

### گزارش آزمایشگاه 3

کیانا حسینی 39912341054031    امیر حسین فخاری 39912341054050

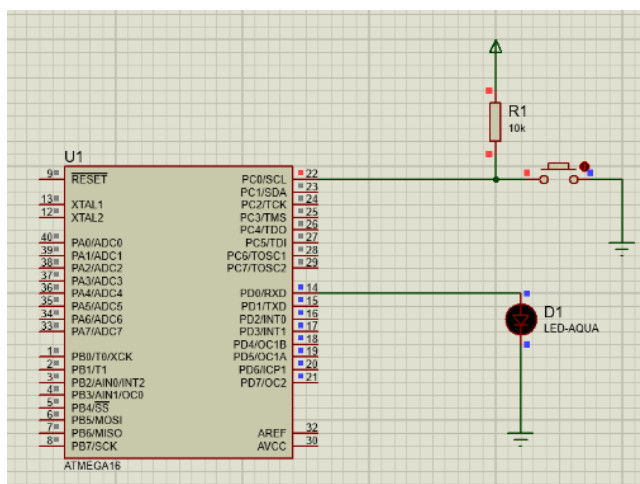
تمرین 1 : با استفاده از کلید (سوئیچ) یک لامپ که در حال تند تند چشمک زدن است را کند کنید.

قطعات مورد نیاز : یک LED- یک مقاومت 10 k -at mega 16- یک کلید یا button

شرح آزمایش : مقاومت به صورت pull up با سوئیچ می بندیم . در حالت قطع کلید ورودی ای که به آی سی atmega می دهد یک می باشد و وقتی سوئیچ وصل شود ورودی ای که می دهد به علت برقراری اتصال با زمین صفر است .

لامپ باید در حالت عادی به فاصله 200 ms روشن و خاموش شود و با فشردن کلید ، چشمک زدنش کند شود و با فاصله 1 ثانیه روشن و خاموش شود . برای این کار سوئیچ با مقاومت PULL UP را به عنوان ورودی آی سی به پورت c وصل می کنیم و LED را به عنوان خروجی به PORTD متصل می کنیم و با استفاده از دستور شرط چک میکنیم که اگر کلید فشرده شود(ورودی از یک به صفر تبدیل شود ) ، فواصل زمانی DELAY بین روشن خاموش شدن LED بیشتر شود .

نتیجه : در هنگام قطع کلید لامپ به سرعت روشن و خاموش می شود و تا وقتی در حال فشردن کلید هستیم چشمک زدنش کند می شود .



```

5 void main(void)
6 {
7     DDRC = 0x00 ;
8     DDRD= 0xff ;
9     while (1)
10    {
11        if (PINC.0 == 0 )
12        {
13            PORTD = 0x01 ;
14            delay_ms(1000);
15            PORTD = 0x00 ;
16            delay_ms(1000);
17        }
18        else
19        {
20            PORTD = 0x01 ;
21            delay_ms(200);
22            PORTD = 0x00 ;
23            delay_ms(200);
24        }
25    }
26 }

```

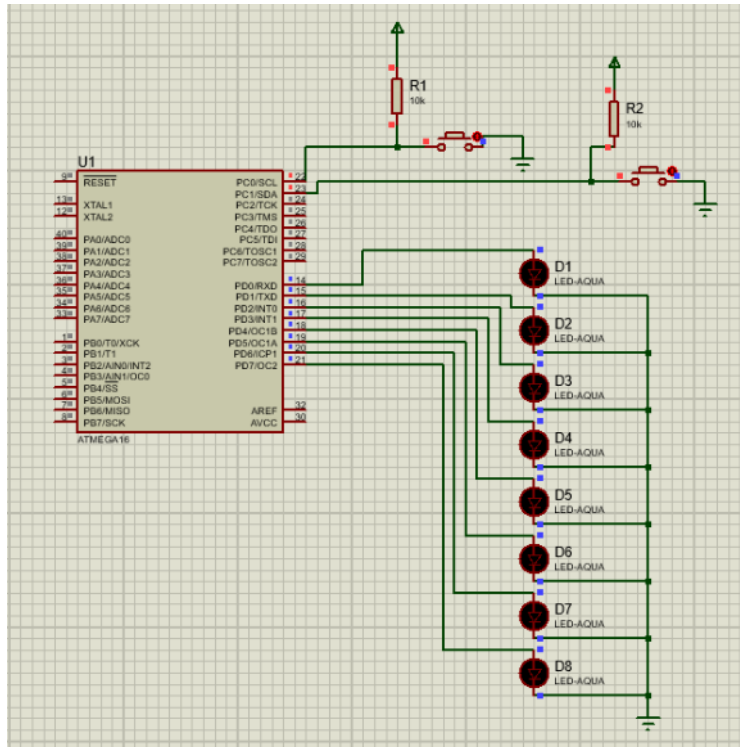
**تمرین 2 : 8 led** داریم . با فشردن کلید اول یکی درمیان روشن خاموش شوند . با فشردن کلید دوم تا درمیان روشن خاموش شوند.

**قطعات مورد نیاز :** 8 عدد LED- دو مقاومت 10 k- 16 mega-at دو کلید یا button

**شرح آزمایش :** هر دو مقاومت را به صورت pullup به دو کلید متصل کرده . خروجی کلید ها در صورت قطع بودن 1 و در صورت وصل بودن 0 است و این دو را به پین اول پورت C متصل می کنیم . 8 led را به 8 پایه پورت D آی سی متصل می کنیم . برای چشمک زدن تمامی led ها به صورت عادی و بدون فشردن کلید ها ، وضعیت تمامی لامپ ها را 1 می کنیم ، بعد از مدت کوتاهی delay وضعیت همه لامپ ها را 0 می کنیم . با if چک میکنیم اگر ورودی اول c صفر شد یعنی کلید اول فشرده شده و لامپ ها باید یکی در میان روشن خاموش شوند . با یک شرط دیگر چک میکنیم اگر ورودی دوم C صفر شد یعنی کلید دوم فشرده شده و لامپ ها باید دوتا درمیان روشن خاموش شوند .

**نتیجه :** در حالت دیفالت و فشرده نشدن هیچ کلید ، همه 8 چراغ تند تند چشمک میزنند .

با فشردن کلید اول و تا زمانی که کلید را میفشاریم و آزاد نمیکنیم ، در هر لحظه کلید ها یکی در میان روشن خاموش هستند با فشردن کلید دوم تا زمانی که کلید را می فشاریم و آزاد نمیکنیم در هر لحظه کلید ها دوتا در میان روشن خاموش هستند . یعنی led اول و دوم روشن ، led سوم و چهارم خاموش و به همین ترتیب .



```

1  \
2  \
3  \
4  \
5  \
6  \
7  \
8  DDRC = 0x00 ;
9  DDRD= 0xff ;
10 while (1)
11 {
12     if (PINC.0 == 0 )
13     {
14         PORTD = 0b01010101 ;
15         delay_ms(200);
16         PORTD = 0b10101010 ;
17         delay_ms(200);
18     }
19     else if (PINC.1 == 0 )
20     {
21         PORTD = 0b00110011 ;
22         delay_ms(200);
23         PORTD = 0b11001100 ;
24         delay_ms(200);
25     }
26     else
27     {
28         PORTD = 0b11111111 ;
29         delay_ms(100);
30         PORTD = 0b00000000 ;
31         delay_ms(100);
32     }
33 }
34 }
35 }

```

**تمرین 3 :** شمارنده 0 تا 9 با استفاده از 7 سگمنت طراحی کنید که با فشردن کلید اول از عدد یکی کم شود و با فشردن کلید دوم یکی اضافه شود .

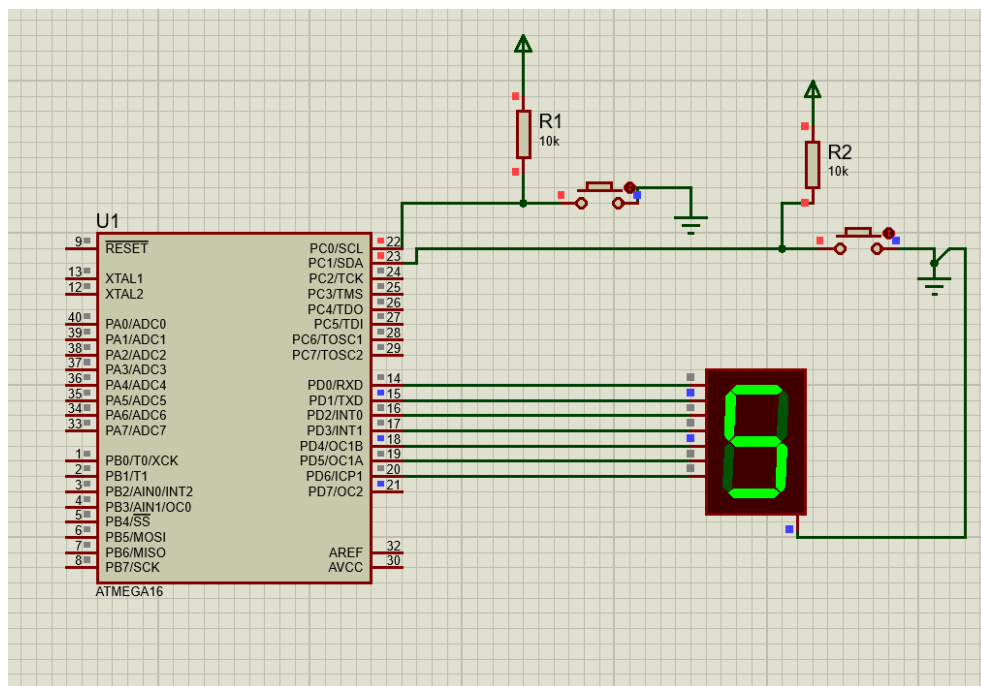
**قطعات مورد نیاز :** یک 7segment - دو مقاومت 10 k -at mega 16- دو کلید یا button

**شرح آزمایش :** ابتدا دو کلید را با مقاومت pull up به دو ورودی اول و دوم PORTC متصل می کنیم . سپس 7 پایه ی یک 7 سگمنت کاتدی را به PORT D متصل می کنیم تا خروجی هارا نمایش دهد .

وقتی کلید اول به عنوان ورودی را فشار می دهیم ( PINC.0 ) باید عدد روی نمایشگر 7 سگمنت یکی افزایش پیدا کند و اگر 9 بود صفر شود و اگر کلید دو فشار داده شود عدد روی نمایشگر یکی کمتر شود و اگر 0 بود 9 شود . برای این کار از یک متغیر شمارنده i استفاده می کنیم که روی index های آرایه ارقام(a) حرکت می کند . چون در ابتدا رقم نمایش داده شده باید صفر باشد  $i = a[0]$  تنظیم میکنیم .

با 2 شرط چک می کنیم که کدام کلید مقدارش 0 شده (فشرده شده ) . برای ثابت شدن و استیبل شدن وضعیت کلید (bounce) پس از انجام عملیات های بعد از فشرده شده کلید ، یک delay میگذاریم.

میخواهیم کاری کنیم که اگر کلید را برای مدتی نگه داریم ، فقط 1 بار عدد تغییر کند برای همین از حلقه while استفاده می کنیم تا درون شرط تا وقتی که کلید فشرده شده بماند .



```

Notes / PROGRAM2.c
1 #include <mega16.h>
2 #include <io.h>
3 #include <delay.h>
4 int a[10] = {0b00111111,0b00000110, 0b01011011, 0b01001111, 0b01100110,
5             0b01101101, 0b01111101,0b00000111,0b01111111,0b01101111} ;
6 int i = 0 ;
7 void main(void)
8 {
9     DDRC = 0x00 ;
10    DDRD= 0xff ;
11    PORTD = a[i];
12
13    while (1)
14    {
15        if (PINC.0 == 0 ){
16            if ( i == 9 ) {
17                i= 0 ;
18            }
19            else{
20                i++;
21            }
22            PORTD = a[i];
23            while ( PINC.0 == 0 ) {
24                // wait for button to unplug
25            }
26            delay_ms(200) ;
27        }
28    }

```

```

    }
    else{
        if (PINC.1 == 0 ){
            if ( i == 0 ) {
                i= 9 ;
            }
            else{
                i--;
            }
            PORTD = a[i];
            while ( PINC.1 == 0 ) {
                // wait for button to unplug
            }
            delay_ms(200);
        }
    }
}

```