

67808

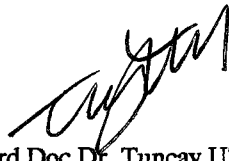
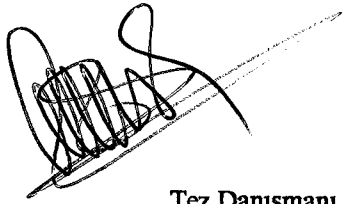
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MİKRODENETLEYİCİ SİSTEM İLE AĞIRLIK ÖLÇÜMÜ

Elektronik ve Hab. Müh. Yunus KARABÖREK

F.B.E. Elektronik ve Haberleşme Anabilim Dalında
hazırlanan

YÜKSEK LİSANS TEZİ



Tez Danışmanı : Yrd.Doc.Dr. Tuncay UZUN

İSTANBUL, 1997

İÇİNDEKİLER

TEŞEKKÜR	III
ÖZET	IV
ABSTRACT	V
1. GİRİŞ.....	1
2. GENEL BİLGİLER.....	3
2.1 GERGİNLİK- ÖLÇER	3
2.1.1 Yük-Hücresi	4
2.1.2 Gerginlik-Ölçerlerin Çalışma Prensipleri	6
2.2 ANALOG -SAYISAL DÖNÜŞTÜRÜCÜ	9
2.3 AĞIRLIK ÖLÇÜM SİSTEMİNDE KULLANILAN MİKRODENETLEYİCİ.....	20
2.3.1 8031 Mikrodenetleyicisi	20
2.3.2 8051 ve 8751'in Özellikleri.....	21
2.3.3 8051 Komut Biçimleri	22
3. MİKRODENETLEYİCİ SİSTEM İLE AĞIRLIK ÖLÇÜMÜ.....	24
3.1 AĞIRLIK ÖLÇÜM SİSTEMİNİN DONANIMI	25
3.1.1 Besleme Yapısı	26
3.1.2 Mikrodenetleyici ile Ölçüm Kısımı.....	26
3.1.3 Kalibrasyon Belleği ile Haberleşme Kısımı.....	28
3.1.4 Ağırlık Ölçüm Sisteminin Gösterge Bölümü.....	30
3.1.5 Ağırlık Ölçüm Sisteminin Tuş Takımı Kısımı.....	32
3.2 AĞIRLIĞIN ÖLÇÜLMESİ VE DEĞERLENDİRİLMESİ	35
3.2.1 ADC Okuma-Yazma İşlemi.....	37
3.3 AĞIRLIK ÖLÇÜM SİSTEMİNİN YAZILIMI.....	42
4. SONUÇ	45
5. KAYNAKLAR	47
6. EKLER	48
EK1.YAZILIM AKIŞ DİYAGRAMLARI	49
EK2.BESLEME KISMI ELEKTRONİK DEVRE ŞEMASI	60
EK3.ANOLG KISMI ELEKTRONİK DEVRE ŞEMASI.....	61
EK4.MİKRODENETLEYİCİ KISMI ELEKTRONİK DEVRE ŞEMASI.....	62
EK5.EEPROM KISMI ELEKTRONİK DEVRE ŞEMASI	63
EK6.GÖRÜNTÜ KISMI ELEKTRONİK DEVRE ŞEMASI	64

EK7.AĞIRLIK ÖLÇÜMÜ YAZILIMI.....	65
---	-----------



TEŞEKKÜR

Bu çalışmamda, beni yönlendiren ve destek olan değerli hocam Yrd.Doç.Dr.Tuncay UZUN'a, yardımlarından dolayı başta Arş.Gör Beşir TAYFUR ve Elektronik ve Haberleşme Mühendisi Turgut GÜVERCİN olmak üzere bütün arkadaşlarıma ve benim bu günlere ulaşmamı sağlayan aileme teşekkürü bir borç bilirim.

Yunus KARABÖREK

Ekim, 1997

İstanbul



ÖZET

Günümüzde mikroişlemciler ve mikrodnetleyiciler endüstrinin her dalında kullanılarak teknolojiye çok önemli bir noktaya ulaşılmıştır. Bu çalışmada, mikroişlemcilere oranla çok daha az çevre donanımına ihtiyaç duyan mikrodnetleyici tümdevresini kullanarak, günümüzde sanayinin her alanında, otomasyon sistemlerinde, ticari kuruluşlarda veya diğer alanlarda kullanılabilecek ağırlık ölçüm sistemi gerçekleştirilmiştir.

Tasarlanan sistemde dönüştürücü algılayıcı olarak gerginlik-ölçer (strain- gage) ailesinden Yük- Hücresi (Load -Cell) kullanılmıştır. Bu sistemde kullanılan dönüştürücü doğrusal değişim gösteren bir karakteristiğe sahiptir. Kalibrasyon sistemi ek yazılım kullanılarak tuş takımı ile gerçekleştirilmiştir. Bu nedenle tasarlanan bu sistem, sadece ağırlık ölçen bir sistem olarak değil, doğrusal bir değişime sahip bütün dönüştürücülerden gelen bilgiyi ölçebilecek kabiliyete sahiptir.

Bu sistemde analog sayısal dönüştürücü olarak en son ürünlerden biri olan 16-bit ADC tümdevresi kullanılmıştır. Bu dönüştürücünün en büyük özelliği, algılayıcılardan gelen işareti, her hangi bir çevre donanımına gereksinim duymadan direkt olarak kabul etmesidir. Böylece çevre elemanları asgari düzeye indirilmek suretiyle, gerek maliyetin düşmesi ve gerekse sistemin biraz daha donanım açısından küçülmesi ile, beraberinde birçok problemde ortadan kalkmış olacaktır.

Bu sistemde kalibrasyon bilgisini saklamak amacıyla mikrodnetleyiciye bağlanan bir hafıza birimi olan EEPROM tümdevresi kullanılmıştır. Bu nedenle sistemde ayrıca seri haberleşmede gerçekleştirilmiştir.

Gerçekleştirilen ağırlık ölçme sistemi, 5gr ile yük hücresinin özelliğine bağlı olarak 30kg ağırlığı 5gr duyarlılık ile ölçmeye programlanabilir ve sayısal LED gösterge üzerinde gösterebilir. Ayrıca kullanılan mikrodnetleyici temelli sayısal sistemin üstünlükleri nedeniyle, sistem endüstride ihtiyaç duyulan diğer ölçümlerde kullanılabilir.

ABSTRACT

Today, microprocessors and microcontrollers have reached very important point with wide use in various industrial applications. In this study, by using of microcontroller IC, since it requires fewer hardware comparing to microprocessors, the weight measurement system, which can be used today's industrial, automation systems, traditional or other application.

Load-Cell from Strain-Gage family used as a transducers sensor in this realization. transducers, which was used in this system, has a linear changeable characteristic. By using additional software, the calibration system realized with keyboard. That is why, the system which is mentioned above, measures not only weight, but also the data which come from all the transducers, which has a linear characteristic.

In this system a 16-Bit ADC IC was used as a new analogue to digital converter. The ADC can accept a signal coming any peripherals and this is its important feature. As the number of the peripherals was optimized to its minimum, following the increase at the system price and hardware complexity, lost of problem could be eliminated.

In this system, an EEPROM IC which is connected to microcontroller was used to save the calibration data. Because of that, a serial communication interface was also applied in this system.

This designed weight measurement system can be programmed to measure 30 kg weight with 5 gr. Sensibility by using 5 gr. depending on the Load-Cell and it can show the result with a LED display. In addition, because of the features of the microcontroller based digital system, it can be used for any other measurements using industrial application.

1. GİRİŞ

Teknolojinin büyük bir hızla ilerlediği günümüzde bu gelişimin etkileri endüstride de görülmektedir. Mikrodenetleyiciler, bu ilerlemenin sonucu olarak ortaya çıkmış ve çok hızlı bir ilerleme göstermiştir. Maliyetlerinin gittikçe azalması ve özelliklerinin artması nedeniyle mikroişlemcilerin yerini alarak, endüstride tercih edilir hale gelmiştir.

Bu çalışmada, ağırlık dönüştürücüsünden gelen bilginin değerlendirilmesine yönelik çalışma yapılmıştır. Gıda, kimya ve birçok sanayi sistemlerinde ağırlık ölçümüne gereksinim vardır.

Gerçeklenen bu sistemde, 87C51 mikrodenetleyicisi kullanılmıştır. Ağırlık ölçüm devresinde dönüştürücü olarak Yük-Hücresi (Load -Cell) kullanılmıştır. Ayrıca EEPROM olarak 24C16 entegresi kullanılmıştır. Ağırlık ölçüm devresinde Yük- Hücresi (Load -Cell)'nin 2 ve 3 numaralı uçlarında ağırlıkla doğrusal olarak değişen ve seviyesi mV'lar düzeyinde olan bir gerilim değişimi meydana gelmektedir. Meydana gelen bu değişim, AD 7715 analog sayısal dönüştürücünün girişine uygulanmaktadır. AD 7715 ve buna bağlı olan analog kısımda, dönüştürücüde oluşan gerilim değişimi sayısal bir işarete dönüştürülür. Daha sonra sayısal hale çevrilmiş olan işaret, mikrodenetleyicinin giriş portuna uygulanır. Mikrodenetleyici kısmında arda arda on değer okunur. En büyük ve en küçük değer atılır. Kalan sekiz değerın ortalaması alınır. Böylece gürültülü ve titreşimli ortamlarda da çalışması sağlanmış olmaktadır. Daha sonra bu sonuç değer tuş takımıyla kalibrasyona tabii tutulur. Değerlendirilen değer göstergeye aktarılır.

Kalibrasyon işlemi 4x4lük tuş takımı yardımıyla yapılır. Sistem ilk olarak açıldığı zaman tuş takımı ile kalibrasyon şifresi girilir. (Örnek olarak gerçekleştirilen devrede kalibrasyon şifresi 2357dir). Şifre girildikten sonra sistem kalibrasyon programına daldanmaktadır.

Kalibrasyon için ağırlığı önceden bilinen bir parca tartılır. Daha sonra göstergeye aktarılan değer görülür. Sonra, ağırlığını daha önce bildiğimiz değer tuş takımıyla girilir ve bir başka tuş olan hafıza tuşuna basılır. Böylece daha sonra gelecek olan ağırlık değerleri bu referans değeri doğrultusunda orantıya tabii tutulur. Böylece sistem kalibre

edilmiş olur. Daha sonraki reset durumlarında sistem önceden kalibre edildiği için sürekli olarak gerçek ağırlığı ölçmeye devam edecektir.

Tasarlanarak gerçekleştirilen sistemde gösterge kısmında, 1 adet 74HC138, 2 adet 74 HC574, 1adet ULN2803, 1adet ULN 2981 entegreleri kullanılmıştır. Mikrodenetleyicinin 0 no'lu portu hem tuş takımı ve hemde gösterge ile ilgili bilgileri iletme amacıyla kullanılması nedeniyle sistemde 1 adet 74HC574 Tutucu (Flip-Flop) kullanılarak, bu iki ayrı bilgi arasında istenileni iletecektir.



2. GENEL BİLGİLER

2.1 GERGİNLİK- ÖLÇER

1930lu yılların sonlarına doğru Gerginlik-Ölçer (Strain -Gage)'in bulunuşundan sonraki yıllar boyunca Gerginlik-Ölçer dönüştürücüler çok önemli bir mühendislik aracı olarak kullanılmaya başlanmıştır. Daha sonraki on yıl içerisinde Gerginlik-Ölçer dönüştürücü uygulamaları, dikkate değer bir oranda gelişme kaydetmiştir. Sonuç olarak, bir çok endüstriyel uygulamalarda ve mühendislik kullanımında tüketici ürünü olarak ve ticari olarak kullanılır hale gelmiştir. Kısaca dönüştürücü teknolojisinde, belirgin bir büyüklükte iş olanağı doğmuştur.

Geniş anlamda bir dönüştürücü, bir enerji türünü bir başka enerji türüne dönüştüren devredir. Mesela akümülatörde kimyasal enerji elektriksel enerjiye dönüşmektedir. Bu uygulamada mekanik olarak meydana gelen değişimleri ölçüm ve kontrol sistemlerinde kullanmak için elektriksel enerjiye çeviren Yük- Hücresi (Load-Cell) kullanılmıştır. Yük-Hücresi yaygın olarak basınç, ağırlık, hız ve dönme momenti gibi uygulamalarda kullanılan Strain -Gage ailesinden bir dönüştürücüdür. Hücre (Cell) kelimesi dönüştürücüyü tanımlamak için sık sık kullanılan bir kelimedir. Load-Cell, Pressure-Cell dönüştürücülerinde olduğu gibi.

Dönüştürücüler farklı çalışma prensiplerine sahiplerdir ve bu farklı prensipler doğrultusunda üretilirler. Rezistif, endüktif, kapasitif, piezoelektrik olarak üretilebilirler. Örnek olarak minyatür hız ölçer, ağırlık ölçer gibi uygulamalarda kullanılmaktadır. Ayrıca uygulama ihtiyaçlarına uygun olarak, kapasitif veya endüktif olarakta tasarlanabilmektedir.

Gerginlik-Ölçer (Strain Gage) dönüştürücüler ile ilgili olarak Lord Kelvin 1800lı yıllarda elektriksel bağlantıların dirençlerinin değişimleri ile ilgili olarak bir çalışma yapmıştır. 1908de ise Lindec of Charlottenburg Almanyada Gerginlik-Ölçer (Strain-Gage) basınç dönüştürücüler üzerine bir çalışma yapmış ve bu konuda birçok probleme çözüm yolları geliştirmiştir. Günümüzde makinaların, binaların, uçakların en önemli yapısal bölümlerinde kullanıma sahiptir.

2.1.1 Yük-Hücre

Yük-Hücreleri (Load-Cell) en sık kullanılan dönüştürücülerdendir. Yük-Hücrelerinin tasarlanması ile birlikte daha sonraki aşamalarda ticari hayata geçirilmiştir. Uygulamaların en temel olanı proses kontrolünde, ağır makinalarda, test mühendisliğinde ve benzeri alanlarda kullanılmasıdır. Günümüzde Gerginlik-Ölçerler ağırlık ölçüm alanlarında sıkça kullanılmaktadır. Terazilere olan ihtiyacın ortaya çıkması ile sayısal göstergeler için gerekli olan elektrikselsinyali üretmesi önemini biraz daha artırmıştır. Mikroişlemciler ve bilgisayarlar ile arabirim sağlamaktadır. Gerginlik- Ölçer ailesinden Yük-Hücre en pratik olarak ağırlıkları göstermek için düşünülmüştür. Elektronik teraziler şu anda basit işlevli sıradan Yük-Hücre kullanılarak gerçekleştirilmektedir. En büyük kullanım alanlarından biri ise perakende satış sektörüdür. Ayrıca postanelerde, laboratuvarlarda, gemilerde ve diğer yerlerde kullanılmaktadır. Herhangi bir Yük-Hücrede en kırıltık mekanik malzeme, diğer dönüştürücülerde olduđu gibi yaylanma (spring) elementidir. Spring elementler uygulanan yüklerin bir tepkimesi olarak hizmet etmektedir.

Dönüştürücülerde (Transducers) ideal durum uygulanan yük ile gerginlik (strain) arasındaki ilişkinin doğrusal bir karakteristik gösteriyor olmasıdır. Ancak bu durum bütün dönüştürücüler için geçerli değildir. Gerginlik-Ölçerleri genel olarak yedi ayrı tipte incelemek mümkündür. Şimdi bunları kısaca genel hatları ile inceleyelim.

2.1.1.1 Metal Gerginlik-Ölçer

1930 ların son yarısında Charles Wheatstone tarafından yapılan çalışma sonunda bulunmuştur. Gerginlik sonucu bir telin üzerindeki direncin çok küçük değişimi sonucunu ortaya koymuştur. Daha sonra ise William Thomson ve Lord Kelvin bu konuda yapılan çalışmalara katkıda bulunmuşlardır. Bu gerginlik ölçerin başlıca kullanım alanları:

- Deneysel gerginlik analizi,
- Ölçüm teknikleri,
- Biomedical uygulamalar,
- Dönüştürücü üretimi

2.1.1.2 Yarıiletken Gerginlik- Ölçer

Metal Gerginlik -Ölçerler dışında elektrikselsel Gerginlik-Ölçerlerden olan ve bu ailenin diğeri tipi olan Gerginlik-Ölçerlerdir. Yarıiletken Gerginlik-Ölçerler büyük bir uygulama alanına sahiptir. Milimetrenin onda biri kadarlık değışimleri sezebilirler. Bu Gerginlik-Ölçerlerin özellikleri ise;

- Yarı iletkenlerin (semikonduktor) lineer olmayan karakteristiğıe sahip olanları çok daha yüksek doğrulukta ölçüm gerektirir.
- Yarıiletken Gerginlik-Ölçerler, Metal Gerginlik-Ölçerlere göre daha pahalıdır.
- Herhangi bir şekilde ellemek suretiyle doğal durumunu yitirmesi oldukça zordur.

2.1.1.3 Thin Film Gerginlik- Ölçer

Elektrikselsel Rezistif Gerginlik-Ölçerlerin bir üçüncü tipidir.

2.1.1.4 Kapasitif Gerginlik-Ölçer

Metal Gerginlik-Ölçerlerin belli bir sınırının ötesinde, yüksek sıcaklıklarda ölçüm yapmak için geliştirilmiş olan alternatif bir sistemdir. Kullanılan değışik versiyonları bulunmaktadır.

2.1.1.5 Piezoelektrik Gerginlik-Ölçer

Piezoelektrik gerginlik-ölçerler aktif devrelerdir. Gerginlik algılama elemanı olarak Baryum Titanat kullanılmaktadır. Algılama elemanı kadar , piezoelektrik dönüştürücü gerginlik ölçer yüzeyinde elektrikselsel olarak şarj olayı oluşturmaktadır.

2.1.2 Gerginlik-Ölçerlerin Çalışma Prensipleri

Gerginlik ölçümünde kullanılan Gerginlik-Ölçerler o nesne üzerindeki gerginliği araştırırken oluşan büyüklüğün kayıpsız olarak iletimini sağlamaya çalışır. Bu ise nesne ile Gerginlik-Ölçer arasında kapalı bir bağ gerektirmektedir. Hemen hemen bütün durumlarda, ölçülen objenin açık yüzeyi ölçüm için erişilebilirdir. Gerginlik-Ölçerlerin en önemli elemanları olan metal ve yarıiletken Gerginlik-Ölçerler ile ilgili çalışma koşullarını açıklayalım.

Metal Strain Gage

Whatstone ve Thomson tarafından keşfedilen elektrikselsel iletkenlerin dirençleri arasındaki ilişki ve gerginlikleri arasındaki ilişki, Metal Gerginlik-Ölçerlerin çalışma prensibini oluşturmaktadır. Mekanik gerilme sonucunda herhangi bir iletkenin direncinin değişimi burada temel mantığı oluşturmaktadır. Direnç değişimi kısmen iletkenin bozulması ve kısmende mikroyapısal değişimin bir sonucu olarak iletken metaryalin δ direncinin değişimidir. Bu çalışma aşağıdaki ilişki gibi tanımlanır.

$$\frac{dR}{R} = \epsilon \cdot (1 + 2 \cdot \nu) \cdot \frac{d\delta}{\delta} \quad (2. 1)$$

Geometrik kısım

R=Elektrikselsel rezistans

ϵ =Strain(gerginlik)

ν =Poisson oranı

δ =direnç

Yarıiletken Gerginlik-Ölçer

Yarıiletken Gerginlik-Ölçer içinde Metal Gerginlik-Ölçerlerde olduğu gibi direncin değişimi sonucu olarak, ayrıca sıcaklık faktöründe aynı şekilde hesaba katılması ile aşağıdaki denklem elde edilmektedir.

$$\frac{dR}{R} = \text{dirençteki değişim}$$

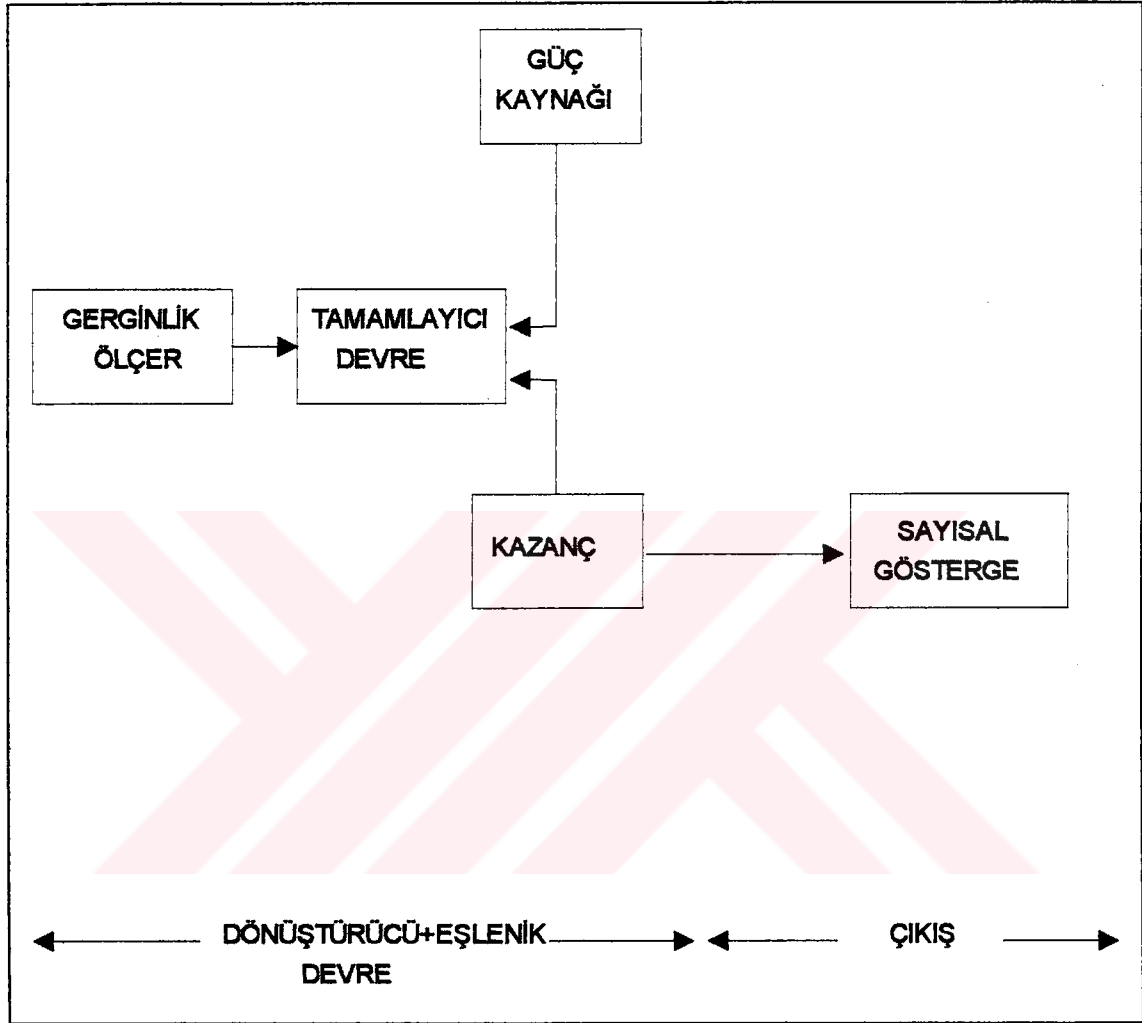
(2. 2)

$$\frac{dR}{R} = k \cdot \varepsilon \cdot \frac{T_0}{T} + c \cdot \varepsilon^2 \cdot \left(\frac{T_0}{T}\right)^2$$

değişimi elde edilir

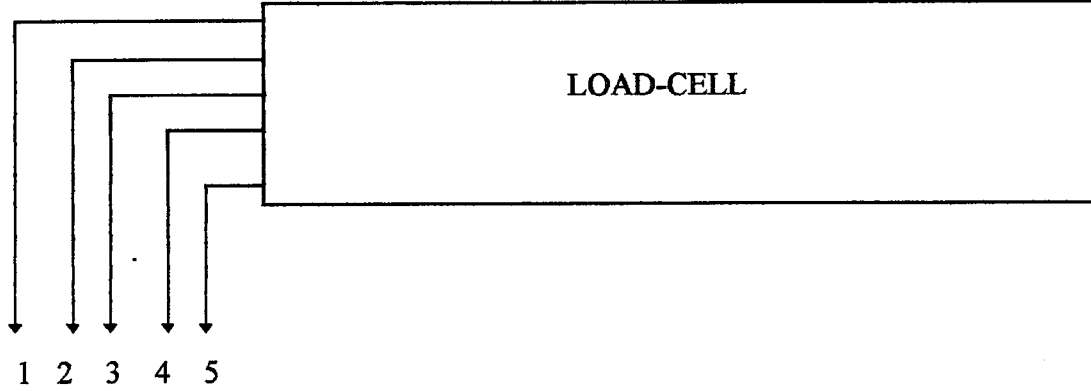
Yük-Hücrelerinden gerçekleştirilen ölçüm sistemi aşağıda verilmektedir.

Ölçüm Sistemi



Şekil 2. 1 Bir Yük-Hücre ölçüm sistemi

YÜK HÜCRESİ BAĞLANTI ŞEMASI



Şekil 2.2 Yük-Hücresi bağlantı şeması

Load-Cell bağlantı şekilleri aşağıdaki gibidir.

Kırmızı	⇒	1	GND
Beyaz	⇒	2	AIN(-)
Sarı	⇒	3	AIN(+)
Siyah	⇒	4	VCC
Şase	⇒	5	GND

2.2 ANALOG -SAYISAL DÖNÜŞTÜRÜCÜ

Günümüz teknolojisinde sayısal ölçme tekniği endüstride yaygın olarak kullanılmaya başlanmıştır. Bunun nedenlerinden biri üretimde otomasyona gidilmesidir. Bugün üretim, modern kuruluşlarda bilgisayarlarla kontrol edilmektedir. Örneğin bir kazanın ısı

ayarını veya bir sistemin uyguladığı ağırlık, uygun sensörler kullanılarak elde edilen bilgilerin değerlendirilmesiyle bilgisayar yardımıyla yapılmaktadır. Sensörün verdiği gerilim analog olarak bilgisayara verildiğinde hat boyunca oluşabilecek gerilim düşümlerini de dikkata alırsak sakıncalıdır. Bununla beraber bilgilerin bilgisayara sayısal olarak aktarılması gerekmektedir. Sayısal bilgilerin iletimi analog bilgilerin iletiminden daha küçük hatayla gerçekleşmektedir.

Bütün bu nedenlerden dolayı gerilim veya akımlar analog-sayısal çeviriciler yardımıyla sayısal duruma getirilirler. Proje çalışmalarında kullanılmak üzere yapılan çalışmalar doğrultusunda Analog Devices Firmasının üretmiş olduğu en son ürünlerinden biri olan AD 7715 Analog- Sayısal Dönüştürücüsü kullanılmıştır.

AD 7715 ANALOG SAYISAL DÖNÜŞTÜRÜCÜSÜNÜN ÖZELLİKLERİ

- 16 Bit eksiksiz kodlama
- 0. 0015 % lineer olmama oranı
- Programlanabilir kazanç kontrolü: 1, 2, 32, 128 gibi farklı kazançlarda programlanabilme
- Farksal giriş kabiliyeti
- 3 bağlı seri arayüz
- 3V, 5V gerilimlerde çalışma
- Düşük kaynak akımı: 450 μ A
- Programlanabilir Alçak Geçiren Filtre
- 16- Pin SOIC/DIP

AD7715, alçak frekans ölçüm uygulamaları için kullanılan ve girişine analog işaretin kendisinin uygulandığı bir Analog - Sayısal çeviricidir. Bu kısım bir sensörden direk olarak düşük seviyeli giriş sinyallerini kabul etmekte ve seri bir sayısal kelime çıkışı üretmektedir. AD7715, 16 bit eksiksiz kodlama performansına sahiptir.

Giriş sinyali, yapısında bulunan analog modülatörden önce programlanabilen kazanç kontrolü kısmına uygulanmaktadır. Modülatör çıkışı ise bir sayısal filtre yongası

tarafından işlenmektedir. Bu sayısal filtrenin kesim frekansı entegre üzerindeki kontrol yazmacındaki bitlerle belirlenir.

AD 7715, farksal referans girişleri kadar iyi bir farksal analog giriş özelliğine sahiptir. AD 7715, +3 Volt veya +5 Volt gerilimlerde çalışabilir. AD7715 (0mV, -20mV), (0mV, +80mV), (0V, -80mV), (0V, +1. 25V), (0V, -1. 25V), (0V, +2. 5V), (0V, -2. 5V) aralıkları içerisinde bulunan sinyal girişlerini kabul etmektedir.

Negatif değerdeki girişler farksal analog girişin negatif girişine uygulanmaktadır. AD 7715, sayısal işaret işleme ve mikrodenetleyici tabanlı sistemler için idealdir. Kazanç kontrolü yazılım kullanılarak belirlenmektedir.

AD 7715 3Volt ve 1MHz ana frekansında çalışırken toplam olarak 450 μ Aden daha az güç kaybı meydana gelmektedir. Bu nedenle düşük güç uygulamaları için idealdir. AD7715 bir Gerginlik- Ölçer (Strain- Gage)den veya herhangi bir sensörden direkt olarak gelen sinyali kabul etmektedir. AD7715, DSP ve mikrodenetleyiciler için uygundur. AD7715 için analog besleme voltajı ; (+5Volt), sayısal besleme voltajı +3Volt veya (+5Volt) tur. Pozitif referans gerilimi (+2. 5 Volt), negatif referans gerilim girişi devrenin analog kısmına ait olan toprağa bağlanır. AD7715in salınım frekansı (Osilatör frekansı) 1MHz veya 2. 4576 MHz olabilir. Çalışma sıcaklığı -40C ile +85C arasındadır.

AD 7715İN YAZMAÇ (REGISTER) YAPISI

Haberleşme (Comunication) Yazmacı (RS1, RS0=0)

Haberleşme yazmaçsı okunabilir ve yazılabilir bir yazmaçdır. Tüm haberleşmeler, haberleşme yazmaçsına bir yazma orasyonu ile başlamak zorundadır. Haberleşme yazmaçsına yazılan veri bir sonraki operasyonun okuma veya yazma olup olmadığını ve bu operasyonun hangi yazmaçda meydana geldiğini tayin eder. Seçilen yazmaçda bir

okuma veya yazma gerekleřtiėinde, arabirim yazma operasyonu beklemek zere haberleřme yazmaına dner.

Haberleřme Yazmacı Bitleri

0/DRDY: Bir yazma iřlemi iin bu bit 0 olmak zorundadır. Bu bit 1 yapıldıėı zaman diėer bitlere saat iřareti gelmeyecek ve bu bite0 yazılana kadar bekleyecektir.

RS1-RS0: Yazma seme bitleridir. Bu bitlerin deėerine gre yonga zerindeki 4 yazmadan herhangi biri seilebilmektedir.

R/W, (OKU-YAZ SEİMİ): Bu bit bir sonraki operasyonun, seilmiş olan yazmada bir okuma veya bir yazma operasyonu olup olmadıėını belirlemektedir.

STBY: Bu bite 1 yazmakla (Susma) Standby modu veya Power-Down moduna geilir. Bu durumda sadece 10 μ A kaynak akımı ekilir. Standby konumunda, kalibrasyon bilgisi ve 3 kontrol bilgisi ilgili yazmada oluřturulur. Bu bite bir0 yazıldıėında normal alıřma durumuna geilmektedir.

Setup Yazmalar (Register) (RS1, RS0=0, 1)

Setup yazmaı okuma veya yazma yapabilen 8 bitlik bir yazmadır. Bu yazma setup2 yi kontrol ederek kalibrasyon modu, ıkıř oranı, tekkutup (unipolar)-iftkutup (bipolar) operasyon gibi arzu edilen alıřmayı dzenler. Bu yazmanın ilgili bitlerinin alacaėı deėerlere gre ařaėıdaki alıřma modları geerli olmaktadır.

MD1 MD0

0 0 Normal çalışma modu

0 1 Self-Kalibrasyon modudur. Bu durumda Self-Kalibrasyon modu aktif hale getirilir. Daha sonra kalibrasyon tamamlandığında bu bitlerin (0,0) a dönmesiyle tekrar normal moduna dönecektir.

1 0 Sıfır-Sıkala sistem kalibrasyonu : Bu durumda Sıfır -Sıkala sistem kalibrasyonu aktif hale getirilir. Analog giriş gerilimi kalibrasyon boyunca sabit kalmalıdır. Kalibrasyon tamamlandığında MD1 ve MD0' ın (0, 0)' a dönmesiyle normal çalışma moduna geri dönülür.

1 1 Bu durum Tam -Sıkala sistem kalibrasyonu durumudur. Bu durumda tam-sıkala sistem kalibrasyonu durumu aktif hale getirilmiştir. Analog giriş teki giriş voltajı kalibrasyon boyunca sabit kalmalıdır. Kalibras sonunda MD1 ve MD0ın normale dönmesiyle sistem normal mod a dönecektir.

Setup Yazmaç Bitleri

1. Saat (clock) Biti:

Bu bit AD7715in çalışma frekansı ile uyumlu olarak programlanmalıdır. Eğer master saat frekansımız 2. 4576 MHz ise bu bit1 yapılmalıdır. Eğer master saat frekansımız 1MHz ise bu bit0 yapılmalıdır.

2. FS1, FS0 Filtre Seçim Bitleri

Bu bitler saat biti ile beraber çıkış işaretinin güncelleştirilmesi (update) oranını tayin eder. Yonga üzerindeki sayısal filtre kazanç seçimine bağlı olarak aynı zamanda çıkış gürültüsünü de tayin eder.

3. B/U Tekkutup (Bipolar) -Çiftkutup (Unipolar) Çalıştırma Bitleri

Bu bitin 0 olmasıyla tekkutup(bipolar) çalışma modu seçilmiş olur. Bu bitin 1 olması ile ise çiftkutup(unipolar) çalışma modu seçilmiş olur.

4. BUF, Ayırıcı (Buffer) Kontrolü

Bu bitin düşük değerli yapılmasıyla analog giriş üzerindeki buffer kısa devre edilmiştir. Bufferın kısa devre edilmesiyle analog kaynaktan akan akım $250\mu A$ e düşürülmüş olacaktır.

5. FSYNC, Filtre Uyumu

Bu bit yüksek değerde olduğu zaman sayısal filtre düğümü, filtre kontrol lojigi ve kalibrasyon lojigi bir reset durumunda tutulmuştur. Bu bit düşük değerde olduğu zaman ise modulator ve filtre veri işlemeye başlayacaktır.

Test Yazmaç Bitleri (RS1, RS0=1, 0)

Sistemi test etmek için yonga üzerine test yazmaçsı yerleştirilmiştir. Bu bitlerin hatalı durumu 0, 0dır.

Veri (Data) Yazmaçlar

16 bitlik ve sadece okuma yapabilen yazmaçdır. AD7715in sayısal işarete çevirdiği değer ile ilgili bilgiyi içeren yazmaçdır.

KALİBRASYON SIRASI

AD 7715, 3 tip kalibrasyon seçimi (opsiyon) içermektedir. Tabloda gösterildiği gibi Self -kalibrasyon, Sıfır sıkala sistem kalibrasyon, Tam sıkala sistem kalibrasyon olmak üzere 3 tip kalibrasyon modu bulunmaktadır. Kalibrasyon bitini tayin etmenin iki yolu vardır. Birincisi adımın sonunda DRDYnin düşük değere geldiğini izlemektir. DRDY sadece kalibrasyon adımının ne zaman sona erdiğini göstermez. Aynı zamanda veri (data) yazmaçsında yeni bir verinin bulunduğunu gösterir.

Kalibrasyonun ne zaman tamamlandığının tayini için ikinci metod Setup registerin MD0, MD1 bitlerinin izlenmesidir. Bir kalibrasyon komutunu takiben bu bitlerin 0, 0a dönmesi, kalibrasyon adımının tamamlandığını göstermektedir. Bu yöntem veri yazmaçsında yeni bir geçerli sonuç bulunduğunu göstermez. Bununla birlikte kalibrasyonun tamamlandığını DRDYden daha erken göstermektedir.

DEVRE TANIMLAMASI:

AD 7715 yonga üzerinde bir sayısal filtreleme ile geniş aralıkta veri ölçülmesi için tasarlanmıştır. Bu bir Δ/Σ (Delta /Sigma) çeviricidir. AD 7715 aşağıdaki bölümleri içermektedir:

1. Delta/Sigma çevirici
2. Yonga üzerinde statik RAM ve bir kalibrasyon mikrokontrolörü
3. Bir sayısal filtre
4. Çift yönlü seri haberleşme portu

Devre sadece 450 μ A besleme akımı sarfeder. Bu nedenle batarya veya benzeri besleme için idealdir. Analog giriş master saat işareti ve seçilmiş kazanç tararından tayin edilmiş oranda sürekli olarak giriş sinyali örneklenmektedir. Bir yük dengeleyen ADC (Delta/Sigma Modülatör) örneklenmiş sinyali, darbe süreleri sayısal bilgi içeren

darbe dizisine çevirmektedir. Programlanabilir kazanç işlevi, giriş örnekleme frekansının değişimiyle daha yüksek kazanç vermek için ayarlanabilir.

Analog Giriş

AD 7715 bir farksal analog giriş çifti olan, AIN(+) ve AIN(-) içerir. Bu giriş çifti programlanabilir bir kazanç, tek kutup (unipolar) ve çift kutup (bipolar) giriş sinyallerinden biri tarafından çalıştırılması ile farksal giriş sağlar. Giriş sızdırma akımı tamponlanmamış modda 1nA'dır. Örnekleme kapasitesi (CSAMP) her giriş örnekleme çevriminde AIN(+) tarafından dolar ve AIN(-) tarafından boşalır.

Anahtarın etkin direnç değeri olan (RSW), yaklaşık 7k Ω 'dur. CSAMP kapasitesi, RSW direnci ve herhangi kaynak empedansları üzerinden, herhangi bir giriş örnek saykılında doldurulmalıdır.

Giriş Örnekleme Oranı

Modülatör örnekleme frekansı, AD 7715 için seçilmiş olan kazançla bakılmaksızın fclk/128'de kalır. Çoklu örneklemenin bir sonucu olarak, cihazın giriş örnekleme oranı seçilmiş kazançla bağlı olarak değişmektedir.

Referans Girişi

AD 7715'in referans girişleri REF IN(+) ve REF IN(-) bir farksal referans giriş yeteneğine sahiptir. Bu farksal girişler için ortak mod aralığı AGND'den AVDD'ye'dir. Nominal referans gerilimi VREF, {REFIN(+)-REFIN(-)}'dir. Özel çalıştırma şekli AD 7715 -5 için +2.5 Volt ve AD 7715-3 için ise 1.25 Volt'tur. AD 7715'in uygun bir şekilde verimli çalışabilmesi için REFIN(+) daima REFIN(-)'den daha büyük olmalıdır. Referans girişler üzerindeki örnekleme oranı, fclk /64'dür. Bu değer kazançla değişmez. Kazançların 1 ve 2 olması durumlarında CREF 8pF'dir. Kazançın 32 olması

durumunda ise 4. 25 pF'dır. Eğer kazanç 128 olarak programlanmış ise bu değer 3. 3125 pF'dır. AD 7715-5 için tavsiye edilen referans gerilim kaynakları AD 780, REF43 ve REF192 entegreleridir. Üzerinde çalıştığım projede REF 195 kullanılmıştır.

Sayısal Filtreleme

AD 7715 yonga üzerinde bir Alçak Geçiren Filtre içermektedir. Sayısal filtreleme yapıldıktan sonra A/D çevirme işlemi yapıldığı için çevirme işlemi boyunca oluşabilecek gürültü bastırılmaktadır.

Sayısal Arabirim

Yonga üzerindeki yazmaçlara, seri arabirim sayesinde veri yazılabilir. Tüm haberleşmeler haberleşme yazmaçsına bir yaz işlemi ile başlar. Power _ON veya Reset sonrası sistem haberleşme yazmaçsına bir yaz verisinin gelmesini bekler. Bu yazmaçya yazılan veri, bir sonraki işlemin biroku veya bir yaz işlemi olduğunu ve bu işlemin hangi yazmaçda olacağını tayin eder.

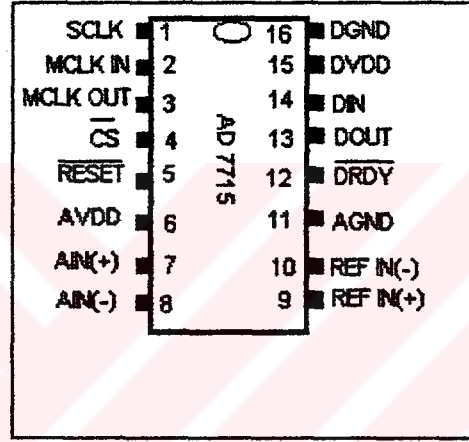
AD 7715'in seri arabirimi 5 sinyal içermektedir. CS, SCLK, DIN, DOUT ve DRDY'dir. DIN hattı, DOUT hattı üzerindeki yazmaçlardan veriye erişmek için kullanıldığında DIN hattı, yonga üzerindeki yazmaçlara veri iletmek için kullanılır. SCLK sinyali, sistem için bir seri saat girişidir. DRDY hattı, veri hazır olduğunda AD 7715'in veri yazmaçsından veri okumak için kullanılan bir sinyalidir girişidir. CS sistemi seçmek için kullanılmaktadır. Ayrıca kod çözmek içinde kullanılmaktadır.

AD 7715 seri arabirimi CS girişini düşük değere çekmek suretiyle üç-uçlu modda çalıştırılabilir. Bu durumda SCLK, DIN ve DOUT hatları AD 7715 ile haberleşmek için kullanılmaktadır. Seri arabirim RESET girişinin işletilmesi ile sağlanabilir.

Mikroişlemci Arabirimi

AD 7715in esnek seri arabirimi, pekçok mikrobilgisayar ve mikroişlemcilere kolaylıkla arabirim olmaya olanak tanımaktadır. AD7715in sayısal girişlerinin yükselme ve düşme zamanları 1µsden daha uzun olmalıdır. AD 7715in üzerindeki yazmaçların pekçoğu 8-bit yazmaçlardır. Bu durum mikroişlemcilerin 8-bit seri portlarına olan arabirimi kolaylaştırır.

Uç (Pin) Tanımlamaları



Şekil 2. 1 AD 7715 için Pin Görünümü

SCLK: Seri saat lojik girişidir. AD 7715den seri bilgi erişimi için bu girişe harici olarak bir seri saat uygulanmalıdır.

MCLK IN: Devre için gerekli olan master saat işaretidir. Bu sinyal bir kristal veya bir rezonatör ile sağlanabilir. Rezonatör veya kristalin bir ucu MCLKa ve diğer ucuda MCLK OUTa gelecek şekilde bağlanmalıdır.

MCLK OUT: Devre için master saat işareti bir rezonatör veya kristal olduğu zaman, kristal veya rezonatör MCLK IN ve MCLK OUT arasına bağlanır. Eğer MCLK IN girişine bir harici sinyal uygulanırsa, MCLK OUT ters çevrilmiş bir işaret sağlamaktadır. Bu uç harici devreler için bir saat işareti sağlamak için kullanılmaktadır.

CS: Yonga seçim ucudur. AD 7715 i seçmek için kullanılan bir logic giriştir.

RESET: Lojik giriş.

AVDD: Analog pozitif kaynak voltajıdır. AD 7715 -3 için +3. 3 Volt, AD 7715-5 için +5Voldur.

AIN(+): Analog giriştir. AD 7715e giren ve kazancı programlanabilen pozitif giriş işaretidir. Kısaca dönüştürücüden gelen farksal gerilimin pozitif girişidir.

AIN(-): Analog giriştir. AD 7715e giren ve kazancı programlanabilen negatif giriş işaretidir. Kısaca dönüştürücüden gelen farksal işaretin negatif girişidir.

REF IN(+): Referans giriş. AD 7715e farksal pozitif gerilim sağlamak için kullanılır. AD 7715için REF IN(+), REF IN(-)den büyük olmak zorundadır.

REF IN(-): Referans giriş. AD 7715e farksal referans gerilimi sağlayan negatif giriş işaretidir. Bu giriş AVDD ve AGND arasına bağlı olmalıdır.

AGND: Analog devrenin toprak noktasıdır.

DRDY:Bu giriş aktif düşük değerli kesme girişidir. Yeni bir bir verinin okunabileceğini göstermektedir.

DOUT: Seri veri çıkışı.

DIN: Seri veri girişi.

DVDD: Sayısal kaynak gerilimidir. +3. 3Volt veya +5 Volttur.

DGND: Sayısal devre için toprak noktasıdır.

UYGULAMALAR

- Ölçme ve proses kontrolü
- Endüstriyel ağırlık terazileri
- Parça sayma terazileri
- Laboratuvar ölçümleri
- Hareket ölçümleri
- Sarsıntı izleyici
- Manyetik alan izleyici

2.3 Ağırlık Ölçüm Sisteminde Kullanılan Mikrodenetleyici

Mikrodenetleyiciye dayalı sistemler, mikroişlemciye dayalı sistemlerden oldukça farklıdır. Mikroişlemcili sistem; büyük çapta ve çeşitli programları yürütmek, çok sayıda veriyi işlemek, duyarlı sayısal hesaplamaları gerçekleştirmek için kullanıldığında yüksek verim sağlamaktadır. Mikrodenetleyiciye dayanan bir sistem ise yeniden programlanma gerektirmeyen ve sürekli sabit bir programın yürütüldüğü durumlarda çok kullanışlıdır. Bunun nedeni mikrodenetleyicilerin tek-yongalık bilgisayarlar olmasıdır. Mikrodenetleyiciyi tek-yongalık bilgisayar yapan özellik, bir bilgisayarın temel bileşenlerinin aynı yonga üzerinde içerilmesidir.

Bunun için merkezi işlem birimi (MIB), bellek, giriş/çıkış ve çevre birimleri MCS-51 mikrodenetleyicisi üzerinde yer almaktadır.

2.3.1 8031 Mikrodenetleyicisi

8031, 8051 ve 8751 orijinal 8051 ailesinin üyeleridir. 64K-Byte'a kadar program ve veri belleği gereksinimi olan denetleyici sistemler için etkin çözümler sunmaktadır. Bu ürünlerin özellikleri birbirine bağlı olarak aşağıda tanımlanmaktadır.

- 128 Byte iç veri belleği
- 4x8 biçiminde düzenlenmiş, 32 adet giriş / çıkış ucu
- İki adet 16-Bit'lik zamanlayıcı
- İki öncelik düzeyi olan beş kesme kaynağı
- Programlanabilir seri giriş / çıkış kapısı
- Yonga üzerinde osilatör ve saat işareti oluşturma devresi

2.3.2 8051 ve 8751'in Özellikleri

8051 tüm 8031 özelliklerine ek olarak yonga üzerinde 4Kbyte ROM program belleği içermektedir. 8751, 8051 'in EPROM'lu tipidir ve 8031 in tüm özelliklerine ek olarak yonga üzerinde 4Kbyte EPROM program belleği içermektedir. 8031-8051 mikrodenetleyicilerinin özellikleri, MCS-51 ailesinin üyesi olan tüm mikrodenetleyicilerde çekirdek yapıyı oluşturmaktadır. Bu yüzden MCS-51 ailesinin herhangi bir üyesinden söz edildiğinde 8031 - 8051 ürünlerinin özellikleri ve bazı ek özellikler söz konusu olmaktadır.

8031, 8051 ve 8751 HMOS teknolojisiyle üretilmektedir. Bunlara tümüyle özdeş niteliklerde olan ve CMOS teknolojisiyle üretilenler ise sırasıyla, 80C31, 80C51 ve 87C51 'dir. CMOS ürünler HMOS olanlardan daha az güç tükettikleri gibi (20mA), daha yüksek çalışma frekanslarına çıkabilmektedirler.

8051 Mikrodenetleyicisi Princeton Mimarisiyle üretilmiştir. Bu mimariye göre bellek üç ayrı bölgeden oluşmaktadır: program belleği, dış veri belleği ve iç veri belleği. Geleneksel mimariye ise program ve veri bellekleri çakışık olarak gerçekleşmektedir. MCS -51 ailesi 64Kbyte program belleğine erişebilmektedirler. 8051 'de 0-4K arasında kalan 4Kbyte program belleği, ROM olarak yonga üzerinde yer almaktadır. İç veri belleği 8-Bit adresleme gereksiniminden dolayı 256 Byte ile sınırlanmıştır. Bu belleğin alt 128 Byte 'ı program içerisindeki değişkenleri saklamak için kullanılmaktadır. Üst 128 Byte ise özel işlev yazmaçlarına ayrılmıştır.

8051 yongası üzerinde çevre arabirimleriyle birlikte yonganın tümünü denetleyen yazmaçlar'dan oluşmaktadır. Bu yazmaçlar 128-255 adresleri arasında yer almakla birlikte 128 Byte'ın tümünü kapsamazlar. Adresleri 8'e tam olarak bölünebilen yazmaçlar ayrıca bit-bit adreslenebilmektedirler. İç veri belleğinin 16-Byte'lık bölgesi ve özel işlev yazmaçlarının bir bölümü bit-bit adreslenebilmektedir. 8051 'in bu özelliği bit boyutundaki değişimleri işlemeye yüksek verim sağlamaktadır. 8051, istenirse yonga

dışında, program belleğinden tümüyle ayrık olan ve 64Kbyte 'a kadar genişletebilen dış veri belleğini adresleyebilmektedir.

8051 'in veri yolları 8 bit'tir. 64 KByte genişlikteki adres bölgesini kapsamak için 16-bit adres kullanılmaktadır.

2.3.3 8051 Komut Biçimleri

- Bir komutun ilk byte'ı her zaman opkodunu ve aşağıdaki iki bilgiyi taşır.
- Yürütülecek olan komutun ne tipte olduğunu belirler.
- Opkodu ile ilgili olarak aynı komutta yer alan byte sayısını (0, 1, 2) merkezi işlem birimi'ne (MİB)'e bildirir.
- İşlenen bytlar ise doğrudan veri byte'ı, 8-bit adres bit'leri, 16-bit adres yada doğrudan verilmektedir.

2.3.3.1 Akümülatör

8051 'de aritmetik işlemleri, akümülatörü hedef işlenen olarak kullanılmaktadır. Döndürme, evirme gibi işlemler akümülatör üzerinde gerçekleştirilmektedir.

2.3.3.2 Yazmaçlar

Yazmaçlar yonga üzerindeki RAM 'ın altındaki sekiz byte'ta yer almaktadır. Burada R0 adresi '0', R1 adresi '1' ve R7 adresi '7' olan byte 'a karşı gelmektedir. Bu yazmaçlardan birisini işlenen olarak kullanan komutların düşük anlamlı üç bitinde yazmacın numarası yer almaktadır. Böylece opkodu ile işlenen adresi tek byte içinde birleştirilmektedir. Sonuçta program uzunluğu azalmış olmaktadır.

2.3.3.3 Özel İşlev Yazmaçları

Yonga üzerinde dolaysız adreslenebilen bölgedeki 127-255 aralığı özel işlev yazmaçlarına ayrılmıştır. Özel işlev yazmaçları yonga üzerindeki çeşitli çevrebirimlerine karşılık gelmekte ve bunların çalışmalarını denetlemektedir. Bunlar genel olarak :

- Zamanlayıcılar
- Seri giriş/çıkış kapısı
- Kesme düzeneği
- Paralel giriş/çıkış kapıları (P0, P1, P2, P3) olarak sayılabilir.

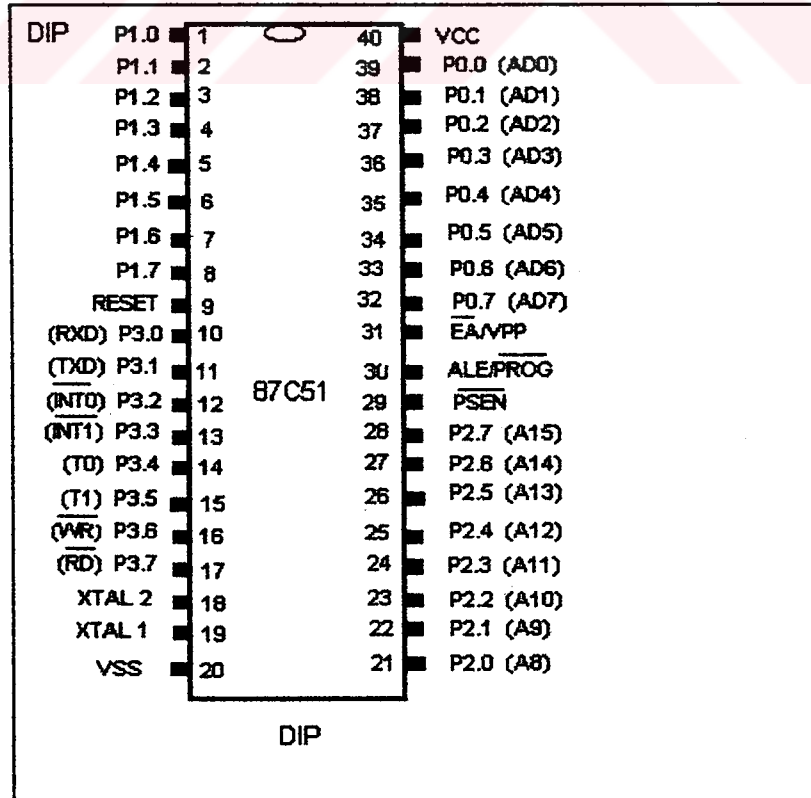
2.3.3.4 Program Belleği

8051 yongası üzerindeki 4KByte program belleğine (8052 'de 8K) yonga dışından 60Kbte'a kadar dış program belleği eklenebilmektedir. Toplam olarak 64 KByte'a kadar çıkabilmektedir.

2.3.3.5 Program Durum Belirteci

8051 de çeşitli özel-amaçlı bayrakların bulunduğu byte geleneksel bir ad olan Program Durum Belirteci (PSW) adı verilir. Bu bayrakların bir kısmı kullanıcı tarafından tanımlanabilmektedir. Bayraklar bit-bit işlenebilmektedir.

Şekil 5. 2 8751 bacak bağlantıları



3. MİKRODENETLEYİCİ SİSTEM İLE AĞIRLIK ÖLÇÜMÜ

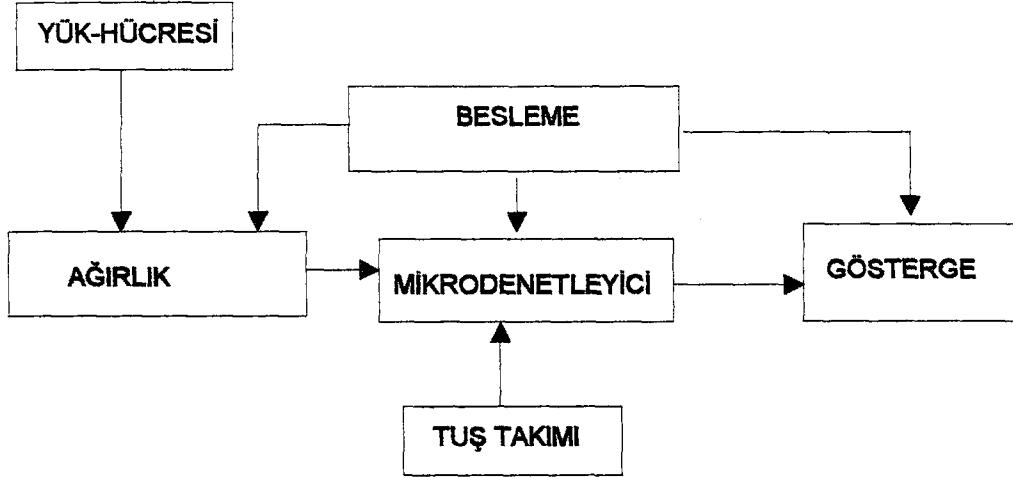
Ölçüm sisteminde çok önemli yer tutan ağırlığın, çok hassas bir şekilde ölçülmesi, ölçüm sistemlerinde çok önemli bir yer almaktadır. Çünkü, ağırlık ölçüm sistemleri, gerek sanayinin bazı alanlarında ve gerekse ticari alanda kullanılan önemli bir sistemdir. Ölçülen değer adı verilen fiziksel nicelik, bir ölçüm sistemi kullanılarak elektriksel işaret gibi ölçülebilen bir değere dönüştürülür. Bu işaret daha sonra kontrol amacıyla da kullanılabilir.

Ağırlığı ölçmek için kullanılabilecek en uygun yöntem ağırlık bilgisini elektriksel bir işarete dönüştürmektir. Bunu gerçeklemek için ise dönüştürücüler (Transdüzer) içinde çok önemli yeri olan Yük -Hüçresi (Load-Cell) kullanılmaktadır. Transdüzerler farklı tür enerjiler arasında dönüşüm yaparlar. En büyük dezavantajları ise kalibrasyonlarının zor olmasıdır. Çıkışı elektriksel olan dönüştürücüler, elektriksel işaretlerin ölçülmesinin, işlenmesinin ve bir yerden bir başka yere iletilmesinin kolay olmasından dolayı daha çok tercih edilmektedir. Dönüştürücü eleman seçiminde aşağıdaki özelliklere dikkat etmek gerekmektedir.

- Duyarlılık
- Çalışma sınırları
- Frekans cevabı veya rezonans frekansı
- Çevresel uyum
- Gürültüye karşı duyarlılık.
- Lineerlik veya doğrusallık.

Seçilen elemanların bir kısmı pasif eleman oldukları için dışardan uygun bir enerjinin verilmesi gerekmektedir.

AĞIRLIK ÖLÇÜM SİSTEM BLOK DİYAGRAMI



Şekil 3. 1 Ağırlık ölçüm sistemi blok diyagramı

3.1 AĞIRLIK ÖLÇÜM SİSTEMİNİN DONANIMI

Mikrodenetleyici temelli sistemimizde, donanım basit bir şekilde aşağıdaki bölümlerden oluşmaktadır.

- Analog ölçüm ve sayısal işarete çevrim
- Besleme Yapısı
- Mikrodenetleyici ile ölçüm kısmı
- Gösterge bölümü
- Tuş takımı bölümü

3.1.1 Besleme Yapısı

Bu yapı içinde tüm sistemin ihtiyacı olan gerilimler üretilmektedir. Devrede transformatörden gelen gerilim diyotlar vasıtasıyla tam dalga doğrultulduktan sonra tümdevre gerilim regülatörleri (LM 7808, 7805) ile sistemin ihtiyacı olan +8V, +5V DC gerilimler üretilmektedir. Analog kısım, sayısal kısım ve gösterge kısımlarını beslemek için bu voltajlar elde edilmektedir. Ağırlık ölçüm devresinde besleme devresinin yapısı ile ilgili olarak şekil ek-2 de verilmektedir.

3.1.2 Mikrodenetleyici ile Ölçüm Kısımı

Bu kısımda iki kısımdan oluşmaktadır. Bunların birincisi, analog kısım ve diğeri ise sayısal kısımdır. Analog kısımda Yük-Hücrenden (Load-Cell) gelen ağırlıkla ilgili hem negatif ve hemde pozitif fark gerilimi 1K Ω luk iki direnç ile sınırlandırıldıktan sonra, 10nF / 63 Volt iki kondansatör ile filtreleme işlemine tabii tutulur . Daha sonra bu ağırlık ile ilgili elektiriksel farksal işaretler herhangi bir şekilde kuvvetlendirme işlemine tabii tutulmaksızın direkt olarak ağırlıkla ilgili bilginin kendisi, Analog olarak AD 7715in 7 ve 8 nolu girişlerine uygulanmaktadır. Tasarlanan devredeki analog kısmın en büyük özelliği her hangi bir şekilde kuvvetlendirici, OP-AMP veya başkabir elemana gereksinim duymadan sayısal işarete çevirme işlemini gerçekleştirmesidir. Bu kısımda kullanılan AD 7715 için aşağıdaki bağıntı doğrultusunda giriş örnekleme oranı olan fs değeri hesaplanır.

$$f_s = \text{GAIN} \times f_{clk} / 64$$

(3. 1)

Eğer kazanç oranları 1 ve 2 ise

$$f_s = f_{clk} / 8$$

(3. 2)

Eğer kazanç oranları 32 ve 128 ise

Giriş farksal gerilim aralığı ise aşağıdaki bağıntı ile hesaplanır.

$$+V_{ref} / \text{GAIN}$$

(3. 3)

$$-V_{ref} / GAIN \quad (3.4)$$

Pozitif Tam sıklık kalibrasyon sınırı ise aşağıdaki bağıntı ile hesaplanır.

$$(1.05 \times V_{ref}) / GAIN \quad (3.5)$$

Negatif tam sıklık kalibrasyon sınırı ise

$$-(1.05 \times V_{ref}) / GAIN \quad (3.6)$$

bağıntıları ile hesaplanmaktadır.

Tasarlanarak gerçekleştirilen devrede master saat frekansı 1 MHz olarak belirlenmiş ve bu değere göre setup ayarı yapılmıştır. AD 7715'in reset kısmında ise entegrenin 9 nolu RESET girişine bir direnç ve birde kondansatör bağlanır veya ek-4 de verilen şemadanda görülebileceği gibi ayrıca bir entegre kullanılarak devrenin reset alma işlemi giderilmiş olacaktır.

AD 7715'de iki farklı GND bulunmaktadır. Bunlardan birincisi DGND olarak görülen sayısal GND ;devrenin sayısal kısmına ait olan GND'dir. Diğeri ise AGND'dir. Bu ise devrenin analog kısmına ait olan GND'dir. Entegrede böyle bir ayarın yapılmasının nedeni analog kısımda olabilecek gürültüleri sayısal kısımdan bir şekilde yalıtılmış olacaktır.

Daha sonraki aşamada, AD 7715'in çıkışından sayısal işarete çevrilmiş ve seri bir şekilde çıkışa aktarılan ağırlık bilgisi, mikrodeneleyicinin P1. 0 girişine uygulanır. AD 7715'in 1 nolu SCLK çıkışı ise mikrodeneleyicinin P1. 1 girişine uygulanmaktadır. DRDY ucu ise mikrodeneleyicinin harici kesme girişi olan, INTO girişine uygulanır. Bu bağlantı ile AD 7715 e mikrodeneleyici sistemden gerekli olan kontrol işaretleri gönderilmektedir. AD 7715 programlanabilir yazmaç bankasına sahiptir. Bu nedenle

gerekli olan sayısal bilgiyi alabilmek için AD 7715 gönderilecek kontrol darbeleri bu bağlantı ile gerçekleşmektedir. Mikrodenetleyicili sistem ile ölçüm kısmına ait olan şekil ek-4 de verilmiştir.

Ek-4 de verilen şekildende görüleceği gibi AD 7715 ile 87C51 arasındaki haberleşme seri olarak gerçekleşmektedir. Bu ise AD7715in bilgi çıkış ucu olan DOUT ve bilgi giriş ucu olan DIN girişinin birbirine bağlanması suretiyle gerçekleşmektedir. AD 7715e bilgi aktarımı ve AD 7715den bilgi okunması tek çıkıştan seri olarak yapılmaktadır. Gerçeklenen devrede ağırlık ile ilgili olarak kalibrasyon bilgisini saklamak için EEPROM (24C16) kullanılmıştır. Bu kısım mikrodenetleyici tümdevre olan 87C51in 15 nolu ucu olan TIMER1 İnterrupt girişi ve 17 nolu READ ucuna bağlı olacak şekilde tasarlanmıştır. Seri haberleşme için seri interrupt rutini kullanılmıştır. Bu ise mikrodenetleyicinin TIMER1 girişi ile sağlanmaktadır.

3.1.3 Kalibrasyon Belleği ile Haberleşme Kısım

EEPROM ile yalnızca kalibrasyon için gerekli olan kalibrasyon programı saklanmaktadır. Kalibrasyon ise aşağıdaki sıra ile gerçekleşmektedir. Devre ilk açılış esnasında 4*4lük tuş takımını kullanarak, kalibrasyon için gerekli olan şifre, 4*4 lük tuş takımı ile (Bu değer isteğe bağlı olarak programdan ayarlanıyor) girilmektedir. Bu değer girilince kalibrasyon alt programına dallanılır. Daha sonra A/D çeviriciden kalibrasyona girildiği anda mikrodenetleyiciye ulaşan değer okunur ve bu değer göstergede görülür. Yalnız burada şunu belirtmek gerekir ki ; o anda okunacak değere karşılık gelen ağırlık değerini bilmemiz gerekmektedir. Bu değer ise referans değeridir. Daha sonraki aşamada göstergede görülen ve sabit olması gereken bu değere karşılık olarak bildiğimiz ağırlık değeri tuş takımı ile girilir ve hafıza tuşuna basılarak kaydedilir. Daha sonraki ölçümler yapılırken bu sabit değer referans alınarak, orantı programına dallanmak gerekmektedir. Bu referans değerler bir kez yapılan kalibrasyon işlemi sonunda EEPROMda saklanır ve her defasında mikrodenetleyici EEPROM ile yapılan seri haberleşme protokolü

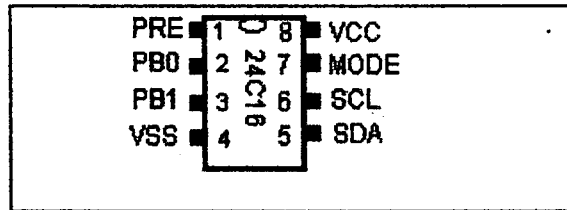
doğrultusunda kalibrasyon bilgisine ulaşmaktadır. Kalibrasyon işlemi, kullanılan ağırlık dönüştürücü elemanı olan Yük Hücresinin doğrusal bir karakteristik göstermesinden dolayı bütün ağırlık değerleri için aynı doğrusallık bağıntısı geçerlidir. Bu ise yapılan alt program ile gerçekleştirilmektedir. EEPROM ile mikrodenetleyici arasındaki bağlantı şekli ek-5 de verilmiştir.

Gerçeklenen devrede kalibrasyon bilgisini saklamak için kullanılan seri erişimli CMOS 16K ST 24C16 EEPROM ile ilgili olarak aşağıdaki bilgiler verilmiştir.

EEPROM (ST24C16)

- 100 000 kez silinip yazılabilme
- 4. 5 V ile 5 V arasında güç kaynağına ihtiyaç duyar.
- 16 BYT üzerinde sayfa yazımı
- Otomatik adres artırımı
- Koruma

ST 24C16, 16K kapasiteye sahip ve elektirikselsel olarak silinip programlanabilir bir EEPROMdur. Yapısı 256*8 şeklinde düzenlenmiş 8 blokluk 256 Bytlık alanlara bölünmüştür. Bu alanlardan istenileni seçebiliyoruz. Bu EEPROMdan bilgi okuma ve yazma seri olarak gerçekleştirilmektedir. ST 24C16a ait olan pin bağlantı şekli aşağıdaki gibidir.



Şekil 3. 2 EEPROM uç gösterimi

24C16 seri saat yolu ile bütün hafıza (Memori) çalışmalarında IIC protokolünde bir slave devre olarak davranış gösterir. Okuma ve yazma operasyonları, bus master tarafından üretilen bir START durumu ile başlatılmalıdır. START durumu, 4 devre seçim biti, blok seçim biti, okuma veya yazma bitinin birine izin vermektedir. Hafızaya (memoriye) bilgi yazımı sırasında alınan 9. bit ile bilginin alınıp alınmadığını, devreye ileterek anlayabilmektedir. Aynı yol ile veri bytlarının alımını kabul etmektedir. Veri iletimi STOP biti ile sonlandırılır. Hafızanın üst 4 bloğundaki bilgiyi koruma seçimi (opsiyon) bulunmaktadır. PB0 ve PB1 girişleritarafından koruma bölgesinin başlangıcı seçilmektedir.

24C16da iki yazma ve okuma modu vardır. Herbiri SDA yolu üzerinde devreye gönderilmektedir. Bir START durumundan sonra gönderilen 8 bitin, 4 biti devre seçim bitine aittir ki; bu devre tipini belirler, 3ü blok seçme biti ve diğeri ise READ (RW), WRITE (WR) durumuna aittir.

Eğer R/W=1 ise OKUMA (READ) durumuna aittir.

Eğer R/W=0 ise YAZMA (WRITE) durumuna aittir. Okuma ve yazma için 4 mod vardır. Aşağıdaki tabloda görülmektedir.

Tablo 3.1 Devre Seçim Kodları

	Devre Seçim Kodları					Blok Seçim		
Bit	b7	b6	b5	b4	b3	b2	b1	b0
Devre Seçim	1	0	1	0	A10	A9	A8	R/W

3.1.4 Ağırlık Ölçüm Sisteminin Gösterge Bölümü

Gösterge kısmında 5 adet Led (Light Emited Diod) Gösterge (display) kullanılmıştır. Led göstergeler ortak anotludur. Her bir gösterge, 20 mslik bir periyot içerisinde taranacak şekilde gerçekleştirilen program ile insan gözünün algılayabileceği bir

konumda olacaktır. Gösterge kısmında 2 adet 74HC574 kullanılmaktadır. Bu entegrelerden biri, ilgili pinine (11 nolu) gelen saat (clock) işareti ile göstergeler için gerekli olan satır bilgisini saklamaktadır. Diğer entegre'de ile bu displayleri tetiklemek için gerekli olan kolon bilgisi saklanmaktadır. Bu iki bilginin hangisinin aktif olacağına, sayısal kısımda kullanılan 74HC138 entegresi kullanılarak, mikrodenetleyicinin 21, 22, 23 nolu uçlarından girişine gelen bilgi ile çıkışta oluşan uygun kod çıkışı ile 10 ve 13 nolu çıkışları kullanılarak, 74 HC 574ün saat (clock) ucuna uygulanması ile gerçekleştirilmektedir.

Gösterge kısmında kullanılan ULN2803 ile göstergeler için gerekli olan satır bilgisi kuvvetlendirilerek displayi sürmek için uygun şekle çevirilir. Kullanılan diğer entegre olan UDN 2981 ile displayleri tetiklemek için gerekli olan kolon bilgisi oluşturulur. Gösterge kısmına ait olan şekil ek-6 de verilmiştir.

Yazılım incelendiği zaman ilk olarak T0; Timer, FD00 H yüklenerek 20ms de bir kesme gelmesi sağlanır. Bu ise insan gözünün göstergedeki bilgiyi algılayacağı değerdir. Daha sonraki aşamada Port 0dan 0h değeri gönderilerek ilk anda göstergelerde karartma bilgisi oluşturularak, göstergelerin kararması sağlanır.

Sonra Port2 ye sırayla 2h, 3h, 4h değerleri yükleyerek ve P0a 10000000b değeri yüklenerek tarama işlemi başlatılır. Böylece 8. kolon aktif edilir. 74HC574 ü düşen kenarda tetiklemek için P2 ye 5h değerini yükleyerek ve 0a çekilir. Daha sonra ise DPTR da yazılması istenen değerin karşılığını, ASCII_TO_DISPLAY tablosundan bulduktan sonra sırayla bu değerleri Port 0a yüklenir ve tarama işlemi devam ettirilir. Tasarlanan bu devrede 74HC574ün 10 ve 13 nolu çıkışları kullanılmıştır.

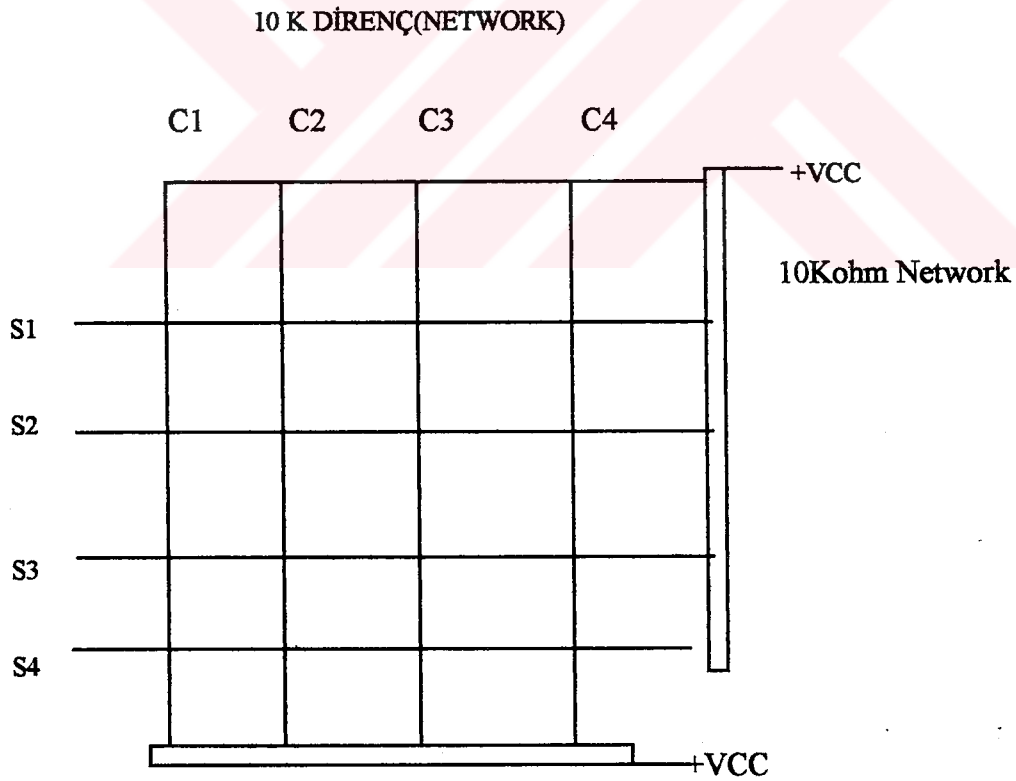
Gösterge taraması için TIMER_0 kesmesi kullanılıyor. Her bir kesme'de gösterge kolonu taranıyor. Bu sırada yazmaç olarak 6 Bytlık DBUF0 saklayıcısı, ağırlık için kullanılmaktadır. İlk olarak göstege Tutucu (Latch)larına karartma bilgisi, yani sıfır gönderiliyor. Bunun için ise ilk olarak Port 0dan bilginin gönderilmesi gerekmektedir. Daha sonra kenar tetiklemeli Flip-Flop olan 74HC 574 için gerekli olan tetikleme bilgisi gönderilmektedir. Daha sonrada değerlendirilmiş bilgi, göstergeye aktarılmaktadır. Bu işlem yapılırken R2 yazmacına DBUF0dan hangisinin aktif olacağını belirlemektedir.

Göstergeye aktarılabacak bilgi için daha önceden karakter tablosu oluşturulmuştur. Bu tablo ise ASCII_TO_DISPLAY isimli tablodur.

Değerlendirme işlemi tamamlanmış olan değerin göstergeye aktarımı sırasında bu tabloya başvurulur. R3 yazmacında kolon bilgisi bulunmaktadır. İlk anda R3 e 01111111b bilgisi yüklenir. Daha sonra ise R3 her defasında (CPL) tümlenmesi ile ilk anot tetiklemesi veriliyor. Daha sonra ise ağırlık bilgisi gönderiliyor. Böylece 8 kesme'de gösterge taranmış olacaktır.

3.1.5 Ağırlık Ölçüm Sisteminin Tuş Takımı Kısmı

Tuş takımı olarak 4*4lük bir matris düşünülmüştür. Tuş takımının prensip olarak tarama yapısı aşağıdaki gibi gösterilebilir.



Şekil 3. 3 Tuş Takımı Prensip Şeması

S1, S2, S3, S4 satır girişleri ve C1, C2, C3, C4 sütun girişleri, aynı portu (0 no'lu port), 74HC 574 entegresi yardımıyla kullanmaktadır. Yukarıda blok diyagramdanda görüleceği gibi ilk başta bütün satır ve sütunlarda 1 bilgisi bulunmaktadır. Hangi tuşun aktif olmasını istiyorsak o tuşa ait olan sütundan 0 bilgisi göndermemiz gerekmektedir. Mesala S2 ve C2nin kesişimindeki tuşun değeri ; $4 \times 1 + 2 = 6$ ncı tuşa karşı gelmektedir. Bu ise porta gönderilecek bilgiyi sürekli döndürme işlemini uygulayarak, tuş takımının sütunlarının sürekli olarak taranması ile gerçekleşmektedir. Örnek olarak bu sırada S2 ile C4 nin kesişimindeki tuşa basılırsa, $4 \times 1 + 3 = 7$ no'lu tuşa basılmış olacaktır. Bu tarama yapılırken üstüste aynı tuşa basılırsa o tuşa ait olan tuş değerinin sürekli yazmasını önlemek için üst üste 6 kez kontrol edilir ve her defasında ilk basıldığı değeri kontrol ederek, o değer korunur.

Tuş tarama işlemi, 0 nolu port üzerinden yapılmaktadır. Hat üzerinde bulunan diyodlar, birden fazla aktif olan girişleri önlemek için kullanılır. Tarama işlemi, TIMER0 kesme kullanarak gerçekleşmektedir. Tarama işlemi sürerken hangi tuşa basılmış ise o ilgili tuşa ait olan satır ve sütun bilgileri port üzerinde mikrodenetleyiciye ulaştıktan sonra, mikrodenetleyici gelen bu tuş bilgisine ait olan tuş karşılığını ilgili tablodan belirledikten sonra gerekli işlemi yürütür.

Tuş takımını tarama işlemi, gösterge taraması işlemi gibidir. Her bir kesmede bir satır taranmaktadır. Burada R3 yazmacı, hangi satırın aktif olacağının bilgisinin bulunduğu yerdir. R1 ise tuş takımındaki kolon belirleyicisidir. Tuş takımını tarama aşağıdaki sıraya göre yapılmaktadır.

1. P0 dan 0 bilgisi gönderilmektedir. Daha sonra bu bilgiler keyboard tutucu (Latch)ya yüklenmektedir. Daha sonra yine aynı porttan 74HC 574 bilgileri okunmaktadır. Eğer bilgi #FF ise tuşlara hiç basılmadığı anlaşılmaktadır. Direk olarak interrupttan çıkılır. Eğer bilgi #FFden farklı bir değer ise R1 yazmacı saydırı olarak hangi kolon olduğu bulunur. Böylece tuş belli olmaktadır.

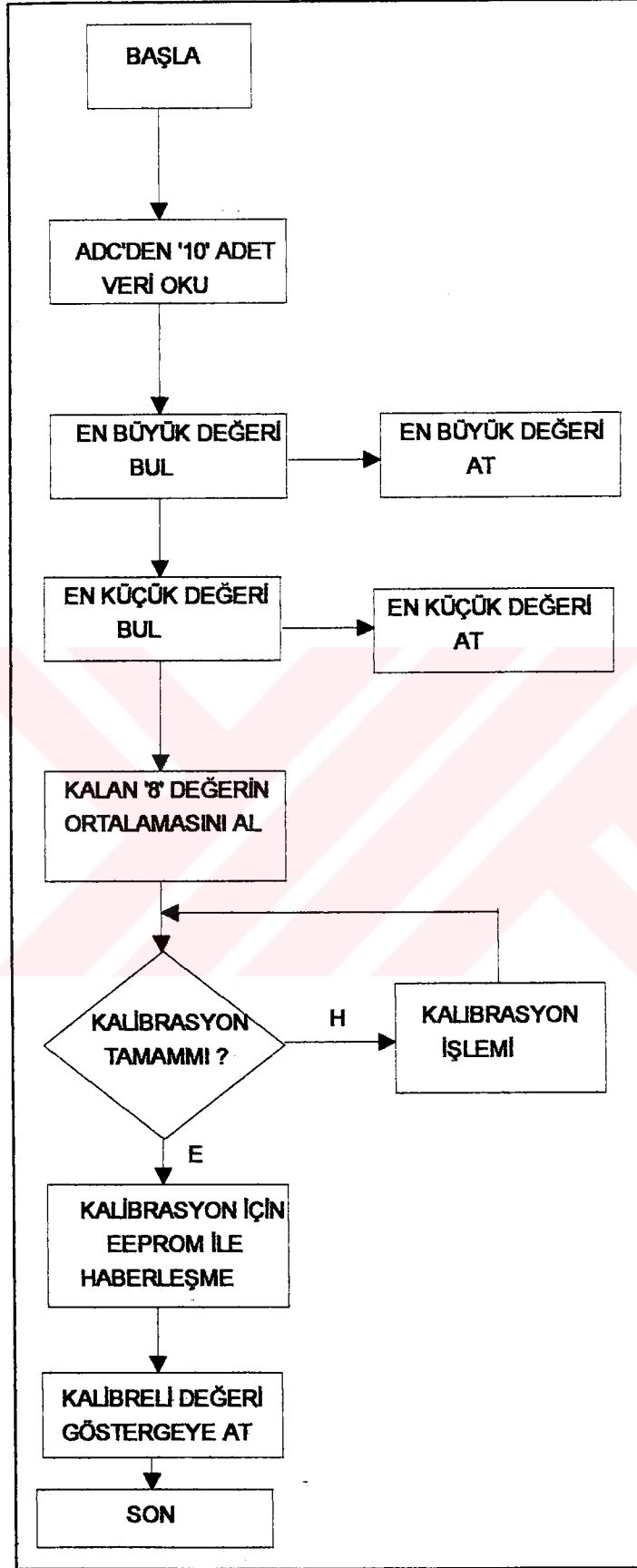
2. Burada mikrodnetleyiciye kesme gelmesi çok hızlı olduğundan, bir insanın normalde tuşa basmasını, sanki birden çok basıyormuş gibi algılar. Bu anda ekran o değer görülür. Ayrıca tuşa basma süreleri kişiden kişiye farkedebilir veya istenmeyen anlık tuşa basma gerçekleşebilir. Bütün bu olayların çözümü için 6 interrupt süresinde aynı tuşa basılmış ise program tuşa basılmış olduğunu kabul eder. Programda YTUS ve ETUŞ registerlerinin karşılaştırılması söz konusudur. Burada eğer $YTUS=ETUŞ$ ise TUSC ye bakılır. Daha sonra 6 kez kontrol edilir. Eğer 6 ya eşit ise, YTUS GTUSa atanır. Eğer 6dan büyük ise, TUSC 1 eksiltir. Eğer 6dan küçük ise, TUSC 1arttırılır. Böylece sadece 6 kez yapılan karşılaştırma sonunda, tuş kabul edilir. Böylece sürekli bir tuşa basma ve anlık temaslar engellenmiş olur.
3. Aynı anda birden fazla tuşa basılmış olabilir. Bu durumda karşılaşılan ilk tuş kabul edilir. Diğerleri basıldı kabul edilmez. Bunuda KRED ile belirlemekteyiz. Matris tarama bittiği zaman gösterge bayrağı (DISF) set edilir. R2 sıfırlanır. KRED0 ise matris taramada ilk tuş basılmış demektir. Daha sonra KRED set edilir. Bir matrix taramasında eğer herhangi bir tuşa basılmamış ise $TUSC=\#0h$ ve $YTUS=\#0h$ atanır.

3.2 AĞIRLIĞIN ÖLÇÜLMESİ VE DEĞERLENDİRİLMESİ

Gerçeklenen sistemde ADC olarak 16 -Bit Sigma Delta AD 7715 kullanılmıştır. AD 7715 Analog Devices firmasının en son üretmiş olduğu analog- sayısal dönüştürücülerden biridir. Tasarlanan sistemde kullanılan AD 7715 'in en büyük avantajı, günümüzde birçok yerde kullanılan analog -sayısal dönüştürücülerde kullanılması zorunlu harici arabirimler olan kuvvetlendirme katı, filtreleme katı ve diğer arabirimlerinin kendi içinde bulunması nedeniyle bu birimlere gereksinim olmamasıdır. Bunun yanısıra referans voltajına göre (-80 mV) ve (+2.5V) arasındaki voltaj değerlerini giriş olarak kabul etmektedir. Bünyesinde bulunan yazmaçlardan haberleşme (comunication) yazmaç ile uygun şekilde programlayarak kazanç ayarı, yazma modu ayarı, setup yazmaç seçimi, test yazmaç ayarı, veri yazmaç ayarı yapılmaktadır. Setup yazmacında ise filtre seçim çalışma modu seçimi, master osilatör seçimi, uygun bitler programlanarak gerçekleştirilmektedir.

Analog-sayısal dönüştürücüden çıkan sayısal bilgi, mikrodenetleyicinin P1. 0 nolu portu olan TIMER2 girişinden okunmaktadır. Ayrıca AD 7715'den bilgi iletişimi sırasında AD7715 için gerekli olan saat (clock) işaretleri mikrodenetleyicinin P1. 1 nolu portundan gönderilmektedir.

AD7715'den hazır olan bilgilerin mikrodenetleyiciye ulaşması için gerekli olan kesme bilgisi ise denetleyicinin dış kesme girişlerinden biri olan INT0 ile AD 7715 de ilgili uç olan DRDY arasında yapılmaktadır. AD 7715 için gerekli olan master saat (clock) işareti 1 Mhz olarak seçilmiştir. Bu seçim ise setup yazmacında ilgili bitin seçimi ile gerçekleştirilmektedir. Bu devrede ise setup yazmacındaki CLK biti '0'olarak seçilerek AD7715'in 1Mhz'de çalışması sağlanmıştır. Sisteme ait basit blok diyagramı aşağıda verilmektedir.



Ağırlık ölçme işleminde, AD7715 'den ardarda 10 değer okunur. Bunun için NET1, NET2, NET3, NET4, NET5, NET6, NET7, NET8, NET9, NET10 değişkenleri kullanılmaktadır. Okunan 10 değer içerisinde enbüyük olan ve en küçük olan değer bulunur. Bu işlem için ayrıca en büyük ve en küçük bulma alt programı yazılmıştır. Daha sonraki işlemde bulunan en büyük ve en küçük değerler atılır. Kalan sekiz değer ortalama değeri alınır. Böylece gürültülü ve titreşimli ortamlarda da çalışması sağlanmış olmaktadır. Ortalaması alınan son değer baz alınarak, diğer işlemler yapılmaktadır. Kalibrasyon işlemi ise aşağıdaki gibi yapılmaktadır. Kalibrasyon işleminden amaç istenilen gerçek ağırlık değerine ulaşmaktır. Bunun için aşağıdaki bağıntı geçerli olacaktır. Ağırlık için son değer,

$$SON(kg) = \frac{REF(kg).V_{ADCREf}}{V_{ADCSON}} \quad (3.1)$$

bağıntısı referans alınarak hesaplanmaktadır. Böyle bir doğrusal işlemi gerçekleştirmemizin nedeni Yük Hücresi (Load-Cell) 'nin ağırlıkla doğrusal değişen bir değişim göstermesinden dolayıdır. İlk açılış anında AD7715'den okunan referans ağırlığa ilişkin değer göstergeye aktarılır. Bu değer gösterge'de görülür. Daha sonra bu değere karşılık gelen referans ağırlık değeri tuş takımı ile girilir. Böylece orantı işlemine geçilmektedir. Dolayısıyla daha sonraki ağırlık değerleri bu referans değer baz alınarak, yukarıdaki bağıntı doğrultusunda hesaplanır ve göstergeye aktarılır.

3.2.1 ADC Okuma-Yazma İşlemi

AD 7715'den sayısal bilgi elde edilebilmesi için, gerekli şartların sağlanması gerekmektedir. Bu ise yapısında bulunan yazmaçların uygun bir şekilde programlanması ile gerçekleşebilmektedir. Bunun içinde, AD7715'e bilgi gönderilmesi gereklidir. Bu işlem ise aşağıdaki gibi yapılır.

ADC'ye bilgi yazılırken ilk önce SCLK girişi '1 yapılır. Bunun yapılmasının nedeni porta bilgi yazılırken SCLK girişi '1 olmalıdır. ADC 'ye bilgi yazarken ve bilgi okurken işlemler 8 bit olarak gerçekleşir. Bunun için B yazmacı sayaç olarak kullanılmaktadır. Bu işlem için B yazmacına 8 değeri atanır. Yazma ve okuma işlemi ADC'nin DOUT ve DIN uçlarının birbirine bağlanması ile iki yönlü bilgi alışverişi ile sağlanmaktadır. ADC'ye bilgi yazılırken, bilgi akümülatör'den elde(carry) bitine sırayla döndürme işlemi ile aktarılır. Daha sonra bu sıraya paralel olarak elde bitindeki bilgiler P1.0 nolu çıkış portuna sıra ile aktarılır. Bu sırada bir döngü işlemi ile B sayacı '0 olana kadar bu işlem tekrarlanır. ADC'den bilgi okunması işlemide aynı mantıkla gerçekleştirilir. B yazmacı aynı şekilde 8 bit sayaç olarak kullanılır. Bilgi okunurken SCLK '1 yapılır ve DOUT '1 yapılarak veri hattı boşaltılmaktadır. B registeri '0 olana dek aynı işlem sürdürülür.

ADC'den 16 bit bilgi okunmalıdır. Bunu gerçeklerken ilk önce düşük anlamlı 8 biti okunur ve daha sonra yüksek anlamlı 8 biti okunmaktadır. AD 7715'den okunan düşük anlamlı 8 bit GELENGE+0 olarak tanımlanan yazmaca aktarılır. Diğer kalan 8 bit ise GELENGE+1 olarak tanımlanan yazmaca aktarılmaktadır. ADC okunması ve ADC'ye yazma işlemine ait olan alt program ile bu işlemi daha iyi bir şekilde açıklayabiliriz.

Bu program parçası A'da verilen veriyi ADC'ye yazar |

C_YAZADC:

SETB SCLK

MOV B, #8

C_YAZL1:

CLR SCLK

NOP

RLC a

MOV DOUT, c ; Veriyi çıkışa yaz

```

SETB  SCLK
DJNZ  B, C_YAZL1
SETB  DOUT      ; Veri hattini bosalt
RET

```

Bu program parcasi ADC'den okunan veriyi A'ya yazar |

C_OKUAD8:

```

SETB  SCLK      ;
SETB  DOUT      ; Veri yolunu bosalt
MOV   B, #8
JB    DRDY, $    ; Yazmaya hazir degilse bekle

```

C_OKUL18:

```

CLR   SCLK
NOP
MOV   C, DOUT    ; Veriyi oku
RLC   A
SETB  SCLK
DJNZ  b, C_OKUL18
RET

```

Bu program parcasi ADC'den okunan 16bit veriyi |

NET1+0, 1'e yazar. |

C_OKUAD16:

```

MOV   A, #DATA_COM
CALL  C_YAZADC
CALL  C_OKUAD8
MOV   GELENG+1, a
CALL  C_OKUAD8
MOV   GELENG+0, a
RET

```

Bu program parçası ADC on kosullama islemlerini yapar |

C_KOSUL:

```

MOV  A, #SETUP_COM ; 0001 0011 değerine karşılık gelmektedir.
CALL C_YAZADC
MOV  a, #SETUP1   ;    0000 0101 değerine karşılık gelmektedir.
CALL C_YAZADC
CALL DELAY2
MOV  A, #SETUP_COM
CALL C_YAZADC
MOV  A, #SETUP2   ;0000 0100
CALL C_YAZADC
CALL DELAY2
MOV  A, #SETUP_COM
CALL C_YAZADC
MOV  A, #SETUP3 ;    0100 0100 değerine karşılık gelmektedir.
CALL C_YAZADC
CALL DELAY2      ; SELF KALIBRASYONU BEKLE
MOV  A, #DATA_COM
CALL C_YAZADC
RET

```

İlk olarak AD 7715 dönüştürücünün yazmaçlarının koşullanması gerekmektedir. Mikrodenetleyici ile AD7715 arasında bilgi alış verişi P1. 0 girişinden yapıldığından seri haberleşme işlemleri yapılmaktadır. Tasarlanan devrede ağırlık okumak için AD7715 için gerekli olan konfigürasyonlar aşağıdaki gibi verilmektedir.

1. SETUP-COM13h , 0001 0011
2. SETUP1..... 05h , 0000 0101
3. SETUP2.....04h , 0000 0100
4. SETUP344h, 0100 0100
5. DATA-COM.....3Bh, 0011 1011

Temel olarak AD7715 de iki adet yazmaç bulunmaktadır.

1.Haberleşme Yazmacı

2.Setup Yazmacı

1.HABERLEŞME YAZMACI

DRDY	ZERO	RS1	RS0	R / W	STBY	G1	G0
------	------	-----	-----	-------	------	----	----

RS1	RS0	
0	0	Haberleşme (Communication) yazmacı
0	1	Setup yazmacı
1	0	Test yazmacı
1	1	Veri yazmacı

1. ADIM

SET-COM yazmacı ile haberleşme yazmacından, setup yazmacı ve kazanç değeri 128 seçilir.

2. ADIM

Çiftkutup (bipolar) ve tekkutup(unipolar) çalışma seçeneği ile FSYNC bayrağının seçimi gerçekleştirilmelidir. Burada filtre işlemleri başlatılmaktadır.

3. ADIM

FSYNC bayrağı (flag) ‘0’ konumuna getiriliyor.

4. ADIM

AD 7715’den veri çıkışlarının olabilmesi için gerekli olan self kalibrasyon moduna girilmesi işlemleri başlatılıyor.

5. ADIM:

Veri yazmacına geçiliyor.

İlk olarak DATA-COM yazmacına yazma işlemleri yapılır. Daha sonra ise AD7715 ‘den bilgiler 8bit-8bit okunmaktadır. Burada bilgilerin seri olarak girip çıkması esnasında, SCLK ‘ı çalıştırmak gerekmektedir. Ağırlık okumak için C-OKUAD16 alt programı çağırılıyor ve, ağırlık bayrağı (AĞIRF) set edilerek ağırlık değeri okunmuş oluyor.

3.3 Ağırlık Ölçüm Sisteminin Yazılımı

Sistem bu programa ağırlık kesme bilgisi (DRDY) ucundan geldikten sonra dallanmaktadır. Daha sonra gelen 16 bitlik veriler NET1+0, NET1+1 yazmaçlarının’da saklanmaktadır. Ağırlık değerlendirme aşağıdaki sıraya göre yapılmaktadır.

1. GELENG+0, GELENGE+1 yazmaçlarındaki bilgiler NET yazmaçlarına atanmaktadır. Burada 10 değer atanmakta ve saklanmaktadır. . Her bir kesmede (interrupt) yeni bir değer alınır ve bu değer 10 değer ötelenir.

2. Medyan Filtreleme (Yazılım Filtreleme):

Gelen 10 değerin ilk önce enbüyük(maksimum) ve en küçük (mimumları) olanları bulunur ve bu değerler atılır. Geri kalan 8 değer toplanır ve 8 'e bölünür. Şimdi sırayla bu işlemleri açıklayalım.

a. En küçük değeri bulma

En büyük anlamlı BOS+1(Bu yazmaç karşılaştırma işlemleri sırasında kullınılmaktadır) A'ya atanır. Sonra bir alt en büyük anlamlı ile karşılaştırılır. Büyük durumunda, direk bir alttaki küçük olur. Küçük durumunda ise kendisi direk en küçük olur. Düşük anlamlı bitler (16 bit) 8bit-8bit karşılaştırılır. BOS+0, NET9+0 ile karşılaştırılır. Çünkü BOS+0, BOS+1 yazmaçları karşılaştırma işlemleri sırasında değerleri atadığımız yazmaçlardır. NET yazmaçları ise gelen ağırlık bilgilerini saklamaktadır. En küçük bulma işlemleri sırasında kullanılan BOS+5 ve BOS+4 değeri hangi NET 'in en küçük olduğunu göstermektedir.

b. En büyük değeri bulma

Burada ise kullanılan BOS+2 ve BOS+3 'değerleri ise hangi NET'in en büyük olduğunu göstermektedir. Karşılaştırma işlemi yine aynı şekilde yapılmaktadır.

Oturma kontrolü

Bürüt ağırlığın ağırlık kontrolünün yapılması gerekmektedir. 10 değerden enbüyük olan değer bulunarak ortalama değerden çıkartılır. Eğer fark 40d 'den küçük ise OTURDU2F bayrağı (flagı) set edilir. Sonraki adımda 10d 'den küçük olup olmadığına bakılır. Eğer küçük ise OTURDUF (flag) set edilir. Aksi durumda ise OTURDU2F ve OTURDUF '0yapılmaktadır.

Yukarıdaki ADC alt programında belirtilen DELAY yani gecikme devresi;kesme olduğu zaman '1'den '0a düşmesi sırasında, bu geçişin denetleyici tarafında

anlaşılabilmesi için bir çevrim gerekmektedir. Bu nedenle gecikme programına ihtiyaç duyulmaktadır.

DELAY 1

```
DELAY:    MOV    COUNT, #250D
DELAY_1:   NOP
          NOP
          NOP
          DJNZ   COUNT, DELAY_1
          RET
```

***** END OF DELAY1 *****

***** DELAY 2 *****

```
DELAY2:   PUSH   AR5
          MOV    R5, #100D
DELAY2_2:  MOV    COUNT, #255D
DELAY2_3:  NOP
          NOP
          NOP
          DJNZ   COUNT, DELAY2_3
          DJNZ   R5, DELAY2_2
          POP    AR5
          RET
          END OF DELAY2
```

4. SONUÇ

Gerçeklenen ağırlık ölçüm devresinde değişik ağırlık değerlerine sahip Yük-Hücreleri için yapılan ölçüm ve test sonuçları aşağıda verilmiştir.

SM: Sapma Miktarı.

SA: Sapma Aralığı.

AÇ: Azami Çeker.

Tablo 4.1 30 Kg Yük-Hücresi için ölçüm sonuçları

AÇ:33.098Gr 1Gram Hassasiyet

Ağırlık (Kg)	SM(Gr)	SA(Gr)
30	3	29.992-29.995
20	0	20.000-20.000
15	2	15.003-15.005
10	2	10.000-10.002
5	0	5.000-5.000
0	0	0-0

AÇ:33.095 Gr 2Gram Hassasiyet

Ağırlık(Kg)

SM(Gr)

SA(Gr)

30	4	29.996-29.992
20	0	20.002-20.002
15	0	15.000-15.000
10	0	10.000-10.000
5	2	4.998-4.996
0	0	0.000-0.000

Yapılan ölçümler sonucunda elde edilen değerler, yukarıdaki tablolardan incelendiğinde, endüstride kullanılan ağırlık ölçüm cihazları ile karşılaştırıldığında, tasarlanarak gerçekleştirilen mikrodenetleyici temelli ağırlık ölçüm sisteminin özelliklerinin bu cihazları yakaladığı görülmüştür.

5. KAYNAKLAR

- [1] Karl Hoffmann “An Introduction to Measurement using Strain Gages”, 1989
- [2] Embedded Microcontroller, Intell 1995
- [3] High Speed Design ,Analog Devices 1996
- [4] Design-In Reference Manual ,Analog Devices 1994
- [5] Memory Products, Second Edition, SGS Thomson 1995.
- [6] Industrial and Computer Peripheral ICs Data Book, Second Edition, SGS Thomson , 1992



6. EKLER

Ağırlık ölçüm sistemi yazılımının akış diyagramları :



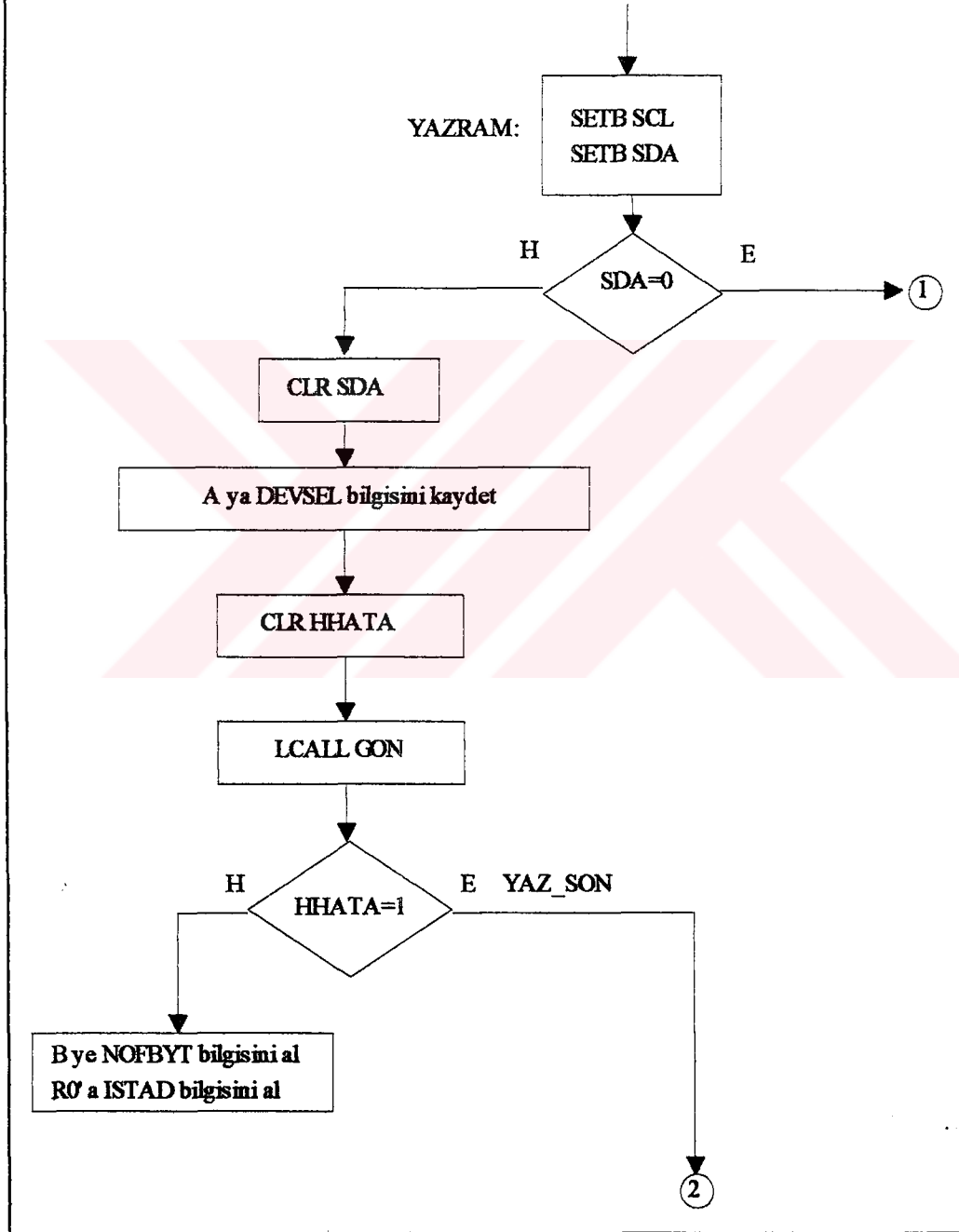
EEPROM' A YAZMA İŞLEMİ

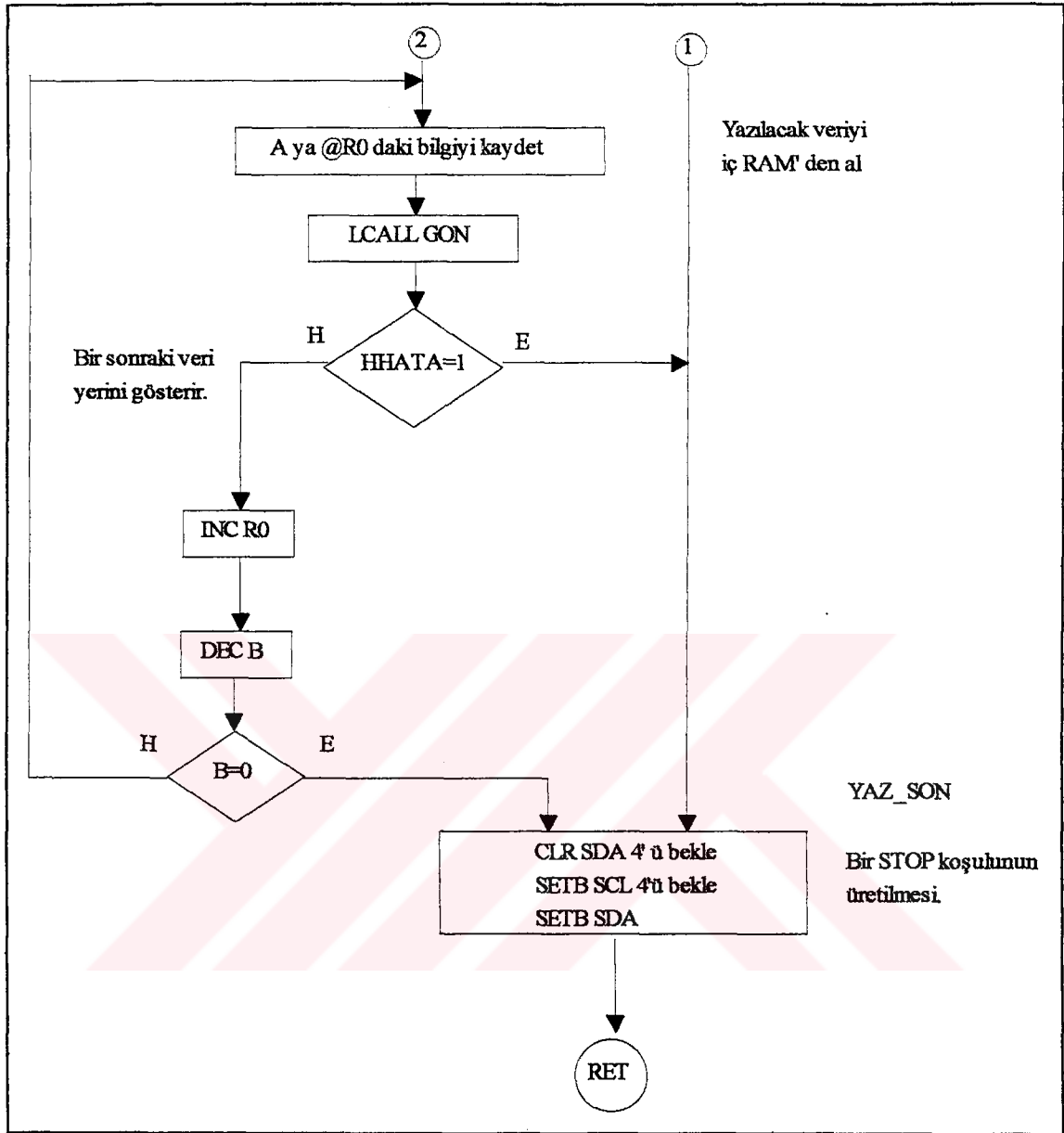
DEVSEL:Haberleşmenin kiminle olduğunu

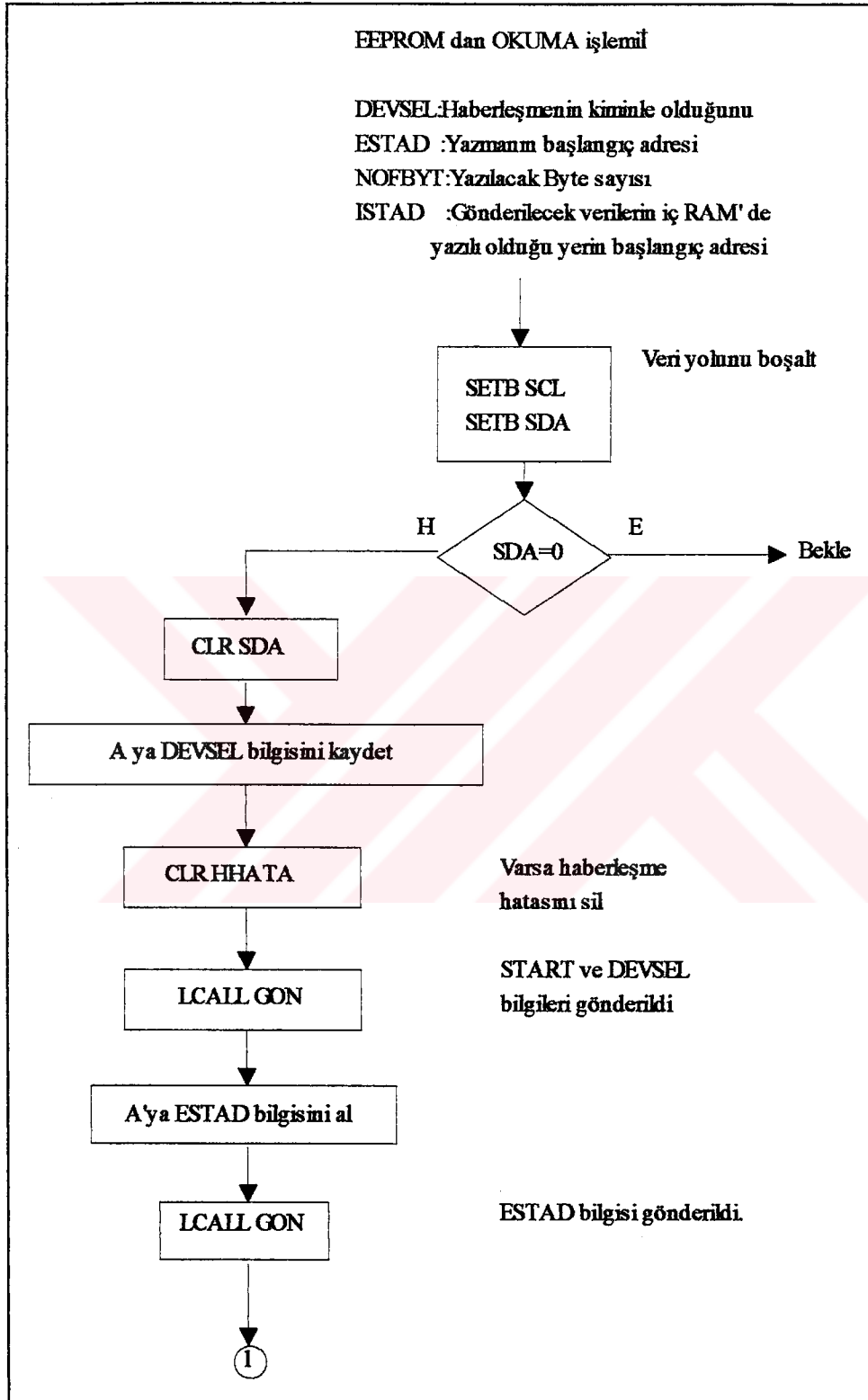
ESTAD :Yazmanın başlangıç adresi

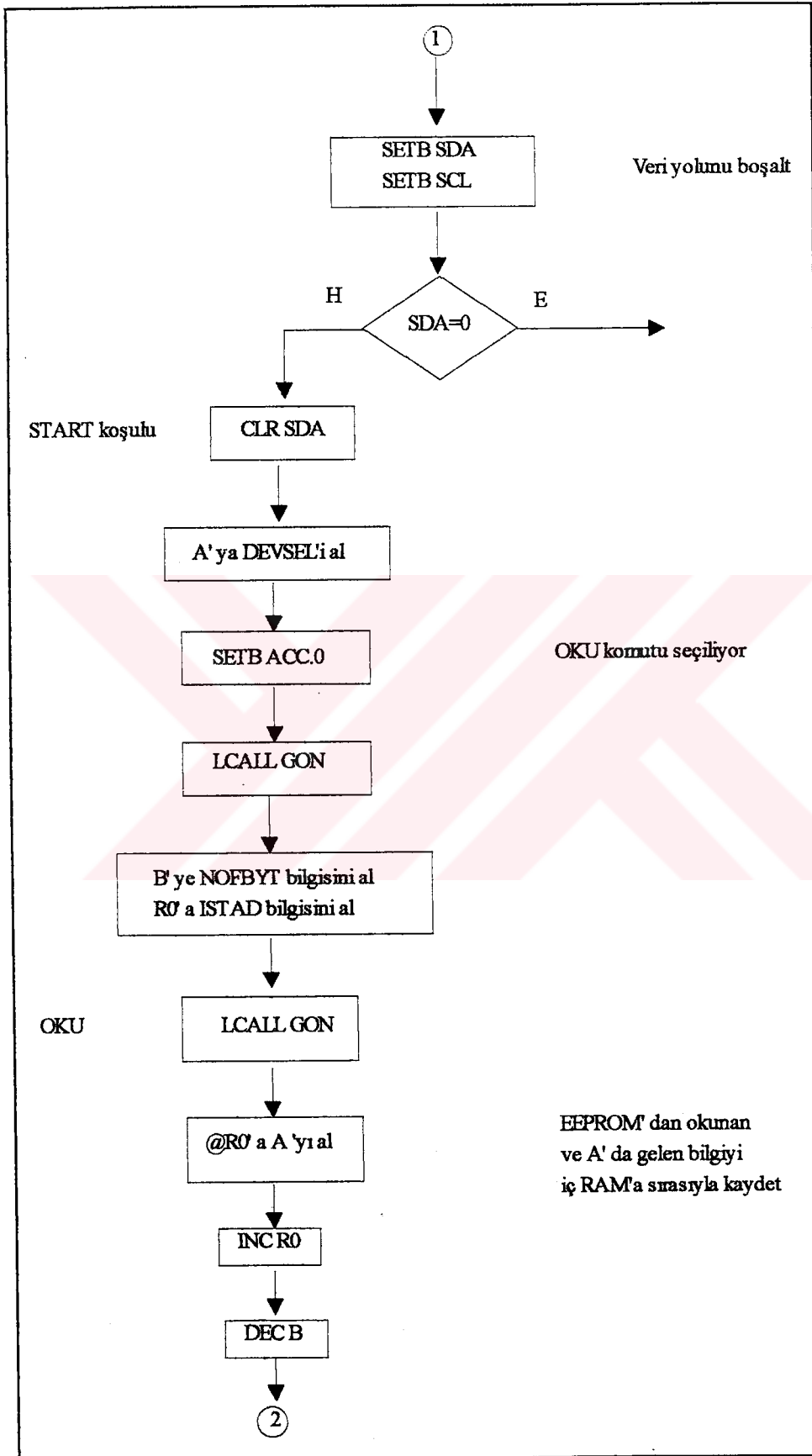
NOFBYT:Yazılacak Byte sayısı

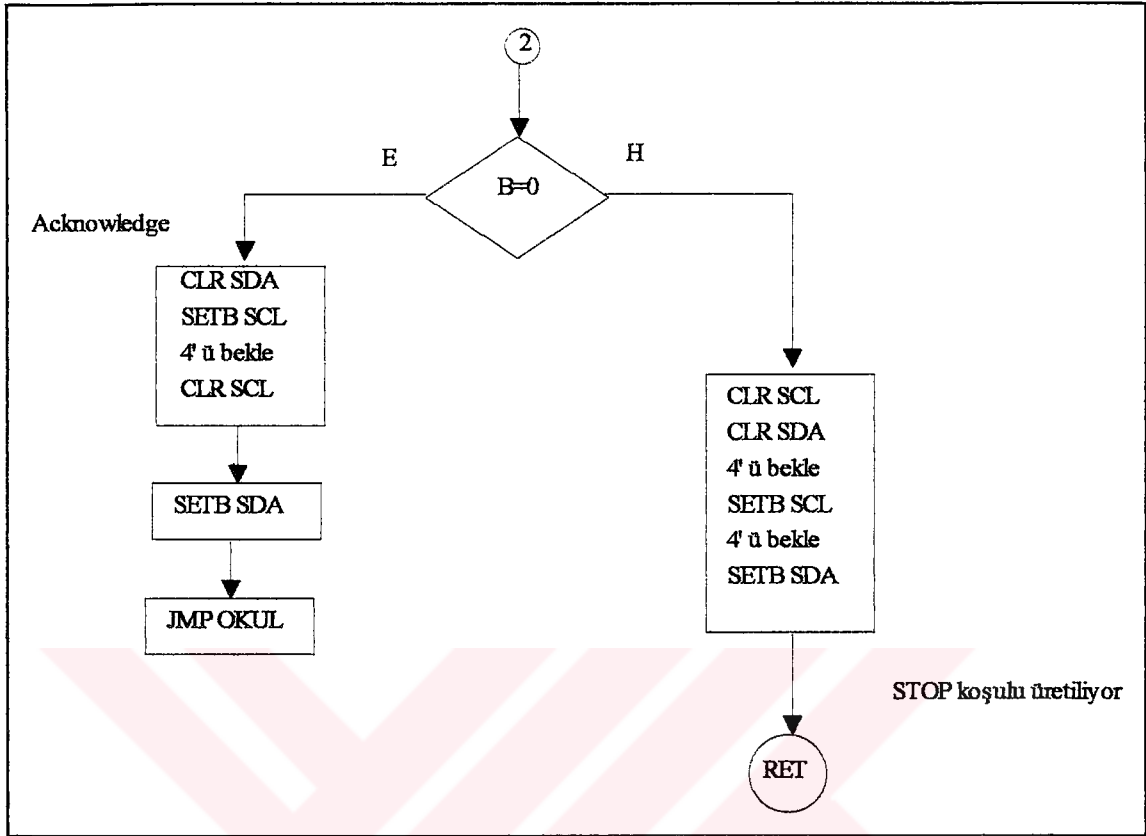
ISTAD :Gönderilecek verilerin iç RAM' de yazılı olduğu yerin başlangıç adresi

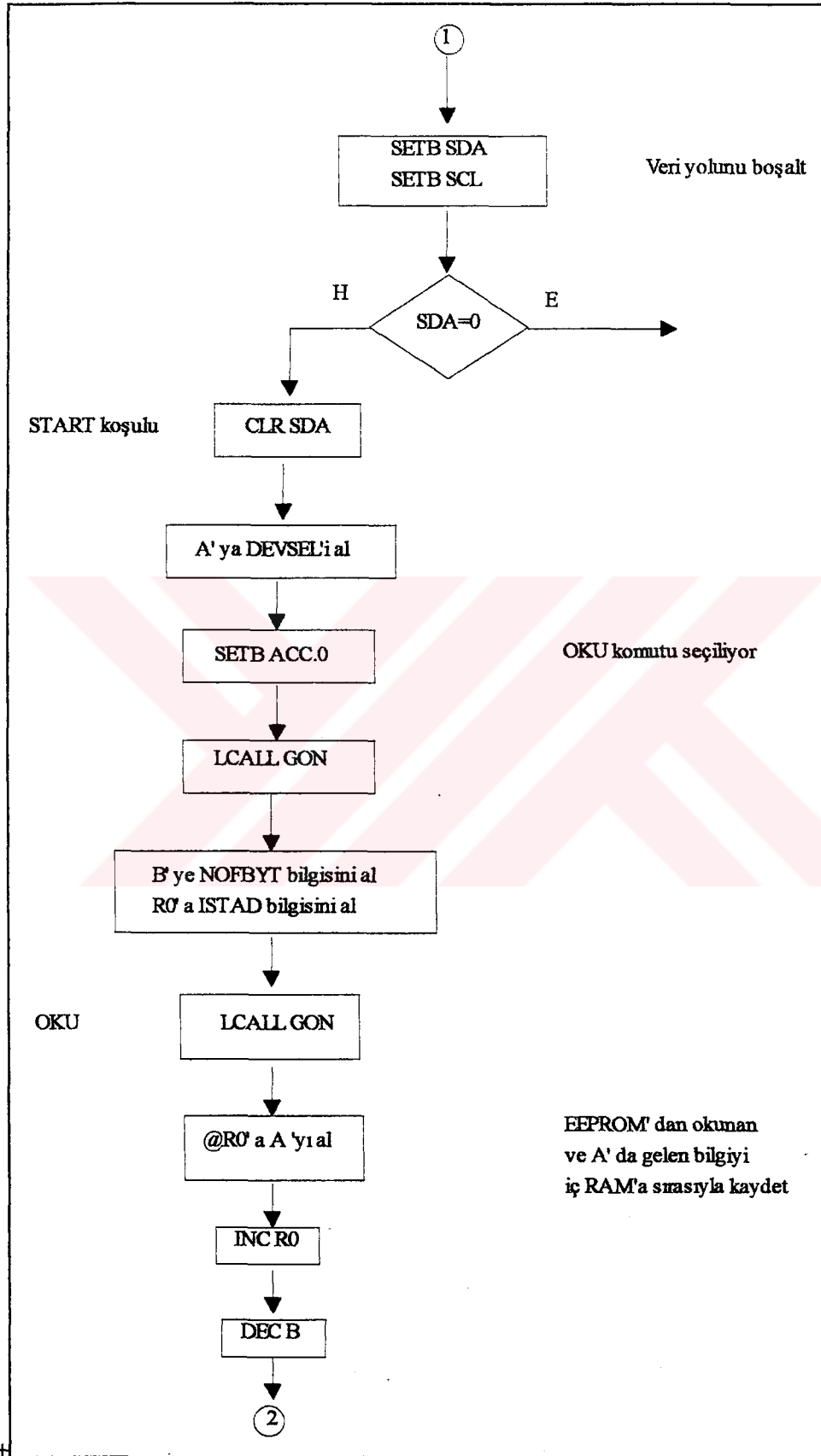


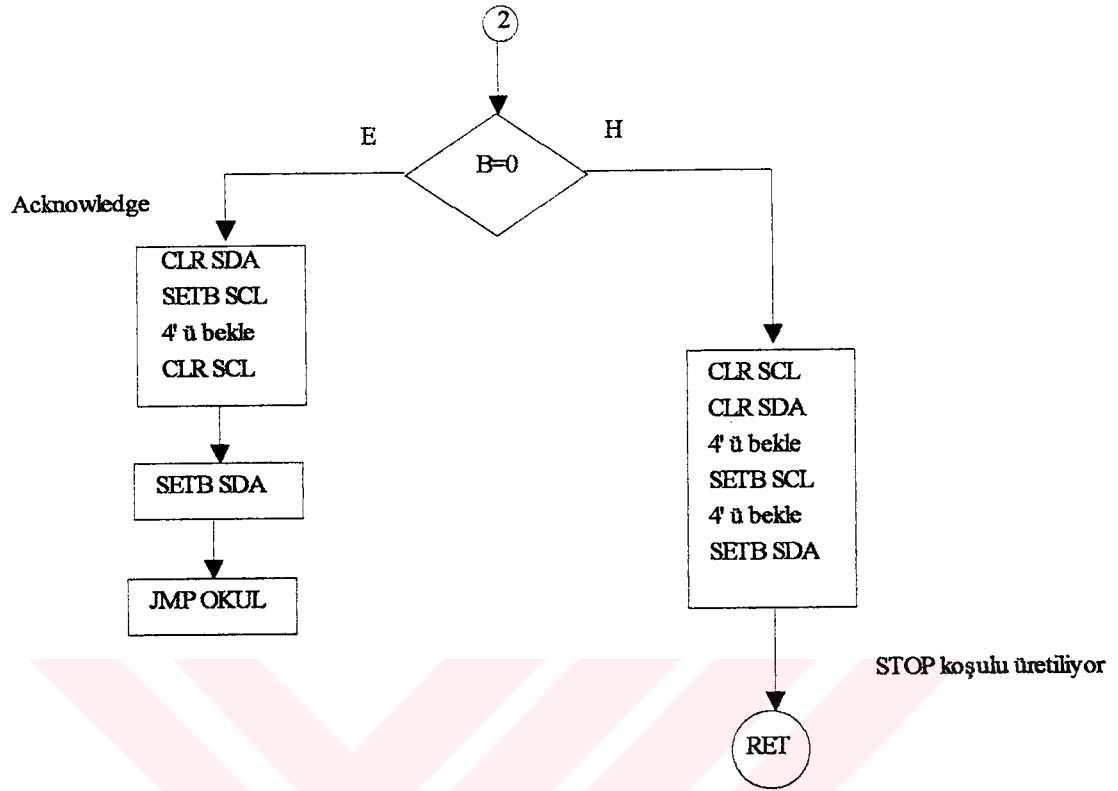






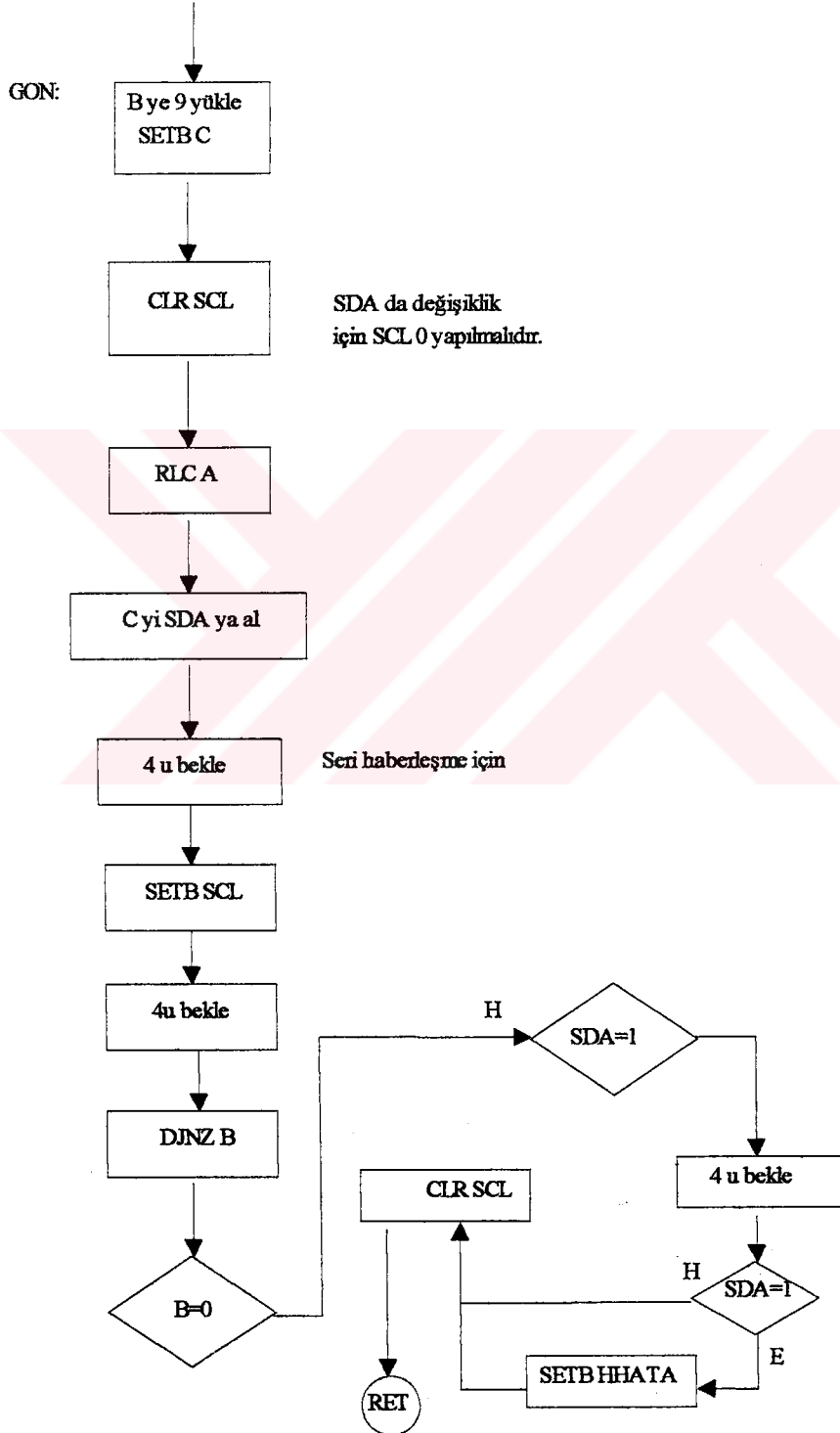




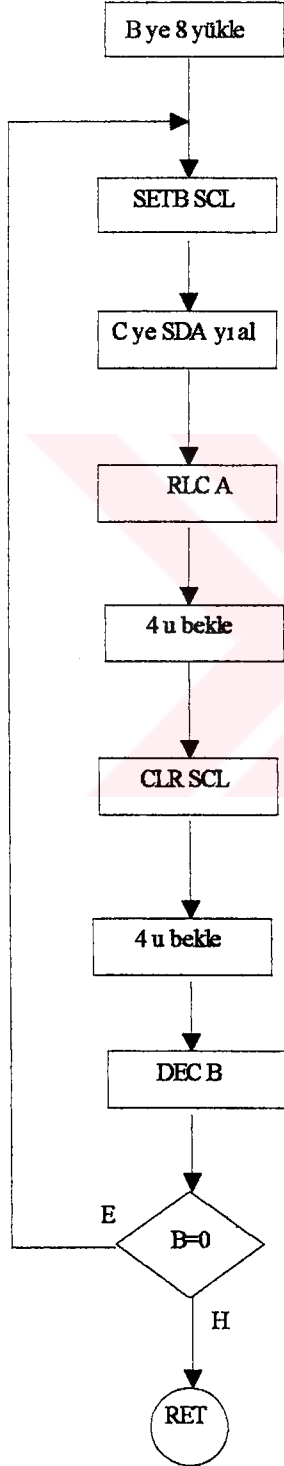


EEPROM a 8 bit veri gönderme işlemi

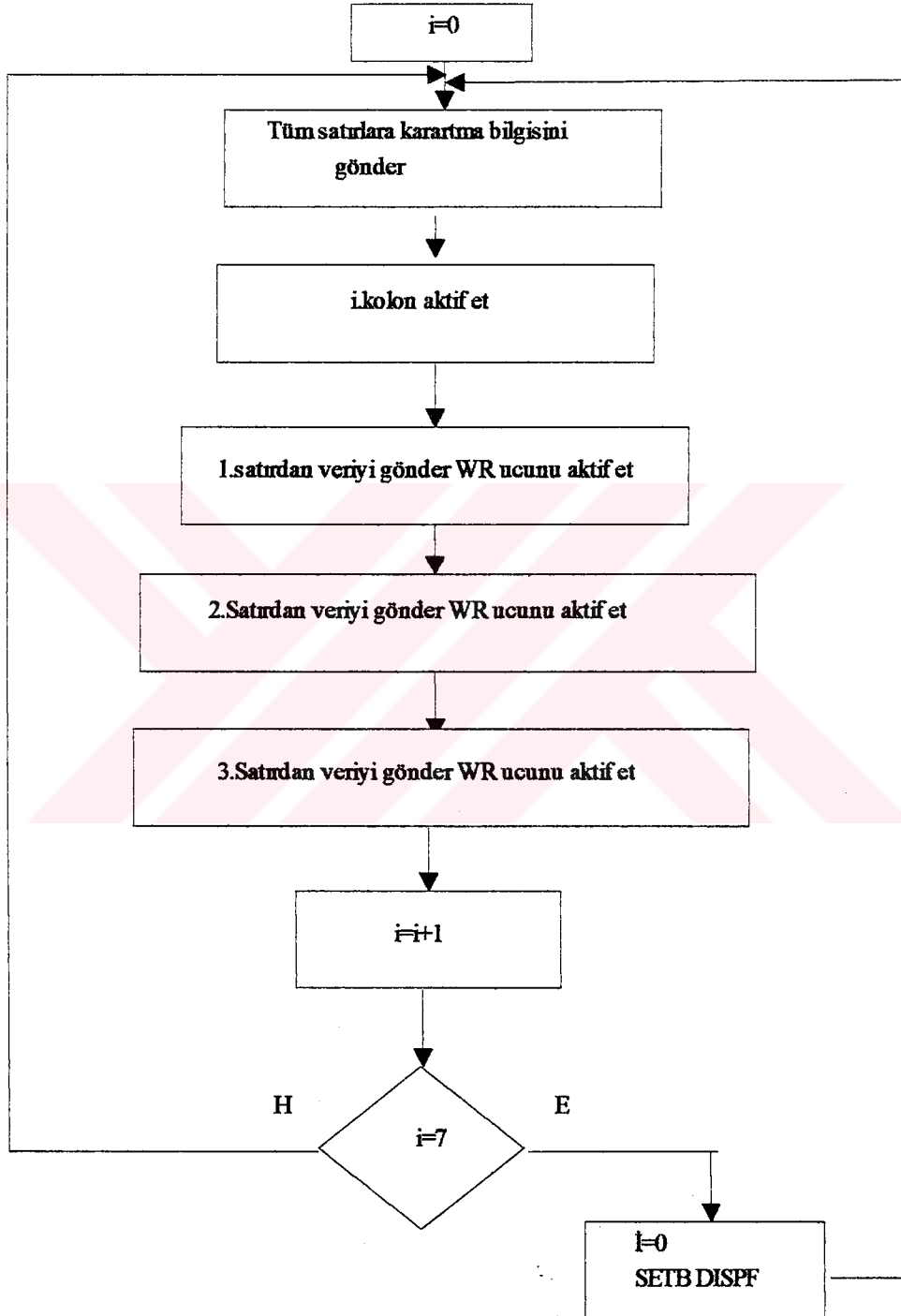
8 Bit veri gönderir ve anlaşıldı bitini kontrol eder, anlaşıldı alamazsa haberleşme biti olan HHATA yı 1 'ler. Gönderilecek bilgi programa A da verilmelidir.



EEPROM dan 8 bit veri okuma
Okuma sonunda yanlışlığı bilgisi bildirilir ve okunan bilgi
A da döner

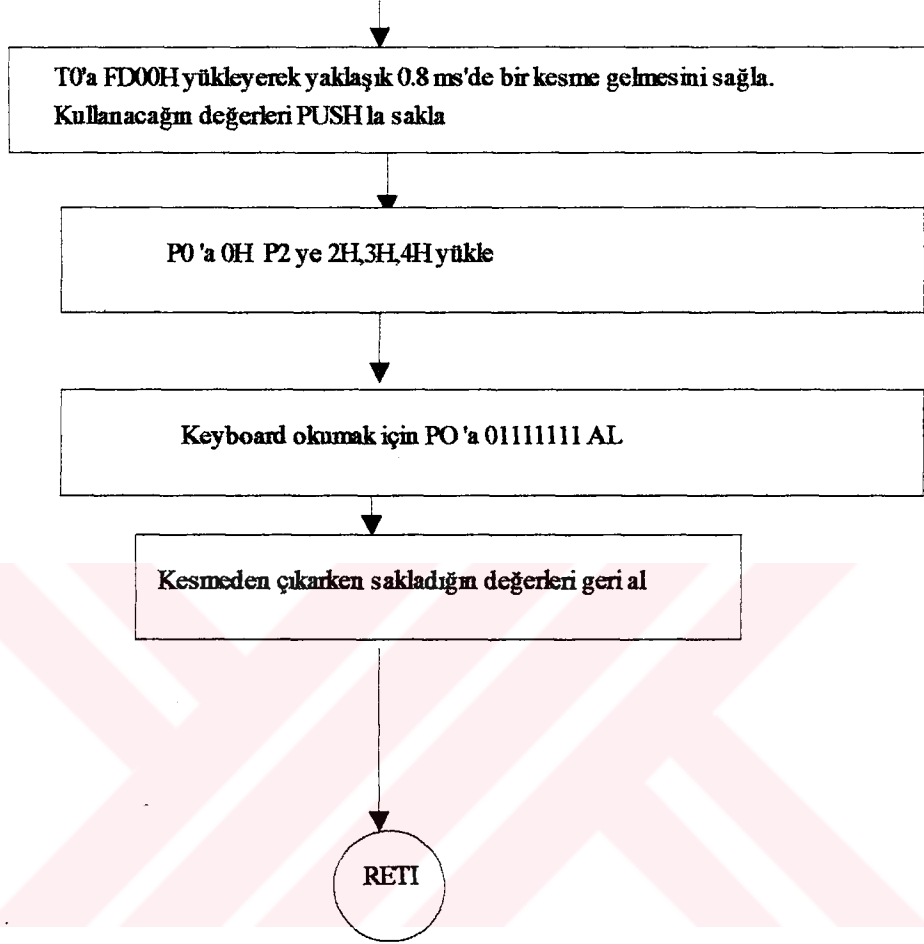


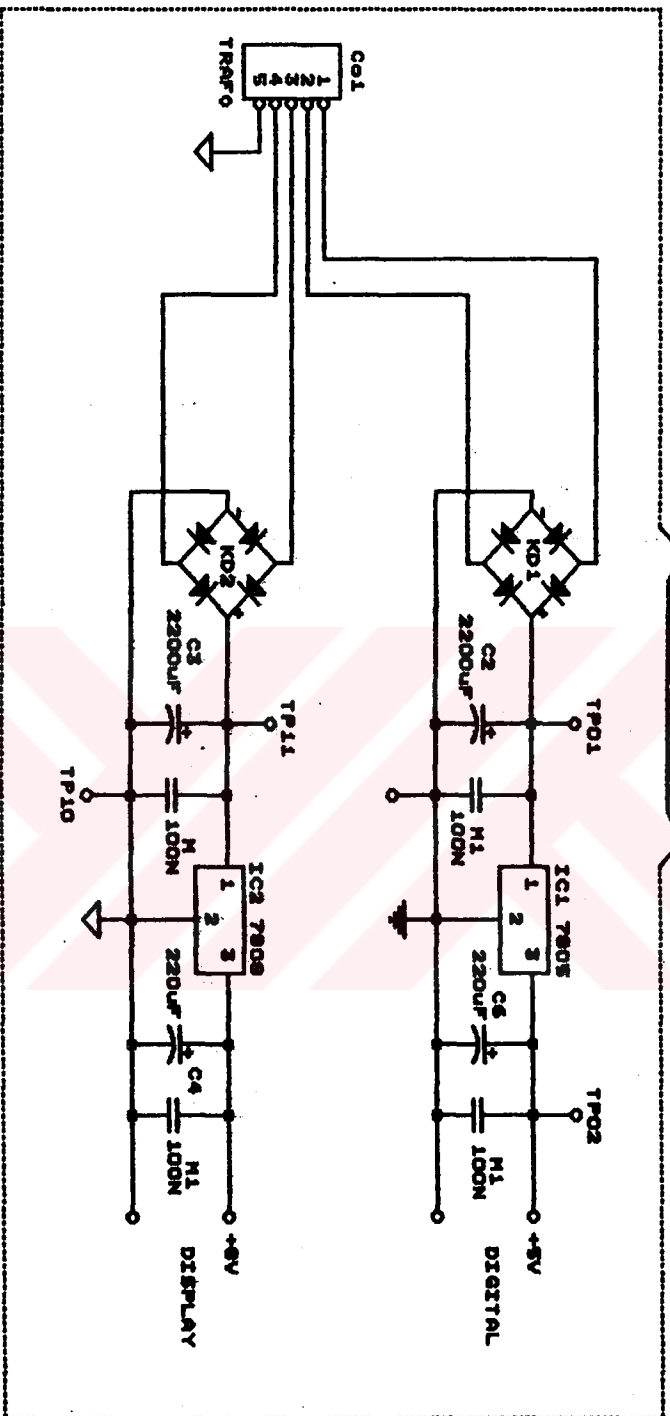
TIMER_0 KESMESİ İLE
DISPLAY SURULMESİ



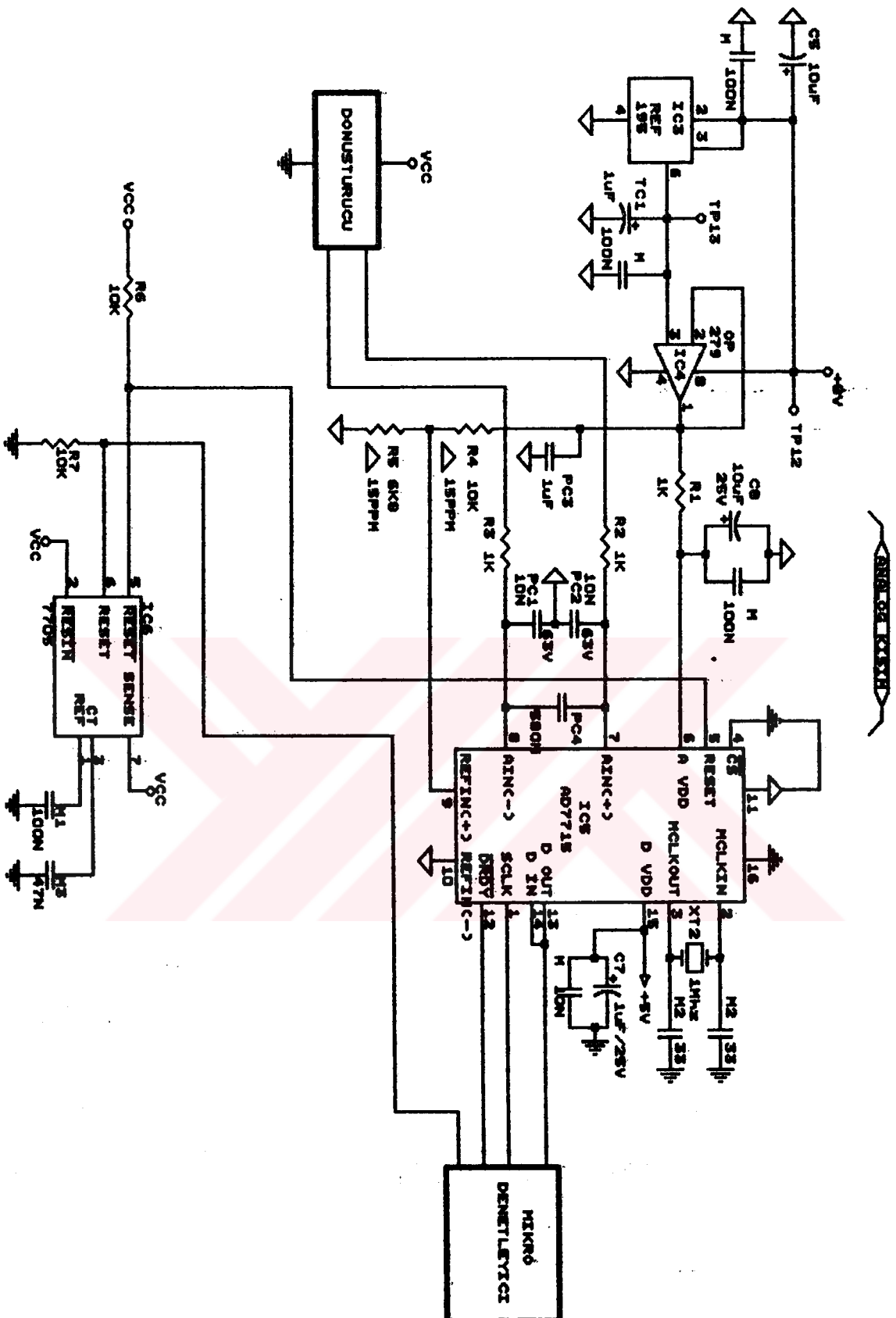
Bir sonraki kolon adresi =Tablo başlangıç adresi+(2*i)

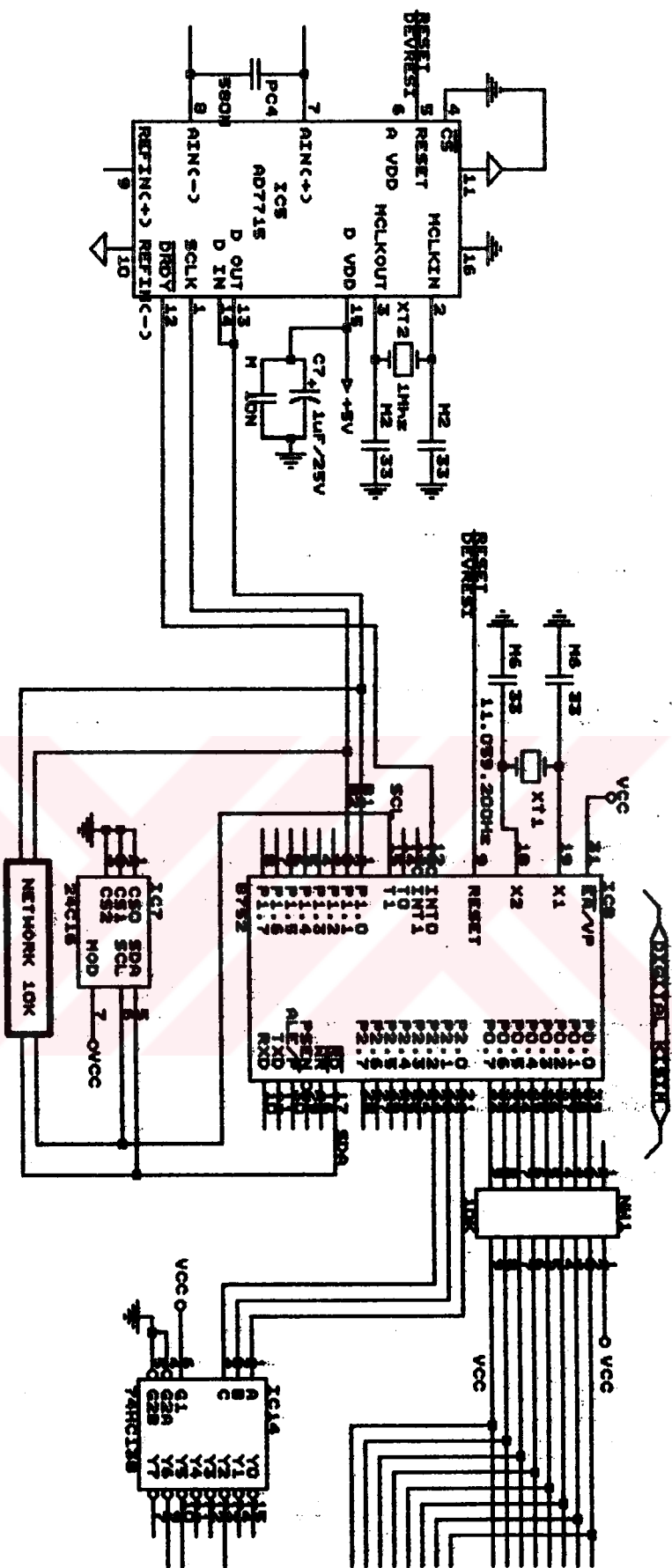
TIMER_0 KESME HİZMET PROGRAMI
DISPLAY ve KEYBOARD TARAMA

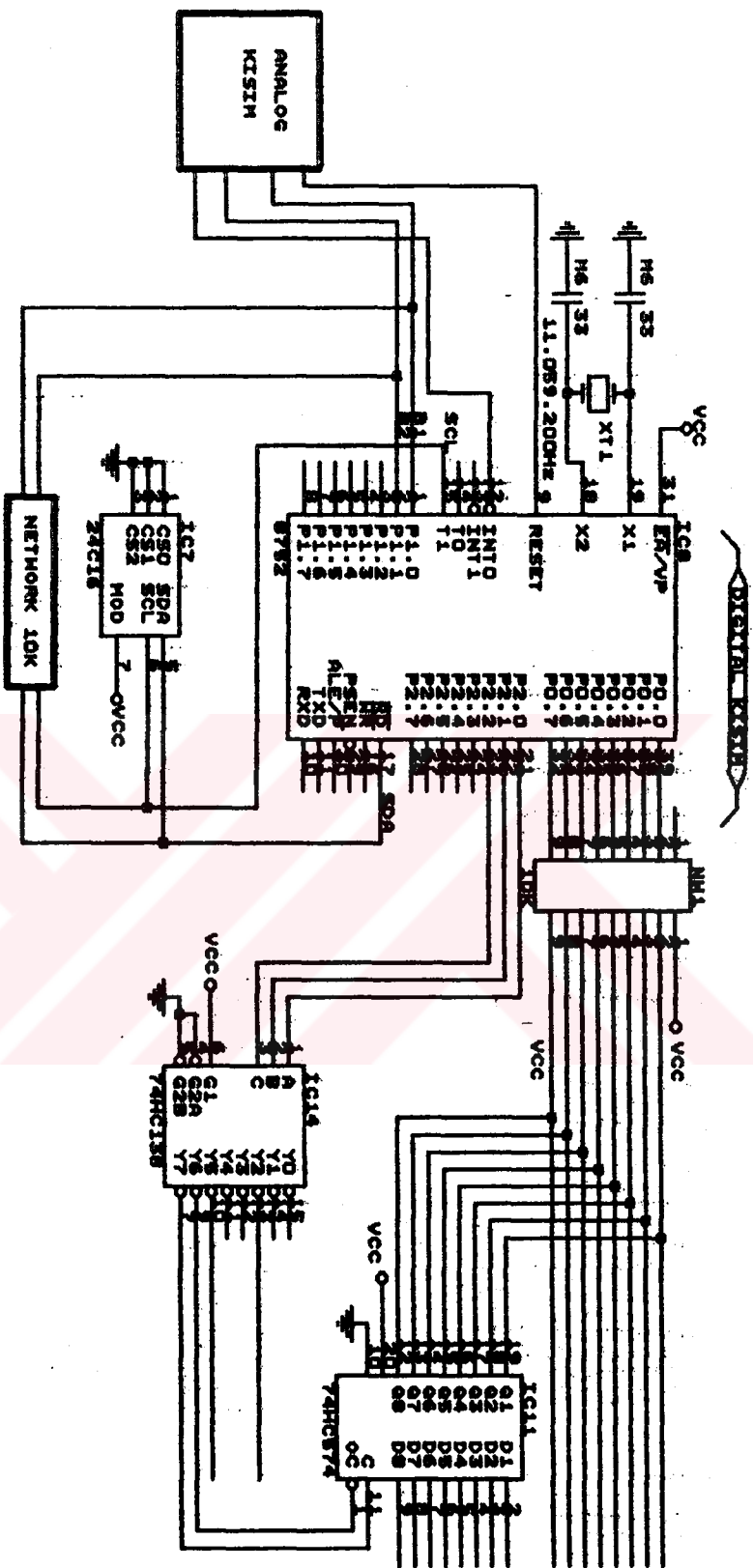




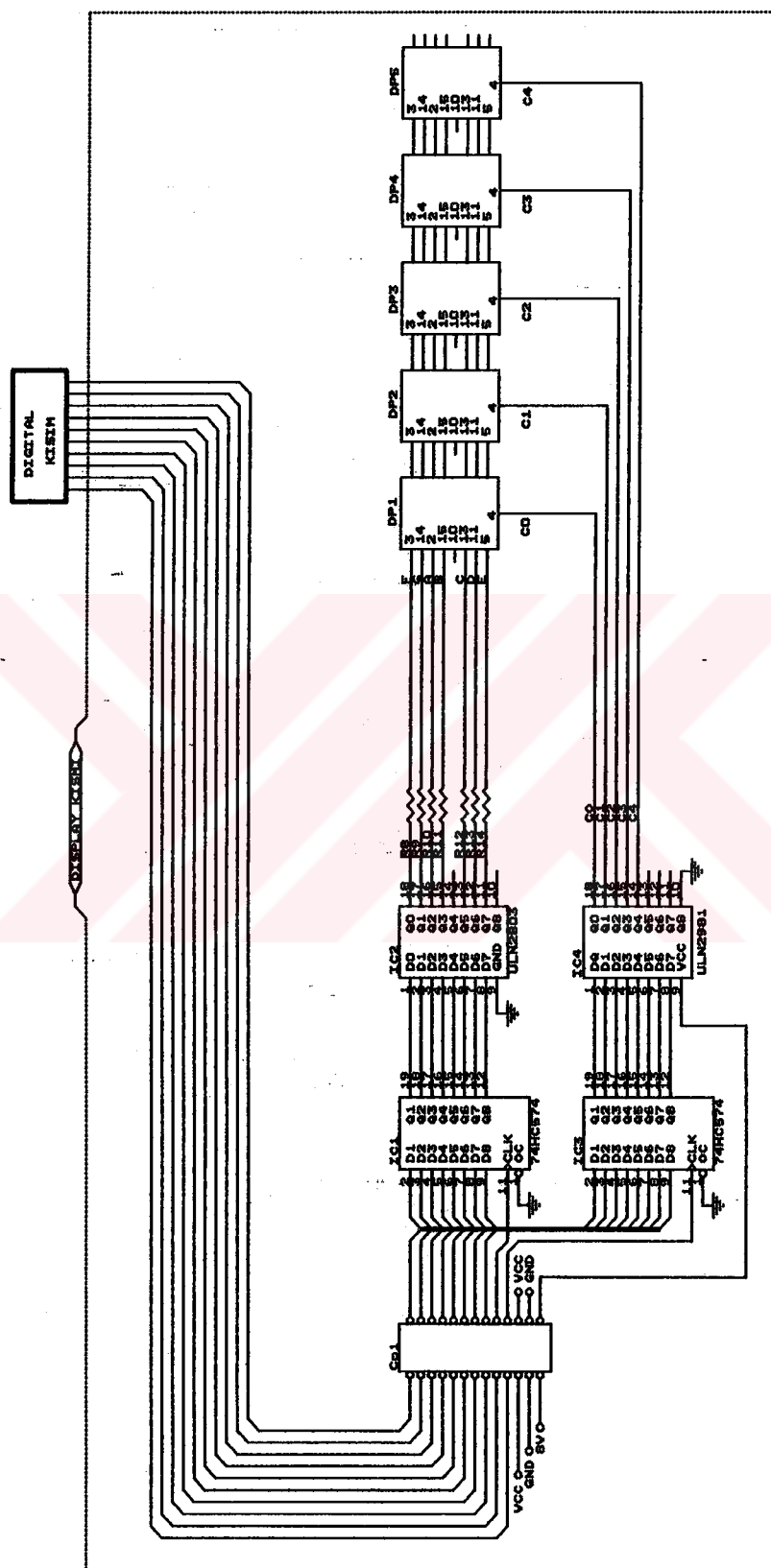
60







63



```

;*****
;MEMORY MAP
;*****

```

```

;NAME      ;ADDRESS  AÇIKLAMALAR
;_____

```

```

DSEG AT 09H

```

```

NET1:      DS  2
NET2:      DS  2
NET3:      DS  2
NET4:      DS  2
NET5:      DS  2
NET6:      DS  2
NET11:     DS  2
SNET:      DS  2
YUTPC:     DS  1
YUTNC:     DS  1
MEMO:      DS  4
MEMOC:     DS  1

```

```

DSEG AT 20H

```

```

FLAG:      DS  1
FLAG2:     DS  1
FLAG3:     DS  1
FLAG4:     DS  1
FLAG5:     DS  1
FLAG6:     DS  1
FLAG7:     DS  1
FLAG8:     DS  1
FLAG9:     DS  1
mflag:     DS  1
TRANS:     DS  3
DBUF0:     DS  8
DBUF1:     DS  8
DBUF2:     DS  8
COUNT:    DS  1
NEGA_COUNT: DS  1
COLN:      DS  1
YTUS:      DS  1
ETUS:      DS  1
TUSC:      DS  1
GTUS:      DS  1
GELENG:    DS  2
DARA:      DS  2
EKDARA:    DS  2
VGRAM:     DS  3
VADET:     DS  2
BOS:       DS  7
BOS4:      DS  9
BOS2:      DS  2
BOS5:      DS  1
BOS3:      DS  6
S_YUT_PC:  DS  1
S_YUT_NC:  DS  1
NET6:      DS  2
NET7:      DS  2
NET8:      DS  2
NET9:      DS  2

```

```

;*****
;YENI DISPLAY DEĞİSKENLERİ
;*****

```

```

GRAM       EQU  DBUF0+3
aa         EQU  04h
bb         EQU  08h
cc         EQU  20h

```

```

dd         EQU  40h
ee         EQU  80h

ff         EQU  01h
gg         EQU  02h
hh         EQU  10h
STACK      EQU  215
DEVSEL     EQU  BOS3+2
NOFBYT     EQU  BOS3+3
ESTAD      EQU  BOS3+4
ISTAD      EQU  BOS3+5
EPROM1     EQU  0A0H ; EEPROM
;ilk256byte(DEVSEL)
EPROM2     EQU  0A2H ; EEPROM ikinci
;256 byte(DEVSEL)
EPROM3     EQU  0A4H ; EEPROM ucuncu
;256 byte(DEVSEL)
EPROM4     EQU  0A6H ; EEPROM 4.
;256 byte(DEVSEL)
EPROM5     EQU  0A8H ; EEPROM 5.
;256 byte(DEVSEL)
EPROM6     EQU  0AAH ; EEPROM 6.
;256 byte(DEVSEL)
EPROM7     EQU  0ACH ; EEPROM 7.
;256 byte(DEVSEL)
EPROM8     EQU  0AEH ; EEPROM 8.
;256 byte(DEVSEL)
KAL_DEVSEL EQU  EPROM8 ;Kalibrasyon
;EPROM adresi
KAL_ESTAD  EQU  240d ;Kalibrasyon
;EPROM adresi
KAL_ISTAD  EQU  VGRAM ;Kalibrasyon
;ic RAM adresi
KAL_NOFBYT EQU  7d ; Kalibrasyonda
;okunacak Byt sayısı
SETUP1     EQU  05h ; 0000 0101
SETUP2     EQU  04h ; 0000 0100
SETUP3     EQU  44h ; 0100 0100

DATA_COM   EQU  3Bh ; 0011 1011
SETUP_COM  EQU  13h ; 0001 0011

```

```

;****Port Pin Bit Tanımlamaları

```

```

SCL        BIT  P3.5 ; T1 TIMER1 Clock ucu
SDA        BIT  P3.7 ; READ STROBU
;Veri ucu 1. eeprom
SDA2       BIT  P2.7 ; Veri ucu 2. eeprom
SCLK       BIT  P1.1 ; T2EX TIMER2 Clock
;AD 7715 için
DOUT       BIT  P1.0 ; T2 TIMER2 Veri ucu
;AD 7715 için
DRDY       BIT  P3.2 ; INT0
POLARPIN   BIT  P1.7 ;

```

```

;*****
;BIT TANIMLAMALARI
;*****

```

```

EKSILED    BIT  DBUF0+0.0
PROGLED    BIT  DBUF0+0.1
AGIRF      BIT  FLAG.1 ; Ağırlık
;Okundu Bayrağı

```

```

TUSF      BIT    FLAG.2      ; Tuş basıldı
;bayrağı
KRED      BIT    FLAG.3      ;Tuş takımı tarandı
;bayrağı
OTURDUF   BIT    FLAG.4
OTURDU2F  BIT    FLAG.5
CAR2      BIT    FLAG.6      ; CARRY 2
YENIGRAMF BIT    FLAG.7
TUSKALF   BIT    FLAG2.5
UYGUNDURF BIT    FLAG2.6
POLARITYF BIT    FLAG2.7
KALF      BIT    FLAG4.3
DISPF     BIT    FLAG4.4
AGIRGOF   BIT    FLAG4.5
MEMORYF   BIT    FLAG5.3      ; 1 Bit
CHECKF    BIT    FLAG5.4      ; 1 Bit
POZAGC    BIT    FLAG5.6      ; 1 Bit
SIFIRTF   BIT    FLAG7.0      ; 1 Bit
BIRTF     BIT    FLAG7.1      ; 1 Bit
IKITF     BIT    FLAG7.2      ; 1 Bit
UCTF      BIT    FLAG7.3      ; 1 Bit
DORTTF    BIT    FLAG7.4      ; 1 Bit
BESTF     BIT    FLAG7.5      ; 1 Bit
ALTTTF    BIT    FLAG7.6      ; 1 Bit
YEDITF    BIT    FLAG7.7      ; 1 Bit

SEKIZTF   BIT    FLAG8.0      ; 1 Bit ;
DOKUZTF   BIT    FLAG8.1      ; 1 Bit ;
HHATA     BIT    FLAG8.4      ; Haberlesme hatasi
EPROM2F   BIT    FLAG8.5      ; IKINCI EPROM
kalibredef BIT    FLAG8.6      ; 1 Bit ;
CLEARF    BIT    FLAG9.3      ; CLEAR TUSU
BASILDI
BOSF      BIT    FLAG9.4      ; 1 Bit ;

DB 0      ; 0Bh
DB 0      ; 12
DB 0      ; 3
DB 0      ; 4
DB 0      ; 5
DB 0      ; 6
DB 0      ; 7
DB 0      ; 8
DB 0      ; 9
DB 0      ; 20
DB 0      ; 21
DB 0      ; 2
DB 0      ; 3
DB 0      ; 4
DB 0      ; 5
DB 0      ; 6
DB 0      ; 7
DB 0      ; 8
DB 0      ; 9
DB 0      ; 30
DB 0      ; 31
DB 0      ; 2
DB 0      ; 3
DB 0      ; 4
DB 0      ; 5
DB 0      ; 6
DB 0      ; 7
DB 0      ; 8
DB 0      ; 9
DB 0      ; 40
DB 0      ; 41
DB 0      ; 2
DB 0      ; 3
DB gg+cc      ; " 4
DB 0      ; 5
DB gg      ; " 6
DB 0      ; 47
DB aa+bb+cc+dd+ee+ff ; "0" 48
DB bb+cc      ; "1" 9
DB aa+bb+gg+cc+dd ; "2" 50
DB aa+bb+gg+cc+dd ; "3" 1
DB ff+gg+bb+cc   ; "4" 2
DB aa+ff+gg+cc+dd ; "5" 3
DB ff+ee+cc+dd+gg ; "6" 4
DB aa+bb+cc      ; "7" 5
DB aa+bb+cc+dd+ee+ff+gg ; "8" 6
DB aa+bb+cc+dd+gg+ff ; "9" 57
DB 0      ; 58
DB 0      ; 9
DB 0      ; 60
;*****
; ** ASCII TO DISPLAY
; ** ASCII degerlerini ekrana dogru cikmasini saglar.
;*****
ASCII_TO_DISP: DB aa+bb+cc+dd+ee+ff ; "0"
DB bb+cc      ; "1"
DB aa+bb+gg+cc+dd ; "2"
DB aa+bb+gg+cc+dd ; "3"
DB ff+gg+bb+cc   ; "4"
DB aa+ff+gg+cc+dd ; "5"
DB ff+ee+cc+dd+gg ; "6"
DB aa+bb+cc      ; "7"
DB aa+bb+cc+dd+ee+ff+gg ; "8"
DB aa+bb+cc+dd+gg+ff ; "9"
DB 0      ;

DB ee+ff+aa+bb+cc+gg ; "A" 65
DB ff+ee+dd+cc+gg    ; "B" 6
DB gg+cc+dd          ; "C" 7
DB gg+ee+dd+cc+bb    ; "D" 8
DB aa+ff+ee+dd+gg    ; "E" 9
DB aa+ff+ee+gg        ; "F" 70
DB aa+ff+ee+dd+cc+gg ; "G" 1
DB ff+ee+bb+cc+gg    ; "H" 2
DB ff+ee             ; "I" 3

```

```

DB bb+cc+dd      ;"J" 4
DB aa+cc+gg+ff+ee ;"K" 5
DB ff+ee+dd      ;"L" 6
DB ee+ff+aa+bb+cc ;"M" 7
DB cc+ee+gg      ;"N" 8
DB ee+dd+cc+gg   ;"O" 9
DB ff+ee+aa+bb+gg ;"P80
DB aa+bb+cc+gg+ff ;"Q1
DB ee+gg         ;"R" 2
DB aa+ff+gg+cc+dd ;"S3
DB ff+ee+dd+gg   ;"t"4
DB cc+dd+ee      ;"U5
DB bb+cc+dd+ee   ;"V"6
DB bb+cc+dd+ee   ;"W" 7
DB bb+cc+gg+ee   ;"X" 8
DB ff+gg+bb+cc+dd ;,9
DB aa+bb+dd+ee+gg ;"Z90

```

```

DB 0      ;" " 91
DB 0      ;" " 2
DB 0      ;" " 3
DB 0      ;" " 4
DB 0      ;" " 5
DB 0      ;" " 6

```

```

DB ee+ff+aa+bb+cc+gg ;"a" 97
DB ff+ee+dd+cc+gg    ;"b" 8
DB aa+ff+gg+dd        ;"c" 9
DB gg+ee+dd+cc+bb    ;"d" 100
DB aa+ff+ee+dd+gg     ;"e" 1
DB aa+ff+ee+gg         ;"f" 2
DB aa+ff+ee+dd+cc+gg  ;"g" 3
DB ff+ee+bb+cc+gg     ;"h" 4
DB ee                  ;"i" 5
DB bb+cc+dd           ;"j" 6
DB aa+cc+gg+ff+ee     ;"k" 7
DB ff+ee+dd           ;"l" 8
DB ee+ff+aa+bb+cc     ;"m" 9
DB cc+ee+gg           ;"n" 110
DB ee+dd+cc+gg+aa     ;"o" 1
DB ff+ee+aa+bb+gg     ;"p" 2
DB aa+bb+cc+gg+ff     ;"q" 3
DB ee+gg              ;"r" 4
DB aa+ff+gg+cc+dd     ;"s" 5
DB ff+ee+dd+gg        ;"t" 6
DB cc+dd+ee+aa        ;"u" 7
DB bb+cc+dd+ee        ;"v" 8
DB 0                   ;"w" 9
DB 0                   ;"x" 120

```

```

DB 0      ;"y" 1
DB 0      ;"z" 2
DB 0      ;"{" 3
DB 0      ;"}" 4
DB 0      ;"_" 5
DB 0      ;" " 6
DB 0      ;" " 7
DB aa+ff+gg+dd        ;" "
DB 0      ;" " 9
DB 0      ;" " 130

```

```

DB 0      ;" " 1
DB 0      ;" " 2
DB 0      ;" " 3
DB 0      ;" " 4
DB 0      ;" " 5
DB 0      ;" " 6
DB 0      ;" " 7
DB 0      ;" " 8
DB 0      ;" " 9

```

```

DB 0      ;" " 140
DB 0      ;" " 1
DB 0      ;" " 2
DB 0      ;" " 3
DB 0      ;" " 4
DB 0      ;" " 5
DB 0      ;" " 6
DB 0      ;" " 7
DB 0      ;" " 8
DB 0      ;" " 9
DB 0      ;" " 150

```

```

DB 0      ;" " 1
DB 0      ;" " 2
DB ee+dd+cc+gg+aa    ;" " 153

```

OU

```

DB ee+dd+cc+aa      ;" " 154 U
DB 0                 ;" " 5
DB 0                 ;" " 6
DB 0                 ;" " 7
DB 0                 ;" " 8
DB 0                 ;" " 9
DB 0                 ;" " 160

```

```

DB 0      ;" " 1
DB 0      ;" " 2
DB 0      ;" " 3
DB 0      ;" " 4
DB 0      ;" " 5
DB 0      ;" " 6
DB 0      ;" " 7
DB 0      ;" " 8
DB 0      ;" " 9
DB 0      ;" " 170

```

```

DB 0      ;" " 1
DB 0      ;" " 2
DB 0      ;" " 3
DB 0      ;" " 4
DB 0      ;" " 5
DB aa+ff+ee+dd+cc+gg ;" " 176

```

Soft G

```

DB ee      ;" " 177 i
DB aa+ff+gg+cc+dd ;" " 178

```

she

```

DB 0      ;" " 9
DB 0      ;" " 180

```

;***** END OF ASCII TO DISP *****

```

; <<<<*****
; <<<< CONVERT TABLE
; <<<< RAKAMLARI ASCII KODUNA CEVIRIR
; <<<<*****

```

CONV_TABLE: DB '0' ;0

```

DB '1' ;1
DB '2' ;2
DB '3' ;3
DB '4' ;4
DB '5' ;5
DB '6' ;6
DB '7' ;7
DB '8' ;8
DB '9' ;9
DB ' ' ;Ah
DB ' ' ;Bh
DB ' ' ;Ch
DB ' ' ;Dh

```

```
DB " ;Eh
DB " ;Fh
```

```
***** END OF CONVERT TABLE*****
; <<< HAFIZA TABLE
; <<< Gecerli Tus numarasini hafiza numarasina ;cevirir.
; <<< NOT: 60d Hafizada olmayan tus anlamina gelir.
; <<< *****
```

```
HAFIZ_TABLE: DB 60d ; 0
DB 34d ; 1
DB 24d
DB 14d
DB 04d
DB 60d ; 5
DB 60d
DB 60d
DB 60d
DB 35d
DB 25d ; 10
DB 15d ; 11
DB 05d
DB 60d
DB 60d
DB 60d ; 15
DB 60d
DB 36d
DB 26d
DB 16d
DB 06d ; 20
DB 60d ; 21
DB 60d
DB 60d
DB 60d
DB 37d ; 25
DB 27d
DB 17d
DB 07d
DB 60d
DB 60d ; 30
DB 60d ; 31
DB 60d
DB 33d
DB 23d
DB 13d ; 35
DB 03d
DB 40d
DB 30d
DB 20d
DB 10d ; 40
DB 32d ; 41
DB 22d
DB 12d
DB 02d
DB 39d ; 45
DB 29d
DB 19d
DB 09d
DB 31d
DB 21d ; 50
```

```
DB 11d ; 51
DB 01d
DB 38d
DB 28d
DB 18d ; 55
DB 08d
DB 60d
DB 60d
DB 60d
DB 60d ; 60
```

H_TABLE_SON:

```
***** END OF HAFIZA TABLE *****
```

```
; <<< *****
; <<< HARF TABLE
; <<< Gecerli Tus numarasini ASCII Koduna cevirir.
; <<< *****
```

```
HARF_TABLE: DB ' ' ; ; 0
DB 'Y' ; ; 1
DB 'R' ;
DB 'J' ;
DB 128d ; CHE
DB ' ' ; ; 5
DB ' ' ;
DB ' ' ;
DB ' ' ;
DB 'Z' ;
DB 'S' ; ; 10
DB 'K' ; ; ; 11
DB 'D' ;
DB ' ' ;
DB '3' ;
DB '6' ; ; 15
DB '9' ;
DB '!' ; ;
DB 178d ; SHE
DB 'L' ;
DB 'E' ; ; 20
DB '0' ; ; ; 21
DB '2' ;
DB '5' ;
DB '8' ;
DB '!' ; ; 25
DB 'T' ;
DB 'M' ;
DB 'F' ; ;
DB ' ' ; ;
DB 'I' ; ; 30
DB '4' ; ; 31
DB '7' ;
DB 'X' ;
DB 'Q' ;
DB 177d ; i ; 35
DB 'C' ;
DB ' ' ;
DB ' ' ;
DB ' ' ;
DB ' ' ; ; 40
DB 'W' ; ; 41
DB 'P' ;
DB 'T' ;
DB 'B' ;
DB ' ' ; ; 45
```

```

DB 154d ; UO
DB 'O' ;
DB 176d ; SOFT G
DB 'V' ;
DB 153d ; OU ; 50

DB 'H' ; ; 51
DB 'A' ; ;
DB '' ; ;
DB 'U' ; ;
DB 'N' ; ; 55
DB 'G' ; ;
DB '100' ; ;
DB '100' ; ;
DB '100' ; ;
DB '100' ; ; 60

HARF_T SON:
;***** END OF HARF TABLE *****

;*****
;Bu program parçası A'da verilen veriyi ADC'ye yazar
;*****

C_YAZADC:
    setb SCLK
    mov B,#8
C_YAZL1:
    clr SCLK
    NOP
    rlc a
    mov DOUT,c ; Veriyi cikisa yaz
    setb SCLK
    djnz b,C_YAZL1
    setb DOUT ; Veri hattini bosalt
    ret

;*****
;Bu program ADC'de verilen veriyi A2ya yazar.
;*****

C_OKUAD8:
    setb SCLK ;
    setb DOUT ; Veri yolunu bosalt
    mov B,#8
    jb DRDY,$ ; Yazmaya hazır değilse bekle
C_OKUL18:
    clr SCLK
    NOP
    mov C,DOUT ; Veriyi oku
    rlc a
    setb SCLK
    djnz b,C_OKUL18
    ret

;*****
;Bu program parçası ADC'den okunan 16bit veriyi
;NET1+0,1'e yazar.
;*****

C_OKUAD16:
    mov a,#DATA_COM ;DATA_COM:0011 1011
    call C_YAZADC
    call C_OKUAD8 ;8 bit bilgi okunuyor
    mov GELENG+1,a ;Okunan 8 bit bilgi GELENG+1
    call C_OKUAD8 ;
    mov GELENG+0,a
    ret

```

```

;*****
;Bu program parçası ADC on kosullama islemlerini yapar |
;*****
C_KOSUL:

    mov a,#SETUP_COM ;13h 0001 0011 .
    call C_YAZADC ;ADC
    mov a,#SETUP1 ;05h 0000 0101
    call C_YAZADC
    CALL DELAY2

    mov a,#SETUP_COM
    call C_YAZADC
    mov a,#SETUP2
    call C_YAZADC

    CALL DELAY2

    mov a,#SETUP_COM
    call C_YAZADC
    mov a,#SETUP3
    call C_YAZADC

    CALL DELAY2 ; SELF
KALIBRASYONU BEKLE

    mov a,#DATA_COM
    call C_YAZADC
    ret

;*****
;AGIRLIK
;*****
AGIRLIK:    PUSH PSW
            PUSH ACC
            PUSH B

            CALL C_OKUAD16

            SETB AGIRF

            POP B
            POP ACC
            POP PSW
            RETI

;
END OF AGIRLIK

;*****
;TIMER 1 INTERRUPT
;*****
TIMER_1:    ;NOP
            RETI

;END OF TIMER_1

;*****
;DELAY 1
;*****
DELAY:      MOV COUNT,#250D
DELAY_1:    NOP

            NOP
            NOP
            DJNZ COUNT,DELAY_1
            RET

END OF DELAY1
;*****
;DELAY 2
;*****
DELAY2:     PUSH AR5

```

```

                MOV R5,#100D
DELAY2_2:  MOV COUNT,#255D
DELAY2_3:  NOP
                NOP
                NOP
                DJNZ COUNT,DELAY2_3
                DJNZ R5,DELAY2_2
                POP AR5
                RET

```

```

; <<<***** END OF DELAY2
;

```

```

<<<*****
; <<<      CHECK_LED
; <<<*****

```

```

CHECK_LED:  CLR AGIRGOF
                MOV B,#9d
OTHER_LED:  mov gram+4,#0bh
                mov gram+3,#'Y'
                MOV GRAM+2,#'T'
                MOV GRAM+1,#'U'
                MOV GRAM+0,#0BH

```

```

                MOV R1,#100D
CHECK_LED1: JNB TUSF,CHECK_LED1A
                CLR TUSF
                MOV MEMO+0,GTUS

```

```

CHECK_LED1A: JNB DISPF,CHECK_LED1
                CLR DISPF

```

```

CHECK_LED1B: JNB AGIRF,CHECK_LED4
                CLR AGIRGOF
                CALL AGIR_DEG
                SETB SIFIRLED
                SETB HATALED

```

```

CHECK_LED6: MOV DBUF0+0,#0H
                MOV DBUF0+1,#0H
                SETB AGIRGOF

```

```

                MOV A,MEMO+0
                CJNE A,#5H,CHECK_LED7 ;STAR
TUSUNA BASILIRSA K'YE GIR

```

```

                SETB TUSKALF
                JMP CHECK_LED_SON

```

```

CHECK_LED7: CLR TUSKALF
CHECK_LED_SON: RET

```

```

; <<<***** END OF CHECK_LED
*****>>>

```

```

; -----
; Bu program parcasi saat veya EEPROM'dan okuma islemini |
; yapar. Haberlesmenin kiminle oldugu DEVSEL, Okumanin |
; baslangic ardesi ESTAD, okunacak byte sayisi NOFBYT, |
; alinan verilerin ic RAM'de yazilacagi baslangic adresi |
; ISTAD saklayicilarinda vermek gerekir. |
; -----

```

```

OKURAM:

```

```

                SETB SCL
                SETB SDA ; Veri yolunu bosalt
                JNB SDA,$ ; Veri yolu dolu ise bekle
                CLR SDA ; Start bit
                MOV A,DEVSEL ; Okunacak eleman adresi
                CLR HHATA ; Haberlesme hatasini sil
                LCALL GON ; Gonder

```

```

                MOV A,ESTAD ; Okunacak bolge
baslangic adresi
                LCALL GON ; Gonder
                SETB SDA ; Veri yolunu bosalt
                SETB SCL
                JNB SDA,$ ; Veri yolu bosalmadiysa

```

```

bekle
                CLR SDA ; Start bit
                MOV A,DEVSEL ; Okunacak eleman adresi
                SETB ACC.0 ; Oku komutu
                LCALL GON ; Gonder
                MOV B,NOFBYT ; Okunacak byte sayisini

```

```

yukle
                MOV R0,ISTAD ; Okunacak verinin
saklanacagi yer
OKUL:

```

```

                LCALL OKU ; Oku
                MOV @R0,A ; Sakla
                INC R0 ; Bir sonraki saklama yerini

```

```

goster
                DJNZ B,OKUL1 ; Okuma bitmediyse don
                SETB SDA
                SETB SCL
                LCALL CLBEK

```

```

OKUS:
                CLR SCL ; NoAcknowledge
                CLR SDA
                LCALL CLBEK
                SETB SCL
                LCALL CLBEK
                SETB SDA ; Stop
                RET

```

```

OKUL1:
                CLR SDA
                SETB SCL
                LCALL CLBEK
                CLR SCL ; Acknowledge
                SETB SDA
                JMP OKUL

```

```

; -----
; Bu program parcasi saat veya EEPROM'a 8bit veri |
; gonderir ve anlasildi bitini kontrol eder. Anlasildi |
; alamazsa haberlesme hatasi biti HHATA'yi 1 yapar |
; Gonderilecek bilgi programa A'da verilmelidir |
; -----

```

```

GON:

```

```

                PUSH B
                MOV B,#9
                SETB C

```

```

GONL:

```

```

                CLR SCL
                RLC A
                MOV SDA,C
                LCALL CLBEK ; Tch
                SETB SCL
                LCALL CLBEK ; Tchcl
                DJNZ B,GONL
                JB SDA,GON1 ; Henuz anlasilmadi ise

```

```

GONBE:

```

```

                CLR SCL

```

```

GONS:

```

```

                POP B
                RET

```

```

GON1:

```

```

                LCALL CLBEK
                JNB SDA,GONBE ; Anlasildi
                SETB HHATA ; Hala anlasilmadi ise hata

```

```

var

```

```

                SJMP GONBE

```

OKU:

OKUD1:

YAZRAM:

baslangic adresi

yukle

saklandigi yer

YAZL:

YAZS:

CLBEK: NOP

.....

GRAM ACC 1: JNC GRAM ACC HATA

GRAM ACC HATA; SETB HAF_500F

GRAM ACC SON: POP AR5

RET

```

; <<<*****      END      OF      GRAM      TO      ACC
*****>>>

```

```

; <<<*****      END      OF      HANGI      EPROM
*****>>>

```

```

; <<<*****
****>>>

```

: <<< FIND BIGGEST

***FIND BIGGEST

```
FIND_BIGGEST: MOV  BOS+0,NET10+0
               MOV  BOS+1,NET10+1
               MOV  BOS+3,#2D
               MOV  BOS+2,#0D
```

```

;***COMPARE WITH NET9

```

```
COMP_NET9:    MOV     A,BOS+1
              CJNE    A,NET9+1,MMAX9
              MOV     A,BOS+0
              CJNE    A,NET9+0,MMAX9
```

```

;JMP    COMP NET8

```

```
MMAX9:      JNC  COMP_NET8
UP_MAX9:    MOV  BOS+0,NET9+0
            MOV  BOS+1,NET9+1
            MOV  BOS+3,#1D
            MOV  BOS+2,#0D
```

```

;***COMPARE WITH NET8

```

```
COMP_NET8:    MOV     A,BOS+1
              CJNE    A,NET8+1,MMAX8
              MOV     A,BOS+0
              CJNE    A,NET8+0,MMAX8
```

```

:JMP    COMP NET7

```

```
MMAX8:      JNC  COMP_NET7
UP_MAX8:    MOV  BOS+0,NET8+0
            MOV  BOS+1,NET8+1
            MOV  BOS+2,#128D
            MOV  BOS+3,#0D
```

```

;***COMPARE WITH NET7
COMP_NET7:  MOV  A,BOS+1
            CJNE A,NET7+1,MMAX7
            MOV  A,BOS+0
            CJNE A,NET7+0,MMAX7
            JMP  COMP_NET6
MMAX7:      JNC  COMP_NET6
UP_MAX7:    MOV  BOS+0,NET7+0
            MOV  BOS+1,NET7+1
            MOV  BOS+2,#64D
            MOV  BOS+3,#0D

```

```

;***COMPARE WITH NET6
COMP_NET6:  MOV  A,BOS+1
            CJNE A,NET6+1,MMAX6
            MOV  A,BOS+0
            CJNE A,NET6+0,MMAX6
            JMP  COMP_NET5
MMAX6:      JNC  COMP_NET5
UP_MAX6:    MOV  BOS+0,NET6+0
            MOV  BOS+1,NET6+1
            MOV  BOS+2,#32D
            MOV  BOS+3,#0D

```

```

;***COMPARE WITH NET5
COMP_NET5:  MOV  A,BOS+1
            CJNE A,NET5+1,MMAX5
            MOV  A,BOS+0
            CJNE A,NET5+0,MMAX5
            JMP  COMP_NET4
MMAX5:      JNC  COMP_NET4
UP_MAX5:    MOV  BOS+0,NET5+0
            MOV  BOS+1,NET5+1
            MOV  BOS+2,#16D
            MOV  BOS+3,#0D

```

```

;***COMPARE WITH NET4
COMP_NET4:  MOV  A,BOS+1
            CJNE A,NET4+1,MMAX4
            MOV  A,BOS+0
            CJNE A,NET4+0,MMAX4
            JMP  COMP_NET3
MMAX4:      JNC  COMP_NET3
UP_MAX4:    MOV  BOS+0,NET4+0
            MOV  BOS+1,NET4+1
            MOV  BOS+2,#8D
            MOV  BOS+3,#0D

```

```

;***COMPARE WITH NET3
COMP_NET3:  MOV  A,BOS+1
            CJNE A,NET3+1,MMAX3
            MOV  A,BOS+0
            CJNE A,NET3+0,MMAX3
            JMP  COMP_NET2
MMAX3:      JNC  COMP_NET2
UP_MAX3:    MOV  BOS+0,NET3+0
            MOV  BOS+1,NET3+1
            MOV  BOS+2,#4D
            MOV  BOS+3,#0D

```

```

;***COMPARE WITH NET2
COMP_NET2:  MOV  A,BOS+1
            CJNE A,NET2+1,MMAX2
            MOV  A,BOS+0
            CJNE A,NET2+0,MMAX2
            JMP  COMP_NET1
MMAX2:      JNC  COMP_NET1

```

```

UP_MAX2:    MOV  BOS+0,NET2+0
            MOV  BOS+1,NET2+1
            MOV  BOS+2,#2D
            MOV  BOS+3,#0D

```

```

;***COMPARE WITH NET1
COMP_NET1:  MOV  A,BOS+1
            CJNE A,NET1+1,MMAX1
            MOV  A,BOS+0
            CJNE A,NET1+0,MMAX1
            JMP  FIND_BIG_SON
MMAX1:      JNC  FIND_BIG_SON
UP_MAX1:    MOV  BOS+0,NET1+0
            MOV  BOS+1,NET1+1
            MOV  BOS+2,#1D
            MOV  BOS+3,#0D

```

```

FIND_BIG_SON: RET
;***** END OF FIND BIGGEST *****

```

```

;.<<<*****
;.<<<      AGIRLIK DEGERLENDIR
;.<<<*****

```

```

AGIR_DEG:   PUSH  B
            CLR   AGIRF
            MOV   NET10+0,NET9+0
            MOV   NET10+1,NET9+1
            MOV   NET9+0,NET8+0
            MOV   NET9+1,NET8+1
            MOV   NET8+0,NET7+0
            MOV   NET8+1,NET7+1
            MOV   NET7+0,NET6+0
            MOV   NET7+1,NET6+1
            MOV   NET6+0,NET5+0
            MOV   NET6+1,NET5+1
            MOV   NET5+0,NET4+0
            MOV   NET5+1,NET4+1
            MOV   NET4+0,NET3+0
            MOV   NET4+1,NET3+1
            MOV   NET3+0,NET2+0
            MOV   NET3+1,NET2+1
            MOV   NET2+0,NET1+0
            MOV   NET2+1,NET1+1
            MOV   NET1+0,GELENG+0
            MOV   NET1+1,GELENG+1
            CALL  FIND_BIGGEST

```

```

;***FIND SMALLEST
FIND_SMALL: MOV  BOS+0,NET10+0
            MOV  BOS+1,NET10+1
            MOV  BOS+5,#2D
            MOV  BOS+4,#0D

```

```

;***COMPARE WITH NET9
COMPS_NET9: MOV  A,BOS+1
            CJNE A,NET9+1,MMIN9
            MOV  A,BOS+0

```

```

                CJNE A,NET9+0,MMIN9
                ;JMP UP_MIN8
MMIN9:          JC COMPS_NET8
UP_MIN9:        MOV BOS+0,NET9+0
                MOV BOS+1,NET9+1
                MOV BOS+5,#1D
                MOV BOS+4,#0D

                ;***COMPARE WITH NET8
COMPS_NET8:     MOV A,BOS+1
                CJNE A,NET8+1,MMIN8
                MOV A,BOS+0
                CJNE A,NET8+0,MMIN8
                ;JMP UP_MIN7
MMIN8:          JC COMPS_NET7
UP_MIN8:        MOV BOS+0,NET8+0
                MOV BOS+1,NET8+1
                MOV BOS+4,#128D
                MOV BOS+5,#0D

                ;***COMPARE WITH NET7
COMPS_NET7:     MOV A,BOS+1
                CJNE A,NET7+1,MMIN7
                MOV A,BOS+0
                CJNE A,NET7+0,MMIN7
                ;JMP UP_MIN6
MMIN7:          JC COMPS_NET6
UP_MIN7:        MOV BOS+0,NET7+0
                MOV BOS+1,NET7+1
                MOV BOS+4,#64D
                MOV BOS+5,#0D

                ;***COMPARE WITH NET6
COMPS_NET6:     MOV A,BOS+1
                CJNE A,NET6+1,MMIN6
                MOV A,BOS+0
                CJNE A,NET6+0,MMIN6
                ;JMP UP_MIN5
MMIN6:          JC COMPS_NET5
UP_MIN6:        MOV BOS+0,NET6+0
                MOV BOS+1,NET6+1
                MOV BOS+4,#32D
                MOV BOS+5,#0D

                ;***COMPARE WITH NET5
COMPS_NET5:     MOV A,BOS+1
                CJNE A,NET5+1,MMIN5
                MOV A,BOS+0
                CJNE A,NET5+0,MMIN5
                ;JMP UP_MIN4
MMIN5:          JC COMPS_NET4
UP_MIN5:        MOV BOS+0,NET5+0
                MOV BOS+1,NET5+1
                MOV BOS+4,#16D
                MOV BOS+5,#0D

                ;***COMPARE WITH NET4
COMPS_NET4:     MOV A,BOS+1
                CJNE A,NET4+1,MMIN4
                MOV A,BOS+0
                CJNE A,NET4+0,MMIN4
                ;JMP UP_MIN4
MMIN4:          JC COMPS_NET3
UP_MIN4:        MOV BOS+0,NET4+0
                MOV BOS+1,NET4+1
                MOV BOS+4,#8D
                MOV BOS+5,#0D

                ;***COMPARE WITH NET3

```

```

COMPS_NET3:     MOV A,BOS+1
                CJNE A,NET3+1,MMIN3
                MOV A,BOS+0
                CJNE A,NET3+0,MMIN3
                ;JMP UP_MIN3
MMIN3:          JC COMPS_NET2
UP_MIN3:        MOV BOS+0,NET3+0
                MOV BOS+1,NET3+1
                MOV BOS+4,#4D
                MOV BOS+5,#0D

                ;***COMPARE WITH NET2
COMPS_NET2:     MOV A,BOS+1
                CJNE A,NET2+1,MMIN2
                MOV A,BOS+0
                CJNE A,NET2+0,MMIN2
                ;JMP UP_MIN2
MMIN2:          JC COMPS_NET1
UP_MIN2:        MOV BOS+0,NET2+0
                MOV BOS+1,NET2+1
                MOV BOS+4,#2D
                MOV BOS+5,#0D

                ;***COMPARE WITH NET1
COMPS_NET1:     MOV A,BOS+1
                CJNE A,NET1+1,MMIN1
                MOV A,BOS+0
                CJNE A,NET1+0,MMIN1
                ;JMP UP_MIN1
MMIN1:          JC BOS_2_FLAG
UP_MIN1:        MOV BOS+4,#1D
                MOV BOS+5,#0D

BOS_2_FLAG:     MOV A,BOS+2
                ORL A,BOS+4
                MOV FLAG3,A

                MOV A,BOS+3
                ORL A,BOS+5
                MOV BOS+5,A

                ; SUM = 0
                MOV BOS+0,#0H
                MOV BOS+1,#0H
                MOV BOS+2,#0H

                ;*** ADD NET10 TO BOS
ADD_NET10:      MOV A,BOS+5
                JB ACC.1,ADD_NET9
                MOV A,BOS+0
                ;
                ADD A,NET10+0
                MOV BOS+0,A

                MOV A,BOS+1
                ADDC A,NET10+1
                MOV BOS+1,A

                ; MOV A,BOS+2
                ; ADDC A,#0h
                ; MOV BOS+2,A

                ;*** ADD NET9 TO BOS
ADD_NET9:       MOV A,BOS+5
                JB ACC.0,ADD_NET8
                MOV A,BOS+0
                ;
                ADD A,NET9+0
                MOV BOS+0,A

```

```

MOV A,BOS+1
ADDC A,NET9+1
MOV BOS+1,A

MOV A,BOS+2
ADDC A,#0h
MOV BOS+2,A

;*** ADD NET8 TO BOS
ADD_NET8: JB FLAG3.7,ADD_NET7
MOV A,BOS+0
BOS=BOS+NET8
ADD A,NET8+0
MOV BOS+0,A

MOV A,BOS+1
ADDC A,NET8+1
MOV BOS+1,A

MOV A,BOS+2
ADDC A,#0h
MOV BOS+2,A

;*** ADD NET7 TO BOS
ADD_NET7: JB FLAG3.6,ADD_NET6
MOV A,BOS+0
BOS=BOS+NET7
ADD A,NET7+0
MOV BOS+0,A

MOV A,BOS+1
ADDC A,NET7+1
MOV BOS+1,A

MOV A,BOS+2
ADDC A,#0h
MOV BOS+2,A

;*** ADD NET6 TO BOS
ADD_NET6: JB FLAG3.5,ADD_NET5
MOV A,BOS+0
BOS=BOS+NET6
ADD A,NET6+0
MOV BOS+0,A

MOV A,BOS+1
ADDC A,NET6+1
MOV BOS+1,A

MOV A,BOS+2
ADDC A,#0h
MOV BOS+2,A

;*** ADD NET5 TO BOS
ADD_NET5: JB FLAG3.4,ADD_NET4
MOV A,BOS+0
BOS=BOS+NET5
ADD A,NET5+0
MOV BOS+0,A

MOV A,BOS+1
ADDC A,NET5+1
MOV BOS+1,A

MOV A,BOS+2
ADDC A,#0h
MOV BOS+2,A

;*** ADD NET4 TO BOS
ADD_NET4: JB FLAG3.3,ADD_NET3
MOV A,BOS+1
BOS=BOS+NET4
ADD A,NET4+0
MOV BOS+0,A

MOV A,BOS+1
ADDC A,NET4+1
MOV BOS+1,A

MOV A,BOS+2
ADDC A,#0h
MOV BOS+2,A

;*** ADD NET3 TO BOS
ADD_NET3: JB FLAG3.2,ADD_NET2
MOV A,BOS+0
BOS=BOS+NET3
ADD A,NET3+0
MOV BOS+0,A

MOV A,BOS+1
ADDC A,NET3+1
MOV BOS+1,A

MOV A,BOS+2
ADDC A,#0h
MOV BOS+2,A

;*** ADD NET2 TO BOS
ADD_NET2: JB FLAG3.1,ADD_NET1
MOV A,BOS+0
BOS=BOS+NET2
ADD A,NET2+0
MOV BOS+0,A

MOV A,BOS+1
ADDC A,NET2+1
MOV BOS+1,A

MOV A,BOS+2
ADDC A,#0h
MOV BOS+2,A

;*** ADD NET1 TO BOS
ADD_NET1: JB FLAG3.0,DIVIDE_FOUR
MOV A,BOS+0
BOS=BOS+NET1
ADD A,NET1+0
MOV BOS+0,A

MOV A,BOS+1
ADDC A,NET1+1
MOV BOS+1,A

MOV A,BOS+2
ADDC A,#0h
MOV BOS+2,A

;**** 3'e bol
DIVIDE_FOUR: CLR C
MOV A,BOS+2
RRC A
MOV BOS+2,A

MOV A,BOS+1
RRC A
MOV BOS+1,A

```

```

MOV A,BOS+0
RRC A
MOV BOS+0,A

CLR C          ;**** 2'ye bol
MOV A,BOS+2
RRC A
MOV BOS+2,A

MOV A,BOS+1
RRC A
MOV BOS+1,A

MOV A,BOS+0
RRC A
MOV BOS+0,A

CLR C          ;**** 2'ye bol
MOV A,BOS+2
RRC A
MOV BOS+2,A

MOV A,BOS+1
RRC A
MOV BOS+1,A

MOV A,BOS+0
RRC A
MOV BOS+0,A

MOV A,BOS+2
JZ AGIR_DEVAM
JMP AGIR_SON

AGIR_DEVAM: ;**** NET10'a BURUT AGIRLIGI YAZ
MOV NET10+0,BOS+0
MOV NET10+1,BOS+1

MOV sNET+0,BOS+0
MOV sNET+1,BOS+1

YENI_GRAM_2: MOV BOS+3,sNET+0
MOV BOS+4,sNET+1

CALL BIN_2_BCD

MOV R0,#(VGRAMADR+0)
MOV @R0,BOS+2
INC R0
MOV @R0,BOS+1
INC R0
MOV @R0,BOS+0

MOV A,BOS+2
ANL A,#00001111b
MOV BOS+4,A

MOV A,BOS+1
ANL A,#11110000b
SWAP A
MOV BOS+3,A

MOV A,BOS+1
ANL A,#00001111b
MOV BOS+2,A

MOV A,BOS+0

ANL A,#11110000b
SWAP A
MOV BOS+1,A

MOV A,BOS+0
ANL A,#00001111b
MOV BOS+0,A

GENEL_0: CALL SAYMA

GENEL_8: MOV GRAM+4,BOS+4
MOV GRAM+3,BOS+3
MOV GRAM+2,BOS+2
MOV GRAM+1,BOS+1
MOV GRAM+0,BOS+0

RET_AG_DEG: POP B
RET

END OF AGIRLIK DEGERLENDIR

;*****
;*** BIN TO BCD
;***<<<BOS+4,3'deki Binary sayiyi BOS+2,1,0'a BCD olarak
;yazar
;***<<<*****
BIN_2_BCD: MOV BOS+2,#0d
MOV BOS+1,#0d
MOV BOS+0,#0d

MOV B,#16d

BIN_2_BCD_1: MOV A,BOS+3
CLR C
RLC A
MOV BOS+3,A

MOV A,BOS+4
RLC A
MOV BOS+4,A

MOV A,BOS+0
ADDC A,BOS+0
DA A
MOV BOS+0,A

MOV A,BOS+1
ADDC A,BOS+1
DA A
MOV BOS+1,A

MOV A,BOS+2
ADDC A,BOS+2
DA A
MOV BOS+2,A

DJNZ B,BIN_2_BCD_1

BIN_2_BCD_SON: RET
;***** END OF BIN_2_BCD*****
;***<<<*****
;*** GRAM SIFIRLA
;***<<<*****
G_SIFIRLA:
G_SIF1: MOV GRAM+4,#0Bh
G_SIF3: MOV GRAM+3,#0h
G_SIF4: MOV GRAM+2,#0h
MOV GRAM+1,#0h

```

```

MOV GRAM+0,#0h
RET
; <<<***** END OF GRAM SIFIRLA *****

; <<<*****
; <<< GRAM KARART
; <<<*****
G_KARART: MOV GRAM+4,#0Bh
          MOV GRAM+3,#0Bh
          MOV GRAM+2,#0Bh
          MOV GRAM+1,#0Bh
          MOV GRAM+0,#0Bh
          MOV DBUF0+2,#0Bh
          RET
; <<<***** END OF GRAM karart *****

; <<<*****
; <<< DISPLAY & KEYBOARD
; <<<*****

TIMER_0: MOV TL0,#00H
          MOV TH0,#0FDH
          PUSH DPH
          PUSH DPL
          PUSH PSW
          PUSH ACC
          PUSH B
          PUSH AR1

KEY1: MOV P0,#0h
      ANL P2,#11111000b
      ORL P2,#2h
      ANL P2,#11111000b
      ANL P2,#11111000b
      MOV A,R3
      CPL A
      MOV P0,A
      ORL P2,#5h
      ANL P2,#11111000b
      MOV DPTR,#ASCII_TO_DISP
      MOV A,DBUF0+3
      MOVC A,@A+DPTR

KEY3: MOV P0,A
      ORL P2,#2h
      ANL P2,#11111000b

      ;***** KEYBOARD OKUMA*****
      MOV P0,R3
      ORL P2,#7h
      ANL P2,#11111000b
      MOV P0,#0FFh
      ORL P2,#6h
      MOV A,P0
      ANL P2,#11111000b
      CJNE A,#0FFh,KEY3B
      JMP KEY8

KEY3B: MOV R1,#0h
KEY3A: INC R1
      RRC A
      JC KEY3A
      MOV A,R2
      MOV B,#8h
      MUL AB
      ADD A,R1
      MOV YTUS,A
      JNB KRED,KEY8
      CLR KRED
      CJNE A,ETUS,KEY6

INC TUSC
MOV A,TUSC
CJNE A,#6h,KEY4
MOV GTUS,YTUS
SETB TUSF
JMP KEY7

KEY4: JC KEY7
      DEC TUSC
      JMP KEY7

KEY6: MOV TUSC,#0h
KEY7: MOV ETUS,YTUS
      JMP KEY8

KEY8: MOV A,R3
      RR A
      MOV R3,A
      INC R2
      CJNE R2,#8h,KEY10
      MOV R2,#0h
      SETB DISPF
      JNB KRED,KEY9
      MOV TUSC,#0h
      MOV YTUS,#0h

KEY9: SETB KRED

KEY10: POP AR1
      POP B
      POP ACC
      POP PSW
      POP DPL
      POP DPH

KEY11: RETI
; ***** END OF DISPLAY & KEYBOARD*****

; <<<*****
; <<< TUS DEGERLENDIRME
; <<<*****
TUS_DEG: CLR TUSF
          MOV A,GTUS

TUS1: CJNE A,#13D,TUS2 ; MEMORY TUSU
      SETB MEMORYF
      JMP TUS_SON

TUS2: CJNE A,#16D,TUS3
      SETB DOKUZTF
      JMP TUS_SON

TUS3: CJNE A,#24D,TUS4
      SETB SEKIZTF
      JMP TUS_SON

TUS4: CJNE A,#32D,TUS5
      SETB YEDITF
      JMP TUS_SON

TUS5: CJNE A,#15D,TUS5
      SETB ALTITF
      JMP TUS_SON

TUS5: CJNE A,#23d,TUS6
      SETB BESTF
      JMP TUS_SON

TUS6: CJNE A,#31D,TUS7

```

```

                SETB DORTTF
                JMP TUS_SON

TUS7:   CJNE A,#14D,TUS8
                SETB UCTF
                JMP TUS_SON

TUS9:   CJNE A,#22D,TUS10
                SETB IKITF
                JMP TUS_SON

TUS10:  CJNE A,#30D,TUS11
                SETB BIRTF
                JMP TUS_SON

TUS11:  CJNE A,#21D,TUS12
                SETB SIFIRTF
                JMP TUS_SON

TUS12:  CJNE A,#29D,TUS0
                LCALL F_SIFIRLA
                SETB CLEARF
                JMP TUS_SON

TUS_SON: SETB P3.4      ; BUZZER
                CALL DELAY ;
                CALL DELAY ;
                CALL DELAY ;
                CALL DELAY ;
                CALL DELAY ;
                CLR P3.4      ; END OF
BUZZER
                RET
;***** END OF KEYBOARD *****

; <<<*****
; <<< TUS DEGERLENDIR / TUS TO GRAM
; >>>
; <<<*****

TUS_2_GRAM: CALL TUS_2_BOS5

                JNB RAKAMF,TUS_G_SON
                CLR RAKAMF

TUS_G1:  MOV GRAM+2,GRAM+1
                MOV GRAM+1,GRAM+0
TUS_G2:  MOV GRAM+0,BOS5+0

TUS_G_SON: RET
; <<<***** END OF TUS TO GRAM *****

; <<<*****
; <<< TUS DEGERLENDIR / TUS TO BOS5
; >>>
; <<<*****

TUS_2_BOS5: MOV A,FLAG8
                ANL A,#3h
                ORL A,FLAG7
                JNZ TUS_B1
                CLR RAKAMF
                JMP TUS_B_SON

TUS_B1:  SETB RAKAMF

TUS_B3:  JNB SIFIRTF,TUS_B4
                MOV BOS5+0,#0d

TUS_B4:  JNB BIRTF,TUS_B5
                MOV BOS5+0,#1d

TUS_B5:  JNB IKITF,TUS_B6
                MOV BOS5+0,#2d

TUS_B6:  JNB UCTF,TUS_B7
                MOV BOS5+0,#3d

TUS_B7:  JNB DORTTF,TUS_B8
                MOV BOS5+0,#4d

TUS_B8:  JNB BESTF,TUS_B9
                MOV BOS5+0,#5d

TUS_B9:  JNB ALITTF,TUS_B10
                MOV BOS5+0,#6d

TUS_B10: JNB YEDITF,TUS_B11
                MOV BOS5+0,#7d

TUS_B11: JNB SEKIZTF,TUS_B12
                MOV BOS5+0,#8d

TUS_B12: JNB DOKUZTF,TUS_B_SON
                MOV BOS5+0,#9d

TUS_B_SON: MOV FLAG7,#0h
                ANL FLAG8,#11111100b

                RET
; <<<***** END OF TUS TO BOS5 *****

; <<<*****
; <<< EKTRAN
; <<<*****

EKTRAN:  MOV A,BOS4+5
                SWAP A
                ANL A,#00001111b
                MOV BOS+4,A

                MOV A,BOS4+5
                ANL A,#00001111b
                MOV BOS+3,A

                MOV A,BOS4+4
                SWAP A
                ANL A,#00001111b
                MOV BOS+2,A

                MOV A,BOS4+4
                ANL A,#00001111b
                MOV BOS+1,A

                MOV A,BOS4+3
                SWAP A
                ANL A,#00001111b
                MOV BOS+0,A

                MOV A,BOS4+3
                ANL A,#00001111b
                MOV BOS3+5,A

                MOV A,BOS4+2
                SWAP A

```

```

ANL A,#00001111b
MOV BOS3+4,A

MOV A,BOS4+2
ANL A,#00001111b
MOV BOS3+3,A

MOV A,BOS4+1
SWAP A
ANL A,#00001111b
MOV BOS3+2,A

MOV A,BOS4+1
ANL A,#00001111b
MOV BOS3+1,A

MOV A,BOS4+0
SWAP A
ANL A,#00001111b
MOV BOS3+0,A

;KARARTMALAR
MOV A,BOS+4
JNZ EKRAM_SON
MOV BOS+4,#0BH

MOV A,BOS+3
JNZ EKRAM_SON
MOV BOS+3,#0BH

MOV A,BOS+2
JNZ EKRAM_SON
MOV BOS+2,#0BH

MOV A,BOS+1
JNZ EKRAM_SON
MOV BOS+1,#0BH

MOV A,BOS+0
JNZ EKRAM_SON
MOV BOS+0,#0BH

MOV A,BOS3+5
JNZ EKRAM_SON
MOV BOS3+5,#0BH

MOV A,BOS3+4
JNZ EKRAM_SON
MOV BOS3+4,#0BH

EKRAM_SON: RET
;*** END OF EKRAM ***

;*** COMPRESS ***
;*** Bu program, BOS4+6,5,4,3,2,1,0'deki ayrik BCD
;sayilardan 0Bh olanlari sifirlar. ***
;*** KARARTMALARI SIFIRLA ***
KARA_SIFIRLA: MOV A,BOS4+6
CJNE A,#0BH,KARA_S1
MOV BOS4+6,#0H

MOV A,BOS4+5
CJNE A,#0BH,KARA_S1
MOV BOS4+5,#0H

MOV A,BOS4+4
CJNE A,#0BH,KARA_S1
MOV BOS4+4,#0H

MOV A,BOS4+3
CJNE A,#0BH,KARA_S1
MOV BOS4+3,#0H

MOV A,BOS4+2
CJNE A,#0BH,KARA_S1
MOV BOS4+2,#0H

MOV A,BOS4+1
CJNE A,#0BH,KARA_S1
MOV BOS4+1,#0H

KARA_S1: RET
;*** END OF KARARTMALARI SIFIRLA ***

;*** EKLE BOS4E ***
;*** BOS4+4,3,2'ye BOS3+2,1,0'i ekler. ***
EKE_BOS4E: MOV A,BOS4+2
ADD A,BOS3+0
DA A
MOV BOS4+2,A

MOV A,BOS4+3
ADDC A,BOS3+1
DA A
MOV BOS4+3,A

MOV A,BOS4+4
ADDC A,BOS3+2
DA A
MOV BOS4+4,A
RET
;*** END OF EKLE BOS4E ***

;*** COMPLEMENT BOS3+2,1,0 ***
COMP_BOS3: MOV A,#99h
CLR C
SUBB A,BOS3+0
MOV BOS3+0,A

```

```

MOV A,#99h
CLR C
SUBB A,BOS3+1
MOV BOS3+1,A

MOV A,#99h
CLR C
SUBB A,BOS3+2
MOV BOS3+2,A

; ADD 1
MOV A,BOS3+0
ADD A,#1h
DA A
MOV BOS3+0,A

MOV A,BOS3+1
ADDC A,#0h
DA A
MOV BOS3+1,A

MOV A,BOS3+2
ADDC A,#0h
DA A
MOV BOS3+2,A
RET
;***** END OF COMPLEMENT BOS3+0,1,2 ****

; <<<<*****
; <<<< 5 baytlik BCD sola kaydir
; <<<< BOS4+7,6,5,4,3,2'yi birer nibble sola kaydir.
; <<<<*****
BOS4_BCD_KAY6: ANL BOS4+7,#0Fh
MOV A,BOS4+6
ANL A,#0F0h
ORL A,BOS4+7
SWAP A
MOV BOS4+7,A

BOS4_BCD_KAY5: ANL BOS4+6,#0Fh
MOV A,BOS4+5
ANL A,#0F0h
ORL A,BOS4+6
SWAP A
MOV BOS4+6,A

BOS4_BCD_KAY4: ANL BOS4+5,#0Fh
MOV A,BOS4+4
ANL A,#0F0h
ORL A,BOS4+5
SWAP A
MOV BOS4+5,A

BOS4_BCD_KAY3: ANL BOS4+4,#0Fh
MOV A,BOS4+3
ANL A,#0F0h
ORL A,BOS4+4
SWAP A
MOV BOS4+4,A

ANL BOS4+3,#0Fh
MOV A,BOS4+2
ANL A,#0F0h
ORL A,BOS4+3
SWAP A
MOV BOS4+3,A

MOV A,BOS4+2
ANL A,#0Fh
SWAP A
MOV BOS4+2,A

RET
;***** END OF 5 BAYTLIK BOS4 BCD SOLA
KAYDIR

; <<<<*****
; <<<< EKLEME7
; <<<< BOS+6,5,4,3,2,1,0'a BOS4+8,7,6,5,4,3,2'yi ekler.
; <<<<*****
EKLEME7: MOV A,BOS+0
ADD A,BOS4+2
DA A
MOV BOS+0,A

```

```

MOV A,BOS+1
ADDC A,BOS4+3
DA A
MOV BOS+1,A

MOV A,BOS+2
ADDC A,BOS4+4
DA A
MOV BOS+2,A

MOV A,BOS+3
ADDC A,BOS4+5
DA A
MOV BOS+3,A

MOV A,BOS+4
ADDC A,BOS4+6
DA A
MOV BOS+4,A

MOV A,BOS+5
ADDC A,BOS4+7
DA A
MOV BOS+5,A

MOV A,BOS+6
ADDC A,BOS4+8
DA A
MOV BOS+6,A

EKLEME7 SON: RET
; <<<***** END OF EKLEME7 *****>>>
; <<<*****>>>
; <<< ALT CARPMA
; <<< BOS3+5,4,3,2,1,0 ile BOS2'yi carpar.
; <<< sonucu BOS4+6,5,4,3,2,1,0'e yazar.
; <<<*****>>>
BOS3_x_BOS2: MOV A,BOS3+0
MOV B,BOS2
MUL AB
MOV B,#10d
DIV AB
MOV BOS4+0,B
MOV BOS4+1,A

MOV A,BOS3+1
MOV B,BOS2
MUL AB
ADD A,BOS4+1
MOV B,#10d
DIV AB
MOV BOS4+1,B
MOV BOS4+2,A

MOV A,BOS3+2
MOV B,BOS2
MUL AB
ADD A,BOS4+2
MOV B,#10d
DIV AB
MOV BOS4+2,B
MOV BOS4+3,A

MOV A,BOS3+3
MOV B,BOS2
MUL AB
ADD A,BOS4+3

MOV B,#10d
DIV AB
MOV BOS4+3,B
MOV BOS4+4,A

MOV A,BOS3+4
MOV B,BOS2
MUL AB
ADD A,BOS4+4
MOV B,#10d
DIV AB
MOV BOS4+4,B
MOV BOS4+5,A

MOV A,BOS3+5
MOV B,BOS2
MUL AB
ADD A,BOS4+5
MOV B,#10d
DIV AB
MOV BOS4+5,B
MOV BOS4+6,A

RET
; ***** END OF BOS4 = BOS3 x BOS2 *****
; <<<*****>>>
; <<< BOS4 SIFIRLA
; <<< BOS4+7,6,5,4,3,2,1,0'i sifirlar.
; <<<*****>>>
BOS4_SIFIRLA: MOV BOS4+8,#0h
MOV BOS4+7,#0h
MOV BOS4+6,#0h
MOV BOS4+5,#0h
MOV BOS4+4,#0h
MOV BOS4+3,#0h
MOV BOS4+2,#0h
MOV BOS4+1,#0h
MOV BOS4+0,#0h
RET
; ***** END OF BOS4 SIFIRLA *****
; <<<*****>>>
; <<< CARPMA2
; <<<*****>>>
CARPMA2: CALL BOS4_SIFIRLA

MOV A,VADET+0
ANL A,#0Fh
MOV BOS2,A

CALL BOS3_x_BOS2

CALL COMPRESS

MOV BOS+6,#0h
MOV BOS+5,#0h
MOV BOS+4,#0h
MOV BOS+3,BOS4+5
MOV BOS+2,BOS4+4
MOV BOS+1,BOS4+3
MOV BOS+0,BOS4+2

; *****KINCI CARPAN
CALL BOS4_SIFIRLA

```

```

MOV A,VADET+0
SWAP A
ANL A,#0Fh
MOV BOS2,A

CALL BOS3_x_BOS2
CALL COMPRESS
CALL BOS4_BCD_KAY5
CALL EKLEME7

;****UCUNCU CARPAN
CALL BOS4_SIFIRLA

MOV A,VADET+1
ANL A,#0Fh
MOV BOS2,A

CALL BOS3_x_BOS2
CALL COMPRESS
CALL BOS4_BCD_KAY5
CALL EKLEME7

CARPMA2_SON: RET
;,<<<***** END OF CARPMA2 *****>>>

;,<<<*****
;,<<< EKLEME3
;,<<< BOS+4,3,2'ye BOS3+2,1,0'i ekler
;,<<<*****

EKLEME3: MOV A,BOS+2
ADD A,BOS3+0
DA A
MOV BOS+2,A

MOV A,BOS+3
ADDC A,BOS3+1
DA A
MOV BOS+3,A

MOV A,BOS+4
ADDC A,BOS3+2
DA A
MOV BOS+4,A
RET

;***** END OF EKLEME3 *****

;,<<<*****
;,<<< VGRAM KAYDIR
;,<<< BOS4+4,3,2'deki sayiyi sola kaydırma sayisini bulur.
;,<<< sonucu BOS2+0'a yazar.
;,<<<*****

VGRAM_KAY: MOV BOS2+0,#0h
VGRAM_K1: MOV A,BOS4+4
JNZ VGRAM_K_SON
CALL BOS4_BCD_KAY3

INC BOS2+0
JMP VGRAM_K1

VGRAM_K_SON: RET
;***** END OF VGRAM KAYDIR *****

;,<<<*****
;,<<< 7 baytlık BCD SAGA KAYDIR
;,<<< BOS4+6,5,4,3,2,1,0'i birer nibble saga kaydırır.
;,<<<*****

BOS4_SAGA_KAY: ANL BOS4+0,#0F0h
MOV A,BOS4+1
ANL A,#0Fh
ORL A,BOS4+0
SWAP A
MOV BOS4+0,A

ANL BOS4+1,#0F0h
MOV A,BOS4+2
ANL A,#0Fh
ORL A,BOS4+1
SWAP A
MOV BOS4+1,A

ANL BOS4+2,#0F0h
MOV A,BOS4+3
ANL A,#0Fh
ORL A,BOS4+2
SWAP A
MOV BOS4+2,A

ANL BOS4+3,#0F0h
MOV A,BOS4+4
ANL A,#0Fh
ORL A,BOS4+3
SWAP A
MOV BOS4+3,A

ANL BOS4+4,#0F0h
MOV A,BOS4+5
ANL A,#0Fh
ORL A,BOS4+4
SWAP A
MOV BOS4+4,A

ANL BOS4+5,#0F0h
MOV A,BOS4+6
ANL A,#0Fh
ORL A,BOS4+5
SWAP A
MOV BOS4+5,A

MOV A,BOS4+6
ANL A,#0F0h
SWAP A
MOV BOS4+6,A

RET
;,<<<***** END OF 7 BAYTLIK BOS4 BCD SAGA
KAYDIR *****>>>

;,<<<*****
;,<<< 5 baytlık BCD Bolunen kaydır
;,<<< BOS+4,3,2,1,0'i birer nibble sola kaydırır.
;,<<<*****

BOS_BCD_KAY5: ANL BOS+4,#0Fh
MOV A,BOS+3
ANL A,#0F0h
ORL A,BOS+4
SWAP A
MOV BOS+4,A

ANL BOS+3,#0Fh
MOV A,BOS+2
ANL A,#0F0h
ORL A,BOS+3
SWAP A
MOV BOS+3,A

```

```

        ANL  BOS+2,#0Fh
        MOV  A,BOS+1
        ANL  A,#0F0h
        ORL  A,BOS+2
        SWAP A
        MOV  BOS+2,A

        ANL  BOS+1,#0Fh
        MOV  A,BOS+0
        ANL  A,#0F0h
        ORL  A,BOS+1
        SWAP A
        MOV  BOS+1,A

        MOV  A,BOS+0
        ANL  A,#0Fh
        SWAP A
        MOV  BOS+0,A

    RET
;***** END OF 5 BAYTLIK BCD KAYDIR *****

; <<<*****
; <<<      BCD BOLME
; <<<      BOS+4,3,2,1,0i  BOS4+4,3,2YE  BOLER
; >>>
; <<<*****

BCD_BOLME:  MOV  R5,#10d

            CALL  VGRAM_KAY
            MOV  BOS3+0,BOS4+2
            MOV  BOS3+1,BOS4+3
            MOV  BOS3+2,BOS4+4

            MOV  BOS3+3,BOS4+2
            MOV  BOS3+4,BOS4+3
            MOV  BOS3+5,BOS4+4
            CALL  COMP_BOS3
            CALL  BOS4_SIFIRLA

BOL1:      MOV  COUNT,#0h

BOL2:      MOV  A,BOS+4
            CJNE A,BOS3+5,BOL3
            MOV  A,BOS+3
            CJNE A,BOS3+4,BOL3
            MOV  A,BOS+2
            CJNE A,BOS3+3,BOL3

BOL3:      JC  BOL4

            CALL  EKLEME3
            INC  COUNT
            JMP  BOL2

BOL4:      CALL  BOS_BCD_KAY5
            CALL  BOS4_BCD_KAY5
            MOV  A,COUNT
            ORL  BOS4+2,A

            DJNZ R5,BOL1

            CLR  C
            MOV  A,#5d
            SUBB A,BOS2+0
            MOV  R5,A

BOL5:      CALL  BOS4_SAGA_KAY
            DJNZ R5,BOL5

    RET

; <<<***** END OF BCD BOLME *****

; <<<*****
; <<<      SAYMA
; <<<*****

SAYMA:      MOV  BOS3+0,BOS+0
            MOV  BOS3+1,BOS+1
            MOV  BOS3+2,BOS+2
            MOV  BOS3+3,BOS+3
            MOV  BOS3+4,BOS+4
            MOV  BOS3+5,#0h

            ; (VADET*SNET) / VGRAM
            CALL  CARPMA2

            MOV  BOS4+2,VGRAM+0
            MOV  BOS4+3,VGRAM+1
            MOV  BOS4+4,VGRAM+2

            CALL  BCD_BOLME

            MOV  A,BOS4+2
            SWAP A
            ANL  A,#00001111b
            MOV  BOS+4,A

            MOV  BOS+3,BOS4+2
            ANL  BOS+3,#00001111b

            MOV  A,BOS4+1
            SWAP A
            ANL  A,#00001111b
            MOV  BOS+2,A

            MOV  BOS+1,BOS4+1
            ANL  BOS+1,#00001111b

            MOV  A,BOS4+0
            SWAP A
            ANL  A,#00001111b
            MOV  BOS+0,A

            MOV  A,BOS4+0
            ANL  A,#00001111b
            MOV  SONDIGIT,A

SAYMA_SON:  RET
;***** END OF SAYMA *****

; <<<*****
; <<<      TUS TAKIMIYLA KALIBRASYON
; >>>
; <<<*****

TUS_TAK_KAL:  CLR  TUSKALF
            SETB  SILMEF
            SETB  AGIRGOF
            setb  kalibredef

            ;***ORANITYI DEVREDEN CIKAR
            GERCEK GELENI YAZ
            MOV  VGRAM+2,#00h
            MOV  VGRAM+1,#10h
            MOV  VGRAM+0,#00h
            MOV  VADET+1,#0h
            MOV  VADET+0,#1h

```

```

TUS_TAK_1: JNB TUSF,TUS_TAK_2
            CALL TUS_DEG
            call tus_to_GRAM
TUS_TAK_2: JNB AGIRF,TUS_TAK_4
            CALL AGIR DEG
            JMP TUS_TAK_4

TUS_TAK_3: JNB CLEARF,TUS_TAK_4
            CLR CLEARF
TUS_TAK_3A: JMP TUS_TAK_SON

TUS_TAK_10: JNB MEMORYF,TUS_TAK_6
            CLR MEMORYF

TUS_TAK_12A: MOV VADET+1,GRAM+0
            MOV VADET+0,GRAM+1

TUS_TAK_15: MOV R0,#(VGRAMADR+0)
            MOV VGRAM+2,@R0
            INC R0
            MOV VGRAM+1,@R0
            INC R0
            MOV VGRAM+0,@R0

            MOV A,VGRAM+2
            JNZ D_VGRAM_2
            MOV A,VGRAM+1
            JNZ D_VGRAM_2
            MOV A,VGRAM+0
            JNZ D_VGRAM_2
            MOV VGRAM+0,#10h

D_VGRAM_2: MOV DEVSEL,#KAL_DEVSEL
            MOV NOFBYT,#KAL_NOFBYT
            MOV ESTAD,#KAL_ESTAD
            MOV ISTAD,#KAL_ISTAD
            CLR IE.7 ; DISABLE

ALL        CALL YAZRAM
            SETB IE.7 ; ENABLE

ALL

TUS_TAK_SON: JNB AGIRF,TUS_TAK_16
            CALL AGIR_DEG

TUS_TAK_16: clr kalibredef
            RET
;***** END OF TUS_TAK_KAL *****

;*****
;***** BIZLA
;*****

BIZLA: SETB P3.4 ; BUZZER
        CALL DELAY
        CALL DELAY
        CLR P3.4 ; END OF

BUZZER    CALL DELAY2

BIZLA_SON: RET

;***** END OF BIZLA *****

```

```

;*****
;***** YUNUS
;*****
YUNUS:: CALL G_KARART

        MOV GRAM+4,#'Y'
        CALL BIZLA
        MOV GRAM+3,#'U'
        CALL BIZLA
        MOV GRAM+2,#'N'
        CALL BIZLA
        MOV GRAM+1,#'U'
        CALL BIZLA
        MOV GRAM+0,#'S'
        CALL BIZLA

        CALL DELAY2
        CALL DELAY2
        CALL DELAY2
        CALL DELAY2
        CALL DELAY2
        CALL DELAY2
        SETB P3.4 ; BUZZER
        CALL DELAY2
        CLR P3.4 ; END OF

BUZZER

        CALL G_SIFIRLA

: RET
;***** END OF YUNUS*****

;*****
;***** INIT
;*****
INIT: CLR DATAALF

        MOV MEMO+0,#0H

        MOV ETUS,#0H
        MOV YTUS,#0H
        MOV GTUS,#0H
        MOV TUSC,#0H

        MOV NET11+0,#0H
        MOV NET11+1,#0H

        MOV FLAG4,#0H
        MOV FLAG7,#0H
        MOV FLAG8,#0H

        MOV R2,#0D
        MOV R3,#01111111b
        MOV R7,#5H

        MOV NEGA_COUNT,#0h
        MOV YUTPC,#0h
        MOV YUTNC,#0h
        MOV S_YUT_PC,#0h
        MOV S_YUT_NC,#0h
        CLR AGIRF
        CLR TUSF
        SETB KRED
        CLR DISPF
        CLR SILMEF

```

```

MOV FLAG6,#0h
SETB ONBESF
SETB YENIAGF

SETB AGIRGOF
CALL FLAG_SIL

CALL G_KARART
INIT_SON: RET
;***** END OF INIT *****

;*****
;*** FLAG SIL
;***
FLAG_SIL: CLR STARF
CLR MEMORYF
CLR CLEARF
CLR SATKALF
RET
;***** END OF FLAG SIL *****

;*****
;*** MICRO ISLEMCİ SETUP
;***
MP_SETUP: CLR RS0 ; PSW ban_0 is
selected
CLR RS1 ; " "

;***TIMERS
MOV TL0,#33H
MOV TH0,#0F5H
MOV 87h,#0H ;PCON
MOV TMOD,#21H
;Timer/Counter Mode Control Register

SETB TCON.0 ;Timer/Counter
Control Register
SETB TCON.4 ;
SETB TCON.6 ;
MOV TL1,#0
MOV TH1,#0FDH ; 9600BAUD

MOV 0B9h,#0FFh ;SADEN

MOV SCON,#0C0H ;Serial Port
Control
SETB SCON.5 ;SM2'yi set
ederek adres baslangici yap
SETB SCON.4 ;Enable Serial
Reception

MOV IE,#92H ; EA EC ET2 ES
ET1 EX1 ET0 EX0 ; 1 0 0 1
0 0 1 0 ; sonra duzelt

MOV IP,#3H

MP_SETUP_SON: RET
;***** END OF MICRO ISLEMCİ SETUP*****

;*****
;*** MAIN
;***
COLD_UP: CLR P3.4 ; CLEAR BUZZER
MOV SP,#STACK

CALL MP_SETUP
CALL INIT

call delay2
SETB DRDY
SETB DOUT
SETB SCLK
CALL C_KOSUL

MOV IE,#93H ; EA EC ET2 ES
ET1 EX1 ET0 EX0 ; 1 0 0 1
0 0 1 1

BEGINING: MOV DEVSEL,#KAL_DEVSEL
MOV NOFBYT,#KAL_NOFBYT
MOV ESTAD,#KAL_ESTAD
MOV ISTAD,#KAL_ISTAD
CLR IE.7 ; DISABLE
ALL
CALL OKURAM
SETB IE.7 ; ENABLE
ALL

MOV A,VGRAM+2
CJNE A,#0h,DOGRU_VGRAM
MOV A,VGRAM+1
CJNE A,#0h,DOGRU_VGRAM
MOV A,VGRAM+0
CJNE A,#0h,DOGRU_VGRAM
MOV VGRAM+1,#10h
DOGRU_VGRAM:

MOV IE,#93H ; EA EC ET2 ES
ET1 EX1 ET0 EX0 ; 1 0 0 1
0 0 1 1

CALL CHECK_LED

BEG7A: CALL YUNUS

MAIN2: JNB TUSKALF,MAIN3
MAIN2A: CALL TUS_TAK_KAL
MOV MEMO+0,#0H
JNB AGIRF,MAIN3
CALL AGIR_DEG

MAIN3:

LOOP: SETB AGIRGOF

LOP1: JNB AGIRF,LOP2
CALL AGIR_DEG

LOP2: JNB TUSF,LOP3
CALL TUS_DEG

LOP17: JMP LOOP
;*****
;*** END
;***
end

```

ÖZGEÇMİŞ

Doğum Tarihi : 23 Nisan 1973
Doğum Yeri : Bayburt
Eğitim : 1987-1990 İzmir Atatürk Lisesi, İzmir
1990-1994 Y.T.Ü Elektrik-Elektronik Fakültesi
Elektronik ve Haberleşme Müh. Böl.

Mesleki Geçmiş : 1994-1995 Erdem Elektronik A.Ş AR-GE
1995- Tarsan Elektronik A.Ş AR-GE

İlgi Alanları : Mikroişlemciler ve mikroişlemcili sistemler,
mikrodenetleyiciler, anahtarlama güç kaynakları, programlama dilleri

