DENEY 1: PIC 16F84'DEN BİLGİSAYARA VERİ GÖNDERME

Bu uygulamada verici kısım PIC16F84, alıcı kısım ise bilgisayardır. Asenkron iletişim kurallarına göre her iki tarafta aynı parametreler kullanılacaktır. Yani iletişim hızı (baud rate), veri uzunluğu, stop bit'i aynı olmalıdır. Uygulamada kullanılan parametreler aşağıda listelenmiştir.

Başlat → Programlar → Donatılar → İletişim → Hyper Terminal

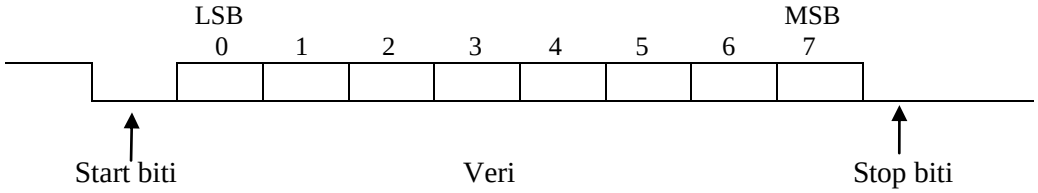
Parametreler
2400 Baud
Non-Parity
8-Bit veri
1 STOP Bit

Bilgisayarın verileri alması için hazır halde beklemesi gerekir. Bunun için Hyper Terminal programı çalıştırıp Uygun COM portu ve iletişim parametrelerinin ayarları yapılıp bekletilmelidir.

DENEY 1: PIC16F84'de kayıtlı olan "Merhaba Dünya" bilgisini bilgisayara gönderecek devreyi tasarlayıp, programını yazınız.

Programda iletişim parametreleri; 2400 Baud, 1 Stop bit, Parity yok, 8-bit veri uzunluğu olarak tanımlanacak.

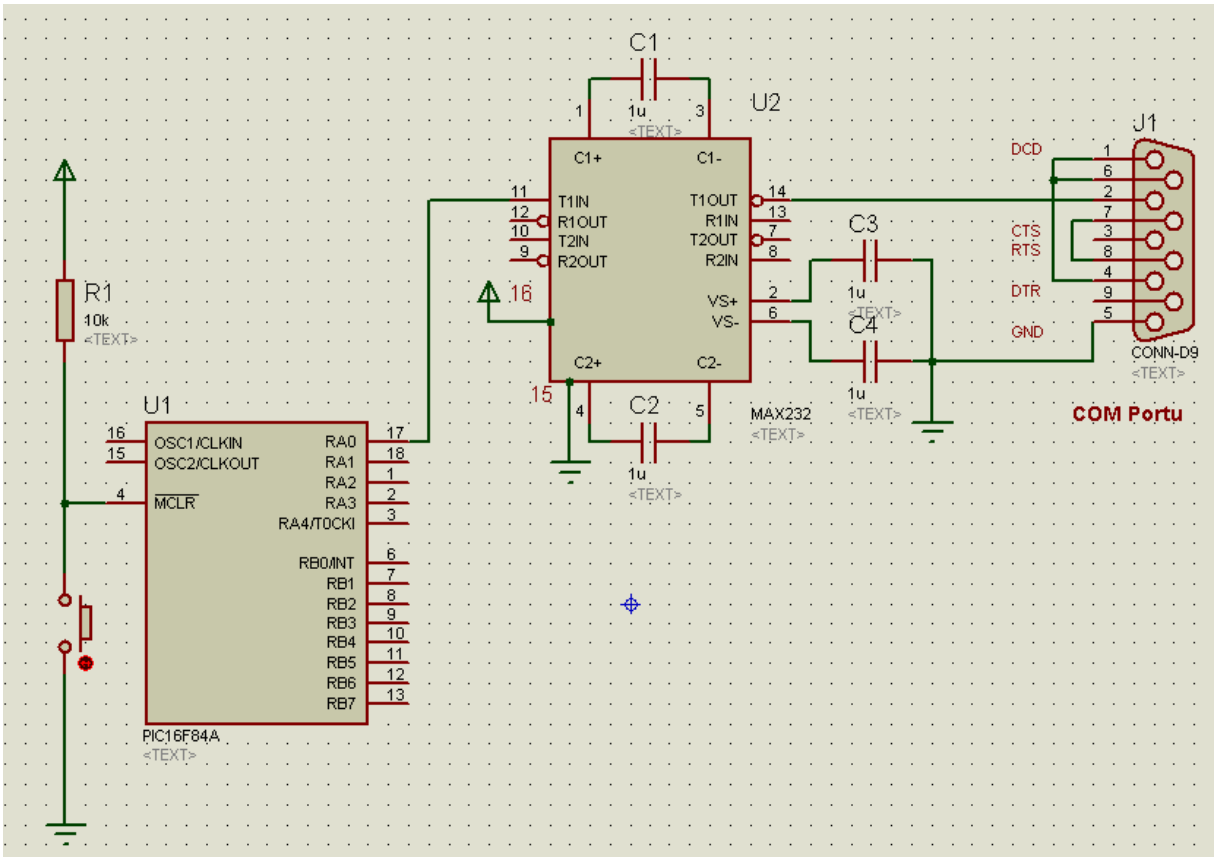
PIC16F84 içerisinde UART olmadığı için bu parametreler bir dizi komut halinde tanımlanacaktır. Parametreler bilgisayarda da aynı olmalıdır. Asenkron veri iletişiminde bir byte'lık veri aşağıdaki formatta verilir.



Asenkron iletişimde parity biti olmayan 1 byte'lık veri formatı

Programda start, veri ve Stop bitleri gönderilirken 2400 baudluk hız kullanılmalıdır. Bilindiği gibi baud saniyede gönderilen bit sayısıdır. 2400 baud için 1/2400=417μs'lik aralıklarla verilerin gönderilmesi gerekir.

START biti gönderilir
417μs beklenir
İlk veri biti gönderilir (LSB)
417μs beklenir
İkinci bit gönderilir
417μs beklenir
.
.
.
Yedinci bit gönderilir (MSB)
417μs beklenir
Stop biti gönderilir
417μs beklenir
Böylece 1 byte'lık veri gönderilmiş olur. Programda hata kontrol işlemleri yapılamayacağı için saniyede yaklaşık 200 civarında karakter gönderilebilir.

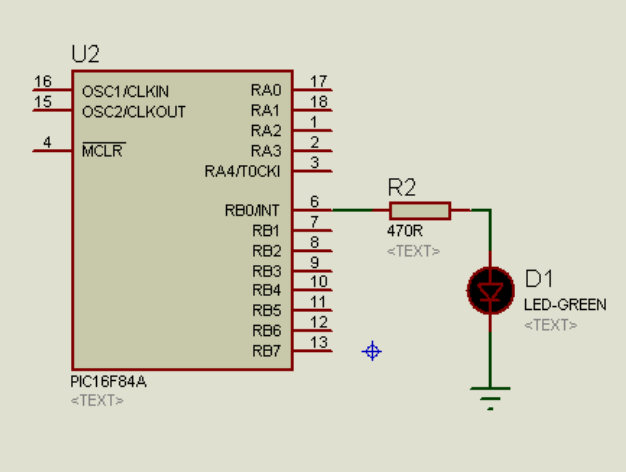


PIC 16F84'ten bilgisayara veri gönderen devre şeması

```
                TITLE       "verici.asm pıc 16f84-> pc seri iletişim"
                LIST        P=16F84
                INCLUDE     "P16F84.INC"
#define         SERI_OUT    PORTA,0         ;Seri data çıkışı
BT              EQU         0x16            ;Zaman gecikmesi için sayaç
VERICNT         EQU         0x13            ;Veri sayacı
SDATA           EQU         0x14            ;Seri olarak gönderilecek veri
TEMP            EQU         0x15            ;Rotate sayıcı
PC              EQU         0x02            ;Program counter

                ORG         0x00
BASLA;----------------------------------------------------------------------------
                CLRWDT
                CALL        INITIAL         ;Portları kur
                CALL        MESAJ           ;Verileri yaz
TEKRAR          GOTO        TEKRAR          ;Programı sonlandır

INITIAL;--------------------------------------------------------------------------
                BSF         STATUS,5
                MOVLW       0x00
                MOVWF       TRISA           ;PortA Çıkış
                BCF         STATUS,5
                CLRF        VERICNT         ;verici<- 0
                RETURN
```

```
MESAJ;-------------------------------------------------------------------------------------------------
MNEXT       MOVF        VERICNT,0           ;w <- vericnt
            CALL        MESAJ_VERISI        ;veriyi al
            IORLW       0                   ;0 ile test et
            BZ          MEND                ;0 ise mend etiketine git
            MOVWF       SDATA               ;0 değilse veriyi sdata'ya yaz
            CALL        DATA_GONDER         ;Seri olarak veri gönder
            INCF        VERICNT,1           ;bir sonraki veriyi adresle
            GOTO        MNEXT               ;Veri yazma işine devam et
MEND        RETURN

MESAJ_VERISI;-----------------------------------------------------------------------------------------
            ADDWF       PC,1                ;pc <- pc+w ile veriyi adresle
MSJ1        DT          "Merhaba Dünya",0
            MOVF        PC,0                ;İstenen karakteri w register'ına al
            RETURN

BAUD_TIME;--------------------------------------------------------------------------------------------
            MOVLW       0x86                ;Baud rate oranını sağlamak için
            MOVWF       BT                  ;gerekli zaman gecikmesi
NEXT1       DECFSZ      BT,F                ;bt=bt-1, bt=0 mı?
            GOTO        NEXT1               ;Hayır. next1'e git
            RETURN                          ;Evet, Alt programdan çık

DATA_GONDER;------------------------------------------------------------------------------------------
            BCF         SERI_OUT
            CALL        BAUD_TIME           ;START biti
            MOVLW       8                   ;Döndürme sayısı 8
            MOVWF       TEMP                ;temp<- 8

SKOMUT      RRF         SDATA,1             ;sdata bilgisini 1 bit sola kaydır
            BTFSS       STATUS,C            ;Kayan bit  1 mi?
            GOTO        $+3                 ;Hayır. Gelecek 2 komutu geç
            BSF         SERI_OUT            ;Evet. SD <- 1
            GOTO        $+2                 ;Bir sonraki komutu geç
            BCF         SERI_OUT            ;SD <- 0
            CALL        BAUD_TIME
            DECFSZ      TEMP                ;Bir sonraki bit için
            GOTO        SKOMUT              ;döndürme işlemine devam et
            BSF         SERI_OUT            ;STOP biti
            CALL        BAUD_TIME
            CALL        BAUD_TIME
            RETURN

            END
```

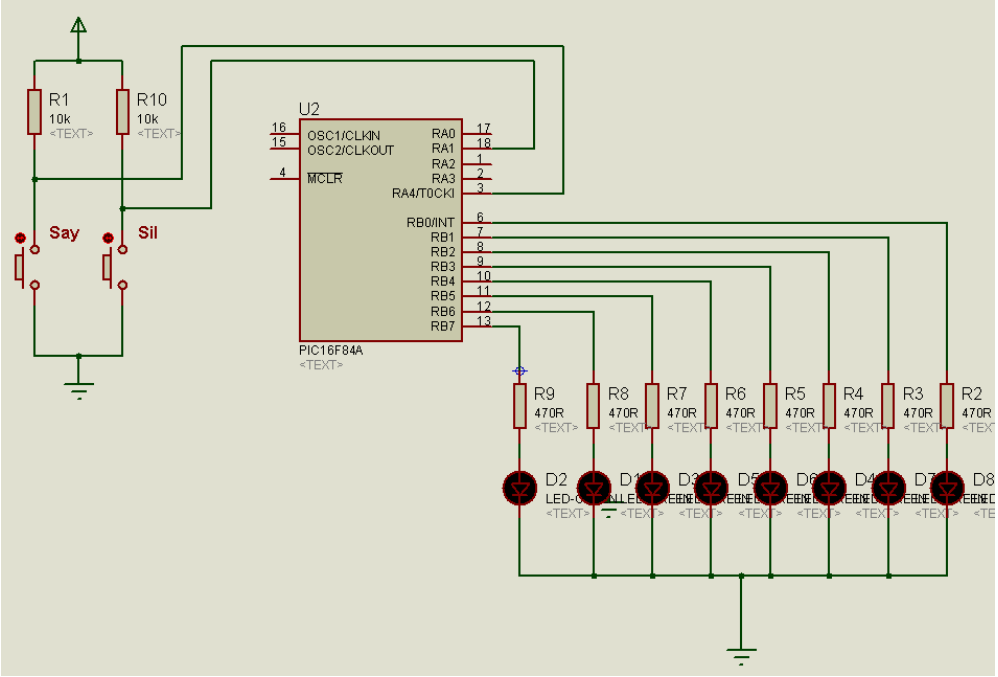
DENEY 2: TMR0 Kesmesinin Kullanımı



;PortB'nin 0. bit'ine bağlı LED'i flash yaptıran program.

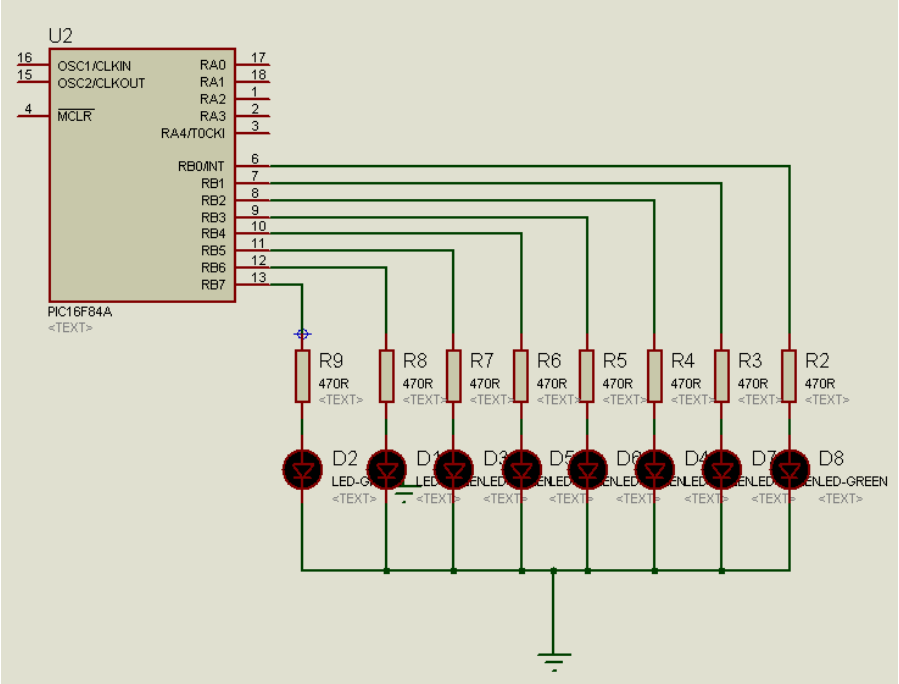
```
; LED'in yanıp sönme aralıklarındaki gecikmeyi TMR0 sayıcısı yapmaktadır. Bu program
;dijital çıkış sinyali (kare dalga) üretmek için kullanılmıştır. TMR0'ın sinyal kaynağı olarak
;dahili komut saykılı kullanılmış ve TMR0 oranı 1/256 seçilmiştir. Kristal osilatör kullanılan
;uygulamalarda kesme gecikmesi çok kısa olduğundan LED'in flash yapması görülmeyebilir.
;Bu durumda RB0 ucuna osilaskop bağlayarak çıkış izlenebilir. Eğer RC osilatör
;kullanılıyorsa seçilen R ve C değerleri değiştirilip frekans çok düşürülürse LED'in yanıp
;söndüğü görülür.
            LIST        P=16F84
             INCLUDE    "P16F84.INC"
            BSF         STATUS,5
            CLRF        TRISB
BASLA
            CLRWDT                      ; Prescaler atama işlemine hazırla
             MOVLW      B'11010111'     ; TMR0'ı, yeni prescaler değerini ve
                                        ;sinyal kaynağını seç
            MOVWF       OPTION_REG      ; Option registere yaz
            BCF         STATUS,5        ; Bank0'a geç
            CLRF        PORTB           ; PortB'nin tüm çıkışlarını 0 yap
YAK
            BSF         PORTB,0         ; LED'i yak
            CALL        GECIKME         ; Gecikme alt programını çağır
SONDUR
            BCF         PORTB,0         ; LED'i söndür
            CALL        GECIKME         ; Gecikme alt programını çağır
            GOTO        YAK             ; Yakıp-söndürmeye devam et
GECIKME
            CLRF        TMR0            ; TMR0'ı, h'00'a kur, saymaya başla

T_BIT
            BTFSS       TMR0,5          ; TMR0'ın 5. bit'i 1 mi?
            GOTO        T_BIT           ; Hayır, 5. bit'i tekrar test et
            RETURN
            END
```

DENEY 3: TMR0 Kesmesinin Kullanımı (Harici)



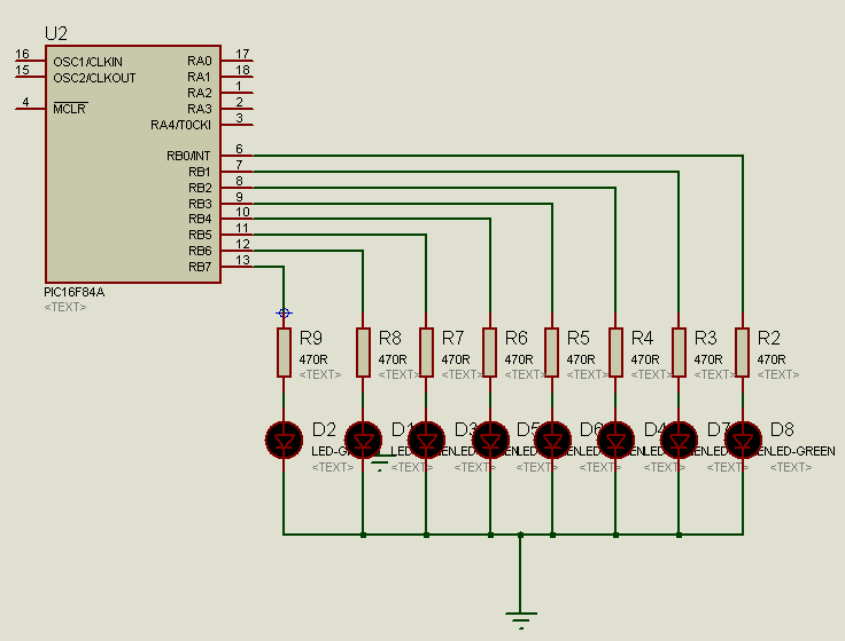

```
; TMR0 sayıcısının sinyal kaynağı olarak harici dijital sinyal (RA4/TOCKI ucu)
; kullanılmasına örnek programdır. Ayrıca TMR0'ın okunabilmesi ve sinyal sayıcı olarak
; kullanılabilme özelliğini de gösterir. Program, PortA'nın 4. bitine bağlı olan butona
; basıldığında PortB'deki LED'lerde binary olarak artan sayıları gösterir. RA1 butonuna
; basınca TMR0 registerini sıfırlar ve saymaya 0'dan itibaren tekrar saymaya başlar.
        LIST    P=16F84
        INCLUDE "P16F84.INC"
        CLRF    PORTB
        BSF     STATUS,5        ; Bank1'e git
        CLRF    TRISB           ; PortB'nin tüm uçlarını çıkış yap
        MOVLW   H'FF'           ; W ← h'FF'
        MOVWF   TRISA           ; PortA'nın tüm uçlarını giriş yap
        BCF     STATUS,5        ; Bank0'a git
BASLA
        CLRF    PORTB           ; PortB'yi sil
        CLRF    TMR0            ; TMR0 sayıcısı ve prescaler'ı sil
        CLRWDT                  ; WDT'yi sil
        BSF     STATUS,5        ; Bank1'e git
        MOVLW   B'00101111'     ; Prescaler değerini ata
        MOVWF   OPTION_REG      ; Option registere yaz
        BCF     STATUS,5        ; Bank0'a git
        CLRF    PORTB           ; PortB'yi sil
DONGU
        MOVF    TMR0,W          ; W ← TMR0
        MOVWF   PORTB           ; PortB ← W
        BTFSS   PORTA,1         ; PortA 1. bit 0 mı?
        CLRF    TMR0            ; Hayır, TMR0'ı sıfırla
        GOTO    DONGU           ; Evet, tekrar test et
        END
```

DENEY 4: TMR0 Kesmesinin Kullanımı (Gecikme)



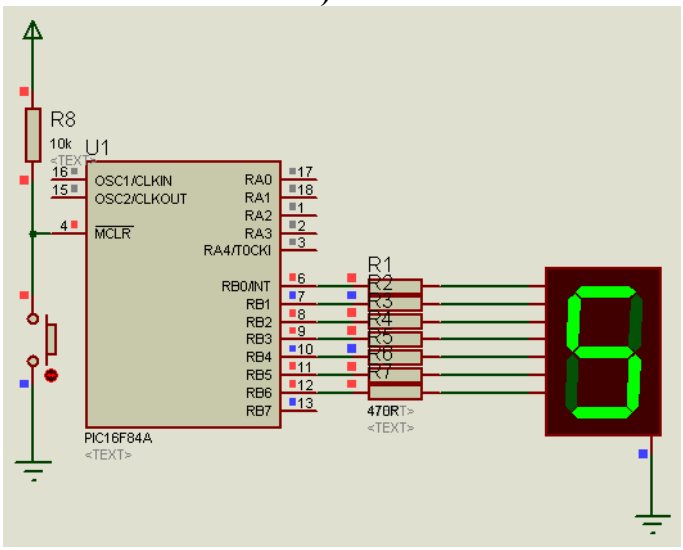

```
; TMR0 sayıcı kesmesine örnek programdır. PortB'deki LED'lerin binary olarak artan sayıları
; göstermesini sağlar. LED'lerin sayma aralıklarındaki duruşu için TMR0 sayıcısı
; kullanılmıştır.
                LIST        P=16F84
                INCLUDE     "P16F84.INC"
                ORG         H'00'
                GOTO        BASLA
                ORG         H'04'
                GOTO        LED_YAK         ;Kesme alt programına git
BASLA
                BSF         STATUS,5        ; Bank1'e git
                CLRF        TRISB           ; PortB'nin tüm uçlarını çıkış yap
                MOVLW       B'00000111'     ; W ← b'00000111' 1/256
                MOVWF       OPTION_REG      ; OPTION_REG ← W
                BCF         STATUS,5        ; Bank0'a git
                MOVLW       H'00'           ; W ← h'00'
                MOVWF       TMR0            ; TMR0 ← W
                MOVLW       B'10100000'     ; GIE ← 1, TOIE ← 1, TOIF ← 0
                MOVWF       INTCON          ; INTCON←W
                CLRF        PORTB           ;PortB'yi sil
DONGU
                GOTO        DONGU
LED_YAK
                BCF         INTCON,T0IF     ; TOIF←0
                INCF        PORTB,F         ; PortB←PortB+1
                MOVLW       H'00'           ; W ← h'00'
                MOVWF       TMR0            ; TMR0 ← W
                RETFIE                      ; Alt programdan dön
                END
```

DENEY 5: WDT Kesmesinin Kullanımı



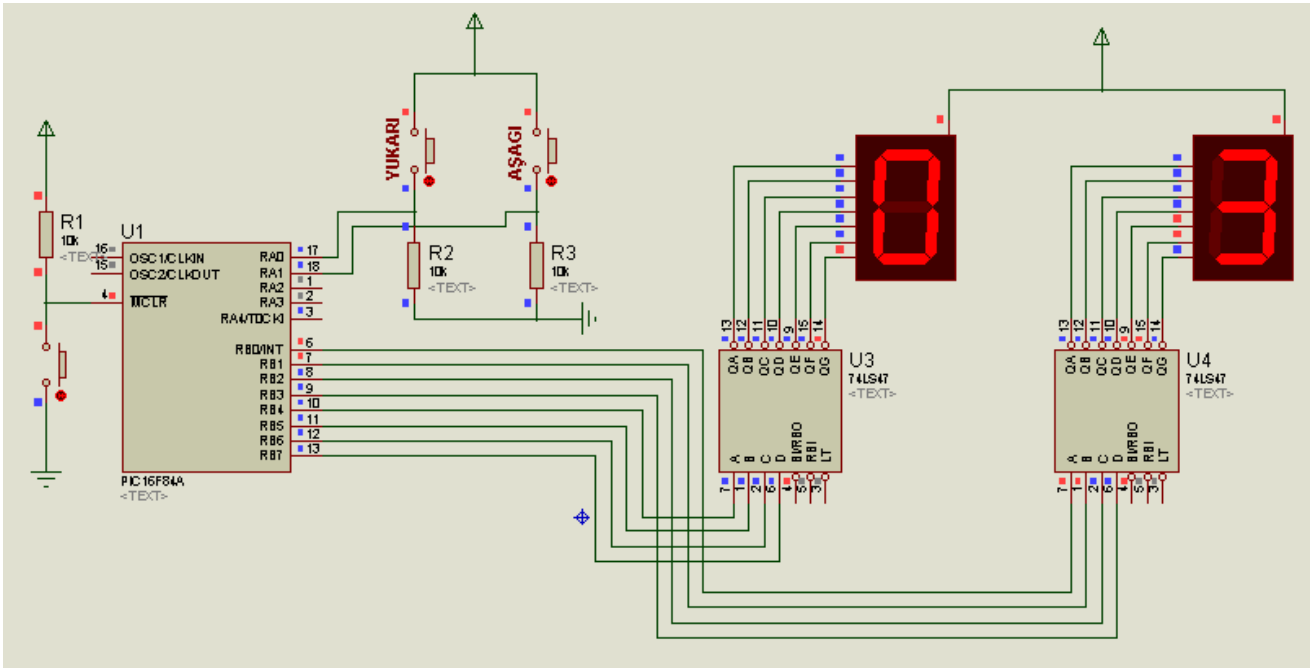

```
;  WDT sayıcı kesmesine örnek programdır. PortB'deki LED'lerin binary olarak artan sayıları
;  göstermesini sağlar. LED'ler sayma devam ederken PortB içerisindeki veri h'FF'e
;  ulaşmadan WDT zaman aşımı sinyali nedeniyle LED'lerin yanışı 0'dan itibaren tekrar
;  başlatılır.
                LIST        P=16F84
                INCLUDE     "P16F84.INC"
SAYAC1          EQU         H'0C'
SAYAC2          EQU         H'0D'
                __CONFIG    _WDT_ON&_XT_OSC
BASLA
                BSF         STATUS,5            ; Bank1'e geç
                MOVLW       B'00001110'         ; WDT seçilir
                MOVWF       OPTION_REG          ; Option_Reg ← W
                CLRF        TRISB               ; PortB'nin tüm uçlarını çıkış yap
                BCF         STATUS,5            ; Bank0'a geç
SONDUR
                CLRF        PORTB               ; PortB'yi sil
YAK
                CALL        GECIKME             ; Gecikme alt programını çağır
                INCF        PORTB,1             ; PortB ← PortB+1
                GOTO        YAK
GECIKME
                MOVLW       H'4F'               ; W ← h'4F'
                MOVWF       SAYAC1              ; Sayac1 ← W
DONGU1
                CLRF        SAYAC2              ; Sayac2 ←h'00'
DONGU2
                DECFSZ      SAYAC2,1
                GOTO        DONGU2
                DECFSZ      SAYAC1,1
                GOTO        DONGU1
                RETURN
                END
```

DENEY : 7 Segment ortak katodlu display'de "5" sayısını gösteren program (Çevrim Tablosunun Kullanımı)




```
; ********7 Segment ortak katodlu display'de 5 sayısını gösteren program **************
                LIST        P=16F84
                INCLUDE     "P16F84.INC"
                CLRF        PORTB       ; PortB'yi sil
                BSF         STATUS,5    ; Bank1'e geç
                CLRF        TRISB       ; PortB'nin tüm uçlarını çıkış yap
                BCF         STATUS,5    ; Bank0'a geç

BASLA
                MOVLW       H'05'       ; W <-- H'05' (Test sayısı)
                CALL        CEV_TAB     ; Cevrim tablosunu çağır
                MOVWF       PORTB       ; Çevrim tablosundan aldığın değeri
                                        ; PortB'ye gönder
DONGU
                GOTO DONGU              ; Sonsuz döngüye gir

CEV_TAB
                ADDWF       PCL,F       ; PCL <-- PCL + W (H'05')
                RETLW       H'3F'       ; 0
                RETLW       H'06'       ; 1
                RETLW       H'05'       ; 2
                RETLW       H'4F'       ; 3
                RETLW       H'66'       ; 4
                RETLW       H'6D'       ; 5
                RETLW       H'7D'       ; 6
                RETLW       H'07'       ; 7
                RETLW       H'7F'       ; 8
                RETLW       H'6F'       ; 9
                RETLW       H'77'       ; A
                RETLW       H'7C'       ; b
                RETLW       H'39'       ; C
                RETLW       H'5E'       ; d
                RETLW       H'79'       ; E
                RETLW       H'71'       ; F
                RETLW       H'80'       ; .
                END
```

DENEY 6: İki Dijit İleri-Geri Sayıcı



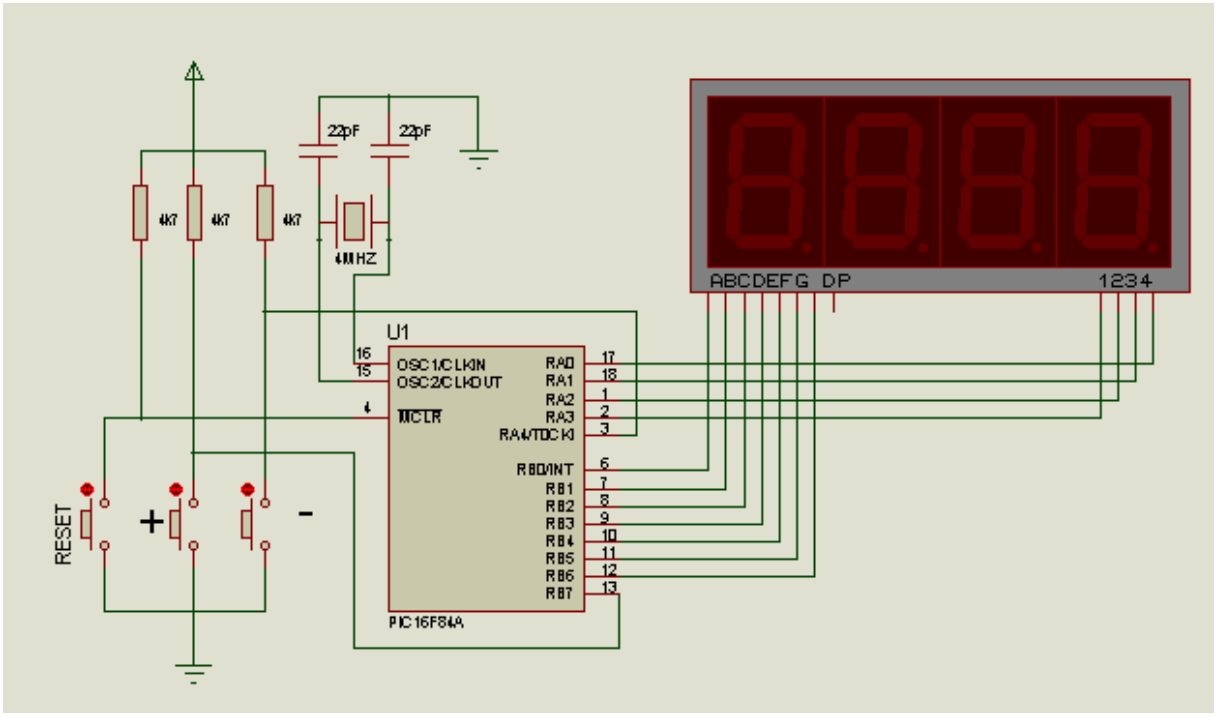
; **0-99 yukarı-aşağı sayıcı**
; RA0 butonuna her basışta yukarı sayar
; RA1 butonuna sasınca aşağı sayar
; Yukarı ve aşağı tuşlarına beraber basılırsa yukarı sayar
; Displayler PortB'ye 7 segment decoder ile bağlanmıştır

```
                LIST        P=16F84
                INCLUDE     "P16F84.INC"

SAY             EQU         11H         ; counter to turn on the pins on PortB
D0              EQU         12H         ; delay counter 0
D1              EQU         13H         ; delay counter 1
D2              EQU         14H         ; delay counter 2
TEMP            EQU         15H         ; Geçici register
X1              ORG         00H         ; Power on
                GOTO        START       ; 0000
START           BSF         STATUS,5
                MOVLW       0H
                MOVWF       TRISB
                MOVLW       0FH
                MOVWF       TRISA
                BCF         STATUS,5
TOP1            MOVLW       00H
                MOVWF       SAY
TOP2            MOVF        SAY,W
                MOVWF       PORTB
Y_TEST          BTFSS       PORTA,0
                GOTO        A_TEST
                CALL        Y_TUS
```

```
A_TEST      BTFSS       PORTA,1
            GOTO        Y_TEST
            CALL        A_TUS
            GOTO        Y_TEST

;***********************************
;YUKARI SAY TUSU  Subrutine
;***********************************

Y_TUS       BTFSS       PORTA,0
            GOTO        Y_TUS
            CALL        DELAY

Y_BIRAK     BTFSC       PORTA,0
            GOTO        Y_BIRAK
            INCF        SAY,1
            MOVF        SAY,W
            ANDLW       .15
            SUBLW       .10
            BTFSS       STATUS,2
            GOTO        S_CIKIS
            MOVF        SAY,W
            ANDLW       H'F0'
            ADDLW       .16
            MOVWF       SAY
            ANDLW       H'F0'
            SUBLW       H'A0'
            BTFSS       STATUS,2
            GOTO        S_CIKIS
            MOVLW       00H
            MOVWF       SAY
S_CIKIS     MOVF        SAY,W
            MOVWF       PORTB
            RETURN

;***********************************
;ASAGI SAY TUSU  Subrutine
;***********************************

A_TUS       BTFSS       PORTA,1
            GOTO        A_TUS
            CALL        DELAY
A_BIRAK     BTFSC       PORTA,1
            GOTO        A_BIRAK
            DECF        SAY,1
            MOVF        SAY,W
            ANDLW       .15
            SUBLW       .15
            BTFSS       STATUS,2
            GOTO        DEVAM
```

```
                MOVF        SAY,W
                ANDLW       H'F9'
                MOVWF       SAY
                ANDLW       H'F0'
                SUBLW       H'F0'
                BTFSS       STATUS,2
                GOTO        DEVAM
                MOVLW       99H
                MOVWF       SAY
                DEVAM       MOVF SAY,W
                MOVWF       PORTB
                RETURN

;***********************************
; DELAY Subroutine
;***********************************
DELAY           MOVLW       .2
                MOVWF       D0
ZD0             MOVLW       .2
                MOVWF       D1
ZD1             DECFSZ      D1,F
                GOTO        ZD1
                DECFSZ      D0,F
                GOTO        ZD0
                RETLW       00
                END
```

DENEY 10 : Dört Dijit İleri-Geri Sayıcı




```
;0-9999 ileri-geri sayıcı
                LIST        P=16F84A
                #include    "P16F84A.INC"
                __CONFIG    _WDT_OFF &_XT_OSC &_PWRTE_ON &_CP_OFF
;
LSB             EQU         H'0021'
MSB             EQU         H'0022'
SAYI1           EQU         H'0023'
SAYI2           EQU         H'0024'
SAYI3           EQU         H'0025'
SAYI4           EQU         H'0026'
SAYI5           EQU         H'0027'
DEGER           EQU         H'0028'
BIR             EQU         H'0029'
ON              EQU         H'002A'
YUZ             EQU         H'002B'
BIN             EQU         H'002C'
RAKAM           EQU         H'002D'
RAKAM1          EQU         H'002E'
TEMP            EQU         H'002F'
;-------------
BASLA
                CLRF        MSB
                CLRF        LSB
                BSF         STATUS,5
                MOVLW       B'11110000'
                MOVWF       TRISA
```

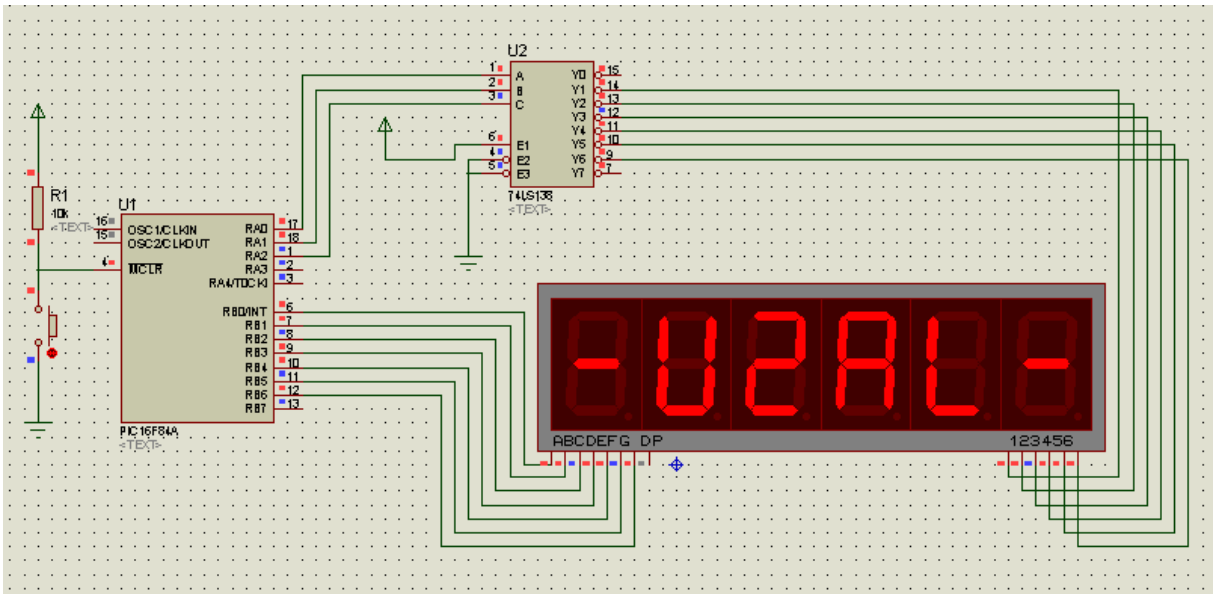
```
                MOVLW   B'10000000'
                MOVWF   TRISB
                BCF     STATUS,5
                CLRF    PORTB
                CLRF    PORTA
                CLRF    BIR
                CLRF    ON
                CLRF    YUZ
                CLRF    BIN
                CLRF    SAYI1
                CLRF    SAYI2
                CLRF    SAYI3
                CLRF    SAYI4
                GOTO    ANA
;-----------
ART
                INCF    BIR,F
                MOVLW   .10
                SUBWF   BIR,W
                BTFSS   STATUS,Z
                GOTO    ASON
                CLRF    BIR
                INCF    ON,F
                MOVLW   .10
                SUBWF   ON,W
                BTFSS   STATUS,Z
                GOTO    ASON
                CLRF    ON
                INCF    YUZ,F
                MOVLW   .10
                SUBWF   YUZ,W
                BTFSS   STATUS,Z
                GOTO    ASON
                CLRF    YUZ
                INCF    BIN,F
                MOVLW   .10
                SUBWF   BIN,W
                BTFSS   STATUS,Z
                GOTO    ASON
                CLRF    BIN
ASON
                CALL    EKRAN
                BTFSS   PORTB,7
                GOTO    ASON
                GOTO    ANA
;-----------
AZAL
                MOVLW   .1
                SUBWF   BIR,F
                BTFSC   STATUS,C
```

```
                GOTO    ESON
                CLRF    BIR
                MOVLW   .1
                SUBWF   ON,F
                BTFSC   STATUS,C
                GOTO    BIR9
                CLRF    ON
                MOVLW   .1
                SUBWF   YUZ,F
                BTFSC   STATUS,C
                GOTO    ON9
                CLRF    YUZ
                MOVLW   .1
                SUBWF   BIN,F
                BTFSC   STATUS,C
                GOTO    YUZ9
                CLRF    BIN
                GOTO    ESON
;----------
YUZ9
                MOVLW   .9
                MOVWF   YUZ
ON9
                MOVLW   .9
                MOVWF   ON
BIR9
                MOVLW   .9
                MOVWF   BIR
ESON
                CALL    EKRAN
                BTFSS   PORTA,4
                GOTO    ESON
                GOTO    ANA
;-----------
ANA
                CALL    EKRAN
                BTFSS   PORTA,4
                GOTO    AZAL
                BTFSS   PORTB,7
                GOTO    ART
                GOTO    ANA
;----------
EKRAN
                MOVLW   .5
                MOVWF   RAKAM
                CLRF    PORTB
                MOVLW   .255
                MOVWF   PORTA
GOSTER
                BCF     PORTA,0
```

```
        BSF     PORTA,1
        BSF     PORTA,2
        BSF     PORTA,3
        MOVF    BIR,W
        CALL    TABLO
        MOVWF   PORTB
        CALL    GECIKME
        CALL    GECIKME
        CLRF    PORTB
        BSF     PORTA,0
        BCF     PORTA,1
        BSF     PORTA,2
        BSF     PORTA,3
        MOVF    ON,W
        CALL    TABLO
        MOVWF   PORTB
        CALL    GECIKME
        CLRF    PORTB
        BSF     PORTA,0
        BSF     PORTA,1
        BCF     PORTA,2
        BSF     PORTA,3
        MOVF    YUZ,W
        CALL    TABLO
        MOVWF   PORTB
        CALL    GECIKME
        CLRF    PORTB
        BSF     PORTA,0
        BSF     PORTA,1
        BSF     PORTA,2
        BCF     PORTA,3
        MOVF     BIN,W
        CALL    TABLO
        MOVWF   PORTB
        CALL    GECIKME
        DECFSZ  RAKAM,F
        GOTO    GOSTER
        RETURN
GECIKME
        MOVLW   .5
        MOVWF   MSB
D11
        MOVLW   .55
        MOVWF   LSB
D22
        DECFSZ  LSB,F
        GOTO D22
        DECFSZ  MSB,F
        GOTO D11
        RETURN
```

```
TABLO
                ADDWF   PCL,F
                RETLW   h'3F'
                RETLW   h'06'
                RETLW   h'5B'
                RETLW   h'4F'
                RETLW   h'66'
                RETLW   h'6D'
                RETLW   h'7D'
                RETLW   h'07'
                RETLW   h'7F'
                RETLW   h'6F'
                RETLW   h'77'
                RETLW   h'7C'
                RETLW   h'39'
                RETLW   h'5E'
                RETLW   h'79'
                RETLW   h'71'
                RETLW   h'80'

;-----------------------------------------------------------------

                END
```

DENEY 7: 6 Display İle Sabit Yazı Yazma



6'lı taramalı display bağlantı devresi

;**S_YAZI.ASM**
;Bu program taramalı çalışan 6 display üzerinde sabit bir yazıyı yazar.
;**Display bağlantısı:**
;a=RB0
;b=RB1
;c=RB2
;d=RB3
;e=RB4
;f=RB5
;g=RB6
;**Sıralama** d1,d2,d3,d4,d5,d6
;**Select uçları** RA2,RA1,RA0 üzerine bağlanmış 3 to 8 mux ile yapılmaktadır.
;Multiplexerin Y0 çıkışı boş bırakılmıştır.

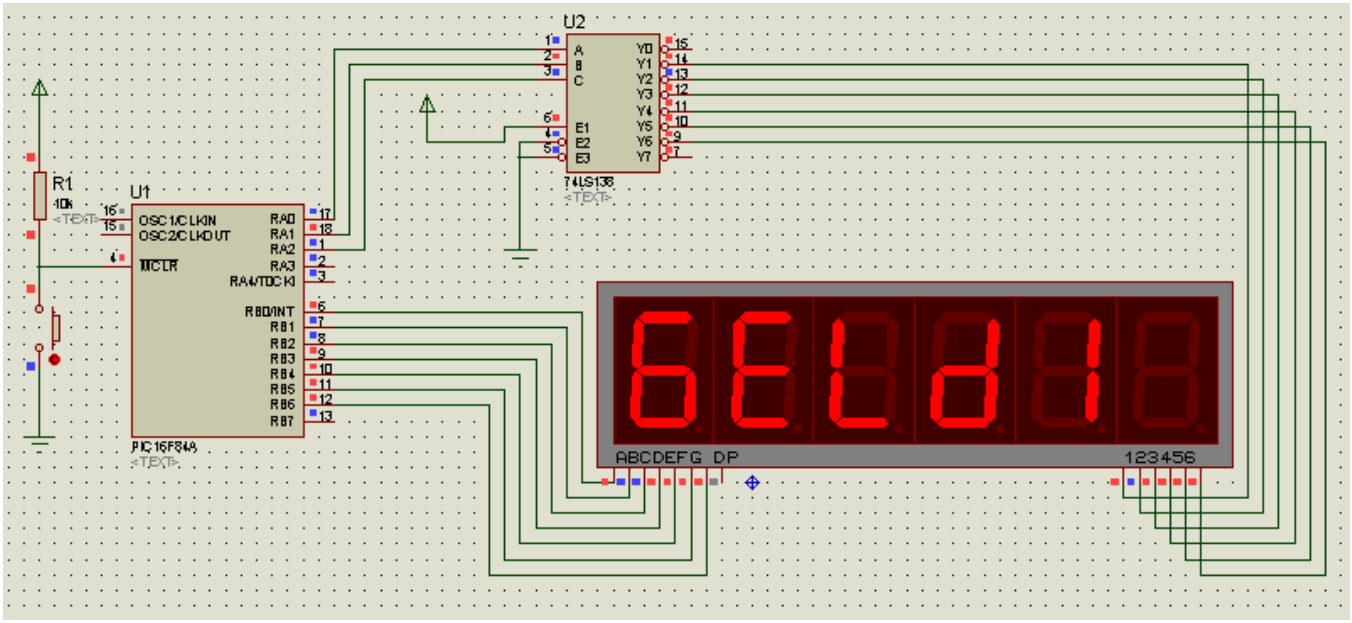
;Diğer uçlar sırayla d1..d6 ya bağlanmış ve Y7 boş bırakılmıştır.
;**Örnek data :** -UZAL- şeklindedir.

```
                LIST        P=16f84
                INCLUDE     "P16F84.INC"
; Değişkenler
D1              EQU         11H
D2              EQU         12H
D3              EQU         13H
D4              EQU         14H
D5              EQU         15H
D6              EQU         16H

X1              org         0h
                goto        START
```

```
START   bsf     STATUS,5    ; Bank 1
        clrf    TRISB       ; PortB çıkış
        clrf    TRISA       ; PortA çıkış
        bcf     STATUS,5    ; Bank 0

hazir   movlw   40h         ;1. harf datasını (-) D1 adresine gönder
        movwf   D1

        movlw   3eh         ;2. harf datasını (U) D2 adresine gönder
        movwf   D2

        movlw   5bh         ;3. harf datasını (Z)D3 adresine gönder
        movwf   D3

        movlw   77h         ;4. harf datasını (A)D4 adresine gönder
        movwf   D4

        movlw   38h         ;5. harf datasını (L)D5 adresine gönder
        movwf   D5

        movlw   40h         ;6. harf datasını (-)D6 adresine gönder
        movwf   D6

tt      call    yazar       ;Yazı yazma alt programını çağır
        goto    tt          ;Yazma işlemini sürekli yap

yazar   movf    D1,w        ;1. datayı 1. displayda göster
        clrf    PORTA
        movwf   PORTB
        movlw   01h
        movwf   PORTA

        movf    D2,w        ;2. datayı 2. displayda göster
        clrf    PORTA
        movwf   PORTB
        movlw   02h
        movwf   PORTA

        movf    D3,w        ;3. datayı 3. displayda göster
        clrf    PORTA
        movwf   PORTB
        movlw   03h
        movwf   PORTA

        movf    D4,w        ;4. datayı 4. displayda göster
        clrf    PORTA
        movwf   PORTB
        movlw   04h
        movwf   PORTA
```

```
movf        D5,w        ;5. datayı 5. displayda göster
clrf        PORTA
movwf       PORTB
movlw       05h
movwf       PORTA

movf        D6,w        ;6. datayı 6. displayda göster
clrf        PORTA
movwf       PORTB
movlw       06h
movwf       PORTA
return

end
```

DENEY 8: 6 Display İle Kayan Yazı Deneyi



;**K_YAZI:ASM**
;Bu program taramalı çalışan 6 display üzerinde sabit bir yazıyı kaydırarak yazar.
;**Display bağlantısı:**
;a=RB0
;b=RB1
;c=RB2
;d=RB3
;e=RB4
;f=RB5
;g=RB6
;**Sıralama** d1,d2,d3,d4,d5,d6
;**Seleçt ucları** RA2,RA1,RA0 üzeine ba lanm. 3 to 8 mux ile yapılmaktadır.
;Multiplexerin Y0 çıkışı boş bırakılmıştır.
;Diğer uçlar sırayla d1..d6 ya bağlanmış ve Y7 boş bırakılmıştır.

;**Örnek data :** bAhAr GELdI hOSGELdI şeklindedir.

```
                LIST        P=16f84
                INCLUDE     "P16F84.INC"

PCL             EQU         02h

; Değişkenler
ZD1             EQU         0Fh
ZD2             EQU         0Eh
h_ad            EQU         17h
ilk             EQU         18h
```

```
;Display değişkenleri adresleri
D1          EQU         11H
D2          EQU         12H
D3          EQU         13H
D4          EQU         14H
D5          EQU         15H
D6          EQU         16H

X1          org         00h
            goto        START

START       bsf         STATUS,5    ; Bank 1
            clrf        TRISB       ; PortB çıkış
            clrf        TRISA       ; PortA çıkış
            bcf         STATUS,5    ; Bank 0

tekrar      movlw       .30         ;Harf adedini tespit et
            movwf       h_ad
            movlw       00h         ;lk data adresi 0 olacak
            movwf       ilk

hazir       movf        ilk,w       ;data adresindeki harfi almak için indexi ayarla
            call        tablo       ;tablodan harfi seç
            movwf       D1          ;ilgili display adresine gönder
            incf        ilk,1       ;indexi 1 artır

            movf        ilk,w       ;Aynı işlemi 2. display için tekrarla
            call        tablo
            movwf       D2
            incf        ilk,1

            movf        ilk,w       ;Aynı işlemi 3. display için tekrarla
            call        tablo
            movwf       D3
            incf        ilk,1

            movf        ilk,w       ;Aynı işlemi 4. display için tekrarla
            call        tablo
            movwf       D4
            incf        ilk,1

            movf        ilk,w       ;Aynı işlemi 5. display için tekrarla
            call        tablo
            movwf       D5
            incf        ilk,1

            movf        ilk,w       ;Aynı işlemi 6. display için tekrarla
            call        tablo
            movwf       D6
```

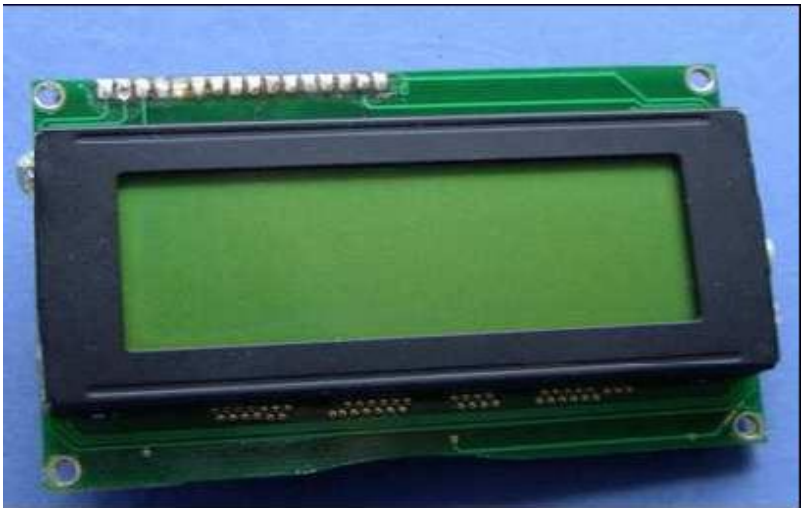
```
                decf    ilk,1       ;Bir sonraki tur için indexi ayarla (4 azalt)
                decf    ilk,1
                decf    ilk,1
                decf    ilk,1

                movlw   .1          ;Bir turun ekrandaki süresini ayarla
                movwf   ZD1

t2              movlw   .200
                movwf   ZD2

t1              call    yazar       ;Ekrana yazma programını döngü süresince tekrar
                                    ;tekrar çağır
                decfsz  ZD2,1
                goto    t1

                decfsz  ZD1,1
                goto    t2

                decfsz  h_ad,1      ;Mesajın tamam olup olmadığını kontrol et
                goto    hazir
                goto    tekrar      ;Mesaj tamam ise baştan başla

;Display adreslerindeki dataları ekrana yazdıran alt program

yazar           movf    D1,w        ;d1 adresindeki datayı al
                clrf    PORTA           ;PortA'yı sil
                movwf   PORTB           ;d1 datasını portB'ye gönder
                movlw   01h         ;A portundan 1. displayi seç
                movwf   PORTA

                movf    D2,w        ;Aynı işlemi ikinci display için tekrarla
                clrf    PORTA
                movwf   PORTB
                movlw   02h
                movwf   PORTA

                movf    D3,w        ;Aynı işlemi üçüncü display için tekrarla
                clrf    PORTA
                movwf   PORTB
                movlw   03h
                movwf   PORTA

                movf    D4,w        ;Aynı işlemi dördüncü display için tekrarla
                clrf    PORTA
                movwf   PORTB
                movlw   04h
                movwf   PORTA

                movf    D5,w        ;Aynı işlemi beşinci display için tekrarla
```

```
                clrf    PORTA
                movwf   PORTB
                movlw   05h
                movwf   PORTA

                movf    D6,w        ;Aynı işlemi altıncı display için tekrarla
                clrf    PORTA
                movwf   PORTB
                movlw   06h
                movwf   PORTA
                return

;Mesaj datalarını tutan alt program

tablo           addwf   PCL,1       ;Mesaj harfler tablosu
                retlw   00h         ;ilk 5 data 00 (boşluk)
                retlw   00h
                retlw   00h
                retlw   00h
                retlw   00h
                retlw   7ch         ;b Gerçek data başlangıç adresi
                retlw   77h         ;A
                retlw   74h         ;H
                retlw   77h         ;A
                retlw   50h         ;r
                retlw   00h         ;boşluk
                retlw   7dh         ;G
                retlw   79h         ;E
                retlw   38h         ;L
                retlw   5eh         ;d
                retlw   06h         ;i
                retlw   00h         ;boşluk
                retlw   74h         ;h
                retlw   3fh         ;O
                retlw   6dh         ;S
                retlw   7dh         ;G
                retlw   79h         ;E
                retlw   38h         ;L
                retlw   5eh         ;d
                retlw   06h         ;i Dataların sonu
                retlw   00h         ;Sonunda 6 adet boşluk
                retlw   00h
                retlw   00h
                retlw   00h
                retlw   00h
                retlw   00h
                end
```

DENEY 9: 8-Bitlik Veri İle LCD Ekrana Veri Yazmak

LCD (Liquid Crystal Display)

LCD ekranlar bize birçok harfi, sayıları, sembolleri görüntüleme imkanı verirler. LCD'lerde hane sayıları değişik olabilir. Bir hane 35 nokta içerdiğinden, 7 bölmeli göstergede olduğu gibi bir tarama yaparak 20 haneli bir Dot Matris LCD'yi çalıştırmak için kullanılan mikroişlemcinin görevi sadece LCD'ye veri yazmak olur. Araya diğer işlemler girerse ekrana yazılan veriler hatalı olur. Bu yüzden LCD'ler için ekranı kontrol edecek ayrı işlemcilere ihtiyaç duyulur. LCD'ler, bilgisayarda kullanılan ekran kartları gibi ekrana yazılan bilgilerin sürekli görünebilmeleri için tarama işlemini yapan entegrelerle birlikte üretilip satılırlar. Bu entegrenin özelliklerini tam olarak bilmek ekranı her yönüyle kontrol etmek manasına gelir. Genelde kullanılan bir LCD sürücü entegresi Hitachi firmasının üretmiş olduğu HD44780 entegresidir. Bu entegrenin kullanıldığı tüm göstergeleri aynı mantıkla kontrol etmek mümkündür. Tek değişiklik LCD'nin kaç satır ve haneden olduğunu bilmektir.



Resim 1. LCD Ekran

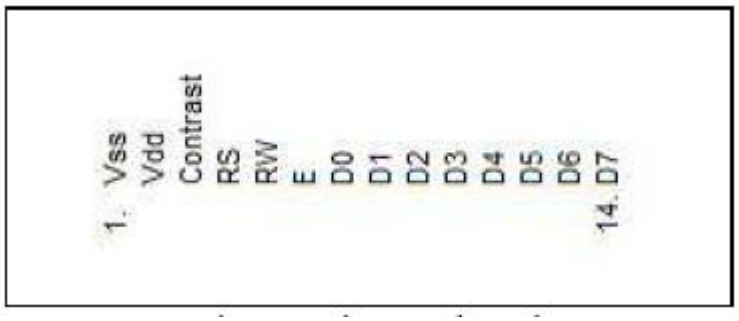
2 Satır x 40 Hane LCD Ekran

- 4-bit yada 8-bit mikroişlemci bağlantısı
- 80 x 8-bit RAM
- 5 x 7 nokta 160 karakter veya 5 x 10 nokta 32 karakter yada üretici ROM
- Hem gösterge verileri hem de karakter üreteci RAM verileri mikroişlemci tarafından okunabilir.
- Geniş komut fonksiyonları:
  - Display temizleme
  - Display karakter karartması (blink)
  - Kursor başa alma (return home)
  - Kursor kaydırma (shift)
  - Display ON/OFF
  - Hane (gösterge) kaydırma
  - Kursor ON/OFF
- Power On otomatik iç reset devresi

Resim 2. 2 x 40 LCD Ekran

Pin Bağlantıları

80 veya daha az karaktere ait LCD ekranlar için pin isimleri aşağıda tablo olarak verilmiştir. LCD ekranlar yanlış bağlantılardan dolayı çabuk bozulur. LCD üzerindeki rakamlar ve bağlantı yönlerinden emin olunmalıdır.



Resim 3. Pin Bağlantıları

Pin No	Sembol	Seviye	I / O	Fonksiyonu
1	Vss			Güç kaynağı (GND)
2	Vdd			Güç kaynağı (+5V)
3	Vo			Kontrast ayarı
4	RS	H / L	I	L: Komut kodu girişi; H: Veri girişi
5	R / W	H / L	I	H: Veri okuma; L: Veri yazma
6	E	H, H -> L	I	Etkinleştirme sinyali
7	D0	H / L	I / O	Veri Hatları HD44780 ile iki türlü veri yolu bağlantısı yapılabilir.
8	D1	H / L	I / O	
9	D2	H / L	I / O	
10	D3	H / L	I / O	
11	D4	H / L	I / O	
12	D5	H / L	I / O	
13	D6	H / L	I / O	
14	D7	H / L	I / O	

İlk Kullanıma Hazırlama (Initalization)

Gösterge tarama işlemi nasıl program ile belli bir sırada yapılıyorsa, LCD kullanırken de belli bir işlem sırası takip edilmelidir.

LCD'nin çalışması aşağıdaki işlem sırasına göre olmaktadır.

1. Resetlenir
2. Veri uzunluğu, satır adedi ve karakter fontu belirtilir.
3. Display ON/OFF yapılır.
4. Kursor ayarlanır.
5. Verilerin sağa doğru mu, yoksa sola doğru mu kayacağı belirtilir.
6. Display Data RAM'e istenilen karakterler yazılır.

Not 1: Entegrede, RS pini komut yada veri bilgisini bildirir. Eğer bu pin 0 ise gelen bilgilerin komut, 1 ise göstergeye yazılmasını istediğimiz verilerin olduğunu bildirir.

Not 2: RW pini göstergeden veri okumak yada göstergeye veri yazmak için kullanılır. Eğer bu pin 0 ise göstergeye veri yazılır, 1 ise göstergeden veri okunur.

Not 3: E pini göstergenin veri yada komut kabul etmesini sağlar. Yalnız bu pin düşen kenarda tetiklenir. Göstergeye bir karakter gönderirken, okurken yada onu programlarken önce bu ucun 1 ardından da 0 yapılması gerekir.

Komut	**Kod** **R/S RW D7 D6 D5 D4 D3 D2 D1 D0**	**Açıklama**
Clear Display	0 0 0 0 0 0 0 0 0 1	Tüm ekran temizlenir ve kursor ekranın başına döner.
Return Home	0 0 0 0 0 0 0 0 1 *	Kursor ekran başına döner, normal durumda kayma devam eder, DDRAM içeriği değişmez.
Entry Mode Set	0 0 0 0 0 0 0 1 I/D S	Kursorun hareket yönünü (I/D) ve göstergenin özelliğini (S) belirtir. Veri okuma ve yazma konumlarında geçerlidir.
Display ON/OFF Control	0 0 0 0 0 0 1 D C B	(D) gösterge ON/OFF, (C) kursor ON/OFF, (B) kursor pozisyon karakterinin karartılması
Cursor and Display Shift	0 0 0 0 0 1 S/C R/L * *	(S/C) kursor hareketi ve gösterge kayması, (R/L) kayma yönü, DDRAM içeriği değişmez.
Function Set	0 0 0 0 1 DL N F * *	Dahili veri uzunluğu (DL), gösterge satır sayısı (N) ve karakter fontu (F) belirtir.
Set CGRAM Address	0 0 0 1 CGRAM adresi	CGRAM adresini kurar. CGRAM verisi alınması ve gönderilmesi bu komuttan sonradır.
Set DDRAM Address	0 0 1 DDRAM adresi	DDRAM adresini kurar. DDRAM verisi alınması ve gönderilmesi bu komuttan sonradır.
Read Busy Flag & Address	0 1 B/F CGRAM veya DDRAM adresi	BF ve adres sayıcının içeriği okunur.

DDRAM	Gösterge veri belleği
CGRAM	Karakter üretici bellek
BF = 1	Komut kabul edilemez
BF = 0	Komut kabul edilir
N = 0	1 satır
N = 1	2 satır
*	Önemsiz (0 yada 1)
F = 1	5 x 10 karakter fontu
F = 0	5 x 7 karakter fontu
I/D = 0	Kursor pozisyonunu 1 azalt
I/D = 1	Kursör pozisyonunu 1 arttır
S = 0	Gösterge kaymaz
S = 1	Gösterge kayar
C = 0	Kursor yok
C = 1	Kursor var,

Dahili Reset Devresinin Hazırlanması

HD44780 güç kaynağı açıldığı zaman otomatik olarak dahili reset devresini çalıştırır. İlk kullanıma hazırlama işlemi aşağıdaki komutlar ile yapılır. 'Busy Flag' ilk kullanıma hazırlamanın sonuna kadar busy (meşgul) konumundadır, yani BF = 1'dir. BF, meşgul konumuna Vcc gerilimi 4.5V'ta yükseldikten 10 ms sonra geçer.

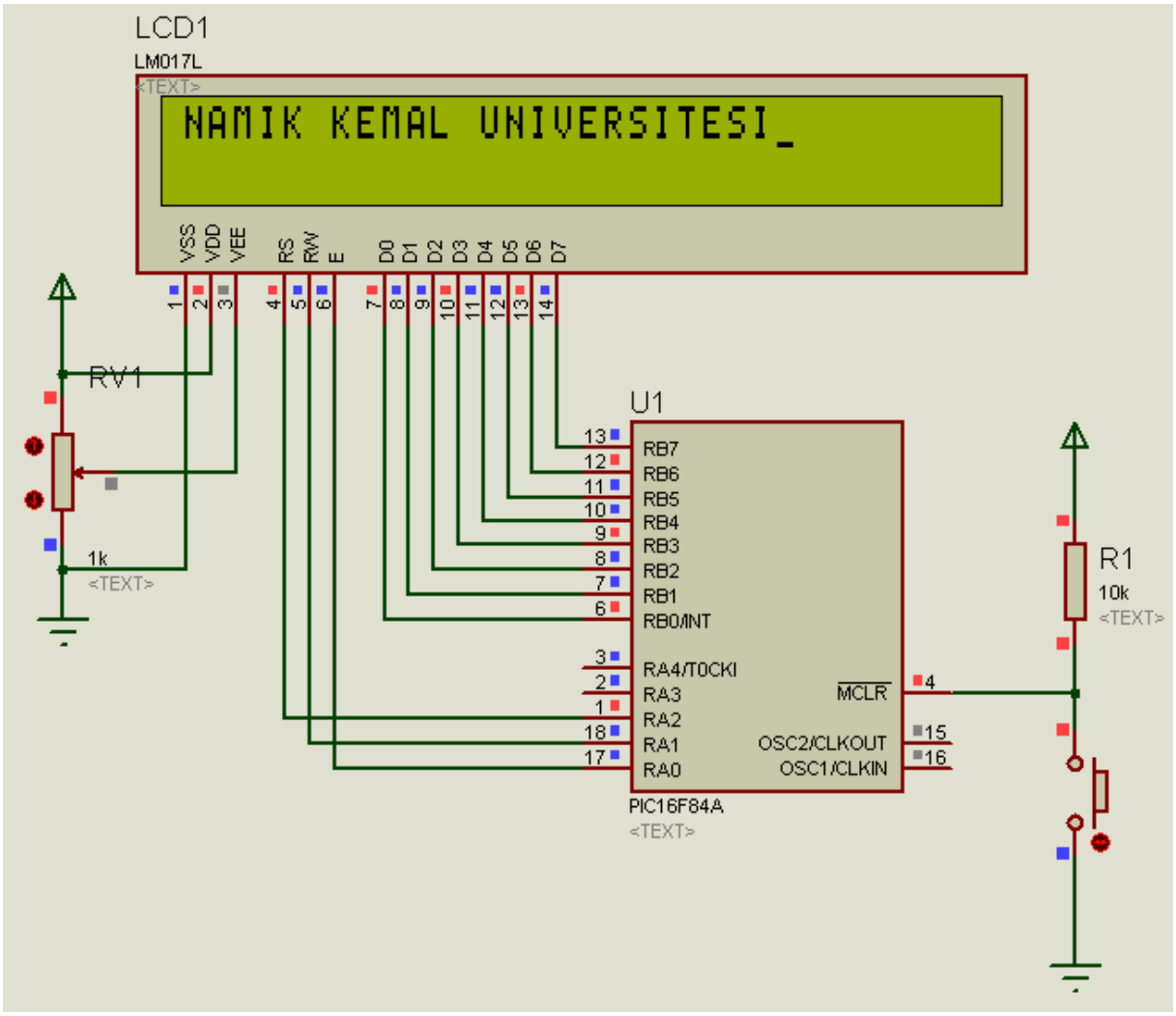
1. Gösterge temizlenir, silinir.
2. Fonksiyon kurulur. DL = 1 :8-bit uzunluğunda veri
   N = 0 : 1 hat gösterge
   F = 0 : 5 x7 nokta karakter fontu
3. Göstergenin ON/OFF kontrolu yapılır. D = 0 : Gösterge OFF
   C = 0 : Kursor OFF
   B = 0 : Karartma OFF
4. Entry Mode kurulur. I/D = 1 : +1 artar
   S = 0 : Kayma yok
5. DDRAM'e veri yazılır

Çalışması

LCD için 4 komut vardır.

1. HD44780'in gösterge formatını, data uzunluğunu, …….vb. düzenleyen komutlar.
2. İç RAM adreslerini veren komutlar.
3. Dahili RAM'den veri transferini sağlayan komutlar.
4. Diğerleri.

Normal kullanımda 3. gruptaki komutlar çok sık kullanılır. Mikroişlemci tarafından her veri için yazılan programda adres bir artar yada azalır. Göstergenin kayması, özellikle göstergeye veri yazarken bir performans artışı sağlayabilir. Bu yüzden göstergeye veri yazılmadan önce BF kontrol edilmelidir. Göstergede bir komut icra edilirken, BF = 1'dir. CGRAM / DDRAM veri yazma komutu icra edildikten sonra veya CGRAM / DDRAM'den veri okunduktan sonra RAM adres sayıcı otomatik olarak 1 artar veya azalır. Bu nedenle BF = 0 olduktan sonra gösterge kayma işlemi icra edilir.



8-Bitlik veri ile LCD ekrana veri yazma devresi

```
TITLE "8-BITLIK DATA İLE LCD EKRANA VERİ YAZMAK"
                LIST        P=16F84
                INCLUDE     "P16F84.INC"

K               EQU         0X2C            ;Sayı dizisi. Display içeriği
CNTL            EQU         0X11            ;Zaman gecikmesi için sayaç
CNTH            EQU         0X12            ;Zaman gecikmesi için sayaç
CX              EQU         0X13            ;Genel amaçlı sayaç
VERICNT         EQU         0X14            ;Veri sayacı
RS              EQU         0X02            ;0 -> Control, 1 -> Data
RW              EQU         0X01            ;0 -> Write, 1 <- Read
EN              EQU         0X00            ; LCD Enable
PC              EQU         0X02            ;Program counter

                org         0x00

BASLA;-----------------------------------------------------------------------
                CLRWDT
                CALL        INITIAL         ;Portları kur
```

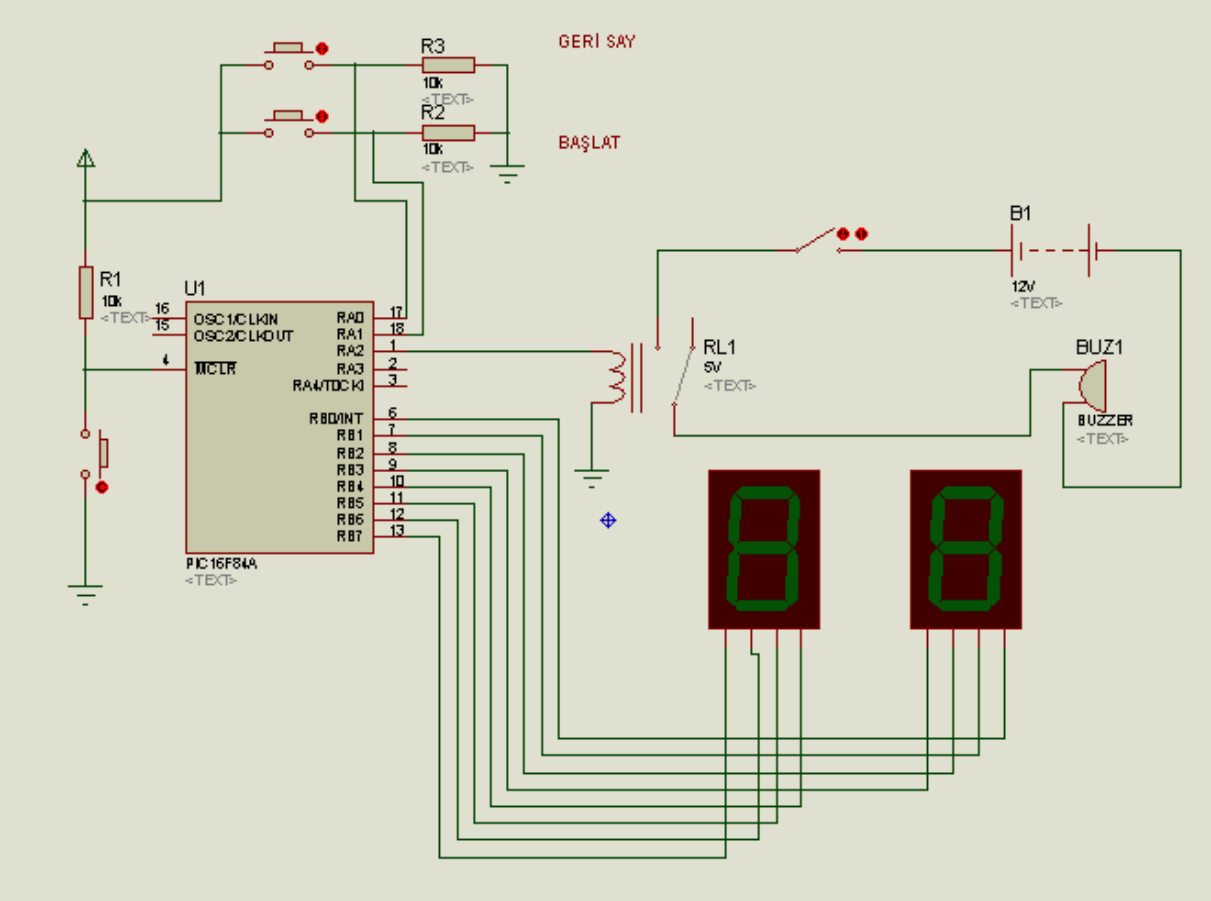
```
                CALL    LCD_RESET       ;LCD reset
                CALL    CLEAR           ;Ekranı temizle
                CALL    TWO_LINE        ;İki satır aktif
                CALL    DISPLAY_ON      ;Display on
                CALL    CURSOR_INC      ;Kursör 1 artan modda
                CALL    CLEAR           ;Ekranı temizle
                CALL    MESAJ           ;Verileri yaz

TEKRAR          GOTO    TEKRAR          ;Programı sonlandır

INITIAL;--------------------------------------------------------------------------
                BSF     STATUS,5        ;Bank1'e geç
                MOVLW   0X00
                MOVWF   TRISB           ;PORTB -> Çıkış
                MOVLW   0X00
                MOVWF   TRISA           ;PORTA -> Çıkış
                BCF     STATUS,5        ;Bank0'a geç
                CLRF    VERICNT         ;VERICNT <- 0
                RETURN

CLEAR;----------------------------------------------------------------------------
                CALL    TIMER_LOW       ;Bekle
                MOVLW   0X01            ;Display'i temizle
                MOVWF   PORTB           ;Kursörü 1.satırın başına al
                BCF     PORTA,RS        ;RS -> Komut
                BCF     PORTA,RW        ;RW -> Yaz
                BSF     PORTA,EN        ;--
                NOP                     ;\
                BCF     PORTA,EN        ;--
                RETURN

LCD_RESET;------------------------------------------------------------------------
                MOVLW   0X03
                MOVWF   CX
NEXTI           CALL    TIMER_LOW       ;Bekle
                BCF     PORTA,RS        ;RS -> Komut
                BCF     PORTA,RW        ;RW -> Yaz
                MOVLW   0X30
                MOVWF   PORTB
                BSF     PORTA,EN        ;--
                NOP                     ;\
                BCF     PORTA,EN        ;--
                DECFSZ          CX,1    ; CX <- CX + 1
                GOTO    NEXTI
                RETURN

TWO_LINE;-------------------------------------------------------------------------
                CALL    TIMER_LOW       ;Bekle
                MOVLW   0X38            ;İki satır aktif
                MOVWF   PORTB
```

```
                BCF         PORTA,RS        ;RS -> Komut
                BCF         PORTA,RW        ;RW -> Yaz
                BSF         PORTA,EN        ;--
                NOP                         ;\
                BCF         PORTA,EN        ;--
                RETURN

CURSOR_INC;-------------------------------------------------------------------
                CALL        TIMER_LOW       ;Bekle
                MOVLW       0x06            ;Kursör 1 artan mod
                MOVWF       PORTB
                BCF         PORTA,RS        ;RS -> Komut
                BCF         PORTA,RW        ;RW -> Yaz
                BSF         PORTA,EN        ;--
                NOP                                 ;\
                BCF         PORTA,EN    ;--
                RETURN

DISPLAY_ON;-------------------------------------------------------------------
                CALL        TIMER_LOW       ;Bekle
                MOVLW       0X0E            ;LCD ON
                MOVWF       PORTB
                BCF         PORTA,RS        ;RS -> Komut
                BCF         PORTA,RW        ;RW -> Yaz
                BSF         PORTA,EN        ;--
                NOP                         ;\
                BCF         PORTA,EN        ;--
                RETURN

MESAJ;------------------------------------------------------------------------
MNEXT           CALL        TIMER_LOW       ;Bekle
                BSF         PORTA,RS        ;RS -> Veri
                BCF         PORTA,RW        ;RW -> Yaz
                MOVF        VERICNT,0       ;W <- VERICNT
                CALL        MESAJ_VERISI    ;Veriyi al
                IORLW       0               ;0 İle test et
                BZ          MEND            ;0 ise MEND etiketine git
                MOVWF       PORTB           ;0 değilse veriyi PORTB'ye yaz
                INCF        VERICNT,1       ; bir sonraki veriyi adresle
                BSF         PORTA,EN        ;--
                NOP                         ;\
                BCF         PORTA,EN        ;--
                GOTO        MNEXT           ;Veri yazma işine devam et
MEND            RETURN


MESAJ_VERISI;-----------------------------------------------------------------
                ADDWF       PCL,1
                DT          "NAMIK KEMAL UNIVERSITESI ",0
```

```
TIMER_LOW;-------------------------------------------------------------------------
          MOVLW     0XB0      ;LCD işlem yaparken gerekli
          MOVWF     CNTH      ;zamanı sağlamak için
NEXT2     MOVLW     0XFF      ;bekle
          MOVWF     CNTL      ;COUNT <- 0XFF
NEXT1     DECFSZ    CNTL,F    ;COUNT=COUNT-1, C=0 mı?
          GOTO      NEXT1     ;Hayır.next1'e git
          DECFSZ    CNTH,F    ;
          GOTO      NEXT2
          RETURN              ;Evet.Timer alt programından çık

          END
```

DENEY 11: Zamanlayıcı (60-1 saniye ayarlı)

60 s süre içerisinde, istenilen süreye ayarlanabilen ve ayarlanan süreden itibaren birer saniye aralıklarla geri sayarak süre tamamlandığında bir zili çalan devre ve programı.



;RA0 ucu ayar butonu her bas. ta 1 geri sayd.r.r
;RA1 ucu zaman ba latma butonudur, bas.l.nca sistem
;geri sayar ve di er buton iptal olur
;Süre 0 olunca RA2 ucundaki zil ç.k. . 1 olur
;B portunda 2 adet 4056 ile iki display ba l.d.r

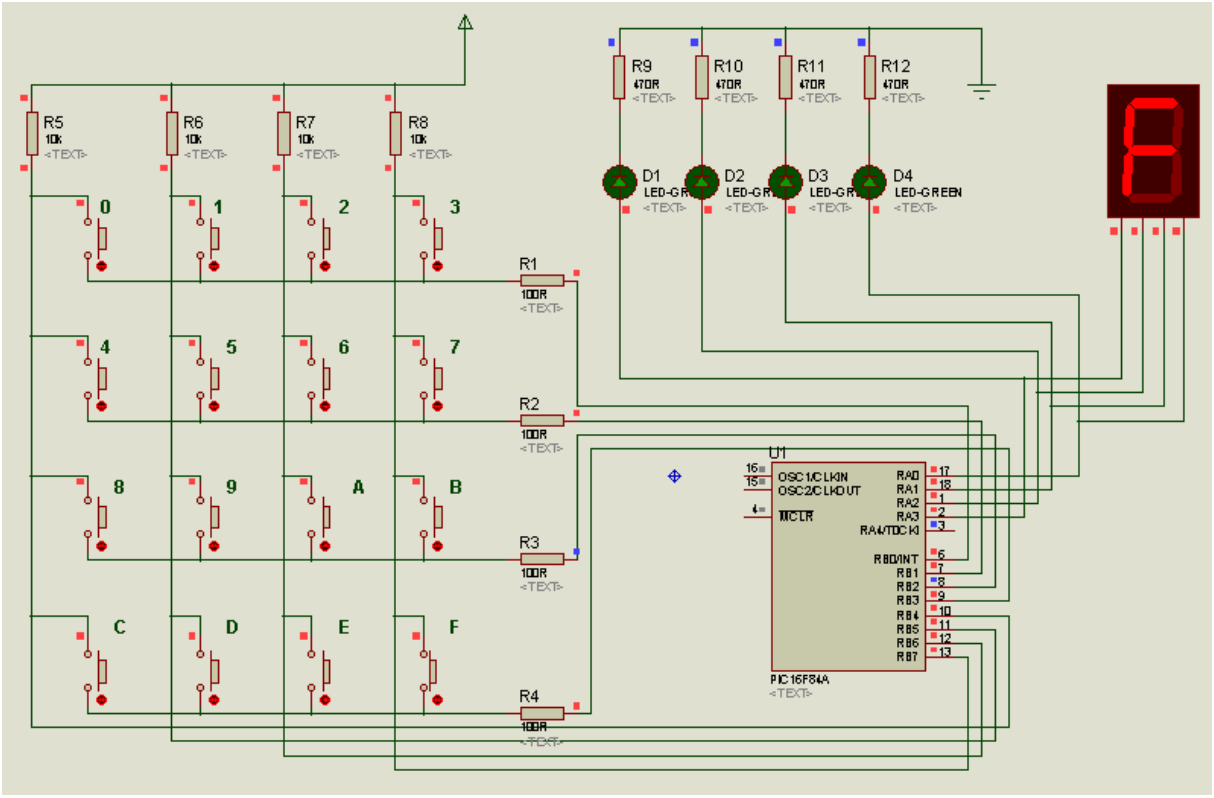
```
                LIST    P=16F84
                INCLUDE     "P16F84.INC"

sayi            EQU         0Ch         ; PORTB'ye gönderilecek say.
D1              EQU         0Eh         ; BEKLEME SAYACI 1
D2              EQU         0Fh         ; BEKLEME SAYACI 2
SAYAC1          EQU         10h
SAYAC2          EQU         11h
X1              org         0h          ; Power on
                goto        START       ; 0000

START           bsf         STATUS,5
                movlw       0h
                movwf       TRISB
```

```
        movlw   03h
        movwf   TRISA
        bcf     STATUS,5

        movlw   60H
        movwf   sayi
TOP     movf    sayi,W
        movwf   PORTB

ayar    btfsc   PORTA,0
        goto    eksil
ates    btfss   PORTA,1
        goto    ayar
say     movf    sayi,w
        movwf   PORTB
        call    BEKLE

        decf    sayi,1
        movf    sayi,W
        andlw   0Fh
        sublw   0Fh
        btfss   STATUS,2
        goto    sifir
onluk   decf    sayi,1
        decf    sayi,1
        decf    sayi,1
        decf    sayi,1
        decf    sayi,1
        decf    sayi,1
        goto    say
sifir   movf    sayi,W
        sublw   0h
        btfss   STATUS,2
        goto    say
dur     bsf     PORTA,2
        goto    dur

eksil   call    GECIKME
        decf    sayi,1
        movf    sayi,W
        andlw   0Fh
        sublw   0Fh
        btfss   STATUS,2
        goto    sifir2
onluk2          decf    sayi,1
        decf    sayi,1
        decf    sayi,1
        decf    sayi,1
        decf    sayi,1
        decf    sayi,1
```

```
                goto        yaz

sifir2          movf        sayi,W
                sublw       0h
                btfss       STATUS,2
                goto        yaz
                goto        TOP

yaz             movf        sayi,w
                movwf       PORTB
                goto        ayar

;*******************************************************
; BEKLETME ALT PROGRAMI
;*******************************************************

BEKLE           movlw       .200        ;200*200 lük iki döngü yakla .k 1 saniye
                movwf       D1          ;kabul edilecektir.

ZD1             movlw       .200
                movwf       D2
ZD2             decfsz      D2,1
                goto        ZD2
                decfsz      D1,1
                goto        ZD1
                return
GECIKME ;=============================================================
                MOVLW       H'FE'       ; W <-- h'4F'
                MOVWF       SAYAC1      ; Sayac1 <-- W
DONGU1
                MOVLW       H'FE'
                MOVWF       SAYAC2      ; Sayac2 <--h'00'
DONGU2
                DECFSZ      SAYAC2,1
                GOTO        DONGU2
                DECFSZ      SAYAC1,1
                GOTO        DONGU1
                RETURN

                END
```

DENEY 12: Klavye (Tarama Yöntemi)

4x4'lük bir tuş takımı kullanarak tarama yöntemi ile basılan tuş bilgisini PortA'ya bağlı LED'lerde binary olarak gösteren devre ve programı.



Program önce ilk satır (RB0) ve ilk sütundan (RB4) taramaya başlar. Bu satırdaki tuşlar "0", "1", "2" ve "3"tür. Sıra ile sütunlar kontrol edilir. "0" sayısını elde etmek için ilk satırı tararken satır sayıcı =0, sütun sayıcı=0 ataması, programda tuş tespitini basitleştirir. 0. sütun basılı değil ise program sütun sayısını bir artırır ve 1. sütuna bakar. Eğer bu sütundaki tuşa basılı ise basılan tuşun tespiti;

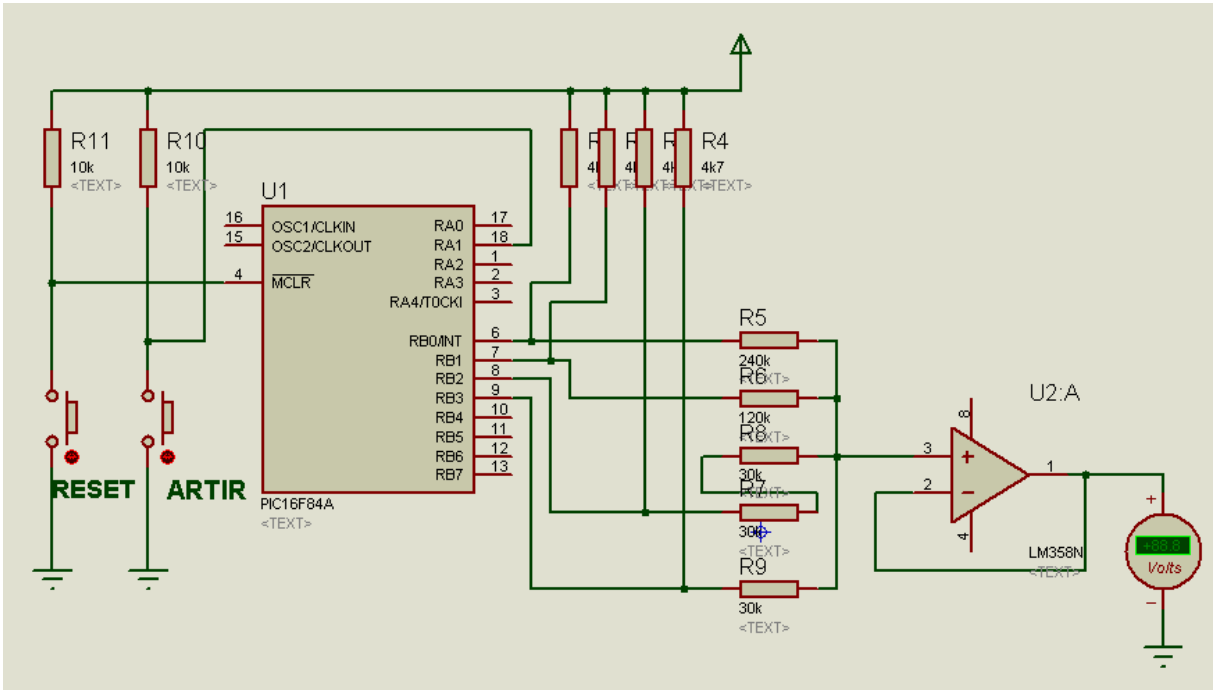
tus=satir+sutun
tus=0+1
tus=1

mantığı ile hesaplanır.Program ilk satırı taradığı anda herhangi bir tuşa basılmadıysa row bilgisini bir sola kaydırarak ikinci satırı lojik-0, diğerlerini lojik-1 yapar. Bu satırdaki tuşlar ise "4", "5", "6" ve "7"dir. Yani ilk satırdakilerin 4 fazlasıdır. Bu yüzden satır sayıcı register'ının içeriği diğer bir satıra geçildiğinde 4 artırılır ve tarama işlemi 4. satır taranana kadar devam eder. Bir diğer satırdaki rakamları test ederken satır bilgisi 4 fazla olarak başlatılmalıdır.

```
;========================== KLAVYE.ASM ==========================
;* Bu program PIC 16F84 ile  4x4 klavye tasarımını gösterir.                *
;* PortA'ya bağlı 4 adet LED ile tuş bilgisi gözlenir.                      *
;================================================================
                LIST    P=16F84
                INCLUDE "P16F84.INC"

                __CONFIG  _CP_OFF&_WDT_OFF&_PWRTE_OFF&_XT_OSC
SATIR           EQU     H'0C'
SUTUN           EQU     H'0D'
TUS             EQU     H'0E'
ROW             EQU     H'0F'

                ORG     H'00'
MAIN
                CALL    INITIAL
TEKRAR          CALL    TUS_TARA
                CALL    LED_OUT
                GOTO    TEKRAR
INITIAL ;-------------------------------------------------------------------------------------------
                BSF     STATUS,5        ; Bank1'e geç
                MOVLW   H'F0'           ; RB0-RB3 çıkış
                MOVWF   TRISB           ; RB4-RB7 giriş
                MOVLW   H'00'           ; RA0-RA4 çıkış
                MOVWF   TRISA
                CLRF    PORTA           ; PortA ←H'00'
                CLRF    TUS             ; tus ← H'00'
                BCF     STATUS,5        ; Bank0'a geç
                RETURN

TUS_TARA ;------------------------------------------------------------------------------------------
                CLRF    SATIR           ; satır=0
                MOVLW   H'0E'           ; 00001110
                MOVWF   PORTB           ; 0. satır aktif
                MOVWF   ROW
SUTUN_OKU
                CLRF    SUTUN           ; sutun=0
                BTFSC   PORTB,4         ; 0. sütundaki tuşa basılı mı?
                GOTO    BIR             ; Hayır, diğer sütuna geç
                GOTO    RAKAM           ; Evet
BIR
                INCF    SUTUN,F         ; sutun=1
                BTFSC   PORTB,5         ; 1. sütundaki tuş basılı mı?
                GOTO    IKI             ; Hayır, diğer sütuna geç
                GOTO    RAKAM           ; Evet
IKI
                INCF    SUTUN,F         ; sutun=2
                BTFSC   PORTB,6         ; 2. sütundaki tuş basılı mı?
                GOTO    UC              ; Hayır, diğer sütuna geç
                GOTO    RAKAM           ; Evet
```

```
UC
                INCF    SUTUN,F         ; sutun=3
                BTFSC   PORTB,7         ; 3. sütundaki tuş basılı mı?
                GOTO    DIGER_SATIR     ; Hayır, diğer satıra geç
                GOTO    RAKAM           ; Evet
DIGER_SATIR
                BSF     STATUS,0        ; Carry=1
                RLF     ROW,F           ; Bir bit “0” bilgisini sola kaydır
                BTFSS   ROW,4           ; 4 satırda tarandı mı?
                GOTO    TUS_TARA_SON    ; Evet, alt programdan çık
                MOVF    ROW,W           ; Hayır
                MOVWF   PORTB           ; satırı “0” yap
                MOVLW   H'04'           ; W ← H’04’
                ADDWF   SATIR,F         ; satır=satır+4
                GOTO    SUTUN_OKU       ; bu satıra ait sütunları tara
RAKAM
                MOVF    SATIR,W
                MOVWF   TUS
                MOVF    SUTUN,W
                ADDWF   TUS,F           ; tus=satır+sutun
TUS_TARA_SON
                RETURN
LED_OUT ;-------------------------------------------------------------------------------------------
                MOVF    TUS,W           ; W ← tus
                MOVWF   PORTA           ; tus → PortA
                RETURN
                END
;================================================================
```

DENEY 13: DAC (Merdiven Direnç Devresi Kullanarak)

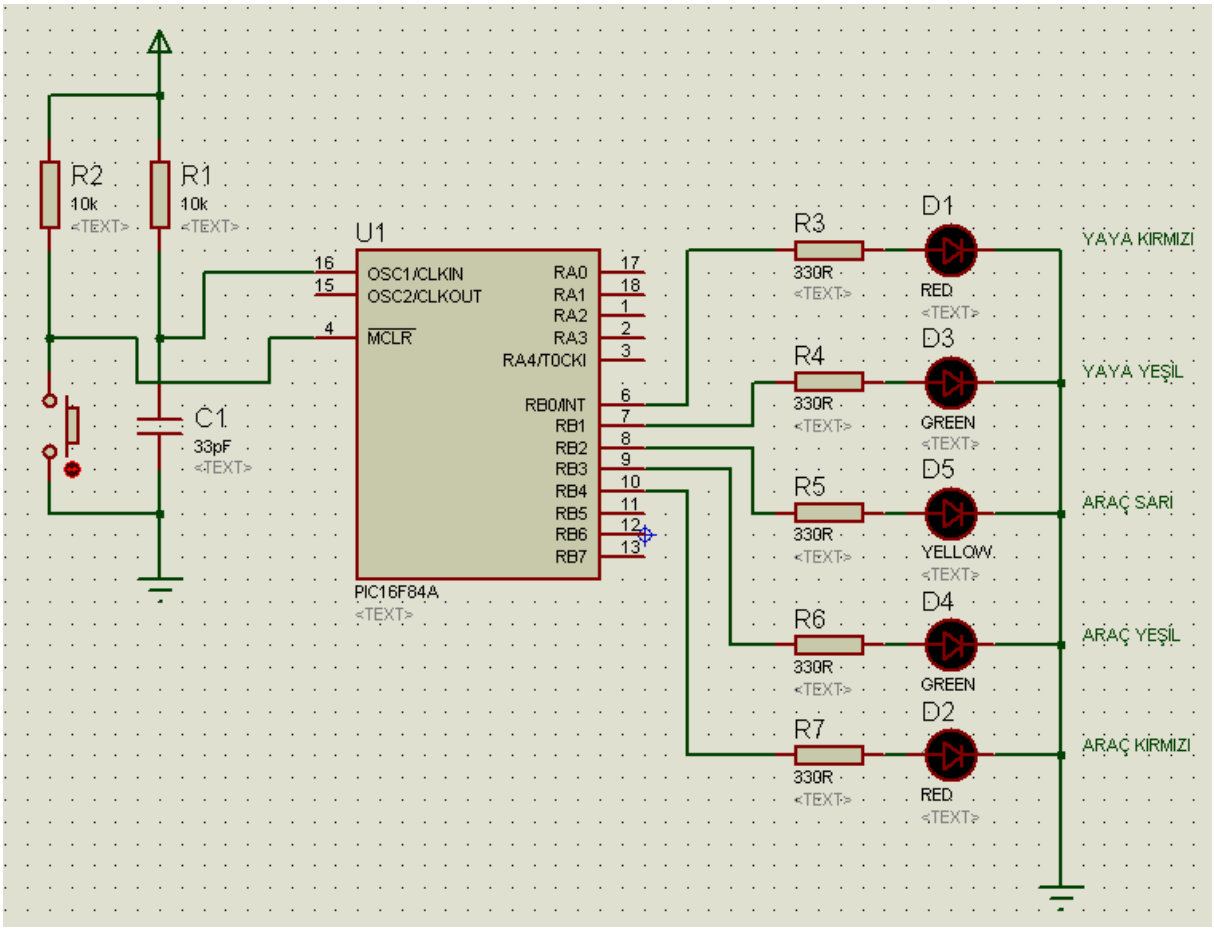



```
            LIST    P=16F84
            INCLUDE "P16F84.INC"
ARTIR       EQU     H'0C'
SAYAC1      EQU     H'0D'
SAYAC2      EQU     H'0E'

BASLA       ;-------------------------------------------------------------------------
            CLRF    PORTB       ; PortB'yi sil

            BSF     STATUS,5    ; Bank1'e geç
            CLRF    TRISB       ; PortB'nin tüm uçlarını çıkış yap
            MOVLW   H'FF'       ; PortA'nın tüm uçlarını giriş yap
            MOVWF   TRISA
            CLRF    ARTIR       ; ARTIR=0
            BCF     STATUS,5    ; Bank0'a geç

TEST        ;-------------------------------------------------------------------------
            BTFSC   PORTA,1     ; Artır butonuna basıldı mı?
            GOTO    TEST        ; Hayır, butonu test et
            CALL    GECIKME     ; Gecikme alt programını çağır
            INCF    ARTIR,F     ; Artır ← Artır + 1
            MOVF    ARTIR,W     ; W ← Artır

            CALL    CEV_TAB     ; Cevrim tablosunu çağır
            MOVWF   PORTB       ; Çevrim tablosundan aldığın değeri
                                ; PortB'ye gönder
            GOTO    TEST        ;
```

```
GECIKME ;-------------------------------------------------------
        MOVLW   H'8F'
        MOVWF   SAYAC1
DONGU1
        MOVLW   H'FF'
        MOVWF   SAYAC2
DONGU2
        DECFSZ  SAYAC2,F
        GOTO    DONGU2
        DECFSZ  SAYAC1,F
        GOTO    DONGU1
        RETURN

CEV_TAB ;-------------------------------------------------------
        ADDWF   PCL,F       ; PCL <-- PCL + W

        RETLW   H'00'       ; 0.00 V
        RETLW   H'01'       ; 0.33 V
        RETLW   H'02'       ; 0.67 V
        RETLW   H'03'       ; 1.00 V
        RETLW   H'04'       ; 1.33 V
        RETLW   H'05'       ; 1.67 V
        RETLW   H'06'       ; 2.00 V
        RETLW   H'07'       ; 2.33 V
        RETLW   H'08'       ; 2.67 V
        RETLW   H'09'       ; 3.00 V
        RETLW   H'0A'       ; 3.33 V
        RETLW   H'0B'       ; 3.67 V
        RETLW   H'0C'       ; 4.00 V
        RETLW   H'0D'       ; 4.33 V
        RETLW   H'0E'       ; 4.67 V
        RETLW   H'0F'       ; 5.00 V

        END
```

Örnek 1: Bir kavşaktaki trafik ışıklarının aşağıda verilen zaman ve sıra içerisinde çalışması isteniyor. Trafik ışığı olarak LED kullanarak gerekli devreyi tasarlayınız ve programını yazınız.

Süre (Saniye	Araç	Yaya
15	Yeşil	Kırmızı
5	Sarı	Kırmızı
25	Kırmızı	Yeşil
5	Sarı ve Kırmızı	Kırmızı



Devre şekli

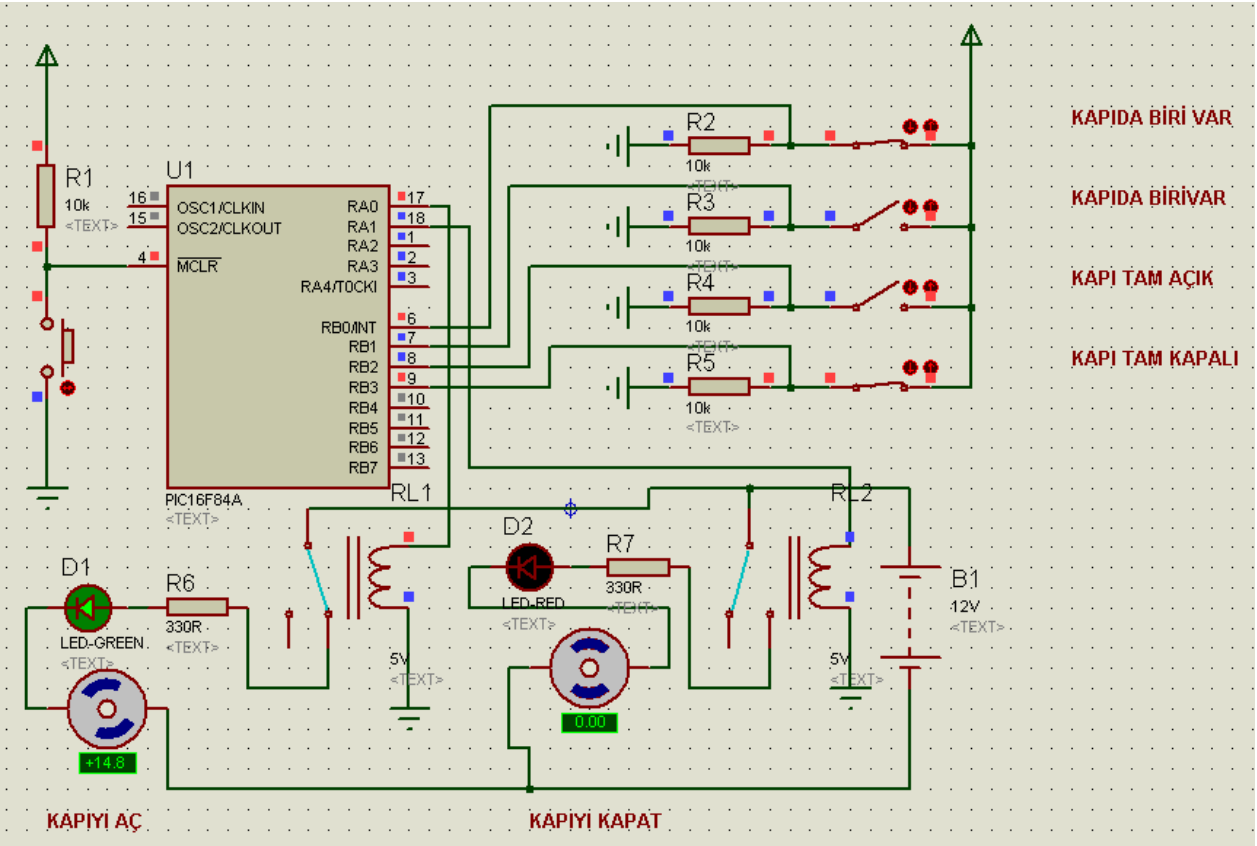
Çözüm: Devre şekli yukarıda verilmiştir. Zamanlama için 200*200=40000 turluk bekleme döngüsünü 1saniye olarak kabul ediyoruz.

```
;TRAFIK.ASM
;
;Lambalar: RB0: Yaya-Kırmızı, RB1: Yaya-Yeşil
;RB2:Araç-Sarı, RB3:Araç-Yeşil, RB4:Araç-Kırmızı
;Süre ve Durumlar
;15 s Araç-Yeşil, Yaya-Kırmızı
;  5 s Araç-Sarı, Yaya-Kırmızı
;15 s Araç-Kırmızı, Yaya-Yeşil
;15 s Araç-Sarı-Kırmızı, Yaya-Kırmızı

            LIST    P=16F84
            INCLUDE "P16F84.INC"

; Değişkenler
ZD1         EQU     0Fh
ZD2         EQU     0Eh
zaman       EQU     0Dh

X1          org     0h              ; Power on
            goto    START           ; 0000

START       bsf     STATUS,5        ; Page 1
            movlw   0h              ; 0000-0000 sayısını  W registerine al
            movwf   TRISB           ; PortB yi çıkış  olarak ayarla
                                    ;TRISB=00000000
            bcf     STATUS,5        ; Page 0

TOP         movlw   09h             ; 0 0 0 0 1 0 0 1
            movwf   PORTB           ; Araç-Yeşil, Yaya- Kırmızı
            movlw   0Fh
            movwf   zaman
            call    BEKLE           ; Bekle 15 saniye

            movlw   05h             ; 0 0 0 0 0 1 0 1
            movwf   PORTB           ; Araç-Sarı , Yaya-Kırmızı
            movlw   05h
            movwf   zaman
            call    BEKLE           ; Bekle 5 Saniye

            movlw   12h             ; 0 0 0 1 0 0 1 0
            movwf   PORTB           ; Araç-Kırmızı , Yaya-Yeşil
            movlw   19h
            movwf   zaman
            call    BEKLE           ; Bekle 25 saniye

            movlw   15h             ; 0 0 0 1 0 1 0 1
            movwf   PORTB           ; Araç-Sarı-Kırmızı , Yaya-Kırmızı
            movlw   05h
            movwf   zaman
            call    BEKLE           ; Bekle 5 Saniye
```

```
                goto            TOP                     ; Tekrarla

;*******************************************************
; BEKLE alt programı
;*******************************************************

BEKLE           movlw           .200
                movwf           ZD1
D1              movlw           .200
                movwf           ZD2
D2              decfsz          ZD2,F
                goto            D2
                decfsz          ZD1,F
                goto            D1
                decfsz          zaman,F
                goto            BEKLE
                return
                END
```

Örnek 2: Bir süpermarkette kapıların otomatik çalışması istenmektedir. Bu iş için kullanılacak optik sensörler görüş açısındaki cismi algılayınca 1, boşta iken 0 vermektedirler. Bir kapının giriş-çıkış şeklinde çalışması için gerekli devreyi tasarlayınız. Kapı ortasında kimsenin sıkışmaması için gerekli tedbiri alınız.

Çözüm: Burada en az 3 sensöre ihtiyaç olacaktır. Bunlar kapıda biri olduğunu algılayan sensör ile kapı tam açık ya da kapı tam kapalı şeklindeki değerleri veren limit switch şeklindeki kapı açık-kapalı sensörleridir. Bunlar olduğunda problem kapıda biri varsa ve kapı tam açık değilse kapı açma motorunu çalıştır. Kapıda kimse yoksa ve kapı kapalı değilse kapı kapatma motorunu çalıştır şeklinde düşünülür.



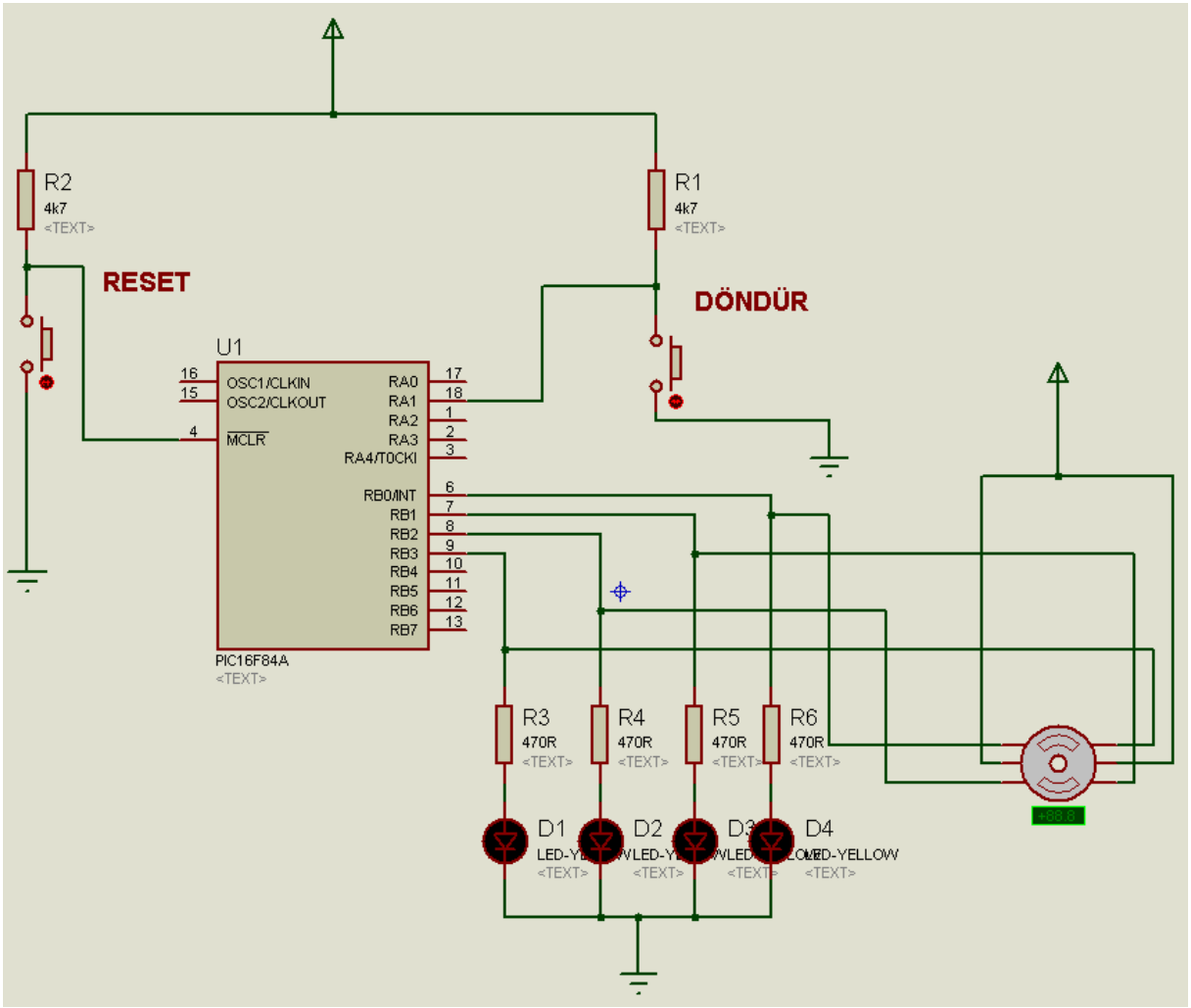
Otomatik Kapı Devresinin şekli

;**O_KAPI.ASM**
;RB0,RB1 uçlarında kapıda biri var işareti veren sensörler bağlı
;RB2 de kapı tam açık sensörü bağlı
;RB3 te kapı tam kapalı sensörü bağlı
;RA0 ucunda kapıyı açan motor bağlı
;RA1 ucunda kapıyı kapatan motor bağlı

```
            LIST   P=16F84
            INCLUDE    "P16F84.INC"

X1          org        0h          ; Power on
            goto       START       ; 0000
```

```
START   bsf     STATUS,5
        movlw   h'FF'       ;B portu giriş
        movwf   TRISB
        movlw   00h         ;A portu çıkış
        movwf   TRISA
        bcf     STATUS,5

sil     clrf    PORTA

kontrol movf    PORTB,w
        andlw   03h         ;sadece RB0 ve RB1 bilgilerini ayırmak için
        btfsc   STATUS,2    ;kapıda biri yoksa ANDLW işleminin sonucu
                            ;0 dır
        goto    kapat

ac      btfsc   PORTB,2
        goto    sil
        movlw   01h
        movwf   PORTA
        goto    kontrol

kapat   btfsc   PORTB,3
        goto    sil
        movlw   02h
        movwf   PORTA
        goto    kontrol

        END
```

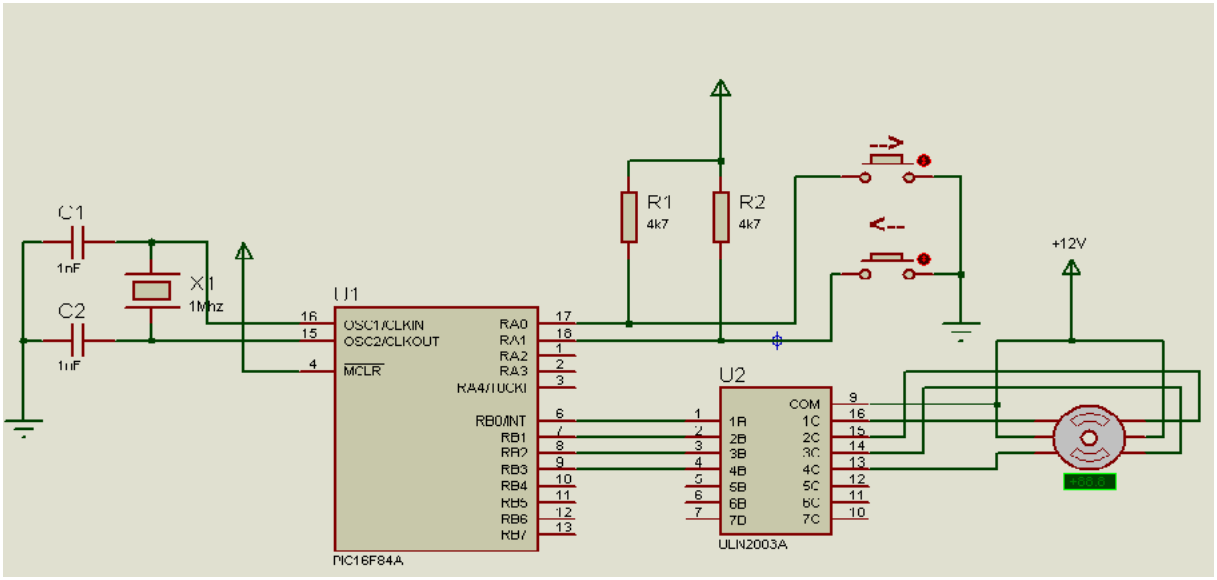
DENEY 14: Adım Motor Sürücüsü (Tek Yönlü)




```
;==============Step Motor Kontrol======
                LIST        P=16F84
                INCLUDE     "P16F84.INC"
SAYAC1          EQU         H'0C'
SAYAC2          EQU         H'0D'
ADIM            EQU         H'0E'

                CLRF        PORTB       ; PortB'yi sil
                BSF         STATUS,5    ; Bank1'e geç
                MOVLW       H'FF'       ; W ←H'FF'
                MOVWF       TRISA       ; PortA'nın tüm uçları giriş
                CLRF        TRISB       ; PortB'nin tüm uçları çıkış
                BCF         STATUS,5    ; Bank0'a geç
                MOVLW       H'FF'       ; W ←H'FF'
                MOVWF       ADIM        ; Adım ← W (H'FF')
```

```
BASLA   ;-----------------------------------------------
        BTFSC   PORTA,1     ; RA1'e bağlı butona basıldı mı?
        GOTO    BASLA       ; Hayır, butonu test et
        INCF    ADIM,F      ; Evet, Adım = Adım + 1
        MOVF    ADIM,W      ; W ← Adım
        ANDLW   B'00000111' ; W'nin üst 5 bitini maskele
        CALL    ADIMTBL     ; Tablodan bit biçimini seç
        ANDLW   B'00001111' ; W'nin üst 4 bitini maskele
        MOVWF   PORTB       ; Bit biçimini PortB'de göster
        CALL    GECIKME     ; Gecikme yap
        GOTO    BASLA       ; Yeni bir bit biçimine git

ADIMTBL ;-----------------------------------------------
        ADDWF   PCL,F       ; PCL = PCL + W

        RETLW   B'0001'
        RETLW   B'1001'
        RETLW   B'1000'
        RETLW   B'1010'
        RETLW   B'0010'
        RETLW   B'0110'
        RETLW   B'0100'
        RETLW   B'0101'

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
```

DENEY 15: Adım Motor Sürücüsü (Çift Yönlü)




```
                LIST        p=16F84

                #include    "P16F84.INC"        ; Include header file

                CBLOCK      0x10                ; Temporary storage
                  pos
                  dc1
                  dc2
                ENDC

                LIST        p=16F84
                #include    "P16F84.INC"

                CBLOCK      0x10                ; Geçici depolama
                ENDC

                ORG         0
entrypoint      goto        start

                ORG         4
intvector       goto        intvector

start           clrw                            ; W ← h'00'.

                movwf       PORTB               ; PORTB ← W
                bsf         STATUS,RP0          ; Bank 1'i seç
                movlw       0xF0                ; Port B'nin  0-3 bitleri çıkış olarak ayarla
                movwf       TRISB               ; TRISB register'ini ayarla.

                bcf         STATUS,RP0          ; Bank 0'ı seç
```

```
                movlw   3               ; Motor pozisyonunu başlat
                movwf   pos
                movwf   PORTB
                call    delay
                clrf    PORTB           ; Motor sürücü kapalı

;Main loop -------------------------------------------------------------------------------------------
loop            btfss   PORTA,0         ; Saat yönü butonunu test et
                call    stepcw
                btfss   PORTA,1         ; Saat yönün tersi butonunu test et
                call    stepccw
                 goto   loop

;Bir adım saat yönünde döndürme ----------------------------------------------------------------------
stepcw          bcf     STATUS,C        ; Carry bayrağını sil
                btfsc   pos,3           ; Eğer bu bit "1" ise carry'i ayarla
                bsf     STATUS,C
                rlf     pos,W           ; İlerle ve Motorun temel pozisyonuna döndür
                andlw   0x0F            ; Alt dört biti maskele
                movwf   pos
                movwf   PORTB           ; Çıkışları sür
                call    delay           ; Bekle
                clrf    PORTB           ; Çıkışı temizle
                return

;Bir adım sayısı saat yönünde döndür -----------------------------------------------------------------

stepccw         bcf     STATUS,C        ; Carry bayrağını sil
                btfsc   pos,0
                bsf     pos,4
                rrf     pos,W           ; İlerle ve Motorun temel pozisyonuna döndür
                andlw   0x0F            ; Alt dört biti maskele
                movwf   pos
                movwf   PORTB           ; Çıkışları sür
                call    delay           ; Bekle
                clrf    PORTB           ; Çıkışı temizle
                return

; Adımlar arası gecikmeyi sağlama ve motor hızı kontrolleri -------------------------------------------
delay           movlw   18              ; Dış döngü sayısı
                movwf   dc1
dl1             clrf    dc2             ; İç döngüyü başlatma
dl2             nop
                nop
                decfsz  dc2,F
                goto    dl2
                decfsz  dc1,F
                goto    dl1
                return
                END
```