

磐芯电子触摸库使用说明

一、M9F680 触摸库使用说明：

1、M9F680 三种触摸库的区别：

①、M9F680_CM_Touch_Demo_DV3xx：

该库占用的 RAM 较多，但是 CS 动态较好，对于对所占 RAM 无要求，CS 动静态要求较高的客户建议使用此库，考虑到触摸手感问题，对于要求触摸手感较好的客户，建议使用低于 8 个按键；

②、M9F680_CM_Touch_LP_Demo_V2xx：

该库相对于“M9F680_CM_Touch_Demo_DV3xx”库，所占 RAM 较小，CS 动态也要相对差一点，建议对 RAM 大小有要求，12 个按键以内的客户，使用此库；

③、M9F680_CM_Touch_LP_Demo_FV4xx：

该库相对于“M9F680_CM_Touch_LP_Demo_V2xx”库，所占 RAM 相同，但 CS 动态相对差一点，但触摸手感较好，对于需要按键数量较多的客户建议使用此库；

④、触摸 DEMO 板 CS、EFT 测试结果：

版本	CS 测试		EFT 测试	
	静态	动态	SYN	SEQ
DV3xx	15V	12V	4KV	4KV
V2xx	15V	10V	4KV	4KV
FV4xx	15V	10V	4KV	4KV

⑤、休眠功耗：

以触摸 DEMO 板 8 个按键为例		
版本	电压	休眠功耗
V2XX	3V	12.5uA
	5V	19.5uA
FV4XX	3V	12.5uA
	5V	19.5uA
DV3XX	无低功耗模式	

2、M9F680 触摸库参数配置说明：

①、触摸通道选择：

0：该通道按键不检测；1：该通道按键需检测。

```
//-----  
//          0：该通道按键不检测；1：该通道按键需检测。  
#define EnTk0 0  
#define EnTk1 0  
#define EnTk2 0  
#define EnTk3 0  
#define EnTk4 1  
#define EnTk5 1  
#define EnTk6 1  
#define EnTk7 1  
#define EnTk8 1  
#define EnTk9 1  
#define EnTk10 1  
#define EnTk11 1  
#define EnTk12 0  
#define EnTk13 0  
#define EnTk14 0  
#define EnTk15 0  
#define EnTk16 0  
#define EnTk17 0  
#define EnTk18 0  
#define EnTk19 0  
#define EnTk20 0  
#define EnTk21 0  
#define EnTk22 0  
#define EnTk23 0  
#define EnTk24 0  
#define EnTk25 0  
#define EnTk26 0  
#define EnTk27 0  
#define EnTk28 0  
#define EnTk29 0  
//-----
```

②、触摸阈值设置：

设置所使用触摸按键的阈值，建议取按键最大变化值的 40%，越小越灵敏，越大抗干扰能力越强；

```
//-----  
// 触摸阈值：取值需具体应用环境决定，最小30，最大255，值越小越灵敏  
#define TKLib_TK0_F_THRESHOLD 50  
#define TKLib_TK1_F_THRESHOLD 50  
#define TKLib_TK2_F_THRESHOLD 50  
#define TKLib_TK3_F_THRESHOLD 50  
#define TKLib_TK4_F_THRESHOLD 50  
#define TKLib_TK5_F_THRESHOLD 50  
#define TKLib_TK6_F_THRESHOLD 50  
#define TKLib_TK7_F_THRESHOLD 50  
#define TKLib_TK8_F_THRESHOLD 50  
#define TKLib_TK9_F_THRESHOLD 50  
#define TKLib_TK10_F_THRESHOLD 50  
#define TKLib_TK11_F_THRESHOLD 50  
#define TKLib_TK12_F_THRESHOLD 50  
#define TKLib_TK13_F_THRESHOLD 50  
#define TKLib_TK14_F_THRESHOLD 50  
#define TKLib_TK15_F_THRESHOLD 50  
#define TKLib_TK16_F_THRESHOLD 50  
#define TKLib_TK17_F_THRESHOLD 50  
#define TKLib_TK18_F_THRESHOLD 50  
#define TKLib_TK19_F_THRESHOLD 50  
#define TKLib_TK20_F_THRESHOLD 50  
#define TKLib_TK21_F_THRESHOLD 50  
#define TKLib_TK22_F_THRESHOLD 50  
#define TKLib_TK23_F_THRESHOLD 50  
#define TKLib_TK24_F_THRESHOLD 50  
#define TKLib_TK25_F_THRESHOLD 50  
#define TKLib_TK26_F_THRESHOLD 50  
#define TKLib_TK27_F_THRESHOLD 50  
#define TKLib_TK28_F_THRESHOLD 50  
#define TKLib_TK29_F_THRESHOLD 50  
//-----
```

③、设置允许同时响应的最多按键数量：

客户可根据实际需求更改；

```
// 设置允许同时响应的最多按键数量： (1 <= value <= 开启的按键总数) 请根据实际使用时，同时触摸最多按键个数配置  
#define TKLib_TOUCH_NUM 1
```

④、设置按键最长有效时间：

客户可根据实际需求更改；

```
// 设置按键最长有效时间： 0：一直有效，即按键标志产生后，只要手指不离开，按键标志就会一直存在，不建议配置成0
//                               非0：当有任何触摸标志次数大于设定值时，所有的按键标志会被清除并复位所有通道的触摸基线，
//                               注TKLib_LONG_TOUCH_TIME非0时，必须大于（TKLib_Debounce_TIME + TKLib_CONFIRM_TOUCH_TIME）。
//                               按键最长有效时间 = TKLib_LONG_TOUCH_TIME * aSCAN_TIME，其中aSCAN_TIME为Touch_Key_Process调用间隔时间
//                               在CPU = 16M/4T时 aSCAN_TIME 估算值 > 开启的通道数 * 1ms
//                               (1<= value <=65535) 通常最长有效时间控制在8-30S左右，有特定时长需求的可根据实际需求调整，
//                               但不建议时间设定过长，因为时间过长，意味着异常时自恢复需要的时间越长
#define TKLib_LONG_TOUCH_TIME          2500
```

⑤、设置触摸按键基线更新速度：

值越小基线更新速度越快，一般不需要客户更改，默认配置即可；

```
// 设置触摸按键基线更新速度：          (1<= value <=100)
#define TKLib_BASELINE_CALIBRATE_TIME    5 //一般不需要修改
```

⑥、用于按键超时的计数：

配置值越大，用于判断按键超时时间越久，一般配置超时时间在 1ms 到 2ms 之间即可，配置方式为：Timer0*80*TKLib_TIMEOUT_CNT，DEMO 程序 TC0 中断时间为 125us，非必要不要修改；

```
// 用于按键超时判断：取值为 1ms < value*time0 < 2ms 即可
//                               (1 <= value <= 255)
#define TKLib_TIMEOUT_CNT              10
```

⑦、上电按键有效延时时间：

初次上电一个等待电源稳定的延时时间，该时间内触摸按键不会响应，配置大小为 TKLib_PowerOn_Delay_Time*10ms；

```
// 上电按键有效延时时间，可根据实际需求修改
//                               (0<= value<=254) value*10ms
#define TKLib_PowerOn_Delay_Time        200
```

⑧、触摸省电模式设定：

如果需要设置省电模式，则先开启省电模式，然后设置触摸低功耗模式唤醒时间，值越大功耗越小，但唤醒速度也越慢，最后设置低功耗滤波等级，等级越高功耗越大；休眠程序中使用了 TC0、TC2,如外部程序有使用相关时基，休眠唤醒后需恢复。

```
// =====
// 触摸省电模式设定，一般运用请关闭此模式
// =====
// 设置触摸低功耗模式是否开启 0：不使用触摸省电模式；1：使用触摸省电模式。
#define TKLib_TouchLowerPower          1
// 设置触摸低功耗模式唤醒时间： (0 <= value <= 7) 值越大功耗越小，但唤醒速度会更慢
#define TKLib_LP_WAKEUP_TIMER          2 // 0 -> 25mS,1 -> 50mS,2 -> 100mS,3 -> 200mS, 4 -> 0.4S,5 -> 0.8S,6 -> 1.6S,7 -> 3.2S
// 设置低功耗滤波等级：          0 -> 普通, 1 -> 中 (等级越高，功耗越大)
#define TKLib_LP_FILTER_LEVEL          0
```

```
if(!TouchKeyFlag)
{
    if(++R_GOTOWLOWPOWERMODE_COUNT>T_GOTOWLOWPOWERMODE_C_Init_DF)
    {
        R_GOTOWLOWPOWERMODE_COUNT = 0;

        bLowPowerModeRun = 1;
#if TKLib_TouchLowerPower
        /*下面加睡眠前其它需要低功耗处理的操作(除触摸外的，触摸相关的_TkLowPowerMode会自行处理)
        如低功耗模式下端口处理，外设模块等
        WDT与CPU工作模式用户不需要处理*/

        //关闭其他唤醒源，本demo code外部只打开了T0，固只关闭T0
        TCOIE = 0;
        //触摸低功耗模式

        if(bLowPowerModeRun)
        {
            TkLowPowerMode();

            //下面加唤醒后其它需要恢复的操作
            //注：休眠程序中使用了TC0和TC2,恢复之后如需使用,请重新配置

            //恢复TC2
            TC2_Init();

            //恢复TC0
            TCOCH = 0;
            TCOCL = T0_LOAD_C_Init;
            TOCR = Bin(10101100); //16M/1:16
            TCOIE = 1;
#endif
    }
}
else
{
    R_GOTOWLOWPOWERMODE_COUNT = 0;
}
```

***注意：以上配置客户可根据实际需求适当修改，以下配置参数客户不可自行修改，如果必须修改，则必须询问厂家意见；**

⑨、设置确认手指触摸、离开次数：

设置按键有效和无效次数，次数越少则灵敏度越好，但相对抗干扰会差一些，配置越大灵敏度会相对差一些，但抗干扰能力会强，一般不需要修改；

```
// 设置确认手指触摸次数:          (1 <= value <= 15)
#define TKLib_EFFECTIVE_TOUCH_TIME  4

// 设置确认手指离开次数:          (1 <= value <= 15)
#define TKLib_INVALID_TOUCH_TIME    3
```

⑩、以下参数保持默认，不可修改：

```
//                                (1<= value<=3)
#define TKLib_TIM                  2

//                                (0<= value<=11)
#define TKLib_TKCKS                10

// 设置触摸放电电阻阻值:          (1 <= value <= 255)
#define TKLib_TKRCTR_DATA          5

// 触摸数据采集次数，不需要更改    (2 <= value <= 255)
#define TKLib_ScanCnt              5
```

3、旧版触摸库替换和新版库使用注意事项：

- ①、配置触摸按键检测通道“EnTkx”；
- ②、配置触摸阈值“TKLib_TKx_F_THRESHOLD”，一般取最大变化量的40%；
- ③、配置上电按键有效延长时间“TKLib_PowerOn_Delay_Time”，大于电源电压稳定时间；
- ④、如需触摸低功耗模式，则按需求配置参数即可；
- ⑤、使用旧版库程序替换和新版库使用时，需注意添加主程序中新增的一些函数和参数，防止替换后程序功能异常。

```

//触摸占用了低优先级中断
Touch_Init();

GIEH=1;

#if TouchUartDebugLib_SW
    Touch_UartDebug_Init();
#endif

while(1)
{
    CLRWDT();

    TouchFailDealMode();

    if(OneCycleScanDone)
    {
        //一轮转换完成
        OneCycleScanDone = 0;
    }

    #if TouchUartDebugLib_SW
        //GIEH=0;
        Touch_UartDebug();
        //GIEH=1;
    #endif

    Touch_Key_Process();
}

/* User code */
KEY_TASK(); //按键标志用户处理
    
```

1


```

TouchFailDealMode();

if(OneCycleScanDone)
{
    //一轮转换完成
    OneCycleScanDone = 0;
}

#if TouchUartDebugLib_SW
    //GIEH=0;
    Touch_UartDebug();
    //GIEH=1;
#endif

Touch_Key_Process();
    
```

2


```

//-----
//以下程序触摸库使用到，不可修改或删除
if(ChannelCalOvertimeStartFlag)
{
    if(++TouchKeyFailCnt > CFG_TK_TIMEOUT_CNT)
    {
        ChannelCalOvertimeFlag = 1;
        ChannelCalOvertimeStartFlag = 0;
        TouchKeyFailCnt = 0;
    }
}

if(++T_LMS_COUNT >= T_LMS_DF)
{
    T_LMS_COUNT = 0;
    bT_LMS_F = 1;
}

if(bT_LMS_F)
{
    bT_LMS_F = 0;
    if(++T_10MS_COUNT >= T_10MS_DF)
    {
        T_10MS_COUNT = 0;
        bT_10MS_F = 1;
        PowerOn_DelayFlag = 1;
    }
}
//-----
    
```

3


```

        PowerOn_DelayFlag = 1;
    
```

4

- 1、上电调用触摸初始化 Touch_Init(), 打开中断使能位。
 - 2、主循环中调用函数 TouchFailDealMode(), 每次转换完成需调用 Touch_Key_Process()。
 - 3、配合中断中的处理, 用于防止触摸通道短路而导致其他触摸通道无法响应的问题。
 - 4、中断里 10ms 调用标志位 PowerOn_DelayFlag , 为新增上电延时功能标志位。
- 以上主程序中新增内容替换和使用时务必要注意保留。

4、新版触摸库与旧版触摸库对比:

新版本库和旧版本库新增部分功能参数, 删除部分旧功能。

①、删除触摸消抖测试、触摸噪声阈值、触摸按键迟滞值。新增按键超时判断、上电按键有效延长时间。

```

79
80 //-----
81 // 设置允许同时响应的最多按键数量: (1 <= value <= 开启的按键总数) 请根据实际使用时, 同时触摸最多按键个数配置
82 #define TKLib_TOUCH_NUM 1
83
84 // 设置触摸消抖次数:
85 // 消抖时间 = TKLib_Debounce_TIME * aSCAN_TIME, 其中aSCAN_TIME 为 Touch_Key_Process 调用间隔
86 // 在CPU = 16M/4T时 aSCAN_TIME 估算值 > 开启的通道数 * 1ms
87 // (2 <= value <= 254) 值越小按键响应越快, 反之更慢。不宜过快, 以免影响抗干扰效果, 故请根
88 #define TKLib_Debounce_TIME 3
89
90 // 设置按键最长有效时间: 0: 一直有效, 即按键标志产生后, 只要手指不离开, 按键标志就会一直存在, 不建议配置成0
91 // 非0: 当有任何触摸标志次数大于设定值时, 所有的按键标志会被清除并复位所有通道
92 // 注TKLib_LONG_TOUCH_TIME非0时, 必须大于 (TKLib_Debounce_TIME + TKLib_CONFIRM_TOUCH_TIME)
93 // 按键最长有效时间 * TKLib_LONG_TOUCH_TIME * aSCAN_TIME, 其中aSCAN_
94 // 在CPU = 16M/4T时 aSCAN_TIME 估算值 > 开启的通道数 * 1ms
95 // (1<= value <=65535) 通常最长有效时间控制在8~30s左右, 有特定时长需求的可根据实际需求调整
96 // 但不建议时间设定过长, 因为时间过长, 意味着异常时自恢复需要的时间越
97 #define TKLib_LONG_TOUCH_TIME 2500
98
99 // 设置触摸噪声阈值: 建议 10 <= value <= 30, 但必需小于开启的按键通道设置的阈值中最低值的1/2
100 #define TKLib_NOISE_THRESHOLD 15
101
102 // 设置触摸按键迟滞值: 最小值为5, 最大值必须小于所有开启通道中设置最小的触摸阈值 - 噪声阈值
103 #define TKLib_FT_HYS 10
104
105 // 设置触摸按键基线更新速度: (5<= value <=200) 值越小更新越快, 反之更慢
106 #define TKLib_BASELINE_CALIBRATE_TIME 25 //一般不修改
107
108
109
110
111 #define TKLib_IK20_F_THRESHOLD 70
112 #define TKLib_IK29_F_THRESHOLD 70
113
114 //-----
115 // 设置允许同时响应的最多按键数量: (1 <= value <= 开启的按键总数) 请根据实际使用时, 同时触摸最多按键个数配置
116 #define TKLib_TOUCH_NUM 1
117
118 // 设置按键最长有效时间: 0: 一直有效, 即按键标志产生后, 只要手指不离开, 按键标志就会一直存在, 不建议配置成0
119 // 非0: 当有任何触摸标志次数大于设定值时, 所有的按键标志会被清除并复位所有通道
120 // 注TKLib_LONG_TOUCH_TIME非0时, 必须大于 (TKLib_Debounce_TIME + TKLib_CONFIRM_TOUCH_TIME)
121 // 按键最长有效时间 * TKLib_LONG_TOUCH_TIME * aSCAN_TIME, 其中aSCAN_
122 // 在CPU = 16M/4T时 aSCAN_TIME 估算值 > 开启的通道数 * 1ms
123 // (1<= value <=65535) 通常最长有效时间控制在8~30s左右, 有特定时长需求的可根据实际需求调整
124 // 但不建议时间设定过长, 因为时间过长, 意味着异常时自恢复需要的时间越长
125 #define TKLib_LONG_TOUCH_TIME 2500
126
127 // 设置触摸按键基线更新速度: (1<= value <=100)
128 #define TKLib_BASELINE_CALIBRATE_TIME 5 //一般不需要修改
129
130 // 用于按键超时判断: 取值为 1ms < value*time0 < 2ms 即可
131 // (1 <= value <= 255)
132 #define TKLib_TIMEOUT_CNT 10
133
134 // 上电按键有效延长时间, 可根据实际需求修改, 默认2s
135 // (0<= value<=254) value*10ms
136 #define TKLib PowerOn Delay Time 10
137 //-----

```

②、新增触摸省电模式

```

//-----
// 触摸省电模式设定，一般运用请关闭此模式
//-----
// 设置触摸低功耗模式是否开启 0: 不使用触摸省电模式；1: 使用触摸省电模式。
#define TKLib_TouchLowerPower      1

// 设置触摸低功耗模式唤醒时间：(0 <= value <= 7) 值越大更功耗越小，但唤醒速度会更慢
#define TKLib_LP_WAKEUP_TIMER      2      // 0 -> 25mS,1 -> 50mS,2 -> 100mS,3 -> 200mS, 4

// 设置低功耗滤波等级：
// 0 -> 普通, 1 -> 中 (等级越高, 功耗越大)
#define TKLib_LP_FILTER_LEVEL     0
    
```

③、删除灵敏度挡位、滤波等级。新增触摸放电电阻、触摸采集次数

```

// 设置灵敏度挡位：
// 0: 普通；1: 高
#define TKLIB_SENSITIVITY_LEVEL   0

// 设置滤波等级：
// 0 -> 普通, 1 -> 中, 2 -> 高
#define TKLib_FILTER_LEVEL       1

// 设置确认手指触摸次数：
// (1 <= value <= 15)
#define TKLib_EFFECTIVE_TOUCH_TIME 3

// 设置确认手指离开次数：
// (1 <= value <= 15)
#define TKLib_INVALID_TOUCH_TIME  2

//
// (1<= value<=3)
#define TKLib_TIM                 2

//
// (0<= value<=11)
#define TKLib_TKCKS               7

#endif /* __Touch_Lib_CFG_H__ */
    
```

```

//-----
// 以下参数用户不可自行修改
//-----
// 设置确认手指触摸次数：
// (1 <= value <= 15)
#define TKLib_EFFECTIVE_TOUCH_TIME 3

// 设置确认手指离开次数：
// (1 <= value <= 15)
#define TKLib_INVALID_TOUCH_TIME  2

//
// (1<= value<=3)
#define TKLib_TIM                 2

//
// (0<= value<=11)
#define TKLib TKCKS               10

// 设置触摸放电电阻阻值：
// (1 <= value <= 255)
#define TKLib_TKRCTR_DATA         5

// 触摸数据采集次数，不需要更改
// (2 <= value <= 255)
#define TKLib_ScanCnt             5

#endif /* __Touch_Lib_CFG_H__ */
    
```