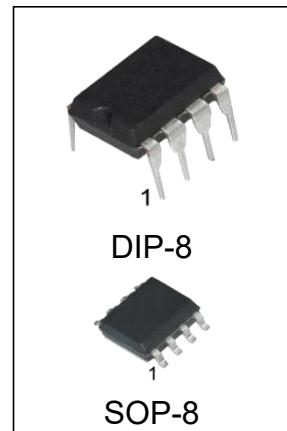


串行时钟电路

特点

- 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: SOP-8、DIP-8



产品订购信息

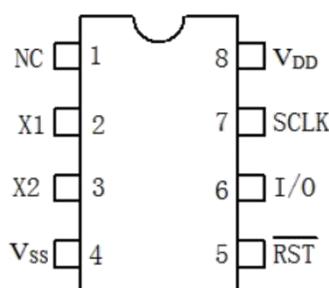
产品名称	封装	打印名称	包装	包装数量
HT1381PG	DIP-8	HT1381	管装	2000 只/盒
HT1381DRG	SOP-8	HT1381	编带	2500 只/盘

概述

HT1381 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，HT1381 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，HT1381 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 HT1381 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。

引脚排列图及引脚说明

引脚排列图



DIP-8/SOP-8

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	$\overline{\text{RST}}$	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

电特性

极限参数

参数名称	符号	条件	额定值	单位
电源电压	V_{DD}	—	$-0.3V \sim 5.5V$	V
输入电压	V_{IN}	—	$V_{SS}-0.3V \sim V_{DD}+0.3V$	V
工作环境温度	T_{amb}	—	$0^{\circ}C \sim 70^{\circ}C$	$^{\circ}C$
贮存温度	T_{stg}	—	$-50^{\circ}C \sim 125^{\circ}C$	$^{\circ}C$
焊接温度	T_L	10 秒	260	$^{\circ}C$

说明：绝对最大额定值是指超出该工作范围，器件有可能被损坏。长期工作于绝对最大额定值条件下，会影响器件的可靠性。绝对最大额定值仅是应力规格值。

电气特性

DC 特性

符号	参数	测试条件		最小	典型	最大	单位
		VDD	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	5.0	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、 \bar{RST} 开路，时钟停止位CH为跳高1，即时钟停振。

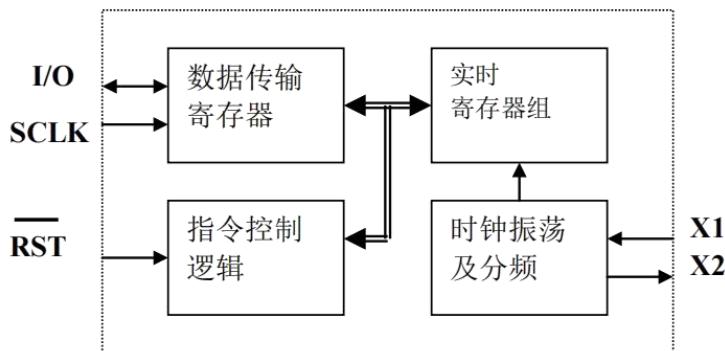
AC 特性

符号	参数	测试条件		最小	最大	单位
		VDD	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
		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	

典型应用线路与应用说明

工作原理

功能框图



功能描述

HT1381 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。HT1381 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

HT1381 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

HT1381 内部寄存地址及数据模式

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，如图表中显示。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(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 / 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 HT1381 芯片连接。为了获得精确的频率应当外接两平衡电容。

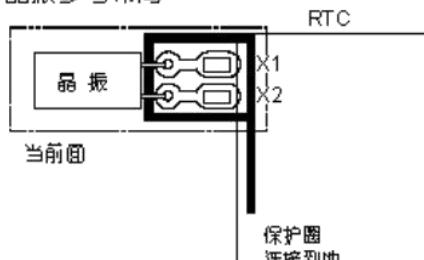
精确时钟

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

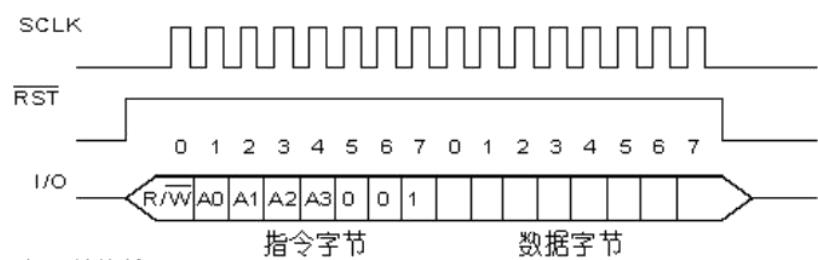
为了初始化芯片，写保护字节（WP）及内部晶振工作字节（CH）在复位输入信号由低电平变为高电平时首先被释放。写保护字节的释放是通过写一个指令字节 8E 到芯片中，接下来数据位字节的最高有效位设为逻辑 0。同时，CH 字节的释放通过写指令字节 80 到芯片中，接下来数据位字节的最高有效位设为逻辑 0。必须注意到在写保护字节同时钟终止字节之间复位信号的输入有一双重转换。换句话说，复位信号需要重新设置在两个连续存取阶段。写保护字节及时钟终止字节释放后，芯片准备通过外部主机（例如微处理器）发生存取。在单字节模式下，需要 16 个串行时钟信号脉冲存取数据，或者在多字节模式下需要 72 个串行时钟信号脉冲。

晶振参考布局

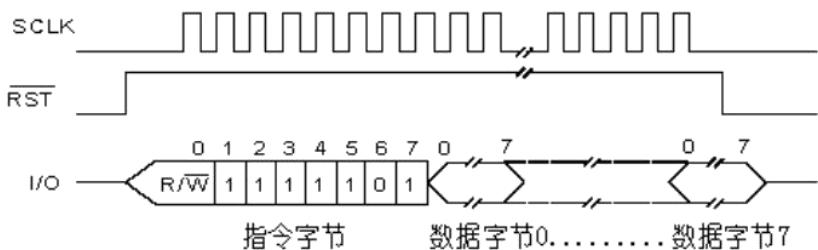


下图显示单字节及多字节传输

单字节传输

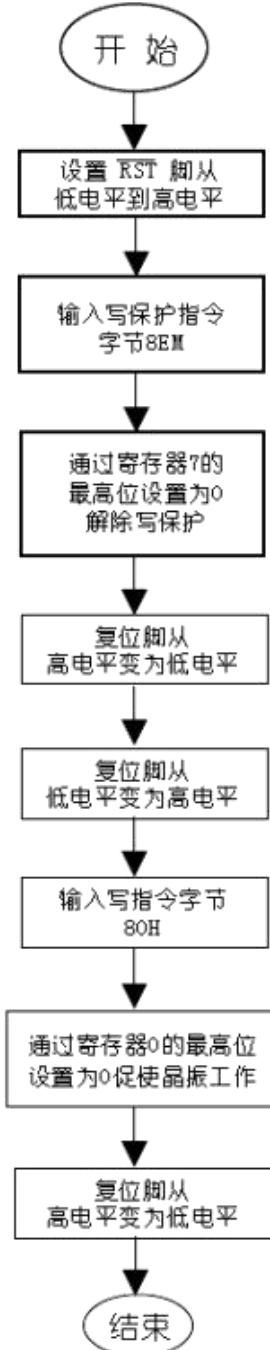


多字节传输

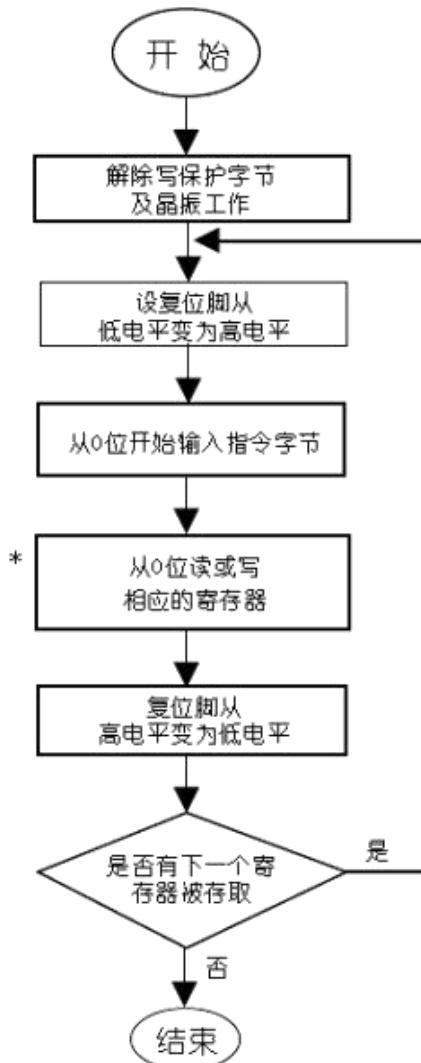


流程图

* 解除写保护字节(WP=0)
及晶振工作(CH=0)



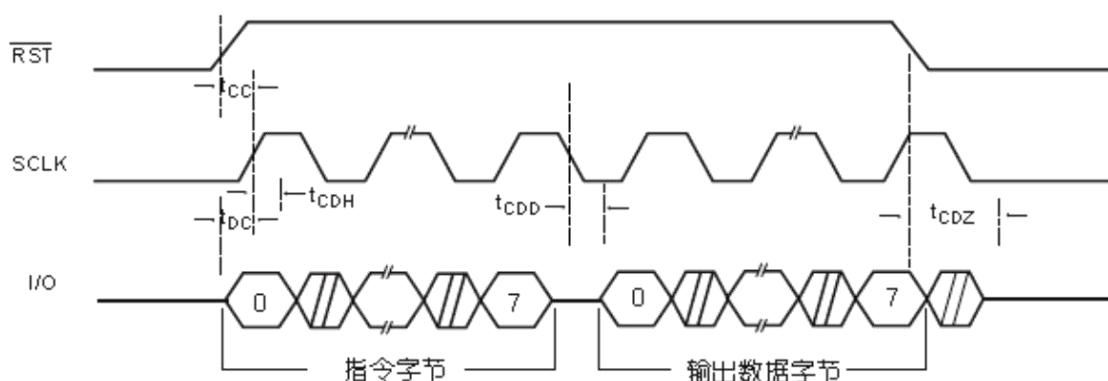
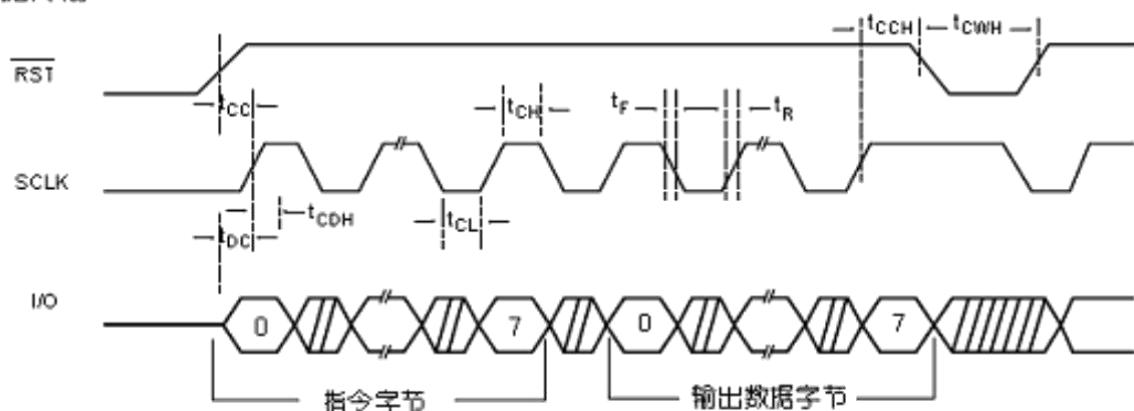
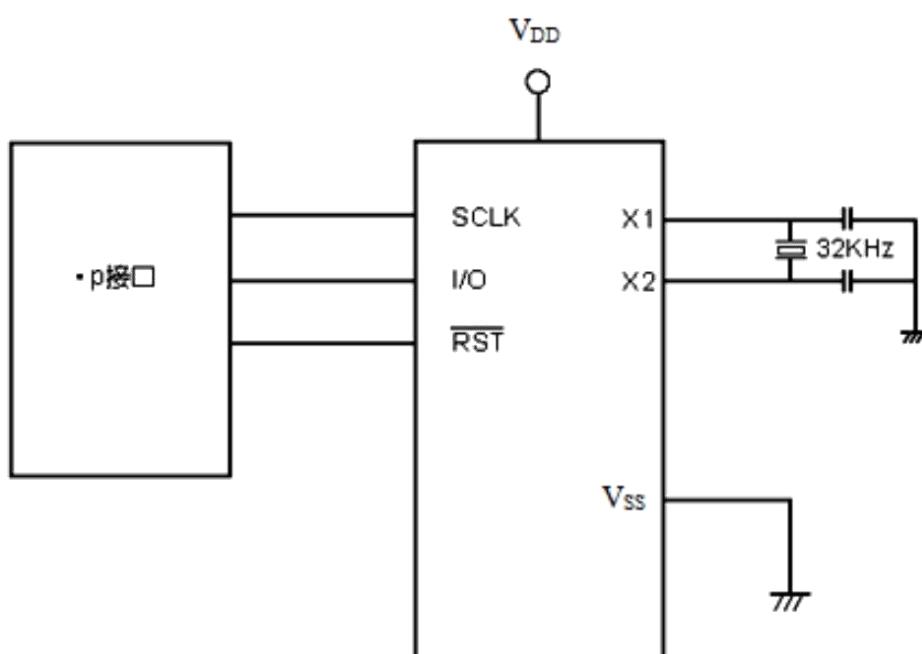
* 单字节数据传输



* 多字节数据传输



注 *为了从 HT1381 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


HT1381 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，HT1381 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, HT1381 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_Byt       ; 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 HT1381)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_Byt

描述:发送(ByteCnt)个字节给被控器 HT1381

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_Byt:

```
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 HT1381 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
```

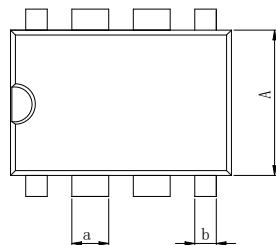
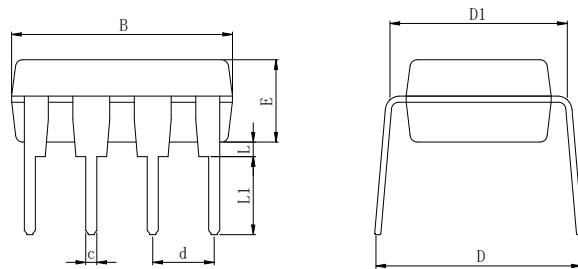
```
R_Byte1:          ; 准备接收数据
CLR          A          ; 清累加器
CLR          C          ; 清进位位 C
MOV          BitCnt,#08h ; 接收位数为 8
```

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
```

封装外形尺寸

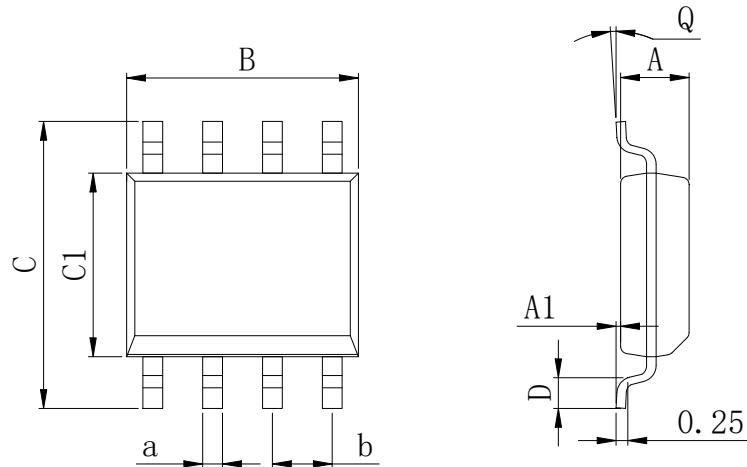
DIP-8



Dimensions In Millimeters(DIP-8)

Symbol:	A	B	D	D1	E	L	L1	a	b	c	d
Min:	6.10	9.00	8.10	7.42	3.10	0.50	3.00	1.50	0.85	0.40	2.54 BSC
Max:	6.68	9.50	10.9	7.82	3.55	0.70	3.60	1.55	0.90	0.50	

SOP-8 (150mil)



Dimensions In Millimeters(SOP-8)

Symbol:	A	A1	B	C	C1	D	Q	a	b
Min:	1.35	0.05	4.90	5.80	3.80	0.40	0°	0.35	1.27 BSC
Max:	1.55	0.20	5.10	6.20	4.00	0.80	8°	0.45	

修订历史

版本编号	日期	修改内容	页码
V1.0	2015-6	新修订	1-14
V1.1	2024-12	文档重新格式化, 增加封装尺寸图	1-18

重要声明：

汉芯半导体保留未经通知更改所提供的产品和服务。客户在订货前应获取最新的相关信息，并核实这些信息是否最新且完整的。汉芯半导体对篡改过的文件不承担任何责任或义务。

客户在使用汉芯半导体产品进行系统设计和整机制造时有责任遵守安全标准并采取安全措施。您将自行承担以下全部责任：针对您的应用选择合适的汉芯半导体产品；设计、验证并测试您的应用；确保您的应用满足相应标准以及任何其他安全、安保或其他要求。以避免潜在风险可能导致人身伤害或财产损失情况的发生。

汉芯半导体产品未获得生命支持、军事、航空航天等领域应用之许可，汉芯半导体将不承担产品在这些领域应用造成的后果。因使用方超出该产品适用领域使用所产生的一切问题和责任、损失由使用方自行承担，与汉芯半导体无关，使用方不得以本协议条款向汉芯半导体主张任何赔偿责任。

汉芯半导体所生产半导体产品的性能提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，测试和其他质量控制技术的使用只限于汉芯半导体的质量保证范围内。每个器件并非所有参数均需要检测。

汉芯半导体的文档资料，授权您仅可将这些资源用于研发本资料所述的产品的应用。您无权使用任何其他汉芯半导体知识产权或任何第三方知识产权。严禁对这些资源进行其他复制或展示，您应全额赔偿因在这些资源的使用中对汉芯半导体及其代理造成的任何索赔、损害、成本、损失和债务，汉芯半导体对此概不负责。