**آزمایش 4**

**نمایش چهار رقم**

**1. برای نمایش چهار رقم متفاوت بر روی 7segment از چه روشی استفاده کردید؟**

مقدار تأخیر تایمر را بر روی یک عدد بسیار کوچک، مثلا 1ms تنظیم می‌کنیم و در هر بار اجرای تابع تایمر، تنها یکی از ارقام عدد مورد نظر را نمایش می‌دهیم. چون تغییر رقم به سرعت اتفاق می‌افتد، چشم انسان قادر به تشخیص تغییرات نیست و همه ارقام متفاوت را به صورت ثابت در کنار هم می‌بیند.

**2. برای چشمک‌زن کردن ارقام نمایش داده شده بر روی 7segment از چه روشی استفاده کردید؟**

برای اینکه اعداد چشمک زن شوند، باید مقدار تأخیر تایمر از 1ms فراتر رود. بدین منظور با تعریف متغیر کمکی و شرط بررسی در تابع تایمر، کاری می‌کنیم تا مثلاً از هر 100 بار اجرای تابع تایمر، تنها 1 یک بار کدهای اصلی ما که مربوط به تغییر نمایش رقم مورد نظر است، اجرا شود.

**3. کد برنامه**

char SEG\_DATA[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};

int num = 1400;

char d0 = 0;

char d1 = 0;

char d2 = 0;

char d3 = 0;

char di = 0;

int delay = 1;

int delay\_count = 0;

void setDigits() {

    d0 = (num / 1) % 10;

    d1 = (num / 10) % 10;

    d2 = (num / 100) % 10;

    d3 = (num / 1000) % 10;

}

void switchDigit() {

    setDigits();

    PORTD.7 = 1;

    PORTD.6 = 1;

    PORTD.5 = 1;

    PORTD.4 = 1;

*switch*(di) {

*case* 0:

           PORTD.7 = 0;

           PORTB = SEG\_DATA[d0];

*break*;

*case* 1:

           PORTD.6 = 0;

           PORTB = SEG\_DATA[d1];

*break*;

*case* 2:

           PORTD.5 = 0;

           PORTB = SEG\_DATA[d2];

*break*;

*case* 3:

           PORTD.4 = 0;

           PORTB = SEG\_DATA[d3];

*break*;

    }

    di = (di + 1) % 4;

}

*// External Interrupt 0 service routine*

interrupt [EXT\_INT0] void ext\_int0\_isr(void)

{

*// Place your code here*

*if* (delay == 1) {

        delay = 100;

    } *else* {

        delay = 1;

    }

}

*// Timer 0 overflow interrupt service routine*

interrupt [TIM0\_OVF] void timer0\_ovf\_isr(void)

{

*// Reinitialize Timer 0 value*

TCNT0=0x53;

*// Place your code here*

*if* (delay\_count == 0) {

        switchDigit();

    }

    delay\_count = (delay\_count + 1) % delay;

}