

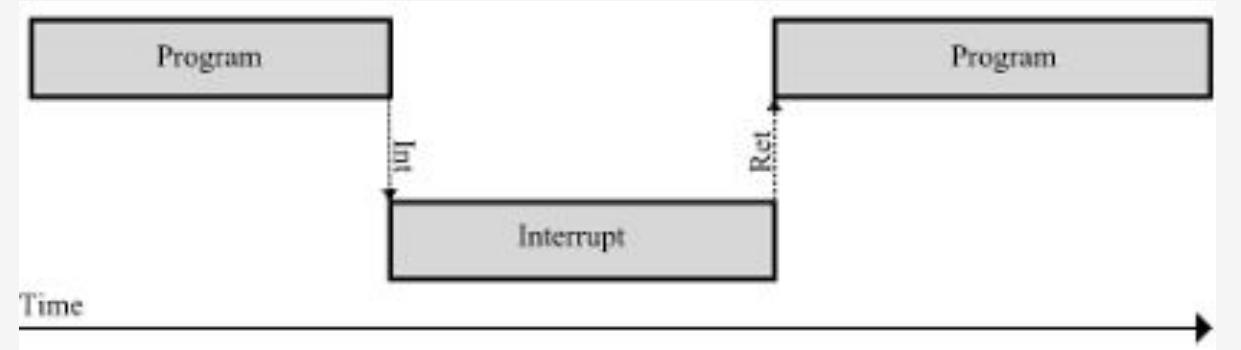
Mikroişlemciler / Mikrodenetleyiciler

Ders-3

Kesmeler (Interrupts)

Kesme

- Program rutin akışı sırasında meydana gelen *tanımlı sinyallerce* ana akışın durdurularak daha önemli olan bir başka işlevin/ işlevlerin yerine getirilmesi «kesme (interrupts) prosedürü» olarak adlandırılır
- Günümüz pek çok MCU 'su bu desteği sunmaktadır
- Pek çok işlev kesme prosedürlerini yerine getirerek daha optimize bir şekilde MCU 'yu kullanabilir
- Burada önemli olan kesmelerin doğru tanımlanması ve kesme prosedürü temel adımlarının doğru yerine getirilmesidir



Kesme Register 'ı – INTCON

- PIC16F877A 'da temel interrupt register 'ı INTCON 'dur
- Bu register üzerinden tüm kesmeler aktif veya pasif yapılabilir
- Çevre birim tanımlı kesmeler aktif veya pasif edilebilir
- Basic interrupts (TMR0 / RB0 / RB4/7) tanımlı kesmelere ait ayarlar yapılabilir

INTCON register

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

GIE enable/disable unmasked interrupts
PEIE enable/disable unmasked peripheral interrupts
TOIE enable/disable Tmr0 interrupts
INTE enable/disable RB0/Int interrupt
RBIE enable/disable RB port change interrupt
TOIF Tmr0 register overflow
INTF RB0/Int interrupt occurred
RBIF one of the PORTB (rb4...rb7) has changed state

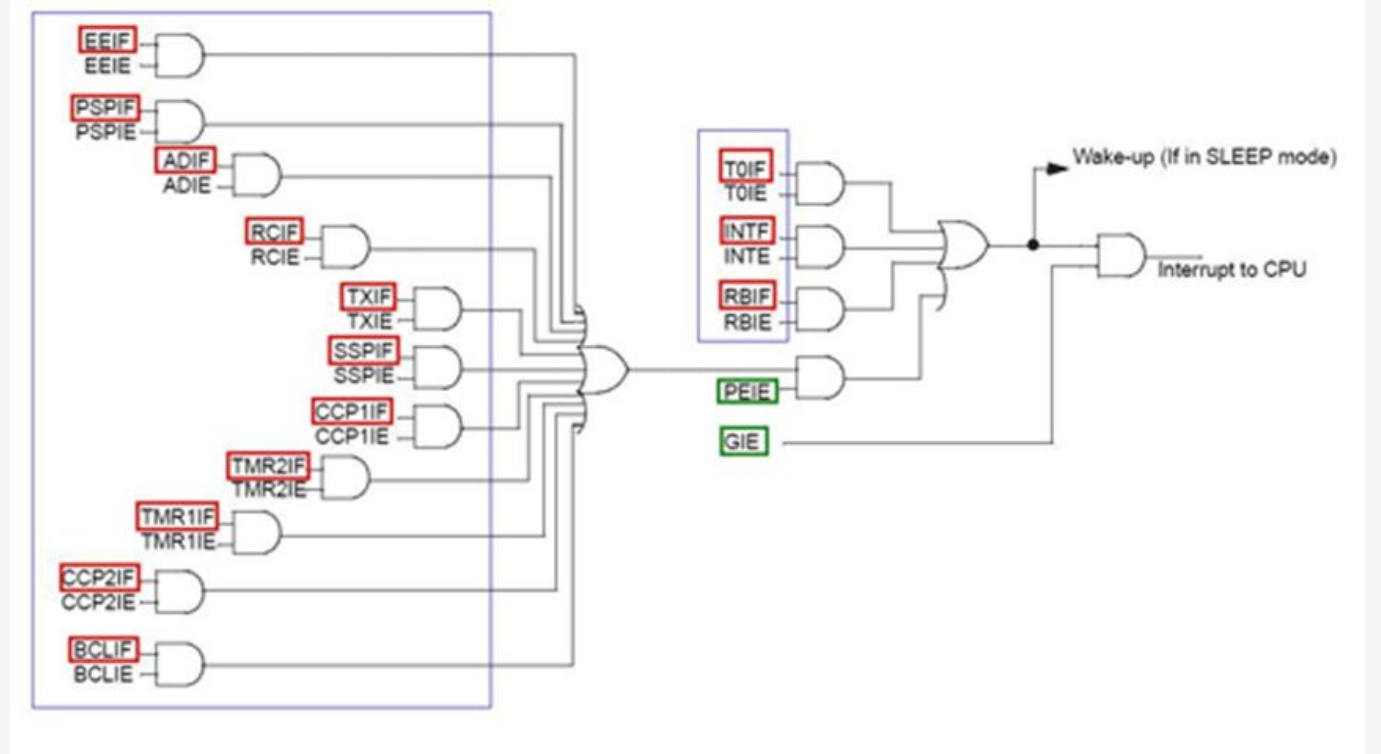
OPTION_REG register

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	INTEDG						

INTEDG rising edge or falling edge

PIC16F877A Kesmeleri

- Bu derste PIC16F877A kesmeleri genelde iki başlık altında ifade edilir;
- Basic Interrupts (BIs)
- Peripheral Interrupts (PIs)



BASIC INTERRUPTS (BIs)

- Üç tip BIs vardır
 - Timer 0 'ın (TMR0) 255 'i aşp tekrar 0 dan başlaması (Temsili sembolü «**TMR0I**» veya «**TOI**»)
 - RBO 'a gelen sinyalin durum değıştirmesi (Yükselen veya düşen kenar kesmesi) (Temsili sembol «**INT**»)
 - RB4/5/6/7 a gelen sinyallerden birinin durumunu değıştirmesi (temsili sembolü «**RBI**»)
- Her bir BIs 'ın bir tane aktif/pasif yapma biti «**E**», bir de o an kesmenin oluşup oluşmadığını gösteren bayrak biti «**F**» vardır

INTCON register

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

GIE enable/disable unmasked interrupts
PEIE enable/disable unmasked periferal interrupts
TOIE enable/disable Tmr0 interrupts
INTE enable/disable RB0/Int interrupt
RBIE enable/disable RB port change interrupt
TOIF Tmr0 register overflow
INTF RB0/Int interrupt occurred
RBIF one of the PORTB (rb4...rb7) has changed state

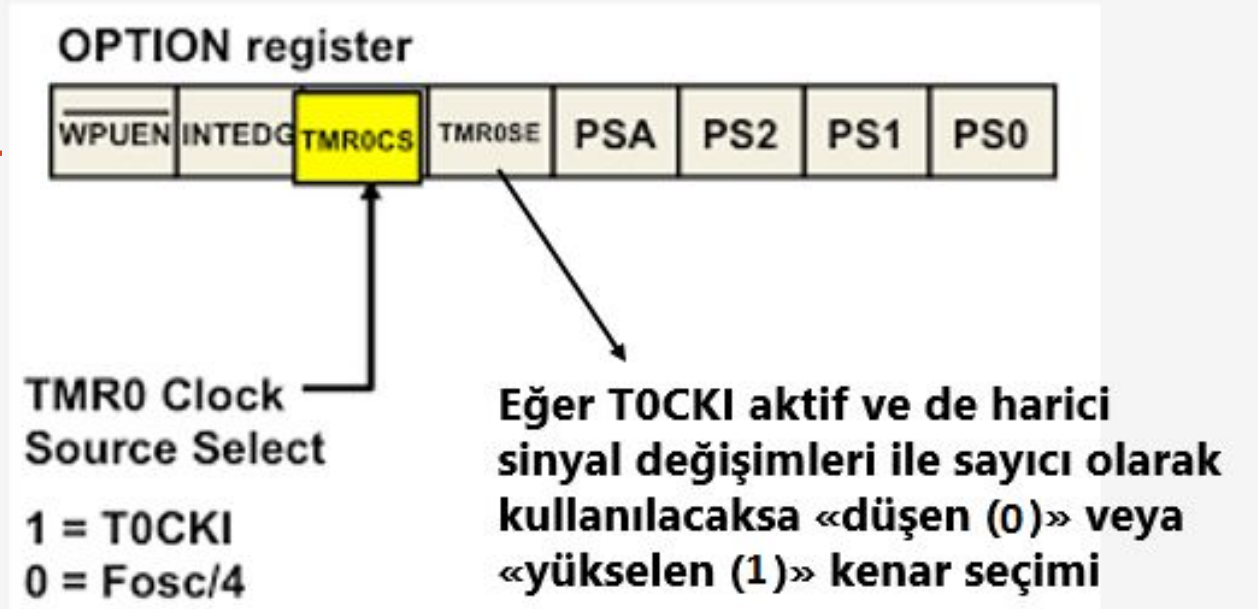
OPTION_REG register

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	INTEDG						

INTEDG rising edge or falling edge

TMRO/TOI Kesmeleri

- **TMRO**, «8 bitlik» bir «counter/timer» register 'dır
- Saat vuruşlarını ya sistemden (Ossilatör Frekans/4) ya da RA4 'den alınan harici bir devreden alabilir (Bunun ayarı **OPTION** register 'ından yapılır)
- «Prescaler» özelliğine sahiptir. Bu özellik «**OPTION**» register 'ından ayarlanır
 - $PSA=0$ ise TMRO, «1» ise WDT
 - $PS0/PS1/PS2$ (1:2, 1:4, 1:8,1:128 ölçekleri için)
- TMRO, istenilen bir değerden başlatılabilir
- Kesme oluşumu 255 'i aşınca meydana gelir (Etkilediği bayrak **INTCON** 'nun **TMROIF** 'sidir
- Sleep modda pasiftir



PS2/1/0

Bit Value	TMRO Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

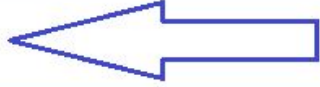
Kesme Durumlarında Yapılması Gereken Bazı İşlemler


w_temp	EQU	0x7D
status_temp	EQU	0x7E
pclath_temp	EQU	0x7F

Shadowed

ORG 0x004  ; kesme vektörü

movwf w_temp ; W'nin yedegini al
movf STATUS,w ; Status'un yedegini almak için onu önce W'ya al
movwf status_temp ; Status'u yedek register'ına al
movf PCLATH,w ; PCLATH'ı yedeklemek için onu önce W'ya al
movwf pclath_temp ; PCLATH'ı yedek register'a al

; gerekli kodlar 

movf pclath_temp,w ; Geri donmeden önce tüm yedekleri geri yükle
movwf PCLATH
movf status_temp,w
movwf STATUS
swapf w_temp,f
swapf w_temp,w
retfie  ; Kesme'den don

TMRO KESMELERİ İÇİN TEMEL ADIMLAR

- 1- Timer0 için gerekli değişkenleri tanımla
- 2- OPTION_REG 'de PSA 'yı TMRO için ayarla
- 3-OPTION_REG 'de istenilen Prescaler değerini set et
- 4- Eğer bir port kullanılacaksa ilgili portu TRIS registeri ile ayarla
- 5- Timer için uygun değeri yükle
- 6- TMRO ve GIE 'i aktif et

```
TUR_DEGISKENI1 EQU 0x20
TUR_DEGISKENI2 EQU 0x21
```

ON_AYARLAR

```
BANKSEL OPTION_REG
```

```
MOVLW b'00000010'
```

```
MOVWF OPTION_REG
```

```
CLRF TRISB
```

```
BANKSEL PORTB
```

```
CLRF PORTB
```

```
MOVLW 0x82
```

```
MOVWF TMRO
```

```
MOVLW d'250'
```

```
MOVWF TUR_DEGISKENI1
```

```
MOVLW d'4'
```

```
MOVWF TUR_DEGISKENI2
```

```
BSF INTCON, TOIE
```

```
BSF INTCON, GIE
```

```
RETURN
```


TMRO Kesmesi

- Kesmenin oluşması durumunda yapılacak işlemlerin yazılacak olan bir alt program tarafından yürütülmesi kodlama tekniği bakımından avantajlı olacaktır
- Örnek videodaki örneği önce izleyiniz, sonra kendiniz bakmadan yapmaya çalışınız. Debug işlemi ile sizin belirlediğiniz register 'lardaki değişimleri izleyiniz. Oluşan Hex kodunu Proteus veya benzeri bir simülatörde deneyiniz

```
BIR_SN_OLDU
    BANKSEL PORTB

    MOVLW    0x01
    BTFSC    PORTB, 0
    CLRW
    MOVWF    PORTB

    RETURN
```

RBO Kesmesi

- PORTB 'nin 0 nolu pininde meydana gelen düşen veya yükselen kenar tetiklemesi sonucu oluşan BIs kesmesidir
- INTCON 'da INTE ve INTF bitleri ile yönetilir
- Aynı zamanda OPTION_REG 'de INTEDG ile düşen(0) veya yükselen(1) kenar tetiklemesi ayarlanmalıdır
- Bir diğer önemli nokta kesme alt programına dallanma gerçekleştiğinde INTF =0 yapılmalıdır çünkü aksi takdirde RETFIE 'den sonra tekrar kesme rutini çalışacaktır
- PORTB 'nin 0 nolu pinin giriş yapılması unutulmamalıdır

ON_AYARLAR

CLRF PORTB

MOVLW b'10010000'

MOVWF INTCON

BANKSEL OPTION_REG

BCF OPTION_REG, INTEDG

MOVLW b'00000001'

MOVWF TRISB

BCF STATUS, RP0

MOVLW 0xF0

MOVWF PORTB

RETURN

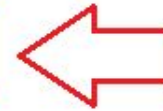
RBO BIs Kesmesi

```
ORG      0x004

movwf    w_temp
movf     STATUS,w
movwf    status_temp
movf     PCLATH,w
movwf    pclath_temp

BCF      INTCON,INTF
MOVLW    b'01100000'
BTFSS    PORTB,7
MOVLW    b'11110000'
MOVWF    PORTB

movf     pclath_temp,w
movwf    PCLATH
movf     status_temp,w
movwf    STATUS
swapf    w_temp,f
swapf    w_temp,w
retfie
```



TMR1 ÇEVRESEL KESME

- Timer 1 kesmesi için üç temel register vardır
 - T1CON register (TIMER1 ayarları)
 - PIE1 register (Aktif/Pasif)
 - PIR1 register (Flag Kontrol)

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	$\overline{T1SYNC}$	TMR1CS	TMR1ON
bit 7		bit 0					

bit 7:6 **Unimplemented:** Read as '0'

bit 5:4 **T1CKPS1:T1CKPS0:** 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

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit

1 = Oscillator is enabled
0 = Oscillator is shut off. The oscillator inverter and feedback resistor are turned off to eliminate power drain

bit 2 **$\overline{T1SYNC}$:** Timer1 External Clock Input Synchronization Select bit

When TMR1CS = 1:

1 = Do not synchronize external clock input
0 = Synchronize external clock input

When TMR1CS = 0:

This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit

1 = External clock from pin T1OSO/T1CKI (on the rising edge)
0 = Internal clock ($F_{osc}/4$)

bit 0 **TMR1ON:** Timer1 On bit

1 = Enables Timer1
0 = Stops Timer1

PIR1: PERIPHERAL INTERRUPT REQUEST REGISTER 1

R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0
EEIF	ADIF	RCIF	C2IF	C1IF	OSFIF	TXIF	TMR1IF
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

- bit 7 **EEIF**: EEPROM Write Operation Interrupt Flag bit
 1 = The write operation completed (must be cleared in software)
 0 = The write operation has not completed or has not been started
- bit 6 **ADIF**: A/D Converter Interrupt Flag bit
 1 = A/D conversion complete (must be cleared in software)
 0 = A/D conversion has not completed or has not been started
- bit 5 **RCIF**: EUSART Receive Interrupt Flag bit
 1 = The EUSART receive buffer is full (cleared by reading RCREG)
 0 = The EUSART receive buffer is not full
- bit 4 **C2IF**: Comparator C2 Interrupt Flag bit
 1 = Comparator output (C2OUT bit) has changed (must be cleared in software)
 0 = Comparator output (C2OUT bit) has not changed
- bit 3 **C1IF**: Comparator C1 Interrupt Flag bit
 1 = Comparator output (C1OUT bit) has changed (must be cleared in software)
 0 = Comparator output (C1OUT bit) has not changed
- bit 2 **OSFIF**: Oscillator Fail Interrupt Flag bit
 1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)
 0 = System clock operating
- bit 1 **TXIF**: EUSART Transmit Interrupt Flag bit
 1 = The EUSART transmit buffer is empty (cleared by writing to TXREG)
 0 = The EUSART transmit buffer is full
- bit 0 **TMR1IF**: Timer1 Overflow Interrupt Flag bit
 1 = The TMR1 register overflowed (must be cleared in software)
 0 = The TMR1 register did not overflow

PSPIE⁽²⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE

PIE1 Register

```
,  
ON_AYARLAR
```

```
    MOVLW    b'11000000'    ;GLOBAL VE ÇEVRESEL KESMELEK AKTİF  
    MOVWF    INTCON  
    MOVLW    b'11110001'  
    MOVWF    T1CON  
    BANKSEL  P1E1            ;TM1 P1E1 DEN AKTİF YAPILIR P1E1 DAN KONTROL EDİLİR  
    BSF      P1E1,0  
  
    CLRF     TRISB  
    BCF      STATUS,RP0  
  
    RETURN
```

```
MOVWF    PORTB,0  
;PORTB 0 YERİNE 00000000  
  
BCF      P1E1,TMR1IF        ;P1E1 DE TMR1 BAYRAĞINI İNDİR  
  
MOVLW    0x01  
BTFSC    PORTB,0  
MOVLW    0x00  
MOVWF    PORTB
```