



TRISB=%10000011(binary)  
 TRISB=\$83(hexadecimal)  
 TRISB=131(decimal)

```
IF portb.0=1 THEN
    portb.3=1
ENDIF
```

Eğer giriş olarak ayırladığım b0 portuna 1 geliyorsa(örn. Butona basıldığında)  
 Çıkış olarak ayarladığım b3 portunu 1 yap yani 5 volt çıksın(örn. ledi yak)

```
portb=8      => portb=%00001000
portb=portb<<1 => portb=%00010000=16
portb=portb*2 => portb=%00100000=32
portb=~portb  => portb=%11011111 =223
portb=portb//10 => portb=3
```

ADCON1=7  
 Analog Digital converter  
 Analog portları digital port olarak kullan

Değişken tanımlama : x VAR BYTE : y VAR WORD  
 Dizi tanımlama: dizi VAR BYTE[8] : dizi[3]=12  
 Porta isim verme: SYMBOL led=portb.3 (led=1 => portb.3=1 )

EPROM'a yazma okuma:  
 WRITE 0,sayı1 (0. registera sayı1'i yaz)  
 READ 0,sayı2 (sayı2'ye 0. registerdaki değeri yaz)

Döngüler:

label:

GOTO label

```
FOR i=0 TO 9
    x=x+1
NEXT i
```

```
i=0
WHILE i<10
    x=x+1
    i=i+1
WEND
```

```
i=0
REPEAT
    x=x+1
    i=i+1
UNTIL i=10
```

Fonksiyon çağırma:

label:

GOSUB fonksiyon

GOTO label  
 fonksiyon:

RETURN

```
sayi1=179
sayi2=sayı1 DIG 0 => sayi2=9
sayi2=sayı1 DIG 1 => sayi2=7
sayi2=sayı1 DIG 2 => sayi2=1
```

Select case ve if else yapısı:

```
SELECT CASE renk
    CASE "kirmizi"
        GOSUB dur
    CASE "yesil"
        GOSUB hareket
END SELECT
```

```
IF x>5 THEN
    ...
ELSEIF x<10 THEN
    ...
ELSE
    ...
ENDIF
```

LCD: registerlar: \$40,\$48,\$50,\$58,\$60,\$68,\$70,\$78  
 reg. Numara: 0, 1, 2, 3, 4, 5, 6, 7  
 registera kaydetme:

```
LCDOUT $fe,$40,1,2,4,8,16,8,4,2
00001
00010
00100
01000
10000
01000
00100
00010
```

ekrana yazdırma:

```
LCDOUT $fe,1 (ekranı temizleme)
LCDOUT $fe,"merhaba"(ilk satır)
LCDOUT $fe,$c0,"dunya"(ikinci satır)
LCDOUT 0 (0. reg.($40) simgeyi yaz)
LCDOUT #sayi (sayinin değerini yaz)
```

ekrandaki haneler:

```
$80,$81,$82...(birinci satır)
$c0,$c1,$c2...(ikinci satır)
```

Terminal:

```
HSEOUT [ "bir sayi girin:",13,10] (terminale yazı yazdırma)
HSEOUT [DEC i,"inci sayi: ",DEC j,13,10]
HSEIN [girdi] (terminalden girilen bir karakteri girdiye kaydeder)
HSEIN [WAIT " ", girdi] (terminalden boşluk girilene kadar bekler)
HSEIN 1000,label, [girdi] (1 sn girdi bekler yoksa label'a atlar)
HSEIN [DEC girdi] (sayı girdi alır enter'a basıldığında tamamlanır)
HSEIN [STR girdi] (4 karakterlik bir stringi alır)
HSEIN [DEC2 girdi] (2 basamaklı decimal sayıyı bekler)
HSEIN [SKIP n,girdi] (n karakteri görmezden gelir)
```

```
BRANCH index,[label0,label1,label2,...] (index=2 -> label ikiye dallanır)
x = 10 MAX 20 -> x=20 , x = 10 MIN 20 -> x=10
x = DCD 0 -> x=%000000001 , x = DCD 3 -> x=%00001000
x=%01000000->y=NCD x->y=6 (en yüksek bitin bulunduğu basamak)
x=%11100001->x=x REV 8->x=%1000111,x=x REV 3->x=%11100100
PULSOUT port, periyot - PULSOUT portb.7,100
FREOUT port, zaman, frekans - FREOUT portb.7, 100, 2 - (0-32767)
SOUND port, [frekans, süre] - SOUND portc.2, [45,100] - (0-255)
PWM port,periyot,süre - PWM portc.2,i,10 (i->0-255)
```