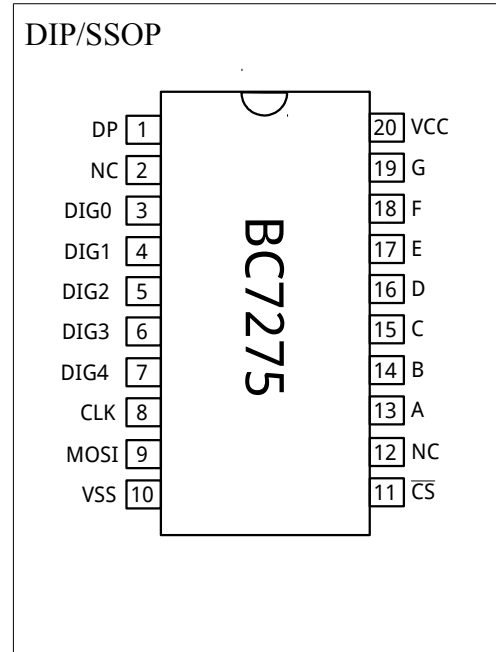


BC7275

5 位 LED 数码管驱动芯片 (第一版)

特点：

- SSOP20 小体积封装，无需外围器件
- 可驱动 5 位共阴式数码管或 40 只独立 LED
- 可实现任意单独 LED 闪烁，闪烁速度可调
- 译码显示时小数点显示不受显示更新影响
- 64Kbps 标准 SPI 串口，可用 2 线或 3 线方式，可使用单片机上硬件 SPI 接口，通讯不占用处理器时间
- 2 种显示译码方式，可单独控制任意 LED
- 可直接访问显示寄存器（显示特殊字符）
- 与其它 BC727X 系列芯片软件兼容，软件无需修改，即可用于其它 BC727X 芯片



摘要

BC7275 具有 5 位数码管显示管理功能。因为支持段寻址，可以独立地控制每一个显示段，故也可用于最多 40 个独立的 LED 驱动。BC7275 支持闪烁显示功能，且闪烁速度可调，可以独立控制每一段的闪烁，在使用独立的 LED 指示灯时，非常有用。

BC7275 内部提供译码功能，用户可以直接向译码寄存器写入数值，而得到相应数字显示。译码显示时，该位的小数点显示不受影响，用户只需更新显示数据，而无需考虑小数点的问题，尤其对于将小数点用作单独指示灯的用户，使用非常简便。同时，也支持直接写入显示寄存器，可以完成一些特殊字符的显示。

BC7275 采用串行接口，可以直接与标准 SPI 接口连接，通讯速率可达 64Kbps，用户可以充分利用微处理器上硬件 SPI 接口资源，当使用中断方式时，可使显示部分的通讯几乎不占用主程序时间。BC7275 的 SPI 接口可以接为 2 线或三线方式。三线方式时，可以通过 CS 片选信号，在一个 SPI 总线上使用多个器件，而在 MCU I/O 口资源紧张时，片选 CS 线可以直接接地，其内部独特的 SPI 口计时复位逻辑可以使得即便没有片选信号的接口清零功能，也可以保障通讯不会出错。

极限参数：（注：超出所列范围有可能造成器件永久损坏）

储存温度	-65 至+150°C
工作温度	-40 至+85°C
任意脚对地电压	-0.5 至 6.0V

电特性：（除特别说明外， $T_A=25^{\circ}\text{C}$ ， $V_{CC}=5.0\text{V}$ ）

参数	最小值	典型值	最大值	单位	备注
电源电压	2.7	5.0	5.5	V	
工作电流		4.9		mA	无 LED 负载时
输入低电平			1.4	V	
输入高电平	3.7			V	当 $V_{CC}=3\text{V}$ 时，为 1.9V
输出低电平			0.1	V	
输出高电平	4.4			V	
显示扫描周期		15		mS	

引脚说明：

名称	序号	说明
DP	1	小数点段驱动
DIG0-DIG4	3,4,5,6,7	位驱动，分别接数码管的共阳端
CLK	8	SPI 串行输入时钟端，最大时钟频率 64KHz，空闲时高电平
MOSI	9	SPI 口数据输入端，接 MCU 的 SPI 口数据输出端
GND	10	接地端
$\overline{\text{CS}}$	11	片选端，低电平有效
A-G	13,14,15,16, 17,18,19	A 段-G 段段驱动
VCC	20	正电源端，电压范围 2.7-5.5V

内部寄存器

BC7275 内部具有 18 个寄存器，包括 5 个显示寄存器，以及 13 个特殊寄存器。地址范围为 00H-1DH，其中 00H-0FH 为显示寄存器，其余为特殊寄存器。

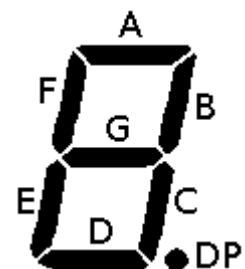
地址	内容	缺省值	说明
00H	第 0 位显示寄存器	FFH	显示寄存器每一位对应 1 个显示段
01H	第 1 位显示寄存器	FFH	

地址	内容	缺省值	说明
02H	第2位显示寄存器	FFH	
03H	第3位显示寄存器	FFH	
04H	第4位显示寄存器	FFH	
...	-(无效寄存器)	...	对BC7275不可用
10H	第0位段闪烁控制寄存器	00H	每一位对应一个显示段，1=闪烁，0=不闪烁
11H	第1位段闪烁控制寄存器	00H	
12H	第2位段闪烁控制寄存器	00H	
13H	第3位段闪烁控制寄存器	00H	
14H	第4位段闪烁控制寄存器	00H	
...	-(无效寄存器)	...	对BC7275不可用
18H	位闪烁控制寄存器	00H	bit0-bit4 分别对应显示位0-4，1=闪烁,0=不闪烁
19H	-(无效寄存器)	...	对BC7275不可用
1AH	闪烁速度控制寄存器	10H	值越小，闪烁速度越快
1BH	译码寄存器	-	写入该寄存器的数据被译码后更新显示寄存器
1CH	段寻址寄存器	-	写入该寄存器可单独控制各显示段状态
1DH	群操作寄存器	FFH	写入该寄存器的值，将被同时写入到所有的显示寄存器，可用于清除显示等操作

显示寄存器：地址 **00H-04H**

显示寄存器直接映射各个LED显示段，数码管上的显示段与各位的映射关系为如图：

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
DP	G	F	E	D	C	B	A



用户可以直接改变显示寄存器的内容，从而改变显示，这主要用在需要显示译码表中没有的特殊字符的时候。

显示寄存器中，如某一位被置0，则该显示段被点亮。复位后，所有显示寄存器的内容被置为FFH。

段闪烁控制寄存器：地址 **10H-14H**

和显示寄存器类似，段闪烁控制寄存器也采用位映射的方式控制显示段的闪烁，每一个

位对应一个显示段，影射的方法同显示寄存器。

当段闪烁控制寄存器中相应的位为**1**时，对应的显示段具有闪烁属性。闪烁仅发生在该显示段被点亮的情况下，如该显示段未点亮（相应显示寄存器位为**1**），则该显示段没有任何显示。当闪烁的显示段被清除（相应显示寄存器位被置**1**）时，其闪烁属性并不受影响，当该显示寄存器位再次被置为**0**时，对应的显示段将依然为闪烁显示。

复位时，段闪烁控制寄存器被全部清零（不闪烁）。

位闪烁控制寄存器：地址 **18H**

位闪烁控制寄存器控制显示位的整体闪烁属性，寄存器内每一个位对应一个显示位，对应关系如下表

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
-	-	-	位 4	位 3	位 2	位 1	位 0

显示位数据为**1**时，该位为闪烁显示。闪烁仅发生在该显示位有显示内容的情况下，如该显示位没有显示（显示寄存器所有位为**1**），则不会有任何显示。复位时，位闪烁控制寄存器被置为**00H**（不闪烁）。

闪烁速度控制寄存器：地址 **1AH**

BC7275的闪烁速度可调，只需要通过改变闪烁速度控制寄存器，就可以方便地控制闪烁的速度。闪烁速度控制寄存器中的值越大，闪烁速度越慢，相反值越小闪烁速度越快。在复位后，该寄存器的值为**10H**，在这个值下面，其闪烁频率大约为**2Hz**。

译码寄存器：地址 **1BH**

通过译码寄存器，用户可以通过送入数值，直接得到数字显示，省去了用户自己编制解码表的烦恼。写入译码寄存器的数据格式如下：

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
A ₃	A ₂	A ₁	A ₀	d ₃	d ₂	d ₁	d ₀

A₃:A₀为位地址，决定数字显示的位置，即显示位，为**0000**时，译码结果显示在第**0**位，**0011**时，则显示在第**3**位上。该地址有效范围为**0-4**，如地址超出该范围，则显示不会有任何改变。d₃:d₀为待显示的数值。译码表如下：

d ₃	d ₂	d ₁	d ₀	d ₃ :d ₀ (16进制值)	显示
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3

d ₃	d ₂	d ₁	d ₀	d ₃ :d ₀ (16进制值)	显示
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	A	A
1	0	1	1	B	b
1	1	0	0	C	C
1	1	0	1	D	d
1	1	1	0	E	E
1	1	1	1	F	F

通过译码寄存器更新显示内容，将不影响小数点位，即 **DP** 位的状态将保持不变。通过这种设计，当显示有小数点的数据，或在将小数点用作其它指示灯的情况下，数据的更新将非常方便，因为可以不用考虑重新刷新小数点显示。小数点的控制，可以通过段寻址指令来完成。

段寻址寄存器：地址 **1CH**

BC7275 可以实现以段（单独 **LED**）为单位的显示控制，这是通过段寻址寄存器来实现的。

通过给每个显示段（**LED**）分配一个地址，可以通过段寻址寄存器控制每个显示段的点亮和关闭。每个显示段的地址如下：

显示位	DP	G	F	E	D	C	B	A
0	07H	06H	05H	04H	03H	02H	01H	00H
1	0FH	0EH	0DH	0CH	0BH	0AH	09H	08H
2	17H	16H	15H	14H	13H	12H	11H	10H
3	1FH	1EH	1DH	1CH	1BH	1AH	19H	18H

4	27H	26H	25H	24H	23H	22H	21H	20H
---	-----	-----	-----	-----	-----	-----	-----	-----

写入段寻址寄存器的数据格式如下:

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
Seg	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀

其中 A₀-A₆ 为段地址, 其有效范围为 0-27H, 如地址超出该范围, 将不会有任何显示效果。Seg 为写入该段的数据, 当 Seg 为 0 时, 该 LED 被点亮, Seg 为 1 时, LED 熄灭。

全局操作寄存器: 地址 1DH

全局操作寄存器是个特殊的寄存器, 所有写入该寄存器的值, 都会被同时复制到所有的显示寄存器中。对于需要清屏或者点亮所有 LED 的操作, 可以通过向该寄存器写入 FFH 或者 00H 轻易地实现。注意, 写入该寄存器, 并不会改变闪烁控制等其它寄存器, 所以如果写入前有显示段或显示位被置为闪烁显示, 即便对群操作寄存器写入 FFH 清除显示, 原来闪烁的显示段也仍然维持闪烁显示的属性, 如该再次被点亮, 依然会以闪烁方式显示。

串行接口

BC7275 采用 SPI 串行接口, 可以连接标准的 3 线 SPI 接口, 或普通的 2 线制同步串行接口, 接口速率最高为 64Kbps, 可以直接与多种 MCU 的硬件串行接口相连, 充分利用控制器的硬件资源, 节省处理器时间。对没有 SPI 接口资源的 MCU, 也可用非常短的代码用 I/O 口实现其通讯协议。

一、数据格式

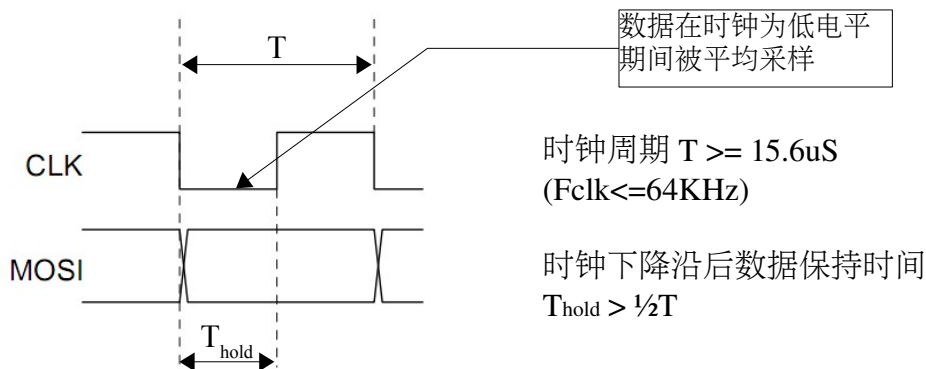
BC7275 的指令都是以两个字节为一个单位。这两个字节第一个字节为寄存器地址, 第二个字节为数据, 传送的时候高位 (MSB) 在前, 数据结构如下:

寄存器地址								数据字节							
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0	A ₄	A ₃	A ₂	A ₁	A ₀	d ₇	d ₆	d ₅	d ₄	d ₃	d ₂	d ₁	d ₀
MSB								▶ LSB							

指令字节的低 5 位 A₄-A₀ 为寄存器地址, 有效地址范围为 00H-1DH. 如果输入的地址范围超出了有效的地址范围, 则该指令会被忽略。数据的传输为单方向传输, 只能由 MCU 传向 BC7275.

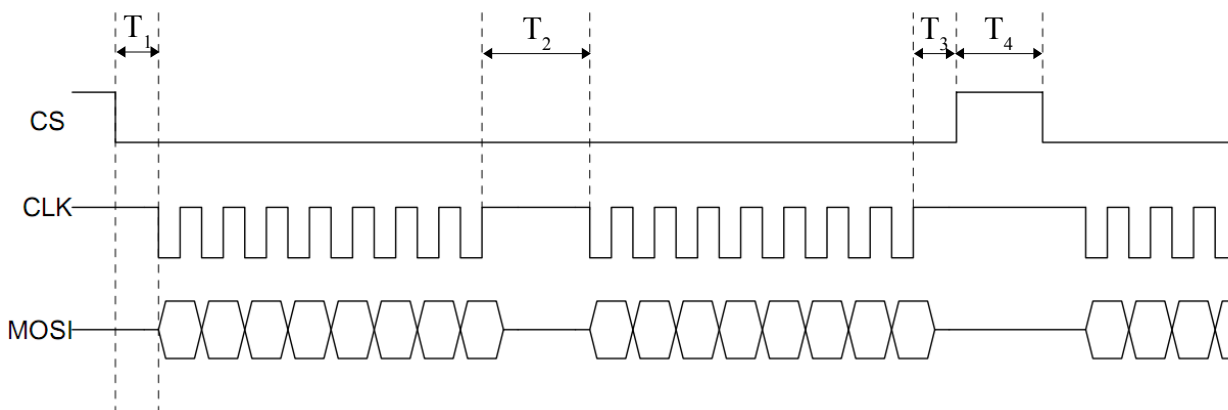
二、时序

BC7275 可以连接标准的同步串行接口, 比如 SPI 接口, 可以连接为标准 3 线的方式, 包括数据 (MOSI), 时钟 (CLK) 和片选 (\overline{CS}). 当 MCU 的 I/O 口线紧张的时候, 也可以将 \overline{CS} 直接接地, 这时只需要 2 跟口线 MOSI 和 CLK, 即可完成与 MCU 的接口。



对于使用硬件 SPI 接口的情况, 请设置 SPI 接口为如下参数: CLK 空闲状态为高电平, 数据在第一个时钟沿(下降沿)改变, 在第二个时钟沿(上升沿)被采样, 串口速率 $\leq 64\text{Kbps}$.

如果使用 I/O 口模拟 SPI 接口, 因为数据在 CLK 低电平期间被 BC7275 平均采样, 因此需要在将 CS 置为低电平前将数据线设置好相应的高低电平, 并维持数据线稳定至少半个时钟周期以上($T_{\text{hold}} > \frac{1}{2}T$).



图中, T_1 为指令中 CS 下降沿到第一个 CLK 时钟脉冲下降沿的时间, 此时间要求大于 $\frac{1}{2}$ 个时钟周期 T, 当使用硬件 SPI 接口时, 从数据写入 SPI 接口发送寄存器到第一个时钟脉冲输出, 一般会加入约 1 个时钟周期的延时, 用户使用程序中将 CS 置低电平, 然后写入 SPI 接口寄存器即可, 对此延时可不予考虑, 但如果所用 MCU 的 SPI 接口没有此特性, 或者使用软件模拟 SPI 接口, 则需人工加入此延时。

T_2 为一个指令中, 寄存器地址字节和数据字节之间的延时, 此时间要求大于 1 个时钟周期 T. 当使用硬件 SPI 接口时, 一般硬件会自动在两个传送的字节之间加入此延时, 如果所用 MCU 不能产生此延时, 或使用软件模拟 SPI 接口时, 则需人工加入延时。

T_3 为最后一个时钟脉冲上升沿到 CS 上升沿的时间, 要求大于 $\frac{1}{2}$ 个时钟周期 T. 当使用硬件 SPI 接口时, 如果用户程序在等待 SPI 接口状态变为“发送完成”状态后再将 CS 置 1, 则此时间可由硬件保证, 因为时钟脉冲的最后一个上升沿到整个时钟周期结束正好还有 $\frac{1}{2}$ 个时钟周期。如果所用 SPI 接口不符合该特性, 或使用软件模拟 SPI 接口, 则该延时需要人工加入。

T_4 为前一个 CS 信号的上升沿到下一个 CS 信号的下降沿之间的延时, 要求大于 $\frac{1}{2}$ 个时钟周期 T.

在每个指令的 2 个字节之间, 必须保持 CS 为低电平状态, CS 可以在每个指令之间置为高电平。CS 的作用有 2 个, 1 个是在系统中有多多个 SPI 器件时作为片选信号, 另外一个作用是在 CS 的下降延, BC7275 内部会对 SPI 串口寄存器做复位操作, 保证数据传输正确。每个

指令传送完成后，恢复CS为高电平的操作并不是必须的，当连续传送批量指令时，可一直维持CS为低电平，在所有指令传送完成后再恢复为高电平。

三、接口复位机制

SPI口必须有某种机制使得串行接口可以被复位，以防当因某种原因时钟和数据线失去同步时，造成传送的数据混乱。当使用CS信号时，接口在CS的下降沿被复位。

当CS直接接地时，BC7275具有独特的时间复位机制，此机制生效时，当时钟线上“静音”超过2个显示扫描周期（约30mS）时，SPI接口就会被复位。也就是说，当CLK引脚上无电平变化超过2个扫描周期，当前所接收的数据将被抛弃，下一个CLK脉冲对应的数据将作为新指令的MSB。

在时间复位机制下，即便前一个指令因为某种因素造成了时钟与数据不同步（比如主机发送了16个时钟脉冲，数据已经传送完，但BC7275只接收到了15个时钟脉冲，还在继续等待最后一个数据位），只要后面一个双字节指令和前面一个指令之间的时间间隔大于两个扫描周期（期间时钟线CLK上须没有时钟脉冲），BC7275就会对内部接收缓存器复位，这样后面一个指令仍可以正确被接收，而不会将后一个指令的最高位当作前一个指令的最后一位而使错位持续下去，造成不可预估的后果。

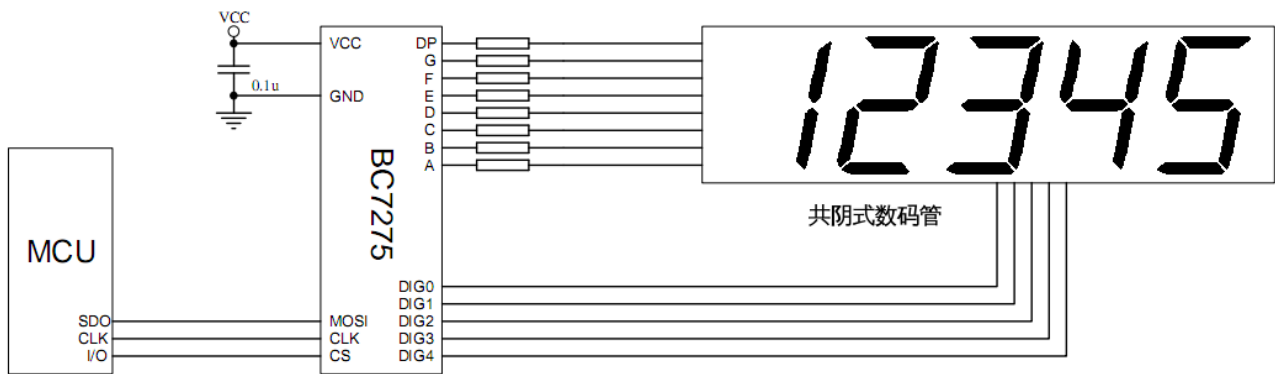
此时间复位机制，只有在系统上电时CS处于低电平的情况下才会启动，如果当系统上电时CS为高电平，则时间复位机制不会启动，BC7275不会检测CLK信号“静音”的时间，只有CS信号可以将SPI接口复位。

时间复位机制生效的情况下，CS片选端依然具有作用，即如果CS变为高电平，BC7275将不会接收任何数据，同时在CS信号的下降沿，BC7275的串口缓存器仍然将被复位。

不同接口方式比较		
接口方式	优点	缺点
使用CS片选信号	通讯可靠性高；可以和其它SPI接口器件共用接口；对CS最大脉冲宽度无要求，若使用软件模拟，通讯过程中允许被其它中断长时间打断	占用口线多
不使用CS(直接接地)	节省I/O口资源	接口不能复用，两个指令中间最好间隔2个扫描周期以上。最大CLK脉冲宽度不能大于2个扫描周期，如使用软件模拟SPI接口，需注意通讯不能被执行时间大于两个扫描周期的中断打断，调试程序时需注意数据传送中途不可设置断点。

外围电路

BC7275外围电路简单，只需很少的外部元件，就可以构成多位LED显示电路。典型的电路如下：



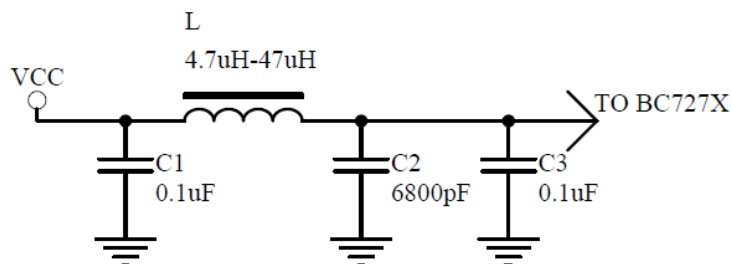
段驱动限流电阻，可以用下面近似公式计算：

$$R = 67 * (V_{CC} - U_{LED}) - 100$$

其中， V_{CC} 为 BC7277 电源电压， U_{LED} 为 LED 正向管压降。电压单位为‘伏’，结果电阻单位为‘欧姆’。当 $V_{CC}=5V$ ， $U_{LED}=1.85V$ ，可得结果 $R=111\Omega$ ，可取近似值 100Ω ；当 $V_{CC}=3.3V$ 时，计算结果为 -2.85Ω ，取近似值为 0Ω ，即限流电阻可以省略；当 $V_{CC}=3V$ 时，计算结果 $R=-23\Omega$ ，表明此时限流电阻可以省略，且在此条件下使用亮度会有所降低。

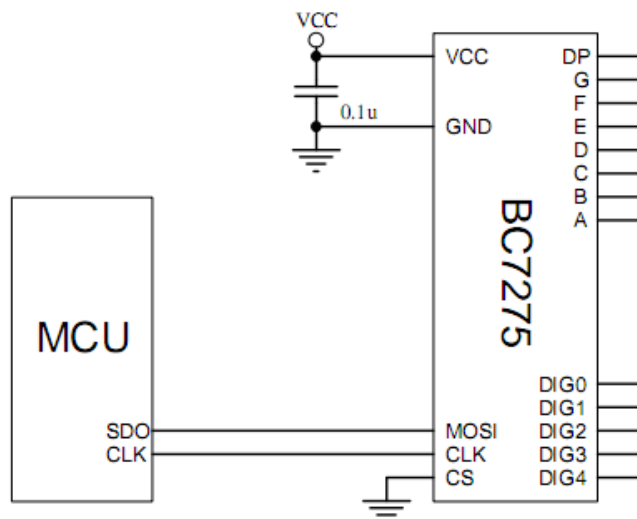
一、滤波电路

当电路具有以下几种情况之一：使用于高干扰环境、使用开关电源供电，以及电路中有模拟电路部分的情况，最好在 BC7275 的供电电路中，串入如下的滤波电路，一方面减小干扰可能对芯片正常工作的影响，另一方面也可滤除显示电路给模拟电路带来的噪音。



二、串行接口电路

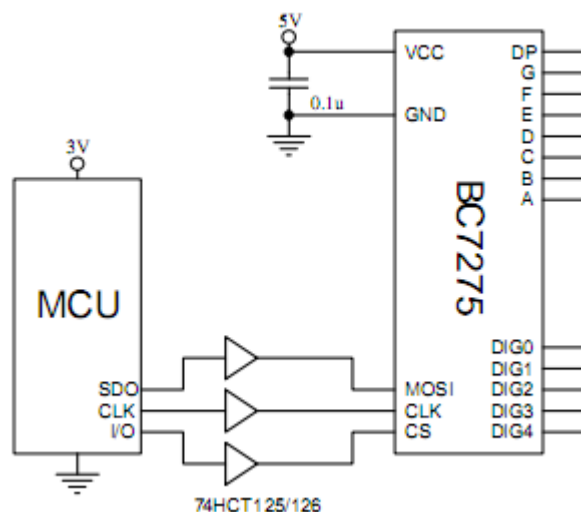
BC7275 采用 SPI 接口，接口速率可以达到 64Kbps，可以使用 MOSI, CLK, CS 3 线连接标准的 SPI 接口(见典型应用图)，也可以接为 2 线方式，将 CS 直接接地，只使用时钟线和数据线。



一般 MCU 上 SPI 接口均可设置不同的工作模式，与 BC7275 相连时，应该设置成主机 (Master) 模式，CLK 空闲时高电平，数据在第一个时钟沿传递，在第二个时钟沿采样。接口的时钟频率，必须在 64KHz 以下。

因为接口部分的高电平最低输入电压在电源电压为 5V 时为 3.7V，因此，如果 BC7275 的电源电压为 5V，无法直接与 3V 系统接口，因 BC7275 本身可工作在 3V 电源电压下，因此当系统 MCU 为 3V 或 3.3V 供电时，建议 BC7275 也采用相同的供电电压。

如果确需令 BC7275 工作于 5V，需要外加电平转换电路。74HCT 的逻辑电路，当电源电压为 5V 时，高电平最低输入电压为 2V，可以直接接受 3V 系统的输入。在通讯线路中串入这样的缓冲器，即可解决电平匹配的问题。如图：



程序流程

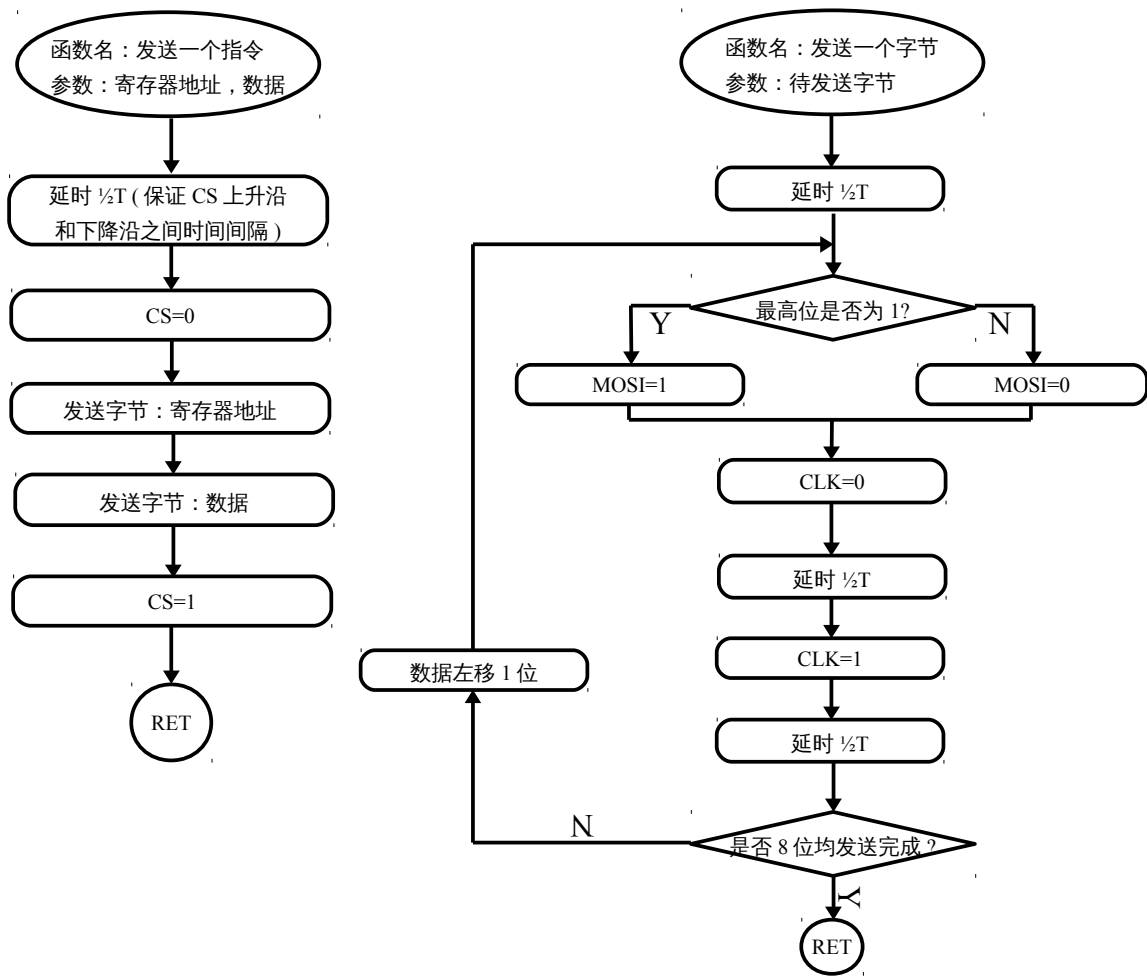
因为接口协议简单，而且很多新型单片机都带有内置的硬件 SPI 接口，因此，与 BC7275 的软件接口也非常容易实现。简单来说，可以分为三种方式：

- 纯软件使用 I/O 口模拟
- 使用硬件 SPI 接口，但使用查询方式

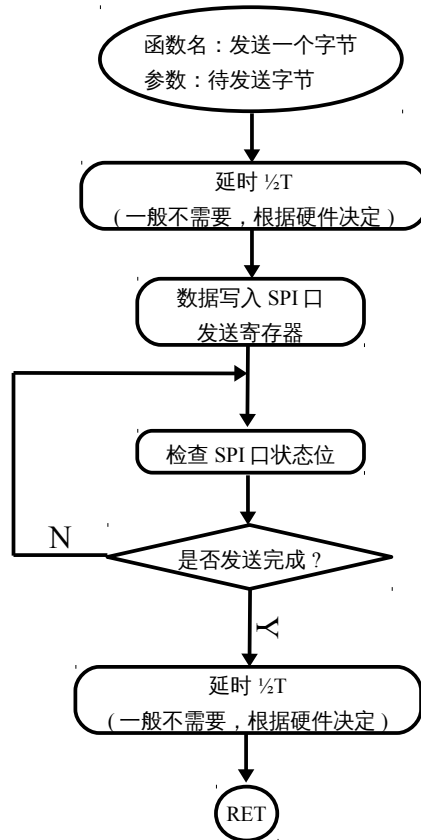
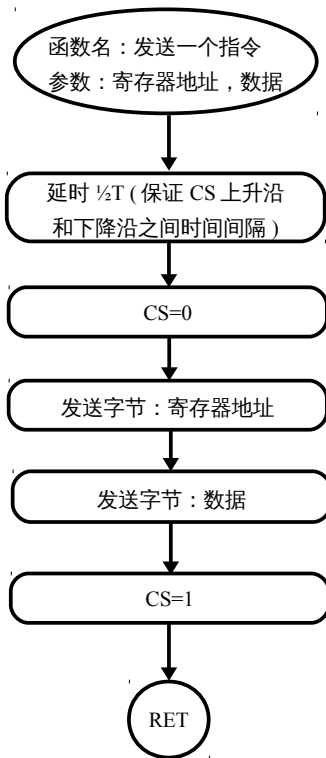
- 使用硬件 SPI 接口，且使用 SPI 接口中断
三种方式可谓各有优缺点，下面是三种接口方式的比较：

	软件模拟 SPI 接口	硬件 SPI 接口/查询方式	硬件 SPI 接口/中断方式
所需代码长度	较短	最短	较长
是否可以灵活安排 I/O 口使用	可以	不能，由 MCU 硬件决定	不能，由 MCU 硬件决定
通讯时是否可同时执行其它任务	不可以	有限，必须等待前一字节发送完成，连续通讯时几乎与软件模拟方式一样	可以。主程序只需将所需发送数据写入缓冲区
当使用 2 线方式 (CS 直接接地) 时，通讯过程中是否需要屏蔽其它执行时间超过 2 个扫描周期的中断	需要屏蔽	不需要	不需要

因为数字显示往往是系统中对实时性要求最低的、同时也是优先级最低的任务，并不需要使用中断来做后台的 SPI 接口控制，因此，一般建议使用硬件 SPI 接口+查询方式的方案。
软件模拟方式流程图：

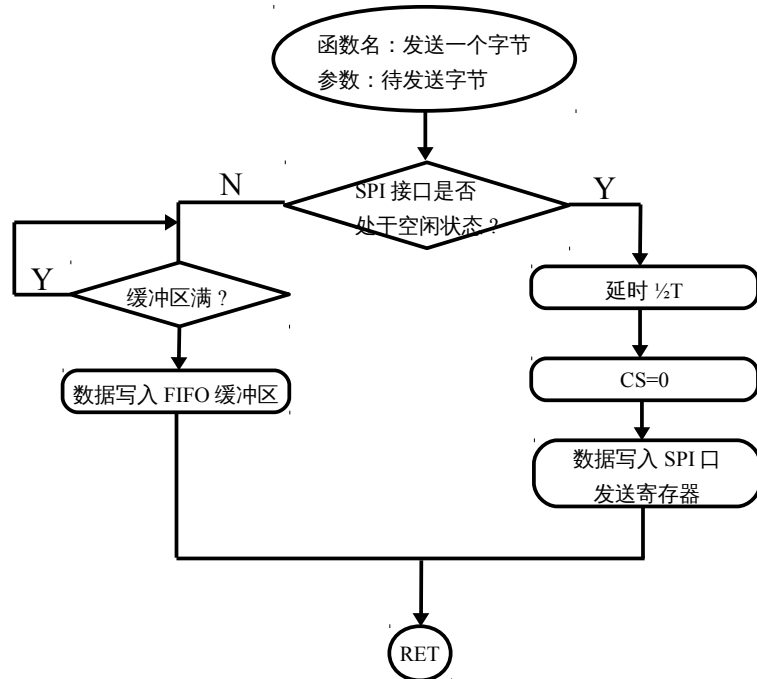
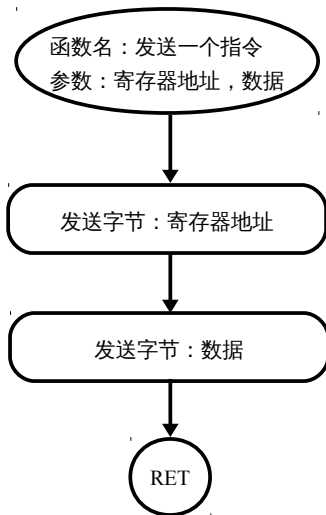


硬件 SPI 接口+查询方式流程图:

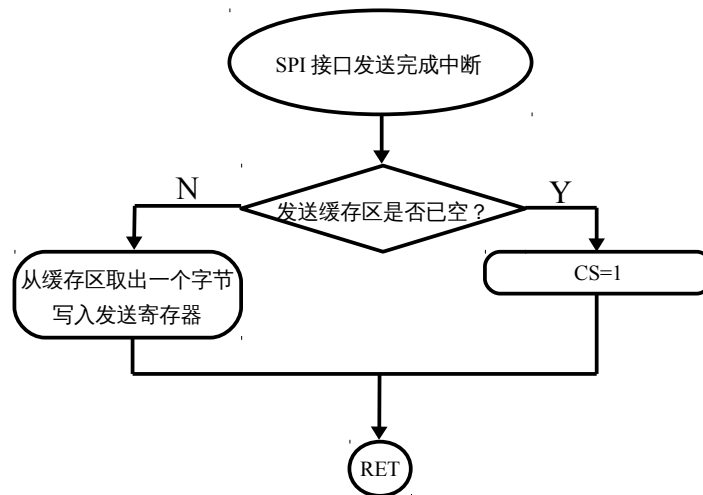


硬件 SPI 接口+中断方式流程图:

(注: 流程图中省略了可能需要的在 CS 下降延后和上升沿前的 1/2T 延时)



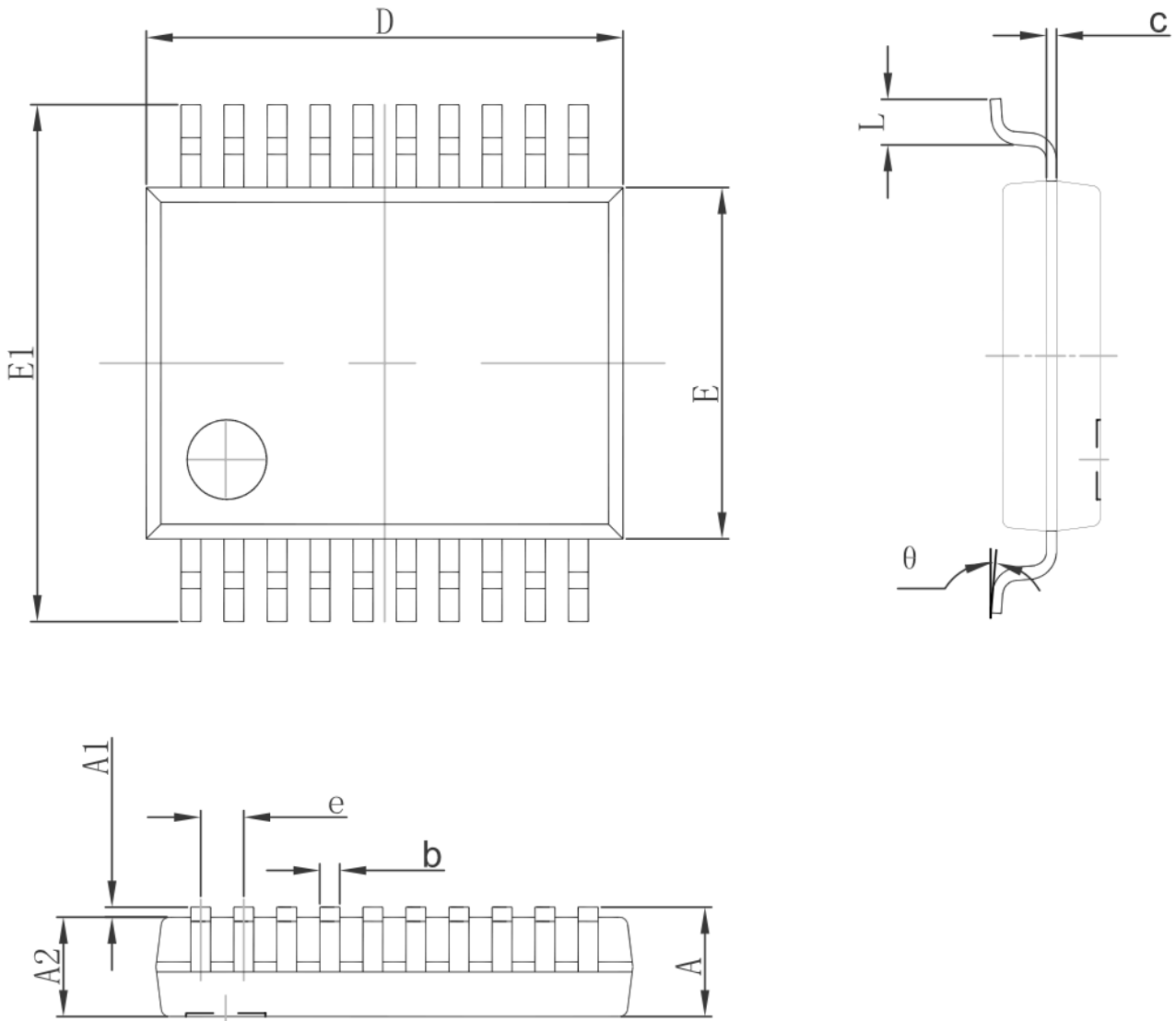
中断服务程序：



注：在中断方式下，因为缓冲区的存在，“发送一个指令”子程序的调用和数据实际在 SPI 口上发出的时间会有不同步的问题，因此，不能如其它两种方式一样在“发送一个指令”子程序中控制 CS 片选信号的电平变化。比较好的方式是，在“发送一个字节”子程序中，检查如果当前 SPI 口处于空闲状态，则先将 CS 信号置低，然后将数据写入 SPI 口发送寄存器开始发送。在 SPI 口中断处理程序中，如果发现当前缓冲区已经为空，表示所有数据均发送完成，此时将 CS 恢复为高电平。这个方式和在“发送指令”子程序中控制 CS 效果有所区别，在“发送指令”子程序中控制 CS 时，CS 信号随每个指令都会跳变一次；中断方式下，CS 则按指令组为跳变，每组连续发出的指令中间，CS 都将维持在低电平，直至所有数据发送完成，才会恢复为高电平。

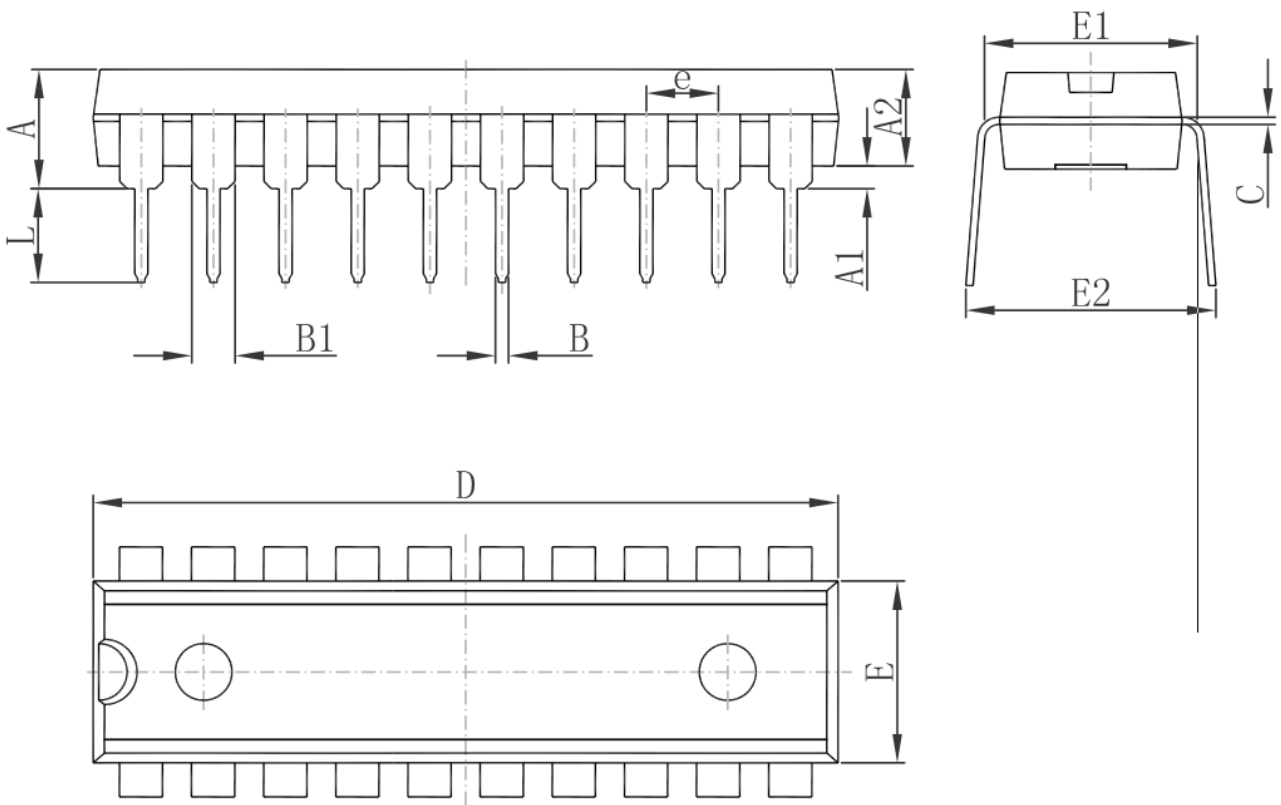
封装信息

BC7275 有 DIP20 和 SSOP20 两种封装，封装的尺寸分别如下：
SSOP20 封装尺寸



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A		1.730		0.068
A1	0.050	0.230	0.002	0.009
A2	1.400	1.600	0.055	0.063
b	0.220	0.380	0.009	0.015
c	0.090	0.250	0.004	0.010
D	7.000	7.400	0.276	0.291
E	5.100	5.500	0.201	0.217
E1	7.600	8.000	0.299	0.315
e	0.65(BSC)		0.026(BSC)	
L	0.550	0.950	0.022	0.037
θ	0°	8°	0°	8°

DIP20 封装尺寸



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	3.710	4.310	0.146	0.170
A1	0.510		0.020	
A2	3.200	3.600	0.126	0.142
B	0.380	0.570	0.015	0.022
B1	1.524 (BSC)		0.060 (BSC)	
C	0.204	0.360	0.008	0.014
D	25.950	26.550	1.022	1.045
E	6.200	6.600	0.244	0.260
E1	7.320	7.920	0.288	0.312
e	2.540 (BSC)		0.100 (BSC)	
L	3.000	3.600	0.118	0.142
E2	8.400	9.000	0.331	0.354

附录:

BC727X 系列芯片

	BC7275	BC7276	BC7277
芯片封装	DIP20/SSOP20	DIP20/SSOP20	SDIP24/SSOP24
可驱动数码管位数	5	8 或 16	9
键盘接口	无	16 键	16 键
数码管类型	共阴	共阳	共阴
外接驱动	不需要	三极管和移位寄存器	不需要
驱动大尺寸数码管	不可以	可以	不可以