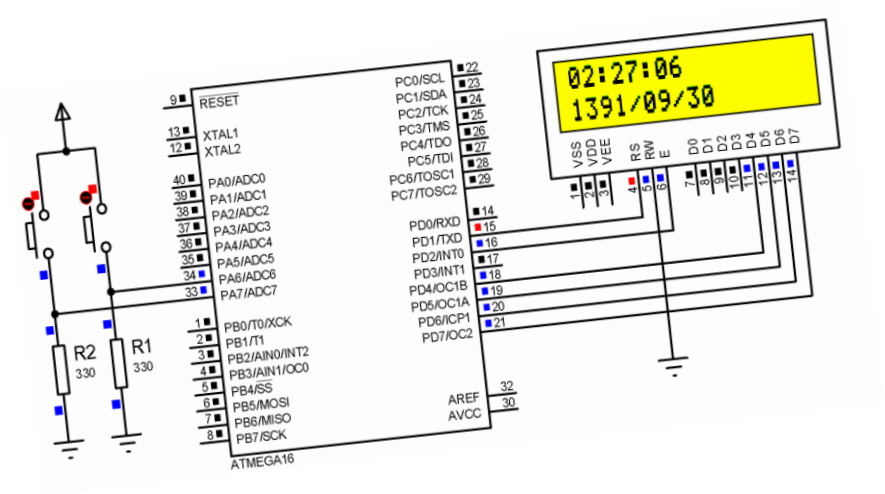
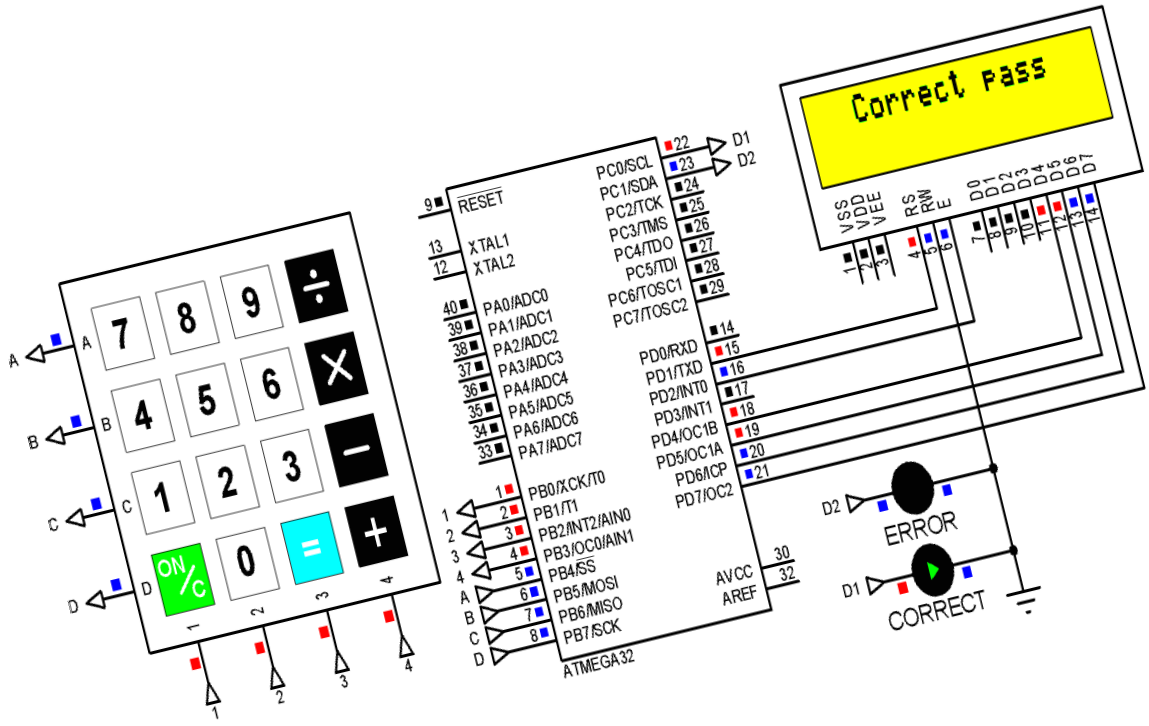
میکرو کنترلرهای

AVR





میکروکنترلرهایAVR

مدرس:مهندس جماعتی کامه علیا

دانشکده فنی مهندسی شهید منتظری

تهیه ونگارش:اسماعیل منفردی

فهرست

میکرو کنترلر یک المان برنامه پذیر به وسیله کاربر است که دارای یک سری امکانات جانبی از قبیل حافظه،

تایمر/کانتر،مولدهای موج مربی،PWM و...می باشد.

انواع میکروکنترلرها:80C51 ،80S51،89S51،AVR ،DSPIC وARM

میکرو کنترلر ARM داری سه مدل 7 ،9 و11است که این مدل ها 32 بیتی هستند.

انواع AVR :

1 - CLASSIC : قدیمی

2 - Tiny : امکانات کم و ابعاد کوچکتر(مورداستفاده درمدارهای مخابراتی)

3 -Mega : امکانات بیشتر و ابعاد بزرگتر

نکته:در مدارهای مخابراتی هرچقدرابعاد مدارکوچکترباشد پهنای باند مدار برای حجم اطلاعات ارسالی بیشتر میشود.

تکنولوژی ← ابعاد مدار → هزینه

معرفی امکانات atmega16 :

1- ازمعماریRISCاستفاده شده است.

معماری کامپیوتربه دو دسته ی CISC(تعداد دستورات زیاد،ابعادمداربیشتر) وRISC (تعداد دستورات کمتر،ابعاد مدارکوچکتر) تقسیم میشود.

نکته : درتکنولوژی یکسان کاهش دستورات باعث افزایش سرعت می شود ولی از طرفی باعث کاهش قدرت کاربر در اعمال برنامه نیز میشود.

2- کارایی بالا و توان مصرفی کم

3- دارای 131 دستورالعمل قدرتمند

4- دارای 32 رجیستر8 بیتی(رجیستر:ذخیره موقت اطلاعات وداخل CPU،هرچه تعداد رجیسترهابیشترباشد سرعتCPUافزایش می یابد .ولی این افزایش سرعت باعث افزایش هزینه نیز میشود)

5- سرعت16MHZ(انجام16میلیون دستور درثانیه)

6- حافظه برنامه وداده غیر فرار : 16KBحافظهFLASH داخلی قابل برنامه ریزی با قابلیت 10000بارR/W

نکته : FLASHها حافظه هایی هستندکه به صورت ضربه ای نوشته یا خوانده می شوند.

7- 1024بایت(1KB) حافظه داخلی SRAM

8- 512بایت حافظه EEPROM داخلی قابل برنامه ریزی با قابلیت10000 بارR/W

9- قفل برنامه FLASH وحفاظت EEPROM

10- قابلیت ارتباط JATG:برنامه ریزیFLASH،EEPROM،فیوز بیت از طریق ارتباط JATG(کابل اتصال)

11- دارای رابط سریال TWIکه اتصال چندین میکروکنترلر را توسط دوباس،دیتا وپالس فراهم می کند.

خصوصیات جانبی :2 عدد تایمر/کانتر8 بیتی با فرکانس مجزا،یک تایمر/کانتر 16 بیتی با فرکانس مجزا،4کانالPWM، 8کانالA/D(ADC)،یک مقایسه کننده آنالوگ داخلی،قابلیت ارتباط با پروتکل دو سیمه I_2C،ارتباط SPI،دارای اسیلاتور داخلی کالیبره شده،دارای چندین وقفه داخلی وخارجی و...

دستورات BASCOM :

معرفی میکرو: $regfile= "اسم میکرو def.dat"

مثال : $regfile=”m16def.dat”

مثال : $regfile=”m8def.dat”

مثال : $regfile=”at12def.dat”

کریستال : برای تولید فرکانس استفاده میشود که می توان این کار را توسط کریستال داخلی(توسط خود میکرو تولید میشود) ویا کریستال خارجی (با استفاده ازنوسان ساز کریستالی و یا نوسان سازهای کولپیتس،هارتلی و...)انجام داد.

فرکانس کریستال تعیین کننده فرکانس کاری میکرو است. f_{max}=16mhz

مرفی کریستال :

$crystal= فرکانس برحسب هرتز

مثال $crystal=8000000

نکته:حتی اگر از کریستال داخلی استفاده کنیم نوشتن دستور فوق الزامی است.

یک سیکل ساعت مدت اجرای هر دستور توسط میکرو است. 125^{ns}=1/8m=فرکانس کار/1=سیکل ساعت

یادداشت اختیاری : برای دادن توضیحات اضافی در مورد برنامه یا دستورمعمولاً از علامت (') استفاده میشود.

مثال : B=45 ' this is for test

پایان برنامه(end) : دستور end در پایان برنامه قرار می گیرد واجرای برنامه را متوقف می کند.با اجرای دستور end تمام وقفه ها غیر فعال شده و یک حلقه بینهایت تولید و برنامه خاتمه می یابد.

مثال : $regfile="m16def.dat"

$crystal=8000000

DO

PORTB=50

LOOP

END

حلقه بینهایت

بعد ازاین دستور میکروهیچ کاری انجام نمی دهد

اعداد،متغییرها و جداول look up :

دیمانسیون متغییر : برای نمایش بعد یک متغییر از دستورزیر استفاده میشود.

Dim var as [xram,sram,eram]

Data type [at location]

نوع داده	اندازه به بیت	رنج قابل قبول
Bit	1	**0,1**
Byte	8	**0 to 255**
Integer	16	**-32768 to 32767**
Word	16	**0 to 65535**
Long	32	**-2147483648 to 2147483647**
Single	32	**3.4* 10^ 38 1.5* 10^ -45**
String	متغییر رشته	

دستور decreas,increase :

decr var یک واحد از متغییر کم میکند

incr var یک واحد به متغییر اضافه میکند

مثال : Decr A ‘A=A-1

مثال : Incr A ‘A=A+1

نکته مهم : در همگام عملیات محاسبات

- در هر خط برنامه بیش از یک دستور عملیاتی وجود نداشته باشد.

- بعد حاصل دو طرف تساوی باید یکسان باشد.

A=B+C+D : به عنوان مثال این دستور اشتباه است ومی توان به روش های زیر اصلاح کرد.

مثال : E=B+C ‘E=B+C

A=E+D ‘ A=E+D

مثال : Dim A as byte,B as word

A=255

B=255

Incr A ‘A=0

Incr B ‘B=256

مثال: Dim A as word,B as integer

A=0

B=0

Decr A ‘A=65535

Decr B ‘B=-1

در صورت استفاده از متغییر string بایستی طول آن نیز قید گردد.

مثال : Dim s as string*10

در مثال فوق متغییر s از نوع string ونهایت به طول 10 کاراکتر است.

نمایش اعداد HEX و BIN :برای نمایش اعدادHEX وbin از دستور &B و&H در ابتدای عدد استفاده می شود.

مثال : PORT B= &B10010001

مثال : PORT B=&H55

تعریف یک PIN یا PORT بصورت خروجی :

1- تعریف یک پورت بصورت خروجی :

Config port x =output x=A or B or C or D

مثال : Config portC=output

2- تعریف یک پایه بصورت خروجی :

Config portX.Y=output x=name of port y=0 to 7

مثال : Config portC.2=output

تعریف یک PIN یا PORT بصورت ورودی :

1- تعریف یک پورت بصورت ورودی :

Config pinx=input

Config portx=input

مثال : Config pinA=input

مثال : Config portA=input

2- تعریف یک پایه بصورت ورودی :

Config pin x.y=input

مثال : Config pin D.2=input

دستورconst :

symbol const : عددی، رشته ای ویا رابطه ریاضی

مثال : Const s=”test”

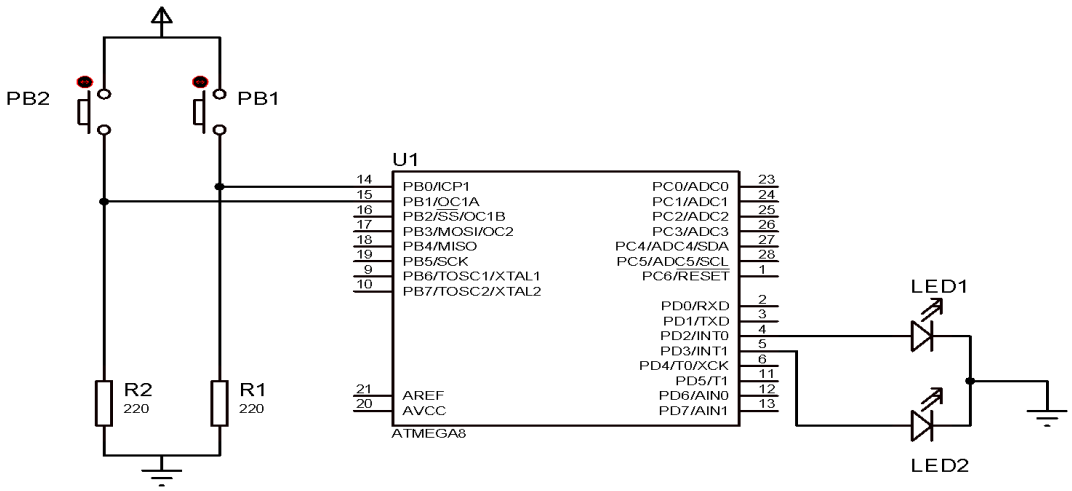
مثال : Const A= 5

مثال : Const B1=&B1011

مثال : Const x=(B1*3)^2

دستور ALIAS : از این دستوربرای تغییر نام یک متغییر استفاده می شود.

مثال : direction alias portb.0




```
$regfile = "m8def.dat"
$crystal=8000000
Config Pinb.0 = Input
Config Pinb.1 = Input
Config Portd.2 = Output
Config Portd.3 = Output
Pb1 Alias Pinb.0
Pb2 Alias Pinb.1
Led1 Alias Portd.2
Led2 Alias Portd.3
```

دستورlow : var=low(X)

دستور فوق 8 بیت پایین متغییر X را در var ذخیره می کند.

مثال : A=low(&H56F2) ‘A=F2H

دستور High : var=high(x)

دستور فوق 8 بیت بالای متغییر X را در var ذخیره می کند.

```
B2=High(&H56F2)        'B2=56H
```

دستور LEN : var=len(string)

این دستور طول یا به عبارتی تعداد کاراکترهای یک رشته را مشخص می کند.

مثال : s="test"

```
A=len(s)               'A=4
```

مثال : S="this is a test"

```
A=len(s)               'A=14
```

دستورMID : var=mid(var1,st,L)

این دستور قسمتی از رشته var1 را با شروع از کاراکتر st و به طول L کاراکتر برداشته و در متغییر var ذخیره می کند.

مثال : dim s as string*15, 2 as string*15

```
s="ABCDEF"
Z=mid(s,2,3)
print Z                'Z=BCD
```

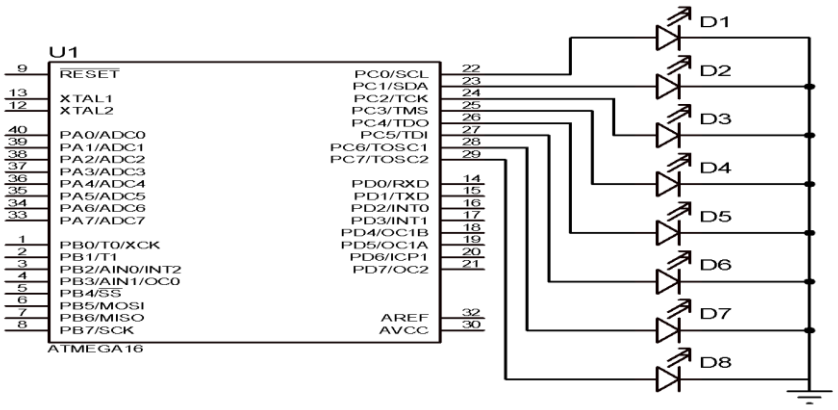
دستورRotate : Rotate var,left/right,shift

```
A=&B01111100
Rotate A,left,2        'A=&B11110001
```

مثال : برنامه ای بنویسید که 8 عدد Led متصل به پورت C را به تربیت هر1 ثانیه روشن نماید.

حالت اول : در هر لحظه فقط یک Led روشن باشد.

حالت دوم : درهر لحظه فقط یک led و led های قبل آن روشن باشد.



حالت اول :

با استفاده از یک الگوریتم ساده

```
$regfile = "m16def.dat"
$crystal=8000000
config    port c=outout
DO
port c=0
wait  1
port c=1
wait  1
port c=2
wait  1
port c=4
wait  1
port c=8
wait  1
port c=16
wait  1
port c=32
wait  1
port c=64
wait  1
port c=128
wait  1
loop
end
```

با استفاده از الگوریتم ریاضی یا دستوری :

```
Dim A As Byte , S As Byte
A=0:S=0
DO
port c=A
A=2^S
incr  S :  wait 1
if  s>7  then  s=0
loop
end
```

با استفاده از دستور rotate :

```
Dim A As Byte
A = 1
Do
Portc = A
Rotate A , Left , 1
Wait 1
Loop
End
```

حالت دوم :

با استفاده از یک الگوریتم ساده

```
DO
port  c=0
wait  1
port c=1
wait  1
port c=3
wait  1
port c=7
wait  1
port c=15
wait  1
port c=31
wait  1
port c=63
wait  1
port c=127
wait  1
port c=255
wait  1
loop
end
```

با استفاده از ساختار ریاضی :

```
Dim A As Byte
s=o
DO
A=2^s
port c=A
incr  s
wait  1
if  s>8   then  s=0
loop
end
```

با استفاده از ساختار Rotate :

```
dim   A  as  byte
A=0
DO
port c=A
Rotate  A,left
incr   A
wait  1
loop
end
```

دستور format :

این دستور یک رشته ی عددی را شکل دهی می کند.

Target = Format(source , "mask")

مثال : s=”123”

s=format(s,”+”) ‘s=+123

مثال : s=”123”

s=format(s,”00000”) ‘s=00123

دستور fusing :

این دستور برای روند کردن رشته های عددی استفاده می شود.

جدول lookup :

توسط این جدول می توان مقدار دلخواهی را از جدولی بازگرداند.

```
var=lookup(value,label)
```

بااستفاده از این دستور،داده مورد نظر از خانه n ام جدول(n=value) خوانده شده که نام جدول با label مشخص شده است وسپس مقدار خوانده شده در متغییر var ذخیره می شود.

1- اسم جدول باید منحصر به فرد باشد

2- جدول بایستی بعد از دستور end نوشته شود

3- خانه های جدول ازصفر به بعد شروع می شود

4- بسته به نوع متغییر var داده از جدول خوانده شده و درvar ذخیره می شود

5- هر خانه جدول 8 بیتی است

6- مقدار داده برگشتی می تواند بین 0 تا65535 باشد

مثال : a=lookup(0 , tab_1) ‘a=4

```
b=lookup(6 , tab1)          ‘b=150
  .
  .
 end
tab1:
data    4,8,16,32,75 ,100,150
```

کاربرد:

1- خواندن اطلاعات از صفحه کلید

2- خواندن دقیق اطلاعات از سنسورها

3- کالیبراسیون

جدول lookupstr :

توسط این جدول می توان رشته دلخواهی را از جدولی برگرداند.

```
var=lookupstr(value , label)
```

توابع ریاضی و محاسباتی :

بزرگتر یا مساوی =< کوچکتریا مساوی => مخالف <>

تابع exp : ‘var=e^{var_1} var=exp(var_1)

تابع log10 : ‘var=$\log_{10} var_1$ var=log10(var_1)

تابع log : ‘var=ln(var_1) var=log(var_1)

تابع rnd : var=rnd(limit)

این دستور یک عدد تصادفی مثبت بین 0 تا limit تولید می کند.

تابع round : برای روند کردن یک عدد اعشاری

مثال: a=round(2.3) ‘a=2

مثال: a=round(2.8) ‘a=3

دستور set : برای یک کردن یک پایه یا یک بیت از یک پورت بکار می رود.

```
set  port x.y
```

دستور reset : برای صفر کردن یک بیت یا یک پایه از پورت به کار می رود.

```
reset   port  x.y
```

دستورtoogle : با استفاده از این دستور می توان مقدار منطقی یک بیت را معکوس کرد(کاربرد:تولید شکل موج مربع)

```
toogle   port  x.y
```

دستور bit wait : bit wait x.y,set/reset

توسط این دستور تا زمانیکه بیت y از پورتx ،set یا reset نشده باشد در خط جاری متوقف می ماند.درصورت برقراری شرط،اجرای برنامه به خط بعد منتقل می شود.

```
bitwait   pin x.y , set/reset
```

دستورالعمل های حلقه و پرش :

دستورgoto وjmp :میکرو وقتی به این دستورات می رسد به آدرسی که مشخص کرده اند می رود

jmp/goto label

نکاتی که باید هنگام نوشتن آدرسlabel رعایت کرد :

1- اسم labelباید منحصر به فرد باشد

2- پایان اسم label علامت " : " قرار دارد

3- حرف ابتدایی label نمی تواند عدد باشد

4- اسم label می تواند ترکیب حرف و عدد باشد

حلقه DO - LOOP :

```
DO
  دستورات
loop [ until شرط ]
```

در این حالت تا زمانی که شرط حلقه برقرار باشد دستورات اجرا می شوند.

نکته : ابتدا دستورات اجرا میشود سپس شرط بررسی میشود.پس حداقل یک بار دستورات اجرا خواهد شد.

مثال : برنامه ای بنویسید که اعداد 1 تا 100 را برروی پورت B نشان دهد.

```
$regfile="m16def.dat"
$crystal=8000000
config PORTB=output
dim a as byte
DO
PORTB=a
incr a
wait 1
loop [until a<100]
end
```

حلقه FOR – NEXT :

```
for   var=start   to   end [step  value]
دستورات
next
```

var متغییری است که به صورت شمارنده عمل می کند که مقدار اولیه آن start ومقدار نهایی آن end.

step value عددی است که مشخص میکند از start تا end با چه گامی حرکت کند.

مثال : for j=1 to 100 [2]

```
portb=j
next
```

مثال : for k=250 to 10 [-5]

```
portb=k
next
```

حلقه while –wend :

این دستورالعمل تشکیل یک حلقه ی تکراری می دهد که تکرار حلقه تا زمانی ادامه می یابد که شرط حلقه برقرارباشد.

قبل از ورود به حلقه شرط حلقه بررسی میشود ودر صورت برقرار بودن شرط دستورات حلقه اجرا میشود.

```
while        شرط
دستورات
wend
```

مثال : a=1

```
while    a<=10
portb=a
wait  1
incr a
wend
```

دستور IF : در حالت کلی ساختار این دستور به صورت زیر است

if شرط then دستورات

end if

در صورتی که شرط برقرار باشد دصتور یا دستورات اجرا میشوند.

1- اگر فقط یک دستور داشته باشیم نیازی به پایان حلقه if (end if) نداریم ودستوربایستی در همان خط نوشته شود.

مثال : if pinb.3=1 then set portb.4

مثال : if pinb.3=1 then incrA

2- اگر تعداد دستورات بیش از یک دستور باشد،بایستی دستورات به خط بعد ازthen انتقال یابد وپایان دستورات با عبارت end if مشخص گردد.

مثال : if a=4 then end if

set port d.2

reset portb.1

wait 1

3- در صورتیکه چند شرط مختلف داشته باشیم که به ازای آنها یک سری دستور یا دستورات مشخص و یکسان اجرا گردد دو حالت زیر رخ خواهد داد.

3-1- if شرط1 and شرط2 then

دستورات

end if

در صورت برقرار بودن شرط 1 و2 دستورات اجرا خواهد شد.

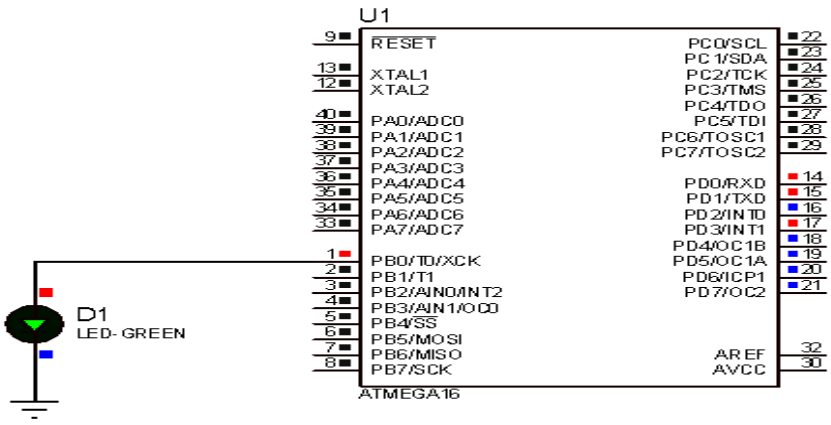
3-2- if شرط1 or شرط2 then

دستورات

end if

در صورت برقرار بودن یکی از شروط 1یا 2 دستورات اجرا خواهد شد.

مثال : برنامه ای بنویسید که اعداد 1 تا 50 را برروی پورت c نمایش دهد با ذکر این نکته که اگر اعداد بین 10 تا 20 بودند یک led نیز روشن شود.

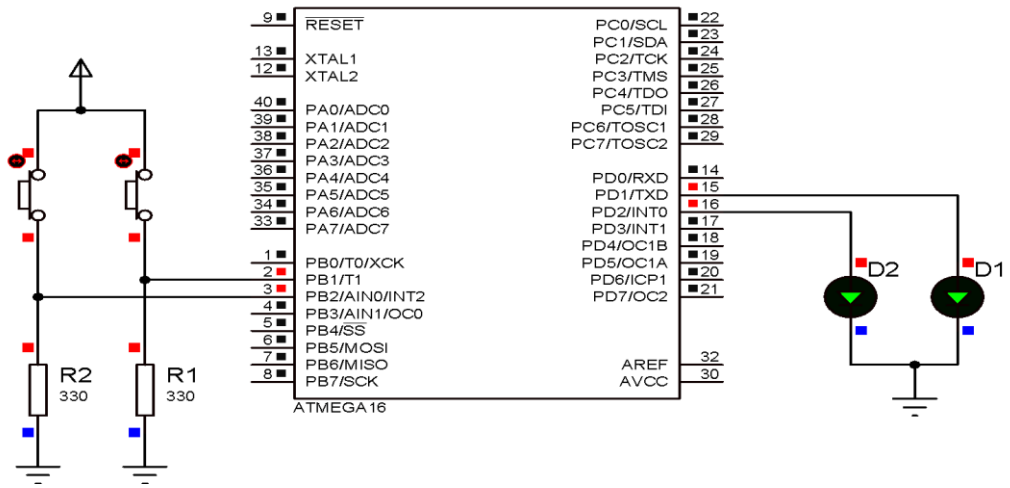



```
$regfile = "m16def.dat"
$crystal = 8000000
Config Portd = Output
Config Portb.0 = Output
Dim A As Byte
A = 0
D0
Portd = A
If A > 10 And A < 20 Then
Set Portb.0
Wait 1
Reset Portb.0
End If
Wait 1
Incr A
Loop
```

4- درصورتی که دو یا بیش از دو دستور یا دستورات متفاوت به ازای شرط مختلف داشته باشیم از ساختار if –else استفاده می شود

```
if    1شرط    then
            1دستورات
else  if    2شرط    then
            2دستورات
else
(end  if )دستورات  و بعد از آن
```

مثال : برنامه ای بنویسید که با توجه به تغییر وضعیت دو کلید متصل به pinb.1 و pinb.2 وضعیت دو led متصل به portd.1 و portd.2 را تغییر دهد.




```
$regfile = "m16def.dat"
$crystal = 8000000
Config Pinb.1 = Input
Config Pinb.2 = Input
Config Portd.1 = Output
Config Portd.2 = Output
Do
If Pinb.1 = 0 And Pinb.2 = 0 Then
Reset Portd.1 : Reset Portd.2
End If
If Pinb.1 = 1 And Pinb.2 = 0 Then
Set Portd.1 : Reset Portd.2
End If
If Pinb.1 = 0 And Pinb.2 = 1 Then
Reset Portd.1 : Set Portd.2
End If
If Pinb.1 = 1 And Pinb.2 = 1 Then
Set Portd.1 : Set Portd.2
End If
Loop
End
```

یا به طور ساده تر می توان نوشت :

```
$regfile = "m16def.dat"
$crystal = 8000000
Config Pinb.1 = Input
Config Pinb.2 = Input
Config Portd.1 = Output
Config Portd.2 = Output
Do
If Pinb.1 = 0 And Pinb.2 = 0 Then
Reset Portd.1 : Reset Portd.2
Elseif Pinb.1 = 1 And Pinb.2 = 0 Then
```

```
Set Portd.1 : Reset Portd.2
Elseif Pinb.1 = 0 And Pinb.2 = 1 Then
Reset Portd.1 : Set Portd.2
Else
Set Portd.1 : Set Portd.2
End If
Loop
end
```

حالت دیگر :

```
$regfile = "m16def.da"
$crystal = 8000000
Config Pinb.1 = Input
Config Pinb.2 = Input
Config Portd.1 = Output
Config Portd.2 = Output
Do
If Pinb.1 = 1 Then
Set Portd.1
Else
Reset Portd.1
End If
If Pinb.2 = 1 Then
Set Portd.2
Else
Reset Portd.2
End If
Loop
```

وساده ترین برنامه ممکن :

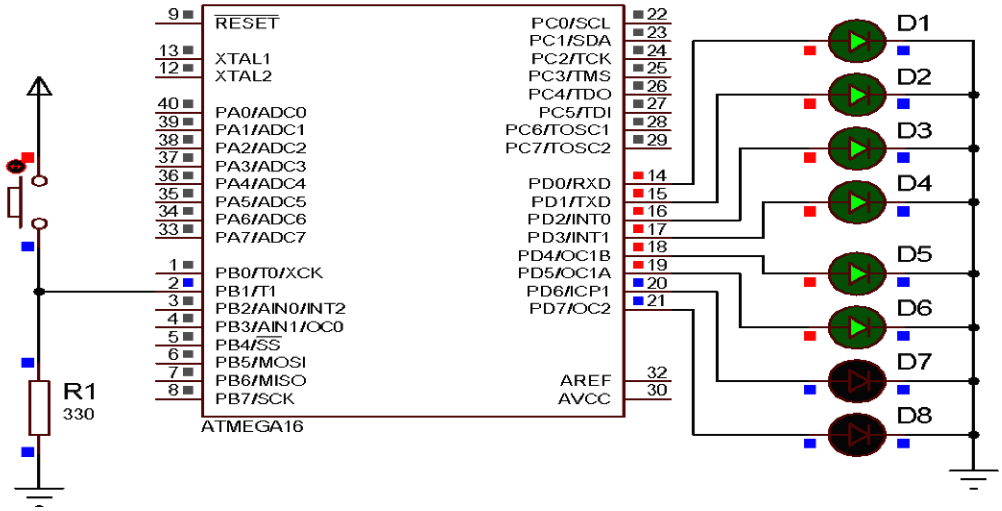
```
$regfile = "m16def.dat"
$crystal = 8000000
Config Pinb.1 = Input
Config Pinb.2 = Input
Config Portd.1 = Output
Config Portd.2 = Output
Do
Portd.1 = Pinb.1 : Portd.2 = Pinb.2
Loop
End
```

دستور select case :

با استفاده از این دستورالعمل می توان یکی از چندین دستورات را با توجه به var اجرا نمود.

```
select case  var
case test1 :   دستورات1
case test2:   دستورات2
case test3:   دستورات3
end select
```

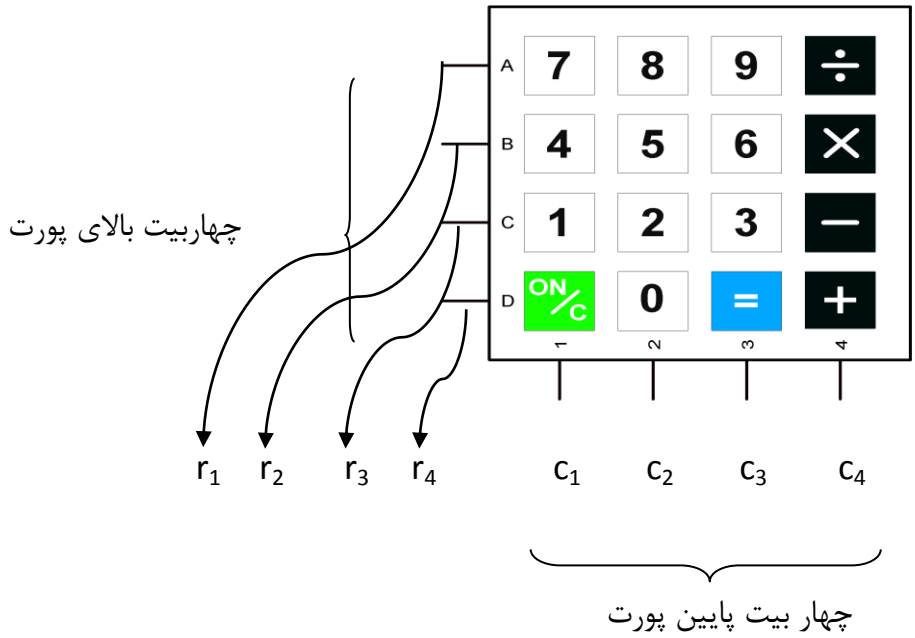
مثال : برنامه ای بنویسید که با هر بار فشاردادن کلید یک واحد به متغییر اضافه نمایید ودر صورتیکه بزرگتر از8 شود دوباره صفرشود.با توجه به عدد متغییر ledهای متصل به پورت d روشن گردد.




```
$regfile = "m16def.dat"
$crystal = 8000000
Config Pinb.1 = Input
Config Portd = Output
Dim A As Byte
A = 0
Do
If Pinb.1 = 1 Then
Incr A
Wait 1
End If
If A > 8 Then A = 0
Select Case A
Case 0 : Portd = 0
Case 1 : Portd = 1
Case 2 : Portd = 3
Case 3 : Portd = 7
Case 4 : Portd = 15
Case 5 : Portd = 31
Case 6 : Portd = 63
Case 7 : Portd = 127
Case 8 : Portd = 255
End Select
Loop
End
```

کار با امکانات AVR :

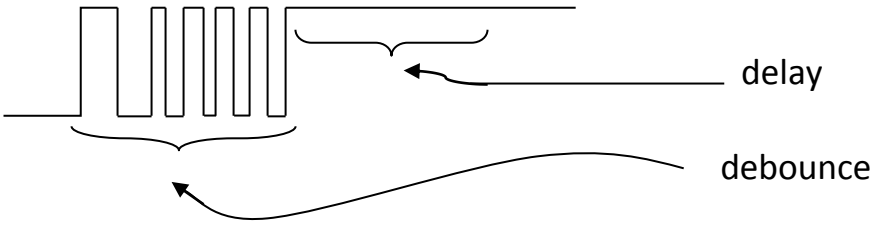
پیکربندی صفحه کلید4 ×4 :



config kbd=portx , debounce=val1 , delay=val2

debounce (لرزش) : برای از بین بردن لرزش کلید استفاده میشود.بصورت پیش فرض 20_{ms} است.برای تغییرآن

می توان به val1 مقداری بین 0 تا 255 داد.از نظر عملی بهتر است که val1=50 باشد



delay : پس از تأخیر برای از بین بردن لرزش کلید توسط دستور debounce از دستور delay می توان برای تأخیر

بیشتر در خواندن کلید استفاده نمود.

نکته : زمانی که نیاز است صفحه کلید با ساختار6×4 را بخوانید می توانید از دستورزیر به جای دستور فوق استفاده کنید

config kbd=portx , debounce=val1 , delay=val2 ,row 5 =pinx.y ,row 6=pinx.y

دستور get kbd( ) :

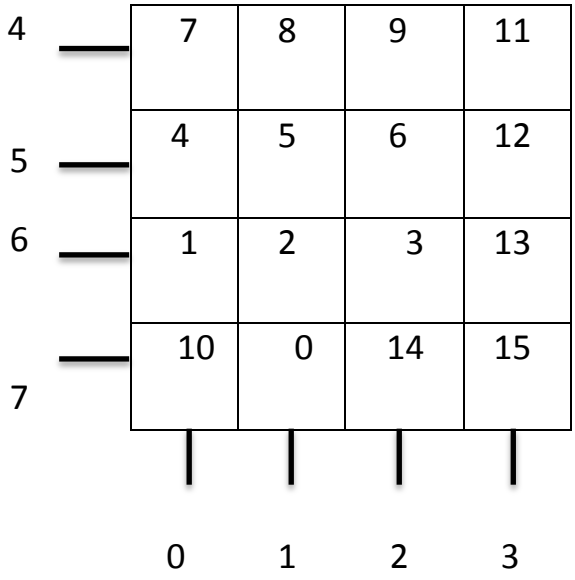
از این دستورمی توان برای خواندن کلید استفاده نمود.ساختار این دستور به صورت زیر است.

```
var=get kbd( )
```

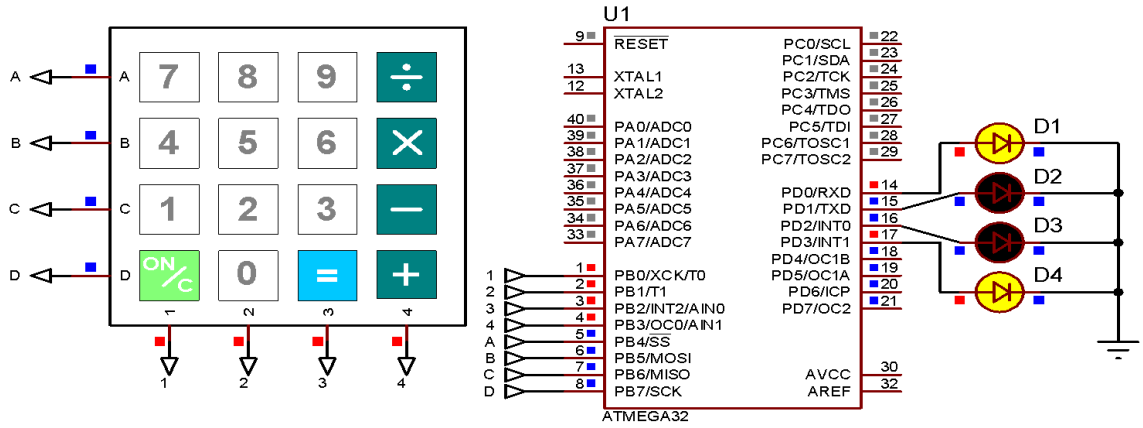
بعد از خواندن کلید مقدار آن را در متغییر var ذخیره می کند.نوع var از byte می باشد.

اگر کلیدی فشرده شود،کد کلید که عددی بین 0 تا15 است در varذخیره می شود.

در صورتیکه هیچ کلیدی فشرده نشود مقدار var=16 خواهد بود.

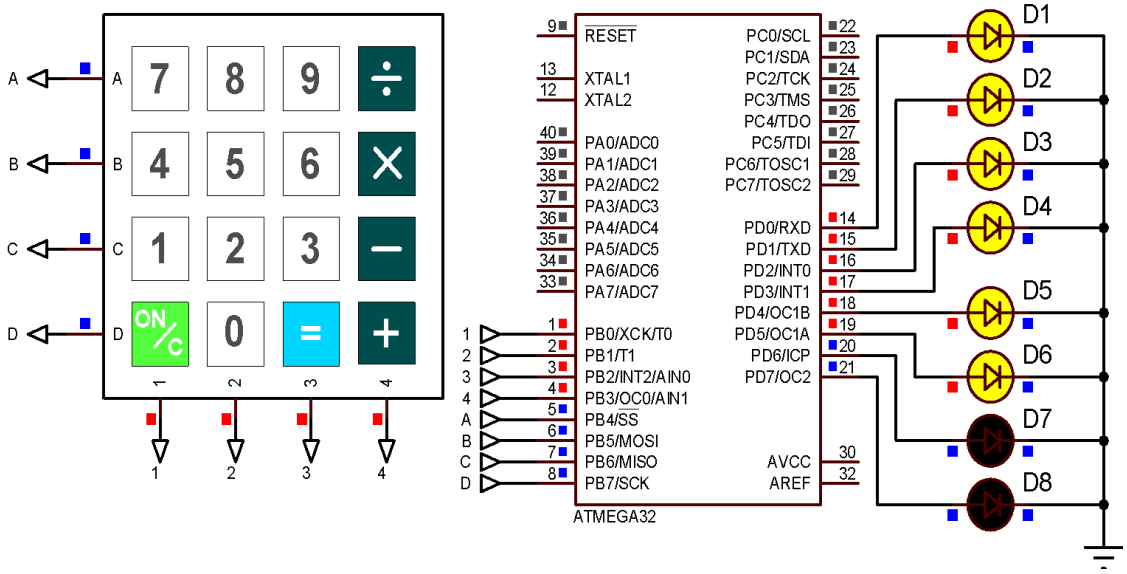


مثال : برنامه ای بنویسید که یک عدد یک رقمی را ازصفحه کلید گرفته و بر روی پورت d نمایش دهد .




```
$regfile = "m32def.dat"
$crystal = 8000000
Config Portd = Output
Config Kbd = Portb , Debounce = 50 ,
Delay = 100
Dim A As Byte
Do
L1:
A = Getkbd()
If A = 16 Then Goto L1
A = Lookup(a , Dat1)
If A > 9 Then Goto L1
Portd = A
Loop
End
Dat1:
Data  7 , 8 , 9 , 11,4 , 5 , 6 , 12,1 , 2 ,
3 , 13,10 , 0 , 14 , 15
```

مثال : برنامه ای بنویسید که متناسب با کلید فشرده شده(0~8) به همان تعداد led روشن نماید.



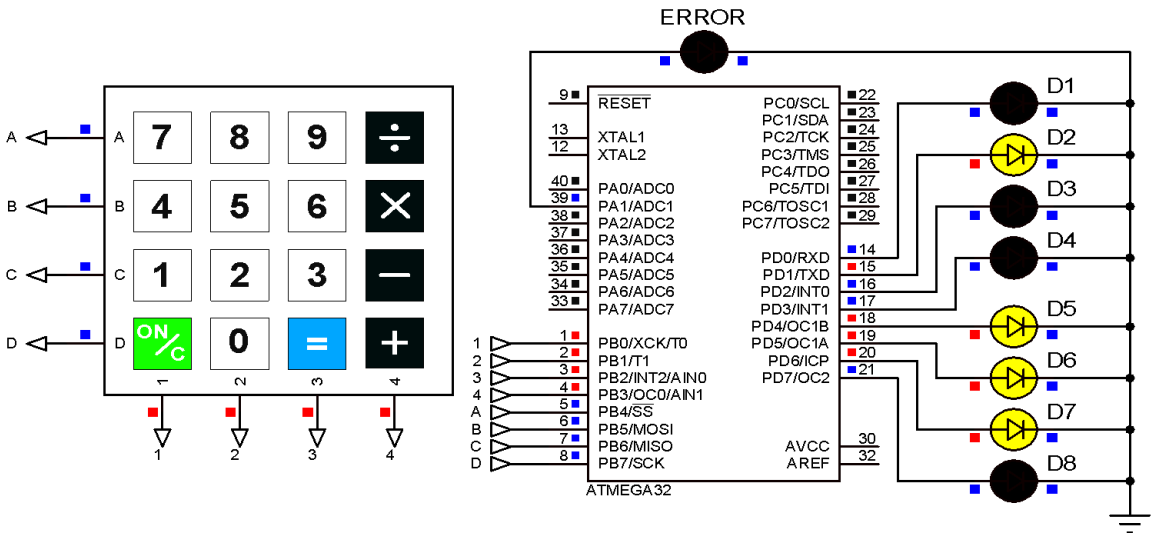
```
$regfile = "m32def.dat"
$regfile = "m32def.dat"
$crystal = 8000000
Config Portd = Output
Config Kbd = Portb ,
Debounce = 50 , Delay = 100
Dim A As Byte
Do
L1:
A = Getkbd()
If A = 16 Then Goto L1
A = Lookup(a , Dat1)
If A = 16 Then Goto L1
Portd = A
Loop
End
Dat1:
Data 127 , 255 , 16 , 16 , 15 , 31 , 63 , 16 , 1 , 3 , 7 , 16 , 16 , 0 , 16 , 16
```

به این صورت نیز می توان نوشت (همان برنامه قبلی که می توان آن را با دستورcase نوشت)

```
$crystal = 8000000
Config Portd = Output
Config Kbd = Portb ,
Debounce = 50 , Delay = 100
Dim A As Byte
Do
L1:
A = Getkbd()
If A > 15 Then Goto L1
A = Lookup(a , Dat1)
If A > 8 Then Goto L1
Select Case A
Case 0 : Portd = 0
Case 1 : Portd = 1
Case 2 : Portd = 3
Case 3 : Portd = 7
Case 4 : Portd = 15
Case 5 : Portd = 31
Case 6 : Portd = 63
Case 7 : Portd = 127
Case 8 : Portd = 255
End Select
Loop
End
Dat1:
Data 7 , 8 , 9 , 11 , 4 , 5 , 6 , 12 , 1 , 2 , 3 , 13 , 10 , 0 , 14 , 15
```

مثال : برنامه ای بنویسید که یک عدد سه رقمی (0~255) را از صفحه کلید گرفته و بر روی پورت d نمایش دهد.

اگرعدد وارد شده بزرگتراز255بود led-error را 5ثانیه روشن نگهدارد ودر این مدت عددی را نپذیرد.




```
$regfile = "m32def.dat"
$crystal = 8000000
Config Portd = Output
Config Pina.1 = Output
Config Kbd = Portb ,
Debounce = 50 , Delay = 100
Dim A As Byte , B As Byte
DimC As Byte , Pass As Word
Do
L1:
A = Getkbd()
If A > 15 Then Goto L1
A = Lookup(a , Dat1)
If A > 9 Then Goto L1
L2:
B = Getkbd()
If B > 15 Then Goto L2
B = Lookup(b , Dat1)
If B > 9 Then Goto L2
L3:
C = Getkbd()
If C > 15 Then Goto L3
C = Lookup(c , Dat1)
If C > 9 Then Goto L3
Pass = A * 100
B = B * 10
B = B + C
Pass = Pass + B
If Pass > 255 Then
Portd = 0
Set Porta.1 : Wait 5 : Reset Porta.1
 Goto L1
End If
Portd = Pass
```

```
Loop
End
Dat1:
Data 7 , 8 , 9 , 11 , 4 , 5 , 6 , 12 , 1 ,
 2 , 3 , 13 , 10 , 0 , 14 , 15
```

به این صورت نیز میتوان نوشت :

```
$regfile = "m32def.dat"
$crystal = 8000000
Config Portd = Output
Config Pina.1 = Output
Config Kbd = Portb ,
Debounce = 50 ,Delay = 100
Dim A(3) As Byte , I As Byte
Dim Pass As Word , B As Byte
L2:
Do
For I = 1 To 3
L1:
A(i) = Getkbd()
If A(i) > 15 Then Goto L1
A(i) = Lookup(a(i) , Dat1)
If A(i) > 9 Then Goto L1
Next
Pass = A(1) * 100
B = A(2) * 10
B = B + A(3)
Pass = Pass + B
If Pass > 255 Then
Portd = 0
Set Porta.1 : Wait 5 : Reset Porta.1
Goto L2
End If
Portd = Pass
Loop
End
Dat1:
Data 7 , 8 , 9 , 11 , 4 , 5 , 6 , 12 , 1 ,
2 , 3 , 13 , 10 , 0 , 14 , 15
```

پیکر بندی lcd کاراکتری

صفحه lcd کاراکتری توانایی نمایش اعداد و حروف انگلیسی را دارد.

LCD دارای دونوع : کاراکتری (1×16،2×16،1×20،2×20، 4×20 و...) وپیکسلی میباشد.

انواع lcdپیکسلی : ساده (64×128 ،128×128، 64×256) و رنگی ((16m) 320×280 و....)

پایه های lcd : داده (D0~D7) و فرمان(RS,EN ,RD/WR)

اتصال پایه ها: VSS:1 ⟵ Ground

VDD: 2 ⟵ +5

VO: 3 ⟵ Controst

RS : 4

R/W: 5

En: 6

D0~D7 : 7~14

led back light: 15,16

پایه شماره 5 جهت خواندن یا نوشتن در صفحه ی lcd به کار می رود.با توجه به اینکه از صفحات lcd کاراکتری نمی توان عملیات خواندن را انجام داد.این پایه همواره به زمین وصل می شود.

تعیین نوع lcd :

Config lcd=lcd type

مثال : Config lcd=20*2

در حالت پیش فرض نوع lcd ،1*16 در نظر گرفته می شود.

پیکربندی باس lcd :

برای ارسال داده بر روی lcd می توان از چهار پایه D4~D7 استفاده نمود.

درصورتیکه بخواهیم از D0~D7 برای ارسال داده استفاده شود بایستی دستورconfig lcd bus=8 پیکر بندی گردد.

در حالت پیش فرض از 4 پایه برای ارسال داده استفاده می شود.

اتصال پایه های lcd به میکرو :

Config lcd=pin , db4=pinx.y_a , db5=pinx.y_b , db6=pinx.y_c , db7=pinx.y_d ,Rs=pinx.y_e ,E=pinx.y_f

مثال : Config lcd=pin , db4=pinb.2, db5=pinb.3 , db6=pinb.4, db7=pinb.5,Rs=pinb.0,E=pinb.1

دستور lcd x :

X می تواند یک عدد ثابت،یک رشته و یا یک متغییر و یک کاراکتر باشد.

مثال : Lcd 23

مثال : Lcd “23”

مثال : B=45

Lcd “b”

مثال : S=”this is a test”

Lcd s

دستورhome :

با استفاده از این دستور می توان مکان نما را به اولین ستون سطری که در آن قرار دارد انتقال داد.

دستورlocate : locate x.y

از این دستور برای مشخص کردن محل نمایش استفاده می شود.(y ستون ،x سطر)

برای نمایش چند کاراکتر از(;) استفاده می شود.

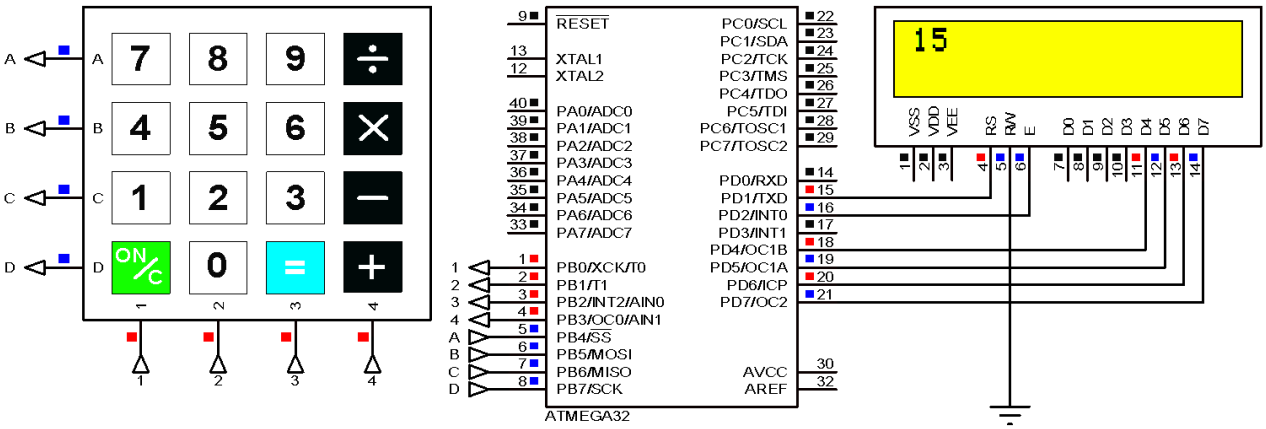
مثال : Lcd a ; b ; “hi” ; 13

برای نوشتن چند دستوراز(:) استفاده می شود.

مثال : Cls : home : lcd a

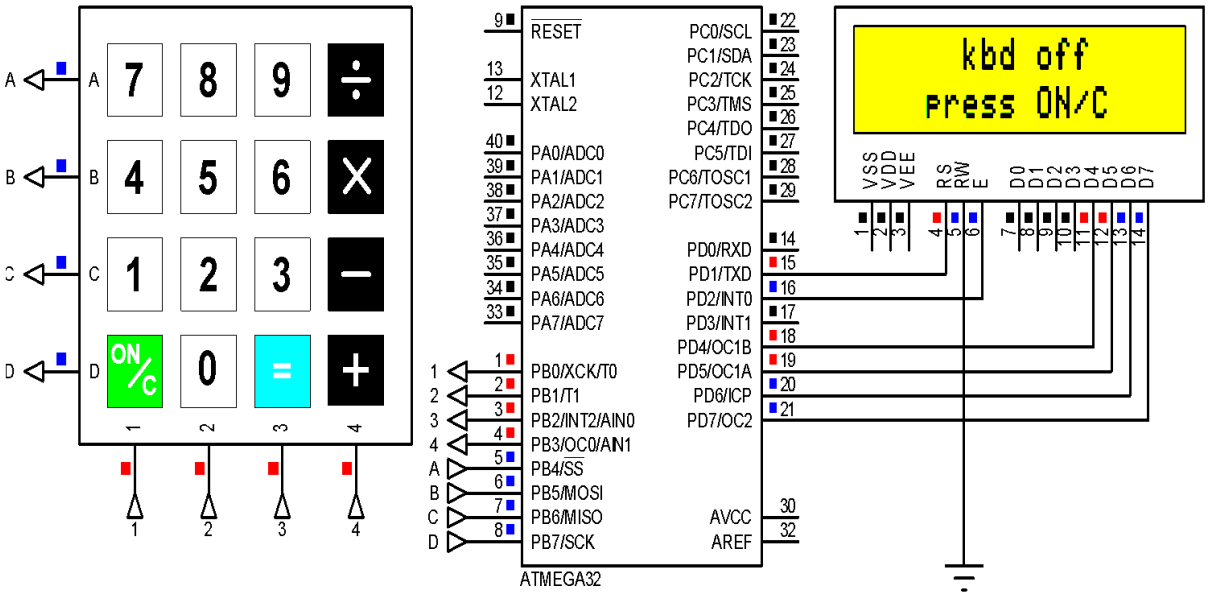
مثال : Cls : locate 2,14 : lcd 3

مثال : برنامه ای بنویسید که عدد را از صفحه کلید گرفته وبرروی lcdنمایش داده شود




```
$regfile = "m32def.dat"

$crystal = 8000000

Config Lcdpin = Pin , Db4 = Pind.4 , Db5 = Pind.5 , Db6 = Pind.6 , Db7 = Pind.7 ,

Rs = Pind.1 , E = Pind.2

Config Lcd = 16 * 1

Config Kbd = Portb , Debounce = 50 ,

 Delay = 100

Dim A As Byte

Do

L1:

A = Getkbd()

If A = 16 Then Goto L1

A = Lookup(a , Dat1)

Cls :cursor off: Home : Lcd A

Loop

End

Dat1:

Data 7 , 8 , 9 , 11 , 4 , 5 , 6 , 12 , 1 , 2 , 3 , 13 , 10 , 0 , 14 , 15
```

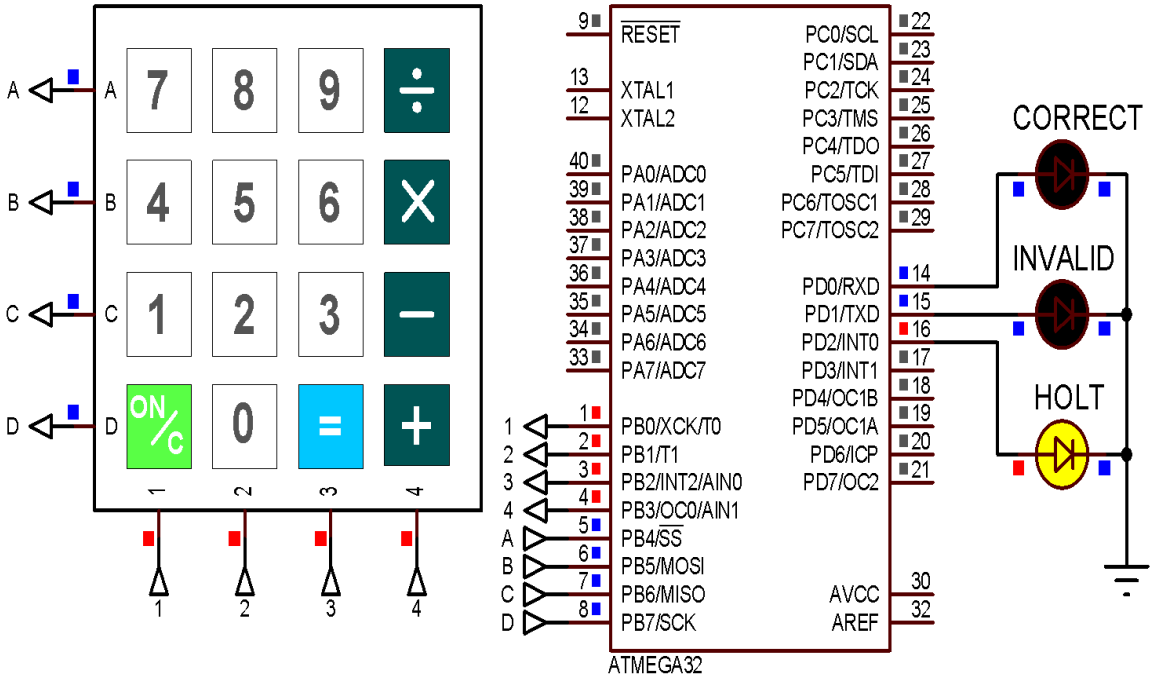
مثال : برنامه ای بنویسید که عدد را از صفحه کلید دریافت نموده و بر روی lcd نمایش دهد.اگر کلید فشرده شده مربوط به اعمال ریاضی بود(+،-،/،*،=) بود،کاراکتر آنها را نمایش دهد.در ابتدا صفحه کلید قفل باشد وپیغام kbd off بروی lcd نمایش داده شود وبا زدن کلید ON/C فعال شود و با زدن دوباره آن غیرفعال شود .




```
$regfile = "m32def.dat"
$crystal = 8000000
Config Lcdpin = Pin , Db4 = Pind.4 ,
 Db5 = Pind.5 , Db6 = Pind.6 ,
 Db7 = Pind.7 , Rs = Pind.1 ,
 E = Pind.2
Config Lcd = 16 * 2
Config Kbd = Portb ,
Debounce = 50 , Delay = 100
Dim A As Byte
Do
L1:
Cls : Locate 1 , 6
Lcd "kbd off"
locate 2,4:lcd"press ON/C"
A = Getkbd()
If A = 16 Then Goto L1
A = Lookup(a , Dat1)
If A <> 10 Then Goto L1
Cls : Home : Cursor Off
Lcd "enter:"
L2:
A = Getkbd()
If A > 15 Then Goto L2
A = Lookup(a , Dat1)
If A = 10 Then Goto L1
```

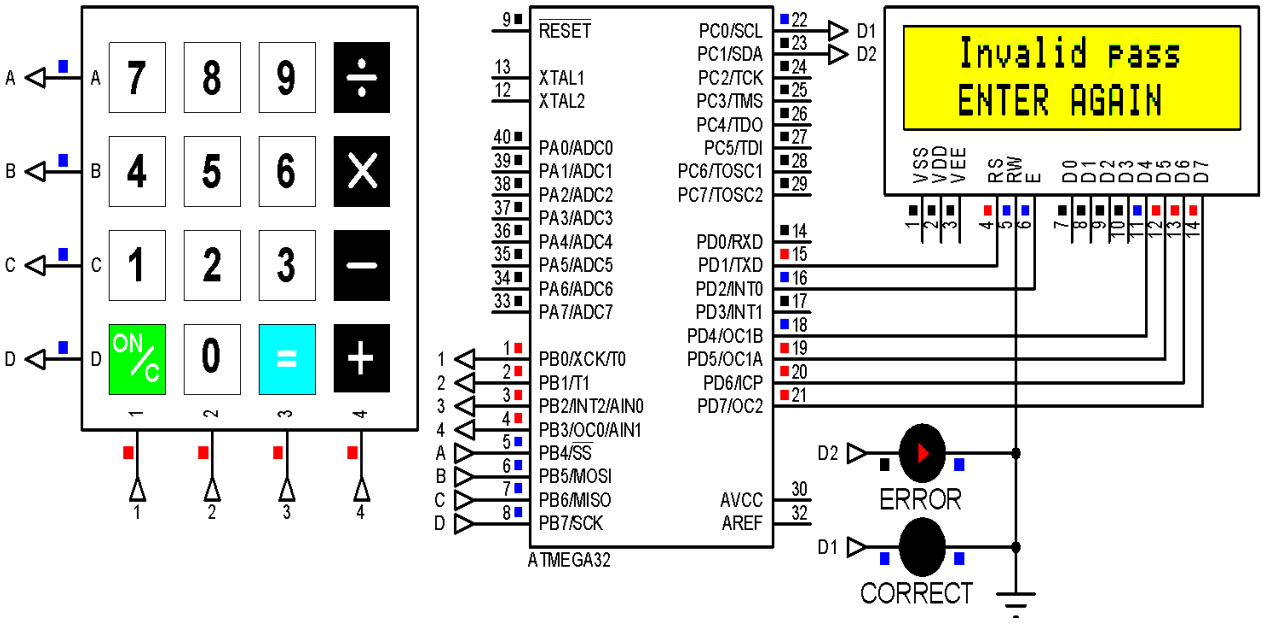
```
If A > 10 Then Goto L3
Cls : Cursor Off : Lcd "enter:" ; A
 Goto L2
Loop
End
Dat1:
Data 7 , 8 , 9 , 11 , 4 , 5 , 6 , 12 , 1 , 2
, 3 , 13 , 10 , 0 , 14 , 15
L3:
select case a
Case 11:
Locate 1 , 7 : Lcd "/"
Case 12:
Locate 1 , 7 : Lcd "*"
Case 13:
Locate 1 , 7 : Lcd "-"
Case 14:
Locate 1 , 7 : Lcd "="
Case 15:
Locate 1 , 7 : Lcd "+"
End Select
Goto L2
Return
```

مثال : برنامه ای بنویسید که یک عدد سه رقمی را از کاربر گرفته و اگر با 999 برابر باشد led سبز را به مدت 1s روشن نگه دارد و در صورت غلط بودن led قرمز را 1s روشن نگه دارد.اگر سه بار پسورد اشتباه وارد شد led زرد روشن شود و از صفحه کلید،کلیدی دریافت نکند.



```
$regfile = "m32def.dat"
$crystal = 8000000
Config Pind.0 = Output
Config Pind.1 = Output
Config Pind.2 = Output
Config Kbd = Portb , Debounce = 50
, Delay = 100
Dim A(3) As Byte , I As Byte
Dim B As Word , Pass As Word
Dim Error As Byte
Do
For I = 1 To 3
L1:
A(i) = Getkbd()
If A(i) = 16 Then Goto L1
A(i) = Lookup(a(i) , Dat1)
If A(i) > 9 Then Goto L1
Next
B = A(1) * 100
A(2) = A(2) * 10
Pass = B + A(2)
Pass = Pass + A(3)
If Pass = 999 Then
Set Portd.0
Wait 1
Reset Portd.0
Error = 0
Else
Incr Error
Set Portd.1
Wait 1
Reset Portd.1
End If
If Error = 3 Then
Set Portd.2
L2:
Goto L2
End If
Loop
End
Dat1:
Data 7 , 8 , 9 , 11 , 4 , 5 , 6 , 12 , 1 , 2
 , 3 , 13 , 10 , 0 , 14 , 15
```

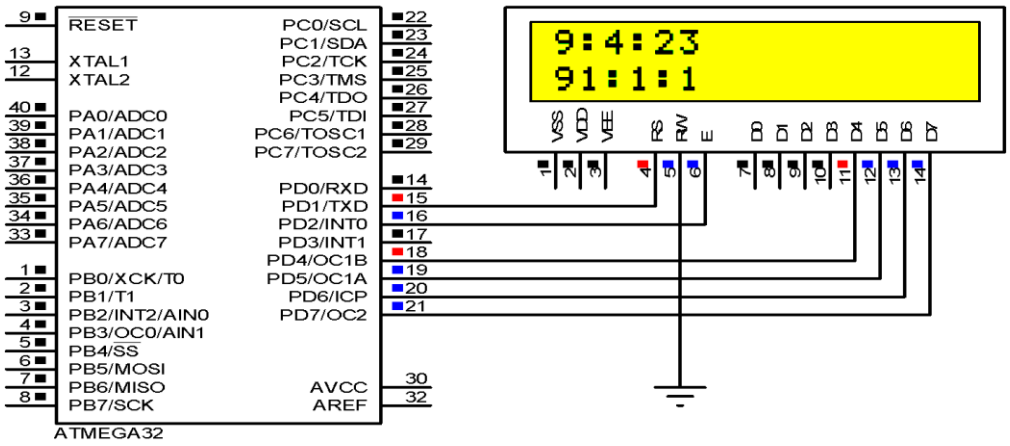
مثال : برنامه ای بنویسید که از کاربر یک عدد چهار رقمی را گرفته و با passاصلی مقایسه کند ودر صورت برابر بودن پیغام correct pass نمایش داده شود و در غیر این صورت invalid pass را نمایش دهد واگر سه بار passاشتباه باشد صفحه کلید قفل شود و پیغام holt نمایش داده شود.(صفحه کلید در ابتدا قفل باشد وپیغام kbd off واگردر هر زمان ازبرنامه که قفل صفحه کلید زده شد،صفحه کلید قفل شود)




```
$regfile = "m32def.dat"
$crystal = 8000000
Config Pinc.0 = Output
Config Pinc.1 = Output
Config Lcdpin = Pin , Db4 = Pind.4 ,
 Db5 = Pind.5 , Db6 = Pind.6 ,
Db7 = Pind.7 , Rs = Pind.1 ,
 E = Pind.2
Config Lcd = 16 * 2
Config Kbd = Portb , Debounce = 50
 , Delay = 100
Dim A(4) As Byte , I As Byte , W(4)
 DimAs Word , Pass As Word
Dim Code As Word , Error As Byte
dimB As Byte
Code = 7421
Do
L1:
Cls : Locate 1 , 6 : Lcd "kbd off"
Locate 2 , 5 : Lcd "pers ON/C"
B = Getkbd()
If B = 16 Then Goto L1
B = Lookup(b , Dat1)
If B <> 10 Then Goto L1
Do
L4:
```

```
Cls : Home : Cursor Off
Lcd "enter pass:" : Locate 2 , 7
For I = 1 To 4
L2:
A(i) = Getkbd()
If A(i) > 15 Then Goto L2
A(i) = Lookup(a(i) , Dat1)
If A(i) = 10 Then Goto L1
If A(i) > 10 Then
Cls : Locate 1 , 6 : Lcd "error"
Locate 2 , 1 : Lcd "plese enter 0~9"
Set Portc.1 : Wait 3 : Reset Portc.1
Goto L4
End If
Lcd "*"
W(1) = A(1) * 1000
W(2) = A(2) * 100
W(3) = A(3) * 10
W(4) = W(1) + W(2)
W(4) = W(4) + W(3)
Pass = W(4) + A(4)
Next
If Pass = Code Then
Cls : Locate 1 , 3 : Lcd "Correct pass"
Set Portc.0
Reset Portc.1
Wait 3
Reset Portc.0
Error = 0
Else
Cls : Locate 1 , 3 : Lcd "Invalid pass"
 Locate 2 , 3 : Lcd "ENTER AGAIN"
Set Portc.1
Reset Portc.0
Wait 3
Reset Portc.1
Incr Error
End If
If Error = 3 Then
Cls : Locate 1 , 7 : Lcd "Holt"
Locate 2 , 4 : Lcd "plese rerun"
Do
Loop
End If
Loop
End
Loop
End
Dat1:
Data 7 , 8 , 9 , 11 , 4 , 5 , 6 , 12 , 1 , 2
 , 3 , 13 , 10 , 0 , 14 , 15
```

مثال : برنامه ساعت وتقویم را با استفاده از دستور wait بنویسید.(نمایش برروی lcd وشروع تاریخ آن از91/1/1 باشد ساعت شروع ،ساعت 9 باشد)




```
$regfile = "m16def.dat"
$crystal = 8000000

Config Lcdpin = Pin , Db4 = Pd.4 ,

 Db5 = Pd.5 , Db6 = Pd.6 , Db7 =

 Pd.7 , Rs = Pd.1 , E = Pd.2

Dim S As Byte , M As Byte

DimH As Byte , D As Byte

Dim Mo As Byte , Y As Byte

D = 1 : Mo = 1 : Y = 91 : H = 9

Do

Wait 1:Incr S

If S > 59 Then
S = 0 : Incr M
End If
If M > 59 Then
M = 0 : Incr H
End If
If H > 23 Then
H = 0 : Incr D
End If
Select Case Mo

Case 1 To 6:

If D > 31 Then

D = 1 : Incr Mo

End If

Case 7 To 11:

If D > 30 Then

D = 1 : Incr Mo

End If

Case Else

If D > 29 Then

D = 1 : Mo = 1 : Incr Y

End If
End Select
Cls : Home : Lcd H ; ":" ; M ; ":" ; S
Locate 2 , 1 : Cursor Off
 Lcd Y ; ":" ; Mo ; ":" ; D
Loop
End
```

پیکربندی تایمرها :

تایمرها به دو دسته 8 بیتی (0~255) که شامل تایمر0 وتایمر 2 می شود و16بیتی (0~65535) که شامل تایمر1 تقسیم بندی می شوند.

وقتی مقدار تایمر به مقدار نهایی خود می رسد پرچمی به نام پرچم سرریزتایمر(ovf) یک می شود.و با فرا خوانی زیر برنامه وقفه می توان وقفه را اجرا نمود.مدت زمانی که طول میکشد که تایمر به مقدار نهایی اش برسد به 2عامل بستگی دارد 1- مقدار اولیه تایمر 2- فرکانس تایمر

مقدار اولیه یک تایمر را می توان با استفاده از دستور timer x=val تغییر داد.

فرکانس تایمر می تواند تقسیم شده فرکانس کریستال میکرو یا از یک منبع فرکانس خارجی گرفته شود.معمولاً در بیشتر موارد فرکانس تایمر تقسیم شده فرکانس کریستال میکرو است .

$$f_t = \frac{f_{crystal}}{prescale}$$

Prescale می تواند 1,8,64,256,1024باشد.

عکس فرکانس تایمر مدت زمانی است که طول می کشد تا یک واحد به محتوی رجیستر تایمر اضافه شود.

از این رو مدت زمانی که طول می کشد تا تایمر به مقدار نهایی خود برسد از رابطه زیر محاسبه می شود

$$T_{ovf}=\frac{\text{مقدار اولیه}-\text{مقدار نهایی}}{\text{فرکانس تایمر}}$$

بعنوان مثال فرض کنیم فرکانس کریستال 8 مگاهرتز و prescale=8 و مقدار اولیه تایمر صفربرابر 5 با شد در این صورت: $f_t=\frac{8M}{8}$=1M یعنی هر1μثانیه یک واحد به رجیستر اضافه می کند.

$T_{ovf}=\frac{255-5}{1M}=250\mu s$ 250µ ثانیه طول می کشدتا پرچم سرریزتایمر صفر یک شود.

با دستور start timer x تایمر شروع به شمارش می کند و با دستور stop timer تایمرمتوقف می شود

نکته :در ابتدای برنامه حتماً باید از دستورstart timer استفاده شود،در غیر این صورت شروع به شمارش نمی کند.

برای استفاده از وقفه تایمرها :

1- فراخوانی کل وقفه ها با دستور enable interrupts

2- فراخوانی وقفه مربوطه enable timer x یا enable ovfx

3- آدرس زیر برنامه on ovfx label

بعد از اینکه اجرا برنامه به زیر برنامه وقفه انتقال یابد بصورت خودکار ovfx صفر خواهد شد.

تمام زیر برنامه های وقفه بایستی بعد از دستور end نوشته شوند ودر پایان آنها باید دستورreturn نوشته شود.

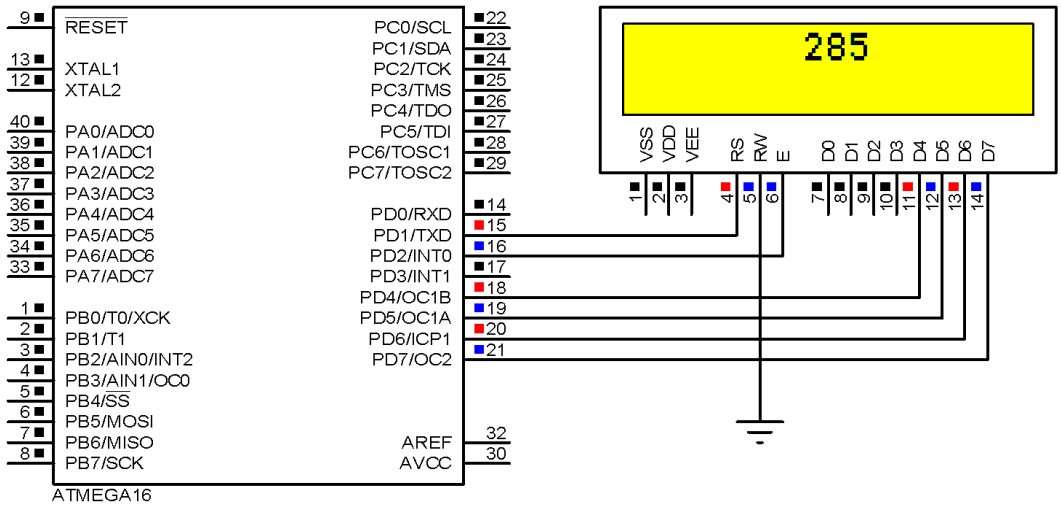
نکته:باید سعی شود که برنامه های وقفه کوتاه باشند تا مدت زمان اجرای کمتری داشته باشند زیرا اگر مدت زیادی داشته باشند در صورت یک شدن دوباره ovfx دیگر تایمر وقفه را اجرا نخواهد کرد.

دستور debounce : debounce pinx.y,1/0 ,label,sub

پیکربندی تایمر0 در محیط بسکام :

```
Config  timer0=timer,prescale=1/8/64/256/1024
```

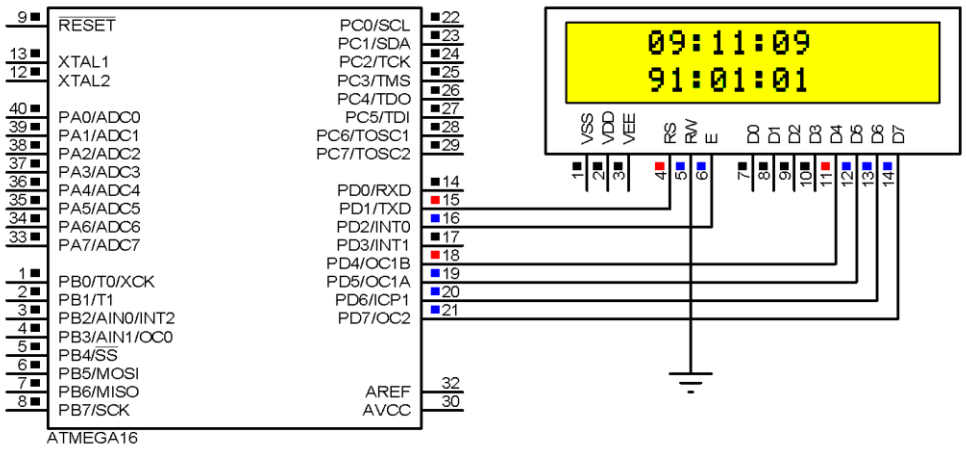
مثال : برنامه ای بنویسید که هریک ثانیه یک واحد به متغییرA اضافه کند و بروی lcdنمایش دهد.




```
$regfile = "m16def.dat"
$crystal = 8000000
Config Lcdpin = Pin , Db4 = Pd.4 , Db5 = Pd.5 , Db6 = Pd.6 , Db7 = Pd.7
, Rs = Pd.1 , E = Pd.2
Config Lcd = 16 * 2
```

```
Config Timer0 = Timer , Prescale = 8
Dim A As Word , B As Word
Timer0 = 5
Start Timer0
Enable Interrupts
Enable Ovf0
On Ovf0 Label
Do
Loop
End
Label:
Incr A
If A = 4000 Then
Incr B
A = 0
Cls : Cursor Off : Locate 1 , 8 : Lcd B
End If
Return
```

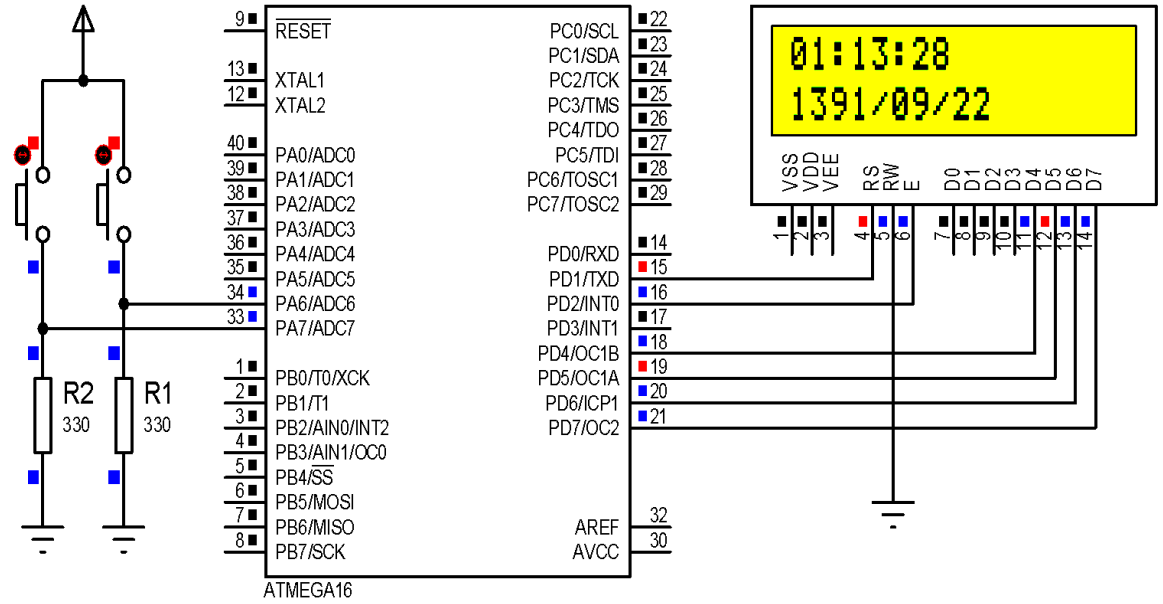
مثال : برنامه ساعت با استفاده از تایمر صفر به همراه تقویم شمسی را بنویسید.




```
$regfile = "m16def.dat"
$crystal = 8000000
Config Lcdpin = Pin , Db4 = Pd.4 ,
Db5 = Pd.5 , Db6 = Pd.6 ,
Db7 = Pd.7 , Rs = Pd.1 , E = Pd.2
Config Lcd = 16 * 2
Config Timer0 = Timer , Prescale = 8
Dim A As Word , I As Word
Dim N(6) As Byte , S(6) As String * 2
Timer0 = 5
Start Timer0
Enable Interrupts
Enable Ovf0
On Ovf0 Label
N(3) = 9 : N(4) = 1
N(5) = 1 : N(6) = 91
```

```
Do
If N(1) > 59 Then
N(1) = 0 : Incr N(2)
End If
If N(2) > 59 Then
N(2) = 0 : Incr N(3)
End If
If N(3) > 23 Then
N(3) = 0 : Incr N(4)
End If
Select Case N(5)
Case 1 To 6:
If N(4) > 31 Then
N(4) = 1 : Incr N(5)
end If
Case 7 To 11:
If N(4) > 30 Then
N(4) = 1 : Incr N(5)
End If
Case Else
If N(4) > 29 Then
N(4) = 1 : N(5) = 1 : Incr N(6)
End If
End Select
For I = 1 To 6
If N(i) < 10 Then
S(i) = "0" + Str(n(i))
Else
S(i) = Str(n(i))
End If
Next
Loop
End
Label:
Incr A
If A = 4000 Then
Incr N(1)
A = 0
Cls : Cursor Off : Locate 1 , 4
Lcd S(3) ; ":" ; S(2) ; ":" ; S(1)
Locate 2 , 4 : Cursor Off
Lcd S(6) ; ":" ; S(5) ; ":" ; S(4)
End If
Return
```

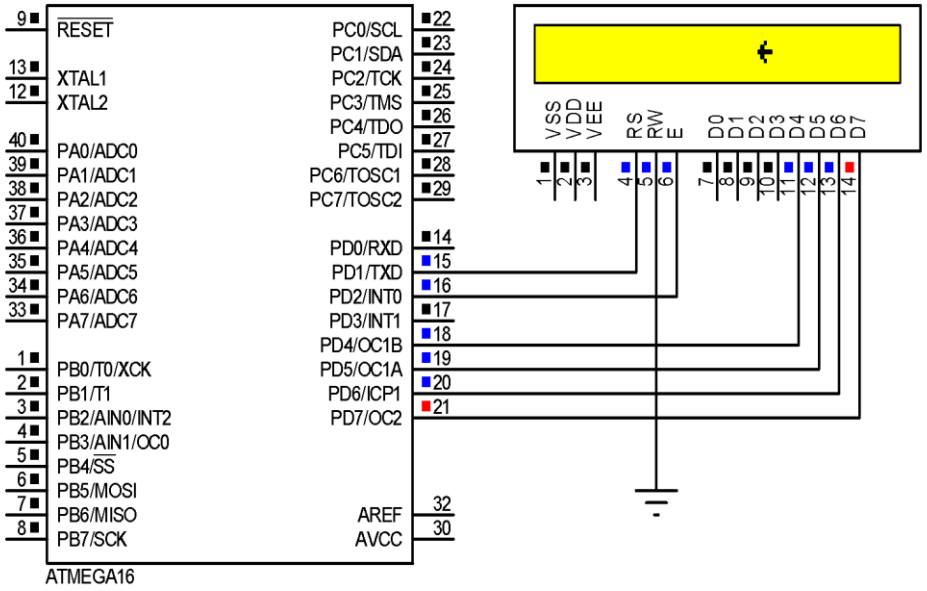
مثال : برنامه ساعت با استفاده از تایمر صفر به همراه تقویم شمسی با قابلیت تنظیم توسط دو کلید فشاری را بنویسید.




```
$regfile = "m16def.dat"
$crystal = 8000000
Config Lcdpin = Pin , Db4 = Pd.4 ,
Db5 = Pd.5 , Db6 = Pd.6 , Db7 = Pd.7
 , Rs = Pd.1 , E = Pd.2
Config Lcd = 16 * 2
Config Timer0 = Timer , Prescale = 8
Config Pina.6 = Input
Config Pina.7 = Input
Config Pina.5 = Input
Dim A As Word , I As Word
Dim N(7) As Byte , S(7) As String * 2
Dim D As Byte
Timer0 = 5
Start Timer0
Enable Interrupts
Enable Ovf0
On Ovf0 Label
N(3) = 9 : N(4) = 1 : N(5) = 1
 N(6) = 91 : N(7) = 13
Do
Debounce Pina.6 , 1 , L1 , Sub
Debounce Pina.7 , 1 , L2 , Sub
If N(1) > 59 Then
N(1) = 0 : Incr N(2)
End If
If N(2) > 59 Then
N(2) = 0 : Incr N(3)
End If
If N(3) > 23 Then
N(3) = 0 : Incr N(4)
End If
Select Case N(5)
Case 1 To 6:
```

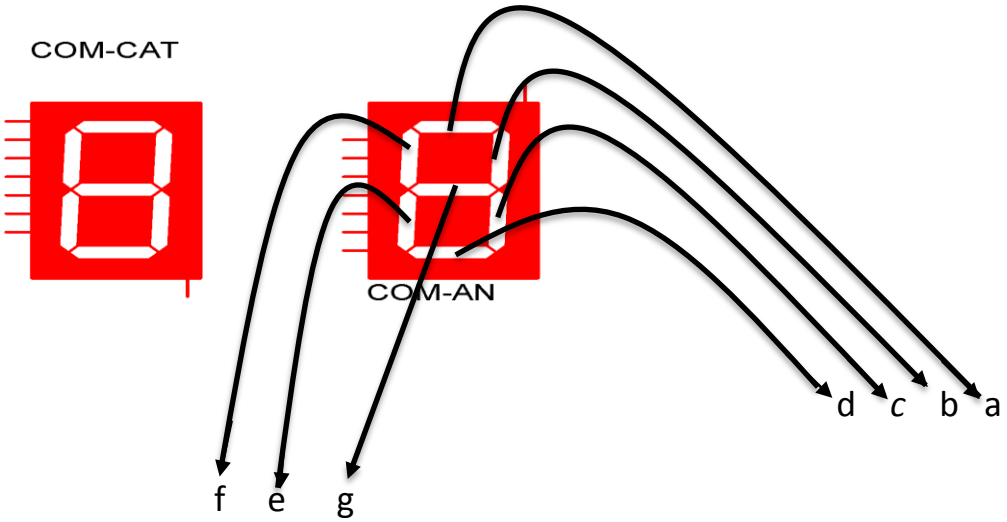
```
If N(4) > 31 Then
N(4) = 1 : Incr N(5)
End If
Case 7 To 11:
If N(4) > 30 Then
N(4) = 1 : Incr N(5)
End If
Case 12:
If N(4) > 29 Then
N(4) = 1 : N(5) = 1 : Incr N(6)
End If
Case Else
If N(5) > 12 Then
N(4) = 1 : N(5) = 1 : Incr N(6)
End If
End Select
If N(6) > 99 Then
N(6) = 0 : Incr N(7)
End If
For I = 1 To 7
If N(i) < 10 Then
S(i) = "0" + Str(n(i))
Else
S(i) = Str(n(i))
End If
Next
Loop
End
Label:
Incr A
If A = 3080 Then
Incr N(1)
A = 0
Cls : Cursor Off : Locate 1 , 1
 Lcd S(3) ; ":" ; S(2) ; ":" ; S(1)
Locate 2 , 1 : Cursor Off
 Lcd S(7) ; S(6) ; "/" ; S(5) ; "/" ; S(4)
End If
Return
L1:
Incr D
Locate 2 , 13:
Select Case D
Case 0 : Lcd "   "
Case 1 : Lcd "sec"
Case 2 : Lcd "min"
Case 3 : Lcd "hor"
Case 4 : Lcd "day"
Case 5 : Lcd "mon"
Case 6 : Lcd "yer1"
Case 7 : Lcd "yer2"
End Select
If D > 7 Then D = 0
Return
L2:
Incr N(d)
Return
```

مثال : برنامه ای بنویسید که یک کاراکتر در lcd نمایش دهد(←) و به سمت راست رود وقتی به انتهای صفحه نمایش رسید،جهت فلش تغییر کند وبه سمت چپ حرکت کند.




```
$regfile = "m16def.dat"
$crystal = 8000000
Config Lcdpin = Pin , Db4 = Pd.4 ,
 Db5 = Pd.5 , Db6 = Pd.6 ,
 Db7 = Pd.7 , Rs = Pd.1 , E = Pd.2
Config Lcd = 16 * 1
Dim A As Byte
Do
Cls : Home : Cursor Off:Lcd Chr(126)
For A = 1 To 15
Shiftlcd Right
Wait 1
Next
Cls
Locate 1 , 16
Lcd Chr(127)
For A = 1 To 15
Shiftlcd Left
Wait 1
Next
Loop
End
```

کار با 7segment



7segment دونوع است :

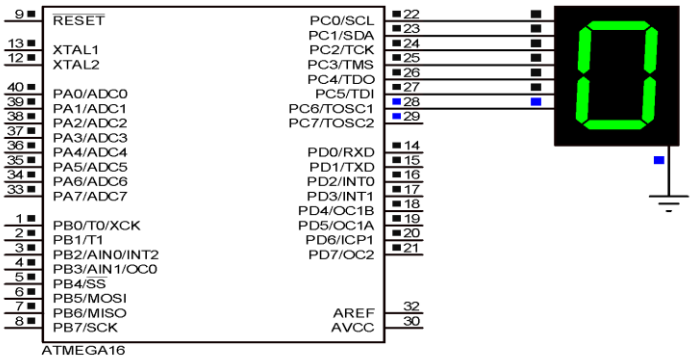
1- آند مشترک : تمامی آندها به یکدیگر متصل شده وکاتد ها به صورت ورودی در نظرگرفته شده است.

2- کاتدمشترک : تمامی کاتدها به هم وصل شده اند و آندها بعنوان ورودی هستند.

آند مشترک : برای فعال کردن 7segment آند مشترک باید پایه مشترک یک شود و برای روشن کردن هر ledباید پایه مربوط به آن صفر شود.

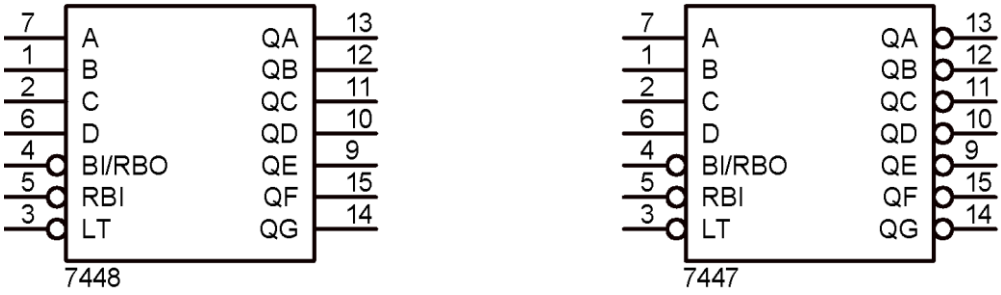
کاتد مشترک : برای فعال کردن 7segment کاتد مشترک باید پایه مشترک صفرشود وبرای روشن کردن هر led بایستی پایه مربوط به آن یک شود.

مثال: برنامه ای بنویسید که عدد 0 را برروی 7seg کاتدمشترک نمایش دهد.



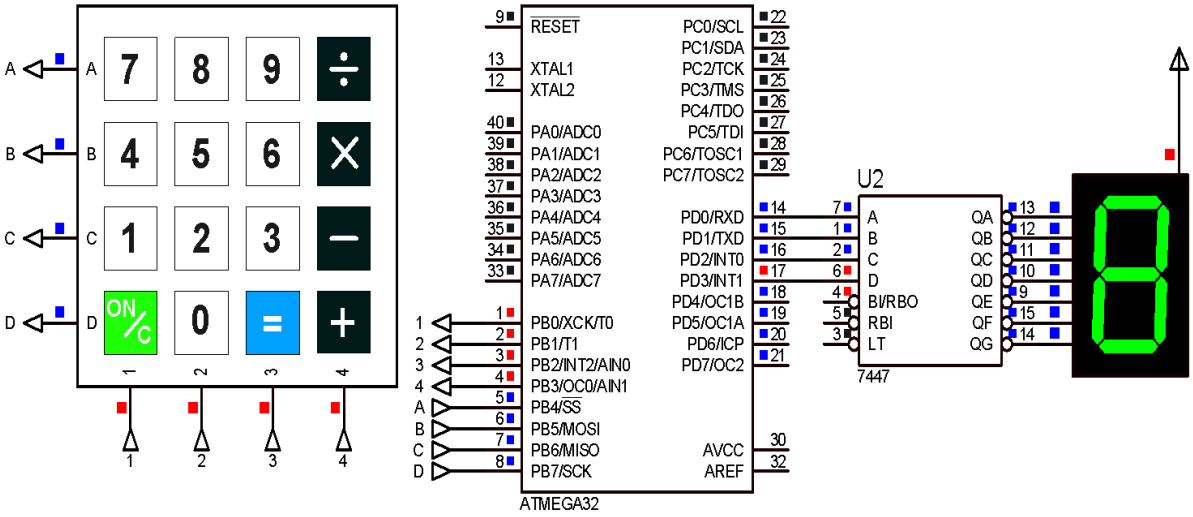

```
$regfile = "m16def.dat"
$crystal = 8000000
Config Portc = Output
Do
Portc = &B00111111
Loop
```

معرفی ICهای 7447 و7448 :



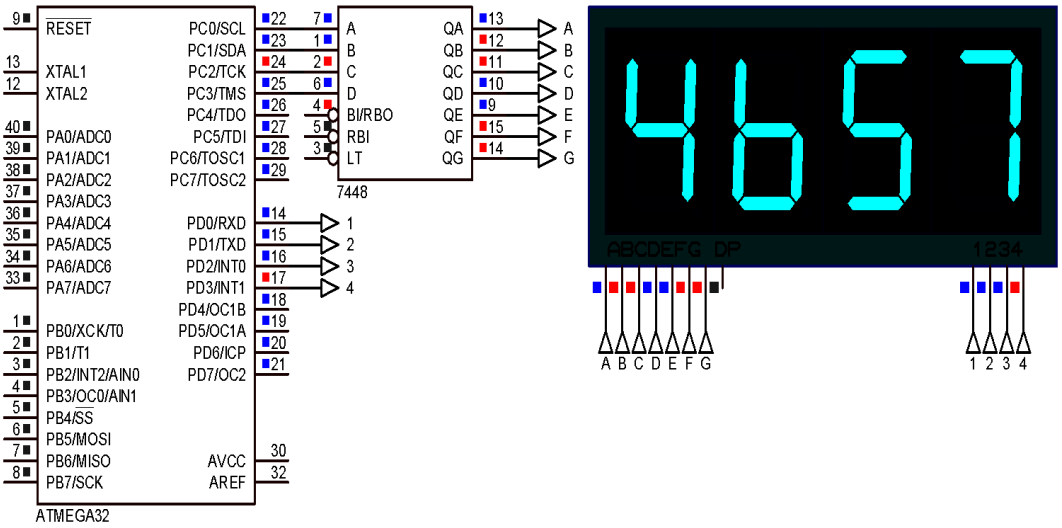
7447برای اتصال به 7seg آند مشترک و7448برای اتصال 7segکاتد مشترک به کار می رود.

مثال : برنامه ای بنویسید که یک عدد تک رقمی فشرده شده از صفحه کلید گرفته و برروی 7segment نمایش دهد.



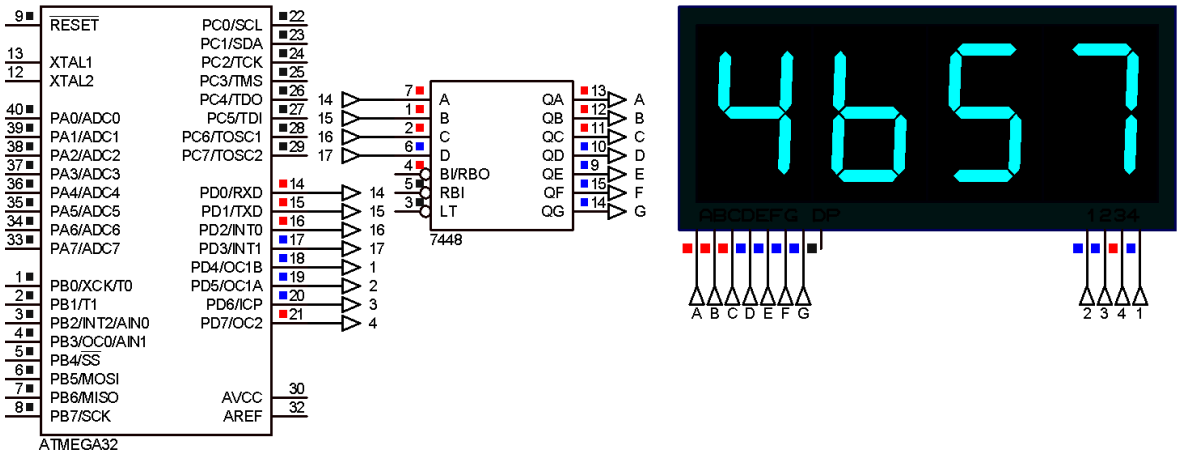

```
$regfile = "m32def.dat"
$crystal = 8000000
Config Portd = Output
Config Kbd = Portb , Debounce = 50
 , Delay = 100
Dim A As Byte
Do
L1:
A = Getkbd()
If A = 16 Then Goto L1
A = Lookup(a , Dat1)
If A > 9 Then Goto L1
Portd = A
Loop
End
Dat1:
Data 7 , 8 , 9 , 11 , 4 , 5 , 6 , 12 , 1 , 2
 , 3 , 13 , 10 , 0 , 14 , 15
```

مثال : برنامه ای بنویسید که عدد 4657 را برروی یک 7seg چهارتایی آندمشترک نمایش دهد.




```
$regfile = "m32def.dat"
$crystal = 8000000
Config Portd = Output
Config Portc = Output
Do
Portc = 4
Waitms 10
Portd = 1
Portc = 6
Waitms 10
Portd = 2
Portc = 5
Waitms 10
Portd = 4
Portc = 7
Waitms 10
Portd = 8
Loop
End
```

مثال : برنامه مثال قبل را با استفاده از یک پورت بنویسید.



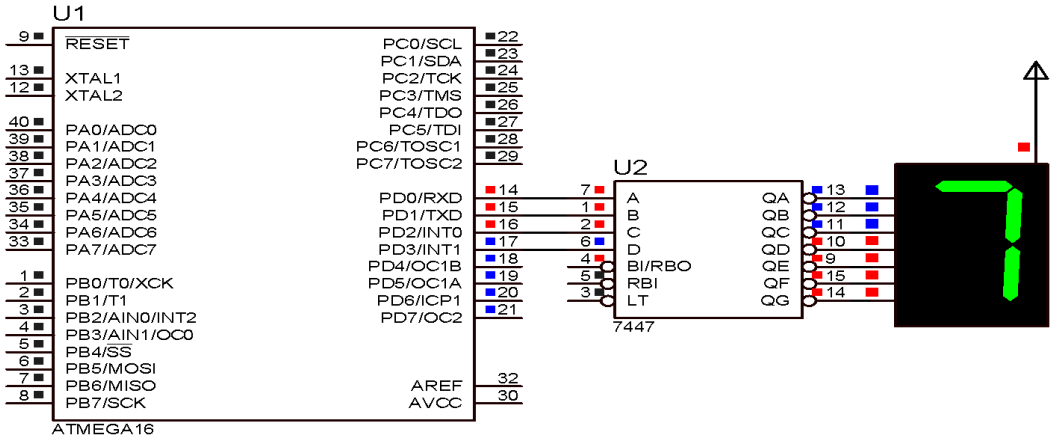

```
$regfile = "m32def.dat"
$crystal = 8000000
Config Portd = Output
Dim A As Byte
Do
A = 4 Or 16 : Portd = A
Waitms 10
A = 6 Or 32 : Portd = A
Waitms 10
A = 5 Or 64 : Portd = A
Waitms 10
A = 7 Or 128 : Portd = A
Waitms 10
Loop
End
```

مثال : برنامه مثال قبل را با استفاده از مبحث تایمرها انجام دهید.

```
$regfile = "m32def.dat"
$crystal = 8000000
Config Portd = Output
Config Timer0 = Timer , Prescale = 8
Dim A As Byte , B As Byte
Dim Count As Byte
Enable Interrupts
Enable Ovf0
On Ovf0 Label
Timer0 = 5
Start Timer0
Do
Select Case Count
Case 1:
A = 4 Or 16
Case 2:
A = 6 Or 32
Case 3:
A = 5 Or 64
Case 4
```

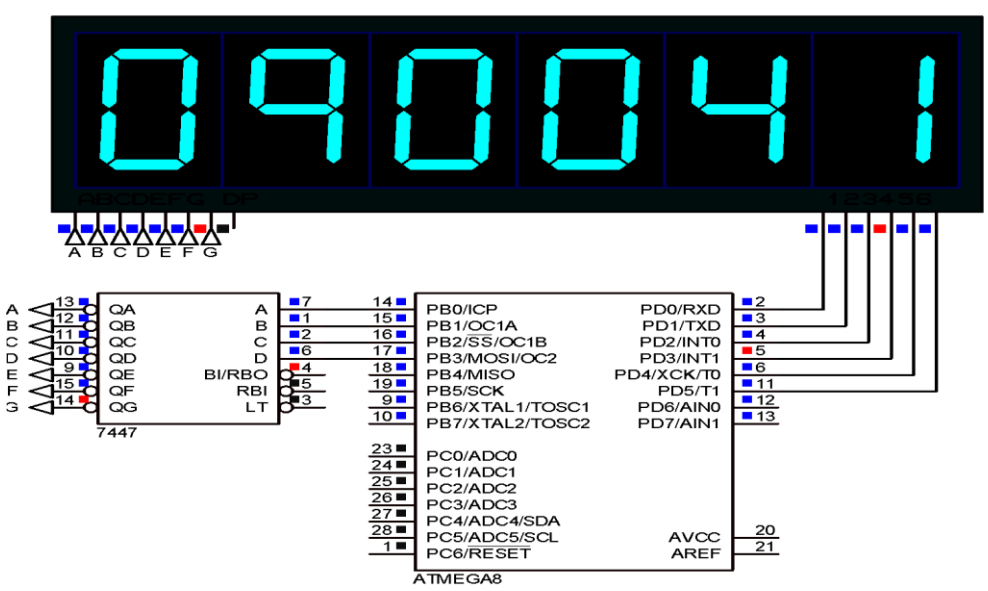
```
A = 7 Or 128
End Select
Portd = A
Loop
End
Label:
Incr B
If B = 30 Then
Incr Count
B = 0
End If
If Count > 4 Then Count = 0
Return
```

مثال : برنامه ای بنویسید که اعداد 0~9 را با فاصله 1 ثانیه نمایش دهد.




```
$regfile = "m16def.dat"
$crystal = 8000000
Config Portd = Output
Dim A As Byte
Do
A = Lookup(a , Dat1)
Portd = A
Incr A
Wait 1
If A > 9 Then A = 0
Loop
End
Dat1:
Data 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9
```

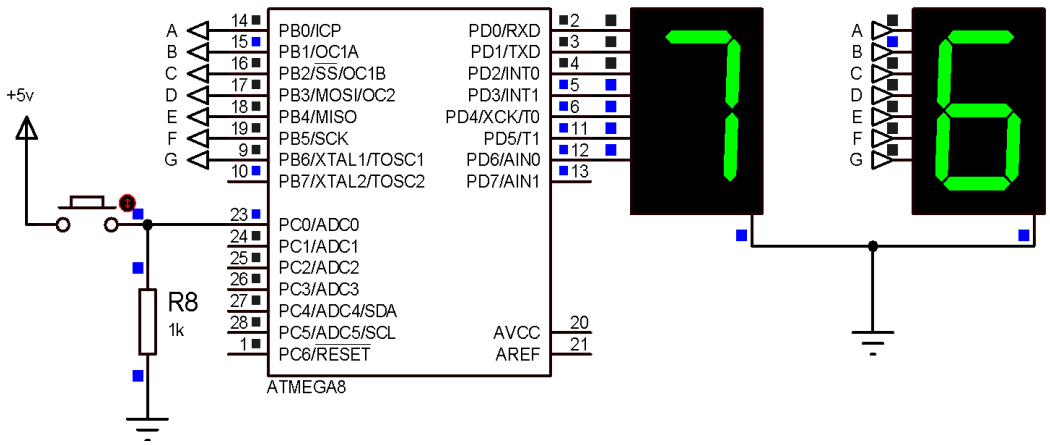
مثال : برنامه ساعت را با استفاده ازنمایشگر7seg طراحی نمایید.




```
$regfile = "m8def.dat"
$crystal = 8000000
Config Portb = Output
Config Portd = Output
Config Timer0 = Timer , Prescale = 8
Timer0 = 5
Enable Interrupts
Enable Timer0
On Ovf0 Lable
Start Timer0
Dim A As Byte , B As Word
DimS As Byte , H As Byte , M As Byte
DimC As Byte , Count As Byte
H = 9
Do
If S > 59 Then
S = 0 : Incr M
End If
If M > 59 Then
M = 0 : Incr H
End If
If H > 23 Then H = 0
Loop
End
Lable:
Incr B
If B = 4000 Then
B = 0 : Incr S
End If
Incr C
If C = 32 Then
Incr Count
If Count > 6 Then Count = 1
C = 0
```

```
Portd = 0
Select Case Count
Case 1:
Portb = H / 10
Portd = 1
Case 2:
Portb = H Mod 10
Portd = 2
Case 3:
Portb = M / 10
Portd = 4
Case 4:
Portb = M Mod 10
Portd = 8
Case 5:
Portb = S / 10
Portd = 16
Case 6:
Portb = S Mod 10
Portd = 32
End Select
End If
Return
```

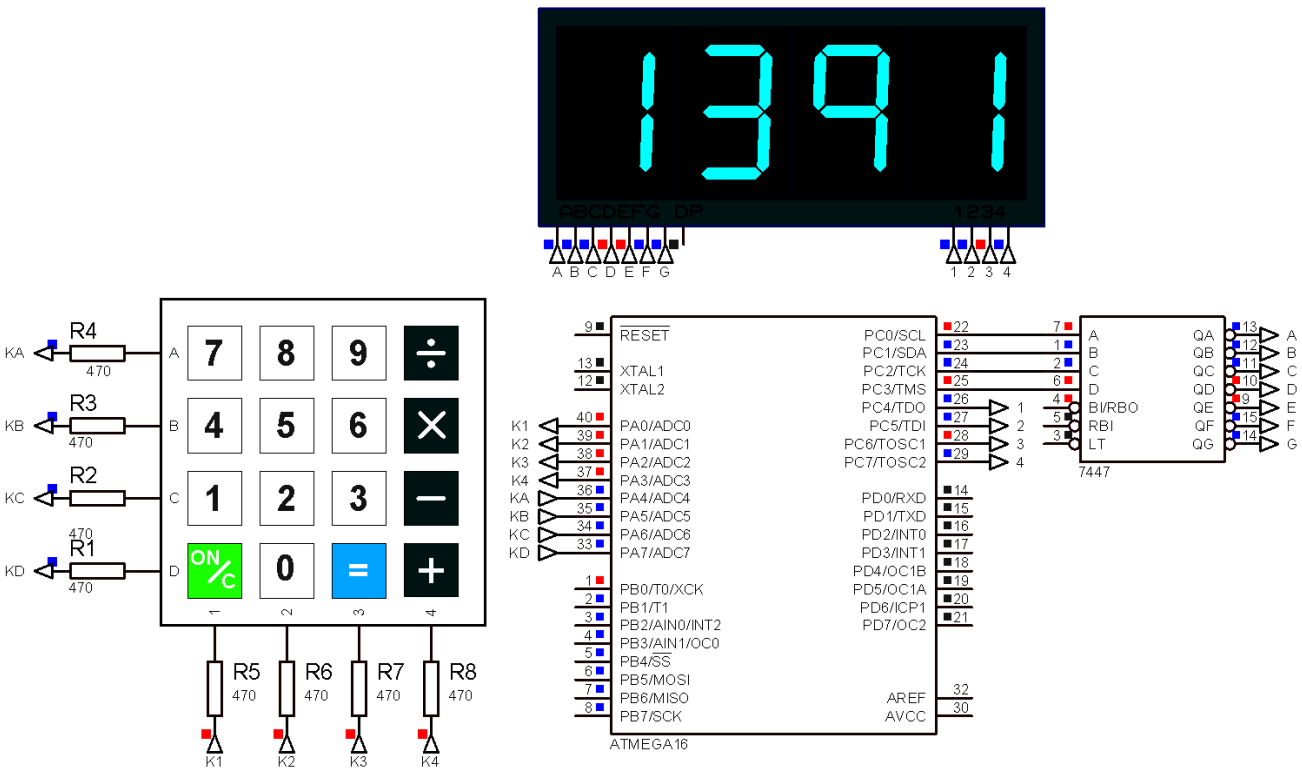
مثال : برنامه ای بنویسید که با هر بارفشار دادن کلید،یک واحد به متغییر اضافه کند وبروی دوعدد 7seg نمایش دهد.




```
$regfile = "m8def.dat"
$crystal = 8000000
Config Pinc.0 = Input
Config Portb = Output
Config Portd = Output
Dim B As Byte , K As Byte , A As Byte
DimW As Byte
K = Lookup(b , Table)
W = Lookup(a , Table)
Portb = K
Portd = W
Do
Debounce Pinc.0 , 1 , L1 , Sub
Loop
```

```
End
L1:
Incr B
If B > 9 Then
B = 0
Incr A
End If
If A > 9 And B = 0 Then
A = 0
End If
K = Lookup(b , Table)
W = Lookup(a , Table)
Portb = K
Portd = W
Return
Table:
Data &B0111111 , &B0000110 , &B1011011 , &B1001111 , &B1100110 , &B1101101 , &B1111101 , &B0000111 , &B1111111 , &B1101111
```

مثال : برنامه ای بنویسید که یک عدد چهار رقمی را از صفحه کلید بگیرد وبرروی 7seg نمایش دهد.با زدن کلید enter تمامی 7segها خاموش شود ومنتظرعدد بعدی باشد.

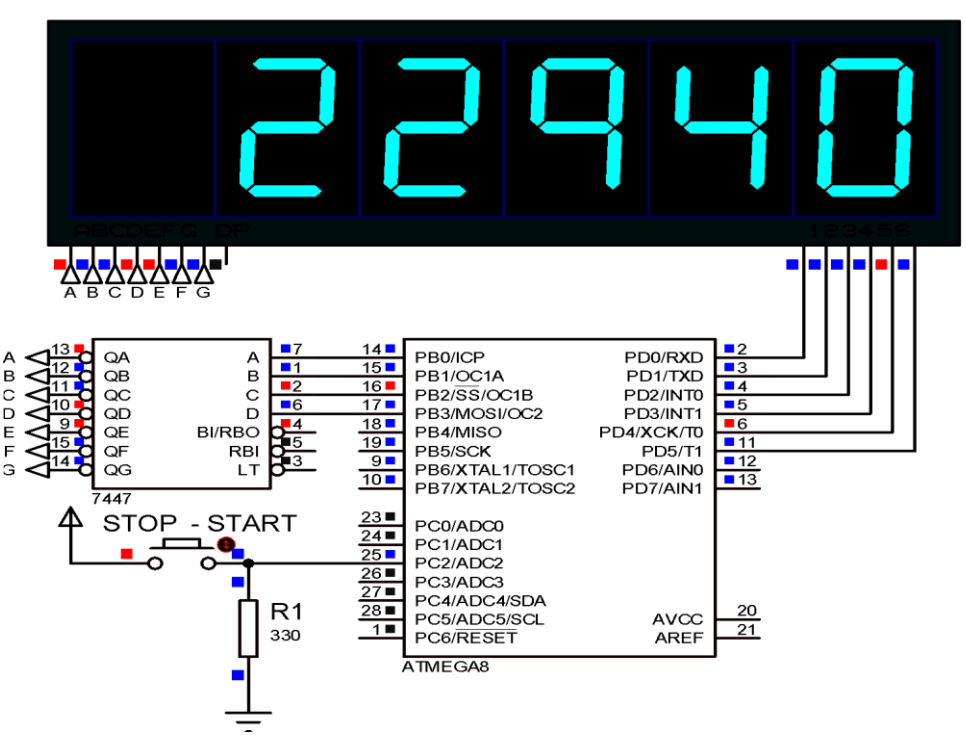



```
$regfile = "m16def.dat"
$crystal = 8000000
Config Portc = Output
Config Kbd = Porta , Debounce = 50
, Delay = 100
Dim A(4) As Byte , I As Byte
Dim C(3) As Word , Count As Byte
Dim B As Byte , D As Byte
```

```
Dim Bb As Byte
Config Timer0 = Timer , Prescale = 1
Timer0 = 5
Enable Interrupts
Enable Timer0
On Ovf0 Label
Start Timer0
Do
For I = 1 To 4
L1:
A(i) = Getkbd()
If A(i) > 15 Then Goto L1
A(i) = Lookup(a(i) , Dat1)
If A(i) > 9 Then Goto L1
Next
L2:
Bb = Getkbd()
If Bb > 15 Then Goto L2
Bb = Lookup(bb , Dat1)
If Bb = 15 Then
For I = 1 To 4
A(i) = 0
Next
End If
Loop
End
Label:
Incr B
If B = 20 Then
Incr Count
B = 0
If Count > 4 Then Count = 1
End If
Select Case Count
Case 1:
Portc = A(1) Or 16
Case 2:
Portc = A(2) Or 32
Case 3:
Portc = A(3) Or 64
Case 4:
Portc = A(4) Or 128
End Select
Return
Dat1:
Data 7 , 8 , 9 , 11 , 4 , 5 , 6 , 12 , 1 , 2
 , 3 , 13 , 10 , 0 , 14 , 15
```

مثال : با استفاده از تایمر 1برنامه کرنومتری بنویسید که با زدن کلید شروع به شمارش کند وبا زدن دوباره آن STOP شود.(صفرهای بی ارزش نشان داده نشوند)

این کرنومترصدم ثانیه،ثانیه و دقیقه را نمایش می دهد.




```
$regfile = "m8def.dat"
$crystal = 8000000
Config Portb = Output
Config Portd = Output
Config Timer1 = Timer , Prescale = 1
Config Timer0 = Timer , Prescale = 8
Config Pinc.2 = Input
Timer0 = 5
Timer1 = 55535
Enable Interrupts
Enable Timer1
Enable Timer0
On Ovf1 L1
On Ovf0 L2
Stop Timer1
Start Timer0
Dim A As Word , B As Byte
Dim M(6) As Byte , C As Byte
Dim Count As Byte , I As Byte
Dim D(6) As Long
Do
Debounce Pinc.2 , 1 , L3 , Sub
If M(1) > 9 Then
M(1) = 0 : Incr M(2)
End If
If M(2) > 9 Then
```

```
M(2) = 0 : Incr M(3)
End If
If M(3) > 9 Then
M(3) = 0 : Incr M(4)
End If
If M(4) > 5 Then
M(4) = 0 : Incr M(5)
End If
If M(5) > 9 Then
M(5) = 0 : Incr M(6)
End If
If M(6) > 5 Then M(6) = 0
D(6) = M(6) * 100000
D(5) = M(5) * 10000
D(4) = M(4) * 1000
D(3) = M(3) * 100
D(2) = M(2) * 10
D(1) = M(1)
D(6) = D(6) + D(5)
D(6) = D(6) + D(4)
D(6) = D(6) + D(3)
D(6) = D(6) + D(2)
D(6) = D(6) + D(1)
Select Case D(6)
Case 0 To 9:
If Count > 1 Then Count = 1
Case 10 To 99:
If Count > 2 Then Count = 1
Case 100 To 999:
If Count > 3 Then Count = 1
Case 1000 To 9999:
If Count > 4 Then Count = 1
Case 10000 To 99999:
If Count > 5 Then Count = 1
Case 100000 To 999999:
If Count > 6 Then Count = 1
End Select
Select Case Count
Case 1:
Portb = M(1):Portd = 32
Case 2:
Portb = M(2):Portd = 16
Case 3:
Portb = M(3):Portd = 8
Case 4:
Portb = M(4):Portd = 4
Case 5:
Portb = M(5):Portd = 2
Case 6:
Portb = M(6)
Portd = 1
End Select
Loop
End
```

```
L1:
Incr M(1)
Return
L2:
Incr B
If B = 30 Then
Incr Count
B = 0 : Portd = 0
End If
Return
L3:
Incr C
If C > 2 Then C = 1
Select Case C
Case 1:
For I = 1 To 6
M(i) = 0
Next
Start Timer1
Case 2:
Stop Timer1
End Select
Return
```

پیکربندی تایمر0بعنوان کانتر:

```
Config  timer0=conter , edge=rising/falling
```

درصورتیکه پالس مورد نظر به پایه pinb.0 اعمال شود با استفاده ازپیکربندی فوق می توان تعداد لبه های بالارونده/پایین رونده شکل موج اعمال شده را شمارش نمود.

با اعمال هر لبه بالارونده/پایین رونده یک واحد به محتوای کانتر اضافه می شود.

دادن مقداراولیه: conter0=val که مقدارvalمی تواند بین 255~0 باشد.

خواندن مقدارconter : var=conter0 که متغییر var ازنوع بایت است.

نکته: طریقه استفاده ازوقفه ها درکانتر مشابه با تایمر است.

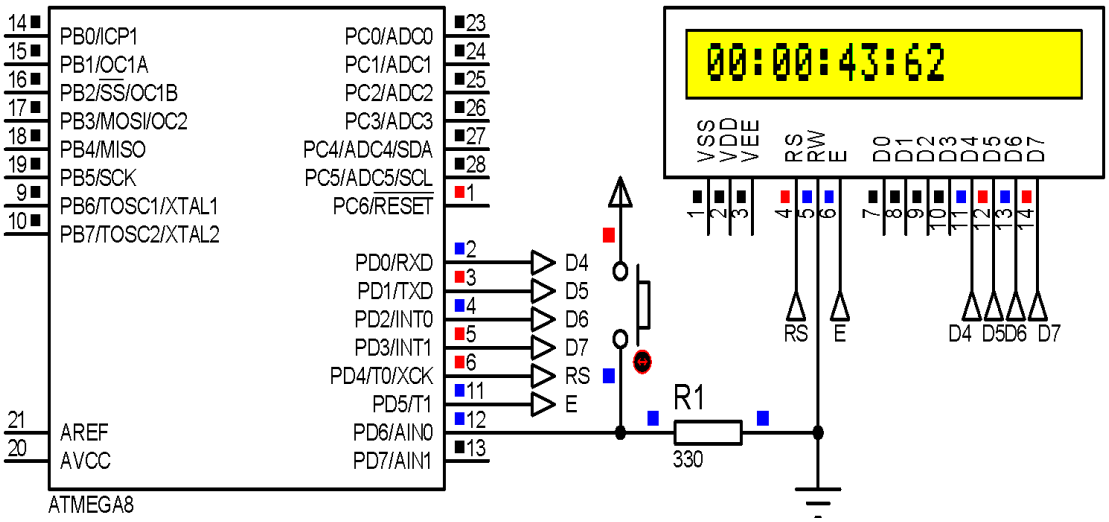
پیکربندی تایمر1بعنوان تایمر:

```
Config  timer1=timer, prescale=1/8/64/256/1024
```

دادن مقداراولیه به تایمر1 : timer1=val که مقدارval می تواند بین 65535~0 باشد.

با دستور start تایمر شروع به کار می کند وبا دستور stopمتوقف می شود.

مثال : با استفاده از تایمر1 برنامه کرنومتری را بنویسید که با زدن کلید شروع به شمارش کند و با زدن دوباره آن متوقف شود.(نمایش برروی LCD)



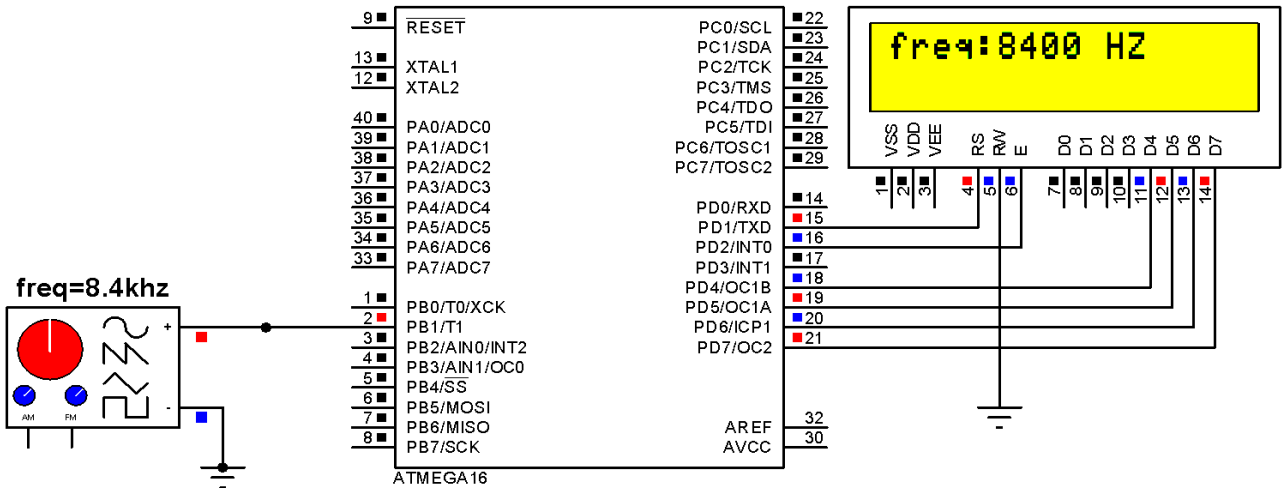

```
$regfile = "m8def.dat"
$crystal = 8000000
Config Pind.6 = Input
Config Lcdpin = Pin , Db4 = Pind.0 ,
Db5 = Pind.1 , Db6 = Pind.2 , Db7 =
Pind.3 , Rs = Pind.4 , E = Pind.5
Config Lcd = 16 * 1
Config Timer1 = Timer , Prescale = 1
Timer1 = 55535
Enable Interrupts
Enable Timer1
On Ovf1 Label
Stop Timer1
Dim T(4) As Byte , S(4) As String * 2
Dim I As Byte , Count As Byte
Do
Debounce Pind.6 , 1 , L1 , Sub
If T(1) > 99 Then
T(1) = 0 : Incr T(2)
End If
If T(2) > 59 Then
T(2) = 0 : Incr T(3)
End If
If T(3) > 59 Then
T(3) = 0 : Incr T(4)
End If
If T(4) > 99 Then T(4) = 0
For I = 1 To 4
If T(i) > 10 Then
S(i) = Str(t(i))
Else
S(i) = "0" + Str(t(i))
End If
Next
```

```
Home : Cursor Off  Lcd S(4) ; ":" ;
S(3) ; ":" ; S(2) ; ":" ; S(1) ; "   "
Loop
End
Label:
Incr T(1)
Return
L1:
Incr Count
If Count > 3 Then Count = 1
Select Case Count
Case 1:
For I = 1 To 4
T(i) = 0
Start Timer1
Next
Case 2:
Stop Timer1
End Select
Return
```

طبق تعریف فرکانس یک شکل موج مربعی برابربا تعداد لبه های بالارونده یا پایین رونده آن درمدت زمان یک ثانیه است.

ازتایمرصفربرای تولید مدت زمان 1s استفاده میشود.ازکانتر1 برای شمارش تعداد پالس ها استفاده میشود.

مثال :برنامه ای بنویسید که فرکانس یک شکل موج مربعی را نشان دهد.




```
$regfile = "m16def.dat"
$crystal = 8000000
Config Lcdpin = Pin , Db4 = Pd.4 ,
 Db5 = Pd.5 , Db6 = Pd.6 , Db7 =
 Pd.7 , Rs = Pd.1 , E = Pd.2
Config Lcd = 16 * 2
Config Timer0 = Timer , Prescale = 8
Config Timer1 = Counter , Edge =
Rising
Enable Interrupts
```

```
Enable Ovf0
Enable Ovf1
On Ovf0 L1
On Ovf1 L2
Timer0 = 5
Dim A As Word , B As Byte
DimF As Long
Start Timer0
Home : Lcd "frequency meter"
Do
Loop
End
L1:
Incr A
If A = 3906 Then
Stop Timer0 : Stop Timer1
F = B * 65536
F = F + Counter1
Cls : Cursor Off : Home
Lcd "freq:" ; F ; " HZ"
A = 0 : B = 0 : Counter1 = 0
Start Timer0 : Start Timer1
End If
Return
L2:
Incr B
Counter1 = 0
Return
```

PWM :

پیکربندی تایمر/کانتر یک درحالت مدولاسیون عرض پالس (pwm):

Config timer=pwm , pwm=8/9/10,coparea A pwm=clear up/clear down/disconnect,

compare b pwm= clear up/clear down/disconnect , prescale=1/8/64/256/1024

PWM=8 در این حالت تایمر به صورت یک شمارنده 8 بیتی است.که از 255~0 وسپس بلعکس 0~255 می شمارد.

PWM=9 در این حالت تایمربه صورت یک شمارنده 9 بیتی است.که از511~0 وسپس بلعکس 0~511 می شمارد.

PWM=10 در این حالت تایمر به صورت یک شمارنده 10 بیتی است.که از1023~0 وسپس بلعکس 0~1023

درصنعت مخابرات برای آنکه اطلاعات ازیک مکان به مکان دیگر ارسال شود از مدولاسیون استفاده میشود.

مدولاسیون سوارکردن سیگنال پیام برروی یک موج حامل می باشد.

کاربردهای pwm :

1- تولید یک منبع ولتاژ 5~0 ولت

2- تغییردور موتورDC

3- کنترل روشنایی منازل یا هتلها و...

تایمر/کانتر یک دارای دو خروجی PWM 8 و9 و10 بیتی است.در این حالت پایه های OC1A(Pd.5)
وoc1b(Pd.4)بعنوان خروجی pwm عمل می کنند.

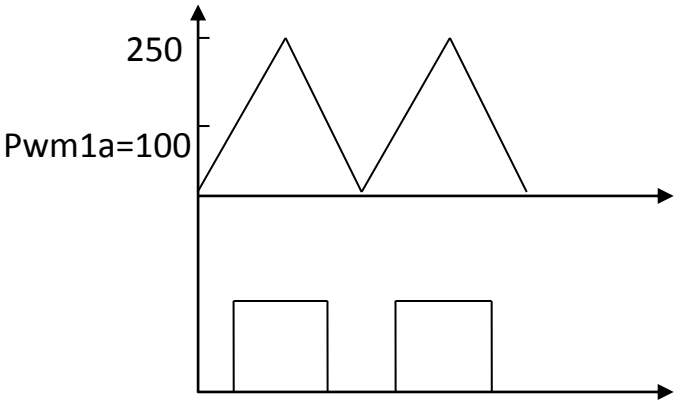
نکته:قبل از استفاده از پایه های oc1A وoc1b باید به صورت خروجی پیکربندی شوند.

دو رجیستر به نام هایpwm1A وpwm1B وجود دارد که مقدار اولیه آن ها صفراست که می توان مقدار آن را با

دستور pwm1A= value تغییر داد.

نکته : مقدار value با توجه به 8/9/10 بیتی بودن pwm حداکثربرابر است با:254/510/1022

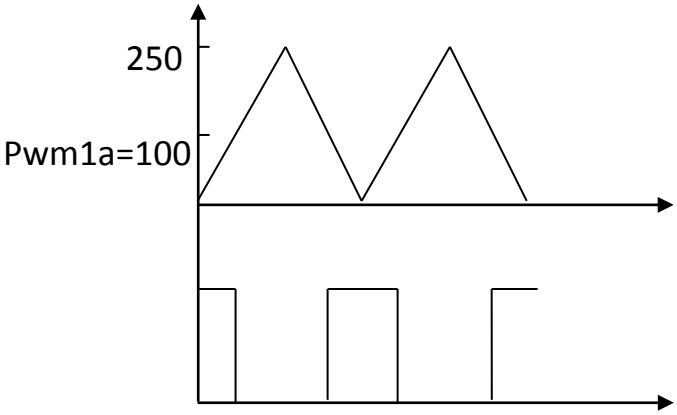
Clear up :در این حالت تا قبل از تطابق خروجی pwm صفراست و بعد ازتطابق یک می شود



نکته:دراین حالت با افزایش رجیسترمقایسه ای (pwm1b، pwm1a) D.C شکل موج pwm خروجی کاهش می یابد.

Clear down :در این حالت تا قبل ازتطابق خروجی pwm یک است و بعد ازآن صفرمی شود.

نکته: در این حالت با افزایش رجیسترمقایسه ای D.C شکل موج خروجی pwm افزایش می یابد.



Disconnect :در این حالت تا قبل از تطابق وصل است و بعد از تطابق قطع می شود.

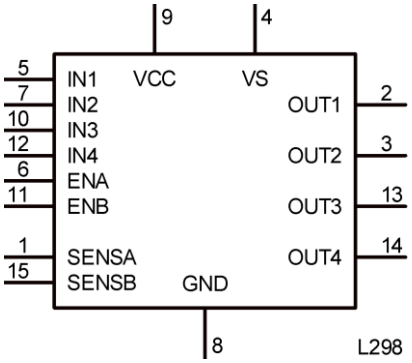
Pwm=8 $$f_{pwm} = \frac{f_{crystal}}{prescale \times 510}$$

Pwm=9 $$f_{pwm} = \frac{f_{crystal}}{prescale \times 1022}$$

Pwm=10 $$f_{pwm} = \frac{f_{crystal}}{prescale \times 2046}$$

معرفی IC L298,297 :

L298 برای موتورهای DC و297 برای استپر موتور و سرو موتورهاست



پایه های vss وvcc به ولتاژ 5~46 وصل میشود.

پایه های sensA وsensB برای محدود کردن جریان خروجی است که می توان با اتصال مقاومت کنترل کرد.

پایه هایENA وENB به خروجی های PWM میکرو وصل میشوند.

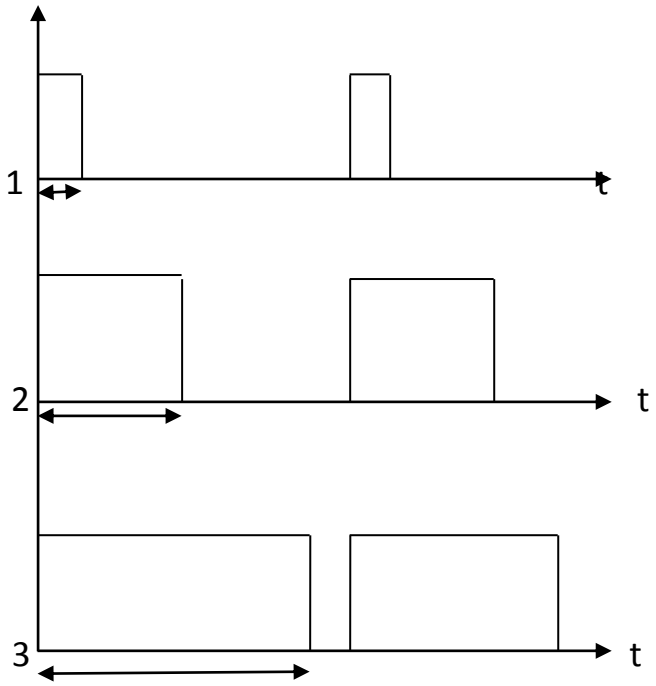
پایه های IN1 و IN2 ورودی های کنترلی پلاریته موتورشماره یک هستند.

پایه های IN3 و IN4 ورودی های کنترلی پلاریته موتورشماره دو هستند.

پایه های OUT1 و OUT2 پایه های خروجی برای موتور شماره یک هستند.

پایه های OUT3 و OUT4 پایه های خروجی برای موتورشماره دو هستند.

در مدولاسیون pwm بر اساس نوع اطلاعات پهنای پالس تغییر می کند درحالیکه دامنه ،فرکانس وفاز ثابت اند.



$$D.C = 100 \times \frac{\text{مدت زمان یک بودن}}{\text{مدت زمان کل}}$$ برحسب درصد

که در رابطه فوق D.C=0 نشان دهنده ی پالس صفر و D.C=100 نشان دهنده پالس یک است.

طبق تعریف $$vav = \frac{1}{T}\int_0^T V\ dt$$

اگرزمان یک بودن سیگنال را برابر با T_1 در نظر گرفته شود در این صورت

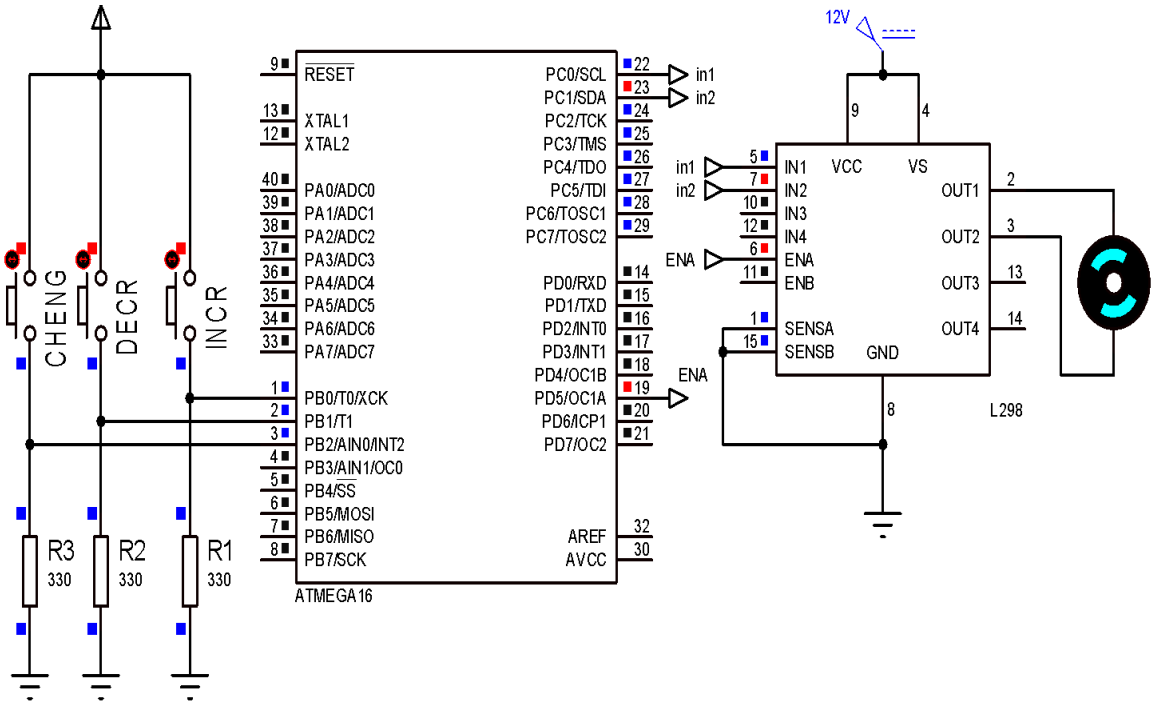
$$vav = \frac{1}{T}\int_0^{T1} VCC\ dt = \frac{1}{T} \times VCC \times t1$$

$$vav = D.C \times VCC$$

به عنوان مثال اگر: D.C=0.05 ⟶ vav=2.5^v

D.C=0.25 ⟶ Vav=1.25^v

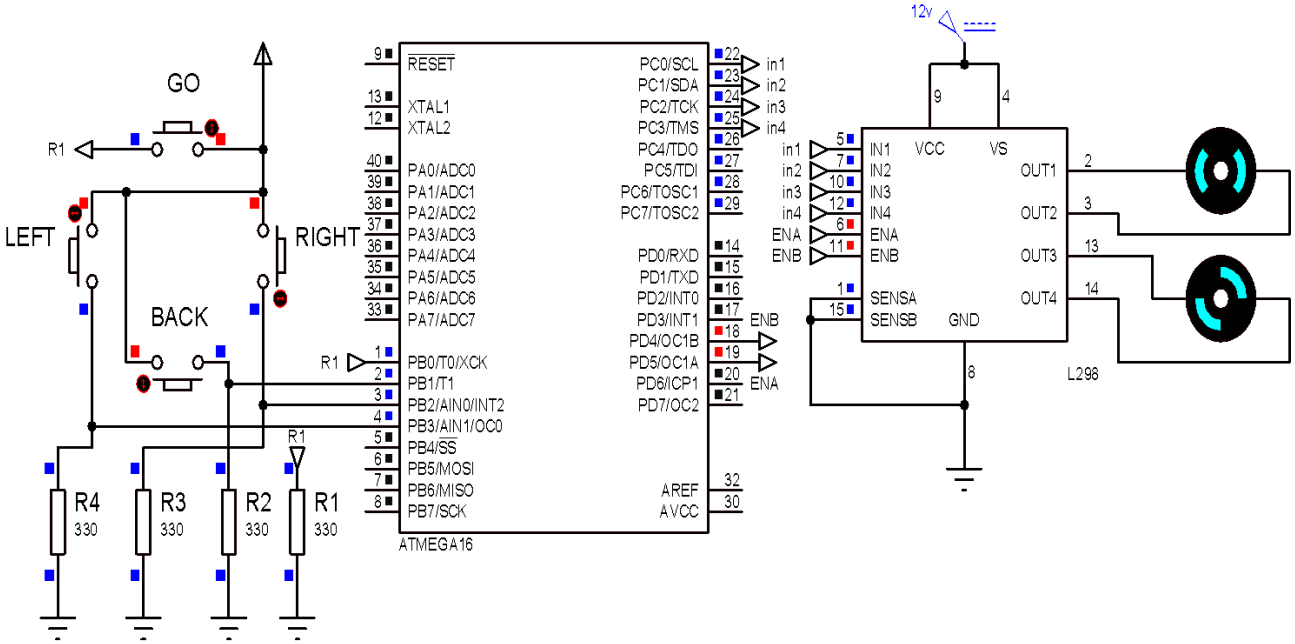
مثال : برنامه ای بنویسید که توسط دو کلید بتوان دور موتور را کم و زیاد کرد و توسط کلید دیگری دور موتور را تغییرداد.




```
$regfile = "m16def.dat"
$crystal = 8000000
Config Timer1 = Pwm , Pwm = 8 ,
 Compare A Pwm = Clear Down ,
 Prescale = 8
Config Portb = Input
Config Portd.5 = Output
Config Portc = Output
Portc.0 = 1
Portc.1 = 0
Pwm1a = 1
Do
Debounce Pinb.0 , 1 , L1 , Sub
Debounce Pinb.1 , 1 , L2 , Sub
Debounce Pinb.2 , 1 , L3 , Sub
Loop
End
L1:
Incr Pwm1a
If Pwm1a > 254 Then Pwm1a = 254
Return
L2:
Decr Pwm1a
If Pwm1a = 255 Then Pwm1a = 1
Return
L3:
Toggle Portc.0 : Toggle Portc.1
Return
```

مثال : برنامه ای بنویسید که توسط چهار کلید کنترلی مطابق با جدول زیربرروی دو موتور انجام داد.

حرکت	M1	M2
جلو	راستگرد وmax	**راستگرد وmax**
عقب	چپگرد وmax	**چپگرد وmax**
چپ	راستگرد وmid	**چپگرد وmax**
راست	راستگرد وmax	**چپگرد وmid**




```
$regfile = "m16def.dat"
$crystal = 8000000
Config Timer1 = Pwm , Pwm = 8 ,
 Compare A Pwm = Clear Down ,
 Compare B Pwm = Clear Down ,
 Prescale = 8
Config Portb = Input
Config Portd.4 = Output
Config Portd.5 = Output
Config Portc = Output
Do
Debounce Pinb.0 , 1 , L1 , Sub
Debounce Pinb.1 , 1 , L2 , Sub
Debounce Pinb.2 , 1 , L3 , Sub
Debounce Pinb.3 , 1 , L4 , Sub
If Pinb.0 = 0 And Pinb.1 = 0 And
Pinb.2 = 0 And Pinb.3 = 0 Then
Portc = 0
Loop
End
```

L1:

Pwm1a = 254 : Pwm1b = 254

Pinc.0 = 1 : Pinc.1 = 0 : Pinc.2 = 1

Pinc.3 = 0

Return

L2:

Pwm1a = 254 : Pwm1b = 254

Pinc.0 = 0 : Pinc.1 = 1 : Pinc.2 = 0

Pinc.3 = 1

Return

L3:

Pwm1a = 254 : Pwm1b = 100

Pinc.0 = 1 : Pinc.1 = 0 : Pinc.2 = 0

Pinc.3 = 1

Return

L4:

Pwm1a = 100 : Pwm1b = 254

Pinc.0 = 1 : Pinc.1 = 0 : Pinc.2 = 0

Pinc.3 = 1

Return

مبدل ADC

ADC یک IC است که وظیفه آن تبدیل یک مقدارآنالوگ به دیجیتال است.مقدار دیجیتال بدست آمده وابسته به دقت ADC ،ولتاژمرجع و مقدار ورودی آنالوگ است.

دقت ADC :می تواند 8/9/10 بیتی باشد بر این اساس بین مقدارmin وmax ولتاژآنالوگ ورودی(ولتاژمرجع) را به 256/512/1024 سطح مختلف تقسیم بندی می کند.

مثال: $V_{min}A=0$ و $V_{max}A=5$ و با فرض اینکه دقت ADC ،10 بیتی باشد

$$\frac{V_{max}-V_{min}}{1024}=\frac{5}{1024}=4.9\,mVolts$$

در این حالت به ازای افزایش 4.9 میلی ولت در ولتاژ آنالوگ ورودی یک واحد به عدد دیجیتال افزوده می شود.

مثلاً سنسور lm35 برای اندازه گیری دمای محیط به کار می رود.در این سنسور به ازای یک درجه افزایش دما ولتاژ خروجی سنسور 10mv کاهش پیدا می کند،از این رو بایستی از ADC استفاده کرد که دقت آن کمتر از 10mv باشد.

ولتازمرجع:حداکثرولتاژی که ADC می تواند آن را اندازه گیری کند.یک پایه برروی ADC تعبیه شده که به آن V_{ref} می گویند،در مبدل ADC میکرو 3 نوع مختلف V_{ref} وجود دارد.

V_{ref} : مرجع داخلی : internal ..... V_{ref}= 2.56 volts

مرجع خارجی : V_{ref}=0~5

ولتاژمرجع : AVCC

نکته : VCC – 0.3<AVCC<VCC +0.3

درATMEGA8 پورت C به عنوان ADC است که 8 کانال C_0~C_7 برای وصل کردن شکل موج آنالوگ به کار می رود

درATMEGA16 و بالاترپورت A به عنوان ADC است که 8 کانال A_0~A_7 برای وصل کردن شکل موج آنالوگ به کار می رود.

نکته: از پایه ای که به عنوان ADC تعریف شده است نمی توان به عنوان I/O استفاده کرد.

پیکربندی ADC :

Config adc=single/free , prescaler=auto , reference=internal/Avcc/Vref

دستورشروع به کارADC :

START ADC

خواندن مقدارADC :

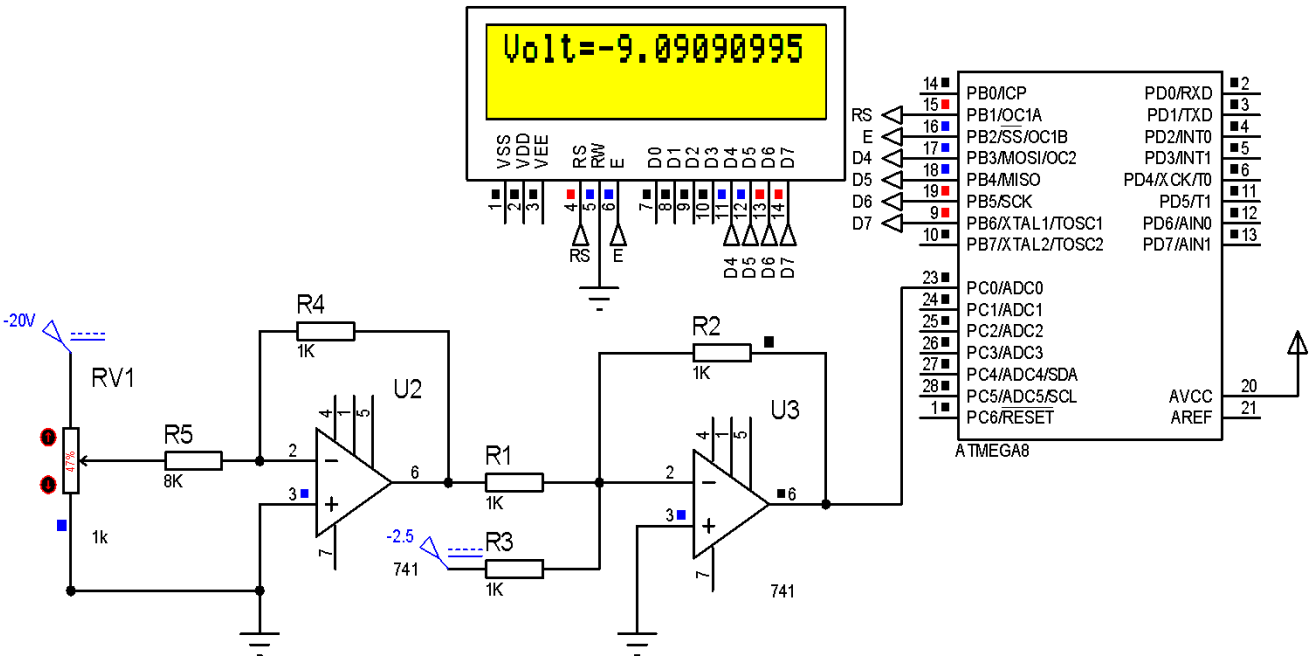
var=getadc (شماره کانال موردنظر)

نکته : متغییرvar بای از نوع word باشد

تبدیل آنالوگ ورودی به عدد دیجیتال:

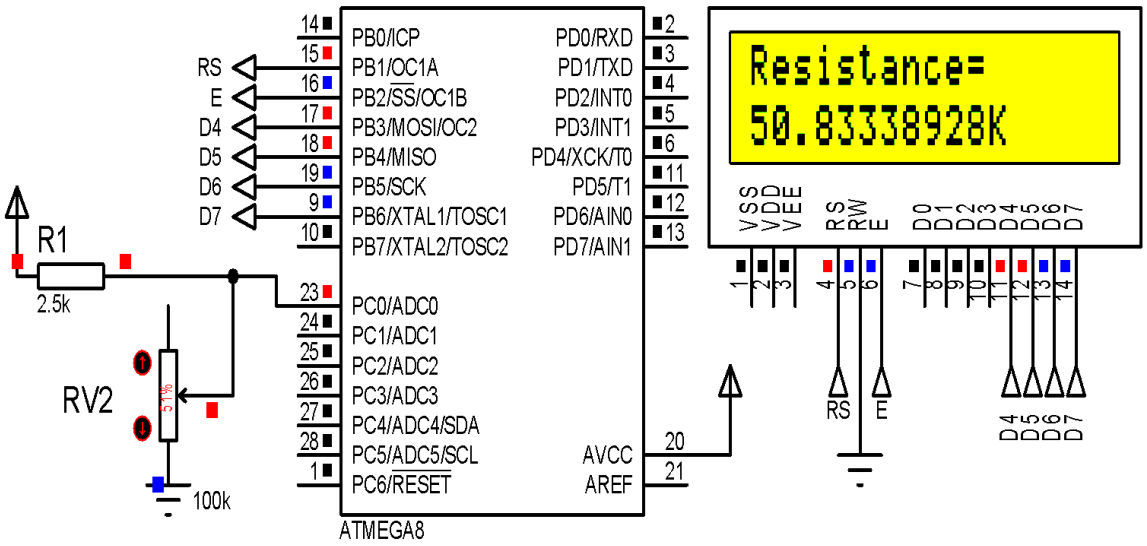
$$\text{عدد دیجتال} = \frac{Vin}{\text{دقت}}$$

مثال : برنامه ای بنویسید که ولتاژ آنولوگ DC ورودی 20~20- را اندازه گیری کند.




```
$regfile = "m8def.dat"
$crystal = 8000000
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = Pb.3 ,
 Db5 = Pb.4 , Db6 = Pb.5 , Db7 =
Pb.6 , Rs = Pb.1 , E = Pb.2
Config Adc = Single , Prescaler =
 Auto , Reference = Avcc
Start Adc
Dim W As Word , S As Single
Cls : Cursor Off
Do
W = Getadc(0)
S = W * 5
S = S / 1023
S = S - 2.5
S = S * 8
Home : Lcd "Volt=" ; S ; "   "
Loop
End
```

مثال :برنامه ای بنویسید که مقاومت در رنج 100k~ 0 اهم را اندازه گیری کند و برروی lcd نمایش دهد.




```
$regfile = "m8def.dat"
$crystal = 8000000
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = Pb.3 ,
Db5 = Pb.4 , Db6 = Pb.5 , Db7 = Pb.6
 , Rs = Pb.1 , E = Pb.2
Config Adc = Single , Prescaler =
 Auto , Reference = Avcc
Start Adc
Dim W As Word , S As Single
Cls : Cursor Off
Do
W = Getadc(0)
S = W * 5
S = S / 1024
S = 5 / S
S = S - 1
S = 2.5 / S
If S < 0 Then S = 0
Home : Lcd "Resistance="
Home Lower : Lcd S ; "K          "
Loop
End
```