GigaDevice Semiconductor Inc.

GD32 TSI TouchKey 软件库使用指南

应用笔记

AN089

1.0 版本

（2023 年 5 月）

目录

目录 2
图索引 3
表索引 4
1. 前言 5
2. 术语 6
3. TouchKey 库 7
3.1. TouchKey 库文件结构 7
3.2. TouchKey 库架构 8
4. TouchKey 库配置 9
4.1. TSI 引脚配置 9
4.2. BANK 配置 9
4.3. TSI 参数配置 10
4.4. TouchKey 参数配置 11
4.5. TouchKey 功能配置 12
5. TouchKey 库使用 13
6. 版本历史 17

图索引

图 3-1. TouchKey 软件库文件结构 7
图 3-2. TouchKey 软件库架构 8
图 4-1. TSI 引脚配置 9
图 4-2. BANK 配置 10
图 4-3. BANK 数组定义 10
图 4-4. 通道优先级定义 10
图 4-5. TSI 参数配置 11
图 4-6. TouchKey 参数配置 12
图 4-7. TouchKey 功能配置 12

表索引

表 6-1. 版本历史 17

1. 前言

触摸传感控制器（TSI）为触摸按键、滑块、电容近距感测等应用提供了简易的解决方案。控制器基于自容式的电荷转移方法，当手指接近电极时会引起整个系统的电容变化。TSI 可以通过电荷转移的方法来检测这种变化，从而感知到手指接近这一行为。

该应用笔记介绍了对于触摸按键（TouchKey）软件库的设计原理以及使用方式，能够帮助开发者快速地使用 TouchKey 软件库进行工程配置和开发。

2. 术语

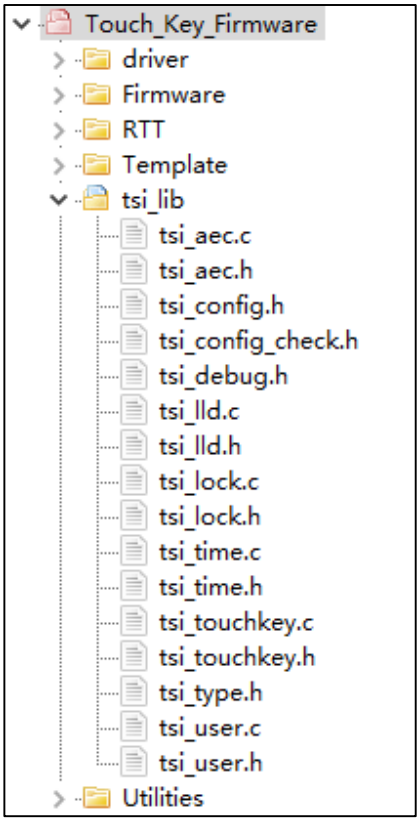
- 自适应环境变化检测（AEC）；
- 检测超时（DTO）；
- 通道锁（LOCK）；
- 噪声滤波器（FILTER）；
- 状态机（SM）；
- 计时管理（TM）；
- 日志输出（LOG）。

3. TouchKey 库

3.1. TouchKey 库文件结构

TouchKey 库文件结构如*图 3-1. TouchKey 软件库文件结构*所示，其中 tsi_lib 包含了 TouchKey 库的主要文件。

图 3-1. TouchKey 软件库文件结构



tsi_aec.c / tsi_aec.h 包括自适应环境变化检测相关代码，其主要实现对通道参考值的周期修正功能。

tsi_lock.c / tsi_lock.h 包含各通道优先级配置及加锁/解锁处理功能。

tsi_time.c / tsi_time.h 包含 TouchKey 库的时间管理功能。

tsi_touchkey.c / tsi_touchkey.h 包括 TouchKey 的配置及检测处理状态机功能。

tsi_user.c / tsi_user.h 包括对 TouchKey 的初始化和处理功能。

tsi_lld.c / tsi_lld.h 包括 TSI 底层硬件初始化及相关的 bank 配置及处理功能。

tsi_config.h 包括对 TSI 参数配置及 TouchKey 参数及功能的宏定义。

tsi_type.h / tsi_debug.h / tsi_config_check.h 包括变量结构体类型声明，调试接口宏定义，配置参数宏检查。

3.2. TouchKey 库架构

TouchKey 的软件库主要分为三层：

- 采集层
- 数据层
- 应用层

TouchKey 的软件库的架构参考*图 3-2. TouchKey 软件库架构*。

采集层：

采集层实现了对各传感器通道的数据采集并将采集到的原始数据传递给数据层作为输入。

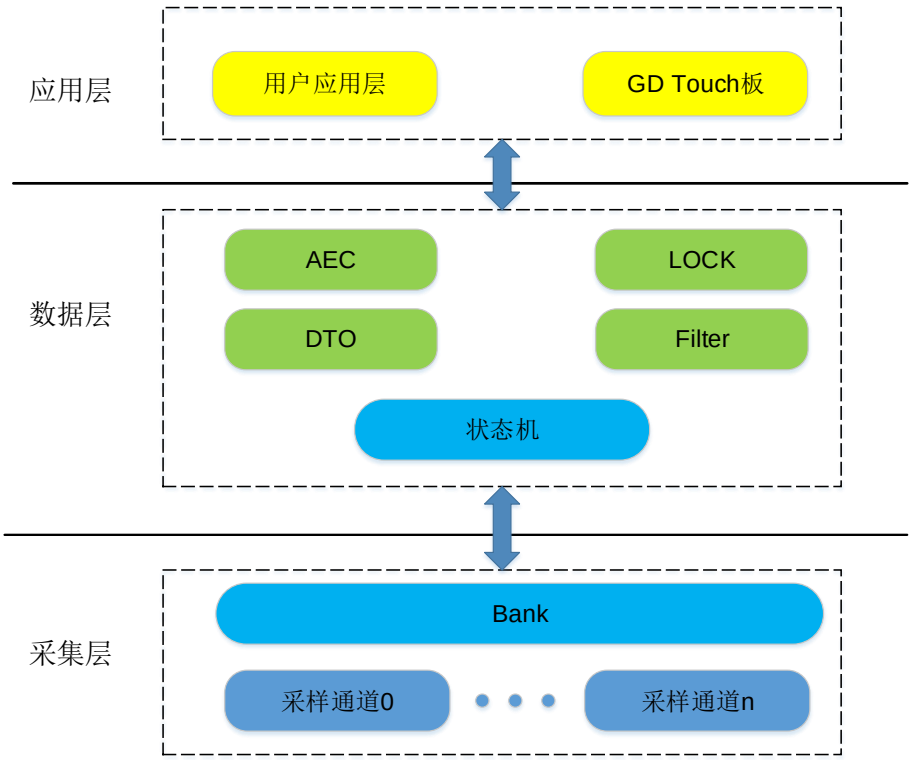
数据层：

数据层是将采集层的原始数据作进一步处理（包括 AEC，LOCK，DTO，Filter），得到各传感器通道的触摸状态作为应用层作输入。

应用层：

应用层是用户根据数据层得到的各传感器通道的触摸状态做的具体逻辑代码实现，以满足特定的触摸应用场景。

图 3-2. TouchKey 软件库架构



4. TouchKey 库配置

4.1. TSI 引脚配置

用户需要根据应用中使用的 TSI 引脚来配置 TSI 的 IO 口模式（可在 tsi_user.h 中进行配置）。用户根据实际引脚模式将其配置为采样引脚（SAMPIN）、通道引脚（CHPIN）、屏蔽引脚（SHPIN）或不使用（NU），如*图 4-1. TSI 引脚配置*所示。

图 4-1. TSI 引脚配置

```
#ifdef GD32_TSI_USER_BOARD
#define TSI_GROUP0_IO0   NU              //PA0
#define TSI_GROUP0_IO1   NU              //PA1
#define TSI_GROUP0_IO2   NU              //PA2
#define TSI_GROUP0_IO3   NU              //PA3

#define TSI_GROUP1_IO0   NU              //PA4
#define TSI_GROUP1_IO1   NU              //PA5
#define TSI_GROUP1_IO2   NU              //PA6
#define TSI_GROUP1_IO3   NU              //PA7

#define TSI_GROUP2_IO0   CHPIN           //PC5
#define TSI_GROUP2_IO1   SAMPIN          //PB0
#define TSI_GROUP2_IO2   NU              //PB1
#define TSI_GROUP2_IO3   NU              //PB2

#define TSI_GROUP3_IO0   NU              //PA9
#define TSI_GROUP3_IO1   NU              //PA10
#define TSI_GROUP3_IO2   NU              //PA11
#define TSI_GROUP3_IO3   NU              //PA12

#define TSI_GROUP4_IO0   NU              //PB3
#define TSI_GROUP4_IO1   NU              //PB4
#define TSI_GROUP4_IO2   NU              //PB6
#define TSI_GROUP4_IO3   NU              //PB7

#define TSI_GROUP5_IO0   SAMPIN          //PB11
#define TSI_GROUP5_IO1   CHPIN           //PB12
#define TSI_GROUP5_IO2   CHPIN           //PB13
#define TSI_GROUP5_IO3   CHPIN           //PB14
#endif
```

4.2. BANK 配置

用户根据 TSI 引脚的分配，可将不同组的引脚组成为一个 BANK，BANK 里的引脚可实现并行采样，从而达到提高 Touch 库运行效率的目的。（可在 tsi_user.c 中进行配置）。*图 4-2. BANK 配置*显示了如何将 PB12（组 5）组成 BANK0，将 PB13（组 5）组成 BANK1 以及将 PB14（组 5）组成 BANK2。其中 BANKx_CHANNEL_NUMS 表示该 BANK 中的通道引脚数，BANKx_CHANNEL_MSK 表示该 BANK 中的用到的 TSI 通道引脚，BANKx_GROUP_MSK 表示该 BANK 中用到的 TSI 组，CHANNELx_GROUP_IDX 表示 BANK 中各通道引脚所在组的索引。另外宏 USE_SHIELD_PIN 表示是否开启主动屏功能。

图 4-2. BANK 配置

```
#define USE_SHIELD_PIN              (1U)

#if USE_SHIELD_PIN
#define SHIELD_CHANNEL              TSI_PC5
#define SHIELD_GROUP                TSI_GROUP2
#define SHIELD_GROUP_IDX            TSI_GROUP_IDX2
#else
#define SHIELD_CHANNEL              NU
#define SHIELD_GROUP                NU
#endif

#define BANK0_CHANNEL_NUMS          (1U)
#define BANK0_CHANNEL_MSK           TSI_PB12 | SHIELD_CHANNEL
#define BANK0_GROUP_MSK             TSI_GROUP5 | SHIELD_GROUP
#define CHANNEL0_GROUP_IDX          TSI_GROUP_IDX5

#define BANK1_CHANNEL_NUMS          (1U)
#define BANK1_CHANNEL_MSK           TSI_PB13 | SHIELD_CHANNEL
#define BANK1_GROUP_MSK             TSI_GROUP5 | SHIELD_GROUP
#define CHANNEL1_GROUP_IDX          TSI_GROUP_IDX5

#define BANK2_CHANNEL_NUMS          (1U)
#define BANK2_CHANNEL_MSK           TSI_PB14 | SHIELD_CHANNEL
#define BANK2_GROUP_MSK             TSI_GROUP5 |SHIELD_GROUP
#define CHANNEL2_GROUP_IDX          TSI_GROUP_IDX5
```

在完成 BANK 配置后，用户需要修改*图 4-3. BANK 数组定义*中的变量。其中 group_id_arrary 定义了各通道所在的组 ID，该数组元素的顺序也代表了各通道数据在 key_data 数组中存储的位置，便于用户在应用/调试时查看各通道的数据。tsi_bank_array 定义了 bank 相关的数据，包括各 bank 中通道数、通道引脚，通道所在的组及 bank 的初始状态。

图 4-3. BANK 数组定义

```
/* group_id_array array for map group id of each channel */
uint8_t group_id_array[CHANNEL_NUMS] = {
    CHANNEL0_GROUP_IDX, CHANNEL1_GROUP_IDX, CHANNEL2_GROUP_IDX,
};

/* bank array for map bank_channel nums, bank_channel_mask, bank_group_mask, bank initial state */
tsi_bank_struct tsi_bank_array[TSI_BANK_NUMS] = {
    {BANK0_CHANNEL_NUMS, BANK0_CHANNEL_MSK, BANK0_GROUP_MSK, BANK_IDLE},
    {BANK1_CHANNEL_NUMS, BANK1_CHANNEL_MSK, BANK1_GROUP_MSK, BANK_IDLE},
    {BANK2_CHANNEL_NUMS, BANK2_CHANNEL_MSK, BANK2_GROUP_MSK, BANK_IDLE},
};
```

当使用 LOCK 锁定时，各通道优先级可通过如下数组定义，数值越高，优先级越大，参考*图 4-4. 通道优先级定义*。

图 4-4. 通道优先级定义

```
/* TSI channel priority when use lock, bigger value for higher priority */
uint16_t tsi_channel_priority_level[TOUCH_KEY_NUM] = {1, 2, 3};
```

4.3. TSI 参数配置

*图 4-5. TSI 参数配置*展示了 TSI 的各参数配置（可在 tsi_config.h 中进行配置），如下：

1) 宏 TSI_CLK_DIV 定义了电荷转移时钟（CTCLK）分频系数；

2) 宏 TSI_CHARGE 定义了充电状态持续时间；
3) 宏 TSI_TRANSFER 定义了电荷转移状态持续时间；
4) 宏 TSI_EC_EN 定义了是否开启扩展充电功能；
5) 宏 TSI_EC_CLK_DIV 定义了扩展充电状态时钟（ECCLK）分频系数；
6) 宏 TSI_EC_MAX_TIME 定义了扩展充电状态最大持续时间;
7) 宏 TSI_SEQ_MAX_NUM 定义了最大序列周期数；
8) 宏 TSI_TRG_EN 定义了是否开启 TSI 外部触发功能；
9) 宏 TRIG_FALLING 定义了 TSI 外部触发模式；
10) 宏 TSI_INT_EN 定义了是否开启 TSI 中断功能。

图 4-5. TSI 参数配置

```
135 #if defined (GD32_TSI_USER_BOARD)
136   #define TSI_CLK_DIV                (5U)
137   #define TSI_CHARGE                 (1U)
138   #define TSI_TRANSFER               (1U)
139 #endif
140   #define TSI_EC_EN                  (1U)
141   #define TSI_EC_CLK_DIV             (1U)
142   #define TSI_EC_MAX_TIME            (127U)
143   #define TSI_SEQ_MAX_NUM            (5U)
144   #define TSI_TRG_EN                 (0U)
145   #define TRIG_FALLING               (1U)
146   #define TSI_INT_EN                 (0U)
```

4.4. TouchKey 参数配置

*图 4-6. TouchKey 参数配置*展示了 TouchKey 各参数配置（可在 tsi_config.h 中进行配置），如下：

1) 宏 TOUCH_KEY_CALIB_NUM 定义了通道数据校准次数；
2) 宏 TOUCH_KEY_CALIB_DELAY 定义了校准前延迟次数；
3) 宏 TOUCH_KEY_PROX_EN 定义是否开启接近检测功能；
4) 宏 TOUCH_KEY_PROX_LOW 定义了触摸接近低阈值；
5) 宏 TOUCH_KEY_PROX_HIGH 定义了触摸接近高阈值；
6) 宏 TOUCH_KEY_DETECT_LOW 定义了触摸检测低阈值；
7) 宏 TOUCH_KEY_DETECT_HIGH 定义了触摸检测高阈值；
8) 宏 TOUCH_KEY_RECALIB_VALUE 定义了通道数据重校准阈值；
9) 宏 TOUCH_KEY_PROX_DEBOUNCE 定义了触摸接近消抖次数；
10) 宏 TOUCH_KEY_DETECT_DEBOUNCE 定义了触摸检测消抖次数；
11) 宏 TOUCH_KEY_RELEASE_DEBOUNCE 定义了触摸释放消抖次数；
12) 宏 TOUCH_KEY_RECALIB_DEBOUNCE 定义了触摸重校准消抖次数。

图 4-6. TouchKey 参数配置

```
#define TOUCH_KEY_CALIB_NUM          (4U)
#define TOUCH_KEY_CALIB_DELAY        (4U)
#define TOUCH_KEY_PROX_EN            (1U)
#if defined (GD32_TSI_USER_BOARD)
#define TOUCH_KEY_PROX_LOW           (40U)
#define TOUCH_KEY_PROX_HIGH          (50U)
#define TOUCH_KEY_DETECT_LOW         (70U)
#define TOUCH_KEY_DETECT_HIGH        (80U)
#define TOUCH_KEY_RECALIB_VALUE      (50U)
#endif
#define TOUCH_KEY_PROX_DEBOUNCE      (3U)
#define TOUCH_KEY_DETECT_DEBOUNCE    (3U)
#define TOUCH_KEY_RELEASE_DEBOUNCE   (3U)
#define TOUCH_KEY_RECALIB_DEBOUNCE   (3U)
```

4.5. TouchKey 功能配置

*图 4-7. TouchKey 功能配置*展示了 TouchKey 各功能配置（可在 tsi_config.h 中进行配置），如下：

1) 宏 TOUCH_USE_LOCK 定义了是否使用 TouchKey 上锁功能，防止多个触摸同时被检测；
2) 宏 TOUCH_USE_DTO 定义了是否使用检测超时功能，防止外部障碍物导致的误触发；
3) 宏 TOUCH_USE_FLT 定义了是否使用滤波器；
4) 宏 TOUCH_MEAS_RECORD 定义了是否记录上次测量值；
5) 宏 TOUCH_USE_AEC 定义了是否使用自适应环境变化检测；
6) 宏 TOUCH_AEC_A_FAST 与 TOUCH_AEC_A_SLOW 定义了自适应环境检测用到的一阶低通滤波器比例因子；
7) 宏 TOUCH_AEC_DELAY 定义了自适应环境检测周期；
8) 宏 TSI_USE_LOG 定义了是否使用日志输出功能；
9) 宏 USE_RTT_LOG 定义了是否使用 RTT 日志输出。

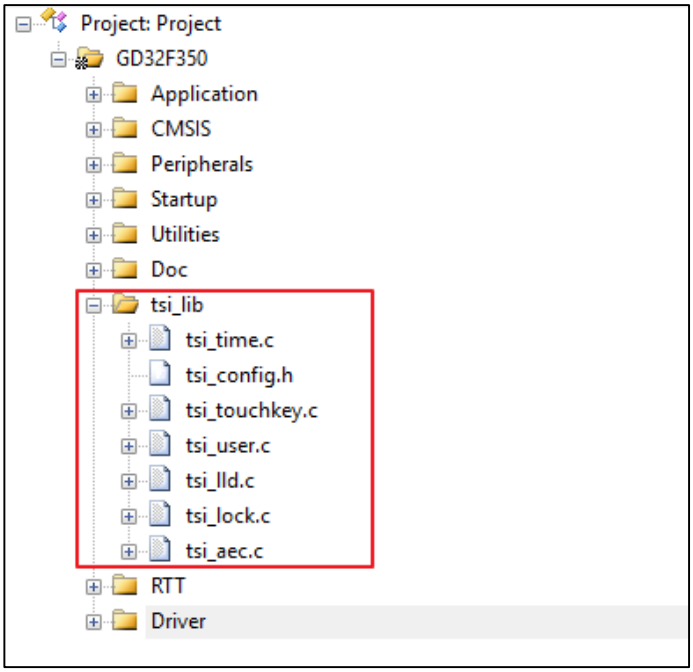
图 4-7. TouchKey 功能配置

```
#define TOUCH_USE_LOCK               (0U)
#define TOUCH_USE_DTO                (0U)
#define TOUCH_USE_FLT                (1U)
#define TOUCH_MEAS_RECORD            (1U)
#define TOUCH_USE_AEC                (1U)
#define TOUCH_AEC_A_FAST             (20U)
#define TOUCH_AEC_A_SLOW             (10U)
#define TOUCH_AEC_DELAY              (500U)
#define TSI_USE_LOG                  (1U)
#define USE_RTT_LOG                  (1U)
```

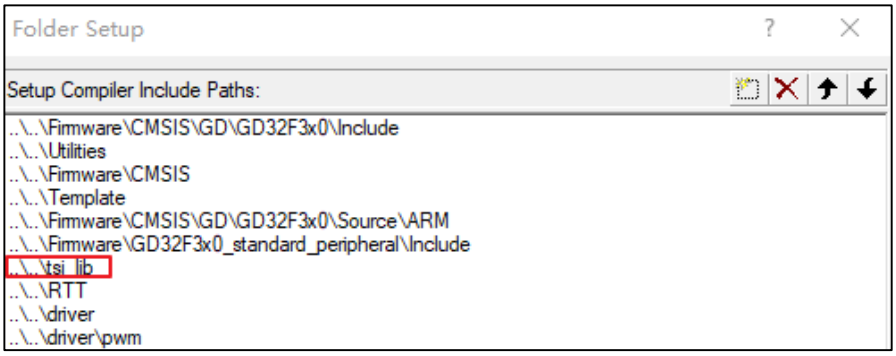
5. TouchKey 库使用

下面将已 GD32350_EVAL 板为例，介绍如何在 KEIL V5.35 工程中添加 TouchKey 库。

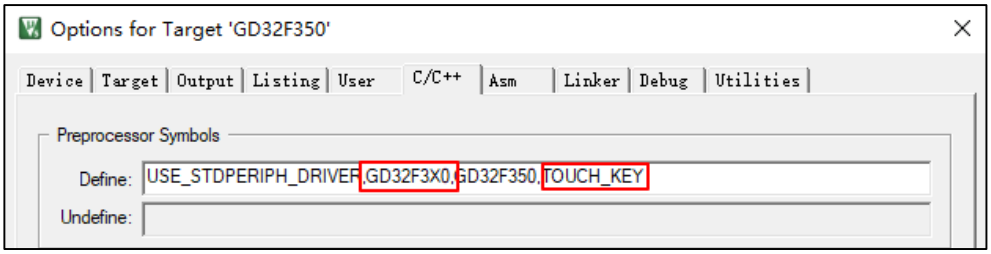
1) 添加 tsi_lib 中的.c 文件



2) 包含 tsi_lib 中的.h 文件



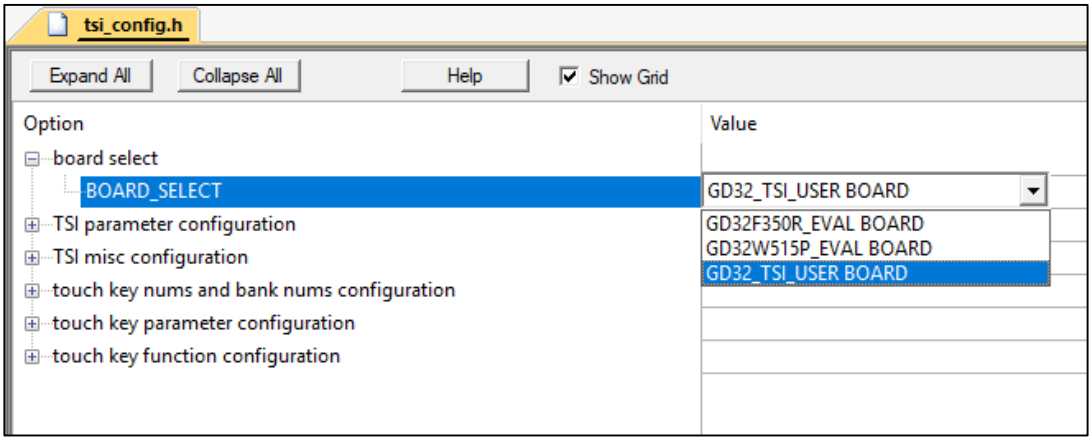
3) 添加预定义宏 TOUCH_KEY 及 GD32F3X0 或 GD32W515



4) 根据需要，选择对应的开发板

如果使用预定义开发板，可选择 GD32F350R_EVAL BOARD 或 GD32W515P_EVAL BOARD；

如果使用客户的开发板应选择 GD32_TSI_USER BOARD。



注意：如果使用预定义开发板 GD32F350R_EVAL BOARD 或 GD32W515P_EVAL BOARD，用户不在需要进行其它 ***TouchKey 库配置***。

5) main.c 中用户代码实现

```
#include "gd32f3x0.h"
#include "tsi_user.h"

uint8_t touch_sate = TSI_BUSY;

/*!
    \brief      main function
    \param[in]  none
    \param[out] none
    \retval     none
*/
int main(void)
{
    touch_init();
    while(1) {
        touch_sate = touch_process();
        if(TSI_OK == touch_sate) {
            uint8_t i = 0;
            for(i = 0; i < TOUCH_KEY_NUM; i++) {
                if(key_data[i].key_state == KEY_DETECT) {
                    /* do something 1 */
                } else {
                    /* do something 2 */
                }
            }
        } else {
            if(TSI_ERROR == touch_sate) {
                /* error process */
            }
        }
    }
}
```

6) 在滴答中断中添加时间管理

工程中使用 Systick 作为滴答定时器，代码部分添加如下。另外，需要在该 C 文件中添加“tsi_time.h”。

gd32f3x0_it.c

```
128 /*!
129     \brief      this function handles SysTick exception
130     \param[in]  none
131     \param[out] none
132     \retval     none
133 */
134 void SysTick_Handler(void)
135 {
136     delay_decrement();
137     tsi_time_process();
138 }
```

7) 工程调试

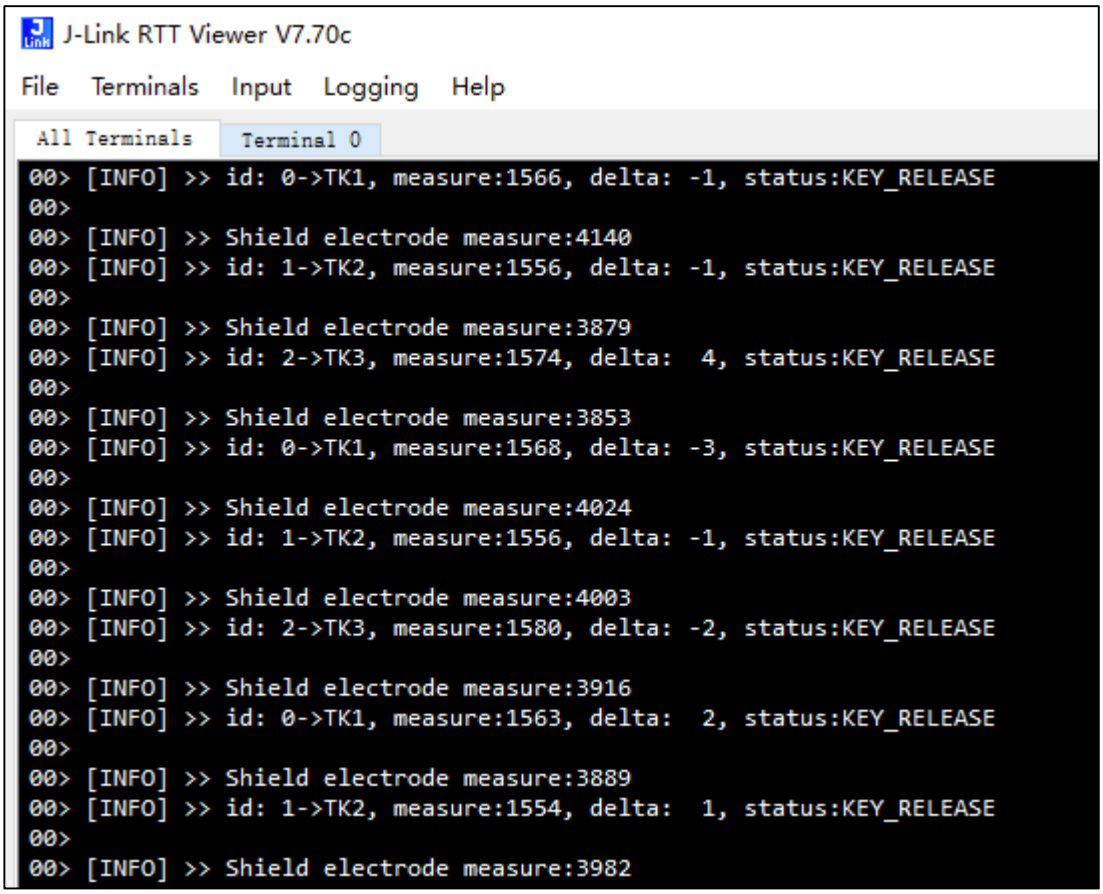
首先，调试并运行工程；然后添加变量“key_data”到观察窗口。触摸按键通道数据（包括状态、参考值、delta 值…）可以在如下的窗口中显示。

Watch 1

Name	Value	Type
key_data	0x20000178 key_data	struct <untagged>[3]
[0]	0x20000178 &key_data	struct <untagged>
key_para	0x20000178 &key_data	struct <untagged>
key_state	0x02 KEY_RELEASE	enum (uchar)
ref	0x00000621	unsigned int
ref_reset	0x000000AD	unsigned int
meas	0x0621	unsigned short
last_meas	0x0620	unsigned short
delta	0x0000	short
last_delta	0x0001	short
counts	0x0000	unsigned short
last_delta_state	0x01 KEY_RELEASE_DE...	enum (uchar)
group_id	0x0005	unsigned short
last_time	0x00000000	unsigned int
aec_start_time	0x00005590	unsigned int
aec_wait	0x01	unsigned char
[1]	0x200001B0	struct <untagged>
[2]	0x200001E8	struct <untagged>
<Enter expression>		

8) 日志输出

如果工程中启用了日志输出功能，以 J-Link RTT 为例，需要在代码中对 RTT 进行初始化并对 printf 进行重定向。日志输出如下（包括触摸按键通道到 ID，测量值，delta 值，状态）：


J-Link RTT Viewer V7.70c
File Terminals Input Logging Help
All Terminals
Terminal 0
00> [INFO] >> id: 0->TK1, measure:1566, delta: -1, status:KEY_RELEASE
00>
00> [INFO] >> Shield electrode measure:4140
00> [INFO] >> id: 1->TK2, measure:1556, delta: -1, status:KEY_RELEASE
00>
00> [INFO] >> Shield electrode measure:3879
00> [INFO] >> id: 2->TK3, measure:1574, delta: 4, status:KEY_RELEASE
00>
00> [INFO] >> Shield electrode measure:3853
00> [INFO] >> id: 0->TK1, measure:1568, delta: -3, status:KEY_RELEASE
00>
00> [INFO] >> Shield electrode measure:4024
00> [INFO] >> id: 1->TK2, measure:1556, delta: -1, status:KEY_RELEASE
00>
00> [INFO] >> Shield electrode measure:4003
00> [INFO] >> id: 2->TK3, measure:1580, delta: -2, status:KEY_RELEASE
00>
00> [INFO] >> Shield electrode measure:3916
00> [INFO] >> id: 0->TK1, measure:1563, delta: 2, status:KEY_RELEASE
00>
00> [INFO] >> Shield electrode measure:3889
00> [INFO] >> id: 1->TK2, measure:1554, delta: 1, status:KEY_RELEASE
00>
00> [INFO] >> Shield electrode measure:3982

6. 版本历史

表 6-1. 版本历史

版本号.	说明	日期
1.0	首次发布	2023 年 5 月 25 日

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.