

به نام خدا

آزمایش شماره ۱

قسمت اول:

برنامه ای بنویسید که یک ال ای دی را 500 میلی ثانیه روشن و 500 میلی ثانیه خاموش نگه دارد.

برای نوشتن هر برنامه ای چند خط همیشه تکرار می شود که عبارتند از:

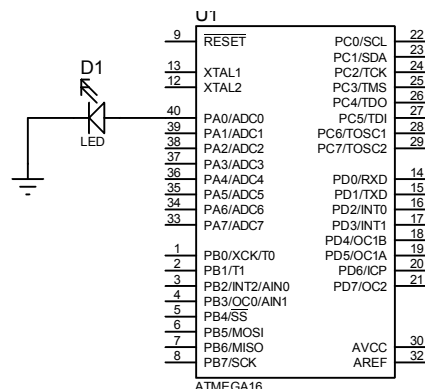
1. **\$regfile** از ای دستور برای معرفی میکرو استفاده می شود.
 2. **\$crystal** از این دستور برای تعیین فرکانس میکرو استفاده می شود.
 3. **End** در پایان هر برنامه باید از این دستور استفاده شود.
- نکته: نحوه به کار بردن این دستورات و سایر دستورات در برنامه ملاحظه شود.

```
$regfile = "m16def.dat"
$crystal = 1000000

Config PORTA = Output

Do
    PORTA.0 = Not PORTA.0
    Waitms 500
Loop

End
```



در خط اول برنامه بالا میکرو کنترلر AVR mega16 معرفی شده است.
 در خط دوم فرکانس سیستم 1 مگاهرتز تعریف شده است.
 با استفاده از دستور Config می توان ورودی یا خروجی بودن پورت را مشخص کرد.
 همانطور که ملاحظه می شود در خط سوم PORTA خروجی تعریف شده است.
 در خط بعد با استفاده از دستور حلقه Do.....Loop یک حلقه ایجاد شده است چون برای حلقه شرطی قرار ندادهایم این یک حلقه ی بی نهایت است، یعنی عبارت دهخل حلقه بی نهایت تکرار می شود.
 برنامه اصلی داخل حلقه Do.....Loop قرار دارد.
 پایه صفرم PORTA را برابر معکوس منطقی خودش قرار داده ایم یعنی در هر بار اجرای حلقه تاگل می شود.
 و در پایان از دستور End استفاده شده است.

قسمت دوم:

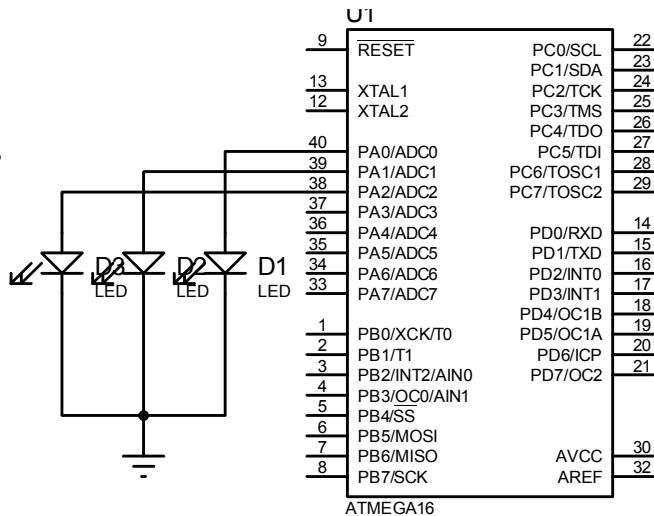
یک شمارنده سه بیتی باینری بنویسید.

```
$regfile = "m16def.dat"
$crystal = 1000000

Config PORTA = Output
Dim N As Byte

Do
  For N = 0 To 7
    PORTA = N
    Waitms 500
  Next
Loop

End
```



با استفاده از دستور Dim متغیر تعریف می شود.
که ما در اینجا متغیر N را از نوع داده (0-255) Byte تعریف کرده ایم.
باز هم با استفاده از دستور Do.....Loop یک حلقه بی نهایت ایجاد کرده ایم.
با استفاده از دستور For.....Next یک حلقه دیگر ایجاد کرده ایم به گونه ای که در این حلقه متغیر از صفر تا هفت می شمارد و بعد از هفت ریست می شود.

قسمت سوم:

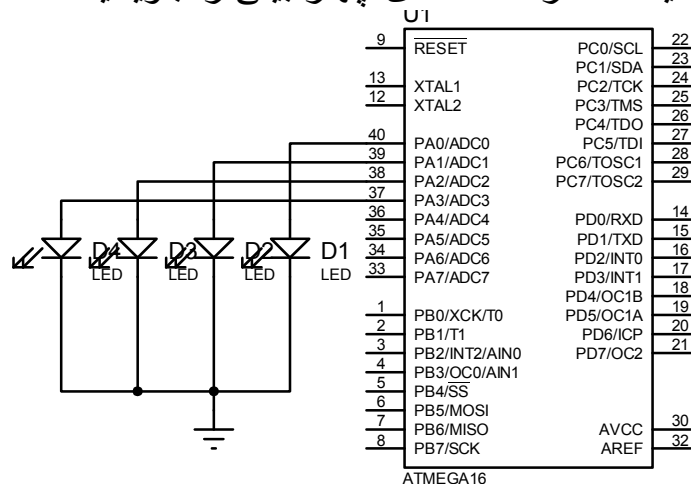
برنامه یک شمارنده حلقه ای چهار بیتی را بنویسید.

```
$regfile = "m16def.dat"
$crystal = 1000000

Config PORTA = Output
Dim I As Byte
Dim X As Byte

Do
  X = &H80
  For I = 0 To 4
    PORTA = X And &H0F
    Waitms 500
    Rotate X, Left
  Next
Loop

End
```



در این برنامه از دو متغیر استفاده شده است که یکی برای کنترل است و دیگری برای قرار دادن اطلاعات مورد نیاز در آن است.
I برای کنترل و X برای اطلاعات.

X برابر عدد باینری 10000000 مقدار دهی می شود.
حلقه For.....Next پنج مرتبه اجرا می شود در هر بار اجرا متغیر X با استفاده از

دستور Rotate X , Left یک خانه به چپ حرکت می کند
 وقتی ا به چهار رسید از حلقه For.....Next خارج شده و وارد حلقه Do.....Loop می شود و X دوباره مقدار دهی و به مقدار اولیه برمی گردد.
 در خط نهم یک خط دستور نوشته شده است که به این عمل اصطلاحاً عمل ماسک کردن گفته می شود این کار برای این است که چهار بیت MSB پورت A دیده نشود.

قسمت چهارم:

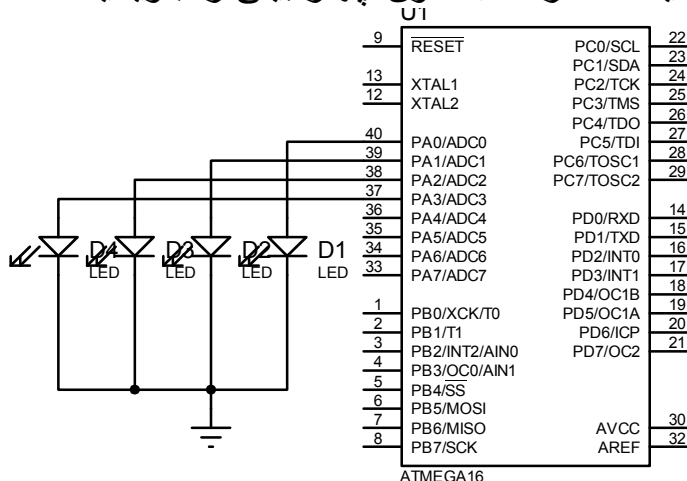
برنامه یک شمارنده جانسون چهار بیتی را بنویسید.

```
$regfile = "m16def.dat"
$crystal = 1000000

Config PORTA = Output
Dim X As Byte
X = &HFO

Do
    PORTA = X And &H0F
    Waitms 500
    Rotate X , Left
Loop

End
```



در این برنامه نیز یک متغیر تعریف شده و سپس برابر مقدار باینری 11110000 قرار گرفته است.

در این برنامه نیز همان طوری که ملاحظه می شود در خط اول حلقه Do.....Loop از عمل ماسک کردن استفاده شده است.
 با استفاده از دستور Rotate X , Left عدد داخل متغیر را در پوات می چرخانیم.

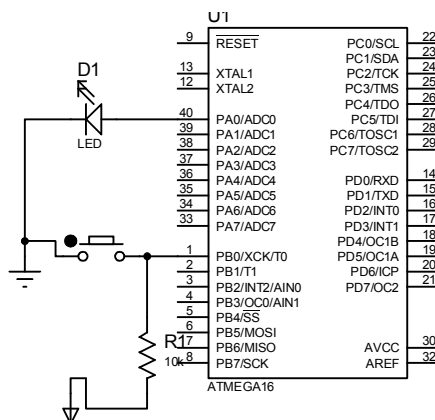
قسمت پنجم:

```
$regfile = "m16def.dat"
$crystal = 1000000

Config PORTA = Output
Config PORTB = Input

Do
    PORTA.0 = Not PINB.0
Loop

End
```



PORTA به دلیل این که می خواهیم نتایجی را روی آن قرار دهیم خروجی
 PORTB به دلیل این که می خواهیم اطلاعاتی را از آن دریافت کنیم ورودی قرار
 می دهیم.

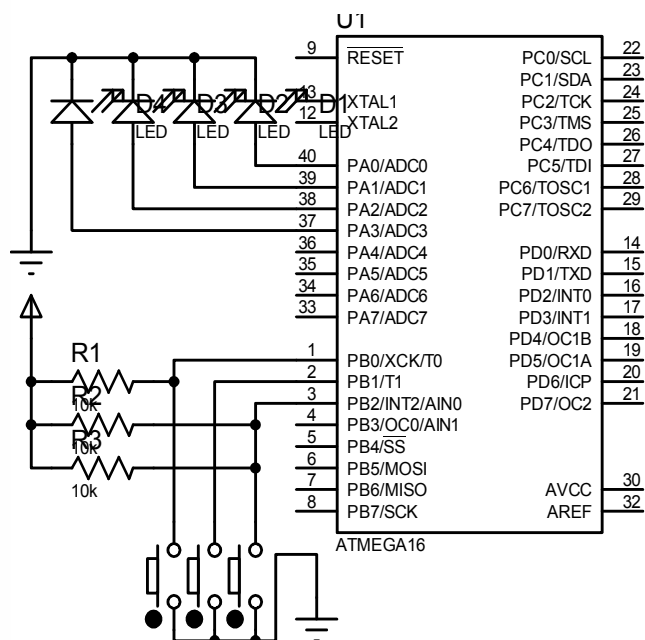
پایه صفرم PORTB به دلیل پول آپ شدن در حالت معمولی فعال است پس برای فهمیدن این که کلید زده شده یا نه باید معکوس آنرا به PORTA ارسال کنیم.

قسمت ششم:

برنامه ای بنویسید که اگر شاسی یک زده شد شمارنده سه بیتی معمولی، اگر شاسی دوم زده شد شمارنده دایره ای و اگر شاسی سوم زده شد شمارنده جانشون شود.

```
$regfile = "m16def.dat"
$crystal = 1000000
```

```
Config PORTA = Output
Config PORTB = Input
Dim Sw As Byte
Dim N As Byte
Dim X As Byte
Dim X2 As Byte
X2 = &HF0
X = &H80
Do
Sw = PINB And &H07
Select Case Sw
Case &H06:
While Sw = &H07 OR Sw = &H06
Sw = PINB And &H07
PORTA = N
N = N + 1
If N = 8 Then N = 0
Waitms 500
Vend
Case &H05:
While Sw = &H07 OR Sw = &H05
Sw = PINB And &H07
PORTA = X And &H0F
Waitms 500
Rotate X, Left
If X = &H10 Then X = &H80
Vend
Case &H03:
While Sw = &H07 OR Sw = &H03
Sw = PINB And &H07
PORTA = X2 And &H0F
Waitms 500
Rotate X2, Left
Vend
End Select
Loop
End
```



با توجه به این که اطلاعات بعد از پردازش را به PORTA می فرستیم خروجی و چون اطلاعات را از PORTB دریافت می کنیم ورودی قرار می دهیم. در این برنامه ما از چهار متغییر به شرح زیر استفاده کرده ایم:

Sw: انتخاب نوع شمارنده

N: متغییر شمانده سه بیتی

X: متغییر شمارنده دایره ای

X2:متغیر شمارنده جانسون

با دستور Select Case که یکی از دستورات حلقه است حالت های احتمالی را تعریف می کنیم.

نکته:

چون PORTB پول آپشده است باید NOT حالت های ممکن را در نظر بگیریم.
برای مثال اگر کلید اول زده شود (110)₂ باینری از PORTB دریافت می شود.

این برنامه ترکیبی از برنامه های گذشته است با این تفاوت که برای اجرای هر کدام از حالات ممکن یک شرط گذاشته می شود تا در آن حلقه بماند.
تمامی شرط ها مانند هم هستند پس به بررسی یکی از شرط ها می پردازیم.
اگر کلید اول زده شود عدد (110)₂ که برابر (05)₁₆ است وارد میکرو می شود که این حالت اول کار مدار است، پس باید شمانده معمولی سه بیتی فعال شود.
حال اگر شاسی را رها کنیم شمارش قطع می شود یعنی (111)₂ به میکرو ارسال می شود. پس باید شرطی بگذاریم که اگر کلیدی زده شد و سپس رها شد مدار به کار خود ادامه دهد.

برای این منظور از دستور حلقه شرطی While.....Wend استفاده می کنیم.
اگر به طرز به کار برده شدن این دستور دقت کنید متوجه می شوید که به این صورت به کار برده شده که:

اگر کلیدی زده شود وارد یک حلقه متناظر با کلید می شود اگر کلیدی زده نشود همچنان منتظر زده شدن کلید می ماند.

حال با فرض این که کلیدی زده شود به این صورت تعریف می شود که وارد حلقه متناظر با کلید می شود اگر کلید رها شد یا اگر کلید دوباره زده شد یا همچنان کلید نگه داشته شده باشد مدار به کار خود ادامه دهد.

نویسنده : حمید لطفعلی زاده

استاد: مهندس قاسم زاده