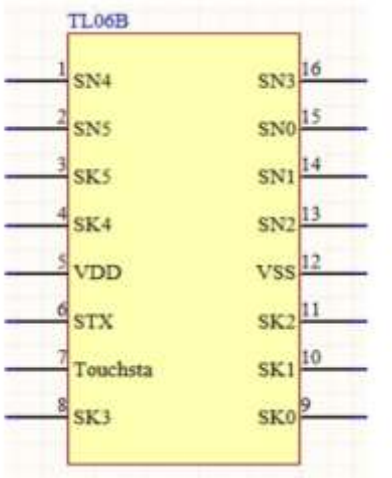


- 6 通道电容式触摸感应 IC
- 工作电压：2.5V～5.5V
- 工作电流：<2mA
- 工作温度：-40～85° C
- 强抗干扰，可以通过 4kv EFT,10v注入电流测试
- 多键并行扫描处理，响应快速
- 自适应环境温度变化
- 设计简单：无需编写代码，每个按键灵敏度的设置可通过按键上的外部调制电容大小调整实现。Touchsta 引脚为用户观测触摸按键灵敏度调试效果提供了方便简单的接口—有键按下，Touchsta 输出高电平，无键按下，Touchsta 输出低电平。
- 接口简单：单线 STX 通信。TL06BS 每次触摸按键扫描处理结束后，将每个按键 on/off 状态信息进行脉宽调制并通过 STX 线发送给主机系统。

应用范围：

家用电器、消费类电子产品、安防和楼宇产品、医疗保健产品、手持装置、工业控制、照明产品、玩具以及计算机周边等等。用于取代薄膜、按钮以及普通开关。

1、简介：

TL06BS 是一款 6 通道，单线通信接口触摸控制芯片，并可以根据客户需求定制开发各种功能。TL06BS 可在非导电类材质（如玻璃、亚克 力、塑胶、陶瓷等材质）的隔离下达到触摸功能，也可通过弹簧、普通导线等连接至小金属 片作为感应电极，按键灵敏度可根据实际情况单独自由调节，外围元件少，电路简单，加工方便，成本低廉。

TL06BS 芯片采用了低输入阻抗，以及电压采样与测量分离的模拟电容电压转换电路，从而对外部噪声构成低阻滤除和隔离通路，确保 TL06BS 芯片的强抗干扰性能。基于 TL06BS 芯片的的触摸应用方案可以通过+/-4kv EFT，8kv ESD 测试，和经受手机射频，日光灯，电磁炉等各种辐射噪声的干扰。TL06BS 采用了创新而先进的触摸基准更新算法，自动跟踪外界环境变化和补偿，使得触摸灵敏度和可靠性不受外界温度，湿度等环境变化的影响。

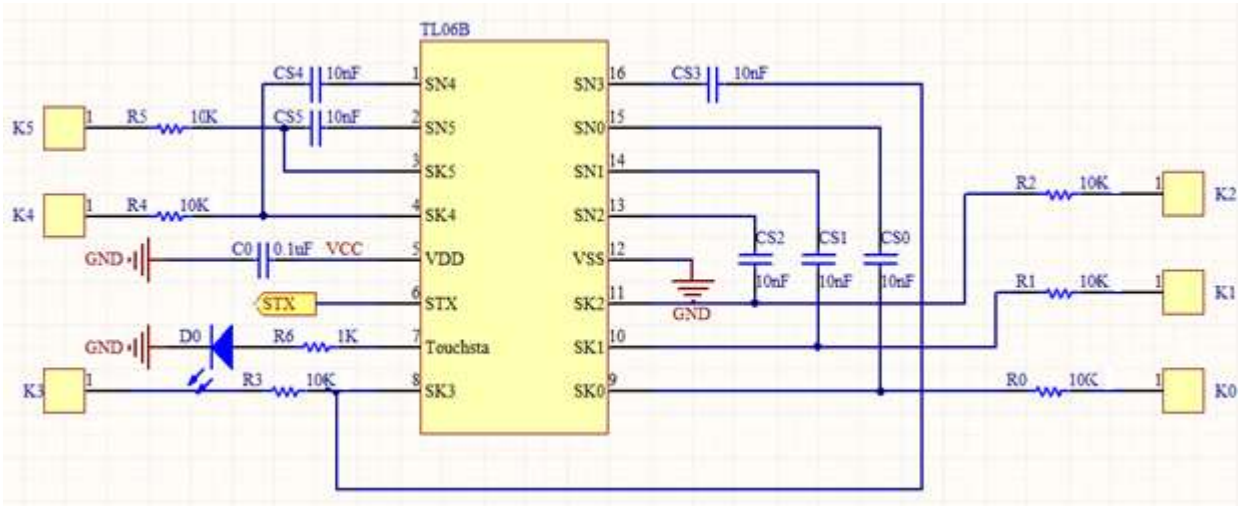
2、管脚定义

管脚名称	类型	功能描述
SK0~SK5	**I**	电容触摸感应输入端脚（不用时悬空）
SN0~SN5	**I**	调制电容采样端引脚（不用时悬空）
VDD	**P**	电源正极
VSS	**P**	电源负极
STX	**O**	触摸按键状态输出单线输出接口
Touchsta	**O**	芯片触摸有无状态输出—任一按键按下，输出高电平，所有键释放，输出低电平

3、工作参数额定值：

参 数	符号	额定值	单位
电源电压	**VDD**	**GND-0.5**～VSS+6.0	**V**
输入电压	**VIN**	**GND-0.3** to VDD+0.3	**V**
输出电压	**VOUT**	**GND<VOUT<VDD**	**V**
工作温度	**TOP**	**-40ºC** ～ +85**ºC**	**ºC**
储存温度	**TSTG**	**-40ºC** ～ +125**ºC**	**ºC**
工作频率	**FOP**	**8M**	**Hz**

4、应用电路



R0-R5: 触摸按键串联电阻，可以用来改善触摸按键抗射频辐射噪声性能，通常情况下选用10kohm。如果噪声比较大时，可以选用更大阻值的串联电阻。

CS0-CS5:触摸调制采样电容，通常推荐选用X7R(125度工作温度)或X5R(85度工作温度)，5～10%精度10nf电容。电容容值越大，对应按键灵敏度也越高。因此，用户可以通过调整每个按键上的调制电容大小，实现每个触摸按键灵敏度调整。

C0: 0.1uF电源滤波电容。

R6,D0: 推荐电路设计时在PCB上预留元件位置，在灵敏度调试时焊上方便直观察调试效果，量产时空置不焊。

5、PCB布局布线设计指南

TL06BS芯片在使用时推荐使用独立的一路电源，尽量保证触摸芯片电源干净。TL06BS芯片在设计时就充分考虑了减小电源差模噪声和共模噪声影响，用户无需在已有的电子系统中增加LDO。为了获得更好的触摸性能，建议PCB布局布线时遵守如下规则：

A. 触摸按键串联电阻（参考电路中的RS元件）尽量靠近SK引脚放置,如果电路板传感器本身电容过大，电阻阻值需要调小；调试时候可以用示波器探头量传感器焊盘信号波形，如果波形非方波信号，意味着电容充放电时间常数过大，充放电不彻底，则需要将电阻调小。

B. 调制电容（参考电路中的CS0-CS5元件）尽量靠近SNxx引脚放置。

C. 电源滤波电容（参考电路中的C0元件）尽量靠近电源引脚VDD放置。

D. 触摸按键走线在PCB制造工艺许可下尽量用比较细的走线，而且尽量短。

E. 触摸按键走线不要与电源，或其它触摸按键，信号平行走线。同时触摸按键走线与电源，地线，其它触摸按键或信号走线之间距离保持1mm以上。

F. 触摸电路部分进行地线设计时，请在网格铺地，铺铜率为30%左右，例如设置网格走线宽度为8mil，网格铺铜间距为100mil。

G. 触摸按键可以采用弹簧，导电泡棉，PCB焊盘等感应物体实现人体触摸检测，通常推荐设计为圆形，直径大小为(9+触摸面板厚度)mm；如由于结构设计限制，需设置为方形，则正方形边长为（9+触摸面板厚度）mm。

H. 触摸面板要求使用绝缘物质，阻值>100Mohm。如果触摸按键是PCB板焊盘，要求焊盘与触摸面板使用3M 467或468胶粘紧，中间不能有空隙。

I. 每个触摸按键灵敏度都可以通过调整调制电容CS0-CS5大小进行单独调节。调制电容容值增大，则按键灵敏度就越高，反之调制电容容值越小，按键灵敏度就越低。触摸按键灵敏度调节时，用户需要

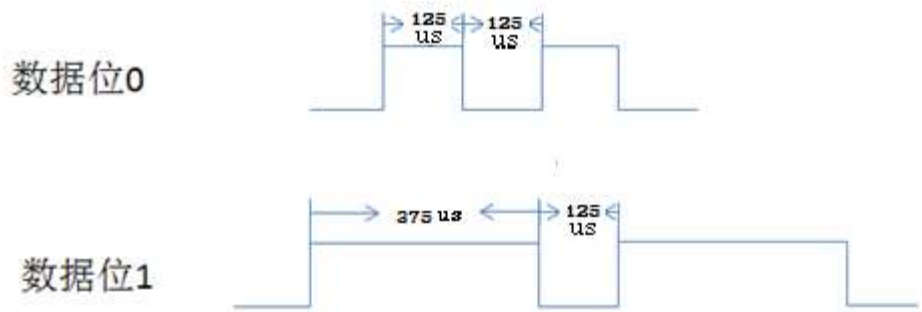
将按键与触摸面板安装或粘接好，并通过万用表或外焊一个 LED 监测 Touchsta 输出电平变化情况，然后将调制电容大小调节到手指轻触按键就输出高电平信号即可。

J. TL06BS 采用了分组扫描响应加速机制，即 0,2,4 为一组按键并行扫描，1,3,5 为另一组按键并行扫描，为了取得高靠性设计，在进行硬件设计时，物理按键位置请按照 0,1,2,3,4,5 顺序排放，从而在一组键扫描时，邻近的一组按键不扫描，默认为接地状态，避免邻近键之间的发生扫描信号串扰。

6、单线通信接口协议：

TL06BS 芯片的按键状态信息采用带有反码校验功能的双字节，即 12 位数据脉宽调制方式表示，按照从低位（第 0 位）到高位（第 12 位）的顺序依次将调制好的脉宽信息通过 STX 引脚发送出去，前面 6 位 6 个按键状态原始数据，后面 7-12 位为按键状态数据反码。如下图所示，TL06BS 的每位数据用一个脉冲进行表示，‘0’数据时脉冲高低电平部分都持续 125us；‘1’数据时脉冲低电平部分与‘0’数据相同，都为 125us，但是高电平部分为 375us。

TL06BS 每发送完 12 位数据后，自动将 STX 引脚设置成高电平并保持 2ms 以上，以便通知主机系统本次数据发送完毕。一旦下次触摸按键状态数据需要发射时，TL06BS 芯片先将 STX 引脚拉低 62.5us，再将数据从低位到高位依次进行脉宽调制发送出去，发送完后将 STX 引脚拉高保持 2ms 以上。



7、采用 IO 口中断上升沿+计数器实现主机单线通信接收参考代码

```
//外部全局变量设置
uint32_t countervalue;
uint16_t pulse[12],pulseshadow[12];
uint16_t keyvalue,keyvaluenot;
uint8_t  globalvar;
Volatile uint8_t counteroverflow, bitorder,datareadend;
struct tc_module timer_50us;
```

```
//IO 口中断引脚配置
static void setup_extint_callback_mode(void)
{
     //MCU touch data pin IO configure to input
     struct port_config pin_conf;
     port_get_config_defaults(&pin_conf);
     pin_conf.direction  = PORT_PIN_DIR_INPUT;
     pin_conf.input_pull = PORT_PIN_PULL_NONE;
     port_pin_set_config(touchdatapin, &pin_conf);


     / * Configure the external interrupt channel */
    extint_chan_get_config_defaults(&eic_conf);
    eic_conf.gpio_pin           = EIC_touchdata_PIN;
    eic_conf.gpio_pin_mux       = EIC_touchdata_PIN_MUX;
    eic_conf.gpio_pin_pull      = EXTINT_PULL_NONE;
    eic_conf.detection_criteria = EXTINT_DETECT_RISING;
    extint_chan_set_config(EIC_touchdata_CHANNEL, &eic_conf);
    /* Register and enable the callback function */
    extint_register_callback(extint_user_callback,EIC_touchdata_CHANNEL,EXTINT_CALLBACK_TYPE_DETECT);
    extint_chan_enable_callback(EIC_touchdata_CHANNEL, EXTINT_CALLBACK_TYPE_DETECT);
  }
//50us 计数器配置
void configure_tc(void)
{
    struct tc_config config_tc;
    tc_get_config_defaults(&config_tc);
    config_tc.counter_size = TC_COUNTER_SIZE_16BIT;
    config_tc.clock_source = GCLK_GENERATOR_1;
    config_tc.clock_prescaler = TC_CLOCK_PRESCALER_DIV512;
    config_tc.count_direction=TC_COUNT_DIRECTION_UP;
    tc_init(&timer_50us, TC4, &config_tc);
     tc_enable(&timer_50us);
    tc_register_callback(&timer_50us, tc_callback_process,TC_CALLBACK_OVERFLOW);
     tc_enable_callback(&timer_50us, TC_CALLBACK_OVERFLOW); //使能计数器溢出中断

}

//外部中断服务函数—每来一次中断，读取当前计数器的值，存入到 12 个 pulse 数组阵列里去，并将计数
//器起始值设置为 0；一旦接收完 12 位数据，将 datareadend 变量置“1”,告诉 MCU 主函数作相应的处理
static void extint_user_callback(void)
{
     countervalue=tc_get_count_value(&timer_50us);
     tc_set_count_value(&timer_50us,0);

     //if((countervalue>300)||(counteroverflow==1))  //计数器溢出标志可以帮助设置数据接收起始位从0开始
     if(countervalue>40) //不使用计数器溢出标志，有可能第一帧数据接收错误，由于本协议带有反码校验，可以自
                         //识别丢弃
    {
         bitorder=0;
         datareadend=0;
         counteroverflow=0;
     }
```

```
    else
    {
          pulse[bitorder]=(uint16_t)countervalue;
          if(bitorder<11)
               bitorder++;
          else  //一帧数据接收结束，将每位脉冲宽度数据送入影子寄存器，并将datareadend置1，以便MCU主函数知
                //知道一帧数据接收结束并时行相应的处理动作
          {
               datareadend=1;
              for(globalvar=0;globalvar<12;globalvar++)
                  pulseshadow[globalvar]=pulse[globalvar];

          }
    }
```

//50us 定时器中断服务函数

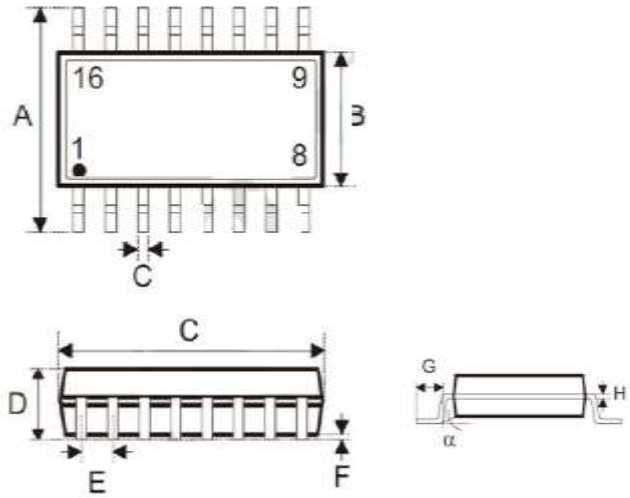
```
void tc_callback_to_toggle_led(struct tc_module *const module_inst)
{
      counteroverflow=1;
}
```

//MCU 主处理函数代码

```
Void main(void)
{
      ……
      ……
      setup_extint_callback_mode();
      configure_tc();
      //  configure_tc_callbacks();

      ……
      ……

      While(1)
      {
         //while loop里其它代码处理这里
          ……
          ……
          ……

          if(datareadend==1)
          {
               keyvalue=0;
               for(i=0;i<6;i++)
               {
                    if(pulse[i]>7) keyvalue|=(1<<i); //7代表350us脉冲周期

                }


          }
```

```
                keyvaluenot=0;
                for(i=6;i<12;i++)
                {
                      if(pulse[i]<7) keyvaluenot|=(1<<(i-6));

                }

                if(keyvalue==keyvaluenot)
                     display(keyvalue,0,0);
                datareadend=0;
            }

        //其它处理代码这里
        ……..
        ……..

      }  //while loop 结束

} //main 结束
```

9、封装信息（**SOP-16**）



	INCHES			MILLIMETERS		
	MIN	TYP	MAX	MIN	TYP	MAX
A	0.236 BSC			6.00 BSC		
B	0.154 BSC			3.90 BSC		
C	0.012	-	0.020	0.31	-	0.51
C'	0.390 BSC			9.90 BSC		
D	0.065	-	0.069	1.64	-	1.75
E	0.050 BSC			1.27 BSC		
F	0.004	-	0.010	0.10	-	0.25
G	0.016	-	0.050	0.40	-	1.27
H	0.004	-	0.010	0.10	-	0.25
α	-	-	8°	-	-	8°