



**Teknoloji Fakültesi Elektrik Elektronik Mühendisliği Bölümü**

# **EE-302**

# **Mikroişlemciler**

**Port Giriş/Çıkış ve Bit/Byte işlemleri**

**3. Hafta**

**Prof. Dr. Mehmet DEMİRTAŞ**

# Port Giriş/Çıkış Komutları

- Port giriş/çıkış komutları portların giriş veya çıkış olarak kullanılıp kullanılmayacağı, port ve/veya port pin'lerinin çıkış durumlarını port girişlerindeki bilgilerin okunması gibi işlemlerin gerçekleştirilmesini sağlayan komutlardır.
- Bu komut gurubunu, giriş/çıkış tanımlama komutları, çıkışa bilgi yönlendirme komutları ve girişten bilgi okuma komutları olarak genel bir sınıflandırma ile anlatabiliriz. Ayrıca bit bazında uygulama komutları da bulunmaktadır.
- Giriş/Çıkış tanımlama : `set_tris_x ( );`
- Çıkışa bilgi yönlendirme : `output_x ( );`, `output_low/high/bit ( );`, `output_toggle ( );`
- Giriş bilgisi okuma : `input ( );`, `input_x ( );`, `input_state ( );`

# Port Giriş/Çıkış Komutları

## SET\_TRIS\_X( ) Komutu

Bu komut port pin'lerinin hangisinin giriş pini hangisinin çıkış pin'i olacağını belirtir. Komuttaki "X" yerine işlem yapılacak port'un ismi (A, B, C, ...) parantez içine ise heksadesimal olarak 8 bitlik değer girilir. Pin değeri "0" girilmişse o pin'in çıkış pin'i olarak kullanılacağı, "1" girilmişse o pin'in giriş pin'i olarak kullanılacağı anlamındadır.

Kullanımı : `set_tris_b (0b00001111);` veya `set_tris_b (0x0f);`

şeklindedir. Bu iki komutta aynı anlamdadır. Sadece birinde değer binary olarak, diğerinde heksadesimal sayı olarak girilmiştir. Bu komut bize B port'unun B0, B1, B2, B3 pin'lerinin giriş olarak kullanılacağını, B4, B5, B6, B7 pin'lerinin ise çıkış olarak kullanılacağını bildirir.

# Port Giriş/Çıkış Komutları

## GET\_TRIS\_X ( ); Komutu

İstenen port'un TRIS kaydedicisi değeri ile geri dönen bir fonksiyondur. Komutta "X" yerine TRIS kaydedicisi değeri okunacak port'un ismi girilir.

```
set_tris_a(0xFE);
```

```
....
```

```
....
```

```
bilgi=get_tris_a( );
```

yukarıdaki örnekte bilgi değişkeninin değeri "FE" olur.

# Port Giriş/Çıkış Komutları

## **OUTPUT\_LOW( ); Komutu**

Bu komut ile istenen port'un istenen bit'i lojik "0" yapılır. Yani istenen ucun çıkışı 0 (sıfır) yapılır. Komutta parantez içine işlem yapılacak pin ismi girilir.

`output_low (pin ismi);`

`output_low (pin_A2);`

komutu ile A port'unun A2 numaralı ucu lojik 0 yapılmış olur.

# Port Giriş/Çıkış Komutları

## **OUTPUT\_HIGH( ); Komutu**

Bu komut ile istenen port'un istenen bit'i lojik "1" yapılır. Yani istenen ucun çıkışı 1 (bir) yapılır. Komutta parantez içine işlem yapılacak pin ismi girilir.

`output_high (pin ismi);`

`output_high (pin_B3);`

komutu ile B port'unun B3 numaralı ucu lojik 1 yapılmış olur.

# Port Giriş/Çıkış Komutları

## **OUTPUT\_BIT ( ); Komutu**

Bu komut ile istenen port'un istenen bit'i lojik "0" veya lojik "1" yapılır.

`output_bit (pin, deger);`

`output_bit (PINC4, 1);`

komutu ile C port'unun 4. bit'inin çıkışı lojik 1 yapılmıştır.

## **OUTPUT\_X ( ); Komutu**

Bu komut ile bir port'un tüm bit'lerine tek komutla çıkış değerleri yüklenir.

`output_port ismi (deger);`

`output_B (0x0f);`

komutu ile A port'unun B0, B1, B2, B3 bitlerinin çıkışları lojik 1, B4, B5, B6, B7 bit'lerinin çıkışı ise lojik 0 yapılıyor (0x0f=00001111). Parantez içine yazılan değer 8 bit'lik bir sayı olmak zorundadır.

# Port Giriş/Çıkış Komutları

## **OUTPUT\_TOGGLE ( ) Komutu**

Bu komut istenen port'un istenen bitinin çıkışının durumunu değiştirir. Çıkış lojik-1 ise lojik-0, lojik-0 ise lojik-1 olur.

`output_toggle (pin ismi);`

`output_toggle (pin_ C4);`

komutu ile C4 pin'inin çıkışı lojik-1 seviyesinden lojik-0 seviyesine getirilir.



# Port Giriş/Çıkış Komutları

## INPUT ( ) Komutu

- Bu komut PIC mikrodenetleyicisinin istenen pin'nin durumunu okumaya yarar. Yani istenen giriş pin'inin lojik olarak değerini okur.

***degisken ismi = input (pin ismi);***

***bilgi=input(pin\_A2);***

komutu ile A2 pin'i girişinden dijital bilgi (0 veya 1) okunmakta ve bilgi değişkenine aktarılmaktadır.

# Port Giriş/Çıkış Komutları

## INPUT\_X ( ) Komutu

- Bu komut ile istenen giriş port'u durumu tümüyle okunur.

*degisken=input\_port ismi ( );*

*data=input\_c( );*

- komutu ile C portunun komple durumu "data" adlı değişkene aktarılmış olunur.

Değişken 8 bit'lik olmalıdır.

# Port Giriş/Çıkış Komutları

## INPUT\_STATE ( )

- Bu komut ile istenen pin'in çıkış mı yoksa giriş mi olduğu öğrenilmektedir. Geri dönüş değeri "1" ise pin giriş, «0" ise pin çıkış olarak yönlendirilmiş demektir.

***degisken ismi = input\_state(pin ismi);***

***i=input\_state(pin\_b2) ;***

komutu ile B2 pin'inin giriş veya çıkış olarak mı yönlendirildiğini belirten bilgi değişkenine aktarılmaktadır.

# Bit ve Byte İşlemleri Komutları

Bit/Byte ile ilgili işlem yapan komutlar ile programımızda bit seviyesindeki işlemleri rahatlıkla yapabiliriz. Bu komutlarla istenen bir değerin (örneğin 0xA5 değeri) istenen bit'i üzerinde (örneğin değerin 5.bit'i) istenen değer verme ve kontrol döngüleri işlemleri gerçekleştirilebilir.

## **BIT\_SET ( ) Komutu**

Bu komut istenen değişkenin istenen bit'ini "1 (bir)" yapar.

`bit_set (değişken, bit numarası);`

`int x;`

`x=7;`

`bit_set (x,3);` // x'in yeni değeri şimdi 15 oldu. Yukarıdaki örnekte x değişkeni tanımlanıyor ve içine 7 değeri atanıyor. Bit\_set komutu ile de x değişkeninin 3.bit'i bir yapılıyor. Yani x değişkeni (7 = 0111) iken, 3.bit'i bir olur ve (1111 = 15) değerini alıyor.

# Bit ve Byte İşlemleri Komutları

## **BIT\_CLEAR ( )**Komutu

Bu komut istenen değişkenin istenen bit'ini "0 (sıfır)" yapar.

`bit_clear (değişken ismi, bit numarası);`

`int x;`

`x=6;`

`bit_clear (x,2); // x'in yeni değeri şimdi 2 oldu.`

Yukarıdaki örnekte x değişkeni tanımlanıyor ve içine 6 değeri atanıyor. Bit\_clear komutu ile de x değişkeninin 2.bit'i sıfır yapılıyor. Yani x değişkeni (6 = 0110) iken, 2.bit'i sıfırlanıyor ve (0010 = 2) değerini alıyor.

# Bit ve Byte İşlemleri Komutları

## **BIT\_TEST ( ) Komutu**

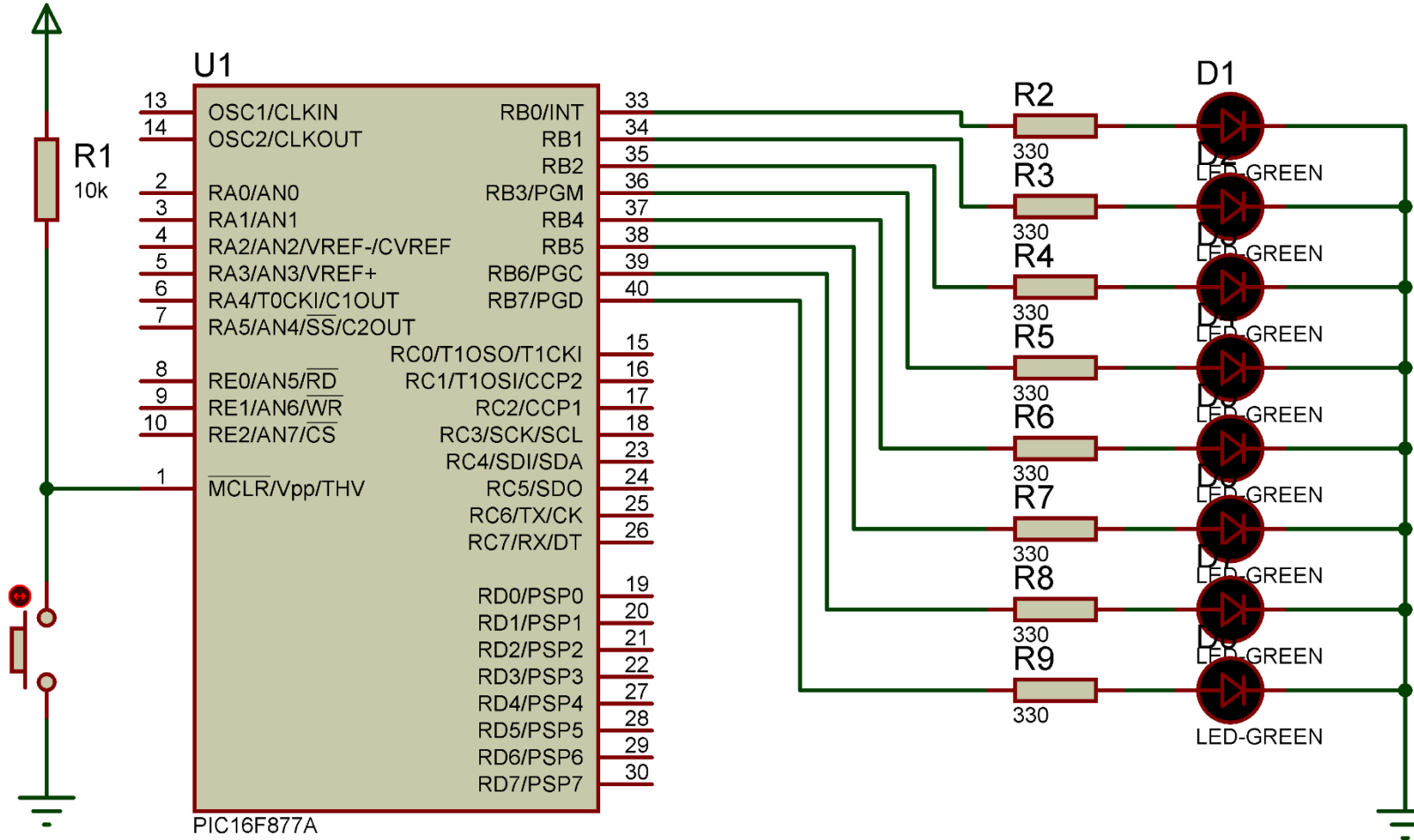
Bit\_test komutu ile istenen değişkenin istenen bit'inin değerinin durumu öğrenilir (Bit 0'mı?, 1 mi? o öğrenilir). Fonksiyon 0 veya 1 bilgisi ile geri döner.

bit\_test ( değişken ismi, bit numarası);

x=0x35; // Heksadesimal olarak 35 değeri x değişkenine yükleniyor.

y=bit\_test( x, 5 );

# Örnek Uygulama-1



# Örnek Uygulama-1

```
1  #include <16f877A.h>           // Kullanılacak denetleyicinin başlık dosyası tanıtılıyor
2  #fuses XT,NOWDT,NOPROTECT      // Denetleyici konfigürasyon ayarları
3  #use delay(clock=4000000)      // Gecikme fonksiyonu için kullanılan osilatör frekansı
4  #use fast_io(b)               // Port yönlendirme komutları B portu için geçerli
5  int i;                        // integer "i" değişkenini tanımla
6                               // *** ANA PROGRAM *** /
7  void main()
8  {
9      set_tris_b(0x00);          // B portunun tüm pinlerini çıkış olarak ayarla
10     output_b(0x00);            // PortB'nin tamamını sıfırlama (LED'leri söndürme)
11     while(1)
12     {
13         output_b(0x0A);
14         delay_ms(500);
15         output_b(0x00);
16         delay_ms(500);
17         output_b(0x05);
18         delay_ms(500);
19         output_b(0x00);
20         delay_ms(500);
21     }
22 }
```



# Örnek Uygulama-2

```
1  #include <16f877.h>
2  #fuses XT,NOWDT,NOPROTECT
3  #use delay(clock=8000000)
4  #use fast_io(b)
5  int i;    //integer "i" değişkenini tanımla
6  /** ANA PROGRAM ***/
7  void main()
8  {
9      set_tris_b(0x00); // B portunun tüm pinlerini çıkış olarak ayarla
10     output_b(0x00);   // B portunun çıkışlarını temizle
11
12     for(i=0;i<10;i++)
13     {
14         output_b(0xFF); // PortB'ye bağlı LED'leri yak
15         delay_ms(200);  // Gecikme
16         output_b(0x00); // PortB'ye bağlı LED'leri söndür
17         delay_ms(200);  // Gecikme
18     }
19 }
```

# Kaynaklar

- CCS C Programlama Kitabı, Serdar Çiçek, Altaş Yayıncılık
- Mikroelektronika C programlama e-kitabı «<https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c>»
- Mikroelektronika C programlama e-kitabı «<https://www.mikroe.com/ebooks/pic-microcontrollers-programming-in-c/inputoutput-ports>»