

版本	V1.1
日期	2022-10



APT32F110x TOUCH 应用指南

适用于**CSI SDK**

版权所有©深圳市爱普特微电子有限公司

本资料内容为深圳市爱普特微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，深圳市爱普特微电子有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，深圳市爱普特微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，公司保留未经预告的修改权。

Copyright © 2012-2022 APTCHIP Mircoelectronics, Inc. All Rights Reserved

Revision History

版本	日期	描述	作者
V1.0	2022.9.14	初版	DXY/XB/WNN
V1.1	2022.10.19	增加对4种类型touch库的描述	XB

目录

1. 文档说明4

2. 资源列表5

3. 硬件环境6

4. 软件环境7

5. 硬件设计注意事项14

1. 文档说明

本文档适用于APT32F110x系列芯片，用以说明在CSI SDK中使用TOUCH的方法。

2. 资源列表

为了在APT32F110x CSI SDK上使用touch功能，您将可能需要下列资源

- Touch Evaluation Board
- APT32F110x 数据手册
- APT32F110x 使用手册
- QuickStart_APT32F110x系列_CSI
- Touch Key使用串口工具波形分析使用指南

3. 硬件环境

以Touch Evaluation Board为例介绍。

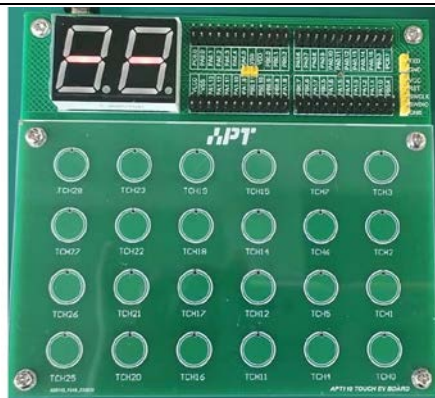


Figure 1 APT32F110x Touch Evaluation Board

该板与touch的原理图如图所示：

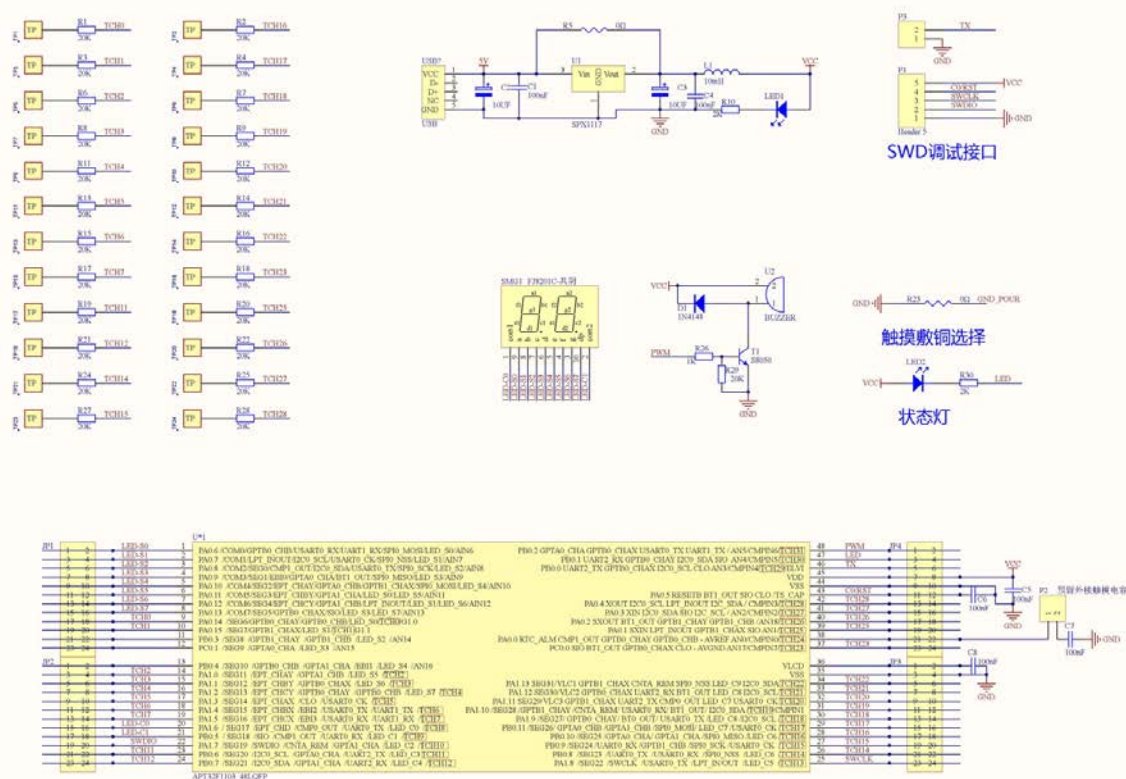


Figure 2 原理图

4. 软件环境

4.1 获取SDK

从CDK或我司获取APT32F110x SDK（V1.1.0以上版本）。从CDK获取SDK的方法请参考文档《QuickStart_APT32F110x系列_CSI》。

4.2 TOUCH demo的使用

4.2.1 型号适配

APT32F110x系列有4个子系列，其中只有1101和1103子系列支持touch功能。所以请首先确定自己使用的芯片具体型号，然后开启对应的宏。如使用APT32F1101，则定义 IS_CHIP_1101=1；如使用APT32F1103，则定义IS_CHIP_1103=1。如果没有选择正确的型号，如没有选择带touch的型号却调用了touch的相关函数，编译连接的时候就会报错。

下图示例了使用APT32F1103的情况。

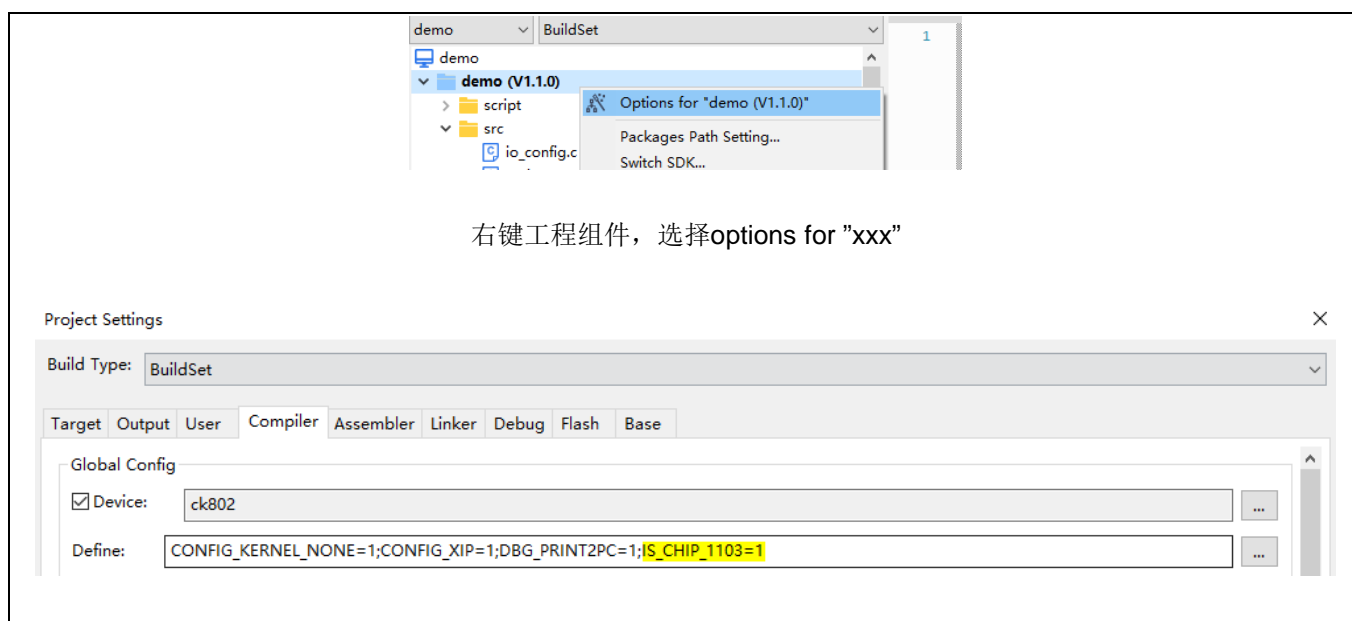


Figure 3 型号适配

4.2.2 touch库的说明和使用

4.2.2.1 Touch库类型说明

为了便于版本管理，我们的touch驱动以库的形式发布。我们为APT32F110x系列的touch准备了四种库，以匹配

不同的应用需求。区别如下：

- lib_110tkey_F_x_xx: 跳频版本。Touch使用中断处理，占用更多的SRAM及程序空间，抗干扰能力最强
- lib_110tkey_C_x_xx: 默认版本。Touch使用中断处理，SRAM及程序占用空间更小，速度更快，抗干扰性能稍低
- lib_110tkey_MC_x_xx: 主循环版本。使用Touch需要调用tkey_main_prog();函数，扫描时间更长，完全不占用中断，按键越多扫描时间越长，按键速度也会越慢
- lib_110tkey_DMA_x_xx: TOUCH->DMA版本。使用DMA3传递touch采样数据，不使用touch中断，速度最快，占用最多的SRAM，使用此版本后不再允许配置DMA3。

4.2.2.2 将touch库加入到工程中

可以在sdk_110x下apt32f110x_chip组件的组件配置linker tab中查看是否已包含touch库，及当前所用touch库版本号。

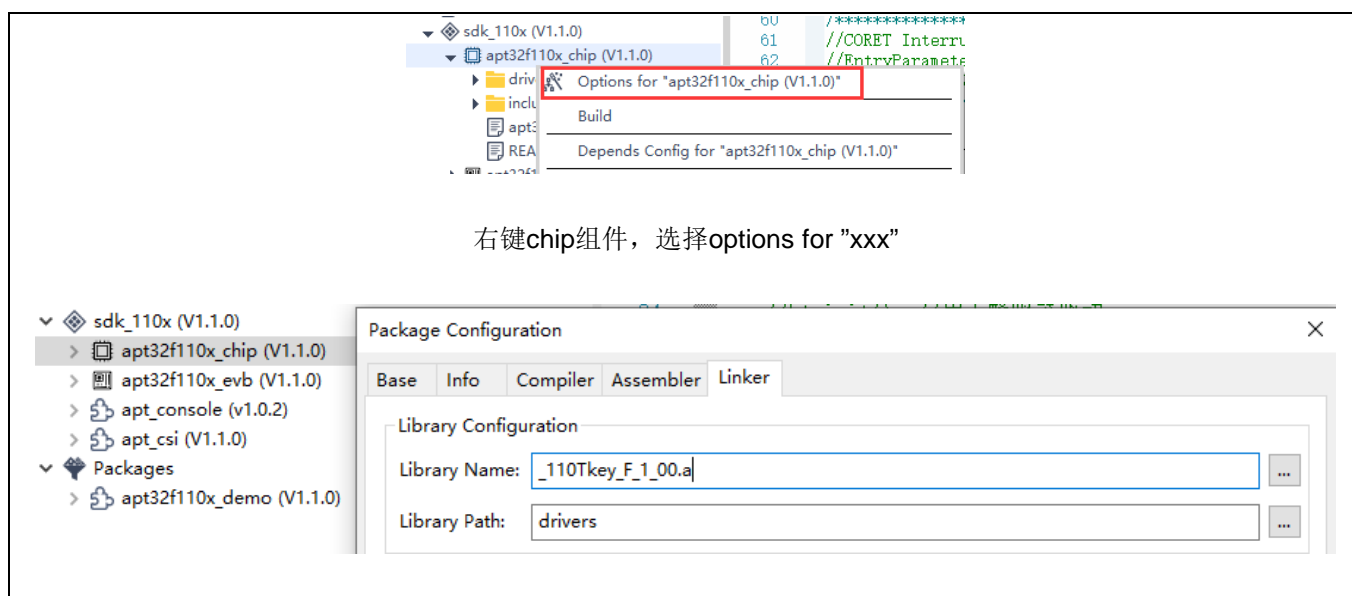


Figure 4 查看/增加/替换库

如果当前工程没有包含touch库，或要使用新版本的touch库，在确保components\chip\drivers里有目标touch库的前提下，在上图Library Name选择即可。

4.2.3 调用tkey_demo()函数

直接在主函数适当位置调用tkey_demo()即可。该demo支持第一个按键值识别。


```

/*
 * 操作步骤
 * ① 右键 chip 组件，选择 ptions for "XXX"选项
 * ② linker 选项卡界面，Library Name 输入对应的库名称，如:_110Tkey_F_1_00.a，版本说明请参考
 相关 appnote
 * ③ Library Path 后面输入路径: drivers
 * 然后在 sdk_110x->apt32f110x_chip->drivers->tkey_parameter.c 里配置相关的参数，配置说明
 请参考相关 appnote
 */
#define serialplot_debug_en           //如果不需要在 PC 端打印调试信息，可以去掉该行
void tkey_demo(void)
{
    SYSCON->CLCR = 0xa5a09500; //B4
    csi_pin_set_mux(PB01,PB01_OUTPUT); //LED
    gptb0_init(); //蜂鸣器
#ifdef _serialplot_debug_en
    uart_init();
#endif
    led_init(); //LED 数码管
    csi_tkey_init();
    delay_ums(3000);
    while(1)
    {
        tk_led();
        if(dwKey_Map!=0)
        {
            if(dwKey_Map!=dwKey_Map_bk)
            {
                dwKey_Map_bk=dwKey_Map;
                csi_pin_set_low(PB01);
                csi_gptb_start(GPTB0); //start timer
            }
        }else
        {
            dwKey_Map_bk=0;
            csi_pin_set_high(PB01);
            csi_gptb_stop(GPTB0); //stop timer
        }
        //tkey 采样值打印，用于配合上位机观察实时波形
#ifdef _serialplot_debug_en
        csi_uart_putc(UART2, 0X0D);
        csi_uart_putc(UART2, 0X0A);
        for(int i =0;i<16;i++)
        {
            csi_uart_putc(UART2, (hwSampling_data0[i]>>8));
            csi_uart_putc(UART2, (hwSampling_data0[i]&0xff));

            csi_uart_putc(UART2, (hwBaseline_data0[i]>>8));
            csi_uart_putc(UART2, (hwBaseline_data0[i]&0xff));
        }
#endif
    }
}

```

Figure 5 tkey_demo函数

在touch evaluation board上正常运行起来后，每按一个键，LED灯亮、蜂鸣器响的同时，8段数码管上会显示当前按键的键值。

其中，dwKey_map对应触摸按键值。在不同的按键模式下，有不同的意义。具体参看[TOUCH键值读取](#)。

4.2.4 Touch调试

除了在touch evaluation board上的显示外，还可以通过串口将touch的采样值实时打印到PC端。如下图所示：

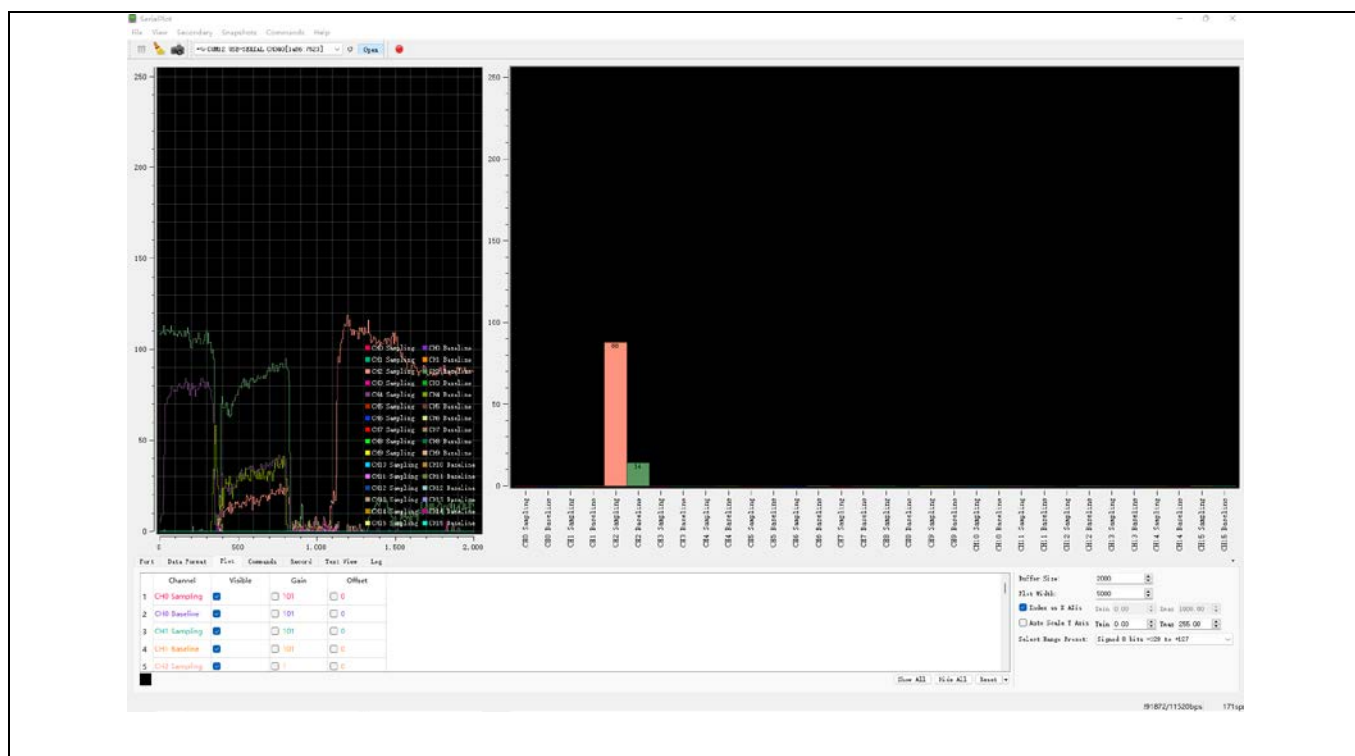


Figure 6 PC端调试程序

要使用打印功能，需要确保在tkey_demo() 中，宏定义serialplot_debug_en处于打开的状态，同时在csi_uart_putc() 中选择目标打印内容，当前支持各通道按键差值（hwOffset_data0_abs [32]）、通道采样值（hwSampling_data0 [32]）、通道基准值（hwBaseline_data0 [32]）。参数意义参看[TOUCH触摸数据读取](#)。

详情参考《AN1511 Touch Key使用串口工具波形分析使用指南》。

4.3 TOUCH触摸数据读取

参数名	意义
dwKey_Map	<p>触摸按键值。不同的按键模式该值有不同的意义。</p> <ul style="list-style-type: none"> 在First key模式时，dwKey_Map寄存器对应只有1个bit位置1。如dwKey_Map=0x01表示TCH0被按下。此模式下，同时按下多个按键时，系统将响应最先被按下那个按键。

	<ul style="list-style-type: none"> 在Multi key模式时，同时按下多个按键对应的bit位会同时置1。如dwKey_Map=0x03表示TCH0和TCH1同时被按下。 在Strongest single-key模式时，dwKey_Map寄存器对应只有1个bit位置1。如dwKey_Map=0x01表示TCH0被按下。在此模式下，同时按下多个按键时，系统将响应按键差值最大那个。
<i>hwOffset_data0_abs [32]</i>	<p>该数组对应TCH0~TCH31的按键差值。</p> <p>该数组可以通过tkey_demo()中打印到PC端。见调用tkey_demo()函数。</p>
<i>hwSampling_data0 [32]</i>	<p>该数组对应TCH0~TCH31通道采样值。</p> <p>该数组可以通过tkey_demo()中打印到PC端。见调用tkey_demo()函数。</p>
<i>hwBaseline_data0 [32]</i>	<p>该数组对应TCH0~TCH31通道基准值。</p> <p>该数组可以通过tkey_demo()中打印到PC端。见调用tkey_demo()函数。</p>

4.4 TOUCH参数

在实际使用中，默认的参数并不一定能正常使用。可以通过修改apt32f110x_chip ->drivers->tkey_parameter.c 中csi_tkey_parameter_init() 函数，调整touch配置参数。

4.4.1 常用参数

通常需要修改的参数有：

Table 1 Touch常用参数

参数名	意义	使用说明
<i>dwTkeyIoEnable</i>	Touch IO使能。 bit=1 表示使能对应的TCHx做touch key功能，低位至高位的顺序对应TCH0~TCH31	<code>dwTkeyIoEnable=0B00011111111000000;</code> 或使用函数 <code>dwTkeyIoEnable=TCH_EN(4) TCH_EN(5) TCH_EN(8) TCH_EN(9) TCH_EN(29) TCH_EN(26) TCH_EN(22) TCH_EN(23);</code>
<i>byKeyMode</i>	Touch按键模式。 0: 第一个按键的按键有效 1: 多键有效 2: 按键的最强键有效	<code>byKeyMode=2;</code> //0=First key , 1=multi key , 2=strongest single-key
<i>byTkeyGlobalSens</i>	□ Touch全局灵敏度配置 此值越大，灵敏度越高，扫描时间也越长。	<code>byTkeyGlobalSens=3;</code> 此值不建议大于10
<i>byTkeyGlobalcon</i>	Touch充电电流配置。 控制TCHx通道充电电流大小。此值越小Touch采样的数据越大。如果太小，可能造成通道扫描异常；如果太大，可能造成通道采样数值过小。	<code>byTkeyGlobalcon=4;</code> //1~7 默认值为4。
<i>byPressDebounce</i>	按下去抖设置。	<code>byPressDebounce=5;</code> //按下去抖1~10，默认配置为5
<i>byReleaseDebounce</i>	释放去抖设置。	<code>byReleaseDebounce=5;</code> //释放去抖1~10，默认配置为5
<i>byMultiTimesFilter</i>	OFFSET滤波倍数。 ≥4时，表示开启相应的倍数滤波，即当按键差值大于按键门槛值的4倍时，按键无效；小于4时表示倍数滤	<code>byMultiTimesFilter=0;</code> //默认关闭

	波关闭；默认关闭。	
<i>byBaseUpdateSpeed</i>	Baseline更新速度。 数值越小，baseline更新速度越快； 数值越大，baseline更新速度越慢； 该值设置的是每调用 n 次 <code>csi_tkey_process()</code> ，baseline更新一次。	<i>byBaseUpdateSpeed=5;</i> // 基准值每调用 5 次 <code>csi_tkey_process()</code> 更新一次。
<i>byTkeyRebuildTime</i>	长按强制更新时间设置 表示按键被按下的时间超过设定的时间时，会强制更新并释放按键。 该值设置的是调用 100xn 次 <code>csi_tkey_process()</code> 后，强制更新并释放按键。	<i>byTkeyRebuildTime=16;</i> //100x16 次 <code>csi_tkey_process()</code> 均检测到长按后，强制更新并释放按键。如 <code>csi_tkey_process()</code> 调用频次是10ms一次，该设置值意味着16s。

4.4.2 微调参数

针对某些特殊情况，如特殊的layout排列或者特殊的按键介质造成某些通道灵敏度异常，库文件提供了对特定通道进行微调的功能。

Table 2 Touch微调参数

参数名	意义	使用说明
<i>byTkeyFineTurn</i>	Touch 微调开关	<i>byTkeyFineTurn=ENABLE;</i> //开启微调功能
<i>byTkeyChxSens [x]</i>	对特定通道单独配置灵敏度，当使能了此配置，对应通道将不使用 <code>byTkeyGlobalSens</code> 中的灵敏度参数；数组的序列号表示对应的通道编号。比如 <code>byTkeyChxSens [n]</code> 表示调整TCHn通道	<i>byTkeyChxSens [0]=4;</i> //TCH0 manual Sensitivity <i>byTkeyChxSens [1]=4;</i> //TCH1 manual Sensitivity

4.4.3 特殊配置参数

Table 3 Touch特殊配置参数

参数名	意义	使用说明
<i>hwTkeyPselMode</i>	触摸参考电压源配置。	<i>hwTkeyPselMode= TK_PWRSRC_AVDD;</i> 请注意touch库是否支持FVR。

<i>hwTkeyEcLevel</i>	C0充放电电压选择，默认配置1V，可配置1V/2v/3/v/3.6v 选择的C0充放电电压必须比VDD电压小500mV以上	<i>hwTkeyEcLevel= TK_ECLVL_1V;</i>
<i>hwTkeyFvrLevel</i>	选择FVR做参考时，需根据实际VDD选择4.096V或2048V两个参考电压点，默认选择为4.096V。 当选择TK_PSEL_AVDD时，此位无效。选择的FVR参考电压必须比VDD电压小500mV以上	<i>hwTkeyFvrLevel=TK_FVR_4096V;</i> 请注意touch库是否支持FVR。

5. 硬件设计注意事项

使用触摸功能时，MCU供电电源必须稳定，建议增加稳压电路，已获得更好的稳定性及更强的抗干扰性能。

使用触摸功能时，无论选择何种电源参考，C0(PA0.5)都必须接一个103 电容到地。

TouchKey 电源参考选择：

1. 选择FVR 做电源参考PA0.0 (TCH24)需要额外接 104 电容到地。
2. 选择VDD 做电源参考PA0.0 (TCH24)可以做其他功能。(默认)