

文档编号: AN160

上海东软载波微电子有限公司

# 应用笔记

---

## ES7P2131/2124 触控 SDK 使用说明

## 修订历史

版本	修订日期	修改概要
V1.0	2024-05-31	初版，本说明仅针对 ES7P2131 V1.4 版本触控 SDK

地 址：中国上海市徐汇区古美路 1515 号凤凰园 12 号楼 3 楼

E-mail: support@essemi.com

电 话：+86-21-60910333

传 真：+86-21-60914991

网 址：<http://www.essemi.com/>

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不承担或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系。

## 目 录

### 内容目录

目 录.....	<b>3</b>
<b>第 1 章 开发环境.....</b>	<b>5</b>
1.1 版本要求.....	5
1.2 软件安装.....	5
1.3 工程配置.....	5
1.4 编译运行.....	7
<b>第 2 章 系统固件.....</b>	<b>8</b>
2.1 概述.....	8
2.2 流程图.....	8
2.3 工程结构.....	9
2.3.1 app 文件夹.....	9
2.3.2 customer 文件夹.....	9
2.3.3 interrupt 文件夹.....	9
2.3.4 system 文件夹.....	9
2.3.5 timer 文件夹.....	9
2.3.6 tk_debug 文件夹.....	9
2.3.7 tk_lib 文件夹.....	10
2.4 参数说明.....	10
2.4.1 按键通道设定.....	10
2.4.2 触控调试设定.....	11
2.5 变量说明.....	11
2.5.1 按键状态: tk_state;.....	11
2.5.2 按键计时: tk_arr_t[],tk_to_cnt;.....	11
2.5.3 10ms 计时时基: timer_10ms;.....	11
2.6 函数说明.....	12
2.6.1 init.c 文件.....	12
2.6.2 timer.c 文件.....	13
2.6.3 tk_lib.c 文件.....	13
2.6.4 tk_ES7P2131_V1.0.hrlib 文件.....	14
2.6.5 customer.c 文件.....	14
<b>第 3 章 工程应用.....</b>	<b>15</b>
3.1 概述.....	15
3.2 初始化部分.....	15
3.2.1 void ram_clear(void).....	15
3.2.2 void int_vectors_init(void).....	15
3.2.3 void tk_pin(void).....	15
3.2.4 void timer_t10_init(void).....	15
3.3 主循环部分.....	15
3.3.1 void timer_check(void).....	15
3.3.2 void tk_service(void).....	15
3.4 中断部分.....	16

3.4.1	void isr_t10_handler(void) interrupt_low 0x0010.....	16
3.4.2	void isr_tk_handler(void) interrupt_low 0x0020.....	16
<b>第4章</b>	<b>常见问题.....</b>	<b>17</b>
4.1	按键误触.....	17
4.1.1	软件程序.....	17
4.1.2	电源供电.....	17
4.1.3	整机系统.....	17
4.1.4	结构安装.....	17
4.1.5	环境影响.....	17
4.2	结果异常.....	17
4.2.1	转换结果偏小.....	17
4.2.2	转换结果偏大.....	17
4.3	EMC 测试.....	18
4.3.1	CS 测试.....	18
4.3.2	EFT 测试.....	18
4.3.3	EMI 测试.....	18
<b>附录1</b>	<b>阈值调试.....</b>	<b>19</b>
附录 1.1	STEP1.....	19
附录 1.2	STEP2.....	20
附录 1.3	STEP3.....	21

## 第1章 开发环境

### 1.1 版本要求

要求 iDesignerV4.2.3.186 和 HRCCV1.2.0.118 编译器及以上版本。较低版本只能针对 HR7P201 芯片进行开发。

### 1.2 软件安装

从官网下载开发软件及芯片开发应用例程包。

集成开发环境：<https://www.essemi.com/index/product/detail?id=802>

安装预安装环境、C 编译器工具链 HRCC、集成开发环境软件 iDesigner。

参考设计例程：<https://www.essemi.com/index/product/detail?id=904>

文件标题中有 ESTKLIB 的为触控库，此版说明仅针对 ES7P2131 V1.4 版本触控 SDK。

### 1.3 工程配置

打开工程文件



打开工程后再双击打开 main.c 文件，iDesigner 页面概览如下：

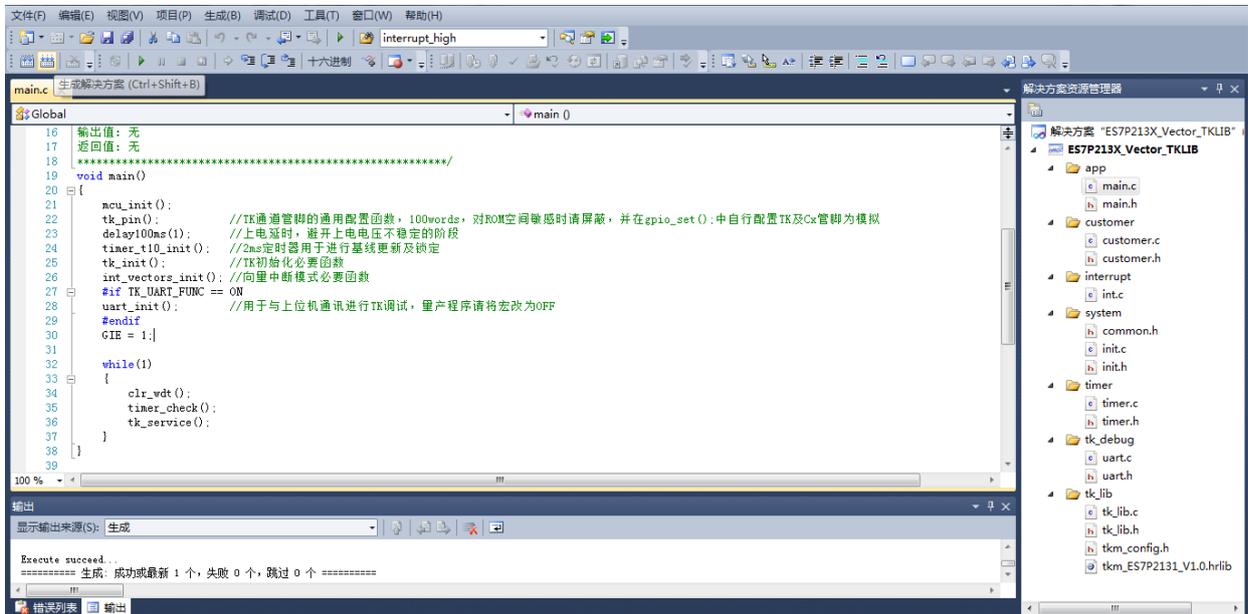


图 1-1 IDE 开发环境界面

然后确认芯片型号与配置字。通过菜单中的项目->属性->设备可以查看和选择芯片。



图 1-2 芯片选择界面

同时查看配置字设定，在工具->启动项目配置字也可查看相同内容。

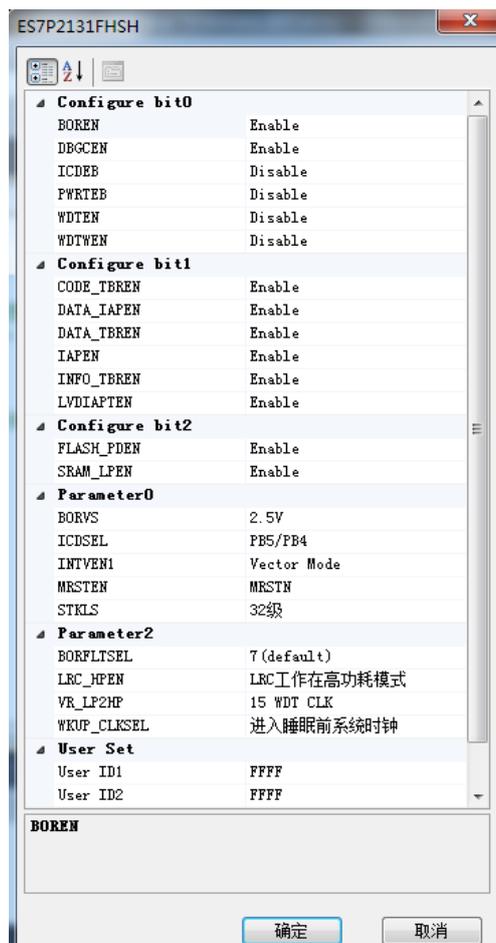


图 1-3 配置字界面

## 1.4 编译运行

在工程名上右键重新生成，编译通过时如下所示：



图 1-4 编译工程

在项目配置字中配置 ICD 接口功能，如下所示：

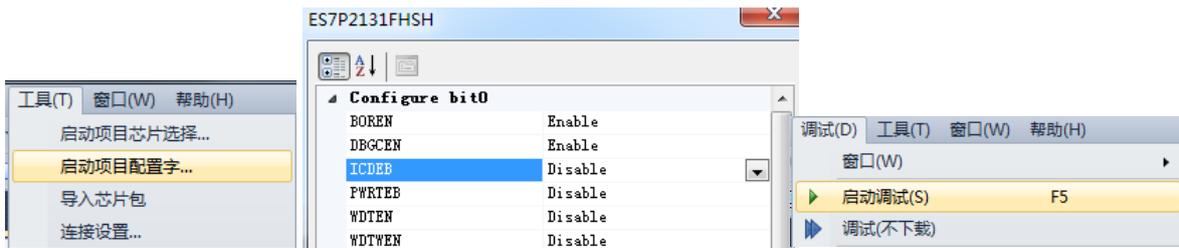


图 1-5 程序调试配置

配置为 Disable 时点启动调试按钮，为下载程序，下载完成后如下所示：



图 1-6 程序烧录

配置为 Enable 时点启动调试按钮，为调试程序，注意需在配置字中正确配置调试口，否则会提示需要检查端口设置，如下所示：

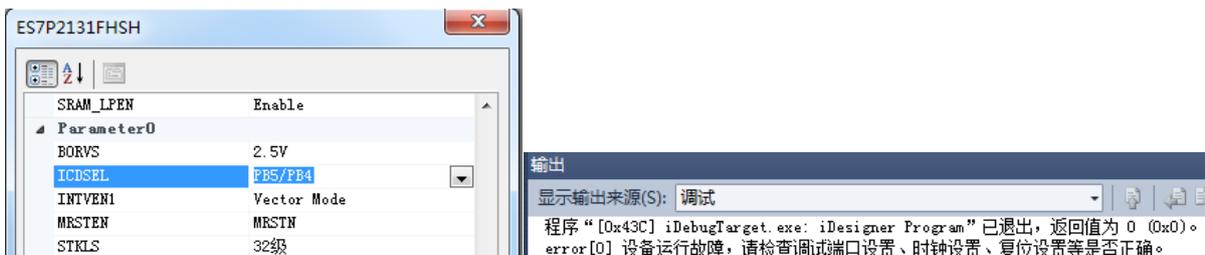


图 1-7 程序调试

## 第2章 系统固件

### 2.1 概述

用户可根据芯片的数据手册自主开发触控按键底层驱动软件，也可以选用本公司提供的 TK 驱动库函数，此固件约占 2.4Kwords ROM 空间，其封装文件为：tk\_芯片型号\_版本号 .hrlib

### 2.2 流程图

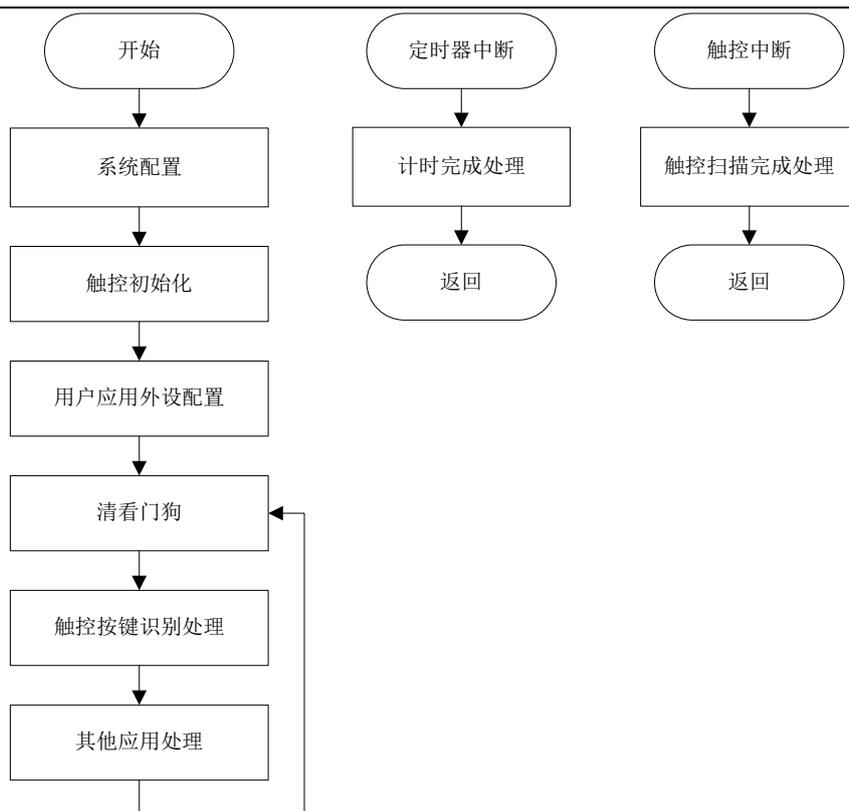


图 2-1 中断及主流程图

## 2.3 工程结构

---

### 2.3.1 app 文件夹

main.c

函数处理主循环。

main.h

链接其他各个声明头文件。

### 2.3.2 customer 文件夹

customer.c

预留应用层按键初始化及按键识别处理函数。

customer.h

对应的预留函数声明。

### 2.3.3 interrupt 文件夹

int.c

中断处理函数入口，调用触控中断，定时器中断等。

### 2.3.4 system 文件夹

common.h

工程支持文件，工程中用到的宏定义，及常用指令宏定义。

init.c

系统时钟、变量、看门狗等配置。

init.h

对应的宏定义及函数声明。

### 2.3.5 timer 文件夹

timer.c

定时器初始化函数，用户可根据需求修改，同时需要修改中断服务函数及定时时基。

timer.h

对应的函数声明。

### 2.3.6 tk\_debug 文件夹

uart.c

与上位机连接数据传输串口相关函数，开发调试时使用。

uart.h

对应的宏定义及函数声明。

### 2.3.7 tk\_lib 文件夹

tk\_lib.c

触控外部函数源文件，包含变量定义，触控初始化、中断处理、抖频处理函数等。

tk\_lib.h

对应的宏定义、变量及函数声明等。

tk\_config.h

存放触控各项参数的配置，以及条件编译开关等。

tk\_xxxx\_Vx.x.hrlib

触控内部数据处理算法函数库。

## 2.4 参数说明

所有常用的触控配置参数都已集中在 tk\_config.h 文件中，请尽量在此文件中修改，其它文件请慎动。

### 2.4.1 按键通道设定

参数名	默认值	设定范围	作用
TK_NUM	3	1~20 (ES7P2131FHSB)	按键个数设定

定义了需要使用的按键个数，包含所有用到的触控通道。

参数名	默认值	设定范围	作用
TK_CHANNEL_SEL	0x0000000E	/	通道选择设定

对应通道选择寄存器 TKCHSL，确认与触控按键相连的通道，将通道对应的位置位。

参数名	默认值	设定范围	作用
TK_CHANNEL_X	/	TK0~TK19 (ES7P2131FHSB)	软件通道与引脚通道映射设定

只有前 TK\_NUM 个定义有效，为了不同的硬件设计也能够使用同一库做处理，将硬件选择的按键通道与数据处理通道进行了映射。tk\_state 的状态位与之一一对应，如其 bit0 即为 TK\_CHANNEL\_0 状态。定义先后顺序仅影响数据处理的先后，可重复定义通道。

参数名	默认值	设定范围	作用
TK_THD_CHANNEL_X	100	100~500	按键阈值设定

用于定义各通道的阈值，影响按键灵敏度，阈值设定原则为最终产品手触摸变化量（差值）的一半。阈值调试的详细操作请参考上位机软件操作手册相关章节。

### 2.4.2 触控调试设定

参数名	默认值	设定范围	作用
TK_UART_FUNC	OFF	ON/OFF	调试控制开关

与上位机连接用的调试串口，此宏定义用于设定此调试串口是否使能。

参数名	默认值	设定范围	作用
TK_UART_PORT (位于文件 <code>uart.h</code> 中)	PB45	PB45/PA01	数据输出端口选择

当芯片有多组串口时，可选择需要的串口组。

参数名	默认值	设定范围	作用
TK_UART_BAUD (位于文件 <code>uart.h</code> 中)	115200	115200/57600/38400/19200...9600	串口波特率设定

用于设定串口波特率，当通信迟钝时可尝试降低波特率。

## 2.5 变量说明

### 2.5.1 按键状态: `tk_state`;

每一位对应一个按键状态。1 代表有键按下，0 代表松开。

### 2.5.2 按键计时: `tk_arr_t[].tk_to_cnt`;

打开按键超时功能后，用于对按键按下时间进行计时，每 10ms 累加一次。按下时间不能超过 `TK_PRESS_TIMEOUT` 时长，如果超过，则强制更新基线。

### 2.5.3 10ms 计时时基: `timer_10ms`;

当定时器修改时，请将此变量的累加处理转移到新的定时中断中，并修改对应的计时判断条件 (`timer_check` 中)。

## 2.6 函数说明

### 2.6.1 init.c 文件

函数名	<b>void</b> delay100ms(uchar time)
函数说明	延时通用函数
输入参数	计数时间，单位 100ms
返回值	无
调用方法	有延时需求（比如上电延时）时调用

函数名	<b>void</b> ram_clear(void)
函数说明	RAM 区清零子程序，执行后对应存储在 RAM 中的变量也会清零。
输入参数	无
返回值	无
调用方法	在变量初始化前调用，注意调用后将初值非零的变量赋初值

函数名	<b>void</b> wdt_set(uchar time)
函数说明	看门狗配置函数
输入参数	看门狗设定时间，已有对应的宏定义
返回值	无
调用方法	在系统做初始化配置时调用

函数名	<b>void</b> sysclk_set(void)
函数说明	系统时钟配置函数
输入参数	无
返回值	无
调用方法	在系统做初始化配置时调用

函数名	<b>void</b> int_vectors_init(void)
函数说明	向量中断模式配置函数
输入参数	无
返回值	无
调用方法	在系统做初始化配置时调用

函数名	<b>void</b> gpio_set(void)
函数说明	GPIO 状态配置函数，用户根据应用需求做修改
输入参数	无
返回值	无
调用方法	在系统做初始化配置时调用

函数名	<b>void</b> mcu_init(void)
函数说明	系统初始化封装函数
输入参数	无

返回值	无
调用方法	在系统做初始化配置时调用

### 2.6.2 timer.c 文件

函数名	<code>void timer_t10_init(void)</code>
函数说明	定时器初始化函数，用户根据应用需求进行修改，同时注意修改 10ms 计时时基。
输入参数	无
返回值	无
调用方法	在系统做初始化配置时调用

### 2.6.3 tk\_lib.c 文件

函数名	<code>void tk_init(void)</code>
函数说明	触控模块状态参数等初始化函数
输入参数	无
返回值	无
调用方法	在系统做初始化配置时调用

函数名	<code>void tk_reg_config(void)</code>
函数说明	触控模块寄存器初始化配置函数
输入参数	无
返回值	无
调用方法	在触控模块运行时调用

函数名	<code>void tk_jitter(void)</code>
函数说明	触控抖频函数，可选功能，可选软件或硬件抖频，改善 CS 测试效果
输入参数	无
返回值	无
调用方法	在触控模块运行时调用

函数名	<code>void tk_convert_complete(void)</code>
函数说明	触控采样完成处理函数，扫描异常时将重新启动
输入参数	无
返回值	无
调用方法	在触控中断服务函数中调用

函数名	<code>void timer_check(void)</code>
函数说明	触控计时处理函数，用于基线的更新及锁定计时以及按键的超时处理等。
输入参数	无
返回值	无
调用方法	在定时中断或主循环中调用

#### 2.6.4 tk\_ES7P2131\_V1.0.hrlib 文件

函数名	<code>void tk_pin(void)</code>
函数说明	触控通道对应引脚配置通用函数，占用 100words。
输入参数	无
返回值	无
调用方法	在系统做初始化配置时调用，对 ROM 空间敏感时将功能移植到 <code>gpio_set()</code> 中

函数名	<code>void tk_force_average(unsigned char ch)</code>
函数说明	强制更新基线函数
输入参数	触摸映射通道编号
返回值	无
调用方法	根据应用需求（如异常状态恢复）进行调用

函数名	<code>void tk_press_to(void)</code>
函数说明	按键计时及超时处理函数
输入参数	无
返回值	无
调用方法	在定时中断或计时函数中调用

函数名	<code>void tk_base_up_lock_cnt(void)</code>
函数说明	基线更新或锁定计时处理函数
输入参数	无
返回值	无
调用方法	在定时中断或计时函数中调用

函数名	<code>void tk_service(void)</code>
函数说明	触控数据处理及按键识别处理封装函数
输入参数	无
返回值	无
调用方法	在主循环中调用

#### 2.6.5 customer.c 文件

函数名	<code>void customer_key_handler(void)</code>
函数说明	用户按键处理预留函数，用户根据按键状态变量 <code>tk_state</code> 处理应用层事务
输入参数	无
返回值	无
调用方法	在主循环中调用

函数名	<code>void customer_key_init(void)</code>
函数说明	用户按键处理初始化预留函数，初始化按键控制的相关外设，如蜂鸣器等
输入参数	无
返回值	无
调用方法	在系统做初始化配置时调用

## 第3章 工程应用

### 3.1 概述

用户根据应用需要可对此固件功能做修改删减。为了触控功能的正常使用，用户在工程中应当注意包含 `tk_lib` 文件夹中的所有文件，并在正确的位置调用以下触控功能必要函数：`tk_init()`（初始化）、`timer_check()`（主循环）、`tk_service()`（主循环）、`tk_convert_complete()`（中断）。下面列出了其他功能函数在使用时的一些注意事项。

### 3.2 初始化部分

#### 3.2.1 `void ram_clear(void)`

调用后将所有 RAM 区的变量清零，应用时若有变量初值非零，注意在此函数之后再赋值。

#### 3.2.2 `void int_vectors_init(void)`

用户可根据应用需求，自行修改中断的优先级。通过 `INTP` 设成高优先级的，注意将中断入口函数地址前写上 `interrupt_high`

#### 3.2.3 `void tk_pin(void)`

触摸引脚配置通用函数，会占用约 100words 的 ROM 空间。可将此函数屏蔽，自行在 `gpio_set()` 中配置 TK 及 Cx 管脚为模拟。

#### 3.2.4 `void timer_t10_init(void)`

用户可根据应用需求修改定时器配置。触控运行需要一个 10ms 的定时时基，修改定时器配置后需要在 `timer_check()` 中计算修正 10ms 计时的判断条件。

### 3.3 主循环部分

#### 3.3.1 `void timer_check(void)`

修改定时器后，注意此函数中 10ms 时基判断条件的修改。用户可根据应用需求，自行选择是否使用其调用的按键超时强制更新功能。用户还可以直接使用强制更新函数 `tk_force_average(unsignedchar ch)` 来做异常恢复处理（使用前请做充分地咨询与评估）。

#### 3.3.2 `void tk_service(void)`

请确保 `timer_check()` 与 `tk_service()` 处在主循环中，并且主循环中其他应用的处理时间少于一次按键操作的时间。

## 3.4 中断部分

---

### 3.4.1 `void isr_t10_handler(void) interrupt_low 0x0010`

用户可根据应用需求修改定时器中断函数。触控运行需要一个 10ms 的定时时基，修改时请保持变量 `timer_10ms` 在定时中断函数中正常累加计数。

### 3.4.2 `void isr_tk_handler(void) interrupt_low 0x0020`

当触控功能异常时，可在中断函数中查看是否有相关异常标志置起以进行分析。使用向量中断模式时，IIC 中断与 TK 中断使用同一个中断地址，相关处理在此函数中添加。

## 第4章 常见问题

### 4.1 按键误触

---

#### 4.1.1 软件程序

TK 或 Cx 引脚被配置上拉等异常程序操作。

硬件堆栈溢出，中断嵌套或函数调用层数过多。

产品 PCB 寄生电容较大，软件进行了不合理的跳频处理。

#### 4.1.2 电源供电

上电电源不稳定，导致基线初值异常。

电源纹波较大，影响了充电电压的稳定性。

#### 4.1.3 整机系统

射频影响，可以尝试将触控与射频分开操作或改变工作频率。

电机影响，可尝试根据数据做屏蔽或数据滤波操作。

#### 4.1.4 结构安装

面板材料厚度，材料有导电性导致触摸时有整体影响。

面板贴合度不够，有空隙一旦形变采样数据易受影响。

#### 4.1.5 环境影响

高低温变化，可能在表面形成水汽或使面板产生形变等，进而影响采样数据。

表面溅水影响，薄水加快基线更新适应变化，较多时考虑采用多按键触发屏蔽。

覆盖物影响，如智能锁刷卡时，考虑采用整体门限做屏蔽。

### 4.2 结果异常

---

#### 4.2.1 转换结果偏小

所用 Cx 电容偏大，易导致按键不灵敏。

TK 通道上串接的电阻过大或虚焊，会导致按键操作无反应。

#### 4.2.2 转换结果偏大

Cx 管脚被设为上拉，同时易引起按键误触发。

Cx 电容偏小，同时易发生因反应过于灵敏而导致的误触发问题。

TK 管脚被设为上拉，较其他通道采样值偏大，同时会发生误触无反应等情况。

## 4.3 EMC 测试

---

### 4.3.1 CS 测试

发生误触，产生问题的原因是干扰频段与产品的固有频率之间产生了同频干扰，所以比较有效的方式是尝试改变充放电频率，看能否有工作频率能够避开干扰频段，然后做跳频处理。发生迟钝，一般是因为有了较大干扰，而软件中识别到数据有较大波动时会屏蔽按键，可以尝试放宽改大 TK\_JITTER\_THD\_LV2。

### 4.3.2 EFT 测试

主要依赖于硬件，去耦电容应尽量靠近芯片的电源和地管脚。连到触控芯片上的电源线不要再引出去驱动其它负载。

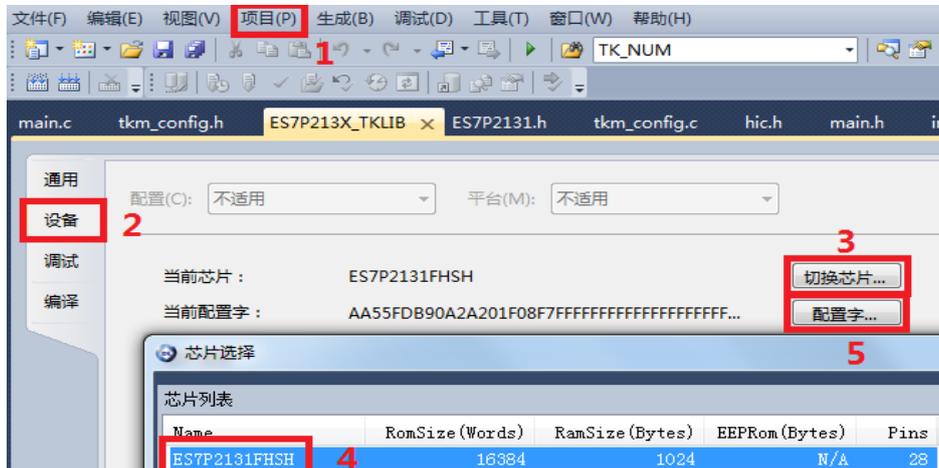
### 4.3.3 EMI 测试

注意将无用引脚串电阻接地并设为输出低电平。适当降低 TK 扫描频率可减少对外的电磁干扰。在有硬件抖频的芯片上打开硬件抖频功能，可以大幅降低尖峰频率出现。合理铺地也可有效地减少干扰。

## 附录1 阈值调试

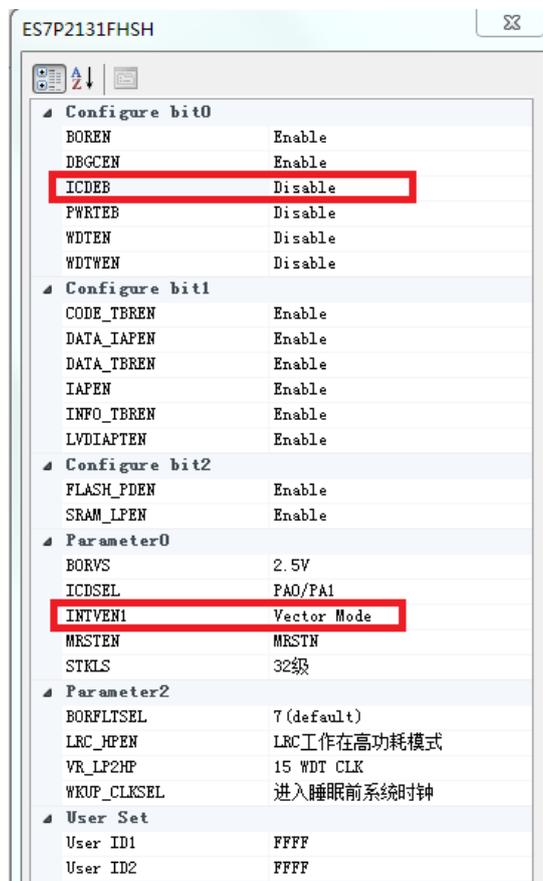
### 附录1.1 STEP1

我司 8 位机有多种产品，在调试前应根据使用的芯片确认芯片型号和配置字。



附录 1-1 芯片型号确认

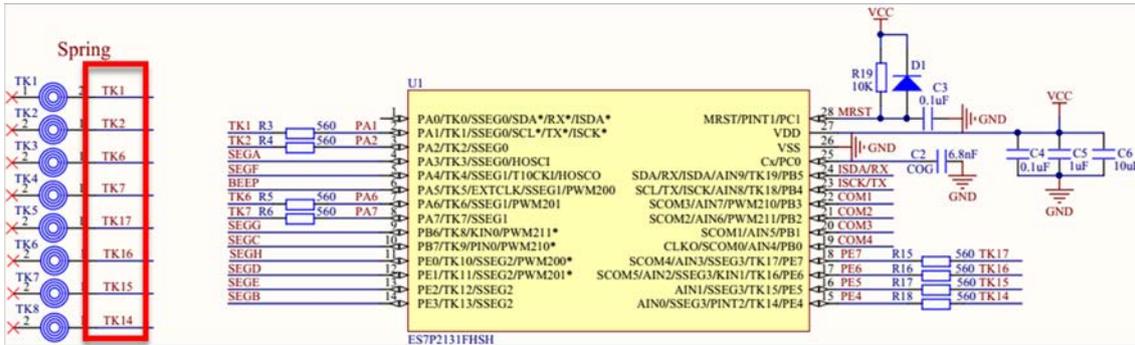
配置字一般修改以下两项即可



附录 1-2 配置字确认

## 附录1.2 STEP2

例如应用设计的原理图如下，是一个 8 按键的设计。



附录 1-3 原理图对照

根据硬件设计修改按键配置

```
tkm_config.h x tk_lib.h main.c
13
14 //*****
15 #define TK_NUM 8 // 触控按键个数: 20 按键个数
16
17 #define TK_CHANNEL_SEL 0x0003c0c6 // 根据实际应用配置 通道寄存器设置
18
19 #define TK_CHANNEL_0 TK1 // 以下只有前TK_NUM个定义有效
20 #define TK_CHANNEL_1 TK2
21 #define TK_CHANNEL_2 TK6
22 #define TK_CHANNEL_3 TK7
23 #define TK_CHANNEL_4 TK17 通道映射设置
24 #define TK_CHANNEL_5 TK16
25 #define TK_CHANNEL_6 TK15
26 #define TK_CHANNEL_7 TK14
```

附录 1-4 按键配置修改

设置一个比较小的门限比如 50，否则按键可能无法触发

```
tkm_config.h x tk_lib.h main.c
#define TK_THD_CHANNEL_0 50
#define TK_THD_CHANNEL_1 50
#define TK_THD_CHANNEL_2 50
#define TK_THD_CHANNEL_3 50
#define TK_THD_CHANNEL_4 50
#define TK_THD_CHANNEL_5 50
#define TK_THD_CHANNEL_6 50
#define TK_THD_CHANNEL_7 50
```

附录 1-5 按键阈值修改

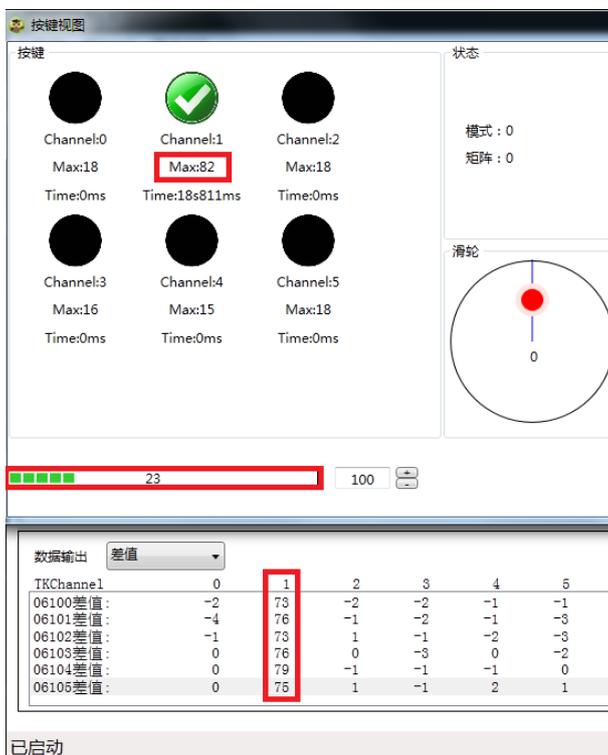
根据电路原理设计选择数据输出口，打开串口调试功能

```
tkm_config.h x tk_lib.h main.c
#define TK_UART_FUNC ON
```

附录 1-6 调试串口输出

### 附录1.3 STEP3

用 TKM 观察手指差值(软件使用方法请参考 TKM 手册 5.1.2 门限调试章节)。



附录 1-7 上位机调试

若按键差值为 160，则修改门限为调试差值的一半 80。

```
tkm_config.h x tk_lib.h main.c
#define TK_THD_CHANNEL_0 80
#define TK_THD_CHANNEL_1 80
#define TK_THD_CHANNEL_2 80
#define TK_THD_CHANNEL_3 80
#define TK_THD_CHANNEL_4 80
#define TK_THD_CHANNEL_5 80
#define TK_THD_CHANNEL_6 80
#define TK_THD_CHANNEL_7 80
```

附录 1-8 按键阈值调整

这样触摸按键功能就完成了，注意返回工程开发时将串口调试输出关闭。