

MC51F8124 触摸板及触摸库使用说明

2021/07/02



SinoMCU

目录

1	概述.....	4
2	运行环境.....	4
3	硬件准备	4
3.1	硬件环境.....	4
3.2	硬件连接.....	6
3.2.1	SN-LINK-S1 仿真器图示	6
3.2.2	仿真器和目标板连线示意图	7
4	软件准备	8
4.1	软件启动.....	8
4.1.1	打开 demo 工程	8
4.1.2	编译并烧录程序	8
4.1.3	脱机运行程序	11
4.2	运行效果.....	11
5	demo 板原理图.....	14
6	触摸库快速入门.....	15
6.1	触摸库占用资源.....	15
6.2	触摸库运行环境.....	15
6.3	触摸库工作原理.....	15
6.3.1	判键触发的 4 个基本概念	15
6.3.2	判键触发基本公式	16
6.4	触摸库文件结构.....	17
6.5	新工程植入触摸库	18
6.5.1	新建工程.....	18
6.5.2	移植触摸库资源	22
7	触摸库详细介绍.....	25
7.1	tkm_config.h 配置文件说明	25
7.1.1	按键通道设定	25
7.1.2	通用触控参数设定	25
7.1.3	寄存器设定	27
7.2	函数及常用变量说明	28
7.2.1	函数功能说明	28
7.2.2	常用变量使用说明	29
7.3	TK 库调试步骤说明	30
8	TouchKey 上位机简介	34
9	软件运行环境.....	34
9.1	硬件连接.....	34
9.2	软件运行	35
10	软件使用	36
10.1	串口设置.....	36
10.2	开始采样.....	37
10.3	图表显示设置.....	38
10.4	数据保存.....	39

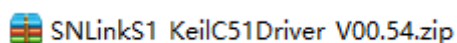
10.4.1	“txt” 文本格式	40
10.4.2	“csv” 文件格式	41
11	demo 板验证测试	43
12	版本及更新记录	43

1 概述

本文档主要指导大家完成对我司 MC51F8124 芯片开发的 TK demo 板程序的烧录操作, 演示 TK 效果; 并对如何使用 TK 库进行详细说明。

2 运行环境

1. 需安装 KEIL 51 软件。
2. 安装我司 SNLinkS1 插件安装包 (在我司提供的 “Keil 及仿真器工具 \SNLinkS1_KeilC51Driver_Vxx.xx.zip 中, 如下图红框所示)。



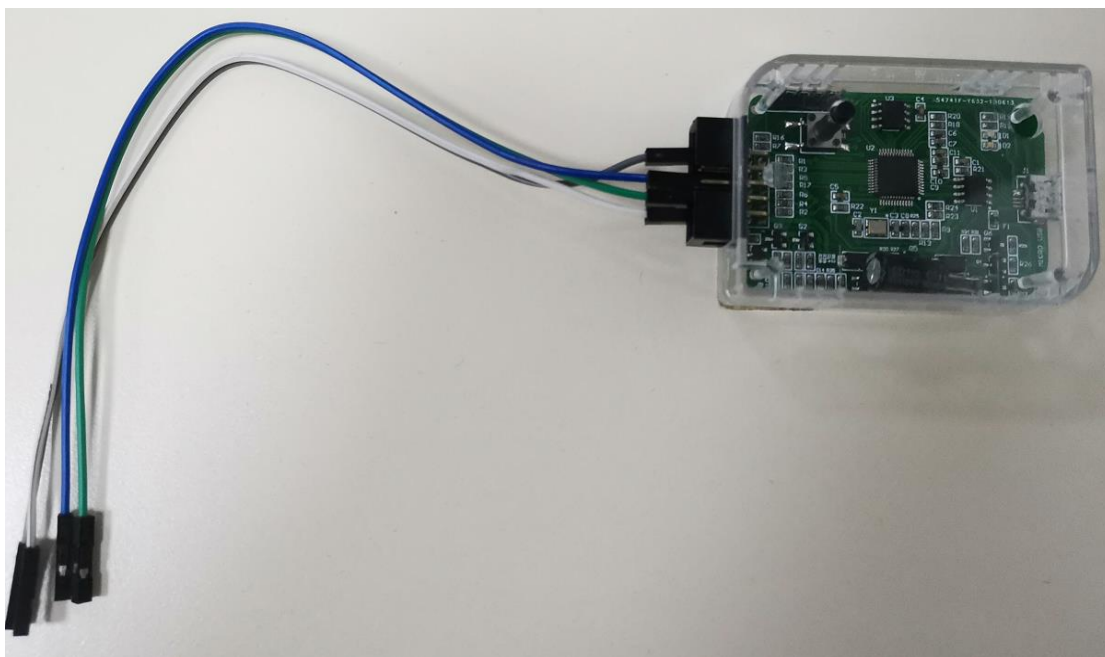
3 硬件准备

3.1 硬件环境

1. 配备 MC51F8124 芯片的 TK demo 板一块 (以下都以此 demo 板为例演示讲解), 如下图所示。



2. SN-LINK-S1 仿真器 1 个及 4 根孔对孔杜邦线。



3. micro usb 线一根。



3.2 硬件连接

3.2.1 SN-LINK-S1 仿真器图示

1. 仿真器正面图示：

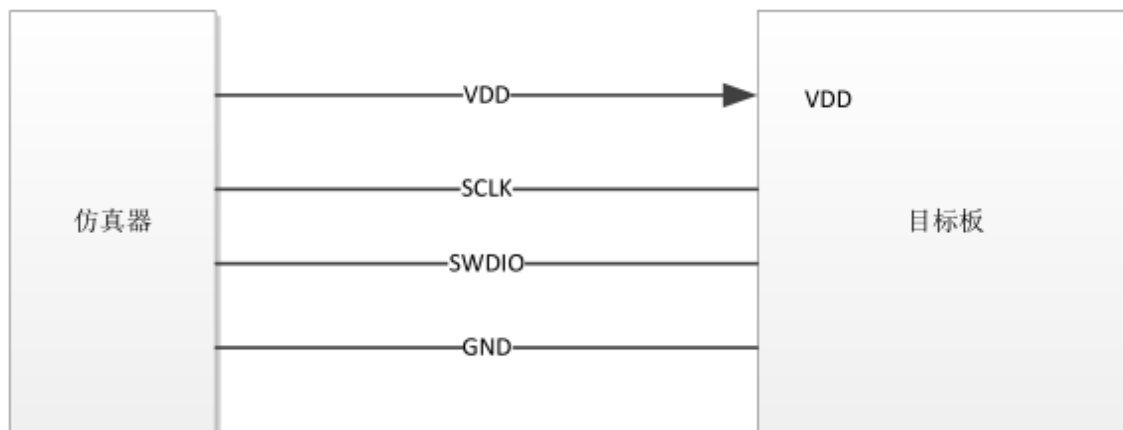


2. 仿真器反面图示

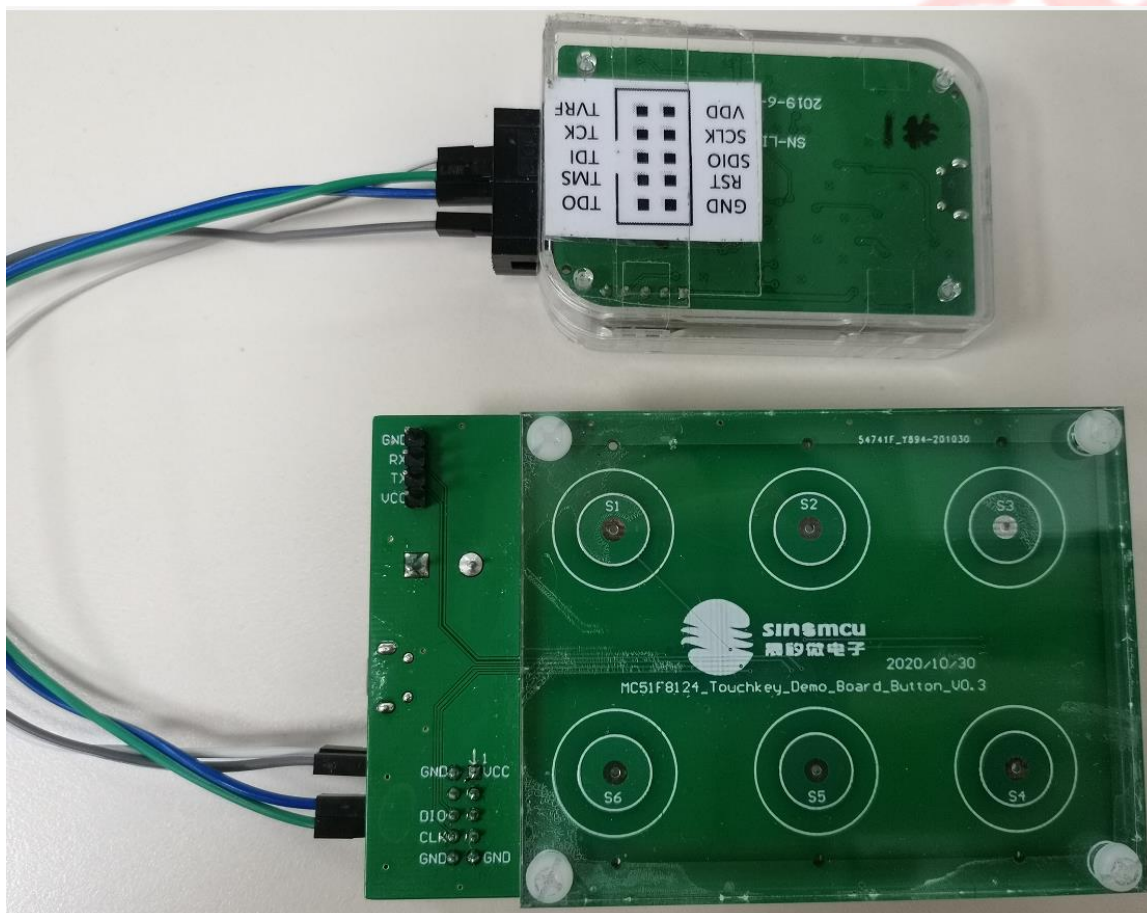


- 1: 10Pin 仿真、下载 接口：仿真器支持串行 SWDIO 连接方式。
- 2: micro usb: 连接 PC，用于仿真和下载的数据通讯。
- 3: led 指示灯：红色表示正常运行状态，绿色表示处于固件升级状态。
- 4: 接口丝印：指示仿真、下载接口在 10pin 牛角公头的定义。

3.2.2 仿真器和目标板连线示意图



连接全图如下所示:

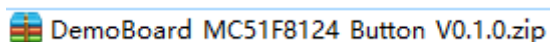


4 软件准备

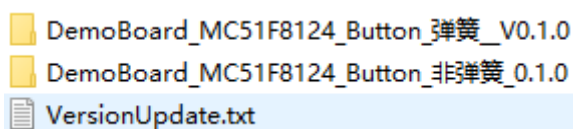
4.1 软件启动

4.1.1 打开 demo 工程

如下图所示，在“demo 程序”文件夹中存有我司的 demo 板样例程序。



解压该文件夹后，内容如下图所示：



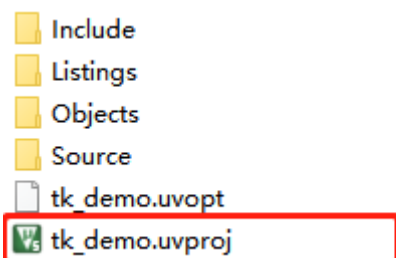
根据 demo 板硬件（弹簧和非弹簧按键）不同，此文件夹相应包含两个 demo 程序：

- 1) 弹簧按键：DemoBoard_MC51F8124_Button_弹簧_Vx.x.x。
- 2) 非弹簧按键：DemoBoard_MC51F8124_Button_非弹簧_Vx.x.x。

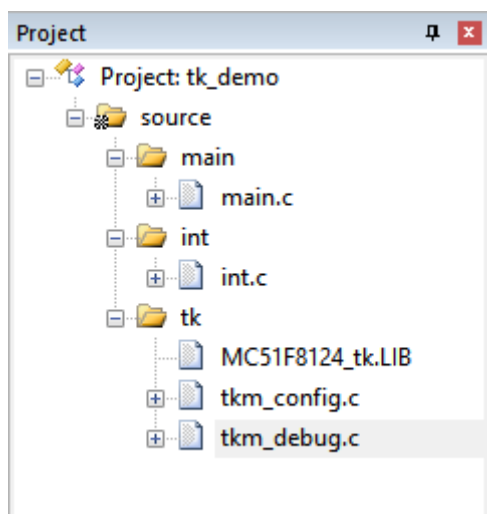
文件夹中 VersionUpdate.txt 中记录了 demo 程序的更新细节。下面以非弹簧的 demo 板及其程序为例进行讲解。

4.1.2 编译并烧录程序

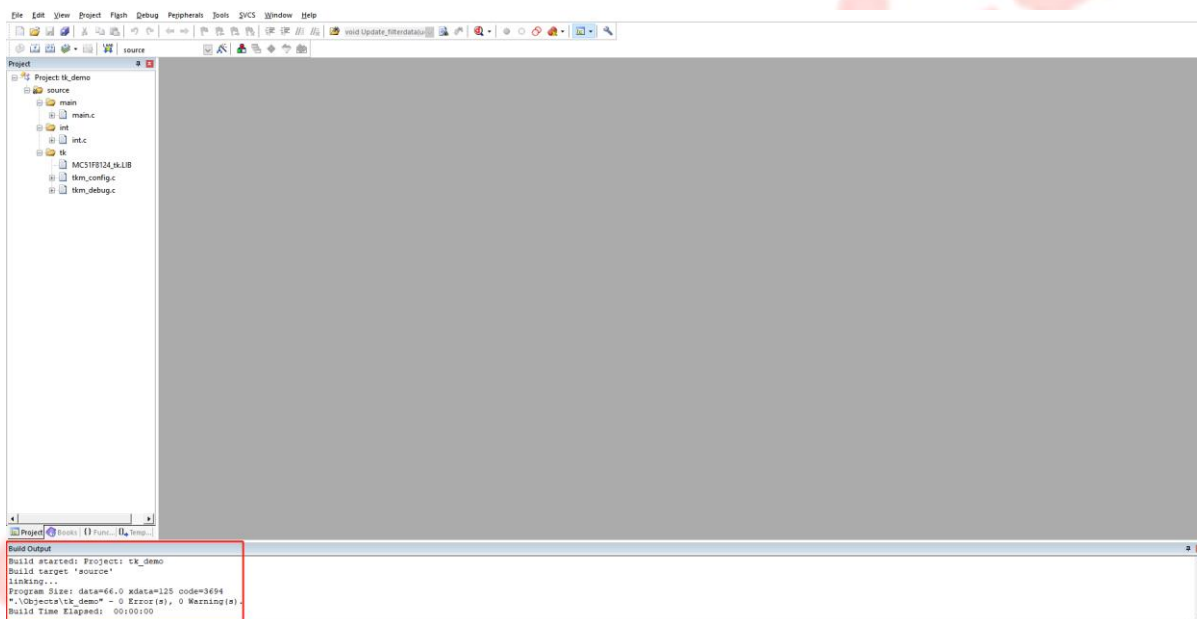
- 1 打开“DemoBoard_MC51F8124_Button_非弹簧_Vx.x.x”的工程文件夹，找到下图所示工程文件并使用 Keil C51 软件打开。



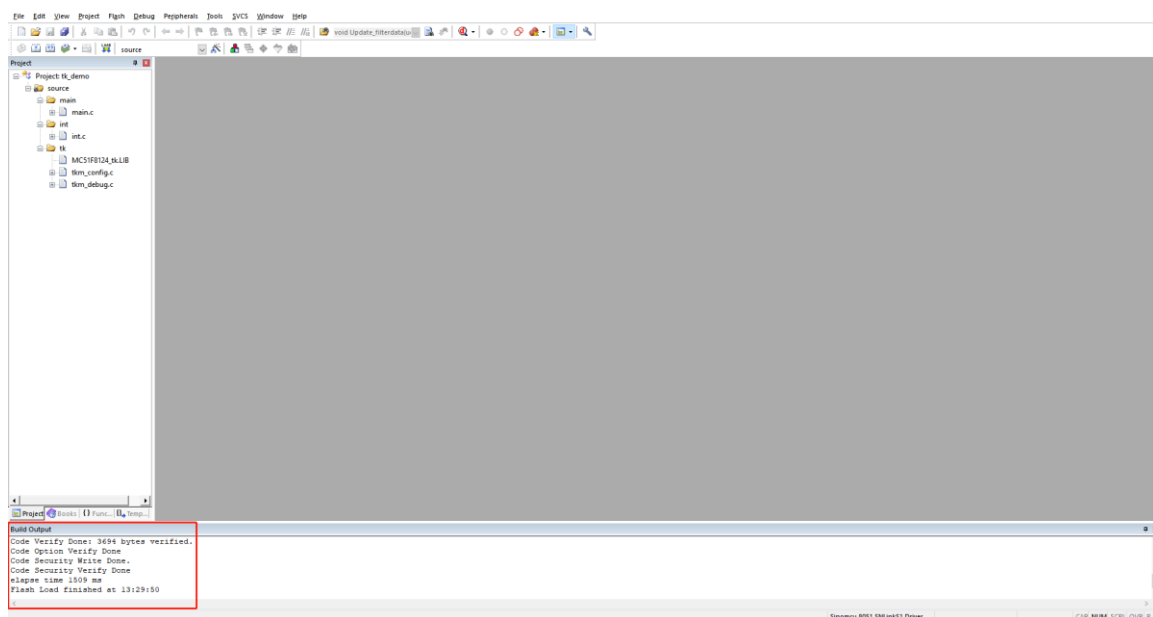
2 打开工程文件后如下图所示。



3 编译并运行程序，编译成功后提示信息如下图所示红框中所示。



4 烧录程序到 demo 板中，烧录成功后提示信息如下图红框中所示。



4.1.3 脱机运行程序

建议拔除仿真器，使用 micro usb 线供电给 demo 板。如下图所示。



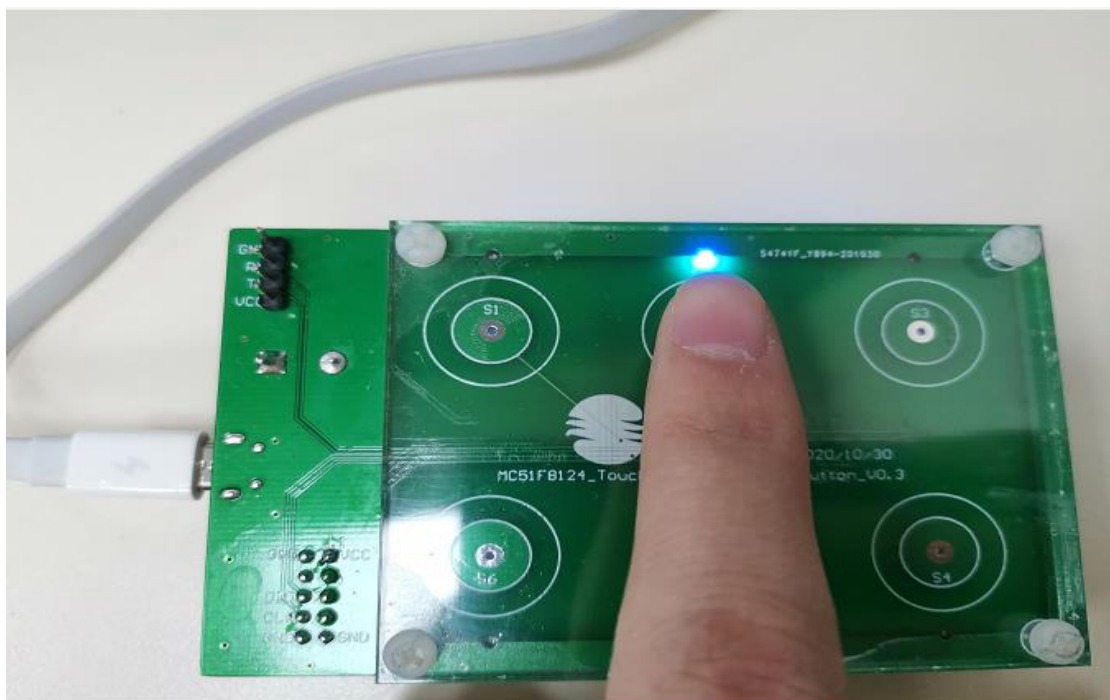
4.2 运行效果

程序烧录完毕开始运行，蜂鸣器会鸣叫一声作为响应。每个按键被触发时，蜂鸣器鸣叫，对应按键旁边的 LED 灯被点亮。demo 板按键触发时效果如下图所示。

1. 按键 S1 被触发



2. 按键 S2 被触发



3. 按键 S3 被触发



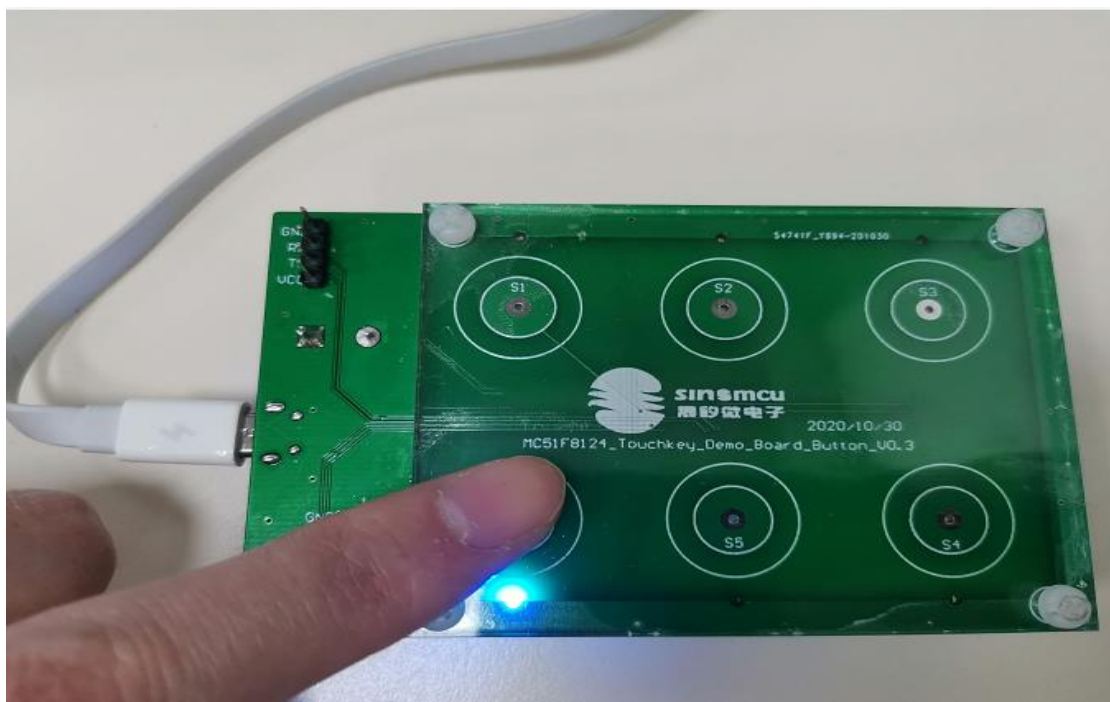
4. 按键 S4 被触发



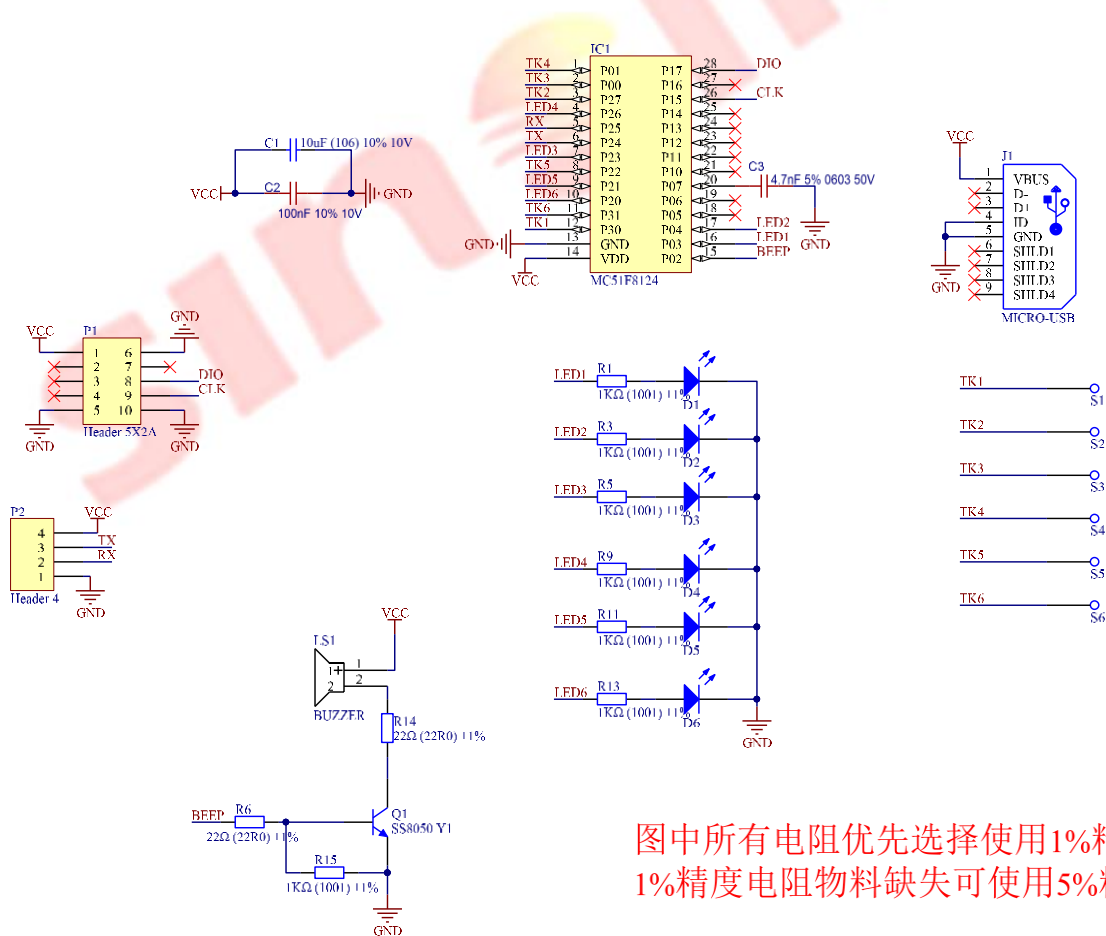
5. 按键 S5 被触发



6. 按键 S6 被触发



5 demo 板原理图



6 触摸库快速入门

6.1 触摸库占用资源

- 1 ROM 用量: (1) 单个 TK 约为 3370bytes; (2) 每增加 1 个 TK, ROM 增加 3bytes。
- 2 RAM 用量: (1) 单个 TK 约为 92bytes; (2) 每增加 1 个 TK, RAM 增加 20bytes。

6.2 触摸库运行环境

- 1 需安装 KEIL 51 平台 (Keil uVision4 及以上版本)。
- 2 安装 SNLinkS1 安装包: SNLinkS1_KeilC51Driver_Vxx.xx.zip。

6.3 触摸库工作原理

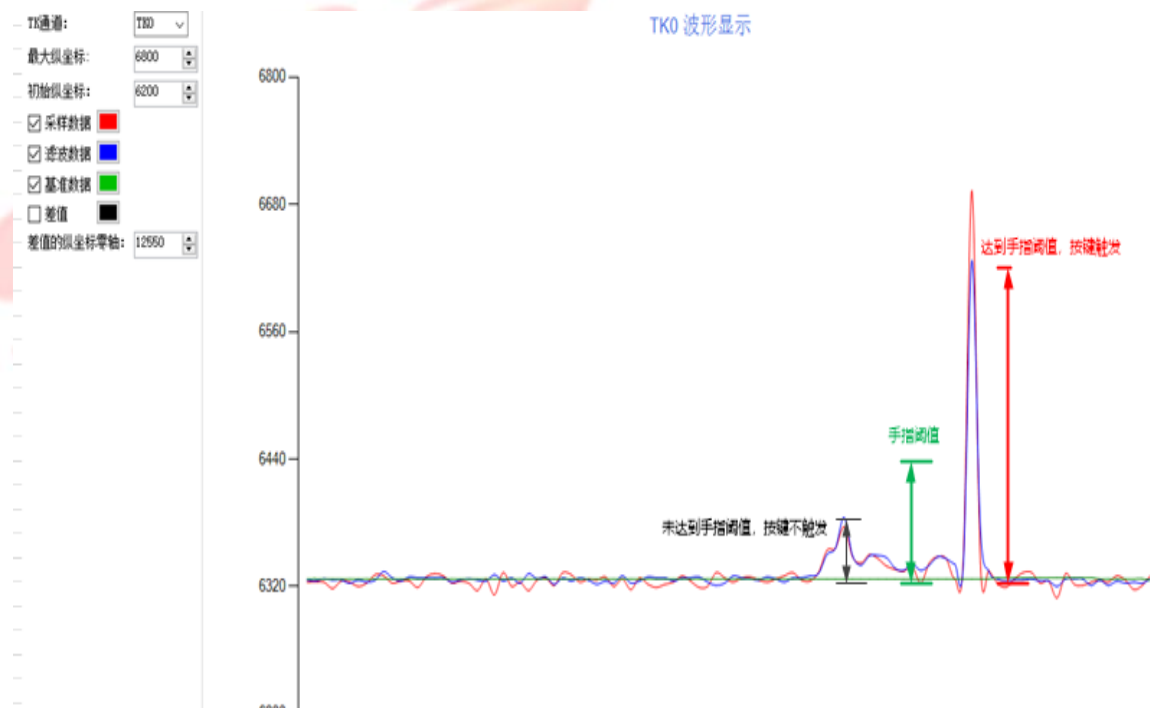
芯片 TK 模块工作过程中, 当有按键触发时, TK 采集数值会突然增大, 与按键触发前, TK 采集数值形成一定的差值, 根据此差值判断按键触发。

由于外界环境的干扰导致 TK 采集数值产生的变化和上述描述基本一致, 因此需要对 TK 采集数值进行软件滤波算法处理, 还原数据的真实性。

为有效判断按键触发, 需要引入 4 个基本概念: 采样值、滤波值、基准值、手指阈值。

6.3.1 判键触发的 4 个基本概念

下图为实际触摸曲线图。



1 采样值

芯片 TK 模块工作过程中，TK 模块的采样数值。

2 滤波值

如上图所示，由于电容式触控按键易受外界环境干扰导致 TK 采集数值异常，因此需要将采样值进行滤波处理，排除干扰还原数据的真实性，此数据结果就是滤波值。

3 基准值

为有效判断按键触发，必须确保相对滤波值有一个较为稳定的参考对象（常态环境的数值），因此需要引入一个基准值概念，同时该曲线也需要根据环境进行自适应的变化（不受按键和噪声的影响），如上图所示。

4 手指阈值

如上图所示，通过设定手指阈值，确保每次手指触摸按键时变化的差值范围大于该数值，判断按键触发有效。

6.3.2 判键触发基本公式

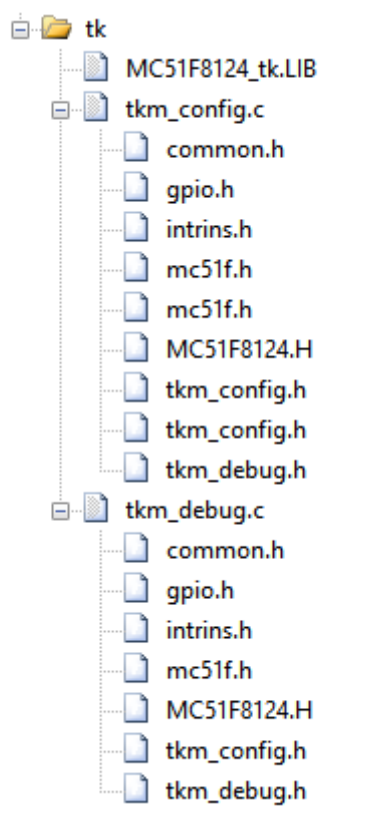
$$\Delta X = T_f - T_a \quad (1)$$

$$\Delta X > T_s \quad (2)$$

式中：

- 1 T_a 为基准值；
- 2 T_f 为滤波值；
- 3 T_s 为手指阈值。

6.4 触摸库文件结构



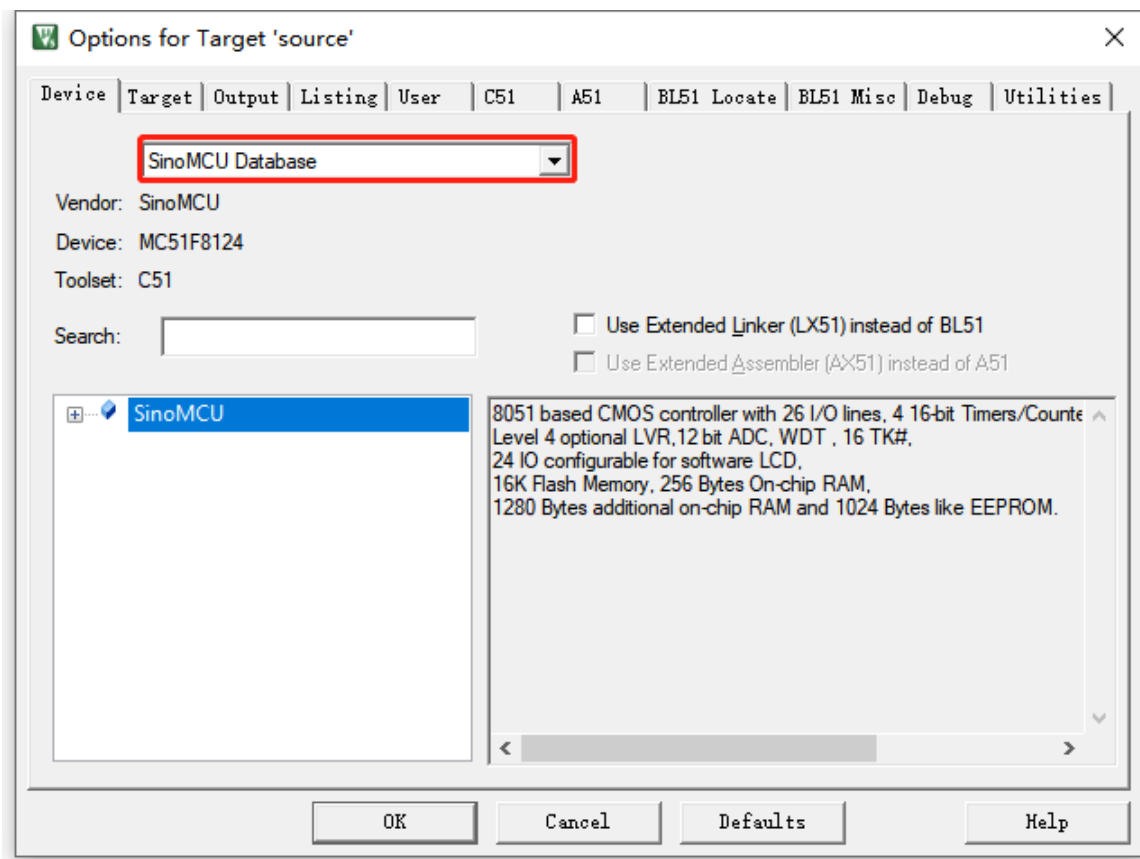
文件类型	文件名	文件功能
Include	tkm_config.h	TK 参数配置文件
	mc51f.h	包含 MC51F8124.H、gpio.h 和 common.h 文件
	MC51F8124.H	MC51F8114 芯片寄存器配置文件
	gpio.h	IO 端口配置宏定义文件
	common.h	通用数据类型和功能模块宏定义文件
	tkm_debug.h	TK 与上位机工具通信配置文件
Source	MC51F8124_tk.LIB	MC51F8124 芯片 TK 算法库
	tkm_config.c	TK 参数配置的实现文件
	tkm_debug.c	TK 与上位机工具通信程序实现文件 (占用 UART 资源)

6.5 新工程植入触摸库

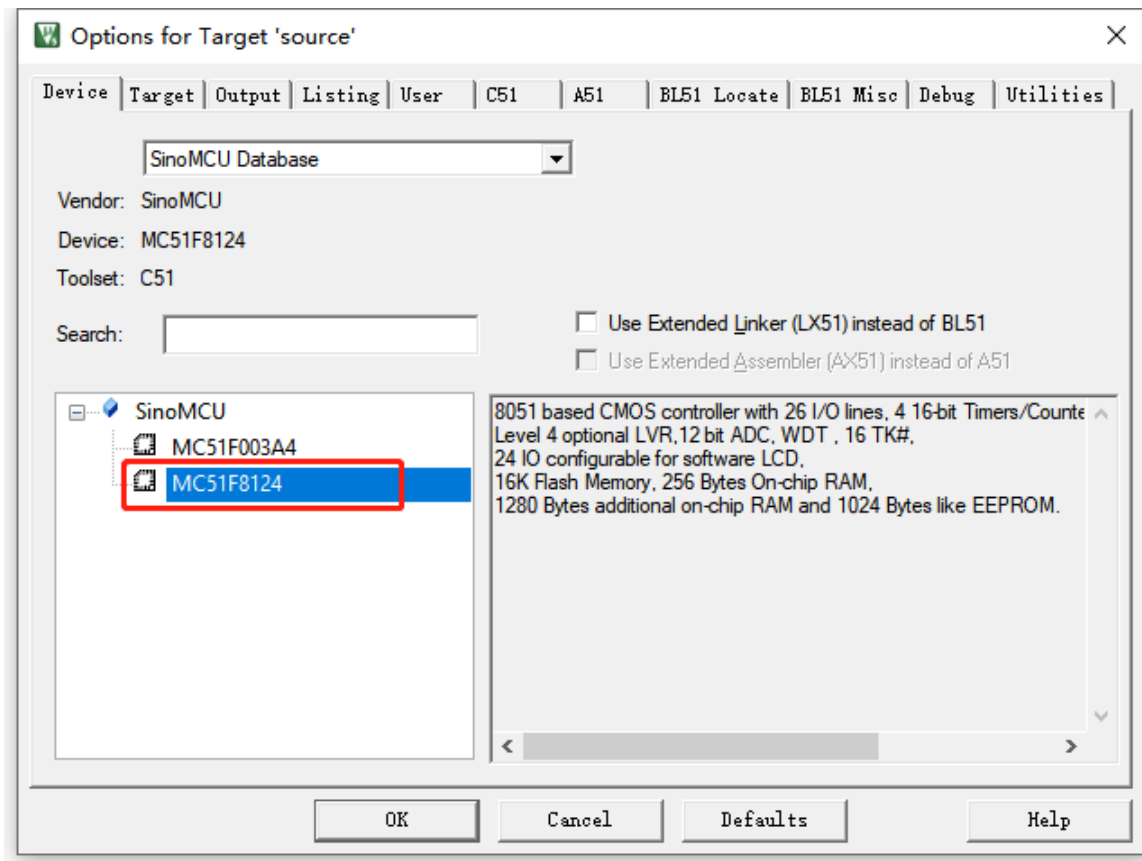
6.5.1 新建工程

1 Device 芯片类型

- 选择 SinoMCU Database



➤ 选择具体型号



2 SinoMCU_S1 文件路径检查

若 1 中“Device 芯片类型”已顺利执行，则可跳过此标题所述操作。若未查找到 Device 芯片，请执行下述操作。

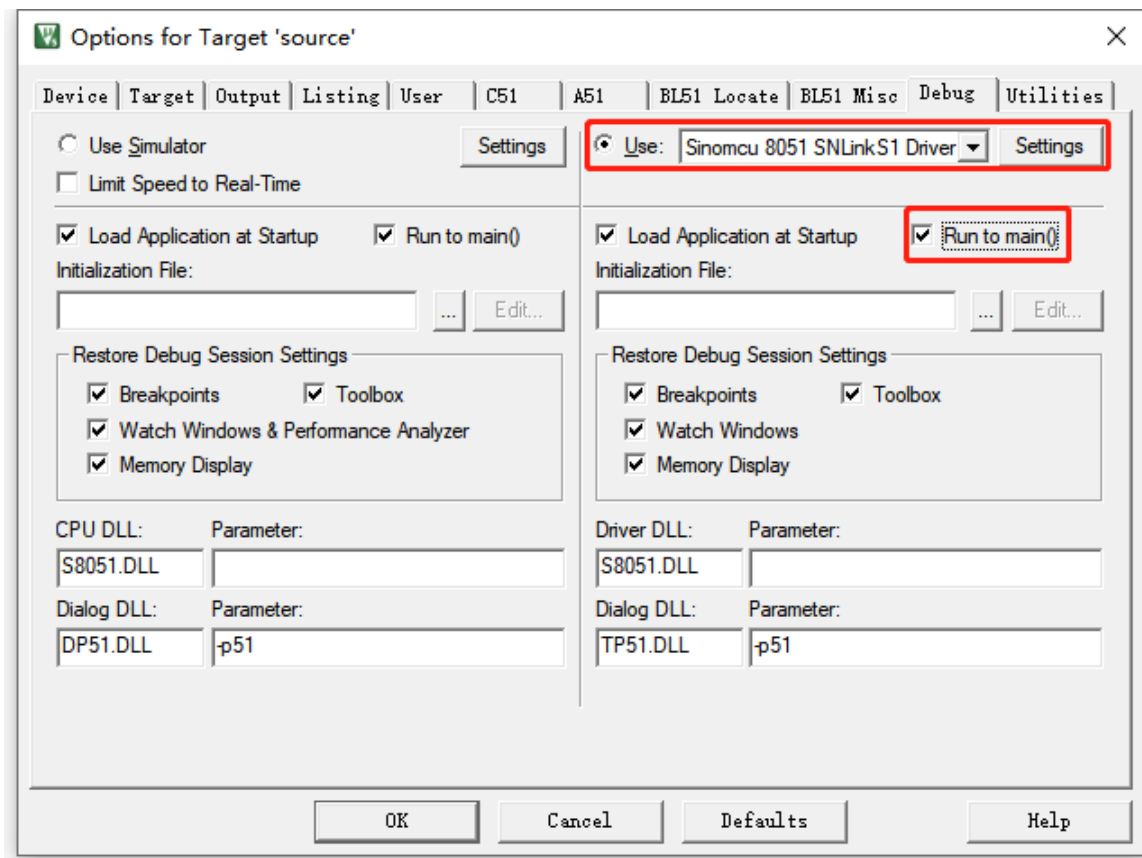
如下图所示，确认 SinoMCU_S1 文件夹是否添加在 UV4、C51 同一级目录（均应该放在 keil 根目录下），若 SinoMCU_S1 文件夹不和 UV4、C51 在同一级目录，请将 SinoMCU_S1 文件夹剪切到与 UV4、C51 同一级目录下。然后请重新执行“Device 芯片类型”操作。

	C51	2020/12/3 14:33	文件夹
	SinoMCU_S1	2020/12/3 14:33	文件夹
	UV4	2020/12/3 14:32	文件夹

3 Debug 设置

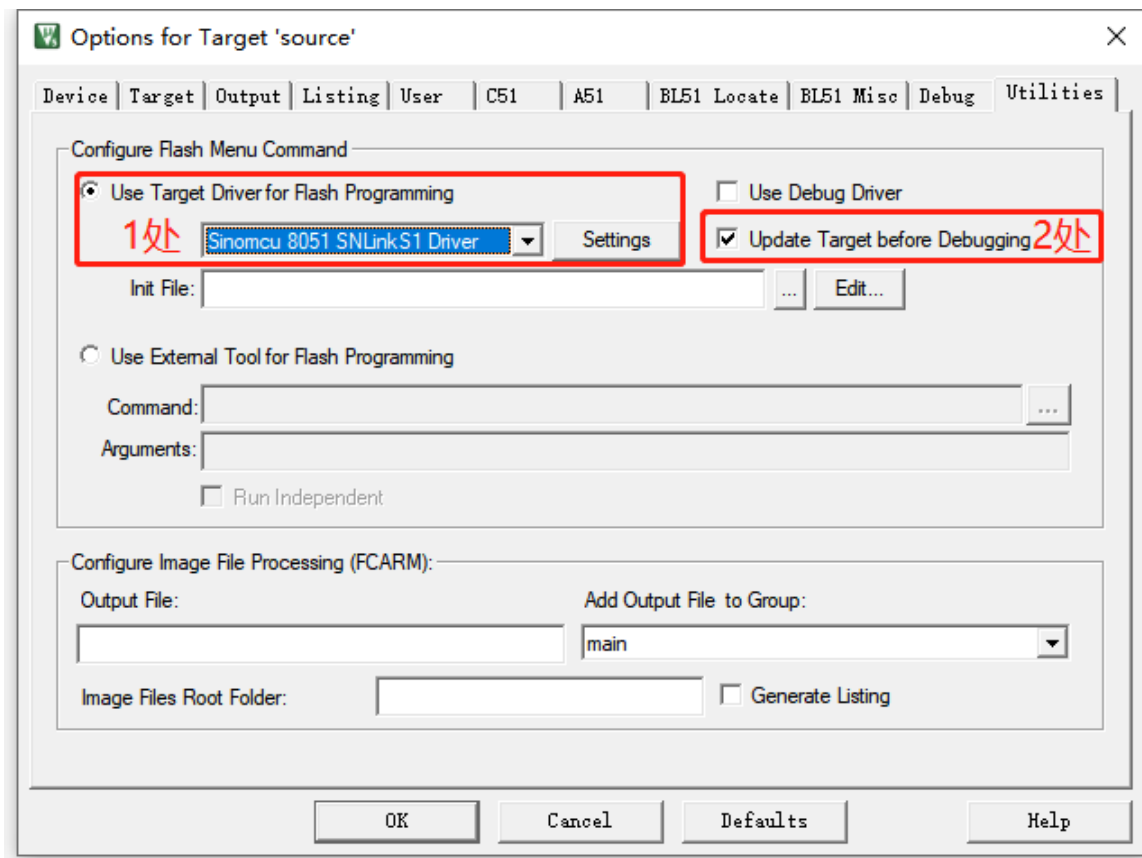
1: 选择调试驱动, 如下图红框中标注, 选择 “Use” 单选按钮和 “Sinomcu 8051 SLinkS1 Driver”。

另外如果勾选 “Run to main”, 在 C 语言调试时可直接跳到 main 函数。



4 Utilities 设置

配置 Utilities, 1 处选择 “Use Target Driver for Flash Programming” 单选按钮和 “Sinomcu 8051 SNLinkS1 Driver”; 2 处 “Use Debug Driver” 和 “Update Target before Debugging” 项为必选项。按 OK 完成配置。

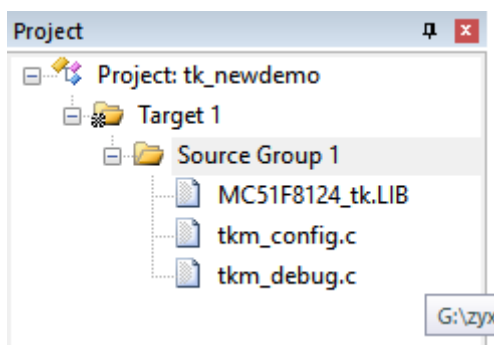


6.5.2 移植触摸库资源

- 1 将 Include 文件夹下除 main.h、Source\tk 文件夹下所有文件移植到新工程文件夹中，如下图红框中标注所示。

common.h	2020/10/23 13:22	C/C++ Header	36 KB	
gpio.h	2020/9/21 15:33	C/C++ Header	9 KB	
mc51f.h	2020/9/21 17:08	C/C++ Header	1 KB	
MC51F8124 tk.LIB	2020/12/2 18:49	Object File Library	9 KB	
tk_newdemo.uvproj	2020/12/3 15:15	Microvision4 Project	0 KB	新建工程文件名
tkm_config.c	2020/11/26 9:39	C Source	8 KB	
tkm_config.h	2020/12/2 19:05	C/C++ Header	10 KB	
tkm_debug.c	2020/11/26 9:39	C Source	11 KB	
tkm_debug.h	2020/11/26 9:39	C/C++ Header	5 KB	

- 2 将工程下.c 和 tk.LIB 添加到工程中，如下图所示。



- 3 建立 main.c 和 main.h 文件，并在 main.h 中包含触摸库相关头文件，如下图所示。

```

/*
 * -----
 * Copyright (C) 2019 SINOMCU
 * All Rights Reserved
 * -----
 * filename: main.c
 * version:  v1.0.0
 * author:   zyx and yj
 * date:    2020/09/21
 * The file is only for SINOMCU MC51F8124 product.
 * -----
 */

/*****
 * Includes
 *****/
#include "mc51f.h"
#include "tkm_config.h"
#include "tkm_debug.h"
#include "main.h"

/*****
 * Macros and Definitions
 *****/

```

- 4 新工程 main.c 中需要包含如下图所示函数。

- 1 TKInit()函数，进行 TK 模块初始化操作。
- 2 Tk_service()函数，对 TK 通过算法处理进行按键的触发判断。
- 3 App_Process()函数，对触发的 TK 进行响应，例如点灯或者蜂鸣器鸣叫等。

```

/*****
函数名: void main(void)
描 述: 主函数
输入值: 无
输出值: 无
*****/
void main(void)
{
    #if TK_Lowpowermode == ON
        Timer1_Init();
    #endif
    TKInit();
    while (1)
    {
        #if WDT_FUNCTION == ON
            CLR_WDT;
        #endif
        Tk_service(ENABLE);
    }
}

```


- 5 将 TKIntProcess()放置到 TK 中断服务子程序中如下图所示，进行获取 TK 模块采样值。

```

/*****
函数名: void tk_interrupt(void)
描 述: TK中断服务子程序
输入值: 无
输出值: 无
*****/
void tk_interrupt(void) interrupt TK_IRQn
{
    TKIntProcess();
}
    
```

- 6 若用户需要通过 TK 库与上位机工具通讯进行调试,则可在 Uart 服务子程序中添加 ReceivePCData()和 SendPCData()如下图所示。

```

/*****
函数名: void uart0_interrupt(void)
描 述: UART0中断服务子程序
输入值: 无
输出值: 无
*****/
void uart0_interrupt(void) interrupt UART0_IRQn
{
    #if UART_PC_COMMUNICATEFUNCTION == ON
        ReceivePCData();
        SendPCData();
    #endif
}
    
```

7 触摸库详细介绍

7.1 tkm_config.h 配置文件说明

7.1.1 按键通道设定

```
1  #define TK_NUM                6    // 触控按键个数 1--16
    选择使用的 TK 个数（本文档中使用 6 个触摸按键）。
2  #define TK_Channel0          TK15  // TK00--TK15 对应 TK 引脚
    #define TK_Channel1          TK06
    #define TK_Channel2          TK05
    #define TK_Channel3          TK04
    #define TK_Channel4          TK11
    #define TK_Channel5          TK14
    #define TK_Channel6
    #define TK_Channel7
    #define TK_Channel8
    #define TK_Channel9
    #define TK_Channel10
    #define TK_Channel11
    #define TK_Channel12
    #define TK_Channel13
    #define TK_Channel14
```

根据目标板实际原理图选择要使用的 TK 通道。（本文档配置使用了 TK15、TK06、TK05、TK04、TK11 以及 TK14 共 6 个通道，对于要使用的 TK 通道对应的芯片引脚，请参见 MC51F8124 用户手册）。

7.1.2 通用触控参数设定

```
1  按键阈值设定
    #define TK_Threshold_Channel0 145 //按键阈值，推荐为手指按下前后的滤波值的差值 50%~80%
    #define TK_Threshold_Channel1 130 //
    #define TK_Threshold_Channel2 145 //
    #define TK_Threshold_Channel3 145 //
    #define TK_Threshold_Channel4 130 //
    #define TK_Threshold_Channel5 130 //
    #define TK_Threshold_Channel6 75
    #define TK_Threshold_Channel7 75
    #define TK_Threshold_Channel8 75
    #define TK_Threshold_Channel9 75
    #define TK_Threshold_Channel10 75
    #define TK_Threshold_Channel11 75
    #define TK_Threshold_Channel12 75
    #define TK_Threshold_Channel13 75
```

```
#define TK_Threshold_Channel14 75
```

定义各通道的门限，门限设定原则为最终产品手触摸滤波值变化量（差值）的 50%~80%。

2 采样次数累加设定

```
#define TK_Samples_perscan 4 // 采样次数累加设定，推荐累加后数值大于 5500
```

1. 累加设定后 TK 采样结果不能超过 32768（int 型数据最大值），否则请减小此数据。
2. 通过采样次数累加和触摸放大系数控制寄存器的设置对采样数据进行放大（由于某些硬件环境下会导致滤波值和基准值差值过小，导致无法设置合理手指阈值，所以我们需要对数据进行放大）。

3 基准值更新次数

```
#define TK_BaseSamples_perscan 60 // 基准值更新次数，推荐(30~120)，可设置(1~255)
```

基准值自动修正以适应环境，数值越大，基准值修正速度越慢。

4 按键按下灵敏度设定

```
#define TK_Threshold_press 9 // 8-12 按键按下灵敏度设定
```

按键响应灵敏度调节，越小，判断按键触发越灵敏。

5 按键松开灵敏度设定

```
#define TK_Threshold_release 11 // 8-12 按键松开灵敏度设定
```

按键离开灵敏度调节，越大，判断按键离开越灵敏。

6 连续按下的触发次数设定

```
#define TK_Debounce_press 3 // 连续按下的触发次数设定，推荐(3~5),可设置(1~10)
```

次数越多，按键的抗干扰能力越强。

7 连续离开的触发次数设定

```
#define TK_Debounce_release 1 // 连续离开的触发次数设定，推荐(3~5),可设置(1~10)
```

次数越多，按键的抗干扰能力越强。

8 滤波系数增量

```
#define TK_Filter_1st_Increase 48 // 滤波系数增量，推荐(4~32),可设置 (1~255)
```

越大，滤波值滤波幅度越小。

9 基准系数增量

```
#define TK_Average_1st_Increase 8 // 基准系数增量，推荐(4~32),可设置 (1~255)
```

越大，基准值滤波幅度越小。

7.1.3 寄存器设定

如下图所示为TK 寄存器参数设定。详细说明请查阅 MC51F8124 用户手册。

```

/*TouchKe寄存器配置=====开始=====
/* TK模块使能配置 */
#define TK_MODULE_EN                0x80
#define TK_MODULE_DIS              0x00

/* TK 模块低功耗模式使能配置 */
#define TK_MODULE_LOWPOWERMODE_EN  0x40
#define TK_MODULE_LOWPOWERMODE_DIS 0x00

/* TK 模块触摸扫描使能配置 */
#define TK_MODULE_SCAN_EN          0x20
#define TK_MODULE_SCAN_DIS        0x00

/* TK 所有被使用的通道内部并联控制配置 */
#define TK_MODULE_PARALLELCONTROL_EN 0x10 // 并联仅针对被选中的通道,未选
#define TK_MODULE_PARALLELCONTROL_DIS 0x00

/* TK 模块扫描模式控制位配置 */
#define TK_MODULE_CHARGENUMBER_EN  0x00
#define TK_MODULE_CHARGETIME_EN    0x08

/* TK 模块输出电压控制位配置 */
#define TK_MODULE_OUTPUTVOLTAGE_OP_EN 0x00
#define TK_MODULE_OUTPUTVOLTAGE_VDD_EN 0x04

/* TK 模块采样次数配置 */
#define TK_MODULE_SAMPLING_ONCE    0x00
#define TK_MODULE_SAMPLING_THREETIMES 0x01
#define TK_MODULE_SAMPLING_SIXTIMES 0x02
#define TK_MODULE_SAMPLING_TENTIMES 0x03

/* TK 低功耗模式触发频率配置 */
#define TK_MODULE_LOWPOWER_TRIG_64MS 0x00
#define TK_MODULE_LOWPOWER_TRIG_128MS 0x40
#define TK_MODULE_LOWPOWER_TRIG_256MS 0x80
#define TK_MODULE_LOWPOWER_TRIG_512MS 0xc0

/* TK 工作频率配置 */
#define TK_MODULE_SWEEPFREQUENCY    0x02 // 范围: 0x00~0x3f,以1为单位进

/* TK 模块随机频率使能配置 */
#define TK_MODULE_RANDOM_EN         0x80
#define TK_MODULE_RANDOM_DIS        0x00

/* TK 模块比较阈值方向控制配置 */
#define TK_MODULE_COMPARE_GREATERTHAN_THRESHOLD 0x00
#define TK_MODULE_COMPARE_EQUALORLESSTHAN_THRESHOLD 0x40

/* TK 模块随机频率配置 */
#define TK_MODULE_RANDOMJITTER_1    0x00
#define TK_MODULE_RANDOMJITTER_1_2  0x10
#define TK_MODULE_RANDOMJITTER_1_2_3 0x20

```

7.2 函数及常用变量说明

7.2.1 函数功能说明

序号	函数名	函数功能
1	void TKInit(void)	TK 初始化子程序
2	void TK_Config_Init(ulong amplificationcoefficient)	TK 配置参数初始化程序（用户不可访问）
3	void TK_Interrupt_EN(uchar state)	TK 及总中断初始化
4	void Tk_getbaselinedata(void)	上电获取触摸按键初始的滤波值和基准值(调用后会强行更新滤波值和基准值，用户不可访问)
5	void Tk_service(uchar update_basevalue)	TK 按键滤波处理（用户不可访问） ENABLE: TK 触发效果随时间加长会消失 DISABLE: TK 一直保持触发效果不消失
6	void TKIntProcess(void)	扫描 TK 通道，获取 TK 采样值（用户不可访问）
7	void Customer_Keystatejustpress(void)	触控任意通道被刚刚检测到按下状态
8	void Customer_Keystatejustrelease(void)	触控任意通道被刚刚检测到释放状态
9	void Customer_Tkdataready(void)	TK 数据刚刚完成一次更新
10	void Customer_TKdatareadytoupdate(void)	数据刚刚采样完毕，准备对采样数据进行滤波处理
11	void Customer_wakeupfromsleepmode_init(void)	唤醒后的初始化程序 可以用来恢复正常工作时状态初始化
12	void App_Process(void)	根据触发的按键进行响应操作
13	void ReceivePCData(void)	1. 接收上位机工具发送数据（用户需按上述说明将此函数放至指定位置） 2. 通过将 tkm_debug.h 文件中 UART_PC_COMMUNICATEFUNCTION 中宏体 OFF->ON 后调用此函数开启串口通讯功能， 占用 UART 资源
14	void SendPCData(void)	1. 向上位机工具发送数据（用户需按上述说明将此函数放至指定位置） 2. 通过将 tkm_debug.h 文件中 UART_PC_COMMUNICATEFUNCTION 中宏体 OFF->ON 后调用此函数开启串口通讯功能， 占用 UART 资源

7.2.2 常用变量使用说明

1 采样数据

采样数据: `TK_value_origin[TK_NUM] = {0};`

采集 TK 通道上未经处理的数据。

2 滤波数据

滤波数据: `TK_value_filter[TK_NUM] = {0};`

对采样数据进行软件滤波算法后的数据。

3 基线数据

基线数据: `TK_value_average[TK_NUM] = {0};`

对滤波数据进行处理后的值, 作为是否产生按键事件的基准值。

注: `TK_value_average` 与 `TK_value_filter` 对应的按键通道差值大于设置的按键阈值即判定按键触发。

4 TK 状态变量

TK 状态变量: `TK_state`

根据用户选择的 TK 个数, `TK_state` 从 LSB 开始每一个 bit 代表一个选择的 TK 通道状态:

1: TK 触发

0: TK 处于松开状态

7.3 TK 库调试步骤说明

1 配置按键个数 (TK_NUM)

如图所示在 tkm_config.h 中红框标注设置使用的 TK 个数。

```

] /*****
 * Includes
 *****/
#include "mc51f.h"

] /*****
 * Public Macros and struct
 *****/
/*TouchKe算法参数配置=====开始=====
#define TK_NUM                6        // TK个数1--16

#define TK_Channel0            TK15    // TK00--TK15 对应TK引脚
#define TK_Channel1            TK06
#define TK_Channel2            TK05
#define TK_Channel3            TK04
#define TK_Channel4            TK11
#define TK_Channel5            TK14
#define TK_Channel6

```

2 配置使用的触摸按键通道 (TK_Channel(n),n=0~15)

如下图所示，根据 MC51F8124 用户手册设置使用的触控通道，如图本例中使用的第一个触控通道为 TK15 代表的管脚。

```

#define TK_Channel0            TK15    // TK00--TK15 对应TK引脚
#define TK_Channel1            TK06
#define TK_Channel2            TK05
#define TK_Channel3            TK04
#define TK_Channel4            TK11
#define TK_Channel5            TK14
#define TK_Channel6
#define TK_Channel7
#define TK_Channel8
#define TK_Channel9
#define TK_Channel10
#define TK_Channel11
#define TK_Channel12
#define TK_Channel13
#define TK_Channel14
#define TK_Channel15

```

3 配置按键采样次数 (TK_Samples_perscan)，可与触摸放大系数控制寄存器配合调试

推荐所有 TK 通道的采样值至少在 5500 以上。用户可根据按键按下前后滤波值变化量进行设置，若按键前后滤波值变化较小（例滤波值变化不超过 100），可增加按键采样次数。按键按下前后阈值变化越大，按键触发判断越准确。

4 配置按键阈值 (TK_Threshold_Channel(n),n=0~15)

设置按键阈值的方法如下:

- 1) 记录按键按下前,要设置的对应通道滤波值。
- 2) 记录按键按下后,要设置的对应通道滤波值。
- 3) 取按键按下前后两次滤波值差值的 50%~80%作为对应按键通道按键阈值。

查看 TK 通道滤波值方法有 2 种, 分别如下:

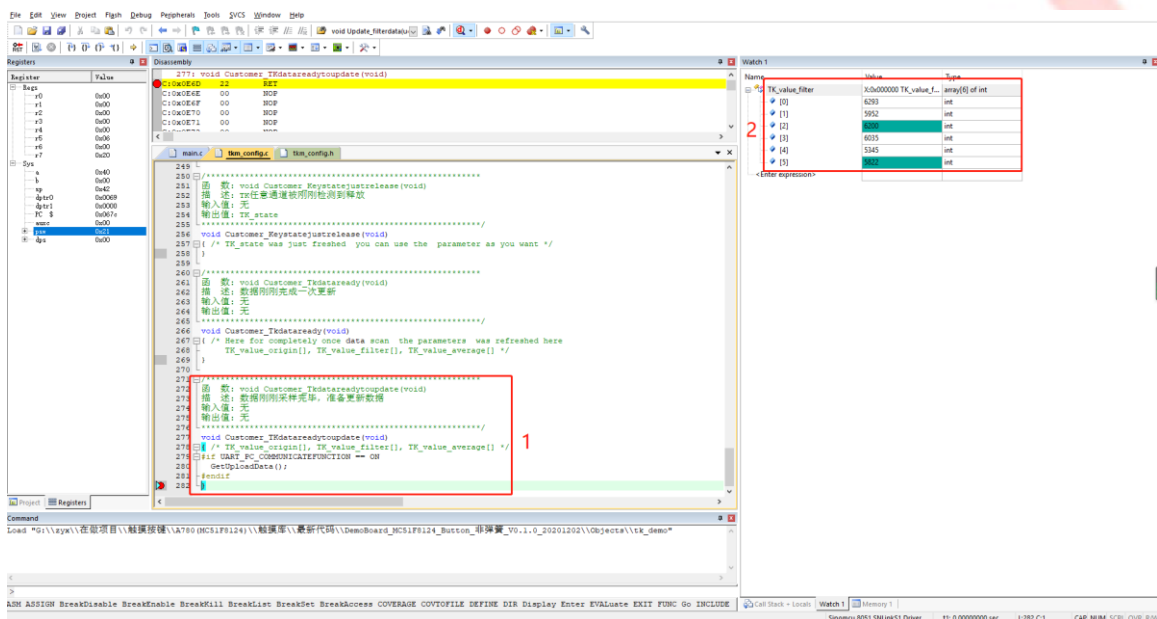
1. 通过我司开发 TouchKey 上位机软件, 将数据更直观地以波形形式显示出来, 具体使用方法, 请参见本文档 § 8~10 章节说明。

2. 进入仿真模式查看 TK 通道滤波值。具体方法如下:

(1) 进入仿真程序模式

为观察按键触发效果以及触摸按键相关数据, 需进入仿真程序模式。

(2) 如下图所示, 在红框中 1 标注函数 void Customer_Tkdatareadytoupdate(void)中设置断点, 查看右图中红框 2 标注按键滤波值 TK_value_filter。



设置后, 重新编译下载后脱机运行程序, 测试手指触摸按键效果。

5 按键触摸频率调节

- 1) TK_Debounce_press (连续按下的触发次数设定, 推荐 (3~5), 可设置 (1~10)): 数值越小, 判键按下越快
- 2) TK_Debounce_release (连续离开的触发次数设定, 推荐 (3~5), 可设置 (1~10)): 数值越小, 判键离开越快
- 3) TK_Filter_lst_Increase (滤波系数增量, 推荐 (4~32), 可设置 (1~255)): 数值越大, 判键频率越快, 涉及滤波, 设置后需测试

6 按键触摸力度调节

- 1) TK_Threshold_press (8-12 按键按下灵敏度设定): 数值越小按键越灵敏
- 2) TK_Threshold_release (8-12 按键松开灵敏度设定): 数值越大按键越灵敏

7 按键长按效果调节

TK_BaseSamples_perscan (基准值更新次数, 推荐 (30~120), 可设置 (1~255)): 数值越大, 长按效果越好, 同时下次判断按键按下时间会加长

8 低功耗模式调试步骤

打开低功耗模式开关 (TK_Lowpowermode)

```
#define TK_Lowpowermode      ON    // ON:开启低功耗模式 OFF:关闭低功耗模式
```

设定常态模式下 TK 运行时间

```
#define TK_NORMALMODE_RUNTIME 1000    // 定时时间: n * 5ms, n 范围: 1~65535 常态模式运行指定的时间, 无按键触发, 进入低功耗模式, 占用 timer1 资源
```

设定低功耗模式下使用的 TK 通道

```
#define TK_Lowpower_TK_NUM    2        // 低功耗时使用的 TK 通道个数, 建议不要超过 2 个
```

设定低功耗模式时唤醒的手指阈值

```
#define TK_Lowpower_Threshold 12       // 低功耗时触发的手指阈值
```

降低功耗调节

- (1) 减少低功耗模式使用的 TK 通道。
- (2) 降低低功耗模式下 TK 扫描的触发频率。通过设置 TK_Reginit_table 数组设置低功耗触发频率。如下图所示表明设置为 LIRC/4096。

```
//
uchar xdata TK_Reginit_table[] = { TK_NUM,
TK_MODULE_EN,
TK_MODULE_SCAN_EN,
TK_MODULE_PARALLELCONTROL_DIS,
TK_MODULE_CHARGENUMBER_EN,
TK_MODULE_OUTPUTVOLTAGE_OP_EN,
TK_MODULE_SAMPLING_ONCE,
TK_MODULE_LOWPOWER_TRIG 4096 DEVBY LIRC,
TK_MODULE_SWEEPFREQUENCY,
TK_MODULE_RANDOM_DIS,
TK_MODULE_COMPARE_GREATERTHAN_THRESHOLD,
TK_MODULE_RANDOMJITTER_1,
TK_MODULE_EXTERNALCAPACITOR_EN,
TK_MODULE_WAVEFORMCOMPENSATION_DIS,
TK_MODULE_WAVEFORMCOMPENSATION_VDD,
TK_MODULE_WAVEFORMCOMPENSATION_WEAK,
TK_MODULE_COMPARER_WORKMODE_0,
TK_MODULE_CHARGECURRENT_DIS,
TK_MODULE_SWEEPRESISTANCE_0,
TK_MODULE_VREF_2P0V,
TK_MODULE_DEBOUNCETIME_32_DEVBY_FSYS,
TK_MODULE_OP_2P5V,
TK_MODULE_DISCHARGETIME_256_DEVBY_FSYS };
```

测试功耗的步骤

- (1) 建议选择专用测功耗的数字万用表串接芯片的 VCC 和 GND，通常的万用表只能测量功耗的大概范围
- (2) 确保芯片的 TK 处于松开即未触发状态
- (3) 等待一段时间，观察电子万用表功耗保持在相对微小的个位数 μA 级变化范围

8 TouchKey 上位机简介

我司提供了专门的触控按键电脑界面软件 SinoMCU TK Debugger，方便用户通过一系列的人机交互完成调试工作。用户通过此软件配合 MC51F8114 芯片开发的触摸库，可迅速直观地找到用户 PCB 最合适的触控按键参数，简化对触控按键的开发工作的难度。为方便用户更好使用我司 SinoMCU TK Debugger 软件，本文将对 SinoMCU TK Debugger 软件使用进行简要说明。

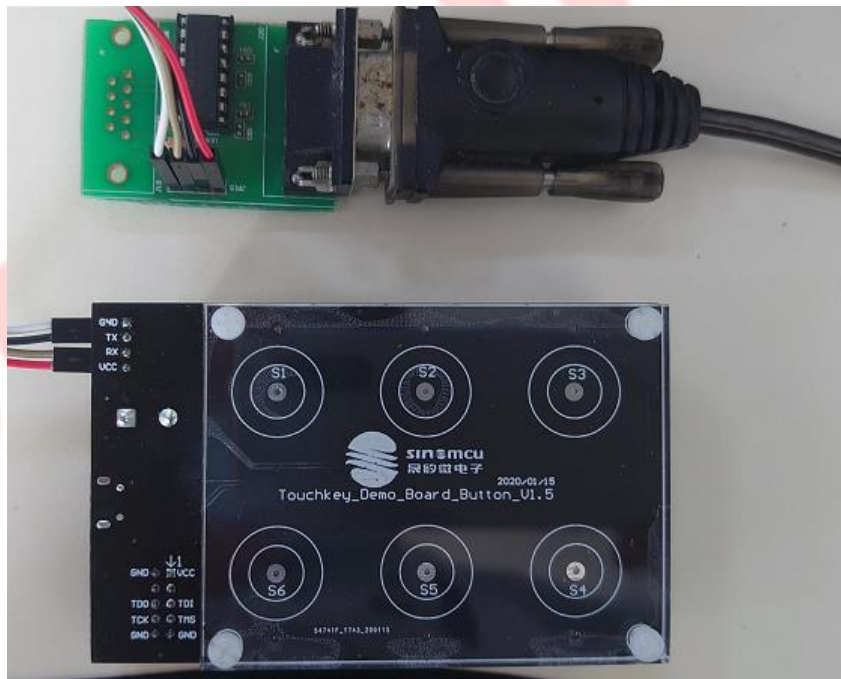
9 软件运行环境

9.1 硬件连接

1 连接 demo 板与串口通讯工具，连接关系如下：

- (1) demo 板 VCC 连接串口工具 VCC；
- (2) demo 板 RX 连接串口工具发送端；
- (3) demo 板 TX 连接串口工具接收端；
- (4) demo 板 GND 连接串口工具 GND。

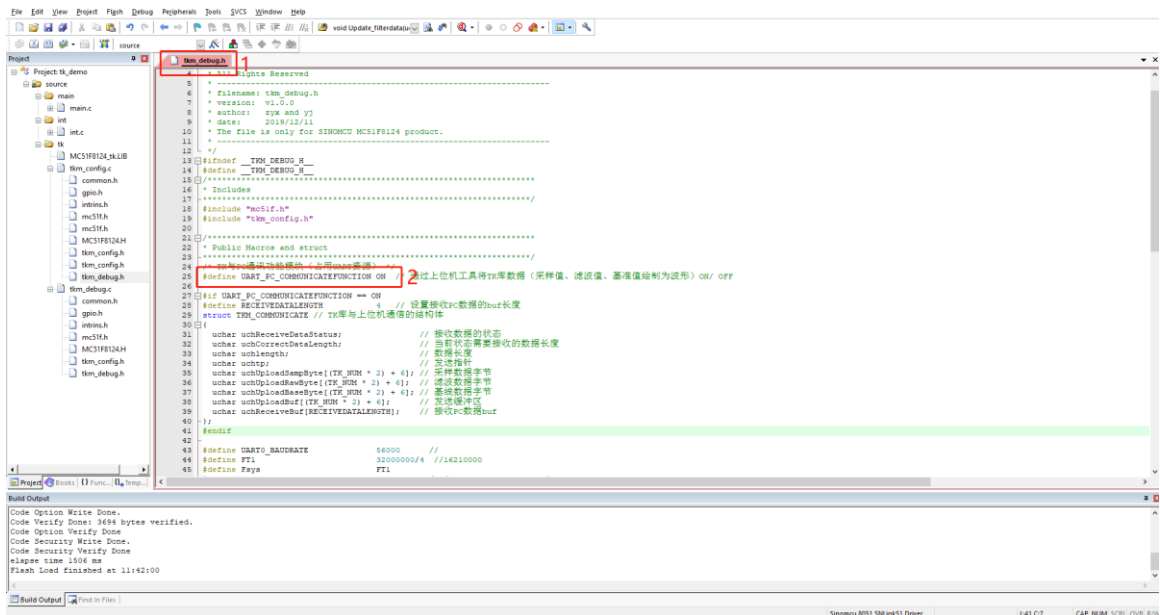
然后将串口通讯工具连接到电脑上。连接图示如下。



2 通过 micro usb 线给 demo 板供电。

9.2 软件运行

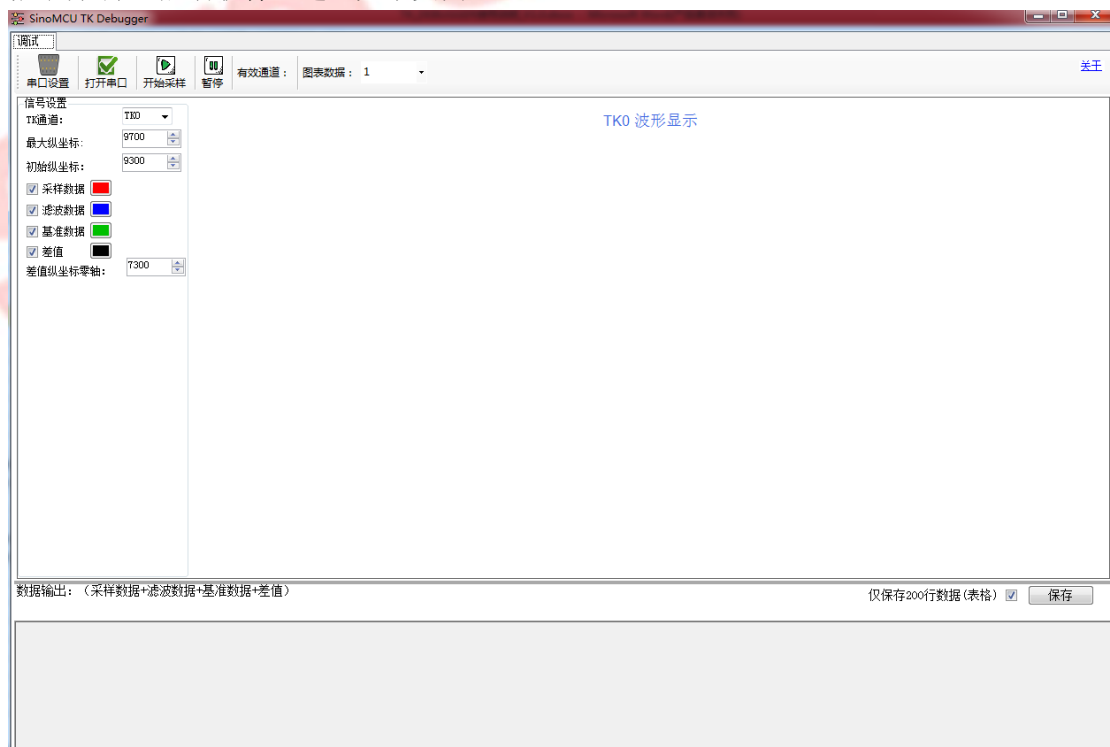
- ◆ 支持的操作系统：Win XP 及以上版本；
- ◆ CPU 主频：1.6GHz，推荐 2.2GHz 以上；
- ◆ PC 软件环境：Net Framework4.0。
- ◆ demo 板程序环境：在我司 demo 工程中打开 tkm_debug.h 文件，打开上下位机通讯开关如下图红框中 2 标注所示。



编译并下载程序到 demo 板中。

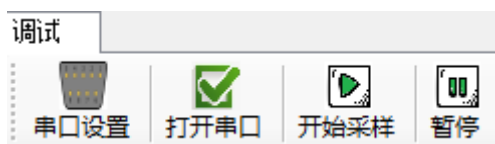
本软件无需安装，直接点击 TK_Debugger.exe 运行应用程序启动软件。

双击应用程序，启动软件，进入如下页面。



10 软件使用

调试页面有串口设置及调试控制几个按钮。

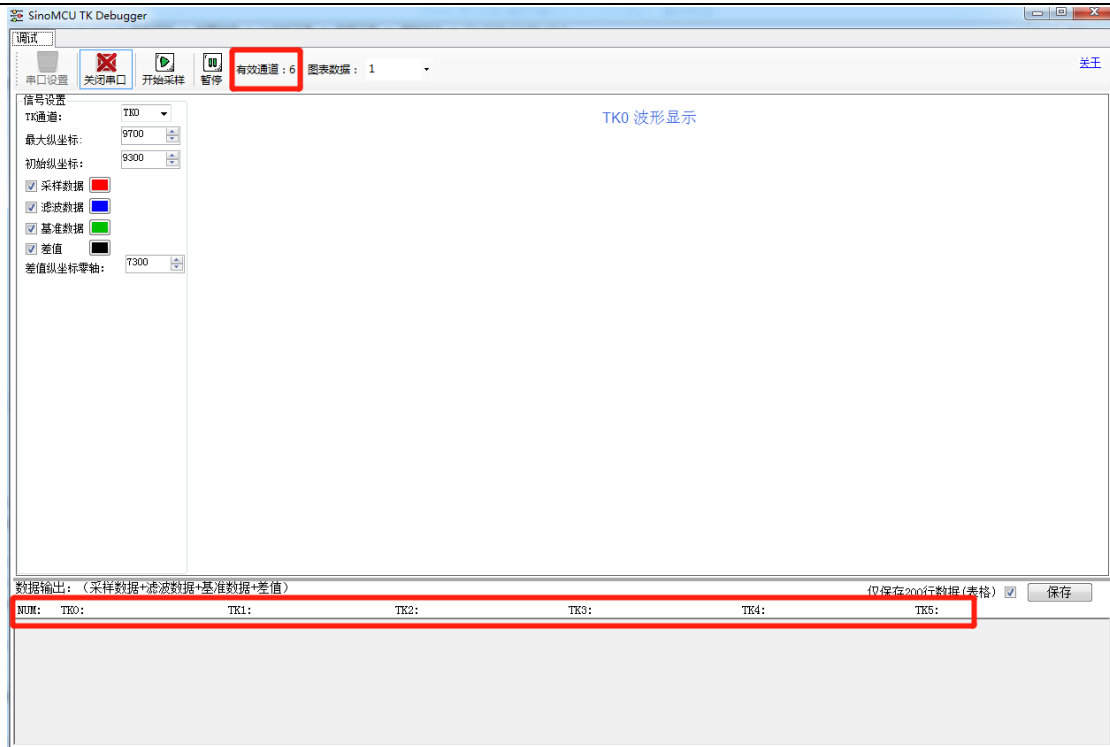


10.1 串口设置

进行调试前先要配置串口。点击“串口设置”按钮打开串口设置窗口，依次设置串口名、波特率、数据位、校验位和停止位，点击确定。再点击“打开串口”按钮以打开串口。串口打开后无法再配置串口，必须先关闭已打开的串口。



“打开串口”成功后将显示用户所配置 TK 通道。因 demo 板中配置使用 6 个 TK 通道，因此显示下图中红框中标注显示有效通道数：6，以及 TK0~TK5 等 6 个有效通道。



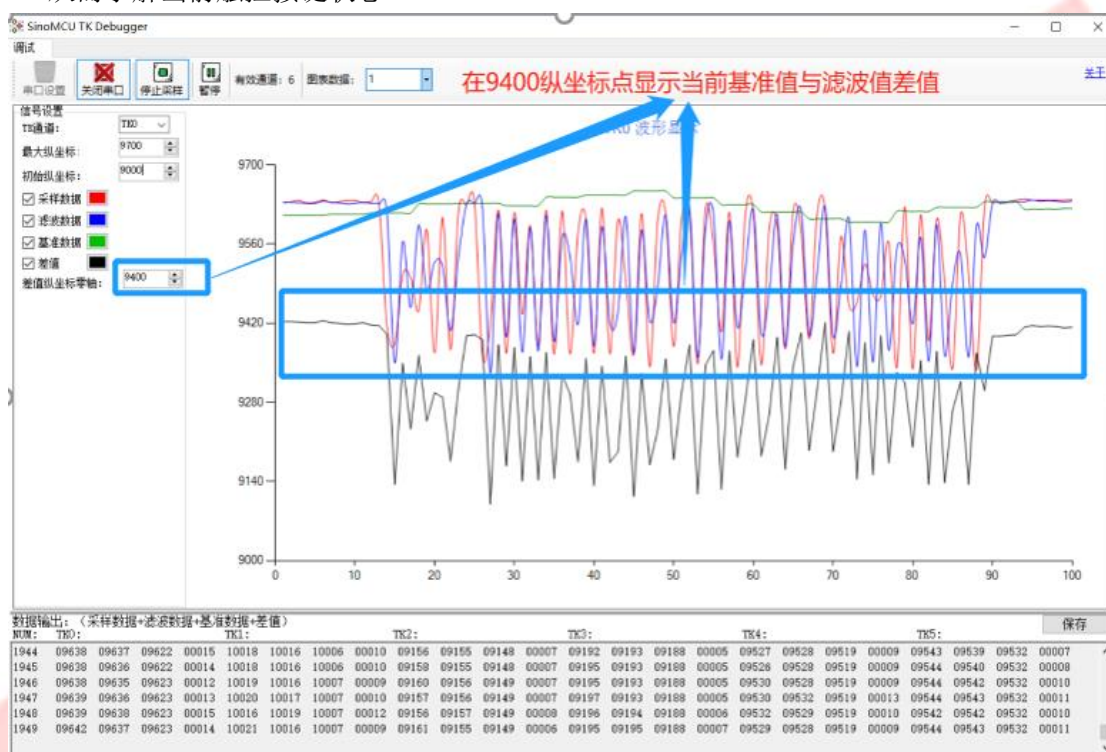
10.2 开始采样

点击“开始采样”后，软件将与固件持续进行通讯，读取当前触控通道状态，并以图表形式显示。调试时可配置显示不同图表数量：主界面显示的图表数量，支持 1、2、3、4、6、9 等不同个数图表。设置后，相应数量的图表将排列在主界面中，每个图表可以分别显示不同的数据。如下图所示显示 6 个图表。



10.3 图表显示设置

- ◆ **TK 通道：**选择当前需要显示的 TK 通道。可选通道由“打开串口”时，由软件向固件查询而来。
- ◆ **最大纵坐标：**显示数据的上限值。
- ◆ **初始纵坐标：**显示数据的下限数值。可通过最大纵坐标和初始纵坐标放大或缩小当前显示波形。
- ◆ **采样数据：**显示触控通道中的采样值。
- ◆ **滤波数据：**显示触控通道的滤波值。
- ◆ **基准数据：**显示触控通道的基准值。
- ◆ **差值：**显示当前触控通道基准值与滤波值的差值。
- ◆ **差值纵坐标零轴：**如下图所示，通过调节此参数，可以更直观的观察基准值与滤波值差值曲线，从而了解当前触控按键状态。

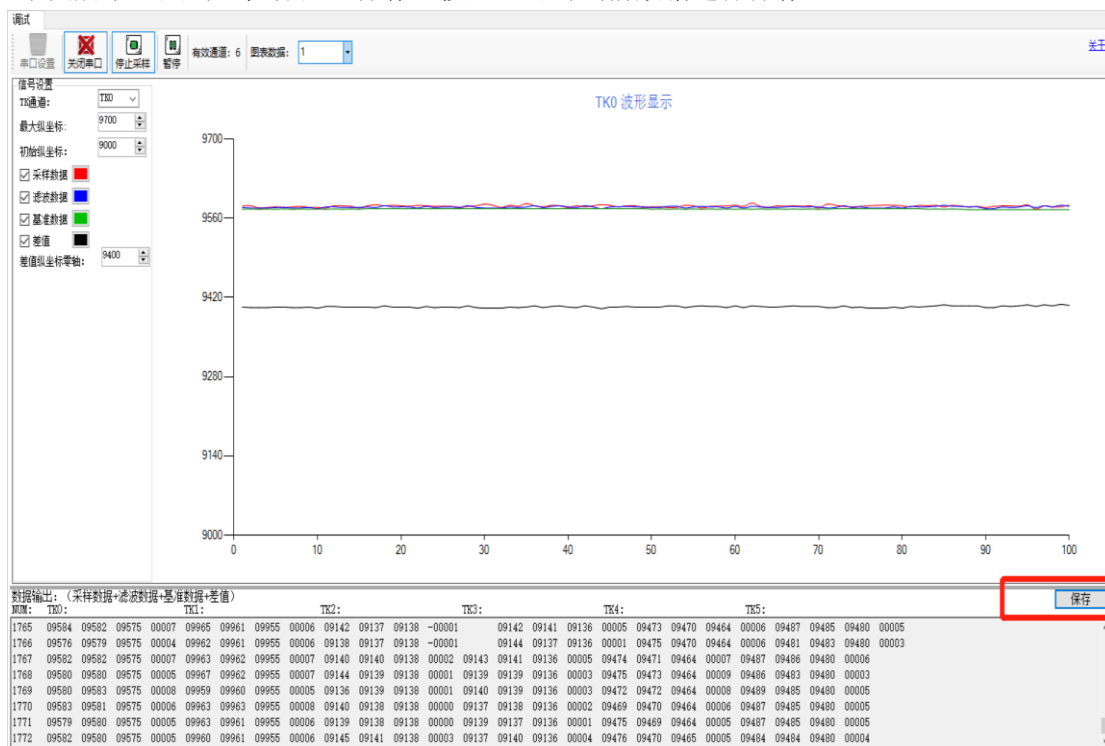


用户可调节图表中的采样数据、滤波数据、基准数据以及差值绘制曲线的颜色，从而设置便于用户观察的绘制曲线。

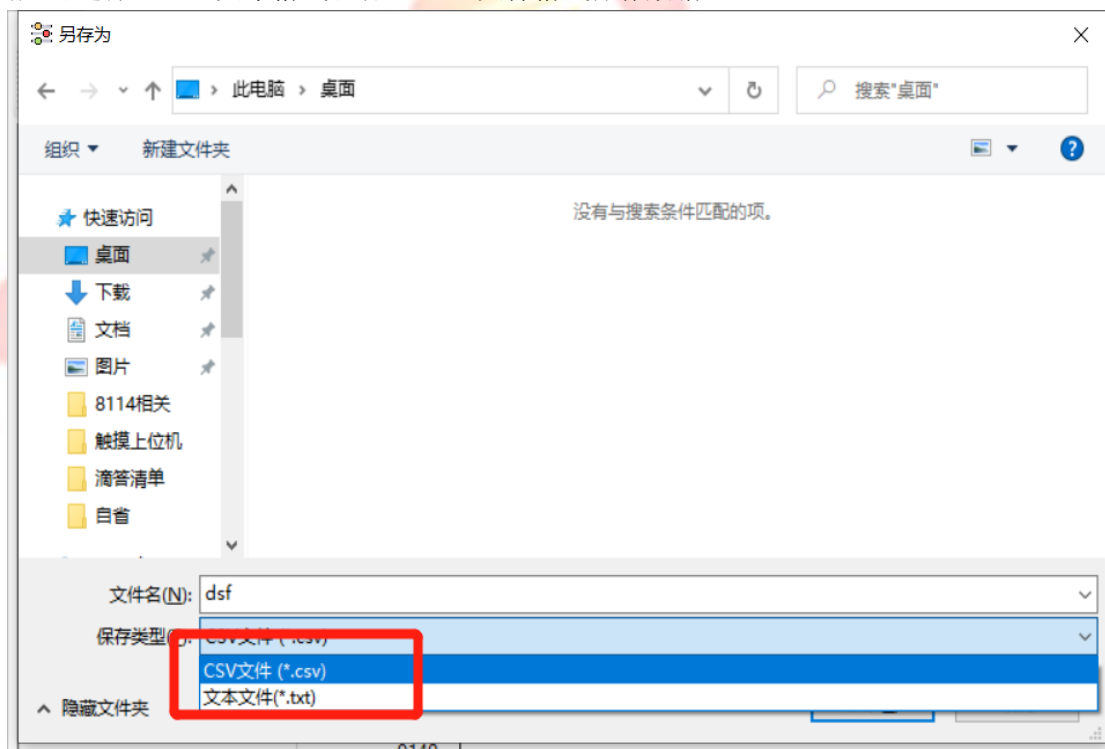
启动后用户可以选择“停止采样”或“暂停”。暂停后再恢复，之前的数据保留；停止后再启动，之前的数据将清除。

10.4 数据保存

如下图所示，点击红框标注“保存”按钮，可对当前数据进行保存。



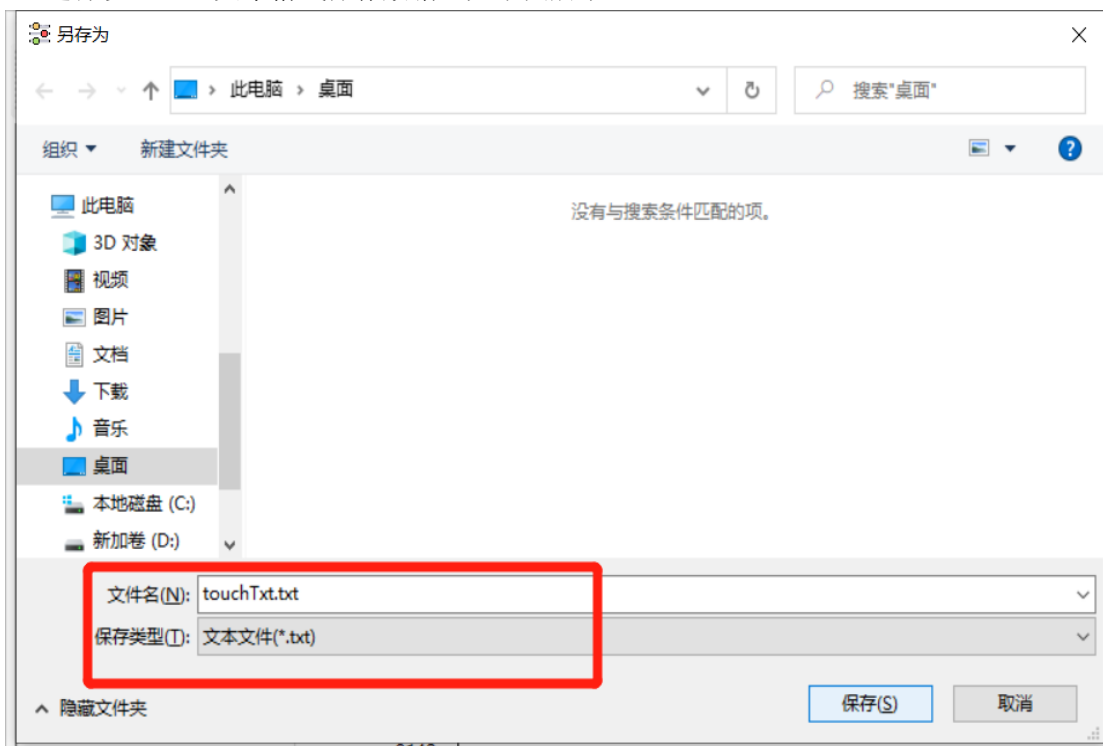
用户可选择“txt”文本格式或者“csv”文件格式保存数据。



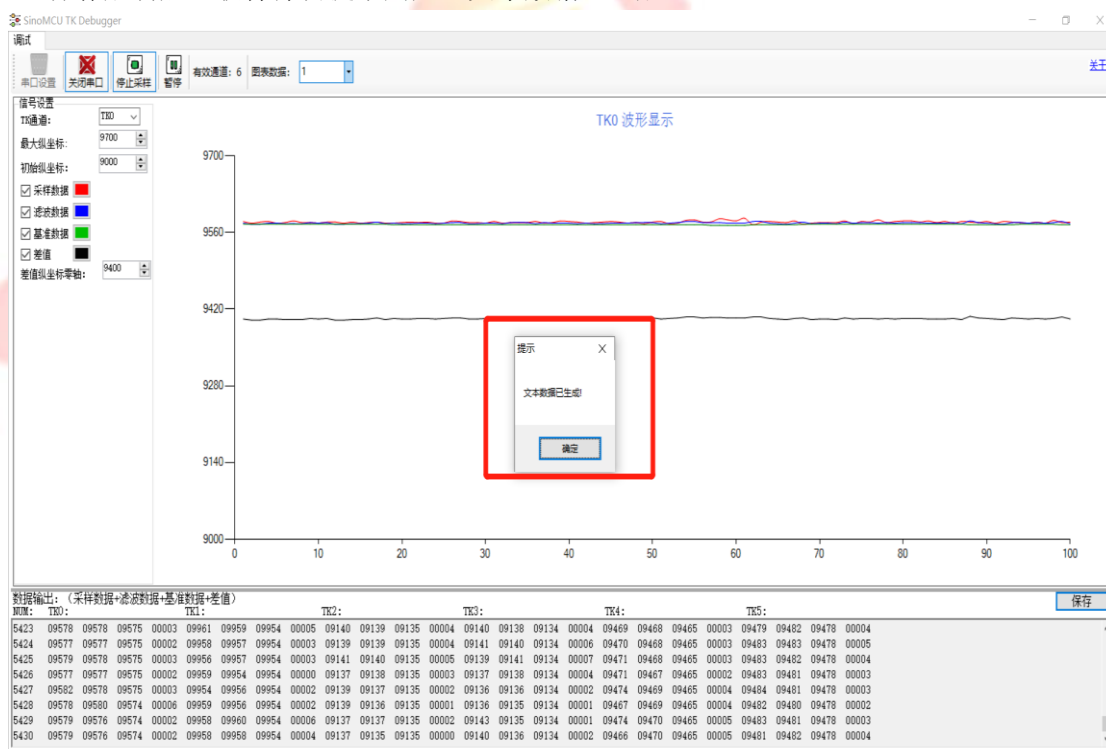
下面将对两种保存数据格式分别介绍。

10.4.1 “txt” 文本格式

1. 选择以“txt”文本格式保存数据。如下图所示。



2. 保存成功后，软件弹窗提示用户“文本数据已生成”。



3. 打开文件，保存内容如下图所示。

touchTk4.csv

文件(F) 编辑(E) 格式(O) 视图(V) 帮助(H)

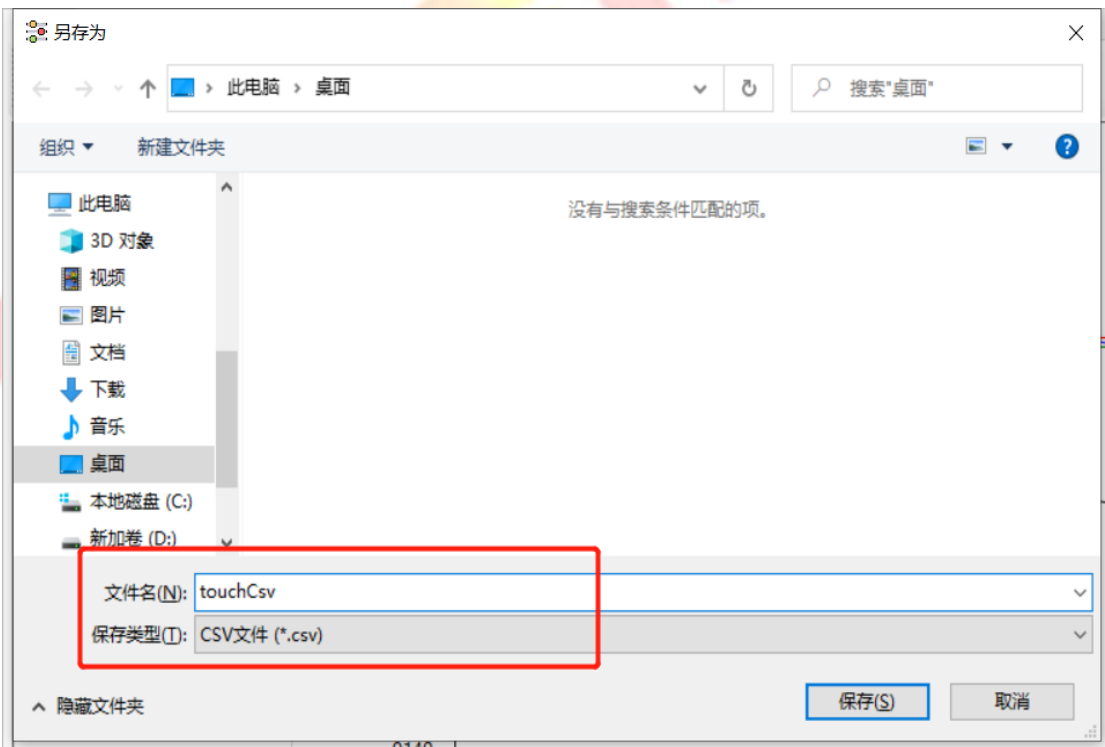
NUM: TK1 TK2 TK3 TK4 触控通道号

NUM:	TK1	TK2	TK3	TK4	触控通道号
0	09115	09115	09110	00005	10001
1	09120	09117	09110	00007	09996
2	09120	09118	09110	00008	09996
3	09115	09116	09110	00006	10000
4	09115	09116	09110	00006	10000
5	09120	09113	09110	00003	09994
6	09119	09117	09110	00007	09995
7	09118	09116	09110	00006	10002
8	09115	09117	09110	00007	09999
9	09114	09117	09110	00007	09995
10	09119	09114	09110	00004	10000
11	09119	09115	09110	00005	09996
12	09117	09116	09110	00006	09994
13	09119	09116	09110	00006	09996
14	09113	09117	09110	00007	09995
15	09120	09117	09110	00007	09997
16	09120	09118	09110	00008	10002
17	09119	09119	09110	00009	09999
18	09118	09115	09110	00005	09999
19	09120	09116	09110	00006	09998
20	09121	09118	09110	00008	10002
21	09119	09116	09111	00005	10001
22	09117	09118	09111	00007	09998
23	09116	09116	09111	00005	09998
24	09117	09116	09111	00005	09999
25	09115	09116	09111	00005	09999
26	09120	09116	09111	00005	09999
27	09122	09115	09111	00004	09998
28	09117	09115	09111	00004	09997
29	09116	09117	09111	00006	09999
30	09116	09114	09112	00002	09999
31	09117	09116	09112	00004	09994
32	09118	09117	09112	00005	09998

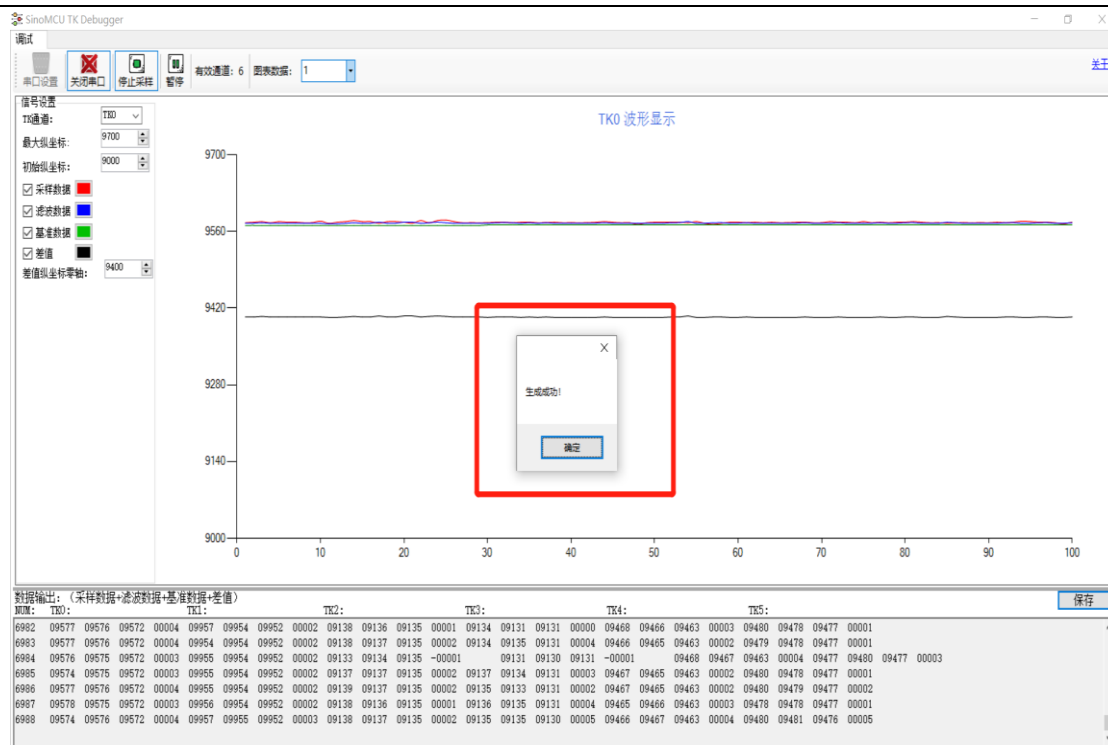
第1行, 第1列 100% Windows (CRLF) UTF-8

10.4.2 “csv” 文件格式

1. 选择以“csv”文件格式保存数据。如下图所示。



2. 保存成功后，软件弹窗提示用户数据“生成成功”。



3. 打开文件后，保存内容如下图所示。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	TK0采样数	TK0滤波数	TK0基准数	TK0差值		TK1采样数	TK1滤波数	TK1基准数	TK1差值		TK2采样数	TK2滤波数	TK2基准数	TK2差值		TK3采样数	TK3滤波数	TK3基准数	TK3差值		TK4采样数	TK4滤波数	TK4基准数
2	9615	9615	9610	5		10001	9998	9993	5		9174	9170	9171	-1		9177	9174	9171	3		9510	9506	9501
3	9620	9617	9610	7		9996	9996	9993	3		9173	9172	9171	1		9173	9170	9170	0		9505	9505	9501
4	9620	9618	9610	8		9996	9995	9993	2		9173	9172	9171	1		9173	9170	9170	0		9505	9504	9501
5	9615	9616	9610	6		10000	9995	9993	2		9175	9171	9171	0		9174	9169	9170	-1		9507	9504	9501
6	9615	9616	9610	6		10000	9997	9993	4		9174	9170	9171	-1		9170	9172	9170	2		9506	9506	9501
7	9620	9613	9610	3		9994	9997	9993	4		9169	9173	9171	2		9174	9170	9170	0		9505	9507	9501
8	9619	9617	9610	7		9995	9994	9993	1		9180	9172	9171	1		9176	9170	9170	0		9506	9506	9501
9	9618	9616	9610	6		10002	10000	9993	7		9179	9173	9171	2		9173	9171	9169	2		9506	9508	9501
10	9615	9617	9610	7		9999	9995	9993	2		9172	9171	9171	0		9172	9171	9169	2		9506	9506	9501
11	9614	9617	9610	7		9995	9996	9993	3		9171	9172	9171	1		9170	9171	9169	2		9506	9509	9501
12	9619	9614	9610	4		10000	9996	9993	3		9171	9171	9171	0		9172	9171	9169	2		9506	9509	9501
13	9619	9615	9610	5		9996	9998	9993	5		9177	9170	9171	-1		9173	9171	9169	2		9507	9507	9501
14	9617	9616	9610	6		9994	9995	9993	2		9171	9175	9171	4		9168	9171	9169	2		9506	9504	9501
15	9619	9616	9610	6		9996	9996	9993	3		9175	9173	9171	2		9172	9172	9169	3		9507	9505	9501
16	9613	9617	9610	7		9995	9997	9993	4		9173	9174	9171	3		9170	9169	9169	0		9503	9508	9501
17	9620	9617	9610	7		9997	9996	9993	3		9177	9172	9171	1		9171	9169	9169	0		9508	9507	9501
18	9620	9618	9610	8		10002	9995	9993	2		9173	9173	9171	2		9177	9175	9169	6		9502	9505	9501
19	9619	9619	9610	9		9999	9997	9993	4		9173	9170	9171	-1		9172	9171	9169	2		9508	9506	9501
20	9618	9615	9610	5		9999	9997	9993	4		9175	9170	9171	-1		9173	9169	9169	0		9505	9505	9501
21	9620	9616	9610	6		9998	9996	9993	3		9176	9171	9171	0		9169	9169	9169	0		9507	9505	9501
22	9621	9618	9610	8		10002	9997	9993	4		9174	9173	9171	2		9170	9168	9169	-1		9512	9506	9501
23	9619	9616	9611	5		10001	9996	9993	3		9175	9172	9171	1		9173	9168	9169	-1		9508	9508	9501
24	9617	9618	9611	7		9998	9997	9993	4		9175	9171	9171	0		9172	9170	9169	1		9506	9505	9501
25	9616	9616	9611	5		9998	9998	9993	5		9173	9170	9171	-1		9172	9170	9169	1		9509	9507	9501
26	9617	9616	9611	5		9999	9995	9993	2		9172	9173	9171	2		9174	9173	9169	4		9505	9507	9501
27	9615	9616	9611	5		9999	9996	9993	3		9177	9174	9170	4		9172	9172	9169	3		9511	9504	9501
28	9620	9616	9611	5		9999	9994	9993	1		9175	9172	9170	2		9170	9172	9169	3		9507	9506	9501
29	9622	9615	9611	4		9998	9997	9993	4		9170	9173	9170	3		9173	9172	9169	3		9506	9506	9501
30	9617	9615	9611	4		9997	9995	9993	2		9175	9174	9170	4		9176	9173	9169	4		9509	9504	9501

11 demo 板验证测试

demo 板经过以下测试：

1. EFT 测试
 - (1) 5KHz: -4000V~+4000V 测试;
 - (2) 100KHz: -4000V~+4000V 测试。
2. CS 动态 10V 0.15MHz~80MHz 测试。

12 版本及更新记录

版本号	更新日期	更新内容	人员
V0.1	2021-07-02	初版发布	张玉鑫