Real-Time Clock (RTC)

Introduction

The Real-Time Clock (RTC) peripheral is a sophisticated timer that keeps track of Calendar, Month, and Time information. It operates in Binary or BCD modes; whichever is most useful for your application.

The RTC affords you the ability to set Calendar/Time based Alarms (i.e. Interrupts).

This peripheral is extremely power sensitive and operates in many low-power modes. In fact, on the MSP430FR5969, it even operates in LPM3.5 mode.

Learning Objectives

- Describe the architecture of the Real-Time Clock module.
- Learn to set alarms/interrupts for the RTC.

Chapter Topics

What is a Real-Time Clock?

What is a Real-Time Clock (RTC)?

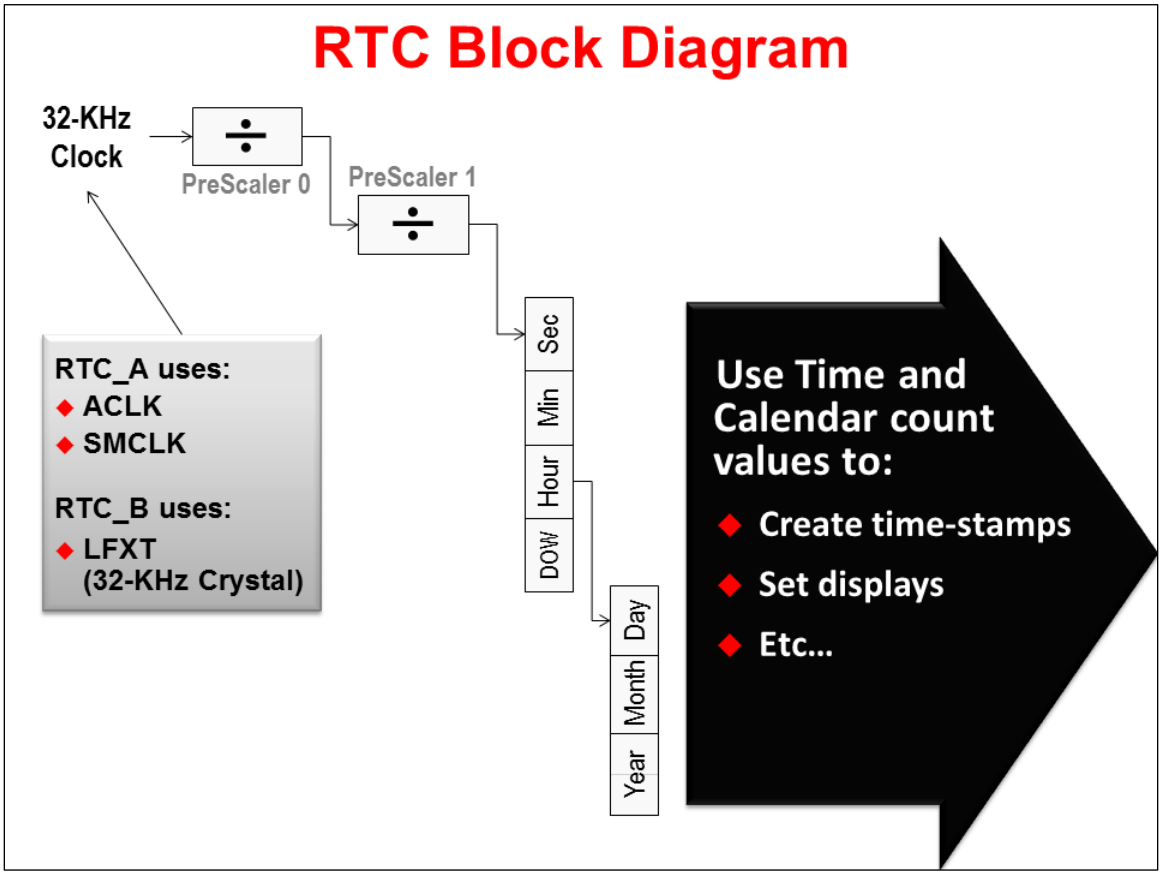
It's just that...

an alarm Clock

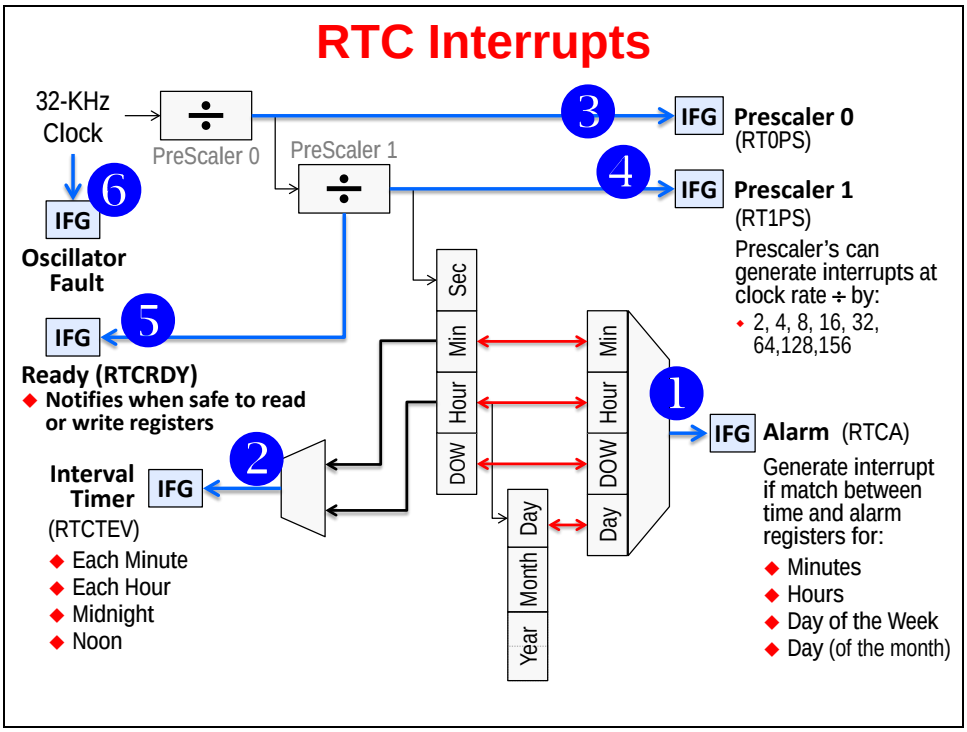
with Calendar functions

How Does the RTC Work?

RTC Block Diagram

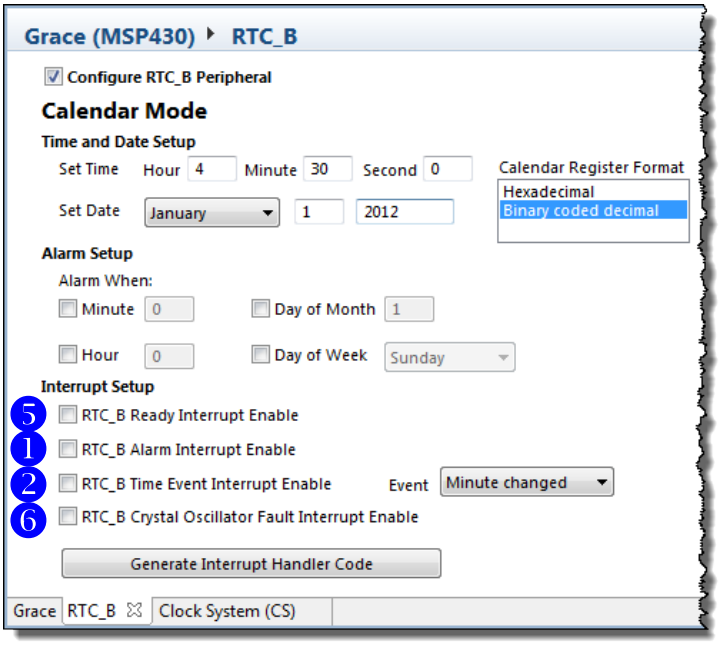


RTC Interrupts



Programming the RTC

Setting RTC using GRACE



Grace supports:

- Devices
  - F2xx, G2xx
  - FR5xx
- RTC use cases
  - Calendar mode only
- BCD or Hex modes
- Creates interrupt handler template:
  - Alarm
  - Events
  - Ready
  - Osc Fault

DriverLib Example Code

```
// Initialize Calendar Mode of RTC
RTC_B_calendarInit ( RTC_B_BASE, currentTime, RTC_B_FORMAT_BINARY);

// Setup Calendar Alarm for 5:00pm on the 5th day of the week.
// Note: Does not specify day of the week.
RTC_B_setCalendarAlarm ( RTC_B_BASE,0x00,0x17, RTC_B_ALARMCONDITION_OFF,0x05);

// Specify an interrupt  to assert every minute
RTC_B_setCalendarEvent ( RTC_B_BASE, RTC_B_CALENDAREVENT_MINUTECHANGE);

// Clear interrupt bits before starting RTC
RTC_B_clearInterrupt ( RTC_B_BASE, RTC_B_CLOCK_READ_READY_INTERRUPT + RTC_B

// Enable interrupt for RTC Ready Status, that let's us know RTC registers are ready to read.
// Also, enable interrupts for the Calendar alarm & event.
RTC_B_enableInterrupt ( RTC_B_BASE, RTC_B_CLOCK_READ_READY_INTERRUPT + RTC_

// Start RTC Clock
RTC_B_startClock ( RTC_B_BASE);

// Enter LPM3 mode with interrupts enabled
__low_power_mode_3 ();
__no_operation();
```

Additional Considerations

Additional Features

- **Using RTC in LPM3.5 Mode**
  - **All RTC's work in LPM3 mode**
  - **Since RTC_B and RTC_C can directly access the LF crystal, they can operate in the "3.5" low-power mode**
  - **LPM3.5 provides the lowest possible power dissipation with RTC wake-up capability**
- **Easy Conversion Between BCD and Hex**
  - **RTC_B/RTC_C provide BCD conversions in hardware**
  - **Driver Library function provides easy access to this hardware feature**
- **Counter Mode**
  - **RTC_A can be used as a 32-bit counter (rather than Calendar mode)**
  - **Counter mode generates overflow interrupts at 8-, 16-, 24- and 32-bits**

Exercise Caution

- **Clear bit-fields before setting counters and alarms**
  - Prior to setting an alarms, clear all alarm registers, including the alarm enable (AE) bits
  - To prevent potential erroneous alarms when setting time values, clear the interrupt enable (IE) bits, as well as the AE bits
  - Writes to count registers takes effect immediately. Note that the RTC clock is stopped during the write and both pre-scale registers are reset. This could result in losing up to 1 second during a write.
- **Invalid time and alarm settings are not validated or handled via hardware (measure twice, program once)**
- **Reading Registers**
  - Care should be taken when reading (or writing) RTC time/calendar/prescale registers so that your actions do not occur during counter transitions
  - These options can help to prevent erroneous results:
    1. Let the RTC Ready (RTCRDY) interrupt you just after an update – you'll have ~1 sec before the next update
    2. Check the RTCRDY bit before reading or writing the registers
    3. Read the registers multiple times and take the majority vote
    4. Hold the RTC before reading or writing any registers

Summary

RTC Comparison

	Feature	RTC_A	RTC_B	RCT_C
	Highlights	32-bit Counter Mode	LPM3.5, Calendar Mode Only	Protection Plus Improved Calibration & Compensation
Modes	**Calendar Mode with Programmable Alarms**	Yes	Yes	Yes
	Counter Mode	Yes	No	Device-dependent
	Input Clocks	ALCK, SMCLK	32-kHz crystal	32-kHz crystal
	LPM3.5 Support	No	Yes	Yes
Compensation & Calibration	**Offset Calibration Register**	Yes	Yes	Yes
	Temperature Compensation Register	No	No	Yes
	Temperature Compensation	With software, manipulating offset calibration value		With software using separate temperature compensation register
	Calibration and Compensation Period	64 min	60 min	1 min
Features	**BCD to Binary Conversion**	Integrated for Calendar Mode	Integrated for Calendar Mode plus separate conversion registers	
	Event/Tamper Detect With Time Stamp	No	No	Device-dependent
	Password Protected Calendar Registers	No	No	Yes