



WTC6508BLV

WTC6508BLV 低电压八通道电容式触摸感应按键芯片

快速浏览(V2.0)

型号区分	WTC6508BLV 为低电压无休眠功能版
按键数量	3-8 个按键
技术机理	电容数字转换技术
按键反应模式	标准的 WTC6508BLV 为 SHIFT 模式，多键触摸时按操作顺序依次反应。
按键感应盘大小	最小 3mm×3mm，最大 30mm×30mm，根据实际需要和面板厚度而定
相邻按键感应盘间隙	标准的 WTC6508BLV 的相邻按键最小间隙 0.5mm,根据实际需要而定，可用作密集键盘。
按键感应盘形状	任意形状多边形,圆形或椭圆形,中间可留孔或镂空.
按键感应盘材料	PCB 铜箔,金属片,平顶圆柱弹簧,导电橡胶,导电油墨,导电玻璃的 ITO 层等
对 PCB 的要求	单面 PCB 板，双面 PCB
面板材质	绝缘材料,如有机玻璃,普通玻璃,钢化玻璃,塑胶,木材,纸张,陶瓷,石材等
面板厚度	0~20mm
按键灵敏度调节方式	改变基准电容 CSEL 的值可对按键的灵敏度进行调节。
有效触摸反应时间	小于 80ms
防水性能	对面板撒水,喷水按键不发生误动 ,面板漫水,积水时触摸按键无异常反应,
抗射频干扰性能	能有效抑制 GSM 手机贴近面板拨打或接听电话，大功率对讲机贴近面板进行对讲操作产生的射频干扰。
工作电压范围	2.4V-5.5V
工作温度范围	-40℃ - +85℃
数据传输接口	两线串行口（一条时钟线，一条数据线）
储存温度范围	-50℃ - +125℃
芯片封装形式	NSOP16 (150MIL)
典型应用	各种家用电器,安防设备,通讯设备,工业控制设备仪器仪表,,娱乐设备，医疗设备,体育设备,玩具等.



WTC6508BLV

目录

1	产品简介.....	4
1.1	技术参数.....	4
1.2	典型应用.....	4
1.3	芯片封装形式.....	4
1.4	WTC6508BLV 的引脚图和引脚定义.....	4
2	触摸感应按键.....	5
2.1	WTC6508BLV 触摸感应按键的技术特点.....	5
2.2	WTC6508BLV 的工作电路图.....	6
2.3	电容传感器（按键感应盘）.....	7
2.3.1	电容传感器（按键感应盘）的材质、形状、面积.....	7
2.3.2	相邻电容传感器（按键感应盘）的间隙.....	7
2.3.3	电容传感器（按键感应盘）到 WTC6508BSI 的引脚连线.....	7
2.3.4	WTC6508BLV 的空置传感器通道.....	8
2.3.5	面板的材质和厚度.....	8
2.3.6	触摸按键板的安装.....	8
2.3.7	灵敏度的设定和灵敏度设定电容 CSEL.....	8
2.3.8	触摸感应灵敏度的软件辅助调节.....	9
2.3.9	通道匹配电阻.....	9
2.3.10	批量生产免调试.....	9
3	WTC6508BLV 蜂鸣器和背光控制信号.....	9
3.1	背光控制.....	9
3.2	蜂鸣器控制.....	10
4	输入输出接口.....	10
4.1	两线串行接口简介.....	10
4.2	读取触摸按键信息.....	11
4.3	设置触摸按键的感应灵敏度.....	12
4.4	软件设置触摸按键感应灵敏度的注意事项.....	13
5	组合键设计.....	14
5.1	标准 WTC6508BLV 的多键组合（SHIFT）工作模式.....	14
6	WTC6508BLV 的电源.....	14
6.1	直流稳压器.....	14
6.2	稳压器组件的放置.....	14
6.3	高噪声条件下的注意事项.....	15
6.4	电源滤波:.....	15
6.5	使用主机的+5V 电源.....	15
7	订货信息:.....	15



专业的触摸感应芯片供应商

WINCOM

<http://www.wincomtech.com>

WTC6508BLV

8	WTC6508BLV 封装图及尺寸	16
9	WTC6508BLV 与 MCU 8051 的接口电路和操作程序范例.....	17



WTC6508BSI 低电压八通道触摸感应按键芯片
规格书(V2.0)

1 产品简介

WTC6508BLV 是一种触摸感应按键芯片，该芯片内部集成了自主设计的 8 通道电容测量电路、自校准电路和 RISC 处理器，为用户提供 8 个电容式触摸感应按键的人机界面方案。

1.1 技术参数

- 满足工业应用规格要求可以为用户提供更可靠的性能，和更宽的应用范围
- 工作电压： $2.4V < V_{cc} < 5.5V$
- 输出电压范围 $GND < V_{out} < V_{cc}$
- 感应厚度(绝缘介质)： 0—20mm
- 有效触摸反映时间： 小于 100ms
- 工作温度： $-40^{\circ}C - +85^{\circ}C$
- 储存温度： $-50^{\circ}C - +125^{\circ}C$

1.2 典型应用

适用于各种厨房设备，音视频设备，空调器，卫浴电器，灯具开关，安全防卫，仪器仪表，便携式播放器，电子玩具及学习机等产品。

1.3 芯片封装形式

WTC6508BLV 采用标准 16PIN NSOP16(150mil)封装。

1.4 WTC6508BLV 的引脚图和引脚定义

图 1 是 WTC6508BLV 的引脚图



WTC6508BLV

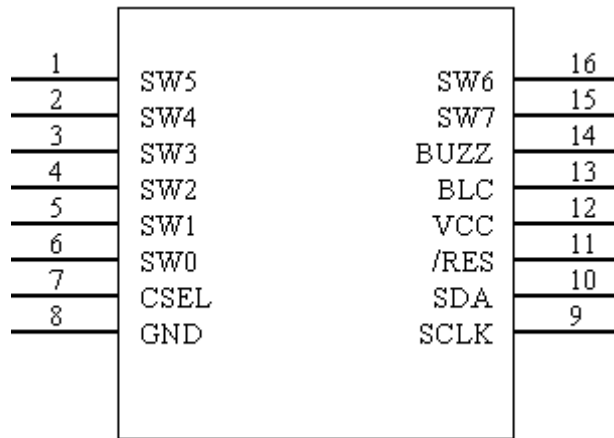


图1 WTC6508BLV 引脚图

表1是WTC6508BLV引脚定义

表1

管脚序号	管脚名称	用法	功能描述
1	SW5	I	电容传感器（感应按键）接口5
2	SW4	I	电容传感器（感应按键）接口4
3	SW3	I	电容传感器（感应按键）接口3
4	SW2	I	电容传感器（感应按键）接口2
5	SW1	I	电容传感器（感应按键）接口1
6	SW0	I	电容传感器（感应按键）接口0
7	CSEL	I	灵敏度调整电容接口
8	GND	I	电源地
9	SCLK	I	数据传送的时钟输入脚
10	SDA	I/O	数据传送的数据输出脚
11	/RES	I	芯片复位脚
12	VCC	I	正电源输入
13	BLC	O	接近背光控制脚
14	BUZZ	O	蜂鸣器控制脚
15	SW7	I	电容传感器（感应按键）接口7
16	SW6	I	电容传感器（感应按键）接口6

2 触摸感应按键

2.1 WTC6508BLV 触摸感应按键的技术特点

- WTC6508BLV 通过 8 个独立的电容传感器输入通道来侦测触摸按键感应盘上电容的变化，通过一系列高效的算法来识别手指的有效触摸，用户可以通过 2 线串行接口读取按键数据。



WTC6508BLV

- **采用特殊的防水设计：**键盘不仅可以防溅水、漫水而且可以在完全被水淹没后正常使用，不同于目前一般的感应按键面板溅水、漫水时容易误动，积水后反应迟钝。或手指按下后出现不准确的误动。
- **优良的抗电磁干扰能力：**在家电和一般应用场合的产品上应用时。用户使用单面PCB板即可获得良好的抗射频信号干扰的能力。可以轻松对抗包括GSM手机在内的绝大多数射频干扰源对感应按键的干扰。
- **独特的线长自修正功能：**目前市场上的同类产品，普遍存在着感应灵敏度随着感应按键到IC引脚的连线长度不同而有很大差异的现象。我们独创的线长自修正技术可以自动修正这种差异。用户不需复杂的调整就能获得整个感应面板各个按键几乎一致的感应灵敏度。

2.2 WTC6508BLV 的工作电路图

WTC6508BLV 的外围电路很简单，只需少量阻容元件即可工作，图 2 是带有推荐串行接口电路的 WTC6508BLV 应用电路图。

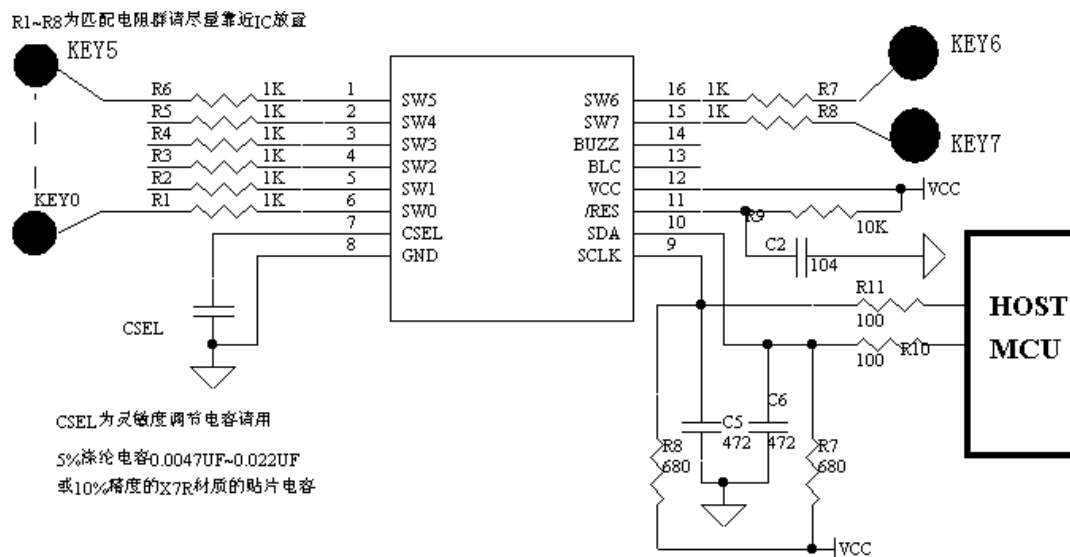


图 2;WTC6508BLV 的工作电路图



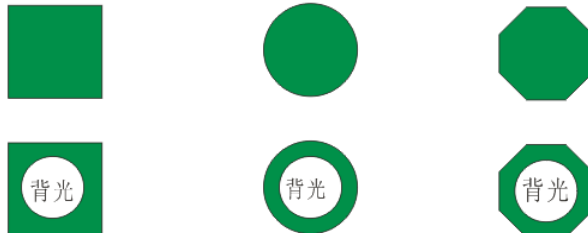
2.3 电容传感器（按键感应盘）

2.3.1 电容传感器（按键感应盘）的材质、形状、面积

有一定平面面积的导电物体，如金属箔、金属片、导电油墨、金属棉、导电橡胶，导电玻璃上的 ITO 镀膜等，都可以作为按键感应盘，在家用电器中一般采用金属弹簧（图 3）或 PCB 上的铜箔（图 4）。按键感应盘可以是圆形、椭圆形、多边形或任何有一定面积的几何图形，一般情况下，每个感应盘的面积应尽量保持相同，以确保灵敏度相同。加大感应盘面积有助于提高电容感应的穿透能力，提高灵敏度



图 3: 金属弹簧做电容传感器（按键感应盘）



按键感应盘可以是实心或中空矩形、圆形、多边形

图 4: PCB 铜箔做电容传感器（按键感应盘）

2.3.2 相邻电容传感器（按键感应盘）的间隙

WTC6508BLV 采用了相邻按键抑制技术，可以防止相邻按键的误动作（串键）。用户可以设计成间距较小（间距不小于 1mm）的密集键盘。

2.3.3 电容传感器（按键感应盘）到 WTC6508BLV 的引脚连线

电容传感器与 WTC6508BLV 的引脚间的连线应尽量短和细（0.1~0.3mm），最好能将 WTC6508BLV 放置在按键板上。连线的背面和周围 0.5mm 不要铺铜和放置其它回路，以保证传感器有好的灵敏度并避免误触发。



2.3.4 WTC6508BLV 的空置传感器通道

WTC6508BLV 要求必须至少使用 3 个触摸按键来保证芯片的稳定工作, WTC6508BLV 应用在少于 8 个键的场合时 SW7~SW0 会有空置不用的传感器输入通道。空置输入通道只需要简单的悬空, 空置输入通道不能加任何上拉或下拉电阻。

如果需要使用 3 个以下的触摸按键, 请和我们联系以获得相应的技术支持。

2.3.5 面板的材质和厚度

面板必须采用绝缘材料, 如玻璃、塑胶、亚克力等。不同的面板材质具有不同的介电常。一般情况下同一厚度的不同材质的面板, 介电常数越大, 触摸感应灵敏度越高。介电常数越小, 灵敏度越低。

以下是几种常用面板的介电常数

普通玻璃(或钢化玻璃): 7

亚克力(有机玻璃): 3.5

空气: 1

这就是在安装触摸板时, 一定要消除空气间隙的原因。

对于同一种材质的面板, 面板越厚, 按键灵敏度降低, 面板越薄, 灵敏度越高。

2.3.6 触摸按键板的安装

电容传感器(按键感应盘)和面板背面必需牢固贴合以消除间隙, 如果面板背面与按键感应盘结合处有凹凸, 可用绝缘的胶状填充物(如绝缘硅脂)将接触面填平, 以保证与面板与按键感应盘之间的结合面无空气间隙。

2.3.7 灵敏度的设定和灵敏度设定电容 CSEL

在设计好触摸感应按键板, 选定了面板材质和厚度并且确定了安装方式以后, 只需调节 1 个电容 Csel 的值即可改变所有通道的灵敏度以适应不同厚度的面板, 从而达到最佳的触摸感应效果。通过选择适当 Csel 的电容值, 可使 WTC6508BSI 适应 0-20mm 的不同厚度的绝缘面板。隔离介质越厚使用的 Csel 容量越大, 一般建议在 0.0047UF 和 0.022UF 之间由小到大地选择合适的电容, 在此范围内电容值越大灵敏度越高。

CSEL 是灵敏度的基准, CSEL 的材质和精度对于工作稳定性十分重要, 切忌使用稳定性差和温漂过大的电容。WTC6508BLV 的 CSEL 要求使用 10%或以上精度的涤纶电容、NPO 材质电容或 X7R 材质贴片电容。

加大感应盘面积有助于提高电容感应的穿透能力, 提高灵敏度。



2.3.8 触摸感应灵敏度的软件辅助调节

应用于对灵敏度有较苛刻要求的场合，如果找不到最满意的 CSEL 来实现所要求的灵敏度时，还可以固定 CSEL 的值后，用软件通过串行接口对触摸按键的灵敏度进行辅助调节，以达到最佳的触摸感应灵敏度。用软件辅助调整触摸按键的灵敏度的方法请阅读第 4 章的相关内容。

触摸按键感应灵敏度软件辅助调节为 32 个灵敏度等级。相应的级数数据为 1~32，由主控制器通过串行接口写入 WTC6508BLV 的灵敏度级数缓冲区。级数越高触摸按键的灵敏度就越高。但设定的级数数据如果超过了 32 或为 0。本次设置将无效，仍然会使用以前的灵敏度参数。

WTC6508BLV 初次上电后灵敏度内部自动设定为第 29 级，在绝大多数情况下，用户不必使用灵敏度软件辅助调节，只需调整 CSEL 的大小就能得到满意的灵敏度。

2.3.9 通道匹配电阻

每个电容传感器通道串接的 1K 的电阻可以使 WTC6508BLV 工作在最佳的电容测量状态，在 PCB layout 时务必将 CSEL 和匹配电阻尽量贴近 IC 放置，一般不需要调整匹配电阻的阻值，在一些特殊情况下，可以通过调节某个测量通道上的匹配电阻的阻值，来改变该通道的触摸感应灵敏度。加大匹配电阻，可使灵敏度降低，减小匹配电阻值，可使得灵敏度增高。特别注意：通过调整 CSEL 的容值已经容易得到最佳的触摸感应灵敏度，因此一般不需要通过改变匹配电阻值来调整灵敏度。

2.3.10 批量生产免调试

在研发和实验阶段确定灵敏度设定电容 Csel 的值后系统便无需校准。系统可以自动克服由于静电放电，电磁干扰，温度变化，湿气和污染物在表面堆积带来的干扰，提供良好的精确性和各种环境下的操作一致性。可以让产品进行长途运输并能在各种环境下使用。独特的补偿算法和高强度的抗干扰设计可以保证产品长期工作时不会出现误动现象。生产免调试，生产效率极高。

3 WTC6508BLV 蜂鸣器和背光控制信号

3.1 背光控制

WTC6508BLV 的 BLC 脚可以作为触控板的背光控制信号输出。当检测到手指接近感应盘时 BLC 输出高电平，当手指离开触控板后 9 秒钟后 BLC 恢复低电平。BLC 输出高电平时



WTC6508BLV

可以提供 4mA 的源电流驱动。如果 LED 背光所需电流超过 4mA 需外加驱动电路以免损坏 IC。

3.2 蜂鸣器控制

当检测到感应盘（sense element）上有有效触摸发生在 80ms 内 WTC6508BLV 的 BUZZ 脚输出 50ms 的低电平信号，客户可以利用这个信号来实现**按键唤醒功能**，还可以同时外接一个三极管推动一个直流蜂鸣器实现触摸按键的声音指示。

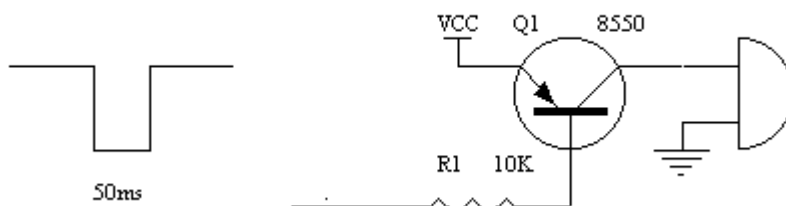


图 5

BUZZ 是触摸按键被触摸后输出的一个指示脉冲。并不表示键的触摸时间。触摸按键键值会一直保持到手指离开，只需要在程序中持续读取并判断就可以得到按键的实际状态。

4 输入输出接口

4.1 两线串行接口简介

WTC6508BLV 采用两线串行接口和主控 MCU 进行通信，主控 MCU 可以通过 SCLK 和 SDA 读取触摸按键的开关信息，并可以通过串行接口设置触摸按键的感应灵敏度。SCLK 是串行接口的时钟信号，SDA 是串行接口的数据信号。

SCLK 的速度

因为触摸芯片处理串行数据需要一定的延时时间，并且为了在总线上有偶发的噪声脉冲的情况下 WTC6508BSI 能够有自动总线复位能力。**建议主控 MCU 产生的 SCLK 方波速度为 20KHZ~2KHz 之间，并且每隔 15ms 以上的时间进行一次操作。**这样可以准确稳定的读写触摸芯片的串行总线。

建议主控 MCU 与 WTC6508BBLV 的串行接口采用以下电路，如图 6，以尽量减少串行接口上的噪声，尤其是在信号线较长的情况下。



WTC6508BLV

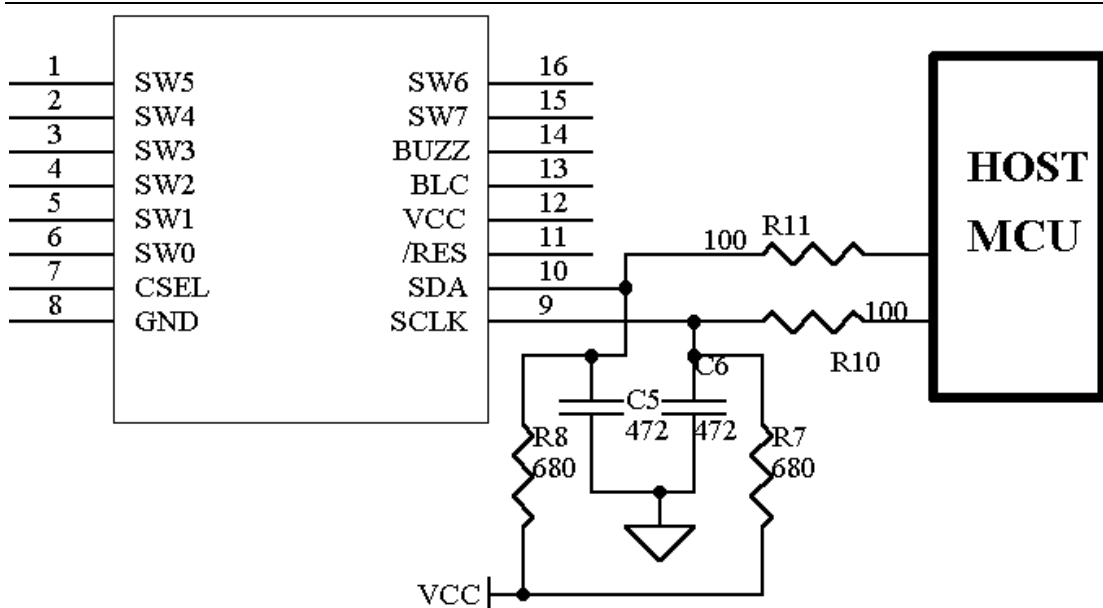


图 6: MCU 与 WTC6508BLV 的串行接口电路

R7, R8 为串行总线的上拉电阻, 因为触摸芯片的 SCLK 和 SDA 平时是高阻输入状态, 所以外部需要加上拉电阻。

R9,R10 和 C5,C6 构成两个 RC 滤波器, 用于滤除 SCLK 和 SDA 上的“毛刺”噪声。

触摸芯片的 SCLK 脚总是保持高阻的输入状态, 触摸芯片的 SDA 脚在主控 MCU 读取按键信息时是输出状态, 其他任何时候都保持高阻的输入状态。

4.2 读取触摸按键信息

图 7 是主控 MCU 读取按键信息时触摸芯片的输出时序图

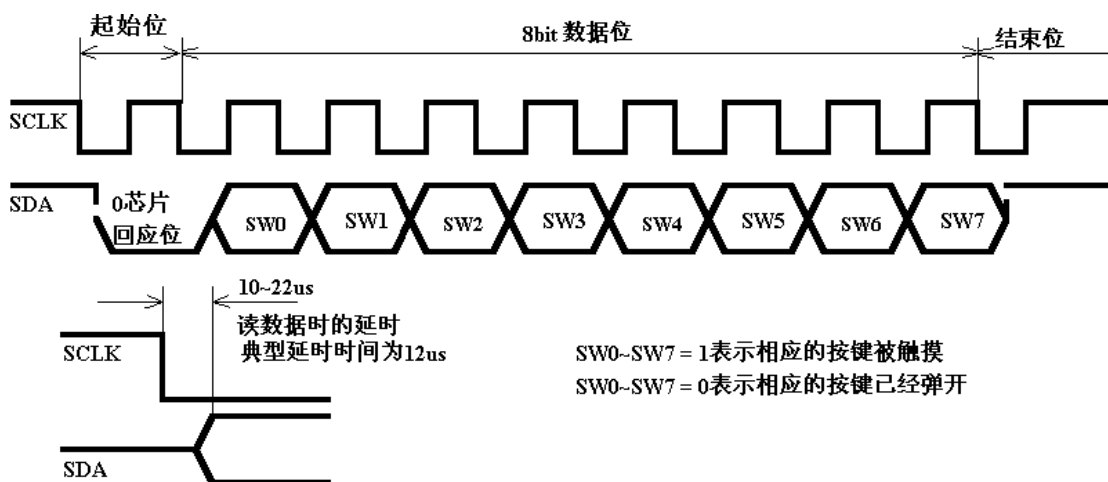


图 7: 主控 MCU 读取按键信息时序

完成一次完整的读取按键信息操作, 主控 MCU 需要在 SCLK 上产生 10 个低电平脉冲, 分为 1 个起始位, 8 个触摸按键数据位, 1 个结束位。

1: 产生起始位

触摸芯片的 SDA 脚平时为高阻的输入状态, 当主控 MCU 也将 SDA 置为输入状态时, 外部上拉电阻将 SDA 上拉为高电平, 主控 MCU 将 SCLK 置为低电平, 在 **10~22us 的延时后** 触摸芯片会将 SDA 脚置为输出态并输出低电平作为开始传输按键信息的回应信号。主控 MCU 将 SCLK 置高完成起始位设定

2: 读取 8 bit 触摸按键信息位

主控 MCU 再次将 SCLK 置为低电平, 在 **10~22us 的延时后** 触摸芯片会将 SW0 上连接的感应按键的状态放到 SDA 上。如果感应按键没有被触摸或已经弹开 SDA 为“0”, 感应按键被触摸 SDA 为“1”。主控 MCU 读取 SDA 状态后将 SCLK 置高

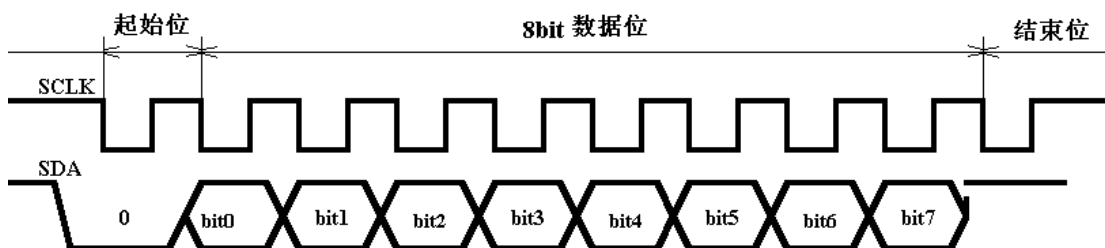
依此类推主控 MCU 在 SCLK 上产生 8 个方波读取 8bit 的触摸按键信息。

3: 产生 1 个结束位

MCU 读取 8bit 的触摸按键信息完成后再次产生一个脉冲以产生 1 个结束位, 触摸芯片收到结束位后将 SDA 脚恢复置为高阻的输入状态。主控 MCU 完成一次读取操作。

MCU 不产生结束位的话触摸芯片将保持 SDA 的输出状态, 这会影响芯片的工作。

4.3 设置触摸按键的感应灵敏度



bit0~bit7是需要设定的灵敏度数据Subtle的0~7位
Subtle 必须小于或等于1FH

图 8: 主控 MCU 设定触摸按键灵敏度操作时序

完成一次用软件设置触摸按键感应灵敏度的操作, 主控 MCU 需要在 SCLK 上产生 10 个低电平脉冲, 分为 1 个起始位, 8 个灵敏度数据位, 1 个结束位。



1: 产生起始位

触摸芯片的 SDA 脚平时为高阻的输入状态，当主控 MCU 先将 SDA 置为低电平，然后再将 SCLK 置为低电平，在 **10~22us 的延时后**触摸芯片会转入接收数据的读取准备状态。主控 MCU 将 SCLK 置高完成起始位设定

2: 将要设定的灵敏度级数传入灵敏度数据的暂存区

主控 MCU 先将需要设定的灵敏度级数的第 1 位放到 SDA 上，然后将 SCLK 置为低电平，在 **10~22us 的延时后**触摸芯片会将 SDA 上的数据读入暂存区的第 1 位。主控 MCU 将 SCLK 置高

以此类推，主控 MCU 将后面 7 位数据放到 SDA 上然后在 SCLK 上产生 7 个低电平脉冲，完成将数据传入暂存区的操作。

3: 产生 1 个结束位

完成上述传输后主控 MCU 将 SDA 置为输入状态，同时在 SCLK 上产生一个低电平脉冲，以完成结束位的设定。

WTC6508BLV 的串行输入电路采用双缓冲结构，主控 MCU 将数据写入的 8bit 期间，电路将串行输入的数据送入独立的串入暂存区，触摸芯片的灵敏度不会变化。只有当主控 MCU 在 SCLK 上产生结束位后触摸芯片才会将触摸芯片重新复位并且使用本次设定的灵敏度级数重新设置内部相关参数。

4.4 软件设置触摸按键感应灵敏度的注意事项

主控 MCU 每次设定灵敏度后触摸芯片都会重新复位，复位需要 50ms 的时间。在触摸芯片重新复位的 50ms 时间内，不论读写都不会得到正确的结果。所以主控 MCU 设定灵敏度完成后必须等待 50ms 以上的时间才能对 WTC6508BSI 的串行接口进行读写。

建议主控 MCU 在程序中不要频繁的对触摸芯片的灵敏度进行设定。只要在程序初始化时设定一次就可以了。

触摸按键感应灵敏度分为 32 级，相应的级数数据为 1~32。级数越高触摸按键的灵敏度就越高。但设定的级数数据如果超过了 32 或为 0，本次设置将无效，无效设置发送结束位后触摸芯片不会重新复位，也不会进行内部参数的调整，仍然会使用以前的灵敏度参数。

触摸芯片初次上电后灵敏度内部自动设定为第 29 级，用户也可以不使用软件调整灵敏度。直接使用芯片的默认参数。

详细的 DEMO 程序请参考第 7 章的内容，也可以参考我们提供的 C 语言 demo 程序源代码

5 组合键设计

5.1 标准 WTC6508BLV 的多键组合 (SHIFT)工作模式

使用 WTC6508BLV 时, 如果用户先后按下多个按键不释放, 则多个按键都能依次作出反应。系统设计者可以据此设计出多种按键组合操作功能。

6 WTC6508BLV 的电源

6.1 直流稳压器

WTC6508BLV 测量的是电容的微小变化, 要求电源的纹波和噪声要小, 要注意避免由电源串入的外界强干扰。尤其是应用于电磁炉, 微波炉时, 必须能有效隔离外部干扰及电压突变, 因此要求电源有较高稳定度。建议采用如图 9 所示 78L05 组成的稳压线路。C2 容量会影响断电后的放电时间, 太大的容量需要较长的复位时间。圈中为推荐值, 不同的应用可灵活调整。图 9 是电源稳压电路

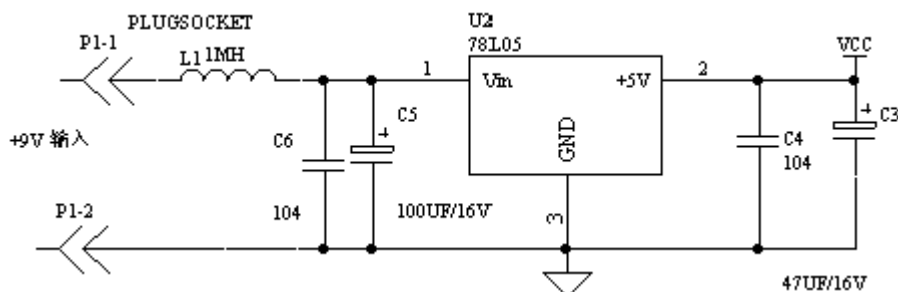


图 9: 直流稳压电路

6.2 稳压器组件的放置

PCB LAYOUT 时, 该 78L05 电源组器件必须紧靠 WTC6508BSI Vcc 管脚。

78L05 及外围器件与 WTC6508BLV 必须放置在同一电路板上, 并集中放置。杜绝电源连接线过长带来噪声。

接地 需将图中器件的共地单独连接成一独立群, 再由一个点连接到整机的共地。(使用星形一点接地)

6.3 高噪声条件下的注意事项

在高噪声环境应用时，应避免高压(220V)、大电流、高频率操作的主板与触摸电路板上下重叠安置。如无法避免，应尽量远离高压大电流的器件区域或在主板上加屏蔽。

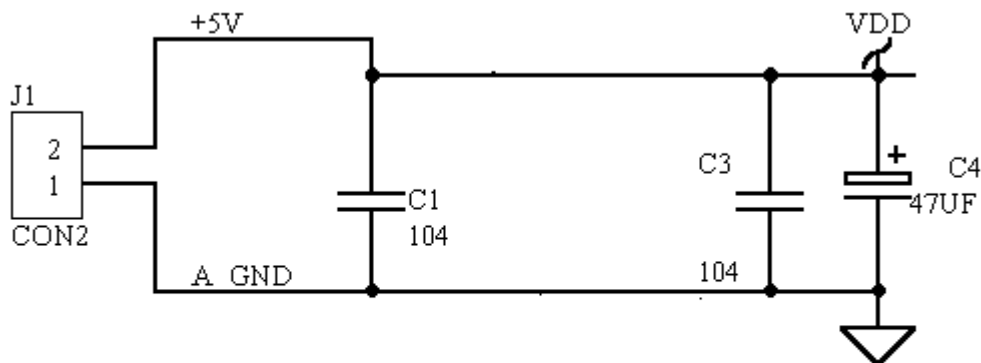
6.4 电源滤波:

在 PCB 排版时，建议预留电感 L1(1MH)焊盘，但是一般的非特殊应用并不需要此电感。用户也可取消它。

输入 78L05 的电源，需注意纹波的大小。勿使纹波的彼谷低于 DC 9V。

6.5 使用主机的+5V 电源

如果用户直接使用主机的 5V 电源，需在模组或感应芯片组的电源前加如图 9 所示的电源滤波电路。PCB layout 的要求同上一电路。



电路的模拟和数字部分的电源和地请分开用星型接法连接
电容的排列顺序请按原理图标明的顺序，不要随意排列

图 10 电源滤波电路

注意事项:

上图电路请按照图上的顺序连接滤波电路的 104 电容和电解电容，104 电容应该接到电解电容的前方才能较好的抑制高频噪声。

7 订货信息:

WTC6508BLV 从型号上加以区分，用户订货时须提供完整的产品型号



WTC6508BLV

产品型号	按键反应模式	应用注意事项
WTC6508BLV	SHIFT 模式，即多键顺序反应	可用作密集键盘

8 WTC6508BLV 封装图及尺寸

Symbol	Dimensions in mm		
	Min.	Nom.	Min.
A	1.35	1.60	1.75
A1	0.10	—	0.25
A2	—	1.45	—
B	0.33	—	0.51
C	0.19	—	0.25
D	9.80	—	10.00
E	3.80	—	4.00
e	—	1.27	—
H	5.80	—	6.20
L	0.40	—	1.27
θ	0.00	—	8.00



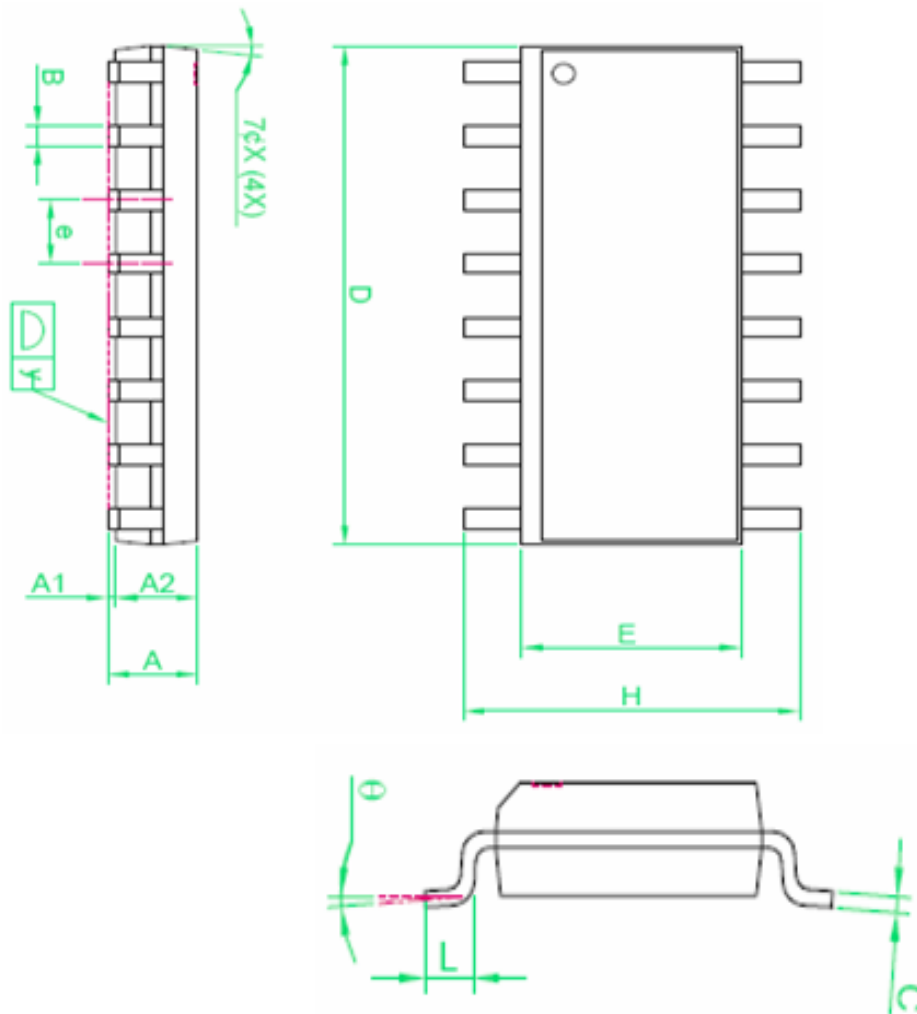


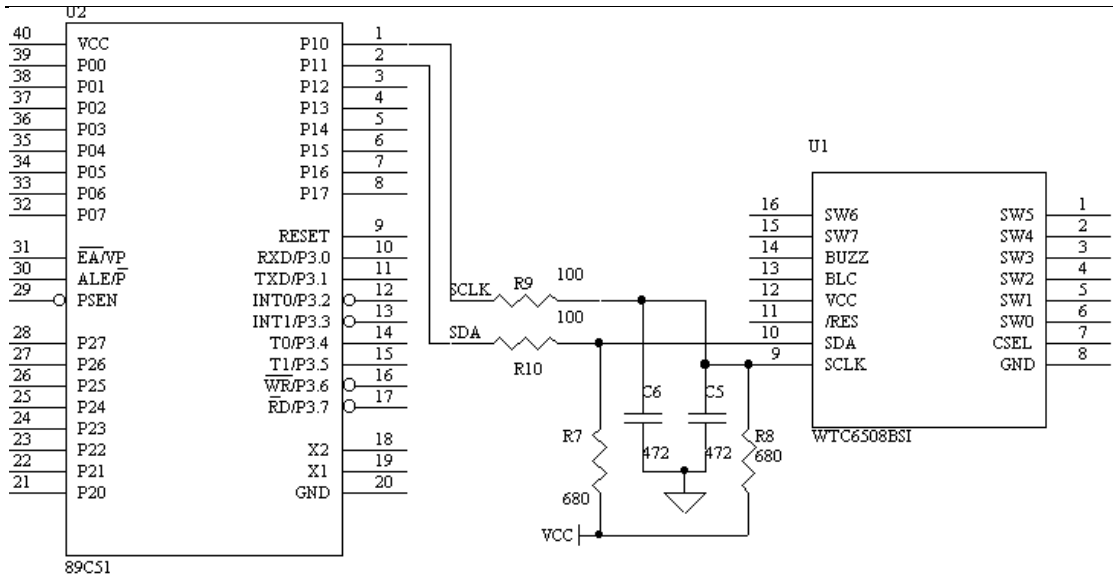
图 11: WTC6508BLV 封装和尺寸图

9 WTC6508BLV 与 MCU 8051 的接口电路和操作程序范例

WTC6508BLV 与 8051 接口的典型应用:



WTC6508BLV



典型操作程序程序对应电路

8051 与 WTC6508BLV 接口的典型操作程序

/*

项目描述:

触摸芯片 WTC6508BLV 的 DEMO 板配套程序

DEMO 板功能:

检测触摸芯片输出的信息, 并点亮相应的 LED 完成显示

MCU: AT89C51

晶振频率: 12MHz

版本信息: V1.0

*/

```
#include <reg51.h>
#include <INTRINS.H>
#define uchar unsigned char
#define uint unsigned int

#define TIMER1_H 0xFC //1ms 定时器初值
#define TIMER1_L 0x17
```

```
//-----主机和触摸芯片的接口-----
sbit SDA = P1^1; //数据线
sbit SCLK = P1^0; //时钟信号
```



WTC6508BLV

```
sbit    LED0    =    P0^0;    //触摸按键对应的状态 LED 灯，低电平时点亮
sbit    LED1    =    P0^1;
sbit    LED2    =    P0^2;
sbit    LED3    =    P0^3;
sbit    LED4    =    P0^4;
sbit    LED5    =    P0^5;
sbit    LED6    =    P0^6;
sbit    LED7    =    P0^7;
//-----the funtion define-----
void    init(void);          //89C51 的初始化程序
void    delay_24us(void);    //延时程序(延时 24us)

uchar   read_key_data(void);    //读触摸按键信息
void    set_subtle_SP(uchar subtle);    //用软件设置触摸芯片的触摸灵敏度

void    disp_key_led(uchar KeyValue); //用 LED 显示相应触摸按键的开关状态
//-----the register define -----

uchar   i,j;                //程序中使用的中间变量

uint    Counter1ms;         //timer1 每 1ms 中断一次对 Counter1ms 加 1
uchar   KeyValue;          //从触摸芯片读取的按键信息

//-----
// 主程序
//-----
void main(void)
{
    init();
    Counter1ms = 0;
    do
    {
    } while(Counter1ms <= 100);    //等待 100ms 确保触摸芯片完成复位
    set_subtle_SP(28);            //设定触摸芯片的触摸灵敏度为 28 级

    Counter1ms = 0;
    do
    {
    } while(Counter1ms <= 50);    //等待 50ms 确保触摸芯片完成重新复位
```



```
while(1)
{
    while(Counter1ms >= 15)    //每 15ms 读一次键值，并完成显示
    {
        Counter1ms = 0;        //15ms 计数器归零

        KeyValue = read_key_data();    //读触摸按键的按键信息

        disp_key_led(KeyValue);    /显示相应的 LED
    }
}

//-----
//功能：从触摸芯片读取触摸按键的信息
//入口：无
//返回值：从触摸芯片读取到的触摸按键的开关信息
//-----
uchar read_key_data(void)
{
    uchar KeyValue;
    TR1 =0;
    i = 0;

    //-----起始位的设定-----
    SDA =1;                //SDA 设置为输入状态,上拉电阻将 SDA 拉高置 1,表示要从
                          //触摸芯片读出触摸按键的按键信息

    SCLK = 0;              //产生串行时钟的下降延信号
    delay_24us();

    if(SDA == 1)          //发送起始位后检测触摸芯片是否发送了确认信号
    {
        //触摸芯片没有将 SDA 设置为"0".表明触摸芯片没有做好发送准备
        SCLK = 1;          //SCLK 重新置为高电平
        return(0);         //函数直接返回 0 表示没有读到有效的键值
    }
    SCLK = 1;              //产生串行时钟的上升延信号
    delay_24us();
    //-----读取触摸按键的开关信息（8 bit）-----
    do
```



```
{

    KeyValue >>= 1;
    SCLK = 0;          //产生串行时钟的下降延信号
    delay_24us();
    if(SDA == 1)      //读取 SDA 上的数据
    {
        KeyValue |= 0x80;
    }
    else
    {
        KeyValue &= 0x7F;
    }
    SCLK = 1;          //产生串行时钟的上升延信号
    delay_24us();
    i++;
}
while(i < 8);        //读 8bit 数据
//-----//发送结束位（总线恢复）
SCLK = 0;            //产生串行时钟的下降延信号
delay_24us();
SCLK = 1;            //产生串行时钟的上升延信号
delay_24us();
TR1 = 1;
return(KeyValue);   //返回读取的触摸按键信息
}

/*-----
//功能： 设置触摸芯片的触摸灵敏度
//入口： subtle: 准备设定的触摸灵敏度级数(1~32)
//      subtle 的值在 1~32 内级数越高灵敏度越高.
//      如果 subtle > 32 或 subtle =0 本次设定操作无效.触摸芯片保持灵敏度不变
//返回值： 无
-----*/
void set_subtle_SP(uchar subtle)      //用软件设置触摸芯片的触摸灵敏度
{
    TR1 = 0;
    //-----//发送 SDA 总线写数据的起始位-----
}
```



WTC6508BLV

```
SDA = 0;          //起始位置 0,表示要向触摸芯片写入灵敏度数据
SCLK = 0;        //产生串行时钟的下降延信号
delay_24us();
SCLK = 1;        //产生串行时钟的上升延信号
delay_24us();

//-----向触摸芯片写入需要设定的灵敏度数据-----
i = 0;
do
{
    if((subtle & 0x01) == 0) {SDA = 0;} //准备写入的数据
    else                      {SDA = 1;}
    SCLK= 0;                  //产生串行时钟的下降延信号
    delay_24us();
    SCLK = 1;                //产生串行时钟的上升延信号
    delay_24us();
    subtle >>= 1;
    i++;
}
while(i <8);              //写入 8 位数据
//-----产生结束位,触摸芯片用新的灵敏度级数重新设定参数并重新复位-----
SDA = 1;                //将 SDA 置为输入状态
SCLK = 0;              //产生串行时钟的下降延信号
delay_24us();
SCLK = 1;              //产生串行时钟的上升延信号
delay_24us();
TR1 = 1;
}

/*-----
//功能:手指触摸按键时点亮被触摸按键上方的指示 LED,手指从触摸按键上移开后熄灭相应
按键上方的指示 LED
//入口:从触摸芯片读取到的按键信息
//返回值:无
-----*/

void disp_key_led(uchar KeyValue)
{
    if((KeyValue & 0x01) != 0) {LED0 = 0;} //KEY0 被按下点
                                //亮相应的 LED
    else                      {LED0 = 1;} //KEY0 弹起,熄灭
```



```
if((KeyValue & 0x02) != 0){ LED1 = 0;} //KEY1 被按下
else { LED1 = 1;} //KEY1 弹起

if((KeyValue & 0x04) != 0){ LED2 = 0;} //KEY2 被按下
else { LED2 = 1;} //KEY2 弹起

if((KeyValue & 0x08) != 0){ LED3 = 0;} //KEY3 被按下
else { LED3 = 1;} //KEY3 弹起

if((KeyValue & 0x10) != 0){ LED4 = 0;} //KEY4 被按下
else { LED4 = 1;} //KEY4 弹起

if((KeyValue & 0x20) != 0){ LED5 = 0;} //KEY5 被按下
else { LED5 = 1;} //KEY5 弹起

if((KeyValue & 0x40) != 0){ LED6 = 0;} //KEY6 被按下
else { LED6 = 1;} //KEY6 弹起

if((KeyValue & 0x80) != 0){ LED7 = 0;} //KEY7 被按下
else { LED7 = 1;} //KEY7 弹起

}

//-----
void init(void)
{
    EA = 1; //open the gloabe interrupt enable
    EX0 = 0; //disable expend 0 interrupt
    EX1=1; //enable expend 1 interrupt for remote
    ET0 = 1;
    ET1 = 1;
    ES = 0;
    //IT0 = 0; //level tigger
    //IT1 = 0; //level tigger
    IT0 = 1; //edge tigger
    IT1 = 1; //edge tigger
    //TMOD &= 0xF0;
    TMOD= 0x11; //timer0, timer1 work as 16 bit timer
}
```



WTC6508BLV

```
TH0 = 0x00;
TL0 = 0x00;
TH1 = TIMER1_H;
TL1 = TIMER1_L;
TR0 = 0;
TR1 = 1;                                //timer1 start work
}

//-----
void delay_24us(void)
{
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
}

//-----
void timer1(void) interrupt 3 using 0    //8051 的 timer1 中断服务程序
{
    TH1 = 0xFC;    //1ms 定时器重新设置初值
    TL1 = 0x17;
    TR1 = 1;
    Counter1ms++;    //1ms 计数器加一
}
```