Chapter 1

SLAU420F–August 2012–Revised March 2018

Real-Time Clock C (RTC_C)

NOTE: This chapter is an excerpt from the *MSP430x5xx and MSP430x6xx Family User's Guide*. The latest version of the full user's guide is available from http://www.ti.com/lit/pdf/slau208.

The Real-Time Clock C (RTC_C) module provides clock counters with calendar mode, a flexible programmable alarm, offset calibration, and a provision for temperature compensation. The RTC_C also supports operation in LPM3.5. This chapter describes the RTC_C module.

Topic **Page**

1.1 Real-Time Clock (RTC_C) Introduction

The RTC_C module provides configurable clock counters.

RTC_C features include:

- Real-time clock and calendar mode that provides seconds, minutes, hours, day of week, day of month, month, and year (including leap year correction)
- Protection for real-time clock registers
- Interrupt capability
- Selectable BCD or binary format
- Programmable alarms
- Real-time clock calibration for crystal offset error
- Real-time clock compensation for crystal temperature drift
- Operation in LPM3.5

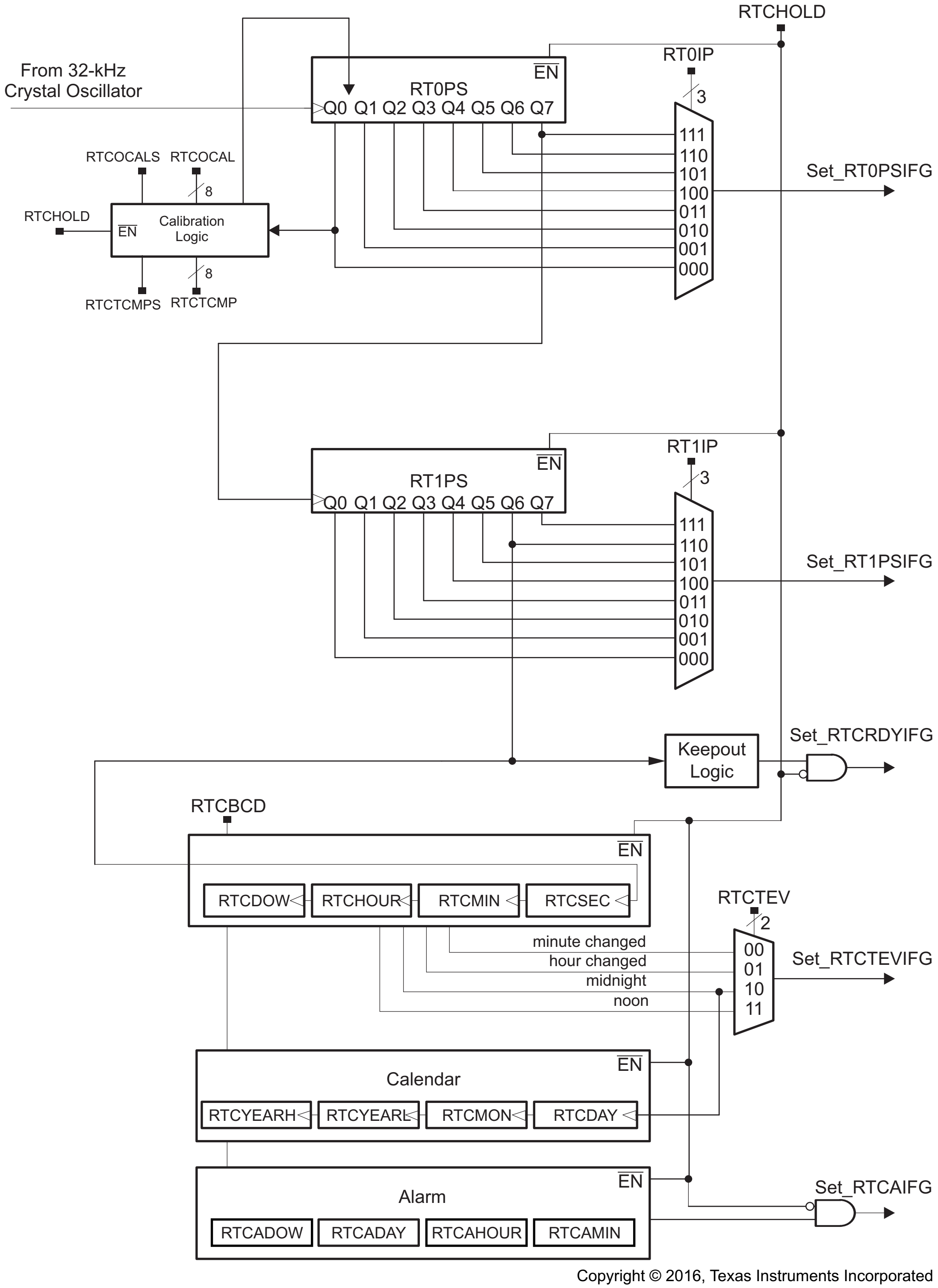
The RTC_C module can provide the following device-dependent features. Refer to the device-specific data sheet to determine if these features are available in a particular device.

- General-purpose counter mode (see Section 1.3.1)
- Event and tamper detection with time stamp (see Section 1.3.2)
- Operation from a separate voltage supply

NOTE: Real-time clock initialization

Most RTC_C module registers have no initial condition. These registers must be configured by user software before use.

Figure 1-1 shows the RTC_C block diagram.



Copyright © 2016, Texas Instruments Incorporated

Figure 1-1. RTC_C Block Diagram (RTCMODE = 1)

1.2 RTC_C Operation

1.2.1 Calendar Mode

Calendar mode is selected when RTCMODE is set. In calendar mode, the RTC_C module provides seconds, minutes, hours, day of week, day of month, month, and year in selectable BCD or hexadecimal format. The calendar includes a leap-year algorithm that considers all years evenly divisible by four as leap years. This algorithm is accurate from the year 1901 through 2099. Switching from counter mode (if available) to calendar mode **does not** reset the calendar registers (RTCSEC, RTCMIN, RTCHOUR, RTCDAY, RTCDOW, and RTCYEAR) and the prescale counters (RT0PS, RT1PS). These registers must be configured by user software before use.

1.2.2 Real-Time Clock and Prescale Dividers

The prescale dividers, RT0PS and RT1PS, are automatically configured to provide a 1-second clock interval for the RTC_C. The low-frequency oscillator must be operated at 32768 Hz (nominal) for proper RTC_C operation. RT0PS is sourced from the low-frequency oscillator XT1. The output of RT0PS divided by 256 (Q7) sources RT1PS. RT1PS is further divided by 128 to source the real-time clock counter registers that provide the required 1-second time interval.

When RTCBCD = 1, BCD format is selected for the calendar registers. It is possible to switch between BCD and hexadecimal format while the RTC is counting.

Setting RTCHOLD halts the real-time counters and prescale counters, RT0PS and RT1PS.

NOTE: For reliable update to all Calendar Mode registers

Set RTCHOLD = 1 before writing into any of the calendar or prescalar registers (RTCPS0, RTCPS1, RTCSEC, RTCMIN, RTCHOUR, RTCDAY, RTCDOW, RTCMON, and RTCYEAR).

1.2.3 Real-Time Clock Alarm Function

The RTC_C module provides for a flexible alarm system. There is a single user-programmable alarm that can be programmed based on the settings contained in the alarm registers for minutes, hours, day of week, and day of month.

Each alarm register contains an alarm enable (AE) bit that can be used to enable the respective alarm register. By setting AE bits of the various alarm registers, a variety of alarm events can be generated.

- Example 1: A user wishes to set an alarm every hour at 15 minutes past the hour (that is, 00:15:00, 01:15:00, 02:15:00, and so on). This is possible by setting RTCAMIN to 15. By setting the AE bit of the RTCAMIN and clearing all other AE bits of the alarm registers, the alarm is enabled. When enabled, the RTCAIFG is set when the count transitions from 00:14:59 to 00:15:00, 01:14:59 to 01:15:00, 02:14:59 to 02:15:00, and so on.
- Example 2: A user wishes to set an alarm every day at 04:00:00. This is possible by setting RTCAHOUR to 4. By setting the AE bit of the RTCHOUR and clearing all other AE bits of the alarm registers, the alarm is enabled. When enabled, the RTCAIFG is set when the count transitions from 03:59:59 to 04:00:00.
- Example 3: A user wishes to set an alarm for 06:30:00. RTCAHOUR would be set to 6, and RTCAMIN would be set to 30. By setting the AE bits of RTCAHOUR and RTCAMIN, the alarm is enabled. When enabled, the RTCAIFG is set when the time count transitions from 06:29:59 to 06:30:00. In this case, the alarm event occurs every day at 06:30:00.
- Example 4: A user wishes to set an alarm every Tuesday at 06:30:00. RTCADOW would be set to 2, RTCAHOUR would be set to 6, and RTCAMIN would be set to 30. By setting the AE bits of RTCADOW, RTCAHOUR, and RTCAMIN, the alarm is enabled. When enabled, the RTCAIFG is set when the time count transitions from 06:29:59 to 06:30:00 and the RTCDOW transitions from 1 to 2.
- Example 5: A user wishes to set an alarm the fifth day of each month at 06:30:00. RTCADAY would be set to 5, RTCAHOUR would be set to 6, and RTCAMIN would be set to 30. By setting the AE bits of RTCADAY, RTCAHOUR, and RTCAMIN, the alarm is enabled. When enabled, the RTCAIFG is set when the time count transitions from 06:29:59 to 06:30:00 and the RTCDAY equals 5.

NOTE: Invalid alarm settings

Invalid alarm settings are not checked by hardware. It is the user's responsibility that valid alarm settings are entered.

NOTE: Invalid time and date values

Writing of invalid date or time information or data values outside the legal ranges specified in the RTCSEC, RTCMIN, RTCHOUR, RTCDAY, RTCDOW, RTCYEARH, RTCYEARL, RTCAMIN, RTCAHOUR, RTCADAY, and RTCADOW registers can result in unpredictable behavior.

NOTE: Setting the alarm

Before setting an initial alarm, all alarm registers including the AE bits should be cleared.

To prevent potential erroneous alarm conditions from occurring, the alarms should be disabled by clearing the RTCAIE, RTCAIFG, and AE bits before writing initial or new time values to the RTC time registers.

1.2.4 Real-Time Clock Protection

RTC_C registers are key protected to ensure clock integrity and module configuration against software crash or from runaway code. Key protection does not apply for reads from the RTC_C registers. That is, any RTC_C register can be read at any time without having to unlock the module. Some predefined registers of RTC_C are key protected for write access. The control registers, clock registers, calendar register, prescale timer registers, and offset error calibration registers are protected. RTC_C alarm function registers, prescale timer control registers, interrupt vector register, and temperature compensation registers are not protected. RTC_C registers that are not protected can be written at any time without unlocking the module. Table 1-2 shows which registers are affected by the protection scheme.

The RTCCTL0_H register implements key protection and controls the lock or unlock state of the module. When this register is written with correct key, 0A5h, the module is unlocked and unlimited write access possible to RTC_C registers. After the module is unlocked, it remains unlocked until the user writes any incorrect key or until the module is reset. A read from RTCCTL0_H register returns value 96h. Write access to any protected registers of RTC_C is ignored when the module is locked.

```
RTC_C Key Protection Software Example
; Unlock/lock sequence for RTC_C
MOV.B   #RTCKEY, &RTCCTL0_H            ; Write correct key to unlock RTC_C
MOV.B   #8bit_value, &RTCSEC           ; Write 8 bit value into RTCSEC
MOV.B   #8bit_value, &RTCMIN           ; Write 8 bit value into RTCMIN
MOV.W   #16bit_value, &RTCTIM1         ; Write 16bit value into RTCTIM1
MOV.W   #RTCKEY+8bit_value, &RTCCTL0   ; Write into RTCCTL0 with correct key in word mode
...

MOV.B   #00h, &RTCCTL0_H               ; Write incorrect key to lock RTC_C
```

1.2.5 Reading or Writing Real-Time Clock Registers

Because the system clock may be asynchronous to the RTC_C clock source, special care must be used when accessing the real-time clock registers.

The real-time clock registers are updated once per second. To prevent reading any real-time clock register at the time of an update that could result in an invalid time being read, a keep-out window is provided. The keep-out window is centered approximately 128/32768 seconds around the update transition. The read-only RTCRDY bit is reset during the keep-out window period and set outside the keep-out the window period. Any read of the clock registers while RTCRDY is reset is considered to be potentially invalid, and the time read should be ignored.

An easy way to safely read the real-time clock registers is to use the RTCRDYIFG interrupt flag. Setting RTCRDYIE enables the RTCRDYIFG interrupt. When enabled, an interrupt is generated based on the rising edge of the RTCRDY bit, causing the RTCRDYIFG to be set. At this point, the application has nearly a complete second to safely read any or all of the real-time clock registers. This synchronization process prevents reading the time value during transition. The RTCRDYIFG flag is reset automatically when the interrupt is serviced or can be reset with software.

NOTE: Reading or writing real-time clock registers

When the counter clock is asynchronous to the CPU clock, any read from any RTCSEC, RTCMIN, RTCHOUR, RTCDOW, RTCDAY, RTCMON, RTCYEARL, or RTCYEARH register while the RTCRDY is reset may result in invalid data being read. To safely read the counting registers, either polling of the RTCRDY bit or the synchronization procedure previously described can be used. Alternatively, the counter register can be read multiple times while operating, and a majority vote taken in software to determine the correct reading. Reading the RT0PS and RT1PS can only be handled by reading the registers multiple times and a majority vote taken in software to determine the correct reading.

Any write to any counting register takes effect immediately. However, the clock is stopped during the write. In addition, RT0PS and RT1PS registers are reset. This could result in losing up to 1 second during a write. Writing of data outside the legal ranges or invalid time stamp combinations results in unpredictable behavior.

1.2.6 Real-Time Clock Interrupts

At least six sources for interrupts are available, namely RT0PSIFG, RT1PSIFG, RTCRDYIFG, RTCTEVIFG, RTCAIFG, and RTCOFIFG. These flags are prioritized and combined to source a single interrupt vector. The interrupt vector register (RTCIV) is used to determine which flag requested an interrupt.

The highest-priority enabled interrupt generates a number in the RTCIV register (see register description). This number can be evaluated or added to the program counter (PC) to automatically enter the appropriate software routine. Disabled RTC interrupts do not affect the RTCIV value.

Writes into RTCIV register clear all pending interrupt conditions. Reads from RTCIV register clear the highest priority pending interrupt condition. If another interrupt flag is set, another interrupt is immediately generated after servicing the initial interrupt. In addition, all flags can be cleared by software.

The user-programmable alarm event sources the real-time clock interrupt, RTCAIFG. Setting RTCAIE enables the interrupt. In addition to the user-programmable alarm, the RTC_C module provides for an interval alarm that sources real-time clock interrupt, RTCTEVIFG. The interval alarm can be selected to cause an alarm event when RTCMIN changed or RTCHOUR changed, every day at midnight (00:00:00) or every day at noon (12:00:00). The event is selectable with the RTCTEV bits. Setting the RTCTEVIE bit enables the interrupt.

The RTCRDY bit sources the real-time clock interrupt, RTCRDYIFG, and is useful in synchronizing the read of time registers with the system clock. Setting the RTCRDYIE bit enables the interrupt.

RT0PSIFG can be used to generate interrupt intervals selectable by the RT0IP bits. RT0PS is sourced with low-frequency oscillator clock at 32768 Hz, so intervals of 16384 Hz, 8192 Hz, 4096 Hz, 2048 Hz, 1024 Hz, 512 Hz, 256 Hz, or 128 Hz are possible. Setting the RT0PSIE bit enables the interrupt.

RT1PSIFG can be used to generate interrupt intervals selectable by the RT1IP bits. RT1PS is sourced with the output of RT0PS, which is 128 Hz (32768/256 Hz). Therefore, intervals of 64 Hz, 32 Hz, 16 Hz, 8 Hz, 4 Hz, 2 Hz, 1 Hz, or 0.5 Hz are possible. Setting the RT1PSIE bit enables the interrupt.

NOTE: Changing RT0IP or RT1IP

Changing the settings of the interrupt interval bits RT0IP or RT1IP while the corresponding prescaler is running or is stopped in a non-zero state can result in setting the corresponding interrupt flags.

The RTCOFIFG bit flags the failure of the 32-kHz crystal oscillator. Its main purpose is to wake up the CPU from LPM3.5 if an oscillator failure occurs. On devices with separate supply for RTC, this flag also stores a failure event that occurs when the core supply is not available.

1.2.6.1 RTCIV Software Example

The following software example shows the recommended use of RTCIV and the handling overhead. The RTCIV value is added to the PC to automatically jump to the appropriate routine.

The numbers at the right margin show the necessary CPU cycles for each instruction. The software overhead for different interrupt sources includes interrupt latency and return-from-interrupt cycles, but not the task handling itself.

```
; Interrupt handler for RTC interrupt flags.

RTC_HND                        ; Interrupt latency          6
         ADD   &RTCIV,PC       ; Add offset to Jump table    3
         RETI                  ; Vector  0: No interrupt     5
         JMP   RTCOFIFG_HND    ; Vector  2: RTCOFIFG         2
         JMP   RTCRDYIFG_HND   ; Vector  4: RTCRDYIFG        2
         JMP   RTCTEVIFG_HND   ; Vector  6: RTCTEVIFG        2
         JMP   RTCAIFG         ; Vector  8: RTCAIFG          5
         JMP   RT0PSIFG        ; Vector  A: RT0PSIFG         5
         JMP   RT1PSIFG        ; Vector  C: RT1PSIFG         5
         RETI                  ; Vector  E: Reserved         5

RTCOFIFG_HND                   ; Vector 2: RTCOFIFG Flag
         ...                   ; Task starts here
         RETI                  ; Back to main program        5

RTCRDYIFG_HND                  ; Vector 4: RTCRDYIFG Flag
         ...                   ; Task starts here
         RETI                  ; Back to main program        5

RTCTEVIFG_HND                  ; Vector 6: RTCTEVIFG
         ...                   ; Task starts here
         RETI                  ; Back to main program        5

RTCAIFG_HND                    ; Vector 8: RTCAIFG
         ...                   ; Task starts here
         RETI                  ; Back to main program        5

RT0PSIFG_HND                   ; Vector A: RT0PSIFG
         ...                   ; Task starts here
         RETI                  ; Back to main program        5

RT1PSIFG_HND                   ; Vector C: RT1PSIFG
         ...                   ; Task starts here
         RETI                  ; Back to main program        5
```

1.2.7 Real-Time Clock Calibration for Crystal Offset Error

The RTC_C module can be calibrated for crystal manufacturing tolerance or offset error to enable better accuracy of time keeping accuracy. The crystal frequency error of up to ±240 ppm can be calibrated smoothly over a period of 60 seconds. RTCOCAL_L register is used to adjust the frequency. The calibration value is written into RTCOCAL_L register, and each LSB in this register represent approximately ±1-ppm correction based on RTCOCALS bit in RTCOCAL_H register. When RTCOCALS bit is set (up calibration), each LSB in RTCOCAL_L represent +1-ppm adjustment. When RTCOCALS is cleared (down calibration), each LSB in RTCOCAL_L represent –1-ppm adjustment to frequency. Both RTCOCAL_L and RTCOCAL_H registers are protected and require RTC_C to be unlocked before writing into these registers.

1.2.7.1 Calibration Frequency

To calibrate the frequency, the RTCCLK output signal is available at a pin. RTCCALFx bits in RTCCTL3 register can be used to select the frequency rate of the output signal. When RTCCALFx = 00, no signal is output on RTCCLK pin. The other settings of RTCCALFx select one the three frequencies: 512 Hz, 256 Hz, or 1 Hz. RTCCLK can be measured, and the result of this measurement can be applied to the RTCOCALS and RTCOCALx bits to effectively reduce the initial offset of the clock.

1.2.7.1.1 Calibration Mechanism

RTCOCAL_L is an 8-bit register. Software can write a value of up to 256 ppm into this register, but the maximum frequency error that can be corrected is only 240 ppm. Software must make sure to write legal values into this register. A read from RTCOCAL always returns the value that was written by software. Real-time clock offset error calibration is inactive when RTC_C is not enabled (RTCHOLD = 0) or when RTCOCALx bits are zero. RTCOCAL should only be written when RTCHOLD = 1. Writing RTCOCAL resets temperature compensation to zero.

In RTC_C, the offset error calibration takes place over a period of 60 seconds. To achieve approximately ±1-ppm correction, the 16-kHz clock (Q0 output of RT0PS) is adjusted to add or subtract one clock pulse. For +1-ppm correction, one clock pulse is added to the 16-kHz clock, and for –1-ppm correction, one clock pulse is subtracted from the 16-kHz clock. This correction happens once every quarter second until the programmed ppm error is compensated.

$f_{ACLK,meas} < 32768\text{ Hz} \rightarrow \text{RTCOCALS} = 1, \text{RTCOCALx} = \text{Round}(60 \times 16384 \times (1 - f_{ACLK,meas}/32768))$

$f_{ACLK,meas} \geq 32768\text{ Hz} \rightarrow \text{RTCOCALS} = 0, \text{RTCOCALx} = \text{Round}(60 \times 16384 \times (1 - f_{ACLK,meas}/32768))$

As an example for up calibration, when the measured frequency is 511.9658 Hz against the reference frequency of 512 Hz, the frequency error is approximately 67 ppm low. To increase the frequency by 67 ppm, RTCOCALS should be set, and RTCOCALx should be set to Round (60 × 16384 × (1 – 511.9658 × 64 / 32768)) = 66.

As an example for down calibration, when the measured frequency is 512.0241 Hz against the reference frequency of 512 Hz, the frequency error is approximately 47 ppm high. To decrease the frequency by 47 ppm, RTCOCALS should be cleared, and RTCOCALx should be set to Round (60 × 16384 × (1 – 512.0241 × 64 / 32768)) = 46.

All three possible output frequencies (512 Hz, 256 Hz, and 1 Hz) at RTCCLK pin are affected by calibration settings. RT0PS interrupt triggered by RT0PS – Q0 (RT0IPx = 000) is based on the uncalibrated clock, while RT0PS interrupt triggered by RT0PS – Q1 to Q7 (RT0IPx ≠ 000) is based on the calibrated clock. RT1PS interrupt (RT1PSIFG) and RTC counter interrupt (RTCTEVIFG) are also based on the calibrated clock.

1.2.8 Real-Time Clock Compensation for Crystal Temperature Drift

The frequency output of the crystal varies considerably due to drift in temperature. It would be necessary to compensate the real-time clock for this temperature drift for higher time keeping accuracy from standard crystals. A hybrid software and hardware approach can be followed to achieve temperature compensation for RTC_C.

The software can make use of an (on-chip) temperature sensor to measure the temperature at desired intervals (for example, once every few seconds or minutes). The temperature sensor parameters are calibrated at production and stored in the nonvolatile memory. Using the temperature sensor parameters and the measured temperature, software can do parabolic calculations to find out the corresponding frequency error in ppm.

This frequency error can be written into RTCTCMP_L register for temperature compensation. RTCTCMP_L is an 8-bit register that allows correction for a frequency error up to ±240 ppm. Each LSB in this register represent ±1 ppm based on the RTCTCMPS bit in the RTCTCMP_H register. When RTCTCMPS bit is set, each LSB in RTCTCMP represents +1-ppm adjustment (up calibration). When RTCTCMPS is cleared, each LSB in RTCTCMP represents –1-ppm adjustment (down calibration). RTCTCMP register is not protected and can be written any time without unlocking RTC_C.

1.2.8.1 Temperature Compensation Scheme

RTCTCMP_L is an 8-bit register. Software can write up to value of 256 ppm into this register, but the maximum frequency error that can be corrected including the crystal offset error is 240 ppm. Real-time clock temperature compensation is inactive when RTC_C is not enabled (RTCHOLD = 0) or when RTCTCMPx bits are zero.

When the temperature compensation value is written into RTCTCMP_L, it is added to the offset error calibration value, and the resulting value is taken into account from next calibration cycle onwards. The ongoing calibration cycle is not affected by writes into the RTCTCMP register. The maximum frequency error that can be corrected to account for both offset error and temperature variation is ±240 ppm. This means the sign addition of offset error value and temperature compensation value should not exceed maximum of ±240 ppm; otherwise, the excess value above ±240 ppm is ignored by hardware. Reading from the RTCTCMP register at any time returns the cumulative value which is the signed addition of RTCOCALx and RTCTCMPx values. (Note that writing RTCOCAL resets the temperature compensation value to zero.)

For example, when RTCOCAL value is +150 ppm, and the value written into RTCTCMP is +200 ppm, the effective value taken in for next calibration cycle is +240 ppm. Software is expected to do temperature measurement at certain regularity, calculate the frequency error, and write into RTCTCMP register to not exceed the maximum limit of ±240 ppm.

Changing the sign bit by writing to RTCTCMP_H becomes effective only after also writing RTCTCMP_L. Thus TI recommends writing the sign bit together with compensation value as a 16-bit value into RTCTCMP.

1.2.8.2 Writing to RTCTCMP Register

Because the system clock can be asynchronous to the RTC_C clock source, the RTCTCRDY bit in the RTCTCMP_H register should be considered for reliable writing into RTCTCMP register. RTCTCRDY is a read-only bit that is set when the hardware is ready to take in the new temperature compensation value. A write to RTCTCMP should be avoided when RTCTCRDY bit is reset. Writes into RTCTCMP register when RTCTCRDY is reset are ignored.

RTCTCOK is a status bit that indicates if the write to RTCTCMP register is successful or not. RTCTCOK is set if the write to RTCTCMP is successful and reset if the write is unsuccessful. The status remains the same until the next write to the RTCTCMP register. If the write to RTCTCMP is unsuccessful, then the user needs to attempt writing into RTCTCMP again when RTCTCRDY is set.

Figure 1-2 shows the scheme for real-time clock offset error calibration and temperature compensation.

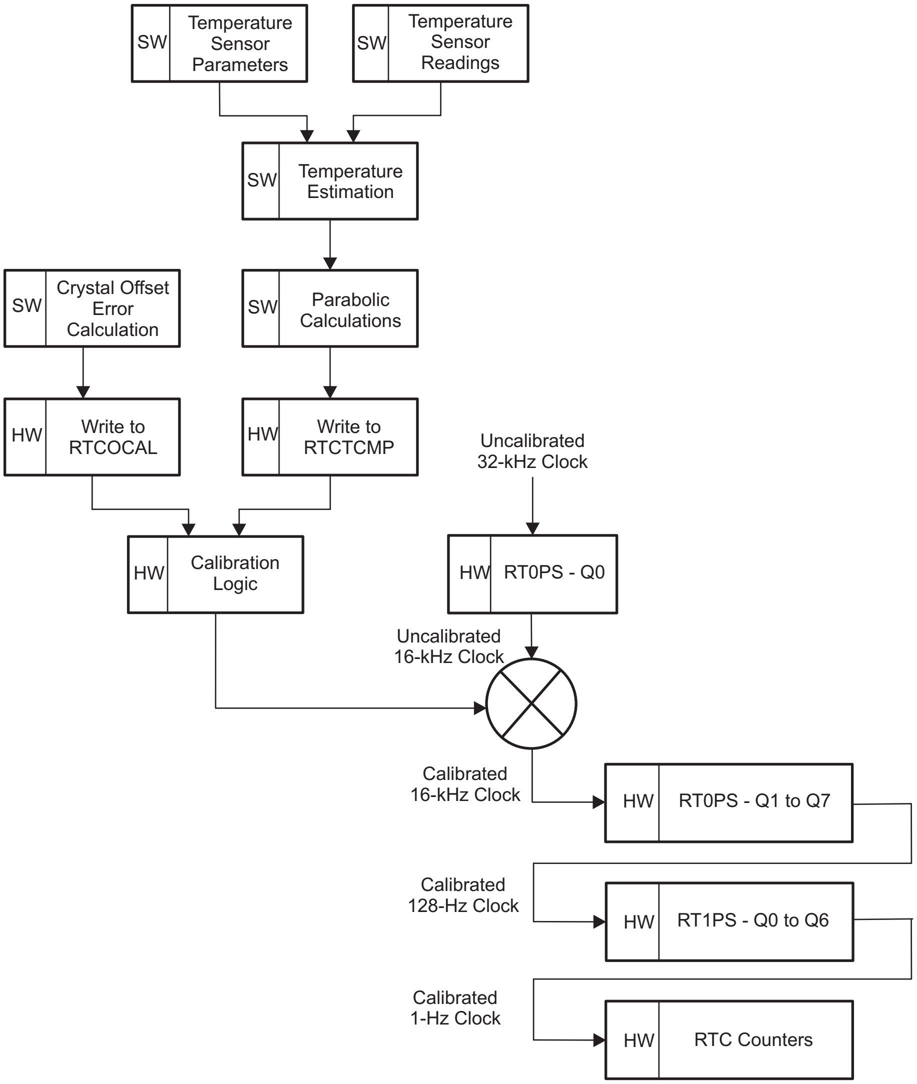


Figure 1-2. RTC_C Offset Error Calibration and Temperature Compensation Scheme

1.2.8.3 Temperature Measurement and Updates to RTC_C

The user may wish to perform temperature measurement once every few seconds or once every minute or once in several minutes. Writing to RTCTCMP register for temperature compensation is effective always once in one minute. This means that if the user performs temperature measurement every minute and updates RTCTCMP register with the frequency error, compensation would immediately work fine. But if software performs temperature measurement more frequently than once per minute (for example once every 5 seconds) then it needs to average the error over one minute and update RTCTCMP register once per minute. If the software performs temperature measurement less frequently than once per minute (for example, once every 5 minutes) then it needs to calculate the frequency error for the measured temperature and write into RTCTCMP register. The value written into RTCTCMP in this case would be effective until it is updated again by software.

1.2.9 Real-Time Clock Operation in LPM3.5 Low-Power Mode

The regulator of the Power Management Module (PMM) is disabled when the device enters LPM3.5, which causes most of the RTC_C configuration registers to be lost; only the counters and calibration registers are retained. Table 1-2 shows which registers are retained in LPM3.5. Also the configuration of the interrupt enables is stored so that the configured interrupts can cause a wakeup upon exit from LPM3.5. Interrupt flags that are set before entering LPM3.5 are cleared upon entering LPM3.5 (Note: this can only happen if the corresponding interrupt is not enabled). The interrupt flags RTCTEVIFG, RTCAIFG, RT1PSIFG, and RTCOFIFG can be used as RTC_C wakeup interrupt sources. Any interrupt event that occurs during LPM3.5 is stored in the corresponding flags, but only enabled interrupts can wake up the device. After restoring the configuration registers (and clearing LOCKLPM5), the interrupts can be serviced as usual.

The detailed flow is as follows:

1. Set all I/Os to general-purpose I/Os and configure as needed. Optionally, configure input interrupt pins for wakeup. Configure RTC_C interrupts for wake-up (set RTCTEVIE, RTCAIE, RT1PSIE, or RTCOFIE. If the alarm interrupt is also used as wake-up event, the alarm registers must be configured as needed).
2. Enter LPM3.5 with LPM3.5 entry sequence:

```
bic #RTCHOLD, &RTCCTL13
bis #PMMKEY + REGOFF, &PMMCTL0
bis #LPM4, SR
```

3. LOCKLPM5 is automatically set by hardware upon entering LPM3.5, the core voltage regulator is disabled, and all clocks are disabled except for the 32-kHz crystal oscillator clock as the RTC_C is enabled with RTCHOLD = 0.
4. An LPM3.5 wake-up event like an edge on a wake-up input pin or an RTC_C interrupt event starts the BOR entry sequence and the core voltage regulator. All peripheral registers are set to their default conditions. The I/O pin state and the interrupt configuration for the RTC_C remain locked.
5. The device can be configured. The I/O configuration and the RTC_C interrupt configuration that was not retained during LPM3.5 should be restored to the values that they had before entering LPM3.5. Then the LOCKLPM5 bit can be cleared, which releases the I/O pin conditions and the RTC_C interrupt configuration. Registers that are retained during LPM3.5 should not be altered before LOCKLPM5 is cleared.
6. After enabling I/O and RTC_C interrupts, the interrupt that caused the wake-up can be serviced.

If the RTC_C is enabled (RTCHOLD = 0), the 32-kHz oscillator remains active during LPM3.5. The fault detection also remains functional. If a fault occurs during LPM3.5 and the RTCOFIE was set before entering LPM3.5, a wake-up event is issued.

In F677xA, F673xA, and F672xA devices, the RTCLOCK bit in RTCCTL3 allows to keep all LPM3.5 retention logic from write and reset operations. In addition, the following interrupt bits are also protected:

- RTCTCCTL1: RTCCAPIFG
- RTCCAPxCTL: CAPEV

When set, this bit also affects XT1 LF control bits which can be retained in LPM3.5. These XT1 control bits include:

- UCSCTL6: XT1DRIVE, XTS, XT1BYPASS, XCAP

When this bit is set, LPM3.5 retention logic is not writable, and any reset cannot change its operation unless a power cycle occurs. Meanwhile, an unlock operation is required to open the write access of interrupt flags before they are cleared in ISR. If this bit is clear, these logics are writable and can be reset by PUC, POR, and BOR.

1.3 RTC_C Operation - Device-Dependent Features

1.3.1 Counter Mode

NOTE: This feature is available only on selected devices. See the device-specific data sheet to determine if this feature is available.

The RTC_C module can be configured as a real-time clock with calendar function (calendar mode) or as a 32-bit general-purpose counter (counter mode) with the RTCMODE bit.

Counter mode is selected when RTCMODE is reset. In this mode, a 32-bit counter is provided that is directly accessible by software. Figure 1-3 shows the functional block diagram of a RTC_C module in counter mode (RTCMODE = 0).

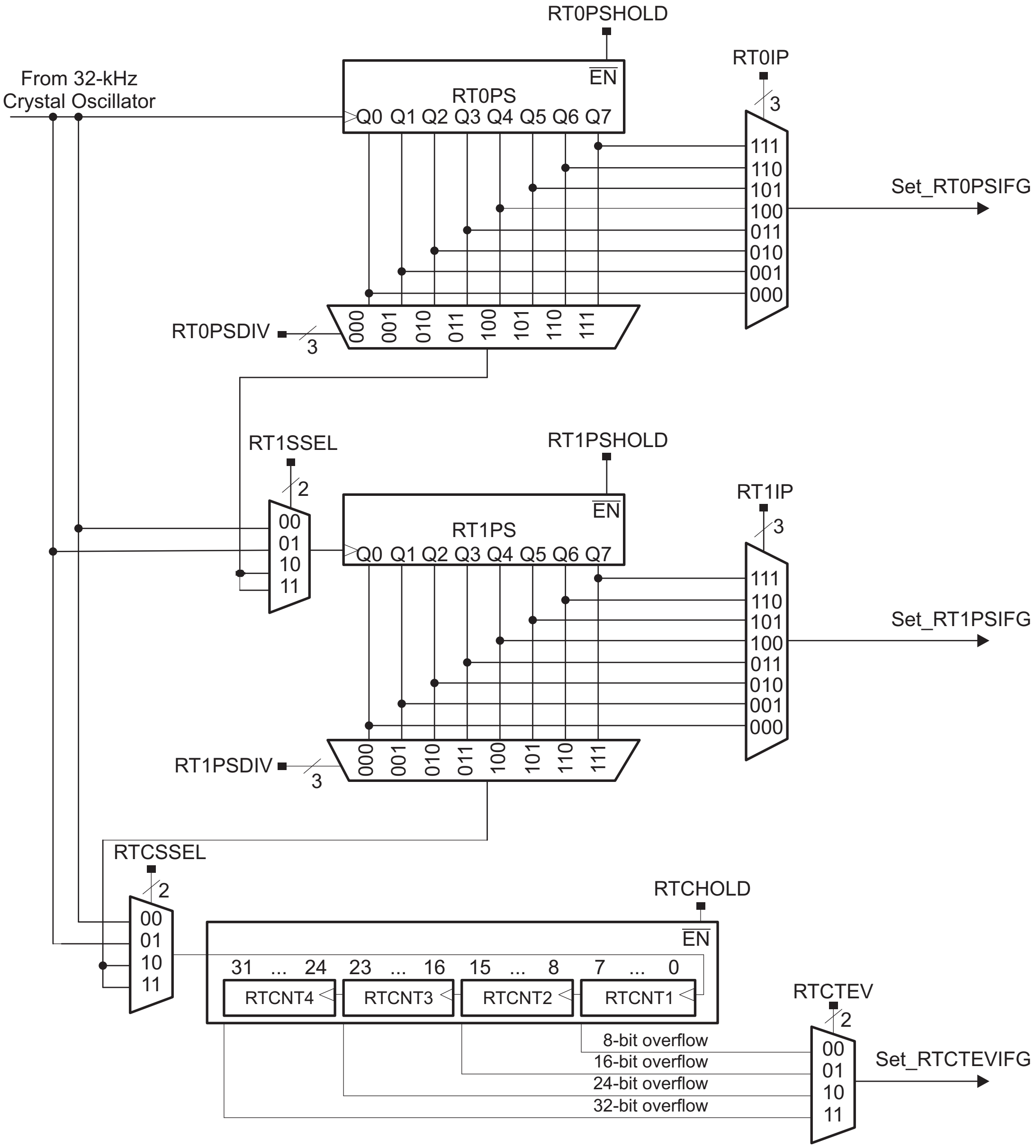


Figure 1-3. RTC_C Functional Block Diagram in Counter Mode (RTCMODE = 0)

1.3.1.1 Switching between Calendar and Counter Mode

To switch from calendar mode to counter mode do the following steps:

1. Stop the RTC with RTCHOLD=1.
2. Clear the RTCMODE bit to enable the counter mode.
3. Initialize the count registers (RTCNT1, RTCNT2, RTCNT3, RTCNT4) and the prescale counters (RT0PS, RT1PS) as needed. (The switch from calendar mode to counter mode **does not** reset the count value (RTCNT1, RTCNT2, RTCNT3, RTCNT4) or the prescale counters (RT0PS, RT1PS). These registers must be configured by user software before use.)
4. Clear the RTCHOLD bit to start the counters.

To switch back from counter mode to calender mode do the following steps:

1. Stop the counters with RTCHOLD=1.
2. Set RTCMODE=1 to enable the calendar mode.
3. Initialize the clock and calendar registers and the prescale counters (RT0PS, RT1PS) as needed. (The switch from counter mode to calendar mode **does not** reset the clock and calendar registers nor the prescale counters (RT0PS, RT1PS). These registers must be configured by user software before use.)
4. Clear the RTCHOLD bit to start the RTC.

1.3.1.2 Counter Mode Operation

The clock that increments the counter can be sourced from the 32-kHz crystal oscillator or from prescaled versions of the 32-kHz crystal oscillator clock. Prescaled versions are sourced from the prescale dividers (RT0PS and RT1PS). RT0PS and RT1PS can output /2, /4, /8, 16, /32, /64, /128, and /256 versions of the 32-kHz clock. The output of RT0PS can be cascaded with RT1PS. The cascaded output can also be used as a clock source input to the 32-bit counter.

Four individual 8-bit counters are cascaded to provide the 32-bit counter. This provides 8-bit, 16-bit, 24-bit, or 32-bit overflow intervals of the counter clock. The RTCTEV bits select the respective trigger event. An RTCTEV event can trigger an interrupt by setting the RTCTEVIE bit. Each counter, RTCNT1 through RTCNT4, is individually accessible and may be written.

RT0PS and RT1PS can be configured as two 8-bit counters or cascaded into a single 16-bit counter. RT0PS and RT1PS can be halted on an individual basis by setting their respective RT0PSHOLD and RT1PSHOLD bits. When RT0PS is cascaded with RT1PS, setting RT0PSHOLD causes both RT0PS and RT1PS to be halted. The 32-bit counter can be halted several ways, depending on the configuration. If the 32-bit counter is sourced directly by the 32-kHz crystal clock, it can be halted by setting RTCHOLD. If it is sourced from the output of RT1PS, it can be halted by setting RT1PSHOLD or RTCHOLD. Finally, if it is sourced from the cascaded outputs of RT0PS and RT1PS, it can be halted by setting RT0PSHOLD, RT1PSHOLD, or RTCHOLD.

NOTE: Accessing the RTCNT1, RTCNT2, RTCNT3, RTCNT4, RT0PS, RT1PS registers

When the counter clock is asynchronous to the CPU clock, any read from any RTCNT1, RTCNT2, RTCNT3, RTCNT4, RT0PS, or RT1PS register should occur while the counter is not operating. Otherwise, the results may be unpredictable. Alternatively, the counter may be read multiple times while operating, and a majority vote taken in software to determine the correct reading. Any write to these registers takes effect immediately.

NOTE: For reliable update to all Counter Mode registers

Depending on the cascading of counters, when a write occurs, hold all subsequent counters. For example, if RTPS0 is being updated, set RTCPS1HOLD = 1, and if RTPS1 is being updated, set RTCHOLD = 1.

1.3.1.3 Real-Time Clock Interrupts in Counter Mode

In counter mode, four interrupt sources are available: RT0PSIFG, RT1PSIFG, RTCTEVIFG, and RTCOFIFG. RTCAIFG and RTCRDYIFG are cleared. RTCRDYIE and RTCAIE are don't care.

RT0PSIFG can be used to generate interrupt intervals selectable by the RT0IP bits. In counter mode, divide ratios of /2, /4, /8, /16, /32, /64, /128, and /256 of the clock source are possible. Setting the RT0PSIE bit enables the interrupt.

RT1PSIFG can be used to generate interrupt intervals selectable by the RT1IP bits. In counter mode, RT1PS is sourced with low-frequency oscillator clock, or the output of RT0PS, so divide ratios of /2, /4, /8, /16, /32, /64, /128, and /256 of the respective clock source are possible. Setting the RT1PSIE bit enables the interrupt.

In Counter Mode, the RTC_C module provides for an interval timer that sources real-time clock interrupt, RTCTEVIFG. The interval timer can be selected to cause an interrupt event when an 8-bit, 16-bit, 24-bit, or 32-bit overflow occurs within the 32-bit counter. The event is selectable with the RTCTEV bits. Setting the RTCTEVIE bit enables the interrupt.

The RTCOFIFG bit flags a failure of the 32-kHz crystal oscillator. It's main purpose is to wake-up the CPU from LPM3.5 in case an oscillator failure occurred.

1.3.2 Real-Time Clock Event/Tamper Detection With Time Stamp

NOTE: This feature is available only on selected devices. See the device-specific data sheet to determine if this feature is available.

The RTC_C module provides an external event or tamper detection and time stamp for up to two external events. The pins RTCCAP0 and RTCCAP1 [(1)] can be used as an event or tamper detection input of an external switch (mechanical or electronic). After device power-up, this feature can be enabled by setting the TCEN bit in the RTCTCCTL0 register. Event and tamper detection with time stamp is supported in all MSP430 operating modes, as long as there is a valid RTC power supply.

- When there is an event on RTCCAPx pin and the time capture feature is enabled (TCEN = 1), the corresponding CAPEV bit in RTCCAPxCTL register is set and the corresponding time stamp information (seconds, minutes, hours, day of month, month and year) is stored in the respective backup registers (RTCSECBAKx, RTCMINBAKx, RTCHOURBAKx, RTCDAYBAKx, RTCMONBAKx and RTCYEARBAKx).
- In case of multiple events, **ONLY** the time stamp of the event that occurred first is stored in the respective backup registers. After CAPEV is set by the first event on RTCCAPx, all subsequent events on RTCCAPx are ignored until the CAPEV bit is cleared by the user.
- The CAPES bit in the RTCCAPxCTL register sets the event edge for the corresponding RTCCAPx pin.
  - Bit = 0: CAPEV flag is set with a low-to-high transition.
  - Bit = 1: CAPEV flag is set with a high-to-low transition.

NOTE: **Writing to CAPESx**

Writing to CAPES can result in setting the corresponding interrupt flags.

CAPESx	RTCCAPx	RTCCAPIFG
0 → 1	0	May be set
0 → 1	1	Unchanged
1 → 0	0	Unchanged
1 → 0	1	May be set

- The interrupt flag RTCCAPIFG is set when any of the individual CAPEV bits are set. If the RTCIV is read, RTCCAPIFG is cleared but not the status flags (CAPEV bits). They are then read by the CPU and must be cleared by software only.
- By setting the RTCCAPIE bit, an event on RTCCAPx generates an interrupt. This interrupt can be used as LPM3.5 or LPM4.5 wake-up event in modules that support LPM3.5 or LPM4.5.
- When the time capture feature is enabled (TCEN = 1), all of the backup registers (RTCSECBAKx, RTCMINBAKx, RTCHOURBAKx, RTCDAYBAKx, RTCMONBAKx, and RTCYEARBAKx) are read-only to user and can be written only by the RTC hardware. When RTCBCD = 1 and TCEN = 1, BCD format is selected for the backup registers. If the backup registers were written to by the hardware before TCEN was set, then the previous values are retained until there is a time capture event that overrides the values with the time stamp.
- When the time capture feature is disabled (TCEN = 0), all of the backup registers (RTCSECBAKx, RTCMINBAKx, RTCHOURBAKx, RTCDAYBAKx, RTCMONBAKx, and RTCYEARBAKx) can be written only by the CPU. When TCEN = 0, the RTCBCD bit setting is ignored for the backup registers. The data in the backup registers when TCEN = 1 is retained until the user writes new values after TCEN is cleared.
- When TCEN is cleared, all CAPEV bits and RTCCAPIFG are cleared.
- Table 1-1 shows how to use the DIR, REN, and OUT bits in RTCCAPxCTL for proper configuration of RTCCAPx pins.

[(1)] These pins are present only on devices that support this feature of RTC_C. Refer to the device-specific data sheet to determine the availability of this feature.

Table 1-1. RTCCAPx Pin Configuration

DIR	REN	OUT	RTCCAPx Configuration
0	0	x	Input
0	1	0	Input with pulldown resistor
0	1	1	Input with pullup resistor
1	x	x	Output

1.3.2.1 Real-Time Clock Event/Tamper Detection Interrupts

With the event or tamper detection feature, one additional interrupt sources is available, RTCCAPIFG. This flag is prioritized and combined with the other interrupt flags to source a single interrupt vector. The interrupt vector register (RTCIV) is used to determine which flag requested an interrupt.

The RTCCAPIFG bit flags the occurrence of a tamper event. The exact source of the interrupt among multiple tamper events can be found out by reading the CAPEV bit in the respective RTCCAPxCTL registers (one per tamper source). When RTCIV is read, the RTCCAPIFG is cleared but not the status flags (CAPEV bits).

1.4 RTC_C Registers

The RTC_C module registers are shown in Table 1-2. This table also shows which registers are key protected and which are retained during LPM3.5. The registers that are retained during LPM3.5 and given with a reset value are not reset on POR; they are reset based on a signal derived from the RTC supply. Registers that are not retained during LPM3.5 must be restored after exit from LPM3.5.

The high-side SVS must not be disabled by software if the real-time clock feature is needed. When the high-side SVS is disabled, the RTC_C registers with LPM3.5 retention are not accessible by the CPU.

The base address for the RTC_C module registers can be found in the device-specific data sheet. The address offsets are shown in Table 1-2.

The additional registers that are available if Event/Tamper Detection is implemented are shown in Table 1-3 together with the corresponding address offsets.

If the counter mode is supported, the register aliases shown in Table 1-4 can be used to access the counter registers.

NOTE: Most registers have word or byte register access. For a generic register *ANYREG*, the suffix "_L" (*ANYREG_L*) refers to the lower byte of the register (bits 0 through 7). The suffix "_H" (*ANYREG_H*) refers to the upper byte of the register (bits 8 through 15).

Table 1-2. RTC_C Registers

Offset	Acronym	Register Name	Type	Access	Reset	Key Protected	LPM3.5 Retention
00h	RTCCTL0	Real-Time Clock Control 0	Read/write	Word	9600h	yes	not retained
00h	RTCCTL0_L	Real-Time Clock Control 0 Low	Read/write	Byte	00h	yes	not retained
01h	RTCCTL0_H	Real-Time Clock Control 0 High	Read/write	Byte	96h	n/a	not retained
02h	RTCCTL13	Real-Time Clock Control 1, 3	Read/write	Word	0070h	yes	high byte retained
02h	RTCCTL1	Real-Time Clock Control 1	Read/write	Byte	70h	yes	not retained
	or RTCCTL13_L						
03h	RTCCTL3	Real-Time Clock Control 3	Read/write	Byte	00h	yes	retained
	or RTCCTL13_H						
04h	RTCOCAL	Real-Time Clock Offset Calibration	Read/write	Word	0000h	yes	retained
04h	RTCOCAL_L		Read/write	Byte	00h	yes	retained
05h	RTCOCAL_H		Read/write	Byte	00h	yes	retained
06h	RTCTCMP	Real-Time Clock Temperature Compensation	Read/write	Word	4000h	no	retained
06h	RTCTCMP_L		Read/write	Byte	00h	no	retained
07h	RTCTCMP_H		Read/write	Byte	40h	no	retained
08h	RTCPS0CTL	Real-Time Prescale Timer 0 Control	Read/write	Word	0100h	no	not retained
08h	RTCPS0CTL_L		Read/write	Byte	00h	no	not retained
09h	RTCPS0CTL_H		Read/write	Byte	01h	no	not retained
0Ah	RTCPS1CTL	Real-Time Prescale Timer 1 Control	Read/write	Word	0100h	no	not retained
0Ah	RTCPS1CTL_L		Read/write	Byte	00h	no	not retained
0Bh	RTCPS1CTL_H		Read/write	Byte	01h	no	not retained
0Ch	RTCPS	Real-Time Prescale Timer 0, 1 Counter	Read/write	Word	none	yes	retained
0Ch	RT0PS	Real-Time Prescale Timer 0 Counter	Read/write	Byte	none	yes	retained
	or RTCPS_L						

Table 1-2. RTC_C Registers (continued)

Offset	Acronym	Register Name	Type	Access	Reset	Key Protected	LPM3.5 Retention
0Dh	RT1PS	Real-Time Prescale Timer 1 Counter	Read/write	Byte	none	yes	retained
	or RTCPS_H						
0Eh	RTCIV	Real Time Clock Interrupt Vector	Read	Word	0000h	no	not retained
10h	RTCTIM0	Real-Time Clock Seconds, Minutes	Read/write	Word	undefined	yes	retained
10h	RTCSEC	Real-Time Clock Seconds	Read/write	Byte	undefined	yes	retained
	or RTCTIM0_L						
11h	RTCMIN	Real-Time Clock Minutes	Read/write	Byte	undefined	yes	retained
	or RTCTIM0_H						
12h	RTCTIM1	Real-Time Clock Hour, Day of Week	Read/write	Word	undefined	yes	retained
12h	RTCHOUR	Real-Time Clock Hour	Read/write	Byte	undefined	yes	retained
	or RTCTIM1_L						
13h	RTCDOW	Real-Time Clock Day of Week	Read/write	Byte	undefined	yes	retained
	or RTCTIM1_H						
14h	RTCDATE	Real-Time Clock Date	Read/write	Word	undefined	yes	retained
14h	RTCDAY	Real-Time Clock Day of Month	Read/write	Byte	undefined	yes	retained
	or RTCDATE_L						
15h	RTCMON	Real-Time Clock Month	Read/write	Byte	undefined	yes	retained
	or RTCDATE_H						
16h	RTCYEAR	Real-Time Clock Year[1]	Read/write	Word	undefined	yes	retained
18h	RTCAMINHR	Real-Time Clock Minutes, Hour Alarm	Read/write	Word	undefined	no	retained
18h	RTCAMIN	Real-Time Clock Minutes Alarm	Read/write	Byte	undefined	no	retained
	or RTCAMINHR_L						
19h	RTCAHOUR	Real-Time Clock Hours Alarm	Read/write	Byte	undefined	no	retained
	or RTCAMINHR_H						
1Ah	RTCADOWDAY	Real-Time Clock Day of Week, Day of Month Alarm	Read/write	Word	undefined	no	retained
1Ah	RTCADOW	Real-Time Clock Day of Week Alarm	Read/write	Byte	undefined	no	retained
	or RTCADOWDAY_L						
1Bh	RTCADAY	Real-Time Clock Day of Month Alarm	Read/write	Byte	undefined	no	retained
	or RTCADOWDAY_H						
1Ch	BIN2BCD	Binary-to-BCD conversion register	Read/write	Word	0000h	no	not retained
1Eh	BCD2BIN	BCD-to-binary conversion register	Read/write	Word	0000h	no	not retained

[1] Do not access the year register RTCYEAR in byte mode.

Table 1-3. RTC_C Event and Tamper Detection Registers

Offset	Acronym	Register Name	Type	Access	Reset	Key Protected	LPM3.5 Retention
20h	RTCTCCTL0	Real-Time Clock Time Capture Control Register 0	Read/write	Byte	02h	yes	retained
21h	RTCTCCTL1	Real-Time Clock Time Capture Control Register 1	Read/write	Byte	00h	yes	not retained
22h	RTCCAP0CTL	Tamper Detect Pin 0 Control Register	Read/write	Byte	00h	yes	not retained
23h	RTCCAP1CTL	Tamper Detect Pin 1 Control Register	Read/write	Byte	00h	yes	not retained
30h	RTCSECBAK0	Real-Time Clock Seconds Backup Register 0	Read/write	Byte	00h	yes	retained
31h	RTCMINBAK0	Real-Time Clock Minutes Backup Register 0	Read/write	Byte	00h	yes	retained
32h	RTCHOURBAK0	Real-Time Clock Hours Backup Register 0	Read/write	Byte	00h	yes	retained
33h	RTCDAYBAK0	Real-Time Clock Days Backup Register 0	Read/write	Byte	00h	yes	retained
34h	RTCMONBAK0	Real-Time Clock Months Backup Register 0	Read/write	Byte	00h	yes	retained
36h	RTCYEARBAK0	Real-Time Clock year Backup Register 0	Read/write	Word	00h	yes	retained
38h	RTCSECBAK1	Real-Time Clock Seconds Backup Register 1	Read/write	Byte	00h	yes	retained
39h	RTCMINBAK1	Real-Time Clock Minutes Backup Register 1	Read/write	Byte	00h	yes	retained
3Ah	RTCHOURBAK1	Real-Time Clock Hours Backup Register 1	Read/write	Byte	00h	yes	retained
3Bh	RTCDAYBAK1	Real-Time Clock Days Backup Register 1	Read/write	Byte	00h	yes	retained
3Ch	RTCMONBAK1	Real-Time Clock Months Backup Register 1	Read/write	Byte	00h	yes	retained
3Eh	RTCYEARBAK1	Real-Time Clock Year Backup Register 1	Read/write	Word	00h	yes	retained

Table 1-4. RTC_C Real-Time Clock Counter Mode Aliases

Offset	Acronym	Register Name	Type	Access	Reset	Key Protected	LPM3.5 Retention
10h	RTCCNT12	Real-Time Counter 1, 2	Read/write	Word	undefined	yes	retained
10h	RTCCNT1	Real-Time Counter 1	Read/write	Byte	undefined	yes	retained
11h	RTCCNT2	Real-Time Counter 2	Read/write	Byte	undefined	yes	retained
12h	RTCCNT34	Real-Time Counter 3, 4	Read/write	Word	undefined	yes	retained
12h	RTCCNT3	Real-Time Counter 3	Read/write	Byte	undefined	yes	retained
13h	RTCCNT4	Real-Time Counter 4	Read/write	Byte	undefined	yes	retained

1.4.1 RTCCTL0_L Register

Real-Time Clock Control 0 Low Register

Figure 1-4. RTCCTL0_L Register

7	6	5	4	3	2	1	0
RTCOFIE[(1)]	RTCTEVIE[(1)]	RTCAIE[(1)]	RTCRDYIE	RTCOFIFG	RTCTEVIFG	RTCAIFG	RTCRDYIFG
rw-0	rw-0	rw-0	rw-0	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[(1)] The configuration of these bits is retained during LPMx.5 until LOCKLPM5 is cleared, but not the register bits themselves; therefore, reconfiguration is required after wakeup from LPMx.5 before clearing LOCKLPM5.

Table 1-5. RTCCTL0_L Register Description

Bit	Field	Type	Reset	Description
7	RTCOFIE	RW	0h	32-kHz crystal oscillator fault interrupt enable. This interrupt can be used as LPM3.5 wake-up event. 0b = Interrupt not enabled 1b = Interrupt enabled (LPM3.5 wake-up enabled)
6	RTCTEVIE	RW	0h	Real-time clock time event interrupt enable. In modules supporting LPM3.5 this interrupt can be used as LPM3.5 wake-up event. 0b = Interrupt not enabled 1b = Interrupt enabled (LPM3.5 wake-up enabled)
5	RTCAIE	RW	0h	Real-time clock alarm interrupt enable. In modules supporting LPM3.5 this interrupt can be used as LPM3.5 wake-up event. 0b = Interrupt not enabled 1b = Interrupt enabled (LPM3.5 wake-up enabled)
4	RTCRDYIE	RW	0h	Real-time clock ready interrupt enable 0b = Interrupt not enabled 1b = Interrupt enabled
3	RTCOFIFG	RW	0h	32-kHz crystal oscillator fault interrupt flag. This interrupt can be used as LPM3.5 wake-up event. It also indicates a clock failure during backup operation. 0b = No interrupt pending 1b = Interrupt pending. A 32-kHz crystal oscillator fault occurred after last reset.
2	RTCTEVIFG	RW	0h	Real-time clock time event interrupt flag. In modules supporting LPM3.5 this interrupt can be used as LPM3.5 wake-up event. 0b = No time event occurred 1b = Time event occurred
1	RTCAIFG	RW	0h	Real-time clock alarm interrupt flag. In modules supporting LPM3.5 this interrupt can be used as LPM3.5 wake-up event. 0b = No time event occurred 1b = Time event occurred
0	RTCRDYIFG	RW	0h	Real-time clock ready interrupt flag 0b = RTC cannot be read safely 1b = RTC can be read safely

1.4.2 RTCCTL0_H Register

Real-Time Clock Control 0 High Register

Figure 1-5. RTCCTL0_H Register

7	6	5	4	3	2	1	0
RTCKEY							
rw-1	rw-0	rw-0	rw-1	rw-0	rw-1	rw-1	rw-0

Table 1-6. RTCCTL0_H Register Description

Bit	Field	Type	Reset	Description
7-0	RTCKEY	RW	96h	Real-time clock key. This register should be written with A5h to unlock RTC_C. A write with a value other than A5h locks the module. A read from this register always returns 96h.

1.4.3 RTCCTL1 Register

Real-Time Clock Control Register 1

Figure 1-6. RTCCTL1 Register

7	6	5	4	3	2	1	0
RTCBCD	RTCHOLD[1]	RTCMODE[1]	RTCRDY	RTCSSELx[1]		RTCTEVx[1]	
rw-(0)	rw-(1)	rw-(1)	r-(1)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] The configuration of these bits is retained during LPMx.5 until LOCKLPM5 is cleared, but not the register bits themselves; therefore, reconfiguration is required after wakeup from LPMx.5 before clearing LOCKLPM5.

Table 1-7. RTCCTL1 Register Description

Bit	Field	Type	Reset	Description
7	RTCBCD	RW	0h	Real-time clock BCD select. Selects BCD counting for real-time clock. Applies to calendar mode (RTCMODE = 1) only; setting is ignored in counter mode. 0b = Binary (hexadecimal) code selected 1b = Binary coded decimal (BCD) code selected
6	RTCHOLD	RW	1h	Real-time clock hold 0b = Real-time clock (32-bit counter or calendar mode) is operational. 1b = In counter mode (RTCMODE = 0), only the 32-bit counter is stopped. In calendar mode (RTCMODE = 1), the calendar is stopped as well as the prescale counters, RT0PS and RT1PS. RT0PSHOLD and RT1PSHOLD are don't care.
5	RTCMODE	RW	1h	Real-time clock mode. In RTC_C modules without counter mode support this bit is read-only and always reads 1. 0b = 32-bit counter mode 1b = Calendar mode. Switching between counter and calendar mode does not reset the real-time clock counter registers. These registers must be configured by user software before use.
4	RTCRDY	R	1h	Real-time clock ready 0b = RTC time values in transition (calendar mode only) 1b = RTC time values safe for reading (calendar mode only). This bit indicates when the real-time clock time values are safe for reading (calendar mode only). In counter mode, RTCRDY remains cleared.
3-2	RTCSSELx	RW	0h	Real-time clock source select. In counter mode, selects clock input source to the 32-bit counter. In calendar mode, these bits are don't care. The clock input is automatically set to the output of RT1PS. 00b = 32-kHz crystal oscillator clock 01b = 32-kHz crystal oscillator clock 10b = Output from RT1PS 11b = Output from RT1PS
1-0	RTCTEVx	RW	0h	Real-time clock time event Calendar Mode (RTCMODE = 1) 00b = Minute changed 01b = Hour changed 10b = Every day at midnight (00:00) 11b = Every day at noon (12:00) Counter Mode (RTCMODE = 0) 00b = 8-bit overflow 01b = 16-bit overflow 10b = 24-bit overflow 11b = 32-bit overflow

1.4.4 RTCCTL3 Register

Real-Time Clock Control 3 Register

Figure 1-7. RTCCTL3 Register

7	6	5	4	3	2	1	0
Reserved					RTCLOCK[1]	RTCCALFx[2]	
r0	r0	r0	r0	r0	rw-[0]	rw-(0)	rw-(0)

[1] This bit is implemented in only the F677xA, F673xA, and F672xA devices. For other devices, this bit is reserved and always reads as 0.

[2] These bits are not reset on POR; they are reset based on a signal derived from the AUXVCC3 supply voltage level.

Table 1-8. RTCCTL3 Register Description

Bit	Field	Type	Reset	Description
7-3	Reserved	R	0h	Reserved. Always reads as 0.
2	RTCLOCK[1]	RW	0h	Real-time clock lock. When this bit is set, all control in LPM3.5 retention is held and not accessible even when the device is under BOR. 0b = LPM3.5 retention logic unlocked 1b = LPM3.5 retention logic locked
1-0	RTCCALFx	RW	0h	Real-time clock calibration frequency. Selects frequency output to RTCCLK pin for calibration measurement. The corresponding port must be configured for the peripheral module function. The RTCCLK is not available in counter mode and remains low, and the RTCCALF bits are don't care. 00b = No frequency output to RTCCLK pin 01b = 512 Hz 10b = 256 Hz 11b = 1 Hz

[1] This bit is implemented in only the F677xA, F673xA, and F672xA devices. For other devices, this bit is reserved and always reads as 0.

1.4.5 RTCOCAL Register

Real-Time Clock Offset Calibration Register

Figure 1-8. RTCOCAL Register

15	14	13	12	11	10	9	8
RTCOCALS[1]	Reserved						
rw-(0)	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
RTCOCALx[1]							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the AUXVCC3 supply voltage level.

Table 1-9. RTCOCAL Register Description

Bit	Field	Type	Reset	Description
15	RTCOCALS	RW	0h	Real-time clock offset error calibration sign. This bit decides the sign of offset error calibration. 0b = Down calibration. Frequency adjusted down. 1b = Up calibration. Frequency adjusted up.
14-8	Reserved	R	0h	Reserved. Always reads as 0.
7-0	RTCOCALx	RW	0h	Real-time clock offset error calibration. Each LSB represents approximately +1 ppm (RTCOCALS = 1) or –1 ppm (RTCOCALS = 0) adjustment in frequency. Maximum effective calibration value is ±240 ppm. Excess values written above ±240 ppm are ignored by hardware.

1.4.6 RTCTCMP Register

Real-Time Clock Temperature Compensation Register

Figure 1-9. RTCTCMP Register

15	14	13	12	11	10	9	8
RTCTCMPS[1]	RTCTCRDY[1]	RTCTCOK[1]	Reserved				
rw-(0)	r-(1)	r-(0)	r0	r0	r0	r0	r0
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RTCTCMPx[1]							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the AUXVCC3 supply voltage level.

Table 1-10. RTCTCMP Register Description

Bit	Field	Type	Reset	Description
15	RTCTCMPS	RW	0h	Real-time clock temperature compensation sign. This bit decides the sign of temperature compensation.[1] 0b = Down calibration. Frequency adjusted down. 1b = Up calibration. Frequency adjusted up.
14	RTCTCRDY	R	1h	Real-time clock temperature compensation ready. This is a read only bit that indicates when the RTCTCMPx can be written. Write to RTCTCMPx should be avoided when RTCTCRDY is reset.
13	RTCTCOK	R	0h	Real-time clock temperature compensation write OK. This is a read-only bit that indicates if the write to RTCTCMP is successful or not. 0b = Write to RTCTCMPx is unsuccessful 1b = Write to RTCTCMPx is successful
12-8	Reserved	R	0h	Reserved. Always reads as 0.
7-0	RTCTCMPx	RW	0h	Real-time clock temperature compensation. Value written into this register is used for temperature compensation of RTC_C. Each LSB represents approximately +1 ppm (RTCTCMPS = 1) or –1 ppm (RTCTCMPS = 0) adjustment in frequency. Maximum effective calibration value is ±240 ppm. Excess values written above ±240 ppm are ignored by hardware.

[1] Changing the sign-bit by writing to RTCTCMP_H becomes effective only after also writing RTCTCMP_L.

1.4.7 RTCNT1 Register

Real-Time Clock Counter 1 Register – Counter Mode

Figure 1-10. RTCNT1 Register

7	6	5	4	3	2	1	0
RTCNT1							
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-11. RTCNT1 Register Description

Bit	Field	Type	Reset	Description
7-0	RTCNT1	RW	undefined	The RTCNT1 register is the count of RTCNT1

1.4.8 RTCNT2 Register

Real-Time Clock Counter 2 Register – Counter Mode

Figure 1-11. RTCNT2 Register

7	6	5	4	3	2	1	0
RTCNT2							
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-12. RTCNT2 Register Description

Bit	Field	Type	Reset	Description
7-0	RTCNT2	RW	undefined	The RTCNT2 register is the count of RTCNT2

1.4.9 RTCNT3 Register

Real-Time Clock Counter 3 Register – Counter Mode

Figure 1-12. RTCNT3 Register

7	6	5	4	3	2	1	0
RTCNT3							
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-13. RTCNT3 Register Description

Bit	Field	Type	Reset	Description
7-0	RTCNT3	RW	undefined	The RTCNT3 register is the count of RTCNT3

1.4.10 RTCNT4 Register

Real-Time Clock Counter 4 Register – Counter Mode

Figure 1-13. RTCNT4 Register

7	6	5	4	3	2	1	0
RTCNT4							
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-14. RTCNT4 Register Description

Bit	Field	Type	Reset	Description
7-0	RTCNT4	RW	undefined	The RTCNT4 register is the count of RTCNT4.

1.4.11 RTCSEC Register – Calendar Mode With Hexadecimal Format

Real-Time Clock Seconds Register – Calendar Mode With Hexadecimal Format

Figure 1-14. RTCSEC Register

7	6	5	4	3	2	1	0
0		Seconds					
r-0	r-0	rw	rw	rw	rw	rw	rw

Table 1-15. RTCSEC Register Description

Bit	Field	Type	Reset	Description
7-6	0	R	0h	Always 0
5-0	Seconds	RW	undefined	Seconds (0 to 59)

1.4.12 RTCSEC Register – Calendar Mode With BCD Format

Real-Time Clock Seconds Register – Calendar Mode With BCD Format

Figure 1-15. RTCSEC Register

7	6	5	4	3	2	1	0
0	Seconds – high digit			Seconds – low digit			
r-0	rw	rw	rw	rw	rw	rw	rw

Table 1-16. RTCSEC Register Description

Bit	Field	Type	Reset	Description
7	0	R	0h	Always 0
6-4	Seconds – high digit	RW	undefined	Seconds – high digit (0 to 5)
3-0	Seconds – low digit	RW	undefined	Seconds – low digit (0 to 9)

1.4.13 RTCMIN Register – Calendar Mode With Hexadecimal Format

Real-Time Clock Minutes Register – Calendar Mode With Hexadecimal Format

Figure 1-16. RTCMIN Register

7	6	5	4	3	2	1	0
0		Minutes					
r-0	r-0	rw	rw	rw	rw	rw	rw

Table 1-17. RTCMIN Register Description

Bit	Field	Type	Reset	Description
7-6	0	R	0h	Always 0
5-0	Minutes	RW	undefined	Minutes (0 to 59)

1.4.14 RTCMIN Register – Calendar Mode With BCD Format

Real-Time Clock Minutes Register – Calendar Mode With BCD Format

Figure 1-17. RTCMIN Register

7	6	5	4	3	2	1	0
0	Minutes – high digit			Minutes – low digit			
r-0	rw	rw	rw	rw	rw	rw	rw

Table 1-18. RTCMIN Register Description

Bit	Field	Type	Reset	Description
7	0	R	0h	Always 0
6-4	Minutes – high digit	RW	undefined	Minutes – high digit (0 to 5)
3-0	Minutes – low digit	RW	undefined	Minutes – low digit (0 to 9)

1.4.15 RTCHOUR Register – Calendar Mode With Hexadecimal Format

Real-Time Clock Hours Register – Calendar Mode With Hexadecimal Format

Figure 1-18. RTCHOUR Register

7	6	5	4	3	2	1	0
0			Hours				
r-0	r-0	r-0	rw	rw	rw	rw	rw

Table 1-19. RTCHOUR Register Description

Bit	Field	Type	Reset	Description
7-5	0	R	0h	Always 0
4-0	Hours	RW	undefined	Hours (0 to 23)

1.4.16 RTCHOUR Register – Calendar Mode With BCD Format

Real-Time Clock Hours Register – Calendar Mode With BCD Format

Figure 1-19. RTCHOUR Register

7	6	5	4	3	2	1	0
0		Hours – high digit		Hours – low digit			
r-0	r-0	rw	rw	rw	rw	rw	rw

Table 1-20. RTCHOUR Register Description

Bit	Field	Type	Reset	Description
7-6	0	R	0h	Always 0
5-4	Hours – high digit	RW	undefined	Hours – high digit (0 to 2)
3-0	Hours – low digit	RW	undefined	Hours – low digit (0 to 9)

1.4.17 RTCDOW Register – Calendar Mode

Real-Time Clock Day of Week Register – Calendar Mode

Figure 1-20. RTCDOW Register

7	6	5	4	3	2	1	0
0					Day of week		
r-0	r-0	r-0	r-0	r-0	rw	rw	rw

Table 1-21. RTCDOW Register Description

Bit	Field	Type	Reset	Description
7-3	0	R	0h	Always 0
2-0	Day of week	RW	undefined	Day of week (0 to 6)

1.4.18 RTCDAY Register – Calendar Mode With Hexadecimal Format

Real-Time Clock Day of Month Register – Calendar Mode With Hexadecimal Format

Figure 1-21. RTCDAY Register

7	6	5	4	3	2	1	0
0			Day of month				
r-0	r-0	r-0	rw	rw	rw	rw	rw

Table 1-22. RTCDAY Register Description

Bit	Field	Type	Reset	Description
7-5	0	R	0h	Always 0
4-0	Day of month	RW	undefined	Day of month (1 to 28, 29, 30, 31)

1.4.19 RTCDAY Register – Calendar Mode With BCD Format

Real-Time Clock Day of Month Register – Calendar Mode With BCD Format

Figure 1-22. RTCDAY Register

7	6	5	4	3	2	1	0
0		Day of month – high digit		Day of month – low digit			
r-0	r-0	rw	rw	rw	rw	rw	rw

Table 1-23. RTCDAY Register Description

Bit	Field	Type	Reset	Description
7-6	0	R	0h	
5-4	Day of month – high digit	RW	undefined	Day of month – high digit (0 to 3)
3-0	Day of month – low digit	RW	undefined	Day of month – low digit (0 to 9)

1.4.20 RTCMON Register – Calendar Mode With Hexadecimal Format

Real-Time Clock Month Register – Calendar Mode With Hexadecimal Format

Figure 1-23. RTCMON Register

7	6	5	4	3	2	1	0
0				Month			
r-0	r-0	r-0	r-0	rw	rw	rw	rw

Table 1-24. RTCMON Register Description

Bit	Field	Type	Reset	Description
7-4	0	R	0h	Always 0
3-0	Month	RW	undefined	Month (1 to 12)

1.4.21 RTCMON Register – Calendar Mode With BCD Format

Real-Time Clock Month Register – Calendar Mode With BCD Format

Figure 1-24. RTCMON Register

7	6	5	4	3	2	1	0
0			Month – high digit	Month – low digit			
r-0	r-0	r-0	rw	rw	rw	rw	rw

Table 1-25. RTCMON Register Description

Bit	Field	Type	Reset	Description
7-5	0	R	0h	Always 0
4	Month – high digit	RW	undefined	Month – high digit (0 or 1)
3-0	Month – low digit	RW	undefined	Month – low digit (0 to 9)

1.4.22 RTCYEAR Register – Calendar Mode With Hexadecimal Format

Real-Time Clock Year Low-Byte Register – Calendar Mode With Hexadecimal Format

Figure 1-25. RTCYEAR Register

15	14	13	12	11	10	9	8
0				Year – high byte			
r-0	r-0	r-0	r-0	rw	rw	rw	rw
7	6	5	4	3	2	1	0
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-26. RTCYEAR Register Description

Bit	Field	Type	Reset	Description
15-12	0	R	0h	Always 0
11-8	Year – high byte	RW	undefined	Year – high byte. Valid values for Year are 0 to 4095.
7-0	Year – low byte	RW	undefined	Year – low byte. Valid values for Year are 0 to 4095.

1.4.23 RTCYEAR Register – Calendar Mode With BCD Format

Real-Time Clock Year Low-Byte Register – Calendar Mode With BCD Format

Figure 1-26. RTCYEAR Register

15	14	13	12	11	10	9	8
0	Century – high digit			Century – low digit			
r-0	rw	rw	rw	rw	rw	rw	rw
7	6	5	4	3	2	1	0
Decade				Year – lowest digit			
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-27. RTCYEAR Register Description

Bit	Field	Type	Reset	Description
15	0	R	0h	Always 0
14-10	Century – high digit	RW	undefined	Century – high digit (0 to 4)
11-8	Century – low digit	RW	undefined	Century – low digit (0 to 9)
7-4	Decade	RW	undefined	Decade (0 to 9)
3-0	Year – lowest digit	RW	undefined	Year – lowest digit (0 to 9)

1.4.24 RTCAMIN Register – Calendar Mode With Hexadecimal Format

Real-Time Clock Minutes Alarm Register – Calendar Mode With Hexadecimal Format

Figure 1-27. RTCAMIN Register

7	6	5	4	3	2	1	0
AE	0	Minutes					
rw	r-0	rw	rw	rw	rw	rw	rw

Table 1-28. RTCAMIN Register Description

Bit	Field	Type	Reset	Description
7	AE	RW	undefined	Alarm enable 0b = This alarm register is disabled 1b = This alarm register is enabled
6	0	R	0h	Always 0.
5-0	Minutes	RW	undefined	Minutes (0 to 59)

1.4.25 RTCAMIN Register – Calendar Mode With BCD Format

Real-Time Clock Minutes Alarm Register – Calendar Mode With BCD Format

Figure 1-28. RTCAMIN Register

7	6	5	4	3	2	1	0
AE	Minutes – high digit			Minutes – low digit			
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-29. RTCAMIN Register Description

Bit	Field	Type	Reset	Description
7	AE	RW	undefined	Alarm enable 0b = This alarm register is disabled 1b = This alarm register is enabled
6-4	Minutes – high digit	RW	undefined	Minutes – high digit (0 to 5)
3-0	Minutes – low digit	RW	undefined	Minutes – low digit (0 to 9)

1.4.26 RTCAHOUR Register

Real-Time Clock Hours Alarm Register – Calendar Mode With Hexadecimal Format

Figure 1-29. RTCAHOUR Register

7	6	5	4	3	2	1	0
AE	0		Hours				
rw	r-0	r-0	rw	rw	rw	rw	rw

Table 1-30. RTCAHOUR Register Description

Bit	Field	Type	Reset	Description
7	AE	RW	undefined	Alarm enable 0b = This alarm register is disabled 1b = This alarm register is enabled
6-5	0	R	0h	Always 0
4-0	Hours	RW	undefined	Hours (0 to 23)

1.4.27 RTCAHOUR Register – Calendar Mode With BCD Format

Real-Time Clock Hours Alarm Register – Calendar Mode With BCD Format

Figure 1-30. RTCAHOUR Register

7	6	5	4	3	2	1	0
AE	0	Hours – high digit		Hours – low digit			
rw	r-0	rw	rw	rw	rw	rw	rw

Table 1-31. RTCAHOUR Register Description

Bit	Field	Type	Reset	Description
7	AE	RW	undefined	Alarm enable 0b = This alarm register is disabled 1b = This alarm register is enabled
6	0	R	0h	Always 0
5-4	Hours – high digit	RW	undefined	Hours – high digit (0 to 2)
3-0	Hours – low digit	RW	undefined	Hours – low digit (0 to 9)

1.4.28 RTCADOW Register – Calendar Mode

Real-Time Clock Day of Week Alarm Register – Calendar Mode

Figure 1-31. RTCADOW Register

7	6	5	4	3	2	1	0
AE	0				Day of week		
rw	r-0	r-0	r-0	r-0	rw	rw	rw

Table 1-32. RTCADOW Register Description

Bit	Field	Type	Reset	Description
7	AE	RW	undefined	Alarm enable 0b = This alarm register is disabled 1b = This alarm register is enabled
6-3	0	R	0h	Always 0
2-0	Day of week	RW	undefined	Day of week (0 to 6)

1.4.29 RTCADAY Register – Calendar Mode With Hexadecimal Format

Real-Time Clock Day of Month Alarm Register – Calendar Mode With Hexadecimal Format

Figure 1-32. RTCADAY Register

7	6	5	4	3	2	1	0
AE	0		Day of month				
rw	r-0	r-0	rw	rw	rw	rw	rw

Table 1-33. RTCADAY Register Description

Bit	Field	Type	Reset	Description
7	AE	RW	undefined	Alarm enable 0b = This alarm register is disabled 1b = This alarm register is enabled
6-5	0	R	0h	Always 0
4-0	Day of month	RW	undefined	Day of month (1 to 28, 29, 30, 31)

1.4.30 RTCADAY Register – Calendar Mode With BCD Format

Real-Time Clock Day of Month Alarm Register – Calendar Mode With BCD Format

Figure 1-33. RTCADAY Register

7	6	5	4	3	2	1	0
AE	0	Day of month – high digit		Day of month – low digit			
rw	r-0	rw	rw	rw	rw	rw	rw

Table 1-34. RTCADAY Register Description

Bit	Field	Type	Reset	Description
7	AE	RW	undefined	Alarm enable 0b = This alarm register is disabled 1b = This alarm register is enabled
6	0	R	0h	Always 0
5-4	Day of month – high digit	RW	undefined	Day of month – high digit (0 to 3)
3-0	Day of month – low digit	RW	undefined	Day of month – low digit (0 to 9)

1.4.31 RTCPS0CTL Register

Real-Time Clock Prescale Timer 0 Control Register

Figure 1-34. RTCPS0CTL Register

15	14	13	12	11	10	9	8
Reserved		RT0PSDIV[1]			Reserved		RT0PSHOLD[1]
r0	r0	rw-(0)	rw-(0)	rw-(0)	r0	r0	rw-(1)
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			RT0IP[1]			RT0PSIE	RT0PSIFG
r0	r0	r0	rw-(0)	rw-(0)	rw-(0)	rw-0	rw-(0)

[1] The configuration of these bits is retained during LPMx.5 until LOCKLPM5 is cleared, but not the register bits themselves; therefore, reconfiguration is required after wake-up from LPMx.5 before clearing LOCKLPM5.

Table 1-35. RTCPS0CTL Register Description

Bit	Field	Type	Reset	Description
15-14	Reserved	R	0h	Reserved. Always reads as 0.
13-11	RT0PSDIV	RW	0h	Prescale timer 0 clock divide. These bits control the divide ratio of the RT0PS counter. In real-time clock calendar mode, these bits are don't care for RT0PS and RT1PS. RT0PS clock output is automatically set to /256. RT1PS clock output is automatically set to /128. 000b = Divide by 2 001b = Divide by 4 010b = Divide by 8 011b = Divide by 16 100b = Divide by 32 101b = Divide by 64 110b = Divide by 128 111b = Divide by 256
10-9	Reserved	R	0h	Reserved. Always reads as 0.
8	RT0PSHOLD	RW	1h	Prescale timer 0 hold. In real-time clock calendar mode, this bit is don't care. RT0PS is stopped by the RTCHOLD bit. 0b = RT0PS is operational 1b = RT0PS is held
7-5	Reserved	R	0h	Reserved. Always reads as 0.
4-2	RT0IP	RW	0h	Prescale timer 0 interrupt interval 000b = Divide by 2 001b = Divide by 4 010b = Divide by 8 011b = Divide by 16 100b = Divide by 32 101b = Divide by 64 110b = Divide by 128 111b = Divide by 256
1	RT0PSIE	RW	0h	Prescale timer 0 interrupt enable 0b = Interrupt not enabled 1b = Interrupt enabled
0	RT0PSIFG	RW	0h	Prescale timer 0 interrupt flag 0b = No time event occurred 1b = Time event occurred

1.4.32 RTCPS1CTL Register

Real-Time Clock Prescale Timer 1 Control Register

Figure 1-35. RTCPS1CTL Register

15	14	13	12	11	10	9	8
RT1SSELx[1]		RT1PSDIVx[1]			Reserved		RT1PSHOLD[1]
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r0	r0	rw-(1)
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			RT1IPx[1]			RT1PSIE	RT1PSIFG
r0	r0	r0	rw-(0)	rw-(0)	rw-(0)	rw-0	rw-(0)

[1] The configuration of these bits is retained during LPMx.5 until LOCKLPM5 is cleared, but not the register bits themselves; therefore, reconfiguration is required after wake-up from LPMx.5 before clearing LOCKLPM5.

Table 1-36. RTCPS1CTL Register Description

Bit	Field	Type	Reset	Description
15-14	RT1SSELx	RW	0h	Prescale timer 1 clock source select. Selects clock input source to the RT1PS counter. In real-time clock calendar mode, these bits are do not care. RT1PS clock input is automatically set to the output of RT0PS. 00b = 32-kHz crystal oscillator clock 01b = 32-kHz crystal oscillator clock 10b = Output from RT0PS 11b = Output from RT0PS
13-11	RT1PSDIVx	RW	0h	Prescale timer 1 clock divide. These bits control the divide ratio of the RT0PS counter. In real-time clock calendar mode, these bits are don't care for RT0PS and RT1PS. RT0PS clock output is automatically set to /256. RT1PS clock output is automatically set to /128. 000b = Divide by 2 001b = Divide by 4 010b = Divide by 8 011b = Divide by 16 100b = Divide by 32 101b = Divide by 64 110b = Divide by 128 111b = Divide by 256
10-9	Reserved	R	0h	Reserved. Always reads as 0.
8	RT1PSHOLD	RW	1h	Prescale timer 1 hold. In real-time clock calendar mode, this bit is don't care. RT1PS is stopped by the RTCHOLD bit. 0b = RT1PS is operational 1b = RT1PS is held
7-5	Reserved	R	0h	Reserved. Always reads as 0.
4-2	RT1IPx	RW	0h	Prescale timer 1 interrupt interval 000b = Divide by 2 001b = Divide by 4 010b = Divide by 8 011b = Divide by 16 100b = Divide by 32 101b = Divide by 64 110b = Divide by 128 111b = Divide by 256
1	RT1PSIE	RW	0h	Prescale timer 1 interrupt enable 0b = Interrupt not enabled 1b = Interrupt enabled (LPMx.5 wake-up enabled)

Table 1-36. RTCPS1CTL Register Description (continued)

Bit	Field	Type	Reset	Description
0	RT1PSIFG	RW	0h	Prescale timer 1 interrupt flag. This interrupt can be used as LPMx.5 wake-up event. 0b = No time event occurred 1b = Time event occurred

1.4.33 RTCPS0 Register

Real-Time Clock Prescale Timer 0 Counter Register

Figure 1-36. RTCPS0 Register

7	6	5	4	3	2	1	0
RT0PS							
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-37. RTCPS0 Register Description

Bit	Field	Type	Reset	Description
7-0	RT0PS	RW	undefined	Prescale timer 0 counter value

1.4.34 RTCPS1 Register

Real-Time Clock Prescale Timer 1 Counter Register

Figure 1-37. RTCPS1 Register

7	6	5	4	3	2	1	0
RT1PS							
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-38. RTCPS1 Register Description

Bit	Field	Type	Reset	Description
7-0	RT1PS	RW	undefined	Prescale timer 1 counter value

1.4.35 RTCIV Register

Real-Time Clock Interrupt Vector Register

Figure 1-38. RTCIV Register

15	14	13	12	11	10	9	8
RTCIVx							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
RTCIVx							
r0	r0	r0	r-(0)	r-(0)	r-(0)	r-(0)	r0

Table 1-39. RTCIV Register Description

Bit	Field	Type	Reset	Description
15-0	RTCIVx	R	0h	Real-time clock interrupt vector value Without Event/Tamper Detection implemented: 00h = No interrupt pending 02h = Interrupt Source: RTC oscillator failure; Interrupt Flag: RTCOFIFG; Interrupt Priority: Highest 04h = Interrupt Source: RTC ready; Interrupt Flag: RTCRDYIFG 06h = Interrupt Source: RTC interval timer; Interrupt Flag: RTCTEVIFG 08h = Interrupt Source: RTC user alarm; Interrupt Flag: RTCAIFG 0Ah = Interrupt Source: RTC prescaler 0; Interrupt Flag: RT0PSIFG 0Ch = Interrupt Source: RTC prescaler 1; Interrupt Flag: RT1PSIFG 0Eh = Reserved 10h = Reserved ; Interrupt Priority: Lowest With Event/Tamper Detection implemented: 00h = No interrupt pending 02h = Interrupt Source: RTC oscillator failure; Interrupt Flag: RTCOFIFG; Interrupt Priority: Highest 04h = Interrupt Source: RTC Tamper Event; Interrupt Flag: RTCCAPIFG 06h = Interrupt Source: RTC ready; Interrupt Flag: RTCRDYIFG 08h = Interrupt Source: RTC interval timer; Interrupt Flag: RTCTEVIFG 0Ah = Interrupt Source: RTC user alarm; Interrupt Flag: RTCAIFG 0Ch = Interrupt Source: RTC prescaler 0; Interrupt Flag: RT0PSIFG 0Eh = Interrupt Source: RTC prescaler 1; Interrupt Flag: RT1PSIFG 10h = Reserved ; Interrupt Priority: Lowest

1.4.36 BIN2BCD Register

Binary-to-BCD Conversion Register

Figure 1-39. BIN2BCD Register

15	14	13	12	11	10	9	8
BIN2BCDx							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
BIN2BCDx							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 1-40. BIN2BCD Register Description

Bit	Field	Type	Reset	Description
15-0	BIN2BCDx	RW	0h	Read: 16-bit BCD conversion of previously written 12-bit binary number. Write: 12-bit binary number to be converted.

1.4.37 BCD2BIN Register

BCD-to-Binary Conversion Register

Figure 1-40. BCD2BIN Register

15	14	13	12	11	10	9	8
BCD2BINx							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
BCD2BINx							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Table 1-41. BCD2BIN Register Description

Bit	Field	Type	Reset	Description
15-0	BCD2BINx	RW	0h	Read: 12-bit binary conversion of previously written 16-bit BCD number. Write: 16-bit BCD number to be converted.

1.4.38 *RTCSECBAKx Register – Hexadecimal Format*

Real-Time Clock Seconds Backup Register – Hexadecimal Format

Figure 1-41. RTCSECBAKx Register

7	6	5	4	3	2	1	0
0	0	Seconds[1]					
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-42. RTCSECBAKx Register Description

Bit	Field	Type	Reset	Description
7-6	0	RW	0h	Always 0.
5-0	Seconds	RW	0h	Seconds. Valid values are 0 to 59.

1.4.39 *RTCSECBAKx Register – BCD Format*

Real-Time Clock Seconds Backup Register – BCD Format

Figure 1-42. RTCSECBAKx Register

7	6	5	4	3	2	1	0
0	Seconds – high digit[1]			Seconds – low digit[1]			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-43. RTCSECBAKx Register Description

Bit	Field	Type	Reset	Description
7	0	RW	0h	Always 0.
6-4	Seconds – high digit	RW	0h	Seconds – high digit. Valid values are 0 to 5.
3-0	Seconds – low digit	RW	0h	Seconds – low digit. Valid values are 0 to 9.

1.4.40 RTCMINBAKx Register – Hexadecimal Format

Real-Time Clock Minutes Backup Register – Hexadecimal Format

Figure 1-43. RTCMINBAKx Register

7	6	5	4	3	2	1	0
0	0	Minutes[1]					
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-44. RTCMINBAKx Register Description

Bit	Field	Type	Reset	Description
7-6	0	RW	0h	Always 0.
5-0	Minutes	RW	0h	Minutes. Valid values are 0 to 59.

1.4.41 RTCMINBAKx Register – BCD Format

Real-Time Clock Minutes Backup Register – BCD Format

Figure 1-44. RTCMINBAKx Register

7	6	5	4	3	2	1	0
0	Minutes – high digit[1]			Minutes – low digit[1]			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-45. RTCMINBAKx Register Description

Bit	Field	Type	Reset	Description
7	0	RW	0h	Always 0.
6-4	Minutes – high digit	RW	0h	Minutes – high digit. Valid values are 0 to 5.
3-0	Minutes – low digit	RW	0h	Minutes – low digit. Valid values are 0 to 9.

1.4.42 RTCHOURBAKx Register – Hexadecimal Format

Real-Time Clock Hours Backup Register – Hexadecimal Format

Figure 1-45. RTCHOURBAKx Register

7	6	5	4	3	2	1	0
0	0	0	Hours[1]				
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-46. RTCHOURBAKx Register Description

Bit	Field	Type	Reset	Description
7-5	0	RW	0h	Always 0.
4-0	Hours	RW	0h	Hours. Valid values are 0 to 23.

1.4.43 RTCHOURBAKx Register – BCD Format

Real-Time Clock Hours Backup Register – BCD Format

Figure 1-46. RTCHOURBAKx Register

7	6	5	4	3	2	1	0
0	0	Hours – high digit[1]		Hours – low digit[1]			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-47. RTCHOURBAKx Register Description

Bit	Field	Type	Reset	Description
7-6	0	RW	0h	Always 0.
5-4	Hours – high digit	RW	0h	Hours – high digit. Valid values are 0 to 2.
3-0	Hours – low digit	RW	0h	Hours – low digit. Valid values are 0 to 9.

1.4.44 RTCDAYBAKx Register – Hexadecimal Format

Real-Time Clock Day of Month Backup Register – Hexadecimal Format

Figure 1-47. RTCDAYBAKx Register

7	6	5	4	3	2	1	0
0	0	0	Day of month[1]				
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-48. RTCDAYBAKx Register Description

Bit	Field	Type	Reset	Description
7-5	0	RW	0h	Always 0.
4-0	Day of month	RW	0h	Day of month. Valid values are 1 to 31.

1.4.45 RTCDAYBAKx Register – BCD Format

Real-Time Clock Day of Month Backup Register – BCD Format

Figure 1-48. RTCDAYBAKx Register

7	6	5	4	3	2	1	0
0	0	Day of month – high digit[1]		Day of month – low digit[1]			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-49. RTCDAYBAKx Register Description

Bit	Field	Type	Reset	Description
7-6	0	RW	0h	Always 0.
5-4	Day of month – high digit	RW	0h	Day of month – high digit. Valid values are 0 to 3.
3-0	Day of month – low digit	RW	0h	Day of month – low digit. Valid values are 0 to 9.

1.4.46 RTCMONBAKx Register – Hexadecimal Format

Real-Time Clock Month Backup Register – Hexadecimal Format

Figure 1-49. RTCMONBAKx Register

7	6	5	4	3	2	1	0
0	0	0	0	Month[1]			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-50. RTCMONBAKx Register Description

Bit	Field	Type	Reset	Description
7-4	0	RW	0h	Always 0.
3-0	Month	RW	0h	Month. Valid values are 1 to 12.

1.4.47 RTCMONBAKx Register – BCD Format

Real-Time Clock Month Backup Register – BCD Format

Figure 1-50. RTCMONBAKx Register

7	6	5	4	3	2	1	0
0	0	Month – high digit[1]		Month – low digit[1]			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-51. RTCMONBAKx Register Description

Bit	Field	Type	Reset	Description
7-6	0	RW	0h	Always 0.
5-4	Month – high digit	RW	0h	Month – high digit. Valid values are 0 to 3.
3-0	Month – low digit	RW	0h	Month – low digit. Valid values are 0 to 9.

1.4.48 RTCYEARBAKx Register – Hexadecimal Format

Real-Time Clock Year Low-Byte Backup Register – Hexadecimal Format

Figure 1-51. RTCYEARBAKx Register

15	14	13	12	11	10	9	8
0	0	0	0	Year – high byte[1]			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-52. RTCYEARBAKx Register Description

Bit	Field	Type	Reset	Description
15-12	0	RW	0h	Always 0.
11-8	Year – high byte	RW	0h	Year – high byte. Valid values of Year are 0 to 4095.
7-0	Year – low byte	RW	0h	Year – low byte. Valid values of Year are 0 to 4095.

1.4.49 RTCYEARBAKx Register – BCD Format

Real-Time Clock Year Low-Byte Backup Register – BCD Format

Figure 1-52. RTCYEARBAKx Register

15	14	13	12	11	10	9	8
0	Century – high digit[1]			Century – low digit[1]			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
Decade[1]				Year – lowest digit[1]			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-53. RTCYEARBAKx Register Description

Bit	Field	Type	Reset	Description
15	0	RW	0h	Always 0.
14-12	Century – high digit	RW	0h	Century – high digit. Valid values are 0 to 4.
11-8	Century – low digit	RW	0h	Century – low digit. Valid values are 0 to 9.
7-4	Decade	RW	0h	Decade. Valid values are 0 to 9.
3-0	Year – lowest digit	RW	0h	Year – lowest digit. Valid values are 0 to 9.

1.4.50 RTCTCCTL0 Register

Real-Time Clock Time Capture Control Register 0

Figure 1-53. RTCTCCTL0 Register

7	6	5	4	3	2	1	0
Reserved						AUX3RST[1]	TCEN[1]
r-0	r-0	r-0	r-0	r-0	r-0	rw-(1)	rw-(0)

[1] These bits are not reset on POR; they are reset based on a signal derived from the RTC supply.

Table 1-54. RTCTCCTL0 Register Description

Bit	Field	Type	Reset	Description
7-2	Reserved	R	0h	Reserved. Always reads as 0.
1	AUX3RST	RW	1h	Indication of power cycle on AUXVCC3 0b = No power cycle on AUXVCC3 since the last clear by the User 1b = Indication of AUXVCC3 power cycle. Needs to be cleared by User to observe the next power cycle on AUXVCC3
0	TCEN	RW	0h	Enable for RTC tamper detection with time stamp 0b = Tamper detection with time stamp disabled 1b = Tamper detection with time stamp enabled

1.4.51 RTCTCCTL1 Register

Real-Time Clock Time Capture Control Register 1

Figure 1-54. RTCTCCTL1 Register

7	6	5	4	3	2	1	0
Reserved						RTCCAPIE	RTCCAPIFG
r-0	r-0	r-0	r-0	r-0	r-0	rw-(0)	rw-(0)

Table 1-55. RTCTCCTL1 Register Description

Bit	Field	Type	Reset	Description
7-2	Reserved	R	0h	Reserved. Always reads as 0.
1	RTCCAPIE	RW	0h	Tamper event interrupt enable. In modules that support LPM3.5 or LPM4.5, this interrupt can be used as LPM3.5 or LPM4.5 wake-up event. 0b = Interrupt not enabled 1b = Interrupt enabled (LPM3.5 and LPM4.5 wake-up enabled)
0	RTCCAPIFG	RW	0h	Common interrupt flag for all tamper events. In modules that support LPM3.5 or LPM4.5, this interrupt can be used as LPM3.5 or LPM4.5 wake-up event. 0b = Tamper event did not occur 1b = At least one tamper event occurred. Status of individual tamper events can be found from the CAPEV bit in RTCCAPxCTL.

1.4.52 RTCCAPxCTL Register

Tamper Detect Pin Control Register

Figure 1-55. RTCCAPxCTL Register

7	6	5	4	3	2	1	0
Reserved	OUT[1]	DIR[1]	IN[1]	REN[1]	CAPES[1]	Reserved	CAPEV[1]
r-0	rw-(0)	rw-(0)	r	rw-(0)	rw-(0)	r-0	r/w0

[1] The configuration of these bits is retained during LPMx.5 until LOCKLPM5 is cleared, but not the bits themselves; therefore, reconfiguration is required after wakeup from LPMx.5 before clearing LOCKLPM5.

Table 1-56. RTCCAPxCTL Register Description

Bit	Field	Type	Reset	Description
7	Reserved	R	0h	Reserved. Always reads as 0.
6	OUT	RW	0h	RTCCAPx output 0b = Output low 1b = Output high
5	DIR	RW	0h	RTCCAPx pin direction 0b = RTCCAPx pin configured as input 1b = RTCCAPx pin configured as output
4	IN	R	0h	RTCCAPx input. The external input on RTCCAPx pin can be read by this bit. 0b = Input is low 1b = Input is high
3	REN	RW	0h	RTCCAPx pin pullup or pulldown resistor enable. When respective pin is configured as input, setting this bit enables the pullup or pulldown (see Table 1-1). 0b = Pullup or pulldown disabled 1b = Pullup or pulldown enabled
2	CAPES	RW	0h	Event edge selection 0b = Event on a low-to-high transition 1b = Event on a high-to-low transition
1	Reserved	R	0h	Reserved. Always reads as 0.
0	CAPEV	RW	0h	Tamper event status flag. All subsequent events on RTCCAPx after CAPEV is set are ignored until CAPEV is cleared by the user. Can only be written as 0. 0b = Tamper event did not occur 1b = Tamper event occurred

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated (‘TI”) technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, “TI Resources”) are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI’s provision of TI Resources does not expand or otherwise alter TI’s applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED “AS IS” AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI’s standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265