



T.C
FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
EET-302 MİKROİŞLEMCİLER LABORATUVARI
DENEY FÖYÜ

Doç. Dr. Melih Cevdet İNCE

DENEYLER

Deney_1: Program yazma, derleme, pic'e yükleme, Port okuma yazma, pic16f84a ve pic16f877a da aynı programı çalıştırmak için gerekenler, döngü kavramı, 7 parçalı göstergeye yazma

Deney_2: Program yazma, derleme, pic'e yükleme, 2 tane 7 parçalı göstergeye yazma, tablodan okuma, step motor sürme

Deney_3: 2 tane 7 parçalı gösterge ile 00-99 ileri-geri sayıcı, eksik tamamlamak için serbest çalışma.

Deney_4: ADC(Analog Digital Converter), RB0 Harici kesmesi

Deney_5: TIMER0'ın sayaç olarak kullanılması, TIMER0 Kesmesi

Deney_6: Yazılımla pwm üretilmesi

Deney_7: Yazılımla pwm üretilmesi yeni bir program

NOT:

- ☐ Deneyler her hafta programda belirtilen saatte mesleki yazılım uygulamaları laboratuvarında yapılacaktır, deney başlamadan 15 dk. önce laboratuvarda hazır bulununuz.
- ☐ Deney föyünde bulunan bilgilere, deneye gelmeden önce çalışarak geliniz.
- ☐ Deney föyü olmayanlar laboratuvara alınmayacaktır.
- ☐ Herkes kendi grubunda ve saatinde gelecektir. Kendi grubunda gelmeyenler diğer gruplarla alınmayacaktır.

DENEY-1:MİKRODENETLEYİCİ UYGULAMALARI

Konular: program yazma, derleme, pic'e yükleme, Port okuma yazma, pic16f84a ve pic16f877a da aynı programı çalıştırmak için gerekenler, döngü kavramı,7 parçalı göstereye yazma

UYGULAMA-1 Aşağıdaki programı bilgisayarınızda yeni metin belgesi açarak yazınız. Burada verilen programlar pic16f84a için yazılmıştır. Ancak biz deney setimizdeki hazır kurulu devreyi ve 16f877amikrodenetleyicisini kullanacağız. Böyle basit programlar biraz değişiklikle 16f877a da da çalışır. İlk iki satırı aşağıdaki şekilde değiştiriniz. Böylece yazdığınız program deney setinizdeki 16f877a üzerinde çalışacaktır. Bu iki satır sözde komut olup derleyici içindir.

LIST P=16F877A ;PIC ' in tanıtılması

INCLUDE "P16F877A.INC" ; MPASM klasörü içinde bu dosyayı bulunuz, açınız ne işe yaradığını araştırınız

Yazdığınız programı uyg1.asm adıyla kaydediniz. MPASMWIN programını kullanarak uyg1.asm'yi derleyiniz. İşlemci türünü 16f877a olarak seçmeyi unutmayınız. Hatasız derlendi ise micropro programını kullanarak uyg1.hex dosyasını deney setindeki mikrodenetleyiciye yükleyiniz. Yüklenmesi bittiğinde portb0'a bağlı led yanacaktır. Portb seçme anahtarı turuncu dip switch portb-led konumunda olmalıdır.

```

LIST    P=16F84A
INCLUDE    "P16F84A.INC"
;***** AYARLAR *****
        CLRF    PORTB
        BSF     STATUS,5
        CLRF    TRISB
        BCF     STATUS,5
;***** ANA PROGRAM *****
        BSF     PORTB,0
DONGU    GOTO DONGU
        END

```

UYGULAMA-2 Aşağıdaki programı bilgisayarınızda yeni metin belgesi açarak yazınız. Uygulama2 de istenenin yapılması için pic16f877a'da ADCON1 registerine 06H yazılması unutulmamalıdır. Bunu da eklersek program aşağıdaki gibi olur. Öncekinde yaptığınız gibi uzantısı .asm olacak şekilde örneğin uyg2.asm adıyla kaydediniz, MPASM kullanarak bu dosyayı derleyiniz. Micropro.exe yi kullanarak deney setindeki 16f877a' ya yükleyiniz. Bu program RA0 butonuna basıldığında RB0'a bağlı ledi yakar.

```

LIST    P=16F877A
INCLUDE    "P16F877A.INC"
;***** AYARLAR *****
        BSF     STATUS,5
        MOVLW   H'00'           ;PORTb nın 8 pininde çıkış yapıyoruz
        MOVWF   TRISB
        MOVLW   H'06'
        MOVWF   ADCON1         ;PORTA nın digital olması için adcon1 e 06h yazıyoruz
        MOVLW   H'3F'           ;PORTA nın 6 pininde giriş yapıyoruz
        MOVWF   TRISA
        BCF     STATUS,5
;***** ANA PROGRAM *****
        CLRF    PORTB ;PORTB silindi
OKU
        BTFSS   PORTA,0 ; porta nın 0.biti test ediliyor 1 ise alt satır atlanacak
        GOTO    OKU
        BSF     PORTB,0 ; led yakıldı
        GOTO    OKU
        END

```

Pull-down anahtarı down konumunda ise(down olmalı) prog.yükledikten sonra rb0 ledi yanmaz, ra0 a bağlı butona basılırsa rb0 a bağlı led yanar yanık kalır söndürmek için reset butonuna basılmalıdır. Aslında istenen budur.

Pull-down anahtarı pull konumunda ise program yükledikten sonra rb0 ledi yanar, yanık kalır, söndürmek için resetlemek gerekir, resetleyince söner tekrar yanar çünkü pull konumunda anahtar 5 v a çekilmiştir

ÖDEV1:Bu programda gerekli değişiklikleri yaparak RA0'a basılıp basılı tutuldukça RB0'daki ledi yakan, RA0 bırakıldığında RB0'daki ledi söndüren program haline getiriniz. Hemen yapamazsanız araştırıp haftaya deneyiniz.

Ödev1 in cevabı OKU etiketini CLRF satırına almaktır. Pull-down anahtarı down konumundayken RA0'a basılıp basılı tutulduğunda CLRF PORTB satırına her gelindiğinde portb çok kısa süre için sönecektir ancak süre çok kısa olduğundan bu durum gözlenemeyecektir. Bu kısa süreyi hesaplayınız.

Başka bir çözüm ana programı aşağıdaki gibi yazmak olabilir. Deneyiniz. Başka çözümler üretiniz.

```
;***** ANA PROGRAM *****
BAS    CLRF    PORTB
OKU

    BTFSS    PORTA,0
    GOTO    OKU
    BSF      PORTB,0
    BTFSC    PORTA,0
    GOTO    OKU
    GOTO    BAS
END
```

UYGULAMA-3 Bu program RA0'a bağlı butona **basılıp bırakılınca** PORTB'ye bağlı 8 led de ikilik tabanlı sayıları 1'er arttırır. BASILIP BIRAKILINCA kısmını inceleyiniz. PULL VE DOWN konumunda inceleyiniz. En son sayı FFh artırılınca kaç oluyor.

```
LIST    P=16F877A    ; LIST P=16F84A
INCLUDE    "P16F877A.INC"; INCLUDE    "P16F84A.INC"
;***** AYARLAR *****
    BSF      STATUS,5
    MOVLW    H'00'    ;PORTb nın 8 pininde çıkış yapıyoruz
    MOVWF    TRISB
    MOVLW    H'06'
    MOVWF    ADCON1;PORTA nın digital olması için adcon1 e 06h yazıyoruz
    MOVLW    H'01'    ;PORTA nın 0 pinini giriş yapıyoruz
    MOVWF    TRISA
    BCF      STATUS,5
;***** ANA PROGRAM *****
    CLRF    PORTB
TEST

    BTFSS    PORTA,0
    GOTO    TEST
OKU

    BTFSC    PORTA,0
    GOTO    OKU
    INCF    PORTB
    GOTO    TEST
END
```

UYGULAMA-4 Uygulama-3 'deki buton yerine gecikme programı kullanarak portb'ye bağlı 8 ledi 1 saniye aralıklarla arttıran program. Yukardakiler gibi derleyiniz, yükleyiniz. Gecikme programının ne kadar zaman tükettiğini hesaplamayı araştırınız. Gecikmeyi alt program olarak düzenlemeyi CALL ve RETURN komutlarını kullanmayı öğreniniz.

```

LIST    P=16F877A
INCLUDE "P16F877A.INC"
;***** AYARLAR *****
SAYAC1    EQU H'20'
SAYAC2    EQU H'21'
SAYAC3    EQU H'22'
BSF    STATUS,5
MOVLW    H'00'    ;PORTb nın 8 pininide çıkış yapıyoruz
MOVWF    TRISB
BCF    STATUS,5
;***** ANA PROGRAM *****
CLRF    PORTB
TEKRAR
INCF    PORTB
MOVLW    H'45'
MOVWF    SAYAC1    ;
DONGU1
MOVLW    H'45'
MOVWF    SAYAC2
DONGU2
MOVLW    H'45'
MOVWF    SAYAC3
DONGU3
DECFSZ    SAYAC3, F
GOTO    DONGU3
DECFSZ    SAYAC2, F
GOTO    DONGU2
DECFSZ    SAYAC1, F
GOTO    DONGU1
GOTO    TEKRAR
END

```

Ödev2: PORTA'yı okuyup PORTB'ye yazan programı yazınız. Hemen yapamazsanız araştırıp haftaya deneyiniz.

```

LIST    P=16F877A
INCLUDE "P16F877A.INC"
;***** AYARLAR *****
BSF    STATUS,5
MOVLW    H'00'    ;PORTb nın 8 pininide çıkış yapıyoruz
MOVWF    TRISB
MOVLW    H'06'
MOVWF    ADCON1    ;PORTA nın digital olması için adcon1 e 06h yazıyoruz
MOVLW    H'3F'    ;PORTA nın 6 pininide giriş yapıyoruz
MOVWF    TRISA
BCF    STATUS,5
;***** ANA PROGRAM *****
DONGU    MOVF    PORTA,W    ;PORTA nın değeri working registere alındı
MOVWF    PORTB    ;w nin değeri portb ye yazıldı
GOTO    DONGU
END

```

UYGULAMA-5 Aşağıdaki program 7 parçalı göstergeye 8 yazar. Pull-down anahtarı pull konumunda olmalı. Başka sayılar yazmak için tablodaki bilgileri kullanarak gerekli değişiklikleri yapınız, deneyiniz.

İleri geri sayıcı yapmayı düşününüz.

;7 PARÇALI GÖSTERGEYE 8 YAZAR PULL KONUMUNDA OLMALI

```

LIST    P=16F877A
INCLUDE "P16F877A.INC"
;***** AYARLAR *****
        CLRF    PORTB      ;PORTB ' ye bağlı LED ' leri söndür
        BSF     STATUS,5   ;BANK1 ' e geç
        CLRF    TRISB      ;PORTB 'nin uçlarını çıkış yap
        MOVLW   H'FF'
        BCF     STATUS,5
;***** ANA PROGRAM *****
        MOVLW   H'7F'      ;W YE 7F YAZ BU SAYI 7PARÇALI GÖSTERGEDE 8 DİR
        MOVWF   PORTB      ;PORTB SEÇME ANAHTARINI 7SEGLERDEN BİRİNE
GETİRİNİZ
DON     GOTO   DON
        END

```

(Çevrilecek kod. Hex sayı)	Çevrilen 7 segment kodu (PORTB'ye)	7 segment uçlarındaki veri	7 segment'te görülecek sayı
h'00'	h'3F'	00111111	0
h'01'	h'06'	00000110	1
h'02'	h'5B'	01011011	2
h'03'	h'4F'	01001111	3
h'04'	h'66'	01100110	4
h'05'	h'6D'	01101101	5
h'06'	h'7D'	01111101	6
h'07'	h'07'	00000111	7
h'08'	h'7F'	01111111	8
h'09'	h'6F'	01101111	9
h'0A'	h'77'	01110111	A
h'0B'	h'7C'	01111100	B
h'0C'	h'39'	00111001	C
h'0D'	h'5E'	01011110	D
h'0E'	h'79'	01111001	E
h'0F'	h'71'	01110001	F
Nokta	h'80'	10000000	.

DENEY-2:MİKRODENETLEYİCİ UYGULAMALARI

Konular: program yazma, derleme, pic'e yükleme, 2 tane7 parçalı göstergeye yazma, tablodan okuma, step motor sürme

UYGULAMA-6 Aşağıdaki programı bilgisayarınızda yeni metin belgesi açarak yazınız. Yazdığınız programı uyg6.asm adıyla kaydediniz. MPASMWIN programını kullanarak uyg6.asm’yi derleyiniz. İşlemci türünü 16f877a olarak seçmeyi unutmayınız. Hatasız derlendi ise micropro26 programını kullanarak uyg6.hex dosyasını deney setindeki mikrodenetleyiciye yükleyiniz. 7 parçalı göstergenin altındaki dip switch’in üstten 2 ve 3. Anahtarlarını sağa çekerek her iki 7 parçalı göstergeninde devrede olmasını sağlayınız. Aşağıdaki bağlantı şemasını inceleyiniz. Deney setimizdeki bu bağlantıya göre PIC16F877A’nın a portunun son iki ucu ra1 ve ra0, 7SEG’in sırasıyla 2 ve 1 uçlarına bağlıdır. ra0=1,ra1=0 ise onlar basamağı, ra0=0,ra1=1 ise birler basamağı seçiliyor.

Yüklenmesi bittiğinde 2 tane7 parçalı göstergede 81 yazacaktır. Aşağıdaki incelemeleri yapınız.

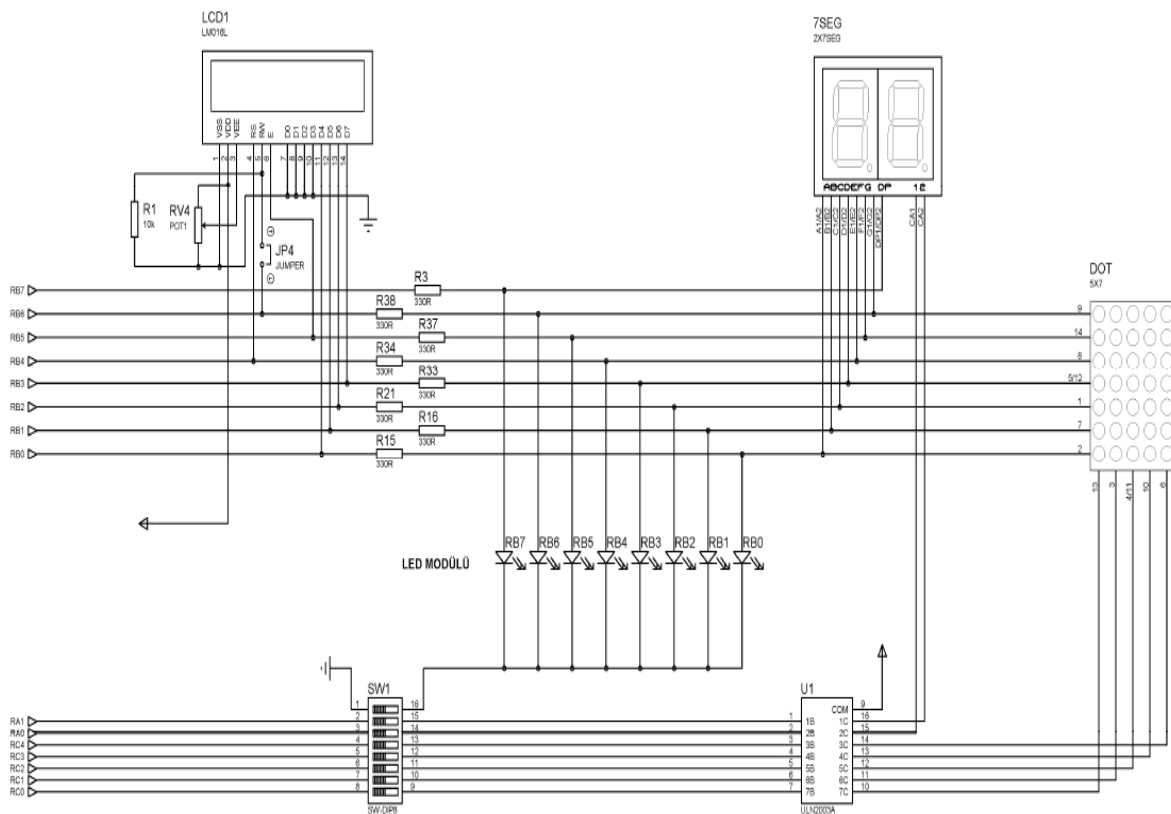
+25 yazmak için gerekli değişikliği yapınız. Gecik alt programındaki sayaçların her birinin değerini 01 yapınız, yeniden derleyiniz ve çalıştırınız. Ne görüyorsunuz. Açıklayınız.

+Gecik alt programındaki sayaçların her birinin değerini 45 yapınız, yeniden derleyiniz ve çalıştırınız. Ne görüyorsunuz. Açıklayınız.

+Bu durumu düzeltmek için uygun bir gecikmeden başka bir yol önerebilir misiniz? 7447 entegresini inceleyiniz.

+İkiden fazla sayıda 7 parçalı gösterge olsaydı ne yapılacaktı. Düşününüz.

+Pull-down anahtarını pull konumuna getiriniz. Ra0 ve ra1 butonlarına sırayla basınız. Ne gözlemlediniz, açıklayınız.



ÖDEV 3: UYG6.ASM ‘yi kullanarak bu programı 00-99 arasında ileri geri sayan program haline getiriniz. RA4’ e bağlı butona her basıldığında sayıyı bir arttıran,RA5’e bağlı butona basıldığında sayıyı bir azaltsın. Alt limiti 00 da üst limiti de 99 da kalsın.

Dönem içinde yapacağınız ödevlerden biri olarak ödev3 ü ve uyg6 hakkında sorulan soruların cevaplarını yazıp getirebilirsiniz.

;uyg6 2 tane 7 parçalı göstergeye 81 yazmak

```

LIST    P=16F877A
INCLUDE "P16F877A.INC"
;***** AYARLAR *****
SAYAC1 EQU H'20'           ;gecik alt programının sayaçları tanımlandı
SAYAC2 EQU H'21'
SAYAC3 EQU H'22'
BCF     STATUS,RP0
BCF     STATUS,RP1         ;BANK0 a geçildi
CLRF    PORTA              ;çıkış veri latchları silinerek porta hazırlandı
CLRF    PORTB              ;çıkış veri latchları silinerek portB hazırlandı
BSF     STATUS,RP0         ;BANK1 seçildi
MOVLW   H'00'              ;PORTb nın 8 pininde çıkış yapıyoruz
MOVWF   TRISB
MOVLW   H'06'
MOVWF   ADCON1             ;PORTA nın digital olması için adcon1 e 06h yazıyoruz
MOVLW   B'00111100'        ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz
MOVWF   TRISA
BCF     STATUS,RP0         ;BANK0 a geçildi
;***** ANA PROGRAM *****
;ra0=1,ra1=0 ise onlar basamağı,ra0=0,ra1=1 ise birler basamağı seçiliyor
DONGU
    MOVLW   b'00000010'
    MOVWF   PORTA
    MOVLW   H'06'
    MOVWF   PORTB          ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı
    CALL    GECİK
    MOVLW   b'00000001'
    MOVWF   PORTA
    MOVLW   H'7F'
    MOVWF   PORTB          ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı
    CALL    GECİK
    GOTO    DONGU
GECİK
    MOVLW   H'0F'
    MOVWF   SAYAC1        ;
DONGU1
    MOVLW   H'0F'
    MOVWF   SAYAC2
DONGU2
    MOVLW   H'0F'
    MOVWF   SAYAC3
DONGU3
    DECFSZ   SAYAC3, F
    GOTO     DONGU3
    DECFSZ   SAYAC2, F
    GOTO     DONGU2
    DECFSZ   SAYAC1, F
    GOTO     DONGU1
    RETURN
END

```

UYGULAMA-7 Aşağıdaki program STEP motoru ra0 a basılınca(basilip bırakılınca 1 adım basılı tutunca sürekli) sağa ra1 e basılınca sola döndürür. Setimizde step motor olmadığı için portb ye bağlı ledlerde çıkışı

izleyeceğiz. dip switchin en üst anahtarı sağa doğru olmalı. Programı adım adım inceleyiniz. Gerekli açıklamalar program satırlarına yazılmıştır

;DENEY SETİNDE PULL-DOWN ANAHTARI DOWN KONUMUNDA OLACAK

;PULL KONUMUNDADA ÇALIŞIR BU ÇALIŞMA DURUMUNUN FARKINI BULUNUZ.

;step motoru sağa ve sola döndüren program

```

LIST    P=16F877A
INCLUDE    "P16F877A.INC"
;***** AYARLAR *****
        SAYAC1      EQU H'20'
        SAYAC2      EQU H'21'
        SOL         EQU H'22'
        SAG         EQU H'23'
        ADIM        EQU H'24'
        BCF         STATUS,RP0
        BCF         STATUS,RP1    ;BANK0 a geçildi
        CLRF        PORTA         ;çıkış veri latchları silinerek porta hazırlandı
        CLRF        PORTB         ;çıkış veri latchları silinerek portB hazırlandı
        BSF         STATUS,RP0    ;BANK1 seçildi
        MOVLW       H'00'         ;PORTb nın 8 pininde çıkış yapıyoruz
        MOVWF       TRISB
        MOVLW       H'06'
        MOVWF       ADCON1        ;PORTA nın digital olması için adcon1 e 06h yazıyoruz
        MOVLW       B'00111111'   ;PORTA nın 6 pini giriş
        MOVWF       TRISA
        BCF         STATUS,RP0    ;BANK0 a geçildi
;***** ANA PROGRAM *****
        MOVLW       h'07'
        MOVWF       ADIM
        MOVLW       B'00000010'
        MOVWF       SAG
        MOVLW       B'00000001'
        MOVWF       SOL
;DENEY SETİNDE PULL-DOWN ANAHTARI DOWN KONUMUNDA OLACAK
        OKU
        MOVF        PORTA, W      ; PORTAnın değeri w registera alındı. Yani porta okundu. Butona basılınca 0
oluyor.
                                ;porta1=1 ve porta0=0, ise sağa; porta1=0 ve porta0=1 ise sola; porta1=1 ve porta0=1
                                ;ise duracak
        ANDLW       B'00000011'   ; üst 6 bit maskelendi.
        XORWF       SAG,W         ;w reg yani porta dan okunan değer SAG ile aynıysa sonuç 0, zero bayrağı
1 olur
        BTFSC      STATUS,Z       ; z bayrağı 0 ise alt satırı atla diğer seçeneğe bak
        GOTO       SAGADON

        MOVF        PORTA, W      ; W deki değer bozuldu porta tekrar okunur
        ANDLW       B'00000011'   ; üst 6 bit maskelendi.
        XORWF       SOL,W         ;w reg yani porta dan okunan değer SOL ile aynıysa sonuç 0, zero bayrağı 1
olur
        BTFSC      STATUS,Z       ; z bayrağı 0 ise alt satırı atla başa git
        GOTO       SOLADON
        GOTO       OKU            ;01 veya 10 dışında ne gelirse gelsin(00 ve 11) porta yı okumaya devam eder
        SAGADON
        INCF        ADIM
        GOTO       DEVAM

```

```

SOLADON
DECF ADIM
DEVAM
MOVF ADIM,W          ; adım değeri w ye alındı
ANDLW B'00000111'    ; adım değeri 7 den büyük olamaz(0 dahil 8 adım). üst 5 bit maskelendi
CALL ADIMTBL
MOVWF PORTB
CALL GECIKME
GOTO OKU              ;butonların durumu değişti mi kontrol edilecek
ADIMTBL
ADDWF PCL,F           ;program sayacının bu andaki değerine w deki değer eklenecek
RETLWB'00000001'      ;w=0 ile gelirse w=01 ile döner
RETLWB'00001001'
RETLWB'00001000'
RETLWB'00001010'
RETLWB'00000010'
RETLWB'00000110'
RETLWB'00000100'
RETLWB'00000101'     ;7.satır,w=07 ile gelirse w=05 ile döner
GECIKME
MOVLW H'FF'
MOVWF SAYAC1
DONGU1
MOVLW H'FF'
MOVWF SAYAC2
DONGU2
DECFSZ SAYAC2,F
GOTO DONGU2
DECFSZ SAYAC1,F
GOTO DONGU1
RETURN
END

```

ÖDEV4. UYGULAMA 7 yi kullanarak bir step motor sürücü ve step motor ekleyerek projeyi tamamlayınız. Dönem içinde yapacağınız ödevlerden biri olarak ödev4 ü ve uyg7 hakkında sorulan soruların cevaplarını yazıp getirebilirsiniz.

DENEY-3:MİKRODENETLEYİCİ UYGULAMALARI

Konular: ödev3 olarak verilen 2 tane7 parçalı gösterge ile 00-99 ileri-geri sayıcı, yeni bir ödev, eksik tamamlamak için serbest çalışma.

ÖDEV 3: UYG6.ASM ‘yi kullanarak bu programı 00-99 arasında ileri geri sayan program haline getiriniz. RA4’ e bağlı butona her basıldığında sayıyı bir arttıran,RA5’e bağlı butona basıldığında sayıyı bir azaltsın. Alt limiti 00 da üst limiti de 99 da kalsın.

;0-99 ileri GERİ sayıcı PORTA4 BASILIP BIRAKILINCA 1 ARTAR.

;PORTA5 E BASILIP BIRAKILINCA 1 AZALIR

;ARTIRMAYA DEVAM EDİLİRSE 99 DA KALIR AZALTMAYA DEVAM EDİLİR 00 DA KALIR

;7447 LİSİ S99ILGER.ASM DİR.O PROGRAM 16F84A İÇİNDİ.SİZE VERİLEN CD DE

BULABİLİRSİNİZ

;DENEY SETİNDE 7 PARÇALI GÖST.ALTINDAKİ DİP SWITCH 2 VE 3. ANAHTARLAR SAĞA ÇEKİLECEK.

;ALTTAKİ PULL-DOWN PULL KONUMUNDA OLACAK

UYGULAMA 8

LIST P=16F877A

INCLUDE "P16F877A.INC"

;***** AYARLAR *****

BIRLER EQU H'20'

ONLAREQU H'21'

SAYAC1 EQU H'22'

SAYAC2 EQU H'23'

SAYAC3 EQU H'24'

BCF STATUS,RP0

BCF STATUS,RP1 ;BANK0 a geçildi

CLRF PORTA ;çıkış veri latchları silinerek porta hazırlandı

CLRF PORTB ;çıkış veri latchları silinerek portB hazırlandı

BSF STATUS,RP0 ;BANK1 seçildi

MOVLW H'00' ;PORTb nın 8 pininide çıkış yapıyoruz

MOVWF TRISB

MOVLW H'06'

MOVWF ADCON1 ;PORTA nın digital olması için adcon1 e 06h yazıyoruz

MOVLW B'00111100' ;PORTA nın 4 pini giriş,ra0 ve ra1 çıkış yapıyoruz

MOVWF TRISA

BCF STATUS,RP0 ;BANK0 a geçildi

;***** ANA PROGRAM *****

CLRF BIRLER

CLRF ONLAR

CLRF PORTB

TEST4

BTFSS PORTA,4 ;BASILI İSE 0 GELMELİ.....PULL KONUMU

GOTO BASIL4

GOTO TEST5

BASIL4BTFSS PORTA,4

GOTO BASIL4

GOTO ARTTIR

TEST5

BTFSS PORTA,5

GOTO BASIL5

GOTO YY

BASIL5BTFSS PORTA,5

GOTO BASIL5

GOTO AZALT

YY

CALL YAZ

```

        GOTO TEST4
ARTTIR
        INCF BIRLER
        MOVLW D'10'
        SUBWFBIRLER,W
        BTFSC STATUS,2
        GOTO DUZEN1
        CALL YAZ
        GOTO TEST4
DUZEN1      CLRF BIRLER
        INCF ONLAR
        MOVLW D'10'
        SUBWFONLAR,W           ;ONLAR BASAMAĞI 10 OLDUYSA TEKRAR 9 YAPIYORUZ
        BTFSC STATUS,2       ;0 DAN 9 A KADARSA YAZDIRIYORUZ
        GOTO DUZEN2           ;BİRLER BASAMAĞI 10 OLDUYSA TEKRAR 9 YAPIYORUZ
        CALL YAZ
        GOTO TEST4
DUZEN2      MOVLW D'9'
        MOVWF BIRLER           ;BİRLER VE ONLAR 99 DA KALIYOR
        MOVWF ONLAR
        CALL YAZ
        GOTO TEST4
;AZALTMA KISMI ARTTIRMA KISMINA BENZER OLARAK
AZALT
        DECF BIRLER
        MOVLW H'FF'
        SUBWFBIRLER,W
        BTFSC STATUS,2
        GOTO DUZE1
        CALL YAZ
        GOTO TEST4
DUZE1      MOVLW D'9'
        MOVWF BIRLER
        DECF ONLAR
        MOVLW H'FF'           ;00 BİR AZALIRSA FF OLUR
        SUBWFONLAR,W         ;ONLAR BASAMAĞI FF OLDUYSA TEKRAR 0 YAPIYORUZ
        BTFSC STATUS,2       ;0 DAN 9 A KADARSA YAZDIRIYORUZ
        GOTO DUZE2           ;BİRLER BASAMAĞI FF OLDUYSA TEKRAR 0 YAPIYORUZ
        CALL YAZ
        GOTO TEST4
DUZE2      MOVLW D'0'
        MOVWF BIRLER           ;BİRLER VE ONLAR 00 DA KALIYOR
        MOVWF ONLAR
        CALL YAZ
        GOTO TEST4
CEVTBL
        ADDWF PCL,F           ;program sayacının bu andaki değerine w deki değer eklenecek
        RETLWH'3F'           ;w=0 ile gelirse w=3F ile döner
        RETLWH'06'
        RETLWH'5B'
        RETLWH'4F'
        RETLWH'66'
        RETLWH'6D'
        RETLWH'7D'
        RETLWH'07'
        RETLWH'7F'
        RETLWH'6F'           ;w=9 ile gelirse w=6F ile döner
;*** YAZ ALT PROGRAMI *****

```

```

YAZ    MOVLW      b'00000010'
        MOVWF      PORTA
        MOVF  BİRLER,W
        CALL  CEVTBL
        MOVWF      PORTB      ;w nin değeri portb ye BİRLER BASAMAĞI yazıldı
        CALL  GECİK
        MOVLW      b'00000001'
        MOVWF      PORTA
        MOVF  ONLAR,W
        CALL  CEVTBL
        MOVWF      PORTB      ;w nin değeri portb ye ONLAR BASAMAĞI yazıldı
        CALL  GECİK
        RETURN

```

```

;*** GECİK ALT PROGRAMI *****

```

```

GECİK
MOVLW      H'15'
MOVWF      SAYAC1
DONGU1
MOVLW      H'15'
MOVWF      SAYAC2
DONGU2
MOVLW      H'05'
MOVWF      SAYAC3
DONGU3
DECFSZ     SAYAC3, F
GOTO  DONGU3
DECFSZ     SAYAC2, F
GOTO  DONGU2
DECFSZ     SAYAC1, F
GOTO  DONGU1
RETURN
END

```

UYGULAMA 9

```

LIST  P=16F877A
INCLUDE "P16F877A.INC"
MOVLW      B'10000001'
MOVWF      ADCON0
BSF  STATUS,RP0
MOVLW      H'00'
MOVWF      TRISB
MOVLW      B'000001110'
MOVWF      ADCON1
BCF  STATUS,RP0

```

```

BASLA

```

```

BSF  ADCON0,GO

```

```

TEST

```

```

BTFSC ADCON0,GO
GOTO  TEST
MOVF  ADRESL,W
MOVWF      PORTB
GOTO  BASLA
END

```

Ödev 5 Yukarıdaki programın ne iş yaptığını 16f877a'nın ADCON0 ve ADCON1 adlı özel amaçlı kaydedicilerini inceleyerek anlayınız. ADRESL nedir?

ADCON0

7	6	5	4	3	2	1	0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADCON
Osilatör seçme		Analog giriş seçme			0=Çevrim bitti 1=Start		0=ADC off 1=ADC on
Bit 7-6	ADCS1, ADCS0: A/D'nin çalışması için osilatör seçme bitleri 00=f _{osc} /2 01=f _{osc} /8 10=f _{osc} /32 11=f _{RC} (RC osilatör)						
Bit 5-3	000=RA0/AN0 100=RA5/AN4 001=RA1/AN1 101=RE0/AN5 010=RA2/AN2 110=RE1/AN6 011=RA3/AN3 111=RE2/AN7						

ADCON1

D7	D6	D5	D4	D3	D2	D1	D0
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0

DENEY-4:MİKRODENETLEYİCİ UYGULAMALARI

Konular: ödev5 olarak verilen UYGULAMA 8 - analog-dijital dönüştürücü, ADC(Analog Digital Converter), RB0 Harici kesmesi,TIMER0 ödevi

UYGULAMA 10

UYGULAMA 9 HATALARI DÜZELTTİM. BU SEÇİMDE ADRESH EN ÖNEMLİ 8 BİTİ PORTB YE YAZIYORUZ.rb0 a bir potansiyometreden analog giriş uygulayınız. pot yoksa rb0 a iki tane en az 1k lık direnç kullanarak 5v ve 2.5v uygulayınız. dirençte yoksa 5v(vcc) uygulayınız

```
LIST    P=16F877A
INCLUDE "P16F877A.INC"
BCF STATUS,RP1
BCF STATUS,RP0
MOVLW    B'10000001'    ; adcon1 deki ads2, 6.bit 0 ve adcon0 daki adcs1=1 ve adsc0 = 0
```

seçildiğinden

; clock fosc/32 seçilmiş oldu. 5.4.ve 3. Bitler 0 seçildiğinden kanal0=an0=ra0

seçildi.bit0=0 a/d dön.kapalı

```
MOVWF    ADCON0          ;BANK0 DA
```

```
BCF STATUS,RP1
```

```
BSF STATUS,RP0          ;BANK1
```

```
MOVLW    H'00'
```

```
MOVWF    TRISB          ;BANK1 DE
```

```
;    MOVLW    B'10001110'    ;ADRESH=000000XX ADRESL=XXXXXXXX ;ALTTAKİNİ
```

KULLANDIK

```
MOVLW    B'00001110'    ;ADRESH=XXXXXXXX ADRESL=XX000000
```

```
MOVWF    ADCON1          ;BANK1 DE
```

```
BCF STATUS,RP1
```

```
BCF STATUS,RP0          ;BANK0 DAYIZ
```

BASLA

```
BSF    ADCON0,GO    ;BANK0 DA,,,analog digital dönüşüm başlasın
```

TEST

```
BTFSC ADCON0,GO    ;BANK0 DA,,, bu bit sıfırda dönüşüm tamamlanmış demektir
```

```
GOTO TEST
```

```
MOVF ADRESH,W    ;BANK0 DA..EN ÖNEMLİ 8 BİTİ GÖSTERİR
```

```
MOVWF PORTB      ;BANK0 DA
```

```
GOTO BASLA
```

```
END
```

ÇOK BENZİYOR KARIŞMASIN DİYE TEKRAR YAZDIM. Farkın ne olduğuna dikkat ediniz.

;UYGULAMA 8 HATALARI DÜZELTTİM BU SEÇİMDE ADRESL EN ÖNEMSİZ 2 BİTİ PORTB YE YAZIYORUZ

```
LIST    P=16F877A
INCLUDE "P16F877A.INC"
BCF STATUS,RP1
BCF STATUS,RP0
MOVLW    B'10000001'    ; adcon1 deki ads2, 6.bit 0 ve adcon0 daki adcs1=1 ve adsc0 = 0
```

seçildiğinden

; clock fosc/32 seçilmiş oldu. 5.4.ve 3. Bitler 0 seçildiğinden kanal0=an0=ra0

seçildi.bit0=0 a/d dön.kapalı

```
MOVWF    ADCON0          ;BANK0 DA
```

```
BCF STATUS,RP1
```

```
BSF STATUS,RP0          ;BANK1
```

```
MOVLW    H'00'
```

```
MOVWF    TRISB          ;BANK1 DE
```

```
MOVLW    B'00001110'    ;ADRESH=XXXXXXXX ADRESL=XX000000
```

```
MOVWF    ADCON1          ;BANK1 DE
```

```
BCF STATUS,RP1
```

```
        BCF STATUS,RP0          ;BANK0 DAYIZ
BASLA
        BSF    ADCON0,GO        ;BANK0 DA,,,analog digital dönüşüm başlasın
TEST
        BTFSC  ADCON0,GO        ;BANK0 DA,,,bu bit sıfırda dönüşüm tamamlanmış demektir
        GOTO   TEST
        BCF STATUS,RP1
        BSF STATUS,RP0          ;BANK1 DEYIZ
        MOVF  ADRESL,W          ;BANK1 DE. EN ÖNEMSİZ 2 BİTİ GÖSTERİR
        BCF STATUS,RP1
        BCF STATUS,RP0          ;BANK0 DAYIZ
        MOVWF    PORTB          ;BANK0 DA
        GOTO   BASLA
END
```

ADCON0 Kaydedicisi (Adres Kontrol Kaydedicisi 0 - Adres 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE'	-	ADON
Bit 7							Bit 0

Bit 7 → **ADCS1** : A/D çevirici için clock (saat) frekansı seçim bit'leridir. Bu bit'lere verilecek değerler ile A/D çevirme işlemi esnasında kullanılacak frekans değeri bize sunulan değerler içerisinde seçilir. (ADCON1 kaydedicisindeki ADCS2 bit'i ile birlikte kullanılırlar.)
 Bit 6 → **ADCS0**

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock (saat)
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	Frc (Dahili RC osilatör kaynağından gelen clock)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	Frc (Dahili RC osilatör kaynağından gelen clock)

Bit 5 → **CHS2** : A/D çevirici için analog kanal seçim bit'leri. Bu bit'lere verilecek değerlerle, A/D çevirme için hangi kanalın seçileceği belirlenir. Daha önceden de belirtildiği gibi PIC 16F877A'da A/D çevirici için 8 kanal mevcuttur.
 Bit 4 → **CHS1**
 Bit 3 → **CHS0**
 000 = kanal 0, (AN0)
 001 = kanal 1, (AN1)
 010 = kanal 2, (AN2)
 011 = kanal 3, (AN3)
 100 = kanal 4, (AN4)
 101 = kanal 5, (AN5)
 110 = kanal 6, (AN6)
 111 = kanal 7, (AN7)

Bit 2 → **GO/DONE** : A/D dönüştürücü durum bit'i.
ADON = 1 iken:
 1 = A/D çevirici işlem yapıyor demektir.
 0 = A/D çevirici herhangi bir işlem yapmıyordu.

Bit 1 → **U** : Kullanılmaz.

Bit 0 → **ADON** : A/D çeviriciyi yetkilendirme bit'idir.
 1 = A/D çevirici açıktır ve işlem yapılabilir.
 0 = A/D çevirici kapalıdır.

R : İlgili bit'in yalnızca okunabileceğini gösterir.
W : İlgili bit'in yazılabilir bit olduğunu gösterir.
U : Kullanılmayan bit'i gösterir.

Tablo 7.1 ADCON0 özel amaçlı kaydedicisinin bit açıklamaları.

294 PIC Programlama Teknikleri ve PIC16F877A**ADCON1 Kaydedicisi****(Adres Kontrol Kaydedicisi 1 - Adres 9Fh)**

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	-	-	PCFG3	PCFG2	PCFG1	PCFG0
Bit 7							Bit 0

Bit 7 → **ADFM** : A/D çevirme işlemi esnasında meydana gelen verinin biçimini belirlemeye yarayan bit'tir. ADFM=1 ise ADRESH kaydedicisinin MSB kısmındaki altı bit'i 0 kabul edilir ve A/D çevirme sonucunda elde edilen veri ADRESH'in 2 bit'lik LSB kısmına ve ADRESL'ye yazılır. ADFM=0 ise ADRESL'nin LSB kısmındaki 6 bit'i 0 kabul edilir ve A/D çevirme sonucu elde edilen veri ADRESL'nin son iki bit'ine ve ADRESH'a yazılır.

Bit 6 → **ADCS2** : A/D çevirici için clock (saat) frekansı seçim bit'idir. (ADCON0 kaydedicisindeki ADCS1 ve ADCS0 bit'leri ile birlikte kullanılır.)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (Dahili RC osilatör kaynağından gelen clock)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (Dahili RC osilatör kaynağından gelen clock)

Bit 5 → **U** : Kullanılmaz, "0" okunur.
Bit 4

Bit 3 → **PCFG3** : A/D çevirici port'unun biçimini düzenlemeyi sağlayan bit'lerdir. Yani A/D çevirme işleminde kullanılacak pin'lerin nasıl davranacağını belirlememize olanak sağlar.
Bit 2 **PCFG2**
Bit 1 **PCFG1**
Bit 0 **PCFG0**

PCFG3:PCFG0 Bit'lerinin Aldığı Değere Göre Yaptığı İşlemler:

PCFG3: PCFG0	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	KANAL/REF
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	-	-	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

Bu tabloda A=Analog giriş D=Dijital giriş/çıkış anlamına gelir.

R : İlgili bit'in yalnızca okunabileceğini gösterir.

W : İlgili bit'in yazılabilir bit olduğunu gösterir.

U : Kullanılmayan bit'i gösterir.

Tablo 7.2 ADCON1 özel amaçlı kaydedicisinin bit açıklamaları.

UYGULAMA 11 RB0 HARİCİ KESMESİ. Programın üstüne yazdığım açıklamaları uygulayınız

;Deney setinde rb0 ucu bir tel ile vcc ye bağlanır. ;rb0 ucu 5 v dan ayrılıp gnd a bağlandığında kesme gelir ve rb7 ye bağlı led yanar ve öyle kalır tekrar çalıştırmak için rb0 ucu vcc ye bağlanır deney seti resetlenir; tekrar rb0 ucu sıfıra çekildiğinde kesme oluşur...

```

LIST    P=16F877A
INCLUDE  "P16F877A.INC"
;***** AYARLAR *****
ORG     H'000'
GOTO    ANAPROG
ORG     H'004'
GOTO    RB0KES
ANAPROG
    CLRF  PORTB
    MOVLW B'10010000' ;GIE VE RB0 KESMESİNİ AÇ(ENABLE)
    MOVWF INTCON
    BSF   STATUS,RP0 ;BANK1 E GEÇ
    CLRF  OPTION_REG ;KESME DÜŞEN KENARDA 6.BİT 0
    MOVLW B'00000001' ;PORTB NİN 0.PİNİ GİRİŞ RB0 KESMESİ İÇİN, DİĞERLERİ ÇIKIŞ
    MOVWF TRISB
    BCF   STATUS,RP0
DÖNGÜ
    GOTO  DÖNGÜ ;SONSUZ DÖNGÜDE KESME BEKLER
RB0KES
    BCF   INTCON,INTF ;RB0 HARİCİ KESME BAYRAĞINI İNDİR SIFIR YAP
    BSF   PORTB,7 ;RB7 Yİ BİR YAP LED YANAR
    RETFIE ;KESME ALTPROGRAMINDAN DÖN
    END

```

INTCON Kaydedicisi**(Kesme Kontrol Kaydedicisi – Adres 0Bh, 8Bh, 10Bh, 18Bh)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMROIE	INTE	RBIE	TMROIF	INTF	RBIF
Bit 7							Bit 0

Bit 7	→	GIE	:	Tüm kesme işlemlerine izin verme bit'i. 0 = Tüm kesmelere izin verilmez. 1 = Aktif yapılmış olan tüm kesmelere izin verilir.
Bit 6	→	PEIE	:	Çevre ara birimlerden gelen kesmelere izin verme bit'i. 0 = Tüm çevresel kesmelere izin verilmez. 1 = Aktif yapılmış olan tüm çevresel kesmelere izin verilir.
Bit 5	→	TMROIE	:	TMR0 sayıcı taşma kesmesine izin verme bit'i. 0 = TMR0 kesmesine izin verilmez. 1 = TMR0 kesmesine izin verilir.
Bit 4	→	INTE	:	RB0/INT ucundan gelen kesmeye izin verme bit'i. 0 = RB0/INT kesmesine izin verilmez. 1 = RB0/INT kesmesine izin verilir.
Bit 3	→	RBIE	:	PORTB (4, 5, 6, 7. bit'leri) değişiklik kesmesine izin verme bit'i. 0 = PORTB'deki değişiklik kesmesine izin verilmez. 1 = PORTB'deki değişiklik kesmesine izin verilir.
Bit 2	→	TMROIF	:	TMR0 sayıcısı taşma kesmesi bayrağı bit'i. 0 = Zaman aşımı yok. 1 = Zaman aşımı var. (h'FF' den h'00' a geçiş.)
Bit 1	→	INTF	:	RB0/INT harici kesme bayrağı bit'i. 0 = RB0/INT harici kesme oluşmadığında. 1 = RB0/INT harici kesme oluştuğunda.
Bit 0	→	RBIF	:	PortB değişiklik kesmesi bayrağı bit'i. 0 = RB4~RB7 uçlarında değişiklik yok. 1 = RB4~RB7 uçlarından en azından birisinde değişiklik var.

R : İlgili bit'in yalnızca okunabileceğini gösterir.
W : İlgili bit'in yazılabilir bit olduğunu gösterir.
U : Kullanılmayan bit'i gösterir.

OPTION_REG Kaydedicisi (Seçme Kaydedicisi – Adres 81h, 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU'	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
Bit 7							Bit 0

- Bit 7 → **RBPU'** : PortB'nin pull-up dirençlerini aktif etme bit'i.
1 = Pull-up dirençleri aktif.
0 = Pull-up dirençleri pasif.
- Bit 6 → **INTEDG** : Harici kesme (RB0/INT) sinyali seçme bit'i.
1 = RB0/INT pininden gelen sinyalin yükselen kenarında kesme.
0 = RB0/INT pininden gelen sinyalin düşen kenarında kesme.
- Bit 5 → **T0CS** : Timer0 sinyal kaynağı seçme bit'i.
1 = Harici sinyal kaynağı (RA4/T0CKI).
0 = Dahili komut çevrimi seçilir.
- Bit 4 → **T0SE** : Timer0 sinyal kaynağı kenar seçme bit'i.
1 = RA4/T0CKI pininden yükselen kenar tetiklemesi.
0 = RA4/T0CKI pininden düşen kenar tetiklemesi.
- Bit 3 → **PSA** : Frekans bölücü seçme bit'i.
1 = Frekans bölme sayısı WDT için geçerli.
0 = Frekans bölme sayısı Timer0 için geçerli.
- Bit 2-0 → **PS2:PS0** : Prescaler değeri seçme bit'leri.

Bit Değeri			TMR0 Oranı	WDT Oranı
PS2	PS1	PS0		
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

- R** : İlgili bit'in yalnızca okunabileceğini gösterir.
W : İlgili bit'in yazılabilir bit olduğunu gösterir.
U : Kullanılmayan bit'i gösterir.

ÖDEV 6 UYGULAMA 4 de ki iç içe 3 çevrimli gecikme programının kaç milisaniye(veya kaç peryot) gecikme yaptığını hesaplayınız. Aynı gecikmeyi yapan bir altprogramı TIMER0'ı sayaç olarak kullanarak yazınız.

DENEY-5:MİKRODENETLEYİCİ UYGULAMALARI**Konular: TIMER0'ın sayaç olarak kullanılması, TIMER0 Kesmesi****UYGULAMA 12**

;TIMER0 ın sayaç olarak kullanılması

```

LIST    P=16F877A
INCLUDE "P16F877A.INC"
__CONFIG h'3F31';__CONFIG b'11111100110001';__CONFIG _CP_OFF & _WDT_OFF & _PWRT_ON &
_XT_OSC &
;
;_BODEN_OFF & _LVP_OFF & _CPD_OFF
;configürasyonu yukarıda verdik micropro26 sormayacak
BCF     STATUS,RP1
BSF     STATUS,RP0    ;BANK1 E GEÇTİK
MOVLW   B'11010111'   ; Prescaler(ön ölçekleme) timer0 a ait. 1/256, B'xxxx0111' x ler önemsiz
MOVWF   OPTION_REG
CLRF    TRISB          ;BANK1 DE
BCF     STATUS,RP1
BCF     STATUS,RP0    ;BANK0 DAYIZ
CLRF    PORTB

YAK
BSF     PORTB,0        ;portb0 daki led yanar
CALL    GECİKME

SONDUR
BCF     PORTB,0        ;portb0 daki led söner
CALL    GECİKME
GOTO    YAK

;*****gecikme alt programı timer0 ı sayaç olarak kullanıyor*****
GECİKME
CLRF    TMR0          ;sayıcı 00h den itibaren saymaya başladı. 256 peryodu 1 diye sayıyor

TESTBIT
BTFS    TMR0,7        ;timer0 ın 7.biti 1 olduysa döngüden çıkıyor, değilse devam ediyor
GOTO    TESTBIT
RETURN
END

```

BTFS TMR0,7 satırını 6 ve 5 yaparak tekrar derleyiniz ve programı deney setinde çalıştırınız.**BTFS TMR0,7** iken **CLRF TMR0** satırı yerine **MOVLW H'xx'** ve **MOVWF TMR0** satırlarını yazıp xx yerine 22, 33 ve 44 yazarak deneyiniz. Ne gözlüyorsunuz?**UYGULAMA 13**

;TIMER0 kesmesi

```

LIST    P=16F877A
INCLUDE "P16F877A.INC"
__CONFIG h'3F31';__CONFIG b'11111100110001';__CONFIG _CP_OFF & _WDT_OFF & _PWRT_ON &
_XT_OSC &
;
;_BODEN_OFF & _LVP_OFF & _CPD_OFF
;configürasyonu yukarıda verdik micropro26 sormayacak
;***** AYARLAR *****
ORG     H'000'
GOTO    ANAPROG
ORG     H'004'
GOTO    TMR0KES
ANAPROG
CLRWDT
BSF     STATUS,RP0    ;BANK1 E GEÇ

```

```

MOVLW    B'11010111'    ;tmr0 siny kayn dahili, Prescaler(ön ölçekleme) timer0 a ait. 1/256,
B'xx0x0111' x ler önemsiz
MOVWF    OPTION_REG
MOVLW    B'10100000'    ;GIE VE TMR0IE KESMESİNİ AÇ(ENABLE)
MOVWF    INTCON
CLRF    TRISB
BCF    STATUS,RP0
CLRF    PORTB

```

DÖNGÜ

```

GOTO    DÖNGÜ            ;SONSUZ DÖNGÜDE KESME BEKLER
TMR0KES
BCF    INTCON,T0IF    ;TMR0IF KESME BAYRAĞINI İNDİR SIFIR YAP. TÜRKÇE TABLODA
TMR0IF YAZIYOR YANLIŞ
MOVLW    H'01'            ;1 bu üç satırda ne yapıldığını inceleyiniz
ANDWF    PORTB,F            ;2
XORWF    PORTB,F            ;3
CLRF    TMR0            ;timer0 yeniden 00h den başlatılarak 256 peryotta 1 saydırılıyor
RETFIE    ;KESME ALTPROGRAMINDAN DÖN
END

```

Bu programda uygulama 11 de verdiğim programla benzer işi yapar. Ancak bu defa timer0 kesmesi kullanılmıştır.

MOVLW B'11010111' satırını **MOVLW B'11010110'** olarak değiştirerek deneyiniz. Benzer şekilde **TMR0KES** içindeki **CLRF TMR0** yerine uygulama 10 daki gibi **MOVLW H'xx'** ve **MOVWF TMR0** satırlarını yazıp xx yerine 22, 33 ve 44 yazarak deneyebilirsiniz.

Bu bilgileri ve sonuçları kullanarak deney4 de verilen ödev6 yı tekrar sorgulayınız.

INTCON Kaydedicisi

(Kesme Kontrol Kaydedicisi – Adres 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF
Bit 7							Bit 0

Bit 7	→	GIE	:	Tüm kesme işlemlerine izin verme bit'i. 0 = Tüm kesmelere izin verilmez. 1 = Aktif yapılmış olan tüm kesmelere izin verilir.
Bit 6	→	PEIE	:	Çevre ara birimlerden gelen kesmelere izin verme bit'i. 0 = Tüm çevresel kesmelere izin verilmez. 1 = Aktif yapılmış olan tüm çevresel kesmelere izin verilir.
Bit 5	→	TMR0IE	:	TMR0 sayıcı taşma kesmesine izin verme bit'i. 0 = TMR0 kesmesine izin verilmez. 1 = TMR0 kesmesine izin verilir.
Bit 4	→	INTE	:	RB0/INT ucundan gelen kesmeye izin verme bit'i. 0 = RB0/INT kesmesine izin verilmez. 1 = RB0/INT kesmesine izin verilir.
Bit 3	→	RBIE	:	PORTB (4, 5, 6, 7. bit'leri) değişiklik kesmesine izin verme bit'i. 0 = PORTB'deki değişiklik kesmesine izin verilmez. 1 = PORTB'deki değişiklik kesmesine izin verilir.
Bit 2	→	TMR0IF	:	TMR0 sayıcısı taşma kesmesi bayrağı bit'i. 0 = Zaman aşımı yok. 1 = Zaman aşımı var. (h'FF' den h'00' a geçiş.)
Bit 1	→	INTF	:	RB0/INT harici kesme bayrağı bit'i. 0 = RB0/INT harici kesme oluşmadığında. 1 = RB0/INT harici kesme oluştuğunda.
Bit 0	→	RBIF	:	PortB değişiklik kesmesi bayrağı bit'i. 0 = RB4~RB7 uçlarında değişiklik yok. 1 = RB4~RB7 uçlarından en azından birisinde değişiklik var.

R : İlgili bit'in yalnızca okunabileceğini gösterir.
W : İlgili bit'in yazılabilir bit olduğunu gösterir.
U : Kullanılmayan bit'i gösterir.

OPTION_REG Kaydedicisi (Seçme Kaydedicisi – Adres 81h, 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU'	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
Bit 7							Bit 0

- Bit 7 → **RBPU'** : PortB'nin pull-up dirençlerini aktif etme bit'i.
1 = Pull-up dirençleri aktif.
0 = Pull-up dirençleri pasif.
- Bit 6 → **INTEDG** : Harici kesme (RB0/INT) sinyali seçme bit'i.
1 = RB0/INT pininden gelen sinyalin yükselen kenarında kesme.
0 = RB0/INT pininden gelen sinyalin düşen kenarında kesme.
- Bit 5 → **T0CS** : Timer0 sinyal kaynağı seçme bit'i.
1 = Harici sinyal kaynağı (RA4/T0CKI).
0 = Dahili komut çevrimi seçilir.
- Bit 4 → **T0SE** : Timer0 sinyal kaynağı kenar seçme bit'i.
1 = RA4/T0CKI pininden yükselen kenar tetiklemesi.
0 = RA4/T0CKI pininden düşen kenar tetiklemesi.
- Bit 3 → **PSA** : Frekans bölücü seçme bit'i.
1 = Frekans bölme sayısı WDT için geçerli.
0 = Frekans bölme sayısı Timer0 için geçerli.
- Bit 2-0 → **PS2:PS0** : Prescaler değeri seçme bit'leri.

Bit Değeri			TMR0 Oranı	WDT Oranı
PS2	PS1	PS0		
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

- R** : İlgili bit'in yalnızca okunabileceğini gösterir.
W : İlgili bit'in yazılabilir bit olduğunu gösterir.
U : Kullanılmayan bit'i gösterir.

DENEY-6:MİKRODENETLEYİCİ UYGULAMALARI**Konular: yazılımla pwm üretilmesi****UYGULAMA 14**

;o.altınbaşak kitaptaki program 34 ün 877 ye uyarlanmış hali

```

LIST    P=16F877A
INCLUDE  "P16F877A.INC"
__CONFIG h'3F31' ;__CONFIG b'11111100110001' ;__CONFIG _CP_OFF & _WDT_OFF & _PWRT_ON &
_XT_OSC &
;_BODEN_OFF & _LVP_OFF & _CPD_OFF
;***** AYARLAR *****
IS      EQU    H'20'
PER     EQU    H'21'
ISYED   EQU    H'22'
SAYAC1  EQU    H'23'
SAYAC2  EQU    H'24'
        CLRF   PORTB
        BSF    STATUS,5
        CLRF   TRISB
        MOVLW  H'06'
        MOVWF  ADCON1      ;PORTA nın digital olması için adcon1 e 06h yazıyoruz
        MOVLW  H'FF'
        MOVWF  TRISA
        BCF    STATUS,5
;***** ANA PROGRAM *****
BASLA
        MOVLW  D'25'
        MOVWF  IS
        MOVWF  ISYED
DONGU
        MOVF   ISYED,W
        MOVWF  IS
        MOVLW  D'255'
        MOVWF  PER
        BSF    PORTB,0
        BTFSS  PORTA,1      ;PULL UP-DOWN ANAHTARI PULL OLACAK
        GOTO   M2
        BTFSS  PORTA,2
        GOTO   M3
PWM0
        DECFSZ  IS,F
        GOTO   PWM1
        BCF     PORTB,0
PWM1
        DECFSZ  PER,F
        GOTO   PWM0
        GOTO   DONGU
M2      CALL    GECİKME
        MOVF   ISYED,W
        MOVWF  IS
        INCF   IS,F
        MOVF   IS,W
        MOVWF  ISYED
        GOTO   DONGU
M3      CALL    GECİKME
        MOVF   ISYED,W
        MOVWF  IS
        DECF   IS,F
        MOVF   IS,W
        MOVWF  ISYED

```

```
        GOTO DONGU
GECİKME
        MOVLW      H'0F'
        MOVWF      SAYAC1
D1      MOVLW      H'FF'
        MOVWF      SAYAC2
D2      DECFSZ     SAYAC2,F
        GOTO D2
        DECFSZ     SAYAC1,F
        GOTO D1
        RETURN
        END
```

Ödev:

Programın çalışması tatmin edici değildir. İyileştirmek için uğraşınız. Varsa hataları belirleyiniz. Kendiniz görev periyodu %25 ve %35 olan pwm programı yazınız. 877 nin pwm modülünü inceleyiniz.

DENEY-7:MİKRODENETLEYİCİ UYGULAMALARI**Konular: yazılımla pwm üretilmesi yeni bir program****UYGULAMA 15**

```

LIST    P=16F877A
INCLUDE    "P16F877A.INC"
__CONFIG h'3F31' ;__CONFIG b'11111100110001' ;__CONFIG _CP_OFF & _WDT_OFF & _PWRT_ON &
_XT_OSC &
                ;_BODEN_OFF & _LVP_OFF & _CPD_OFF
;***** AYARLAR *****
SAYAC1      EQU    H'22'
SAYAC2      EQU    H'23'
DUTY EQU     H'24'
KALANEQU    H'25'
USTSINEQU   H'26'   ;DUTY(1 DE KALMA,GÖREV) SÜRESİ İÇİN ÜST SINIR
ALTSIN      EQU    H'27'   ;DUTY SÜRESİ İÇİN ALT SINIR

BSF    STATUS,5
MOVLW  H'00'           ;PORTb nın 8 pininde çıkış yapıyoruz
MOVWF  TRISB
MOVLW  H'06'
MOVWF  ADCON1          ;PORTA nın digital olması için adcon1 e 06h yazıyoruz
MOVLW  H'3F'           ;PORTA nın 6 pininde giriş yapıyoruz
MOVWF  TRISA
BCF    STATUS,5        ;BANK0 a geri dön
;***** ANA PROGRAM *****
CLRF   PORTB
MOVLW  H'01'
MOVWF  ALTSIN
SUBLW  H'FE'           ;W=L-W,YANI W=255-ALTSIN = ÜSTSINIR
MOVWF  USTSIN

MOVLW  H'FE'
MOVWF  DUTY            ;DUTY BAŞLANGIÇ DEĞERİ
;**BASBIRAK BASLANGIÇ*****
TEST
    BTFSS PORTA,1      ;PUL-UP ANAHTARI UP BASILINCA 0 GELİYOR BASILMAYINCA 1
    GOTO  BIRAK1
    BTFSS PORTA,2
    GOTO  BIRAK2
    GOTO  YAZ

BIRAK1
    BTFSC PORTA,1      ;BUTON BIRAKILINCA 1 GELECEK ALT SATIRA GEÇECEK
    GOTO  ART
    GOTO  BIRAK1      ; BUTON BIRAKILINCAYA KADAR BIRAK1 E GİDECEK

ART
    MOVF  USTSIN,W
    XORWF DUTY,W       ;SONUÇ W DE DUTY Yİ BOZMAMAK İÇİN
    BTFSC STATUS,Z     ;DUTY FEH İSE Z=1 OLUR
    GOTO  SINIRÜ       ;ÜST SINIR DEĞERİNE GİT
    INCF  DUTY,F       ;DUTY USTSINIR OLMAMIŞ ARTSIN
    GOTO  YAZ

SINIRÜ
    MOVF  USTSIN,W
    MOVWF DUTY         ;DUTY ÜSTSINIRA SABİTLENDİ
    GOTO  YAZ

BIRAK2
    BTFSC PORTA,2      ;BUTON BIRAKILDIMI
    GOTO  AZAL

```

```

        GOTO  BIRAK2
AZAL
        MOVF      ALTSIN,W
        XORWF     DUTY,W           ;SONUÇ W DE DUTY Yİ BOZMAMAK İÇİN
        BTFSZ     STATUS,Z        ;DUTY ALSINIRA ULAŞTI İSE İSE Z=1 OLUR
        GOTO      SINIRA          ;ALT SINIR DEĞERİNE GİT
        DECF      DUTY,F          ;DUTY ALT SINIR OLMAMIŞ AZALSIN
        GOTO      YAZ
SINIRA
        MOVF      ALTSIN,W
        MOVWF     DUTY            ;DUTY ALTSINIRA SABİTLENDİ
        GOTO      YAZ
YAZ
        MOVF      DUTY,W
        MOVWF     SAYAC1
        MOVLW     H'01'
        MOVWF     PORTB
        CALL      GECİK
        MOVF      DUTY,W          ;W=DUTY
        SUBLW     W'FF'           ;W=L-W,YANI W=255-DUTY   KALAN SIFIR SÜRESİ
        MOVWF     SAYAC1
        MOVLW     H'00'
        MOVWF     PORTB
        CALL      GECİK
        GOTO      TEST
;*** GECİK ALT PROGRAMI *****
GECİK
DONGU1
        MOVLW     H'01'
        MOVWF     SAYAC2
DONGU2
        DECFSZ    SAYAC2,F
        GOTO      DONGU2
        DECFSZ    SAYAC1,F
        GOTO      DONGU1
        RETURN
        END

```

Programı derleyip çalıştırınız. Bir voltmetre ile portb0 ve GND arasını DC olarak ölçünüz. Porta1 e bağlı buton arttırma butonudur. Bu butona bastıkça portb0 a bağlı ledin parlaklığı artacaktır. Ancak programın baş kısmında DUTY başlangıç değeri FE olarak verildiğinden led en parlak durumundadır. Yine de arttırmak için porta1 e bağlı butona basıp bırakırsanız voltmetrede ve led deki durumun değişmediğini izleyebilirsiniz. Porta2 azaltma butonuna basıldığında voltmetreden 0.017v bir azalma olduğunu ölçebilirsiniz. $5v/255=0.019v$ olduğuna göre ölçtüğünüz 0.017v u hesaplamaaya çalışınız.

INCF DUTY,F bulunduğu satıra 4 dört kere alt alta yazınız ve benzer şekilde
 DECF DUTY,F bulunduğu satıra 4 dört kere alt alta yazınız. Derleyip çalıştırınız. 0.07V luk artma ve azalmayı belirleyiniz.0.07 v u hesaplayınız. Programda uygun değişikliği yaparak 0.17 voltluk adımlarla arttırma ve azalma yapar hale getiriniz. Burada oluşabilecek aksaklıkları ve çözümlerini bulmaya çalışınız

Tekrar orijinal hale dönünüz. Bu defa GECİK alt programında sayac2 ye yazdırılan 01 sayısı yerine 55h yazınız. Programı derleyip çalıştırınız. Ne gözlemlediniz.