



CS1381 串行时钟电路

产品说明书

说明书发行履历:

版本	发行时间	新制/修订内容
2010-01-A1	2010-01	更换新模板
2012-01-B1	2012-01	增加说明书编号及发行履历
2019-02-B2	2019-02	更换新模板
2019-06-B3	2019-06	添加订购信息
2022-01-B4	2022-01	修改订购信息



1、概述

CS1381是一款带秒、分、时、日、星期、月、年的串行时钟计时芯片，每个月日期以及闰年自动调节，CS1381具有低功耗工作方式并用若干寄存器存储对应信息，一个32.768KHz晶振校准时钟，为了使用最小管脚，CS1381使用一个I/O口与微信息处理机相连仅使用三根引线(1)RST 复位；(2) SCLK 串行时钟；(3) I/O 口数据就可以传送1字节或8字节的字符组。因而CS1381是一种性价比极高的时钟芯片，它广泛应用于电话、传真、便携式仪器以及电池供电的仪器仪表等产品领域。

其主要特点如下：

- 工作电压：2.0V~5.5V
- 最大输入串行时钟：2.0V 时 500KHz
5.0V 时 2MHz
- 工作电流：2.0V 时至少 400nA
5.0V 时至少 1μA
- 与 TTL 兼容
- 串行 I/O 口传送
- 两种数据传送方式：单字节传送
多字节传送（字符组方式）
- 所有寄存器都以 BCD 码格式存储
- 封装形式：SOP8

订购信息：

管装：

产品料号	封装形式	打印标识	管装数	盒装管	盒装数	备注说明
CS1381SA8.TB	SOP8 (1)	CS1381	100 PCS/管	100 管/盒	10000 PCS/盒	塑封体尺寸： 4.9mm×3.9mm 引脚间距：1.27mm
CS1381SA8.TB	SOP8 (2)	CS1381	100 PCS/管	200 管/盒	20000 PCS/盒	塑封体尺寸： 4.9mm×3.9mm 引脚间距：1.27mm

编带：

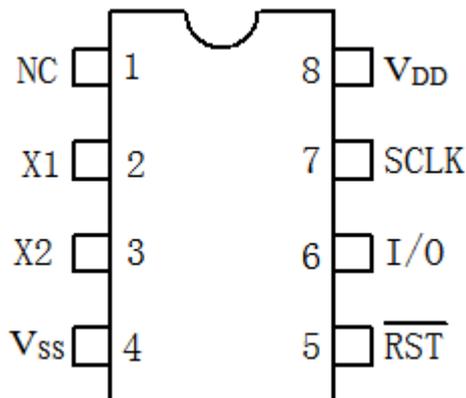
产品料号	封装形式	打印标识	编带盘装数	编带盒装数	备注说明
CS1381SA8.TR	SOP8	CS1381	4000PCS/盘	8000PCS/盒	塑封体尺寸： 4.9mm×3.9mm 引脚间距：1.27mm

注：如实物与订购信息不一致，请以实物为准。



2、引脚排列图及引脚说明

2.1、引脚排列图



2.2、引脚说明

管脚号	符号	输入/输出	说明
1	NC	—	空脚
2	X1	输入	振荡器输入
3	X2	输出	振荡器输出
4	V _{SS}	地	地
5	\overline{RST}	输入	串行传输的复位管脚
6	I/O	输入/输出	串行传输的数据输入/输出管脚
7	SCLK	输入	串行传输的串行时钟脉冲脚
8	V _{DD}	电源	正电源

3、电特性

3.1、极限参数

参数名称	符号	条件	额定值	单位
电源电压	V _{DD}	—	-0.3V~5.5V	V
输入电压	V _{IN}	—	V _{SS} -0.3V~V _{DD} +0.3V	V
工作环境温度	T _{amb}	—	0°C~70°C	°C
贮存温度	T _{stg}	—	-50°C~125°C	°C
焊接温度	T _L	10 秒	245	°C



3.2、电气特性

3.2.1、DC 特性

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{DD}	工作电压	—	—	2	—	5.5	V
I _{STB}	静态电流	2V	—	—	—	100	nA
		5V		—	—	100	nA
I _{DD}	工作电流	2V	无负载	—	0.7	1.0	uA
		5V		—	0.7	1.2	uA
I _{OH}	输出源电流	2V	V _{OH} =1.8V	-0.2	-0.4	—	mA
		5V	V _{OH} =4.5V	-0.5	-1.0	—	mA
I _{OL}	输出漏电流	2V	V _{OL} =0.2V	0.7	1.5	—	mA
		5V	V _{OL} =0.5V	2.0	4.0	—	mA
V _{IH}	高电平输入电压	5V	—	2	—	—	V
V _{IL}	低电平输入电压	5V	—	—	—	0.8	V
F _{OSC}	工作频率	5V	32.768KHz 晶振	—	32.768	—	KHz
F _{SCLK}	通信频率	2V	—	—	—	0.5	MHz
		5V		—	—	2	MHz

注: I_{STB}: 指 SCLK、I/O、RST 开路, 时钟停止位 CH 为逻辑 1, 即时钟停振。

3.2.2、AC 特性

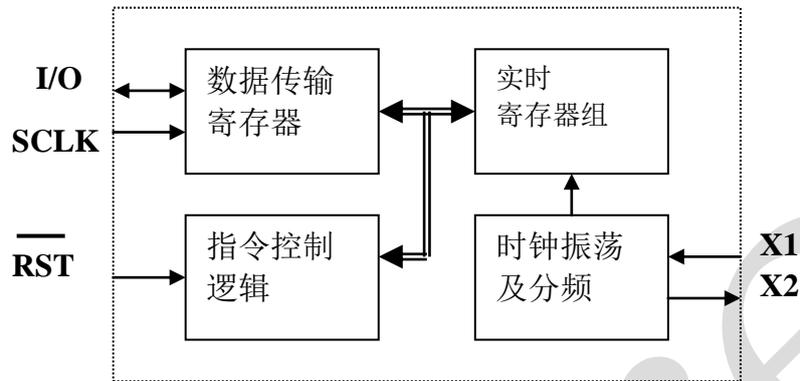
符号	参数	测试条件		最小	最大	单位
		V _{DD}	条件			
t _{DC}	数据建立时间	2V	—	200	—	ns
		5V	—	50	—	
t _{CDH}	数据保持时间	2V	—	280	—	ns
		5V	—	70	—	
t _{CDD}	数据延迟时间	2V	—	—	800	ns
		5V	—	—	200	
t _{CL}	数据低电平时间	2V	—	1000	—	ns
		5V	—	250	—	
t _{CH}	数据高电平时间	2V	—	1000	—	ns
		5V	—	250	—	
f _{CLK}	时钟频率	2V	—	—	0.5	MHz
		5V	—	—	2.0	
t _r	时钟上升和下降时间	2V	—	—	2000	ns
t _f		5V	—	—	500	
t _{CC}	复位到时钟的建立时间	2V	—	4	—	us
		5V	—	1	—	
t _{CCH}	时钟到复位的保持时间	2V	—	240	—	ns
		5V	—	60	—	
t _{CWH}	复位静止时间	2V	—	4	—	us
		5V	—	1	—	
t _{CDZ}	复位到 I/O 高阻抗时间	2V	—	—	280	ns
		5V	—	—	70	



4、典型应用线路与应用说明

工作原理

功能框图



功能描述

CS1381 芯片具有精确的计时功能，它包含四个内部单元模块。第一是振荡器及时钟分频模块，它连接一个同内部时钟同步的外部 32.768KHz 晶振；第二是实时寄存器组模块，该电路内部一共有 8 个寄存器组，可分别读写；用于提供实时时钟及日历信息；第三是指令及控制模块，它用来编译不同的输入指令及输出一个正确的串行控制信号；第四是数据转移寄存器，它用来把串行输入数据转换为内部并行数据，提供内部命令译码或内部时钟寄存器组数据，或把内部各寄存器并行数据转换为串行数据输出。CS1381 芯片还包含两个附加的指令，即时钟终止指令（CH）及写保护指令（WP）。时钟终止指令决定内部时钟信号的操作与否，写保护字节决定内部寄存器能否被写入，芯片在正确实现功能前，上述两字节首先被指定。

指令字节

CS1381 的指令用于对内部的 8 个实时时钟/日期寄存器进行写入设置及读出，其控制模式多种，即分为：单字节读/写模式、多字节读写模式、测试模式，命令格式如下：

	C7	C6	C5	C4	C3	C2	C1	C0
单字节模式	1	0	0	0	A2	A1	A0	R/ \bar{W}
多字节模式	1	0	1	1	1	1	1	R/ \bar{W}
测试模式	1	0	0	1	X	X	X	1



CS1381 内部寄存地址及数据模式

寄存器地址 A2~A0	特征	命令 地址	读写 控制	数据 (BCD)	寄存器定义							
					7	6	5	4	3	2	1	0
0	秒	80	写	00~59	CH	10 秒			秒			
		81	读									
1	分	82	写	00~59	0	10 分			分			
		83	读									
2	12 小时	84	写	01~12	12/24	0	AP	HR	时			
	24 小时	85	读	00~23		0	10	HR				
3	日期	86	写	01~31	0	0	10 日期		日期			
		87	读									
4	月份	88	写	01~12	0	0	0	10 月	月			
		89	读									
5	星期	8A	写	01~07	0	0	0	0	星期			
		8B	读									
6	年	8C	写	00~99	10 年			年				
		8D	读									
7	写保护	8E	写	00~80	WP	通常 0						
		8F	读									

CH: 时钟停止位

寄存器 2 的第 7 位: 12/24 小时标志

CH=0 振荡器工作允许

bit7=1, 12 小时模式

CH=1 振荡器停止

bit7=0, 24 小时模式

WP: 写保护位

寄存器 2 的第 5 位: AM/PM 定义

WP=0 寄存器数据能够写入

AP=1, 下午模式

CH=1 寄存器数据不能写入

AP=0, 上午模式

● 字节C0 (读/写模式)

指令字节的最低位字节定义为标志位, 它决定数据寄存器的状态。当字节C0被设定为“0”时, 进入写操作, 此时数据可以被写入到寄存器。相反, 芯片处于读操作, 此时, 数据仅仅能从寄存器中读出。

● C3-C1字节(A2-A0字节)

指令字节C1-C3用来指明进入哪一个寄存器。芯片总共有八个控制不同时间单元的寄存器, 如图中显示。在写数据到这些寄存器以前, 通过指令字节C0来指明进入写操作。

● 多字节模式

当指令字节被设定为BE(BF), 芯片进入多字节模式, 在这种模式下, 八个寄存器可以按次序的从寄存器0的0字节读(或写)。(参照多字节模式下时钟波形)

● 测试模式

当指令字节被设定为1001XXX1时, 芯片处于测试模式。测试模式仅仅使用于内部芯片供应者, 它不会推荐给使用于正常操作状态的用户。



- **写保护寄存器**

当设定该寄存器的字节7(WP)为逻辑1时, 另外7个寄存器将被保护防止数据无意写入。数据可以写入这些寄存器仅仅在字节WP被设定逻辑0时。在芯片重新工作或写新数据之前, 写保护寄存器的字节WP应首先被设为逻辑0。在读操作时它应设为逻辑1。另外, 在多字节模式期间, WP字节不能设定。

时钟终止字节

秒寄存器的字节D7被定义为时钟终止字节。当这个字节被设定为逻辑1, 内部时钟振荡器被终止并且芯片进入静态模式。当重新设D7为逻辑0时, 内部时钟振荡器将被再次唤醒。

- **12小时制/24小时制模式**

12小时制或24小时制模式的选择是通过小时寄存器的字节D7。当字节D7设为1时, 12小时制模式被选择。相反, 24小时制模式被选择。

- **上午/下午模式**

通过对小时寄存器的字节D7定义, 有两种功能可通过字节D5来选择。第一, 当12小时模式被设置时, 可使用上午—下午模式选择; 当字节D5设定为逻辑1时, 它处于下午模式, 相反处于上午模式。当处于24小时制模式时, 字节D5的另一个功能是用来表示第二个十小时字节(20-23); 当D5设为逻辑1时, 结合字节D4。20-24小时时间段能够计时。

- **复位(\overline{RST})及串行时钟控制(SCLK)**

复位脚用于清除内部寄存器的内容。为了初始化芯片, 它需要设置从低电平到高电平, 并且当芯片正常操作时保持逻辑1。除此之外, 复位脚应该在每一存取周期开始时由低电平触发为高电平, 并在存取周期结束时由高电平触发为低电平。复位脚也用于终止单字节或多字节操作。串行时钟控制(SCLK)脚是同步读或写的输入时钟管脚。在读操作时, 输出数据在指令操作完成后串行时钟的最后一个下降沿有效, 输入数据必须在写指令操作或写数据操作的第一个串行时钟上升沿处输入。读操作及写操作的波形将在后面说明。

- **I/O信号**

输入/输出脚是将数据从芯片读出或写入的双向端口。当数据写入芯片前, 指令字节 R/\overline{W} 应当设为逻辑0, 随后在第一个时钟信号输入时数据(指令字节或数据)能够被写入到芯片中。除此之外, 首先在字节0及接下的7个连续周期内数据(指令字节或数据)被写入到芯片中。额外的串行时钟周期将被忽略。同时, 在数据读出时指令字节的读/写字节被设为逻辑1, 在最后一个读指令字节写入芯片后, 字节0在串行时钟及接下7个连续周期的下降沿首先被读出, 除总共8个串行时钟读操作外, 只要串行时钟信号保持逻辑1, 其它的周期将重复地读数据。

- **晶振**

一个32768HZ的晶振可通过管脚2及管脚3(X1, X2)直接同CS1381芯片连接。为了获得精确的频率应当外接两平衡电容。

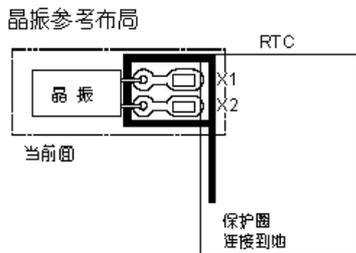


● 精确时钟

精确时钟取决于精确的晶振及在振荡器电路中的电容负载及平衡晶振的电容负载的最佳匹配。另外由于温度变化引起晶振频率的漂移将使得额外错误增加。耦合到振荡电路的外部电容噪声将导致时钟偏快。

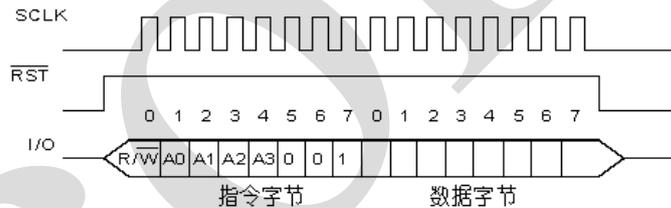
● 流程图表示法

为了初始化芯片,写保护字节(WP)及内部晶振工作字节(CH)在复位输入信号由低电平变为高电平时首先被释放。写保护字节的释放是通过写一个指令字节8E到芯片中,接下来数据位字节的最高有效位设为逻辑0。同时,CH字节的释放通过写指令字节80到芯片中,接下来数据位字节的最高有效位设为逻辑0。必须注意到在写保护字节同时钟终止字节之间复位信号的输入有一双重转换。换句话说,复位信号需要重新设置在两个连续存取阶段。写保护字节及时钟终止字节释放后,芯片准备通过外部主机(例如微处理器)发生存取。在单字节模式下,需要16个串行时钟信号脉冲存取数据,或者在多字节模式下需要72个串行时钟信号脉冲。

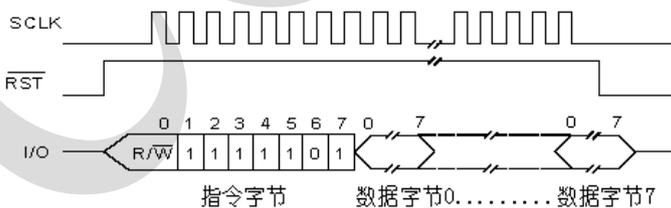


下图显示单字节及多字节传输

单字节传输



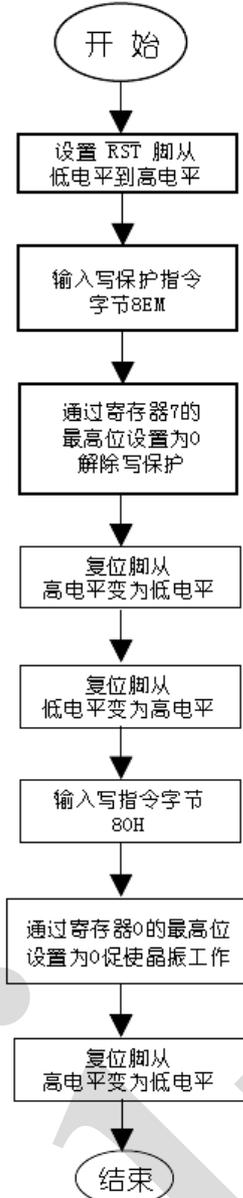
多字节传输



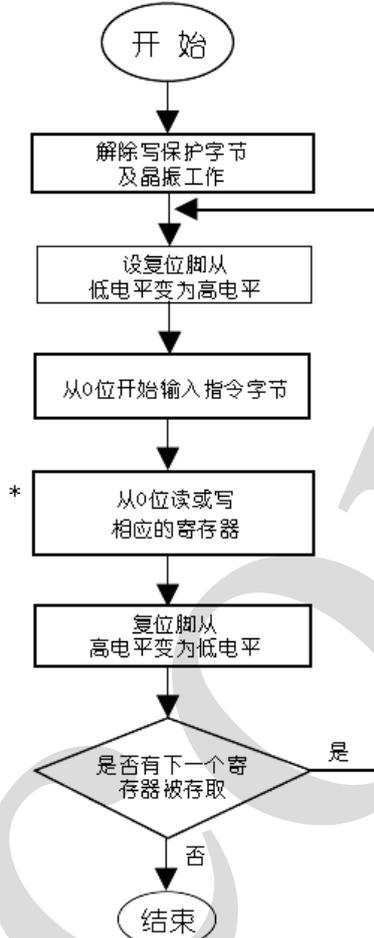


流程图

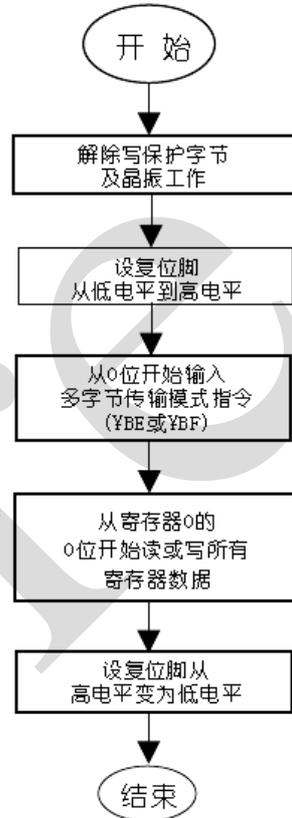
* 解除写保护字节(WP=0)及晶振工作(CH=0)



* 单字节数据传输



* 多字节数据传输

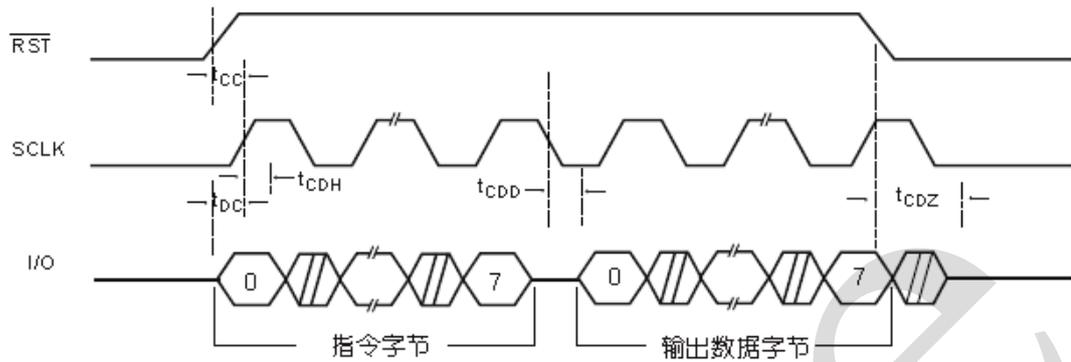


注：“*”为了从 CS1381 寄存器中读数据，最初数据位在指令字节最后位写入后的第一个下降沿时传送。

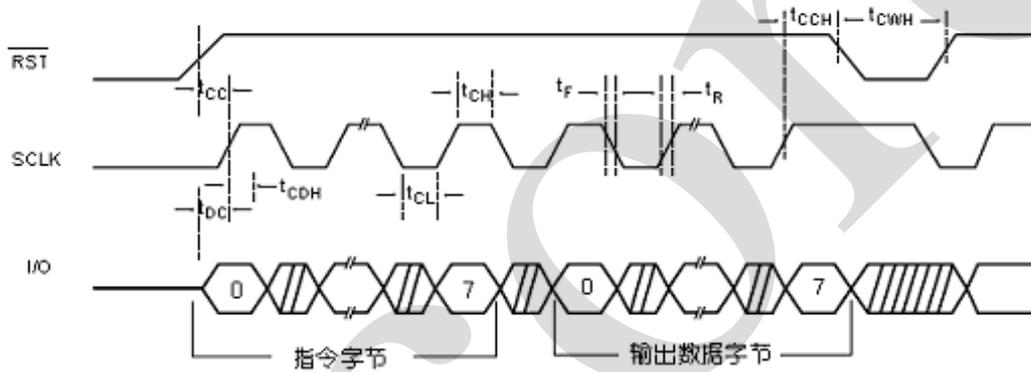


时序图

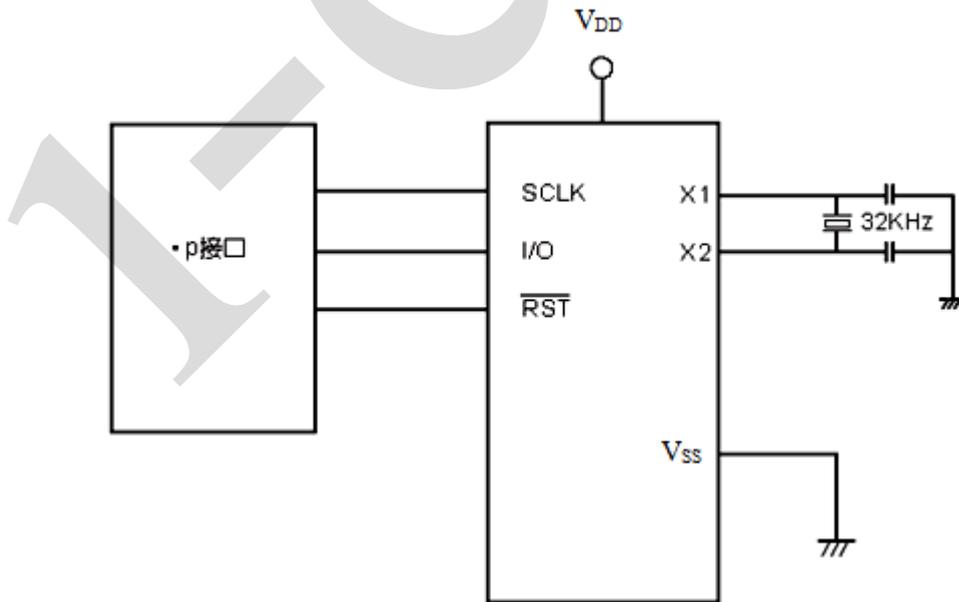
读数据传输



写数据传输



典型应用



**CS1381 与微控制器的接口软件及功能应用举例**

下面首先给出基本的接口软件，然后举例说明各种功能的应用。

● 写保护寄存器操作

当写保护寄存器的最高位为0时，允许数据写入寄存器，写保护寄存器可以通过命令字节8E、8F来规定禁止写入/读出。写保护位不能在多字节传送模式下写入。

Write_Enable:

```
MOV    Command,#8Eh          ; 命令字节为8E
MOV    ByteCnt,#1           ; 单字节传送模式
MOV    R0,#XmtDat          ; 数据地址覆给R0
MOV    XmtDat,#00h         ; 数据内容为0（写入允许）
ACALL  Send_Byte           ; 调用写入数据子程序
RET
```

当写保护寄存器的最高位为1时，禁止数据写入寄存器

Write_Disable:

```
MOV    Command,#8Eh        ; 命令字节为8E
MOV    ByteCnt,#1         ; 单字节传送模式
MOV    R0,#XmtDat         ; 数据地址覆给R0
MOV    XmtDat,#80h        ; 数据内容为80h（禁止写入）
ACALL  Send_Byte          ; 调用写入数据子程序
RET
```

以上程序调用了基本数据发送(Send_Byte)模块及一些内存单元定义，其源程序清单在附录中给出。下面的程序亦使用了这个模块。

● 时钟停止位操作

当把秒寄存器的第7位（时钟停止位）设置为0时，起动时钟开始。

Osc_Enable:

```
MOV    Command,#80h        ; 命令字节为80
MOV    ByteCnt,#1         ; 单字节传送模式
MOV    R0,#XmtDat         ; 数据地址覆给R0
MOV    XmtDat,#00h        ; 数据内容为0（振荡器工作允许）
ACALL  Send_Byte          ; 调用写入数据子程序
RET
```

当把秒寄存器的第7位时钟停止位设置为1时，时钟振荡器停止，CS1381进入低功耗方式，

Osc_Disable:

```
MOV    Command,#80h        ; 命令字节为80
MOV    ByteCnt,#1         ; 单字节传送模式
MOV    R0,#XmtDat         ; 数据地址覆给R0
MOV    XmtDat,#80h        ; 数据内容为80h（振荡器停止）
ACALL  Send_Byte          ; 调用写入数据子程序
RET
```



- **多字节传送方式**

当命令字节为BE或BF时，CS1381 工作在多字传送模式，8 个时钟/日历寄存器从寄存器0地址开始连续读写从0位开始的数据。

例如：写入00年、6月21日、星期三、13 时、59分、59秒、程序设置如下：

Write_Multiplebyte:

```

MOV      Command,#0Beh      ; 命令字节为BEh
MOV      ByteCnt,#8         ; 多字节写入模式（此模块为8个）
MOV      R0,#XmtDat         ; 数据地址覆给R0
MOV      XmtDat,#59h        ; 秒单元内容为59h
MOV      XmtDat+1,#59h      ; 分单元内容为59h
MOV      XmtDat+2,#13h      ; 时单元内容为13h
MOV      XmtDat+3,#21h      ; 日期单元内容为21h
MOV      XmtDat+4,#06h      ; 月单元内容为06h
MOV      XmtDat+5,#03h      ; 星期单元内容为03h
MOV      XmtDat+6,#0        ; 年单元内容为00h
MOV      XmtDat+7,#0        ; 写保护单元内容为00h
ACALL    Send_Byte          ; 调用写入数据子程序
RET

```

读出寄存器0-7的内容，程序设置如下：

Read_Multiplebyte:

```

MOV      Command,#0BFh      ; 命令字节为BFh
MOV      ByteCnt,#8         ; 多字节读出模式（此模块为8个）
MOV      R1,#RcvDat         ; 数据地址覆给R1
ACALL    Receive_Byte       ; 调用读出数据子程序
RET

```

以上程序调用了基本数据接收(Receive_Byte)模块及一些内存单元定义，其源程序清单在附录中给出。下面的程序亦使用了这个模块。

- **单字节传送方式**

例如：写入8 时(12小时模式)，程序设置如下：

Write_Singlebyte:

```

MOV      Command,#84h       ; 命令字节为84h
MOV      ByteCnt,#1         ; 单字节传送模式
MOV      R0,#XmtDat         ; 数据地址覆给R0
MOV      XmtDat,#88h        ; 数据内容为88h
ACALL    Send_Byte          ; 调用写入数据子程序
RET

```

上面所列出的程序模块“Write_Enable”、“Write_Disable”、“Osc_Enable”、“Osc_Disable”与单字节写入模块“Write_Singlebyte”的程序架构完全相同，仅只是几个入口参数不同，本文是为了强调功能使用的不同才将其分为不同模块，用户在使用中可灵活简略。



下面模块举例说明如何单字节读出“小时”单元的内容。.

Read_Singlebyte:

```
MOV      Command,#85h      ; 命令字节为85h
MOV      ByteCnt,#1        ; 单字节传送模式
MOV      R1,#RcvDat        ; 数据地址覆给R1
ACALL    Receive_Byte      ; 调用读出数据子程序
RET
```

附录：数据发送与接收模块源程序清单

; CPU工作频率最大不超过20MHz

; P87LPC762/4 主控器发送接受数据程序

; 说明：本程序是利用Philips公司的P87LPC764 单片机(任何具有51 内核或其它合适的单片机都可在此作为主控器)的普通I/O口(如P1.2/P1.3/P1.4)实现总线的功能,对总线上的器件(本程序采用CS1381)进行读写操作。命令字节在Command，传送字节数在ByteCnt中，所发送的数据在XmtDat中，所接收的数据在RcvDat中。

;P87LPC762/4 主控器总线发送接受数据程序头文件

;内存数据定义

```
BitCnt   DATA      30h      ; 数据位计数器
ByteCnt   DATA      31h      ; 数据字节计数器
Command   DATA      32h      ; 命令字节地址
RcvDat    DATA      40H      ; 接收数据缓冲区
XmtDat    DATA      50H      ; 发送数据缓冲区
```

;端口位定义

```
IO_DATA   Bit        P1.3      ; 数据传送总线
SCLK      Bit        P1.4      ; 时钟控制总线
RST       Bit        P1.2      ; 复位总线
```

;发送数据程序

;名称:Send_Byte

;描述:发送(ByteCnt)个字节给被控器CS1381

;命令字节地址在Command中

;所发送数据的字节数在ByteCnt中，发送的数据在XmtDat缓冲区中

Send_Byte:

```
CLR      RST              ; 复位管脚为低电平所有数据传送终止
NOP
CLR      SCLK             ; 清时钟总线
NOP
SETB     RST              ; 复位管脚为高电平逻辑控制有效
NOP
MOV      A,Command        ; 准备发送命令字节
MOV      BitCnt,#08h      ; 传送位数为8
```

S_Byte0:

```
RRC      A                ; 将最低位传送给进位位C
MOV      IO_DATA,C        ; 位传送至数据总线
```



```

NOP
SETB      SCLK          ; 时钟上升沿发送数据有效
NOP
CLR       SCLK          ; 清时钟总线
DJNZ     BitCnt,S_Byte0 ; 位传送未完毕则继续
NOP
S_Byte1:  ; 准备发送数据
MOV      A,@R0         ; 传送数据过程与传送命令相同
MOV      BitCnt,#08h

```

```

S_Byte2:
RRC      A
MOV      IO_DATA,C
NOP
SETB     SCLK
NOP
CLR      SCLK
DJNZ     BitCnt,S_Byte2
INC      R0             ; 发送数据的内存地址加1
DJNZ     ByteCnt,S_Byte1 ; 字节传送未完毕则继续
NOP
CLR      RST           ; 逻辑操作完毕清RST
RET

```

```

;*****
;接收数据程序;
;名称:Receive_Byte
;描述:从被控器CS1381接收(ByteCnt)个字节数据
;命令字节地址在Command中
;所接收数据的字节数在ByteCnt中,接收的数据在RcvDat缓冲区中
;*****

```

```

Receive_Byte:
CLR      RST           ; 复位管脚为低电平所有数据传送终止
NOP
CLR      SCLK          ; 清时钟总线
NOP
SETB     RST           ; 复位管脚为高电平逻辑控制有效
MOV      A,Command     ; 准备发送命令字节
MOV      BitCnt,#08h   ; 传送位数为8

```

```

R_Byte0:
RRC      A             ; 将最低位传送给进位位C
MOV      IO_DATA,C    ; 位传送至数据总线
NOP
SETB     SCLK          ; 时钟上升沿发送数据有效
NOP
CLR      SCLK          ; 清时钟总线
DJNZ     BitCnt,R_Byte0 ; 位传送未完毕则继续
NOP
R_Byte1:  ; 准备接收数据
CLR      A             ; 清类加器

```

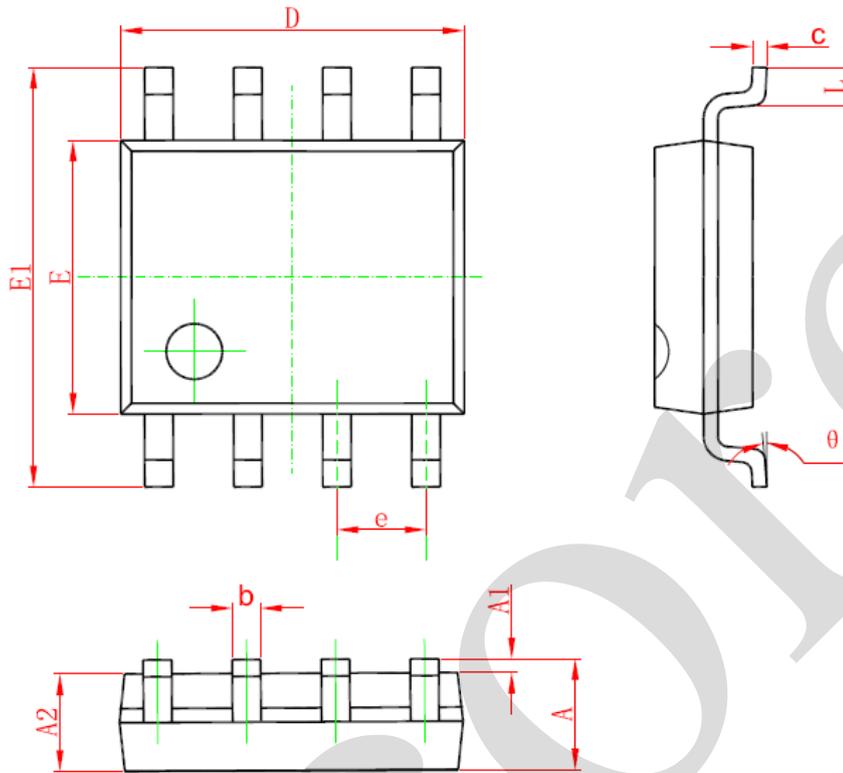


CLR	C	; 清进位位C
MOV	BitCnt,#08h	; 接收位数为8
R_Byte2:		
NOP		
MOV	C,IO_DATA	; 数据总线上的数据传送给C
RRC	A	; 从最低位接收数据
SETB	SCLK	; 时钟总线置高
NOP		
CLR	SCLK	; 时钟下降沿接收数据有效
DJNZ	BitCnt,R_Byte2	; 位接收未完毕则继续
MOV	@R1,A	; 接收到的完整数据字节放入接收内存缓冲区
INC	R1	; 接收数据的内存地址加1
DJNZ	ByteCnt,R_Byte1	; 字节接收未完毕则继续
NOP		
CLR	RST	; 逻辑操作完毕清RST
RET		
END		



5、封装尺寸与外形图

5.1、SOP8 外形图与封装尺寸



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	1.350	1.750	0.053	0.069
A1	0.100	0.250	0.004	0.010
A2	1.350	1.550	0.053	0.061
b	0.330	0.510	0.013	0.020
c	0.170	0.250	0.006	0.010
D	4.700	5.100	0.185	0.200
E	3.800	4.000	0.150	0.157
E1	5.800	6.200	0.228	0.244
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

**6、声明及注意事项:****6.1、产品中有毒有害物质或元素的名称及含量**

部件名称	有毒有害物质或元素									
	铅 (Pb)	汞 (Hg)	镉 (Cd)	六价铬 (Cr (VI))	多溴联苯 (PBBs)	多溴联苯醚 (PBDEs)	邻苯二甲酸二丁酯 (DBP)	邻苯二甲酸丁苄酯 (BBP)	邻苯二甲酸二(2-乙基己基)酯 (DEHP)	邻苯二甲酸二异丁酯 (DIBP)
引线框	○	○	○	○	○	○	○	○	○	○
塑封树脂	○	○	○	○	○	○	○	○	○	○
芯片	○	○	○	○	○	○	○	○	○	○
内引线	○	○	○	○	○	○	○	○	○	○
装片胶	○	○	○	○	○	○	○	○	○	○
说明	○: 表示该有毒有害物质或元素的含量在 SJ/T11363-2006 标准的检出限以下。 ×: 表示该有毒有害物质或元素的含量超出 SJ/T11363-2006 标准的限量要求。									

6.2、注意

在使用本产品之前建议仔细阅读本资料;

本资料中的信息如有变化, 恕不另行通知;

本资料仅供参考, 本公司不承担任何由此而引起的任何损失;

本公司也不承担任何在使用过程中引起的侵犯第三方专利或其它权利的责任。