/\* 包含头文件 \*/

#include <ioCC2530.h>

#include "hal\_defs.h"

/\*宏定义\*/

#define LED1 P1\_0

#define LED2 P1\_1

#define SW1 P1\_2

/\*定义变量\*/

uint8 counter = 0; //统计定时器溢出次数

uint8 buff[2];

uint8 flag = 0;

uint16 value = 0;

/\*声明函数\*/

void Delay1Ms(uint8 time);//延时函数，32MHz系统时钟下，约1ms延时函数

void InitCLK(void);//系统时钟初始化函数，为32MHz

void InitLED(void);//LED灯IO端口初始化函数

void InitTime1(void);//定时器1初始化函数，数据溢出周期为50ms

void InitADC(void);//ADC模块初始化

uint16 get\_adc(void);//获取ADC转换结果，返回分辨率为0.01V的电压值

void InitUart0(void);//串口0初始化函数

void UART0SendByte(unsigned char c);//UART0发送一个字节函数

void UART0SendString(unsigned char \*str);//UART0发送整个字符串

void UART0SendData(unsigned char \*str,int len);//UART0发送指定数量字节数据

/\*定义函数\*/

void Delay1Ms(uint8 time)

{

unsigned int i,j;

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

{

for(j=0;j<1100;j++);

}

}

void InitCLK(void)

{

CLKCONCMD &= 0x80;

while(CLKCONSTA & 0x40);

}

void InitLED(void)

{

P1SEL &= ~0x07;//设置P1\_0、P1\_1、P1\_2为GPIO口

P1DIR |= 0x03;//设置P1\_0和P1\_1为输出

}

void InitTime1(void)

{

T1CC0L = 50000 & 0xff;

T1CC0H = (50000 &0xff00)>>8;

T1CCTL0 |= 0x04;//设定定时器1通道0比较模式

T1CTL = 0x0a;//设置定时器1为32分频、模模式，并开始运行

TIMIF &= ~0x40;//不产生定时器1的溢出中断

T1IE = 1;//使能定时器1中断

}

void InitADC(void)

{

APCFG |=0x01;

P0SEL |=0x01;

P0DIR &=~0x01;

}

uint16 get\_adc(void)

{

uint32 value;

ADCIF = 0; //清ADC 中断标志

//采用基准电压avdd5:3.3V，通道0，启动AD转化

ADCCON3 = (0x80 | 0x10 | 0x00);

while (!ADCIF); //等待AD转化结束

value = ADCH;

value = value<< 8;

value |= ADCL;

// AD值转化成电压值

// 0 表示 0V ，32768 表示 3.3V

// 电压值 = (value\*3.3)/32768 （V)

value = (value \* 330);

value = value >> 15; // 除以32768

// 返回分辨率为0.01V的电压值

return (uint16)value;

}

void InitUart0(void)

{

PERCFG = 0x00;

P0SEL = 0x3c;

U0CSR |= 0x80;

U0BAUD = 216;

U0GCR = 11;

U0UCR |= 0x80;

UTX0IF = 0; // 清零UART0 TX中断标志

}

void UART0SendByte(unsigned char c)

{

U0DBUF = c;// 将要发送的1字节数据写入U0DBUF

while (!UTX0IF) ;// 等待TX中断标志，即U0DBUF就绪

UTX0IF = 0;// 清零TX中断标志

}

void UART0SendString(unsigned char \*str)

{

while(\*str != '\0')

{

UART0SendByte(\*str++);

}

}

void UART0SendData(unsigned char \*str,int len)

{

for(int i=0;i<len;i++)

{

U0DBUF = str[i]; // 将要发送的1字节数据写入U0DBUF

while (!UTX0IF) ; // 等待TX中断标志，即U0DBUF就绪

UTX0IF = 0; // 清零TX中断标志UART0SendByte(\*str++); // 发送一字节

}

}

/\*主函数\*/

void main(void)

{

InitCLK();

InitLED();

InitTime1();

InitADC();

InitUart0();

/\*.......答题区1开始：LED灯初始状态设置...........\*/

LED1 = LED2 = 0;//设置LED1和LED2的初始状态

/\*.......答题区1结束...........\*/

/\*.......答题区2开始：SW1按键中断输入功能初始化...........\*/

P1DIR &= ~0x04;//设置P1\_2为输入

P1INP &= ~0x04;//设置P1\_2端口为“上拉/下拉”模式

P2INP &= ~0x40;//设置所有P1端口为“上拉”

PICTL |= 0x02;//设置P1\_2端口中断触发方式为：下降沿触发

IEN2 |= 0x10;//使能P1端口中断

P1IEN |= 0x04;//使能P1\_2端口中断

/\*.......答题区2结束...........\*/

EA = 1;//使能总中断

while(1)

{

/\*..答题区3开始：每隔2s采集光照度数据并通过串口发送，每次采集LED2闪烁..\*/

if(counter >= 40)

{

counter = 0;

if(flag == 1)

{

value = get\_adc();

buff[0] = value>>8;

buff[1] = value;

UART0SendData(buff,2);

LED2 = 1;

Delay1Ms(200);

LED2 = 0;

}

}

/\*.......答题区3结束...........\*/

}

}

/\*中断服务函数\*/

/\*.......答题区4开始：按键中断服务函数...........\*/

#pragma vector = P1INT\_VECTOR

\_\_interrupt void P1\_ISR(void)

{

if(P1IF == 1)

{

if(P1IFG & 0x04)

{

if(flag == 0)

{

flag = 1;

LED1 = 1;

}

else

{

flag = 0;

LED1 = 0;

}

P1IFG &= ~0x04;

}

P1IF = 0;

}

}

/\*.......答题区4结束...........\*/

#pragma vector = T1\_VECTOR

\_\_interrupt void T1\_ISR(void)

{

counter++;

T1STAT &= ~0x01; //清除通道0中断标志

}