文档编号：AN157

上海东软载波微电子有限公司

应用笔记

ES7P2131/2124 ES32H0403 触控 SDK 使用说明

修订历史

版本	修订日期	修改概要
V1.0	2023-10-8	初版发布

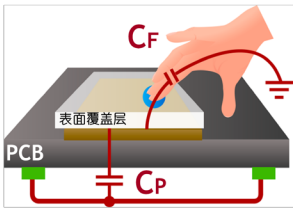
地　址：中国上海市徐汇区古美路 1515 号凤凰园 12 号楼 3 楼
E-mail：support@essemi.com
电　话：+86-21-60910333
传　真：+86-21-60914991
网　址：http://www.essemi.com/

目　录

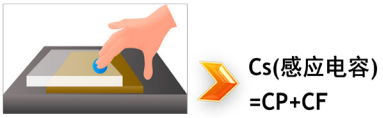
内容目录

第1章　触控实现原理

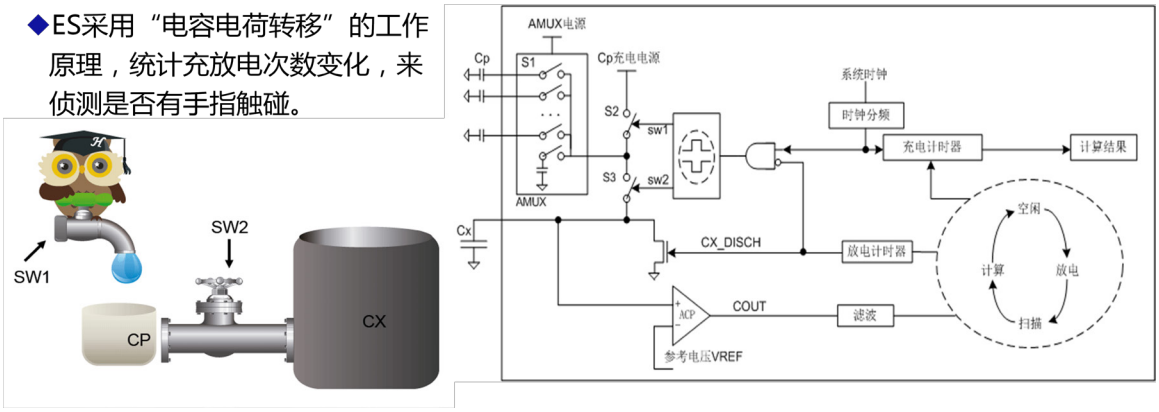
◆触摸按键的基本原理如右所示，按键即是一个焊盘，与地构成一个感应电容Cp，在周围环境不变的情况下电容值固定为微小值，具有固定的充放电时间，而当有一个导体向电极靠近时，会形成耦合电容Cf，改变固有的充放电时间，手指就是这样的导体。



◆ 如右图所示，通过芯片检测到充放电时间的改变判断是否有按键被按下。



◆ES采用“电容电荷转移”的工作原理，统计充放电次数变化，来侦测是否有手指触碰。



第2章 系统固件

2.1 概述

一方面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，另一方面为了能满足快速应用的需求，用户也可以选用本公司提供的 TKM 驱动库函数，其封装文件为：

tkm_芯片型号_版本号 .hrlib

2.2 系统环境

8 位机：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。

32 位机：Keil5 MDK-ARM V5.20 及以上版本，或者 IAR Embeded Workbench for ARM 8.11 及以上版本。

2.3 软件安装

8 位机：从官网下载芯片开发应用例程包。

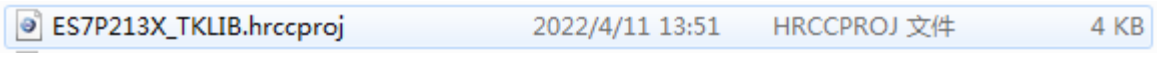
32 位机：从官网下载 keil5 或 IAR 芯片支持包及 ES32_SDK 应用例程。

从官网下载并安装上位机软件 TKM。在例程包中找到 TK 工程直接打开，或是在上位机软件中生成工程后打开。

2.4 快速入门

2.4.1 固件开发环境

8 位机：以 ES7P2131FHSH 为例。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：

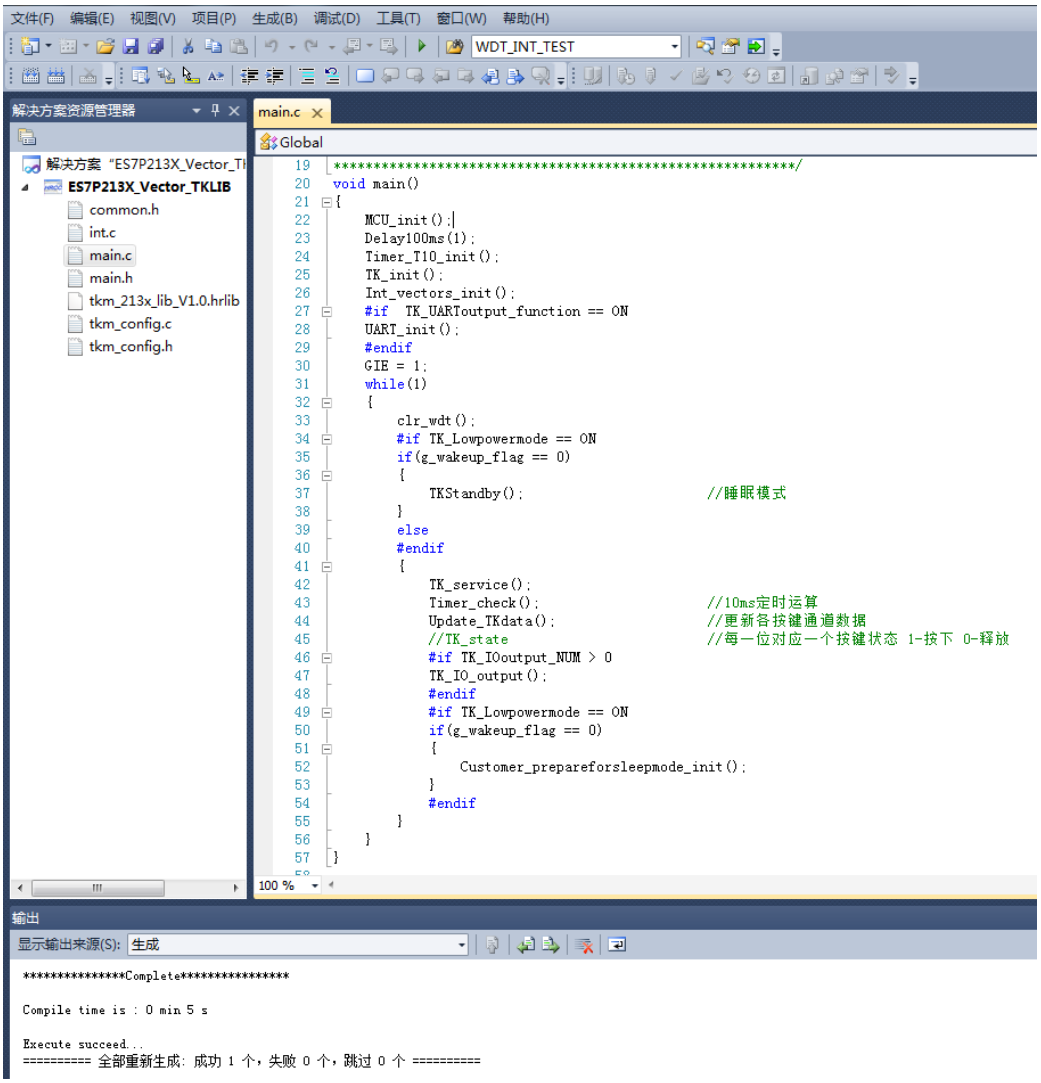


图 2-1 IDE 开发环境界面

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。

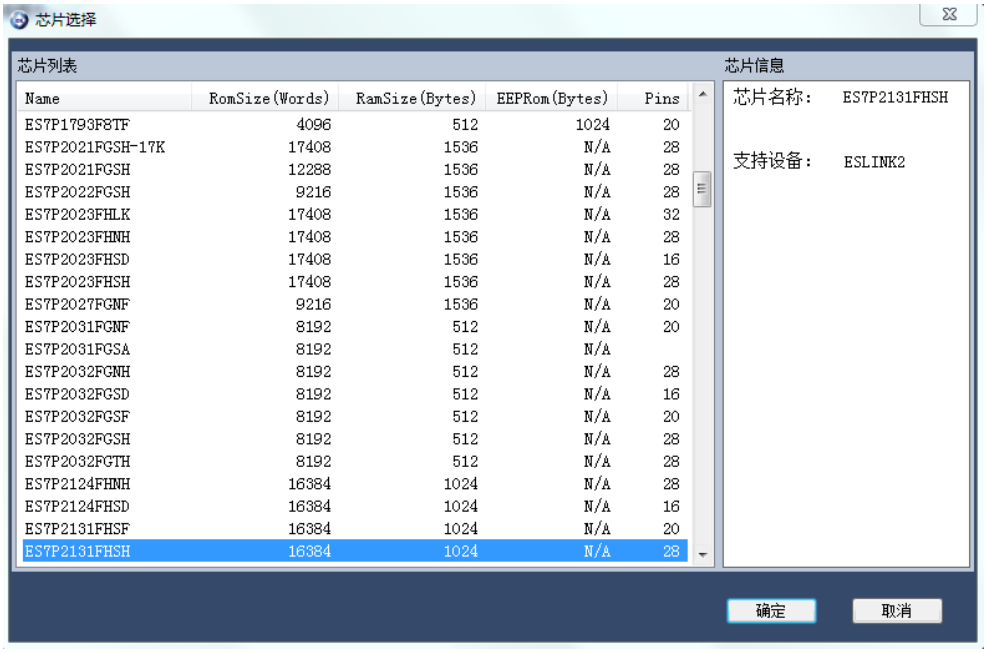


图 2-2　芯片选择界面

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。

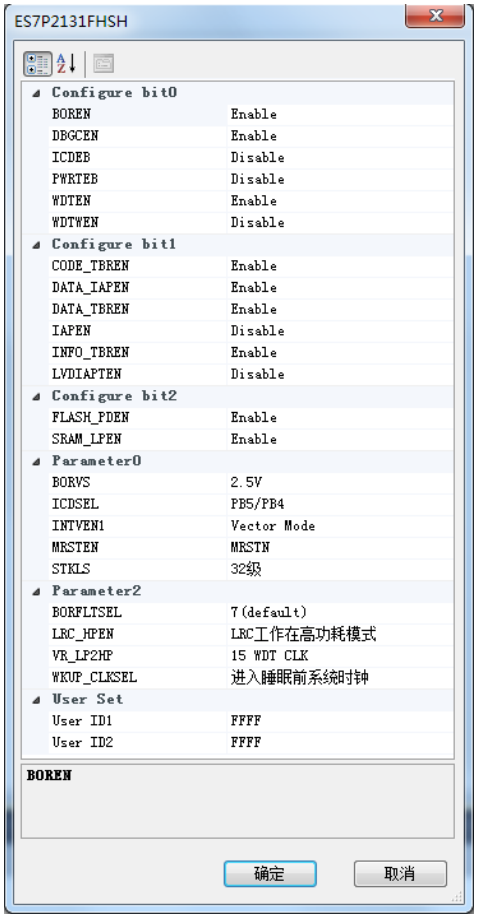


图 2-3　配置字界面

32 位机：打开工程文件

打开工程后再双击打开 main.c 文件，Keil 页面概览如下：

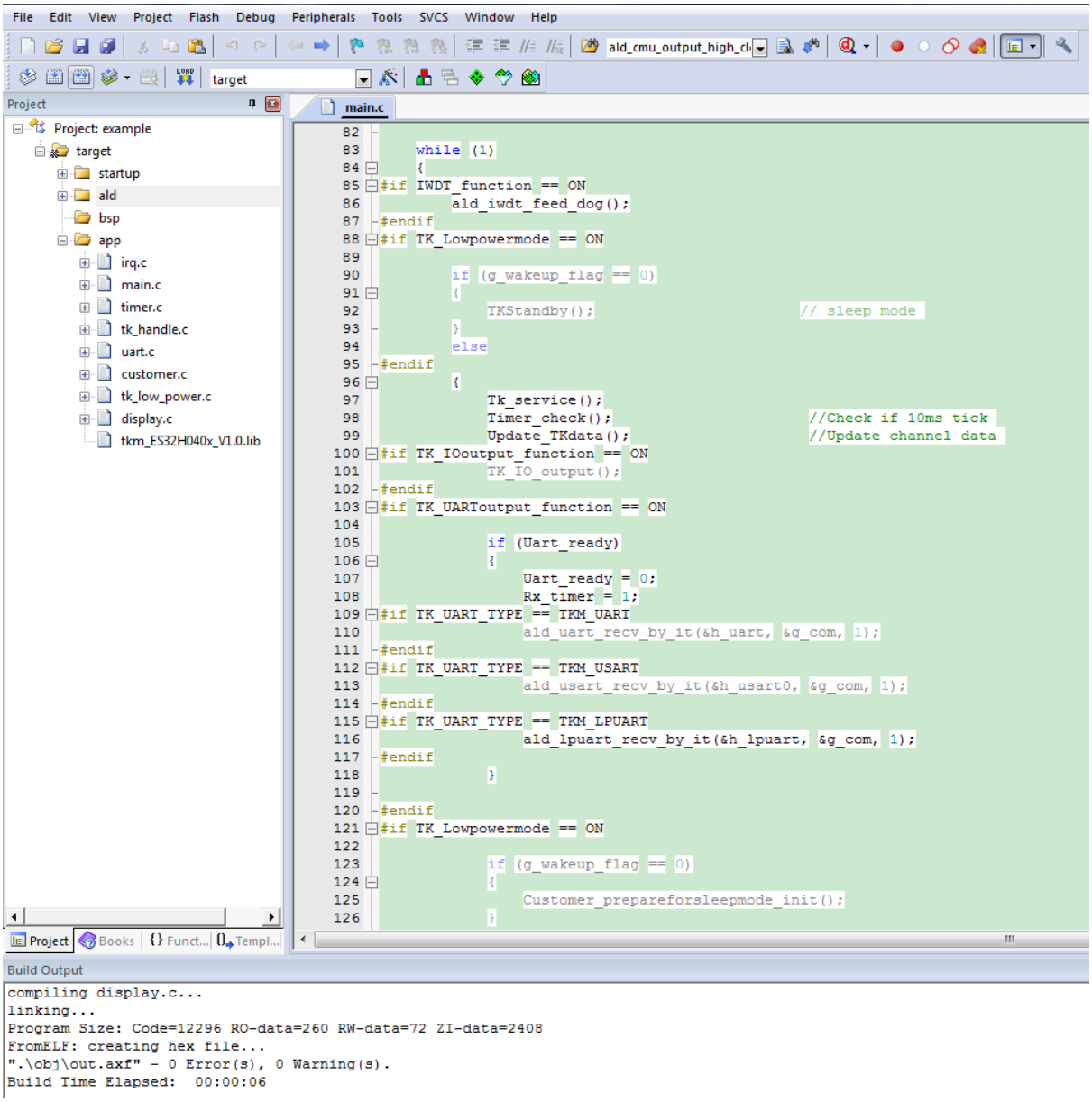


图 2-4　Keil 开发环境界面

本固件库目前支持：

1. ES7P2131，最多 20 个按键通道。
2. ES7P2124，最多 20 个按键通道。
3. ES32H0403，最多 31 个按键通道。

2.4.2 TK例程流程图

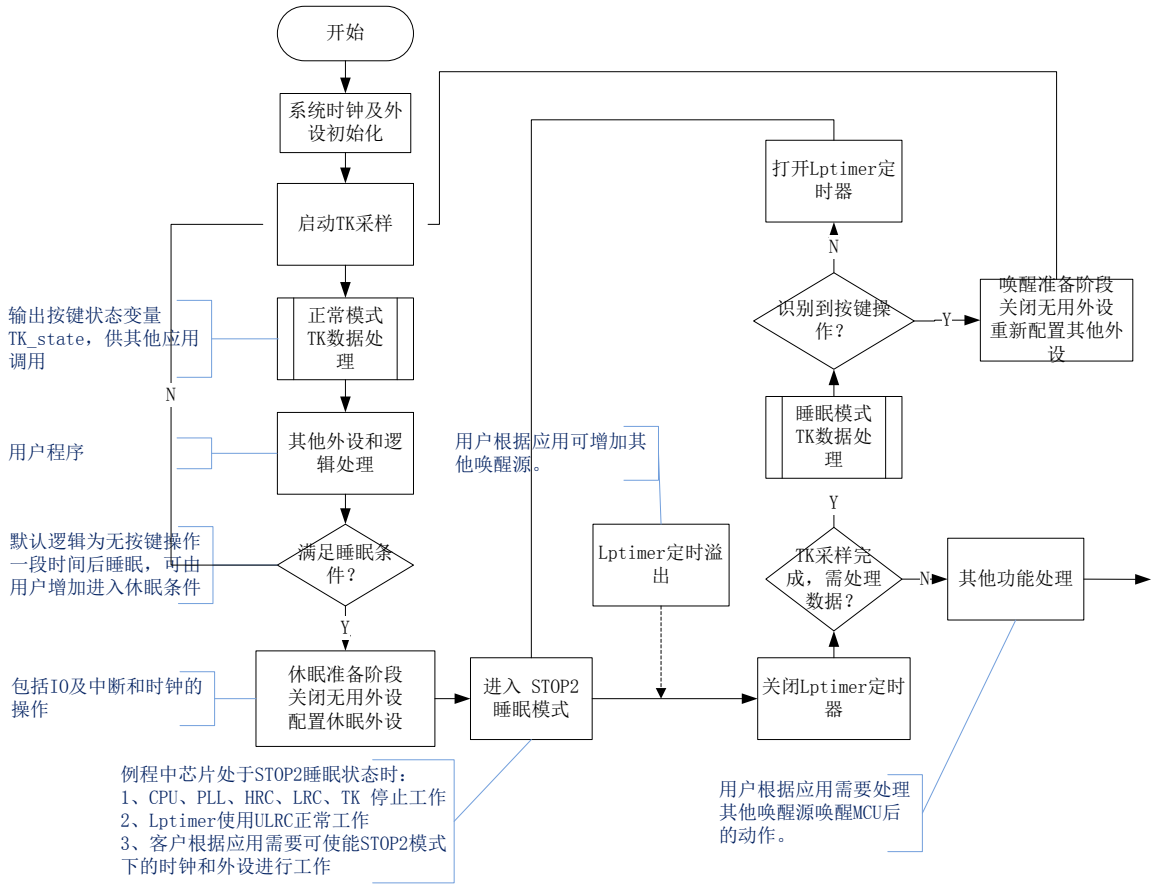


图 2-5 32 位机流程图

2.4.3 工程文档结构

8 位机固件工程包含如下文件：

1. 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。

2. main.h

主要用来存放全局变量的声明文件。

3. main.c

主函数处理，系统配置以及客户按键处理等函数。

4. tkm_config.h

主要用来存放 TK 各项参数的配置，与条件编译开关，与 tkm_config.c 中用到的函数与变量声明，此为核心文件，上位机参数更改的目标文件。

5. int.c

中断服务程序，存放 TK 中断，定时器中断，UART 串口中断处理函数。

6. tkm_xxx_xxx.hrlib

针对 TK 的各种数据处理算法函数库，包含按键，矩阵，滑轮，滑条及数据更新算法函数库。

7. tkm_config.c

TK 外围函数源文件，包含变量定义，TK 初始化函数，及可选数据输出函数，及 TK 服务函数，接口函数。

32 位机固件工程包含如下文件：

1. irq.c

中断服务程序，存放 TK 中断，定时器中断，UART 串口中断处理入口函数。

2. main.c

主函数处理过程。

3. timer.c

定时器配置及中断处理。

4. tk_handle.c

TK 外围函数源文件，包含变量定义，TK 初始化函数，及可选数据输出函数，及 TK 服务函数，接口函数。

5. uart.c

串口配置及中断处理，用于与 TKM 上位机通讯。

6. customer.c

预留函数，供客户编写按键应用及休眠唤醒处理。

7. tk_low_power.c

按键的休眠配置及处理过程。

8. display.c

可选的按键指示灯配置，一个 IO 对应一个按键，可接 LED 输出高低电平显示按键状态。

9. tkm_xxx_xxx.hrlib

针对 TK 的各种数据处理算法函数库，包含按键，矩阵，滑轮，滑条及数据更新算法函数库。

2.4.4 配置参数使用说明

为方便工程师使用触控库，本库所有常用的触控配置都集成在 tkm_config.h 文件中，请尽量在此文件中修改，其它文件请慎动。(8 位机与 32 位机以及不同版本之间存在差异，可能未使用所有参数)

◆ 按键通道设定

```
#define   TK_NUM                    3
```

定义了需要使用的按键个数，包含所有用到的触控通道以及滑条滑轮矩阵。

```
#define   TK_Channel_Sel            0x0000000E
```

对应通道选择寄存器 TKCHSL，应用时与触控按键相连引脚的通道应当被置位

```
#define   TK_Channel0               TK1
#define   TK_Channel1               TK2
#define   TK_Channel2               TK3
```

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，将按键重新进行了映射。TK_state 的状态位与之一一对应，如其 bit0 即为 TK_Channel0 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。

```
#define   TK_Threshold_Channel0     150
#define   TK_Threshold_Channel1     150
#define   TK_Threshold_Channel2     150
```

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

◆ 触控参数设定

```
#define   TK_Debounce_press         10        //连续按下的触发次数设定
#define   TK_Debounce_release       5         //连续离开的触发次数设定
```

类似于按键消抖的作用，连续满足设定次数按键按下或按键释放的判断才认为有真正的按键操作，数值越大越稳定但灵敏度会降低。

```
#define   TK_Samples_perscan        3         //采样次数累加设定3--16
```

将每次采样数据累加并做去最大最小值处理，最终每个通道用于运算的有效数据为(TK_Samples_perscan -2)个，数值越大越稳定但灵敏度会降低。

```
#define   Jitter_level1_Threshold   (TK_Threshold_Channel0*1.5*TK_NUM)
#define   Jitter_level2_Threshold   (TK_Threshold_Channel0*2*TK_NUM)
```

将所有通道最大最小值的差值累加，以此来衡量采样数据的稳定性。当数值超过 Jitter_level1_Threshold 时锁定基线不做更新处理。当数值超过 Jitter_level2_Threshold 时屏蔽按键不做响应。数值越大可接受干扰就越大，会有按键误触风险。

```
#define   Auto_Jitter_software      ON        // ON/OFF  软件跳频开关
```

与 Jitter_level 联动，使能时当数据抖动超过 Jitter_level1 则进行跳频。

```
#define   TK_Press_timeout          (20*100)  //超过此时间后强制取消此按下状态并平均基线
```

根据应用需求配置时间，可能与按键长按应用冲突。

```
#define   TK_BaseSamples_perscan    100       //基线更新周期设定越大基线更新越慢
```

用于设定基线更新的周期，若单个按键的充放电扫描时间为 150us，则基线更新时间约为：

150*TK_NUM*TK_Samples_perscan*TK_BaseSamples_perscan us

```
#define   TK_Mult_inhibition        0 //临键抑制系数  0无抑制  1:50% 2:25% 3:12.5% 4:6.25%
```

可选功能，硬件设计按键距离较近时用于解决手指按压临键误触问题

#define TK_Singlepress OFF //同一刻仅允许有一个最强按键按下

可选功能，可能与组合按键应用冲突。

#define TK_Guardsensor_output OFF //降耦功能开关 提升SNR

可选功能，需硬件支持，设计时在按键周围用 guard 通道包围。

#define TK_Threshold_release 9 //按键松开迟滞设定

判定按键释放条件，如此设定为差值小于门限的90%时认为按键松开。

#define Max_Minvalueoff_filter ON //去最大与最小值开关

默认使能，对数据做去最大最小值滤波处理。

#define Auto_Jitter_software OFF //软件抖频开关,该功能使用需打开去最大去最小开关

TK充放电频率抖频开关，数据抖动较大时自动修改TK工作频率寄存器配置。

#define Auto_Jitter_hardware OFF //硬件抖频开关,该功能使用需打开去最大去最小开关

TK充放电频率抖频开关，主频会自动变化，根据应用谨慎使用。

#define Jitter_Function OFF //容错开关

原按键按下与释放为判断连续满足判定条件N次后有效，使能容错后允许N次判定不连续。

#define Jitter_tolerance 0 //容错设定

与容错开关一起使用，用于设定不连续的次数。

◆ 低功耗设定

#define TK_Lowpowermode OFF //ON /OFF

需要低功耗应用时使能

#define TK_Wdttimer_sleepsetting WDT_128ms //ms 16/32/64/128/256/512/1024/2046

设定低功耗定时唤醒时间(8 位机)

#define TK_Mode0_nokeytoswitch_time 10*100 //10S 0--65535从正常模式进入休眠模式切换时间

设定多久进入低功耗状态

#define TK_LP_dataprocess_mode Poll //Merge/Poll

设定按键扫描方式为轮询或合并通道

#define TK_LPtimer_sleepsetting LPtimer_300ms //ms 50/100/200/300/400/500

设定低功耗定时唤醒时间(32 位机)

#define TK_Sleep_Threshold 100

通道合并模式下按键门限

#define TK_NUM_LP TK_NUM//触控按键个数1--20

低功耗状态时需要的按键个数

#define TK_Channel_Sel_LP TK_Channel_Sel

低功耗时通道选择，对应寄存器 TKCHSL

#define TK_Channel0_LP TK_Channel0//以下只有前TK_NUM_LP个定义有效

#define TK_Channel1_LP TK_Channel1

#define TK_Channel2_LP TK_Channel2

低功耗下应用的按键通道

#define TK_Threshold_Channel0_LP TK_Sleep_Threshold

#define TK_Threshold_Channel1_LP TK_Sleep_Threshold

#define TK_Threshold_Channel2_LP TK_Sleep_Threshold

低功耗下按键门限

#define PINT_WAKE_function OFF

```
#define   KINT_WAKE_function                OFF
```

在按键低功耗模式下需要用 PINT 或 KINT 唤醒时可使能

◆ UART 输出设定

```
#define   TK_UARToutput_function            OFF          // ON/ OFF，功能开关
```

可选功能，与上位机通讯串口调试开关。

```
#define   TK_UARToutput_port                PB45         // PB45/ PA01
```

数据输出口选择(8位机)

```
#define   TK_UART_TYPE                      TKM_LPUART// TKM_UART/TKM_USART/TKM_LPUART
```

数据输出口选择(32位机)

```
#define   TK_UARToutput_Baudrate            9600         // 115200   57600 38400 19200 .....9600
```

如果使用该波特率通信不成功，用户可尝试将波特率调整为9600，再作测试。

```
#define   TK_UARToutput_Databit             8            // 4,5,6,7,8.9
#define   TK_UARToutput_Stopbit             1            // 1,2,3(1.5)
#define   TK_UARToutput_Verifybit           No           //No
```

串口相关配置，一般不做修改。

◆ IO 输出设定

```
#define   TK_IOoutput_function              OFF      // ON/ OFF
```

可选功能，IO指示按键状态输出功能，对应IO连接LED后显示按键状态。

```
#define   TK_IOoutput_NUM                   0        // 0无输出  按顺序对应  TK_Channelx  的状态
```

IO输出数量设定，一般根据按键数量决定，与按键一一对应。

```
#define   TK_IOLED_Function                 Indicate  // Indicate/Toggle
```

设定指示灯指示方式，按下亮松开灭或是开关触发模式。

```
#define   TK_IOoutput_port0                 PB7      // IO bit
#define   TK_IOoutput_port1                 PE0
#define   TK_IOoutput_port2                 PE1

#define   TK_IOoutput_trise0                PBT7     // IO TRISE bit
#define   TK_IOoutput_trise1                PET0
#define   TK_IOoutput_trise2                PET1
```

定义指示灯输出管脚通道，指示灯用于指示按键按下状态。（8 位机）

```
#define   TK_IOoutput_port0                 GPIOA              // PA PB PC    Any bit
#define   TK_IOoutput_port1                 GPIOB
#define   TK_IOoutput_port2                 GPIOB

#define   TK_IOoutput_pin0                  GPIO_PIN_8         // PA PB PC    TRISE bit
#define   TK_IOoutput_pin1                  GPIO_PIN_15
#define   TK_IOoutput_pin2                  GPIO_PIN_15
```

定义指示灯输出管脚通道，指示灯用于指示按键按下状态。（32 位机）

◆ 寄存器设定

以下设定请详查芯片数据手册定义，慎改！

```
#define   TK_reg_TKCTL0          0x38//设定按键合并轮询模式以及模块启动
#define   TK_reg_TKCTL1          0x02//充放电占空比设定
#define   TK_reg_TKCTL2          0x20//充放电频率设定
#define   TK_reg_TKCTL3          0x00//比较器迟滞设定
#define   TK_reg_TKCTL4          0x32//充放电功耗及放电时间设定
#define   TK_reg_TKCTL5          0x3D//充电电压及比较器滤波设定
#define   TK_reg_TKCTL6          0x80//TK模块参考电压设定
#define   TK_reg_TKINTE          0x0F//TK模块中断设定
#define   TK_reg_TKFJCTL         0x01//充放电频率抖频设定

#define   TK_reg_CON0            0x00508519//默认工作模式配置，详细内容参考手册
#define   TK_reg_CON0_bk1        0x00308519//用于软件抖频，设定切换的频率
#define   TK_reg_CON0_bk2        0x00708519//预留未用
#define   TK_reg_CON1            0x6038047B//默认工作模式配置，详细内容参考手册
#define   TK_reg_SFJTR           0//硬件抖频设定
#define   TK_reg_CON0_merge      0x00528508//睡眠时的合并模式按键配置
#define   TK_reg_CON0_poll       0x00528518//睡眠时的轮询模式按键配置
#define   TK_reg_CON1_sleep      0x6038047B//睡眠时的按键配置
```

2.4.5 常用变量使用说明

本节提到的常用变量都是库函数中某一特定算法的输出值，这些变量存在于 TK_Service 中，在调试时，可以通过这些值来观察按键效果。

程序运行状态控制字：Opr_state

Bit15	扫描使能 Scan Enable Bit	1-On	0-Off
Bit14	扫描数据满 Tempdata is Full	1-Full	0-Receive
Bit13	去抖采样使能 Jillter Sample Enable Bit	1-Enable	0-Disable
Bit12	一次扫描完成标志 Once Scan Complete Bit	1-Complete	0-Busy
Bit11	滑条处理标志 Slider Bit	1-Touched	0- Not Touched
Bit10	滑轮处理标志 Wheel Bit	1-Touched	0- Not Touched
Bit9	矩阵行状态标志 Matrix Row Statue	1-Touched	0- Not Touched
Bit8	矩阵列状态标志 Matrix Column Status	1-Touched	0- Not Touched
Bit7	模式状态标志 Mode Status Bit	1-Mode 2	0-Mode 1
Bit6	模式切换标志 Mode Switch Start Switching	1-Mode need switching 0- switching done	
Bit5	低功耗模式使能标志 Power save Flag Setting	1-Enable Power save 0-Disable Power save function	
Bit4	保留		
Bit3	保留		
Bit2	保留		
Bit1	保留		
Bit0	保留		

按键通道数： TK_chnum;
系统使用 TK 功能通道总数。

当前扫描通道扫描次数计数：Tkscan_sampcounter;
与宏定义 TK_Samples_perscan 相关，计数超过该定义值时，进行下一步数据处理。

采样数据：TK_Value_Arr[TK_NUM].tk_value_origin;
采集 TK 通道上未经处理的数据。

滤波数据：TK_Value_Arr[TK_NUM].tk_value_filter;
对采样数据进行软件滤波算法后的数据。

基线数据：TK_Value_Arr[TK_NUM].tk_value_average;
在没有按键事件时，对采样数据进行处理后的值，作为是否产生按键事件的基准值。

差值数据：TK_Value_Arr[TK_NUM].tk_D_value;
滤波数据与基线数据之差，当该差值超过门限时，判为有按键产生。

数据波动值：TK_jitter_Value;
由各通道的采样数据的最大最小值之差累加得到，反应当前环境状态稳定情况，以此作为按键处理的依据。

按键超时计数：TK_Value_Arr[TK_NUM].tk_timeout_counter;
与 TK_Press_timeout 相关，所有通道按下连续时间不能超过该宏定义时长，如果超过，则自动更新基线。

低功耗基线数据：TK_md2average_backup[TK_NUM];
用于低功耗模式下的按键处理。

低功耗采样数据：TK_md2value_origin[TK_NUM];
用于低功耗模式下的按键处理。

按键按下累积次数计数：TK_Value_Arr[TK_NUM].tk_press_table;
超过该计数判为有按下事件。

按键离开累积次数计数：TK_Value_Arr[TK_NUM].tk_release_table;
超过该计数判为有按键离开事件。

当前按键值：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。

当前按键值：TK_state_single;
TK_Singlepress 使能时有效，每一位对应一个按键状态。1 代表有键按下，0 代表松开。若为三个或三个以下按键按下时，仅差值最大的对应位为 1，若三个以上按键按下则值为零，即 TK_state_single 值为零时不一定是没有键按下也可能是有三个以上的按键被按下。

基线锁定恢复延时计数：Lock_averageconter;
判断到有采样数据与基线数据差值大于门限后会锁定基线，在恢复更新基线操作时延时处理。

基线更新计数：TK_Base_percounter;
每进行一次数据处理时累加，达到设定的TK_BaseSamples_perscan值后更新基线数据。

无按键操作定时计数：TK_mode1_nopresstimer;
低功耗模式TK_Lowpowermode使能时有效，用于计时无按键操作时间。

睡眠标志：g_wakeup_flag;
低功耗模式TK_Lowpowermode使能时有效，用于控制执行按键正常模式和休眠模式，无按键操作到达设定时间TK_Mode0_nokeytoswitch_time后进入休眠模式。

2.4.6 函数说明

函 数 名	TK_service ()
源文件名	tkm_config.c
函数说明	TK SDK 服务函数主函数
输入参数	tkm_config.h 中各参数
返 回 值	全局变量按键状态值
调用方法	初始化完成后可直接调用

函 数 名	Timer_check()
源文件名	tkm_config.c
函数说明	定时计数变量的累加及达到设定值后的处理
输入参数	各定时计数变量
返 回 值	按键相关数据
调用方法	初始化完成后可直接调用

函 数 名	Update_TKdata()
源文件名	tkm_config.c
函数说明	对采样数据进行初步处理
输入参数	未做处理的采样数据
返 回 值	各通道最大最小值及多次采样数据累加值
调用方法	初始化完成后可直接调用

函 数 名	Singlekeyprocess()
源文件名	tkm_xxx_xxx.hrlib
函数说明	单按键模式判断处理函数，选择信号最强的按键输出
输入参数	全局变量中的TK_Value_Arr[TK_NUM].tk_D_value
返 回 值	TK_state_single最强按键状态
调用方法	数据处理函数后直接调用

函 数 名	TKStandby()
源文件名	tkm_config.c
函数说明	定功耗模式下按键数据的处理
输入参数	正常模式下按键数据及未做处理的采样数据
返 回 值	按键相关数据
调用方法	初始化完成后可直接调用

在 main.c 中已添加部分常用的客户版空函数，并有详细的使用时间节点说明，客户只需要添加上合适的动作指令，就能在空函数中完成主流的触控动作应用，函数如下：

```
void Customer_Keystatejustpress(void)
void Customer_Keystatejustrelease(void)
void Customer_prepareforsleepmode_init(void)
void Customer_wakeupfromsleepmode_init(void)
```

2.4.7 编译运行

8 位机：在上述参数设定下，在工程名上右键重新生成，编译通过时如下所示：

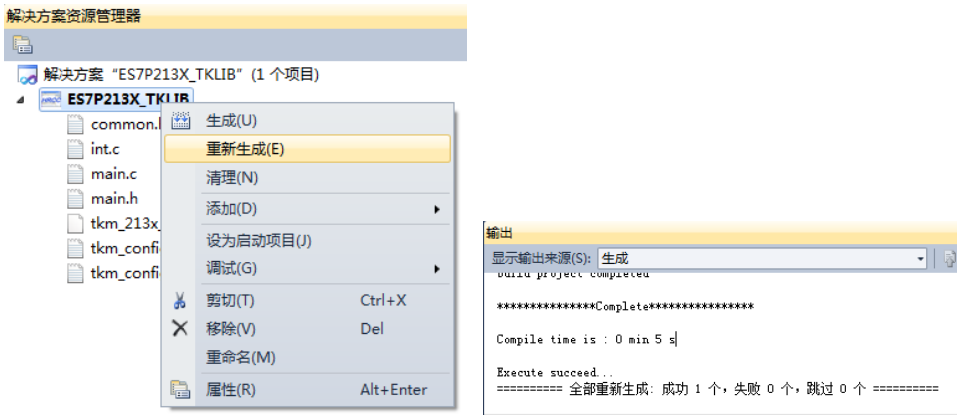


图 2-4 编译工程

在项目配置字中配置 ICD 接口功能，如下所示：

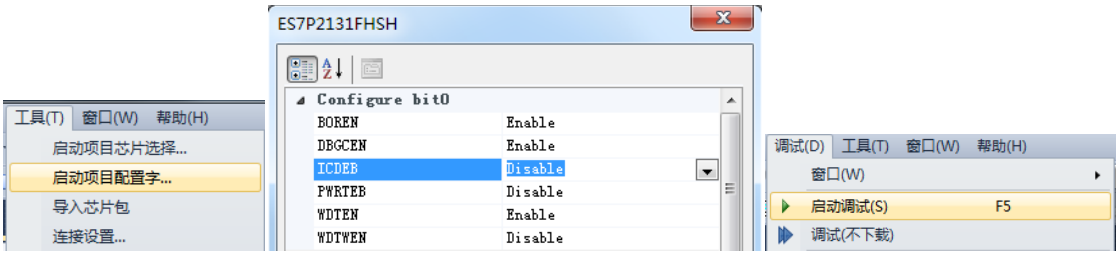


图 2-5 程序调试配置

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：

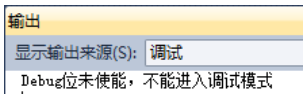


图 2-6 程序烧录

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：

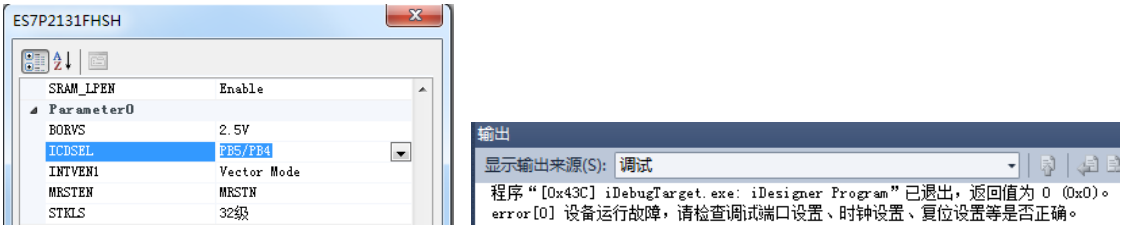


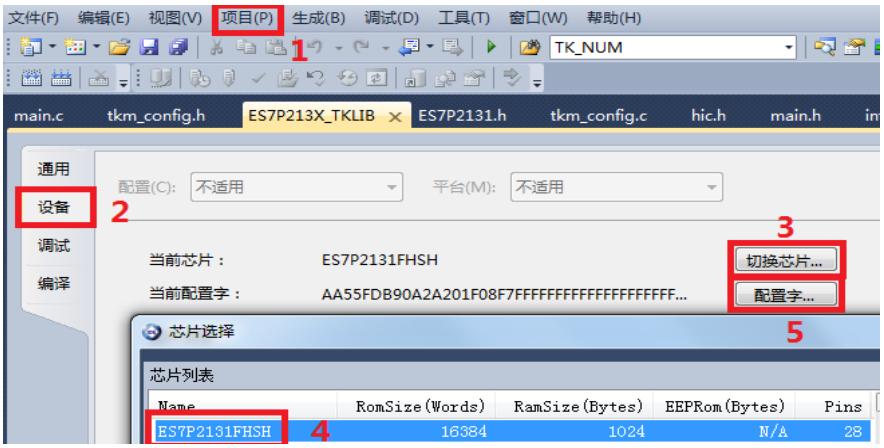
图 2-7 程序调试

32 位机：请参考 Keil 编译及 debug 使用相关说明，此处不做详述。

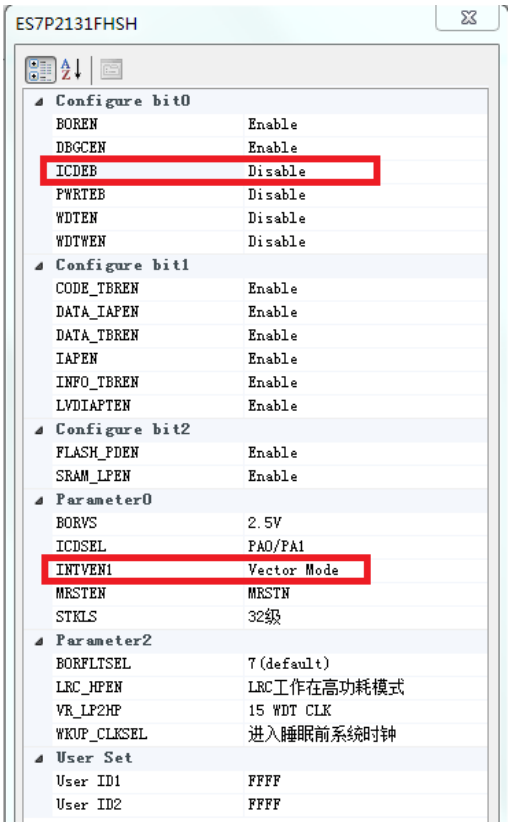
第3章　按键调试

3. 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。32 位机可直接进行 STEP2 操作，调试过程可参考此 8 位机示例。

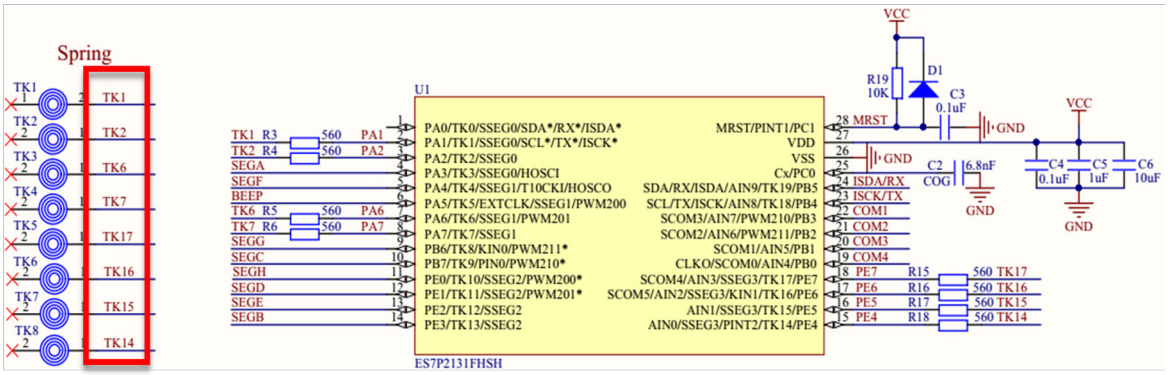


配置字一般修改以下两项即可



3. 2 STEP2

例如应用设计的原理图如下，是一个 8 按键的设计。



根据硬件设计修改按键配置

```
tkm_config.h | ES7P213X_TKLIB | ES7P2131.h | tkm_config.c

13
14  /*************************************************
15  #define  TK_NUM                    8
16
17  #define  TK_reg_TKSELL             0xC6
18  #define  TK_reg_TKSELM             0xC0
19  #define  TK_reg_TKSELH             0x03
20
21  #define  TK_Channel0               TK1
22  #define  TK_Channel1               TK2
23  #define  TK_Channel2               TK6
24  #define  TK_Channel3               TK7
25  #define  TK_Channel4               TK17
26  #define  TK_Channel5               TK16
27  #define  TK_Channel6               TK15
28  #define  TK_Channel7               TK14
```

按键个数

通道寄存器设置

通道映射设置

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 | ES7P213X_TKLIB | ES7P2131.h | tkm_config.c

43  //#define  TK_Threshold              80
44  #define  TK_Threshold_Channel0     50      //
45  #define  TK_Threshold_Channel1     50      //
46  #define  TK_Threshold_Channel2     50      //
47  #define  TK_Threshold_Channel3     50      //
48  #define  TK_Threshold_Channel4     50      //
49  #define  TK_Threshold_Channel5     50      //
50  #define  TK_Threshold_Channel6     50      //
51  #define  TK_Threshold_Channel7     50      //
```

根据电路原理设计选择数据输出口，打开串口调试功能

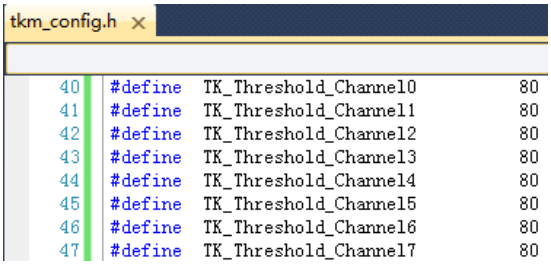
```
tkm_config.h

127  //调试配置
128  #define  TK_UARToutput_function        ON
129  #define  TK_UARToutput_port            PB45
130  #define  TK_UARToutput_Baudrate        9600
131  #define  TK_UARToutput_Databit         8
132  #define  TK_UARToutput_Stopbit         1
133  #define  TK_UARToutput_Verifybit       No
```

3. 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) 。

修改门限为调试差值的一半



这样一个触摸按键应用就完成了，回到工程继续开发时需将串口调试输出关闭。