

RX-8010SJ

低功耗实时时钟模块

RX-8010SJ

内置频率校准的 32.768k 晶体单元

实时时钟功能 : 计时/日历功能, 定时中断等功能

用户 RAM : 内置 128 bit RAM

32.768k 频率输出功能 : C-MOS 或 N-ch 开漏输出

接口类型 : I2C 总线

接口电压范围 : 1.6 V ~ 5.5V

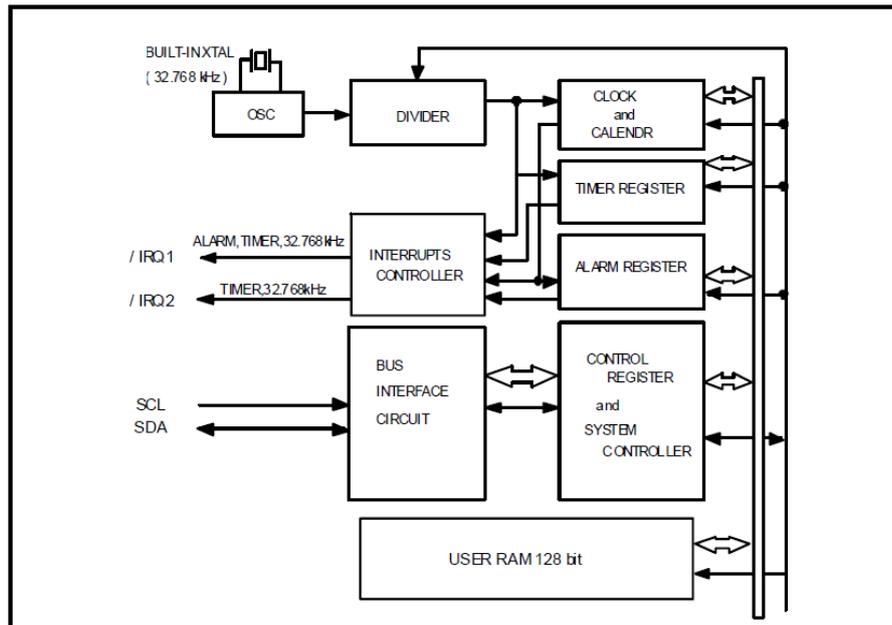
时钟数据保持电压 : 1.1V ~ 5.5V

电流功耗 : 160nA / 3V Typ.

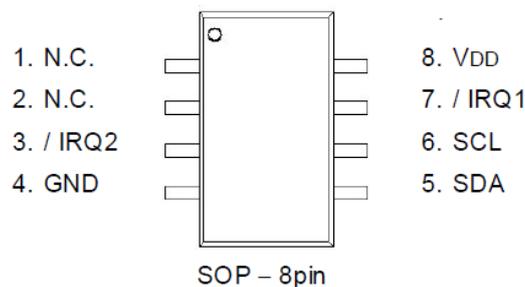
1. 概述

这是一款内置 32.768k 晶体单元的实时时钟模块。除了具有计时、日历的基本功能以外, 还具有定时中断、固定周期中断、频率缺失检测、频率输出、用户 RAM 等功能。8 Pin 的 SOP 封装使之适合于多种小型化电子设备的应用。

2. 结构框图



3. 引脚定义



RX-8010SJ

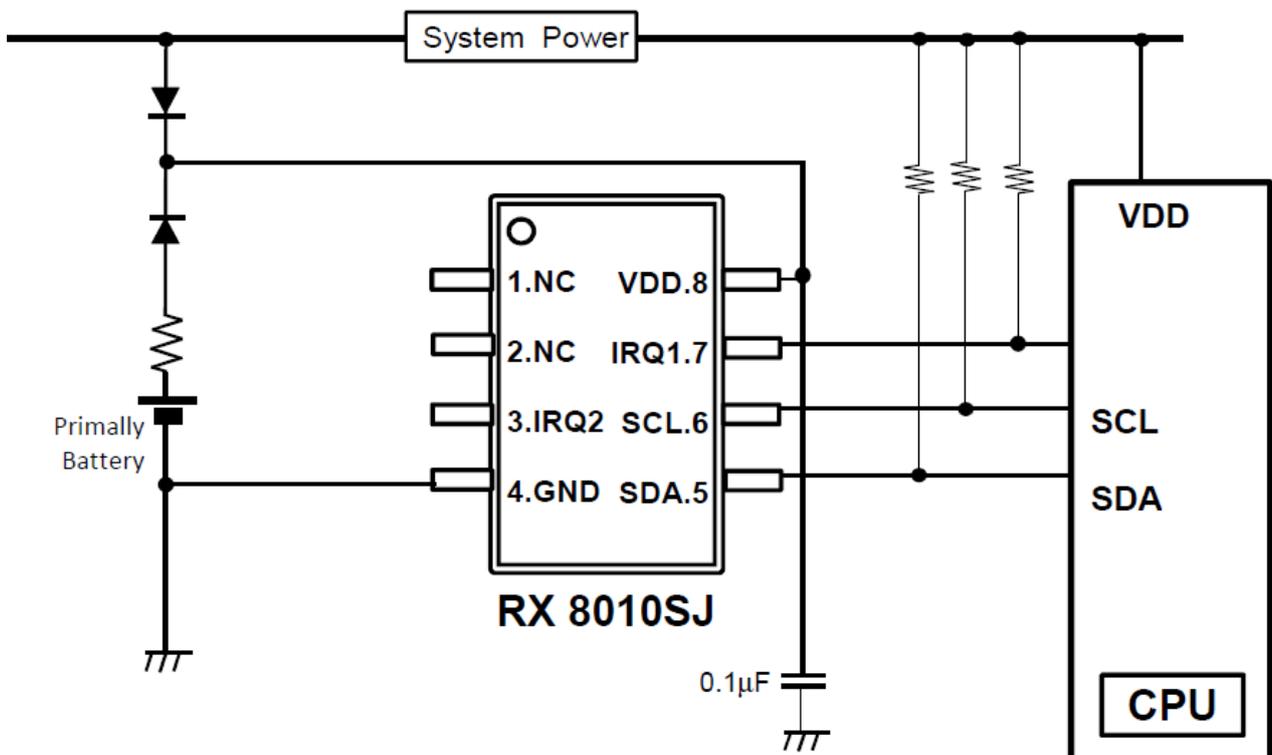
4. 推荐操作条件

*Unless otherwise specified, GND = 0 V, Ta = -40 °C to +85 °C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Operating supply voltage	V _{ACC}	-	1.6	3.0	5.5	V
Clock supply voltage	V _{CLK}	-	1.1	3.0	5.5	V
Backup power supply Low voltage detection	V _{LOW}	-			1.10	V
Applied voltage when OFF	V _{PUP}	SDA, /IRQ1pin			5.5	V
Operating temperature	T _{OPR}	No condensation	-40	+25	+85	°C

*Minimum value of Clock supply voltage V_{CLK} is the timekeeping continuation lower limit value that initialized RX8010 in operating supply voltage V_{ACC}.

5. 硬件设计



RX8010SJ 具有两个中断输出引脚，可以通过寄存器来选择，设计可以根据实际情况选择。NC 引脚请不要有任何器件连接。

RX-8010SJ

6. 软件设计

系统在上电复位后需要对 RTC 进行软件初始化，以保证 RX8010SJ 的正常操作。

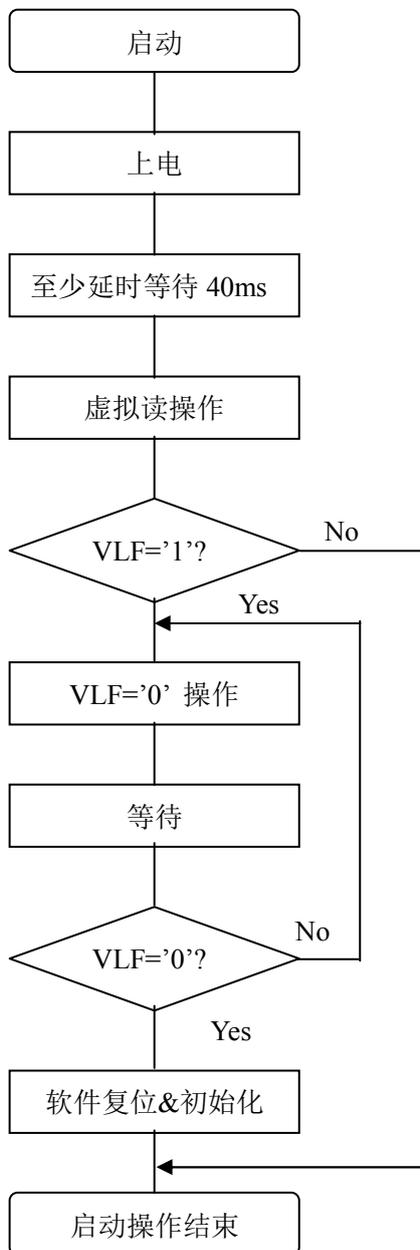
首先，要保证系统复位的可靠性，在复位以后系统需要等待一个稳定时间，使 RX8010SJ 的内部晶体稳定以后再对 RX8010SJ 进行读写操作。这个时间一般在 200ms 左右。

6.1 初始化设置

在进行初始化的时候，一定注意不能将非法的数据或时间写入相应的寄存器，如果设置了错误的的数据，那么不能保证计时的正确性。

1) 初始化的例子

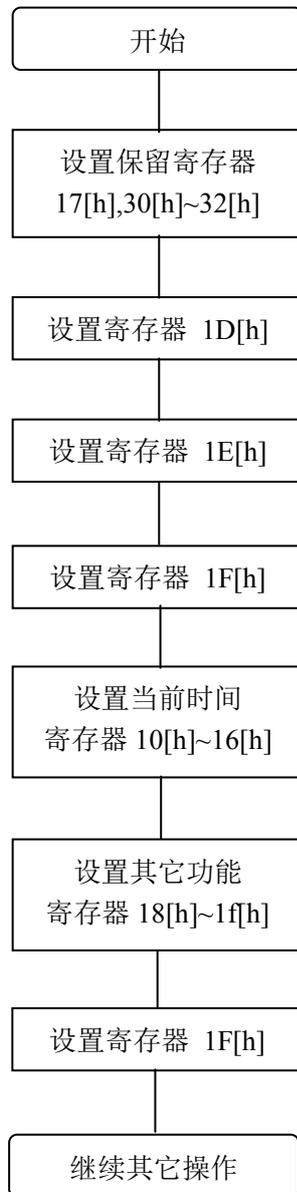
例 1，初次上电



- 当上电复位时不能确保电源稳定可靠，那么执行一条虚拟 读操作 判断总线是否可用
- 判断 VLF (低电压检测位) 的状态
- 将 VLF 位清零
- 请根据系统的实际情况设置等待时间
- 判断 VLF (低电压检测位) 的状态
- 如果有必要：对系统寄存器，时间进行初始化设置

RX-8010SJ

例 2，寄存器初始化



- 0xD8 写入 17[h]寄存器
- 0x00 写入 30[h]寄存器
- 0x08 写入 31[h]寄存器
- 根据实际功能设置 32[h]寄存器

- 根据实际功能设置 1D[h]寄存器

- 根据实际功能设置 1E[h]寄存器
 - 使 VLF= '0'
- 根据实际功能设置 1F[h]寄存器
 - 使 STOP= '1'

- 设置正确的时间数据到对应的寄存器

- 根据实际功能设置 1F[h]寄存器
 - 使 STOP= '0'

RX-8010SJ

例 3，对时间的读写操作

• 时间写：



- STOP 置 1，防止设置时间时计数器干扰寄存器值
- 必要时对时间信息重新设置
- STOP 清 0，时钟功能开始运行

• 时间读：



- 读操作在任何时候都可以进行

RX-8010SJ

7 通过 I2C 总线接口读写数据

7.1 器件地址 (Device Address/Slave Address)

所有的通讯操作都是以 [START 条件] + [从设备地址 + (R/W 读写选择)] 开始的。
从设备地址如下：

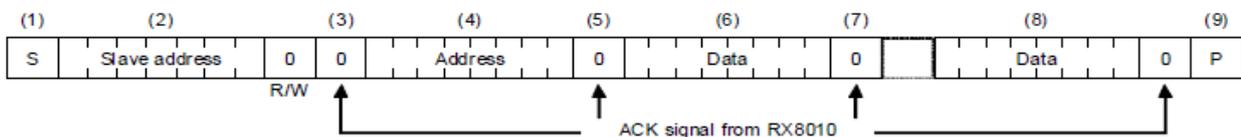
	Transfer data	Slave address							R/W bit
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Read	65 h	0	1	1	0	0	1	0	1 (= Read)
Write	64 h	0	1	1	0	0	1	0	0 (= Write)

7.2 I2C 总线协议

7.2.1 寄存器写操作时序图

指定地址写操作：

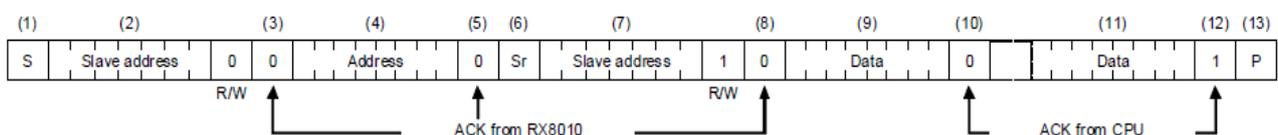
- 1) CPU 发送开始条件[S]
- 2) CPU 传输 RX-8010SJ 的从地址，用 R/W 位设定写模式。
- 3) 检测从 RX-8010SJ 发出的 ACK 信号。
- 4) CPU 传输写入寄存器的地址到 8010SJ
- 5) 检测从 RX-8010SJ 发出的 ACK 信号
- 6) CPU 将要写入的数据写道指定的寄存器
- 7) 检测从 RX-8010SJ 发出的 ACK 信号
- 8) 如果有需要可重复 (6) 和 (7) 步骤，写入的地址自动增加
- 9) CPU 发送停止位[P]



7.2.2 寄存器读操作时序图

指定地址读操作：

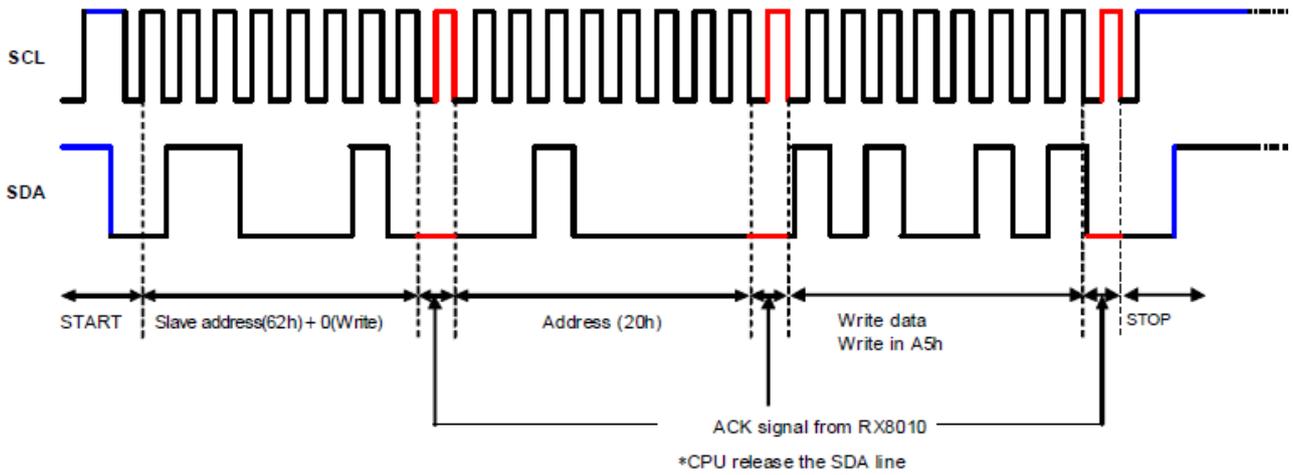
- 1) CPU 发送开始条件[S]
- 2) CPU 传输 RX-8010SJ 的从地址，用 R/W 位设定写模式
- 3) 检测从 RX-8010SJ 发出的 ACK 信号
- 4) CPU 传输读寄存器的地址到 8010SJ
- 5) 检测从 RX-8010SJ 发出的 ACK 信号
- 6) CPU 发送 RESTART 条件[Sr]
- 7) CPU 传输 RX-8010SJ 的从地址，用 R/W 位设定读模式
- 8) 检测从 RX-8010SJ 发出的 ACK 信号
- 9) 从 8010SJ 中读取步骤 (4) 指定的寄存器内容
- 10) CPU 发送 ACK 信号给 8010SJ
- 11) 如果有需要可重复 (9) 和 (10) 步骤，读取的地址自动增加
- 12) CPU 发送一个 '1' 作为 ACK 信号
- 13) CPU 发送停止信号[P]



RX-8010SJ

7.3 通讯波形示例

1) 指定地址写时序，将 A5[h] 写入地址为 20[h] 寄存器



2) 指定地址读时序，从地址为 20[h]的寄存器读取数据

