Section 37. Real-Time Clock and Calendar (RTCC)

HIGHLIGHTS

This section of the manual contains the following major topics:

37.1 INTRODUCTION

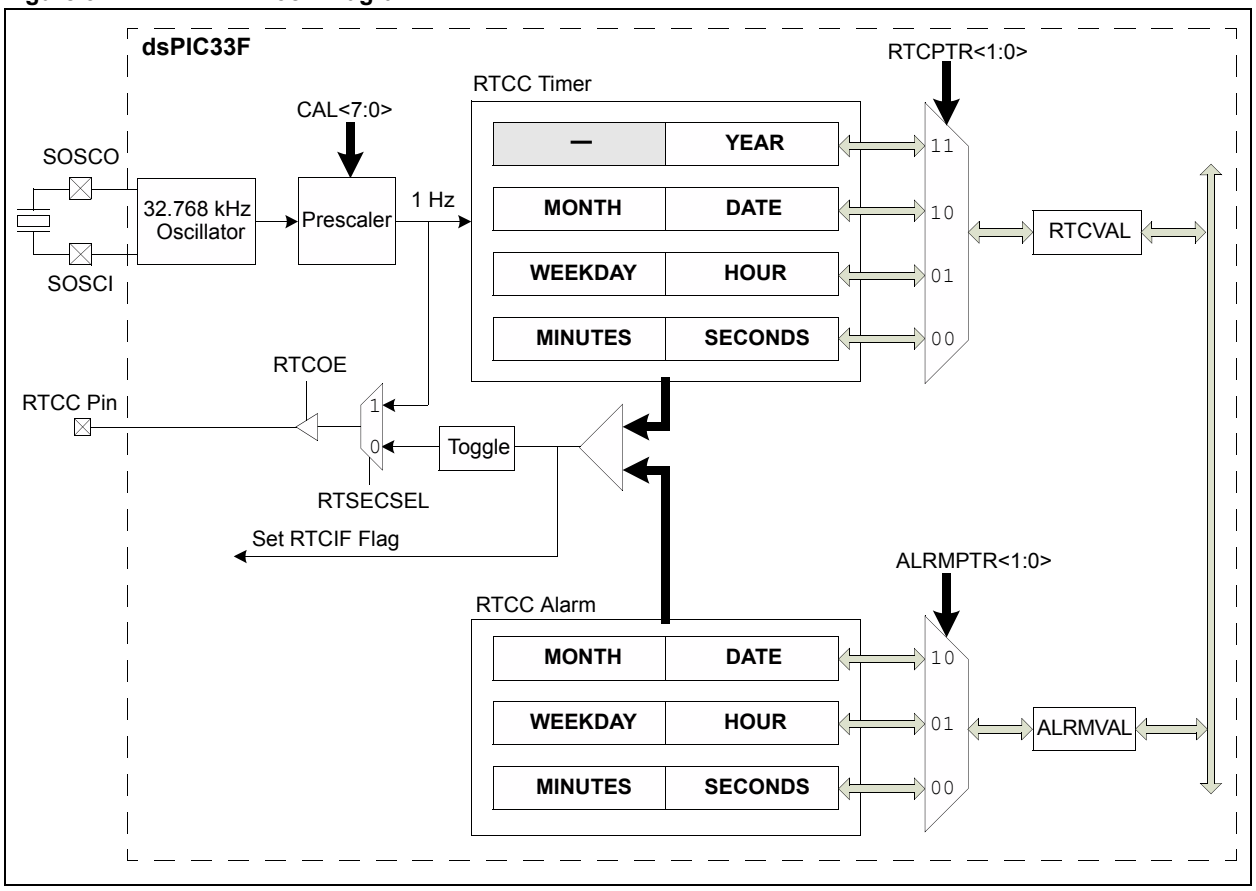
This section discusses the Real-Time Clock and Calendar (RTCC) module that provides a full Binary-coded Decimal (BCD) clock calendar. Listed below are the some of the key features of the RTCC module:

- 24-hour (military time) clock
- 100-year calendar up to year 2099
- Counts seconds, minutes, hours, weekday, date, month and year with leap year compensation
- BCD representation of time, calendar and alarm
- Programmable Alarm with repeat mode
- Square wave generation using Alarm or 1 Hz Clock Output on RTCC pin
- RTCC Calibration

The RTCC module provides a time reference to an application running on the device with minimum to no intervention from the CPU. The current date and time is tracked in a set of counter registers that update once per second.

Figure 37-1 shows a block diagram of the RTCC module.

Figure 37-1: RTCC Block Diagram



37.2 RTCC MODULE REGISTERS

The RTCC module registers are organized into three categories:

- **RTCC Control Registers**
  - RCFGCAL: RTCC Calibration and Configuration Register
  - PADCFG1: Pad Configuration Control Register
  - ALCFGRPT: Alarm Configuration Register
- **RTCC Value Registers**
  - RTCVAL (when RTCPTR<1:0> = 11): Year Value Register
  - RTCVAL (when RTCPTR<1:0> = 10): Month and Day Value Register
  - RTCVAL (when RTCPTR<1:0> = 01): Weekday and Hours Value Register
  - RTCVAL (when RTCPTR<1:0> = 00): Minutes and Seconds Value Register
- **Alarm Value Registers**
  - ALRMVAL (when ALRMPTR<1:0> = 10): Alarm Month and Day Value Register
  - ALRMVAL (when ALRMPTR<1:0> = 01): Alarm Weekday and Hours Value Register
  - ALRMVAL (when ALRMPTR<1:0> = 00): Alarm Minutes and Seconds Value Register

37.2.1 Register Mapping

To limit the register interface, the RTCC Timer and RTCC Alarm registers are accessed through corresponding register pointers. The RTCC Value register uses the RTCPTR bits (RCFGCAL<9:8>) to select the desired timer register pair (see Table 37-1).

The RTCPTR<1:0> bits (RCFGCAL<9:8>) are automatically decremented each time the upper 8 bits of the RTCVAL register are accessed by the user application. Once they reach a value of '00', any further access of these upper 8 bits by the user application has no effect on the RTCPTR bits.

Table 37-1: RTCVAL Register Mapping

RTCPTR <1:0>	RTCC Value Register Window	
	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

The RTCC Alarm Value register uses the ALRMPTR bits (ALCFGRPT<9:8>) to select the desired Alarm register pair (see Table 37-2).

The ALRMPTR<1:0> bits (ALCFGRPT<9:8>) are automatically decremented each time the upper 8 bits of the ALRMVAL register are accessed by the user application. Once they reach a value of '00', any further access of these upper 8 bits by the user application has no effect on the ALRMPTR bits.

Table 37-2: ALRMVAL Register Mapping

ALRMPTR <1:0>	Alarm Value Register Window	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

Considering that the 16-bit core does not distinguish between 8-bit and 16-bit read operations, reading either the upper or lower bytes decrements the ALRMPTR<1:0> value. The same applies to the RTCPTR<1:0> value. While writing, the RTCPTR and ALRMPTR register value will decrement only when writing to the RTCVALH and ALRMVALH bytes, respectively. Writing to the RTCVALL and ALRMVALL does not affect the corresponding pointer bits.

Note: Displaying RTCVAL or ALRMVAL Registers in the MPLAB IDE watch window will cause these Registers to decrement.

37.2.2 RTCC Control Registers

Register 37-1: RCFGCAL: RTCC Calibration and Configuration Register

R/W-0	U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
RTCEN(1)	—	RTCWREN	RTCSYNC	HALFSEC(2)	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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **RTCEN:** RTCC Enable bit(1)
1 = RTCC module is enabled
0 = RTCC module is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **RTCWREN:** RTCC Value Registers Write Enable bit
1 = RTCVAL register can be written to by the user application
0 = RTCVAL register is locked out from being written to by the user application

bit 12 **RTCSYNC:** RTCC Value Registers Read Synchronization bit
1 = A rollover is about to occur in 32 clock edges (approximately 1 ms)
0 = A rollover will not occur

bit 11 **HALFSEC:** Half-Second Status bit(2)
1 = Second half period of a second
0 = First half period of a second

bit 10 **RTCOE:** RTCC Output Enable bit
1 = RTCC output enabled
0 = RTCC output disabled

bit 9-8 **RTCPTR<1:0>:** RTCC Value Register Pointer bits
Points to the corresponding RTCC Value register when reading the RTCVAL register; the RTCPTR<1:0> value decrements on every access of RTCVAL until it reaches '00'.
See Section 37.2.1 for bit description.

Note 1: A write to the RTCEN bit is only allowed when RTCWREN = 1.

2: This bit is read-only. It is cleared to '0' on a write to the lower half of the MINSEC register.

Note: The RCFGCAL register is only affected by a POR.

Register 37-1: RCFGCAL: RTCC Calibration and Configuration Register (Continued)

bit 7-0 **CAL<7:0>:** RTCC Drift Calibration bits

01111111 = Maximum positive adjustment; adds 508 RTCC clock pulses every one minute
•
•
•
01111111 = Minimum positive adjustment; adds 4 RTCC clock pulses every one minute
00000000 = No adjustment
11111111 = Minimum negative adjustment; subtracts 4 RTCC clock pulses every one minute
•
•
•
10000000 = Maximum negative adjustment; subtracts 512 RTCC clock pulses every one minute

Note 1: A write to the RTCEN bit is only allowed when RTCWREN = 1.

2: This bit is read-only. It is cleared to '0' on a write to the lower half of the MINSEC register.

Note: The RCFGCAL register is only affected by a POR.

Register 37-2: PADCFG1: Pad Configuration Control Register

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as '0'

bit 1 **RTSECSEL:** RTCC Seconds Clock Output Select bit(1)
1 = RTCC seconds clock is selected for the RTCC pin
0 = RTCC alarm pulse is selected for the RTCC pin

bit 0 Not used by the RTCC module. (See device data sheet for description of this bit).

Note 1: To enable the actual RTCC output, the RTCOE (RCFGCAL<10>) bit needs to be set.

Register 37-3: ALCFGRPT: Alarm Configuration Register

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ALRMEN:** Alarm Enable bit

1 = Alarm is enabled (cleared automatically after an alarm event whenever ARPT<7:0> = 00h and CHIME = 0)

0 = Alarm is disabled

bit 14 **CHIME:** Chime Enable bit

1 = Chime is enabled; ARPT<7:0> bits are allowed to roll over from 00h to FFh

0 = Chime is disabled; ARPT<7:0> bits stop once they reach 00h

bit 13-10 **AMASK<3:0>:** Alarm Mask Configuration bits

0000 = Every half second
0001 = Every second
0010 = Every 10 seconds
0011 = Every minute
0100 = Every 10 minutes
0101 = Every hour
0110 = Once a day
0111 = Once a week
1000 = Once a month
1001 = Once a year (except when configured for February 29, once every 4 years)
101x = Reserved
11xx = Reserved

bit 9-8 **ALRMPTR<1:0>:** Alarm Value Register Window Pointer bits

Points to the corresponding Alarm Value registers when reading ALRMVALH and ALRMVALL registers; the ALRMPTR<1:0> value decrements on every read or write of ALRMVALH until it reaches '00'.

See Section 37.2.1 for bit description.

bit 7-0 **ARPT<7:0>:** Alarm Repeat Counter Value bits

11111111 = Alarm will repeat 255 more times

•
•
•

00000000 = Alarm will not repeat

The counter decrements on any alarm event. The counter is prevented from rolling over from 00h to FFh unless CHIME = 1.

37.2.3 RTCC Value Registers

Register 37-4: RTCVAL (when RTCPTR<1:0> = 11): Year Value Register

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7-4 **YRTEN<3:0>:** Binary Coded Decimal Value of Year's Tens Digit; contains a value from 0 to 9

bit 3-0 **YRONE<3:0>:** Binary Coded Decimal Value of Year's Ones Digit; contains a value from 0 to 9

Note: A write to this register is only allowed when RTCWREN = 1.

Register 37-5: RTCVAL (when RTCPTR<1:0> = 10): Month and Day Value Register

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **MTHTEN0:** Binary Coded Decimal Value of Month's Tens Digit; contains a value of 0 or 1

bit 11-8 **MTHONE<3:0>:** Binary Coded Decimal Value of Month's Ones Digit; contains a value from 0 to 9

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day's Tens Digit; contains a value from 0 to 3

bit 3-0 **DAYONE<3:0>:** Binary Coded Decimal Value of Day's Ones Digit; contains a value from 0 to 9

Note: A write to this register is only allowed when RTCWREN = 1.

Register 37-6: RTCVAL (when RTCPTR<1:0> = 01): Weekday and Hours Value Register

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit; contains a value from 0 to 6

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour's Tens Digit; contains a value from 0 to 2

bit 3-0 **HRONE<3:0>:** Binary Coded Decimal Value of Hour's Ones Digit; contains a value from 0 to 9

Note: A write to this register is only allowed when RTCWREN = 1.

Register 37-7: RTCVAL (when RTCPTR<1:0> = 00): Minutes and Seconds Value Register

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute's Tens Digit; contains a value from 0 to 5

bit 11-8 **MINONE<3:0>:** Binary Coded Decimal Value of Minute's Ones Digit; contains a value from 0 to 9

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **SECTEN<2:0>:** Binary Coded Decimal Value of Second's Tens Digit; contains a value from 0 to 5

bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second's Ones Digit; contains a value from 0 to 9

Note: A write to this register is only allowed when RTCWREN = 1.

37.2.4 Alarm Value Registers

Register 37-8: ALRMVAL (when ALRMPTR<1:0> = 10): Alarm Month and Day Value Register

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **MTHTEN0:** Binary Coded Decimal Value of Month's Tens Digit; contains a value of 0 or 1

bit 11-8 **MTHONE<3:0>:** Binary Coded Decimal Value of Month's Ones Digit; contains a value from 0 to 9

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day's Tens Digit; contains a value from 0 to 3

bit 3-0 **DAYONE<3:0>:** Binary Coded Decimal Value of Day's Ones Digit; contains a value from 0 to 9

Note: A write to this register is only allowed when RTCWREN = 1.

Register 37-9: ALRMVAL (when ALRMPTR<1:0> = 01): Alarm Weekday and Hours Value Register

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit; contains a value from 0 to 6

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour's Tens Digit; contains a value from 0 to 2

bit 3-0 **HRONE<3:0>:** Binary Coded Decimal Value of Hour's Ones Digit; contains a value from 0 to 9

Note: A write to this register is only allowed when RTCWREN = 1.

Register 37-10: ALRMVAL (when ALRMPTR<1:0> = 00): Alarm Minutes and Seconds Value Register

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

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute's Tens Digit; contains a value from 0 to 5

bit 11-8 **MINONE<3:0>:** Binary Coded Decimal Value of Minute's Ones Digit; contains a value from 0 to 9

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **SECTEN<2:0>:** Binary Coded Decimal Value of Second's Tens Digit; contains a value from 0 to 5

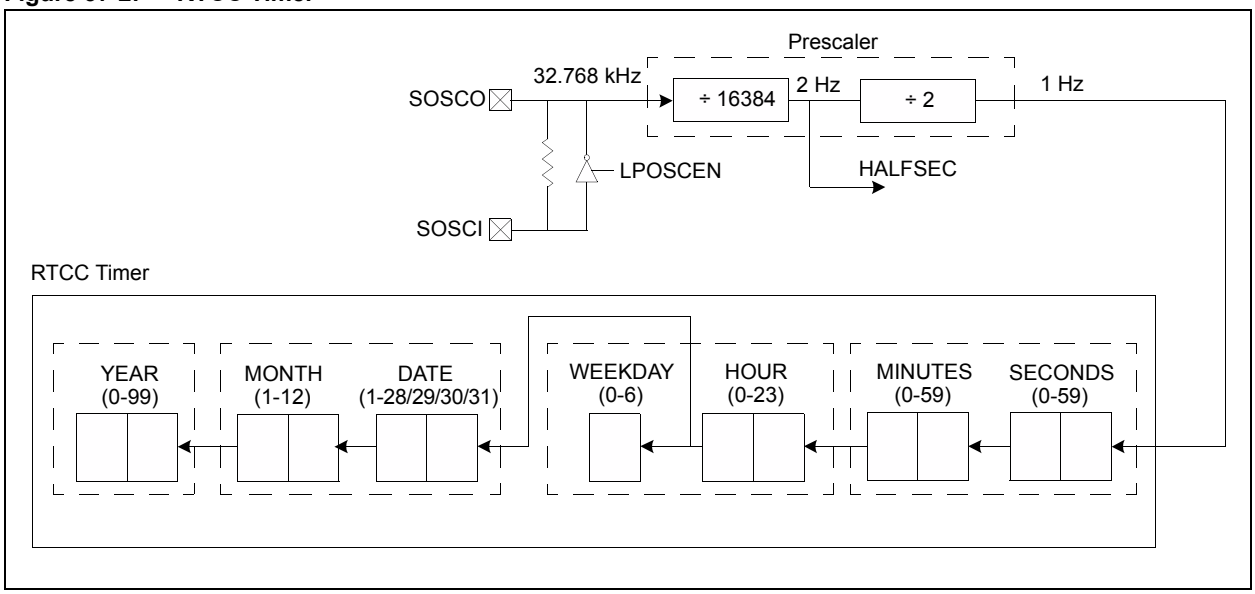
bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second's Ones Digit; contains a value from 0 to 9

37.3 RTCC OPERATION

The RTCC module is intended to be clocked by an external real-time clock crystal oscillating at 32.768 kHz. The prescaler divides the crystal oscillator frequency to produce the 1 Hz update frequency for the clock and calendar. The current date and time is tracked in a 7-bytes counter register that updates once per second.

Each counter counts in BCD, as it allows easy conversion to decimal digits for display or printing. The count sequence of the individual byte counters is shown in Figure 37-2. Note that the day of month and month counters roll over to one. All other counters roll over to zero. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year valid up to 2100. Upon initial application of power the counters contain random information.

Figure 37-2: RTCC Timer



Note:	To allow the RTCC module to be clocked by the secondary crystal oscillator, the Secondary Oscillator Enable (LPOSCEN) bit in the Oscillator Control (OSCCON<1>) register must be set. For further details, refer to **Section 39. "Oscillator (PART III)"**.

37.3.1 Write RTCC Timer

The user application can configure the time and calendar by writing the desired seconds, minutes, hours, weekday, date, month and year to the RTCC registers. Under normal operation, writes to the RTCC timer registers are not allowed. Attempted writes will appear to execute normally, but the contents of the registers will remain unchanged. To write to the RTCC register, the RTCWREN bit (RCFGCAL<13>) must be set. Setting the RTCWREN bit allows writes to the RTCC registers. Conversely, clearing the RTCWREN bit prevents writes.

To set the RTCWREN bit, a specific command sequence must be executed and it can be cleared at any time:

1. Write 0x55 to NVMKEY.
2. Write 0xAA to NVMKEY.
3. Set the RTCWREN bit using a single cycle instruction.

The RTCC module is enabled by setting the RTCEN bit (RCFGCAL<15>). To set or clear the RTCEN bit, the RTCWREN bit (RCFGCAL<13>) must be set.

If the entire clock (hours, minutes and seconds) needs to be corrected, it is recommended to disable the RTCC module, which stops the clock from counting while writing to the RTCC Timer Register. This avoids coincidental write operation with the timer increment.

It is not required to stop the clock if only one location (hours or minutes or seconds) needs to be updated. Doing so would introduce an error in the timekeeping. An example of where only one field needs to be updated is while correcting the hours for daylight savings time. Also, the prescaler resets when the minute or seconds register is written to.

To write to the clock “on the fly" the best method is to wait for the RTCSYNC bit (RCFGCAL<12>) to be ‘0’, and then write to the timekeeping registers. The RTCSYNC bit is set 32 clock edges (~1 ms) before a rollover is about to occur in the prescaler. The code in Example 37-1 demonstrates a RTCC timekeeping register write operation.

Example 37-1: RTCC Timekeeping Register Write Operation

```
;*********************************************
; 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: The Alarm must be disabled (ALRMEN = 0) while writing to real-time clock registers, otherwise it could generate a false alarm.

37.3.2 Read RTCC Timer

The time and calendar information is obtained by reading the appropriate register bytes. The RTCC timer cannot be stopped to read the current time as stopping the RTCC would introduce an error in its timekeeping. Therefore, the RTCC timer is read “on the fly”.

Since clocking of the counter occurs asynchronously to reading of the counter, it is possible to read the counter while it is being incremented (rollover). This may result in an incorrect time reading. Therefore, to ensure a correct reading of the entire contents of the clock (or that part of interest), it must be read without a clock rollover occurring.

To read the clock “on the fly" the best method is to wait for the RTCSYNC bit (RCFGCAL<12>) to be ‘0’, and then read the timekeeping registers. The RTCSYNC bit is set 32 clock edges (~1 ms) before a rollover is about to occur in the prescaler.

If the user application cannot wait for RTCSYNC bit to become ‘0’, the alternate method is to read the timekeeping registers twice. If the data is the same both times, it is considered to be valid. Therefore, a minimum of two and a maximum of three accesses are required.

The code in Example 37-2 shows a RTCC timekeeping register read operation.

Example 37-2: RTCC Timekeeping register Read Operation

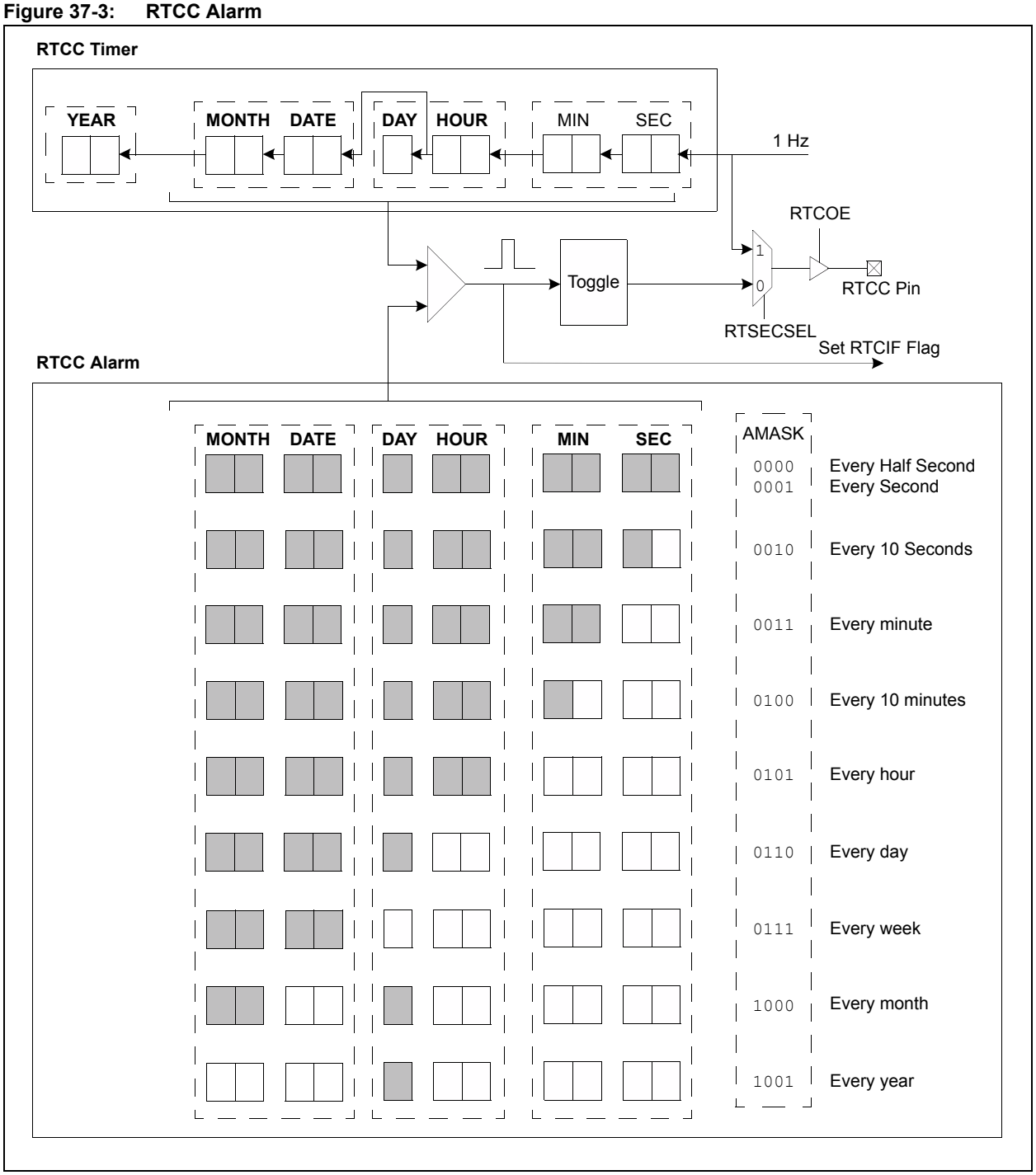
```
;*********************************************
; Wait for RTCSYNC bit to become ‘0’
;*********************************************
while(RCFGCALbits.RTCSYNC==1);

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

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

37.4 RTCC ALARM

Alarms are available to interrupt the CPU at a particular time, or at periodic time intervals, such as once per minute or once per day. During each clock update, the RTCC compares the selected Alarm registers with the corresponding clock registers. When a match occurs, an alarm event is generated.

The Alarm Mask (AMASK<3:0>) bits in the Alarm Configuration (ALCFGRPT<13:10>) are used to mask registers that do not have to be compared. Figure 37-3 illustrates the alarm BCD register nibbles that are compared with the timekeeping register for different AMASK settings.

Figure 37-3: RTCC Alarm



For example, if you want to set an alarm for each morning at six o'clock (06:00:00 in 24-hour format):

1. In the Alarm register, load the hours byte with 06 (BCD), the minutes byte with 00 (BCD), and the seconds byte with 00 (BCD).
2. Mask off the month, date and weekday bytes and just compare hour, minute and second by setting the addressing mask (AMASK) bits to 0b0110. Each day when the time rolls over from 05:59:59, an alarm event is generated.

Note: The alarm must be disabled (ALRMEN = 0) while writing to the Alarm register, otherwise it may result in a false alarm. The Alarm register can be read at any time.

Example 37-3 shows the code to configure the Alarm registers to generate an alarm every morning at six o'clock.

Example 37-3: Writing to Alarm Register

```
;************************************************
; Disable Alarm
;************************************************
ALCFGRPTbits.ALRMEN=0;

;************************************************
; Write to Alarm Register (Time: 06:00:00)
;************************************************
ALCFGRPTbits.ALRMPTR=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;
```

37.4.1 Alarm Mode Selection

The alarm is enabled using the ALRMEN bit (ALCFGRPT<14>), which can generate a one-short alarm and a recurring alarm.

37.4.1.1 ONE-SHORT ALARM

This Alarm event is generated when the clock matches the selected alarm nibbles. The Alarm is automatically disabled on an alarm event. To configure a one-short alarm:

1. Clear the Chime Enable (CHIME) bit in Alarm Configuration (ALCFGRPT<14>) register.
2. Program the Alarm repeat counter (ARPT<7:0>) to '0'.

37.4.1.2 CHIME DISABLE

When CHIME = 0, the Alarm repeat counter (ARPT<7:0>) is decremented on every alarm event and the alarm is disabled when the repeat counter becomes zero.

37.4.1.3 CHIME ENABLE

Indefinite repetition of the Alarm can occur if the Enable (CHIME) bit is set. When CHIME = 1. The Alarm repeat counter (APRT<7:0>) is decremented indefinitely after each time the alarm is issued.

37.4.2 Alarm Interrupt and Output

The Alarm event can generate an RTCC interrupt or toggle the RTCC output pin. A RTCC interrupt is enabled as a source of interrupt via the respective RTCC Interrupt Enable (RTCIE) bit. The interrupt priority level (RTCIP<2:0>) bits must be written with a non-zero value for the RTCC to be a source of interrupt. For further details, refer to **Section 32. "Interrupts (Part III)"**.

In addition, an Alarm event can toggle the RTCC pin thereby generating a periodic clock at half the alarm rate. The RTCC pin is also capable of outputting the seconds clock. The user application can select between the alarm output, or the seconds clock output. The RTSECSEL (PADCFG1<1>) bit selects between these two outputs. When RTSECSEL = 0, the alarm toggles the pin on every alarm event. When RTSECSEL = 1, the seconds clock is selected.

37.5 RTCC CALIBRATION

Calibration is provided to compensate the nominal crystal frequency and for variations in the external crystal frequency over temperature. Calibration is accomplished by adding or subtracting counts every one minute. Adding counts speeds up the clock and subtracting counts slows the clock. The number of pulses blanked (subtracted for negative calibration) or inserted (added for positive calibration) is set by the 8-bit signed value loaded into the calibration (CAL<7:0>) bits in the RTCC Configuration and Calibration Control (RCFGCAL<7:0>) register.

Each calibration step either adds four or subtracts four oscillator cycles for every one minute (60 x 32.768 kHz = 1,966,080 oscillator cycles). This equates to ±2.034 parts per million (ppm) of adjustment per calibration step or ±5.35 seconds per month. It allows calibrating the timekeeping accuracy to within three seconds per month.

Table 37-3 details the amount of adjustment for each value in the calibration register.

Table 37-3: Calibration Adjustment Values

Calibration Value (CAL<7:0>)	Adjustment Accuracy (ppm)	Time (sec/month)	Calibration Value (CAL<7:0>)	Adjustment Accuracy (ppm)	Time (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

To establish how much calibration is required in a given application, the following method, which is specially suited to manufacturing environments can be used:

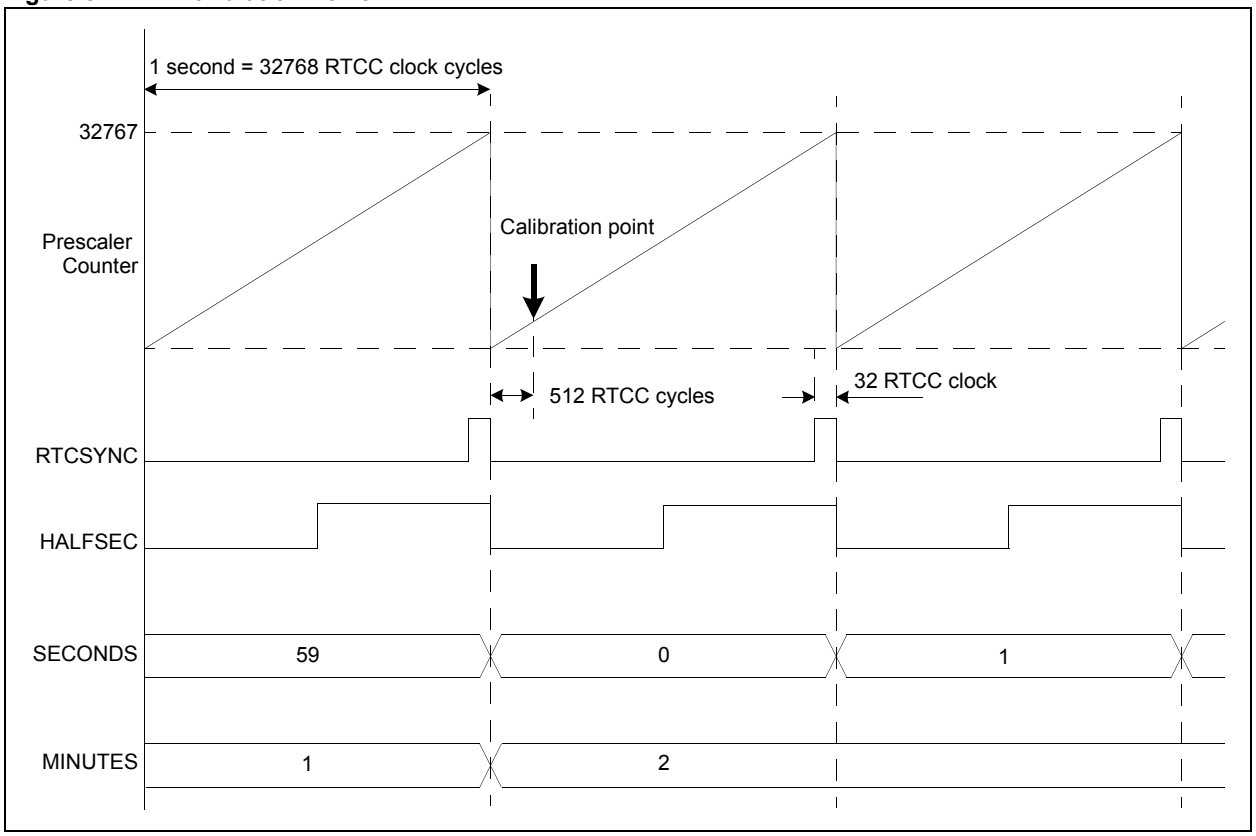
1. Output the seconds clock (1 Hz) on the RTCC pin and measure the crystal oscillator frequency and calculate the crystal frequency deviation in Hz.
2. Frequency Error = 32.768 kHz – Measured Frequency.
3. Calculate counter error per minute due to frequency deviation and Count Error = (Frequency Error) x 60.
4. Set the CAL<7:0> bits to (Count Error/4).

Note: The user application can use a temperature sensor to adjust the calibration value to compensate for the temperature drift.

37.5.1 Writing the Calibration Value

As shown in Figure 37-4, the RTCC module performs calibration or adjusts the prescaler for every minute. The calibration point is at the 512th clock edge following the seconds BCD counter rollover (59 to 00).

Figure 37-4: Calibration Point



At the calibration point, the RTCC module reads the 8-bit signed value (CAL<7:0>) and adjusts the prescaler counter accordingly. To ensure that the RTCC module reads the correct calibration value, it should not be updated by the CPU at the same time the RTCC module reads the value. When the seconds byte is zero, the calibration value must be updated when the HALFSEC (RCFGCAL<11>) bit becomes '1'.

The following code in Example 37-4 shows how to update the calibration value “on the fly”.

Example 37-4: Writing the Calibration Value

```
;************************************************
; 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 OPERATION IN POWER SAVINGS MODES

37.6.1 Sleep Mode

When the device enters Sleep mode, the system clock is disabled. As the RTCC timer is clocked from the Secondary oscillator, it will continue to run in Sleep mode.

If enabled, the Alarm interrupt wakes the device from Sleep and the following occurs:

- If the assigned priority for the interrupt is less than, or equal to, the current CPU priority, the device wakes up and continues code execution from the instruction following the `PWRSAV` instruction that initiated Sleep mode.
- If the assigned priority level for the interrupt source is greater than the current CPU priority, the device wakes up and the CPU exception process begins. Code execution continues from the first instruction of the timer Interrupt Service Routine (ISR). For further details, refer to **Section 9. "Watchdog Timer and Power Savings Modes"**.

37.6.2 Idle Mode

Idle mode does not affect the operation of the RTCC module.

37.7 REGISTER MAP

Table 37-4: Real-Time Clock and Calendar Register Map

File Name	Addr	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	All Resets
ALRMVAL	0620	Alarm Value Register Window based on APTR<1:0>																xxxx
ALCFGRPT	0622	ALRMEN	CHIME	AMASK<3:0>				ALRMPTR<1:0>		ARPT<7:0>								0000
RTCVAL	0624	RTCC Value Register Window based on RTCPTR<1:0>																xxxx
RCFGCAL	0626	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR<1:0>		CAL<7:0>								0000
PADCFG1	02FC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTSECSEL	PMPTTL	0000

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

37.8 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the dsPIC33F product family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Real-Time Clock and Calendar (RTCC) module are:

Title	Application Note #
No related application notes at this time.	

Note: Please visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the dsPIC33F family of devices.

37.9 REVISION HISTORY

Revision A (November 2007)

This is the initial release of this document.