

6.外部中断控制LED倒流水灯

提示：重点看红色字体部分的代码，其它的和上一章一样，如果掌握不熟的可再重看一次

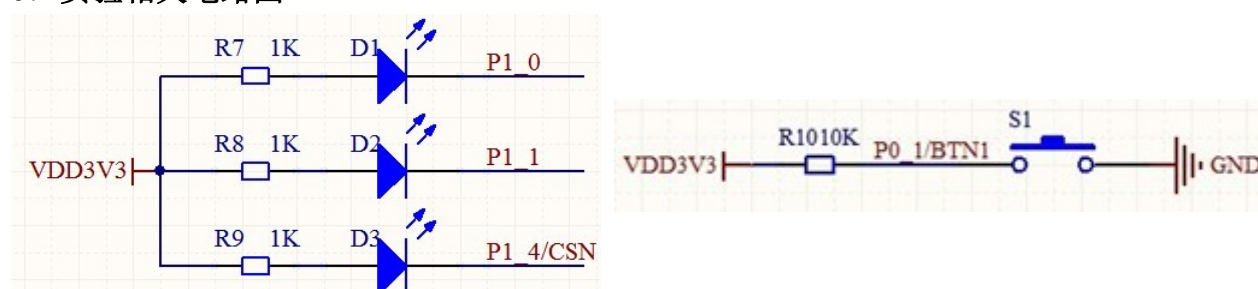
1. 实验目的

- 1) 通过实验掌握 CC2530 芯片 GPIO 的配置方法及中断概念与方法
- 2) 掌握 Led 驱动电路及开关 Led 的原理
- 3) 通过按键 S1 产生外部中断改变 LED1、LED2、LED3 执行倒流水灯

2. 实验设备

- 硬件：PC 机 一台
EB2530（底板、核心板、仿真器、USB线） 一套
- 软件：2000/XP/win7 系统，IAR 8.10 集成开发环境

3. 实验相关电路图



由于发光二极管单向导电特性，即只有在正向电压（二极管的正极接正，负极接负）下才能导通发光。所以 P1.0、P1.1、P1.4 引脚输出低电平 LED 亮，引脚输出高电平 LED 熄灭。当按下按键会产生中断，P0IFG 寄存器为中断状态标志。

4. 实验相关寄存器

CC2530外部中断需要配置P0IEN、PICTL、P0IFG、IEN1寄存器。P1_0的配置请参考前面章节，外部中断寄存器说明如下表所示：

| 寄存器 | 作用 | 描述 |
|-------------|------------------------|--|
| P0IEN(0xAB) | 端口 0 中断屏蔽 | 端口P0.7到P0.0中断使能 0：中断禁用 1：中断使能。 |
| PICTL(0x8C) | 端口中断控制 P0ICON(Bit0) | 端口0，7到0输入模式下的中断配置。该位为所有端口0的输入选择中断请求条件。 0：输入的上升沿引起中断 1：输入的下降沿引起中断 |
| P0IFG(0x89) | 端口 0 中断状态标志 | 端口 0，位 7 到 0 输入中断状态标志。当输入端口中断请求未决信号时，其相应的标志位将置 1。 |
| IEN1 (0xB8) | 中断使能 1 P0IE(Bit5) | 端口0中断使能 0：中断禁止 1：中断使能 |

按照表格寄存器的内容，对P1.0口进行配置，当P1.0输出低电平时D1被点亮。S1按下时P0.1产生外部中断从而控制LED1的亮灭所以配置如下：

```
P1DIR |= 0x01;    // P1.0 定义为输出
```

按键 S1 配置如下:

```
POIEN |= 0x2;    // P0.1 设置为中断方式 1: 中断使能
```

```
PICTL |= 0x2;    //下降沿触发
```

```
IEN1 |= 0x20;    //允许 P0 口中断;
```

```
POIFG = 0x00;    //初始化中断标志位
```

```
EA = 1;          //打开总中断
```

5. 源码分析

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

```
* 文件 名: main.c
```

```
* 作    者: Andy
```

```
* 修    订: 2013-01-08
```

```
* 版    本: 1.0
```

```
* 描    述: 通过按键 S1 产生外部中断改变 LED 倒流水灯
```

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

```
#include <ioCC2530.h>
```

```
typedef unsigned char uchar;
```

```
typedef unsigned int  uint;
```

```
#define LED1 P1_0    //定义 P1.0 口为 LED1 控制端
```

```
#define LED2 P1_1    //定义 P1.1 口为 LED2 控制端
```

```
#define LED3 P1_4    //定义 P1.4 口为 LED3 控制端
```

```
#define KEY1 P0_1    //定义 P0.1 口为 S1 控制端
```

```
uchar KeyValue=0;    //产生中断保存中断状态, 1 为产生按键中断
```

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

```
* 名    称: DelayMS()
```

```
* 功    能: 以毫秒为单位延时, 系统时钟不配置时默认为 16M(用示波器测量相当精确)
```

```
* 入口参数: msec 延时参数, 值越大, 延时越久
```

```
* 出口参数: 无
```

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

```
void DelayMS(uint msec)
```

```
{
```

```
    uint i, j;
```

```
    for (i=0; i<msec; i++)
```

```

        for (j=0; j<535; j++);
    }

/*****
* 名    称: InitLed()
* 功    能: 设置 LED 灯相应的 IO 口
* 入口参数: 无
* 出口参数: 无
*****/
void InitLed(void)
{
    P1DIR |= 0x13;          //P1.0、P1.1、P1.4 定义为输出
    LedOnOrOff(1);          //使所有 LED 灯默认为熄灭状态
}

/*****
* 名    称: InitKey()
* 功    能: 设置 KEY 相应的 IO 口,采用中断方式
* 入口参数: 无
* 出口参数: 无
*****/
void InitKey()
{
    P0IEN |= 0x2;           // P0.1 设置为中断方式 1: 中断使能
    PICTL |= 0x1;           //下降沿触发
    IEN1 |= 0x20;           //允许 P0 口中断
    P0IFG = 0x00;           //初始化中断标志位
    EA = 1;                 //打开总中断
}

/*****
* 名    称: P0_ISR(void) 中断处理函数
* 描    述: #pragma vector = 中断向量,紧接着是中断处理程序
*****/
#pragma vector = POINT_VECTOR
__interrupt void P0_ISR(void)
{
    if(P0IFG > 0)           //按键中断

```

```

    {
        DelayMS(10);          //延时去抖
        if(P0IFG > 0)          //按键中断
        {
            KeyValue = 1;      //产生中断保存中断状态
        }
    }

    P0IFG = 0;                //清中断标志
    P0IF = 0;                 //清端口 0 中断标志
}

/*****
* 程序入口函数
*****/
void main(void)
{
    InitLed();                //设置 LED 灯相应的 IO 口
    InitKey();                //设置 KEY 相应的 IO 口

    while(1)
    {
        if(KeyValue == 1)      //如果按键按下 LED3、LED2、LED1 将倒序流水灯闪烁
        {
            LED3 = !LED3;
            DelayMS(200);
            LED2 = !LED2;
            DelayMS(200);
            LED1 = !LED1;
            DelayMS(200);
            KeyValue = 0;
        }
    }
}

```

6. 实验步骤

编译下载程序后按下按键 S1 产生外部中断改变 LED 倒流水灯