

ChipON 滑轮滑条应用

开发指南

(第二版)

上海芯旺微电子有限公司

2016.03

目录

1	触摸芯片选型	4
2	条件说明	4
3	产品开发设计约束	5
4	软件库使用说明	6
4.1	使用触摸步骤	6
4.2	创建触摸库头文件说明	7
4.3	定义触摸库变量	9
4.4	配置芯片寄存器说明	13
4.5	配置 CTCTL1 寄存器和 VDAC 寄存器说明	14
4.6	调用触摸初始化内部参数函数	16
4.7	调用电容触摸通道处理函数	16
5	触摸库参数及设置说明	17
5.1	MX_CH/ CHS_AMOUNT	17
5.2	SLIDE_CH/ SLIDE_AMOUNT	17
5.3	TCS_AMOUNT/ TCS_SLIDE_AMOUNT:	17
5.4	_KF8_DISTURB_PROTECT_CIRCLE	17
5.5	_KF8_DOUBLEKEY_COMPAGES_LIMITABLE_CIRCLE	18
5.6	_KF8_KEY_MAX_PRESSED_LIMITABLE_CIRCLE	18
5.7	_KF8_BASELINE_UPDATE_CIRCLE_COUNT_FOR_UP_/ DOWN	18
5.8	_KF8_DEFINE_DISTURB_CHANNELS_AMOUNT	18
5.9	_KF8_BASICLINE_UP(DOWN)_NOISE_THRESHOLD	18
5.10	_KF8_INSIDE_REFERENCE_CHANNEL_DISTURB_THRESHOLD_SET	18
5.11	_KF8_INSIDE_REFERENCE_CHANNEL_DECLARE	19
5.12	_KF8_TOUCH_CH_EN[MX_CH]	19
5.13	_SLIDE_DISTRIBUTION[SLIDE_CH]	19
5.14	_SLIDE_RESOLUTION[SLIDE_CH]	19
5.15	_SLIDE_SINGLE_RESOLUTION[SLIDE_CH]	20
5.16	_SLIDE_OFFSET[SLIDE_CH]	20
5.17	_KF8_CONFIG_FINGER_THRESHOLD[MX_CH]	20
5.18	_KF8_CONFIG_FINGER_THRESHOLD_IN_DISTURB_[MX_CH]	22
5.19	_KF8_LIB1_CHANNEL_FLAG	22
5.20	_KF8_LIB1_SLIDE_PRESS_FLAG	22
5.21	_KF8_LIB1_SLIDE_CHANNEL_OUT_[SLIDE_CH]	22
5.22	_KF8_LIB1_SLIDE_DISTANCE_OUT_[SLIDE_CH]	22
附录一	功能引脚和触摸结果标志查询对应关系	23



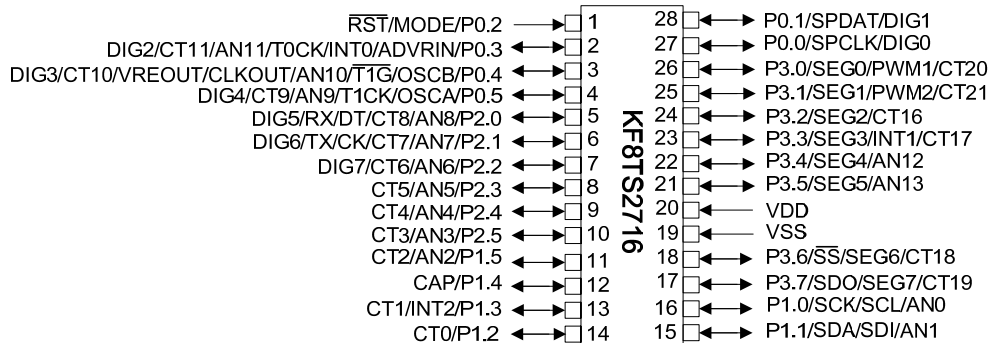
附录二	通道变化率的采集和计算	24
附录三	KF8F2716 滑轮滑条 DEMO 实物图.....	29
附录四	KF8F2716 滑轮滑条 DEMO 功能介绍.....	30

1 触摸芯片选型

目前 ChipON 推出的滑轮滑条触控芯片有 KF8TS25XX、KF8TS27XX 两个系列，该指南介绍针对 8K 的 KF8TS2716 做如下说明。

KF8TS2716 芯片资源如下：

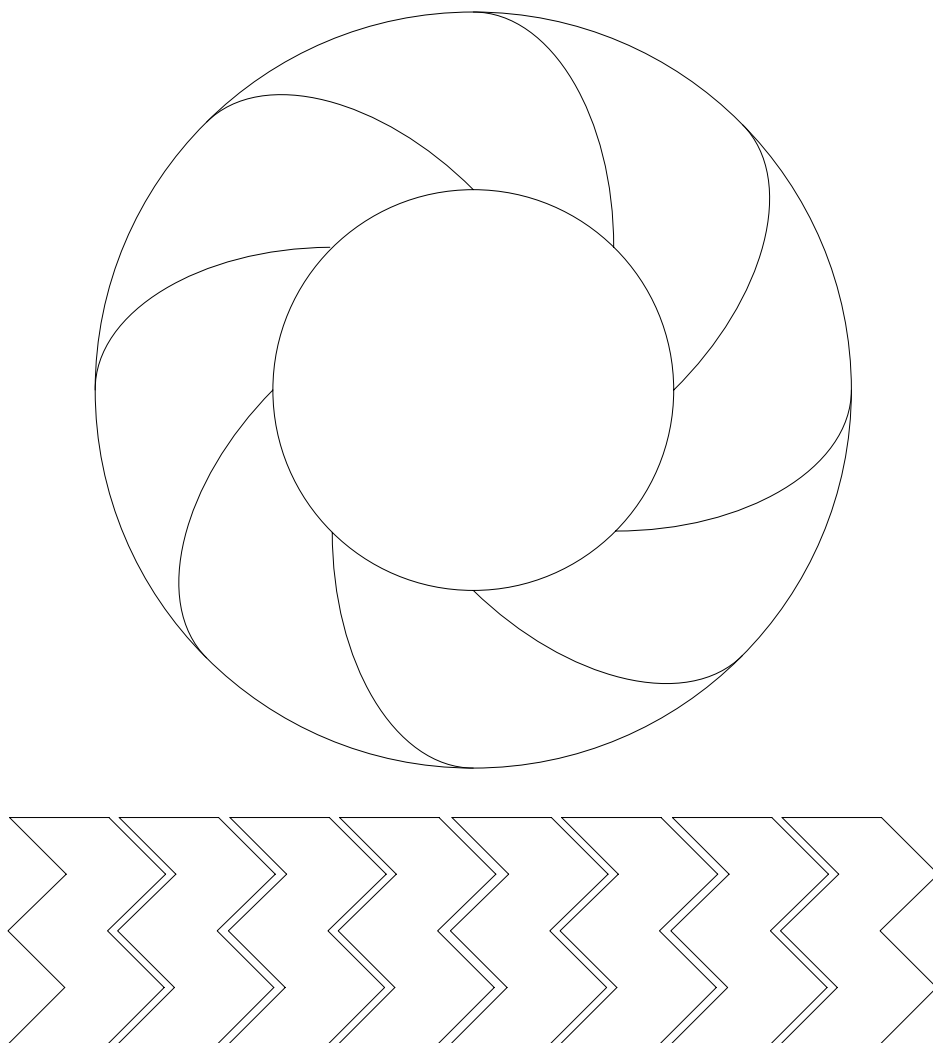
- ✚ 16M 主频，2-5.5V 工作电压
- ✚ 8kFLASH，1040BRAM
- ✚ 32QFP 封装拥有 22 个电容触摸通道
- ✚ 28SOIC 封装拥有 18 个电容触摸通道
- ✚ 24SSOP 封装拥有 14 个电容触摸通道
- ✚ 可以任意分配通道形成最多 4 个滑条/滑轮组，同时支持普通按键功能。
- ✚ 灵敏度高；
- ✚ 抗干扰能力强，具有防电磁等功能；
- ✚ 电容触摸通道可调范围大，用户可根据需要更改外挂电容、分频比和参考电压；
- ✚ 资源丰富，还可实现控制，比如开关信号，温度采样，PWM 控制实现等。



2 条件说明

ChipON 触摸芯片需添加外挂电容，使用 P14 脚，外挂电容的大小影响对采样值有一定影响，可选择 2.2nf 到 44nf 的电容，一般可使用标称 472/103 的电容，具体参数根据产品的实际情况进行更改，可以借助 TS TOOL 软件采集采样值和变化率数据，建议系统采样值调整在 1000-8000 的范围内。触摸设计细节请参考文档《ChipON 电容触摸设计指南》。

3 产品开发设计约束



- 1、滑轮滑条的 pad 样式参照上图设计 pcb，采用 4-8 个触摸 pad 组成一个滑轮或滑条组。
- 2、触摸通道的走线应该设计的尽可能细，如 0.254mm。
- 3、触摸走线最好同其他线隔离，尤其信号线，应避免平行走势。
- 4、滑轮滑条算法中需要参考相邻通道的电容采样值来进行相对位置判定，因此相邻 pad 之间的间隙要小，并且相邻通道交界尽量构成三角状，这样手指在滑动过程中触摸到的通道面积逐渐增大或者逐渐减小，通道采样值是一个逐渐升或者逐渐降的过程，能够提高识别等级的均匀性。
- 5、触摸芯片的工作电源建议稳定可靠，瞬态波动算法可实现滤除，整个时间内的波动可能会造成识别结果不佳等问题。

- 6、为了更好的实现抗干扰能力，芯片电源供应端应设低通滤波器，至少 1 级。
- 7、要求触摸通道上的 1k 滤波电阻靠近芯片引脚。
- 8、产品的通道参数只有整定后才适合自身产品，因此要做阈值整定。
- 9、滑轮滑条整定阈值参数需要参考手指在滑动过程中处于相邻 pad 的交界面时，根据通道的变化率来设定。
- 10、产品建议预留或焊接 ICSP 端口或端点，便于调试维护和在线升级功能。
- 11、Cap 脚的外接电容建议采用容值偏差小，温度波动范围小的电容。电容值容值范围可选择 1nF~10nF 之间，要求使用 10%或以上精度的涤纶电容、X7R 材质电容或 NP0 材质贴片电容。
- 12、滑轮滑条一般采用 PCB 上面直接覆盖板，为了使 pad 紧贴盖板，可以采用粘性材料避免空气间隙导致触摸变化率过小。

4 软件库使用说明

4.1 使用触摸步骤

开发触摸按键系统需要通过以下 6 个步骤完成软件平台的搭建（或者直接使用 ChipON 提供的范例程序进行修改）：

① 创建触摸库头文件

需要在 main.h 中声明相应芯片的头文件和触摸库的头文件，触摸库头文件为

“kf8ts27_touch_s.h”

② 定义触摸库变量

在项目中创建触摸库参数源文件，文件中的参数值需要用户根据产品的特性进行修改，见文件 **“kf8ts27_touch_s.c”**

③ 配置芯片功能寄存器

在“main.c”中根据需要对 MCU 进行初始化，包括初始化 IO 口和芯片功能，具体见 3.4 节。

④ 配置触摸 CTCTL1 和 VDAC 功能寄存器。

在“main.c”的主函数中进行配置，根据芯片手册 CTCTL1 寄存器设置电容触摸时钟分频比和电容触摸基准电压，时钟分频比越大，通道采样值越大，相应的采样速度变慢。

KF8TS2716 可以使用 VDD 或者内部 LDO 作为基准电压，根据不同产品应用，可做适当选择。如不需要使用 AD 等功能时，内部 LDO 的选择更能适应电源变化或波动，具体参数含义可阅读芯片手册，一般参数采用默认 0x41 即可。VDAC 寄存器用于设置电容触摸的比较器翻转电压，翻转电压相对基准电压的比例越高，通道采样值越大。一般设置为 0x90 即可，当选择内部 LDO 时，VDAC 建议分压源也选择 LDO。

⑤ 调用触摸模块初始化函数

调用触摸初始化函数对触摸参数进行初始化，用户只需在主函数中直接调用下面的函数即可，无需修改其内容参数。

滑轮滑条触摸初始化函数：“_KF8TS27_Init_Touch_Slide_W_()”

⑥ 调用触摸库函数

在“main.c”中调用触摸库处理函数。此函数实现触摸按键识别功能。

滑轮滑条库函数名称为“_KF8TS27_Touch_Process_Slide_W_()”

4.2 创建触摸库头文件说明

滑轮滑条库“kf8ts27_touch_s.h”文件如下：

```
/*
*****
* 文件名:kf8ts27_touch_s.h
* 版 本:   V2.0
* 日 期:   2015-12-20
* 作 者:   上海芯旺微电子有限公司
* 说明:     电容触摸库函数头文件
*****
*/

#ifndef KF8TS27_TOUCH_W_H_
#define KF8TS27_TOUCH_W_H_

#include "main.h"

#define MX_CH  17 // 所有开通触摸通道数的个数
#define SLIDE_CH 2 // 滑条组数
/*
*****
*****
*/
//声明给触摸库使用
/*
*****
*****
*/
extern unsigned char const CHS_AMOUNT; //最大通道号
extern unsigned char const SLIDE_AMOUNT; //最大滑条组数
extern unsigned char const TCS_AMOUNT; //普通单键按下延时
```

```
extern unsigned char const TCS_Slide_AMOUNT; //滑条组识别延时
extern unsigned int const _KF8_Disturb_Protect_Circle_; //异常保护时间
extern unsigned int const
    _KF8_DoubleKey_Compages_Limitable_Circle_; //允许双键进入时间限定
extern unsigned int const
    _KF8_Key_Max_Pressed_Limitable_Circle_; //最长按键有效时间
extern signed char const
    _KF8_BaseLine_Update_Circle_Count_For_UP_; //向上更新计数到采样IIR
滤波
extern signed char const
    _KF8_BaseLine_Update_Circle_Count_For_Down_; //向下更新计数

//同时变化通道数属于异变,主要用于向下,最小值2,最大值根据通道数量
extern signed char const _KF8_Define_Disturb_Channels_Amount_;

//量化定义向上/向下变化率多于x属于突变,突变计数,多于一定数量属于异变
extern signed char const _KF8_BasicLine_UP_Noise_Threshold_;
extern signed char const _KF8_BasicLine_Down_Noise_Threshold_;
extern signed char const _KF8_Data_Steady_Threshold_Set_; //上下次波动
//量化参考通道变化率多于x属于电源波动
extern signed char const
    _KF8_Inside_Reference_Channel_Disturb_Threshold_Set_; //仅用于参考通道

extern unsigned long _KF8_LIBi_channel_flag_; //普通单键标志,按下置
对应bit为1
extern unsigned char _KF8_LIBi_Slide_Press_Flag_; //滑条组标志,按下置对应
bit为1
extern unsigned int _KF8_LIBi_Slide_channel_out_[SLIDE_CH]; //滑条组 输出
当前分辨率
extern signed int _KF8_LIBi_Slide_distance_out_[SLIDE_CH]; //滑条组 输出
滑动距离
extern unsigned char _KF8_LIBc_channel_; //采集了新数据的通道号
extern unsigned char const _KF8_Inside_Reference_Channel_Declare_; //引入
基准通道的通道号值
extern unsigned char const _KF8_TOUCH_CH_EN[MX_CH]; //逻辑化通道扫描的
顺序
extern unsigned char const _SLIDE_DISTRIBUTION[SLIDE_CH]; //滑条通道分组
extern unsigned int const _SLIDE_RESOLUTION[SLIDE_CH]; //滑条最大输出分辨
率设置
extern unsigned char const _SLIDE_SINGLE_RESOLUTION[SLIDE_CH]; //滑条单
通道承担分辨率设置
extern signed int const _KF8_Config_Finger_Threshold[MX_CH]; //手指阈
值设置
```



```
extern signed int const
    _KF8_Config_Finger_Threshold_In_Disturb_[MX_CH]; //扰动下手指阈值设置
extern signed int      _KF8_LIBi_Date_Change_[MX_CH]; //当前
变化率
extern signed char     _KF8_LIBc_baseline_count_[MX_CH]; //用作
基准线计数值
extern unsigned int    _KF8_LIBi_buff_hit_[MX_CH]; //保存采样值
extern unsigned int    _KF8_LIBi_buff_baseline_[MX_CH]; //保存基准线
extern unsigned char   _KF8_LIBc_touch_count_[MX_CH]; //按键识别计
数, 按键滤波
extern unsigned char   _KF8_lib_Slide_Finded_MAX_I_T27_SW_[SLIDE_CH]; //
滑条组变化率最大通道号
extern signed int      _KF8_lib_Slide_Finded_MAX_I_Buf_T27_SW_[SLIDE_CH]; //
滑条组变化率最大备份
extern signed char     _KF8_LIBi_Slide_round_count_[SLIDE_CH]; //滑条组滑过圈
数统计
extern unsigned int    _KF8_LIBi_Slide_channel_back_[SLIDE_CH]; //滑条组初
始值备份
/*****
//声明给触摸库使用结束
/*****
/*****
//触摸库函数声明
/*****
unsigned char _KF8TS27_Get_Ver_Slide_W(); //获取当前库的版本号
void          _KF8TS27_Init_Touch_Slide_W(); //初始化电容触摸
void          _KF8TS27_Touch_Process_Slide_W(); //电容触摸通道处理
void          _KF8TS27_Touch_TEST_Slide_W(); //简化接收开启函数, 仅调
试曲线时用
#endif /* KF8_LIB_TOUCH_H_ */
```

4.3 定义触摸库变量

定义如下变量, 变量含义及参数设定原则详见第 5 章说明。

滑轮滑条库 “kf8ts27_touch_s.c”, 文件的参数需与头文件一致, 如下所示:

```
*****
* 文件名: kf8ts27_touch_s.c
* 版本:   v2.0
* 日期:   2015-12-20
* 作者:   上海芯旺微电子有限公司
* 说明:   电容触摸库变量资源
```

```
***** /
#include "main.h"
/***** /
/***** /
/以下为触摸算法所用到的通用变量数据，数组的元素个数必须与所开通的通道数一致！！/
signed char _KF8_LIBc_baseline_count_[MX_CH]; //用作补偿基准线，
满足条件的一定数量后采样IIR滤波方式更新
unsigned int _KF8_LIBi_buff_hit_[MX_CH]; //记录当前通道的采样值
unsigned int _KF8_LIBi_buff_baseline_[MX_CH]; //当前通道的基准值
unsigned char _KF8_LIBc_touch_count_[MX_CH]; //记录当前按键的判定时
次数，满足TCS_AMOUNT时置按下标志
signed int _KF8_LIBi_Date_Change_[MX_CH]; //按键采样变化量
级，正数表示低于，负数表示高于
/以下为滑轮滑条算法所用到的通用变量数据，数组的元素个数必须与滑条组数的通道数一致 /
unsigned char _KF8_lib_Slide_Finded_MAX_I_T27_SW_[SLIDE_CH]; //滑条组变
化率最大通道号
signed int _KF8_lib_Slide_Finded_MAX_I_Buf_T27_SW_[SLIDE_CH]; //滑
条组变化率最大备份
signed char _KF8_LIBi_Slide_round_count_[SLIDE_CH]; //滑条组滑过圈数统计
unsigned int _KF8_LIBi_Slide_channel_back_[SLIDE_CH]; //滑条组初始值备份
//当前处理通道位置
unsigned char _KF8_LIBc_channel_;
//最大通道数，达到该值时，下次通道切换选择第一个映射的通道号
unsigned char const CHS_AMOUNT = MX_CH;
//最大滑条组数，达到该值时，下次判断普通按键
unsigned char const SLIDE_AMOUNT = SLIDE_CH;
//通道按键按下滤波0到255可设，仅满足最大手指阀，且计数到该数值时判定按键被按下
unsigned char const TCS_AMOUNT = 4;
//滑条按键按下滤波0到255可设，仅满足最大手指阀，且计数到该数值时判定按键被按下
unsigned char const TCS_Slide_AMOUNT = 4; //4
//异常识别起始在接下来的0到65535个扫描周期特殊处理，多通道一定幅度下降认为水，识别
动作后的周期内不会进行按键判断。
unsigned int const _KF8_Disturb_Protect_Circle_=100;

//双键允许时间设定，有效参数11-65535，当为0时，限定时间概念失效
/**效果说明：如首个按键识别后，在接下来时间内允许组合双键，超过时间仅识别首个按键(单
键)**/
/**设置说明：这里的数值为扫描循环周期，示例：如800,约时间(800乘(通道数4)乘(单通
道处理时间1ms)=3秒**/
unsigned int const _KF8_DoubleKey_Compares_Limitable_Circle_=0;

//按键最长有效时间设定，有效参数1-65535，当为0时，最长时间概念失效
/**效果说明：如按键最长时间为35秒，超过的计为异常，超时的清除标志，失效时无时长概念
```

```
*/
/**设置说明：这里的数值为扫描循环周期，示例：如8000,约时间(8000乘(通道数4)乘(单
通道处理时间1ms)=32秒*/
unsigned int const _KF8_Key_Max_Pressed_Limitable_Circle_=0;

//更新基准线快慢参数，值越小更新的约快，正负0-125可设，建议为按键有效计数的正数倍
signed char const _KF8_BaseLine_Update_Circle_Count_For_UP_=120;    //
正方向计数到采样IIR滤波
signed char const _KF8_BaseLine_Update_Circle_Count_For_Down_=-120;
//负方向计数到采样IIR滤波，对等地位

//同时变化通道数属于异变，主要用于向下,最小值2，最大值根据通道数量
signed char const _KF8_Define_Disturb_Channels_Amount_=5;

//量化定义向上/向下变化率多于x属于突变，突变计数，多于一定数量属于异变
signed char const _KF8_BasicLine_UP_Noise_Threshold_=-30;
signed char const _KF8_BasicLine_Down_Noise_Threshold_=30;
signed char const _KF8_Data_Steady_Threshold_Set_=15; //上下次波动限定，
低于属于趋势稳定

//量化参考通道变化率多于x属于电源波动
signed char const
_KF8_Inside_Reference_Channel_Disturb_Threshold_Set_=5; //仅用于参考通道
unsigned char const _KF8_Inside_Reference_Channel_Declare_=25; //内容值
为参考通道，不使用时赋予一个不存在的值即可

//使用的通道数据及通道号，这里需和芯片对应，如CT3写3，不同芯片的内部参考通道可能不
同，需要_KF8_Inside_Reference_Channel_Declare_定义，如这为CT10
unsigned char const _KF8_TOUCH_CH_EN[MX_CH]={
    8,  7,  6,  5,          //对应第一组滑条的通道排列顺序

    18, 17, 16, 20,
    11, 1,  0,  19,        //对应第二组滑条的通道排列顺序

    3,  14, 15, 13,
    12,                    //普通单键排列顺序

    4,                    //空

};

//滑条组通道分配情况
unsigned char const _SLIDE_DISTRIBUTION[SLIDE_CH]={
```

```
4, //对应第一组滑条的通道个数
8, //对应第二组滑轮的通道个数
};

//滑条组输出最大分辨率设置
unsigned int const _SLIDE_RESOLUTION[SLIDE_CH]={
    16, //第一组滑条输出0-15级
    256, //第二组滑轮输出0-255级
};

//滑条组单通道均摊分辨率设置，这里需要区分滑轮与滑条的概念。滑条由于头尾不相接，相当于头尾各少半个按键。
//滑条设置：
_SLIDE_SINGLE_RESOLUTION[i]=_SLIDE_RESOLUTION[i]/(_SLIDE_DISTRIBUTION[i]-1)
//滑轮设置：
_SLIDE_SINGLE_RESOLUTION[i]=_SLIDE_RESOLUTION[i]/_SLIDE_DISTRIBUTION[i]
//最终值需要输出实际分辨率范围后敲定，一般在上述计算结果上加一或者减一即可
unsigned char const _SLIDE_SINGLE_RESOLUTION[SLIDE_CH]={
    5, //第一组滑条均摊16/3
    33, //第二组滑轮均摊256/8
};

//滑条组偏置设置，这里需要区分滑轮与滑条的概念。滑条由于头部少半个键需要减去偏置
//滑条设置：_SLIDE_OFFSET[i]=_SLIDE_SINGLE_RESOLUTION[i]/2
//滑轮设置：
_SLIDE_OFFSET[i]=(_SLIDE_SINGLE_RESOLUTION[i]*_SLIDE_DISTRIBUTION[i]-_SLIDE_SINGLE_RESOLUTION[i])/2
//最终值需要输出实际分辨率范围后敲定，一般在上述计算结果上加一或者减一即可
unsigned char const _SLIDE_OFFSET[SLIDE_CH]={
    2, //第一组滑条偏置
    2, //第二组滑轮偏置
};

/*;;理想条件变化量从小到大依次分为：小范围波动，噪声阀，来水阀，手指阀;;
* 即：超过噪声阀通道数量满足大于等于3个，判断按键存在水，来水后变化率
* 应该比手指阀小从而水不会产生按键误识别，手指阀，延时滤波直接关系到灵敏度
* 防水建议不应按键太过灵敏，建议1秒可识别2次的速度
*/

/**这里调整正常模式下按下需要幅度，针对滑轮滑条，阈值不需要太多，否则滑动过程中会中断*/
signed int const _KF8_Config_Finger_Threshold[MX_CH]={
    30,    30,    30,    30, //对应第一组滑条的手指阈值
```

```
30,      30,      30,      30,
30,      30,      30,      30, //对应第二组滑条的手指阈值

40,      50,      50,      50, //对应普通单键的手指阈值
50,
35,

};

/*多个通道的有幅度下降, 记为异常, 异常的通道识别后使用该阈值, 相对来说该数值比手指
阀设置要大,

如果干扰不需要识别, 可将阈值设定到一个按键无法到达的幅度, 即不识别, 当不考虑该能够
能也可以和手指阀设置一样*/
signed int const _KF8_Config_Finger_Threshold_In_Disturb_[MX_CH]={
                                500,      500,      500,      500,
                                500,      500,      500,      500,
                                500,      500,      500,      500,
                                500,      500,      500,      500,
                                500,      500,
};

unsigned long _KF8_LIBi_channel_flag_; //对外提供按键信息
unsigned char _KF8_LIBi_Slide_Press_Flag; //滑条组标志, 按下置对应bit为1
unsigned int _KF8_LIBi_Slide_channel_out_[SLIDE_CH]; //滑动组 输出当前分
辨率
signed int _KF8_LIBi_Slide_distance_out_[SLIDE_CH]; //滑动组 输出滑动距离
/*****
//触摸库使用资源结束
/*****
/*****引用按键标志*****/
/*****触摸通道配置结束*****/
```

4.4 配置芯片寄存器说明

- 1、参考晶振初始化, 如 $OSCCTL = 0x60$; 该数值对应 8M 晶振, 晶振频率不宜选择太低, 该结果会影响触摸反应速度。
- 2、端口初始化
 - ◎ 将电容触摸通道对应的 I/O 口设置为输入, 即给 TRx 方向寄存器的对应位置 1。
 - ◎ 外接电容端口需设置成输入口, 这里 P1.4 口为外接电容引脚
- 3、中断寄存器配置
 - ◎ 开启高低优先级中断使能位 IPEN, 两个中断入口不同。当芯片选择工作在 T4 计数模式时, 需要将电容触摸中断设置为高优先级, 即 PT3=1, 中断标志使用 T3IF。

芯片程序范例如下:

```
void init_mcu()
{
    //晶振频率选择
    OSCCTL = 0x60; //max 0x70 16m

    //端口初始化
    TR0 = 0x3C; //00111100
    TR1 = 0xDC; //11011100
    TR2 = 0xFF; //11111111
    TR3 = 0xCD; //11001101

    P0LR=0x00;
    P1LR=0x00;
    P2LR=0x00;
    P3LR=0x00;

    P0=0x00;
    P1=0x00;
    P2=0x00;
    P3=0x00;

    T0CS = 0;
    T0IE = 1;
    T0 = 0x20; //125us为一个时间分度 非触摸需要
    IPEN=1;
    PT3 = 1; // 这里选择触摸为高优先级
    AIEL = 1;
    AIEH = 1;
}
```

4.5 配置 CTCTL1寄存器和VDAC寄存器说明

寄存器“CTCTL1”是 KF8TS2716 设置触摸分频比和基准电压源的寄存器,用户通过设置 CTCTL1 的值调节对外挂电容充电的频率并选择电容充电的电压源。分频比越大,则频率越低,通道采样值则越大。

寄存器9.2: CTCTL1: 控制寄存器(地址: 64H)

复位值 00-- ----0	bit7						bit0	
	CTCLKSE L1	CTCLKSE L0	-	-	-	-	-	VSEL
	R/W	R/W	U	U	U	U	U	R/W

CTCLKSEL<1:0>: 电容触摸时钟预分频比选择位

00 = SCLK/4

01 = SCLK/8

10 = SCLK/16

11 = SCLK/32

VSEL: 输入电容充电电压源选择位

0 = 输入电容充电电压源为 V_{REF}

1 = 输入电容充电电压源为 VDD

如 CTCTL1 = 0x41; 则 设置触摸时钟为 8 分频, 基准电压源为 VDD, 这里分频源为系统时钟, 即内部时钟 8M 时, 触摸时钟为 1M, 一般触摸时钟设定为 1M 或 2M 时最佳。

寄存器“VDAC”是 KF8TS2716 设置触摸比较器翻转电压的寄存器, 用户通过设置 VDAC 的值调节电压阈值, 电压阈值越大, 通道采样值则越大。

寄存器9.3: VDAC: 电阻分压比设置寄存器(地址: 112H)

复位值 0000 ----	bit7				bit0			
	VDAC3	VDAC2	VDAC1	VDAC0	-	-	-	-
	R/W	R/W	R/W	R/W	U	U	U	U

VDAC<3:0> :电阻分压比设置位

0000 = 0.05VDAC

0001 = 0.15 VDAC

0010 = 0.15 VDAC

0011 = 0.2 VDAC

0100 = 0.25 VDAC

0101 = 0.3 VDAC

0110 = 0.35 VDAC

0111 = 0.4 VDAC

1000 = 0.45 VDAC

1001 = 0.5 VDAC

1010 = 0.55 VDAC

1011 = 0.6VDAC

1100 = 0.65 VDAC

1101 = 0.7VDAC

1110 = 0.75 VDAC

1111 = 0.8 VDAC

如 VDAC = 0x90; 则比较器翻转电压为 0.5 倍的基准电压源。

4.6 调用触摸初始化内部参数函数

触摸初始化函数主要是对触摸算法的内部参数以及部分寄存器进行初始化,如配置触摸使能、定时器工作设定等。因开机时采样并计算参数数据。在进行初始化前建议作适当延时,避免上电波动使基准线数据不准确而带来异常,参考 200ms。示例如下:

```
init_mcu();  
delay_ms(200);    //随后开始功能的运行
```

4.7 调用电容触摸通道处理函数

电容触摸按键处理函数能够对触摸按键进行手指按下与否的判断,可以在中断内、外中执行,滑轮滑条库的返回值为有4个:

- 1、**unsigned long** _KF8_LIBi_channel_flag_;//普通单键标志,按下置对应bit为1
- 2、**unsigned char** _KF8_LIBi_Slide_Press_Flag_;//滑条组标志,按下置对应bit为1
- 3、**unsigned int** _KF8_LIBi_Slide_channel_out_[SLIDE_CH]);//滑条组,输出当前分辨率
- 4、**signed int** _KF8_LIBi_Slide_distance_out_[SLIDE_CH];//滑条组,输出连续滑过距离

_KF8_LIBi_channel_flag_的每个 bit 位对应相应的通道,对应关系见附录 1。当有通道被触摸时,_KF8_LIBi_channel_flag_的对应 bit 位被置 1,无触摸时对应 bit 位置 0。电容触摸中断标志位 TS 系列芯片为 T3IF。每次进中断时判断,如果 T3IF 为 1 时调用电容触摸通道处理函数,还可以在不影响中断内部处理时间的条件下,设置一个标志位,然后再在主程序中调用电容触摸通道处理函数。

```
void INT_FUN() __interrupt (0)  
{  
    if(T3IF)  
    {  
        T3IF=0;  
        T4CTL=0x00; // T4同步计时模式  
        kf8_touch_flag=1;  
        _KF8TS27_Touch_Process_Slide_W_();    // 在中断中执行触摸函数  
    }  
}
```

也可以在中断中做一个标志后退出,然后在 main.c 中如下处理

```
if(kf8_touch_flag)    // 判断标志位  
{  
    kf8_touch_flag = 0;    // 清标志位  
    _KF8TS27_Touch_Process_Slide_W_(); // 在主函数中执行触摸函数  
}
```


5 触摸库参数及设置说明

通过上一章节的七个步骤后，触摸按键产品的触摸功能配置完成。但在当前产品上能否运行正确还需要对相应的参数进行调试。

名词解释：

1、 扫描周期时间：即程序对所有按键扫描并处理一遍所使用的时间。文档的关于时间的概念设定值均为周期数。如设置为100，可以想象如果每个按键占用1ms时间，7个按键就是7ms，100个周期则代表着700ms，具体每个按键的扫描时间与频率选择和电容的大小等相关。

2、 通道变化率：通道变化率是有符号数，正数表示采样值在基准线下方，负数表示采样值在基准线上方。关于此参数的说明和计算方式详见参考附录二。

5.1 MX_CH/ CHS_AMOUNT

MX_CH参数设定产品触摸通道数，在程序上采用宏定义的方法，可在kf8ts27_touch_s.h中修改。CHS_AMOUNT参数同样为使用的通道数据，该数值应用于通道循环扫描判断，赋值为MX_CH。

5.2 SLIDE_CH/ SLIDE_AMOUNT

SLIDE_CH参数设定产品滑条组个数，在程序上采用宏定义的方法，可在kf8ts27_touch_s.h中修改，该数值用于设定滑条相关数组的长度，赋值为SLIDE_CH。
SLIDE_AMOUNT参数同样为使用的滑条组个数，赋值为SLIDE_CH。

5.3 TCS_AMOUNT/ TCS_SLIDE_AMOUNT

TCS_AMOUNT为普通单键识别周期参数，即当通道变化率大于手指阈值时每一周期计数加一，当计数值与该参数相等时按键有效，电容触摸处理函数返回有效的按键结果。数值越大，判断一个结果用的时间越长。TCS_SLIDE_AMOUNT为滑条识别周期参数，在同一个pad上停留一定时间才能被识别，数值越大，快速滑动越不灵敏。参数设置和灵敏度有关，参数越小，越容易被识别，建议至少为3。

5.4 _KF8_Disturb_Protect_Circle_

此参数实现异常状态的保护周期，可以设定异常状态的保护时长，在保护期间内，不识别按键判断。值设置为200时，即为200个扫描周期，该参数有效范

围为 0-65535，根据实际情况设置保护周期的大小。

时间与扫描周期关系：比如假定单个按键的执行时间为 1ms。根据按键（包括基准通道）的数量计数出一个扫描周期使用的时间，单个按键的扫描时间与频率选择、电容大小等相关。

5.5 _KF8_DoubleKey_Compares_Limitable_Circle_

此参数设定双按键间隔时间，即按下一键后再按下二键之间允许的最长等待时间，超过时间后则不再识别第二按键。当为 0 时，最长时间概念失效。该值对应为扫描周期数。

5.6 _KF8_Key_Max_Pressed_Limitable_Circle_

此参数设定按键按下持续最长有效时间，超过时间后按键自动释放。有效参数 1-65535，当为 0 时，最长时间概念失效。同 5.5。

5.7 _KF8_BaseLine_Update_Circle_Count_For_UP_/DOWN

该参数决定基准线更新的快慢，当参考通道数据稳定且满足当前设定计数时长时调用基准线更新算法。该数据同样以扫描周期为单位。可以根据扫描周期的长短灵活设定该值的大小。参考数据为 -120/120，上限为 -128/127。

5.8 _KF8_Define_Disturb_Channels_Amount_

该参数设定关系到触摸库受到干扰保护机制，此参数配合 5.9 使用。当多于此参数的通道个数发生采样值异变时触发通道干扰保护，通道不识别。一般设置为大于 3，保护时间见 5.4。

5.9 _KF8_BasicLine_UP(Down)_Noise_Threshold_

该参数设定为通道采样值发生异变的阈值范围，超过该参数设定范围则认为通道发生异变，计数异常通道个数，配合 5.4、5.8 使用，判断是否触发保护。一般设计要求信噪比满足 1:5，系统噪声小时该值可以设小，可以参考按键按下量的 20-30% 考虑。

5.10 _KF8_Inside_Reference_Channel_Disturb_Threshold_Set_

该参数设定为参考通道异常阈值。5.11 指定参考通道，如果该通道的变化率

大于参数设定值则认为存在波动，同时触发保护通道保护机制，异常时间内不做触摸识别。该参数适合不可触碰的悬空通道或内部专用通道，通道声明见 5.11。

5.11 _KF8_Inside_Reference_Channel_Declare_

参考通道号设置，如果不用参考通道作为保护依据，则需设置为一个 _KF8_TOUCH_CH_EN[MX_CH] 中不存在的值。如果使用参考通道，则需要对应通道悬空并设置为输入口或内部专用通道，不使用时可设置数值 50。

5.12 _KF8_TOUCH_CH_EN[MX_CH]

定义当前产品所需开通的通道号。通道号是芯片数据手册给出的通道号，并且需要按照第一组滑条（滑轮）+第二组滑条（滑轮）+……第N组滑条（滑轮）+普通单键的顺序排列，且滑条（滑轮）组内的通道号需要按滑条（滑轮）pad 位置按顺序排列。为了解释数组的含义，举一个例子来说明以下滑条数组的设置。例如：某调光项目需要使用1组5通道滑条实现0-99级输出、1组4通道滑轮0-29级输出、3个普通按键。

其中第一组滑条使用通道顺序为CT6、CT5、CT4、CT3、CT2

其中第二组滑轮使用通道顺序为CT11、CT9、CT0、CT1

其中普通单键顺序为CT15、CT12、CT8

则数组元素定义为

_KF8_TOUCH_CH_EN[MX_CH]={6,5,4,3,2,11,9,0,1,15,12,8};

5.13 _SLIDE_DISTRIBUTION[SLIDE_CH]

数组定义了滑条（滑轮）各组的通道个数，上述例子可以设置 _SLIDE_DISTRIBUTION[SLIDE_CH]={5,4};滑条库会根据 _SLIDE_DISTRIBUTION[SLIDE_CH]滑条分配情况和 _KF8_TOUCH_CH_EN[MX_CH]通道排列情况以及MX_CH总通道个数的设定来自动判断各通道的组合情况。

5.14 _SLIDE_RESOLUTION[SLIDE_CH]

数组定义了滑条各组输出的分辨率等级个数。根据需求定义。如上述例子要

求第一组输出 0-99 共 100 级，第二组输出 0-29 共 30 级。则数组元素定义为
_SLIDE_RESOLUTION[SLIDE_CH]={100,30,};

5.15 _SLIDE_SINGLE_RESOLUTION[SLIDE_CH]

数组定义了滑条各组均摊到单个通道的分辨率等级数，此处设定需要区分滑轮与滑条的概念。滑轮与滑条的区别特性在于：滑轮头尾相接，滑条则头尾分离，滑条算法中需要涉及到当前通道与左右两个相邻通道的采样值来判定相对按下位置，而显然滑条头尾分离后，头尾两个按键不能找到与之相邻的左边或右边通道，所以在按下头部通道（默认左边第一通道为头部）时，算法只能识别到靠近右边通道的距离信息，而无法判断靠近左边通道的距离信息。因此头部通道只能识别一半的信息，以此类推尾部通道也只能识别一半。于是算法中滑条通道数=实际通道数-1。而滑轮的算法通道数=实际通道数。

上述例子可知第一组为滑条均摊为 $100/(5-1)=25$ ，第二组均摊为 $30/4=7.5$ 。第二组不能整除，为了输出等级的完整性，这里采用进一法，7.5 需要设置为 8。数组元素定义为 _SLIDE_SINGLE_RESOLUTION[SLIDE_CH]={25,8,};

该参数的调整需要根据实际情况来确定，调用
_KF8_LIBi_Slide_channel_out_[i] 输出采样值，根据实际滑动后输出范围来适当进行微调。

5.16 _SLIDE_OFFSET[SLIDE_CH]

数组定义了滑条各组的输出偏置值。最终输出值=采样值-偏置值。即针对滑条组输出范围进行修正，初始情况下可以先把该参数设置为 0。等确定了 _SLIDE_SINGLE_RESOLUTION[SLIDE_CH] 参数后再来设置此参数，该参数设定一般需要配合 _SLIDE_SINGLE_RESOLUTION[SLIDE_CH] 设定。

滑条设置：_SLIDE_OFFSET[i]=_SLIDE_SINGLE_RESOLUTION[i]/2

滑轮设置：

_SLIDE_OFFSET[i]=(_SLIDE_SINGLE_RESOLUTION[i]*_SLIDE_DISTRIBUTION[i]-_SLIDE_SINGLE_RESOLUTION[i])/2

滑条最终输出结果为 _KF8_LIBi_Slide_channel_out_[i]=

_SLIDE_SINGLE_RESOLUTION[i]*X-_SLIDE_OFFSET[i]，其中 X 表示传感器输出值（例如手指在 pad3 和 pad4 之间，根据触摸坐标 X 输出如 3.3）。对于一个给定的系统 X 的输出范围是固定不变的，当需要调节输出范围时可以改变倍率和偏置。微调 _SLIDE_SINGLE_RESOLUTION[i]、_SLIDE_OFFSET[i] 参数可以调整输出范围上述例子中数组元素可以定义为 _SLIDE_OFFSET[SLIDE_CH]={12,1}。

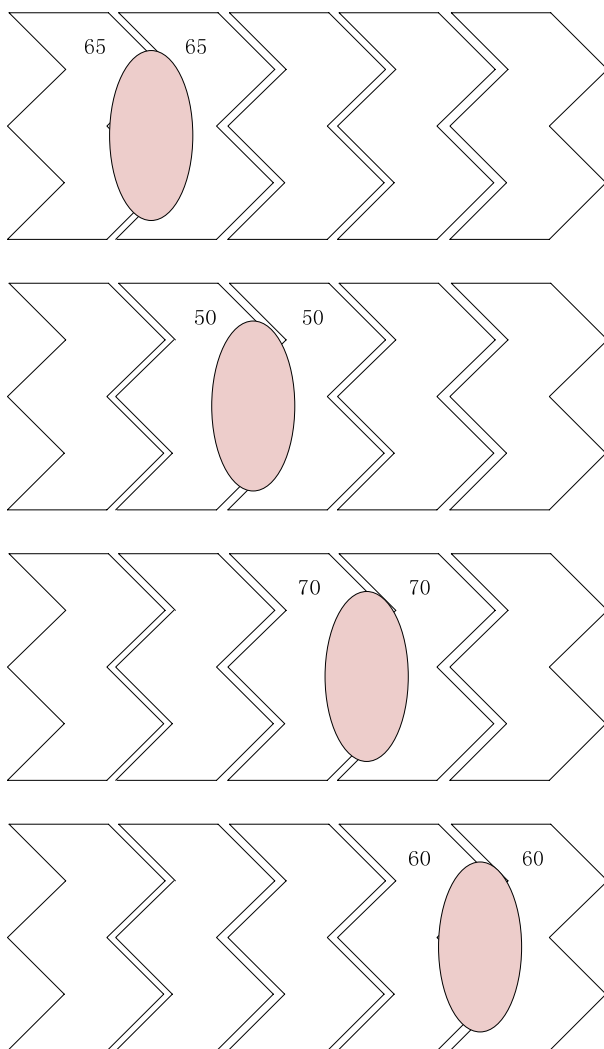
5.17 _KF8_Config_Finger_Threshold[MX_CH]

无水手指阈值。此参数数组的每一个元素对应一个按键通道，在通道变化率大于对应的手指阈值并且累加到 TCS_AMOUNT 或 TCS_SLIDE_AMOUNT 规定

的扫描周期数量，则被判定为手指识别。其数组排列顺序对照
_KF8_TOUCH_CH_EN[MX_CH]。

滑条组设置：同一滑条组建议设置成同一阈值，建议阈值设定为滑条组滑动过程中，到达各个相邻pad交界处后调整至相邻pad的变化率基本相等，收集这一组所有变化率得出最小值，这一组都设定为该最小值的60%微调即可。其考虑的原理是在滑动过程中必须有一个pad的变化率大于手指阈值，而手指在pad交界处所读取的变化率则是最不容易识别的，于是手指阈值必须小于该值。

上述例子中，利用ChipON TSTool获取的相邻变化率情况如下：椭圆为手指按下位置，数值为两通道变化率相等时的数值。



第一组滑条得到的变化率情况是65、50、70、60.取最小值 $50 \times 60\% = 30$ ，则

第一组滑条的阈值设置为

`_KF8_Config_Finger_Threshold[MX_CH]={30,30,30,30,30,……}`

普通单键设置：利用ChipON TSTool获取单个按键手指按下的最大变化率情况，该按键的手指阈值设置为最大变化率的60%即可，根据灵敏度可以进行调整。

5.18 _KF8_Config_Finger_Threshold_In_Disturb_[MX_CH]

异动保护下识别手指阈值，当触摸功能进入保护时，发生突变的通道进入保护模式，随后按键识别阈值采用该数值判断，作用同 5.17。针对滑轮滑条，不做保护处理，因此对应通道的阈值可设置为如 1000，即保护状态不做识别处理。

该参数具有一定的防水作用，即多个通道被按下（或波动），对应的通道阈值使用该阈值，如果不需要识别，可以将阈值设定为一个很高的不能到达的水平，或者使用 5.17 的阈值，也可以采用最大手指变化量加一个数值上去，不如 10。

5.19 _KF8_LIBi_channel_flag_

`_KF8_LIBi_channel_flag_` 输出为普通单键的输出键值，见附录一

5.20 _KF8_LIBi_Slide_Press_Flag

`KF8_LIBi_Slide_Press_Flag` 按位输出滑条组按下情况，如果第一组有按下则该寄存器 0 位置 1，如果第二组有按下则 1 位置 1，以此类推。

5.21 _KF8_LIBi_Slide_channel_out_[SLIDE_CH]

`_KF8_LIBi_Slide_channel_out_[SLIDE_CH]` 数组输出各个滑条组的当前分辨率。第一组结果输出在 `_KF8_LIBi_Slide_channel_out_[0]`，第一组结果输出在 `_KF8_LIBi_Slide_channel_out_[1]`，以此类推。

5.22 _KF8_LIBi_Slide_distance_out_[SLIDE_CH]

`_KF8_LIBi_Slide_distance_out_[SLIDE_CH]` 数组输出各个滑条组连续划过长度。第一组结果输出在 `_KF8_LIBi_Slide_distance_out_[0]`，第一组结果输出在 `_KF8_LIBi_Slide_distance_out_[1]`，以此类推。

附录一 功能引脚和触摸结果标志查询对应关系

触摸通道 使能配置	按下时 _KF8_LIBi_channel_flag_对应值	对应位置	芯片 引脚
CT0	0000 0000 0000 0001	bit0	P12
CT1	0000 0000 0000 0010	bit 1	P13
CT2	0000 0000 0000 0100	bit 2	P15
CT3	0000 0000 0000 1000	bit 3	P25
CT4	0000 0000 0001 0000	bit 4	P24
CT5	0000 0000 0010 0000	bit 5	P23
CT6	0000 0000 0100 0000	bit 6	P22
CT7	0000 0000 1000 0000	bit 7	P21
CT8	0000 0001 0000 0000	bit 8	P20
CT9	0000 0010 0000 0000	bit 9	P05
CT10	0000 0100 0000 0000	bit 10	P04
CT11	0000 1000 0000 0000	bit 11	P03
CT12	0001 0000 0000 0000	bit 12	P16
CT13	0010 0000 0000 0000	bit 13	P17
CT14	0100 0000 0000 0000	bit 14	P26
CT15	1000 0000 0000 0000	bit 15	P27

如果用户选择使用通道 P20，P05。对应到表中就是 CT8 和 CT9,则需给 _KF8_TOUCH_CH_EN 数组元素赋上 8 和 9，顺序可调换，即当调换顺序时对输出值 _KF8_LIBi_channel_flag 是不会造成影响的，但如果是滑轮滑条中的需要按照顺序书写。

变量_KF8_LIBi_channel_flag的某位为1说明相应通道按下，如果为0则说明没有按下，通道8和通道9分别按下时，变量_KF8_LIBi_channel_flag的第9位和第10位分别为1（从第1位开始，最低位代表CT0的识别结果）。

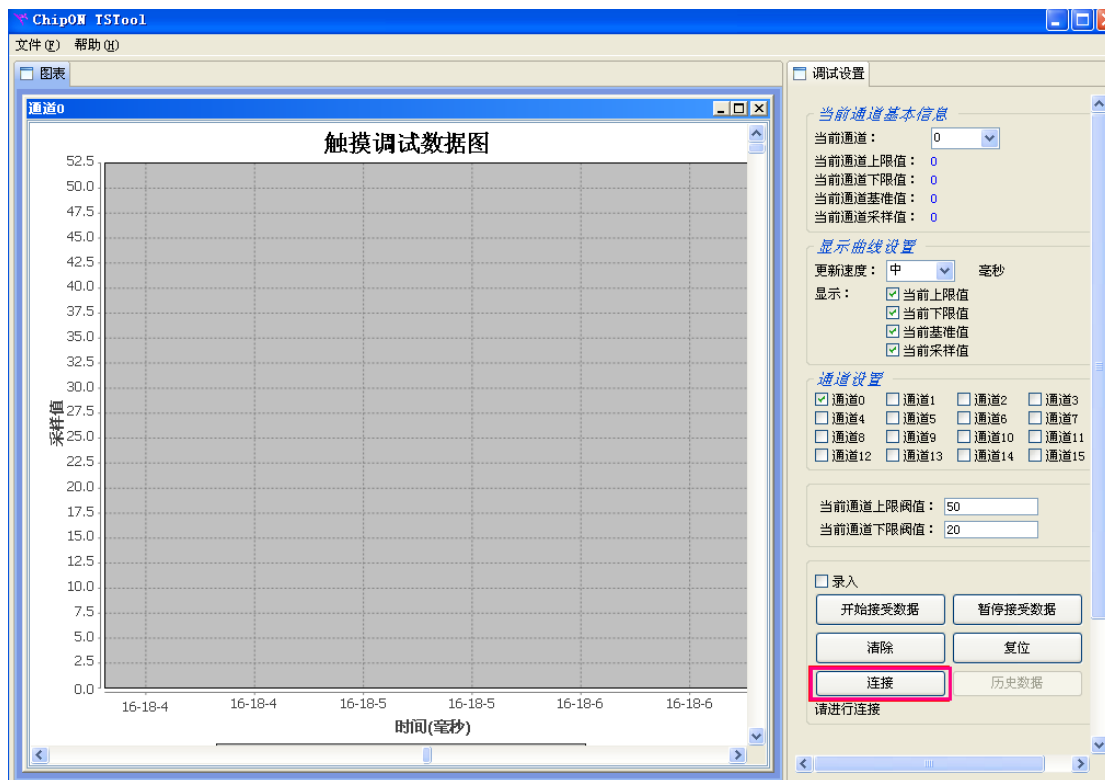
附录二 通道变化率的采集和计算

通道变化率的采集，我们使用到一个 ChipON TSTool 上位机软件，通过此软件，用户可以直观地看到各个通道的采样值、基准线，使用之前必须处理调试相关函数的使能，debug_touch.c 提供了 2 个函数接口，其中 TOUCH_DEBUG_TRS_DEAL 负责完成曲线输出的使能，TOUCH_DEBUG_SEND_OUT 可输出 4 个 byte 数据，可通过串口实现产品代码调试监控功能。使用之前还要在 debug_touch.h 文件中设定调试用的 2 个 IO 端口，要求为双向口，默认使用编程接口。

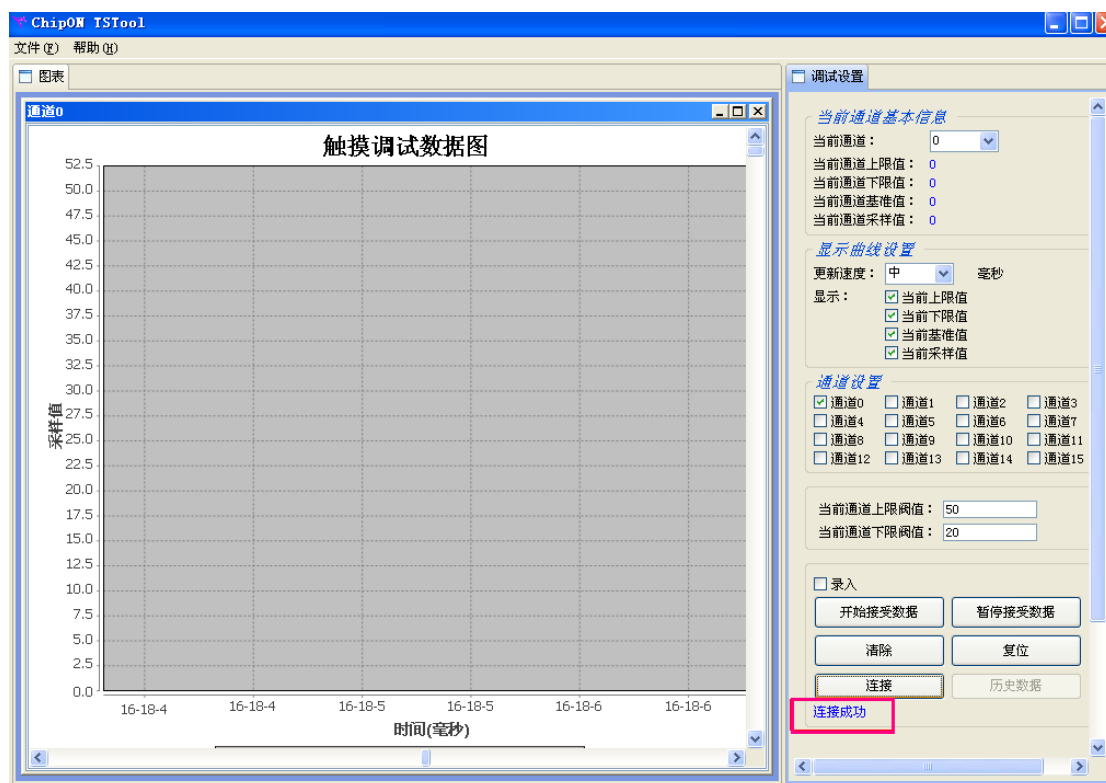
下面介绍如何使用此工具软件。

步骤 1、连接好编程器和产品板，下载触摸库程序完毕。

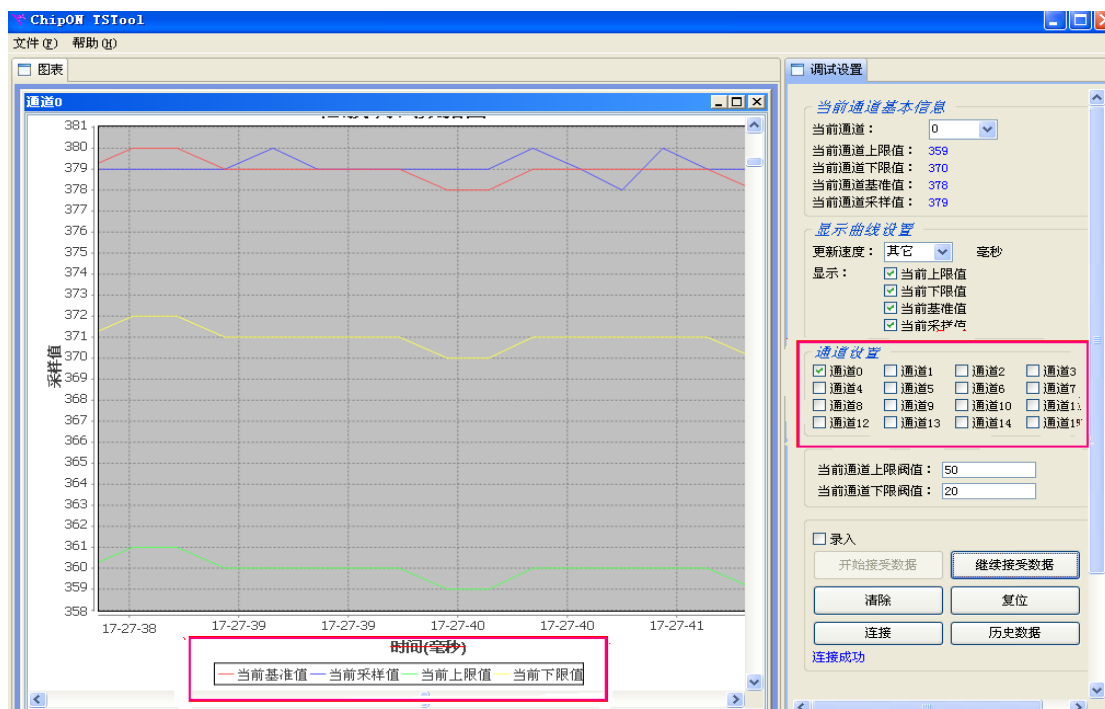
步骤 2、保持编程器和产品板之间的连接，打开 ChipON TSTool 上位机软件，并点击红色区域的【连接】按钮，打开后如下：



步骤 3、连接成功后如下，红色区域提示连接成功。



步骤 4、通过点击【通道设置】中相应的通道并点击【开始接受数据】按钮，在数据图区域则可以看到选定通道的采样值和基准线，选择未开通的通道软件不能正常输出曲线，会报数据接收异常，如果只开启了 CT8, CT9，选择 8, 9 以外的通道均报异常。点击【暂停接受数据】按钮则暂停接受数据，点击【清除】按钮则清除当前数据图的数据，并根据数据自动适应坐标，将显示区分最大化。点击【复位】按钮则上位机复位，点击【复位】按钮后若要重新接受数据，则需回到步骤 1 重新开始操作。通道选择应该和单片机使用序号一致。

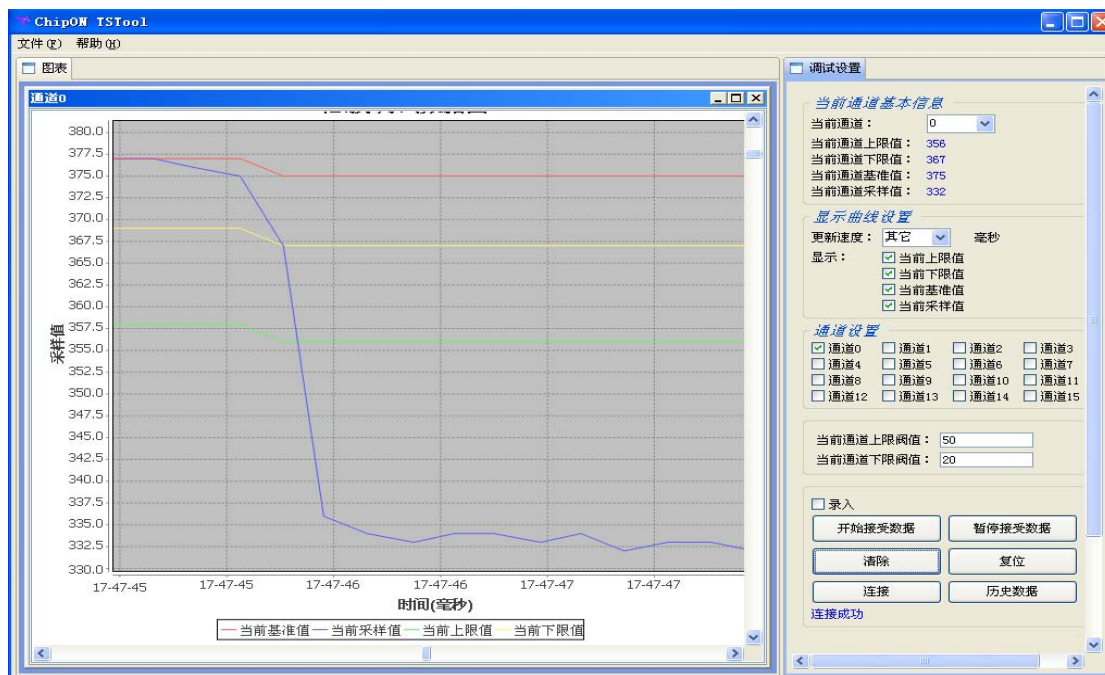


如上图是采集了通道 0 的采样值（蓝色）和基准线（红色），，上位机的通道号和_KF8_TOUCH_CH_EN 数组元素是对应的，即上位机的通道 0 对应数组_KF8_TOUCH_CH_EN 元素值为 0 的通道，芯片的 CT0，上位机的通道 1 对应数组_KF8_TOUCH_CH_EN 元素值为 1 的通道，芯片的 CT1，以此类推。

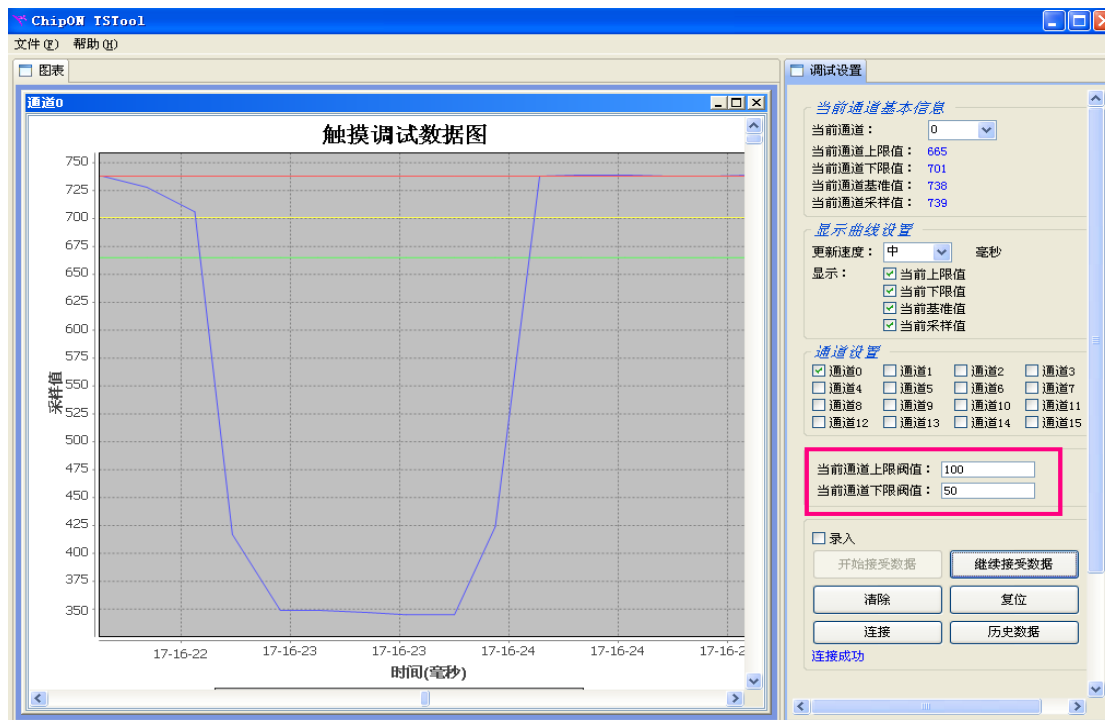
```
unsigned char const _KF8_TOUCH_CH_EN [MX_CH]={
    11,      10,
    9,       8,
    5,       4,
    3,       2,
};
```

如上的_KF8_TOUCH_CH_EN 设定，上位机的通道 10 则分别对应数组_KF8_TOUCH_CH_EN 元素的 10，访问下标为 1。

步骤 5、计算通道变化率，当通道 0 按下时，从上位机上可看出采样值（蓝色）出现突然下降，按下后取采样值为 332，基准值为 375，则 变化率 = (375-332) / 375 = 0.114，变化率中我们使用千分比进行比较，所以通道变化率为 114，通道变化率为有符号数，正数表示采样值在基准线下方（按下/触水），负数表示采样值在基准线上方。



步骤 6 设置当前通道上限阈值和当前通道下限阈值，这两个值是为了方便用户可以直观地看到通道按下和释放时通道变化率的范围，用户可以根据不同产品的通道变化率来进行设定。如图：



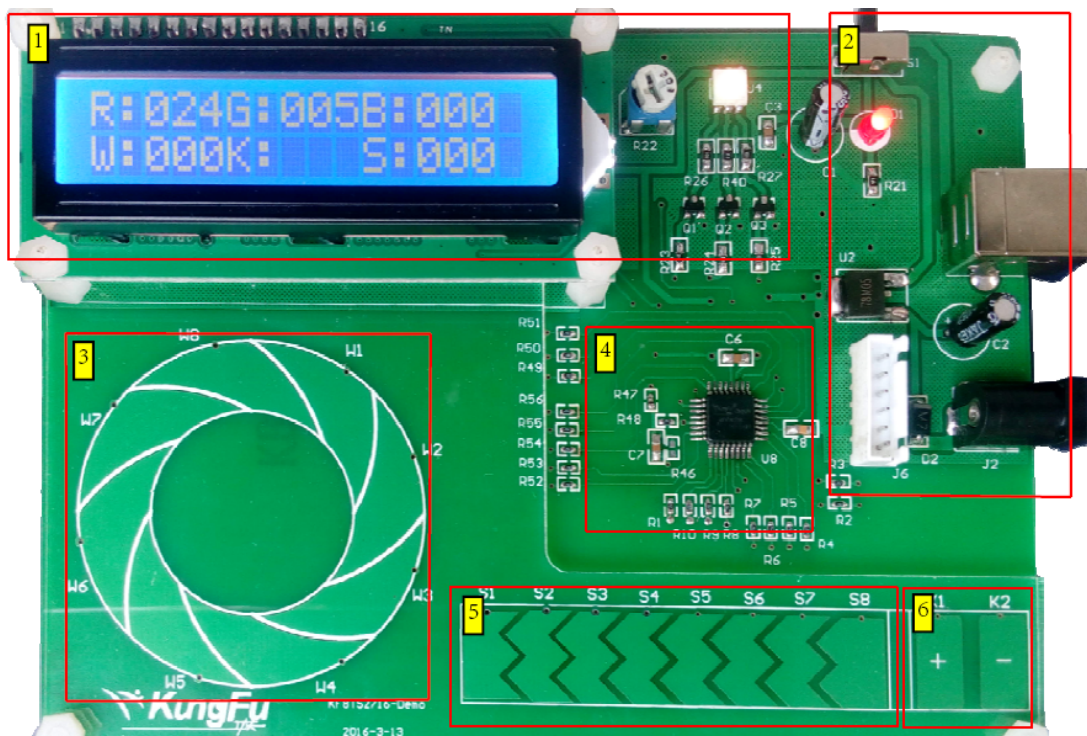
当前通道上限值：为了比较直观的看到当通道按下时采样值与基准值之间差值的千分比

的范围，我们设定了当前通道上限阈值，当采样值曲线在上限值曲线之下时，我们判定为按下操作。如上图，当前通道上限阈值设定为 100，我们可以看到上限值为 663（浅绿色），基准值为 737（蓝色）， $(737 - 663) / 737 = 0.1$ ，即千分之 100。

当前通道下限值：同理，为了比较直观地判别按键是否处于释放状态，我们设定了当前通道下限阈值，当采样值曲线在下限值曲线之上时，我们判定为释放状态，如上图，当前通道下限阈值设定为 50，我们可以看到下限值为 700（黄色），基准值为 737（蓝色）， $(737 - 367) / 375 = 0.05$ ，即接近千分之 50。

实际过程中仅需要针对单个通道，查看计数满足是否满足要求，根据按下的数据变化幅度同默认的 20、50 的参数进行对比。通过估算的方式修改 20、50 的数值，使对应的数据线同蓝线处于一个水平即可，这时填入的数值即为该通道按下带来的变化量数据。该变化量单位为千分率。

附录三 KF8F2716 滑轮滑条 DEMO 实物图



1. 1602 液晶屏及 RGB 指示灯
2. 电源输入模块及 ICSP 下载调试接口
3. 8pads 滑轮触摸面板
4. KF8F2716 主控
5. 8pads 滑条触摸面板
6. 2pads 按键触摸面板

附录四 KF8F2716 滑轮滑条 DEMO 功能介绍

1. 液晶屏第一行内容 R:XXXG:XXXB:XXX,数值为 RGB 指示灯的各自的 PWM 占空比, 上限为 255。
2. 8pads 滑轮输出 0—99 级分辨率, 显示在液晶屏第二行 W:000; 同时根据滑轮划过距离控制 Red LED 的亮度, 即顺时针亮度增加, 逆时针减小。
3. 8pads 滑条输出 0—99 级分辨率, 显示在液晶屏第二行 S:000; 同时根据滑条划过距离控制 Blue LED 的亮度, 即从左向右增加, 从右向左减小。
4. 2pads 按键输出显示在液晶屏第二行 K: , 按下加号键显示 “+”, 按下减号键显示 “-”, 同时按下不予识别。按键控制 Green LED 的亮度, 支持长按加速功能, 即长按时亮度数值增减会越来越快。