# 32 位 RSIC 架构的微控制器

SDK 关于 TK 说明

V1.1

## 修订历史记录

变更版本号	日期	变更类型	修改人	审核	摘要

变更类型: A - 增加 M - 修订 D - 删除

## 版权声明

本资料是为了让用户根据用途选择合适的产品而提供的参考资料,不转让属于我公司或者第三方所有的知识产权以及其他权利的许可。在使用本资料所记载的信息并对有关产品是否适用做出最终判断前,请您务必将所有信息作为一个整体系统来评价。对于本资料所记载的信息使用不当而引起的损害、责任问题或者其他损失,我公司将不承担责任。未经我公司的许可,不得翻印或者复制全部或部分本资料的内容。

今后日常产品的更新会在适当的时候发布, 恕不另行通知。在购买本资料所记载的产品时, 请预先向我公司确认最新信息, 并请您通过各种方式关注我公司公布的信息。

如果您需要了解有关本资料所记载的信息或产品的详情,请与我公司的技术服务部门联系,我们会为您提供全方位的技术支持。

## 目录

1.	S	DK 文档结构及项目结构介绍	. 4
2.	SI	DK 中 TK 的 Demo 的分析	. 5
	2.1.	只有 TK 为硬件自动扫描的情况下:	. 5
	2.2.	TK 与 LED 同时为硬件自动扫描的情况下:	. 7
3.	在	E客户的方案板中调试 tk 按键方式:	. 9
	3.1.	使用调试工具进行调试:	. 9
	3.2.	使用 jlink 配合 keil 进行调试	13

#### 1. SDK 文档结构及项目结构介绍



下面为 SDK 的文档结构:

Examples (例程文件夹):将芯片各个模块制作的 demo 放置于此文件夹中,方便用户查看。 Hal (抽象层文件夹):目前只把 TK 的接口抽象后放置于此文件夹里面。 Libraries (库文件夹):里面包含芯片的启动文件,及各个模块的底层库等重要文件。 Project (项目工程文件夹):包含项目启动文件及烧录生成的加密文件等。 User (用户文件夹):放置用户自定义的文件或者可以将 Demo 例程文件拷贝至该文件夹使 用。

双击打开工程,工程路径为······\Project\KEIL-ARM\Project.uvprojx,打开的工程如下:



## 2. SDK 中 TK 的 Demo 的分析

#### 2.1. 只有 TK 为硬件自动扫描的情况下:

客户可以参考 Demo 在该路径下·····\Examples\TK\TK\_ONLY,将此目录下的所有文件复制拷贝 到 User 目录 ( <u>默认的 SDK 已经完成拷贝</u>), 之后打开工程·····\Project\KEIL\_ARM\Project.uvprojx。

下面介绍该 demo 的主要参数及调用方法

a) 先看下使能的 TK 的按键及个数和是否开启 led 与 tk 的硬件扫描宏定义,通过查看 ts32fx\_tk\_cfg.h (在项目的 TS32Fx\_HAL\_driver 下有 ts32fx\_hal\_tk\_bsp.c 双击打开后,查看代 码中的#include "ts32fx\_tk\_cfg.h"的位置,并鼠标点击此位置右击再点击 open documer "ts32fx\_tk\_cfg.h"文件即可打开该文件)即可查看到使能的 tk 的位为: 0x0017f, 个数为 8, 根 据使能位可以知道使能 tk0, tk1, tk2, tk3, tk4, tk5, tk6, tk8, 共 8 个 tk, 且 TK LED BOTH\_HW AUTO SCAN 宏定义为 0,表示只使用 tk 进行硬件扫描。如下图:





b) demo 中是如何使用 tk 的:

1、初始化 TK 的 IO: hal\_tk\_gpio\_init() (该函数为抽象层自带)

2、初始化 TK 的控制: hal\_tk\_ctl\_init() (该函数为抽象层自带)

3、在需要获取按键的地方调用 hal\_tk\_key\_map\_get(TK) 获取 key 被按下的 bit-map(该函数为抽象层自带)。如下图:



#### 2.2. TK 与 LED 同时为硬件自动扫描的情况下:

客户可以参考 Demo 在该路径下……\Examples\TK\ TK&LED\_HW\_AUTO\_SCAN, 将此目录下的所有文件复制拷贝到 User 目录, 之后打开工程……\Project\KEIL\_ARM\Project.uvprojx。 下面介绍该 demo 的主要参数及调用方法

a) 先看下使能的 TK 的按键及个数和是否开启 led 与 tk 的硬件扫描宏定义,通过查看 ts32fx\_tk\_cfg.h (在项目的 TS32Fx\_HAL\_driver 下有 ts32fx\_hal\_tk\_bsp.c 双击打开后,查看代 码中的#include "ts32fx\_tk\_cfg.h"的位置,并鼠标点击此位置右击再点击 open documer "ts32fx\_tk\_cfg.h"文件即可打开该文件)即可查看到使能的 tk 的位为: 0x0017f, 个数为 8, 根 据使能位可以知道使能 tk0, tk1, tk2, tk3, tk4, tk5, tk6, tk8, 共 8 个 tk, 需要将 TK\_LED\_BOTH\_HW\_AUTO\_SCAN 宏定义置为 1, 因为这样表示只使用 tk 与 led 都为硬件自 动扫描。如下图:



b) demo 中是如何初始化 tk 及 led 的,及使用 tk:

1、初始化 led: led\_ctl\_init() (该函数需要用户根据自己定义修改)

2、初始化没有被 led 复用的 tk 引脚,将这些引脚配置为输出低: tk\_no\_reuse\_pins\_init() (该函数需要用户根据自己定义修改)

3、初始化 tk 控制: hal\_tk\_ctl\_init() (该函数为抽象层自带)

4、在需要获取按键的地方调用 hal\_tk\_key\_map\_get(TK) 获取 key 被按下的 bit-map(该函数为抽象层自带),如下图:



## 3. 在客户的方案板中调试 tk 按键方式:

调试过程分两种,一种为使用调试工具上位机进行调试(推荐,但需要占用芯片一路 uart 资 源),另一种为使用 jlink 配合 keil 进行在线调试。

#### 3.1. 使用调试工具进行调试:

(1) 确认芯片的引脚数,并在 sdk 中设置相应的宏定义;打开 ts32fx\_system.h 文件将相应的引脚宏定义置 1

E:\ts32f020_sdk_V2.00-20190903\Project\K	-ARM\Project.uvprojx - μVision		– 🗆 X
File Edit View Project Flash Debug Per	nerals Tools SVCS Window Help		
🗎 🖆 🛃 🕼 👗 🛍 🛍 🖉 🖛	- 陀 🏗 🥂 🔃 ቹ 🎼 ///////////////////////////////////	DEBUG 🔤 🗟 🥐 🍭 🗉 🔿 🔗 🅀 🔚 🖣	
🧼 🔛 🕮 🧼 🔛 🙀 TS32Fx	🖂 🔊 📥 🖶 🗇 🌚		
Project 4 🗵	ts32fx_hal_tk_bsp.c ts32fx_hal_tk_bsp.h t	s32fx_tk_cfg.h* ts32fx_system.c ts32fx_system.h	<b>→</b> ×
🖃 🍄 Project: Project	61 * You must must close this mac	ro , if it is formal Mass production chip	chip already do it 🔺
🖮 🐲 TS32Fx	62 * or else it is very very seri	ous mistake.	
🖨 🦢 User	63 - */		
🗊 📄 main.c	64 #define USE_ADC_NON_MASS_PRODUCT	U	
Is32fx_irq.c	66 - */	364643	
🕀 📄 user.c	67 #define USE CHIP 16 PIN	0	
🖃 🦢 TS32Fx HAL driver	68 #define USE_CHIP_20_PIN	◎ ┃ ← ● 根据自己选	择的芯片,使能相
ts32fx hal tk bsp.c	69 #define USE_CHIP_24_PIN	○ 成位2100字	空火 図山山工店
TS32Ev 11 driver	70 #define USE_CHIP_28_PIN		足又, 国中田」 医
RVMDK	71 Faerine USE_CHIP_48_PIN	用的是48pi	in芯片所以选择了
	73 H#if (USE CHIP 48 PIN    USE CHIP		10DINI为1 甘仝
tsozix_asseric	74 /*! no remap add channel for pb	pin 03L_CHIF	
tsszix_debug.c	75 - */	为0	
ts32fx_misc.c	76 #defineCHIP_PB_NO_REMAP		
s32tx_system.c	77 Fendir 79		
ts32fx_startup.c	79 Hif FPGA EN		
CMSIS	80 #define SYS CLK	4800000UL	
	81 #else		
	<pre>82 #define SYS_CLK</pre>	5200000UL	
	83 #endif		
	84 - 85 //forsema import/ use no semiho	ating)	
	86 // #pragma import (use_no_semino	scing)	
	87 #define DEBUG LEVEL	DL DGB	
	<pre>88 #define DMA_DBG_EN</pre>	0 -	
	<pre>89 #define SD_DBG_EN</pre>	1	
	90   #define LCD_DBG_EN	0	``````````````````````````````````````
BOOKS   W FUNCTIO   U., Iemplat			
Build Output			<b>д </b> 🛛
Watchpoints: 2 JTAG speed: 4000 kHz			^
Full Chip Erase Done.			
Programming Done.			
Verify OK.			×
<			>
📧 Build Output 🥁 Find In Files 🔯 Browser			
		J-LINK / J-TRACE Cortex	L:18 C

(2) 确认芯片是否同时使用 tk 与 led 内部硬件扫描,并设置相应的宏定义; 打开 ts32fx\_tk\_cfg.h

E:\ts32f020_sdk_V2.00-20190903\Project\K	CEIL-ARM\Project.uvprojx - µVision ×
File Edit View Project Flash Debug Per	ripherals Tools SVCS Window Help
	🛶   陀 臨 臨   谭 谭 ////////// //////////////////
🧼 🏭 🚅 🥔 🔜 🛱 TS32Fx	🖂 🔊 🛔 🕾 🔶 ớ 🏨
Project 🛛 🕱 🔀	ts32fx hal tk bsp.c ts32fx hal tk bsp.h ts32fx tk cfg.h* ts32fx system.c ts32fx system.h
🖃 🍄 Project: Project	13 4 */
🖻 🚂 TS32Fx	14 /* Define to prevent recursive inclusion*/
🖮 🗁 User	15
🖽 📄 main.c	
H ts32fx irg.c	
TC225 HALL	20 Destern "C" (
TS32FX_HAL_driver	21 fendif
	22 -
TS32Fx_LL_driver	23 /************************************
😑 🦢 RVMDK	24 #define TK_LED_BOTH_HW_AUTO_SCAN 0 ← 配署为0 丰玉口右TK 为硬件白动扫
ts32fx assert.c	
ts32fx debug c	26 #define TK_BIT_ENABLE 0x0017F 描方式,
a a toongacought	27 #define TK_ENABLE_NUM 8
in an issue	28
	29
ts32fx_startup.c	
🛶 CMSIS	31 - TK Charge CAP Hode Config
	33 idefine TK CCVPS 0
	34 #define TK FSDIV 9
	35 #define TK DISCHARGE TIME 11
	36 #define TK CHSWUP SPEED 0
	37 #define TK CHSWUP 0
	38
	39
	40 #define CC_PRECYC_NUM 1
	41 #define CC_PER_SAMPLE_CLK 0
Repeat Books O Europia D. Templat	42 #define CC_PERIOD_SAMPLE_NUM 3
Books   Branchow Branchow	
build Output	4 · · · · · · · · · · · · · · · · · · ·
Watchpoints: 2 JTAG speed: 4000 kHz	
Eull Chin Frase Done.	
Programming Done.	
Verify OK.	
<	>
📧 Build Output 🙀 Find In Files 📑 Browser	
	LINK / LTRACE Contex 1-27.0

(3) 确认硬件使用 tk 的索引, 配置使能的 tk map 位及数量; 打开 ts32fx\_tk\_cfg.h

E:\ts32f020_sdk_V2.00-20190903\Project\K	EIL-ARM\Project.uvprojx - µVision —	$\Box$ $\times$
File Edit View Project Flash Debug Per	ipherals Tools SVCS Window Help	
🗎 🖻 🛃 🎒 🐰 🖻 🛍 🎐 🥲 듲	→   陀 隐 隐 [] 律 律 //: //::/ //: //: //: //: //: //: /	
🧼 🍱 🕮 🥔 🖳 🙀 TS32Fx	🖂 🔊 📥 🗟 🗢 🐡 🃾	
Project 🏨 🔀	ts32fx_hal_tk_bsp.c	<b>▼</b> ×
🖃 🔧 Project: Project	13 4 */	^
🖮 ᇶ TS32Fx	14 /* Define to prevent recursive inclusion*/	
🖨 🗁 User	15	
iii 📄 main.c		
H ts32fx ira.c	17 HINGET IS32FX IK CFG H	
Iliser c	19 stifter columnus	
TS22En HAL driver	20 Hextern "C" (	
m Ac226 belak ber a	21 #endif	
tsiztx_nal_tk_osp.c	22 -	
IS32Fx_LL_driver	23 /************************************	
E C RVMDK	24 #define TK_LED_BOTH_HW_AUTO_SCAN 0	
Is32fx_assert.c	25 tdefine TK BIT ENABLE 0x0017E	
ts32fx_debug.c	27 fdefine TK ENABLE NUM 8	
⊞- 🔝 ts32fx_misc.c		
Is32fx_system.c	29	
ts32fx startup.c	30 0/*	
A CMSIS	31 * TK Charge CAP Mode Config	
	32 - */	
	33 Factine IK COVKS 0	
	35 fdefine TK_FSDIV 9	
	36 #define TK CHSWUP SPEED 0	
	37 #define TK CHSWUP 0	
	38	
	39	
	40 #define CC_PRECYC_NUM 1	
	41 #define CC PER SAMPLE CLK 0	
Preinet Realize & Sunctio D Templat	42 FORTING CC_PERIOD_SAMPLE_NUM 3	Ň
and the second s	1.	
Build Output		ф <mark>х</mark>
Watchpoints: 2 JTAG speed: 4000 kHz		^
Full Chip Erase Done.		
Varify OK		
CITIY OR.		~
Puild Output		
Langer and the state of the sta		L
	LUNK / LTBACE Contex	1.21 (

#### (4) 调用 tk 抽象层的初始化函数

E:\ts32f020_sdk_V2.00-20190903\Project\H	(EIL-ARM\Project.uvprojx - µVision	– 🗆 ×
File Edit View Project Flash Debug Pe	ripherals Tools SVCS Window Help	
🔊 📾 🐇 🕼 🖬 🛀		
🧼 🖾 🎬 🥔 🖳 🙀 TS32Fx	🖂 🔊 🛔 🗟 🗇 🏟	
Project 📮 🗵	user.c main.c ts32fx_irq.c ts32fx_hal_tk_bsp.c ts32fx_hal_tk_bsp.h* ts32fx_tk_cfg.h*	₹ ×
<ul> <li>Project Project</li> <li>TS32Fx</li> <li>Use</li> <li>Imain.c</li> <li>Its32h.cir.q.c</li> <li>Its32h.cir.q.k</li> <li>Its32h.kbsp.c</li> <li>TS32Fx/LL_driver</li> <li>Its32h.cir.g.ksset.c</li> <li>Its32h.cir.g.ksset.c</li> <li>Its32h.cir.g.ksset.c</li> <li>Its32h.cir.g.ksset.c</li> <li>Its32h.cir.g.ksset.c</li> <li>Its32h.cir.g.ksset.c</li> <li>Its32h.cir.g.ksset.c</li> <li>Its32h.gstet.c.</li> <li></li></ul>	77     delay_ms(l);       78     ;       81     ::       82     :       83     :       84     :       85     :       86     :       86     :       87     :       88     :       98     :       181     :       99     :       90     :       91     :       92     :       93     :       94     :       95     :       96     :       97     :       98     :       99     :       99     :       101     :       102:     :       103     :       104     :       105:     ::       106:     ::	^
Project Reports O Function De Templation	106 hai_tk_tool_debug_excute();	>
Build Output		a 🗴
Watchpoints: 2		
JTAG speed: 4000 KHz Full Chip Erase Done. Programming Done. Verify OK.		•
🖻 Build Output 🕞 Find In Files 🔯 Browser		
	J-LINK / J-TRACE Cortex	L:86 (

(5) 在程序循环主体中,插入tk工具调试的接口



(6) 开启 tk 工具调试的宏定义,并配置相应的单 pin 调试串口参数;

E:\ts32f020_sdk_V2.00-20190903\Project\	EIL-ARM\Project.uvprojx - μVision	- 🗆 ×
File Edit View Project Flash Debug Pe	ripherals Tools SVCS Window Help	
🗈 💕 🖬 🕔 🐰 🖦 🖍 🖃 🖛	→   陀 隐 隐 段   詳 詳 /// /// /// 💆 USE_TK_JLINK_DEBUG 🔍 🔜 🎺 🔍 🔍 ● ○ 🔗 🚓 🔚 🖬 -	4
🖉 🏭 🍘 🥩 🔛 🙀 TS32Fx		
Project 4 X	user c b ts32fy hal tk hsp. c b ts32fy hal tk hsp.h* b ts32fy tk dp.h* b ts32fy system c	] ts32fx system.h ₹
Project: Project		
🖮 🚌 TS32Fx	17	
😑 🦢 User	18 /* Define to prevent recursive inclusion	*/
i main.c	19 #ifndefTS32FX_HAL_TK_BSP_H	
s ts32fx irg.c	20 #defineTS32FX_HAL_TK_BSP_H	
	22	
TERRE LIAL drives	23 Hextern "C" {	
- TSS2PX_HAL_driver	24 #endif	
. ts32tx_hal_tk_bsp.c	25 -	
TS32Fx_LL_driver	26 /* Includes	*/
😑 🤛 RVMDK	27 #include "include.h"	
ts32fx_assert.c		
ts32fx_debug.c	29 / thefine USE TW JULINE DEBUG	
ts32fx_misc.c	30 FREINE OSE_IK_DEBOG 0	
ts32fx system.c	32 /******************** via TOOL debug tk ***********************************	***/
trazific statum c	33 #define USE TK TOOL DEBUG 1 ← 打开TK工具调试宏定	₽₩
	34	
CMBIS	35 Hif USE TK TOOL DEBUG	
	36 #define TKDEBUG_GPIOx     GPIOB	L具调试串口参数,
	37 #define TKDEBUG PINx LL GPIO PIN 10	1両空光, DD10
	36 FORTING TRUERING NEW LL GPIO PIN SOURCE 10 79	PELE / FDIU,
	40 idefine TKDEBUG HARTY HARTO funct	ion功能为AF1,为
	41 #define TKDEBUG UARTX IROn UARTO IROn	
		JARIU, 中断万
	43 //#define TKDEBUG_GPIOx GPIOC UAR	F0 IROn
	44 //#define TKDEBUG_PINx LL_GPIO_PIN_1	
	45 //#define TKDEBUG_PIN_SOURCEx LL_GPIO_PIN_SOURCE_1	
E Project Books   {} Functio U, Templat		>
Build Output		Ф.
Watchpoints: 2		
JTAG speed: 4000 kHz		
Full Chip Erase Done.		
Programming Done.		
Verify OK.		
<		>
📰 Build Output 🔤 Find In Files 📑 Browser		
	J-LINK / J-TRACE Cortex	1:41 (

(7) 插入串口中断调试函数

如上图使用的串口调试为 UARTO, 即在 ts32fx\_irq.c 中的 UARTO 中断中加入函数

E:\ts32f020_sdk_V2.00-20190903\Project\K	EIL-ARM\Project.uvprojx - µVision — 🗆	×
File Edit View Project Flash Debug Per	pherals Tools SVCS Window Help	
	→   陀 隐 隐 读 读 //: //:: //:: 🖄 USE_TK_JLINK_DEBUG 🛛 🗟 🌮 🍭 🗕 • • ◇ 🔗 🛃 🖬 🔹	
😵 🖾 🔛 😻 🔜 🛱   TS32Fx		
Project A 🗵	ts32fx_irq.c 🚺 ts32fx_hal_tk_bsp.c 🚺 ts32fx_hal_tk_bsp.h* 🛄 ts32fx_tk_cfg.h* 🚺 ts32fx_system.c 🚺 ts32fx_system.h	₹ ×
Project Proje	186日/**       #Brief Interrupt WARI Interrupt Handle function         187       * Brief Interrupt Handle function         188       * 9(         199       * (         191       * (         192       * Brief Uarto Interrupt handler         193       * (         194       * (         195       * (         196       * (         197       * (         198       * (         197       * (         198       * (         197       * (         198       * (         199       * (         198       * (         199       * (         199       * (         199       * (         199       * (         199       * (         199       * (         199       * (         199       * (         199       * (         199       * (         199       * (         199       * (         199       * (         199       * (         199       (         199       * (	^
CMSIS	206 ⊟/** @brief Uartl Interrupt handler 206 * @param None. 207 ± @terval None. 208 void UART1_IRQHandler(void) 210 ⊟ ( 211 /* Get the value of the interrupt flag bit register */ 212 213 } 214 ±	×
🔚 Project 🔞 Books 🛛 🕄 Functio 🗛 Templat		>
Build Output		<b>д X</b>
Watchpoints: 2 JTAG speed: 4000 kHz Full Chip Erase Done. Programming Done. Verify OK.		~
<		>
📰 Build Output 🛛 🙀 Find In Files 🛛 🗃 Browser		
	J-LINK / J-TRACE Cortex	L:96 C

(8) 使用串口单 pin 模块连接方案板及上位机,根据上位机提示进行调试,生成 "ts32fx\_tk\_cfg.h"替换工程下的同名文件即可。

## 3.2. 使用 jlink 配合 keil 进行调试

- (1) 确认芯片的引脚数,并在 sdk 中设置相应的宏定义,同上面方法 1;
- (2) 确认芯片是否同时使用 tk 与 led 内部硬件扫描,并设置相应的宏定义,同上面方法 1;
- (3) 确认硬件使用 tk 的索引, 配置使能的 tk map 位及数量, 同上面方法 1;
- (4) 调用 tk 抽象层的初始化函数, 同上面方法 1;
- (5) 插入调试代码;打开 ts32fx\_irq.c 文件在 TK\_DONE\_IRQHandler 中断函数中插入调试代码

🗑 E:\ts32f020_sdk_V2.00-20190903\Project\KI	EIL-ARM\Project.uvprojx - μVision	– 🗆 🗙
File Edit View Project Flash Debug Peri	ipherals Tools SVCS Window Help	
	→   陀 陰 陰 段   淳 淳 //: //::  🖄 USE_TK_LIINK_DEBUG 🛛 🗟 🌮   🍭   🌢 ○ 🔗 🍓 🖬 🔹 🔦	
🗇 🔛 🎬 🥏 📖 🙀 TS32Fx	V 🔊 🛔 🗟 🔶 🗇 🏟	
Project 🛛 📮 🗵	user.c main.c ts32fx irq.c ts32fx hal tk bsp.c ts32fx hal tk bsp.h* ts32fx tk cfq.h*	<b>₹</b> ×
🖃 🍄 Project: Project	132 /* TK->KEYPND = 0xFFFFFFF; */	^
🖃 💭 TS32Fx	133 TK->KEYPND = TK->KEYPND;	
🖻 🦾 User	134 - }	
main.c	136	
	137 - /**	
user.c	138 * @brief TK_DONE Interrupt handler	
S32Fx_HAL_driver	140 * @retval None.	
	141 4/	
IS32Fx_LL_driver	142 void TK_DONE_IRQHandler(void)	
	144 if (TK->TNTPND & LL TK INTPND SCDN PENDING) (	
tssztx_assert.c	145 TK->INTPND = LL TK INTPND SCDN PENDING;	
trazin mission	146 = #if USE_TK_JLINK_DEBUG	
tr22fx purtem c	147 //tk_jlink_debug_print_sampval(8); 148 bal_tk_ilink_debug_func();	
s32fx startun c	149 - fendif	
	150 /* TK->SCDOPND = 0xFFFFFFF; */	
•	151 TK->SCDOPND = TK->SCDOPND;	
	152 }	
	154 L	
	155 -/**	
	156 * gorier TK OVF Interrupt nandler	
	158 * @retval None.	
	159 4 */	
	160 void TK_OVF_IRQHandler(void)	
E Project Books   {} Functio 0, Templat	<	>
Build Output		<b>4</b> 🗵
Watchpoints: 2		^
JTAG speed: 4000 kHz		
Full Chin Frage Done		
Programming Done.		
Verify OK.		~
<		>
📰 Build Output 🙀 Find In Files 🔯 Browser		
	J-LINK / J-TRACE Cortex	L:96 C

#### (6) 开启 tk 调试的宏定义;

#### 打开文件 ts32fx\_hal\_tk\_bsp.h 中 USE\_TK\_JLINK\_DEBUG 置为1

E:\ts32f020_sdk_V2.00-20190903\Projec	t\KEIL-ARM\Project.uvprojx - μVision	- U ×
The Edit View Project Flash Debug	Peripherals loois SVCS Window Help	
🖂 🚰 🍕 🖇 🖻 🛃 🍬 🗠   -	← →   🍢 🐘 🐘   梁 譯 //= //=   @ USE_TK_JUNK_DEBUG 🛛 🖳 🆓   🍭   ● ○ 🔗 🍓   🔜 📲	
🤌 🏭 🕮 🥔 🔜 🛛 🗱 TS32Fx	🖂 🔊 📥 🗟 🗇 🏟	
oject 🎗	🔟 🚺 user.c 🗋 main.c 🗋 ts32fx_irq.c 📑 ts32fx_hal_tk_bsp.c 📄 ts32fx_hal_tk_bsp.h* 🗋 ts32fx_tk_cfg.h*	-
🛯 🔧 Project: Project	16 */	
😑 🚛 TS32Fx	17 -	
🖨 🦢 User	18 /* Define to prevent recursive inclusion	1
🗉 📄 main.c	19 Hifder TS32FX HAL IK BSP H	
ts32fx irq.c		
I USELC	22 #ifdef cplusplus	
TS32Ev HAL driver	23 extern "C" {	
tr22fr hal tk hen c	24 #endif	
TCODE LL L	25 -	
ISS2FX_LL_driver	26 /* Includes	1
RVMDK	27 #include "include.n"	
Is32fx_assert.c	29 /************************************	
iii 🛄 ts32fx_debug.c	30 #define USE TK JLINK DEBUG 1	
ts32fx_misc.c	31	
ts32fx_system.c	32 /********************** via TOOL debug tk ***********************************	
⊞ Its32fx startup.c	33 #define USE_TK_TOOL_DEBUG 0	
	34	
• • • • • • • • • • • • • • • • • • • •	35 #if USE_TK TOOL DEBUG	
	36 #define TKDEBUG GPIOX GPIOB	
	38 fdefine TKDEBUG_FINX LL_GFIO_FIN_10	
	39 define TKDEBUG AFX LL GPIO AF 1	
	40 #define TKDEBUG UARTX UARTO	
	41 #define TKDEBUG UARTx_IRQn UART0_IRQn	
	42	
	43 //#define TKDEBUG_GPIOx GPIOC	
	44 //#define TKDEBUG PINx LL GPIO PIN 1	
Project Books { Europia D. Templa	45 //#define TKDEBUG_PIN_SOURCEX LL_GPIO_PIN_SOURCE_1	>
arroject		
Jild Output		4
atchpoints: 2		
IAG Speed: 4000 kHz		
ull Chip Erase Done.		
rogramming Done.		
erify OK.		
		>
Build Output 🙀 Find In Files 📑 Browser		
	LINK ( TRACE Cotor	1,20

(7) 点击编译后再点击 keil 中的在线调试,添加 tk\_debug\_data 到调试窗口,按下按键进行 tk 调试。



如下图为该数组的含义,可根据这些参数设置相应的软件阈值及硬件阈值

