
第 29 章 实时时钟和日历（RTCC）

目录

本章包括下列主题：

29.1 简介	29-2
29.2 RTCC 模块寄存器	29-3
29.3 工作原理	29-11
29.4 闹钟	29-17
29.5 休眠模式	29-19
29.6 复位	29-19
29.7 外设模块禁止（PMD）寄存器	29-19
29.8 寄存器映射	29-20
29.9 相关应用笔记	29-21
29.10 版本历史	29-22

29.1 简介

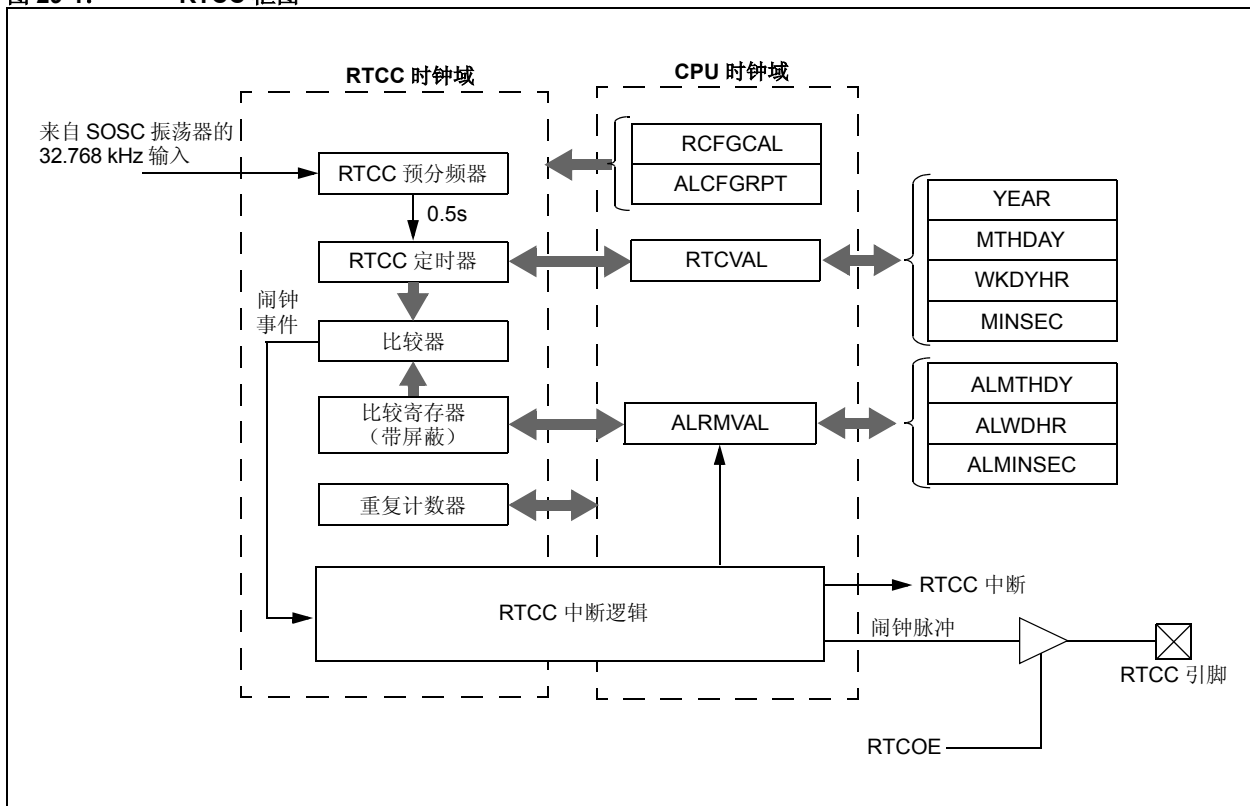
本章将讨论 PIC24F 器件的实时时钟和日历的硬件模块及其操作。下面列出了该模块的一些关键特性：

- 时间：时、分和秒
- 24 小时格式（军用时间）
- 日历：星期、日期、月和年
- 闹钟可配置
- 年份范围：2000 至 2099
- 闰年修正
- 用于小型固件的 BCD 格式
- 为低功耗操作进行了优化
- 带自动调节的用户校准
- 校准范围：每月 ± 2.64 秒误差
- 要求：外部 32.768 kHz 时钟晶振
- RTCC 引脚上的闹钟脉冲或秒时钟输出

该模块提供了实时时钟和日历（Real-Time Clock and Calendar，RTCC）功能。该模块是为需要长时间维持精确时间的应用设计的，无需或很少需要 CPU 干预。该模块为低功耗使用作了优化，以便在跟踪时间时延长电池寿命。

RTCC 模块是百年时钟和日历，能自动检测闰年。时钟范围从 2000 年 1 月 1 日 00:00:00（午夜）到 2099 年 12 月 31 日 23:59:59。小时数以 24 小时（军用时间）格式提供。该时钟提供一秒的时间间隔时间，用户可看到半秒的时间间隔。

图 29-1: RTCC 框图



29.2 RTCC 模块寄存器

本节将讨论 RTCC 模块寄存器，这些寄存器可分为以下三类：

RTCC 控制寄存器

- RCFGCAL
- PADCFG1
- ALCFGRPT

RTCC 值寄存器

- RTCVAL (以下 4 个寄存器通过 RTCVAL 寄存器寻址)
 - YEAR
 - MTHDY
 - WKDYHR
 - MINSEC

闹钟值寄存器

- ALRMVAL (以下 3 个寄存器通过 ALRMVAL 寄存器寻址)
 - ALMTHDY
 - ALWDHR
 - ALMINSEC

注： 为了方便引述，RTCVAL 寄存器的高位部分用 RTCVAL<15:8> 表示，低位部分用 RTCVAL<7:0> 表示。相同的表示方法也适用于 ALRMVAL，其中高位部分是 ALRMVALH，而低位部分是 ALRMVALL。

29.2.1 RTCC 控制寄存器

寄存器 29-1: RCFGAL: RTCC 校准和配置寄存器⁽¹⁾

R/W-0	U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
RTCEN ⁽²⁾	—	RTCWREN	RTCSYNC	HALFSEC ⁽³⁾	RTCOE	RTCPtr1	RTCPtr0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
bit 7							bit 0

图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = POR 值	1 = 置 1	0 = 清零
		x = 未知

- bit 15 **RTCEN:** RTCC 使能位⁽²⁾
 1 = 使能 RTCC 模块
 0 = 禁止 RTCC 模块
- bit 14 **未实现:** 读为 0
- bit 13 **RTCWREN:** RTCC 值寄存器写使能位
 1 = RTCVAL<15:8> 和 RTCVAL<7:0> 寄存器可由用户写入
 0 = RTCVAL<15:8> 和 RTCVAL<7:0> 寄存器已锁定, 不可由用户写入
- bit 12 **RTCSYNC:** RTCC 值寄存器读同步位
 1 = 由于计满返回的波及, RTCVAL<15:8>、RTCVAL<7:0> 和 ALCFGRPT 寄存器在读操作过程中可能改变, 从而导致读取的数据无效。如果两次读取寄存器得到的数据相同, 可认为数据是有效的。
 0 = RTCVAL<15:8>、RTCVAL<7:0> 或 ALCFGRPT 寄存器在读取时无需考虑计满返回的波及
- bit 11 **HALFSEC:** 半秒状态位⁽³⁾
 1 = 一秒的后一半
 0 = 一秒的前一半
- bit 10 **RTCOE:** RTCC 输出使能位
 1 = 使能 RTCC 时钟输出
 0 = 禁止 RTCC 时钟输出
- bit 9-8 **RTCPtr<1:0>:** RTCC 值寄存器窗口指针位
 读取 RTCVAL<15:8> 和 RTCVAL<7:0> 寄存器时, 指向相应的 RTCC 值寄存器; 每当 RTCVAL<15:8> 读或写时 RTCPtr<1:0> 的值就减 1, 直到达到 00。
RTCVAL<15:8>:
 00 = 分钟数
 01 = 星期
 10 = 月
 11 = 保留
RTCVAL<7:0>:
 00 = 秒数
 01 = 小时数
 10 = 日
 11 = 年

- 注 1: RCFGAL 寄存器只受 POR 的影响。
 2: 仅当 RTCWREN = 1 时允许写入 RTCEN 位。
 3: 该位是只读的。写入 MINSEC 寄存器的低半部分时, 它被清零。

寄存器 29-1: RCFGCAL: RTCC 校准和配置寄存器⁽¹⁾ (续)

bit 7-0 **CAL<7:0>**: RTC 漂移校准位
 01111111 = 最大正向调整: 每分钟增加 508 个 RTC 时钟脉冲
 ...
 00000001 = 最小正向调整: 每分钟增加 4 个 RTC 时钟脉冲
 00000000 = 无调整
 11111111 = 最小负向调整: 每分钟减少 4 个 RTC 时钟脉冲
 ...
 10000000 = 最大负向调整: 每分钟减少 512 个 RTC 时钟脉冲

- 注 1: RCFGCAL 寄存器只受 POR 的影响。
 2: 仅当 RTCWREN = 1 时允许写入 RTCEN 位。
 3: 该位是只读的。写入 MINSEC 寄存器的低半部分时, 它被清零。

寄存器 29-2: PADCFG1: 焊垫配置控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	RTSECSEL ⁽¹⁾	PMP TTL
bit 7							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 值 1 = 置 1 0 = 清零 x = 未知

bit 15-2 **未实现**: 读为 0
 bit 1 **RTSECSEL**: RTCC 秒时钟输出选择位⁽¹⁾
 1 = 为 RTCC 引脚选择了 RTCC 秒时钟
 0 = 为 RTCC 引脚选择了 RTCC 闹钟脉冲
 bit 0 **PMP TTL**: PMP 模块 TTL 输入缓冲器选择位
 1 = PMP 模块使用 TTL 输入缓冲器
 0 = PMP 模块使用施密特输入缓冲器

- 注 1: 要启用实际 RTCC 输出, RTCOE (RCFGCAL<10>) 位必须置 1。

PIC24F 系列参考手册

寄存器 29-3: **ALCFGRPT: 闹钟配置寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 值

1 = 置 1

0 = 清零

x = 未知

bit 15

ALRMEN: 闹钟使能位

1 = 闹钟已使能 (每当 ARPT<7:0> = 00 和 CHIME = 0 时, 发生闹钟事件后都自动清零)

0 = 闹钟已禁止

bit 14

CHIME: 响铃 (Chime) 使能位

1 = 响铃已禁止; ARPT<7:0> 允许从 00h 进位到 FFh

0 = 响铃已禁止; ARPT<7:0> 到达 00h 就停止

bit 13-10

AMASK<3:0>: 闹钟屏蔽配置位

0000 = 每半秒

0001 = 每秒

0010 = 每 10 秒

0011 = 每分钟

0100 = 每 10 分钟

0101 = 每小时

0110 = 一天一次

0111 = 一周一次

1000 = 一月一次

1001 = 一年一次 (除了配置在 2 月 29 日, 亦即每 4 年一次的情况外)

101x = 保留——未使用

11xx = 保留——未使用

bit 9-8

ALRMPTR<1:0>: 闹钟值寄存器窗口指针位

读取 ALRMVALH 和 ALRMVALL 寄存器时, 指向相应的闹钟值寄存器; 每当读或写 ALRMVALH 时 ALRMPTR<1:0> 的值就减 1, 直到达到 00。

ALRMVAL<15:8>:

00 = ALRMMIN

01 = ALRMWD

10 = ALRMMNTH

11 = 未实现

ALRMVAL<7:0>:

00 = ALRMSEC

01 = ALRMHR

10 = ALRMDAY

11 = 未实现

bit 7-0

ARPT<7:0>: 闹钟重复计数器值位

11111111 = 闹钟将再重复 255 次

...

00000000 = 闹钟将不再重复

每当发生闹钟事件时计数器就减 1。除非 CHIME = 1, 否则计数器不能从 00h 进位到 FFh。

29.2.2 RTCVAL 寄存器映射

寄存器 29-4: 年: 年值寄存器⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15				bit 8			

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7-4 **YRTEN<3:0>**: 年份的十位数的二 - 十进制码 (Binary Coded Decimal, BCD) 值; 值为 0 到 9bit 3-0 **YRONE<3:0>**: 年份的个位数的 BCD 值; 值为 0 到 9

注 1: 仅当 RTCWREN = 1 时允许写入 YEAR 寄存器。

寄存器 29-5: MTHDY: 月和日值寄存器⁽¹⁾

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MHTTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15				bit 8			

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 值

1 = 置 1

0 = 清零

x = 未知

bit 15-13 未实现: 读为 0

bit 12 **MHTTEN0**: 月份的十位数的 BCD 值; 值为 0 或 1bit 11-8 **MTHONE<3:0>**: 月份的个位数的 BCD 值; 值为 0 到 9

bit 7-6 未实现: 读为 0

bit 5-4 **DAYTEN<1:0>**: 日的十位数的 BCD 值; 值为 0 到 3bit 3-0 **DAYONE<3:0>**: 日的个位数的 BCD 值; 值为 0 到 9

注 1: 仅当 RTCWREN = 1 时允许写入该寄存器。

寄存器 29-6: WKDYHR: 星期和小时数值寄存器⁽¹⁾

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15						bit 8	

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7						bit 0	

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 值 1 = 置 1 0 = 清零 x = 未知

bit 15-11 未实现: 读为 0
 bit 10-8 **WDAY<2:0>**: 星期的 BCD 值; 值为 0 到 6
 bit 7-6 未实现: 读为 0
 bit 5-4 **HRTEN<1:0>**: 小时的十位数的 BCD 值; 值为 0 到 2
 bit 3-0 **HRONE<3:0>**: 小时的个位数的 BCD 值; 值为 0 到 9

注 1: 仅当 RTCWREN = 1 时允许写入该寄存器。

寄存器 29-7: MINSEC: 分钟数和秒数值寄存器

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15						bit 8	

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7						bit 0	

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
 -n = POR 值 1 = 置 1 0 = 清零 x = 未知

bit 15 未实现: 读为 0
 bit 14-12 **MINTEN<2:0>**: 分钟的十位数的 BCD 值; 值为 0 到 5
 bit 11-8 **MINONE<3:0>**: 分钟的个位数的 BCD 值; 值为 0 到 9
 bit 7 未实现: 读为 0
 bit 6-4 **SECTEN<2:0>**: 秒的十位数的 BCD 值; 值为 0 到 5
 bit 3-0 **SECONE<3:0>**: 秒的个位数的 BCD 值; 值为 0 到 9

29.2.3 ALRMVAL 寄存器映射

寄存器 29-8: ALMTHDY: 闹钟月和日值寄存器 ⁽¹⁾

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							
							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							
							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
-n = POR 值 1 = 置 1 0 = 清零 x = 未知

- bit 15-13 未实现: 读为 0
bit 12 **MHTEN0**: 月份的十位数的 BCD 值; 值为 0 或 1
bit 11-8 **MTHONE<3:0>**: 月份的个位数的 BCD 值; 值为 0 到 9
bit 7-6 未实现: 读为 0
bit 5-4 **DAYTEN<1:0>**: 日的十位数的 BCD 值; 值为 0 到 3
bit 3-0 **DAYONE<3:0>**: 日的个位数的 BCD 值; 值为 0 到 9

注 1: 仅当 RTCWREN = 1 时允许写入该寄存器。

寄存器 29-9: ALWDHR: 闹钟星期和小时数值寄存器 ⁽¹⁾

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							
							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							
							bit 0

图注:

R = 可读位 W = 可写位 U = 未实现位, 读为 0
-n = POR 值 1 = 置 1 0 = 清零 x = 未知

- bit 15-11 未实现: 读为 0
bit 10-8 **WDAY<2:0>**: 星期的 BCD 值; 值为 0 到 6
bit 7-6 未实现: 读为 0
bit 5-4 **HRTEN<1:0>**: 小时的十位数的 BCD 值; 值为 0 到 2
bit 3-0 **HRONE<3:0>**: 小时的个位数的 BCD 值; 值为 0 到 9

注 1: 仅当 RTCWREN = 1 时允许写入该寄存器。

寄存器 29-10: ALMINSEC: 闹钟分钟数和秒数值寄存器

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

图注:							
R = 可读位		W = 可写位		U = 未实现位, 读为 0			
-n = POR 值		1 = 置 1		0 = 清零		x = 未知	

- bit 15 未实现: 读为 0
- bit 14-12 **MINTEN<2:0>**: 分钟的十位数的 BCD 值; 值为 0 到 5
- bit 11-8 **MINONE<3:0>**: 分钟个位数的 BCD 值; 值为 0 到 9
- bit 7 未实现: 读为 0
- bit 6-4 **SECTEN<2:0>**: 秒的十位数的 BCD 值; 值为 0 到 5
- bit 3-0 **SECONE<3:0>**: 秒的个位数的 BCD 值; 值为 0 到 9

29.2.4 RTCEN 位写

RTCWREN = 0 时写入 RTCEN 位的尝试将被忽略。RTCWREN 必须置 1, 然后才会发生对 RTCEN 的写入。

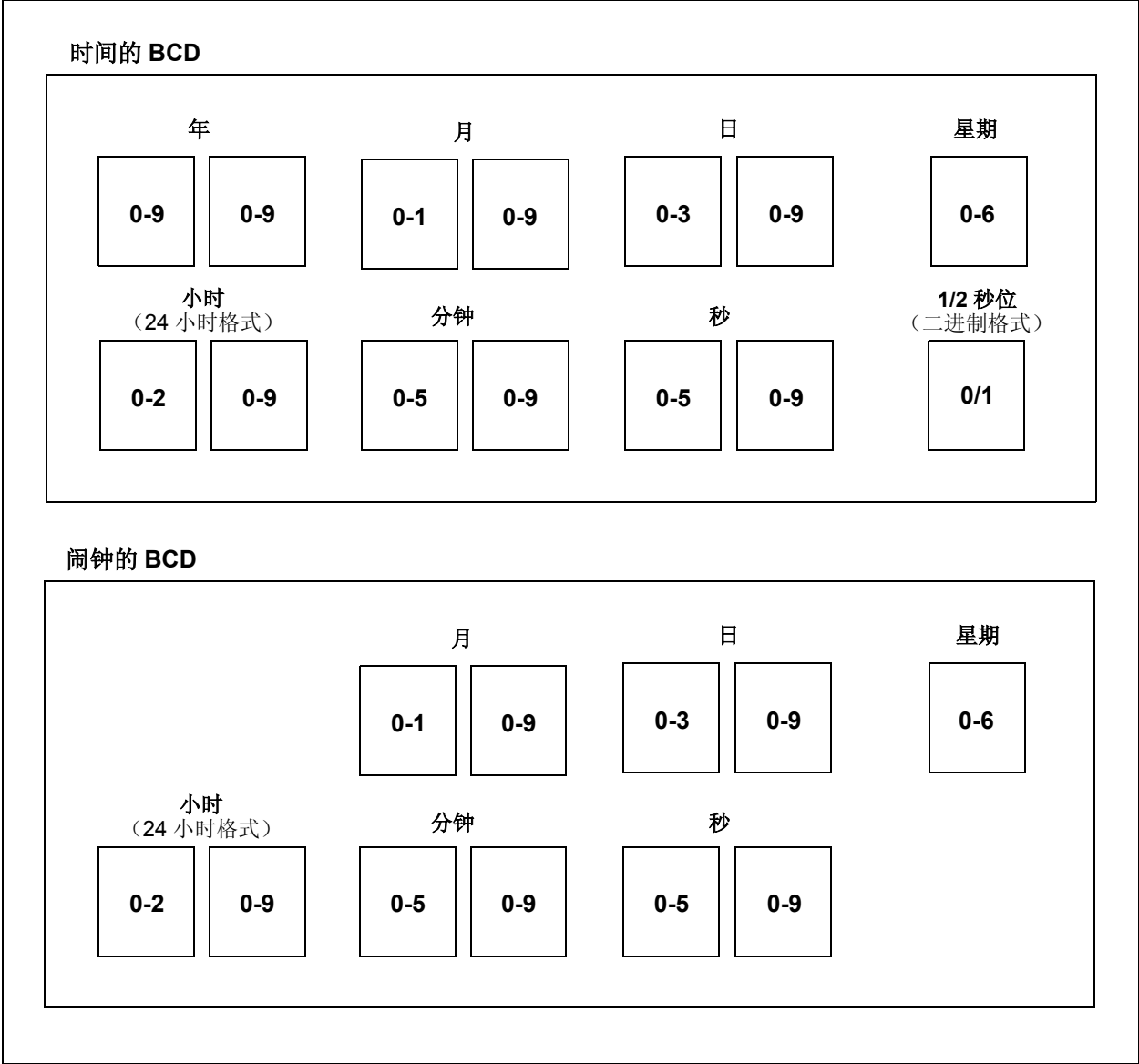
与 RTCEN 位相同, RTCVAL<15:8> 和 RTCVAL<7:0> 寄存器只能在 RTCWREN = 1 时写入。RTCWREN = 0 时写入这些寄存器也会被忽略。

29.3 工作原理

29.3.1 寄存器接口

RTCC 和闹钟值的寄存器接口是用二 - 十进制码（BCD）格式实现的。这在使用该模块时简化了固件，因为每个位（digit）的值都包含在它自己的 4 位（bit）值中了（见图 29-2）。

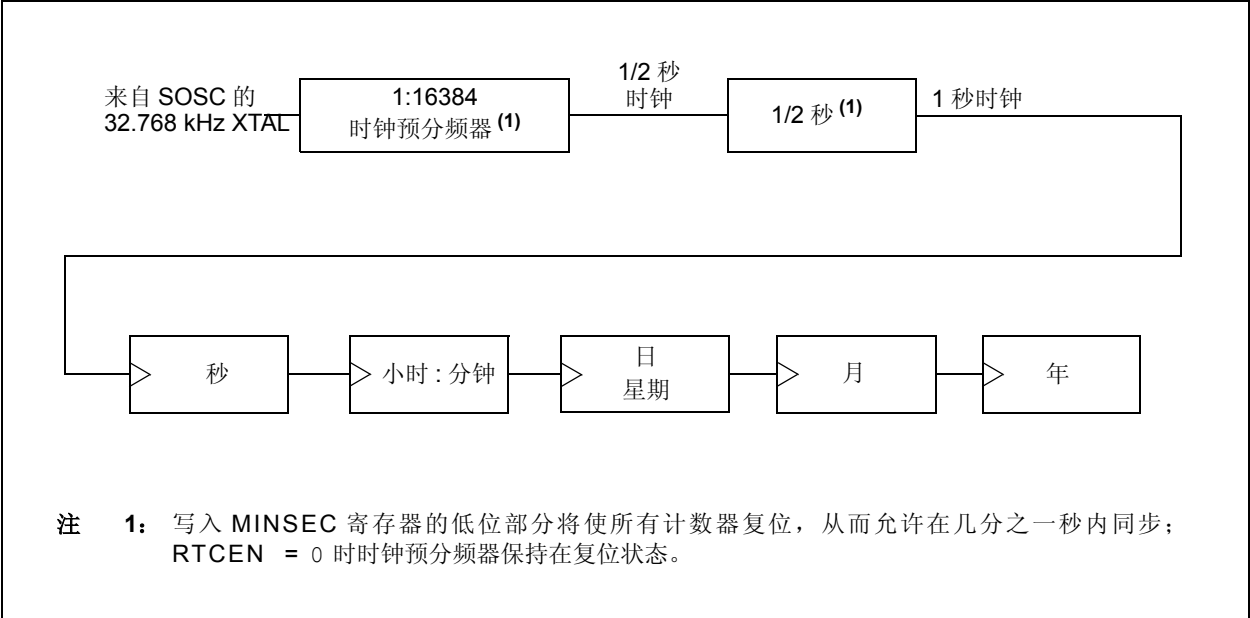
图 29-2: 定时器和闹钟位格式



29.3.2 时钟源

如前所述，RTCC 模块应由 32.768 kHz 的外部实时时钟晶振提供时钟源。晶振校准可通过该模块实现，产生的误差为每月 3 秒或更少（更多详细信息，请参见第 29.3.9 节“校准”）。

图 29-3: 时钟源复用



29.3.2.1 实时时钟晶振使能

要允许 RTCC 模块由外部 32.768 kHz 晶振提供时钟源，OSCCON 寄存器中的 SOSCEN 位（见第 6 章“振荡器”的 Register 6-1）必须置 1。在使能 RTCC 时，该位是在 RTCC 模块以外用户惟一需要操作的位。

29.3.3 半进位规则

本节说明发生进位时，会影响哪些定时器值。

- 每日的时间：从 23:59:59 到 00:00:00 向日字段进位
- 月：从 12/31 到 01/01 向年字段进位
- 星期：从 6 到 0 无进位（见表 29-1）
- 年进位：从 99 到 00；这也标志着 RTCC 使用的终结

关于日到月的进位设定，请参见表 29-2。

考虑到以下值是 BCD 格式，进位到 BCD 的高位将在计数为 10 时发生，而不是在计数为 16 时（秒数、分钟数、小时数、星期、日和月）。

表 29-1： 星期设定

星期	
周日	0
周一	1
周二	2
周三	3
周四	4
周五	5
周六	6

表 29-2： 日到月进位设定

月	最大日字段
01（一月）	31
02（二月）	28 或 29（见第 29.3.4 节“闰年”）
03（三月）	31
04（四月）	30
05（五月）	31
06（六月）	30
07（七月）	31
08（八月）	31
09（九月）	30
10（十月）	31
11（十一月）	30
12（十二月）	31

29.3.4 闰年

由于 RTCC 模块的年份范围是 2000 到 2099，闰年是通过以上范围内的年份能否被 4 整除来确定的。闰年中惟一受影响的月份是二月。二月在闰年有 29 天，其他年份中是 28 天。

29.3.5 通用功能

所有包含几秒或更大时间值的定时器寄存器都可写。用户只需通过寄存器指针将要写入的年、月、日、时、分和秒写入这些寄存器，就可以配置时间（见第 29.3.8 节“寄存器映射”）。随后定时器就会用新写入的值从所需起点开始计数。通过将 **RTCEN** 位（**RCFGCAL<15>**）置 1 来使能 **RTCC** 模块。如果定时器在调节这些寄存器时是使能的，则定时器仍将继续递增。但是，任何时候只要写入 **MINSEC** 寄存器，两个定时器预分频器都将复位为 0。这就允许在几分之一秒内同步。定时器寄存器的更新周期和 CPU 执行写指令的周期相同。用户需负责确保 **RTCEN** = 1 时，更新的寄存器不会同时递增。这可以通过几种方式实现：

- 检查 **RTCSYNC** 位（**RCFGCAL<12>**）
- 检查前面可能发生进位的位
- 在秒脉冲（或闹钟中断）后立即更新寄存器

用户可看到计数器的半秒字段。该值是只读的，只能通过写入 **MINSEC** 寄存器的低位部分复位。

注： 尽管更新寄存器时应注意以上事项，但有些寄存器仍有较大的时间段可更新。例如，**MTHDY** 寄存器高位字节每月只时钟同步一次，而低位字节每天时钟同步一次。

29.3.6 寄存器读写安全窗口

RTCSYNC 位表示可安全读写 **RTCC** 时钟域寄存器的时间段，在这段时间内无需担心计满返回的问题。**RTCSYNC** = 0 时，CPU 可安全访问寄存器。无论 **RTCSYNC** = 1 还是 0，用户都应采用固件方法确保数据读操作未落在计满返回边界，从而导致无效或部分读取。该固件方法包括读取每个寄存器两遍，然后比较两个值。如果两个值匹配，则未发生计满返回。

29.3.7 写锁定

要对任何 **RTCC** 定时器寄存器的执行写操作，必须先将 **RTCWREN** 位（**RCFGCAL<13>**）置 1（见例 29-1）。

例 29-1: 将 **RTCWREN** 位置 1

MOV	#NVMKEY, W1	;move the address of NVMKEY into W1
MOV.b	#0x55, W2	
MOV.b	#0xAA, W3	
MOV.b	W2, [W1]	;start 55/AA sequence
MOV.b	W3, [W1]	
BSET	RCFGCAL, #13	;set the RTCWREN bit

注： 为避免意外写入定时器，建议其他任何时候 **RTCWREN** 位（**RCFGCAL<13>**）都保持清零。要将 **RTCWREN** 位置 1，在 55h/AA 序列和 **RTCWREN** 置 1 之间只允许 1 个指令周期的时间段；因此，建议遵循例 29-1 中的代码示例。

29.3.8 寄存器映射

为限制寄存器接口，RTCC 定时器和闹钟定时器寄存器通过相应的寄存器指针访问。RTCC 值寄存器窗口（RTCVAL<15:8> 和 RTCVAL<7:0>）使用 RTCPTR 位（RCFGCAL<9:8>）选择所需定时器寄存器对（见表 29-3）。

通过读或写 RTCVAL<15:8> 寄存器，RTCC 指针值 RTCPTR<1:0> 减 1，直到达到 00。一旦达到 00，MINUTES 和 SECONDS 值可通过 RTCVAL<15:8> 和 RTCVAL<7:0> 访问，直到手动更改指针值。

表 29-3: RTCC 值寄存器映射

RTCPTR<1:0>	RTCC 值寄存器窗口	
	RTCVAL<15:8>	RTCVAL<7:0>
00	分钟	秒
01	星期	小时
10	月	日
11	—	年

闹钟值寄存器窗口（ALRMVALH 和 ALRMVALL）使用 ALRMPTR 位（ALCFGPT<9:8>）选择所需闹钟寄存器对（见表 29-4）。

通过读或写 ALRMVALH 寄存器，闹钟指针值 ALRMPTR<1:0> 减 1，直到达到 00。一旦达到 00，ALRMMIN 和 ALRMSEC 值可通过 ALRMVALH 和 ALRMVALL 访问，直到手动更改指针值。

表 29-4: ALRMVAL 寄存器映射

ALRMPTR<1:0>	闹钟值寄存器窗口	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

考虑到 16 位内核并不区分 8 位和 16 位读操作，用户要注意，读 ALRMVALH 或 ALRMVALL 寄存器时都会使 ALRMPTR<1:0> 值减 1。同样的规律也适用于 RTCVAL<15:8> 或 RTCVAL<7:0> 寄存器，读取它们会使 RTCPTR<1:0> 减 1。

注： 这只适用于读操作，不适用于写操作。写操作可能和具体的字节有关。

29.3.9 校准

实时晶振输入可用周期性自动调节功能校准。正确校准后，RTCC 可提供小于每月 3 秒的误差。这是通过找到误差时钟脉冲数，将该值存储到 RCFGAL 寄存器的低位部分实现的。装入 RCFGAL 低位部分的 8 位有符号值乘以 4，每分钟一次从 RTCC 定时器中加上或减去。关于 RTCC 校准，请参见以下步骤：

1. 用器件上的其他定时器资源，用户必须找出 32.768 kHz 晶振的误差。
2. 知道误差后，必须将它转换为每分钟误差时钟脉冲数。公式如下：
(理想频率 (32,758) - 测得频率) * 60 = 每分钟误差时钟数
3. a) 如果振荡器快于理想频率（从步骤 2 得出的负的结果），RCFGAL 寄存器值必须为负。这会导致每分钟从定时器计数器减去指定的时钟脉冲数。
b) 如果振荡器慢于理想频率（从步骤 2 得出的正的结果），RCFGAL 寄存器值必须为正。这会导致每分钟在定时器计数器上加上指定的时钟脉冲数。
4. 将正确的值装入 RCFGAL 寄存器。

只有当定时器关闭或紧接秒脉冲的上升沿时，才会发生对 RCFGAL 寄存器低位部分的写入。

注：	是否在误差值中包含晶振初始误差、温度造成的漂移和晶振老化造成的漂移，由用户自行决定。
-----------	--

29.4 闹钟

- 可在半秒到一年的范围内配置
- 使用 **ALRMEN** 位（寄存器 29-3 的 **ALCFGRPT<15>**）使能
- 有一次性闹钟和重复闹钟选项可用

29.4.1 配置闹钟

闹钟功能用 **ALRMEN** 位使能。发出闹钟后该位清零。如果 **CHIME** 位 = 1 或者 **ALCFGRPT** 的低位部分是 $\neq 00$ ，则不会将该位清零。

闹钟的间隔选择通过 **AMASK** 位（**ALCFGRPT<13:10>**）配置，请参见图 29-4。这些位决定了要触发闹钟，闹钟的哪些位、多少位必须和时钟值匹配。也可以配置闹钟使之根据预先配置的间隔重复。闹钟使能后发生的总次数存储在 **ALCFGRPT** 寄存器的低位部分。

注： 闹钟使能时（**ALRMEN** = 1），更改除 **RCFGCAL** 和 **ALCFGRPT** 寄存器以外的任何寄存器以及 **CHIME** 位，都会导致误闹钟事件，进而导致错误的闹钟中断。为避免误闹钟事件，只应在闹钟禁止时（**ALRMEN** = 0）更改定时器和闹钟值。建议在 **RTCSYNC** = 0 时更改 **ALCFGRPT** 寄存器和 **CHIME** 位。

图 29-4: 闹钟屏蔽设置

闹钟屏蔽设置 AMASK<3:0>	星期	月	日	小时	分钟	秒
0000 ——每半秒	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
0001 ——每秒	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
0010 ——每 10 秒	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> s
0011 ——每分钟	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> s <input type="checkbox"/> s
0100 ——每 10 分钟	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> m	<input type="checkbox"/> s <input type="checkbox"/> s
0101 ——每小时	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> m <input type="checkbox"/> m	<input type="checkbox"/> s <input type="checkbox"/> s
0110 ——每天	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> h <input type="checkbox"/> h	<input type="checkbox"/> m <input type="checkbox"/> m	<input type="checkbox"/> s <input type="checkbox"/> s
0111 ——每周	<input type="checkbox"/> d	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> h <input type="checkbox"/> h	<input type="checkbox"/> m <input type="checkbox"/> m	<input type="checkbox"/> s <input type="checkbox"/> s
1000 ——每月	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> d <input type="checkbox"/> d	<input type="checkbox"/> h <input type="checkbox"/> h	<input type="checkbox"/> m <input type="checkbox"/> m	<input type="checkbox"/> s <input type="checkbox"/> s
1001 ——每年 ⁽¹⁾	<input type="checkbox"/>	<input type="checkbox"/> m <input type="checkbox"/> m	<input type="checkbox"/> d <input type="checkbox"/> d	<input type="checkbox"/> h <input type="checkbox"/> h	<input type="checkbox"/> m <input type="checkbox"/> m	<input type="checkbox"/> s <input type="checkbox"/> s

注 1： 每年，除非配置为 2 月 29 日。

ALCFGRPT = 00 并且 CHIME 位 = 0 (ALCFGRPT<14>) 时，重复功能被禁止，只发生单次闹钟。通过将 FFh 装入 ALCFGRPT 寄存器的低位部分，闹钟可重复多达 255 次。

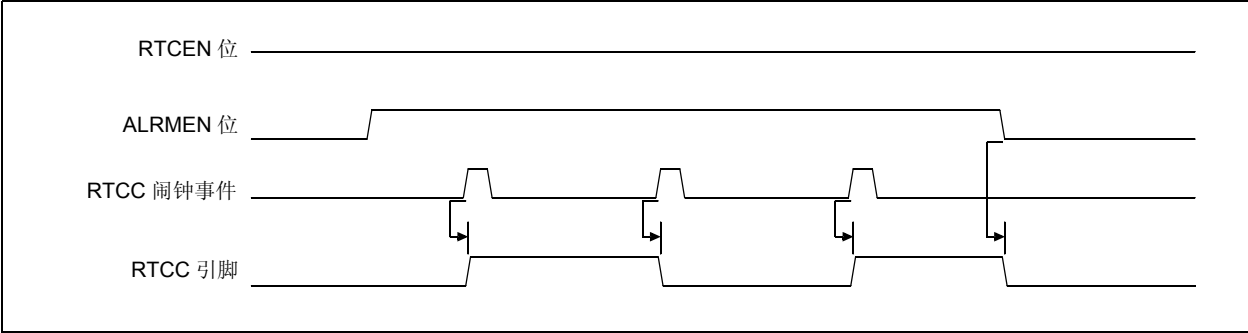
每个闹钟发出后，ALCFGRPT 寄存器都减 1。寄存器达到 00 后，将最后一次发出闹钟，此后 ALRMEN 位将自动清零，闹钟将关闭。如果 CHIME 位 = 1，闹钟可能不断重复。CHIME = 1 时，ALCFGRPT 寄存器达到 00 时不会禁止闹钟，而是进位到 FF，继续无限计数。

29.4.2 闹钟中断

每个闹钟事件发生时，都会产生中断。此外会提供闹钟脉冲输出，其频率是闹钟频率的一半。该输出完全和 RTCC 时钟同步，可用作其他外设的触发时钟。该输出在 RTCC 引脚上可用。输出脉冲时钟占空比为 50%，频率为闹钟时间频率的一半（见图 29-5）。

RTCC 引脚也能输出秒时钟。用户可在 RTCC 模块或秒时钟输出生成的闹钟脉冲间选择。RTSECSEL (PADCFG1<1>) 位在这两个输出间选择。RTSECSEL = 0 时，选择闹钟脉冲。RTSECSEL = 1 时，选择秒时钟。

图 29-5: 定时器脉冲的产生



29.5 休眠模式

定时器和闹钟在休眠模式中也继续运行。闹钟的运行不受休眠影响，因为闹钟事件总能唤醒 CPU。

空闲模式不影响定时器或闹钟的运行。

29.6 复位

29.6.1 器件复位

发生器件复位时，ALCFGRPT 寄存器强制进入其复位状态，导致闹钟被禁止（如果复位前是使能的）。如果 RTCC 是使能的，闹钟在发生基本器件复位后将继续运行。

29.6.2 上电复位 (POR)

RCFGCAL 和 ALCFGRPT 寄存器只在 POR 时复位。器件退出 POR 状态后，时钟寄存器应重新装入所需值。

定时器预分频器值只能通过写入秒数寄存器复位。器件复位不会影响预分频器。

29.7 外设模块禁止 (PMD) 寄存器

外设模块禁止 (PMD) 寄存器提供了一种方法，可通过停止向其供应的所有时钟源来禁止 RTCC 模块。通过对应的 PMD 控制位禁止某个外设时，外设将处于最低功耗状态。与外设相关的控制和状态寄存器也会被禁止，所以读写这些寄存器的操作都无效。只有 PMDx 寄存器中的 RTCCMD 位清零时，才会使能外设模块。

29.8 寄存器映射

表 29-5 和表 29-6 中提供了与 PIC24F RTCC 模块相关的寄存器汇总。

表 29-5: 焊垫配置映射

寄存器名称	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时
PADCFG1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTSECSEL	PMPTTL	0000

图注： — = 未实现，读为 0。所示 100 引脚器件的复位值为十六进制。

表 29-6: 实时时钟和日历寄存器映射

寄存器名称	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时
ALRMVAL	基于 APTR<1:0> 的闹钟值寄存器窗口																xxxx
ALCFGRPT	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVAL	基于 RTCPTR<1:0> 的 RTCC 值寄存器窗口																xxxx
RCFGCAL ⁽¹⁾	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPT1	RTCPT0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	0000

图注： x = 复位时的未知值， — = 未实现，读为 0。所示复位值为十六进制。

注 1： RCFGCAL 寄存器复位值取决于复位类型。

29.9 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC24F 器件系列而编写的，但其概念是相关的，通过适当修改即可使用，但使用中可能会受到一定限制。当前与实时时钟和日历（RTCC）模块相关的应用笔记有：

标题

应用笔记编号

目前没有相关的应用笔记。

注： 如需获取更多 PIC24F 系列器件的应用笔记和代码示例，请访问 Microchip 网站（www.microchip.com）。

29.10 版本历史

版本 A（2006 年 4 月）

这是本文档的初始发行版。