

57737

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

MİKROİŞLEMCİ TEMELLİ SİSTEM TASARIMI

YÜKSEK LİSANS TEZİ

Müh. İsmail Ergüden

57737

Ana Bilim Dalı : ELEKTRİK

Programı : MAKİNA

KASIM 1995

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

MİKROİŞLEMCİ TEMELLİ SİSTEM TASARIMI

YÜKSEK LİSANS TEZİ

Elektrik Müh. İsmail ERGÜDEN

Tezin Estitüye Verildiği Tarih : 25 .01.1996

Tezin Savunulduğu Tarih : 26.02.1996

Tez Danışmanı

Yrd. Doc.Dr. Nurettin ABUT

(*Nurettin Abut*)

Üye

Prof. Dr. Halit PASTACI

(*Halit Pastacı*)

Üye

Doc.Dr. Oruç BİLGİÇ

(*Oruç Bilgiç*)

ŞUBAT 1996

MIKROİŞLEMCİ TEMELLİ SİSTEM TASARIMI

İsmail Ergüden

Anahtar Kelimeler: Mikrokontrolör, Otomatik Kontrol Sistemi, Çipler, Film Banyo Makinası, Otomatik Film Banyo Sistemi

Özet: Bu çalışmada, mikrokontrolör temelli bir kontrol sistemi tasarlanarak pratik olarak gerçekleştirilmiş, parçaları ve bütünü incelenmiştir. Bu kontrol sistemi, bir otomatik film banyo makinasının kontrol ünitesi olarak tasarlanmıştır. Film banyo makinasının bütün mekanik bölümleri, bu tez çalışmasında üretilen elektronik devrenin denetiminde çalışmaktadır. Teorik açıklamada, kullanılan örnek mikrokontrolör donanım ve yazılım olarak tanıtılmış, daha sonra da tasarımın detayları belirtilmiştir. Sistem, bir mikrokontrolör ve onu destekleyen yardımcı devre gruplarından oluşmaktadır. Bu yardımcı devrelerin bazıları zamanlama, motor kontrol gibi özel fonksiyonlar yerine getirmek için üretilmiş bağımsız mikroişlemcilerdir. Burada ana mikrokontrolör, organizasyon görevini üstlenmiştir. Böylece birkaç mikroişlemcinin, aynı sistemde iş bölümü yaparak çalışması gerçekleştirilmiştir.

MICROPROCESSOR BASED SYSTEM DESIGN

İsmail Ergüden

Keywords: Microcontroller, Automatic Control System, Chips, Film Processor, Automatic Film Processor System

Abstract: In this work, subparts and mainparts of a microprocessor based control system are designed and examined. This control system is designed as control unit of an automatic film processor. All mechanical parts of automatic film processor works under control of designed electronic circuit. Teorically, used microcontroller is identified as hardware and software, and then details of design are showed. System has a microcontroller and co-circuit groups that aids it. Some of these co-circuits are independent microprocessors that produced for special control functions like timing and motor control. Microcontroller makes organization mission in this system. So, together working of some microprocessors are realized.

ÖNSÖZ ve TEŞEKKÜR

Günümüz kontrol sistemlerinde mikroişlemcilerin kullanımı ve önemi her geçen gün artmaktadır. Yeni tasarlanan hemen her komplike sistemde, bir veya birkaç mikroişlemci bulunmakatadır. Mikroişlemcilerin belirli ve küçük hata oranları, esnek tasarım özellikleri, kararlı çalışmaları ve tasarımında yalnızca program değiştirilerek modifikasyon yapılabilmesi onları çok popüler devre elemanları haline getirmiştir.

Bu tezde, ülkemiz için yeni sayılabilecek bir konu olan mikroişlemcilerle bir kontrol sisteminin nasıl gerçekleştirilebileceği incelenmiştir. Bu konunun teorik detayları belirtildikten sonra, bir fotoğraf filmi yıkama makinasının kontrol sistemi gerçekleştirilmiş ve bunun yapısı açıklanmıştır. Çok kullanılan kontrol fonksiyonlarının hemen hepsi bu çalışmada yer almaktadır. Yalnızca programı değiştirilerek bu örnek sistem çok farklı bir sisteme dönüştürülebilir.

Tezin asıl sonucu olan bu örnek sistemin, bu tarz kontrol sistemlerinin tasarımında çalışacaklara ve yerli tasarım modern kontrol sistemlerinin ülkemizde yaygınlaşmasına katkıda bulunmasını dilerim.

Bana bu konuda çalışma olanağı veren danışmanım Sayın Yrd.Doç.Dr. Nurettin Abut'a teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET.....	ii
ABSTRACT.....	iii
ÖNSÖZ ve TEŞEKKÜR.....	iv
İÇİNDEKİLER.....	v
SİMGELER DİZİNİ.....	vi
ŞEKİLLER LİSTESİ.....	vii
BÖLÜM 1. GİRİŞ.....	I
BÖLÜM 2. MCS 51 MİKROKONTROLÖR AİLESİ.....	4
2.1. Donanım Yapısı.....	4
2.2. Yazılım Yapısı.....	7
2.2.1. Aritmetik komutları.....	8
2.2.2. Mantık komutları.....	8
2.2.3. Veri transfer komutları.....	9
2.2.4. Boolean matematiği komutları.....	10
2.2.5. Sıçrama komutları.....	10
BÖLÜM 3. 87C51 TEMELLİ SİTEM TASARIMI.....	11
3.1. Makina Fonksiyonları.....	11
3.2. Genel Sistem Organizasyonu.....	12
3.3. Devrenin Yapısı ve Çalışması.....	15
BÖLÜM 4. PROGRAM.....	21
SONUÇLAR ve ÖNERİLER.....	52
KAYNAKLAR.....	54
ÖZGEÇMİŞ.....	55

SİMGELER DİZİNİ VE KISALTMALAR

F : Kapasite

Hz : Frekans

B : Hafıza (1 Bayt=8bit)

KB : Hafıza (1KB=1024B)

s : Zaman

V : Gerilim

W : Güç

Ω : Direnç

EPROM : Electrically Programmable Read Only Memory

RAM : Random Accessable Memory

ROM : Read Only Memory

TTL : Transistor Transistor Logic

ŞEKİLLER DİZİNİ

- Şekil 1. 8051'in mimari yapısının blok gösterimi.....
- Şekil 2. Film yıkama makinası kontrol sistemi.....
- Şekil 3. Mikroişlemci temelli kontrol sistemi elektronik devre şeması.....



BÖLÜM 1. GİRİŞ

İnsanoğlu, elektriğin bulunmasından bugüne kadar bu ilginç olayı ve etkilerini araştırmış, sonuçta ortaya elektronik bilimi çıkmıştır. Elektronik günümüzde ulaştığı nokta göz kamaştırıcıdır. Yarıiletkenlerin bulunması elektroniğe yepyeni ufuklar açmış, günden güne ilerleyen teknoloji bugünkü mikroçipleri ortaya çıkarmıştır. İlk bulduklarında önce standart mantık devrelerinin küçültülmesi amacıyla kullanılan mikroçipler artık daha komplike devreler olan mikroişlemciler olarak karşınıza çıkmaktadırlar. Fiziki boyutlar olarak çok küçük bir hacme sahip olan bu mucizevi elemanlar, hacimleriyle hiç de orantılı olmayan büyük çapta karmaşık işleri büyük bir hızla yapabilmektedirler.

Temel olarak bir mikroişlemci, oldukça fazla sayıda karmaşık aritmetik ve mantık devresinin birleşiminden oluşur. Mikroişlemci, bu devreler sayesinde basit işlemler yapabilir ve basit kararlar verebilir. Dolayısıyla mikroişlemcinin detaylı ve karmaşık işler yapabilmesi için bir program ile yönlendirilmesi gerekir. Bu programda yapılması istenen işler, mikroişlemcinin anlayabileceği basit işler olarak sıralanır. Mikroişlemciden beklenen, bu işleri sırayla ve büyük bir hızla yapmasıdır. Zaten konunun özü de budur. Mikroişlemciler, yalnızca basit işler yapabilirler. Fakat bunları çok büyük bir hızla yaparlar. Böylece biraz daha uzun zaman dilimlerinde çok komplike işleri bitirebilirler.

Mikroişlemcilerin günümüzdeki en popüler uygulamaları bilgisayarlardır. Bir bilgisayarın beyni mikroişlemcidir. Bilgisayarın vermesi gereken bütün kararları mikroişlemci verir. Bilgisayarın yapısına bağlı olarak mikroişlemci, değişik özellikteki hafızalar (RAM, ROM, Harddisk vb.) ve giriş-çıkış birimleri tarafından desteklenir. Böylece mikroişlemcinin oldukça fazla sayıda girdiyi

değerlendirerek daha detaylı karşılaştırmalar yapabilmesi sağlanır ki bu sisteme de bilgisayar denir.

Mikroişlemcilerin sıradan insanlar tarafından pek bilinmeyen fakat en az bilgisayarlar kadar önemli bir diğer uygulamaları da otomatik kontrol sistemleridir. Otomatik kontrol sistemlerinde, sistemin belli girdi parametrelerine bağlı olarak çıkış büyüklüklerinin kararlı olması istenir. Yani sistem küçük çapta kararlar vermelidir. Mikroişlemcilerin otomatik kontrol sistemlerinde bu kadar yaygınlaşmalarının sebebi, fazla sayıda girdiyi büyük bir hızla karşılaştırarak her zaman aynı kararlılıkta çıktı üretebilmeleridir. Mikroişlemci denetimindeki bir kontrol sistemi, analog benzerlerinden her zaman için üstündür. Üstünlüğün temel sebepleri; giriş değişikliklerine son derece hızlı cevap verebilmeleri ve şartlar ne olursa olsun aynı giriş değişkenlerine karşı aynı çıkış büyüklüklerinin alınmasıdır. Sonuç, hatasız çalışan mükemmel kontrol sistemidir.

Mikroişlemcilerin otomatik kontrol sistemlerinde daha rahat kullanılabilmesi amacıyla bir ileri türevleri olan mikrokontrolörler geliştirilmiştir. Bir mikrokontrolör temel olarak bir mikroişlemci, programın saklandığı bir ROM hafıza, değişkenlerin saklanması için bir RAM hafıza ve giriş-çıkışların kontrolü için bir ya da birden fazla arabirimden oluşur. Mikrokontrolör, mikroişlemci aracılığıyla arabirime ulaşan girdileri ROM'daki programda anlatıldığı şekilde işler ve ürettiği sonuçları yine arabirim aracılığıyla dış sisteme verir. Bir mikrokontrolörün bir mikroişlemciye karşı avantajı, daha modüler ve birleşik bir yapıya sahip olmasından dolayı dışarıdan bir ilave devre gerektirmeden çalışabilmesidir. Oysa bir mikroişlemci, yukarıda bahsedilen hafıza vb. devreler olmadan tek başına çalışamaz.

Bu tezde, mikrokontrolör denetimli bir otomatik kontrol sisteminin nasıl tasarlanabileceği araştırılmaktadır. Örnek olarak bir film yıkama makinasını kontrol eden sistem dizayn edilmiştir. Bu dizaynın aşamaları adım adım detaylarıyla anlatılmış, kullanılan mikrokontrolör ve yardımcı elemanların özellikleri incelenmiştir. Bahsi geçen tüm sistem denenerek sağlıklı çalışması gözlenmiştir. Sistemin kalbi, oldukça güçlü ve popüler bir mikrokontrolör olan Intel 87C51'dir. 87C51, sistemin için gerekli olan yardımcı kontrol birimleriyle desteklenmiş ve ortaya bir çok bağımsız kontrol fonksiyonunu hatasız gerçekleştirebilen bir elektronik devre çıkmıştır.



BÖLÜM 2. MCS 51 MİKROKONTROLÖR AİLESİ

2.1. Donanım Yapısı

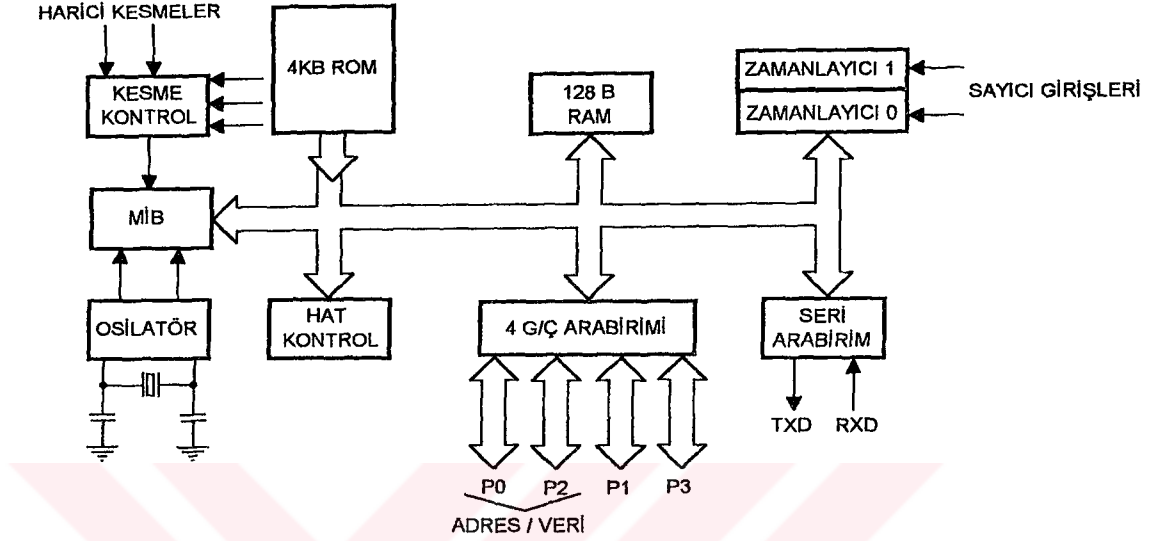
MCS 51 ailesi Intel firmasının ürettiği, günümüzün en çok kullanılan mikrokontrolör türlerinden birisidir. Ailenin genel özellikleri şöyle sıralanabilir:

- * Kontrol özellikleri için optimize edilmiş 8 bit merkezi işlem birimi.
- * Özel Boolean işlemlerini her bit için bağımsız yapabilme yeteneği.
- * 64KB program hafızasını adresleyebilme.
- * 64KB veri hafızasını adresleyebilme.
- * Program hafızasının bir bölümü çipin içinde.
- * Veri hafızasının 128B'ı çipin içinde.
- * 32 adet çift yönlü bağımsız adreslenebilir giriş-çıkış hattı.
- * 2 adet 16 bit zamanlayıcı / sayıcı.
- * Çift yönlü seri arabirim.
- * İki öncelik seviyesinde 6 bağımsız kesme kaynağı.

Görüldüğü gibi MCS 51 mikrokontrolör ailesi oldukça detaylı ve güçlü özelliklere sahiptir. Bu ailenin ortak özelliklerini taşıyan 8051'in mimari yapısı Şekil-1'de gösterilmiştir.

Bu tezde anlatılan sistemde MCS 51 ailesinden 87C51 kullanılmıştır. 87C51'in 8051'den farkı, programın saklandığı hafızasının EPROM olmasıdır. Bu

özelliği sayesinde sistemde değişiklik yapılması gerektiğinde programı silinerek kolayca değiştirilebilir. Diğer bütün özellikleri 8051 ile aynıdır.



Şekil 1. 8051'in mimari yapısının blok gösterimi

87C51, maksimum 12MHz. saat frekansı ile çalışabilir. Her bir komut periyodu 12 saat periyoduna eşittir. Dolayısıyla kısa komutlardan 1000000 tanesini 1 saniye içerisinde gerçekleştirebilir. Osilatör için gerekli donanım çipin içinde olduğundan dışarıdan yalnızca çalışmasının istendiği frekansta bir kristal bağlanması yeterlidir. Çipin içinde 4KB dahili EPROM program hafızası mevcuttur. Mikroişlemci çalışmaya başladığında bu hafızanın sıfırıncı adresinden itibaren komutları uygulamaya başlar. Programlama, herhangi bir standart EPROM programlama cihazı ile yapılabilir. Programın çalışması esnasında üretilen geçici verilerin saklanması için ayrıca bir de 128B RAM hafıza çipin içinde bulunmaktadır. Program hafızasının ya da veri hafızasının çipin içinde bulunan bölümlerinin yetersiz kalması durumunda her ikisi de 64KB'a kadar dışarıdan arttırılabilir. Dışarıdan hafıza eklendiği durumlarda 32 adet

giriş-çıkış hattından 8 ya da 16 tanesi çoğullamalı olarak bu hafizya ulaşmak için kullanılır.

87C51, 2 adet 16 bit dahili sayıcı veya zamanlayıcıya sahiptir. Bunların her ikisi birbirinden bağımsız çalışabilir. Tetikleme ya da sayma girişleri çipin dışarısına ayak olarak çıkarılmıştır. Bu ayaklardan her iki sayıcıya da harici olarak TTL uyumlu sinyallerle ulaşmak mümkündür. Sayıcı veya zamanlayıcı seçimi program aracılığıyla yapılır.

Mikrokontrolörün dış sistemlerle haberleşmek ya da bir başka deyişle kumanda fonksiyonlarını yerine getirebilmek için kullandığı 32 adet bağımsız giriş-çıkış hattı vardır. Bu hatların hepsi çoğullama olmaksızın direkt ve bire bir ayak olarak çip dışına çıkarılmıştır. Bu hatlara bağımsız bit adreslerini kullanarak tek tek veya 8'erli 4 bayt grubu olarak ulaşmak mümkündür. Sistemin ihtiyacına göre bu hatlardan her biri TTL seviyesinde bir giriş veya yine TTL seviyesinde bir çıkış olarak kullanılabilir. Bir giriş-çıkış hattının herhangi bir anda hangi yönde çalışacağını program belirler. Bu hatlar, mikrokontrolörün kumanda ettiği sistemin durumuyla ilgili geri besleme girdilerini aldığı duyu organlarıdır. Ayrıca sonuç fonksiyonları da bağımsız bitler seviyesinde veya analoga çevrilmek üzere baytlar seviyesinde bu hatlardan dışarı gönderilir. Giriş-çıkış hatlarından ikisi de gerektiğinde kullanılmak üzere seri iletişim için ayrılmıştır. Bu iletişimde standart seri iletişim protokolü kullanılır. İletişimin hızı, seçilen kristal hızına ve programa bağlıdır.

Mikrokontrolörün en can alıcı fonksiyonlarından birisi de kesme yapısıdır. Kesmenin (interrupt) ne olduğuna kısaca değinirsek, kesme; bir mikroişlemcinin çalışmasına dışarıdan müdahale etmek manasına gelir. Normal şartlarda mikroişlemci bir program kontrolünde çalışırken, program rutinlerinin gereği

olan kontrolleri belli bir sırayla yapar. Bu kontrollere karşı verdiği cevaplar her zaman farklı olabileceğinden, bu kontroller çok düzenli olmayan aralıklarla yapılır. Dış sistemin ikazlarına mikrosaniyeler mertebesinde hızlarla cevap verilmesinin istendiği durumlarda programın hızı buna cevap veremez. İşte bunun çözümü olarak kesme sistemi kullanılır. Mikroişlemcinin en az bir kesme girişi vardır. Bu girişe uygun seviyede (yüksek veya düşük) bir dijital sinyal gönderildiğinde mikroişlemci program çalışmasını derhal keser ve programda kaldığı yeri hafızaya not ederek, daha kısa ve modüler olan kesme altprogramına sıçar. Kesme altprogramında, kesme sinyaline cevap olarak hemen yapılması istenen işler sıralanmıştır. Mikroişlemci bu işleri derhal yaptıktan sonra, ana programa kaldığı yerden geri döner.

87C51, 2 adet bağımsız kesme girişine sahiptir. Ayrıca dahili zamanlayıcı ve sayıcılar ile seri haberleşme arabirimi de yine bağımsız olarak kesme üretebilirler. Bunların hangi şartlar oluştuğunda kesme üretecekleri program tarafından belirlenir. Bağımsız iki birimin aynı anda kesme istemeleri durumunda hangisine önce cevap verileceği programda belirtilir. Böylece önceliği büyük olanın kesme isteğine cevap verildikten sonra diğerinin kesmesi de uygulanır.

2.2. Yazılım Yapısı

MCS 51 ailesi oldukça güçlü programlama komutlarına sahiptir. Diğer popüler mikroişlemcilerde bulunan standart komutların hemen hepsi mevcuttur. Ayrıca bu aileye ait özel amaçlı komutlar da bulunmaktadır. Genel olarak bütün komutlar, hızlı çalışmaya yönelik tasarlanmışlardır. Böylece çok hızlı çalışan ve kısa program rutinleri üretmek mümkündür.

Temel olarak 87C51 de bütün mikroişlemciler gibi kaydedicilerle (register) çalışır. Mikroişlemciler, çalışırken hesaplamayla veya başka yollarla ürettikleri geçici verileri kaydedicilerde saklayarak gerektiğinde işlerler. Hemen bütün mikroişlemcilerde bulunan akümülatör, yığın sayıcı, program sayıcı gibi standart kaydedicilerin hemen hepsi 87C51'de de vardır. Ayrıca programlamayı kolaylaştırmak ve kısaltmak için oldukça geniş (32 adet) bir kaydedici kütüphanesi bulunmaktadır. Bunların hepsini bağımsız olarak komutlar içinde kullanmak mümkündür.

87C51'in bütün komutları şu alt başlıklar altında özetlenebilir:

2.2.1. Aritmetik komutları:

add a,(bayt)	:a=a+(bayt)	:Bir baytlık toplama işlemi.
addc a,(bayt)	:a=a+(bayt)+C	:Bir baytlık eldeli toplama işlemi.
subb a,(bayt)	:a=a-(bayt)-C	:Bir baytlık eldeli çıkarma işlemi.
inc a	:a=a+1	:Akümülatörü bir arttırma işlemi.
inc (bayt)	:(bayt)=(bayt)+1	:Baytın gösterdiği sayıyı bir arttırma işlemi.
inc dptr	:dptr=dptr+1	:dptr kaydedicisinin değerini bir arttırma işlemi.
dec a	:a=a-1	:Akümülatörü bir azaltma işlemi.
dec (bayt)	:(bayt)=(bayt)-1	:Baytın gösterdiği sayıyı bir azaltma işlemi.
mul ab	:a=axb	:Akümülatör ile b kaydedicisini çarpma işlemi.
div ab	:a=a/b, b=kalan	:Akümülatörü b kaydedicisine bölme işlemi.
da a	:Onluk ayar.	:Akümülatörü 2 adet BCD parçaya ayırma işlemi

2.2.2. Mantık komutları:

anl a,(bayt)	:a=a AND (bayt)	:Akümülatör ile bayt arasında VE işlemi.
anl (bayt),a	:(bayt)=(bayt) AND a	:Bayt ile akümülatör arasında VE işlemi.
anl (bayt),#veri	:(bayt)=(bayt) AND #veri	:Bayt ile veri arasında VE işlemi.
orl a,(bayt)	:a=a OR (bayt)	:Akümülatör ile bayt arasında VEYA işlemi.

orl (bayt),a	:(bayt)=(bayt) OR a	:Bayt ile akümülatör arasında VEYA işlemi.
orl (bayt),#veri:(bayt)=(bayt) OR #veri		:Bayt ile veri arasında VEYA işlemi.
xrl a,(bayt)	:a=a XOR (bayt)	:Akümülatör ile bayt arasında ÖZEL OR işlemi.
xrl (bayt),a	:(bayt)=(bayt) XOR a	:Bayt ile akümülatör arasında ÖZEL OR işlemi.
xrl (bayt),#veri:(bayt)=(bayt) XOR #veri		:Bayt ile veri arasında ÖZEL OR işlemi.
clr a	:a=0	:Akümülatörü sıfırlama işlemi.
cpl a	:a=NOT a	:Akümülatörün deęilini alma işlemi.
rl a	:	:Akümülatörü 1 bit sola döndürme işlemi
rlc a	:	:Akümülatörü elde ile birlikte 1 bit sola döndürme işlemi.
rr a	:	:Akümülatörü 1 bit saęa döndürme işlemi
rrc a	:	:Akümülatörü elde ile birlikte 1 bit saęa döndürme işlemi.
swap a	:	:Akümülatörün ilk 4 bitiyle son 4 bitinin yerlerini deęiştirme işlemi.

2.2.3. Veri transfer komutları:

mov a,(kaynak)	:a=(kaynak)	:Akümülatöre kaynak deęerini yükle.
mov (hedef),a	:(hedef)=a	:Hedefe akümülatörü yükle.
mov (hedef),(kaynak)	:(hedef)=(kaynak)	:Hedefe kaynak deęerini yükle.
mov dptr,#veri16	:dptr=#veri16	:dptr kaydedicisine 16 bitlik veriyi yükle.
push (kaynak)	:	:Kaynak deęerini yığında sakla.
pop (hedef)	:	:Hedef deęerini yığından oku.
xch a,(bayt)	:	:a ile baytın deęerlerini takas et.
xchd a,@Ri	:	:Akümülatör ile Ri kaydedicisinin deęerlerini takas et.
movx a,@Ri	:a=@Ri	:Harici RAM'in Ri adresindeki deęeri akümülatöre yükle.
movx @Ri,a	:@Ri=a	:Akümülatörün deęerini harici RAM'in Ri adresine yaz.
movx a,@dptr	:a=@dptr	:Harici RAM'in dptr adresindeki deęeri akümülatöre yükle.
movx @dptr,a	:@dptr=a	:Akümülatörün deęerini harici RAM'in dptr adresine yaz.
movc a,@a+dptr	:a=a+@dptr	:Akümülatöre harici program hafızasının a ile dptr kaydedicisinin toplanmasıyla bulunan adresindeki deęeri yükle.
movc a,a+pc	:a=a+pc	:Akümülatöre harici program hafızasının a ile program sayacının toplanmasıyla bulunan adresindeki deęeri yükle.

2.2.4. Boolean matematiđi komutları:

and C,bit	:C=C AND bit	:Elde biti ile bit arasında VE işlemi.
and C,/bit	:C=C AND NOT bit	:Elde biti ile bitin deđili arasında VE işlemi.
orl C,bit	:C=C OR bit	:Elde biti ile bit arasında VEYA işlemi.
orl C,/bit	:C=C OR NOT bit	:Elde biti ile bitin deđili arasında VEYA işlemi.
mov C,bit	:C=bit	:Elde bitine biti yükle.
mov bit,C	:bit=C	:Bite elde bitini yükle.
clr C	:C=0	:Elde bitini sıfırla.
clr bit	:bit=0	:Biti sıfırla.
setb C	:C=1	:Elde bitini 1 yap.
setb bit	:bit=1	:Biti 1 yap.
cpl C	:C=NOT C	:Elde bitinin deđilini al.
cpl bit	:bit=NOT bit	:Bitin deđilini al.
jc bađıl	:	:Elde biti 1 ise sıçra.
jnc bađıl	:	:Elde biti 0 ise sıçra.
jb bit,bađıl	:	:Bit 1 ise sıçra.
jnb bit,bađıl	:	:Bit 0 ise sıçra.
jbc bit,bađıl	:	:Bit 1 ise sıçra ve biti sıfırla.

2.2.5. Sıçrama komutları:

jmp adres	:Adrese sıçra.
jmp @a+dptr	:a+dptr adresine sıçra.
call adres	:Adrese sıçra ve geri dön.
ret	:Geri dön.
reti	:Kesme programından ana programa geri dön.
nop	:İşlem yapma.
jz bađıl	:Akümülatörün deđeri 0 ise sıçra.
jnz bađıl	:Akümülatörün deđeri 0 deđil ise sıçra.
djnz (bayt),bađıl	:Baytı 1 azalt ve 0 deđilse sıçra.
cjne a,(bayt),bađıl	:Akümülatör ile bayt eşit deđil ise sıçra.
cjne (bayt),#veri,bađıl	:Bayt ile veri eşit deđil ise sıçra.

BÖLÜM 3. 87C51 TEMELLİ SİSTEM TASARIMI

Bir önceki bölümde, 87C51'in donanım ve yazılım özelliklerine özet olarak değinilmiştir. Buradan da anlaşılacağı gibi 87C51, son derece esnek tasarım olanaklarına sahiptir. Bu tezde asıl incelenecek olan, 87C51 kullanılarak geliştirilmiş bir makina tasarımıdır. Söz konusu makina, fotoğraf filmlerini otomatik olarak banyo etmek için dizayn edilmiştir. Çekilmiş negatif fotoğraf filmleri makinanın giriş bölümüne yüklendiğinde, makina mekanik olarak filmi çekerek içine alır ve sırayla banyo ilaçlarının içinden geçirir. Gerekli ilaçların hepsine giren ve çıkan film, en son bölümde yıkanarak kurutulur ve çıkıştan banyo edilmiş negatif film olarak alınır.

3.1. Makina Fonksiyonları

Bir negatif fotoğraf filminin banyo edilmesi birkaç aşamada olur. Film, 3 adet farklı kimyasal ilacın içinde belli sürelerde kalmalıdır. Ayrıca her ilacın da sıcaklığı özel belli bir sıcaklıkta sabit olmalıdır. İlacın içinde kalma süresinin veya ilacın sıcaklığının değişmesi film üzerindeki renkleri direkt olarak etkiler. Bunu önlemek için makina, 6 adet bağımsız ilaç tankının sıcaklığını 0.1°C toleransla ve filmin makina içinde ilerleme hızını 1s toleransla sabit tutar. Filmin banyo edildikten sonra kurutulması da istendiğinden makina, kurutma rezistanslarının ısıtma ve çalışma-durma zamanlamalarına da kumanda eder.

Makina kullanılmak üzere açıldığında içindeki ilaçlar gereken sıcaklıktan daha soğukturlar. İlk film makinaya verilmeden önce banyoların hızla ısıtılarak gereken sıcaklığa ulaşması beklenmelidir. Bu bekleme dezavantajını gidermek için makinaya otomatik açılma fonksiyonu eklenmiştir. Makina haftanın her günü için bağımsız bir açılma ve kapanma zamanına programlanabilir.

Makinanın dahili tanklarında bulunan ilaçlar film banyo edildiğinde nötrale olur ve eksilir. Eksilen ilaçların yerine aktif yeni ilaç takviyesi yapılması gerekir. Makina verilen filmlerin genişliklerini ve boylarını optik olarak ölçer ve bu filmelere karşılık eksilen banyoyu bir pompa aracılığıyla takviye eder. Ne kadar takviye yapılacağı direkt olarak makinaya giren filmlerin alanlarıyla orantılıdır. Ayrıca tanklardaki banyoların takviye edilmesine rağmen başka sebeplerden dolayı belli bir seviyenin altına düşmesi durumunda, seviye sensörleri sayesinde bu eksiklik otomatik olarak giderilir.

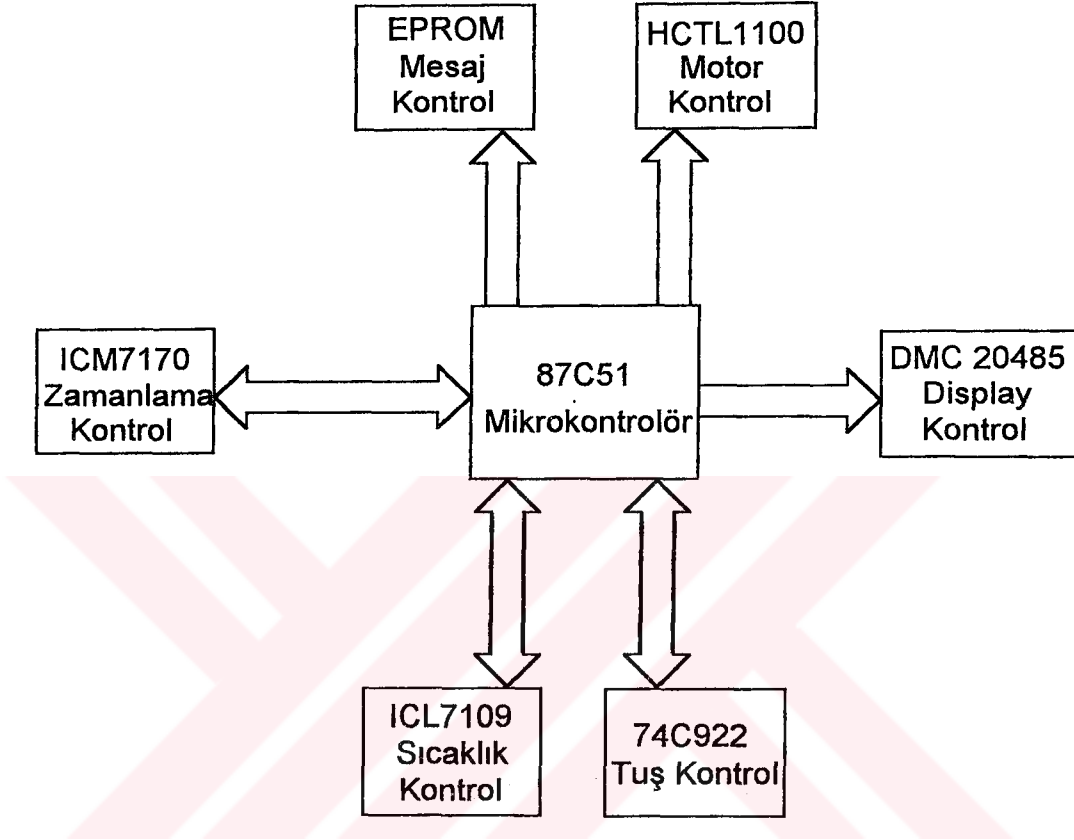
Bir negatif filmin banyo edilmesi esnasında olabilecek bir aksaklık, filmin bozulmasına ve çekilen görüntülerin kaybolma neden olur. Bunun telafisi yoktur. Dolayısıyla makina bir problem olduğunda kullanıcıyı uyarmalıdır. Bu tip uyarılar kullanıcıya direkt sesli mesajlarla yapılır. Mesajlar EPROM'lar içine dijital sesler olarak kaydedilmiştir. Makina gerektiğinde bu yolla uyarıyı yapar.

Görüldüğü üzere bu makinanın çalışması için, sıcaklık kontrol, motor devir kontrol, zamanlama kontrol gibi otomatik kontrol fonksiyonlarını yerine getiren bir sisteme ihtiyaç vardır. Bütün bu işler, 87C51 kontrolünde yardımcı devrelerle yapılmaktadır. Bu sistemde 87C51, daha sonra da açıklanacağı üzere organizatör görevini yapmaktadır.

3.2. Genel Sistem Organizasyonu

Film banyo makinasına kumanda eden sistemin blok diyagramı Şekil 2'de gösterilmiştir. Görüldüğü gibi sistem, otomatik kontrol için gereken tüm

fonksiyonları, bağımsız devre parçalarıyla 87C51'in uyumlu çalışması sonucu yerine getirmektedir. Her bir parçanın çalışması, 87C51 tarafından sürekli denetlenir.



Şekil 2. Film yıkama makinası kontrol sistemi

Sisteme kullanıcı müdahalesi bir tuş takımı aracılığıyla yapılmaktadır. Bazı sistem parametrelerinde örneğin; banyo sıcaklıklarında, motor hızında, takviye miktarlarında kullanıcının belli sınırlar içinde değişiklikler yapabilmesine izin verilmiştir. Kullanıcı tuşları kullanarak değiştirmek istediği parametreyi seçer ve yine tuşlarla istediği arttırmayı ya da azaltmayı yapar. Tuş takımının kod çözme işlemini 74C922 entegre devresi gerçekleştirir. 74C922 basılan tuşun kodunu belirler ve bunu bir kesme üreterek mikrokontrolöre bildirir. Buna

karşılık mikrokontrolör de o tuşun fonksiyonunu değerlendirerek o an için uygunsu yerine getirir.

Sistemin otomatik açma kapama fonksiyonunu gerçekleştirebilmesi için bir gerçek zaman saatine ihtiyacı vardır. Bu işi ICM7170 entegre devresi yapar. Makina ilk çalıştırıldığında tuş takımı aracılığıyla 7170 bir kez gerçek zamana ayarlanır. Bundan sonra sistem kapatılsa bile pil sayesinde bu ayar kaybolmaz. Mikrokontrolör, gerektiğinde zamanı 7170'den okur ve yine gerektiğinde ayarlanan zamanı 7170'e yazar.

Makinada 6 adet bağımsız banyo tankı bulunmaktadır. Her birinin içinde bağımsız ısıtma rezistansı ve sıcaklık bilgisini alan sensör vardır. Bu sensörlerden alınan sıcaklık bilgisi, ICL7109 analog-dijital dönüştürücüsü tarafından 12 bit dijital sinyale çevrilir. Mikrokontrolör, belli aralıklarla banyo sıcaklıklarını kontrol ederek, sıcaklığı düşen banyonun rezistansını devreye alır. Kurutma bloğunun sıcaklığı da aynı şekilde kontrol edilir. 7109'un bir girişi vardır. Banyo tanklarının ve kurutma bloğunun toplam 7 adet sıcaklık sensörü (LM35), bir analog çoğullayıcı olan 4051 entegresi tarafından sırayla 7109'un girişine bağlanarak sıcaklık bilgilerinin mikrokontrolöre ulaşması sağlanır. Mikrokontrolör 4051'i, uygun sensörü 7109'a bağlamak için her ölçümden sonra yeniden adresler. Böylece bütün sensörlerin bilgisi sırayla okunur.

Makinanın çalışması esnasında, o anki durumuyla ilgili bilgiler ve istatistikler bir DMC20483 dot matrix likid kristal göstergede gösterilir. DMC20483 bir modüldür. Kendi üzerinde göstergelyi sürmek için sürücüler ve karakter jeneratörü mevcuttur. 4 adet 20'şer karakterlik satırı vardır. Mikrokontrolör göstergelye bir veri yazmak istediğinde, yazılacak karakterin pozisyonunu ve kodunu

gönderir. Bu karakteri gösterme işini DMC20483 otomatik olarak yapar. Mikrokontrolörün yapması gereken sadece, aynen bir standart RAM hafızaya bilgi yazar gibi, yazılacak karakterin ASCII kodunu göstergeye yazmaktır. Bunun için önce DMC20481'in yazma girişini aktif (lojik 0) yapar ve hemen ardından kodu gönderir. Böylece 1 karakterlik bilgi göstergede gösterilmiş olur. Daha uzun bilgiler için bu işlem gereği kadar tekrarlanır.

Makinaya verilen film, merdane sistemi yardımıyla makinanın içinde ilerler ve banyolara girer-çıkır. Bu merdaneler bir 12V'luk DC motor tarafından döndürülür. Bu motorun dönüş hızı direkt olarak filmin banyolarda kalma süreleriyle ilgilidir. Bu yüzden çok hassas olarak kararlı olması istenir. DC motorun kontrolü işini HCTL1100 entegre devresi yapar. Motor hızının olması gereken değerini HCTL1100'e mikrokontrolör gönderir. Bundan sonra HCTL1100 motora bu hızda dönmesi için gereken PWM darbelerini otomatik olarak gönderir. Motorun herhangi bir andaki hızını, mile bağlı olan bir enkoder sürekli olarak entegreye geri besler. Böylece hassas bir devir regülasyonu sağlanır.

Makinanın kullanımı esnasında kullanıcıya yapılması gereken uyarılar 2 adet 256KBaytlık 27C020 EPROM'a kaydedilmiştir. Kayıtlar, 32KBaytlık modüller halinde yapılmıştır. Mikrokontrolörün programında bütün kayıtların adresleri mevcuttur. EPROM'ların çıkışına bir dijital-analog dönüştürücü ile ses frekans amplifikatörü bağlanmıştır. Bir uyarı mesajı verilmesi gerektiğinde mikrokontrolör, EPROM'ların ilgili adreslerini adresleyerek, hoparlörden ses olarak duyulmasını sağlar.

3.3. Devrenin Yapısı ve Çalışması

Elektronik devrede kullanılan bütün elemanlar, üretici kataloglarında belirtilen standart bağlantı şekillerinde bağlanmıştır. Bunların birbirleriyle ilgileri genel sistem organizasyonu bölümünde açıklanmıştır. Sistemi oluşturan elektronik devrenin şeması ekte verilen Şekil-3.a, Şekil-3.b ve Şekil-3.c'de gösterilmiştir.

87C51, sistemin beynidir. Şemadan da görüleceği üzere diğer bütün entegrelerin kontrol hatları 87C51'e girmektedir. Bütün veri hatları paralel olarak bağlıdır. Herhangi bir anda hangi hatların aktif olacağını ve veri alışverişinde kullanılacağını mikrokontrolör bu kontrol hatlarını kullanarak belirler. 87C51'in 4 adet olan 8'er bitlik arabirimlerinden birisi bu kontrol hatlarına ayrılmıştır. Bunlar çipin 21'den 29'a kadar olan ayaklarıdır. Bir diğer arabirim olan 32'den 40'a kadar olan ayaklar da veri giriş-çıkışı için ayrılmıştır. 8 bitlik bütün okuma ve yazmalar bu arabirim aracılığıyla yapılır. 40 nolu ayak çipin pozitif besleme girişidir. Devredeki bütün elemanlar gibi 87C51 de 5V'luk sabit gerilimle beslenir. 20 nolu ayak da beslemenin negatif girişidir. 18 ve 19 nolu ayaklar çipin çalışmasını sağlayan kristalin bağlandığı uçlardır. Kristal, 10.5MHz frekansta titreşir. Yük kapasitesi olarak birer adet 27pf değerinde kondansatör bağlanmıştır.

12 ve 13 nolu ayaklar mikrokontrolörün harici kesme girişleridir. Bu kesme girişlerinden birisi zamanlamayı kontrol eden ICM7170'e, diğeri de tuş takımını kontrol eden 74C922'ye ayrılmıştır. Bu girişlerdeki lojik sinyaller normalde yüksek seviyede durur. Kesmenin oluşması için bunlardan birisinin düşük seviyeye inmesi gerekir. 9 nolu ayak reset girişidir. Devreye ilk enerji verildiğinde mutlaka bu girişe düşük seviyede bir reset sinyali uygulanmalıdır. Programın her zaman sıfırcı

adresten başlaması için bu gereklidir. Otomatik reset sinyali üretme işini bu ayağa bağlı olan $10\mu\text{f}$ değerindeki kondansatör ile $8.2\text{K}\Omega$ değerindeki direnç yapar.

Tuş takımını kontrol eden 74C922 entegresi 4×4 tuş matrisini okuyabilecek yapıda tasarlanmıştır. Fakat devrede bu matrisin 10 tuşu kullanılmıştır. 74C922'nin besleme gerilimi 18 ve 9 nolu ayaklardan uygulanır. Tuş takımının taranması için gereken frekansı 5 nolu ayağına bağlı olan 10nf 'lık kondansatör ile entegre kendisi üretir. 1, 2, 3, 4, 8, 10,11 nolu ayaklar tuş matrisi için ayrılmıştır. Devredeki bağlantı şekliyle $3\times 4=12$ tuş taranabilir. Tuşa basılması esnasında titremeden ya da kötü kontaklamadan kaynaklanabilecek parazitleri engellemek için, bu amaçla ayrılmış olan 6 nolu ayağa 100nf 'lık bir kondansatör bağlanmıştır. Herhangi bir tuşa basıldığında, o tuşu ifade eden 4 bitlik dijital bilgi üretilir. Tuşa basılmasıyla 12 nolu ayaktaki normalde düşük seviyede duran sinyal yüksek seviyeye çıkar. Bu ayak 87C51'in kesme girişlerinden birine bağlıdır. Tuşun bırakılmasıyla bu ayaktaki sinyal düşük seviyeye iner ve bir kesme üretir. Kesme isteğine cevap veren mikrokontrolör 13 nolu, entegrenin veri çıkışının tri-state kontrol ucunu aktifleştirerek üretilen tuş bilgisinin veri yoluna çıkmasını sağlar. Veri yoluna çıkan 4 bitlik tuş bilgisi mikrokontrolör tarafından derhal okunur ve değerlendirilir. Görüldüğü gibi tuşa basmak değil bırakmak mikrokontrolörü uyarmaktadır. Hangi tuşa karşılık hangi işlerin yapılacağı programda belirtilmiştir.

ICM7170 devrede, besleme kesilse dahi piliyle sürekli çalışan elektronik bir saat görevi yapmaktadır. ICM7170'in besleme gerilimi, 23 ve 13 nolu ayaklardan uygulanır. 9 ve 10 nolu ayaklar osilatör giriş uçlarıdır. Entegrenin 2.097152MHz 'lik özel kristali bu uçlara şemada görüldüğü gibi bağlanır. Kristale bağlı diğer kondansatörler yük olarak kullanılmıştır. ICM7170'e yazma ve okuma işlemleri ALE, CS, RD, WR hatları kullanılarak yapılır. Okuma ve yazma, standart

bir RAM hafızada olduğu gibidir. 5 bitlik adres yolu ve 8 bitlik veri yolu bağımsız olarak çip dışına çıkarılmıştır. ALE hattı sayesinde bunların çoğullamalı olarak kullanılması mümkündür. ALE hattı yüksek seviyede iken adres hattı, düşük seviyede iken veri hattı aktiftir. Her iki veri yolu da tri-state tasarlandığından bu bağlamda herhangi bir problem çıkmaz. ALE hattına uygulanacak sinyalleri direkt olarak mikrokontrolör üretir.

ICM7170'in içinde bir alarm hafızası mevcuttur. Mikrokontrolör, alarmın istendiği zamanı detaylarıyla çipe yazar. Bundan sonra, programlanan bu zamanla, sürekli ilerleyen gerçek zaman eşit olduğunda 7170 bir kesme üretir. Entegrenin kesme çıkışı direkt olarak 87C51'in ikinci kesme girişine bağlı olduğundan bu kesme derhal işleme konur. Bu alarm hafızası, otomatik açılma ve kapanma fonksiyonlarını yerine getirmek için kullanılır. 14 nolu ayak batarya giriş ucudur. Sistemin enerjisi kesilse dahi saatin çalışması gerektiğinden, buraya bağlanan 3V'luk batarya sayesinde entegre devre sürekli çalışır.

Makinaya verilen filmlerin makina içinde ilerlemesini sürekli dönen merdane sistemi sağlamaktadır. Bu merdaneleri 12V'luk, 15W gücünde, statoru sabit mıknatıslı bir DC motor çevirir. Daha önce de belirtildiği gibi bu motorun hızının sabit tutulması istenmektedir. Bu işi HCTL1100 entegre devresi yapar. Mikrokontrolör, motorun olması gereken hızını HCTL1100'ün dahili kaydedicisine yazar. Bundan sonra HCTL1100, her 1ms'de motorun miline bağlı olan optik kod çözücünden gelen darbeleri sayarak uygun besleme darbelerini üretir. Motor PWM darbeleriyle beslenmektedir. Motorun yükü filmlerden dolayı arttığında yada besleme gerilimleri düştüğünde, darbelerin genişlikleri artırılarak veya tersi olduğunda darbelerin genişlikleri azaltılarak devir regülasyonu sağlanır. Motoru süren PWM darbe sinyalleri 16 nolu ayaktan alınır. ALE, R/W, CS, OE ayakları, aynen ICM7170'de

olduđu gibi görev yaparlar. Mikrokontrolör gerektiğinde bu hatları kullanarak HCTL1100'e ulařır. Entegre devrenin çalışması için gereken osilatör sinyalleri şemada görülen 4060 tarafından üretilir ve 34 nolu ayaktan entegreye girer. Besleme, 11 ve 35 nolu ayaklar ile 10 ve 32 nolu ayaklardan uygulanır. 6 bitlik adres yolu ile 8 bitlik veri yolu çipin içerisinde birleştirilerek beraberce 2'den 10'a kadar olan ayaklarala dışarı çıkarılmıştır. Çoğullamalı çalışma için yine ICM7170'de olduđu gibi ALE hattı kullanılır. 36 nolu ayak entegrenin reset girişidir. Sistem her enerjilendiğinde entegre resetlenmelidir. Bunu otomatik olarak bu ayađa bađlı olan 8.2K Ω 'luk direnç ile 10 μ F'lık kondansatör yapar.

Makinada bulunan 6 adet bađımsız banyo tankının belli sıcaklıklarda sabit tutulması gerekmektedir. 87C51, her banyoyu gerektiğinde ısıtarak bunu sađlar. Bunu yapabilmesi için, 87C51'in sürekli olarak banyo sıcaklıklarını ölçmesi gerekir. Dijital sıcaklık ölçümü işini 12 bitlik bir analog-dijital dönüřtürücü olan ICL7109 yapar. 6 tankın herbirinin içinde birer adet LM35 sıcaklık sensörü bulunur. LM35, °C başına 10mV gerilim çıkışı verir. Sıcaklıkla dođru orantılı olarak artan bu gerilim, 7109 tarafından 12 bitlik dijital bilgiye çevrilir. 6 adet banyo sensörünün herbiri, 4051 analog çođullayıcı entegresi tarafından sırayla 7109 girişine bađlanır. Her bir ölçüm yaklaşık 150ms sürer. Ölçüm sonunda 7109, sonucun hazır olduđunu belirten bir sinyali 87C51'e gönderir. Sinyali alan mikrokontrolör, 12 bit sıcaklık bilgisinin önce ilk 8 bitini, daha sonra da kalan 4 bitini okur. Bu sıcaklık olması gerekenle karşılaştırılarak, ilgili banyo ısıtıcısının durumuna karar verilir.

7109'a analog giriş sinyali, 35 ve 34 nolu ayaklar arasından uygulanır. Buraya bađlı olan 1M Ω 'luk direnç ile 10nF'lık kondansatör, yüksek frekanslı parazitik sinyalleri süzerler. 7109, çift meyilli integral alma prensibine göre çalışan bir analog-dijital dönüřtürücüdür. İntegral alma ile ilgili devre elemanları 30,

31 ve 32 nolu ayaklara bağlıdır. Analog dijital dönüştürücüler, bir analog sinyali dijitalle çevirirken referans gerilimine ihtiyaç duyarlar. Bu referans gerilimi LM336 2.5V'luk referans kaynağı tarafından sağlanır ve kalibrasyon için kullanılan bir ayarlı direnç üzerinden referans girişi olan 36 nolu ayağa uygulanır. Dönüşüm için gereken saat darbeleri, 22 ve 23 nolu ayaklar arasına bağlı olan kristal ile çipin içinde üretilir. Besleme gerilimi, 40 ve 1 nolu ayaklar arasından uygulanır. Referans gerilimine binebilecek parazitleri engellemek için 37 ve 38 nolu ayaklar arasına 10 μ F'lık bir kondansatör bağlanmıştır.

12 bitlik sonuç bilgisi, 5'den 17'ye kadar olan ayaklar olarak bağımsız şekilde çip dışına çıkarılmıştır. ICM7170 ve HCTL1100'de olduğu gibi çoğullamalı çalışmaya imkan tanımak için ilk 8 bitin ve son 4 bitin üç-konumlu (tri-state) kontrol uçları 18 ve 19 nolu ayaklarla kontrol edilir. 4 nolu ayak taşma (overflow) bitidir. Ölçülen sıcaklık 40.96°C'yi aştığında bu bit lojik-1 olur. Entegre devrenin bu sistemde kullanılmayan ayaklara uygun pasif lojik seviyelere bağlanmıştır.

Makinanın durumuyla ilgili, kullanıcıya yapılması gereken uyarılar 2 adet 27C020 256KBaytlık EPROM hafızaya dijitize edilmiş sesler olarak kaydedilmiştir. Bu sesler, 32KBaytlık parçalar halinde hafızaya yazılmışlardır. Bunlardan birisine ulaşılmak istendiğinde, EPROM'ların A15, A16, A17 adres girişleri, uygun şekilde adreslenir. Bu 3 bit adresleme toplam 8 adet ses modülünden birini seçebilir. Ayrıca iki EPROM'un CS girişleri de 4. seçme girişi olarak kullanılır. Böylece 2 entegre devrede toplam 16 adet ses modülü seçilebilir. A0'dan A15'e kadar olan adres hatları ortaktır. Mikrokontrolörden ses izni geldiğinde, 14 bitlik bir sayıcı olan 4020 entegresi, bu adresleri tarayarak 32KBaytlık hafıza bölgesindeki tüm dijital kaydın çıkışa ulaşmasını sağlarlar. Tarama bittiğinde, sayıcı mikrokontrolör tarafından resetlenir. 4060 entegresi burada, tarama için gereken osilasyonu sağlar. EPROM'lar

tri-state çıkışlı olduklarından, çıkışları paralel bağlıdır fakat yalnızca birisi aktiftir. Dijital ses bilgileri, DAC800 dijital-analog çevirici tarafından gerçek elektriksel ses sinyallerine dönüştürülürler. DAC800'ün 4 nolu çıkış ayağından alınan analog ses sinyali, LM380 ses frekans amplifikatörü tarafından yükseltilerek hoparlörden ses olarak alınır. Burada görülen 74373, 87C51'den bildirilen ses koduna kilitlenir ve ilgili ses parçasının seçilmesini sağlar.

BÖLÜM 4. PROGRAM

Bir önceki bölümde anlatılan elektronik devre, daha önce de belirtildiği gibi 87C51 mikrokontrolör kontrolünde çalışmaktadır. 87C51 ise, aşağıdaki program denetiminde çalışır. 87C51'in komutları ve işlevleri daha önce tanıtılmıştı. Bu programda bu basit komutların kombinasyonları yer almaktadır. Mikrokontrolörün kontrol etmesi gereken her birimle ilgili bağımsız bir program rutini vardır. Bunların bölüm başlıkları program içinde belirtilmiştir.

Programda iki adet kesme rutini bulunmaktadır. Bunlardan birisi tuş takımını kontrol eden 74C922'ye, diğeri de zamanlamayı kontrol eden ICM170'e ayrılmıştır. Ana program içinde çok kullanılan program rutinleri, programın son bölümüne yazılmıştır. Ayrıca orijinalde Türk alfabesinden ğ,ş,ç gibi harfleri olmayan sıvı kristal gösterge modülü DMC20483'e bu harfleri programlamak için gereken datalar programın sonunda yer almaktadır.

```
CPU "8052.TBL"
HOF "INT8"
```

```
ORG 00H
sjmp PROGRAM
```

```
ORG 03H
ljmp TIMER
```

```
ORG 13H
ljmp TUS
```

```
*****
,
*****
```

```
ORG 30H
PROGRAM: mov p2,#10111110b ;Peripherallerin start pozisyonu.
          mov ista,#11111000b ;ADC bolumu start durumu.
          mov p0,ista
          setb p1_1
          clr p1_1
          mov sp,#30H
          lcall Dispres ;Display'i resetle.
          lcall Timerstart ;7170'i hazirla.
          lcall Motres ;Motor kontrolu hazirla.
          mov TCON,#00000101b
          mov IE,#10000101b
          mov dista,#10000000b
          Kontrol:jb psw_5,Kon1
                jb dista_7,Tazele
                jnb p3_1,che7109
                setb genreg_0
                sjmp Kontrol
          che7109:jnb genreg_0,Kontrol
                clr genreg_0
                ljmp ISI
          Tazele:mov flasta,#0
                lcall CLS
                clr dista_7
                mov a,dista
                anl a,#00011111b
                mov b,#3
                mul ab
                mov dptr,#sicra
                jmp @a+dptr
          sicra: ljmp Anaekran ;0
                ljmp Zamanset ;1
                ljmp TimerProg ;2
                ljmp ISIsset1 ;3
                ljmp ISIsset2 ;4
                ljmp Motkont ;5
```

Kon1: ljmp Tus6

```
Anaekran:  mov  dptr,#POLYFORM
            mov  b,#134
            lcall  Mesaj
            ljmp  Kontrol
```

```
Zamanset:  mov  dptr,#ZamAy ;Saat&Tarih Set ekrani.
            lcall  yeniharf
            mov  p2,#10111100b
            mov  r0,#01h           ;Saat okuma.
            lcall  timeroku
            mov  r1,a
            inc  r0           ;Dakika okuma.
            lcall  timeroku
            mov  r2,a
            mov  r0,#04H       ;Ay okuma.
            lcall  timeroku
            mov  r3,a
            inc  r0           ;Gun okuma.
            lcall  timeroku
            mov  r4,a
            inc  r0           ;Yil okuma.
            lcall  timeroku
            mov  r5,a
            inc  r0           ;Haftanin gununu okuma.
            lcall  timeroku
            mov  r6,a
            setb p2_1
            mov  a,r1
            mov  b,#199
            lcall  Decimal
            mov  a,#":"
            lcall  dispdat
            mov  a,r2
            lcall  Decibel
            mov  a,r4
            mov  b,#154
            lcall  decimal
            mov  a,#"/"
            lcall  dispdat
            mov  a,r3
            lcall  decibel
            mov  a,#"/"
            lcall  dispdat
            mov  a,r5
            lcall  decibel
            mov  a,r6
            lcall  Gunyaz
```

```

        mov    b,#218
        lcall  Mesaj
        ljmp   Kontrol
timeroku:clr    p3_7
        setb   p3_7
        movx   a,@r0
        ret
;*****
TimerProg:  mov    b,#128      ;Timer Programlama ekranı.
            mov    dptr,#TimerProg
            lcall  Mesaj
            mov    b,#150
            mov    dptr,#Acilis
            lcall  Mesaj
            mov    b,#214
            mov    dptr,#Kapanis
            lcall  Mesaj
            mov    a,setno
            jz     scpc
            dec   a
            mov   b,#4
            div   ab
scpc:       lcall  Gunyaz
            mov   b,#197
            lcall  Mesaj
sfpc:      mov   a,setno
            jnz   sgpc
            inc   a
sgpc:      dec   a
            mov   b,#4
            div   ab
            mov   b,#4
            mul   ab
            mov   r0,a
            inc   a
            subb  a,setno
            jz    shpc
            mov   p2,#00111110b
            movx  a,@r0
            setb  p2_7
            sjmp  sipc
sacma:     mov   b,#160
            mov   dptr,#iptal
            lcall  Mesaj
            inc   r0
            sjmp  srpc
shpc:     mov   a,flashmem+1
sipc:     mov   r1,a
            clr   c

```



```

    subb    a,#24
    jnc     sacma
    mov     a,r1
    mov     b,#160
    lcall   decimal
    mov     a,#":"
    lcall   dispdat
    inc     r0
    mov     a,r0
    inc     a
    subb    a,setno
    jz      sjpc
    mov     p2,#00111110b
    movx    a,@r0
    setb    p2_7
    sjmp    skpc
sjpc:  mov     a,flashmem+1
skpc:  lcall   decibel
srpc:  inc     r0
       mov     a,r0
       inc     a
       subb    a,setno
       jz      slpc
       mov     p2,#00111110b
       movx    a,@r0
       setb    p2_7
       sjmp    smpc
sacma2: mov     b,#224
       mov     dptr,#iptal
       lcall   Mesaj
       sjmp    sapc
slpc:  mov     a,flashmem+1
smpc:  mov     r1,a
       clr     c
       subb    a,#24
       jnc     sacma2
       mov     a,r1
       mov     b,#224
       lcall   decimal
       mov     a,#":"
       lcall   dispdat
       inc     r0
       mov     a,r0
       inc     a
       subb    a,setno
       jz      snpc
       mov     p2,#00111110b
       movx    a,@r0
       setb    p2_7

```

```

        sjmp  sopc
snp:   mov   a,flashmem+1
sopc:  lcall decibel
sapc:  ljmp  Kontrol
*****
*****
ISIsset1:  mov   b,#134
           mov   dptr,#ISIPanel
           lcall  Mesaj
           mov   b,#192
           mov   dptr,#Developer
           lcall  Mesaj
           mov   r1,#0
           lcall  isiyaz
           mov   b,#148
           mov   dptr,#Bleach
           lcall  Mesaj
           mov   r1,#2
           lcall  isiyaz
           mov   b,#212
           mov   dptr,#Fixer
           lcall  Mesaj
           mov   r1,#4
           lcall  isiyaz
           lcall  bufyaz
           ljmp  Kontrol
ISIsset2:  mov   b,#128
           mov   dptr,#Stabilizer1
           lcall  Mesaj
           mov   r1,#6
           lcall  isiyaz
           mov   b,#192
           mov   dptr,#Stabilizer2
           lcall  Mesaj
           mov   r1,#8
           lcall  isiyaz
           mov   b,#148
           mov   dptr,#Stabilizer3
           lcall  Mesaj
           mov   r1,#10
           lcall  isiyaz
           mov   b,#212
           mov   dptr,#Kurutma
           lcall  Mesaj
           mov   r1,#12
           lcall  isiyaz
           lcall  bufyaz
           ljmp  Kontrol
bufyaz:  mov   b,flashx

```

```

        lcall    dispkom
        mov     a,flashmem+1
        mov     b,setno
        jb     b_0,bufyaz1
        orl    a,#00110000b
        lcall    dispdat
        ret
bufyaz1:lcall    decibel
        ret
isiyaz: mov     a,#28
        add     a,r1
        mov     r0,a
        clr     p2_7
        movx    a,@r0
        setb    p2_7
        lcall    decibel
        mov     a,#"."
        lcall    dispdat
        inc     r0
        clr     p2_7
        movx    a,@r0
        setb    p2_7
        orl    a,#00110000b
        lcall    dispdat
        mov     a,#" "
        lcall    dispdat
        lcall    dispdat
        mov     a,#isilar
        add     a,r1
        mov     r0,a
        mov     a,@r0
        lcall    decibel
        mov     a,#"."
        lcall    dispdat
        inc     r0
        mov     a,@r0
        orl    a,#00110000b
        lcall    dispdat
        ret

```

```

;*****
;*****

```

```

Motkont:  mov     dptr,#MotorKont
          lcall    Mesaj
          ljmp     Kontrol

```

```

;*****
;*****

```

```

TIMER:   push     p2
          push     acc
          push     b0r0

```

```

push b0r1
push b0r2
push b0r3
push dpl
push dph
push b
push b2r0
push b3r1
push psw
setb p3_6
setb p3_7
mov p2,#10111100b
mov r0,#10h
lcall timeroku
mov kesme,a
jb acc_0,kapkes
lcall Yuzms
lcall Birs
setb p2_1
pop psw
pops: pop b3r1
pop b2r0
pop b
pop dph
pop dpl
pop b0r3
pop b0r2
pop b0r1
pop b0r0
pop acc
pop p2
reti
Yuzms: jb kesme_2,Flash
ret
kapkes: pop psw
setb psw_5
setb p2_1
ljmp pops
;
*****
Flash: jb dista_6,flash2
ret
fladon: inc flasta
ret
flash2: setb p2_1
jb flasta_1,flash1
jb flasta_0,fladon
inc flasta
mov r0,flashuz

```

```

        mov    r1,#flashmem
        mov    b,flashx
        lcall  dispkom
fla:    lcall  busytest
        setb   p2_6           ;Displayden okuma.
        setb   p2_0
        movx   a,@r0
        clr    p2_0
        clr    p2_6
        mov    @r1,a         ;Datayi yaz.
        dec    r1           ;Bir sonraki icin adres.
        djnz   r0,fla
        mov    b,flashx
        lcall  dispkom
        mov    r0,flashuz
        mov    a,#32
sil:    lcall  dispdat
        djnz   r0,sil
        ret
flash1: inc    flasta
        mov    b,flashx
        lcall  dispkom
        mov    r0,flashuz
        mov    r1,#flashmem
flas:   mov    a,@r1
        lcall  dispdat
        dec    r1
        djnz   r0,flas
        ret

```

```

,*****
*****

```

```

Birs:  jb     kesme_3,zaman1
        ret
zaman1: mov    a,dista
        clr    acc_5
        jz     zaman2
        mov    a,dista
        clr    c
        subb   a,#5
        jz     zaman3
        ret
zaman3: setb   p2_1
        lcall  hizyaz
        ret
zaman2: mov    r0,#01h
        lcall  timeroku
        mov    r1,a
        inc    r0
        lcall  timeroku

```

```

mov r2,a
inc r0
lcall timeroku
mov r3,a
setb p2_1
mov b,#198
mov a,r1
lcall Decimal
mov a,#":"
lcall dispdat
mov a,r2
lcall decibel
mov a,#":"
lcall dispdat
mov a,r3
lcall decibel
ret

```

```

.*****
,
*****

```

```

ISI:  clr p2_4
      movx a,@r0 ;Low byte.
      setb p2_4
      mov b2r4,a
      mov a,ista
      clr acc_4
      mov p0,a ;High byte.
      setb p1_1
      clr p1_1
      clr p2_4
      movx a,@r0
      setb p2_4
      mov b2r5,a
      anl b2r5,#00001111b ;Son 4 biti temizle
      mov b2r0,#0 ;Degiskenleri sifirla
      mov b2r1,#0 ;
      mov b2r2,#0 ;
      mov b2r3,#0 ;
      mov b2r6,#0
      mov a,b2r5
      jnb acc_3,aaa1 ;12. biti isle
      mov b2r3,#2 ;
      mov b2r6,#48 ;
aaa1: jnb acc_2,aaa2 ;11. biti isle
      inc b2r3 ;
      mov a,b2r6 ;
      add a,#24 ;
      mov b2r6,a ;
      mov a,b2r5 ;
aaa2: jnb acc_1,aaa3 ;10. biti isle

```

```

        mov    b2r2,#5          ;
        mov    a,b2r6          ;
        add    a,#12           ;
        mov    b2r6,a          ;
        mov    a,b2r5          ;
aaa3:   jnb    acc_0,aaa4      ;9 biti isle
        mov    a,b2r2          ;
        add    a,#2            ;
        mov    b2r2,a          ;
        mov    a,b2r6          ;
        add    a,#56           ;
        mov    b2r6,a          ;
aaa4:   mov    a,b2r6          ;
        mov    b,#100          ;b2r6'daki artigi temizle
        div    ab              ;
        jz     aaaa            ;
        inc    b2r2            ;
aaaa:   mov    a,b              ;
        mov    b,#10           ;
        div    ab              ;
        mov    b2r1,a          ;
        mov    b2r0,b          ;
        mov    a,b2r4          ;Low byte'i isle
        mov    b,#100          ;
        div    ab              ;
        add    a,b2r2          ;
        mov    b2r2,a          ;
        mov    a,b              ;
        mov    b,#10           ;
        div    ab              ;
        add    a,b2r1          ;
        mov    b2r1,a          ;
        mov    a,b              ;
        add    a,b2r0          ;
        mov    b2r0,a          ;
        mov    b,#10           ;Son temizlik
        div    ab              ;
        mov    b2r0,b          ;
        add    a,b2r1          ;
        mov    b,#10           ;
        div    ab              ;
        mov    b2r1,b          ;
        add    a,b2r2          ;
        mov    b,#10           ;
        div    ab              ;
        mov    b2r2,b          ;
        add    a,b2r3          ;
        mov    b2r3,a          ;
        mov    a,ista          ;

```

```

anl    a,#00000111b
add    a,acc
add    a,#isilar
mov    r0,a
mov    a,b2r3
mov    b,#10
mul    ab
add    a,b2r2
mov    @r0,a
inc    r0
mov    @r0,b2r1
lcall  Termostat
mov    a,ista
anl    a,#00000111b
inc    a
mov    r0,a
clr    c
subb   a,#7
jnz    aaa7
mov    r0,#0
aaa7:  orl    b0r0,#11111000b
mov    ista,r0
mov    a,ista
clr    acc_3
mov    p0,a           ;Low byte.
setb   p1_1
clr    p1_1
ljmp   Kontrol

```

```

,*****
*****

```

```

Termostat:  mov    a,ista
anl    a,#00000111b
add    a,acc
add    a,#28
mov    r0,a
clr    p2_7
movx   a,@r0
mov    b,#10
div    ab
mov    b2r6,a
mov    b2r5,b
inc    r0
movx   a,@r0
setb   p2_7
mov    b2r4,a
clr    c           ;
mov    a,b2r3     ;10'lar hanesi
subb   a,b2r6     ;
jz     aaa5       ;

```



```

        jc      isit      ;
        sjmp   sogut     ;
aaa5:   clr    c          ;
        mov   a,b2r2    ;1'ler hanesi
        subb  a,b2r5    ;
        jz    aaa6      ;
        jc    isit      ;
        sjmp  sogut     ;
aaa6:   clr    c          ;
        mov   a,b2r1    ;0.1'ler hanesi
        subb  a,b2r4    ;
        jc    isit      ;
sogut:  mov   r0,#01111111b
        mov   a,ista
        anl   a,#00000111b
        inc   a
        mov   r1,a
        mov   a,r0
aaa8:   rl    a
        djnz  r1,aaa8
;
        anl   p1,a
        mov   r2,#"- "
        sjmp  dene
isit:   mov   r0,#10000000b
        mov   a,ista
        anl   a,#00000111b
        inc   a
        mov   r1,a
        mov   a,r0
aaa9:   rl    a
        djnz  r1,aaa9
;
        orl   p1,a
        mov   r2,#"+ "
dene:   mov   a,dista
        anl   a,#00000111b
        mov   r0,a
        clr   c
        subb  a,#3
        jz    cyaz1
        mov   a,r0
        subb  a,#4
        jz    cyaz2
        ret
cyaz1:  mov   a,ista
        anl   a,#00000111b
        mov   r0,a
        clr   c
        subb  a,#3
        jc    aaa12

```

```

ret
cyaz2: mov a,ista
      anl a,#00000111b
      mov r0,a
      clr c
      subb a,#3
      jnc aaa12
      ret
aaa12: mov a,r0
      mov b,#5
      mul ab
      mov dptr,#aaa10
      jmp @a+dptr
aaa10: mov b,#204
      sjmp aaa11
      mov b,#160
      sjmp aaa11
      mov b,#224
      sjmp aaa11
      mov b,#140
      sjmp aaa11
      mov b,#204
      sjmp aaa11
      mov b,#160
      sjmp aaa11
      mov b,#224
aaa11: lcall dispkom
      mov a,b2r3
      orl a,#00110000b
      lcall dispdat
      mov a,b2r2
      orl a,#00110000b
      lcall dispdat
      mov a,#"."
      lcall dispdat
      mov a,b2r1
      orl a,#00110000b
      lcall dispdat
      mov a,#" "
      lcall dispdat
      lcall dispdat
      lcall dispdat
      mov a,r2
      lcall dispdat
      ret

```

```

.*****
,
*****

```

```

TUS:   push acc
       push dph

```

```

push    dpl
push    p2
push    psw
mov     p2,#10111010b
movx   a,@r0
setb   p2_2
anl    a,#00001111b
mov     b,#3
mul    ab
mov     dptr,#cc
jmp    @a+dptr
don:   pop    psw
       pop    p2
       pop    dpl
       pop    dph
       pop    acc
       reti
cc:    ljmp   TUS0      ;Saat&Takvim set tusu.
       ljmp   TUS1      ;On/Off Program tusu.
       ljmp   tus2
       ljmp   tus3
       ljmp   TUS4      ;Set + tusu.
       ljmp   TUS5      ;Set - tusu.
       ljmp   TUS6      ;STOP tusu.
       ljmp   tus7
       ljmp   TUS8      ;ISIsset tusu.
       ljmp   TUS9      ;OK tusu.
       ljmp   TUS10     ;Motor Kontrol tusu.

```

```

*****
,
*****

```

```

TUS0:  mov    a,dista      ;Saat&Tarih set tusu.
       anl    a,#00011111b
       jz     don00
       subb   a,#1
       jnz   don0
       inc   setno
       mov   a,setno
       mov   dista,#11000001b
       subb   a,#7
       jz     don00
       lcall  setler0
don0:  ljmp   don
don00: mov    dista,#10000001b;Timer set ekrani.
       mov    setno,#0
       sjmp  don0
setler0:mov   a,setno
       add   a,acc
       mov   dptr,#sets0-2
       jmp   @a+dptr

```

```

sets0: sjmp  setdak
        sjmp  setsaat
        sjmp  setgun
        sjmp  setay
        sjmp  setyil
        sjmp  sethafgun
setdak: mov  flashuz,#2
        mov  flashx,#202
        ret
setsaat:mov  flashuz,#2
        mov  flashx,#199
        ret
setgun:  mov  flashuz,#2
        mov  flashx,#154
        ret
setay:   mov  flashuz,#2
        mov  flashx,#157
        ret
setyil:  mov  flashuz,#2
        mov  flashx,#160
        ret
sethafgun:mov flashuz,#9
        mov  flashx,#218
        ret

```

```

*****
,
*****

```

```

TUS1:  mov  a,dista      ;On/Off Program tusu.
        anl  a,#00011111b
        jz   don10
        subb a,#2
        jnz  don1
        mov  r0,setno
        mov  a,r0
        jz   don12
        dec  r0
        mov  p2,#00111110b
        movx a,@r0
        setb p2_7
        subb a,flashmem+1
        jz   don11
        mov  p2,#00111110b
        mov  a,flashmem+1
        movx @r0,a
        setb p2_7
        mov  b,#100
        lcall waitms
don11: inc  r0
don12: mov  p2,#00111110b
        movx a,@r0

```

```

setb    p2_7
mov     flashmem+1,a
inc     setno
mov     a,setno
mov     dista,#11000010b
subb   a,#29
jz      don10
lcall  setler1
don1:   ljmp  don
don10:  mov   dista,#10000010b;On/Off program ekrani.
        mov   setno,#0
        sjmp  don1
setler1:mov  a,setno
        add   a,acc
        mov   dptr,#sets1-2
        jmp   @a+dptr
sets1:  sjmp  setacsa
        sjmp  setacda
        sjmp  setkpsa
        sjmp  setkpda
        sjmp  setacsa
        sjmp  setacda
        sjmp  setkpsa
        sjmp  setkpda
        sjmp  setacsa
        sjmp  setacda
        sjmp  setkpsa
        sjmp  setkpda
        sjmp  setacsa
        sjmp  setacda
        sjmp  setkpsa
        sjmp  setkpda
        sjmp  setacsa
        sjmp  setacda
        sjmp  setkpsa
        sjmp  setkpda
        sjmp  setacsa
        sjmp  setacda
        sjmp  setkpsa
        sjmp  setkpda
setacsa:mov  flashuz,#2
        mov   flashx,#160
        ret
setacda:mov  flashuz,#2
        mov   flashx,#163

```

```

ret
setkpsa:mov flashuz,#2
mov flashx,#224
ret
setkpda:mov flashuz,#2
mov flashx,#227
ret
.*****
,
*****
tus2: mov p2,#10111100b ;Reset tusu.
clr a
mov r0,#01h
movx @r0,a
mov r0,#02h
movx @r0,a
mov r0,#03h
movx @r0,a
setb p2_1
ljmp don
.*****
,
*****
tus3: ljmp don
.*****
,
*****
TUS4: setb dista_5 ;+ tusu.
lcall SET
setb dista_7
ljmp don
.*****
,
*****
TUS5: clr dista_5 ;- tusu.
lcall SET
setb dista_7
ljmp don
.*****
,
*****
TUS6: mov p2,#10111100b ;STOP tusu.
mov r0,#10h
movx a,@r0
mov a,#00000001b ;Yalniz alarm acik.
movx @r0,a
mov r0,#07h
movx a,@r0
setb p2_1
mov b,#4
mul ab
mov r0,a
clr p2_7
movx a,@r0

```

```

mov    r1,a
inc    r0
inc    r0
movx   a,@r0
clr    c
subb   a,r1
dec    r0
dec    r0
jc     bb1
jnz    bb2
inc    r0
movx   a,@r0
mov    r2,a
inc    r0
inc    r0
movx   a,@r0
clr    c
subb   a,r2
dec    r0
dec    r0
dec    r0
jc     bb1
bb2:   inc    r0
inc    r0
inc    r0
inc    r0
mov    a,r0
clr    c
subb   a,#28
jnz    bb4
mov    r0,#0
bb4:   movx   a,@r0
sjmp   bb3
bb1:   mov    a,r1
bb3:   clr    c
subb   a,#24
jz     aciptal
bb5:   inc    r0
movx   a,@r0
mov    r2,a
setb   p2_7
clr    p2_1
mov    a,r2
mov    r0,#0Ah
movx   @r0,a
dec    r0
mov    a,r1
movx   @r0,a
kapan: mov    p2,#10111110b

```

```

        clr    p1_7          ;Ana roleyi biraktir.
        sjmp   $
aciptal:mov  r0,#10h
        clr    a
        movx  @r0,a
        movx  a,@r0
        sjmp  kapan

```

```

;*****
;*****

```

```

tus7:  ljmp   don

```

```

;*****
;*****

```

```

TUS8:  mov    a,dista      ;ISI set tusu.

```

```

        anl    a,#00011111b

```

```

        jz     don20

```

```

        mov    r0,a

```

```

        subb   a,#3

```

```

        jz     asil

```

```

        mov    a,r0

```

```

        subb   a,#4

```

```

        jnz    don2

```

```

asil:  mov    a,setno

```

```

        add    a,#27

```

```

        mov    r0,a

```

```

        mov    a,setno

```

```

        jz     don21

```

```

        mov    p2,#00111110b

```

```

        movx  a,@r0

```

```

        setb   p2_7

```

```

        subb   a,flashmem+1

```

```

        jz     don21

```

```

        mov    p2,#00111110b

```

```

        mov    a,flashmem+1

```

```

        movx  @r0,a

```

```

        setb   p2_7

```

```

        mov    b,#100

```

```

        lcall  waitms

```

```

don21: inc    r0

```

```

don22: mov    p2,#00111110b

```

```

        movx  a,@r0

```

```

        setb   p2_7

```

```

        mov    flashmem+1,a

```

```

        inc   setno

```

```

        mov    a,setno

```

```

        clr    c

```

```

        subb   a,#7

```

```

        jnc   don23

```

```

        mov    dista,#11000011b

```

```

don24: lcall  setler2

```



```

don2: ljmp    don
don23: mov    a,setno
      subb   a,#15
      jz     don20
      mov    dista,#11000100b
      sjmp   don24
don20: mov    dista,#10000011b;ISI set ekrani.
      mov    setno,#0
      sjmp   don2
setler2:mov   a,setno
      add   a,acc
      mov   dptr,#sets2-2
      jmp   @a+dptr
sets2:  sjmp  set2xx
      sjmp  set2x
      sjmp  set3xx
      sjmp  set3x
      sjmp  set4xx
      sjmp  set4x
      sjmp  set1xx
      sjmp  set1x
      sjmp  set2xx
      sjmp  set2x
      sjmp  set3xx
      sjmp  set3x
      sjmp  set4xx
      sjmp  set4x
set1xx: mov    flashuz,#2
      mov    flashx,#134
      ret
set1x:  mov    flashuz,#1
      mov    flashx,#137
      ret
set2xx: mov    flashuz,#2
      mov    flashx,#198
      ret
set2x:  mov    flashuz,#1
      mov    flashx,#201
      ret
set3xx: mov    flashuz,#2
      mov    flashx,#154
      ret
set3x:  mov    flashuz,#1
      mov    flashx,#157
      ret
set4xx: mov    flashuz,#2
      mov    flashx,#218
      ret
set4x:  mov    flashuz,#1

```

```

        mov  flashx,#221
        ret
;
;*****
TUS9:  mov  p2,#10111100b      ;OK tusu.
        lcall kaprog
        mov  dista,#10000000b
        mov  setno,#0
        ljmp don
;
;*****
tus10: mov  dista,#10000101b;Motor set tusu.
        ljmp don
;
;*****
SET:   mov  a,setno
        juz  setkont
        ret
setkont:mov a,dista
        anl  a,#00011111b
        mov  dptr,#setekad-3
        mov  b,#3
        mul  ab
        jmp  @a+dptr
setekad:ljmp settim
        ljmp setonf
        ljmp setisi
        ljmp setisi
        ret
settim: mov  a,setno
        mov  dptr,#settim0-4
        mov  b,#4
        mul  ab
        jmp  @a+dptr
settim0:mov r0,#02H
        sjmp settimx
        mov  r0,#01H
        sjmp settimx
        mov  r0,#05H
        sjmp settimx
        mov  r0,#04H
        sjmp settimx
        mov  r0,#06H
        sjmp settimx
        mov  r0,#07H
settimx:mov p2,#10111100b
        lcall timeroku
        inc  a
        jb   dista_5, settimxa

```

```

        subb    a,#2
settimxa:movx  @r0,a
        setb    p2_1
        ret
setonf: inc    flashmem+1
        jb     dista_5,setonf2
        dec    flashmem+1
        dec    flashmem+1
        mov    a,flashmem+1
        subb   a,#255
        jnz    setonf1
        mov    flashmem+1,#0
setonf1:ret
setonf2:mov   a,setno
        jnb   acc_0,setonf3
        mov   a,flashmem+1
        subb  a,#25
        sjmp  setonf4
setonf3:mov   a,flashmem+1
        subb  a,#60
setonf4:jnz   setonf1
        mov   flashmem+1,#0
        ret
setisi: inc   flashmem+1
        jb   dista_5,setisi2
        dec  flashmem+1
        dec  flashmem+1
        mov  a,flashmem+1
        subb a,#255
        jnz  setisi1
        mov  flashmem+1,#0
setisi1:ret
setisi2:mov   a,setno
        jnb   acc_0,setisi3
        mov   a,flashmem+1
        subb  a,#41
        sjmp  setisi4
setisi3:mov   a,flashmem+1
        anl   a,#00001111b
        subb  a,#10
setisi4:jnz   setisi1
        mov   flashmem+1,#0
        ret

```

```

*****
,
*****

```

```

Dispres:  clr    p3_6
          mov    b,#255      ;20ms bekle.
          lcall  waitms
          mov    a,#00111000b

```



```

mov r0,a
clr p2_7
movx a,@r0
setb p2_7
mov r1,a
subb a,#24
jz kapiptal
inc r0
clr p2_7
movx a,@r0
setb p2_7
mov r0,#0AH
clr p2_1
movx @r0,a
dec r0
mov a,r1
movx @r0,a
mov a,#00001101b
tms1: mov r0,#10H
movx @r0,a
movx a,@r0
setb p2_1
ret
kapiptal:clr p2_1
mov a,#00001100b
sjmp tms1

```

```

.*****
,
*****

```

```

Motres: mov p0,#05h
clr p1_2
setb p1_2
clr p3_6
mov p0,#03h
clr p2_3
setb p2_3
div ab
div ab
mov p0,#24H
clr p1_2
setb p1_2
mov p0,#255
clr p2_3
setb p2_3
div ab
div ab
mov p0,#23H
clr p1_2
setb p1_2
mov p0,#0

```

```

clr    p2_3
setb   p2_3
div    ab
div    ab
mov    p0,#0FH
clr    p1_2
setb   p1_2
mov    p0,#255
clr    p2_3
setb   p2_3
div    ab
div    ab
mov    p0,#22H
clr    p1_2
setb   p1_2
mov    p0,#64
clr    p2_3
setb   p2_3
div    ab
div    ab
mov    p0,#0
clr    p1_2
setb   p1_2
mov    p0,#00001011b
clr    p2_3
setb   p2_3
setb   p3_6
ret
hizyaz: mov    p0,#34H
clr    p1_2
setb   p1_2
clr    p2_3
setb   p2_3
div    ab
div    ab
clr    p1_3
movx   a,@r0
setb   p1_3
mov    r1,a
mul    ab
mul    ab
mov    p0,#09H
clr    p1_2
setb   p1_2
clr    p2_3
setb   p2_3
div    ab
div    ab
clr    p1_3

```

```

movx a,@r0
setb p1_3
mov r2,a
mul ab
mul ab
mov b,#148
lcall dispkom
mov a,r2
mov b,#100
div ab
orl a,#00110000b
lcall dispdat
mov a,b
lcall decibel
mov a,r1
mov b,#100
div ab
orl a,#00110000b
lcall dispdat
mov a,b
lcall decibel
mov p0,#22h
clr p1_2
setb p1_2
clr p2_3
setb p2_3
div ab
div ab
clr p1_3
movx a,@r0
setb p1_3
mov b,#100
div ab
orl a,#00110000b
lcall dispdat
mov a,b
lcall decibel
ret

```

```

,*****
*****

```

```

Mesaj:      lcall  dispkom
yeniharf:   clr    a
            movc  a,@a+dptr
            jz    tamam
            lcall dispdat
            inc  dptr
            sjmp yeniharf
tamam:      ret

```

```

*****
,
*****
busytest:   push   acc
            clr    p3_6
            setb   p2_6           ;Display okuma.
oku:        setb   p2_0
            movx   a,@r0
            clr    p2_0
            jb     acc_7,oku
            clr    p2_6
            setb   p3_6
            pop    acc
            ret

*****
,
*****
Decimal:    lcall   dispkom
decibel:    mov     b,#10         ;a'daki sayiyi BCD'ye cevir ve yaz.
            div    ab            ;Pozisyon:b
            orl    a,#00110000b
            orl    b,#00110000b
            lcall   dispdat
            mov     a,b
            lcall   dispdat
            ret

*****
,
*****
CLS:        mov     b,#1
dispkom:    lcall   busytest      ;b komutunu display'e gonder.
            clr    p3_6
            setb   p2_0
            mov    p0,b
            nop
            clr    p2_0
            setb   p3_6
            ret
dispdat:    lcall   busytest      ;a'yi display'e yaz.
            setb   p2_0
            mov    p0,a
            nop
            clr    p2_0
            ret

*****
,
*****
waitms:     push    b
ws1:        mov     r1,#25
            djnz   r1,$
            djnz   b,ws1
            pop    b
            ret

```



```

*****
,
*****

```

```

Gunyaz:  mov  dptr,#Guns
          mov  b,#4
          mul  ab
          jmp  @a+dptr
Guns:    mov  dptr,#Gunler
          ret
          mov  dptr,#Gunler+10
          ret
          mov  dptr,#Gunler+20
          ret
          mov  dptr,#Gunler+30
          ret
          mov  dptr,#Gunler+40
          ret
          mov  dptr,#Gunler+50
          ret
          mov  dptr,#Gunler+60
          ret

```

```

*****
*****

```

```

POLYFORM: EQU 0E00H
CGrad:     EQU POLYFORM+9
Gunler:    EQU CGrad+3
ZamAy:     EQU Gunler+70
TimProg:   EQU ZamAy+21
Acilis:    EQU TimProg+20
Kapanis:   EQU Acilis+9
iptal:     EQU Kapanis+9
ISlpanel:  EQU iptal+6
Developer: EQU ISlpanel+15
Bleach:    EQU Developer+7
Fixer:     EQU Bleach+7
Stabilizer1: EQU Fixer+7
Stabilizer2: EQU Stabilizer1+7
Stabilizer3: EQU Stabilizer2+7
Kurutma:   EQU Stabilizer3+7
MotorKont: EQU Kurutma+7
ORG 0E00H
DFB "POLYFORM",0
DFB 11011111b,"C",0
DFB "Pazartesi",0," Sal",1," ",0,8,"ar",3,"amba ",0
DFB "Per",3,"embe ",0," Cuma ",0,"Cumartesi",0," Pazar ",0
DFB "**** Zaman Ayar ****",0
DFB "* Otomatik ON/OFF *",0
DFB "A",7,1,"I",1,3,".",0
DFB "Kapan",1,3,":",0
DFB "--:--",0

```

```

DFB "Set",223,"C Is",1,223,"C R",0
DFB "Dev.: ",0,"Ble.: ",0,"Fix.: ",0
DFB "St.1: ",0,"St.2: ",0,"St.3: ",0,"Kur.: ",0
DFB "**** H",1,"z Kontrol ****",0

```

```

.*****
,
*****

```

```

HarfProg: EQU 0F00H
          ORG 0F00H
          DFB
            11100000b,11100000b,11101100b,11100100b,11100100b,11100100b,11101
110b,11100000b ;kucuk I
          DFB
            11100000b,11100000b,11110001b,11110001b,11110001b,11110011b,11101
101b,11100000b ;kucuk u
          DFB
            11100000b,11100000b,11101110b,11110000b,11101110b,11100101b,11111
110b,11100100b ;kucuk s
          DFB
            11101110b,11100000b,11101111b,11110001b,11110001b,11101111b,11100
001b,11101110b ;kucuk yumusak g
          DFB
            11100100b,11100000b,11101110b,11100100b,11100100b,11100100b,11101
110b,11100000b ;buyuk i
          DFB
            11101111b,11110000b,11110000b,11101110b,11100001b,11100101b,11111
110b,11100100b ;buyuk S
          DFB
            11100000b,11100000b,11101110b,11110000b,11110000b,11110101b,11101
110b,11100100b ;kucuk c
          DFB
            11101110b,11110001b,11110000b,11110000b,11110000b,11110101b,11101
110b,11100100b ;buyuk C

```

```

.*****
,
*****

```

```

dista: EQU 020H ;Display durumu.
dista_0: EQU 00H ;1 ise timer set ekranini gosterir.
dista_5: EQU 05H ;1 ise + tusu.
dista_6: EQU 06H ;Ekran flash var mi? 1=evet.
dista_7: EQU 07H ;Ekran tazeleme biti.
flasta: EQU 021H ;Flash sayaci.
flasta_0: EQU 08H
flasta_1: EQU 09H
kesme: EQU 022H ;Timer kesme registeri.
kesme_2: EQU 012H ;1/10s. kesme biti.
kesme_3: EQU 013H ;1s. kesme biti.
genreg: EQU 023H ;Genel maksat registeri.
genreg_0: EQU 018H ;p3_1=7109 status ayaginin bir onceki durumu.
setno: EQU 07FH ;Ekran set edilen parametrenin numarasi.
flashx: EQU 07EH ;Flash yapilacak ilk karakterin ekran adresi.

```

flashuz: EQU 07DH ;Flash yapılacak karakter sayisi.
ista: EQU 07CH ;Isi baglantilarinin son durumu.
isilar: EQU 063H ;Su anki isilarin saklandigi ilk adres.
flashmem: EQU 055H ;Flash yapılacak karakterlerin saklandigi hafiza.

END



SONUÇLAR ve ÖNERİLER

Bu çalışmada, 87C51 mikrokontrolör temelli bir sistem dizayn edilmiştir. Teorik çalışma yapılarak proje ortaya çıktıktan sonra, sistemi oluşturan elektronik devre gerçekleştirilmiştir. Bu aşamada öncelikle sistem, kullanıcı arabirimi ve ana kart olmak üzere 2 temel parçaya ayrılarak baskılı devre tasarımı yapılmıştır. Çift taraflı ve elemanların takıldığı delik içleri iletken kaplı olan baskılı devreler piyasaya yaptırıldıktan sonra elemanlar monte edilerek elektronik devre fiziki olarak gerçekleştirilmiştir. Bu haliyle önce bağımsız daha sonra da film banyo makinasında denenilen devrenin kusursuz çalıştığı gözlemlenmiştir.

Dizaynın çalışır hale getirilmesi sonucu anlaşılmıştır ki 87C51, oldukça hızlı ve güvenli çalışan, parazit duyarlılığı son derece iyi olan, çok geniş komut dağarcığıyla esnek tasarım olanakları sağlayan ve birçok farklı mikroişlemciyle sorunsuz ve uyumlu çalışabilen güçlü bir mikrokontrolördür. Yeni tasarımlarda kullanılması tavsiye edilebilir. Ayrıca, ülkemizde pek kullanılmayan HCTL1100 motor kontrol entegresinin de çok önemli ve kullanışlı özelliklere sahip olduğu görülmüştür. HCTL1100, bir mikrokontrolör desteğinde her türlü motorun her türlü kontrolünü mükemmel şekilde yapabilir.

Tasarımın bitirilmesiyle, başlangıçta düşünülmeyen bazı özelliklerin eklenmesi gerekmiştir. Bu ilaveler için devre üzerinde hiçbir eleman değişmediği halde yalnızca program değiştirilerek bu gerekli ilaveler yapılmıştır. Bu, mikroişlemci destekli bir sistemin asıl büyük avantajıdır. Dizayn donanım olarak ortaya çıktıktan sonra, daha da geliştirmek mümkündür.

Devrenin donanım yapısında, özellikle mikrokontrolörün ve diğer çiplerin besleme ayaklarının hemen dibine 10 μ F'lık bypass kondansatörleri bağlanmalıdır. Bu, gürültülü çalışma ortamları için gereklidir. Ayrıca, kristallere beğli olarak çalışan entegre devrelerin kristal bağlantıları mümkün olduğunca kısa tutulmalıdır. Bunun yanı sıra 5V besleme gerilimi de çok iyi regüle edilmeli ve filtre kondansatörleri büyük seçilmelidir.

Teknik konuların dışında, üzücü olan ve belirtilmesi gereken bir konu, ülkemizde yüksek teknoloji ürünü ve yeni tasarlanmış elektronik devre elemanlarının temin edilmesinde karşılaşılan güçlüklerdir. Elektronik devre elemanı piyasamız, tasarımcıların bu konudaki ihtiyaçlarını karşılayamamaktadır. Bir diğer üzücü konu da, delik içi kaplama olarak tabir edilen baskılı devre teknolojisinin ülkemizdeki yetersizliğidir. Kompleks baskılı devre tasarlayan tasarımcılar, çok yüksek adetli siparişler hariç geri çevrilmektedirler.

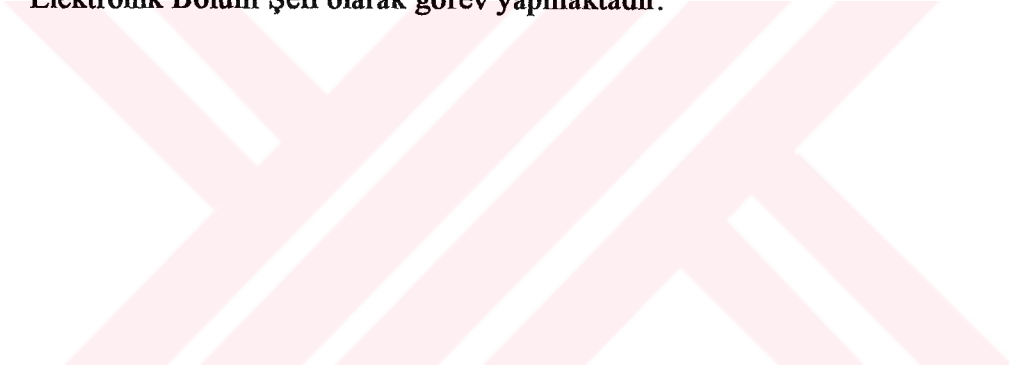
KAYNAKLAR

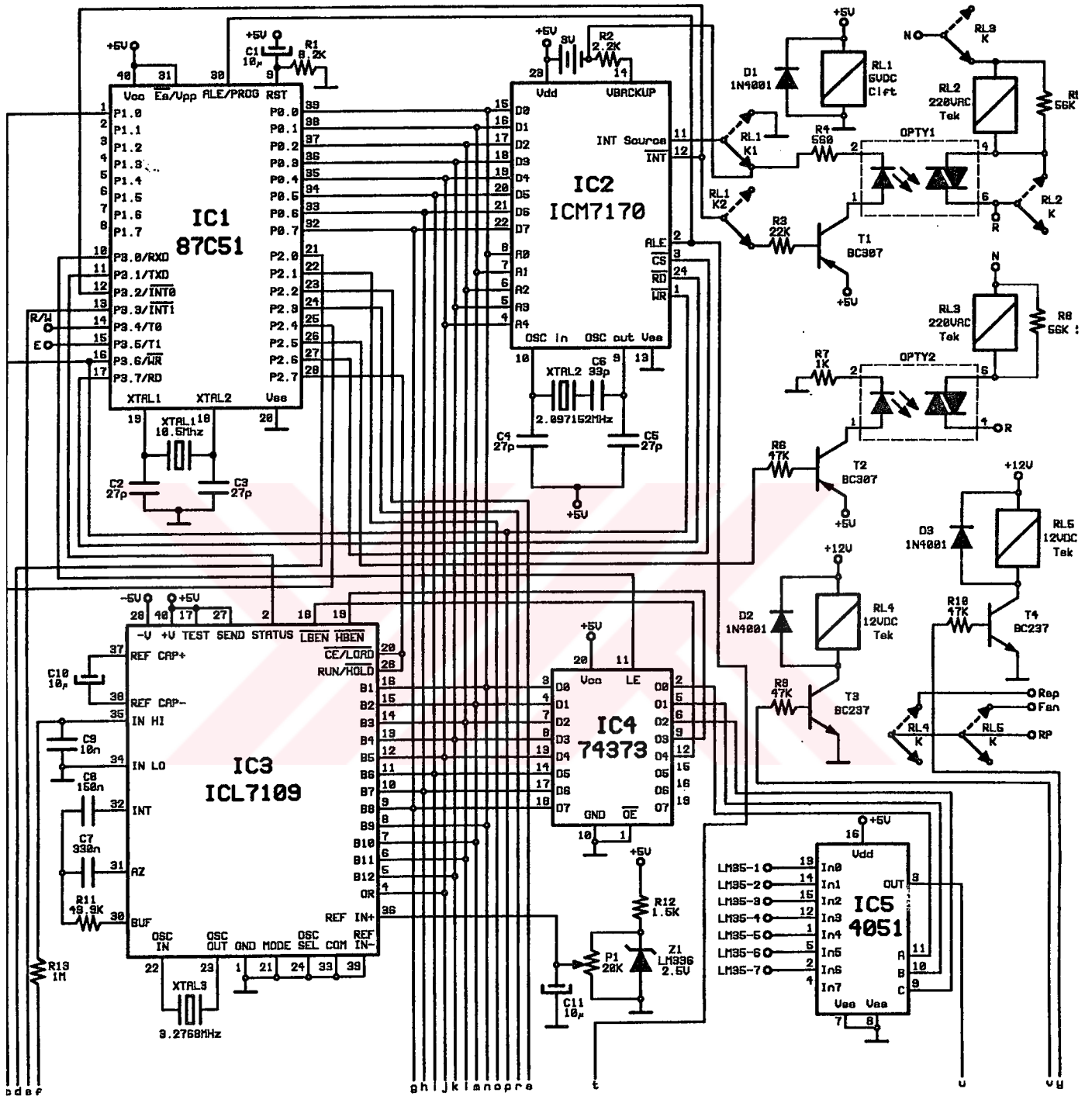
- 1- AMD, Advanced Micro Devices. 1989. 8051 Family. AMD, 1-1 - 7-47, USA.
- 2- AMD, Advanced Micro Devices. 1993. EPROM Products. AMD, 2-84 - 2-96, USA.
- 3- Harris Semiconductor. 1994. Data Acquisition. Harris S., 3-17 - 4-17, USA.
- 4- Hewlett-Packard Corporation. 1994. Optoelectronics Designer's Catalog. H.-P., 1-104 - 1-142, USA.
- 5- INTEL Corporation. 1994. Embedded Microcontrollers. Intel Semiconductor Product Groups, 2-1 - 2-146, USA.
- 6- National Semiconductor. 1988. CMOS Logic Databook. National S., 5-166, USA.
- 7- Optrex Corporation. 1995. Liquid Crystal Display. Optrex Corp., 96-109, Japan

ÖZGEÇMİŞ

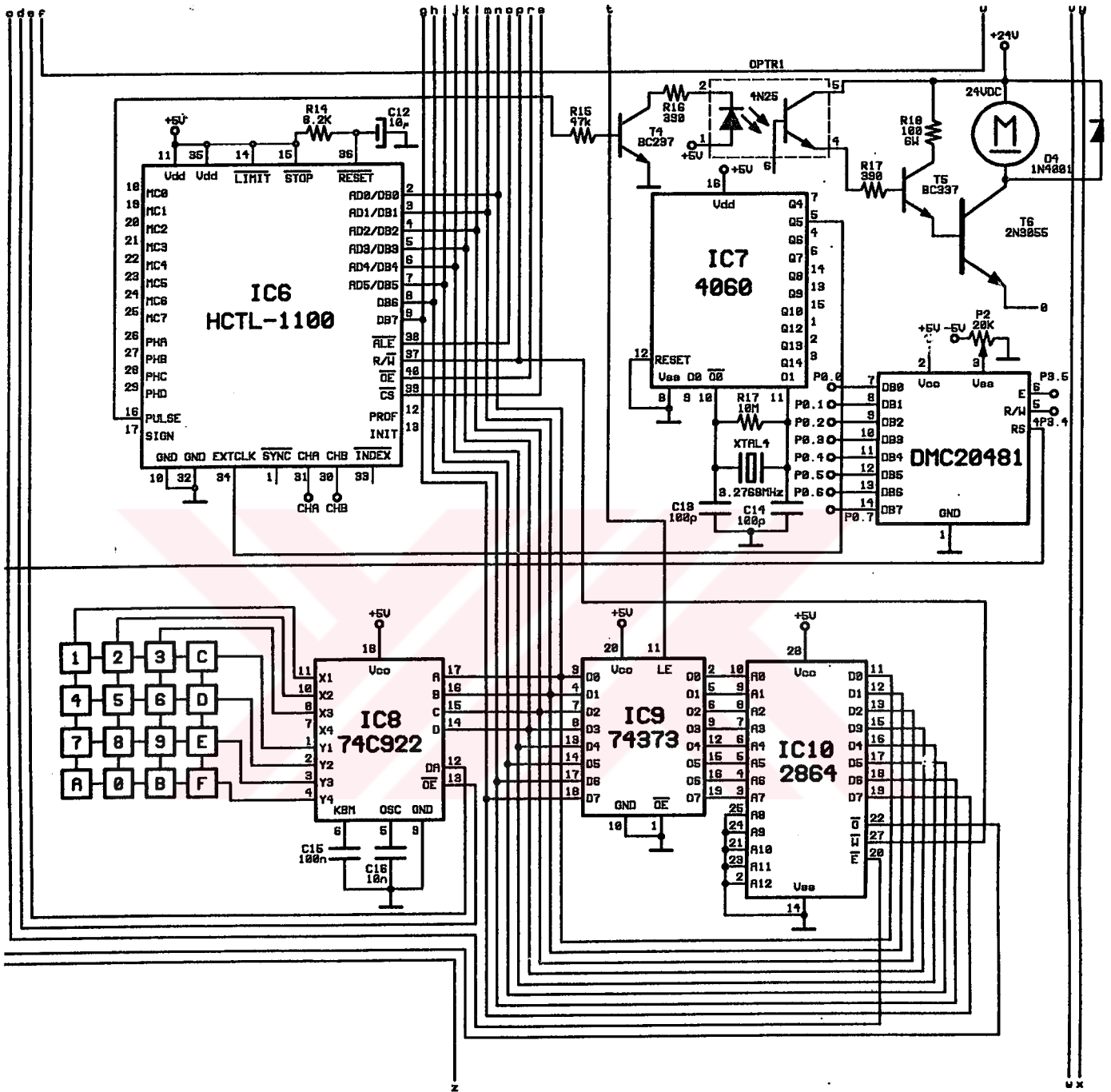
1971 yılında İzmit'te doğdu. İlk, orta, lise öğrenimini İzmit'te tamamladı. 1989 yılında girdiği Yıldız Üniversitesi Kocaeli Mühendislik Fakültesi Elektrik Mühendisliği Bölümü'nden 1993 yılında Elektrik Mühendisi olarak mezun oldu.

1991 yılından beri POLYFORM Limited Şirketi'nde Elektrik-Elektronik Bölüm Şefi olarak görev yapmaktadır.

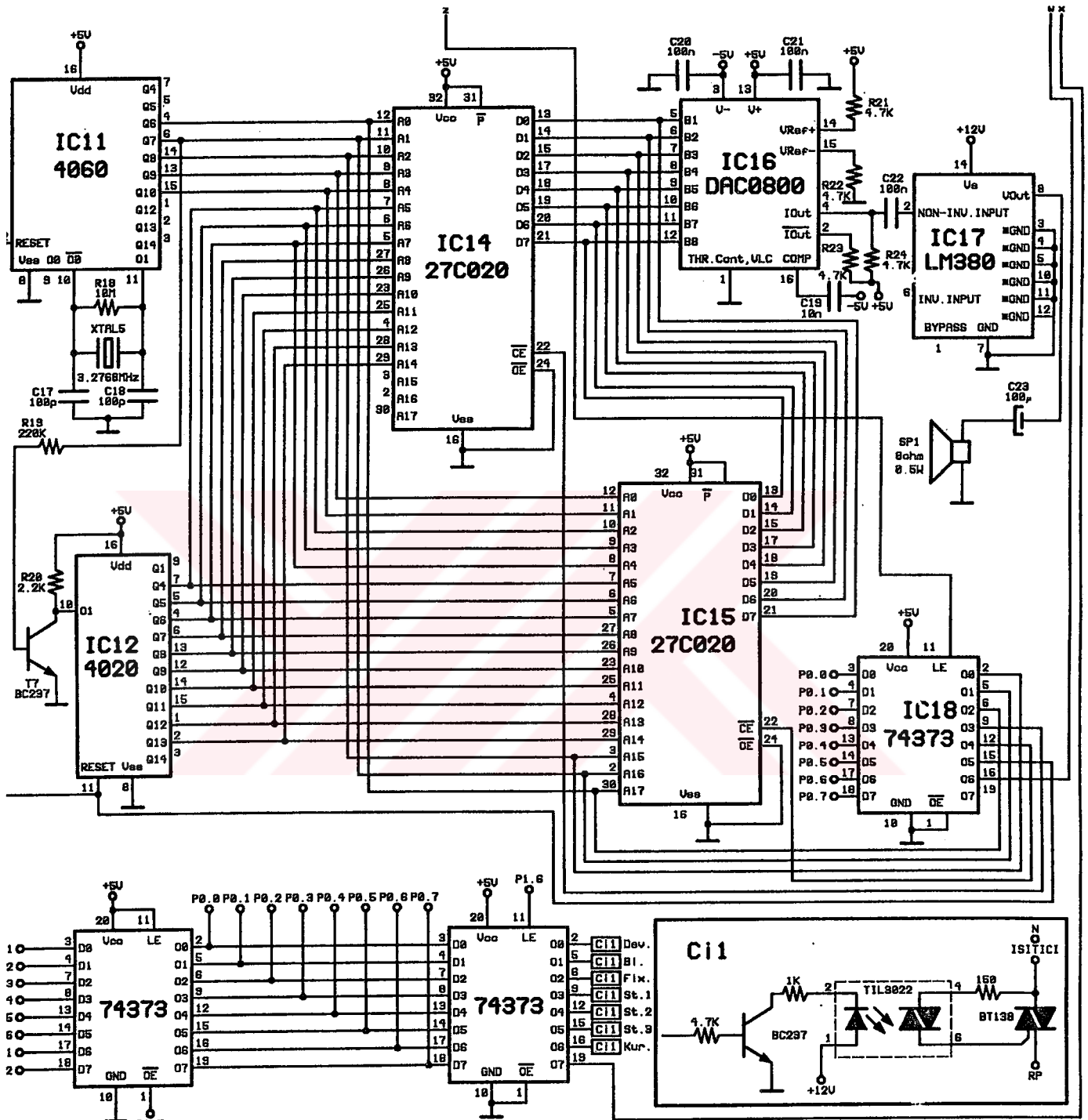




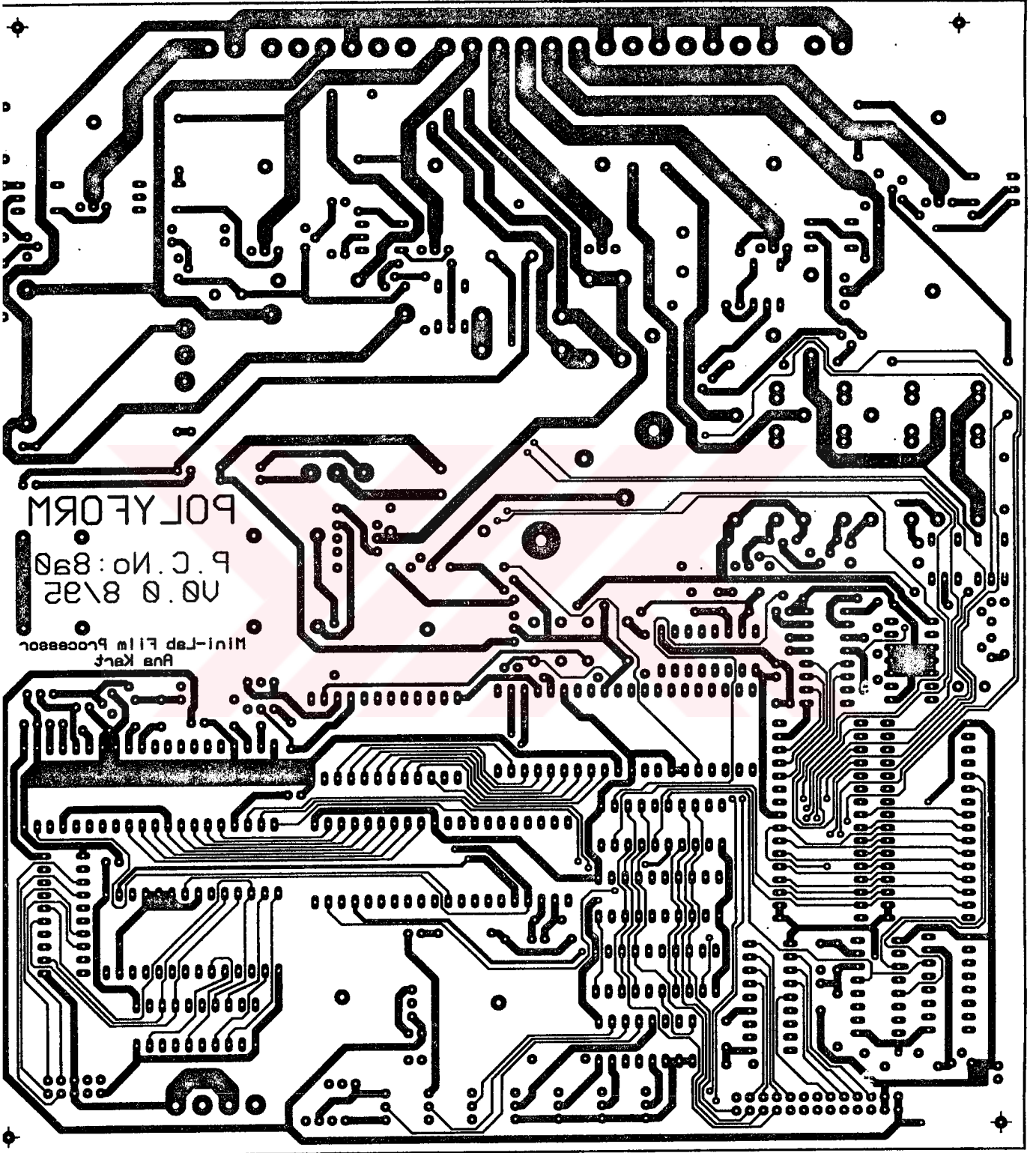
Şekil-3.a



Şekil-3.b



Şekil-3.c



Mini-Lab Film Processor
Ans. Karl
NO. 0 8732
P. C. No: 8280
POLYFORM

POLYFORM

P.C.No: 8a0
U0.0 8/95

Mini-Lab Film Processor
Ana Kart

