相互匹配的) 时置 1。 1: SS[14:1] 不参与 Alarm A 匹配。只有 SS[0] 参与匹配。 2: SS[14:2] 不参与 Alarm A 匹配。只有 SS[1:0] 参与匹配。 3: SS[14:2] 不参与 Alarm A 匹配。只有 SS[2:0] 参与匹配。 ... 12: SS[14:12] 不参与 Alarm A 匹配。只有 SS[11:0] 参与匹配。 13: SS[14:13] 不参与 Alarm A 匹配。只有 SS[12:0] 参与匹配。 14: SS[14] 不参与 Alarm A 匹配。只有 SS[13:0] 参与匹配。 15: 15 个 SS 位均需参与匹配，且需匹配成功以激活报警。 同步计数器溢出位(位15) 始终不参与匹配。只有在移位操作后，该位才不为“0”。
BIT[14:0]	SS[14:0]	亚秒值 比较亚秒值与同步预分频器计数器中的内容，从而判断报警是否已经被激活。 只有位 0 到 MASKSS-1 参与比较。

MASKSS 以 3 为例（即 SS[2:0]参与匹配）；

15bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_SSR	未使用							计数值							
RTC_ALRMASSR --SS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RTC_ALRMASSR --MSAKSS	不参与匹配												参与匹配		

计数值低 3 位为 0 时匹配，对应时间 31.25ms；

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
未使用							1000							
未使用							500							
未使用							250							
未使用							125							
未使用							62.5							
未使用							31.25							
未使用							15.625							
未使用							7.8							

MASKSS 与所需时间的对应关系公式：

$$\left(\frac{1000}{(PREDIV\_S+1)}\times 2^{匹配位数}\right)\ ms$$

例程中匹配位数为 3，带入计算（$\frac{1000}{(255+1)}\times 2^3$）=31.25ms。

这种实现方式需 PREDIV_S 为 2N；若 PREDIV_S 不为 2N，整秒 RTC_SSR 第一次匹配有差值（原因 PREDIV_S 与 2N 差值）。

3 修订记录

版本	修订日期	修订内容
V1.0	2022-02-16	1359，初版