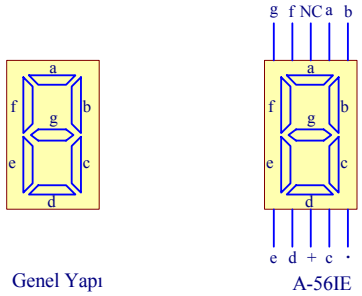
4 Display

- 7 Segment Display ve Tarama İşlemi
- Tarama Yöntemi ile Display'lere 1234 Yazmak
- Register Kullanarak Display'e Veri Yazmak
- Array Değişken ile Verileri Display'de Görüntülemek

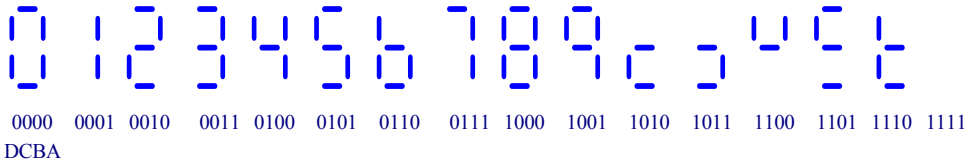
7 SEGMENT DISPLAY ve TARAMA İŞLEMİ

Önceki bölümlerde yapılan devrelerde sadece RA portuna bağlanan LED'ler çıkış olarak kullanılmıştı. Bu bölümde ise 4 adet 7 segment display çıkış olarak kullanılacaktır. 7 segment display'in pin isimleri aşağıdaki Şekil 4.1'de görüldüğü gibidir.



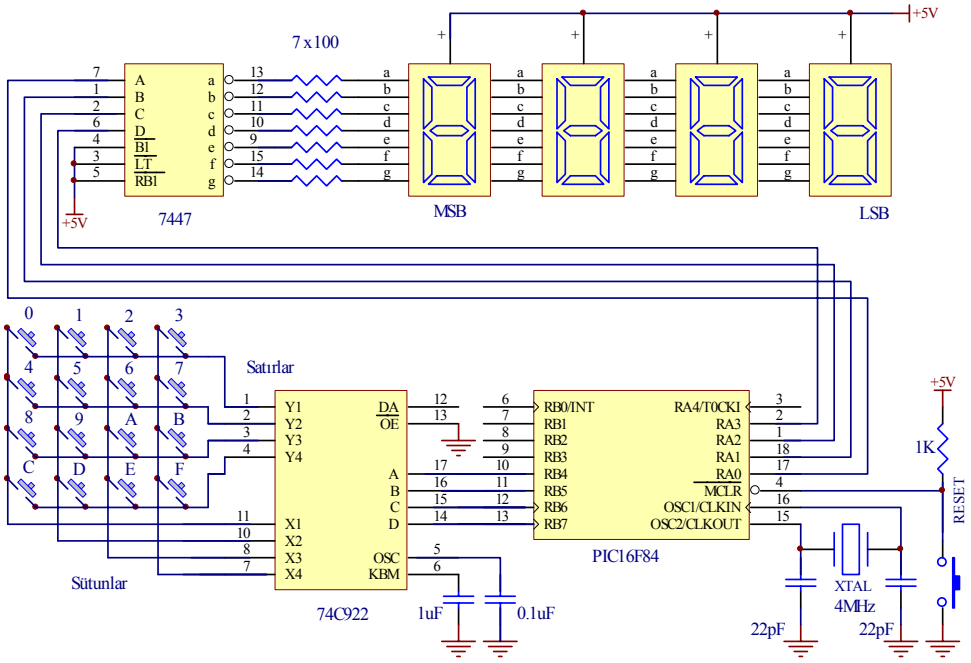
Şekil 4.1 7 Segment display'in yapısı ve A56IE ortak anod displayin pin isimleri

Burada kullanılan display'ler ortak anoddur. A-56IE ortak anodlu display'in ayak bağlantısı yukarıda örnek olarak gösterilmiştir. Bunları sürmek için bir 7447 entegresi kullanılır. 7447 kullanılmasının sebebi, rakamlardan sonraki veriler (9'dan sonraki semboller) için de çıkış verebilmesidir. Şekil 4.2'de bu semboller görülmektedir. Böylece 4x4 klavyedeki her tuşa (F hariç) ait bir sembol display'lerde görülebilir.



Şekil 4.2 7447 ile elde edilen çıkış işaretleri

Uygulamada sadece bir adet 7447 display sürücü entegresi kullanılmıştır. Oysa her display için bu entegreden bir adet gerekmektedir. 4x4 klavye devresindeki gibi tarama yöntemi kullanan program ile 4 adet display'i tek sürücü ile kontrol etmek mümkündür. Bunun için display'lerin ortak anod uçları hariç diğer segment uçları kısa devre edilerek 100 ohm'luk direnç üzerinden 7447'ye bağlanır. Bu durumda 7447'den gelen bilgi hangi display'in ortak anodunda gerilim varsa o display'de görülecektir. Devre kurulduktan sonra test için tüm display'lerin ortak anod uçları +5V'a bağlanarak display'lerde aynı sayıyı elde etmeye çalışılmalı ve bağlantı hataları ortadan kaldırılmalıdır. RBHigh konusunda anlatılan programı hiç değiştirmeden -aynı programlanmış PIC 16F84 ile- Şekil 4.3'de verilen devre kurulursa, klavyeden basılan tuşun değeri 4 display'de de aynı şekilde görülecektir.

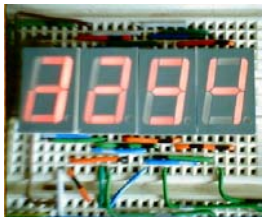


Şekil 4.3 4 Adet display'in paralel bağlanarak kontrolünün yapılması

İlk yapacağımız programda "1234" rakamlarını display'lerde görüntüleyeceğiz. 1 rakamını 7447'den gönderirken sadece MSB (en yüksek değerlikli, en soldaki) display'in ortak anodunda gerilim olacak, diğerlerinde gerilim olmayacaktır. 2 rakamı ise onun sağındaki display'de gözükmesi gerektiğinden bu display aktif olurken diğerleri aktif olmayacaktır. Bu durumu tablo halinde gösterirsek;

7447 nin giriş bilgisi	Ortak Anod Voltajları			
	4.Display	3.Display	2.Display	1.Display
1 rakamı	5V	0V	0V	0V
2 rakamı	0V	5V	0V	0V
3 rakamı	0V	0V	5V	0V
4 rakamı	0V	0V	0V	5V

olur. Sıralı bilgilerin ilgili display'e gönderilmesi ve bu işin sürekli tekrar edilmesi durumunda display'lerden elde edilecek görüntü Şekil 4.4'deki gibi olur.

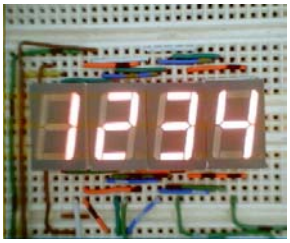


Şekil 4.4 Sürekli taramada elde edilen görüntü

Görüldüğü gibi display'lerdeki görüntü net olmamaktadır. Bunun sebebi display içerisindeki fosfor tabakası aydınlanmadan diğer display'in çalışmasıdır. Oysa display'in çok kısa bir süre aynı bilgi ile beslenmesi, içerisindeki fosforu belirginleştirecek ve enerjisi kesilse bile sönmesi yavaş olacaktır. Sönmeden ikinci kez aynı bilgi ile tazelendiği için görüntü durağan olacaktır. (İnsan gözü saniyede 16'dan fazla tekrar eden aynı hareketi durağan olarak algılar). Eğer;

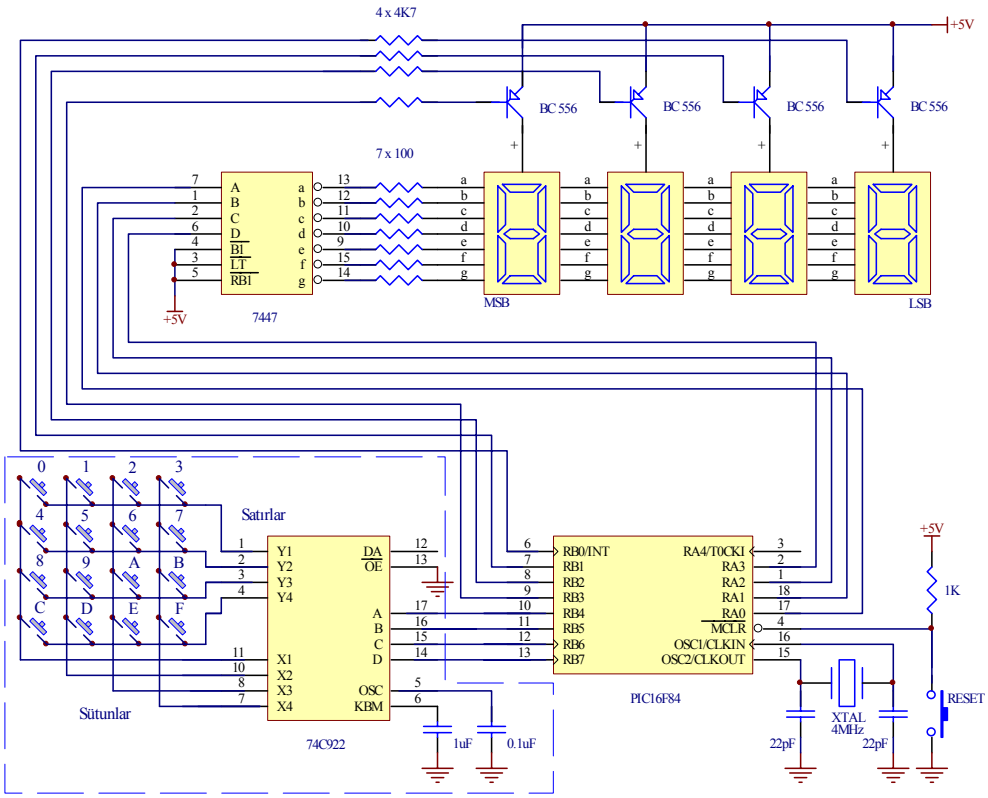
7447 nin giriş bilgisi	Ortak Anod Voltajları			
	4.Display	3.Display	2.Display	1.Display
1 rakamı	5V	0V	0V	0V
		zaman geçikmesi		
2 rakamı	0V	5V	0V	0V
		zaman geçikmesi		
3 rakamı	0V	0V	5V	0V
		zaman geçikmesi		
4 rakamı	0V	0V	0V	5V
		zaman geçikmesi		

şeklinde tarama yapılırsa displayler'deki görüntü Şekil 4.5'de görüldüğü gibi olur.



Şekil 4.5 Belirli bir zaman süresince bilgi bekletilen display'lerdeki görüntü

Devre Şeması



Şekil 4.6 4 display kullanarak "1234" bilgisini görüntüleyen uygulama devresi

Not: Kesik çizgiler içinde kalan kısım bir önceki devrede kullanılan klavye devresidir. Bu uygulamada ise herhangi bir fonksiyonu yoktur. Ancak bir sonraki uygulamada kullanılacağı için board üzerinde kalmasında sakınca yoktur.

Devrede RA portu yine çıkış olarak kullanılmıştır. Ancak buraya binary – BCD seven segment kod çözücü entegresi bağlanmıştır. Her bir display'in ortak anot ucu PNP bir transistörle kontrol edilmektedir. RB portunun low kısmı bu transistörleri sürmektedir. Burada bir transistörün iletime geçirilmesi için beyz ucunun lojik-0 olması gerekir. İletime geçmesi istenmeyen dijitler içinde lojik-1 gönderilir. Bu durum kullanılan transistörün yapısından dolayı yukarıdaki tablolara göre ters gözükse de mantık olarak doğrudur.

PROGRAMLAR

Bu konuda aynı devreye ait üç değişik program örneği vardır. Programlar sıra ile geliştirilmişlerdir. Bu yüzden devrenizi sökmeden her üç programı da deneyebilirsiniz.

Program 1: Tarama Yöntemi ile Display'lere "1234" Yazmak.

Programda 4 display'ede gönderilecek olan veriler display alt programı içerisinde iken tanımlanır.

```
;==========1234.ASM=============================================v2===02/09/2003
;Bu program PIC 16F84'e bağlı 4 adet display'de   tarama yöntemi ile 1234 sayılarını yazar.
;                                                                Hakan KARAKAŞ       Tuzla
;_____________________________________________________________________________________________
        list      p=16f84
        include   "p16f84.inc"
        radix     hex
        __CONFIG          _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
;_____________________________________________________________________________________________
count   equ       0x0c                        ;sayma register'ı

                  org       0x00
main;.....................................................................................
                  call      initial             ; PIC 16F84'ün portlarını ayarla
tekrar            call      display             ; Display'lere yaz
                  goto      tekrar
initial ;.................................................................................
                  bsf       STATUS,RP0          ; Bank 1'e geç
                  movlw     0xF0
                  movwf     TRISB               ; PORTB High ← input PORTB Low → Output
                  movlw     0x00
                  movwf     TRISA               ; PORTA → Output
                  bcf       STATUS,RP0          ; Bank 0'a dön
                  clrf      PORTA               ; PORTA 'da ki  LED'ler OFF
```

```
                return
display;.....................................................................
                movlw   b'11111110'     ; 1.Dijiti seç (LSB)
                movwf   PORTB
                movlw   4               ; 4 rakamını yaz
                movwf   PORTA
                call    timer           ; Bir süre bekle
                movlw   b'11111101'     ; 2.Dijiti seç
                movwf   PORTB
                movlw   3               ; 3 rakamını yaz
                movwf   PORTA
                call    timer           ; Bir süre bekle
                movlw   b'11111011'     ; 3.Dijiti seç
                movwf   PORTB
                movlw   2               ; 2 rakamını yaz
                movwf   PORTA
                call    timer           ; Bir süre bekle
                movlw   b'11110111'     ; 4.Digiti seç (MSB)
                movwf   PORTB
                movlw   1               ; 1 rakamını yaz
                movwf   PORTA
                call    timer           ; Bir süre bekle
                return
timer;.....................................................................
                movlw   0xFF
                movwf   count           ; count ← 0xFF
next            decfsz  count,F         ; count = count - 1, sonuç 0 mı?
                goto    next            ; Hayır.  next'e git
                return                  ; Evet. Timer alt programından  çık
                end
;==========================================================================
```

Program 2: Register Kullanarak Display'e Veri Yazmak

Bir önceki programda "1234" verisinden birkaç sn sonra "5678" verisinin display'lere yazılması istenseydi, display alt programının tekrar yazılması ve 5, 6, 7, 8 rakamlarının ikinci display alt programında tanımlanması gerekecekti. Oysa aşağıdaki programda display'lere yazılacak her veri için ayrı bir alt program yazmaya gerek yoktur. Her dijit için bir kaydedici belirlenmiştir. Ana programda bu kaydedicilerin içeriği değiştirilerek yeni rakamların display'lere kolaylıkla yazılması sağlanmıştır.

```
;==========5678.ASM=============================================v2===02/09/2003
;Bu program PIC 16F84'e bağlı 4 adet display'de                    Hakan KARAKAŞ
;tarama yöntemi display register'larından alarak  5678 sayılarını yazar.      Tuzla
```

```
;_____________________________________________________________________________________________
        list      p=16f84
        include   <p16f84.inc>
        radix     hex
        __CONFIG          _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
;_____________________________________________________________________________________________
count           equ       0x0c            ; sayma register'i
QA              equ       0x0d            ; 1.Dijit register'i
QB              equ       0x0e            ; 2.Dijit register'i
QC              equ       0x0f            ; 3.Dijit register'i
QD              equ       0x10            ; 4.Dijit register'i
                org       0x00
main ;..............................................................................................
                call      initial     ; PIC 16F84'ün portlarını ayarla
                movlw     5
                movwf     QD          ; QD ← 5
                movlw     6
                movwf     QC          ; QC ← 6
                movlw     7
                movwf     QB          ; QB ← 7
                movlw     8
                movwf     QA          ; QA ← 8
tekrar    call  display               ; Display'lere yaz
                goto      tekrar
initial ;...........................................................................................
                bsf       STATUS,RP0        ; Bank 1'e geç
                movlw     0xF0
                movwf     TRISB             ; PORTB High ← Input, PORTB Low → Output
                movlw     0x00
                movwf     TRISA             ; PORTA → Output
                bcf       STATUS,RP0        ; Bank 0'a dön
                clrf      PORTA             ; LED'ler OFF
                return
display;............................................................................................
                movlw     b'11111110'       ; 1.Dijiti seç (LSB)
                movwf     PORTB
                movf      QA,W              ; QA'nın içeriğini yaz
                movwf     PORTA
                call      timer             ; Bir süre bekle
                movlw     b'11111101'       ; 2.Dijiti seç
                movwf     PORTB
                movf      QB,W              ; QB'nin içeriğini yaz
                movwf     PORTA
                call      timer             ; Bir süre bekle
                movlw     b'11111011'       ; 3.Dijiti seç
```

```
                movwf   PORTB
                movf    QC,W            ; QC'nin içeriğini yaz
                movwf   PORTA
                call    timer           ; Bir süre bekle
                movlw   b'11110111'     ; 4.Dijiti seç (MSB)
                movwf   PORTB
                movf    QD,W            ; QD'nin içeriğini  yaz
                movwf   PORTA
                call    timer           ; Bir süre bekle
                return
timer;..........................................................................................
                movlw   0xFF
                movwf   count           ; count ← 0xFF
next            decfsz  count,F         ; count = count - 1, Sonuç 0 mı?
                goto    next            ; Hayır.  next'e git
                return                  ; Evet. Timer alt programından  çık
                end
;==============================================================================
```

Program 3: Array Değişken ile 4 Elemanlı Bir Sayıyı Display'lerde Görüntülemek

Programlar içerisinde en çok kullanılan değişken saklama ve ulaşma yöntemlerinden biri de dizilerdir (array). Bir dizi belli sayıda elemandan oluşur ve her elemanın dizi içerisinde saklandığı belli bir sırası vardır. Bu programda veriler önce bir dizide saklanıp, daha sonra buradan çağrılarak display'lere gönderilmektedir.

4 dijit display olduğundan dizinin eleman sayısı da 4 olsun. Diziye "sayi" ismi verilebilir. Eleman boyutunu tutan register'da "i" isminde olsun. Ekrana gönderilmek istenen sayıda "9876" olsun. Bu durumda

Sayi dizinin boyutu = 4

Sayi dizinin 1. elemanı = "6"

Sayi dizinin 2. elemanı = "7"

Sayi dizinin 3. elemanı = "8"

Sayi dizinin 4. elemanı = "9" olur.

PIC assembly programında FSR register'ını kullanarak sıralı adreslere ulaşılır. Önce normal bir register olarak sayi register'ı tanıtılır.

```
Sayi    equ     0x10
```

Bu register'dan sonra dizi kaç elemanlı ise o kadar adresi ayrılması gerekir. Dizi 4 elemanlı ise yeni bir file register 0x14 adresinde tanımlanmalıdır. Çünkü

Sayi [0] → 0x10 adresinde

Sayi [1] → 0x11 adresinde

Sayi [2] → 0x12 adresinde

Sayi [3] → 0x13 adresindedir.

FSR ile bu adreslere ulaşılırken, adresteki veri INDF register'ından alınır. Aşağıdaki programın **main** kısmında "sayi" dizisinin elemanları tanımlanmış, **display** alt programında ise bu elemanlara ulaşılmıştır. Burada kullanılan dizi mantığı ileriki programlarda daha çok işe yarayacaktır.

```
;==========Array.ASM==================================================23/07/2001
; 4 elemanlı dizinin elemanlarını 4 displayde  array değişken kullanarak  görüntüler.
;                                                          Hakan   Karakaş          TUZLA
;_____________________________________________________________________________________
        title "array.asm"
        list    p=16f84
        include <p16f84.inc>
        radix   hex
        __CONFIG        _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
;_____________________________________________________________________________________
count   equ     0x0c                    ; sayma register'ı
sayi    equ     0x10                    ; sayi dizisi
i       equ     0x14                    ; sayi dizisinin boyutu
                org     0x00
main;.....................................................................................
                call    initial         ; PIC 16F84'ün portlarını ayarla
                movlw   sayi            ; Sayi dizisinin başlangıç adresini
                movlw   6               ;
                movwf   sayi            ; sayi[0] ← 6
                movlw   7
                movwf   sayi+1          ; sayi[1] ← 7
                movlw   8
                movwf   sayi+2          ; sayi[2] ← 8
                movlw   9
                movwf   sayi+3          ; sayi[4] ← 9
tekrar          call    display         ; Display'lere yaz
                goto    tekrar
initial ;.................................................................................
                bsf     STATUS,RP0      ; Bank 1'e geç
                movlw   0xF0
```

```
                movwf   TRISB           ; PORTB High ← input, PORTB Low → Output
                movlw   0x00
                movwf   TRISA           ; PORTA → Output
                bsf     STATUS,RP0      ; Bank 0'a dön
                clrf    PORTA           ; PORTA → LED'ler OFF
                return
display ;.....................................................................................
                movlw   4               ; sayi dizisinin boyutu
                movwf   i               ; i ← 4
                movlw   sayi            ; sayi dizisinin başlangıç adresini
                movwf   FSR             ; FSR'e yaz
                movlw   b'11111110'     ; 1.Dijiti seç (LSB)
                movwf   PORTB           ;
tara            movf    INDF,W          ; dizinin elemanını w'e yaz
                movwf   PORTA           ; PORTA'dan gönder
                call    timer           ; Bir süre bekle
                incf    FSR             ; dizinin bir sonraki elemanın adresine ulaş
                bsf     STATUS,C        ; Carry set
                rlf     PORTB,F         ; aktif olan dijit bir soldaki
                decfsz  i               ; dizinin son elemanına ulaşıldı mı?
                goto    tara            ; Hayır. Taramaya  devam et
                return                  ; Evet. Alt programdan çık
timer;.......................................................................................
                movlw   0xFF
                movwf   count           ; count ← 0xff
next            decfsz  count,F         ; count = count - 1, c=0 mı?
                goto    next            ; Hayır.  next'e git
                return                  ; Evet. Timer alt programından çık
                end
;==============================================================================
```