内置晶振、IIC 串行接口的高精度实时时钟模块-SD2403API

1.概述

SD2403API 是一种内置晶振、具有标准 IIC 接口的实时时钟模块,CPU 可使用该接口通过 5 位地址寻址来读写片内 32 字节寄存器的数据(包括时间寄存器、报警寄存器、控制寄存器、通用 SRAM 寄存器)。

SD2403API 内置晶振, 该模块可保证时钟精度为±5ppm(在 25℃±1℃下), 即年误差小于 2.6 分钟。

SD2403API 内置单路定时/报警中断输出,报警中断时间最长可设至 100 年。

SD2403API 内置时钟精度数字调整功能，可以在很宽的范围内校正时钟的偏差(-189ppm～+189ppm,分辨力为 3.05ppm)，并通过外置的温度传感器可设定适应温度变化的调整值，实现在宽温范围内高精度的计时功能。

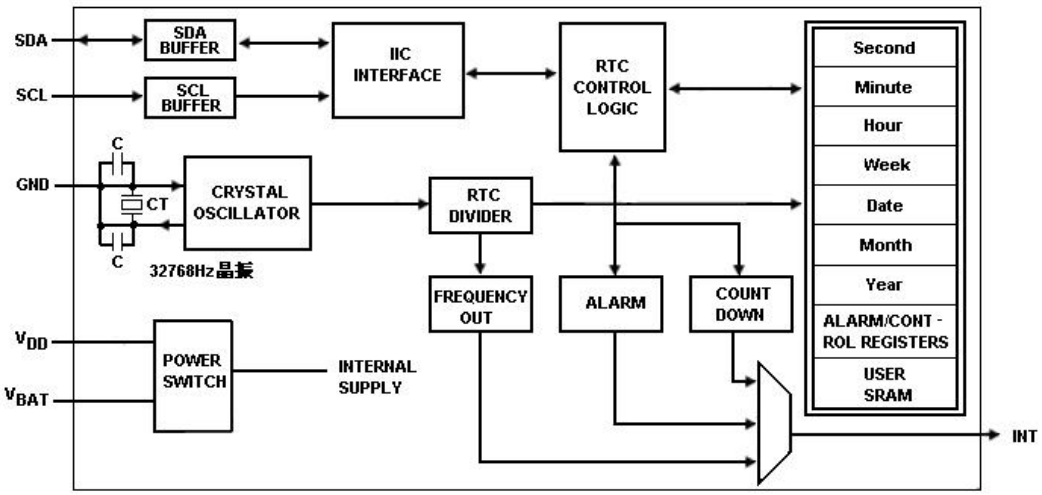
SD2403API 具有一个后备电池输入脚 VBAT,当模块检测到主电源 VDD 掉到后备电池电压以下,模块会自动转为由接在 VBAT 的后备电池供电。

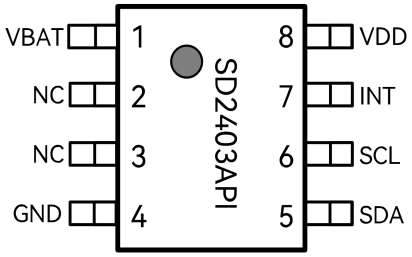
2.特性

- 低功耗： 1.0μA 典型值(VBAT =3.0V，Ta=25℃)。
- 工作电压：1.8～5.5V，工作温度：-40℃～+85℃。
- 标准IIC总线接口方式,最高速度400KHz(4.5V～5.5V)
- 年、月、日、星期、时、分、秒的BCD码输入/输出,并可通过独立的地址访问各时间寄存器。
- 闰年自动调整功能(从2000年～2099年)
- 可选择12/24小时制式
- 内置年、月、日、星期、时、分、秒共7字节的报警数据寄存器及1字节的报警允许寄存器。
- 内置12字节通用SRAM寄存器可用于存储用户的一般数据
- 三种中断均可选择从INT脚输出,并具有两个中断标志位。
- 可设定并自动重置的单路报警中断功能（时间范围最长设至100年）,年、月、日、星期、时、分、秒报警共有96种组合方式，并有单事件报警和周期性报警两种中断输出模式.
- 周期性频率中断输出:从32768Hz～1/16Hz……1秒共十五种方波脉冲。
- 自动重置的8位倒计时定时器,可选的4种时钟源（4096Hz、64Hz、1Hz、1/60Hz）。
- 内置时钟精度数字调整功能，可通过程序来调整走时的快慢。用户采用外置的温度传感器，设定适应温度变化的调整值，可实现在宽温范围内高精度的计时功能。
- 具有一个后备电池输入脚VBAT ，模块依据不同的电压自动从VDD切换到VBAT或从VBAT切换到VDD。

- 在VBAT模式下，模块具有中断输出允许或禁止的功能,可满足在备用电池供电时输出中断的需要。
- 内置IIC总线0.5秒自动复位功能(从Start命令开始计时),保证时钟数据的有效性及可靠性,避免IIC总线挂死问题。
- 内置三个时钟数据写保护位，避免对数据的误写操作，可更好地保护时钟数据。
- 内置VBAT模式IIC总线通信禁止功能,从而避免在电池供电时CPU对时钟操作所消耗的电池电量，也可避免在主电源上、下电的过程中因CPU的I/O端口所输出的不受控的杂波信号对时钟模块的误写操作，进一步提高时钟模块的可靠性。
- 内置上电复位电路及指示位。
- 内置电源稳压,内部计时电压可低至1.5V。
- 模块管脚抗静电(ESD)>4KV。
- 模块在兴威帆的评估板上可通过4KV的群脉冲(EFT)干扰
- 内置晶振，出厂前已对时钟进行校准，时钟精度为±5ppm(在25℃±1℃下)，即年误差小于2.6 分钟。
- 封装形式:8脚的DIP封装，SD2403API为工业级型号。

3. 管脚定义和原理框图





封装形式:DIP-8

表 1　SD2403API 管脚功能表

脚号	名称	功能	特征
1	V_{BAT}	备用电源(电池)输入脚. 当 V_{DD} 电源失效时 V_{BAT} 为模块提供电源。(V_{BAT} 与电池之间的接法请参见本资料中的“**应用参考电路**”章节)	1.5V～5.5V,不用时应将其接 GND。
2、3	NC	没有与模块内部连接	悬空或接地
4	GND	负电源（GND）	
5	SDA	串行数据输入/输出脚，此管脚通常用一电阻上拉至 V_{DD}，并与其它漏极开路或集电器开路输出的器件通过线与方式连接。	N 沟道开路输出，CMOS 输入;当 V_{BAT} 引脚的后备电源被激活时,该引脚被禁止。
6	SCL	串行时钟输入脚，由于在 SCL 上升/下降沿处理信号，要特别注意 SCL 信号的上升/下降升降时间，应严格遵守说明书。	CMOS 输入,当 V_{BAT} 引脚的后备电源被激活时,该引脚被禁止。
7	INT	报警中断输出脚，根据控制寄存器来设置其工作的模式，它可通过重写控制寄存器来禁止。	N-沟道开路输出
8	V_{DD}	正电源	1.8V～5.5V

4. 基本功能定义

4.1 寄存器列表

地址	寄存器段	寄存器名称	BIT								数值范围	缺省值
			D7	D6	D5	D4	D3	D2	D1	D0	(十进制)	(二进制)
00H	实时时钟寄存器	秒	0	S40	S20	S10	S8	S4	S2	S1	0-59	xxxx-xxxx
01H		分钟	0	MN40	MN20	MN10	MN8	MN4	MN2	MN1	0-59	xxxx-xxxx
02H		小时	12_/24	0	H20P/A_	H10	H8	H4	H2	H1	0-23	xxxx-xxxx
03H		星期	0	0	0	0	0	W4	W2	W1	0-6	xxxx-xxxx
04H		日	0	0	D20	D10	D8	D4	D2	D1	1-31	xxxx-xxxx
05H		月	0	0	0	MO10	MO8	MO4	MO2	MO1	1-12	xxxx-xxxx
06H		年	Y80	Y40	Y20	Y10	Y8	Y4	Y2	Y1	0-99	xxxx-xxxx
07H	时间报警寄存器	秒报警	0	AS40	AS20	AS10	AS8	AS4	AS2	AS1	0-59	0000-0000
08H		分钟报警	0	AMN40	AMN20	AMN10	AMN8	AMN4	AMN2	AMN1	0-59	0000-0000
09H		小时报警	0	0	AH20 API/A_	AH10	AH8	AH4	AH2	AH1	0-23	0000-0000
0AH		星期报警	0	AW6	AW5	AW4	AW3	AW2	AW1	AW0	N/A	0000-0000
0BH		日报警	0	0	AD20	AD10	AD8	AD4	AD2	AD1	1-31	0000-0000
0CH		月报警	0	0	0	AMO10	AMO8	AMO4	AMO2	AMO1	1-12	0000-0000
0DH		年报警	AY7	AY6	AY5	AY4	AY3	AY2	AY1	AY0	0-99	0000-0000
0EH		报警允许	0	EAY	EAMO	EAD	EAW	EAH	EAMN	EAS	N/A	0000-0000
0FH	控制寄存器	CTR1	WRTC3	0	INTAF	INTDF	0	WRTC2	0	RTCF	N/A	0000-0000
10H		CTR2	WRTC1	IM	INTS1	INTS0	FOBAT	INTDE	INTAE	INTFE	N/A	0000-0000
11H		CTR3	ARST	0	TDS1	TDS0	FS3	FS2	FS1	FS0	N/A	0000-0000
12H		时间调整	0	F6	F5	F4	F3	F2	F1	F0	N/A	0000-0000
13H		倒计时定时器	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	0-255	0000-0000
14~1FH	通用 RAM	(12Bytes)	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	N/A	xxxx-xxxx

注：“X”表示随机值，可以为 0 或者 1。

4.2 实时时钟数据寄存器(00H~06H)

实时时钟数据寄存器是 7 字节的存储器，它以 BCD 码方式存贮包括年、月、日、星期、时、分、秒的数据。

年数据[06H 地址]（00~99）：设置千年（20XX）的后两位数字（00~99），通过自动日历功能计至 2099 年。（注意：2000 年为闰年）

月数据[05H 地址（01~12）：每月包含的天数通过自动日历功能来更改。

1,3,5,7,8,10,12：1~31

4,6,9,11：1~30

2(闰年)：1~29

2(普通)：1~28

日数据[04H 地址]（01~31）

星期数据[03H 地址]（00~06）：七进制计数器,00 对应星期天,01 对应星期一,依次类推。

小时数据[02H 地址]（00~23 或 00~11）：小时的最高位 12_/24 是 12 或 24 小时制选择位。

当 12_/24=1 时,24 小时制；当 12_/24=0 时， 12 小时制。

12 小时制时,H20 为 AM/PM 指示位,H20=0 为 AM,H20=1 为 PM,

见下表:(位 H20H10H8H4H2H1)

24 小时显示系统	12 小时显示系统	24 小时显示系统	12 小时显示系统
00	12（AM12）	12	32（PM12）
01	01（AM1）	13	21（PM1）
02	02（AM2）	14	22（PM2）
03	03（AM3）	15	23（PM3）
04	04（AM4）	16	24（PM4）
05	05（AM5）	17	25（PM5）
06	06（AM6）	18	26（PM6）
07	07（AM7）	19	27（PM7）
08	08（AM8）	20	28（PM8）
09	09（AM9）	21	29（PM9）
10	10（AM10）	22	30（PM10）
11	11（AM11）	23	31（PM11）

注意：当读取小时数据时,要屏蔽掉小时的最高位 12_/24,否则在 24 小时制时会因为 12_/24=1 而显示不对。

分数据[01 地址]（00~59）

秒数据[00H 地址]（00~59）

例如:设时间为 2006 年 12 月 20 日星期三 18 点 19 分 20 秒(24 小时制),则寄存器 00～07H 的赋值应分别为 20h、19H、98h、03h、20h、12h、06h。要特别注意此处小时位的赋值,因为是 24 小时制式,小时的 12_/24 位=1,所以小时的赋值为 98h(1001 1000B)。

注:

1.在上电复位时，模块内部不对实时时钟数据寄存器作清零或置位处理。

2.当写实时时间数据时(00H~06H),不可以单独对 7 个时间数据中的某一位进行写操作,否则可能会引起时间数据的错误进位,所以要修改其中某一个数据，应一次性写入全部 7 个实时时钟数据。

3.当模块收到读实时时钟数据命令,则所有实时时钟数据被锁存(时钟走时并不受影响),此功能可以避免时间数据的错读现象。

4.3 中断(08H~13H 地址)

SD2403API 有 3 种不同的中断，它们由控制寄存器 2(10H)中的位 INTAE、 INTFE、INTDE 位来使能:

中断允许位/标志位列表

序号	中断允许位 (1=允许,0=禁止)	中断名	中断标志位 (1=有中断,0=无中断)
1	INTAE	报警中断	INTAF
2	INTFE	频率中断	无
3	INTDE	倒计时中断	INTDF

当报警中断产生时,置中断标志位 INTAF 为 1;当倒计时中断产生时， 置中断标志位 INTDF 为 1;频率中断没有标志位，标志位被置 1 后,需要手动清除。

三种中断都是以 INT 脚为输出脚， 通过控制寄存器 2 中的 INTS1、INTS0 位来选择确定 INT 脚输出何种中断:

INT 脚中断输出选通表

序号	INTS1	INTS0	描述
0	0	0	禁止输出,高阻态
1	0	1	报警中断输出
2	1	0	频率中断输出
3	1	1	倒计时中断输出

（1）报警中断

当 INTAE=1 时报警中断被允许,报警中断何时发生由时间报警寄存器(07H~0EH)来确定。这其中 07H~0DH 依次用于存放报警时间的秒、分钟、小时、星期、日、月、年数据,除小时报警数据寄存器的最高位始终为”0” 、星期位的定义不同以外，其它的格式与实时时钟寄存器相同。

0EH 为时间报警允许寄存器,如下:

BIT	D7	D6	D5	D4	D3	D2	D1	D0
位名(值)	0	EAY	EAMO	EAD	EAW	EAH	EAMN	EAS
报警允许	-	年 (0DH)	月 (0CH)	日 (0BH)	星期 (0AH)	小时 (09H)	分钟 (08H)	秒 (07H)

注:1=允许,0=禁止。

时间报警允许寄存器的使能位是用于确定哪些时间报警寄存器(秒、分钟等)需要与实时时钟寄存器之间作比较。当实时时钟运行时,一旦被允许的报警寄存器均与对应的实时时钟寄存器相匹配，就会触发一次报警中断,同时报警中断标志位 INTAF 位被置”1”。

特别：

1.当日报警与星期报警均被允许即 EAD=EAW=1 时，只有日报警有效,而星期报警无效，所以时间报警中断共有 96 种组合方式。

2.星期报警寄存器的数据格式与实时时钟数据星期的格式不同,星期报警寄存器的位 AW6、AW5、AW4、AW3、AW2、AW1、AW0 分别对应星期六、星期五、星期四、星期三、星期二、星期一、星期日,并可多位置 1,例如 AW6,AW1=1,其它位为 0,则对应在星期六，星期一会有报警。

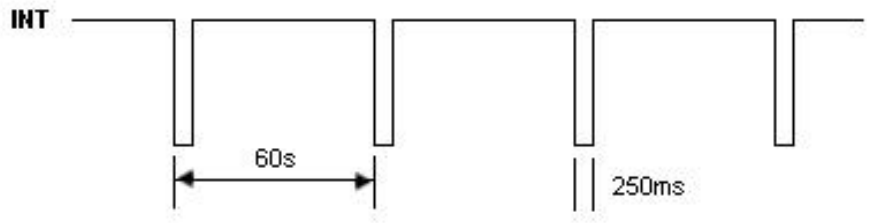
每一次对时间报警允许寄存器的写入都会清 INTAF 为"0"，当设置 INTS1=0、INTS0=1 时,即允许报警中断从 INT 脚输出。该报警中断有两种模式,即单事件报警和周期性报警，模式的选定由控制寄存器 2 中的报警中断模式位 IM 的值来定:

IM 位	报警中断模式	INT 脚
0	单事件报警	输出低电平直至 INTAF 位清零
1	周期性报警	输出低电平有效、宽度为 250ms 的周期性脉冲直至 INTAF 位清零

为清除报警中断,可通过写操作将控制寄存器 1 的 INTAF 位置”0”，但当 ARST 位置为”1”,则在控制寄存器 1 被读取时,INTAF 位会自动清零。

举例:

1.设寄存器 0EH=00000001B,秒报警寄存器 07H=20h,位 INTAE=1、IM=1、INTS1=0、INTS0=1 ,则每当时间秒数据进位到 20h 的时候，INT 脚都会输出宽度为 250ms 的低电平,即频率为 1 分钟的方波。如下图：



2.设寄存器 0EH=00001111B,星期报警寄存器 0AH=0010 0110B,小时报警寄存器 09H=08h,分钟报警寄存器 08H=30h,秒报警寄存器 07H=00h,位 INTAE=1、IM=1、INTS1=0、INTS0=1 ，则每到星期一、星期二、星期五的 8 点 30 分 0 秒的时候，INT 脚都会输出宽度为 250ms 的低电平。

3.设寄存器 0EH=00010111B,日报警寄存器 0BH=01h,小时报警寄存器 09H=08h,分钟报警寄存器 08H=30h,秒报警寄存器 07H=00h,位 INTAE=1、IM=1、INTS1=0、INTS0=1 ，则到每个月 1 号的 8 点 30 分 0 秒的时候，INT 脚都会输出宽度为 250ms 的低电平。

4.设寄存器 0EH=0111 0100B, 年报警寄存器 0DH=08h,月报警寄存器 0CH=08h,日报警寄存器 0BH=08h,小时报警寄存器 09H=20h,位 INTAE=1、IM=0、INTS1=0、INTS0=1、12_/24=1，则到 2008 年 8 月 8 日 20 点 0 分 0 秒时，INT 脚会输出低电平。此后如清零 INTAF，则 INT 脚从低电平变成高电平。

（2）频率中断

当 INTFE=1 时频率中断被允许， INTFE=0 时频率中断被禁止。

当设置 INTS1=1、INTS0=0 时,即允许频率中断从 INT 脚输出，频率中断没有标志位。

INT 脚输出频率中断由控制寄存器 3（CTR3)中的 FS3、FS2、FS1、FS0 位来选择确定:

频率(Hz)	FS3	FS2	FS1	FS0
0	0	0	0	0
32768	0	0	0	1
4096	0	0	1	0
1024	0	0	1	1
64	0	1	0	0
32	0	1	0	1
16	0	1	1	0
8	0	1	1	1
4	1	0	0	0
2	1	0	0	1
1	1	0	1	0
1/2	1	0	1	1
1/4	1	1	0	0
1/8	1	1	0	1
1/16	1	1	1	0
1 秒	1	1	1	1

注:INT 脚输出的频率中断除了秒均是由 32768Hz 晶体振荡电路整形及分频得到的;1 秒频率中断是指输出 500ms 低电平、500ms 高电平的方波,输出低电平的下降沿与秒进位同步;在数字调整寄存器起作用时,1 秒和 1Hz 应是不相同的。

（3）倒计时中断

与倒计时中断相关的寄存器是 8 位的倒数定时器，当控制寄存器 2 中的位 INTDE=1 时倒计时中断被允许。

倒数定时器的频率源由控制寄存器 2 中的位 TDS1、TDS0 来选定。

TDS1	TDS0	定时器频率源(Hz)
0	0	4096
0	1	64
1	0	1
1	1	1/60

当 INTDE=1 且倒数定时器写入一个八位自动重置的二进制数后，倒数定时器会按照 TDS1、TDS0 选定的频率时间来减一。每次当倒数定时器为零时,会置倒计时中断的标志位(控

制寄存器 1 中的位 INTDF)为 1。倒计时中断最长周期为 256 分钟。

当设置 INTS1=1、INTS0=1、IM=0 时,即允许倒计时中断的低电平从 INT 脚输出. 置 INTDF=0 时 INT 脚输出变为高电平直到下一次倒计时中断 INT 脚再次变为低电平;当设置 INTS1=1、INTS0=1、IM=1 且倒数定时周期大于 250ms 时,则从 INT 脚输出低电平时间为 250ms 连续脉冲。

当设置 INTDE=0 时倒计时中断被禁止或复位。

特别的:当重新配置倒计时中断时,需要复位倒计时计数器,即置 INTDE=0,然后再置 INTDE=1,才可以启用新的倒计时中断。

4.4 数字化时间精度调整电路(12H 地址)

利用数字化时间精度调整电路可以每 20 秒改变当前 1 秒所包含的 32768Hz 脉冲的个数,从而到达时钟走时调整,使 SD2403API 保持高的走时精度.相关寄存器为时间调整寄存器(内部地址 12H):

时间调整寄存器:(缺省值为 00H)

D7	D6	D5	D4	D3	D2	D1	D0
0	F6	F5	F4	F3	F2	F1	F0

F6~F0:时间调整位;时间调整电路是在当秒计数为 00,20,40 时刻,根据预先设置的数据(F6~F0)改变 1 秒时钟内计数的个数.通常每 32768 个脉冲为 1 秒(对寄存器预设初值,才能激活整个调整电路)。

当 F6 为"0"时,产生 1 秒的寄存器计数脉冲将增加为 32768+((F5,F4,F3,F2,F1,F0)-1) × 2;当 F6 为"1"时,产生 1 秒的寄存器计数脉冲将减少为 32768-((/F5,/F4,/F3,/F2,/F1,/F0)+1) × 2;(/F5 是 F5 的反码,其它类同)当(F6,F5,F4,F3,F2,F1,F0)预设为(*,0,0,0,0,0,*)时, 产生 1 秒的寄存器计数脉冲不变。

例:当(F6,F5,F4,F3,F2,F1,F0)=(0,1,0,1,0,0,1)且当 00,20,40 秒时刻时,寄存器计数脉冲变为 32768+(41-1) ×2=32848;当 (F6,F5,F4,F3,F2,F1,F0)=(1,1,1,1,1,1,0) 且当 00,20,40 秒时刻时,寄存器计数脉冲变为 32768-(1+1) × 2=32764;当(F6,F5,F4,F3,F2,F1,F0)=(0,0,0,0,0,0,1)且当 00,20,40 秒时刻时,寄存器计数脉冲保持为 32768 不变。

因为每 20 秒增加或减少计数脉冲的最小个数为 2,,所以时钟调整寄存器的最小调整精度是:2/(32768×20)=3.05ppm。

注意: 时钟调整电路仅是调整的时钟走时,并不对晶振本身频率调整,所以 32.768KHz 脉冲输出没有变化。

时钟调整寄存器的调整值计算方法:

当晶振频率大于目标频率(32768Hz),则需要增加 1 秒内的计数脉冲:

$$\text{调整值}=\frac{(\text{晶振频率}-\text{目标频率}+0.1)}{\text{晶振频率}*2/\text{目标频率}*20}=(\text{晶振频率}-\text{目标频率})*10+1$$

其中:晶振频率是从 INT 脚测出的频率中断标称为 32768Hz 的频率(即 FS3、FS2、FS1、FS0=0001),下同;

调整值是设置 F6~F0 的数值,该数值是用二进制补码形式表示.下同

当晶振频率小于目标频率(32768Hz),则需要减少 1 秒内的计数脉冲:

$$\text{调整值}=\frac{(\text{晶振频率-目标频率})}{\text{晶振频率*2/目标频率*20}}=(\text{晶振频率-目标频率})*10$$

当晶振频率等于目标频率(32768Hz),则不需要改变 1 秒内的计数脉冲。

调整值=0,+1,-64,-63 或者缺省。

计算调整数值大小示例:

晶振频率=32770Hz, 目标频率=32768Hz

调整值=(32770-32768+0.1)/(32770*2/(32768*20))=(32770-32768)*10+1=21

故设置(F6,F5,F4,F3,F2,F1,F0)=(0,0,1,0,1,0,1)

2) 晶振频率=32762Hz, 目标频率=32768Hz

调整值=(32762-32768)/(32762*2/(32768*20))=(32762-32768)*10=-60

因为-60 的 7 位补码是 80H-3CH=44H

故设置(F6,F5,F4,F3,F2,F1,F0)=(1,0,0,0,1,0,0)

经过数字化时间精度电路调整后,时钟精度与目标频率相差±1.5ppm(在室温下)

但是要注意:时间调整电路不能改变从 INT 输出频率中断的频率;

最大调整范围:晶振频率大于目标频率时,调整数值范围(F6,F5,F4,F3,F2,F1,F0)=(0,0,0,0,0,0,0)到(0,1,1,1,1,1,1),实际可调范围从-3.05ppm到-189.2ppm。

b.晶振频率小于目标频率时,调整数值范围(F6,F5,F4,F3,F2,F1,F0)=(1,1,1,1,1,1,1)到(1,0,0,0,0,1,0),实际可调范围从+3.05ppm 到+189.2ppm。

特别的:虽然 SD2403API 的上电复位功能会在所有电源(含电池)下电时复位数字调整寄存器 12H 为 0,但在电源环境比较恶劣的条件下并不能绝对保证模块每一次上电的可靠复位。针对绝大多数不使用数字调整功能的用户, 为了保证走时精度的可靠性,强烈建议能在写时间的程序里加上清 12H 寄存器为 0 的程序。

4.5 通用 RAM(14H~1FH 地址)

SD2403API 内部提供 12 个字节的通用静态 RAM 供用户存储数据.

通用 SRAM 的缺省值是 XXH.(即不确定值)

4.6 其它控制/状态位

(1)WRTC1、WRTC2、WRTC3 位: 寄存器(00H~1FH)写允许位。即 WRTC1=1、WRTC2=1、WRTC3=1 时写允许。注意置位有先后顺序, 先置 WRTC1 为 1,后置 WRTC2、WRTC3 为 1;当 WRTC1=0、WRTC2=0、WRTC3=0 时则写禁止, 同样置位有先后顺序, 先置 WRTC2、WRTC3 为 0,后置 WRTC1 为 0。当写禁止时, 除了以上三位可以写以外, 从 00H 到 1FH 所有的寄存器均不可以写。写

禁止并不影响读操作。

(2)ARST 位: 自动复位使能位。对控制寄存器 1 的 INTAF、INTDF、BAT 位的自动复位进行使能/禁止。当 ARST=1 时,对控制寄存器 1 进行一次有效的读操作后,以上三个状态位均复位为”0”。若 ARST=0,则需要对 INTAF、INTDF、BAT 位进行手动复位方可清零。

(3)FOBAT 位:FOBAT=0 时,当处于 VBAT 模式下,INT 脚输出禁止;FOBAT=1 时,当处于 VBAT 模式下,INT 脚输出允许。该位在 VDD 模式下不起作用。(要注意在 VBAT 模式下 INT 脚输出会引起电池的消耗)

(4)RTCF 位:上电位,全部电源失效后再上电则该位置”1”,为只读位。上电后的第一次有效写(只要写一个字节即可)就可以将 RTCF 位清为”0”。

5.串行 IIC 接口

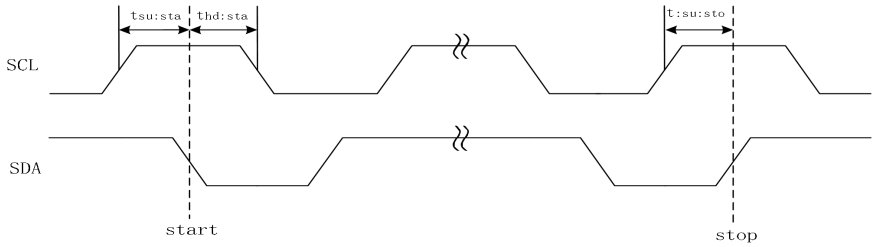
5.1 SD2403API 通过两线式 IIC 串行接口方式接收各种命令并读写数据。两线式串行 IIC 接口方式描述如下:

（1）开始条件

当 SCL 处于高电平时，SDA 由高电平变成低电平构成一个开始条件，对 SD2403API 的所有操作均必须由开始条件开始。

（2）停止条件

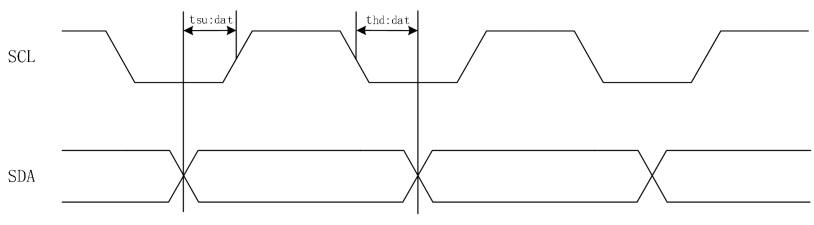
当 SCL 处于高电平时，SDA 由低电平变成高电平构成一个停止条件，对 SD2403API 的所有操作均停止，系统进入待机状态。



实时时钟串行接口

（3）数据传输

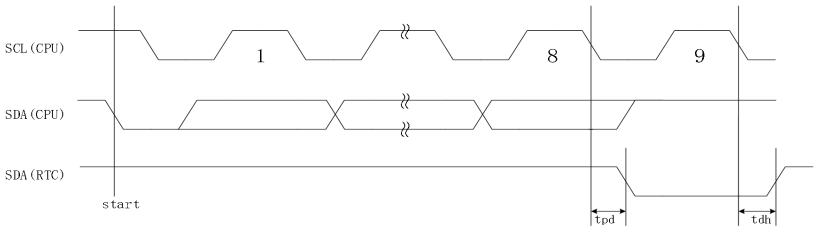
当 SCL 为低电平，且 SDA 线电平变化时，则数据由 CPU 传输给 SD2403API（高位在前、低位在后，下同）；当 SCL 为高电平，且 SDA 电平保持不变时，则 CPU 读取 SD2403API 发送来的数据：当 SCL 为高电平，且 SDA 电平变化时，SD2403API 收到一个开始或停止条件。



实时时钟数据传输时序

（4）确认

数据传输以 8 位序列进行。SD2403API 在第九个时钟周期时将 SDA 置位为低电平，即送出一个确认信号（Acknowledge bit，以下简称“ACK”），表明数据已经被其收到。



实时时钟确认信号

5.2 数据/指令传输格式

当 CPU 发出开始条件与实时时钟建立连接后， CPU 首先通过 SDA 总线连续输出 7 位器件地址和 1 位读/写指令来唤醒 SD2403API。

(1) 器件代码：

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	1	1	0	0	1	0	R/W

其中高 7 位 BIT7~BIT1 称“器件代码”，它代表实时时钟的器件地址，固定为“0110010”;BIT0 为读/写位,”1”为读操作,”0”为写操作。

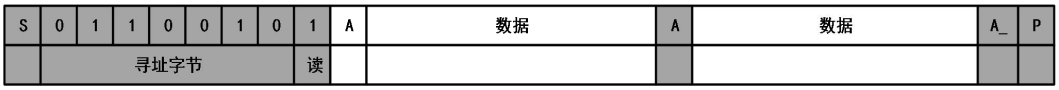
(2) 数据传输格式

在数据发送/接收时停止信号到来时,将结束其数据传输,同时内部 5 位地址归零(注:内部 5 位地址的缺省值为 00000B)。如果只有开始信号,而没有结束信号,接着重新产生起始信号,则还要重新设置器件代码(在传输方向需要改变时,就用这种传输方式,如下面的读数据方式 1)。

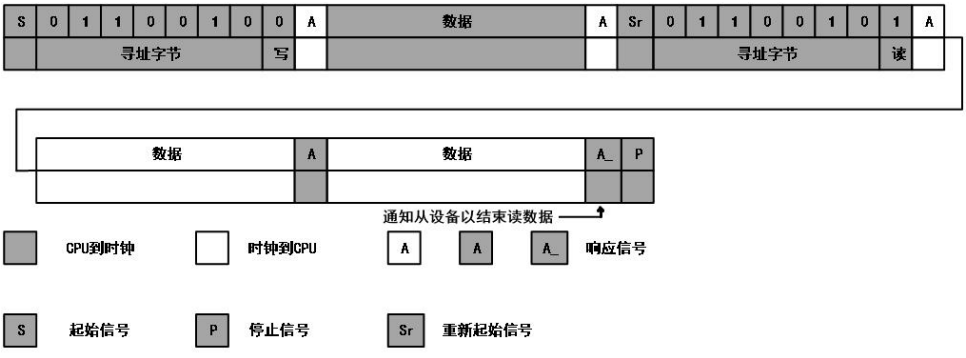
主设备向从设备写入数据过程如下图：



主设备向从设备直接读取数据过程如下图：



数据传输时改变其传输方向过程图：



（3）SD2403API 数据传输的写模式

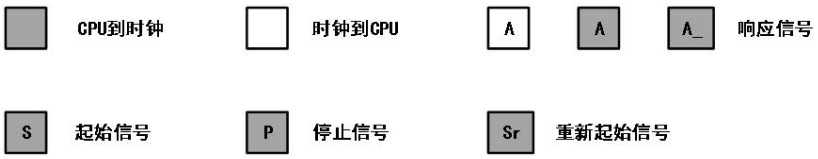
先送 7 位器件地址(0110010),第 8 位送入写命令(“0”), 第 9 位是 SD2403API 的响应位(ACK),SD2403API 进入写状态;

接下来一个字节, 高 3 位是传输模式(写状态时, 传输模式定义为”000”),低 5 位确定 SD2403API 的内部地址(00H~1FH), , 第 9 位是 SD2403API 的响应位;

开始写数据,每写完 1 个字节的数据之后,都经过 1 位的响应信号才能写下 1 字节的数据;如果要结束写数据过程,则在 ACK 后送出停止命令即可。

SD2403API 写数据示例(向 14H,15H 地址写数据):





特别注意:

1.除了 WRTC1、WRTC2、WRTC3 三个写允许位，对寄存器(00H~1FH)的写操作必须确认模块处于写允状态,否则写无效。

2.写时间同步：每次对实时时间寄存器的写操作时，都会对秒以下的内部计数器清零，使时间同步。

3.从当前地址开始,每次读写完一个字节地址自动加 1 直到 1FH,读写完 1FH 后地址自动加 1 后为 00h。

4.如果写入的时间数据不存在,则不改写相对应的时间寄存器的值。

5.为了提高数据的可靠性，当写完成后，**应将模块置于写禁止状态。（参见 4.6）**

（4）SD2403API 数据传输的读模式

SD2403API 有两种读数据方法:

I)读方法 1:从指定的内部地址中读取数据。

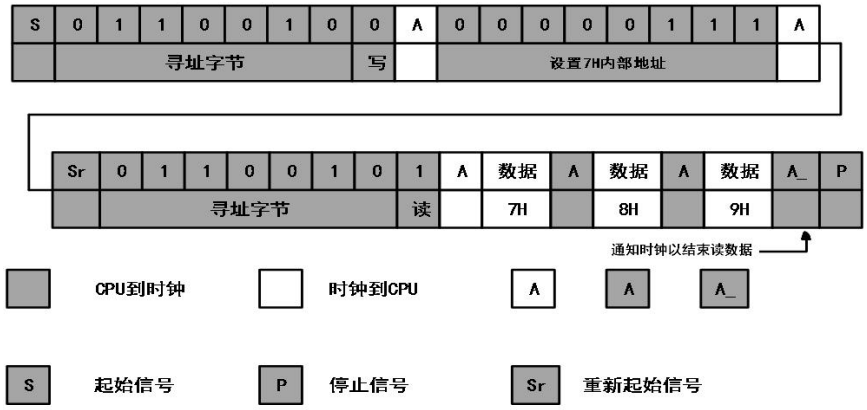
. 与写模式的前两步一样。

. 重新发出开始命令以改变两线接口数据传输方向。

. 再送 7 位器件地址(0110010),第 8 位送入读命令(“1”), 第 9 位是 SD2403API 的响应位(ACK),SD2403API 进入读状态。

. 开始读数据,每读完 1 个字节的数据之后,CPU 都要送出 1 位的响应信号(ACK,低电平)才能读下 1 字节的数据;如果想要结束读数据过程,则 CPU 要送出 1 位的响应信号(ACK_,高电平), ACK_后送出停止命令即可。

SD2403API 读数据方法 1 示例(从 7H~9H 地址读取数据):

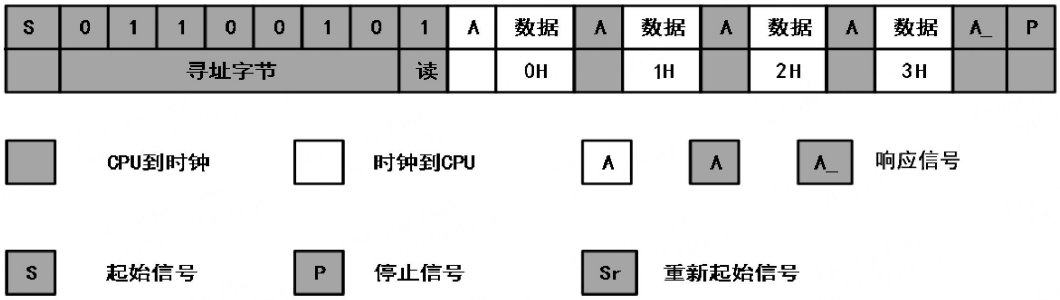


II)读方法 2:直接读取数据(从内部地址 00H 开始)

. 开始信号后,先送 7 位器件地址(0110010),第 8 位送入读命令(“1”), 第 9 位是 SD2403API 的响应位(ACK),SD2403API 进入读状态。

. 每读完 1 个字节的数据之后,CPU 都要送出 1 位的响应信号(ACK,低电平)才能读下 1 字节的数据;如果想要结束读数据过程,则CPU要送出1位的响应信号(ACK_,高电平), ACK_ 后送出停止命令即可。

SD2403API 读数据方法 2 示例(从 00H 地址开始读取数据):



（5）SD2403API 在特殊条件下的数据传输

为了保证读写数据的有效性,SD2403API 的两线通信开始到结束仅在此 0.5S 秒之内,如此可避免总线挂死的现象。

因此在 SD2403API 中, IIC 通信方式会在第一个起始信号到来之后的 0.5 秒之内自动终止本次通信。所以,要注意: 从开始信号进行读/写数据,直到停止信号, 读/写操作过程必须在 0.5 秒之内完成。特别：在进入 VBAT 模式后模块会禁止 IIC 总线的通信。

6. 备用电源切换电路

SD2403API 具有一个后备电池输入脚 V_{BAT},当模块检测到主电源 V_{DD} 掉到后备电池电压下时自动转为由接在 V_{BAT} 的后备电池供电。

从正常模式（V_{DD}）切换到后备电池供电模式（V_{BAT}）

要从 V_{DD} 过渡到 V_{BAT} 模式，必须同时具备以下条件：

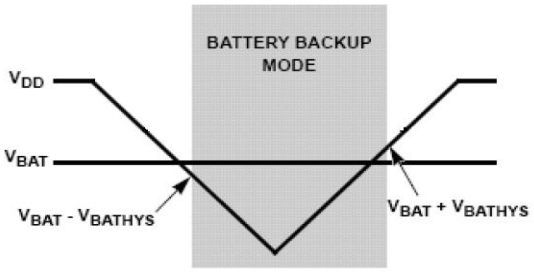
条件：V_{DD}＜V_{BAT} -V_{BATHYS}，其中 V_{BATHYS}≈300mV

从后备电池供电模式（V_{BAT}）切换到正常模式（V_{DD}）

在遇到以下条件时，SD2403API 会从 V_{BAT} 切换到 V_{DD} 模式：

条件：V_{DD}＞V_{BAT}＋V_{BATHYS}，其中 V_{BATHYS}≈300mV

电源控制情况如图所示:



电池切换

为提供更低的电源，IIC 总线在后备电池供电模式中被禁止。除此之外，在后备电池供电模式中实时时钟的所有功能均可实现。除了 SCL 和 SDA 以外，在后备电池供电模式中，SD2403API 所有的输入端和输出端均有效，除非通过控制寄存器来禁止。在后备电池供电模式电池电压低到 1.5V 时，用户 SRAM 仍可保存数据。

电源失效检测

SD2403API 有一个实时时钟失效位（RTCF），用于检测总电源失效。它在器件丢失所有电源（V_{DD} 和 V_{BAT}）之后使用户可以确定器件是否已上电。

特别情况：当 V_{DD}=3.3V 时，V_{BAT} 接 3V 电池会有一些问题：通常标称 3V 的电池在刚开始使用时电压会达到 3.3V 或更多，这是模块会进入 V_{BAT} 模式而读不出数据。要解决此问题，应将 V_{BAT} 脚接地，电池和主电源均通过二极管向模块的 V_{DD} 供电请见**“10.应用参考电路”**。

7. 上电复位

模块内部具有上电复位电路,复位操作对内部部分寄存器进行置初值但不包括实时时钟数据寄存器、通用 RAM。

8. 使用说明

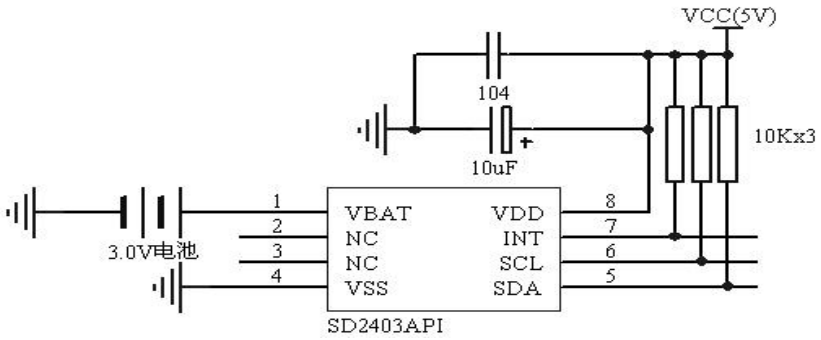
1.为了防止电路噪声问题,请在此模块的旁边放置两个旁路电容,分别是 0.1uF 电容和 22uF 电容。

2.为了防止干扰,在 PCB 制作时请保证模块底部无大电流信号通过,最好能铺地。

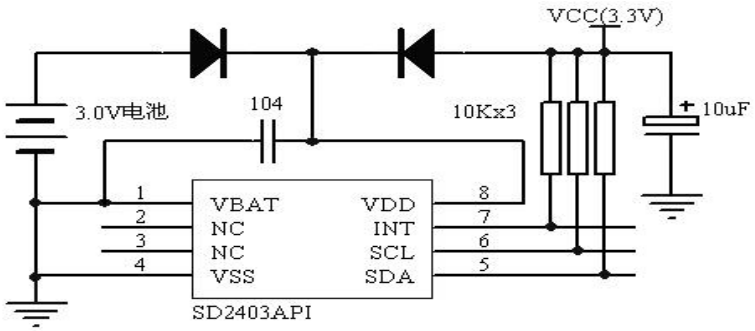
3.使用时,除 NC 脚外,其它不用的脚如 INT 可通过 510K 电阻接地以省电。

9. 应用参考电路

a.VCC=5.0V



b.VCC=3.3V



10. 极限参数

VDD 、VBAT 、SCL、SDA 和 INT 引脚上的电压（相对于地）…… －0.5V 至 7.0V

引线温度（焊接，10 秒）…… 260℃

注：强度超出所列的极限参数可能导致器件的永久性损坏。这些仅仅是极限参数，并不意味着在极限条件下或在任何其它超出推荐工作条件所示参数的情况下器件能有效地工作。延长在极限参数条件上的工作时间会影响器件的可靠性。

11. 直流特性

符号	参数	条件	最小值	典型值	最大值	单位	注意
V_{DD}	主电源		1.8		5.5	V	
V_{BAT}	电池供电电压		1.5		5.5	V	
I_{DD1}	电源电流	V_{DD}=5V		1.6	3.0	μA	
		V_{DD}=3V		1.0	1.6	μA	
I_{DD2}	供电电流 IIC 活动	V_{DD}=5V		40	120	μA	
I_{BAT}	电池供电电流	V_{BAT}=3V		800		nA	
I_{LI}	SCL 的输入漏电流			100		nA	
I_{LO}	SDA 上的 I/O 泄漏电流			100		nA	
V_{BATHYS}	VBAT 和VDD之间切换的迟滞电压			300		mV	
INT V_{OL}	低电平输出电压	V_{DD}=5V I_{OL}=3mA			0.4	V	

12. 掉电时序(温度:－40℃至+85℃)

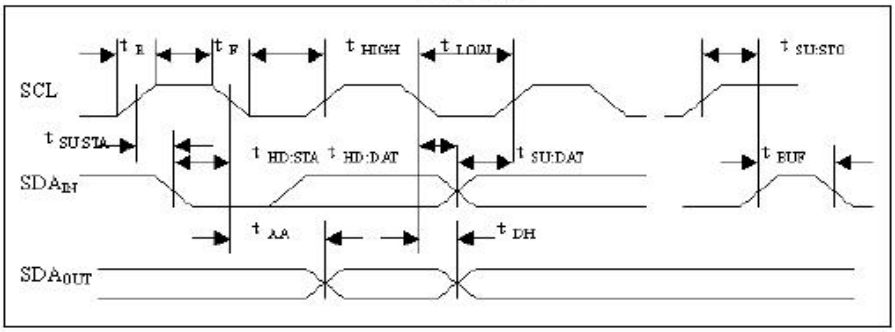
符号	参数	条件	最小值	典型值	最大值	单位	注意
$V_{DD\ br}$	V_{DD} 下电速率				10	V/ms	

13. 交流特性

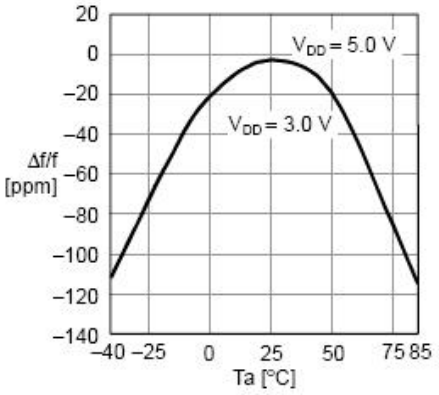
符号	参数	条件	最小值	典型值	最大值	单位
V_{IL}	SDA 和 SCL 低电平输入电压		-0.3		0.3×V_{DD}	V
V_{IH}	SDA 和 SCL 高电平输入电压		0.7×V_{DD}		V_{DD}+0.3	V
Vhys	SDA和SCL施密特触发输入滞后		0.05×V_{DD}			V
V_{OL}	低电平输出电压	SDA 输出低电平状态下，吸收 3mA 电流时的电压	0		0.4	V
C_{pin}	SDA 和 SCL 引脚电容	T_A=25℃ f=1MHz V_{DD}=5V V_{IN}=0V V_{OUT}=0V			10	pF
f_{SCL}	SCL 频率				400	
t_{IN}	SDA和SCL输入端的脉冲宽度抑制时间				50	ns
t_{AA}	SCL下降沿到SDA输出数据有效	SCL 下降到 0.3×V_{DD}，直到 SDA 不在 0.3×V_{DD}至 0.7×V_{DD}区间。			900	ns
t_{BUF}	总线在 STOP 和 START 之间的空闲时间	SDA 在 STOP 条件下上升到 0.7×V_{DD}，在开始条件下下降到 0.7×V_{DD}。	1300			ns
t_{LOW}	时钟低电平时间	在 0.3×V_{DD}处测量	1300			ns
t_{HIGH}	时钟高电平时间	在 0.7×V_{DD}处测量	600			ns

$t_{SU:STA}$	启动条件设置时间	SCL 上升到 0.7×V_{DD} 至 SDA 下降沿到 0.7×V_{DD}	600			ns
$t_{HD:STA}$	起始条件保持时间	在 SDA 下降沿到 0.3×V_{DD}，SCL 下降沿到 0.7×V_{DD}	600			ns
$t_{SU:DAT}$	输入数据的建立时间	从 SDA 不在 0.3×V_{DD} 至 0.7×V_{DD} 范围，到 SCL 上升沿的 0.3×V_{DD}	100			ns
$t_{HD:DAT}$	输入数据保持时间	SCL 下降沿的 0.3×V_{DD}，到 SDA 在 0.3×V_{DD} 至 0.7×V_{DD} 区间	0		900	ns
$t_{SU:STO}$	停止条件的建立时间	从 SCL 上升沿经过 0.7×V_{DD}，到 SDA 上升沿 0.3×V_{DD}	600			ns
$t_{HD:STO}$	输出条件的保持时间	从 SDA 上升沿到 SCL 下降沿。两者均超过 VDD 的 70%。	600			ns
t_{DH}	数据输出的保持时间	从 SCL 下降沿 0.7×V_{DD}，到 SDA 0.3×V_{DDD} 至 0.7×V_{DD} 区间	0			ns
t_R	SDA 和 SCL 的上升时间	0.3×V_{DD} 至 0.7×V_{DD} 区间	20+0.1×Cb		300	ns
t_F	SDA 和 SCL 的下降时间	0.3×V_{DD} 至 0.7×V_{DD} 区间	20+0.1×Cb		300	ns
Cb	SDA 或 SCL 的容性负载	片内和片外合计	10		400	PF
R_{PU}	片外 SDA 和 SCL 总线上拉电阻	最大值由 t_R 和 t_F 确定， 当 Cb=400pF 时，最大值约为 2~2.5kΩ 当 Cb=40pF 时，最大值约为 15~20kΩ。	1			kΩ

总线时序图

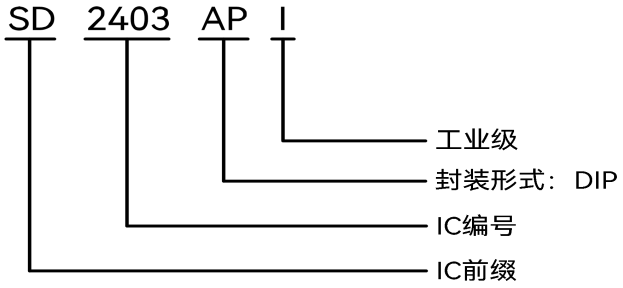


14. 频率误差&温度关系曲线

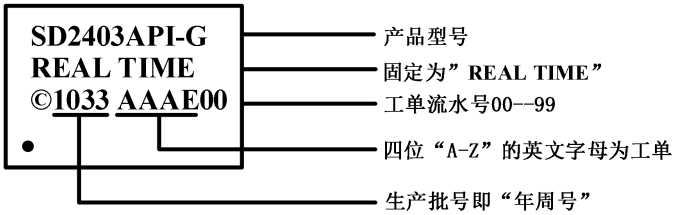


15. 模块型号及顶部字符说明

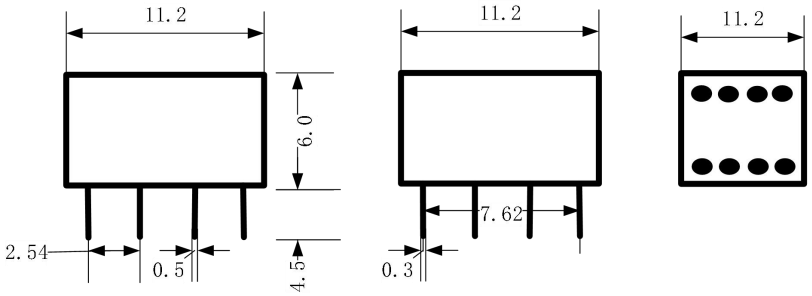
（1）型号说明



（2）模块顶部字符说明



16. 封装尺寸（DIP8, 单位：mm；尺寸误差±0.2mm, 特别说明除外。）



SD2403API 封装尺寸（管脚尺寸：0.5*0.3mm，管脚公差±0.05mm）

■ 编后语

感谢您阅读本资料。由于经验和水平的欠缺，本文难免有错误和遗漏。如果您在使用过程中发现错误或不恰当的地方，请拨打电话：0755-83246178 或请 E-mail：support@whwave.com.cn,我们将尽快予以答复。

谢谢您的支持与合作！

注：

本资料中的内容如有变化，恕不另行通知。

本资料提供的应用线路及程序仅供参考，本公司不承担由此而引起的任何损失。

由于本公司的产品不断更新和提高,希望您经常与本公司联系，以索取最新资料。

本公司不承担在任何使用过程中引起的侵犯第三方专利和其它权利的责任。

注：本文档受中国版权法保护,非授权禁止拷贝、复制、引用或传播

（SD 及 WAVE 均为我公司注册商标）

深圳市兴威帆电子技术有限公司

版本号	修改时间	修改记录	修改人
V2.2	2026.03.31	补充封装尺寸部分	CHENMY

附录:

一、SD2403 应用中硬件注意事项:

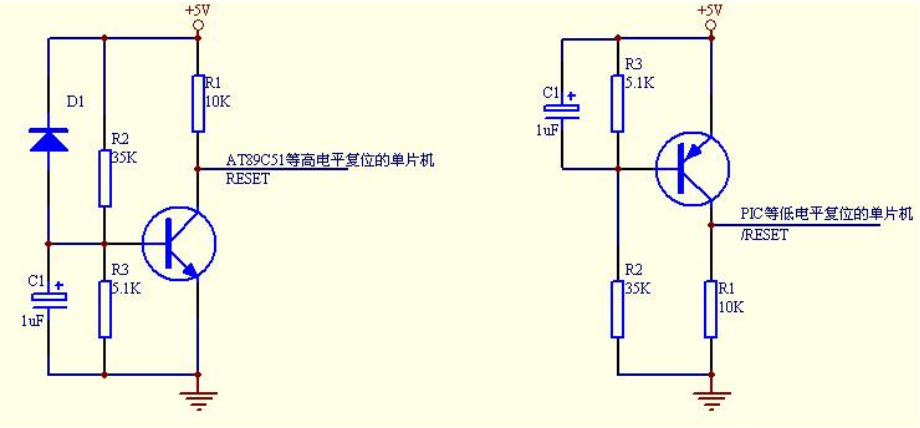
(1) 对 SD2403 及 MCU 的电源~地之间加 104 电容去高频。

(2) 对 SD2403 及 MCU 所在的板数字电源~地的输入端加 220uF 以上的电解及 104 电容去除电源扰动.

(3) 为了防止干扰, 在 PCB 制作时请保证模块底部无大电流信号通过, 最好能铺地.

(4) 对 SD2403,在 VDD 和电源之间串一个 200 欧的电阻防干扰.

(5) 对 MCU 的复位端尽量采用可靠的复位方式,而摒弃阻容复位或直接连到 VDD 的方式.以下推荐廉价的三极管复位方式,当然用电压检测器或专用复位电路就更好啦.



图中 R2,R3 为分压电阻,其选值可用以下公式:

Vbe/Vrst=R2/(R2+R3)

其中:Vbe 为三极管 b 极和 e 极之间导通电压

Vrst 为单片机的复位电压

C1 为延时电容

D1 为延时电容的放电二极管,可选用.

(6) 电源电压>=3.3V !!! 主要是因为正常工作时需要 VDD>VBAT。

二、SD2403 应用中软件注意事项

(1) 软件上电开始做一个几百毫秒的延时，等 MCU 端口稳定后再去进行读写操作。

(2) 时钟最多半秒才读一次：这样做主要目的是考虑到时间数据的改变需要 1S，频繁去读的话是在做无用功，不但会加大电源功耗，而且会引入干扰。

IIC 总线"START"里:在置 SDA 为高后要再判断 SDA 是否为高,即 SDA 是否被箝位为低,否则等 SDA 线置高。.

在写命令字时:要判断 ACK 是否正常,否则退出

(3) IIC 总线"STOP"里:在置 SDA 为高后要再判断 SDA 是否为高,即 SDA 是否被箝位为低,否则多置

SCL 脉冲让 SDA 线释放为高.在写命令字时:要判断 ACK 是否正常,否则退出，不要再进行下面的操作。

(4) 写时间时，强烈建议能对时间调整寄存器(12H)清零,在网上有 51 程序可参考。

三、有关 SD2403 的一些问题回答

(1) 时钟精度:

SD2403 的精度指标为常温 25 度下±5ppm,即年误差少于 2.5 秒.

出厂之前对每一只 SD2403 都用超高精度的仪器和严格的程序做了校准,所以能保证精度.有人说他拿到 SD2403 的精度好像不准,一般问题出在参考时钟上,通常可用电信 12117 台作基准来验证,但如用电脑那就差远了,因为电脑的时钟精度一般都不高,通常大于 30ppm.最好用网络校时软件.

电脑版校时软件：

http://www.whwave.com.cn/download/wave/AboutTime.rar

手机安卓版网络/GPS 校时软件：

http://zhushou.360.cn/detail/index/soft_id/3179034?from=onebox

(2) 温度对精度的影响:

对目前没有温度补偿的 SD2403 时钟:在低温(-40)或高温(80)时,因为晶振的温度特性的原故,时钟误差一般大于-100ppm.即年误差超过 52.56 分钟

如果要更高精度,可选内置温度传感的 SD2404、SD2500 时钟或在 SD2403 外加温度传感器来补偿。