

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

128652

MİKRODENETLEYİCİLİ EV OTOMASYONU
ve GÜVENLİK SİSTEMİ

Elektronik ve Haberleşme Mühendisi Kamer KAYAER

FBE Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Elektronik Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

TE. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

Tez Danışmanı : Doç. Dr. Herman SEDEF (YTÜ)



Doç. Dr. Herman SEDEF


Prof. Şefik SARIKAYALAR

İSTANBUL, 2002


Y. Doç. Dr. Ferit ATTI

128652

İÇİNDEKİLER

Sayfa

KISALTMA LİSTESİ.....	iii
ŞEKİL LİSTESİ.....	iv
ÖNSÖZ.....	v
ÖZET.....	vi
ABSTRACT.....	vii
1. GİRİŞ.....	1
2. SİSTEMİN GENEL TANITIMI.....	2
2.1 Gerçekleştirilen Sistemin Genel Özellikleri.....	2
2.2. Cihazın Kullanımı.....	3
2.3 Sistemin Kullanım Alanları.....	5
3. GERÇEKLEŞTİRİLEN SİSTEMİN DONANIMI.....	6
3.1 Mikrodenetleyicinin Genel Özellikleri.....	6
3.2 Donanımın Blok Diyagramlar Halinde Açıklanması.....	7
3.3 Mikrodenetleyici Portlarının Üstlendiği Görevler.....	19
3.4 Devrenin Çalışması.....	20
3.5 Baskı Devre Yapısı.....	22
4. YAZILIMIN AÇIKLANMASI.....	24
5. SONUÇ.....	26
KAYNAKLAR.....	27
EKLER.....	28
EK 1 DEVRE ŞEMALARI.....	29
EK 2 BASKI DEVRE ŞEMALARI.....	33
EK 3 AKIŞ DİYAGRAMLARI.....	40
EK 4 YAZILIMIN KAYNAK KODU.....	56
ÖZGEÇMİŞ.....	110

KISALTMA LİSTESİ

Hz	Hertz
MHz	Mega Hertz
KHz	Kilo Hertz
kb	kilo byte
V	Volt
A	Ampere
Ω	Ohm
ms	mili saniye
AC	Alternative Current
DC	Direct Current
LCD	Liquid Crystal Display
CPU	Central Processor Unit
RAM	Random Access Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
MOS	Metal Oxide Semiconductor
SCI	Serial Communication Interface Module
SPI	Serial Peripheral Interface Module
COP	Computer Operating Module
IRQ	External Interrupt
PLL	Phase Lock Loop
PTA	Port A
PTB	Port B
PTC	Port C
PTD	Port D
PTE	Port E
TXD	Transmit X Data
RXD	Receive X Data
ADC	Analog Digital Converter
PPM	Pulse Position Modulation
PWM	Pulse Width Modulation
OSC	Oscillator

ŞEKİL LİSTESİ

	Sayfa
Şekil 2.1 LCD menüsü.....	3
Şekil 3.1 Donanımın blok diyagramı.....	7
Şekil 3.2 “Güç Kaynağı” bloğunun iç yapısı.....	8
Şekil 3.3 “Kontrol” bloğunun iç yapısı.....	10
Şekil 3.4 “RS232 ve Devre İçi Programlama” bloğunun iç yapısı.....	11
Şekil 3.5 “Tuş Takımı” bloğunun iç yapısı.....	13
Şekil 3.6 “LCD ve Buzzer” bloğunun iç yapısı.....	14
Şekil 3.7 “IRQ Devresi” bloğunun iç yapısı.....	15
Şekil 3.8 “Tutucular” bloğunun iç yapısı.....	16
Şekil 3.9 “Güç Çıkışları” bloğunun iç yapısı.....	17
Şekil 3.10 “Sensörler” bloğunun iç yapısı.....	18
Şekil 3.11 Güç baskı devresi.....	22
Şekil 3.12 Kontrol baskı devresi.....	23



ÖNSÖZ

Bu tezin hazırlanmasındaki katkılarından dolayı değerli hocam Doç. Dr. Herman SEDEF 'e ve tüm Devreler ve Sistemler Anabilim Dalı öğretim üyelerine teşekkür ederim. Ayrıca elektronik malzeme teminindeki yardımlardan dolayı, MOTOROLA firmasına da teşekkür ederim.



ÖZET

Endüstride, otomasyon sistemleri yüzyılı aşkın bir süredir kullanılmaktadır. Bu sistemlerin evlerde kullanımı ise, ancak üretim teknolojisindeki gelişmeler sayesinde gerçekleşebilmiştir. Yapılan pazar araştırmaları sonucunda, evlerin büyük bir çoğunluğunda, ev otomasyonu sistemlerine gereksinim duyulduğu ortaya çıkmıştır. Bu nedenle, sistem maliyetlerinin düşürülmesi, evlerde bu tür sistemlerin kullanılmasının yaygınlaşması açısından, büyük önem taşımaktadır.

Bu çalışmada; giriş verilerini (saat, sıcaklık, ışık şiddeti, vb. analog ve dijital veriler) değerlendiren ve bu değerlendirme sonucunda elektrikli ev aletlerini kısmi veya tam güçte çalıştıran veya durduran mikrodenetleyici temelli, düşük maliyetli bir sistem gerçekleştirilmiştir. Çalışmanın sunumunda, tasarlanan sistemin donanım ve yazılımının çalışma prensibi anlatılmış, sistemin kullanım alanları ve faydalarından söz edilmiştir. Sistemin devre şemaları, makine dili programının kaynak kodu ve akış diyagramları da verilmiştir.

Sistemde kullanılan tümleşik devreler, hem maliyet hem de güç tüketiminin az olması amacıyla, oldukça az sayıda tutulmuştur. Sistem, şu anda tek başına çalışmak için tasarlanmış olmasına rağmen, ileride cihazların kendi aralarında ve/veya bir kişisel bilgisayar ile haberleşebilecek şekilde geliştirilebilmesi için bazı sistem kaynakları saklanmıştır. Sistemin 16 girişi ve 16 çıkışı vardır. Girişlerin 8'i hem analog, hem de dijital olarak kullanılabilir. Diğer 8 giriş ise sadece dijitaldir.

Cihaz, girişlerine bağlanan algılayıcılardan (sensörlerden) bilgi alır ve kullanıcının belirlediği durumlara göre, girişler bağlı olarak, çıkışları kontrol eder. Örneğin; kullanıcı belli bir saatte evin dış aydınlatmasının açılmasını istemekte ise; cihazı, üzerindeki tuşları kullanarak bu durum için programlayabilir. Hatta istemesi durumunda, dış aydınlatmanın hem saat, hem de dışarıdaki aydınlığa bağlı olarak ve istediği güçte çalışmasını da kontrol edebilir.

Bu çalışma, BEKO Elektronik A.Ş.'nin 2001 yılında düzenlediği, Dijital Teknoloji Proje Yarışmasında üçüncülük ödülü almıştır. Bu yarışmada, birinciliğe layık bir proje bulunamamıştır.

Anahtar Kelimeler: Ev Otomasyonu, kontrol, 68HC908GP32

ABSTRACT

Industrial automation systems have been used for over a century. The usage of these systems at home has been realized by the improvement of the production technologies. According to the result of market researches; most of the houses need home automation. For this reason, to make this kind of systems common, among the houses, it is getting more important to lower the system cost.

This application describes a low cost home-automation system, which is based on a microcontroller. In this study, the working principles of the hardware and the software are explained and the possible usage of the system is presented. The schematic diagram, machine code source and flowchart of the system are also given.

Relatively, less electronic elements are used in the design to minimize the cost and the power consumption of the system. Although this system is designed as a stand-alone system, some system resources are reserved for further improvements like communication with other devices and/or PC. System has 16 inputs and 16 outputs. The 8 of the inputs can be used as analog or digital while the others can only be used as digital inputs.

Device controls the outputs according to the user settings by evaluating time and the input data (temperature, light, humidity, logic switch, door pass sensor, etc.), that is acquired from sensors and external devices. For example; user can set the device to turn the outdoor lights on at a certain time by using the keypad of the device. The device can also be set to control the outdoor lights to work at a desired power depending on time and outdoor illumination level.

This study has been awarded with the 3rd prize in the Digital Technology Project Contest, which is organized by BEKO Elektronik A.Ş in 2001. No project has been found to be eligible for the first prize, in this contest.

Keywords: Home automation, control, 68HC908GP32

1. GİRİŞ

Elektronik endüstrisi, geliştirdiği yeni teknolojileri hem ucuza üretme, hem de bu teknolojiler için yeni pazar yarata amacını güder. Yıllardan beri üretim teknolojileri alanında yapılan çalışmalar sonucunda; elektronik devre elemanlarının ucuza üretilmesi, maliyet engeline takılan birçok sistemin dünya pazarına girmesine olanak sağlamıştır. Son zamanlarda, bu gelişmelerden payını almış en önemli sistemlerden biri de ev otomasyonudur.

Otomasyon, bir sistemin belirli bir senaryoya göre, herhangi bir operatöre gerek duymaksızın yönetilmesidir. Senaryoların akışı, algılayıcılarla algılanan olaylara ve zamana göre belirlenir. Endüstride, otomasyon sistemleri yüzyılı aşkın bir süredir kullanılmaktadır. Bu sistemlerin evlerde kullanımı ise ancak, üretim teknolojisindeki gelişmeler sayesinde gerçekleşebilmiştir. Yapılan pazar araştırmaları sonucunda, evlerin büyük bir çoğunluğunda, ev otomasyonu sistemlerine gereksinim duyulduğu ortaya çıkmıştır. Ayrıca, ev otomasyon sistemlerinin enerji tasarrufunu artırması, her türlü soruna karşı güvenlik önlemi alması ve defalarca yapılan işleri otomatik hale getirmesi gerektiği anlaşılmıştır. Ev otomasyon sistemleri, bu ihtiyaçlara cevap vermek üzere tasarlanmaktadır.

2. SİSTEMİN GENEL TANITIMI

2.1 Gerçekleştirilen Sistemin Genel Özellikleri

Tezin amacı, ev otomasyonu ve güvenlik sistemleri için, düşük maliyetli, programlanabilen, mikrodenetleyicili bir kontrol sistemi tasarlamaktır.

Tasarımın genel özellikleri aşağıda verilmiştir.

- 8 analog, 8 dijital giriş
- 16 güç çıkışı
- Analog girişlerin dijital giriş olarak da kullanılabilme imkanı
- Çıkış gücünün ayarlanabilmesi
- Girişler ve programlar arasında lojik "ve" ve "veya" işlemlerini yapılabilmesi
- Elektrik kesildiğinde, sistemin çalışmaya devam etmesi
- İçindeki pilleri şarj etme yeteneği
- Pil olarak, 4 tane NI-CD pil, 6V veya 12V akü kullanma olanağı
- Çok amaçlı kullanım
- Düşük güç tüketimi
- Düşük maliyetli tasarım
- Az sayıda devre elemanı içermesi
- Seri haberleşme portunun bulunması
- Her bir çıkış için 8 ayrı program yapabilme imkanı
- Kullanıcı verilerinin mikrodenetleyicinin FLASH belleğinde saklanması
- Mikrodenetleyicinin devre içinde programlanabilme imkanı
- Devrenin programlanarak herhangi bir girişin herhangi bir sensör tipi için ayarlanabilmesi

2.2 Cihazın Kullanımı

Otomasyon devresinin giriş uçlarına, birçok çeşit algılayıcı (sensör) bağlanabilir (sıcaklık, ışık şiddeti, kapı eşiğinden geçişleri sayma vb.). Cihazın üzerindeki standart telefon tuş takımı kullanılarak, çıkışların hangi giriş değerleri (sensör girişlerine ek olarak saat ve haftanın günleri) için aktif olacağı ayarlanır. Bu ayarlamalar, her çıkış için 8 farklı biçimde belirlenebilir. Bu farklar; değişik günler veya aynı gün içindeki değişik saatler olabilir. Yani; her çıkış için girişlere bağlı 8 farklı program hazırlanabilir. Bir çıkış, en fazla 3 tane analog girişe bağlı olarak kontrol edilebilir. Dijital girişler için ise herhangi bir sınırlama yoktur.

Cihazın çıkışları programlanırken, kullanıcı dostu olarak tasarlanmış menüleri ile 2 satır 16 hanelik bir likit kristal gösterge (LCD) kullanılır. Örnek bir LCD menü şekli aşağıda görülmektedir.

ANA MENÜLER	ALT MENÜLER
1. ÇIKIŞ	ÇIKIŞ GÜCÜ
2. ÇIKIŞ	SAAT
3. ÇIKIŞ	SICAKLIK
4. ÇIKIŞ	IŞIK ŞİDDETİ
5. ÇIKIŞ	KAPI GİRİŞİ
:	LOJİK GİRİŞ
:	LOJİK İŞLEMLER
16. ÇIKIŞ	HANGİ GÜNLER

Şekil 2.1 LCD menüsü

Menünün Açıklaması:

Yukarıdaki şekilde görülen “1. ÇIKIŞ” menüsü, bir çıkış ana menüsüdür. Bu menülerin sayısı çıkış sayısı kadar, yani 16 tanedir ve alt menüleri birbirinin aynıdır. Bu menüye “Enter” tuşu ile girildiğinde, sağ tarafta görülen alt menülere erişilebilir. Alt menülere de “Enter” tuşu ile girilir ve çıkış için istenen çalışma şekli ayarlanır. Alt menülere ilişkin açıklama aşağıda verilmiştir.

1. ÇIKIŞ GÜCÜ : Çıkış aktif duruma geldiğinde, hangi güçte çalışacağı belirlenir.
2. SAAT : Çıkışın hangi saatler arasında aktif olacağı belirlenir.
3. SICAKLIK : Çıkışın hangi sensöre bağlı olarak, hangi sıcaklığın altında aktif olacağı belirlenir.
4. IŞIK ŞİDDETİ : Çıkışın hangi sensöre bağlı olarak, hangi ışık şiddetinde aktif olacağı belirlenir.
5. KAPI GİRİŞİ : Çıkışın hangi kapı girişi sensörüne bağlı olarak aktif olacağı belirlenir.
6. LOJİK GİRİŞ : Çıkışın hangi lojik girişe bağlı olarak aktif olacağı belirlenir.
7. LOJİK İŞLEMLER : Çıkış kontrol eden girişler ve programlar arasında yapılacak lojik “ve” ve “veya” işlemleri belirlenir.
8. HANGİ GÜNLER : Çıkış için yapılan programın, hangi günler için geçerli olacağı belirlenir.

Her çıkış için yukarıda açıklanan alt menüler vardır ve hepsi aynı yapıdadır. Her bir çıkış için 8 ayrı program yapılabilir. Örneğin; hafta sonunda, dış mekan ışıklarının hafta içine göre daha erken yanması isteniyorsa, aynı çıkış için yeni bir program da istenen bu durum için yapılabilir. Aynı çıkış için değişik programlar, bir gün içindeki değişik saatler için de yapılabilir. Örneğin; dış mekan ışıklarının 17:00 – 18:00 saatleri arasında daha düşük güçte, 18:00 – 06:00 saatleri arasında tam güçte yanması isteniyorsa; aynı çıkışta, iki değişik saat aralığı için iki değişik program yapılabilir. Farklı günler “HANGİ GÜNLER”, farklı saatler ise “SAAT” alt menüsü kullanılarak belirlenir.

2.3 Sistemin Kullanım Alanları

Aşağıda, sistemin (cihazın) kullanım alanlarına ilişkin birkaç örnek verilmiştir.

- Çocukların, geç vakte kadar bilgisayar ile uğraşmalarının istenmediği durumlarda; cihaz, bilgisayarın bağlı olduğu fişi saat 20:00'den sonra kapatacak şekilde programlanabilir.
- Cihaz programlanarak, sabah otomatik olarak çalışması istenen, çay makinesi veya radyo gibi, elektrikli ev aletlerinin çalışması sağlanabilir Bu durum, zaman tasarrufu yapılmasını da sağlayacaktır.
- Elektrik ile çalışan bir ısıtma sistemi varsa; cihaz, eve geliş saatinden önce ısıtma sistemini çalıştıracak şekilde ayarlanabilir. Böylece eve gelindiğinde ev ısınmış olacaktır. Odaların ne kadar ısınacağı da cihazın kontrolündedir. Bir ısıtma sisteminin bütün kontrolü bu cihaza yüklenebilir.
- Bir binanın dış ışıklandırması ve bahçe sulaması, bu cihaz kullanılarak otomatik hale getirilebilir.
- Odaya girildiğinde ışıkları yakıp, odada kimse kalmadığında ışıkları kapatacak şekilde bir programlama yapılabilir. Bu durum, enerji tasarrufu yapılmasını sağlayarak, hem doğayı hem de aile bütçesini koruyacaktır.
- Bir akvaryumun bütün kontrolü bu cihaz ile gerçekleştirilebilir.
- Çamaşır ve/veya bulaşık makinelerinin evde kimse yokken çalışması sağlanabilir. Böylece hem zamandan tasarruf edilir hem de evin içinde daha sessiz ve huzurlu bir ortam sağlanmış olur.

Bu örnekleri çoğaltmak mümkündür.

3. GERÇEKLEŞTİRİLEN SİSTEMİN DONANIMI

3.1 Mikrodenetleyicinin Genel Özellikleri

Kullanılan mikrodenetleyici, Motorola firmasının düşük fiyatlı, yüksek performanslı, 8-bit M68HC08 mikrodenetleyici ailesinin bir üyesi olan MC68HC908GP32 tümleşik devresidir. Bu ailenin bütün üyeleri, geliştirilmiş M68HC08 merkezi işlem birimini (CPU08) kullanır ve çeşitli modüllere ve bellek boyutlarına sahip bir şekilde, değişik paket tipinde üretilmektedirler.

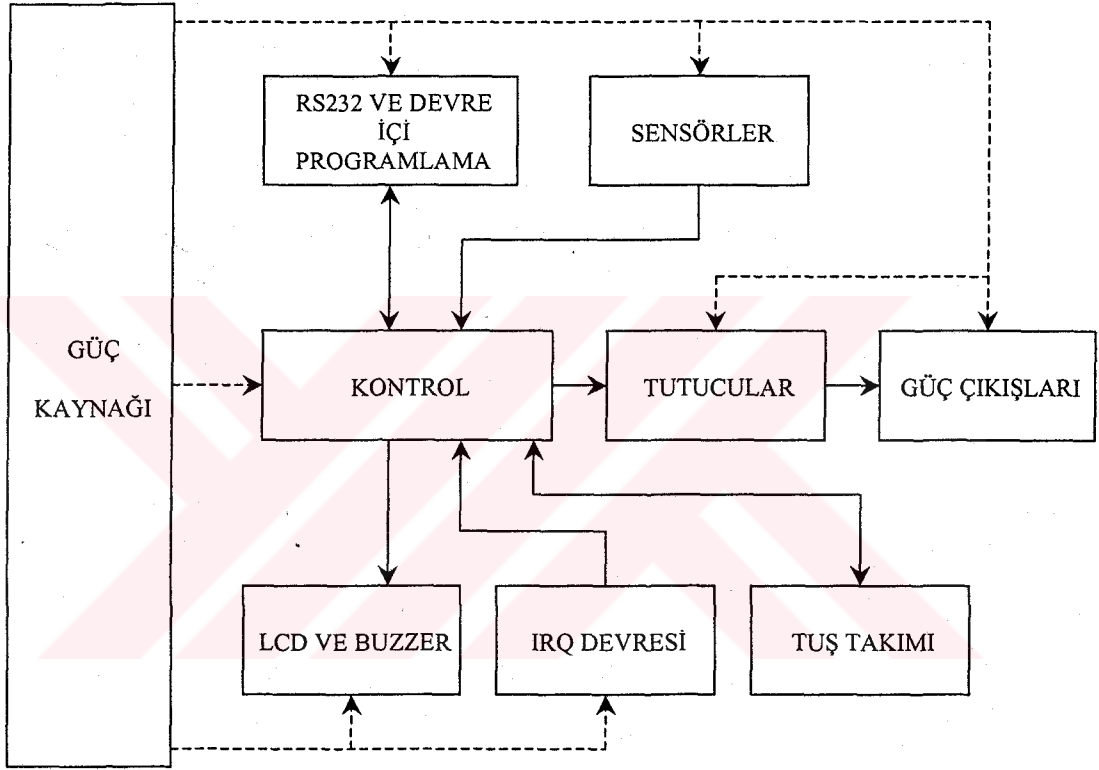
Özellikleri:

- C derleyicileri için uygun hale getirilen yüksek performanslı M68HC08 mimarisi
- M6805, M146805, ve M68HC05 aileleri ile yukarı doğru tam kod uyumluluğu
- 8 MHz iç veri yolu hızı
- Sistem içi programlamada yüksek gerilim gerektirmeyen, tek telli arayüz kullanımı (FLASH wire teknolojisi)
- İstendiğinde FLASH güvenliği kullanabilme özelliği
- Sistem koruma özelliği
 - Sistem düzgün çalışıyor (COP) reseti
 - 3V ve 5V olarak seçilebilen düşük gerilim algılama ve isteğe bağlı resetlenme özelliği
 - Hatalı kod algılama ve resetlenme
 - Hatalı adres algılama ve resetlenme
- Güç tüketimini azaltan DURMA (STOP) ve BEKLEME (WAIT) modları
- 32 Kb iç FLASH hafıza
- 512 byte kullanıcı RAM 'ı
- Seri çevre birimi arayüzü (SPI)
- Seri haberleşme arayüzü (SCI)
- İki tane, iki kanal, PWM özellikli 16-bit zamanlayıcı modülü
- 8-kanal 8-bit analog-dijital dönüştürücü
- RESET ve IRQ uçlarından V_{DD} 'ye içeriden pullup bağlantısı
- 32 kHz kristal uyumlu PLL ile saat sinyali üretme modülü
- 33 taneye kadar genel amaçlı giriş/çıkış ucu
- Her giriş/çıkış ucu için 10mA sürebilme yeteneği
- PTC0 ve PTC4 uçlarından 15mA sürebilme yeteneği
- 8-bit tuş takımı kesmesi ucu

3.2 Donanımın Blok Diyagramlar Halinde Açıklanması

Donanım, üstlendikleri göreve göre, 9 bloğa ayrılarak incelenebilir. Donanımın blok diyagramı aşağıda verilmiştir. Diyagramda, güç kaynağı bloğundan çıkan okların yönü güç akış yönünü, diğer bloklardan çıkan okların yönü ise veri akış yönünü göstermektedir.

Sistemin genel şeması EK 1’de verilmiştir.



Şekil 3.1 Donanımın blok diyagramı

Blokların iç yapısı (devre şeması) ve görevlerine ilişkin açıklamalar aşağıda verilmiştir.

Güç Kaynağı:

220V AC olan şebeke gerilimini sistemin ihtiyaç duyduğu 13.2, 7.2, 5.2V DC gerilimlerine dönüştürür.

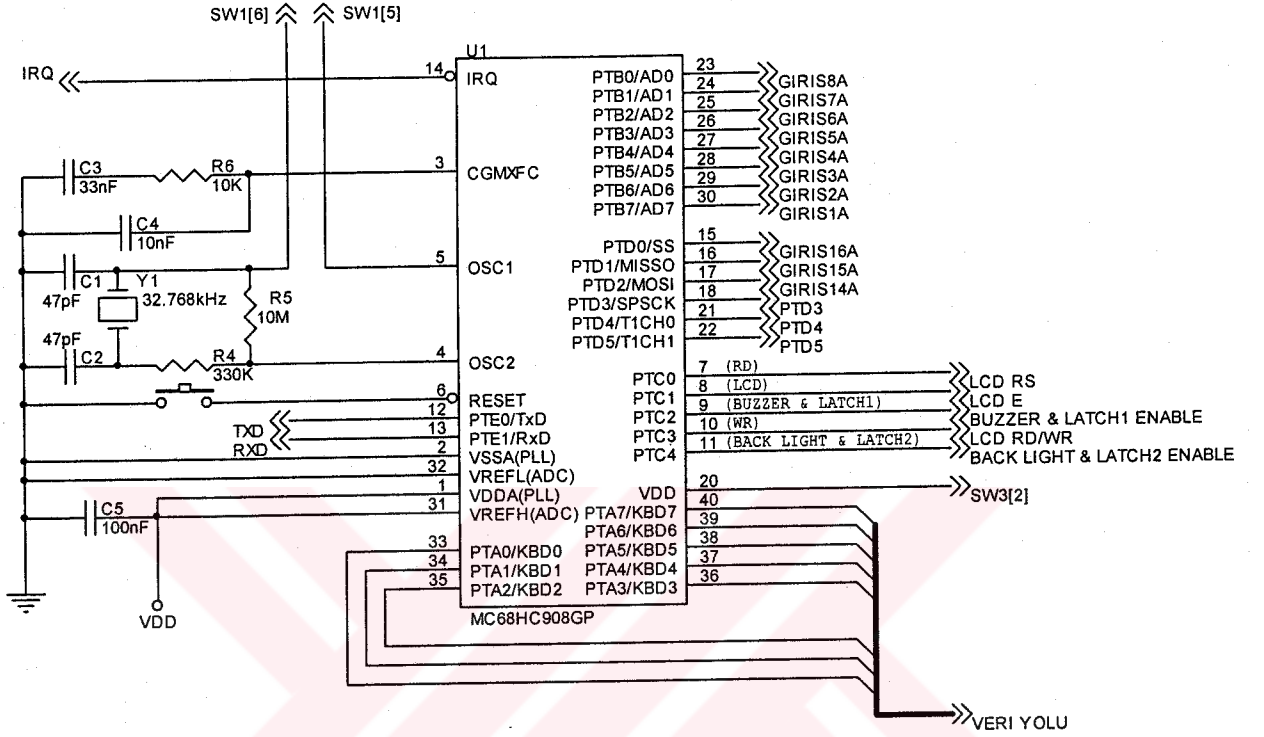
**T.C. YÜKSEKÖĞRETİM KURU
DOKÜMANTASYON MERKE**

Şebeke gerilimi, bir trafodan geçirilerek 12V AC gerilime dönüştürülür. Sonra bir köprü diyot ve kondansatörle, yaklaşık 17V'luk DC gerilime dönüştürülür. Bu gerilimden, 7812 tümleşik devresi, yeşil led ve bir diyot kullanılarak, yaklaşık 13.2V'luk bir gerilim elde edilir. Yeşil led 12V çıkış veren 7812 tümleşik devresini çıkış gerilimini yaklaşık 13.8V'a çıkarmak için kullanılır. Diyotun ise, iki görevi vardır. Bunlar; 13.8V olan gerilimi, 12V'luk bir aküyü şarj etmek için ideal bir gerilim olan, 13.2V a düşürmek ve eğer akü bağlıyken elektrik kesilecek olursa 7812'nin yanlış bir şekilde kutuplanarak güç harcamasını engellemektir. Elde edilen bu gerilim güç çıkışlarındaki röleleri tetiklemek için kullanılır. Bundan sonra gelen 7806 tümleşik devresi ve ona bağlı diyotlar, 12V'luk aküde olduğu gibi, 6V'luk aküyü şarj eden gerilimi oluşturmak için kullanılırlar. Aküden sonraki diyotlar ise akü gerilimini tümleşik devrelerin çalışabileceği yaklaşık 5.2V'a düşürmek için kullanılır.

12V'luk akü kullanılmıyorken ve jumperler takılmamışken, sistemin elektrik kesildiğinde de çalışmasını sağlayan gerilim, 6V'luk aküden elde edilir. Eğer J1, J3 ve J4 jumperleri takılırsa sistem 4 adet NI-CD pil (şarjlı pil) ile çalışabilecek duruma gelir. Sistemde aynı gerilim değerinde olan, VDD ve VDD2 adlı iki referans vardır. VDD referans gerilimi sadece mikrodenetleyiciyi ve LCD'yi beslerken, VDD2 diğer tümleşik devreleri ve sensörleri beslemek için kullanılır. Elektrik kesildiğinde, sistemde sadece yeni program yapabilme özelliği kullanılacaksa J2 açık tutularak 6V'luk akünün sadece mikrodenetleyici ve LCD'yi beslemesi sağlanarak enerji tasarrufu yapılabilir. Eğer sistemin sensörlerle beraber tamamı çalışsın isteniyorsa J2 kapatılarak (jumper takılarak) istenen bu durum sağlanabilir. Ancak bu durumda sistemde 12V olmadığından röleler çalışmayacaktır. Eğer sistemde 12V'luk akü varsa bütün sistem ve rölelerle çalışacaktır. 7.2V'luk gerilimin (VDD3), ileride sistemde yapılabilecek geliştirmeler göz önüne alınarak, sistem dışına çıkışı sağlanmıştır.

Kontrol:

Sistemin tüm kontrolünü bu blok sağlar.



Şekil 3.3 “Kontrol” bloğunun iç yapısı

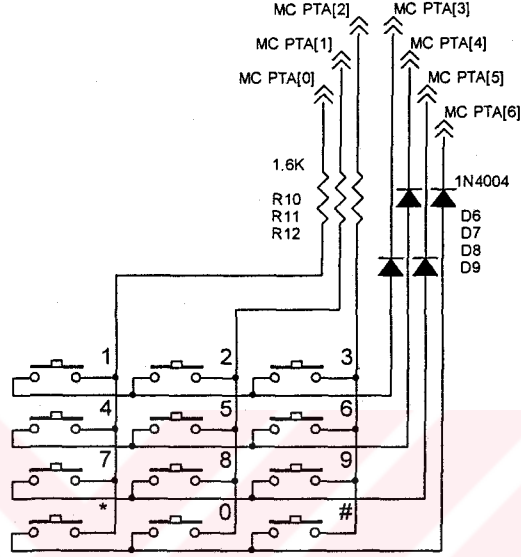
Mikrodenetleyici 32 MHz olan saat işaretini oluşturmak için, 32.768 kHz olan bir kristal ve iç yapısında bulunan bir faz kilitleme çevrimi (PLL) devresi kullanmaktadır. Daha genel bir yol olan dışarıdan 32 MHz kristal bağlamak yerine, bu yöntemin kullanılma nedeni; 32.768 kHz kristallerin yaygın kullanılmalarından dolayı kolay bulunmaları ve ucuz olmalarıdır. Ayrıca mikrodenetleyicinin dışında 32 MHz gibi yüksek bir frekansla osilasyon yapan bir osilatör kullanılmadığından, çevreye yayılan elektromagnetik gürül de azalacaktır. Bu yöntemle çalışma frekansı mikrodenetleyicinin programı tarafından ayarlandığından, aynı kristalle mikrodenetleyiciyi birçok farklı çalışma frekansında çalıştırmak da mümkündür. Bu durum sistemin esnekliğini arttırmaktadır. Mikrodenetleyicinin CGMXFC ucuna bağlı olan devre, mikrodenetleyicideki PLL'nin gereksinim duyduğu referans gerilimini oluşturabilmesi için kullanılmıştır. Mikrodenetleyicinin diğer uçlarının, ne amaçla kullanıldığı, daha sonraki bölümlerde anlatılacaktır.

Mikrodenetleyicinin FLASH belleğine makine kodlarının yüklenebilmesi için, “monitör modu” adındaki moda geçmesi gerekir. Bu moda geçebilmek için, mikrodenetleyicinin IRQ ucuna en az $VDD+2.5V$ olan bir gerilim, OSC1 ucuna 9.8304 MHz’lik saat işareti, PTC1 ve PTA7 uçlarına yaklaşık 0V, PTC0 ve PTC3 uçlarına ise yaklaşık 5V gerilim uygulamak gerekmektedir. PTC0 ve PTC3 uçlarına uygulanması gereken 5V gerilim, LCD’nin içindeki pull-up dirençleri sayesinde, ek bir donanıma gerek kalmaksızın, uygulanmaktadır. PTC1 ucu için gereken koşul, bu ucun $10k\Omega$ ’luk bir direnç üzerinden toprağa bağlanması ile sağlanır. PTA7 ucu giriş olarak da kullanılmaktadır, PTC1 için uygulanan yöntem giriş işaretini etkileyeceğinden, ancak IRQ ucu 9V olduğunda, PTA7 ucuna toprak gerilimi uygulayan MOS-transistörlü bir devre kullanılmıştır. IRQ ucundaki gerilim, çalışma gerilimi sınırları içindeyken (Yaklaşık 5.2V), 6.2V kırılma gerilimi bulunan zener diyotun kırılma gerilimini aşmadığından transistör iletimde değildir. IRQ ucundaki gerilim 9V olduğunda ise; bu gerilim 6.2V zener diyotun kırılma gerilimi ve MOS-transistörün eşik gerilimi toplamından yüksek olduğu için transistör iletime geçer ve PTA7 ucunu toprak gerilimine çeker. Mikrodenetleyici proramlanırken veri girişi PTA0 ucundan yarı-dubleks olarak yapılır. Yani, bilgisayar mikrodenetleyiciye bilgi gönderdikten sonra, mikrodenetleyiciden gelecek bilgiyi beklemeye başlar. Mikrodenetleyici gereken bilgileri bilgisayara gönderdikten sonra, bilgisayar tekrar mikrodenetleyiciye bilgi gönderir ve bu işlemler programlama tamamlanıncaya kadar sıra ile devam eder. Bu sıralı haberleşme programlama işleminin tek tel üzerinden gerçekleşebilmesini sağlar. ICL232 tümleşik devresi tam-dubleks haberleşme için tasarlanmıştır. Tam-dubleks haberleşmeyi, yarı-dubleks haberleşmeye uyarlamak için bir diyot kullanılmıştır. R1OUT ucu ICL232’nin çıkış ucudur. Bilgisayardan gelen elektriksel işaretler bu uçtan mikrodenetleyiciye ulaştırılır. T1IN ucu ise ICL232’nin giriş ucudur. Bilgisayara gönderilecek veriler, bu uca girilir. R1OUT ve T1IN ucu diyot ile birbirine bağlanmıştır. Bilgisayar mikrodenetleyiciden bilgi bekleme durumundayken, R1OUT ucundaki gerilim, RS232 standartlarına göre, 5V olacaktır. Bu gerilimin, T1IN ucuna giden veriyi etkilememesi için diyot, bu durumda tıkamada olacak şekilde konulmuştur. Bilgisayardan veri gelirken ise R1OUT 0V olduğunda diyot iletimdedir ve PTA0 ucu da lojik 0 gerilimindedir. R1OUT lojik 1 olduğunda ise diyot tıkama yönündedir ancak PTA0 ucunun bağlı olduğu LCD ucunun içindeki pull-up direnci PTA0’ı lojik 1 gerilimine çeker. Böylece R1OUT ucundaki lojik bilgi aynen PTA0 da görülür. Mikrodenetleyici bilgisayara bilgi gönderirken ise PTA0 ucu çıkıştır ve PTA0, T1IN ucuna bağlı olduğu için PTA0 ucundaki lojik bilgi aynen T1IN ucunda görülür ve bilgisayara iletilir. Böylece bilgisayar ile mikrodenetleyici arasında, yarı-dubleks bir haberleşme oluşturulmuş olur. Yeşil LED’in

yanması, bilgisayarın haberleşme portunu açtığını gösterir. Monitör moduna geçişle ilgili ayrıntılı açıklamalar, “Devrenin Çalışması” konu başlığı altında verilmiştir.

Tuş Takımı:

Kullanıcının menü içinde dolaşmak ve cihazı programlamak için kullandığı matris tuş takımı.

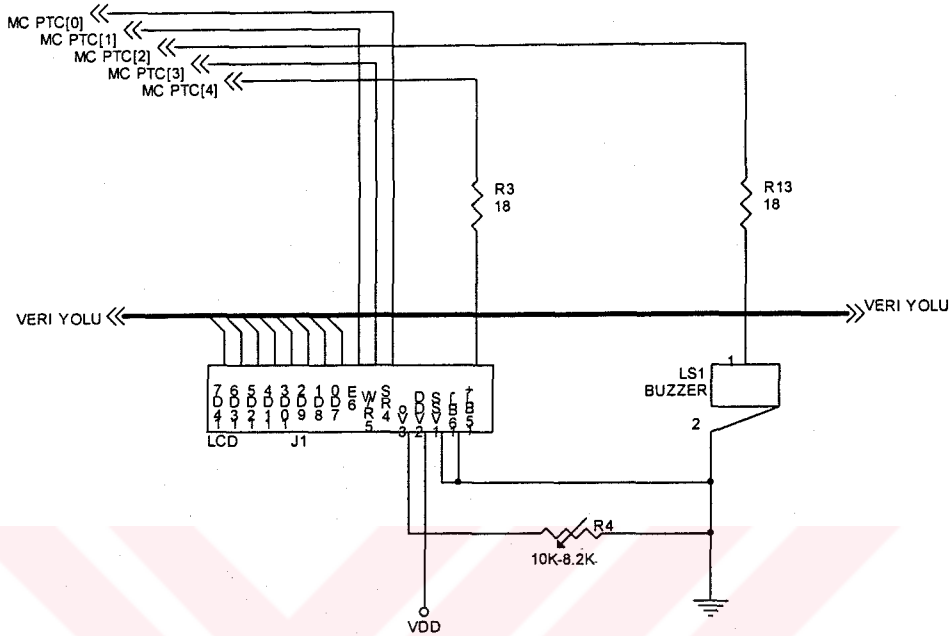


Şekil 3.5 “Tuş Takımı” bloğunun iç yapısı

Bir tuşa basıldığında mikrodenetleyiciye bir kesme üretilir ve bu kesme sonucunda mikrodenetleyici tuş takımını tarayarak hangi tuşa basıldığını bulur. Tarama işlemi şu şekilde yapılır: klavyenin diyot bağlı uçlarına, yani tuş matrisinin satırlarına, sadece biri lojik 0 diğerleri lojik 1 gerilim seviyelerinde olacak şekilde bir işaret uygulanır ve klavyenin direnç bağlı uçları, yani klavye matrisinin sütunları, okunur. Eğer 0 gerilim verilmiş satır üzerinde bir tuşa basılmamışsa, bilgi 111 olarak okunur. Eğer 0 olan satırda bir tuşa basılmışsa örneğin birinci satır aktifken 2 tuşuna basılmışsa sütun bilgisi 101 olarak olunur ve birinci satırda basılan tuş belirlenmiş olur. Eğer birinci satırda basılı bir tuş yoksa ikinci satıra bakılır. Bu işlem bütün satırlar bitinceye kadar devam eder ve basılan tuş bulunur. Tuşa basılmadığında sütunlardan okunan bilginin 1 olma nedeni, aynı uçlara bağlı olan LCD portlarındaki pull-up dirençleridir.

LCD ve Buzzer:

Menülerin görüntülenmesini ve tuşa basıldığında ses çıkmasını sağlar.



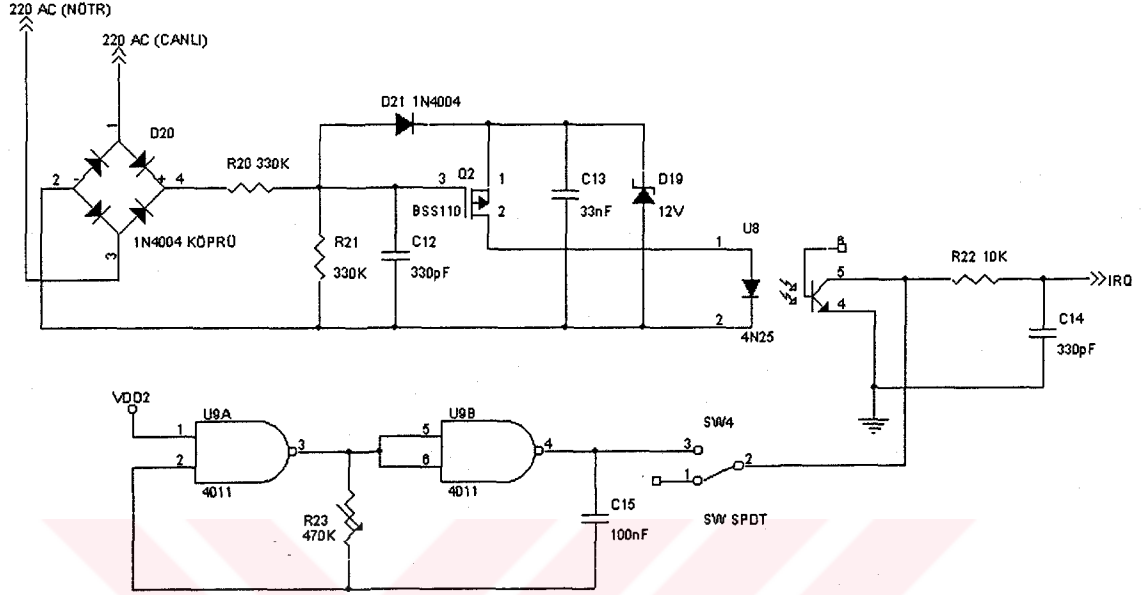
Şekil 3.6 “LCD ve Buzzer” bloğunun iç yapısı

Bir tuşa basıldığında mikrodenetleyici basılan tuşu algılar ve bu doğrultuda yapılması gereken işleri yaparak, bulunduğu konumu kullanıcıya bildirmek için LCD üzerinde bir görüntü oluşturur. Bu sayede, kullanıcı, menüler arasında dolaşarak istediği durum için, sistemi programlayabilir. Basılan tuşun algılandığını kullanıcıya bildirerek, kullanım kolaylığı sağlamak için, buzzer her tuşa basıldığında bir “biiip” sesi çıkartır. Bu sesi oluşturmak için buzzere kısa bir süre (100ms) elektrik vermek yeterlidir. LCD’nin arka-ışığı (back-light), bir tuşa basılınca yanar ve belli bir süre (8 saniye) tuşa basılmadığında, enerjiden tasarruf sağlamak için, söner.

LCD'ye veri girişi yapmadan önce, LCD'nin meşgul olmadığından emin olmak gerekir. Bunun için LCD'nin RS ucuna 0 ve R/W ucuna 1 girilerek LCD'nin DB7 ucu okunmalıdır. Eğer bu uç 1 ise LCD meşgul demektir ve veri girişi yapılamaz. Bu uç 0 olana kadar beklendikten sonra, önce R/W ve RS uçlarındaki veri girilir. Sonra, veri yoluna, oluşturulmak istenen görüntüye göre, gereken bilgi girilir ve LCD önce aktif, sonra tekrar pasif hale getirilir.

IRQ Devresi:

Şebeke geriliminin 0V noktasından her geçişinde mikrodenerleyiciye bir dış kesme üretir.



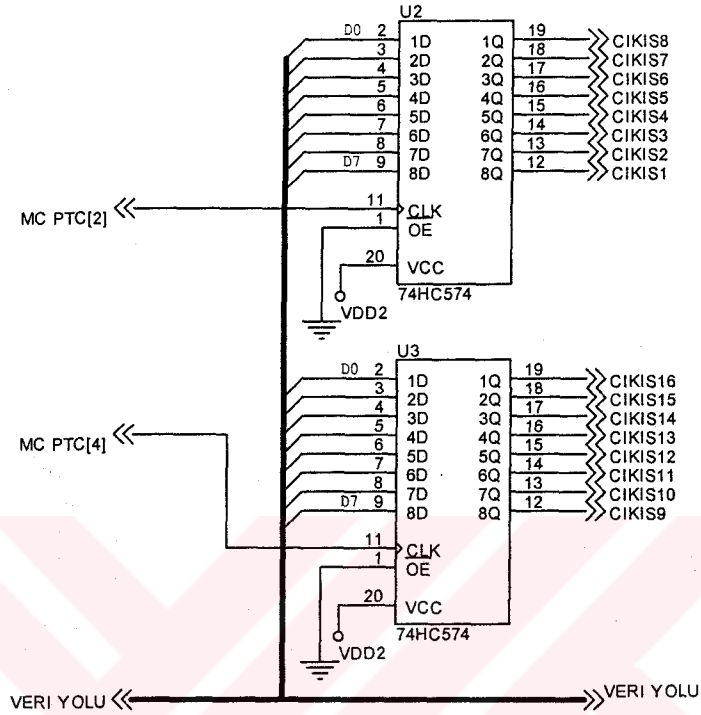
Şekil 3.7 “IRQ Devresi” bloğunun iç yapısı

IRQ devresi iki ayrı devreden oluşmaktadır. Yukarıdaki şeklin üst kısmındaki devre, şebeke gerilimi her 0V noktasından geçtiğinde, mikrodenerleyiciye bir kesme üretmek için kullanılır. Altındaki devre ise, eğer sistem şebeke gerilimine bağlanmadan, yani pil ile kullanılacak ise, sistemin ihtiyaç duyduğu yaklaşık 100 Hz frekanslı kesmeleri üretmek üzere kullanılan, bir osilatör devresidir. Bu devre Anahtar4’e (SW4) basıldığında aktif hale gelir.

Şeklin üst kısmındaki devrede şebeke geriliminin mutlak değeri 12V’un üzerindeyken MOS-transistör iletimde değildir. 33nF ve 330pF sığalı kondansatörler ise yaklaşık 12V’a kadar dolmuştur. Şebeke geriliminin mutlak değeri 12V’un altına düştükçe 330pF sığalı kondansatörün gerilimi giderek düşer ve 33nF sığalı kondansatör ile arasındaki gerilim farkı transistörün eşik gerilimini geçtiğinde transistör iletime geçer ve 4N25 tümleşik devresini tetikleyerek mikrodenerleyiciye bir kesme üretir. Kesmeler şebeke geriliminin hem düşen hem yükselen kenarında üretilmektedir. Şebeke gerilimi yönündeki 330pF sığalı kondansatörün kullanılma nedeni; şebeke gerilimini filtrelemektir. Çıkıştaki direnç ve kondansatörün kullanılma nedeni ise; oluşan kesme işaretini filtrelemektir. Anahtar4’e basıldığında, mikrodenerleyiciye, şebeke geriliminden bağımsız 100 Hz frekanslı kesmeler üretilir.

Tutucular:

Çıkış verisini saklar ve güç çıkışlarına iletir.

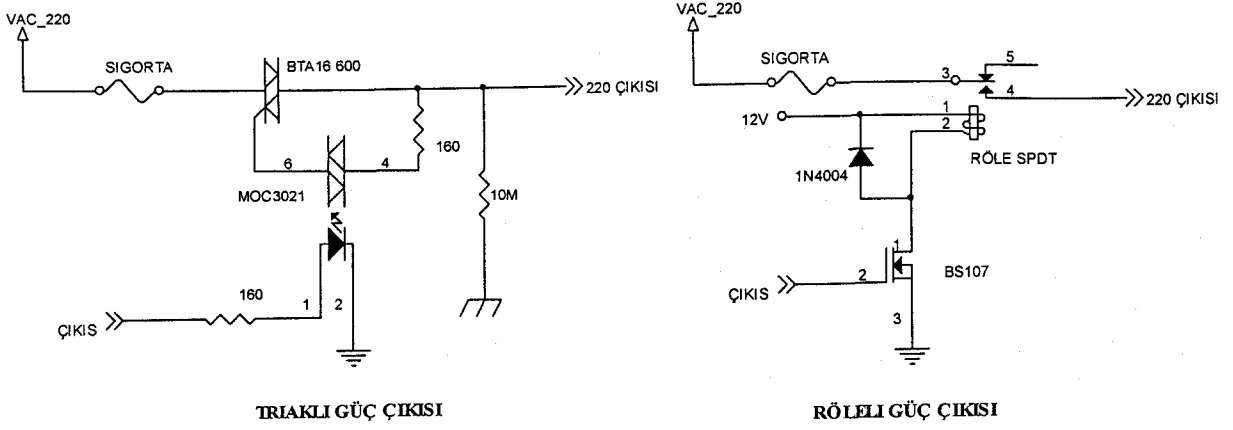


Şekil 3.8 “Tutucular” bloğunun iç yapısı

Mikrodenetleyici çıkış verilerini tutuculara yazar ve tutucular da bu verileri güç çıkışı devrelerine ileterek çıkışlara bağlı olan cihazları, açar veya kapatır. Tutuculara verileri yazılırken; önce veri, veri yoluna çıkartılır, sonra tutucunun CLK ucundaki lojik seviyesinin tersi bu uca uygulandıktan sonra tekrar eski lojik seviyesine getirilir. Tutucuların CLK uçları aynı zamanda buzzer veya LCD’nin arka-ışığına da bağlıdır. Tutuculara yazma işlemi çok kısa sürdüğünden, bu işlem sırasında buzzer ses çıkarmaz veya arka-ışığın yandığı fark edilmez. Eğer arka-ışık yanıyor ve/veya buzzer aktifse, bu durum bir problem oluşturmaz çünkü, tutucular yükselen kenar tetiklemelidir ve o anki durum iki kere ters çevrilerek aynı duruma gelindiğinden, tutucular mutlaka tetiklenecektir. Ancak buzzer veya arka-ışık aktif olurken, tutuculara yanlış bilgi yazılabilir. Bu durumu önlemek için, buzzer veya arka-ışık aktif edilirken, tutucularda bulunması gereken verinin, veri yoluna yazılmış olması, yazılım tarafından garanti edilir.

Güç Çıkışları:

220V AC prizine giden çıkışların kontrolü için kullanılır.

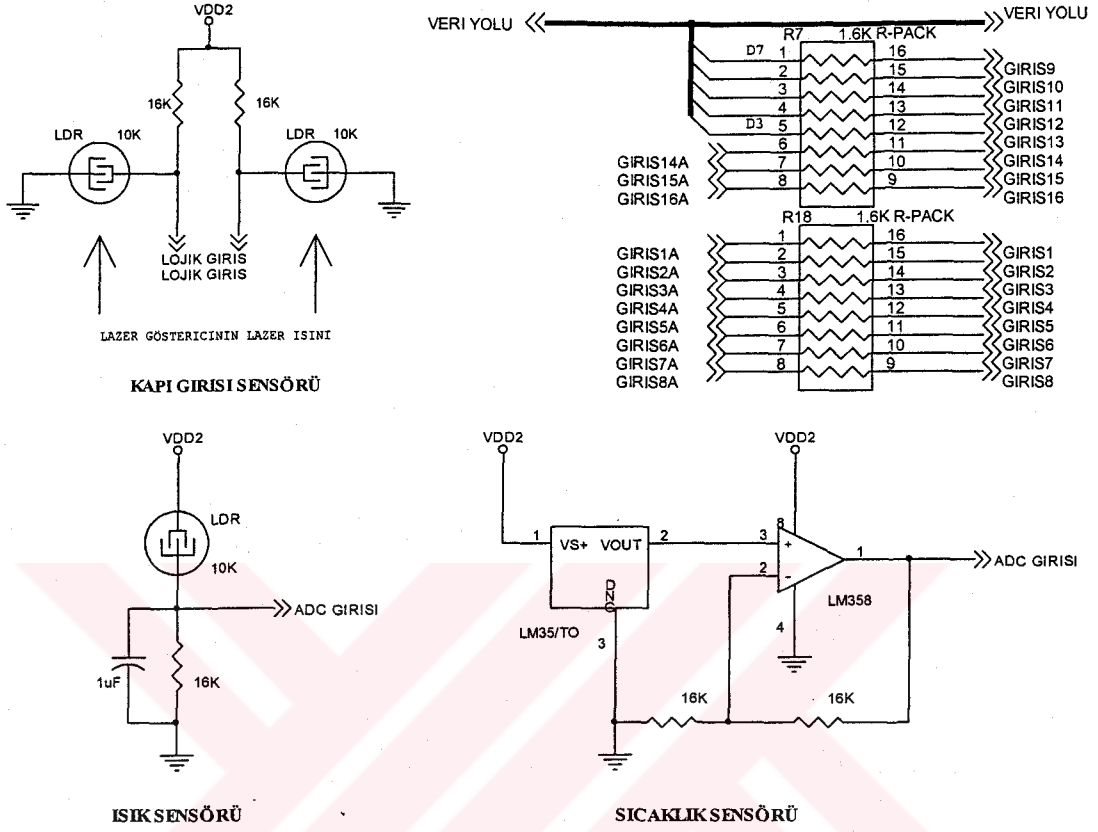


Şekil 3.9 “Güç Çıkışları” bloğunun iç yapısı

Triaklı güç çıkışındaki triak, bir opto-triak kullanılarak tetiklenir. Böylece şebeke gerilimi ile sistem birbirinden yalıtılmış olur. Röle devresindeki diyot, röle iletimden çıkarken, endüktansta oluşacak gerilimin, transistöre zarar vermesini önlemek için kullanılır.

Sensörler:

Dış ortamdan sıcaklık, ışık şiddeti, lojik veriler gibi bilgileri almak için kullanılır.



Şekil 3.10 “Sensörler” bloğunun iç yapısı

Işık sensörü olarak LDR’li bir gerilim bölücü devre kullanılmıştır. Ortamdaki ışık şiddeti arttıkça LDR’nin direnci düşer ve mikrodenetleyicinin ADC girişine uygulanan gerilim artar. Böylece ortamın ışığı mikrodenetleyici tarafından algılanmış olur. Sıcaklık sensörü olarak LM35 tümleşik devresi kullanılmıştır. Bu tümleşik devre, her 1°C’lik sıcaklık artışında çıkış ucundaki gerilimi 10mV arttırır. Eğer ortam sıcaklığı 20°C ise LM35’in çıkış ucundaki gerilim 200mV olarak okunur. Mikrodenetleyicideki ADC’nin hassasiyeti 20mV olduğu için, kazancı iki olan, op-ampli, evirmeyen bir kuvvetlendirici devresi kullanılmıştır. Kapı girişi sensörü, kapı eşliğinden geçen insanların, odaya girdiklerinin veya odadan çıktıklarının mikrodenetleyici tarafından algılanmasını sağlar. Mikrodenetleyici hangi ışığın önce kesildiğine bakarak, bu ayrımı yapar. Her giriş 1.6kΩ direnç üzerinden mikrodenetleyiciye bağlanır. Bu dirençlerin kullanım amacı sistem güvenliğini sağlamaktır. Ayrıca, veri yolunun giriş olarak da kullanılabilmesini sağlarlar.

3.3 Mikrodenetleyici Portlarının Üstlendiği Görevler

Sistemde kullanılan mikrodenetleyicinin A portu (PTA) üç tane görev üstlenmiştir.

Bu görevler:

- | | |
|------------------------------------|----------|
| 1. Sistemin veri yolunu oluşturmak | PTA[0-7] |
| 2. Tuş takımını taramak | PTA[0-6] |
| 3. Beş adet giriş oluşturmak | PTA[3-7] |

Mikrodenetleyicinin B portu (PTB) 8 tane analog ve/veya dijital giriş oluşturur.

Mikrodenetleyicinin C portu (PTC) beş tane kontrol ucu oluşturmak üzere kullanılmıştır.

1. PTC [0] =RD veya LCD'nin RS ucu
2. PTC [1] = LCD aktif ucu
3. PTC [2] = Buzzer ve LATCH1 aktif
4. PTC [3] = WR veya LCD'nin R/W ucu
5. PTC [4] = Backlight ve LATCH2 aktif

D portunun ilk üç ucu, 3 dijital giriş oluşturmaktadır. Bu portun diğer uçları ise, ileride yapılabilecek sistem geliştirmelerinde kullanılmak üzere saklanmıştır.

PTE uçları (TXD, RXD), sistemin RS232 portunu oluşturmak üzere kullanılmıştır.

3.4 Devrenin Çalışması

Devrenin çalışması, beş ana başlık altında anlatılabilir.

Dış Kesmeler:

Mikrodenetleyici iki tane iç, iki tane de dış kesme alır. Donanımla ilgili olduğu için, dış kesmeler bu bölümde, iç kesmeler ise yazılım bölümünde anlatılacaktır.

Mikrodenetleyicinin dış ortamdan aldığı kesmelerden biri, IRQ ucundan alınır. Bu uç, sistemde şebeke gerilimi 0V noktasından geçtiğinde, mikrodenetleyiciye bir kesme oluşturur. Bu kesmenin yardımı ile mikrodenetleyici şebeke geriliminin hangi noktada olduğunu anlar ve istenen güç çıkışına göre triakları tetikler. IRQ ucu şebeke geriliminden bir opto-kuplör ile izole edilmiştir.

Diğer bir kesme ise, mikrodenetleyicinin PTA[0-2] uçlarından alınan tuş takımı kesmesidir. Bu kesme oluştuğunda bir tuşa basılmış demektir ve mikrodenetleyici bu kesmeyi aldığı anda, tuş takımını tarayarak, hangi tuşa basıldığını anlar.

Girişler:

Mikrodenetleyicinin PTB uçları, sistemin analog girişleridir. Bu girişler mikrodenetleyicinin içinde bulunan ADC (analog digital converter) modülüne bağlıdır. Bu girişlere nem, ışık şiddeti vb. gibi analog sensörler bağlanabilir. İstenildiğinde dijital giriş olarak da kullanılabilirler.

PTD[0-2] ve PTA[3-7] uçları sistemin dijital girişleridir. PTA[3-7] uçları aynı zamanda veri yolunda da kullanılırlar.

İki çeşit analog giriş vardır. Bunlar sıcaklık ve ışık şiddeti girişleridir. Işık şiddeti girişi genel bir analog girişken, sıcaklık girişi sıcaklık ölçmek üzere özelleştirilmiştir.

Sistem programlanarak girişlerin tipleri kolayca değiştirilebilir. Bu sistemin esnekliğini artırarak kullanım alanını genişletir.

Çıkışlar:

Triak ve röle olmak üzere iki tane çıkış tipi vardır. Her bir çıkışın hangi tipte olacağı sisteme önceden programlanır. Triak çıkışlar, vuru pozisyonu modülasyonu (PPM) işaretler ile tetiklenir. Röle çıkışları ise puls şeklinde olmayıp sürekli (açık veya kapalı). Bütün çıkışların kontrolü iki tane tutucu (latch) ile (74HC574) sağlanır. Bu çıkışlar opto triak kullanarak triakları veya bir transistör üzerinden röleleri tetikler ve güç çıkışlarını oluşturur. Triakların tetiklenme zamanları istenen güce ve IRQ ucundan gelen kesmeye göre ayarlanır. Güç kullanıcı tarafından, 0-99 arasında seçilen bir değer ile belirlenir. Bu değere göre, triakların IRQ kesmesinden ne kadar zaman sonra tetiklenecekleri hesaplanır. Güç büyüdükçe kesmeden sonra geçen zaman küçülür. Örneğin güç için 99 yazılmışsa, tetikleme kesmeden hemen sonra yapılır. 0 değeri için ise hiçbir zaman tetikleme yapılmaz. Röle çıkışları genel amaçlı lojik çıkışlar olarak da kullanılabilir.

Devre İçi Programlama:

Devre içi programlama (In-circuit programming) devresi, üç anahtar, bir 9.8304 Mhz osilatör, bir 1N4004 diyot ve bir ICL232 entegre devresinden oluşmaktadır. ANAHTAR1 birinci konumdayken IRQ ve OSC1 uçları normal çalışma durumundadır. Monitör moduna girmek için, anahtar ikinci konuma getirildiğinde, IRQ ucu ICL232 entegresinin V+ ucuna (9V) ve OSC1 ucu 9.8304 MHz osilatöre bağlanır. ANAHTAR2 birinci konumdayken, mikrodnetleyicinin TXD ve RXD uçları ICL232'nin T1IN ve R1OUT uçlarına, RS232 portu oluşturmak üzere, bağlanır. Bu anahtar ikinci konuma alındığında, R1OUT ucu 1N4004'e ve T1IN ise PTA0 ucuna bağlanır. ANAHTAR3, mikrodnetleyiciye güç-açıldı (power-on) reseti oluşturmak üzere kullanılır. Monitör moda geçmek için, hem ANAHTAR1, hem de ANAHTAR2 ikinci konumdayken, ANAHTAR3 ile bir power-on reseti oluşturmak gerekir. Böylece mikrodnetleyici devre içinde programlanabilir. Sonra, ANAHTAR1 ve ANAHTAR2 tekrar birinci konuma alınarak bir power-on reseti oluşturulursa, sistem tekrar normal çalışma konumuna döner.

Kontrol Uçları:

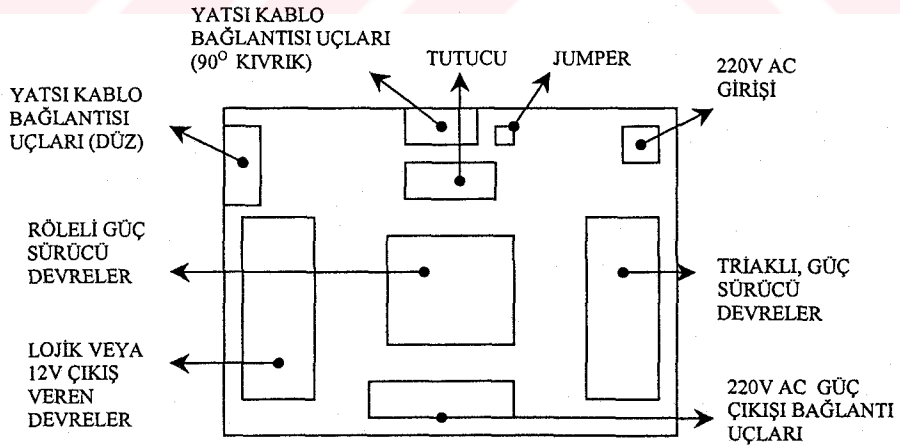
Çevre birimleri (mikrodenetleyici dışındaki tümleşik devreler) seçmek için bir adres çözümleme devresi kullanılmamıştır. Sistemde kullanılan entegre devre sayısının az olması için, bu kontrol mikrodenetleyicinin uçları kullanılarak sağlanmıştır.

PTC0 LCD'nin RS ucunu kontrol eder. Bu uç aynı zamanda ileride yapılacak sistem geliştirmelerinde RD (oku) ucu olarak da kullanılabilir. PTC1 LCD'nin enable ucunu kontrol eder. PTC2 BUZZER ve LATCH1'i kontrol etmek için kullanılır. Sistemin kullandığı uç sayısını azaltmak için, BUZZER ve LATCH'i kontrol etmek üzere iki ayrı uç kullanılmamıştır. PTC3 LCD'nin R/W ucunu kontrol eder. Bu uç aynı zamanda genel bir WR (yaz) ucu olarak da kullanılabilir. PTC[2] BACKLIGHT ve LATCH2'yi kontrol etmek için kullanılır.

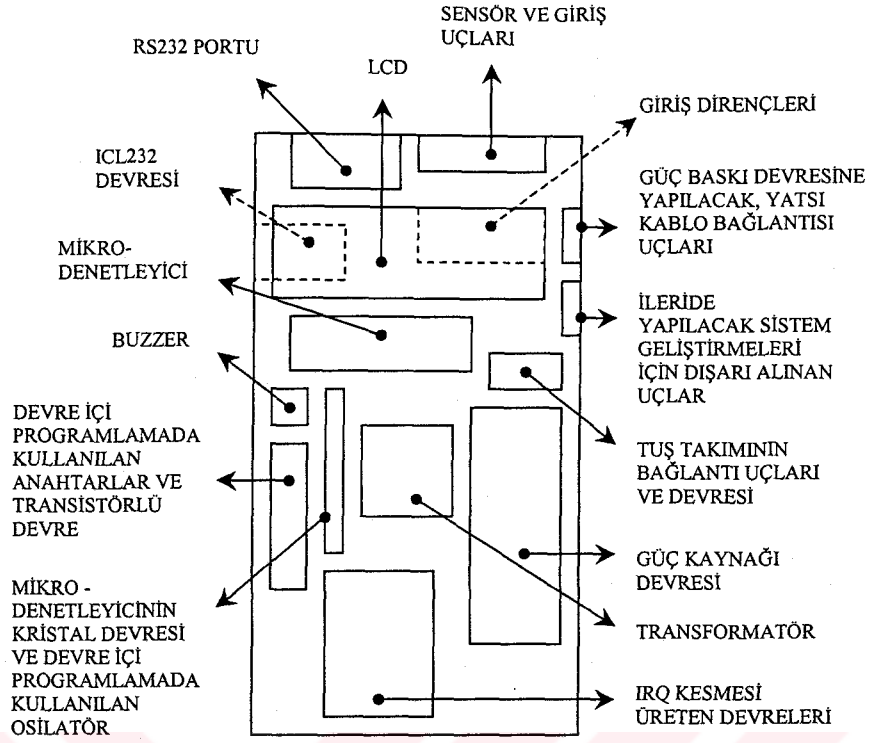
Devre şemaları EK 1'de verilmiştir.

3.5 Baskı Devre Yapısı

Sistem iki tane baskı devreden oluşmaktadır. Bunlardan biri güç baskı devresi, diğeri ise kontrol baskı devresidir. Elektronik eleman bloklarının baskı devreler üzerindeki yerleşimleri aşağıdaki şekillerde görülmektedir.



Şekil 3.11 Güç baskı devresi



Şekil 3.12 Kontrol baskı devresi

Kontrol ve güç baskı devreleri birbirine 16 uçlu bir yatsı kablo ile bağlanır. Güç baskı devresinde, iki tane yatsı kablo girişi vardır. Soldaki kablo girişi kablonun düz bir şekilde bağlanması için tasarlanmıştır. Üstteki kablo girişi ise, kablonun 90° kıvrılarak takılması için tasarlanmıştır. Baskı devrenin üzerindeki JUMPER'in iki konumu vardı. Birinci konumda güç kartı, 1-8 çıkışları iken, ikinci konuma getirildiğinde 9-16 çıkışları olur. JUMPER kullanılarak 16 çıkış için büyük bir kart yapma zorunluluğundan kurtulmuş olunur. Eğer 16 çıkış da kullanılacaksa, iki güç kartı kullanılır. Birinci kartın JUMPER'i birinci konumdayken kontrol kartına bağlanır ve ikinci kartın JUMPER'i de ikinci konumdayken yatsı kablo ile birinci güç kartına bağlanır. Böylece 16 çıkış da kullanılabilir hale gelir. Güç kartının tasarımı, her hangi bir çıkış için, isteğe göre, röleli, triaklı veya doğrudan lojik çıkış devresi kurulabilecek şekilde yapılmıştır.

Baskı devre şemaları EK 2'de verilmiştir.

4. YAZILIMIN AÇIKLANMASI

Sistem yazılımı altı bölümde incelenebilir.

1. İlk koşullar programı
2. Ana program
3. Zamanlayıcı1 (Timer1) kesmesi alt programı
4. Zamanlayıcı2 (Timer2) kesmesi alt programı
5. Dış kesme (IRQ) alt programı
6. Tuş takımı (keyboard) kesmesi alt programı

İlk Koşullar:

Yazılımın bu bölümünde, sistemin reset sonrası ilk çalışma koşulları belirlenir. Mikrodenetleyicinin çalışma özellikleri, portların ilk yönleri ve LCD 'nin çalışma modu yazılımın bu ilk bölümünde belirlenir. Cihazın çalışması sırasında, RESET kesmesi oluşmadığı sürece, bu bölüme bir daha geri dönüş yapılmaz.

Ana Program:

LCD menülerini oluşturma, basılan tuşun değerlendirme, otomatik tuşa basma, buzzer backlight ve menü geri dönüşü zamanlamaları, kullanıcı verilerinin kontrolü ve FLASH 'a yazılması, analog ve dijital girişlerin okunması, yazılımın bu bölümünde yapılır. Bu programın işi bittiğinde sistemin harcadığı gücün azalması için, yazılım bekleme (WAIT) moduna girer. Bu modda işlemci bir kesme bekler ve moddan sadece bu kesme ile çıkabilir.

Zamanlayıcı1 Kesmesi Alt Programı:

Zamanlayıcı1 (Timer1), saniyenin 5000 'de biri kadar küçük aralıklarla kesmeler üretir. Üretilen bu kesme sonucunda Zamanlayıcı1 kesmesi alt programına dallanılır. Bu programda, çıkış gücünün kontrolü (dimming) yapılır. Kesmelerin bu kadar sık olma nedeni, çıkış gücünün hassas bir şekilde ayarlanabilmesidir. Çok fazla hassasiyet istenmiyorsa bu kesmeler daha geniş aralıklar ile de yapılabilir. Bu aralık ile, 50 kademeli çıkış gücü ayarlaması yapılabilir. Kesme alt programı koşullurken, çıkış gücünün daha kararlı bir şekilde ayarlanabilmesi için, IRQ kesmesi maskelenmemiştir.

Zamanlayıcı2 Kesmesi Alt Programı:

Zamanlayıcı2 (Timer2), saniyede 5000 kesme üretir. Bu alt programda, kullanıcı verileri değerlendirilerek, çıkışların açık veya kapalı olacağı belirlenir. Bu kesme alt programı koşulurken, çıkış gücünün daha kararlı bir şekilde ayarlanabilmesi için, IRQ ve Zamanlayıcı1 kesmesi maskelenmemiştir. Gerçek zamanlı saat (Real-time clock) de bu kesme alt programında çalışır.

Dış Kesme Alt Programı:

Şebeke gerilimi her 0V noktasından geçtiğinde sistem bir dış kesme (IRQ) oluşturur ve yazılım dış kesme alt programına dallanır. Bu kesme programında, Zamanlayıcı1 kesmelerini sayan bir değişken sıfırlanır ve Zamanlayıcı1 resetlenir. Bu alt program koşulurken, maskelenebilen bütün kesmeler maskelenmiştir. Program bittiğinde, çıkış gücünün daha kararlı bir şekilde ayarlanabilmesi için, program hemen Zamanlayıcı1 kesmesi alt programına dallanır.

Tuş Takımı Kesmesi Alt Programı:

Bir tuşa basıldığında ana programdan tuş takımı kesmesi alt programına dallanılır. Bu kesme alt programında tuş takımı taranarak, hangi tuşa basıldığı anlaşılır ve bir RAM bölgesine yazılır. Tuşun basılı tutulması halinde bu kesme programına tekrar dönülmez. Kesme programının bir daha çalışabilmesi için önce bütün tuşların 25ms süreyle serbest bırakılması gerekir. Bundan sonra herhangi bir tuşa basıldığında, kesme programına tekrar dallanılır. Bu kesme alt programı çalışırken hiçbir kesme maskelenmemiştir.

Yazılımın akış diyagramları EK 3’de, kaynak kodu EK 4’de verilmiştir.

5. SONUÇ

Bu tezde mikrodnetleyici temelli, düşük maliyetli bir ev otomasyon sistemi gerekleřtirilmiřtir. Sistemin gerek yazılımı, gerekse donanımı tamamen orijinaldir. Sistemde kullanılan t mleřik devre sayısı olabildiğince az sayıda tutulmuř, b ylece hem maliyet hem de g c t ketimi azaltılmıřtır.

Ev otomasyonunda kullanılması hedeflenmiř olan bu sistem, sistem ıkıřlarının programlanma yeteneğine sahip olmasından dolayı, genel bir kontrol cihazı olarak da kullanılabilir. Sens r ve ıkıř tipleri sabit olmayıp istenen tipe g re ayarlanabilir. Cihaz, giriřler ve programlar arasında lojik “ve” ve “veya” iřlemleri yapmaktadır. Bu  zellik sayesinde karmařık kontrol iřlemleri de cihaz tarafından gerekleřtirilebilmektedir. Sistem her ıkıř iin 8, 16 ıkıř iin toplam 128 programı aynı anda y r tebilmektedir. řebeke gerilimi kesildiğinde, sistemde var olan 12V’luk ak , 6V’luk ak  veya řarj edilebilir d rt kalem pil seeneklerinden biri veya birkaçı ile alıřmaya devam etmektedir. řarj iřlemi cihaz tarafından otomatik olarak yapılır.

Sistem, kullanıcı verilerini mikrodnetleyicinin FLASH belleğinde saklar. Bu sayede, bir EEPROM t mleřik devresi kullanmaya gerek kalmamıřtır. Mikrodnetleyicinin kaynak kodu, mikrodnetleyici devredeyken y klenebilir veya değilřtirilebilir. Sistem ayrıca bir seri haberleřme portuna da sahiptir.

Bu alıřma, BEKO Elektronik A.ř.’nin 2001 yılında d zenlediğ, Dijital Teknoloji Proje Yarıřması’nda   nc l k  d l  almıřtır. Bu yarıřmada, birinciliğ layık bir proje bulunamamıřtır.

KAYNAKLAR

Motorola MC68HC908GP32 Technical Data REV5 – July, 2001

Motorola HC08 CPU08 Central Processor Unit Reference Manual REV2



EKLER

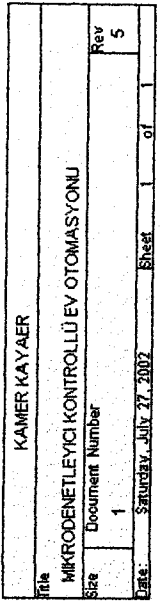
- EK 1 DEVRE ŞEMALARI
- EK 2 BASKI DEVRE ŞEMALARI
- EK 3 AKIŞ DİYAGRAMLARI
- EK 4 YAZILIMIN KAYNAK KODU



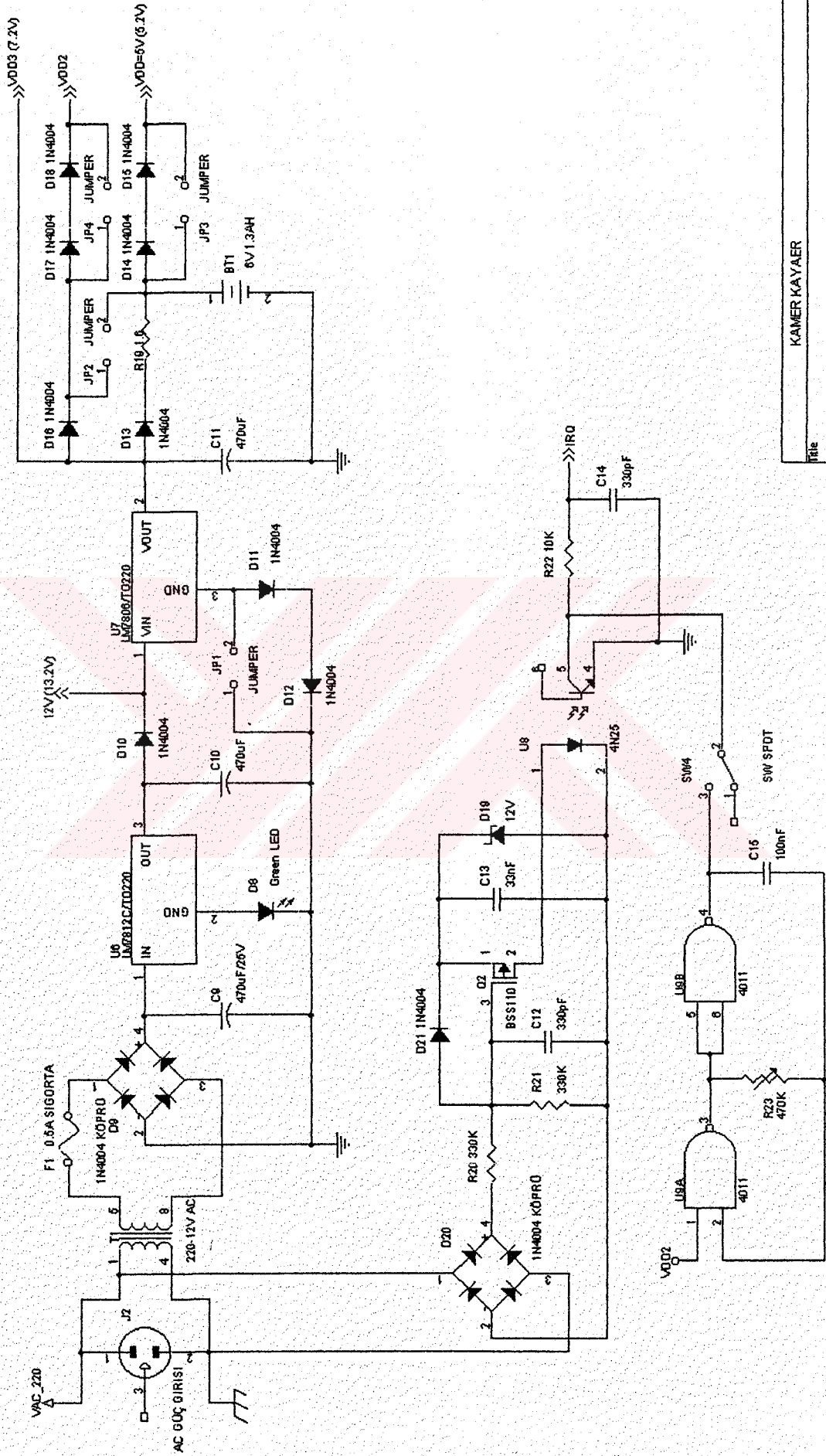
EK 1

DEVRE ŞEMALARI



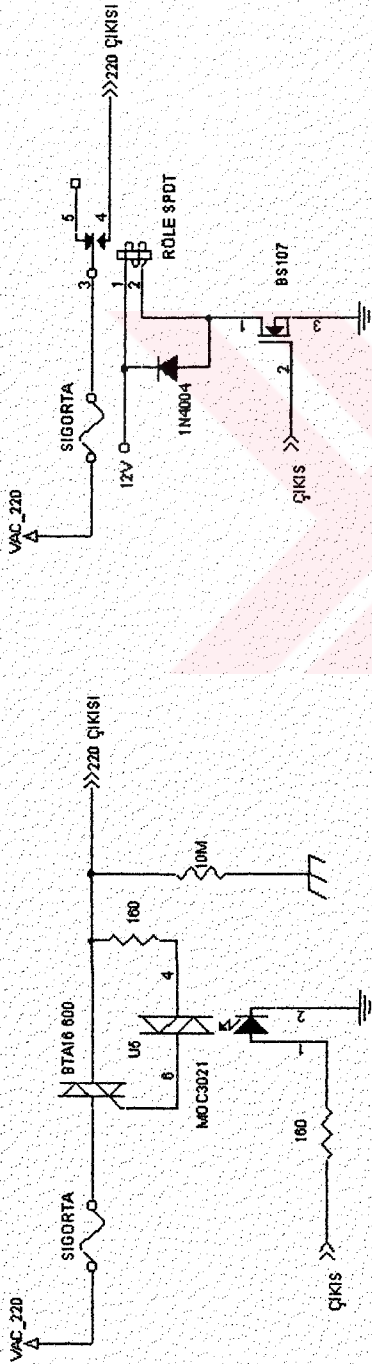


VDD=5V

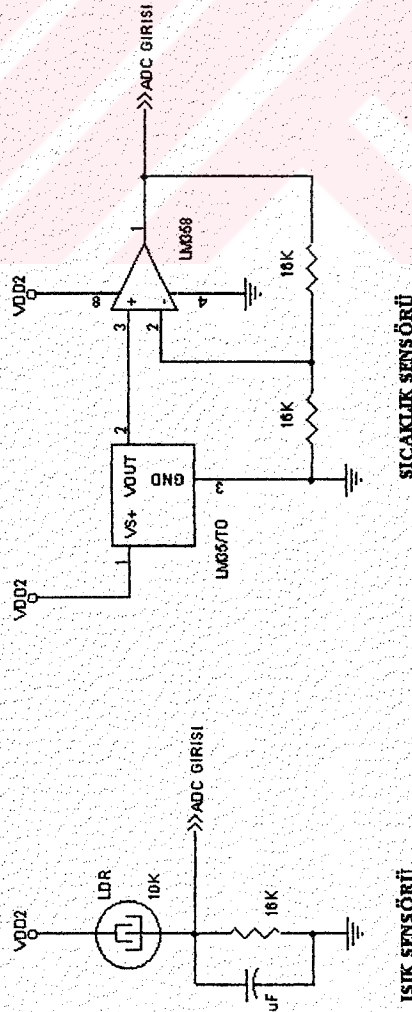
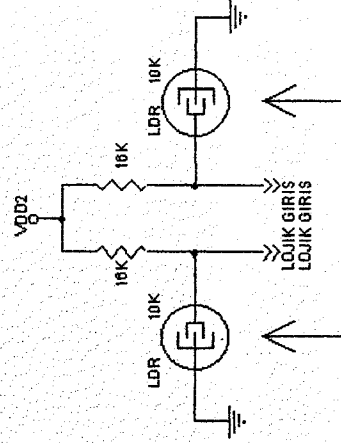


KAMER KAYAER			
MİKRODENETLEYİCİ KONTROLLÜ EV OTOMASYONU			
Size	Document Number	Rev	5
1	1	1	1
Date	Satirdev July 27, 2002	Sheet	of 1

VDD=5V



RÖLELİ GÜÇ ÇIKIŞI



SICAKLIK SENSÖRÜ

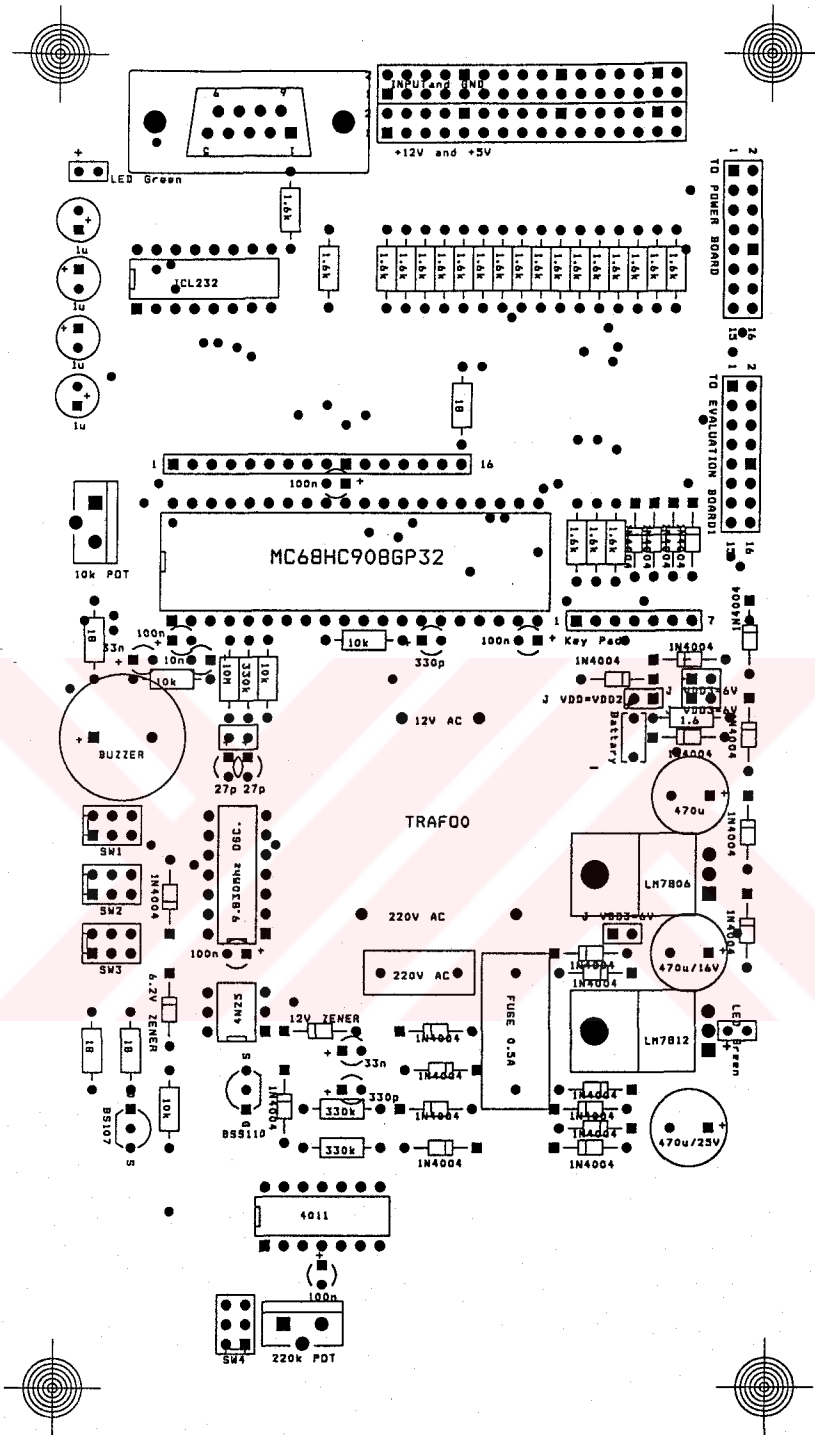
KAPI GİRİŞİ SENSÖRÜ

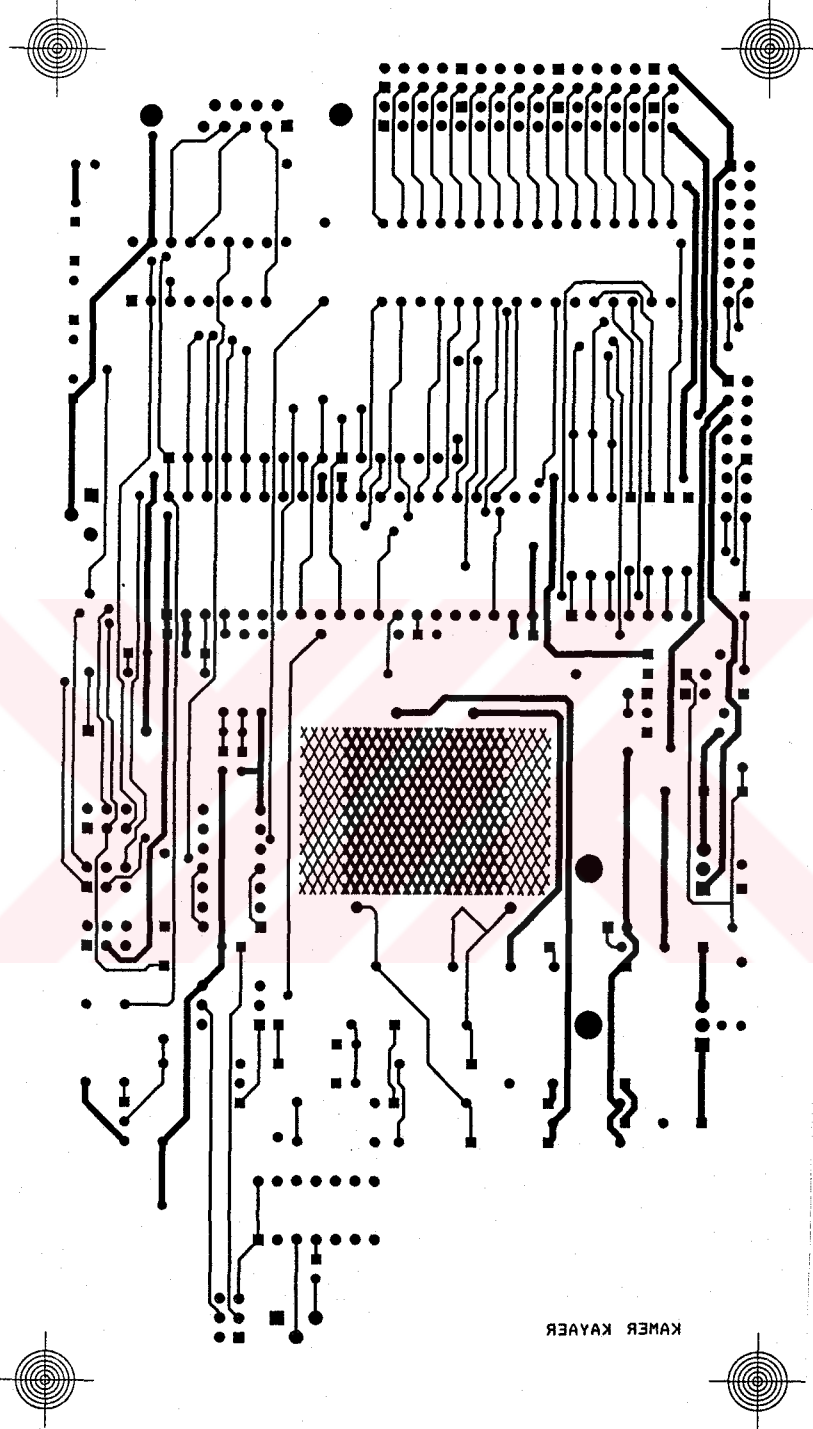
LAZER GÖSTERGİNİN LAZER İLİŞİ

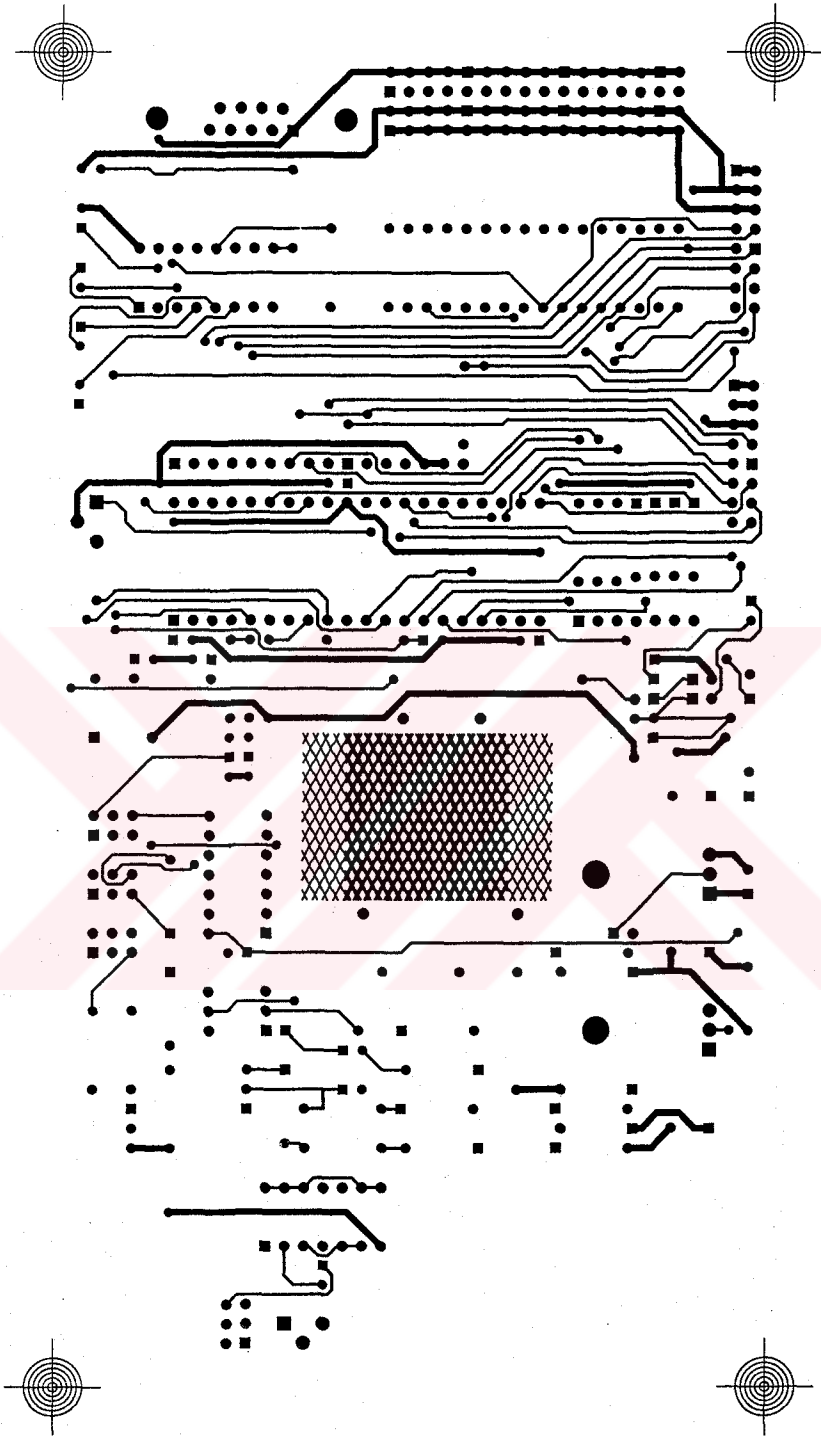
KAPI GİRİŞİ SENSÖRÜ

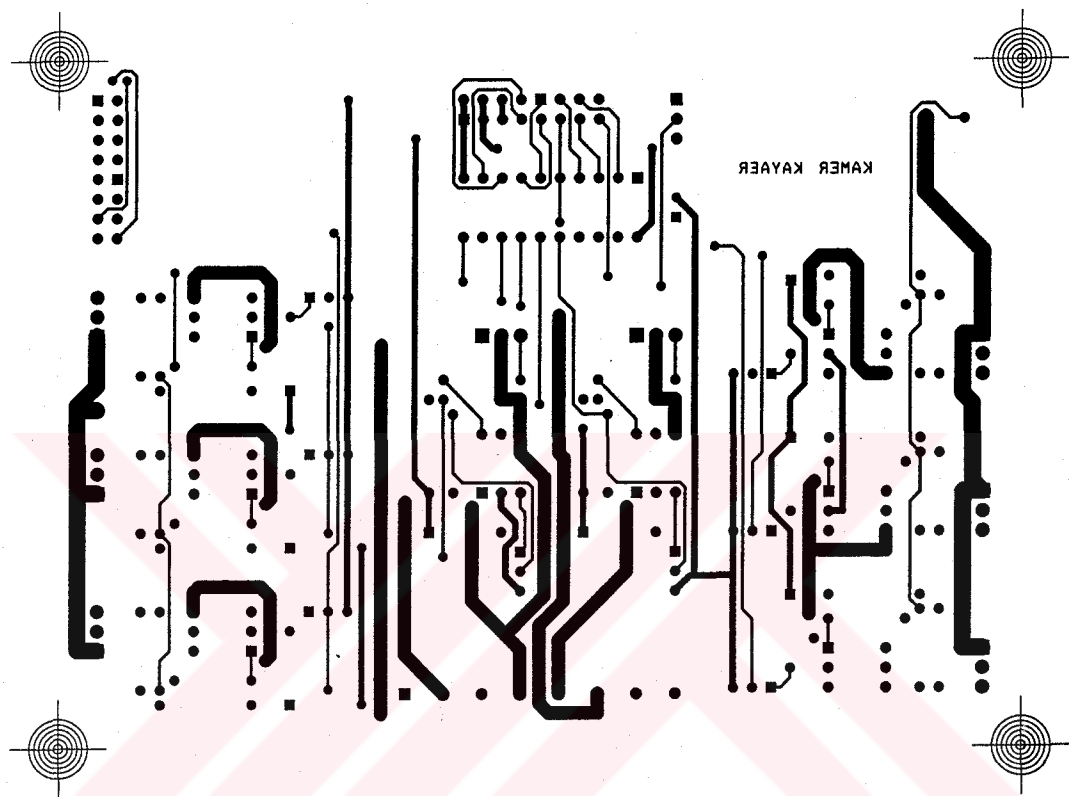
File	KAMER KAYAER
Size	1
Document Number	1
Rev	5
Date:	Wednesday, June 12, 2002
Sheet	1 of 1

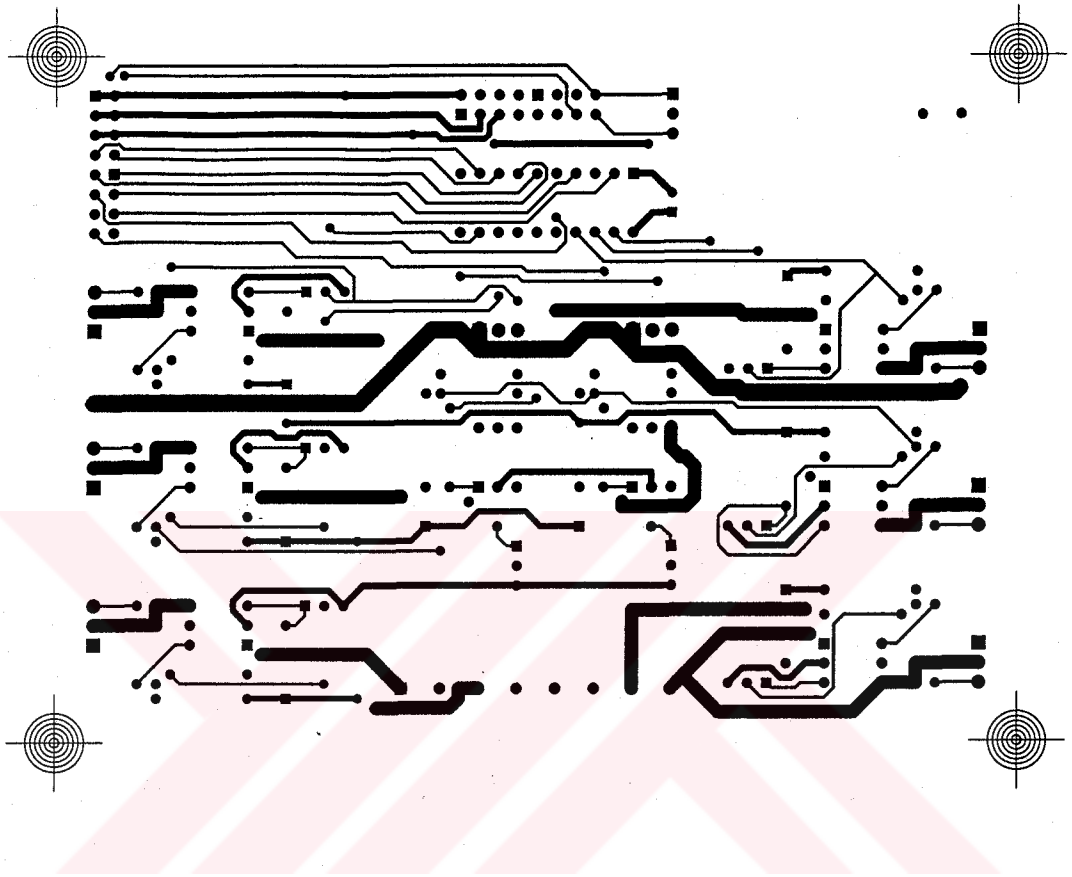
EK 2**BASKI DEVRE ŞEMALARI**









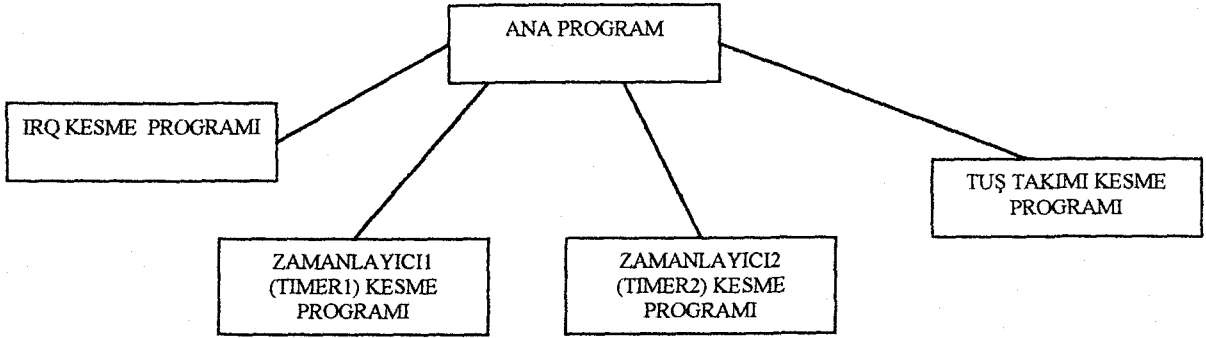


EK 3

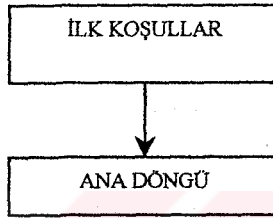
AKIŞ DİYAGRAMLARI



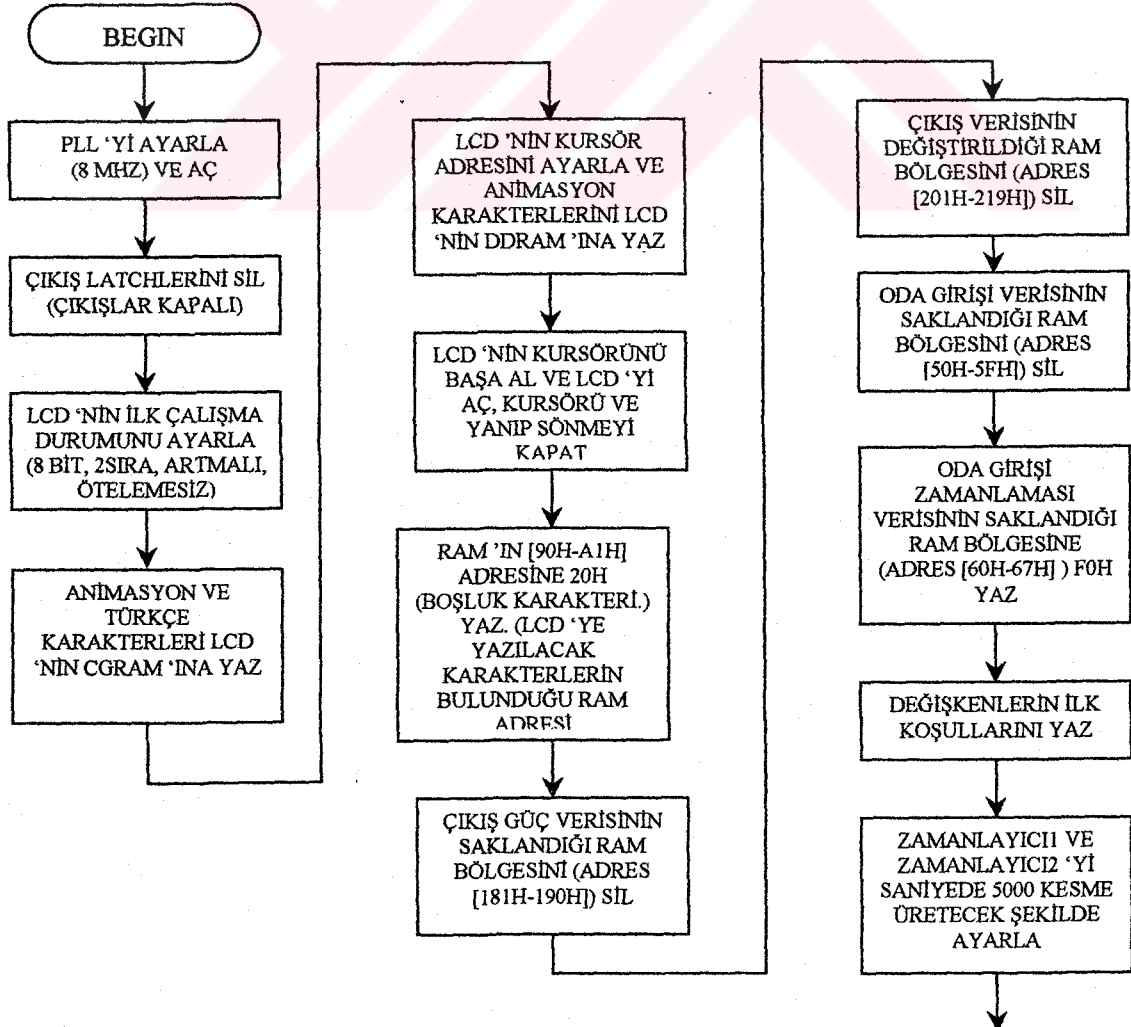
PROGRAMIN GENEL BÖLÜMLERİ



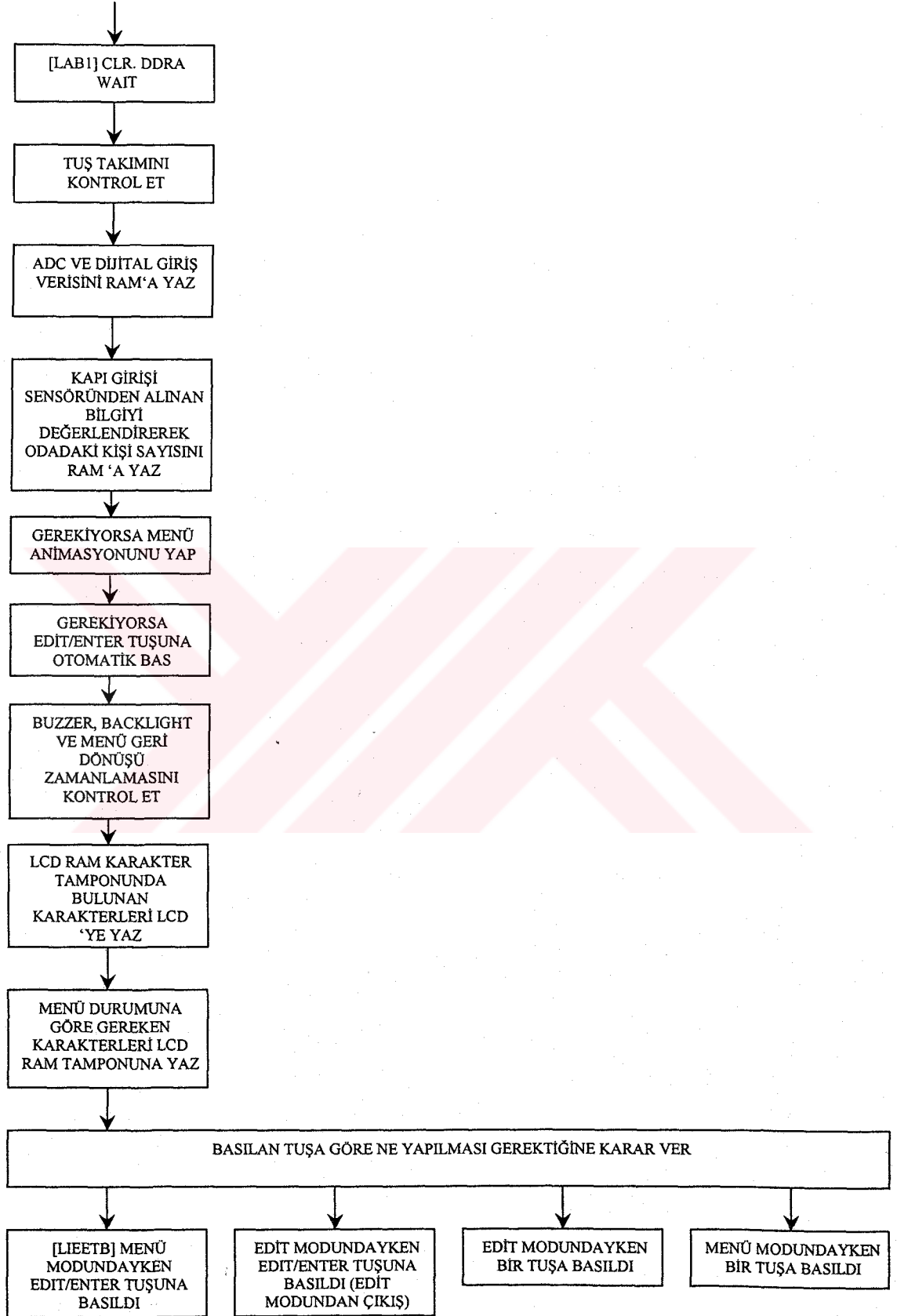
ANA PROGRAM



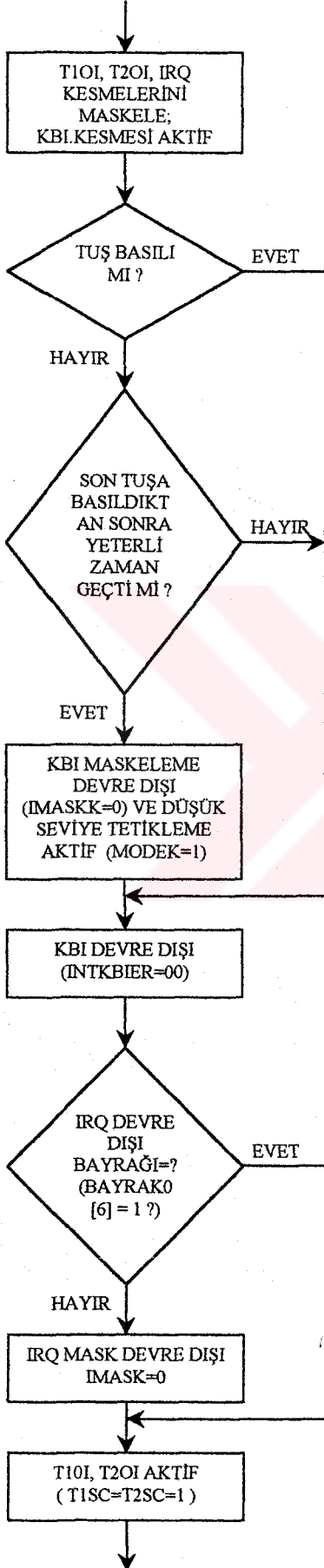
İLK KOŞULLAR



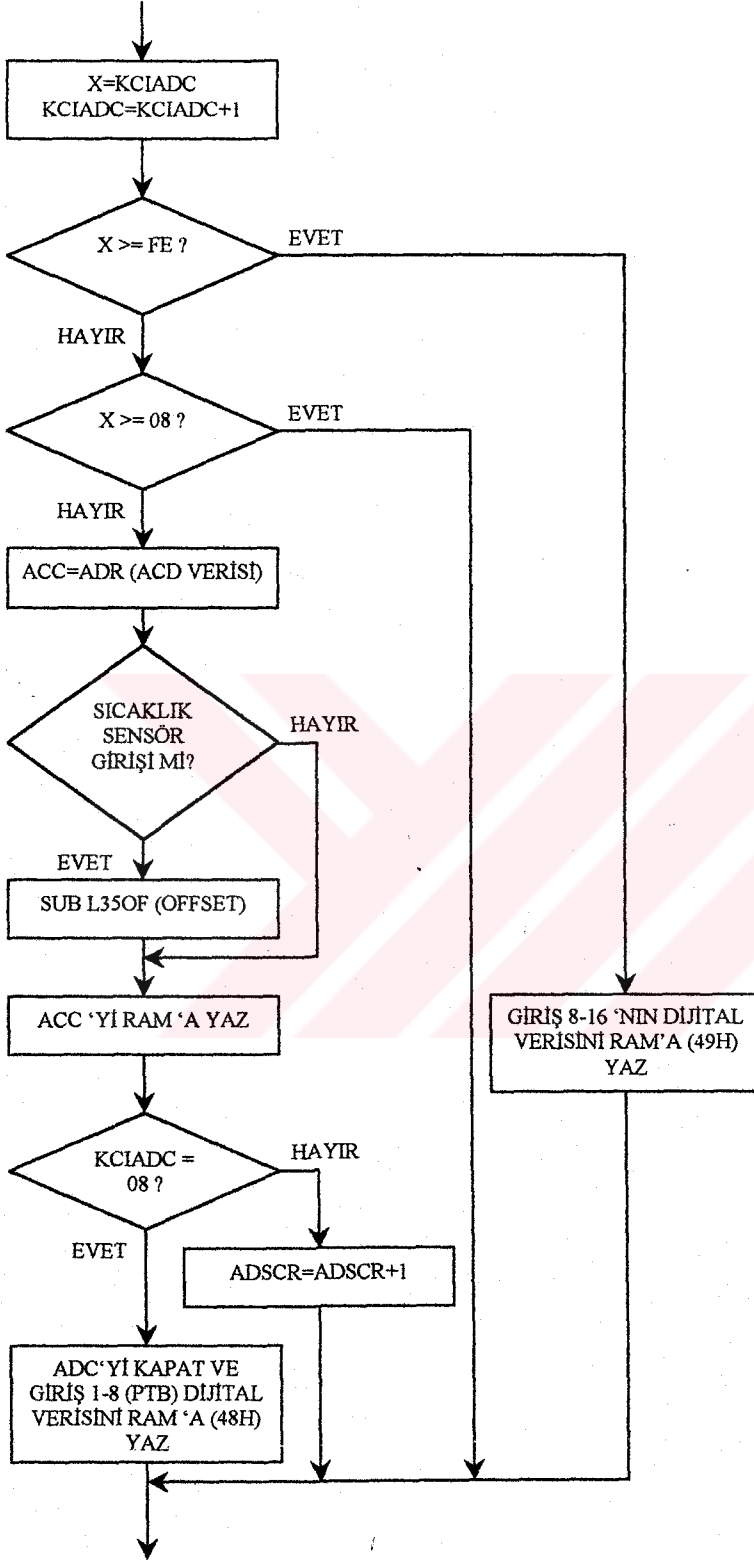
ANA DÖNGÜ



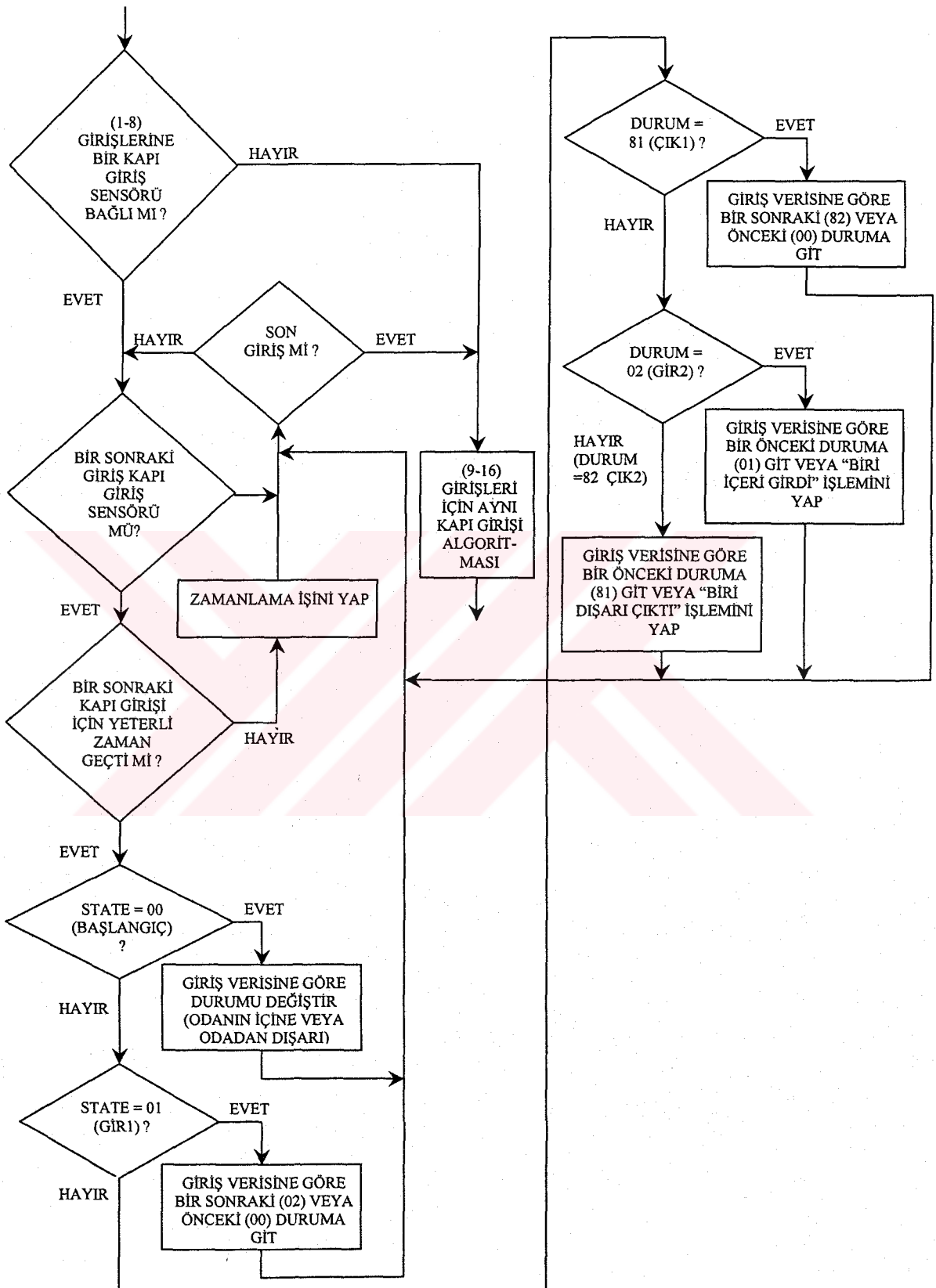
TUŞ TAKIMINI KONTROL ET



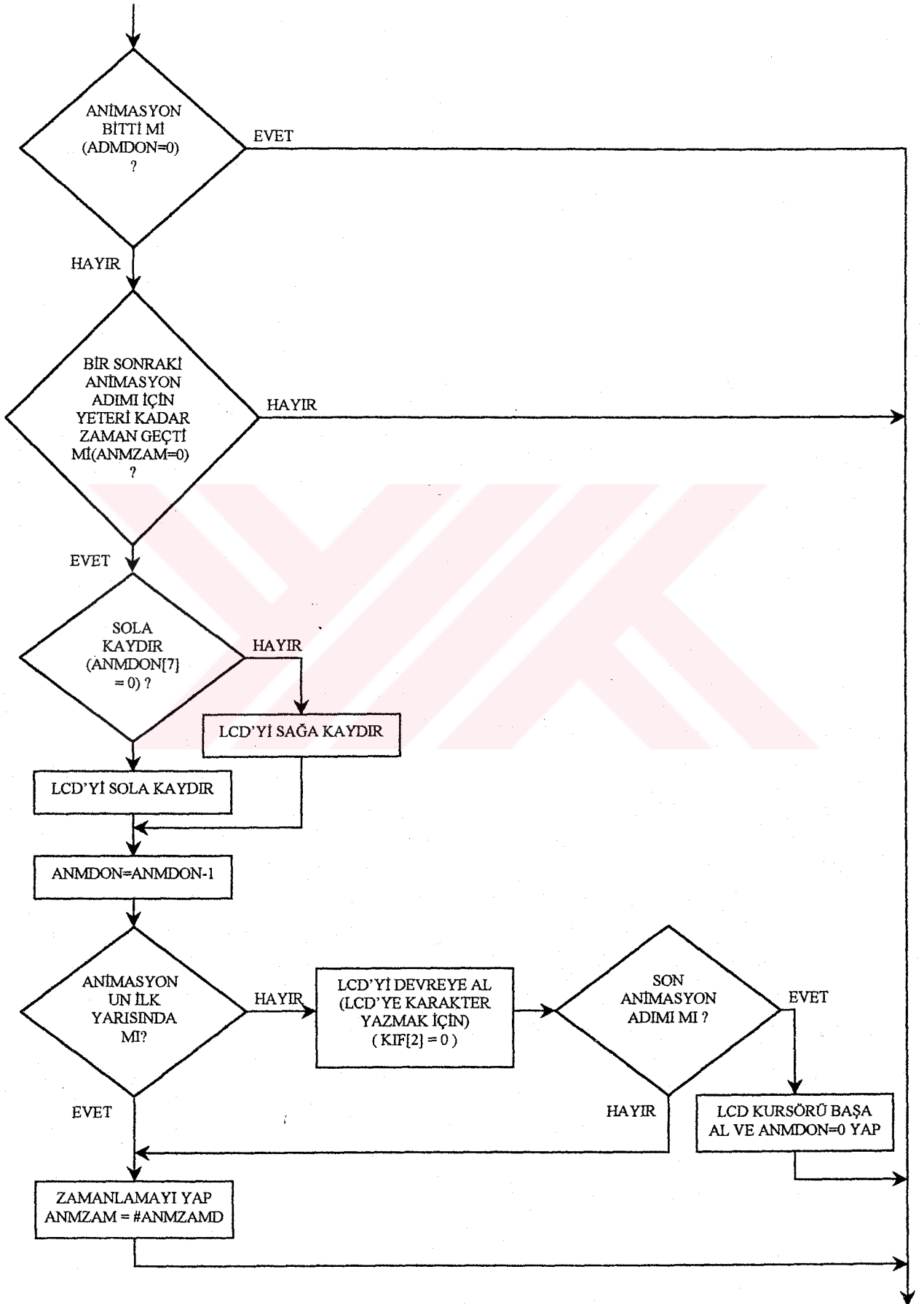
ADC VE DİJİTAL GİRİŞ VERİSİNİ RAM'A YAZ



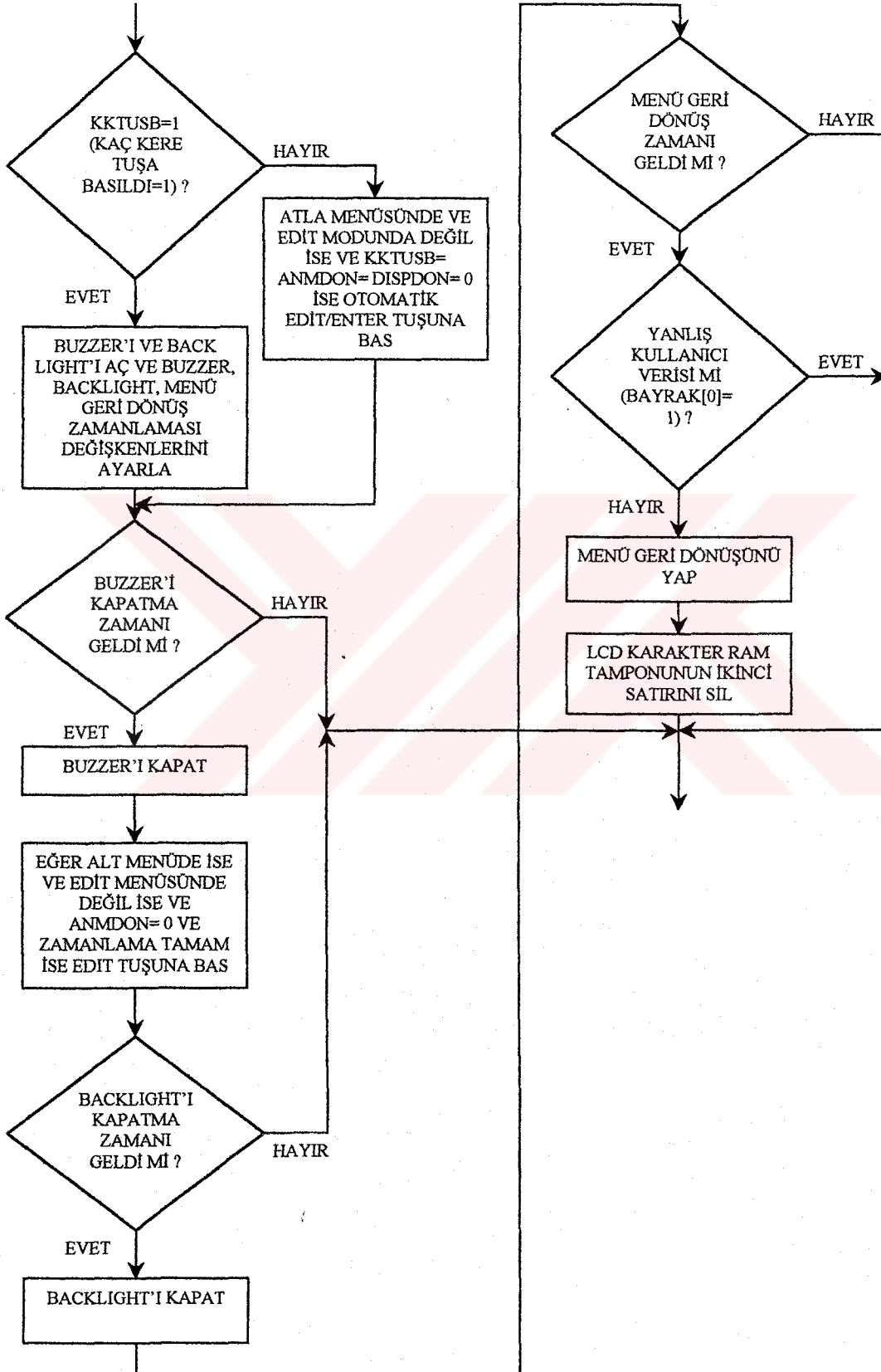
KAPI GİRİŞİ SENSÖRÜNDEN ALINAN BİLGİYİ DEĞERLENDİREREK ODADAKİ KİŞİ SAYISINI RAM'A YAZ



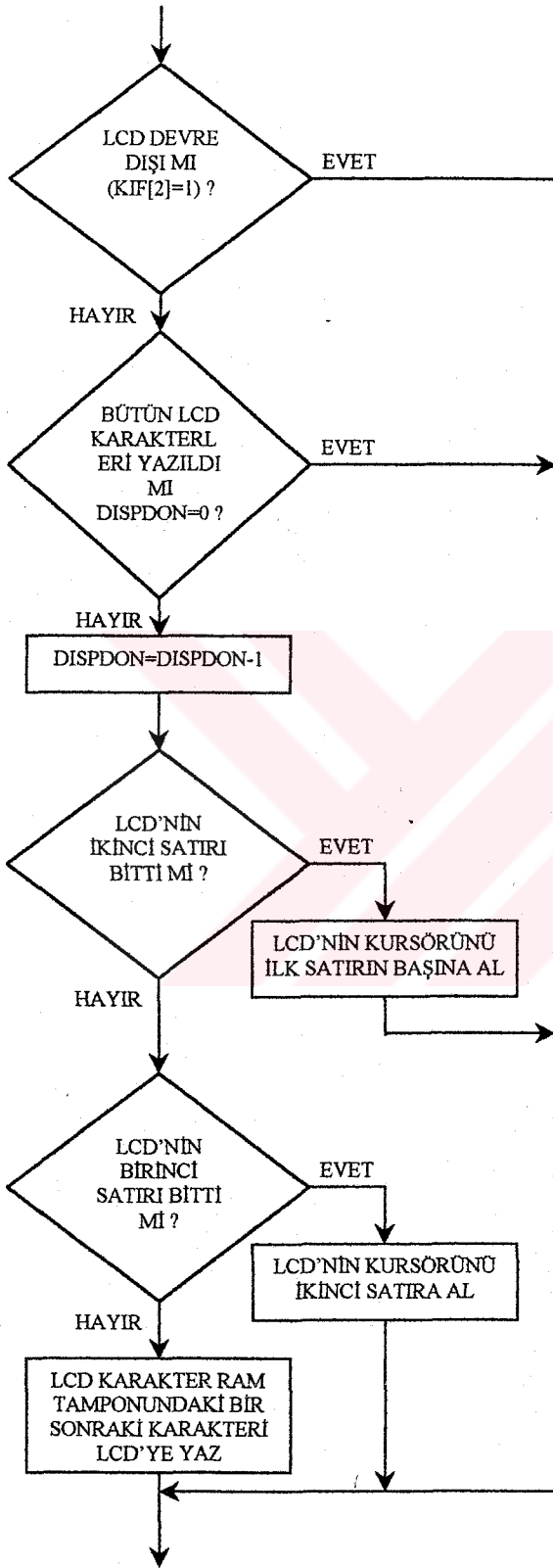
MENÜ ANİMASYONUNU YAP



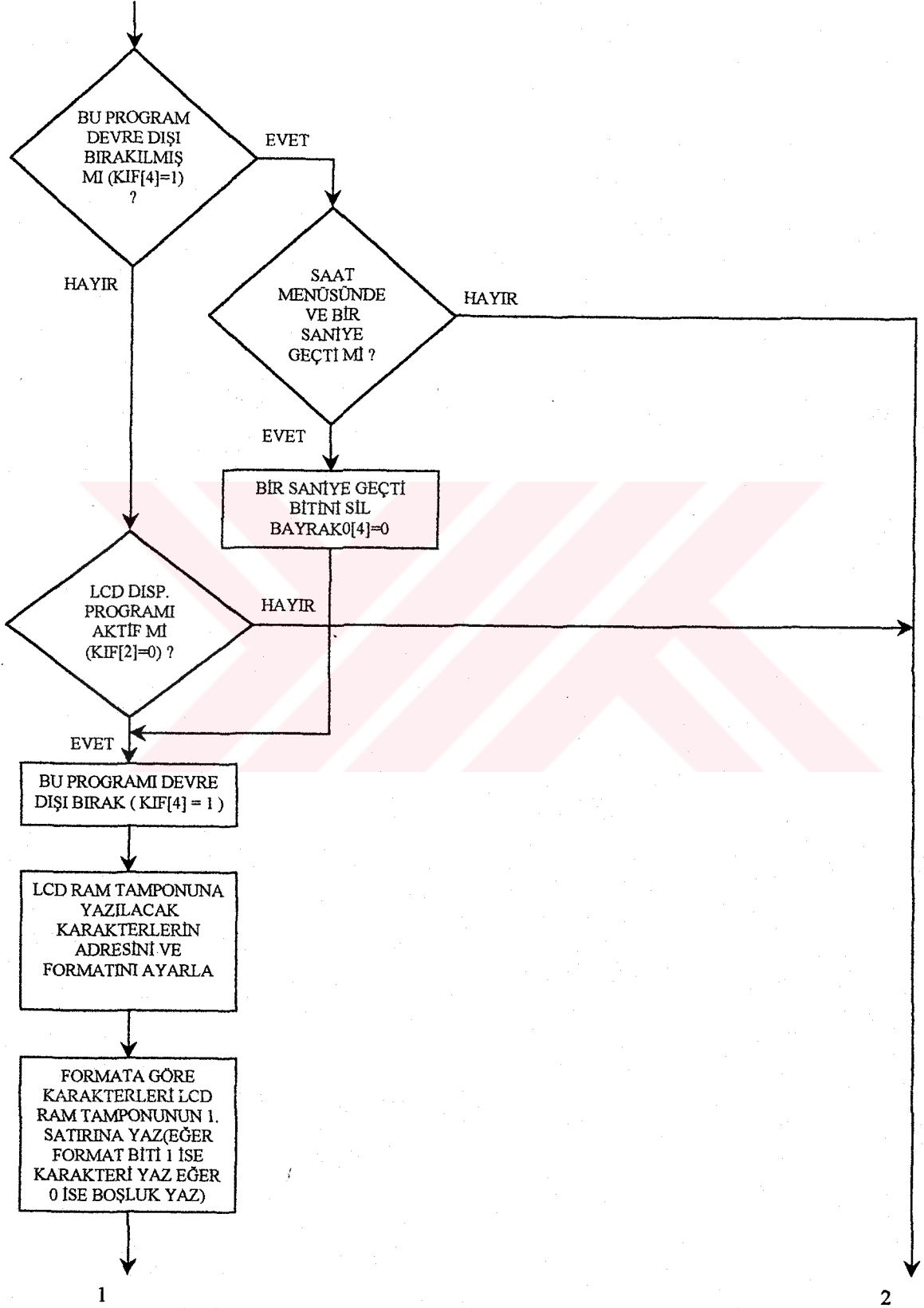
EDİT/ENTER TUŞUNA OTOMATİK BAS VE BUZZER, BACKLIGHT VE MENÜ GERİ DÖNÜŞÜ ZAMANLAMASINI KONTROL ET

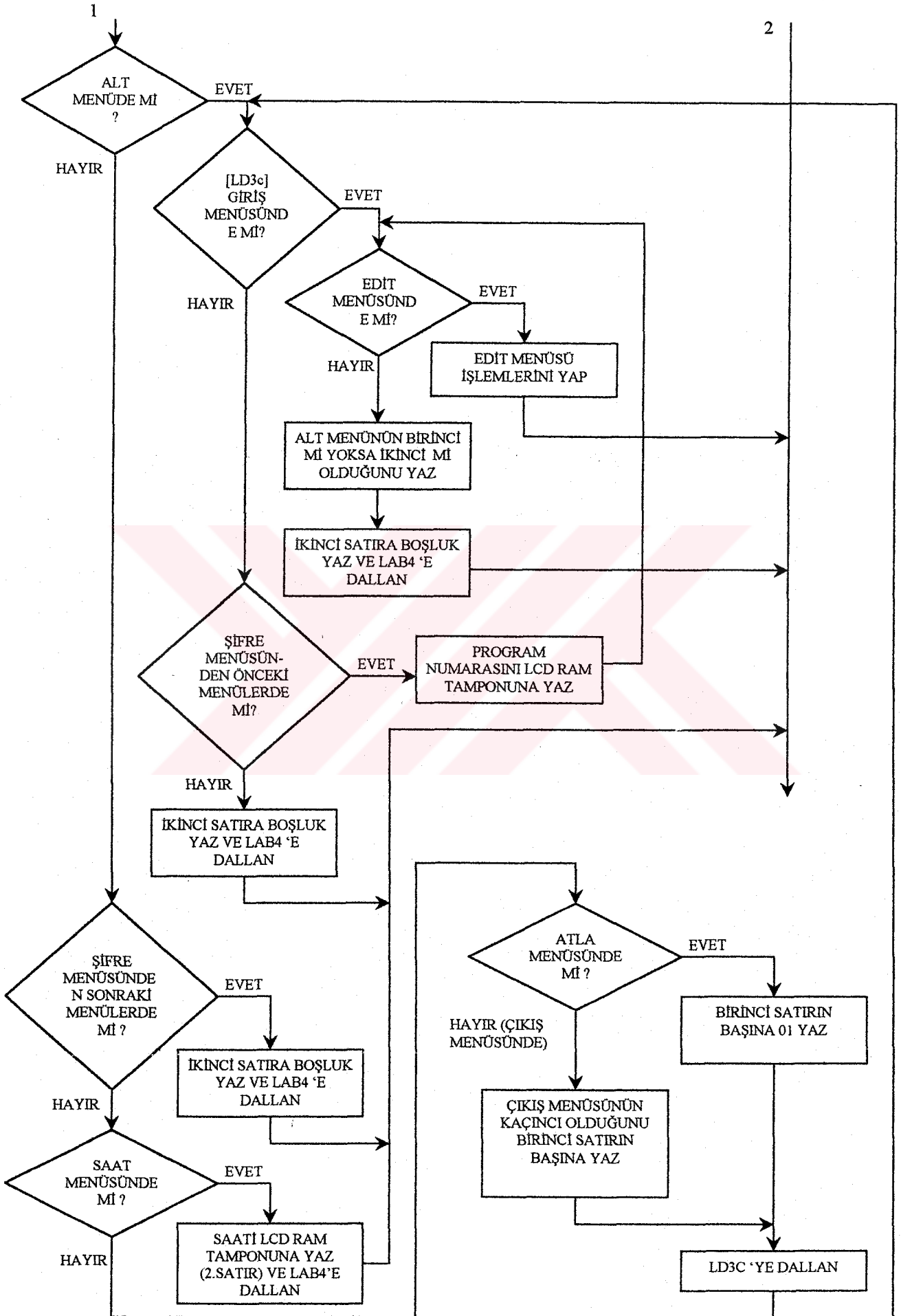


LCD RAM KARAKTER TAMPONUNDA BULUNAN KARAKTERLERİ LCD 'YE YAZ

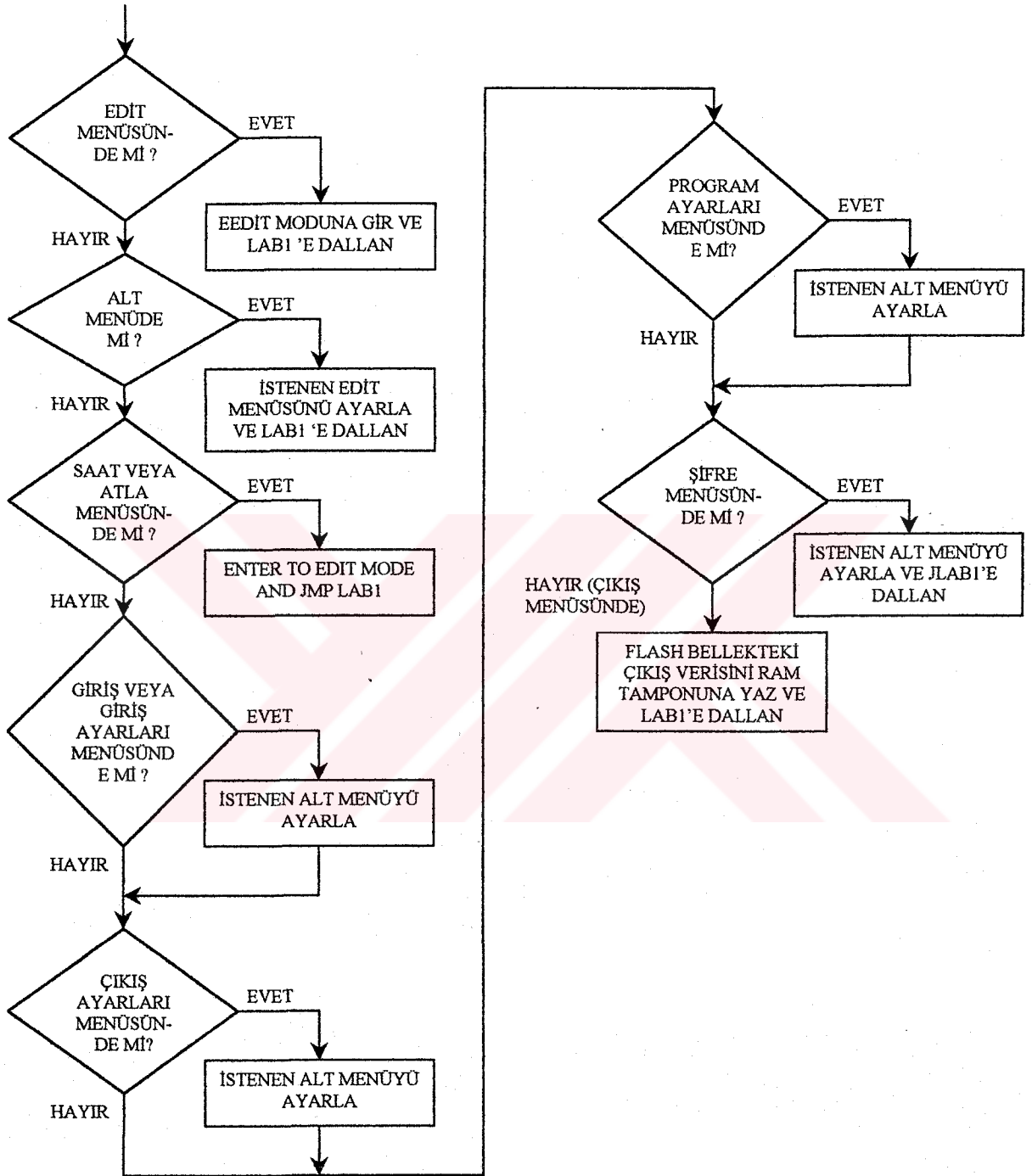


MENÜ DURUMUNA GÖRE GEREKEN KARAKTERLERİ LCD RAM TAMPONUNA YAZ

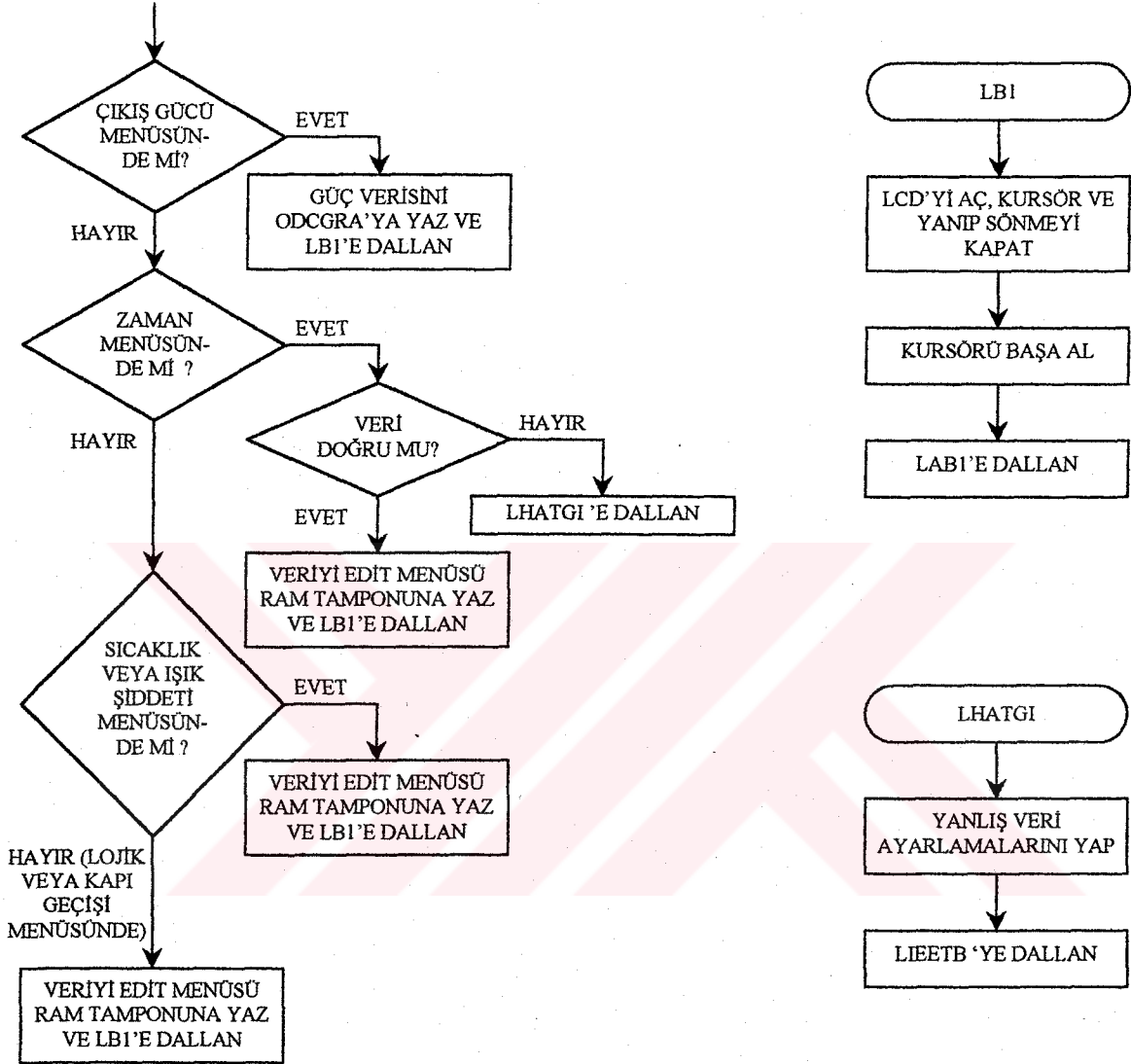




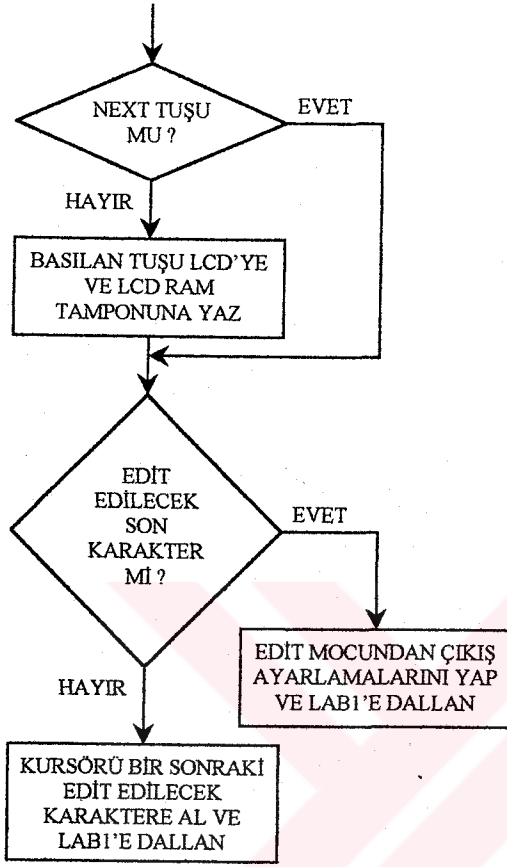
MENÜ MODUNDAYKEN EDIT/ENTER TUŞUNA BASILDI



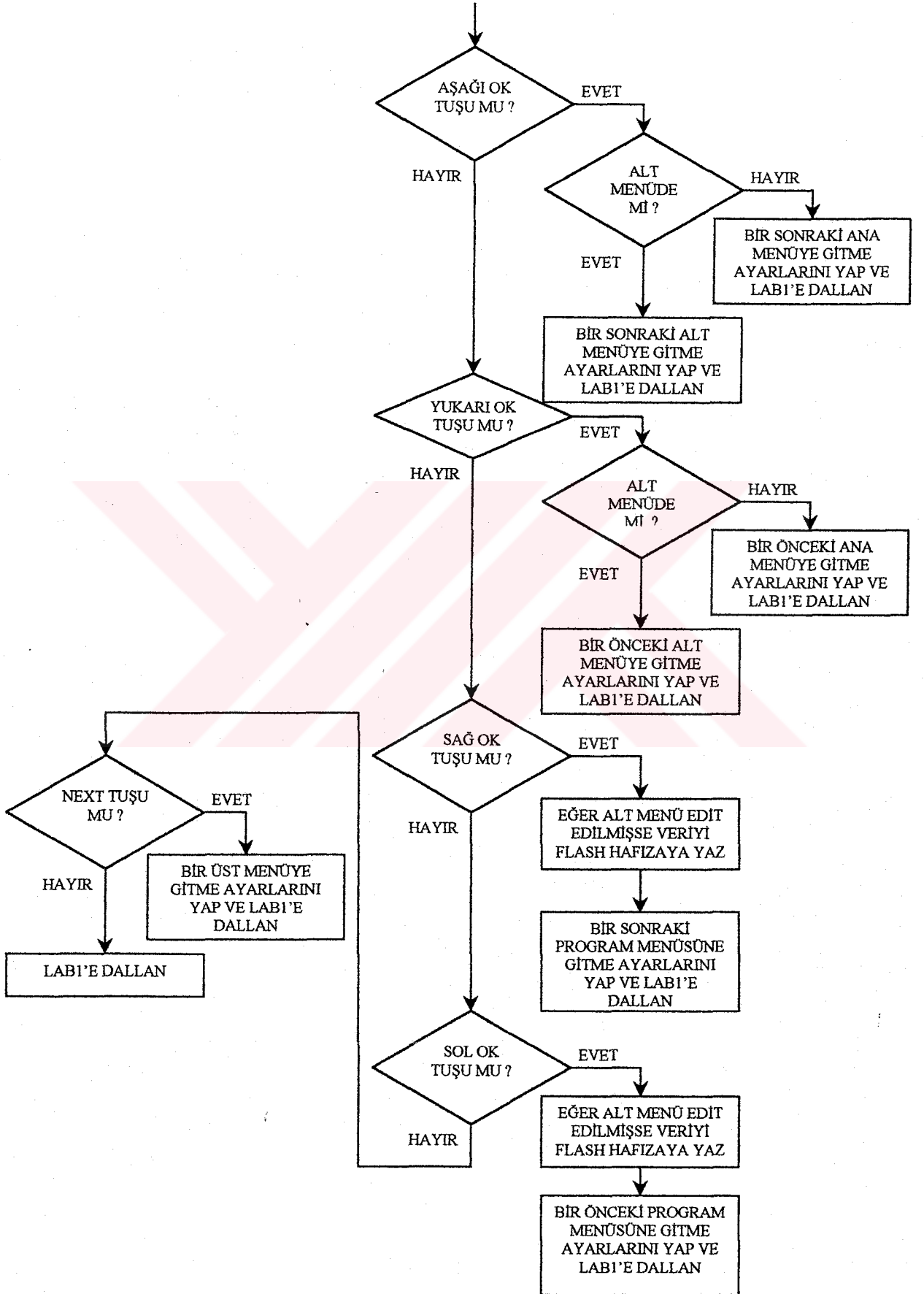
**EDİT MODUNDAYKEN EDIT/ENTER TUŞUNA BASILDI
(EDİT MODUNDAN ÇIKIŞ)**

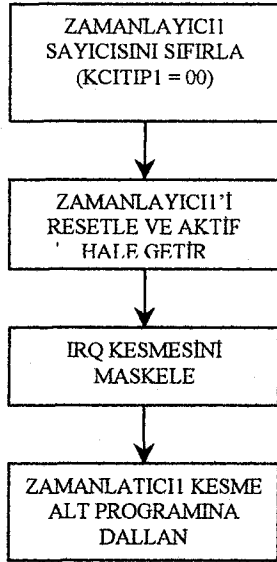
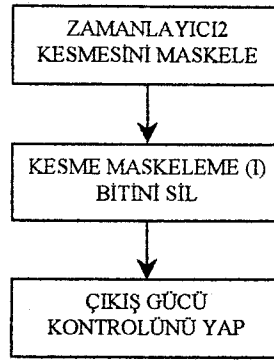
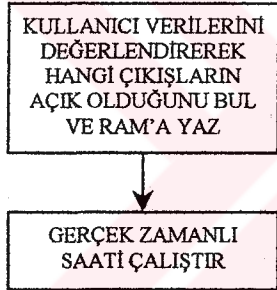
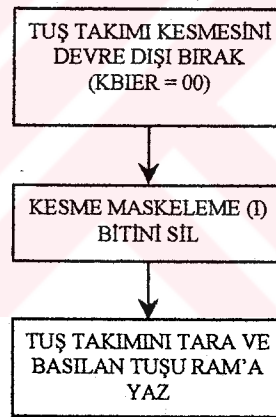


EDİT MODUNDAYKEN BİR TUŞA BASILDI



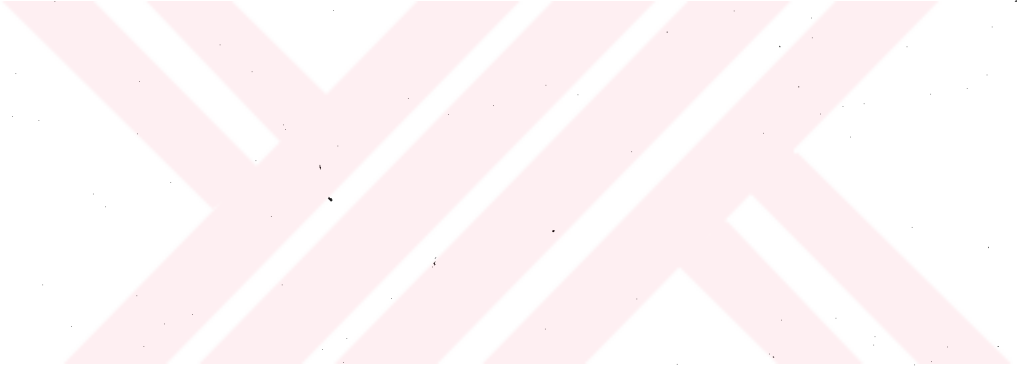
MENÜ MODUNDAYKEN BİR TUŞA BASILDI



IRQ KESME PROGRAMI**ZAMANLAYICI1 KESME PROGRAMI****ZAMANLAYICI2 KESME PROGRAMI****TUŞ TAKIMI KESME PROGRAMI**

EK 4

YAZILIMIN KAYNAK KODU



```

*****-----*****-----*****-----*****-----*****
*
* MİKRO DENETLEYİCİ KONTROLLÜ EV OTOMASYONU VE GÜVENLİK SİSTEMİ *
*
*****-----*****-----*****-----*****

```

```

*****-----*****-----*****-----*****-----*****
* Memory Equates
*****-----*****-----*****-----*****-----*****

```

```

ROMSPACE EQU $8000      ;ROM başlangıç adresi
FBRBAD EQU $0100        ;Flash Buffer RAM Baş Adr
CGRBAD EQU $0180        ;Çıkış Gücü Ramı Baş Adr
ODRBAD EQU $0200        ;Output data RAM Baş Adr
ODCGRA EQU $0211        ;Output data Çıkış Gücü Ram Adr
ODRSAD EQU $0212        ;Çıkış Data RAM Saat Adr
ODLGRA1 EQU $0216        ;Outpet Data Loj Gir Adr1
ODLGRA2 EQU $0217        ;Outpet Data Loj Gir Adr2
ODLIRA EQU $0218        ;Outpet Data Loj İşlem Adr
ODHGRA EQU $0219        ;Outpet Data Hangi Günler RAM adr
KEYINT EQU $E400        ;Keyboard kesme adresi
IRQINT EQU $E600        ;IRQ kesme adresi
ADCINT EQU $E680        ;ADC kesme adresi
TIMINT EQU $E700        ;Timer kesme adresi
TIM2INT EQU $E800        ;Timer2 kesme adresi
FADRDBA EQU $F942        ;Flash Adr Data Baş Adresi (128 byte aralıklı 16
adres)
LDCDCGRAM EQU $F980      ;LCD Char. Generator RAM
MENUF1 EQU $F9C2         ;MENUFORMAT1'lerin başlangıç adresi (Kursör yeri)
MENUCHAR EQU $FA00        ;Menu karakterlerinin yazıldığı adr. baş.
AMNUCHAR EQU $FA60        ;Alt Menu karakterlerinin yazıldığı adr. baş.
EMNUCHAR EQU $FB00        ;Alt Menü karakterlerinin yazıldığı adr. baş.
SABLON2C EQU $FB62        ;SABLON2'lerin başlangıç adr.(Karakter bşlk yeri)
AMNUSAB EQU $FBA2        ;AMNUSAB'lerin başlangıç adr.(Karakter bşlk yeri)
EMNUSAB EQU $FBD2        ;EMNUSAB'lerin başlangıç adr.(Karakter bşlk yeri)
RAMSPACE EQU $40         ;RAM başlangıç adresi
PTBD EQU $48             ;Port B Data
PTLGD EQU $49            ;Port Lojik Giriş Data
KCIADC EQU $4A           ;Kaçınıcı ADC çevrimi
ISLTMP EQU $68           ;İşlemler alt menüsünün içeriği geçici
AKTMP1I EQU $69          ;Interrupta iken (TIM2INT) Akümülatör geçici
KAAMOFST EQU $6A        ;Kaçınıcı AAmu nün yazılacağı LCDRAM yerini gösteren
Offset
KCIPRG EQU $6B           ;Kaçınıcı Pogram
CKDATA EQU $6C           ;Çıkış Datası
RLOCON1 EQU $6D          ;Röle ve Lojik Çıkış On (Açık) 1 (1-8. Çıkışlar)
RLOCON2 EQU $6E          ;Röle ve Lojik Çıkış On (Açık) 1 (9-16. Çıkışlar)
PRGACK EQU $6F           ;Programlar Arası çıkış
HXTPHI EQU $70           ;HX Geçici yüksek
HXTPLI EQU $71           ;HX Geçici düşük

;LCHDTP EQU $72          ;Latch Data Temporary

KCITIP1 EQU $72          ;Kaçınıcı Timer Interrupt Plus1
KCITI EQU $73            ;Kaçınıcı Timer Interrupt
DONGUI EQU $74           ;Kesmeler için Dongu
RLOCK1 EQU $75           ;Röle ve Lojik Çıkışları gösterir 1 den 8. çıkışa
RLOCK2 EQU $76           ;Röle ve Lojik Çıkışları gösterir 9 dan 16. çıkışa
;RLCKTP EQU $77          ;Röle ve Lojik Çıkışları gösterir Temporary geçici
RLCKTP EQU $77           ;Temporary geçici
HXTMPH EQU $78           ;HX Geçici yüksek
HXTMPL EQU $79           ;HX Geçici düşük
;FLADRH EQU $7A          ;Flash Adres yüksek
;FLADRL EQU $7B          ;Flash Adres düşük

```

OEDZAM	EQU \$7C	;Otomatik Edite Basma Zamanlaması
ANMZAM	EQU \$7D	;Animasyon Zamanlaması
ANMDON	EQU \$7E	;Animasyon Döngüsü
DSPDON	EQU \$7F	;LCD Display Döngüsü
DAKİKA0	EQU \$80	;DAKİKA LSB adresi
DAKİKA1	EQU \$81	;DAKİKA MDSB adresi
DAKİKA2	EQU \$82	;DAKİKA MSB adresi
SANIYE0	EQU \$83	;SANIYE LSB adresi
SANIYE1	EQU \$84	;SANIYE MSB adresi
SANIYE	EQU \$85	;SANIYE adresi
DAKİKA	EQU \$86	;DAKİKA adresi
SAAT	EQU \$87	;SAAT adresi
GUN	EQU \$88	;GUN adresi
BAYRAK0	EQU \$89	;BAYRAK0 adresi
KIF	EQU \$8A	;Key int. bayrağı
SICAK1	EQU \$8B	;SICAKLIK Girişlerini Gösterir (İlk 7 sini)
SICAK2	EQU \$8C	;SICAKLIK Girişlerini Gösterir (Son 7 sini)
ISIKS1	EQU \$8D	;ISIKSIDDETI Girişlerini Gösterir (İlk 7 sini)
ISIKS2	EQU \$8E	;ISIKSIDDETI Girişlerini Gösterir (Son 7 sini)
KKTUSB	EQU \$8F	;Kaç Kere Tuşa Basıldı
LCDS1D1	EQU \$8F	;LCDRAMının 1.satır adresi-1
LCDS1	EQU \$90	;LCDRAMının 1.satır adresi
LCDS1S	EQU \$9F	;LCDRAMının 1.satırın son adresi
LCDS1SP1	EQU \$A0	;LCDRAMının 1.satırın son adresi+1
LCDRAM	EQU \$A0	;LCDRAMının sıra adresi
LCDS2D1	EQU \$A0	;LCDRAMının 2.satır adresi-1
LCDS2	EQU \$A1	;LCDRAMının 2.satır adresi
LCDS2P1	EQU \$A2	;LCDRAMının 2.satır adresi+1
LCDS2S	EQU \$B0	;LCDRAMının 2.satırın son adresi
LCDS2SP1	EQU \$B1	;LCDRAMının 2.satırın son adresi+1
;P8255A	EQU \$B1	;8255 in A portunun değeri
LTCH1D	EQU \$B1	;LATCH1 data nin değeri
LCDCA	EQU \$B2	;LCD karakter adresi
LCDCA2	EQU \$B3	;LCD karakter adresi 2
LCDDA	EQU \$B4	;LCD display adresi
KAPIG1	EQU \$B5	;ISIKSIDDETI Girişlerini Gösterir (İlk 8 ini)
KAPIG2	EQU \$B6	;ISIKSIDDETI Girişlerini Gösterir (Son 8 ini)
LOJG1	EQU \$B7	;LOJIK Girişlerini Gösterir (İlk 8 ini)
LOJG2	EQU \$B8	;LOJIK Girişlerini Gösterir (Son 8 ini)
SICAMSAY	EQU \$B9	;SICAKLIK ALT menüsü sayısı
ISKAMSAY	EQU \$BA	;ISIKSIDDETI ALT menüsü sayısı
KAGAMSAY	EQU \$BB	;KAPI GIRISI ALT menüsü sayısı
LOGAMSAY	EQU \$BC	;LOJIK GIRIS ALT menüsü sayısı
KCIAAMU	EQU \$BD	;Kaçınclı Aynı Alt Menü
BUZZAM	EQU \$BE	;Buzzer çalma zamanlaması
BLGZAM	EQU \$BF	;Becklight yanma zamanlaması
MGDZAM	EQU \$C0	;Menü Geri Dönüş zamanlaması
KEYTEMP1	EQU \$C1	;KEYINT için gerekli bir bellek adr. (Tuş bulma)
KEYTEMP3	EQU \$C2	;KEYINT için gerekli bir bellek adr. (Bekleme)
KEYPR	EQU \$C3	;Basılan tuşun yazıldığı adres
XTEMP1	EQU \$C4	;X için geçici bellek adr.
XTEMP2	EQU \$C5	;X için Geçici Değer Saklama Adresi2
ASAB1	EQU \$C6	;A daki şablonu saklama adresi1
KCIMNU	EQU \$C8	;Kaçınclı menu olduğunu gösteren adr.
KCIAMU	EQU \$C9	;Kaçınclı alt menü adresi
KCIEMU	EQU \$CA	;Kaçınclı edit menüsü adresi
KCIECA	EQU \$CB	;Kaçınclı edit karakteri
FM1ADH	EQU \$CC	;Format1 adresi yüksek
FM1ADL	EQU \$CD	;Format1 adresi düşük
LTCH2D	EQU \$CE	;LATCH2 data nin değeri
FM2ADL	EQU \$CF	;Format2 adresi düşük
FM2ADH	EQU \$D0	;Format2 adresi yüksek
SABLON2	EQU \$D1	;Şablonun adresi

RECEIVED
INFORMATION SYSTEMS

```

MF2ADL EQU $D2      ;Menu format2 lerin başlangıç adresi düşük (Yazı
karakteri)
MF2ADH EQU $D3      ;Menu format2 lerin başlangıç adresi yüksek (Yazı
karakteri)
MF3ADL EQU $D4      ;Menu format3 lerin başlangıç adresi düşük (Adresleri)
MF3ADH EQU $D5      ;Menu format3 lerin başlangıç adresi yüksek
(Adresleri)
MRADL EQU $D6       ;Menu RAM adresi düşük
MRADH EQU $D7       ;Menu RAM adresi yüksek
WRITE EQU $D8       ;Yazılacak bilginin bulunduğu adr
HIADR EQU $DA       ;Yüksek adr yazıldığı adr
DONGU EQU $DB       ;DONGU nün adresi
AKMTMP1 EQU $DC     ;Akümülatör geçici adr. 1
AKMTMP2 EQU $DD     ;Akümülatör geçici adr. 2
P8255C EQU $DE      ;8255 in C portunun değeri
SM2ADH EQU $DF      ;Sablon2 Menü adres yüksek
SM2ADL EQU $E0      ;Sablon2 Menü adres düşük
WLCDCAH EQU $E1     ;LCD ye yazılacak karakter adr. yüksek
WLCDCAL EQU $E2     ;LCD ye yazılacak karakter adr. düşük
WLCDCAT EQU $E3     ;Yaz LCD karakter geçici
MCOFFSET EQU $E4    ;Menü Karakter Offseti
AMOFFSET EQU $E5    ;Alt Menü Karakter Offseti
EMOFFSET EQU $E6    ;Edit Menü Karakter Offseti

```

```

*****-*****-*****-*****-*****-*****-*****
* Register Equates
*****-*****-*****-*****-*****-*****-*****

```

```

PTA EQU $0000      ; Ports and data direction
PTB EQU $0001
PTC EQU $0002
PTD EQU $0003
DDRA EQU $0004
DDRB EQU $0005
DDRC EQU $0006
DDRD EQU $0007
PTE EQU $0008
DDRE EQU $000C

PTAPUE EQU $000D   ; Port pull-up enables
PTCPUE EQU $000E
PTDPUE EQU $000F

SPCR EQU $0010     ; SPI (Synchronous communications)
SPSCR EQU $0011
SPDR EQU $0012

SCC1 EQU $0013     ; SCI (Asynchronous communications)
SCC2 EQU $0014
SCC3 EQU $0015
SCS1 EQU $0016
SCS2 EQU $0017
SCDR EQU $0018
SCBR EQU $0019

KBSCR EQU $001A
KBIER EQU $001B
INTSCR EQU $001D
CONFIG2 EQU $001E
CONFIG1 EQU $001F

T1SC EQU $0020     ; Timer 1
T1CNTH EQU $0021

```

```

T1CNTL EQU $0022
T1MODH EQU $0023
T1MODL EQU $0024
T1SC0 EQU $0025
T1CH0H EQU $0026
T1CH0L EQU $0027
T1SC1 EQU $0028
T1CH1H EQU $0029
T1CH1L EQU $002a

```

```

T2SC EQU $002b ; Timer 2
T2CNTH EQU $002c
T2CNTL EQU $002d
T2MODH EQU $002e
T2MODL EQU $002f
T2SC0 EQU $0030
T2CH0H EQU $0031
T2CH0L EQU $0032
T2SC1 EQU $0033
T2CH1H EQU $0034
T2CH1L EQU $0035

```

```

PCTL EQU $0036 ; Phase lock loop (for crystals)
PBWC EQU $0037
PMSH EQU $0038
PMSL EQU $0039
PMRS EQU $003A
PMDS EQU $003B

```

```

ADSCR EQU $003C ; A to D converter
ADR EQU $003D
ADCLK EQU $003E

```

```
FLCR EQU $FE08
```

```

HVEN equ 3
MASS equ 2
ERASE equ 1
PGM equ 0
HVEN. equ $08
MASS. equ $04
ERASE. equ $02
PGM. equ $01

```

```

FLBPR EQU $FF7E ; Flash block protect
COPCTL EQU $FFFF ; COP (Computer operating properly) control

```

* Interrupt Vectors

```

KEY EQU $FFE0 ;Keyboard int. vektörü adresi
TIM2OV EQU $FFEC ;TIM2 overflow vektörü adresi
TIMOV EQU $FFF2 ;TIM overflow vektörü adresi
IRQ EQU $FFFA ;IRQ vektörü adresi
RESET EQU $FFFE ;Reset vektörü adresi
ADC EQU $FFDE ;ADC dönüşümü tamam vektör adresi
SWIV EQU $FFFC ;SWI Vector

```

* Constant Equates

```

;SANIYE0D EQU $00 ;SANIYE LSB data
;SANIYE1D EQU $27 ;SANIYE MSB data
;DAKIKA0D EQU $C0 ;DAKIKA LSB data

```

;DAKICA1D EQU \$27	;DAKICA MIDSb data
;DAKICA2D EQU \$09	;DAKICA MSb data
SANIYE0D EQU \$90	;SANIYE LSB data
SANIYE1D EQU \$13	;SANIYE MSb data
DAKICA0D EQU \$E0	;DAKICA LSB data
DAKICA1D EQU \$93	;DAKICA MIDSb data
DAKICA2D EQU \$04	;DAKICA MSb data
LCDE EQU \$02	;LCD aktif
LCDDE EQU \$1D	;LCD pasif
RD EQU \$1E	;Oku
WR EQU \$17	;Yaz
RW EQU \$16	;Hem oku hem yaz
NRW EQU \$09	;Okuma, yazma yapma
LTCH1E EQU \$04	;LATCH1 aktif
LTCH2E EQU \$10	;LATCH2 aktif
LTCH1DE EQU \$1B	;LATCH1 pasif
LTCH2DE EQU \$0F	;LATCH2 pasif
A8255 EQU \$08	;8255 A portu aktif
B8255 EQU \$09	;8255 B portu aktif
C8255 EQU \$0A	;8255 C portu aktif
CO8255 EQU \$0B	;8255 kontrol kaelimesi aktif
RAME EQU \$30	;RAM aktif
YUOKT EQU \$32	;Yukarı ok tuşu
SOOKT EQU \$34	;Sol ok tuşu
SAOKT EQU \$36	;Sağ ok tuşu
ASOKT EQU \$38	;Aşağı ok tuşu
EDIT EQU \$3C	;EDIT tuşu
NEXT EQU \$3A	;NEXT tuşu
MF1ADL EQU \$C0	;Menu format1 lerin başlangıç adresi düşük (Yazı var yok)
MF1ADH EQU \$F9	;Menu format1 lerin başlangıç adresi yüksek (Yazı var yok)
M1LCDAL EQU \$00	;Menü 1 adres düşük
M1LCDAH EQU \$FA	;Menü 1 adres yüksek
MS2ADL EQU \$60	;Menü şablon 2 adres düşük
MS2ADH EQU \$FB	;Menü şablon 2 adres yüksek
AMLCDALEQU \$60	;Alt Menü Karakter adres düşük
AMLCDAH EQU \$FA	;Alt Menü Karakter adres yüksek
AMS2ADL EQU \$A0	;Alt Menü Şablon adres düşük
AMS2ADH EQU \$FB	;Alt Menü Şablon adres yüksek
EMLCDAL EQU \$00	;Edit Menü Karakter adres düşük
EMLCDAH EQU \$FB	;Edit Menü Karakter adres yüksek
EMS2ADL EQU \$D0	;Edit Menü Şablon 2 adres düşük
EMS2ADH EQU \$FB	;Edit Menü Şablon 2 adres yüksek
MNUSAY EQU \$17	;Menü sayısı=0F
AMNUSY EQU \$08	;Alt Menu Sayısı=8
MNUSTB EQU \$16	;Menü Sınır Tuşa Basımı (16H 22D)
CMBSAY EQU \$03	;Çıkış menüsü başlangıç sayısı
CMBSYD1 EQU \$02	;Çıkış menüsü başlangıç sayısı-1
SMNBSY EQU \$01	;SAAT menusu başlangıç sayısı
ATMBSY EQU \$02	;Arttırma menüsü başlangıç sayısı
SCMSY EQU \$12	;Son Çıkış Menüsü Sayısı
;SCMSYP1 EQU \$13	;Son Çıkış Menüsü Sayısı+1
SFMBYS EQU \$13	;Sifre Menüsü başlangıç sayısı
;SFMBSYPI EQU \$14	;Sifre Menüsü başlangıç sayısı+1
GAYMBSY EQU \$14	;Giriş Ayarları Menüsü Başlangıç Sayısı
CAYMBSY EQU \$15	;Çıkış Ayarları Menüsü Başlangıç Sayısı
PAYMBSY EQU \$16	;Program Ayarları Menüsü Başlangıç Sayısı
GRMBSY EQU \$17	;Girişler Menüsü Başlangıç Sayısı
CAMSTB EQU \$03	;Çıkış Alt Menüsü Sınır Tuşa Basımı
SAMBSY EQU \$08	;Sifre Alt Menüsü başlangıç sayısı
SAMBSYG EQU \$09	;Sifre Alt Menüsü başlangıç sayısı Gerçek
SAMSSY EQU \$0B	;Sifre Alt Menüsü Son sayısı

```

SAMSTB EQU $02 ;Şifre Alt Menüsü Sınır Tuşa Basımı
GAAMBSY EQU $02 ;Giriş Ayarları Alt Menüsü Başlangıç Sayısı
GAAMBSYG EQU $03 ;Giriş Ayarları Alt Menüsü Başlangıç Sayısı Gerçek
CAAMBSY EQU $0B ;Çıkış Ayarları Alt Menüsü Başlangıç Sayısı
CAAMBSYG EQU $0C ;Çıkış Ayarları Alt Menüsü Başlangıç Sayısı Gerçek
PAAMBSY EQU $0D ;Program Ayarları Alt Menüsü Başlangıç Sayısı
PAAMBSYG EQU $0E ;Program Ayarları Alt Menüsü Başlangıç Sayısı Gerçek
GAAMSSY EQU $06 ;Giriş Ayarları Alt Menüsü Son sayısı
CAAMSSY EQU $0D ;Çıkış Ayarları Alt Menüsü Son sayısı
GAAMSTB EQU $03 ;Giriş Ayarları Alt Menüsü Sınır Tuşa Basımı
CAAMSTB EQU $01 ;Çıkış Ayarları Alt Menüsü Sınır Tuşa Basımı
AMDND EQU $28;2A ;ANMDON data anim. dongü sayısı (Alt oka basılırsa)
AMDND2 EQU $15;16 ;ANMDON data anim. dongü sayısının yarısı + 1
AMDD2 EQU $A8;AA ;ANMDON data anim. don. say.+80 (Üst oka basılırsa)
AMDD22 EQU $95;96 ;ANMDON data anim. dongü sayısının yarısı + 80h + 1
AMZMD2 EQU $40 ;ANMZMN data animasyon zamanlama sayısı (Yavaş)
CGEMBSY EQU $01 ;Çıkış Gücü Edit Menüsü Başlangıç Sayısı
SAENBSY EQU $02 ;SAAT Edit Menüsü Başlangıç Sayısı
SIEMBSY EQU $03 ;Sıcaklık Edit Menüsü Başlangıç Sayısı
ISEMBSY EQU $04 ;Işık Şiddeti Edit Menüsü Başlangıç Sayısı
KGEMBSY EQU $05 ;Kapı Girişi Edit Menüsü Başlangıç Sayısı
LGEMBSY EQU $06 ;Lojik Giriş Edit Menüsü Başlangıç Sayısı
LIEMBSY EQU $07 ;Lojik İşlem Edit Menüsü Başlangıç Sayısı
OEDZAMD EQU $A0 ;Otomatik Edite Basma Zamanlaması Datası
FADRDH EQU $F9 ;Flash Adr Data High
FADRDL EQU $40 ;Flash Adr Data Low
L35OF EQU $01 ;Sıcaklık Offseti
;L35OF EQU $65 ;Sıcaklık Offseti

```

* Çıkış Programları

ORG \$A000

; Çıkış1

```

FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001

```

; Çıkış2

```

FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001

```

; Çıkış3

```

FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001

```

; Çıkış4

```

FDB $0000,$0008,$0000,$0001,$0101,$0100,$8001,$4001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001

```

[illegible]

```

FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
; Çıkış12
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
; Çıkış13
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
; Çıkış14
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
; Çıkış15
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
; Çıkış16
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001
FDB $0000,$0008,$0000,$0001,$0101,$0100,$0001,$0001

```

* 128 Byte Aralıklarla Adres Değerleri

```

ORG FADRDBA ;Flash Adr Data Baş Adresi
FDB $A000,$A080,$A100,$A180
FDB $A200,$A280,$A300,$A380
FDB $A400,$A480,$A500,$A580
FDB $A600,$A680,$A700,$A780

```

* LCD nin CGRAM dataları

```

ORG LCDCGRAM
FDB $0E00,$0E10,$1711,$0E00 ;Ğ (05)
FDB $0400,$0E04,$0404,$0E00 ;İ (06)
; FDB $0E11,$100E,$0115,$0E00 ;Ş (03)
FDB $0F10,$0E01,$1E04,$0C00 ;$ (07)

```

* Kursör duraklarını gösterir

```

ORG  MENUF1
FCB  $DB      ;İkinci satır 1. yarı formatı SAAT menüsü
FCB  $10      ;İkinci satır 2. yarı formatı SAAT menüsü
FDB  $C000    ;1.satır formatı ATLA menüsü
FDB  $3000    ;2.satır formatı ÇIKIŞ alt menüsü
FDB  $D9B0    ;2.satır formatı SAAT alt menüsü
FDB  $36D8    ;2.satır formatı SICAKLIK alt menüsü
FDB  $36D8    ;2.satır formatı ISIK SIDDETI alt menüsü
FDB  $2490    ;2.satır formatı KAPI GIRISI alt menüsü
FDB  $2490    ;2.satır formatı LOJIK GIRIS alt menüsü
FDB  $2220    ;2.satır formatı LOJIK ISLEMLER alt menüsü
;      FDB  $2040 ;2.satır formatı LOJIK ISLEMLER alt menüsü
FDB  $AAA8    ;2.satır formatı HANGI GÜNLER alt menüsü

```

* Menü yazıları

```

ORG  MENUCHAR
FDB  $5341,$4154,$479A      ;SAAT GÜ
FCB  $4E                    ;N
FDB  $2E80,$494B,$4907      ;.ÇIKIS
FDB  $0706,$4652,$4544,$4505,$0607 ;SIFRE DEĞİŞ
FCB  $54,$06,$52,$4D,$45    ;TİRME
FDB  $4706,$5206,$0741,$5941 ;GIRIS AYA
FCB  $52,$4C,$41,$52,$49    ;RLARI
FDB  $8049,$4B49,$0741,$5941 ;ÇIKIS AYA
FCB  $52,$4C,$41,$52,$49    ;RLARI
FDB  $5052,$4F47,$5241,$4D41,$5941 ;PROGRAM AYA
FCB  $52,$4C,$41,$52,$49    ;RLARI
FDB  $4706,$5206,$074C,$4552 ;GIRISLER

```

* Alt menü yazıları

```

ORG  AMNUCHAR
FDB  $8049,$4B49,$0747,$9A43 ;ÇIKIS GÜC
FCB  $9A                      ;Ü
FDB  $5341,$4154              ;SAAT
FDB  $5349,$4341,$4B4C,$494B ;SICAKLIK
FDB  $4907,$494B,$0706,$4444,$4554 ;ISIK SIDDET
FCB  $06                      ;I
FDB  $4B41,$5049,$4706,$5206,$0706 ;KAPI GIRISI
FDB  $4C4F,$4A06,$4B47,$0652,$0607 ;LOJIK GIRIS
;      FDB  $4C4F,$4A49,$4B49 ;LOJIK I
FDB  $0607,$4C45,$4D4C,$4552    ;ISLEMLER
FDB  $4841,$4E47,$0647,$9A4E,$4C45 ;HANGI GÜNLE
FCB  $52 ;R

```

```

FDB  $4B55,$4C4C,$414E,$4943 ;KULLANIC
FCB  $49                      ;I
FDB  $5945,$544B,$064C        ;YETKIL
FCB  $06                      ;I
FDB  $479A,$5645,$4E4C,$064B ;GÜVENLIK

```

```

FDB  $429A,$549A,$4E80,$494B,$4907 ;BÜTÜN ÇIKIŞ
FCB  $4C,$41,$52                  ;LAR
FDB  $8049,$4B49,$074C,$4152      ;ÇIKIŞLAR

```

```

FDB  $5031,$3233,$3435,$3637 ;P1 2 3 4 5 6 7
FCB  $38                      ; 8

```

* Edit menüsü yazıları

```

ORG  EMNUCHAR
FDB  $8049,$4B49,$0747,$9A43 ;ÇIKIS GÜC
FCB  $9A                      ;Ü

```

```

FDB $4180,$494B,$4B41,$5041,$4C49 ;AÇIK KAPALI
FDB $5347,$4747 ;SG G G
FCB $47 ;G
FDB $4947,$4747 ;IG G G
FCB $47 ;G
FDB $4B47,$4747 ;KG G G
FCB $47 ;G
FDB $4C47,$4747 ;LG G G
FCB $47 ;G
FDB $5341,$4154,$4752,$0750,$524F ;SAAT GRŞ PRO
FCB $47 ;G
; FDB $4706,$5206,$0750,$524F,$4752 ;GIRIS PROGR
; FDB $414D ;AM
FDB $5074,$5361,$8061,$5065,$4375 ;PtSaÇaPeCu
FDB $4374,$5061 ;CtPa

```

* Menü karakterlerinin yerlerini gösterir

```

ORG SABLON2C
FCB $78 ;Birinci satır 1. yarı formatı
FCB $38 ;Birinci satır 2. yarı formatı
FDB $3F00 ;.ÇIKIŞ
FDB $FBFF ;SIFRE DEĞİŞTİRME
FDB $FBFC ;GIRIS AYARLARI
FDB $FBFC ;ÇIKIS AYARLARI
FDB $FEFF ;PROGRAM AYARLARI
FDB $FF00 ;GIRISLER

```

* Alt menü karakterlerinin yerlerini gösterir

```

ORG AMNUSAB
FCB $FB ;İkinci satır 1. yarı formatı
FCB $C0 ;İkinci satır 2. yarı formatı
FDB $F000 ;SAAT
FDB $FF00 ;SICAKLIK
FDB $F7F0 ;ISIK SIDDET
FDB $F7E0 ;KAPI GIRIS
FDB $FBEO ;LOJIK GIRIS
FDB $FF00 ;ISLEMLER
FDB $FBFO ;HANGI GÜNLER

FDB $FF80 ;KULLANICI
FDB $FE00 ;YETKIN
FDB $FF00 ;GÜVENLIK

FDB $FBFC ;BÜTÜN ÇIKIŞLAR
FDB $FF00 ;ÇIKIŞLAR

FDB $D555 ;P1 2 3 4 5 6 7 8

```

* Edit menüsü karakterlerinin yerlerini gösterir

```

ORG EMNUSAB
FDB $FBC0 ;ÇIKIS GÜCÜ
FDB $F1F8 ;AÇIK KAPALI
FDB $C920 ;SG G G G
FDB $C920 ;IG G G G
FDB $C920 ;KG G G G
FDB $C920 ;LG G G G
FDB $F778 ;SAAT GRŞ PROG
; FDB $FBF8 ;GIRIS PROGRAM
FDB $FFFC ;PtSaÇaPeCuCtPa

```

```

*****_*****_*****_*****_*****_*****_*****
* Start of Program Code *
*****_*****_*****_*****_*****_*****_*****

```

```

      ORG    ROMSPACE
      JMP    BEGIN          ;BEGIN e dallan
; Flash1 Silen (128 byte) RAM Programı
      ;LDA    FLCR
      ;ORA    #00000010Q    ;ERASE=1
      ;STA    FLCR          ;FLCR=A
FLSHSIL LDA    FLBPR        ;A=FLBPR
      STA    ,X            ;(X)=A
      MOV    #$0C,DONGU    ;DONGU=Ch
LFF1   DBNZ   DONGU,LFF1    ;10us GECİK
      LDA    FLCR          ;A=FLCR
      ORA    #00001000Q    ;HEVEN=1
      STA    FLCR          ;HEVEN=1
      MOV    #$05,AKMTMP1  ;AKMTMP1=5
LFF3   MOV    #$FF,DONGU    ;DONGU=Ch
LFF2   DBNZ   DONGU,LFF2
      DBNZ   AKMTMP1,LFF3  ;1MS GECİK
      AND    #11111101Q    ;ERASE=0
      STA    FLCR          ;ERASE=0
      MOV    #$06,DONGU
LFF4   DBNZ   DONGU,LFF4    ;5us GECİK
      AND    #11110111Q    ;HVEN i sil
      STA    FLCR
      MOV    #$2,DONGU
LFF5   DBNZ   DONGU,LFF5    ;1us GECİK
      RTS

; Flasha Yazan (128 byte) RAM Programı
      ;LDA    FLCR
      ;ORA    #00000001Q    ;PGM=1
      ;STA    FLCR          ;FLCR=A
FLSHYAZ LDA    FLBPR        ;A=FLBPR
      STA    ,X            ;(X)=A
      MOV    #$0C,DONGU    ;DONGU=Ch
LFG1   DBNZ   DONGU,LFG1    ;10us GECİK
      LDA    FLCR          ;A=FLCR
      ORA    #00001000Q    ;HEVEN=1
      STA    FLCR          ;HEVEN=1
      MOV    #$06,DONGU    ;DONGU=06
LFG2   DBNZ   DONGU,LFG2    ;5US GECİK
      TXS                    ;SP=H:X-#$01
      AIS    #$01          ;SP=SP+1
      LDHX   #$0080        ;H:X=0080
LFG4   LDA    $FF,X        ;A=(SP)
      STA    $00,SP        ;(SP)=A
      AIS    #-1t          ;SP=SP-1
      MOV    #$23,DONGU    ;DONGU=Ch
LFG3   DBNZ   DONGU,LFG3    ;30us gecik
      DBNZX  LFG4          ;X-1=0 değilse LFG4 e dallan
      LDA    FLCR          ;A=FLCR
      AND    #11111110Q    ;PGM=0
      STA    FLCR          ;ERASE=0
      MOV    #$06,DONGU
LFG5   DBNZ   DONGU,LFG5    ;5us GECİK
      AND    #11110111Q    ;HVEN i sil
      STA    FLCR
      MOV    #$2,DONGU
LFG6   DBNZ   DONGU,LFG6    ;1us GECİK
      LDX    #$FC          ;X=FC
      TXS                    ;SP=H:X-1 (FD)
      RTS

```

; RAM Programlarını RAM a yazar

```

WRAMPGM LDHX HXTMPH ;H:X=HXTMPH:HXTMPL
        LDA ,X ;A=,X
        INC HXTMPL ;HXTMPL=HXTMPL+1
        LDHX XTEMP1 ;H:X=XTEMP1:XTEMP2
        STA ,X ;,X=A
        INC XTEMP2 ;XTEMP2=XTEMP2+1
        DBNZ DONGU,WRAMPGM ;DONGU-1=0 değil ise WRAMPGM e dallan
        RTS ;Alt prog geri dön

```

```

BEGIN MOV #$00,CONFIG1 ;LVI devrede ve 3V
      MOV #$03,CONFIG2
; PLL yi ayarlar
      BCLR 5,PCTL ;PLL off
      MOV #$02,PCTL ;P=0 E=2
      MOV #$D0,PMRS ;L=D0H
      MOV #$03,PMSH ;N yüksek=03H
      MOV #$D1,PMSL ;N düşük=D1H
      MOV #$01,PMDS ;R=1
      BSET 7,PBWC ;Otomatik band genişliği aktif
      BSET 5,PCTL ;PLL on

```

```

; LATCH leri sil
      CLR PTA ;PTA=0
      MOV #$FF,DDRA ;PTA çıkış
      MOV #$09,PTC ;PTC=9 RD=WR=1
      MOV #$1F,DDRC ;PTC çıkış
      BSR WLTCH1 ;LATCH1 e yaz
      BSR WLTCH2 ;LATCH2 ye yaz

      BSET 4,PCTL ;PLL i saat kaynağı olarak seç BCS=1

```

```

; LCD Function Set i Ayarlar
      MOV #$38,PTA ;LCD 2 satır ve 8 bit
      JSR WMLCD ;LCD ye yaz

```

```

; LCD Entry Mode Set i ayarla
      MOV #$06,PTA ;PTA=02
      BSR WMLCD2 ;WMLCD2 alt prog dallan

```

```

; LCD CGRAMA YAZ
      MOV #$40,PTA ;CGRAM Adr. Belirler
      BSR WMLCD2 ;WMLCD2 alt prog dallan
      MOV #$01,PTA ;PTA=1F (Sağda bir çizgi)
      BSR WDLCD2X8 ;WMLCD2X8 alt prog dallan
      MOV #$48,PTA ;CGRAM Adr. Belirler
      BSR WMLCD2 ;WMLCD2 alt prog dallan
      MOV #$03,PTA ;PTA=1F (Sağda bir kalın çizgi)
      BSR WDLCD2X8 ;WMLCD2X8 alt prog dallan
      MOV #$07,PTA ;PTA=1F (Sağda bir kalın çizgi2)
      BSR WDLCD2X8 ;WMLCD2X8 alt prog dallan
      MOV #$0F,PTA ;PTA=1F (Sağda bir kalın çizgi3)
      BSR WDLCD2X8 ;WMLCD2X8 alt prog dallan
      MOV #$1F,PTA ;PTA=1F (Bütün Bitler dolu)
      BSR WDLCD2X8 ;WMLCD2X8 alt prog dallan

```

```

; Türkçe Karakterleri LCD CGRAM a yazar
      LDHX #LCDCGRAM ;HX=LCDCGRAM
      MOV #$18,DONGU ;DONGU=18h
LCGR1 MOV X+,PTA ;PTA=X ,X=X+1 (İ , Ş ve Ğ)
      BSR WDLCD2 ;WMLCD2 alt prog dallan
      DBNZ DONGU,LCGR1 ;DONGU-1 0 değilse LCGR1 e dallan
      CLRH ;H=0

```

```

; LCD yi siler
      MOV #$01,PTA ;PTA=01
      BSR WMLCD2 ;WMLCD2 alt prog dallan

```

```

;          JMP    LANM3          ;Diğer ANİM için

; LCD yi Animasyom için ayarlar
MOV    #$2,ANMDON      ;ANMDON=2
LDA    #$90            ;LCD nin DDRAM adr. yaz (ilk)
LANM1  STA    PTA        ;LCD nin DDRAM adr. yaz
BSR    WMLCD2          ;WMLCD2 alt prog dallan

LDX    #$05            ;X=04
MOV    #$00,PTA        ;PTA=00
MOV    #$04,DONGU      ;DONGU=0C
BRA    LANM2C          ;LANM2c ye dallan
LANM2b MOV    #$05,DONGU  ;DONGU=0C
LANM2C BSR    WDLCD3X     ;WDLCD3X alt prog dallan
INC    PTA              ;PTA=PTA+1
DBNZX  LANM2b          ;X-1=0 değilse LANM2b ye dallan

DEC    ANMDON          ;ANMDON=ANMDON-1
BEQ    LLANM3          ;ANMDON=0 ise LANM3 e dallan
LDA    #$D0            ;A=D0
BRA    LANM1           ;LANM1 e dallan

LLANM3  JMP    LANM3

* LATCH1 e yaz Giriş:Yok Çıkış:Yok Değişenler:A
WLTC1  LDA    PTC        ;A=PTC
EOR    #LTCH1E          ;LATCH1 aktif
STA    PTC              ;PTC=A
EOR    #LTCH1E          ;LATCH1 pasif
STA    PTC              ;PTC=A
RTS

* LATCH2 ye yaz Giriş:Yok Çıkış:Yok Değişenler:A
WLTC2  LDA    PTC        ;A=PTC
EOR    #LTCH2E          ;LATCH2 aktif
STA    PTC              ;PTC=A
EOR    #LTCH2E          ;LATCH2 pasif
STA    PTC              ;PTC=A
RTS

* WMLCD2 yi 8 kere tekrarlar
WDLCD2X8 MOV    #$08,DONGU ;DONGU=08h
WDLCD3X BSR    WDLCD2     ;WDLCD2 alt prog dallan
DBNZX  DONGU,WDLCD3X     ;DONGU-1 0 değilse WDLCD3X8 e dallan
RTS    ;Alt programdan geri dön

* LCD Uygun mu diye bakar ve WMLCDye dallanır
WMLCD2 BSR    LCDA        ;LCD yi bekle
BSR    WMLCD            ;LCD ye yaz
RTS    ;Alt programdan geri dön

* LCD Uygun mu diye bakar ve WDLCDye dallanır
WDLCD2 BSR    LCDA        ;LCD yi bekle
BSR    WDLCD            ;LCD ye yaz
RTS    ;Alt programdan geri dön

* LCD ye yaz Giriş:Yok Çıkış:Yok Değişenler:A
; LCD ye Modu/Komut yazar RS=0 R/W=0
WMLCD  LDA    PTC        ;A=PTC
AND    #RW              ;LCD RS=0 R/W=0
BRA    WLCD            ;WLCD ye dallan

; LCD ye data yaz
WDLCD  LDA    PTC        ;A=PTC

```

```

        AND    #WR                ;LCD R/W=0
; LCD ye hehengi bir şey yazmak için ör. RS=0 ise
WLCD    STA    PTC                ;PTC=A
        ORA    #LCDE              ;LCD aktif
        STA    PTC                ;PTC=A
        AND    #LCDDE             ;LCD disable
;        SUB    #LCDE              ;LCD pasif
        STA    PTC                ;PTC=A
        ORA    #NRW               ;Okuma, yazma yapma
        STA    PTC                ;PTC=A
        RTS                      ;Alt programdan geri dön

* LCD işlem yapmaya müsait mi diye bakar
LCDA    LDA    PTC                ;A=PTC
        CLR    DDRA               ;A portu giriş
LLCDA    AND    #LCDDE             ;LCD disable
        STA    PTC                ;PTC=A
        CLI
        AND    #RD                ;LCD nin BF sini oku
        STA    PTC                ;PTC=A
        ORA    #LCDE              ;LCD 'yi aktif yap
        SEI
        STA    PTC                ;PTC=A
        BRSET 7,PTA,LLCDA         ;BF 1 ise LCDA ya dallan
        AND    #LCDDE             ;LCD 'yi pasif yap
        STA    PTC                ;PTC=A
        CLI
        ORA    #NRW               ;Read Write yapma
        STA    PTC                ;PTC=A
        MOV    #$FF,DDRA          ;PTA çıkış
        RTS

; LCD yi Cursor At Home'a ayarlar
LANM3    MOV    #$02,PTA          ;PTA=02
        BSR    WMLCD2             ;WMLCD2 alt prog dallan
; LCD Display ON/OFF u ayarlar
        MOV    #$0C,PTA          ;LCD modu D açık C,B kapalı
        BSR    WMLCD2             ;WMLCD2 alt prog dallan
; LCD RAMA Boşluk yaz
        LDX    #$21               ;Xlye 33 yaz
        LDA    #$20               ;Aya 20H yaz
LAB0     STA    LCDS1D1,X          ;LCDS1-1+X e boşluk yaz
        DBNZX LAB0                ;X-1=0 değilse LAB0 'adallan
; Çıkış Gücü RAM Datasına Boşluk Yaz
        LDX    #$10               ;X=80
        CLRA                      ;A=0
LAB0b    STA    $0180,X            ;(X+FF)=0
        DBNZX LAB0b              ;X-1=0 değilse LAB0b 'adallan
; İşlem Gören Çıkış Data RAMına Boşluk Yaz
        LDX    #$19               ;X=80
LAB0c    STA    $0200,X            ;(X+FF)=0
        DBNZX LAB0c              ;X-1=0 değilse LAB0c 'adallan
; Kapı Girişi Verilerinin İlk Koşullarını Ayarlar
        LDX    #$10               ;X=10h
LAB0d    STA    $4F,X              ;İçeride Kaç Kişi Var=Durumu=00
        DBNZX LAB0d              ;X-1=0 değilse LAB0d 'adallan
; Kapı Girişi Zamanını Ayarlar
        LDA    #$F0               ;A=80h
        LDX    #$08               ;X=08h
LAB0e    STA    $5F,X              ;İçeride Kaç Kişi Var=Durumu=00
        DBNZX LAB0e              ;X-1=0 değilse LAB0e 'adallan
; RAM VE Register ların ilk durumlarını ayarlar

```

```

CLR   SANIYE
CLR   SANIYE0
CLR   SANIYE1
CLR   DAKIKA
CLR   DAKIKA0
CLR   DAKIKA1
CLR   DAKIKA2
CLR   BAYRAK0      ;BARYAK0 '1 sıfırla
CLR   KCIEMU       ;KCIEMU=0
CLR   KCIAMU       ;KCIAMU=0
CLR   MCOFFSET     ;Menu Karakter Offset=0
CLR   AMOFFSET     ;Alt Menu Karakter Offset=0
CLR   EMOFFSET     ;Edit Menu Karakter Offset=0
CLR   KKTUSB       ;KKTUSB=0
CLR   LTCH1D       ;LTCH1D=0
CLR   LTCH2D       ;LTCH2D=0
CLR   ANMDON       ;ANMDON=0
CLR   MGDZAM       ;MGDZAM=1
CLR   KCITI        ;KCITI=0
CLR   KCIADC       ;KCIADC=0
CLR   PRGACK       ;PRGACK=0
CLR   PTD          ;PTD=00
CLR   HXTPLI       ;HXTPLI=00
CLR   PTBD         ;PTBD=00
CLR   PTLGD        ;PTLGD=00

LDA   #$01
STA   GUN          ;Pazartesi
STA   SAAT         ;SAAT 1
STA   KCIMNU       ;KCIMNU=1
STA   KCIPRG       ;KCIPRG=1
STA   KCIAAMU      ;KCIAAMU=1
STA   BUZZAM       ;BUZZAM=1
STA   BLGZAM       ;BLGZAM=1
STA   KCITIP1      ;KCITIP1=1

MOV   #LCDS1,LCDRAM ;LCDRAM '1 başa al
MOV   #$30,SICAK1  ;SICAK1=0011 0000
MOV   #$00,SICAK2  ;SICAK2=0000 0000
MOV   #$0F,ISIKS1  ;ISIKS1=0000 1111
MOV   #$00,ISIKS2  ;ISIKS2=0000 0000
MOV   #$80,KAPIG1  ;KAPIG1=1000 0000
MOV   #$00,KAPIG2  ;KAPIG2=0000 0000
MOV   #$00,LOJG1   ;LOJG1=0000 0000
MOV   #$FF,LOJG2   ;LOJG2=1111 1111
MOV   #$01,SICAMSAY ;SICAMSAY=01
MOV   #$01,ISKAMSAY ;ISKAMSAY=01
MOV   #$01,KAGAMSAY ;KAGAMSAY=01
MOV   #$02,LOGAMSAY ;LOGAMSAY=02
MOV   #MS2ADH,SM2ADH ;SM2ADH=Menu sab.2 adr. yüksek
MOV   #LCDS1,LCDCA2 ;LCDCA2=LCD RAMının 1. satır adr.
MOV   #MF1ADH,FM1ADH ;FM1ADH=Menu format1 adr. yüksek
MOV   #$22,ANMZAM   ;ANMZAM=22h
MOV   #$4A,DSPDON   ;DSPDON=44h
MOV   #OEDZAMD,OEDZAM ;OEDZAM=OEDZAMD
MOV   #11111000Q,RLOCK1 ;1-5. ÇIKIŞ Röle veya Lojik Çıkış
MOV   #11111000Q,RLOCK2 ;9-13. ÇIKIŞ Röle veya Lojik Çıkış
MOV   #$70,ADCLK    ;ADCLK=70h ADCSaati=Bus Frekansı/8
MOV   #$F8,DDR      ;PTD[3-7] output (Değeri=0)
MOV   #$A0,HXTPHI   ;HXTPHI=A0h

```

; Timer1 Saniyede 5000 kere (200us de bir) int. üretsin

```
MOV   #$06,T1MODH   ;T1MODH=06
```

```

MOV    #$3F,T1MODL    ;T1MODL=3FH
MOV    #$40,T1SC      ;TOIE aktif
; Timer2 Saniyede 5000 kere (200us de bir) int. üretsın
MOV    #$06,T2MODH    ;T2MODH=06
MOV    #$3F,T2MODL    ;T2MODL=3FH
MOV    #$40,T2SC      ;TOIE aktif

CLR    KEYTEMP1        ;KEYTEMP1 i sil
CLR    KIF              ;KIF 1 sil
CLR    ADSCR            ;ADSCR=0

LAB1    CLR    DDRA      ;PTA giriş
        WAIT

        STA    COPCTL    ;COP sayıcısını sil
        CLR    PTA        ;PTA=0
        MOV    #$78,DDRA  ;PTA[0-2] giriş, PTA[3-6] çıkış
        BCLR   6,T2SC     ;TOIE=0 Timer Int disadle
        BCLR   6,T1SC     ;TOIE=0 Timer Int disadle
        BSET   1,INTSCR    ;IMASK1=1 IRQ int disable
        MOV    #$07,KBIER  ;PTA[0-2] için key. int. aktif
        BSET   2,KBSCR    ;int. req. sil
        BRSET  3,KBSCR,LAB4ib ;KEYF 1(Tuş Bırakılmadı) ise LAB4i ye dallan
        DBNZ   KEYTEMP3,LAB4ib ;KEYTEMP3 0 değilse LAB4i ye dallan
        MOV    #$01,KBSCR  ;imask devre dışı ve 0 seviye tetiklemeli
LAB4ib  CLR    KBIER      ;KEYINT devre dışı
        BRSET  6,BAYRAK0,LIDE1 ;BAYRAK0[6]=1 ise LIDE1 e dallan
        BCLR   1,INTSCR    ;IMASK1=0 IRQ int eneble
LIDE1   ;BRSET  5,BAYRAK0,LIDE2 ;BAYRAK0[5]=1 ise LIDE2 e dallan
        BSET   6,T1SC     ;TOIE=1 Timer Int eneble
LIDE2   BSET   6,T2SC     ;TOIE=1 Timer Int eneble
        CLRH

* ADC Girişlerini (PTB0-PTB7) ve LOjik Girişleri RAMa Okur
; PTB0-PTB7=$0040-$0047 PTBD=$0048 PTLGD=$0049
        LDX    KCIADC      ;X=KCIADC
        INC    KCIADC      ;KCIADC=KCIADC+1
        CPX    #$FE        ;X i FE ile karşılaştır
        BHS    LADC2       ;FE den yüksek veya aynı ise LADC2 ye dallan
        CPX    #$08        ;X i 08 ile karşılaştır
        BLO    LADC0       ;Düşük ise LADC0'a dallan
        RORX           ;X'i sağa döndür
        BCS    LADC1       ;Tek ise LADC1 ye dallan
        BRA    LADC2       ;(Çift ise) LADC2 ye dallan

; Sıcaklık girişi mi diye bakar
LADC0   LDA    #$8         ;A=8h
        SUB    KCIADC      ;A=A-KCIADC
        TAX           ;X=A
        MOV    KCIADC,AKMTMP1 ;AKMTMP1=KCIADC
        LDA    SICAK1      ;A=SICAK1
LADC3   RORA          ;A yı sağa döndür
        DBNZ   AKMTMP1,LADC3 ;AKMTMP1-1=0 değil ise LADC3 e dallan
        LDA    ADR         ;A=ADR
        BCC    LADC4       ;C=0 ise LADC4 e dallan (Sıcaklık girişi
değil)
        SUB    #L350F      ;Sıcaklık girişi ise LM35 offseti çıkar
LADC4   STA    $40,X       ;(X+40)=A(yani ADR)
        LDX    KCIADC      ;X=KCIADC
        CBEQX  #$08,LADC1  ;X=07 ise LADC1 e dallan
        INC    ADSCR       ;ADSCR=ADSCR+1
        BRA    LAB4i       ;LAB4i ye dallan
LADC1   MOV    #$1F,ADSCR  ;ADSCR=1Fh ADC kapalı(ADC Power OFF)
        LDA    PTB         ;A=PTB

```

```

COMA                                ;A nın Tersini al
STA PTBD                            ;PTBD=A
BRA LAB4i                           ;LAB4i ye dallan
LADC2 CLR DDRA                      ;DDRA=00 PTA=Giriş
CLR ADSCR                           ;ADSCR=00
LDA PTA                             ;A=PTA
AND #$F8                           ;A=A AND F8h
ADD PTD                             ;A=A+PTD
COMA                                ;A nın tersini al
STA PTLGD                           ;PTLGD
BRA LAB4i                           ;LAB4i ye dallan

LLAB4ii JMP LAB4ii                  ;LAB4i ye dallan
LLKG4 JMP LKG4                      ;LKG4 e dallan
LLKG4c JMP LKG4c                   ;LKG4c ye dallan

; Veya iken Hepsi 0 VE iken hepsi 1 Bağlı değil
; Kapı Girişini Değerlendir
; (1-8) Girişler
LAB4i LDA KAPIG1                    ;A=Bağlı olduğu lojik ve kapı girişleri (1-8)
BEQ LLAB4ii                         ;Lojik Girişe Bağlı Değil ise LAB4ii ye dallan
CLR AKMTMP2                         ;RLCKTPI=00
MOV #$04,AKMTMP1                   ;DONGUI=04
; CLRH                               ;H=00
LKG2 INC AKMTMP2                    ;RLCKTP=RLCKTP+1 (TEMP)
ROLA                                ;A yı sola çevir
STA XTEMP1                         ;AKTMP1I=A (TEMP Geçici)
BCS LKG1                            ;C=1 is LKG1 e dallan
LKG1b LDA XTEMP1                   ;A=AKTMP1I
ROLA                                ;A yı sola çevir
DBNZ AKMTMP1,LKG2                  ;DONGUI-1=0 değil ise LKG2 ye dallan
BRA LLAB4ii                        ;LAB4ii ye dallan

LKG1 LDX AKMTMP2                   ;X=RLCKTPI
LDA $5F,X                          ;A=(5F+X) KG Zamanlaması
CMP #$F0                           ;80h ile karşılaştır
BNE LLKG4                          ;Eşit değil ise LKG4 e dallan
LDA $57,X                          ;A=(X+57) Kaçınıcı durumda olduğunu A ya yaz
BEQ LLKG4c                         ;00. Durumda ise LKD4c e dallan
CBEQA #$01,LKG5                    ;01. Durumda ise LKD5 e dallan
CBEQA #$81,LKG6                    ;81. Durumda ise LKD6 e dallan
CBEQA #$02,LKG7                    ;02. Durumda ise LKD7 e dallan
; CBEQA #$82,LKG8                  ;82. Durumda ise LKD8 e dallan
* 82. Durumda ise
LKG8 JSR LKG8c                     ;LKG8c alt programına dallan
LKG8b AND #11000000Q               ;A=A AND C0
CBEQA #$C0,LKG8d                   ;LKG8d ye dallan (Çıkan Oldu)
CBEQA #$40,LKG8e                   ;LKG8e ye dallan (bir önceki duruma al)
;LKG8a BRCLR 2,ISLTMP,SSSKPLI3 ;0 (ve) ise SKPLI (Kapalı) ya dallan
BRA LKG1b                          ;LKG1b ye dallan (Aynı konumda)
; Çıkan Oldu
LKG8d ;LDX RLCKTP                  ;X=RLCKTPI
CLR $57,X                          ;(57+X)=00 (DURUM=00)
LDA SANIYE                         ;A=SANIYE
STA $5F,X                          ;(5F+X)=SANIYE
DEC $4F,X                          ;(4F+X)=(4F+X)-1 (İçerideki Kişi Sayısı)
BPL LKG8g                          ;Pozitif ise LKG8g ye dallan
CLR $4F,X                          ;(4F+X)=00 (İçerideki Kişi Sayısı)

LKG8g BRA LKG1b                    ;LKG1b ye dallan
; Bir Önceki Duruma al
LKG8e ;LDX RLCKTP                  ;X=RLCKTPI

```

```

LDA    #$81          ;Bir önceki konum
STA    $57,X         ;(57+X)=81 (Bir Önceki Konum)
BRA    LKG1b         ;LKG1b ye dallan

```

* 02. Durumda ise

```

LKG7    BSR    LKG8c      ;LKG8c alt programına dallan
        AND    #11000000Q ;A=A AND C0
        CBEQA  #$C0,LKG7d ;LKG8d ye dallan (Giren Oldu)
        CBEQA  #$80,LKG7e ;LKG8e ye dallan (bir önceki duruma al)
        BRA    LKG1b      ;LKG1b ye dallan (Aynı konumda)

; Giriş Oldu
LKG7d    ;LDX    RLCKTP    ;X=RLCKTPI
        CLR    $57,X      ;(57+X)=00 (DURUM=00)
        LDA    SANIYE     ;A=SANIYE
        STA    $5F,X      ;(5F+X)=SANIYE
        INC    $4F,X      ;(4F+X)=(4F+X)+1 (İçerideki Kişi Sayısı)
        BRA    LKG1b      ;LKG1b ye dallan

; Bir Önceki Duruma al
LKG7e    ;LDX    RLCKTP    ;X=RLCKTPI
        LDA    #$01       ;Bir önceki konum
        STA    $57,X      ;(57+X)=81 (Bir Önceki Konum)
        BRA    LKG1b      ;LKG1b ye dallan

```

* 81. Durumda ise

```

LKG6    BSR    LKG8c      ;LKG8c alt programına dallan
        AND    #11000000Q ;A=A AND C0
        CBEQA  #$C0,LKG6d ;LKG6d ye dallan (Bir önceki duruma al)
        CBEQA  #$80,LKG6e ;LKG6e ye dallan (bir sonraki duruma al)
        BRA    LKG1b      ;LKG1b ye dallan (Aynı konumda)

; Bir Sonraki Duruma al
LKG6e    LDA    #$82       ;A=82 (Bir Sonraki Durum)
        STA    $57,X      ;(57+X)=A (Bir Sonraki Durum)
        BRA    LKG1b      ;LKG1b ye dallan

; Bir Önceki Duruma al
LKG6d    CLR    $57,X      ;(57+X)=00 (Bir Önceki Konum)
        JMP    LKG1b      ;LKG1b ye dallan

```

* 01. Durumda ise

```

LKG5    BSR    LKG8c      ;LKG8c alt programına dallan
        AND    #11000000Q ;A=A AND C0
        CBEQA  #$C0,LKG5d ;LKG5d ye dallan (Bir önceki duruma al)
        CBEQA  #$40,LKG5e ;LKG5e ye dallan (bir sonraki duruma al)
        ;      BRA    LKG8a ;LKG8a ya dallan
        ;      BRCLR 2,ISLTMP,SKPLI3 ;0 (ve) ise SKPLI (Kapalı) ya dallan
        JMP    LKG1b      ;LKG1b ye dallan (Aynı konumda)

; Bir Sonraki Duruma al
LKG5e    LDA    #$02       ;A=82 (Bir Sonraki Durum)
        STA    $57,X      ;(57+X)=A (Bir Sonraki Durum)
        ;      BRA    LKG8a ;LKG8a ya dallan
        ;      BRCLR 2,ISLTMP,SKPLI3 ;0 (ve) ise SKPLI (Kapalı) ya dallan
        JMP    LKG1b      ;LKG1b ye dallan

; Bir Önceki Duruma al
LKG5d    CLR    $57,X      ;(57+X)=00 (Bir Önceki Konum)
        ;      BRA    LKG8a ;LKG8a ya dallan
        ;      BRCLR 2,ISLTMP,SKPLI3 ;0 (ve) ise SKPLI (Kapalı) ya dallan
        JMP    LKG1b      ;LKG1b ye dallan

```

```

LKG8c    LDA    PTBD      ;A=PTBD
LKG8cc    DECX            ;X=X-1
        BEQ    LKG8bb     ;X=0 ise LK8bb ye dallan
        ROLA           ;A yı sola döndür
        ROLA           ;A yı sola döndür

```

```

LKG8bb  BRA    LKG8cc      ;LKG8c ye dallan
        LDX    AKMTMP2    ;X=RLCKTPI
        RTS

```

* 00 Durumunda ise

```

LKG4     ;LDA    $59,X      ;A=(59+X) Kapı Girişi Zamanlaması
        SUB     SANIYE     ;A=A-SANIYE
        CBEQA   #$FE,LKG4b ;Üç saniye geçmiş ise LKG4b ye dallan (6-8)
        CBEQA   #$58,LKG4b ;Üç saniye geçmiş ise LKG4b ye dallan (59-01)
        CBEQA   #$F8,LKG4b ;Üç saniye geçmiş ise LKG4b ye dallan (19-21)
        JMP     LKG1b      ;LKG1b ye dallan
LKG4b    LDA     $F0        ;A=80h
        STA     $5F,X      ;(5F+X)=80h yeni giriş yapılabilir
        JMP     LKG1b      ;LKG1b ye dallan

```

* 00. Durumda ise

```

LKG4c    BSR     LKG8c      ;LKG8c alt programına dallan
        AND     #11000000Q  ;A=A AND C0
        CBEQA   #$80,LKG4e ;LKG5e ye dallan (Bir sonraki duruma al) Giriş
        CBEQA   #$40,LKG4d ;LKG5d ye dallan (bir sonraki duruma al) Çıkış
        JMP     LKG1b      ;LKG1b ye dallan (Aynı konumda)
; Bir Sonraki Duruma al GİRİŞ
LKG4e    LDA     $01        ;A=01 (Bir Sonraki Durum)
        STA     $57,X      ;(57+X)=A (Bir Sonraki Durum)
        JMP     LKG1b      ;LKG1b ye dallan
; Bir Önceki Duruma al ÇIKIŞ
LKG4d    LDA     $81        ;A=82 (Bir Sonraki Durum)
        STA     $57,X      ;(57+X)=A (Bir Sonraki Durum)
        JMP     LKG1b      ;LKG1b ye dallan

```

* TUŞA BASILDIĞINDA BUZZER BLG MGD VE ANM ZAMANLAMASINI AYARLAR

```

LAB4ii   CLC          ;C=0
        MOV     #$FF,DDRA ;PTA çıkış
; Animasyon Zamanlaması
        LDA     ANMDON    ;A=ANMDON
        BEQ     LANM5     ;0 ise LANM5 e dallan
        DBNZ    ANMZAM,LANM5 ;ANMZAM-1=0 değilse LANM5 e dallan
        ROLA    A         ;A y1 sola döndür
        BCC     LCDSL     ;C=0 ise LCDSL ya dallan
LCDSR    ;SEI          ;Kesmeler pasif
        MOV     $1C,PTA   ;LCD yi sağa kaydır
        BRA     LANM7     ;LANM7 ye dallan
LCDSL    ;SEI          ;Kesmeler pasif
        MOV     $18,PTA   ;LCD yi sola kaydır
;        BRA     LANM7     ;LANM7 ye dallan
LANM7    JSR     LCDA     ;LCD yi bekle
        JSR     WMLCD     ;LCD ye yaz
        LDA     ANMDON    ;A=ANMDON
        DECA    A         ;A=A-1
        STA     ANMDON    ;ANMDON=A
        LSLA    A         ;A y1 sola döndür
        CMP     #AMDND    ;AMDND ile karşılaştır
        BHI     LANM8b    ;ANM ilk bölümü bitmemiş ise dallan
        BCLR    2,KIF     ;LCDDISP devrede
LANM8c   TSTA     A        ;A y1 testet
        BNE     LANM8b    ;ANMDON=0 değil ise LANM8 e dallan
        CLR     ANMDON    ;ANMDON=0
; LCD yi ve LCDRAM 1 Cursor At HOME a Ayarla

```

```

MOV    #LCDS1,LCDRAM ;LCDRAM '1 başa al
MOV    #$02,PTA       ;PTA=02
JSR    LCDA           ;LCD yi bekle
JSR    WMLCD          ;LCD ye yaz

```

```

LANM8b  MOV    #AMZMD2,ANMZAM ;ANMZAM=AMZMD2 (22h)

```

```

LANM5   LDA    KKTUSB
        CMP    #$01
        BNE    LC0
        MOV    #$22,DSPDON    ;DSPDON=22
LC0      LDA    KKTUSB
        BNE    LC1

```

```

; Otomatik EDITE BASMA ZAMANLAMASINI AYARLAR (Atla Menüünde İse)

```

```

LDA    KCIMNU        ;A=KCIMNU
CMP    #ATMBSY       ;Atla Menüünde mi
BNE    LAB4j         ;Değil ise LAB4j ye dallas
BRSET  1,KIF,LAB4j   ;Edite basılmışı ise LAB4j e dallas
LDA    ANMDON        ;ANMDON=0 mı
BNE    LAB4j         ;Değilse LAB4j ye dallas
LDA    DSPDON        ;DSPDON=0 mı
BNE    LAB4j         ;Değilse LAB4j ye dallas
MOV    #$01,KKTUSB   ;KKTUSB=1
MOV    #EDIT,KEYPR   ;KEYPR=EDIT
BRA    LAB4j         ;LAB4j ye dallas

```

```

; BUZ BLG ve MGD zamanlaması

```

```

LC1      MOV    #$60,BUZZAM ;BUZZAM=A0
        CLR    BLGZAM      ;BLGZAM=00
        MOV    #$08,MGDZAM ;MGDZAM=03
        MOV    #OEDZAMD,OEDZAM ;OEDZAM=OEDZAMD
        MOV    LTCH1D,PTA   ;PTA=LATCH1D
        LDA    PTC          ;A=PTC
        ORA    #LTCH1E      ;Buzzer açık
        STA    PTC          ;PTC=A
        MOV    LTCH2D,PTA   ;PTA=LATCH2D
        ORA    #LTCH2E      ;Back light (BLG) açık
        STA    PTC          ;PTC=A
;        CLI          ;Ints aktif
LAB4j    DBNZ   BUZZAM,LAB4d
;        SEI          ;Ints pasif
        LDA    PTC          ;A=PTC
        AND    #LTCH1DE     ;Buzzer kapalı
        STA    PTC          ;PTC=A

```

```

; Otomatik Edite Basma Zamanlaması (Alt Menüde İse)

```

```

LDA    KCIAMU        ;A=KCIAMU
BEQ    LAB14k        ;0 ise LAB14k ye dallas
LDA    KCIEMU        ;A=KCEAMU
BNE    LAB14k        ;0 değil ise LAB14k ye dallas
LDA    ANMDON        ;ANMDON=0 mı
BNE    LAB14k        ;Değilse LAB14k ye dallas
DBNZ   OEDZAM,LAB14k ;OEDZAM-1=0 değil ise LAB14k ye dallas
MOV    #$01,KKTUSB   ;KKTUSB=1
MOV    #EDIT,KEYPR   ;KEYPR=EDIT
CLR    BLGZAM        ;BLGZAM=00
MOV    #OEDZAMD,OEDZAM ;OEDZAM=OEDZAMD
LAB14k  DBNZ   BLGZAM,LAB4d
        LDA    PTC          ;A=PTC
        AND    #LTCH2DE     ;BLG kapalı

```

```

STA    PTC                ;PTC=A
DBNZ   MGDZAM,LAB4d
BRSET  0,BAYRAK0,LAB4d ;Hatalı giriş varsa geri dönme
CLRA                   ;A=0
STA    KCIEMU            ;KCIEMU=0
STA    KCIAMU            ;KCIAMU=0
STA    AMOFFSET          ;AMOFFSET=0
STA    EMOFFSET          ;EMOFFSET=0
INCA                   ;A=1
STA    KCIMNU
STA    KCIAAMU            ;GEREK OLMAYABİLİR
BCLR   2,KIF              ;KIF[2]=0 (LCDRAM 1 tara)
BCLR   4,KIF              ;KIF[4]=0 (LCDRAM 1 tara)

```

; 2. Satır Siler (Son Çıkış Menüsünden Büyük Menülerde ise)

```

LDX    #$10              ;LDX=16D
LDA    #$20              ;A=20 (Boşluk)
LD3db  STA    LCDS2D1,X   ;LCDS2-1+X e boşluk yaz
DBNZX  LD3db             ;X-1=0 değilse LD3db ye dallan
JMP    LB1               ;LB1 e dallan

```

* LCD ye LCD RAMdaki KARAKTERLERİ YAZAR

```

LAB4d  BRSET  2,KIF,LAB4dd ;LCDDSP devre dışı ise dallan (LCDDSP)
TST    DSPDON            ;DSP DONU testet
BEQ    LAB4dd            ;DSPDON=0 ise L11b ye dallan
DEC    DSPDON            ;DSPDON=DSPDON-1

```

; Display programı başlangıcı

```

LDSPB  LDA    LCDRAM
CMP    #LCDS2SP1         ;İkinci satır bitti mi
BNE    L11               ;Bitmedi ise L11 'e dallan
MOV    #LCDS1,LCDRAM     ;LCDRAM 'ı başa al
MOV    #$80,PTA          ;Cursorü üst satırın başına al
JSR    LCDA              ;LCD yi bekle
JSR    WMLCD             ;LCD ye yaz
L11b   BRA    LAB4dd      ;LAB4dd ye dallan

L11     CMP    #LCDS1SP1   ;Birinci sıra bitti mi
BNE    L12               ;Bitmedi ise L12 'e dallan
INC    LCDRAM            ;LCD RAM ının son kaldığı yer+1
MOV    #$C0,PTA          ;Cursorü alt satırın başına al
JSR    LCDA              ;LCD yi bekle
JSR    WMLCD             ;LCD ye yaz
BRA    LAB4dd            ;L04b ye dallan

```

```

L12     CLRH                ;H=0
LDX    LCDRAM
MOV    X+,PTA
JSR    LCDA              ;LCD yi bekle
JSR    WDLCD             ;LCD ye yaz
STX    LCDRAM            ;LCD RAM ının son kaldığı yer+1

```

* LCDRAMR'ı Tarayarak Menü yazılarını oluşturur

```

LAB4dd BRSET  4,KIF,LAB4g ;KIF[4]=1 ise LAB4g ye dallan
BRCLR  2,KIF,LAB4e       ;LCDDSP devrede ise LAB4e ye dallan
BRA    LD1b              ;LD1b ye dallan

```

; SAAT Menüsünde ve Saniye veya Dakika Geçmiş ise veya

; Bir Tuşa Basılmış ise LCDRAMı Taratır

```

LAB4g  LDA    KCIMNU       ;A=KCIMNU
CMP    #SMNBSY            ;SAAT Menüsünde mi
BNE    LD1b              ;Değilse LD1b ye dallan
BRSET  4,BAYRAK0,LD1c     ;BAYRAK0[4]=1 ise LD1c ye dallan
LD1b   JMP    LAB4        ;LAB4 e dallan

```

; SAAT Menüsünde ve Saniye veya Dakika Geçmiş ise

```

LD1c    BCLR  4,BAYRAK0    ;BAYRAK0[4]=0
; Bir Tuşa Basılmış ise
LAB4e    BSET  4,KIF        ;LCD RAM tarama devre dışı
        JSR   MUDSPADR      ;Menü Display Alt prog. dalan
LD1      ROLA              ;A yı C üzerinden sola döndür
        BHS   LD0          ;C=0 ise LD0 a dallan
        JSR   WCMLCDR      ;WCMLCDR alt prog dallan
        BRA   LD2          ;LD2 ye dallan
LD0      JSR   WSMLCDR      ;W2MLCDR alt prog dallan
LD2      DBNZ  DONGU,LD1    ;DOGU-1=0 değil ise LD1 e dallan
        BRSET 1,BAYRAK0,LD3 ;BAYRAK0[1]=1 ise LD3 e dallan
        LDA   SABLON2      ;A=SABLON2
        MOV   #$08,DONGU   ;DONGU=8H
        BSET  1,BAYRAK0    ;BAYRAK0[1]=1
        BRA   LD1          ;LD1 e dallan
LD3      BCLR  1,BAYRAK0    ;BAYRAK0[1]=0
        MOV   #LCDS1,LCDCA2 ;LCDCA2=LCD RAMının 1. satır adr.
        TST   KCIAMU       ;Kaçınıcı alt menüyü test et
        BNE   LD3c         ;0 değil ise LD3c ye dallan
        LDA   KCIMNU       ;A=KCIMNU
        CMP   #SFMSY       ;A yi #SFMSY (13H) ile karşılaştır
        BPL   LD6          ;Pozitif ise LD6 ya dallan
        CBEQA #SMNBSY,LD3b ;SAAT menüsünde ise (1) LD3b ye dallan
        CMP   #ATMBSY      ;Atla menüsü mü (2)
        BNE   LD3e         ;değilse LD3e ye dallan
        LDA   #$3          ;A=3
; Kaçınıcı Çıkış Olduğunu Yazar (Çıkış Menüsünde ise)
LD3e      SUB   #CMBSAY     ;A=A-CMBSAY(3)
        DAA
        INCA
        DAA
        LDX   #$91         ;LCD Ramın adresi
        JSR   WLCDRAM      ;LCD Ram a yaz
        BRA   LD3c         ;LD3C ye dallan
; Saat ve Günü Yazar
LD3b      LDX   #$A2        ;LCDRAM adr.=A2H
        LDA   SAAT         ;A=SAAT
        CMP   #$0A         ;0F ile karşılaştır
        BHI   LD3b1        ;Yüksek ise LD3b! e dallan
        ADD   #$F0         ;A=A+F0h
LD3b1     JSR   WLCDRAM      ;WLCDRAM alr prog. dallan
        MOV   #$3A,$A3     ;LCDRAM A3H a : yaz
        LDX   #$A5         ;LCDRAM adr.=A5H
        LDA   DAKIKA       ;A=DAKIKA
        JSR   WLCDRAM      ;WLCDRAM alr prog. dallan
        LDX   #$A8         ;LCDRAM adr.=A8H
        LDA   SANIYE       ;A=SANIYE
        JSR   WLCDRAM      ;WLCDRAM alr prog. dallan
        LDX   #$AC         ;LCDRAM adr.=ACH
        LDA   GUN          ;A=GUN
        ADD   #$F0         ;A=A+F0h
        JSR   WLCDRAM      ;WLCDRAM alr prog. dallan
;        BSET  4,KIF        ;LCD RAM tarama devre dışı
        BRA   LLAB4        ;LAB4 e dallan
; Şifre Menüünden Büyük (ise) mü diye kontrol eder (Alt Menüde ise)
LD3c      LDA   KCIMNU      ;A=KCIMNU
        CBEQA #GRMBSY,LD3i ;Giriş Menüünde ise LD3i ye dallan
        CMP   #SFMSY       ;SFMSY ($13) ile karşılaştır
        BMI   LD3h         ;Negatif ise LD3h ye dallan
        BRA   LD6          ;LD6 ya dallan
; Kaçınıcı Programda Olduğunu Yazar (Şifre Mnüsünden Küçük Menülerde)
LD3h      LDA   KCIPRG      ;A=KCIPRG
        LDX   #LCDS1S      ;X=LCDS1S

```

```

        JSR    WLCDRAM        ;WLCDRAM alt programına dallan
        MOV    #$50,$9E      ;LCDRAMa P yaz
LD3i    LDA    KCIEMU        ;KCIEMU yu test et
        BNE    LEM1          ;EDIT menüsünde ise

```

; Alt menülerin kaçınıcı olduğunu yazar

```

        LDA    KCIAMU        ;A=KCIAMU
        CMP    #$03          ;03 mü
        BMI    LD6           ;Eksi ise LD6 ya dallan
        CMP    #$07          ;07 ile karşılaştır
        BPL    LD6           ;Eksi ise LD6 ya dallan
        MOV    #$90,KAAMOFST ;KAAMOFST=90
        CMP    #$03          ;03 ile karşılaştır
        BEQ    LD8           ;Eşit ise LD8 e dellan
        INC    KAAMOFST      ;KAAMOFST=KAAMOFST+1
LD8     JSR    MUDSPADR      ;MUDSPADR alt programına dallan
        LDX    #KAAMOFST     ;X=#KAAMOFST
        JSR    CSAYISI       ;CSAYISI alt programına dallan
        LDX    KAAMOFST     ;X=#KAAMOFST
LD7     LDA    KCIAAMU       ;A=KCIAAMU
        CMP    #$01          ;01 mi
        BLE    LD6           ;1 veya 0 ise LD6 ya dallan
        ADD    #$30          ;A=A+30H
        STA    ,X            ;(X)=A
; 2. Satırı Siler (Son Çıkış Menüsünden Büyük Menülerde ise)
LD6     LDX    #$10          ;LDX=16D
        LDA    #$20          ;A=20 (Boşluk)
LD3d    STA    LCDS2D1,X     ;LCDS2-1+X e boşluk yaz
        DBNZX LD3d          ;X-1=0 değılsre LD3d ye dallan
;        BSET    4,KIF       ;LCD RAM tarama devre dışı
LLAB4   MOV    #$24,DSPDON   ;DSPDON=4Ah
        JMP    LAB4          ;LAB4 e dallan

```

; Edit menüsünde ise (İlk satırın değişkenlerini yazar)

```

LEM1    ;BRA    LLAB4        ;LAB4 e dallan
;        LDA    KCIEMU        ;A=KCIEMU
        CMP    #SIEMBSY     ;Sıcaklık Edir M. başlangıç sayısı
        BNE    LD9          ;Eşit değil ise LD9 a dallan
; Sıcaklık Edit Alt Menüsünde İse
        LDX    SICAK1       ;X=SICAK1
        MOV    SICAK2,XTEMP2 ;XTEMP2=SICAK2
        BRA    LD14         ;LD14 e dallan
LD9     CMP    #ISEMBSY     ;Işık Şiddeti Edir M. başlangıç say
        BNE    LD10         ;Eşit değil ise LD10 a dallan
; Işık Şiddeti Edit Alt Menüsünde İse
        LDX    ISIKS1       ;X=ISIKS1
        MOV    ISIKS2,XTEMP2 ;XTEMP2=ISIKS2
        BRA    LD14         ;LD14 e dallan
LD10    CMP    #KGEMBSY     ;Kapı Girişi Edir M. başlangıç say
        BNE    LD11         ;Eşit değil ise LD11 a dallan
; Kapı Girişi Edit Alt Menüsünde İse
        LDX    KAPIG1       ;X=KAPIG1
        MOV    KAPIG2,XTEMP2 ;XTEMP2=KAPIG2
        BRA    LD14         ;LD14 e dallan
LD11    CMP    #LGEMBSY     ;Lojik Girişi Edir M. başlangıç say
        BEQ    LD14b        ;Eşit ise LD14b ye dallan
        JMP    LD12         ;LD12 ye dallan
; Lojik Girişi Edit Alt Menüsünde İse
LD14b   LDX    LOJG1        ;X=LOJG1
        MOV    LOJG2,XTEMP2 ;XTEMP2=LOJG2
;        BRA    LD14         ;LD14 e dallan
;LD12    BRA    LD13        ;LD13 edalan

```

```

LD14    MOV    #$93,XTEMP1    ;XTEMP1=93
        CLRA                      ;A=0
        BSR    WEMCDS1        ;WEMCDS1 alt prog dallan
        PSHA                      ;A i yığına at
        LDA    KCIAAMU        ;A=KCIAAMU
        CMP    #$01          ;01 ile karşılaştır
        PULA                      ;A i yığından çek
        BEQ    LD15          ;0 değil ise LD15 e dallan
        MOV    #$93,XTEMP1    ;XTEMP1=93
        BSR    WEMCDS1b      ;WEMCDS1b alt prog dallan
        PSHA                      ;A i yığına at
        LDA    KCIAAMU        ;A=KCIAAMU
        CMP    #$02          ;02 ile karşılaştır
        PULA                      ;A i yığından çek
        BEQ    LD15          ;0 değil ise LD15 e dallan
        MOV    #$93,XTEMP1    ;XTEMP1=93
        BSR    WEMCDS1b      ;WEMCDS1b alt prog dallan
        PSHA                      ;A i yığına at
        LDA    KCIAAMU        ;A=KCIAAMU
        CMP    #$03          ;03 ile karşılaştır
        PULA                      ;A i yığından çek
        BEQ    LD15          ;0 değil ise LD15 e dallan
        MOV    #$93,XTEMP1    ;XTEMP1=93
        BSR    WEMCDS1b      ;WEMCDS1b alt prog dallan
; Edit Menüünde Giriş Kalmadı ise Boşluk Yazar
LD15    LDX    XTEMP1        ;X=XTEMP1
LD15b   CBEQX  #$9F,LD12     ;X=9F ise LD12 ye dallan
        LDA    #$20          ;A=20
        STA    ,X            ;(X)=A
        DECX                      ;X=X-1
        STA    ,X            ;(X)=A
        DECX                      ;X=X-1
        STA    ,X            ;(X)=A
        AIX    #$05          ;X=X+5
        BRA    LD15b        ;LD15b ye dallan

WEMCDS1 MOV    #$08,DONGU    ;DONGU=04
WEMCDS1b INCA                      ;A=A+1 Write Edir Mnu Char Dara Satırlı
        CLC                      ;C=0
        ROLX                      ;X i sola döndür
        BCC    LDB1b        ;C=0 ise LDB1b e dallan
        PSHX                      ;X i yığına at
        PSHA                      ;A i yığına at
        LDX    XTEMP1        ;X=XTEMP1(LCD RAMın Adresi)
        CLC                      ;C=0
        DECA                      ;A=A-1
        DAA                      ;A yı 10 luk düzene çevir
        INCA                      ;A=A+1
        DAA                      ;A yı 10 luk düzene çevir
        JSR    WLCDRAM        ;WLCDRAM alt prog dallan
        AIX    #$04          ;X=X+4
        STX    XTEMP1        ;XTEMP1=X
        CPX    #$9F          ;STX yapıldığında X=9F miydi
        PULA                      ;A i yığından çek
        PULX                      ;X i yığından çek
        BEQ    LDB1          ;9F ise LDB1 e dallan
;      RTS                      ;Alt programdan geri dön
LDB1b   DBNZ   DONGU,WEMCDS1b ;DONGU-1=0 değil ise WEMCDS1b ye dallan
        TST    XTEMP2        ;XTEMP2 yi testet
        BEQ    LDB1          ;0 ise LDB1 e dallan
        LDX    XTEMP2        ;X=XTEMP2(Sablon2)
        CLR    XTEMP2        ;XTEMP2=0

```

```

LDB1      BRA    WEMCDS1      ;WEMCDS1 ye dallan
          RTS                ;Alt programdan geri dön

```

```

WCMLCDR   LDHX   WLCDCAH      ; (H:X) = (WLCDCAH:WLCDCAH+1)
          MOV    X+, WLCDCAT   ; WLCDCAT = (H:X)
          STHX   WLCDCAH      ; (WLCDCAH:WLCDCAH+1) = (H:X)
          CLRH                   ; H=0
          LDX    LCDCA2       ; X=LCDCA2
          MOV    WLCDCAT, X+   ; LCDCA2=WLCDCAT
          STX    LCDCA2       ; LCDCA2+1
LNOF      RTS                ;Alt programdan geri dön

```

```

WSMLCDR   LDX    LCDCA2       ; X=LCDCA2
          MOV    #$20, AKMTMP1 ; AKMTMP1=20H (Boşluk)
          MOV    AKMTMP1, X+   ; LCD ye boşluk yaz
          STX    LCDCA2       ; LCDCA2+1
          RTS                ;Alt programdan geri dön

```

; Edit Menüünün Alt Satırını Yazar

```

LD12      LDA    KCIEMU       ; A=KCIEMU
          CMP    #CGEMBSY     ; Çıkış Gücü Edit Menüünde mi
          BNE    LEMS1       ; Değil ise LEMS1 e dallan
; Çıkış Gücü Edit Menüünde ise
          LDA    ODCGRA       ; A=(Output Data Çıkış Gücü RAM Adr)
          BEQ    LEMC0       ; 0 ise LEMC0 a dallan
          LSLA                   ; A yı lojik sola kaydır (iki ile çarp)
          INCA                   ; A=A+1
          JSR    BINBCD       ; BINBCD alt prog dalan
LEMC0     LDX    #$A4         ; X=A4h
          JSR    WLCDRAM      ; WLCDRAM alt programına dallan
          BRA    LABB4       ; LAB4 e dallan

```

```

LEMS1     CMP    #SAENBSY     ; SAAT edit menüsünde mi
          BNE    LEMS1b      ; Değilse LEMS1b ye dallan

```

; SAAT Edit Menüünde ise

```

          LDHX   #ODRSAD      ; H:X=#ODRSAD (Output data RAM Saat Adr)
          STHX   HXTMPH      ; HXTMPH:HXTMPL=H:X
;          MOV    #ODRSAD, XTEMP1 ; XTEMP1=Output data RAM Saat Adr
          CLR    XTEMP1       ; XTEMP1=0
          MOV    #LCDS2P1, XTEMP2 ; XTEMP2=LCDS2P1 (A2h)
          MOV    #$02, AKMTMP1 ; AKMTMP1=02
LEMS4     MOV    #$02, DONGU   ; DONGU=02
LEMS3     LDHX   HXTMPH       ; X=HXTMPH
          INC    HXTMPL       ; HXTMPL=HXTMPL+1
          LDA    ,X           ; A= (X)
          LDHX   XTEMP1       ; X=XTEMP1
          JSR    WLCDRAM      ; WLCDRAM alt programına dallan
          AIX    #$4          ; X=X+4
          STX    XTEMP2       ; XTEMP2=X
          DBNZ   DONGU, LEMS3  ; DONGU-1=0 değil ise LEMS3 e dallan
          INC    XTEMP2       ; XTEMP2=XTEMP2+1
          DBNZ   AKMTMP1, LEMS4 ; AKMTMP1-1=0 değil ise LEMS4 e dallan
          LDA    #$3A         ; A=3Ah
          STA    $A3          ; (3A)=A(:)
          STA    $AA          ; (AA)=A(:)
          BRA    LABB4       ; LAB4 e dallan

```

```

LEMS1b    CMP    #ISEMBSY     ; Işık Şiddeti veya Sıcaklık E Menüünde mi
          BHI    LEMS2       ; Değil ise LEMS2 ye dallan

```

; SICAKLIK veya IŞIK ŞİDDETİ Edit Menüünde ise

```

MOV    #$04,DONGU    ;DONGU=04h
MOV    #92,XTEMP1    ;XTEMP1=92h
LEMS4b LDX    XTEMP1    ;X=XTEMP1
JSR    LCDCHAR        ;LCDCHAR alt prog dallan
CMP    #$F0           ;A=EF mi (Boşluk mu)
BEQ    LABB4          ;Evet ise LABB4 e dallan
CMP    #$10           ;16 dan küçük mü
BLO    LEMS3b         ;Evet ise LEMS3b ye dallan
SUB    #$6            ;6 çıkar BCD yi binary sayıya çevir
LEMS3b TAX           ;X=A
LDA    KCIMNU         ;A=KCIMNU
CBEQA  #GRMBSY,LEMS3c ;A=GRMBSY ise LEMS3c
LDA    $200,X         ;A=(200+X)
BRA    LEMS3d         ;LEMS3d ye dallan
; Girişler Menüsünde ise Girişleri Yazar
LEMS3c LDA    $3F,X    ;A=(3F+X)
CMP    #100t         ;99 ile karşılaştır
BLO    LEMS3d         ;Düşük ise LEMS3d ye dallan
LDA    #99t          ;A=99t
LEMS3d JSR    BINBCD   ;BINBCD alt prog dallan
LDX    XTEMP1        ;X=XTEMP1
AIX    #18t          ;X=X+18d
JSR    WLCDRAM        ;WLCDRAM alt programına dallan
AIX    #-14t         ;X=X-14d
STX    XTEMP1        ;XTEMP1=X
DBNZ   DONGU,LEMS4b   ;DONGU-1=0 değil ise LEMS4b ye Dallan
LABB4  MOV    #$24,DSPDON ;DSPDON=4Ah
JMP    LAB4          ;LAB4 e dallan
LEMS2  CMP    #LGEMBSY  ;Lojik veya kapı girişi edit menüsünde mi
BHI    LEMS5         ;Değil ise LEMS5 e dallan
; Lojik veya kapı girişi edit menüsünde ise
MOV    #$04,DONGU    ;DONGU=04h
MOV    #92,XTEMP1    ;XTEMP1=92h
LEMS8  LDX    XTEMP1    ;X=XTEMP1
JSR    LCDCHAR        ;LCDCHAR alt prog dallan
CMP    #$F0           ;A=EF mi (Boşluk mu)
BEQ    LABB4          ;Evet ise LABB4 e dallan
CMP    #$09           ;9 dan küçük mü
BLO    LEMS6          ;Evet ise LEMS6 ya dallan
BEQ    LEMS6b         ;9 ise LEMS6b ye dallan
SUB    #$6            ;A=A-6 (BCD yi Binary yap)
LEMS6b SUB    #$08     ;8 çıkar BCD yi binary sayıya çevir
STA    AKMTMP1        ;AKMTMP1=A
LDA    ODLGRA1        ;Outpet Data Loj Gir Adr1
BRA    LEMS7          ;LEMS7 ye dallan
LEMS6  STA    AKMTMP1  ;AKMTMP1=A
LDA    ODLGRA2        ;Outpet Data Loj Gir Adr2
LEMS7  ASLA         ;A yı aritmetik Sola Kaydır
DBNZ   AKMTMP1,LEMS7  ;AKMTMP1-1=0 değil ise LEMS7 ye dallan
LDA    #$18           ;A=18h
ROLA           ;A yı sola kardır
AIX    #16t          ;X=X+17d
STA    ,X            ;(X)=A
AIX    #-14t         ;X=X-14d
STX    XTEMP1        ;XTEMP1=X
DBNZ   DONGU,LEMS8    ;DONGU-1=0 değil ise LEMS8 e Dallan
BRA    LABB4          ;LAB4 e dallan
LEMS5  CMP    #LIEMBSY  ;Lojik işlem edit menüsünde mi
BNE    LEMS9         ;Değilise LEMS9 a dallan
; Lojik İşlem Edit Menüsünde ise
LDA    ODLIRA         ;A=(Output data Loj İş RAM Adr)
MOV    #$03,DONGU     ;DONGU=03h
MOV    #$31,AKMTMP1   ;AKMTMP1=31h

```

```

MOV    #$30, AKMTMP2    ;AKMTMP2=30h
ASLA                    ;A y1 aritmetik Sola Kaydır
ASLA                    ;A y1 aritmetik Sola Kaydır
LDX    #$A3              ;X=A3h
LEMS11b ASLA              ;A y1 aritmetik Sola Kaydır
ASLA                    ;A y1 aritmetik Sola Kaydır
BCC    LEMS10b           ;C=0 ise L10EMSb ye dallan
MOV    AKMTMP1, X+       ; (X)=AKMTMP1(31h)
AIX    #03t              ;X=X+3
BRA    LEMS12b           ;LEMS12b ye dallan
LEMS10b MOV    AKMTMP2, X+ ; (X)=AKMTMP2(30h)
AIX    #03t              ;X=X+3
LEMS12b DBNZ    DONGU, LEMS11b ;DONGU-1=0 değil ise LEMS11b ye dallan
LABBB4 BRA    LABB4      ;LABB4 e dallan
; Hangi Günler Edit Menüsünde İse
LEMS9   LDA    ODHGRA     ;A=(Output data Hng Gun RAM Adr)
MOV     #$07, DONGU       ;DONGU=03h
MOV     #$31, AKMTMP1     ;AKMTMP1=31h
MOV     #$30, AKMTMP2     ;AKMTMP2=30h
ASLA                    ;A y1 aritmetik sola kaydır
LDX     #LCDS2            ;X=LCDS2
LEMS11  ASLA              ;A y1 aritmetik sola kaydır
BCC     LEMS10           ;C=0 ise L10EMS ye dallan
MOV     AKMTMP1, X+       ; (X)=AKMTMP1(31h)
INX                     ;X=X+1
BRA     LEMS12           ;LEMS12 ye dallan
LEMS10  MOV     AKMTMP2, X+ ; (X)=AKMTMP2(30h)
INX                     ;X=X+1
LEMS12  DBNZ    DONGU, LEMS11 ;DONGU-1=0 değil ise LEMS11 ye dallan
BRA     LABBB4           ;LABB4 e dallan
;      JMP     LD6        ;LD36 ya dallan

```

* Şablon ve karakter adreslerinin değerlerini ayarlar

```

MUDSPADR LDA    KCIEMU     ;A=KCIEMU
DECA                    ;A=A-1
BMI     LTSTAMU           ;Eksi ise ise LTSTAMU ya dallan
LMC1    LDA     #EMLCDAL   ;A=#EMLCDAL
ADD     EMOFFSET          ;A=A+EMOFFSET
STA     WLCDCAL           ;WLCDCAL=A
MOV     #EMLCDAH, WLCDAH ;WLCDAH=#EMLCDAH
LDA     KCIEMU            ;A=KCIEMU
LMC1b   CLC              ;C=0
ROLA                    ;A=A*2
ADD     #EMS2ADL          ;Edit Menü Şablon2 adr. düşük
MOV     #EMS2ADH, SM2ADH ;SM2ADH=#EMS2ADH
BRA     LMNU              ;LMNU ya dallan
LTSTAMU LDA     KCIAMU     ;A=KCIAMU
DECA                    ;A=A-1
BMI     LMNU1             ;KCIAMU=0 ise LMNU1 e dallan
LMC2    LDA     #AMLCDAL   ;A=#AMLCDAL
ADD     AMOFFSET          ;A=A+AMOFFSET
STA     WLCDCAL           ;WLCDCAL=A
MOV     #AMLCDAH, WLCDAH ;WLCDAH=#AMLCDAH
LDA     KCIAMU            ;A=KCIAMU
ROLA                    ;A=A*2
ADD     #AMS2ADL          ;Alt Menü Şablon2 adr. düşük
MOV     #AMS2ADH, SM2ADH ;SM2ADH=#AMS2ADH
BRA     LMNU              ;LMNU ya dallan
LMNU1   LDA     KCIMNU     ;A=KCIMNU
DECA                    ;A=A-1
BNE     LMC3              ;KCIMNU=1 değil ise LMC3 e dallan
CLR     MCOFFSET          ;Menu Karakter Offset=0
LMC3    LDA     #M1LCDAL   ;A=#M1LCDAL

```

```

ADD    MCOFFSET      ;A=A+MCOFFSET
STA    WLCDCAL        ;WLCDCAL=A
MOV    #M1LCDAH,WLCDAH ;WLCDAH=#M1LCDAH
LDA    KCIMNU         ;A=KCIMNU
CMP    #$12           ;12H mi
BLS    LMC4b          ;Düşük veya aynı ise LMC4b ye dallan
SUB    #$10           ;A=A-10
BRA    LMC4           ;LMC4 e dallan
LMC4b  CMP    #ATMBSY  ;A yı ATMBSY (2) ile karşılaştır
BLS    LMC4           ;ATMBSY den düşük veya eşit ise LMC4 e dallan
LDA    #ATMBSY        ;A=ATMBSY (2)
LMC4   CLC           ;C=0
ROLA   ;A yı C üzerinden sola döndür (2 ile çarp)
ADD    #MS2ADL        ;Menü Şablon2 adr. düşük
MOV    #MS2ADH,SM2ADH ;SM2ADH=#EMS2ADH
LMNU   STA    SM2ADL    ;Şablon2 menü adres düşük
LDHX   SM2ADH         ;(H:X)=(SM2ADH:SM2ADH+1)
MOV    #$08,DONGU     ;DONGU=8H
LDA    ,X              ;A=SABLON1
INX    ;X=X+1
MOV    X+,SABLON2     ;SABLON2=(SM2ADH:SM2ADH+1)
CLR    H              ;H=0
RTS    ;ALT programdan geri dön

```

* Basılan Tuşa Göre İşlem Yapar

```

LLAB4b JMP    LAB4b      ;Saat-Atla Mnsünde ise (Eksi) ise LAB4b ye
dallan
LLAB4c JMP    LAB4C      ;0 değilse LAB4c ye dallan
LLEMU1 JMP    LEMU1      ;EDIT menüsünde ise LEMU1 e dalan
LAB9bb JMP    LAB9b      ;Edit değilse LAB9b ye dallan
LEE1   JMP    LAB1       ;LAB1 (WAIT) e dallan
LAB4   TST    KKTUSB     ;KKTUSB yi test et
BEQ    LEE1             ;0 ise WAIT e dallan
DEC    KKTUSB          ;KKTUSB=KKTUSB-1
BNE    LAB4m           ;0 değilse LAB4m ye dallan
BCLR   4,KIF           ;LCD RAM tarama devrede
LAB4m  LDA    KEYPR      ;KEYPR i A ya yükle
CMP    #EDIT           ;Edit tuşu mu
BNE    LAB9bb          ;Edit değilse LAB9b ye dallan
BRSET  1,KIF,LAB9bb    ;Edite basılmışdı ise LAB9b ye dallan

```

* İlk kez Edit/Enter tuşuna basıldı

```

LIEETB TST    KCIEMU     ;Kaçınıcı EDIT Menüsünü test et
BNE    LLEMU1          ;EDIT menüsünde ise LEMU1 e dalan
LAB4f  TST    KCIAMU     ;Kaçınıcı alt Menüü test et label???
BNE    LLAB4c          ;0 değilse LAB4c ye dallan

```

* İlk kez Edit/Enter tuşuna basıldığında MENÜ de ise

```

INC    KCIAMU          ;KCIAMU=1
LDA    KCIMNU          ;A=KCIMNU
LAB4h  CMP    #CMBSAY    ;CMBSAY ile karşılaştır (Saat vy Atla Mnsünde
mi)
BMI    LLAB4b          ;Saat-Atla Mnsünde ise (Eksi) ise LAB4b ye
dallan

```

* SAAT veya ATLA MENÜSÜNDE DEĞİL İSE

```

MOV    #AMDND,ANMDON ;ANMDON=AMDND(20h)
CBEQA  #GRMBSY,LAB4ha ;Girişler menüsünde ise LAB4ha ya dallan
CMP    #GAYMBSY       ;Giriş Ayarları menüsünde mi
BNE    LAB41          ;Değil ise LAB41 ye dallan
; Giriş Ayarları veya Giriş Menüsünde ise
LAB4ha MOV    #ASOKT,KEYPR ;KEYPR=#ASOKT
MOV    #GAAMBSY,KKTUSB ;KKTUSB=#GAAMBSY
JMP    LAB1           ;LAB1 (WAIT) e dalan
LAB41  CMP    #CAYMBSY   ;Çıkış Ayarları menüsünde mi

```

```

        BNE     LAB4n          ;Değil ise LAB4n ye dallan
; Çıkış Ayarları Menüünde ise
        MOV     #ASOKT,KEYPR   ;KEYPR=#ASOKT
        MOV     #CAAMBSY,KKTUSB ;KKTUSB=#CAAMBSY
LAB4n    CMP     #PAYMBSY       ;Program Ayarları menüsünde mi
        BNE     LAB4o          ;Değil ise LAB4o ya dallan
; Program Ayarları Menüünde ise
        MOV     #ASOKT,KEYPR   ;KEYPR=#ASOKT
        MOV     #PAAMBSY,KKTUSB ;KKTUSB=#PAAMBSY

; Yeni Alt Menüleri Buraya Ekle

LAB4o    CMP     #SFMBYSY       ;SFMBYSY ile karşılaştıır
        BNE     LAB4k          ;0 değil ise LAB4k ye dallan
; Şifre Menüünde ise
        MOV     #ASOKT,KEYPR   ;KEYPR=#ASOKT
        MOV     #SAMBSY,KKTUSB ;KKTUSB=#GAAMBSY
;Çıkış Menüünde ise (Flashdaki Çıkış Bilgilerini RAM Bufferına yazar)
;0100-017F=128 byte 0201-0219=25 byte 0181-190=16 byte (Çıkış Gücü)
; İşlem Gören Çıkış Data RAMına Boşluk Yaz
LAB4k    LDX     #$19           ;X=80
        CLRA                    ;A=0
LAB0cc   STA     $0200,X         ;(X+FF)=0
        DBNZX   LAB0cc          ;X-1=0 değilse LAB0cc ye dadallan
        BSR     FSHBADR         ;FSHBADR alt prog dallan
        LDHX    #FBRBAD         ;Flash Buffer Ram Baş Adr
        STHX    XTEMP1          ;XTEMP1:XTEMP2=H:X
;        STA     COPCTL          ;COP sayıcısını sil
        MOV     #$80,DONGU       ;DONGU=80h(128 byte)
LRB1     LDHX    HXTMPH          ;H:X=HXTMPH:HXTMPL
        LDA     ,X               ;A=,X
        INCX                    ;X=X+1
        STX     HXTMPL          ;HXTMPL=X
        LDHX    XTEMP1          ;H:X=XTEMP1:XTEMP2
        STA     ,X               ;,X=A
        INCX                    ;X=X+1
        STX     XTEMP2          ;XTEMP2=X
        DBNZ    DONGU,LRB1       ;DONGU-1=0 değil ise LRB1 e dallan
; Çıkış Verilerini RAMA(201-219) yazar
        STA     COPCTL          ;COP sayıcısını sil
        LDHX    #FBRBAD         ;Flash Buffer Ram Baş Adr
        BSR     FSHPBADR        ;FSHPBADlt prog dallan
        STHX    HXTMPH          ;HXTMPH:HXTMPL=Prog Data Bulun RAM Baş Adr
        MOV     #$03,DONGU       ;DONGU=03h
LRB4     LDHX    HXTMPH          ;H:X=HXTMPH:HXTMPL
        LDA     ,X               ;A=,X
        INCX                    ;X=X+1
        STX     HXTMPL          ;HXTMPL=X
        AIX     #$03             ;X=X+3
        MOV     X+,AKMTMP1       ;AKMTMP1=,X
        CLRH                    ;H=0
        TAX                    ;X=A
        LDA     AKMTMP1          ;A=AKMTMP1
        STA     ODRBAD,X         ;,X+ODRBAD(200h)=AKMTMP1(Çıkış değeri)
        DBNZ    DONGU,LRB4       ;DONGU-1=0 değil ise LRB4 e dallan
; Çıkış Gücünü Çıkış Verisine Yazar
        LDHX    HXTMPH          ;H:X=HXTMPH:HXTMPL
        LDA     ,X               ;A=(X)
        STA     ODCGRA           ;,Çıkış G O D RAM Adr=A
; Saat Lojik İşlem ve Hangi Günler verilerini Çılış Datası RAMına yazar
ATX     #404

```

```

        STHX  XTEMP1          ;XTEMP1:XTEMP2=H:X
        MOV   #$08,DONGU      ;DONGU=08h
LRB5    LDHX  HXTMPH          ;H:X=HXTMPH:HXTMPL
        LDA   ,X              ;A=,X
        INCX             ;X=X+1
        STX   HXTMPL          ;HXTMPL=X
        LDHX  XTEMP1          ;H:X=XTEMP1:XTEMP2
        STA   ,X              ;,X=A
        INCX             ;X=X+1
        STX   XTEMP2          ;XTEMP2=X
        DBNZ  DONGU,LRB5      ;DONGU-1=0 değil ise LRB5 e dallan
        JMP   LAB1            ;WAIT e dallan

```

; Flash bşlama adresini bulur

```

FSHBADR LDA  KCIMNU          ;A=KCIMNU
        SUB   #CMBSYD1       ;A=Kaçınıcı Çıkış
        ASLA             ;A yı aritmetik sola ötele (2 ile çarp)
        ADD   #FADRDL        ;A=A+Flash Adr Data Low
        STA   HXTMPL        ;Flash Adr Düşük=A
        MOV   #FADRDL,HXTMPH ;HXTMPH=#FADRDL(F9)
        LDHX  HXTMPH        ;H:X=FLADRH:FLADRL
        MOV   X+,HXTMPH      ;FLADRH=,X
        MOV   X+,HXTMPL      ;FLADRL=,X
        LDHX  HXTMPH        ;H:X=HXTMPH:HXTMPL
        RTS                ;Alt prog geri dön

```

; Flash datasındaki program başlama adresini bulur

```

FSHPBAD LDA  KCIPRG          ;A=KCIPRG
LRB3    DECA             ;A=A-1
        BEQ   LRB2          ;A=0 ise LRB2 ye dallan
        AIX   #$10          ;H:X=H:X+10h
        BRA   LRB3          ;LRB3 e dallan
LRB2    RTS
LAB9b   JMP   LAB5

```

* İlk kez Edit/Enter tuşuna basıldığında ALT menüde ise

```

LAB4c   ;LDA  KCIAMU          ;A=KCIAMU
        ;ADD  AEMNAL          ;A=A+alt-edit menu adr. düşük
        ;STA  AEMNAL          ;AEMNAL=A
        ;LDHX AEMNAL          ;H:X=(AEMNAL:AEMNAL+1)
        ;LDA  ,X              ;A=(AEMNAL:AEMNAL+1)
        ;STA  KCIEMU          ;KCIEMU=A
        MOV   KCIAMU,KCIEMU   ;KCIEMU=KCIAMU
        JMP   LAB1            ;WAIT e dallan

```

* İLKKEZ EDIT/ENTER TUŞUNA BASILDIĞINDA EDIT MENÜSÜNDE İSE

```

LEMU1   BSET  0,KIF          ;KIF[0]=1
        BRA   LAB4b2         ;LAB4b2 ye dallan

```

* İLKKEZ EDIT/ENTER TUŞUNA BASILDIĞINDA SAAT veya ATLA MENÜSÜNDE İSE

```

LAB4b   CLR  KCIAMU          ;KCIAMU=0
LAB4b2  BSR  FRMADR          ;Format adresi alt prog. dallan
LE0d    LDA  KCIMNU          ;A=KCIMNU
        CBEQA #ATMBSY,LE0f   ;A==#ATMBSY ise LE0f e dallan
        MOV   #LCDS2D1,LCDCA ;LCDCA=LCDRAM ının 2. sıra adresi-1
        MOV   #$BF,LCDDA     ;(LCD display adr)LCDDA=BFH
        BRA   LE0g           ;LE0g ye dallan
LE0f    MOV   #LCDS1D1,LCDCA ;LCDCA=LCDRAM ının 1. sıra adresi-1
        MOV   #$7F,LCDDA     ;(LCD display adr)LCDDA=7FH
LE0g    LDHX  FM1ADH          ;(H:X)=(FM1ADH:FM1ADL)
LE0e    MOV   #$9,DONGU       ;DONGU=9

```

```

      LDA      ,X                ;A=FORMATMENU1
LE0    DBNZ    DONGU,LE0b        ;DONGU-1=0 DEĞİLSE LE0b ye dallan
      BRCLR   3,KIF,LE0c        ;KIF[3]=0 ise LE0c ye dallan
LE1b   BCLR   3,KIF              ;KIF[3]=0
      MOV     #EDIT,KEYPR       ;KEYPR=EDIT
      INC     KKTUSB            ;KKTUSB=KKTUSB+1
LE1    JMP     LAB1              ;WAIT e dallan
LE0C   BSET   3,KIF              ;KIF[3]=1
      LDHX    FM1ADH            ;H:X=FM1ADH:FM1ADL
      INCX                     ;X=X+1
      BRA     LE0e              ;LE0e ye dallan
LE0b   ROLA    A,C                ;A yı C üzerinden sola döndür
      STA     ASAB1              ;ASAB1=A
      INC     LCDDA              ;LCDDA+1
      INC     LCDCA              ;LCDCA+1
      BHS     LE0                ;C=0 ise LE0 a dallan
      BRSET   1,KIF,LE3         ;Edite basıldı bay 1 ise LE3 e dallan
      BSET    2,KIF              ;LCDDSP devre dışı (LCD yi tarama)
      MOV     #$0F,PTA          ;LCD Disp,Cursor,Blink=ON
      JSR     LCDA               ;LCD yi bekle
      JSR     WMLCD              ;LCD ye yaz
      ;      JSR     WMLCD2        ;WMLCD2 alt prog dallan
      BSET    1,KIF              ;Edite basıldı bayrağını 1 yap
LE3    ;SEI                     ;Set IMASK Bit
      MOV     LCDDA,PTA          ;Cursorü aktif karaktere al
      JSR     LCDA               ;LCD yi bekle
      JSR     WMLCD              ;LCD ye yaz
      JMP     LAB1              ;WAIT e dallan

```

* Menü (Kursör) Format Adresini Ayarlar

```

FRMADR LDA  KCIMNU              ;KCIMNU yu A ya yükle
;      DBNZ   LF1                ;A-1=0 değil ise dallan
      CMP     #CMBSAY            ;CMBSAY ile karşılaştır
      BPL     LF1                ;Pozitif ise LF1 e dallan

```

; SAAT veya ATLA menüsünde ise

```

      CLC                     ;C=0
      ROLA    A,A*2            ;A=A*2
      ADD     #MF1ADL          ;A=A+#MF1ADL
;      ADD     #$02              ;A=A+2
      STA     FM1ADL            ;Format1 adr. düşüğe yaz
      MOV     #MF1ADH,FM1ADH   ;FM1ADH=#MF1ADH
      RTS                     ;Alt programdan geri dön

```

; SAAT veya ATLA menüsünde değil (EDIT menüsünde) ise

```

LF1    LDA     KCIEMU           ;KCIEMU yu A ya yükle
      ADD     #CMBSYD1          ;A=A+#CMBSYD1
      ROLA    A,A*2            ;A=KCIEMU*2
      ADD     #MF1ADL          ;MFormat1 adr. düşüğü A ya topla
      STA     FM1ADL            ;Format1 adr. düşüğe yaz
      LDA     KCIAMU           ;KACIANU yu A ya yükle
      ADD     #MF2ADL          ;MFormat2 adr. düşüğü A ya topla
      STA     FM2ADL            ;Format2 adr. düşüğe yaz
      RTS                     ;Alt programdan geri dön

```

; BCD Sayıyı Binary Sayıya Çevirir

```

BCDBIN LDX     #$0A            ;X=Ah
      STA     AKMTMP1          ;AKMTMP1=A
      AND     #$0F              ;A=A AND 0Fh
      STA     AKMTMP2          ;AKMTMP2=A
      LDA     AKMTMP1          ;A=AKMTMP1
      NSA                     ;A=-A
      AND     #$0F              ;A=A AND 0Fh
      MUL     A,X              ;X:A=(X)*(A)

```

```

ADD    AKMTMP2      ;A=A+AKMTMP2
RTS                      ;Alt prog geri dön

```

; Binary Sayıyı BCD Sayıya Çevirir

```

BINBCD LDX    #$0A      ;X=Ah
        DIV          ;A=(H:A)/(X) H=Kalan
        STHX    HXTMPH   ;HXTMPH:HXTMPH=H:X
        NSA
        ADD     HXTMPH   ;A=A+HXTMPH
        CLRH          ;H=0
        RTS            ;Alt prog geri dön

```

```

LAB6    JMP    LAB14
LAB7    JMP    LAB15
LLB0    JMP    LB0
LLHATGI JMP    LHATGI

```

```

LAB5    BRCLR 1,KIF,LAB6 ;Edite basılmamıştı ise LAB6 ya dallan
        CMP    #EDIT     ;EDIT tuşu mu
        BNE    LAB7      ;Değilse LAB7 ye dallan

```

* Editten çıkarken

```

MOV     #AMDND,ANMDON ;ANMDON=AMDND(20h)
CLRH          ;H=0
LDA     KCIMNU        ;A=KCIMNU
CMP     #CMBSAY       ;Çıkış Menüsü BSY ile karşılaştır
BLO     LLB0          ;Düşük menülerde LB0 a dallan
CMP     #SFMBSY       ;Şifre Menüsü BSY ile karşılaştır
BHS     LLB0          ;Şif M Büyük veta aynı menüde ise LB0 a dallan

```

* Edit Menüünden Çıkarken

```

LDA     KCIEMU        ;A=KCIEMU
CMP     #CGEMBSY      ;Çıkış Gücü Edit Menüünde mi
BNE     LEMC1         ;Değil ise LEMC1 e dallan

```

; Çıkış Gücü Edit Menüünden Çıkarken

```

LDX     #$A3          ;X=A3h
JSR     LCDCHAR       ;LCDCHAR alt prog dallan
BSR     BCDBIN        ;BCDBIN alt prog dallan
LSRA          ;A yı ikiye böl
BEQ     LEMC1b        ;0 ise LEMC1b ye dallan
CMP     #$02          ;5 ile karşılaştır
BHS     LEMC1b        ;5 den büyük veya eşit ise LEMC1b ye dallan
LDA     #$02          ;A=05h

```

; LEMC1c LSRA ;A yı sağa döndür

```

LEMC1b  STA     ODCGRA ;ODCGRA=A
        BRA     LLB1   ;LB1 e dallan
LEMC1   CMP     #SAENBSY ;SAAT edit menüsünde mi
        BNE     LEMC2   ;Değilse LEMC2 ye dal

```

; SAAT Edit Menüünden Çıkarken

```

LDHX    #ODRSAD       ;H:X=#ODRSAD (Output data RAM Saat Adr)
STHX    HXTMPH        ;HXTMPH:HXTMPL=H:X
CLR      XTEMP1        ;XTEMP1=0
MOV     #LCDS2,XTEMP2 ;XTEMP2=LCDS2 (A1h)
MOV     #$02,AKMTMP1  ;AKMTMP1=02

```

```

LEMC4   MOV     #$02,DONGU ;DONGU=02
LEMC3   LDHX    XTEMP1    ;X=XTEMP1
        JSR     LCDCHAR   ;LCDCHAR alt prog dallan
        INCX          ;X=X+1
        INCX          ;X=X+1
        STX     XTEMP2    ;XTEMP2=X

```

; Saatin doğru yazılıp yazılmadığını kontrol eder

```

LDX     DONGU          ;X=DONGU
CBEQX   #$01,LEMC3b    ;DONGU=01 ise LEMC3b ye dallan
CMP     #$24           ;A=A-24h

```

```

      BMI    LEMC3d          ;Eksi ise LEMC3c ye dallan
      BSET   0,BAYRAK0      ;BAYRAK0[0]=1
      BRA    LLHATGI        ;LLHATGI ye dallan
      BRA    LEMC3d         ;LEMC3d ye dallan
LEMC3b  CMP    #$60         ;A=A-60h
      BMI    LEMC3d         ;Negatif ise LEMC3d ye dallan
      BSET   0,BAYRAK0      ;BAYRAK0[0]=1
      BRA    LLHATGI        ;LLHATGI ye dallan
LEMC3d  LDHX   HXTMPH       ;X=HXTMPH
      INC    HXTMPL         ;HXTMPL=HXTMPL+1
      STA    ,X             ;(X)=A
      DBNZ   DONGU,LEMC3    ;DONGU-1=0 değil ise LEMC3 e dallan
      INC    XTEMP2         ;XTEMP2=XTEMP2+1
      DBNZ   AKMTMP1,LEMC4  ;AKMTMP1-1=0 değil ise LEMC4 e dallan

      BRA    LLB1           ;LB1 e dallan

LEMC2   CMP    #ISEMBSY     ;Işık Şiddeti veya Sıcaklık E Menüünde mi
      BHI    LEMC5          ;Değil ise LEMC5 e dallan
; SICAKLIK veya IŞIK ŞİDDETİ Edit Menüünden Çıkarken
      MOV    #$04,DONGU     ;DONGU=04h
      MOV    #$A3,XTEMP1    ;XTEMP1=92h
      MOV    #$92,XTEMP2    ;XTEMP2=92h
LEMC7   LDX    XTEMP1       ;X=XTEMP1
      JSR    LCDCHAR        ;LCDHAR alt prog dallan
      INCX   ,X             ;X=X+1
      INCX   ,X             ;X=X+1
      STX    XTEMP1         ;X=XTEMP1
      CMP    #$F0           ;Bosluk mu
      BEQ    LLB1           ;Eveyt ise LB1 e dallan
      JSR    BCDBIN         ;BCDBIN alt prog dallan
      STA    AKMTMP1        ;AKMTMP1=A
      LDX    XTEMP2         ;X=XTEMP2
      JSR    LCDCHAR        ;LCDHAR alt prog dallan
      INCX   ,X             ;X=X+1
      INCX   ,X             ;X=X+1
      STX    XTEMP2         ;X=XTEMP2
      CMP    #$10           ;16 dan küçük mü
      BLO    LEMSC6         ;Evet ise LEMSC6 ye dallan
      SUB    #$6            ;6 çılar BCD yi binary sayıya çevir
LEMSC6  TAX              ;X=A
      LDA    AKMTMP1        ;A=AKMTMP1
      STA    $200,X         ;(200+X)=A
      DBNZ   DONGU,LEMC7    ;DONGU-1=0 değil ise LEMC7 ye Dallan
LLB1    JMP    LB1          ;LB1 e Dallan
LEMC5   CMP    #LGEMBSY     ;Lojik veya kapı girişi edit menüsünde mi
      BHI    LEMC8          ;Değil ise LEMC8 e dallan
; Lojik veya kapı girişi edit menüsünden çılarken
      MOV    #$04,DONGU     ;DONGU=04h
      MOV    #$A3,XTEMP1    ;XTEMP1=A3h
      MOV    #$92,XTEMP2    ;XTEMP1=92h
LEMC16  LDX    XTEMP1       ;X=XTEMP1
      LDA    ,X             ;A=(X)
      CBEQA  #$20,LLB1      ;A=20h ise LB1 e dallan
      AIX    #$03           ;X=X+3h
      STX    XTEMP1         ;XTEMP1=X
      RORA   ,X             ;A yı sağa döndür
      CLRA   ,X             ;A=0
      ROLA   ,X             ;A yı sola döndür
      STA    AKMTMP1        ;AKMTMP1=A
      LDX    XTEMP2         ;X=XTEMP2
      JSR    LCDCHAR        ;LCDCHAR alt prog dallan
      INCX   ,X             ;X=X+1

```

```

INCX                ;X=X+1
STX  XTEMP2        ;XTEMP2=X
CMP  #$09           ;9 dan küçük mü
BLO  LEMC9          ;Evet ise LEMC9 ya dallan
BEQ  LEMC10         ;9 ise LEMC10 a dallan
SUB  #$6            ;A=A-6 (BCD yi Binary yap)
LEMC10 SUB  #$08     ;8 çıkar BCD yi binary sayıya çevir
BSR  LVKGAP1        ;LVKGAP1 alt prog dallan
TST  AKMTMP1        ;AKMTMP1 i test et
BEQ  LEMC12         ;0 ise LEMC12 ye dallan
ORA  ODLGRA1        ;A=A OR ODLGRA1
BRA  LEMC13         ;LEMC13 e dallan
LEMC12 COMA         ;A nın tersini al
AND  ODLGRA1        ;A=A AND ODLGRA1
LEMC13 STA  ODLGRA1  ;ODLGRA1=A
BRA  LEMC14b        ;LEMC14b ye dallan
LEMC9 BSR  LVKGAP1   ;LVKGAP1 alt prog dallan
TST  AKMTMP1        ;AKMTMP1 i test et
BEQ  LEMC14         ;0 ise LEMC14 ye dallan
ORA  ODLGRA2        ;A=A OR ODLGRA2
BRA  LEMC15         ;LEMC15 e dallan
LEMC14 COMA         ;A nın tersini al
AND  ODLGRA2        ;A=A AND ODLGRA2
LEMC15 STA  ODLGRA2  ;ODLGRA2=A
LEMC14b DBNZ DONGU,LEMC16 ;DONGU-1=0 değil ise LEMC16 ya Dallan
BRA  LB1            ;LB1 e Dallan

; L ve K girişlerinin kaçınıcı çıkışı etkilediğini bulur
LVKGAP1 STA  AKMTMP2 ;AKMTMP2=A
CLRA                ;A=0
SEC                 ;C=1
LEMC11 RORA         ;A yı sağa çevir
DBNZ AKMTMP2,LEMC11 ;AKMTMP2-1=0 değil ise LEMC11 e dallan
RTS                 ;Alt prog geri dön

LEMC8  CMP  #LIEMBSY ;Lojik işlem edit menüsünde mi
BNE  LEMC17         ;Değilise LEMC17 a dallan
; Lojik İşlem Edit Menüsünden Çıkarken
MOV  #$03,DONGU     ;DONGU=03h
CLRA                ;A=0
LDX  #$A3           ;X=A3h
LEMC18 LSLA         ;A yı lojik sola kaydır
LSLA         ;A yı lojik sola kaydır
ADD  ,X             ;A=A+ (X)
ORA  #$01           ;A=A OR $01
SUB  ,X             ;A=A- (X)
EOR  #$01           ;A=A EOR 1h
AIX  #$04           ;X=X+4
DBNZ DONGU,LEMC18   ;DONGU-1=0 değil ise LEMC18 e dallan
STA  ODLIRA         ;ODLIRA=A
BRA  LB1            ;LB1 e Dallan
; Hangi Günler Edit Menüsünden Çıkarken
LEMC17 MOV  #$07,DONGU ;DONGU=03h
CLRA                ;A=0
LDX  #$A1           ;X=A1h
LEMC19 LSLA         ;A yı lojik sola kaydır
ADD  ,X             ;A=A+ (X)
ORA  #$01           ;A=A OR $01
SUB  ,X             ;A=A- (X)
EOR  #$01           ;A=A EOR 1h
INCX                ;X=X+1
INCX                ;X=X+1

```

```

DBNZ DONGU,LEMC19 ;DONGU-1=0 değil ise LEMC19 a dallan
STA ODHGRA ;ODHGRA=A
BRA LB1 ;LB1 e Dallan

LB0 CBEQA #ATMBSY,LB2 ;A==#ATMBSY ise LB2 ye dallan
; SAAT menüsünde ise
; CLRA ;A=0 (GEÇİCİ)
CLR DAKIKA0 ;DAKIKA0=0 (GEÇİCİ)
CLR DAKIKA1 ;DAKIKA1=0 (GEÇİCİ)
CLR DAKIKA2 ;DAKIKA2=0 (GEÇİCİ)
CLR SANIYE0 ;SANIYE0=0 (GEÇİCİ)
CLR SANIYE1 ;SANIYE1=0 (GEÇİCİ)
LDX #LCDS2 ;X=LCD SATIR2 ADR. (GEÇİCİ)
BSR SIL ;SIL alt prog. dallan (GEÇİCİ)
CMP #$24 ;24h ile karşılaştır
BLO LBS1 ;24 den düşük ise LBS1 e dallan
BSET 0,BAYRAK0 ;BAYRAK0[0]=1
BRA LHATGI ;LHATGI ye dallan
; LDA #01h ;A=01h
LBS1 STA SAAT ;SAAT=A (GEÇİCİ)
BSR SIL ;SIL alt prog. dallan (GEÇİCİ)
CMP #$60 ;60h ile karşılaştır
BLO LBS2 ;60 dan düşük ise LBS2 ye dallan
BSET 0,BAYRAK0 ;BAYRAK0[0]=1
BRA LHATGI ;LHATGI ye dallan
; CLRA ;A=0
LBS2 STA DAKIKA ;DAKIKA=A (GEÇİCİ)
BSR SIL ;SIL alt prog. dallan (GEÇİCİ)
CMP #$60 ;60h ile karşılaştır
BLO LBS3 ;60 dan düşük ise LBS3 e dallan
CLRA ;A=0
LBS3 STA SANIYE ;SANIYE=A (GEÇİCİ)
INX ;X=X+1 (GEÇİCİ)
BSR SIL ;SIL alt prog. dallan (GEÇİCİ)
TSTA ;A yı testet
BEQ LBS5 ;A=0 ise LBS5 e dallan
CMP #$08 ;08h ile karşılaştır
BLO LBS4 ;08 dan düşük ise LBS4 e dallan
LBS5 BSET 0,BAYRAK0 ;BAYRAK0[0]=1
BRA LHATGI ;LHATGI ye dallan
; LDA #01h ;A=01h
LBS4 STA GUN ;GUN=A (GEÇİCİ)
; BRSET 0,BAYRAK0,LHATGI ;BAYRAK0[0]=1 LHATGI ye dallan
LB1 ;SEI ;Set IMASK Bit
; LCD Display ON/OFF u ayarlar
MOV #$0C,PTA ;LCD modu D açık C,B kapalı
JSR LCDA ;LCD yi bekle
JSR WMLCD ;LCD ye yaz
; JSR WMLCD2 ;WMLCD2 alt prog dallan
MOV #LCDS1,LCDRAM ;LCDRAM 'ı başa al
; Kursörü başa al
MOV #$80,PTA ;DD RAM=00
JSR LCDA ;LCD yi bekle
JSR WMLCD ;LCD ye yaz
LDA KIF ;A=KIF
AND #$F1 ;KIF[1,2,3]=0 yap
STA KIF ;KIF=A
BCLR 0,BAYRAK0 ;BAYRAK0[0]=0
JMP LAB1 ;LAB1 e dallan (GEÇİCİ)

LHATGI BCLR 1,KIF ;Edite basıldı bayrağını 0 yap (GEÇİCİ)
BCLR 3,KIF ;KIF[3]=0 (GEÇİCİ)
; BCLR 0,BAYRAK0 ;BAYRAK0[0]=0

```

```

CLR    ANMDON          ;ANMDON=00
JMP    LIEETB          ;LIEETB ye dallan

; ATLA menüsünde ise
LB2     LDX    #LCDS1    ;X=LCDS1
        JSR    LCDCHAR    ;LCDCHAR alt programına dallan
        NSA
        RORA
        BCC    LB3
        ROLA
        NSA
        AND    #$0F
        ADD    #$09
        BRA    LB4
LB3     ROLA
        NSA
        AND    #$0F
        DECA
        BMI    LB1
LB4     INC    KCIMNU
        STA    KKTUSB    ;KKTUSB=A
        MOV    #$38,KEYPR ;KEYPR=38 (Alt ok)
        BRA    LB1      ;LB1 e dallan

SIL     JSR    LCDCHAR    ;LCDCHAR alt programına dallan (GEÇİCİ)
        INCX    ;X=X+1 (GEÇİCİ)
        INCX    ;X=X+1 (GEÇİCİ)
        RTS      ; (GEÇİCİ)

LCDCHAR LDA    ,X        ;LCD deki karakteri A ya yaz
        CMP    #$20      ;20h (Boşluk) ile karşılaştır
        BNE    LLCH1     ;eşit değil ise LLCH1 e dallan
        ADD    #$10      ;A=A+10h
LLCH1   NSA          ;A nın nibble larını değiştir
        INCX    ;X i 1 arttır
        ADD    ,X        ;X adr. değeri A ya topla
        SUB    #$33      ;A dan 33H çıkar
        RTS      ;Alt programdan geri dön

```

* Edit basılmışken tuşa basılırsa

```

LAB15   LDA    KEYPR      ;A=KEYPR
        CBEQA  #NEXT,LAB15b ;A=NEXT ise LAB15b ye dallan
;       SEI          ;Set IMASK Bit
        MOV    KEYPR,PTA   ;Basılan tuşu PTA ye yaz
        JSR    LCDA       ;LCD yi bekle
        JSR    WDLCD      ;LCD ye yaz
        CLRH          ;H=0
        LDX    LCDCA      ;X=LCDCA
        MOV    KEYPR,X+    ;LCDRAM[X]=KEYPR
        BRA    LAB15e     ;LAB15e ye dallan
LAB15b  CLRH          ;H=0
LAB15e  BSR    EMDISE1     ;EMDISE1 alt programına dallan
        CBEQA  #$20,LAB15c ;A=20 (Boşluk) ise LAB15c ye dallan
        LDA    ASAB1      ;A=ASAB1
        JMP    LE0        ;LE0 a dallan
LAB15c  JMP    LE1b       ;LE1b ye dallan

```

; Edit Menüsünd ise 1 (Editin Bitip Bitmediğini ASAB1 ve
; üstteki yazıya bakarak anlar)

```

EMDISE1 LDA    KCIEMU      ;A=KCIEMU
;       BEQ    LAB15d      ;0 ise LAB15d ye dallan

```

```

CMP    #SIEMBSY      ;Sıcaklık E Menüünde mi
BLO    LAB15d         ;S E M den küçük ise LAB15d ye dallan
CMP    #LGEMBSY      ;Lojik Giriş E Menüünde mi
BHI    LAB15d         ;L G E M den BÜYÜK ise LAB15d ye dallan
LDA    ASAB1          ;A=ASAB1
ROLA                   ;A yı sola döndür
BCC    LAB15f         ;C=0 ise LAB15f ye dallan
CLRA                   ;A=0
RTS                   ;Alt programdan geri dön

LAB15f  LDX    LCDCA      ;X=LCDCA
        AIX    #-14t      ;X den 0Eh çıkar
        LDA    ,X         ;A=(X)
LAB15d  RTS                   ;Alt programdan geri dön

```

* Edite basılmamışken tuşa basılırsa

```

LAB14   CLRH          ;H=0
        LDA    KEYPR    ;A=KEYPR
        CMP    #$38     ;8 mi?
        BEQ    AASOK     ;8 ise aşağı oka dallan
        CMP    #$32     ;2 mi?
        BEQ    YUOK1     ;2 ise yukarı oka dallan
        CMP    #$36     ;6 mi?
        BEQ    SAOK1     ;6 ise aşağı oka dallan
        CMP    #$34     ;4 mü?
        BEQ    SOOK1     ;4 ise sol oka dallan
        CMP    #NEXT     ;NEXT mi?
        BEQ    NEXTT     ;NEXT ist NEXTT e dallan
        JMP    LAB1      ;WAIT e dallan

SOOK1   JMP    SOOK      ;SOOK a dallan
SAOK1   JMP    SAOK      ;SAOK a dallan
YUOK1   JMP    YUOK      ;YUOK a dallan
AASOK   JMP    ASOK      ;ASOK a dallan
LLNEXT1 JMP    LNEXT1    ;LNEXT1 e dallan

NEXTT   MOV    #AMDD2,ANMDON ;ANMDON=AMDD2 (A0h)
        LDA    KCIEMU    ;A=KCIEMU
        BEQ    LNEXT2    ;0 ise LNEXT2 ye dallan
        CLR    KCIEMU    ;KCIEMU=0
LLAB1   JMP    LAB1      ;LAB1 e dallan
LNEXT2  LDA    KCIAMU    ;A=KCIAMU
        BEQ    LLNEXT1   ;0 ise LNEXT1 e dallan
        CLR    KCIAMU    ;KCIAMU=0
        CLR    AMOFFSET  ;AMOFFSET=0
        CLR    EMOFFSET  ;EMOFFSET=0
        BSR    SAVEFLSH  ;SAVEFLSH alt prog dallan
        JMP    LAB1      ;LAB1 (Wait) e dallan

```

; Alt menüden çıkarken Data RAM daki bilgileri Flash Bafıra
; ve Flasha yazar

```

LLAC5   JMP    LAC5      ;LAC5 e dallan
SAVEFLSH BRCLR 0,KIF,LLAC5 ;KIF[0]=0 ise LAC5 e dallan
;        JSR    FSHBADR   ;FSHBADR alt prog dallan
        LDHX   #FBRBAD   ;H:X=#$FBRBAD
        JSR    FSHPBAD   ;FSHPBAD alt prog dallan
        STHX   HXTMPH    ;HXTMPH:HXTMPL=H:X
        CLR    XTEMP1    ;XTEMP1=0
        CLR    XTEMP2    ;XTEMP2=0
        MOV    #$03,AKMTMP1 ;AKMTMP1=03h
        MOV    #$10,DONGU ;DONGU=10h
LAC2    TST    AKMTMP1    ;AKMTMP1yi test et

```

```

BEQ    LAC3b          ;0 ise LAC3b ye dallan
LDHX   XTEMP1         ;H:X=XTEMP1:XTEMP2
INX    XTEMP1         ;X=X+1
STX    XTEMP2         ;XTEMP2=X
LDA    ODRBAD,X       ;A=(ODRBAD+X)
BNE    LAC1           ;A=0 değil ise L1 e dallan
DBNZ   DONGU,LAC2     ;DONGU-1=0 değil ise LAC2 ye dallan
BRA    LAC3           ;LAC3 e dallan
LAC1   LDHX   HXTMPH   ;H:X=HXTMPH:HXTMPL
      STA    $04,X     ;(X+4)=A
      LDA    XTEMP2    ;A=XTEMP2
      STA    ,X        ;(H:X)=A
      INC    HXTMPL    ;HXTMPL=HXTMPL+1
      DEC    AKMTMP1   ;AKMTMP1=AKMTMP1-1
      DBNZ   DONGU,LAC2 ;DONGU-1=0 değil ise LAC2 ye dallan
LAC3   LDHX   HXTMPH   ;H:X=HXTMPH:HXTMPL
      LDA    AKMTMP1   ;A=AKMTMP1
      BEQ    LAC3b     ;0 ise LAC3b ye dallan
      CLR    AKMTMP1   ;AKMTMP1=0
LAC3c  MOV    AKMTMP1,X+ ;(X)=AKMTMP1
      DBNZ   LAC3c     ;A-1=0 değil ise LAC3C ye dallan
      STX    HXTMPL    ;HXTMPL=X
LAC3b  LDHX   HXTMPH   ;H:X=HXTMPH:HXTMPL
      LDA    ODCGRA    ;A=ODCGRA
      STA    ,X        ;(X)=A
      AIX    #$4       ;X=X+4
      STX    HXTMPL    ;HXTMPL=X
      CLR    XTEMP2    ;TEMP2=0
      MOV    #$08,DONGU ;DONGU=08h
LAC4   LDHX   XTEMP1   ;H:X=XTEMP1:XTEMP2
      INCX   XTEMP1    ;X=X+1
      LDA    ODCGRA,X  ;A=(ODCGRA+X)
      STX    XTEMP2    ;XTEMP2=(X)
      LDHX   HXTMPH   ;H:X=HXTMPH:HXTMPL
      STA    ,X        ;(X)=A
      INCX   XTEMP1    ;X=X+1
      STX    HXTMPL    ;HXTMPL=X
      DBNZ   DONGU,LAC4 ;DONGU-1=0 değil ise LAC4 e dallan

      MOV    #$35,DONGU ;DONGU=35h
      MOV    #$80,HXTMPH ;HXTMPH=80h
      MOV    #$03,HXTMPL ;HXTMPL=03h
      MOV    #$01,XTEMP1 ;XTEMP1=01h
      MOV    #$A0,XTEMP2 ;XTEMP2=A0h
      JSR    WRAMPGM    ;WRAMPGM ye dallan
      JSR    FSHBADR    ;Flash Data Adr. HXTMPH:HXTMPL a yazar
;      STA    COPCTL    ;COP sayıcısını sil

      SEI
      BSET   5,T2SC     ;STOP Timer2
      BSET   5,T1SC     ;STOP Timer1
; 1-8. Çıkışlar
      MOV    RLOCON1,LTCH1D ;LTCH1D=RLOCON1
      MOV    RLOCON1,PTA   ;PTA=RLOCON1
      JSR    WLTCH1       ;WLTCH1 alt prog dallan
; 9-16. Çıkışlar
      MOV    RLOCON2,LTCH2D ;LTCH2D=RLOCON2
      MOV    RLOCON2,PTA   ;PTA=RLOCON1
      JSR    WLTCH2       ;WLTCH1 alt prog dallan

;      SEI
      LDA    FLCR         ;A=FLCR
      ORA    #00000010Q   ;ERASE=1

```

T.C. MİLLÎ EĞİTİM BAKANLIĞI
 İKTİSADİ VE İŞTİRAKİ
 BİLİMLER ANA BİLİM DALI
 12. SINIF
 FİZİK DERSİ
 ÖĞRETİM YILI 2022-2023

```

STA  FLCR                ;FLCR=A
JSR  $01A0               ;01A0 adr dallan

MOV  #$50,DONGU          ;DONGU=50h
MOV  #$80,HXTMPH         ;HXTMPH=80h
MOV  #$38,HXTMPL         ;HXTMPL=03h
MOV  #$01,XTEMP1         ;XTEMP1=01h
MOV  #$A0,XTEMP2         ;XTEMP2=A0h
JSR  WRAMPGM             ;WRAMPGM ye dallan
JSR  FSHBADR             ;Flash Data Adr. HXTMPH:HXTMPL a yazar
AIX  #$7F                ;H:X=H:X+7Fh

LDA  FLCR
ORA  #00000001Q         ;PGM=1
STA  FLCR                ;FLCR=A
JSR  $01A0               ;01A0 adr dallan
BCLR 0,KIF               ;KIF[0]=0
BCLR 5,T1SC              ;START Timer1
BCLR 5,T2SC              ;START Timer2
BCLR 6,BAYRAK0           ;IRQINT eneble
BCLR 1,INTSCR            ;IMASK=0
CLI
LAC5 RTS                 ;Alt prog geri dön

LNEXT1 MOV #SMNBSY,KCIMNU ;KCIMNU=SMNBSY(1)
LLLAB22 BRA LLAB22        ;LAB22 ye dallan
ASOK21 JMP ASOK2          ;ASOK2 ye dallan
LAOKK3 JMP LAOK3          ;LAOK3 e dallan
ASOK MOV #AMDND,ANMDON    ;ANMDON=AMDND(20h)
LDA KCIAMU                ;KCIAMUyu test et
BEQ ASOK21                ;0 ise ASOK2 ye dallan
CLR KCIEMU                ;KCIEMU=0
LDA KCIMNU                ;A=KCIMNU
CMP #SFMBYSY              ;SFMBYSY ile karşılaştır
BNE LAOK8                 ;Sifre Menüsünde değil ise LAOK8 e dallan
; Sifre Alt Menüsünde ise
LDA KCIAMU                ;A=KCIAMU
CMP #SAMSSY               ;Sifre Alt Menüsünün sonunda mı
BNE LAOKK3                ;değil ise LAOK3 e dallan
MOV #YUOKT,KEYPR          ;KEYPR=YUOKT
MOV #SAMSTB,KKTUSB        ;KKTUSB=SAMSTB
BRA LLAB22                ;LB22 ye dallan
LAOK8 CMP #GAYMBSY        ;Giriş Ayarları Menüsünde mi
BNE LAOK10                ;Değil ise LAOK10 a dallan
; Giriş Ayarları Alt Menüsünde ise
LDA KCIAMU                ;A=KCIAMU
CMP #GAAMSSY              ;Giriş Ayarları Alt Menüsü Son sayısı mı
BNE LAOKK3                ;Değil ise LAOK3 e dallan
MOV #YUOKT,KEYPR          ;KEYPR=YUOKT
MOV #GAAMSTB,KKTUSB       ;KKTUSB=GAAMSTB
BRA LLAB22                ;LAB22 ye dallan
LAOK10 CMP #CAYMBSY       ;Giriş Ayarları Menüsünde mi
BNE LAOK11                ;Değil ise LAOK11 e dallan
; Çıkış Ayarları Alt Menüsünde ise
LAOK10b LDA KCIAMU        ;A=KCIAMU
CMP #CAAMSSY              ;Çıkış Ayarları Alt Menüsü Son sayısı mı
BNE LAOKK3                ;Değil ise LAOK3 e dallan
MOV #YUOKT,KEYPR          ;KEYPR=YUOKT
MOV #CAAMSTB,KKTUSB       ;KKTUSB=CAAMSTB
BRA LLAB22                ;LB22 ye dallan
LAOK11 CMP #PAYMBSY       ;Program Ayarları Menüsünde mi
BNE LAOK12                ;Değil ise LAOK12 ye dallan

```

; Program Ayarları Alt Menüsünde ise

```
LDA KCIAMU ;A=KCIAMU
CMP #PAAMBSYG ;Program Ayarları Alt Mnu GBaşlangıç sayısı mı
BNE LAOK3 ;Değil ise LAOK3 e dallan
BRA LLAB22 ;LB22 ye dallan
```

;Yeni Menüleri Buraya Ekle

; Çıkış veya Giriş Alt Menüsünde ise

```
LAOK12 LDA KCIAMU ;A=KCIAMU
CMP #AMNUSY ;A=AMNUSY
BNE LAOK1 ;Eşit değilse LAOK1 e dallan
MOV #$01,KCIAMU ;KCIAMU=01
CLR AMOFFSET ;AMOFFSET=0
CLR EMOFFSET ;EMOFFSET=0
MOV #AMDD2,ANMDON ;ANMDON=AMDD2 (A0h)
LLAB22 JMP LAB22 ;LAB22 ye dallan
```

* ALTMENÜYÜ Girişlere göre değiştirir

```
LAOK1 CMP #$02 ;02 mi
BEQ LAOK4 ;2 ise LAOK4 e dallan
CMP #$03 ;03 mü
BEQ LAOK4b ;2 ise LAOK4b ye dallan
CMP #$04 ;04 mü
BEQ LAOK5 ;3 ise LAOK5 e dallan
CMP #$05 ;05 mü
BEQ LAOK6 ;5 ise LAOK6 e dallan
CMP #$06 ;06 mi
BEQ LAOK7 ;6 ise LAOK7 e dallan
BRA LAOK3 ;LAOK3 e dallan
LAOK4 JSR AMCSAY ;Alt Menü Karakter Sayısı alt prog dallan
MOV #$01,KCIAAMU ;KCIAAMU=1
LDA SICAMSAY ;A=SICAMSAY
BNE LLAB22 ;0 değil ise LAB22 ye dallan
CLR KCIAAMU ;KCIAAMU=0 (Gerekli mi?)
BRA LAOK5b ;LAOK5b ye dallan

LAOK4b LDA SICAMSAY ;A=SICAMSAY
BEQ LAOK5b ;0 ise LAOK5b ye dallan
BRA LAOK9 ;LAOK9 a dallan
LAOK5b JSR AMCSAY ;Alt Menü Karakter Sayısı alt prog dallan
LAOK5 LDA ISKAMSAY ;A=ISKAMSAY
BEQ LAOK5c ;0 ise LAOK5c e dallan
BRA LAOK9 ;LAOK9 a dallan
LAOK5c JSR AMCSAY ;Alt Menü Karakter Sayısı alt prog dallan
LAOK6 LDA KAGAMSAY ;A=KAGAMSAY
BEQ LAOK6b ;0 ise LAOK6b ye dalla
BRA LAOK9 ;LAOK9 a dallan
LAOK6b JSR AMCSAY ;Alt Menü Karakter Sayısı alt prog dallan
LAOK7 LDA LOGAMSAY ;A=LOGAMSAY
BEQ LAOK3c ;0 ise LAOK3c ye dallan
CMP KCIAAMU ;Kaçınıcı Aynı Alt Menu ile karşılaştır
BEQ LAOK3C ;Eşit ise LAOK3C ye dallan
LAOK9 CMP KCIAAMU ;Kaçınıcı Aynı Alt Menu ile karşılaştır
BEQ LAOK3b ;Eşit ise LAOK3b e dallan
INC KCIAAMU ;KCIAAMU=KCIAAMU+1
JMP LAB22 ;LAB22 ye dallan
LAOK3b JSR AMCSAY ;Karskter sayısı alt prog. dallan
CLR KCIAAMU ;KCIAAMU=0
JMP ASOK ;ASOK e dallan
LAOK3c LDA KCIMNU ;A=KCIMNU
CBEQA #GRMBSY,LAOK3d ;Giriş Menüsünde ise LAOK3d ye dallan
LAOK3 JSR AMCSAY ;Karskter sayısı alt prog. dallan
```

```

MOV    #$01,KCIAAMU    ;KCIAAMU=1
JMP    LAB22            ;LAB22 ye dallan
LAOK3d MOV    #$01,KCIAAMU    ;KCIAAMU=1
MOV    #YUOKT,KEYPR    ;KEYPR=YUOKT
LDA    SICAMSAY        ;A=SICAMSAY
ADD    ISKAMSAY        ;A=A+ISKAMSAY
ADD    KAGAMSAY        ;A=A+KAGAMSAY
ADD    LOGAMSAY        ;A=A+LOGAMSAY
DECA   ;A=A-1
STA    KKTUSB          ;KKTUSB=A
BRA    LLAB22          ;LAB22 ye dallan

ASOK2  LDA    KCIMNU      ;A=KCIMNU
CMP    #MNUSAY          ;A=MNUSAY
BNE    LAOK2            ;Eşit değilse LAOK2 e dallan
MOV    #$01,KCIMNU      ;KCIMNU=01
LAAB22 JMP    LAB22      ;LAB22 ye dallan

LAOK2  CMP    #SCMSY      ;SCMSY (12H) ile karşılaştır
BPL    LAOK2b          ;Pozitif ise LAOK2b ye dallan
SUB    #ATMBSY          ;A=A-ATMBSY(2)
BPL    LASO1            ;Pozitif ise LASO1 e dallan
;Saat veya Çıkış Menüsünden Büyük Menülerde ise
LAOK2b JSR    MUDSPADR    ;Menü Display Alt prog. dalan
LDX    #MCOFFSET        ;X=#MCOFFSET
JSR    CSAYISI          ;Karskter sayısı alt prog. dallan
;Çıkış veya Atla Menüsünde ise
LASO1  INC    KCIMNU      ;KCIMNU=KCIMNU+1
BRA    LAAB22          ;LAB22 ye dallan

YUOKK2 JMP    YUOK2        ;YUOK2 ye dallan
YUOK   MOV    #AMDD2,ANMDON ;ANMDON=AMDD2(A0h)
TST    KCIAMU           ;KCIAMUyu test et
BEQ    YUOKK2          ;0 ise YUOK2 ye dallan
CLR    KCIEMU           ;KCIEMU=0
LDA    KCIMNU           ;A=KCIMNU
CMP    #SFMSY           ;SFMSY ile karşılaştır
BNE    LYOK7            ;Sifre Menüsünde değil ise LYOK7 ye dallan
; Şifre Alt Menüsünde ise
LDA    KCIAMU           ;A=KCIAMU
CMP    #SAMBSYG          ;Şifre Alt Menüünün Başında mı
BNE    LYOK1b           ;değil ise LYOK1b ye dallan
MOV    #ASOKT,KEYPR     ;KEYPR=ASOKT
MOV    #SAMSTB,KKTUSB   ;KKTUSB=#SAMSTB
BRA    LAAB22           ;LB22 ye dallan
LYOK7  CMP    #GAYMSY     ;Giriş Ayarları Mnu'sünde mi
BNE    LYOK8            ;Değilse LYOK8 e dallan
; Giriş Ayarları Alt Menüsünde ise
LDA    KCIAMU           ;A=KCIAMU
CMP    #GAAMBSYG        ;Giriş Ayarları A. Mnu. başında mı
BNE    LYOK1b           ;Değilse LYOK1b ye dallan
MOV    #ASOKT,KEYPR     ;KEYPR=ASOKT
MOV    #GAAMSTB,KKTUSB  ;KKTUSB=#GAAMSTB
BRA    LAAB22           ;LB22 ye dallan
LYOK8  CMP    #CAYMSY     ;Giriş Ayarları Menüsünde mi
BNE    LYOK9            ;Değilse LYOK9 a dallan
; Çıkış Ayarları Alt Menüsünde ise
LDA    KCIAMU           ;A=KCIAMU
CMP    #CAAMBSYG        ;Çıkış Ayarları A. Mnu. başında mı
BNE    LYOK1b           ;Değilse LYOK1b ye dallan
MOV    #ASOKT,KEYPR     ;KEYPR=ASOKT
MOV    #CAAMSTB,KKTUSB  ;KKTUSB=#CAAMSTB

```

```

      BRA    LAAB22          ;LB22 ye dallan
LYOK9  CMP    #PAYMBSY      ;Program Ayarları Menüsünde mi
      BNE    LYOK10         ;Değilse LYOK10 a dallan
; Program Ayarları Alt Menüsünde ise
      BRA    LAAB22          ;LB22 ye dallan
LYOK10 CMP    #GRMBSY       ;Giriş Menüsünde mi
      BNE    LYOK11         ;Değilse LYOK11 e dallan

```

; Yeni Alt Menüleri Buraya Yaz

; Çıkış Alt Menüsünde ise

```

LYOK11 LDA    KCIAMU        ;A=KCIAMU
      DECA    A              ;A=A-1
      BNE    LYOK1          ;0 değil ise LYOK1 ye dallan
      MOV    #ASOKT,KEYPR    ;KEYPR=ASOKT
      LDA    SICAMSAY
      ADD    ISKAMSAY
      ADD    KAGAMSAY
      ADD    LOGAMSAY
      ADD    #CAMSTB
      STA    KKTUSB
      BRA    LAB22          ;LAB22 ye dallan

LYOK1  DEC    KCIAMU        ;KCIAMU=KCIAMU-1
      BNE    LAB22          ;0 değilse LAB22 ye dallan
      INC    KCIAMU        ;KCIAMU=1
LYOK1b JSR    AMNCSAY        ;Alt Mnu. Neg. Karakter say. alt prog. dal.
      LDA    KCIAMU        ;A=KCIAMU
      CMP    #$03           ;03 mü
      BNE    LYOK3          ;3 değil ise LYOK3 e dallan
      LDA    SICAMSAY        ;A=SICAMSAY
      BRA    LYOK6          ;LYOK6 ya dallan
LYOK3  CMP    #$04           ;04 mü
      BNE    LYOK4          ;4 değil ise LYOK4 e dallan
      LDA    ISKAMSAY        ;A=ISKAMSAY
      BRA    LYOK6          ;LYOK6 ya dallan
LYOK4  CMP    #$05           ;05 mü
      BNE    LYOK5          ;5 değil ise LYOK5 e dallan
      LDA    KAGAMSAY        ;A=KAGAMSAY
      BRA    LYOK6          ;LYOK6 ya dallan
LYOK5  CMP    #$06           ;06 mi
      BNE    LYOK1c         ;6 değil ise LYOK1c ye dallan
      LDA    LOGAMSAY        ;A=LOGAMSAY
LYOK6  BEQ    LYOK1b         ;0 ise LYOK1b ye dallan
      STA    KCIAMU        ;KCIAMU=A
      BRA    LAB22          ;LAB22 ye dallan

LYOK1c LDA    KCIAMU        ;A=KCIAMU
      CMP    #GRMBSY        ;Giriş Menüsünde mi
      BNE    LAB22          ;Değil ise LAB22 ye dallan
      MOV    #ASOKT,KEYPR    ;KEYPR=ASOKT
      LDA    SICAMSAY        ;A=SICAMSAY
      ADD    ISKAMSAY        ;A=A+ISKAMSAY
      ADD    KAGAMSAY        ;A=A+KAGAMSAY
      ADD    LOGAMSAY        ;A=A+LOGAMSAY
      STA    KKTUSB        ;KKTUSB=A
      BRA    LAB22          ;LAB22 ye dallan

YUOK2  DEC    KCIAMU        ;KCIAMU=KCIAMU-1
      BNE    LYOK2          ;0 değil ise LYOK2 ye dallan
      INC    KCIAMU        ;KCIAMU=1
      MOV    #ASOKT,KEYPR    ;KETPR=#ASOKT

```

```

MOV    #MNUSTB,KKTUSB ;KKTUSB=#MNUSTB(16H 22D)
;      BRA    LAB22      ;LAB22 ye dallan

LAB22   JMP    LAB1      ;WAIT e dallan

LYOK2   LDA    KCIMNU    ;A=KCIMNU
        CMP    #SCMSY    ;SCMSY (12) ile karşılaştır
        BMI    LYU01     ;Eksi ise LYU01 e dallan
;Çıkış Menüünden Büyük Menülerde ise
        JSR    MUDSPADR   ;Menü Display Alt prog. dalan
        LDX    #MCOFFSET  ;X=#MCOFFSET
        BSR    NCSAYISI   ;Negatif Karskter sayısı alt prog. dallan
;Çıkış veya Atla Menüünde ise
LYU01   BRA    LAB22      ;LAB22 ye dallan
SAOK    BRCLR  0,KIF,LSAK2 ;KIF[0]=0 ise LSAK2 ye dallan
        JSR    SAVEFLSH   ;SAVEFLSH alt prog dallan
LSAK2   LDA    KCIPRG     ;A=KCIPRG
        CMP    #$08       ;A=08 mi
        BNE    LSAK1      ;Değil ise LSAK1 e dallan
        CLR    KCIPRG     ;KCIPRG=0
LSAK1   INC    KCIPRG     ;KCIPRG=KCIPRG+1
        JMP    LAB4k      ;LAB4k ye dallan
;      BRA    LAB22b      ;LAB22b ye dallan
SOOK    BRCLR  0,KIF,LSOK2 ;KIF[0]=0 ise LSOK2 ye dallan
        JSR    SAVEFLSH   ;SAVEFLSH alt prog dallan
LSOK2   LDA    KCIPRG     ;A=KCIPRG
        CMP    #$01       ;A=04 mü
        BNE    LSOK1      ;Değil ise LSOK1 e dallan
        MOV    #$09,KCIPRG ;KCIPRG=09
LSOK1   DEC    KCIPRG     ;KCIPRG=KCIPRG-1
        JMP    LAB4k      ;LAB4k ye dallan
;      BRA    LAB22b      ;LAB22b ye dallan

* ALT OKA BASILDIĞI DURUM İÇİN OFFSETİ AYARLAR
AMCSAY  JSR    LTSTAMU    ;Menü Disp. Alt prog.(LTSTAMU) dalan
;      JSR    MUDSPADR   ;Menü Display Alt prog. dalan
;      INC    KCIAMU     ;KCIAMU=KCIAMU+1
        LDX    #AMOFFSET  ;X=#AMOFFSET
        BSR    CSAYISI    ;CSAYISI alt prog. dallan
        LDA    KCIAMU     ;A=KCIAMU
        INC    KCIAMU     ;KCIAMU=KCIAMU+1
        JSR    LMC1b      ;Menü Disp. Alt prog.(LMC1b) dalan
        LDX    #EMOFFSET  ;X=#EMOFFSET
CSAYISI  ROLA          ;A yı C üzerinden sola döndür
        BHS    LD0y       ;C=0 ise LD0y ye dallan
        INC    ,X
LD0y    DBNZ    DONGU,CSAYISI ;DOGU-1=0 ise CSAYISI ye dallan
        BRSET  1,BAYRAK0,LD3y ;BAYRAK0[1]=1 ise LD3y ye dallan
        LDA    SABLON2    ;A=SABLON2
        MOV    #$08,DONGU  ;DONGU=8H
        BSET  1,BAYRAK0    ;BAYRAK0[1]=1
        BRA    CSAYISI     ;CSAYISI ye dallan
LD3y    BCLR  1,BAYRAK0    ;BAYRAK0[1]=0
;      CLC              ;C=0
        RTS              ;Alt programdan geri dön

```

* ÜST OKA BASILDIĞI DURUM İÇİN OFFSETİ AYARLAR

```

AMNCSAY DEC    KCIAMU    ;KCIAMU=KCIAMU-1
        JSR    LTSTAMU    ;Menü Disp. Alt prog.(LTSTAMU) dalan
;      JSR    MUDSPADR   ;Menü Display Alt prog. dalan
        LDX    #AMOFFSET  ;X=#AMOFFSET
        BSR    NCSAYISI   ;NCSAYISI alt prog. dallan
        LDA    KCIAMU     ;A=KCIAMU

```

```

JSR    LMC1b          ;Menü Disp. Alt prog.(LMC1b) dalan
LDX    #EMOFFSET      ;X=#EMOFFSET
NCSAYISI ROLA         ;A yı C üzerinden sola döndür
BHS    LD0z           ;C=0 ise LD0y ye dallan
DEC    ,X
LD0z   DBNZ DONGU,NCSAYISI ;DOGU-1=0 ise NCSAYISI ye dallan
BRSET  1,BAYRAK0,LD3z ;BAYRAK0[1]=1 ise LD3z ye dallan
LDA    SABLON2        ;A=SABLON2
MOV    #$08,DONGU     ;DONGU=8H
BSET   1,BAYRAK0      ;BAYRAK0[1]=1
BRA    NCSAYISI       ;NCSAYISI ye dallan
LD3z   BCLR 1,BAYRAK0 ;BAYRAK0[1]=0
;      CLC            ;C=0
RTS     ;Alt programdan geri dön

ORG    KEYINT
CLR    KBIER          ;KEYINT devre dışı
MOV    #$06,KBSCR     ;int. req. sil imask aktif kenar tetik.
CLI    ;Kesmeler Aktif
BRSET  6,BAYRAK0,LIDE3 ;BAYRAK0[6]=1 ise LIDE1 e dallan
BCLR   1,INTSCR       ;IMASK1=0 IRQ int eneble
LIDE3  ;BRSET 5,BAYRAK0,LIDE4 ;BAYRAK0[5]=1 ise LIDE2 e dallan
BSET   6,T1SC         ;TOIE=1 Timer Int eneble
LIDE4  BSET 6,T2SC     ;TOIE=1 Timer Int eneble
MOV    #$FF,KEYTEMP3 ;KEYTEMP3 e FFH yaz
BRSET  0,KEYTEMP1,LK0 ;KEYTEMP0=1 ise LK0 a dallan
;      MOV    #$06,KBSCR ;int. req. sil imask aktif kenar tetik.
INC    KEYTEMP1       ;KEYTEMP1=1
RTI
LK0    PSHH           ;H yi RAMa it
CLR    CLRX           ;X e 00H yaz
CLR    CLRH           ;H=0

;      MOV    #$03,PTAPUE ;DENEME

MOV    #$70,PTA       ;PTA ya 70H yaz 1.sarır
LDA    PTA            ;PTA yı A ya oku
COMA   ;A nın tersini al
AND    #$07           ;A yı 07H ile AND le
BNE    LK1            ;0 değil ise LK1 e dallan
LDX    #$3            ;X e 3H yaz
MOV    #$68,PTA       ;PTA ya 68H yaz 2.sarır
LDA    PTA            ;PTA yı A ya oku
COMA   ;A nın tersini al
AND    #$07           ;A yı 07H ile AND le
BNE    LK1            ;0 değil ise LK1 e dallan
LDX    #$6            ;X e 6 yaz
MOV    #$58,PTA       ;PTA ya 58H yaz 3.sarır
LDA    PTA            ;PTA yı A ya oku
COMA   ;A nın tersini al
AND    #$07           ;A yı 07H ile AND le
BNE    LK1            ;0 değil ise LK1 e dallan
LDX    #$9            ;X e 9 yaz
;      CLR    PTA      ;PTA ya 00H yaz 4.sarır
MOV    #$38,PTA       ;PTA ya 38H yaz 4.sarır
LDA    PTA            ;PTA yı A ya oku
COMA   ;A nın tersini al
AND    #$7            ;A yı 7H ile AND le
BNE    LK1            ;0 değil ise LK1 e dallan
BRA    LK7b           ;Kesme programından çık

```

```

LK1      STX    KEYTEMP1      ;X i KEYTEMP1 e yaz
        CBEQA  #$1,LK2        ;1.sütunda ise LK2 ye dallan
        CBEQA  #$2,LK3        ;2.sütunda ise LK3 ye dallan
        CBEQA  #$4,LK4        ;3.sütunda ise LK4 ye dallan
        BRA    LK7b           ;Kesme programından çık
LK2      LDA    #$1            ;A ya 1 yaz
        BRA    LK5            ;LK5 e dallan
LK3      LDA    #$2            ;A ya 2 yaz
        BRA    LK5            ;LK5 e dallan
LK4      LDA    #$3            ;A ya 3 yaz
LK5      ADD    KEYTEMP1       ;KEYTEMP1 i A ya topla
        CMP    #$B            ;Tuş sıfır mı
        BNE    LK7            ;Sıfır değil ise LK7 ye dallan
        CLRA                     ;A yı sil
LK7      ADD    #$30           ;A ya 30 topla
        TST    ANMDON          ;ANMDON 0 mı
        BEQ    LK7C           ;0 ise LK7C ye dallan
        CBEQA  #EDIT,LK7b     ;A=#EDIT ise LK7b ye dallan
LK7C     INC    KKTUSB         ;KKTUSB=KKTUSB+1
        STA    KEYPR          ;Tuşu KEY e yaz
;        MOV    #$01,KKTUSB
LK7b     CLR    KEYTEMP1       ;KEYTEMP1 i sil
        MOV    #$07,KBSCR      ;int. req. sil imask aktif,seviye tetiklmeli
        PULH                     ;Hyi RAMdan çek
        RTI                     ;Kesme programından çık

```

```

        ORG    IRQINT
        BRCLR  3,BAYRAK0,LEIR1 ;TIMINT te değildi ise LEIR1 e dallan
        AIS    #5t
LEIR1    CLR    KCITIP1        ;KCITIP1=00
        LDA    T1SC           ;A=T1SC
        MOV    #$50,T1SC      ;TOF=0 TOIE=1 TRST=1
        BSET   1,INTSCR        ;IRQINT disable
        BSET   6,BAYRAK0      ;IRQINT disable
        JMP    TIMINT         ;Software interrupt

```

```

        ORG    TIMINT
        BCLR   7,T1SC          ;TOF 'u sıfırla
LTMITO   BRSET  3,BAYRAK0,LTMIT1
        BSET   3,BAYRAK0
        LDA    DDRA
        PSHA
        MOV    #$FF,DDRA      ;PTA çıkış
        LDA    PTA
        PSHA
        PSHH                     ;H yi RAMa it
LTMIT1   BCLR   6,T2SC         ;TOIE=0 Timer Int disadle
        CLI
;        MOV    #$FF,DDRA      ;PTA çıkış
LTMIT2   INC    KCITIP1        ;KCITIP1=KCITIP1+1
        LDHX   #$0108          ;H:X=0181h
        CLR    RLOCON1        ;RLOCON=00
        CLR    RLOCON2        ;RLOCON=00
        CLR    CKDATA         ;CKDATA=00
        CLC                     ;C=0

```

```

; 8-1 .Çalış için diming kontrolü
LTI2b    LDA    $80,X          ;A=(H:X+80) 8.Çıkışın çıkış gücü
;        LDA    ,X              ;A=(X)
        BNE    LTI2            ;0 değil ise LTI2 ye dallan
        ROR    RLOCON1         ;RLOCON yi sol döndür

```

```

LTI2    BRA    LTI5            ;LTI5 e dallan
        SEC                    ;C=1
        ROR    RLOCON1        ;RLOCON yi sol döndür
        SUB    KCITIP1        ;A=A-KCITIP1
        BNE    LTI3            ;0 (Eşit) değilse LTI3 e dallan
        SEC                    ;0 ile -2 arasında ise (Eşit veya 1 fazla)
        ROR    CKDATA         ;CKDATA yı sola döndür
        BRA    LTI4            ;LTI4 e dallan
LTI3    CLC                    ;C=0
LTI5    ROR    CKDATA         ;CKDATA yı sola döndür
LTI4    DBNZX LTI2b            ;X-1=0 değil ise LTIEB te dallan
        LDA    RLOCON1        ;A=RLOCON
        AND    RLOCK1         ;A=A AND RLOCK1 (On olan Lojik çılışlar(1-8))
        STA    RLOCON1        ;A=RLOCON1
        ORA    CKDATA         ;Son Çıkış Datası (1-8)
        STA    LTCH1D         ;LTCH1D=A
        STA    PTA            ;PTA=A
;        JSR    WLTCH1        ;WLTCH1 alt prog dallan

; 16-7. Çıkışlar
        LDX    #$08            ;X=08h
        CLR    CKDATA         ;CKDATA=00
        CLC                    ;C=0

; 16-9 .Çılış için diming kontrolü
LTI2bbb LDA    $88,X           ;A=(H:X+80) 8.Çıkışın çıkış gücü
        BNE    LTI2bb         ;0 değil ise LTI2 ye dallan
        ROR    RLOCON2        ;RLOCON yi sol döndür
        BRA    LTI5b          ;LTI5 e dallan
LTI2bb  SEC                    ;C=1
        ROR    RLOCON2        ;RLOCON yi sol döndür
        SUB    KCITIP1        ;A=A-KCITIP1
        BNE    LTI3b          ;0 (Eşit) değilse LTI3b ye dallan
        SEC                    ;-1 ile -2 arasında ise (KCITIP1 Çkş gcndn
1vy2 fzla)
        ROR    CKDATA         ;CKDATA yı sola döndür
        BRA    LTI4b          ;LTI4 e dallan
LTI3b   CLC                    ;C=0
LTI5b   ROR    CKDATA         ;CKDATA yı sola döndür
LTI4b   DBNZX LTI2bbb         ;X-1=0 değil ise LTIEB te dallan
;        JSR    WLTCH1        ;WLTCH1 alt prog dallan
        MOV    #$FF,DDRA      ;PTA çıkış
        LDA    PTC            ;A=PTC
        EOR    #LTCH1E        ;LATCH1 aktif
        SEI                    ;
        STA    PTC            ;PTC=A
        EOR    #LTCH1E        ;LATCH1 pasif
        STA    PTC            ;PTC=A
        CLI                    ;
        LDA    RLOCON2        ;A=RLOCON
        AND    RLOCK2         ;A=A AND RLOCK1 (On olan Lojik çılışlar(1-8))
        STA    RLOCON2        ;A=RLOCON1
        ORA    CKDATA         ;Son Çıkış Datası (1-8)
        STA    LTCH2D         ;LTCH2D=A
        STA    PTA            ;PTA=A
;        JSR    WLTCH2        ;WLTCH2 alt prog dallan
        LDA    PTC            ;A=PTC
        EOR    #LTCH2E        ;LATCH2 aktif
        SEI                    ;
        STA    PTC            ;PTC=A
        EOR    #LTCH2E        ;LATCH2 pasif
        STA    PTC            ;PTC=A
        CLI                    ;
        LDA    KCITIP1

```

```

CMP    #49T          ;47T ile karşılaştır
BLO    LTMIT3
BSET   5,T1SC        ;STOP Timer1
LTMIT3 CMP    #$18
BNE    LTI6
BSET   2,INTSCR       ;ACK1
BCLR   6,BAYRAK0     ;IRQINT eneble
BCLR   1,INTSCR       ;IMASK=0

LTI6   SEI
BSET   6,T2SC         ;TOIE=0   Timer Int eneble
PULH
PULA
STA    PTA
PULA
STA    DDRA
BCLR   3,BAYRAK0
LTI7   RTI

```

```

ORG    TIM2INT
BCLR   7,T2SC         ;TOF 'u sıfırla
BRCLR  2,BAYRAK0,LHG1 ;TIM2INT e girmemişdi ise LHG1 e dallan
RTI                ;Interruptan geri dön

SSKPLI2 JMP    SKPLI2      ;SKPLI2 ye dallan
LSATBAK JMP    SATBAK      ;SATBAK a dallan

LHG1   BSET   2,BAYRAK0    ;BAYRAK[0]=1 TIM2INT e girdi bayrağı
      CLI    ;I=0 interrupt gelebilir
      PSHH   ;H yi RAMa it
      ;      MOV    #$FF,DDRA ;PTA çıkış
      INC    KCITI         ;KCITI=KCITI+1
LHG2   LDHX   HXTPHI       ;H:X=HXTPHI:HXTPLI
      LDA    13t,X         ;A=İşlem
      STA    ISLTMP        ;ISLTMP=A
      ; Güne Göre Çıkışın AÇIK veya KAPALI olduğunu Bulur
      AIX    #14T         ;H:X=H:X+14d
      MOV    X+,RLCKTP     ;RLCKTP(Geçici)=X
      LDA    GUN           ;A=GUN
      INCA    ;A=A+1
LHG3   ROL    RLCKTP       ;RLCKTP (TEMP) yi sola döndür
      DBNZA  LHG3         ;A-1=0 değil ise LHG3 e dallan
      BCC    SSKPLI2      ;C=0 ise Kapalıya dallan
      ; Girişler Göre Çıkışın AÇIK veya KAPALI olduğunu Bulur
      LDHX   HXTPHI       ;H:X=HXTPHI:HXTPLI
      LDA    ,X           ;A=Bağlı Giriş
      ADD    11t,X         ;Bağlı Olduğu Lojik Girişler1
      ADD    12t,X         ;Bağlı Olduğu Lojik Girişler2
      BEQ    LSATBAK      ;Hiçbir girişe Bağlı Değil ise SATBAK a dallan
      MOV    #$03,DONGUI   ;DONGUI=03h
LG0    LDHX   HXTPHI       ;H:X=HXTPHI:HXTPLI
      LDX    ,X           ;X=(X)
      BEQ    LG2b         ;X(Bağlı olunan giriş)=0 ise LG2b ye dallan
      CPX    #$08         ;X i 8 ile karşılaştır
      BHI    LG2b         ;8. girişten büyük ise LG2b ye dallan
      CLRH   ;H=0
      LDA    $3F,X        ;A=($3F+X) (Giriş portunun değeri)
      LDHX   HXTPHI       ;H:X=HXTPHI:HXTPLI
      INC    HXTPLI       ;HXTPLI=HXTPLI+1
      CMP    $04,X        ;Girilen giriş değeri ile karşılaştır
      BLO    LG2          ;Okunan değer düşük ise LG2 (Açığa) dallan

```

```

BHI    LG3                ;Okunan değer yüksek ise LG3 (Kapalıya) dallan
CLR    CLRH               ;H=0
LDX    KCITI              ;X=KCITI
DECX                   ;X=X-1
LSRX                   ;X i lojik sağa döndür (Okunan değer eşit ise)
LSRX                   ;X i lojik sağa döndür (ikiye böl)
LSRX                   ;X i lojik sağa döndür
LDA     $181,X            ;A=(181+X) Çalışan çıkış gücü
BEQ     LG3               ;Çıkış gücü=00 ise LG3 (kapalı) e dallan
BRA     LG2C              ;LG2C ye dallan (Devamet)

```

```

SSACIK3 JMP    SACIK3      ;SACIK3 e dallan
SSKPLI3 JMP    SKPLI3     ;SKPLI3 e dallan

```

```

LG3     BRCLR 2,ISLTMP,SSKPLI3 ;0 (ve) ise SKPLI (Kapalı) ya dallan
LG2c    DBNZ  DONGUI,LG0      ;DONGUI-1=0 değil ise LG0 A dallan
LG2b    LDA    HXTPLI        ;A=HXTPLI
        AND    #$F0          ;A=A AND F0h
        STA    HXTPLI        ;HXTPLI=A(Programın başlangıç adrsi)
        LDHX   HXTPHI        ;H:X=HXTPHI:HXTPLI

```

```

; Lojik Girişleri Değerlendirir
; PTB[7-0] (1-8) Girişleri için

```

```

LG4     LDA    12t,X          ;A=(11+X) (Bağlı olduğu lojik giriş)
        AND    LOJG1          ;A=A AND LOJG1
        BEQ    LG5            ;0 ise (9-16) Girişlerine dallan Bağlı değil
        STA    AKTMP1I        ;AKTMP1I=A
        AND    PTBD          ;A=A AND PTBD
        BEQ    LG4b           ;Hepsi 0 ise LG4b ye dallan
        BRSET 2,ISLTMP,SSACIK3 ;1 (Veya) ise SACIK (AÇIK) a dallan
        CMP    AKTMP1I        ;A yı AKTMP1I ile karşılaştır
        BEQ    LG5            ;0 ise (9-16) Girişlerine dallan ve iken hepsi

```

```

1
LG4b    BRCLR 2,ISLTMP,SSKPLI3 ;0 (ve) ise SKPLI (Kapalı) ya dallan

```

```

; PTA[7-3] PTD[2-0] (9-16) Girişleri için

```

```

LG5     LDA    11t,X          ;A=(11+X) (Bağlı olduğu lojik giriş)
        AND    LOJG2          ;A=A AND LOJG2
        BEQ    KPIGRS         ;0 ise KPIGRS ye dallan(BURAYA YAZ)
LG6GCI  STA    AKTMP1I        ;AKTMP1I=A
        AND    PTLGD          ;A=A AND PTLGD
        BEQ    LG5b           ;Hepsi 0 ise LG5b ye dallan
        BRSET 2,ISLTMP,SSACIK3 ;1 (Veya) ise SACIK (AÇIK) a dallan
        CMP    AKTMP1I        ;A yı AKTMP1I ile karşılaştır
        BEQ    KPIGRS         ;0 ise KPIGRS ye dallan(BURAYA YAZ)
LG5b    BRCLR 2,ISLTMP,SSKPLI3 ;0 (ve) ise SKPLI (Kapalı) ya dallan
        BRA    KPIGRS         ;0 ise KPIGRS ye dallan(BURAYA YAZ)

```

```

; Veya iken Hepsi 0 VE iken hepsi 1    Bağlı değil

```

```

; Kapı Girişini Değerlendirir

```

```

; (1-8) Girişler

```

```

KPIGRS  CLR    RLCKTP        ;RLCKTPI=00
        MOV    #$04,DONGUI    ;DONGUI=04
        LDA    12t,X          ;A=Bağlı olduğu lojik ve kapı girişleri (1-8)
        AND    KAPIG1         ;Bağlı olduğu kapı girişleri (1-8)
        BEQ    LKG8h          ;Kapı girişi (1-8) e bağlı değil ise LKG8h ye

```

```

dallan

```

```

        CLR    CLRH           ;H=00
LKG22   INC    RLCKTP         ;RLCKTP=RLCKTP+1    (TEMP)
        ROLA                   ;A yı sola çevir
        STA    AKTMP1I        ;AKTMP1I

```

```

BCS    LKG11          ;C=1 is LKG1 e dallan
LKG11b LDA    AKTMP1I  ;A=AKTMP1I
        ROLA          ;A yı sola çevir
        DBNZ DONGUI,LKG22 ;DONGUI-1=0 değil ise LKG2 ye dallan
LKG8h  BRSET 2,ISLTMP,SKPLI3 ;1 (veya) ise SKPLI3 (Kplı) ya veya da
hepsi 0 BRCLR 2,ISLTMP,SACIK3 ;0 (ve) ise SACIK (AÇIK) a dallan ve de
hepsi 1
LKG11  LDX    RLCKTP    ;X=RLCKTPI
        LDA    $4F,X    ;A=(4F+X) (İçerideki Kişi Sayısı)
        BEQ    LKG12    ;0 ise LKG12 ye dallan
        BRSET 2,ISLTMP,SACIK3 ;1 (veya) ise SACIK (AÇIK) a dallan
        BRA    LKG11b   ;LKG1b ye dallan
LKG12  BRCLR 2,ISLTMP,SKPLI3 ;0 (ve) ise SKPLI (Kapalı) ya dallan
        BRA    LKG11b   ;LKG1b ye dallan

LKG3
SACIK3 LDA    HXTPLI    ;A=HXTPLI
        AND    #$F0     ;A=A AND F0h
        STA    HXTPLI    ;HXTPLI=A(Programın başlangıç adrsi)
        BRSET 4,ISLTMP,SACIK ;1 (veya) ise SACIK (AÇIK) a dallan
        BRA    SATBAK    ;SATBAK a dallan (Devamet)

SKPLI3 LDA    HXTPLI    ;A=HXTPLI
        AND    #$F0     ;A=A AND F0h
        STA    HXTPLI    ;HXTPLI=A(Programın başlangıç adrsi)
        BRCLR 4,ISLTMP,SKPLI ;0 (ve) ise SKPLI (Kapalı) ya dallan

; SAAT e Göre Çıkışın AÇIK veya KAPALI olduğunu Bulur
;
; AIX    #-9T          ;H:X=H:X-9
SATBAK LDHX    HXTPHI    ;H:X=HXTPHI:HXTPLI
        AIX    #7T      ;H:X=H:X+7
        LDA    ,X        ;A=(X)
        CMP    $02,X     ;A yı (X+2) ile karşılaştır
        BHI    LSA2      ;A>(X+2) ise LAS2 ye dallan
        BLO    LSA3      ;A<(X+2) ise LAS3 e dallan
;A=(X+2) ise yani Aşık ve Kapalı Saatlerinin Saat kısmı eşit ise
        LDA    $01,X     ;A=(X+1)
        CMP    $03,X     ;Daikikaları karşılaştır
        BHI    LSA2      ;A>(X+2) ise LAS2 ye dallan
        BLO    LSA3      ;A<(X+2) ise LAS3 e dallan
; Açık ve Kapalı Saatleri Birbirine Eşit ise
        BRA    SACIK      ;ACIK a dallan

LLTISA  JMP    LTISA      ;LLTISA ye dallan

; AÇIK KAPALIDAN KÜÇÜK İSE
; AÇIK değeri ile kontrol yapar
LSA3    LDA    ,X        ;A=(X) (saat)
        CMP    SAAT      ;A yı SAAT ile karşılaştır
        BHI    SKPLI2    ;A>SAAT ise KPLI e dallan          C=0
        BLO    LSA4      ;A<SAAT ise LSA4 ya dallan
        LDA    $01,X     ;A=SAAT ise A=dakika
        CMP    DAKIKA     ;A yı DAKIKA ile karşılaştır
        BHI    SKPLI2    ;A>DAKIKA ise KPLI ya dallan        C=0
        BLO    LSA4      ;A<DAKIKA ise LSA4 e dallan
        BRA    SACIK      ;Eşit ise ACIK a dallan          C=0
; KAPALI değeri ile kontrol yapar
LSA4    LDA    $02,X     ;A=(X+2) saat kapalı
        CMP    SAAT      ;A yı SAAT ile karşılaştır
        BHI    SACIK      ;A>SAAT ise ACIK a dallan          C=0
        BLO    SKPLI      ;A<SAAT ise KPLI a dallan          C=1
; saatler aynı. dakikayı kontrol eder (KAPALI için)

```

```

LDA    $03,X          ;A=(X+3) saat kapalı
CMP    DAKIKA          ;A yı DAKIKA ile karşılaştır
BHI    SACIK           ;A>DAKIKA ise ACIK a dallan      C=0
BRA    SKPLI           ;A<DAKIKA ise KPLI ya dallan

; AÇIK KAPALIDAN BÜYÜK İSE
; KAPALI değeri ile kontrol yapar
LSA2   LDA    $02,X    ;A=(X)      (saat)
CMP    SAAT           ;A yı SAAT ile karşılaştır
BHI    SACIK           ;A>SAAT ise ACIK a dallan      C=0
BLO    LSA4b          ;A<SAAT ise LSA4b ye dallan     C=1
LDA    $03,X          ;A=SAAT ise A=dakika
CMP    DAKIKA          ;A yı DAKIKA ile karşılaştır
BHI    SACIK           ;A>DAKIKA ise ACIK a dallan      C=0
BLO    LSA4b          ;A<DAKIKA ise LSA4b ye dallan     C=1
BRA    SKPLI2         ;Eşit ise SKPLI ye dallan      C=0

; AÇIK değeri ile kontrol yapar
LSA4b  LDA    ,X       ;A=(X)      (saat)
CMP    SAAT           ;A yı SAAT ile karşılaştır
BHI    SKPLI2         ;A>SAAT ise ACIK a dallan      C=0
BLO    SACIK2         ;A<SAAT ise KPLI ya dallan     C=1

; saatler aynı. dakikayı kontrol eder (AÇIK için)
LDA    $01,X          ;A=SAAT ise A=dakika
CMP    DAKIKA          ;A yı DAKIKA ile karşılaştır
BHI    SKPLI2         ;A>SAAT ise ACIK a dallan      C=0
; BRA    ACIK          ;A>DAKIKA ise ACIK ya dallan

SACIK  SEC           ;C=1
SACIK2 ROL    PRGACK   ;PRGACK yi sola döndür
BRA    LSA5          ;LAS5 e Dallan

SKPLI  CLC           ;C=0
SKPLI2 ROL    PRGACK   ;PRGACK yi sola döndür
LSA5   LDA    KCITI    ;A=KCITI
AND    #$07           ;A=A AND 0Fh
BEQ    LPA5d          ;Eşit ise LPA5d ye dallan
LDA    HXTPLI         ;A=HXTPLI
AND    #$F0           ;A=A AND F0h
ADD    #$10           ;A=A+$10h
STA    HXTPLI         ;HXTPLI=A(Programın başlangıç adrsi)
BRA    LLTISA         ;Değil ise LTISA ya dallan

; Programlar Arası Çıkışı Değerlendirir ve Çıkış Gücü RAMına yazar
LPA5d  LDA    HXTPLI   ;A=HXTPLI
AND    #$80           ;A=A And 80
TAX
STA    HXTPLI         ;HXTPLI=A(Çıkış prog başlangıç adrsi)
MOV    #$07,DONGUI    ;DONGUI=08
MOV    DONGUI,RLCKTP  ;LCHDTP=DONGUI
LPA5   LDA    $0D,X    ;A=(X+13d) Lojik işlem
AND    #00000011Q     ;A AND 00000011b
AIX    #$10           ;H:X=H:X+10 ikinci prog baş adr
BEQ    LPA1           ;A=0 ise LAP1 e dallan

; Bir Sonraki program ile OR işlemi yapıkacak ise
LDA    PRGACK         ;A=PRGACK
ASLA   ;A yı sola Shiftet
BCC    LPA2b          ;C=0 ise LPA2b ye dallan
ASLA   ;A yı sola Shiftet
BCS    LPA2c          ;C=1 ise LPA2C ye dallan
SEC    ;C=1
RORA   ;A yı sağa döndür
STA    PRGACK         ;PRGACK=A
BRA    LPA3           ;LPA3 e dallan
LPA2c  RORA          ;A yı sağa döndür

```

```

LPA2b  STA  PRGACK      ;PRGACK=A
      LDA  DONGUI      ;A=DONGUI
      DECA      ;A=A-1
      STA  RLCKTP      ;RLCKTP=A
      BRA  LPA3        ;LPA3 e dallan
; Bir Sonraki program ile AND işlemi yapılacak ise
LPA1   LDA  PRGACK      ;A=PRGACK
      ASLA      ;A yı sola Shiftet
      BCS  LPA4        ;C=1 ise LPA4 ye dallan
      AND  #$7F        ;A[7]=0
LPA4   STA  PRGACK      ;PRGACK=A
      LDA  DONGUI      ;A=DONGUI
      DECA      ;A=A-1
      STA  RLCKTP      ;RLCKTP=A
LPA3   DBNZ  DONGUI,LPA5 ;DONGUI-1=0 değil ise LPA5 e dallan

```

; Çıkışın Çalışan Programına Göre Gücü Güç data RAMına Yazar

; Güç Çalışan en son programın gücü

```

      LDHX  HXTPHI      ;X=Flash Çıkış Veri Baş Adr
      LDA  #$08         ;A=08
      SUB  RLCKTP      ;A=A-LCHDTP (Kaçınıcı programın gücü)
;      JSR  LRB3        ;LRB3 alt prog dallan
LRB3I  DECA      ;A=A-1
      BEQ  LRB2I        ;A=0 ise LRB2 ye dallan
      AIX  #$10         ;H:X=H:X+10h
      BRA  LRB3I        ;LRB3 e dallan
LRB2I  LDA  $03,X       ;A=Programın Çıkış Gücü
      CLRH      ;H=0
      LDX  KCITI        ;X=KCITI
      LSRX      ;X i sola shiftet 2 ye döl
      LSRX      ;X i sola shiftet 2 ye döl
LPA6b  LSRX      ;X i sola shiftet 2 ye döl
      TST  PRGACK      ;PRGACK 0 mı
      BNE  LPA6        ;Değil ise LPA6 ya dallan
      CLRA      ;A=0
      BRA  LPA7        ;LPA7 ye dallan
LPA6   COMA      ;A nın tersini al
      INCA      ;A=A+1
      BEQ  LPA7        ;A=0 ise LPA7 ye dallan
      ADD  #50T        ;A=A+100
LPA7   STA  $0180,X     ;A yı Çıkış Gücü RAM Datasına yaz
      LDA  KCITI        ;A=KCITI
      CBEQA #$80,LPA8   ;A=80 ise LPA8 e dallan
;      CBEQA #64T,LPA8   ;A=64d ise LPA8 e dallan
      LDA  HXTPLI      ;A=HXTPLI
      BNE  LPA9        ;A=0 değil ise LPA9 a dallan
      MOV  #$80,HXTPLI ;HXTPLI=80h
      BRA  LTISA       ;LTISA ye dallan
LPA9   INC  HXTPHI      ;HXTPHI=HXTPHI+1
      BRA  LPA10       ;LPA10 a dallan
LPA8   MOV  #$A0,HXTPHI ;HXTPHI=A0h
      CLR  KCITI        ;KCITI=00
LPA10  CLR  HXTPLI      ;HXTPLI=00

```

* SAATi Çalıştırır ve LCD ye yazar

```

LTISA  LDA  DAKIKA0     ;Dakika için timer int. sayımı
      ADD  #$01
      STA  DAKIKA0
      LDA  DAKIKA1
      ADC  #$00
      STA  DAKIKA1
      LDA  DAKIKA2
      ADC  #$00

```

```

STA    DAKIKA2
LDA    SANIYE0      ;Saniye için timer int. sayımı
ADD    #$01
STA    SANIYE0
LDA    SANIYE1
ADC    #$00
STA    SANIYE1
CBEQA  #SANIYE1D,L01 ;1 Saniye oldu mu
BRA    L02          ;Olmadıysa L02 ye dallan
L01    LDA    SANIYE0
        CMP    #SANIYE0D
        BNE    L02
; Bir Saniye Oldu
        BSET   4,BAYRAK0      ;BAYRAK0[4]=1
        LDA    SANIYE
        INCA
        DAA
        CMP    #$60          ;60 Oldu mu
        BEQ    L06          ;Oldu ise L06 ya dallan
        STA    SANIYE        ;1 Saniye oldu
        CLR    SANIYE0
        CLR    SANIYE1
L02    LDA    DAKIKA2        ;1 Dakika oldu mu
        CBEQA  #DAKIKA2D,L03
        BRA    L04          ;Olmadıysa L04 e dallan
L03    LDA    DAKIKA1
        CBEQA  #DAKIKA1D,L05
        BRA    L04          ;Olmadıysa L04 e dallan
L05    LDA    DAKIKA0
        CMP    #DAKIKA0D
        BNE    L04          ;Olmadıysa L04 e dallan
        BSET   4,BAYRAK0      ;BAYRAK0[4]=1
L06    LDA    DAKIKA
        INCA
        DAA                ;1 Dakika oldu
        CMP    #$60          ;60 Oldu mu
        BNE    L08          ;Olmadı ise L08 e dallan
        LDA    SAAT
        INCA
        DAA
        CMP    #$24          ;24 Oldu mu
        BNE    L09          ;Olmadı ise L09 a dallan
        LDA    GUN
        INCA
        CMP    #$08
        BNE    L07          ;GUN=8 değilse L07 ye dallan
        LDA    #$01          ;Pazartesi
L07    SEI
        STA    GUN
        CLRA
L09    SEI
        STA    SAAT          ;1 Saat oldu
        CLRA                ;A=0
L08    STA    DAKIKA        ;Dakikayı rama yaz
        CLI
;        CLRA                ;A=0
        CLR    DAKIKA0
        CLR    DAKIKA1
        CLR    DAKIKA2
        CLR    SANIYE0
        CLR    SANIYE1
        CLR    SANIYE
L04    PULH                ;Hyi RAMdan çek

```

```

        BCLR    2,BAYRAKO      ;BAYRAK[0]=0 TIM2INT den çıktı
        RTI

WLCDRAM PSHA
        AND     #$0F
        ADD     #$30
        STA     ,X
        DECX
        PULA
        NSA
        AND     #$0F
        CMP     #$0F
        BEQ     LWL1
        ADD     #$30
        STA     ,X
        RTS

LWL1    LDA     #$20
        STA     ,X
        RTS

```

```

ORG     RESET
FDB     ROMSPACE      ;ROMun başına git
ORG     TIMOV
FDB     TIMINT        ;Timer int. adresine git
ORG     TIM2OV
FDB     TIM2INT       ;Timer2 int. adresine git
ORG     IRQ
FDB     IRQINT        ;IRQ int. adresine git
ORG     KEY
FDB     KEYINT        ;Keyboard int. adresine git
ORG     ADC
FDB     ADCINT        ;ADC int. adresine git
ORG     SWIV
FDB     TIMINT        ;Timer int. adresine git

```

ÖZGEÇMİŞ

Kişisel Bilgiler :

Doğum Yeri/Tarihi : İstanbul, 04.06.1977
 Sürücü Ehliyeti : B - Sınıfı
 Adres : Murat Cankat sok. No.:3/2, 34800 Yeşilköy - İstanbul
 Telefon - ev : (0-212) 663 58 61
 - iş : (0-212) 259 70 70 Dahili : 2890
 E-posta Adresi : kkayaer@rt.net.tr

Eğitim :

- 1988-1992 : Özel Çavuşoğlu Koleji
- 1992-1994 : Özel Üsküdar Fen Lisesi
- 1994-1995 : Özel Bilfen Lisesi
- 1995-1999 : Yıldız Teknik Üniversitesi
Elektronik ve Haberleşme Mühendisliği Bölümü
- 1999-2002 : Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Elektronik Programı, Yüksek Lisans,

İş Tecrübesi :

- Ağustos 97 – Eylül 97: ONTROL
AR-GE Bölümünde
Genel Staj
- Ağustos 98 – Eylül 98: ISIKO
Teknik Servis Bölümünde
Mesleki Staj
- Kasım 2000 - : Yıldız Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Araştırma Görevlisi

Yabancı Dil :

- İngilizce : Konuşma : İyi Yazma : İyi Anlama : İyi

Bitirme Projesi Konusu :

- Proje : 8032 Mikrodenetleyici Kontrollü Dijital AC/DC Voltmetre