

Mikroişlemcili Sistemler

Ders Sunumları

Timer (Zamanlayıcılar)

Hazırlayan: Arş. Gör. Hakan ÜÇGÜN

Timer Nedir?

- Timer modülleri birçok mikro denetleyicide bulunan ve temel görev zamanlama yapmak olan birimlerdir.
- Zamanlama da kullanılacağı gibi sayıcı olarak da kullanılabilir.
- Belirli bir sayıdan aşağı birer birer düşer yada istenilen noktaya çıkartılabilir. Bu sayma işlemini clock hızına göre yapabildikleri gibi dışarıdan verilen işaretlerle de yönetilebilir.
- İşlemci frekansına bağlı olarak çalışırlar.

Mikroişlemcili Sistemler Dersi 5.11.2018

{ 2 }

Timer Nedir?

- İşlemci içerisine kaydedilen bir süre ile geçen süreyi karşılaştırma,
- Belli bir süre sonunda kesme üretme,
- Sayıcı veya sayaç uygulamaları,
- Dışarıdan gelen pals değerlerini sayma, gibi işlemlerde kullanılmaktadır.

Mikroişlemcili Sistemler Dersi 5.11.2018

{ 3 }

PIC16F877 Timer Özellikleri

- Pic16f877 mikrodenetleyicisi 3 adet zamanlayıcı (timer) birimi bulundurmaktadır.
- **Timer0**
- **Timer1**
- **Timer2**

Mikroişlemcili Sistemler Dersi 5.11.2018

{ 4 }

Timer 0 Modülü

- 8 bit zamanlayıcı/sayıcı
 - Yazılabilir ve okunabilir
 - 8 bit yazılım ile programlanabilir Prescaler
 - Dahili 4 Mhz ya da harici clock sinyali
 - FF'ten 00'a taşma kesmesi
 - Düşen yada yükselen kenar clock sinyali seçimi
- Option register ile ayarlamaları yapılmaktadır.

Mikroişlemli Sistemler Dersi 5.11.2018

5

OPTION REGISTER (DURUM KAYDEDİCİSİ)

- Bu register değişik kontrol bitlerini içeren özel bir register'dır.
- B portunun çıkışlarını pull-up yapan, interrupt sinyalinin tetikleme kenarını seçen, TMR0 ve WDT için frekans bölme sayısını belirten ve TMR0 veya WDT seçme bayrağı bulunduran bir registerdir.

OPTION_REG REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBP	INTEGD	TOCS	TOSE	PSA	PS2	PS1	PS0
bit 7							bit 0

Mikroişlemli Sistemler Dersi 5.11.2018

6

7. RBP : PORTB pull-up aktif etme bayrağı

- 1 : PORTB pull-up' lar pasifleştirilir.
- 0 : PORTB pull-up' lar aktifleştirilir.

6. INTEGD : Interrupt kenarı seçme biti

- 1 : Interruptlar RBO/INT sinyalin yükselen kenarında oluşur
- 0 : Interruptlar RBO/INT sinyalin düşen kenarında oluşur.

5. TOCS : TMR0 clock sinyali kaynağı seçme biti

- 1 : RA4/TOCKI pinden alınan clock sinyali
- 0 : Komut dizimi ile oluşturulan clock (CLK0)

4. TOSE : TMR0 harici clock sinyali kenar seçme biti

- 1 : Düşen kenarda artış
- 0 : Yükselen kenarda artış

Mikroişlemli Sistemler Dersi 5.11.2018

7

3. PSA : Frekans bölücü seçme biti

- 1 : Prescaler (frekans bölme sayısı) değeri WDT için geçerli
- 0 : Prescaler değeri Timer0 için geçerli

2. PS2, 1.PS1, 0.PS0 : Prescaler değeri seçme bitleri

PS2	PS1	PS0	TMR0 Değeri	WDT Değeri
0	0	0	1/2	1/1
0	0	1	1/4	1/2
0	1	0	1/8	1/4
0	1	1	1/16	1/8
1	0	0	1/32	1/16
1	0	1	1/64	1/32
1	1	0	1/128	1/64
1	1	1	1/256	1/128

Mikroişlemli Sistemler Dersi 5.11.2018

8

Timer1 Modülü

- 16 bitlik zamanlayıcı/sayıcı
- Okunabilir ve yazılabilir 2 adet 8-bit yazmaç bulunmaktadır. (TMR1H, TMR1L)
- 8 bitlik programlanabilir prescaler değeri
- Dahili ve harici clock sinyali seçimi
- FFFFh'ten 0000h taşma kesmesi
- CCP modülünden resetleme,
- T1CON register ile ayarlamalar yapılmaktadır.

Mikroişlemci Sistemler Dersi 5.11.2018

9

T1CON Register TIMER1 Kontrol Kaydedicisi

- T1CON registerı timer1'in ayarlamalarının yapıldığı registerdir. Bu registerda sırası ile şu bitler bulunmaktadır;
- 7. ve 6. bitler boştur ve değer yazılamaz.

T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	
bit 7								bit 0

- 5. T1CKPS1 ve 4. T1CKPS0 bitleri prescaler değerinin ayarlandığı bitlerdir. Bu bitlerin aldıkları değerlere göre oluşan prescaler değeri aşağıdaki resimde görülmektedir.

bit 5-4 T1CKPS<1:0>: Timer1 Input Clock Prescale Select bits
 11 = 1:8 Prescale value
 10 = 1:4 Prescale value
 01 = 1:2 Prescale value
 00 = 1:1 Prescale value

Mikroişlemci Sistemler Dersi 5.11.2018

10

3. T1OSCEN: Timer1 osilatör kaynağı yetkilendirme bitidir.

- T1OSCEN 1 ise osilatör kaynağı etkindir,
- 0 ise kullanım dışıdır.

2. T1SYNC: Harici clock kaynağının senkron kontrolünü yapan bittir.

- T1SYNC 1 ise harici kaynak ile Timer1 eşzamanlı çalışmaz,
- 0 ise eşzamanlı bir çalışır.

1. TMR1CS: Timer 1 için clock kaynağının seçildiği bittir.

- TMR1CS 1 ise clock kaynağı olarak RCO/T1OSO/T1CKI pininden gelen sinyalin yükselen kenarları seçilir.
- TMR1CS'nin 0 olması halinde ise dahili clock kaynağı kullanılır.

0. TMR1ON: Timer1'e yetki veren bittir.

- Timer1 kullanılır durumdadır,
- 0 ise kullanım dışıdır.

Mikroişlemci Sistemler Dersi 5.11.2018

11

Timer2 Modülü

- 8 bit zamanlayıcı (TMR2 kaydedici),
- 8 bit periyot kaydedici (PR2),
- Okunabilir ve yazılabilir,
- Yazılım ile programlanabilir prescaler,
- TMR2, PR2 eşlemesinde kesme,
- Saat kaymasını üretmek için TMR2 çıkışının seçimli kullanımı SSP modülü,
- T2CON register ile ayarlamalar yapılmaktadır.

Mikroişlemci Sistemler Dersi 5.11.2018

12

T2CON Register TIMER2 Kontrol Kaydedicisi

- T2CON registeri timer2'in ayarlamalarının yapıldığı registerdir. Bu registerda sırası ile şu bitler bulunmaktadır;

T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

- 7. bit boştur ve değer yazılamaz.
- 6-3 TOUTPS3:TOUTPS0: Timer2'de postscaler ayarlarının yapıldığı bitlerdir.
 - 0000 = 1:1 postscale
 - 0001 = 1:2 postscale
 - 0010 = 1:3 postscale
 -
 -
 -
 - 1111 = 1:16 postscale
- 2.TMR2ON: Timer2' ye yetki veren bittir.
 - 1 = Timer2 is on
 - 0 = Timer2 is off
- 1-0 T2CKPS1:T2CKPS0: Timer2'de prescaler'ın ayarlandığı bitlerdir.
 - 00 = Prescaler is 1
 - 01 = Prescaler is 4
 - 1x = Prescaler is 16

Mikroişlemci Sistemler Dersi 5.11.2018

13

Prescaler Hesaplamaları

Timer 0 için prescaler zamanı hesabı;

- Timer0 8 bitlidir. Yani maksimum tutabileceği sayı değeri 255'tir.

$$\frac{1 \times (256 - \text{TMR0}) \times \text{Prescaler}}{\text{Osilatör frekansı} / 4}$$

- [Online Timer Hesabi](#)

Mikroişlemci Sistemler Dersi 5.11.2018

14

Timer Hesaplamaları

- Biz bu örnekte PIC saat kaynağını kullanacağız ve 4ms'lik bir kesme oluşturacağız.
- Dahili osilatörün frekansı 4MHZ. Fakat PIC işlem yaparken bunun 1/4'ünü kullanır. Bu yüzden benim frekansım 1Mhz oluyor.
- Yani mikrodenetleyici her 1 mikro saniyede (us) bir işlem yapmaktadır.
- Prescaler değerini 32 seçiyorum. Buradan $32 * 1\text{us} = 32\text{us}$ çıkıyor. Yani Timer0 32 mikro saniyede bir sayacak. Bizim istediğimiz süre 4 ms olduğuna göre Timer'ın kaç kez saydıktan sonra kesme oluşturacağını bulalım.
- $4000/32=125$ kez sayması gerekiyor. Timer255'ten 256'ya geçerken kesme oluşturduğundan $256 - 125 = 131$ çıkar.
- Yani biz TMR0 registerına 131 değerini yükleyeceğiz. Böylece TMR0, 131den saymaya başlayacak ve 256'ya gelince kesme oluşacak. Kesme oluştuğunda TMR0 içeriği 0 olur ve Timer0 Kesme bayrağı 1 olur(TOIF=1).
- Bizim tekrar kesme oluşturabilmemiz için, kesme altprogramı içinde bu bayrağı sıfırlamamız ve TMR0 içerisine tekrardan 131 değerini girmemiz gerekir.

Mikroişlemci Sistemler Dersi 5.11.2018

15

MikroC Timer Calculator

Timer Calculator

1 Select device: PIC16

2 MCU clock frequency: 4 MHz

3 Choose timer: Timer0

4 Interrupt time: 1 ms

Calculate

```
//Place/Copy this part in declaration section
void InitTimer0() {
    OPTION_REG = 0x82;
    TMR0      = 6;
    INTCON    = 0x20;
}

void Interrupt() {
    if (TMR0IF_bit) {
        TMR0IF_bit = 0;
        TMR0      = 6;
    }
    //Enter your code here
}
```

mikroc / mikroBasic / mikroPascal

Loaded File: PIC16_T0_8MHz_1ms

Mikroişlemci Sistemler Dersi 5.11.2018

16

MikroC Timer Örnekleri

```

unsigned cnt;

void interrupt() {
    if (TMR0IF_bit) {
        cnt++;           // increment counter
        TMR0IF_bit = 0; // clear TMR0IF
        TMRO = 6;
    }
}

void main() {
    OPTION_REG = 0x82; // Assign prescaler to TMR0
    ANSEL = 0;        // Configure AN pins as digital
    ANSELH = 0;
    C1ON_bit = 0;     // Disable comparators
    C2ON_bit = 0;
    TRISB = 0;        // PORTB is output
    PORTB = 0xFF;     // Initialize PORTB
    TMRO = 6;         // Timer0 initial value
    INTCON = 0xA0;    // Enable TMR0 interrupt
    cnt = 0;          // Initialize cnt

    do {
        if (cnt >= 1000) {
            PORTB = ~PORTB; // Toggle PORTB LEDs
            cnt = 0;        // Reset cnt
        }
    } while(1);
}

```

17

Mikroİşlemcili Sistemler Dersi 5.11.2018

MikroC Timer Örnekleri

```

unsigned short cnt;

void interrupt() {
    if (TMR1IF_bit) {
        cnt++;           // increment counter
        TMR1IF_bit = 0; // clear TMR1IF
        TMR1H = 0x80;
        TMR1L = 0x00;
    }
}

void main() {
    ANSEL = 0;        // Configure AN pins as digital
    ANSELH = 0;
    C1ON_bit = 0;     // Disable comparators
    C2ON_bit = 0;
    PORTB = 0xFF;     // Initialize PORTB
    TRISB = 0;        // PORTB is output
    T1CON = 1;        // Timer1 settings
    TMR1IF_bit = 0;   // clear TMR1IF
    TMR1H = 0x80;     // Initialize Timer1 register
    TMR1L = 0x00;
    TMR1IE_bit = 1;   // enable Timer1 interrupt

    cnt = 0;          // initialize cnt
    INTCON = 0xC0;    // Set GIE, PEIE

    do {
        if (cnt >= 72) {
            PORTB = ~PORTB; // Toggle PORTB LEDs
            cnt = 0;        // Reset cnt
        }
    } while(1);
}

```

18

Mikroİşlemcili Sistemler Dersi 5.11.2018

Watchdog Timer

- Watchdog, Türkçe karşılığı bekçi köpeği anlamına gelmektedir.
- Peki mikrodenetleyicideki amacı nedir?
 - Mikrodenetleyici harici sebeplerden veya kodlardaki bir hata sebebiyle kilitlenebilir.
 - Mikrodenetleyici kilitlendiğinde yürüttüğü işlemleri durdurur.
 - Bu tür durumlarda mikrodenetleyicinin yeniden başlatılması veya resetlenmesi gereklidir.
 - İşte Watchdog timerlar burada devreye girerler.
- Watchdog timerlar belirlenen bir süre sonunda sıfırlanırlar ve işlemciye reset atırırlar.

19

Mikroİşlemcili Sistemler Dersi 5.11.2018

SORULARINIZ ?

Geri Bildirim ve Tavsiyeleriniz...



20

Mikroİşlemcili Sistemler Dersi 5.11.2018