

# W800\_touch\_sensor 软硬件设计指南 v1.0

北京联盛德微电子有限责任公司 (winner micro)

地址：北京市海淀区阜成路 67 号银都大厦 18 层 1802

电话：+86-10-62161900

公司网址：www.winnermicro.com

## 文档历史

版本	完成日期	修订记录	作者	审核	批准
V1.0	2021-3-04	创建	CZX		

## 目录

1	引言 .....	2
1.1	编写目的 .....	2
1.2	预期读者 .....	2
1.3	术语定义 .....	2
2	设计原理 .....	3
3	软硬件设计指导建议 .....	6
3.1	软件相关 .....	6
3.2	硬件相关 .....	10

## 1 引言

### 1.1 编写目的

本文档主要目的是为 W800\_touch\_sensor 功能软硬件设计提供指导参考。

### 1.2 预期读者

本文档读者包括相关的电路设计人员和相关的软件系统设计人员。

### 1.3 术语定义

touch\_sensor: 触摸检测功能&触摸传感器

## 2 设计原理

W800 芯片集成 Touch sensor 触摸检测功能，电容式触摸传感器，支持检测 11 个 IO 上触摸按键的状态，并通过中断上报按键状态。因为电路结构限制，同一时间，只能检测一个 IO 的状态。

Touch sensor 功能开启时，对应 IO 的触摸电容，会采用模块内部设定的偏置电流，对其进行周期性充放电，进而在模块内部生成一个对应的输出脉冲信号。由于触摸管脚受到触碰（高电容）与未受到触碰（低电容）时的充电电压变化速率不同，所以输出脉冲信号的频率（周期 T）也不同。用寄存器设定 N 个输出脉冲信号的周期，为计数窗口。用系统时钟 40Mhz，对计数窗口进行计数，并把计数结果存到相应的寄存器。通过计数值和设定的计数阈值，判断 IO 是否被触碰。

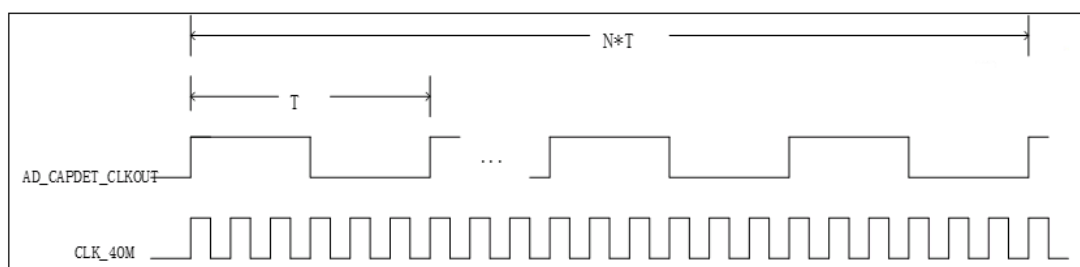


图 2.1 计数示意图

图 2.2 是触摸电容分布示意图，其中  $C_{ground}$  是触摸电路参考地和大地之间的电容， $C_{component}$  是芯片内部寄生电容， $C_{trace}$  走线与电路参考地之间的寄生电容， $C_{electrode}$  触摸电极与电路参考地之间的寄生电容， $C_{touch}$  手指与触摸电极所形成的相对于大地的电容。

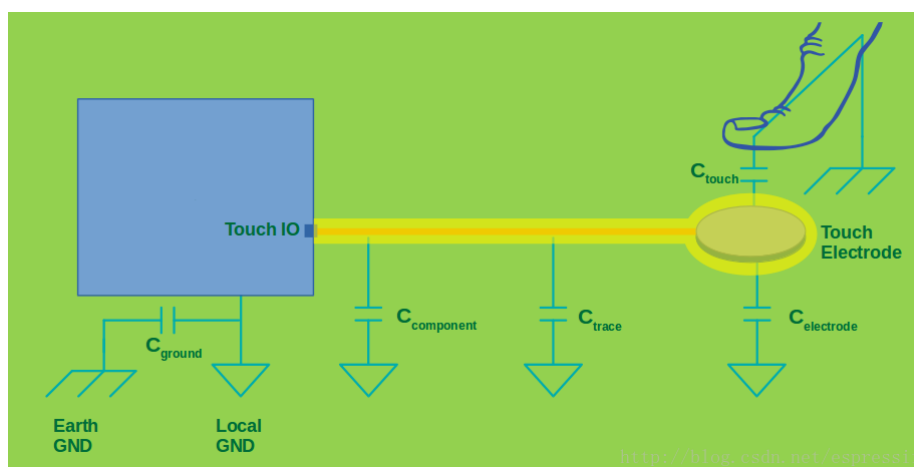


图 2.2 触摸电容分布示意图

寄生电容  $C_p$ （即未发生触摸动作时的电容）=  $C_{component} + C_{trace} + C_{electrode}$

发生触摸动作时，系统总电容的变化量  $\Delta C = C_{\text{touch}}$ ，常见  $C_{\text{touch}}$  约为 5pF~15pF  
可见，当寄生电容  $C_p$  越小， $C_{\text{touch}}$  越大时，触摸动作越容易被系统检测到，即触摸传感系统的灵敏度就越高。

Touch sensor 相关寄存器描述如表 2-1 所示。

表 2-1 Touch sensor 寄存器地址为 0x4001 2400

偏移量	名称	bit 位	属性	描述	复位值
0x00	touch_cr				
		31:26	RW	Scan_period 扫描周期，单位为 16ms； 其中 0 不可用	6'd10
		25: 20	RW	CAPDET_CNT 选择多少个 CAPDET 输出脉冲数 作为计数窗口 其中 0、1、2 不可用，计数的脉冲 数=N-2	6'd20
		19:4	RW	Touch Sensor 扫描选择； 0x0000：不扫描； 0x0001：扫描第一个 Touch Sensor； 0x0002：扫描第二个 Touch Sensor； 0x0003：顺序扫描第一个和第二个 Touch Sensor； 0x0004：扫描第三个 Touch Sensor； ... 0xFFFF：顺序扫描 16 个 Touch Sensor，从低到高；	16'd0
		3:1	RW	RSV	3'd0
		0	RW	Touch Sensor Enable 1: enable Touch Sensor Scan； 0: Disable Touch Sensor Scan；	1'b0
0x04	Touch_Sensor0	21:8	RO	Touch Senor 0 计数值	14'd0
		[7]	RO	RSV	1'b0
		6:0	RW	Touch Senor 0 阈值	7'd50
0x08	Touch_Sensor3	21:8	RO	Touch Senor 3 计数值	14'd0
		[7]	RO	RSV	1'b0

		6:0	RW	Touch Senor 3 阈值	7'd50
0x0C	Touch_Sensor4	21:8	RO	Touch Senor 4 计数值	14'd0
		[7]	RO	RSV	1'b0
		6:0	RW	Touch Senor 4 阈值	7'd50
0x10	Touch_Sensor5	21:8	RO	Touch Senor 5 计数值	14'd0
		[7]	RO	RSV	1'b0
		6:0	RW	Touch Senor 5 阈值	7'd50
0x14	Touch_Sensor6	21:8	RO	Touch Senor 6 计数值	14'd0
		[7]	RO	RSV	1'b0
		6:0	RW	Touch Senor 6 阈值	7'd50
0x18	Touch_Sensor7	21:8	RO	Touch Senor 7 计数值	14'd0
		[7]	RO	RSV	1'b0
		6:0	RW	Touch Senor 7 阈值	7'd50
0x1C	Touch_Sensor8	21:8	RO	Touch Senor 8 计数值	14'd0
		[7]	RO	RSV	1'b0
		6:0	RW	Touch Senor 8 阈值	7'd50
0x20	Touch_Sensor9	21:8	RO	Touch Senor 9 计数值	14'd0
		[7]	RO	RSV	1'b0
		6:0	RW	Touch Senor 9 阈值	7'd50
0x24	Touch_Sensor10	21:8	RO	Touch Senor 10 计数值	14'd0
		[7]	RO	RSV	1'b0
		6:0	RW	Touch Senor 10 阈值	7'd50
0x28	Touch_Sensor11	21:8	RO	Touch Senor 11 计数值	14'd0
		[7]	RO	RSV	1'b0
		6:0	RW	Touch Senor 11 阈值	7'd50
0x2C	Touch_Sensor12	21:8	RO	Touch Senor 12 计数值	14'd0
		[7]	RO	RSV	1'b0
		6:0	RW	Touch Senor 12 阈值	7'd50
0x44	int_source	31:16	RW	INT_EN 对应 bit 为 1 时表示对应 IO 被触发会产生中断; 对应 bit 为 0 时表示对应 IO 被触发不会产生中断;	16'd0
		15:0	RW	PAD_STATUS/INT_SOURCE bit 为 1 时表示对应 PAD 被触发; bit 为 0 时表示对应 PAD 未触发;	16'd0

另外，电容充电电流控制 Ibias<2:0>需要在 PMU 寄存器里设置，见表 2-2。

配置举例，如果采用 5uA 充电电流，即 Ibias<2:0>=100，可用 at 指令设置

at+&REGR=40000d1c,1

at+&REGW=40000d1c,6c9

表 2-2 PMU 寄存器基地址为 0x4000 0D00

偏移地址	寄存器名称	名称	位宽	属性	说明	复位值
0x1C	WLAN_STTS	DA_CAPDET_IBIAS	[6:4]	RW	CAPDET bias current programmable: $5\mu^{*}<2>+2.5\mu^{*}<1>+1.25\mu^{*}<0>$ 正常工作时，不能设为 000，此时无充电电流，工作不正常	3'b0

### 3 软硬件设计建议

本章节指导建议，受硬件 PCB 电路和工作环境影响较大，建议在用实际使用时，多测试验证灵敏度和稳定性，另外用户在不同产品上使用时，也可能需微调相关参数，达到最优的效果。下述分别从软件和硬件设计角度给出一些建议。

#### 3.1 软件相关

- 1) 使用前，需要正确配置相关 GPIO(高阻状态)和系统时钟状态。
- 2) 参照表 2-1 和表 2-2 配置相关寄存器，下述从 3)~9)为相关配置补充说明。
- 3) Scan\_period 扫描周期，单位为 16ms；其中 0 不可用；该时长包括一次扫描的时间和扫描后的空闲时间（正常工作时，一次扫描的时间，必定小于扫描周期，按最大计数值和 16 个 IO 计算， $0.025\mu s * 16384 * 16 = 6.6ms$ ；空闲状态时，电路保持在 touch sensor0 选中扫描的状态）。
- 4) 在空闲状态时，因为电路保持在 touch sensor0 选中扫描的状态，如果扫描选择设置里，没选择扫描 touch sensor0（对应 GPIO A7），而此时在空闲状态，电路仍然会选中 touch sensor0，W800 内部电路输出会导通对应 PADA7，如果此时如果 A7 被分配给其他功能，就可能因为 touch sensor0 的干扰，发生错误。
- 5) 使用时，需务必注意当使用 touch sensor 功能时，需使用 touch sensor0。如果不使用 touch sensor0，此时对应 touch IO（GPIO A7）也不能复用其他功能（必须置于高阻状态），否则可能烧毁 touch sensor 内部电路。
- 6) CAPDET\_CNT 项是选择多少个 CAPDET 输出脉冲数作为计数窗口，其中 0、1、2 不可用，计数的脉冲数=N-2。
- 7) 扫描选择，不同扫描方式（每次单独扫描一个 touch IO 和顺序扫描所有 touch IO）对计数的影响很小，两种方式选择哪一种都可以。
- 8) Touch Sensor Enable 项，1 使能 touch sensor 扫描，0 关闭 touch sensor 扫描；当此项从 0 变为 1 时，按照寄存器设置，对相关 IO 进行第一次扫描后得到的计数值



- (未设置扫描的 IO，其计数值为默认 0)，存入 CPU 作为无触摸计数 base 值（期望值），并保持不变。因此，touch sensor 使用过程中，如果更改了 CAPDET\_CNT 项、扫描选择、IBIAS 充电电流等影响计数值的寄存器选项，或者有温度、电压、湿度等外界环境较大变化时，必须关闭 touch sensor 再重新使能，即实现此项从 0 到 1，校准更新无触摸计数 base 值。
- 9) 寄存器每个 touch IO 的计数值，bit 位应该是 21:8；若寄存器的计数值-无触摸计数 base 值 > 寄存器设置的阈值，则硬件判断该 touch IO 被触摸，PAD\_STATUS/INT\_SOURCE 项的相应位被置 1
  - 10) DA\_CAPDET\_IBIAS 电容充电电流设置  $5\mu^{*}<2>+2.5\mu^{*}<1>+1.25\mu^{*}<0>$ 。正常工作时，不能设为 000，此时无充电电流，工作不正常。默认可先设为 100
  - 11) 根据测试结果，灵敏度=阈值/(base 值+阈值) 不应太低，故对总体容值有限制，这里列出了对应的计数值限制，供使用设计时参考。表 4-1-1 是常温下采用充电电流默认配置 100、1 个 capdet 输出脉冲作为计数窗口时，各个 touch IO 的计数值参考限制。使用时，常温同样寄存器配置下，需要各个 IO 的无触摸计数值 < 表 4-1-1 所列出的对应数值（十进制），若不满足，考虑更换 IO、更换触摸按键或者更改 PCB 走线。如果配置为多个 capdet 输出脉冲作为计数窗口，则比较数值时，表 4-1-1 的数值需要乘以脉冲数，即前面的 6) CAPDET\_CNT 项 (N-2)。

表 4-1-1 常温下各个 io 的计数值（十进制）参考限制

DA_CAPDET_IBIAS<2:0>		100		CAPDET_CNT		3	
Touch0	Touch3	Touch4	Touch5	Touch6	Touch7	Touch8	Touch9
254	247	239	212	203	199	186	181
Touch10	Touch11	Touch12					
172	161	160					

- 12) 由于 W800 内部电路特点，各个 IO 之间不存在统一的灵敏度=阈值/(base 值+阈值)，需要每个 IO 独立检测并配置合适的阈值。
- 13) 推荐各个 touch IO 的使用优先级为 touch0>touch3>touch4> touch5> touch6> touch7> touch8> touch9> touch10> touch11，touch io 和 gpio name 的对应关系，详见芯片规格书。
- 14) 寄存器 PAD\_STATUS/INT\_SOURCE 项，只能实现最简单的触摸判断，无法减弱误触、干扰等的影响，如果对按键设计有更高的要求，需要软件直接对读取的计数值进行处理判断，而不是直接读取寄存器 PAD\_STATUS/INT\_SOURCE 项。下面 15)~18)是部分方案，供软件设计参考。
- 15) IIR 滤波器功能，开启此功能后，可对脉冲计数值滤波，衰减高频噪声成分，并忽略低频信号。可调整滤波周期，W800 扫描周期最小是 16ms，建议滤波周期的范

围 16ms~25ms, 也可以更大, 但是因为延迟, 同时也需要考虑触摸按键响应时间。使用滤波器的优点, 可以减少 wifi 发包造成的周期性电源噪声, 还可以实现触摸按键消抖功能, 防止水滴经过, 皮肤误碰等。滤波器的具体参数设定, 还需要软件在 wifi 持续发包状态以及人为误触发状态下, 实际调试。

- 16) 单工滑块功能, 布置多个触摸电极, 一个触摸电极使用一个触摸传感器, 用户可以读取触碰点在滑块上的相对位置。

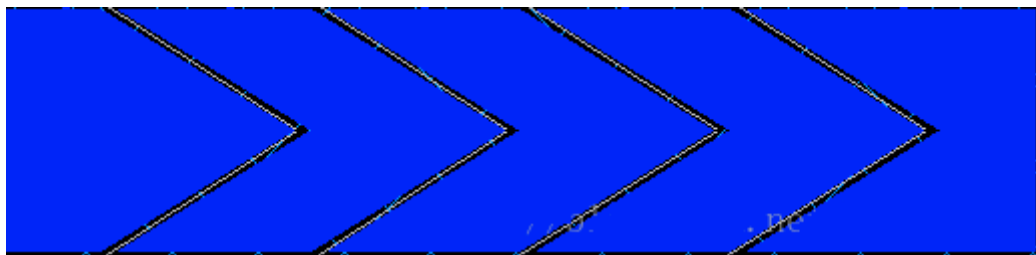


图 4.1.3 单工滑块

设  $n$  个触摸传感器组成滑块, 按位置顺序, 相应传感器的位置权重设为  $W_i = i * k$ , 其中,  $i$  是位置顺序参数 (例如从左到右第  $i$  个, 从 0 开始排序, 0、1、2... $n-1$ ),  $k$  是精度参数 (决定了定位精度, 越大位置分割相对越细)。

若第  $i$  个触摸传感器的实时计数值  $count$  变化量为  $\Delta CNT_i$ , 则相对位置为  $POS = (\Delta CNT_0 * W_0 + \Delta CNT_1 * W_1 + \dots + \Delta CNT_{n-1} * W_{n-1}) / (\Delta CNT_0 + \Delta CNT_1 + \dots + \Delta CNT_{n-1})$  计算出来的  $POS$  值, 是 0 到  $(n-1)*k$  之间的一个值, 表征触碰点相对位置。需要有传感器被触发, 才按上面公式计算, 如果  $n$  个触摸传感器都没有被触发, 则返回一个预设值, 预设值不应在 0 到  $(n-1)*k$  范围内。

软件驱动步骤:

初始化时按位置顺序创建触摸传感器实例, 为每一个触摸传感器实例添加 `push` 与 `release` 回调函数。

在 `push` 和 `release` 回调函数中, 读取此滑块中每一个触摸传感器的值, 按上面的公式计算相对位置。

- 17) 双工滑块功能, 可利用有限的触摸传感器驱动更长的滑块。

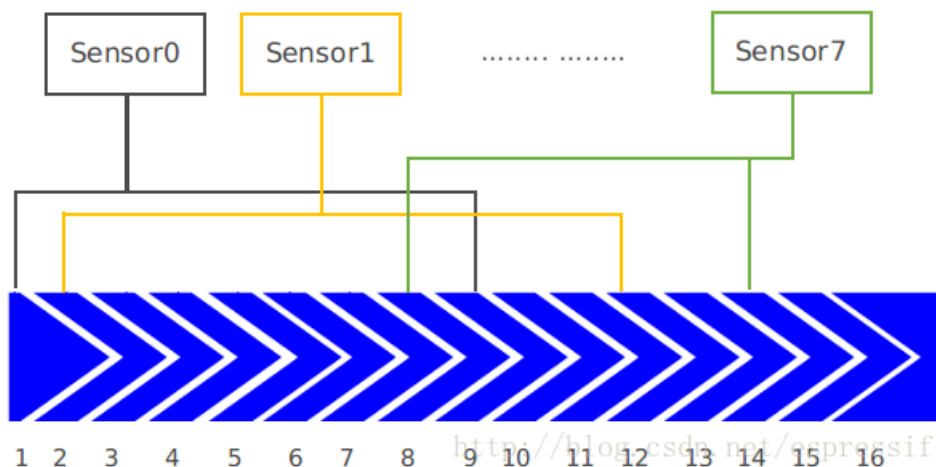


图 4.1.4 双工滑块

双工滑块和单工滑块共用一个函数，主要通过算法和 pcb 布线排版实现，用户不需要区分单工滑块和双工滑块。例如，在图 4.1.4 中的双工滑块用到了 16 个触摸电极，但只需要使用 8 个触摸传感器。左半部分 8 个触摸电极按顺序使用 8 个传感器。右半部分 8 个触摸电极以连续使用 8 个传感器，且在左半部分相邻的传感器在右半部分不能相邻。当左半部分有触摸电极被触摸时，右半部分连接同一传感器的触摸电极也会被认为有触摸，此时算法会寻找相邻触摸电极都有读数变化的区域，去除孤立的触发触摸电极。在图 4.1.4 中，当我们在 1、2 位置触摸滑块时，传感器 0 和 1 上的 1、2、9、12 号触摸电极都会被认为触发，但是只有 1 和 2 是相邻触摸电极触发，而 9 和 12 是两个孤立的触发触摸电极，所以算法会将位置定位在 1、2 区域。

- 18) 矩阵按键功能，在按键使用数量较大的应用场景下，触摸传感器不够用，可以使用矩阵按键的驱动方式。

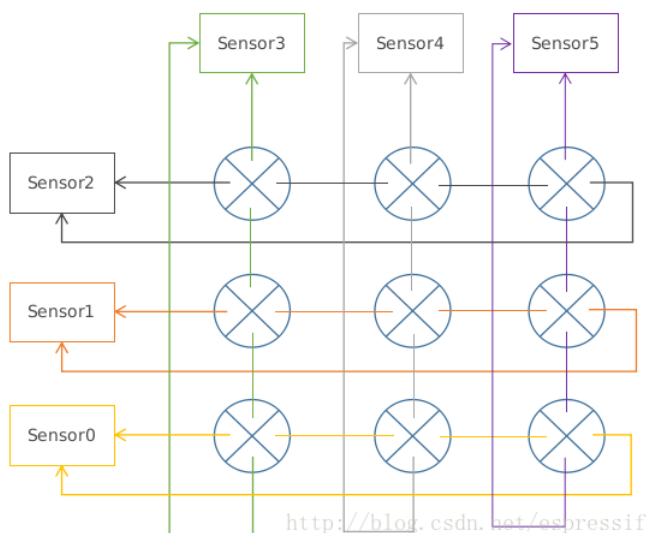


图 4.1.5 矩阵按键

例如，在图 4.1.5 中，每个触摸按键被分成 4 块，相对的两块连接成同一个触摸电极，同时矩阵中每一行（每一列）的水平块（垂直块）连接到同一个触摸传感器。当一个触摸按键上横竖两个对应传感器同时都被触发时，该触摸按键才会被认为有触摸。例如图中 sensor2 和 sensor3 同时触发时，左上角的触摸按键被判定为有触摸事件。由于矩阵按键驱动方式的限制，同时只能按一个触摸按键。当多个触摸按键同时按下时不会有触摸事件触发，当一个触摸按键正被触摸时，触摸其他触摸按键不会有事件触发。

### 3.2 硬件相关

硬件设计需尽可能降低寄生电容  $C_p$  和走线电阻，同时提高触摸电容  $C_{touch}$ 。由于 PCB 上铜线的电阻值相对很低，所以走线电阻的影响可忽略，主要考虑降低寄生电容。

- 1) 当使用 touch sensor 功能时，建议使用 touch sensor0。如果不使用 touch sensor0，此 touch sensor0 对应的 touch IO（GPIO A7）也不能复用其他功能（必须置于高阻状态），否则可能烧毁 touch sensor 内部电路。所以，使用 touchsensor 功能，硬件布线时，必须包含 touch sensor0（即 GPIO A7），或者不使用 touch sensor0 时，A7 不可复用为其他功能，必须为高阻态。
- 2) 根据测试结果，各个 touch IO 的使用优先级为 touch0>touch3>touch4> touch5> touch6> touch7> touch8> touch9> touch10> touch11> touch12>; 另外，touch io 和 gpio name 的对应关系，详细参见芯片规格书。
- 3) 根据 W800 相关测试，推荐容值范围  $C_p < 30pF + (6pF \sim 8pF) - C_{touch}(5pF \sim 15pF)$ ，即  $C_p < 21pF \sim 33pF$ ，其中寄生电容  $C_p$  包括芯片内和板级寄生电容，以及未发生触摸的触摸电容，参考图 2.2。所以，建议（PCB 寄生电容+未发生触摸的触摸电容） $< 15pF$
- 4) 增大触摸电极面积、延长走线长度和宽度以及减少离地间隙，都会使  $C_p$  变大（非线性关系）。扩大传感器和接地之间的间隙可以降低  $C_p$ ，但同时也会降低抗噪能力。不同因素对  $C_p$  影响的程度大小为：走线长度>触摸电极周围铺地>走线离地间隙>过孔数>触摸电极面积>走线宽度。  
以下 5)~15) 在硬件设计时需要针对不同的应用方式做适当的调整。
- 5) 缩短走线、走线变窄均会减少寄生电容：  
走线长度不应超过 300mm；  
走线宽度（W）不能大于 0.18mm（7mil）；  
走线夹角（R）不应小于  $90^\circ$ ；  
触摸电极与走线离地间隙（S）不应小于 1mm；

触摸电极直径 (D) 范围 8mm 到 15mm;

触摸电极和走线应被栅格地围绕;

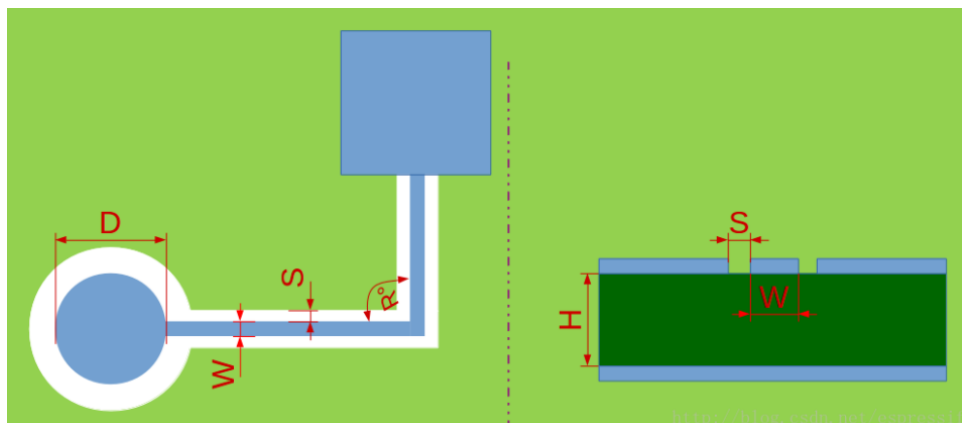
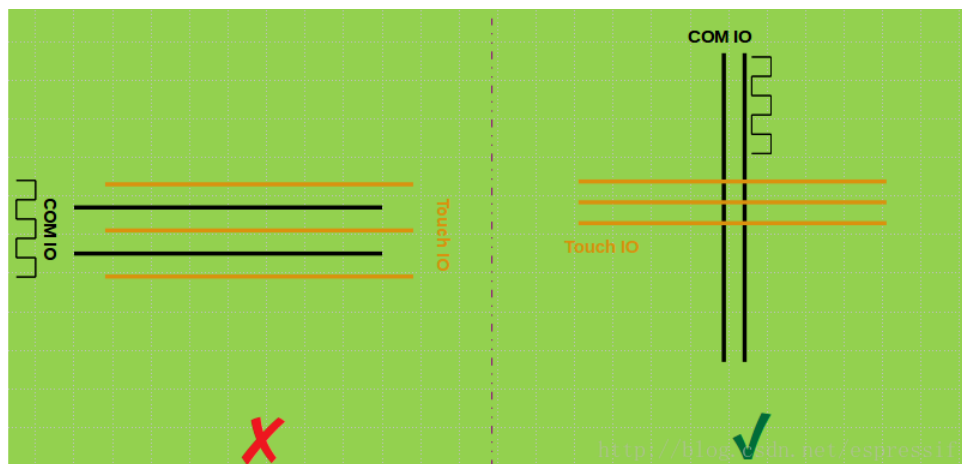


图 4.2.1 走线示例

- 6) 触摸传感器周围使用铺地屏蔽隔离，防止其他数字电路或环境对触摸电路干扰；  
在触摸电极的背面走线，防止用户误触；  
除与之相连的触摸传感器走线，触摸电极下方不能走线；  
走线不应与高频数字线并行排列（或使用接地屏蔽走线）；  
如需与通信数字线（如 SPI、I2C）相交，应确保走线与数字线其呈直角相交；  
走线应远离高功率电路，如开关电源、LCD、电机驱动、射频天线电路等；



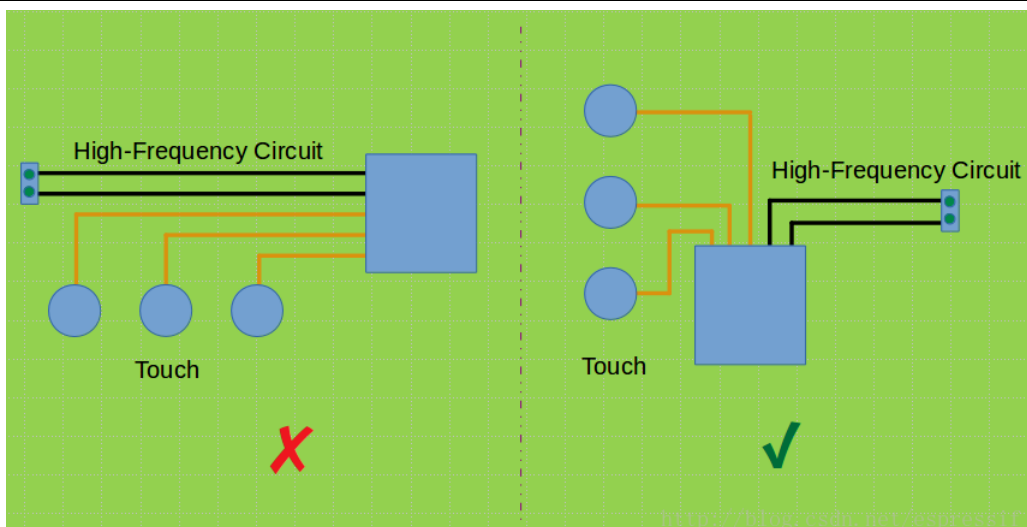


图 4.2.2 走线位置

- 7) 走线时使用过孔数量越少越好，尽量减少寄生电容；  
过孔放置在触摸电极的边缘上，减少走线长度；  
过孔直径应为 10mil；

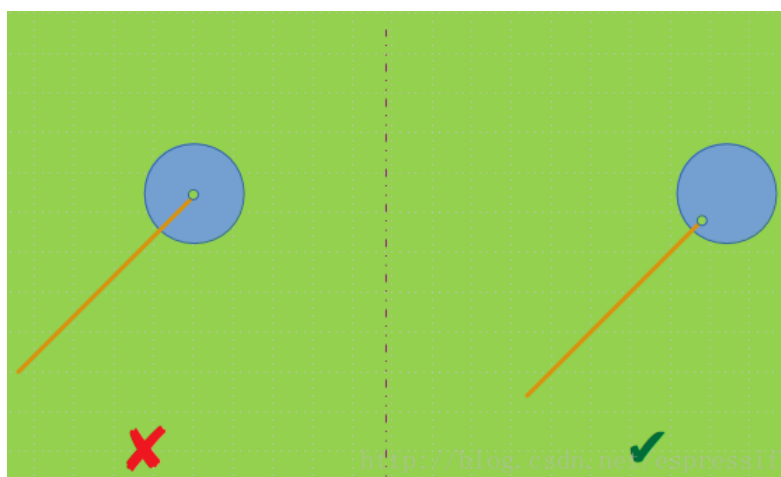


图 4.2.3 走线过孔

- 8) LED 应用相关：

LED 是触摸传感器设计中经常出现的电路，做触摸背光使用。如果触摸电极是 PCB 板上的圆形铜箔，背光 LED 应该放在触摸电极的反面。

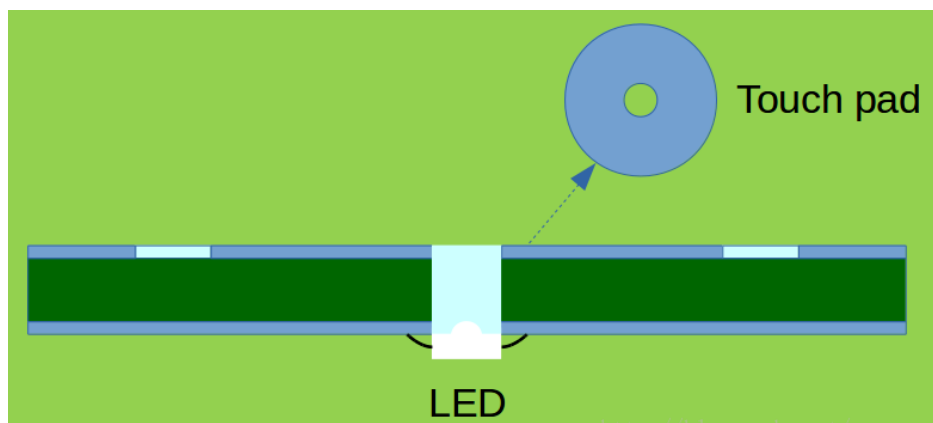


图 4.2.4 触摸背光示例

为防止 LED 的驱动电压耦合到触摸传感器的输入，应将触摸传感器走线和 LED 驱动的走线位置分开，间距最小值应为 4mm，也可以使用网格地隔离。

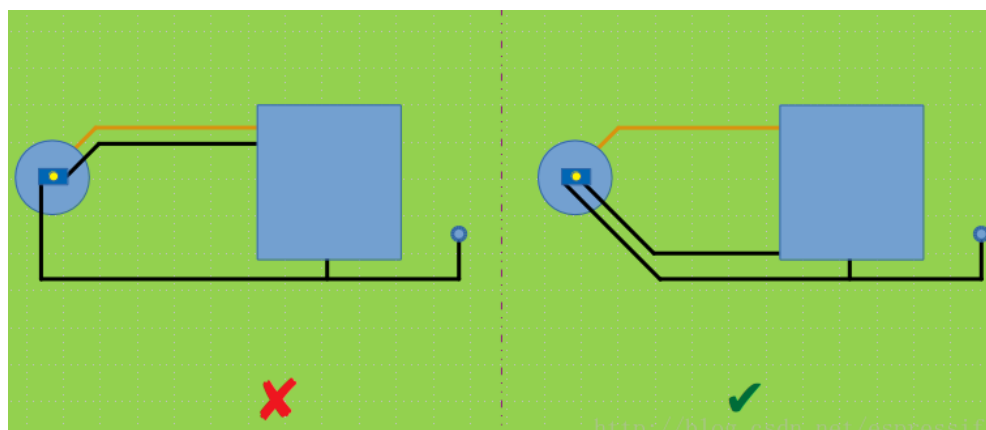


图 4.2.5 触摸背光 LED 走线位置

另一种降低耦合串扰的方法是使用滤波器电容将 LED 驱动电压的上升和下降沿放缓。所添加的电容值取决于 LED 的驱动电流要求，典型应用中使用  $0.1\mu\text{F}$ 。

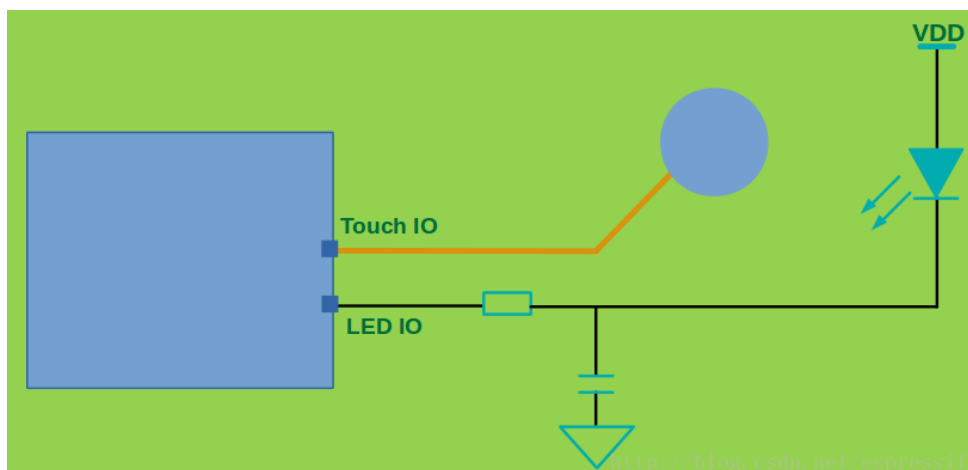


图 4.2.6 LED 走线滤波电容



如果将 LED 放置在触摸传感器附近（在 4mm 范围内），而且 LED 的某一端频繁变为非低阻抗状态（例如通过下拉或上拉来打开 LED 或通过悬空来关闭 LED），那么受 LED 影响，在 LED 分别处于打开和关闭两种不同状态时，触摸传感器电容的容值也不同。这样，当 LED 的状态发生变化时，触摸传感器的电容也会随之发生变化，这种电容的变化可能会导致误触发，应使用旁路电容减少这种影响。旁路电容应并联在 LED 的旁边，旁路电容的典型值 1nf。

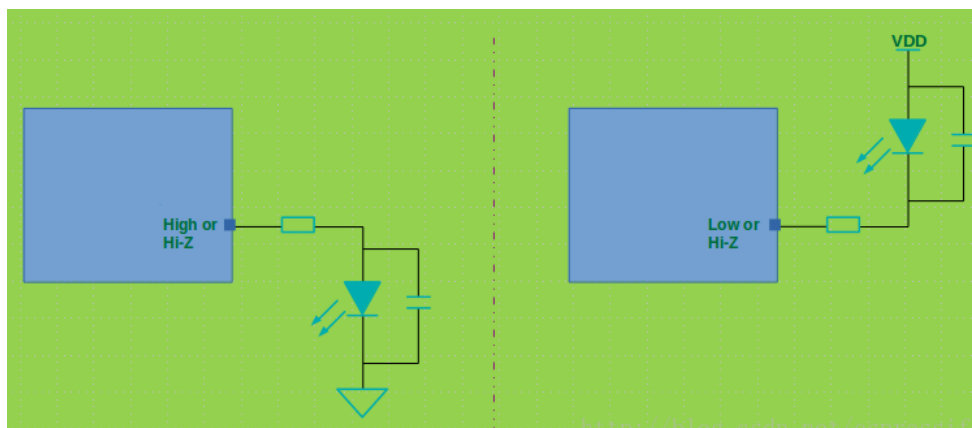


图 4.2.7 LED 旁路电容

- 9) 铺地使用栅格地可以增强抗干扰能力并保证高灵敏度的折中的一种选择  
 触摸电极周围使用栅格地，触摸电极反面如无干扰源不应铺地  
 铺地应距离触摸传感器网络至少 1mm  
 栅格地的规格应为：19%（5mil 线宽，50mil 间距）  
 触摸电极及触摸走线 10mm 范围不应有实心地

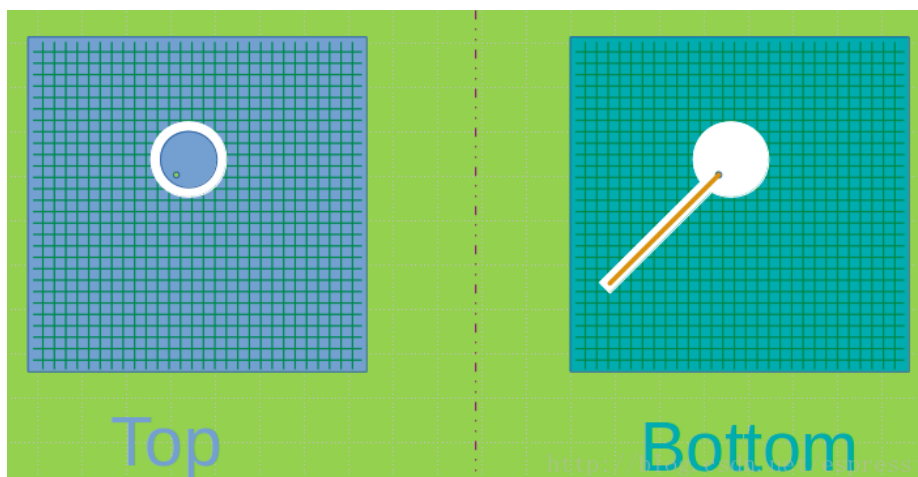


图 4.2.8 铺地示意图

- 10) 由于芯片内部电路限制，不能在触摸传感器走线上串联电阻滤波。
- 11) 连接器的选型和布局：  
 连接器焊盘封装面积不应过大，并减少其周围的铺地；



应选用连接牢固的连接器，不应出现接触不良的现象；

触摸传感器走线经过大面积焊盘时会产生寄生电容，应避免走线穿过焊盘；

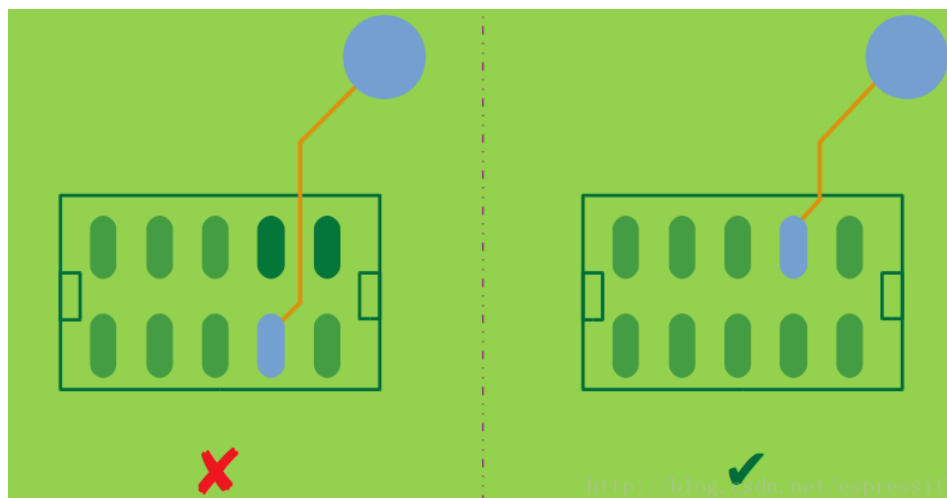


图 4.2.9 焊盘走线

#### 12) 按键设计：

按键的最佳形状是圆形，也可以使用圆角的矩形焊盘，应避免尖角（小于  $90^\circ$ ）；  
按键直径范围应介于 8mm 到 15mm 之间，典型值为 12mm。如果覆盖层较厚可以  
选用较大直径的触摸电极；

触摸电极距离栅格地网络应介于 1mm 与 2mm 之间，且应等于覆盖层的厚度。如  
果覆盖层厚度超过 2mm，触摸电极离地间隙应为 2mm；

两个相邻按键应间距 5mm 以上，避免手指触摸时互相影响；

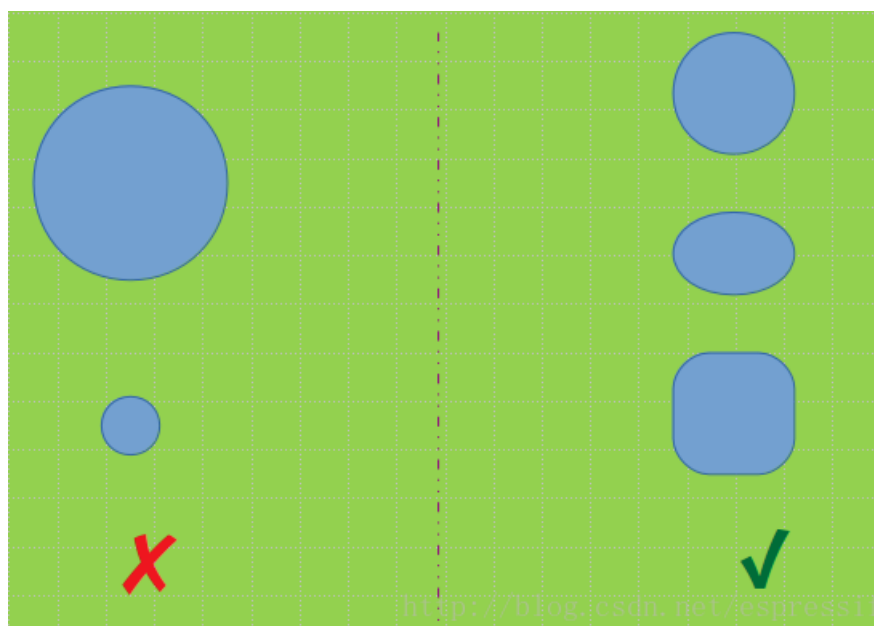
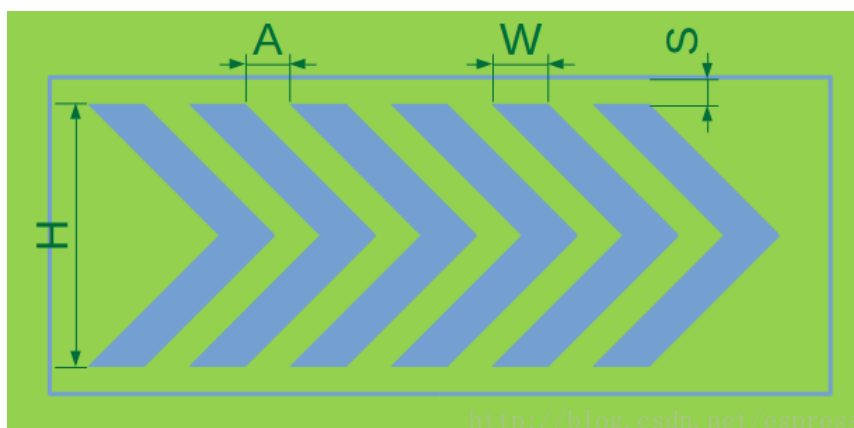
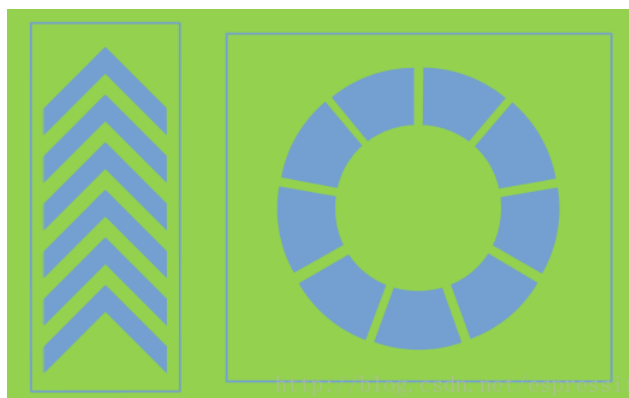


图 4.2.10 按键 PCB 示意图

#### 13) 滑条（滚轮）设计：

滑条具体排列形状，取决于产品的使用要求，人的手值直径大约 9mm，所以滑条的宽度（W）建议为 8mm，滑条间距（A）建议为 0.5mm（相邻两个触摸电极之间的距离不应过大，否则无法计算手指准确位置）。



参数	最小值 (mm)	最大值 (mm)	建议值 (mm)
宽度 (W)	4	-	8
高度 (H)	8	15	12
间距 (A)	0.5	2	0.5
距离栅格地 (S)	1	2	覆盖层厚度

图 4.2.11 建议滑条设计尺寸

#### 14) 矩形键盘设计:

矩阵键盘中单个按键被划分成两部分，行块连接到行线，列块连接到列线；  
按键直径介于 8mm 到 15mm 之间，典型值为 12mm；  
行块和列块的间隔 0.5mm；

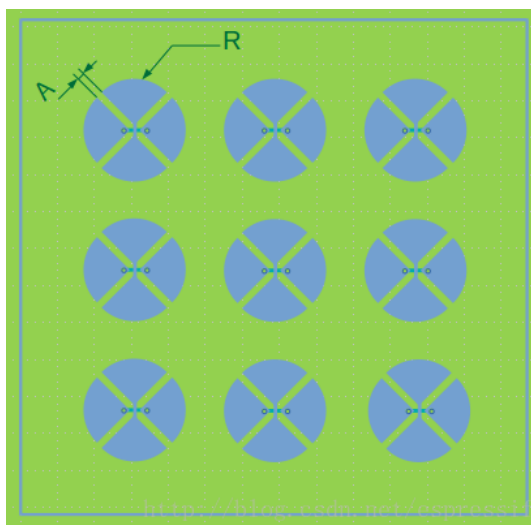


图 4.2.12 矩形键盘示意图

15) 模块参考:

触摸传感器走线不应经过芯片和模块下方;

触摸传感器走线不应靠近射频天线电路;

模块贴装的焊盘设计不应过大, 不应延伸到模块底层的铺地区, 否则触摸通道会紧邻 GND, 造成过大的寄生电容, 降低灵敏度;

模块上的触摸通道的走线是在底层, 触摸传感器的走线和引脚(红色)位置区域内, 其对应的贴装区不应铺地, 蓝色区域表示的是可铺地的贴装区域;

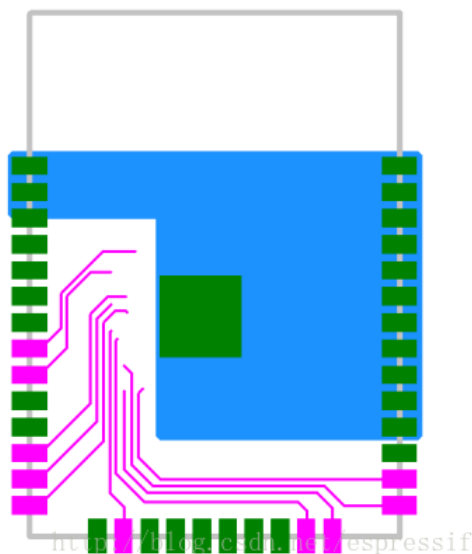


图 4.2.13 模块贴装铺地

由于不同模块 IO 定义差别, 图 4.2.13 供参考, 需要硬件根据具体的应用和模块的管脚定义, 针对 Touch Sensor 设计做适当调整。