

文档编号: AN2046

上海东软载波微电子有限公司

应用笔记

ES32H0403 触控 SDK 使用说明

修订历史

版本	修订日期	修改概要
V1.0	2024-07-19	初版, 本说明仅针对 ES32H0403 V1.1 版本及以上触控 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.....	3
第 1 章 开发环境.....	5
第 2 章 系统固件.....	6
2.1 概述	6
2.2 流程图.....	6
2.3 工程结构.....	7
2.3.1 startup 文件夹	7
2.3.2 ald 文件夹.....	7
2.3.3 tk_bsp 文件夹.....	7
2.3.4 app 文件夹	7
2.4 参数说明	8
2.4.1 按键通道设定	8
2.4.2 触控调试设定	8
2.4.3 休眠设定.....	9
2.4.4 寄存器设定	9
2.5 变量说明	9
2.5.1 按键状态: tk_state;	9
2.5.2 按键计时: tk_arr_t[],tk_to_cnt;.....	9
2.5.3 10ms 计时时基: timer_10ms;	9
2.6 函数说明	10
2.6.1 tk_customer.c 文件.....	10
2.6.2 tk_low_power.c 文件	10
2.6.3 tk_timer.c 文件.....	11
2.6.4 tk_lib.c 文件.....	11
2.6.5 tk_ES32H0403_V1.1.hrlib 文件	12
第 3 章 工程应用	14
3.1 概述	14
3.2 初始化部分	14
3.3 主循环部分	14
3.4 中断部分	14
3.5 休眠部分	14
第 4 章 常见问题.....	16
4.1 按键误触.....	16
4.1.1 软件程序.....	16
4.1.2 电源供电.....	16
4.1.3 整机系统.....	16
4.1.4 结构安装.....	16
4.1.5 环境影响.....	16
4.2 结果异常	16
4.2.1 转换结果偏小	16
4.2.2 转换结果偏大	16

4.3	EMC 测试	17
4.3.1	CS 测试	17
4.3.2	EFT 测试	17
4.3.3	EMI 测试.....	17
附录 1	版本差异	18
附录 1.1	空间资源.....	18
附录 1.2	命名规则.....	18
附录 2	阈值调试.....	20
附录 2.1	STEP1	20
附录 2.2	STEP2	22

第1章 开发环境

从官网下载芯片支持包及芯片开发 SDK。

Keil5 芯片支持包下载网址 <https://www.essemi.com/index/article/plist?cid=141>

适用于 Keil5 MDK-ARM 5.20 及以上版本，用于支持我司 ES32 系列的 ARM 内核芯片。

IAR 芯片支持包下载网址 <https://www.essemi.com/index/article/plist?cid=141>

适用于 IAR Embedded Workbench for ARM 8.11 及以上版本，用于支持我司 ARM 内核芯片。

嵌入式软件开发包下载网址 <https://www.essemi.com/index/product/detail?id=799>

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，其封装文件为：tk_芯片型号_版本号 .hrlib

2.2 流程图

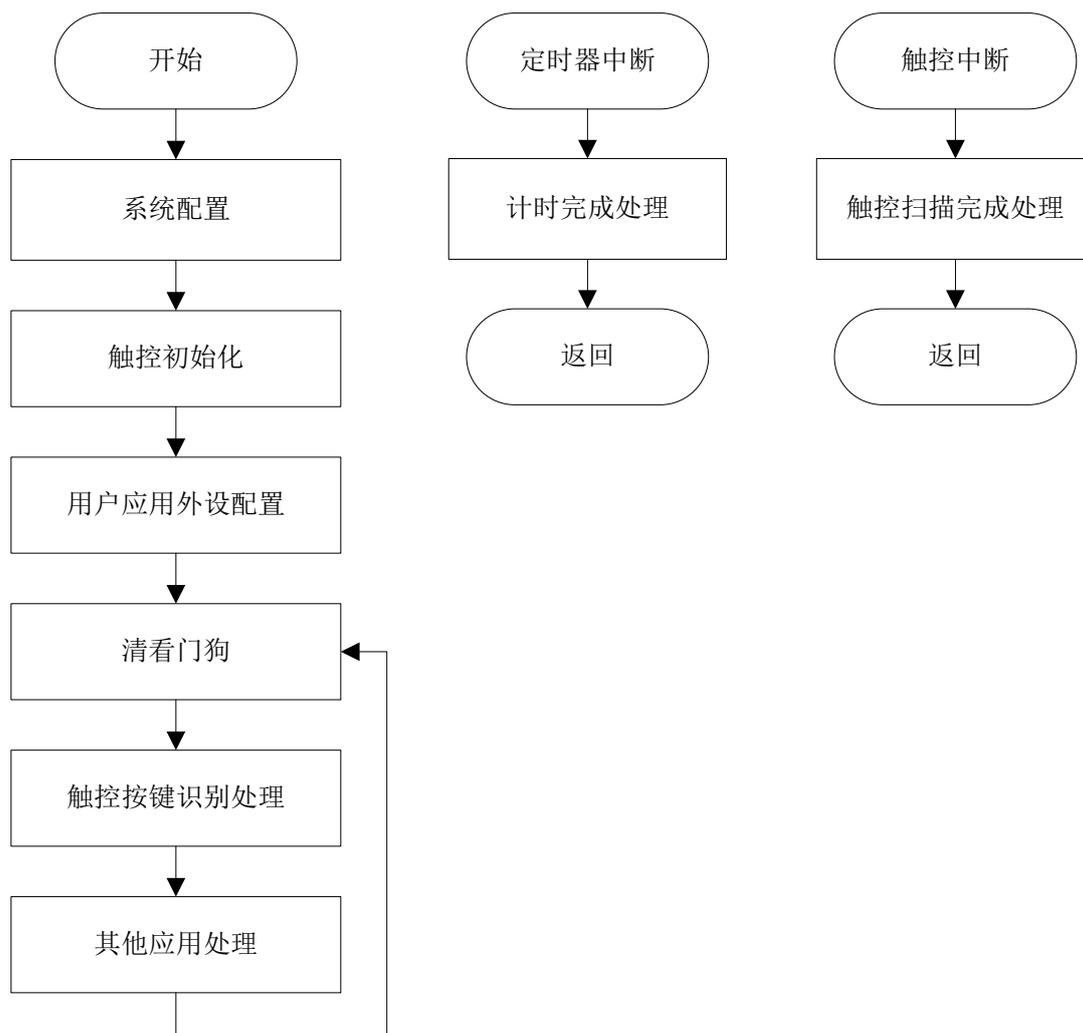


图 2-1 中断及主流程图

2.3 工程结构

2.3.1 startup 文件夹

startup_es32h040x.s

工程启动文件。

2.3.2 ald 文件夹

存放 ald 库驱动文件。

2.3.3 tk_bsp 文件夹

tk_customer.c

预留应用层按键初始化及按键识别处理函数。

tk_low_power.c

存放按键低功耗处理相关函数。

tk_timer.c

定时器初始化函数，用户可根据需求修改定时时间，同时需要修改中断服务函数及定时时基。

tk_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。

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_xxxx_Vx.x.hrlib

触控内部数据处理算法函数库。

2.3.4 app 文件夹

irq.c

应用系统其他中断处理函数入口。

Main.c

函数处理主循环。

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	1~31 (ES32H0403)	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	0x00000380	/	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	/	TK0~28TK30~31 (ES32H0403)	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_BAUD (文件 tk_uart.h 中)	9600	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。

参数名	默认值	设定范围	作用
TK_UART_TYPE (文件 tk_uart.h 中)	TKM_UART	TKM_UART/USART/LPUART	数据输出端口选择
UART_FUNC (文件 tk_uart.h 中)	GPIO_FUNC_4	GPIO_FUNC_2/3/4	
UART_PORT (文件 tk_uart.h 中)	GPIOA	GPIOA/B	
UART_TX (文件 tk_uart.h 中)	GPIO_PIN_13	-	
UART_RX (文件 tk_uart.h 中)	GPIO_PIN_14	-	

当芯片有多组串口时，可选择需要的串口组。

2.4.3 休眠设定

参数名	默认值	设定范围	作用
TK_LOWPOWER_MODE	OFF	ON/OFF	休眠控制开关

使能时启用按键低功耗模式。

参数名	默认值	设定范围	作用
TK_LP_IDLE_TIME	1000	-	进入休眠等待时间

无按键操作时间，即由正常模式到低功耗模式的等待时间。

2.4.4 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_CON0	0x508519U//充放电频率 2M
#define TK_REG_CON0_BK	0x308519U//充放电频率 4M
#define TK_REG_CON1_VDD	0x6038047FU//充电电压设定 VDD
#define TK_REG_CON1_REF	0x6038047DU//充电电压设定内部 2.6V
#define TK_REG_GAIN	0x000FFFFFFU//放大倍数设定
#define TK_REG_SFJTR	0x0U//充放电抖频设定

2.5 变量说明

本节中的变量是库函数中算法的输出值，在调试时，可以通过这些值来观察按键效果。

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 tk_customer.c 文件

函数名	void customer_key_handler(void)
函数说明	用户按键处理预留函数，用户根据按键状态变量 tk_state 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	void customer_key_init(void)
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

2.6.2 tk_low_power.c 文件

函数名	void tk_lp_prep(void)
函数说明	按键休眠准备函数，关闭无用外设及进行 TK 低功耗配置，确定休眠基线
输入参数	无
返回值	无
调用方法	在满足进入休眠条件时调用

函数名	void tk_wakeup_prep(void)
函数说明	按键唤醒准备函数，重启外设及进行 TK 功能配置
输入参数	无
返回值	无
调用方法	在满足唤醒条件时调用

函数名	void tk_lp_enter(void)
函数说明	执行休眠过程，并设定低功耗定时器进行唤醒
输入参数	无
返回值	无
调用方法	低功耗模式下无按键操作时调用

函数名	void tk_lp_init(void)
函数说明	系统及外设休眠配置函数
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	<code>void LP16T0_Handler(void)</code>
函数说明	低功耗定时器中断处理
输入参数	无
返回值	无
调用方法	在低功耗定时器计时溢出时调用

函数名	<code>void __tk_lp_value_get(void)</code>
函数说明	低功耗模式下的按键数据处理过程
输入参数	无
返回值	无
调用方法	在低功耗模式下唤醒后调用

函数名	<code>void tk_lp_handler(void)</code>
函数说明	低功耗模式下的按键状态处理过程
输入参数	无
返回值	无
调用方法	在低功耗模式下数据处理完成后调用

2.6.3 tk_timer.c 文件

函数名	<code>void BS16T0_Handler(void)</code>
函数说明	定时器中断处理函数，根据应用修改 10ms 计时时基
输入参数	无
返回值	无
调用方法	在计时溢出时调用

函数名	<code>void bs16t0_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

2.6.4 tk_lib.c 文件

函数名	<code>void TKS_Handler(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 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 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

函数名	<code>void bspTk_handler(void)</code>
函数说明	触控及低功耗处理等封装函数。
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 bspTk_init(void)</code>
函数说明	触控模块及变量等初始化封装函数。
输入参数	无
返回值	无
调用方法	在系统初始化阶段调用

2.6.5 tk_ES32H0403_V1.1.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
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_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

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：`bsp_tk_init()`（初始化）、`bsp_tk_handler()`（主循环）、`TKS_Handler()`（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

`void tk_pin(void)`

触摸引脚配置通用函数，主要防止用户误操作部分通道引脚而引起功能异常。为了节省程序空间，可将此函数屏蔽，相关的触摸引脚配置保持默认。

`void bs16t0_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

`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(unsigned char ch)` 来做异常恢复处理（使用前请做充分地咨询与评估）。

`void bsp_tk_handler(void)`

请确保函数处在主循环中，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

`Void BS16T0_Handler(void)`

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。

`void TKS_Handler(void)`

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。

3.5 休眠部分

因需要处理触控数据及按键状态，此触控例程中的休眠模式是指：系统进入休眠，到 `LPtimer` 定时唤醒，再到触控处理的过程，而非一般所指的芯片处于低功耗状态。

`void tk_lp_prep(void)`

进入休眠的条件默认为满足无按键操作一段时间（由宏 `TK_LP_IDLE_TIME` 设定），应用时系统若需满足其他条件才可进入休眠模式，可在调用此函数的条件判断语句中加入判断条件。

`void tk_wakeup_prep(void)`

注意在休眠模式前后关闭（推荐在 `tk_lp_prep()` 中关闭相关外设使能及时钟）或开启（推荐在 `tk_wakeup_prep()` 中重新启用相关外设）相关的外设以降低系统功耗。

`void LP16T0_Handler(void)`

默认由 `LPtimer` 唤醒，根据应用需要可自行添加其他唤醒源，注意调用 `tk_wakeup_prep()` 函数返回正常模式。

第4章 常见问题

4.1 按键误触

软件调试发生误触时，考虑加快基线更新速度以适应变化，同时增大 TK_JITTER_THD_LV1 值，否则会锁定基线。另外可增大 TK_DEBOUNCE_PRESS 值增强滤波效果。应用允许时可考虑利用 TK_JITTER_THD_LV2 屏蔽按键。

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，根据数据变化做屏蔽或自适应操作。

电机影响，可尝试根据采样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 版本差异

附录1.1 空间资源

V1.0

```
Program Size: Code=8280 RO-data=260 RW-data=56 ZI-data=2240
```

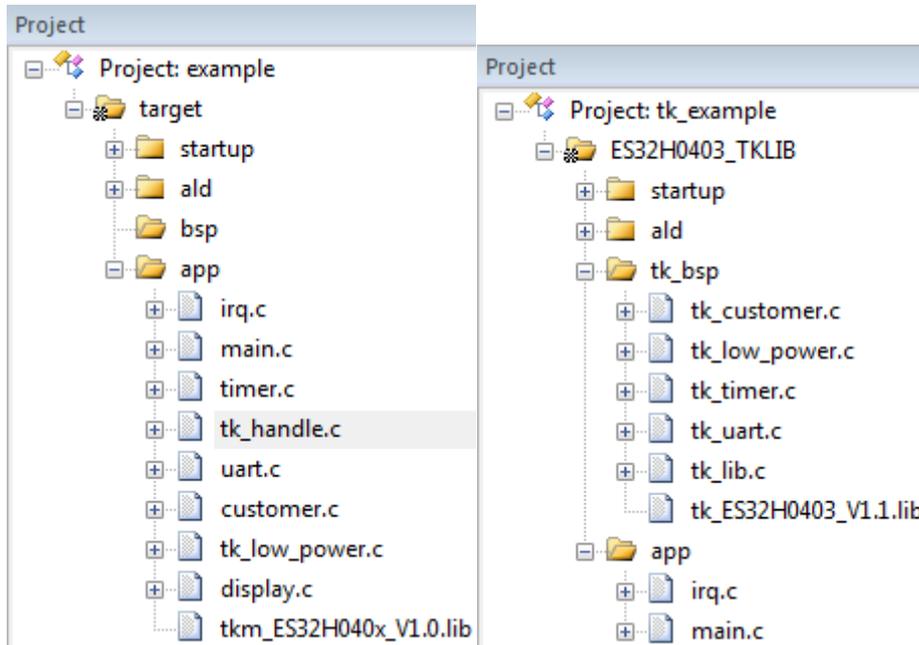
V1.1

```
Program Size: Code=8388 RO-data=528 RW-data=68 ZI-data=2108
```

附录 1-1 空间资源对比（3 个按键为例）

附录1.2 命名规则

将文件名改为标准形式，与触控相关的源代码，名称前加“tk_”前缀，触控相关文件放入 tk_bsp 文件夹



附录 1-2 文件名对比

将宏定义由首字母大写改为全大写

#define TK_NUM	3	#define TK_NUM	3
#define TK_Channel_Sel	0x00000380	#define TK_CHANNEL_SEL	0x00000380
#define TK_Channel0	TK7	#define TK_CHANNEL_0	TK_7
#define TK_Channel1	TK8	#define TK_CHANNEL_1	TK_8
#define TK_Channel2	TK9	#define TK_CHANNEL_2	TK_9

附录 1-3 宏定义对比

函数名及变量名由首字母大写改为全小写加下划线

```
void Timer_check(void)
{
    uint8_t i;

    if (Timer_10ms > Timercounter_10ms)
    {
        Timer_10ms = 0;

        if (BaseLineInitStart == 1)
        {
            BaseLineInitStart = 0;
            BaseLineInitCount++;
        }
    }
}

void timer_check(void)
{
    if (timer_10ms < 1)
        return;

    timer_10ms = 0;

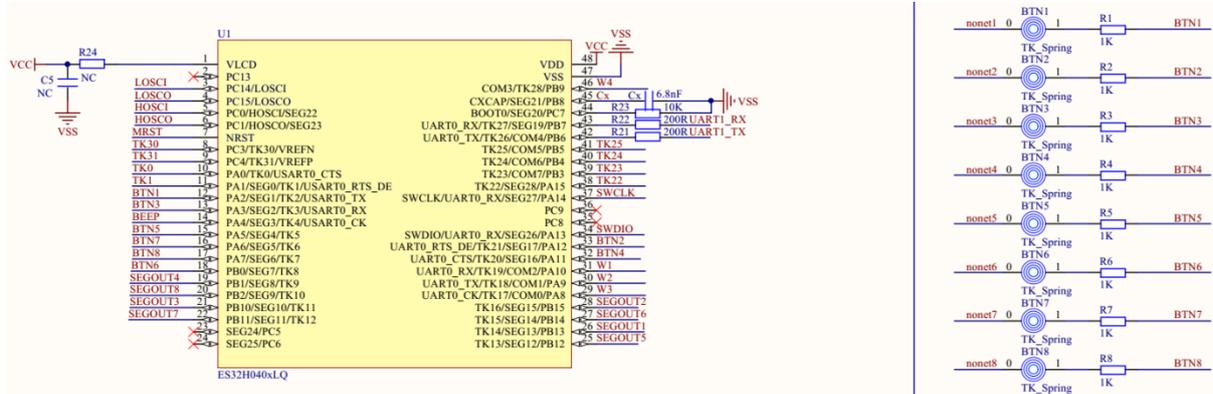
    tk_base_up_lock_cnt();
    #if TK_LOWPOWER_MODE == ON
```

附录 1-4 函数及变量名对比

附录2 阈值调试

附录2.1 STEP1

例如应用设计的原理图如下，是一个 8 按键的设计。



附录 2-1 原理图对照

根据硬件设计修改按键配置

```

#endif
#define TK_CONFIG_H

/* Normal channel */
#define TK_NUM 8
#define TK_CHANNEL_SEL 0x003001EC

#define TK_CHANNEL_0 TK_2
#define TK_CHANNEL_1 TK_3
#define TK_CHANNEL_2 TK_5
#define TK_CHANNEL_3 TK_6
#define TK_CHANNEL_4 TK_7
#define TK_CHANNEL_5 TK_8
#define TK_CHANNEL_6 TK_20
#define TK_CHANNEL_7 TK_21
    
```

附录 2-2 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

```

tk_config.h
73 #define TK_THD_CHANNEL_0 50
74 #define TK_THD_CHANNEL_1 50
75 #define TK_THD_CHANNEL_2 50
76 #define TK_THD_CHANNEL_3 50
77 #define TK_THD_CHANNEL_4 50
78 #define TK_THD_CHANNEL_5 50
79 #define TK_THD_CHANNEL_6 50
80 #define TK_THD_CHANNEL_7 50
81 #define TK_THD_CHANNEL_8 50
    
```

附录 2-3 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，并设置对应的串口引脚

```
#define TK_UART_FUNC ON /* ON/OFF */
```

```
tk_uart.h
```

```
44 #if TK_UART_FUNC == ON
```

```
45 |
```

```
46 |     #define TK_UART_BAUD 115200
```

```
47 |     #define TK_UART_TYPE TKM_UART
```

```
48 |     // UART
```

```
49 |     #define UART_FUNC GPIO_FUNC_4
```

```
50 |     #define UART_PORT GPIOA
```

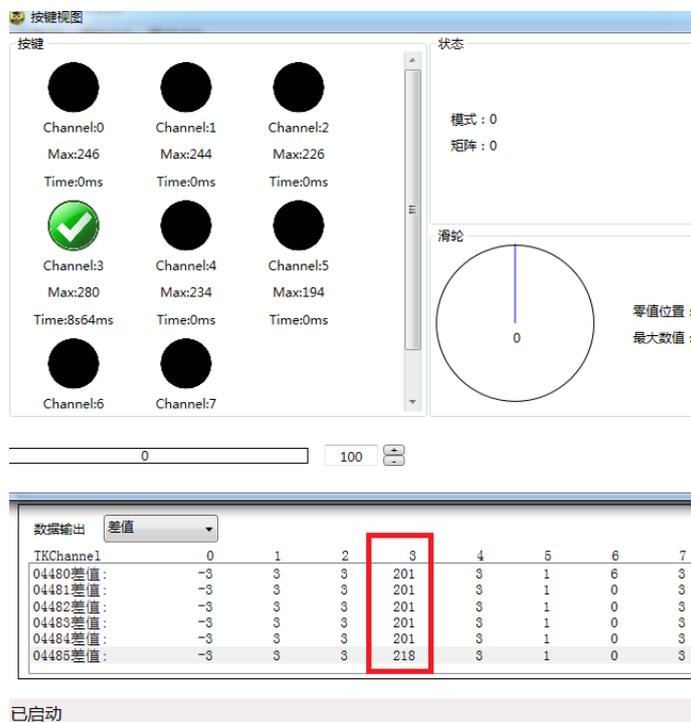
```
51 |     #define UART_TX GPIO_PIN_13
```

```
52 |     #define UART_RX GPIO_PIN_14
```

附录 2-4 调试串口输出

附录2.2 STEP2

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 2-5 上位机调试

若按键差值为 200，则修改门限为调试差值的一半 100。

```

tk_config.h
73 #define TK_THD_CHANNEL_0 100
74 #define TK_THD_CHANNEL_1 100
75 #define TK_THD_CHANNEL_2 100
76 #define TK_THD_CHANNEL_3 100
77 #define TK_THD_CHANNEL_4 100
78 #define TK_THD_CHANNEL_5 100
79 #define TK_THD_CHANNEL_6 100
80 #define TK_THD_CHANNEL_7 100
    
```

附录 2-6 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。