

8051 AİLESİ MİKRODENETLEYİCİ EĞİTİM SETİNİN TASARIMI VE GERÇEKLEŞTİRİLMESİ

Ömer Faruk BAY, Salih GÖRGÜNOĞLU
Gazi Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü
06500 Teknikokullar, ANKARA

ÖZET

Günümüzde mikrodeneleyici adı verilen işlemciler yaygın olarak bir çok alanda kullanılmaktadır. Çok sayıda analog ve sayısal tümleşik elemanlar kullanılarak gerçekleştirilebilecek devre tasarımları tek bir mikrodeneleyici ile yapılabilir. Mikrodeneleyici kullanılarak yapılan devreler daha az yer kaplamaktadır. Harici donanım elemanları ile yapılan işler mikrodeneleyici kontrol yazılımı ile yapılabilir. Böylece donanımda değişiklik yapmadan sadece yazılımı değiştirerek aynı cihaz üzerinde farklı işlemler yapma imkanı sunulmaktadır. Bu yüzden mikrodeneleyicilerin öğrenilmesi ve öğretilmesi lisans eğitiminde büyük önem arz etmektedir. Bu çalışmada mikrodeneleyicilerin öğrenilmesini kolaylaştırmak ve deneysel çalışmalarda kullanabilmek için genel amaçlı bir mikrodeneleyici eğitim seti tasarlanmış ve gerçekleştirilmiştir. Deney setinin kullanımı ile ilgili örnek yazılımlar verilmiş, setin kullanımına yönelik bilgisayar ortamında bir arayüz yazılımı da yapılmıştır. Böylece mikrodeneleyicinin ve eğitim seti içinde yer alan modüllerin daha iyi anlaşılması sağlanmıştır.

Anahtar Kelimeler : 8051 Mikrodeneleyici, RS232 Seri port, C++ Builder

DESIGN AND IMPLEMENTATION OF AN 8051 MICROCONTROLLER EDUCATIONAL SET

ABSTRACT

In recent years, the processors which is called microcontroller have been used in many areas. The implementation of circuit design by using a lot of integrated analog and digital element could be done using only one microcontroller. The design that uses microcontroller require less board area. Because many tasks which is implemented using extra hardware elements are performed by software. Thus, several processes can be implemented on a microcontroller by software without changing hardware parts. So, learning and using microcontrollers are very important concept for undergraduate education in electrical engineering and computer engineering. This paper presents an educational set specifically designed for teaching and learning the 8051 family microcontrollers. Control software and related information for using of educational set was given, and a computer interface program related to use different module in educational set was prepared. Thus, understanding the microcontroller and the modules in the educational set was made easy.

Keywords: 8051 microcontroller, RS232 serial port, C++ Builder

1. GİRİŞ

Mikrodeneleyiciler elektronik devre tasarımlarında denetleme elemanı olarak çok yaygın kullanıma sahip tümleşik elemanlardır. Bünyelerinde zamanlayıcı, sayıcı, giriş çıkış portları, analog sayısal çevirme ünitesi, RAM bellek, ROM bellek, PWM darbe üretici, seri haberleşme birimi, ALU vb. birimler bulundurlar. ROM bellek içinde kullanıcı tarafından yazılıp yerleştirilen kontrol programı sayesinde sözü edilen birimler çalıştırılır ve istenen işler yerine getirilir. Mikrodeneleyici, ALU sayesinde matematiksel işlemler yapabilme ve işlem sonuçlarını mantıksal olarak değerlendirip karar verme yeteneğine sahiptir (1-4). Bu sayede donanımda değişiklik yapmadan sadece mikrodeneleyicinin işleyeceği yazılımı değiştirerek, değişik işlerin yapılması sağlanabilmektedir. Bu ise tasarımcıya büyük bir kolaylık ve devre tasarımında esneklik sağlamaktadır. Ay-

rica donanım maliyeti de oldukça azalmakta, yer tasarrufu sağlanarak daha küçük boyutlu cihazların yapılmasına imkan sağlanmaktadır.

Günümüzde tümleşik elemanlar üreten birçok firma (Intel, Atmel, Michrochip, Texas Instrument, National Semiconductror vb.) değişik özelliklere sahip mikrodeneleyiciler üretmektedirler. Mikrodeneleyiciler birbirlerinden sahip oldukları üniteler(ADC, PWM, Zamanlayıcı, vb), Giriş/Çıkış bacak sayıları, çalışma hızları, bellek kullanım şekilleri açısından farklılıklar arz etmektedirler. Devre tasarımları yapılırken özellikle maliyet açısından amaca en uygun mikrodeneleyici bu özellikler göz önünde bulundurularak seçilmektedir.

Mikrodeneleyicilerin çok yaygın olarak kullanımı bu konunun lisans eğitiminde ağırlıklı olarak verilmesini gerektirmektedir. Bu dersin ve-

rilmesinde ise hem donanım hem de yazılım kısmının birlikte koordineli bir şekilde verilmesi büyük önem arz etmektedir. Bunun için mikrodenetleyicinin öğrenilmesini kolaylaştırmaya yönelik bir çok çalışma yapılmaktadır (5-11). Yapılan bu çalışmaların çoğu simülatör çalışması olduğundan öğrencinin bir bütün olarak mikrodenetleyici tabanlı bir sistemi tasarlayıp sonrada fiziksel olarak gerçekleştirmesi mümkün olmamaktadır. Yapılan çalışmaların bir kısmı ise ticari geliştirme bordları kullanılarak labaratuvarında yapılan deneysel çalışmaların verimli hale getirilmesine yöneliktir. Ticari geliştirme bordları ise gelişmekte olan ülkelerdeki okullar için pahalı seçimlerdir. Bu çalışmada ise bir mikrodenetleyicinin hem donanım hem de yazılım olarak öğrenilmesini kolaylaştırmaya yönelik modüler yapıda bir eğitim seti tasarlanmakta ve gerçekleştirilmektedir. Eğitim setinin maliyeti ticari geliştirme bordlarından oldukça düşük ve modüler yapıda tasarlandığı için de geliştirmeye açıktır.

8051 mikrodenetleyici seçilmesinin sebebi ise, bu mikrodenetleyici endüstriyel uygulamalarda çok yaygın olarak kullanılmakta ve basit bir mimariye sahip bulunmaktadır. 20 den fazla birbirinden bağımsız 8051 tabanlı eleman satan firma ve 12 den fazla bu mikrodenetleyici ile ilgili ders kitabı bulunmaktadır (6). Bu mikrodenetleyiciye ait temeller öğrenildikten sonra bunlar diğer mikrodenetleyicilere veya daha karmaşık mimariyle kolaylıkla aktarılabilir.

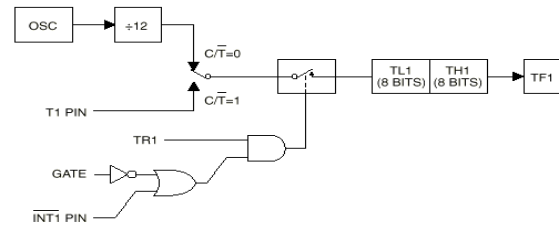
Bu makalede, eğitim ve öğretim için düşük maliyetli bir mikrodenetleyici eğitim setinin tasarımı ve gerçekleştirilmesi sunulmaktadır. 2. bölümde mikrodenetleyicinin yapısı anlatılmaktadır. 3. bölümde mikrodenetleyici eğitim setinin nasıl tasarlandığı ve gerçekleştirildiği detaylı bir şekilde sunulmaktadır. 4. bölümde bilgisayar seri haberleşme portunun nasıl kullanılacağı anlatılmaktadır. 5. bölümde ise mikrodenetleyici seti ile bilgisayarın haberleşmesini sağlayan bilgisayar arayüz yazılımı tanıtılmaktadır.

2. MİKRODENETLEYİCİNİN YAPISI

Eğitim setinin nasıl tasarlandığına geçmeden önce mikrodenetleyicinin bazı önemli özelliklerinin bilinmesi gerekmektedir. Bunlar kesme (Interrupt) kullanımı, zamanlayıcı kullanımı ve özel fonksiyon kaydedicileridir.

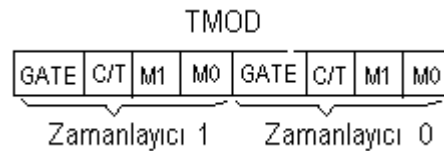
2.1. Zamanlayıcı ve Sayıcı yapıları

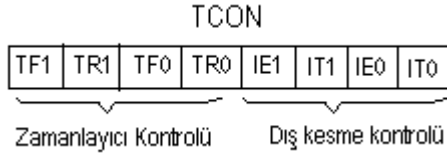
Zamanlayıcı/Sayıcılar iç veya dış kaynaklı bir zaman periyodunu ölçmek için veya olayları saydırmak için kullanılır. Girişe uygulanan bir darbenin genişliğinin veya frekansının ölçülmesi buna örnek olarak verilebilir. 8051 ürünlerinin çoğunda iki adet 16 bitlik zamanlayıcı (T0, T1) ve her zamanlayıcı için iki tane özel fonksiyon kaydedicisi (Special Function Register-SFR) vardır. Bunlar T0 için TH0 ve TL0, T1 için TH1 ve TL1 dir. Bu sekiz bitlik zamanlayıcılar birleştiklerinde 16 bitlik zamanlayıcıyı oluştururlar. Çalışma esnasında her giriş darbesinde bir veya iki SFR deki sayma değeri bir artırılır. Şekil 1'de zamanlayıcının yapısı görülmektedir.



Şekil-1. Zamanlayıcı/sayıcının yapısı

Zamanlayıcı/sayıcı zamanlayıcı durumunda iken makine periyodunu sayar. Makine periyodu saat frekansının 1/12 sidir. Sayıcı durumunda makine periyoduna bağlı olarak P3.4(T0) veya P3.5 (T1) dış giriş ucundan örnek alır. Giriş uçlarının 1 den 0'a geçişlerinde saklayıcının değeri 1 artar. Bir sonraki örnekleme için bir makine periyodunun geçmesi gerekir. Bu yüzden örnekleme frekansı osilatör frekansının 1/24 üne eşittir. Daha yüksek frekanslarda hatalı sayımlar oluşur. Zamanlayıcı sayıcı ayrıca sayma değeri taşıdığına bir kesme üretir. Bunun için ilgili kesme bayrağı yetkilendirilmesi gerekir. Zamanlayıcı/sayıcı TMOD ve TCON kaydedicileri programlanarak kullanılırlar. Şekil 2'de TMOD ve TCON Kaydedicileri ve işlev tanımlamaları görülmektedir.





Şekil -2. TMOD ve TCON kaydedicileri

TMOD ilk 4 biti Zamanlayıcı-0 ve ikinci 4 biti Zamanlayıcı-1 ayarlamak için kullanılır. GATE=1 olduğunda Zamanlayıcı x'in INTx dış kesmesinin 1 olması yeterlidir. C/T zamanlayıcı veya sayıcı seçme bitidir. Zamanlayıcı olarak çalışacaksa bu bit 0 yapılır. Bu durumda sistem saatini sayar. Eğer bu bit 1 yapılırsa harici Tx bacağından gelen darbeleri sayar.

M1 ve M0 bitleri 4 çalışma modundan birini seçer. Mod 1 seçildiğinde 16 bitlik sayma yapar yani TH ve TL nin mevcut değerlerinden başlayarak 65535(FFFF) değerine kadar sayar. Bu değere ulaştığında kesme üretir TFx bayrağını set eder. Zamanlayıcıyı çalıştırmak veya durdurmak için TCON kaydedicisindeki TRx bitlerinden yararlanır(4).

2.1.1 Zamanlayıcı ile zaman gecikmesi yapmak

Zamanlayıcı THx ve TLx kaydedicisinin birlikte oluşturduğu değerden itibaren 65535 değerine kadar sayar. Bundan yararlanarak istenen zaman gecikmesi sağlanabilir. Örnek olarak 50 ms'lik bir zaman gecikmesi istensin. Sistem saat frekansı 11.0592 MHz olsun bunun için şu sıra izlenir.

Sayıcı saat frekansı = Sistem saat frekansı / 12

$$= 11.0592 / 12$$

$$= 921.6 \text{ KHz}$$

Her sayma için geçen

zaman = 1 / sayıcı saat frekansı

$$= 1 / 921.6 \text{ KHz}$$

$$= 1.085 \mu\text{s}$$

Gecikme için sayma Sayısı

$$= \text{Gecikme zamanı} / \text{Her sayım için geçen zaman}$$

$$= 50 \text{ ms} / 1.085 \mu\text{s}$$

$$= 46080$$

THx ve TLx kaydedicilerine yüklenecek değer

Ön yükleme sayısı = 65536 - Gecikme için sayma sayısı

$$= 65536 - 46080$$

$$= 19456 = 4C00H$$

Dolayısı ile THx=4C ve TLx=00 değerleri onaltılık olarak yüklenip zamanlayıcı çalıştırıldığında 50 ms lik gecikme sağlanacaktır

2.2. Kesmelerin Kullanımı

Mikrodenetleyici özellikle giriş çıkış işlemlerinde bir porttan değer gelip gelmediğini sürekli kontrol etmek zorunda kalır. Bu ise sürekli olarak mikrodenetleyiciyi meşgul ederek diğer işleri aksatabilir. Bunu önlemek için kesmelerden yararlanır. Mikrodenetleyici bir işle meşgul iken kendi işini bırakarak çevre birimlerden gelen isteğe cevap vermesi ve daha sonra işine kaldığı yerden devam etmesine kesme (interrupt) denir. Çevre birimler isteklerini mikrodenetleyiciye kesme girişlerine sinyal göndererek veya bazı kaydedicilerdeki bitleri set ederek iletirler.

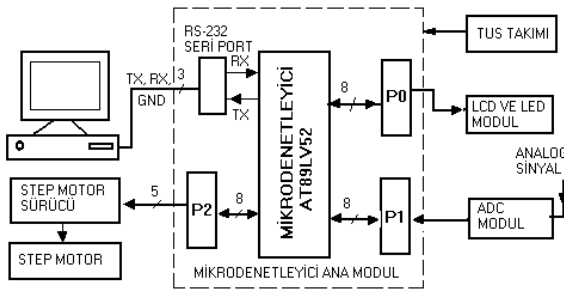
Tablo-1. 8051 kesme adresleri

Kesme	Kaynağı	Kesme sebebi	Kesme vektör adresi
Harici kesme 0	INT0 pin	Düşük düzey veya düşen kenar	0003H
Zamanlayıcı 0	TF0	FFFF den taşma	000BH
Harici kesme 1	INT1 pin	Düşük düzey veya düşen kenar	0013H
Zamanlayıcı 1	TF1	FFFF den taşma	001BH
Seri port	RI veya TI	Veri gönderiminin veya alımının tamamlanması	0023H

8051 in beş adet kesme üreten kaynağı vardır. Bunlardan iki tanesi harici diğerleri dahildir. Bir kesme geldiğinde işlemci derhal işini bırakıp gelen kesme kaynağına göre kesme vektör adresine dallanır. Burada kesme hizmeti için 8 baytlık bir yer ayrılmıştır. Eğer bu adres alanı hizmet görecektir alt programın yazılımı için yeterli ise alt program buraya yazılır. Eğer yeterli değilse kesme alt programına bu adrese konulan bir LJMP komutu ile gidilip RETI komutu ile dönlür. Mikrodenetleyicili sisteme güç verildiğinde veya RESET kesmesi üretildiğinde mikrodenetleyici 0000H adresine dallanarak buradaki ilk komutu işlemeye başlar. Tablo 1'de kesmeler ve kesme vektör adresleri görülmektedir(1). C derleyicisi kesme olaylarını kendisi yönetmektedir.

3. MİKRODENETLEYİCİ EĞİTİM SETİNİN TASARIMI

Mikrodenetleyici eğitim seti, Şekil 3'deki genel blok şemada görüldüğü gibi, mikrodenetleyici ana modülü, analog/sayısal çevirici modülü (ADC), LCD ve LED modülü, step motor sürme modülü, tuş takımı modülü olmak üzere beş modülden meydana gelmektedir. Bilgisayar ara yüz yazılımı C++ Builder görsel programlama dili kullanılarak, mikrodenetleyici modülleri kontrol yazılımları ise 8051 mikrodenetleyicisi için geliştirilen standart C derleyicisi kullanılarak gerçekleştirilmektedir.



Şekil-3. Eğitim setinin genel blok şeması

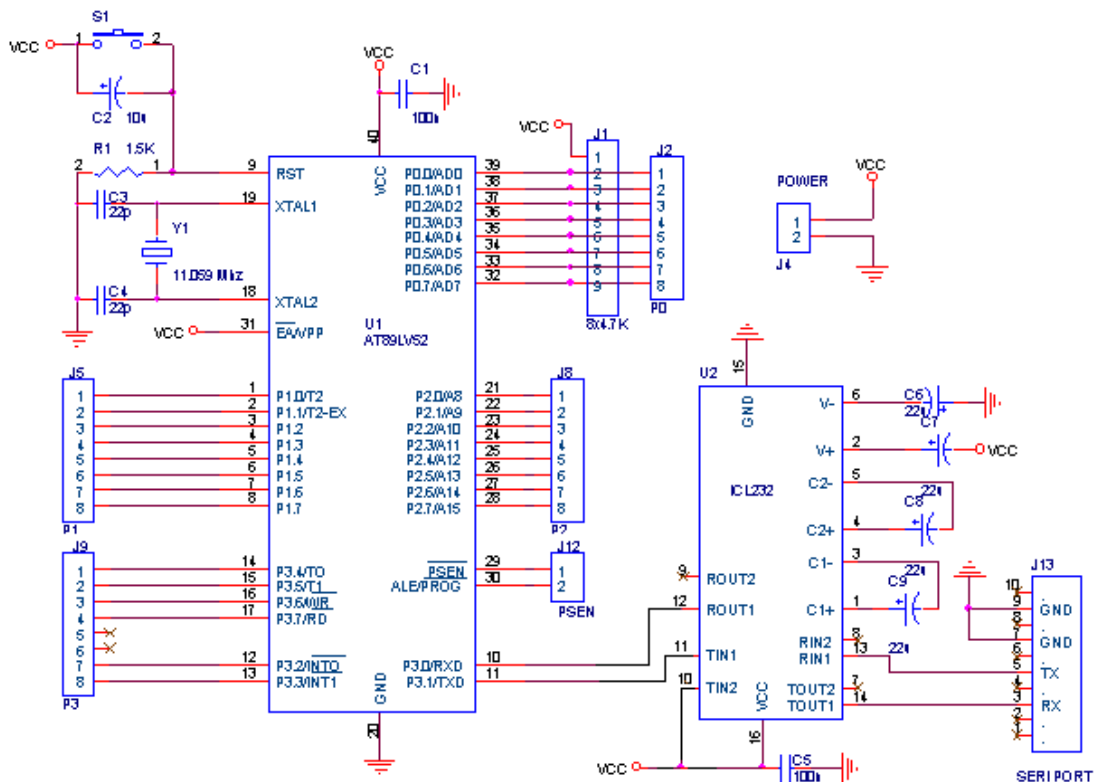
3.1 Ana kart modülü

Eğitim setinde yer alan anakart modülünde AT89LV52 mikrodenetleyici kullanılmıştır ancak 8051 uyumlu ve 40 bacaklı DIP şeklindeki diğer işlemcilerde kullanılabilir. Anakart üzerinde mikrodenetleyicinin tüm uçları kullanım esnekliği

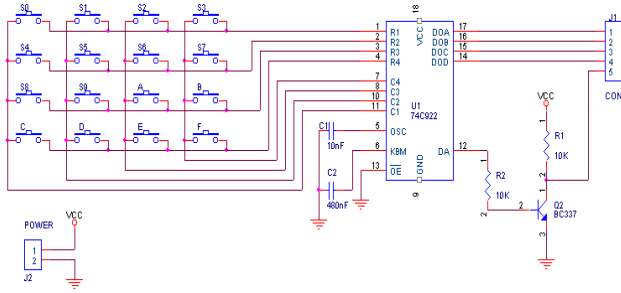
sağlamak için konnektörlerle dışarı çıkarılmıştır. Mikrodenetleyicinin saat sinyali sağlamak için 11 MHz.'lik kristal kullanılmıştır. Ayrıca anakart üzerinde seri haberleşme yapabilmek için seri haberleşme birimi (RS232) eklenmiştir. Bu modülde seri haberleşme için ICL232 tümleşik devresi kullanılmıştır. Şekil-4 de anakart modülünün devre şeması verilmektedir. Devredeki S1 anahatı mikrodenetleyiciyi resetlemek için kullanılmaktadır. Ayrıca mikrodenetleyicinin P0 portuna 4.7 k'lık pull up dirençleri, P0 portunun yapısı gereği konulmuştur. Diğer portlara pull up dirençleri koymaya gerek yoktur.

3.2. Tuş takımı Modülü

Şekil-5 de görülen tuş takımı devresinde 74C922 kod çözücü tümdevresi kullanılmıştır. Bu tümdevrenin tuşların bağlandığı satır ve sütun girişleri ile bir kesme çıkışı ve veri çıkış portları vardır. Herhangi bir tuşa basıldığında 74C922 tümdevresi basılan tuşu satır ve sütunları tarayarak bulmaktadır. Veri çıkış uçlarına tuşun onaltılık sayı kodunu yerleştirir ve tuş kodunun hazır olduğunu belirtmek için DA çıkış ucunu belli bir süre (yaklaşık 100ms) aktif yapar. Daha sonra tekrar mantıksal 0 değerine çevirir. 74C922 kod çözücü entegresinin DA çıkışının kullanımı ile kesme üretilerek tuş kodlarının işlemciye gönderilmesi sağlanır. Mikroişlemci kesme girişleri mantıksal 0 da aktif olduğundan bu tümdevrenin çıkışına bir transistör konularak deşillemeye işlemi yapılmıştır.



Şekil-4 Mikrodenetleyici ana kart modülü devre şeması



Şekil-5. Tuştakımı modülü devre Şeması

3.2.1 Mikrodenetleyici tuştakımı yazılımı

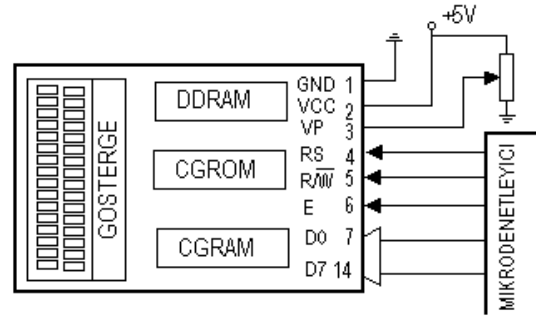
Aşağıda 8051 mikrodenetleyicisi için C derleyicisi ile yapılan tuştakımı devresi için örnek yazılım formatı verilmektedir. C derleyicisi kesmelere kendisi bir kesme numarası vererek kullanılmaktadır. INT0 harici kesmesinin numarası 0, Seri haberleşme kesme numarası 4 dür. Bu bilgiler kullanılan C derleyicisinin kullanım kılavuzundan öğrenilebilir(12).

```
#include<AT89X52.h> /* tanımlı register isimlerini kullanabilmek için gerekli header dosyası*/
#include<intrins.h> /*_nop_; komutunu kullanabilmek için gerekli header dosyası*/
Tuş takımını kullanabilmek için gerekli değişken tanımlamaları
sfr TUSPORTU=0x90 ; /* P0 portunun P2.0,P2.1,P2.2,P2.3 port uçları tus takimi data girişi*/
unsigned char idata tuskodu;
Bir tuşa basıldığında tuş takımı INT0 girişine bir sinyal gönderir. Bu sinyal kesme girişinden aşağıdaki alt program ile algılanır.
void EXT0(void) interrupt 0 /* 74c922 int0 kesme girişini kullanmaktadır*/
{
EX0=0; /*INT0 yetkisiz*/
TUSPORTU=0xff; /*tuspotunu giriş olarak ayarla*/
tuskodu=TUSPORTU; /*tuş kodunu oku*/
tuskodu=tuskodu & 0xf0; /*MSB 4 bite 74922 nin data hatları bağlı olduğundan LSB 4 bit maskelendi*/
switch(tuskodu) /*tus koduna göre işlem yap*/
{
case 0x00:/*Komutlar*/;break;
case 0x10: /*Komutlar*/;break;
case 0x20: /*Komutlar*/;break;
case 0x30: /*Komutlar*/;break;}
EX0=1; /*INT0 tekrar yetkili*/}
```

3.3. LCD ve LED Modülü

Devre tasarımında işlem sonuçlarını veya istenen mesajları görmek için yedi segment LED

gösterge elemanlar veya LCD göstergeler kullanılmaktadır. LED gösterge sadece sayısal değerler ve birkaç karakter gösterebilmektedir. Buna karşılık LCD göstergeler ile her türlü yazı ve sayısal değeri göstermek mümkündür. Kontrolleri yazılımla daha kolaydır. Bu yüzden uygulamalarda LCD göstergeler daha çok tercih edilirler. LCD ler çeşitli firmalar tarafından üretilmesine rağmen kontrolleri standartlaşmıştır. Tüm LCD göstergelerde yetki (Enable), oku yaz(R/W), ve kaydedici seç(RS) uçları ile veri giriş hatları vardır. Şekil 6'da LCD göstergenin blok şeması görülmektedir.



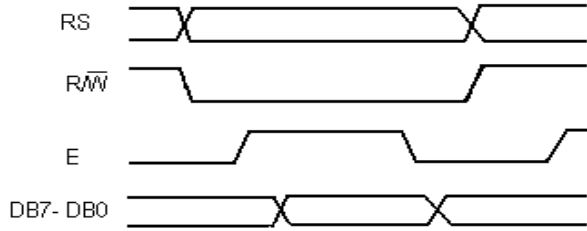
Şekil -6 LCD iç yapısı

LCD göstergeler karakter ve satır sayısı olarak çeşitli seçenekler sunarlar. LCD göstergeler karakterlerin kodlarını saklayabilmek için bir dahili RAM bulundurlar. Bu RAM'a Gösterge Veri RAM'ı (Display Data RAM) adı verilir. Bir satırında 16 karakteri olan 2 satırlık bir göstergelyi her bir satırında 40 karakteri olan 2 sanal satır olarak düşünmek mümkündür. Ancak bir anda 40 karakterden 16 sı görülebilmektedir. Kalan karakterler ancak kaydırma işleminden sonra görülebilirler. arzu edilen herhangi bir noktaya bir karakter yazdırılmak istenirse önce DDRAM adresi seçilmeli ve daha sonra karakter kodunun buraya yazılması gerekmektedir(2). LCD gösterge modülünde aynı zamanda 8 adet LED ışık da yerleştirilmiştir. Böylece LED ışıklarla da değişik uygulamaların yapılması sağlanmıştır.

3.3.1. LCD Kontrol İşlemleri

LCD göstergeye gönderilen veri ya bir komut kodu veya bir karakterdir. LCD göstergelerde ekranın temizlenmesi, imlecin başa alınması , DDRAM adresinin seçilmesi, resetleme gibi komut işlemleri vardır. Bunun için LCD ye gönderilen verinin komut mu yoksa ekranda gösterilecek bir karakter mi olduğu belirtilmelidir.

Şekil-7’de LCD göstergeye yazma işlemine ait zamanlama diyagramı görülmektedir. RS (Register Select) ucu düşük seviyeye çekilirse yapılacak işlem bir kontrol işlemidir. Eğer yüksek seviyede tutulursa gönderilen bir komut değil karakterdir.



Şekil 7. LCD göstergeye yazma işlemine ait zamanlama diyagramı

3.3.2. LCD komutları

Burada en çok kullanılan LCD komutları anlatılacaktır.

Fonksiyon belirleme(Function Set) : Bu komut ile veri yolunun büyüklüğü göstergedeki satır sayısı ve font büyüklüğü belirlenir.

Göstergesi kapat/aç (Display off/on) : LCD göstergesi açmak yada kapatmak için kullanılır. Göstege kapalı iken DDRAM daki veriler korunur.

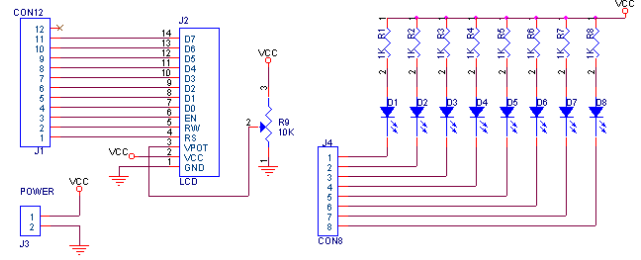
Göstergesi temizle(Clear Display): Göstergenin temizlenmesi DDRAM daki her hücreye boşluk(20H) karakterinin yazılmasını ve imlecin sol baştan göstergenin ilk karakterine konumlanmasını sağlar.

İmleci kaydır(Shift Display): İmleci veya göstergesi sola veya sağa kaydırmak için kullanılır.

İmleç Başa (Cursor Home) : İmleci en sola ve en üst satıra konumlandırır.

Giriş Kipi(Entry mode) : Giriş kipi her karakter okuma veya yazma işlemini takiben, imlecin veya göstergenin işlemini belirler. En çok kullanılan işlem modu, göstergenin o anki değerlerinin korunarak imlecin bir sağa kaydırılmasıdır. Bu düzenleme ile bir sonraki karakter o anki konumun bir sağına yazılır

Şekil-8’de LCD Modül devre şeması ve LED modül devre şeması görülmektedir



(a) LCD Modül devre şeması

(b) LED modül devre şeması

3.3.3. LCD ve LED yazılımı

Mikrodenetleyici LCD ve LED denetimi için yapılması gereken değişken tanımlamaları şu şekildedir.

```
#include<AT89X52.h> /* tanımlı register isimlerini kullanabilmek için gerekli header dosyası*/
#include<intrins.h> /*_nop_ ; komutunu kullanabilmek için gerekli header dosyası*/
sfr LCD_DATA=0x80; /* LCD VE ADC verisi* P0 portu uçları*/
sbit LCD_RS=0xa7 ; /* LCD ekran RS ucu P2.7 port ucu*/
sbit LCD_RW=0xa6 ; /* LCD ekran Read/Write ucu P2.6 port ucu*/
sbit LCD_E=0xa5 ; /* LCD ekran Enable ucu P2.5 port ucu*/
sfr LEDDATA=0x80 ; /* P1 portu aynı zamanda LED leri yakmak için de kullanılıyor*/
unsigned char idata leddegeri;
LCD kontrolü için gerekli alt programlar aşağıda verilmektedir.
/*-Gecikme alt programı--*/
void DELAY(unsigned int x)
{unsigned int data j;
for(j=0;j<x;j++){_nop_ ;_nop_ ;}}
/*Enable darbesi üretme alt programı*/
void CLOCK()
{
LCD_RW=0;LCD_E=0;DELAY(1);
LCD_E=1;DELAY(1);LCD_E=0;
}
/*-LCD resetleme alt programı */
/*function set 3c, display on 0c, clear display 01, entry mode 06, cursor home 02 */
void RESETLEME()
{
LCD_RS=0;LCD_DATA=0x3c;
CLOCK();DELAY(18000);
LCD_RS=0;LCD_DATA=0x3c;
CLOCK();DELAY(400);
}
```

```

LCD_RS=0;LCD_DATA=0x3c;
CLOCK();DELAY(400);
LCD_RS=0;LCD_DATA=0x3c;
CLOCK();DELAY(200);
LCD_DATA=0x0c; CLOCK();DELAY(200);
LCD_DATA=0x01; CLOCK();DELAY(8000);
LCD_DATA=0x06; CLOCK();DELAY(200);
}
/*--LCD data gönder alt programı-- */
void DATA_GONDER()
{
LCD_DATA=ACC;LCD_RS=1;CLOCK();
}
/*--LCD imleç konumlandırma-- */
/*128 line1 ,136 line2,192 line3, 200 line4 */
void LINE(unsigned char y)
{
LCD_DATA=y;LCD_RS=0;CLOCK();
}
/*Girilen bir sayıyı LCD göstergeye yazan alt
program*/
void DISPLAY(unsigned int x)
{
unsigned int data sayi;
unsigned char data s2,s1,s0;
sayi=x;
s2=sayi/100;
sayi=sayi%100;
s1=sayi/10;
sayi=sayi%10;
s0=sayi;
ACC=s2+48;DATA_GONDER(); /* s2 rakamı LCD
displayde görünür*/
ACC=s1+48;DATA_GONDER();
ACC=s0+48;DATA_GONDER();
}
/*Girilen bir sayıya göre LCD göstegeye mesaj yazan
altprogram*/
unsigned char code msg0(16)="GAZI UNV. ";
unsigned char code msg1(16)="TEK. EGT. FAK. ";
void MESAJ(unsigned char x)
{unsigned char i;
LINE(128);
for(i=0;i<=15;i++)
{switch (x)
{ case 0:ACC=msg0(i);DATA_GONDER();break;
case 1: ACC=msg1(i);DATA_GONDER();break;
}
}
}
/*LEDleri sağa veya sola yakan alt program*/
void LEDLER(unsigned char x)
{
unsigned char direction;
direction=x;
switch(direction)
{

```

```

case 0:
LEDDATA=~(leddegeri);
leddegeri=leddegeri >> 1;
if(leddegeri<=0) leddegeri=0x80;break;
case 1:
LEDDATA=~(leddegeri);
leddegeri=leddegeri << 1;
if(leddegeri<=0) leddegeri=0x01;break;
}
}
}

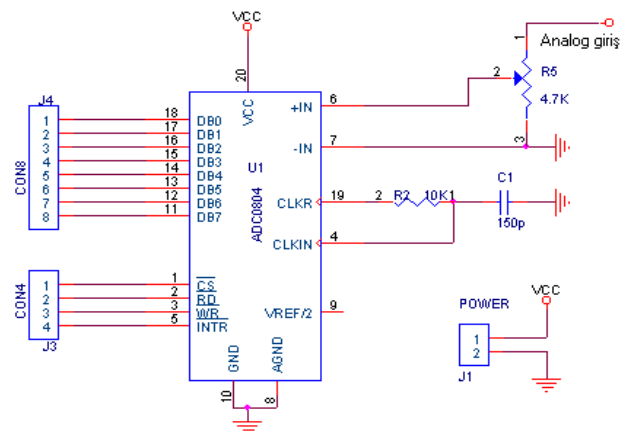
```

3.4. Analog/Sayısal çevirici modülü

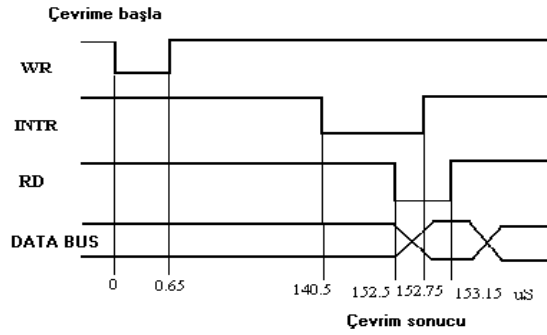
Bu modülde analog/sayısal çevirici olarak ADC804 tümdevresi kullanılmıştır. ADC804 tümdevresi 8 bitlik bir dönüştürücüdür. Bu modülün devre şeması Şekil-9'da görülmektedir. Dönüştürücünün içinde kendi referans üreticisi vardır ancak istenirse dışarıdan VCC/2 değerinde bir referans da verilebilir. Bu taktirde ADC804 5 voltluk bir giriş sinyalini FFH(255) olarak çevirmektedir. ADC girişine analog sinyal bir potansiyometre üzerinden verilmektedir.

ADC804'ün mikrodenetleyici yazılımı yapılırken Şekil-10'daki zamanlama diyagramı kullanılmıştır. ADC804'ün 4 ve 19 nolu bacaklarına bağlanan direnç ve kondansatör ile ADC804 için gerekli olan saat(clock) sinyali üretilmektedir. Üretilen sinyalin frekansı:

$F_{clock}=1/1.1 \cdot R \cdot C$ dir. $R=10K$ ve $C=150pF$ seçilirse Frekans değeri 606 Hz olmaktadır. Devrede yüksek hız kullanılmamıştır. Şekil-10 da 500 kHz için örnek zamanlama diyagramı görülmektedir.



Şekil-9 a) ADC804 analog/sayısal çevirici modül devre şeması



Şekil-10 ADC804 zamanlama diyagramı

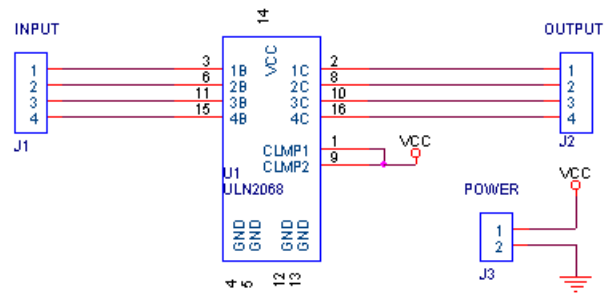
3.4.1. ADC yazılımı

Mikrodenetleyicinin ADC kontrol yazılımı aşağıda verilmektedir.

```
#include<AT89X52.h> /* tanımlı register isimlerini
kullanabilmek için gerekli header dosyası*/
#include<intrins.h> /*_nop_; komutunu
kullanabilmek için gerekli header dosyası*/
sfr ADC_DATA=0x90; /* ADC data girişi
P1 port uçları */
sbit ADC_CS=0xb4 ; /* P3.4 port ucu*/
sbit ADC_RD=0xb5 ; /* P3.5 port ucu*/
sbit ADC_WR=0xb6 ; /* P3.6 port ucu*/
sbit ADC_INTR=0xb7 ; /* P3.7 port ucu*/
unsigned char idata dcdger;
void ADCOKU(void)
{
ADC_CS=0; ADC_INTR=1;
ADC_WR=1;ADC_RD=1;
ADC_WR=0;DELAY(1);
ADC_WR=1;DELAY(50);
ADC_RD=0;DELAY(1);
/* while(ADC_INTR==1); ucu kullanılmamıştır bu uç
mikroislemciye bağlanmayabilir*/
ADC_DATA=0xff;
dcdger=ADC_DATA;
ADC_RD=1;
ADC_CS=1;
}
```

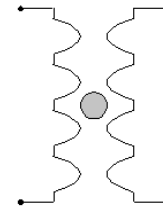
3.5.Step Motor Sürücü Modülü

Step motor sürücü devresi ULN2068 tümdevresi ile yapılmıştır. Bu tümdevre içinde herbir çıkış üç transistörle sürülmekte olup 1.5 Amper akım verebilmekte ve 10 Volt besleme ile çalışmaktadır. Hem bipolar hemde unipolar step motoru sürebilmektedir. Şekil-11’de bu modülün devre şeması verilmiştir.

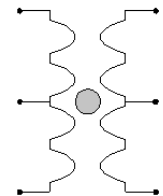


Şekil-11 Step motor sürücü modülü devre şeması

Bipolar step motor 4 uca sahip olup şematik gösterimi Şekil-12’deki gibidir. Unipolar step motor 5 veya 6 uca sahip olup şematik gösterimi Şekil 13 deki gibidir. Unipolar step motor devreye bağlanırken ortak uçlar 10-12 voltluk bir Vcc kaynağa bağlanır. Diğer uçlardan sürücü sinyaller gönderilir.



Şekil-12 Bipolar step motor



Şekil-13 Unipolar step motor

3.5.1. Step motor yazılımı

Step motor için mikrodenetleyici kontrol yazılımı aşağıda verilmektedir.

```
#include<AT89X52.h> /* tanımlı register isimlerini
kullanabilmek için gerekli header dosyası*/
#include<intrins.h> /*_nop_; komutunu
kullanabilmek için gerekli header dosyası*/
sfr MOTORDATA=0xa0; /*P2 portu P2.0 P2.1 P2.2
P2.3*/
unsigned char ptr;
void stepmotor(unsigned char x)
{
unsigned char code steptabright(
)= {0x08,0x0c,0x04,0x06,0x02,0x03,0x01,0x09,
```



```
0x08,0x0c,0x04,0x06,0x02,0x03,0x01,0x09}; /*yarım
adım sağa döndürme tablosu*/
  unsigned char code steptableft(
)= {0x09,0x01,0x03,0x02,0x06,0x04,0x0c,0x08,
```

```
0x09,0x01,0x03,0x02,0x06,0x04,0x0c,0x08}; /*yarım
adım sola döndürme tablosu */
  unsigned char direction;
  direction=x;
  switch(direction)
  { case 0: /*Sığa döndür*/
    MOTORDATA=steptableft(ptr) & 0x0f;
    ptr++;
    if (ptr>15) ptr=0;
    break;
  case 1: /*Sola döndür*/
    MOTORDATA=steptableft(ptr) & 0x0f;
    ptr++;
    if (ptr>15) ptr=0;
    break; }
```

3.6. Mikrodenetleyici seri port haberleşme yazılımı

Yukarda bahsedilen modüllerin bilgisayar vasıtası ile kontrolünü gerçekleştirmek için gerekli olan seri port yazılımı ise aşağıda verilmektedir.

```
#include<AT89X52.h> /* tanımlı register isimlerini
kullanabilmek için gerekli header dosyası*/
#include<intrins.h> /*_nop_; komutunu
kullanabilmek için gerekli header dosyası*/
  unsigned char rchar,tchar,bekle;
/* bu alt program seri port veri alış verişi için gerekli*/
void WAIT(void)
{ bekle=0xff;
  while(bekle !=0);}
/* Seri portu ayarlama */
  PCON=0x80; /*for 19200 baud
0x80*/
  SCON = 0x50; /* mode 1: 8-bit
UART, enable receiver */
  TMOD = 0x21; /* timer 1 mode 2 8-
Bit reload , Timer0 mode1 16 bit */
  TH1 = 0xfd; /* reload value 57600 baud
the same for 19200 baud*/
  TR1 = 1; /* timer 1 run for serial
communication */
/*Bilgisayara veri göndermek için*/
  tchar=adcdeger;
  SBUF=tchar;
  WAIT();
/*Bilgisayardan veri almak için*/
void serial_int() interrupt 4
{ EX0=0;ES=0;
  if(TI==1)
    {TI=0; bekle=0;ES=1;}
  else
```

```
{
  RI=0;
  ACC=SBUF;
  rchar=ACC;
  if(rchar==241) {Motoru sağa döndür
komutları}
  if(rchar==242){motoru sola döndür komutları}
  if(rchar==243){Ledleri sağa doğru yak
komutları}
  if(rchar==244){Ledleri sola doğru yak }
  if(rchar==245){ADC çevrimini başlat}
  if(rchar==246){Motor, led ve ADC dur}
  if(rchar>0 && rchar<240){hiz=rchar;}
  bekle=0;
  ES=1;
}
EX0=1;
}
```

4. BİLGİSAYAR SERİ HABERLEŞME PORTUNUN KULLANIMI

Bilgisayar arayüz yazılımı Borland C++ Builder görsel programlama dili ile hazırlanmıştır. C++ Builder, C++ dilinin özelliklerini bünyesine barındıran ve güçlü bir program geliştirme ortamı kullanıcıya sunan görsel bir programlama dilidir(13). Programlama dillerinde seri port kullanımı çok iyi bilinen bir konu değildir. Yapılan bilgisayar arayüz yazılımı mikrodenetleyiciye verileri seri porttan göndermektedir. Bu yüzden seri haberleşmenin bu dilde nasıl yapıldığının bilinmesi gerekmektedir. Bu işlem üç adımda gerçekleşmektedir.

- Port'un kullanıma açılması
- Veri gönderme ve alma
- Port'un kapatılması

Port'un kullanıma açılması için öncelikle aşağıdaki global değişkenler tanımlanır.

```
HANDLE hComm = NULL;
COMMTIMEOUTS ctmoNew = {0}, ctmoOld;
DWORD dwBytesRead;
DCB dcbCommPort;
  unsigned char Buffer(10);
```

Daha sonra Windows'un CreateFile fonksiyonu kullanılarak aşağıdaki şekilde kullanıma açılır.

```
hComm = CreateFile("COM2",
  GENERIC_READ | GENERIC_WRITE,
  0,
  0,
```

```

OPEN_EXISTING,
0,
0)

```

Portun kullanıma açılıp açılmadığı aşağıdaki fonksiyon ile öğrenilebilir.

```

if(hComm == INVALID_HANDLE_VALUE)
{
ShowMessage("Port açılmadı");
Application->Terminate();
}

```

Port'un veri gönderme ve almada kullanacağı buffer büyüklüğü aşağıdaki fonksiyon ile ayarlanır. Hem giriş hemde çıkış bufferi 128 bayt olarak ayarlanmıştır

```
SetupComm(hComm, 128, 128);
```

Port'un zaman aşım değerleri aşağıdaki fonksiyonlar ile ayarlanır.

Kullanılmakta olan zaman aşım değerlerini sakla

```
GetCommTimeouts(hComm, &ctmoOld);
```

Yeni zaman aşım değerlerini ayarla

```

ctmoNew.ReadTotalTimeoutConstant = 100;
ctmoNew.ReadTotalTimeoutMultiplier = 0;
ctmoNew.WriteTotalTimeoutMultiplier = 0;
ctmoNew.WriteTotalTimeoutConstant = 0;
SetCommTimeouts(hComm, &ctmoNew);

```

Portun kullanacağı protokol 19200 baud rate, no parity, 8 bit veri uzunluğu ve 1 stop biti olarak aşağıdaki şekilde ayarlanır.

```

dcbCommPort.DCBlength = sizeof(DCB);
GetCommState(hComm, &dcbCommPort);
BuildCommDCB("19200,N,8,1", &dcbCommPort);
SetCommState(hComm, &dcbCommPort);

```

Veri haberleşmesi bittikten sonra programdan çıkarken port'u tekrar eski haline getirmek için aşağıdaki komutlar kullanılır.

```

PurgeComm(hComm, PURGE_RXABORT);
SetCommTimeouts(hComm, &ctmoOld);
CloseHandle(hComm);

```

Bir baytlık veri göndermek için örneğin 246 verisini göndermek için

```
TransmitCommChar(hComm, 246);fonksiyonu kullanılır.
```

Veri alışı yapmak için ise aşağıdaki program kullanılabilir

```

while(basla)
{
if(say == 0) {
say=1;TransmitCommChar(hComm,246);}

ReadFile(hComm, &Buffer, 1, &dwBytesRead,
NULL);

if(dwBytesRead) { say=0; adcdeger=Buffer(0); }

Application->ProcessMessages();
}

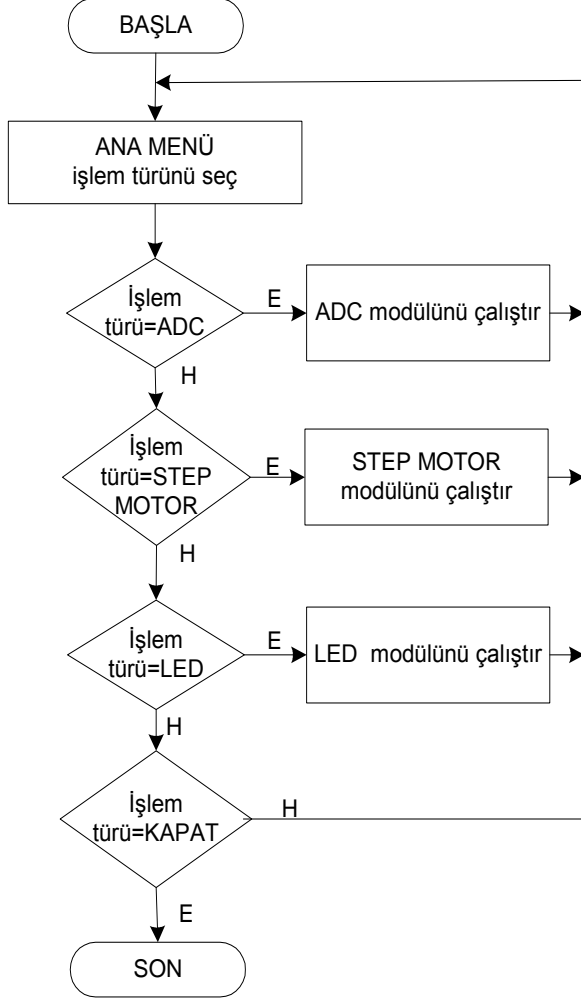
```

Bilgisayar 246 verisini göndermekte, ReadFile fonksiyonu ile de port'u okumaktadır. Porta bir değer gelmişse dwBytesRead değişkeni 0 dan farklı bir değer almaktadır. Bu durumda değer hemen okunmakta ve say=0 sıfır yapılarak yeni bir okuma için porta 246 değeri tekrar gönderilmektedir. Basla değişkeni boolean olarak tanımlanmış ve programda BAŞLA butonuna tıkladığında true(1) değerini almakta ve DUR butonuna basıldığında ise false(0) değerini almaktadır. Buton tıklama olaylarını algılayabilmek ve sonsuz döngüden çıkabilmek için Application->ProcessMessages() fonksiyonu gereklidir.

5. BİLGİSAYAR ARAYÜZ YAZILIMI

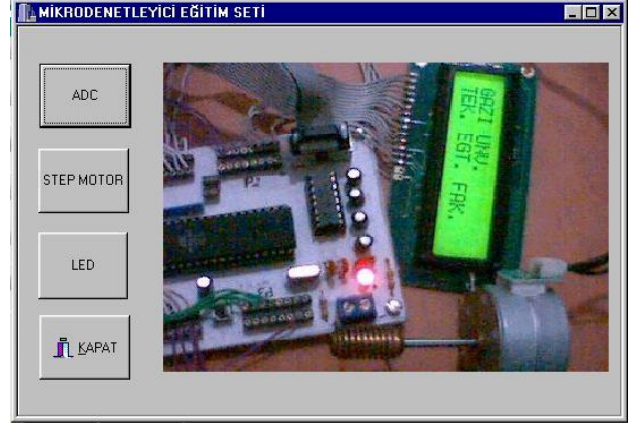
Mikrodenetleyici ana modülü bilgisayara RS-232 seri haberleşme port'u ile bağlanmıştır. Bilgisayar seri port'tan mikrodenetleyiciye, analog/sayısal çevirme modülü(ADC), LED modülünü veya step motoru sürmek için gerekli kontrol verilerini gönderir. Bu veriler yön (sağ, sol), başla, dur, ve hız bilgilerinden oluşur. Mikrodenetleyici bu verileri aldığı zaman gerekli sürücü işaret sinyallerini üreterek modüllere gönderir. Örneğin bilgisayar ADC çevrimini başlat verisini(245) gönderdiğinde mikrodenetleyici ADC çevrimini başlatır ve okuduğu değeri tekrar bilgisayara gönderir. Bilgisayar bu değeri hem sayısal hemde grafik olarak ekranda gösterir. Mikrodenetleyicinin giriş çıkış için kullanılan portları (P0, P1, P2, P3) belirli bir kullanım amacı için ayrılmıştır. Her modül istenilen port'a bağlanabilir. Tuş takımı LCD göstergeye birtakım bilgiler yazdırmak için kullanılmıştır. 0,1,2,3 tuşlarına bastıkça ekrana değişik mesajlar yazdıran bir yazılım da örnek olarak eklenmiştir. Buradaki amaç, tuş takımının ve LCD göstergenin kullanımını göstermektir. Bilgisayar arayüz yazılımı C++

Builder 4.0 görsel programlama dili ile ve mikrodeneleyici kontrol yazılımları ise C dili kullanılarak yapılmıştır. Arayüz yazılımının akış şeması Şekil-14’de verilmiştir.

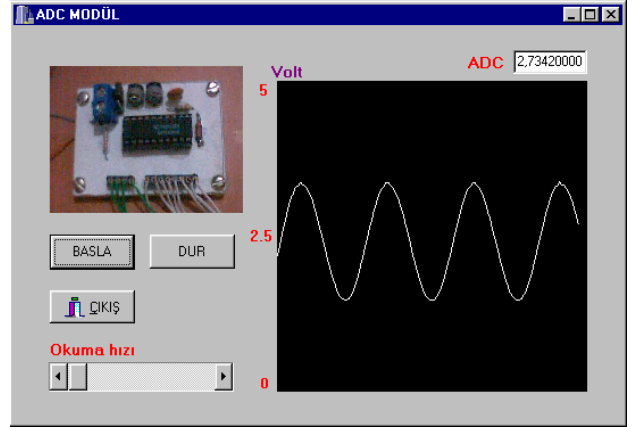


Şekil-14. Bilgisayar arayüz yazılımı akış şeması

Bilgisayar arayüz yazılımı çalıştırıldığında ilk olarak ana menü gelmektedir. Ana menü ekran çıktısı Şekil-15’de görülmektedir. ADC butonuna basıldığında ADC modül ekranı, Step motor butonuna basıldığında Step motor modülü ekranı, LED butonuna basıldığında Step motor modülü ekranı gelmektedir.

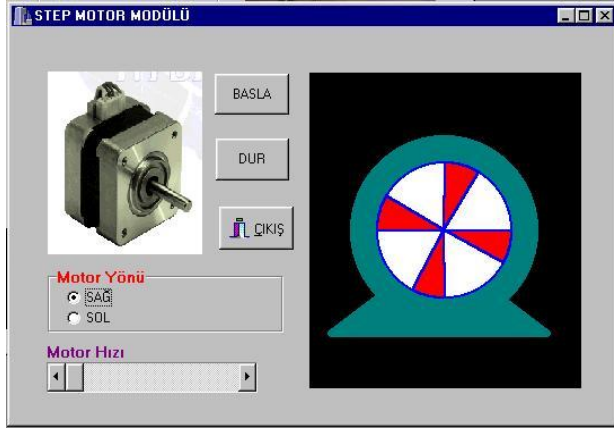


Şekil-15 Bilgisayar arayüz yazılımı ana menü ekranı



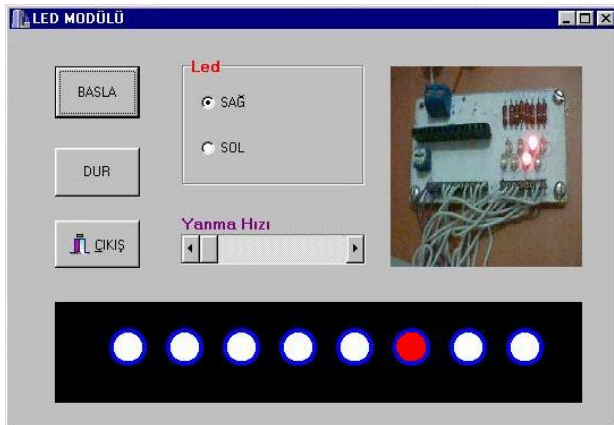
Şekil-16 ADC modülü ekranında gösterilen sinüs sinyali

ADC modül ekranında BAŞLA butonuna basıldığında seri port’tan 245 kodunu, DUR butonuna basıldığında 246 kodunu, Okuma hızı için ise 0-240 arası bir değer seri port’tan mikrodeneleyiciye göndermektedir. Mikrodeneleyici başla verisini alır almaz ADC çevrimini başlatmakta ve okuduğu değeri tekrar seri port’tan bilgisayara göndermektedir. Bilgisayar bu değeri ekranda hem sayısal hem de grafik olarak göstermektedir. Şekil-16’de ADC girişine sinyal jeneratöründen pozitif ofset değeri ayarlanarak, bir sinüs sinyali uygulayarak elde edilen değerler ekranda görülmektedir.



Şekil-17 Step motor modülü ekranı

Şekil-17’de görülen step motor modülü ekranında da benzer bir işlem yerine getirilmektedir. BAŞLA butonuna basıldığında bilgisayar mikrodnetleyiciye 241 kodunu, DUR için ise 246 kodunu mikrodnetleyiciye göndermektedir. Mikrodnetleyici 241 kodunu alır almaz step motoru çalıştırmaktadır. Motorun dönüş yönünü değiştirmek için 241 ve 242 kodu mikrodnetleyiciye gönderilir. Mikrodnetleyici 241 kodunu almakla motoru sağa 242 kodunu almakla motoru sola döndürmektedir. Step motorun dönme hızı ise kaydırma çubuğu ile yapılmaktadır. Bilgisayar, kaydırma çubuğunun konumu değiştirildiğinde, konum bilgisini sayısal değer olarak(0-240 arasında bir değer) mikrodnetleyiciye göndermektedir. Mikrodnetleyici de step motora gönderilen sayısal kontrol sinyallerinin frekansını gelen değere göre değiştirerek step motorun hızını değiştirmektedir. Ayrıca motor dönüş yönü ve hızı ekranda aynı zamanda simüle edilmektedir.



Şekil-18 LED kontrol ekranı

Şekil-18’de görülen LED modülü ekranında ise LED ışıkların sağa, sola, hızlı veya yavaş olarak kayarak yanması uygulaması yapılmaktadır.

Bu işlemler step motora benzer bir şekilde yerine getirilmektedir.

6. SONUÇ

Bu çalışmada endüstriyel amaçlı uygulamalarda çok yaygın olarak kullanılan mikrodnetleyicilerin öğrenimini kolaylaştırmak için bir model eğitim seti ve uygulama devreleri tasarlanmış ve gerçekleştirilmiştir. Ayrıca bir bilgisayar arayüz yazılımı yapılmış ve bu arayüz yazılımı ile bir bilgisayarla bir mikrodnetleyicinin birlikte nasıl kullanılacağı gösterilmiştir. Örnek uygulama devreleri ve bunlara ait mikrodnetleyici yazılımları en basit halleri ile verilerek öğrenilmeleri kolaylaştırılmıştır. Ayrıca yazılımlar C dili kullanılarak yapılmış böylece C dilinin bir mikrodnetleyicide nasıl kullanılacağı gösterilmiştir. Bunun için kullanılan mikrodnetleyiciye ait C derleyicisinin temin edilmesi gerekmektedir.

Kontrol yazılımları istenirse Assembly dili kullanılarak da yazılabilir. C derleyicisi mikrodnetleyicinin bellek kullanımı, kesme kullanımı gibi kaynaklarını kendisi yöneterek kullanıcı ile mikrodnetleyici arasında belli bir ölçüde donanım soyutlaması sağlamaktadır. Böylece programcının daha az veya asgari düzeyde mikrodnetleyicinin iç yapısını ve özelliklerini bilmesi yeterlidir. Ayrıca C dilinin sağladığı program kontrol ve döngü yapılarının kullanılması ile daha kolay, hatasız ve geliştirilebilir programlar yazmak mümkün olmaktadır. Mikrodnetleyicinin program belleği sınırlı olduğundan kullanılacak C derleyicisi de dikkatli seçilmelidir. Çünkü C derleyicilerinin aynı tür işlemler için ürettikleri kod büyüklükleri farklı olmaktadır. Bu hem bellek kullanımı hem de performans açısından önem arz etmektedir.

KAYNAKLAR

1. Barnett, R.H., The Family of Microcontrollers, Prentice Hall, New Jersey, 1995
2. Sinha, N.K., Microprocessor- Based Control Systems, D. Reidell Publishing Company, Holland., 1986
3. Intel 8-bit Embedded Controller Handbook, 1991
4. Atmel Corporation CDROM product Guide, 1999
5. Santos, A.; Boemo, E.; Faura, J.; Meneses, J. , Microcontrollers in education: a case-study, Frontiers in Education Conference, Twenty-

- fourth Annual Conference. Proceedings , Page(s): 378 –382, 1994
6. Beetner, D.; Pottinger, H.; Mitchell, K. Laboratories teaching concepts in microcontrollers and hardware-software co-design, Frontiers in Education Conference, 30th Annual, Volume:2,.2000.
 7. Del Rio, A., Andina, J.J.R., UV151:a simulation tool for teaching/learning the 8051microcontroller, Frontiers in Education Conference, 30th Annual , Volume: 2 , Page(s): F4E/11 -F4E/16, 2000
 8. Nunnally, C.E.,Teaching microcontrollers, Frontiers in Education Conference, FIE '96. 26th Annual Conference., Proceedings of , Volume: 1 , Page(s): 434 -436 vol.1, 1996
 9. del Rio, A.; Rodriguez, J.J.; Nogueiras, A.A., Learning microcontrollers with a CAI oriented multi-micro simulation environment, Education, IEEE Transactions on , Volume: 44 Issue: 2 , 2001
 10. A.H.G. Al - dhaher, Integrating hardware and software for the development of microcontroller - based systems, Microprocessor and microsystems”, 25 , 2001
 11. T.K. Hamrita., R.W. McClendon, A New Approach for Teaching Microcontroller Courses, Int. J. Engng Ed. Vol. 13, 1997
 12. Keil Software Inc, μ Vision2 IDE, C51 Compiler's user guide
 13. Kargülle,İ., Zeydin, P., C++ Builder 4, Türkmen Kitabevi, İstanbul, 2000