

外部中断-按键检测

1. 实验目的

- 1) 通过实验掌握 CC2540 芯片 GPIO 的配置方法
- 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 状态

2. 实验设备

硬件：

PC 机一台

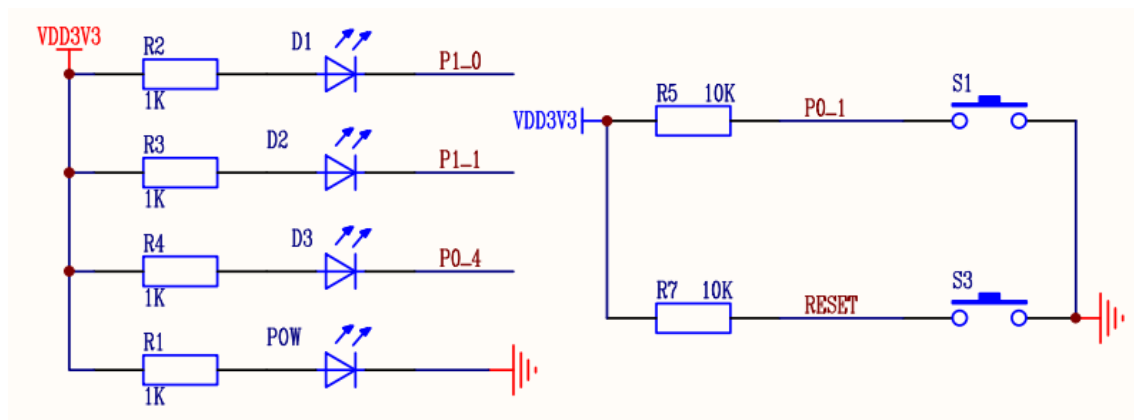
SmartRF cc254x (底板、核心板、仿真器、USB 线) 一套

软件：

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。当按下按键会产生外部中断，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。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
P1CTL(0x8C)	端口中断控制 P0ICON(Bit0)	端口 0, 7 到位 0 输入中断状态标识。当输入端口中断请求未决信号时，其相应的标识位将置 1。
P1IFG(0x89)	端口 0 中断状态标识	端口 0, 位 7 到位 0 输入中断状态标识。当输入端口中断请求未决信号时，其相应的标识位将置 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 输出高电平时 D1 被点亮。S1 按下时 P0.1 产生外部中断从而控制 LED1 的亮灭。

LED1 所以配置如下：

P1DIR|= 0x01; // P1.0 定义为输出

按键 S1 配置如下：

P0IEN |=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

P0IFG =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: 创思通信 QQ:2357481431  
* 网站: 公司网站: http://www.csqsm.com/  
店铺:http://520ic.taobao.com/  
* 修订: 2016-06-08  
* 版本: 1.0  
* 描述: 通过按键 S1 产生外部中断改变 LED1 状态  
*/  
  
#include <ioCC2540.h>  
  
#define uchar unsigned char  
#define uint unsigned int  
  
#define LED1 P1_0    // P1.0 口控制 LED1  
#define KEY1 P0_1    // P0.1 口控制 S1  
  
/*  
* 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 len)  
{  
while(len--)  
for (inti=0; i<535; i++)  
}  
  
/*  
* 名称: InitLed()
```

* 功能: 设置 LED 灯相应的 IO 口

* 入口参数: 无

* 出口参数: 无

*****/

```
voidInitLed(void)
{
    P1DIR |= 0x01; //P1.0 定义为输出口
    LED1 = 0;      //LED1 灯灭
}
```

*****/

* 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

*****/

```
voidInitKey()
{
    P0IEN |= 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;
    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 = P0INT_VECTOR
__interrupt void P0_ISR(void)
{
    DelayMS(10); //延时去抖
    LED1 = ~LED1; //改变 LED1 状态
```

```
P0IFG = 0;    //清中断标志
P0IF = 0;    //清中断标志
}

/*****
* 程序入口函数
*****/
void main(void)
{
    InitLed(); //设置 LED 灯相应的 IO 口
    InitKey(); //设置 S1 相应的 IO 口
    while(1);
}
```