

高精度 WIFI 网络校时实时时钟 – SD2431API

内置无线网络校时、晶振、充电电池、IIC 串行接口、多种中断输出、超高精度

1. 概述

SD2431API 是一种具有内置 WIFI 网络校时电路、晶振、IIC 串行接口的高精度实时时钟芯片, CPU 可使用该接口通过 6 位地址来寻址读写片内 64 字节的数据(包括时间寄存器、报警寄存器、控制寄存器、通用 SRAM 寄存器), 还可以通过该接口来收发网络数据。

SD2431API 芯片在无网络校时的情况下, 可保证时钟精度为 $\pm 5\text{ppm}$ (在 $25\pm 1^\circ\text{C}$ 下), 即年误差小于 2.5 分钟; SD2431API 芯片在有网络校时的情况下, 可保证时钟精度为 ± 1 秒; 该芯片内置充电电池, 在电池满充的情况下可保证内部时钟走时超过半年时间, 累计电量超过 550mAh, 电池使用寿命为五至八年。该芯片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(分辨力 3ppm), 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。

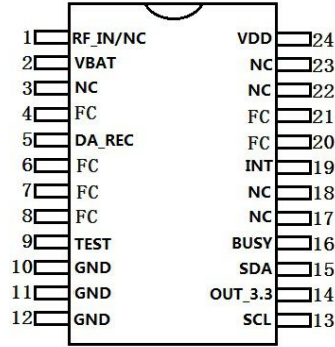
SD2431API 芯片内置单路定时/报警中断输出, 报警中断时间可最长设至 100 年; 该系列芯片可满足对实时时钟芯片的各种需要, 是在选用高精度实时时钟时的理想选择。

2. 主要性能特点:

- 低功耗: 正常工作典型值 7uA ($V_{\text{BAT}}=3.0\text{V}$, $T_a=25^\circ\text{C}$), WIFI 校时时功耗: 典型值 90mA。
- 超高精度: WIFI 网络校时, 精度最多 1S 误差, 且没有累计误差。
- 内置电源管理: WIFI 网络校时完成后, 自动关闭 WIFI 电路电源, 减小功耗。
- 标准 IIC 总线进行网络数据传输。
- 工作电压: 有 3.3V 和 5.0V 两个版本, 工作温度: $-40^\circ\text{C}\sim+85^\circ\text{C}$ 。
- 标准 IIC 总线接口方式, 最高速度 400kHz。
- 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。
- 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, 可选从 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。
- 43Bytes 通用 SRAM 寄存器可用于存储用户的一般数据。
- 内置充电电路, 充电型电池累计电量超过 550mAh, 使用寿命为 5~8 年。
- 外加后备电池输入脚 V_{BAT} : 当内部电池因寿命等原因报废时, 可用外加的电池给时钟作备电
- 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 V_{BAT} 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芯片的误写操作, 进一步提高时钟芯片的可靠性。
- 内置电源稳压, 内部计时电压可低至 1.5V。
- 内置上电复位电路及指示位。
- 芯片管脚 ESD>2KV。
- 芯片在兴威帆的评估板上可通过 4KV 的群脉冲(EFT)干扰。

- CMOS 工艺
- 内置晶振，出厂前已对时钟进行校准，在无 WIFI 网络校时的情况下保证精度 $\pm 5\text{ppm}$ ，即时钟年误差小于 2.5 分钟(在 $25\pm 1^\circ\text{C}$ 下)。

3. 管脚设置



SD2431API 芯片管脚图

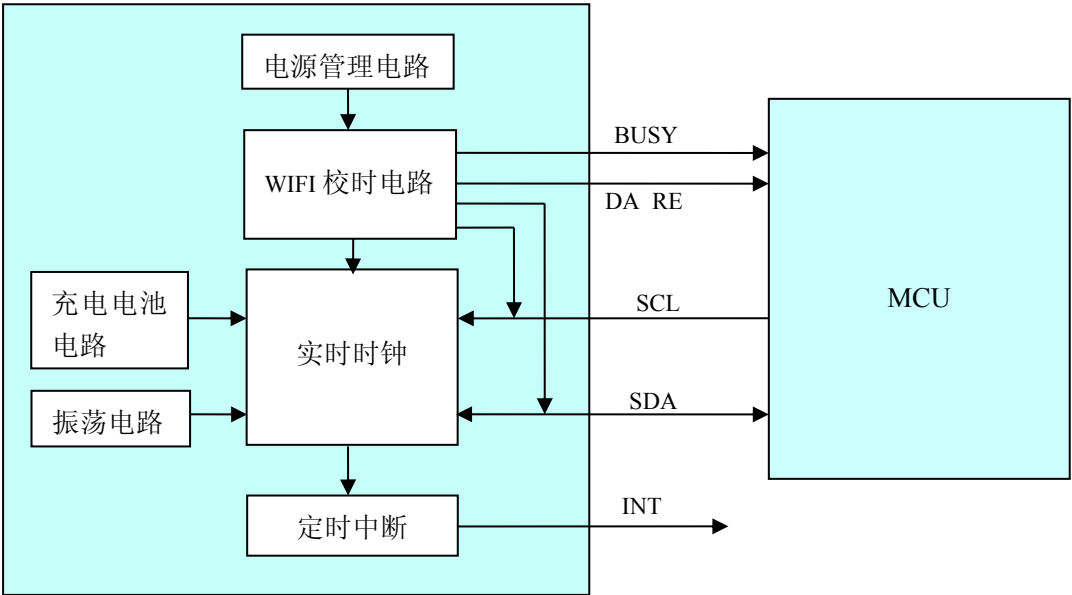
注：FC (Forbid connect)引脚和芯片的内部电路相连，禁止与外部电路连接。
NC 引脚没有和芯片的内部电路相连，脚位悬空。

SD2431API 管脚功能表

标号	功能	特征
RF_IN	WIFI 信号输入，通过天线接入。	
DA_REC	用于模块收到网络数据的信号指示 ERR_INFO=0 有网络数据 ERR_INFO=1 无网络数据	在连接网络之后可以通过读 DA_REC 来判断是否接收到网络数据
BUSY	用于与模块通信忙信号指示 BUSY=0: IIC 总线忙 BUSY=1: IIC 总线空闲，可以使用	在对 IIC 总线进行操作之前，必须先读 BUSY;
VBAT	外加备用电池引脚	针对 SD2431API，由于在模块内部有电池，故在其电池能量未耗尽之前不接
TEST	测试	内部电池电压检测脚（通常不接）
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/下降时间，应严格遵守说明书。	CMOS 输入(内部 10K 电阻上拉)
SDA	串行数据输入/输出脚，此管脚通常用一电阻上拉至 VDD，并与其它漏极开路或集电极开路输出的器件通过线与方式连接。	N 沟道输出(内部 10K 电阻上拉), CMOS 输入。
INT	报警中断输出脚，根据中断寄存器与状态寄存器来设置其工作的模式，当定时时间到达时输出低电平或时钟信号。可通过重写状态寄存器来禁止	N-沟道开路输出
VOUT_3.3	3.3V 稳压输出脚，当 VDD $\geq 4.5\text{V}$ 时有效	可供电流 $\leq 100\text{mA}$ ，精度 $3.3\text{V} \pm 2\%$ 。
VDD	正电源	芯片启动电流为 300mA 左右(建议

		使用 500mA 的电源供电, 否则 WIFI 可能无法正常工作)
GND	负电源 (GND)	

3. 原理框图



SD2431API 功能框图

注：在芯片内部 SDA、SCL 对 VDD 已经有 10K 上拉，实际应用中不用外加上拉电阻。

5. 寄存器

5.1 寄存器列表

地址	寄存器段	寄存器名称	BIT								数值范围 (十进制)	缺省值 (二进制)
			D7	D6	D5	D4	D3	D2	D1	D0		
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	H20 P/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	M010	M08	M04	M02	M01	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 AP/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	AM010	AM08	AM04	AM02	AM01	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	EAM0	EAD	EAW	EAH	EAMN	EAS	N/A	0000-0000
0FH		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	控制	CTR3	ARST	0	TDS1	TDS0	FS3	FS2	FS1	FS0	N/A	0000-0000
12H	寄存	时间调整	0	F6	F5	F4	F3	F2	F1	F0	N/A	0000-0000
13H	器	倒计时定时器	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	0-255	0000-0000
14H		无线网络	-	-	S5	S4	S3	S2	S1	S0	0-32	XXXX-XXXX
15~3FH	通用RAM	(43Bytes)	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	N/A	XXXX-XXXX

5.2 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 1~12)：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, 芯片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. 当芯片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。

5.3 中断(08H~13H 地址)

SD2431API 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”。

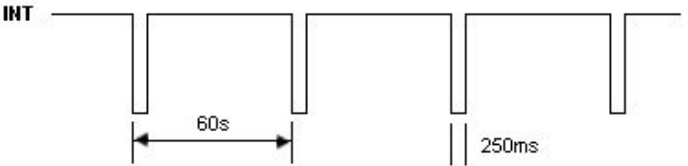
当设置 INTS1=0、INTS0=1 时,即允许报警中断从 INT 脚输出.该报警中断有两种模式,即单事件报警和周期性报警,模式的选定由控制寄存器 2 中的报警中断模式位 IM 的值来定:

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=00100110B,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 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.

5.4 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 SD2431API 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 $32768 + ((F5, F4, F3, F2, F1, F0) - 1) \times 2$;
当 F6 为"1" 时, 产生 1 秒的寄存器计数脉冲将减少为 $32768 - ((\neg F5, \neg F4, \neg F3, \neg F2, \neg F1, \neg F0) + 1) \times 2$; ($\neg F5$ 是 F5 的反码, 其它类同)

当 (F6, F5, F4, F3, F2, F1, F0) 预设 (*, 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 $32768 + (41 - 1) \times 2 = 32848$;

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 $32768 - (1 + 1) \times 2 = 32764$;

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: $2 / (32768 \times 20) = 3.05\text{ppm}$.

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:

- 1) 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

- 2) 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:

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

- 3) 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:

- 1) 晶振频率=32770HZ, 目标频率=32768HZ

$$\text{调整值} = (32770 - 32768 + 0.1) / (32770 * 2 / (32768 * 20)) = (32770 - 32768) * 10 + 1 = 21$$

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:

- 1) 时间调整电路不能改变从 INT 输出频率中断的频率;
- 2) 最大调整范围:
 - a. 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.

5.5 WIFI 网络出错寄存器 (14H)

BIT	D7	D6	D5	D4	D3	D2	D1	D0
位名	-	-	S5	S4	S3	S2	S1	S0
			建立网络连接出错	网络数据发送失败	网络校时失败	WIFI 电路启动失败	网络配置数据写入出错	WIFI 模块初始化错误

S0~ S5 相应位为 1 时,网络连接出错。当出现 S0、S2 位为 1 时,应检查线路连接并检查供电电源是否满足要求。当出 S1 位为 1 时,检查网络传输配置指令和网络连接配置指令是否传输正确。当出现 D3 位为 1 时,检查连接的 WIFI 是否能够连接 Internet。当出现 S4、S5 位为 1 时,应检查网络服务器是否开启、网络传输配置是否正确。

5.6 通用 RAM 区(15H~3FH 地址)

SD2431API 内部提供 43 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)

5.7 其它控制/状态位

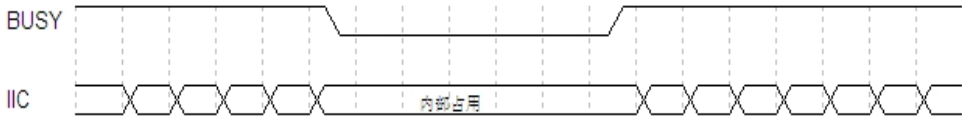
- (1) **WRTC1、WRTC2、WRTC3** 位: 寄存器(00H~3FH)写允许位。即 WRTC1=1、WRTC2=1、WRTC3=1 时写允许。**注意置位有先后顺序,先置 WRTC1 为 1,后置 WRTC2、WRTC3 为 1;**当 WRTC1=0、WRTC2=0、WRTC3=0 时则写禁止,同样置位有先后顺序,先置 WRTC2、WRTC3 为 0,后置 WRTC1 为 0。当写禁止时,除了以上三位可以写以外,从 00H 到 3FH 所有的寄存器均不可以写。**写禁止并不影响读操作。**
- (2) **ARST** 位: 自动复位使能位。对控制寄存器 1 的 INTAF、INTDF、BAT 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” .

6. 用户接口

6.1 SD2431API 与用户通过 4 线进行通信, BUSY, DA_REC, SDA,SCL。BUSY 线用于为了避免内部 WIFI 校时电路总线竞争的忙信号, 发送网络校时指令来开启校时电路, 两线式 IIC 串行接口方式接收各种命令并读写数据。在网络数据传输时, 当获取到网络数据时 DA_REC 脚为低电平。

在使用 IIC 总线前, 先判断 BUSY 的状态, BUSY=0, 内部 WIFI 校时电路正在使用 IIC 总线, 此时禁止操作 IIC; BUSY=1, 总线空闲, 可以对 IIC 总线进行操作。**在传输网络指令完成之后, 需要延时 2us 再读取 BUSY。**



注: 1、主机应避免在 WIFI 校时时长时间占用 IIC 总线, 这样会导致 WIFI 校时模块抢占不到 IIC 总线而不能正常校时。

2、可自由设置校时次数, 推荐每天校时一次, 可以采用如下两种方法: 查询法, 查询在特定的时间校时, 比如在 12:00:00 校时; 中断法: 使用时钟中断报警功能, 有中断即校时。

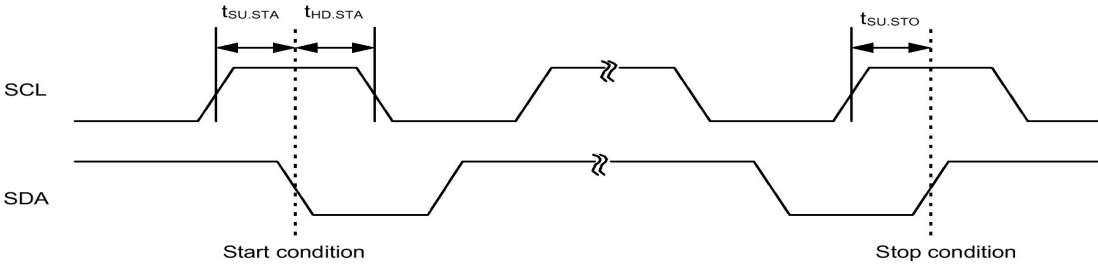
6.2 SD2431API 通过两线式 IIC 串行接口方式接收各种命令并读写数据。两线式串行 IIC 接口方式描述如下:

(1) 开始条件

当 SCL 处于高电平时, SDA 由高电平变成低电平时构成一个开始条件, 对 SD2431API 的所有操作均必须由开始条件开始。

(2) 停止条件

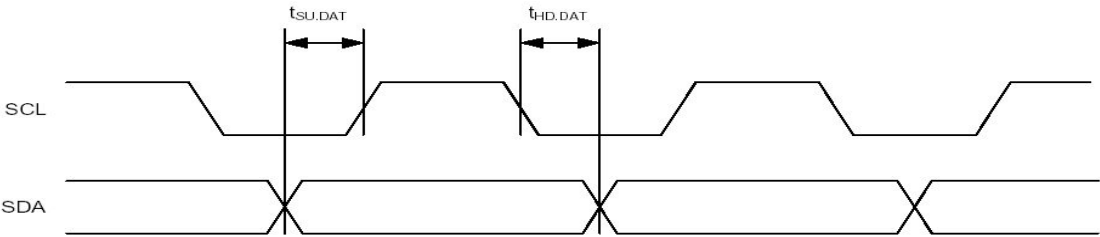
当 SCL 处于高电平时, SDA 由低电平变成高电平时构成一个停止条件, 此时 SD2431API 的所有操作均停止, 系统进入待机状态。



实时时钟的串行接口

(3) 数据传输

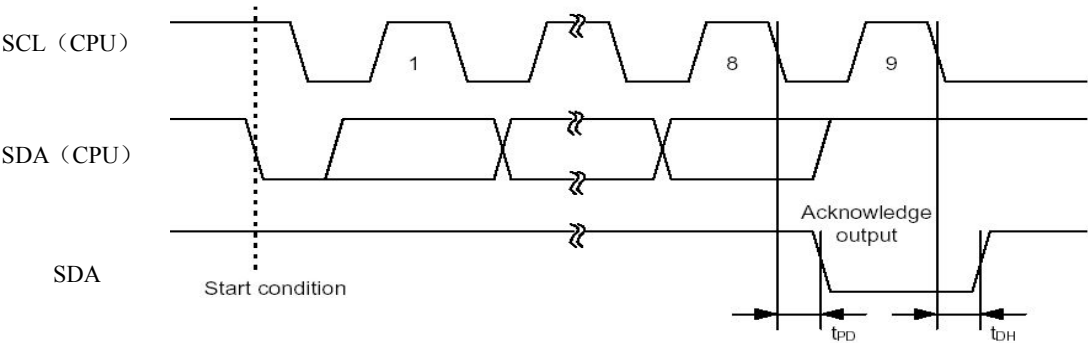
当 SCL 为低电平, 且 SDA 线电平变化时, 则数据由 CPU 传输给 SD2431API(高位在前、低位在后,下同); 当 SCL 为高电平, 且 SDA 线电平不变时, 则 CPU 读取 SD2431API 发送来的数据; 当 SCL 为高电平, 且 SDA 电平变化时, SD2431API 收到一个开始或停止条件。



实时时钟数据传输时序

(4) 确认

数据传输以 8 位序列进行。SD2431API 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。**特别地对地址为 20H~3FH 的 SRAM，在写时 SD2431 不会发出 ACK 信号,读时仍然需要 CPU 发出 ACK 或 NOACK 信号。**



实时时钟确认信号

6.3 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 SD2431API.

(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,同时内部六位地址归零(注:内部六位地址的缺省值为 000000B).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).

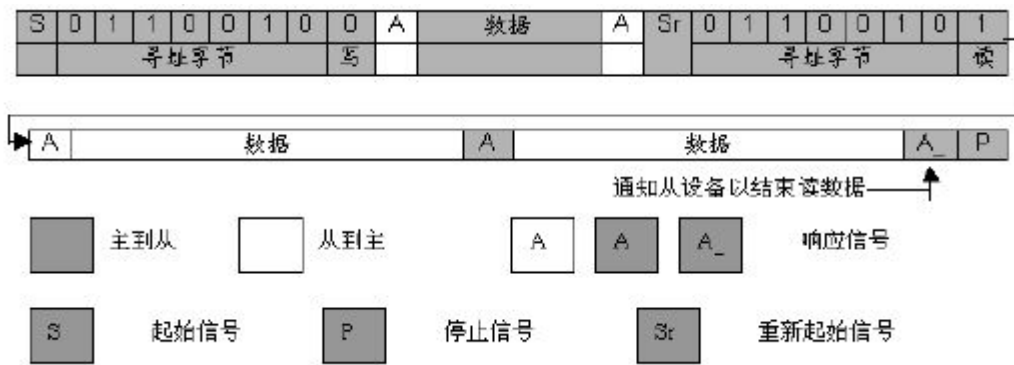
主设备向从设备写入数据过程图

S	0	1	1	0	0	1	0	0	A	数据	A	数据	A	P
寻址字节									写					

主设备向从设备直接读取数据过程图

S	0	1	1	0	0	1	0	1	A	数据	A	数据	A	P
寻址字节									读					

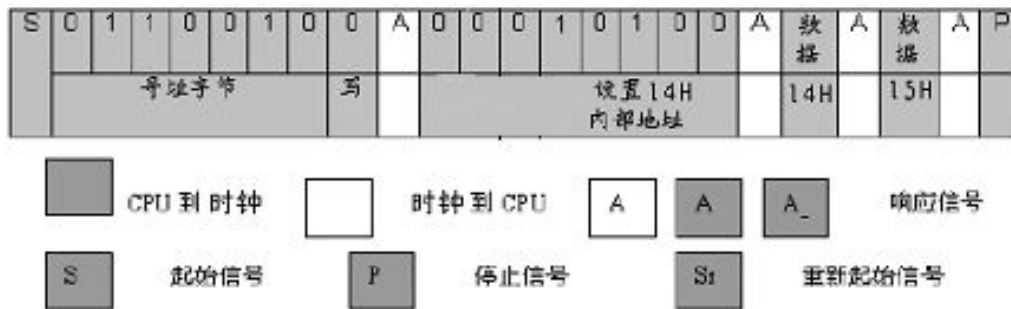
数据传输时改变其传输方向过程图



(3)SD2431API 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 SD2431API 的响应位(ACK),SD2431API 进入写状态;
- 接下来一个字节是确定 SD2431API 的内部地址(00H~3FH),第 9 位是 SD2431API 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.

SD2431API 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~3FH)的写操作必须确认芯片处于写允状态,否则写无效.

- 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。
- 从当前地址开始,每次读写完一个字节地址自动加 1 直到 3FH,读写完 3FH 后地址自动加 1 后为 00h.
- 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.
- 为了提高数据的可靠性,当写完成后,应将芯片置于写禁止状态。(参见 5.6)

(4)SD2431API 数据传输的读模式

SD2431API 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 SD2431API 的响应位 (ACK),SD2431API 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.

SD2431API 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 SD2431API 的响应位(ACK),SD2431API 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.

SD2431API 读数据方法 2 示例(从 00H 地址开始读取数据):



(5) SD2431API 在特殊条件下的数据传输

为了保证读写数据的有效性,SD2431API 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 SD2431API 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。

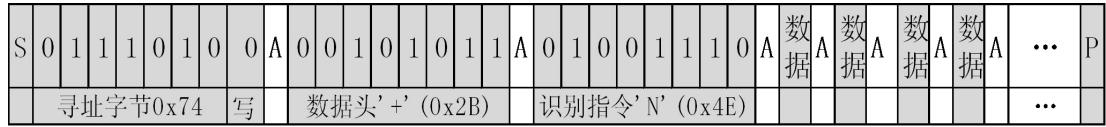
所以,要注意: 从开始信号进行读/写数据,直到停止信号,读/写操作过程必须在 0.5 秒之内完成。

6.4 WIFI 网络指令传输格式

6.4.1 网络连接配置指令:

在使用 WIFI 连接前必须配置需要连接的 SSID (无线接入点名称)与 密码,用户在不更改 SSID 与密码的情况下只需配置一次网络连接指令信息。配置指令如下:

IP 地址、连接服务器的端口号。在不更改网络传输配置的情况下，网络传输配置只需要配置一次。网络传输指令与网络连接指令类似：



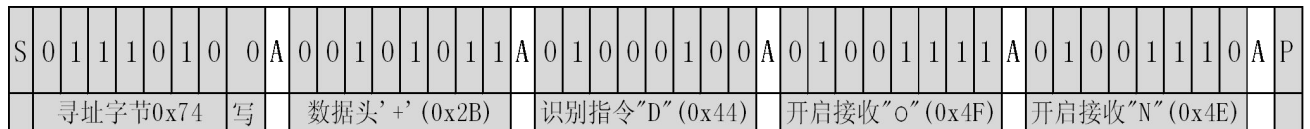
CPU 到芯片 芯片到 CPU S 起始信号 P 结束信号 A 响应信号

网络传输配置数据需要在识别指令之后传输，传输的格式是先传协议类型（UDP/TCP），再传输 IP 地址，然后传输端口号。只有网络传输配置的网络协议类型（UDP/TCP）与 IP 地址需要使用引号引起来，而端口号不需要使用引号。传输方式、IP 地址、端口号需要使用逗号隔开。例如（“TCP”，“192.168.0.1”，8080），即网络协议类型为：TCP 协议，IP 地址为：192.168.0.1，传输端口号为：8080。

6.4.5 网络传输开启/关闭指令

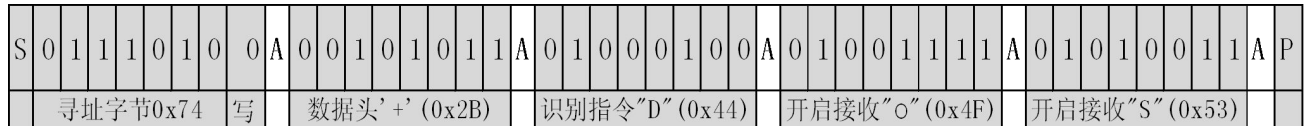
在使用网络数据传输前必须进行网络传输配置（具体参照 6.4.1）和网络连接配置（具体参照 6.4.4），如使用静态 IP 连接无线网络必须进行静态 IP 配置（具体参照 6.4.2）具体传输方式如下：

开启动态 IP 数据传输指令 I2C 时序：



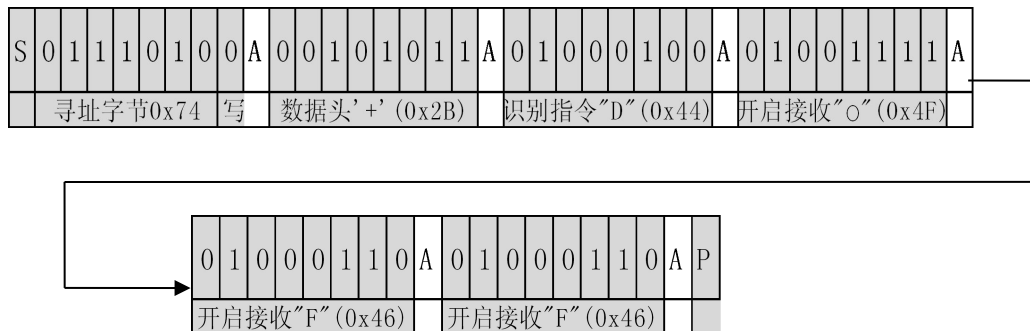
CPU 到芯片 芯片到 CPU S 起始信号 P 结束信号 A 响应信号

开启静态 IP 数据传输指令 I2C 时序：



CPU 到芯片 芯片到 CPU S 起始信号 P 结束信号 A 响应信号

关闭数据传输指令 I2C 时序：



用户在开启数据传输指令之前需要配置网络连接配置指令（6.4.1）和网络传输配置指令（6.4.4），如果使用静态 IP 传输前需要先配置静态 IP 配置指令，在传输网络数据过程中传输校时指令会自动关闭网络数据传输，如需开启需要重新开启数据传输指令，建议先进行网络校时之后，再传输数据。当网络传输完成后可以通过关闭数据传输指令来关闭无线网络，从而降低电源功耗。

6.4.6 网络数据传输指令

使用网络数据传输前必须先通过开启网络数据传输指令来启动网络连接，开启成功后才可配置网络数据传输。网络传输的 I2C 时序如下：



CPU 到芯片 芯片到 CPU S 起始信号 P 结束信号 A 响应信号

用户在识别指令 D 后传输需要传输的网络数据，传输的数据不超过 126 字节。当连接服务器出错将关闭无线传输，重新连接需要再次开启网络传输指令。

6.4.7 网络数据接收指令

用户可通过读取 DA_REC 引脚来判断是否接收到了数据，如有数据只需要从设备地址 0x75 中读取数据即可。I2C 时序如下：

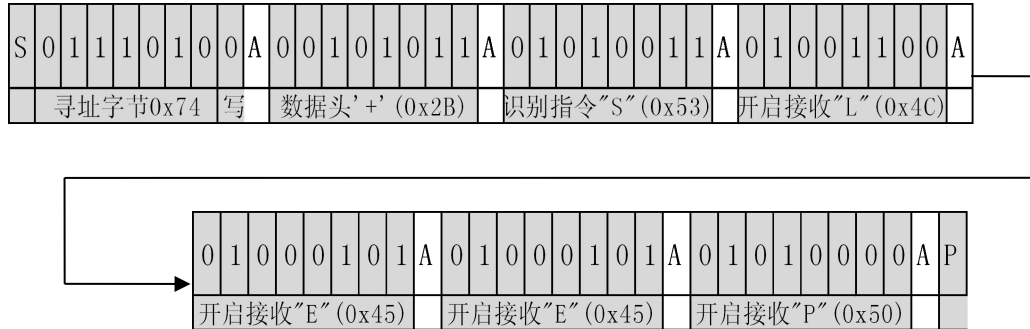


CPU 到芯片 芯片到 CPU S 起始信号 P 结束信号 A 响应信号

服务器传输的数据不能超过 22 字节，当用户读取完数据后，接收寄存器存储的数据将会被自动清除。

6.4.8 休眠与唤醒指令

在传输休眠指令后，SD2431 进入低功耗模式，在传输休眠指令会关闭网络数据传输，可在唤醒指令后重新开启网络数据传输指令。休眠指令的 I2C 时序如下：



唤醒指令的 I2C 时序如下：



6.4.9 网络配置指令

开启此指令后可以通过 TCP 网络协议对模块进行网络连接配置、网络传输配置和静态 IP 配置，具体配置方法参照 9.5.1。网络配置的 I2C 时序如下：

S	0	1	1	1	0	1	0	0	A	0	0	1	0	1	0	1	1	A	0	1	0	0	0	0	1	1	A	0	1	0	0	1	1	1	1	A	0	1	0	0	1	1	1	0	A	P
寻址字节 0x74									写	数据头 '+' (0x2B)									识别指令 "C" (0x43)									开启接收 "O" (0x4F)									开启接收 "N" (0x4E)									

CPU 到芯片 芯片到 CPU S 起始信号 P 结束信号 A 响应信号

完成本指令传输后可以通过电脑或手机连接到模块，可以通过 TCP 协议与模块进行网络配置（可以使用网络调试助手进行测试）。传输完此指令后需要在两分钟内完成无线网络数据配置，否则将释放 busy 线，WIFI 进入关闭状态。

6.5 TCP 网络配置指令

用户使用无线网络配置指令时，需连接模块的无线网络。再使用 TCP 协议连接到模块，模块的 IP 地址为：192.168.4.1，端口号为 1010。在进行下面的配置前需先通过 TCP 协议与模块进行连接。

6.5.1 TCP 网络连接配置

使用 TCP 配置网络连接，只需要将用户名与密码通过网络的形式传输到模块即可，传输完成返回 "OK"。具体的传输格式为：\$u" 用户名"，" 密码"。传输的字符用 \$u 作为开头，后面加上用户名与密码。用户名与密码需要加上双引号，中间使用逗号分隔开。例如：\$u" ABC"，" 12345678" 即用户名为：ABC，密码为：12345678。

6.5.2 TCP 网络传输配置

TCP 网络传输的配置与 TCP 连接的网络配置类似，只需传输特定的指令即可，完成配置后会返回 "OK"。TCP 网络传输格式：\$n" 网络协议 (TCP/UDP)"，" 网络连接的 IP 地址"，端口号。传输的字符使用 \$u 作为数据开头，后面加网络协议、IP 地址、端口号。网络协议与 IP 地址需要使用引号，中间使用逗号分隔。例如：\$n" TCP"，" 192.168.1.1"，8080。即传输的为 TCP 协议，网络连接的 IP 地址为：192.168.1.1，网络连接的端口号为：8080。

6.5.3 TCP 静态 IP 配置

TCP 静态 IP 配置只需在 \$s 后传输静态 IP 地址、默认网关、子网掩码即可，指令配置完成后返回 "OK"。TCP 静态 IP 配置格式：\$s" 静态 IP"，" 默认网关"，" 子网掩码"。例如 \$s" 192.168.0.100"，" 192.168.0.1"，" 255.255.255.0"，即无线静态 IP 地址为：192.168.0.100，网关为：192.168.0.1，子网掩码为：255.255.255.0。

6.5.4 关闭 TCP 网络配置

完成上述配置后需要关闭 TCP 网络配置，关闭 TCP 网络配置的指令为：\$c。完成关闭后会自动释放 busy 线，可进行校时和数据传输。

7. 备用电源切换电路

SD2431API 具有后备电池自动切换功能，当芯片检测到主电源 V_{DD} 掉到内部后备电池电压以下时自动转为由内部的后备电池供电，芯片进入后备电池供电模式。

在后备电池供电模式中实时时钟的所有功能均为正常。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：

SD2431API 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} 和后备电池) 之后使用户可以确定器件是否已上电。

8. 上电复位

芯片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.

9. 出厂设置

在出厂之前, 对 SD2431API 已做了时间和控制寄存器初始化设置:

时间为北京标准时间, 禁止 INT 脚输出信号, 数字调整寄存器初值。

实时寄存器: 当前的北京时间, xx (年), xx (月), xx (日), x (星期), xx (时), xx (分), x (秒)

10. 电源管理电路

1 内部电源管理电路(电源切换电路)

当 VDD>3.0V 时, 则内部电池停止供电, 改由外部电源 VDD 供电。在外部电源掉电情况下, 内部电池保证时钟继续可靠运行, 内置电池使用寿命为 5~8 年; 当电池能量耗尽后, 可通过 VBAT 引脚采用备份电源供电, 并可以通过 TEST 引脚测量电池电压。

芯片内部有充电电路, 只要 VDD 超过 3.0V, 充电电路就会开始工作, 可保证对充电电池进行连续可靠的充电。在电池满充的情况下可保证内部时钟走时超过半年时间(可满充 100 次), 累计电量超过 550mAh。

2. 稳压电路

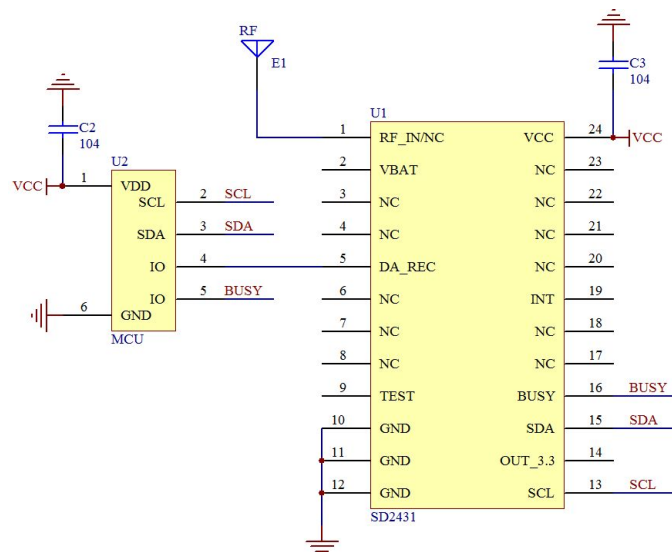
模块内部有稳压电路, 可以对外加电源电压进行滤波、稳压, 使模块工作始终工作在稳定状态。

11. 使用说明

1. 为了防止电路噪声问题, 请在此芯片的旁边放置 0.1uF 旁路电容。
2. 为了防止干扰, 在 PCB 制作时请保证芯片底部无大电流信号通过, 最好能铺地。

12. 应用电路与程序

(1) 使用外置天线的应用电路:

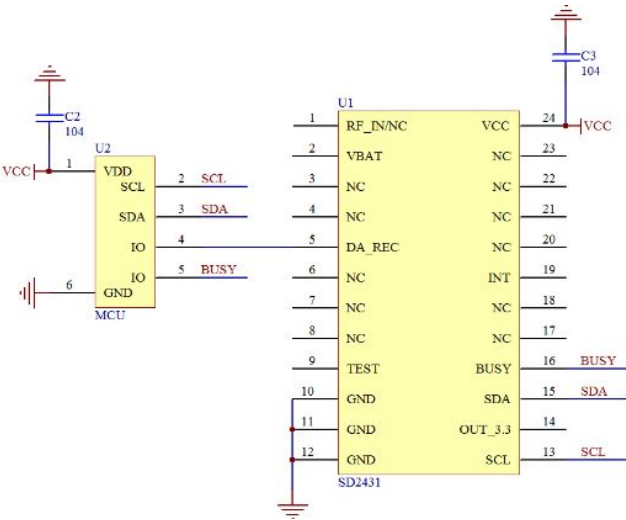


SD2431APEI (5V 版本) 和 SD2431APIEI (3.3V 版本) 的典型应用电路图

有源天线的要求: 阻抗: 50 欧; 驻波比: ≤ 3.0

频率: 2.4G-2.5G (2400M-2483.5M)

(2) 使用内置天线的 SD2431APEI 或 SD2431ALPEI 的应用电路：



备注：
SD2431 的 SDA, SCL 内部已经上拉，不需要外接上拉电阻；INT 内部无上拉，使用时需外接上拉电阻。
程序下载网址：<http://www.whwave.com.cn/download/sd2431/SD2431.asp>

13. 极限参数

V_{DD} 、SCL、SDA、SCLE、SDAE 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。

14. 时钟部分交直流特性

a) 直流特性(5V 版本)

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
V_{DD}	Main Power Supply		4.5	5.0	5.5	V	
V_{BAT}	Battery Supply Voltage		1.8		5.5	V	
I_{DD1}	Supply Current(except WIFI current and charge current)	$V_{DD}=5.0V$		7.0	10.0	μA	
I_{DD2}	Supply Current win IIC Active	$V_{DD}=5.0V$		50.0	120	μA	
I_{BAT}	Battery Supply Current	$V_{BAT}=3.0V$		800		nA	
I_{L1}	Input Leakage Current On SCL			100		nA	
I_{LO}	I/O Leakage Current On SDA			100		nA	
V_{BATHYS}	V_{BAT} Hysteresis		50	100	200	mV	
INT V_{OL}	Output Low Voltage	$V_{DD}=5V$ $I_{OL}=3mA$			0.4	V	
		$V_{DD}=5V$ $I_{OL}=3mA$			0.4	V	

直流特性(3.3V 版本)

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
V _{DD}	Main Power Supply		3.0	3.3	3.5	V	
V _{BAT}	Battery Supply Voltage		1.8		5.5	V	
I _{DD1}	Supply Current(except WIFI current and charge current)	V _{DD} =3.3V		6.0	9.0	μA	
I _{DD2}	Supply Current win IIC Active	V _{DD} =3.3V		50.0	120	μA	
I _{BAT}	Battery Supply Current	V _{BAT} =3.0V		800		nA	
I _{L1}	Input Leakage Current On SCL			100		nA	
I _{LO}	I/O Leakage Current On SDA			100		nA	
V _{BATHYS}	V _{BAT} Hysteresis		50	100	200	mV	
INT V _{OL}	Output Low Voltage	V _{DD} =3.3V I _{OL} =3mA			0.4	V	
		V _{DD} =3.3V I _{OL} =3mA			0.4	V	

b) 掉电时序温度=-40℃至+85℃

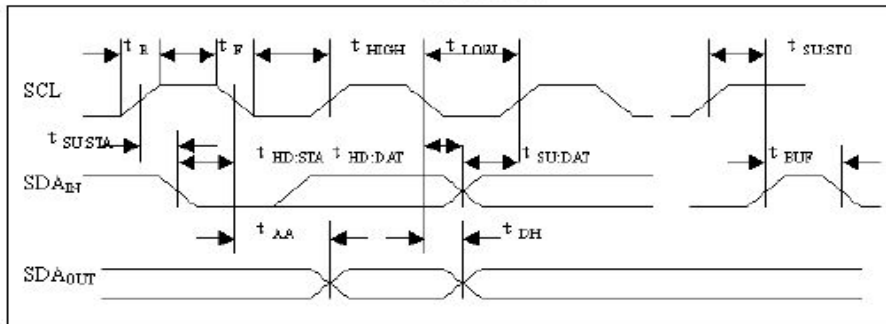
SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
V _{DD br}	V _{DD} negative Siewrate				10	V/ms	

c) 交流特性

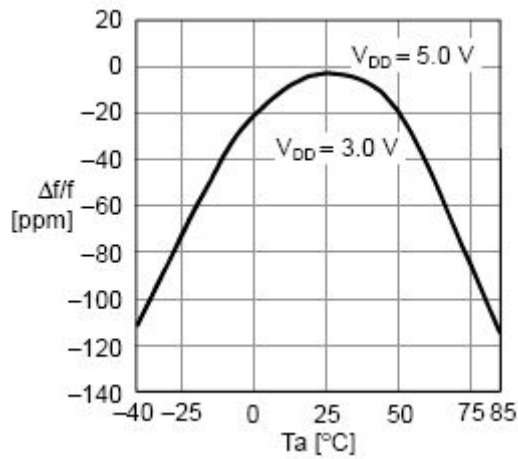
SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
V _{IL}	SDA and SCL input buffer LOW voltage		-0.3		0.3×V _{DD}	V	
V _{IH}	SDA and SCL input buffer HIGH voltage		0.7×V _{DD}		V _{DD} +0.3	V	
Hysteresis	SDA and SCL input buffer hysteresis		0.05×V _{DD}			V	
V _{OL}	SDA output buffer LOW voltage sinking 3mA		0		0.4	V	
C _{pin}	SDA and SCL pin capacitance	T _A =25℃ f=1MHZ V _{DD} =5V V _{IN} =0V V _{OUT} =0V			10	pF	
f _{SCL}	SCL frequency				400	KHZ	
t _{IN}	Pulse width suppression time at SDA and SCL inputs				50	ns	
t _{AA}	SCL falling edge to SDA output data valid	SCL falling edge crossing 30%of V _{DD} until SDA exits the 30%to 70%of V _{DD} window			900	ns	
t _{BUF}	Time the bus must be free before the start of a new transmission	SDA crossing 70%of V _{DD} during a STOP condition, to SDA crossing 70%of V _{DD} during the following START condition	1300			ns	
t _{LOW}	Clock LOW time	Measured at the 30% of V _{DD} crossing	1300			ns	
t _{HIGH}	Clock HIGH time	Measured at the 70% of V _{DD} crossing	600			ns	
t _{SU:STA}	START condition setup time	SCL rising edge to SDA falling edge Both crossing 70% of V _{DD}	600			ns	
t _{HD:STA}	START condition hold time	From SDA falling edge crossing 30% of V _{DD} to SCL falling edge crossing 70% of V _{DD}	600			ns	
t _{SU:DAT}	Input data setup time	From SDA exiting the 30% to 70% of V _{DD} window ,to SCL rising edge crossing 30% of V _{DD}	100			ns	

$t_{HD:DAT}$	Input data hold time	From SCL falling edge crossing 30% of V_{DD} to SDA entering the 30% to 70% of V_{DD} window	0		900	ns	
$t_{SU:STO}$	STOP condition setup time	From SCL rising edge crossing 70% of V_{DD} to SDA rising edge crossing 30% of V_{DD}	600			ns	
$t_{HD:STO}$	Output condition hold time	From SDA rising edge to SCL falling edge .Both crossing 70% of V_{DD}	600			ns	
t_{DH}	Output data hold time	From SCL falling edge crossing 30% of V_{DD} ,until SDA enters the 30% to 70% of V_{DD} window.	0			ns	
t_r	SDA and SCL rise time	From 30% to 70% of V_{DD}	20+ $0.1 \times C_b$		300	ns	
t_f	SDA and SCL fall time	From 70% to 30% of V_{DD}	20+ $0.1 \times C_b$		300	ns	
C_b	Capacitive loading of SDA or SCL	Total on-chip and off-chip	10		400	PF	
R_{PU}	SDA and SCL bus pull-up resistor off-chip	Maximum is determined by t_r and t_f For $C_b=400pF$,max is about 2~2.5k Ω For $C_b=40pF$,max is about 15~20k Ω	1			k Ω	

总线时序图



15.频率误差&温度关系曲线



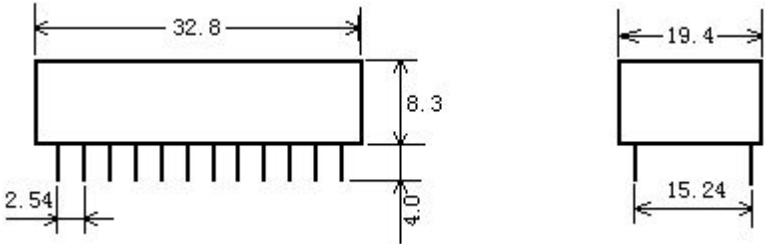
16. 芯片顶部字符说明



SD2431 型号说明

顶部型号字符	模块工作电压	WIFI 天线
SD2431API	5V	天线内置
SD2431APEI	5V	天线内置
SD2431ALPI	3.3V	天线外置
SD2431ALPEI	3.3V	天线外置

17. 封装尺寸（单位：毫米）



SD2431API 封装尺寸（管脚直径：0.5mm）

■ 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。

感谢您的支持及合作！

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。

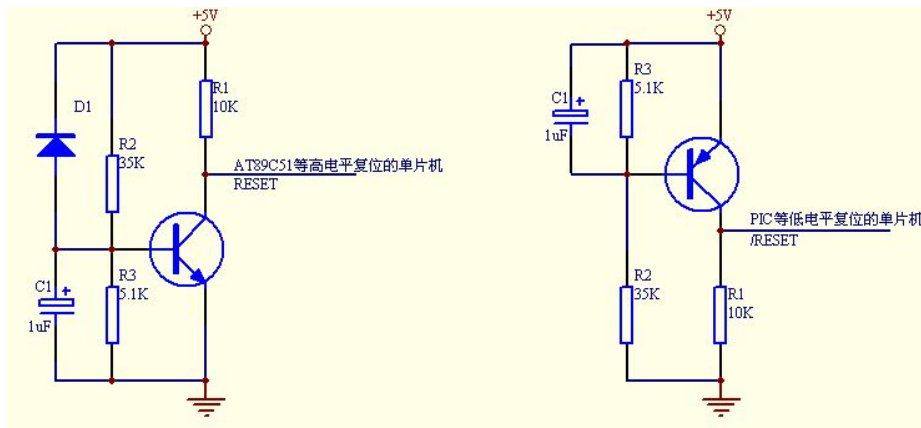
注：本文档受中国版权法保护, 非授权禁止拷贝、复制、引用或传播
SD 及 WAVE 为我公司注册商标.

深圳市兴威帆电子技术有限公司

附录：

一.SD2431API 应用中硬件注意事项：

- (1) 对 SD2431API 及 MCU 的电源~地之间加 104 电容去高频。
- (2) 对 SD2431API 及 MCU 所在的板数字电源~地的输入端加 22uF 以上的电解及 104 电容去除电源扰动。
- (3) 为了防止干扰, 在 PCB 制作时请保证芯片底部无大电流信号通过, 最好能铺地。
- (4) 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:

$$V_{be}/V_{rst}=R2/(R2+R3)$$

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。

- (5) SD2431API 不用的脚接地(VBAT.VOUT.INT 和 TEST 脚除外)。
- (6) 电源电压有 3.3V 和 5V 两个版本。
- (7) 模块天线脚至天线座采用 50ohm 特征阻抗的微带线, 微带线越短越好, 不要走蛇形线, 且微带线下的地尽量大, 微带线不要走线。
- (8) 远离强干扰源, 如 DSP、SDRAM、DC-DC、OLED、TFT 等。

二.SD2431API 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) 每次对时钟进行操作,应先判断 busy 的状态,否则可能会与内部校时电路发生总线竞争,引起时间数据错乱。
- (5) 写时间时,最好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 SD2431API 的一些问题回答

- (1) 时钟精度:

SD2431API 的精度指标为常温 25 度下 $\pm 5\text{ppm}$,即未用 WIFI 网络校时年误差少于 2.5 分钟.

出厂之前对每一只 SD2431API 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 SD2431API 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.

PC 版本: <http://www.whwave.com.cn/download/wave/AboutTime.rar>

安卓手机版: <http://www.whwave.com.cn/download/wavetime.rar>

- (2) 内部电池电压的测量:

利用 SD2431API 测试脚-TEST 脚可以测量内部电池的电压,方法是:

- a. 在 VDD 脚没加电时,拿万用表测得 TEST 脚电压,记为 V1
 - b. 给 SD2431API 的 VDD 脚外电源(通常 VDD 为 5.0v),拿万用表测得 TEST 脚电压,记为 V2
- 则内部电池电压 $V_{BAT}=VDD \cdot V1/V2$

注意:以上测 TEST 脚电压须用同一个万用表,因为不同的万用表内阻可能有差别,

而内部电池到 TEST 脚之间的电阻是 10M 欧姆,所以如用不同内阻的万用表会从 TEST 脚测得不同的电压.

通常内部电池电压 >1.2v 时都可以保证时钟内部的走时.

- (3) 充电电池的充电方式:

- a. 在出产前已经充满电,充电电池电压为 3.0V 左右
- b. 在用户板上电的情况下,SD2431API 就一直在充电→放电→充电的循环,其中放电主要是充电电池的自放电,通常一年时间电池的自放电电量有 2~5%,当电池电压降到 3.0V 时就开始充电即通常所说的浮充.注:这些过程都是自动完成的,您不用管它.
- c. 充电次数: 在整过电池寿命里,这一过程,如完全放完电再充满电,最多 100 次;
如果每次放电至电量的 10%再充满电,最多 1000 次;
如果每次放电至电量的 1%再充满电,最多 10000 次;.
- d. 在用户板下电的情况下,充电电池可维持时钟工作近半年时间,也就是说,您半年时间不上电,等再次上电时钟仍然在工作.
- e. 如 d 的情况,等电充满的时间通常要三天(72 小时).