

按键

前言：

相信大家经过例程 1 点亮 LED 实验后对 CC2540 的编程以及 IAR 的编译方法有一定的了解。我们来讲解一下 BlueTooth 模块的按键实验，按键是实现人机交互必不可少的东西，我们实验就用来实现按键控制 LED。

实验现象：依次按下按键 S1 控制 LED1 的亮和灭

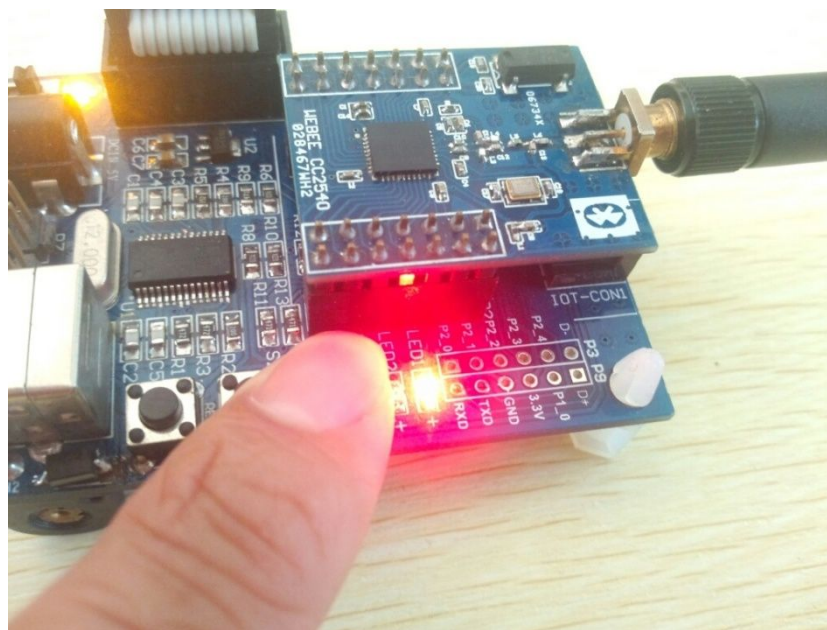


图 2.4 通过 S1 控制 LED1 的亮灭

实验讲解：

我们先来看看 BlueTooth 底板的 KEY 和 LED 部分原理图，如图 2.5 所示。

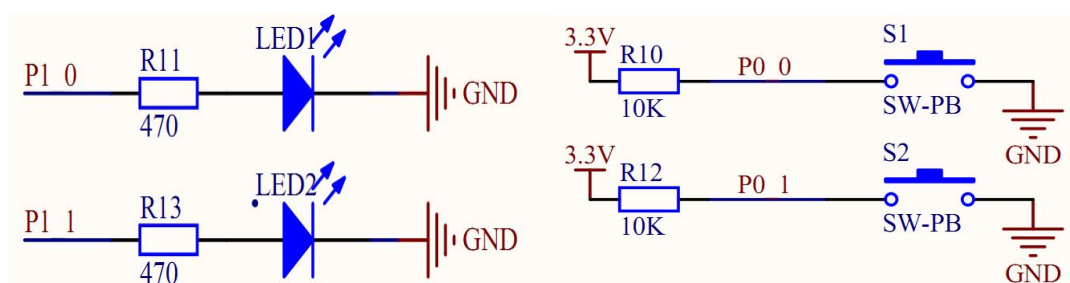


图 2.5 LED、按键电路图

CC2540 的 IO 口配置我们需要配置三个寄存器 **P1SEL** 、 **P1DIR** 、 **P1INP**。功能如下表所示： （详细参考 CC253x and CC2540 User's Guide.pdf）

表 2.2 CC2540 P0 口寄存器

P0SEL(0XF3)	P0 [7:0]功能设置寄存器，默认设置为普通 I/O 口
P0INP(0X8F)	P0[7:0] 作输入口时的电路模式寄存器
P0(0X80)	P0[7:0] 可位寻址的 I/O 寄存器
P0DIR(0XFD)	P0 口输入输出设置寄存器，0：输入 ， 1：输出

- * **P1SEL** （0：普通 IO 口 1：第二功能）
- * **P1DIR** （0：输入 1：输出 ）
- * **P1INP** （0：上拉/下拉 1：三态 ）

按照表格寄存器内容，我们对 LED1 和按键 S1，也就是 P1.0 和 P0.4 口进行配置，当 P1.0 输出低电平时 LED1 被点亮, S1 按下时 P0.4 被拉低。所以配置如下：

LED1 初始化：

```
P1SEL &= ~0x01; //作为普通 IO 口
P1DIR |= 0x01; //P1_0 定义为输出
P1INP &= ~0x01; //打开上拉
```

按键 S1 初始化：

```
P0SEL &= ~0x01; //设置 P0_0 为普通 IO 口
P0DIR &= ~0x01; //按键在 P0_0 口，设置为输入模式
P0INP &= ~0x01; //打开 P0_0 上拉电阻, 不影响
```

由于 CC2540 寄存器初始化时默认是：

```
P1SEL = 0x00;
P1DIR = 0xff;
P1INP = 0x00;
```

所以 IO 口初始化我们可以简化初始化指令：

```
P1DIR |= 0x01;    //P1_0 定义为输出
P0DIR &= ~0x01;   //按键在 P0_0 口，设置为输入模式
```

源程序代码（全）

```
/******
程序描述：依次按下按键 S1 控制 LED1 的亮和灭
*****/

#include <ioCC2540.h>

#define uint unsigned int
#define uchar unsigned char

//定义控制 LED 灯的端口
#define LED1 P1_0        //LED1 为 P1.0 口控制
#define KEY1 P0_0        //KEY1 为 P0.0 口控制

//函数声明
void Delayms(uint);      //延时函数
void InitLed(void);      //初始化 LED1
void KeyInit();           //按键初始化
uchar KeyScan();         //按键扫描程序

/******
```

延时函数

```
*****/  
void Delayms(uint xms)    //i=xms 即延时 i 毫秒  
{  
    uint i, j;  
    for(i=xms;i>0;i--)  
        for(j=587;j>0;j--);  
}
```

```
/*****
```

LED 初始化函数

```
*****/  
void InitLed(void)  
{  
    P1DIR |= 0x01;    //P1_0 定义为输出  
    LED1 = 0;        //LED1 灯熄灭  
}
```

```
/*****
```

按键初始化函数

```
*****/  
void InitKey()  
{  
    POSEL &= ~0X01;    //设置 P00 为普通 IO 口  
    P0DIR &= ~0X01;    //按键在 P00 口，设置为输入模式  
    P0INP &= ~0x01;    //打开 P00 上拉电阻, 不影响  
}
```

```
/*****
```

按键检测函数

```
*****/  
uchar KeyScan(void)  
{  
    if(KEY1==0)  
    {Delayms(10);  
        if(KEY1==0)  
        {  
            while(!KEY1); //松手检测  
            return 1;      //有按键按下  
        }  
    }  
    return 0;              //无按键按下  
}  
/*****
```

主函数

```
*****/  
void main(void)  
{  
    InitLed(); //调用初始化函数  
    InitKey();  
    while(1)  
    {  
        if(KeyScan()) //按键改变 LED 状态  
            LED1=~LED1;  
    }  
}
```

实验图片：

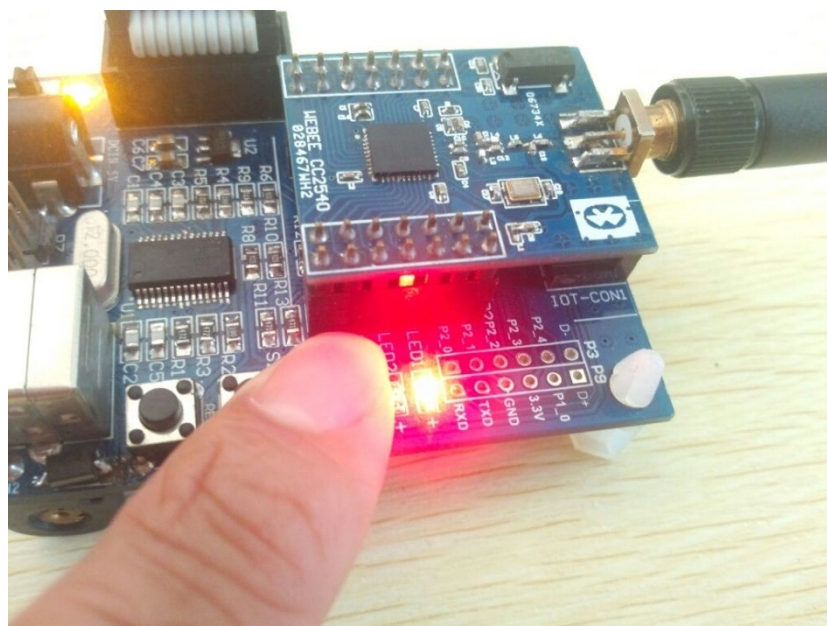


图 2.6 按键控制 LED