



YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK – ELEKTRONİK FAKÜLTESİ
ELEKTRİK MÜHENDİSLİĞİ BÖLÜMÜ

AKILLI GÜNEŞ TAKİPLEYİCİ SİSTEM (SMART SOLAR TRACKER)

BİTİRME TEZİ

Hazırlayan

05012094 Uğur YILMAZ
05013004 Borahan KÜÇÜK

Tez Danışmanı

Yrd. Doç. Dr. İsmail AKSOY

İSTANBUL, 2009

İÇİNDEKİLER.....	Sayfa
ÖNSÖZ.....	4
ÖZET.....	5
ABSTRACT.....	6
1.GİRİŞ.....	7
2.TASARIM AŞAMASI.....	8
2.1. Mekanik Düzenek.....	8
2.1.1 Step motorlar.....	10
2.1.1.1 Step motorların kullanım alanları.....	11
2.1.1.2 Step motorların avantajları.....	11
2.1.1.3 Step motorların dezavantajları.....	12
2.1.1.4 Step motorlara ait bazı önemli kavramlar.....	12
2.1.1.4.1 Adım Tepkisi / Tek Adım Tepkisi / (Single Step Response).....	12
2.1.1.4.2. Adım Oranı (Stepping Rate / Starting Pulse Rate).....	13
2.1.1.4.3. Adım Açısı (Step Angle).....	13
2.1.1.4.4. Adım Açısı Doğruluğu (Step Angle Accuracy).....	13
2.1.1.4.5. Çözünürlük.....	14
2.1.1.4.6. Overshoot.....	14
2.1.1.4.7. Step Motor Tork Parametreleri.....	15
2.1.1.4.7.1 Tork - Hız Grafikleri.....	17
2.1.1.4.8. Etiket Voltajının Anlamı.....	17
2.1.1.5. Step motor çeşitleri.....	17
2.1.1.5.1. Sabit Mıknatıslı (Permanent Magnet, PM) Step Motorlar.....	18
2.1.1.5.2. Değişken Relüktanslı (Variable Reluctance, VR) Step Motorlar.....	19
2.1.1.5.3. Hibrid Step Motorlar.....	20
2.1.1.6. Step motor çalışma prensipleri.....	21
2.1.1.7. Bipolar ve unipolar motor ne demektir?	22
2.1.1.8. Step motorlarda uçların bulunması.....	23
2.1.1.9. Step motorların uyarımı.....	25
2.1.1.9.1. Tek-faz uyarım (1 Fazlı Tam Adımlı Sürüş)	25
2.1.1.9.2. İki-faz uyarım (2 Fazlı Tam Adımlı Sürüş)	26

2.1.1.9.3. Karma uyartım (2 Fazlı Yarım Adımlı Sürüş)	26
2.1.1.10. Step motorların denetimi.....	27
2.1.1.10.1. Açık Döngü Denetim.....	27
2.1.1.10.2. Kapalı Döngü Denetim.....	28
2.2 Kontrol sistemi.....	29
2.2.1 NPN transistörlü unipolar step motor sürücü devresi.....	29
2.2.2. ULN2003 entegresi ile unipolar step motor sürücü devresi.....	30
2.2.3. MOSFET transistörlü unipolar motor sürücü devresi.....	31
2.2.4. UCN5804B entegresi ile unipolar step motor sürücü devresi.....	31
2.2.5. Transistörlü bipolar step motor sürücü devresi.....	32
2.2.6. L297 + L298 entegreli bipolar step motor sürücü devresi.....	33
2.2.7 Sistemdeki ana kontrol kartı.....	33
2.3. Algoritma ve program.....	38
2.3.1 PIC16F877.....	38
2.3.1.1. PIC16F877'nin Özellikleri.....	39
2.3.1.2. RAM Bellek.....	40
2.3.1.3. PIC16F877'nin Uçları.....	41
2.3.1.3.1. PIC16F877'nin Besleme Ucu.....	41
2.3.1.3.2. Reset Devresi.....	42
2.3.1.3.3. Osilatör Devresi.....	42
2.3.1.3.4.Kristal Osilatör:	42
2.3.1.4. Minimum PIC devresi.....	43
3. SONUÇ VE ÖNERİLER.....	43
3.1 Sonuçlar.....	44
3.2 Öneriler.....	44
KAYNAKLAR.....	46
ÖZGEÇMİŞ.....	47

ÖNSÖZ

Projemizde, günün herhangi bir anında güneş ışınlarını en dik şekilde alacak biçimde panelin hareket etmesini sağlayacak bir sistem gerçekleştirdik. Bu sistem mikro kontrolör, güç elektroniği elemanları ve mekanik sistemlerin birleştirilmesiyle oluşturulmuştur.

Birçok kaynağı inceleyip, prototipini tasarladığımız bu düzeneğin oluşmasında bize büyük destek veren sayın Doç.Dr. Nurettin UMURKAN hocamıza, mekanik düzeneğin oluşmasında yardımlarını esirgemeyen İlhan KÜÇÜK'e, elektronik bilgisiyle bize büyük destek olan Levent BAŞARAN'a sonsuz teşekkürlerimizi sunarız

ÖZET

Bu proje çalışmasında, gün içinde herhangi bir anda güneşi sürekli olarak iki ekseninde takip ederek güneş ışınlarını en dik şekilde alacak bir sistem tasarlandı. Projede otomatik kontrol sistemleri, mekanik ve güç elektroniği gibi çalışma alanlarına ait bir çok teknikten yararlanılmıştır..

Gerçekleştirilen projede gün içinde güneşin sisteme göre olan konumu her an sensörlerden alınan bilgiler doğrultusunda yazılım aracılığıyla algılanmakta daha sonra PIC16F877 mikrokontrolör kullanılarak tespit edilen konuma göre mekanik sistem üzerinde bulunan iki adet step motoruna uygun sinyaller gönderilerek güneşin takip edilmesi işlemi gerçekleştirilmiştir.

Sensör olarak kullanılan LDR'ler üzerlerine gelen ışığa belirli miktarda geçirirler. Bu bilgiler mikrokontrolörün analog girişlerine verilir.PIC'in içine atılan yazılım dahilinde karşılaştırma yapılarak step motorların dönmesi için gerekli pulseler motorun giriş uçlarına verilir. Böylece güneş panelimizin güneşten maksimum şekilde yararlanması sağlanır. Ayrıca LDR'ler den alınan bu bilgiler LCD ekran üzerine yazılarak motorun hareketi gözlenmiş olur.

ABSTRACT

In this project, a system was designed that follows the sun everyminute in a day in two axis so can be observe the sun rays in normal angle. In this project, we utilized auto-control systems, mecanic and power electronic systems.

In project, we know the position of panel in a day by taking signals from LDR's. According to signal comes from LDR's PIC16F877 microcontroller can find the sun position. The movement of mecanical system is performed by two step motors. We can move the motors by usisng software which exported to PIC. On the other hand a LCD is used to display these position values everytime we need.

The system outlined in this report proved to be successful in maximising the energy received from the sun. Although it is not the only solution available, it is however, the most efficient given the time, budget and material restrictions involved with an undergraduate thesis.

1.GİRİŞ

Çevre kirliliğindeki artış günümüzde yenilenebilir enerji kaynaklarına olan ilgiyi giderek arttırmaktadır. Öte yandan enerji kullanımındaki artışın karşılanması, tükenmekte olan fosil yakıtlar ve kullanılsa da kullanılmasa da yalnız nükleer yakıtlarla mümkün görülmemekte ve anacak yenilenebilir enerjilerin kullanılması ve enerji kullanımını veriminin artırılması şartlarına bağlı olmaktadır.

Elektrik enerjisi üretiminde en çok kullanılan yöntem kömür gibi fosil yakıtların yakılması ile gerçekleştirilir. Fakat fosil yakıtların yakılması ile ortaya çıkan karbondioksit gazı sebebiyle meydana gelen sera etkisi sebebiyle küresel ısınma olayı ortaya çıkar. Fotovoltaikler çevre kirletici etki oluşturmayan enerji üretim seçeneklerinin başında gelmektedir.

Güneş enerjisinin alternatif bir enerji kaynağı olarak çekiciliğinin artmasındaki diğer faktörler güneş pillerinin yüksek güvenilirliği, kararlı performans artışı ve üretim masraflarının azaltılması ve diğer bir husus olarak güneş pillerinin yakıt giderine sahip olmamasıdır.

Dünyada artan enerji ihtiyacını karşılamak için güneş enerjisinin alternatif bir enerji kaynağı olarak kullanılmasından önce güneş pillerinin verimliliğinin artırılması gerekmektedir.

Günümüzde normal boyutlardaki bir güneş pilinin ortalama verimi sadece küçük güçlü ticari aygıtlara (hesap makinesi ve oyuncaklar gibi) yetecek miktardadır. Daha büyük güç isteyen cihazları besleyebilmek için daha büyük boyutlu güneş panelleri gerekmektedir. Fakat bunların büyük boyutlu olmaları uygulamalarda sorun yaratmaktadır. Panelin boyutunu arttırmaktansa her bir güneş pilinin verimini arttırmak daha mantıklıdır.

Güneş pilinin verimliliği iki ana etkene bağlıdır;

- 1) Hücre verimliliği
- 2) Hücre üzerindeki kaynak ışınım şiddeti

Güneş pillerinin üretilmesinde kullanılan materyaller hücre verimliliğini sınırlayan faktörlerin başında gelmektedir. Bu verimliliğin artırılmasını zorlaştırmakta ve bu

sebeple hücrenin tüm performansını sınırlamaktadır. Buna karşın pil üzerine düşen ışın miktarının artırılması daha kolay bir yöntemdir.

Güneş pilinin üzerine düşen ışınım şiddetinin artırılmasında 3 yöntem kullanılmaktadır. Bunlar;

- 1) Güneşten gelen ışınların sabit hücre üzerine odaklanması
- 2) Sabit algoritma kullanılarak güneşin konumunun izlenmesi
- 3) Güneş konumunun dinamik sistemler kullanılarak izlenmesi

Yapılan projede 3. yöntem kullanılarak güneşin iki ekseninde takip edilmesi işlemi gerçekleştirilmiştir.

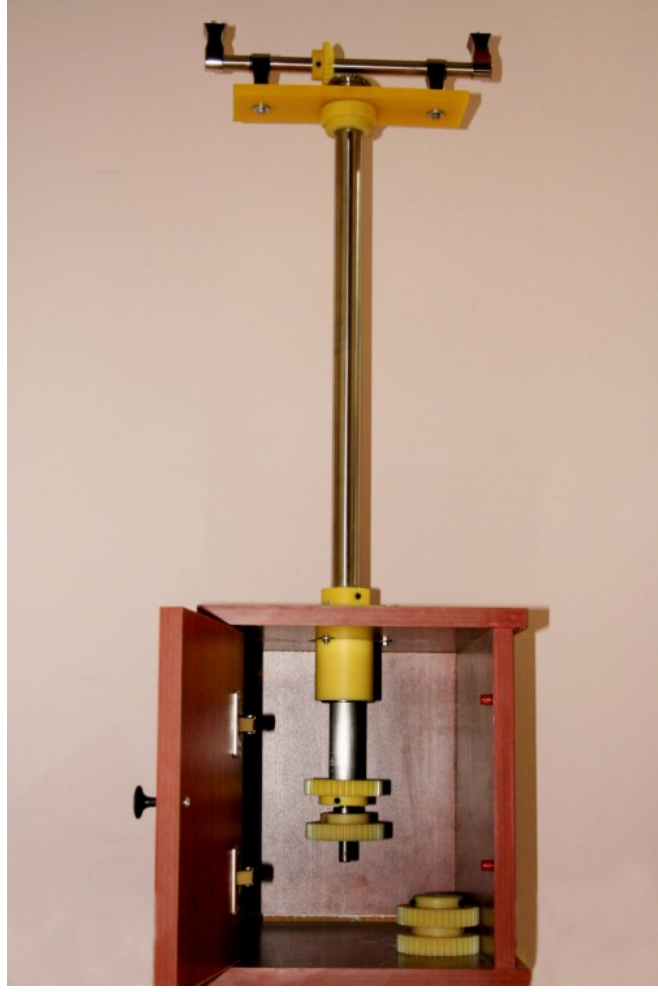
2. TASARIM AŞAMASI

Projenin tasarım aşaması 3 ana bölümden oluşmaktadır. Bunlar;

- **Mekanik düzeneğin oluşturulması:** Yatay ve dikey hareketi sağlayan 2 step motor, sensörlerin ve sürücü devrelerinin bulunduğu ana kontrol kartı, dişliler ve aktarma elemanları sistemin mekanik tasarımını kapsamaktadır.
- **Kontrol sistemi:** Bu bölümde mekanik sistemin iki temel hareketinin kontrolü amaçlanmıştır. Bu hareketler doğu-batı ve kuzey-güney hareketleridir. Sensör ve bu sensörlerden alınan konum bilgilerinin değerlendirilmesi incelenmiştir.
- **Algoritma ve program:** Mekanik sistemin tüm hareketlerinin kontrolü için mikro kontrolör kullanılmıştır. Sistem hareketlerini bir algoritma doğrultusunda yapmaktadır. Tasarım sırasında bu algoritma esnek tutulmuş ve güneşin en az hareketle bulunması amaçlanmıştır.

2.1. Mekanik düzenek

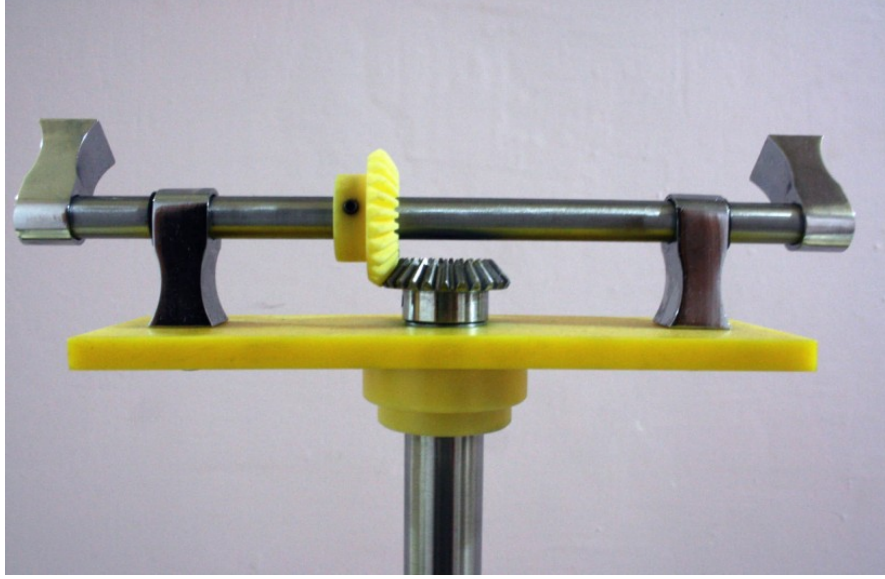
Mekanik sistem yatay-dikey ve sağa-sol hareketlerinin sağlayacak 2 adet step motor ve bunların hareketlerini panele aktaracak olan dişlilerden oluşmaktadır. Yapılan hesaplara göre yaklaşık 20W'lık bir güneş panelini taşıyabilecek büyüklükte ve güçte bir sistem tasarlanmıştır. LDR'lerden gelen voltaj bilgisi mikro kontrolörün analog girişlerine verilmiştir ve hazırlanan yazılım sayesinde bu voltaj bilgilerine göre step motor uçlarına darbe uygulanmıştır. PIC16F877'nin analog çıkışları step motorları döndürmek için gerekli olan akımı veremeyeceğinden dolayı 2 motor içinde birer sürücü devresi tasarlanmıştır ve bu devreler ana kontrol kartının içinde bulunmaktadır.



Şekil 2.1 Sistemimizin mekanik tasarımı



Şekil 2.2 Sistemde motorların dönüş hareketini panele aktaracak alan dişliler



Şekil 2.3 Kuzey-güney motorunun dönüşünün panele aktarılmasını sağlayan dişli grubu

2.1.1 Step motorlar



Şekil 2.4 Step motor

Step motorlar (adım motorları), girişlerine uygulanan darbe dizilerine karşılık (bu, dijital, sayısal giriş olarak da ifade edilebilir...), analog dönme hareketi yapabilen elektromagnetik elemanlardır. Bu özellikleri nedeniyle “dijital makine” olarak da tanınan adım motorları, dijital sistemlerde kullanılırken büyük kolaylıklar sağlarlar. Adım motorları, adından da anlaşılacağı gibi belirli adımlarla hareket ederek rotorun açısal konumunu değiştirirler. Bu adımlar, motor sargılarına uygun sinyaller gönderilerek kontrol edilir. Herhangi bir uyarımda, rotorun yapacağı hareketin ne kadar olacağı, motorun adım açısına bağlıdır. Adım açısı, motorun yapısına bağlı olarak 90, 45, 18, 7,5, 1,8... derece veya çok daha değişik açılarda olabilir. Motora uygulanacak sinyallerin frekansı değiştirilerek motorun hızı da kontrol edilebilir. Adım motorlarının dönüş yönü ise, uygulanan sinyallerin sırası değiştirilerek, saat ibresi yönünde (CW) veya saat ibresinin tersi yönünde (CCW) olabilir.

2.1.1.1 Step motorların kullanım alanları

Step motorlar uzun yıllardır var olmalarına rağmen ticari olarak kullanılmaları ancak 1960'lı yıllarda yüksek seviyeli doğru akımları anahtarlayabilen transistörlerin üretimine başlanmasıyla yaygınlaşmıştır. 1970'li yıllardan beri dijital elektronikteki ve mikroişlemci teknolojisindeki gelişmelerle birlikte adım motorlarının kullanımı giderek cazipleşmekte ve tüm dünyada bu motorların üretim ve uygulamalarıyla ilgili geliştirme çalışmaları yapılmaktadır.

Günümüzde adım motorları endüstride birçok kontrol sistemlerinde, hassas konum kontrolü yapmak amacıyla kullanılmaktadır. En çok yazıcılar (printer), çiziciler (plotter), disket sürücüler (floppy driver), harddisk sürücüler (harddisk driver), kart okuyucular... vb gibi bilgisayar çevre cihazlarında bu elemanlardan yararlanılmaktadır. Ayrıca sayısal kontrol sistemlerinde, CNC tezgâhlarda, proses kontrol sistemlerinde, robot teknolojisinde (milimetrik hareketlerin kontrolünde) ve uzay endüstrisine ait birçok sistemde adım motorları tahrik elemanı olarak yer almaktadır.

2.1.1.2 Step motorların avantajları

Step motorların bu kadar çok kullanım alanı bulmasının nedeni bu motorların bazı avantajlara sahip olmasıdır. Bu avantajlar aşağıdaki gibi sıralanabilir.

- Step motorlar dijital giriş işaretlerine cevap verirler, bu nedenle mikroişlemci veya bilgisayarlarla kontrol için ideal elemanlardır.
- Step motorların hangi yöne döneceği, devir sayısı, dönüş hızı gibi değerlerin mikroişlemci veya bilgisayar yardımı ile kontrol edilebilmesi, her an bu motorların dönüş yönü, hızı ve konumunun bilinebilmesini sağlamaktadır. Bu özelliklerinden dolayı step motorlar ile çok hassas konum kontrolü yapılabilir.
- Step motorların dijital girişlere cevap vermesi, geri beslemeye ihtiyaç duyulmaksızın açık çevrim çalıştırılabilmesini sağlamaktadır. Yani açık çevrim çalıştırılan bir adım motoru ile hız, ivme ve konum kontrolü daha basit ve daha az maliyetle gerçekleştirilebilir. Böylece alışılmış kararsızlık problemlerinin de önüne geçilmiş olur.
- Step motorlar, giriş işaretlerinin frekansına bağlı olarak çok geniş bir hız aralığında sürülebilirler.

- Step motorlar, herhangi bir hasara yol açmadan defalarca durdurulup çalıştırılabilirler.(Sürerken aniden durdurma ya da aniden ters yönde sürme isteğine karşı mükemmel cevap verebilirler.)
- Aşırı yüklenmeden hasar görmezler, oldukça dayanıklıdırlar.
- Her yeni adımla artan (kümülatif) konum hataları yoktur.
- Mekanik yapısı basit olduğundan bakım gerektirmezler.
- Yağlanma ve kirlenme problemleri yoktur.

2.1.1.3 Step motorların dezavantajları

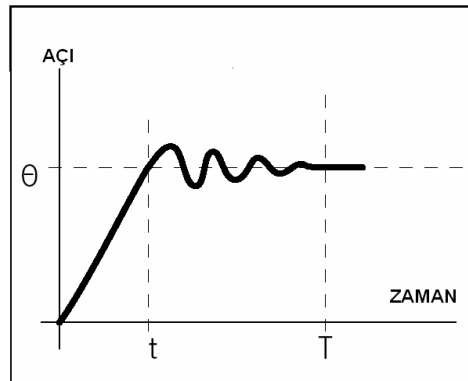
Step motorların bütün bu avantajlarına karşılık bazı dezavantajları da aşağıdaki gibi sıralanabilir.

- Adım açıları sabit olduğundan rotordan alınan hareket sürekli değil darbelidir.
- Klasik sürücülerle kullanıldıklarında verimleri düşüktür.
- Adım cevapları nispeten büyük aşım ve salınımlıdır.
- Yüksek eylemsizlikli yüklerde yetenekleri sınırlıdır.
- Sürtünme kaynaklı yükler, hata kümülatif olmasa dahi açık çevrim çalışmada konum hatası meydana getirebilirler.
- Elde edilebilecek çıkış gücü ve momenti sınırlıdır.
- İyi kontrol edilmezse rezonans meydana gelebilir.
- Oldukça yüksek hızlarda çalıştırmak pek kolay değildir.

2.1.1.4 Step motorlara ait bazı önemli kavramlar

Step motorlara ait bazı önemli kavramlar bu bölümde anlatılmıştır.

2.1.1.4.1 Adım Tepkisi / Tek Adım Tepkisi / Cevap Süresi / (Single Step Response)



Şekil 2.5 Step motorların adım tepkisi

Motor fazlarından biri uyarılmış durumdaysa, motor kararlı bir adım konumundadır. Bu fazın uyarımını kesilip yeni bir faz uyarılırsa rotor bir adım atacaktır.

Rotor konumunun zamana göre deęişimi “tek adım tepkisi” olarak tanımlanır. Bir başka deyişle motorun girişine bir komut palsı (sinyali) uygulandıktan sonra motorun adımlara cevap vermek için gerekli olarak aldığı zamana (T) “tek adım tepkisi”, “adım tepkisi” ya da “cevap süresi” denir. Bu süre hem motor parametrelerine hem de motorun sürücü devresine bağlıdır ve yaklaşık olarak milisaniye civarlarındadır.

Tek adım tepkisi, motorun adım hareketinin hızını, tepkinin aşım ve salınım miktarını, adım açısının hassaslığını veren önemli bir parametredir.

Adım motorlarından maksimum performans elde edebilmek için tek adım tepkisindeki aşım ve salınımların azaltılması ve yerleşme zamanının kısaltılması gerekmektedir. Bu nedenle tek adım tepkisinin iyileştirilmesi adım motorlarının kontrolünde çok büyük öneme sahiptir.

Motora uygulanan giriş darbeleri ile çıkış hareketi arasındaki senkronizasyonu bozmamak için, sürme esnasında iki darbe arasındaki süre, cevap süresinden daha kısa olmamalıdır. Dolayısıyla adım motorunun cevap süresinin kısaltılabilmesi, motorun daha hızlı adım atabilmesini sağlayacaktır.

Adım motorlarının cevap davranışlarında dikkat edilmesi gereken dięer bir nokta da yaptıkları aşımın(overshoot) ve osilasyonun miktarıdır. Adım motorları, bilgisayar sistemlerinde veya bilgisayarla kontrolü gerektiren hassas sistemlerde kullanıldığında, aşım ve osilasyon sistemi kötü yönde etkileyen ve önemli boyutta hatalara sebep olan bir davranıştır.

2.1.1.4.2. Adım Oranı (Stepping Rate / Starting Pulse Rate)

Bir saniyede rotorun yapabildięi adım sayısıdır. Bu adım sayıları, tipik olarak saniyede 300 ila 800 arasındadır.

2.1.1.4.3. Adım Açısı (Step Angle)

Motora bir tek pals uygulandıęında rotorun döndüğü açıdır. Yapımcı firmalar, deęişen sayılarda rotor ve stator sargıları ile çeşitli açılarda adım motorları yapmaktadırlar.

2.1.1.4.4. Adım Açısı Doğruluęu (Step Angle Accuracy)

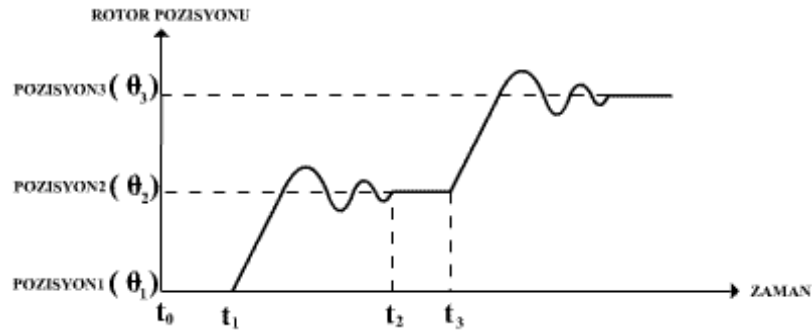
Rotorun yaptığı her bir adımdaki hata miktarını gösterir. Bu parametre genellikle bir yüzde ile verilir. Rotor bir adım döndüğünde meydana gelen toplam hata oranını gösteren bu parametrenin yazılması önemlidir. Bu hata deęeri

kümülatif(birikimli) değildir. Yani rotorun yaptığı her adım ile bu hata miktarı toplanarak gitmez. Rotorun gerçek hızı adım açısı ve adım hızına bağlıdır.

2.1.1.4.5. Çözünürlük

Çözünürlük, dönen motorlar için adım açısı(derece), lineer motorlar için ise adım uzunluğu(mm) olarak tanımlanır. Bu sabit değer, üretim sırasında tespit edilen bir büyüklüktür. Bir adım motorunun adım büyüklüğü, çeşitli uyartım metotları ile değiştirilebilir. Mesela yarım adım çalışmada, adım büyüklüğü normal değerinin yarısına indirilir.

2.1.1.4.6. Overshoot



Şekil 2.6 Step motorun rotor pozisyonu-zaman eğrisi

Step motora gönderilen palsların frekansı arttırılırsa, step motorun adımları hızlanır ancak adımlarının hızlanmasıyla step motor istenmeyen bir karakteristik gösterir. Buna “overshoot” denir. Bu durum yandaki şekilde görülmektedir.

Motorun rotoru “Pozisyon1” ve “t₀” anında durmaktadır. “t₁” anında uygun stator sargısına güç uygulanmıştır. Bu uygulama rotorun “Pozisyon2” ye dönmesini sağlar. Rotor “Pozisyon2” de tam olarak durmaz çünkü bir dönme momenti vardır. Bundan dolayı dönmek isteyecektir. Rotor biraz daha döner. Buna “overshoot” denir. İki kutup arasındaki magnetik ilişkiden dolayı rotor tekrar geri döner ama yine bir miktar “overshoot” meydana gelir. Rotorun hareketi kesin ve net olmayacağından tam durması istenen yerde duramayacaktır. Kutupların manyetik etkisinden dolayı geri gelip yerinde durması gerekirken fazla dönerek tekrar istenmeyen bir durum meydana gelir. Yani rotor basit harmonik hareket ya da sönümlü osilasyon yapar. Neticede motora pals uygulandıktan sonra rotorun durması bir miktar zaman alır. Rotorun durması için geçen zamana “oturma(yerleşme) zamanı (settling-time)” denir. Bazı devrelerde rotorun durmasının çabuk olması istenir. Bunun yapılabilmesi için bazı işlemlere ihtiyaç vardır. Bu işleme “damping” denir.

2.1.1.4.7. Step Motor Tork Parametreleri

Dönme momentine ‘Tork’ denir. Tork, elektrik devrelerindeki elektrik potansiyeline, milin açısal hızı da elektrik akımına benzetilebilir. Nasıl elektrik devrelerindeki direnç potansiyel etkisinde akımı sınırlarsa, mekanik devrelerde de sürtünme kuvvetleri milin tork etkisinde açısal hızına sınır getirir. Bir elektrik devresinde akü, potansiyel kaynağıdır aynı şekilde mekanik sistemlerde de motor, tork kaynağıdır. Motor mili, tork etkisinde dönmeye zorlanır.

Bir aküden çekilen akımı sıfırdan itibaren arttırsak, akünün iç direncinden dolayı akım arttıkça akü voltajı düşer. Benzer olarak bir motorun milinden alınan devir arttıkça motorun verdiği tork düşer.

Elektrik devrelerinde akım ve gerilimin çarpımı, düzgün harekette ise bir nesneye etkiyen kuvvet ile nesnenin hızının çarpımı gücü verirken, motorda tork ile açısal hızın çarpımı mekanik gücü verir.

$$P = I \times V \text{ (Güç = Akım x Gerilim)} \quad (2.1)$$

$$P = F \times V \text{ (Güç = Kuvvet x Hız)} \quad (2.2)$$

$$P = T_m \times \omega_n \text{ (Mekanik Güç = Tork x Açısal Hız)} \quad (2.3)$$

$$T_m = F \times r \text{ (Dönme Momenti [Tork] = Kuvvet x Yol)} \quad (2.4)$$

Bir motordan, herhangi bir devirde beklenen torkun büyüklüğü, motorun gücü ile orantılıdır. Sabit güç altında itme kuvveti artarken hız düşer, hız artarken ise itme kuvveti düşer ve dolayısıyla tork da düşer.

Step motor kataloglarında dört çeşit tork parametresine rastlanılır. Bu parametreler genellikle tork-hız grafikleriyle birlikte kataloglarda sunulur.

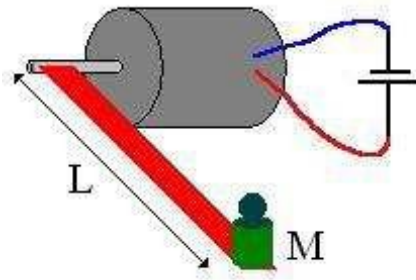
- a. Detent Torque (Yüksüz ve Enerjisiz Tutma Torku)
- b. Holding Torque (Yüksüz ve Enerjili Tutma Torku)
- c. Pull in Torque (Kalkıştaki Maksimum Yük Momenti)
- d. Pull Out Torque (Sürekli Rejimdeki Maksimum Yük Momenti)

a. **Detent Torque** (Yüksüz ve Enerj i siz Tutma Torku)

Detent Torque, motor enerjisiz iken rotor stator dişlerindeki etkileşimden dolayı oluşan tork demektir. Diğer bir ifadeyle sargılara besleme gerilimi uygulanmıyorken, step motor şaftını döndürmek için gerekli olan maksimum tork miktarı demektir. Bu parametre step motor tipine göre değişir.

b. **Holding Torque** (Yüksüz ve Enerjili Tutma Torku)

Holding Torque, “tutma torku” ya da “statik tork” olarak da bilinir. Sargılardan sadece birisinden nominal değerde akım akıtılıyorken milden alınan tork demektir. Diğer bir ifadeyle enerjili halde motor duruyorken üretilen maksimum tork miktarıdır. Tutma torku bir adım motorunun en temel moment karakteristiğidir ve pratik olarak şu şekilde ölçülebilir.



Şekil 2.7 Motor mil bağlantısı

Motor miline şekilde görüldüğü gibi ağırlıksız bir çubuk bağlanır ve çubuğun yatayla sıfır derece açı yapması sağlanır. Motor sargısından nominal değerde akım akıtılır. Şekildeki yeşil ağırlık çubuğun en ucuna oturtulur ve değişik ağırlıktaki kütlelerle denenerek hangi ağırlıktan sonra motor mili sapma yapıyor tespit edilir. Çubuğun boyu L ve Kütlenin ağırlığı M olması durumunda;

$$\text{Tutma Torku} = (\text{N.m}) = L \times M \times (m) \times \text{Kg} \times 9,81 \text{ (Yerçekimi İvmesi) (N/Kg)} \quad (2.5)$$

Örneğin Enerjili motorda, 10cm lik çubuk ucuna 250gr lık ağırlık konduğunda mil aşağı ya da yukarıya kaymıyorsa oluşan tork $T = 0.1 \times 0.25 \times 9.81 = 0.245 \text{ Nm}$ dir. Sonucun Amerikan sistemindeki karşılığı: $1\text{Nm}=141,6 \text{ ozf inc}$ ise $T=34,7 \text{ ozf inc}$ (onz forse inc) olur.

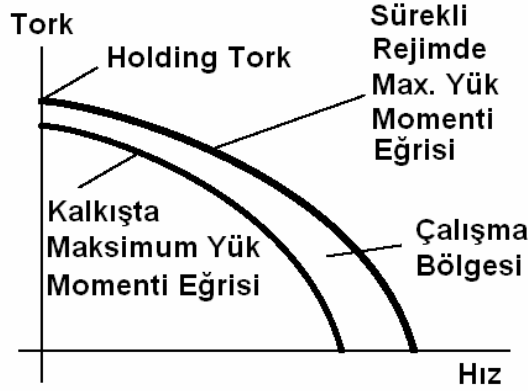
c. Pull in Torque (Kalkıştaki Maksimum Yük Momenti)

Özellikle açık döngülü sistemlerde duran bir sistemi istenen pozisyona getirebilmek için motora uygulanan uyarım darbelerinin motor tarafından hiç kaçırılmadan takip edilmesini sağlamak çok önemlidir. Fakat uygulanan uyarım sinyallerin sıklığı, motorun miline bağlı yükü sıfır hızından itibaren kaldırıp hızlandırmasına izin vermeyebilir. Bu yüzden adım motorları için, kalkışta maksimum yük momenti eğrileri tanımlanır.

d. Pull Out Torque (Sürekli Rejimdeki Maksimum Yük Momenti)

Bu parametre de motor hızına bağlı olarak değişeceğinden sürekli rejimde maksimum yük momenti eğrisinden bahsedilir.

2.1.1.4.7.1 Tork - Hız Grafikleri



Şekil 2.8 Tork – Hız Grafiği

Sürekli rejimde maksimum yük momenti/ hız eğrisi herhangi bir sabit dönüş hızında, rotor hareketinin giriş darbe dizisiyle olan senkronizasyonunu bozmadan ve rotorun durmasına neden olmadan sürekli halde motor miline uygulanabilecek maksimum yük momentini verir.

Bu moment aynı zamanda, söz konusu hızda motorda meydana gelecek maksimum moment anlamına da gelmektedir.

Klasik motorlarda bu eğriye karşılık gelebilecek bir karakteristik yoktur. Maksimum yük momenti eğrisi çalışma noktalarını göstermediği gibi bir transfer fonksiyonu eğrisi de değildir. Sadece, çalışma bölgesini sınırlar. Bu eğrinin sınırladığı bölge içinde herhangi bir noktada motor giriş darbe dizilerini kaybetmeden ve durma tehlikesi olmadan ilgili hız ve yük momenti ile çalışır. Sınırların dışına çıktığında bu durum değişebilir.

2.1.1.4.8. Etiket Voltajının Anlamı

Etikette belirtilen voltaj rotor hareketsizken tutma torkunun oluşturulması için gerekli voltaj değeridir. Bu voltaj değeri etiket akımına bölünürse ilgilenilen fazın sıcak sargı direnci bulunmuş olur. Etiket akımının üstüne çıkılmaması gerekir.

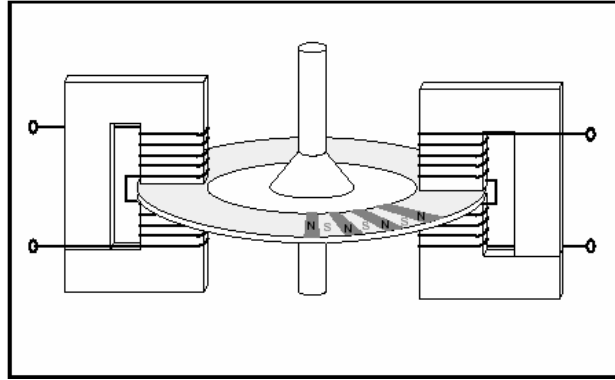
2.1.1.5. Step motor çeşitleri

Kullanımda olan birçok elektrik motorunda olduğu gibi step motorlar da makinenin yapısına ve çalışmasına göre sınıflandırılabilir.

Step motorlar için ilk günlerden bu yana birçok değişik tasarım yapılmıştır. Bu motorların kayda değer bir moment üretebilmeleri için hem rotor hem de statorları önemli miktarda magnetik akı taşıyabilecek yapıda olmalıdır. Bu nedenle de

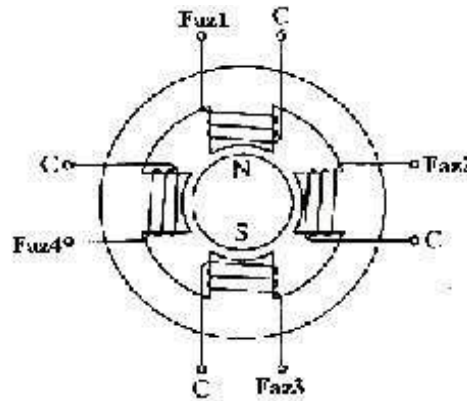
olabildiğince çok sayıda demir diş ile çevrili olmaları gerekmektedir. Günümüzde hem bu özelliği taşıyıp, hem de ticari bakımdan kullanılacak nitelikteki step motorların başlıcaları Değişken Relüktanslı, Sabit Mıknatıslı ve Hibrid step motorlardır. Temelde bunlar aynı türden cihazlardır, ama farklı yapım yöntemlerinden kaynaklanan farklı karakteristikleri belli bir uygulama için step motor seçilirken önem kazanır. Diğer tür step motorlar, bu üç tip step motorların varyasyonları olarak tanımlanabilirler. Step motorların en belirgin özelliği, girişine uygulanan uyarma darbelerinin, “adım” olarak tam tanımlanmış rotor konumlarına dönüştürmesidir. Rotorun hassas konumlama yapması, hareketli ve sabit parçalar üzerindeki demir dişlerin magnetik olarak karşılıklı gelmesi, dizilmesi ile mümkündür.

2.1.1.5.1. Sabit Mıknatıslı (Permanent Magnet, PM) Step Motorlar



Şekil 2.9

Sabit mıknatıslı step motorlardaki rotor, sabit mıknatıstan oluşur. Bu tip motorların rotorlarında diş bulunmaz. Yukarıdaki şekillerde de görüldüğü gibi iki farklı yapıda (“disk” ya da “tin-can” yapısı) rotor yapıları mevcuttur. Stator yapıları, değişken relüktanslı step motorların stator yapılarıyla aynıdır.



Şekil 2.10 4 fazlı bir step motor

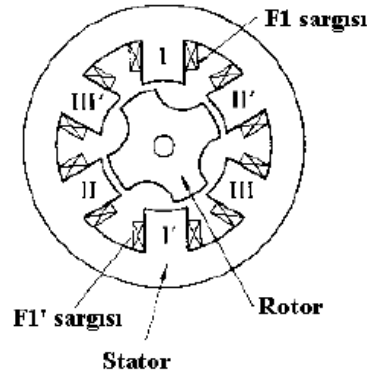
Basit olarak 4-fazlı sabit mıknatıslı bir step motor örneği yandaki şekilde gösterilmiştir. İki kutuplu sabit mıknatıslı rotor, oyuklu 4(dört) kutuplu stator içinde döner.

Burada C ile adlandırılan terminal, her bir fazın birer uçlarının birleştirilerek güç kaynağının pozitif ucuna bağlandığı ortak uçtur. Eğer fazlar Faz1, Faz2, Faz3, Faz4 sırasıyla uyarılırsa; rotor saat ibresi yönünde (CW) hareket edecektir. Bu motorda, adım açısının 90° olduğu açıkça görülmektedir.

Küçük boyutlu sabit mıknatıslı rotorda çok sayıda kutup oluşturmak zor olduğundan bu tip motorlarda adım genişliği $30-90$ derece arasındadır. Bu tür motorların momenti diğerlerine göre oldukça düşüktür.

2.1.1.5.2. Değişken Relüktanslı (Variable Reluctance, VR) Step Motorlar

Değişken relüktanslı step motorlar, sabit mıknatıslı step motorlar gibi bir kaç kutbu olan statora sahiptirler. Bu motorları sabit mıknatıslı step motorlardan ayıran en önemli fark rotorlarında sabit mıknatısın olmamasıdır. Ayrıca rotorları da dişli yapıdadır.



Şekil 2.11 3fazlı değişken relüktanslı step motor

Yukarıdaki şekilde üç-fazlı değişken relüktanslı step motorun kesit görünüşü verilmiştir.

Bu tür motorlarda rotor ve stator dişlerinin sayısı aynı değildir. Şekildeki motorda rotor 4, stator 6 diş sahiptir. Bir faz uyarıldığında asıl askıyı sadece iki rotor dişi taşır. Bu arada diğer rotor diş çifti ise uyarılmamış stator dişlerinin tam ortasındadır. Eğer uyarılan fazlarda değişiklik yapılırsa, yeni uyarılan stator fazıyla karşı karşıya gelecek olan rotor dişleri bu dişler olacaktır.

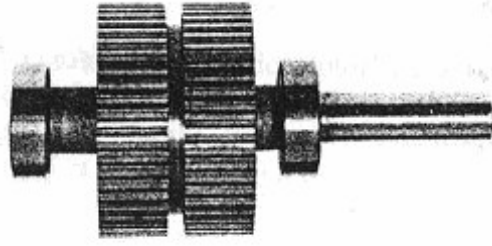
Değişken relüktanslı motorun stator ve rotor dişleri arasındaki hava aralığı, küçük çaplı rotordan mümkün olduğunca büyük tork üretebilmek ve konumlamada yüksek doğruluk elde edebilmek için mümkün mertebe küçük tutulmalıdır. Hava aralığı

diğerinden biraz daha büyük olan iki step motora aynı değerde manyetik kuvvet uygulandığında, küçük hava aralığı yüksek torku üreten manyetik akıyı verecektir ve ayrıca rotora dışarıdan bir kuvvet uygulandığında küçük aralığın denge pozisyonundan uzaklaşmasının da az olacağı açıktır. Modern motorlarda bu aralık 30 ila 100 μm 'dir.

Bu tip motorlarda rotor, manyetize olmamalı hafif ve küçük boyutlu yapılmalıdır. Rotor ölçülerinin küçük yapılması eylemsizlik momentinin de küçük olmasını sağlar. Böylece uygulanan gerilimin meydana getireceği moment sebebiyle rotor çok çabuk hareketlenir. Değişken relüktanslı step motorların başlama, durma ve adımları, sabit mıknatıslı step motorlardan daha hızlıdır.

2.1.1.5.3. Hibrid Step Motorlar

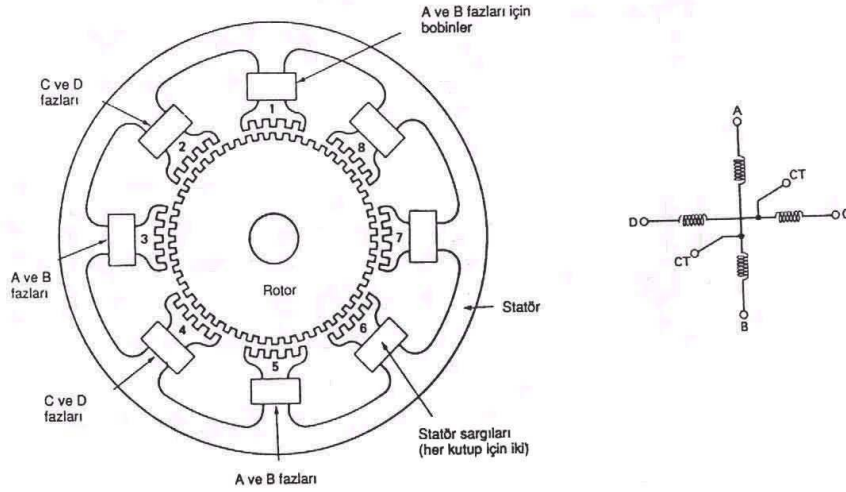
Hibrid step motorların rotorunda sabit mıknatıs bulunur. Bu tip motorlara Hibrid denmesinin sebebi, motorun değişken relüktanslı ve sabit mıknatıslı motorların çalışma prensiplerinin bir kombinasyonu ile çalışmasıdır.



Şekil 2.12

Rotorun her bir ucuna, tipik olarak üzerinde elli diş bulunan dişli türü göbekler vardır. Göbekler amaçlı olarak ayarsız yapılır, böylece mıknatısın kuzey ucundaki dişler, güney ucundakiyle 180° faz dışı kalır. Şekildeki fotoğraf, rotor yapısını net olarak göstermektedir.

Hibrid adım motorlarında tipik olarak 8 stator kutbu bulunur. Her kutupta bulunan diş sayısı iki ila altı arasındadır. Ayrıca rotorun istenen konuma gelmesini sağlamak üzere mıknatıs akısının ilgili kutuplar üzerinden akışını desteklemek veya engellemek amacıyla stator kutuplarına sargılar da ilave edilmiştir.



Şekil 2.13

Hibrid adım motorlarında iki farklı sargı kullanılır. Her bir sargı (faz), sekiz stator kutbundan dördünü dolaşır. A ve B sargıları 1, 3, 5, 7 kutupları üzerinde ise, C ve D sargıları 3, 4, 6, 8 kutuplarındadır. Her faza ait yakın kutuplar birbirleriyle zıt yönde sarılmışlardır.

Öncekilerde olduğu gibi, Hibrid adımlı bir motorun rotoru da, stator tahrik akımlarının uygun bir şekilde sıralanmasıyla adım adım hareket ettirilir.

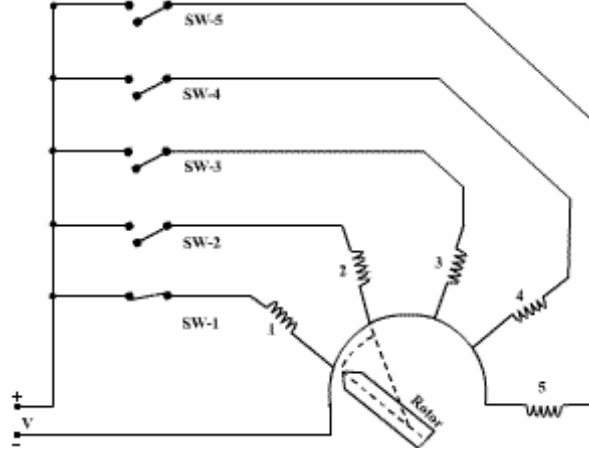
Bu tip motorların rotoru sabit mıknatıslı olduğundan, her zaman için bir tutma veya kalıntı torku vardır. Motor, düşük hızla ivmelendiği takdirde saniyede 30.000 adıma kadar hızlarda çalışabilir.

Tüm adım motorları içinde Hibrid tip, her türlü doğrusal ve açısal konumlama sisteminde en yaygın kullanılan tiptir.

2.1.1.6. Step motor çalışma prensipleri

Step motorlar bir dizi kısa elektrik akımıyla hareket ederler. Stator (hareketsiz kısım) birbirine dik manyetik alan üreten iki ayrı bobinden oluşur. Bu bobinlere sırayla elektrik akımı verilerek statorun içerisinde döndürme etkisine sahip bir manyetik alan oluşması sağlanır. Statorun içindeki rotor (hareketli kısım) bobinler tarafından sırayla oluşturulan manyetik alanla polarize olarak döner. Her bir elektrik akım vurgusu (pulse) rotorun belli bir açı kadar (bir adım) dönmesine neden olur. Eğer güç sadece bir bobine verilirse manyetik alanın etkisiyle rotor sabitlenecektir, bu da motorun durdurulmasında kullanılır.

Aşağıdaki şekilde bir step motorun temel prensip şeması verilmiştir.



Şekil 2.14 Temel prensip şeması

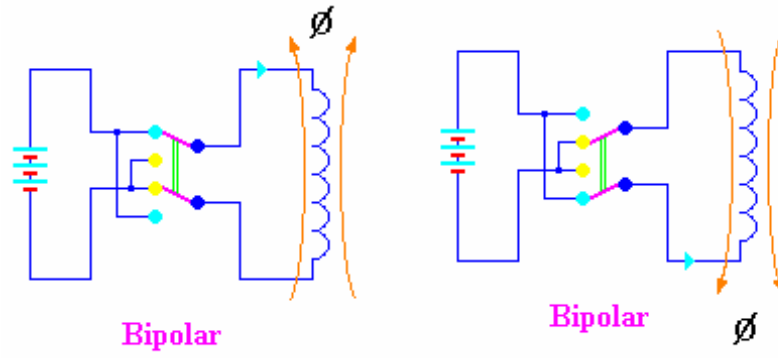
Step motor bir daire içinde elektromagnetik alanların dönüşü ile ifade edilebilir. 1 nolu anahtar kapandığında (SW-1) rotor kendiliğinden 1. elektromanyetik alan ile aynı hizaya gelecektir. Bundan sonra 1 nolu anahtar açılıp 2 nolu anahtar kapatılırsa sabit mıknatıs 2. elektromagnetik alanın karşısına gelecektir. Bu olaylar sırayla tekrarlanırsa rotor bir daire içinde düzgün şekilde döner.

Bu şekilde verilen elektrik akım vurgularının frekansı motorun dönme hızını belirler. Hızlı ivmelenme sonucunda step motorda kayma meydana gelebilir bunun önlenmesi için ivmelenme sırasında vurgu sıklığı ayarlanmalıdır.

2.1.1.7. Bipolar ve unipolar motor ne demektir?

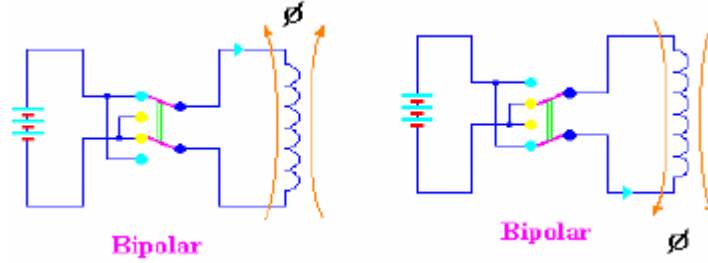
Bir bobinin oluşturmuş olduğu magnetik akının yönü, bobinin sarım yönüne ve bobinin içinden geçen akımın yönüne bağlıdır. Motor gibi elektromagnetik prensiple çalışan makinelerde, şiddeti ve yönü değişen magnetik alanlara ihtiyaç duyulur.

Manyetik akının yönü bobinden geçen akımın yönü ile değiştirilebilir. Ancak bazı uygulamalarda akım yönünü değiştirmek yerine bobinin sargı yönü değiştirilmek istenir. Tabi ki bu işlem sargıyı söküp ters yönde sarmayı gerektirir. Ancak biri birine göre ters sarılmış iki sargı kullanılırsa, magnetik akının yönü içinden akım geçen sargının sarım yönüne bağlı olacaktır.



Şekil 2.15 Bipolar step motor

Bipolar iki yönlü beslenen anlamına gelir ve Bipolar step motor, iki yönde de akım akabilen motor demektir. Bipolar motorların sadece 4 ucu vardır. Unipolar tek yönlü besleme olup sargılardan sadece tek yönde akım akıtılır demektir.



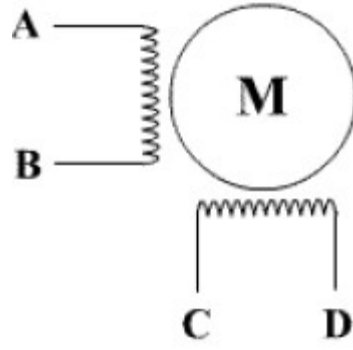
Şekil 2.16 Unipolar step motor

Verilen şekillerde manyetik akının nasıl değiştiği görülmektedir. Şekillerden de anlaşılacağı üzere, unipolar motorda daha çok uç (6 adet) vardır. İki motor tipinin birbirine göre avantajlarına bakacak olursak, Bipolar motorlar akım yönünün değişmesi ihtiyacından dolayı daha karmaşık sürücü devreleri ister. Unipolar motorlarda sürücü daha basittir.

Aynı boyutlardaki bipolar ve unipolar motordan, unipolar olanında aynı sargı alanına 2 bobin yerleştirildiğinden tel çapı küçük seçilir hâlbuki bipolar olana daha kalın telden sargı konabilir. Kalın sargı daha çok akım dolayısı ile daha çok tork demektir. En yaygın olarak 2 fazlı step motorlar kullanılmaktadır. Bunun anlamı 2 fazlı, Bipolar motorlarda 2 sargı, Unipolar motorlarda da 4 sargı olacak demektir.

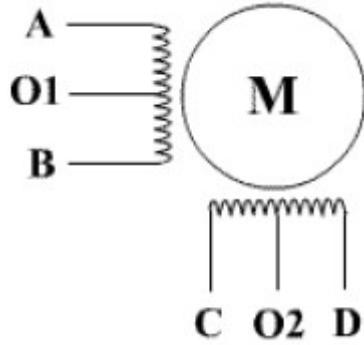
2.1.1.8. Step motorlarda uçların bulunması

Step motorlarda uçları ya kataloğuna bakarak ya da basit metotlarla bulabiliriz.



Şekil 2.17 Bipolar sargı tipi

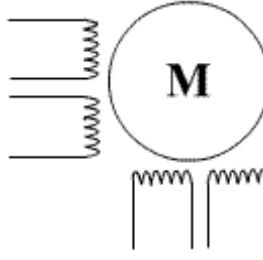
Örneğin elimizde 4 ucu olan bir step motor olduğunu farz edelim. Bir ölçü aleti ile uçlar arasında direnç ölçümü yaparsak ohm metreden sadece A - B arasında ya da C - D arasında bir direnç değeri okuyabiliriz. A - C, A - D ya da B - C, B - D uçlarının birbirleri ile bağlantıları olmadığı için ohm metrede hiç bir değer okuyamayız (ohm metre sonsuz direnç gösterir). Bu yol vasıtasıyla hangi uçların birbiri ile bağlantılı olduğunu yani motor bobinlerinin karşılıklı uçlarını belirleyebiliriz.



Şekil 2.18 Unipolar sargı tipi

Eğer elimizdeki motorun 6 ucu varsa, bu kez yine bir ölçü aleti yardımıyla uçlar arasında direnç ölçümü yaparak hangi üç ucun birbiri ile bağlantılı olduğunu buluruz. Şekilde O1 ve O2 ortak uçlardır. Dolayısıyla A - B arasında ölçülen direnç A - O1 ya da B - O1 arasında ölçülen direncin 2 katıdır. Aynı şekilde C - D arasında ölçülen direnç C - O2 ya da D - O2 arasında ölçülen direncin 2 katıdır.

NOT: Bazı motorlarda Unipolar sargıda iki bobinin ortak ucu dışarıya tek kablo ile çıkarılabilir. Bu durumda step motorun 5 ucu mevcuttur.



Şekil 2.19 8 uçlu step motor

Bazı unipolar step motorların sargılarından ayrı ayrı uçlar çıkarılabilir. Bu durumda da step motorun 8 ucu mevcuttur. Yukarıdaki şekilde 8 uçlu bir step motorun bağlantı uçları görülmektedir. Yapılması gereken yine bir ölçü aleti yardımıyla hangi uçların birbiri ile bağlantılı olduğunu belirlemektir.

2.1.1.9. Step motorların uyarıtımı

Step motorların uyarıtım şekilleri bu şekildedir.

2.1.1.9.1. Tek-faz uyarıtım (1 Fazlı Tam Adımlı Sürüş)

Motor sargılarının sadece birinin uyarıtıldığı uyarıtım cinsine tek-faz uyarıtımı adı verilir. Aşağıdaki tabloda 4-fazlı adım motoru için tek-faz uyarıtım sırasındaki fazların durumu görülmektedir. Bu uyarıtım metodunda rotor her bir uyarıtım sinyali için tam adımlık bir hareket yapmaktadır. Uyarıtım dönüş yönüne bağlı olarak sıra ile yapılır.

Step	Bobin 4	Bobin 3	Bobin 2	Bobin 1	
a.1	on	off	off	off	
a.2	off	on	off	off	
a.3	off	off	on	off	
a.4	off	off	off	on	

Şekil 2.20 Tek fazlı uyarıtım

2.1.1.9.2. İki-faz uyartım (2 Fazlı Tam Adımlı Sürüş)

Motor sargılarının ikisinin sıra ile aynı anda uyarıldığı uyartım cinsine iki-faz uyartımı adı verilir. Aşağıdaki tabloda 4-fazlı adım motoru için iki-faz uyartım sırasındaki fazların durumu görülmektedir. İki faz uyartımında rotorun geçici durum tepkisi tek-faz uyartımlıya göre daha hızlıdır. Fakat burada güç kaynağından çekilen güç iki katına çıkmaktadır.

Step	Bobin 4	Bobin 3	Bobin 2	Bobin 1	
b.1	on	on	off	off	
b.2	off	on	on	off	
b.3	off	off	on	on	
b.4	on	off	off	on	

Şekil 2.21 İki fazlı uyartım

2.1.1.9.3. Karma uyartım (2 Fazlı Yarım Adımlı Sürüş)

Bu uyartım yönteminde tek-faz uyartımı ile iki-faz uyartımı ard arda uygulanır. Burada rotor her bir uyartım sinyali için yarım adımlık bir hareket yapmaktadır. Aşağıdaki tabloda fazların uyartım sırası görülmektedir. Bu uyartım metodunda adım açısı yarıya düştüğünden adım sayısı iki katına çıkmaktadır.

Step	Bobin 4	Bobin 3	Bobin 2	Bobin 1	
a.1	on	off	off	off	
b.1	on	on	off	off	
a.2	off	on	off	off	
b.2	off	on	on	off	
a.3	off	off	on	off	
b.3	off	off	on	on	
a.4	off	off	off	on	
b.4	on	off	off	on	

Şekil 2.22 Karma uyarım

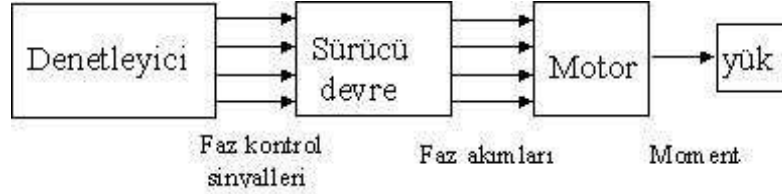
2.1.1.10. Step motorların denetimi

Step motorların 2 tip denetim vardır. Bunlar su şekildedir;

2.1.1.10.1. Açık Döngü Denetim

Şekilde açık döngü denetim için blok diyagramı görülmektedir. Sayısal kontrol sinyalleri denetleyici tarafından üretilir ve sürücü devre tarafından yükseltilip adım motorunun sargılarına uygulanır. Eğer denetleyici olarak mikroişlemci veya bilgisayar kullanılırsa bu elemanların getirdiği esnekliklerden dolayı aynı denetleyici ile farklı adım motorları kontrol edilebilir. Kontrol edilecek adım motorları 3, 4 veya daha farklı

faz sayısına sahip olabilir. Ayrıca kullanılacak uyarım metodu için tek-fazlı, iki-fazlı veya yarım adım uyarımlarından herhangi biri seçilebilir. Bu uyarım metodlarından hangisinin kullanılacağı daha önce de açıklandığı gibi motorun kullanılacağı sisteme bağlıdır.

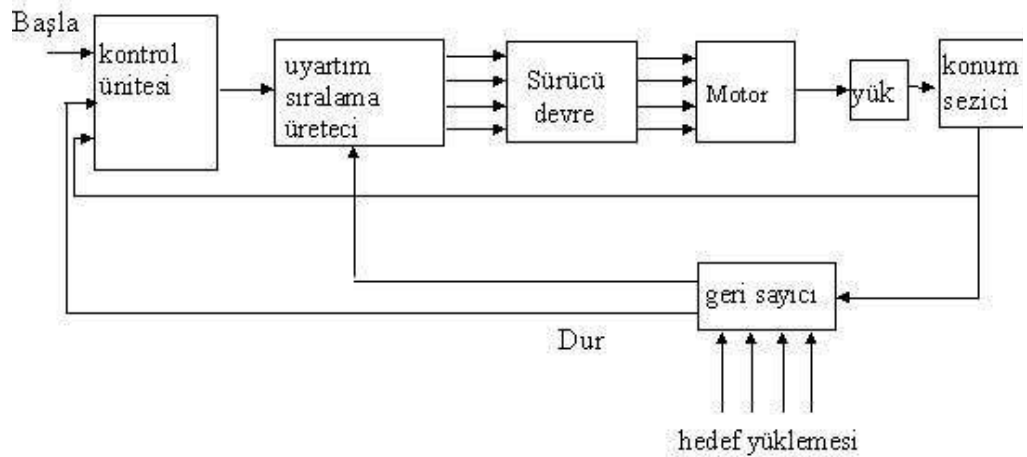


Şekil 2.23

Denetleyici tasarlanırken motorun cinsi ve yükün durumu göz önünde bulundurulmalıdır. Bu sırada meydana gelen sınırlamalar kalıcı veya geçici durum sınırlamaları olabilir. Açık döngülü denetimde motorun konumu bilinmediğinden dolayı motorun gönderilen bütün adım komutlarını yerine getirdiği varsayılmaktadır. Eğer uyarım hızı çok yüksek ise, motor adım komutlarından bir kısmını yerine getiremeyebilir. Bu durumda kalıcı bir hata meydana gelir. Bu tür hataların meydana gelmemesi için motor yükünün en büyük olduğu durum göz önüne alınarak hata yapılmayan en yüksek hız belirlenip, bu hızın üzerindeki hızlarda uyarım yapılmamalıdır.

2.1.1.10.2. Kapalı Döngü Denetim

Kapalı döngü sistemlerde ani rotor konumu sezilerek denetim birimine iletilir. Her adım komutu için bir önceki komutun gerçekleştirildiği adım bilgisi alınarak uygulanır. Bu nedenle motor ile denetleyici arasında herhangi bir adım kaybı olmaz. Kapalı döngü denetime bir örnek aşağıdaki şekilde gösterilmiştir.



Şekil 2.24

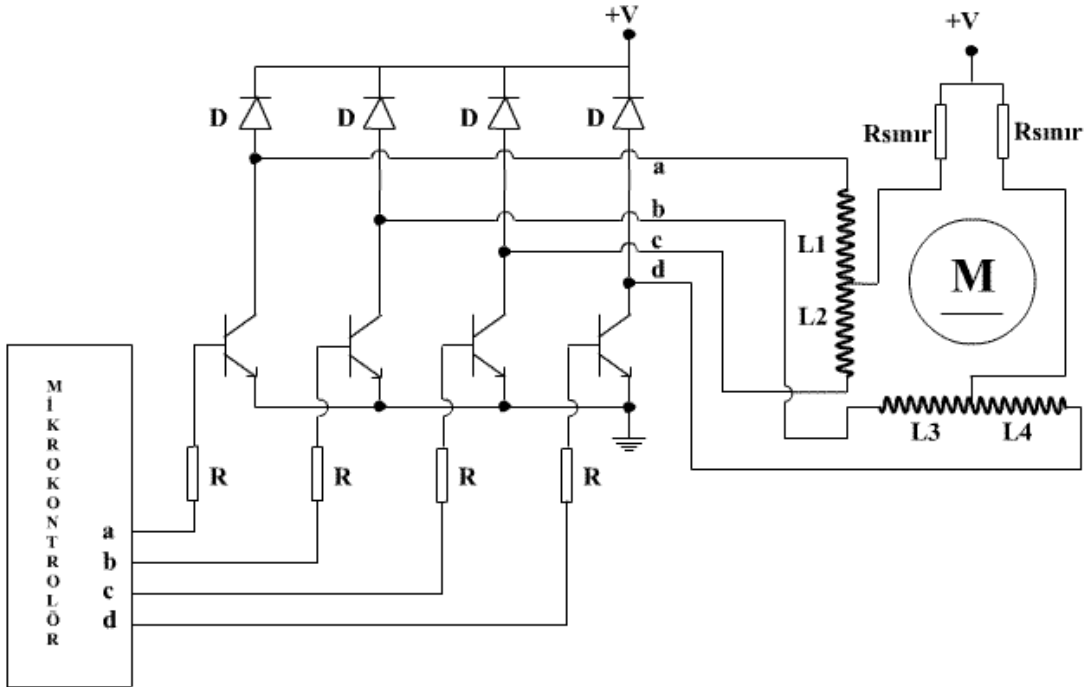
İlk olarak geri sayıcıya hedef konum yüklenir. Daha sonra başla komutu verilerek adım komutlarının sıralayıcıya uygulanması sağlanır. Adım komutlarına bağlı olarak motor adım hareketi yapmaya başlar. İlk adım tamamlanınca, konum sezici geri sayıcıyı ve denetim birimlerini uyarır ve geri sayıcı değeri bir azalır. Eğer bu denetim açık döngülü yapılırsa, geri sayıcı adım komutlarının sayısını yine saklar fakat komutun uygulanıp uygulanmadığı bilinmez. Konum sezici, denetim birimine yeni adım komutu üretimi için sinyal gönderir. Ağır yükler için adım komutları arası sürenin daha büyük olması nedeniyle adım komutlarının ard arda gelmesi istenmez.

Yüke göre hız ayarlaması yapılır ve motor hedef konuma gelene kadar bu olaylar tekrarlanır. Adım motoru hedef konuma gelince denetim birimi dur komutu ile uyarılarak yeni adım komutu üretilmesi engellenir. Kapalı döngü sistemi, adım motorunu yük durumunu da göz önüne alarak uyartım sürelerini ayarlar ve en uygun hız profilinde çalıştırır.

2.2 Kontrol sistemi

Step motorların devrede kullanılabilmesi için sürücü devreleri mevcuttur. Bu devreler motorun çeşidine göre değişiklik göstermektedir.

2.2.1 NPN transistörlü unipolar step motor sürücü devresi



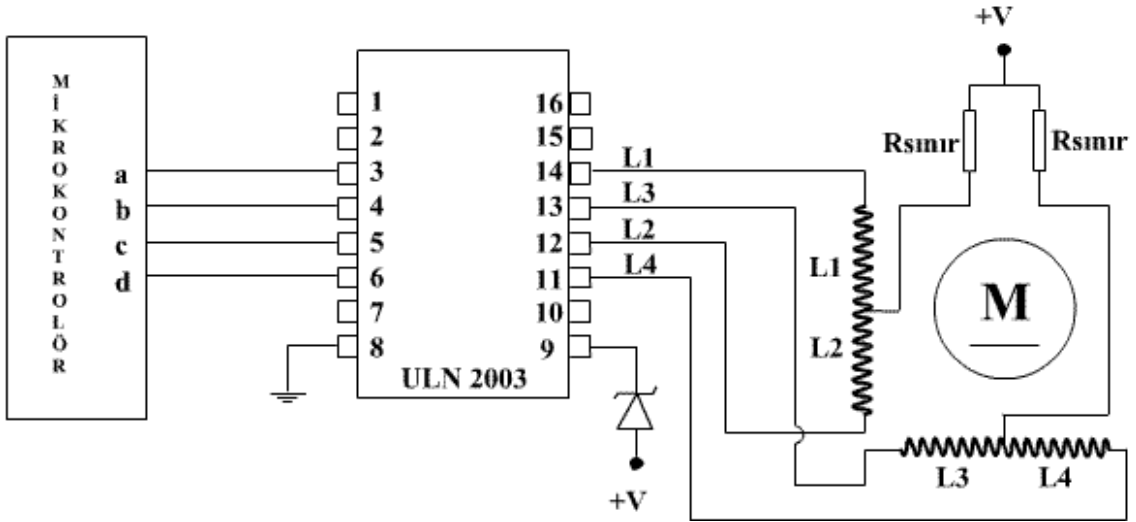
Şekil 2.25 NPN transistörlü unipolar sürücü devresi

Devrenin amacı, motorun ortak kablolarına, motor çalışma gerilimini uyguladıktan sonra, motorun diğer uçlarını, belli bir sırada toprağa göndererek, bir adım hareketi elde etmektir. Transistörün Base ucuna, mikro kontrolörden +5V geldiğinde, transistör tetiklenir ve Collector-Emetor ilettime geçer. Dolayısıyla transistörün Collector ucuna bağlı kablolar toprağa ulaşır.

Devredeki diyotlar, motorun üretmiş olduğu ters EMK'dan, transistörleri korumak için kullanılmışlardır. Yani, motor adım attıktan sonra oluşacak ters EMK, bu diyotlar sayesinde absorbe edilir. Kullanılan diyotun değerinin herhangi bir önemi yoktur.

Devredeki NPN transistörler switch (anahtarlama elemanı) olarak kullanılmışlardır ve bu transistörlerin modelleri, motor için gerekli olan akıma bağlıdır. Yaklaşık 5 Amper civarı akımlar için TIP141 modeli, daha küçük akımlar için ise BC108 veya BC547B tipi transistörler kullanılabilir. Transistörler BC547 seçilirse $R=4.7K$ ve $D=1N4001$ seçilebilir. Motor akımını sınırlamak için, $R_{sınır}$ direnci motor sargılarına seri olarak bağlanabilir.

2.2.2. ULN2003 entegresi ile unipolar step motor sürücü devresi



Şekil 2.26 ULN2003 entegresi ile unipolar step motor sürücü devresi

Step motor sürücü devresi olarak kullanılan ULN2003 entegresi içerisinde 7 adet NPN transistör ve dâhili diyot bulunmaktadır.

Bir önceki devrede transistör bacaklarıyla uğraşmak yerine, bu entegreyi kullanmak oldukça kolaylık sağlayacaktır.

Şekildeki devreyi kurduktan sonra, mikro kontrolörden herhangi bir bacağı +5V uyguladığımızda, karşısındaki bacak toprağa ulaşacaktır.

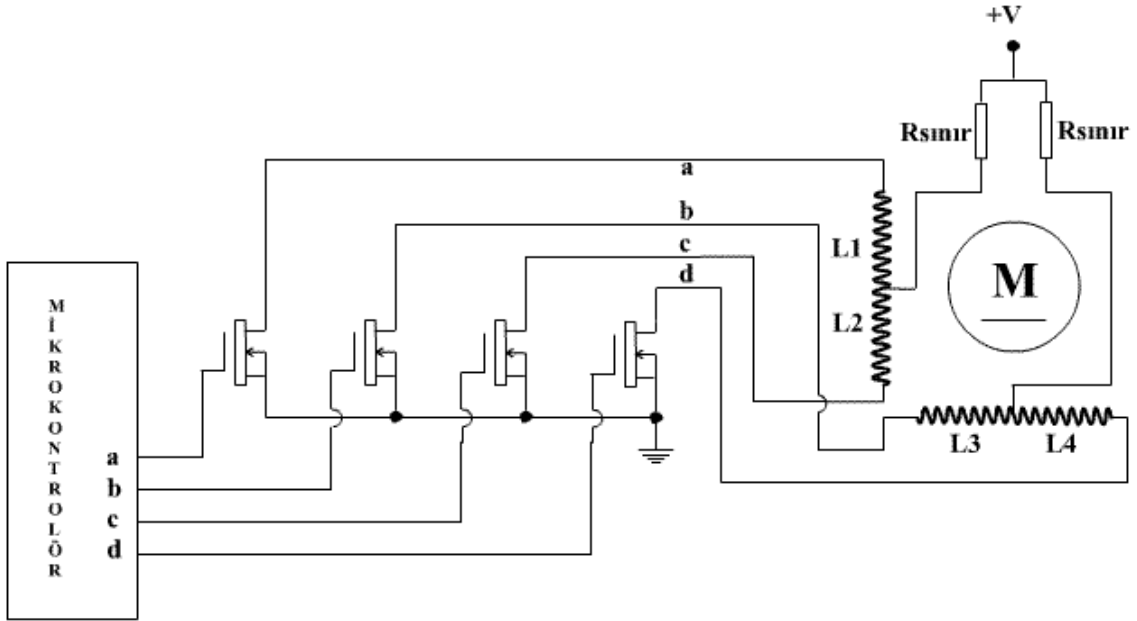
ULN2003 entegresi ile 500mA kadar akım çeken motorlar çalıştırılabilir. +12V'a kadar entegreye besleme gerilimi verilebilir.

2.2.3. MOSFET transistörlü unipolar motor sürücü devresi

Step motor sürücü elemanı olarak, MOSFET güç transistörleri kullanmak daha iyi bir seçimdir çünkü MOSFET'lerde, BJT transistörlerde olduğu gibi ikinci bir kırılma noktası yoktur. Ayrıca anahtarlama hızları da yüksektir.

Şekilde de görüldüğü gibi MOSFET'lerin giriş dirençleri yüksek olduğu için, kontrol devresine doğrudan bağlanabilirler.

Devredeki güç transistörleri (MOSFET) yerine IRLI520N ya da UN40AF ya da BD522 model transistörler kullanılabilir. Bu transistörlerin Drain ve Source bacakları arasında diyot olması nedeniyle, MOSFET'leri korumak için dışarıdan diyot bağlanmasına gerek yoktur. Bu durumda yeni bağlantı şeması aşağıdaki gibi olacaktır.

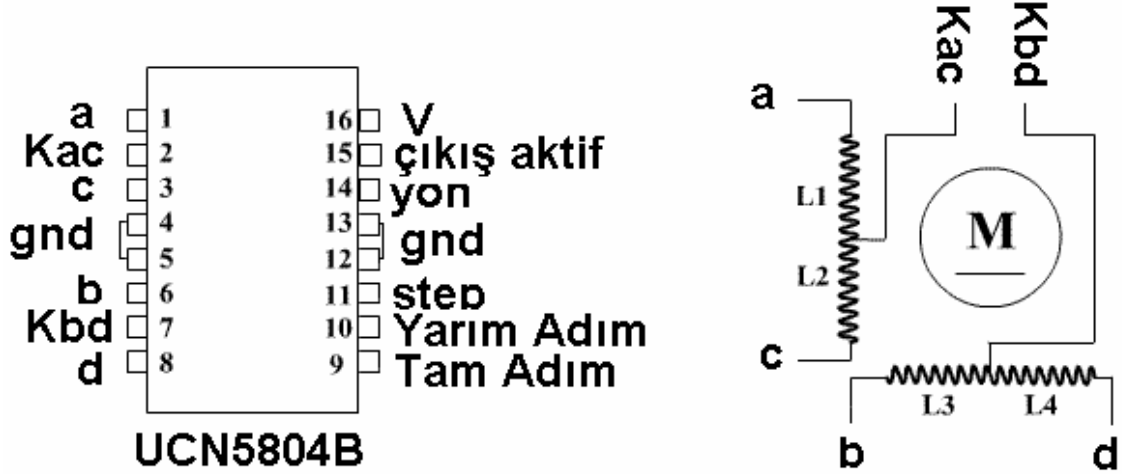


Şekil 2.27 MOSFET transistörlü unipolar motor sürücü devresi

2.2.4. UCN5804B entegresi ile unipolar step motor sürücü devresi

UCN5804B entegre devresi küçük unipolar step motorları sürmek için üretilmiştir. Bu entegre ile +35V gerilime kadar çalışan ve 1.25A akım çeken step motorlar kontrol edilebilir. Entegre devre çıkışları step motora direk olarak bağlanabilir. Buna ilaveten entegre devrenin adım girişi, yarı adım girişi ve yön girişleri de

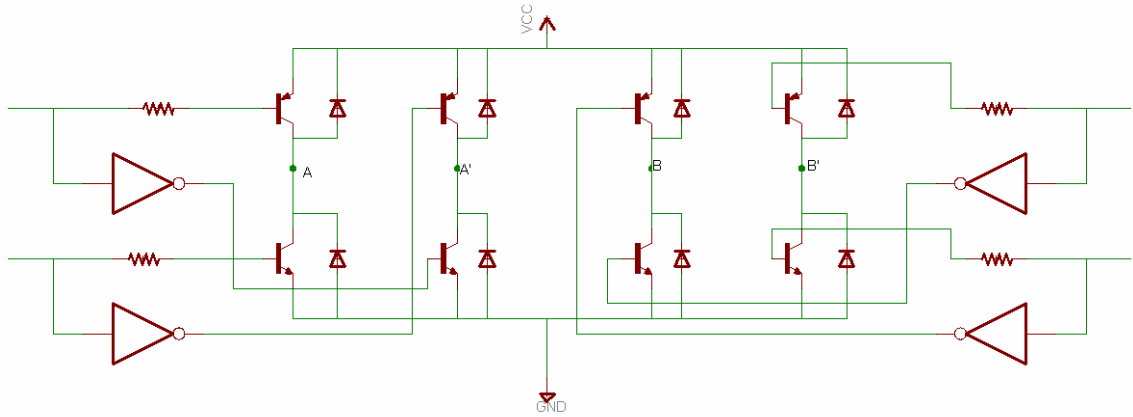
bulunmaktadır. Motorun dönmesi için entegrenin adım girişine puls verilmesi gerekmektedir. Yön girişi lojik 0 olunca motor bir yöne, lojik 1 olunca ise motor diğer yöne dönmektedir.



Şekil 2.28 UCN5804B entegresi ile unipolar step motor sürücü devresi

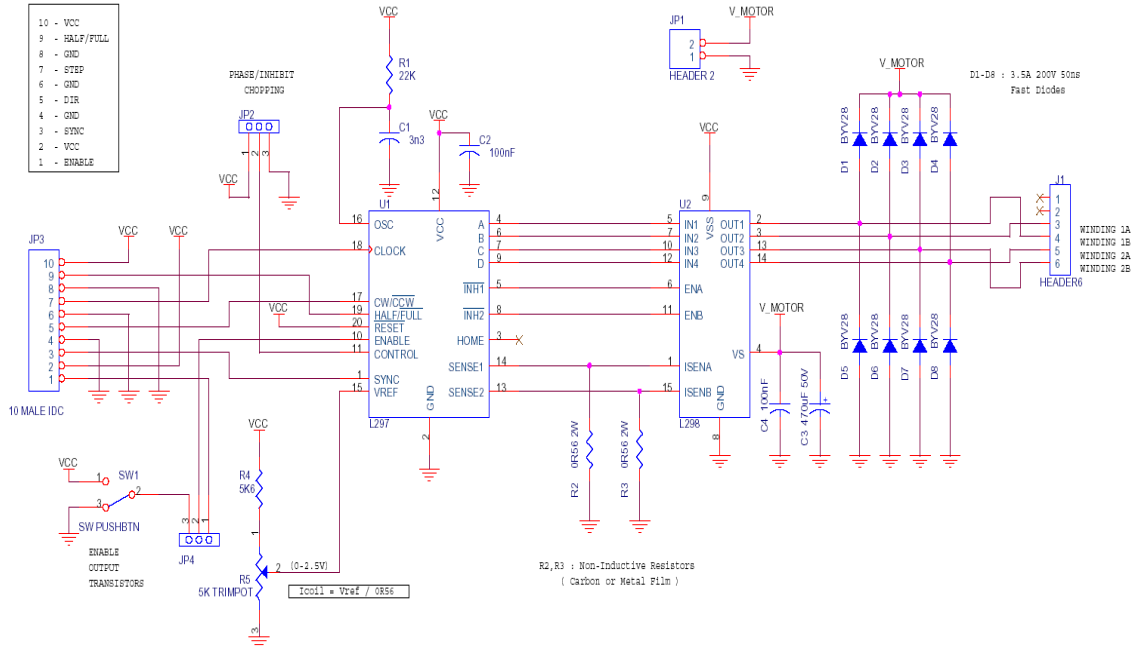
2.2.5. Transistörlü bipolar step motor sürücü devresi

Bipolar motorlarda ayrı bobinler olup bu bobinlere verilen gerilim yönünün değiştirilmesi gerekmektedir. Bipolar adım motorlarının sürüşü bundan dolayı daha karışıktır. Bu tip motorları sürmek için genellikle H-köprüsü devreleri kullanılmaktadır.



Şekil 2.29 Transistörlü bipolar step motor sürücü devresi

2.2.6. L297 + L298 entegreli bipolar step motor sürücü devresi



Şekil 2.40 L297 + L298 entegreli bipolar step motor sürücü devresi

L297 step motor kontrol entegresidir. Girişine uygulanan step ve dir sinyalleri ile çıkışında step motor faz sinyallerini üretmektedir. Entegre full-step, half-step ve wave-drive modlarında çalışabilmektedir.

L298 H-bridge sürücü entegresidir. Bipolar step motorların sürülmesi için tasarlanmıştır. Max 2A/phase akım verebilir. Girişine uygulanan faz sinyallerini çıkışa yükseltip vermektedir.

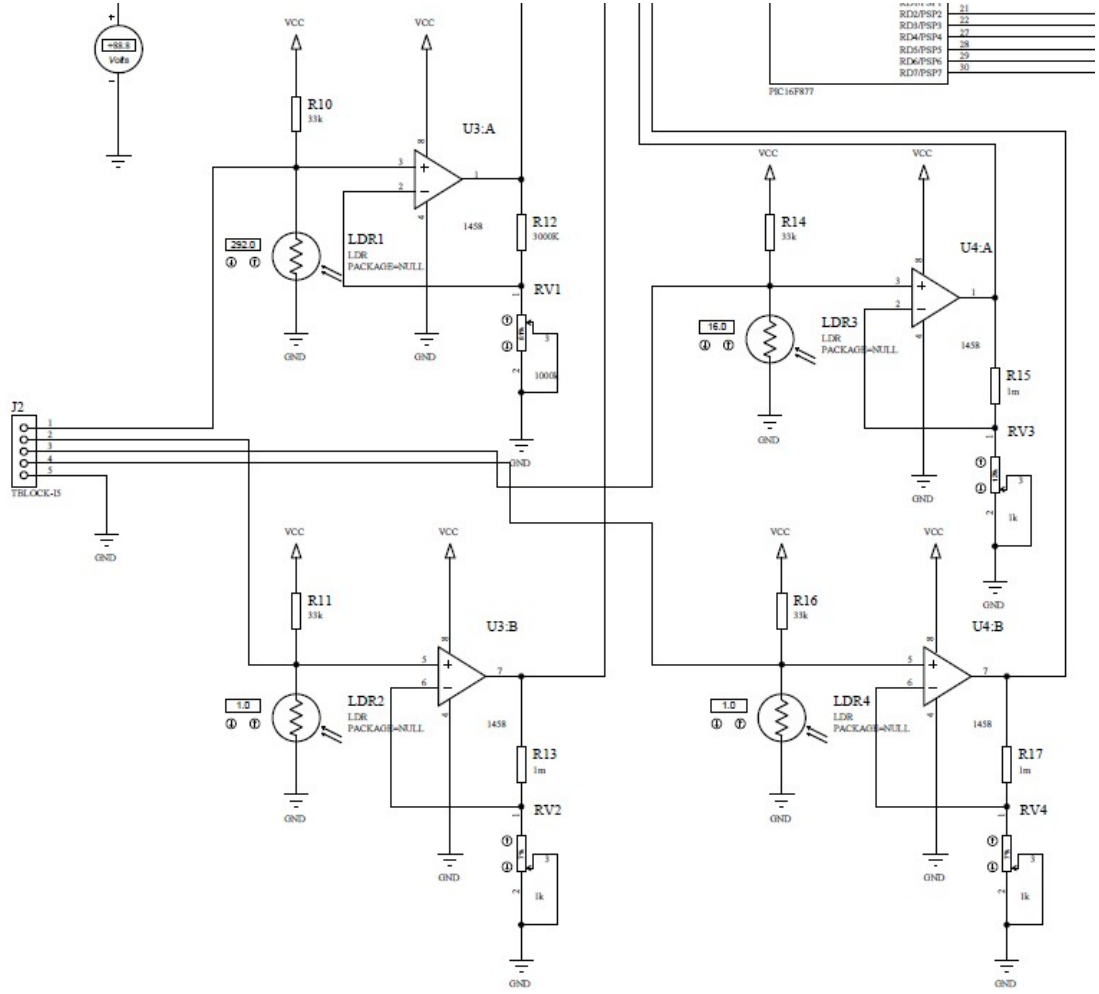
Sürücülerin özellikleri şunlardır.

- Step ve Dir sinyalleriyle çalışma
- Max 45V motor voltajı
- Max 2A faz akımı
- Full-step, Half-step ve Wave-drive modlarında çalışabilme
- Ayarlanabilir faz akımı

Bu sürücü devre chopper mantığında çalışmaktadır. Bu sayede motorlardan maksimum hızda en iyi verimi almak amaçlanmıştır.

2.2.7 Sistemdeki ana kontrol kartı

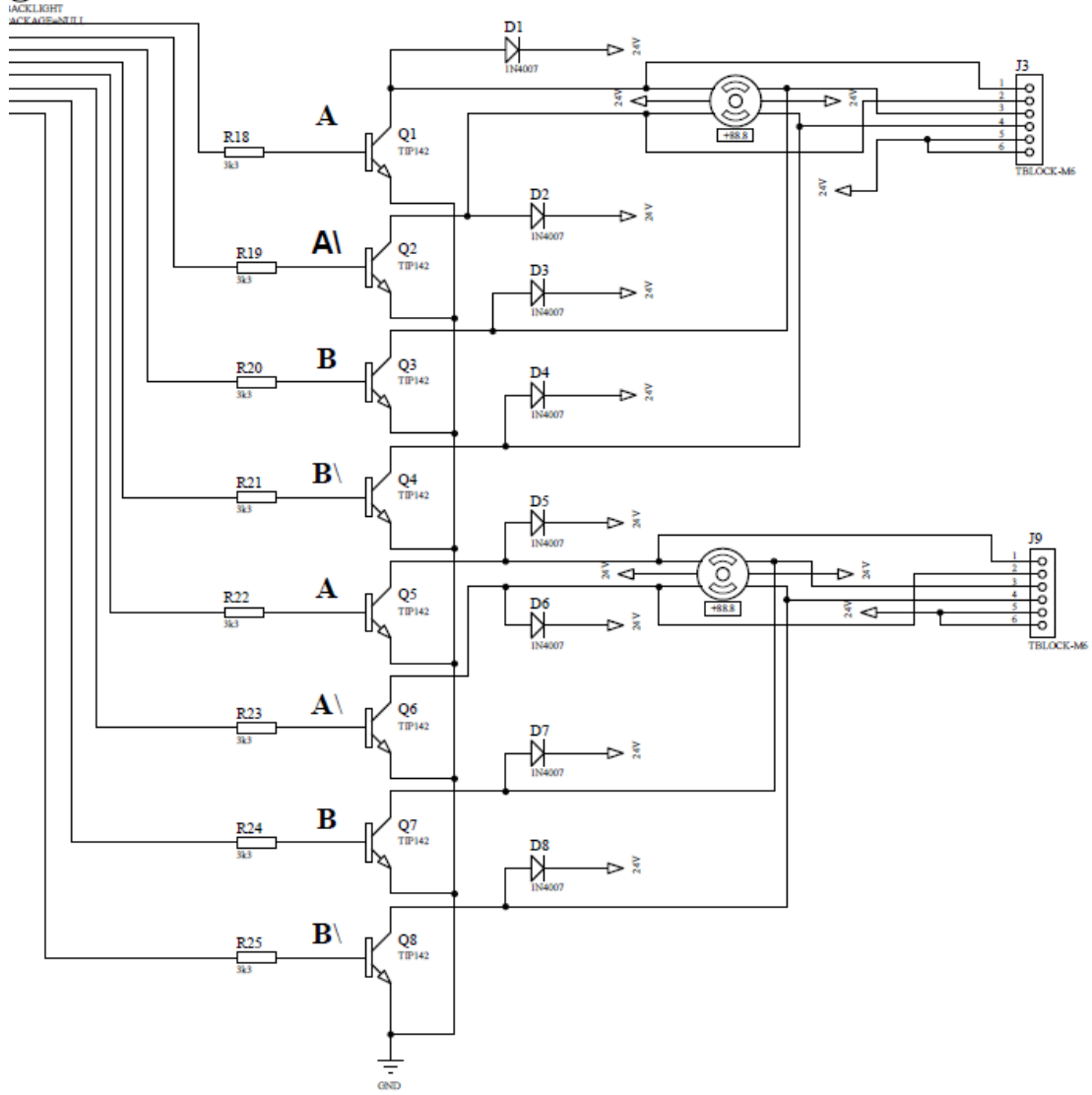
Sistemimizde 2 adet sürücü devresini, 4 adet LDR devresini, LCD ekranı, mikro kontrolörü içinde barındıran bir ana kontrol kartı mevcuttur.



Şekil 2.41 Ana kontrol kartının LDR kısmı

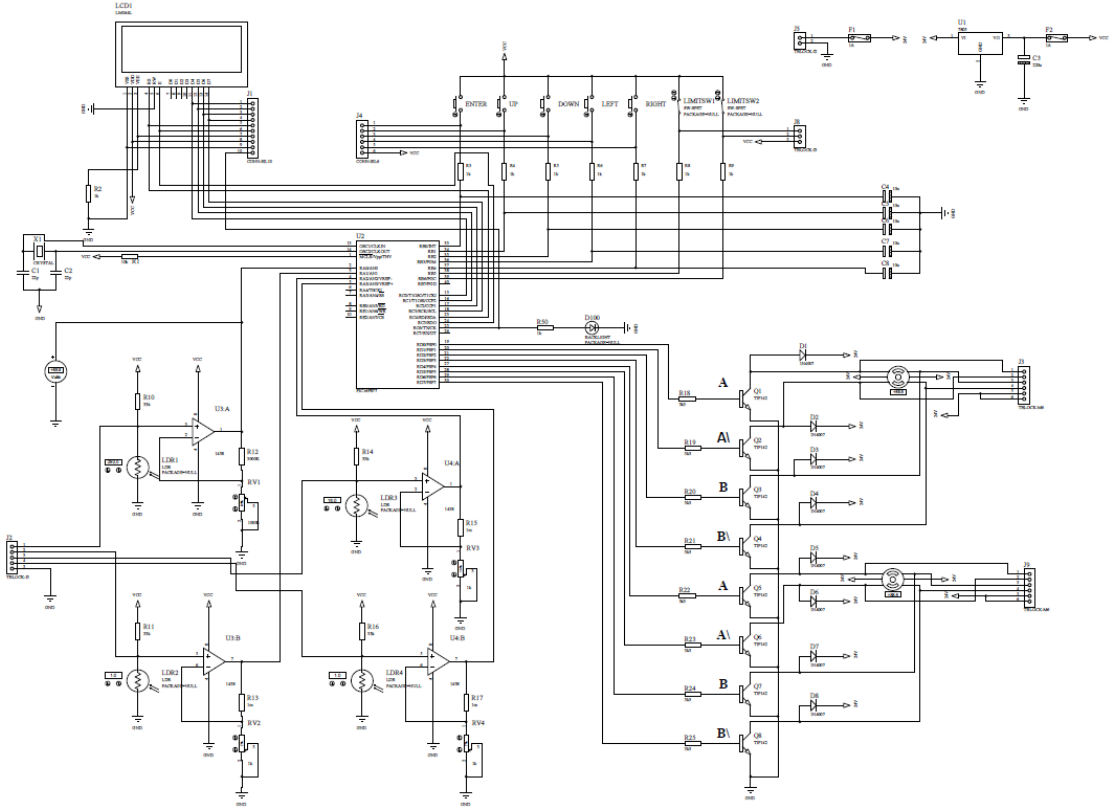
Sistemimizde güneşin yönünün belirlenmesi amacıyla 4 adet ışığa duyarlı direnç(LDR) kullanılmıştır. Tüm LDR'ler aynı karakteristiğe sahip olamayacağından trimpot ile ofset ayarının yapılması sağlanmış ve değerlerinin aynı olması sağlanmıştır. Bu LDR'lerden gelen voltaj bilgisi mikro kontrolörün analog girişlerine bağlanmıştır ve içindeki yazılım sayesinde karşılaştırma yapılması sağlanmıştır.

Ana kontrol kartı üzerinde bulunan switchler yardımıyla motor aynı zamanda el ile manüel olarak da kontrol edilebilir hale getirilmiştir. Buna ek olarak sisteme mikro limit switchlerde ilave edilmiş ve herhangi bir problemde motorun dönüş açısı sınırlandırılmıştır.

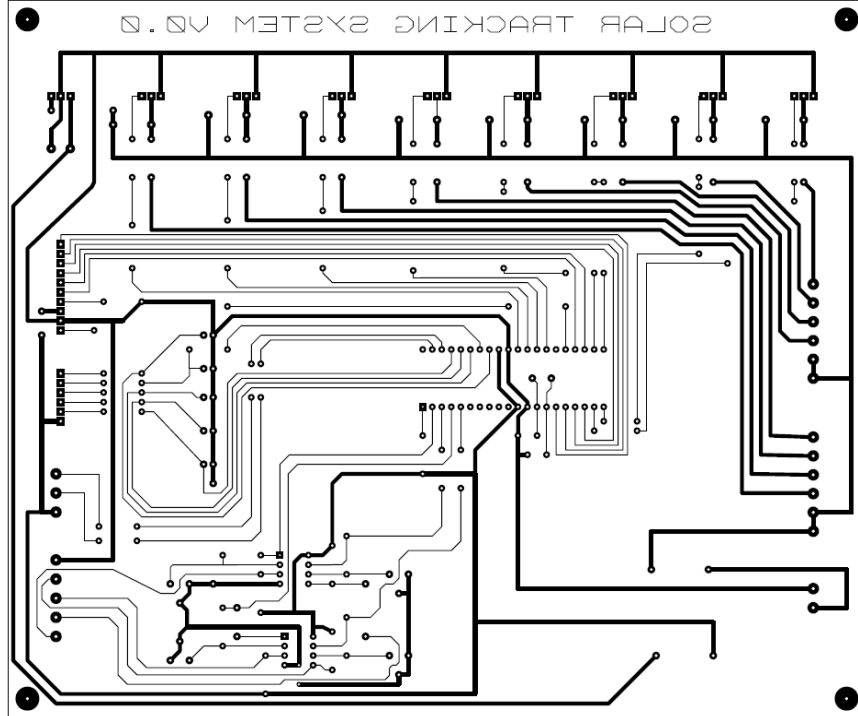


Şekil 2.44 Ana kontrol kartı sürücü devreleri

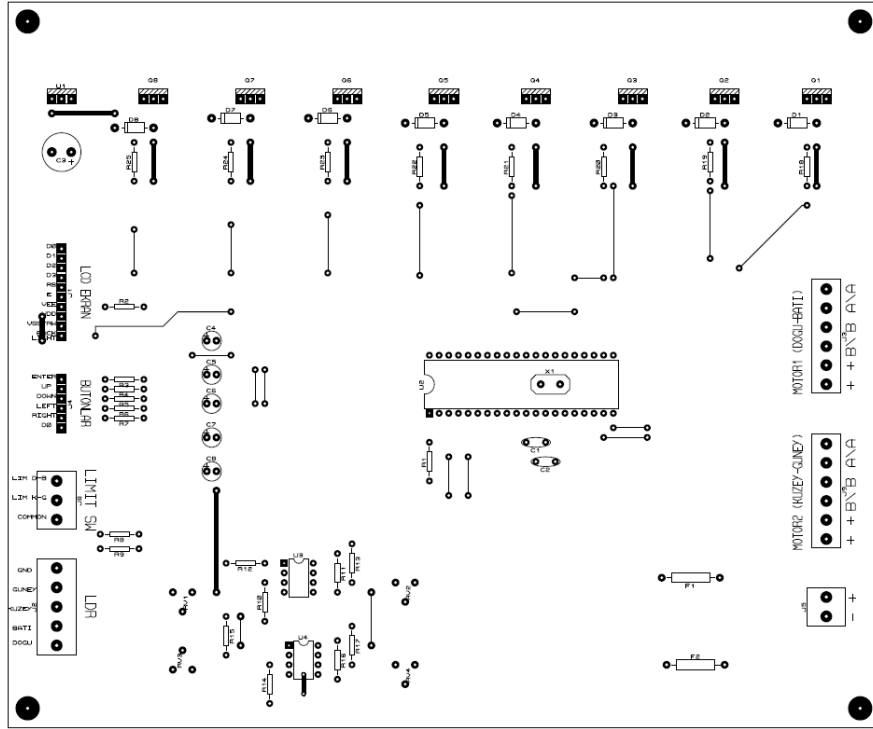
Ana kontrol kartı üzerinde motor sürücü devreleri de mevcuttur. PIC' ten gelen değerler sayesinde step motorların giriş uçlarına darbeler uygulanmakta ve dönüş hareketi motora verilmektedir.



Şekil 2.45 Ana kontrol kartı genel görünümü



Şekil 2.46 Ana kontrol kartı PCB devresinin alttan görünümü



Şekil 2.47 Ana kontrol kartı PCB devresinin üstten görünümü

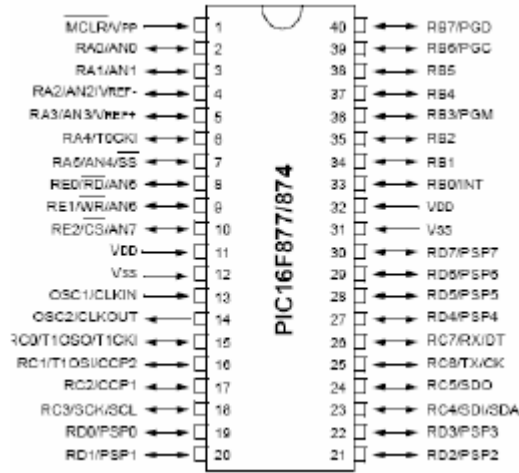
2.3. Algoritma ve program

Step motorlarımız PIC16F877 ile kontrol edilmiştir. PIC içindeki yazılım Flowcode programıyla hazırlanmıştır ve C dili ile yazılmıştır.

2.3.1 PIC16F877

PIC Serisi mikroişlemciler MICROCHIP firması tarafından geliştirilmiş ve üretim amacı çok fonksiyonlu logic uygulamalarının hızlı ve ucuz bir mikroşlemci ile yazılım yoluyla karşılanmasıdır. PIC' in kelime anlamı PERIPHERAL INTERFACE CONTROLLER Giriş Çıkış işlemcisidir.

PIC16F877 kullanıcılara yeni ve gelişmiş olanaklar sunmasıyla hemen göze çarpmaktadır. Kullanıldığı her alana en az ek donanımla adapte olabilecek giriş-çıkış portlarına sahip bir mikro denetleyici ünitesidir. 16F877'nin program belleği FlashRom olup, yüklenen program elektriksel olarak silinip yeniden yüklenebilmektedir. Çalışma hızı DC-20 Mhz olup, içerisinde üç adet zamanlayıcı ve 10 bitlik bir A/D çevirici bulunmaktadır.



Şekil 2.48 PIC16F877 entegresinin uçları

2.3.1.1. PIC16F877'nin Özellikleri

- 8K Flash program bellek
- 368 byte RAM bellek
- 256 byte EEPROM bellek
- 14 bit genişliğinde komutlar
- Kesme (interrupt) kaynağı
- 33 giriş-çıkış portları
- 3 tane zamanlayıcı devresi
- 2 tane capture/compare/PWM devresi
- 8 tane 10 bit analog-sayısal çevirici
- Senkron SSP seri port
- SPI ve I2C bus özelliği
- Paralel/Slave port
- USART seri iletişim
- 25 mA port çıkış akımı
- Bekçi köpek devresi
- Uyku modu
- +5V'da 2 mA akım, 3V'da akım 0,6 mA PIC16F877 mikro denetleyicisi 20 MHz kadar bir saat hızında çalışabilir. RISC yapına sahip olan bu mikro denetleyicinin sadece 35 tane tek kelimelik komutu vardır. PIC16F877 mikro denetleyicisinin 14 tane kesme (interrupt) kaynağı bulunur.

PIC16F877 mikro denetleyicisinin PORTA, PORTB, PORTC, PORTD VE PORTE olmak üzere 5 tane sayısal portu bulunmaktadır. PORTA pinleri analog-sayısal çevirici giriş pini vazifesini de görmektedirler.

BACAĞI İSMİ	BACAĞI NO	TANIM
OSC1/CLKIN	13	Osilatör giriş
OSC2/CLKOUT	14	Osilatör giriş
MCLR/Vpp	1	Reset girişi. Programlama girişi
RA0/AN0	2	PORTA bit 0. Analog giriş 0
RA1/AN1	3	PORTA bit 1. Analog giriş 1
RA2/AN2/Vref	4	PORTA bit 2. Analog giriş 2
RA3	5	PORTA bit 3
RA4/TOCKI	6	PORTA bit 4. TMR0 saat girişi
RA5/SS/AN4	7	PORTA bit 5. SSP slave seçme pini. Analog giriş 4
RB0/INT	34	PORTB bit 0. Dış kesme (interrupt) pin
RB1	34	PORTB bit 1
RB2	35	PORTB bit 2
RB3/PGM	36	PORTB bit 3
RB4	37	PORTB bit 4
RB5	38	PORTB bit 5
RB6/PGC	39	PORTB bit 6. Seri programlama saat pini
RB7/PGD	40	PORTB bit 7. Seri programlamada veri pini
RC0/T1OS0/TICKI	15	PORTC bit 0. Zamanlayıcı 1 osilatör çıkışı. Zamanlayıcı 1 saat girişi
RC1/T1OS1/CCP2	16	PORTC bit 1. Zamanlayıcı 1 osilatör girişi Capture2 girişi / Compare2 çıkışı. PWM2 çıkışı
RC2/CCP1	17	PORTC bit 2. Capture1 girişi / Compare 1 çıkışı. PWM1 çıkışı
RC3/SCK/SCL	18	PORTC bit 3. SPI ve I ² C seri saat giriş-çıkış
RC4/SDI/SDA	23	PORTC bit 4. SPI giriş. I ² C giriş-çıkış
RC5/SDO	24	PORTC bit 5. SPI veri çıkış
RC6/TX/CK	25	PORTC bit 6. USART transmitt, sync, saat.
RC7/RX/DI	26	PORTC bit 7. USART giriş, sync, veri.
VS	8	Toprak
VDD	19	Pozitif kaynak
RD0/PSP0	19	PORTD bit0. Paralel slave 0
RD1/PSP1	20	PORTD bit1. Paralel slave 1
RD2/PSP2	21	PORTD bit2. Paralel slave 2
RD3/PSP3	22	PORTD bit3. Paralel slave 3
RD4/PSP4	27	PORTD bit4. Paralel slave 4
RD5/PSP5	28	PORTD bit5. Paralel slave 5
RD6/PSP6	29	PORTD bit6. Paralel slave 6
RD7/PSP7	30	PORTD bit7. Paralel slave 7
RE0/RD/AN5	8	PORTE bit0. Analog giriş 5
RE1/WR/AN6	9	PORTE bit1. Analog giriş 6
RE2/CS/AN7	10	PORTE bit2. Analog giriş 7
VSS	12.3.1	Toprak
VDD	11.3.2	Pozitif kaynak

Şekil 2.49 PIC16F877 mikro denetleyici bacak tanımları

2.3.1.2. RAM Bellek

PIC16F877'nin 0x00~7Fh adres aralığına ayrılmış olan RAM belleği vardır. Bu bellek içerisindeki file registerleri içerisine yerleştirilen veriler PIC CPU'sunun çalışmasını kontrol etmektedir. File register adı verilen özel veri alanlarının dışında kalan diğer bellek alanları, normal RAM bellek olarak kullanılmaktadırlar.

bağlanır. PIC'e ilk defa enerji verildiği anda meydana gelebilecek gerilim dalgalanmaları nedeniyle, oluşabilecek istenmeyen arızaları önlemek amacıyla 100nF'lık dekuplaj kondansatörünün devreye bağlanması gerekmektedir. PIC'ler CMOS teknolojisi ile üretildiklerinden 2 ila 6 volt arasında çalışabilmektedirler. +5 V'luk bir gerilim ise ideal bir değer olmaktadır.

2.3.1.3.2. Reset Devresi

PIC mikro denetleyicisi reset yapmak için, MCLR bacağına 4.7K bir dirençle +V kaynağına bağlamamız yeterlidir. Bu devre ile otomatik reset sağlanmakta ve devreye elektrik verildiği an PIC mikro denetleyici reset olmaktadır.

2.3.1.3.3. Osilatör Devresi

Mikroişlemci ve mikro denetleyici devrelerinin çalışması için sürekli bir saat (clock) pulsu gerekir. Genellikle bu saat, hassas zamanlama uygulamalarında dıştan bir kristal bağlamakla sağlanmaktadır. Zamanlamanın hassas olması gerekmeyen bazı uygulamalarda ise mikro denetleyici içerisinde bulunan saat veya dıştan direnç ve kapasitör bağlamakla saat pulsları elde edilmektedir.

PIC mikro denetleyicilerde 5 değişik şekilde saat pulsları elde etmek mümkündür:

- Düşük güç kristal kullanarak (LP modu)
- Kristal/rezonatör kullanarak (XT modu)
- Yüksek hızlı kristal/rezonatör kullanarak (HS modu)
- Direnç/kapasitör kullanarak (RC modu)
- İç osilatörü kullanarak (sadece bazı PIC'lerle)

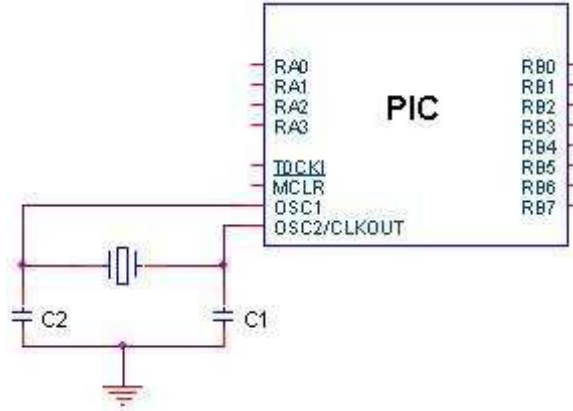
Yukarıdaki osilatör seçimleri mikro denetleyici programlanırken yapılabilmektedir.

2.3.1.3.4. Kristal Osilatör:

Kristal mikro denetleyicinin OSC1 ve OSC2 bacaklarına bağlanır ve ayrıca 2 tane de kapasitör kullanılır. Kapasitör seçimi kullanılan kristal frekansına bağlıdır. Örneğin; 4 MHz kristal kullanıyorsak C1 ve C2 kapasitörlerini 15-33 pF arasında seçebiliriz.

OSİLATÖR MODU	FREKANS	C1, C2
LP	32 KHz	68 – 100 pF
LP	200 KHz	15 – 33 pF
XT	100 KHz	100 – 150 pF
XT	2 MHz	15 – 33 pF
XT	4 MHz	15 – 33 pF
HS	4 MHz	15 – 33 pF
HS	10 MHz	15 – 33 pF

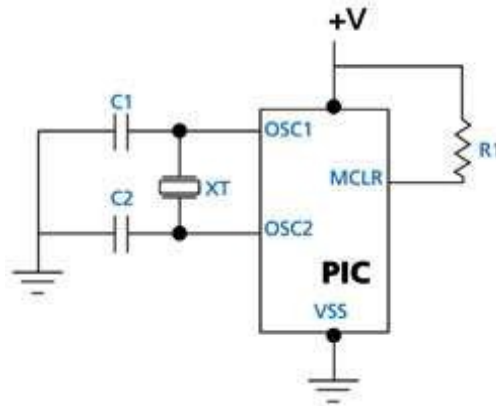
Şekil 2.51 Kristal osilatör seçim grafiği



Şekil 2.32. Kristal tabanlı PIC osilatör devresi

2.3.1.4. Minimum PIC devresi

PIC mikro denetleyicisinin çalışması için sadece bir reset devresi ve bir de osilatör devresi gerekir. Seri iletişimde zamanlama önemli olduğu için, kristal ve kapasitörler kullanılmıştır. Eğer zamanlama önemli değilse, bir RC devresi kullanılabilir.



Şekil 2.52 PIC mikro denetleyicisinin çalışması için gerekli olan minimum bağlantı

3. SONUÇ VE ÖNERİLER

Sistemimizi tasarlarken daha önceden yapılmış birçok projeyi inceledik. Bu veriler ışığında sistemin nasıl daha verimli ve daha uygun olduğuna karar verdik.

Tasarım aşamasında dikkatimizi çeken ve sistemimizin daha gelişmesini sağlayan yönler şunlardır;

3.1 Sonuçlar

- İncelenen projelerde motorların çok hızlı dönmesinden dolayı bazı yön bulma problemlerinin ortaya çıktığını gördük. Bunun için motora gönderilen sinyallerin etkin periyodunu iyi bir şekilde ayarladık.
- Karşılaştığımız bir diğer problem ise kullanılan LDR'lerin aynı karakteristikte olmayışydı. Bu problemi de Ana kontrol kartı üzerinde bulunan trimpotlar sayesinde giderdik. Bu trimpotlar sayesinde range değerini ayarlayıp LDR'lerin belirli bir ofset değerlerinin oluşmasını sağladık.
- Sistemde karşılaşılan bir diğer problem ise güneş ışınlarının sabit bir değerde olmaması problemydi. Şöyle ki LDR'lerden okunan değerlerde oluşan küçük değişimler bile motorların hareketini sağlıyordu bu da istenmeyen bir kararsızlığa sebebiyet veriyordu. Bunu PIC'in içindeki yazılım üzerinde düzeltmeler yaparak giderdik. Yaptığımız düzeltmeye göre panelimiz sadece 2 LDR arasındaki farkın belirli bir değere çıkmasıyla hareket edecek.
- Edindiğimiz bir diğer tecrübe ise Step motor kullanmak oldu çünkü yapılan birçok uygulamada DC motor kullanıldığını gördük. Fakat biz step motor kullanarak motorun her an hangi pozisyonda olduğunu bulabildik ve böylece sistemi istediğimiz pozisyona hatasız getirebildik.
- Tasarladığımız sistemde limit switchler kullanarak istenmeyen bir durum olduğunda problem oluşmaması için panelin hareketini sınırladık.

3.2 Öneriler

Projenin kapsamı aşağıda belirtilen örnekler ile daha da genişletilebilir.

- GPS sistemleri kullanılarak güneşin her an matematiksel konumunun bilinmesi sayesinde sensör kullanılmasına gerek kalmadan bulutlu bir günde bile güneşin takip edilmesi işlemi yapılabilir.
- Akşam batıya yönelmiş olan sisteme sabah ışık arkadan geleceği için güneşin bulunması sırasında sorun çıkabileceği için gece belli bir zamanda sabah güneşi karşılamak için sistemin doğu pozisyonuna alınması gerekmektedir. Bu sorunu çözebilmek için sisteme RTC eklenerek gece istediğimiz saatte sistemin doğu

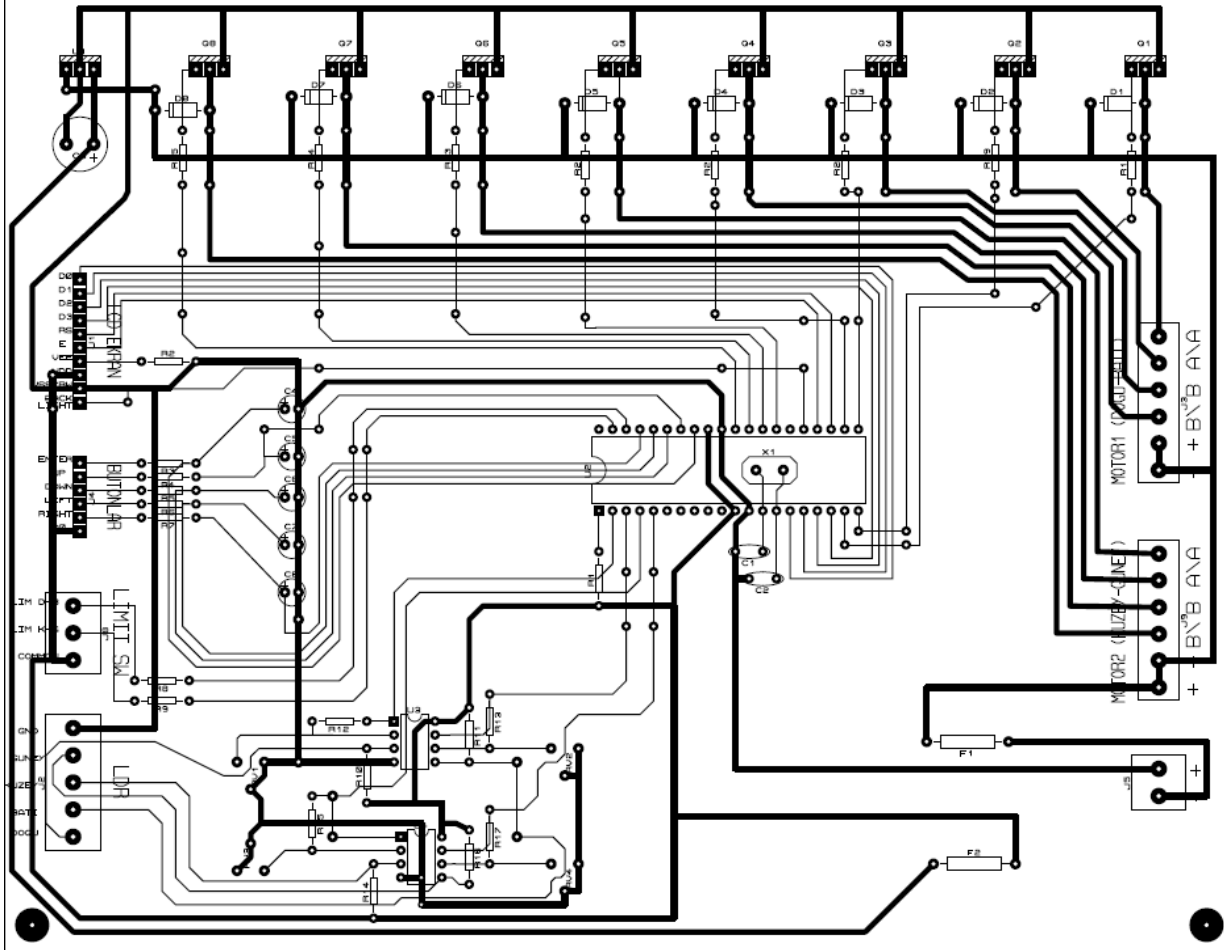
pozisyonuna yönelmesi sağlanabilir. Bunu yaparken akşam sistemin tam batı pozisyonuna gelmiş olacağı varsayılır.

- Daha kaliteli ve sağlam mekanik sistemler tasarlanarak daha büyük paneller ile güneş izlemek mümkün olabilir.

KAYNAKLAR

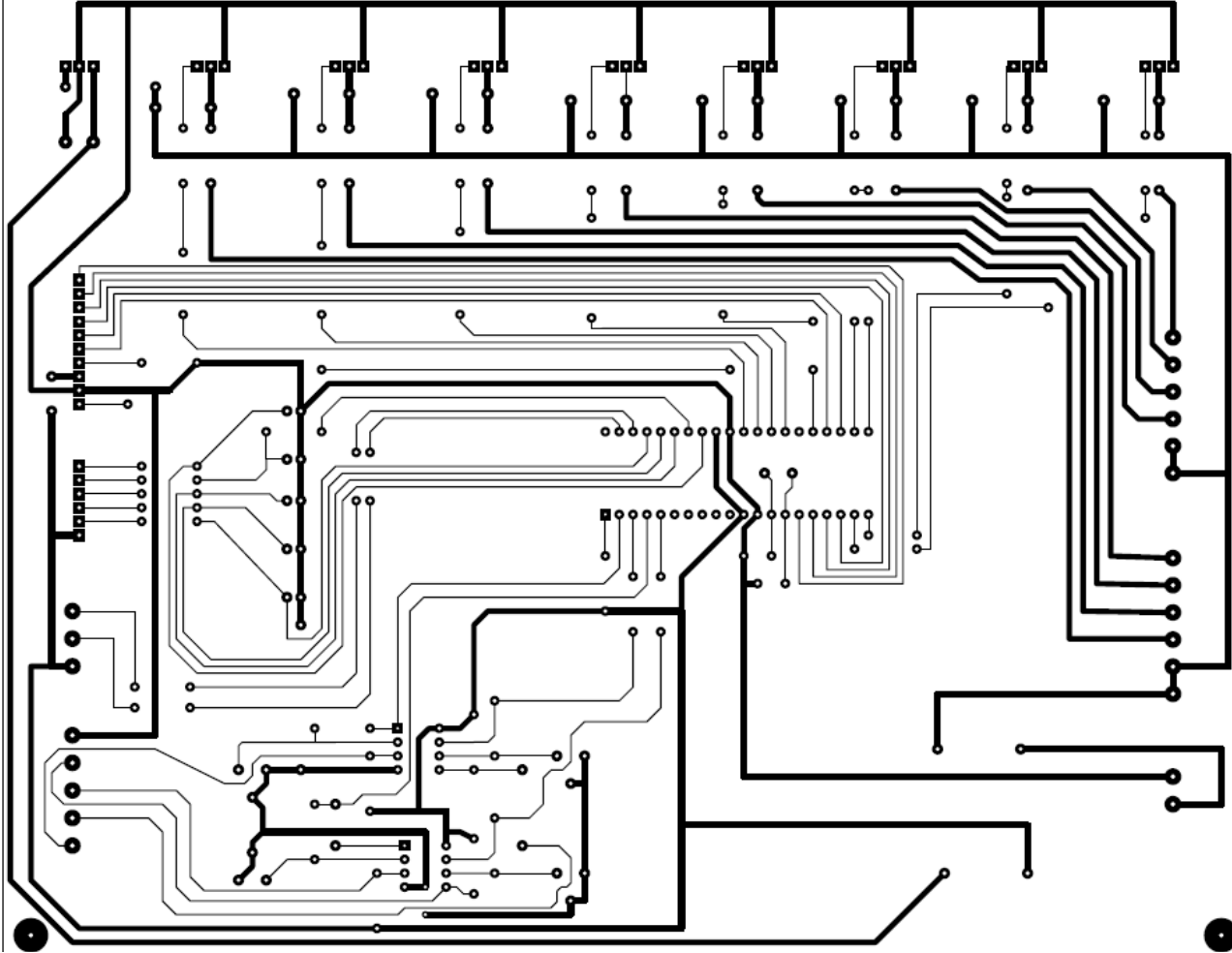
- [1] Dönmez, Ş., Özdemir, A., ”Otomatik Güneş İzleme Sistemi”
- [2] Read, M., Hay, M.,et.al, (2006), ”Design of a Solar Tracker”, Capstone Design
- [3] Lane, B.,(2008),”Solar Tracker”, EEC 517
- [4] S. J. Hamilton,(1999) “Sun-tracking solar cell array system,” University of Queensland Department of Computer Science and Electrical Engineering, Bachelors Thesis,
- [5] Microchip Inc.,(2001), “PIC16F87X Datasheet,” www.microchip.com,
- [6] M. F. Khan and R. L. Ali,(2005), “Automatic sun tracking system,” presented at the All Pakistan Engineering Conference, Islamabad, Pakistan
- [7] A.K. Saxena and V. Dutta,(1990), “A versatile microprocessor based controller for solar tracking,” in Proc. IEEE, pp. 1105 – 1109.

SOLAR TRACKING SYSTEM V0.0

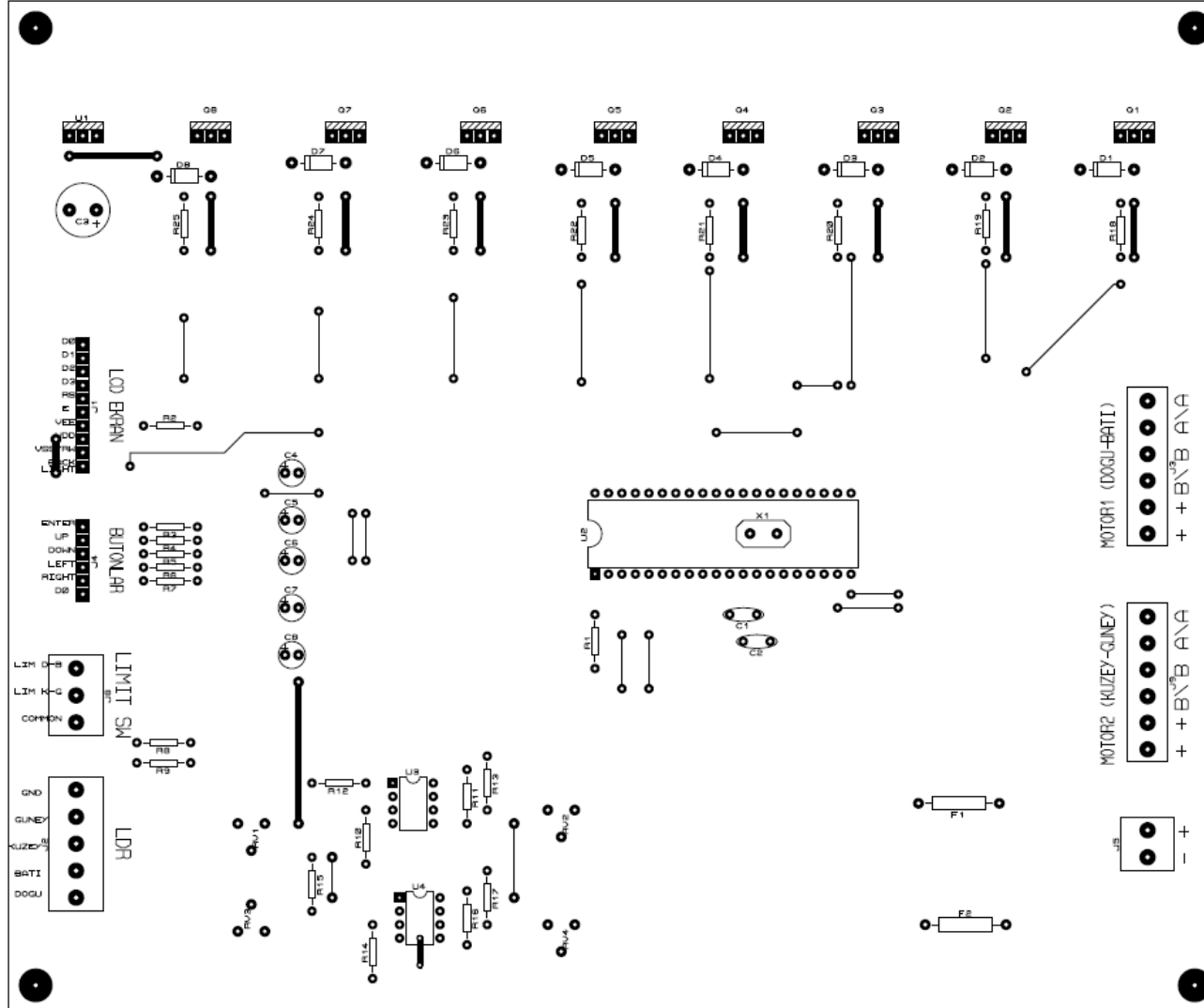


EK – 1 Ana Kontrol Kartı PCB Görünümü

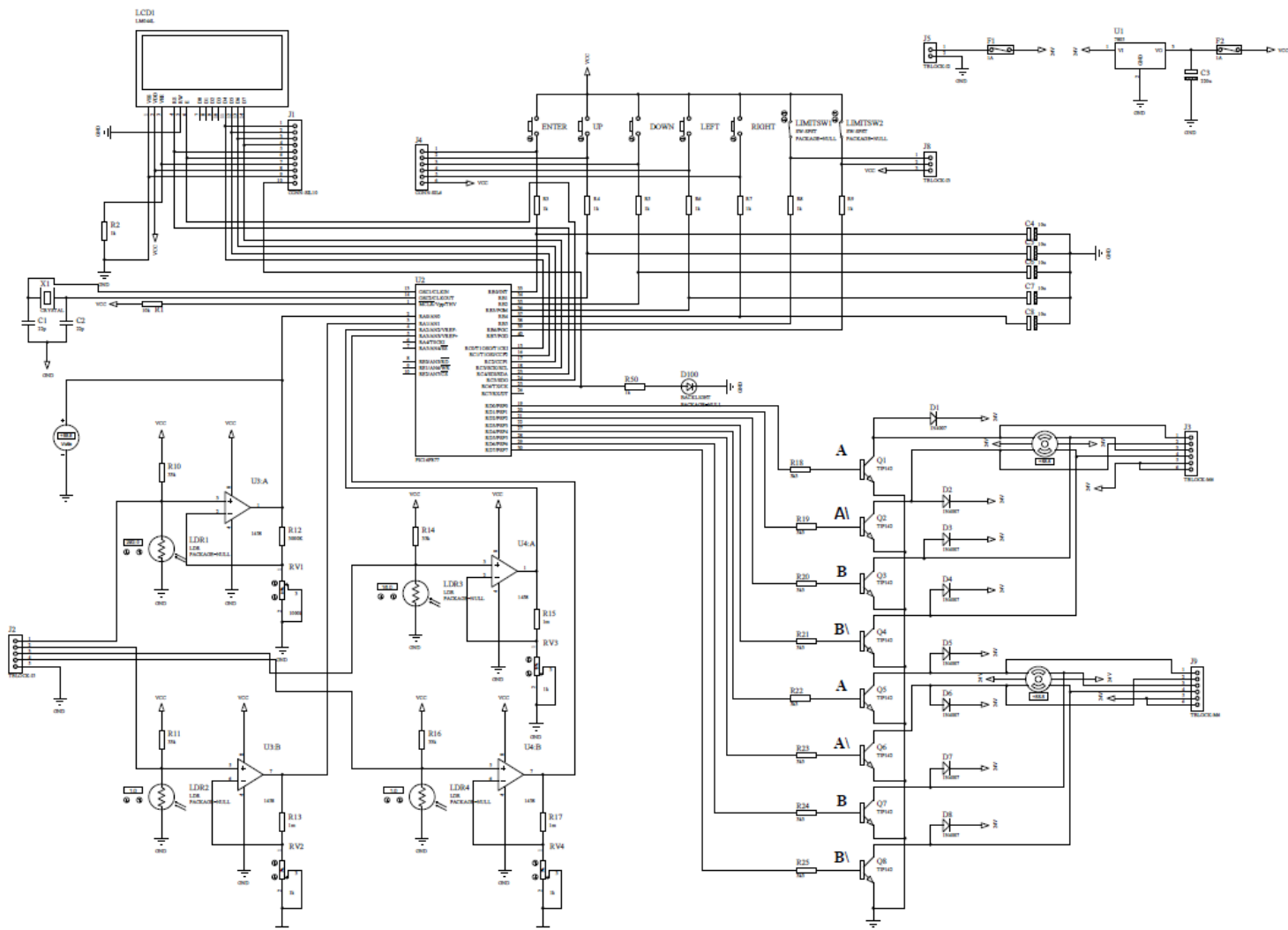
SOLAR TRACKING SYSTEM V0.0



EK – 2 Ana Kontrol Kartı PCB Altın Görünüm



EK – 3 Ana Kontrol Kartı PCB Üstten Görünüm



EK – 4 Ana kontrol kartı genel görünümü

EK – 5 Assembly Program Kodu

```
;/;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; // Code Generator: BoostC Compiler - http://www.sourceboost.com
; // Version : 6.70
; // License Type : Full License
; // Limitations : PIC12,PIC16 max code size:Unlimited, max RAM banks:Unlimited, Non commercial use
; // only
; /;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

include "P16F877A.inc"
; Heap block 0, size:111 (0x00000110 - 0x0000017E)
__HEAP_BLOCK0_BANK EQU 0x00000002
__HEAP_BLOCK0_START_OFFSET EQU 0x00000010
__HEAP_BLOCK0_END_OFFSET EQU 0x0000007E
; Heap block 1, size:96 (0x00000190 - 0x000001EF)
__HEAP_BLOCK1_BANK EQU 0x00000003
__HEAP_BLOCK1_START_OFFSET EQU 0x00000010
__HEAP_BLOCK1_END_OFFSET EQU 0x0000006F
; Heap block 2, size:80 (0x000000A0 - 0x000000EF)
__HEAP_BLOCK2_BANK EQU 0x00000001
__HEAP_BLOCK2_START_OFFSET EQU 0x00000020
__HEAP_BLOCK2_END_OFFSET EQU 0x0000006F
; Heap block 3, size:21 (0x0000005B - 0x0000006F)
__HEAP_BLOCK3_BANK EQU 0x00000000
__HEAP_BLOCK3_START_OFFSET EQU 0x0000005B
__HEAP_BLOCK3_END_OFFSET EQU 0x0000006F
__div_16_1_00003_arg_a EQU 0x00000050 ; bytes:2
__div_16_1_00003_arg_b EQU 0x00000052 ; bytes:2
CompTempVarRet132 EQU 0x00000059 ; bytes:2
__div_16_1_00003_1_r EQU 0x00000056 ; bytes:2
__div_16_1_00003_1_i EQU 0x00000058 ; bytes:1
gbl_status EQU 0x00000003 ; bytes:1
gbl_indf EQU 0x00000000 ; bytes:1
gbl_tmr0 EQU 0x00000001 ; bytes:1
gbl_pcl EQU 0x00000002 ; bytes:1
gbl_fsr EQU 0x00000004 ; bytes:1
gbl_porta EQU 0x00000005 ; bytes:1
gbl_portb EQU 0x00000006 ; bytes:1
gbl_portc EQU 0x00000007 ; bytes:1
gbl_portd EQU 0x00000008 ; bytes:1
gbl_porte EQU 0x00000009 ; bytes:1
gbl_pclath EQU 0x0000000A ; bytes:1
gbl_intcon EQU 0x0000000B ; bytes:1
gbl_pir1 EQU 0x0000000C ; bytes:1
gbl_pir2 EQU 0x0000000D ; bytes:1
gbl_tmr11 EQU 0x0000000E ; bytes:1
gbl_tmr1h EQU 0x0000000F ; bytes:1
gbl_t1con EQU 0x00000010 ; bytes:1
gbl_tmr2 EQU 0x00000011 ; bytes:1
gbl_t2con EQU 0x00000012 ; bytes:1
gbl_sspbuf EQU 0x00000013 ; bytes:1
gbl_sspcon EQU 0x00000014 ; bytes:1
gbl_ccpr1l EQU 0x00000015 ; bytes:1
gbl_ccpr1h EQU 0x00000016 ; bytes:1
gbl_ccp1con EQU 0x00000017 ; bytes:1
gbl_resta EQU 0x00000018 ; bytes:1
gbl_txreg EQU 0x00000019 ; bytes:1
gbl_rcreg EQU 0x0000001A ; bytes:1
gbl_ccpr2l EQU 0x0000001B ; bytes:1
gbl_ccpr2h EQU 0x0000001C ; bytes:1
```

```

gbl_ccp2con      EQU 0x0000001D ; bytes:1
gbl_adresh      EQU 0x0000001E ; bytes:1
gbl_adcon0      EQU 0x0000001F ; bytes:1
gbl_option_reg  EQU 0x00000081 ; bytes:1
gbl_trisa       EQU 0x00000085 ; bytes:1
gbl_trisb       EQU 0x00000086 ; bytes:1
gbl_trisc       EQU 0x00000087 ; bytes:1
gbl_trisd       EQU 0x00000088 ; bytes:1
gbl_trise       EQU 0x00000089 ; bytes:1
gbl_pie1        EQU 0x0000008C ; bytes:1
gbl_pie2        EQU 0x0000008D ; bytes:1
gbl_pcon        EQU 0x0000008E ; bytes:1
gbl_sspcon2     EQU 0x00000091 ; bytes:1
gbl_pr2         EQU 0x00000092 ; bytes:1
gbl_sspadd      EQU 0x00000093 ; bytes:1
gbl_sspstat     EQU 0x00000094 ; bytes:1
gbl_txsta       EQU 0x00000098 ; bytes:1
gbl_spbrg       EQU 0x00000099 ; bytes:1
gbl_cmcon       EQU 0x0000009C ; bytes:1
gbl_cvrcon      EQU 0x0000009D ; bytes:1
gbl_adresl      EQU 0x0000009E ; bytes:1
gbl_adcon1      EQU 0x0000009F ; bytes:1
gbl_eedata      EQU 0x0000010C ; bytes:1
gbl_eeadr       EQU 0x0000010D ; bytes:1
gbl_eedath      EQU 0x0000010E ; bytes:1
gbl_eeadrh      EQU 0x0000010F ; bytes:1
gbl_eecon1      EQU 0x0000018C ; bytes:1
gbl_eecon2      EQU 0x0000018D ; bytes:1
gbl_FCV_EW_ILERI EQU 0x00000033 ; bytes:1
gbl_FCV_DELTA_EKSI EQU 0x00000023 ; bytes:2
gbl_FCV_NS_ILERI EQU 0x00000034 ; bytes:1
gbl_FCV_DAKIKA   EQU 0x00000035 ; bytes:1
gbl_FCV_DOWN     EQU 0x00000036 ; bytes:1
gbl_FCV_ENTER    EQU 0x00000037 ; bytes:1
gbl_FCV_LDR_SOUTH EQU 0x00000025 ; bytes:2
gbl_FCV_CLOCK_1SN EQU 0x00000038 ; bytes:1
gbl_FCV_NS_GERI  EQU 0x00000039 ; bytes:1
gbl_FCV_COUNTER  EQU 0x0000003A ; bytes:1
gbl_FCV_E_W_LIMIT EQU 0x0000003B ; bytes:1
gbl_FCV_N_S_LIMIT EQU 0x0000003C ; bytes:1
gbl_FCV_PULSE    EQU 0x0000003D ; bytes:1
gbl_FCV_LCD_BACKLIGHT EQU 0x0000003E ; bytes:1
gbl_FCV_LEFT     EQU 0x0000003F ; bytes:1
gbl_FCV_RIGHT    EQU 0x00000040 ; bytes:1
gbl_FCV_RUN      EQU 0x00000041 ; bytes:1
gbl_FCV_DELTA_EW EQU 0x00000027 ; bytes:2
gbl_FCV_SANIYE   EQU 0x00000042 ; bytes:1
gbl_FCV_STEP1_EW EQU 0x00000043 ; bytes:1
gbl_FCV_LDR_EAST EQU 0x00000029 ; bytes:2
gbl_FCV_TEMP1    EQU 0x00000044 ; bytes:1
gbl_FCV_TEMP2    EQU 0x00000045 ; bytes:1
gbl_FCV_DELTA_X  EQU 0x0000002B ; bytes:2
gbl_FCV_EW_GERI  EQU 0x00000046 ; bytes:1
gbl_FCV_DELTA_NS EQU 0x0000002D ; bytes:2
gbl_FCV_TEMP3    EQU 0x00000047 ; bytes:1
gbl_FCV_STEP2_NS EQU 0x00000048 ; bytes:1
gbl_FCV_UP       EQU 0x00000049 ; bytes:1
gbl_FCV_LDR_WEST EQU 0x0000002F ; bytes:2
gbl_FCV_LDR_NORTH EQU 0x00000031 ; bytes:2
gbl_FCV_MENU     EQU 0x0000004A ; bytes:1
CompTempVar623  EQU 0x0000004B ; bytes:1
CompTempVar625  EQU 0x0000004B ; bytes:1
CompTempVar627  EQU 0x0000004B ; bytes:1
CompTempVar629  EQU 0x0000004B ; bytes:1

```

```

CompTempVar631      EQU0x0000004B ; bytes:1
CompTempVar633      EQU0x0000004B ; bytes:1
CompTempVar635      EQU0x0000004B ; bytes:1
CompTempVar637      EQU0x0000004B ; bytes:1
CompTempVar638      EQU0x0000004B ; bytes:1
CompTempVar639      EQU0x0000004B ; bytes:1
CompTempVarRet640   EQU 0x0000004D ; bytes:2
CompTempVarRet645   EQU 0x0000004D ; bytes:2
CompTempVarRet650   EQU 0x0000004D ; bytes:2
CompTempVarRet655   EQU 0x0000004D ; bytes:2
CompTempVar680      EQU0x0000004B ; bytes:1
CompTempVar681      EQU0x0000004B ; bytes:1
CompTempVar682      EQU0x0000004B ; bytes:1
CompTempVar683      EQU0x0000004B ; bytes:1
CompTempVar684      EQU0x0000004B ; bytes:1
CompTempVar685      EQU0x0000004B ; bytes:1
CompTempVar686      EQU0x0000004B ; bytes:1
CompTempVar687      EQU0x0000004B ; bytes:1
CompTempVar688      EQU0x0000004B ; bytes:1
CompTempVar689      EQU0x0000004B ; bytes:1
CompTempVar690      EQU0x0000004B ; bytes:1
CompTempVar691      EQU0x0000004B ; bytes:1
CompTempVar692      EQU0x0000004B ; bytes:1
CompTempVar693      EQU0x0000004B ; bytes:1
CompTempVar694      EQU0x0000004B ; bytes:1
CompTempVar695      EQU0x0000004B ; bytes:1
CompTempVar696      EQU0x0000004B ; bytes:1
CompTempVar697      EQU0x0000004B ; bytes:1
CompTempVar698      EQU0x0000004B ; bytes:1
CompTempVar699      EQU0x0000004B ; bytes:1
CompTempVar700      EQU0x0000004B ; bytes:1
CompTempVar701      EQU0x0000004B ; bytes:1
CompTempVar702      EQU0x0000004B ; bytes:1
CompTempVar703      EQU0x0000004B ; bytes:1
CompTempVar704      EQU0x0000004B ; bytes:1
CompTempVar705      EQU0x0000004B ; bytes:1
CompTempVar706      EQU0x0000004B ; bytes:1
CompTempVar707      EQU0x0000004B ; bytes:1
CompTempVar708      EQU0x0000004B ; bytes:1
CompTempVar709      EQU0x0000004B ; bytes:1
CompTempVar710      EQU0x0000004B ; bytes:1
CompTempVar711      EQU0x0000004B ; bytes:1
CompTempVar712      EQU0x0000004B ; bytes:1
CompTempVar713      EQU0x0000004B ; bytes:1
CompTempVar714      EQU0x0000004B ; bytes:1
CompTempVar715      EQU0x0000004B ; bytes:1
CompTempVar716      EQU0x0000004B ; bytes:1
CompTempVar717      EQU0x0000004B ; bytes:1
CompTempVar718      EQU0x0000004B ; bytes:1
CompTempVar725      EQU0x0000004B ; bytes:1
CompTempVar726      EQU0x0000004B ; bytes:1
CompTempVar727      EQU0x0000004B ; bytes:1
CompTempVar728      EQU0x0000004B ; bytes:1
CompTempVar731      EQU0x0000004B ; bytes:1
FCD_LCDDis_0001F_arg_Character EQU 0x0000004B ; bytes:1
FCD_LCDDis_00020_arg_x   EQU0x0000004B ; bytes:1
FCD_LCDDis_00020_arg_y   EQU0x0000004C ; bytes:1
FCD_LCDDis_0002C_arg_Number EQU 0x0000004B ; bytes:2
FCD_LCDDis_0002C_1_tmp_int EQU0x0000004D ; bytes:2
FCD_LCDDis_0002C_1_tmp_byte EQU 0x0000004F ; bytes:1
CompTempVar660          EQU0x00000050 ; bytes:1
CompTempVar663          EQU0x00000054 ; bytes:1
CompTempVar664          EQU0x00000050 ; bytes:1
CompTempVar665          EQU0x00000051 ; bytes:1

```



```

CompTempVar666      EQU 0x00000054 ; bytes:1
CompTempVar667      EQU 0x00000055 ; bytes:1
CompTempVar668      EQU 0x00000050 ; bytes:1
CompTempVar669      EQU 0x00000051 ; bytes:1
CompTempVar670      EQU 0x00000050 ; bytes:1
CompTempVar671      EQU 0x00000054 ; bytes:1
CompTempVar672      EQU 0x00000055 ; bytes:1
CompTempVar673      EQU 0x00000050 ; bytes:1
CompTempVar675      EQU 0x00000050 ; bytes:1
CompTempVar676      EQU 0x00000054 ; bytes:1
CompTempVar677      EQU 0x00000055 ; bytes:1
CompTempVar678      EQU 0x00000050 ; bytes:1
FCD_ADC0_S_00024_1_old_tris EQU 0x0000004B ; bytes:1
FCD_ADC0_S_00024_1_cnt     EQU 0x0000004C ; bytes:1
FCD_ADC0_R_00028_1_iRetVal EQU 0x0000004B ; bytes:2
CompTempVar643           EQU 0x0000004D ; bytes:1
CompTempVar644           EQU 0x0000004E ; bytes:1
FCD_ADC1_S_00025_1_old_tris EQU 0x0000004B ; bytes:1
FCD_ADC1_S_00025_1_cnt     EQU 0x0000004C ; bytes:1
FCD_ADC1_R_00029_1_iRetVal EQU 0x0000004B ; bytes:2
CompTempVar648           EQU 0x0000004D ; bytes:1
CompTempVar649           EQU 0x0000004E ; bytes:1
FCD_ADC2_S_00026_1_old_tris EQU 0x0000004B ; bytes:1
FCD_ADC2_S_00026_1_cnt     EQU 0x0000004C ; bytes:1
FCD_ADC2_R_0002A_1_iRetVal EQU 0x0000004B ; bytes:2
CompTempVar653           EQU 0x0000004D ; bytes:1
CompTempVar654           EQU 0x0000004E ; bytes:1
FCD_ADC3_S_00027_1_old_tris EQU 0x0000004B ; bytes:1
FCD_ADC3_S_00027_1_cnt     EQU 0x0000004C ; bytes:1
FCD_ADC3_R_0002B_1_iRetVal EQU 0x0000004B ; bytes:2
CompTempVar658           EQU 0x0000004D ; bytes:1
CompTempVar659           EQU 0x0000004E ; bytes:1
FCD_EEPROM_0003E_arg_addr EQU 0x0000004B ; bytes:2
CompTempVarRet741        EQU 0x0000004E ; bytes:1
FCD_EEPROM_0003E_1_data   EQU 0x0000004D ; bytes:1
FCD_EEPROM_00033_arg_addr EQU 0x0000004B ; bytes:2
FCD_EEPROM_00033_arg_data EQU 0x0000004D ; bytes:1
FCD_EEPROM_00033_1_bInterr_00034 EQU 0x0000004E ; bytes:1
LCD_131950_00036_arg_nIn  EQU 0x00000050 ; bytes:1
LCD_131950_00036_arg_nMask EQU 0x00000051 ; bytes:1
LCD_131950_00036_1_pt     EQU 0x00000052 ; bytes:1
LCD_131950_00036_1_outVal EQU 0x00000053 ; bytes:1
CompTempVar733           EQU 0x00000054 ; bytes:1
CompTempVar742           EQU 0x0000004B ; bytes:1
delay_10us_00000_arg_del  EQU 0x00000054 ; bytes:1
delay_ms_00000_arg_del    EQU 0x0000004D ; bytes:1
Int1Context              EQU 0x0000007F ; bytes:1
Int1BContext             EQU 0x00000020 ; bytes:3
    ORG 0x00000000
    BSF PCLATH,3
    GOTO_startup
    ORG 0x00000004
    MOVWF Int1Context
    SWAPF STATUS, W
    BCF STATUS, RP0
    BCF STATUS, RP1
    MOVWF Int1BContext
    SWAPF PCLATH, W
    MOVWF Int1BContext+D'1'
    SWAPF FSR, W
    MOVWF Int1BContext+D'2'
    BSF PCLATH,3
    BCF PCLATH,4
    GOTO interrupt

```

```

    ORG 0x00000010
delay_10us_00000
; { delay_10us ; function begin
label4026531848
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    DECFSZ delay_10us_00000_arg_del, F
    GOTO label4026531848
    RETURN
; } delay_10us function end

    ORG 0x0000001A
delay_ms_00000
; { delay_ms ; function begin
    MOVF delay_ms_00000_arg_del, F
    BTFSS STATUS,Z
    GOTO label4026531857
    RETURN
label4026531857
    MOVLW 0xF9
label4026531858
    ADDLW 0xFF
    BTFSS STATUS,Z
    GOTO label4026531858
    NOP
    DECFSZ delay_ms_00000_arg_del, F
    GOTO label4026531857
    RETURN
; } delay_ms function end

    ORG 0x00000026
FCM_CLOCK_00000
; { FCM_CLOCK ; function begin
    BCF STATUS, RP0
    BCF STATUS, RP1
    DECFSZ gbl_FCV_LCD_BACKLIGHT, W
    BTFSS STATUS,Z
    GOTO label268439882
    INCF gbl_FCV_COUNTER, W
    MOVWF gbl_FCV_COUNTER
label268439882
    INCF gbl_FCV_COUNTER, W
    BTFSS STATUS,Z
    GOTO label268439886
    CLRF gbl_FCV_LCD_BACKLIGHT
    CLRF gbl_FCV_COUNTER
label268439886
    MOVF gbl_FCV_DAKIKA, W
    XORLW 0x0F
    BTFSS STATUS,Z
    GOTO label268439891
    MOVLW 0x01
    MOVWF gbl_FCV_RUN
    CLRF gbl_FCV_DAKIKA
    GOTO label268439898
label268439891
    MOVF gbl_FCV_SANIYE, W
    XORLW 0x3C
    BTFSS STATUS,Z

```

```

GOTO label268439898
INCF gbl_FCV_DAKIKA, W
MOVWF gbl_FCV_DAKIKA
label268439898
MOVF gbl_FCV_SANIYE, W
XORLW 0x3C
BTFSS STATUS,Z
GOTO label268439902
CLRF gbl_FCV_SANIYE
GOTO label268439908
label268439902
DECF gbl_FCV_CLOCK_1SN, W
BTFSS STATUS,Z
GOTO label268439908
INCF gbl_FCV_SANIYE, W
MOVWF gbl_FCV_SANIYE
label268439908
MOVF gbl_FCV_CLOCK_1SN, W
XORLW 0x3C
BTFSS STATUS,Z
GOTO label268439912
CLRF gbl_FCV_CLOCK_1SN
RETURN
label268439912
INCF gbl_FCV_CLOCK_1SN, W
MOVWF gbl_FCV_CLOCK_1SN
RETURN
; } FCM_CLOCK function end

ORG 0x00000054
__div_16_1_00003
; { __div_16_16 ; function begin
CLRF __div_16_1_00003_1_r
CLRF __div_16_1_00003_1_r+D'1'
CLRF CompTempVarRet132
CLRF CompTempVarRet132+D'1'
CLRF __div_16_1_00003_1_i
label268435530
BTFSC __div_16_1_00003_1_i,4
RETURN
BCF STATUS,C
RLF CompTempVarRet132, F
RLF CompTempVarRet132+D'1', F
RLF __div_16_1_00003_arg_a, F
RLF __div_16_1_00003_arg_a+D'1', F
RLF __div_16_1_00003_1_r, F
RLF __div_16_1_00003_1_r+D'1', F
MOVF __div_16_1_00003_arg_b+D'1', W
SUBWF __div_16_1_00003_1_r+D'1', W
BTFSS STATUS,Z
GOTO label268435535
MOVF __div_16_1_00003_arg_b, W
SUBWF __div_16_1_00003_1_r, W
label268435535
BTFSS STATUS,C
GOTO label268435536
MOVF __div_16_1_00003_arg_b, W
SUBWF __div_16_1_00003_1_r, F
MOVF __div_16_1_00003_arg_b+D'1', W
BTFSS STATUS,C
DECF __div_16_1_00003_1_r+D'1', F
SUBWF __div_16_1_00003_1_r+D'1', F
BSF CompTempVarRet132,0
label268435536

```

```

INCF __div_16_1_00003_1_i, F
GOTO label268435530
; } __div_16_16 function end

ORG 0x00000073
LCD_131950_00036
; { LCD_131950_RawSend ; function begin
MOVWF gbl_portc, W
MOVWF LCD_131950_00036_1_outVal
BCF LCD_131950_00036_1_outVal,0
BCF LCD_131950_00036_1_outVal,1
BCF LCD_131950_00036_1_outVal,2
BCF LCD_131950_00036_1_outVal,3
BCF LCD_131950_00036_1_outVal,4
BCF LCD_131950_00036_1_outVal,5
SWAPF LCD_131950_00036_arg_nIn, W
ANDLW 0x0F
MOVWF CompTempVar733
MOVLW 0x0F
ANDWF CompTempVar733, W
MOVWF LCD_131950_00036_1_pt
BTFSC LCD_131950_00036_1_pt,0
BSF LCD_131950_00036_1_outVal,0
BTFSC LCD_131950_00036_1_pt,1
BSF LCD_131950_00036_1_outVal,1
BTFSC LCD_131950_00036_1_pt,2
BSF LCD_131950_00036_1_outVal,2
BTFSC LCD_131950_00036_1_pt,3
BSF LCD_131950_00036_1_outVal,3
MOVWF LCD_131950_00036_arg_nMask, F
BTFSS STATUS,Z
BSF LCD_131950_00036_1_outVal,4
MOVWF LCD_131950_00036_1_outVal, W
MOVWF gbl_portc
MOVLW 0x0A
MOVWF delay_10us_00000_arg_del
CALL delay_10us_00000
BSF gbl_portc,5
MOVLW 0x0A
MOVWF delay_10us_00000_arg_del
CALL delay_10us_00000
BCF gbl_portc,5
MOVLW 0x0F
ANDWF LCD_131950_00036_arg_nIn, W
MOVWF LCD_131950_00036_1_pt
MOVLW 0x0A
MOVWF delay_10us_00000_arg_del
CALL delay_10us_00000
MOVWF gbl_portc, W
MOVWF LCD_131950_00036_1_outVal
BCF LCD_131950_00036_1_outVal,0
BCF LCD_131950_00036_1_outVal,1
BCF LCD_131950_00036_1_outVal,2
BCF LCD_131950_00036_1_outVal,3
BCF LCD_131950_00036_1_outVal,4
BCF LCD_131950_00036_1_outVal,5
BTFSC LCD_131950_00036_1_pt,0
BSF LCD_131950_00036_1_outVal,0
BTFSC LCD_131950_00036_1_pt,1
BSF LCD_131950_00036_1_outVal,1
BTFSC LCD_131950_00036_1_pt,2
BSF LCD_131950_00036_1_outVal,2
BTFSC LCD_131950_00036_1_pt,3
BSF LCD_131950_00036_1_outVal,3

```

```

MOVWF LCD_131950_00036_arg_nMask, F
BTFSS STATUS, Z
BSF LCD_131950_00036_1_outVal, 4
MOVWF LCD_131950_00036_1_outVal, W
MOVWF gbl_portc
MOVLW 0x0A
MOVWF delay_10us_00000_arg_del
CALL delay_10us_00000
BSF gbl_portc, 5
MOVLW 0x0A
MOVWF delay_10us_00000_arg_del
CALL delay_10us_00000
BCF gbl_portc, 5
MOVLW 0x0A
MOVWF delay_10us_00000_arg_del
CALL delay_10us_00000
RETURN
; } LCD_131950_RawSend function end

ORG 0x000000BD
FCD_LCDDis_0002C
; { FCD_LCDDisplay0_PrintNumber ; function begin
BTFSS FCD_LCDDis_0002C_arg_Number+D'1', 7
GOTO label268441491
MOVLW 0x2D
MOVWF LCD_131950_00036_arg_nIn
MOVLW 0x10
MOVWF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVWF FCD_LCDDis_0002C_arg_Number, W
SUBLW 0x00
MOVWF CompTempVar660
COMF FCD_LCDDis_0002C_arg_Number+D'1', F
BTFSS STATUS, C
INCF FCD_LCDDis_0002C_arg_Number+D'1', F
MOVWF CompTempVar660, W
MOVWF FCD_LCDDis_0002C_arg_Number
label268441491
MOVWF FCD_LCDDis_0002C_arg_Number, W
MOVWF FCD_LCDDis_0002C_1_tmp_int
MOVWF FCD_LCDDis_0002C_arg_Number+D'1', W
MOVWF FCD_LCDDis_0002C_1_tmp_int+D'1'
MOVLW 0x27
SUBWF FCD_LCDDis_0002C_arg_Number+D'1', W
BTFSS STATUS, Z
GOTO label268441505
MOVLW 0x10
SUBWF FCD_LCDDis_0002C_arg_Number, W
label268441505
BTFSS STATUS, C
GOTO label268441525
BTFSS FCD_LCDDis_0002C_arg_Number+D'1', 7
GOTO label268441525
CLRF CompTempVar663
MOVWF FCD_LCDDis_0002C_1_tmp_int, W
MOVWF __div_16_1_00003_arg_a
MOVWF FCD_LCDDis_0002C_1_tmp_int+D'1', W
MOVWF __div_16_1_00003_arg_a+D'1'
BTFSS FCD_LCDDis_0002C_1_tmp_int+D'1', 7
GOTO label268441515
COMF __div_16_1_00003_arg_a, F
COMF __div_16_1_00003_arg_a+D'1', F
INCF __div_16_1_00003_arg_a, F
BTFSS STATUS, Z

```

```

INCF __div_16_1_00003_arg_a+D'1', F
INCF CompTempVar663, F
label268441515
MOVLW 0x10
MOVWF __div_16_1_00003_arg_b
MOVLW 0x27
MOVWF __div_16_1_00003_arg_b+D'1'
CALL __div_16_1_00003
MOVF CompTempVarRet132, W
MOVWF FCD_LCDDis_0002C_1_tmp_byte
BTFSS CompTempVar663,0
GOTO label268441517
COMF FCD_LCDDis_0002C_1_tmp_byte, F
INCF FCD_LCDDis_0002C_1_tmp_byte, F
label268441517
MOVF FCD_LCDDis_0002C_1_tmp_byte, W
ADDLW 0x30
MOVWF LCD_131950_00036_arg_nIn
MOVLW 0x10
MOVWF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
label268441524
MOVF FCD_LCDDis_0002C_1_tmp_byte, W
SUBLW 0x00
BTFSC STATUS,C
GOTO label268441525
MOVLW 0x10
SUBWF FCD_LCDDis_0002C_1_tmp_int, W
MOVWF CompTempVar664
MOVLW 0x27
BTFSS STATUS,C
MOVLW 0x28
SUBWF FCD_LCDDis_0002C_1_tmp_int+D'1', W
MOVWF CompTempVar665
MOVF CompTempVar664, W
MOVWF FCD_LCDDis_0002C_1_tmp_int
MOVF CompTempVar665, W
MOVWF FCD_LCDDis_0002C_1_tmp_int+D'1'
DECF FCD_LCDDis_0002C_1_tmp_byte, F
GOTO label268441524
label268441525
MOVLW 0x03
SUBWF FCD_LCDDis_0002C_arg_Number+D'1', W
BTFSS STATUS,Z
GOTO label268441533
MOVLW 0xE8
SUBWF FCD_LCDDis_0002C_arg_Number, W
label268441533
BTFSS STATUS,C
GOTO label268441553
BTFSC FCD_LCDDis_0002C_arg_Number+D'1',7
GOTO label268441553
CLRF CompTempVar667
MOVF FCD_LCDDis_0002C_1_tmp_int, W
MOVWF __div_16_1_00003_arg_a
MOVF FCD_LCDDis_0002C_1_tmp_int+D'1', W
MOVWF __div_16_1_00003_arg_a+D'1'
BTFSS FCD_LCDDis_0002C_1_tmp_int+D'1',7
GOTO label268441543
COMF __div_16_1_00003_arg_a, F
COMF __div_16_1_00003_arg_a+D'1', F
INCF __div_16_1_00003_arg_a, F
BTFSC STATUS,Z
INCF __div_16_1_00003_arg_a+D'1', F

```

```

INCF CompTempVar667, F
label268441543
MOVLW 0xE8
MOVWF __div_16_1_00003_arg_b
MOVLW 0x03
MOVWF __div_16_1_00003_arg_b+D'1'
CALL __div_16_1_00003
MOVF CompTempVarRet132, W
MOVWF CompTempVar666
BTFSS CompTempVar667,0
GOTO label268441545
COMF CompTempVar666, F
INCF CompTempVar666, F
label268441545
MOVF CompTempVar666, W
MOVWF FCD_LCDDis_0002C_1_tmp_byte
ADDLW 0x30
MOVWF LCD_131950_00036_arg_nIn
MOVLW 0x10
MOVWF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
label268441552
MOVF FCD_LCDDis_0002C_1_tmp_byte, W
SUBLW 0x00
BTFSC STATUS,C
GOTO label268441553
MOVLW 0xE8
SUBWF FCD_LCDDis_0002C_1_tmp_int, W
MOVWF CompTempVar668
MOVLW 0x03
BTFSS STATUS,C
MOVLW 0x04
SUBWF FCD_LCDDis_0002C_1_tmp_int+D'1', W
MOVWF CompTempVar669
MOVF CompTempVar668, W
MOVWF FCD_LCDDis_0002C_1_tmp_int
MOVF CompTempVar669, W
MOVWF FCD_LCDDis_0002C_1_tmp_int+D'1'
DECFS FCD_LCDDis_0002C_1_tmp_byte, F
GOTO label268441552
label268441553
MOVF FCD_LCDDis_0002C_arg_Number+D'1', W
XORLW 0x80
MOVWF CompTempVar670
MOVLW 0x80
SUBWF CompTempVar670, W
BTFSS STATUS,Z
GOTO label268441563
MOVLW 0x64
SUBWF FCD_LCDDis_0002C_arg_Number, W
label268441563
BTFSS STATUS,C
GOTO label268441583
CLRF CompTempVar672
MOVF FCD_LCDDis_0002C_1_tmp_int, W
MOVWF __div_16_1_00003_arg_a
MOVF FCD_LCDDis_0002C_1_tmp_int+D'1', W
MOVWF __div_16_1_00003_arg_a+D'1'
BTFSS FCD_LCDDis_0002C_1_tmp_int+D'1',7
GOTO label268441573
COMF __div_16_1_00003_arg_a, F
COMF __div_16_1_00003_arg_a+D'1', F
INCF __div_16_1_00003_arg_a, F
BTFSC STATUS,Z

```

```

INCF __div_16_1_00003_arg_a+D'1', F
INCF CompTempVar672, F
label268441573
MOVLW 0x64
MOVWF __div_16_1_00003_arg_b
CLRF __div_16_1_00003_arg_b+D'1'
CALL __div_16_1_00003
MOVF CompTempVarRet132, W
MOVWF CompTempVar671
BTFSS CompTempVar672,0
GOTO label268441575
COMF CompTempVar671, F
INCF CompTempVar671, F
label268441575
MOVF CompTempVar671, W
MOVWF FCD_LCDDis_0002C_1_tmp_byte
ADDLW 0x30
MOVWF LCD_131950_00036_arg_nIn
MOVLW 0x10
MOVWF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
label268441582
MOVF FCD_LCDDis_0002C_1_tmp_byte, W
SUBLW 0x00
BTFSC STATUS,C
GOTO label268441583
MOVLW 0x64
SUBWF FCD_LCDDis_0002C_1_tmp_int, W
MOVWF CompTempVar673
BTFSS STATUS,C
DECF FCD_LCDDis_0002C_1_tmp_int+D'1', F
MOVF CompTempVar673, W
MOVWF FCD_LCDDis_0002C_1_tmp_int
DECF FCD_LCDDis_0002C_1_tmp_byte, F
GOTO label268441582
label268441583
MOVF FCD_LCDDis_0002C_arg_Number+D'1', W
XORLW 0x80
MOVWF CompTempVar675
MOVLW 0x80
SUBWF CompTempVar675, W
BTFSS STATUS,Z
GOTO label268441593
MOVLW 0x0A
SUBWF FCD_LCDDis_0002C_arg_Number, W
label268441593
BTFSS STATUS,C
GOTO label268441613
CLRF CompTempVar677
MOVF FCD_LCDDis_0002C_1_tmp_int, W
MOVWF __div_16_1_00003_arg_a
MOVF FCD_LCDDis_0002C_1_tmp_int+D'1', W
MOVWF __div_16_1_00003_arg_a+D'1'
BTFSS FCD_LCDDis_0002C_1_tmp_int+D'1',7
GOTO label268441603
COMF __div_16_1_00003_arg_a, F
COMF __div_16_1_00003_arg_a+D'1', F
INCF __div_16_1_00003_arg_a, F
BTFSC STATUS,Z
INCF __div_16_1_00003_arg_a+D'1', F
INCF CompTempVar677, F
label268441603
MOVLW 0x0A
MOVWF __div_16_1_00003_arg_b

```



```

CLRF __div_16_1_00003_arg_b+D'1'
CALL __div_16_1_00003
MOVF CompTempVarRet132, W
MOVWF CompTempVar676
BTFSS CompTempVar677,0
GOTO label268441605
COMF CompTempVar676, F
INCF CompTempVar676, F
label268441605
MOVF CompTempVar676, W
MOVWF FCD_LCDDis_0002C_1_tmp_byte
ADDLW 0x30
MOVWF LCD_131950_00036_arg_nIn
MOVLW 0x10
MOVWF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
label268441612
MOVF FCD_LCDDis_0002C_1_tmp_byte, W
SUBLW 0x00
BTFSC STATUS,C
GOTO label268441613
MOVLW 0x0A
SUBWF FCD_LCDDis_0002C_1_tmp_int, W
MOVWF CompTempVar678
BTFSS STATUS,C
DECF FCD_LCDDis_0002C_1_tmp_int+D'1', F
MOVF CompTempVar678, W
MOVWF FCD_LCDDis_0002C_1_tmp_int
DECF FCD_LCDDis_0002C_1_tmp_byte, F
GOTO label268441612
label268441613
MOVF FCD_LCDDis_0002C_1_tmp_int, W
ADDLW 0x30
MOVWF LCD_131950_00036_arg_nIn
MOVLW 0x10
MOVWF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
RETURN
;} FCD_LCDDisplay0_PrintNumber function end

ORG 0x000001B8
FCD_LCDDis_00021
; { FCD_LCDDisplay0_Clear ; function begin
MOVLW 0x01
BCF STATUS, RP0
BCF STATUS, RP1
MOVWF LCD_131950_00036_arg_nIn
CLRF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVLW 0x02
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x02
MOVWF LCD_131950_00036_arg_nIn
CLRF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVLW 0x02
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
RETURN
;} FCD_LCDDisplay0_Clear function end

ORG 0x000001C9
FCD_LCDDis_00020

```

```

; { FCD_LCDDisplay0_Cursor ; function begin
MOVWF FCD_LCDDis_00020_arg_y, F
BTFSS STATUS, Z
GOTO label268441464
MOVLW 0x80
MOVWF FCD_LCDDis_00020_arg_y
GOTO label268441475
label268441464
DECWF FCD_LCDDis_00020_arg_y, W
BTFSS STATUS, Z
GOTO label268441468
MOVLW 0xC0
MOVWF FCD_LCDDis_00020_arg_y
GOTO label268441475
label268441468
MOVWF FCD_LCDDis_00020_arg_y, W
XORLW 0x02
BTFSS STATUS, Z
GOTO label268441472
MOVLW 0x90
MOVWF FCD_LCDDis_00020_arg_y
GOTO label268441475
label268441472
MOVLW 0xD0
MOVWF FCD_LCDDis_00020_arg_y
label268441475
MOVWF FCD_LCDDis_00020_arg_x, W
ADDWF FCD_LCDDis_00020_arg_y, W
MOVWF LCD_131950_00036_arg_nIn
CLRF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVLW 0x02
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
RETURN
; } FCD_LCDDisplay0_Cursor function end

ORG 0x000001E7
FCD_LCDDis_0001F
; { FCD_LCDDisplay0_PrintASCII ; function begin
MOVWF FCD_LCDDis_0001F_arg_Character, W
MOVWF LCD_131950_00036_arg_nIn
MOVLW 0x10
MOVWF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
RETURN
; } FCD_LCDDisplay0_PrintASCII function end

ORG 0x000001ED
FCD_LCDDis_0001E
; { FCD_LCDDisplay0_Start ; function begin
BSF STATUS, RP0
BCF STATUS, RP1
BCF gbl_trisc, 0
BCF gbl_trisc, 1
BCF gbl_trisc, 2
BCF gbl_trisc, 3
BCF gbl_trisc, 4
BCF gbl_trisc, 5
MOVLW 0x0C
BCF STATUS, RP0
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x33

```

```

MOVWF LCD_131950_00036_arg_nIn
CLRF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVLW 0x02
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x32
MOVWF LCD_131950_00036_arg_nIn
CLRF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVLW 0x02
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x2C
MOVWF LCD_131950_00036_arg_nIn
CLRF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVLW 0x02
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x06
MOVWF LCD_131950_00036_arg_nIn
CLRF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVLW 0x02
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x0C
MOVWF LCD_131950_00036_arg_nIn
CLRF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVLW 0x02
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x01
MOVWF LCD_131950_00036_arg_nIn
CLRF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVLW 0x02
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x02
MOVWF LCD_131950_00036_arg_nIn
CLRF LCD_131950_00036_arg_nMask
CALL LCD_131950_00036
MOVLW 0x02
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
RETURN
; } FCD_LCDDisplay0_Start function end

ORG 0x0000022B
FCD_EEPROM_00033
; { FCD_EEPROM0_WriteEEPROM ; function begin
label268441795
BSF STATUS, RP0
BSF STATUS, RP1
BTFSC gbl_eecon1,1
GOTO label268441795
BCF STATUS, RP0
BCF STATUS, RP1
MOVF FCD_EEPROM_00033_arg_addr, W
BSF STATUS, RP1
MOVWF gbl_eeadr

```

```

BCF STATUS, RP1
MOVWF FCD_EEPROM_00033_arg_data, W
BSF STATUS, RP1
MOVWF gbl_eedata
BSF STATUS, RP0
BCF gbl_eecon1,7
BSF gbl_eecon1,2
MOVLW 0x80
ANDWF gbl_intcon, W
BCF STATUS, RP0
BCF STATUS, RP1
MOVWF FCD_EEPROM_00033_1_bInterr_00034
BCF gbl_intcon,7
MOVLW 0x55
BSF STATUS, RP0
BSF STATUS, RP1
MOVWF gbl_eecon2
MOVLW 0xAA
MOVWF gbl_eecon2
BSF gbl_eecon1,1
label268441807
BTFSC gbl_eecon1,1
GOTO label268441807
BCF STATUS, RP0
BCF STATUS, RP1
MOVWF FCD_EEPROM_00033_1_bInterr_00034, F
BTFSC STATUS,Z
GOTO label268441810
BSF gbl_intcon,7
label268441810
BSF STATUS, RP0
BSF STATUS, RP1
BCF gbl_eecon1,2
RETURN
; } FCD_EEPROM0_WriteEEPROM function end

```

```

ORG 0x00000254
FCD_ADC3_S_00027
; { FCD_ADC3_SampleADC ; function begin
MOVLW 0x04
BSF STATUS, RP0
BCF STATUS, RP1
MOVWF gbl_adcon1
MOVWF gbl_trisa, W
BCF STATUS, RP0
MOVWF FCD_ADC3_S_00027_1_old_tris
MOVLW 0x08
BSF STATUS, RP0
IORWF gbl_trisa, W
MOVWF gbl_trisa
MOVLW 0x99
BCF STATUS, RP0
MOVWF gbl_adcon0
CLRF FCD_ADC3_S_00027_1_cnt
label268441756
MOVLW 0x28
SUBWF FCD_ADC3_S_00027_1_cnt, W
BTFSC STATUS,C
GOTO label268441757
INCF FCD_ADC3_S_00027_1_cnt, F
GOTO label268441756
label268441757
MOVLW 0x04
IORWF gbl_adcon0, W

```

```

MOVWF gbl_adcon0
label268441761
BTFSC gbl_adcon0,2
GOTO label268441761
MOVF FCD_ADC3_S_00027_1_old_tris, W
BSF STATUS, RP0
MOVWF gbl_trisa
MOVLW 0x07
MOVWF gbl_adcon1
MOVLW 0x80
BCF STATUS, RP0
MOVWF gbl_adcon0
RETURN
;} FCD_ADC3_SampleADC function end

ORG 0x00000277
FCD_ADC3_R_0002B
; { FCD_ADC3_ReadAsInt ; function begin
BCF STATUS, RP0
BCF STATUS, RP1
MOVF gbl_adresh, W
MOVWF FCD_ADC3_R_0002B_1_iRetVal
CLRF FCD_ADC3_R_0002B_1_iRetVal+D'1'
BCF STATUS, C
RLF FCD_ADC3_R_0002B_1_iRetVal, F
RLF FCD_ADC3_R_0002B_1_iRetVal+D'1', F
BCF STATUS, C
RLF FCD_ADC3_R_0002B_1_iRetVal, F
RLF FCD_ADC3_R_0002B_1_iRetVal+D'1', F
BSF STATUS, RP0
MOVF gbl_adresl, W
BCF STATUS, RP0
MOVWF CompTempVar658
CLRF CompTempVar659
RRF CompTempVar658, F
RRF CompTempVar658, F
RRF CompTempVar658, F
RRF CompTempVar658, F
RRF CompTempVar658, F
RRF CompTempVar658, F
ANDLW 0x03
ADDWF FCD_ADC3_R_0002B_1_iRetVal, F
MOVF CompTempVar659, W
BTFSC STATUS, C
INCF FCD_ADC3_R_0002B_1_iRetVal+D'1', F
ADDWF FCD_ADC3_R_0002B_1_iRetVal+D'1', F
MOVF FCD_ADC3_R_0002B_1_iRetVal, W
MOVWF CompTempVarRet655
MOVF FCD_ADC3_R_0002B_1_iRetVal+D'1', W
MOVWF CompTempVarRet655+D'1'
RETURN
;} FCD_ADC3_ReadAsInt function end

ORG 0x00000298
FCD_ADC2_S_00026
; { FCD_ADC2_SampleADC ; function begin
MOVLW 0x02
BSF STATUS, RP0
BCF STATUS, RP1
MOVWF gbl_adcon1
MOVF gbl_trisa, W
BCF STATUS, RP0
MOVWF FCD_ADC2_S_00026_1_old_tris
MOVLW 0x04

```

```

BSF STATUS, RP0
IORWF gbl_trisa, W
MOVWF gbl_trisa
MOVLW 0x91
BCF STATUS, RP0
MOVWF gbl_adcon0
CLRF FCD_ADC2_S_00026_1_cnt
label268441723
MOVLW 0x28
SUBWF FCD_ADC2_S_00026_1_cnt, W
BTFSC STATUS, C
GOTO label268441724
INCF FCD_ADC2_S_00026_1_cnt, F
GOTO label268441723
label268441724
MOVLW 0x04
IORWF gbl_adcon0, W
MOVWF gbl_adcon0
label268441728
BTFSC gbl_adcon0, 2
GOTO label268441728
MOVF FCD_ADC2_S_00026_1_old_tris, W
BSF STATUS, RP0
MOVWF gbl_trisa
MOVLW 0x07
MOVWF gbl_adcon1
MOVLW 0x80
BCF STATUS, RP0
MOVWF gbl_adcon0
RETURN
; } FCD_ADC2_SampleADC function end

ORG 0x000002BB
FCD_ADC2_R_0002A
; { FCD_ADC2_ReadAsInt ; function begin
BCF STATUS, RP0
BCF STATUS, RP1
MOVF gbl_adresh, W
MOVWF FCD_ADC2_R_0002A_1_iRetVal
CLRF FCD_ADC2_R_0002A_1_iRetVal+D'1'
BCF STATUS, C
RLF FCD_ADC2_R_0002A_1_iRetVal, F
RLF FCD_ADC2_R_0002A_1_iRetVal+D'1', F
BCF STATUS, C
RLF FCD_ADC2_R_0002A_1_iRetVal, F
RLF FCD_ADC2_R_0002A_1_iRetVal+D'1', F
BSF STATUS, RP0
MOVF gbl_adresl, W
BCF STATUS, RP0
MOVWF CompTempVar653
CLRF CompTempVar654
RRF CompTempVar653, F
RRF CompTempVar653, F
RRF CompTempVar653, F
RRF CompTempVar653, F
RRF CompTempVar653, F
RRF CompTempVar653, W
ANDLW 0x03
ADDWF FCD_ADC2_R_0002A_1_iRetVal, F
MOVF CompTempVar654, W
BTFSC STATUS, C
INCF FCD_ADC2_R_0002A_1_iRetVal+D'1', F
ADDWF FCD_ADC2_R_0002A_1_iRetVal+D'1', F
MOVF FCD_ADC2_R_0002A_1_iRetVal, W

```

```

MOVWF CompTempVarRet650
MOV FCD_ADC2_R_0002A_1_iRetVal+D'1', W
MOVWF CompTempVarRet650+D'1'
RETURN
; } FCD_ADC2_ReadAsInt function end

```

```

ORG 0x000002DC
FCD_ADC1_S_00025
; { FCD_ADC1_SampleADC ; function begin
MOVLW 0x04
BSF STATUS, RP0
BCF STATUS, RP1
MOVWF gbl_adcon1
MOV FCD_ADC1_S_00025_1_old_tris, W
BCF STATUS, RP0
MOVWF FCD_ADC1_S_00025_1_old_tris
MOVLW 0x02
BSF STATUS, RP0
IORWF gbl_trisa, W
MOVWF gbl_trisa
MOVLW 0x89
BCF STATUS, RP0
MOVWF gbl_adcon0
CLRF FCD_ADC1_S_00025_1_cnt
label268441690
MOVLW 0x28
SUBWF FCD_ADC1_S_00025_1_cnt, W
BTFSC STATUS, C
GOTO label268441691
INCF FCD_ADC1_S_00025_1_cnt, F
GOTO label268441690
label268441691
MOVLW 0x04
IORWF gbl_adcon0, W
MOVWF gbl_adcon0
label268441695
BTFSC gbl_adcon0, 2
GOTO label268441695
MOV FCD_ADC1_S_00025_1_old_tris, W
BSF STATUS, RP0
MOVWF gbl_trisa
MOVLW 0x07
MOVWF gbl_adcon1
MOVLW 0x80
BCF STATUS, RP0
MOVWF gbl_adcon0
RETURN
; } FCD_ADC1_SampleADC function end

```

```

ORG 0x000002FF
FCD_ADC1_R_00029
; { FCD_ADC1_ReadAsInt ; function begin
BCF STATUS, RP0
BCF STATUS, RP1
MOV FCD_ADC1_R_00029_1_iRetVal, W
MOVWF FCD_ADC1_R_00029_1_iRetVal
CLRF FCD_ADC1_R_00029_1_iRetVal+D'1'
BCF STATUS, C
RLF FCD_ADC1_R_00029_1_iRetVal, F
RLF FCD_ADC1_R_00029_1_iRetVal+D'1', F
BCF STATUS, C
RLF FCD_ADC1_R_00029_1_iRetVal, F
RLF FCD_ADC1_R_00029_1_iRetVal+D'1', F
BSF STATUS, RP0

```

```

MOVWF gbl_adresl, W
BCF STATUS, RP0
MOVWF CompTempVar648
CLRF CompTempVar649
RRF CompTempVar648, F
RRF CompTempVar648, F
RRF CompTempVar648, F
RRF CompTempVar648, F
RRF CompTempVar648, F
RRF CompTempVar648, F
ANDLW 0x03
ADDWF FCD_ADC1_R_00029_1_iRetVal, F
MOVWF CompTempVar649, W
BTFSC STATUS, C
INCF FCD_ADC1_R_00029_1_iRetVal+D'1', F
ADDWF FCD_ADC1_R_00029_1_iRetVal+D'1', F
MOVWF FCD_ADC1_R_00029_1_iRetVal, W
MOVWF CompTempVarRet645
MOVWF FCD_ADC1_R_00029_1_iRetVal+D'1', W
MOVWF CompTempVarRet645+D'1'
RETURN
; } FCD_ADC1_ReadAsInt function end

ORG 0x00000320
FCD_ADC0_S_00024
; { FCD_ADC0_SampleADC ; function begin
MOVLW 0x04
BSF STATUS, RP0
BCF STATUS, RP1
MOVWF gbl_adcon1
MOVWF gbl_trisa, W
BCF STATUS, RP0
MOVWF FCD_ADC0_S_00024_1_old_tris
MOVLW 0x01
BSF STATUS, RP0
IORWF gbl_trisa, W
MOVWF gbl_trisa
MOVLW 0x81
BCF STATUS, RP0
MOVWF gbl_adcon0
CLRF FCD_ADC0_S_00024_1_cnt
label268441657
MOVLW 0x28
SUBWF FCD_ADC0_S_00024_1_cnt, W
BTFSC STATUS, C
GOTO label268441658
INCF FCD_ADC0_S_00024_1_cnt, F
GOTO label268441657
label268441658
MOVLW 0x04
IORWF gbl_adcon0, W
MOVWF gbl_adcon0
label268441662
BTFSC gbl_adcon0, 2
GOTO label268441662
MOVWF FCD_ADC0_S_00024_1_old_tris, W
BSF STATUS, RP0
MOVWF gbl_trisa
MOVLW 0x07
MOVWF gbl_adcon1
MOVLW 0x80
BCF STATUS, RP0
MOVWF gbl_adcon0
RETURN

```



```

; } FCD_ADC0_SampleADC function end

    ORG 0x00000343
FCD_ADC0_R_00028
; { FCD_ADC0_ReadAsInt ; function begin
    BCF STATUS, RP0
    BCF STATUS, RP1
    MOVF gbl_adresh, W
    MOVWF FCD_ADC0_R_00028_1_iRetVal
    CLRF FCD_ADC0_R_00028_1_iRetVal+D'1'
    BCF STATUS,C
    RLF FCD_ADC0_R_00028_1_iRetVal, F
    RLF FCD_ADC0_R_00028_1_iRetVal+D'1', F
    BCF STATUS,C
    RLF FCD_ADC0_R_00028_1_iRetVal, F
    RLF FCD_ADC0_R_00028_1_iRetVal+D'1', F
    BSF STATUS, RP0
    MOVF gbl_adresl, W
    BCF STATUS, RP0
    MOVWF CompTempVar643
    CLRF CompTempVar644
    RRF CompTempVar643, F
    RRF CompTempVar643, F
    RRF CompTempVar643, F
    RRF CompTempVar643, F
    RRF CompTempVar643, F
    RRF CompTempVar643, W
    ANDLW 0x03
    ADDWF FCD_ADC0_R_00028_1_iRetVal, F
    MOVF CompTempVar644, W
    BTFSC STATUS,C
    INCF FCD_ADC0_R_00028_1_iRetVal+D'1', F
    ADDWF FCD_ADC0_R_00028_1_iRetVal+D'1', F
    MOVF FCD_ADC0_R_00028_1_iRetVal, W
    MOVWF CompTempVarRet640
    MOVF FCD_ADC0_R_00028_1_iRetVal+D'1', W
    MOVWF CompTempVarRet640+D'1'
    RETURN
; } FCD_ADC0_ReadAsInt function end

    ORG 0x00000364
FCM_STEP2_00031
; { FCM_STEP2_NS_GERI ; function begin
    BCF STATUS, RP0
    BCF STATUS, RP1
    MOVF gbl_FCV_N_S_LIMIT, F
    BTFSS STATUS,Z
    GOTO label268441113
    CLRF gbl_FCV_STEP2_NS
    MOVLW 0x0F
    BSF STATUS, RP0
    ANDWF gbl_trisd, W
    MOVWF gbl_trisd
    MOVLW 0x0F
    BCF STATUS, RP0
    ANDWF gbl_portd, W
    MOVWF CompTempVar709
    MOVLW 0xF0
    ANDWF gbl_FCV_STEP2_NS, W
    IORWF CompTempVar709, W
    MOVWF gbl_portd
    RETURN
label268441113
    DECF gbl_FCV_NS_GERI, W

```

```
BTFSS STATUS,Z
GOTO label268441122
MOVLW 0x10
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar710
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar710, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x90
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar711
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar711, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x80
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar712
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar712, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0xA0
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar713
MOVLW 0xF0
```

```

ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar713, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x20
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar714
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar714, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x60
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar715
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar715, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x40
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar716
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar716, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x50
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F

```

```

BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar717
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar717, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
RETURN
label268441122
CLRF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar718
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar718, W
MOVWF gbl_portd
RETURN
; } FCM_STEP2_NS_GERI function end

ORG 0x00000411
FCM_STEP2__00030
; { FCM_STEP2_NS_ILERI ; function begin
BCF STATUS, RP0
BCF STATUS, RP1
MOVF gbl_FCV_N_S_LIMIT, F
BTFSS STATUS, Z
GOTO label268441029
CLRF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar699
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar699, W
MOVWF gbl_portd
RETURN
label268441029
DECF gbl_FCV_NS_ILERI, W
BTFSS STATUS, Z
GOTO label268441038
MOVLW 0x50
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar700

```

```

MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar700, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x40
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar701
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar701, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x60
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar702
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar702, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x20
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar703
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar703, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0xA0
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd

```

```
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar704
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar704, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x80
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar705
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar705, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x90
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar706
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar706, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x10
MOVWF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar707
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar707, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
RETURN
label268441038
```

```

CLRF gbl_FCV_STEP2_NS
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar708
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar708, W
MOVWF gbl_portd
RETURN
; } FCM_STEP2_NS_ILERI function end

ORG 0x000004BE
FCM_STEP1_0002F
; { FCM_STEP1_EW_GERI ; function begin
BCF STATUS, RP0
BCF STATUS, RP1
MOVF gbl_FCV_E_W_LIMIT, F
BTFSS STATUS, Z
GOTO label268440953
CLRF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar690
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar690, W
MOVWF gbl_portd
RETURN
label268440953
DECF gbl_FCV_EW_GERI, W
BTFSS STATUS, Z
GOTO label268440962
MOVLW 0x01
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar691
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar691, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x09
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W

```

```

MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar692
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar692, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x08
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar693
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar693, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x0A
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar694
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar694, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x02
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar695
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar695, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x06

```



```

MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar696
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar696, W
MOVWF gbl_portd
MOVWF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x05
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar697
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar697, W
MOVWF gbl_portd
MOVWF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
RETURN
label268440962
CLRF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar698
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar698, W
MOVWF gbl_portd
RETURN
;} FCM_STEP1_EW_GERI function end

ORG 0x0000055A
FCM_STEP1_0002E
; { FCM_STEP1_EW_ILERI ; function begin
BSF STATUS, RP0
BCF STATUS, RP1
MOVWF gbl_FCV_E_W_LIMIT, F
BTFSS STATUS, Z
GOTO label268440869
CLRF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd

```

```

MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar680
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar680, W
MOVWF gbl_portd
RETURN
label268440869
DECf gbl_FCV_EW_ILERI, W
BTFSS STATUS, Z
GOTO label268440878
MOVLW 0x05
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar681
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar681, W
MOVWF gbl_portd
MOVf gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x04
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar682
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar682, W
MOVWF gbl_portd
MOVf gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x06
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar683
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar683, W
MOVWF gbl_portd
MOVf gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000

```

```
MOVLW 0x02
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar684
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar684, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x0A
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar685
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar685, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x08
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar686
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar686, W
MOVWF gbl_portd
MOVF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x09
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar687
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar687, W
```

```

MOVWF gbl_portd
MOVWF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
MOVLW 0x01
MOVWF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar688
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar688, W
MOVWF gbl_portd
MOVWF gbl_FCV_PULSE, W
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
RETURN
label268440878
CLRF gbl_FCV_STEP1_EW
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar689
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar689, W
MOVWF gbl_portd
RETURN
; } FCM_STEP1_EW_ILERI function end

ORG 0x00000607
FCM_STARTU_0001D
; { FCM_STARTUP ; function begin
CALL FCD_LCDDis_0001E
MOVLW 0x53
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4C
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x41
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x52
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x43
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x45
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F

```

```
MOVLW 0x4C
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4C
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x20
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x43
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4E
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x20
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x47
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x55
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4E
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x45
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x53
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
CLRF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5A
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4C
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x45
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x45
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x20
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
```

```

MOVLW 0x53
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x53
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x54
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x45
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x20
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x56
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x30
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x2E
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x30
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x05
MOVWF delay_ms_00000_arg_del
CALL delay_ms_00000
CALL FCD_LCDDis_00021
RETURN
; } FCM_STARTUP function end

ORG 0x00000686
FCM_MENU_00000
; { FCM_MENU ; function begin
BCF STATUS, RP0
BCF STATUS, RP1
MOVF gbl_FCV_MENU, W
XORLW 0x02
BTFSC STATUS,Z
CLRF gbl_FCV_MENU
DECF gbl_FCV_ENTER, W
BTFSS STATUS,Z
GOTO label268440435
MOVF gbl_FCV_TEMP1, F
BTFSS STATUS,Z
RETURN
MOVLW 0x01
MOVWF gbl_FCV_TEMP1
INCF gbl_FCV_MENU, W
MOVWF gbl_FCV_MENU
RETURN
label268440435

```



```

CALL FCD_LCDDis_0001F
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_DELTA_EW, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_DELTA_EW+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
MOVLW 0x07
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x47
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x3D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x09
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_LDR_SOUTH, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_LDR_SOUTH+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
MOVLW 0x07
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4B
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x3D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x09
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_LDR_NORTH, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_LDR_NORTH+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
MOVLW 0x07
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x2D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character

```



```

CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x09
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_DELTA_NS, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_DELTA_NS+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x44
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x45
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4C
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x54
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x41
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x58
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x10
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_DELTA_X, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_DELTA_X+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
MOVLW 0x06
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x06
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020

```

```

MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x06
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x06
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0D
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0D
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0D
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0D
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
DECF gbl_FCV_RIGHT, W
BTFSS STATUS, Z
GOTO label268440292
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4B
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x41
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F

```

```

MOVLW 0x59
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x54
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4B
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
label268440292
MOVF gbl_FCV_E_W_LIMIT, F
BTFS STATUS,Z
GOTO label268440335
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x44
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x2D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x42
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4C
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x54
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
label268440335
MOVF gbl_FCV_N_S_LIMIT, F
BTFS STATUS,Z
RETURN
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x

```

```

MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4B
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x2D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x47
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4C
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x54
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
RETURN
; } FCM_LDR_RW function end

ORG 0x000007F1
FCM_LCD_MA_0002D
; { FCM_LCD_MAN_ADC_RW ; function begin
CALL FCD_LCDDis_00021
MOVLW 0x44
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x3D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_LDR_EAST, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_LDR_EAST+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
CLRF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x42
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x3D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F

```

```

MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_LDR_WEST, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_LDR_WEST+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
CLRF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x2D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_DELTA_EW, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_DELTA_EW+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
MOVLW 0x07
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x47
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x3D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x09
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_LDR_SOUTH, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_LDR_SOUTH+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
MOVLW 0x07
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x01

```

```

MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4B
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x3D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x09
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_LDR_NORTH, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_LDR_NORTH+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
MOVLW 0x07
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x2D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x5F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x09
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVF gbl_FCV_DELTA_NS, W
MOVWF FCD_LCDDis_0002C_arg_Number
MOVF gbl_FCV_DELTA_NS+D'1', W
MOVWF FCD_LCDDis_0002C_arg_Number+D'1'
CALL FCD_LCDDis_0002C
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x41
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4E
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F

```

```

MOVLW 0x55
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x45
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4C
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x06
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x06
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x06
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x06
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0D
MOVWF FCD_LCDDis_00020_arg_x
CLRF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0D
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x01
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0D
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0D

```

```

MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x80
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
DECf gbl_FCV_RIGHT, W
BSF PCLATH,3
BTFSS STATUS,Z
GOTO label268440730
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
BCF PCLATH,3
CALL FCD_LCDDis_00020
MOVLW 0x4B
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x41
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x59
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x54
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4F
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4B
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
label268440730
MOVf gbl_FCV_E_W_LIMIT, F
BSF PCLATH,3
BTFSS STATUS,Z
GOTO label268440773
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
BCF PCLATH,3
CALL FCD_LCDDis_00020
MOVLW 0x44
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x2D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x42
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0E

```



```

MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4C
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x54
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
label268440773
MOVF gbl_FCV_N_S_LIMIT, F
BTFS STATUS, Z
RETURN
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x02
MOVWF FCD_LCDDis_00020_arg_y
BCF PCLATH, 3
CALL FCD_LCDDis_00020
MOVLW 0x4B
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x2D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x47
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x0E
MOVWF FCD_LCDDis_00020_arg_x
MOVLW 0x03
MOVWF FCD_LCDDis_00020_arg_y
CALL FCD_LCDDis_00020
MOVLW 0x4C
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x4D
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x49
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
MOVLW 0x54
MOVWF FCD_LCDDis_0001F_arg_Character
CALL FCD_LCDDis_0001F
RETURN
; } FCM_LCD_MAN_ADC_RW function end

ORG 0x00000944
FCM_INPUT__00023
; { FCM_INPUT_OUTPUT ; function begin

```

```
MOVLW 0x01
BSF STATUS, RP0
BCF STATUS, RP1
IORWF gbl_trisb, W
MOVWF gbl_trisb
MOVLW 0x01
BCF STATUS, RP0
ANDWF gbl_portb, W
MOVWF CompTempVar623
CLRF gbl_FCV_ENTER
DECF CompTempVar623, W
BTFSC STATUS, Z
INCF gbl_FCV_ENTER, F
MOVLW 0x02
BSF STATUS, RP0
IORWF gbl_trisb, W
MOVWF gbl_trisb
MOVLW 0x02
BCF STATUS, RP0
ANDWF gbl_portb, W
MOVWF CompTempVar625
CLRF gbl_FCV_UP
MOVLW 0x02
SUBWF CompTempVar625, W
BTFSC STATUS, Z
INCF gbl_FCV_UP, F
MOVLW 0x04
BSF STATUS, RP0
IORWF gbl_trisb, W
MOVWF gbl_trisb
MOVLW 0x04
BCF STATUS, RP0
ANDWF gbl_portb, W
MOVWF CompTempVar627
CLRF gbl_FCV_DOWN
MOVLW 0x04
SUBWF CompTempVar627, W
BTFSC STATUS, Z
INCF gbl_FCV_DOWN, F
MOVLW 0x08
BSF STATUS, RP0
IORWF gbl_trisb, W
MOVWF gbl_trisb
MOVLW 0x08
BCF STATUS, RP0
ANDWF gbl_portb, W
MOVWF CompTempVar629
CLRF gbl_FCV_LEFT
MOVLW 0x08
SUBWF CompTempVar629, W
BTFSC STATUS, Z
INCF gbl_FCV_LEFT, F
MOVLW 0x10
BSF STATUS, RP0
IORWF gbl_trisb, W
MOVWF gbl_trisb
MOVLW 0x10
BCF STATUS, RP0
ANDWF gbl_portb, W
MOVWF CompTempVar631
CLRF gbl_FCV_RIGHT
MOVLW 0x10
SUBWF CompTempVar631, W
BTFSC STATUS, Z
```

```

INCF gbl_FCV_RIGHT, F
MOVLW 0x20
BSF STATUS, RP0
IORWF gbl_trisb, W
MOVWF gbl_trisb
MOVLW 0x20
BCF STATUS, RP0
ANDWF gbl_portb, W
MOVWF CompTempVar633
CLRF gbl_FCV_E_W_LIMIT
MOVLW 0x20
SUBWF CompTempVar633, W
BTFSC STATUS, Z
INCF gbl_FCV_E_W_LIMIT, F
MOVLW 0x40
BSF STATUS, RP0
IORWF gbl_trisb, W
MOVWF gbl_trisb
MOVLW 0x40
BCF STATUS, RP0
ANDWF gbl_portb, W
MOVWF CompTempVar635
CLRF gbl_FCV_N_S_LIMIT
MOVLW 0x40
SUBWF CompTempVar635, W
BTFSC STATUS, Z
INCF gbl_FCV_N_S_LIMIT, F
MOVLW 0xBF
BSF STATUS, RP0
ANDWF gbl_trisc, W
MOVWF gbl_trisc
BCF STATUS, RP0
MOVF gbl_FCV_LCD_BACKLIGHT, F
BTFSC STATUS, Z
GOTO label268439957
MOVLW 0xBF
ANDWF gbl_portc, W
MOVWF CompTempVar637
MOVLW 0x40
IORWF CompTempVar637, W
MOVWF gbl_portc
GOTO label268439961
label268439957
MOVLW 0xBF
ANDWF gbl_portc, W
MOVWF gbl_portc
label268439961
MOVLW 0xF0
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0xF0
BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar638
MOVLW 0x0F
ANDWF gbl_FCV_STEP1_EW, W
IORWF CompTempVar638, W
MOVWF gbl_portd
MOVLW 0x0F
BSF STATUS, RP0
ANDWF gbl_trisd, W
MOVWF gbl_trisd
MOVLW 0x0F

```

```

BCF STATUS, RP0
ANDWF gbl_portd, W
MOVWF CompTempVar639
MOVLW 0xF0
ANDWF gbl_FCV_STEP2_NS, W
IORWF CompTempVar639, W
MOVWF gbl_portd
BCF PCLATH,3
CALL FCD_ADC0_S_00024
CALL FCD_ADC1_S_00025
CALL FCD_ADC2_S_00026
CALL FCD_ADC3_S_00027
CALL FCD_ADC0_R_00028
MOVF CompTempVarRet640, W
MOVWF gbl_FCV_LDR_EAST
MOVF CompTempVarRet640+D'1', W
MOVWF gbl_FCV_LDR_EAST+D'1'
CALL FCD_ADC1_R_00029
MOVF CompTempVarRet645, W
MOVWF gbl_FCV_LDR_WEST
MOVF CompTempVarRet645+D'1', W
MOVWF gbl_FCV_LDR_WEST+D'1'
CALL FCD_ADC2_R_0002A
MOVF CompTempVarRet650, W
MOVWF gbl_FCV_LDR_NORTH
MOVF CompTempVarRet650+D'1', W
MOVWF gbl_FCV_LDR_NORTH+D'1'
CALL FCD_ADC3_R_0002B
MOVF CompTempVarRet655, W
MOVWF gbl_FCV_LDR_SOUTH
MOVF CompTempVarRet655+D'1', W
MOVWF gbl_FCV_LDR_SOUTH+D'1'
RETURN
; } FCM_INPUT_OUTPUT function end

ORG 0x000009E3
FCM_HESAP_00000
; { FCM_HESAP ; function begin
BCF STATUS, RP0
BCF STATUS, RP1
MOVF gbl_FCV_DELTA_X, W
SUBLW 0x00
MOVWF gbl_FCV_DELTA_EKSI
COMF gbl_FCV_DELTA_X+D'1', W
MOVWF gbl_FCV_DELTA_EKSI+D'1'
BTFSC STATUS,C
INCF gbl_FCV_DELTA_EKSI+D'1', F
MOVF gbl_FCV_LDR_WEST, W
SUBWF gbl_FCV_LDR_EAST, W
MOVWF gbl_FCV_DELTA_EW
INCF gbl_FCV_LDR_WEST+D'1', W
BTFSC STATUS,C
MOVF gbl_FCV_LDR_WEST+D'1', W
SUBWF gbl_FCV_LDR_EAST+D'1', W
MOVWF gbl_FCV_DELTA_EW+D'1'
MOVF gbl_FCV_LDR_NORTH, W
SUBWF gbl_FCV_LDR_SOUTH, W
MOVWF gbl_FCV_DELTA_NS
INCF gbl_FCV_LDR_NORTH+D'1', W
BTFSC STATUS,C
MOVF gbl_FCV_LDR_NORTH+D'1', W
SUBWF gbl_FCV_LDR_SOUTH+D'1', W
MOVWF gbl_FCV_DELTA_NS+D'1'
BTFSS gbl_FCV_DELTA_EW+D'1',7

```

```

GOTO label268441209
MOVF gbl_FCV_DELTA_EW+D'1', W
XORLW 0x80
MOVWF CompTempVar725
MOVF gbl_FCV_DELTA_EKSI+D'1', W
XORLW 0x80
SUBWF CompTempVar725, W
BTFSS STATUS,Z
GOTO label268441214
MOVF gbl_FCV_DELTA_EKSI, W
SUBWF gbl_FCV_DELTA_EW, W
label268441214
BTFSC STATUS,C
GOTO label268441215
MOVLW 0x01
MOVWF gbl_FCV_EW_GERI
GOTO label268441222
label268441215
CLRF gbl_FCV_EW_GERI
GOTO label268441222
label268441209
CLRF gbl_FCV_EW_GERI
label268441222
MOVF gbl_FCV_DELTA_EW, W
SUBLW 0x00
MOVF gbl_FCV_DELTA_EW+D'1', W
BTFSC STATUS,C
BTFSS STATUS,Z
BTFSC gbl_FCV_DELTA_EW+D'1',7
GOTO label268441228
MOVF gbl_FCV_DELTA_X+D'1', W
XORLW 0x80
MOVWF CompTempVar726
MOVF gbl_FCV_DELTA_EW+D'1', W
XORLW 0x80
SUBWF CompTempVar726, W
BTFSS STATUS,Z
GOTO label268441233
MOVF gbl_FCV_DELTA_EW, W
SUBWF gbl_FCV_DELTA_X, W
label268441233
BTFSC STATUS,C
GOTO label268441234
MOVLW 0x01
MOVWF gbl_FCV_EW_ILERI
GOTO label268441241
label268441234
CLRF gbl_FCV_EW_ILERI
GOTO label268441241
label268441228
CLRF gbl_FCV_EW_ILERI
label268441241
BTFSS gbl_FCV_DELTA_NS+D'1',7
GOTO label268441244
MOVF gbl_FCV_DELTA_NS+D'1', W
XORLW 0x80
MOVWF CompTempVar727
MOVF gbl_FCV_DELTA_EKSI+D'1', W
XORLW 0x80
SUBWF CompTempVar727, W
BTFSS STATUS,Z
GOTO label268441249
MOVF gbl_FCV_DELTA_EKSI, W
SUBWF gbl_FCV_DELTA_NS, W

```

```

label268441249
    BTFSC STATUS,C
    GOTO label268441250
    MOVLW 0x01
    MOVWF gbl_FCV_NS_ILERI
    GOTO label268441257
label268441250
    CLRF gbl_FCV_NS_ILERI
    GOTO label268441257
label268441244
    CLRF gbl_FCV_NS_ILERI
label268441257
    MOVF gbl_FCV_DELTA_NS, W
    SUBLW 0x00
    MOVF gbl_FCV_DELTA_NS+D'1', W
    BTFSC STATUS,C
    BTFSS STATUS,Z
    BTFSC gbl_FCV_DELTA_NS+D'1',7
    GOTO label268441263
    MOVF gbl_FCV_DELTA_X+D'1', W
    XORLW 0x80
    MOVWF CompTempVar728
    MOVF gbl_FCV_DELTA_NS+D'1', W
    XORLW 0x80
    SUBWF CompTempVar728, W
    BTFSS STATUS,Z
    GOTO label268441268
    MOVF gbl_FCV_DELTA_NS, W
    SUBWF gbl_FCV_DELTA_X, W
label268441268
    BTFSC STATUS,C
    GOTO label268441269
    MOVLW 0x01
    MOVWF gbl_FCV_NS_GERI
    RETURN
label268441269
    CLRF gbl_FCV_NS_GERI
    RETURN
label268441263
    CLRF gbl_FCV_NS_GERI
    RETURN
; } FCM_HESAP function end

    ORG 0x00000A57
FCM_DELTAX_00032
; { FCM_DELTAX_YAZ ; function begin
    MOVLW 0x64
    BCF STATUS, RP0
    BCF STATUS, RP1
    XORWF gbl_FCV_DELTA_X, W
    BTFSC STATUS,Z
    MOVF gbl_FCV_DELTA_X+D'1', W
    BTFSC STATUS,Z
    GOTO label268441297
    DECF gbl_FCV_UP, W
    BTFSS STATUS,Z
    GOTO label268441286
    MOVF gbl_FCV_TEMP2, F
    BTFSS STATUS,Z
    GOTO label268441297
    INCF gbl_FCV_DELTA_X, F
    BTFSC STATUS,Z
    INCF gbl_FCV_DELTA_X+D'1', F
    MOVLW 0x01

```

```

MOVWF gbl_FCV_TEMP2
GOTO label268441297
label268441286
CLRF gbl_FCV_TEMP2
label268441297
MOVLW 0x01
XORWF gbl_FCV_DELTA_X, W
BTFSC STATUS,Z
MOVF gbl_FCV_DELTA_X+D'1', W
BTFSC STATUS,Z
GOTO label268441316
DECF gbl_FCV_DOWN, W
BTFSS STATUS,Z
GOTO label268441305
MOVF gbl_FCV_TEMP3, F
BTFSS STATUS,Z
GOTO label268441316
MOVLW 0x01
SUBWF gbl_FCV_DELTA_X, W
MOVWF CompTempVar731
BTFSS STATUS,C
DECF gbl_FCV_DELTA_X+D'1', F
MOVF CompTempVar731, W
MOVWF gbl_FCV_DELTA_X
MOVLW 0x01
MOVWF gbl_FCV_TEMP3
GOTO label268441316
label268441305
CLRF gbl_FCV_TEMP3
label268441316
DECF gbl_FCV_RIGHT, W
BTFSS STATUS,Z
RETURN
CLRF FCD_EEPROM_00033_arg_addr
CLRF FCD_EEPROM_00033_arg_addr+D'1'
MOVF gbl_FCV_DELTA_X, W
MOVWF FCD_EEPROM_00033_arg_data
BCF PCLATH,3
CALL FCD_EEPROM_00033
RETURN
; } FCM_DELTAX_YAZ function end

ORG 0x00000A8D
FCM_BACKLI_00022
; { FCM_BACKLIGHT ; function begin
BCF STATUS, RP0
BCF STATUS, RP1
DECF gbl_FCV_ENTER, W
BTFSS STATUS,Z
GOTO label268439855
MOVLW 0x01
MOVWF gbl_FCV_LCD_BACKLIGHT
CLRF gbl_FCV_COUNTER
label268439855
DECF gbl_FCV_UP, W
BTFSS STATUS,Z
GOTO label268439860
MOVLW 0x01
MOVWF gbl_FCV_LCD_BACKLIGHT
CLRF gbl_FCV_COUNTER
label268439860
DECF gbl_FCV_DOWN, W
BTFSS STATUS,Z
GOTO label268439865

```

```

    MOVLW 0x01
    MOVWF gbl_FCV_LCD_BACKLIGHT
    CLRF gbl_FCV_COUNTER
label268439865
    DECF gbl_FCV_LEFT, W
    BTFSS STATUS,Z
    GOTO label268439870
    MOVLW 0x01
    MOVWF gbl_FCV_LCD_BACKLIGHT
    CLRF gbl_FCV_COUNTER
label268439870
    DECF gbl_FCV_RIGHT, W
    BTFSS STATUS,Z
    RETURN
    MOVLW 0x01
    MOVWF gbl_FCV_LCD_BACKLIGHT
    CLRF gbl_FCV_COUNTER
    RETURN
; } FCM_BACKLIGHT function end

    ORG 0x0000AAE
FCD_EEPROM_0003E
; { FCD_EEPROM0_EEPROMRead ; function begin
    CLRF FCD_EEPROM_0003E_1_data
    MOVF FCD_EEPROM_0003E_arg_addr, W
    BSF STATUS, RP1
    MOVWF gbl_eaddr
    BSF STATUS, RP0
    BCF gbl_eecon1,7
    BSF gbl_eecon1,0
    BCF STATUS, RP0
    MOVF gbl_eedata, W
    BCF STATUS, RP1
    MOVWF FCD_EEPROM_0003E_1_data
    MOVF FCD_EEPROM_0003E_1_data, W
    MOVWF CompTempVarRet741
    RETURN
; } FCD_EEPROM0_EEPROMRead function end

    ORG 0x0000ABC
main
; { main ; function begin
    MOVLW 0x07
    BSF STATUS, RP0
    BCF STATUS, RP1
    MOVWF gbl_adcon1
    MOVLW 0xC0
    MOVWF gbl_option_reg
    BCF gbl_option_reg,5
    BSF gbl_option_reg,4
    MOVLW 0xF0
    ANDWF gbl_option_reg, W
    BCF STATUS, RP0
    MOVWF CompTempVar742
    MOVLW 0x05
    IORWF CompTempVar742, W
    BSF STATUS, RP0
    MOVWF gbl_option_reg
    BSF gbl_intcon,7
    BSF gbl_intcon,5
    MOVLW 0x64
    BCF STATUS, RP0
    MOVWF gbl_FCV_PULSE
    CLRF FCD_EEPROM_0003E_arg_addr

```



```

CLRF FCD_EEPROM_0003E_arg_addr+D'1'
CALL FCD_EEPROM_0003E
MOVWF CompTempVarRet741, W
MOVWF gbl_FCV_DELTA_X
CLRF gbl_FCV_DELTA_X+D'1'
MOVWF gbl_FCV_DELTA_X, W
SUBLW 0x64
MOVWF gbl_FCV_DELTA_X+D'1', W
BTFSC STATUS, C
BTFSS STATUS, Z
BTFSC gbl_FCV_DELTA_X+D'1', 7
GOTO label268441917
CLRF FCD_EEPROM_00033_arg_addr
CLRF FCD_EEPROM_00033_arg_addr+D'1'
MOVLW 0x64
MOVWF FCD_EEPROM_00033_arg_data
BCF PCLATH, 3
CALL FCD_EEPROM_00033
label268441917
BCF STATUS, RP0
BCF STATUS, RP1
MOVWF gbl_FCV_DELTA_X, F
BSF PCLATH, 3
BTFSS STATUS, Z
GOTO label268441926
MOVWF gbl_FCV_DELTA_X+D'1', F
BTFSS STATUS, Z
GOTO label268441926
CLRF FCD_EEPROM_00033_arg_addr
CLRF FCD_EEPROM_00033_arg_addr+D'1'
MOVLW 0x0A
MOVWF FCD_EEPROM_00033_arg_data
BCF PCLATH, 3
CALL FCD_EEPROM_00033
label268441926
BCF STATUS, RP0
BCF STATUS, RP1
CLRF FCD_EEPROM_0003E_arg_addr
CLRF FCD_EEPROM_0003E_arg_addr+D'1'
BSF PCLATH, 3
CALL FCD_EEPROM_0003E
MOVWF CompTempVarRet741, W
MOVWF gbl_FCV_DELTA_X
CLRF gbl_FCV_DELTA_X+D'1'
BCF PCLATH, 3
CALL FCM_STARTU_0001D
label268441941
BSF PCLATH, 3
CALL FCM_INPUT_00023
BSF PCLATH, 3
CALL FCM_BACKLI_00022
BCF PCLATH, 3
CALL FCM_MENU_00000
CALL FCM_STEP1_0002E
CALL FCM_STEP1_0002F
CALL FCM_STEP2_00030
CALL FCM_STEP2_00031
BSF PCLATH, 3
CALL FCM_HESAP_00000
MOVWF gbl_FCV_MENU, F
BTFSS STATUS, Z
GOTO label268441960
BCF PCLATH, 3
CALL FCM_LDR_RW_00000

```

```

BSF PCLATH,3
CALL FCM_DELTAX_00032
label268441960
BCF STATUS, RP0
BCF STATUS, RP1
DECf gbl_FCV_MENU, W
BSF PCLATH,3
BTFSS STATUS, Z
GOTO label268441941
BCF PCLATH,3
CALL FCM_LCD_MA_0002D
DECf gbl_FCV_LEFT, W
BSF PCLATH,3
BTFSS STATUS, Z
GOTO label268441972
MOVLW 0x01
MOVWF gbl_FCV_EW_GERI
GOTO label268441976
label268441972
CLRF gbl_FCV_EW_GERI
label268441976
DECf gbl_FCV_RIGHT, W
BTFSS STATUS, Z
GOTO label268441979
MOVLW 0x01
MOVWF gbl_FCV_EW_ILERI
GOTO label268441983
label268441979
CLRF gbl_FCV_EW_ILERI
label268441983
DECf gbl_FCV_UP, W
BTFSS STATUS, Z
GOTO label268441986
MOVLW 0x01
MOVWF gbl_FCV_NS_GERI
GOTO label268441990
label268441986
CLRF gbl_FCV_NS_GERI
label268441990
DECf gbl_FCV_DOWN, W
BTFSS STATUS, Z
GOTO label268441993
MOVLW 0x01
MOVWF gbl_FCV_NS_ILERI
GOTO label268441941
label268441993
CLRF gbl_FCV_NS_ILERI
GOTO label268441941
; } main function end

ORG 0x00000B37
_startup
BSF PCLATH,3
BCF PCLATH,4
GOTO main
ORG 0x00000B3A
interrupt
; { interrupt ; function begin
BTFSS gbl_intcon,2
GOTO label268442004
BCF PCLATH,3
CALL FCM_CLOCK_00000
BCF gbl_intcon,2
label268442004

```

```
BCF STATUS, RP0
BCF STATUS, RP1
SWAPF Int1BContext+D'2', W
MOVWF FSR
SWAPF Int1BContext+D'1', W
MOVWF PCLATH
SWAPF Int1BContext, W
MOVWF STATUS
SWAPF Int1Context, F
SWAPF Int1Context, W
RETFIE
;} interrupt function end

ORG 0x00002007
DW 0x3F3A
END
```

ÖZGEÇMİŞ 1

Ad Soyad: Uğur YILMAZ

Doğum Tarihi: 20-10-1986

Doğum Yeri: Kocaeli

Lise: Mehmet Akif Ersoy Anadolu Lisesi

Staj Yaptığı Yerler: TOYOTA, Adapazarı, (30 gün), FORD OTOSAN, Gölcük, (30 gün)

ÖZGEÇMİŞ 2

Ad Soyad: Borahan KÜÇÜK

Doğum Tarihi: 24-11-1987

Doğum Yeri: Eskişehir

Lise: Derince 19 Mayıs Lisesi

Staj Yaptığı Yerler: TÜPRAŞ, İzmit,(60 gün)