

# 按键控制LED灯光强度

## 1. 实验目的

- 1)、通过实验掌握CC2530芯片GPIO的配置方法
- 2)、掌握LED驱动电路及开关LED的原理
- 3)、掌握按键的使用，实现人机交互
- 4)、掌握PWM原理

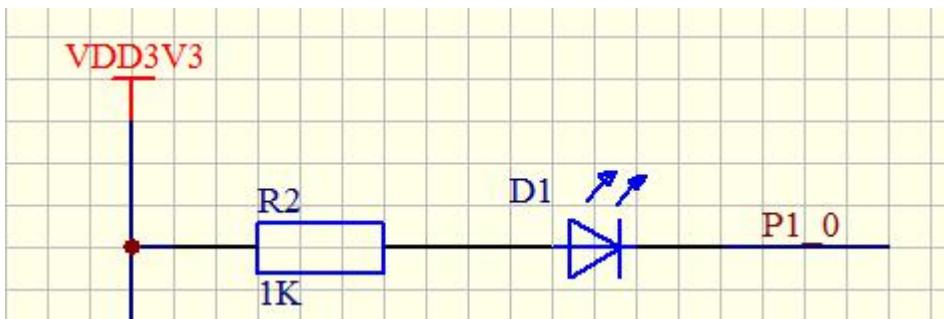
## 2. 实验设备

硬件：PC 机 一台  
EB2530（底板、核心板、仿真器、USB 线）一套

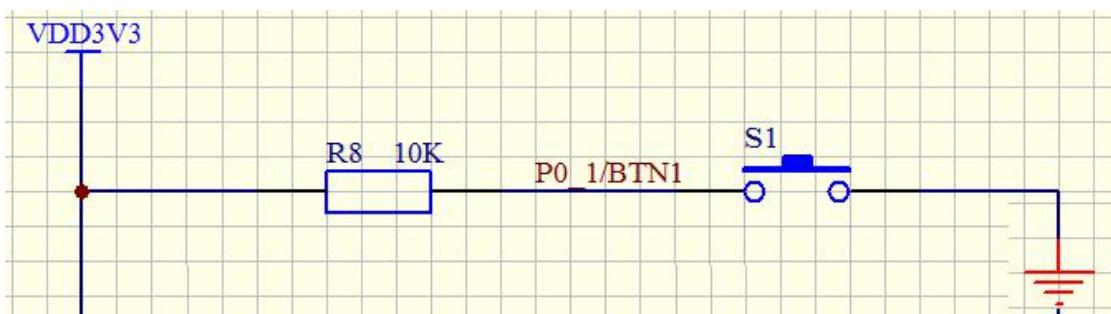
软件：2000/XP/win7 系统，IAR 8.10 集成开发环境

## 3. 实验相关电路图

LED1 电路



KEY1 电路：



发光二极管是属于二极管的一种，具有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。P1.0引脚接发光二极管(D1)的负极，所以P1.0引脚输出低电平D1亮，P1.0引脚输出高电平D1熄灭。

当P0\_1引脚为低电平时说明按键KEY1被按下，高电平时为抬起状态。

## 4. 实验相关寄存器

操作 GPIO 口需要了解的寄存器，P0、P1 相同以下只列出 P1 的寄存器，在头文件“ioCC2530.h”里对所有的寄存器都有定义，如端口 0 的方向选择，即为:PODIR。如下表所示:

寄存器	作用	描述
P1 (0x90)	端口1	端口1。通用I / O端口。可以从SFR位寻址。
P1SEL (0xF4)	端口1 功能选择	P1.7 到P0.0功能选择 0: 通用I / O 1: 外设功能
P1DIR (0xFE)	端口1 方向	P1.7到P1.0的I/O方向 0: 输入 1: 输出
P1INP (0xF6)	端口1 输入模式	P1.7到P1.2的I/O输入模式。由于P1.0 和P1.1 没有上拉/下拉功能，P1INP暂时不需要配置，了解一下为后面的实验打下基础 0: 上拉/下拉(见P2INP (0xF7) - 端口2输入模式) 1: 三态

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。 所以配置如下：

```
P1DIR |= 0x01; // P1.0 定义为输出
LED1 = 1; // LED1灯初始化为熄灭状态
```

按键key1配置如下：

```
POSEL &= ~0x01; //设置 P0.1 为普通 IO 口
PODIR &= ~0x01; //按键接在P0.1口上，设P0.1为输入模式
```

## 5. PWM 原理

PWM 就是 Pulse-Width Modulation (脉冲宽度调制)，这里面的脉冲宽度即在一个周期内输出高电平的时间，假如说周期 T=64US，脉冲宽度 D=32us，则占空比=D/T=32/64=50%，脉冲宽度调整就是占空比的调整。

## 6. 代码分析

```
/*
    名称: 按键 S1 调 LED1 亮度
    内容: 通过按键调节 LED 的亮度
*/
//包含头文件，一般情况不需要改动，头文件包含特殊功能寄存器的定义
#include <ioCC2530.h>
#define uint unsigned int
#define uchar unsigned char

//定义控制灯的端口
```

```

#define LED1 P1_0 //定义 LED1 为 P10 口控制
#define KEY1 P0_1 //KEY1 为 P0.1 口控制

//函数声明
void Delay(unsigned int t); //函数声明
void InitIO(void); //初始化 LED 控制 IO 口函数
void InitKey();

/*-----  

   主函数  

-----*/
void main (void)
{
    unsigned int CYCLE=600;//PWM 的周期
    unsigned int PWM_LOW=0;//PWM 低电平时间
    InitIO();
    InitKey();
    while (1) //主循环
    {
        if(KEY1==0)
        {
            //S1 按下,一直接着哦。
            PWM_LOW++;
            if(PWM_LOW>CYCLE)
            {
                PWM_LOW=CYCLE-1;
            }
        }
        else
        {
            PWM_LOW--;
            if(PWM_LOW<1)
            {
                PWM_LOW=1;
            }
        }

        if(PWM_LOW==1)
        {
            LED1=1;
        }
        else if(PWM_LOW==CYCLE)
        {
            LED1=0;
        }
    }
}

```

```

    }

    else if(CYCLE>PWM_LOW)
    {
        LED1=0;
        Delay(PWM_LOW);
        LED1=1;
        Delay(CYCLE-PWM_LOW);
    }
}

}

/*-----*
延时函数，含有输入参数 unsigned int t，无返回值
unsigned int 是定义无符号整形变量，其值的范围是
0~65535
-----*/
void Delay(unsigned int t)
{
    while(t)
    {
        t--;
        t++;
        t--;
    }
}

/*****//**
//初始化 I/O 口程序
******/
void InitIO(void)
{
    P1DIR |= 0x13; //P10、P11、P14 定义为输出
}

/*****//**
按键初始化函数
******/
void InitKey()
{
    POSEL &= ~0x01; //设置 P0.1 为普通 I/O 口
    PODIR &= ~0X01; //按键接在 P0.1 口上，设 P0.1 为输入模式
}

```

## 7. 实验步骤

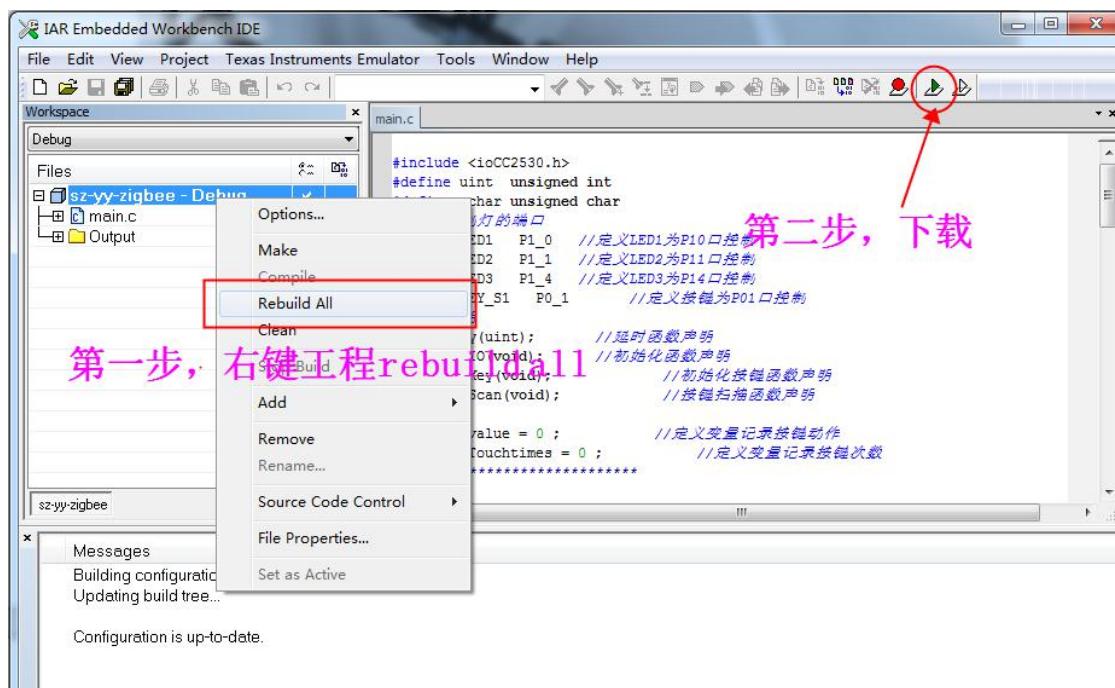
1)、双击打开代码工程。

第三章.基础实验 > 实验24、CC2530 按键控制灯光强度 > sz-yy-zigbee >					搜索"sz-yy-zigbee"
*	名称	修改日期	类型	大小	
	Debug	2020/4/24 21:42	文件夹		
	settings	2020/4/24 21:42	文件夹		
	main.c	2018/6/25 9:37	sourceinsight.c_f...	2 KB	
	sz-yy-zigbee.dep	2018/6/25 9:50	DEP 文件	3 KB	
	sz-yy-zigbee.ewd	2015/1/31 16:09	EWD 文件	34 KB	
	sz-yy-zigbee.ewp	2015/1/31 21:26	EWP 文件	55 KB	
	sz-yy-zigbee.eww	2015/1/31 14:27	IAR IDE Worksp...	1 KB	

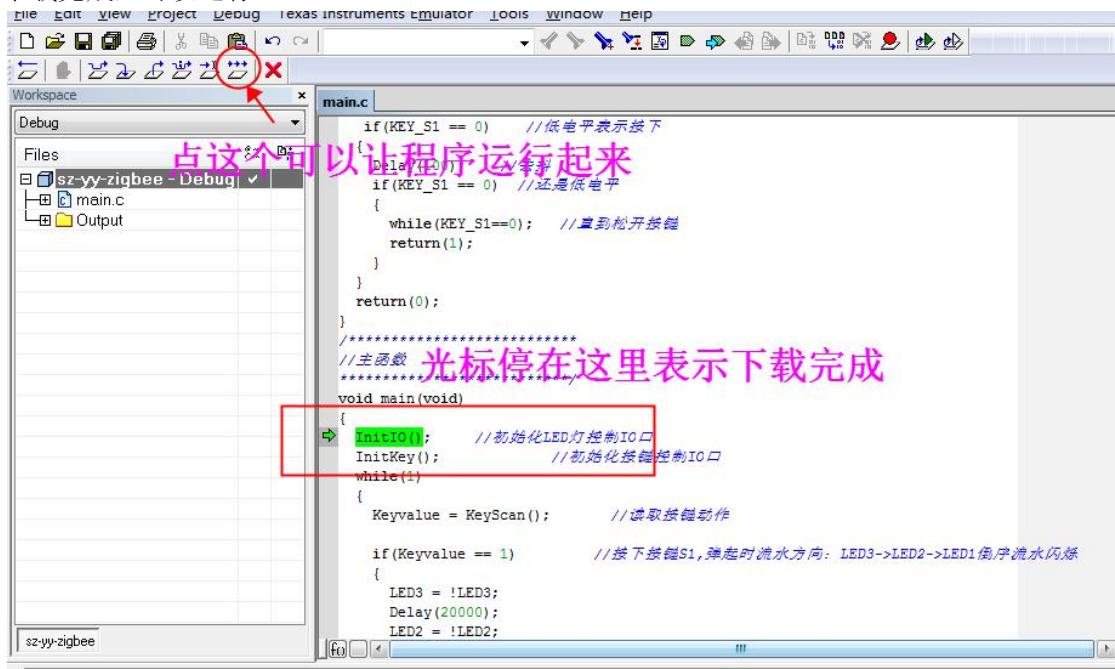
2)、通过仿真器把开发板接到电脑上。如图：



3)、编绎下载。



程序已下载完成，可以运行：



4)、运行程序。

IAR Embedded Workbench IDE

File Edit View Project Debug Texas Instruments Emulator Tools Window Help

Workspace

Debug

Files

sz-yy-zigbee - Debug

- main.c
- Output

**运行程序**

```

//定义控制灯的端口
#define LED0 P1_0 //定义LED1为P10口控制

//函数声明
void Delay(unsigned int t); //函数声明
void InitIO(void); //初始化LED控制IO口函数

/*
主函数
*/
void main (void)
{
    unsigned int CYCLE=600, PWM_LOW=0; //定义周期并赋值
    InitIO();
    while (1) //主循环
    {
        PWM_LOW+=1;
        if(PWM_LOW>CYCLE)
            PWM_LOW=0;
    }
}

```

5)、效果

