/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530 IO口操作（控制LED闪烁）

\* 程序描述：本程序用于演示CC2530 IO口的操作，通过程序控制LED灯闪烁。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define uint unsigned int

#define uchar unsigned char

//定义控制LED灯的端口

#define LED1 P1\_0 //定义LED1为P10口控制

#define LED2 P1\_1 //定义LED2为P11口控制

//函数声明

void Delay(uint); //延时函数

void Initial(void); //初始化P1口

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：Delay

\* 入口参数：n

\* 参数类型：uint

\* 参数描述：无

\* 函数功能：延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Delay(uint n)

{

uint i,t;

for(i = 0;i<5;i++)

for(t = 0;t<n;t++);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_Initial

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_Initial(void)

{

P1DIR |= 0x03; //P1\_0、P1\_1定义为输出

LED1 = 1; //LED1灯熄灭

LED2 = 1; //LED2灯熄灭

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(void)

{

LED\_Initial (); //调用LED初始化函数

LED1 = 0; //LED1点亮

LED2 = 0; //LED2点亮

while(1)

{

LED2 = !LED2; //LED2闪烁

Delay(50000);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530 定时器1运行模式和分频设置

\* 程序描述：本程序用于演示与CC2530定时器1相关的操作，主函数开始对定时器1

\* 的溢出次数进行计数。当定时器1溢出20次后LED2点亮，并持续20次

\* 定时器溢出所需时长后熄灭LED2，如此往复。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define uint unsigned int

#define uchar unsigned char

#define LED1 P1\_0

#define LED2 P1\_1

uint counter; //统计溢出次数

void LED\_init(void);

void timer1\_init(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/void LED\_init(void)

{

//初始化P1

P1DIR = 0x03; //P1\_0 P1\_1为输出

LED1 = 1;

LED2 = 1; //熄灭LED

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：timer1\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化定时器1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/void timer1\_init(void)

{

// 128频模式下的三种定时器运行状态设置

// T1CTL = 0x0D; //中断无效，自由运行(0x0000->0xffff);

// T1CTL = 0x0E; //中断无效, 模计数(0x0000->T1CC0);

// T1CTL = 0x0F; //中断无效, 上下计数(0x0000->T1CC0->0x0000);

// 32分频模式下的三种定时器运行状态设置

// T1CTL = 0x09; //中断无效，自由运行(0x0000->0xffff);

// T1CTL = 0x0A; //中断无效, 模计数(0x0000->T1CC0);

// T1CTL = 0x0B; //中断无效, 上下计数(0x0000->T1CC0->0x0000);

// 8分频模式下的三种定时器运行状态设置

// T1CTL = 0x05; //中断无效，自由运行(0x0000->0xffff);

// T1CTL = 0x06; //中断无效, 模计数(0x0000->T1CC0);

// T1CTL = 0x07; //中断无效, 上下计数(0x0000->T1CC0->0x0000);

// 无分频模式下的三种定时器运行状态设置

T1CTL = 0x01; //中断无效，自由运行(0x0000->0xffff);

// T1CTL = 0x02; //中断无效, 模计数(0x0000->T1CC0);

// T1CTL = 0x03; //中断无效, 上下计数(0x0000->T1CC0->0x0000);

// 模计数或上下计数模式运行时（涉及到T1CC0），需解除以下代码注释

// T1CCTL0 |= 0x04; // T1CCTL0.MODE = 1时，才能写T1CC0H

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* T1CC0寄存器由T1CC0L和T1CC0H两个寄存器构成，向T1CC0L当中写入内容时，

\* 内容先被缓存当T1CC0H中被写入内容时，才会将缓存的数据写入T1CC0L

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// T1CC0L = 0xFF;

// T1CC0H = 0xFF;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main()

{

//调用初始化函数

LED\_init();

timer1\_init();

while(1) //查询定时器1溢出

{

if(IRCON > 0)

{

IRCON = 0; //清溢出标志

counter++; //溢出次数加1

if(counter == 20)

{

LED2 = 0; //溢出20次后，点亮LED2

}

if(counter == 40)

{

LED2 = 1; //LED2点亮时长为20次定时器溢出

counter = 0; //清零

}

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530 定时器1中断

\* 程序描述：本程序用于演示与CC2530定时器1相关的操作，主函数开始对定时器1

\* 的溢出次数进行计数。当定时器1溢出20次后触发中断并使LED2点亮，

\* 持续20次定时器溢出所需时长后熄灭LED2，如此往复。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define uint unsigned int

#define uchar unsigned char

#define LED1 P1\_0

#define LED2 P1\_1

uint counter;//统计溢出次数

void LED\_init(void);

void timer1\_init(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_init(void)

{

//初始化P1

P1DIR = 0x03; //P1\_0 P1\_1为输出

LED1 = 1;

LED2 = 1; //熄灭LED

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：timer1\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化定时器1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void timer1\_init(void)

{

EA = 1; //开总中断，位寻址

T1OVFIM = 1; //T1OVFIM置1（或者T1CTL=0x08），这里使用了位寻址

T1IE = 1; //开定时器1中断，位寻址

T1CTL = 0x02;

// 128频模式下的三种定时器运行状态设置

// T1CTL = 0x0D; //中断无效，自由运行(0x0000->0xffff);

// T1CTL = 0x0E; //中断无效, 模计数(0x0000->T1CC0);

// T1CTL = 0x0F; //中断无效, 上下计数(0x0000->T1CC0->0x0000);

// 32分频模式下的三种定时器运行状态设置

// T1CTL = 0x09; //中断无效，自由运行(0x0000->0xffff);

// T1CTL = 0x0A; //中断无效, 模计数(0x0000->T1CC0);

// T1CTL = 0x0B; //中断无效, 上下计数(0x0000->T1CC0->0x0000);

// 8分频模式下的三种定时器运行状态设置

// T1CTL = 0x05; //中断无效，自由运行(0x0000->0xffff);

// T1CTL = 0x06; //中断无效, 模计数(0x0000->T1CC0);

// T1CTL = 0x07; //中断无效, 上下计数(0x0000->T1CC0->0x0000);

// 无分频模式下的三种定时器运行状态设置

// T1CTL = 0x01; //中断无效，自由运行(0x0000->0xffff);

// T1CTL = 0x02; //中断无效, 模计数(0x0000->T1CC0);

// T1CTL = 0x03; //中断无效, 上下计数(0x0000->T1CC0->0x0000);

// 当计数器运行在模计数或上下计数模式时（涉及到T1CC0），

// 需解除以下几行代码的注释

T1CCTL0 |= 0x04; // T1CCTL0.MODE = 1时，才能写T1CC0H

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* T1CC0寄存器由T1CC0L和T1CC0H两个寄存器构成，向T1CC0L当中写入内容时，

\* 内容先被缓存当T1CC0H中被写入内容时，才会将缓存的数据写入T1CC0L

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

T1CC0L = 0xFF;

T1CC0H = 0xFF;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main()

{

LED\_init();

timer1\_init(); //调用初始化函数

while(1) //查询溢出

{

//空循环，等待定时器溢出后执行中断服务程序

}

}

//定时器1中断服务程序（Interrupt Service Routine ISR）

#pragma vector = T1\_VECTOR

\_\_interrupt void T1\_ISR(void)

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 在中断服务程序中，不需要像上个实验程序文件中那样使用if语句判断IRCON

\* 是否为0，因为一旦中断服务程序被响应，IRCON被自动清零，

\* 所以使用if语句进行判断的话，将不会执行函数中的控制语句。

\* 换句话说，不能直接复制CC2530\_timer1.c文件中while循环体内的代码。

\* 要去掉其中的if判断语句。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

IRCON = 0; //清溢出标志

counter++; //溢出次数加1

if(counter == 20)

{

LED2 = 0; //溢出20次后，点亮LED2

}

if(counter == 40)

{

LED2 = 1; //LED2点亮时长为20次定时器溢出

counter = 0; //清零

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530 定时器1 PWM操作

\* 程序描述：本程序用于演示CC2530定时器1通道2的PWM操作，此时Timer1工作\* 在模计数模式下，运行后可观察到LED1的亮度逐渐发生变化

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define uint unsigned int

#define uchar unsigned char

#define LED1 P1\_0

#define LED2 P1\_1

void LED\_init(void);

void timer1\_init(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_init(void)

{

//初始化P1

P1DIR = 0x03; //P1\_0 P1\_1为输出

LED1 = 1;

LED2 = 1; //熄灭LED

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：timer1\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化定时器1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void timer1\_init(void)

{

CLKCONCMD &= ~0x40; //设置系统时钟源为32MHZ的晶振 P61

while(CLKCONSTA & 0x40); //等待晶振稳定为32MHZ P62

CLKCONCMD &= ~0x07; //设置系统主时钟频率为32MHZ

CLKCONCMD |= 0x38; //定时器标记输出为250KHZ

//定时器通道设置

P1SEL |= 0x01; //定时器1通道2映射至P1\_0，功能选择P78

PERCFG |= 0x40; //备用位置2，说明信息

P2SEL &= ~0x10; //相对于Timer4，定时器1优先P79

P2DIR |= 0xC0; //定时器通道2-3具有第一优先级P80

P1DIR |= 0x01;

//定时器模式设置

T1CTL = 0x02; //250KHZ不分频，模计数模式 P108

//T1CTL = 0x03; //250KHZ不分频，上下计数模式 P108

//根据表7-1（P73），此处P1\_0口必须装定时器1通道2进行比较

T1CCTL2 = 0x1C; //比较相等置1，计数器回0则清零 P110

//装定时器通道0初值（设置T1CC0）

T1CCTL0 |= 0x04; //设置定时器1通道0为比较模式，

//才能写T1CC0H和T1CC0L

T1CC0H = 0x00; // P29

T1CC0L = 0xFA; //PWM信号周期为1ms，频率为1KHZ P29

//时钟速度为250KHz，故T1CC0为0xFA (250)。

//装定时器通道2比较值

T1CC2H = 0x00; // P29

T1CC2L = 0xF7; //1%的正占空比

//T1CC2L = 0xE1; //10%的正占空比

//T1CC2L = 0xC8; //20%的正占空比

//T1CC2L = 0xAF; //30%的正占空比

//T1CC2L = 0x96; //40%的正占空比

//T1CC2L = 0x7D; //50%的正占空比

//T1CC2L = 0x64; //60%的正占空比

//T1CC2L = 0x4B; //70%的正占空比

//T1CC2L = 0x32; //80%的正占空比

//T1CC2L = 0x19; //90%的正占空比

//T1CC2L = 0x0A; //99%的正占空比

//T1CC2L = 0x01; //设置通道2比较寄存器初值

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitUs

\* 入口参数：usec

\* 参数类型：uint

\* 参数描述：延时的微秒数

\* 函数功能：用于实现以微秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitUs(uint usec)

{

while(usec--)

{

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitMs

\* 入口参数：msec

\* 参数类型：uint

\* 参数描述：延时的毫秒数

\* 函数功能：用于实现以毫秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitMs(uint msec)

{

while(msec--)

WaitUs(1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main()

{

uint i;

LED\_init();

timer1\_init();

while(1)

{

for(i=2;i<T1CC0L;i++) //利用PWM渐变调光，LED逐渐变亮

{

T1CC2H = 0;

T1CC2L = i;

WaitMs(5);

}

for(i=2;i<T1CC0L;i++) //利用PWM渐变调光，LED逐渐变暗

{

T1CC2H = 0;

T1CC2L = T1CC0L-i;

WaitMs(5);

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530 按键外部中断控制LED

\* 程序描述：本程序用于演示与CC2530外部中断相关的操作，按下位于试验箱左下角

\* 上的两个按键cancel和ok中的任意一个，即可点亮或熄灭LED灯。

\* 注 意：若程序无法烧录，请检查ZigBee模块1旁边的黄灯是否亮起，并确认模

\* 块下方的模块电源开关Z1\_POWER被打开（旁边的红色指示灯会亮起）

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define uint unsigned int

#define uchar unsigned char

//定义控制LED灯的端口

#define LED1 P1\_0 //定义LED1为P1.0口控制

#define LED2 P1\_1 //定义LED2为P1.0口控制

#define KEY1 P0\_4 //中断口

#define KEY2 P0\_7 //中断口

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitUs

\* 入口参数：usec

\* 参数类型：uint

\* 参数描述：延时的微秒数

\* 函数功能：用于实现以微秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitUs(uint usec)

{

while(usec--)

{

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitMs

\* 入口参数：msec

\* 参数类型：uint

\* 参数描述：延时的毫秒数

\* 函数功能：用于实现以毫秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitMs(uint msec)

{

while(msec--)

WaitUs(1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void led\_init(void)

{

P1DIR |= 0x03; //P1\_0定义为输出

LED1 = 1; //LED1灯熄灭

LED2 = 1; //LED1灯熄灭

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：interrupt\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：外部中断系统初始化

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void interrupt\_init()

{

P0IEN |= 0x90; //P0.1 设置为中断方式 P82

PICTL |= 0x02; // 下降沿触发 P81

IEN1 |= 0X20; // 允许P0口中断; P39

P0IFG = 0x00; // 初始化中断标志位 P81

EA = 1; //使能总中断 P39 IEN0

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：P0\_ISR

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：外部中断服务程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#pragma vector = P0INT\_VECTOR //格式：#pragma vector = 中断向量

\_\_interrupt void P0\_ISR(void)

{

WaitMs(5); //去除抖动

LED1=~LED1; //改变LED1状态

P0IFG = 0; //清引脚中断标志

P0IF = 0; //清端口0中断标志 IRCON.P0IF P42

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(void)

{

led\_init(); //初始化LED外设

interrupt\_init(); //初始化中断系统

while(1)

{

//空循环，等待中断发生

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530 中断优先级实验

\* 程序描述：本程序用于演示与CC2530中断优先级，本程序中Timer1中断被设为最

\* 低级，P0口外部中断被设为最高级。因此，当Timer1中断函数执行时

\* （LED1点亮），P0口的外部中断信号能打断Timer1对应的中断服务程序

\* 从而在Timer1中断服务程序执行期间，通过按下与P0口相连的外部按键

\* 改变LED1的工作状态。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define uint unsigned int

#define uchar unsigned char

#define LED1 P1\_0

#define LED2 P1\_1

uint counter;//统计溢出次数

void LED\_init(void);

void timer1\_init(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitUs

\* 入口参数：usec

\* 参数类型：uint

\* 参数描述：延时的微秒数

\* 函数功能：用于实现以微秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitUs(uint usec)

{

while(usec--)

{

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitMs

\* 入口参数：msec

\* 参数类型：uint

\* 参数描述：延时的毫秒数

\* 函数功能：用于实现以毫秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitMs(uint msec)

{

while(msec--)

WaitUs(1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_init(void)

{

P1DIR = 0x03; //P1\_0 P1\_1为输出

LED1 = 1;

LED2 = 1; //熄灭LED

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：timer1\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化定时器1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void timer1\_init(void)

{

//初始化T1定时器

EA = 1; //开总中断，位寻址

IP0 = 0x20;

IP1 = 0x20;

T1OVFIM = 1; //T1OVFIM置1（或者T1CTL=0x08），这里使用了位寻址

T1IE = 1; //开定时器1中断，位寻址

T1CTL = 0x01; //自由运行(0x0000->0xffff);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：P0\_ISR

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：外部中断服务程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void P0\_interrupt\_init(void)

{

IP0 = 0x20;

IP1 = 0x20;

P0IEN |= 0x90; //P0.1 设置为中断方式 P82

PICTL |= 0x02; // 下降沿触发 P81

IEN1 |= 0X20; // 允许P0口中断; P39

P0IFG = 0x00; // 初始化中断标志位 P81

EA = 1; //使能总中断 P39 IEN0

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main()

{

LED\_init();

timer1\_init(); //调用初始化函数

P0\_interrupt\_init();

while(1) //查询溢出

{

//空循环，等待定时器溢出后执行中断服务程序

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：T1\_ISR

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：定时器1外部中断服务程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#pragma vector = T1\_VECTOR

\_\_interrupt void T1\_ISR(void)

{

IRCON = 0; //清溢出标志

LED2 = 0; //溢出20次后，点亮LED2

WaitMs(3000);

LED2 = 1;

WaitMs(1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：P0\_ISR

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：外部中断服务程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#pragma vector = P0INT\_VECTOR

\_\_interrupt void P0\_ISR(void)

{

WaitMs(5); //去除抖动

LED2=~LED2; //改变LED2状态

P0IFG = 0; //清引脚中断标志

P0IF = 0; //清端口0中断标志 IRCON.P0IF P42

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530 定时器2操作实验

\* 程序描述：本程序用于演示与CC2530定时器2相关的操作，主函数开始对定时器1

\* 的溢出次数进行计数。当定时器2溢出20次后触发中断并使LED2点亮，

\* 持续20次定时器溢出所需时长后熄灭LED2，如此往复

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define uint unsigned int

#define uchar unsigned char

#define LED1 P1\_0

#define LED2 P1\_1

uint counter = 0; //定时器溢出次数

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_init(void)

{

//初始化P1

P1DIR = 0x03; //P1\_0 P1\_1为输出 P79

LED1 = 1;

LED2 = 1; //熄灭LED

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：clear\_timer2\_interrupt

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于清除定时器2中断标志位

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void clear\_timer2\_interrupt(void)

{

//T2CTRL &= ~0XC0;

T2IF = 0; //清除IRCON中的T2IF位（T2IF位仅指示Timer2有中断）

T2IRQF = 0x00; //清除定时器2中断标志寄存器（T2IRQF具体指明了中断源）

T2CTRL|= 0x01; // 启动定时器2

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：timer2\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化定时器2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void timer2\_init(void)

{

EA = 1; //开总中断

T2IE = 1; //IEN1.T2IE

T2IRQM |= 1; //设置中断屏蔽位（置1表示不屏蔽中断请求）

//设置Timer2计数值

T2M0 = 0x55;

T2M1 = 0x55;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(void)

{

LED\_init();

timer2\_init();

T2CTRL|=0x01; //启动Timer2

while(1)

{

//等待Timer2定时器中断

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：T2\_ISR

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：定时器2中断服务程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#pragma vector = T2\_VECTOR

\_\_interrupt void T2\_ISR()

{

T2CTRL &= 0xFE; //停止Timer2

//设置Timer2计数值

T2M0 = 0x55;

T2M1 = 0x55;

clear\_timer2\_interrupt();

if(counter<100)

{counter++;}

else

{counter = 0;

LED2 = !LED2;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530定时器3操作

\* 程序描述：本程序用于演示CC2530定时器3的相关操作，包括运行模式设置、预分

\* 频设置、中断程序编写。当定时器3出现中断后，将会产生中断并使LED

\* 灯闪烁。

\* 注 意：由于定时器3为一个8位定时器，因此溢出频率较高，当预分频值过小时

\* 会使LED灯的闪烁频率很高，几乎为常量，并非程序出现问题。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define LED1 P1\_0

#define LED2 P1\_1

#define uchar unsigned char

#define uint unsigned int

//定义定时器3运行模式宏定义

#define T3\_MODE\_FREE 0

#define T3\_MODE\_DOWN 1

#define T3\_MODE\_MODULO 2

#define T3\_MODE\_UP\_DOWN 3

//定义中断使能宏定义

#define T3\_INTERRUPT\_ENABLE 1

#define T3\_INTERRUPT\_DISABLE 0

//定义定时器3时钟预分频宏定义

#define DIV\_1 1

#define DIV\_2 2

#define DIV\_4 4

#define DIV\_8 8

#define DIV\_16 16

#define DIV\_32 32

#define DIV\_64 64

#define DIV\_128 128

//定义定时器溢出次数计数器全局变量

uchar counter = 0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：set\_timer3\_clock\_prescale

\* 入口参数：div

\* 参数类型：uchar

\* 参数描述：用于设置定时器3的时钟源分频值，取值范围见宏定义

\* 函数功能：用于设置定时器3的时钟源分频值

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void set\_timer3\_clock\_prescale(uchar div)

{ T3CTL &= ~0XE0; // 清零T3CTL[7:5]位 P116

if (div == 2) // 不同的预分频值对应的寄存器设置

T3CTL|=0X20;

if (div == 4)

T3CTL|=0X40;

if (div == 8)

T3CTL|=0X60;

if (div == 16)

T3CTL|=0X80;

if (div == 32)

T3CTL|=0XA0;

if (div == 64)

T3CTL|=0XC0;

if (div == 128)

T3CTL|=0XE0;

else

T3CTL|=0X00;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：set\_timer3\_mode

\* 入口参数：running\_mode, reload

\* 参数类型：uchar

\* 参数描述：running\_mode：用于设置定时器3的运行模式

\* reload：计数器自动重装值

\* 函数功能：用于设置定时器3的工作模式

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void set\_timer3\_mode(uchar running\_mode, uchar reload)

{

T3CTL &= 0xFC; // 清零T3CTL[0:1]位 P116

if(running\_mode == T3\_MODE\_FREE)

{ T3CTL|=0X00;}

if(running\_mode == T3\_MODE\_DOWN)

{ T3CTL|=0X01;

T3CCTL0 = 0x04;

T3CC0 = reload;

}

if(running\_mode == T3\_MODE\_MODULO)

{ T3CTL|=0X02;

T3CCTL0 = 0x04;

T3CC0 = reload;

}

if(running\_mode == T3\_MODE\_UP\_DOWN)

{ T3CTL|=0X03;

T3CCTL0 = 0x04;

T3CC0 = reload;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：timer3\_init

\* 入口参数：running\_mode, prescale, interrupt\_enable, reload

\* 参数类型：uchar

\* 参数描述：running\_mode - 定时器3运行模式

\* prescale - 时钟预分频值

\* interrupt\_enable - 是否启用中断

\* reload - 自动重装值，自由运行模式除外

\* 函数功能：定时器3初始化

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void timer3\_init(uchar running\_mode,uchar prescale,

uchar interrupt\_enable,uchar reload)

{

if(interrupt\_enable == 1)

{

EA = 1; //开总中断

T3IE = 1; // 运行T3向CPU发出中断请求 P30 IEN1.T3IE

T3CTL |= 0x08; // 使能T3中断系统

}

if(interrupt\_enable == 0)

{

EA = 0; //关闭总中断

}

//设置定时器3预分频值

set\_timer3\_clock\_prescale(prescale);

set\_timer3\_mode(running\_mode,reload);

T3CTL |= 0x10; //启动定时器3

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_init(void)

{

P1DIR = 0X03;

LED1 = 1;

LED2 = 1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(void)

{

LED\_init();

timer3\_init(T3\_MODE\_FREE,DIV\_1,T3\_INTERRUPT\_ENABLE,0);

while(1); //等待中断

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：T3\_ISR

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：定时器3中断服务程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#pragma vector = T3\_VECTOR

\_\_interrupt void T3\_ISR(void)

{

if(counter<200)

counter++; //200次中断LED闪烁一轮

else

{

counter = 0; //计数清零

LED1 = !LED1; //改变小灯的状态

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530定时器4操作

\* 程序描述：本程序用于演示CC2530定时器4的相关操作，包括运行模式设置、预分

\* 频设置、中断程序编写。当定时器4出现中断后，将会产生中断并使LED

\* 灯闪烁。

\* 注 意：由于定时器4为一个8位定时器，因此溢出频率较高，当预分频值过小时

\* 会使LED灯的闪烁频率很高，几乎为常量，并非程序出现问题。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define LED1 P1\_0

#define LED2 P1\_1

#define uchar unsigned char

#define uint unsigned int

//定义定时器4运行模式宏定义

#define T4\_MODE\_FREE 0

#define T4\_MODE\_DOWN 1

#define T4\_MODE\_MODULO 2

#define T4\_MODE\_UP\_DOWN 3

//定义中断使能宏定义

#define T4\_INTERRUPT\_ENABLE 1

#define T4\_INTERRUPT\_DISABLE 0

//定义定时器4时钟预分频宏定义

#define DIV\_1 1

#define DIV\_2 2

#define DIV\_4 4

#define DIV\_8 8

#define DIV\_16 16

#define DIV\_32 32

#define DIV\_64 64

#define DIV\_128 128

//定义定时器溢出次数计数器全局变量

uchar counter = 0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：set\_timer4\_clock\_prescale

\* 入口参数：div

\* 参数类型：uchar

\* 参数描述：用于设置定时器4的时钟源分频值，取值范围见宏定义

\* 函数功能：用于设置定时器4的时钟源分频值

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void set\_timer4\_clock\_prescale(uchar div)

{ T4CTL &= ~0XE0; // 清零T4CTL[7:5]位 P116

if (div == 2) // 不同的预分频值对应的寄存器设置

T4CTL|=0X20;

if (div == 4)

T4CTL|=0X40;

if (div == 8)

T4CTL|=0X60;

if (div == 16)

T4CTL|=0X80;

if (div == 32)

T4CTL|=0XA0;

if (div == 64)

T4CTL|=0XC0;

if (div == 128)

T4CTL|=0XE0;

else

T4CTL|=0X00;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：set\_timer4\_mode

\* 入口参数：running\_mode, reload

\* 参数类型：uchar

\* 参数描述：running\_mode：用于设置定时器4的运行模式

\* reload：计数器自动重装值

\* 函数功能：用于设置定时器4的工作模式

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void set\_timer4\_mode(uchar running\_mode, uchar reload)

{

T4CTL &= 0xFC; // 清零T4CTL[0:1]位 P116

if(running\_mode == T4\_MODE\_FREE)

{ T4CTL|=0X00;}

if(running\_mode == T4\_MODE\_DOWN)

{ T4CTL|=0X01;

T4CCTL0 = 0x04;

T4CC0 = reload;

}

if(running\_mode == T4\_MODE\_MODULO)

{ T4CTL|=0X02;

T4CCTL0 = 0x04;

T4CC0 = reload;

}

if(running\_mode == T4\_MODE\_UP\_DOWN)

{ T4CTL|=0X03;

T4CCTL0 = 0x04;

T4CC0 = reload;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：timer4\_init

\* 入口参数：running\_mode, prescale, interrupt\_enable, reload

\* 参数类型：uchar

\* 参数描述：running\_mode - 定时器4运行模式

\* prescale - 时钟预分频值

\* interrupt\_enable - 是否启用中断

\* reload - 自动重装值，自由运行模式除外

\* 函数功能：定时器4初始化

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void timer4\_init(uchar running\_mode,uchar prescale,

uchar interrupt\_enable,uchar reload)

{

if(interrupt\_enable == 1)

{

EA = 1; //开总中断

T4IE = 1; // 运行T4向CPU发出中断请求 P30 IEN1.T4IE

T4CTL |= 0x08; // 使能T4中断系统

}

if(interrupt\_enable == 0)

{

EA = 0; //关闭总中断

}

//设置定时器4预分频值

set\_timer4\_clock\_prescale(prescale);

set\_timer4\_mode(running\_mode,reload);

T4CTL |= 0x10; //启动定时器4

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_init(void)

{

P1DIR = 0X03;

LED1 = 1;

LED2 = 1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(void)

{

LED\_init();

Timer4\_init(T4\_MODE\_FREE,DIV\_1,T4\_INTERRUPT\_ENABLE,0);

while(1); //等待中断

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：T4\_ISR

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：定时器4中断服务程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#pragma vector = T4\_VECTOR

\_\_interrupt void T4\_ISR(void)

{

if(counter<200)

counter++; //200次中断LED闪烁一轮

else

{

counter = 0; //计数清零

LED1 = !LED1; //改变小灯的状态

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530 定时器4 PWM操作

\* 程序描述：本程序用于演示定时器4的PWM功能，通过PWM功能调节LED亮度。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define LED1 P1\_0

#define LED2 P1\_1

#define uchar unsigned char

#define uint unsigned int

//定义定时器4运行模式宏定义

#define T4\_MODE\_FREE 0

#define T4\_MODE\_DOWN 1

#define T4\_MODE\_MODULO 2

#define T4\_MODE\_UP\_DOWN 3

//定义中断使能宏定义

#define T4\_INTERRUPT\_ENABLE 1

#define T4\_INTERRUPT\_DISABLE 0

//定义定时器4时钟预分频宏定义

#define DIV\_1 1

#define DIV\_2 2

#define DIV\_4 4

#define DIV\_8 8

#define DIV\_16 16

#define DIV\_32 32

#define DIV\_64 64

#define DIV\_128 128

//定义定时器溢出次数计数器全局变量

uchar counter = 0;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：set\_timer4\_clock\_prescale

\* 入口参数：div

\* 参数类型：uchar

\* 参数描述：用于设置定时器4的时钟源分频值，取值范围见宏定义

\* 函数功能：用于设置定时器4的时钟源分频值

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void set\_timer4\_clock\_prescale(uchar div)

{

T4CTL &= ~0XE0; // 清零T4CTL[7:5]位 P118

if (div == 2) // 不同的预分频值对应的寄存器设置

T4CTL|=0X20;

if (div == 4)

T4CTL|=0X40;

if (div == 8)

T4CTL|=0X60;

if (div == 16)

T4CTL|=0X80;

if (div == 32)

T4CTL|=0XA0;

if (div == 64)

T4CTL|=0XC0;

if (div == 128)

T4CTL|=0XE0;

else

T4CTL|=0X00;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：set\_timer4\_mode

\* 入口参数：running\_mode, reload

\* 参数类型：uchar

\* 参数描述：running\_mode：用于设置定时器4的运行模式

\* reload：计数器自动重装值

\* 函数功能：用于设置定时器4的工作模式

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void set\_timer4\_mode(uchar running\_mode, uchar reload)

{

T4CTL &= 0xFC; // 清零T4CTL[0:1]位 P118

if(running\_mode == T4\_MODE\_FREE)

{ T4CTL|=0X00;}

if(running\_mode == T4\_MODE\_DOWN)

{

T4CTL|=0X01;

T4CCTL0 |= 0x04; //设置定时器4为比较模式 T4CCTL0 P118

T4CC0 = reload;

}

if(running\_mode == T4\_MODE\_MODULO)

{

T4CTL|=0X02;

T4CCTL0 |= 0x04; //设置定时器4为比较模式 T4CCTL0 P118

T4CC0 = reload;

}

if(running\_mode == T4\_MODE\_UP\_DOWN)

{

T4CTL|=0X03;

T4CCTL0 |= 0x04; //设置定时器4为比较模式 T4CCTL0 P118

T4CC0 = reload;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：timer4\_init

\* 入口参数：running\_mode, prescale, interrupt\_enable, reload

\* 参数类型：uchar

\* 参数描述：running\_mode - 定时器4运行模式

\* prescale - 时钟预分频值

\* interrupt\_enable - 是否启用中断

\* reload - 自动重装值，自由运行模式除外

\* 函数功能：定时器4初始化

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void timer4\_init(uchar running\_mode,uchar prescale,

uchar interrupt\_enable,uchar reload)

{

CLKCONCMD &= ~0x40; //设置系统时钟源为32MHZ的晶振 P61

while(CLKCONSTA & 0x40); //等待晶振稳定为32MHZ P62

CLKCONCMD &= ~0x07; //设置系统主时钟频率为32MHZ

CLKCONCMD |= 0x38; //定时器标记输出为250KHZ

if(interrupt\_enable == 1)

{

EA = 1; //开总中断

T4IE = 1; // 运行T4向CPU发出中断请求 P30 IEN1.T4IE

T4CTL |= 0x08; // 使能T4中断系统

}

if(interrupt\_enable == 0)

{

EA = 0; //关闭总中断

}

//设置定时器4预分频值

set\_timer4\_clock\_prescale(prescale);

set\_timer4\_mode(running\_mode,reload);

T4CCTL1 &= 0x40;

T4CCTL1 |= 0x40;

T4CTL |= 0x10; //启动定时器4

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_init(void)

{

P1DIR = 0X03;

LED1 = 1;

LED2 = 1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitUs

\* 入口参数：usec

\* 参数类型：uint

\* 参数描述：延时的微秒数

\* 函数功能：用于实现以微秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitUs(uint usec)

{

while(usec--)

{

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitMs

\* 入口参数：msec

\* 参数类型：uint

\* 参数描述：延时的毫秒数

\* 函数功能：用于实现以毫秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitMs(uint msec)

{

while(msec--)

WaitUs(1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(void)

{

uchar i = 0;

LED\_init();

timer4\_init(T4\_MODE\_MODULO,DIV\_1,T4\_INTERRUPT\_ENABLE,0xFA);

while(1)

{

for(i=2;i<T4CC0;i++) //利用PWM渐变调光，LED逐渐变亮

{

T4CC1 = i;

WaitMs(5);

}

for(i=2;i<T4CC0;i++) //利用PWM渐变调光，LED逐渐变暗

{

T4CC1 = T4CC0 - i;

WaitMs(5);

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：T4\_ISR

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：定时器4中断服务程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#pragma vector = T4\_VECTOR

\_\_interrupt void T4\_ISR(void)

{

LED1 = !LED1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530 睡眠唤醒（定时器中断方式）

\* 程序描述：本程序用于演示CC2530低功耗睡眠唤醒功能，运行程序后LED首先闪烁\* 3次，然后系统进入低功耗模式PM2。该模式下睡眠定时器开始工作，使用\* 32KHz晶振作为时钟源，当睡眠定时器溢出时唤醒系统。在系统唤醒期间， \* LED会闪烁3次作为指示。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#define uchar unsigned char

#define uint unsigned int

#define ulong unsigned long

#define LED1 P1\_0 //P1.0口控制LED1

#define LED2 P1\_1 //P1.1口控制LED2

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitUs

\* 入口参数：usec

\* 参数类型：uint

\* 参数描述：延时的微秒数

\* 函数功能：用于实现以微秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitUs(uint usec)

{

while(usec--)

{

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitMs

\* 入口参数：msec

\* 参数类型：uint

\* 参数描述：延时的毫秒数

\* 函数功能：用于实现以毫秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitMs(uint msec)

{

while(msec--)

WaitUs(1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_init(void)

{

//初始化P1

P1DIR = 0x03; //P1\_0 P1\_1为输出

LED1 = 1;

LED2 = 1; //熄灭LED

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：SysPowerMode

\* 入口参数：mode

\* 参数类型：uchar

\* 参数描述：mode=0:PM0 mode=1:PM1 mode=2:PM2 mode=3:PM3

\* 函数功能：用于设置CC2530芯片的功耗模式

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void SysPowerMode(uchar mode)

{

if(mode > 0 && mode < 4)

{

SLEEPCMD |= mode; //设置系统睡眠模式P56

PCON = 0x01; //进入睡眠模式 ,通过中断唤醒P56

}

else

PCON = 0x00; //通过中断唤醒系统

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：ST\_ISR

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：睡眠定时器中断服务程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#pragma vector = ST\_VECTOR

\_\_interrupt void ST\_ISR(void)

{

STIF = 0; //清除睡眠定时器中断标志位 IRCON.STIF P42

SysPowerMode(0); //进入正常工作模式

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：InitSleepTimer

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化睡眠定时器

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void InitSleepTimer(void)

{

ST2 = 0x00;

ST1 = 0x00;

ST0 = 0x00;

EA = 1; //开总中断

STIE = 1; //睡眠定时器中断使能 0：禁止 1：使能 IEN0.STIE P29

STIF = 0; //睡眠定时器中断标志 0：无中断未决 1：中断未决 IRCON.STIF

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：Set\_ST\_Period

\* 入口参数：sec

\* 参数类型：uchar

\* 参数描述：睡眠时间长度

\* 函数功能：用于实现以秒为单位的睡眠长度

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Set\_ST\_Period(uint sec)

{

ulong sleepTimer = 0;

// 读取睡眠定时器比较值寄存器时，需要先从ST0开始读取

sleepTimer |= ST0;

sleepTimer |= (ulong)ST1 << 8;

sleepTimer |= (ulong)ST2 << 16;

//睡眠定时器使用的32KHz晶振实际上为32.768KHz,

//因此每计数32768个时钟脉冲即为1秒

sleepTimer += ((ulong)sec \* (ulong)32768);

ST2 = (uchar)(sleepTimer >> 16);

ST1 = (uchar)(sleepTimer >> 8);

ST0 = (uchar) sleepTimer;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(void)

{

uchar i=0;

LED\_init(); //设置LED灯相应的IO口

InitSleepTimer(); //初始化休眠定时器

while(1)

{

for (i=0; i<6; i++) //LED1闪烁3次提醒用户将进入睡眠模式

{

LED1 = ~LED1;

WaitMs(200);

}

Set\_ST\_Period(3); //设置睡眠时间,睡眠3秒后唤醒系统

SysPowerMode(2); //重新进入睡眠模式PM2

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530睡眠唤醒（外部中断模式）

\* 程序描述：本程序用于演示CC2530的低功耗模式功能，运行程序后LED灯首先闪烁\* 3次，然后系统进入低功耗模式PM3。该模式下所有晶振停振，因此睡眠定 \* 时器停止工作，只能通过外部中断唤醒系统，通过按下试验箱上的OK或

\* CANCEL键用来唤醒系统。

\* 注 意：请将该程序烧入左下角的ZigBee模块(Z1)，因为只有改模块配备了按键。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

typedef unsigned char uchar;

typedef unsigned int uint;

#define LED1 P1\_0 //P1.0口控制LED1

#define LED2 P1\_1 //P1.1口控制LED1

#define KEY1 P0\_1 //P0.1口控制S1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_init(void)

{

//初始化P1

P1DIR = 0x03; //P1\_0 P1\_1为输出

LED1 = 1;

LED2 = 1; //熄灭LED

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitUs

\* 入口参数：usec

\* 参数类型：uint

\* 参数描述：延时的微秒数

\* 函数功能：用于实现以微秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitUs(uint usec)

{

while(usec--)

{ asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitMs

\* 入口参数：msec

\* 参数类型：uint

\* 参数描述：延时的毫秒数

\* 函数功能：用于实现以毫秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitMs(uint msec)

{

while(msec--)

WaitUs(1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：interrupt\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：初始化外部中断系统

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void interrupt\_init()

{

P0IEN |= 0x90; //P0.1 设置为中断方式 P82

PICTL |= 0x02; // 下降沿触发 P81

IEN1 |= 0X20; // 允许P0口中断; P39

P0IFG = 0x00; // 初始化中断标志位 P81

EA = 1; //使能总中断 P39 IEN0

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：SysPowerMode

\* 入口参数：mode

\* 参数类型：uchar

\* 参数描述：mode=0:PM0 mode=1:PM1 mode=2:PM2 mode=3:PM3

\* 函数功能：用于设置CC2530芯片的功耗模式

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void SysPowerMode(uchar mode)

{

if(mode > 0 && mode < 4)

{ SLEEPCMD |= mode; //设置系统睡眠模式 P56

PCON = 0x01; //进入睡眠模式 ,通过中断唤醒 P56

}

else

PCON = 0x00; //主动/空闲模 通过中断唤醒系统

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(void)

{

uchar i=0;

LED\_init(); //设置LED灯相应的IO口

interrupt\_init(); //设置KEY相应的IO口

while(1)

{ for (i=0; i<6; i++) //LED1闪烁3次提醒用户将进入睡眠模式

{ LED1 = ~LED1;

WaitMs(200);

}

SysPowerMode(3); //进入睡眠模式PM3,按下按键S1中断唤醒系统

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：P0\_ISR

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：外部中断服务程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#pragma vector = P0INT\_VECTOR

\_\_interrupt void P0\_ISR(void)

{

if(P0IFG > 0)

{ P0IFG = 0; //清标志位}

P0IF = 0;

SysPowerMode(0); //正常工作模式

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 实验名称：CC2530串口传输数据实验

\* 程序描述：本程序通过CC2530的UART0向PC机传输数据，烧入程序后在PC机上

\* 运行串口调试助手，就会看到调试界面中不断收到由CC2530芯片发送的

\* 内容为”ShengDa College.”字样的信息

\* 注 意：波特率为119200

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ioCC2530.h>

#include <string.h>

#define uint unsigned int

#define uchar unsigned char

//定义控制灯的端口

#define LED1 P1\_0

#define LED2 P1\_1

//函数声明

void Delay(uint);

void initUARTSEND(void);

void UartTX\_Send\_String(char \*Data,int len);

char \*Txdata ="ShengDa College.\r\n";

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitUs

\* 入口参数：usec

\* 参数类型：uint

\* 参数描述：延时的微秒数

\* 函数功能：用于实现以微秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitUs(uint usec)

{

while(usec--)

{

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

asm("NOP");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：WaitMs

\* 入口参数：msec

\* 参数类型：uint

\* 参数描述：延时的毫秒数

\* 函数功能：用于实现以毫秒为单位的某一时间长度延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void WaitMs(uint msec)

{

while(msec--)

WaitUs(1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：LED\_init

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：用于初始化LED外设

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LED\_init(void)

{ P1DIR = 0x03; //P1控制LED

LED1 = 1;

LED2 = 1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：initUARTSEND

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：串口初始化函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/void initUARTSEND(void)

{ CLKCONCMD &= ~0x40; //设置系统时钟源为32MHZ晶振

while(CLKCONSTA & 0x40);//等待晶振稳定

CLKCONCMD &= ~0x47; //设置系统主时钟频率为32MHZ

PERCFG = 0x00; //UART0使用可选位置1，P0口（见P73页拓扑表）

P0SEL = 0x3c; //P0\_2,P0\_3,P0\_4,P0\_5用作串口，对应位设为1

P2DIR &= ~0XC0; //P0优先作为UART0，最高两位清零

U0CSR |= 0x80; //UART方式，最高位置1

U0GCR |= 9; //设置BAUD\_E，见P151 Table 16-1波特率设置

U0BAUD |= 59; //设置BAUD\_M，P150式16-1 给出了波特率的计算公式

UTX0IF = 0; //UART0 TX中断标志初始置位0

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：UartTX\_Send\_String

\* 入口参数：Data, len

\* 参数类型：char, int

\* 参数描述：\*Data:指向要发送数据的数据指针 len:将要发送的数据长度

\* 函数功能：串口发送函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void UartTX\_Send\_String(char \*Data,int len)

{

int j;

for(j=0;j<len;j++)

{

U0DBUF = \*Data++; //UART0接收、发送数据缓存

//等待发送完成后，清除UART0发送中断标志位

while(UTX0IF == 0);

UTX0IF = 0;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 函数名称：main

\* 入口参数：无

\* 参数类型：无

\* 参数描述：无

\* 函数功能：主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/void main(void)

{

LED\_init();

initUARTSEND();

UartTX\_Send\_String(Txdata,strlen(Txdata)); //发送数据

while(1)

{

UartTX\_Send\_String(Txdata,strlen(Txdata)); //发送数据

LED2 =!LED2; //每发送完毕一次，LED闪烁一次

WaitMs(500);

}

}