触摸功能使用说明

Version 1.00

2019 年 5 月

上 海 磐 芯 电 子 有 限 公 司

SHANGHAI MASSES ELECTRONIC Co., Ltd.

本公司保留对产品在可靠性，功能和设计方面的改进作进一步说明的权利。
数据手册的更改，恕不另行通知。

http://www.masses-chip.com/

本公司不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，本公司的产品不是专门设计来应用于外科植入、生命维持和任何本公司产品的故障会对个体造成伤害甚至死亡的领域。如果将本公司的产品应用于上述领域，即使这些是由本公司在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接产生的律师费用，并且用户保证本公司及其雇员、子公司、分支机构和销售商与上述事宜无关。

修正记录

版本	日期	描述
V1.00	2019-5-10	初版

目录

1 简述 5

2 如何在 IDE 中加载触摸库文件及使用 6

2.1 在 IDE 工程中加载库文件 6
2.1.1 打开库文件 6
2.1.2 在 IDE 中加载库文件 7
2.2 在 IDE 工程中使用库 8
2.2.1 添加触摸库里 INC 文件 8
2.2.2 设置触摸库里需要的时基 8
2.2.3 在程序中调用库 10
2.2.4 触摸初始化 Init 介绍说明 12

3 库文件及配置说明 14

3.1 M8Pxxx_Touch_M0_Lib_CFG_V202.inc 14
3.1.1 触摸通道设置 14
3.1.2 TKLib_TOUCH_NUM 14
3.1.3 TKLib_Debounce_TIME 14
3.1.4 TKLib_LONG_TOUCH_TIME 15
3.1.5 TKLib_Calibration_TOUCH_TIME 15
3.1.6 TKLib_M0_TKCKS 15
3.1.7 TKLib_EFFECTIVE_TOUCH_TIME 15
3.1.8 TKLib_INVALID_TOUCH_TIME 15
3.1.9 TKLib_FILTER_LEVEL 15
3.1.10 触摸阈值设置 15
3.1.11 TKLib_TouchLowerPower 15
3.1.12 触摸模块设置 16
3.1.13 触摸通道模块分组设置 16
3.1.14 TKLib_Sleep_F_THRESHOLD 16
3.1.15 TKLPLib_RCOUT_EN 16
3.1.16 辅助端口设置 16
3.1.17 用户不可修改部分 17
3.2 M8Pxxx_Touch_M0_Lib_Data_V202.INC 17
3.3 M8Pxxx_Touch_M0_Lib_V202.INC 17
3.3.1 _TkInit 17
3.3.2 _TkScan 17
3.3.3 _TkSignalHandle 17
3.3.4 _ScanContinue 17
3.3.5 _BaseLineReset 17
3.3.6 _TkLowPowerMode 17
3.3.7 _TK_TouchUartDebug_Init 18
3.3.8 _TK_TouchUartDebug 18
3.3.9 bOneScanDone 18

3.3.10 bT_TK_100MS_F 18
3.3.11 bT_TK_4S_F 18
3.3.12 bLowPowerModeReset 18
3.3.13 bLowPowerModeRun 18
3.3.14 触摸按键触摸响应标志 18
3.4 M8PXXX_TOUCH_M0_UARTDEBUG_LIB_INIT.INC 18
3.4.1 TKUartDebugLib_DEBUG_MODE_SWITCH 19
3.4.2 TKUartDebugLib_DEBUG_UART_SELECT 19
3.4.3 TKUartDebugLib_BAUDRATE 19
3.4.4 模拟串口软件端口选择 19
3.4.5 不可修改部分 19
3.5 M8PXXX_TOUCH_M0_LIB_V202.A 19

4 触摸串口调试工具使用说明 20

4.1 触摸串口调试工具文件 20
4.2 触摸串口调试工具界面 20

5 触摸 PCB 设计注意事项 22

5.1 整体布局要求 22
5.1.1 MCU 电源两端电容 22
5.1.2 触摸线串电阻 23
5.1.3 触摸参考电容位置 23
5.1.4 MCU 位置摆放注意 24
5.1.5 触摸按键间距注意 24
5.2 布线要求 25
5.2.1 触摸按键走线 25
5.3 触摸按键形状 26
5.3.1 普通触摸按键 26
5.3.2 滑条滚轮 26
5.4 实际布板参考 27
5.4.1 参考一单面空间够 27
5.4.2 参考二单面空间紧凑 28
5.4.3 参考三双面板空间够 29
5.4.4 参考四双面板空间紧凑 30
5.4.5 参考五模具厚 30
5.5 注意事项总结 31

1 简述

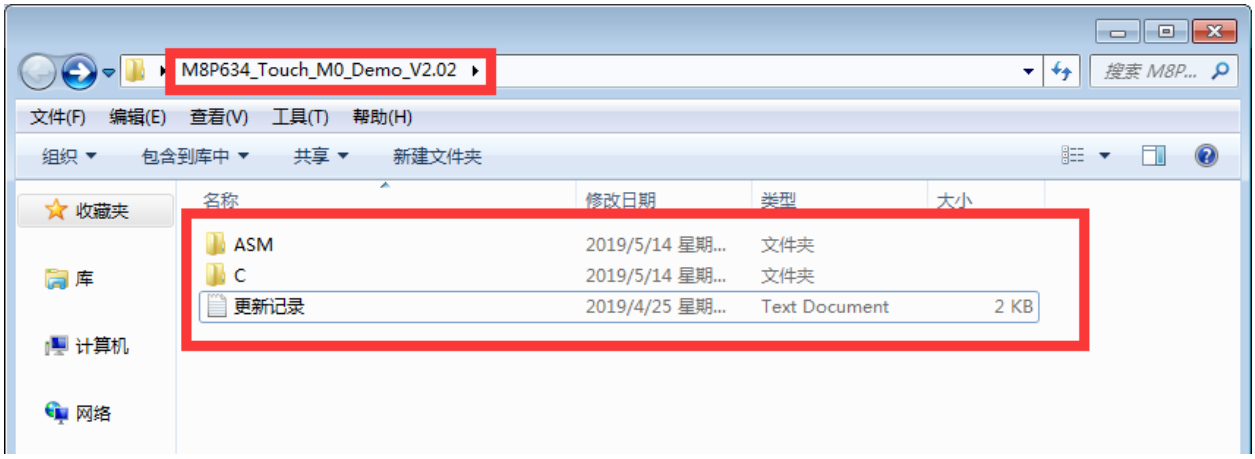
本触摸库是一种通用型库，用户基于库开发可以大大缩减开发周期。本文档主要为用户介绍 IDE 中如何加载库文件，提供库文件中相关配置文件说明，PCB 设计注意事项。

2 如何在IDE中加载触摸库文件及使用

2.1 在 IDE 工程中加载库文件

2.1.1 打开库文件

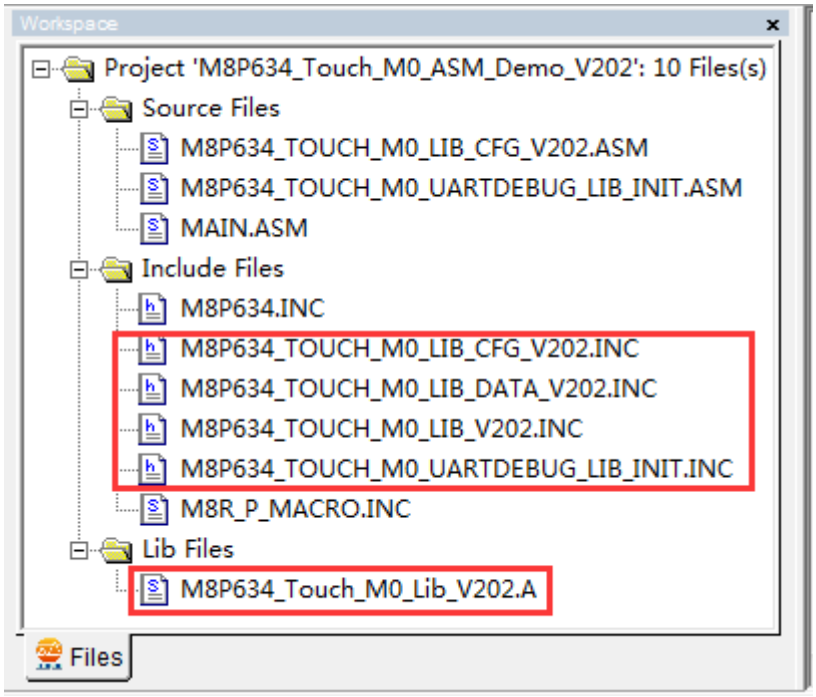
如下图所示，以 M8Pxxx 触摸库为例，V2.02 表示库的版本，文件夹里面有 ASM 和 C 两个版本，更新记录的文件里面是对每个版本的修改做的简单说明。以下的案例以 ASM 为例，C 的说明也是一样通用。



2.1.2 在 IDE 中加载库文件

> 注：特别注意触摸库需要在外部程序产生两个时基来做数据更新和长按清除功能。

库文件为.A 文件，上面框里面的.INC 文件为库里面一些需要定义或者声明的文件。如下图：



2.2 在 IDE 工程中使用库

2.2.1 添加触摸库里 INC 文件

MAIN

```
;
; 16M/4T, WDT EN 72MS, RST IS IO ,BOR 2.4V,RST TIMER 18MS
;
;*******************************************************************

     LIST P= M8P634

;=================================================================
#include      m8P634.inc
#include      M8R_P_MACRO.inc

;=================================================================
#include      M8P634_Touch_M0_Lib_V202.inc
#include      M8P634_Touch_M0_Lib_CFG_V202.inc
#include      M8P634_Touch_M0_Lib_Data_V202.inc
#include      M8P634_Touch_M0_UartDebug_LIB_Init.inc

;=================================================================
 CODE
;=================================================================
```

本处 Demo 是在 MAIN.ASM 文件中使用触摸库。

2.2.2 设置触摸库里需要的时基

MAIN

```
;=================================================================================
 CODE
;=================================================================================
R_Time_Flag       EQU     h'00'
#define bT_125UF_F    R_Time_Flag,0
#define bT_500US_F    R_Time_Flag,1
#define bT_10MS_F     R_Time_Flag,2

R_500US_COUNT     EQU     h'01'
R_10MS_COUNT      EQU     h'02'

R_GOTOLOWPOWERMODE_COUNT      EQU     h'03'

;================================================================
; T0 BASE TIME  SET
#define T0_125_US                   125   ;125US(1 TIMER0 CLOCK) < T0_BASE_MS <4.096MS
#define T0_LOAD_C_Init              T0_125_US*16000000/1000000 ; FOSC 1:1

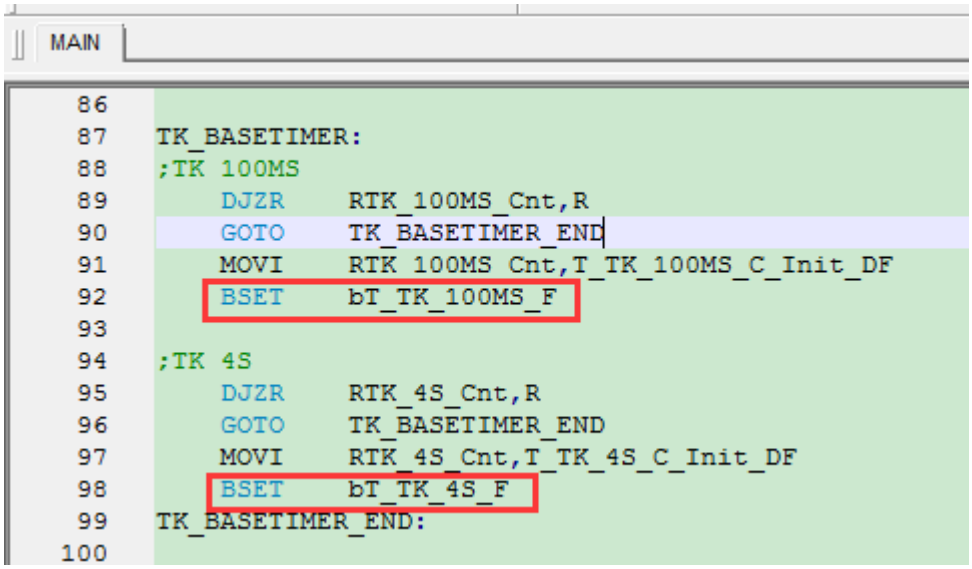
#define T_500US_C_Init_DF           D'4'
#define T_10MS_C_Init_DF            D'20'
#define T_GOTOLOWPOWERMODE_C_Init_DF  D'250'  ;无触摸下超时休眠检测次数，无触摸下超时休眠时间 = 次数*触摸通道检测数*0.5ms

;-----------------------------------------------------------------
#define RTK_100MS_Cnt               _RTkBuff4
#define RTK_4S_Cnt                  _RTkBuff5

#define T_TK_100MS_C_Init_DF        D'200'  ; 触摸更新时基 0.5ms*200 = 100ms
#define T_TK_4S_C_Init_DF           D'40'   ; 触摸长按清楚检测时基 100ms*40 = 4S

```

100ms 和 4s 是触摸库里面需要的时基，用户需要自己在程序里面定义，而且需要有这两个时间产生的标志。如下图：



用户设置完成可以测试下时基是否设置成功，我们测试的方法是：

100ms时基的测试方法： 手指触摸按键，然后再上电，上电等一段时间后，再次触摸按键是否有效，有效则说明100ms的时基设置OK了。

4s时基的测试方法： 打开触摸库里长按清除功能TKLib_LONG_TOUCH_TIME，一直触摸按键超出设定时间时，按键标志是否清零，如果清零则说明4s时基设置OK了。

2.2.3 在程序中调用库

MAIN

```
146 ;=====================================================================
147 MainProgram:
148     call    Init                                                    1
149
150 MainLoop:
151     CALL    _TkScan          ;主循环一直扫描，故主循环间隔不宜过长以免影响响应时间
152     JBTS1   bOneScanDone     ;查询一轮是否转换结束
153     goto    MainLoop_1                                              2
154     call    _TkSignalHandle
155     BCLR    bOneScanDone     ;一轮转换完成,清除标志
156
157 #if TKUartDebugLib_DEBUG_MODE_SWITCH
158 #if(TKUartDebugLib_DEBUG_UART_SELECT==0)
159     BCLR    bTC0IE
160 #endif
161
162     call    _TK_TouchUartDebug                                      3
163
164 #if(TKUartDebugLib_DEBUG_UART_SELECT==0)
165     BSET    bTC0IE
166 #endif
167 #endif
168
169 ; 按键响应
170     CJZ2 _RKeyOnFlgL,_RKeyOnFlgH,ALLKEY_UP_Handle  ;检查是否有按键被按下
171     MOVI R_GOTOLOWPOWERMODE_COUNT,T_GOTOLOWPOWERMODE_C_Init_DF
172
173 ;    jbts0   F_TK0_On
174 ;    bclr    iob,0
175 ;    jbts0   F_TK2_On
176 ;    bclr    iob,1                                                  4
177 ;    jbts0   F_TK4_On
178 ;    bclr    iob,2
179     goto    ALLKEY_UP_Handle_end
180
181 ALLKEY_UP_Handle:
182     DJZR    R_GOTOLOWPOWERMODE_COUNT,R
183     GOTO    ALLKEY_UP_Handle_end
184     MOVI    R_GOTOLOWPOWERMODE_COUNT,T_GOTOLOWPOWERMODE_C_Init_DF   5
185     BSET    bLowPowerModeRun
186
187 ALLKEY_UP_Handle_end:
188
```

```
188
189 #if TKLib_TouchLowerPower
190     JBTS1   bLowPowerModeRun
191     goto    Next_OneScan
192
193 ;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)
194 ;如低功耗模式下端口处理，外设模块等
195 ;WDT 与 CPU 工作模式 用户不需要处理
196
197
198     BCLR    INTCR0,TC0IE   ;关闭其他唤醒源，本demo code外部只打开了T0，固只关闭T0
199 ;触摸低功耗模式
200     call    _TkLowPowerMode
201
202 ;下面加唤醒后其它需要恢复的操作
203     BSET    INTCR0,TC0IE
204
205 #endif
206
207 Next_OneScan:
208     CALL    _ScanContinue
209
210 ; 其它主循环程序
211 MainLoop_1:
212     CLRWDT
213
214     goto    MainLoop
215
```

6

7

8

1：框 1 中，先 Init 初始化，下面会做初始化流程介绍。
2：框 2 中，查询是否完成一轮扫描，完成后清除相关标志及调用_TkSignalHandle 信号处理程序。
3：框 3 中，串口辅助调试部分，不用管，如果不需要这部分也可以直接不用。
4：框 4 中，有触摸按键被触摸，判断触摸给出的标志来判断是哪个按键工作。
5：框 5 中，没有按键，做一个计数达到多少就置休眠标志(如果不需要低功耗就不要置此标志)。
6：框 6 中，触摸低功耗程序处理部分，如果不需要进低功耗此程序可以不需要。
7： 框 7 中， 触摸扫描启动程序，先调用 TkSignalHandle，再清除 bOneScanDone，后调用 ScanContinue。
8：框 8 中，清楚看门狗，然后循环扫描。

2.2.4 触摸初始化 Init 介绍说明

MAIN

```
;=====================================================================
init:
; Stack:    Level 5
IO_init:
    movia   b'11111111'  ;1: output  0: input
    movar   oea
    clrr    ioa
    movia   b'11111111'  ;1: output  0: input
    movar   oeb
    clrr    iob
    movia   b'11101111'  ;1: output  0: input
    movar   oec
    clrr    ioc
    movia   b'11111111'  ;1: output  0: input
    movar   oed
    movia   b'11111111'  ;1: output  0: input
    movar   iod

RAM_INIT:
ClrRam:
    clrr    fsr0     ;0x00
ClrRam_loop:
    clrr    indf0
    incr    fsr0,r
    movr    fsr0,a
    jcmpai  low(0xff+1)    ;0xff (clr ram addar +1)
    goto    ClrRam_loop
ClrRam_end:

; RAM 数据赋值
    MOVI    R_500US_COUNT,T_500US_C_Init_DF
    MOVI    R_10MS_COUNT,T_10MS_C_Init_DF
    MOVI    R_GOTOLOWPOWERMODE_COUNT,T_GOTOLOWPOWERMODE_C_Init_DF
RAM_INIT_END:
```

1

2

3

```

TKM0_INIT:
#if TKUartDebugLib_DEBUG_MODE_SWITCH
    call    _TK_TouchUartDebug_Init
#endif

;AD
    movia   b'00001110'  ; adc ch = gnd
    movar   adcon0
    bset    adcon1,0  ; CDC VREF  ;调用TkInit此程序前，必须打开ADC模块2V/3V/4V之任一参考电源
;    bset    adcon1,1  ; CDC VREF
;TK
    call    _TkInit
TKM0_INIT_END:

T0_INIT:
    CLRR    TC0CH
    CLRR    TC0CL

    MOVIA   high(T0_LOAD_C_Init)
    MOVAR   TC0PRH
    MOVIA   low(T0_LOAD_C_Init)
    MOVAR   TC0PRL

    MOVIA   b'10001000' ;UP FOSC 1:1
    MOVAR   T0CR
    BSET    bTC0IE

T0_INIT_END:
;ENALBE gie
    bset    option,gie
    return
```

4

5

6

1：框 1 中，初始化 IO 口配置。
2：框 2 中，清零所以 RAM 寄存器。
3：框 3 中，初始化特殊 RAM 的数据。
4：框 4 中，初始化串口，ADC 参考，库里面的_TkInit。
5：框 5 中，初始化 TC0 用来做时基。
6：框 6 中，开启总中断标志 GIE。

3库文件及配置说明

文件名称	功能说明	备注
M8Pxxx_Touch_M0_Lib_CFG_V202.inc	触摸参数配置文件	根据需求修改
M8Pxxx_Touch_M0_Lib_Data_V202.inc	触摸参数原型声明及编译提示信息文件	不允许修改
M8Pxxx_Touch_M0_Lib_V202.inc	触摸库外部调用声明文件	不允许修改
M8Pxxx_Touch_M0_UartDebug_LIB_Init.inc	触摸库 UART 设置文件	根据需求修改
M8Pxxx_Touch_M0_Lib_V202.A	触摸库文件	无法修改

3.1 M8Pxxx_Touch_M0_Lib_CFG_V202.inc

此文件包含用户配置触摸库时所有可能修改到的定义信息。

3.1.1 触摸通道设置

设置 CDC 模块 0 需要检测的通道　x=0: 该通道按键不检测；x=1: 该通道按键需检测。

定义名称			说明
#define	EnTk0	x	表示 TK0 通道是否检测
#define	EnTk1	x	表示 TK1 通道是否检测
#define	EnTk2	x	表示 TK2 通道是否检测
#define	EnTk3	x	表示 TK3 通道是否检测
#define	EnTk4	x	表示 TK4 通道是否检测
#define	EnTk5	x	表示 TK5 通道是否检测
#define	EnTk6	x	表示 TK6 通道是否检测
#define	EnTk7	x	表示 TK7 通道是否检测
#define	EnTk8	x	表示 TK8 通道是否检测
#define	EnTk9	x	表示 TK9 通道是否检测
#define	EnTk10	x	表示 TK10 通道是否检测

3.1.2 TKLib_TOUCH_NUM

设置确认允许响应的最多按键数量(1 <= value <= TKLib_M0_OPENED_TK_AMOUNT)。

3.1.3 TKLib_Debounce_TIME

设置触摸消抖次数 (1 <= value <= 255) 。
消抖时间 = TKLib_Debounce_TIME * BASE_TIME * TKLib_M0_OPENED_TK_AMOUNT，
其中 BASE_TIME 为 TkScan 调用间隔，其中本例程 BASE_TIME 为 1MS；
TKLib_M0_OPENED_TK_AMOUNT 为开启的按键数量。

3.1.4 TKLib_LONG_TOUCH_TIME

设置手指触摸最长有效时间 (0<= value <=15) 。
0: 屏蔽此功能；1: 禁止配置；2~15：时间 = value*4S，偏差 4S。
这个 4S 的来源就是上面主程序里面定义的 4S 的时基。

3.1.5 TKLib_Calibration_TOUCH_TIME

设置自动校准时间 (1<= value <=15)；时间 = value*100ms。
这个 100ms 的来源就是上面主程序里面定义的 100ms 的时基。

3.1.6 TKLib_M0_TKCKS

设置 CDC 时钟分频选择位。
0->Fosc/4, 1->Fosc/8, 2->Fosc/16, 3->Fosc/32；一般不需要修改。

3.1.7 TKLib_EFFECTIVE_TOUCH_TIME

设置手指有效触摸检测次数 (1<= value <=15)；一般不需要修改。

3.1.8 TKLib_INVALID_TOUCH_TIME

设置手指无效触摸检测次数 (1<= value <=15)；一般不需要修改。

3.1.9 TKLib_FILTER_LEVEL

设置触摸采样滤波等级 (0<= value <=2)；一般不需要修改; 0->无,1->中,2->高。

3.1.10 触摸阈值设置

触摸阈值： 取值需具体应用环境决定，最小 9，最大 255，值越小越灵敏。

定义名称	说明
TKLib_TK0_F_THRESHOLD	表示 TK0 通道触摸阈值
TKLib_TK1_F_THRESHOLD	表示 TK1 通道触摸阈值
TKLib_TK2_F_THRESHOLD	表示 TK2 通道触摸阈值
TKLib_TK3_F_THRESHOLD	表示 TK3 通道触摸阈值
TKLib_TK4_F_THRESHOLD	表示 TK4 通道触摸阈值
TKLib_TK5_F_THRESHOLD	表示 TK5 通道触摸阈值
TKLib_TK6_F_THRESHOLD	表示 TK6 通道触摸阈值
TKLib_TK7_F_THRESHOLD	表示 TK7 通道触摸阈值
TKLib_TK8_F_THRESHOLD	表示 TK8 通道触摸阈值
TKLib_TK9_F_THRESHOLD	表示 TK9 通道触摸阈值
TKLib_TK10_F_THRESHOLD	表示 TK10 通道触摸阈值

3.1.11 TKLib_TouchLowerPower

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。

3.1.12 触摸模块设置

设置 CDC 模块 0 需要检测的通道　0: 该通道按键不检测；1: 该通道按键需检测。
休眠模式下是按模块来扫描的，这个更省电。

定义名称	说明
TKLPGroup0	表示模块 0 是否检测
TKLPGroup1	表示模块 1 是否检测
TKLPGroup2	表示模块 2 是否检测

3.1.13 触摸通道模块分组设置

低功耗下触摸通道分组设置， 数据 = {tk0chsh:tk0chsl}，若触摸按键较多时，建议分组检测分组原则要求需要检测的组通道数应相同，减少不同组之间的检测差异。
例：若产品 tk0~tk6 为开启检测的通道，分 2 组，7/2 = 3 余 1，则建议 4 个一组，另一组 3 个。即一组 TKLib_M0_TKLPGroup0_MASK = 0x000F; 另一组 TKLib_M0_TKLPGroup0_MASK = 0x0078

定义名称	说明
TKLib_M0_TKLPGroup0_MASK	0x0007
TKLib_M0_TKLPGroup1_MASK	0x0038
TKLib_M0_TKLPGroup2_MASK	0x003F

3.1.14 TKLib_Sleep_F_THRESHOLD

触摸唤醒休眠阈值： 取值需具体应用环境决定，最小 3，最大 255，值越小越灵敏。

注：M8P634 和 M8P626 是模块唤醒的，是几个按键加起来的值大于休眠阈值就唤醒休眠。M8P615 和 M8P515 是单独按键唤醒，建议设置最小按键阈值减 3 到 5 个值(例：三个触摸按键的阈值分别是 10、15、20 那么设置唤醒休眠阈值应该是 10-3=7)。

3.1.15 TKLPLib_RCOUT_EN

触摸低功耗模式下辅助端口是否使能　0：不使用辅助端口；1：使用辅助端口此端口用来调整触摸低功耗模式下触摸检测灵敏度，可接电阻或接电容，具体接法请查看触摸低功耗电路设计参考原理图；如每组分配的通道数较多时，低功耗下触摸检测灵敏度会较低，此时可以考虑辅助端口来调节低功耗模式下触摸检测灵敏度。

3.1.16 辅助端口设置

触摸低功耗模式下，辅助端口选择。

定义名称			说明
#define	OE_RCOUT	OEC	对应端口方向寄存器
#define	P_RCOUT	IOC	对应端口数据寄存器
#define	N_RCOUT	4	对应端口编号

3.1.17 用户不可修改部分

TKLib_M0_OPENED_TK_AMOUNT
TKLib_M0_TK_EN_MASK
TKLib_M0_LP_OPENEDGROUP_AMOUNT
TKLib_M0_LP_TKLPGROUP_MASK

> 注：以上定义用户不可修改。

3.2 M8Pxxx_Touch_M0_Lib_Data_V202.INC

此文件用于声明库里面相关数据和配置信息，不建议用户修改，否则可能无法编译通过。

3.3 M8Pxxx_Touch_M0_Lib_V202.INC

此文件用于介绍库里面的子程序声明及触摸的一写标志位。
触摸库中有用到间接寻址寄存器(FSR0,INDF0,FSR1,HBUF)，若客户在中断中有使用，请用户进入中断后先备份(push 指令后备份)此 4 个寄存器，出中断后再恢复(pop 指令前恢复)。

3.3.1 _TkInit

触摸模块初始化程序，堆栈深度 4 级。

> 注： M8P615，M8P632，M8P634 需要在调用 TkInit 之前打开 ADC 模块 2V/3V/4V 之任一参考电源

3.3.2 _TkScan

触摸按键扫描程序，堆栈深度 3 级，必需保证主循环中调用。

3.3.3 _TkSignalHandle

触摸数据处理程序，堆栈深度 3 级，bOneScanDone 被置"1" 后，调用一次。

3.3.4 _ScanContinue

触摸扫描启动程序，堆栈深度 3 级，先调用 TkSignalHandle，先清除 bOneScanDone，后调用 ScanContinue。

3.3.5 _BaseLineReset

基线强制复位程序，堆栈深度 3 级。

3.3.6 _TkLowPowerMode

触摸模块低功耗程序，堆栈深度 3 级，要求_ScanContinue 程序必须跟随在后面调用此函数前必须将相关端口设置。

3.3.7 _TK_TouchUartDebug_Init

触摸串口调试初始化程序。

3.3.8 _TK_TouchUartDebug

触摸串口调试程序，堆栈深度 3 级。

3.3.9 bOneScanDone

触摸扫描一轮完成标志，RTkBitFlag 中其它 BIT 位用户请勿操作。

3.3.10 bT_TK_100MS_F

触摸更新的时基标志(默认是 100ms，用户可以修改 100ms 时基来调节大小)。

3.3.11 bT_TK_4S_F

长按清除的时基标志(默认是 4s，用户可以修改 4s 时基来调节大小)。

3.3.12 bLowPowerModeReset

触摸低功耗模式复位标志　0: Reset Touch LowPower Mode 1; LowPower Mode Reset complete。

3.3.13 bLowPowerModeRun

触摸低功耗模式运行标志　0: Normal; 1: LowPower。

3.3.14 触摸按键触摸响应标志

TK 按键触摸响应标志　1：表示被触摸；0：未被触摸。

用户只读，不可写入操作，切记！

定义名称			说明
#define	F_TK0_On	_RKeyOnFlgL,0	TK0 按键触摸响应标志
#define	F_TK1_On	_RKeyOnFlgL,1	TK1 按键触摸响应标志
#define	F_TK2_On	_RKeyOnFlgL,2	TK2 按键触摸响应标志
#define	F_TK3_On	_RKeyOnFlgL,3	TK3 按键触摸响应标志
#define	F_TK4_On	_RKeyOnFlgL,4	TK4 按键触摸响应标志
#define	F_TK5_On	_RKeyOnFlgL,5	TK5 按键触摸响应标志
#define	F_TK6_On	_RKeyOnFlgL,6	TK6 按键触摸响应标志
#define	F_TK7_On	_RKeyOnFlgL,7	TK7 按键触摸响应标志
#define	F_TK8_On	_RKeyOnFlgH,0	TK8 按键触摸响应标志
#define	F_TK9_On	_RKeyOnFlgH,1	TK9 按键触摸响应标志
#define	F_TK10_On	_RKeyOnFlgH,2	TK10 按键触摸响应标志

3.4 M8Pxxx_Touch_M0_UartDebug_LIB_Init.INC

此文件用于触摸串口调试的相关设置，需要注意的是有些触摸芯片由于空间不够，串口调试的程序是另外独立的。

3.4.1 TKUartDebugLib_DEBUG_MODE_SWITCH

触摸串口调试开关。

0：禁止编译时加载触摸调试程序，触摸调试完成后，请设为“0”。

1：允许编译时加载触摸调试程序，此时可以通过触摸调试工具及配套软件获取内部相关数据，方便用户开发时调试触摸参数。

3.4.2 TKUartDebugLib_DEBUG_UART_SELECT

触摸串口方式选择　0：软件模拟串口 1：MCU 硬件串口。

3.4.3 TKUartDebugLib_BAUDRATE

触摸串口调试波特率设定，不建议修改，必需修改时，请用户自行确认所设波特率是否正常软件模拟串口时,若 Fcpu 速率较低时，若发现通讯不正常，请降低波特率 MCU 硬件串口时，波特率最高可达 2M，且与 Fcpu 无关，但使用高速波特率时，请确认使用的串口工具是否支持。

3.4.4 模拟串口软件端口选择

定义名称			说明
#define	OE_UART_TX	OEB	对应端口方向寄存器
#define	P_UART_TX	IOB	对应端口数据寄存器
#define	N_UART_TX	6	对应端口编号

3.4.5 不可修改部分

```
#define     R_UART_TX_CNT     _RTkTemp9
#define     R_UART_TX_BUF     _RTkTemp10
```

3.5 M8Pxxx_Touch_M0_Lib_V202.A

此文件为触摸库文件，版本为 V202。

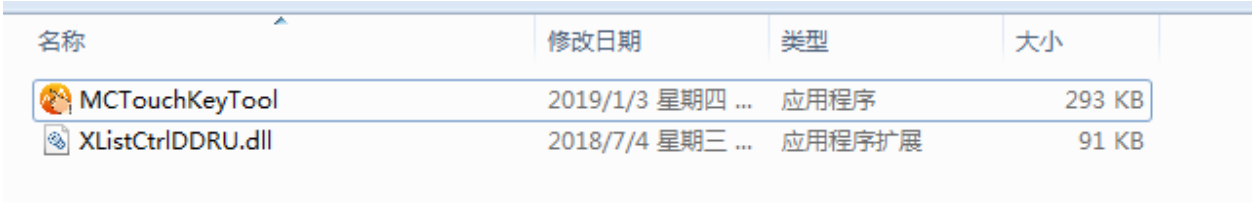
4 触摸串口调试工具使用说明

此工具是帮助客户在调试触摸程序之前对触摸 PCB 加上外壳模具的灵敏度做一个初步的测试，可以初步看到隔着模具触摸按键时，按键所引起的触摸阈值的变化，一般建议触摸软件里设置阈值取串口调试工具看到变化阈值的 50%~80%。

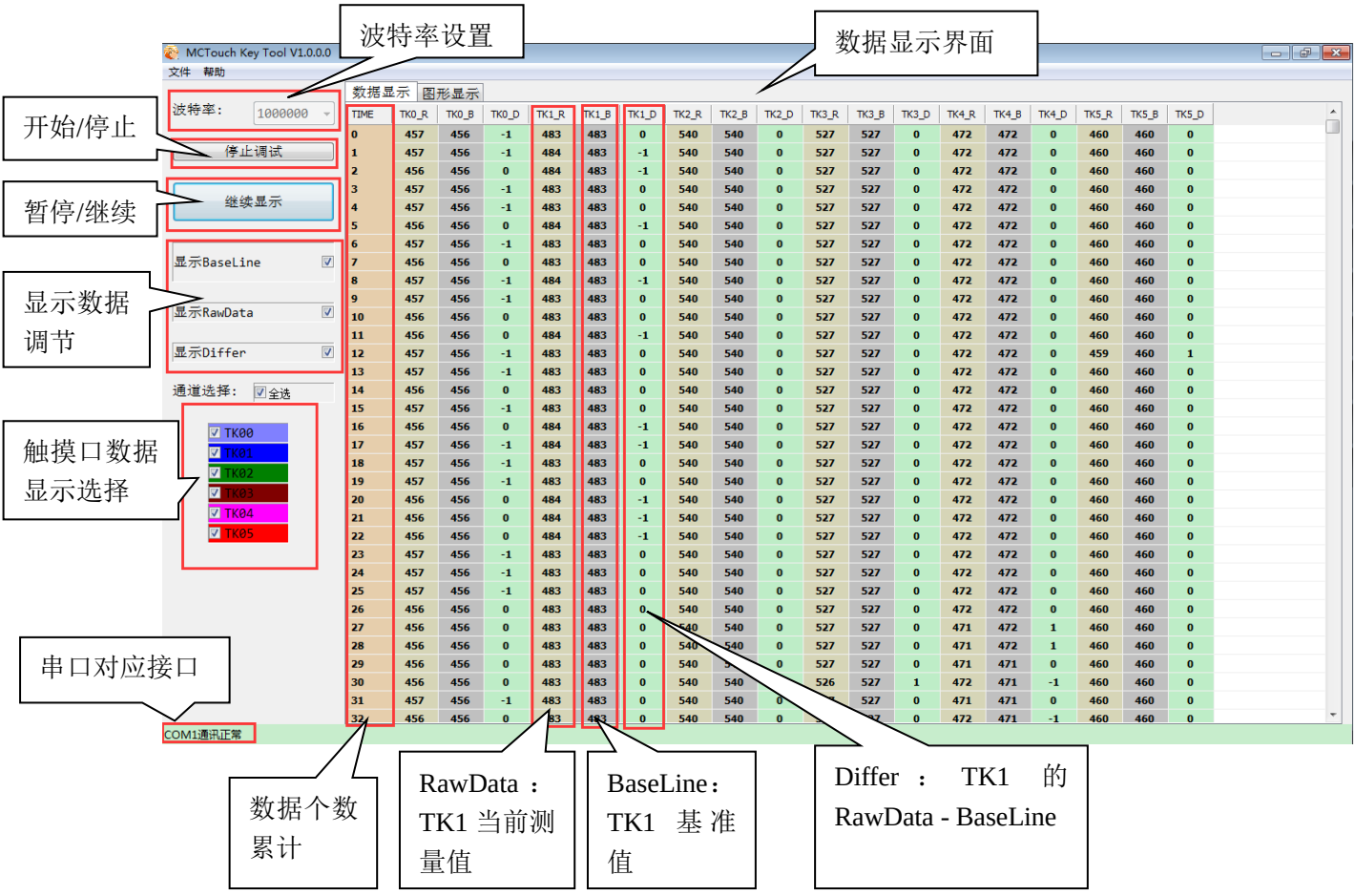
例：触摸串口调试工具看到手指触摸按键的变化有 30 个值，那么触摸软件里面对应按键的触摸阈值设置为 30/2=15。

4.1 触摸串口调试工具文件

触摸工具是 MCTouchKeyTool；另外的.dll 文件是动态库，由于系统的不一致需要稍作更改。



4.2 触摸串口调试工具界面





注：程序软件里面设置的触摸阈值是跟Differ的值相关的，所以一般建议触摸软件里设置阈值取串口调试工具看到Differ(变化值)的50%~80%。

5 触摸PCB设计注意事项

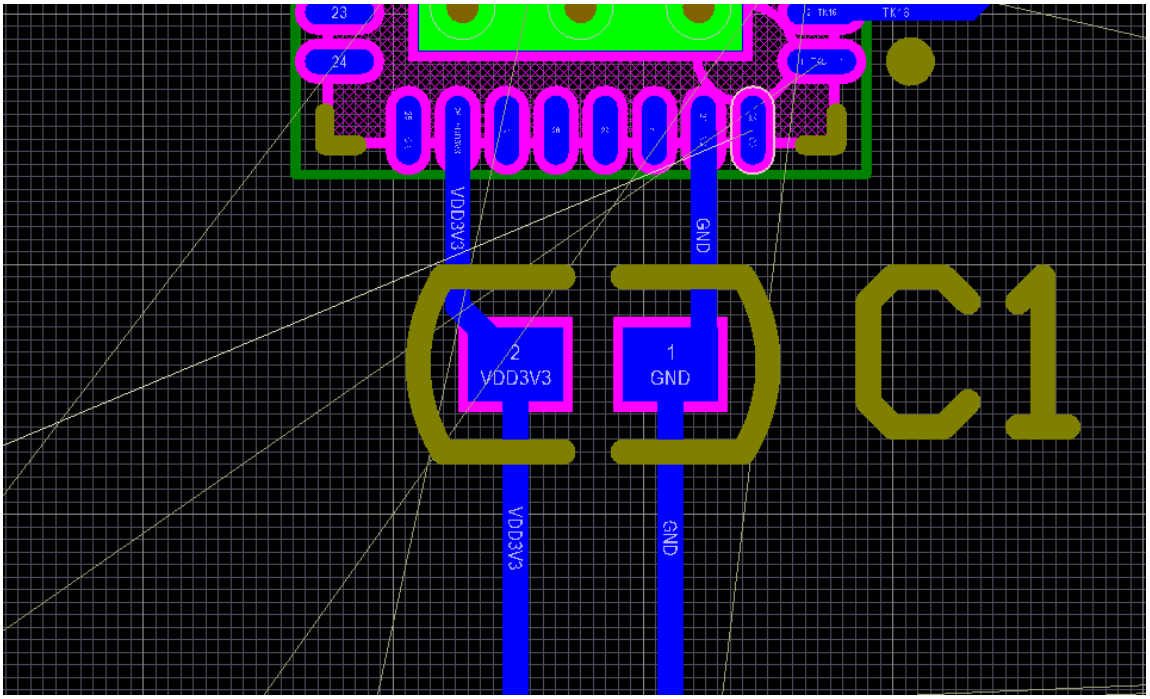
针对 PCB 设计时触摸走线、覆铜、电容摆放位置等一些注意事项做一些说明。

5.1 整体布局要求

5.1.1 MCU 电源两端电容

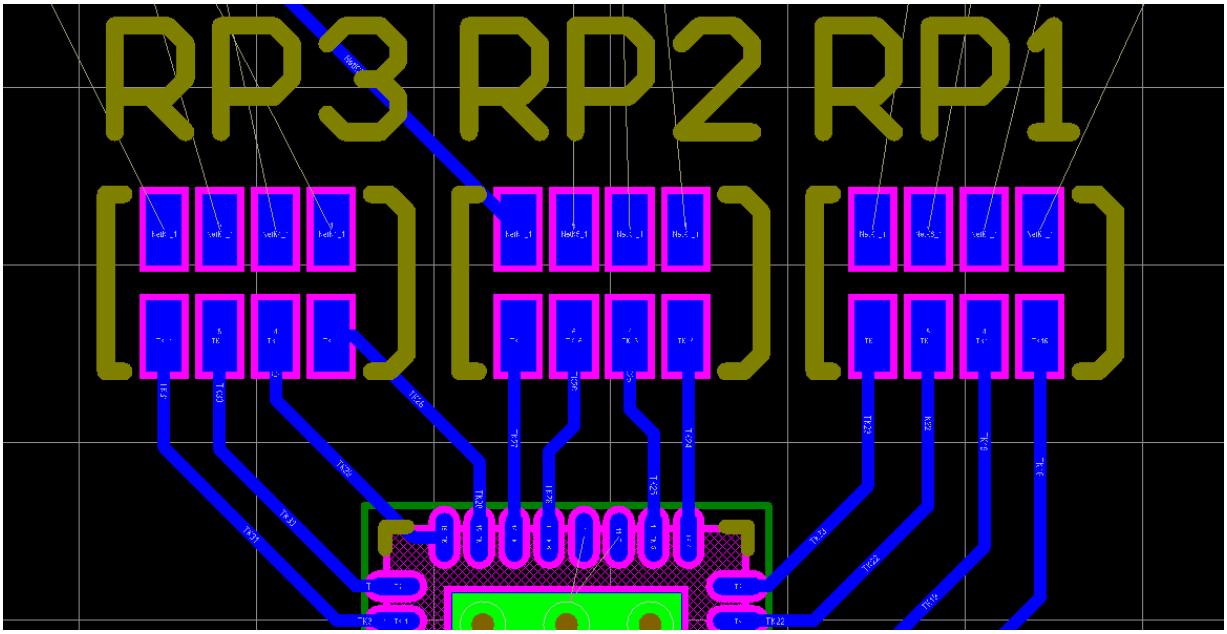
MCU 电源端加 104 电容，**电容位置应靠近 MCU，且 PCB 布线时，电源线应先流进 104 电容后再进 MCU 电源脚，104 电容到芯片的电源线要越短越好**（**注：进入 MCU 的电源要注意尽量避免有很大的干扰和不稳定现象**）。

如下图：



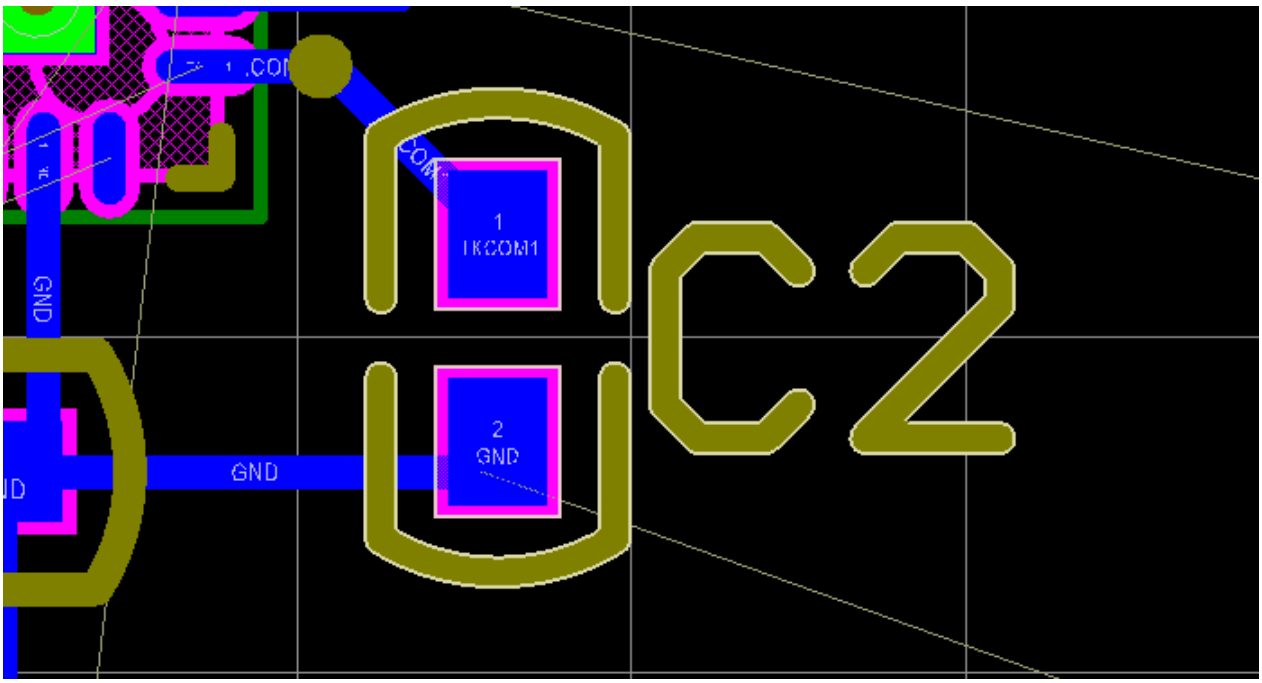
5.1.2 触摸线串电阻

触摸通道有条件时，应串一个 1K~5K 电阻，**电阻位置也应靠近 MCU 触摸通道管脚**，这可以提高触摸抗干扰性能。如下图：



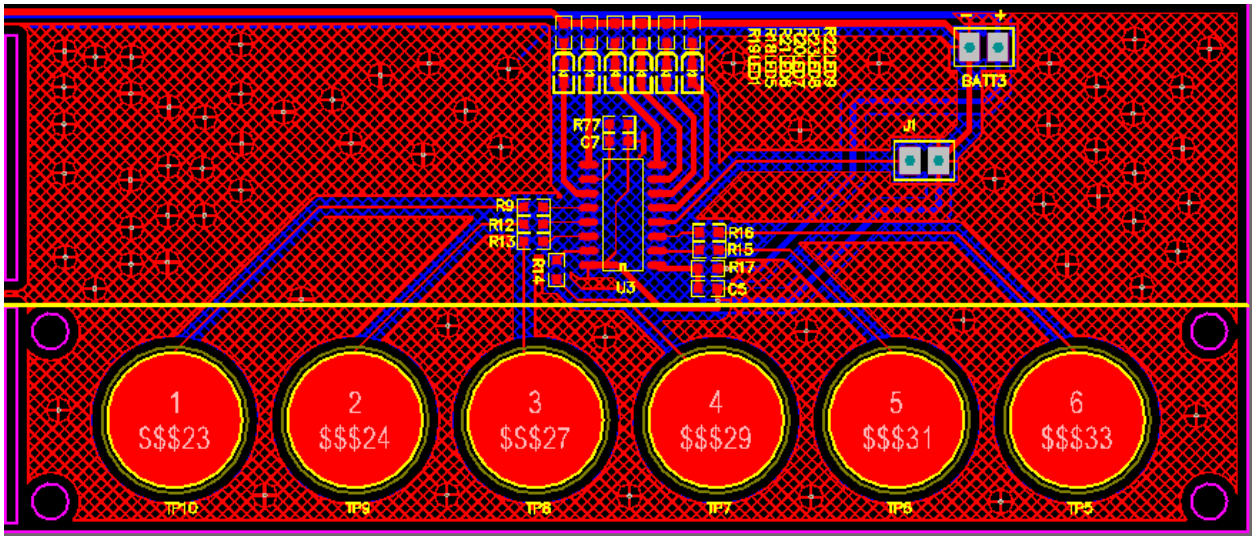
5.1.3 触摸参考电容位置

触摸参考电容位置应靠近 MCU，TKCOM 管脚到触摸参考电容的走线越短越好，触摸参考电容另一端到 MCU GND 地管脚回路尽量短。如下图：



5.1.4 MCU 位置摆放注意

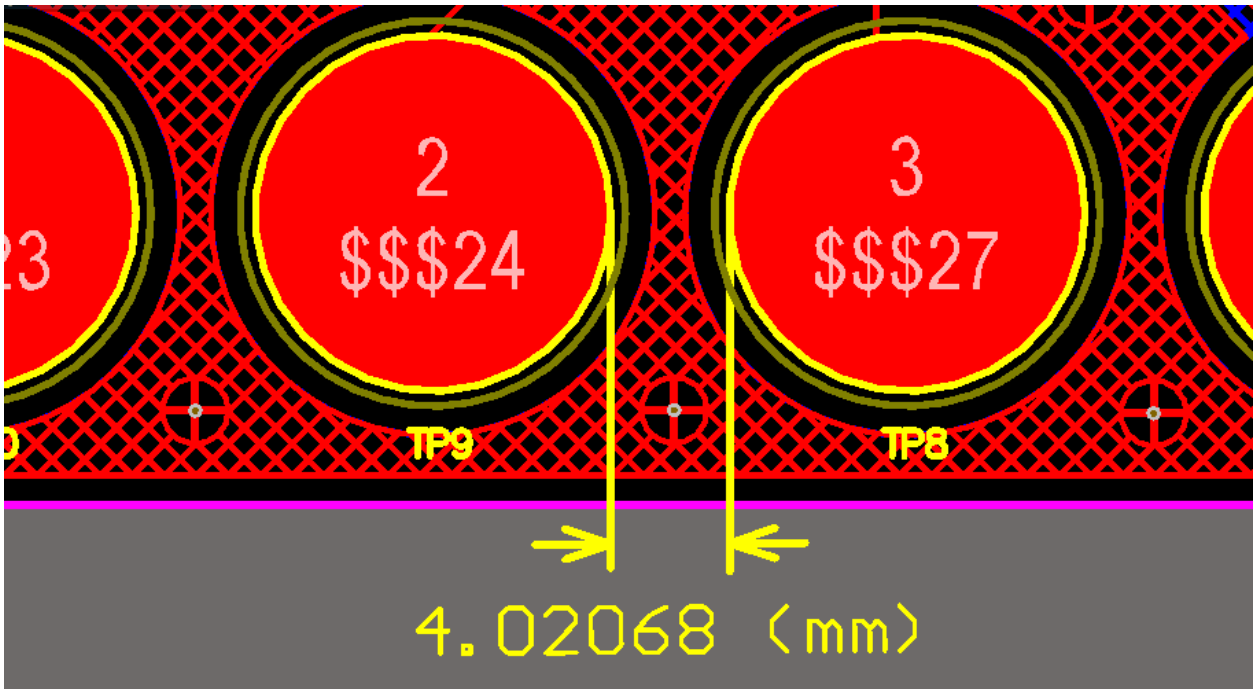
触摸 MCU 摆放应平衡到各个感应按键的距离，即 **MCU 应尽量摆放在触摸按键中间，以降低不同按键间灵敏度差异**。如下图：



5.1.5 触摸按键间距注意

触摸按键间间距应保持一定距离(大于 3mm 以上)，减少按键间相互影响(滑条，滚轮例外)。

如下图：



5.2 布线要求

5.2.1 触摸按键走线

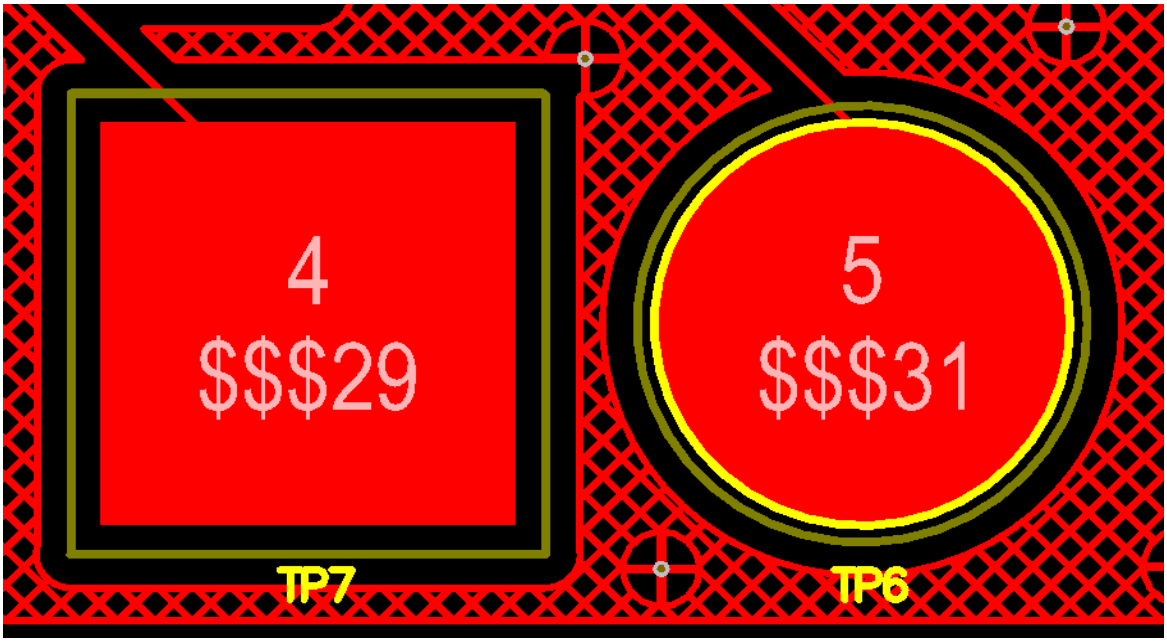
1、触摸按键走线应尽量短，在 PCB layout 时，应优先走触摸线，且尽量减少过孔的使用(最好过孔控制在 3 个以内)，减少干扰引入。
2、触摸按键走线应尽量窄，推荐线宽 6-15mil(做板工艺允许的话越细越好)。
3、触摸按键间走线可并行，但至少 1 倍触摸走线宽度间距。
4、触摸走线原则上不允许与其它数字信号或驱动电源线并行，考虑到实际情况，若必须并行时应使并行距离尽量短，并行线之间至少 2 倍以上触摸走线线宽间距，或走线间插入地线隔离。对于双面板，应尽量让并行信号线改向，如垂直跨过。
5、触摸按键或走线在进行覆铜操作时，应注意与覆铜间距离，具体后面单独介绍。
6、其它非触摸线步线时应适用 EMC 布线规范，保障系统抗干扰能力。

> **注：并行的数字信号线不能为高频信号线，切记！**

5.3 触摸按键形状

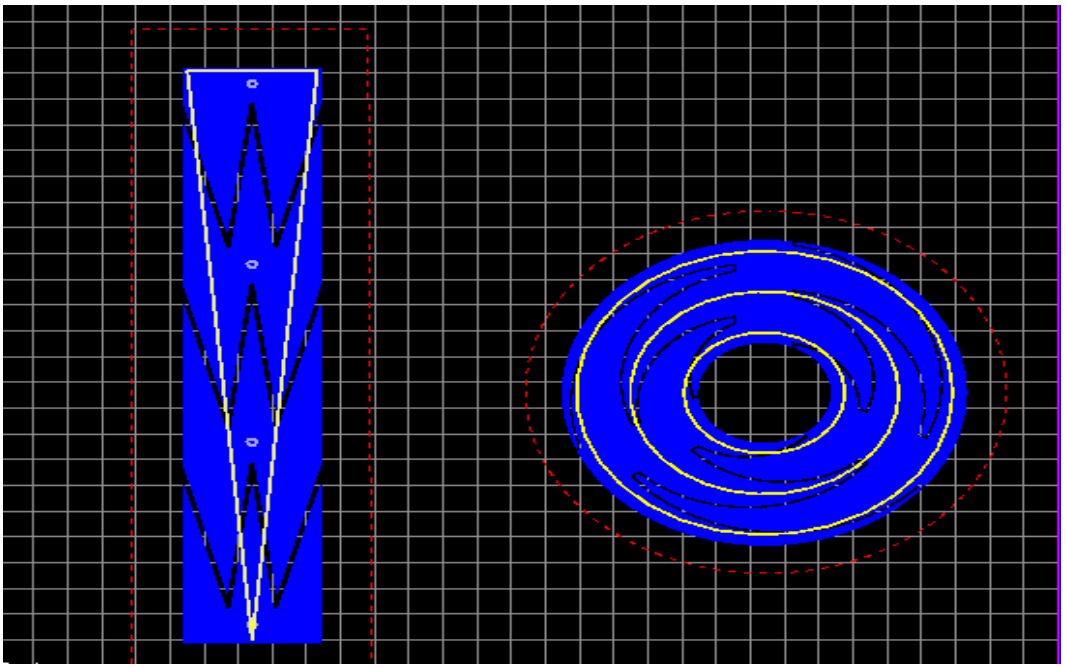
5.3.1 普通触摸按键

针对在 PCB 板上的常规感应按键，如利用 PCB 铜箔做感应按键，应尽量选用常规形状，常规形状包括 圆形，正方形。感觉按键电极面积要适当，以正方形为例，建议 8*8mm~15*15mm 间。且感应焊盘应被阻焊油覆盖，即不露铜。如下图：



5.3.2 滑条滚轮

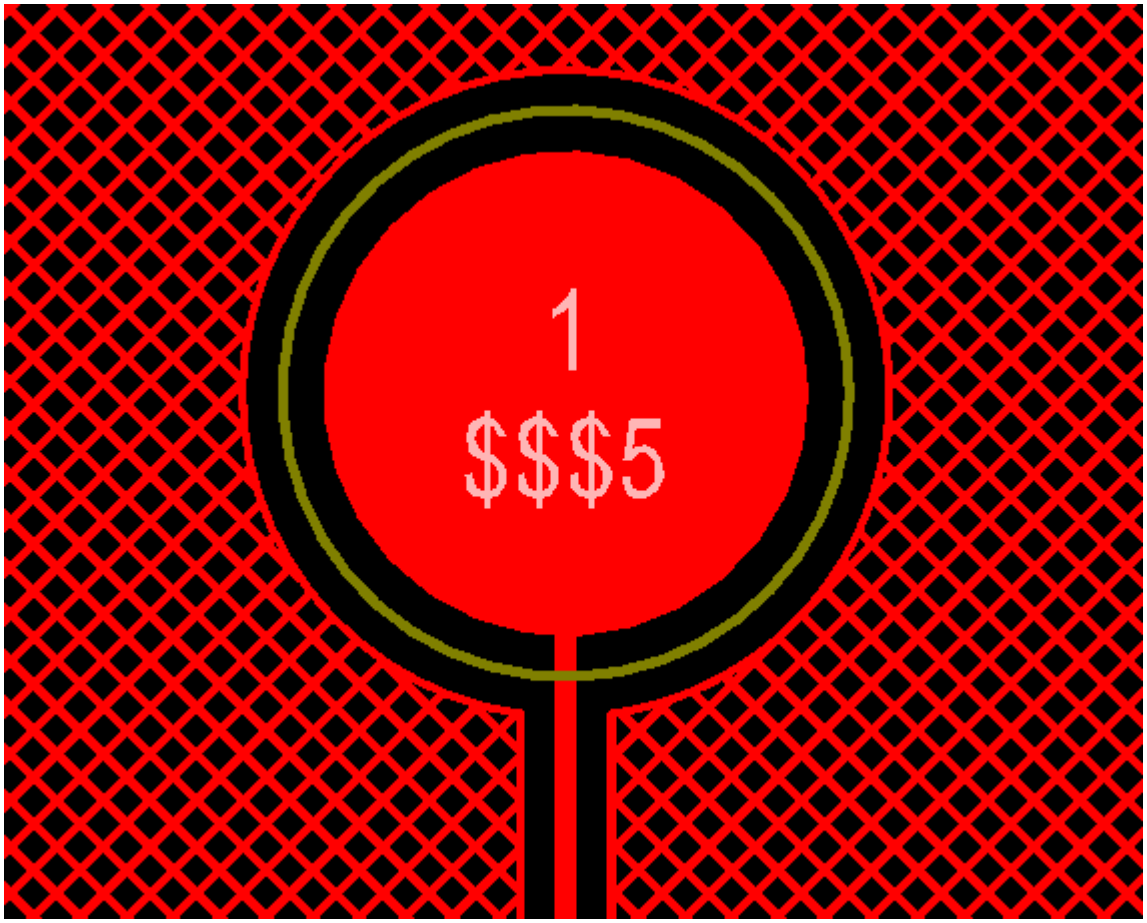
滑条或滚轮，按键电极间间隙应在 0.2~0.5mm 间隙间，不可太近，也不可太远，锯齿间要充分交错。如下图：



5.4 实际布板参考

5.4.1 参考一单面空间够

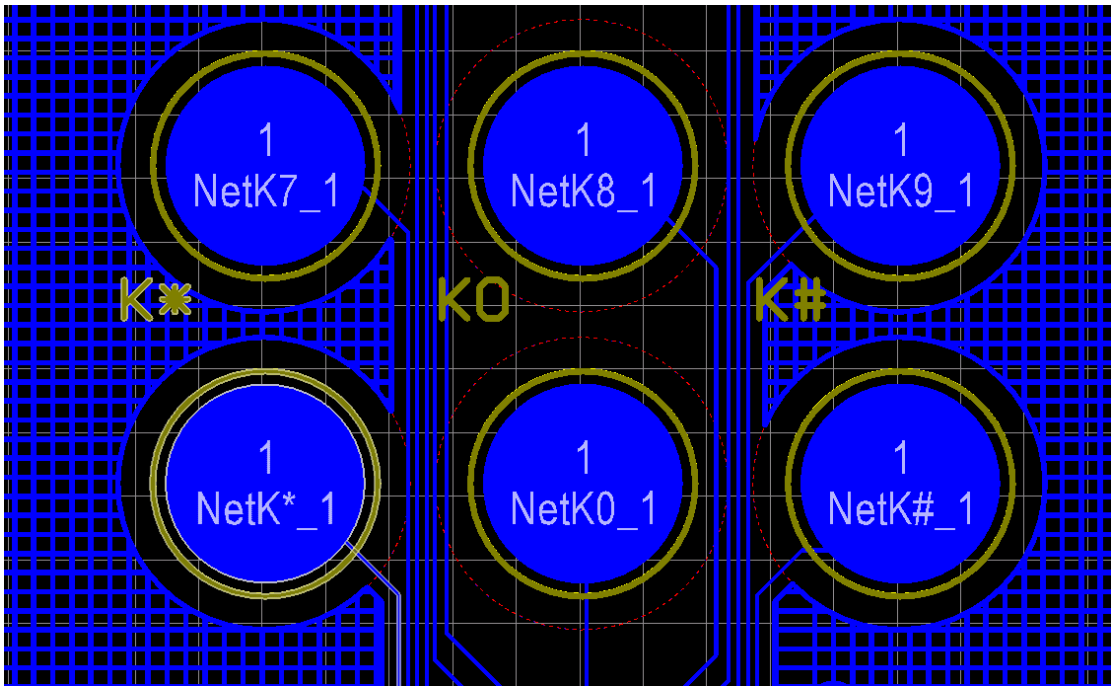
单面板，空间足够，且铜箔层加覆盖物



如上，触摸走线 10mil 按键大小直径 9mm，在覆铜时，按键周围 100mil 内不覆铜，触摸走线 20mil 内不覆铜，覆铜采用 30%网格地。**注：单面板时，必须在正面走线覆铜，以避免手指压在触摸走线时意外感应**。

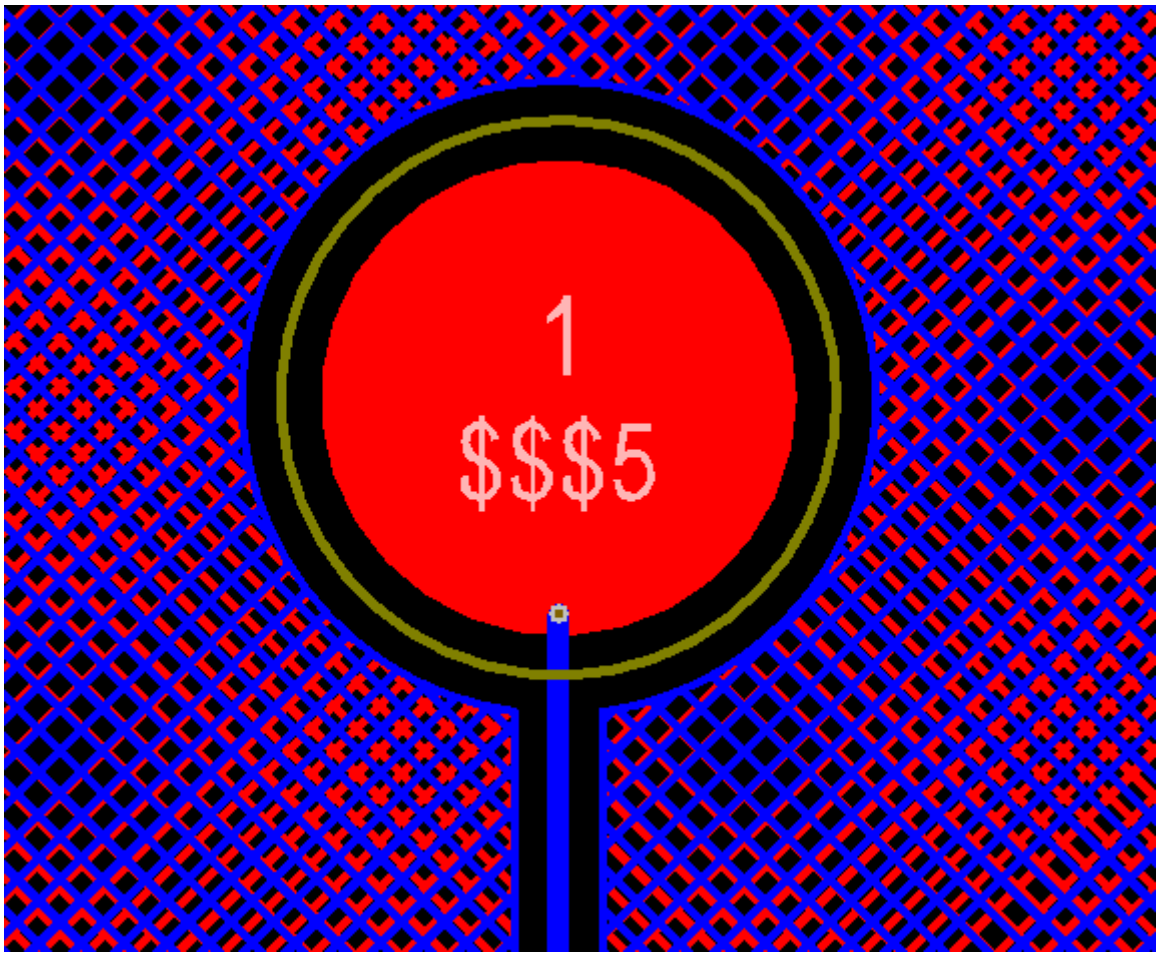
5.4.2 参考二单面空间紧凑

单面板，空间紧凑，且铜箔背面覆盖物



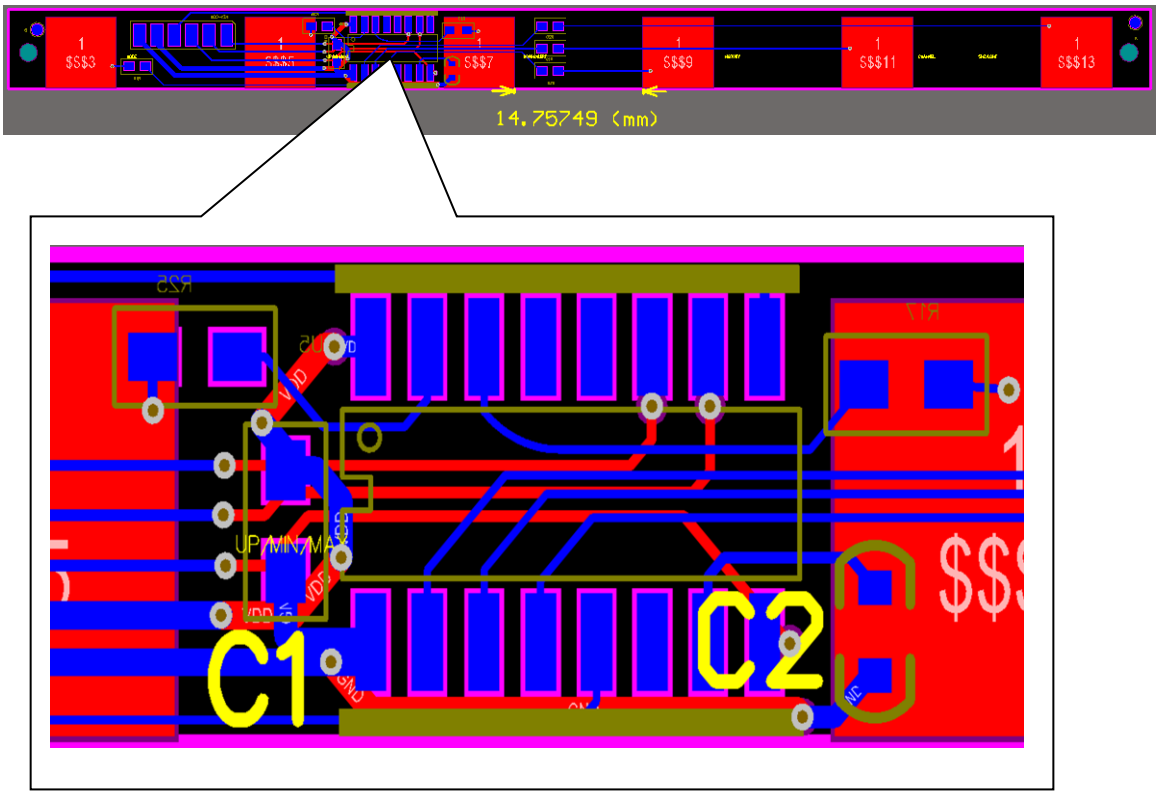
如上，是典型数字键盘PCB布局样式，在此类型PCB布局时，应尽量使触摸线间距靠近，其他周围能覆铜，则30%网格地覆铜，或不覆。同样若单面板，铜箔面加覆盖物，规则相同。

5.4.3 参考三双面板空间够



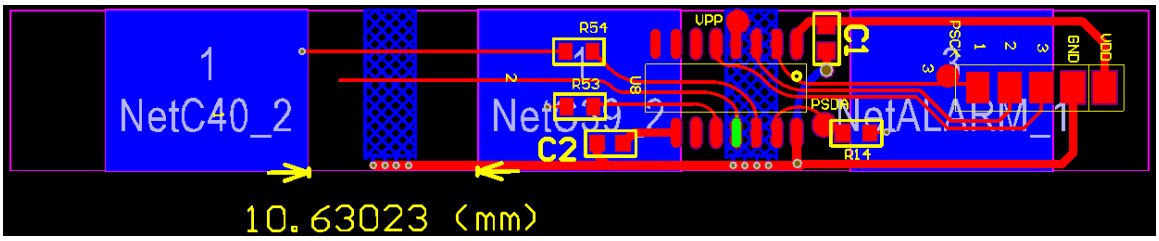
如上，TOP 层加覆盖物，PCB 正反面覆铜度采用 30%网格地(为了让图片看出来我更改两面的覆铜网格的比例可以更清楚看到正反面覆铜)，触摸走线采用覆盖物背面走线。此时，TOP 层也可不覆铜，如果 TOP 层覆铜最好不要盖着触摸走线(就是触摸走线的正反面都要不被覆铜盖着留相同或者更大的距离)，覆铜最好与触摸走线成 45 度(**覆盖物较厚时,如亚克力大于 3mm，TOP 层不覆铜**)。

5.4.4 参考四双面板空间紧凑



如上图，板子长但是比较窄，但是触摸按键之间间距比较大，这种就不需要覆 GND 铜隔离了，芯片电源 104 电容 C1 和触摸 TKCOM 的电容 C2 都需要尽量靠近芯片，然后隔离电阻也要尽量靠近芯片，触摸走线要少过孔，所有走线都要尽量细因为都是需要从触摸按键地下穿过的，这个触摸走线是 6mil，触摸按键选择正方形这样感应面积会更大。

5.4.5 参考五模具厚



如上图，因为模具比较厚，所有需要触摸灵敏度比较高，就尽量把触摸按键加大，触摸按键之间加覆 GND 铜隔离防止串键，走线 TOP 层就不需要覆铜这样灵敏度更高，电源电容 C1 和触摸 TKCOM 电容 C2 要离芯片尽量近，电源电容到 MCU 电源线要尽量短，TOP 走线层走线尽量细，触摸走线是 8mil。

5.5 注意事项总结

1、 触摸布板是相对要求较高的，其性能除 MCU 本身外，布板是否规范也非常重要，但考虑到实际情况，有些情况下我们不得不违背规则，我们需要做的是尽量不违背布板规则，从中找到一个折中点。

2、 覆地，有些情况很难去判断其好坏，它与感应电极大小，感应距离，覆盖物材料与厚度度息息相关，不代表它就适用所有运用场合，故以上对覆铜的规范，通常是经验的参考，如采用 30%网格，就是在覆与不覆间的一个折中选择，这即可提高一定的抗干扰能力，但也降低了 感应电极的灵敏度。

注：受限于对所有运用场景缺乏充分考虑，以上为一般 layout 指导参考，若出现场合与上述不适用时，请联系我们，谢谢！