



Konu 1 : Dış Kesme(External Interrupt)

Örnek Kod

```
#include<16f876a.h> // Ön işlemci fonksiyon ayarları yapılır
//#fuses HS // Pic Wizard ile otomatik olarak da ayarlanabilir.
// High Speed - Yüksek Hızlı Kristal veya Seramik Resonatör kullanılacak.
#fuses XT // XTal - Kristal veya Seramik Resonatör
#fuses NOWDT // No Watch Dog Timer - WDT kullanılmayacak.
#fuses NOLVP // No Low Voltage Programming - LVP kullanılmayacak.
#fuses NOPROTECT // Program Belleğine (ROM) yazılan verilerin okunmasına izin verir.

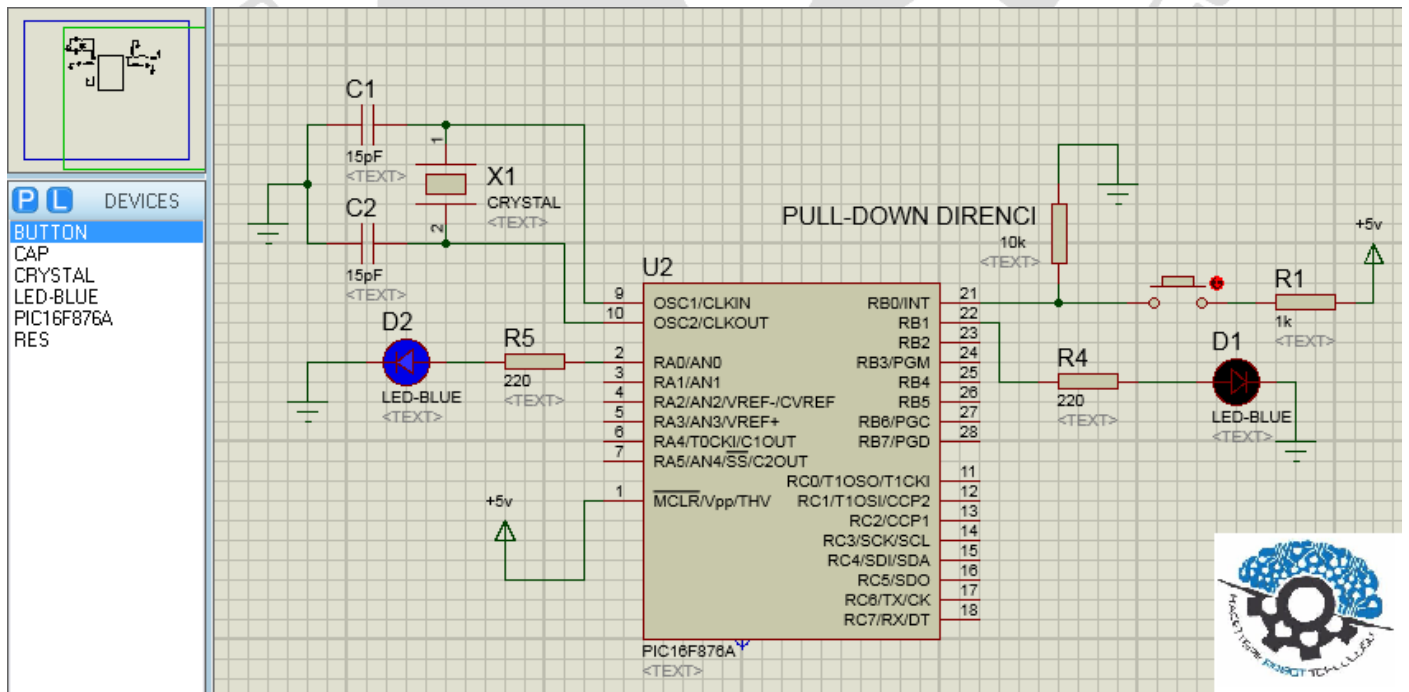
#int_ext // Dış Kesme (External Interrupt)
void DIS_KESME() // Kesme oluştuğunda yapılması istenen komutlar yazılır.
{ // Tanımlanan kesme fonksiyonu Ana(Main) fonksiyondan önce tanımlanmalıdır.
// B1 portuna bağlanmış LED 1 sn arayla yanıp söner.
output_high(pin_b1);
delay_ms(1000);
output_low(pin_b1);
delay_ms(1000);
}

void main(){
set_tris_a(0x00); // A portu, çıktı(output) olacak şekilde ayarlanır.
set_tris_b(0x01); // B0 giriş(input),B1-7 çıktı(output) olacak şekilde ayarlanır.
output_b(0x00); // Portlar sıfırlanır

ext_int_edge(H_TO_L); // Kesmenin düşen kenarda( High to Low) oluşacağı tanımlanır.
enable_interrupts(INT_EXT); // Dış Kesme(External Interrupt) etkinleştirilir.
enable_interrupts(GLOBAL); //Aktif edilen keslemelere yetki verilir( Kesmelerin oluşması için gereklidir.).

while(1){
output_high(pin_a0); // A0 pinine bağlanan LED 1 sn arayla yanıp söner.
delay_ms(1000);
output_low(pin_a0);
delay_ms(1000);
}
}
```

Proteus Çizimi



Konu 2 : Timer0 Kesmesi ile Sayıcı

Örnek Kod

```
#include<16f876a.h>
//#fuses HS
#fuses XT
#fuses NOWDT
#fuses NOLVP
#fuses NOPROTECT

// Ön işlemci fonksiyon ayarları yapılır
// Pic Wizard ile otomatik olarak da ayarlanabilir.
// High Speed - Yüksek Hızlı Kristal veya Seramik Resonatör kullanılacak.
// XTal - Kristal veya Seramik Resonatör
// No Watch Dog Timer - WDT kullanılmayacak.
// No Low Voltage Programming - LVP kullanılmayacak.
// Program Belleğine (ROM) yazılan verilerin okunmasına izin verir.

#USE delay(clock = 4000000)

int sayi = 0; // Global bir değişken tanımlanır.
// Tanımlanan kesme fonksiyonu Ana(Main) fonksiyonundan önce tanımlanmalıdır.

#int_timer0 // Timer 0 Kesmesi
void timer0_kesmesi(){ // Kesme oluştuğunda yapılması istenen komutlar yazılır.
    set_timer0(65534); // Butona ( 65536 - 65534 = 2 ) * ( Bölme oranı = 2 ) = 4 kez basıldığında kesme oluşur.
    // Timer0' ın başlangıç değeri tanımlanır( Kesme oluştuğunda Ana(Main) fonksiyonundaki
    // başlangıç değeri sıfırlandığından dolayı kesmenin içinde tekrar tanımlanır).

    sayi++;
    output_b( sayi );

    if( sayi > 3)
        sayi = 0;
}

void main(){
    set_tris_a(0xff); // A4 pini giriş(input) ,diğerleri çıktı(output) olacak şekilde ayarlanır.
    set_tris_b(0x00); // B portu, çıktı(output) olacak şekilde ayarlanır.
    set_tris_c(0x00);

    output_b(0x00); //Portlar sıfırlanır
    output_c(0x00);

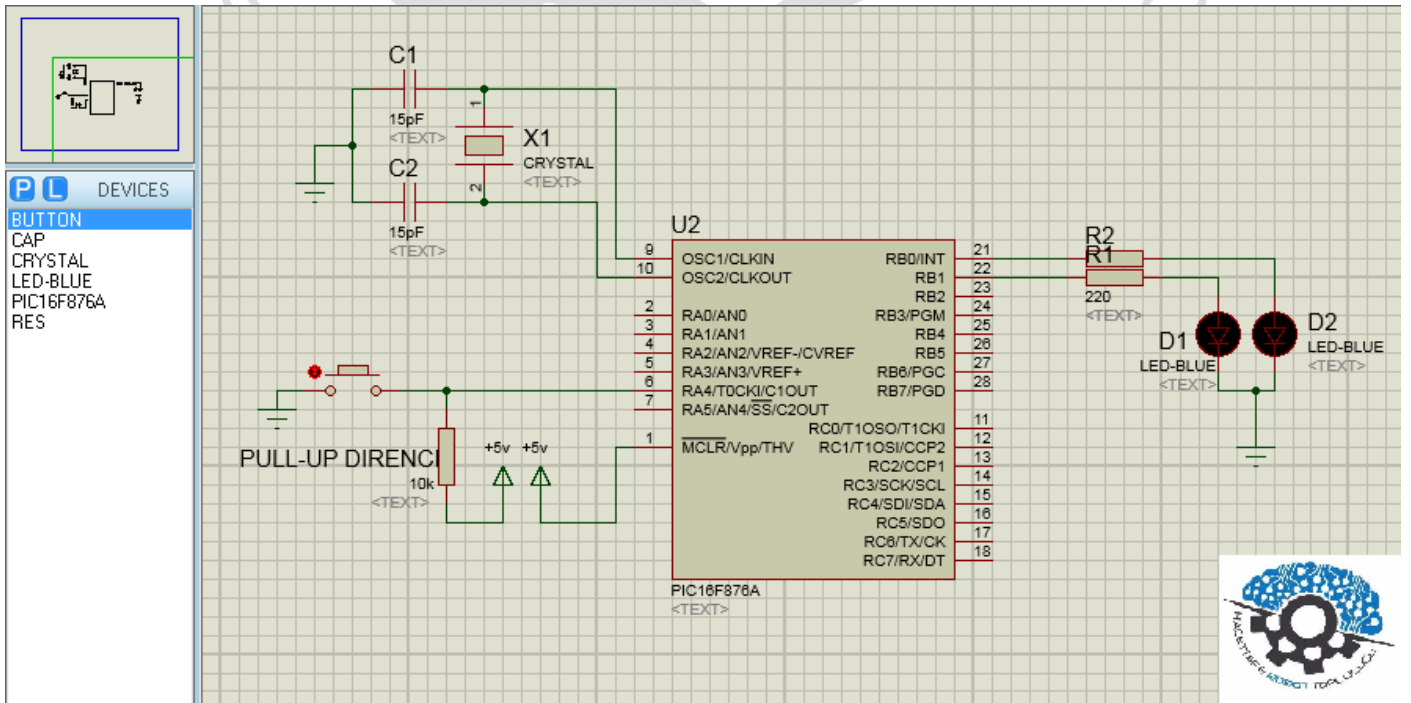
    setup_timer_0( RTCC_EXT_L_TO_H | T0_DIV_2 ); // Timer 0 ayarlanıyor( Kesme düşen kenarda olacak | Bölme oranı = 2 )
    set_timer0(65534); // Timer0' ın başlangıç değeri tanımlanır( 65534'dan başlayıp 65535' e kadar sayar.
    // 65536 olduğunda kesme oluşur);

    enable_interrupts(INT_TIMER0); // Timer 0 Kesmesi etkinleştirilir.
    enable_interrupts(GLOBAL); //Aktif edilen kesmelere yetki verilir( Kesmelerin oluşması için gereklidir.).

    while(1){
        output_c( sayi ) ;
    }
}
```



Proteus Çizimi



Konu 3 : Timer0 Kesmesi ile Zamanlayıcı

Örnek Kod

```
// Ön işlemci fonksiyon ayarları yapılır
#include<16f876a.h> // Pic Wizard ile otomatik olarak da ayarlanabilir.
//#fuses HS // High Speed - Yüksek Hızlı Kristal veya Seramik Resonatör kullanılacak.
#fuses XT // XTal - Kristal veya Seramik Resonatör
#fuses NOWDT // No Watch Dog Timer - WDT kullanılmayacak.
#fuses NOLVP // No Low Voltage Programming - LVP kullanılmayacak.
#fuses NOPROTECT // Program Belleğine (ROM) yazılan verilerin okunmasına izin verir.

#USE delay(clock = 4000000)

int16 i = 0; // Global bir değişken tanımlanır.
// Tanımlanan kesme fonksiyonu Ana(Main) fonksiyonundan önce tanımlanmalıdır.
#INT_timer0 // Timer 0 Kesmesi
void timer0_kesmesi(){ // Kesme oluştuğunda yapılması istenen komutlar yazılır.
    set_timer0(156); // Timer0' ın başlangıç değeri tanımlanır( Kesme oluştuğunda Ana(Main) fonksiyonundaki
// başlangıç değeri sıfırlandığından dolayı kesmenin içinde tekrar tanımlanır)
// Kesme Oluşma Süresi = ( T_Komut ) * ( Bölme Oranı ) * ( 256 - Timer0 başlangıç değeri )
// = 1 us * 64 * ( 256 - 156 ) = 6.4 ms

    i++; // B0 pinine bağlanan LED, 1 sn aralıklarla yanıp sönecek şekilde ayarlanır.

    if( i == 156)
        output_high(pin_b0);

    if( i == 312){
        output_low(pin_b0);
        i = 0;
    }
}

void main(){
    set_tris_a(0x00); // A portu, çıktı(output) olacak şekilde ayarlanır.
    set_tris_b(0x00); // B portu, çıktı(output) olacak şekilde ayarlanır.

    output_a(0x00); //Portlar sıfırlanır
    output_b(0x00);

    setup_timer_0( T0_INTERNAL | RTCC_DIV_64 | T0_8_BIT); // Timer 0 ayarlanıyor( Sinyal kaynağı dahili | Bölme oranı = 64 )
// Timer 0 " T0_8_BIT " modu ile 8 bitlik zamanlayıcı olarak ayarlandı.
// " T0_8_BIT " modu kullanılmazsa 16 bitlik zamanlayıcı olarak çalışır.

    set_timer0(156); // Timer0' ın başlangıç değeri tanımlanır( 156'dan başlayıp 255' e kadar
// sayar. 256 olduğunda kesme oluşur)

    enable_interrupts(INT_timer0); // Timer 0 Kesmesi etkinleştirilir.
    enable_interrupts(GLOBAL); //Aktif edilen kesmelere yetki verilir( Kesmelerin oluşması için gereklidir.).

    while(1);
}
```



Proteus Çizimi

