



WTC6506D32

WTC6506D32 六通道触摸感应按键和 LED 显示驱动芯片
规格书(V1.1)

目录

| | | |
|-------|-------------------------------------|----|
| 1 | 产品简介..... | 2 |
| 1.1 | 技术参数..... | 2 |
| 1.2 | 典型应用..... | 2 |
| 1.3 | 芯片封装形式..... | 2 |
| 1.4 | WTC6506D32 的引脚图和引脚定义..... | 2 |
| 2 | 触摸感应按键..... | 4 |
| 2.1 | WTC6506D32 触摸感应按键的技术特点..... | 4 |
| 2.2 | WTC6506D32 的工作电路图..... | 4 |
| 2.3 | 电容传感器（按键感应盘）..... | 6 |
| 2.3.1 | 电容传感器（按键感应盘）的材质、形状、面积..... | 6 |
| 2.3.2 | 相邻电容传感器（按键感应盘）的间隙..... | 6 |
| 2.3.3 | 电容传感器（按键感应盘）到 WTC6506D32 的引脚连线..... | 6 |
| 2.3.4 | WTC6506D32 的空置传感器通道..... | 7 |
| 2.3.5 | 面板的材质和厚度..... | 7 |
| 2.3.6 | 触摸按键板的安装..... | 7 |
| 2.3.7 | 灵敏度的设定和 CSEL..... | 7 |
| 2.3.8 | 通道匹配电阻..... | 7 |
| 2.3.9 | 批量生产免调试..... | 8 |
| 3 | 可编程 LED 显示驱动..... | 8 |
| 3.1 | LED 驱动界面的动态扫描方式..... | 8 |
| 3.2 | WTC6506D32 的驱动电流..... | 9 |
| 3.3 | 显示内容缓冲区..... | 9 |
| 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 | 设定 LED 显示内容..... | 12 |
| 5 | WTC6506D32 的电源..... | 13 |
| 5.1 | 直流稳压器..... | 13 |
| 5.2 | 稳压器组件的放置..... | 13 |
| 5.3 | 高噪声条件下的注意事项..... | 14 |
| 5.4 | 电源滤波:..... | 14 |
| 5.5 | 使用主机的+5V 电源..... | 14 |



WTC6506D32

6 WTC6506D32 封装图及尺寸 15
7 WTC6506D32 与 MCU 8051 的接口电路和操作程序范例 15

1 产品简介

WTC6506D32 是一种带 LED 显示界面的触摸感应按键芯片，该芯片内部集成了自主设计的 6 通道电容测量电路、自校准电路和 RISC 处理器、LED 显示驱动电路，为用户提供 6 个电容式触摸感应按键和最多 32 个 LED 的可编程显示的人机界面方案。

1.1 技术参数

- Ø 满足工业应用规格要求可以提供更可靠的性能，和更宽的应用范围
- Ø 工作电压：3.3V<Vcc<5.5V
- Ø 输出电压范围 GND<Vout<Vcc
- Ø 感应厚度(绝缘介质)：0—20mm
- Ø 有效触摸反映时间：小于 80ms
- Ø 工作温度：-40℃—+85℃
- Ø 储存温度：-50℃—+125℃

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

WTC6506D32 采用标准 24PIN SSOP24A(150mil)封装。

1.4 WTC6506D32 的引脚图和引脚定义

图 1 是 WTC6506D32 的引脚图



WTC6506D32

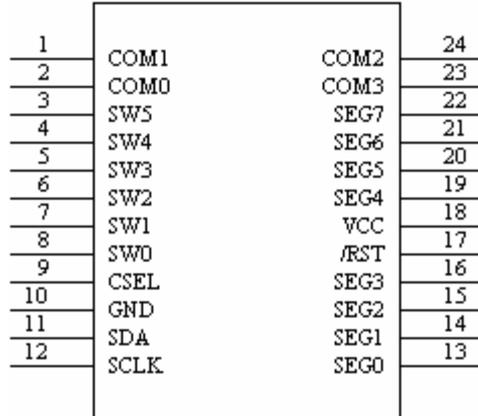


图 1 WTC6506D32 引脚图

表1是WTC6506D32引脚定义

表1

| 管脚序号 | 管脚名称 | 用法 | 功能描述 |
|------|------|-----|----------------|
| 1 | COM1 | O | LED动态驱动的COM1口 |
| 2 | COM0 | O | LED动态驱动的COM0口 |
| 3 | SW5 | I | 电容传感器（感应按键）接口5 |
| 4 | SW4 | I | 电容传感器（感应按键）接口4 |
| 5 | SW3 | I | 电容传感器（感应按键）接口3 |
| 6 | SW2 | I | 电容传感器（感应按键）接口2 |
| 7 | SW1 | I | 电容传感器（感应按键）接口1 |
| 8 | SW0 | I | 电容传感器（感应按键）接口0 |
| 9 | CSEL | I | 灵敏度调整电容接口 |
| 10 | GND | I | 电源地 |
| 11 | SDA | I/O | 数据传送的数据输入和输出脚 |
| 12 | SCLK | I | 数据传送的时钟输入脚 |
| 13 | SEG0 | O | LED 动态驱动的SEG0口 |
| 14 | SEG1 | O | LED动态驱动的SEG1口 |
| 15 | SEG2 | O | LED动态驱动的SEG2口 |
| 16 | SEG3 | O | LED动态驱动的SEG3口 |
| 17 | /RST | I | 芯片复位脚 |
| 18 | VCC | I | 正电源输入 |
| 19 | SEG4 | O | LED动态驱动的SEG4口 |
| 20 | SEG5 | O | LED动态驱动的SEG5口 |
| 21 | SEG6 | O | LED动态驱动的SEG6口 |
| 22 | SEG7 | O | LED动态驱动的SEG7口 |
| 23 | COM3 | O | LED动态驱动的COM3口 |
| 24 | COM2 | O | LED 动态驱动的COM2口 |



2 触摸感应按键

2.1 WTC6506D32 触摸感应按键的技术特点

- Ø WTC6506D32 通过 6 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。
- Ø **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。
- Ø **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。
- Ø **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 WTC6506D32 的工作电路图

WTC6506D32 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 LED 显示电路和串行接口电路的 WTC6506D32 应用电路图，如果需要参考完整带串口电路和 LED 显示的应用电路，请参考我们提供的 DEMO 板的原理图。并根据项目实际情况修改。



WTC6506D32

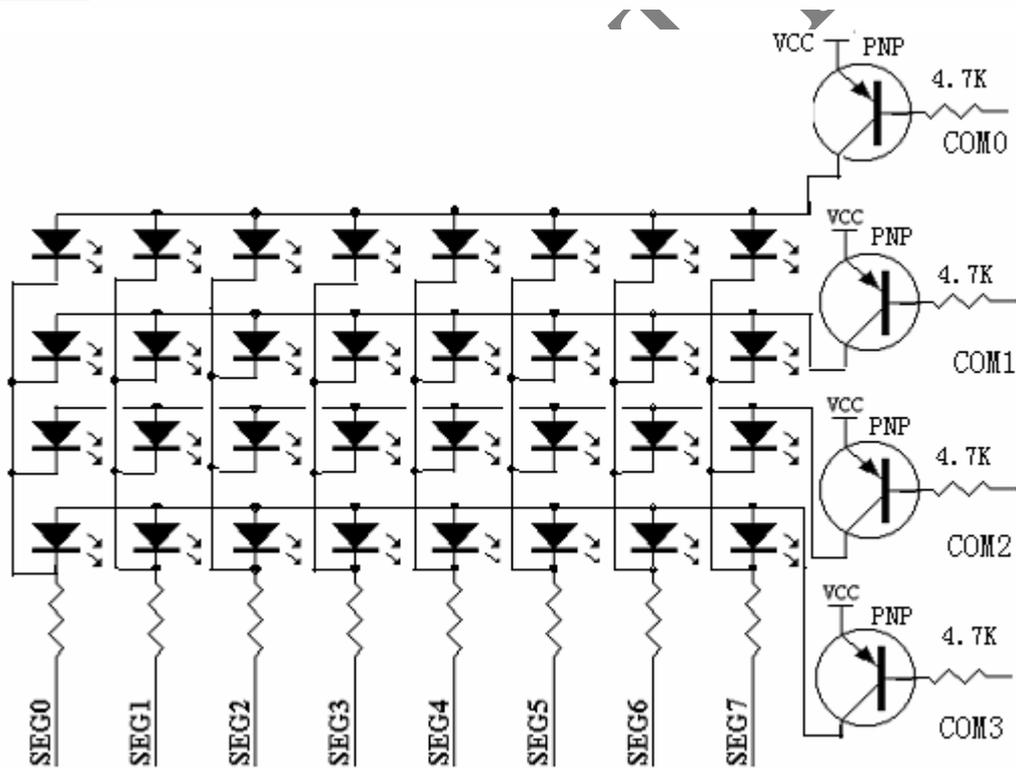
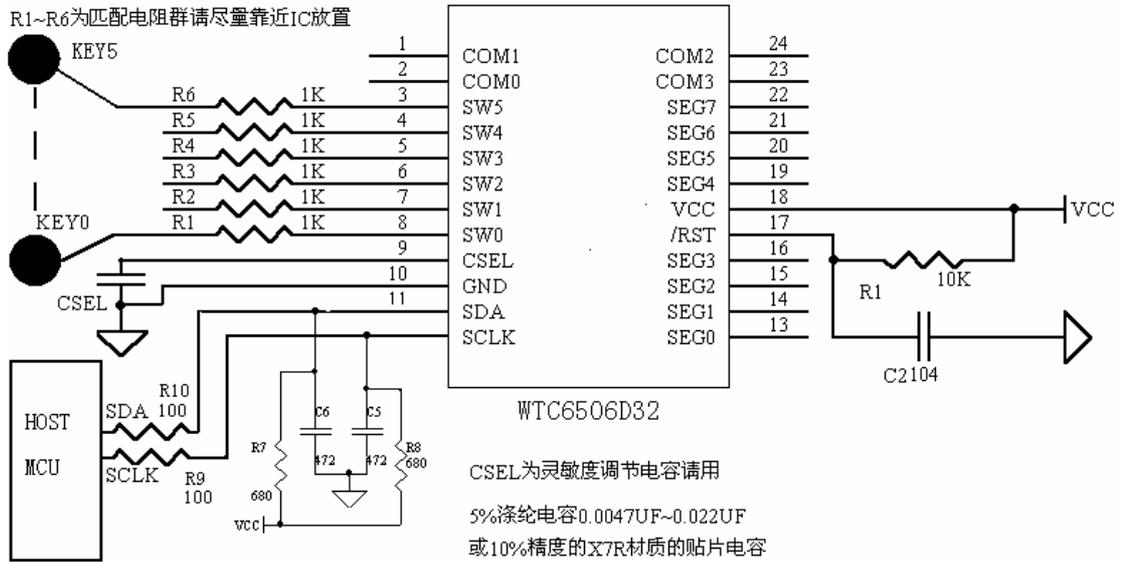


图 2;WTC6506D32 的工作电路图



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的矩形、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

WTC6506D32 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 2mm）的密集键盘。

2.3.3 电容传感器（按键感应盘）到 WTC6506D32 的引脚连线

电容传感器与 WTC6506D32 的引脚间的连线应尽量短和细（0.1~0.3mm），最好能将 WTC6506D32 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 WTC6506D32 的空置传感器通道

WTC6506D32 要求必须至少使用 3 个触摸按键来保证芯片的稳定工作，WTC6506D32 应用在少于 6 个键的场合时 SW5~SW0 会有空置不用的传感器输入通道。空置输入通道只要简单的悬空就可以，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质，如玻璃、塑胶、亚克力等。不同的面板材质具有不同的介电常数，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）和面板背面必须消除间隙牢固贴合，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，并选定了面板材质和厚度，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，达到最佳的触摸感应效果，通过选择适当 Csel 的电容值，可使 WTC6506D32 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。

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。WTC6506D32 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** 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 通道匹配电阻

每个电容传感器通道串接的 1K 的电阻可以为使 WTC6506D32 工作在最佳的最稳定的



WTC6506D32

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。

2.3.9 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 可编程 LED 显示驱动

3.1 LED 驱动界面的动态扫描方式

WTC6506D32 采用动态扫描的方式对芯片内部 4 个 8bit 长度的显示缓冲区的数据内容进行 LED 显示。采用 4 个公用脚和 8 个数据脚的组合最多可实现 32 个 LED 的显示界面。如图 5。

COM0~COM3 脚是 LED 公共端的控制信号输出脚，COM0 在输出低电平时，SEG0~SEG7 同时输出 COM0 的对应的内部显示缓冲区的数据。此时 COM1~3 输出都为高电平。以此类推 COM1~3 输出低电平时，SEG0~SEG7 也同时输出 COM1~3 的对应的内部显示缓冲区的数据。建议用户在 COM0~COM3 上外接 PNP 型的三极管提供 LED 的共阳驱动电流，LED 的阴极通过限流电阻接 SEG0~SEG7 以实现 LED 的正常显示。

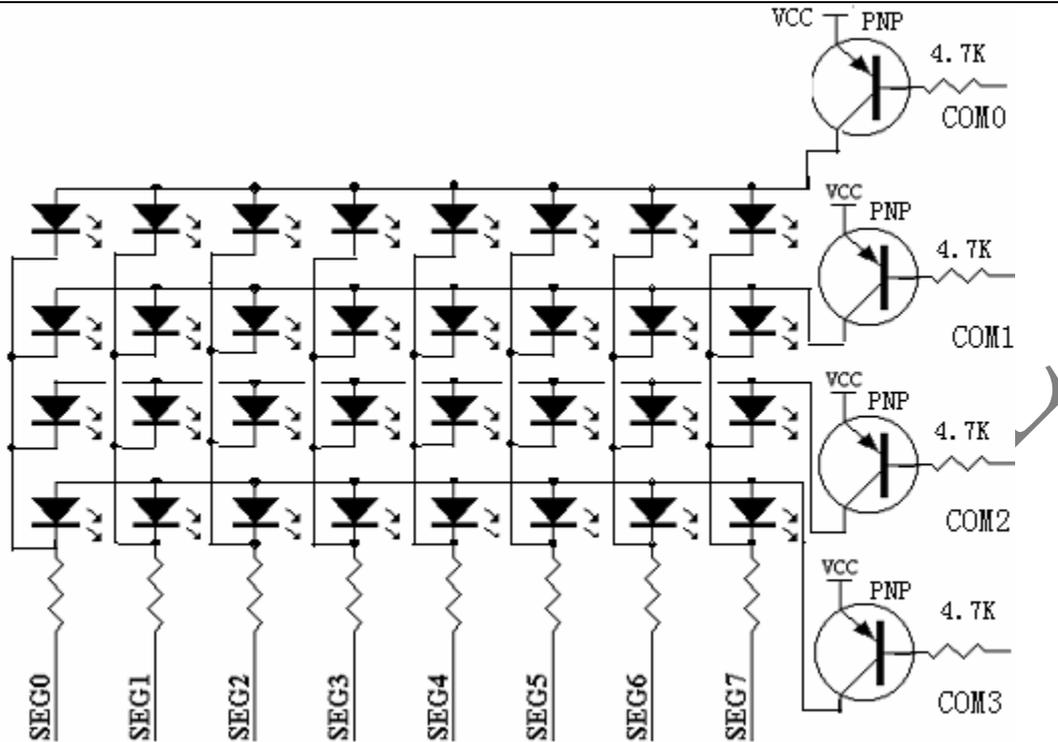


图 5: 4*8LED 显示控制电路

3.2 WTC6506D32 的驱动电流

SEG0~SEG7 每个驱动脚低电平时最大可以提供 20mA 的 LED 灌电流驱动。高电平时每个驱动脚最大可以提供 4mA 的源电流驱动。

3.3 显示内容缓冲区

WTC6506D32 内部有 4 组 8bit 的 RAM 作为显示内容缓冲区，用来提供 COM0~COM3 低电平扫描时 SEG0~SEG7 输出的数据。用户可以通过 SCLK 和 SDA 对显示内容缓冲区寻址并设定显示内容缓冲区的内容，以实现需要的显示内容。

WTC6506D32 内部显示内容缓冲区在初始上电的内容是随机的，上电后需要用户的 MCU 进行显示内容的初始化设定

4 输入输出接口

4.1 两线串行接口简介

WTC6506D32 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，并且向 WTC6506D32 内部的显示缓冲区写入数据进行 LED 界面的显示。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 WTC6506D32 能够有自动总线复位能力。**建议主控 MCU 产生的 SCLK 方波频率在 20KHZ~2KHz 之间，并且每隔 10ms 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 WTC6506D32 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。

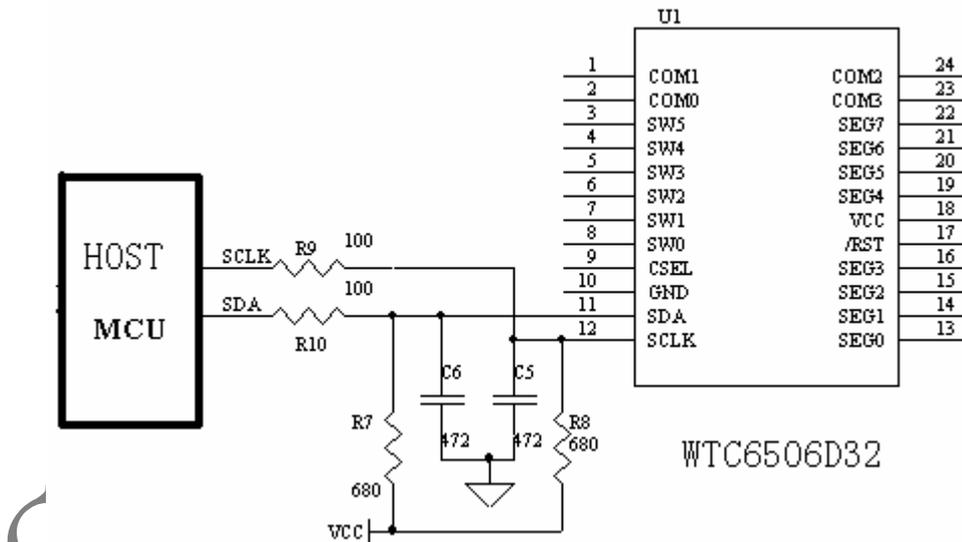


图 6: MCU 与 WTC6506D32 的串行接口电路

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

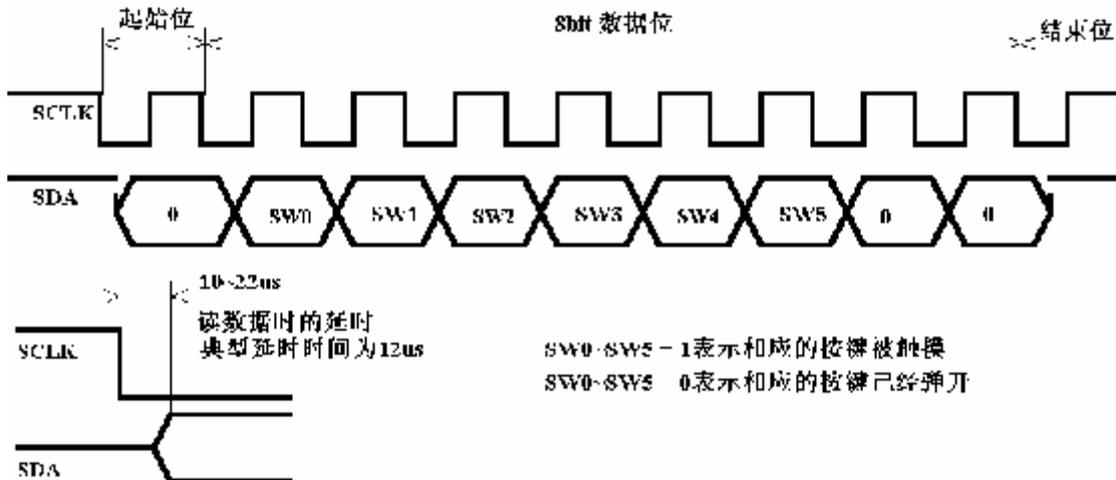


图 7: 主控 MCU 读取按键信息时序

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 将 SCLK 再次置为低电平，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(bit6,bit7 固定为 0, 是系统保留位)。

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到 MCU 对触摸芯片显示内容的设定和 SDA 上其他芯片的工作

4.3 设定 LED 显示内容

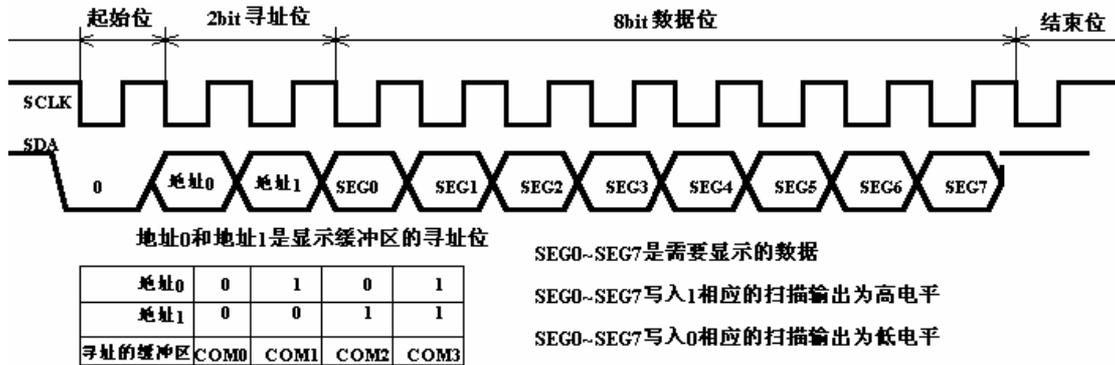


图 8:主控 MCU 写入 LED 显示缓冲区时序图

完成一次完整的显示信息设置主控 MCU 需要在 SCLK 上产生 12 个低电平脉冲，分为 1 个起始位，2 个显示缓冲区寻址位，8 个显示缓冲数据位，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显示缓冲区数据的读取准备状态。主控 MCU 将 SCLK 置高完成起始位设定

2: 设定 2 bit 的显示缓冲区地址

主控 MCU 先将需要设定的缓冲区地址的第 1 位数据放到 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 上的数据读入缓冲区地址识别寄存器的第 1 位。主控 MCU 将 SCLK 置高

以此类推主控 MCU 将缓冲区地址的第 2 位数据放到 SDA 上然后在 SCLK 上产生一个低电平脉冲完成显示缓冲区地址的设定

3: 将要显示的数据传入显示缓冲区的暂存区

主控 MCU 先将需要显示的数据的第 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 上的数据读入显示缓冲区的串入暂存区的第 1 位。主控 MCU 将 SCLK 置高

以此类推主控 MCU 将缓冲区地址的后面 7 位数据放到 SDA 上然后在 SCLK 上产生 7 个低电平脉冲，完成显示数据传入显示缓冲区的串入暂存区的操作。

4: 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。

WTC6506D32

WTC6506D32 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，本次寻址的显示缓冲区内容不会变化。只有当主控 MCU 在 SCLK 上产生结束位后触摸芯片才会将串入暂存寄存器的内容送入本次寻址的显示缓冲区。外部显示才会显示本次设定的内容。没有结束位的设定 WTC6506D32 的外部显示将不会显示本次传送的内容。这样可以避免传送数据期间 LED 显示抖动。

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 WTC6506D32 的电源

5.1 直流稳压器

WTC6506D32 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 7 是电源稳压电路

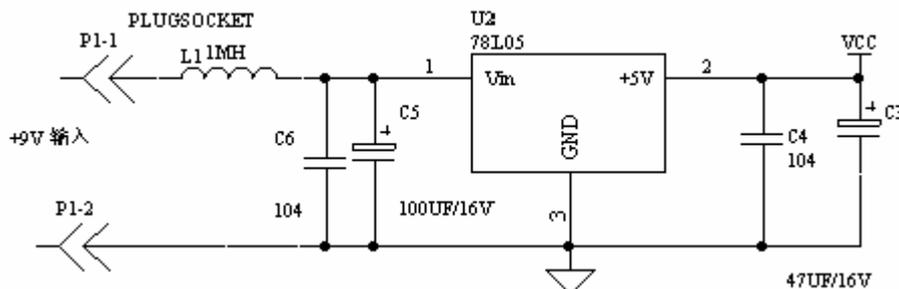


图 9: 直流稳压电路

5.2 稳压器组件的放置

PCB LAYOUT 时，该 78L05 电源组器件必须紧靠 WTC6506D32 Vcc 管脚。

78L05 及外围器件与 WTC6506D32 必须放在同一电路板上，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）

5.3 高噪声条件下的注意事项

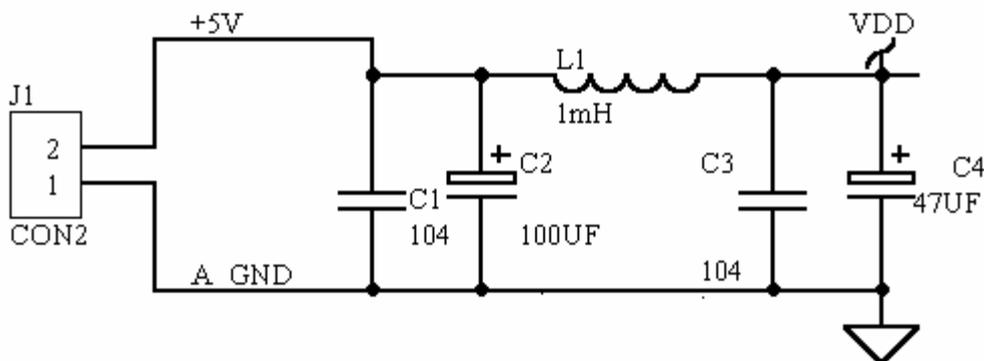
在高噪声环境应用时，应避免高压(220V)、大电流、高频率操作的主板与触摸电路板上下重叠安置。如无法避免，应尽量远离高压大电流的器件区域或在主板上加屏蔽。

5.4 电源滤波:

在 PCB 排版时，建议预留电感 L1(1mH)焊盘，不同的应用可能不需要此电感。输入 78L05 的电源，需注意纹波的大小，勿让纹波的彼谷低于 DC 8V。

5.5 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 10 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:

这个电路抑制电源噪声的能力较好，但接较大的负载容易产生自激。建议用户除了触摸模组或触摸芯片组外不要接其他负载。LED 灯，继电器和其他 IC 等负载需接在此电路前面并另加其他滤波稳压电路。



WTC6506D32

6 WTC6506D32 封装图及尺寸

| Symbol | Dimensions in mil | | |
|----------|-------------------|------|------|
| | Min. | Nom. | Max. |
| A | 228 | — | 244 |
| B | 150 | — | 157 |
| C | 8 | — | 12 |
| C' | 335 | — | 346 |
| D | 54 | — | 60 |
| E | — | 25 | — |
| F | 4 | — | 10 |
| G | 22 | — | 28 |
| H | 7 | — | 10 |
| α | 0° | — | 8° |

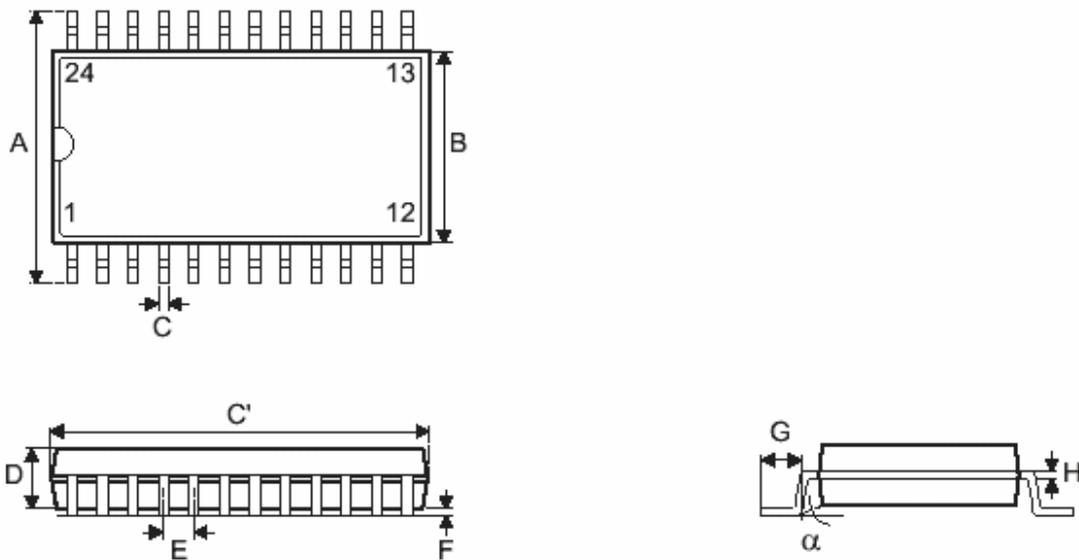


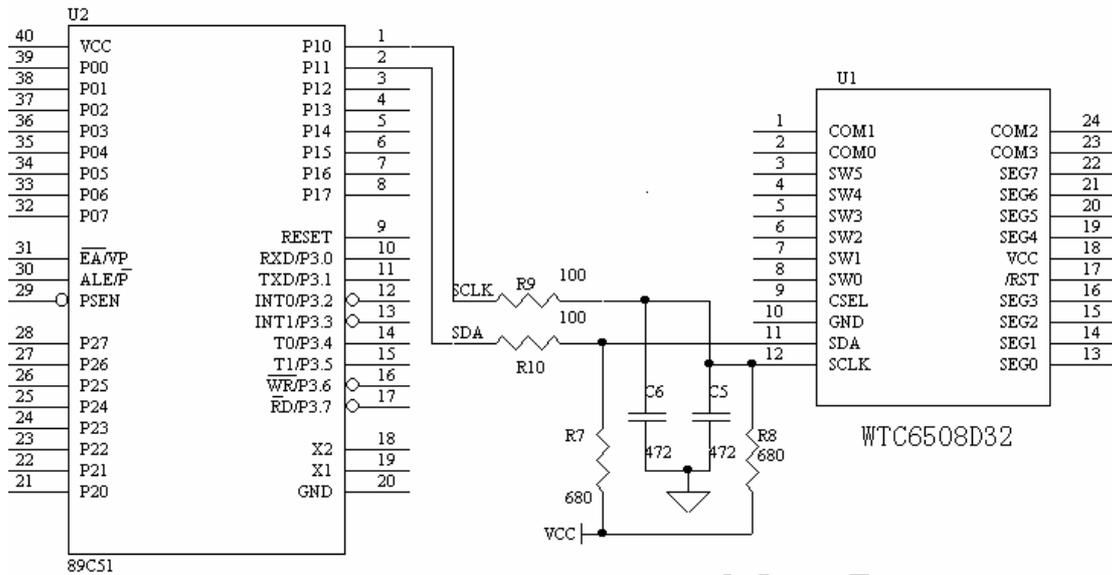
图 11: WTC6506D32 封装和尺寸图

7 WTC6506D32 与 MCU 8051 的接口电路和操作程序范例

WTC6506D32 与 8051 接口的典型应用:



WTC6506D32



典型操作程序程序对应电路

8051 与 WTC6506D32 接口的典型操作程序

/*

项目描述:

触摸与 LED 显示一体化的控制芯片 WTC6506 4*8LED 的 DEMO 板配套程序

DEMO 板功能:

检测触摸芯片输出的信息,并向触摸芯片的显示缓冲区写入数据完成相应的显示

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
//-----the funtion define-----
```



WTC6506D32

```
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    write_disp_buff(uchar ComDa,uchar SegDa);//向触摸芯片的显示缓冲区写入数据

void    disp_key_led(uchar KeyValue);
//-----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
                                //显示缓冲区
uchar   Com0DispData;       //需要 COM0 显示的数据
uchar   Com1DispData;       //需要 COM1 显示的数据
uchar   Com2DispData;       //需要 COM2 显示的数据
uchar   Com3DispData;       //需要 COM3 显示的数据

//-----
// 主程序
//-----
void main(void)
{
    init();
    Counter1ms = 0;
    do
    {
    } while(Counter1ms <= 100); //等待触摸芯片完成复位

    Com0DispData = 0xff;       //关闭 COM0 的所有显示
    Com1DispData = 0xff;       //关闭 COM1 的所有显示
    Com2DispData = 0xff;       //关闭 COM2 的所有显示
    Com3DispData = 0xff;       //关闭 COM3 的所有显示

    write_disp_buff(0,Com0DispData); //
    write_disp_buff(1,Com1DispData); //
    write_disp_buff(2,Com2DispData); //
    write_disp_buff(3,Com3DispData); //
```



WTC6506D32

```
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);    //操作触摸芯片显示相应的 LED
    }
}

//-----
//功能：向触摸芯片的显示缓冲区写入要显示的内容
//入口：1: ComDa 要写入的缓冲区地址
//      2: SegDa 要写入的缓冲区数据
//返回值：无
//-----
void write_disp_buff(uchar ComDa,uchar SegDa)
{
    TR1 =0;
    i = 0;

    //-----//发送 SDA 总线写数据的起始位-----
    SDA =0;    //起始位置 0,表示要向触摸芯片的显示缓冲区写入数据
    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();

    //-----选择触摸芯片的显示缓冲区 COM 口地址-----
    i = 0;
    do
    {
        if((ComDa & 0x01) == 0) {SDA = 0;} //准备设置缓冲区地址
        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
    }
}
```



WTC6506D32

```
    delay_24us();
    ComDa >>= 1;
    i++;
}
while(i <2);           //设置两位缓冲区地址

//-----向触摸芯片的显示缓冲区的暂存 buff 写入数据 SEG-----
i = 0;
do
{
    if((SegDa & 0x01) == 0) {SDA = 0;} //准备写入 SEG 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();
    SegDa >>= 1;
    i++;
}
while(i <8);          //写入 8 位数据
//-----产生结束脉冲，将串入暂存 buff 的内容送入显示缓冲区开始显示写入的数据
    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：从触摸芯片读取触摸按键的信息
//入口：无
//返回值：从触摸芯片读取触摸按键的信息
//-----
uchar  read_key_data(void)
{
    uchar  KeyValue;
    TR1 =0;
    i = 0;
```



WTC6506D32

```
//-----//发送 SDA 总线读数据的起始位-----
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();
SCLK = 1;          //产生串行时钟的上升延信号
delay_24us();
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 SDA 总线恢复时钟脉冲
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: 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



WTC6506D32

```
//-----  
void disp_key_led(uchar KeyValue)  
{  
    if((KeyValue & 0x01) != 0){Com0DispData &= 0xfe;} //KEY0 被按下用低电平点  
                                                //亮相应的 LED  
    else {Com0DispData |= 0x01;} //KEY0 弹起  
    if((KeyValue & 0x02) != 0){Com0DispData &= 0xfd;} //KEY1 被按下  
    else {Com0DispData |= 0x02;} //KEY1 弹起  
  
    if((KeyValue & 0x04) != 0){Com0DispData &= 0xfb;} //KEY2 被按下  
    else {Com0DispData |= 0x04;} //KEY2 弹起  
  
    if((KeyValue & 0x08) != 0){Com0DispData &= 0xf7;} //KEY3 被按下  
    else {Com0DispData |= 0x08;} //KEY3 弹起  
  
    if((KeyValue & 0x10) != 0){Com0DispData &= 0xef;} //KEY4 被按下  
    else {Com0DispData |= 0x10;} //KEY4 弹起  
  
    if((KeyValue & 0x20) != 0){Com0DispData &= 0xdf;} //KEY5 被按下  
    else {Com0DispData |= 0x20;} //KEY5 弹起  
  
    write_disp_buff(0,Com0DispData); //指示 LED 用 COM0 显示  
}  
  
//-----  
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
```



WTC6506D32

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

//-----
void delay_24us(void)
{
    _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;
}
```



专业的触摸感应芯片供应商

<http://www.wincomtech.com>

WTC6506D32

```
Counter1ms++;    //1ms 计数器加一  
}
```

CONFIDENTIAL