



**MİKRODENETLEYİCİ TEMELLİ PLC'LER
İÇİN PROGRAMLAMA YAZILIMININ
GERÇEKLEŞTİRİLMESİ**

Serhat KILIÇARSLAN

**Yüksek Lisans Tezi
Mekatronik Mühendisliği Anabilim Dalı
Yrd. Doç. Dr. Gökhan GELEN
2014**

Her hakkı saklıdır.

T.C.
GAZİOSMANPAŞA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

MİKRODENETLEYİCİ TEMELLİ PLC'LER İÇİN
PROGRAMLAMA YAZILIMININ GERÇEKLEŞTİRİLMESİ

Serhat KILIÇARSLAN

TOKAT

2014

Her hakkı saklıdır.

Yrd. Doç. Dr. Gökhan GELEN danışmanlığında, Serhat KILIÇARSLAN tarafından hazırlanan bu çalışma, 12.09.2014 tarihinde aşağıdaki jüri tarafından oy birliği ile Mekatronik Mühendisliği Anabilim Dalı'nda Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. Murat UZAM

İmza: 

Üye : Yrd. Doç. Dr. Gökhan GELEN

İmza: 

Üye : Yrd. Doç. Dr. Zafer DOĞAN

İmza: 

Yukarıdaki sonucu onaylarım


Prof. Dr. Mehmet Ali SAKIN
Enstitü Müdürü

26.09/2014

TEZ BEYANI

Tez yazım kurallarına uygun olarak hazırlanan bu tezin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezin içerdiği yenilik ve sonuçların başka bir yerden alınmadığını, kullanılan verilerde herhangi bir tahrifat yapılmadığını, tezin herhangi bir kısmının bu üniversite veya başka bir üniversitedeki başka bir tez çalışması olarak sunulmadığını beyan ederim.

Serhat KILIÇARSLAN

ÖZET

Yüksek Lisans Tezi

MİKRODENETLEYİCİ TEMELLİ PLC'LER İÇİN
PROGRAMLAMA YAZILIMININ GERÇEKLEŞTİRİLMESİ
Serhat KILIÇARSLAN

Gaziosmanpaşa Üniversitesi
Fen Bilimleri Enstitüsü
Mekatronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Gökhan GELEN

PLC'ler endüstriyel otomasyon sistemlerinde yaygın olarak kullanılan kontrol cihazlarıdır. Çeşitli üreticilerin farklı modellerde birçok PLC'si mevcuttur. Son zamanlarda, ticari PLC'lere alternatif mikrodeneleyici temelli ucuz PLC'ler tasarlanmıştır. Bu PLC'ler diğerleri gibi komut tabanlı olarak programlanabilmektedir. Bu tezde, PIC mikrodeneleyicisi temelli PLC'ler için çizilen merdiven diyagramı programını PIC koduna dönüştüren programlama arayüzü geliştirilmiştir. Gerçekleştirilen arayüz web sayfası sitilinde olduğundan çeşitli platformlarda çalışabilir. Arayüz çeşitli PLC kod örnekleri kullanılarak test edilmiştir. Tasarlanan arayüzün gerçek sistemlere uygulanabilirliğinin gösterilmesi için deneysel bir dağıtım istasyonu dikkate alınmıştır. Arayüz kullanılarak dağıtım istasyonu için elde edilen kod sistemi başarılı biçimde kontrol etmektedir. Testler sonucunda arayüzün sorunsuz çalıştığı görülmüştür.

2014, 78 sayfa

Anahtar Kelimeler: PIC-PLC, Ladder Diyagramları, Assembly Kodu, Web Tabanlı Arayüz, Görsel C#

ABSTRACT

M.Sc. Thesis

REALIZATION OF A PROGRAMMING SOFTWARE FOR MICROCONTROLLER-BASED PLCS

Serhat KILIÇARSLAN

Gaziosmanpaşa University
Graduate School of Natural and Applied Sciences
Department of Mechatronic Engineering

Supervisor: Asst. Prof. Dr. Gökhan GELEN

PLCs are widely used control devices in industrial automation systems. There are different types of PLCs produced by a lot of manufacturers. Recently, microcontroller based inexpensive PLCs are designed as alternative to commercial PLCs. These PLCs are programmed by using instruction list as other PLCs. In this thesis, an interface is developed to convert Ladder diagram to PIC assembly codes for PIC based PLCs. The interface is realized as a webpage so it can be run on different platforms. The interface is tested by using different PLC code samples. An experimental distribution station is considered to demonstrate applicability of the designed interface for real systems. The code obtained by using the interface successfully controls the distribution station. According to test results, it is seen that the interface works without any problem.

2014, 78 pages

Keywords: PIC-PLC, Ladder Diagrams, Assembly Code, Web Based Interface, Visual C#

TEŐEKKÜRLER

Yüksek lisans çalışmam boyunca, her türlü konuda yardımlarını esirgemeyen, değerli fikirleriyle çalışmama yön veren kıymetli hocam Sayın Yrd. Doç. Dr. Gökhan GELEN'e, Yunus AYGOĐAN'a ve hayatım boyunca desteklerini esirgemeyen anneme, babama, kardeşlerime, eşime ve arkadaşlarıma şükranlarımı sunarım.

Serhat KILIÇARSLAN

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	ix
KISALTMALAR DİZİNİ	x
1. GİRİŞ	1
2. KAYNAK ÖZETLERİ-KURAMSAL TEMELLER	3
2.1. Kaynak Özetleri	3
2.2. Programlanabilir Lojik Denetleyiciler (PLC)	4
2.3. PLC'lerin Yapısı	6
2.4. PLC'lerin Programlanması	7
2.4.1. Merdiven (Ladder) Diyagramı Programlama	7
2.4.2. Deyim Listesi (STL, Statement List)	8
2.4.3. Fonksiyon Blok Diyagram (FBD, Function Block Diagram).....	8
2.5. Mikrodenetleyiciler.....	9
2.5.1. PIC Mikrodenetleyicileri	9
2.5.2. PIC16F877A Mikrodenetleyicisinin Genel Özellikleri	10
2.5.3. PIC16F648A Mikrodenetleyicisinin Genel Özellikleri	10
3. MATERYAL VE YÖNTEM	13
3.1. PIC Mikrodenetleyici Temelli PLC'ler	13
3.2. PIC16F877A Temelli PIC-PLC Donanımı.....	13
3.2.1. Girişler	15
3.2.2. Çıkışlar.....	15
3.3. PIC16F648A Temelli PIC-PLC Donanımı.....	16
3.4. PIC-PLC Yazılımı.....	17
3.5. PIC PLC Program Makroları	17
3.5.1. Kontak ve Röle Temelli Fonksiyonlar	17
3.6. Zamanlayıcı Fonksiyonları	19
3.6.1. Düz Zaman Rölesi (TON).....	19
3.6.2. Ters Zaman Rölesi (TOF).....	20
3.6.3. Puls Zaman Rölesi (TP).....	21
3.6.4. Osilatör Zaman Rölesi	22
3.7. Sayıcı Fonksiyonları	22
3.7.1. İleri Sayıcı (CTU)	22
3.7.2. Geri Sayıcı (CTD).....	23
3.7.3. İleri / Geri Sayıcı (CTUD).....	23
3.8. Karşılaştırma Fonksiyonları.....	24
3.9. Aritmetik Fonksiyonlar.....	25
3.10. Flip-Flop Makroları	26
3.11. Lojik Fonksiyonlar.....	27
3.12. Diğer Makrolar	28
4. BULGULAR VE TARTIŞMA	30
4.1. Mikrodenetleyici Temelli Plc'ler İçin Programlama Yazılımı	30
4.2. Geliştirilen Yazılımın Bölümleri	30

4.2.1.	Menüler Bölümü	32
4.2.2.	Çizim Bölümü	37
4.2.3.	Araçlar (Toolbox) Bölümü	38
4.2.4.	Özellikler Bölümü	41
4.2.5.	Arayüz Kullanarak Ladder Diyagram Çizimi	42
4.3.	Derleme Algoritması	44
4.4.	Tarayıcı ve Performans Testleri	46
4.5.	Uygulamalar	49
4.6.	Temel Ladder Diyagram Dönüşümleri	49
4.6.1.	Kontak ve Röle Temelli Ladder Diyagramları	49
4.6.2.	Zamanlayıcı Uygulaması	53
4.6.3.	Sayıcı Uygulaması	56
4.6.4.	Flip-Flop Uygulaması	59
4.6.5.	Karşılaştırma Fonksiyonları Uygulaması	63
4.6.6.	Aritmetik Fonksiyonların Uygulaması	66
4.7.	Gerçek Sistem Kontrolü Uygulaması	69
5.	SONUÇ	75
	KAYNAKLAR	77
	ÖZGEÇMİŞ	80

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2. 1. PLC'nin genel yapısı	6
Şekil 2. 2. PLC'nin çalışma mantığı.....	7
Şekil 2. 3. Örnek bir merdiven diyagram programı	8
Şekil 2. 4. Örnek bir STL programı	8
Şekil 2. 5. Örnek FBD diyagram	8
Şekil 2. 6. PIC16F877A'nın bacak bağlantıları	11
Şekil 2. 7. PIC16F648A bacak bağlantıları	12
Şekil 3. 1. PIC16F877A temelli PIC-PLC devre şeması	14
Şekil 3. 2. PIC16F877A temelli PIC-PLC'nin resmi.....	14
Şekil 3. 3. Giriş bağlantıları	15
Şekil 3. 4. Çıkış bağlantıları.....	15
Şekil 3. 5. PIC16F648A temelli PIC-PLC devre şeması	16
Şekil 3. 6. PIC16F648A temelli PIC-PLC'nin resmi.....	16
Şekil 3. 7. Temel yazılım dosyası [2]	17
Şekil 3. 8. Düz zaman rölesinin sembolü ve zaman diyagramı [2]	20
Şekil 3. 9. Ters zaman rölesinin sembolü ve zaman diyagramı [2].....	21
Şekil 3. 10. Puls zaman rölesinin sembolü ve zaman diyagramı [2]	21
Şekil 3. 11. İleri sayıcı (CTU) sembol görünümü ve sembol açıklamaları [2].....	22
Şekil 3. 12. Geri sayıcı (CTD) sembol görünümü ve sembol açıklamaları [2]	23
Şekil 3. 13. İleri/Geri sayıcı (CTUD) sembol görünümü ve sembol açıklamaları [2]....	24
Şekil 4. 1. Programın özet akış şeması	31
Şekil 4. 2. Yazılımın ana bileşenleri	31
Şekil 4. 3. Yazılımın ekran görüntüsü	32
Şekil 4. 4. Menü ikonları	32
Şekil 4. 5. Yeni proje açılmış ekran.....	33
Şekil 4. 6. Dosya seçme alanı	34
Şekil 4. 7. Derleme ekranı	35
Şekil 4. 8. Donanım seçme ekranı	36
Şekil 4. 9. Program açılış ekranı	36

Şekil 4. 10. Programın ingilizce menülü versiyonu.....	37
Şekil 4. 11. Çizim alanı.....	38
Şekil 4. 12. Başlangıç noktası.....	38
Şekil 4. 13. Tek değerli parametre atama ekranı	41
Şekil 4. 14. Çok değerli parametre atama ekranı	41
Şekil 4. 15. Zaman hesaplama ekranı	42
Şekil 4. 16. PIC-PLC Assembly makroları dosyası.....	43
Şekil 4. 17. Arayüz kullanarak PIC-PLC'nin programlanması	44
Şekil 4. 18. Seri bağlantı görüntüsü.....	45
Şekil 4. 19. Paralel bağlantı görüntüsü	45
Şekil 4. 20. Derleme algoritması akış şeması	46
Şekil 4. 21. Tüm tarayıcılardaki ekran görüntüsü.....	47
Şekil 4. 22. PIC-PLC için test programı	48
Şekil 4. 23. Test programının derlenme sonucu	48
Şekil 4. 24. Temel kontak ve röle örneğinin arayüz ile çizimi	51
Şekil 4. 25. Temel kontak ve röle örneği derleme çıktısı	52
Şekil 4. 26. Zamanlayıcı fonksiyonları örneği.....	53
Şekil 4. 27. Zamanlayıcı örneğinin arayüz ile çizimi	55
Şekil 4. 28. Zamanlayıcı örneği derleme çıktısı	56
Şekil 4. 29. Sayıcı fonksiyonları örneği.....	57
Şekil 4. 30. Sayıcı örneğinin arayüz ile çizimi	58
Şekil 4. 31. Sayıcı örneği derleme çıktısı	59
Şekil 4. 32. Flip-flop ladder örneği.....	60
Şekil 4. 33. Flip-flop örneğinin arayüz ile çizimi	62
Şekil 4. 34. Flip-Flop örneği derleme çıktısı	63
Şekil 4. 35. Karşılaştırma fonksiyonları örneği	64
Şekil 4. 36. Karşılaştırma örneğinin arayüz ile çizimi.....	65
Şekil 4. 37. Karşılaştırma örneği derleme çıktısı.....	66
Şekil 4. 38. Aritmetik fonksiyonları örneği	67
Şekil 4. 39. Aritmetik örneğinin arayüz ile çizimi.....	68
Şekil 4. 40. Aritmetik uygulama derleme çıktısı	69
Şekil 4. 41. Dağıtım istasyonu resmi [49]	70

Şekil 4. 42. Dağıtım istasyonuna ait otomat modeli	71
Şekil 4. 43. Dağıtım istasyonuna ait kodun ladder diyagramı	73
Şekil 4. 44. Dağıtım istasyona ait kodun asm çıktısı	74

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 2. 1. PIC16F877A'nın genel özellikleri.....	11
Çizelge 2. 2. PIC16F648A genel özellikleri	12
Çizelge 3. 1. Kontak ve Röle temelli makrolar[2].....	18
Çizelge 3. 2. Temel lojik işlem makroları [2].....	18
Çizelge 3. 3. Referans zamanlama frekansları [2]	19
Çizelge 3. 4. Karşılaştırma fonksiyonları [2].....	24
Çizelge 3. 5. Karşılaştırma fonksiyonlarına ait semboller [2]	25
Çizelge 3. 6. Aritmetik işlemler fonksiyonları [2].....	26
Çizelge 3. 7. Flip-Flop fonksiyonları [2]	27
Çizelge 3. 8. Lojik fonksiyonlar [2].....	28
Çizelge 3. 9. Diğer makrolar [2]	29
Çizelge 4. 1. Ladder araçları(a)	39
Çizelge 4. 2. Ladder araçları (b)	40
Çizelge 4. 3. Performans karşılaştırmaları	49
Çizelge 4. 4. Sensör ve PLC bağlantılar	70
Çizelge 4. 5. Otomat modelindeki durumların açıklamaları.....	71
Çizelge 4. 6. Otomat modelindeki olayların açıklamaları	71
Çizelge 4. 7. Otomat modelindeki durumların hafıza biti karşılığı ve olayların plc giriş karşılığı	71

KISALTMALAR DİZİNİ

Kısaltmalar

Açıklamalar

PLC	Programlanabilir Lojik Denetleyiciler
PIC	Peripheral Interface Controller
CPU	Merkezi İşlem Birimi
STL	Statement List – Komut Listesi
FBD	Fonksiyon Blok Diyagramı
PT	Kurma Süresi
CLK	Referans Zaman Sinyali
Tcnst	Zaman Sabiti
R	Reset
CU	İleri Sayma
Q	Çıkış
LD	Yükleme
CD	Geri Sayma
PV	Kurma Değeri

1. GİRİŞ

Teknolojik gelişmelerin, günlük hayatımızın ayrılmaz bir parçası haline geldiğini ve teknolojik cihazlarının hızla gelişmekte olduğunu görmekteyiz. Bu cihazların ilgi görmesinde, maliyet/işgücü'nün azalması kalitenin artırılması gibi faktörler önemli rol almaktadır. Ancak tüm bu gelişmeler insan hayatını kolaylaştırdığı ve insanların ihtiyaçlarını kolay, hızlı bir şekilde giderebildiği zaman esas hedefine ulaşmış olur [1].

İnsanların daha iyiye ulaşma çabaları sonucu gerçekleştirdikleri çalışmalarla her geçen gün teknolojide ilerleme kaydedilmektedir. Gerçekleşen bu ilerlemeler sonucu daha iyi hizmet sağlamak için otomasyon teknolojisinde makineleşme süreci hızlanmıştır. Otomasyon teknolojisi sayesinde insanlar için tehlikeli olan işlerde teknolojinin kullanımı yaygınlaşmış, çalışan kişi sayısı minimuma indirilerek üretimde ucuzluk, kalite artmış, zaman kaybı ve mali kayıp azalmıştır.

Günümüzde Programlanabilir lojik denetleyiciler (Programmable Logic Controllers - PLC) otomasyon işlemlerinin gerçekleştirilmesinde yaygın olarak kullanılmaktadır. Bunun en önemli sebeplerinden birisi PLC'lerin endüstriyel ortamlarda (nem, ısı, mekanik v.b.) çalışmak üzere tasarlanmış olması ve daha iyi hizmet sağlamasından dolayı tercih edilmesidir.

Kullanılan alanlarda işlevlerine ve yapılan işin büyüklüğüne göre PLC'ler farklı özellikler göstermektedirler. Günümüzde çok sayıda firma çeşitli özellikler ve fiyat yelpazesinde ticari PLC üretimi yapmaktadır. Her firma kendi PLC'si ile birlikte programlama yazılımını da satmaktadır. Bu durum ilk alımlarda maliyet artışına neden olmaktadır. PLC'ler için birçok programlama dili mevcut olup yaygın kullanılan dil merdiven (Ladder) diyagramı programlama dilidir.

Ticari PLC'lere alternatif ucuz maliyetli Mikrodenetleyici temelli PLC'ler önerilmiştir. Bu PLC'lerden en kapsamlı olanı Uzam tarafından tasarlanan PIC16F877A ve PIC16F648A temelli PIC-PLC'lerdir [2-27]. Bu PLC donanımları için halihazırda programlama işlemi mikrodenetleyicinin kendi makine dili tabanlı makrolar ile sağlanabilmektedir. PIC-PLC'leri programlamak için en yaygın kullanılan programlama dili olan Ladder dilinin bu makrolara dönüştürülmesi ihtiyacı vardır.

Bu tez kapsamında PIC-PLC'ler için görsel olarak girilen Ladder diyagramlarını PIC-PLC'nin kendi makrolarına dönüştüren bir arayüz tasarlanmış ve gerçekleştirilmiştir. Arayüzün farklı işletim sistemi yüklü bilgisayarlarda ve cihazlarda çalışabilmesi için web sayfası formatında olması tercih edilmiştir. Dolayısıyla gerçekleştirilen arayüz farklı donanımlar kullanılarak PIC-PLC'ler için kod oluşturulabilmektedir. Kullanıcılara çizdikleri ladder diyagramlarını, saklama, yeniden açma ve değiştirme imkânları sunulmaktadır. Arayüze entegre edilen PIC mikrodenetleyicileri için üreticisi tarafından ücretsiz olarak sunulan MPASM programı ile çizilen kod kolayca mikrodenetleyiciye yüklenecek makine koduna dönüştürülür ve yüklemeye hazır dosya elde edilebilmektedir.

Arayüz çeşitli ladder diyagramı örnekleri kullanılarak test edilmiştir. İlk olarak temel kodların çevrimini incelemek üzere; kontak ve röle temelli diyagramlar, zamanlayıcı, sayıcı, flip-flop, karşılaştırma ve aritmetik işlemleri içeren kodların dönüşümü dikkate alınmıştır. Arayüz kullanılarak gerçek bir sistem için otomasyon kodunun üretilebileceği deneysel bir dağıtım istasyonu kullanılarak gösterilmiştir. Tezin organizasyonu aşağıdaki şekildedir.

Bu tez çalışması beş bölümden oluşmaktadır. Birinci bölüm "Giriş" bölümüdür. Burada tez çalışmasının içeriğinden bahsedilmiştir. İkinci bölümde PIC-PLC'ler ile ilgili kaynak araştırması ve kuramsal temellere yer verilmiştir. Bu bölümde ayrıca PIC temelli PLC'ler ile ilgili bilimsel çalışmaların özetlerine yer verilmiştir. Üçüncü bölüm "Materyal ve Yöntem" bölümüdür. Bu bölümde kullanılan PIC16F877A ile PIC16F648A temelli PLC donanımları ve çalışmada kullanılan makrolara yer verilmiştir. Dördüncü bölüm "Bulgular ve Tartışma" bölümüdür. Bu bölümde PIC-PLC'leri programlamak için tasarlanan yazılım ve bu yazılıma ait uygulamalara yer verilmiştir. Beşinci bölüm olan sonuç bölümünde ise bu çalışmadan elde edilen sonuçlar açıklanmaktadır.

2. KAYNAK ÖZETLERİ-KURAMSAL TEMELLER

2.1. Kaynak Özetleri

Ticari PLC'lere alternatif olarak mikroişlemci veya mikrodenetleyici temelli daha ucuz sistemler çözüm olarak sunulmuştur [1,28-36]. Atmega128 entegresi kullanarak 14 adet dijital giriş/çıkış ve 5 adet sadece dijital çıkışlı PLC Tongur tarafından gerçekleştirilmiştir [29]. Ancak, tasarlanan bu PLC için kod üretilebilen bir arayüz programı mevcut değildir [29]. Yapılan bir diğer çalışmada ise tasarlanan PLC sadece giriş çıkış ve temel işlemleri yerine getirebilmektedir [32]. Rafat tarafından yapılan çalışmada ise tasarlanan PLC için arayüz mevcuttur ancak PLC'nin giriş çıkış sayısı çok kısıtlıdır [28]. Erkol tarafından yapılan çalışmada tasarlanan PLC için PIC kodları üretebilen bir arayüz programı hazırlanmıştır ancak gerçek bir sistemde test edilmemiştir [30]. Literatürdeki diğer çalışmalarda ise simülasyon amaçlı programlar ve donanımlar tasarlanmıştır [33-35].

Mikrodenetleyici entegreleri incelendiğinde ücretsiz derleyici programı sunulması, fiyatlarının uygun olması ve kaynak kod desteğinin fazla olması gibi sebeplerden dolayı Microchip firmasının üretmiş olduğu PIC serisi denetleyicilerin yaygın olarak kullanıldığı görülmektedir. PIC mikrodenetleyicileri kullanılan ve ticari ürün olarak satışı MikroElektronika firması tarafından gerçekleştirilen PICPLC16 isimli bir PLC de mevcuttur [36]. Ancak bu ürün içinde alışılmış PLC komutları ile programlama yapılmasına olanak sağlayan bir arayüz mevcut değildir.

PIC mikrodenetleyicileri kullanılarak gerçekleştirilen en ciddi tasarımlar Uzam tarafından yapılmış ve raporlanmıştır [2-27]. Bu çalışmalarda PIC16F877A entegresi kullanılarak 16 giriş ve 16 çıkışlı bir PLC ile PIC16F648A entegresi kullanılan ve giriş çıkış sayısı modül eklenerek arttırılabilen iki PLC tasarımı önerilmiştir. Bu PLC'leri programlamada kullanmak üzere yaygın kullanılan PLC programlama dili olan STL diline benzer makro tabanlı komutlar önerilmiştir. Ancak, bu çalışmalarda da kullanıcının görsel olarak merdiven diyagramı programı girebileceği bir arayüz mevcut değildir. Tasarlanmış bu PLC'ler PIC assembly dili kullanılarak programlanmaktadır.

Bu tezde Uzam tarafından önerilen PIC-PLC donanımları için web tabanlı, merdiven (Ladder) diyagramı programlamayı sağlayacak bir programlama arayüzü

gerçekleştirilmiştir. Oluşturulan yazılım sayesinde ortamdan bağımsız olarak kullanılarak merdiven diyagramı çizimi sağlayacak, çizilen PLC diyagramlarını ilgili mikrodenetleyici koduna dönüştürülerek kullanıma hazır bir kod elde edilmektedir.

Tezin bu bölümünde temel konular olan Programlanabilir Lojik Denetleyiciler (PLC) ve mikrodenetleyiciler kısaca tanıtılmaktadır.

2.2. Programlanabilir Lojik Denetleyiciler (PLC)

Programlanabilir Lojik Denetleyici (PLC) otomasyon sistemlerinde kontrol işlemlerinin gerçekleşmesine olanak sağlayan, uygun yapıda tasarlanmış giriş ve çıkış birimi ile bağlantı kombinasyonları içeren, programlanabilir bir endüstriyel otomasyon cihazıdır [37].

Günümüz otomasyon sistemlerinin verim ve kalitenin artmasıyla kendini sürekli yenilediği ve her geçen gün hızlı bir şekilde geliştirdiği görülebilmektedir. Bu hızlı gelişme sürecinde sistemlerin hızlı adaptasyonu için PLC otomasyon sistemlerinin kullanımı son derece önem taşımaktadır.

İlk ticari PLC, 1969 yılında röleli elektriksel kumanda devrelerinin yerine kullanılmak üzere Modicon firması tarafından geliştirilmiş ve üretilmiştir. Röle içerikli kumanda aygıtlarının yerine kullanılması için geliştirilen bu cihaz, sadece temel mantık işlemlerini içerdiğinden dolayı programlanabilir lojik denetleyici (PLC) olarak adlandırılmıştır.

İlk ticari PLC'ler sanayide başarılı şekilde uygulandıktan sonra, General Electric, Siemens, GEC gibi firmalar yüksek maliyette ve kalitede PLC'ler üretmiştir. Bu ürünlere alternatif düşük maliyetli PLC'ler ise Omron, Toshiba gibi firmalar tarafından üretilmiştir.

Günümüzde üretilmekte olan PLC'ler eskileri ile karşılaştırıldığında, başlangıç düzeyine göre işlem yetenekleri, giriş çıkış sayısı gibi özellikleri dikkate alındığında çok ileri düzeyde bir gelişme sağlandığı görülür.

Karmaşık kumanda devrelerinin ve kontrol işlemlerinin PLC'ler ile daha kolay gerçekleştirilmesi mümkündür. Bu durum PLC'lerin program bellek boyutu, giriş çıkış sayısı ve işlem yeteneği gibi birçok özelliğinin günlük hayatta kullandığımız bilgisayarlar düzeyine ulaştığını göstermektedir.

PLC'nin kontrol sistemlerinde tercih edilmeye başlanması, alışılmış PLC adının kullanılmasında anlaşmazlıklara neden olmuştur. Birçok ürün geliştirici firma, bu kontrolörlerin geri beslemeli kontrol sistemlerinde veya mantıksal temelli kumanda devrelerinde kullanılmaları nedeniyle, PLC yerine, programlanabilir kontrolör ifadesini kullanmayı daha geçerli bulmuş ve kişisel bilgisayarlardan ayırmak amacıyla, kısaca PLC olarak tanımlamıştır. Ülkemizde de PLC adı daha çok kullanılmaktadır [37].

PLC'ler genel olarak, sanayiye uygun şekilde tasarımı yapılmış dijital kurallara göre yazılan fonksiyonu gerçekleştiren, giriş-çıkış birimleri sayesinde sistemle veri alışverişinde bulunan, içerisinde barındırdığı sayma, saklama, zamanlama gibi fonksiyonlarıyla kontrol sağlamaya yarayan elektronik aygıtlardır. PLC'ler, kontrol edilen sistemde olabilecek olan fiziksel kombinasyonları farklı ölçüm cihazları ile belirleyerek, karşıdan gelen bilgileri kullanıcının önceden yazmış olduğu programa göre değerlendirir ve değerlendirmeye ait ortaya çıkan sonuçları sisteme yansıtır [38].

Sistemde ortaya çıkan bilgiler, çalışma anında meydana gelen değişimlerin elektriksel sinyallere dönüştürülmüş halidir. Ortaya çıkan bilgi analog ise ortaya çıkan değer belirlenir, dijital ise sinyalin varlığı veya yokluğuna göre sorgulama yapılır. PLC ile sadece bir cihaz kontrolü gerçekleştirilebileceği gibi, bir sanayi tesisinin komple kontrolü de gerçekleştirilebilir. Oluşabilecek olağan fark sadece kullanılan sistemin işlem kapasitesidir. PLC'lerin bugün her sektörde kullanıldığı yadsınamaz bir gerçektir. Gıda sektöründen otomotiv sektörüne, depolama işlemlerine kadar birçok alanda kontrol sistemlerinin vazgeçilemeyecek elemanlarından biridir. Bu nedenle karmaşık otomasyon problemlerini PLC'ler ile hızlı ve güvenli bir şekilde çözmek mümkündür. PLC'lerin avantajları aşağıdaki gibi sıralanabilir [39,47];

- PLC'ler kullanılarak otomasyon sistemleri daha kolay tasarlanabilir.
- PLC'ler fazla yer kaplamaz ve çok az arıza yaparlar.
- PLC'lerin adaptasyon süreçleri hızlıdır, donanım değiştirilmeden sadece yazılımsal bir değişiklikle yeni uygulamaya geçilebilir.
- PLC'ler ortam şartlarından fazla etkilenmezler, bakımları kolaydır.
- PLC'ler bağlantı için fazla kablolama istemezler.
- PLC'ler var olan hazır fonksiyonları kullanabilirler.

- PLC'lerin diğer cihazlar ile haberleşme imkanı vardır, bu özelliği bilgisayarlı otomasyon işlemine olanak sağlar.

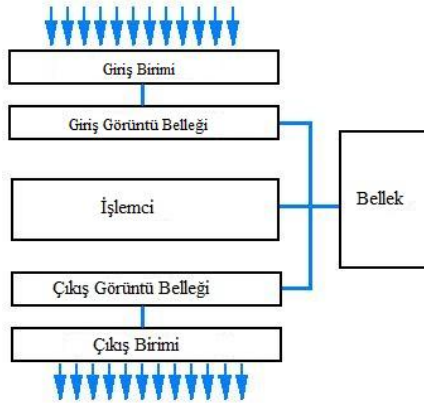
- PLC'lerin giriş ve çıkışlarının durumları izlenebilir, arıza aramayı kolaylaştırırlar ve güvenilirlerdir.

2.3. PLC'lerin Yapısı

Bir PLC temel olarak dört ana bölümden oluşur. Bunlar;

- Giriş Birimi (INPUT Modülü)
- Çıkış Birimi (OUTPUT Modülü)
- İşlemci Birimi (CPU)
- Bellek Birimi'dir.

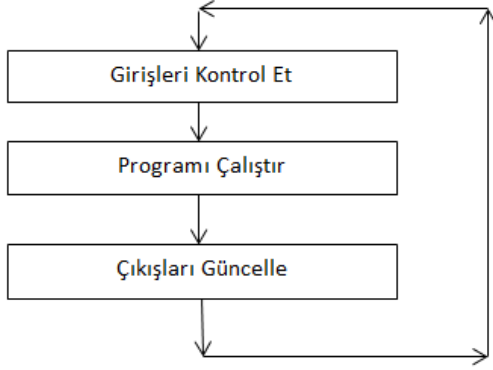
Giriş birimi çeşitli elektriksel özelliklerdeki kumanda ya da geri besleme işaretlerinin PLC'de işlenecek sayısal büyüklüklere dönüştürülmesini sağlar, çıkış birimi ise PLC'de işlenen verilere ilişkin sayısal değerlerin kontrol edilen sisteme uygun elektriksel işaretlere dönüştürülmesini sağlayan birimdir. İşlemci birimi, PLC'nin belleğinde kayıtlı olarak bulunan sistem programı aracılığıyla çalışmasını düzenler ve kullanıcıya ait programları yürüten en önemli birimdir [37]. Şekil 2.1.'de PLC'nin genel yapısı görülmektedir.



Şekil 2. 1. PLC'nin genel yapısı

Bir PLC'nin çalışması Şekil 2.2.'de görüldüğü gibi özetlenebilir. PLC ilk önce tüm girişleri kontrol ederek, girişlere ait bilgileri giriş görüntü belleğine kayıtlı eder. Sonra bellekteki bu giriş bilgilerini kullanan PLC programını adım adım çalıştırır. Daha

sonra programın çıkış olarak ürettiği bilgiler çıkış görüntü belleğinde güncellenir ve tekrar girişler kontrol edilir. Bu işlemler sürekli olarak tekrar edilir.



Şekil 2. 2. PLC'nin çalışma mantığı

2.4. PLC'lerin Programlanması

PLC'leri programlayabilmek için çok çeşitli diller geliştirilmiştir. Bu dillerden bazıları International Electrotechnical Commission tarafından IEC 61131-3 standardı olarak yayınlanmıştır [50]. Bu standartta yer alan ve yaygın olarak kullanılan PLC programlama dilleri,

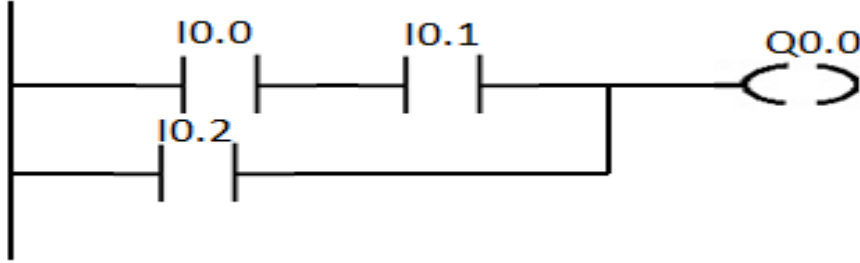
- Merdiven diyagramı (LADDER)
- Deyim listesi (STL)
- Fonksiyon Blok Diyagramı (FBD)'dir.

PLC'lerin programlanmasında yukarıda bahsedilen programlama dillerinden Deyim Listesi (STL) ve merdiven diyagramı (LADDER) programlama, genellikle el programlayıcılarında kullanılır. Üretici firmaların PLC'lerle birlikte verdikleri editör programı yardımıyla üç çeşit programlama yöntemi de kullanılabilir [40].

2.4.1. Merdiven (Ladder) Diyagramı Programlama

Merdiven diyagramı ile programlama geleneksel kumanda devrelerinin standart çizimlerinden esinlenerek geliştirilmiş bir grafiksel programlama şeklidir. Geleneksel röleli kumanda devreleri, normalde açık ve normalde kapalı kontaklar, röle bobinlerinden oluşan devrelerdir. Bu devreler tasarlanırken belirli çizim kuralları ve sembelleri kullanılır. Bu konuda çeşitli standartlar bulunmaktadır. En yaygın kullanılan

standart Avrupa’da IEC (International Electrotechnical Commission) standardıdır. Örnek bir Merdiven (Ladder) diyagramı Şekil 2.3.’de görülmektedir. Merdiven (Ladder) diyagramı programlama yaygın kullanılan bir PLC programlama dilidir.



Şekil 2. 3. Örnek bir merdiven diyagram programı

2.4.2. Deyim Listesi (STL, Statement List)

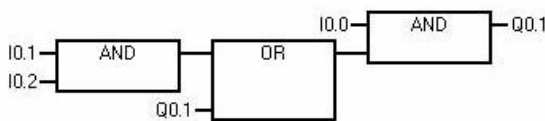
PLC işlem komutlarının art arda yazılarak yapıldığı bir komut ile programlama biçimidir. Bütün PLC programlama yazılımları tarafından desteklenen bir yöntemdir. Her PLC üreticisi kendi PLC’sine ait komutlar içeren Deyim Listesi sunmaktadır. Siemens PLC’lere ait örnek bir STL programı Şekil 2.4.’de görülmektedir.

ld	IO.0
and	IO.1
or	IO.2
out	Q0.0

Şekil 2. 4. Örnek bir STL programı

2.4.3. Fonksiyon Blok Diyagram (FBD, Function Block Diagram)

PLC’lerin lojik işlemleri simgeleyen lojik kapı şekilleri ile görsel programlanmasıdır. Bu programlama biçiminde lojik kapı şekilleri, art arda seri ya da paralel bağlanarak, program yazılır [37]. Örnek bir FBD programı Şekil 2.5.’de görülmektedir.



Şekil 2. 5. Örnek FBD diyagramı

2.5. Mikrodenetleyiciler

İçerisinde giriş-çıkış, bellek ve mikroişlemci üniteleri bulunan, tek bir entegre haline getirilmiş elemanlara mikrodenetleyici denir. Bu sayede hem maliyet azalmış hem de programlama kolaylığı sağlanmıştır [41]. Mikrodenetleyicilerde bulunan giriş-çıkış sayıları, bellek miktarı gibi özellikler mikrodenetleyicinin modeline göre değişmektedir. Çok sayıda mikrodenetleyici üreten firma ve bu firmaların ürettiği mikrodenetleyici mevcuttur.

Tez kapsamında kullanılan mikrodenetleyici temelli PLC'lerde Microchip firması tarafından üretilen PIC mikrodenetleyicileri bulunmaktadır [44]. Tezin bu kısmında ilgili mikrodenetleyiciler kısaca tanıtılacaktır.

2.5.1. PIC Mikrodenetleyicileri

PIC mikrodenetleyiciler hızlı çalışmaları amacıyla RISC (Reduced Instruction Set Controller) işlemci olarak tasarlanmışlardır. Bu mikrodenetleyicilerde komut sayısı oldukça azdır. Komutların tek bir çevrimde işlenmesi mikrodenetleyicinin hızını oldukça arttırmaktadır. Bir komut bir işlemcide iki çevrimde gerçekleşiyorken diğer işlemcide tek çevrimde gerçekleşmesi demek, ikinci işlemcinin birinciye oranla iki kat hızlı çalışması anlamına gelmektedir.

PIC mikrodenetleyiciler bellek mimarisi olarak Harvard mimarisini kullanmaktadır. Bu mimaride program ve veri saklama bellekleri (RAM) birbirinden ayrı yapıdadır. Bu durum, program ve veri saklamak için kullanan Von Neumann mimarili mikrodenetleyicilere göre bir miktar daha pahalı olmaları anlamına gelmektedir.

Microchip firması son on yılda yaptığı büyük atılımla mikrodenetleyici pazarında Motorola'dan sonra en büyük üretici firma konumuna gelmiştir. 1990 öncesinde flash teknolojisine sahip mikrodenetleyiciler yok denecek kadar azken, var olanlar ise fiyat olarak endüstri kullanımına uygun değillerdi. Şu an artan satışlarla düşen maliyetler sayesinde, flash entegreler ucuzlamışlardır.

Mikrodenetleyici temelli PLC donanımlarında Microchip firmasının ürettiği PIC16F877A ile PIC16F648A mikrodenetleyiciler kullanılmıştır.

2.5.2. PIC16F877A Mikrodenetleyicisinin Genel Özellikleri

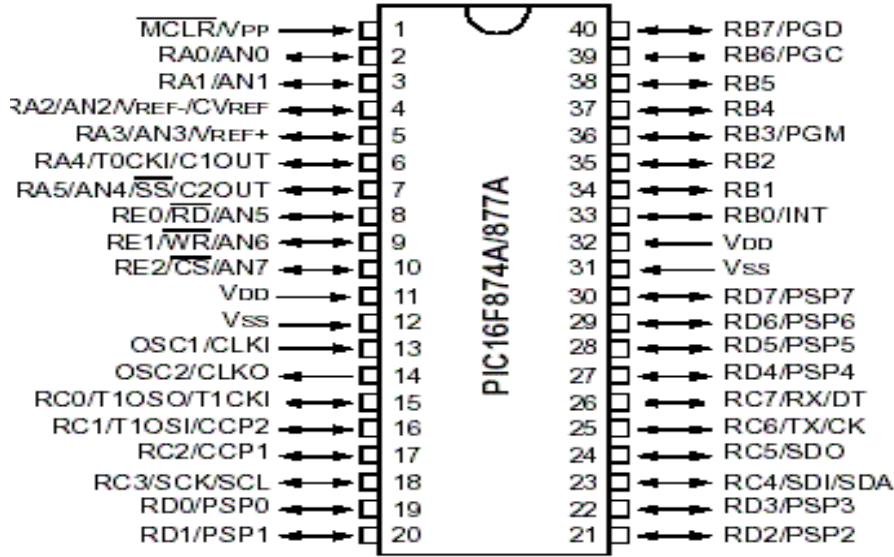
Bu entegre PIC16X ailesine mensup bir mikrodenetleyicidir. 8K x 14 bit flash program belleğine sahiptir. PIC komutları goto, call gibi yönlendirme komutları dışında hep 1 byte olduğundan, yaklaşık 8000 satır program yazılmasına imkan tanır. Flash bellek özelliği olduğundan 100000 kez üst üste program yazma ve silme işlemi yapılabilmektedir. Harvard mimarisinin ayrı komut ve veri taşıyıcısıyla 8 bitlik veri taşınmasına ve 14 bitlik komut kelimesine imkân vermektedir. RAM olarak tanımladığımız, değişkenlerin bulunduğu veri belleği 368 x 8 bit'tir ve 256x8 byte lık EEPROM (elektriksel olarak programlanabilir okuma belleği) hafızaları vardır [43].

PIC16F877A'de 3 adet zamanlayıcı-sayıcı bulunmaktadır. İki adet Capture, Compare, PWM modülü ile 12,5 ns hassasiyetinde yakalama (capture), 200 ns hassasiyetinde karşılaştırma, 10 bit'lik çözünürlükle PWM sinyali uygulama özelliği mevcuttur. Paralel iletişim portuna sahiptir. 10 bit'lik, 8 kanallı Analog/Dijital çeviriciye sahiptir. PIC16F877A, 8 katlı bir yığın yapısına sahiptir. Bu iç içe sekiz alt program çağırabilmemiz anlamına gelir. PIC16F877A'nın bacak bağlantıları Şekil 2.6.'da özellikleri ise Çizelge 2.1.'de görülmektedir [19].

PIC16F877A programlanabilir Code Protection özelliği sayesinde program belleğinin bir kısmını okumaya karşı koruma altına alır, diğer kısmını ise okunabilir kılar. Düşük gerilimlerde kararsız durumlarla karşılaşmamak için BODEN adı verilen özellik ile belli bir gerilim altında reset çalıştırılır. Az enerji tüketmek amacıyla kullanılan Sleep (uyuma) özelliği mevcuttur. Düşük enerji tüketir ve 3 V ile 5,5 V arasında DC gerilimlerle çalışabilir.

2.5.3. PIC16F648A Mikrodenetleyicisinin Genel Özellikleri

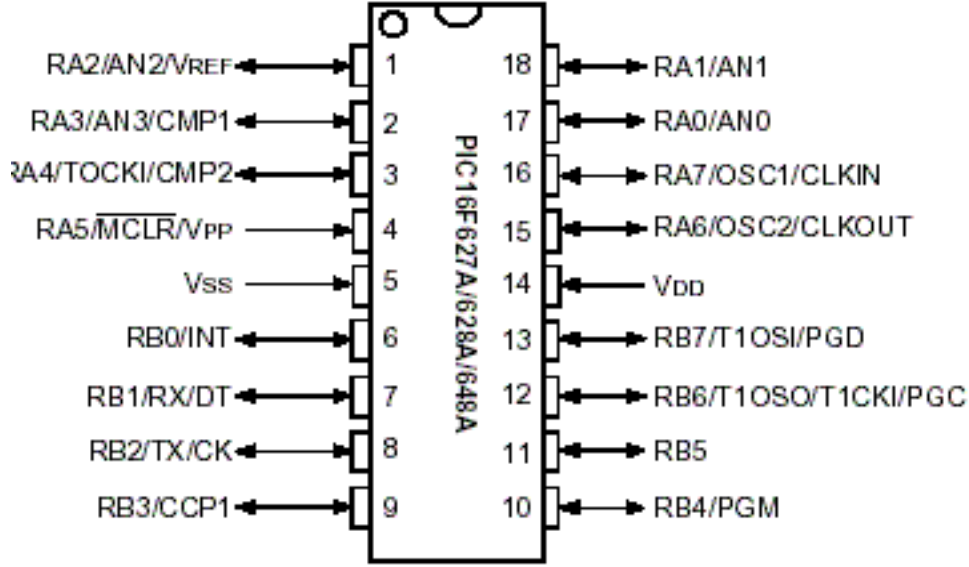
PIC16F648A, Harvard mimarisi ile üretilmiş RISC yapılı, 8 bit'lik bir mikrodenetleyicidir. RISC mimarisinin getirdiği azaltılmış komut seti avantajından ötürü programlanması çok kolaydır. PIC16F648A'da komut uzunluğu 14 bit, işlenecek verinin uzunluğu 8 bit'tir. Flash bellek yapısına sahip olduğu için devrenin elektriği kesilse bile içindeki program saklamaya devam etmektedir. PIC16F648A'nın bacak bağlantıları Şekil 2.7.'de özellikleri ise Çizelge 2.2.'de görülmektedir [44].



Şekil 2. 6. PIC16F877A'nın bacak bağlantıları

Çizelge 2. 1. PIC16F877A'nın genel özellikleri

Saat Frekansı	En Yüksek Çalışma Frekansı	20 MHz
Bellek	Flash Program Belleği	8Kx14 bit
	RAM Veri Belleği	3Kb
	EPROM Veri Belleği	256Bayt
Çevresel Birimler	Zamanlayıcı Modülleri	TMR0,TMR1,TMR2
	Karşılaştırıcı	2
	Yakalama / Karşılaştırma / PWM Modülleri	16 Bit Yak / Kar 10 Bit PWM
	A / D Çevirici	8 Kanal 10 Bit
	USART / SCI	9 Bit Adresli
	Paralel Slave Port	8 Bit, Harici RD, WR e CS
	Dahili Voltaj Referans Birimi	Var
Diğer Özellikler	Kesme Kaynakları	14
	Giriş / Çıkış Bacakları	33
	Çalışma Gerilim Aralığı	3-5,5 V
	Brown-Out Reset	Var



Şekil 2. 7. PIC16F648A bacak bağlantıları

Çizelge 2. 2. PIC16F648A genel özellikleri

Saat Frekansı	En Yüksek Çalışma Frekansı	20 MHz
Bellek	Flash Program Belleği	8Kx14 bit
	RAM Veri Belleği	256 Bayt
	EPROM Veri Belleği	256 Bayt
Çevresel Birimler	Zamanlayıcı Modülleri	TMR0,TMR1,TMR2
	Karşılaştırıcı	2
	Yakalama / Karşılaştırma / PWM Modülleri	1
	Seri Haberleşme	USART
	Dahili Voltaj Referans Birimi	Var
Diğer Özellikler	Kesme Kaynakları	10
	Giriş / Çıkış Bacakları	16
	Çalışma Gerilim Aralığı	3-5,5 V
	Brown-Out Reset	Var

3. MATERYAL VE YÖNTEM

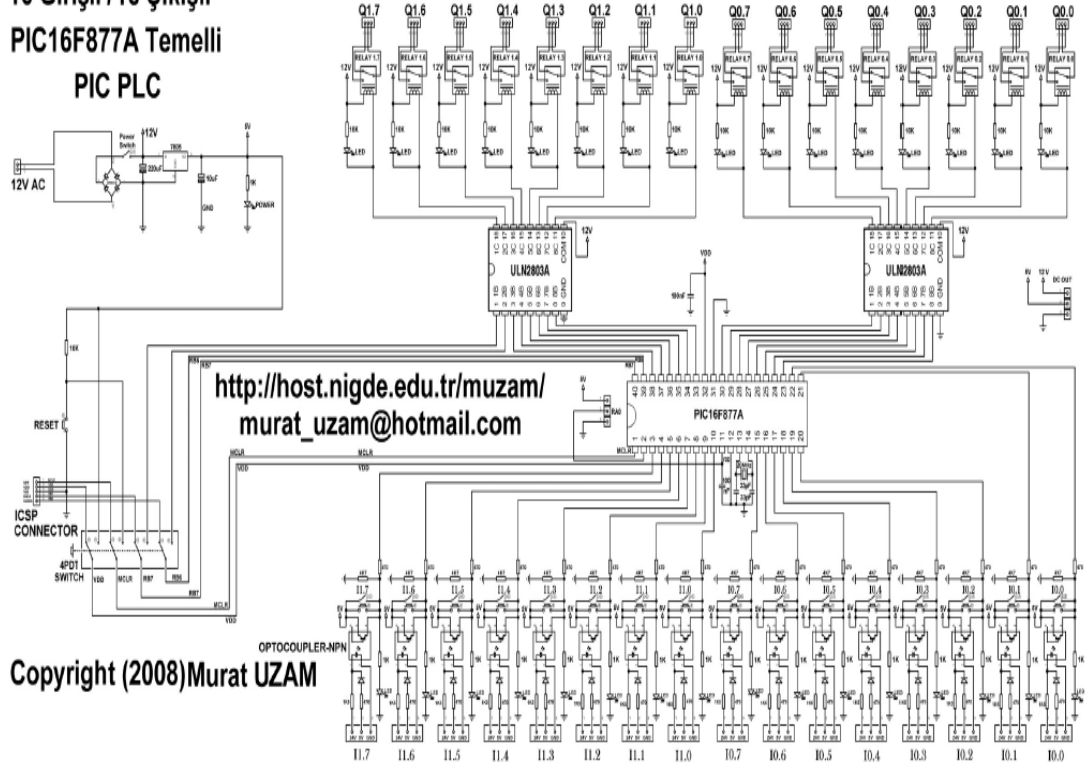
Bu bölümde; tez kapsamında kullanılan mikrodnetleyici temelli PLC'lerin donanımı ve yazılımını oluşturan temel makrolar tanıtılmaktadır. Mikrodnetleyici temelli PLC'lerde bir önceki bölümde tanıtılan PIC16F877A ve PIC16F648A mikrodnetleyicileri kullanılmıştır. PIC16F877A kullanılan donanım 16 giriş ve 16 çıkışa sahiptir. PIC16F648A kullanılan donanım ise üzerinde 8 giriş ve 8 çıkış barındıran ana karta 8'in katları olacak şekilde giriş çıkış içeren modüller eklenerek genişletilebilmektedir. İlgili donanım ve makrolar Uzam tarafından tasarlanmış ve PIC-PLC olarak adlandırılmıştır [2-27]. Tezin ileriki kısımlarında mikrodnetleyici temelli PLC yerine PIC-PLC ifadesi kullanılacaktır.

3.1. PIC Mikrodnetleyici Temelli PLC'ler

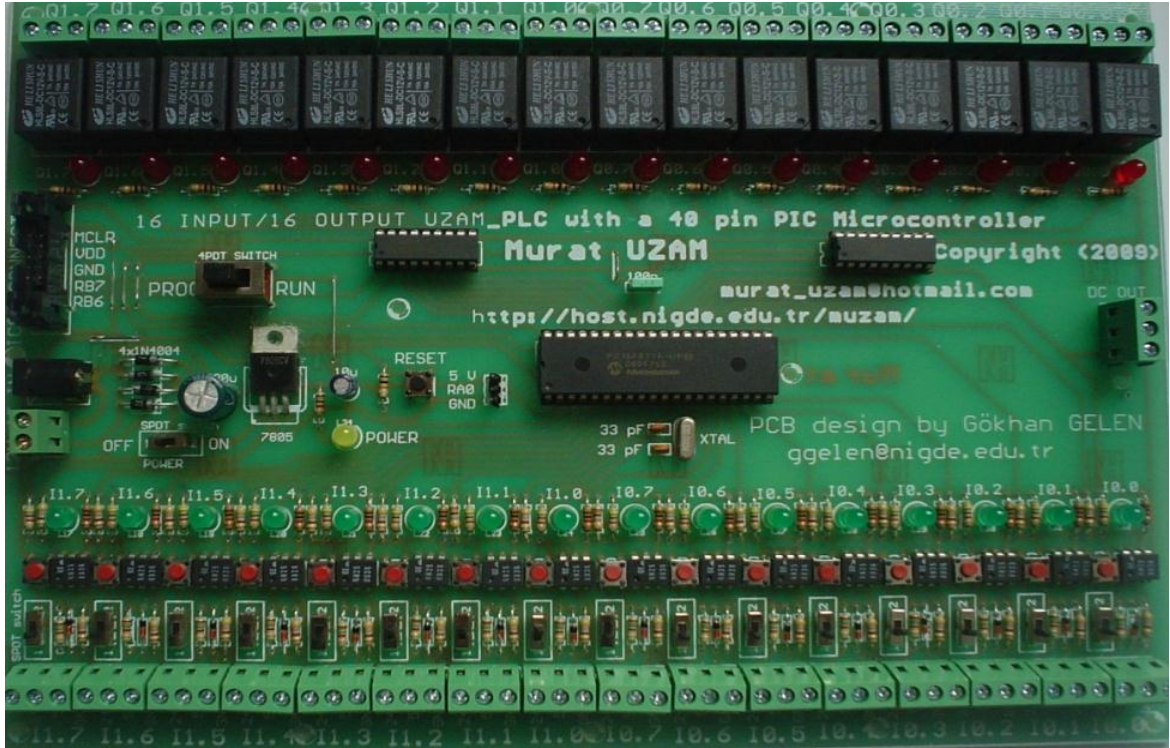
3.2. PIC16F877A Temelli PIC-PLC Donanımı

PIC16F877A temelli PIC-PLC'de 16 adet giriş ve 16 adet çıkış portu bulunmaktadır. Temelde tasarlanan donanım beş kısımdan meydana gelmektedir. Bu bölümler sırasıyla rölelerin çalışması için gereken güç birimi, donanıma yazılımın yüklenmesi için gerekli olan program yükleme birimi, PLC'nin çalışmasını sağlayan merkezi işlem birimi (CPU), donanıma bilgi girişinin sağlandığı Giriş birimi ve bilgilerin çıkışını sağlayan Çıkış birimidir. PIC16F877A temelli PIC-PLC donanımına ait devre şeması Şekil 3.1.'de ve PIC-PLC'nin resmi Şekil 3.2.'de görülmektedir. Donanıma ait daha detaylı bilgi [2-27] numaralı referanslarda bulunabilir.

16 Girişli / 16 Çıkışlı
PIC16F877A Temelli
PIC PLC



Şekil 3. 1. PIC16F877A temelli PIC-PLC devre şeması



Şekil 3. 2. PIC16F877A temelli PIC-PLC'nin resmi

3.2.1. Girişler

PIC16F877A temelli PIC PLC kartında I1.7, I1.6, I1.5, I1.4, I1.3, I1.2, I1.1, I1.0, I0.7, I0.6, I0.5, I0.4, I0.3, I0.2, I0.1 ve I0.0 olarak isimlendirilen 16 adet giriş mevcuttur. Her bir giriş 5V DC ya da 24 V DC gerilim ile çalışmaktadır. Girişlerin tamamı opto-coupler kullanılarak yalıtılmıştır. Girişlerin simülasyonu yapılabilmesi için kart üzerine yerleştirilen LED'ler ile girişlerin aktifliğini pasifliğini kontrol edilebilmektedir. Ayrıca her bir giriş için anî temaslı bir buton ve bir sürgülü anahtar mevcuttur. Kartın giriş bloğuna ait resim Şekil 3.3.'de görülmektedir.



Şekil 3. 3. Giriş bağlantıları

3.2.2. Çıkışlar

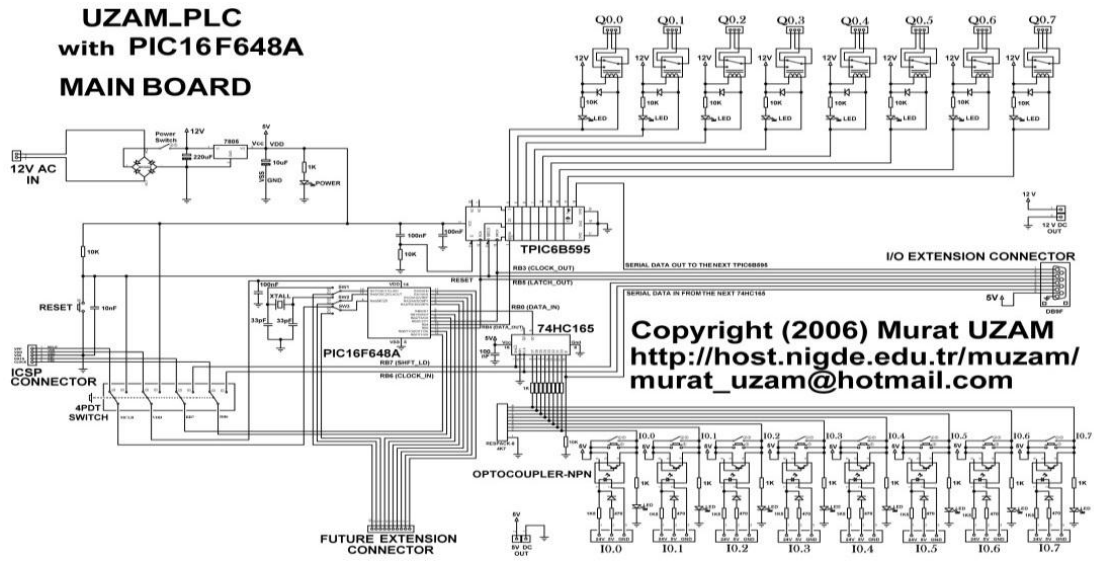
PIC16F877A temelli PIC-PLC kartında 16 adet röle çıkışı Q1.7, Q1.6, Q1.5, Q1.4, Q1.3, Q1.2, Q1.1, Q1.0, Q0.7, Q0.6, Q0.5, Q0.4, Q0.3, Q0.2, Q0.1 ve Q0.0 olarak isimlendirilmiştir. Her bir çıkış rölesi 12 V DC gerilim ile çalışmaktadır. Her biri 8 röleyi süren iki adet ULN2803A transistör sürücü entegresi yardımıyla 16 röle çalıştırılmaktadır. Her rölede, C (*common* – ortak uç), NC (*normally closed* – normalde kapalı) ve NO (*normally open* – normalde açık) olmak üzere üç bağlantı ucu olan SPDT (*single pole double throw* – tek gruplu iki konumlu) kontak mevcuttur. Çıkışların simülasyonunun yapılabilmesi için kart üzerine yerleştirilen LED'ler ile çıkış rölelerinin durumunu kontrol edilebilmektedir. Kartın çıkış bloğuna ait resim Şekil 3.4.'de görülmektedir.



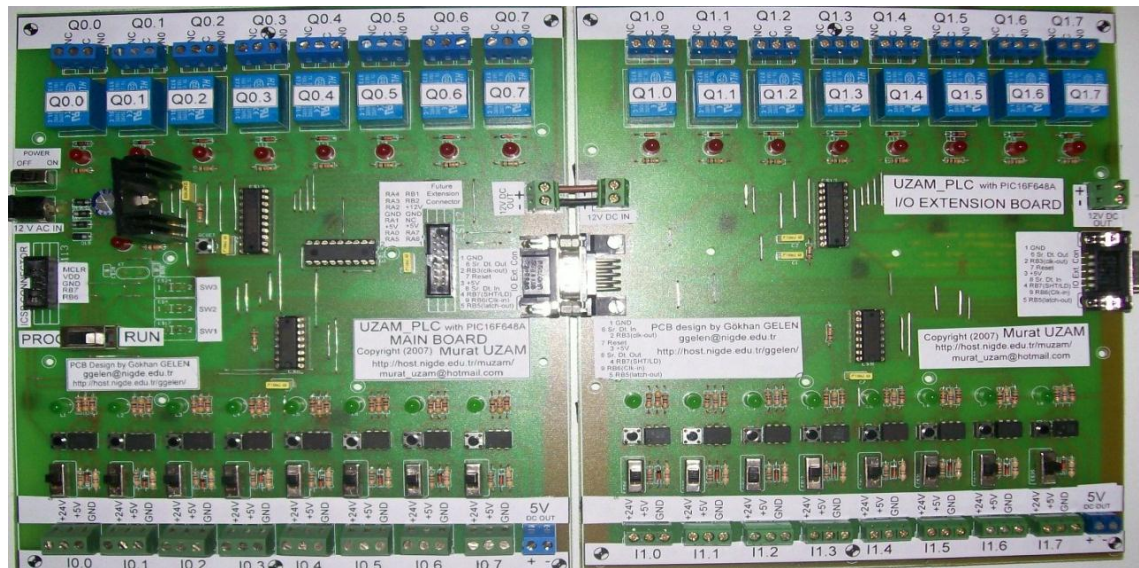
Şekil 3. 4. Çıkış bağlantıları

3.3. PIC16F648A Temelli PIC-PLC Donanımı

PIC16F648A temelli PIC-PLC modüler yapıdadır. Temelde iki farklı modülden oluşur. Bunlardan ilki üzerinde 8 giriş ve 8 çıkış ile işlemciyi barındıran karttır. Diğer modül üzerinde ise sadece 8 giriş / 8 çıkış bulunmaktadır. İşlemcili karta giriş-çıkış kartı eklenerek 8'in katları olacak şekilde giriş-çıkış sayısının artırılması mümkündür. PIC16F648A temelli PIC-PLC donanımına ait şema Şekil 3.5.'de ve PIC-PLC'nin resmi ise Şekil 3.6.'da görülmektedir.



Şekil 3. 5. PIC16F648A temelli PIC-PLC devre şeması



Şekil 3. 6. PIC16F648A temelli PIC-PLC'nin resmi

3.4. PIC-PLC Yazılımı

Bu bölümde, PIC-PLC'nin temel yazılım yapısı incelenmektedir. Bilindiği gibi bir PLC sürekli olarak şu işlemleri yerine getirmektedir: İlk olarak Girişler güncellenir. Sonra kullanıcı programı taranır ve son olarak ta çıkışlar güncellenir. Bu işlemler sonsuz bir program döngüsü içerisinde gerçekleştirilmektedir. PLC çalıştığı sürece bu işlemler sürekli olarak tekrarlanır. Bu sonsuz döngüye girmeden önce PLC başlangıç şartları da düzenlenmelidir. PIC-PLC yazılımında da aynı işlemler yerine getirilmektedir. Bu işlemlere ait temel yazılım dosyası Şekil 3.7.'de görülmektedir. Bu programda “initialize” başlangıç şartlarının gerçekleştirildiği makroyu ifade eder. “send_outputs” makrosu da çıkışların güncellenmesiyle ilgili işlemler için kullanılmaktadır. Kullanıcının yapmış olduğu programlar bu “get_inputs” ve “send_outputs” makrolarının arasına yazılarak PIC-PLC'nin programlanması sağlanır [2].

```
list      p=16F877A                                ;işlemciyi tanımlamak için liste direktifi
#include <p16F877A.inc>                             ;işlemciye has değişken tanımları
#include <PICPLC_877_16I_160_tnmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_16I_160_kontak.inc>;kontak ve röle temelli makrolar

__CONFIG __CP_OFF & __WDT_ON & __BODEN_OFF & __PWRTE_ON & __HS_OSC & __LVP_OFF & __DEBUG_OFF & __CPD_OFF

org      0x00                                ;Reset vektörü
main
  initialize
scan
  get_inputs
  ----- kullanıcı programı burada başlar -----
  ----- kullanıcı programı burada biter -----
  send_outputs
  goto   scan
end
```

Şekil 3. 7. Temel yazılım dosyası [2]

3.5. PIC PLC Program Makroları

Bu kısımda PIC PLC'leri programlamada kullanılan ve Ladder diyagram işlemlerini yerine getiren makrolar tanıtılmaktadır. Bu makrolar PIC assembly dilinde yazılmış olup detaylı açıklama için okuyucu [2-27] referanslara başvurabilir.

3.5.1. Kontak ve Röle Temelli Fonksiyonlar

Bu bölümde olan kontak ve röle temelli fonksiyonlar anlatılmaktadır. Kontak ve röle temelli makrolar bir bitlik (ikili-binary) değişkenler için tanımlanmıştır. Bu tanımlamalara ait fonksiyonlar Çizelge 3.1. ve Çizelge 3.2.'de görülmektedir.

Çizelge 3. 1. Kontak ve Röle temelli makrolar[2]

Merdiven Diyagramı	Makro Adı	İşlem
	LD (LOAD)	Ladder gösterimde bu makro normalde açık bir kontak ile ifade edilir. Giriş değeri 0 ise, çıkış değeri 0 yapmaktadır; eğer giriş değeri 1 ise, çıkış değeri 1 yapmaktadır.
	LD_NOT (LOAD_NOT)	Ladder gösterimde bu makro normalde kapalı bir kontak ile ifade edilir. Giriş değeri 0 ise, çıkış 1 yapmaktadır; eğer giriş değeri 1 ise, çıkış 0 yapılmaktadır.
	NOT	Ladder gösterimde bu makro lojik DEĞİL (NOT) kapısı olarak kullanılır. Alınan giriş değeri 0 ise, çıkış 1 yapılır; eğer giriş değeri 1 ise, çıkış 0 yapılır.
	OUT	Bu makro girişlerinden röle çıkışı olarak ifade edilir. Alınan giriş değeri 0 ise, çıkış değeri 0 yapmaktadır; eğer giriş değeri 1 ise, çıkış değeri 1 yapmaktadır.
	OUT_NOT	Bu makro girişlerinden DEĞİL'lenmiş röle çıkışı olarak ifade edilir. Alınan giriş değeri 0 ise, çıkış 1 yapmaktadır; eğer giriş değeri 1 ise, çıkış değeri 0 yapmaktadır.
	SET	Giriş değişkeni 0 ise hiçbir işlem yapılmaz ve çıkışın konumunda bir değişiklik meydana gelmez. Giriş değişkeni 1 ise çıkış değeri set edilir (1 yapılır).
	RESET	Giriş değişkeni 0 ise hiçbir işlem yapılmaz ve çıkış konumunda bir değişiklik meydana gelmez. Giriş değişkeni 1 ise çıkış değeri reset edilir (0 yapılır).

Çizelge 3. 2. Temel lojik işlem makroları [2]

Makro Adı	İşlem
OR	Ladder gösterimde bu makro iki girişli bir VEYA (OR) kapısı olarak görev yapar.
OR_NOT	Ladder gösterimde bu makro girişlerinden birisinin DEĞİL'i alınmış iki girişli bir VEYA (OR) kapısı olarak görev yapar.
NOR	Bu makro iki girişli bir VEYA DEĞİL (NOR) kapısı olarak görev yapar.
AND	Bu makro iki girişli bir VE(AND) kapısı olarak görev yapar
AND_NOT	Bu makro girişlerinden birisinin DEĞİL'i alınmış iki girişli bir VE (AND) kapısı olarak görev yapar.
NAND	Bu makro iki girişli bir VE DEĞİL (NAND) kapısı olarak görev yapar.

3.6. Zamanlayıcı Fonksiyonları

Belirli bir zaman gecikmesi yapılmasının gerekli olduğu uygulamalarda kullanılan fonksiyonlardır. Bu fonksiyonlar dörde ayrılmaktadırlar. Bunlar düz zaman rölesi, ters zaman rölesi, puls zaman rölesi ve osilatör zaman rölesidir. Zamanlayıcı fonksiyonları; giriş sinyaline bağlı olarak ayarlanan sürede zaman gecikmesi oluşturulup, bu süre sonunda çıkış sinyali üreten fonksiyonlardır. PIC-PLC’lerde zamanlayıcıların çalışması için referans zamanlama sinyallerine ihtiyaç vardır. Referans zamanlama sinyalleri 20 MHz osilatör için Çizelge 3.3.’te sunulmuştur.

Çizelge 3. 3. Referans zamanlama frekansları [2]

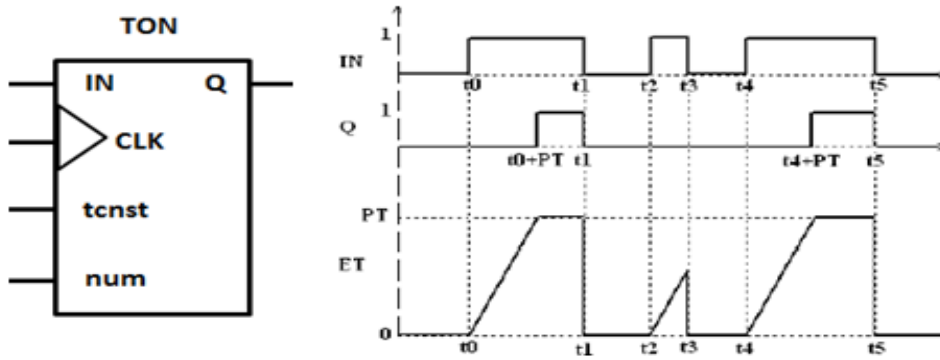
Referans CLK Konağı	Süre (ms)
T0.0	0,1024
T0.1	0,2048
T0.2	0,4096
T0.3	0,8192
T0.4	1,6384
T0.5	3,2768
T0.6	6,5536
T0.7	13,1072
T1.0	26,2144
T1.1	52,4288
T1.2	104,8576
T1.3	209,7152
T1.4	419,4304
T1.5	838,8608
T1.6	1677,7216
T1.7	3355,4432

3.6.1. Düz Zaman Rölesi (TON)

Giriş sinyali (IN), kurma süresi kadar (PT) bir süre uygulandıktan sonra çıkış sinyali gecikmeli olarak lojik 0 seviyesinden lojik 1 seviyesine konum değiştiren zaman rölesine düz zaman rölesi (on delay timer) adı verilir. Şekil 3.8.’de PIC-PLC için oluşturulan düz zaman rölesi makrosuna ait sembol ve zaman diyagramı görülmektedir. Kurma süresi (PT) bir referans zamanlama sinyali “CLK=t_reg,t_bit” ve bir zaman sabiti “tcnst” yardımıyla belirlenir. Kurma süresine ait hesaplama Eşitlik 3.1.’de

görülmektedir. Zamanlayıcı makrosunda bir adet giriş sinyali (IN) ile bir adet çıkış sinyali (Q) bulunmaktadır. CLK Boole giriş değişkeni olarak tanımlanmış olan, zaman gecikmesi için kullanılacak ve süreleri ms olarak ifade edilen T0.0, ..., T1.7 referans zamanlama sinyallerini ifade etmektedir. Tcnst 1-255 arası bit katsayı olup PT'yi elde etmek için kullanılır. PIC-PLC de 8 adet Düz Zaman Rölesi mevcuttur. Bunlar num (0..7) değişkeni ile seçilir [2].

$$PT = tcnst * CLK \quad (3.1)$$



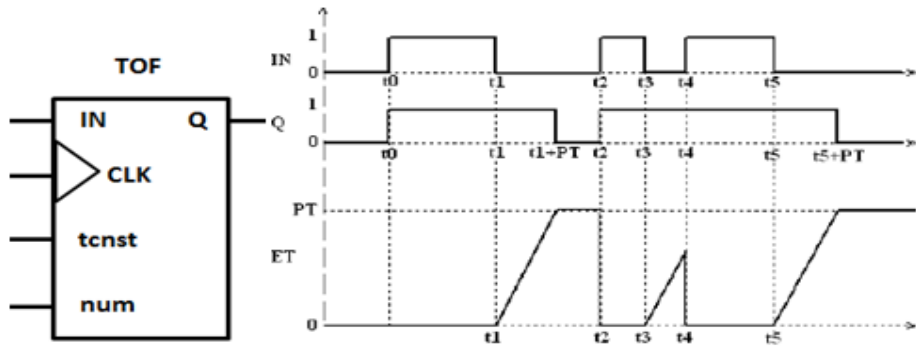
Şekil 3. 8. Düz zaman rölesinin sembolü ve zaman diyagramı [2]

Düz Zaman Rölesine ait PIC Assembly makro kodu örneği aşağıda görülmektedir.

```
TON_8 3, T1.1, .50
```

3.6.2. Ters Zaman Rölesi (TOF)

Çıkış sinyalini (Q) lojik 0 yapma işlemini giriş sinyalinin (IN) lojik 1 seviyesinden lojik 0 seviyesine düşmesinden sonra belirli bir süre (PT) geciktirerek gerçekleştiren zaman rölelerine ters zaman rölesi (off delay timer) denir. Şekil 3.9.'da PIC-PLC için oluşturulan ters zaman rölesi makrosuna ait sembol ve zaman diyagramı görülmektedir.



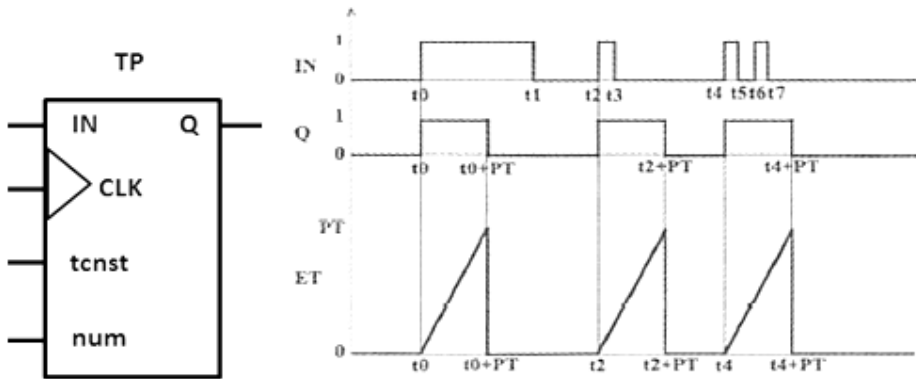
Şekil 3. 9. Ters zaman rölesinin sembolü ve zaman diyagramı [2]

Ters Zaman Rölesine ait PIC Assembly makro kodu örneği aşağıda görülmektedir.

`TOF_8 3, T1.1, .50`

3.6.3. Puls Zaman Rölesi (TP)

Puls zaman rölesi (pulse timer -TP) belirli bir zaman aralığında (PT) puls'lar üretmek için kullanılan bir zaman rölesidir. Şekil 3.10.'da PIC-PLC için oluşturulan puls zaman makrosuna ait sembol ve zaman diyagramı görülmektedir.



Şekil 3. 10. Puls zaman rölesinin sembolü ve zaman diyagramı [2]

Puls Zaman Rölesine ait PIC Assembly makro kodu örneği aşağıda görülmektedir.

`TP_8 3, T1.1, .50`

3.6.4. Osilatör Zaman Rölesi

Giriş sinyali (IN) lojik 1 olduğu sürece çıkış sinyalini (Q) sürekli olarak sırasıyla ayarlanan 0 süresi (PT0) kadar lojik 0 (OFF) ve ayarlanan 1 süresi (PT1) kadar da lojik 1 (ON) yapan zaman rölesine osilatör zaman rölesi (oscillator timer) adı verilir.

Osilatör Zaman Rölesine ait PIC Assembly makro kodu örneği aşağıda görülmektedir.

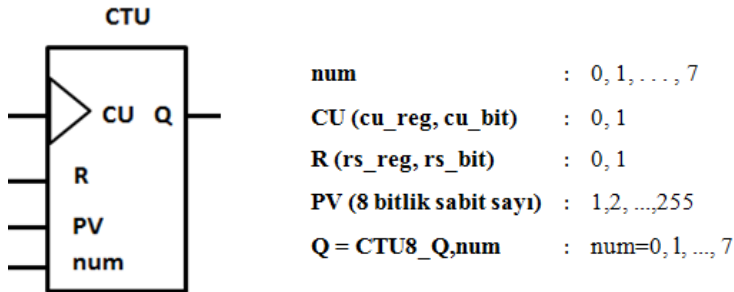
```
TOS_8 3, T1.1, .50, .50
```

3.7. Sayıcı Fonksiyonları

Sayıcılar belirli bir işlemin sayısına göre sinyal üreten fonksiyonlardır. Genelde PLC'lerde tanımlı ileri sayıcı (up counter - CTU), geri sayıcı (down counter - CTD) ve ileri/geri sayıcı (up-down counter - CTUD) isimli üç temel sayıcı fonksiyonu mevcuttur.

3.7.1. İleri Sayıcı (CTU)

İleri sayıcı (up counter-CTU) bir sayma değerinin maksimum değere ulaştığını göstermek için kullanılan sayıcı türüdür. İleri sayma girişi **CU**, sayıcı çıkışı **Q** ve reset girişi **R**, Boole değişkeni olarak tanımlanmıştır. Kurma değeri **PV**, 1-255 arası sabit bir sayıdır ve her bir sayıcı için maksimum sayma değerini belirlemek için kullanılır. **num** girişine, 0-7 arası sabit sayı atanarak 8 farklı ileri sayıcı tanımlanabilmektedir. Sayıcı çıkışları, CTU8_Q, num (num=0,1,2,...,7), yani CTU8_Q0, CTU8_Q1,..., CTU8_Q7 şeklindedir. Şekil 3.11.'de İleri sayıcı (CTU) sembolü ve değişken açıklamaları görülmektedir [2].



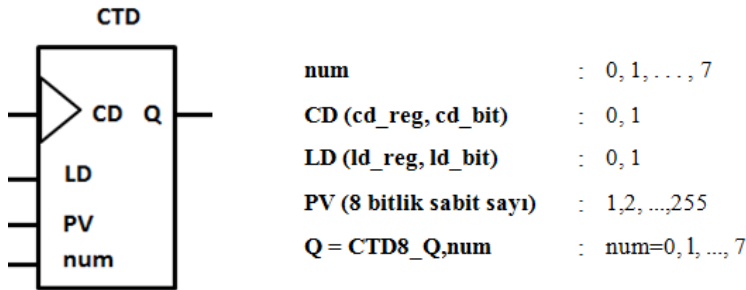
Şekil 3. 11. İleri sayıcı (CTU) sembol görünümü ve sembol açıklamaları [2]

İleri Sayıcıya ait PIC Assembly makro kodu örneği aşağıda görülmektedir.

```
CTU_8 0, I0.0, I0.1, .15
```

3.7.2. Geri Sayıcı (CTD)

Geri sayıcı (down counter - CTD) bir sayma değerinin minimum değere ulaştığını göstermek için kullanılan sayıcı türüdür. Geri sayma girişi **CD**, sayıcı çıkışı **Q** ve yükleme girişi **LD**, Boole değişkeni olarak tanımlanmıştır. Kurma değeri **PV**, 1-255 arası sabit bir sayıdır ve her bir sayıcı için başlangıç sayma değerini belirlemek için kullanılır. **num** girişine, 0-7 arası sabit sayı atanarak 8 farklı geri sayıcı tanımlanmaktadır. Sayıcı çıkışları, CTD8_Q, num (num=0,1,2,...,7), yani CTD8_Q0, CTD8_Q1,..., CTD8_Q7 gibi. Şekil 3.12.'de Geri sayıcı (CTD) sembolü ve değişken açıklamaları görülmektedir [2].



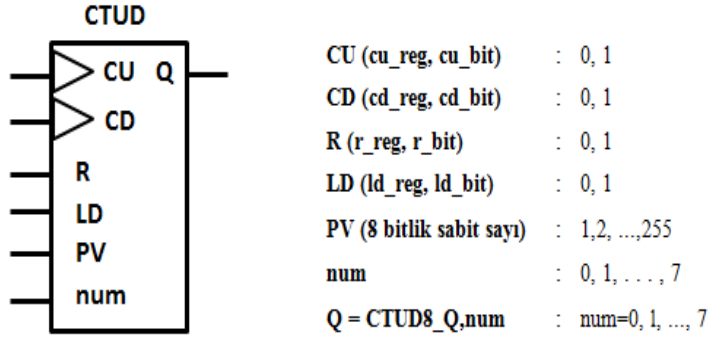
Şekil 3. 12. Geri sayıcı (CTD) sembol görünümü ve sembol açıklamaları [2]

Geri Sayıcıya ait PIC Assembly makro kodu örneği aşağıda görülmektedir.

```
CTD_8 0, I0.0, I0.1, .15
```

3.7.3. İleri / Geri Sayıcı (CTUD)

İleri/geri sayıcıların (up/down counter - CTUD), ileri sayma için kullanılan **CU** ve geri sayma için kullanılan **CD** olmak üzere iki sayma girişi vardır. İleri/geri sayıcı, **CU** girişinde tespit edilen yükselen kenarlarda ileri sayar, **CD** girişinde tespit edilen Yükselen kenarlarda ise geri sayar. **PV** sayıcının saymaya başlayacağı sayı değerini belirler. Sayıcı **PV** değerine ulaştığında sayıcı çıkışı Q set (1) yapılır ve sayma işlemi durur. Şekil 3.13.'te İleri/Geri sayıcı (CTUD) sembolü ve değişken açıklamaları görülmektedir [2].



Şekil 3. 13. İleri/Geri sayıcı (CTUD) sembol görünümü ve sembol açıklamaları [2]

İleri/Geri Sayıcıya ait PIC Assembly makro kodu örneği aşağıda görülmektedir.
CTUD_8 2, I0.0, I0.1, I0.2, I0.3, .15

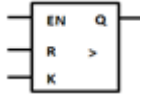
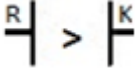

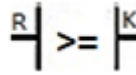
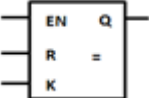
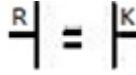
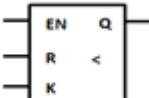
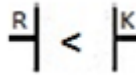

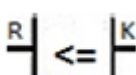
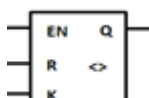
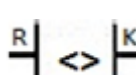
3.8. Karşılaştırma Fonksiyonları

Karşılaştırma işlemi PLC'lerde sıklıkla kullanılan bir fonksiyondur. Sayısal verilerin karşılaştırılması **R1** ve **R2** olarak tanımlanan iki kaydedici içeriğinin karşılaştırılması ya da bir kaydediciyle **R** sabit bir sayı **K**'nın karşılaştırılması şeklinde olur. Karşılaştırma fonksiyonları kendi içinde altı bölüme ayrılmaktadır. Bu fonksiyonlar Çizelge 3.4.'te ve 3.5.'te görülmektedir.

Çizelge 3. 4. Karşılaştırma fonksiyonları [2]

GT	A, B'den büyük	(A > B)
GE	A, B'den büyük veya eşit	(A >= B)
EQ	A, B'ye eşit	(A = B)
LT	A, B'den küçük	(A < B)
LE	A, B'den küçük veya eşit	(A <= B)
NE	A, B'ye eşit değil	(A <> B)

Çizelge 3. 5. Karşılaştırma fonksiyonlarına ait semboller [2]

İşlem	Şematik Sembol	Merdiven Diyagramı Sembolü	Örnek Makro Kodu
GT Büyük ">"			R_GT_K I1,0Fh
GE Büyük ve Eşit ">="			R_GE_K I1,0Fh
EQ Eşit "="			R_EQ_K I1,0Fh
LT Küçük "<"			R_LT_K I1,0Fh
LE Küçük ve Eşit "<="			R_LE_K I1,0Fh
NE Eşit Değil "<>"			R_NE_K I1,0Fh

3.9. Aritmetik Fonksiyonlar

Aritmetik fonksiyonlar bir veya daha fazla değeri kullanarak ilgili işlemi gerçekleştirir ve sonucu hafızaya kaydederler. Hemen hemen tüm PLC'lerde toplama, çıkarma gibi bazı aritmetik işlemler mevcuttur. PIC-PLC'lerde dört adet aritmetik fonksiyon bulunur. Bunlar;

- add (değer1, değer2, hedef çıkış) – İki değeri topla, sonucu çıkışa kaydet
- sub (değer1, değer2, hedef çıkış) – İkinci değeri birinciden çıkar, sonucu çıkışa kaydet
- inc (değer, hedef çıkış) – Değeri bir artır, sonucu çıkışa kaydet
- dec (değer, hedef çıkış) – Değeri bir azalt, sonucu çıkışa kaydet

Bu fonksiyonlara ait şematik semboller ve açıklamaları Çizelge 3.6.'da görülmektedir.

Çizelge 3. 6. Aritmetik işlemler fonksiyonları [2]

İşlem	Şematik Sembol	Merdiven Diyagramı Sembolü	Örnek Makro Kodu
ADD (kaynak değeri 1, kaynak değeri 2, hedef) - iki kaynak değerinin topla ve sonucu hedefe kaydet.			RaddK Q1, .2, Q1
SUB (kaynak değeri 1, kaynak değeri 2, hedef) - ikinci kaynak değerinin birinciden çıkar ve sonucu hedefe kaydet.			RsubK Q1, .3, Q1
INC (kaynak değeri, hedef) - kaynak değeri bir arttır ve sonucu hedefe kaydet.			incR Q1, Q1
DEC (kaynak değeri, hedef) - kaynak değeri bir azalt ve sonucu hedefe kaydet.			decR Q1, Q1

3.10. Flip-Flop Makroları

Flip-floplar bir veya iki giriş işareti ve saat darbesi ile kontrol edilirler. Saat darbesinin yükselen kenarı veya düşen kenarı ile tetiklenen flip-floplar her saat darbesinde bir sonraki duruma geçerler. PIC-PLC'lerde kullanılan kenar algılayıcı ve flip-flop makroları Çizelge 3.7.'de görülmektedir.

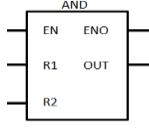
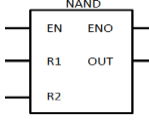
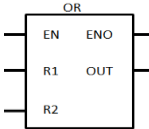
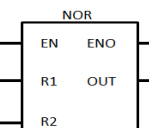
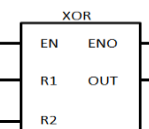
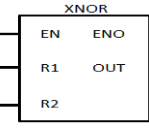
Çizelge 3. 7. Flip-Flop fonksiyonları [2]

İşlem	Şematik Sembol	Örnek Makro Kodu
r_edge: Bu fonksiyon yükselen kenar algılayıcı olarak işlem yapmaktadır.		r_edge 0
f_edge: Bu fonksiyon düşen kenar algılayıcı olarak işlem yapmaktadır.		f_edge 0
latch1: Latch1 makrosu aktif 1 yetkilenme girişli bir D mandalı fonksiyonunu tanımlamaktadır.		latch1 I0.1, Q0.4
latch0: Latch0 makrosu aktif 0 yetkilenme girişli bir D mandalı fonksiyonunu tanımlamaktadır.		latch0 I0.1, Q0.4
dff_r: Yükselen Kenar Tetiklemeli D Flip-flop Makrosu		dff_r 0, I1.1, Q1.4
dff_f: Düşen Kenar Tetiklemeli D Flip-flop Makrosu		dff_f 0, I1.5, Q1.3
tff_r: Yükselen Kenar Tetiklemeli T Flip-flop Makrosu		tff_r 0, I1.1, Q1.4
tff_f: Düşen Kenar Tetiklemeli T Flip-flop Makrosu		tff_f 0, I1.5, Q1.3

3.11. Lojik Fonksiyonlar

Bu bölümde AND, NAND, OR, NOR fonksiyonları anlatılmaktadır. Bu makrolar 8 bitlik bir A kaydedicisinin içeriği ile bir başka 8 bitlik B kaydedicisinin içeriğine lojik işlem uygulamaktadır. Çizelge 3.8.'de lojik fonksiyonlara ait makrolar görülmektedir.

Çizelge 3. 8. Lojik fonksiyonlar [2]

İşlem	Şematik Sembol	Örnek Makro Kodu
AND : Giriş değişkenlerinin her hangi birinin 0 değerini alması ile çıkış 0 değerini alırken, girişlerin hepsinin 1 olması durumunda çıkışta 1 değerini alır.		R1andR2 I1, 50h, Q1
NAND : Bu kapıda girişlerin her hangi birinin '0' olması durumunda çıkış '1' olmaktadır.		R1nandR2 I1, 50h, Q1
OR : Girilen iki ya da daha çok değer toplanmasına yarayan kapıdır. Herhangi bir girişi 1 olduğunda 1 cevabi verir, ancak tüm girişleri 0 ise 0 verir.		R1orR2 I1, 50h, Q1
NOR : Girilen iki ya da daha çok değer toplanmasını sağlayan ve sonucun tersini alan kapıdır. Sadece tüm girişleri 0 ise 1 verir, diğer tüm hallerde 0 verir		R1norR2 I1, 50h, Q1
XOR : Girişindeki işaretler birbirinden farklı olduğu zaman çıkış olarak 1 verir, diğer tüm hallerde 0 verir.		R1xorR2 I1, 50h, Q1
XNOR : Sadece bir girişi 1 olduğunda 0 cevabı verir; diğer tüm durumlarda ise 1 cevabını verir.		R1xnorR2 I1, 50h, Q1

3.12. Diğer Makrolar

PIC-PLC'ler için yukarıda açıklanan makrolar haricinde çeşitli makrolar bulunmaktadır. Bu makrolar **Kayıdırma ve Döndürme Makroları, Veri Seçici Makroları, Veri Dağıtıcı Makroları, Kod Çözücü Makroları ve Öncelik Kodlayıcı Makrolarıdır** [2]. Bu makroların kısaca işlevleri Çizelge 3.9.'da açıklanmaktadır.

Çizelge 3. 9. Diğer makrolar [2]

Makrolar	Açıklamalar
Kaydırma ve Döndürme Makroları	Bir kaydedici içerisindeki bitleri sağa ya da sola hareket ettirmek için kullanılan makrodur.
Veri Seçici Makroları	Bir çoklayıcı birden fazla girişten tek bir çıkışı sağlayan bir anahtar işlevi görür. Çoklayıcıya bağlanan seçici, girdilerden yalnızca bir tanesini seçerek çıkışa yönlendirir. Çoklayıcı ikizkenar yamuk olarak gösterilir; uzun olan paralel kenarı giriş iğnesi, kısa olan paralel kenarın ise çıkış iğnesini bulundurmaktadır.
Veri Dağıtıcı Makroları	Bir devrenin birden fazla aygıtta sinyal gönderilmesi gerektiğinde kullanılan makrodur.
Kod Çözücü Makroları	Bir kodu farklı sinyale dönüştüren bir devredir. Kod çözücünün m-bitlik veriyi alıp bu verinin kodunu 2^m çıkış hattını kullanarak çözen bir hat kod çözücüsüdür.
Öncelikli Kodlayıcı	Kodlayıcı bir grup sinyali bir koda çeviren devre olarak ifade edilir. Standart devre olarak 2^n giriş verisinin n bitlik koda çeviren kodlayıcı, kod çözücünün tersi işlev yapmaktadır.

4. BULGULAR VE TARTIŞMA

4.1. Mikrodenetleyici Temelli Plcer İin Programlama Yazılımı

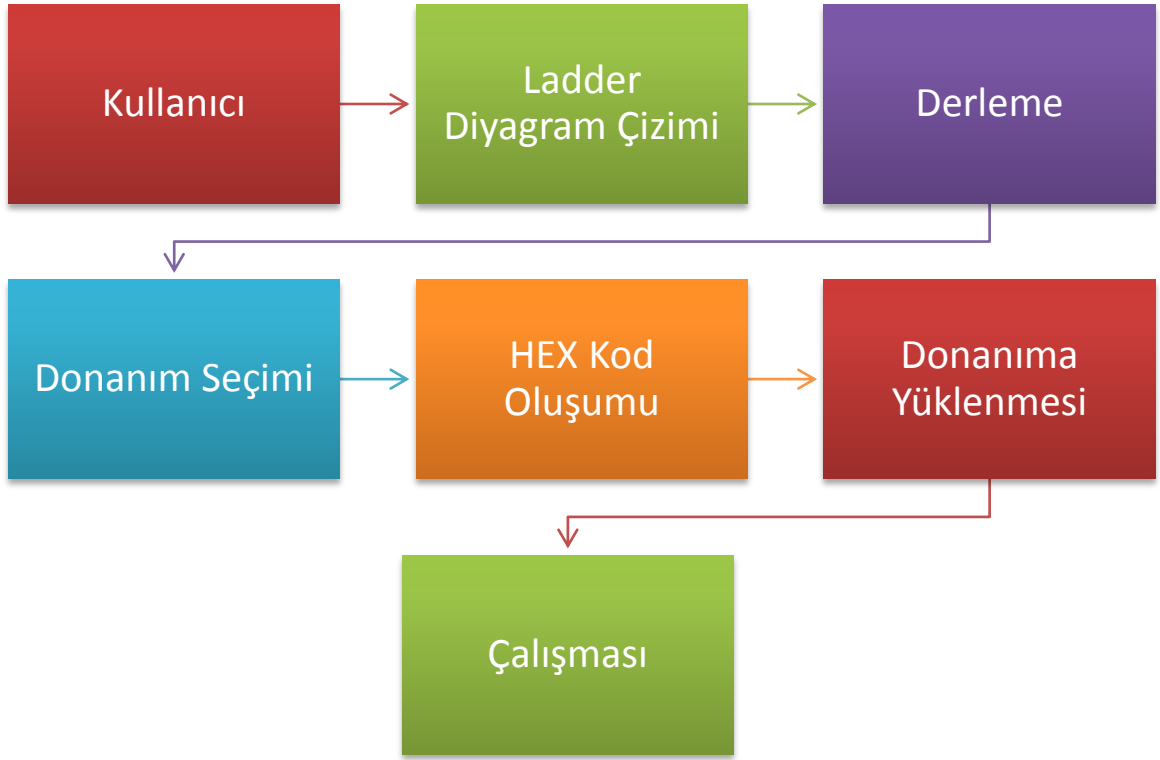
Bu yüksek lisans tezi kapsamında bir önceki bölümde incelenen PIC-PLC donanımları için geliştirilen programlama yazılımı bu bölümde açıklanmaktadır. Geliştirilen yazılımda mikrodenetleyici temelli PLC'ler için kullanılan Ladder diyagramı programlamaya olanak sağlayan ve Ladder programını ilgili PIC assembly koduna çeviren görsel bir yazılım oluşturulmuştur.

Yazılım Microsoft Visual Studio 2012 ortamı kullanılarak geliştirilmiştir. Yazılımın web tabanlı olması tercih edildiğinden ASP.NET ve C# kullanılmıştır. ASP.NET uygulama geliştirme aracının en önemli özelliği uygulamaların doğrudan doğruya web formları üzerinde geliştirilip başka bir işleme gerek kalmaksızın yine internet gezgini üzerinde çalıştırabiliyor olmasıdır [45].

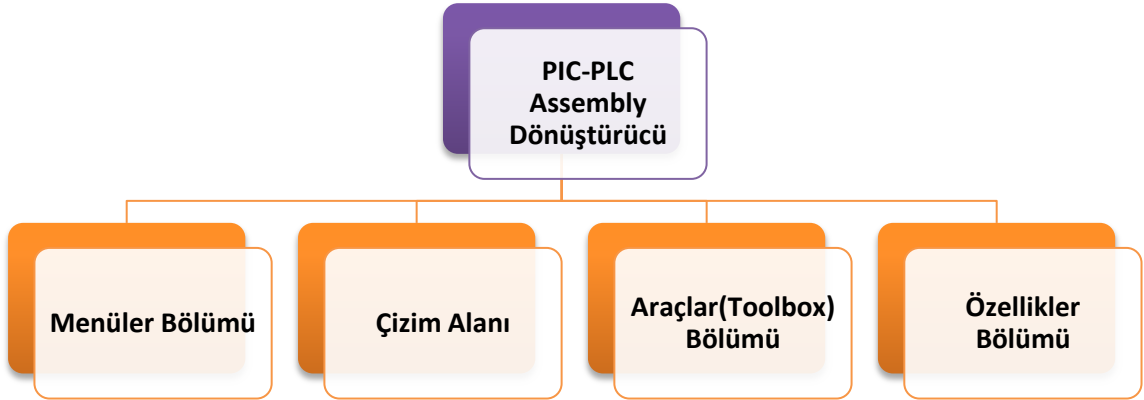
Geliştirilen program temelde görsel olarak çizilen Ladder diyagramlarını PIC assembly koduna dönüştürmektedir. Bu işlemin ardından elde edilen assembly kodu PIC'ler için üreticisi tarafından ücretsiz sunulan MPASM derleyicisi ile makine kodlarına çevirmektedir. Çevrilen kodlar herhangi bir PIC programlayıcısıyla ilgili donanıma yüklenerek kullanıcıların PLC donanımını rahatlıkla programlamasına olanak sağlamaktadır. Açıklanan bu işlem akışı Şekil 4.1.'de genel olarak özetlemektedir.

4.2. Geliştirilen Yazılımın Bölümleri

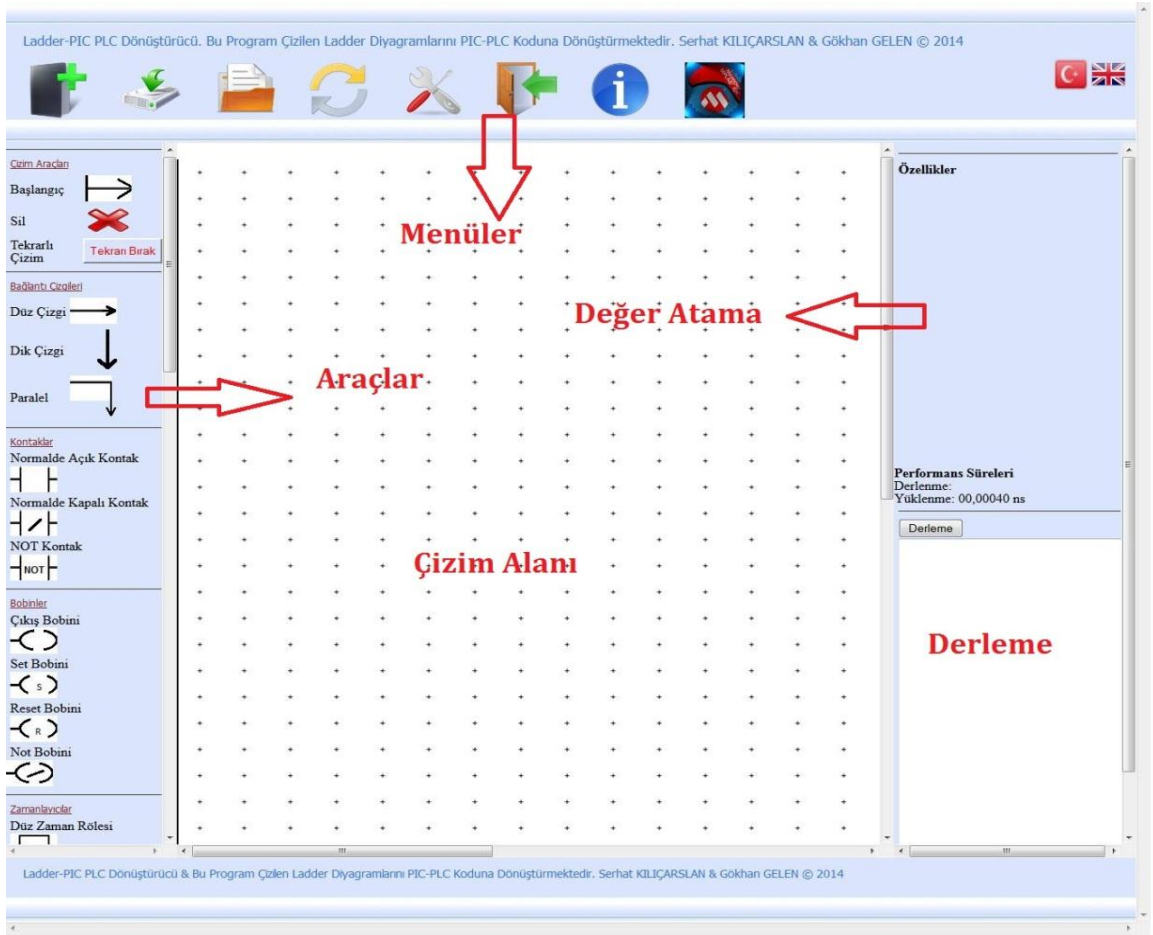
Geliştirilmiş olan yazılım Şekil 4.2.'de görüldüğü üzere birkaç alt bölümden oluşmaktadır. Bu bölümlerdeki menüler aracılığıyla program sağlıklı bir şekilde kullanılabilir. Yazılım dört ayrı bölümden meydana gelmektedir. Bunlardan menüler bölümünde yapılan programda yeni çizim alanı açma, derleme, donanım seçme vb. işlemler yapılabilir. Çizim alanı olarak ifade edilen kısımda Ladder diyagramı çizme işlemi yerine getirilebilir. Araçlar bölümü ile çizim için gerekli olan Ladder makrolara ulaşabileceği ve Özellikler bölümü ile çizilen Ladder makrolara parametre ataması yapılabilir. Geliştirilen yazılıma ait ekran görüntüsü Şekil 4.3.'te görülmektedir.



Şekil 4. 1. Programın özet akış şeması



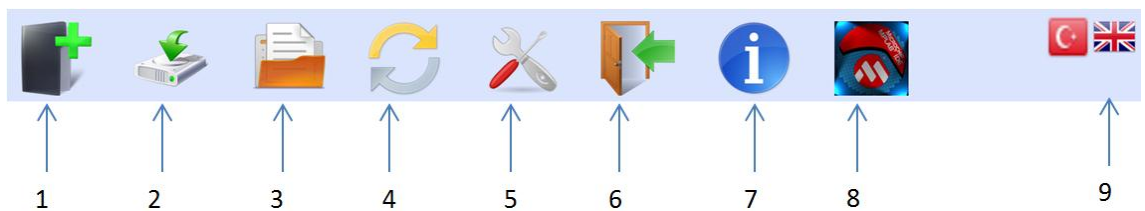
Şekil 4. 2. Yazılımın ana bileşenleri



Şekil 4. 3. Yazılımın ekran görüntüsü

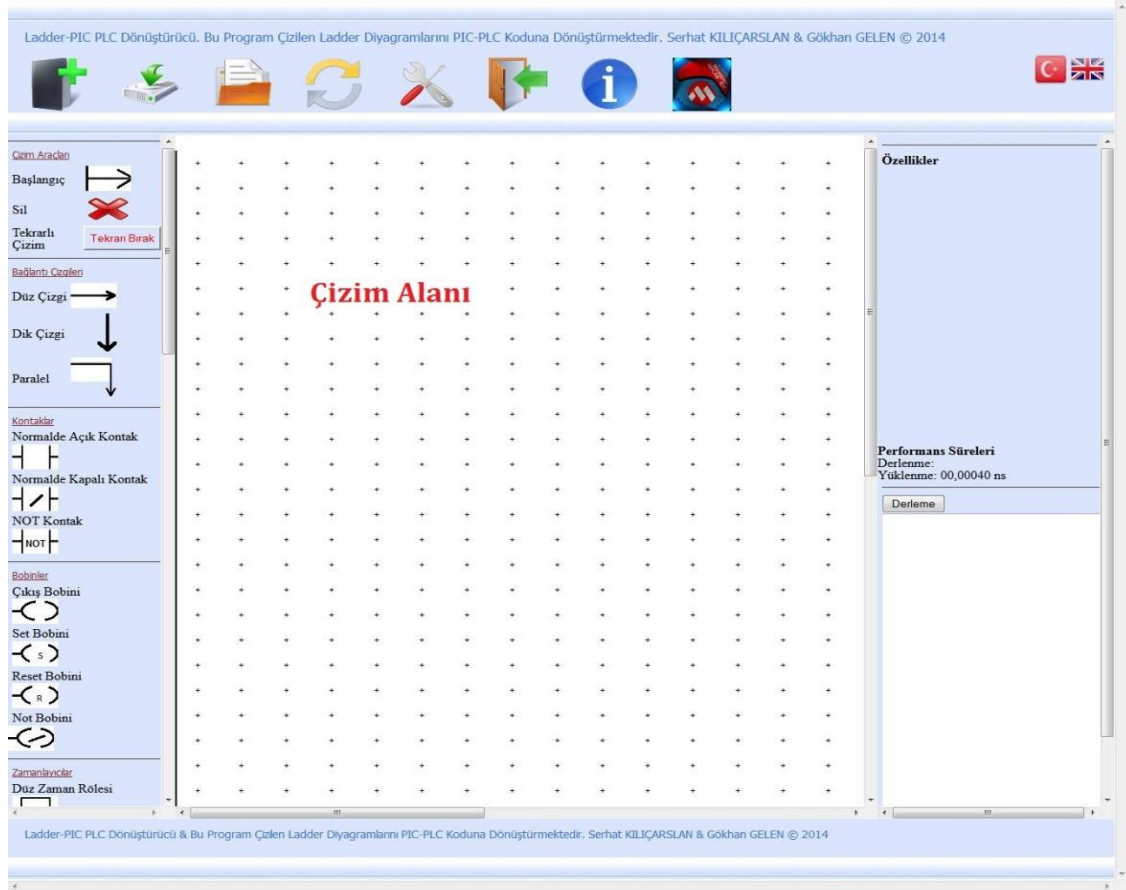
4.2.1. Menüler Bölümü

Bu bölümde geliştirilen “Ladder-PIC PLC dönüştürücü” yazılımının birinci kısım olan menüler bölümü tanıtılmaktadır. Bu bölümde; yeni çizim alanı oluşturma, yapılan çizimi kaydetme, çizimi açma, yapılan çizimin derlenmesi, hangi donanıma göre derlemenin yapılacağına seçilmesi ve yardım kısımlarına ulaşılan ikonlar yer almaktadır. Menü bölümünde yer alan ikonlar Şekil 4.4.’te görülmektedir. Her bir ikon aşağıda açıklanmaktadır.



Şekil 4. 4. Menü ikonları

Yeni Çizim Alanı Oluşturma (1. İkon): Yeni bir proje oluşturarak kullanıcıların Ladder diyagramları çizmesine olanak sağlayan ortamı hazırlamaktadır. Yeni proje oluşturduğunda çizim alanının belirgin olması için noktalar otomatik olarak eklenmekte ve kullanıcının rahatlıkla Ladder programı çizmesine olanak sağlanmaktadır. Yeni bir proje açılmış ekran görüntüsü Şekil 4.5.'de verilmiştir.

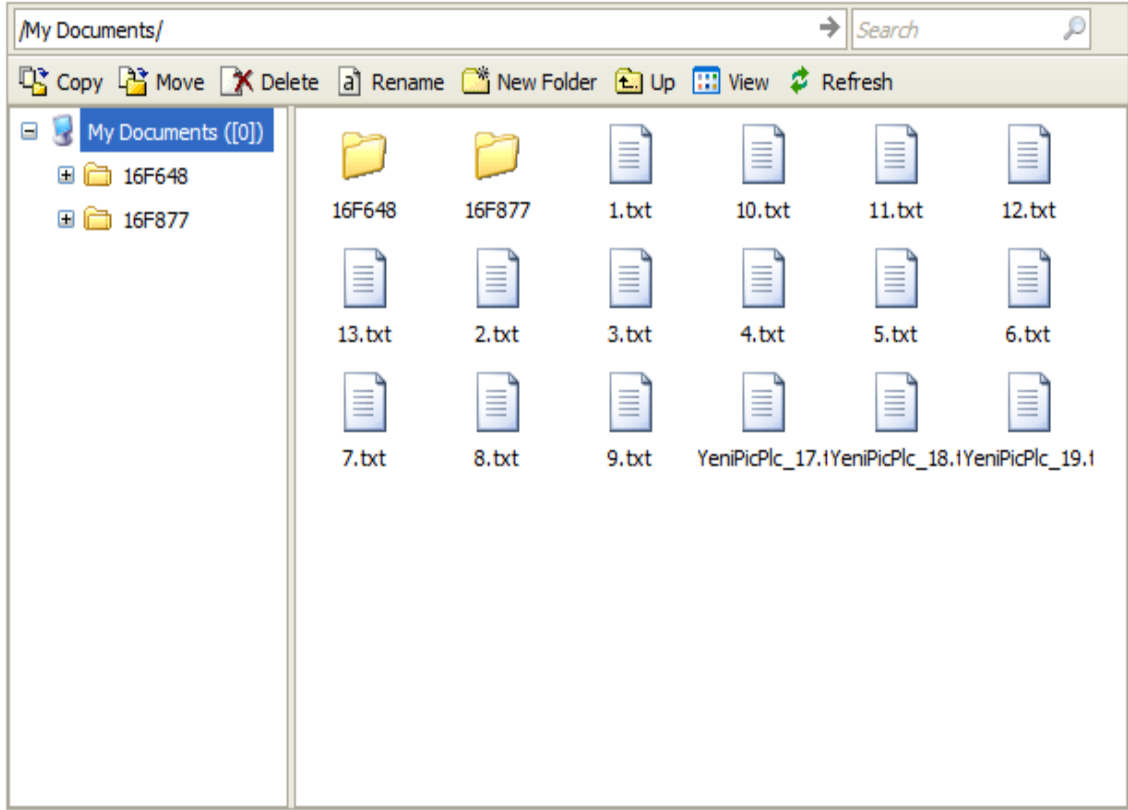


Şekil 4. 5. Yeni proje açılmış ekran

Çizimi Kaydetme (2. İkon): Yapılan projeyi program tarafından belirlenmiş olan klasöre kaydetmek için kullanılır. Bu alanı kullanırken yapılması gereken tek işlem, çizimi tamamladıktan sonra kaydetme butona tıklanmasıdır. Kullanıcı kaydetmiş olduğu projeleri sistem dosyaları altında **Kaydet** klasöründe bulabilmektedir.

Çizimi Açma (3. İkon): Kaydedilmiş olan çalışmaları kullanıcıların açarak üzerinde güncelleme yapmasına olanak tanınmaktadır. Yapılmış olan çalışmaları kullanabilmek için ilgili **Aç** simgesine tıklanarak Şekil 4.6.'daki açılan pencereden

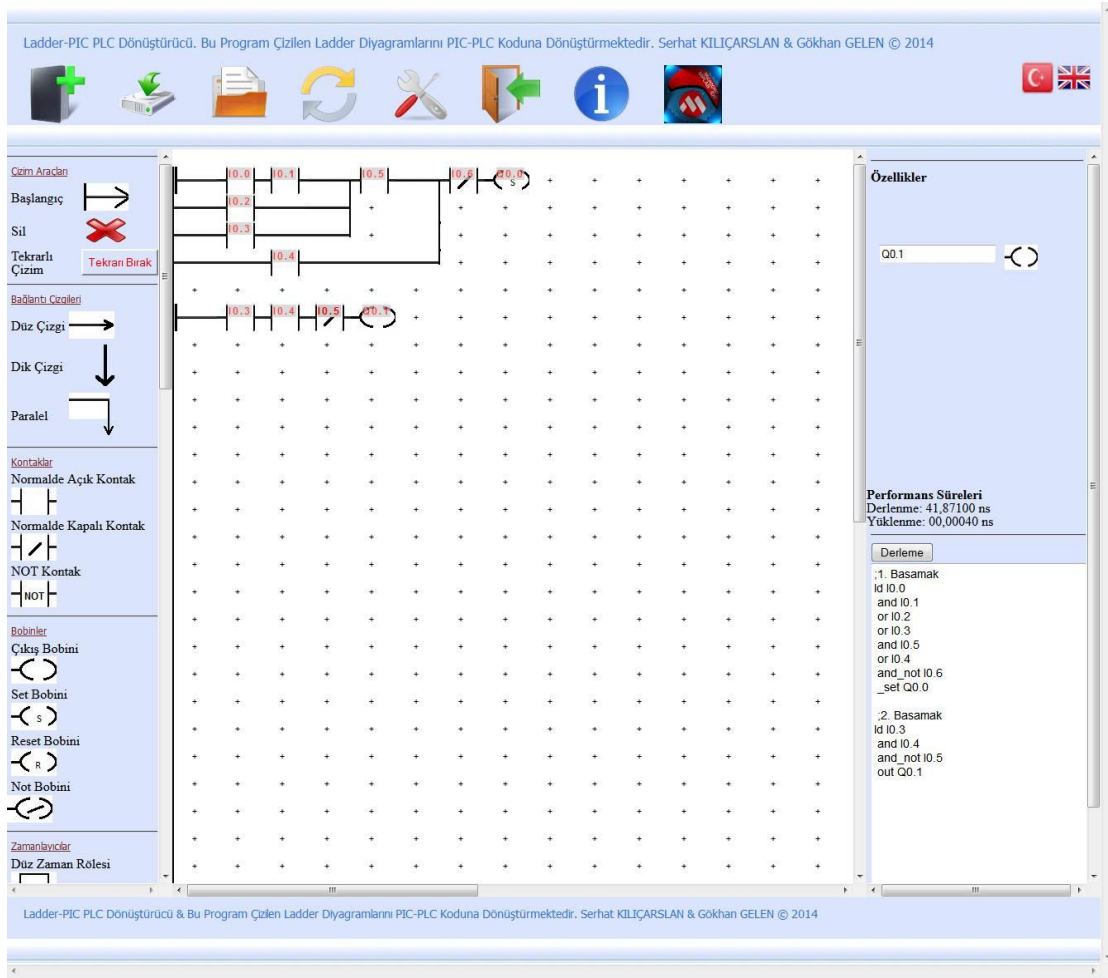
açmak veya güncellemek istenilen proje seçilir ve **Dosyayı Aç** butonuna tıklanarak istenen projenin çalışma alanında açılması sağlanır.



Dosyayı Aç

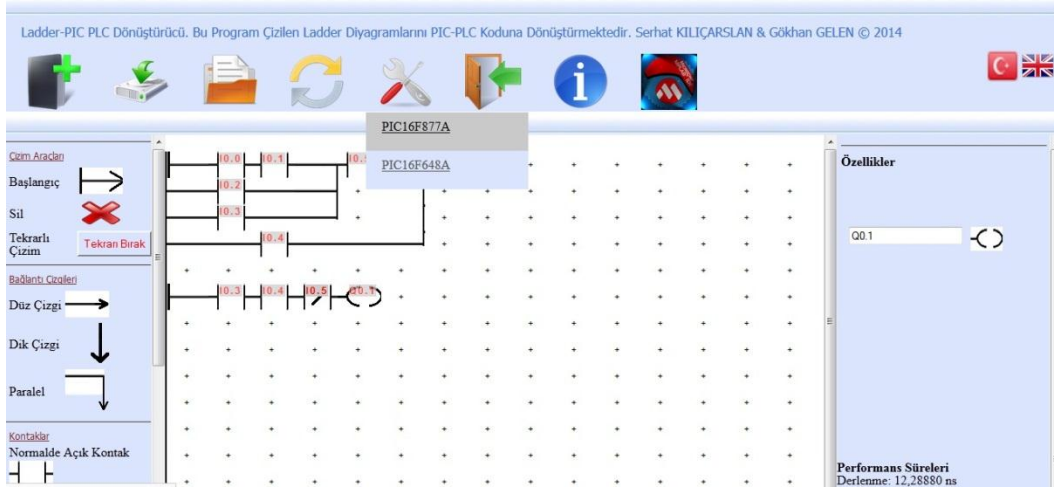
Şekil 4. 6. Dosya seçme alanı

Derleme (4. İkon): Bu menü altında çizilen Ladder programı PIC-PLC assembly koduna dönüştürmektedir. Örnek bir ladder diyagramının PIC-PLC Assembly koduna dönüşümü Şekil 4.7.'de görülmektedir. Kullanıcı **Derleme** alanında bulunan textbox içerisinden derlenmiş olan komutları tek tek görebilmektedir. Program, kullanıcıların kod üzerinde yapacağı değişiklikleri derleme butonuna ya da menüdeki derleme simgesine tıklayarak görebilmesine olanak sağlamaktadır.



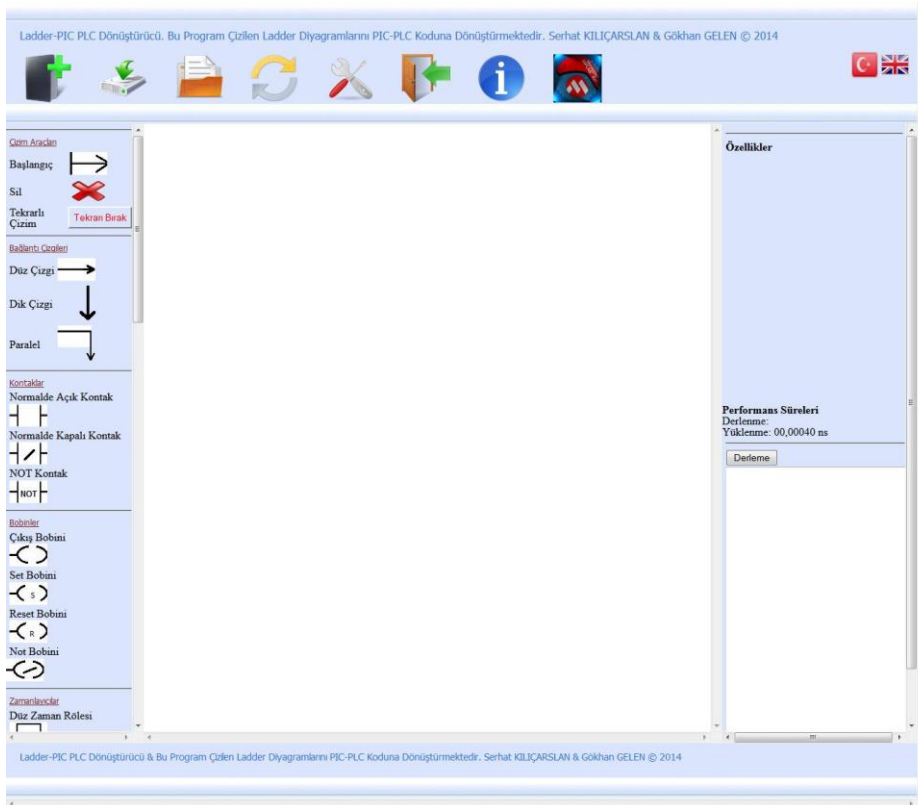
Şekil 4. 7. Derleme ekranı

Donanım Seçimi (5. İkon): Bu ikon kullanılarak çizim işlemi derlendikten sonra kullanıcının donanım seçmesi sağlanmaktadır. İkon altında PIC16F877A ve PIC16F648A yazıları seçilerek donanım seçme işlemi tamamlanmış olur. Derleme sonucu oluşturan PIC-PLC makroları seçilen donanım için tanımlanan alt klasöre kaydedilir. Seçilen donanıma göre üretilen *.asm dosyasını kullanıcı sekizinci (8) simgeye tıklayarak açılan MPASM programı aracılığıyla makine dilene dönüştürebilmektedir. Donanım seçimine ait işlem Şekil 4.8.'den görülebilir.



Şekil 4. 8. Donanım seçme ekranı

Ekranı Temizleme (6. İkon): Programın ilk açıldığı hale getirilebilmesi için kullanabilecek ikondur. Bu ikon ile ekran temizlenerek program, ilk açıldığı Şekil 4.9.'daki hale getirilebilir.

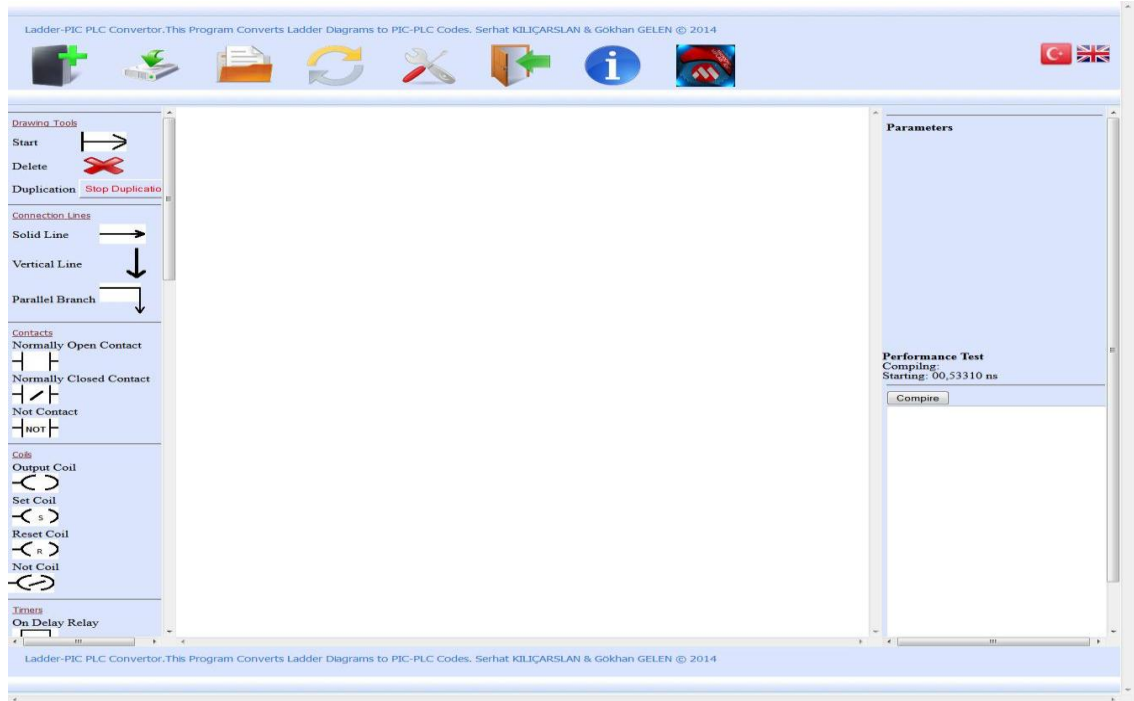


Şekil 4. 9. Program açılış ekranı

Yardım (7. İkon): Bu ikon tıklanarak kullanıcı geliştirilen programın kullanmasına yardımcı olması amacıyla açılan menü yardımıyla PDF ve Video dokümanına ulaşabilmektedir.

MPASM (8. İkon): Donanım kartına uygun oluşturulan *.asm dosyasının PIC-PLC'ye yüklenecek hex diline dönüşümünü gerçekleştirmek için kullanılacak kısımdır. İkon tıklanarak MPASM programı çalıştırılır.

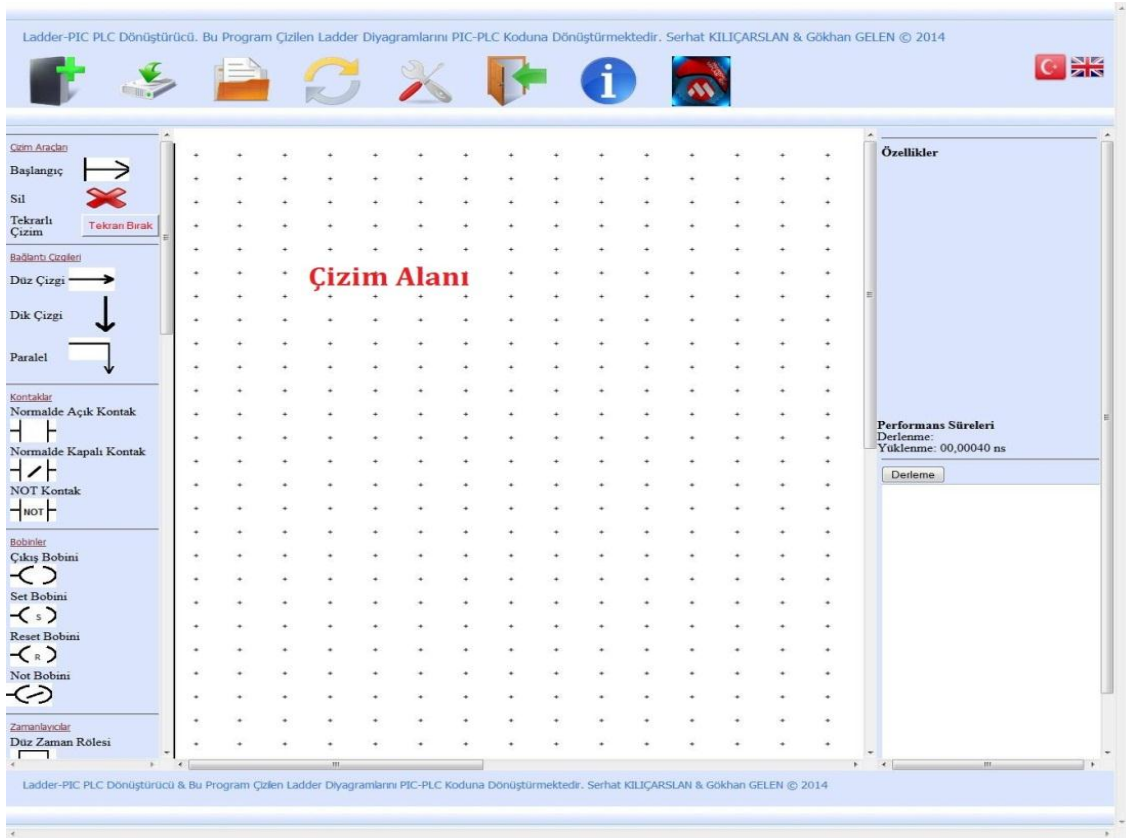
Dil Seçimi (9. İkon): Bu kısımda kullanıcıların Türkçe veya İngilizce dillerinden birini seçerek menü dilini ayarlanması sağlanır. Kullanıcı İngilizceyi seçtiğinde Şekil 4.10.'daki gibi programın İngilizce versiyonu görüntülenmektedir. Program çalıştırıldığında standart olarak Türkçe dilinde açılmaktadır.



Şekil 4. 10. Programın İngilizce menü versiyonu

4.2.2. Çizim Bölümü

Program üzerinde Ladder diyagramlarının çizebileceği bir alan oluşturulmuştur. Bu alan Şekil 4.11.'de görülmektedir. Çizim alanı .Net çatısının ASP.Net platformunda GridView yardımı ile oluşturulmuştur. Çizim için oluşturulan her bir nokta GridView'de bir satır ve sütunun kesiştiği alanı ifade etmektedir. Çizim alanında her bir nokta bir komponent'ı ifade etmektedir. Kullanıcı araçlar (Toolbox) bölümünden istediği bir komponent'ı alarak çizim işlemini gerçekleştirebilmektedir.



Şekil 4. 11. Çizim alanı

















4.2.3. Araçlar (Toolbox) Bölümü

Programın bu kısmında kullanıcıların Ladder diyagramı çizilebilmesi için gerekli olan araçlar bulunmaktadır. Kullanıcılar önceden tanımlanmış Ladder objeleri üzerine tıklayarak, seçilen objeyi çizim alanındaki noktalardan birine tıklamasıyla seçilen Ladder obje yerleştirilir. Çizilen Ladder diyagramlarının çalışabilmesi için her bir basamak için Şekil 4.12.'deki başlangıç noktası çizim alanında birinci sütuna eklenmesi gerekmektedir. Başlangıç noktası eklenmesi yapılmayan veya yanlış yapılan çalışmalarda derleme işlemi yapılmamaktadır. Çizim alanına ait komponent grupları ve açıklamaları Çizelge 4.1. ve Çizelge 4.2.'de gösterilmektedir.























Şekil 4. 12. Başlangıç noktası

Çizelge 4. 1. Ladder araçları(a)

	Araçlar Menü Sembolü	Makro Adı	Açıklama
Kontaklar		LD	Normalde Açık
		LD_NOT	Normalde Kapalı
		NOT	Değil Kontak
Bobinler		OUT	Çıkış
		_SET	Set Çıkış
		_RESET	Reset Çıkış
		OUT_NOT	Çıkış Not
Zamanlayıcılar		TON	Düz Zaman Rölesi
		TOF	Ters Zaman Rölesi
		TP	Puls Zaman Rölesi
		TOS	Osilatör Zaman Rölesi
Sayıcılar		CTU	İleri Sayıcı
		CTD	Geri Sayıcı
		CTUD	İleri/Geri Sayıcı
Çizgiler		Paralel	Paralel Bağlantı Çizgisi
		Seri	Seri Bağlantı Çizgisi

Çizelge 4. 2. Ladder araçları (b)

	Araçlar Menü Sembolü	Makro Adı	Açıklama
Flip-Flop Makroları		LATCH1	Aktif 1 "1"
		LATCH0	Aktif 0 "0"
		r_edge	Yükselen Kenar
		f_edge	Düşen Kenar
		dff_r	Yükselen Kenar Tetiklemeli D Flip-Flop
		dff_f	Düşen Kenar Tetiklemeli D Flip-Flop
		tff_r	Yükselen Kenar Tetiklemeli T Flip-Flop
		tff_f	Düşen Kenar Tetiklemeli T Flip-Flop
Taşıma Makroları		move_r	Taşıma Makrosu
		load_r	Yükleme Makrosu
Karşılaştırma Makroları		EQ	A=B
		GT	A>B
		GE	A>=B
		LT	A<B
		LE	A<=B
		NE	A<>B
Aritmetik Makrolar		ADD	Topla
		SUB	Çıkar
		INC	Bir Arttır
		DEC	Bir Azalt

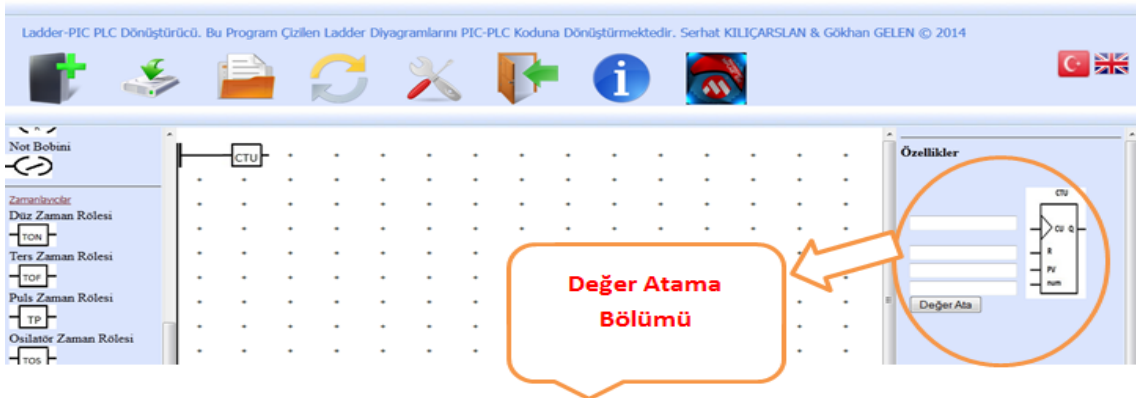
4.2.4. Özellikler Bölümü

Bu alanda çizim işleminde kullanılan komponentlerin değer atamaları gerçekleştirilebilmektedir. Çizim alanına eklenen objelere tıkladığında özellikler panelinde komponente ait resim ve parametrelerin atanabileceği metin kutuları kullanıcının karşısına Şekil 4.13.'de ve Şekil 4.14.'de görüldüğü gibi gelmektedir.

Bileşene Şekil 4.13.'teki gibi tek değerli bir değişken ataması yapılacaksa, değer giriş işleminden sonra **Enter** tuşuna basılarak değer ataması gerçekleştirilir. Eğer Şekil 4.14.'teki gibi birden fazla değer girilecekse ilgili resim karşısındaki alana değerler yazılır **Değer Ata** butonuna basılarak değer ataması yapılabilmektedir. Her komponentin değeri verildikten sonra çizim alanındaki sembol üzerinde görüntülenmektedir. Kullanıcılar, atadıkları değerleri Ladder komponentler üzerlerindeki yazılardan rahatlıkla takip edebilirler.

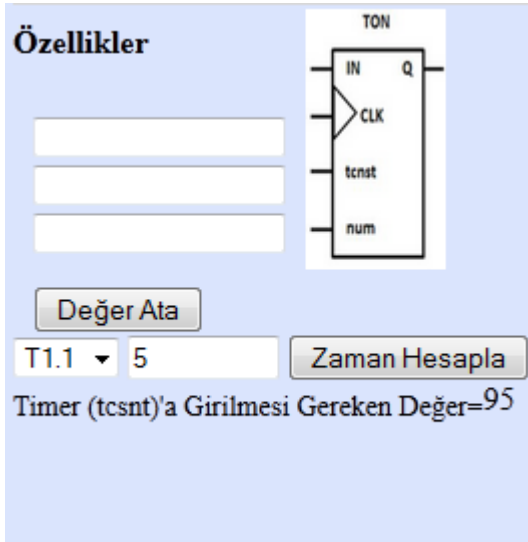


Şekil 4. 13. Tek değerli parametre atama ekranı



Şekil 4. 14. Çok değerli parametre atama ekranı

Özel işlem olarak zamanlayıcılar için PIC-PLC'lerde önceden tanımlı osilatör kontağı seçilerek zaman gecikme süresi ayarlanabilmektedir. Kullanıcıların kolay şekilde zamanlayıcı değerini girebilmesi için eklenen zamanlayıcının Ladder sembolü üzerine tıkladığında özellikler menüsünde zamanlayıcı için hesaplama alanı görüntülenir. Zaman ayarlaması yapılırken T0.0'dan T1.7'ye kadar önceden tanımlı olan osilatör kontaklarından biri seçilir. Örneğin 5sn'lik bir gecikme isteniyorsa, ilgili metin kutusu içerisine 5sn yazılır ve **Zaman Hesapla** butonuna basılarak **tcsnt**'ye girilecek gerçek değer bulunur. Zaman ayarlamasına ait hesaplama ekranı Şekil 4.15.'te görülmektedir. **Tcsnt** sabiti olarak 0-255 aralığında bir değer girilebilir. Eğer bulunan değer bu aralıkta değilse arayüz uyarı mesajı görüntülenmektedir. Bu durumda bir değer osilatör kontağı seçilerek hesaplama tekrarlanmalıdır.



Şekil 4. 15. Zaman hesaplama ekranı

4.2.5. Arayüz Kullanarak Ladder Diyagram Çizimi

Program menüler bölümünden yeni çalışma alanı açılmasıyla aktif hale gelmektedir. Çalışma alanı oluşturulduktan sonra ilk adım olarak başlangıç noktası Şekil 4.12.'deki gibi eklenir ve araçlar bölümünden hangi komponent kullanılacaksa o seçilerek başlangıç noktasından sonraki yere tıklanarak çizim alanına yerleştirilir. Program açıldığında standart olarak tekrarlı çizim modunda açılmaktadır. Yani seçilen objenin istenildiği kadar çizim alanına yeniden seçmeden eklenebilmesi sağlanır. Çizim işlemi gerçekleştikten sonra eklenen her bir komponent'e ait değer ataması

yapılabilmesi için **Tekrarı Bırak** butonuna basılır ve değer atama işlemi yapılmaya başlanır. Değer atamaları çizim alanındaki obje üzerine tıkladığında özellikler bölümünde açılan alandan yapılır. Atama işlemi gerçekleştirildikten sonra derleme butonuna basılarak çizilen Ladder diyagramına ait PIC-PLC Assembly makrolarını içeren temel dosya Şekil 4.16.'daki gibi oluşturulur.

```

list      p=16F877A                                ;işlemciyi tanımlamak için liste direktifi
#include <p16F877A.inc>                             ;işlemciye has değişken tanımları
#include <PICPLC_877_16I_160_tnmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_16I_160_kontak.inc>;kontakt ve röle temelli makrolar

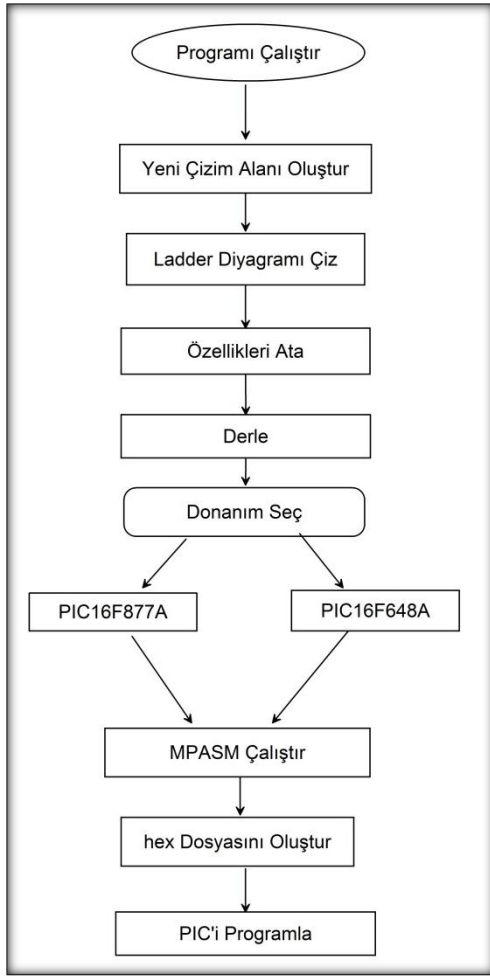
_CONFIG _CP_OFF & _WDT_ON & _BODEN_OFF & _PWRTE_ON & _HS_OSC & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

org      0x00                                ;Reset vektörü
main
        initialize
scan
        get_inputs
;----- kullanıcı programı burada başlar -----
;1.Basamak
ld I0.0
out Q0.0
;2.Basamak
ld_not I0.1
out Q0.1
;3.Basamak
ld I0.2
out_not Q0.2
;4.Basamak
ld I0.3
and I0.4
out Q0.3
;5.Basamak
ld I0.5
and_not I0.6
out Q0.4
;6.Basamak
ld I0.7
nand I1.0
out Q0.5
;----- kullanıcı programı burada biter -----
        send_outputs
goto    scan

```

Şekil 4. 16. PIC-PLC Assembly makroları dosyası

Gerçekleştirilen programda kullanıcıların derleme işlemini tamamladıktan sonra yapması gereken işlem, hangi PLC donanımı kullanılacaksa o donanım ayarlar menüsünden seçilir. Donanıma ait *.asm dosyası program tarafından otomatik olarak oluşturulur. Kullanıcı oluşturulan bu dosyayı PIC'in makine diline MPASM programını çalıştırarak dönüştürür. Herhangi bir PIC programlayıcı kullanılarak derlenmiş olan *.hex dosyaları donanıma yüklenerek PIC-PLC programlanmış olur. Arayüz kullanılarak Ladder diyagramı ile PIC-PLC programlama işleminin akış şeması Şekil 4.17.'de görülmektedir.



Şekil 4. 17. Arayüz kullanarak PIC-PLC'nin programlanması

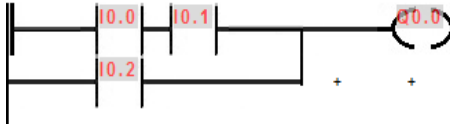
4.3. Derleme Algoritması

Geliştirilen arayüzde ilk olarak sistem tarafından kaç adet başlangıç noktası olduğu belirlenir. Daha sonra başlangıç noktası sayısı kontrol edilir. Her bir satır için eklenen komponentlerin Ladder veya Paralel Çizgi olup olmadığı kontrolü yapılır. Eğer eklenen Ladder komponent ise taranan satırda bulunan Ladderlar arasından ilk eleman tespit edilir. Tespit işleminin ardından eklenen komponentin ilk değeri örneğin NO (Normalde Açık Kontak) ya da NC (Normalde Kapalı Kontak) vb. Ladder komponentlerden biri ise ona ait ilk değer ifadesi olan makrolardan “ld” ya da “ld_not” ifadeleri derleme dosyasına sonuç olarak yazılmaktadır. Eklenen diğer komponentlerin (kontak, zamanlayıcı, sayıcı, flip-flop, aritmetik işlemler, karşılaştırma işlemleri vb.) algılamak için ilk eklenen komponentten sonra çıkış ifadesinin olup olmadığı kontrol edilir. Eğer bir çıkış ifadesi var ise sistem çıkış sayısı kadar çıkış makrosunu ekrana ve

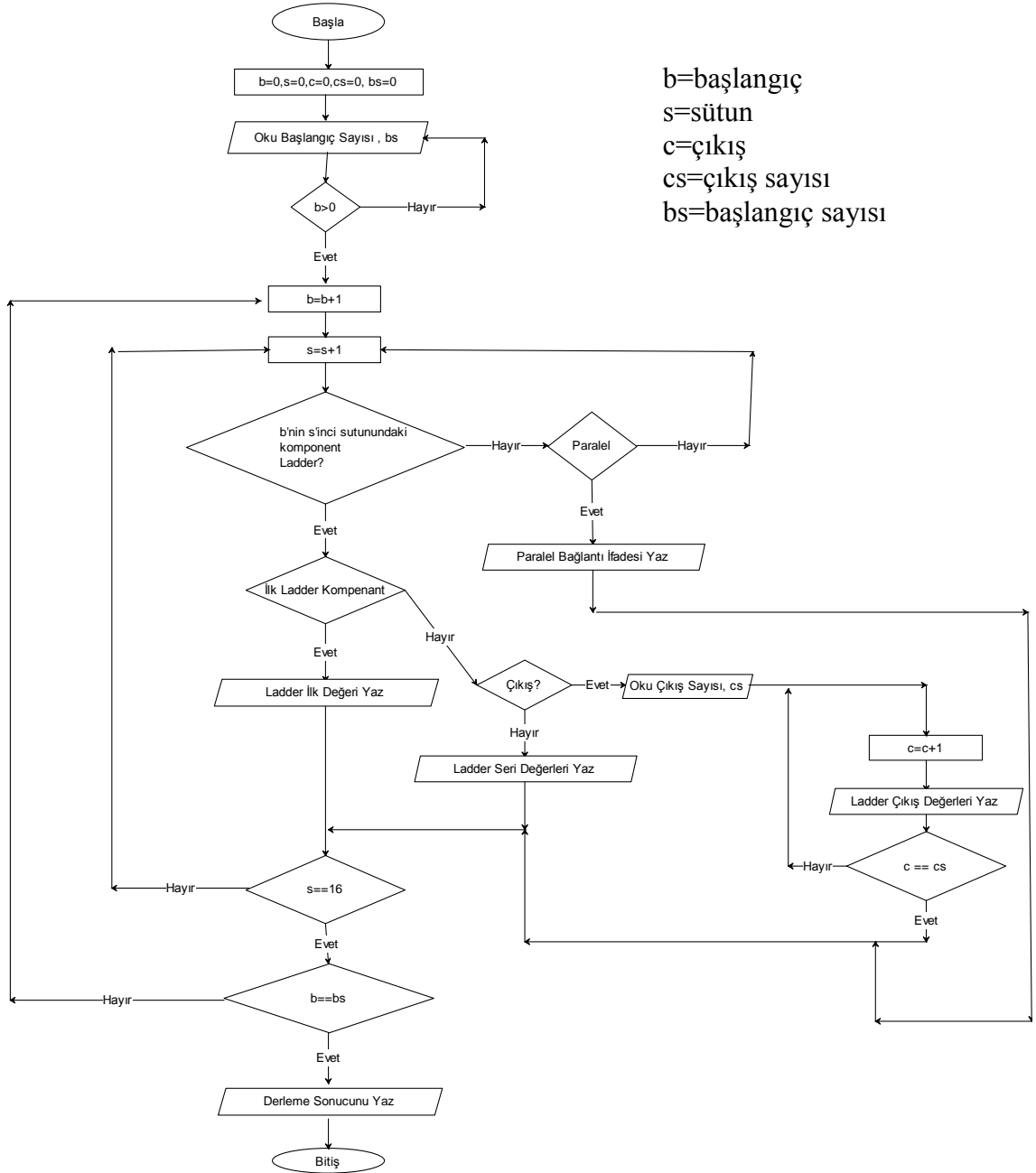
dosyaya yazmaktadır. Eklenen Ladder ifadeye çıkış yok ise assembly makro yazılımına göre seri olarak bağlantısı gerçekleşir. Örneğin bağlanan makrolar NO ya da NC olduğunu varsayarsak seri bağlantı Şekil 4.18.'deki gibi olduğunda; sistem çıktı olarak “and” veya “and_not” ifadesini yazmaktadır. Ladder komponent haricinde eklenenlerin Paralel Çizgi kontrolü yapılır. Eğer Paralel çizgin (Bkz. Çizelge 4. 3.) haricinde bir çizgi türü eklendiyse çizgi sayısı kadar sistem işlem yapmadan döner. Eklenen Paralel komponent ise Şekil 4.19.'daki gibi varsa sistem tarafından paralel ifadenin başlangıç ve bitiş noktasını belirler. Paralel kollar arasında Ladder komponent mevcutsa eklenen NO veya NC kontaklarına “or” veya “or_not” paralel ifadeleri yazılmaktadır. İşlemler eklenen başlangıç sayısı bitene kadar dönmektedir. Bu işlemlere ait akış diyagramı Şekil 4.20.'de gösterilmektedir.



Şekil 4. 18. Seri bağlantı görüntüsü



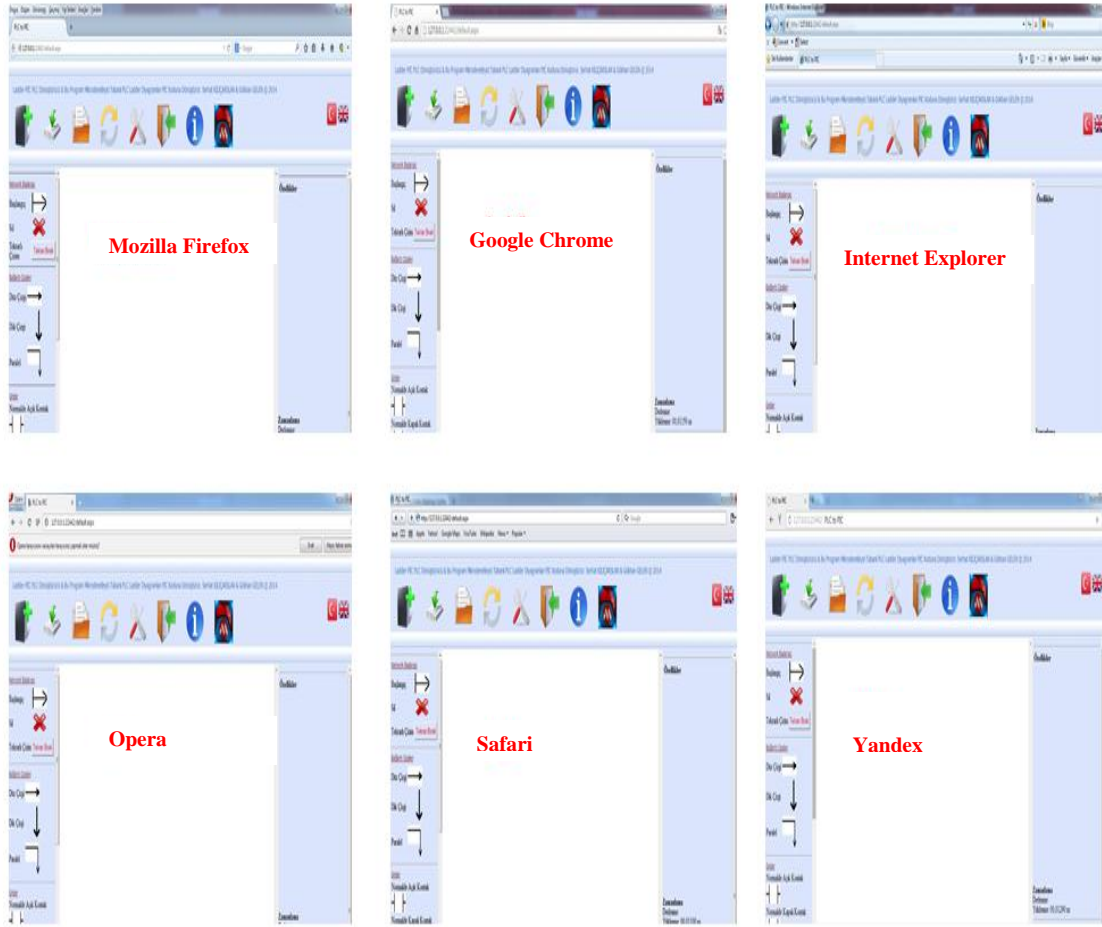
Şekil 4. 19. Paralel bağlantı görüntüsü



Şekil 4. 20. Derleme algoritması akış şeması

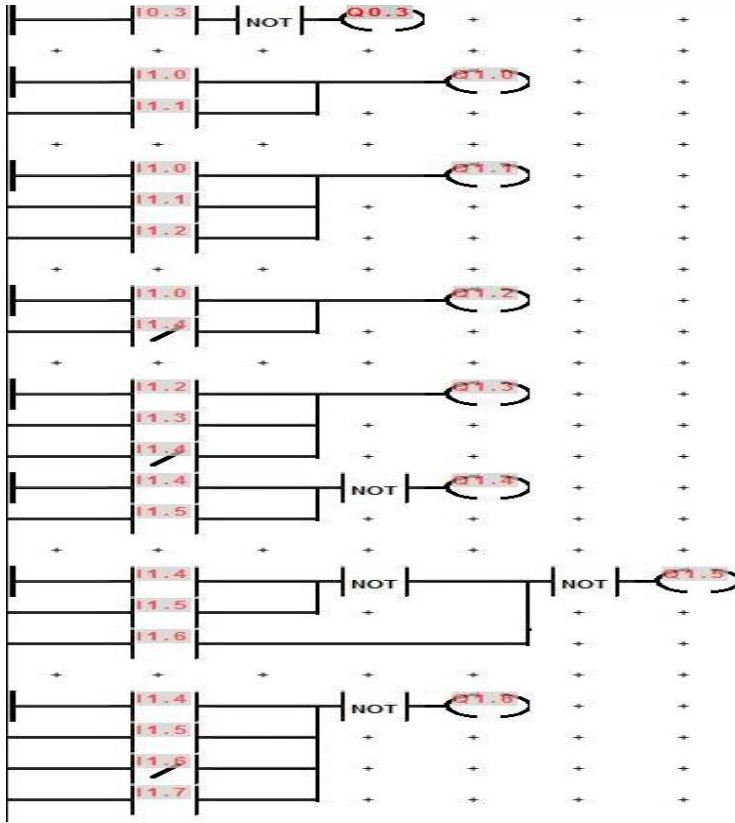
4.4. Tarayıcı ve Performans Testleri

Tez kapsamında PIC-PLC'ler için girilen Ladder diyagramları PIC Assembly makrolarına dönüştüren bir arayüz tasarlanmıştır. Tasarlanan arayüz web tabanlı olup tarayıcılar üzerinde çalıştırılmaktadır. Gerçekleştirilen yazılım tüm var olan tarayıcılarda (Internet Explorer, Google Chrome, Opera, Yandex, Mozilla Firefox gibi.) test edilmiştir ve sorunsuz çalıştığı gözlemlenmiştir. Şekil 4.21.'de arayüzün tarayıcılardaki görsellikleri sunulmuştur.



Şekil 4. 21. Tüm tarayıcılardaki ekran görüntüsü

Geliştirilen programın farklı tarayıcılardaki performans karşılaştırması amacıyla Şekil 4.22.'de görülen örnek Ladder program dikkate alınmıştır. Bu diyagrama sahip dosya farklı tarayıcılardaki yüklenme ve derlenme süreleri arayüz altına eklenen kod ile Şekil 4.23.'te görüldüğü gibi ölçülmüştür. Elde edilen test sonuçları Çizelge 4.3.'te görülmektedir. Test işlemi Pentium D 2.13 GHz işlemci 3 GB RAM ve Windows 7 yüklü bir bilgisayarda gerçekleştirilmiştir. Sonuçlardan görüleceği üzere derleme süreleri tarayıcıların tümünde birbirine çok yakındır. Yüklenme sürelerinde farklılıklar mevcut olup yapılan kullanımlarda İnternet Explorer ve Mozilla Firefox altında programın daha verimli çalıştığı tespit edilmiştir.



Şekil 4. 22. PIC-PLC için test programı

Performans Süreleri
Derlenme: 21,93840 ms
Yüklenme: 0000,00140 ms

Performans Süreleri
Derlenme: 21,93840 ms
Yüklenme: 0000,00140 ms

Derlenme

1. Basamak
İD İ0.0
out Q0.0
2. Basamak
İD İ0.1
out Q0.1
3. Basamak
İD İ0.2
out M2.7
4. Basamak
İD M2.7
out_not Q0.2
5. Basamak
İD İ0.3
out Q0.3
6. Basamak
İD İ1.0

Şekil 4. 23. Test programının derlenme sonucu

Çizelge 4. 4. Performans karşılaştırmaları

Tarayıcılar	Yüklenme Zamanı	Derlenme Zamanı
Google Chrome	21,95960 ms	4,17530 ms
Mozilla Firefox	14,34650 ms	3,95970 ms
İnternet Explorer	52,89860 ms	3,84860 ms
Opera	14,19590 ms	4,49340 ms
Yandex	15,17230 ms	3,81970 ms
Safari	36,39860 ms	3,86250 ms

4.5. Uygulamalar

Bu tez kapsamında gerçekleştirilen Ladder PIC-PLC dönüştürücü yazılımını test etmek amacıyla çeşitli örnek uygulamalar gerçekleştirilmiştir. İlk olarak temel kodların çevrimini incelemek üzere; kontak ve röle temelli makrolar, zamanlayıcı, sayıcı, flip-flop, karşılaştırma ve aritmetik işlemleri içeren kodların dönüşümü dikkate alınmıştır. Sonra gerçek bir sistemin kontrolü PIC-PLC ile sağlanmış ve kod yazımı tasarlanan arayüzle gerçekleştirilmiştir. Bu bölümde yapılan uygulamalara ait merdiven diyagramı ve derlenmiş asm dosyaları sunulmaktadır. Gerçekleştirilen tüm uygulamalarda PIC16F877A temelli PIC-PLC kullanılarak test edilmiştir.

4.6. Temel Ladder Diyagram Dönüşümleri

4.6.1. Kontak ve Röle Temelli Ladder Diyagramları

Bu uygulamada Şekil 4.24.'te görülen kontak ve röle temelli Ladder diyagramın PIC-PLC assembly koduna dönüşümü incelenmektedir. Programa ait merdiven diyagramında ld, ld_not, not, out, out_not, or, or_not, nor, nand makroları kullanılmaktadır. Tez kapsamında oluşturulan arayüz kullanılarak çizilen Ladder kodun her bir basamağının çalışması kısaca aşağıdaki gibi açıklanabilir [48].

Basamak 1. I0.0 numaralı girişe sinyal geldiğinde Q0.0 numaralı çıkış sinyal üretecektir. I0.0'daki sinyal kesildiğinde Q0.0'daki sinyalde kesilecektir.

Basamak 2. I0.1 normalde kapalı olduğundan Q0.1 numaralı çıkış sürekli sinyal üretecektir. I0.1 normalde açık hale geçerse Q0.1 sinyali kesilecektir.

Basamak 3. I0.2 numaralı girişe sinyal geldiğinde M2.7 numaralı dâhili röle enerjilenecektir. I0.2'deki sinyal kesildiğinde M2.7 numaralı dâhili rölenin enerjisi kesilecektir.

Basamak 4. M2.7 dâhili rölesinin kontağı aktif olduğunda Q0.2 numaralı çıkış sinyali kesilecektir. M2.7 dâhili rölesinin kontağı pasif olduğunda Q0.2 sinyal üretmeye başlayacaktır.

Basamak 5. I0.3 numaralı girişe sinyal uygulandığında NOT kontağı ile değıllenecektir. Bu durumda Q0.3 enerjilenemez. I0.3 sinyali kesildiğinde NOT kontağından dolayı Q0.3 enerjilenecektir.

Basamak 6. İki girişli OR bağlantısı, I1.0 ve I1.1 girişlerinden herhangi birisinden sinyal gelmesi durumunda Q1.0 çıkış verecektir.

Basamak 7. Üç girişli OR bağlantısı, I1.0, I1.1, I1.2 girişlerinden herhangi birisinden sinyal gelmesi durumunda Q1.1 çıkış verecektir.

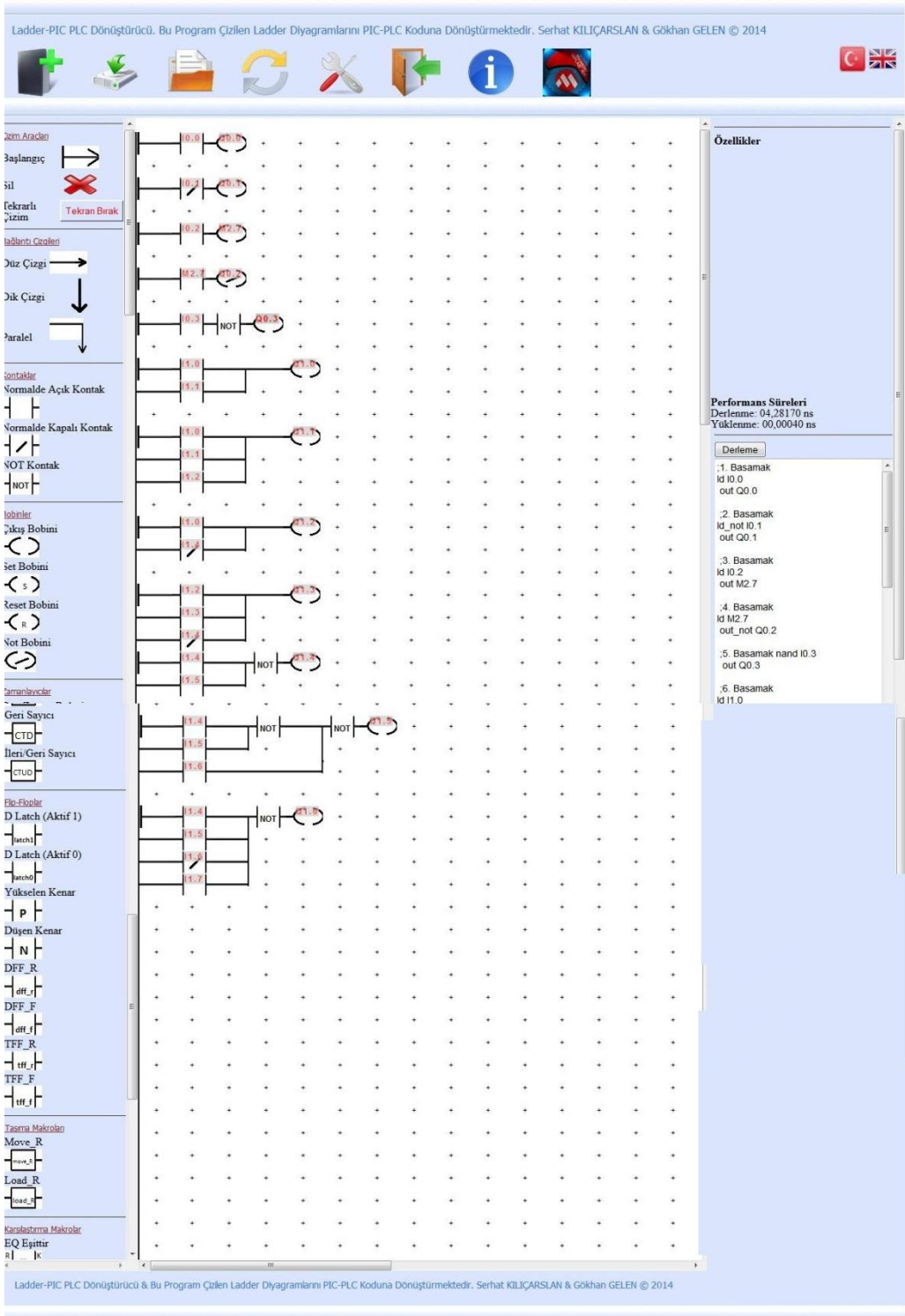
Basamak 8. OR_NOT bağlantısı, I1.4 girişi aktif olduğunda Q1.2 çıkışlı çalışmaz. I1.0 aktif ve I1.4 pasif oldukları durumda Q1.2 çıkış sinyali üretecektir.

Basamak 10. NOR bağlantısı, I1.4 ve I1.5 girişlerine sinyal verilmediği sürece Q1.4 sinyal üretecektir. Diğer durumlarda Q1.4 sinyal üretmeyecektir.

Basamak 11 ve basamak 12'de ise çoklu paralel ve değıl içeren örnek Ladder kodları mevcuttur.

Çizim işlemi tamamlandıktan sonra derleme işlemi ve donanım seçimi (PIC16F877A) işlemlerin ardından elde edilen PIC-PLC assembly kodu Şekil 4.25.'te görülmektedir. Ladder çizimine ait dosya uyg1.txt olarak sistem klasörüne kayıt edilmiştir.

Elde edilen 16F877A_output.asm dosyası MPASM derleyicisi ile derlenip elde edilen 16F877A_output.hex dosyası PIC'e yüklenerek çalışması incelenmiştir. Kodun doğruluğı PIC-PLC kartı üzerindeki girişlerden sinyal uygulanıp çıkışların gözlenmesi ile test edilmiştir. Kart üzerinde yapılan denemelerde üretilen kodun ilgili Ladder programını başarılı biçimde sağladığı görülmektedir.



Şekil 4. 24. Temel kontak ve röle örneğinin arayüz ile çizimi

```

list      p=16F877A                                ;işlemciyi tanımlamak için liste direktifi
#include <p16F877A.inc>                              ;işlemciye has değişken tanımları
#include <PICPLC_877_16I_16O_tnmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_16I_16O_kontak.inc>;kontakt ve röle temelli makrolar

__CONFIG _CP_OFF & _WDT_ON & _BODEN_OFF & _PWRTE_ON & _HS_OSC & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

org 0x00      ;Reset Vektörü

main

    initialize

scan

    get_inputs
;----- kullanıcı programı burada başlar -----
;1. Basamak
ld I0.0
out Q0.0

;2. Basamak
ld_not I0.1
out Q0.1

;3. Basamak
ld I0.2
out M2.7

;4. Basamak
ld M2.7
out_not Q0.2

;5. Basamak nand I0.3
out Q0.3

;6. Basamak
ld I1.0
or I1.1
out Q1.0

;7. Basamak
ld I1.0
or I1.1
or I1.2
out Q1.1

;8. Basamak
ld I1.0
or_not I1.4
out Q1.2

;9. Basamak
ld I1.2
or I1.3
or_not I1.4
out Q1.3

;10. Basamak
ld I1.4
nor I1.5
out Q1.4

;11. Basamak
ld I1.4
nor I1.5
nor I1.6
out Q1.5

;12. Basamak
ld I1.4
or I1.5
or_not I1.6
nor I1.7
out Q1.6

;----- kullanıcı programı burada biter -----
send_outputs

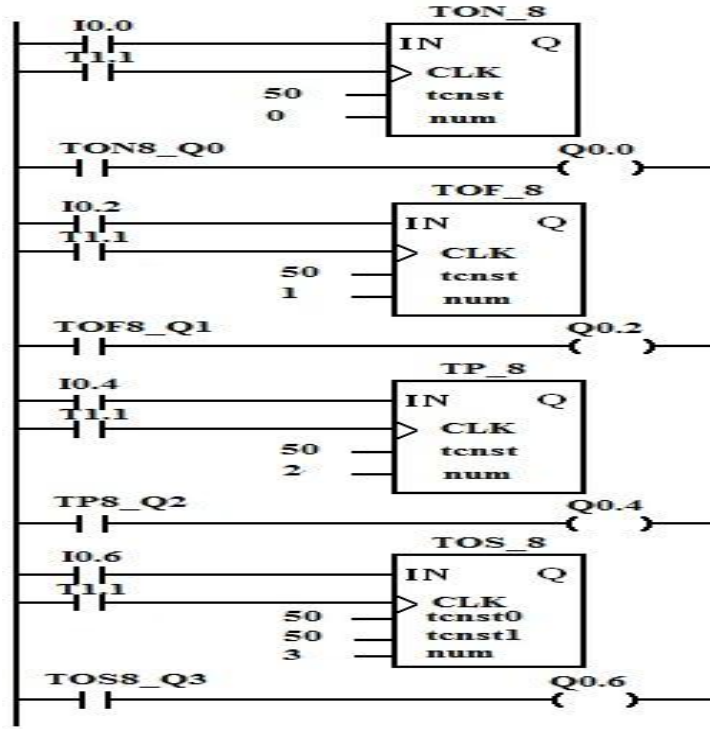
goto scan
end

```

Şekil 4. 25. Temel kontakt ve röle örneği derleme çıktısı

4.6.2. Zamanlayıcı Uygulaması

Zamanlayıcı örneğine ait merdiven diyagramı kodu Şekil 4.26.'da görülmektedir. Bu Ladder diyagramı için geliştirilen yazılımda oluşturulan koda ait ekran görüntüsü Şekil 4.27.'de görülmektedir. Bu örnekte programa ait şematik diyagram ve merdiven diyagramında TON, TOF, TP, TOS zamanlayıcıların kullanımı incelenmektedir. Arayüz ile çizilen PLC programı aşağıda açıklanmıştır.



Şekil 4. 26. Zamanlayıcı fonksiyonları örneği

Basamak1. TON, I0.0 girişine sinyal uygulandığında TON_Q0 zamanlayıcısı çalışmaya başlayacaktır. Zamanlayıcı girişindeki sinyal etkin olduğu sürece zamanlayıcı çalışmaya devam edecektir.

Basamak2. TON_Q0 zamanlayıcısı belirlenen kurma süresi $PT=tcnst \times CLK = 50 \times 52,4288 \text{ ms} = 2,62 \text{ saniye}$ sonunda aktif olacağından Q0.0 çıkış rölesi sinyal vermeye başlayacaktır.

Basamak3. TOF, I0.2 girişine sinyal uygulandığında TOF_Q1 zamanlayıcısı belirlenen süre zarfında çıkış verecektir.

Basamak4. TOF_Q1 zamanlayıcısı belirlenen kurma süresi $PT=tcnst \times CLK =50 \times 52,4288 \text{ ms} = 2,62 \text{ saniye}$ sonunda çıkış vereceğinden Q0.2 çıkış rölesi sinyal verecektir. Süre sonunda Q0.2 otomatik olarak resetlenecektir.

Basamak5. TP, I0.4 girişine sinyal uygulandığında TP_Q2 zamanlayıcısı belirlenen süre zarfında çıkış verecektir.

Basamak6. TP_Q2 zamanlayıcısı belirlenen kurma süresi $PT=tcnst \times CLK =50 \times 52,4288 \text{ ms} = 2,62 \text{ saniye}$ sonunda çıkış vereceğinden Q0.4 çıkış rölesi sinyal verecektir. Süre sonunda Q0.4 otomatik olarak resetlenecektir.

Çizim işlemi tamamlandıktan sonra derleme işlemi ve donanım seçimi (PIC16F877A) işlemlerin ardından elde edilen PIC-PLC assembly kodu Şekil 4.28.'de görülmektedir. Ladder çizimine ait dosya uyg2.txt olarak sistem klasörüne kayıt edilmiştir.

Elde edilen 16F877A_output.asm dosyası MPASM derleyicisi ile derlenip elde edilen 16F877A_output.hex dosyası PIC'e yüklenerek çalışması incelenmiştir. Kodun doğruluğu PIC-PLC kartı üzerindeki girişlerden sinyal uygulanıp çıkışların gözlenmesi ile test edilmiştir. Kart üzerinde yapılan denemelerde üretilen kodun ilgili Ladder programını başarılı biçimde sağladığı görülmektedir.

Ladder-PIC PLC Dönüştürücü. Bu Program Çizilen Ladder Diyagramlarını PIC-PLC Koduna Dönüştürmektedir. Serhat KILIÇARSLAN & Gökhan GELEN © 2014

Özellikler

Performans Süreleri
 Derleme: 01.52820 ns
 Yükleme: 00.00040 ns

Derleme

.1. Basamak
 Id IO.0
 TON_8 0,T1.1,.50

.2. Basamak
 Id TON8_Q0
 out Q0.0

.3. Basamak
 Id IO.2
 TOF_8 1,T1.1,.50

.4. Basamak
 Id TOF8_Q1
 out Q0.2

.5. Basamak
 Id IO.4
 TP_8 2,T1.1,.50

.6. Basamak
 Id TP8_Q2

Ladder-PIC PLC Dönüştürücü & Bu Program Çizilen Ladder Diyagramlarını PIC-PLC Koduna Dönüştürmektedir. Serhat KILIÇARSLAN & Gökhan GELEN © 2014

Şekil 4. 27. Zamanlayıcı örneğinin arayüz ile çizimi

```

list      p=16F877A                                ;işlemciyi tanımlamak için liste direktifi
#include <p16F877A.inc>                             ;işlemciye has değişken tanımları
#include <PICPLC_877_16I_160_tnmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_16I_160_kontak.inc>;kontak ve röle temelli makrolar

__CONFIG _CP_OFF & _WDT_ON & _BODEN_OFF & _PWRTE_ON & _HS_OSC & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

main      org    0x00                                ;Reset Vektörü

          initialize

          scan

          get_inputs

;----- kullanıcı programı burada başlar -----
;1. Basamak
ld IO.0
TON_8 0,T1.1,.50

;2. Basamak
ld TON8_Q0
out Q0.0

;3. Basamak
ld IO.2
TOF_8 1,T1.1,.50

;4. Basamak
ld TOF8_Q1
out Q0.2

;5. Basamak
ld IO.4
TF_8 2,T1.1,.50

;6. Basamak
ld TP8_Q2
out Q0.4

;7. Basamak
ld IO.6
TOS_8 3,T1.1,.50.50

;8. Basamak
ld TOS8_Q3
out Q0.6

;----- kullanıcı programı burada biter -----
          send_outputs

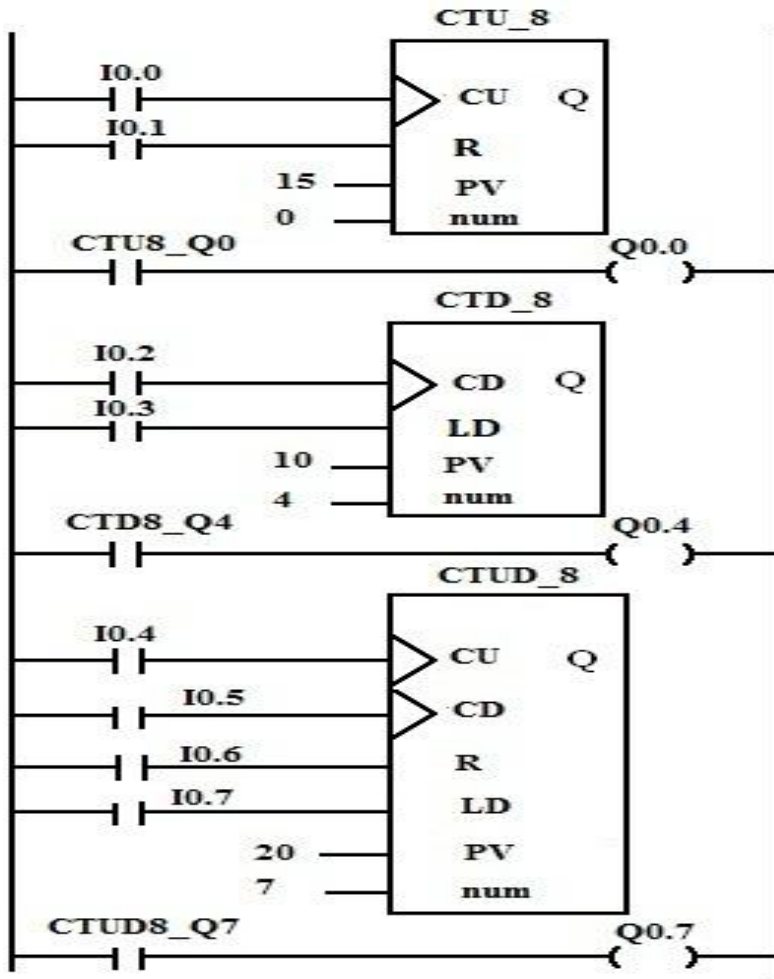
          goto   scan
          end

```

Şekil 4. 28. Zamanlayıcı örneği derleme çıktısı

4.6.3. Sayıcı Uygulaması

Sayıcı örneğine ait merdiven diyagramı kodu Şekil 4.29.'da görülmektedir. Bu Ladder diyagramı için geliştirilen yazılımda oluşturulmuş koda ait ekran görüntüsü Şekil 4.30.'da görülmektedir. Bu örnekte programa ait şematik diyagram ve merdiven diyagramında CTU, CTD, CTUD, makrolarının kullanımı incelenmektedir. Arayüz ile çizilen PLC programı aşağıda açıklanmıştır.



Şekil 4. 29. Sayıcı fonksiyonları örneği

Basamak 1. İleri sayıcı (CTU_8) gerçekleştirilmiştir. CU girişi I0.0'dan ve R girişi de I0.1'den alınmıştır. num=0 olarak alındığı için ilk ileri sayıcı seçilmiştir. Bu ileri sayıcının biti CTU8_Q0'dır. Sayıcının kurma değeri PV=15 olarak belirlenmiştir.

Basamak 2. İleri sayıcının durum biti CTU8_Q0, Q0.0 çıkışına gönderilmektedir.

Basamak 3. Geri sayıcı gerçekleştirilmiştir. CD girişi I0.2'den ve LD girişi de I0.3'ten alınmıştır. num=4 olarak alındığı için beşinci geri sayıcı seçilmiştir. Bu geri sayıcının durum biri CTD8_Q4'tür. Sayıcı kurma değeri PV=10 olarak belirlenmiştir.

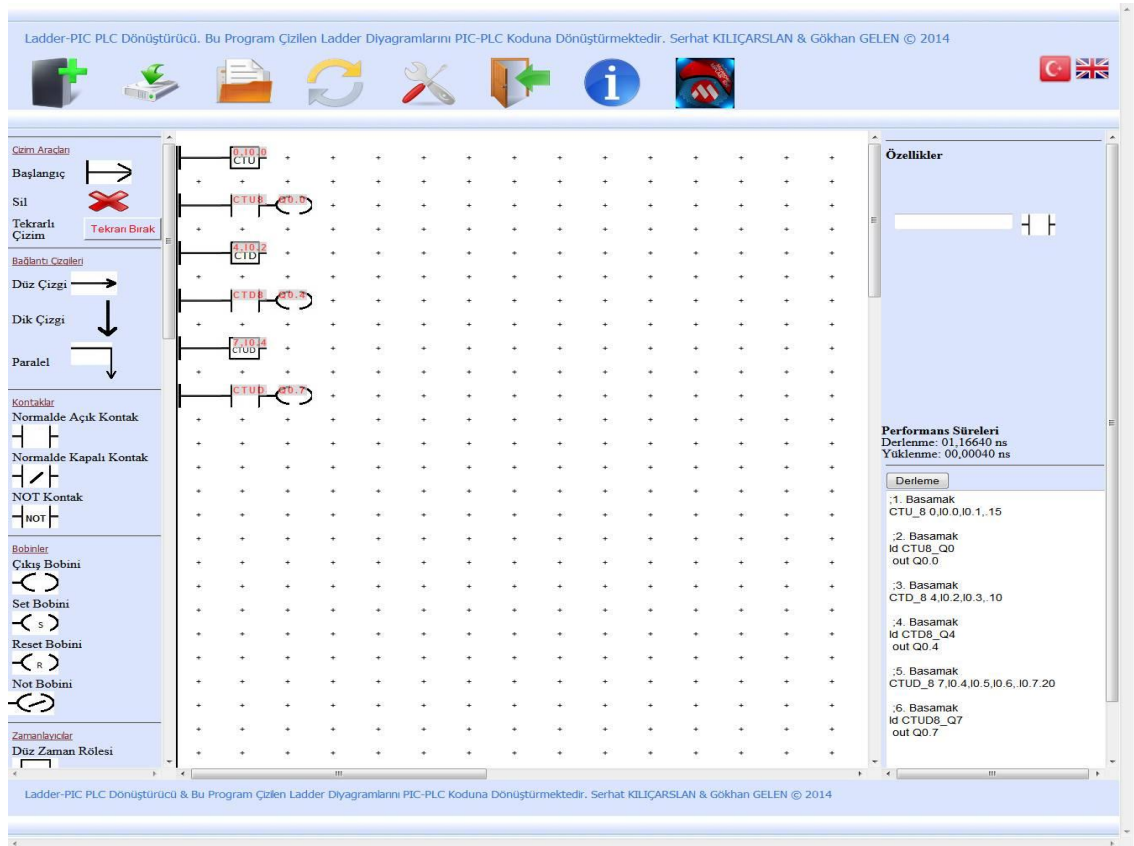
Basamak 4. Geri sayıcı durum biti CTD8_Q4, Q0.4 çıkışına gönderilmiştir.

Basamak 5. İleri/geri sayıcı (CTUD_8) gerçekleştirilmiştir. CU girişi I0.4'ten, CD girişi I0.5'ten, R girişi I0.6'dan ve LD girişi de I0.7'ten alınmıştır. num=7 olarak alındığı için sekizinci ileri/geri sayıcı seçilmiştir. Bu ileri/geri sayıcının durum biti CTUD8_Q7'dir. Sayıcı kurma değeri PV=20 olarak belirlenmiştir.

Basamak 6. İleri/geri sayıcı durum biti CTUD8_Q7, Q0.7 çıkışına gönderilmiştir.

Çizim işlemi tamamlandıktan sonra derleme işlemi ve donanım seçimi (PIC16F877A) işlemlerin ardından elde edilen PIC-PLC assembly kodu Şekil 4.31.'de görülmektedir. Ladder çizimine ait dosya uyg3.txt olarak sistem klasörüne kayıt edilmiştir.

Elde edilen 16F877A_output.asm dosyası MPASM derleyicisi ile derlenip elde edilen 16F877A_output.hex dosyası PIC'e yüklenerek çalışması incelenmiştir. Kodun doğruluğu PIC-PLC kartı üzerindeki girişlerden sinyal uygulanıp çıkışların gözlenmesi ile test edilmiştir. Kart üzerinde yapılan denemelerde üretilen kodun ilgili Ladder programını başarılı biçimde sağladığı görülmektedir.



Şekil 4. 30. Sayıcı örneğinin arayüz ile çizimi

```

list      p=16F877A                                ;işlemciyi tanımlamak için liste direktifi
#include <p16F877A.inc>                               ;işlemciye has değişken tanımları
#include <PICPLC_877_16I_160_tmmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_16I_160_kontak.inc>;kontak ve röle temelli makrolar

__CONFIG _CP_OFF & _WDI_ON & _BODEN_OFF & _PWRTE_ON & _HS_OSC & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

      org 0x000000                                ;Reset Vektörü
main
      initialize

scan
      get_inputs
;----- kullanıcı programı burada başlar -----
;1. Basamak
CTU8_8 0,I0.0,I0.1,.15

;2. Basamak
ld CTU8_Q0
out Q0.0

;3. Basamak
CTD8_8 4,I0.2,I0.3,.10

;4. Basamak
ld CTD8_Q4
out Q0.4

;5. Basamak
CTUD8_8 7,I0.4,I0.5,I0.6,.I0.7.20

;6. Basamak
ld CTUD8_Q7
out Q0.7

;----- kullanıcı programı burada biter -----
      send_outputs

      goto scan
      end

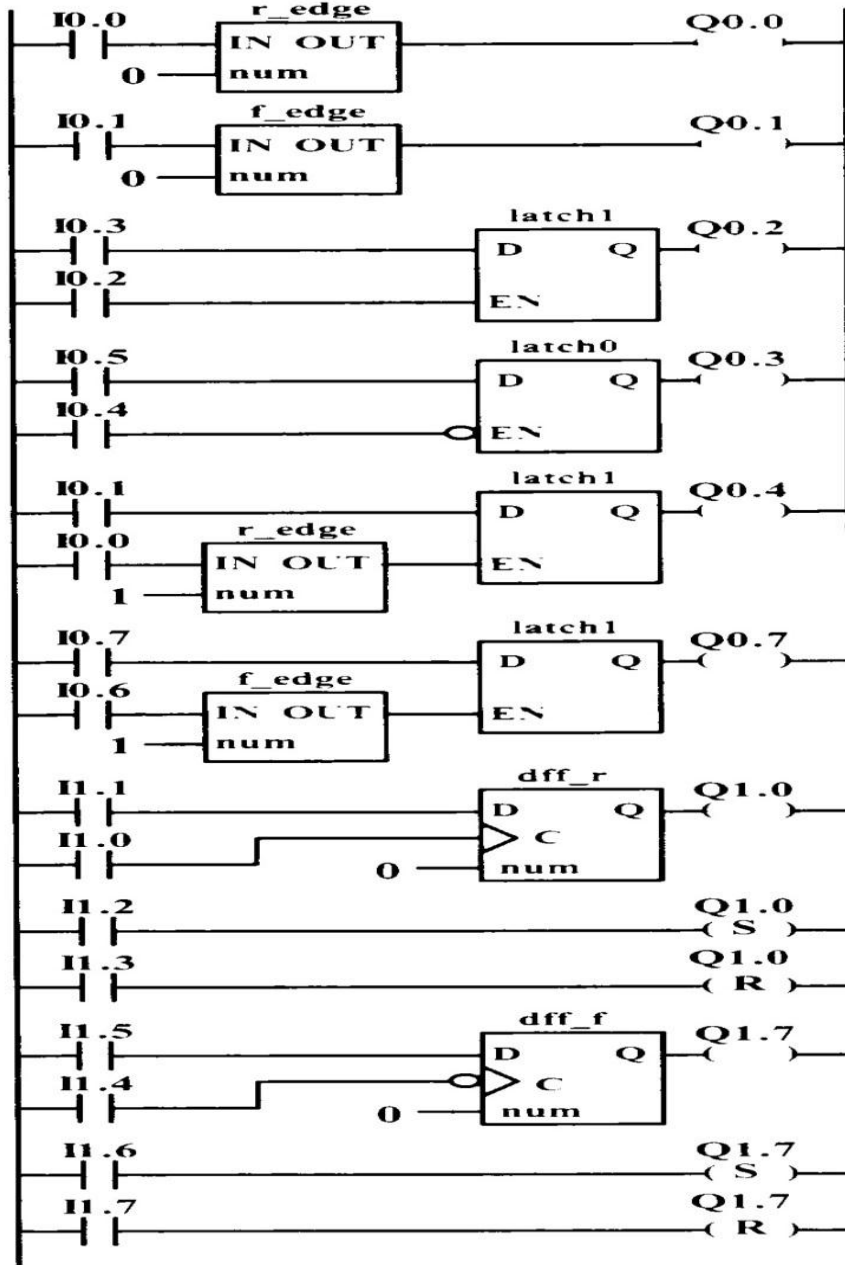
```

Şekil 4. 31. Sayıcı örneği derleme çıktısı

4.6.4. Flip-Flop Ugulaması

Flip-Flop örneğine ait merdiven diyagramı kodu Şekil 4.32.'de görülmektedir. Bu Ladder diyagramı için geliştirilen yazılımda oluşturulmuş koda ait ekran görüntüsü Şekil 4.33.'te görülmektedir. Bu örnekte programa ait şematik diyagram ve merdiven

diyagramında R_EDGE, F_EDGE, LATCH0, LATCH1, DFF_R, DFF_F, makrolarının kullanımı incelenmektedir.



Şekil 4. 32. Flip-flop ladder örneği

Çizim işlemi tamamlandıktan sonra derleme işlemi ve donanım seçimi (PIC16F877A) işlemlerin ardından elde edilen PIC-PLC assembly kodu Şekil 4.34.'te görülmektedir. Ladder çizimine ait dosya uyg4.txt olarak sistem klasörüne kayıt edilmiştir.

Elde edilen 16F877A_output.asm dosyası MPASM derleyicisi ile derlenip elde edilen 16F877A_output.hex dosyası PIC'e yüklenerek çalışması incelenmiştir. Kodun doğruluğu PIC-PLC kartı üzerindeki girişlerden sinyal uygulanıp çıkışların gözlenmesi ile test edilmiştir. Kart üzerinde yapılan denemelerde üretilen kodun ilgili Ladder programını başarılı biçimde sağladığı görülmektedir.

Ladder-PIC PLC Dönüştürücü. Bu Program Çizilen Ladder Diyagramlarını PIC-PLC Koduna Dönüştürmektedir. Serhat KILIÇARSLAN & Gökhan GELEN © 2014

The screenshot displays the Ladder-PIC PLC Dönüştürücü software interface. The main window shows a ladder logic diagram with five rungs. The components are as follows:

- Rung 0:** A normally open contact labeled '0.0' is connected to a pulse coil labeled 'P' and a set coil labeled 'Q0.0'. A latch labeled 'latch1' is also present.
- Rung 1:** A normally open contact labeled '0.1' is connected to a normally closed contact labeled 'N' and a set coil labeled 'Q0.1'. A latch labeled 'latch1' is also present.
- Rung 2:** A normally open contact labeled '0.2' is connected to a set coil labeled 'Q0.3'. A latch labeled 'latch1' is also present.
- Rung 3:** A normally open contact labeled '0.4' is connected to a set coil labeled 'Q0.5'. A latch labeled 'latch0' is also present.
- Rung 4:** A normally open contact labeled '0.0' is connected to a pulse coil labeled 'P' and a set coil labeled 'Q0.1'. A latch labeled 'latch1' is also present.
- Rung 5:** A normally open contact labeled '0.6' is connected to a normally closed contact labeled 'N' and a set coil labeled 'Q0.7'. A latch labeled 'latch1' is also present.

The left sidebar contains the following categories and components:

- Çizim Araçları:** Başlangıç, Sil, Tekrarlı Çizim (Tekrarı Bırak), Bağlantı Çizimleri (Düz Çizgi, Dik Çizgi, Paralel).
- Kontaklar:** Normalde Açık Kontak, Normalde Kapalı Kontak, NOT Kontak.
- Bobinler:** Çıkış Bobini, Set Bobini, Reset Bobini, Not Bobini.
- Zamanlayıcılar:** Düz Zaman Rölesi.

The right sidebar shows the 'Özellikler' (Properties) section with the following performance metrics:

- Performans Süreleri:** Derleme: 04.35480 ns, Yükleme: 00.00040 ns.
- Derleme:** ;1. Basamak, Id I0.0, r_edge 0, out Q0.0; ;2. Basamak, Id I0.1, f_edge 0, out Q0.1; ;3. Basamak, Id I0.2, latch1 I0.3,Q0.2; ;4. Basamak, Id I0.4, latch0 I0.5,Q0.3; ;5. Basamak, Id I0.0, r_edge 1, latch1 I0.1,Q0.4.

Ladder-PIC PLC Dönüştürücü & Bu Program Çizilen Ladder Diyagramlarını PIC-PLC Koduna Dönüştürmektedir. Serhat KILIÇARSLAN & Gökhan GELEN © 2014

Şekil 4. 33. Flip-flop örneğinin arayüz ile çizimi

```

list      p=16F877A                                ;işlemciyi tanımlamak için liste direktifi
#include <pi16F877A.inc>                            ;işlemciye has değişken tanımları
#include <PICPLC_877_16I_16O_tnmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_16I_16O_kontak.inc>;kontak ve röle temelli makrolar

_CONFIG _CP_OFF & _WDI_ON & _BODEN_OFF & _PWRTE_ON & _HS_OSC & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

org      0x00                                ;Reset Vektörü

main

    initialize

scan

    get_inputs
;----- kullanıcı programı burada başlar -----
;1. Basamak
ld IO.0
r_edge 0
out Q0.0

;2. Basamak
ld IO.1
f_edge 0
out Q0.1

;3. Basamak
ld IO.2
latch1 IO.3,Q0.2

;4. Basamak
ld IO.4
latch0 IO.5,Q0.3

;5. Basamak
ld IO.0
r_edge 1
latch1 IO.1,Q0.4

;6. Basamak
ld IO.6
f_edge 1
latch1 IO.7,Q0.7

;7. Basamak
ld I1.0
dff_r 0,I1.1,Q1.0

;8. Basamak
ld I1.2
_set Q1.0

;9. Basamak
ld I1.3
_reset Q1.0

;10. Basamak
ld I1.4
dff_f 0,I1.5,Q1.7

;11. Basamak
ld I1.6
_set Q1.7

;12. Basamak
ld I1.7
_reset Q1.7

;----- kullanıcı programı burada biter -----
send_outputs

goto    scan
end

```

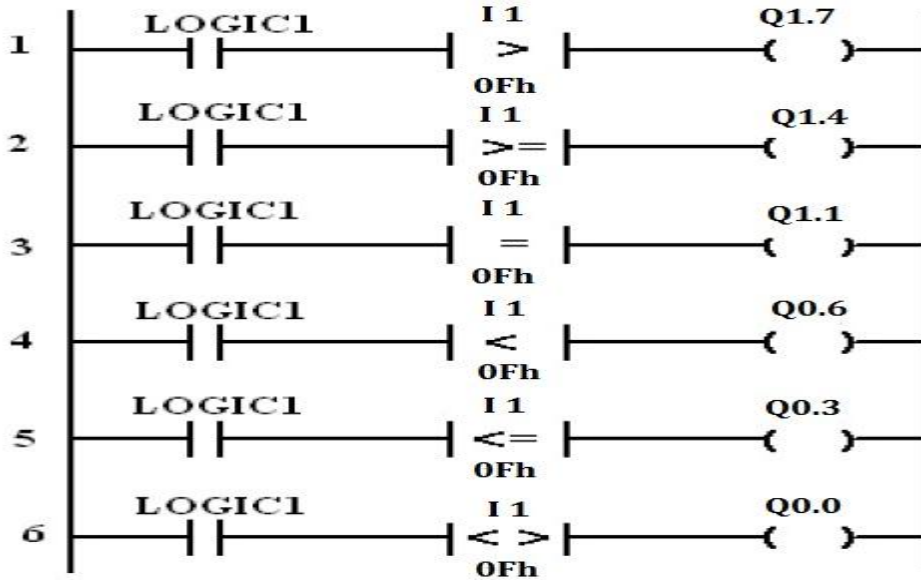
Şekil 4. 34. Flip-Flop örneği derleme çıktısı

4.6.5. Karşılaştırma Fonksiyonları Uygulaması

Karşılaştırma örneğine ait merdiven diyagramı kodu Şekil 4.35.'te görülmektedir. Bu Ladder diyagramı için geliştirilen yazılımda oluşturulan koda ait ekran görüntüsü Şekil 4.36.'da görülmektedir. Bu örnekte programa ait şematik diyagram ve merdiven diyagramında GT, GE, EQ, LT, LE, NE makrolarının kullanımı incelenmektedir.

Bu bölümde karşılaştırma makrolarının kullanımı örnekte incelenmektedir. Birinci basamakta I1'in değeri 0Fh'ten büyük olup olmadığını anlamak için yapılan bir karşılaştırma işlemi yapılmakta ve sonuç Q1.7 no'lu çıkışa gönderilmektedir. İkinci

basamakta I1'in değeri 0Fh'ten büyük veya eşit olup olmadığını anlamak için yapılan bir karşılaştırma işlemi yapmakta ve sonuç Q0.6 no'lu çıkışa gönderilmektedir. Üçüncü basamakta I1'in değeri 0Fh'e eşit olup olmadığını anlamak için yapılan bir karşılaştırma işlemi yapmakta ve sonuç Q1.1 no'lu çıkışa atanmaktadır. Dördüncü basamakta I1'in değeri 0Fh'ten küçük olup olmadığını anlamak için yapılan bir karşılaştırma işlemi yapmakta ve sonuç Q0.6 no'lu çıkışa gönderilmektedir. Beşinci basamakta I1'in değeri 0Fh'ten küçük veya eşit olup olmadığını anlamak için yapılan bir karşılaştırma işlemi yapmakta ve sonuç Q0.3 no'lu çıkışa atanmaktadır. Altıncı basamakta I1'in değeri 0Fh'e eşit olmadığını anlamak için yapılan bir karşılaştırma işlemi yapmakta ve sonuç Q0.0 no'lu çıkışa göndermektedir.

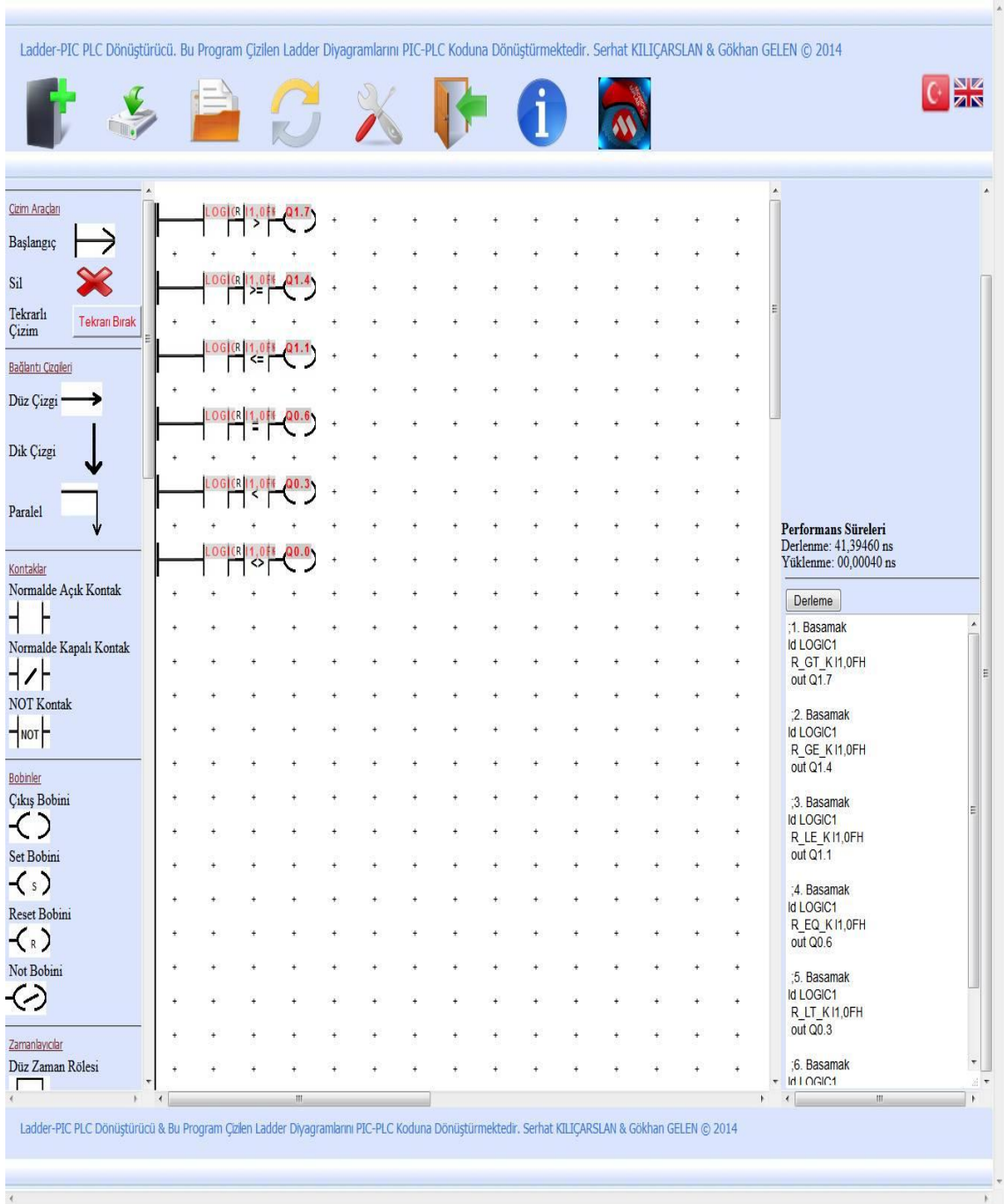


Şekil 4. 35. Karşılaştırma fonksiyonları örneği

Çizim işlemi tamamlandıktan sonra derleme işlemi ve donanım seçimi (PIC16F877A) işlemlerin ardından elde edilen PIC-PLC assembly kodu Şekil 4.37.'de görülmektedir. Ladder çizimine ait dosya uyg5.txt olarak sistem klasörüne kayıt edilmiştir.

Elde edilen 16F877A_output.asm dosyası MPASM derleyicisi ile derlenip elde edilen 16F877A_output.hex dosyası PIC'e yüklenerek çalışması incelenmiştir. Kodun doğruluğu PIC-PLC kartı üzerindeki girişlerden sinyal uygulanıp çıkışların gözlenmesi

ile test edilmiştir. Kart üzerinde yapılan denemelerde üretilen kodun ilgili Ladder programını başarılı biçimde sağladığı görülmektedir.



Şekil 4. 36. Karşılaştırma örneğinin arayüz ile çizimi

```

list      p=16F877A                                ;işlemciyi tanımlamak için liste direktifi
#include <p16F877A.inc>                             ;işlemciye has değişken tanımları
#include <PICPLC_877_16I_16O_tmmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_16I_16O_kontak.inc>;kontak ve röle temelli makrolar

__CONFIG __CP_OFF & __WDT_ON & __BODEN_OFF & __FWRTE_ON & __HS_OSC & __LVP_OFF & __DEBUG_OFF & __CPD_OFF

      org 0x00                                     ;Reset Vektörü
main
      initialize
scan
      get_inputs
;----- kullanıcı programı burada başlar -----
;1. Basamak
ld LOGIC1
R_GT_K I1,0FH
out Q1.7

;2. Basamak
ld LOGIC1
R_GE_K I1,0FH
out Q1.4

;3. Basamak
ld LOGIC1
R_LE_K I1,0FH
out Q1.1

;4. Basamak
ld LOGIC1
R_EQ_K I1,0FH
out Q0.6

;5. Basamak
ld LOGIC1
R_LT_K I1,0FH
out Q0.3

;6. Basamak
ld LOGIC1
R_NE_K I1,0FH
out Q0.0

;----- kullanıcı programı burada biter -----
      send_outputs
      goto scan
      end

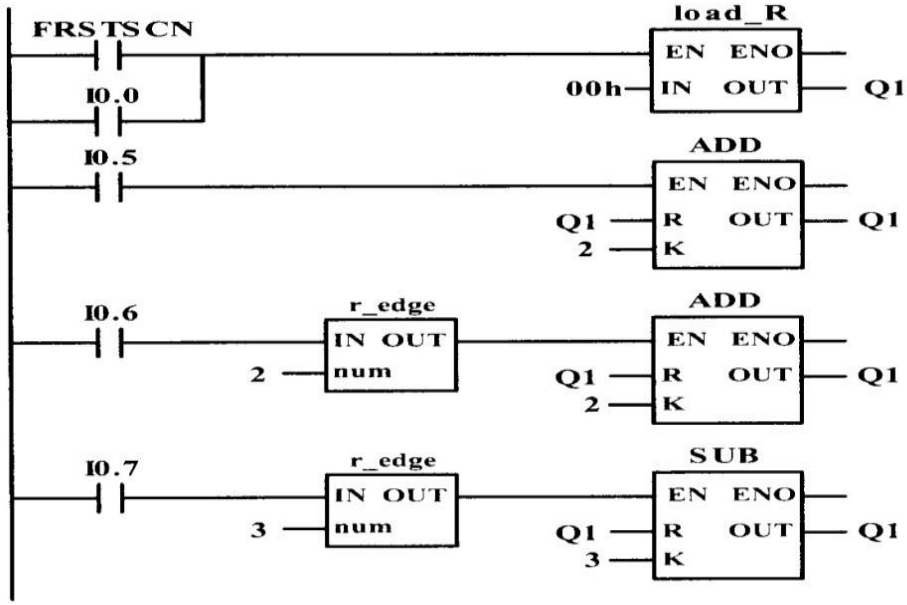
```

Şekil 4. 37. Karşılaştırma örneği derleme çıktısı

4.6.6. Aritmetik Fonksiyonların Uygulaması

Aritmetik fonksiyonlara kullanımına ait örnek merdiven diyagramı kodu Şekil 4.38.'de görülmektedir. Bu Ladder diyagramı geliştirilen yazılımda oluşturulmuş koda ait ekran görüntüsü Şekil 4.39.'da görülmektedir. Bu örnekte programa ait şematik diyagram ve merdiven diyagramında ADD, SUB makrolarının kullanımı incelenmektedir.

Birinci basamakta ilk PLC taramasında lojik 1 olan FRSTSCN biti ile veya daha sonraki PLC taramalarında I0.0'ın lojik 1 olması durumunda Q1=0 yapılmaktadır. İkinci basamakta I0.5=1 yapıldığında her PLC taramasında Q1=Q1+2 işlemi gerçekleşir. Böylelikle Q1 değeri sürekli olarak değişecektir. Üçüncü basamakta I0.6'nın yükselen kenarında Q1=Q1+2 işlemi gerçekleştirilir. Dördüncü basamakta I0.7'nin yükselen kenarında Q1=Q1-3 işlemi gerçekleştirilir.



Şekil 4. 38. Aritmetik fonksiyonları örneği

Çizim işlemi tamamlandıktan sonra derleme işlemi ve donanım seçimi (PIC16F877A) işlemlerin ardından elde edilen PIC-PLC assembly kodu Şekil 4.39.'da görülmektedir. Bu dosya uyg6.txt olarak sistem klasörüne kayıt edilmiştir.

Elde edilen 16F877A_output.asm dosyası MPASM derleyicisi ile derlenip elde edilen 16F877A_output.hex dosyası PIC'e yüklenerek çalışması incelenmiştir. Kodun doğruluğu PIC-PLC kartı üzerindeki girişlerden sinyal uygulanıp çıkışların gözlenmesi ile test edilmiştir. Kart üzerinde yapılan denemelerde üretilen kodun ilgili Ladder programını başarılı biçimde sağladığı görülmektedir.

Ladder-PIC PLC Dönüştürücü. Bu Program Çizilen Ladder Diyagramlarını PIC-PLC Koduna Dönüştürmektedir. Serhat KILIÇARSLAN & Gökhan GELEN © 2014

Çizim Araçları

Başlangıç

Sil

Tekrarlı Çizim **Tekrarı Bırak**

Bağlantı Çizimleri

Düz Çizgi

Dik Çizgi

Paralel

Kontaklar

Normalde Açık Kontak

Normalde Kapalı Kontak

NOT Kontak

NOT

Bobinler

Çıkış Bobini

Set Bobini

Reset Bobini

Not Bobini

Zamanlayıcılar

Düz Zaman Rölesi

Özellikler

Performans Süreleri
 Derlenme: 00,89450 ns
 Yükleme: 00,00040 ns

Derleme

```

;1. Basamak
ld FRSTSCN
or I0.0
load_R 00h,Q1

;2. Basamak
ld I0.5
RaddK Q1.2,Q1

;3. Basamak
ld I0.6
r_edge 2
RaddK Q1.2,Q1

;4. Basamak
ld I0.7
r_edge 3
RsubK Q1.3,Q1

```

Ladder-PIC PLC Dönüştürücü & Bu Program Çizilen Ladder Diyagramlarını PIC-PLC Koduna Dönüştürmektedir. Serhat KILIÇARSLAN & Gökhan GELEN © 2014

Şekil 4. 39. Aritmetik örneğinin arayüz ile çizimi

```

list      p=16F877A                                ;işlemciyi tanımlamak için liste direktifi
#include <pic16f877a.inc>                            ;işlemciye has değişken tanımları
#include <PICPLC_877_16I_160_tmmlr.inc> ;temel tanımlar ve makrolar
#include <PICPLC_877_16I_160_kontak.inc>;kontak ve röle temelli makrolar

__CONFIG _CP_OFF & _WDI_ON & _BODEN_OFF & _PWRTE_ON & _HS_OSC & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

org      0x0000                                ;Reset Vektörü
main
        initialize
scan
        get_inputs
;----- kullanıcı programı burada başlar -----
;1. Basamak
ld FRSTSCN
or IO.0
load_R 00h,Q1

;2. Basamak
ld IO.5
RaddK Q1,.2,Q1

;3. Basamak
ld IO.6
r_edge 2
RaddK Q1,.2,Q1

;4. Basamak
ld IO.7
r_edge 3
RsubK Q1,.3,Q1

;----- kullanıcı programı burada biter -----
        send_outputs
        goto scan
end

```

Şekil 4. 40. Aritmetik uygulama derleme çıktısı

4.7. Gerçek Sistem Kontrolü Uygulaması

Tez kapsamında yapılan program kullanılarak gerçek bir sistem için kod dönüşümünün test edilmesi amacıyla Gaziosmanpaşa Üniversitesi Mekatronik Mühendisliği Bölümü Endüstriyel Otomasyon ve Robotik Laboratuvarında bulunan dağıtım istasyonu deney seti kullanılmıştır. Dağıtım istasyonunun amacı bir iş parçasının besleme haznesinden alınıp üretim tesisindeki başka bir istasyona taşınmasıdır. Dağıtım istasyonunun resmi Şekil 4.41.'de görülmektedir. Silindirik besleme haznesinin alt kısmına yerleştirilmiş bir optik sensör besleme haznesi içinde parça olup olmadığı bilgisini, pistonun geri ve ileri konumlarına yerleştirilen iki tane yakınlık sensörü de aktüatörün konum bilgisini verir. Transfer modülü ya da manipülatör pnömatik aktüatörlerden oluşmuştur. İş parçaları bir vakumlu başlık sayesinde manipülatör tarafından tutulur ve sonraki istasyona bir bırakma aracıyla bırakılır. İki tane temas sensörü manipülatörün konum bilgisini verir. Dağıtım istasyonuna ait sensör ve PLC bağlantılarına ait açıklamalar Çizelge 4.4.'te görülmektedir.



Şekil 4. 41. Dağıtım istasyonu resmi [49]

Çizelge 4. 5. Sensör ve PLC bağlantılar

Sensör	PLC Bağlantısı	Açıklama
SA1	I0.4	Manipülator 0° konumunda
SA2	I0.5	Manipülator 180° konumunda
PI	I0.2	Piston ileride
OS	I0.6	Besleme haznesinde parça var

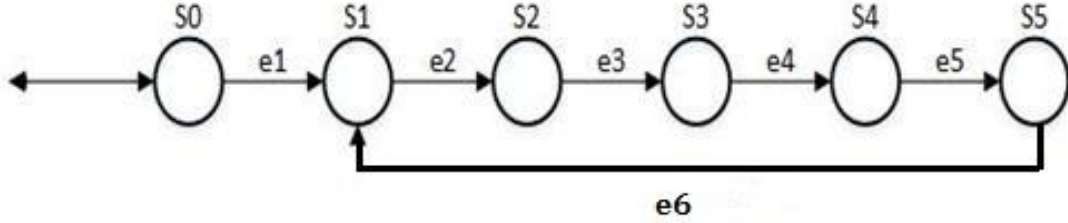
Uygulamada kullanılan kontrol senaryosu aşağıda sunulmuştur.

1. Manipülator 180° pozisyona gider.
2. Pnömatik piston ileri yönde hareket ederek iş parçasını sürer ve geri döner.
3. Manipülator 0° pozisyona gider.
4. Vakum 0.5 saniye açılır. (Parça tutulur)
5. Manipülator 180° pozisyona gider.
6. Parça bıraktırıcı 0.5 saniye açılır.
7. Döngü yeniden başlar.

Bu senaryoda 0° manipülatorün iş parçasını aldığı besleme haznesi konumu. 180° sonraki istasyon konumunu ifade etmektedir.

Yukarıda verilen kontrol senaryosunu yerine getirecek PLC Ladder kodunun formal olarak elde edilmesi açıklanmaktadır. Bu amaçla en kolay modelleme aracı olan otomatlar tercih edilmiştir. Otomata modeli; başlangıç durumu, diğer durumlar, olaylar, etkin olaylar, işaretli durumlar ve durum geçişlerinden oluşan grafiksel bir gösterimdir. Dağıtım istasyonunun verilen kontrol senaryosu için oluşturulan otomat modeli Şekil

4.42.'de görülmektedir. Bu modelde 7 durum ve 7 olay mevcuttur. Durumların ve olayların açıklamaları Çizelge 4.5. ve 4.6.'da verilmiştir.



Şekil 4. 42. Dağıtım istasyonuna ait otomat modeli

Çizelge 4. 6. Otomat modelindeki durumların açıklamaları

Olay	Açıklama
e1	Manipülator 180° de
e2	Manipülator 180° de ve magazinde parça var
e3	Piston ileride
e4	Manipülator 0° de
e5	0.5 saniye gecikme sonu
e6	Manipülator 180° de

Çizelge 4. 7. Otomat modelindeki olayların açıklamaları

Durum	Açıklama
s0	Manipülator 180° ye gidiyor
s1	Başlangıç durumu
s2	Piston ileri hareket ediyor (Parçayı sürüyor)
s3	Manipülator 0° ye gidiyor
s4	Parça tutulur (vakumu açılır)
s5	Manipülator 180° ye gidiyor

Dağıtım istasyonuna ait otomat modelinin gerçekleştirilmesi için modeldeki her bir duruma bir hafıza biti ataması yapılmıştır. Otomat modelindeki durumların hafıza biti karşılığı ve olayların PLC giriş karşılığı Çizelge 4.7.'de sunulmuştur.

Çizelge 4. 8. Otomat modelindeki durumların hafıza biti karşılığı ve olayların plc giriş karşılığı

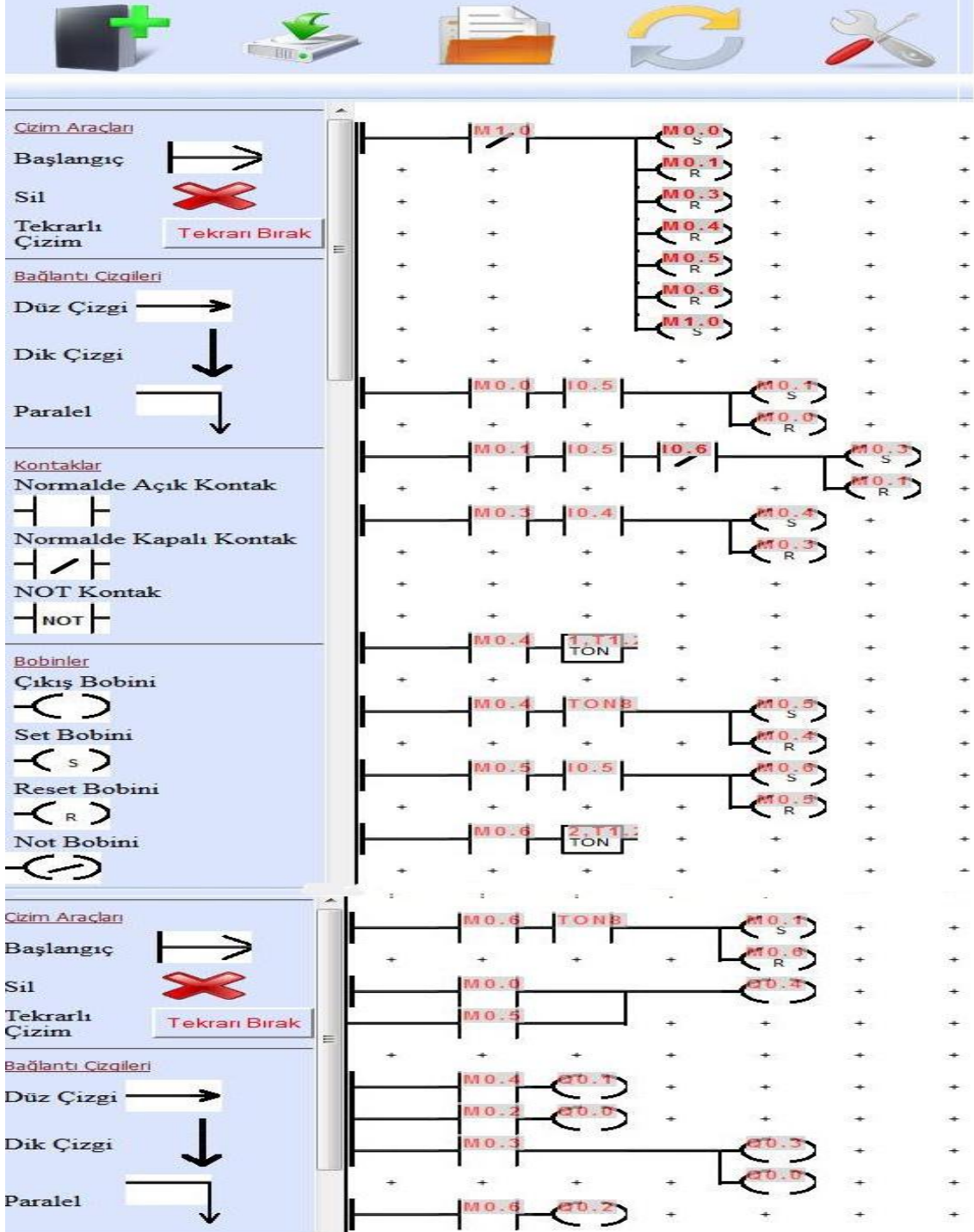
Durumlar	Hafıza Biti	Olaylar	PLC Bağlantısı
s0	M0.0	e1	I0.5
s1	M0.1	e2	I0.5 & I0.6
s2	M0.2	e3	I0.2
s3	M0.3	e4	I0.4
s4	M0.4	e5	T1
s5	M0.5	e6	I0.5

Elde edilen otomat modelinin PLC ladder kodu karşılığı Şekil 4.43.'te PIC-PLC assembly kodu ise Şekil 4.44.'de görülmektedir. Kodun ilk basamağında başlangıç durumu gerçekleştirilmektedir. Otomat modeli başlangıçta s0 durumunda olduğundan kodun başlangıcında bir defalığına s0 için atanan hafıza biti set edilmekte diğer hafıza bitleri ise resetlenmektedir.

Başlangıç atamasının ardından, otomat modelinin durum geçiş değişimi gerçekleştirilmiştir. Sırasıyla s0 durumundan diğer duruma olan geçişleri ilgili olay şartnında gerçekleşmesi ile set – reset komutları yardımıyla hafıza bitleri aracılığıyla gerçekleştirilmektedir.

Çizim işlemi tamamlandıktan sonra derleme işlemi ve donanım seçimi (PIC16F877A) işlemlerin ardından elde edilen PIC-PLC assembly kodu Şekil 4.44.'te görülmektedir. Ladder çizimine ait dosya uyg7.txt olarak sistem klasörüne kayıt edilmiştir.

Elde edilen 16F877A_output.asm dosyası MPASM derleyicisi ile derlenip elde edilen 16F877A_output.hex dosyası PIC'e yüklenerek çalışması incelenmiştir. Kodun doğruluğu PIC-PLC kartı üzerindeki girişlerden sinyal uygulanıp çıkışların gözlenmesi ile test edilmiştir. Kart üzerinde yapılan denemelerde üretilen kodun ilgili Ladder programını başarılı biçimde sağladığı görülmektedir.



Şekil 4. 43. Dağıtım istasyonuna ait kodun ladder diyagramı

```

list      p=16F877A                ;işlemciyi tanımlamak için liste direktifi
#include <p16F877A.inc>            ;işlemciye has değişken tanımları
#include <PICPLC_877.inc>        ;temel tanımlar ve makrolar

_CONFIG _CP_OFF & _WDI_ON & _BODEN_OFF & _PWRTE_ON & _HS_OSC & _LVP_OFF & _DEBUG_OFF & _CPD_OFF

org      0x00                    ;Reset Vektörü

main

      initialize

scan

      get_inputs
;----- kullanıcı programı burada başlar -----
; 1. Basamak
ld_not M1.0
_set M0.0
_reset M0.1
_reset M0.3
_reset M0.4
_reset M0.5
_reset M0.6
_set M1.0

;2. Basamak
ld M0.0
and I0.5
_set M0.1
_reset M0.0

;3. Basamak
ld M0.1
and I0.5
and_not I0.6
_set M0.3
_reset M0.1

;4. Basamak
ld M0.3
and I0.4
_set M0.4
_reset M0.3

;5. Basamak
ld M0.4
TON_S 1,T1.2,.10

;6. Basamak
ld M0.4
and TON8 Q1
_set M0.5
_reset M0.4

;7. Basamak
ld M0.5
and I0.5
_set M0.6
_reset M0.5

;8. Basamak
ld M0.6
TON_S 2,T1.2,.10

;10. Basamak
ld M0.6
and TON8 Q2
_set M0.1
_reset M0.6

;9. Basamak
ld M0.0
or M0.5
out Q0.4

;10. Basamak
ld M0.2
out Q0.0

;11. Basamak
ld M0.3
out Q0.3
out Q0.0

;12. Basamak
ld M0.4
out Q0.1

;13. Basamak
ld M0.6
out Q0.2

;----- kullanıcı programı burada biter -----
send_outputs

goto    scan
end

```

Şekil 4. 44. Dağıtım istasyonuna ait kodun asm çıktısı

5. SONUÇ

Programlanabilir lojik denetleyiciler (PLC) günümüz endüstrisinin vazgeçilmez kontrol ve otomasyon cihazlarıdır. Günümüzde çok sayıda firma çeşitli özellikler ve fiyat yelpazesinde ticari PLC üretimi yapmaktadır. Her firma kendi PLC'si ile birlikte programlama yazılımını da satmaktadır. Bu durum ilk alımlarda maliyet artışına neden olmaktadır. Ticari PLC'lere alternatif ucuz maliyetli mikrodenetleyici temelli PLC'ler önerilmiştir. Bu PLC'lerden en kapsamlı olanı Uzam tarafından tasarlanan PIC16F877A ve PIC16F648A temelli PIC-PLC'lerdir [2,26]. Bu PLC donanımları için hâlihazırda programlama işlemi mikrodenetleyicinin kendi makine dili tabanlı makrolar ile sağlanabilmektedir.

Bu tez çalışmasında Uzam tarafından önerilen PIC mikrodenetleyicisi temelli PLC'ler için programlama arayüzü gerçekleştirilmiş ve bu program "Ladder PIC-PLC dönüştürücü" olarak adlandırılmıştır. Tasarlanan arayüz web sayfaları altında çalışabilir nitelikte olarak tasarlanmıştır. Bu açıdan platformdan bağımsızdır denilebilir.

Geliştirilen arayüz ile merdiven diyagramı olarak çizilen PLC kodu ilgili mikrodenetleyici koduna dönüştürülmektedir. Arayüz çeşitli ladder diyagramı örnekleri kullanılarak test edilmiştir. İlk olarak temel kodların çevrimini incelemek üzere; kontak ve röle temelli diyagramlar, zamanlayıcı, sayıcı, flip-flop, karşılaştırma ve aritmetik işlemleri içeren kodların dönüşümü dikkate alınmıştır. Testler sonucunda arayüzün temel kod dönüşümünü sorunsuz olarak yerine getirdiği gözlemlenmiştir.

Arayüz kullanılarak gerçek bir sistem için otomasyon kodunun üretilebileceği deneysel bir dağıtım istasyonu için gösterilmiştir. Bu uygulama için PIC16F877A temelli PIC-PLC giriş ve çıkışları kullanılarak dağıtım istasyonu deney setine bağlanmıştır. Tasarlanan ladder diyagramı kodu arayüz yardımıyla PIC koduna dönüştürülerek PIC-PLC'ye yüklenmiştir. Yapılan denemelerde üretilen kodun belirlenen kontrol senaryosunu eksiksiz yerine getirdiği gözlemlenmiştir.

Gerçekleştirilen arayüzün kullanıcı dostu olması için işlemler gayet basit olarak düzenlenmiştir. Kullanıcılara çizdikleri ladder diyagramlarını, saklama, yeniden açma ve değiştirme imkânları sunulmaktadır. Arayüze entegre edilen MPASM programı ile çizilen kod kolayca mikrodenetleyiciye yüklenecek makine koduna dönüştürülerek yüklenmeye hazır dosya elde edilebilmektedir.

Arayüzün çeşitli tarayıcılardaki yüklenme ve derleme performansları süre olarak karşılaştırılmıştır. Bu test işlemleri sonucunda arayüzün İnternet Explorer’da en verimli çalıştığı sonucuna ulaşılmıştır. Sonuç olarak bu tez kapsamında PIC-PLC donanımları için Ladder diyagramı programlamaya olanak sağlayan verimli, kullanımı kolay ve web tabanlı olan bir yazılım ortaya konmuştur.

KAYNAKLAR

- [1] Harmanda, A., 2011. 16 Bitlik Bir PIC Mikrodenetleyicisi ile Bir Programlanabilir Lojik Denetleyici Tasarımı ve Uygulaması.(Yüksek Lisans Tezi), Niğde Üniversitesi. Fen Bilimleri Enstitüsü, Niğde.
- [2] Uzam M., 2013. PIC16F877A Temelli PLC. Birsen Yayınevi, 322 s, İstanbul.
- [3] Uzam M., 2008a. PLC with PIC16F648A Microcontroller Part 1, Electronics World, 114 (1871), 21-25.
- [4] Uzam M., 2008b. PLC with PIC16F648A Microcontroller Part 2, Electronics World, 114 (1872), 29-35.
- [5] Uzam M., 2009a. PLC with PIC16F648A Microcontroller Part 3, Electronics World, 115 (1873), 30-34.
- [6] Uzam M., 2009b. PLC with PIC16F648A Microcontroller Part 4, Electronics World, 115 (1874), 34-40.
- [7] Uzam M., 2009c. PLC with PIC16F648A Microcontroller - Part 5, Electronics World, 115 (1875), 30-33.
- [8] Uzam M., 2009d. PLC with PIC16F648A Microcontroller - Part 6, Electronics World, 115 (1876), 26-30.
- [9] Uzam M., 2009e. PLC with PIC16F648A Microcontroller - Part 7, Electronics World, 115 (1877), 30-32.
- [10] Uzam M., 2009f. PLC with PIC16F648A Microcontroller - Part 8, Electronics World, 115 (1878), 30-32.
- [11] Uzam M., 2009g. PLC with PIC16F648A Microcontroller - Part 9, Electronics World, 115 (1879), 29-34.
- [12] Uzam M., 2009h. PLC with PIC16F648A Microcontroller-Part 10, Electronics World, volume 115, issue 1880, pp. 29-34..
- [13] Uzam M., 2009i. PLC with PIC16F648A Microcontroller-Part 11, Electronics World, 115 (1881), 38-42.
- [14] Uzam M., 2009i. PLC with PIC16F648A Microcontroller-Part 12, Electronics World, 115 (1882), 36-41.
- [15] Uzam M., 2009j. PLC with PIC16F648A Microcontroller-Part 13, Electronics World, 115 (1883), 42-44.
- [16] Uzam M., 2009k. PLC with PIC16F648A Microcontroller-Part 14, Electronics World, 115 (1884), 40-42.
- [17] Uzam M., 2009l. PLC with PIC16F648A Microcontroller-Part 14, Electronics World, 115 (1884), 40-42.
- [18] Uzam M., 2010a. PLC with PIC16F648A Microcontroller-Part 15, Electronics World, 116 (1885), 35-39.

- [19] Uzam M., 2010b. PLC with PIC16F648A Microcontroller Part 16, Electronics World, 116 (1886), 41-42.
- [20] Uzam M., 2010c. PLC with PIC16F648A Microcontroller Part 17, Electronics World, 116 (1887), 41-43.
- [21] Uzam M., 2010d. PLC with PIC16F648A Microcontroller Part 18, Electronics World, 116 (1888), 41-43.
- [22] Uzam M., 2010e. PLC with PIC16F648A Microcontroller Part 19, Electronics World, 116 (1889), 39-43.
- [23] Uzam M., 2010f. PLC with PIC16F648A Microcontroller Part 20, Electronics World, 116 (1890), 38, 40.
- [24] Uzam M., 2010g. PLC with PIC16F648A Microcontroller Part 21, Electronics World, 116 (1891), 40-41.
- [25] Uzam M., 2010h. PLC with PIC16F648A Microcontroller Part 22, Electronics World, 116 (1892), 40-42.
- [26] Uzam M., 2013. Building a Programmable Logic Controller with PIC16F648A Microcontroller, CRC Press, 371 s, USA.
- [27] Uzam M., 2009. PIC16F877A Mikro-Denetleyicisi Temelli PIC PLC Donanım, Endüstri & Otomasyon, 153 (1), 22-24
- [28] Rafat, Ö.F., 2010. PIC16F877 Mikrodenetleyicisi ile Bir PLC Tasarımı.(Yüksek Lisans Tezi), Selçuk Üniversitesi. Fen Bilimleri Enstitüsü, Konya.
- [29] Tongur, V., 2008. Atmega128 tabanlı PLC tasarımı.(Yüksek Lisans Tezi), Selçuk Üniversitesi. Fen Bilimleri Enstitüsü, Konya.
- [30] Erkol H., 2008. Mikrodenetleyici tabanlı PLC donanımı ve yazılımının gerçekleştirilmesi. (Yüksek Lisans Tezi), Erciyes Üniversitesi. Fen Bilimleri Enstitüsü, Kayseri.
- [31] Kitiş, Ş., 2007. PIC16F84 Mikrodenetleyicisi ile bir programlanabilir lojik denetleyici tasarımı ve uygulaması.(Yüksek Lisans Tezi), Niğde Üniversitesi. Fen Bilimleri Enstitüsü, Niğde.
- [32] Başkan, M. ve Gazi, M.E., 2012. PIC 16F877A Mikro denetleyicisinin PLC olarak kullanılması PIC LDR Programlama. Proje Tabanlı Mekatronik Eğitim Çalıştayı, 25-27 Mayıs, 2012, Çankırı Karatekin Üniversitesi Meslek Yüksekokulu. Çankırı.
- [33] Al Mashhadany, Y.I., 2012. Design and Implement of a Programmable Logic Controller (PLC) for Classical Control Laboratory. Intelligent Control & Automation, 3 (1), 44-49.
- [34] Kutlu, A., 2008. Çoklu Ürün Dağıtım Sisteminin PIC 16F877A Mikrodenetleyicisi Kullanarak Programlanması ve Uygulanması.(Yüksek Lisans Tezi), Kocaeli Üniversitesi. Fen Bilimleri Enstitüsü, Kocaeli.
- [35] Vural, Ö., 2008. PIC16F84 Mikrodenetleyicisi ile bir programlanabilir lojik denetleyici tasarımı ve uygulaması.(Yüksek Lisans Tezi), Niğde Üniversitesi. Fen Bilimleri Enstitüsü, Niğde.

- [36] <http://www.mikroe.com/picplc16/> 13.06.2014
- [37] Kurtulan, S., 2010. PLC ile Endüstriyel Otomasyon. Birsen Yayınevi, 496 s, İstanbul.
- [38] MEGEP, 2011. Temel PLC Sistemleri. MEGEP Elektrik Elektronik Teknolojisi, 79 s, Ankara.
- [39] Mirzaoğlu, İ. ve Sarıtaş, M., 2013. PLC VE SCADA kullanarak bir irmik üretim sisteminin otomasyonu, Elektrik Mühendisleri Odası, Ankara.
- [40] Özkan A.O., Özcan M., 2004. Otomasyon Sistemlerinde PLC Uygulamaları. Nobel Yayın Dağıtım, 316 s, Ankara.
- [41] Şahin H., Dedeoğlu K.S., 2012. Yeni Başlayanlar için Mikro C ile PIC Programlama. Altaş Yayın, 224 s, İstanbul.
- [42] Şahin H., Dayanık A., Altınbaşak C., 2013. PIC Programlama Teknikleri ve PIC 16F877A. Altaş Yayın, 456 s, İstanbul.
- [43] Anonim, 2014. Devices Included in this Data Sheet <http://ww1.microchip.com/downloads/en/DeviceDoc/30292D.pdf> -(03.06.2014).
- [44] Anonim, 2014. Microchip. [http://ww1.microchip.com/downloads/en/DeviceDoc/8-bit_PIC_MCU_Solutions\(DS39630D_CN\).pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/8-bit_PIC_MCU_Solutions(DS39630D_CN).pdf) - (03.06.2014).
- [45] Birant, D., Birant, K.U., Kasap, M., Kut, A., 2002. Web Tabanlı Bilet Satış Otomasyon Sistemi, VIII Türkiye’de İnternet Konferansı, İstanbul.
- [46] MEGEP, 2007. Bilişim Teknolojileri Modülü, Motorlar, Milli Eğitim Bakanlığı, Ankara,
- [47] Yılmaz, H. ve Sarıtaş, M., 2013. Ekmek Üretim Hattında PLC Cihazının Kullanılması, Elektrik Mühendisleri Odası, Ankara.
- [48] <http://eee.balikesir.edu.tr/sinavlardersler/PLC%20Kitabi.pdf> 13.06.2014
- [49] <http://www.festo-didactic.com/tr-tr/egitim-sistemleri/> 11.06.2014
- [50] John K.H., Tiegelkamp M., 2010, IEC 61131-3: Programming Industrial Automation Systems. Springer Science & Business Media, 396 s, New York.

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Serhat KILIÇARSLAN
Doğum Tarihi ve Yeri : 26.03.1986 - İSTANBUL
Medeni Hali : Evli
Yabancı Dili : İngilizce
Telefon : 536 2264343
e-mail : serhat.kilicarslan@gop.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lisans	Marmara Üniversitesi, İstanbul	2008
Lise	GEML (Bilgisayar Donanım), İstanbul	2003

İş Deneyimi

Yıl	Yer	Görev
2011-Devam .	Gaziosmanpaşa Üni.,Tokat	Öğretim Görevlisi
2008-2011	İstanbul	İstanbul Büyük Şehir Belediyesi (Usta Öğretici-Yazılımcı)
2010-2011	İstanbul	Demtaş- Yazılımcı
2007-2008	İstanbul	Kuveyt Türk Katılım Bankası- Yazılımcı