

# 实验一 MSP430 GPIO 原理以及接口技术应用报告

## 1. 实验要求

具体要求如下

- 1) 用软件延时产生时长为 1s 钟的子函数;
- 2) 用查询方式检测 GPIO 的按键输入
- 3) 按键 S3, 按一下, 点亮一盏 LED 灯亮 1s, 灭 1 秒, 连续闪烁; 再按一下, 两盏 LED 闪; 依次按五下, 五盏 LED 闪, 按键次数与闪烁的灯数一致, 不能错乱;
- 4) 按 S4 全部 LED 关灭;
- 5) 在程序中加入按键防抖动策略

## 2. 设计思路

根据实验要求, 选择 P4.0 与 P4.1 口作为按键输入, P4.5, P4.6, P4.7, P5.7, P8.0 口作为 LED 指示灯的输出。在对各端口初始化设置后, 在主程序内重复执行以下逻辑:

使用变量 cnt 记录当前时刻应该闪烁的 led 的数目。

循环:

判断按键 P4.0 是否被按下, 延时一段时间, 若此按键仍处于按下状态, 则 cnt+1 (若 cnt>5 则重新从 1 盏 led 开始)。

判断按键 P4.1 是否被按下, 延时一段时间, 若此按键仍处于按下状态, 则 cnt 清零。

根据 cnt 的值将前几盏 led 点亮，剩下的熄灭。

延时 1s。

将所有 led 灯全部熄灭。

延时 1s。

循环至判断按键。

### 3. 实验代码

```
1 /*
2  * main.c
3  */
4 #include <msp430.h>
5 #include <stdint.h>
6 typedef struct
7 {
8     const volatile uint8_t* PxIN;    //定义一个不会被编译的无符号字符型变量
9     volatile uint8_t* PxOUT;
10    volatile uint8_t* PxDIR;
11    volatile uint8_t* PxREN;
12    volatile uint8_t* PxSEL;
13 } GPIO_TypeDef;
14
15 const GPIO_TypeDef GPIO4 =
16 { &P4IN, &P4OUT, &P4DIR, &P4REN, &P4SEL};
17 const GPIO_TypeDef GPIO5 = {&P5IN, &P5OUT, &P5DIR, &P5REN, &P5SEL};
18
19 const GPIO_TypeDef GPIO8 = {&P8IN, &P8OUT, &P8DIR, &P8REN, &P8SEL};
20 const GPIO_TypeDef* LED_GPIO[5] = {&GPIO4, &GPIO4, &GPIO4, &GPIO5, &GPIO8};
21 const uint8_t LED_PORT[5] = {BIT5, BIT6, BIT7, BIT7, BIT0};
22 int cnt = 0;
23 int main( void )
24 {
25     // Stop watchdog timer to prevent time out reset
26     WDCTL = WDTPW + WDTHOLD;
27     while(BAKCTL & LOCKIO) // Unlock XT1 pins for operation
28         BAKCTL &= ~(LOCKIO);
29     UCSCTL6 &= ~XT1OFF; //启动XT1
30     while (UCSCTL7 & XT1LFOFF6) //等待XT1稳定
31         UCSCTL7 &= ~(XT1LFOFF6);
32     UCSCTL4 = SELA__XT1CLK + SELS__REFOCLK + SELM__REFOCLK; //时钟设为XT1，频率较低，方便软件延时
33     int i;
34     for(i=0;i<5;++i)
35         *LED_GPIO[i]->PxDIR |= LED_PORT[i]; //设置各LED灯所在端口为输出方向
36     P4OUT |= BIT0; // 选中P4.0为输出方式
37     P4REN |= BIT0; // P4.0使能
38     P4OUT |= BIT1; // 选中P4.1为输出方式
```

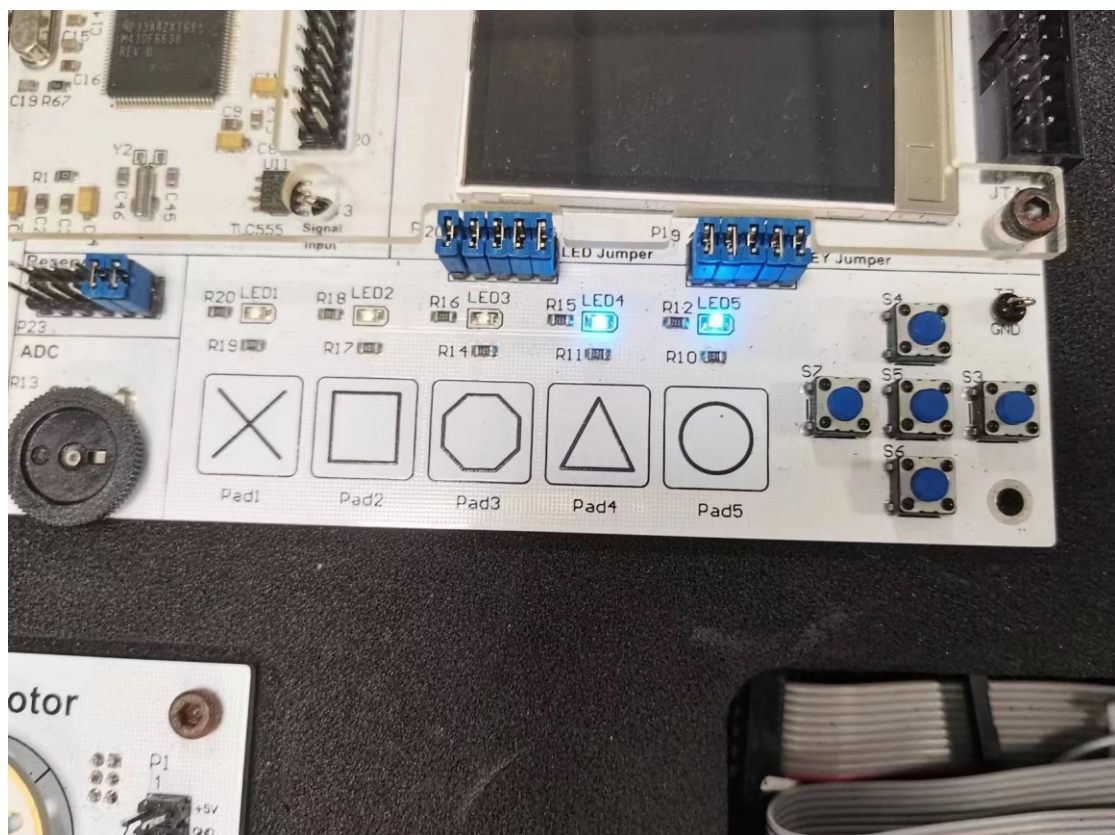
```

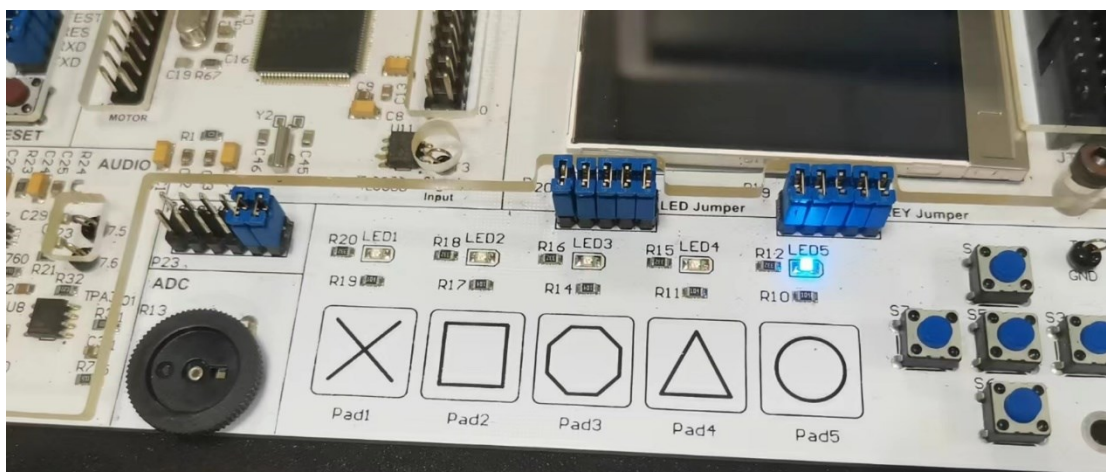
38 P4OUT |= BIT1, // 选中P4.1引脚驱动LED
39 P4REN |= BIT1; // P4.1使能
40 cnt = 0;
41 while(1)
42 {
43     if (P4IN & BIT0) cnt = cnt;
44     else { //P4.0口:按下灯多亮一个
45         __delay_cycles(327); //按键防抖动 延时10ms
46         if (P4IN & BIT0) cnt = cnt;
47         else cnt = (cnt+1)%6;
48     }
49     if (P4IN & BIT1) cnt=cnt;
50     else { //P4.1口:按下灯全灭
51         __delay_cycles(327); //按键防抖动 延时10ms
52         if (P4IN & BIT1) cnt=cnt;
53         else cnt = 0;
54     }
55 }
56 for(i=0;i<5;++i) //点亮要被点亮的灯们
57     if (i<cnt)
58         *LED_GPIO[i]->PxOUT |= LED_PORT[i];
59     else
60         *LED_GPIO[i]->PxOUT &= ~LED_PORT[i];
61     __delay_cycles(32760); //延时大约1s
62     for(i=0;i<5;++i) //LED全部关掉
63         *LED_GPIO[i]->PxOUT &= ~LED_PORT[i];
64     __delay_cycles(32760); //延时大约1s
65 }
66 }
67

```

## 4. 实验结果

运行程序，可实现实验要求中，闪烁的 led 数目依次增加，按下按键后全部熄灭的功能，以下为部分记录照片：





## 5. 分析与思考

经排查，一开始无法正确运行案例程序是由于软件处于在线调试状态，未点击运行图标。

本实验主要使用了 GPIO 口，复习输入与输出模式下引脚各寄存器的原理、配置和使用，并稍微接触时钟源，进行延时函数的基础调用。