

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**

BİLİŞİM TEKNOLOJİLERİ

MİKRODENETLEYİCİ UYGULAMALARI

Ankara, 2015

- 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	2
1. MİKRODENETLEYİCİ İLE AD VE DA ÇEVİRİM.....	2
1.1. Çeviriciler	2
1.2. Dijital-Analog Dönüştürücüler (DAC,Dijital-Analog Converter)	3
1.2.1. DAC.....	3
1.2.2. Kavramlar	4
1.2.3. Çalışma Prensibi	5
1.2.4. DAC Entegreleri	7
1.2.5. Mikroişlemci Uyumlu DAC'lar.....	8
1.2.6. Dijital Analog Çevirici Uygulaması (DAC)	9
1.3. Analog-Dijital Dönüştürücüler (ADC, Analog-Dijital Converter)	11
1.3.1. ADC.....	11
1.3.2. Çalışma Prensibi	11
1.3.3. ADC Entegreleri	15
1.3.4. Mikroişlemci Uyumlu ADC'ler.....	16
1.3.5 Analog Dijital Çevirici Uygulaması (ADC)	16
UYGULAMA FAALİYETİ	20
ÖLÇME VE DEĞERLENDİRME	21
ÖĞRENME FAALİYETİ-2	23
2. MİKRODENETLEYİCİ UYGULAMALARI	23
2.1. Led Gösteri.....	23
2.2. Zamanlayıcı.....	32
2.3. Mikrodenetleyici Bilgisayar Haberleşmesi	37
2.4. Bilgisayar Ayarları.....	40
ÖLÇME VE DEĞERLENDİRME	41
MODÜL DEĞERLENDİRME	42
CEVAP ANAHTARLARI.....	43
KAYNAKÇA	44

AÇIKLAMALAR

ALAN	Bilişim Teknolojileri
DAL/MESLEK	Bilgisayar Teknik Servis
MODÜLÜN ADI	Mikrodenetleyici Uygulamaları
MODÜLÜN TANIMI	Bu modül, mikrodenetleyiciler ile sistem kontrolü uygulamaları yapma yeterliklerini kazandıran bir öğrenme materyalidir.
SÜRE	40/24
ÖN KOŞUL	Mikrodenetleyici ile Dijital İşlemler modülünü tamamlamış olmak
YETERLİK	Mikrodenetleyici ile çevrim kontrolü yapmak
MODÜLÜN AMACI	Genel Amaç Bu modül ile gerekli ortam sağlandığında; mikrodenetleyici ile çevirim kontrollerini ve devre uygulamaların yapabileceksiniz. Amaçlar <ol style="list-style-type: none">1. Mikrodenetleyici ile A-D, D-A çevirim kontrolü yapabileceksiniz.2. Mikrodenetleyicili uygulamalar yapabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam: Bilgisayar, mikrodenetleyici programlama kartı Donanım: Bilgisayar, projeksiyon cihazı, çizim simülasyon programı, kataloglar
ÖLÇME VE DEĞERLENDİRME	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen, modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Mikrodenetleyiciler günlük hayatta kullandığımız pek çok cihaz içerisinde bulunmaktadır. Mikrodenetleyicileri otomobillerde, cep telefonlarında, kameralarda, faks-modem cihazlarında, fotokopi, radyo, TV ve bazı oyuncaklar ile bilgisayar alanında masa üstü ve diz üstü bilgisayarlar, modemler, yazıcılar, tarayıcılar, hub, router, pemcia kartlar, pdalar ve dijital kameralar içerisinde yer almaktadır.

Birçok durumda şimdiye kadar gördüğünüz sayısal sistemlerden başka, analog bilginin de işlem sürecine dâhil edilmesi gerekir. Görüntü, ses gibi analog değerlerin işlenmesinde ve taşınmasında mikrodenetleyici tabanlı devreler de kullanılır. Bu nedenle analog sinyal yapısını bilmek gerekir.

Bu modülde analog sinyal ile dijital bilginin yapısı, bunların birbirine nasıl dönüştürülebileceği açıklanacak sonra da mikrodenetleyici ile analog ve dijital sinyaller kullanılarak uygulamalar yapılacaktır.

Bu modülü tamamladığımızda mikrodenetleyici ve A/D, D/A çevirim kullanarak analog ve dijital sinyal içeren sistemlerin kontrolünün ve devre uygulamalarının yapılmasına yönelik yeterliklere sahip olacaksınız.

Mikrodenetleyici uygulamalarının bu modülünde assembler dilinde programlama kurallarına uygun akış diyagramını çizerek programlarını yazacak ve bu programların mikrodenetleyiciye yükleyerek programı test edip hatalarını ayıklayabilir duruma gelecek ve bu şekilde modül sonunda artık siz de bir mikrodenetleyicilerde AD ve DA çeviricileri tanımış ve amaca göre çevirim programını yazabilme seviyesine çıkmış olacaksınız.

ÖĞRENME FAALİYETİ-1

AMAÇ

Mikrodenetleyici ile A-D, D-A çevirim kontrolü yapabileceksiniz.

ARAŞTIRMA

- ADC ve DAC entegrelerinin yapısını öğrenip devrelerini inceleyiniz.
- ADC ve DAC dönüştürücüler hakkında bilgi edininiz.
- ADC ve DAC işlemlerinde niçin mikrodenetleyicinin kullanıldığını araştırınız.

1.MİKRODENETLEYİCİ İLE AD VE DA ÇEVİRİM

1.1. Çeviriciler

Doğada ölçebildiğimiz ağırlık, ses, sıcaklık vb. bilgilerin çoğu analog bilgidir ve bu bilgilerin ölçülen iki değerinin arasında sonsuz sayıda değer yer alır. Bu şekilde sürekliliği olan bilgilere analog bilgi denir. Sonsuz bilginin saklanması ve işlenmesi sayısal sistemlerde mümkün olmadığı için bu bilginin sayısal sistemler tarafından işlenebilmesi için belirli aralıkta örneklenmesi gereklidir.

Ara değerlerin olmadığı lojik (mantık) bilgiler ise dijital bilgi olarak adlandırılır. Dijital bilginin ikilik sayısal formda iki durumu vardır ve bu iki durumda analog kapının durumu yüksek ve alçak seviye, lojik 1 ve lojik 0, kapı tam açık ya da tam kapalı şeklinde ifade edilebilmektedir.

Doğada ölçülen bu bilgilerin mikroişlemciler tarafından mikrodenetleyicilere ikilik sayısal bilgi şeklinde işlenmesi gerektiğinden bu çevirim işini yapan devreler bulunmaktadır ya da tam tersi düşünülebilir. Bu durumda analog sinyali sayısal bilgiye çeviren devrelere dijital analog çevirici, DAC (Dijital to Analog Converter), ya da analog bilgiyi dijital bilgiye çeviren devrelere analog dijital çevirici ADC (analog-to-dijital converter) denir.

Bilgisayar ve kontrol sistemlerinde DA ve AD çevirimleri pek çok donanım parçası içerisinde kullanılmaktadır.

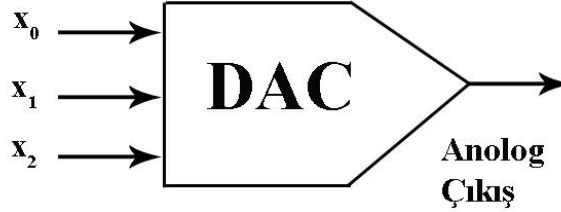
1.2. Dijital-Analog Dönüştürücüler (DAC,Dijital-Analog Converter)

1.2.1. DAC

1 ve 0 gibi dijital bilgileri giriş olarak alan ve çıkışında giriş değerlerindeki değişime göre farklı değerlerde akım veya gerilim üreten devrelere veya entegrelere dijital analog çeviriciler ve bu dönüştürme işlemine de dijitalden anloga çevirme işlemi adı verilir.

Bir başka deyişle işaretli bir binary sayıyı bir elektrik gerilimi veya bir elektrik akımından birine çeviren çeviricilere dijital analog çeviriciler denir. Kısaca DAC olarak adlandırılır.

Dijital analog çeviriciler giriş olarak birden fazla dijital değeri alabilir. Dijital giriş değeri sayısı, dijital analog çeviricinin bağlı olduğu dijital devrenin çıkış sayısına eşittir.



Şekil 1.1: DAC Blok diyagramı

Şekil 1.1'de görüldüğü gibi her bir girişi 0 ve ya 1 sayısını taşıyabilecek 3 giriş hattı ve (x_0, x_1, x_2) her bir giriş değeri için giriş değerine eşit bir analog çıkış voltaj karşılığı vardır.

GİRİŞLER			ÇIKIŞLAR
X_2	X_1	X_0	Analog Çıkış
0	0	0	0 V
0	0	1	1 V
0	1	0	2 V
0	1	1	3 V
1	0	0	4 V
1	0	1	5 V
1	1	0	6 V
1	1	1	7 V

Tablo 1.1: Doğruluk tablosu

Bu giriş –çıkış ilişkisinin incelenmesi, 1 Voltluk ağırlığın girdi değeri (x_2) verilerek çıktı değerinin hesaplanabileceğini gösterir. Her bir giriş 1 ağırlığa sahiptir. Tablo 1.1'de görüldüğü gibi çıkış voltajı ikili girişlerin bir olduğu durumlar için ağırlıkların toplamıdır.

1.2.2. Kavramlar

Dijital verilerin analog veriye çevrilmesinde analog çıkışın değerinin belirlenmesinde etkili olan bazı esaslar ve kavramlar vardır.

1.2.2.1. LSB

Dijital değerlerin daha fazla anlam ifade etmesi için çok sayıda bitin bir arada kullanılması gereklidir. Örneğin bir bit ile sadece 2 farklı (1 ve 0) durum ifade edilirken 2 bit ile 4 farklı durum ifade edilir (00,01,10,11).

Dijital devrelerinde daha fazla çıkış durumu ifade etmek için çok sayıda çıkış biti vermesi olağandır. Ancak bitlerin sayısı çoğaldıkça dijitalden analoga dönüşüm sırasında problemler yaşanmaktadır. Çok sayıda giriş biti alan bir DAC, bunları çıkışa analog değer olarak aktarırken bitlerin çıkış akım ve gerilimine etki oranını nasıl belirleyeceği konusu ikilik sayı sistemi ile çözümlenmiştir.

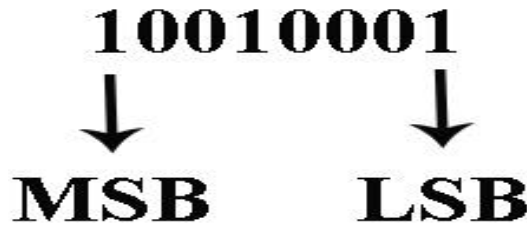
Giriş bitleri peş peşe dizilerek ikilik sistemde rakam elde edilirse, sağdan sola doğru basamakların değerleri artmakta, artış oranı da sayı sisteminin taban değerine göre üstel şekilde belirlenmektedir. Yani, girişlerin sıralaması çıkışa etki oranını belirler.

Binary sayıları yazılırken en sağdaki basamağa en düşük değerlikli bit LSB (Least Significant Bit) adı verilir. Dijitalden analoga dönüşüm sırasında analog çıkış üzerindeki değere en az etki eden dijital değerdir. Tablo 1.1'de LSB değeri x_0 'dir.

1.2.2.2. MSB

Yukarıda bahsedildiği gibi en soldaki basamağa en yüksek değerlikli bit MSB (Most Significant Bit) adı verilir. Dijitalden analoga dönüşüm sırasında analog çıkış üzerindeki değere en fazla etki eden dijital değerdir. Tablo 1.1'de MSB değeri x_2 'dir.

Şekil 1.2'de görüldüğü gibi ikilik sayı sisteminde en soldaki basamağa MSB, en sağdaki basamağa LSB adı verilir. İkilik sayı sisteminde en sağdaki rakamdan itibaren basamaklar ($\dots, 2^7, 2^6, 2^5, 2^4, 2^3, 2^2, 2^1, 2^0$) ikinin katları şeklinde ($\dots, 128, 64, 62, 16, 8, 4, 2, 1$) sola doğru artarak devam eder. Her basamağın ağırlığı farklıdır. Örneğin 3. basamağın 1 ya da 0 olması sayıyı ± 4 etkilerken, 8. basamağın 1 ya da 0 olması sayıyı ± 128 etkiler.



Şekil 1.2: LSB ve MSB

1.2.2.3. Tam Skala

DAC çeviricilerde giriş bitlerinin hepsinin 1 olması durumuna tam skala adı verilir. Giriş olarak verilen tüm bitler 1 olduğundan, çıkış voltajı maksimum değerde olur.

1.2.2.4. Çözünürlük

DAC çıkış geriliminin giriş kodunun sayısal değeri +1 kadar arttığında çıkış geriliminin ne kadar değişeceğini gösteren ifadeye çözünürlük (hassasiyet) denir. Başka bir ifadeyle, bir D/A dönüştürücünün çıkışında oluşabilecek en küçük analog değer en büyük analog değere oranına çözünürlük denilir.

Çözünürlük oran olarak ($1/2^n$) gösterildiği gibi direk giriş bit sayısı olarak da (n bitlik DAC gibi) ifade edilir. DAC'ın giriş bitlerinin sayısı ne kadar fazla ise çözünürlük de o kadar artar.

Örneğin 4 bitlik bir DAC girişinde maksimum $2^4=16$ adet farklı dijital değer olabilir. Bu DAC'ın çözünürlüğü $1/16$ 'dır. Çözünürlüğün artması demek sayısal değer azalması anlamına gelmektedir.

1.2.2.5. Giriş – Çıkış İlişkisi

Giriş bitlerindeki değişim sonucu çıkış akımında veya geriliminde gözlenen değişimdir. LSB'den MSB'ye doğru bitlerdeki ağırlık değeri artacağından çıkış voltajı üzerindeki etkisi de artacaktır.

1.2.2.6. Doğruluk

D/A dönüştürücü çıkışında olması beklenen analog değer ile gerçekleşen değer arasındaki ilişkiye doğruluk denir. Bir DAC'ın bütün dijital girişleri düşük değerlikli "0" olduğunda analog çıkışın "0" olması gerekir. Ancak çıkışta sıfırdan farklı değer oluşursa bu hatadır.

1.2.3. Çalışma Prensibi

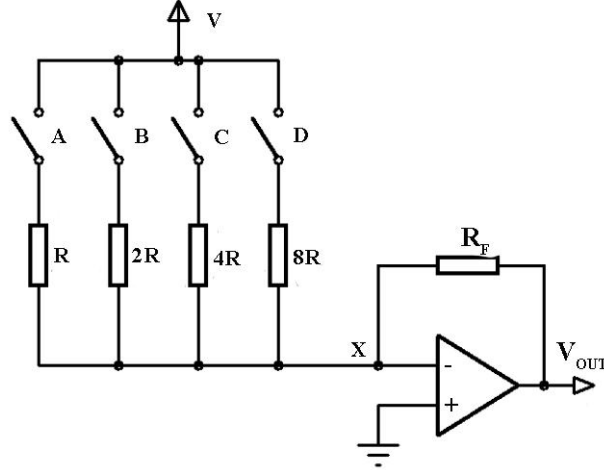
Dijital değerlerin analog değerlere dönüştürülmesinde temel eleman olarak kullanılan işlemsel yükselteçler, girişe uygulanan gerilim değerini yine giriş ve çıkışına bağlanan dirençlerle, belirlenen bir oranla çıkışa aktaran devre elemanıdır. Giriş değerinin çıkışa etki oranının belirlenebilmesi sayesinde girişi oluşturan dijital değerlerin çıkışa aktarılma oranı belirlenebilmektedir.

İşlemsel yükselteçler, DAC devrelerinde toplayıcı olarak kullanılabilen giriş bitlerinin çıkışa etki oranı dirençler ile belirlenerek yükseltilmiş bir analog çıkış elde edilmektedir.

Dijital sinyalleri, analog sinyallere dönüştürmek için iki temel yaklaşım vardır.

- Ağırlık dirençli (paralel girişli) D/A dönüştürücü
- R-2R merdiven tipi D/A dönüştürücü

1.2.3.1. Ağırlık Dirençli (Paralel Girişli) D/A Dönüştürücü



Şekil 1.3: Ağırlık dirençli (paralel girişli) D/A dönüştürücü

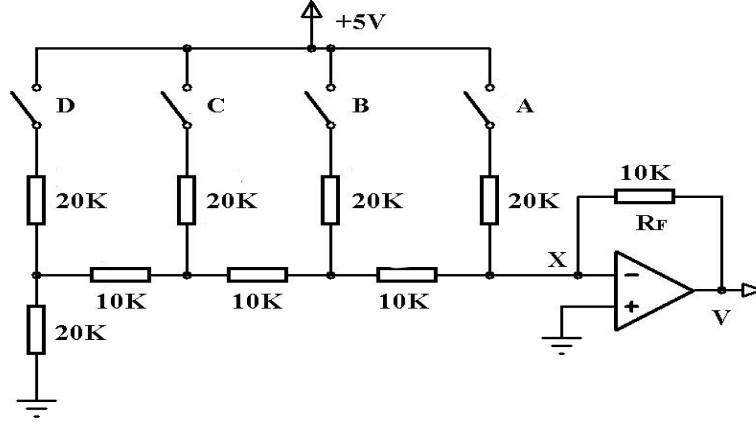
Ağırlık dirençli D/A dönüştürücüde, dijital girişin çıkışa yansıtılma oranı, dijital girişlerin önünde yer alan dirençler ile ters orantılı bir şekilde belirlenir. Dirençler arasındaki oran 2'nin katları şeklinde kullanılır.

Şekil 1.3'te görüldüğü gibi A, B, C, D dijital girişlerin X ile gösterilmiş düğüm etkileri farklıdır. Önünde yüksek direnç değeri olan dijital giriş, X noktasına daha az akım ulaştıracağından çıkıştaki etkisi de daha az olacaktır. D en değerli bit (LSB) olup referans geriliminin 1/16'sı kadar çıkışa etkiler ve devrenin çözünürlüğünü belirler. Bu durumda her bir basamak değeri D'nin etkilediği kadar artar. A ise en değerli bit (MSB) olup çıkışa tam skala değerinin yarısı kadar etki eder.

A, B, C, D dijital girişlerin ağırlıklarının toplamı kadar çıkışta gerilim elde edilir. Şekil 1.3'te görülen devrenin çıkış gerilimi aşağıdaki gibi hesaplanır.

$$V_{\text{çk}} = (-R_f / 8R) * V_{\text{ref}} * (8A + 4B + 2C + D)$$

1.2.3.2. R-2R Merdiven Tipi D/A Dönüştürücü



Şekil 1.4: R-2R merdiven tipi D/A dönüştürücü

Dijital bilginin analog bilgiye çevrilmesi için en çok kullanılan yöntemdir. Direnç değerlerinin R-2R olarak sıralanması ve çıkış dalga şeklinin merdiven basamağı şeklinde artması sebebiyle R-2R merdiven tipi DA dönüştürücü adını almıştır. Şekil 1.4'te görüldüğü üzere X ile gösterilmiş düğüme A, B, C, D dijital girişlerin etkileri farklıdır. Önünde direnç değeri çok olan dijital giriş, X noktasına daha az akım ulaştıracağından, çıkıştaki etkisi daha az olacaktır. Bu durumda D en değerli bit (LSB) olup referans geriliminin 1/16'sı kadar çıkışı etkiler ve devrenin çözünürlüğünü belirler. Her bir basamak değeri D'nin etkilediği değer kadar artar. A ise en değerli bit (MSB) olup tam skala değerinin yarısı kadar çıkışa etki eder. Şekil 1.4'te görülen R-2R merdiven tipi devrenin çıkış gerilimi aşağıdaki gibi hesaplanır.

$$V_{\text{çk}} = (-R_f / 16R) * V_{\text{ref}} * (8A + 4B + 2C + D)$$

1.2.4. DAC Entegreleri

D/A dönüştürücü devreleri hazır entegre olarak üretilmiştir. Çeşitli dijital devrelere direk bağlanabilecek şekilde paralel girişli DAC entegreleri üretildiği gibi mikrodenetleyicilerle birlikte kullanıma uygun paralel veya seri girişli değişik çözünürlüklerde entegreler mevcuttur.

Tablo 1.2'de bazı DAC entegreleri özellikleri ile birlikte verilmiştir. En çok karşılaşılan DAC entegrelerinin çeşitlerini ve özelliklerini inceleyiniz.

DAC	Çözünürlük	DAC Kanal Sayısı	Giriş Haberleşme Tipi	Çevrim Süresi	Çıkış Tipi Gerilim Sınırı	Kullanım Alanı
AD558	8 Bit	1	Paralel	1 μ s	0-10 V	Genel
DAC0800	8 Bit	1	Paralel	100 ns	\pm 20 V	Genel
MC1408	8 Bit	1	Paralel	70 ns	Akım çıkışlı	Hızlı DAC
ZN425E	8 Bit	1	Paralel	2 μ s	0-2,5 V	Genel
AD5320	12 Bit	1	Seri	12 μ s	0-5 V	Mikrokontrolör
LTC1450	12 Bit	1	Paralel	14 μ s	0-5 V	Endüstriyel
MAX5316	16 Bit	1	SPI	3 μ s	Programlanabilir gerilim, akım	Haberleşme, Medikal cihazlar oto. Test sistemleri
MAX5879	14 Bit	1	Direkt RF Sinyal	2,3 Gbps	Akım çıkışlı	Dijital video uyg. Kablosuz altyapı uyg.

Tablo 1.2: DAC entegreleri

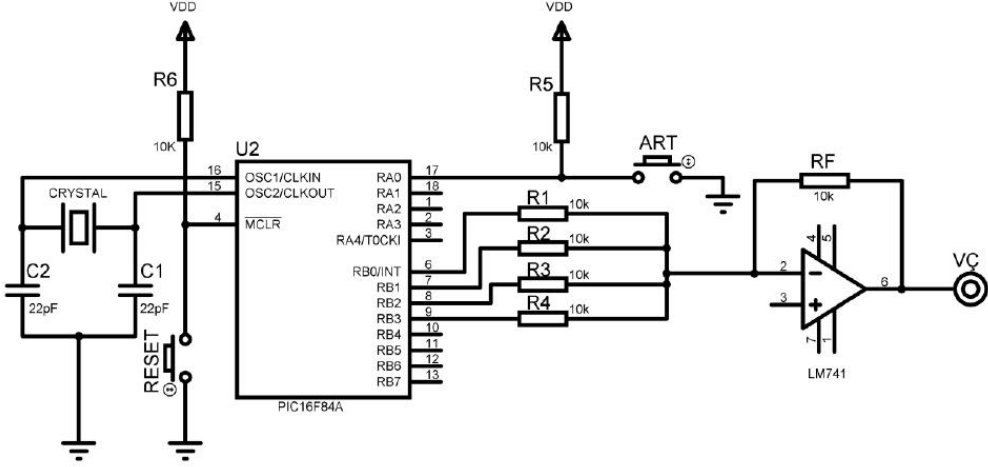
1.2.5. Mikroişlemci Uyumlu DAC'lar

Mikroişlemciler dijital veri ile çalıştığından çalışması sırasında, data yolu adı verilen bir grup iletken hat üzerinde sürekli olarak dijital veri akışı gerçekleşmektedir. Bu dijital veriyi mikroişlemci dışında bir analog devrede değerlendirmek istersek bir DAC dönüştürücü kullanılmalıdır. Ancak yukarıda bahsi geçen devreler ve entegreler mikroişlemciler ile uyumlu çalışmayabilir. Mikroişlemciler ürettiği kontrol sinyalleri sayesinde çevre birimlerinin çalışmalarını idare edebilirler. Bu nedenle mikroişlemciler ile uyumlu olan DAC dönüştürücülerin mikroişlemciden gelen data hatlarının yanında kontrol hatlarından da giriş kabul etmesi amacı ile üretilmiş dijital analog dönüştürücülere mikroişlemci uyumlu DAC adı verilir.

Mikroişlemci uyumlu DAC entegreleri, belirli bir mikroişlemci ile uyumlu olacak şekilde üretildiğinden data ve kontrol hatlarının sayısı değişkendir. Çalışma hızı, data hattı sayısı denkliği ve kontrol sinyalleri ile DAC entegresinin kontrol edilebilmesi gibi özellikler mikroişlemci uyumlu DAC'larda önemlidir.

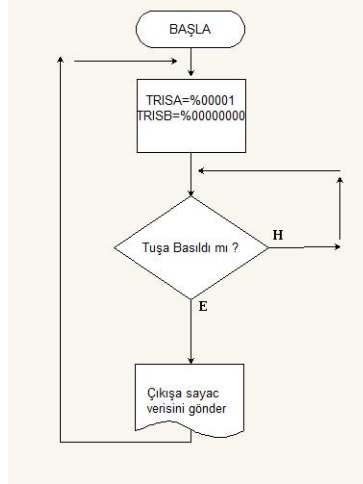
1.2.6. Dijital Analog Çevirici Uygulaması (DAC)

Pic 16F84A yapılan uygulamada PORTA'nın RA0 bitine bağlı olan bir butona her basıldığında çıkıştaki gerilimi artıran, dijital analog çevirici olarak ağırlık dirençli DAC devresinin kullanıldığı dijital analog çevirici uygulama devresini çizerek mikrodenetleyici programını yazınız.



Şekil 1.5: Dijital analog çevirici uygulaması (DAC)

Programın akış diyagramı Şekil 1.6'daki gibi olacaktır.



Şekil 1.6: DAC Akış diyagramı

➤ Uygulama Programı

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

```
SAYAC1    EQU H'0C'      ; Gecikme alt programlarında kullanılan değişken
SAYAC2    EQU H'0D'      ; Gecikme alt programlarında kullanılan değişken
SAYAC3    EQU H'0E'      ; Gecikme alt programlarında kullanılan değişken
ART        EQU H'0F'      ; programlarda kullanılan değişken
```

→portları ayarla

```
CLRF      PORTB          ;PORTB' yi temizle
BSF       STATUS,5      ;BANK1' e geç
CLRF      TRISB         ;PORTB çıkış
MOVLW    H'FF'          ;W <-- H'FF'
MOVWF    TRISA          ;PORTA giriş
BCF       STATUS,5      ;BANK0 a geç
```

→Start butonuna basılıncaya kadar bekle

```
BUTON
BTFSC    PORTA,0        ;PORTA nın 1.biti 0 mı?
GOTO     BUTON          ;Hayır, tekrar test et
```

→Değişkeni artır ve porta gönder

```
MOVLW    H'00'          ;W H'00' 7
MOVWF    ART            ;ART =W
DON
MOVF     ART,W          ;W ART
ANDLW    B'00001111'    ;Portun düşük olan bitlerini sıfırla
MOVWF    PORTB          ;PORTB' ye bilgiyi gönder
INCF     ART,F          ;ART değişkeninin içeriğini bir artır
CALL     GECIKME        ;GECIKME alt programını çağır
GOTO     BUTON          ;Butonu kontrol et
```

→gecikme alt programı

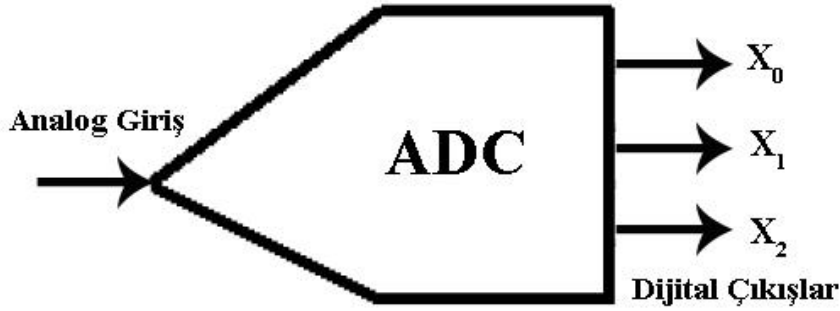
```
GECIKME
MOVLW    H'FF'          ;W<--H'FF'
MOVWF    SAYAC1        ;SAYAC1 <--W
D1
MOVLW    H'FF'          ;W<--H'FF'
MOVWF    SAYAC2        ;SAYAC2 <--W
D2
DECFSZ   SAYAC2,F      ;Sayac2 bir azalt ve sıfır mı? kontrol et
GOTO     D3            ;Hayır D3 e git
DECFSZ   SAYAC1,F      ;Sayac1 bir azalt ve sıfır mı?
GOTO     D1            ;Hayır D1 e git
RETURN
END
```

1.3. Analog-Dijital Dönüştürücüler (ADC, Analog-Dijital Converter)

1.3.1. ADC

Sıcaklık, ağırlık veya basınç gibi ölçülebilen büyüklükler, elektronik terazi, termometre veya barometre gibi cihazlar tarafından sensörler yardımıyla algılanır ve bunlarla orantılı akım veya gerilim cinsinden elektriksel değerlere dönüştürülür. Sensörlerden gelen analog sinyaller önce analog dijital çeviriciler sayesinde ikilik sayı sisteminin rakamları ile ifade edilen dijital veriye, daha sonra insanlar için daha anlamlı olan onluk sayı sistemine çevrilir. Örneğin 50 kg veya 25 derece gibi ifade edilir.

Maksimum ve minimum sınırları arasında farklı değerler alarak değişen elektriksel büyüklüklere analog bilgi ya da analog değer, akım ve gerilim gibi analog sinyallerin dijital sinyallere dönüştürme işlemini yapan devrelere de analog dijital çevirici (ADC) denir.



Şekil 1.7: ADC Blok diyagramı

Şekil 1.7’de tek bir analog girişli ve 3 bitlik binary çıkışlı bir ADC’nin blok diyagramı gösterilmiştir.

1.3.2. Çalışma Prensibi

Bir analog sinyal, dijital sinyale çevrilirken analog değer üzerinden belirlenen zaman dilimlerinde örnekler alınır. Her örnek için seviyesine göre kodlanmış dijital bir değer üretilir. Analog sinyaller sürekli sinyaller olduğundan bunları dijitalleştirmek için belli aralıklarda örnekler alınması ve örnekleme sıklığının uygun seçilmesi gerekir. Alınan örnekler herhangi bir değerde olabilir, bu nedenle işaretin dijitale çevrilmesi için kullanılacak seviyenin sınırlı olması gerekir. Bu sınır her bir örnek için kullanılacak kod uzunluğu ya da bit sayısı tarafından belirlenir. Örneğin 4 bitlik bir kodlama yapılacaksa $2^4 = 16$ seviye kullanılabilir. Seviye ve basamak sayısı arttıkça daha iyi kalitede dönüşüm elde edilir.

Analog işaretlerin dijitale dönüştürülmesi “örnekle, karşılaştı, dijital olarak kodla” şeklinde özetlenebilir. Örnekleme ile çevirme işleminde belirli bir analog değer aralığına bir dijital değer karşılık gelmesi sorunu ile karşılaşılabilir. Örneğin 3 V için 111 dijital çıkışını veren bir çevirici 5 V için de aynı çıkışı verebilir.

ADC devrelerinde örnekleme, karşılaştırma ve dijital kodlama işlemleri yapılırken bazı kavramalar karşımıza çıkar. Kısaca bunlardan bahsedelim.

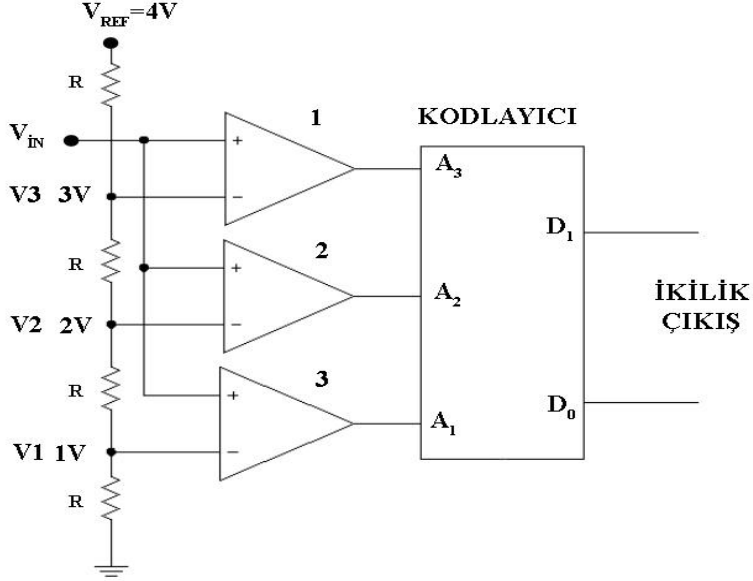
- **Çevirim zamanı:** ADC'nin dönüştürme hızıdır. Kısa olması tercih edilir.
- **Örnekleme frekansı (Çevirim frekansı):** Bir saniye içinde yapılan ADC dönüşüm sayısı anlamına gelir, sps (Sample Per Second) birimi kullanılır.
- **Çözünürlük:** ADC'nin analog girişindeki en küçük değer değişimine karşılık çıkışında dijital farklılık oluşturma yeteneğidir. Analog-dijital dönüştürücüde çözünürlüğün yüksek olması istenir.
- **Quantum seviyesi (Bölüntü seviyesi):** ADC girişine uygulanan analog sinyal, minimum ve maksimum genlik değerleri arasında eşit aralıklara bölünür. Her aralık dijital çıkışta bir bitlik değişime neden olur. Örneğin “n” sayıda dijital çıkışı olan bir ADC 2^n adet ayrık quantum seviyesine sahip demektir. Girişteki analog sinyalin minimum-maksimum arası 2^n adet eşit parçaya bölünmüş olur.
- **Doğruluk:** Girişteki analog değere karşılık olması beklenen dijital çıkış ile gerçekleşen dijital çıkış arasındaki ilişkiye doğruluk denir.
- **Polarite:** Girişe sıfır-pozitif (0, +Vin) sinyaller uygulanabiliyorsa tek yönlü (monolitik) ADC denilir. Tek girişi vardır. Girişe negatif-pozitif (-Vin, +Vin) sinyaller uygulanabiliyorsa çift yönlü (bidirectional) ADC denilir. Çift giriş ucu vardır.

Analog sinyalleri, dijital sinyallere dönüştürmek için 11 farklı teknik bulunmaktadır.

- Paralel karşılaştırıcı (Flash) A/D dönüştürücü
- Sayısal eğimli (Basamak rampalı) A/D dönüştürücü
- Ardışık yaklaşımlı (SAR) A/D dönüştürücü
- Şarj dengeleme sistemli A/D dönüştürücü
- Gerilim/Frekans dönüştürücülü ADC
- Boru hattı tipi (Pipeline) A/D dönüştürücü
- Ayrık zamanlı ADC (Time-Interleaved, TI-ADC)
- Girişi izleyen A/D dönüştürücü
- Tek eğimli A/D dönüştürücü
- Çift eğimli A/D dönüştürücü
- Delta-Sigma A/D dönüştürücü

Bu tekniklerden en çok kullanılan paralel karşılaştırıcı A/D dönüştürücü ile sayısal eğimli A/D dönüştürücüyü açıklayalım.

1.3.2.1. Paralel Karşılaştırıcı (Flash) A/D Dönüştürücü



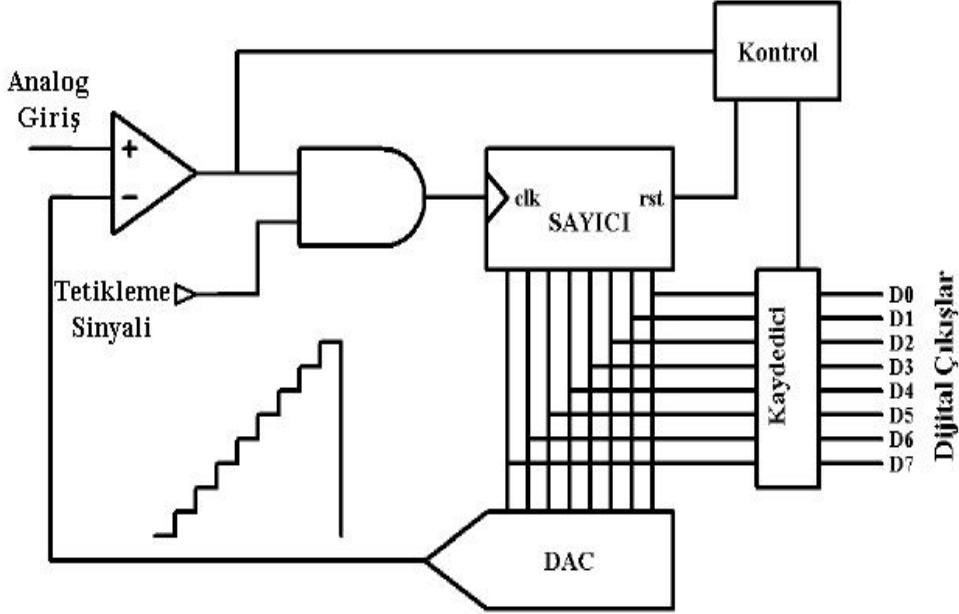
Şekil 1.8: Paralel (Flash) tip A/D dönüştürücü

Analog büyüklüklerin dijital işaretlere dönüştürülmesinde kullanılan en kolay ve hızlı dönüştürücüler flash dönüştürücülerdir. Flash dönüştürücüler çeşitli büyüklükte ve hızlardadır. En hızlı tek yonga dönüştürücüler 8 bitlik dönüşümleri saniye başına 100 milyon dönüştürme ile yerine getirirler. Çevrilecek olan analog sinyal tüm karşılaştırıcı girişlerine aynı anda paralel olarak uygulanır. Karşılaştırıcıların diğer girişlerine ise referans gerilimi uygulanır.

Şekil 1.8’de verilen paralel tip ADC’de uygulanan referans geriliminin 4V olduğu görülmektedir. Bu durumda V3 noktasında 3V, V2 noktasında 2V ve V1 noktasında 1V bulunur. Paralel karşılaştırıcı yönteminde analog giriş işareti, referans gerilimleriyle karşılaştırılır. Girişe ulaşan analog gerilim, karşılaştırıcılardan birinin referans gerilimini aştığında, karşılaştırıcı çıkışında yüksek seviye oluşur ve öncelikli kodlayıcı yardımıyla işaretin sayısal kodu üretilir. Öncelikli kodlayıcı girişine birden fazla yüksek seviye gelebilir. Fakat öncelik en büyüğe verildiği için bu girişin sayısal kodu çıkıştan elde edilir. Uygulanan analog gerilime bağlı olarak karşılaştırıcıların çıkışları 0 ya da 1 lojik durumunu alır. Şekil 1.8’de verilen paralel tip ADC’de kodlayıcıya giren kod (111) iken, referans gerilimi 2V düşürüldüğünde giriş kodu (011) olarak elde edilecektir. Bu çıkışlar bir kodlayıcı devre ile ikili sayı sistemine çevrilir.

Flash dönüştürücülerdeki ana problem çıkış bitlerinin sayısı artırıldığında yüksek sayıda karşılaştırıcıya ihtiyaç duyulmasıdır. n bitlik bir dönüştürücü için $2^n - 1$ tane karşılaştırıcı gereklidir.

1.3.2.1. Sayısal Eğimli (Basamak Rampalı) A/D Dönüştürücü



Şekil 1.9: Sayısal eğimli (Basamak rampalı) A/D dönüştürücü

Analog sinyalin dijitale çevriminde kullanılan diğer bir yöntem sayısal eğimli A/D dönüştürücülerdir. Sayısal eğimli A/D dönüştürücüde; sayıcı, sayıcı için saat darbe (clock pals) üretici, dijital/analog dönüştürücü (DAC), karşılaştırıcı ve kaydedici devre kısımları bulunur. Karşılaştırıcının “+” girişine uygulanan analog giriş gerilimi, girişinden büyük olduğundan çıkış yükseğe çekilecektir. Bu durumda merdiven basamağı şeklinde çıkış gerilimi üretilmeye başlanacak, “VE” kapısının çıkışında tetikleme sinyali oluşacağından sayıcı sayma işlemine başlayacaktır. Bu işlem üretilen merdiven basamağı çıkış geriliminin analog giriş geriliminden büyük olmasına kadar devam edecektir. Böylece karşılaştırıcı çıkış alçalacak, “VE” kapısının çıkışı lojik 0 olacak ve tetikleme sinyali gitmeyen sayıcı sayma işlemini bitirecektir. Kontrol devresi tarafından yetkilendirilen kaydedici, sayıcı verilerini saklayacaktır. Kontrol devresi sayıcıyı sıfırlayıp yeni bir çevrimi başlatır.

1.3.3. ADC Entegreleri

ADC	Çözünürlük Dijital Bit Sayısı	Giriş Sayısı	Çıkış Haberleşme Tipi	Çevrim Frekansı Çevrim Süresi	Dönüştürücü Tipi	Kullanım Alanı
ADC0801 ADC0802 ADC0803 ADC0804 ADC0805	8 Bit	2	Paralel	10 Ksps 100 µs	Ardışık yaklaşım (SAR)	Genel
ZN425E	8 Bit	1	Paralel	1 Ksps 1 ms	Sayısal eğimli	Genel ADC/DAC
MAX186E	12 Bit	8	Seri	133 Ksps	Ardışık yaklaşım (SAR)	Proses kontrol, Otomatik test sistemleri, medikal cihazları, bataryalı araçlar
MAX1447 MAX/1496 MAX1498	4,5 Digit 3,5 Digit 4,5 Digit	2	7 Segment display	512 sps 640 sps 512 sps	Sigma Delta	Dijital voltmetre, Multimetre
ICL7106 ICL7107	3,5 Digit	2	7 Segment display	3 sps 333 ms	Çift Eğim	Dijital voltmetre, Multimetre
LTC2440	24 Bit	2	SPI	Seçilebilir Hız	Sigma Delta	Load cell Termokupl
PCM1870A	16 Bit	4	I2C SPI	5-50 Khz	Sigma Delta	Stereo Audio ADC
ADC820	8 Bit	2	Paralel	720 Ksps 1,4 µs	4+4 Paralel Karşılaştırıcı	Haberleşme
ADC10662 ADC10664	10 Bit	4	Paralel	2.7 Msps 360 ns	6+4 Paralel Karşılaştırıcı	Mobil haberleşme
AD9484	8 Bit	2	Paralel	500 Msps 2 ns	Pipeline	Dijital osilaskop Haberleşme test cihazları Kablosuz haberleşme

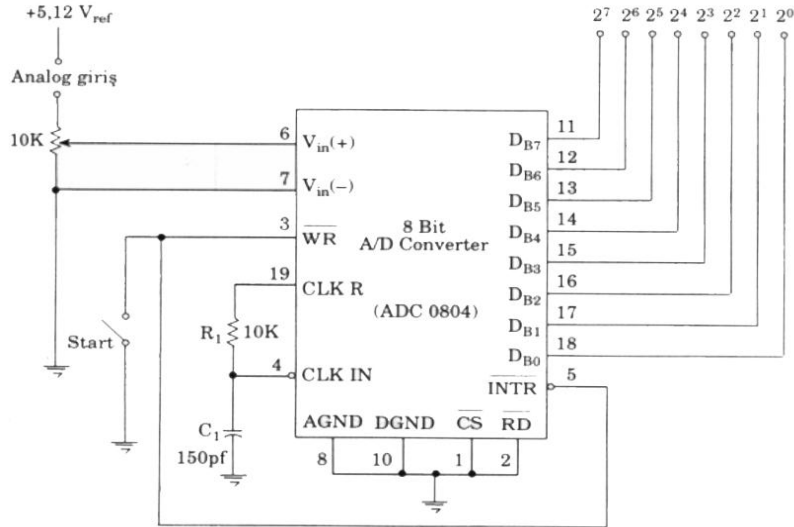
Tablo 1.3: ADC entegreleri

Tablo 1.3'te bazı ADC entegreleri özellikleri ile birlikte verilmiştir. En çok karşılaşılan ADC entegrelerinin çeşitlerini ve özelliklerini inceleyiniz.

1.3.4. Mikroişlemci Uyumlu ADC'ler

Mikroişlemciler dijital veri ile çalıştığından sensörler ile ölçülen değerlerin, mikroişlemci tarafından değerlendirilmesi için dijital değerlere çevrilmeleri gerekmektedir. Ancak bazı ADC devrelerinin ve entegrelerin mikroişlemciler ile uyumlu çalışması mümkün değildir. Mikroişlemciler ürettiği kontrol sinyalleri sayesinde çevre birimlerin çalışmalarını kontrol edebilir. Bu sebeple mikroişlemciler ile uyumlu ADC'lerin mikroişlemcinin data hatlarına bağlandığı gibi kontrol hatlarından giriş kabul etmesi gerekmektedir. Bu amaçla üretilmiş ADC'lere mikroişlemci uyumlu ADC adı verilir.

Şekil 1.10'daki devrede ADC0804 entegresi ile analog dijital çevirici gösterilmektedir. Devrenin girişine uygulanan analog bilgi 8 bitlik dijital bilgiye çevrilmektedir.

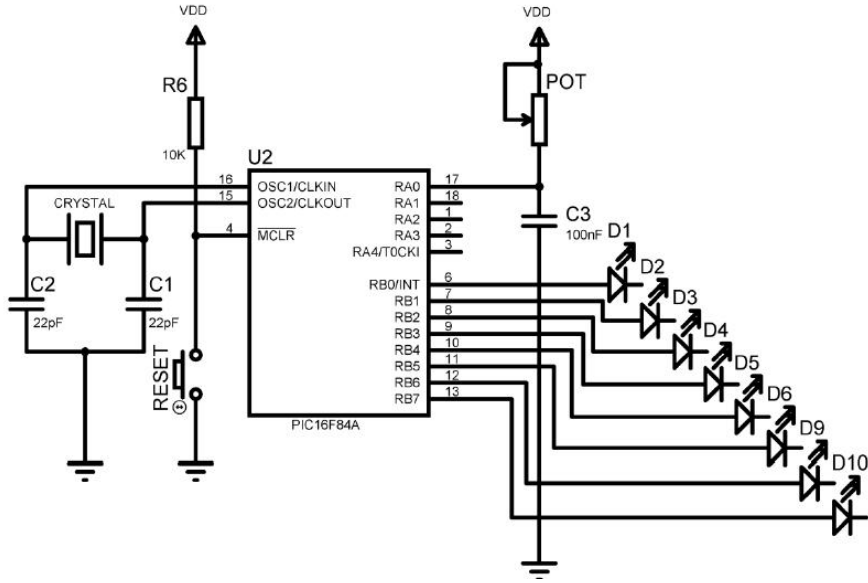


Şekil 1.10: ADC0804 A/D entegresi

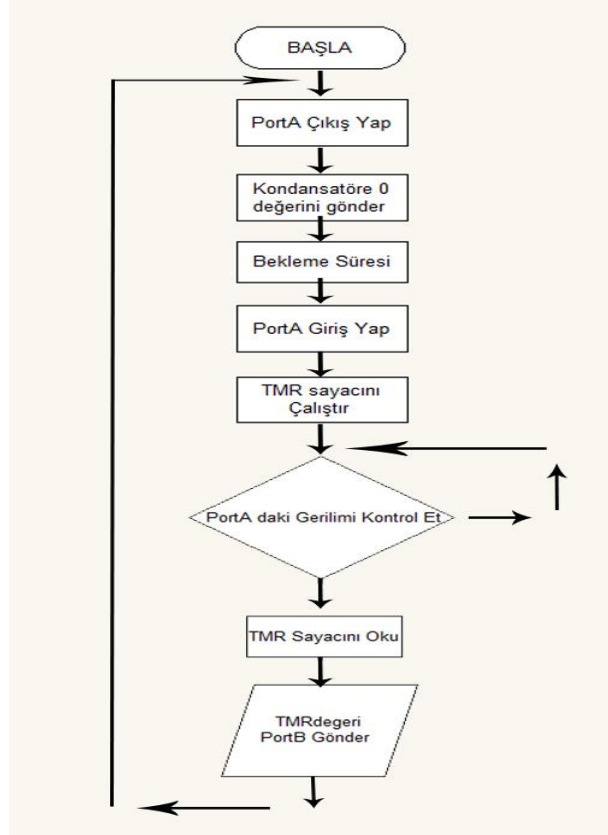
1.3.5 Analog Dijital Çevirici Uygulaması (ADC)

A/D çevrim metodu ile yapılan ölçümde değeri bilinmeyen bir direncin, bir kondansatörü V değerine ne kadar sürede şarj ettiği bulunur. Bunun için TMR0 sayıcısı şarj süresini ölçmek için kullanılır. A/D çevrim metodunun bağlantısı Şekil 1.11'de gösterilmiştir. Ölçme işlemi aşağıdaki gibi yapılır.

- PortA çıkış olarak yönlendirildikten sonra kondansatöre "0" bilgisi gönderilerek deşarj edilir.
- Kondansatörün deşarj süresinden sonra PortA giriş olarak yönlendirilir ve TMR0 sayıcısı çalıştırılır.
- PortA'daki gerilim V değerine ulaştığında TMR0 sayıcısı okunur.
- Şarj süresi direncin değeriyle doğru orantılıdır. Direnç büyüdükçe şarj süresi de büyüyeceğinden TMR0 registeri içerisinde daha büyük bir sayı okunur.



Şekil 1.11: ADC Uygulama devresi



Şekil 1.12 : ADC Akış diyagramı

- ADC uygulama programın kod satırları aşağıdaki gibi olmalıdır.

```
LIST      P=16F84
INCLUDE  "P16F84.INC"
ORG      H'00'
GOTO     BASLA
ORG      H'04'
GOTO     KESME
```

Portları Ayarla

```
CLRF     PORTB           ;PORTB'yi temizle
BSF      STATUS,5       ;BANK1'e geç
CLRF     TRISB          ;PORTB çıkış
CLRF     TRISA          ;PORTA çıkış
BCF      STATUS,5       ;BANK0'a geç
MOVLW   B'10100000'     ;TMR0 registerini kur
MOVWF   INTCON
BSF      STATUS,5       ;BANK1'e geç
MOVLW   B'11010001'     ;TMR0 aktif, dâhili komut saykılı
MOVLW   OPTION_REG
BCF      STATUS,5       ;BANK0 12
BCF      PORTA,0        ;Kondansatörü deşarj et
CLRF     TMR0           ;TMR0 zamanlayıcı başlat
DON
BTFS    TMR0,7         ;deşarj bitti mi?
GOTO    DON            ;Hayır bekle
BSF      STATUS,5       ;BANK1
BSF      TRISA,0        ;PORTA 0 giriş
BCF      STATUS,5       ;BANK0
CLRF     TMR0           ;TMR0 yeniden başlat
DON2
BTFS    PORTA,0        ;0.bit "1" mi?
GOTO    DON2          ;Hayır geri dön
MOVF    TMR0,W         ;Evet, TMR0'ı oku
MOVWF   PORTB          ;Kondansatör dolma süresi göster
BCF      INTCON,5
DONGU
GOTO    DONGU
KESME
BCF      INTCON,5       ;TMR0 kesmesini iptal et
MOVLW   H'AA'
MOVWF   PORTB          ;TMR0 dolma süresini göster
BEKLE
GOTO    BEKLE
END
```

;Programın başlangıcı

```
BASLA
BSF      PORTB, 1      ;Çıkışı "1" yani 5V yap
CALL    GECIKME1      ;Darbe genişliği %40 süresini bekle
BCF     PORTB,1      ;Çıkışı "0" yani 0V yap
CALL    GECIKME2      ;%60 süresince 0V olarak bekle
GOTO    BASLA         ;BASLA etiketine git
GECIKME1
MOVLW   H'66'         ;W<--H'66'
MOVWF   SAYAC1        ;SAYAC1 <--W
D1
DECFSZ  SAYAC1,F      ;Sayac1 bir azalt ve sıfır mı? kontrol et
GOTO    D1            ;Hayır D1'e git
RETURN
GECIKME2
MOVLW   H'9A'         ;W<--H'9A'
MOVWF   SAYAC2        ;SAYAC2 <--WD2 13
DECFSZ  SAYAC2,F      ;Sayac2 bir azalt ve sıfır mı? kontrol et
GOTO    D2            ;Hayır D2'e git
RETURN
END
```

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarını takip edip mikrodenetleyici ve DAC entegresi kullanarak testere dışı gerilimi üretiniz.

İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Programın akış diyagramını çiziniz.➤ Programlama dilinin yazım kurallarına dikkat ederek assembler komutlara göre programınızı yazınız.➤ Programınızı Pic 16F84A için derleyiniz.➤ Derleme sonucunda oluşan Hex dosyasını uygun bir programlayıcı ile mikrodenetleyiciye yazdırınız.➤ Programı test edip varsa hatalarını düzeltiniz.	<ul style="list-style-type: none">➤ Akış diyagram sembollerini amaca uygun kullanınız.➤ Programda kullanıldığınız mikrodenetleyici komutlarının sayı ve karakterlerinin yazılışına dikkat ediniz. Programın tamamını ya büyük ya da küçük harfle yazmak yazım hatalarını azaltacaktır.➤ Besleme gerilimlerinde simetrik 12 V ve asimetrik 5V kullanılmasına dikkat ediniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Mikroişlemci ile analog mantıkla çalışan çeşitli cihazlar kontrol edilirken hangi devrelere kesinlikle ihtiyaç vardır?
 - A) DAC devreleri
 - B) Filtre devreleri
 - C) Dijital göstergeler
 - D) Gerilim / frekans çeviriciler
 - E) ADC devreleri
2. Dijital değerlerde ağırlığı en düşük bite ne ad verilir?
 - A) Önemli bit
 - B) LSB
 - C) EDB
 - D) MSB
 - E) KSB
3. Çıkış bit sayısı 8 olan ADC, analog sinyalin maksimum değerini kaç seviyeye kullanılabilir?
 - A) 256
 - B) 8
 - C) 16
 - D) 128
 - E) 4
4. DAC devresi için yazılanlardan hangisi **yanlıştır**?
 - A) Dijital değerleri analog değerlere dönüştürür.
 - B) DAC devreleri bir analog devrenin çıkışında kullanılır.
 - C) Bazı A/D dönüştürücü devrelerin de DAC kullanılır.
 - D) Ağırlık dirençli DAC çeşidinde bütün direnç değerleri aynıdır.
 - E) R-2R merdiven tipi D/A dönüştürücü de çıkış dalga şekli merdiven basamağı gibi artar.
5. Aşağıdakilerden hangisi yalnızca iki durumu olan bilgiye verilen addır?
 - A) Analog bilgi
 - B) Sayısal bilgi
 - C) İkilik sayısal bilgi
 - D) Logaritmik bilgi
 - E) Hexadecimal bilgi

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

6. () Dijital değerlerde ağırlığı en yüksek basamağa en anlamlı bit (MSB) denir.
7. () Gerçekleşen bir sonucun, beklenen teorik değerden farkına hata denir.
8. () Dijital değerler “1” ve “0” ile gösterilebilen değerlerdir.
9. () 3 bitlik DAC çıkışında 16 farklı gerilim seviyesi oluşabilir.
10. () Bütün mikrodenetleyiciler ADC sahiptir.

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

Mikrodenetleyicili uygulamalar yapabileceksiniz.

ARAŞTIRMA

- Mikrodenetleyici ile yapılmış devreleri ve programları araştırıp inceleyiniz.
- Assembler programlama dilinden başka mikrodenetleyici programı yazmak için kullanılan dilleri araştırıp bu dillerin avantaj ve dezavantajlarını tartışınız.

2. MİKRODENETLEYİCİ UYGULAMALARI

2.1. Led Gösteri

Uygulama 1:

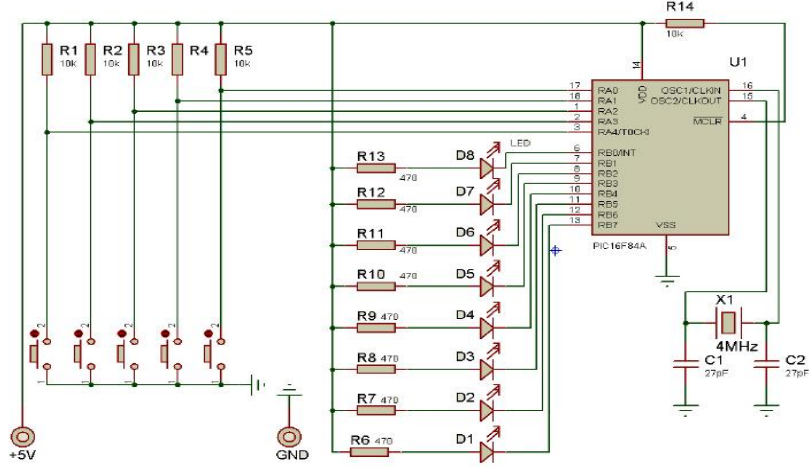
Açıklama

Bu devre basılan butona bağlı olarak ledlerin beş farklı şekilde yanmasını sağlar. Basılan butona bağlı olarak programda yazmış olduğumuz (yanma sırası) paternleri port b'ye yükler.

Malzeme Listesi

- Güç kaynağı
- Deney bordu
- 16F84A (1 adet)
- 4MHz kristal (1 adet)
- 27pF kondansatör (2 adet)
- Led (8 adet)
- Buton (5 adet)
- 10 K Ohm (6 adet)
- 470 Ohm (8 adet)

Devre Şeması



Şekil 2.1: Led gösteri devre şeması

Program

```

;*****
list p=16f84a
include p16f84a.inc
;
__CONFIG_WDT_OFF & _XT_OSC & _PWRTE_ON & _CP_OFF
;Programlayıcı ayarları
;kristal osilatör, wdt kapalı, power reset açık, kod koruma kapalı
;***** Değişken Tanımlamaları *****
ra0 equ 00 ;RA0 bit
ra1 equ 01 ;RA1 bit
ra2 equ 02 ;RA2 bit
ra3 equ 03 ;RA3 bit
ra4 equ 04 ;RA4 bit
cnt500u equ 0c ;500 mikrosaniyelik sayıcının adresi
cnt1m equ 0d ;1 milisaniyelik sayıcının adresi
cnt100m equ 0e ;100 milisaniyelik sayıcının adresi
cnt500m equ 0f ;500 milisaniyelik sayıcının adresi
cnt1s equ 10 ;1 saniyelik sayıcının adresi
;***** paternlerin üretilmesi *****
; '1':Led sönmük '0':Led yanık
;***** Pattern 0 *****
p00 equ b'11111110'
p01 equ b'11111101'
p02 equ b'11111011'

```

p03 equ b'11110111'
p04 equ b'11101111'
p05 equ b'11011111'
p06 equ b'10111111'
p07 equ b'01111111'
;***** Pattern 1 *****
p10 equ b'01111111'
p11 equ b'10111111'
p12 equ b'11011111'
p13 equ b'11101111'
p14 equ b'11110111'
p15 equ b'11111011'
p16 equ b'11111101'
p17 equ b'11111110'
;***** Pattern 2 *****
p20 equ b'01111110'
p21 equ b'10111101'
p22 equ b'11011011'
p23 equ b'11100111'
p24 equ b'11011011'
p25 equ b'10111101'
p26 equ b'01111110'
;***** Pattern 3 *****
p30 equ b'11111110'
p31 equ b'11111101'
p32 equ b'11111010'
p33 equ b'11110101'
p34 equ b'11101010'
p35 equ b'11010101'
p36 equ b'10101010'
p37 equ b'01010101'
p38 equ b'10101011'
p39 equ b'01010111'
p3a equ b'10101111'
p3b equ b'01011111'
p3c equ b'10111111'
p3d equ b'01111111'
;***** Pattern 4 *****
p40 equ b'00000000'
p41 equ b'11111111'
p42 equ b'00000000'
p43 equ b'11111111'
p44 equ b'00000000'
p45 equ b'11111111'
p46 equ b'00000000'
p47 equ b'11111111'

```

p48 equ b'00000000'
;***** Program Başlangıcı *****
org 0 ;Reset vektörü
goto init
org 4 ;Kesme Vectörü
goto init
;*** Bu kısımda giriş ve çıkışlar ayarlanır ***
; A portu giriş B portu Çıkış olarak ayarlandı
; Ledler sönmük

org 5
init bsf STATUS,RP0 ;Bank 1 e geç
movlw h'ff' ;registere H'FF' yükle
movwf TRISA ;PORTA yı giriş yap
clrf TRISB ;PORTB yi çıkış olarak ayarla
bcf STATUS,RP0 ;Bank 0'a geç
movlw h'ff' ;registere H'FF' yükle
movwf PORTB ;PORTB'yi h'FF', d'1111 1111' yap
;***** Basılı butonun tesbiti *****
keyscan btfs PORTA,ra0 ;RA0 Butona basılımı ?
call ptn0 ;Evet patern0 alt programına git
btfs PORTA,ra1 .
call ptn1 .
btfs PORTA,ra2 .
call ptn2 .
btfs PORTA,ra3 .
call ptn3 .
btfs PORTA,ra4 ;RA4 Butona basılımı ?
call ptn4 ;Evet patern4 alt programına git
goto keyscan ;keyscan'a git
;***** 0. pattern çıkış programı *****
;*call t100m satırı 100milisaniyelik gecikme sağlayan altprograma dallandırır*
;
ptn0 movlw p00 ;pater00 in ilk satırını kaydediciye yükle
movwf PORTB ;kaydedici içeriğini portb ye ( çıkışa) yükle
call t100m ;100 mslik gecikme alt programına git
movlw p01 ;sıradaki
movwf PORTB
call t100m
movlw p02
movwf PORTB
call t100m
movlw p03
movwf PORTB
call t100m
movlw p04
movwf PORTB

```

```

call t100m
movlw p05
movwf PORTB
call t100m
movlw p06
movwf PORTB
call t100m
movlw p07
movwf PORTB
call t100m
movlw h'ff'
movwf PORTB           ;PORTB ye h'FF' yükle ledleri söndür
call t100m           ;100 milisaniye bekle
return               ;keyscan'a git
;***** 1. Pattern çıkış programı *****
ptn1 movlw p10
movwf PORTB           ;p10 patternini çıkışa yükle
call t100m           ;100 milisaniye bekle
movlw p11             ;sıradaki
movwf PORTB
call t100m
movlw p12
movwf PORTB
call t100m
movlw p13
movwf PORTB
call t100m
movlw p14
movwf PORTB
call t100m
movlw p15
movwf PORTB
call t100m
movlw p16
movwf PORTB
call t100m
movlw p17
movwf PORTB           ;p17 patternini çıkışa yükle
call t100m           ;100 milisaniye bekle
movlw h'ff'           ;PORTB ye h'FF' yükle
movwf PORTB
call t100m
return               ;keyscan'a git
;***** 2. Pattern çıkış programı *****
ptn2 movlw p20
movwf PORTB           ;p20 patternini çıkışa yükle

```

```

call t100m                ;100 milisaniye bekle
movlw p21                 ;sıradaki
movwf PORTB
call t100m
movlw p22
movwf PORTB
call t100m
movlw p23
movwf PORTB
call t100m
movlw p24
movwf PORTB
call t100m
movlw p25
movwf PORTB
call t100m
movlw p26
movwf PORTB                ;p26 patternini çıkışa yükle
call t100m                ;100 milisaniye bekle
movlw h'ff'               ;PORTB ye h'FF' yükle
movwf PORTB
call t100m
return                    ;keyscan'a git
;***** 3. Pattern çıkış programı *****
p30 movlw p30
movwf PORTB                ;p30 patternini çıkışa yükle
call t100m                ;100 milisaniye bekle
movlw p31                 ;sıradaki
movwf PORTB
call t100m
movlw p32
movwf PORTB
call t100m
movlw p33
movwf PORTB
call t100m
movlw p34
movwf PORTB
call t100m
movlw p35
movwf PORTB
call t100m
movlw p36
movwf PORTB
call t100m
movlw p37

```



```

movwf PORTB
call t100m
movlw p38
movwf PORTB
call t100m
movlw p39
movwf PORTB
call t100m
movlw p3a
movwf PORTB
call t100m
movlw p3b
movwf PORTB
call t100m
movlw p3c
movwf PORTB
call t100m
movlw p3d
movwf PORTB
call t100m
movlw h'ff'           ;PORTB ye h'FF' yükle
movwf PORTB
call t100m           ;100 milisaniye bekle
return ;keyscan'a git
;***** 4. Pattern çıkış programı *****
ptn4 movlw p40
movwf PORTB         ;p40 patternini çıkışa yükle
call t1s           ;1 saniyelik gecikme alt programına git
movlw p41          ;sıradaki
movwf PORTB
call t1s
movlw p42
movwf PORTB
call t1s
movlw p43
movwf PORTB
call t1s
movlw p44
movwf PORTB
call t1s
movlw p45
movwf PORTB
call t1s
movlw p46
movwf PORTB
call t1s

```

```

movlw p47
movwf PORTB
call t1s
movlw p48
movwf PORTB
call t1s
movlw h'ff'
movwf PORTB
call t1s
return ;keyscan'a git
;;
*****Gecikme alt programları*****
; 1 ve 500 mili saniyelik gecikmeler, 100 ms ve 1 saniyelik gecikmeleri
;sağlayabilmek için oluşturulmuşlardır.
;bunlar kullanılmadan da döngü sayısı artırılarak istenen gecikme sağlanabilir.
;*****1 milisaniyelik gecikme alt programı *****
t1m movlw d'2 ' ;(1)
movwf cnt1m ;(1) cnt1 ye d'2' yükle
tm1lp1 movlw d'249' ;(1)*2
movwf cnt500u ;(1)*2 cnt2 ye d'249' yükle
tm1lp2 nop ;(1)*249*2 bekle
nop ;(1)*249*2 bekle
decfsz cnt500u,f ;(1)*249*2 cnt500u nin değerinden 1 çıkar cnt500 0 mı?
goto tm1lp2 ;(2)*248*2 hayırsa devam
decfsz cnt1m,f ;(1)*2 cnt1m 'nin değerinden 1 çıkart 0 mı ?
goto tm1lp1 ;(2) hayırsa devam
return ;(2) cnt1m = 0 geldiği satırın altına dön
;Toplam 2501*0.4usec=1 milisaniye
; iç içe iki döngü kuruldu
;*****100 milisaniyelik gecikme alt programı*****
t100m movlw d'100'
movwf cnt100m ;cnt100m 'e d'100' yükle
tm2lp call t1m ;1 milisaniyelik gecikme sağlayan alt programa git
decfsz cnt100m,f ;cnt100m - 1 = 0 ?
goto tm2lp ;hayırsa devam et
return ;cnt100m = 0 sa geldiği satırın altına dön
;*****500 milisaniyelik gecikme alt programı*****
t500m movlw d'5'
movwf cnt500m ;cnt500m 'e d'5' yükle
tm3lp call t100m ;100 milisaniyelik gecikme sağlayan alt programa git
decfsz cnt500m,f ;cnt500m - 1 = 0 ?
goto tm3lp ;hayırsa devam et
return ;cnt500m = 0 sa geldiği satırın altına dön
;*****1 saniyelik gecikme alt programı *****
t1s movlw d'2'
movwf cnt1s ;cnt1s 'ye d'2' yükle

```

```

tm4lp call t500m           ;500 milisaniyelik gecikme sađlayan alt programa git
decfsz cnt1s,f           ;cnt1s - 1 = 0 ?
goto tm4lp               ;hayırsa devam et
return                   ;cnt1s = 0 sa geldiđi satırın altına dön
end

```

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"> ➤ Şekil 2.1’deki devreyi kurunuz. ➤ Devreyi öğretmeninize kontrol ettiriniz. ➤ Yukarıdaki programı MPLAB IDE editöründe yazarak “ led .asm” olarak kaydediniz. ➤ Dosyayı Pic 16F84A için derleyiniz. ➤ Derlenen “ led.hex” isimli dosyayı herhangi bir yazdırma programı (ICProg vb.) ve uygun bir programlayıcıyla PIC’e yazdırınız. ➤ Devreye enerji (+5 V) veriniz. ➤ 1. butona basınız ve ledlerin yanmasını gözlemleyiniz. ➤ Diğer butonlara da tek tek basarak devrenin çalışmasını gözlemleyiniz. ➤ Program içerisinde paternleri deđiştirip işlem basamaklarını tekrarlayınız. ➤ Farklı yükler çalıştırmak için devre çıkışlarında nasıl deđişiklikler yapılabilir? Tartışınız. 	<ul style="list-style-type: none"> ➤ Besleme gerilimlerine dikkat ediniz. ➤ Programlayıcı yazılımda sigorta ➤ seçimleri yapmaya gerek yoktur. Bu seçimler program içerisinde tanımlanmıştır. ➤ Ledlerin parlaklığını seri direnç deđerlerini deđiştirerek artırabilirsiniz.

2.2. Zamanlayıcı

Uygulama 2:

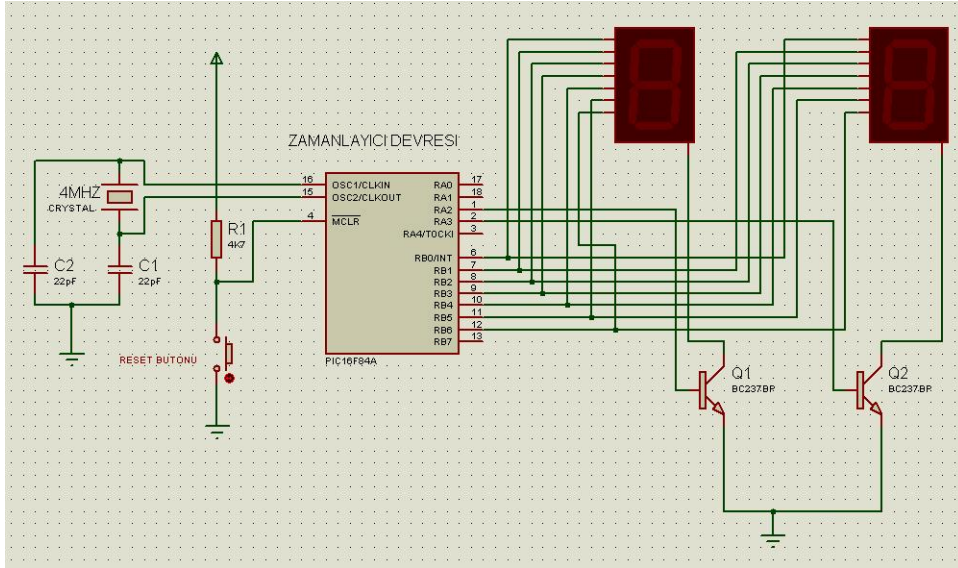
Açıklama

Bu devre mikrodenetleyici enerji verildiğinde, 1 saniye aralıklarla 99'a kadar sayarak 99 saniye gecikme yapabilir.

Malzeme Listesi

- Güç kaynağı
- Deney bordu
- 16F84A (1 adet)
- 4MHz kristal (1 adet)
- 22pF kondansatör (2 adet)
- Buton (1 adet)
- 4,7 K Ohm (1 adet)
- BC237B (2 adet)
- 7 Segment (ortak katot) Display (2 adet)

Devre Şeması



Şekil 2.2: Mikrodenetleyici zamanlayıcı devresi

Program

```
-----  
LIST P=16F84  
INCLUDE "P16F84.INC"  
ORG H'00'  
GOTO START  
ORG H'04'  
GOTO KESME  
-----  
DURUM EQU H'C8'  
SAYAC1 EQU H'C9'  
SAYAC2 EQU H'CA'  
SAYAC3 EQU H'CB'  
SAYAC4 EQU H'CC'  
DISP1 EQU H'CD'  
DISP2 EQU H'CE'  
DISP1A EQU H'B8'  
DISP2A EQU H'B9'  
-----  
START  
    BSF STATUS,5  
    CLRF TRISB  
    MOVLW H'03'  
    MOVWF TRISA  
    MOVLW H'01'  
    MOVWF OPTION_REG  
  
    BCF STATUS,5  
    CLRF DURUM  
    CLRF DISP1  
    CLRF DISP2  
    CLRF DISP1A  
    CLRF DISP2A  
    MOVLW D'50' ;D'50' --> W  
    MOVWF SAYAC1 ;W --> SAYAC1  
    MOVLW D'15' ;15  
    MOVWF SAYAC2  
    MOVLW D'52' ;52  
    MOVWF SAYAC3  
    MOVLW D'200' ;200  
    MOVWF SAYAC4  
    MOVLW B'10100000'  
    MOVWF INTCON  
    MOVLW D'55'  
    MOVWF TMR0  
    CLRF PORTB
```

BASLA

```
MOVLW    D'10'  
XORWF    DISP2A,W  
BTFSC STATUS,2      ;Z=?0  
GOTO A1      ;Z=!0 (W==DISP2)  
MOVF DISP2A,W      ;Z==0 (W!=DISP2),DISP2 --> W  
CALL CEVRIM_TAB    ;CEVRIM_TAB CAGIR  
INCF DISP2A      ;DISP2A=DISP2A+1  
MOVWF    DISP2      ;W --> DISP2  
GOTO GECIKME      ;GECIKME YE GIT
```

A1

```
CLRF DISP2A      ;DISP2A SIL  
MOVF DISP2A,W  
CALL CEVRIM_TAB  
MOVWF    DISP2  
INCF DISP2A  
INCF DISP1A      ;DISP1=DISP1+1  
MOVLW    D'10'  
XORWF    DISP1A,W  
BTFSC STATUS,2      ;Z=?0  
GOTO A2      ;Z=!0 (W==DISP1)  
MOVF DISP1A,W      ;DISP1 --> W  
CALL CEVRIM_TAB    ;CEVRIM_TAB CAGIR  
MOVWF    DISP1      ;W --> DISP1  
GOTO GECIKME      ;GECIKME YE GIT
```

A2

```
CLRF DISP1A      ;DISP1 SIL  
MOVF DISP1A,W      ;DISP1 --> W  
CALL CEVRIM_TAB    ;CEVRIM_TAB CAGIR  
MOVWF    DISP1      ;W --> DISP1  
GOTO GECIKME      ;GECIKME YE GIT
```

GECIKME

```
DECFSZ    SAYAC2,F      ;SAYAC2=SAYAC2-1, SAYAC2=?0  
GOTO B1      ;HAYIR SAYAC2=!0  
GOTO EXIT    ;EVET SAYAC2==0, EXIT E GIT
```

B1

```
DECFSZ    SAYAC3,F      ;SAYAC3=SAYAC3-1, SAYAC3=?0  
GOTO B2  
MOVLW    D'52'      ; D'52' --> W  
MOVWF    SAYAC3      ; W --> SAYAC3  
GOTO GECIKME      ;EVET SAYAC2==0, GIT GECIKME E
```

B2

```
DECFSZ    SAYAC4,F      ;SAYAC4=SAYAC4-1,SAYAC4=?0  
GOTO B2      ;HAYIR SAYAC4=!0  
MOVLW    D'200'      ; D'200' --> W  
MOVWF    SAYAC4      ; W --> SAYAC4
```

```

EXIT      GOTO B1                ;EVET SAYAC4==0, GIT DONGU2 E
          MOVLW      D'15'       ; D'15' --> W
          MOVWF     SAYAC2       ; W --> SAYAC2
          GOTO BASLA            ;BASLA YA DON

```

```

;-----
CEVRIM_TAB

```

```

          ADDWF     PCL,F
          RETLW     B'00111111'
          RETLW     B'00000110'
          RETLW     B'01011011'
          RETLW     B'01001111'
          RETLW     B'01100110'
          RETLW     B'01101101'
          RETLW     B'01111101'
          RETLW     B'00000111'
          RETLW     B'01111111'
          RETLW     B'01101111'

```

```

;-----
KESME

```

```

          BCF      INTCON,T0IF
          MOVLW     H'00'
          XORWF     DURUM,W
          BTFSC    STATUS,2      ;Z=?0
          GOTO C1                ;Z=!0 (W==DURUM)
          MOVLW     B'00001000'  ;Z==0 (W=!DURUM)
          MOVWF     PORTA
          MOVF      DISP1,W
          MOVWF     PORTB
          CALL     DON
          MOVLW     D'55'
          MOVWF     TMR0
          CLRF     PORTA
          MOVLW     H'00'
          MOVWF     DURUM
          RETFIE

```

```

C1

```

```

          MOVLW     B'00000100'
          MOVWF     PORTA
          MOVF      DISP2,W
          MOVWF     PORTB
          CALL     DON
          MOVLW     D'55'
          MOVWF     TMR0
          CLRF     PORTA
          MOVLW     H'01'

```

```

MOVWF    DURUM
RETFIE

;-----
DON
    DECFSZ    SAYAC1,1    ;SAYAC1=SAYAC1-1, SAYAC1=?0
    GOTO D1    ;SAYAC1=!0, GIT D1 E
    MOVLW    D'50'    ;D'50' --> W
    MOVWF    SAYAC1    ;W --> SAYAC1
    RETURN

D1
    NOP
    NOP
    NOP
    NOP
    NOP
    GOTO DON

;-----
END

```

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"> ➤ Şekil 2.2'deki devreyi kurunuz. ➤ Devreyi öğretmeninize kontrol ettiriniz. ➤ Yukarıdaki programı MPLAB IDE editöründe yazarak “ zamanlayıcı .asm “ olarak kaydediniz. ➤ Dosyayı Pic 16F84A için derleyiniz. ➤ Derlenen “ zamanlayıcı.hex” isimli dosyayı herhangi bir yazdırma programı (ICProg vb.) ve uygun bir programlayıcıyla PIC'e yazdırınız. ➤ Devreye enerji (+5 V) veriniz. ➤ Zamanlayıcının doğruluğunu kronometre ile kontrol ediniz. Olası farkın sebeplerini açıklayınız. ➤ Devrenin çalışmasını gözlemleyiniz. 	<ul style="list-style-type: none"> ➤ Besleme gerilimlerine dikkat ediniz. ➤ Displayı mutlaka ortak katot olarak bağlayınız. ➤ Programlayıcı yazılımda sigorta seçimleri yapmaya gerek yoktur. Bu seçimler program içerisinde tanımlanmıştır.

2.3. Mikrodenetleyici Bilgisayar Haberleşmesi

Uygulama 3:

Açıklama

Devre seri iletişim yöntemini kullanarak RS232 portu üzerinden bilgisayara asenkron formatta metin bilgisi gönderir. Mikrodenetleyiciden gelen bilgi Windows'ta hyper terminal programıyla ya da portlardan bilgi okuyan herhangi bir programla alınabilir.

Seri iletişim; çok kullanılan bir iletişim yoludur. Bilgisayarlar için modem, mouse vb. bağlantısı, endüstriyel anlamda cihazların haberleşmesi, uydu alıcılarının yazılımlarının güncelleştirilmesi ya da TV kumandaları gibi cihazlarda kullanılır. Kullanılan hat sayısının az olması bir avantajdır. Tek ya da çift yönlü kullanılabilir iletişim hızı düşüktür. Seri iletişimde birçok standart geliştirilmiştir. Bunlar içerisinde en çok kullanılanı **rs232c** standartıdır. Senkron ve asenkron seri iletişim vardır. Asenkron iletişimde alıcı ve vericinin bazı özellikleri uyumlu olmalıdır ve tek hat üzerinden iletişim yapılmalıdır. TV kumandası içerisinde start, parity ve stop bitleri bulunabilir. 7 bitlik bir veri 10 bitle gönderilir. Senkron iletişimde ise veriler clock darbeleriyle iletilir (Alıcı ve vericinin eş zamanlı çalışması için).

UART bilgisayarlarda kullanılan seri iletişim protokolüdür. Birçok mikrodenetleyici donanımsal **UART** portuna sahiptir.

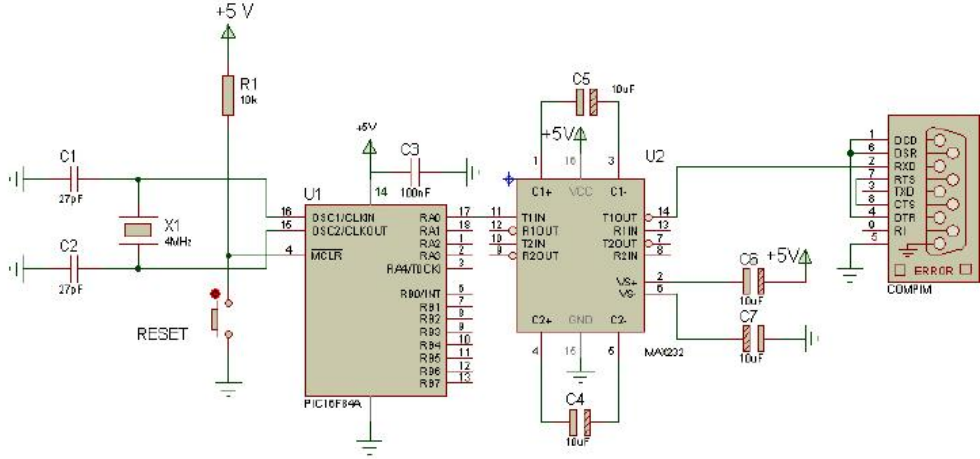
RS232 portuna mikrodenetleyici direnç üzerinden de bağlanabilmesine karşın hem portu korumak hem de gerilim seviyelerindeki farkın zararlarını engellemek için bir uygunlaştırıcı devre kullanmak gerekir. Bu devrelerden birisi de max232 entegresidir ve devrede kullanılmıştır. Max232 entegresi portların gerilim seviyelerini uygunlaştıran ve iki yönlü bilgi aktarımını sağlayabilen bir entegre devredir.

Programda uygun değişiklik yapılarak boşluklar dâhil 200 karakter tek seferde gönderilebilir.

Malzeme Listesi

- Güç kaynağı
- Bilgisayar
- Deney bordu
- Dişi rs232 soket (DB9 dişi soket) (1 adet)
- Max232 (1 adet)
- 16F84 (1 adet)
- 10K (1 adet)
- 27pF (2 adet)
- 10uF (4 adet)
- 100nF (1 adet)
- Kristal 4MHz (1 adet)
- Buton (1 adet)

Devre Şeması



Şekil 2.3: Mikrodenetleyici bilgisayar haberleşmesi

Program

```
list p=16F84A
include <P16F84A.INC>
;*****
#define seri_out PORTA,0
bt equ h'16'
vericnt equ h'13'
sdata equ h'14'
temp equ h'15'
;*****
basla
call kurulum
call mesaj
tekrar
goto tekrar
;
kurulum
bsf STATUS,RP0
clrf PORTA
bcf STATUS,RP0
clrf vericnt
return
;
mesaj
m_devam
movf vericnt,0
call mesaj_verisi
```

```
iorlw 0
bz m_son
movwf sdata
call data_gonder
incf vericnt,1
goto m_devam
return
m_son
mesaj_verisi
addwf PCL,1
msj1 dt "iyi gunler",0
return
baud_time
movlw h'86'
movwf bt
next1
decfsz bt,F
goto next1
return
data_gonder
bcf seri_out
call baud_time
movlw 8
movwf temp
serikomut
rrf sdata,1
btfss STATUS,C
goto $+3
bsf seri_out
goto $+2
bcf seri_out
call baud_time
decfsz temp
goto serikomut
bsf seri_out
call baud_time
call baud_time
return
end
```

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"> ➤ Şekil 2.3'teki devreyi kurunuz. ➤ Devreyi öğretmeninize kontrol ettiriniz. ➤ Yukarıdaki programı MPLAB IDE editöründe yazarak " pic_pc .asm " olarak kaydediniz. ➤ Dosyayı 16F84A için derleyiniz. ➤ Derlenen " pic_pc.hex" isimli dosyayı herhangi bir yazdırma programı (ICProg vb.) ve uygun bir programlayıcıyla PIC'e yazdırınız. ➤ Devreyi bilgisayar seri portuna bağlayınız. ➤ Yukarıda verilen bilgisayar ayarlarını yapınız. ➤ Hyper terminal programında oluşturduğunuz bağlantıyı açınız. ➤ Devreye enerji (+5 V) veriniz. ➤ Devreyi butonla resetleyiniz. 	<ul style="list-style-type: none"> ➤ Besleme gerilimlerine dikkat ediniz. ➤ Sigorta seçimlerinin program içerisinde yapılabileceğini görmüştük. Bu uygulamada programlama esnasında sigorta seçimi yapılacaktır. ➤ Sigorta seçimi ➤ Watchdog timer Disabled ➤ Oscillator XT ➤ Power-up timer Enabled ➤ Code Not protected ➤ Genelde kullanılan port com1 olmakla beraber, bağlantı için kullandığınız portu com2 vb. Hyper terminalde seçmeniz gerekebilir. ➤ Devreyi sık resetlemeyiniz. Bilgi aktarımı etkilenebilir.

2.4. Bilgisayar Ayarları

Bilgisayar ve mikrodenetleyici arasında haberleşme sağlanırken, alıcı program olarak Windows Hyper Terminal programı kullanılacaktır. Aşağıdaki ayarların yapılması gereklidir.

- Windows işletim sistemi içindeki, Hyper terminal programını açınız.
- Yeni bir bağlantı oluşturunuz.
- Bağlantı özelliklerini ayarlayınız.
- Ayarlar, COM1 seri portuna bağlı cihaz 2400 baud 8 bit veri 1 bit stop ve eşlik yok olarak ayarlanacaktır.

ÖLÇME VE DEĞERLENDİRME

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

1. () Assembler dilinde program yazarken sigorta ayarları mutlaka program içerisinde belirtilmelidir.
2. () Bilgisayarlarda portlara sadece Hyper terminal programıyla ulaşılabilir.
3. () PIC16F84 için yazılmış bir program hex'i PIC16F877 için de kullanılabilir.
4. () Assembler dilinde program yazarken Türkçe karakterler kullanılmaz.
5. () Assembler dilinde açıklama satırı için “, “ kullanılır.
6. () PIC 16F84'ü resetlemek için 4 nolu pin kullanılır.
7. () Bilgisayar ve mikrodenetleyici haberleşirken herhangi bir aracı program kullanılmaz.
8. () Mikrodenetleyiciler bilgisayar ile haberleşme yaparken seri portu kullanabilir.
9. () Mikrodenetleyicilere yüklenen programlar, mikrodenetleyiciye göre farklı zamanlarda çalışabilir.

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

10. Mikrodenetleyicilerle buton kullanırken aşağıdakilerden hangisi yapılmamalıdır?
 - A) Program içerisinde butonun durumuna gecikmeli bakılmalıdır.
 - B) Butona paralel kondansatör bağlanmalıdır.
 - C) Buton kablo ve yolu mümkün olduğu kadar kısa olmalıdır.
 - D) Gerilim sınırlayıcı direnç kullanılmalıdır.
 - E) Akım sınırlayıcı entegre kullanılmamalıdır.
11. Zamanlayıcının hassasiyetine etki eden en önemli faktör aşağıdakilerden hangisidir?
 - A) Ortam sıcaklığı
 - B) Kablo uzunlukları
 - C) Displaylerin taranması
 - D) Kristal değeri
 - E) Direnç değerleri

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**, kazanamadıklarınız için **Hayır** kutucuklarına (X) işareti koyarak öğrendiklerinizi kontrol ediniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Çeviricilerin kullanım alanlarına örnekler verebilir misiniz?		
2. Analog sinyali tanımlayabilir misiniz?		
3. Sayısal bilgiyi tanımlayabilir misiniz?		
4. Assembler dilinde yazılmış bir programı derleyebilir misiniz?		
5. Bir hex dosyasını mikrodenetleyiciye yükleyebilir misiniz?		
6. Herhangi bir DAC entegresi kullanarak D/A çevirimi yapabilir misiniz?		
7. Herhangi bir ADC entegresi kullanarak A/D çevirimi yapabilir misiniz?		
8. Mikrodenetleyici ile ADC uygulamasını gerçekleştirdiniz mi?		
9. Mikrodenetleyici ile DAC uygulamasını gerçekleştirdiniz mi?		
10. Mikrodenetleyici ile led gösteri uygulamasını gerçekleştirdiniz mi?		
11. Mikrodenetleyici ile zamanlayıcı uygulamasını gerçekleştirdiniz mi?		
12. Mikrodenetleyici ile mikrodenetleyici bilgisayar iletişim uygulamasını gerçekleştirdiniz mi?		
13. Mikrodenetleyici kullanarak bir projeyi tüm aşamalarıyla tasarlayıp 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	A
2	B
3	A
4	D
5	C
6	DOĞRU
7	DOĞRU
8	DOĞRU
9	YANLIŞ
10	YANLIŞ

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	YANLIŞ
2	YANLIŞ
3	YANLIŞ
4	DOĞRU
5	YANLIŞ
6	DOĞRU
7	YANLIŞ
8	DOĞRU
9	DOĞRU
10	B
11	A

KAYNAKÇA

- YAĞIMLI Mustafa, Fevzi AKAR, Dijital Elektronik, Beta Yayınları, İstanbul, 1999.
- <http://elektrik.gen.tr/teknik-i%C3%A7erik/dac-ve-adc> (02.08.2013/15:24).
- http://www.elektronikhobi.com/dokuman.asp?id=105&user_enc= (03.08.2013/00:08).
- MEGEP, Elektrik-Elektronik Teknolojisi, ADC-DAC Devreleri, Ankara, 2012.
- MEGEP, Elektrik-Elektronik Teknolojisi, Mikrodenetleyiciyle Analog İşlemler, Ankara, 2012.
- MEGEP, Bilişim Teknolojileri, Mikrodenetleyiciler 3, Ankara, 2008.