#include <msp430f249.h>

#define AHout P5OUT = 0x01;

#define ALout P5OUT = 0x02;

#define BHout P5OUT = 0x04;

#define BLout P5OUT = 0x08;

#define EndOut P5OUT = 0x00;

#define LightOut P5OUT |= 0x10

#define Hred BIT0

#define Hyellow BIT1

#define Hgreen BIT2

#define Vred BIT3

#define Vyellow BIT4

#define Vgreen BIT5

#define Vertical 0

#define Horizontal 1

int num[17]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71,0x40};

int timeA,timeB; //输出时间

int red,yellow,green; //红绿黄等定时选择

int dir; //方向选择

void SetLights(int light) //设置灯亮灭

{

P6OUT = light;

LightOut; //输出信号

EndOut; //输出结束

}

void Init()

{

/\*设置输出端口\*/

P4SEL &= 0x00;

P4DIR |= 0xff;

P5SEL &= 0x00;

P5DIR |= 0xff;

P6SEL &= 0x00;

P6DIR |= 0xff;

P1SEL &= ~0x03;

P1DIR |= 0x03;

P1OUT = 0x01;

P4OUT = 0xff;

P5OUT = 0xff;

P6OUT = 0xff;

/\*定时器设置\*/

TACTL = TASSEL\_1 + MC\_1; //时钟选择 ACLK = 32768 Hz

CCR0 = 32768; //定时 1 s

CCTL0 = CCIE; //时钟中断打开

TBCTL = TBSSEL\_1 + MC\_1;

TBCCR0 = 500;

TBCCTL0=CCIE;

/\*\*/

dir=Vertical;

SetLights(Vred + Hgreen);

}

void ReSet() //定时重置

{

red=30;

yellow=5;

green=25;

}

void SetData() //输出写入

{

if(dir == Horizontal) //horizontal-red

{

timeA=red;

if(green > 0)

{

timeB = green;

SetLights(Hred + Vgreen);

}

else

{

timeB = yellow;

SetLights(Hred + Vyellow);

}

}

else //vertical-red

{

timeB=red;

if(green > 0)

{

timeA = green;

SetLights(Vred + Hgreen);

}

else

{

timeA = yellow;

SetLights(Vred + Hyellow);

}

}

}

void WriteA(int high,int low)

{

P4OUT = high; //十位

AHout;

EndOut;

P4OUT = low; //个位

ALout;

EndOut;

}

void WriteB(int high,int low)

{

P4OUT = high; //十位

BHout;

EndOut;

P4OUT = low; //个位

BLout;

EndOut;

}

//Timer\_A定时器中断

#pragma vector=TIMERA0\_VECTOR

\_\_interrupt void Timer\_A()

{

SetData();

WriteA(num[timeA/10],num[timeA%10]);

WriteB(num[timeB/10],num[timeB%10]);

red--;

green--;

if(green < 0)

{

yellow--;

if(yellow <= 0)

{

if(dir == Horizontal) dir = Vertical;

else dir = Horizontal;

ReSet();

}

}

}

//LED驱动信号

#pragma vector=TIMERB0\_VECTOR

\_\_interrupt void Timer\_B()

{

P1OUT = ~P1OUT;

}

void main()

{

WDTCTL = WDTPW + WDTHOLD; //关闭看门狗

Init(); //初始化设置

ReSet(); //输出重置

\_BIS\_SR(LPM3\_bits + GIE); //打开全局中断，并进入低功耗模式

}