Chapter 1
SLAU402F–August 2012–Revised March 2018

Real-Time Clock (RTC_A)

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 (RTC_A) module provides clock counters with a calendar, a flexible programmable alarm, and calibration. This chapter describes the RTC_A module.

Topic **Page**

1.1 RTC_A Introduction

The RTC_A module provides a real-time clock and calendar function that can also be configured as a general-purpose counter.

RTC_A features include:

- Configurable for real-time clock with calendar function or general-purpose counter
- Provides seconds, minutes, hours, day of week, day of month, month, and year in real-time clock with calendar function
- Interrupt capability
- Selectable BCD or binary format in real-time clock mode
- Programmable alarms in real-time clock mode
- Calibration logic for time offset correction in real-time clock mode

The RTC_A block diagram is shown in Figure 1-1.

NOTE: Real-time clock initialization

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

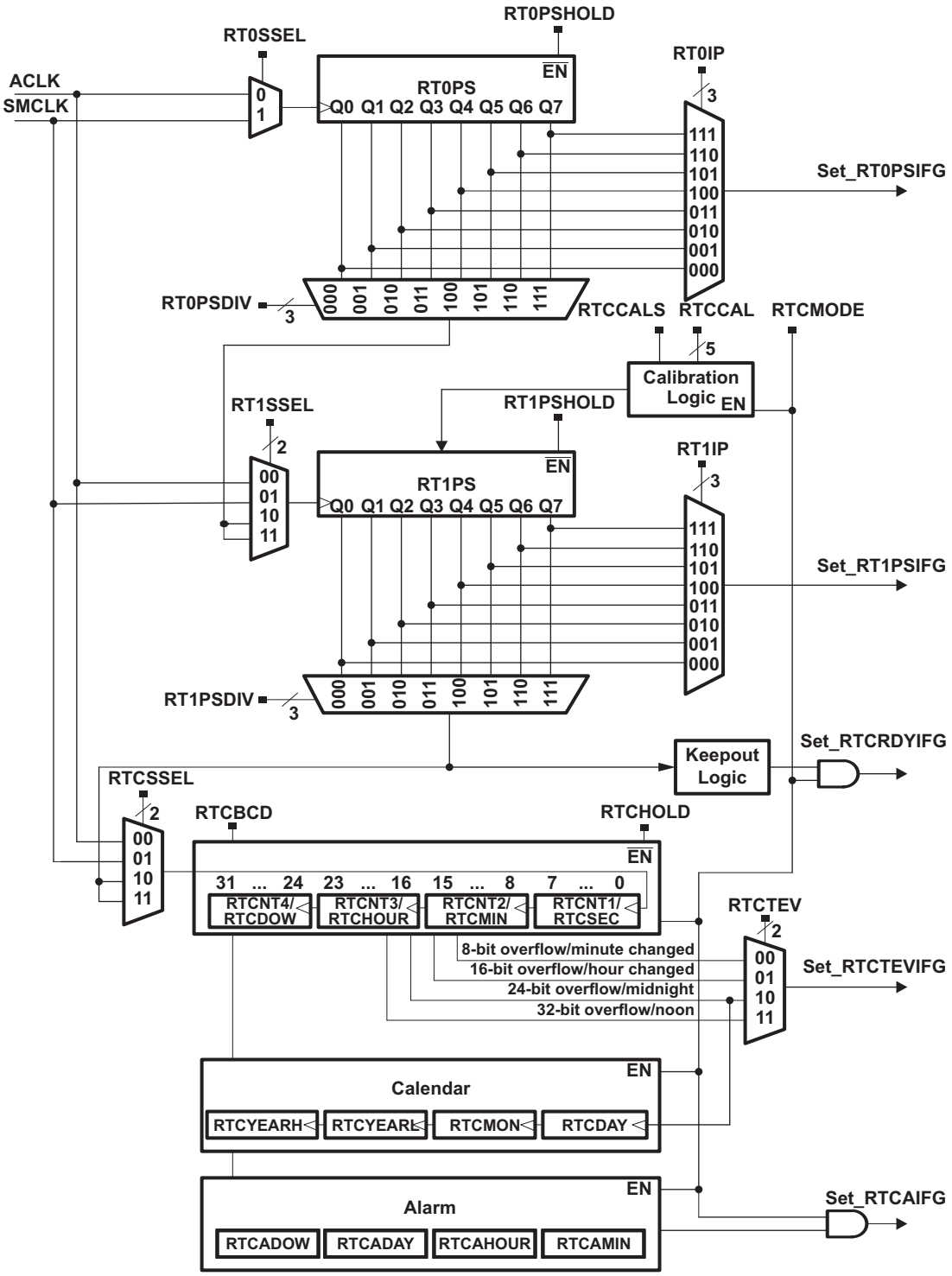

Figure 1-1. RTC_A

1.2 RTC_A Operation

The RTC_A 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.

1.2.1 Counter Mode

Counter mode is selected when RTCMODE is reset. In this mode, a 32-bit counter is provided that is directly accessible by software. Switching from calendar mode to counter mode resets the count value (RTCNT1, RTCNT2, RTCNT3, RTCNT4), as well as the prescale counters (RT0PS, RT1PS).

The clock to increment the counter can be sourced from ACLK, SMCLK, or prescaled versions of ACLK or SMCLK. Prescaled versions of ACLK or SMCLK are sourced from the prescale dividers (RT0PS and RT1PS). RT0PS and RT1PS output /2, /4, /8, 16, /32, /64, /128, and /256 versions of ACLK and SMCLK, respectively. The output of RT0PS can be cascaded with RT1PS. The cascaded output can 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 to.

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 from ACLK or SMCLK, 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.

1.2.2 Calendar Mode

Calendar mode is selected when RTCMODE is set. In calendar mode, the RTC_A 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.

1.2.2.1 Real-Time Clock and Prescale Dividers

The prescale dividers, RT0PS and RT1PS, are automatically configured to provide a 1-s clock interval for the RTC_A. RT0PS is sourced from ACLK. ACLK must be set to 32768 Hz (nominal) for proper RTC_A calendar operation. RT1PS is cascaded with the output ACLK/256 of RT0PS. The RTC_A is sourced with the /128 output of RT1PS, thereby providing the required 1-s interval. Switching from counter to calendar mode clears the seconds, minutes, hours, day-of-week, and year counts and sets day-of-month and month counts to 1. In addition, RT0PS and RT1PS are cleared.

When RTCBCD = 1, BCD format is selected for the calendar registers. The format must be selected before the time is set. Changing the state of RTCBCD clears the seconds, minutes, hours, day-of-week, and year counts and sets day-of-month and month counts to 1. In addition, RT0PS and RT1PS are cleared.

In calendar mode, the RT0SSEL, RT1SSEL, RT0PSDIV, RT1PSDIV, RT0PSHOLD, RT1PSHOLD, and RTCSSEL bits are don't care. Setting RTCHOLD halts the real-time counters and prescale counters, RT0PS and RT1PS.

1.2.2.2 Real-Time Clock Alarm Function

The RTC_A 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. The user-programmable alarm function is only available in the calendar mode of operation.

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, at 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 AF 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, etc.
- 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 AF 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. Once enabled, the AF is set when the 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. Once enabled, the AF is set when the 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. Once enabled, the AF is set when the 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 via hardware. It is the user's responsibility to ensure that valid alarm settings are entered.

NOTE: Invalid time and date values

Writing of invalid date and/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

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

1.2.2.3 Reading or Writing Real-Time Clock Registers in Calendar Mode

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

In calendar mode, the real-time clock registers are updated once per second. To prevent reading any real-time clock register at the time of an update, which could result in an invalid time being read, a keepout window is provided. The keepout window is centered approximately -128/32768 s around the update transition. The read-only RTCRDY bit is reset during the keepout window period and set outside the keepout 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. Once 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.

In counter mode, the RTCRDY bit remains reset. RTCRDYIE is a don't care and RTCRDYIFG remains reset.

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 s during a write. Writing of data outside the legal ranges or invalid time stamp combinations results in unpredictable behavior.

1.2.3 Real-Time Clock Interrupts

The RTC_A module has five interrupt sources available, each with independent enables and flags.

1.2.3.1 Real-Time Clock Interrupts in Calendar Mode

In calendar mode, five sources for interrupts are available, namely RT0PSIFG, RT1PSIFG, RTCRDYIFG, RTCTEVIFG, and RTCAIFG. 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.

Any access, read or write, of the RTCIV register automatically resets the highest-pending interrupt flag. If another interrupt flag is set, another interrupt is immediately generated after servicing the initial interrupt. In addition, all flags can be cleared via 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_A 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. In calendar mode, RT0PS is sourced with ACLK 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 generate interrupt intervals selectable by the RT1IP bits. In calendar mode, 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.

1.2.3.2 Real-Time Clock Interrupts in Counter Mode

In counter mode, three interrupt sources are available: RT0PSIFG, RT1PSIFG, and RTCTEVIFG. 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, RT0PS is sourced with ACLK or SMCLK, so divide ratios of /2, /4, /8, /16, /32, /64, /128, and /256 of the respective 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 ACLK, SMCLK, 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.

The RTC_A 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.

1.2.3.2.1 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.              Cycles
RTC_HND                         ; Interrupt latency          6
         ADD   &RTCIV,PC        ; Add offset to Jump table   3
         RETI                   ; Vector 0: No interrupt     5
         JMP   RTCRDYIFG_HND    ; Vector 2: RTCRDYIFG        2
         JMP   RTCTEVIFG_HND    ; Vector 4: RTCTEVIFG        2
         JMP   RTCAIFG          ; Vector 6: RTCAIFG          5
         JMP   RT0PSIFG         ; Vector 8: RT0PSIFG         5
         JMP   RT1PSIFG         ; Vector A: RT1PSIFG         5
         RETI                   ; Vector C: Reserved         5
RTCRDYIFG_HND                   ; Vector 2: RTCRDYIFG Flag
          to                     ; Task starts here
         RETI                                                5
RTCTEVIFG_HND                   ; Vector 4: RTCTEVIFG
          to                     ; Task starts here
         RETI                   ; Back to main program       5
RTCAIFG_HND                     ; Vector 6: RTCAIFG
          to                     ; Task starts here
RT0PSIFG_HND                    ; Vector 8: RT0PSIFG
          to                     ; Task starts here
RT1PSIFG_HND                    ; Vector A: RT1PSIFG
          to                     ; Task starts here
```

1.2.4 Real-Time Clock Calibration

The RTC_A module has calibration logic that allows for adjusting the crystal frequency in approximately +4-ppm or –2-ppm steps, allowing for higher time keeping accuracy from standard crystals.The RTCCAL bits are used to adjust the frequency. When RTCCALS is set, each RTCCAL LSB causes a ≈ +4-ppm adjustment. When RTCCALS is cleared, each RTCCAL LSB causes a ≈ –2-ppm adjustment. Calibration is available only in calendar mode. In counter mode (RTCMODE = 0), the calibration logic is disabled.

Calibration is accomplished by periodically adjusting the RT1PS counter based on the RTCCALS and RTCCALx settings. In calendar mode, the RT0PS divides the nominial 37268-Hz low-frequency (LF) crystal clock input by 256. A 64-minute period has 32768 cycles/sec × 60 sec/min × 64 min = 125829120 cycles. Therefore a –2-ppm reduction in frequency (down calibration) approximately equates to adding an additional 256 cycles every 125829120 cycles (256/125829120 = 2.035 ppm). This is accomplished by holding the RT1PS counter for one additional clock of the RT0PS output within a 64-minute period. Similary, a +4-ppm increase in frequency (up calibration) approximately equates to removing 512 cycles every 125829120 cycle (512/125829120 = 4.069 ppm). This is accomplished by incrementing the RT1PS counter for two additional clocks of the RT0PS output within a 64-minute period. Each RTCCALx calibration bit causes either 256 LF crystal clock cycles to be added every 64 minutes or 512 LF crystal clock cycles to be subtracted every 64 minutes, giving a frequency adjustment of approximately –2 ppm or +4 ppm, respectively.

To calibrate the frequency, the RTCCLK output signal is available at a pin. The RTCCALF bits can be used to select the frequency rate of the RTCCLK output signal, either no signal, 512 Hz, 256 Hz, or 1 Hz.

The basic flow to calibrate the frequency is as follows:

1. Configure the RTCCLK pin.
2. Measure the RTCCLK output signal with an appropriate resolution frequency counter; that is, within the resolution required.
3. Compute the absolute error in ppm: Absolute Error (ppm) = $|10^6 \times (f_{MEASURED} – f_{RTCCLK}) / f_{RTCCLK}|$, where f_{RTCCLK} is the expected frequency of 512 Hz, 256 Hz, or 1 Hz.
4. Adjust the frequency, by performing the following:
    1. If the frequency is too low, set RTCALS = 1 and apply the appropriate RTCCALx bits, where RTCCALx = (Absolute Error) / 4.069, rounded to the nearest integer.
    2. If the frequency is too high, clear RTCALS = 0 and apply the appropriate RTCCALx bits, where RTCCALx = (Absolute Error) / 2.035, rounded to the nearest integer.

For example, assume that RTCCLK is output at a frequency of 512 Hz. The measured RTCCLK is 511.9658 Hz. The frequency error is approximately 66.8 ppm low. To increase the frequency by 66.8 ppm, RTCCALS would be set, and RTCCAL would be set to 16 (66.8/4.069). Similarly, assume that the measured RTCCLK is 512.0125 Hz. The frequency error is approximately 24.4 ppm high. To decrease the frequency by 24.4 ppm, RTCCALS would be cleared, and RTCCAL would be set to 12 (24.4 / 2.035).

The calibration corrects only initial offsets and does not adjust for temperature and aging effects. This can be handled by periodically measuring temperature and using the crystal's charateristic curve to adjust the ppm based on temperature as required. In counter mode (RTCMODE = 0), the calibration logic is disabled.

NOTE: Minimum Possible Calibration

The minimial calibration possible is -4 ppm or +8 ppm. For example, setting RTCCALS = 0 and RTCCAL = 0h would result in a -4 ppm decrease in frequency. Similarly, setting RTCCALS = 1 and RTCCAL = 0h would result in a +8 ppm increase in frequency.

NOTE: Calibration output frequency

The 512-Hz and 256-Hz output frequencies observed at the RTCCLK pin are not affected by changes in the calibration settings since these output frequencies are generated prior to the calibration logic. The 1-Hz output frequency is affected by changes in the calibration settings. Because the frequency change is small and infrequent over a very long time interval, it can be difficult to observe.

1.3 RTC_A Registers

The RTC_A module registers are listed in and Table 1-1. The base register for the RTC_A module registers can be found in the device-specific data sheet. The address offsets are given in Table 1-1.

NOTE: All 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 1-1. RTC_A Registers

Offset	Acronym	Register Name	Type	Access	Reset
00h	RTCCTL01	Real-Time Clock Control 0, 1	Read/write	Word	4000h
00h	RTCCTL0	Real-Time Clock Control 0	Read/write	Byte	00h
	or RTCCTL01_L				
01h	RTCCTL1	Real-Time Clock Control 1	Read/write	Byte	40h
	or RTCCTL01_H				
02h	RTCCTL23	Real-Time Clock Control 2, 3	Read/write	Word	0000h
02h	RTCCTL2	Real-Time Clock Control 2	Read/write	Byte	00h
	or RTCCTL23_L				
03h	RTCCTL3	Real-Time Clock Control 3	Read/write	Byte	00h
	or RTCCTL23_H				
08h	RTCPS0CTL	Real-Time Prescale Timer 0 Control	Read/write	Word	0100h
08h	RTCPS0CTLL		Read/write	Byte	00h
	or RTCPS0CTL_L				
09h	RTCPS0CTLH		Read/write	Byte	01h
	or RTCPS0CTL_H				
0Ah	RTCPS1CTL	Real-Time Prescale Timer 1 Control	Read/write	Word	0100h
0Ah	RTCPS1CTLL		Read/write	Byte	00h
	or RTCPS1CTL_L				
0Bh	RTCPS0CTLH		Read/write	Byte	01h
	or RTCPS0CTL_H				
0Ch	RTCPS	Real-Time Prescale Timer 0, 1 Counter	Read/write	Word	undefined
0Ch	RT0PS	Real-Time Prescale Timer 0 Counter	Read/write	Byte	undefined
	or RTCPS_L				
0Dh	RT1PS	Real-Time Prescale Timer 1 Counter	Read/write	Byte	undefined
	or RTCPS_H				
0Eh	RTCIV	Real Time Clock Interrupt Vector	Read	Word	0000h
0Eh	RTCIV_L		Read	Byte	00h
0Fh	RTCIV_H		Read	Byte	00h
10h	RTCTIM0	Real-Time Clock Seconds, Minutes	Read/write	Word	undefined
	or RTCNT12	Real-Time Counter 1, 2			
10h	RTCSEC	Real-Time Clock Seconds	Read/write	Byte	undefined
	RTCNT1	Real-Time Counter 1			
	or RTCTIM0_L				
11h	RTCMIN	Real-Time Clock Minutes	Read/write	Byte	undefined
	RTCNT2	Real-Time Counter 2			
	or RTCTIM0_H				
12h	RTCTIM1	Real-Time Clock Hour, Day of Week	Read/write	Word	undefined
	or RTCNT34	Real-Time Counter 3, 4			
12h	RTCHOUR	Real-Time Clock Hour	Read/write	Byte	undefined

Table 1-1. RTC_A Registers (continued)

Offset	Acronym	Register Name	Type	Access	Reset
	RTCNT3	Real-Time Counter 3			
	or RTCTIM1_L				
13h	RTCDOW	Real-Time Clock Day of Week	Read/write	Byte	undefined
	RTCNT4	Real-Time Counter 4			
	or RTCTIM1_H				
14h	RTCDATE	Real-Time Clock Date	Read/write	Word	undefined
14h	RTCDAY	Real-Time Clock Day of Month	Read/write	Byte	undefined
	or RTCDATE_L				
15h	RTCMON	Real-Time Clock Month	Read/write	Byte	undefined
	or RTCDATE_H				
16h	RTCYEAR	Real-Time Clock Year	Read/write	Word	undefined
16h	RTCYEARL		Read/write	Byte	undefined
	or RTCYEAR_L				
17h	RTCYEARH		Read/write	Byte	undefined
	or RTCYEAR_H				
18h	RTCAMINHR	Real-Time Clock Minutes, Hour Alarm	Read/write	Word	undefined
18h	RTCAMIN	Real-Time Clock Minutes Alarm	Read/write	Byte	undefined
	or RTCAMINHR_L				
19h	RTCAHOUR	Real-Time Clock Hours Alarm	Read/write	Byte	undefined
	or RTCAMINHR_H				
1Ah	RTCADOWDAY	Real-Time Clock Day of Week, Day of Month Alarm	Read/write	Word	undefined
1Ah	RTCADOW	Real-Time Clock Day of Week Alarm	Read/write	Byte	undefined
	or RTCADOWDAY_L				
1Bh	RTCADAY	Real-Time Clock Day of Month Alarm	Read/write	Byte	undefined
	or RTCADOWDAY_H				

1.3.1 RTCCTL0 Register

Real-Time Clock Control 0 Register

Figure 1-2. RTCCTL0 Register

7	6	5	4	3	2	1	0
Reserved	RTCTEVIE	RTCAIE	RTCRDYIE	Reserved	RTCTEVIFG	RTCAIFG	RTCRDYIFG
r0	rw-0	rw-0	rw-0	r0	rw-(0)	rw-(0)	rw-(0)

Table 1-2. RTCCTL0 Register Description

Bit	Field	Type	Reset	Description
7	Reserved	R	0h	Reserved. Always reads as 0.
6	RTCTEVIE	RW	0h	Real-time clock time event interrupt enable 0b = Interrupt not enabled 1b = Interrupt enabled
5	RTCAIE	RW	0h	Real-time clock alarm interrupt enable. This bit remains cleared when in counter mode (RTCMODE = 0). 0b = Interrupt not enabled 1b = Interrupt enabled
4	RTCRDYIE	RW	0h	Real-time clock read ready interrupt enable 0b = Interrupt not enabled 1b = Interrupt enabled
3	Reserved	R	0h	Reserved. Always reads as 0.
2	RTCTEVIFG	RW	0h	Real-time clock time event flag 0b = No time event occurred. 1b = Time event occurred.
1	RTCAIFG	RW	0h	Real-time clock alarm flag. This bit remains cleared when in counter mode (RTCMODE = 0). 0b = No time event occurred. 1b = Time event occurred.
0	RTCRDYIFG	RW	0h	Real-time clock read ready flag 0b = RTC cannot be read safely. 1b = RTC can be read safely.

1.3.2 RTCCTL1 Register

Real-Time Clock Control Register 1

Figure 1-3. RTCCTL1 Register

7	6	5	4	3	2	1	0
RTCBCD	RTCHOLD	RTCMODE	RTCRDY	RTCSSEL		RTCTEV	
rw-(0)	rw-(1)	rw-(0)	r-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-3. 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. Changing this bit clears seconds, minutes, hours, day of week, and year to 0 and sets day of month and month to 1. The real-time clock registers must be set by software afterwards. 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	0h	Real-time clock mode 0b = 32-bit counter mode 1b = Calendar mode. Switching between counter and calendar mode resets the real-time clock counter registers. Switching to calendar mode clears seconds, minutes, hours, day of week, and year to 0 and sets day of month and month to 1. The real-time clock registers must be set by software afterwards. RT0PS and RT1PS are also cleared.
4	RTCRDY	RW	0h	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 signal remains cleared.
3-2	RTCSSEL	RW	0h	Real-time clock source select. Selects clock input source to the RTC/32-bit counter. In calendar mode, these bits are don't care. The clock input is automatically set to the output of RT1PS. 00b = ACLK 01b = SMCLK 10b = Output from RT1PS 11b = Output from RT1PS
1-0	RTCTEV	RW	0h	Real-time clock time event Counter mode (RTCMODE = 0) 00b = 8-bit overflow 01b = 16-bit overflow 10b = 24-bit overflow 11b = 32-bit overflow Calendar mode (RTCMODE = 1) 00b = Minute changed 01b = Hour changed 10b = Every day at midnight (00:00) 11b = Every day at noon (12:00)

1.3.3 RTCCTL2 Register

Real-Time Clock Control 2 Register

Figure 1-4. RTCCTL2 Register

7	6	5	4	3	2	1	0
RTCCALS	Reserved	RTCCAL					
rw-(0)	r0	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-4. RTCCTL2 Register Description

Bit	Field	Type	Reset	Description
7	RTCCALS	RW	0h	Real-time clock calibration sign 0b = Frequency adjusted down 1b = Frequency adjusted up
6	Reserved	R	0h	Reserved. Always reads as 0.
5-0	RTCCAL	RW	0h	Real-time clock calibration. Each LSB represents approximately +4ppm (RTCCALS = 1) or a –2ppm (RTCCALS = 0) adjustment in frequency.

1.3.4 RTCCTL3 Register

Real-Time Clock Control 3 Register

Figure 1-5. RTCCTL3 Register

7	6	5	4	3	2	1	0
Reserved						RTCCALF	
r0	r0	r0	r0	r0	r0	rw-(0)	rw-(0)

Table 1-5. RTCCTL3 Register Description

Bit	Field	Type	Reset	Description
7-2	Reserved	R	0h	Reserved. Always reads as 0.
1-0	RTCCALF	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

1.3.5 RTCNT1 Register

Real-Time Clock Counter 1 Register – Counter Mode

Figure 1-6. RTCNT1 Register

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

Table 1-6. RTCNT1 Register Description

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

1.3.6 RTCNT2 Register

Real-Time Clock Counter 2 Register – Counter Mode

Figure 1-7. RTCNT2 Register

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

Table 1-7. RTCNT2 Register Description

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

1.3.7 RTCNT3 Register

Real-Time Clock Counter 3 Register – Counter Mode

Figure 1-8. RTCNT3 Register

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

Table 1-8. RTCNT3 Register Description

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

1.3.8 RTCNT4 Register

Real-Time Clock Counter 4 Register – Counter Mode

Figure 1-9. RTCNT4 Register

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

Table 1-9. RTCNT4 Register Description

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

1.3.9 RTCSEC Register – Calendar Mode With Hexadecimal Format

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

Figure 1-10. RTCSEC Register

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

Table 1-10. RTCSEC Register Description

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

1.3.10 RTCSEC Register – Calendar Mode With BCD Format

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

Figure 1-11. 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 1-11. 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)

1.3.11 RTCMIN Register – Calendar Mode With Hexadecimal Format

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

Figure 1-12. RTCMIN Register

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

Table 1-12. RTCMIN Register Description

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

1.3.12 RTCMIN Register – Calendar Mode With BCD Format

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

Figure 1-13. 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 1-13. 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)

1.3.13 RTCHOUR Register – Calendar Mode With Hexadecimal Format

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

Figure 1-14. RTCHOUR Register

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

Table 1-14. RTCHOUR Register Description

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

1.3.14 RTCHOUR Register – Calendar Mode With BCD Format

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

Figure 1-15. 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 1-15. 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)

1.3.15 RTCDOW Register – Calendar Mode

Real-Time Clock Day of Week Register – Calendar Mode

Figure 1-16. 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 1-16. 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)

1.3.16 RTCDAY Register – Calendar Mode With Hexadecimal Format

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

Figure 1-17. 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 1-17. 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)

1.3.17 RTCDAY Register – Calendar Mode With BCD Format

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

Figure 1-18. 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 1-18. 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)

1.3.18 RTCMON Register – Calendar Mode With Hexadecimal Format

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

Figure 1-19. 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 1-19. RTCMON Register Description

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

1.3.19 RTCMON Register – Calendar Mode With BCD Format

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

Figure 1-20. 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 1-20. 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)

1.3.20 RTCYEARL Register – Calendar Mode With Hexadecimal Format

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

Figure 1-21. RTCYEARL Register

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

Table 1-21. RTCYEARL Register Description

Bit	Field	Type	Reset	Description
7-0	Year	RW	undefined	Year – low byte of 0 to 4095

1.3.21 RTCYEARL Register – Calendar Mode With BCD Format

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

Figure 1-22. RTCYEARL Register

7	6	5	4	3	2	1	0
Decade				Year – lowest digit			
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-22. RTCYEARL Register Description

Bit	Field	Type	Reset	Description
7-4	Decade	RW	undefined	Decade (0 to 9)
3-0	Year – lowest digit	RW	undefined	Year – lowest digit (0 to 9)

1.3.22 RTCYEARH Register – Calendar Mode With Hexadecimal Format

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

Figure 1-23. RTCYEARH Register

7	6	5	4	3	2	1	0
0				Year – high byte of 0 to 4095			
r-0	r-0	r-0	r-0	rw	rw	rw	rw

Table 1-23. RTCYEARH Register Description

Bit	Field	Type	Reset	Description
7-4	0	R	0h	Always 0
3-0	Year	RW	undefined	Year – high byte of 0 to 4095

1.3.23 RTCYEARH Register – Calendar Mode With BCD Format

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

Figure 1-24. RTCYEARH Register

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

Table 1-24. RTCYEARH Register Description

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

1.3.24 RTCAMIN Register – Calendar Mode With Hexadecimal Format

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

Figure 1-25. RTCAMIN Register

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

Table 1-25. 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)

1.3.25 RTCAMIN Register – Calendar Mode With BCD Format

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

Figure 1-26. 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 1-26. 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)

1.3.26 RTCAHOUR Register – Calendar Mode With Hexadecimal Format

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

Figure 1-27. RTCAHOUR Register

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

Table 1-27. 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)

1.3.27 RTCAHOUR Register – Calendar Mode With BCD Format

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

Figure 1-28. 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 1-28. 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)

1.3.28 RTCADOW Register

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

Figure 1-29. 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 1-29. 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)

1.3.29 RTCADAY Register – Calendar Mode With Hexadecimal Format

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

Figure 1-30. 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 1-30. 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)

1.3.30 RTCADAY Register – Calendar Mode With BCD Format

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

Figure 1-31. 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 1-31. 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)

1.3.31 RTCPS0CTL Register

Real-Time Clock Prescale Timer 0 Control Register

Figure 1-32. RTCPS0CTL Register

15	14	13	12	11	10	9	8
Reserved	RT0SSEL	RT0PSDIV			Reserved		RT0PSHOLD
rw-0	rw-0	rw-0	rw-0	rw-0	r0	r0	rw-1
7	6	5	4	3	2	1	0
Reserved			RT0IP			RT0PSIE	RT0PSIFG
r0	r0	r0	rw-0	rw-0	rw-0	rw-0	rw-(0)

Table 1-32. RTCPS0CTL Register Description

Bit	Field	Type	Reset	Description
15	Reserved	R	0h	Reserved. Always reads as 0.
14	RT0SSEL	RW	0h	Prescale timer 0 clock source select. Selects clock input source to the RT0PS counter. In real-time clock calendar mode, these bits are do not care. RT0PS clock input is automatically set to the output of RT0PS. 0b = ACLK 1b = SMCLK
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. 00b = Divide by 2 01b = Divide by 4 10b = Divide by 8 11b = Divide by 16 00b = Divide by 32 01b = Divide by 64 10b = Divide by 128 11b = 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 via the RTCHOLD bit. 0b = RT0PS operational 1b = RT0PS held
7-5	Reserved	R	0h	Reserved. Always reads as 0.
4-2	RT0IP	RW	0h	Prescale timer 0 interrupt interval 00b = Divide by 2 01b = Divide by 4 10b = Divide by 8 11b = Divide by 16 00b = Divide by 32 01b = Divide by 64 10b = Divide by 128 11b = 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

1.3.32 RTCPS1CTL Register

Real-Time Clock Prescale Timer 1 Control Register

Figure 1-33. RTCPS1CTL Register

15	14	13	12	11	10	9	8
RT1SSEL		RT1PSDIV			Reserved		RT1PSHOLD
rw-0	rw-0	rw-0	rw-0	rw-0	r0	r0	rw-1
7	6	5	4	3	2	1	0
Reserved			RT1IP			RT1PSIE	RT1PSIFG
r0	r0	r0	rw-0	rw-0	rw-0	rw-0	rw-(0)

Table 1-33. RTCPS1CTL Register Description

Bit	Field	Type	Reset	Description
15-14	RT1SSEL	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 = ACLK 01b = SMCLK 10b = Output from RT0PS 11b = Output from RT0PS
13-11	RT1PSDIV	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. 00b = Divide by 2 01b = Divide by 4 10b = Divide by 8 11b = Divide by 16 00b = Divide by 32 01b = Divide by 64 10b = Divide by 128 11b = 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 via the RTCHOLD bit. 0b = RT1PS operational 1b = RT1PS held
7-5	Reserved	R	0h	Reserved. Always reads as 0.
4-2	RT1IP	RW	0h	Prescale timer 1 interrupt interval 00b = Divide by 2 01b = Divide by 4 10b = Divide by 8 11b = Divide by 16 00b = Divide by 32 01b = Divide by 64 10b = Divide by 128 11b = Divide by 256
1	RT1PSIE	RW	0h	Prescale timer 1 interrupt enable 0b = Interrupt not enabled 1b = Interrupt enabled
0	RT1PSIFG	RW	0h	Prescale timer 1 interrupt flag 0b = No time event occurred 1b = Time event occurred

1.3.33 RT0PS Register

Real-Time Clock Prescale Timer 0 Counter Register

Figure 1-34. RT0PS Register

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

Table 1-34. RT0PS Register Description

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

1.3.34 RT1PS Register

Real-Time Clock Prescale Timer 1 Counter Register

Figure 1-35. RTPS1 Register

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

Table 1-35. RT1PS Register Description

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

1.3.35 RTCIV Register

Real-Time Clock Interrupt Vector Register

Figure 1-36. RTCIV Register

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

Table 1-36. RTCIV Register Description

Bit	Field	Type	Reset	Description
15-0	RTCIV	R	0h	Real-time clock interrupt vector value 00h = No interrupt pending 02h = Interrupt Source: RTC ready; Interrupt Flag: RTCRDYIFG 04h = Interrupt Source: RTC interval timer; Interrupt Flag: RTCTEVIFG 06h = Interrupt Source: RTC user alarm; Interrupt Flag: RTCAIFG 08h = Interrupt Source: RTC prescaler 0; Interrupt Flag: RT0PSIFG 0Ah = Interrupt Source: RTC prescaler 1; Interrupt Flag: RT1PSIFG 0Ch = Reserved 0Eh = Reserved 10h = Reserved ; Interrupt Priority: Lowest

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