

1. 简介

1.1. 总述

支持 CH57x、CH58x、CH59X、CH32V00X、CH32L103、CH32V20X、CH32V30X 系列 MCU，共有滤波模式 3、CS10。根据需求可在以下滤波模式中选择一种进行采样数据处理，并输出触摸键值。

滤波器模式	功能	备注
CS10	可以通过 IEC 61000-4-6 标准的抗扰度三级测试，即 CS10V 动态测试	目前支持单按键识别，后续升级支持双按键。轮询函数中有最大 8us 的随机阻塞式延迟。
3	轮询函数需要定时调用，调用频率根据实际应用对触摸的响应速度需求确定； 可休眠部分通道，适合对功耗要求较高的场合； 支持多按键和单按键。	单按键模式下，多个键同时按下后，仅识别键值最大的键， 轮询函数阻塞运行，扫描触摸通道越多时间越长，最多在 400us 左右。

1.2. 库使用流程

如图 1 所示，以滤波器 3 为例展示了触摸库的一个完整的最简使用流程。

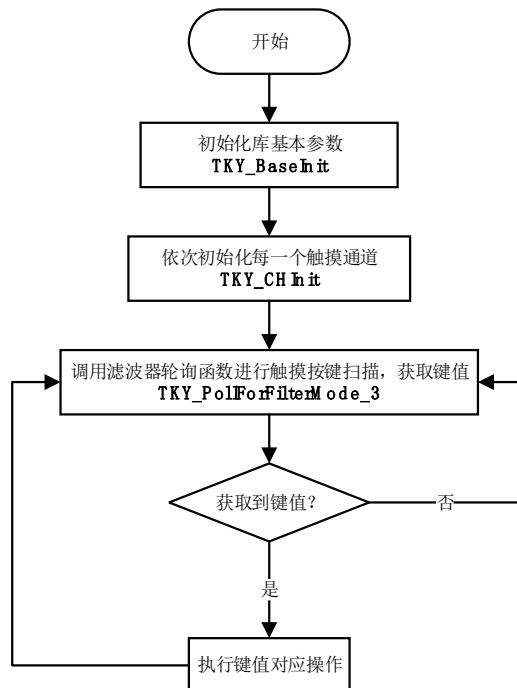


图 1 库使用流程图

2. 函数功能描述

2.1.TKY_BaseInit

uint8_t TKY_BaseInit(TKY_BaseInitTypeDef TKY_BaseInitStruct)

功能描述	对触摸按键进行基础参数初始化
输入参数	TKY_BaseInitStruct，该参数类型为 TKY_BaseInitTypeDef。
返回值	初始化结果。 0x00：初始化成功； 0x01：滤波器模式参数错误 0x02：滤波器等级参数错误。

初始化参数注解：

```
typedef struct
{
    uint8_t maxQueueNum;           //--测试队列数量--
    uint8_t singlePressMod;        //--单按键模式---
    uint8_t shieldEn;              //--屏蔽使能---
    uint8_t filterMode;            //--滤波器模式--
    uint8_t filterGrade;           //--滤波器等级--
    uint8_t peakQueueNum;          //--按键最大偏移队列---
    uint8_t peakQueueOffset;       //--按键最大偏移队列的偏移值---

    uint8_t baseRefreshOnPress;     //--基线在按键按下时是否进行--
    uint8_t baseUpRefreshDouble;    //--基线向上刷新倍速参数---
    uint8_t baseDownRefreshSlow;    //--基线向下更新降速参数---
    uint8_t rfu[2];                 //--保留--
    uint32_t baseRefreshSampleNum;   //--基线刷新采样次数--
    uint32_t *tkyBufP;              //--测试通道数据缓冲区指针--
}TKY_BaseInitTypeDef;
```

- **maxQueueNum**：使用的触摸按键通道数目，其值与实际使用的按键数目要严格一致，一般由触摸调试上位机 WCHTouchKeyTool 生成的配置头文件提供。
- **singlePressMod**：单按键模式开启使能，
0：多按键模式，多个按键按下时同时输出键值；
1：单按键模式，当多个按键同时按下时，输出变化量最大的键值；
2：互斥单按键模式，只有当前按键释放后才可以触发下一个按键。
- **shieldEn**：屏蔽功能使能，该功能需要硬件支持，可显著增大触摸灵敏度。目前仅 CH58x、CH59x、CH32V006、CH32L103 系列支持。
1：启动屏蔽功能；
0：关闭屏蔽功能。
- **filterMode**：选择要使用哪种滤波器模式，配置值参考库头文件，关于每种滤波器的描

述详见第一节总述。

- **filterGrade**：滤波器等级，该参数可设置 0~15。该参数越大，检测结果越稳定，抗干扰能力越强，但是该参数过大，容易造成检测结果滞后或间隔非常短暂的按键值无法检测出来，所以需要根据实际应用中的需求，来设置该参数的大小。此外该参数为 0 时，并不代表滤波器未启用，只是滤波器未得到进一步加强。
- **peakQueueNum**：按键最大偏移队列的队列号。该变量应用场景为——在 CS10V 测试，可设置为波动最大的通道值或最容易误触摸的通道值，若无测试条件，建议设置为手指按下时变化量最大的通道值。非 CS10V 滤波器使用时无需配置。
- **peakQueueOffset**：按键最大偏移队列的偏移值。该变量应用场景同 peakQueueNum，在 CS10V 测试中，对 peakQueueNum 所设置的通道进行偏移修正，通常偏移值可以通过测试修改，通常尽可能赋较小数值。在无需进行 CS10V 测试的场景，该值可设置为 0。
- **baseRefreshOnPress**：基线更新在按键按下时是否进行。
1：触摸通道触发时继续按设定的间隔更新基线；
0：触摸通道触发时，基线停止更新。
- **baseRefreshSampleNum**：基线更新采样次数，该参数范围 0~65535，设置为 0 时，关闭基线刷新，参数越大，基线更新越缓慢。在同样的 baseRefreshSampleNum 设置值下，滤波器等级越高，基线刷新越慢。
- **baseUpRefreshDouble**：基线向上更新的倍数参数，本质上增加基线向上刷新的步进，该参数范围 0~255，设置为 0 或 1 时不加速，如果对基线向上刷新和向下刷新有不同的要求时，尽量选择设置向下减速。
- **baseDownRefreshSlow**：基线向下更新的减速参数，该参数范围为 0~65535。该参数与 baseUpRefreshDouble 主要是为了解决基线向上更新和向下更新不同速的需求。具体请参考示例里的补充说明。
- **tkyBufP**：测试通道数据缓冲区指针，由用户在外部初始化，配置进库，需要注意该缓冲区地址需要 4 字节对齐。

示例代码如下：

```
uint32_t tkyBuf[(MAX_QUEUEBUF_LEN-1)/4 + 1] = {0};
TKY_BaseInitStructure.filterMode = FILTER_MODE_3;
TKY_BaseInitStructure.shieldEn = 1;
TKY_BaseInitStructure.singlePressMod = 2;
TKY_BaseInitStructure.filterGrade = 4;
TKY_BaseInitStructure.maxQueueNum = 12;
TKY_BaseInitStructure.baseRefreshOnPress = 0;
TKY_BaseInitStructure.baseRefreshSampleNum = 1000;
TKY_BaseInitStructure.baseUpRefreshDouble = 0;
TKY_BaseInitStructure.baseDownRefreshSlow = 0;
TKY_BaseInitStructure.tkyBufP = tkyBuf;
TKY_BaseInit(TKY_BaseInitStructure);
```

补充说明：部分用户通道初始化时，直接取当时所测试得平均值赋值给基线变量，这样

会产生一个问题，当上电时用户恰好触碰按键时，基线变量值小于实际值。

解决该问题有两个思路，其一是每隔一个较长时间(数量级可以为 10 分钟以上)保存下基线值，上电时读取非易失性存储器里的值，而非上电临时取平均值。思路二是一般来说产生该问题的是由于上电按压导致基线值小于实际值，提高基线更新速度，当手指第一次释放时，让基线尽快回归正常值。但是基线向下更新过快，容易导致触摸到判决触摸产生的过程中，基线值靠近实测值，从而带来新的问题。所以配合 baseUpRefreshDouble 和 baseDownRefreshSlow 两个变量，配合 baseRefreshSampleNum 使用，来控制向上和向下刷新不同的速度。**但需要注意的是，使用思路二解决此问题时，在进行抗扰度测试时，有一定的可能会造成基线更新偏差。**

2.2.TKY_CHInit

uint8_t TKY_CHInit(TKY_ChannelInitTypeDef TKY_CHInitStruct)

功能描述：	对触摸按键单个触摸通道进行通道参数初始化。
输入参数：	TKY_CHInitStruct，该参数类型为 TKY_ChannelInitTypeDef。
返回值	初始化结果。 0x00：初始化成功； 0x01：触摸通道参数有错； 0x02：通道转换队列位置错误(超过最大转换通道数量)， 0x04：基线值设置错误； 0x08：阈值设置错误。

初始化参数注解：

```
typedef struct
{
    uint8_t queueNum;           //--该通道在测试队列的序号--
    uint8_t channelNum;        //--该通道对应的 ADC 通道标号--
    uint16_t chargeTime;       //--该通道充电时间--
    uint16_t disChargeTime;    //--该通道放电时间--
    uint16_t baseLine;         //--基线--
    uint16_t threshold;        //--阈值--
    uint16_t threshold2;       //--阈值 2--
    uint8_t sleepStatus;       //--休眠--
}TKY_ChannelInitTypeDef;
```

- **channelNum**：触摸按键通道编号，通常也是 ADC 通道编号。
- **queueNum**：该触摸按键在转换队列中的位置。该参数不得超过头文件中的 MAX_QUEUE_NUM 值，同一个触摸按键可以占有转换队列的多个位置(不允许在 CS10V 相关测试场景中重复占位)。
- **chargeTime**：当前通道充放电时间，取值范围参考各芯片手册。
- **disChargeTime**：当前道充放电时间，取值范围参考各芯片手册。

- **sleepStatus** : 当前通道休眠标志, 为 1 休眠, 则轮询中不检测, 用于滤波模式 3 中。
- **baseLine** : 当前通道的初始基线值, 最大不得超过 4095, 建议通过调整充放电时间, 使基线值处于 3000~3600 之间, 该值一般由触摸调试上位机产生或者由库自动校准程序产生。
- **threshold** : 上升阈值, 当基线值和当前检测值的差值大于该参数值时, 则认为有按键按下。
- **threshold2** : 下降阈值, 当基线值和当前检测值的差值小于该参数值时, 则认为有按键释放。

示例代码如下：

```
TKY_ChannelInitStructure.queueNum = 0;
TKY_ChannelInitStructure.channelNum = 7;
TKY_ChannelInitStructure.threshold = 70;
TKY_ChannelInitStructure.threshold2 = 60;
TKY_ChannelInitStructure.chargeTime = 2;
TKY_ChannelInitStructure.disChargeTime = 2;
TKY_ChannelInitStructure.sleepStatus = 0;
TKY_ChannelInitStructure.baseLine=3500;
TKY_CHInit(TKY_ChannelInitStructure);
```

2.3.TKY_GetCurChannelMean

uint16_t TKY_GetCurChannelMean(uint8_t curChNum, uint8_t chargeTime, uint8_t disChargeTime, uint16_t averageNum);

功能描述	对当前触摸通道进行多次触摸测量, 获取当前触摸通道的测量值。该函数可用于调试各通道的充放电时间, 获取基线值和阈值
输入参数	CurChNum : 当前测量触摸通道编号, 通常也是 ADC 通道编号。 ChargeTime : 当前触摸通道的充电时间。 disChargeTime : 当前触摸通道的放电时间。 averageNum : 测试次数。
输出参数	多次测量的平均值

2.4.TKY_GetCurQueueValue

uint16_t TKY_GetCurQueueValue(uint8_t curQueueNum)

功能描述	获取任意触摸通道滤波后的测量值。该函数可用于用户自行处理滤波后数据
输入参数	curQueueNum, 所选择的测量触摸通道队列编号, 注意并非 ADC 通道编号
输出参数	滤波模式 3 和 CS10 时为所选择的通道滤波后的变化值(可直接和阈值比较)
输出参数	各通道按键值, 返回值各个位对应各个队列的按键, 例如队列 0 的触摸通道有按键, 对应最低位置 1。注意并非 ADC 通道编号

2.5.TKY_PollForFilterMode_3

uint16_t TKY_PollForFilterMode_3(void)

功能描述	滤波器模式 3 下的轮询函数，该函数是阻塞式函数，每次执行需要把所有非休眠通道转换完成后退出，执行时间请参考对应 MCU 的数据手册。该函数无需频繁执行，通常几十毫秒级别即可，执行频率和滤波强度设置值成正比
输入参数	无
输出参数	各通道按键值，返回值各个位对应各个队列的按键，例如队列 0 的触摸通道有按键，对应最低位置 1。注意并非 ADC 通道编号

2.6.TKY_PollForFilterMode_CS10

uint16_t TKY_PollForFilterMode_CS10(void)

功能描述	滤波器模式 CS10 下的轮询函数，该函数可应用于需要通过 CS10V 测试场景中，轮询函数中有阻塞式随机延迟(0~8us)
输入参数	无
输出参数	各通道按键值，返回值各个位对应各个队列的按键，例如队列 0 的触摸通道有按键，对应最低位置 1。注意并非 ADC 通道编号

2.7.TKY_ScanForWakeUp

uint16_t TKY_ScanForWakeUp(uint16_t scanBitValue)

功能描述	对指定的通道进行快速扫描，若有疑似触摸信号，则进行快速扫描。主要应用在低功耗休眠场景下
输入参数	scanBitValue，该值的置 1 的位，为指定扫描位，例 0x0013，则对序列 0、1、4 通道进行扫描
输出参数	扫描通道疑似按键值，返回值各个位对应各个队列的按键，例如队列 0 的触摸通道有按键，对应最低位置 1。注意并非 ADC 通道编号

2.8.TKY_SetCurQueueSleepStatus

uint8_t TKY_SetCurQueueSleepStatus(uint8_t curQueueNum, uint8_t sleepStatus)

功能描述	设置指定队列通道的休眠模式，休眠后函数 TKY_PollForFilterMode 将不再对该通道进行轮询测试，休眠仅对滤波模式 3&5 有效
输入参数	curQueueNum：所选择的测量触摸通道队列编号，注意并非 ADC 通道编号。 sleepStatus：所设置的休眠状态，为 0 不休眠，非 0 则休眠
输出参数	设置成功输出 0，队列序号超限输出 1

2.9. TKY_SetSleepStatusValue

void TKY_SetSleepStatusValue(uint16_t setValue)

功能描述	设置所有队列通道的休眠模式，与函数 TKY_SetCurQueueSleepStatus 相比，本函数是直接配置全部通道休眠状态，对应位为 0 则不休眠，为 1 则休眠
输入参数	setValue,所有通道的休眠状态，对应位为 0 则不休眠，为 1 则休眠
输出参数	无

2.10. TKY_ReadSleepStatusValue

uint16_t TKY_ReadSleepStatusValue(void)

功能描述	设置所有队列通道的休眠模式，与函数 TKY_SetCurQueueSleepStatus 相比，本函数是直接配置全部通道休眠状态，对应位为 0 则不休眠，为 1 则休眠
输入参数	无
输出参数	返回所有通道的休眠状态，对应位为 0 则不休眠，为 1 则休眠

2.11. TKY_SetCurQueueChargeTime

uint8_t TKY_SetCurQueueChargeTime(
uint8_t curQueueNum, uint8_t chargeTime, uint8_t disChargeTime)

功能描述	设置指定队列通道的充放电时间。为确保安全更新设置，请查询空闲状态(TKY_GetCurIdleStatus)，在空闲时进行更新
输入参数	curQueueNum：所选择的测量触摸通道队列编号，注意并非 ADC 通道编号。 chargeTime：所设置的充电时间参数，参数值含义请查阅芯片手册中的寄存器相关参数。 disChargeTime：所设置的放电时间参数，参数值含义请查阅芯片手册中的寄存器相关参数。
输出参数	设置成功输出 0，队列序号超限输出 1

2.12. TKY_SetCurQueueThreshold

uint8_t TKY_SetCurQueueThreshold(
uint8_t curQueueNum, uint8_t threshold, uint8_t threshold2)

功能描述	设置指定队列通道的阈值。为确保安全更新设置，请查询空闲状态(TKY_GetCurIdleStatus)，在空闲时进行更新
输入参数	curQueueNum：所选择的测量触摸通道队列编号，注意并非 ADC 通道编号。 Threshold：该通道的上限阈值，即按键检测状态由无变有的判决值。

	Threshold2：该通道的下限门槛值，即按键检测状态由有变无的判决值
输出参数	设置成功输出 0，队列序号超限输出 1

2.13. TKY_GetCurIdleStatus

uint8_t TKY_GetCurIdleStatus(void)

功能描述	查询当前 touchKey 是否空闲状态，若处于空闲状态，可以切换 ADC 状态
输入参数	无
输出参数	touchKey 处于空闲状态输出 1，非空闲状态输出 0

2.14. TKY_GetCurVersion

uint16_t TKY_GetCurVersion (void)

功能描述	查询当前库版本
输入参数	无
输出参数	假设当前版本为 1.03，则输出 103

2.15. TKY_GetCurQueueBaseLine

uint16_t TKY_GetCurQueueBaseLine(UINT8 curQueueNum)

功能描述	获取指定队列通道的基线值
输入参数	curQueueNum，所选择的测量触摸通道队列编号，注意并非 ADC 通道编号
输出参数	所选择队列通道的基线值

2.16. TKY_SetBaseRefreshSampleNum

void TKY_SetBaseRefreshSampleNum(uint8_t newValue)

功能描述	设置基线刷新采样次数，每隔多少次采样刷新一次基线。为确保安全更新设置，请查询空闲状态(TKY_GetCurIdleStatus)，在空闲时进行更新
输入参数	newValue，新采样次数
输出参数	无

2.17. TKY_SetBaseUpRefreshDouble

void TKY_SetBaseUpRefreshDouble (uint8_t newValue)

功能描述	设置基线向上更新的倍数参数。为确保安全更新设置，请查询空闲状态(TKY_GetCurIdleStatus)，在空闲时进行更新
------	--

输入参数	newValue，新采样次数
输出参数	无

2.18. TKY_SetBaseDownRefreshSlow

void TKY_SetBaseDownRefreshSlow(uint16_t newValue)

功能描述	设置基线向下更新的减速参数。为确保安全更新设置，请查询空闲状态(TKY_GetCurIdleStatus)，在空闲时进行更新
输入参数	newValue，新采样次数
输出参数	无

2.19. TKY_SetFilterMode

void TKY_SetFilterMode(uint8_t newValue)

功能描述	设置新滤波模式。一般场景不建议使用，仅使用在休眠场景下特殊场景切换，具体参看低功耗休眠例程。
输入参数	newValue，新滤波模式
输出参数	无

2.20. TKY_ClearHistoryData

void TKY_ClearHistoryData(uint8_t curFilterMode)

功能描述	清除当前滤波器的历史数据。应用在触摸转换被较长时间打断，历史数据无意义的场景。
输入参数	curFilterMode，当前滤波模式，便于清除不同的滤波器。
输出参数	无

2.21. TKY_SaveAndStop

void TKY_SaveAndStop (void)

功能描述	保存触摸相关寄存器值，并且在判断触摸扫描空闲时暂停触摸功能，以腾出 ADC 模块用于 ADC 转换
输入参数	无
输出参数	无

2.22. TKY_LoadAndRun

void TKY_LoadAndRun (void)

功能描述	载入触摸相关寄存器值，并重新启动被暂停的触摸按键功能
输入参数	无
输出参数	无