

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**

DENİZCİLİK

MİKRODENETLEYİCİ 2

ANKARA 2013

-
- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
 - Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
 - **PARA İLE SATILMAZ.**

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. LED KONTROLÜ	3
1.1. LED Devresinin Yapısı ve Bağlantısı	3
1.2. LED Devresinin Kontrol Programı İlk Adım.....	5
1.3. Akış Diyagramı	6
1.4. Programın Komutlar Seviyesinde Açıklamaları	7
1.4.1. İşlemcilerin Özellikleri	7
1.4.2. Portların Kullanımı	8
1.4.3. Portların Kurulumu	9
1.4.4. Ledlerin Yakılması	11
UYGULAMA FAALİYETİ	13
ÖLÇME VE DEĞERLENDİRME	15
ÖĞRENME FAALİYETİ-2	16
2. 7 ELEMANLI DISPLAY KONTROLÜ	16
2.1. 7 segment LED Displayin Kullanımı	16
2.1.1. 7 segment LED Displayin Çalıştırılması	17
2.2. RETLW Komutunun İşlevi	18
2.2.1. RETLW Komutu Örnek Programı.....	18
UYGULAMA FAALİYETİ	21
ÖLÇME VE DEĞERLENDİRME	26
ÖĞRENME FAALİYETİ-3	27
3. SERİ İLETİŞİM	27
3.1. Bilgi İletişim Sistemi	27
3.2. Bilgi İletim (Transmisyon) Sinyal Sistemi	28
3.3. RS232C İletişim Protokolü	28
3.4. Seri Bilgi Aktarım Formatı	29
3.5. Seri İletişim Uygulama Devresi	31
UYGULAMA FAALİYETİ	32
ÖLÇME VE DEĞERLENDİRME	33
MODÜL DEĞERLENDİRME	34
CEVAP ANAHTARLARI	35
KAYNAKÇA	36

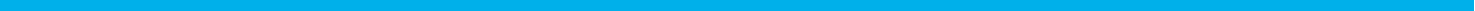
AÇIKLAMALAR

ALAN	Denizcilik
DAL/MESLEK	
MODÜLÜN ADI	Mikrodenetleyici 2
MODÜLÜN TANIMI	Bu modül mikrodenetleyicilerde kullanılan komutların öğretilmesinin yanı sıra, bu komutlarla led, display kontrolünü öğretmek amacıyla yazılmıştır. Bunların dışında mikrodenetleyici ile bilgisayarın nasıl haberleşebileceği öğretilmektedir.
SÜRE	40/32
ÖN KOŞUL	Mikrodenetleyici 1 modülünü almış olmak
YETERLİK	Mikrodenetleyici ile temel giriş çıkış işlemleri yapmak
MODÜLÜN AMACI	Genel Amaç Öğrenci, mikrodenetleyici ile temel giriş çıkış işlemleri yapabilir. Amaçlar: <ol style="list-style-type: none">1. Mikrodenetleyici ile led kontrolünü hatasız olarak yapacaktır.2. Mikrodenetleyici ile display kontrolünü hatasız olarak yapacaktır.3. Mikrodenetleyici ile seri iletişim kontrolünü hatasız olarak yapacaktır.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam: Mikrodenetleyici laboratuvarı Donanım: Bilgisayar, mikrodenetleyici programlama kartı, display elemanları, seri iletişim kablosu, seri iletişim devresi
ÖLÇME VE DEĞERLENDİRME	Her faaliyetin sonunda ölçme soruları ile öğrenme düzeyinizi ölçeceksiniz. Araştırmalarla, grup çalışmaları ve bireysel çalışmalarla öğretmen rehberliğinde ölçme ve değerlendirmeyi gerçekleştirebileceksiniz.

GİRİŞ

Sevgili Öğrenci,

Geleceğin teknolojisi olarak görülen mikrodenetleyiciler, artık yaygın olarak kullanılmaktadır. Yani geleceğimiz mikrodenetleyiciler üzerine inşa edilecek. Bu modülde mikrodenetleyicileri yakından tanıyacak ve programlamayı öğreneceksiniz. Bu modül ders içinde öğreneceğiniz bütün konuların temeli olduğu için lütfen dersleri dikkatli takip edip, çalışmalarınıza özen gösteriniz. Mikrodenetleyicilerin yapısını ve temel komutları öğrendikten sonra yapacağınız uygulamalarla bilgilerinizi pekiştireceksiniz.



ÖĞRENME FAALİYETİ-1

AMAÇ

Mikrodenetleyici ile LED kontrolünü hatasız olarak yapabileceksiniz.

ARAŞTIRMA

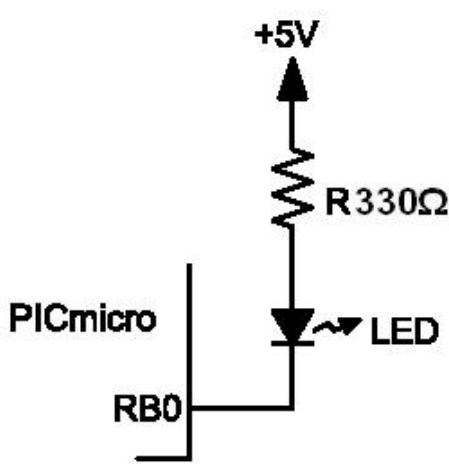
- Binary tabanlı 4 bitlik sayıların desimal değerlerinin nasıl hesaplandığını ve binary sayılarla toplama ve çıkarma işleminin nasıl yapıldığını araştırınız.
- Ayrıca RISC sistemi nedir? Komutları nelerdir? Araştırınız.

1. LED KONTROLÜ

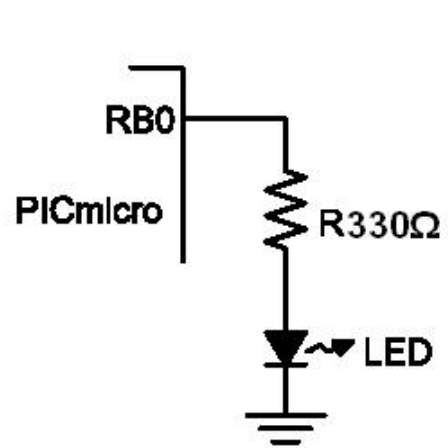
Pic mikrodenetleyicisi ile led kontrolü yapabilmek için bazı alt devrelere ve uygun çıkışları verebilen pic programına ihtiyaç vardır. Şimdi sırası ile bu konuları inceleyelim.

1.1. LED Devresinin Yapısı ve Bağlantısı

Mikrodenetleyicilere LED bağlantısı iki türlü yapılabilir. Akımın kaynaktan PIC'e doğru aktığı bağlantıya SINK akımı, PIC'ten toprağa doğru aktığı bağlantıya ise SOURCE akımı bağlantısı denir.



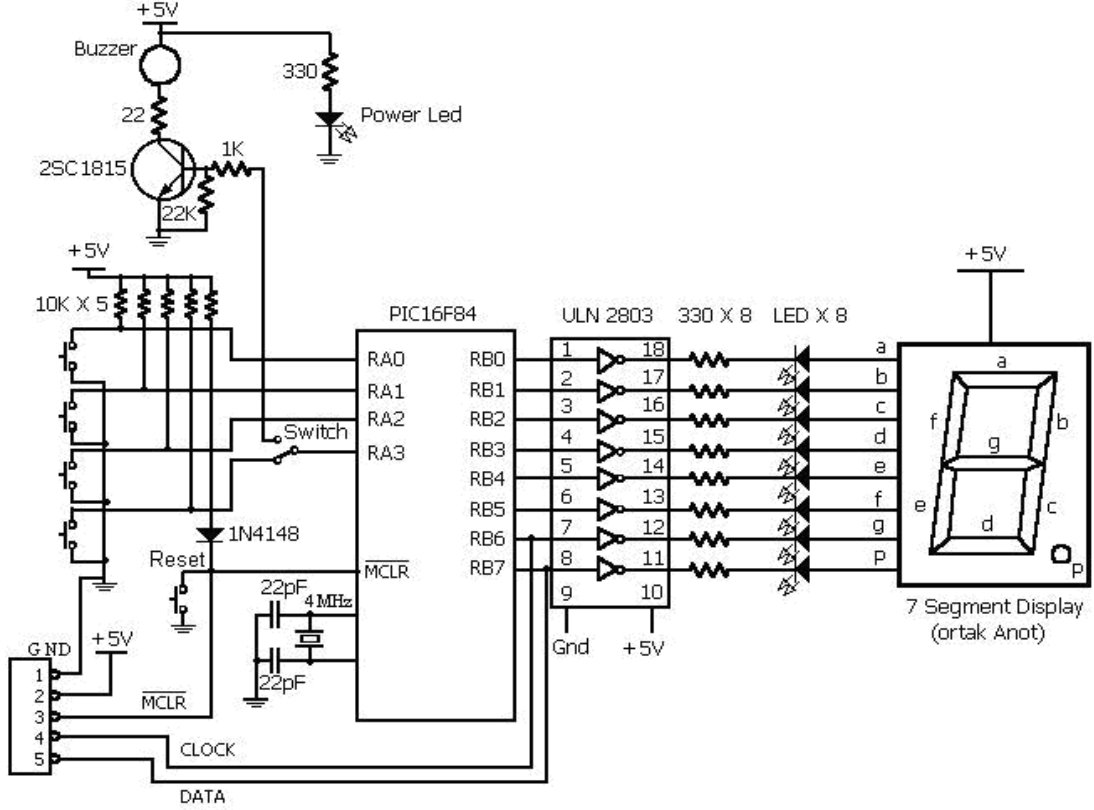
Şekil 1.1.a: Sink akımı



Şekil 1.1.b: Source akımı

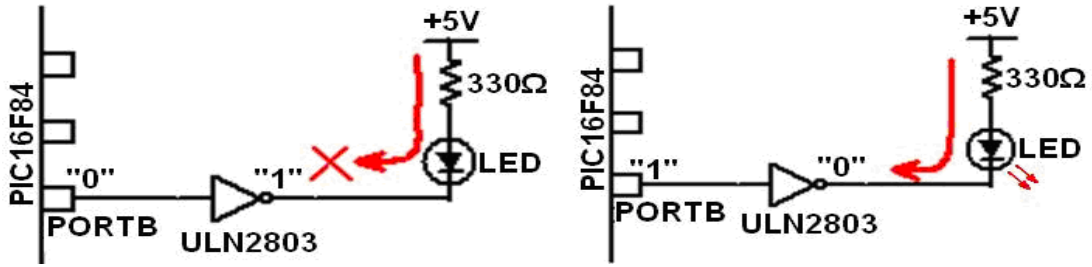
Sink akımı bağlantısında mikrodenetleyicimiz PIC16F84'ten en fazla 25 mA, Source akımı bağlantısında ise 20 mA akım çekebiliriz. Tercih edeceğimiz bu bağlantılara göre yazacağımız programda değişecektir. Sink akımı bağlantısında LED'in ışık verebilmesi için mikrodenetleyicinin PB0 portunun mantıksal "0", source akımı bağlantısında ise mantıksal "1" olması gerekir.

Programlarımızı test edeceğimiz test kartına göre yazmamız gerekir.



Şekil 1.2: Kullanacağımız test kartımızın bağlantı şeması

Test kartımızda LED'ler PIC16F84'ün PORTB pinlerine, 8 elemanlı displayin ledleri ve dirençlerle seri olarak ULN 2803 üzerinden bağlanmıştır. Yukarıda anlattığımız gibi PIC'in portlarından en fazla 25 mA akım çekebilmekteyiz. Bizim bu bağlantımıza göre çektiğimiz akım çok daha fazla olacaktır. Bunun için 500 mA'e kadar akım kapasitesi olan ULN 2803 kullanılmaktadır. Burada unutulmaması gereken şey ULN 2803'ün aynı zamanda bir DEĞİL kapısı olduğudur. Daha basit gösterimle;



Şekil 1.3: Test kartımızdaki LED'lerin bağlantısının basit gösterimi

Eğer programımızda PORTB uçlarını mantıksal "0" yaparsak DEĞİL kapısı çıkışı "1" değerine sahip olacağından LED'ler ışık vermeyecektir. Dolayısıyla PORTB'yi mantıksal "1" yaptığımızda DEĞİL kapısı çıkışı "0" değerinde olacağından akım geçecek ve LED'ler ışık verecektir.

1.2. LED Devresinin Kontrol Programı İlk Adım

PORTB'nin 0'ıncı bitine bağlı olan LED'i yakmak istiyoruz. Bunun için gerekli olan programı yazalım.

```

INCLUDE      "P16F84.INC" ;Adresleri belirten dosyayı yükle      (1)

LIST        P=16F84      ;16F84'ün tanıtımını yap              (2)

BSF        STATUS,5      ;Bank 1'e geç                          (3)

CLRF       TIRSB         ;PORTB'nin hepsini çıkış yap (4)

BCF        STATUS,5      ;Bank 0'a geç                          (5)

CLRF       PORTB         ;PORTB'nin hepsini 0 yap              (6)

BSF        PORTB,0       ;PORTB'nin 0.bitini 1 yap (LED yak) (7)

DEVAM     GOTO          DEVAM      ;Program sonsuz döngüye girer.(8)

END                                               ;Programı sonlandır.      (9)

```

1.3. Akış Diyagramı

```
INCLUDE"P16F84.INC"
```

```
LIST P=16F84
```

```
BSF STATUS,5
```

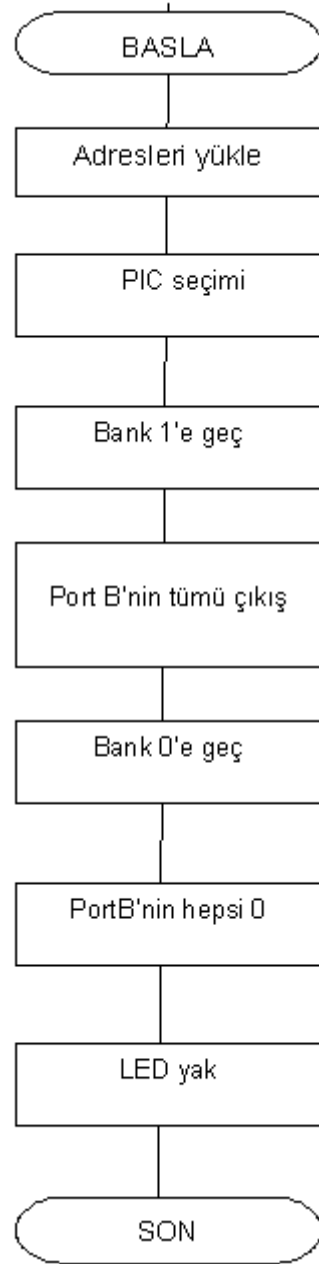
```
CLRF TRISB
```

```
BCF STATUS,5
```

```
CLRF PORTB
```

```
BSF PORTB,0
```

```
GOTO DEVAM
```



Şekil 1.4: Akış diyagramı

1.4. Programın Komutlar Seviyesinde Açıklamaları

1.4.1. İşlemcilerin Özellikleri

LIST P=16F84 (1)

LIST komutu ile işlemcinin özellikleri belirtilir.

➤ Format

List [<list_opsiyon>,...,<list_opsiyon>]

Örnek:

LIST P=16F84, F=INHX8M, R=DEC

➤ Açıklama

Seçenek	Varsayım	Tanım
b=nnn	8	TAB boşluk sayısının tanımı
c=nnn	132	Bir satırdaki harf sayısının tanımı
f=<format>	INHX8M	HEX dosya formatındaki çıktının tanımı
Free	FIXED	Serbest form
Fixed	FIXED	Sabit form
mm=ON OFF	ON	Memory haritasının listeye yazılması ON OFF
n=nnn	60	Bir sayfadaki satır sayısının tanımı
p=<type>	None	İşlemci tanımı (Example: P=,16F84)
r=<radix>	HEX	Numerik değer tipi <HEX,DEC,OCT>
St=ON OFF	ON	Sembol tablosunun listeye yazılması ON OFF
t=ON OFF	OFF	Satır değiştirme < satır taşması sonucu > ON OFF
W=0 1 2	0	Assemblerin mesaj seviye tanımı
x=ON OFF	ON	Makro geliştirilmiştir. ON OFF

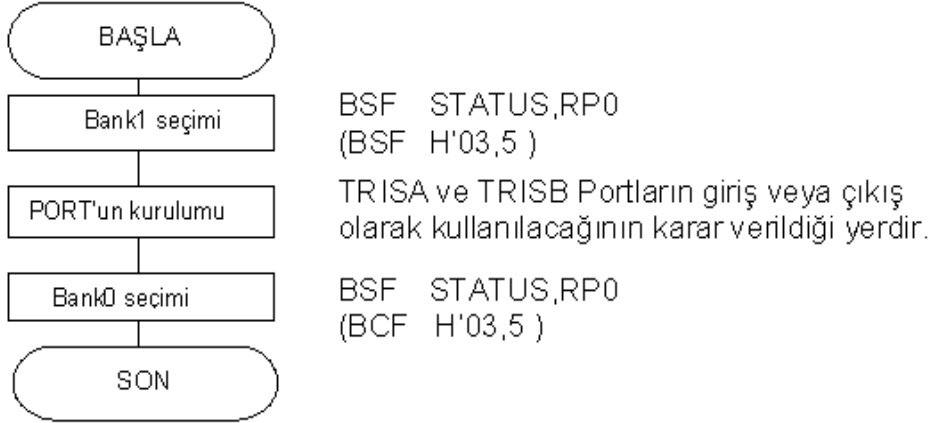
'nnn' desimal numaralarla tanımlama eklenmesidir.

Tablo 1.1: Opsiyonların açıklanması

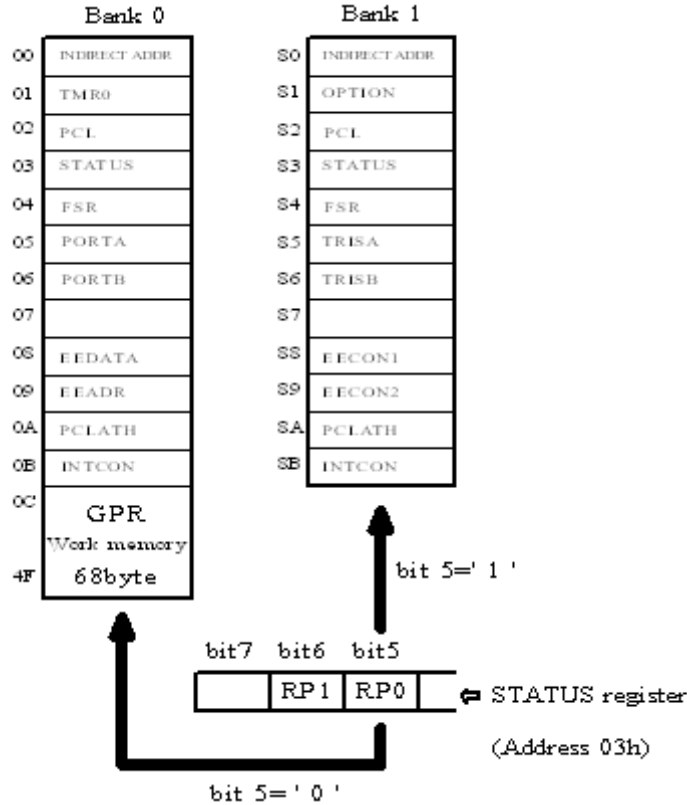
1.4.2. Portların Kullanımı

Bank Seçimi

PIC'i kullanmaya başlamadan önce portların kurulması (I/O belirlenmesi) gerekmektedir.



Şekil 1.5: I/O portun kurulumu



Şekil 1.6: Bank seçimi

PIC'in portlarının giriş/çıkış (I/O) belirlendiği yazmaçlar (TRISA(85h) ve TRISB(86h) olarak adlandırılmıştır.

TRIS registerleri Bank1'dedir. Bank1'e geçiş (03h) adresindeki STATUS yardımı ile olur.

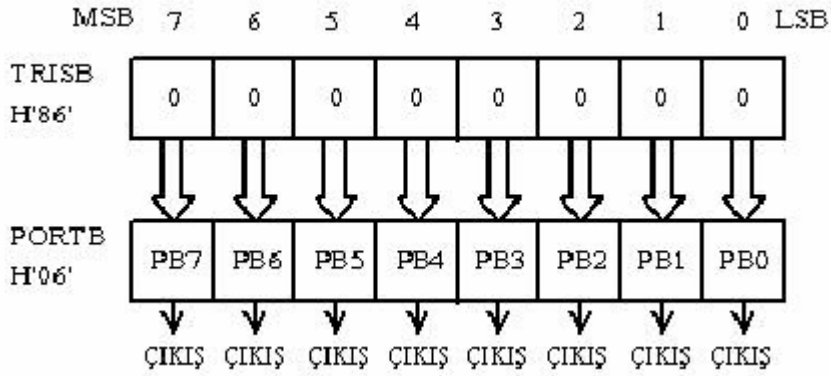
1.4.3. Portların Kurulumu

Örnek 1:

PORTB'nin hepsi ÇIKIŞ

(form 1) CLRF TRISB
(CLRF H'86')

(form 2) MOVLW H'00' ; W register << 00
 MOVWFTRISB ; TRISB << W register
(MOVWF H'86')



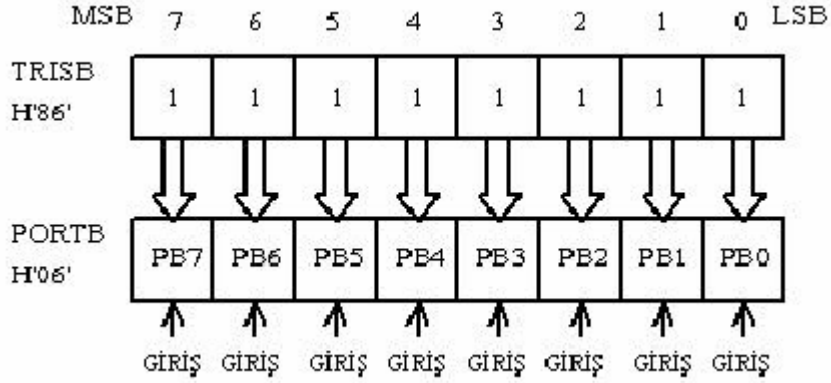
Şekil 1.7: PORTB'nin hepsi ÇIKIŞ

Örnek 2:

PORTB'nin hepsi GİRİŞ

(form 1) MOVLW H'FF' ; W register << FF
 MOVWF TRISB ; TRISB << W register
(MOVWF H'86')

(form 2) MOVLW B'11111111' ; W register << FF
 MOVWF TRISB ; TRISB << W register
(MOVWF H'86')



Şekil 1.8: PORTB'nin hepsi GİRİŞ

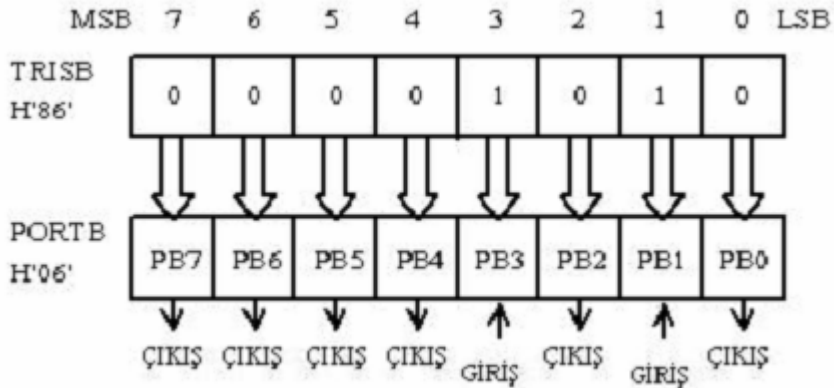
Örnek 3:

PORTB bit 0,bit 2,bit 4,bit 5,bit 6,bit 7 >> ÇIKIŞ

bit 1,bit3 >> GİRİŞ

(form 1) MOVLW H'0A' ; W register << 0Ah
 MOVWF TRISB ; TRISB << W register
 (MOVWF H'86')

(form 2) MOVLW B'00001010' ; W register << FF
 MOVWF TRISB ; TRISB << W register



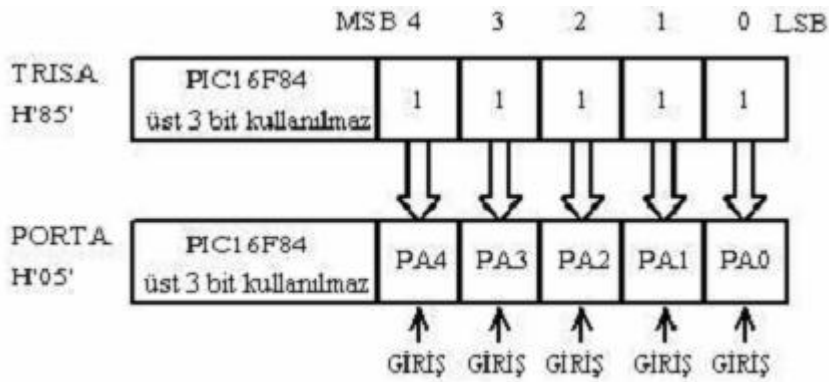
Şekil 1.9: Beraber kullanım I/O giriş çıkışlar

Örnek 4:

PORTA'nın hepsi GİRİŞ

```
(form 1)  MOVLW    H'1F'      ; W register << 1Fh
          MOVWF   TRISA     ; TRISA << W register
          (MOVWF  H'86' )

          MOVLW   B'11111'  ; W register << 1F
          MOVWF   TRISA     ; TRISA << W register
```



Şekil 1.10: PORTA'nın hepsi GİRİŞ

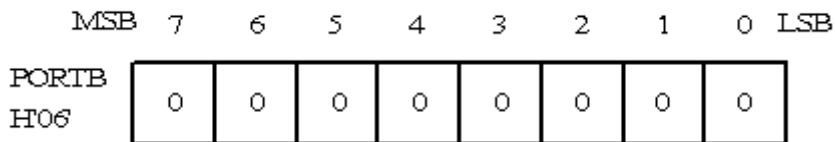
1.4.4. Ledlerin Yakılması

```
CLRF PORTB      ;PORTB'nin hepsi 0 >> Bütün LED'ler sönmük
BSF  PORTB,0    ;PORTB'nin 0. bitindeki LED yanar
(BSF  H'06',0)
```

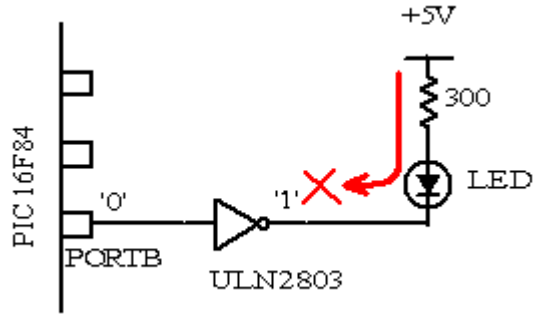
```
STOP GOTO STOP ;program sonsuz döngüye girer.
END
```

1) CLRF H'06'

PORTB'nin adresi 06h dir. CLRF ile bu adresteki bilgileri 0 hale getiririz. (Temizleriz)

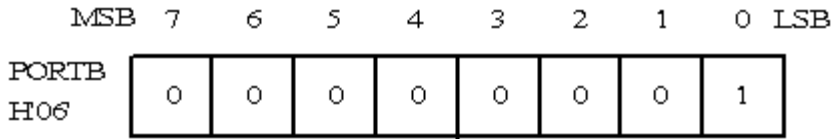


Şekil 1.11: PORTB'nin temizlenmesi

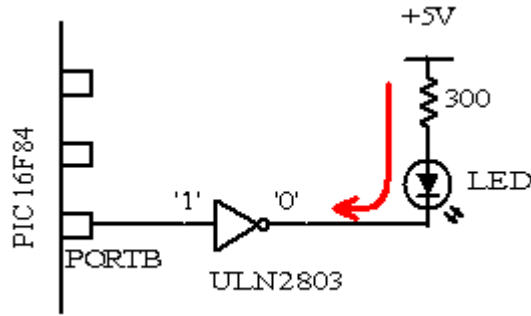


Şekil 1.12: PORTB >> 0 (LED sönmük)

2) BSFPORTB,0 (BSF H'06',0)



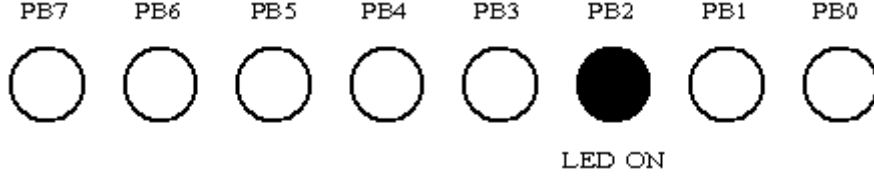
Şekil 1.13: 0. bitin set (1) hâle getirilmesi



Şekil 1.14: PORTB >> 1 (LED yanık)

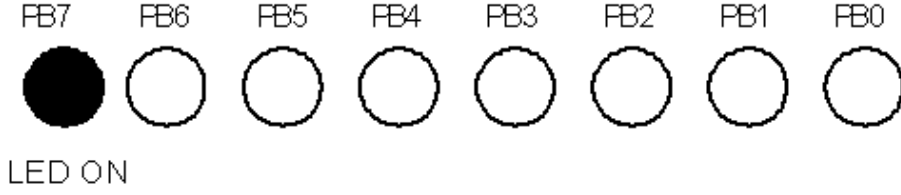
UYGULAMA FAALİYETİ

Aşağıdaki şekle göre PB2'deki LED'in yanmasını sağlayınız.



İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Yazacağınız programın akış diyagramını çiziniz.	<ul style="list-style-type: none">➤ Akış diyagramı hazırlamadan yazılan programlarda hata yapma oranı daha fazladır. Bunun için (özelikle ilk öğrenme aşamasında olduğumuz için) mutlaka akış diyagramı çiziniz.
<ul style="list-style-type: none">➤ MPLAB IDE editöründe program kodlarınızı yazıp derleyiniz.	<ul style="list-style-type: none">➤ Programı yazmaya başlamadan önce MPLAB IDE programının “configure” menüsünde “select device” seçeneği ile gelen pencerede PIC16F84'ün seçili olup olmadığını kontrol ediniz.
<ul style="list-style-type: none">➤ Hata oluşursa hatanın nedenini araştırınız.	<ul style="list-style-type: none">➤ Programı derledikten sonra hata oluşursa ekranda görüntülenen ERR uzantılı dosya hatayı programın hangi satırında olduğunu gösterir. Buradan hatanın nerede olduğunu anlayabilirsiniz.
<ul style="list-style-type: none">➤ Programlama kartınızı (LPT portuna) bilgisayarınıza bağlayınız.	<ul style="list-style-type: none">➤ Bağlantıları yaparken DC adaptörünüzün bağlı olmamasına dikkat ediniz.
<ul style="list-style-type: none">➤ DC güç anahtarı ile devreye gerilim uygulayınız.	<ul style="list-style-type: none">➤ Her HEX dosyası yüklediğinizde konfigürasyon ayarları yapmak zorundasınız. Özellikle kristal ayarı (şu an için XT)ve WDT (disable olmalı) yapmalısınız.
<ul style="list-style-type: none">➤ Programlayıcıya yaptığınız programın derleme sonucu oluşan HEX dosyasını yükleyiniz.	<ul style="list-style-type: none">➤ Programlayıcının alt kısmında “Programing FAILD” mesajı belirdi ise deney kartınızın bilgisayar bağlantısı veya kart üzerinde temassızlık olabilir, kontrol ediniz.
<ul style="list-style-type: none">➤ Kristal ve WDT ile ilgili ayarları yapıp PIC'e programı yükleyiniz.	
<ul style="list-style-type: none">➤ Programın çalışmasının doğru olup olmadığını kontrol ediniz. Eğer çalışmada bir hata var ise gidererek yükleme işini tekrarlayınız.	
<ul style="list-style-type: none">➤ Sonuç raporunu hazırlayınız.	

Aşağıdaki şekle göre PB7'deki LED'in yanmasını sağlayınız.



İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Yazacağınız programın akış diyagramını çiziniz.	<ul style="list-style-type: none">➤ Akış diyagramı hazırlamadan yazılan programlarda hata yapma oranı daha fazladır. Bunun için (özelikle ilk öğrenme aşamasında olduğumuz için) mutlaka akış diyagramı çiziniz.
<ul style="list-style-type: none">➤ MPLAB IDE editöründe program kodlarınızı yazıp derleyiniz.	<ul style="list-style-type: none">➤ Programı yazmaya başlamadan önce MPLAB IDE programının “configure” menüsünde “select device” seçeneği ile gelen pencerede PIC16F84’ün seçili olup olmadığını kontrol ediniz.
<ul style="list-style-type: none">➤ Hata oluşursa hatanın nedenini araştırınız.	<ul style="list-style-type: none">➤ Programı derledikten sonra hata oluşursa ekranda görüntülenen ERR uzantılı dosya hatayı programın hangi satırında olduğunu gösterir. Buradan hatanın nerede olduğunu anlayabilirsiniz.
<ul style="list-style-type: none">➤ Programlama kartınızı (LPT portuna) bilgisayarınıza bağlayınız.	<ul style="list-style-type: none">➤ Bağlantıları yaparken DC adaptörünüzün bağlı olmamasına dikkat ediniz.
<ul style="list-style-type: none">➤ DC güç anahtarı ile devreye gerilim uygulayınız.	<ul style="list-style-type: none">➤ Her HEX dosyası yüklediğinizde konfigürasyon ayarları yapmak zorundasınız. Özellikle kristal ayarı (şu an için XT)ve WDT (disable olmalı) yapmalısınız.
<ul style="list-style-type: none">➤ Programlayıcıya yaptığınız programın derleme sonucu oluşan HEX dosyasını yükleyiniz.	<ul style="list-style-type: none">➤ Programlayıcının alt kısmında “Programing FAILD” mesajı belirdi ise deney kartınızın bilgisayar bağlantısı veya kart üzerinde temassızlık olabilir, kontrol ediniz.
<ul style="list-style-type: none">➤ Kristal ve WDT ile ilgili ayarları yapıp PIC’e programı yükleyiniz.	
<ul style="list-style-type: none">➤ Programın çalışmasının doğru olup olmadığını kontrol ediniz. Eğer çalışmada bir hata var ise gidererek yükleme işini tekrarlayınız.	
<ul style="list-style-type: none">➤ Sonuç raporunu hazırlayınız.	

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. (...) Sink akımı gerilim kaynağından PIC'e doğru akan akım devresidir ve 20 mA akım sağlar.
2. (...) ULN2803 entegresi girişine uygulanan sinyali kuvvetlendirirken aynı zamanda tersler.
3. (...) TRISA ve TRISB, portların giriş veya çıkış olarak kullanılmasını sağlar.
4. (...) IC üzerindeki portlardan PORTA giriş, PORTB ise çıkış için kullanılır.
5. (...) LIST komutunun opsiyonlarından R, program içinde kullanılan sayılar için taban belirlemek için kullanılır.
6. (...) MOVWF H'85',PORTB komutu, 16'lık tabandaki 85 sayısını PORTB'ye taşır.
7. (...) 20 Mhz'lik PIC ile 4 MHz'lik osilatör kullanırsak komut işleme hızı 20 MHz olur.
8. (...) 20 Mhz'lik PIC ile 20 MHz'lik osilatör kullanırsak bir komutun işlenme hızı 0,2 us olur.
9. (...) CALL komutu Program Counter'daki adresi Stack üzerine bir artırarak yazar.
10. (...) RRF PORTB,2 komutu portb üzerindeki bilgiyi sağa doğru iki dijite kaydırır.
11. (...) Buton uygulamalarında mekanik sebeplerle meydana gelen gürültüye chattering denir.
12. (...) Programla chattering önleme, buton üzerindeki sinyalin, zaman aralığı ile iki defa kontrol edilmesidir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Mikrodenetleyici ile display kontrolünü hatasız olarak yapabileceksiniz.

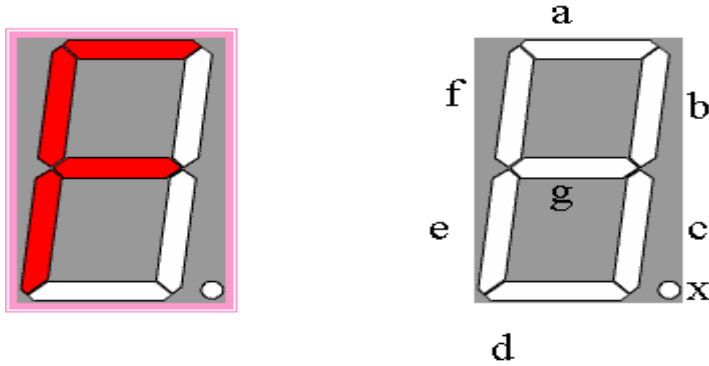
ARAŞTIRMA

- Display çeşitleri ve yapıları hakkında araştırma yapınız.

2. 7 ELEMANLI DISPLAY KONTROLÜ

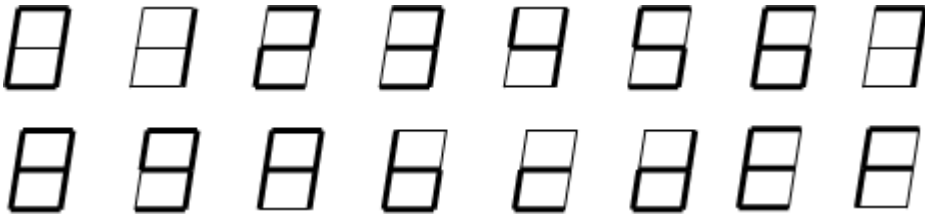
Bu birleşik LED grubu 7 veya 8 adet LED'den (noktanın bulunup bulunmadığına göre) meydana gelmiştir. LEDlerin Anotları birleştirildi ise **ortak anot**, katotları birleştirildi ise **ortak katot** olarak isim alırlar.

2.1. 7 segment LED Displayin Kullanımı



7 segment bilgi	x	g	f	e	d	c	b	a
PORTB	0	1	1	1	0	0	0	1

Şekil 2.1: 7 segment LED display'in kullanımı



Şekil 2.2: Yedi segment display bilgileri

2.1.1. 7 segment LED Displayin Çalıştırılması

Test bordumuzda LED'ler ise display segmentleri seri olarak bağlıdır. PIC uçlarına bağlı olan 7 segment display üzerinde karakterlerin görüntülenebilmesi için aşağıdaki tabloyu doldurunuz.

7 segment PORTB	x PB7	g PB6	f PB5	e PB4	d PB3	c PB2	b PB1	a PB0	HEX data
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
A									
b									
C									
d									
E									
F	0	1	1	1	0	0	0	1	71h

2.2. RETLW Komutunun İşlevi

RETLW komutu RETURN komutu gibi alt programlarda, ana programa dönmek için kullanılır. Fakat RETLW komutu yanında bir sayı ile kullanılır. Alt programda ana programa geçişte bu sayıyı W registerine yazar.

2.2.1. RETLW Komutu Örnek Programı

Bu program 7 segmentli LED'in ardışık olarak 0'dan 9'a kadar olan sayıları göstermesini sağlar.

- W registerine 0'dan 9'a kadar olan değerler atanır ve program bunu altprogram tablosundan çağırır (CALL SEGDAT).
- Örneğin, W register "1" değerine sahipken CALL SEGDAT komutu ile alt program çalıştırılır ve program sayıcı 40h olur.
- Sonra ADDWF PCL,F komutunu çalıştırdığımızda program sayıcı 41h(PCL(41h) + W register (01h)) olur. Böylece program sayıcısına +1 eklenir. Sonra program sayıcısı 42h. olarak belirlenir.
- Program 42h'i uygular. RETLW B'00000110' (7 segment datası = 1).
- Alt programdan sonra gerçek programa geri döner W registeri B'00000110' olur.
- 7 segment LED'de '1' görünür.

```
CALL      SEGDAT
```

```
PC (program sayacı)
```

```
40h SEGDAT      ADDWF      PCL,F      ;PCL( program counter ) + W
    >>PCL

41h          RETLW      B'00111111'  ;segmen data 0 >> W (PCL+0)
42h          RETLW      B'00000110'  ;segmen data 1 >> W (PCL+1)
43h          RETLW      B'01011011'  ;segmen data 2 >> W (PCL+2)
44h          RETLW      B'01001111'  ;segmen data 3 >> W (PCL+3)
45h          RETLW      B'01100110'  ;segmen data 4 >> W (PCL+4)
46h          RETLW      B'01101101'  ;segmen data 5 >> W (PCL+5)
47h          RETLW      B'01111101'  ;segmen data 6 >> W (PCL+6)
48h          RETLW      B'00100111'  ;segmen data 7 >> W (PCL+7)
49h          RETLW      B'01111111'  ;segmen data 8 >> W (PCL+8)
4Ah          RETLW      B'01101111'  ;segmen data 9 >> W (PCL+9)
```

```
*****
;* RETLW komutunun kullanılması
;* 0'dan 9'a kadar 7 segment LED kullanımı
*****
```

```
LIST P=16F84,R=DEC
INCLUDE "P16F84.INC"
```

```
COUNT1 EQU      H'10'      ; MEMORY ADRESİ
```

```

COUNT2 EQU      H'11'      ;
COUNT3 EQU      H'12'      ;
RAM1 EQU         H'13'      ;

INIT   BSF        STATUS,RP0 ; Bank 1'e geç
       CLRF       TRISB      ; PORTB'nin tamamı çıkış
       BCF        STATUS,RP0 ; Bank 0'a geç
       CLRF       PORTB      ; PORTB'yi temizle

LOOP   MOVLW      D'10'      ; segmen data numarası = 10
       MOVWF      COUNT3     ; count3 = 10
       CLRF       RAM1       ; RAM1 = 0

LOOP1  MOVF       RAM1,W     ;RAM1 >> W
       CALL       SEGDAT     ; RAM1 data 0'dan 9'a kadar
       MOVWF      PORTB     ; W(segment data) >> PORTB
       CALL       TIMER      ; 0.3s gecikmeyi çağır
       INCF       RAM1,F     ; RAM1+1 >> RAM1
       DECFSZ     COUNT3,F   ; COUNT3-1 >> COUNT3 = 0
       GOTO       LOOP1

       GOTO       LOOP

STP    GOTO       STP

SEGDAT ADDWF      PCL,F      ;PCL(program counter) + W >>PCL
       RETLW      B'00111111' ;segmen data 0 >> W (PCL+0)
       RETLW      B'00000110' ;segmen data 1 >> W (PCL+1)
       RETLW      B'01011011' ;segmen data 2 >> W (PCL+2)
       RETLW      B'01001111' ;segmen data 3 >> W (PCL+3)
       RETLW      B'01100110' ;segmen data 4 >> W (PCL+4)
       RETLW      B'01101101' ;segmen data 5 >> W (PCL+5)
       RETLW      B'01111101' ;segmen data 6 >> W (PCL+6)
       RETLW      B'00100111' ;segmen data 7 >> W (PCL+7)
       RETLW      B'01111111' ;segmen data 8 >> W (PCL+8)
       RETLW      B'01101111' ;segmen data 9 >> W (PCL+9)

TIMER                                     ;yaklaşık 0.3s zamanlayıcı

;*****
;* 1+1+150*(1+1+200*(2+2+2+1+1+2)+1+1+2)+1+2=300905us      *
;*****

       MOVLW      150        ; 1clock
       MOVWF      COUNT1     ; 1clock
DLY1   MOVLW      200        ; 1clock
       MOVWF      COUNT2     ; 1clock
DLY2   GOTO       $+1        ; 2clock
       GOTO       $+1        ; 2clock
       GOTO       $+1        ; 2clock
       NOP          ; 1clock
       DECFSZ     COUNT2,1   ; 1(2)clock

```

```
GOTO      DLY2      ; 2clock
DECFSZ   COUNT1,1  ; 1(2)clock
GOTO     DLY1      ; 2clock
RETURN   ; 2clock
END
```

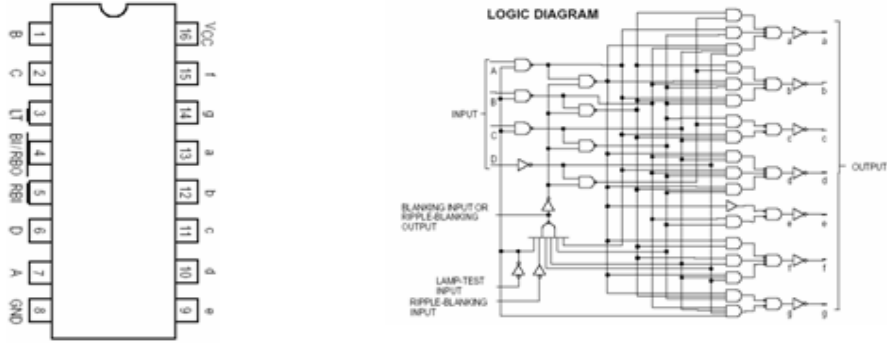

UYGULAMA FAALİYETİ

0'dan F'e kadar ardışık sayan ve bunu 7 segment LED displayde gösteren program yapınız ve uygulayınız.

İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Yazacağınız programın akış diyagramını çiziniz.	<ul style="list-style-type: none">➤ Akış diyagramı hazırlamadan yazılan programlarda hata yapma oranı daha fazladır. Bunun için (özellikle ilk öğrenme aşmasında olduğumuz için) mutlaka akış diyagramı çiziniz.
<ul style="list-style-type: none">➤ MPLAB IDE editöründe program kodlarınızı yazıp derleyiniz.	<ul style="list-style-type: none">➤ Programı yazmaya başlamadan önce MPLAB IDE programının “configure” menüsünde “select device” seçeneği ile gelen pencerede PIC16F84'ün seçili olup olmadığını kontrol ediniz.
<ul style="list-style-type: none">➤ Hata oluşursa hatanın nedenini araştırınız.	<ul style="list-style-type: none">➤ Programı derledikten sonra hata oluşursa ekranda görüntülenen ERR uzantılı dosya hatayı programın hangi satırında olduğunu gösterir. Buradan hatanın nerede olduğunu anlayabilirsiniz.
<ul style="list-style-type: none">➤ Programlama kartınızı (LPT portuna) bilgisayarınıza bağlayınız.	<ul style="list-style-type: none">➤ Bağlantıları yaparken DC adaptörünüzün bağlı olmamasına dikkat ediniz.
<ul style="list-style-type: none">➤ DC güç anahtarı ile devreye gerilim uygulayınız.	<ul style="list-style-type: none">➤ Her HEX dosyası yüklediğinizde konfigürasyon ayarları yapmak zorundasınız. Özellikle kristal ayarı (şu an için XT)ve WDT (disable olmalı) yapmalısınız.
<ul style="list-style-type: none">➤ Programlayıcıya yaptığınız programın derleme sonucu oluşan HEX dosyasını yükleyiniz.	<ul style="list-style-type: none">➤ Programlayıcının alt kısmında “Programing FAILD” mesajı belirdi ise deney kartınızın bilgisayar bağlantısı veya kart üzerinde temassızlık olabilir, kontrol ediniz.
<ul style="list-style-type: none">➤ Kristal ve WDT ile ilgili ayarları yapıp PIC'e programı yükleyiniz.	
<ul style="list-style-type: none">➤ Programın çalışmasının doğru olup olmadığını kontrol ediniz. Eğer çalışmada bir hata var ise gidererek yükleme işini tekrarlayınız.	
<ul style="list-style-type: none">➤ Sonuç raporunu hazırlayınız.	

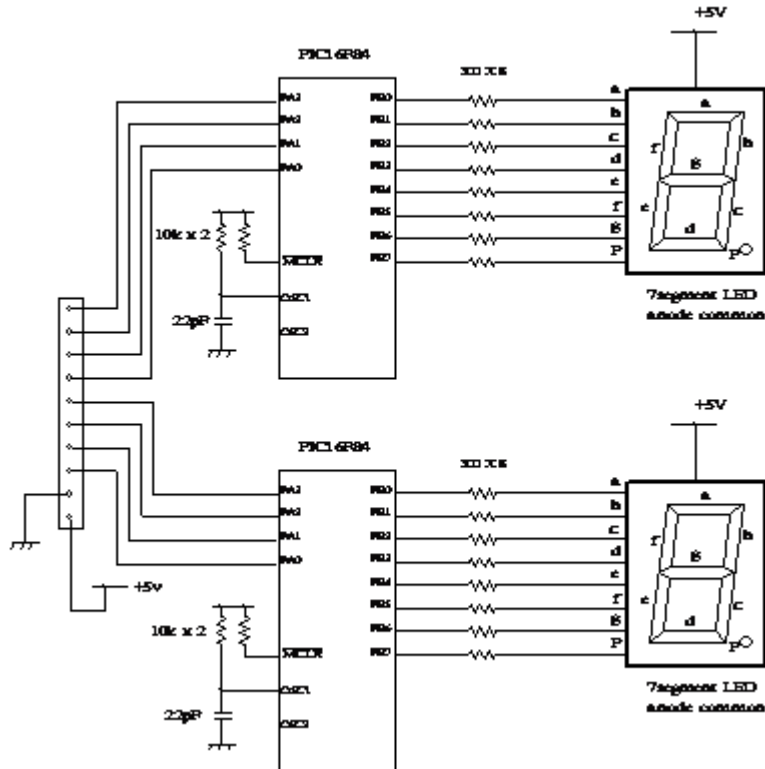
YEDİ ELEMANLI DİSPLAY KOD ÇÖZÜCÜ

PIC'i kullanarak yedi LED segmentli kod çözücü yapalım. Günümüzde bunun hazır entegreleri vardır ve bunlar 7447 ve 9368'dir. Şekilde 7447'nin iç yapısı görülmektedir. Şekilde görüldüğü gibi bu kod çözücü tamamen lojik devrelerden meydana gelmiştir.



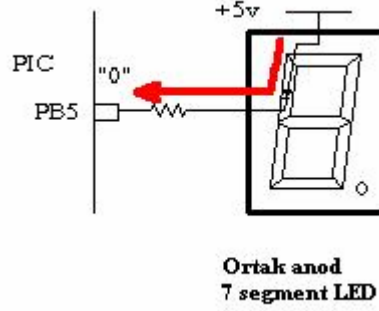
Şekil 2.3: 7447'nin paket ve iç yapısı

Biz burada bu yedi segmentli kod çözücü programını yazacağız. Bunun için uygulama devresine ihtiyacımız var. Bu devrenin şekli aşağıda verilmiştir.



Şekil 2.4: Yedi segmentli kod çözücü devre şeması

Bu devrede test kartımızdan farklı olarak yedi segment LED direkt olarak PORTB'ye bağlanmıştır. Eğer PIC PORTB 'ye "0" gönderirsek display LED'i sink akımından dolayı yanacaktır.



Şekil 2.5: Kod çözücü devresinde 7 segment display bağlantısı

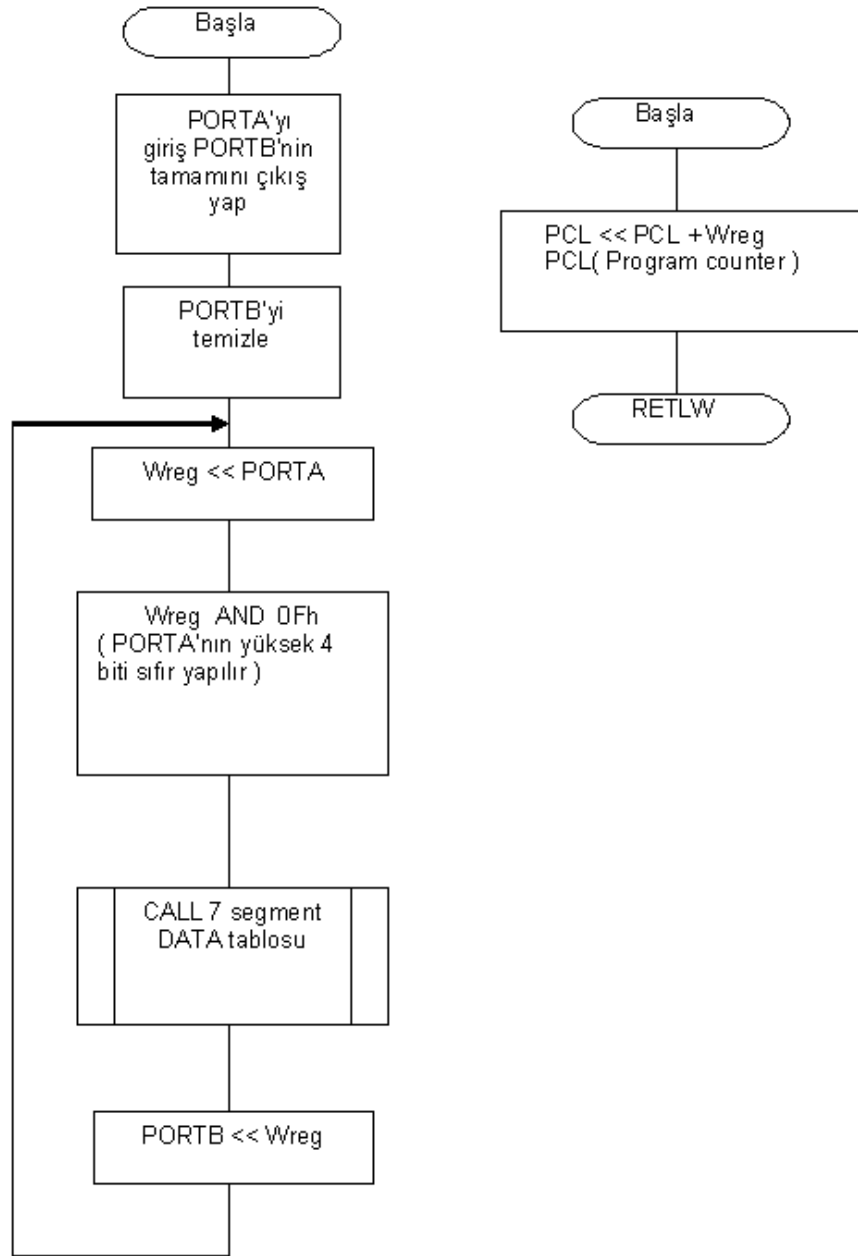
PROGRAMIN AÇIKLANMASI

Bir önceki programda 0'dan 9'a kadar veya 0'dan F'e kadar olan bilgileri ardışık olarak display üzerinde görüntülenmesini öğrendik. Burada yapmak istediğimiz şey ise PORTA'ya bağlı olan butonlardan girilen binary bilgiye göre display üzerinden çıkış almaktır. Yani butonlarla girilen ikili sayı sistemi bilgisi, display üzerinde desimal olarak görüntülenmiş olacak. Bu yaptığımız işleme kod çözme işlemi denir.

Bunun için aşağıdaki tabloyu tekrar doldurunuz.

PA3	PA2	PA1	PA0	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	Segment
				P	g	f	e	d	c	b	a	
0	0	0	0									0
0	0	0	1									1
0	0	1	0									2
0	0	1	1									3
0	1	0	0									4
0	1	0	1									5
0	1	1	0									6
0	1	1	1									7
1	0	0	0									8
1	0	0	1									9
1	0	1	0									A
1	0	1	1									B
1	1	0	0									C
1	1	0	1									D
1	1	1	0									E
1	1	1	1									F

YEDİ ELEMANLI DİSPLAY PROGRAMININ AKIŞ DİYAGRAMI



Şekil 2.6: Programın denemesi

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise **D**, yanlış ise **Y** yazınız.

1. () Birleşik LED grubu 7 veya 8 adet LED'den (noktanın bulunup bulunmadığına göre) meydana gelmiştir.
2. () LED'lerin anotları birleştirilirse **ortak anot**, katotları birleştirilirse **ortak katot** olarak isim alırlar.
3. () RETLW komutu TURN komutu gibi alt programlarda, ana programa dönmek için kullanılır.
4. () RETLW komutu yanında bir sayı ile kullanılır.
5. () Alt programdan ana programa geçişte RETLW komutu, yanındaki sayıyı W registerine yazar.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-3

AMAÇ

Mikrodenetleyici ile seri iletişim kontrolünü hatasız olarak yapabileceksiniz.

ARAŞTIRMA

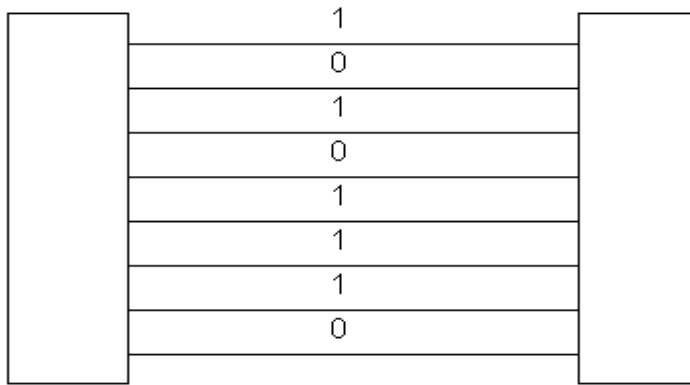
- Seri ve paralel iletişim arasındaki farklar nelerdir? Araştırınız.

3. SERİ İLETİŞİM

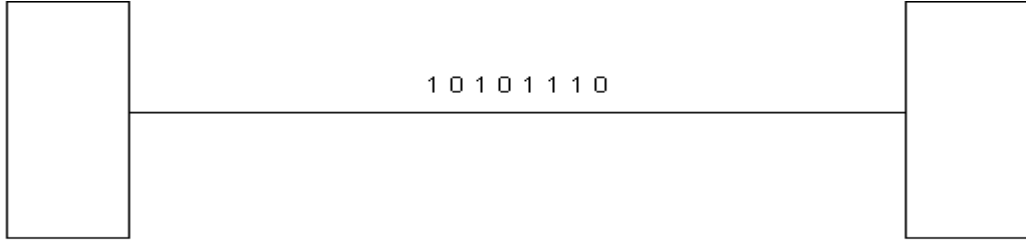
Seri iletişimi Mikrodenetleyicilerin birbiri ile veya bilgisayar ile haberleşmelerinde oldukça kolaylıklar sağlar şimdi seri haberleşme standartları hakkında incelemeler yapalım.

3.1. Bilgi İletişim Sistemi

Bilgi iletiminde düzenli iki sistem mevcuttur. Birisi paralel sistem diğeri ise seri sistemdir. Paralel sistem bilgi iletim hattının tüm bitlerini kullanırken seri sistem ise iletişimde bilgi iletim hatlarının sadece birini kullanır. Transmitter (gönderici), paralelden seriye geçişe dataların her bir bite parçalanması için gereklidir. (Paralel-Seri dönüşüm). Receiver (alıcı) ise bilgileri birleştirmek için gereklidir (Seri-Paralel dönüşüm).



Şekil 3.1: Paralel iletişim sistemi



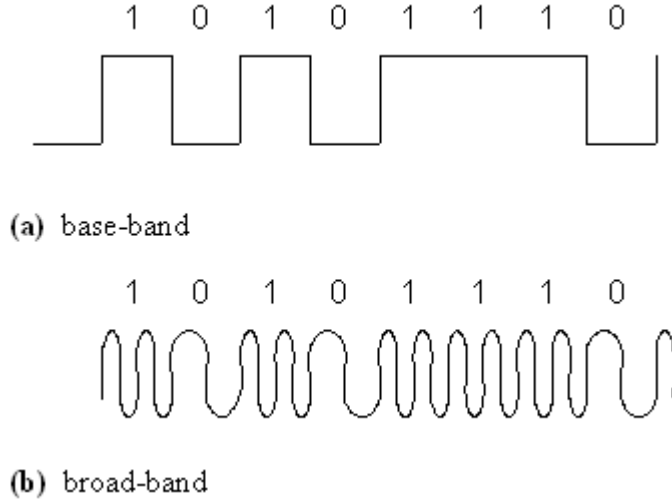
Şekil 3.2: Seri iletişim sistemi

3.2. Bilgi İletim (Transmisyon) Sinyal Sistemi

Burada dijital sinyallerin nasıl sinyal hattına aktarıldığını açıklayacağız.

Alçak band (base-band) transmission 1 ve 0 sinyal için atama yapılmış gerilim sistemidir. Bu sistemde sadece bir band kullanımına izin verilmektedir. Sadece bir tek cihaz bilgi gönderebilir.

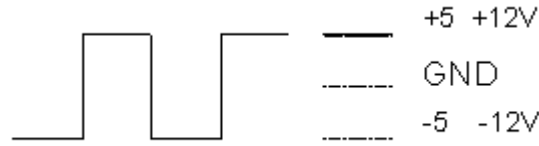
Geniş band (broad-band) transmission ise yüksek frekans 0 ve 1 için düşük frekansa atanmış sistemdir. Base-band'ın tam tersidir. Sanal olarak birçok kanala bölünmüştür. Her kanalın kendine ait taşıyıcı frekansı vardır. Bu farklı frekanslar sayesinde aynı anda birçok farklı bilgi birbirine karışmadan iletilebilir.



Şekil 3.3: Bilgi aktarımında sinyal sistemi

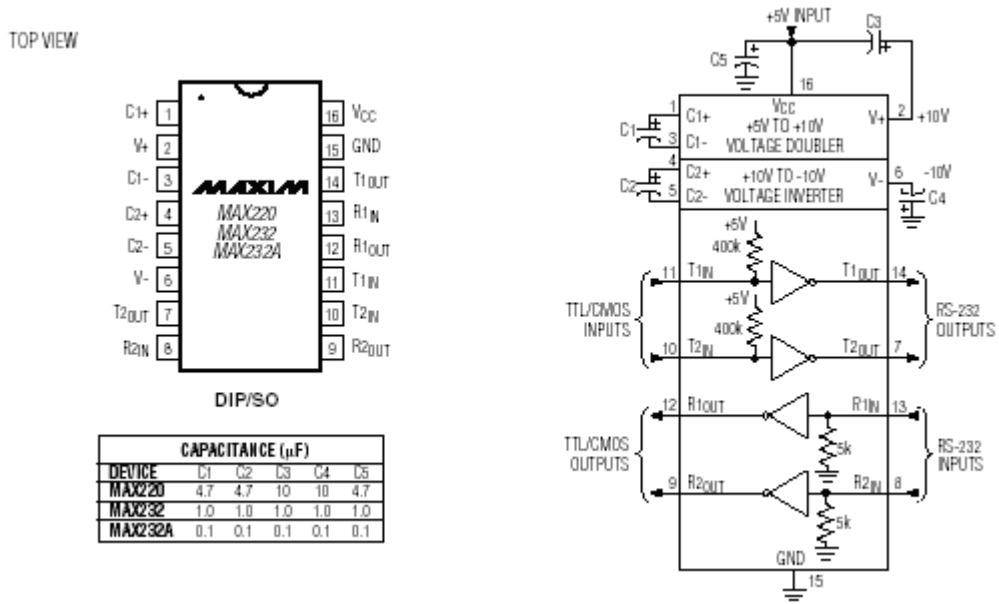
3.3. RS232C İletişim Protokolü

RS232C Amerika Birleşik Devletleri endüstriyel elektrik kurumunca (Electric Industrial Association) standartlaştırılmıştır. RS232C'nin amacı ise kişisel bilgisayarlara terminallerin ve modemlerin kolayca bağlanmasıdır. RS232C'nin gerilimi standarttır ve artı eksi 12 Volttur (artı eksi 3V'tan fazlasıyla da çalışabilir.). 0 Volt ile de bağlantısını kesebiliriz.



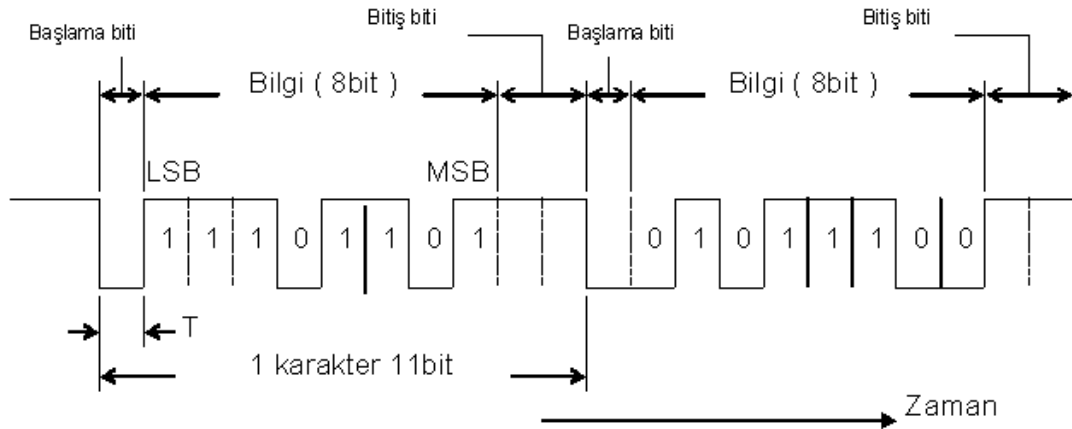
Şekil 3.4: RS232C'nin gerilim seviyesi

Özel entegrelerle artı 5 V'tan artı eksi 9 V gibi gerilimler kullanılabilir. (MAX232)



Şekil 3.5: Seri iletişim entegresi (MAX232)

3.4. Seri Bilgi Aktarım Formatı

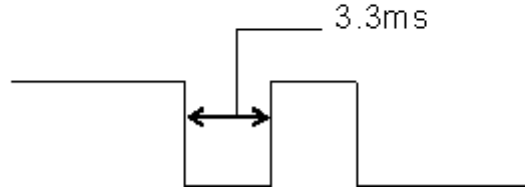


Şekil 3.6: Seri bilgi aktarımı formatı

Seri bilgi iletim aktarım formatı şekil 3.6'da görülmektedir. Bilgi çıkışı iletim hattında Şekildeki gibi soldan sağa doğrudur. Bilgi sürekli 1 ise iletişim yok demektir. Bilgi gönderilmeye başlandığı zaman başlangıç biti olarak 1. bit olarak 0 gönderilir. Sonra çıkış bilgisi ardarda gönderilir. Şekil üzerinde 8 bitlik bir bilgi gönderilmesi örneklenmiştir. Daha sonra gönderilen bilginin sonunu belirtmek için bitiş biti gönderilir. Bu stop biti, şekilde görüldüğü gibi 2 bitten meydana gelmektedir. Böylece bir karakter, 1 bit başlangıç biti, 8 bit bilgi biti, 2 bitten bitiş biti olmak üzere 11 bitten iletişim sağlayabilmektedir. Receiver, bir karakter için başlangıç ve bitiş bitini ayırarak bilgi bitini de ekleyerek paketleme yapar.

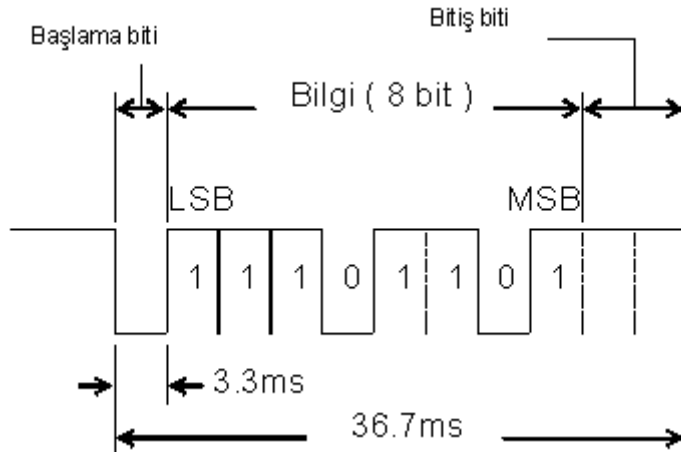
Bilgi aktarım hızı 1 saniyede gönderilen bitlerin sayısı ile ifade edilir. Birim olarak da (**bit per second**) bps kullanılır. Örneğin, bilgi aktarım hızı 300 bps olan olan dalğanın bit genişliği süresini hesaplayınız.

$$T = \frac{1}{bps} = \frac{1}{300} = 0.00333 = 3.3[ms]$$



Şekil 3.7: Bit genişliği

300 bps'den anladığımız bir karakter 11 bitten olduğundan dolayı bir karakterin gidiş hızı $11 \times 3,3 = 36,3$ ms olur. Ayrıca 1 saniyede giden karakter sayısını da yaklaşık olarak bulmak istersek $300 \text{ bps} / 11 \text{ bit} = 27$ karakter olur.

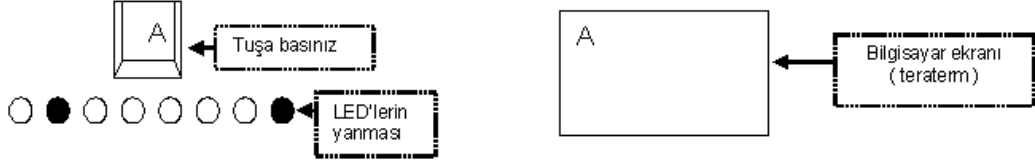


Şekil 3.8: Bilgi (data) gönderme zamanı (300 bps)

Tipik bps hızları (300, 600, 1200, 2400, 4800, 9600)

UYGULAMA FAALİYETİ

Klavyeden girilen karakterin ASCII kod karşılığına göre LED'lerin yanmasının sağlanmasının yanı sıra ASCII karşılığını bilgisayara gönderilmesi ve ekranda ona karşılık gelen karakterin görünmesi programını yazınız.



İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Yazacağınız programın akış diyagramını çiziniz.	<ul style="list-style-type: none">➤ Akış diyagramı hazırlamadan yazılan programlarda hata yapma oranı daha fazladır. Bunun için (özelikle ilk öğrenme aşmasında olduğumuz için) mutlaka akış diyagramı çiziniz.
<ul style="list-style-type: none">➤ MPLAB IDE editöründe program kodlarınızı yazıp derleyiniz.	<ul style="list-style-type: none">➤ Programı yazmaya başlamadan önce MPLAB IDE programının “configure” menüsünde “select device” seçeneği ile gelen pencerede PIC16F84’ün seçili olup olmadığını kontrol ediniz.
<ul style="list-style-type: none">➤ Hata oluşursa hatanın nedenini araştırınız.	<ul style="list-style-type: none">➤ Programı derledikten sonra hata oluşursa ekranda görüntülenen ERR uzantılı dosya hatayı programın hangi satırında olduğunu gösterir. Buradan hatanın nerede olduğunu anlayabilirsiniz.
<ul style="list-style-type: none">➤ Programlama kartınızı (LPT portuna) bilgisayarınıza bağlayınız.	<ul style="list-style-type: none">➤ Bağlantıları yaparken DC adaptörünüzün bağlı olmamasına dikkat ediniz.
<ul style="list-style-type: none">➤ DC güç anahtarı ile devreye gerilim uygulayınız.	<ul style="list-style-type: none">➤ Her HEX dosyası yüklediğinizde konfigürasyon ayarları yapmak zorundasınız. Özellikle kristal ayarı (şu an için XT)ve WDT (disable olmalı) yapmalısınız.
<ul style="list-style-type: none">➤ Programlayıcıya yaptığınız programın derleme sonucu oluşan HEX dosyasını yükleyiniz.	<ul style="list-style-type: none">➤ Programlayıcının alt kısmında “Programing FAILD” mesajı belirdi ise deney kartınızın bilgisayar bağlantısı veya kart üzerinde temassızlık olabilir, kontrol ediniz.
<ul style="list-style-type: none">➤ Kristal ve WDT ile ilgili ayarları yapıp PIC’e programı yükleyiniz.	
<ul style="list-style-type: none">➤ Programın çalışmasının doğru olup olmadığını kontrol ediniz. Eğer çalışmada bir hata var ise gidererek yükleme işini tekrarlayınız.	
<ul style="list-style-type: none">➤ Sonuç raporunu hazırlayınız.	

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. () Seri iletişimde veri taşımakta 1 adet, paralelde ise 8 adet hat kullanılır.
2. () Broad band sistemi bilgi taşıma işlemini kolaylaştırmak için geliştirilmiş değişken dalga formu kullanan sistemdir.
3. () RS232 standardı paralel port için geliştirilmiştir.
4. () Seri iletişim 0 sinyali ile başlar 0 sinyali ile biter.
5. () Bir karakterden oluşan bilgi gönderimi toplam 11 bit ile sağlanır.
6. () 300 bps'nin anlamı bir saatte 300 karakter gönderim hızı demektir.
7. () 1200 bps hızla 0,83 ms zaman aralıkları ile 109 karakter gönderilebilir.
8. () Y harfinin Ascii tablosu üzerindeki heksadesimal karşılığı '59h' dir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise "Modül Değerlendirme"ye geçiniz.

MODÜL DEĞERLENDİRME

Bu modül kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. LED devresi uygulamasını tam olarak yapıp çalıştırabildiniz mi?		
2. 2 veya daha fazla LED ile uygulama yapıp çalıştırabildiniz mi?		
3. Mikrodenetleyici ile display kontrolünü yapabildiniz mi?		
4. Display kod çözücü programını doğru olarak yazabildiniz mi?		
5. Seri iletişimin sinyal yapısını doğru olarak anlayabildiniz mi?		
6. RS232 standardının ne anlam ifade ettiğini anlayabildiniz mi?		
7. Seri iletişim programı yazıp çalıştırabildiniz mi?		

DEĞERLENDİRME

Değerlendirme sonunda “Hayır” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “Evet” ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1.	Yanlış
2.	Doğru
3.	Doğru
4.	Yanlış
5.	Doğru
6.	Yanlış
7.	Yanlış
8.	Doğru
9.	Doğru
10.	Yanlış
11.	Doğru
12.	Doğru

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1.	Doğru
2.	Doğru
3.	Yanlış
4.	Doğru
5.	Doğru

ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

1.	Doğru
2.	Doğru
3.	Yanlış
4.	Yanlış
5.	Doğru
6.	Yanlış
7.	Doğru
8.	Doğru

KAYNAKÇA

- TERAMOTO Koshi, T.İŞBİLEN, M. GÜNEŞ, **PIC16F84 MİKRODENETLEYİCİ TEMEL BİLGİLERİ VE PROGRAMLAMASI**, ETOGM-JICA, Eylül, 2003.