

セクション 37. リアルタイム クロック / カレンダ (RTCC)

ハイライト

本セクションには下記の主要項目を記載しています。

37.1	はじめに	37-2
37.2	RTCC モジュールのレジスタ	37-3
37.3	RTCC の動作	37-11
37.4	RTCC アラーム	37-14
37.5	RTCC の校正	37-17
37.6	省電力モード時の動作	37-19
37.7	レジスタマップ	37-20
37.8	関連アプリケーション ノート	37-21
37.9	改訂履歴	37-22

37.1 はじめに

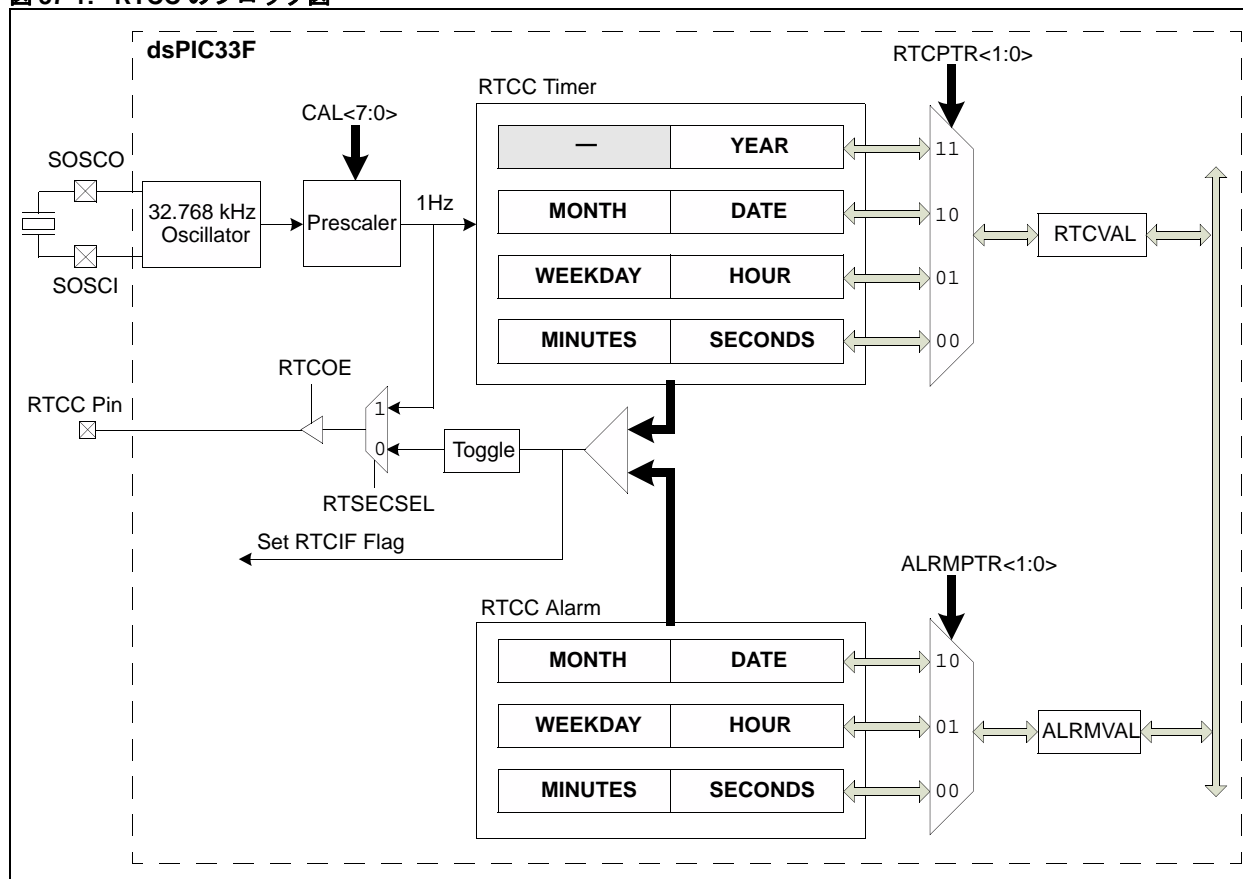
本書では、完全 BCD (Binary Coded Decimal) 対応クロック / カレンダを提供するリアルタイムクロック / カレンダ (RTCC) モジュールについて説明します。RTCC モジュールの主な機能は下記の通りです。

- 24 時間表示クロック
- 2099 年までの 100 年カレンダー
- 秒 / 分 / 時 / 曜日 / 日 / 月 / 年をカウント (うるう年を補正)
- 時刻、日付、アラームの BCD 表現
- プログラマブル アラーム (リピートモード付き)
- RTCC ピンでのアラーム出力または 1 Hz クロック出力による矩形波生成
- RTCC 校正

RTCC モジュールは、CPU に負荷を殆どあるいは全くかける事なく、デバイス上で動作するアプリケーションにタイム リファレンスを提供します。毎秒更新される現在の日付と時刻は 7 バイトのカウンタレジスタに格納されます。

RTCC とセカンダリ オシレータ (Sosc) は、 $\overline{\text{MCLR}}$ ピンが LOW に引き下げられてデバイスがリセット状態に保持されている時も動作を続けます。図 37-1 に RTCC モジュールのブロック図を示します。

図 37-1: RTCC のブロック図



37.2 RTCC モジュールのレジスタ

RTCC モジュールのレジスタは、下記の 3 種類に分類されます。

- **RTCC 制御レジスタ**
 - RCFGCAL: RTCC 校正 / コンフィグレーション レジスタ ⁽¹⁾
 - PADCFG1: ピン コンフィグレーション制御レジスタ
 - ALCFGRPT: アラーム コンフィグレーション レジスタ
- **RTCC 値レジスタ**
 - RTCVAL (RTCPTR<1:0> = 11 の時): 年値レジスタ
 - RTCVAL (RTCPTR<1:0> = 10 の時): 月 / 日値レジスタ
 - RTCVAL (RTCPTR<1:0> = 01 の時): 曜日 / 時値レジスタ
 - RTCVAL (RTCPTR<1:0> = 00 の時): 分 / 秒値レジスタ
- **アラーム値レジスタ**
 - ALRMVAL (ALRMPTR<1:0> = 10 の時): アラーム月 / 日値レジスタ
 - ALRMVAL (ALRMPTR<1:0> = 01 の時): アラーム曜日 / 時値レジスタ
 - ALRMVAL (ALRMPTR<1:0> = 00 の時): アラーム分 / 秒値レジスタ

37.2.1 レジスタの割り当て

RTCC タイマレジスタと RTCC アラームレジスタは、レジスタ インターフェイスを限定するため、対応するレジスタポイントを介してアクセスされます。RTCC 値レジスタの必要なレジスタペアを選択するために、RTCPTR ビット (RCFGCAL<9:8>) を使用します (表 37-1 参照)。ユーザ アプリケーションが RTCVAL レジスタの上位 8 ビットにアクセスするたびに、RTCPTR<1:0> ビット (RCFGCAL<9:8>) が自動的にデクリメントされます。一度「00」までデクリメントされると、それ以降ユーザ アプリケーションがこの上位 8 ビットにアクセスしても、RTCPTR ビットの値は変化しません。

表 37-1: RTCVAL レジスタの割り当て

RTCPTR <1:0>	RTCC 値レジスタ ウィンドウ	
	RTCVAL<15:8>	RTCVAL<7:0>
00	分	秒
01	曜日	時
10	月	日
11	—	年

RTCC アラーム値レジスタの必要なアラーム レジスタペアを選択するために、ALRMPTR ビット (ALCFGRPT<9:8>) を使用します (表 37-2 参照)。

ユーザ アプリケーションが ALRMVAL レジスタの上位 8 ビットにアクセスするたびに、ALRMPTR<1:0> ビット (ALCFGRPT<9:8>) が自動的にデクリメントされます。「00」までデクリメントされると、それ以降ユーザ アプリケーションがこの上位 8 ビットにアクセスしても、ALRMPTR ビットの値は変化しません。

表 37-2: ALRMVAL レジスタの割り当て

ALRMPTR <1:0>	アラーム値レジスタ ウィンドウ	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	分	秒
01	曜日	時
10	月	日
11	—	年

16 ビットコアは、8 ビット読み出し動作と 16 ビット読み出し動作を区別しないため、上位バイトまたは下位バイトのいずれを読み出しても、ALRMPTR<1:0> の値がデクリメントされます。これは RTCPTR<1:0> でも同様です。書き込み動作の場合、RTCVALH または ALRMVALH バイトに書き込んだ時にのみ、それぞれ RTCPTR または ALRMPTR レジスタの値がデクリメントされます。RTCVAL または ALRMVAL に書き込んだ場合、対応するポイントビットには影響しません。

Note: RTCVAL または ALRMVAL レジスタの値を MPLAB[®] IDE の「Watch」ウィンドウに表示すると、これらのレジスタがデクリメントされます。

37.2.2 RTCC 制御レジスタ

レジスタ 37- 1: RCFGAL: RTCC 校正 / コンフィグレーション レジスタ ⁽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	RTCPTR<1:0>	
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
CAL<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

- bit 15 **RTCEN:** RTCC イネーブルビット ⁽²⁾
 1 = RTCC モジュールを有効にする
 0 = RTCC モジュールを無効にする
- bit 14 **未実装:** 「0」として読み出し
- bit 13 **RTCWREN:** RTCC 値レジスタ書き込みイネーブルビット
 1 = ユーザ アプリケーションによる RTCVAL レジスタへの書き込みを許可する
 0 = ユーザ アプリケーションによる RTCVAL レジスタへの書き込みを禁止する
- bit 12 **RTCSYNC:** RTCC 値レジスタ読み出し同期ビット
 1 = 32 クロックエッジ (約 1ms) 以内にロールオーバーが発生する
 0 = ロールオーバーは発生しない
- bit 11 **HALFSEC:** 秒の前半 / 後半ステータスビット ⁽³⁾
 1 = 1 秒の後半
 0 = 1 秒の前半
- bit 10 **RTCOE:** RTCC 出力イネーブルビット
 1 = RTCC 出力を有効にする
 0 = RTCC 出力を無効にする
- bit 9-8 **RTCPTR<1:0>:** RTCC 値レジスタ ポインタビット
 このポインタビットは、RTCVAL レジスタの読み出し時に、どの RTCC 値レジスタを読み出すのかを指定します。RTCPTR<1:0> の値は、RTCVAL レジスタへのアクセスが発生するたびにデクリメントされます (「00」でデクリメントは停止)。このビットの詳細は 37.2.1 「レジスタの割り当て」を参照してください。
- bit 7-0 **CAL<7:0>:** RTCC ドリフト校正ビット
 01111111 = 正方向の最大調整値; 毎分 508 個の RTCC クロックパルスを追加する
 .
 .
 .
 00000001 = 正方向の最小調整値; 毎分 4 個の RTCC クロックパルスを追加する
 00000000 = 調整しない
 11111111 = 負方向の最小調整値; 毎分 4 個の RTCC クロックパルスを削減する
 .
 .
 .
 10000000 = 負方向の最大調整値; 毎分 512 個の RTCC クロックパルスを削減する

Note 1: RCFGAL レジスタは POR にのみ影響されます。

2: RTCEN ビットへの書き込みは、RTCWREN = 1 の時にのみ可能です。

3: このビットは読み出し専用です。このビットは MINSEC レジスタの下位バイトへの書き込み時に「0」にクリアされます。

セクション 37. リアルタイム クロック / カレンダ (RTCC)

レジスタ 37- 2: PADCFG1: ピン コンフィグレーション制御レジスタ

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⁾	PMPTTL
bit 7						bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-2 **未実装:** 「0」として読み出し

bit 1 **RTSECSEL:** RTCC 秒周期クロック出力選択ビット ⁽¹⁾

1 = RTCC 秒周期クロックを RTCC ピンで出力する

0 = RTCC アラームパルス を RTCC ピンで出力する

bit 0 **RTCC モジュールには使用せず。** このビットの詳細は各デバイスのデータシートを参照してください。

Note 1: RTCC 出力を有効化するには、RTCOE (RCFGCAL<10>) ビットをセットする必要があります。

37

リアルタイム
クロック / カレンダ
(RTCC)

dsPIC33F ファミリ リファレンス マニュアル

レジスタ 37- 3: ALCFGRPT: アラーム コンフィグレーション レジスタ

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	AMASK<3:0>				ALRMPTR<1:0>	
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
ARPT<7:0>							
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

- bit 15 **ALRMEN:** アラーム イネーブルビット
 1 = アラームを有効にする (ARPT<7:0> = 0x00 かつ CHIME = 0 の場合、このビットはアラームイベント後に自動的にクリアされます)
 0 = アラームを無効にする
- bit 14 **CHIME:** チャイム イネーブルビット
 1 = チャイムを有効にする (ARPT<7:0> ビットは 0x00 から 0xFF へロールオーバー可能)
 0 = チャイムを無効にする (ARPT<7:0> ビットは 0x00 で停止)
- bit 13-10 **AMASK<3:0>:** アラームマスク コンフィグレーション ビット
 0000 = 毎 1/2 秒
 0001 = 毎秒
 0010 = 毎 10 秒
 0011 = 毎分
 0100 = 毎 10 分
 0101 = 毎時
 0110 = 毎日
 0111 = 毎週
 1000 = 毎月
 1001 = 毎年 (2 月 29 日に設定した場合は 4 年に一度)
 101x = 予約済み
 11xx = 予約済み
- bit 9-8 **ALRMPTR<1:0>:** アラーム値レジスタ ウィンドウ ポインタビット
 このポインタは、ALRMVALH または ALRMVALL レジスタの読み出し時に、どのアラーム値レジスタを読み出すのかを指定します。ALRMPTR<1:0> の値は、ALRMVALH レジスタに対する読み書き動作が発生するたびにデクリメントされます (デクリメントは「00」で停止)。このビットの詳細は 37.2.1 「レジスタの割り当て」を参照してください。
- bit 7-0 **ARPT<7:0>:** アラーム リポートカウンタ値ビット
 11111111 = アラームを 255 回リポートする
 •
 •
 •
 00000000 = アラームをリポートしない
 カウンタはアラームイベントのたびにデクリメントします。CHIME = 1 ではない場合、カウンタは 0x00 から 0xFF へロールオーバーしません。

セクション 37. リアルタイム クロック / カレンダ (RTCC)

37.2.3 RTCC 値レジスタ

レジスタ 37- 4: RTCVAL (RTCPTR<1:0> = 11 の時): 年値レジスタ

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
YRTEN<3:0>				YRONE<3:0>			
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-8 **未実装:** 「0」として読み出し
 bit 7-4 **YRTEN<3:0>:** 「年」の 10 の桁の BCD 値; 0 ~ 9 を格納
 bit 3-0 **YRONE<3:0>:** 「年」の 1 の桁の BCD 値; 0 ~ 9 を格納

Note: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

レジスタ 37- 5: RTCVAL (RTCPTR<1:0> = 10 の時): 月 / 日値レジスタ

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MHTTEN0	MTHONE<3:0>			
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
—	—	DAYTEN<1:0>		DAYONE<3:0>			
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-13 **未実装:** 「0」として読み出し
 bit 12 **MHTTEN0:** 「月」の 10 の桁の BCD 値; 0 または 1 を格納
 bit 11-8 **MTHONE<3:0>:** 「月」の 1 の桁の BCD 値; 0 ~ 9 を格納
 bit 7-6 **未実装:** 「0」として読み出し
 bit 5-4 **DAYTEN<1:0>:** 「日」の 10 の桁の BCD 値; 0 ~ 3 を格納
 bit 3-0 **DAYONE<3:0>:** 「日」の 1 の桁の BCD 値; 0 ~ 9 を格納

Note: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

dsPIC33F ファミリ リファレンス マニュアル

レジスタ 37- 6: RTCVAL (RTCPTR<1:0> = 01 の時): 曜日 / 時値レジスタ

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY<2:0>		
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
—	—	HRTEN<1:0>		HRONE<3:0>			
bit 7		bit 0					

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-11 **未実装:** 「0」として読み出し
 bit 10-8 **WDAY<2:0>:** 「曜日」の BCD 値; 0 ~ 6 を格納
 bit 7-6 **未実装:** 「0」として読み出し
 bit 5-4 **HRTEN<1:0>:** 「時」の 10 の桁の BCD 値; 0 ~ 2 を格納
 bit 3-0 **HRONE<3:0>:** 「時」の 1 の桁の BCD 値; 0 ~ 9 を格納

Note: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

レジスタ 37- 7: RTCVAL (RTCPTR<1:0> = 00 の時): 分 / 秒値レジスタ

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN<2:0>			MINONE<3:0>			
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
—	SECTEN<2:0>			SECONE<3:0>			
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:** 「0」として読み出し
 bit 14-12 **MINTEN<2:0>:** 「分」の 10 の桁の BCD 値; 0 ~ 5 を格納
 bit 11-8 **MINONE<3:0>:** 「分」の 1 の桁の BCD 値; 0 ~ 9 を格納
 bit 7 **未実装:** 「0」として読み出し
 bit 6-4 **SECTEN<2:0>:** 「秒」の 10 の桁の BCD 値; 0 ~ 5 を格納
 bit 3-0 **SECONE<3:0>:** 「秒」の 1 の桁の BCD 値; 0 ~ 9 を格納

Note: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

セクション 37. リアルタイム クロック / カレンダ (RTCC)

37.2.4 アラーム値レジスタ

レジスタ 37- 8: ALRMVAL (ALRMPTR<1:0> = 10 の時): アラーム月 / 日値レジスタ

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE<3:0>			
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
—	—	DAYTEN<1:0>		DAYONE<3:0>			
bit 7							bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-13 **未実装:** 「0」として読み出し
 bit 12 **MTHTEN0:** 「月」の 10 の桁の BCD 値; 0 または 1 を格納
 bit 11-8 **MTHONE<3:0>:** 「月」の 1 の桁の BCD 値; 0 ~ 9 を格納
 bit 7-6 **未実装:** 「0」として読み出し
 bit 5-4 **DAYTEN<1:0>:** 「日」の 10 の桁の BCD 値; 0 ~ 3 を格納
 bit 3-0 **DAYONE<3:0>:** 「日」の 1 の桁の BCD 値; 0 ~ 9 を格納

Note: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

レジスタ 37- 9: ALRMVAL (ALRMPTR<1:0> = 01 の時): アラーム曜日 / 時値レジスタ

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY<2:0>		
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
—	—	HRTEN<1:0>		HRONE<3:0>			
bit 7		bit 0					

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15-11 **未実装:** 「0」として読み出し
 bit 10-8 **WDAY<2:0>:** 「曜日」の BCD 値; 0 ~ 6 を格納
 bit 7-6 **未実装:** 「0」として読み出し
 bit 5-4 **HRTEN<1:0>:** 「時」の 10 の桁の BCD 値; 0 ~ 2 を格納
 bit 3-0 **HRTONE<3:0>:** 「時」の 1 の桁の BCD 値; 0 ~ 9 を格納

Note: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

dsPIC33F ファミリ リファレンス マニュアル

レジスタ 37- 10: ALRMVAL (ALRMPTR<1:0> = 00 の時): アラーム分 / 秒値レジスタ

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN<2:0>			MINONE<3:0>			
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
—	SECTEN<2:0>			SECONE<3:0>			
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
-n = POR 時の値 1 = ビットをセット 0 = ビットをクリア x = ビットは未知

bit 15 **未実装:** 「0」として読み出し
bit 14-12 **MINTEN<2:0>:** 「分」の 10 の桁の BCD 値; 0 ~ 5 を格納
bit 11-8 **MINONE<3:0>:** 「分」の 1 の桁の BCD 値; 0 ~ 9 を格納
bit 7 **未実装:** 「0」として読み出し
bit 6-4 **SECTEN<2:0>:** 「秒」の 10 の桁の BCD 値; 0 ~ 5 を格納
bit 3-0 **SECONE<3:0>:** 「秒」の 1 の桁の BCD 値; 0 ~ 9 を格納

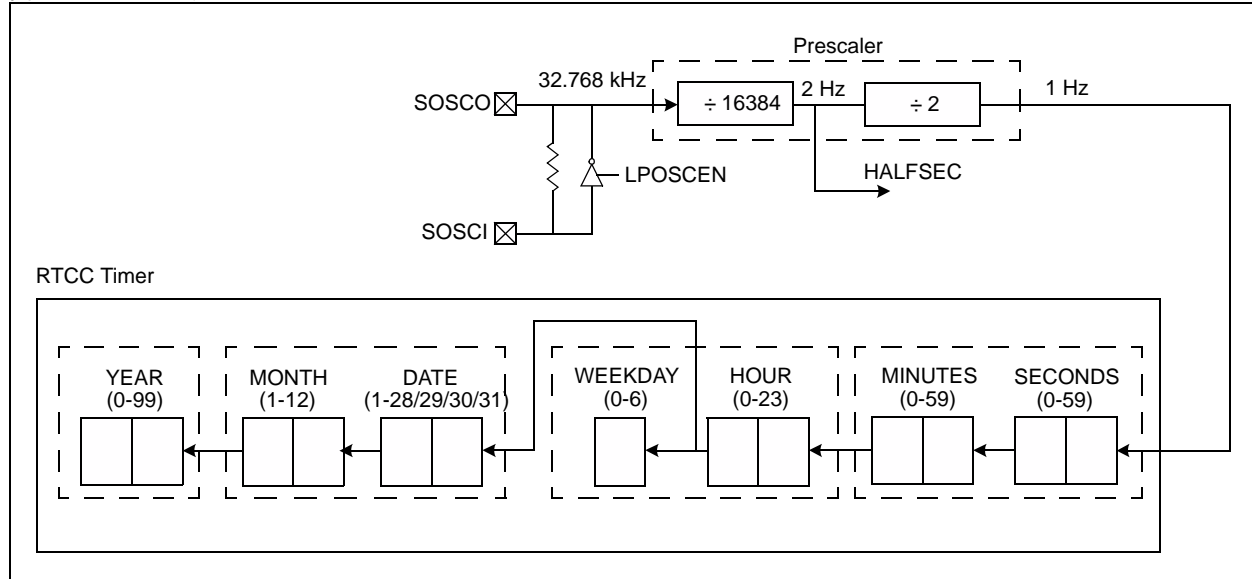
Note: このレジスタへの書き込みは、RTCWREN = 1 の時にのみ可能です。

37.3 RTCC の動作

RTCC モジュールは、32.768 kHz の外部リアルタイムクロック水晶振動子によるクロック供給を前提として設計されています。プリスケールで水晶振動子の発振周波数を分周する事により、クロック / カレンダの更新周波数 (1 Hz) が生成されます。毎秒更新される現在の日付と時刻は、7 バイトのカウンタレジスタに格納されます。

各カウンタは BCD 形式で値をカウントするため、表示または印刷用の 10 進表記に容易に変換できます。各バイトカウンタのカウントシーケンスを図 37-2 に示します。「日」と「月」は「1」にロールオーバーする事に注意してください。その他のカウンタは全て「0」にロールオーバーします。各月の最終「日」は自動的に 28/29/30/31 のいずれかに調整されます (2100 年までの「うるう年」の修正も含む)。初期電源投入時の各カウンタの値はランダムです。

図 37-2: RTCC タイマ



Note: セカンダリ水晶振動子から RTCC モジュールにクロックを供給するために、オシレータ制御 (OSCCON<1>) レジスタのセカンダリ オシレータ イネーブル (LPOSCEN) ビットをセットする必要があります。詳細はセクション 39.「オシレータ (パート III)」(DS70216) を参照してください。

37.3.1 RTCC タイマへの書き込み

ユーザアプリケーションは、各 RTCC レジスタに秒 / 分 / 時 / 曜日 / 日 / 月 / 年の値を書き込む事によって時刻と日付を設定できます。通常動作時に RTCC タイマレジスタに書き込む事はできません。書き込みを実行した場合、正常に実行されたかのように見えますが、実際にはレジスタの内容は変更されません。RTCC レジスタに書き込むには、RTCWREN ビット (RCFGCAL<13>) をセットする必要があります。RTCWREN ビットをセットすると、RTCC レジスタへの書き込みが許可されます。RTCWREN ビットをクリアすると、RTCC レジスタへの書き込みは禁止されます。

RTCWREN ビットをセットするには、下記の特異なコマンドシーケンスを実行する必要があります (クリアはいつでもできます)。

1. 0x55 を NVMKEY に書き込む
2. 0xA5 を NVMKEY に書き込む
3. 1 サイクル命令を使用して RTCWREN ビットをセットする

RTCC モジュールを有効化するには、RTCEN ビット (RCFGCAL<15>) をセットする必要があります。RTCEN ビットをセット / クリアする場合も、RTCWREN ビット (RCFGCAL<13>) をセットする必要があります。

時刻全体 (時 / 分 / 秒) を変更する必要がある場合、RTCC タイマレジスタへの書き込み時に RTCC モジュールを無効化して、クロックのカウントを停止する事を推奨します。これにより、タイマのインクリメントと書き込み動作の同時発生を回避します。

時 / 分 / 秒のいずれか 1 つだけを変更する場合、クロックのカウントを停止する必要はありません。停止すると計時に誤差が生じる可能性があります。1 つのフィールドだけ変更する例として、サマータイム用に「時」だけを修正する場合があります。「分」または「秒」レジスタへの書き込み時にもプリスケアラがリセットします。

クロックのカウントを停止せずにこれらの計時レジスタに書き込む場合、RTCSYNC ビット (RCFGCAL<12>) が「0」になるまで待機してから書き込むのが最良の方法です。RTCSYNC ビットは、プリスケアラのロールオーバー発生予定時期より 32 クロックエッジ (約 1 ms) 前にセットされます。RTCC 計時レジスタへの書き込み方法を示すサンプルコードを例 37- 1 に記載します。

例 37- 1: RTCC 計時レジスタへの書き込み

```
;*****
; Enable RTCC Timer Access
;*****
MOV #0x55,W0
MOV W0, NVMKEY
MOV #0xAA,W0
MOV W0, NVMKEY
BSET RCFGCAL,#RTCWREN ; Set RTCWREN bit

;*****
; Disable RTCC module
;*****
BCLR RCFGCAL,#RTCEN ; Clear RTCEN bit

;*****
; Write to RTCC Timer
;*****
MOV RCFGCAL,w0
OR #0x300,w0
MOV w0,RCFGCAL ; Set RTCPTR to 3

MOV #0x0007,w0 ; Set Year (#0x00YY)
MOV #0x1028,w1 ; Set Month and Day (#0xMMDD)
MOV #0x0110,w2 ; Set Weekday and Hour (#0x0WHH)
MOV #0x0000,w3 ; Set Minute and Second (#0xMMSS)

MOV w0,RTCVAL
MOV w1,RTCVAL
MOV w2,RTCVAL
MOV w3,RTCVAL

;*****
; Enable RTCC module
;*****
BSET RCFGCAL,#RTCEN ; Set RTCEN bit

;*****
; Disable RTCC Timer Access
;*****
BCLR RCFGCAL,#RTCWREN ; Clear RTCWREN bit
```

Note: RTCC レジスタへの書き込み時にはアラームを無効化する (ALRMEN = 0) 必要があります。これを無効化しないと、アラームが正常に動作しない可能性があります。

37.3.2 RTCC タイマの読み出し

必要な時刻 / 日付情報を取得するには、適切なレジスタバイトを読み出す必要があります。現在の時刻を読み出す際に RTCC タイマを停止する事はできません。停止すると計時に誤差が生じる可能性があります。従って RTCC タイマは動作中に読み出す必要があります。

カウンタのインクリメントと読み出し動作は非同期であるため、インクリメント中(ロールオーバー中)にカウンタが読み出され、結果として読み出された時刻が不正確になる可能性があります。クロック カウンタの全内容(または一部の内容)を正確に読み出すには、読み出し中にカウンタのロールオーバーが発生しない事が必要です。

クロックのカウントを停止せずに計時レジスタを正確に読み出すには、RTCSYNC ビット(RCFGAL<12>)が「0」になるまで待機してから読み出すのが最良の方法です。RTCSYNC ビットは、プリスケアラのロールオーバー発生予定時期より 32 クロックエッジ(約 1 ms)前にセットされます。

RTCSYNC ビットが「0」になるまでユーザ アプリケーションが待機できない場合、計時レジスタを 2 回読み出す事によって対処できます。2 回読み出したデータが同じであれば、そのデータは有効であるとみなせます。従ってこの方法では最低 2 回の読み出しが必要であり、2 回の読み出しでデータが一致しなければ最高 3 回の読み出しが必要です。RTCC 計時レジスタの読み出し方法を示すサンプルコードを例 37- 2 に記載します。

例 37- 2: RTCC 計時レジスタの読み出し

```
*****
; Wait for RTCSYNC bit to become '0'
*****
while(RCFGALbits.RTCSYNC==1);

*****
; Read RTCC timekeeping register
*****
RCFGALbits.RTCPTR=3;

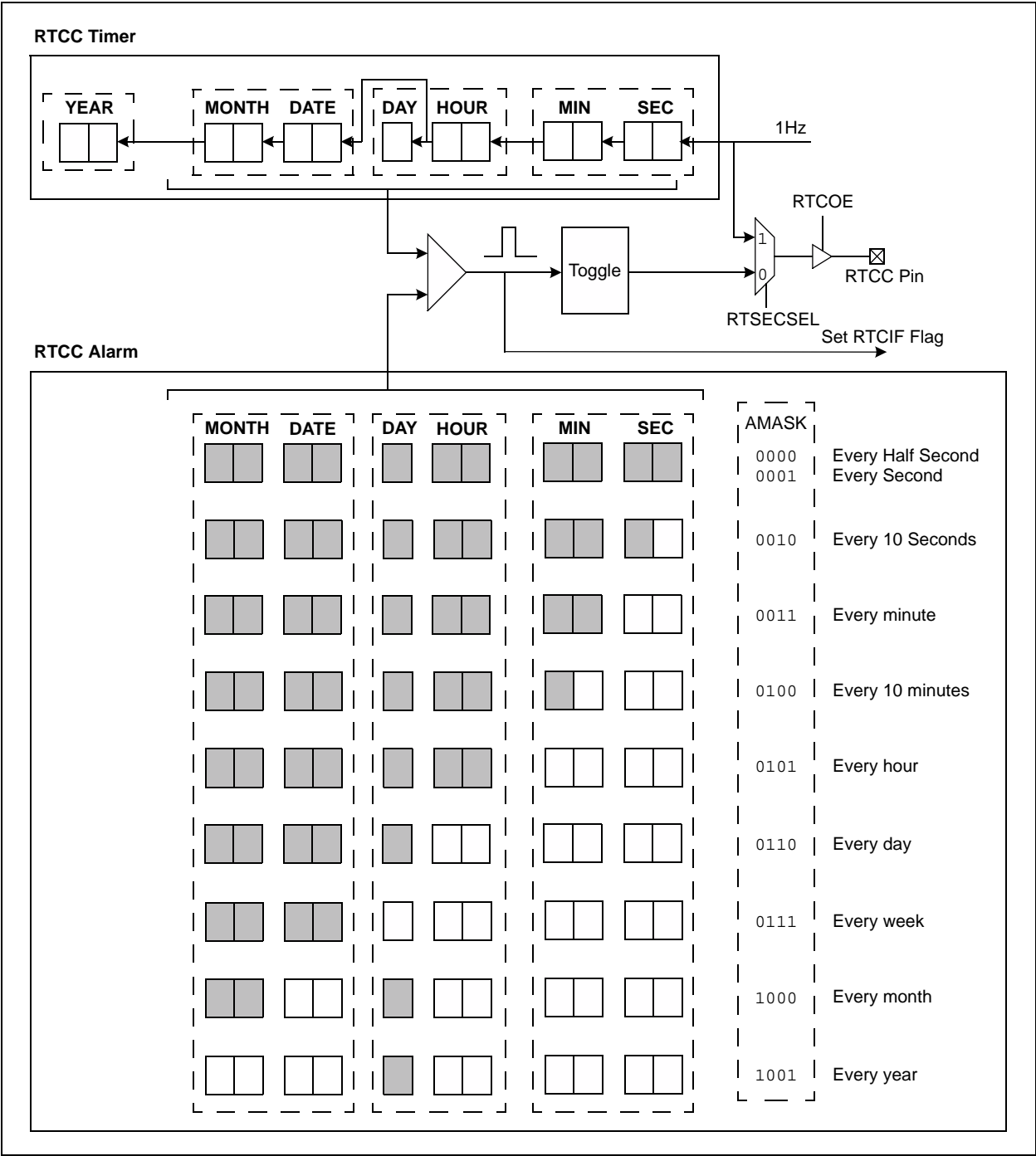
year=RTCVAL;
month_date=RTCVAL;
wday_hour=RTCVAL;
min_sec=RTCVAL;
```

37.4 RTCC アラーム

アラームを使用すると、特定時刻または特定周期 (毎分、毎日等) で CPU に割り込む事ができます。RTCC は、毎回のクロック更新時に、選択されたアラームレジスタと、これに対応するクロックレジスタを比較します。両者が一致した時にアラームイベントが発生します。

アラーム コンフィグレーション (ALCFGRPT<13:10>) レジスタのアラームマスク (AMASK<3:0>) ビットは、比較から除外するレジスタをマスクするために使用します。図 37-3 に示すように、AMASK の設定に応じて、各種アラーム BCD レジスタニブルが計時レジスタとの比較からマスクされます。

図 37-3: RTCC アラーム



セクション 37. リアルタイム クロック / カレンダ (RTCC)

例として、アラームが毎朝 6 時 (24 時間表記で 06:00:00) に発生するように設定する方法を説明します。

1. アラームレジスタの「時」バイトに 06 (BCD)、「分」バイトに 00 (BCD)、「秒」バイトに 00 (BCD) を書き込みます。
2. 「月」/「日」/「曜日」バイトをマスクして「時」/「分」/「秒」だけを比較するために、AMASK ビットを「0b0110」に設定します。
以上により、時刻が 05:59:59 からロールオーバーするたびに毎日アラームイベントが発生します。

Note: アラームレジスタに書き込む時にアラームを無効化する (ALRMEN = 0) 必要があります。無効化しないと、アラームが正しく動作しない可能性があります。アラームレジスタの読み出しはいつでも可能です。

アラームレジスタを設定して毎朝 6 時にアラームを生成する方法を示すサンプルコードを例 37-3 に記載します。

例 37-3: アラームレジスタへの書き込み

```
;*****  
; Disable Alarm  
;*****  
ALCFGRPTbits.ALRMEN=0;  
  
;*****  
; Write to Alarm Register (Time:06:00:00)  
;*****  
ALCFGRPTbits.ALMPTR=2;  
  
ALRMVAL=0;  
ALRMVAL=0x0006;  
ALRMVAL=0x0000;  
  
;*****  
; Select Alarm Mask to compare hour, minute, second  
;*****  
ALCFGRPTbits.AMASK=0b0110;  
  
;*****  
; Enable Alarm  
;*****  
ALCFGRPTbits.CHIME=1;  
ALCFGRPTbits.ALRMEN=1;
```

37

リアルタイム
クロック / カレンダ
(RTCC)

37.4.1 アラームモードの選択

アラームの有効化には ALRMEN ビット (ALCFGRPT<14>) を使用します。これらのビット設定により、単発アラームとリピート アラームを選択できます。

37.4.1.1 単発アラーム

このアラームイベントは、クロックが選択されたアラームニブルに一致した時に発生します。アラームイベントが発生すると、アラームは自動的に無効化されます。単発アラームの設定方法は下記の通りです。

1. アラーム コンフィグレーション (ALCFGRPT<14>) レジスタのチャイム イネーブルビット (CHIME) をクリアする
2. アラーム リピートカウンタ (ARPT<7:0>) に「0」を書き込む

37.4.1.2 チャイムの無効化

CHIME = 0 の時、アラーム リピートカウンタ (ARPT<7:0>) はアラームイベントが発生するたびにデクリメントされ、このカウンタが「0」までデクリメントされるとアラームが無効化されます。

37.4.1.3 チャイムの有効化

チャイム イネーブル (CHIME) ビットをセットすると、アラームを無限にリピートできます。CHIME = 1 の時、アラーム リピートカウンタ (ARPT<7:0>) はアラームが発生するたびに無限にデクリメントされます。

37.4.2 アラームによる割り込みと出力

アラームイベント発生時に RTCC 割り込みを生成するか、RTCC 出力ピンをトグルする事ができます。各 RTCC 割り込みイネーブル (RTCIE) ビットを使用して、RTCC 割り込みを割り込み要因として有効化できます。割り込み要因となる RTCC に対応する割り込み優先度ビット (RTCIP<2:0>) にゼロ以外の値を書き込む必要があります。詳細は[セクション 32.「割り込み \(パート III\)」](#) (DS70214) を参照してください。

また、アラームイベントが発生するたびに RTCC ピンをトグルする事により、アラームの 2 倍周期のクロックを生成する事ができます。RTCC ピンで秒周期クロック (1 Hz のクロック) を出力する事もできます。ユーザ アプリケーションは、アラーム出力または秒周期クロック出力のいずれかを選択できます。この選択には RTSESEL (PADCFG1<1>) ビットを使用します。RTSESEL = 0 の時、アラームイベントが発生するたびにこのピンがトグルされます。RTSESEL = 1 の時、秒周期クロックがこのピンに出力されます。

37.5 RTCC の校正

校正により、水晶振動子の公称周波数からの偏差と外部水晶振動子周波数の温度依存性を補正できます。毎分のカウント数を増減する事によって校正を行います。カウントを追加するとクロック速度が増加し、カウントを削減するとクロック速度が低下します。間引きパルス数 (カウント削減数: 負方向の調整) または挿入パルス数 (カウント追加数: 正方向の調整) は、RTCC コンフィグレーション/校正制御 (RCFGCAL<7:0>) レジスタの校正ビット (CAL<7:0>) に 8 ビット符号付き値を書き込む事によって設定します。

校正ステップを 1 段階増減すると、毎分のオシレータ サイクル数が 4 ずつ増減します (公称値: $60 \times 32.768 \text{ kHz} = 1,966,080$ オシレータ サイクル)。1 校正ステップあたりの調整量は $\pm 2.034 \text{ ppm}$ ($\pm 5.35 \text{ s/month}$) です。これにより、計時精度を 3 s/month 以内に校正する事ができます。表 37-3 に、校正レジスタの値と調整量の関係を示します。

表 37-3: 校正レジスタの値と調整量の関係

校正値 (CAL<7:0>)	調整精度 (ppm)	調整時間 (sec/month)	校正値 (CAL<7:0>)	調整精度 (ppm)	調整時間 (sec/month)
0	0	—	-1	-2.03	-5.27
1	2.03	5.27	-2	-4.07	-10.55
2	4.07	10.55	-3	-6.1	-15.82
3	6.1	15.82	-4	-8.14	-21.09
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
125	254.31	659.18	-126	-256.35	-664.45
126	256.35	664.45	-127	-258.38	-669.73
127	258.38	669.73	-128	-260.42	-675

アプリケーションの必要調整量は下記の方法で特定できます。この方法は特に生産現場での実施に適します。

1. 秒周期 (1 Hz) クロックを RTCC ピンで出力して水晶振動子の周波数を計測し、公称周波数からの偏差 (Hz) を求めます。
2. 周波数偏差 = $32.768 \text{ kHz} - \text{実測周波数}$
3. 上記で求めた周波数偏差を毎分のカウント数偏差に換算します。

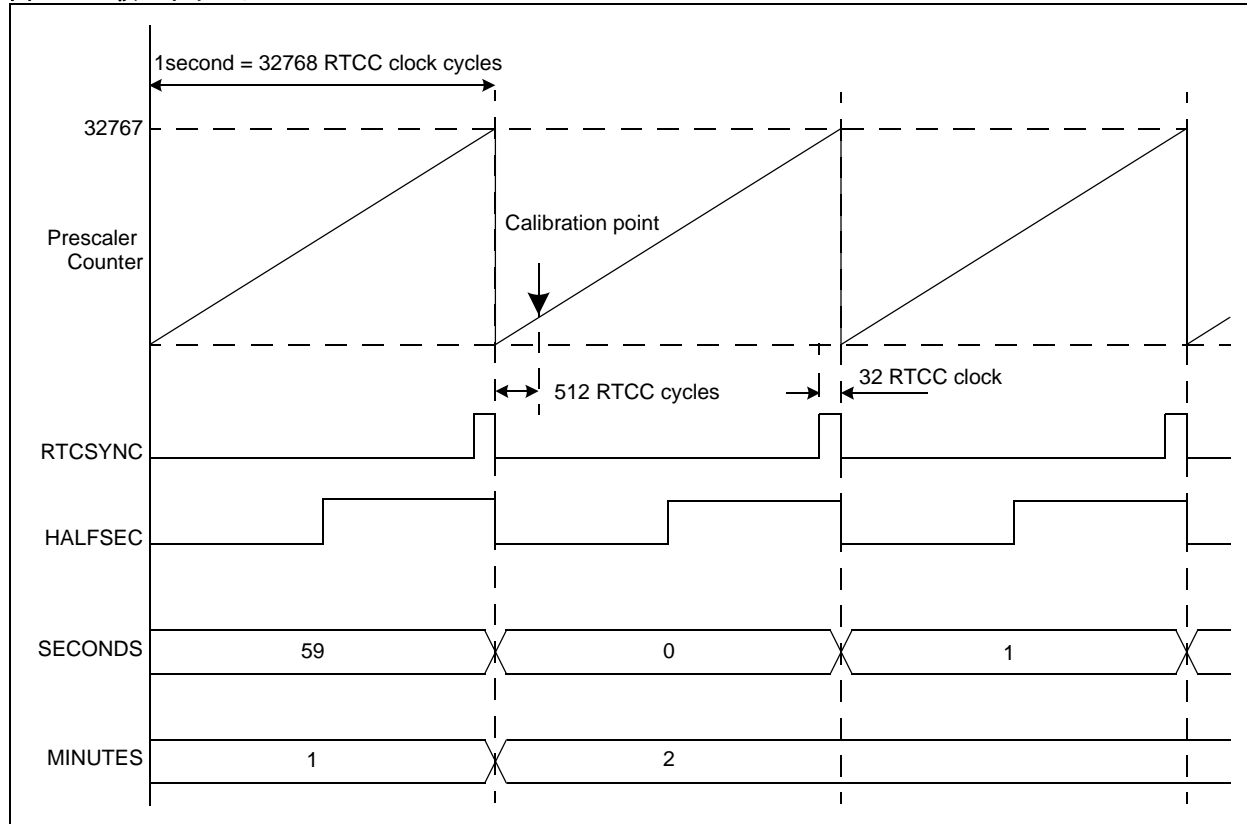
$$\text{カウント数偏差} = (\text{周波数偏差}) \times 60$$
4. CAL<7:0> ビットを (カウント数偏差 /4) に設定します。

Note: 温度センサを使用すると、ユーザ アプリケーションで校正値を調整する事によって、温度ドリフトの影響を補正できます。

37.5.1 校正値の書き込み

RTCC モジュールは、校正（プリスケアラの調整）を毎分実行します（図 37-4 参照）。これは「秒」BCD カウンタのロールオーバー（59 → 00）から 512 番目のクロックエッジ（校正ポイント）で実行されます。

図 37-4: 校正ポイント



校正ポイントにおいて、RTCC モジュールは 8 ビット符号付き値 (CAL<7:0>) を読み出し、これに基づいてプリスケアラ カウンタを調整します。RTCC モジュールが正しい校正値を確実に読み出すには、RTCC モジュールによる値の読み出しと CPU による値の更新が同時に発生しない必要があります。「秒」バイトがゼロである場合、HALFSEC (RCFGCAL<11>) が「1」に変化した時に校正値を更新する必要があります。RTCC の動作中に校正値を更新する方法を示すサンプルコードを例 37-4 に記載します。

例 37-4: 校正値の書き込み

```

;*****
; Read RTCC timer to check the seconds
; If seconds byte is 0, wait for HALFSEC bit to become 1
;*****
RCFGCALbits.RTCPTR=0;
seconds=(RTCVAL & 0xFF);

if(seconds == 0)
    while(RCFGCALbits.HALFSEC==0);

;*****
; Write to the calibration value
;*****
RCFGCALbits.CAL=calvalue;
    
```

37.6 省電力モード時の動作

37.6.1 スリープモード

デバイスがスリープモードに移行するとシステムクロックが停止します。RTCC タイマはにセカンダリ オシレータからクロックが供給されるため、スリープモードでも動作を続けます。

有効化されたアラーム割り込みが発生すると、デバイスがスリープからウェイクアップし、下記のように動作します。

- その割り込みに割り当てられた優先度が現在の CPU 優先度以下である場合、ウェイクアップしたデバイスはスリープモードを起動した PWRSAV 命令の次の命令からコード実行を再開します。
- 割り込み優先度が現在の CPU 優先度よりも高い場合、ウェイクアップしたデバイスは CPU 例外処理を開始します。この場合、タイマ割り込みサービスルーチン (ISR) の先頭命令からコード実行を再開します。詳細はセクション 9.「ウォッチドッグ タイマと省電力モード」(DS70196) を参照してください。

37.6.2 アイドルモード

アイドルモードは RTCC モジュールの動作に影響しません。

37.7 レジスタマップ

表 37-4 に、RTCC レジスタのビット割り当てを示します。

表 37-4: リアルタイム クロック / カレンダ関連のレジスタマップ

レジスタ名	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	全 リセット
ALRMVAL	ALRMPTR<1:0> に基づくアラーム値レジスタ ウィンドウ																xxxx
ALCFGRPT	ALRMEN	CHIME	AMASK<3:0>				ALRMPTR<1:0>		ARPT<7:0>								0000
RTCVAL	RTCPTR<1:0> に基づく RTCC 値レジスタ ウィンドウ																xxxx
RCFGCAL	RTCEN	—	RTCWREN	RTCSYNC	HALFSE C	RTCOE	RTCPTR<1:0>		CAL<7:0>								0000
PADCFG1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTSECSEL	PMP TTL	0000

凡例: x = リセット時に未知の値、— = 未実装、「0」として読み出し、リセット値は 16 進数で表記

セクション 37. リアルタイム クロック / カレンダ (RTCC)

37.8 関連アプリケーション ノート

本セクションに関連するアプリケーション ノートの一覧を下に記載します。一部のアプリケーション ノートは dsPIC33F ファミリ向けではありません。ただし概念は共通しており、変更が必要であったり制限事項が存在するものの利用が可能です。リアルタイム クロック / カレンダ (RTCC) モジュールに関連する最新のアプリケーション ノートは下記の通りです。

タイトル

アプリケーション ノート番号

現在、関連するアプリケーション ノートはありません。

N/A

Note: dsPIC33F ファミリ関連のアプリケーション ノートとサンプルコードはマイクロチップ社のウェブサイト (www.microchip.com) でご覧頂けます。

37

リアルタイム
クロック / カレンダ
(RTCC)

37.9 改訂履歴

リビジョン A (2007 年 11 月)

本書の初版

リビジョン B (2011 年 7 月)

このリビジョンでの変更内容は以下の通りです。

- レジスタ :
 - RCFGCAL: RTCC 校正 / コンフィグレーション レジスタ (レジスタ 37- 1 参照) の bit 7-0 に関する記述を更新
- セクション :
 - 37.1 「はじめに」内の記述を下記に更新 :
RTCC とセカンダリ オシレータ (Sosc) は、MCLR ピンが LOW に引き下げられてデバイスがリセット状態に保持されている間も動作を続けます。
- 上記に加えて、表現および体裁の変更等、本書全体の細部を修正

ISBN: 978-1-61341-042-4