

یا لطیف



دانشگاه صنعتی شاهرود

دانشکده مهندسی برق

تمرین های شبیه سازی ریز پردازنده

تمرین سری ۴

تهیه کننده و نویسنده:

رضا آدینه پور

استاد مربوطه:

جناب آقای دکتر حسین خسروی

تاریخ تهیه و ارائه:

ابان ماه ۱۴۰۰

یک سون سگمنت ۶ تایی به میکرو وصل کنید و ساعت دیجیتال را شبیه سازی کنید. دو دکمه تنظیم هم پیش بینی کنید که با فشردن هر کدام، ساعت یا دقیقه افزایش یابد و وقتی به ۵۹ رسید دوباره صفر شود. یک دکمه هم برای ثانیه پیش بینی کنید که با فشردن آن، عدد ثانیه شمار، صفر شود.

نکته ۱: فرکانس میکرو و مقسم فرکانسی تایمر مورد استفاده را طوری تنظیم کنید که یک ثانیه با کیفیت خوبی تولید شود.

نکته ۲: یک دکمه اضافه پیش بینی کنید که با فشردن آن، ثانیه سریعتر تغییر وضع دهد مثال ثانیه شمار به جای اینکه هر یک ثانیه افزایش پیدا کند، هر ۱۰ میلی ثانیه، افزایش پیدا کند. این کار برای ارزیابی شمارنده دقیقه و ساعت لازم است.

- اختیاری: یک Segment-7 دیگر هم پیش بینی کنید که تاریخ شمسی را نمایش دهد و قابل تنظیم باشد.

- فرکانس کاری میکرو در CodeVision و Proteus، ۸ مگا هرتز تنظیم شده است.

کد نوشته شده به صورت زیر است:

```

#include <mega32.h>
#include <delay.h>

#define display PORTC
#define com PORTD
#define display_calendar PORTA
#define calendar_com PORTB
#define set_time PINB.6
#define set PINB.7
#define rest_second PIND.7
#define set_cal PIND.6
#define on 0
#define off 1

unsigned char digit[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F,
0x6F}, hour = 0, minute = 0, second = 0, day = 30, month = 8, year = 0;
unsigned char i = 0, mode = 122;

enum st
{
    live_time,
    set_hour,
    set_minute,
    set_calendar
}state;

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    i++;
}

void main(void)
{
    DDRA = 0xff;
    PORTA = 0x00;

    DDRB = 0xff;
    PORTB = 0x00;

    DDRC = 0xff;
    PORTC = 0x00;

    DDRD = 0xff;
    PORTD = 0x00;

    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Clock value: 31.250 kHz
    // Mode: Normal top=0xFF
    // OC0 output: Disconnected
    // Timer Period: 8.192 ms
    TCCR0=(0<<WGM00) | (0<<COM01) | (0<<COM00) | (0<<WGM01) | (1<<CS02) | (0<<CS01) |
(0<<CS00);
    TCNT0=0x00;
    OCR0=0x00;

```

```

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=(0<<OCIE2) | (0<<TOIE2) | (0<<TICIE1) | (0<<OCIE1A) | (0<<OCIE1B) |
(0<<TOIE1) | (0<<OCIE0) | (1<<TOIE0);

// Global enable interrupts
asm("sei")

while (1)
{
    //===== display calendar =====
    display_calendar = digit[year / 10];
    calendar_com = 0xfe; //1111 1110
    delay_us(250);
    calendar_com = 0xff;
    display_calendar = digit[year % 10];
    calendar_com = 0xfd; //1111 1101
    delay_us(250);
    calendar_com = 0xff;

    display_calendar = digit[month / 10];
    calendar_com = 0xfb; //1111 1011
    delay_us(250);
    calendar_com = 0xff;
    display_calendar = digit[month % 10];
    calendar_com = 0xf7; //1111 0111
    delay_us(250);
    calendar_com = 0xff;

    display_calendar = digit[day / 10];
    calendar_com = 0xef; //1110 1111
    delay_us(250);
    calendar_com = 0xff;
    display_calendar = digit[day % 10];
    calendar_com = 0xdf; //1101 1111
    delay_us(250);
    calendar_com = 0xff;

    //===== display time =====
    display = digit[hour / 10];
    com = 0xfe; //1111 1110
    delay_us(250);
    com = 0xff;
    display = digit[hour % 10];
    com = 0xfd; //1111 1101
    delay_us(250);
    com = 0xff;

    display = digit[minute / 10];
    com = 0xfb; //1111 1011
    delay_us(250);
    com = 0xff;
    display = digit[minute % 10];
    com = 0xf7; //1111 0111
    delay_us(250);
    com = 0xff;

    display = digit[second / 10];
    com = 0xef; //1110 1111
    delay_us(250);

```

```

com = 0xff;
display = digit[second % 10];
com = 0xdf; //1101 1111
delay_us(250);
com = 0xff;

switch(state)
{
    case live_time:
        if(i > mode) //1000ms / 8.192ms = 122
        {
            i = 0;
            second++;
            if(second > 59)
            {
                second = 0;
                minute++;
                if(minute > 59)
                {
                    minute = 0;
                    hour++;
                    if(hour > 12)
                    {
                        hour = minute = second = 0;
                        day++;
                        if(day > 30)
                        {
                            day = 0;
                            month++;
                            if(month > 12)
                            {
                                month = 0;
                                year++;
                            }
                        }
                    }
                }
            }
        }

        if(set_time == 0)
        {
            while(set_time == 0);
            state = set_hour;
        }
        if(set == 0)
        {
            while(set == 0);
            mode = 1;
        }
        if(rest_second == 0)
        {
            while(rest_second == 0);
            second = 0;
        }
        if(set_cal == 0)
        {
            while(set_calendar == 0);
            state = set_calendar;
        }
        break;

```

```

case set_hour:
    if(set_time == 0)
    {
        while(set_time == 0);
        hour++;
        if(hour > 23)
            hour = 0;
    }
    if(set == 0)
    {
        while(set == 0);
        state = set_minute;
    }
    break;

case set_minute:
    if(set_time == 0)
    {
        while(set_time == 0);
        minute++;
        if(minute > 59)
            minute = 0;
    }
    if(set == 0)
    {
        while(set == 0);
        state = live_time;
        mode = 122;
    }
    break;

case set_calendar:
    if(set_cal == 0)
    {
        while(set_cal == 0);
        year++;
        if(year > 99)
            year = 0;
    }

    if(set_time == 0)
    {
        while(set_time == 0);
        month++;
        if(month > 12)
            month = 0;
    }
    if(set == 0)
    {
        while(set == 0);
        day++;
        if(day > 30)
            day = 0;
    }
    if(rest_second == 0)
    {
        while(rest_second == 0);
        state = live_time;
        mode = 122;
    }
}
} //End Switch()

```

```

    }//End while()
} //End main()

```

برنامه شبیه سازی شده در پروتئوس به صورت زیر است:

