

触摸按键输入检测

1. 实验目的

- 掌握触摸按键检测电路的设计：触摸芯片的工作原理。
- 掌握触摸按键检测的程序设计及算法。

2. 实验内容

- 编写程序实现对触摸按键的检测。

3. 硬件电路设计

3.1. 开发板触摸按键硬件电路

触摸按键是电路中常用的一种开关元器件，也是一种常用的人机接口。因触摸按键具有坚固耐用、反应速度快、节省空间、易于操作等优点，而得到广泛的应用。

根据工作原理不同，触摸按键可以分为四大类：电阻式、电容式、红外线式和表面声波式。电阻式、红外线式和表面声波式触摸按键主要应用于触摸屏上，单个按键很少使用。

进取者 STC15 开发板上设计了 1 个电容式触摸按键，该触摸按键电路采用专门触摸按键检测芯片 JR223B，检测电路如下图。

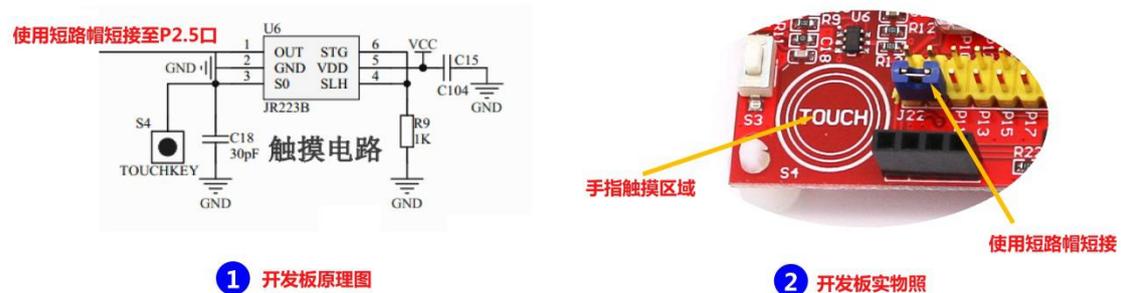


图 1：开发板触摸按键检测电路

◇ 1 个触摸按键占用的单片机的引脚如下表：

表 1：触摸按键引脚分配

| KEY | 形状 | 引脚 | 说明 |
|------|------|------|----------|
| 触摸按键 | 白色环纹 | P2.5 | 非独立 GPIO |

◇ 注：独立 GPIO 表示开发板没有其他的电路使用这个 GPIO，非独立 GPIO 说明开发板有其他电路用到了该 GPIO。进取者 STC15 开发板上 DA 检测电路用到了 P2.5 口。

3.2. 触摸按键检测 IC

JR223B 是电容式单键触摸按键 IC，电压输入范围为 2.0V~5.5V，与 JR223B 功能一样的还有触摸芯片 TTP223。下面主要介绍 JR223B 触摸按键。

JR223B 利用操作者的手指与触摸按键焊盘之间产生电荷电平来进行检测，通过监测电荷的微小变化来确定手指接近或者触摸到感应表面。没有任何机械部件，不会磨损，其感测部分可以放置到任何绝缘层（通常为玻璃或塑料材料）的后面，很容易制成与周围环境相密封的键盘。

下面介绍下 JR223B 电容式单键触摸芯片的引脚图和引脚定义。

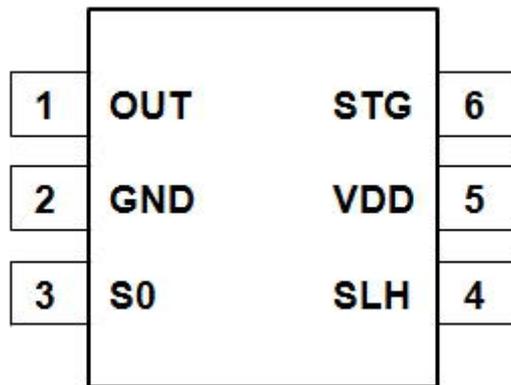


图 2: 触摸芯片引脚图

表 2: 触摸芯片引脚定义

| 引脚号 | 引脚名称 | 类型 | 详细说明 |
|-----|------|--------|-------------------|
| 1 | OUT | OutPut | 电平输出脚 |
| 2 | GND | POWER | 负电源，地 |
| 3 | S0 | InPut | 传感器输入检测脚 |
| 4 | SLH | I-PL | 输出高低电平选择，系统内置上拉电阻 |
| 5 | VDD | POWER | 正电源 |
| 6 | STG | I-PL | 模式选择脚，系统内置上拉电阻 |

■ SLH 引脚用于设置 JR223B 的输出方式。

- 1) SLH = 0: 触摸时，JR223B 的 OUT 引脚输出高电平，即高电平有效。
- 2) SLH = 1: 触摸时，JR223B 的 OUT 引脚输出地电平，即低电平有效。

- STG 引脚用于设置 JR223B 的工作模式。
- 1) STG = 0: JR223B 工作在 LEVEL Hold 模式。
- 2) STG = 1: JR223B 工作在 ON/OFF 模式。

表 3: 触摸芯片模式输出控制

| STG | SLH | 详细说明 |
|-----|-----|-------------------------------|
| 0 | 0 | LEVEL Hold 模式, 高电平有效, CMOS 输出 |
| 0 | 1 | LEVEL Hold 模式, 低电平有效, CMOS 输出 |
| 1 | 0 | ON/OFF 模式, 上电状态为 CMOS 低电平输出 |
| 1 | 1 | ON/OFF 模式, 上电状态为 CMOS 高电平输出 |

- 灵敏度调节:
 - 1) 触摸芯片 S0 引脚上会外接一个外部电容, 该电容称为灵敏度调节电容。
 - 2) 可通过选择不同容值的灵敏度调节电容实现对触摸灵敏度的调节。
 - 3) 灵敏度调节电容的容值必须在(0~50pF)范围内。
 - 4) 灵敏度调节电容的容值越大, 灵敏度就越低, 感应面板的厚度就越薄, 反之, 灵敏度调节电容的容值越小, 灵敏度就越高, 感应面板的厚度就越厚。
- 触摸芯片供电: 触摸芯片 VDD 引脚供电需稳定, 必要时可加 LDO 芯片实现。

3.3. 选择哪种模式输出控制方式

用户可以根据自己项目应用的需要, 来设置触摸芯片的 STG 和 SLH 引脚, 实现对触摸芯片模式输出控制的选择。艾克姆科技进取者 STC15 开发板将触摸芯片的 STG 和 SLH 引脚均通过下拉电阻置低, 这样, 触摸时, JR223B 的 OUT 引脚输出高电平, 不触摸时, JR223B 的 OUT 引脚输出低电平。这样设计的原因是:

- 实现触摸按键的输出和轻触按键的电路输出刚好是反的, 作为开发板, 要方便用户测试, 2 种不同类型的输出方式更方便我们使用。
 - 选择 LEVEL Hold 模式, 可保证单片机对触摸按键的检测引脚电平是恒定的, 有利于检测。
- ✧ 注: 开发板轻触按键电路是按键按下时, 电路输出低电平, 无按键按下时电路输出高电平。

4. 软件设计

4.1. 触摸按键输入检测实验

✧ 注：本节的实验源码是在“实验 2-2-2：GPIO 输入按键检测（多个 c 文件）”的基础上修改。本节对应的实验源码是：“实验 2-3：触摸按键输入检测”。

4.1.1. 工程需要用到的 c 文件

本例需要用到的 c 文件如下表所示，工程需要添加下表中的 c 文件。

表 4：实验需要用到的 c 文件

| 序号 | 文件名 | 后缀 | 功能描述 |
|----|-------|----|-------------------------|
| 1 | led | .c | 包含与用户 led 控制有关的用户自定义函数。 |
| 2 | key | .c | 包含与触摸按键检测有关的用户自定义函数。 |
| 3 | delay | .c | 包含用户自定义延时函数。 |

4.1.2. 头文件引用和路径设置

■ 需要引用的头文件

因为在“main.c”文件中使用了控制 led 的函数和延时函数（延时函数没有在 main.c 中定义），所以需要引用下面的头文件。

```
1. #include "led.h"
2. #include "delay.h"
3. #include "key.h"
```

■ 需要包含的头文件路径

本例需要包含的头文件路径如下表：

表 5：头文件包含路径

| 序号 | 路径 | 描述 |
|----|------------|--------------------------------------|
| 1 | ..\ Source | led.h、key.h 和 delay.h 头文件在该路径，所以要包含。 |
| 2 | ..\User | 15W4KxxS4.h 头文件在该路径，所以要包含。 |

MDK 中点击魔术棒，打开工程配置窗口，按照下图所示添加头文件包含路径。

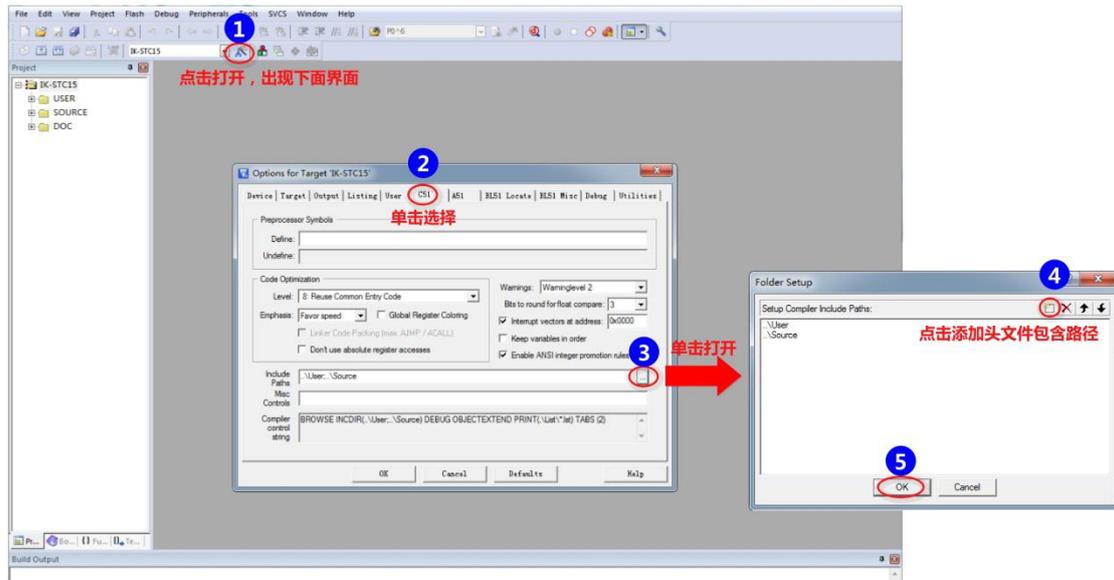


图 3：添加头文件包含路径

4.1.3. 编写代码

首先在 key.h 中，定义寄存器位变量 P2^5，引用延时函数的头文件，声明按键检测函数供外部调用。代码如下。

```

1. #include "delay.h"
2.
3. /*****
4. 引脚别名定义
5. *****/
6. sbit TOUCH_KEY=P2^5; //触摸按键 S4 用 IO 口 P2.5
7.
8. extern void KEY_Scan(void);

```

然后，在 key.c 文件中编写一个触摸按键检测函数 KEY_Scan，代码如下。

程序清单：触摸按键检测函数

```

1. /*****
2. 功能描述：触摸按键检测
3. 入口参数：无
4. 返回值：无
5. *****/
6. void KEY_Scan(void)
7. {
8.     if(TOUCH_KEY == 1) //检测触摸按键 S4 对应引脚 P2.5 是否是低电平（有手指触摸按键 S4 触摸区域，引脚为高电平）
9.     {

```

```

10.     delay_ms(10);           //软件延时 10ms, 如果延时后按键 S4 的电平依然没有变化,
    说明按键确实被有效操作, 简称按键消抖
11.     if(TOUCH_KEY== 1)      //检测触摸按键 S4 对应引脚 P2.5 是否依然是高电平
12.     {
13.         led_on(LED_1);      //点亮蓝色指示灯 DS1
14.         while(TOUCH_KEY == 1) //等待按键 S4 释放, 即如果 P2.5 一直为高电平,
    会一直执行空命令
15.         {
16.             ;                //条件 TOUCH_KEY == 1 成立, 会执行这个空命令
17.         }
18.         led_off(LED_1);     //熄灭蓝色指示灯 DS1
19.     }
20. }
21. }

```

因为按键检测函数 KEY_Scan 中使用了控制 led 的函数、KEY 的位变量和延时函数, 所以需要在 key.c 文件中引用下面的头文件。

```

1. #include "led.h"
2. #include "key.h"

```

最后, 在主函数中先对 P0.6、P0.7 和 P2.5 口进行模式配置, 后主循环中调用触摸按键检测函数, 可观察用手指触摸按键 S4 的触摸区域蓝色指示灯 DS1 亮。

代码清单：主函数

```

1. int main(void)
2. {
3.     //////////////////////////////////////
4.     //注意: STC15W4K32S4 系列的芯片, 上电后所有与 PWM 相关的 IO 口均为
5.     //      高阻态, 需将这些口设置为准双向口或强推挽模式方可正常使用
6.     //相关 IO: P0.6/P0.7/P1.6/P1.7/P2.1/P2.2
7.     //      P2.3/P2.7/P3.7/P4.2/P4.4/P4.5
8.     //////////////////////////////////////
9.     P0M1 &= 0x3F;   P0M0 &= 0x3F;   //设置 P0.6、P0.7 为准双向口
10.    P2M1 &= 0xDF;   P2M0 &= 0xDF;   //设置 P2.5 为准双向口
11.    //P2M1 |= 0x20; P2M0 &= 0xDF;   //设置 P2.5 为高阻输入
12.
13.    while(1)
14.    {
15.        KEY_Scan(); //按键输入检测函数
16.    }
17. }

```

4.1.4. 实验步骤

1. 解压“…\第3部分：配套例程源码\1-基础实验程序\”目录下的压缩文件“实验2-3：触摸按键输入检测”，将解压后得到的文件夹拷贝到合适的目录，如“D:\STC15”。
2. 启动 Keil C51。
3. 在 Keil C51 中执行“Project→Open Project”打开“…\TOUCH_KEY\projec”目录下的工程“TOUCH_KEY.uvproj”。
4. 点击编译按钮编译工程。注意查看编译输出栏，观察编译的结果，如果有错误，修改程序，直到编译成功为止。编译后生成的 HEX 文件“TOUCH_KEY.hex”位于工程目录下的“Output”文件夹中。
5. 打开 STC-ISP 软件下载程序。下载使用内部 IRC 时钟，IRC 频率选择为 11.0592MHZ。
6. 程序运行后，用手指触摸按键 S4 的触摸区域可以观察到兰色 LED 灯被点亮。手指离开按键 S4 的触摸区域，兰色 LED 灯熄灭。