

上海芯旺微电子有限公司	文件编号	版本
		V1.1
研发体系文档中心	生效日期：2021.3.22	页数：共 10 页
标题：KF8A 触摸库应用文档		密级：一般

KF8A 触摸库应用文档

文档作者：	<u>Tim</u>	日期：	<u>2021.3.22</u>
开发 / 测试经理：	<u> </u>	日期：	<u> </u>
总体组：	<u> </u>	日期：	<u> </u>
产品经理：	<u> </u>	日期：	<u> </u>
管 理 办：	<u> </u>	日期：	<u> </u>

修订记录

日期	修订版本	描述	作者
2021.3.22	V1.0	初稿	Tim
2021.4.4	V1.1	增加部分内容	Tim
2021.5.13	V2.0	增加低功耗内容	Tim

目录

1 引言	4
1.1 编写目的与背景	4
2 触摸基础知识	4
2.1 自电容模式简介	4
2.2 常规术语	5
3 触摸库文件结构	6
3.1 常规参数释义及配置	7
3.2 函数释义	9
4 触摸库的使用	9
4.1 正常扫描模式软件设计流程图	10
4.2 低功耗扫描模式软件设计流程图	11
5 参数调试	12

1 引言

1.1 编写目的与背景

本文旨在说明 KF8A 触摸原理简述，触摸参数定义及触摸库的使用。触摸参数如何调整及调试工具 ChipON KF32 TSTool 的使用。

2 触摸基础知识

2.1 自电容模式简介

1、CTOUCH 系统提供了可靠灵敏的电容触摸技术，自电容模式可以用来测量单电极的自电容变化。

2、目前触摸仅支持按键。

3、CTOUCH 系统需要外部参考电容 CX，该外部参考电容于 CAP 引脚，推荐参考电容为 2.2nF~10nF。

4、图 1 为芯片内部触摸原理框图。外部参考电容 CX 为常连接在测量总线的 TOUCHBUS 上，当通道 0 需要进行采样时，其对应的转换单元给 TOUCHBUS 充电（即对 CX 充电），此时定时器开始计数，当 TOUCHBUS 电压达到参考电压时，转换单元停止工作，并将定时器的数据读出，作为当前通道的原始数据采样值（Raw Data）。

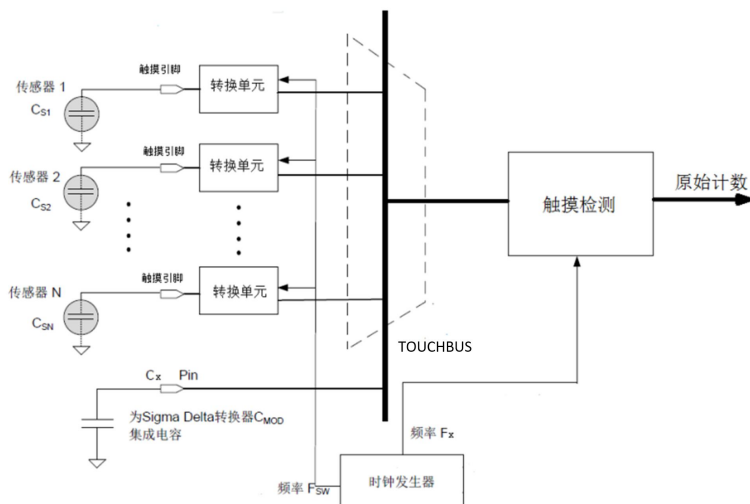


图 1 芯片内部触摸原理框图

5、图 2 为通道采集时电路示意图，时钟发生器控制 CLK1 和 CLK2 成互补不交叠的开关，使得在 CLK1 闭合周期 VDD 给 CS 充电，在 CLK2 闭合周期 CS 给 CX 充电。CLK1 和 CLK2 在这里可以等效为一个电阻 Rs，Rs 公式如下：

$$R_s = \frac{1}{C_s * F_{sw}}$$

其中：

C_s = 传感器电容

F_{sw} = 时钟开关频率

由 R_s 与 C_x 组成的 RC 充电回路，可得 C_x 上的电压 V_{cx} 公式：

$$V_{Cx} = V_{dd} * (1 - e^{-\frac{t}{\tau}})$$

其中：

V_{dd} = 电源电压

$\tau = R_s * C_x$

当 V_{cx} 达到参考电压时停止计数，可得到充电时间 t 公式：

$$t \subset \ln\left(\frac{V_{DD}}{V_{DD} - V_{REF}}\right) * \frac{C_x}{C_s * F_{SW}}$$

其中：

V_{REF} = 触摸内部参考电压

由上式可以看出，在 V_{DD} 、 V_{REF} 、 F_{SW} 一定的情况下，充电时间 t 与 C_x 电容成正比，与 C_s 成反比。故当加大 C_x 电容，可以提高原始数据（信号较小时，可以尝试采用此方式）；有手指按下时，即 C_s 增大（ $C_s = C_p + C_f$ ），充电时间 t 减少（减少的时间 Δt 正比与手指信号值），且 C_f 相对 C_p 变化量较大时， Δt 变化量较大。由此可以看出，减少寄生电容 C_s 有助于提高信号量。

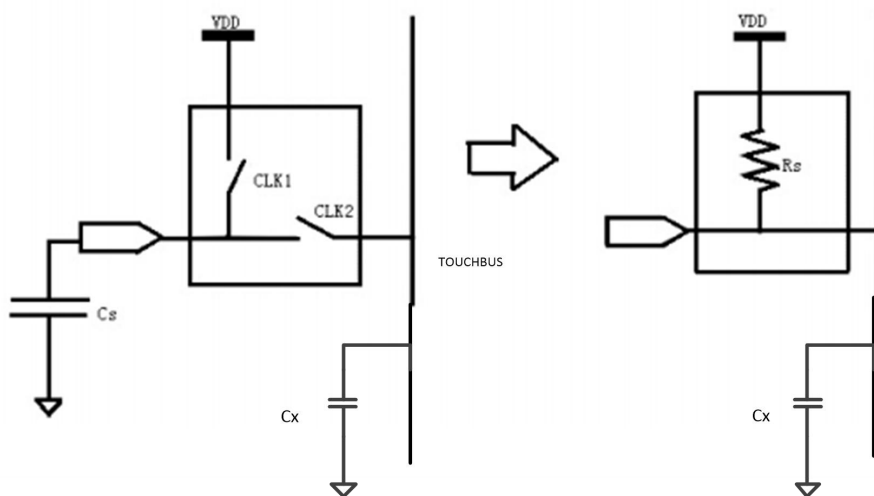


图 2 通道采集电路示意图

2.2 常规术语

按键传感器：用于人体按下的导体，对触摸事件会响应电容变化。

原始计数：按键传感器上直接采集未作处理的数值。

基准线：固件算法计算得到的数值。在外部传感器没有手指按下时，内置的固件算法得到的数值。基准线对原始计数进行处理，并作为按键动作的一个参考点。基准线是一个动态值。

信号：原始计数与基准线间的差值。

信噪比：按键按下状态时的信号值与原始计数值的噪声比值，通常以调整系统软硬件参数，使得信噪比为 5:1 及以上为最优。

手指阈值：与迟滞一起使用，如果信号值高于手指阈值+迟滞，那么按键传感器将被判断为按下状态。如果信号值低于手指阈值-迟滞，那么按键传感器被判断为释放状态。

噪声范围：噪声范围位于噪声与负噪声区间之内，用来参与计算的信号值若在此范围中，则将此数据用于更新基准线。

基准线更新：基本上，基准线就是经过低通滤波后的原始计数值（这里的原始计数值是通过中值滤波得出的）。如果当前原始计数值位于正负稳定区域（黄色线区域）的范围内，则可以进行低通滤波原始计数值以更新基准线。当信号超过正负稳定区域，则进行慢速的更新。

按键传感器状态：按键的状态由信号值、迟滞参数与手指阈值参数一起使用。如下图 3 所示，当信号值大于手指阈值加迟滞，按键将被判断按下；当信号值小于手指阈值减迟滞，按键将被判断为释放。

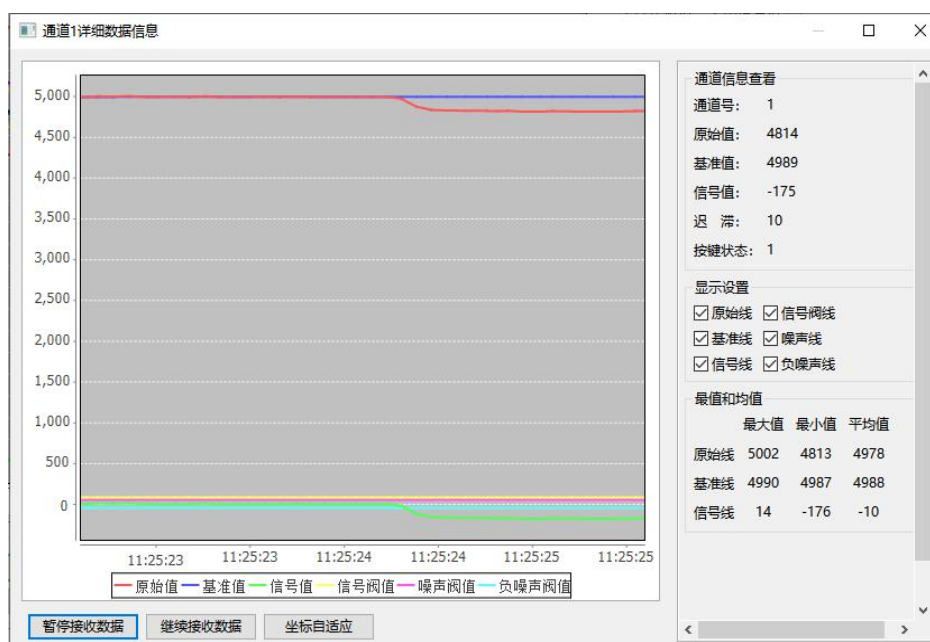


图 3 按键状态判断

3 触摸库文件结构

KF8 触摸库提供三个文件：CTouch.c、CTouch.h 及 kf8a100_touch_driver.o。

kf8a100_touch_driver.o 文件提供触摸底层配置，取值、计算及触摸算法。

CTouch.c 是用户交互的源文件，配置通道及通道阈值等功能。

CTouch.h 是用户交互的头文件，用户程序接口，一些系统参数设置。

3.1 常规参数释义及配置

CTouch_Channel_Tab: 通道设定。该数组元素为通道号，可以通过查看手册确认所支持的通道号，然后添加对应的触摸通道，触摸通道添加可以任意设置（限制于芯片资源），扫描的通道越多，整体扫描的时间越长。数组顺序为扫描顺序，从数组位号 0 开始扫描。

CTouch_Threshold_Tab: 通道信号值阈值设定。该数组元素为各个通道的阈值，与通道设定顺序一一对应。阈值大小建议设置为按下时信号值（信号值=原始计数值-基准线值）的 80%。如为达到更高的灵敏度，可将信号值阈值低至信号值的 60%。

CTouch_Second_Key_Threshold_Tab[]: 通道作为非第二按键按下时增加的信号量，该值为处理按下一个通道时，其他通道如果发生误触发（sensor 离得较近或者其他 sensor 走线在旁边），可以设置此参数。此参数配合 CTouch_Threshold_Tab 使用，为 CTouch_Threshold_Tab 之上增加的信号值作为当前通道（非第一按键触发）的阈值。如果系统不存在此类误触发，则可以将该数组全设置为 0。初次调试时，建议全设置为 0。

CTouch_Single_Hysteresis_Tab: 通道信号迟滞设定。该数组元素为各个通道的迟滞大小，在判断按键的按下及释放时，判断逻辑如下图 5。迟滞大小建议设置为信号值的 10%。

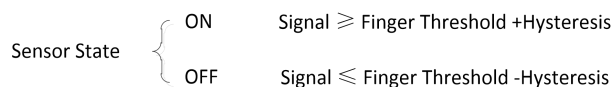


图 4 触摸状态判断条件

CTouch_Noise_threshold_Tab: 通道噪声阈值设定（一般为手指阈值的 50% 左右）。

CTouch_Negative_Noise_threshold_Tab: 通道负噪声阈值设定（一般为手指阈值的 50% 左右）。

CTouch_Max_Threshold_Tab: 最大阈值设定，以人手指最大面积最大力量按下按键时的信号为准（用以抗干扰）。

CTouch_Disturb_Noise_threshold_Tab: 正常情况下，通道异常变化的最大信号（特别注意，这个值要大于手指按下一个通道时，别的通道的最大变化值）。

CTouch_Steady_Noise_threshold_Tab: 稳定噪声线，用于计算一定时间内噪声稳定后，更新基准线。

CTOUCH_VDAC_Tab: 调整系统内部 VDAC 比较电压的阈值（可参考芯片规格书寄存器设定），此参数是为了调整差异通道的采样值。需要注意的是，提高此参数，可一定范围内提高信噪比，但是触摸采样时间会加长。

以上参数的初始设置，可以参考图 5 设置，然后根据实际情况调试更改参数：

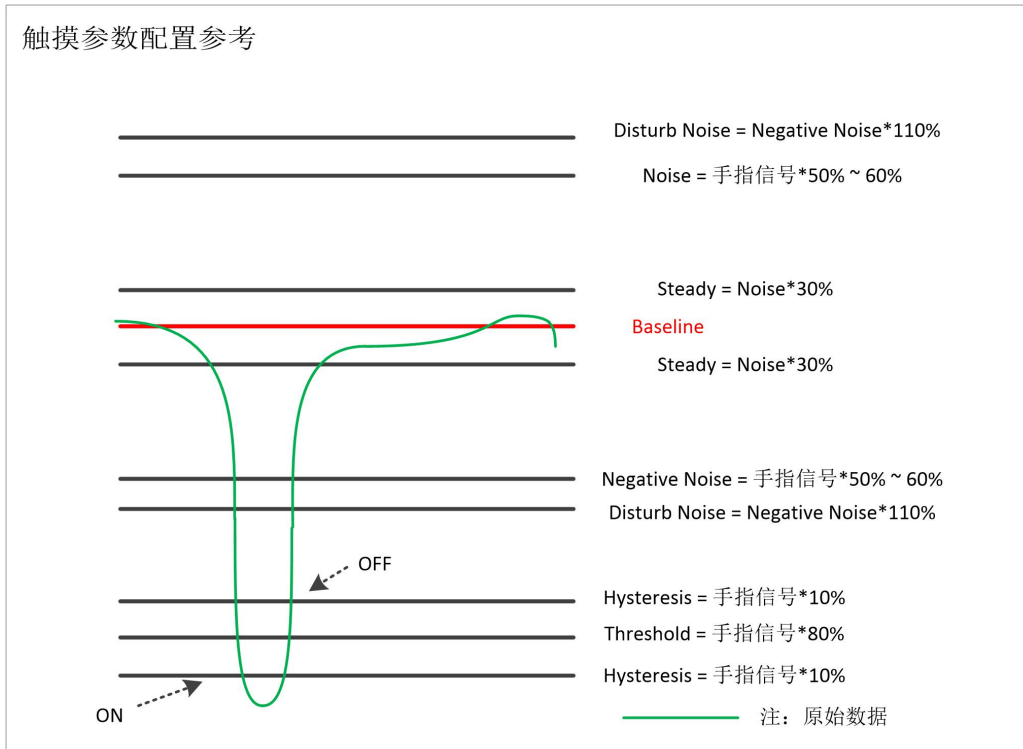


图 5 触摸参数对比

以上参数为 CTouch.c 文件中需要用户手动输入信息。下面简述 CTouch.h 文件中宏参数的释义及配置。

DEF_TOUCH_ON_DEBOUNCE: 按键连续扫描的次数，用于去抖动。去抖动可确保高频率高振幅的噪声不会导致误检。如果检测到误触摸，可以增大去抖动值。否则，可通过降低去抖动值以缩短响应时间。建议设置值 3。

DEF_DATA_STREAM_ENABLE: 使能串口打印（用于 TSTool 调试），要使用功能，必须提前使能对应串口，且在声明 UART_SendBuf(unsigned char data)函数。

DEF_PRESS_ON_MAX_TIME: 最大按下时间参数设置（以扫描圈数为单位）。

DEF_DATA_CHANGE_DISTURB_MAX_CHANNEL: 最大同时存在干扰的通道数量，用于保护系统免于干扰使用，建议设置为 3 及以上。

DEF_PROTECT_MODE_RELEASE_CYCLE: 判断为干扰时，系统退出干扰的时间（以扫描圈数为单位）。

DEF_BASELINE_UPDATE_CNT: 扫描的圈数去更新基准线，越大基准线更新越慢。

DEF_MAX_KEY_CNT: 最大按键数量。（注意如果设置为 2，当有三个手指按下时，系统会进入保护）。

DEF_REFERENCE_CHANNEL_ENABLE: 参考通道使能，如果使能，则 CTouch_Channel_Tab 数组最后一个元素对应的通道为参考通道，且其他数组对应的最后一个元素的参数也为参考通道数据。注意：参考通道设置后，基准线更新相关参数应和其他通道设置方式相似。CTouch_Noise_threshold_Tab 和 CTouch_Negative_Noise_threshold_Tab 会参与参考通道验证保护（原始信号超过此值会保护）；CTouch_Disturb_Noise_threshold_Tab 会参与多通道数据异常保护。

DEF_LOW_POWER_CHANNEL_NUM: 低功耗通道数量。配置方法：如果不使用低功耗通道，设置为 0；如果使用 1 个通道，设置为 1，且 CTouch_Channel_Tab 数组第 1 个

元素为低功耗通道；如果使用 2 个通道，设置为 2，且 CTouch_Channel_Tab 数组前 2 个元素为低功耗通道；以此类推。（注：此参数仅限包含低功耗库，即库版本为 v1.1.x 系列）。

Channel_Press_str: 触摸通道按键状态标志，按照 CTouch_Channel_Tab 所定义通道顺序，如果某一通道按键为按下状态，则 Channel_Press_str 对应的通道的位置 1，否则为 0。

3.2 函数释义

void _CTouch_Key_Init(void): 按键库初始化，调用之前，必须设置相应的系统时钟，触摸扫描时钟等，详见例程库。

unsigned char _CTouch_Scan_Process(void): 调用此函数后，开始扫描通道并处理数据，并返回错误代码。

unsigned int _CTouch_Get_Baseline(unsigned char ch): 返回相应通道的 Baseline 数据。

unsigned int _CTouch_Get_RawData(unsigned char ch): 返回相应通道的 Rawdata 数据。

unsigned int _CTouch_Get_Lib_Version(void): 返回当前库的版本。

unsigned int _CTouch_LowPower_Scan_Process(void): 低功耗扫描函数，调用此函数后，如果扫描到按键被按下，返回对应数据。（注：此函数仅限包含低功耗库，即库版本为 v1.1.x 系列）。

4 触摸库的使用

ChipOn 提供触摸示例工程，请参考例程相应更改。

注意事项：

- 1，触摸通道端口需要设置为输入（包括 Cx 电容端口）。
- 2，如果 Cx 电容使用 IO 接地，则该 IO 需要配置为输出低。
- 3，如果使用低功耗扫描功能，根据外围电路情况，需实际需求配置系统外设模块（包括 GPIO，UART，I2C 等），使得系统平均功耗尽量降低。
- 4，低功耗扫描例程使用的唤醒源是看门狗溢出唤醒，需要根据项目情况，设置合理的看门狗溢出时间。如果不使用看门狗溢出唤醒，也可以使用其他唤醒源，可以参考芯片规格书相应章节。
- 5，库版本为 V1.1.0 以上支持低功耗功能。

4.1 正常扫描模式软件设计流程图

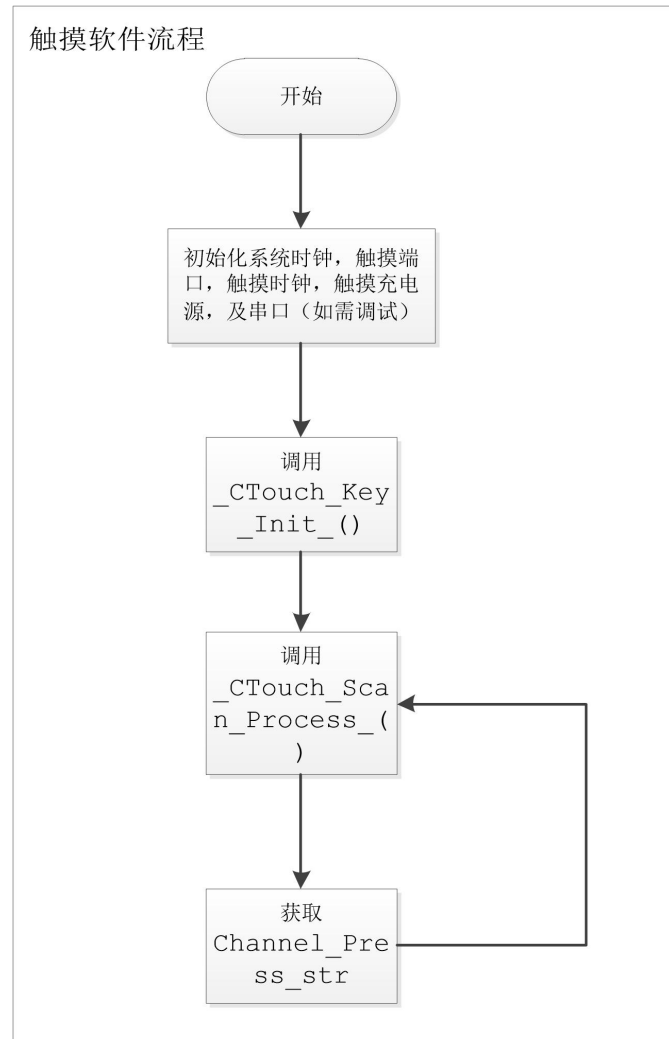


图 6 软件流程图

4.2 低功耗扫描模式软件设计流程图

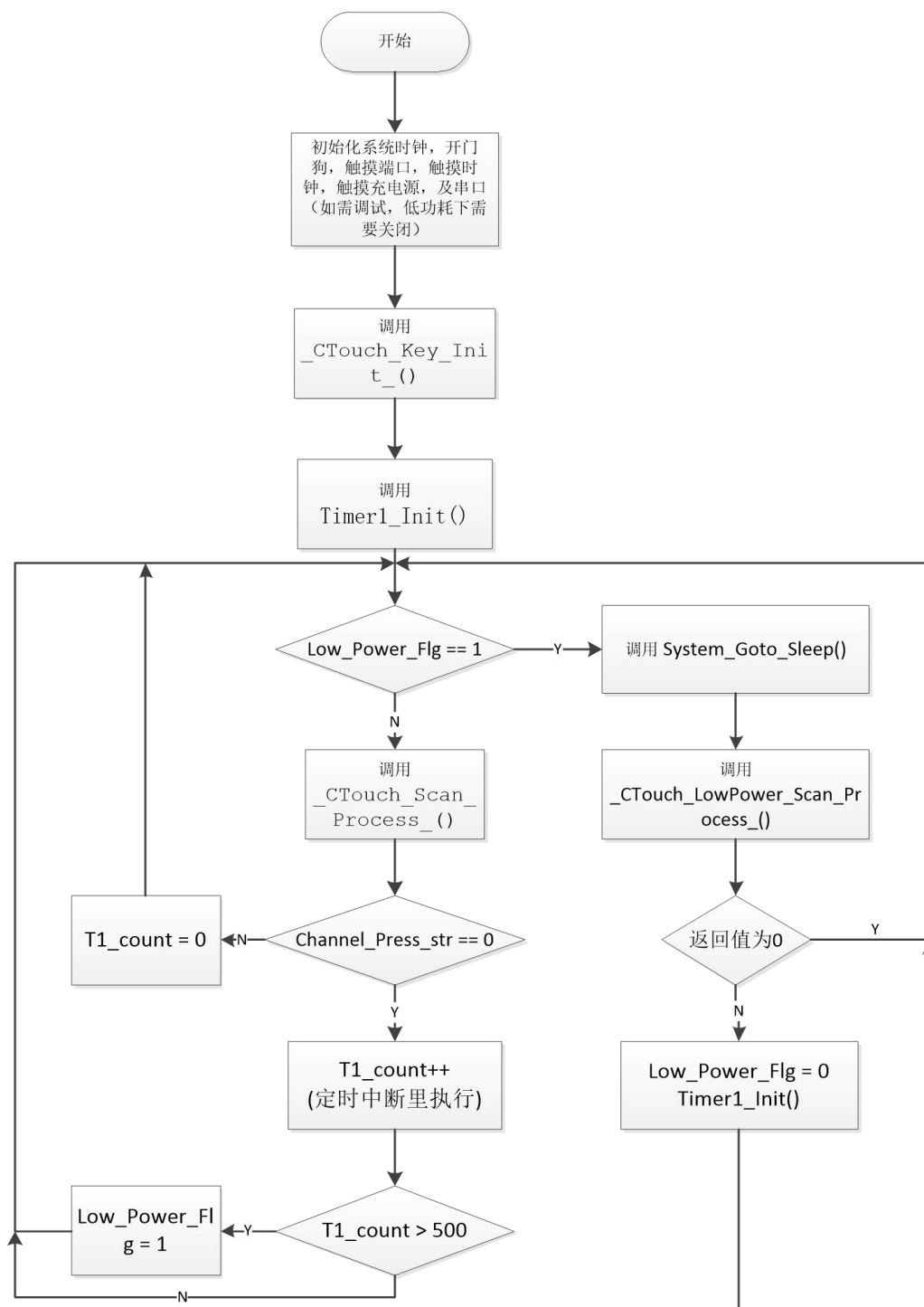


图 7 示例代码低功耗软件流程图

5 参数调试

不同产品，不同按键传感器之间的特性是不一样的，用户可以通过 **ChipON TSTool** 对每个按键进行单独的调试，从而得到按键传感器的最优参数。**ChipON TSTool** 的使用方法请参考软件帮助的 PDF 资料。图 7 以一个通道为例。

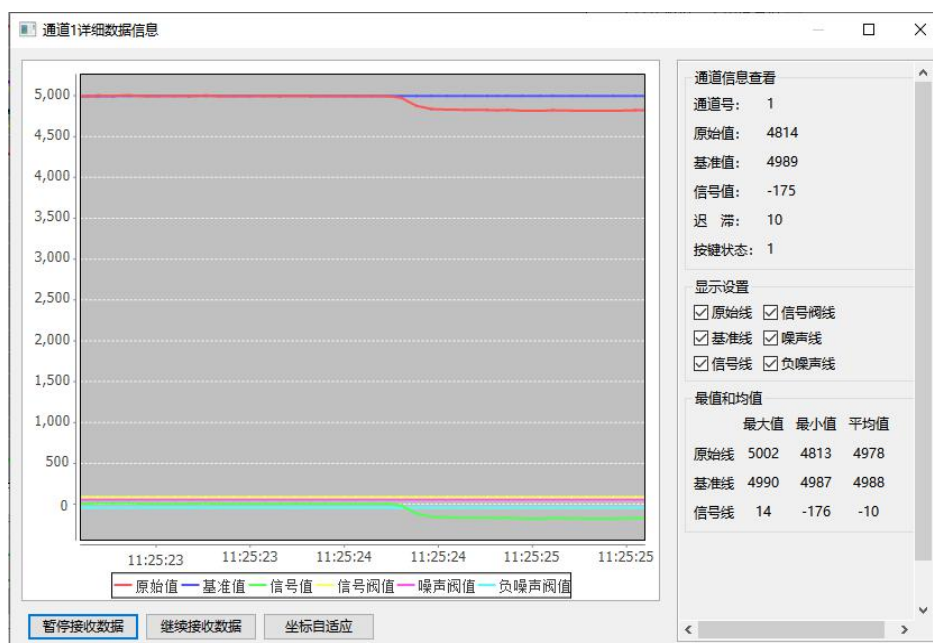


图 7 KF TSTool 界面图