

KIRIKKALE ÜNİVERSİTESİ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ  
Mikroişlemciler Laboratuvarı Dersi

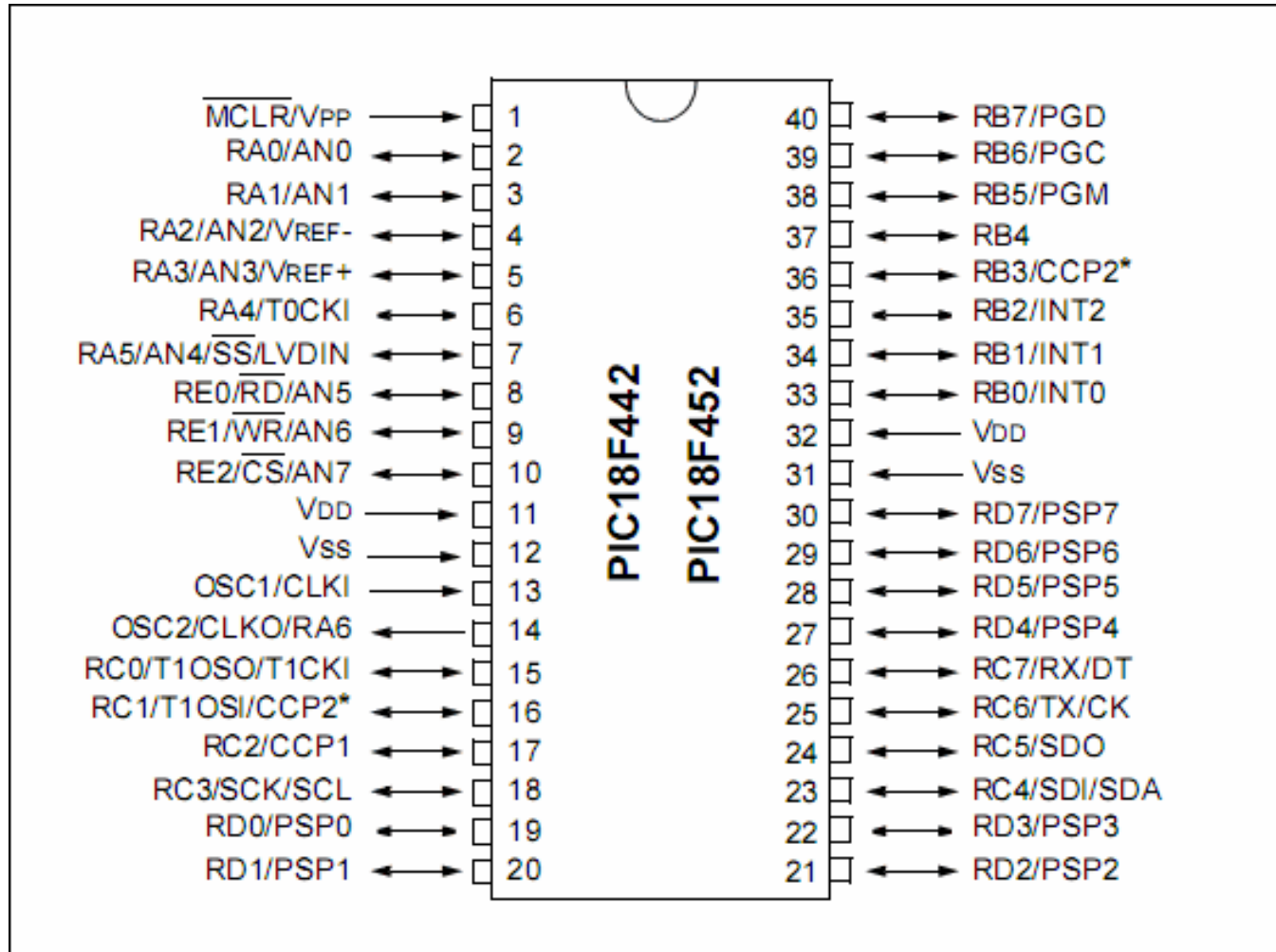
PIC18 Genel Amaçlı Port  
Kullanımı

E.POLAT  
H.AYDİLEK

# GENEL AMAÇLI INPUT/OUTPUT

- PIC 18'de genel amaçlı veri iletişimi için I/O portları kullanılır.
- PIC18F452 için 5 tane genel amaçlı I/O portu vardır:
  - PORTA : 7 bit, çift yönlü (input, output)
  - PORTB : 8 bit, çift yönlü (input, output)
  - PORTC : 8 bit, çift yönlü (input, output)
  - PORTD : 8 bit, çift yönlü (input, output)
  - PORTE : 3 bit, çift yönlü (input, output)
- Bu portların her bir pini input (veri alma) veya output (veri gönderme) amaçlı kullanılabilir.

# PIC 18f452 Bacak Yapısı



- Her bir port 3 adet kaydediciye sahiptir.
  - TRIS kaydedicisi (veri yönlendirme kaydedicisi)
  - PORT kaydedicisi (pin üzerindeki seviyeyi okuyan kaydedici)
  - LAT kaydedicisi ( çıkış kaydedicisi)
- Veri alma veya gönderme işlem seçimi, bu portlarla beraber kullanılan TRISA, TRISB, TRISC, TRISD ve TRISE kaydedicileri aracılığı ile olur (1:Input, 0:Output) .
- LATA, LATB, LATC, LATD ve LATE : Port A,B,C,D ve E'ye gönderilen en son verileri saklayan flip-floplardır (latch).
- Bazı portların genel I/O özelliği dışında özel amaçları vardır. Bu amaçlar ilgili konu geldikçe anlatılacaktır.

# PORTA

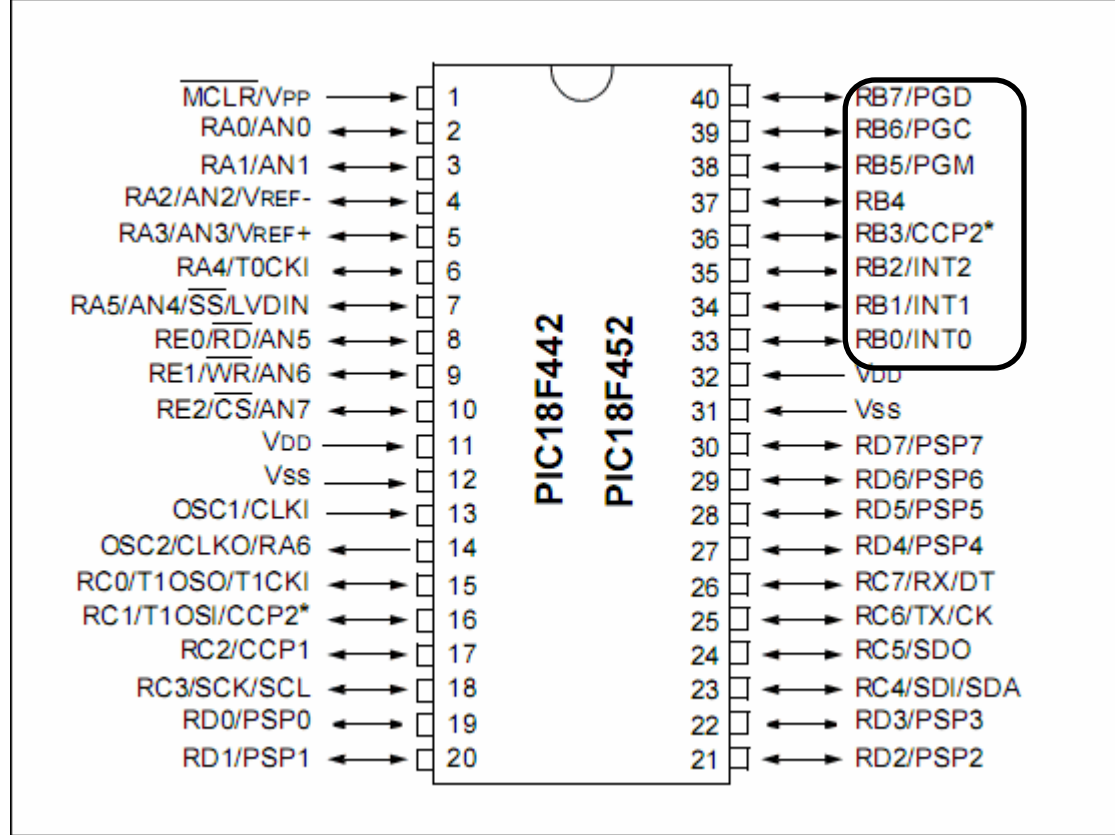
- 7 bit genişliğinde çift yönlü portdur. TRISA kaydedicisi veri yönünü belirler.(TRISA=0 //Output; TRISA=1 // Inputs)
- RA4 pini Timer0 modülü ile ilişkilidir.
- RA6 pini osilatör girişidir ve tek yönlüdür
- Diğer tüm pinler aynı zamanda analog girişler ile ilişkilidir.

- TRISA = 0xFF                   // Tüm bitleri input yap
- PORTA=0X00;                   // A portunu sıfırladık
- PORTAbits.RA0                // A portu RA0 pini

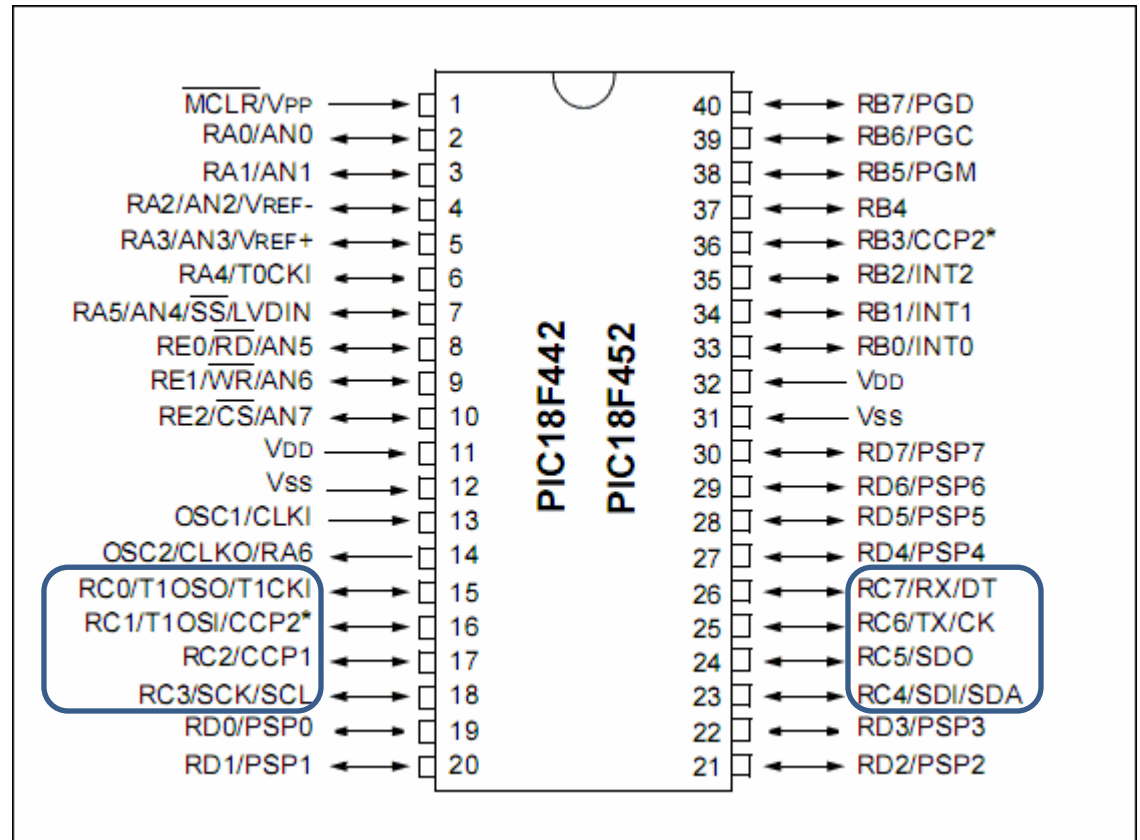
# PORTB

- 8 bit genişliğinde çift yönlü portdur.
- Power-on Reset durumunda RB4:RB0 varsayılan olarak analog giriş olarak yapılandırılır ve '0' okunur. RB7:RB5 Dijital giriş olarak yapılandırılır.
- RB4:RB0 pinleri dijital pin (input veya output) olarak değiştirmek için PBADEN (PORTB A/D Enable Bit) biti 0 yapılmalıdır.

- Örnek:
- PORTB'nin üst 4 bitini output, alt 4 bitini input yap:
  - TRISB = 0x0F
- RB4 ve RB5 pinlerine 1 gönder, RB6 ve RB7 pinlerine 0 gönder:
  - PORTB = 0x30



# PORTC



- PORTC pinleri (RC0-RC7) diğer birçok fonksiyon ile ortak kullanılmaktadır. Bu port çok zorunlu olmadığı sürece kullanılmamalıdır.
- Bir port kullanılması gerekiyorsa önce PORTB daha sonra PORTA düşünülmelidir.

# PORTD

- 8 bit genişliğinde çift yönlü portdur.
- Veri yönü TRISD kaydedicisi ile belirlenir. PORTD tamponludur; bir veri gönderilene kadar PORTD ve LATD kaydedicileri eski bilgileri saklar.



# PORTE

- 3 bit genişliğinde çift yönlü portdur.
- TRISE kaydedicisi veri yönünü belirler.
- Bütün pinleri analog giriş olarakta kullanılabilir.

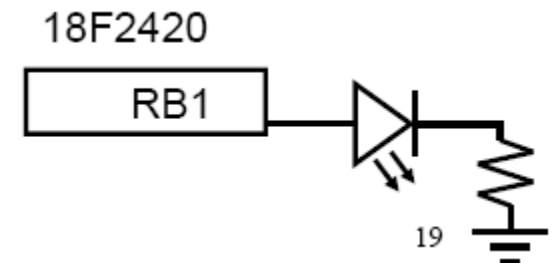
# Örnek C Programı (Led Yak/Söndür)

```
#include<p18f242.h> // Bu program PIC18F için MPLAB C derleyicisi ile derlenebilir.  
void bekle(void)      //Döngü sayıları değiştirilerek bekleme süresi değiştirilebilir.
```

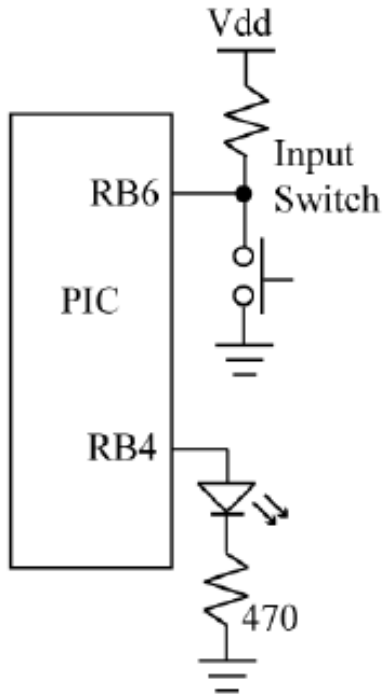
```
{  
    unsigned int i,j;  
    for (i=0; i<1000; i++)  
        for(j=0; j<500; j++) { };  
}
```

<http://www.microchip.com/sourcecode>  
sitesinden mili ve mikro saniye cinsinden bekleme  
fonksiyonu elde edilebilir.

```
void main(void)  
{  
    TRISB=0;                // PORTB tüm bitleri output yap  
    PORTBbits.RB1=0;        // RB1 pinini 0 yap  
    while (1)  
    {  
        bekle();  
        PORTBbits.RB1=1;    // RB1'e bağlı ledi yak  
        bekle();  
        PORTBbits.RB1=0;    // RB1'e bağlı ledi söndür  
    }  
}
```



## Örnek-1: Anahtara Kaç Kere Basılıp/Bırakıldığının Bulunması



Yanlış Program:

```
main()
{
    int i;
    TRISB=0xEF;
    PORTBbits.RB4=0;
    i=0;
    while (1)
    {
        if (! PORTBbits.RB6)
            i++;
    }
}
```

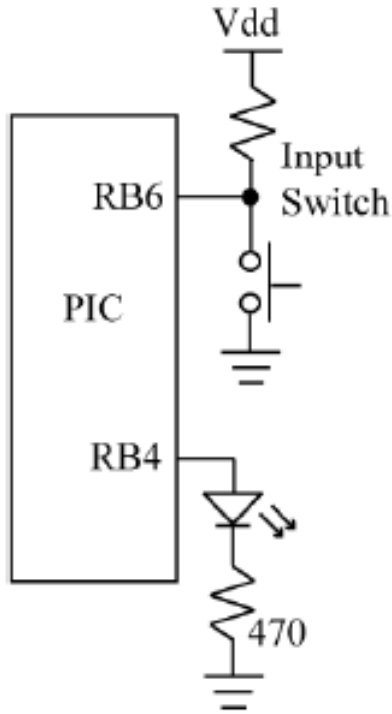
Anahtara basıp bırakmak belli bir zaman alacağından, i bu zaman süresince artırılır ve yanlış değer elde edilir.

Doğru Program:

```
main()
{
    int i;
    TRISB=0xEF;
    PORTBbits.RB4=0;
    i=0;
    while (1)
    {
        while (PORTBbits.RB6); // Döngü 1
        bekle();
        while ( ! PORTBbits.RB6); // Döngü 2
        bekle();
        i++;
    }
}
```

Anahtara basılmadığı sürece döngü 1 çalışır, Basılı tutulduğu sürece döngü 2 çalışır, Bırakıldığı zaman i 1 artırılır ve yanlış sayma önlenmiş olur.

## Örnek-2: Anahtara Her Basılıp/Bırakıldığında Ledin Toggle Yapılması



```
main()
{
    TRISB = 0xEF;
    PORTBbits.RB4 = 0;
    while (1)
    {
        while (PORTBbits.RB6); // anahtara basılana kadar burayı çalıştır
        bekle(); // basılıp bırakılma arasındaki zamanda hiçbir şey yapma
        while ( ! PORTBbits.RB6); // anahtar bırakılana kadar burayı çalıştır
        bekle(); // bırakılma anında hiçbir şey yapma
        PORTBbits.RB4 = 1; // Ledi yak

        while (PORTBbits.RB6); // anahtara basılana kadar burayı çalıştır
        bekle(); // basılıp bırakılma arasındaki zamanda hiçbir şey yapma
        while ( ! PORTBbits.RB6); // anahtar bırakılana kadar burayı çalıştır
        bekle(); // bırakılma anında hiçbir şey yapma
        PORTBbits.RB4 = 0; // Ledi söndür
    }
}
```