Teknoloji Fakültesi Elektrik Elektronik Mühendisliği Bölümü

EE-302
Mikroişlemciler

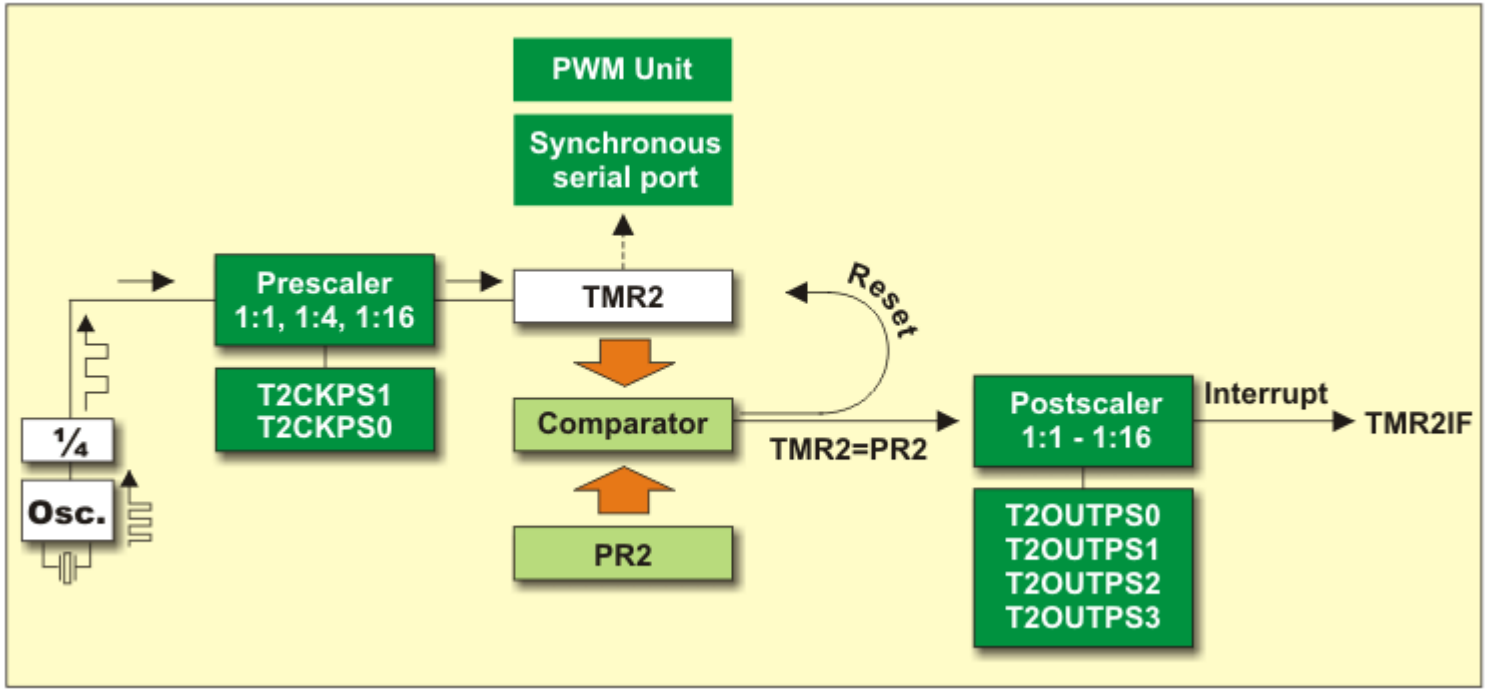
Timer2 ve WDT Uygulamaları

7. Hafta

Prof. Dr. Mehmet DEMİRTAŞ

TIMER2 Birimi

- Timer2 birimi diğer Timer0 ve Timer1 birimlerinden farklı olarak sadece zamanlayıcı olarak kullanılan 8 bit'lik bir birimdir. Timer2 birimi 8 bit'lik bölme oranı (prescaler) ve diğer zamanlayıcı/sayıcı birimlerinde olmayan 8 bit'lik postscaler değerine sahiptir. Timer2 aynı zamanda denetleyicide bulunan CCP modülündeki PWM birimi için sinyal üretmede kullanılan zamanlayıcıdır.



TIMER2 Birimi

Timer2 biriminin blok şemasında görüleceği gibi Timer2 zamanlayıcısı için denetleyicinin dahili saat sinyali (fosc/4) kullanılır. Bu sinyal bölme oranı (prescaler) ile çarpılarak kaynak saat sinyalini oluşturur. 8 bit'lik TMR2 zamanlayıcı kaydedicisi içindeki değer denetleyici frekansı ile bölme oranı ile hesaplanan frekansta artar. PR2 8 bit'lik yazılabilir ve okunabilir bir periyot kaydedicisidir. TMR2 kaydedicisi içindeki değer 00h'tan başlayarak sayar ve PR2 kaydedicisi içine yazılan değere eşit olduğunda (Eşitlik tespitini, blok semada görülen karşılaştırıcı -comparator- birimi yapar) bir eşitlik sinyali üretilir ve TMR2 kaydedicisi sıfırlanır.

TIMER2 Birimi

Eşitlik sinyali sonucunda PR2 kaydedicisi de FFh değerini alır. Postscaler değeri meydana gelen bu eşitlik sinyallerinden kaç defa meydana geldikten sonra Timer2 kesmesi sinyali oluşacağını belirler. Örneğin postscaler değeri 2 ise, 2 defa TMR2 kaydedicisi değeri PR2 değerine eşit olunca Timer2 kesmesi oluşur. Tüm bu işlemler T2CON kaydedicisi ile belirlenir. Bölme oranı (prescaler) ve Postscale değerleri POR, MCLR, WDT vb. sıfırlama (reset) sinyallerinden biri oluştuğunda veya TMR2 kaydedicisine veri yazma işlemi ile T2CON kaydedicisine veri yazma işlemlerinden biri gerçekleştiğinde sıfırlanır.

		R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	Features
T2CON	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Bit 7 = Kullanılmayan bit. Okunurken bu bit "0" bilgisi olarak okunur.

Bit 6-3 **TOUTPS3:TOUTPS0** = Timer2 Postscale değeri seçme bit'leri

0000: 1:1
0001: 1:2
0002: 1:3
0003: 1:4
...
...
1111 = 1:16

Bit 2 **TMR2ON** = Timer2 açma-kapama bit'i

1: Timer2 açık
0: Timer2 kapalı

Bit 1-0 **T2CKPS1-T2CKPS2** = Timer2 bölme (prescaler) değeri seçme bit'leri

00 = Bölme değeri 1
01 = Bölme değeri 4
1x = Bölme değeri 16

R= Okunabilir bit, **W**= İlgili bit'e bilgi yazılabilir, **U**= Kullanılmayan bit, **-0**= İlk enerji geldiğinde "0" olarak okunur, **x**= Bit değerinin 1 veya 0 olması önemli değil, anlamındadır.

Timer2 Kesmesi(#INT_TIMER2)

- TMR2 kaydedicisi içindeki değer 00h'tan başlayarak sayar ve PR2 kaydedicisi içine yazılan değere eşit olduğunda bir eşitlik sinyali üretilir. Daha önceden belirlenen postscale sayısı kadar eşitlik sinyali oluştuğunda Timer2 kesmesi oluşur. Timer2 kesmesine izin verme bit'i PIE1 kaydedicisinin 1.bit'i olan TMR2IE bit'idir. Kesme oluşma süresi ve kesme frekansı aşağıda verilen formül ile hesaplanır.
- TKesme =Tosc x (Timer2 bölme oranı) x (PR2 degeri+1) x (Postscaler değeri)

- **SETUP_TIMER_2 ( ) Fonksiyonu**
- Bu komut Timer2 zamanlayıcı ayarlarını yapmaya yarayan bir fonksiyondur.

  setup_timer_2 (mod, periyod, postscale);
- Fonksiyondaki "mod" kısmına aşağıda belirtilen sabit tanımlamalardan biri yazılır. Bu fonksiyonda kullanılacak sabitler denetleyicinin başlık dosyasında (Örneğin, 16F877.h) yazılıdır.
- **MOD Kısmı Tanımlamaları;** "T2_DISABLED" = Timer2 kapalı demek.
- **Timer2 Zamanlayıcı Bölme Oranları (Prescaler) Sabitleri;** "T2_DIV _BY_1, T2_DIV _BY_4, T2_DIV _BY_16 " Sabitin sonundaki rakamlar bölme oranını belirtir.
- **Periyod Kısmı:** Periyot kısmına 0 ile 255 arasında bir tamsayı girilir. Zamanlayıcı periyod değerine ulaşınca kendini sıfırlar. PR2 değerini temsil eder.
- **Postscale Kısmı:** Postscale kısmına 1 ile 16 arasında bir tamsayı girilir. Postscale değeri, Timer2 kesmesi oluşması için Timer2'nin kendisini kaç kez sıfırlaması gerektiğini bildirir.
- **Örnek;** setup_timer_2 (T2_DIV _BY_16 , 192, 3);

komutu ile Timer2 biriminin bölme oranı 16, PR2 kaydedicisi değeri (periyot) 192 ve postscale değeri 3 olarak ayarlanıyor.

SET_TIMER2() Fonksiyonu

- Bu komut Timer2 biriminin saymaya başlama değerini belirlemek için kullanılır. Tüm sayıcılarda olduğu gibi Timer2'de yukarı doğru sayar.
- **set_timer2 (deger);**
- Fonksiyonda "değer" kısmına Timer2 biriminin saymaya başlama değeri yazılır. Bu değer TMR2 kaydedicisi içeriğidir. Timer2 birimi 8 bitlik olduğu için buraya yazılacak değer 0 ile 255 arasında olmalıdır.
- Örnek;

```
set_timer2(80);
```

komutu ile Timer2 birimi saymaya 80'den itibaren baslar.

WATCHDOG TIMER (WDT) BİRİMİ

- WDT birimi önceden belirtildiği gibi denetleyici içinde gömülü bulunan ve ek olarak dışarıdan hiçbir elemanın bağlantısına gerek duymayan bir R/C osilatördür. Denetleyici osilatör girişlerine bağlanan (OSC1/CLKIN ve OSC2/CLKOUT) osilatörden bağımsızdır. Normalde denetleyicide osilatör bağlanmamış olsa veya denetleyici uyku (sleep) modunda olsa dahi WDT çalışmaya devam eder. WDT birimi 8 bit'lik bir zamanlayıcıdır. Bu nedenle WDT 0'dan 255'e kadar sayar ve 255'den sonra tekrar 0'a döndüğünde denetleyiciyi sıfırlar. WDT'in taşma sinyali üretme süresi yaklaşık 18 ms'dir. Bu değer ortam sıcaklığı, kullanılan denetleyici çeşidi ve Vdd değerlerine göre değişir. WDT taşma süresi arttırılmak isteniyorsa OPTION kaydedicinde bölme oranı değeri (prescaler) değiştirilir ve bu değerin WDT için geçerli olmasını sağlayan kaydedicinin 3. bit'i olan PSA bit'i "1" yapılır. Bu şekilde WDT taşma süresi en fazla yaklaşık 2,304 saniye olabilir.

WATCHDOG TIMER (WDT) BİRİMİ

- WDT normal çalışma esnasında belirlenen süre sonunda denetleyiciyi sıfırlar (resetler). Eğer denetleyici uyku (sleep) modunda iken WDT taşması meydana gelirse denetleyici uyku modundan çıkar ve programı kaldığı yerden icra etmeye devam eder.
- WDT birimini açık (enable) veya kapalı (disable) yapmak için programlamada konfigürasyon bit'lerinde bunun belirtilmesi gerekir. Konfigürasyon bit'leri ile seçilen WDT birimi açık veya kapalı durumu yazılım aşamasında program komutlarıyla değiştirilemez. Yani WDT konfigürasyon bit'lerinde açık olarak belirtilmişse, denetleyiciye yüklenen program kodları WDT birimini kapatamaz, programla sadece WDT'in içeriği sıfırlanabilir.

WATCHDOG TIMER (WDT) BİRİMİ

- CCS C'de PIC12, PIC14 ve PIC16 serisi denetleyiciler için WDT birimi açma ve kapatma işlemleri programın başında belirtilen şu komutlarla sağlanır.
- #FUSE WDT //WDT birimi açık
- #FUSE NOWDT //WDT birimi kapalı

CCS C'de PIC18 serisi denetleyiciler için WDT birimini açma ve kapatmak için ise şu komutlar kullanılır.

- setup_wdt (WDT_ON); // WDT birimi açık
- setup_wdt (WDT_ OFF); // WDT birimi kapalı
- WDT birimi kullanıldığında istenmeyen zamanlarda PIC'in resetlenmesini önlemek için programda gerekli yerlerde mutlaka WDT'ı sıfırlayan komut kullanılmalıdır.

SETUP _WDT ( ) Fonksiyonu

- Watchdog Timer ayarları için kullanılan bir fonksiyondur.
- setup_wdt (mod);
- Fonksiyonda parantez içine WDT taşma süresini belirleyen sabit tanımlamalar yazılır.

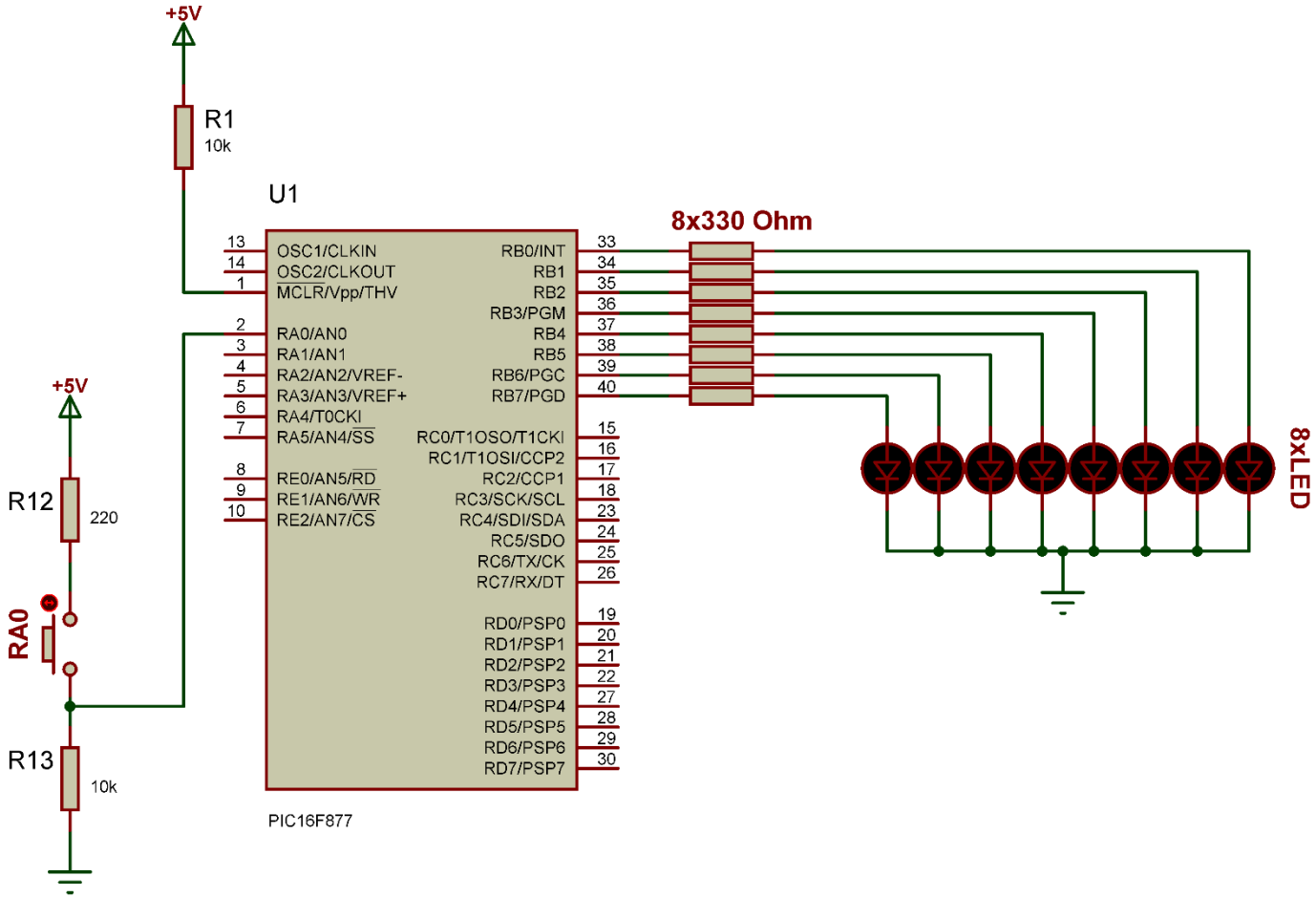
  Bu fonksiyonda ''mod'‘ kısmına şu bölme sabitleri (prescaler) yazılabilir. Bu bölme değerleri (WDT tasma süresini belirleyen sabitler) 12 bit ve 14 bit PIC denetleyiciler için geçerlidir.

12 bit ve 14 bit PIC denetleyiciler için kullanılan sabitler:

- WDT_18MS = WDT tasma süresi 18 ms.
- WDT_36MS = WDT tasma süresi 36 ms.
- WDT_72MS = WDT tasma süresi 72 ms.
- WDT_144MS = WDT tasma süresi 144 ms.
- WDT_288MS = WDT tasma süresi 288 ms.
- WDT_576MS = WDT tasma süresi 576 ms.
- WDT_1152MS = WDT tasma süresi 1152 ms (1.152 saniye).
- WDT_2304MS = WDT tasma süresi 2304 ms (2.304 saniye).

- **RESTART_WDT( ) Fonksiyonu**
- WDT'i sıfırlamak (resetlemek) için kullanılan bir fonksiyondur.
- restart_wdt ( );

Örnek Programlar (WDT uygulaması)



Örnek Programlar (WDT uygulaması)

```
1  #include <16f877.h>
2
3  #fuses XT,WDT,NOPROTECT,NOBROWNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPD
4
5  #use delay (clock=4000000)
6
7  #use fast_io(a) //Port yönlendirme komutları A portu için geçerli
8  #use fast_io(b) //Port yönlendirme komutları B portu için geçerli
9
10 /********* ANA PROGRAM FONKSİYONU********/
11 void main ( )
12 {
13    setup_psp(PSP_DISABLED);        // PSP birimi devre dışı
14    setup_timer_1(T1_DISABLED);     // T1 birimi devre dışı
15    setup_adc_ports(NO_ANALOGS);    // ANALOG giriş yok
16    setup_adc(ADC_OFF);             // ADC birimi devre dışı
17    setup_CCP1(CCP_OFF);            // CCP1 birimi devre dışı
18    setup_CCP2(CCP_OFF);            // CCP2 birimi devre dışı
19
```

Örnek Programlar (WDT uygulaması)

```
19
20      set_tris_a(0x01);    // RA0 pini giriş
21      set_tris_b(0x00);    // B portu komple çıkış
22
23      output_b(0x00);      // B portu çıkışı ilk anda sıfırlanıyor
24
25      setup_WDT(WDT_2304MS); // WDT ayarları yapılıyor
26
27      while(1)     // Sonsuz döngü
28      {
29         output_toggle(pin_b0); // RB0 ucunun çıkış durumu tersleniyor
30
31         while (input(pin_a0)) // Eğer RA0 girişindeki butona basıldı ise
32            restart_wdt();     // WDT'ı sıfırla
33
34         sleep();   // Denetleyici uyku moduna alınıyor.
35      }
36
37   }
```

Kaynaklar

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