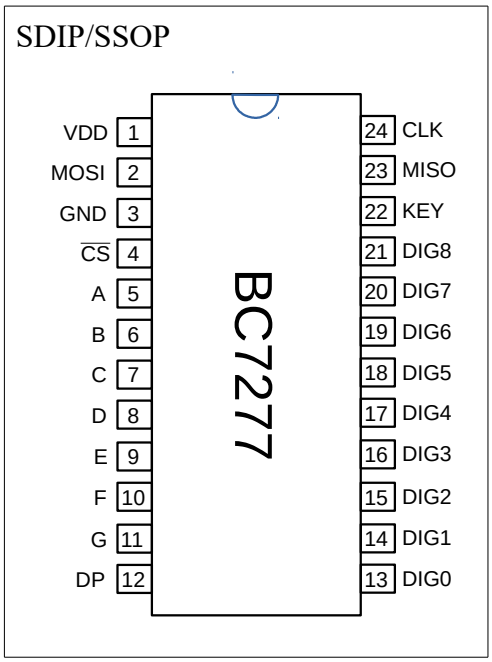
BC7277

9 位 LED 数码管及 16 键键盘接口芯片

(第四版)



特点：

- 可驱动 9 位共阴式数码管或 72 只 LED
- 无需外围器件
- 9 个显示位均可单独闪烁显示
- 单独 LED 闪烁
- 闪烁速度可调
- 段寻址可以单独控制任意显示段
- 译码显示时小数点显示不受显示更新影响
- 可直接访问显示寄存器（显示特殊字符）
- 16 键键盘支持任意组合键和长按键
- 标准 SPI 串口，可用 2 线、3 线或 4 线方式
- SSOP24 小体积封装
- 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

BC7277 具有 9 位数码管显示管理功能，无需外围器件，即可以构成 9 位(72 段)LED 显示和键盘驱动。因为支持段寻址，可以独立地控制每一个显示段，也非常适用于独立的 LED。BC7277 支持闪烁显示功能，且闪烁速度可调，每一位均可独立控制闪烁属性，而前 8 位更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。

BC7277 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。

键盘可最多支持 16 键，芯片内含去抖动电路，可以支持任意的组合键，长按键，可以支持各种常开或常闭开关。

BC7277 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。BC7277 的 SPI 接口可以接为 2 线、3 线或 4 线方式。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℃
工作温度	-40 至+85℃
任意脚对地电压	-0.5 至 6.0V

电特性：

（除特别说明外，T_A=25℃, Vcc=5.0V)

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	
输入低电平			1.4	V	
输入高电平	3.7			V	当 Vcc=3V 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		16		mS	

引脚说明：

名称	序号	说明
VDD	1	正电源端，电压范围 2.7-5.5V
MOSI	2	SPI 口数据输入端，接 MCU 的 SPI 口数据输出端 移位寄存器数据线，内含弱上拉电阻
GND	3	接地端
$\overline{CS}$	4	片选端，低电平有效，内含弱上拉电阻
A-DP	5-12	A 段-DP 段段驱动
DIG0-DIG8	13-21	位驱动输出，接数码管共阴极
KEY	22	按键状态指示，每当按键状态变化时，KEY 的电平会发生翻转
MISO	23	从机数据输出，BC7277 数据输出，接 MCU 的 SPI 数据输入，open drain 输出
CLK	24	SPI 接口时钟输入端，内含弱上拉电阻

内部寄存器

BC7277 内部具有 23 个寄存器，包括 16 个显示寄存器，以及 14 个特殊寄存器。地址

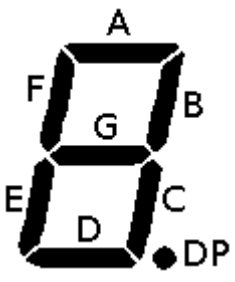
范围为 00H-1DH，其中 00H-08H 为显示寄存器，其余为特殊寄存器。

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	
02H	第 2 位显示寄存器	FFH	
...	第 3-7 位显示寄存器	FFH	
07H	第 8 位显示寄存器	FFH	
...	无效寄存器	...	此地址范围对 BC7277 无效
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	
...	...	...	
17H	第 7 位段闪烁控制寄存器	00H	
18H	0-7 位位闪烁控制寄存器	00H	bit0-bit7 分别对应显示位 0-7， 1=闪烁,0=不闪烁
19H	第 8 位位闪烁控制寄存器	00H	只有 bit0 有效，对应显示位 8
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-08H

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-17H

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，映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, 19H

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

19H								18H							
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
-	-	-	-	-	-	-	位8	位7	位6	位5	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

BC7277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。

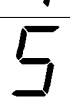
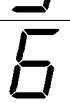
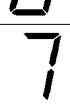
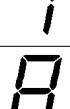
译码寄存器：地址1BH

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。d_3:d_0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	

d_3	d_2	d_1	d_0	d_3:d_0 (16 进制值)	显示
0	0	0	1	1	
0	0	1	0	2	
0	0	1	1	3	
0	1	0	0	4	
0	1	0	1	5	
0	1	1	0	6	
0	1	1	1	7	
1	0	0	0	8	
1	0	0	1	9	
1	0	1	0	A	
1	0	1	1	B	
1	1	0	0	C	
1	1	0	1	D	
1	1	1	0	E	
1	1	1	1	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

BC7277 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
5	2FH	2EH	2DH	2CH	2BH	2AH	29H	28H
6	37H	36H	35H	34H	33H	32H	31H	30H
7	3FH	3EH	3DH	3CH	3BH	3AH	39H	38H
8	47H	46H	45H	44H	43H	42H	41H	40H

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 A_0-A_6 为段地址，其有效范围为 0-47H。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，即便对全局操作寄存器写入 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，如果再次被点亮，依然会以闪烁方式显示。

串行接口

BC7277 采用 SPI 串行接口，可以连接标准的 4 线 SPI 接口，如果不使用键盘，可以不接 MISO 口，当不需要片选时，CS 可直接接地，这样最少只需 2 跟口线，即可完成与 BC7277 的接口。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

BC7277 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	d7	d6	d5	d4	d3	d2	d1	d0

MSB ⟶ LSB

指令字节的低 5 位 A_4-A_0 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范围，则该指令会被忽略，有时可以利用这个特性，向 BC7277 发送一个‘伪指令’，目的只在于得到输出的键盘映射值。

在接收指令的同时，BC7277 会在 MISO 引脚上输出 16bit 的数据，该数据为键盘的映射，用户可以通过该数据获取键盘的状态。输入和输出为同时进行，每次 CLK 脉冲到来时，数据在 CLK 脉冲的低电平期间被 BC7277 平均采样，同时输出键盘映射数据，MCU 一侧使用硬件 SPI 接口时，应该设置成在 CLK 的上升沿采样 MISO 数据。详见时序图。

输出的键盘映射数据输出也是 MSB 在前，具体如下：

键盘映射高字节								键盘映射低字节							
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
S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
MSB ⟶ LSB															

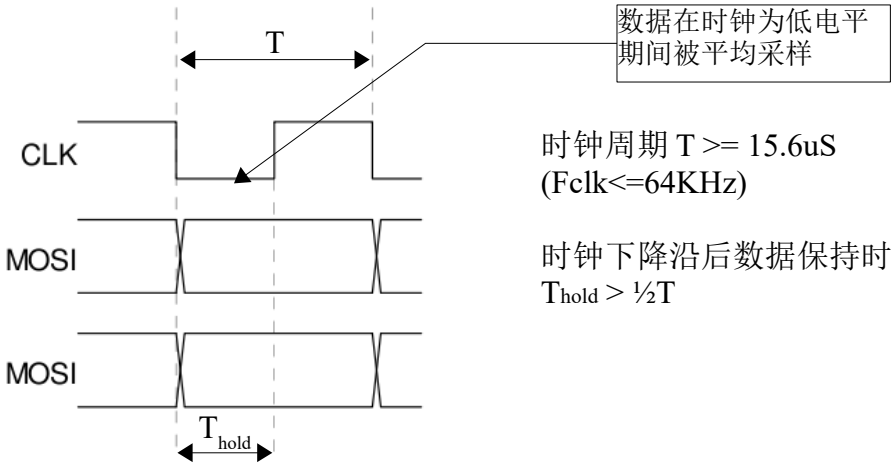
按键处于开路状态时，对应的键盘映射为 1，导通状态时，对应的映射值为 0. 换而言之，如果没有连接键盘，则输出的值将为 0xFFFF. 键盘映射数据位和键盘按键间的对应关系如下表，请参考后文中键盘部分的电路图。

	DIG3	DIG2	DIG1	DIG0
DIG4	S3	S2	S1	S0
DIG5	S7	S6	S5	S4
DIG6	S11	S10	S9	S8
DIG7	S15	S14	S13	S12

当使用 CS 片选信号时，键盘映射值会在 CS 信号的下降沿得到更新，因此每次输出的键盘映射值均为当时最新的键盘状态。当 CS 信号被直接接地时，键盘映射值只在每次指令传输的结尾时得到更新，因此每次每次输出的键盘映射值，实际上是上一个指令传送结束时的键盘状态，如果两个指令的间隔时间比较长，则输出的数据可能并不能及时反应键盘的状态。这时如果想取得最新的键盘状态，可以连续发送两个指令或者伪指令。

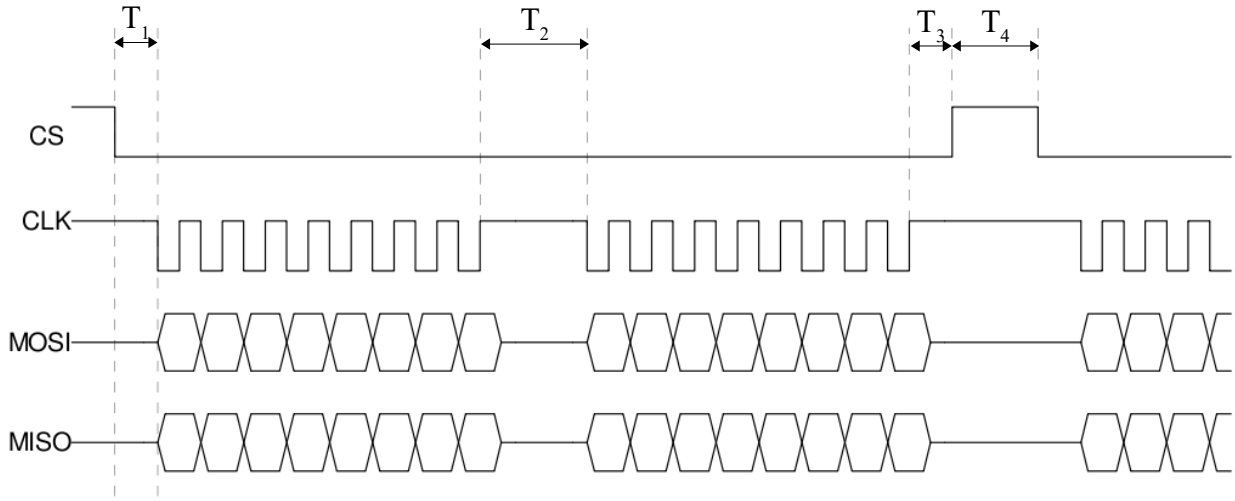
二、时序

BC7277 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 4 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。如果不使用键盘，还可以省去 MISO 的连接，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≤64Kbps**. 如果 SPI 接口支持 16 位模式，也可以设置成 16 位模式将指令（地址）和数据合并发送。

如果使用 I/O 口模拟 SPI 接口，因为数据在 CLK 低电平期间被 BC7277 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_{hold}>½T).



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 ½ 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，一般会自动加入约 1 个时钟周期的延时，用户使用时程序中先将 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，寄存器地址字节最后一个时钟上升沿和数据字节第一个时钟下降沿之间的延时，此时间要求大于等于 ½ 个时钟周期 T. 当使用硬件 SPI 接口时，此时间由硬件保证，如果使用软件模拟 SPI 接口时，则需予以注意，确保此时间满足要求。

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 ½ 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 ½ 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。

T4 为前一个 CS 信号的上升沿到下一个 CS 信号的下降延之间的延时，要求大于 ½ 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 个是在系统中有多个 SPI 器件时作为片选信号，另外一个作用是在 CS 的下降延，BC7277 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，BC7277 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 32mS）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 BC7277 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)，BC7277 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。

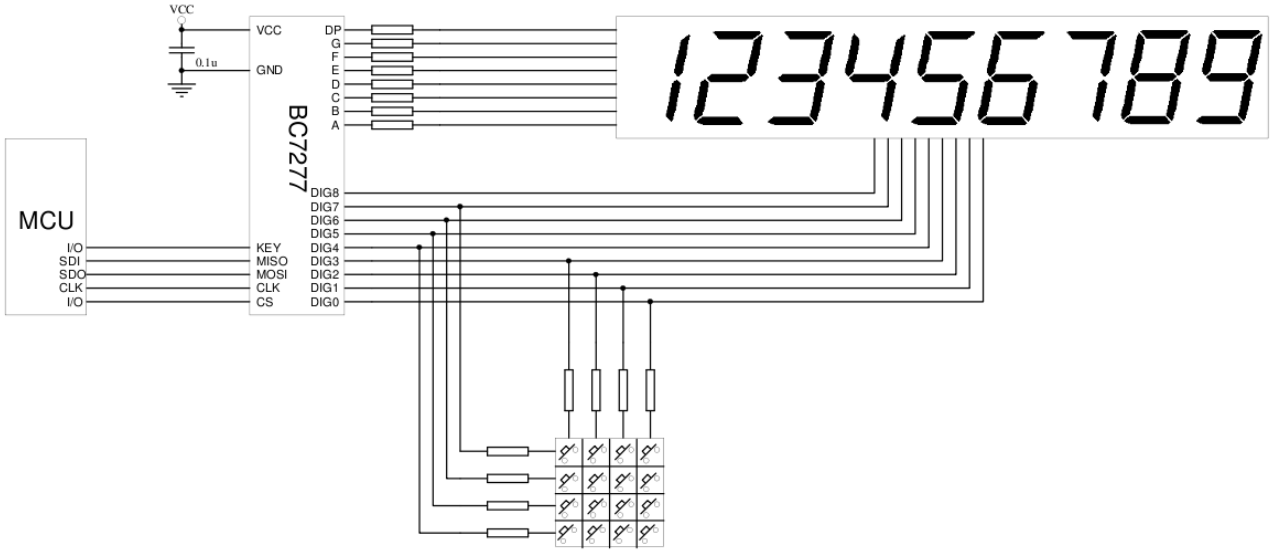
此时间复位机制，只有在系统上电时 CS 处于低电平的情况下才会启动，如果当系统上电时 CS 为高电平，则时间复位机制不会启动，BC7277 不会检测 CLK 信号“静音”的时间，只有 CS 信号可以将 SPI 接口复位。

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，BC7277 将不会接收任何数据，同时在 CS 信号的下降沿，BC7277 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；每次输出键盘映射值为最新值	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（即频率须大于 31Hz），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

BC7277 外围电路简单，只需外接少量限流电阻，就可以构成多位 LED 显示电路。典型的电路如下：



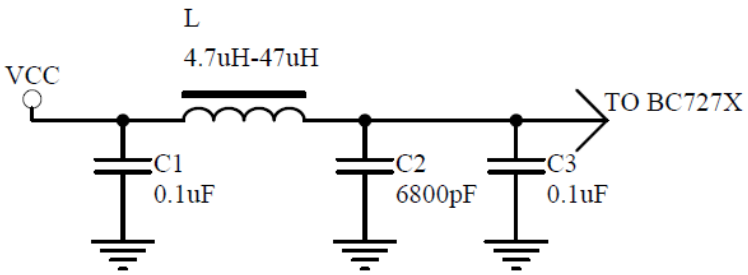
限流电阻，可以用下面近似公式计算：

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

其中，VCC 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 VCC=5V，U_{LED}=1.85V，可得结果 R=111Ω，可取近似值 100Ω；当 VCC=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 VCC=3V 时，计算结果 R=-23Ω，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。

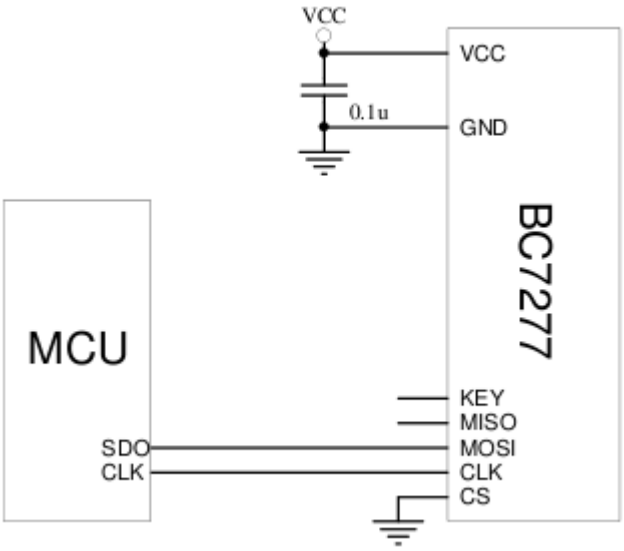
一、滤波电路

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 BC7277 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。



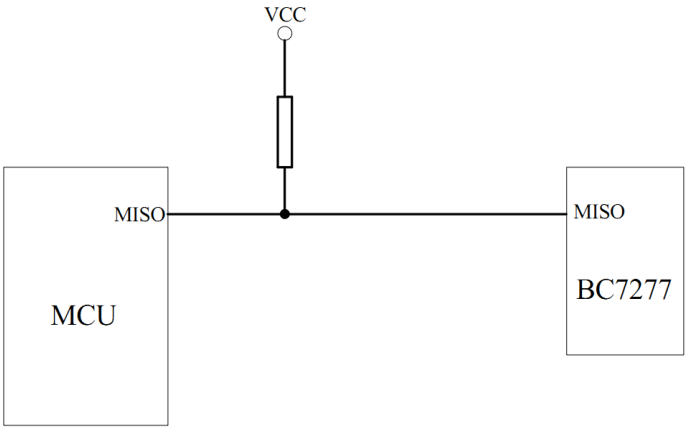
二、串行接口电路

BC7277 采用 SPI 接口，接口速率可以达到 64Kbps，可以使用 MISO, MOSI, CLK, CS 4 线连接标准的 SPI 接口(见典型应用图), 不用键盘时最低可以接为 2 线方式，将 CS 直接接地，只使用时钟线和数据线。



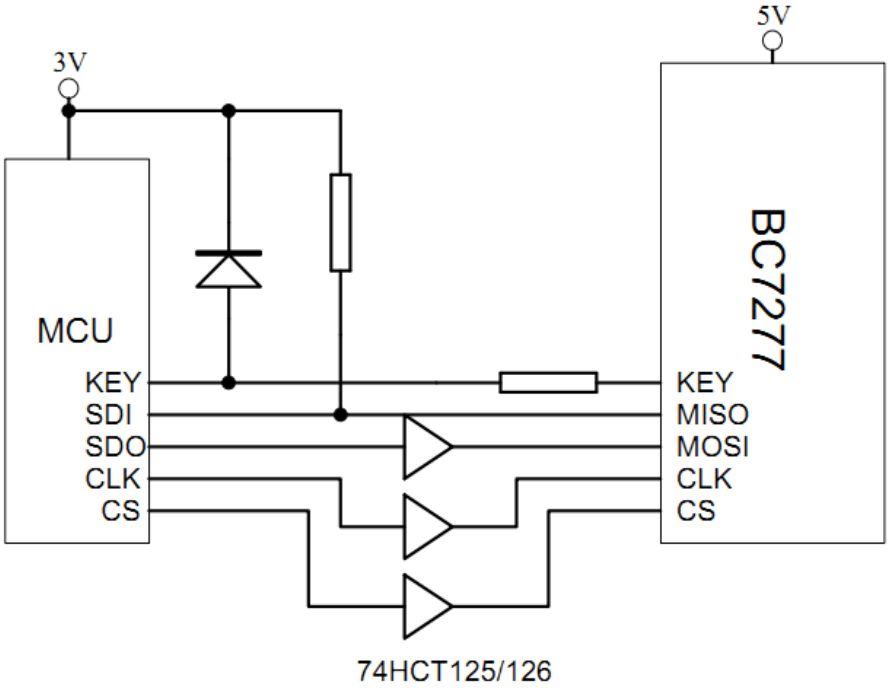
一般 MCU 上 SPI 接口均可设置不同的工作模式，与 BC7277 相连时，应该设置成主机(Master)模式，CLK 空闲时高电平，数据在第一个时钟沿传递，在第二个时钟沿采样。接口的时钟频率，必须在 64KHz 以下。

BC7277 的 MISO 引脚为漏极开路输出(Open Drain)，因此在与 MCU 相连时，此引脚上必须外加上拉电阻，如图：



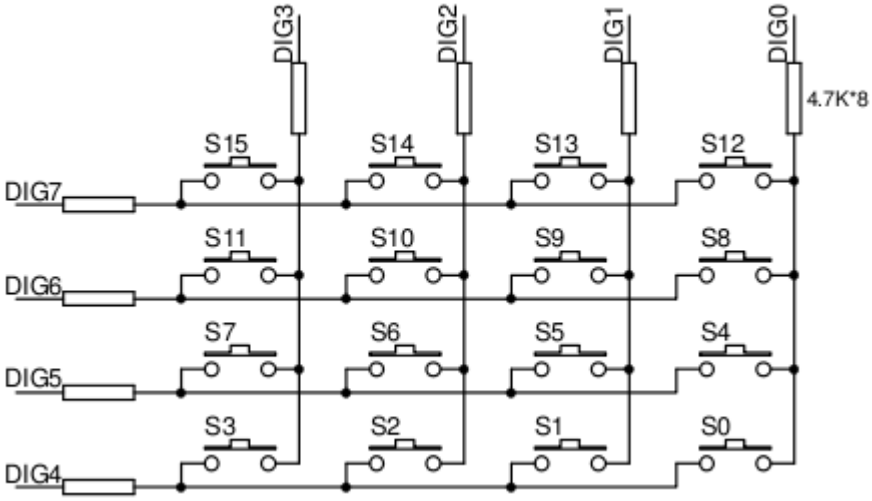
因为接口部分的高电平最低输入电压在电源电压为 5V 时为 3.7V, 因此，如果 BC7277 的电源电压为 5V, 无法直接与 3V 系统接口，但因 BC7277 本身可工作在 3V 电源电压下，因此当系统 MCU 为 3V 或 3.3V 供电时，建议 BC7277 也采用相同的供电电压 。

如果确需令 BC7277 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：



三、键盘电路

16 个键盘接成 4x4 的矩阵，接在 DIG0-DIG7 上，连接方式见下图。



矩阵的行和列上，都需要串入电阻，电阻的作用在于防止位驱动输出之间的短路，4.7K 的阻值可以适用于大多数情况。

因为 BC7277 采用键盘映射寄存器反映每个按键的状态，因此可以适用于各种常开和常闭开关，而不限于常用的常开按键。不过因为上电时系统复位状态键盘映射寄存器为 0xFFFF, 对应所有键盘开路状态，因此如果按键中有的采用的是常闭式，则上电一开始，会有一次 KEY 引脚的电平跳变。

KEY 引脚在上电时可以是任意状态，键盘矩阵每个扫描周期(约 16ms)会被扫描一遍。换而言之，KEY 最短的变化周期，为一个扫描周期。

KEY 在每次键盘状态发生变化时电平跳变一次，但不等于每次电平变化对应一个按键的变化。如果在一个扫描周期内，同时有 2 个或 2 个以上按键的状态发生了改变，KEY 仍然只

会发生一次电平的变化，即一次 KEY 引脚电平变化有可能对应多个按键的同时变化。**KEY 不是键盘状态的反映，用户应该在意 KEY 上电平的变化，而不是该引脚具体为高电平还是低电平，因为同样的电平可以对应任何的键盘状态，没有按键时，KEY 可以是高电平或低电平。**举例说明：如果系统在某一时刻所有按键为开路状态，且 KEY 电平为高电平；如果此时有两个按键同时(时间相差 1 个扫描周期以内)被按下变为短路状态，则 KEY 会变成低电平，过了一定时间，两个被按下的按键中的一个被释放重新变为开路状态，此时 KEY 会恢复成高电平，但此时对应的键盘状态为有一个键被按下，再过一段时间此按键也被释放，此时按键状态恢复成和初始状态一样，即所有按键均为开路状态，但此时 KEY 会再次发生电平翻转，变成低电平。最后键盘的状态和初始时刻一样，但 KEY 的状态由高电平变成了低电平。

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 BC7277 的软件接口也非常容易实现。简单来说，可以分为三种方式：

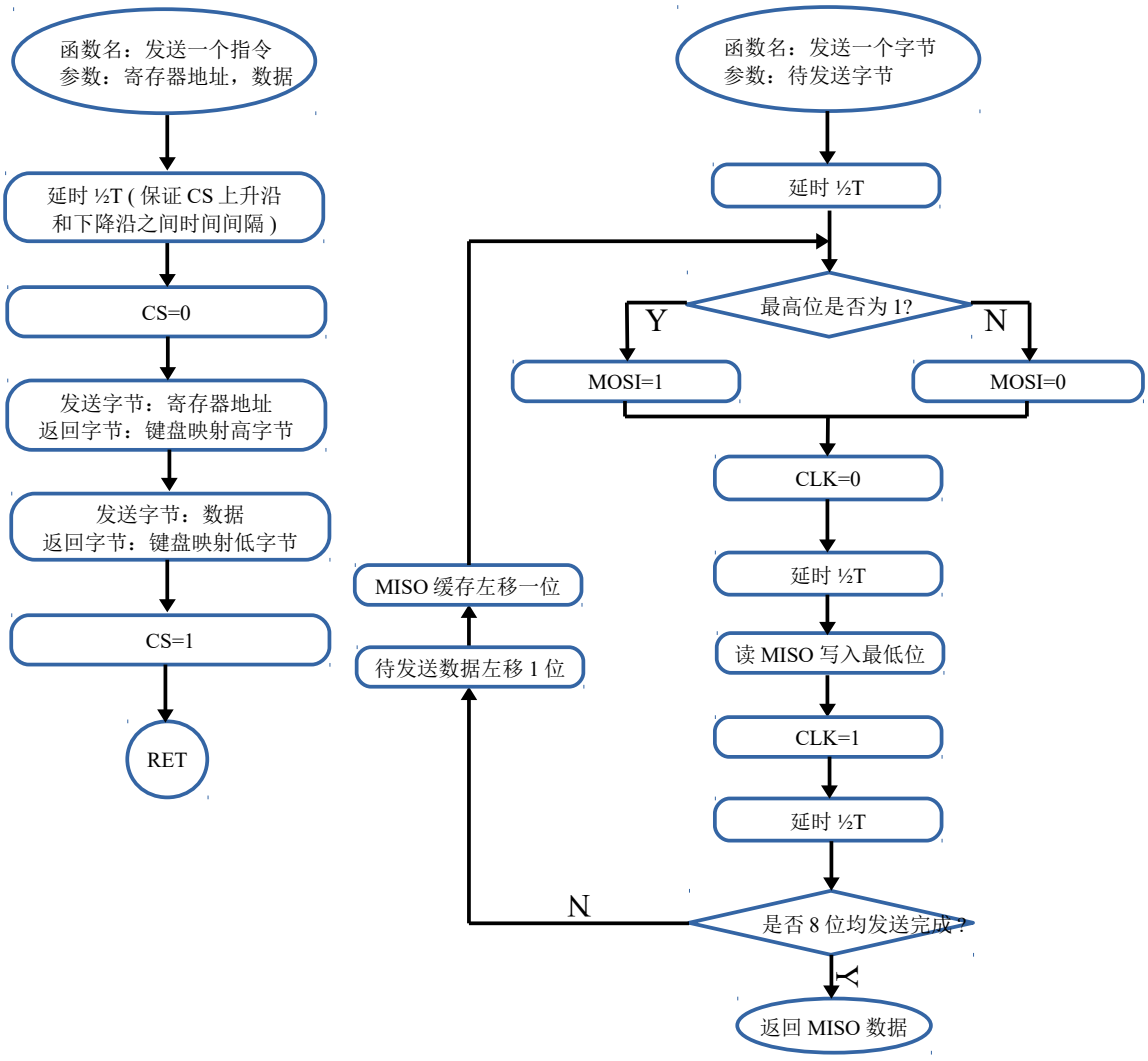
- 纯软件使用 I/O 口模拟
- 使用硬件 SPI 接口，但使用查询方式
- 使用硬件 SPI 接口，且使用 SPI 接口中断

三种方式可谓各有优缺点，下面是三种接口方式的比较：

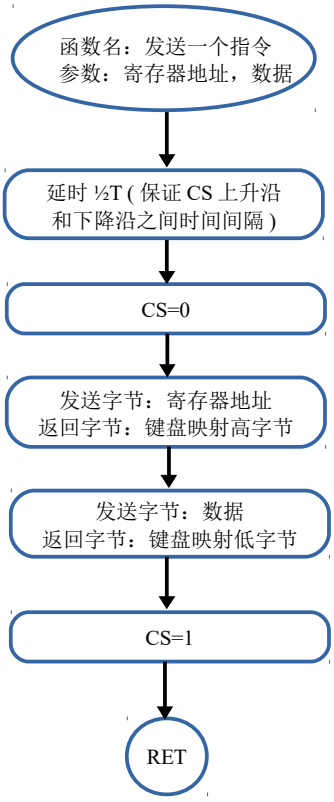
	软件模拟 SPI 接口	硬件 SPI 接口/查询方式	硬件 SPI 接口/中断方式
所需代码长度	较短	最短	较长
是否可以灵活安排 I/O 口使用	可以	不能，由 MCU 硬件决定	不能，由 MCU 硬件决定
处理器占用率	最高	较低	最低
当使用 2 线方式(CS 直接接地)时，通讯过程中是否需要屏蔽其它执行时间超过 2 个扫猫周期的中断	需要屏蔽	不需要	不需要

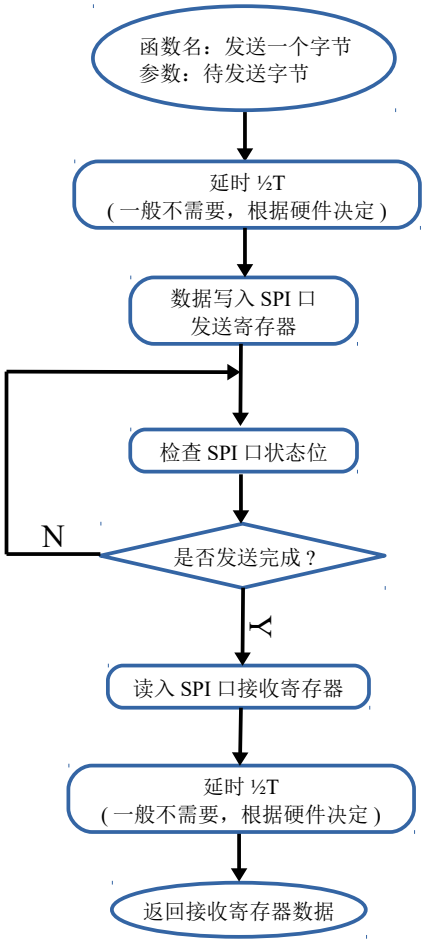
因为数字显示往往是系统中对实时性要求最低的、同时也是优先级最低的任务，并不需要使用中断来做后台的 SPI 接口控制，因此，一般建议使用硬件 SPI 接口+查询方式的方案。

软件模拟方式流程图：



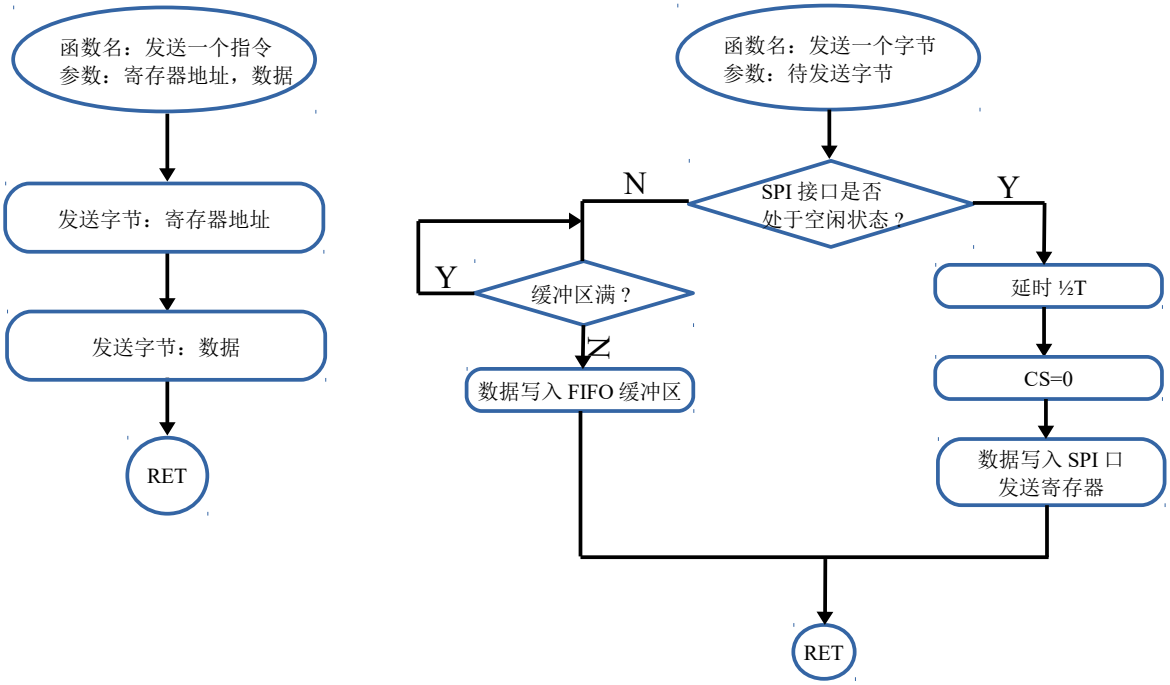
硬件 SPI 接口+查询方式流程图：



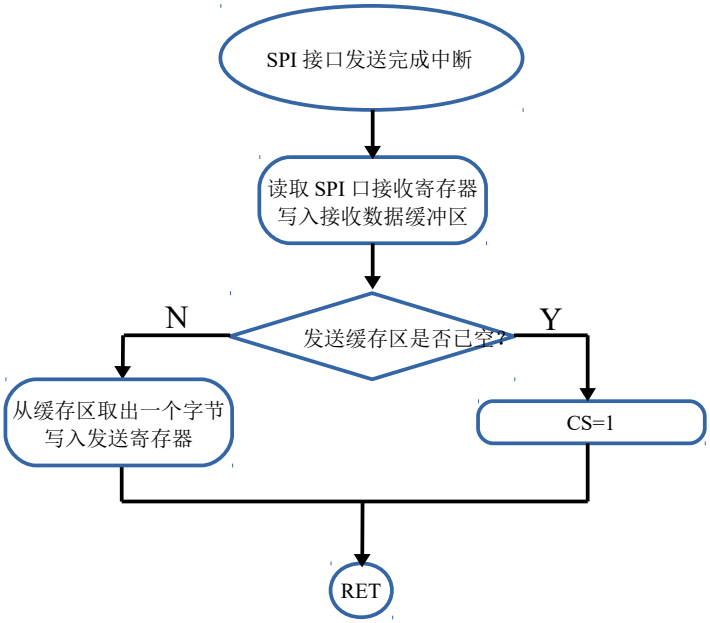


硬件 SPI 接口+中断方式流程图：

(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 ½T 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，同样原因，在“发送一个字节”子程序中，无法返回读入的键盘映射数据。因此在使用中断方式通讯时，在设置发送缓冲区的同时，还需要一个接收缓冲区，用来存放接收到的键盘映射数据。程序在 SPI 口发送完成中断处理子程序中，首先将接收寄存器的内容写入接收缓冲区，然后再将下一个待发送数据写入发送

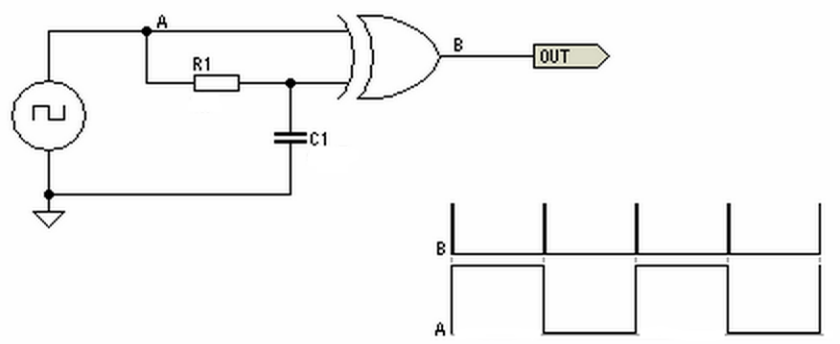
寄存器。SPI 接口缓冲区的使用增加了键盘处理的难度。

键盘处理

BC7277 具有一按键状态变化指示输出 KEY，当键盘的状态发生变化时，KEY 引脚的电平会发生翻转，因此监视 KEY 引脚的变化，是得知键盘事件的最佳办法。不过用户需要注意的是只有 KEY 的电平变化本身代表键盘状态的变化，而其电平维持在高电平或低电平时，只表示键盘状态没有变化，并不表示“无键按下”或“有键按下”。

键盘服务的最简单方式，如果不使用中断，则在程序主循环中不断检查 KEY 引脚的状态，如果发现其产生了变化，则在下一个 BC7277 命令（通常在键盘处理程序中单独发送一个伪指令 0xff, 0xff）发送同时，可以获取当前的键盘映射。因为键盘映射最短的变化周期是一个扫描周期（约 16ms），而人按下键盘的持续时间，一般最短在几百 ms 的时间，因此，只要轮询 KEY 引脚的时间短于这个时间，即可以确保不错过任何键盘事件。

有很多用户习惯用外部中断的方式来处理键盘事件，具体应用在 BC7277 时，需要注意以下两方面：首先 KEY 引脚所接的 MCU 中断引脚，需要能同时接受上升沿和下降沿中断，因为每个电平变化，都代表键盘状态的变化。如果所使用的 MCU 不具有上升下降双方向触发的能力的话，则必须在中断服务程序中动态地改变中断触发沿的设置，下降沿触发后，要在中断返回前，将中断设置改变为上升沿触发。如果硬件只具有单方向触发中断的能力，或者不希望用软件的方式，则可以使用一个异或门配合 RC 电路，将电平变化转变为脉冲信号。如下图，脉冲的宽度 T=0.8RC.



其次，需要注意不要让中断服务程序打断主程序对 BC7277 的操作。如果将读取键盘映射的操作放在键盘中断服务程序中，就必须注意，键盘中断发生时，有可能主程序正在进行 BC7277 的读写操作，因为每个指令由两个字节组成，有可能造成主程序刚发送完命令字，程序却进入了中断，中断服务程序又发出了伪指令来读取键盘映射，这时对于 BC7277 而言，相当于上一个命令的数据字节被伪指令替代了，从而造成显示内容的错误。

为避免这种情况发生，我们建议键盘操作以下两种方式之一进行：

1. 在主程序进行 BC7277 操作之前，先暂时屏蔽键盘中断，操作结束后再重新使能，可以避免这种错误的发生。因为写入一个 BC7277 指令所用的时间相比键盘映射的保持时间来说很短（如果采用 62.5kbps 的速率，一个完整指令传送完成大约为不到 0.3ms），所以将中断暂时屏蔽不会影响到正确读取键盘映射。

2. 在键盘中断中，不做键盘映射读取操作，只设置标志位，通知主程序有键盘事件，然后由主程序处理键盘。因键盘事件一般持续时间在几百 ms 数量级，处理器应有足够的时间来读取键盘。

如果 SPI 接口部分使用了缓冲区，以 SPI 接口中断或 DMA 的方式来完成数据的发送的话，则情况会更为复杂。只适宜在键盘中断服务程序中设置键盘事件标志，然后由主程序负责处理键盘的方式。

KEY 引脚电平的改变，有可能发生在指令的传送过程当中，但键盘映射值的更新，只发生在一次指令传送的开始。比如，在一个指令传送到第二个 bit 的时候 KEY 引脚电平发生翻转，此指令传送完成后处理器接收到的键盘映射，是指令开始传送第一个 bit 时的键盘映射，按键事件还没有发生，所读到的键盘映射并没有反映最新的按键，只有在下一个指令传送时（假设这期间键盘状态没有变化），才能读到正确的键盘映射。用户程序应该注意，KEY 引脚变化后 SPI 口上下一个完整指令所收到的，才是最新的键盘映射。

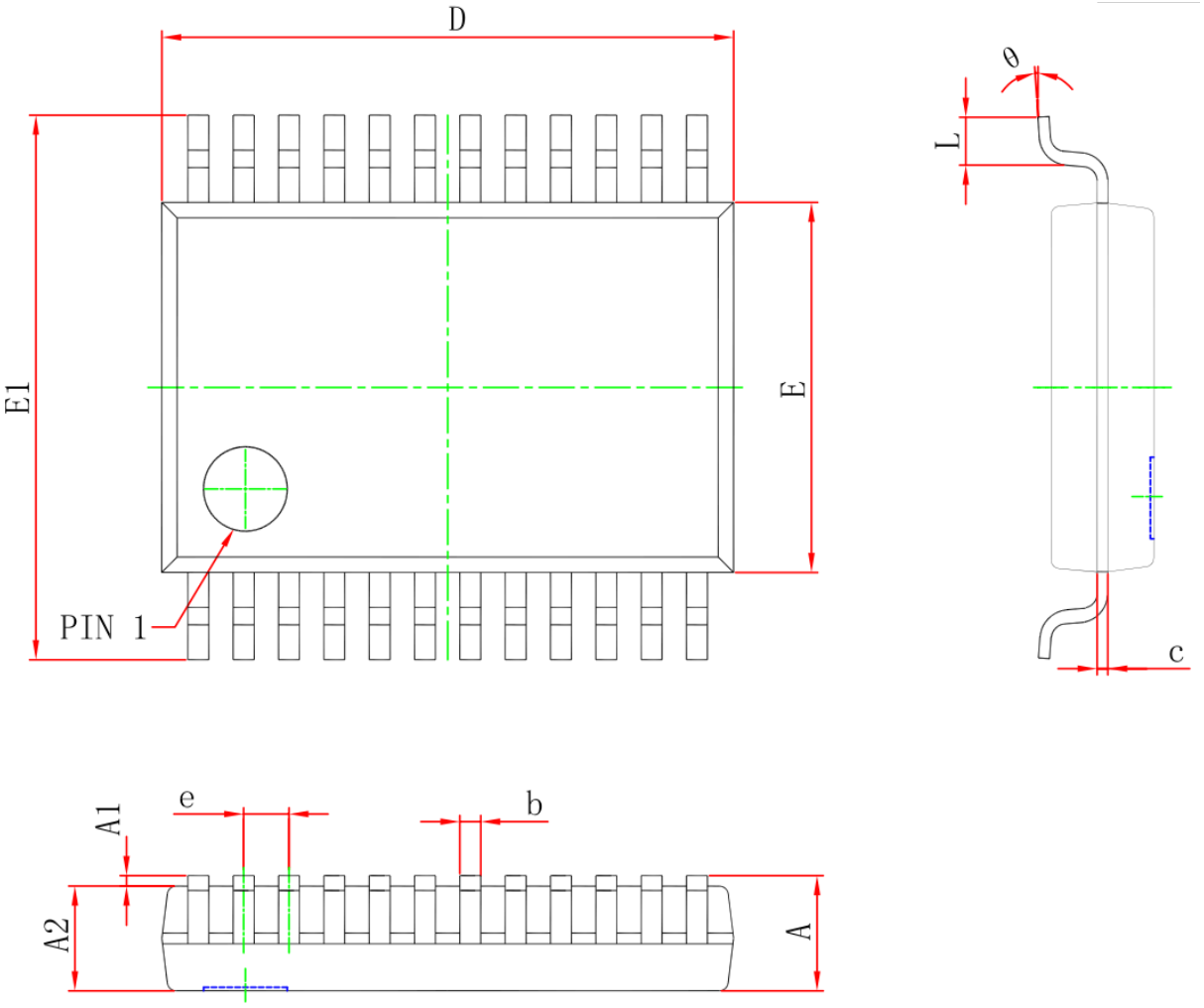
因为每次通讯控制器所得到的，都是反映全部 16 个按键状态的键盘映射，因此可以轻松完成对任意组合，任意数量的组合按键的判断。

对长按键的判断，也很简单。如果在某个或某个组合按键发生后，持续一段时间内 KEY 引脚没有发生变化，则说明按键一直保持着当前的状态，比如 KEY 持续 2s 没有变化，则可判断按键持续保持了 2s.

封装信息

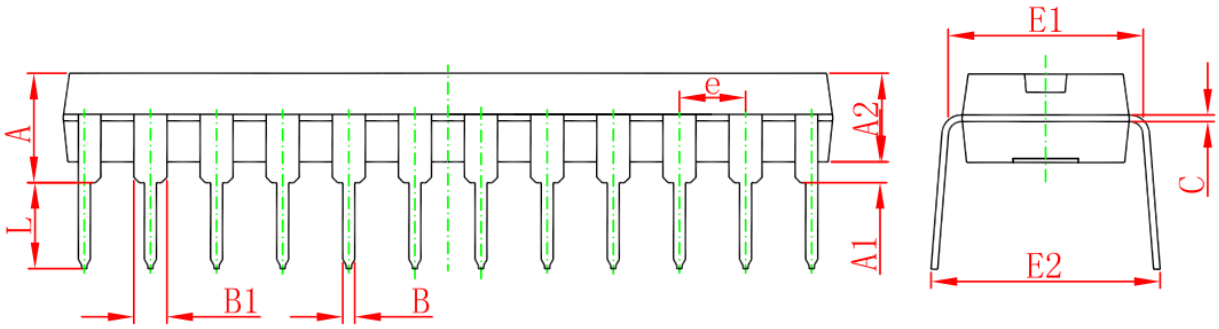
BC7277 有 SDIP24 和 SSOP24 两种封装，封装的尺寸分别如下：

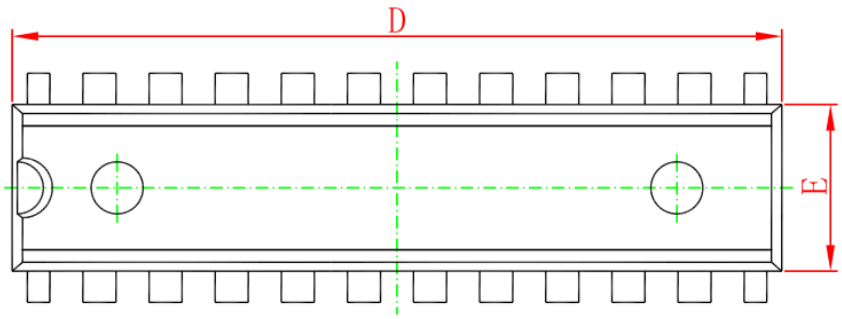
SSOP24 封装尺寸



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	0.01
D	8.000	8.400	0.315	0.331
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°

SDIP24 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	1.270(BSC)		0.050(BSC)	
C	0.204	0.360	0.008	0.014
D	29.250	29.850	1.152	1.175
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	BC7278
芯片封装	DIP20/SSOP20	DIP20/SSOP20	SDIP24/SSOP24	SSOP20
可驱动数码管位数	5	8 或 16	9	4
键盘接口	无	16 键	16 键	16 键
数码管类型	共阴	共阳	共阴	共阴
外接驱动	不需要	三极管和移位寄存器	不需要	不需要
驱动大尺寸数码管	不可以	可以	不可以	不可以