



T.C.  
MALTEPE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

**PROGRAMLANABİLİR LOJİK KONTROLÖR (PLC)  
EĞİTİMİ İÇİN BİR SİMULATÖR TASARIMI**

**ENGİN OĞUZAY**

Yüksek Lisans Tezi

Tez Danışmanı

**Prof. Dr. A. Mesut RAZBONYALI**

İSTANBUL – 2007



**T.C.  
MALTEPE ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR ANABİLİM DALI**

**PROGRAMLANABİLİR LOJİK KONTROLÖR (PLC)  
EĞİTİMİ İÇİN BİR SİMULATÖR TASARIMI**

**YÜKSEK LİSANS (MASTER) TEZİ**

**Hazırlayan  
ENGİN OĞUZAY**

**Tez Danışmanı  
Prof. Dr. A. Mesut RAZBONYALI**

**İstanbul – 2007**

Bu tez çalışması, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... / ..... / ..... tarih ve ..... / ..... sayılı kararıyla oluşturulan jüri tarafından *Bilgisayar Mühendisliği Yüksek Lisansı Tezi* olarak kabul edilmiştir.

JÜRİ

Prof. Dr. A. Mesut RAZBONYALI

Danışman

Prof. Dr. İlhami YAVUZ

Üye

Prof. Dr. E. Murat ESİN

Üye

## ÖZET

Yüksek Lisans Tezi, PLC ( Programlanabilir Lojik Kontrolör) Eğitimi İçin Bir Simulatör Tasarımı, T.C. Maltepe Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı

Kıyasıya rekabetin yaşandığı günümüzde, endüstriyel kuruluşların varlıklarını sürdürebilmeleri için verimli, esnek ve üretim maliyetlerinin düşük olması gereklidir. Klasik kontrol ve kumanda sistemlerin kullanımıyla bu hedefe ulaşmak oldukça zor görülmektedir. Endüstriyel otomasyon sektöründe yerlerini korumak isteyen işletmelerde Programlanabilir Lojik Kontrolörlerin kullanımı her geçen gün artmaktadır. Başlangıçta tek bir makineyi kontrol elemanı olarak üretilen PLC'ler günümüz endüstrisinin kullandığı çok amaçlı bir kontrol elemanıdır. PLC'ler diğer kontrol elemanları ve bilgisayar ağ sistemleriyle uyum içinde çalışmaktadır.

PLC'ler yalnızca röle sistemleri değil aynı zamanda ayırık kontrol cihazlarına karşı başarılı bir şekilde rekabet ederek gelişmesini sürdürmüştür. PLC uygulamalarındaki eğitim görme gereksinimi teknik okul, üniversite ve endüstri alanlarında artmaktadır. Bu tezin amacı, PLC'nin çalışma mantığı ve bir PLC Simulatörü ile birlikte yazılımsal ve görsel olarak PLC'nin bir eğitim simulatörü olarak çalışmasını göstermektir.

Tezin birinci bölümünde; PLC'nin genel çalışma prensibi ve genel uygulama alanları örneklerle açıklanmıştır. İkinci bölümde PLC tarihçesi, avantajları ve PLC elemanları; Üçüncü bölümde, PLC'lerin işlevleri, PLC bellek yapısı ve bu bölgelere erişim teknikleri açıklanmıştır. PLC'lerin programlanmasına dördüncü bölümde değinilmiştir. Beşinci bölümde bit lojik komutları açıklanmıştır. Son bölümde ise; yazılımı yapılan PLC Eğitim Simulatörü uygulamalarıyla beraber anlatılmıştır.

Bu tez 2007 yılında yapılmış ve 85 sayfadan oluşmaktadır.

**Anahtar Kelimeler** : Programlanabilir Lojik Kontrolörler, Kontaktör, Ladder Diyagram, Simulatör, Bit Lojik

## **ABSTRACT**

Master Thesis, For PLC (Programmable Logic Controller) Education A Simulator Design. T.C. Maltepe University, Graduate School of Natural and Applied Sciences, Department of Computer Engineering.

Today, in order to survive in a world of ruthless competition, it is vitally important that the industrial establishments should be profitable and flexible and that the cost of production be as low as possible. It seems rather difficult to reach this goal with the use of classical and command systems. In the establishments which keep their positions in the industrial automation sector, the use of programmable logic control is on the increase day by day. The PLC's which were originally produced as the only element to control a machine are the multi-purpose control elements of today. PLC's have been used harmonously with the other control and network systems.

PLC's, which are not only used as role systems, but also used against the separate control instruments, have made a lot of progress being in competition with the other systems. The application of PLC has been on the increase in the fields of training pertaining to technical schools, universities and the industrial establishments. The purpose of this is to point out thesis, the mentality of PLC's operating systems and the use of PLC Simulator for the wording and visual purposes.

In the first part of the thesis, the general working principles of PLC's and the general application areas have been explained with examples. In the second part of the thesis, the short history and different advantages of PLC elements have been explained. In the third part of the thesis, the functions of PLC's, the memory system of PLC and the techniques of their communicating areas have been explained. The programming of PLC's has been mentioned in the fourth part of the thesis. In the fifth part of the thesis, bit logic commands have been explained. In the last part of the thesis, the PLC's training Simulator has been explained with its applications.

This thesis has been completed in 2007 and consists of 85 pages.

**Keywords:** Programmable Logic Controller, Contactor, Ladder Diagram, Simulator

## TEŐEKKÜR

Bu tez konusunu seçmemde beni yönlendiren, çalışmalarım sırasında tecrübelerinden, bilgilerinden istifade ettiğim, gerekli kaynakların sağlanmasında yardımcı olan, desteğini esirgemeyen danışman hocam sayın Prof. Dr. A. Mesut RAZBONYALI'ya, aynı ilgi ve alakayı gösteren, destek ve yardımlarını esirgemeyen Maltepe Üniversitesi Elektronik Mühendisliği Bölüm Başkanı sayın Prof. Dr. Murat ESİN'e, Fen Bilimleri Enstitüsü Müdürü sayın Prof. Dr. İlhami YAVUZ'a ve Meslek Yüksek Okulu Müdürü Prof. Dr. Şaban EREN başta olmak üzere tüm çalışma arkadaşlarıma, aileme ve çalışmalarım sırasında emeği geçen herkese teşekkürlerimi sunarım.



# İÇİNDEKİLER

ÖZET	I
ABSTRACT	II
TEŞEKKÜR	IV
İÇİNDEKİLER	V
KISALTMALAR	IX
ŞEKİLLER DİZİNİ	X
TABLolar DİZİNİ	XII
<b>1. PROGRAMLANABİLİR LOJİK KONTROLÖR (PLC)</b>	
1.1. Giriş ve Amaç	1
1.2. PLC Genel Uygulama Alanları	3
<b>2. PLC ELEMANLARI</b>	
2.1. PLC Tarihçesi	7
2.2. PLC Neden Doğdu?	9
2.3. PLC Mantiğı ile Kontaklı Devre Mantiğı Arasındaki Fark Nedir?	10
2.4. PLC Kullanım Avantajları	10
2.5. PLC Elemanları	12
2.5.1. Donanım (Hardware)	13
2.5.2. Yazılım (Software)	14
2.5.3. Algılayıcılar ( Sensörler )	14
2.5.4. İş Elemanları	15
2.5.5. Programlayıcı	15
<b>3. PLC'LERİN İŞLEVLERİ ve BELLEK YAPISI</b>	
3.1. PLC Çeşitleri	16
3.2. PLC'lerin İşlevleri	16
3.2.1. PLC'lerin Temel İşlevleri	16
3.2.2. Endüstriyel Otomasyon Devrelerinde PLC'nin Tercih Edilmesinin Nedenleri	17

3.2.3.	PLC İle Bilgisayarlı Kontrol Sistemlerinin Karşılaştırılması	17
3.3.	Genel Olarak PLC Yapısı	18
3.3.1.	Merkezi İşlem Birimi Ve Bellek	19
3.3.1.1.	Tarama	19
3.3.1.2.	Bellekte Yer Alan Bilgiler	23
3.3.2.	Giriş / Çıkış Birimi	23
3.3.2.1.	Giriş Birimi	24
3.3.2.2.	Çıkış Birimi	25
3.4.	PLC'lerin Programlanması	26
3.4.1.	Programlama Birimi	26
3.5.	PLC Bellek Yapısı	26
3.5.1.	Giriş	26
3.5.2.	Yarıiletken Bellekler	26
3.5.2.1.	Ram Bellek	26
3.5.2.2.	Rom Bellek	27
3.5.2.3.	Prom Bellek	27
3.5.2.4.	Eprom Bellek	28
3.5.2.5.	Eeprom Bellek	28
3.5.2.6.	Novram Bellek	29
3.6.	Bellek Kapasitesi	29
3.7.	PLC Bellek Haritası	30
3.8.	Veri Alanı	31
3.8.1.	Değişken Bellek	32
3.8.2.	Giriş İmaj Register	32
3.8.3.	Çıkış İmaj Register	32
3.8.4.	Dahili Bellek Bitleri	33
3.8.5.	Özel Bellek Bitleri	33
3.8.6.	Ardışıl Kontrol Röle Bellek Alanı	34
3.8.7.	Lokal Bellek Alanı	34
3.9.	Bellek Ve Programlama Koşulları	34
3.10.	Analog Giriş / Çıkışlar	35
3.10.1.	Analog Girişler	35

3.10.2.	Analog Çıkışlar	35
3.11.	Digital Giriş / Çıkışlar	35
3.11.1.	Giriş	35
3.11.2.	Dijital Girişler	36
3.11.3.	Dijital Çıkışlar	37
<b>4. PLC'LERİN PROGRAMLAMASI</b>		
4.1.	PLC'lerin Programlama Özellikleri Ve Yöntemleri	38
4.2.	Programlama Dili	38
4.3.	PLC Programları Çalışma İlkesi	38
4.4.	Programlayıcı Birimi	38
4.5.	Programlama	39
4.5.1.	Programlama Yapıları	39
4.6.	PLC'lerin Programlama Modları	40
4.7.	PLC'nin Kontrol Sistemindeki Yeri ve PLC İle Kontrol Sistemlerinin Oluşturulması	42
4.8.	PLC Çalışması	43
4.8.1.	Donanım	43
4.8.1.1.	Donanım Elemanları	44
4.8.2.	Yazılım	45
<b>5. BİT LOJİK KOMUTLARI</b>		
5.1.	Giriş	46
5.2.	Normalde Açık Kontak	46
5.3.	Normalde Kapalı Kontak	47
5.4.	Çıkış Rölesi	48
5.5.	Değil (Not) Kontak	51
5.6.	Set Rölesi	51
5.7.	Reset Rölesi	52

## 6. PLC EĞİTİM SİMULATÖRÜNÜN TASARIM ve UYGULAMASI

6.1.	Giriş	53
6.2.	Ön Çalışmalar ve Sistem Analizi	54
6.2.1.	PLC Eğitim Simulatör Birimleri	55
6.2.1.1.	Input	56
6.2.1.2.	Output	56
6.2.1.3.	Komponent Sürükle-Bırak Alanı (Drag and Drop)	56
6.2.1.4.	Ladder Diyagram Alanı	56
6.2.1.5.	Denklem Kurma Alanı	56
6.3.	Programın Çalışma Mantığı	57
6.4.	Desteklenen Komut, Fonksiyon ve Prosedürler	58
6.4.1.	Kullanılan Diziler	58
6.4.2.	Kullanılan Prosedürler	59
6.4.3.	Bir Örnek ile Procedure ve Function'ların çağırılması	70
6.5.	PLC Eğitim Simulator Programının Genel Yapısı	71
6.6.	Toolbar'dan Bir Komponentin Sökülmesi	72
6.7.	Toolbar'dan Alınan Bir Komponentin Çizim Alanına Yerleştirilmesi	73
6.8.	Yerleşik Komponentler Üzerindeki İşlemler	74
6.9.	DENEMELER ( Uygulamalar )	75
<b>7.</b>	<b>SONUÇ</b>	<b>84</b>
	<b>KAYNAKLAR DİZİNİ</b>	<b>85</b>
	<b>ÖZGEÇMİŞ</b>	<b>87</b>

## KISALTMALAR

Kısaltma	İngilizce	Türkçe
<b>PLC</b>	Programmable Logic Controller	Programlanabilir Lojik Kontrolör
<b>STL</b>	Statement List	Komut Listesi
<b>FBD</b>	Function Block Diagram	Fonksiyon Blok Diyagramı
<b>I/O</b>	Input / Output Unit	Giriş / Çıkış Birimi
<b>NEMA</b>	National Electric Man.Association	Uluslararası Elektrik Üretim Kur.
<b>IC</b>	Integrated Circuit	Entegre
<b>PROM</b>	Programmable PROM	Programlanabilir ROM
<b>EPROM</b>	Erasable Programmable ROM	Silinebilir PROM
<b>EEPROM</b>	Electronically Erasable PROM	Elektronik Silinebilir PROM
<b>CSF</b>	Control System Flow Chart	Lojik Kapı Gösterimi
<b>LAD</b>	Ladder Diagram	Merdiven Şeması

## ŞEKİLLER DİZİNİ

Şekil	Sayfa
Şekil 1.1 : PLC Hardware(Donanım) Konfigürasyonu	3
Şekil 2.1 : PLC'nin Basit Yapısı	7
Şekil 2.2 : PLC'nin elemanları	12
Şekil 2.3 : PLC kumanda sisteminin yapısı	14
Şekil 3.1 : Tipik Bir PLC'nin Bölümleri	18
Şekil 3.2 : Tarama döngüsü	20
Şekil 3.3 : Ladder diyagram programlamada tarama sırası	22
Şekil 3.4 : 220 V AC gerilimle uyarılan bir giriş birimi	24
Şekil 3.5 : 24 V DC gerilimle uyarılma bir giriş birimi	24
Şekil 3.6 : Kontak çıkışlı devre	25
Şekil 3.7 : Transistör çıkışlı devre	25
Şekil 3.8 : 1Kbayt'lık bellek blok diyagramı	30
Şekil 3.9 : PLC bellek haritası	31
Şekil 3.10 : Alternatif gerilim uyarlamalı dijital giriş blok diyagramı	36
Şekil 3.11 : Tipik dijital çıkış blok diyagramı	37
Şekil 4.1 : PLC programlarının çalışma ilkesi	38
Şekil 4.2 : PLC blok şeması	42
Şekil 4.3 : PLC Genel Blok Şeması	45
Şekil 5.1 : Normalde açık kontak	46
Şekil 5.2 : Normalde kapalı kontak	47
Şekil 5.3 : Çıkış rölesi	48
Şekil 5.4 : PLC ladder diyagram programı	48
Şekil 5.5: Değil kontağı	51
Şekil 5.6 : Set rölesi	51
Şekil 5.7 : Reset rölesi	51
Şekil 6.1 : PLC Eğitim Simulatorü Ekranı	52
	55

<b>Şekil 6.2</b> : PLC Eğitim Simulatorü Ekranı Bileşenleri	56
<b>Şekil 6.3</b> : PLC Eğitim Simulatorü Ekranında Seri-Paralel Bağlantı	57
<b>Şekil 6.4</b> : PLC Eğitim Simulatörünün altyapısını oluşturan prosedürlerin ekran görüntüsü	60
<b>Şekil 6.5</b> : Çalıştır prosedürünün akış şeması	62
<b>Şekil 6.6</b> : Yerleştir prosedürünün akış şeması	66
<b>Şekil 6.7</b> : PLC Eğitim Simulatorü ekranında network(basamak) gösterimi(yeşil renk)	68
<b>Şekil 6.8</b> : PLC Eğitim Simulatorü ekranında boşlukların gösterimi (kırmızı renk)	68
<b>Şekil 6.9</b> : SatırDolu prosedürü ile ekran üzerindeki network sayısının bulunması	69
<b>Şekil 6.10</b> : Çizim alanında dizilerin gösterimi	72

## TABLULAR DİZİNİ

<b>Tablo</b>	<b>Sayfa</b>
<b>Tablo 4.1 :</b> PLC'nin kontrol devresindeki yeri	42
<b>Tablo 5.1 :</b> Tüm komutlarda kullanılan veri tipleri ve özellikleri	47
<b>Tablo 6.1 :</b> PLC Eğitim Simulatörü sisteminden beklentiler	59



## **1. PROGRAMLANABİLİR LOJİK KONTROLÖR (PLC)**

PLC'lerde kullanılan programlama dilleri sürekli olarak gelişme kaydetmektedir. Bunun yanında bazı temel yazılım türleri uzun süreden beri kullanılmaktadır. Programlanabilir kontrolörün doğuşu ile birlikte ladder diyagram(merdiven) programlama formatı kullanılmaya başlandı. PLC'ler ladder diyagram programlama dilinin yanında STL (statement list) komut dili ve FBD (function block diyagram) ile programlanabilir.

Programlanabilir lojik kontrolör (PLC), bünyesine yazılan programla zaman, sayma, sıralı kontrol, aritmetik, veri yönlendirme, iletişim vb. kontrol fonksiyonlarını yapabilen, basit anlamda endüstri için tasarlanmış bir bilgisayardır. Öncelikle klasik kumanda panolarında bulunan röle, zaman rölesi vb. elemanların görevini üstlenmek amacıyla tasarlanan PLC'lerle günümüzde karmaşık kontrol sistemlerini gerçekleştirmek mümkündür.

### **1.1 Giriş ve Amaç**

PLC, donanım açısından bilgisayara benzemesine rağmen gürültü sinyallerinden etkilenmemesi, modüler yapıda olması, giriş-çıkış modüllerin montajının basit, programlama dilinin kolay olması ve karmaşık kontrol işlemlerini gerçekleştirebilmesi özelliği ile proses kontrol uygulamaların vazgeçilmez elemanıdır.

19 .yüzyılın başında üretim süreçlerinde elektrikli anahtarlama sistemleri yoğun olarak kullanılmaya başlandı. Mekanik sisteme yön veren otomasyon sistemlerinde elektrikli kontrol devreleri ile donatıldı. Bu devrelerde kullanılan temel elektrik elemanı şu an bile yaygın olarak kullanılan röledir. Rölenin bir kontrol sisteminde üstlendiği başlıca görevler şunlardır:

- Bellek elemanı olarak işlev görmek.
- Düşük güç seviyelerinden yüksek güç seviyelerine geçmek.

- Kontak çoğaltmak.
- Bobin ve kontak tarafındaki gerilimi birbirinden izole etmek.

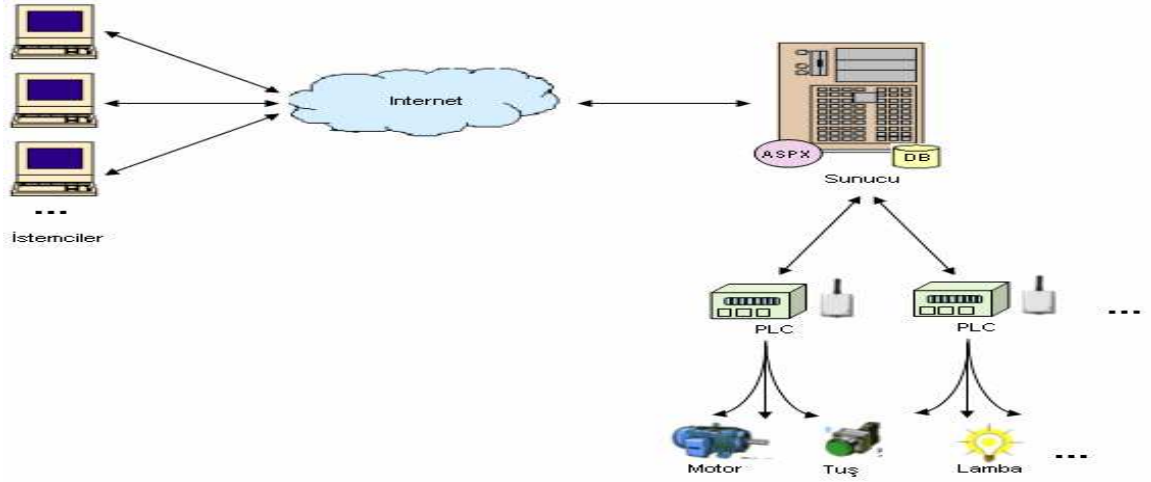
Bu özellikleri sayesinde röleler otomasyon sistemlerinde gerekli lojik bağlantıların yapılabilmesini sağlamaktadır. Röleli kontrol sistemlerinde de kontrol devresi sadece bir kontrol işini yapabilecek şekilde gerçekleştirilir. Kontrol sisteminde yapılacak değişiklik için kablo bağlantılarının yeniden yapılması gerekir.

PLC; programlanabilir lojik kontrol sistemlerinin kısaltması olarak literatüre yerleşmiştir. Günümüzde PLC'ler endüstriyel uygulamaların pek çok alanında kullanım yeri bulmuştur. Ancak otomasyon süreçlerinde genellikle pnömatik kontrolün işlemci grubu olarak kullanılmaktadır. Bir kontrol sürecinde makinenin içinde bulunduğu durum algılayıcılar tarafından algılanmakta ve bu bilgiler PLC giriş modülüne alınmaktadır. PLC'de işlenen bilgiler, çıkış modülü yardımı ile iş elemanlarını tetikleyecek kumanda elemanlarını enerjilemektedir.

PLC'lerin kontrol sistemlerinde kullanılması sistem tasarımı, işletmeye alınması ve bakım çalışmalarında büyük kolaylık getirmektedir. Programlanabilen lojik kontrol olarak tanımlanabilen PLC, elektriksel olarak kontrol edilen sistemlerde otomasyona geçişi sağlamak için geliştirilmiştir. Tüm veri giriş- çıkışlarını istenilen şekilde programlayarak daha az malzeme ve işçilikle kontrolü sağlamaktır.

PLC içerisindeki operasyonlar çeşitli lojik fonksiyonlarla ifade edilirler. Yani tüm değerler "0" ve "1" lerden oluşmaktadır. Dolayısıyla PLC içerisinde kullanılacak değerler de Binary ve BCD gibi ikili sayı sistemleridir. PLC giriş ve çıkışlarındaki elemanları kontrol gerilimi genellikle 24 VDC olur. Yani bir giriş sinyalinin "1" olarak algılanması için +24VDC olması gerekmektedir. Bu sinyaller giriş ve çıkışlardaki optokublörler vasıtasıyla +5VDC'ye dönüştürülür.[1] Çünkü PLC içerisinde kullanılan voltaj daima +5VDC'dir. Farklı giriş ve çıkış gerilimleri için değişecek şey sadece optokublörler olacaktır.

PLC uygulamalarının ilk aşaması işin projelendirilmesidir. Kullanılacak tüm giriş (butonlar, anahtarlar, switchler v.b.) ve çıkış sinyalleri ( kontaktörler, lambalar, valfler v.b.) belirlenir. Bunların sayısına ve kumanda gerilimlerine göre PLC ünit ve gerekli diğer modüller seçilir. Yani yapılacak ilk işlem “Hardware” konfigürasyonun hazırlanmasıdır. “Hardware” konfigürasyonu Şekil 1.1’de gösterilmiştir.



Şekil 1.1 : PLC Hardware(Donanım) Konfigürasyonu [ 2 ]

## 1.2 PLC Genel Uygulama Alanları

PLC uygulamaları iki sınıfta toplanabilir: Genel ve endüstriyel uygulamalar hem ayrı hem de proses sanayilerinde mevcuttur. PLC’lerin doğduğu sanayi olan otomotiv, en büyük uygulama alanı olmayı sürdürmektedir. Yiyecek işleme ve hizmetleri gibi sanayilerde şu an dünyada gelişen alanlar arasında PLC’lerin kullanıldığı 5 genel uygulama alanı vardır. Tipik bir kurulum, kontrol sistemi sorununa çözümü, bunların bir ya da daha çoğunu içererek bulunur. Bu beş alan şunlardır:

### a. Sıra (Sequence) Kontrol

PLC’lerin en büyük ve en çok kullanılan ve “sıralı çalışma “ özelliğiyle röleli sistemlere en yakın olan uygulamasıdır. Uygulama açısından, bağımsız makinalarda ya da makine hatlarında, konveyör ve paketleme makinalarında ve hatta modern asansör denetim sistemlerinde bile kullanılmaktadır.

## **b. Hareket Kontrolü**

Bu doğrusal ve döner hareket denetim sistemlerinin PLC' de tümleştirilmesidir ve servo adım ve hidrolik sürücülerde kullanılabilen tek ya da çok eksenli bir sistem denetimi olabilir. PLC hareket denetimi uygulamaları, sonsuz bir makine çeşitliliği içerir. (örn. metal kesme, metal şekillendirme, montaj makinaları) ve süreç sanayi uygulamalarında koordine edebilirler. Bunlara örnek olarak; film, kauçuk ve dokunmamış kumaş tekstil sistemleri gibi, ağıla ilgili süreçler verilebilir.

## **c. Süreç Denetimi**

Bu uygulama PLC'nin birkaç fiziksel parametreyi (sıcaklık, basınç, debi, hız, ağırlık vb gibi) denetleme yeteneğiyle ilgilidir. Bu da bir kapalı çevrim denetim sistemi oluşturmak için, analog I/O gerektirir. PID yazılımının kullanımıyla PLC, tek başına çalışan çevrim denetleyicilerinin (single loop controllers) işlevini üstlenmiştir. Diğer bir seçenek de her ikisinin en iyi özelliklerini kullanarak PLC ile kontrolörlerin tümleştirilmesidir. Buna tipik örnekler de plastik enjeksiyon makinaları, yeniden ısıtma fırınları ve bir çok diğer yığın denetimi (batch-control) uygulamasıdır.

## **d. Veri Yönetimi**

PLC ile veri toplama, inceleme ve işleme son yıllarda gelişmiştir. İleri eğitim setleri ve yeni PLC'lerin genişletilmiş bellek kapasiteleriyle sistem, denetlediği makine veya proses hakkında veri yoğunlaştırıcı olarak kullanılabilir. Sonra bu veri, denetleyicinin belleğindeki referans veri ile karşılaştırılır ya da inceleme ve rapor alımı için başka bir aygıtta aktarılabilir.

Bir PLC Programlama ortamı;

- Değişken adı ve tipi tanımlamayı
- Değişkenleri ilişkisel ve mantıksal olarak bağlamayı (Binary, Analog)
- Simulasyon imkanlarını
- PLC-PC haberleşme protokolünü
- PLC'deki mevcut programı okuma veya kayıt yapabilmeyi

- PLC'deki programı eşzamanlı gösterebilmeyi

Kullanılan hafıza alanlarının ve giriş-çıkış etiketlerinin oluşturulmasını sağlar.

## 2. PLC ELEMANLARI

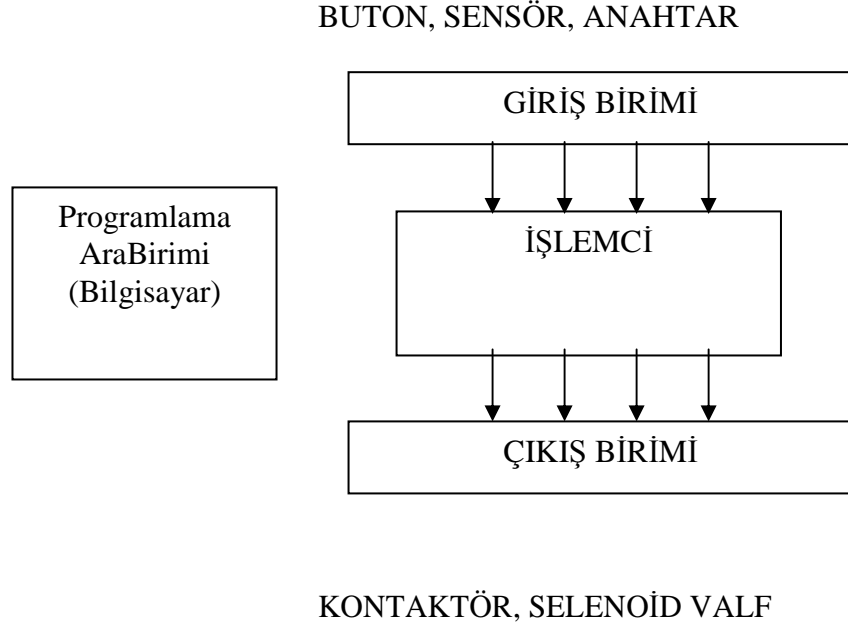
Programlanabilir Lojik Kontrolör (**Programmable Lojik Controller, PLC**) endüstriyel otomasyon sistemlerinin kumanda ve kontrol devrelerini gerçeklemeye uygun yapıda giriş / çıkış birimleri ve iletişim arabirimleri ile donatılmış, kontrol yapısına uygun bir sistem programı altında çalışan bir endüstriyel bilgisayardır. Başlangıçta, röleli kumanda sistemlerinin yerine kullanılmak üzere düşünülmüş ve ilk ticari PLC 1969 yılında Modicon firması tarafından geliştirilmiştir.[3] O yıllarda röleli kumanda devreleri yerine kullanılmak üzere geliştirilen bu aygıt yalnız temel lojik işlem komutları ile işlem yapabilmekteydi. İlk ticari PLC'nin endüstride başarı ile uygulamasından sonra Allen Bradley, General Electric, Siemens, Westinghouse gibi firmalar orta maliyette yüksek performanslı PLC'ler üretmişlerdir.

Günümüzde üretilen PLC'ler ise lojik temelli işlemlerin dışında ilave olarak aritmetik ve özel matematiksel işlemlerin yapılmasını sağlayan komutlar içermektedir. Komut kümesinin gelişmesi ile daha karmaşık kumanda ve kontrol işlemleri yapılabilir. PLC'lerin en yaygın olarak kullanıldığı alanlar endüstriyel otomasyon sistemlerinin kumanda devreleridir. Günümüzde bu tür devrelerin yerini aynı işlevi sağlayan PLC'li kumanda sistemleri almıştır .

PLC'ler endüstriyel otomasyon sistemlerinde doğrudan kullanıma uygun özel giriş ve çıkış birimleri ile donatılmışlardır. Girişe basınç, seviye, sıcaklık algılayıcıları ve buton gibi iki değerli lojik bilgisi taşıyan elemanlar, çıkışa ise kontaktör, valf gibi kumanda devre elemanlarının sürücü elemanları doğrudan bağlanabilir.

Şekil.2.1'de görüldüğü gibi bir PLC temel olarak;

- Bir sayısal işlemci bellek,
- Giriş ve çıkış birimleri,
- Programlayıcı birimi,
- Besleme güç kaynağı gibi temel kısımlardan oluşmaktadır.



**Şekil 2.1 :** PLC'nin Basit Yapısı

## 2.1 PLCTarihçesi

PLC, “Programlanabilir Lojik Kontrolör” İngilizce kelimelerinin baş harflerinin PLC alınarak kısaltılması ile oluşur. 30 yıl önce sanayi uygulamalarında kullanılmaya başlanmış ve son 20 yıldır FESTO, MITSUBISHI, SIMATIC, TOSHIBA, SIEMENS, MERLİN GERİN, GENERAL ELECTRIC gibi firmaların, tabanı ve programlama mantığı birbirine çok yakın, kendi aralarında değişik üstünlükler ile ayrılan PLC sistemlerini geliştirmeleri ile, otomatik kontrol sistemlerinde, hız, kontrol, güvenlik, ürün kalitesi yanı sıra, yeni bir ürün imali için kumanda devrelerinin yeniden oluşturulması montajı ve bağlantıları yerine sadece PLC programlama ile giderilmesi çok büyük bir avantaj sağlamıştır. Bu da PLC tabanlı kontrol sistemlerinin endüstriyel otomasyon, devrelerinden vazgeçilmez bir sistem olarak kullanılmasını ve her geçen gün yeni özellikler ile güncelleştirilmesi gereğini doğurmuştur.

Programlanabilir lojik kontrolörlerin çıkışı 1970'li yılların başlarına dayanır. İlk kumanda kontrolörleri bağlantı programlamalı cihazlardı. Bu cihazların fonksiyonları, lojik modüllerin birbirine bağlantı yapılarak birleştirilmesi ile gerçekleştiriliyordu. Bu cihazlarla çalışmak hem zordu, hem de kullanım ve programlama olanakları sınırlıydı. Bugünkü PLC'ler ile karşılaştırıldığında son derece basit cihazlardı. PLC'lerin ortaya çıkarılma amacı, röleli kumanda sistemlerinin gerçekleştirdiği fonksiyonların mikroşemcili kontrol sistemleri ile yerine getirilebilmesidir.

Programlanabilir Lojik Kontrolörler (PLC) otomasyon devrelerinde yardımcı röleler, zaman röleleri, sayıcılar gibi kumanda elemanlarının yerine kullanılan mikroşemci temelli cihazlardır. Bu cihazlarda zamanlama, sayma, sıralama ve her türlü kombinasyonel ve ardışık lojik işlemler yazılımla gerçekleştirilir. Bu nedenle karmaşık otomasyon problemlerini hızlı ve güvenli bir şekilde çözmek mümkündür.

Programlanabilir lojik kontrolör çalışması ilk kez 1968 yılında General Motors'ta çalışan bir grup mühendis tarafından aşağıdaki kriterler konularak başlatılmıştır:

- Fabrika ortamında kolayca programlanabilmesi ve programı değiştirilebilmeli
- Bakımı ve tamiri kolay olmalı, tercihen modüler olmalı.
- Fabrika ortamında güvenilir ve röle eşdeğerinden daha küçük boyutta olmalı
- Maliyet düşük olmalı.

1969 yılında, klasik kumandanın elektronik karşılığı olan ilk PLC yapıldı. 1Kbaytlık belleğe sahip, 128 giriş/çıkışlı, lojik komutları işleyebilen PLC üretildi. Çok kısa sürede PLC'lerin işleyebildikleri komut setleri basit lojik komutlardan zamanlayıcı, sayıcı ve gelişmiş matematiksel fonksiyonları da kapsayacak şekilde gelişti.

İlk mikroşemcili PLC, 1977 yılında Allen-Bradley firması tarafından piyasaya sürüldü. 1978 yılında ise dört yıllık bir çalışmanın ürünü olarak NEMA (National Electric Manufactureres Association) kuruluşu tarafından ilk PLC'ler piyasaya sürüldü.



8080 mikroişlemcisine dayanan PLC bit lojik komutlarını yüksek hızda işleyebilmek için ilave işlemci kullanmaktaydı. Mikroişlemcinin üretilmesiyle her türlü otomasyon sistemlerin gerçekleştirilmesinde PLC gücü artmaya başladı. Yüksek seviyede yazılan programlar, bu işle uğraşanlar için daha kolay anlaşılır olmaya başladı.

1980'li yıllarla birlikte PLC endüstrisinde yeni teknolojik ilerlemeler kaydedildi. Çok az (ona kadar) sayıda röle kullanan sistemler yerine kullanılabilir düşük fiyatlı PLC lerin piyasaya sürülmesi, Mikroişlemci akıllı giriş/çıkışla dağıtılmış işleme (ASCII işletim arabağları) ısı ve basınç ölçen alıcıların doğrudan PLC'ye bağlanabilmesine olanak sağlayan arabağlar donanım açısından kaydedilen ilerlemelerle birer örnek olarak sayılabilirler.

Küçük PLC'lerin 1983 yılında Japon firmaları tarafından piyasaya sürülmesiyle küçük kontrolör pazarı hızlı şekilde gelişim gösterdi. Günümüzde PLC'deki gelişmeler bilgisayarlarla aynı hızda gitmektedir. Tıpkı bilgisayarlar gibi network (1985'den günümüzde) sistemiyle çalışabilen PLC sistemleri fabrika ortamında konum kontrol, nümerik kontrol uygulamalarında vazgeçilmez elemanlardır.

## **2.2 PLC Neden Doğdu?**

İlk PLC 1960'lı yıllarda Amerikan otomobil endüstrisinde çalışan mühendislerin yeni bir üretim tekniği geliştirme çalışmaları esnasında geliştirilmiştir. Bu buluştan önce, kontrol sistemleri tek bir üretim işleminin yapılmasında ayrı ayrı geliştirilmiş otomasyon sistemleri ile bir cihazı çalıştırıyorlardı. Bunlar elektromekaniki zamanlayıcılar, sayıcılar, başlama durdurma anahtarları, röleler diğer farklı yapıdaki cihazlar elle kumandalı işlem kontrolleri üzerindeki belli bir seviyede geliştirilmişlerdir.

### **2.3 PLC Mantığı ile Kontaktlı Devre Mantığı Arasındaki Fark**

PLC'ler için geliştirilmiş olan programlama dilleri, kontaktlı (röleli) kumanda devreleri tasarımcılarının kolayca anlayıp uygulayabileceği biçimindedir. Bir kontaktlı kumanda devresinden PLC'ye geçmek oldukça kolaydır. Bununla beraber, kontaktlı kumanda devresi ile aynı amaçla gerçekleştirilen PLC programı farklı sonuçlar verebilir. Bunun nedeni kontaktlı kumanda devresi ile PLC arasındaki yapısal farklardır.

Kontaktlı kumanda devrelerinde her röle veya kontaktör eş zamanlı olarak çalışır. Yani aynı anda röle ve kontaktör bobinleri enerjilenir ve kontaklar konum değiştirir. PLC'lerde ise lojik işlemler sırasıyla yapılır.

### **2.4 PLC Kullanım Avantajları**

Geçmiş yıllarda elektronik olarak kontrol edilen her makinanın kendine özgü bir kontrolöre ihtiyacı vardı. Günümüzde bir PLC ile çok çeşitli makinelerin kontrolünü yapmak mümkündür. PLC'ye yazılan programla uygulamaya özgü çalışma şekillerini oluşturmak mümkündür. Bu açıdan PLC, genel amaçlı bir kontrolördür. Klasik kumanda sistemiyle oluşturulmuş kontrol sisteminin çalışma şeklinde değişiklik yapılması durumunda kumanda panosunun yeniden düzenlenmesi gerekir.[4] Oysaki, PLC ile kontrol edilen bir düzeneğin çalışma şeklini değiştirmek için, donanımı değiştirmeden sadece PLC programını değiştirmek yeterli olacaktır.

PLC programı, laboratuvar veya çalışma ortamında önceden çalıştırılabilir. Yazılan programın test edilmesi, değiştirilmesi, hataların bulunması için gerekli olan süre azalır. Klasik kumanda sistemleri ancak fabrika ortamında test edilebilir. Bu durumda sistemin işletmeye alınması için gerekli süre artar. PLC programının çalışması bilgisayar ortamında izlenebilir. Doğru ve hatalı çalışmalar anında tespit edilebilir. Görsel ortamda hataları bulmak kolaylaşır. Programın uygulanması esnasında oluşacak hatalar PLC tarafından algılanarak, operatör panelde hata mesajı üretmesi sağlanabilir. PLC'nin çalışma hızı çok yüksektir. PLC'nin çalışma hızı,

birkaç milisaniye seviyesindeki tarama süresiyle ölçülür. Bu hız, mekanik rölelerden çok yüksektir.

PLC programları, görsel programlama yöntemi olan ladder diyagram programlama şeklinde yazılabilir. Yarıiletken elemanlar, mekanik zamanlayıcı veya rölelerden daha güvenilirdir. PLC, güvenilirliği çok yüksek yarı iletken elemanlardan yapılmıştır. PLC ile yapılan kontrol panoları, klasik röle eşdeğerine göre daha az yer işgal eder. PLC'li kontrol sistemlerinin bakım maliyetleri düşük, arıza bulma süreleri minimumdur. Klasik kumanda sistemlerinde kayıt dışı değişikliklerin yapılması her zaman mümkün olsa da PLC programının yetkisiz kişiler tarafından değiştirilmesi mümkün değildir.

PLC'nin en büyük avantajı, düşük voltajlarda, bakım maliyetlerinin elektromekanik röle kontrol sistemlerine göre oldukça ucuz olması, buna ilave olarak birçok avantajlar sağlamaktadır. [5]

**BASİTLİK:** PLC'nin modüller yapısı her türlü özel uygulamalara ve sistemleri değiştirebilme, hataları düzeltme ve sistem değişikliklerin tamamına cevap vermektedir.

**ÖZELLİKLERİ:** PLC'nin modüler yapısı her türlü özel uygulamalara ve sistemlerin uzantılarına cevap verecek biçimde çalışmalıdır.

**UYGUNLUK:** Elektromekanik sistem kontrolleri ve bunların devre bağlantıları göz önüne alınırsa PLC'nin yaptığı işe göre kapladığı alan parçaları oldukça farklı ölçüde olduğundan yerden tasarruf edilir.

**DEĞİŞKENLİK:** PLC'nin mekanikli parçaları olmayıp genel amaçlı kontrol aygıtlarıdır. PLC'nin tekrar tekrar program yapacak biçimde hafızası vardır, birçok değişik bağlantıları yerine getirebilecek ilave devre tasarımları da yapılabilir.

**GERCEKÇİLİK:** PLC'lerin elektromekanik kısımları olmadığı için kırılacak

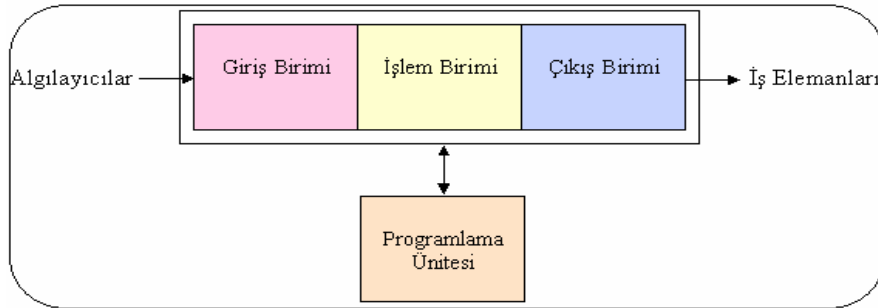
bozulacak parçaları yoktur. PLC'ler sonra kullanılmak üzere komple olarak depolanabilirler.

Endüstriyel otomasyon devrelerinde programlanabilir kontrolörlerin tercih edilmelerinin nedenleri şu şekilde sıralanabilir:

- Kumanda devresi, yazılımla sağlandığından kumanda devresini tasarlamak kontaklı (röleli) bir devrenin tasarımından daha kolaydır.
- Bütün kumanda fonksiyonları yazılımla gerçekleştirildiğinden, farklı bir uygulama için adaptasyon kolaydır.
- Kumanda devrelerine göre çok az yer kaplar.
- Güvenilirliği yüksek, bakımı kolaydır.
- Bilgisayarlarla ve diğer kontrolörlerle haberleşme olanağı vardır.
- Arıza ihtimali düşüktür.
- Kötü çevre koşullarında, özellikle tozlu ortamlarda röleli kumanda devrelerine göre daha güvenilirdir.

## 2.5 PLC Elemanları

Mikroişlemci tabanlı sistem olan CPU sistemin beynidir. PLC'nin girişine bağlı sensörlerden gelen bilgiyi okur, bünyesine yazılmış programı uygular ve program sonuçlarına göre çıkışına bağlanan elemanları kontrol eder. Giriş-çıkış birimi (I/O), giriş ve çıkışı terminallerinden oluşur. Bu birimler PLC'nin giriş ve çıkışına bağlanan elemanların CPU'ya bağlantısını sağlar. Giriş ünitesi, girişine uygulanan sinyalleri CPU'nun anlayacağı dile çevirir.



Şekil 2.2 : PLC'nin elemanları [6]

Benzer şekilde çıkış ünitesi, program sonuçlarını PLC'nin çıkışına bağlanan elemanların anlayacağı dile çevirir. Limit sviç, buton, fotosel birer giriş elemanları olup PLC'nin girişine bağlanır. Selenoid valf, sinyal lambası, kontaktör, motor sürücüsü birer çıkış elemanı olup PLC'nin çıkışına bağlanır.

Ladder diyagramda kullanılan format kumanda sistemine benzemektedir. Her bir sembol bir komutu temsil etmektedir. Sembollerin üzerinde bulunan harf ve rakamlar komut adreslerini ifade eder. Yazılan bir programı PLC'ye yüklenmeden önce mikroşlemcinin anlayabileceği makine diline dönüştürmek gerekir. Bu işleme *derleme* denir. Derleme işlemi yazılımla yapılır. Derlenen programda hatalar yok ise PLC'ye yüklenir. Programı çalıştırmak için CPU, RUN moduna getirilmelidir. PLC sürekli olarak girişine bağlanmış elemanların durumunu okur. Örneğin PLC'nin I0.0 girişine bağlı basınç svici açık ise bu giriş 0 kabul edilir. I0.0 girişine bağlı basınç svici kapandığında PLC giriş modülünden akım akacağından bu giriş 1 kabul edilir.

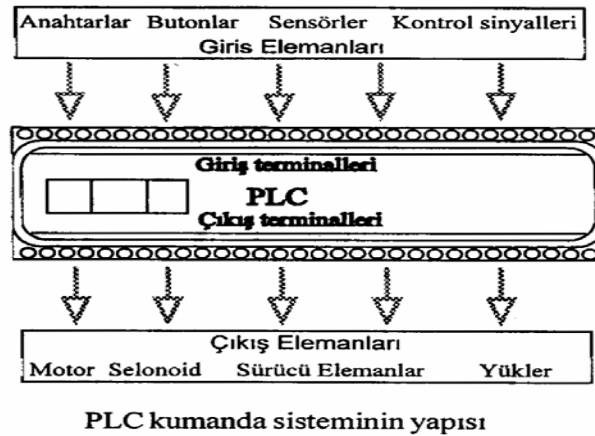
Öncelikle girişlerin durumu okunarak PLC belleğe yazılır. Girişe bağlı kontak açık ise belleğe 0, kapalı ise belleğe 1 yazılır. Daha sonra ladder diyagram program uygulanır. Belleğe yazılan giriş durumlarına göre (1 veya 0) programdaki açık kontaklar açılır veya kapanır. Örneğin I0.0 girişi 1 olarak okunmuş ise I0.0 kontağı kapalı olarak değerlendirilir. Benzer şekilde I0.1 girişi 0 olarak okunmuş ise I0.1 kontağı açık olarak değerlendirilir. Eğer bu kontaklar ladder diyagram programda soldan sağa sözde bir akım akışı sağlarsa, çıkış rölesi enerjilenir. Çıkış modülüne bağlı olan kontak kapanır ve Q0.0 çıkışına bağlı kontaktör enerjilenir. Benzer şekilde merdiven programda soldan sağa sözde enerji akışı yoksa Q0.0 çıkış rölesi enerjilenmez, 0 olur. Çıkış modülüne bağlı kontak açık olur ve Q0.0 çıkışına bağlı kontaktör enerjisiz kalır.

### **2.5.1 Donanım (Hardware):**

Donanım, elektronik modüller anlamında kullanılır. Bu modüller sistemin bütün fonksiyonlarını veya makinayı kontrol edebilir, adresleyebilir ve belirli bir iş

akışın sırasında harekete geçebilirler. PLC'nin donanım elemanlarını şu şekilde sınıflandırabiliriz:

- Merkezi işlem birimi (CPU)
- Giriş birimi (INPUTS)
- Çıkış birimi (OUTPUTS)
- Programlayıcı birimi (PROGRAMMABLE)



Şekil 2.3 : PLC kumanda sisteminin yapısı [7]

### 2.5.2 Yazılım (Software):

Yazılım, lojik işlemler, makine veya bir sistemdeki elemanların harekete geçirilmesini belirleyen programlardır. Yazılımlar, donanımda bulunan bellek birimi içerisinde saklanırlar ve istenildiğinde değiştirilebilirler. Kontrol akışı, donanımda herhangi bir değişikliğin gereksinim duyulmadan yazılan bir program ile değiştirilebilir.

### 2.5.3 Algılayıcılar ( Sensörler ):

Bu elemanlar kontrol edilecek bir makine veya bir sisteme direkt olarak bağlanırlar. Bilgiler, bu elemanların elektriksel akım değerlerine göre algılamalar PLC'ye iletilir. Algılayıcılara örnek olarak; Sınırlama anahtarları, İşaret üreticiler, Fotoseller, Sıcaklık algılayıcıları verilebilir.

#### **2.5.4 İş Elemanları:**

Bu elemanlar kontrol edilecek bir makine veya sisteme direkt olarak bağlanırlar. PLC'nin gönderdiği işaretlere göre durum değiştirirler. İş akışı bu durum değişikliğine göre belirlenir. İş elemanlarına örnek olarak; İkazlar (Lambalar, sesli ikazlar, ziller), Pnömatik silindirlere (Valf sistemler), Göstergeler, Kontaktörler, Motor yol vericilerini verebiliriz.

#### **2.5.5 Programlayıcı:**

Programlayıcı yazılımın oluşturulmasında ve PLC belleğine aktarılmasında kullanılır. Birçok uygulamada da yazılımın, algılayıcıların ve iş elemanların test edilmesinde kullanılır.

### **3. PLC'LERİN İŞLEVLERİ ve BELLEK YAPISI**

#### **3.1 PLC Çeşitleri**

Motor kontrol devrelerinde veya diğer yüklerin kumandasında değişik özelliklere sahip kumanda elemanları ve kumanda sistemleri kullanılmaktadır. Genel olarak kumanda sistemlerini iki grupta ele alınabilir.

Bunlar; klasik kumanda sistemleri ve elektronik kumanda sistemleridir. Klasik kumanda sistemleri çoğunlukla mekanik açılıp kapanan kontaklarla yani; buton, kontaktör, zaman rölesi vb. gibi elemanlarla gerçekleştirilir. Elektronik kumanda sistemleri ise iletim ve kesime geçen diyot, transistör, entegre ( IC ) gibi yarı iletken elemanlarla gerçekleştirilir.

#### **3.2 PLC'lerin İşlevleri**

##### **3.2.1 PLC'lerin Temel İşlevleri**

Kontrol sistemlerinde belli lojik bağlantıların yapılması gerekir. Belirli girişlerin sağlanması ya da sağlanmaması durumunda sisteme ait durumun değiştirilmesi için bazı çıkışların üretilmesi gerekir. Bu da PLC'ler tarafından sağlanır. PLC'lerin bu işlevini yerine getirebilmesi için aşağıdaki işlevleri yerine getirebilmesi gerekir.

- Temel ve kombinasyonel lojik ifadeler (AND, OR, AND-NOT(NAND), OR-NOT (NOR)
- Zamanlama işlevi
- Sayıcı işlevi

Bunlardan başka değişik matematiksel işlevleri; oransal (proportional), türev ve integral işlevlerini gerçekleştirme kapasitesine sahip programlanabilir kontrol cihazları da vardır.



### **3.2.2 Endüstriyel Otomasyon Devrelerinde PLC'nin Tercih Edilmesinin Nedenleri**

- Kontrol devresinin işlevi yazılımla sağlandığından, kontrol devresini tasarlamak, röleli bir devrenin tasarımından daha kolaydır.
- Bütün kontrol işlevleri yazılımla gerçekleştirildiğinden, farklı uygulama ve çalışma programlarını sağlamak çok kolaydır.
- Röleli kontrol devrelerine göre çok az yer kaplarlar.
- Güvenilirliği yüksek, bakımı kolaydır. Devrelerde arıza aramayı kolaylaştırır.
- Bilgisayarlar ve diğer kontrolörlerle haberleşme olanağı vardır. Bu özelliği bilgisayarlı otomasyon işlevine olanak sağlar. Arıza yapma ihtimali azdır.
- Kötü çevre koşullarında, özellikle tozlu ortamlarda, röleli kumanda devrelerine göre daha güvenilirdir.
- Kontrol algoritmasının tasarımı kolaydır. Bu konuda çalışan personele zaman ve emek tasarrufu sağlar.

### **3.2.3 PLC ile Bilgisayarlı Kontrol Sistemlerinin Karşılaştırılması**

PLC'lerin çalışma ilkeleri bilgisayarların çalışma ilkelerine çok benzer. Ancak bu sistemleri birbirinden ayıran birinci etken, PLC'ler endüstriyel üretim sistemlerinin çalışma bölgelerinde bulunan yüksek derecede elektriksel gürültü ve yüksek sıcaklıklara dayanıklı olarak imal edilmeleridir. Bilgisayarlar ve mikroişlemciler bu çevresel etkenler daha az dayanıklılık gösterir.

İkinci önemli ayrım noktası ise; PLC'lerin programlama, kullanım ve arıza arama yönlerinden daha uygun olanaklar sunmasıdır. PLC'nin merkezi işlem ünitesinde mikrokontrollü ünite bulunur. Bu yüzden her PLC bir bilgisayardır. Fakat her bilgisayar bir PLC değildir. PLC'ler üretimin yapıldığı tozlu, kirli ve elektriksel gürültü gibi ağır şartlarda çalışacak şekilde tasarlanmıştır. Bununla birlikte farklı bir programlama dili, arıza bulma ve bakım kolaylıklarının olması gibi özellikleri ile sanayi uygulamalarında bilgisayardan farklıdır. PLC programlama dili klasik kumanda devrelerine uygunluk sağlayacak şekildedir. Bütün PLC'lerde hemen hemen aynı olan AND, OR, NOT gibi Boolean ifadeleri kullanılır.

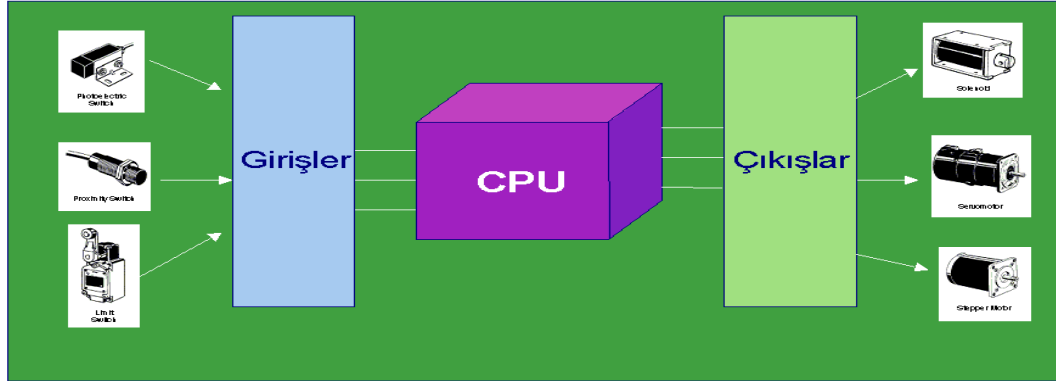
Sonuç olarak; küçük ve orta büyüklükteki her türlü kumanda sisteminde, küçük yapıya yüksek güvenilirlikli ve deęişebilir kontrolör olarak PLC'ler otomasyon üretiminin vazgeçilmez birer elemanı olmuştur.

### 3.3 Genel Olarak PLC Yapısı

PLC'ler; otomasyon devrelerinde yardımcı röleler, zaman röleleri, sayıcılar gibi kumanda elemanlarının yerine kullanılan mikroişlemci temelli cihazlardır. Bu cihazlarda zamanlama, sayma, sıralama ve ardışık lojik işlemler yazılımla gerçekleşir. Bu nedenle karmaşık otomasyon problemlerini hızlı ve güvenli bir şekilde çözmek olanaklıdır.

PLC'ler üç ana kısımdan meydana gelir.

- Merkezi işlem bölümü
- Giriş - Çıkış bölümü
- Program giriş elemanı



Şekil 3.1 : Tipik Bir PLC'nin Bölümleri [8]

### 3.3.1 Merkezi İşlem Birimi ve Bellek

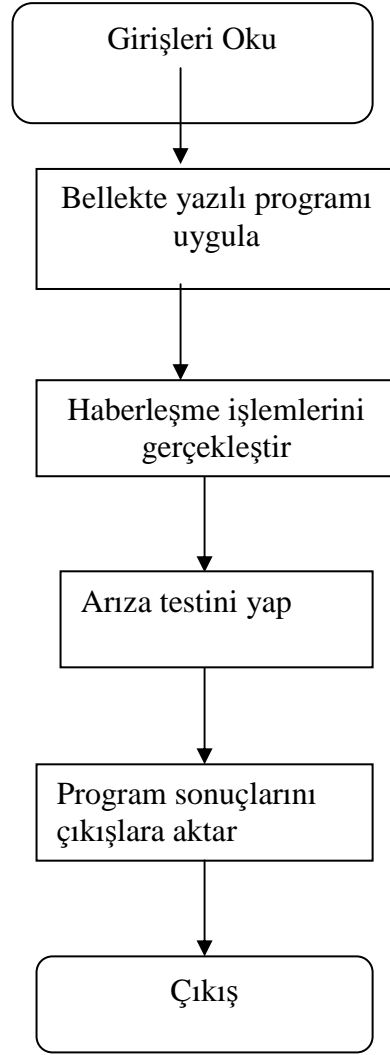
Giriş değerini işaret giriş belleğinden okur. Program belleğine yüklenmiş olan programı yürütür. Sonuç değerlerini çıkış sinyal belleğine yazar. Merkezi işlem biriminin çalışması ise şu şekilde olur: Merkezi işlem birimi ve bellek elemanı PLC'ye lojik bağlantıları yapma olanağı verir. Çalışmaları aşağıda verilen işlemlerin sürekli yinelenmesi şeklinde gerçekleşir.

1. Giriş tarafından gelen bilgileri değerlendirerek kontrol edilen sürecin durumunu sürekli gözetleme,
2. Giriş bilgilerini değerlendirerek bellekte yazılı programı çalıştırma,
3. Çıkış işaretinin gerekli olup olmadığına karar verilmesi,
4. Çıkış gerekiyorsa çıkış biriminin ilgili çıkışına işaret gönder,
5. Birinci adıma dönüş ve aynı işlemlerin yeniden gerçekleşmesi.

CPU, bellek elemanları ile birlikte PLC'ye mantık yürütme yetkisi veren bölümdür. Lojik bağlantıların yapıldığı bölüm CPU'dur. CPU'nun çalışacağı işletim sistemi ile belirlenir. Bu işletim sistemi üretici firmalara göre değişiklikler gösterir. Aynı mikroişlemciyi kullanan iki firmaya ait PLC'ler bile işletim sisteminin farklılığından ötürü değişik sonuçlar verebilir. Performansları farklı olabilir. İşletim sistemi kullanıcıya açık değildir.

#### 3.3.1.1 Tarama

CPU, giriş arayüzü aracılığı ile giriş verilerini okur, bellekte yazılı programı uygular, haberleşme işlemlerini gerçekleştirir, kendi içinde arıza testi yapar ve program sonuçlarını çıkışa aktarır. Bu işlemin bir kez yapılmasına tarama denir. Tarama zamanı PLC'nin çalışma hızının bir ölçüsüdür. Tarama döngüsü Şekil 3.2'de verilmiştir.



**Şekil 3.2 :** Tarama döngüsü

Tarama, girişleri okuma, denetim mantığını gerçekleştirme ve çıkışları yenilemeden oluşan sürekli bir sıralı süreçtir. Tarama hızı PLC seçiminde önemli bir parametredir. Örneğin kontrol edilen süreçte 6 milisaniye içinde iki defa durum değiştiren işaretler var ise programda bunların izlenmesi 12 milisaniyelik tarama hızı olan bir PLC ile yapılamaz. Çünkü tarama sırasında bu işaret durum değiştirip, yine eski durumuna örneğin “1” iken “0” veya “0” iken “1” durumuna geçebilir. Böyle bir durumda daha hızlı tarama yapan bir PLC veya özel I/O modülüne sahip devre kullanılır.

Tarama kavramı, Şekil 3.3’de verilen ladder diyagram program üzerinde ayrıntılı şekilde açıklanmıştır. CPU merdiven programı uygulamadan önce taramanın başlangıcında giriş terminallerine bağlı elemanların durumlarını okur. PLC’nin IO.0 giriş terminaline bağlı buton elemanı kapalı ise, giriş modülünden bir akım akar ve bu durum giriş imaj registerindeki IO.0 adresine 1 olarak yazılır. PLC’nin IO.1 girişine bağlı eleman açık olursa giriş devresinden bir akım akmaz. CPU bu durumu giriş imaj registerdeki IO.1 adresine 0 olarak yazar. Diğer girişler yukarıda ifade edildiği gibi değerlendirilip giriş imaj registre yazılır.

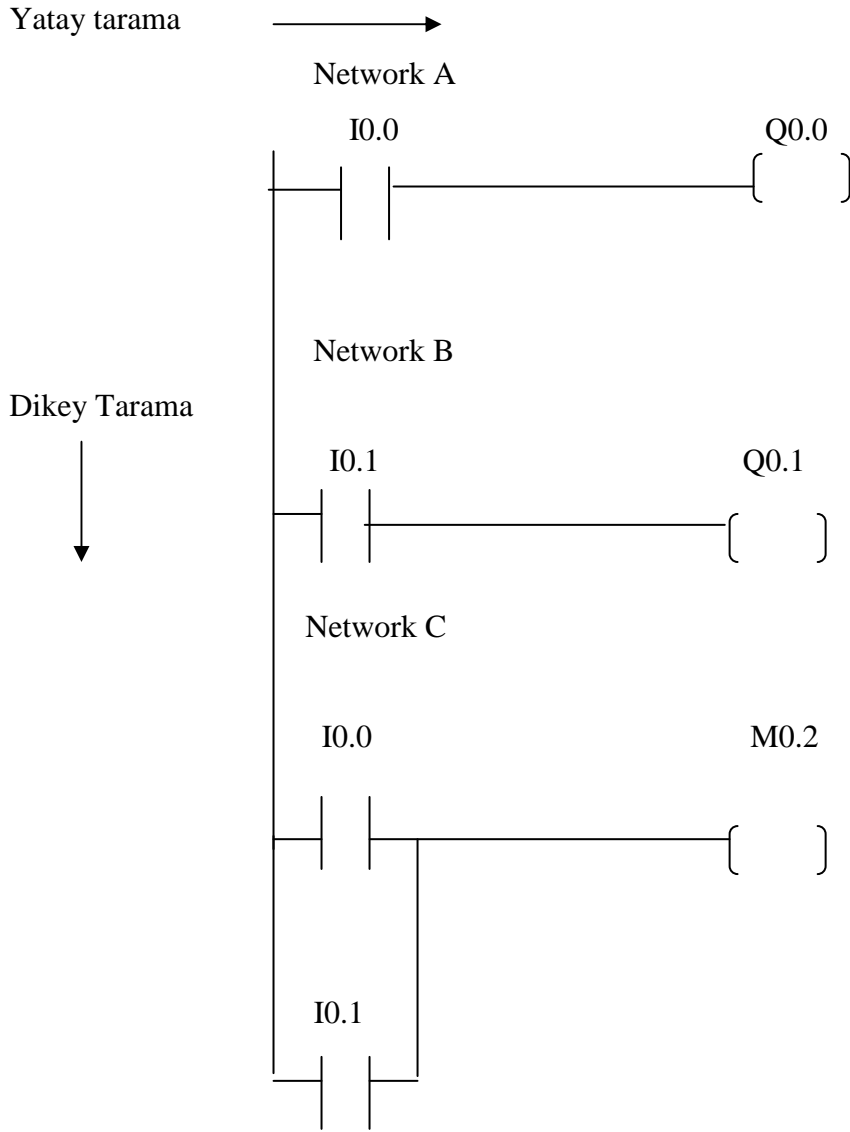
CPU, kontrol programını sırasıyla soldan sağa ve yukarıdan aşağıya doğru uygular. CPU öncelikle network A’yı uygular. IO.0’in durumu giriş imaj registerden okunur. Kontak sembolü okumaya karşılık gelmektedir. Giriş imaj registerde IO.0’in durumu 1 ise bu kontak kapalı olarak değerlendirilir. Q0.0’a lojik süreklilik sağlandığında Q0.0, 1 olur. Röle yazmaya karşılık geldiğinden Q0.0 çıkış imaj registerine 1 yazılır.

Daha sonra CPU network B’yi uygular. CPU, IO.1’in durumunu giriş imaj registerden okur. IO.1 sıfır olduğundan kontak kapanmaz ve Q0.1’e enerji akışı olmaz. Q0.1’in durumu çıkış imaj registerdeki yerine 0 olarak yazılır. Benzer şekilde IO.1, 1 olduğunda Q0.1 rölesine enerji akışı olur ve CPU Q0.1 çıkış imaj registerine 1 yazar.

Son olarak, CPU network C’yi uygulayacaktır. IO.0 ve IO.1’in kontakların durumu giriş imaj registerden okunur. Okunan değerlere göre söz konusu kontaklar açık veya kapalı olabilir. Bu kontaklardan biri kapalı konumunda olursa M0.2 rölesine lojik süreklilik (enerji akışı) sağlanır. Bu durumda CPU M0.2 rölesinin durumunu ilgili dahili bellek bitleri bölümüne yazar.

Haberleşme işlemleri ve arıza testinden sonra CPU, çıkış imaj registerdeki bilgileri çıkış terminaline aktarır. Çıkış imaj registerdeki Q0.0 bitinin durumu 1 ise Q0.0 terminaline bağlı eleman enerjilenir. Q0.0 biti 0 ise Q0.0 terminaline bağlı eleman enerjilenmez. M0.2 yardımcı rölesi direk bir çıkış olmadığından çıkış terminaline aktarılmaz.

Böylece CPU 1.taramayı tamamlamış olur. Bundan sonra CPU girişleri tekrar okuyacak, kontrol programını uygulayacak, haberleşme ve arıza testini yapacak ve program sonuçlarını çıkışlara aktaracaktır. Bu işlemler sürekli bir döngü şeklinde yapılır. Girişlerde herhangi bir değişiklik olduğunda, CPU bu durumu algılayacak ve program sonuçlarına göre çıkışların konumunu değiştirecektir. CPU'nun komut uygulama hızı çok hızlı olduğundan, girişlerdeki herhangi bir değişiklik çıkışlara hemen yansır.



**Şekil 3.3 :** Ladder diyagram programlamada tarama sırası

### 3.3.1.2 Bellekte Yer Alan Bilgiler

Belleklerde aşağıda belirtilen dört ayrı grup bilgi yer alır.

- Yönetim programı veya işletim sistemi :

İşletim sistemi herhangi bir mikroişlemciyi PLC haline getiren programdır. Çevre birimleri ile iletişim sağlar, kontrol programını yürütür.

- Kontrol uygulama programı :

Belirli bir kontrol görevi için yapılmış programdır. Ladder diyagram veya diğer programlama dillerinden birinde yazılmış olabilir.

- Veri tabloları :

Bu alanda kontrol işleminin gerçekleştirilmesinde kullanılacak denklem sabitleri, zaman ve sayma işlemleri için ön değerler ve uygulamalı programının kullanılacağı diğer sabitler gibi veriler saklanır. Burada ayrıca son olarak okunan sistem girişleri ve son olarak dışarı verilen sistem çıkışları da bulunur.

- Yazboz alanı :

Burada program tarafından kullanılacak geçici değerler bulunur. Ana bellek (veri tablolarının olduğu yer) yerine bu bellek alanının kullanımı, bu verilere çok kısa sürede ulaşılmasını sağlar.

Yukarıda açıklanan değişik bilgi türleri, özellikleri nedeni ile değişik bellek türlerini gerektirirler. Örneğin işletim sistemi devamlı olarak kalmalı, güç kesilmelerinden etkilenmemelidir. Fakat bu tür bir bellek uygulama programı için uygun olmayabilir, çünkü kullanıcı bu programı kolaylıkla değiştirebilme olanağına sahip olmak ister.

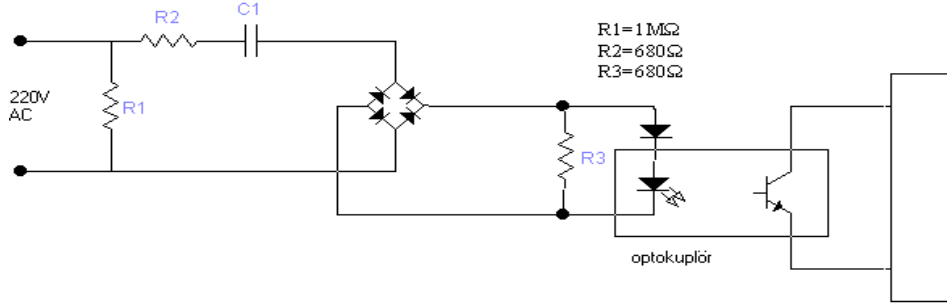
### 3.3.2 Giriş / Çıkış Birimi

PLC ile kontrol edilen sistem arasındaki iletişimi sağlar. İşaret belleği ve ara devreden meydana gelir. Giriş ve çıkışlar genellikle I/O (Input/Output) olarak isimlendirilir. I/O'un temel işlevleri şunlardır:

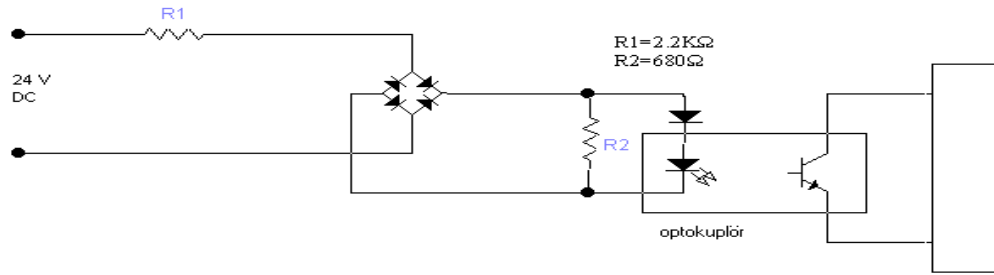
- Girişlerin herhangi birine işaret gelip gelmediğini sürekli kontrol etmek, gelmişse bunu değerlendirmek.
- Merkezi işlem biriminden gelen işaretleri çıkış lojik seviyesine geliştirmek.

### 3.3.2.1 Giriş Birimi

Kontrol edilen sisteme ilişkin basınç, seviye, sıcaklık algılayıcıları, butonlar ve sınır anahtarları gibi iki değerli işaretler (var – yok, 1 veya 0) giriş birimi üzerinden alınır. Genellikle giriş gerilim seviyesi biçimindedir. Gerilim seviyesi değerleri 24 V, 48 V, 100 – 120 V, 200 – 240 V doğru veya alternatif akım olabilir. [9]



Şekil 3.4 : 220 V AC gerilimle uyarılan bir giriş birimi



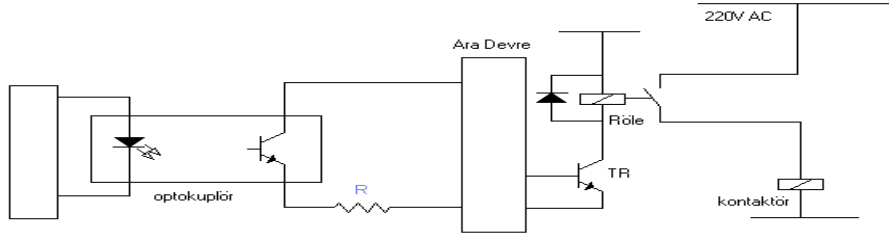
Şekil 3.5 : 24 V DC gerilimle uyarılın bir giriş birimi

PLC giriş birimine gelen bir işaretin lojik “1” seviyesinde olabilmesi için bir alt sınır lojik “0” kabul edilmesi için de bir üst sınır vardır. Giriş bilgisinin doğru olarak alınabilmesi için işaret gerilim seviyesinin bu değerleri sağlaması gerekir. Örneğin şekil 3.4’de verilen giriş devresinden verilen işaretin “1” kabul edilmesi için gerilim seviyesinin maksimum (OFF Voltage) 40 V olması gerekir. 40 – 120 V arasındaki değerlerin kullanılmaması gerekir.

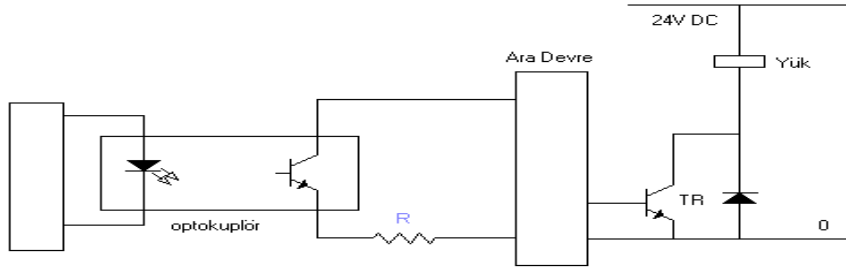


### 3.3.2.2 Çıkış Birimi

Kontrol edilen sistemdeki, kontaktör, röle, selenoid gibi kumanda elemanlarını harekete geçirmeye uygun donanımda olan birimdir. Bunlar; röle çıkışlı, triyaklı veya transistörlü olabilir. Özellikle çalışma sırasında çok sayıda, yüksek hızlı açma – kapama gerektiren durumlarda; doğru akımda transistörlü, alternatif akımda triyaklı olan çıkışlar tercih edilir. PLC üzerinde çıkışlardan büyük akımlar çekilmez. Çıkış birimi için kontak çıkışlı (röleli) bir devre şekil 3.6’de, transistör çıkışlı bir devre de şekil 3.7’de verilmiştir. [10]



Şekil 3.6 : Kontak çıkışlı devre



Şekil 3.7 : Transistör çıkışlı devre

Küçük boyutlu işler için kullanılan PLC’lerde giriş ve çıkış birimi, merkezi işlem birimi, yüksek hızlı sayıcı vs. tek modül içindedir. Daha geniş kapsamlı uygulamalarda kullanılan gelişmiş büyük boyutlu PLC’lerde bu birimler ayrı modüller halindedir.

### **3.4 PLC'lerin Programlanması**

PLC'ler ya bir programlama cihazı ile veya uygun bir yazılım yardımıyla bilgisayarla programlanabilir. Programlama cihazı ile programlamada her değişik marka PLC için özel programlayıcıya ihtiyaç duyulması açısından sorun olabilir. Bilgisayarda uygun yazılımla programlamada da belli bir yazılıma gereksinim duyulur. Bilgisayar programlama yönteminde ayrıca PLC ile bilgisayar arasındaki bağlantıyı yapmaya yarayan bir bağlantı kablosuna da gerek vardır.

Programlama birimi; PLC'leri programlamak amacı ile kullanılan ortamdır. Tasarlanan kontrol devresinin yazılmasını, PLC'ye aktarılmasını ve istenirse çalışma sırasında giriş-çıkış durumlarının gözlenmesini sağlar. Ayrıca giriş çıkışlara istenilen durumlar verilerek programın çalışması izlenebilir.

PLC'leri programlamak için kişisel bilgisayarlar da kullanılabilir. Bu amaçla bilgisayara yüklenen derleyici programdan yararlanır.

### **3.5 PLC Bellek Yapısı**

#### **3.5.1 Giriş**

PLC'nin CPU bölümü üç kısımdan oluşur: mikroişlemci, bellek ve güç kaynağı. Mikroişlemci tarafından uygulanan kontrol programı, PLC'nin girişine bağlanan elemanların durumları (açık, kapalı), program sonuçları gibi tüm veriler bellekte saklanır. Bellek sisteminin yakından incelenmesi, PLC'nin genel çalışma prensibi ve tarama kavramının anlaşılmasını kolaylaştırır. PLC'ler farklı türde kalıcı ve kalıcı olmayan bellek türlerini kullanır.

#### **3.5.2 Yariletken Bellekler**

##### **3.5.2.1 RAM Bellek (Rasgele Erişimli Bellek- Random Access Memory)**

Hem yazılabilir hem de okunabilir bellek türüdür. Enerji kesildiğinde bünyesindeki bilgiler kaybolduğundan kalıcı olmayan bellek sınıfına girerler. RAM'lar PLC

programların ve verilerin geçici olarak saklandıkları yerdir. PLC, uygulama programlarını RAM adresindeki yerden okur ve program sonuçlarını RAM bölgesinde belirlenmiş adreslere yazar. PLC'nin çalışma hızını düşürmemek için RAM okuma ve yazma sürelerinin kısa olması gereklidir. RAM'ların en büyük dezavantajı ise enerji kesildiğinde içindeki bilgilerin silinmesidir. Bunun yanında, CMOS RAM'lar stand-by modunda (okuma ve yazmanın olmadığı zaman) düşük miktarda enerji tüketirler. CMOS RAM'ların enerji kesildiğinde yedek batarya hemen devreye girer ve bünyesindeki bilgilerin kalıcı olması sağlanır.

### **3.5.2.2 ROM Bellek ( Salt Okunur Bellek- Read Only Memory)**

Kalıcı programların veya değişmeyen verilerin sürekli olarak saklanması için tasarlanmış yarıiletken bellek elemanlarıdır. Normal çalışma sırasında yeni veriler ROM'a yazılamaz fakat sadece okunabilir. Bazı ROM türlerine veriler ancak üretim aşamasında yazılabilir. Diğer türlerini ise (PROM, EPROM, EEPROM) kullanıcılar programlayabilir. Bazı ROM'lara (PROM) veri yazıldıktan sonra yeniden değiştirilemez. EPROM, EEPROM gibi ROM türleri istendiği kadar yeniden programlanabilir.

Enerji kesildikten sonra ROM içeriği kaybolmaz. PLC'deki işletim sistemi gibi kalıcı programların saklandığı ortam olarak ROM'lar kullanılır. ROM'lar sadece okunabilir olduğundan PLC'lerde uygulama belleği alanı olarak kullanılmaz.

### **3.5.2.3 PROM Bellek (Programlanabilir Salt Okunur Bellek)**

Programlanabilir ROM'lar(PROM), ROM'ın özel bir türü olup bir kez programlandıktan sonra yeniden programlanamaz. PROM'un programlanmasında bir hata yapıldığında yeniden silinip programlanması mümkün değildir. Bu yüzden bu tür bellek elemanlarına bir kez programlanabilir (one time programable, OTP) bellek denir. PROM'ların bünyesine bilgi yazıldıktan sonra kalıcı olur.

PROM hafıza hücreleri 0 veya 1 olarak programlanabilen bağlantılardan meydana gelmiştir. Kullanıcı istediği hafıza verisini oluşturmak için bu bağlantılardan kimilerini eritir, kimilerine hiç dokunulmaz. Bu eritme işlemi gerilimi hassas olarak kontrol edilebilen PROM programlayıcılar ile yapılır. PROM bellek, doğruluğu tamamen tespit edilmiş programların saklanması için kullanılır.

#### **3.5.2.4 EPROM Bellek (Silinebilen Programlanabilir Salt Oku Bellek)**

Silinebilir, programlanabilir bellek türü olan EPROM'lar kullanıcı tarafından istenildiği kadar programlanabilir. EPROM'lar bir kez programlandıktan sonra kalıcı olurlar. Programlama özel EPROM programlayıcı cihazı ile yapılır.

EPROM içindeki bilgi depolama hücresi MOSFET transistörlerdir. Programlanmamış durumdaki her bir EPROM hücresi 1 konumdadır. Yüksek gerilimli (10-25V) programlama palsı ile yüksek enerjili elektronların enjekte edilmesiyle transistör iletim konumuna getirilir ve bu elektronlar deşarj olamayacağı bir bölgede sıkıştırılır. Bu yükler, EPROM'un enerjisi kesildikten sonra bile o transistör ü iletim konumunda tutar. Bir kez programlanan EPROM'u silmek için entegre devrenin üzerinde bulunan pencereden ultraviolet ışınlar uygulanır. UV ışınlar, sıkıştırılmış olan elektronların silikon tabakaya akması için fotoakım yolu oluşturarak o hücreyi tekrar 1 konumuna getirir.

#### **3.5.2.5 EEPROM Bellek**

Silinebilir, programlanabilir EPROM'u yeniden programlamak ve silmek için buldukları devreden çıkarmak gerekir. EPROM'un silinmesi işlemi tüm bilgiler kaybolur. Bu sakıncaları ortadan kaldırmak için 1980 yıllarında elektriksel olarak programlanabilir ROM'lar geliştirildi. EEPROM'un silinmesi ve programlanması devre üzerinde yapılır. Silme işlemi için UV ışığına ihtiyaç yoktur. EEPROM'un programlanması için Vpp programlama girişine (12 V gibi) bir gerilimin uygulanması gerekir.

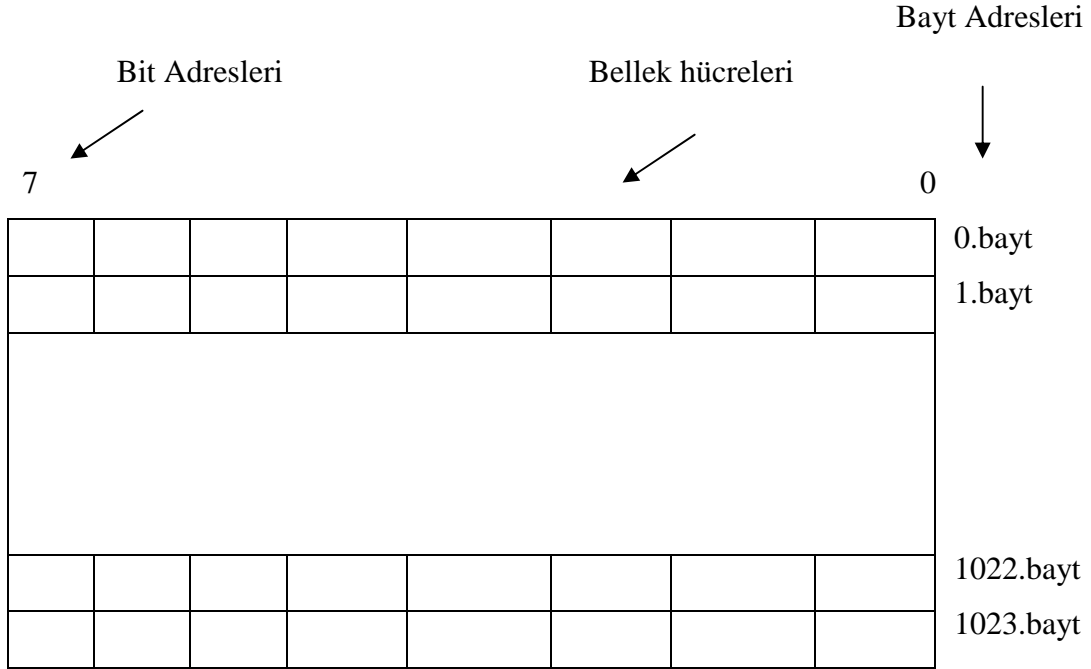
EEPROM kolay şekilde programlanabildiğinden ve silinebildiğinden dolayı PLC'deki kontrol programları ve kalıcı verilerin sürekli olarak saklanabileceği ortamlar olarak kullanılır.

### **3.5.2.6 NOVRAM Bellek**

RAM ve EEPROM belleklerin tek bir çipin içine yerleştirilmesiyle oluşmuştur. RAM bölümündeki her bir bitin karşılığı EEPROM bölümünde de vardır. Kalıcı datalar EEPROM'a yazılırken aynı zamanda RAM'a da yazılır. Kullanıcı kontrol programı direkt RAM'a yazıldığından kontrol programının uygulanması RAM'dan yapılır. Aynı zamanda kullanıcının herhangi bir müdahalesi olmaksızın aynı program EEPROM'a da yazılarak kalıcı yapılır. PLC'nin enerjisi kesildiğinde RAM'daki bilgiler silinir fakat EEPROM'daki bilgiler kalıcı olur. PLC enerjisi geldiğinde otomatik olarak EEPROM'daki bilgiler RAM'a yüklenir. Böylece programın bir kopyası sürekli olarak EEPROM'da muhafaza edilir.

## **3.6 Bellek Kapasitesi**

PLC belleği, verileri 0 veya 1 formunda saklayabilen iki boyutlu saklama hücreleri olarak düşünülebilir. Belleğin en küçük saklama ünitesi olan bit'te veriler 1'ler ve 0'lar şeklide saklanır. İşlemci kimi zaman birden fazla biti bir arada işlemek zorunda kalabilir. Verilerin bir yerden bir yere aktarılmasında bit grupları olarak değerlendirilmesi daha etkin olur. Sayıları ve kodları bellekte saklayabilmek için bit gruplarına ihtiyaç vardır. Sekiz bitlik verinin oluşturduğu gruba *bayt* denir. Daha büyük sayıları ifade etmek için bayt'lar yeterli olmayabilir. İki baytın yan yana gelmesiyle *word*, 32 bitin yan yana gelmesiyle *double word* oluşur.[11]



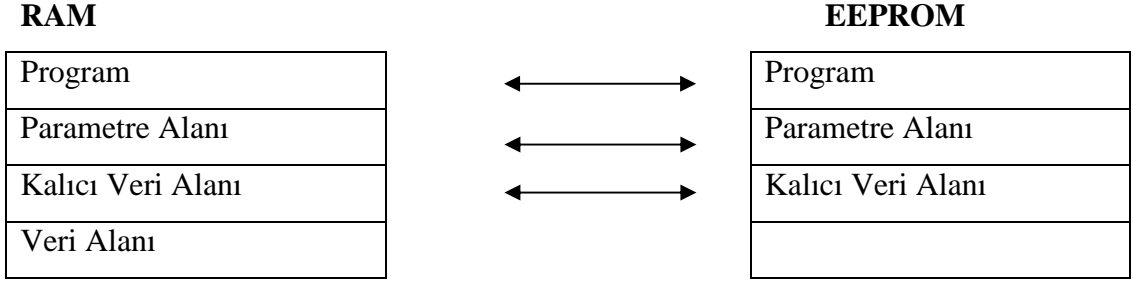
**Şekil 3.8 :** 1Kbayt'lık bellek blok diyagramı

Kapasite, bir bellekte saklanabilecek bit sayısını ifade etmenin bir yoludur. 1024 tane 8 bitlik bilgiyi saklayabilen bir belleğin olduğunu düşünelim. Bellek kapasitesi 1024x8 şeklinde ifade edilir. Bellekteki 1024 farklı bayt alanına 1K (kilo) denir. Buna göre Şekil 3.8'deki bellek kapasitesi 1 Kbayt şeklinde ifade edilir.

### 3.7 PLC Bellek Haritası

Bir PLC'de belleğin nasıl kullanıldığını gösteren şemalara bellek haritası denir. PLC belleği, program alanı, parametre alanı ve veri alanı olmak üzere üç kısımdan oluşur. [12] Şekil 3.9'dan görüleceği üzere program alanı, parametre alanı ve veri alanının bir kısmı EEPROM'da kalıcı hale getirilmiştir. Batarya kartuşu ile RAM'daki bilgiler kalıcı yapılabilir.

**Program alanı** : Kontrolörün uyguladığı kullanıcı programının bulunduğu alandır. Program ayrıca kalıcı bellek (EEPROM) alanında saklandığından enerji kesintisinde silinmez.



**Şekil 3.9** : PLC bellek haritası

**Parametre alanı** : Şifre, istasyon adresi, kalıcı bellek alanları gibi konfigürasyon bilgilerin saklandığı bölgedir. Aynı zamanda kalıcı bellek alanında saklanır.

**Veri alanı** : Kontrol programı tarafından erişilir bir çalışma alanıdır. Veri alanında sabit kontrol parametreleri, hesaplamalar ve ara sonuçlar için registerler vardır. Ayrıca zamanlayıcılar, sayıcılar, yüksek hızlı sayıcılar, analog giriş ve çıkışlar için registerler, değişken bellek alanı, giriş ve çıkış imaj registerleri, dahili ve özel bellek bitleri vardır.[13] Veri alanının bir kısmı kalıcı bellekte saklanır. RAM'daki bilgiler süper kondansatör sayesinde enerji kesildiğinde bir süre kalıcı olarak saklanır.

### 3.8 Veri Alanı

Veri alan, veri belleği ve veri nesnelere olmak üzere iki kısımdan oluşur. Veri belleği, değişken bellek, giriş imaj register, çıkış imaj register, dahili bellek bitleri, özel bellek bitleri, lokal bellek alanı ve ardışıl kontrol röle bellek alanından oluşur. Özel bellek bitlerinin bir kısmı hariç tüm veri belleğine okuma ve yazma işlemi yapılabilir. Özel bellek bitlerinin bir kısmına sadece okuma işlemi yapılabilir. Tüm veri alanına bit, byte, word, double word olarak erişmek mümkündür.

Veri nesneleri bölümünde zamanlayıcılar, sayıcılar, analog giriş ve çıkışlar, akümülatör registerler ve yüksek hızlı sayıcılar vardır. Bu bölgeye erişim sınırlı olup nesnelere kullanımına bağlıdır.

### **3.8.1 Değişken Bellek**

Sabit veya değişken verilerin, hesaplama sonuçlarının saklanması için ayrılan bölgedir. Değişken bellek V harfiyle gösterilir. Bir kısmı EEPROM'da saklandığından kalıcıdır. Bu bölgeye hem okuma (R) hem de yazma (W) yapılabilir.

### **3.8.2 Giriş İmaj Register**

PLC'nin dijital giriş terminaline bağlanmış girişlerin durumlarının kaydedildiği alandır. Giriş imaj registerdaki bit sayısı en az PLC girişine bağlanabilecek giriş sayısı kadar olmalıdır. Eğer bir PLC'nin 64 girişi varsa, giriş imaj registerde 64 bite ihtiyaç vardır. Giriş imaj register bölgesi I harfiyle gösterilir. Bu bölgeye hem okuma hem de yazma yapılabilir.[14] PLC'nin çalışması sırasında işlemci, giriş terminaline bağlı her girişin durumunu okur ve buna karşılık gelen (0 veya 1) ikili sayı değerine giriş imaj registre yükler. Giriş imaj register, girişe bağlı elemanlarının durumunu yansıtır.

### **3.8.3 Çıkış İmaj Register**

Dijital çıkış terminaline bağlı elemanların durumunu kontrol eden bit dizileridir. Kullanıcı programı sonuçlarına göre bu registerdaki bitlerin içerikleri değişir. Örneğin, Q0.0 biti 1 olduğu zaman Q0.0 çıkışına bağlı eleman çalışır. Q0.0 biti 0 olduğu zaman bu çıkışa bağlı eleman çalışmaz. Çıkış imaj registerdaki bit sayısı en az PLC dijital çıkış terminaline bağlanabilecek eleman sayısı kadardır. Çıkış imaj register Q harfi ile gösterilir. Bu bölgeye hem okuma hem de yazma yapılabilir.



### 3.8.4 Dahili Bellek Bitleri

Programdaki ara sonuçları ve kontrol bilgilerini saklamak için kullanılan bölgedir. Dahili bellek bitleri M harfi ile gösterilir. Dahili bellek bitlerinin bulunduğu bölgeye hem okuma hem de yazma yapılabilir.

### 3.8.5 Özel Bellek Bitleri

Özel bellek bitleri sayesinde PLC'nin gücü ve yetenekleri artırılmış olur. Özel bellek bitleri durum ve kontrol fonksiyonları gibi PLC sistemi ile kullanıcı programı arasında bir anlamda bilgi iletişimi sağlar. Özel bellek bitlerinin bulunduğu alan sadece okunabilir ve hem okunur hem de yazılabilir olmak üzere ikiye ayrılır. Özel bellek bitleri SM harfiyle gösterilir.

Özel bellek bitleri 1, 8, 16 ve 32 bit olarak kullanılabilir. Sadece okunabilir alanlar SM0'dan başlar SM29'a kadar devam eder.[15] Aşağıda bazı özel bellek bitleri verilmiştir :

**SM0.0** : Bu bit her zaman 1'dir.

**SM0.1** : Bu bit ilk taramada 1, diğer taramalarda 0'dır.

**SM0.4** : Bir dakikalık saat palsları üretir. 30sn bir, 30sn sıfır.

**SM0.5** : Bir saniyelik saat palsları üretir. 0.5sn bir, 0.5sn sıfır.

**SM1.0** : Belirli komutların uygulanması sonucunda işlem sonucu 0 olursa bu bit 1'e kurulur.

**SM1.1** : Belirli komutların uygulanması sonucunda taşma (overflow) olursa bu bit 1'e kurulur.

**SM1.2** : Belirli komutların uygulanması sonucunda işlem sonucu negatif olursa bu bit 1'e kurulur.

**SM1.3** : Sıfır ile bölme yapılmak istendiğinde 1'e kurulur.

### **3.8.6 Ardışıl Kontrol Röle Bellek Alanı**

Ardışıl kontrol röle bitleri (S), sıralı kontrol programını segmentler şeklinde düzenlemek için kullanılır. Bu komutların kullanılmasıyla kontrol programı segmentlere bölünebilir. S bitlere bit, bayt, word, double word olarak erişilebilir. Ardışıl kontrol röle bellek alanı S ile gösterilir. Ardışık kontrol röle bellek alanına hem okuma hem de yazma yapılabilir.

### **3.8.7 Lokal Bellek Alanı**

Lokal belleğin V değişken bellekten farklı bir özelliği vardır. Değişken belleğe erişim kontrol programının herhangi bir parçası (ana program, altıyordam, kesme ) tarafından olabilir. L lokal bellek alanına erişim kontrol programının belirli parçası tarafından olabilir. Ana program için ayrılan L bellek alanına altıyordamdan veya kesme programları tarafından erişilemez. Bir altıyordam veya kesme programından ana program için ayrılan L bellek alanına erişilemez. Benzer şekilde, bir kesme programı, ana program veya altıyordam için ayrılan L bellek alanına erişemez.

L bellek bölgeleri ihtiyaca göre oluşur. Ana program uygulanırken altıyordam ve kesme programı için bellek alanı ayrılmaz. Kesme uygulanırken sadece kesme programı için L bellek alanı oluşur. L bellek bitlerine bit, bayt, word, double word olarak erişilir. L belleği indirekt adresleme için işaretçi (pointer) olarak kullanılabilir, fakat L alanı indirekt olarak adreslenemez.

## **3.9 Bellek ve Programlama Koşulları**

PLC tipine bağlı olarak, sistem belleği CPU ile aynı kart üzerinde bulunabilir veya ayrı bir kart tahsil edilebilir. Bu metot, CPU kartında karşıt bir değiştirme gerekmesizin sistemde gereken maksimum değişikliği mümkün kılar.

Bellek genişliği normalde sistemde gereken Giriş / Çıkış uçlarının miktarına bağlıdır. Diğer bir etken ise kurulan kontrol programıdır. Herhangi bir programın tam kapasitesi tüm tasarlama, kod çözme, kurma ve test etmeyi içeren yazılıma kadar

belirlenemez. Program büyüklüğü, Giriş / Çıkış uç sayılarına bağlıdır. Çünkü her uç yazma ve okuma komutları içermelidir. Kartlar arasında geçişe müsaade etmek için özel fonksiyonlar, ana PLC belleği içinde bellek boşlukları gerektirebilir. Sonuçta bu boşluklar sayesinde kontrolde değişiklikler yapılabilir. Bellek RAM veya EPROM seçilebilir. RAM bellek sistemin kurulmasından önce ve sonra programın süratli ve kolayca çalıştırılmasını sağlar. Giriş / Çıkış ve veri fonksiyonları dinamik olduklarından bunlar için daima RAM kullanılmalıdır. EPROM bellek ise program saklama ve saklanan programların gerektiğinde özel bir silici ile temizlenmesinde kullanılabilir. EPROM'lar özellikle PLC üzerinde yüklü bir programın birkaç makinayı birden kontrolünde tercih edilebilir. Yine de RAM, program yazılımı sırasında tamamlanıp test edilinceye kadar saklama için kullanılabilir.

### **3.10 Analog Giriş-Çıkışlar**

#### **3.10.1 Analog Girişler**

Analog girişler, sıcaklık basınç vb. transmitterlerinden gelen analog akım, gerilim bilgilerini dijital büyüklüklere dönüştüren elemandır. Dijital değerlere dönüştürülmüş analog büyüklükler sadece okunabilir. 16 bit uzunluğundaki registerlerde saklanır. Analog girişler AI harfleriyle gösterilir.

#### **3.10.2 Analog Çıkışlar**

Program sonunda hesaplanan büyüklükleri analog sinyale(gerilim, akım) dönüştüren modüllerdir. 16 bitlik bu registerler sadece yazılabilir bölgelerdir. Bu registerlere yazılan rakama karşılık gelen akım veya gerilim değerine analog çıkış modülü üretir. Analog çıkışlar AQ harfleriyle gösterilir.

### **3.11 Digital Giriş-Çıkışlar**

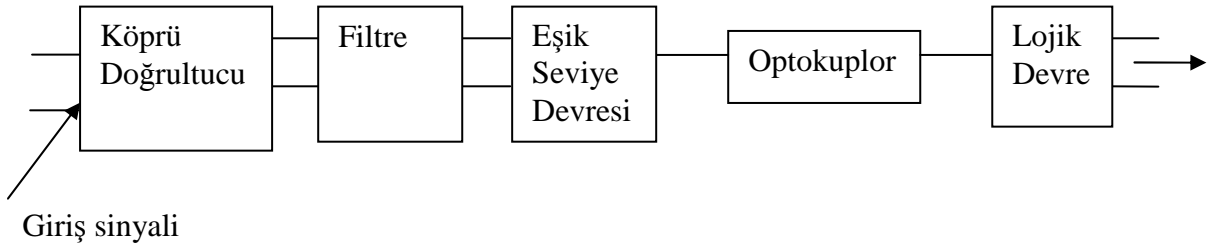
#### **3.11.1 Giriş**

Dijital giriş modülü, PLC girişine bağlı elemanların hangisinin açık veya kapalı olduğunu sezer, girişin durumunu CPU'nun anlayacağı dile çevirir. Dijital çıkış

modülü, CPU'nun ürettiği sinyalleri çıkışa bağlı elemanların anlayacağı dile dönüştürür.

### 3.11.2 Dijital Girişler

Dijital giriş modülü elektronik olarak giriş terminaline bağlı elemanların hangisinin açık veya kapalı olduğunu sezer. Dijital giriş devrelerinin çalışma prensipleri aynıdır. Alternatif gerilim uyarlamalı dijital giriş blok diyagramı aşağıdaki Şekil 3.11'de verilmiştir.



**Şekil 3.10** : Alternatif gerilim uyarlamalı dijital giriş blok diyagramı

Alternatif gerilim uyarlamalı dijital giriş modülün girişine uygulanan sinyal köprü doğrultucuda doğru gerilim seviyesine dönüştürülür.[16] Giriş hattındaki sinyaller filtre edilir. Eşik seviye devresi, gelen gerilimin belirli eşik seviyesine gelip gelmediğini sezer. Dijital girişe uygulanan sinyal eşik seviyesini aşarsa geçerli bir giriş (lojik 1) olarak değerlendirilir. Giriş devresinde oluşabilecek aşırı gerilimler, ani gerilim artışları optokuplor izolasyonu ile CPU'ya iletilmez. Çoğu PLC giriş devrelerinde uygun giriş gerilimi olduğunu göstermek için led göstergeler vardır. Led yanıyor ise söz konusu girişin 1 olduğu anlaşılır.

Doğru gerilimle çalışan dijital giriş modüllerin çalışma prensibi alternatif uyarlamalı dijital giriş modüllerine benzer fakat bu modüllerde köprü doğrultucu yoktur. Dijital giriş modülün çalışma gerilimi 24V'tur. 15V'un altındaki gerilimler lojik 0, 15V'un üstündeki gerilimler lojik 1 olarak kabul edilir. Dijital girişlerin iki konumu vardır. Bu girişlere buton, kontak, limit sviç gibi sadece açık veya kapalı olmak üzere iki

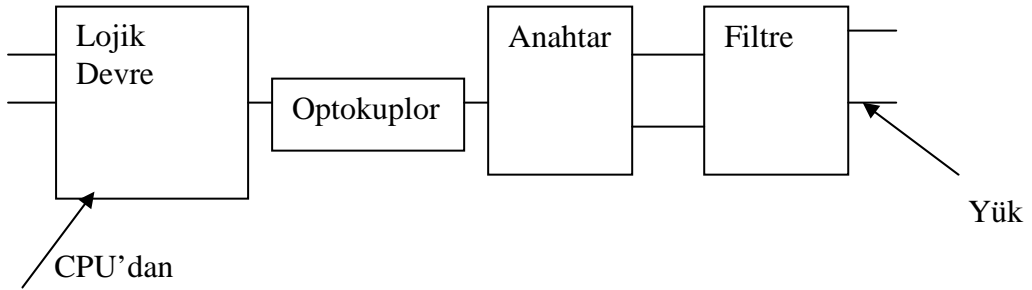
konumlu elemanlar bağlanabilir. PLC'nin dijital girişlerine bağlanabilecek bazı elemanlar şöyle sıralanabilir:

- Kontaktör kontağı
- Buton
- Limit sviç
- Fotosel
- Yaklaşım sviç'i (proximity sviç)
- Seviye sensör kontakları
- Röle kontakları

### 3.11.3 Dijital Çıkışlar

Dijital çıkışların çalışma prensibi dijital girişlerin tam tersidir. Dijital çıkış modülü, CPU tarafından üretilen sinyalleri kontrol edilen büyüklüklerin anlayacağı dile dönüştürür.

Dijital çıkış devresinin blok diyagramı aşağıdaki Şekil 3.11'de verilmiştir.



Şekil 3.11 : Tipik dijital çıkış blok diyagramı

Her tarama sonunda CPU'dan lojik devreye bir sinyal gönderir. Lojik devrenin aldığı sinyal optokuplor vasıtasıyla anahtar elemanı bloğuna iletilir. CPU'dan gelen 1 olduğunda anahtar elemanı iletme geçer ve yüke enerji sağlanır. CPU'dan gelen sinyal 0 ise anahtar elemanı iletme geçmez ve yük enerjisiz kalır. Filtre elemanı açma-kapama esnasında oluşan aşırı gerilimlerden anahtar elemanını korur.

Dijital çıkışlara bağlanabilecek bazı elemanlar şöyle sıralanabilir: Küçük güçlü motor, kontaktör, alarm elemanları, sinyal lambaları, motor starterleri.

## 4. PLC'LERİN PROGRAMLAMASI

### 4.1 PLC'lerin Programlama Özellikleri Ve Yöntemleri

Ladder (merdiven şeması) , STL (komut listesi) program yazımını mümkün kılar.

Uygulama programları bloklar halinde gerçekleştirilir. Bu olanak ;

- Programların basit ve anlaşılır halde yazılablmesini
- Program bölümlerinin standardizasyonunu
- Programın kolayca değiştirilebilir olmasını
- Programın testinin daha kolay yapılmasını sağlar.

#### ➤ Programlama Yöntemleri:

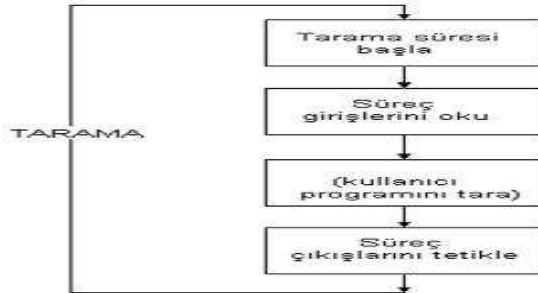
Programlar üç ayrı gösterim şeklinde yazılabilir. Bunlar:

- STL : Komut Listesi (Statement List)
- LAD: Merdiven Şeması (Ladder Diagram)
- CSF : Lojik Kapı Gösterimi (Control System Flow Chart)

### 4.2 Programlama Dili

Üç dilden biri ile yazılmış olan program PLC'nin anlayabileceği şekle dönüştürülür.

### 4.3 PLC Programları Çalışma İlkesi :



Şekil 4.1 : PLC programlarının çalışma ilkesi

### 4.4 Programlayıcı Birimi

Kumanda ve kontrol amacıyla yazılan bir programın PLC program belleğine yüklenmesi bir programlayıcı birimi ile sağlanır. Bu birim; programın yazılması,

PLC'ye aktarılması ve çalışma anında giriş/çıkış veya saklayıcı durumlarının gözlenmesi ya da değiştirilmesi gibi olanakları da sağlar.[17]

## **4.5 Programlama**

Programlama; belirli bir görevi yapmak üzere tasarlanmış, bir kontrol ya da kumanda sisteminin PLC'de gerçekleşmesi için, yürütülmesi gereken işlemleri sağlayan, komut dizisinin yazılmasıdır.

### **4.5.1 Programlama Yapıları**

Tüm PLC sistemlerinde, genel olarak, doğrusal ve yapısal programlama olarak tanımlanan iki tür yapı kullanılır.

**1-Doğrusal programlama:** Tüm komutların aynı program içinde yer aldığı ve komutların arka arkaya yazıldığı programlama biçimidir. Tüm komutlar, bir çevrim boyunca, işleme girer ve yazılış sırasına göre yürütülürler. Doğrusal programlamada tüm komutlar ana programda bulunur ve kesme altprogramları ana programın program sonu komutundan sonra sırayla yazılır.

**2-Yapısal programlama:** Program ve fonksiyon blokları ve de organizasyon bloğu biçiminde düzenlenmiş programlama biçimidir. Sistem programı organizasyon bloklarını, organizasyon blokları ise program ve fonksiyon bloklarını işletir.

Programlama yapılırken komut, ladder diyagram ya da lojik kapı gösterimi ile programlama biçimleri kullanılır. Bu programlama biçimlerinden lojik kapı gösterimi (LKG) ve ladder diyagramı gösterimi (LDG) ile programlama biçimleri tüm PLC'ler için benzerdir. Komut ile programlama biçimi ise, bazı PLC modelleri için farklılık gösterir.

- **LDG ile programlama:** Kumanda devresinde, kontakların seri ve paralel bağlandığı ve sonuçta bir rölenin veya bir özel fonksiyonun çalıştırıldığı programlama biçimidir.

- **LKG ile programlama:** Kumanda devresinin lojik işlemleri simgeleyen lojik kapı şekilleri ile programlanmasıdır. Bu programlama biçiminde lojik kapı şekilleri ,art arda seri ya da paralel bağlanarak program yazılır.
- **Komut ile programlama:** Tüm anahtarlı (kontaklı) devreler lojik fonksiyonlar ile ifade edilebilir ve her lojik fonksiyon “VE, VEYA, DEĞİL” işlemlerini sağlayan komutlar kullanılarak programlanabilir.

#### 4.6 PLC'lerin Programlama Modları

Kullanıcı tarafından yazılan ve bellekte saklanan program, verilen bir süreç durumu için doğru denetim işaretlerinin üretilmesini sağlar. Bu programın yazılış biçimi (programlama dili) çeşitli türlerden biri olabilir. Programlama aracı, programlayıcı ile PLC arasındaki bağlantıyı sağlar. En basit şekil, PLC'lerin yanındaki bir tuş takımı veya PLC' ye bağlı bir kablo ile bağlanan, elde tutulan, bir dizi işlev tuşundan meydana gelmiş bir ayardır. Daha gelişmiş şekillerinde ekran ve klavye bulunur. Bu olanaklarla programın daha iyi bir şekilde görülmesi ve otomatik program dokümantasyonu sağlanır.

PLC'ler için geliştirilmiş olan programlama dilleri, kontaktörlü ve röleli kontrol devreleriyle uğraşanların kolayca anlayıp uygulayabileceği şekilde tasarlanmıştır. İşlemlerde BOOLE cebri ilişkileri geçerli olup aşağıdaki programlama yöntemlerinden söz edilebilir :

- Komut listesi (Statement List)
- Kontak planı (Ladder Diagram)
- Delphi programlama dili
- Lojik şema (Flow Chart)
- Matris programlama

Bu programlama yöntemlerinden komut listesi ile program, genellikle el ile programlama elemanlarında kullanılır.

Ladder diyagram ile programlama tekniği ile yazılan programlar biçim olarak birbirine benzemekte, sadece giriş – çıkış sembollerinin (kodlarının) ve PLC için



adreslerin (sayıcı, zamanlayıcı gibi) kodları değişmektedir. Lojik diyagram programlama tekniği ile programlamada biçim olarak benzerlik vardır. Programlamada genellikle aynı tip semboller kullanılmaktadır.

Programlama teknikleri yazılış biçimine göre iki gruba ayrılabilir. Bunlar ;

- Genellikle küçük boyutlu PLC'lerde kullanılan adım-adım programlama (lineer programlama)
- Büyük boyutlu PLC'lerde kullanılabilen yapısal programlama biçimindedir.

Lineer programlamada; deyimler sırasıyla yürütülür ve bir çevrim boyunca bütün deyimler işleme girer. Yapısal programlamada ise programlar bloklar halinde oluşturulur ve bir organizasyon bloğu yazılarak bir çevrimde hangi blokların yürütüleceği belirlenir. Yapısal programlamada bir çevrim boyunca program belliğindeki programın bütün deyimlerinin işlenmesi zorunluluğu yoktur. Organizasyon bloğunda belirtilen prosedüre göre bazı bloklar işleme girmeyebilir. İşleme girmeyen bloklara ilişkin sonuçlar işaret belleğinde tutulur.

Hangi programlama tekniği kullanılırsa kullanılsın PLC'ler için yazılan bir programın yürütülme süreci aynıdır. Bir PLC'de yazılan programın PLC program belleğine yüklenip çalıştırılması ile gerçekleşen işlemler şu şekildedir.

### **1. Adım:**

Giriş ve çıkış durumları okunur.

### **2. Adım:**

Yazılan programa göre adım – adım sırasıyla BOOLE cebri ilişkilerine göre işlemler yapılır. Bu işlemler yapılırken bir önceki adımda hesaplanan değerler bir sonraki adımda kullanılır fakat giriş belleğinde işlem anında değişen durumlar değerlendirilmez. Ancak çok hızlı değişen giriş işaretlerini kullanmak ve değerlendirmek için bazı PLC'lerde hızlı sayıcı girişleri kullanılır. Bu girişlere gelen işaretler program işletimi sırasında bir tampon bellekte yada ayrı bir sayıcıda depolanarak veri okuma noktasına gelindiğinde değerlendirilir.

### 3. Adım:

Bütün işler tamamlandıktan sonra hesaplanan değerler çıkış tamponuna (belleğine) yazılır ve tekrar birinci adıma dönülür.

PLC'lerdeki programın yürütülmesi sonsuz çevirimde çalışan bir program parçası gibi düşünülebilir. PLC'lerde bir çevrimin tamamlanması için geçen süreye tarama zamanı denir. Tarama zamanı PLC'de kullanılan mikro denetleyici frekansına, program içeriği ve uzunluğuna bağlıdır. Tarama zamanı kullanılan PLC türüne göre 3ms – 200ms arasında değişir. Örneğin 500 kelime (komut) kapasiteli ; 10 giriş ve 6 çıkışlı bir PLC' de[18] G/Ç tarama zamanı 2.6 ms , program işleme zamanı 12.4 ms'dir.

PLC'ler için geliştirilmiş olan programlama dilleri, kontaklı (röleli) kontrol devreleri tasarımcıların kolayca anlayıp uygulayabileceği şekildedir. Kontaklı kumanda devresinde herhangi bir programlama tekniği ile PLC'ye geçmek oldukça kolaydır.

#### 4.7 PLC'nin Kontrol Sistemindeki Yeri ve PLC ile Kontrol Sistemlerinin Oluşturulması

PLC, kontrol sistemi içinde işlemci eleman görevini üstlenir. PLC algılama elemanlarından gelen bilgileri işler ve çıkışlarına ikili veya analog işaret verir.

ALGILAYICILAR	PLC	KUMANDA ve İŞ ELEMENLARI
---------------	-----	-----------------------------

**Tablo 4.1 :** PLC'nin kontrol devresindeki yeri

PLC'ler genellikle ikili işaretleri işler. Ancak analog işaretleri işleyen PLC'lerde vardır. Analog işlem için, analog giriş çıkış olanağı tanıyan PLC ye gerek vardır.

Bir PLC ile kontrol sistemlerinin oluşturulması:

- 1) Kontrol probleminin tanımlanması, ifade edilmesiyle sorunun kağıda dökülmesi,
- 2) Sorunun çözümü için gerekli program veya fonksiyonların belirlenmesi,
- 3) Programın time diyagramı ve dalga şekilleriyle çalışırılığının kontrolünün yapılması,
- 4) Programın Diagrama aktarılması (LADDER, STL, FBD),
- 5) Programın yazılması olarak sıralanabilir.

## **4.8 PLC çalışması**

Bir PLC fonksiyon olarak başlıca iki bölümden meydana gelir. Bunlardan birisi donanım, diğeri yazılımdır. Bu iki öge birbirini tamamlayıcı nitelikte çalışır. Birbirinden ayrı olarak düşünülemez.

### **4.8.1 Donanım**

PLC'nin içinde bulunan elektronik ve yarı iletken elemanlardan veya kartlardan meydana gelir.[19] PLC ye gelen bilgiler bu elemanlar sayesinde alınır, işlenir ve çıkışa yönlendirilir.

#### **4.8.1.1 Donanım elemanları**

##### **➤ Giriş devresi**

Kontrol edilen makine üzerinden gelen, makinenin içinde bulunduğu durumu PLC ye ileten elemanların bağlandığı bölümdür. Ana işlevi dış dünyadan gelen kontak kapanması, analog gerilim ve akım değerleri gibi işaretleri almak ve CPU'nun (merkezi işlem birimi) kullanabileceği bir şekle getirmektir. Girişler ikili işaret ve analog işaret özellikli olabilir. Ancak PLC kullanımı sırasında girişlerin gerilim seviyesine dikkat edilmelidir. Giriş işaretleri şu elemanlardan gelebilir: Sınır anahtarları, el kumandalı düğmeler, basınç anahtarları, seviye anahtarları, seçici anahtarlar

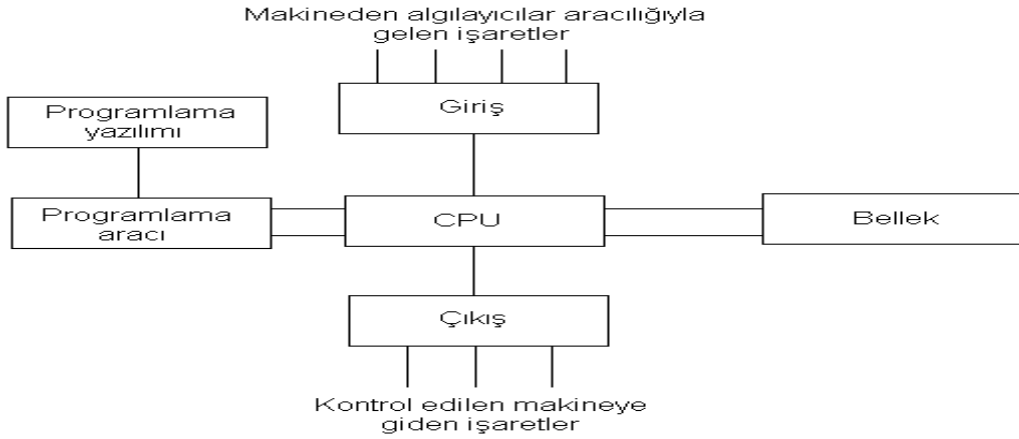
### ➤ Çıkış devresi

PLC de işlenen bilgilerin çıkış yeridir. Buradan çıkan işaretler kontrol edilen makinenin durumunu değiştirmeye yönelik olarak çalışır. Kontrol görevinin yerine getirilmesini sağlar.

PLC çıkışları röle çıkış, transistör çıkış, veya triyak çıkış olabilir. Hangi çıkışın tercih edilmesi gerektiği yapılacak kontrol işine bağlıdır.

- Röle çıkışı, genellikle çok hızlı olmayan sistemler için kullanılır.
- Tranzistör çıkışı çok hızlı sistemler için kullanılır.
- Triyak çıkışı ise yüksek güç için kullanılır.

Şekil 4.2’de bir PLC’nin çalışma blok şeması gösterilmiştir.



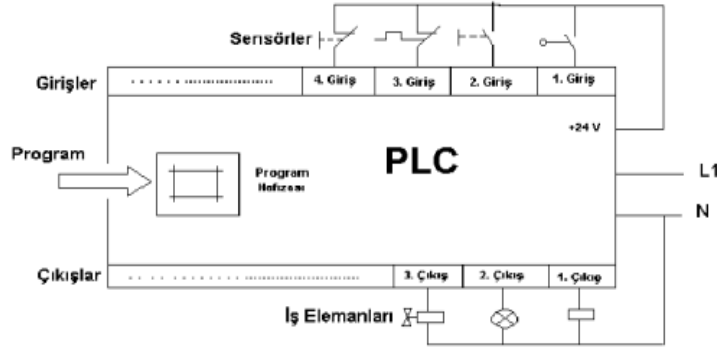
Şekil 4.2 : PLC blok şeması [20]

Otomasyon sistemlerinde genellikle aşağıda belirtilen elemanlar PLC çıkışlarından tetiklenir ;

- Selenoid valfler (elektrik kumandalı yönlendirme valfleri)
- Röleler ve kontaktörler
- Lambalar
- Göstergeler (LED , display)
- Motor sürücüleri (Driver)

PLC modülü ile birlikte ya da ayrı olabilir. PLC tipine göre giriş gerilimi 24 V DC veya 110 V, 220 V AC olabilir. Güç kaynağı; CPU, bellek ve giriş/çıkış modülleri için enerji kaynağı görevini üstlenir.

PLC bir bilgisayara benzetilirse; girişlerinde mouse ve klavye yerine basit giriş bağlantıları vardır. Yine çıkışlarında ekran yerine basit çıkış bağlantıları vardır. Girişlere bağlanan elemanlara sensör, çıkışlara bağlanan elemanlara da iş elemanı denir.



**Şekil 4.3 :** PLC Genel Blok Şeması [21]

Şekil 4.3'deki blok diyagramda gösterildiği gibi PLC sensörlerden aldığı bilgiyi kendine göre işleyen ve iş elemanlarına göre aktaran bir mikroişlemci sistemidir. Sensörlere örnek olarak, herhangi bir metali algılayan sensör, PLC girişine uygun gerilim vermede kullanılan buton ve anahtarlar verilebilir. İş elemanları için PLC çıkışından alınan gerilimi kullanan kontaktörler, bir cismi itme veya çekmede kullanılan pnömatik silindirleri süren elektro-valfler, lambalar uygun örneklerdir.

#### 4.8.2 Yazılım

Lojik bağlantılar dizisinden meydana gelen bu bölüm, makineye dışardan girilir. Donanımın yönlendirilmesine yönelik bilgiler yazılım sayesinde üretilir. Yazılımın değiştirilmesi çok kolaydır. Yazılımın değiştirilmesi ile kontrol programının davranışı çok kolaylıkla değiştirilebilir, PLC üstün ve kullanışlı kılan özelliklerden birisi zaten bu özelliğidir.

## 5. BİT LOJİK KOMUTLARI

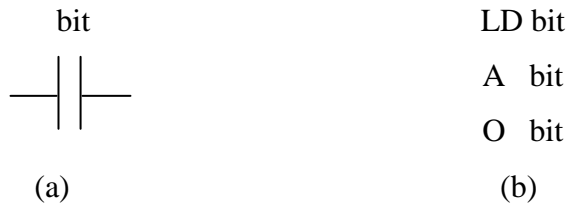
Kontak ve röle, ladder diyagram programlama komut setinin en önemli iki elemanıdır. Bir basamaktaki kontak sembolü, çıkışı kontrol etmek için değerlendirilmesi gereken bir koşulu temsil eder. Dahili ve harici çıkışlar röle sembolü ile gösterilir. Tüm kontak ve rölelerin bir adresi vardır.

### 5.1 Giriş

Kontrol lojiğine göre kontakların dizilimi farklı olabilir. Bir çıkışı kontrol etmek için kontaklar seri, paralel veya seri-paralel olarak dizilebilir. Soldan sağa en azından, bir yoldan lojik süreklilik varsa, basamak koşulu doğru denir. Basamak sonundaki röle enerjilenir. Eğer bir yoldan lojik süreklilik sağlanmıyorsa, basamak sonundaki röle enerjisiz kalır. PLC'deki röle ve kontak komutları ile klasik kumanda sistemindeki röle ve kontaklarla gerçekleştirilebilecek kontrol problemlerine çözüm getirilebilir. Bu komutlarla bellekteki belirli bit adreslerindeki veri değerleri kontrol edilerek dahili ve harici çıkışlar kontrol edilebilir.

### 5.2 Normalde Açık Kontak

Normalde açık kontak bit ile ifade edilen referans adresteki veriyi okur. İlgili adresten okunan bit değeri 1 ise kontak kapanır (lojik süreklilik sağlanır), 0 ise kontak olduğu gibi kalır (lojik süreklilik sağlanmaz). Kontak komutunun okumaya karşılık gelmektedir. Normalde açık kontak komutunun ladder diyagram ve komut dili gösterimi Şekil 5.1'de verilmiştir. [22]



**Şekil 5.1 :** Normalde açık kontakın (a) ladder diyagram ve (b) komut dili gösterimi

Komut dili ile programlamada, normalde açık kontak kullanıldığı yere göre Load, And veya Or ile gösterilir. Bu komutlar (LD bit, A bit, O bit) bit adresinin içeriğini yığının tepesine yükler. Ardı ardına veya paralel bağlanabilecek kontak sayısı PLC türüne göre sınırlandırılmıştır.

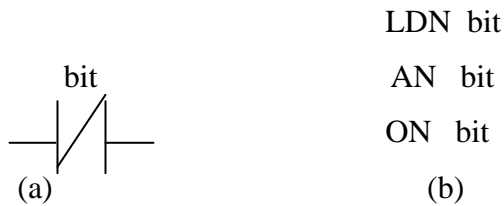
Tüm komutlarda kullanılan veri tipleri ile ilgili özellikler tablo 5.1’de açıklanmıştır.

Veri Tipi	Veri Büyüklüğü	Açıklama	Veri Aralığı
Bool (1 bit)	1 bit	Boolean	0-1
Bayt (8 bit)	8 bit	İşaretsiz bayt	0-255
Word (16 bit)	16 bit	İşaretsiz tam sayı	0-65535
Integer (16 bit)	16 bit	İşaretleli tam sayı	-32768 +32767
Dword (32 bit)	32 bit	İşaretsiz 32 bit tam sayı	$0+2^{32}-1$
Dint (32 bit)	32 bit	İşaretleli 32 bit tam sayı	$-2^{31}+2^{31}-1$
Real (32 bit)	32 bit	32 bit floating point	$-10^{38}+10^{38}$

**Tablo 5.1 :** Tüm komutlarda kullanılan veri tipleri ve özellikleri

### 5.3 Normalde Kapalı Kontak

Normalde açık kontakta olduğu gibi CPU, programın uygulanması sırasında normalde kapalı kontakın bit ile temsil edilen adresindeki veriyi okur. Okunan veri 0 ise kontak olduğu gibi kalır, enerji akışı (lojik süreklilik) sağlanır. CPU’nun okuduğu değer 1 ise normalde kapalı kontak açılır ve enerji akışı (lojik süreklilik sağlanmaz) kesilir.



**Şekil 5.2 :**Normalde kapalı kontakın (a) ladder diyagram ve (b) komut dili gösterimi





### Network 1

LD I0.0

AN I0.1

= M0.0

### Network 2

LD M0.0

= Q0.0

### Network 3

LDN M0.0

= Q0.1

I0.0 ve I0.1: Normalde açık kontaklar

Operatör I0.0 ve I0.1 butonlarına basmadığı sürece I0.0 ve I0.1 girişlerinden akım akmayacağından CPU bu girişlerin durumunu giriş imaj registerdeki I0.0 ve I0.1 adreslerine 0 olarak yazar.

7	6	5	4	3	2	1	0	
						0	0	I0
								I1

Program uygulaması öncelikle birinci basamaktaki I0.0 kontağından başlar. CPU, I0.0'ın değerini giriş imaj registerden okur. I0.0 adresinin içeriği 0 olduğundan kontak olduğu gibi kalacaktır. CPU, ikinci olarak, I0.1'in değerini yine giriş imaj registerden okur. I0.1 değeri 0 olduğundan bu kontak da olduğu gibi kalacaktır. M0.0 yardımcı rölenin durumu ile ilgili dahili bellek bitinde, I0.0, 0, I0.1, 0 olduğundan M0.0 rölesine lojik süreklilik sağlanmaz (enerji gelmez) ve CPU M0.0 adresine 0 değerini yazar.

İkinci basamakta (network 2), M0.0 kontağın durumu dahili bellek bitinin ilgi adresinden okunur. M0.0 adresinin içeriği 0 olduğundan, M0.0 kontağı olduğu gibi kalacak, Q0.0'a enerji akışı olmayacak, Q0.0 bitine 0 yazılacaktır. CPU, üçüncü basamaktaki programı uygularken M0.0 dahili bellek bitini okuyacaktır. M0.0, 0 olduğunda bu kontak olduğu gibi kapalı kalacak, Q0.1 rölesine enerji akışı olacaktır. CPU, Q0.1 röle komutu uygulandığında, röleye enerji geldiğinden Q0.1 bit adresine 1 yazar.

CPU, haberleşme ve kendi arıza testi işlemlerini gerçekleştirdikten sonra çıkış imaj registerdeki bitleri çıkış terminaline aktarır. Bu durumda  $Q0.0 = 0$ ,  $Q0.1 = 1$  olduğundan Q0.0 çıkışına bağlı eleman enerjilenmez, Q0.1 çıkışına bağlı eleman enerjilenir. I0.0 ve I0.1 butonlarına basılmadığı sürece program çalışması yukarıda açıklandığı gibi olur.

I0.0 butonuna basıldığında ve I0.1 butonuna basılmadığında devrenin çalışmasında ise; CPU taramanın başlangıcında giriş terminallerini okur ve giriş imaj registerin I0.0 ve I0.1 adreslerine aşağıdaki verileri yazar.

7	6	5	4	3	2	1	0		
							0	1	IO
									II

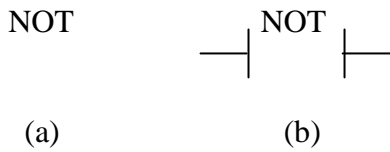
Program uygulama sırası birinci basamağın birinci elemanından başlar. Kontak okumaya karşılık geldiğinden CPU giriş imaj registerin I0.0 adresindeki değeri okur.  $I0.0 = 1$  olduğundan normalde açık olan kontak kapalı olarak değerlendirilir. İkinci olarak, I0.1 değeri imaj registerden okunur.  $I0.1 = 0$  olduğundan normalde kapalı kontak olduğu gibi kalır. I0.0 ve I0.1 kontakları kapalı olduğundan M0.0 rölesine lojik süreklilik sağlanır. Komut uygulaması M0.0 rölesine gelmiştir. Röleye lojik süreklilik sağlandığından CPU, M0.0 adresine 1 yazar.

İkinci satırda, CPU, dahili bellek bitinin M0.0 adresindeki bit değerini okur. M0.0 adresindeki veri 1 olduğundan bu kontak kapalı olarak değerlendirilir. Q0.0'a lojik süreklilik olduğundan, CPU, Q0.0 adresine 1 yazar. Komut uygulama sırası üçüncü basamaktaki normalde kapalı gelmiştir. M0.0'ın durumu dahili bellek bitinden okunur. M0.0 =1 olduğundan, CPU normalde kapalı olan kontağı açık olarak değerlendirir. Bu durumda, Q0.1 rölesine lojik süreklilik sağlanmaz. Röle yazmaya karşılık geldiğinden, CPU çıkış imaj registerin Q0.1 adresine 0 yazar.

Haberleşme ve arıza testlerinden sonra çıkış imaj registerdeki bitler çıkış terminaline aktarılır. Q0.0 =1 ve Q0.1 =0 olduğundan Q0.0 çıkışına bağlı eleman sürülür, Q0.1 çıkışına bağlı eleman sürülmez. Her iki giriş butonuna basıldığında M0.0 yardımcı rölesi 0, Q0.0 rölesi 0, Q0.1 rölesi 1 olur. Bu durumda sadece Q0.1 çıkışına bağlı eleman sürülür.

### 5.5 Değil (NOT) Kontakı

Değil kontakı enerji akışının durumunu değiştirir. Eğer enerji “değil” kontakına ulaşırsa, değil kontakından sonra enerji akışı devam etmez. Enerji akışı “değil” kontakına gelmez ise, “değil kontakından sonra devam eder. Değil kontakı komutunun merdiven ve komut dili gösterimi Şekil 5.5’de verilmiştir.

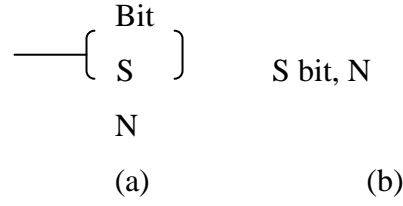


**Şekil 5.5:** Değil kontakının (a) komut dili ve (b) ladder diyagram gösterimi

Komut dilinde “değil” kontakı yığının tepesindeki değeri 0’dan 1’e veya 1’den 0’a değiştirir. Değil kontakının adresi yoktur.

## 5.6 Set Rölesi

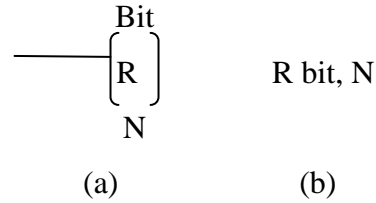
Set rölesi enerji akışı olduğunda, bit tarafından belirlenen adresten itibaren N tane bit 1 yapılır. Setlenen bu bölgeler resetlenmediği sürece 1 olarak kalır. Set rölesinin ladder diyagram ve komut dili gösterimi Şekil 5.6'de verilmiştir.



Şekil 5.6 : Set rölesinin (a) ladder diyagram ve (b) komut dili gösterimi

## 5.7 Reset Rölesi

Reset rölesine enerji akışı olduğunda, bit ile belirlenen adresten itibaren N tane biti reset (0) yapar. Eğer bit adresi T veya C olarak belirlenmişse, zamanlayıcı veya sayıcının o anki değerleri sıfırlanır. Reset rölesinin ladder diyagram ve komut dili gösterimi Şekil 5.7'de verilmiştir.



Şekil 5.7 : Reset rölesinin (a) ladder diyagram ve (b) komut dili gösterimi

## **6. PLC EĞİTİM SİMULATORÜNÜN TASARIMI ve UYGULAMASI**

### **6.1 GİRİŞ**

Sistemin kendi mantığı içinde çalışmasının modellenmesi, sonra da bu modele göre belirli durumlardaki davranışının gözlenebilmesi benzetim(simulation) alanının konusudur. Sistemin çalışmasının belirli şartlar altında tutulması, bir başka deyişle sistemin nasıl çalışması gerektiğinin gösterilmesi ise kontrol biliminin ilgi alanıdır. Ama her iki alanda da, o an için sistemin hangi durumda olduğunun anlaşılması için ölçme işleminin yapılması gerekir. Bu nedenle; sistemler hakkında çalışanlar yoğun olarak ölçme, modelleme, benzetim(simulasyon) ve kontrol alanlarında etkinlik gösterir.

Geliştirilen modellerin binlerce iteratif işleme dayalı benzetimlerini, bir yandan da kontrolünü sağlayan donanım ve yazılımlar son 40 yılda büyük gelişme göstermiştir.[23] Bunun son 20 yıla damgasını vuran gelişmesi ise bilgisayar ekranlarının grafik yeteneğinin artırılması ve Grafik Kullanıcı Arayüzü (GKA, Graphical User Interface, GUI) tasarımının ayrı bir çalışma alanı olarak ortaya çıkmasıdır. GKA, endüstride esnek mantık sistemleri yerine geçirilen bilgisayarlarla iletişim kurmak için de önemli bir araçtır

### **6.2 Ön Çalışmalar ve Sistem Analizi**

Bu tezde genel amaçlı bir PLC eğitim simulatörünün gerçekleştirilmesi ve Ladder diyagram tekniğiyle görsel ortamda bir programın yazılması amaçlanmıştır. Ladder diyagram tabanlı PLC programlamanın temel fonksiyonlarının belirlenmesi, her bir fonksiyonun sağlanabilmesi için gerekli alt program ve mantıkların oluşturulması, Ladder diyagramların işlem bloklarının görsel nesnelere olarak tanımlanması, sürükleyip bırak tekniğiyle program yazma arayüzünün geliştirilmesi, tanımlanmış sayıda giriş çıkış için simulasyon ortamında verilecek değerlerin gösterilmesi işlemlerini sağlayacak yazılımın hazırlanmasını hedeflenmektedir.

Bu tez çalışmasında, PLC Eğitim Simulator yazılımının altyapısını oluşturan;

- Dijital Giriş ve Çıkışlar
- Ladder Diyagram Programlama
- Komut Listesiyle Programlama
- Bit Lojik Komutları

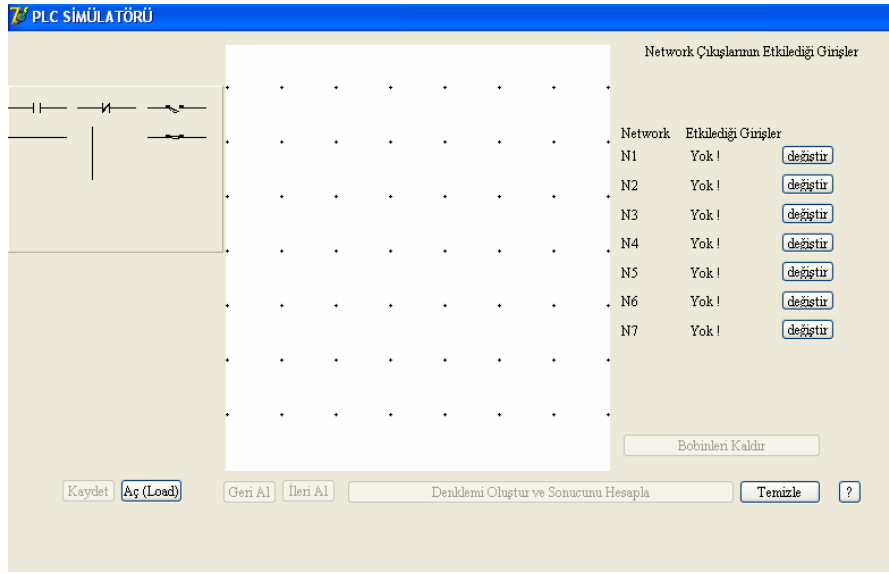
gibi temel fonksiyonlar ve mantıksal işlemler yazılımsal olarak Delphi programlama dillerinde görsel olarak tasarlanmıştır.

Bu çalışma PC tabanlı olarak tasarlandığından günümüzde yaygın işletim sistemi olan Windows kullanılmıştır. Kullanılacak yazılım geliştirme amacının hem GKA tasarımı bakımından güçlü hem de donanıma yakın program kodlarının yazılabilmesine elverişli olması gerekmektedir. Donanıma yakın ve mühendislik algoritmalarının kodlanmasında ve güçlü GKA tasarımı aletleri nedeniyle yazılım geliştirme aracı olarak Delphi tercih edilmiştir.

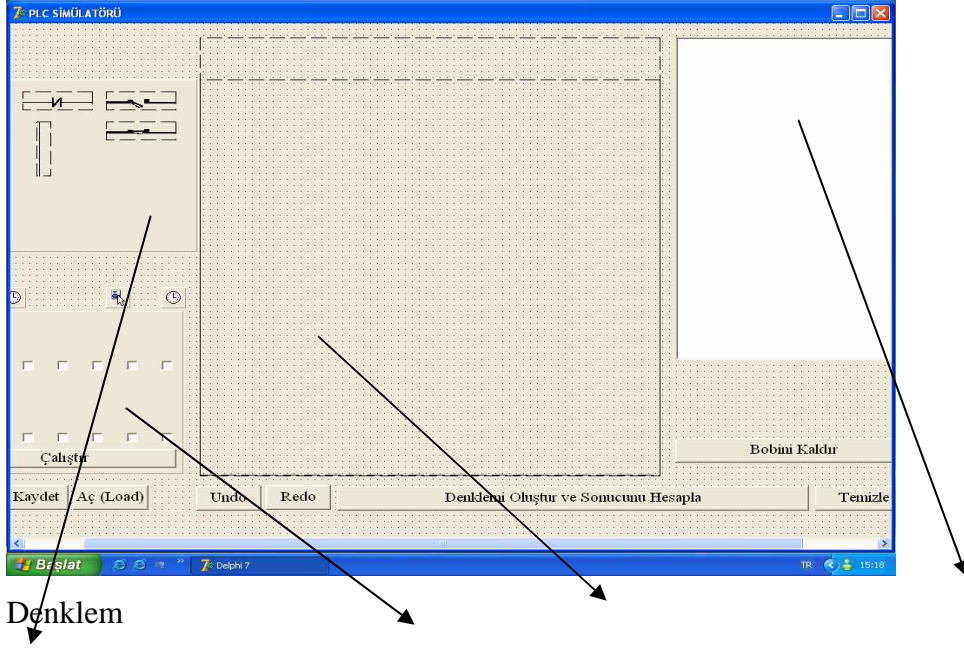
PLC programı, laboratuvar veya çalışma ortamında önceden çalıştırılabilir. Yazılan programın test edilmesi, değiştirilmesi, hataların bulunması için gerekli olan süre azalır. Klasik kumanda sistemleri ancak fabrika ortamında test edilebilir. Bu durumda sistemin işletmeye alınması için gerekli süre artar. PLC programının çalışması bilgisayar ortamında izlenebilir. Doğru ve hatalı çalışmalar anında tespit edilebilir. Görsel ortamda hataları bulmak kolaylaşır. Programın uygulanması esnasında oluşacak hatalar PLC tarafından algılanarak, operatör panelde hata mesajı üretmesi sağlanabilir. PLC'nin çalışma hızı çok yüksektir. PLC'nin çalışma hızı, birkaç milisaniye seviyesindeki tarama süresiyle ölçülür. Bu hız, mekanik rölelerden çok yüksektir. PLC'nin bütün bu pozitif değerlerinin daha detaylı gösterilmesi amacıyla bilgisayar ortamında görsel programlama yöntemi olan ladder diyagram programlama ile bu eğitim simülasyonu tasarlanmıştır.

### 6.2.1 PLC Eğitim Simulatör Birimleri (Simulator Ana Ekranı)

Program PLC birimlerinin eşlenikleri olan giriş çıkış üniteleri ve bunların çalışmalarının simüle edildiği alt birimler şeklinde tasarlanmıştır (Komponent Sürükle-Bırak Alanı, Ladder Diyagram Alanı, Denklem Kurma Bölümü, Input, Output, Kaydet, Aç, Geri Al, Yinele, Bobini Kaldır, Help ve Denklem oluştur-Sonucu hesapla). Bu birimler aşağıda anlatılmıştır.



Şekil 6.1 : PLC Eğitim Simulatorü Ekranı



Denklem

Simulasyon Ekranı (Ladder Diyagram)

Input – Output

Komponentleri sürükleyip bırak

Bölümü (Drag and Drop)

(Image3.....Image7)

## Şekil 6.2 : PLC Eğitim Simulatorü Ekranı Bileşenleri

### 6.2.1.1 Giriş Birimi

Açık-Kapalı çalışma mantığını kullanırlar. Bir Input biriminden diğer birimler için referans ve tetikleyici olarak yararlanılır. Birimin o anki durumu PLC kodu içerisinde kullanılır ve durumun değişmesi ile bağlı birimlerin etkilenmesi sağlanır.

### 6.2.1.2 Çıkış Birimi

Çıkış birimleri genel olarak durum bildiren çıkışlardır ancak Input'lar gibi giriş birimi olarak da kullanılabilirler. Bir Output birimi başka bir Output birimini giriş olarak kullanabilir.



### 6.2.1.3 Komponent Sürükle-Bırak Alanı (Drag and Drop)

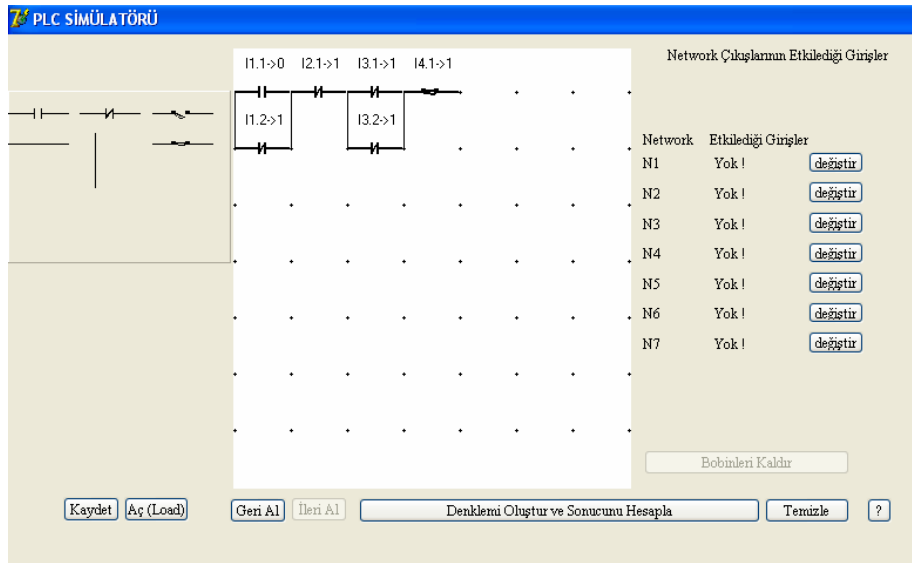
Bir panel üzerinde birden çok Image nesnesiyle (Image3.....Image7) tasarlanan bu alandan alınan komponentler Sürükle-Bırak yöntemiyle simulasyon alanına(Ladder Diyagram) yerleştirilir.

### 6.2.1.4 Ladder Diagram Alanı

Drag and Drop (Sürükle-Bırak) yöntemiyle alınan devre komponent elemanları bu alanda ladder(merdiven) diyagram şeklinde yer alırlar. Bu komponentler seri veya paralel olarak birbirlerine bağlanmışlardır.

### 6.2.1.5 Denklem Kurma Bölümü

Ladder diyagram alanında yapılan uygulamaların sonuçlarının hesaplanması bu bölümde detaylı olarak (seri ve paralel bağlantılar) gösterilmiştir.



Şekil 6.3 : PLC Eğitim Simulatorü Ekranında Seri-Paralel Bağlantı

## 6.3 Programın Çalışma Mantığı

Program derleme sırasında, yazılan kodu kendi yorumlayabileceği bir bağlantılar dizisine çevirir. Bu bağlantılar girişlerin ve değişkenlerin neleri etkileyeceğinin tutulduğu ve bağlantıların nasıl yapılacağını belirten çeşitli listelerde tutulur.

Komponent yerleştirilmek istenen yerler işaretlenir(diziler üzerinde) ve sonra dizi değerlerine bakarak çizim alanına gerekli çizimler yapılır. Hesapla prosedürü de tam olarak bunu yapar.

Çizim koordinatlarını dizilerde denk gelen indis değerlerinden hesaplanır ve tam olarak çizim işlemini yaptırmak için döngülerde Ciz procedure kullanılır.

Aşağıdaki Tablo 6.1’de [24] PLC Eğitim Simulatörü sisteminden beklentiler gösterilmiştir.

<b>Görsel</b>	<b>Mantıksal</b>
Birim çizimleri kütüphanesinin oluşturulması	Birim argümanları dosyalarının oluşturulması
Birimlerin görsel olarak sisteme dahil edilmesi	Bimirlere isim verilmesi
Birimlerin silinmesi, taşınması, adlarının yazdırılması	Birim özelliklerinin tayini
Çalışma dosyası işlemleri	Çalışma dosyası işlemleri
Drag- Drop (Sürükle-Bırak) işlemleri	Komponentlerin istenilen noktalara bırakılıp Ladder diyagramın oluşturulması
Denklem Alanı	Network bazında arka planda yapılan işlemlerin gösterimi
Ladder Diyagram alanına komponentlerin taşınması	Sürükle-bırak bölümünden alınan Image resimlerinin ladder diyagram alanına taşınması
CheckBox ların aktif edilmesi	CheckBox değerlerinin 1 ve 0 olarak değiştirilmesi
Çıkış elemanlarının elde edilmesi	Çıkış elemanının(sonuç) durumunun belirlenmesi
Panel üzerindeki I / O gösterimi	Simulasyon ekranından değerlerin 1 ve 0 olarak değiştirilmesi

<b>Proses resminin çizdirilmesi</b>	
Gösterge ekranının ayarlanması	Proseste yer alan birimlerin değişkenlerinden oluşan bağlantı dosyasının oluşturulması
İterasyon sayısı ve sınırlarının belirlenmesi	Göstergede kullanılacak değişkenlerin seçilmesi
	Simulator(benzetim) ve/veya kontrol yan programlarının seçilmesi
Benzetim ve/veya başlatılması	
Benzetim programının ürettiği verilerden seçilmişlerin anlık değerinin gösterilmesi	Benzetim ve/veya kontrolün sonu

**Tablo 6.1 :** PLC Eğitim Simulatörü sisteminden beklentiler

#### **6.4 Desteklenen Komut, Fonksiyon ve Prosedürler**

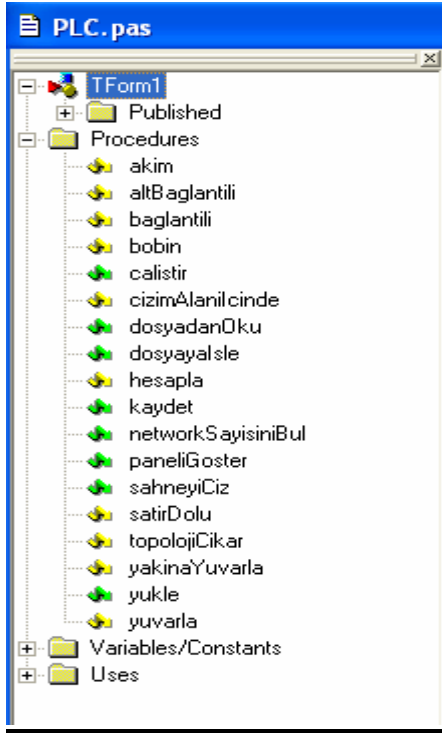
Kod ve veritabanı, prosedür ve fonksiyonlar kullanılarak bu proje hayata geçirilmiştir.

##### **6.4.1 Kullanılan Diziler**

YatayDolu, DikeyDolu, Eleman, Dikey Eleman, Durum Tabelası, Eleman Adı, Yatay, Denklem, Topoloji ve İsim dizileridir. Kullanılan bu diziler içinde en önemlileri; YatayDolu, DikeyDolu, Durum, Yatay ve İsim dizileridir.

##### **6.4.2 Kullanılan Procedürler ( Prosedures)**

Akım, Altbağlantılı, Bağlantılı, Bobin, Çalıştır, ÇizimAlanı İçinde, Dosyadan Oku, Dosyaya İşle, Hesapla, Kaydet, Network Sayısı Bul, Paneli Göster, Satır Dolu, Topoloji Çıkar, Simulasyon Alanı Çiz, Yakına Yuvarla, Yükle, Yuvarla prosedür dizinleri kullanılmıştır.



**Şekil 6.4** : PLC Eğitim Simulatörünün altyapısını oluşturan prosedürlerin ekran görüntüsü

- **Akim** : ( function akim(a : integer) : integer; )

Komponentlerin 'varsayılan' değerini(0/1) atar. Daha sonra elle de değiştirilebilecektir.

```
begin
```

```
result := 1;
```

```
case a of
```

```
1 : result := 0;
```

```
2 : result := 1;
```

```
3 : result := 0;
```

```
4 : result := 1;
```

```
5 : result := 1;
```

```
6 : result := 1;
```

```
end; //Case
```

```
end;
```

- **Altbağlantılı** : function altBaglantili(s1 : string; satir : integer) : bool;

Bir satırın aşağıda herhangi bir satırla ilişkisi kalıp kalmadığını tesbit etmek için kullanılıyor. Networklerin başlangıç ve bitiş noktalarını tesbitte, dolayısıyla da ayrı ayrı hesaplanmalarında kullanılmaktadır.

Bu procedure'ın çağırdığı alt procedure'lar veya function'lar :

- function baglantili(str : string; a : integer; b : integer) : bool;

- **Bağlantılı** : ( function baglantili(str : string; a : integer; b : integer) : bool; )

a ve b satırlarındaki komponentler arasında paralellik olup olmadığını tespit eder. Bunun için dikeyDolu dizisine bakar. İki uçta da kesintisiz paralellik çizgileri(dikeyKablo) ve iki satırda da birer komponent(dikey çizgiler arasında) var mı? Varsa True, yoksa False değerleri döner.

- **Bobin** : (Function bobin(x : integer; sonuc : integer) : integer; )

Bobin fonksiyonu; sonucSatiri değişkeniyle hesaplanan ilgili satıra çıktı 1 ise yeşil, 0 ise boş bir ampul yerleştirir. Ampul yerleştirme işlemi komponent (Image) yaratma ve yerleştirmedir.

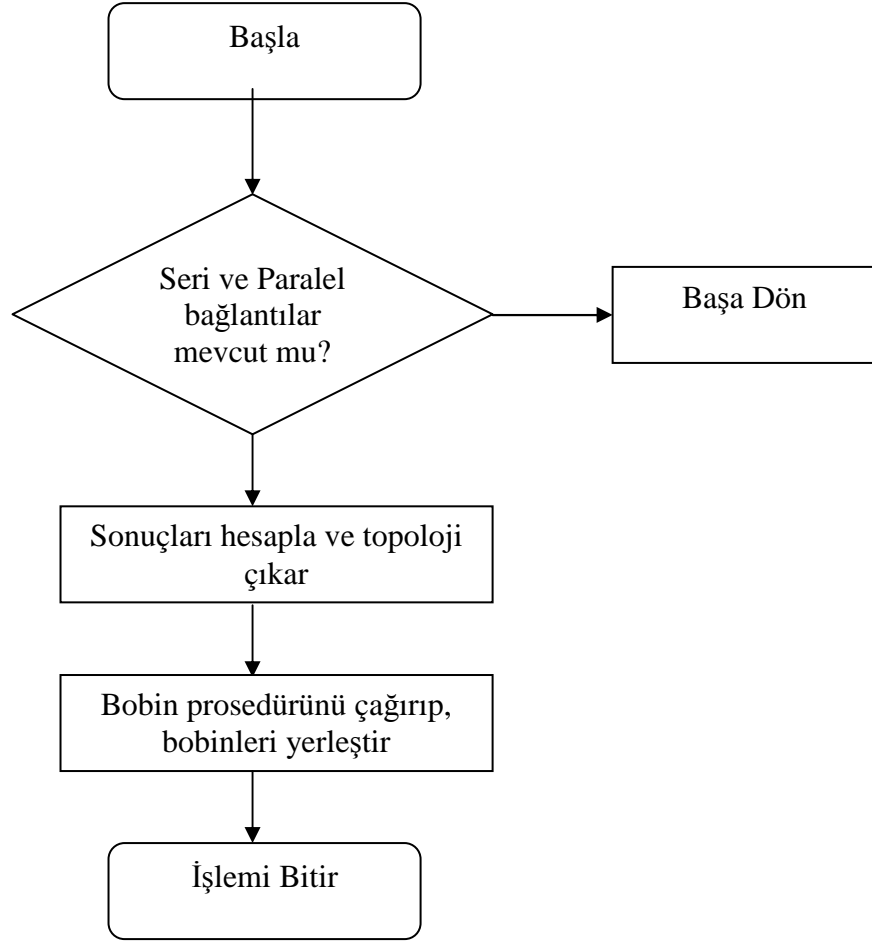
Ayrıca Bobin fonksiyonu, ilgili network'ün “**sonuc**” değerini, kendisini etkileyen bir başka network var ise(**cikisinGiriseEtkisi[]** dizisinden kontrolle tespit ediyor) o network'ün **sonuc** değeriyle **and** işlemine tabi tutmaktadır(**networkSonucu[]** dizisi).

- **Çalıştır** : procedure calistir(satir1 : integer; satir2 : integer);

Projede içindeki en önemli prosedürüdür. Bu procedure'ın içinde tüm paralel ve seri bağlantılar algılanıyor( ki buna “denklem çıkarma” dedim), hesaplanıyor ve topoloji çıkarıyor). Sonuçları hesaplar ve bobin procedure'mı çağırarak bobinleri yerleştirir.

Bu procedure'ın çağırıldığı alt procedure'lar veya function'lar :

- hesapla
- topolojiCikar
- bobin



**Şekil 6.5 :** Çalıştır prosedürünün akış şeması

Topolojiyi çıkarmak için denklem dizisinin değerlerini alınır. Dizi üzerinde oynama yapılacağı için ayrıca bir topoloji dizisi tanımlanmıştır. Topoloji çıkarma işlemi hesaplamalar esnasında gerçekleşir.

Çalıştır procedure'ında Durum dizisinin değerleri üzerinde değişiklikler yapılmaktadır.

**Calistir** procedure'de ayrıca her bir network'ün **başlangıç** ve **bitiş** satırları arasında işlem yapılıyor. Döngü içinde her network'ün başlangıç ve bitiş satırları tespit ediliyor (**Button1.Click** procedure'ında). Bu değerler parametre olarak calistir procedure'ına geçiliyor ( **calistir(satir1, satir2)** ). Böylece her bir network ayrı ayrı hesaplanıyor.

```
procedure calistir;
```

```
var
```

```
i, j : integer;
```

```
x, y, arananX, arananY : integer;
```

```
say : integer;
```

```
str1, str2 : string[6];
```

```
a, birinci, ikinci, topla, yedek : integer;
```

```
paralel : bool;
```

```
yatayDolu_yedek : array [1..7,1..7] of bool; //yatayda nerelerin dolu yedekleyecek
```

```
begin
```

```
for j := 1 to 7 do
```

```
for i := 1 to 7 do
```

```
denklem[i,j] := "
```

```
for j := 1 to 7 do
```

```
for i := 1 to 7 do
```

```
yatayDolu_yedek[i,j] := yatayDolu[i,j];
```

```
end.
```

```
for j := 1 to 7 do
```

```
for i := 1 to 7 do
```

```
if (denklem[i,j] <> "") then
```

```
topoloji[i,j] := '(' + topolojiCikar(denklem[i,j],j) + ')';
```

- **ÇizimAlanı İçinde** : ( function cizimAlaniIcinde : bool;)

Çizim alanının(Image1) içinde işlem yapılıp yapılmadığına yarayan prosedürdür. Bu prosedür, yerleştirme ve silme işlemlerinde kullanılmıştır.

- **Dosyadan Oku**

- **Dosyaya İşle** : (procedure dosyayaIsle(a : integer);)

Bu prosedür; Undo(Geri al) ve Redo(Yinele) işlemleri için Değerler dizilerde tutulup, sonradan okunmak üzere dosyaya aktarılıyor. (yatay, yatayDolu, dikeyDolu, durum, isim)

- **Hesapla** : ( function hesapla(str : string; satir : integer) : integer; )

Hesapla fonksiyonu verilen iki nokta arasındaki ve satır değerindeki 'SERİ' bağlı komponentlerin sonucunu hesaplar.

- **Topoloji Çıkar** : function topolojiCikar(string1 : string; satir : integer) : string;

Bu function sadece topolojiyi oluşturmak için gerekli olacak isimleri elde etmeye yarar. Asıl topoloji çıkarma işlemini **calistir** procedure'ı yapıyor. Bu procedure orada alt procedure olarak çağrılıyor.

- **Kaydet** : procedure kaydet;

Tüm dizilerdeki verilerin dosyaya kaydedilmesini sağlar, ayrıca görüntüyü de JPG olarak kaydeder.

- **Simulasyon Alanı Çiz** : ( procedure sahneyiCiz; )

Bu prosedür tüm programın ana mantığını oluşturur :



Komponent yerleřtirilmek istenen yerler iřaretlenir(diziler üzerinde) ve sonra dizi deęerlerine bakarak çizim alanına gerekli çizimler yapılır. Bu procedure da tam olarak bunu yapar.

Çizim koordinatlarını dizilerde denk gelen indis deęerlerinden hesaplanır ve tam olarak çizim iřlemini yaptırmak için döngülerde Ciz procedure kullanılır.

Bu procedure'ın çağırıldığı alt procedure'lar veya function'lar :

- Form1.Ciz

- **Yakına Yuvarla** : (function yakınaYuvarla(x : extended) : integer;)

Kullanılan bu prosedür; 0.8 ve üzerini yukarı, altını ařaęı yuvarlar. Yerleřtirmelerde ondalıkları yuvarlamaya yarayan prosedürdür.

- **Yükle**

- **Yuvarla** : (function yuvarla(x : extended) : integer;)

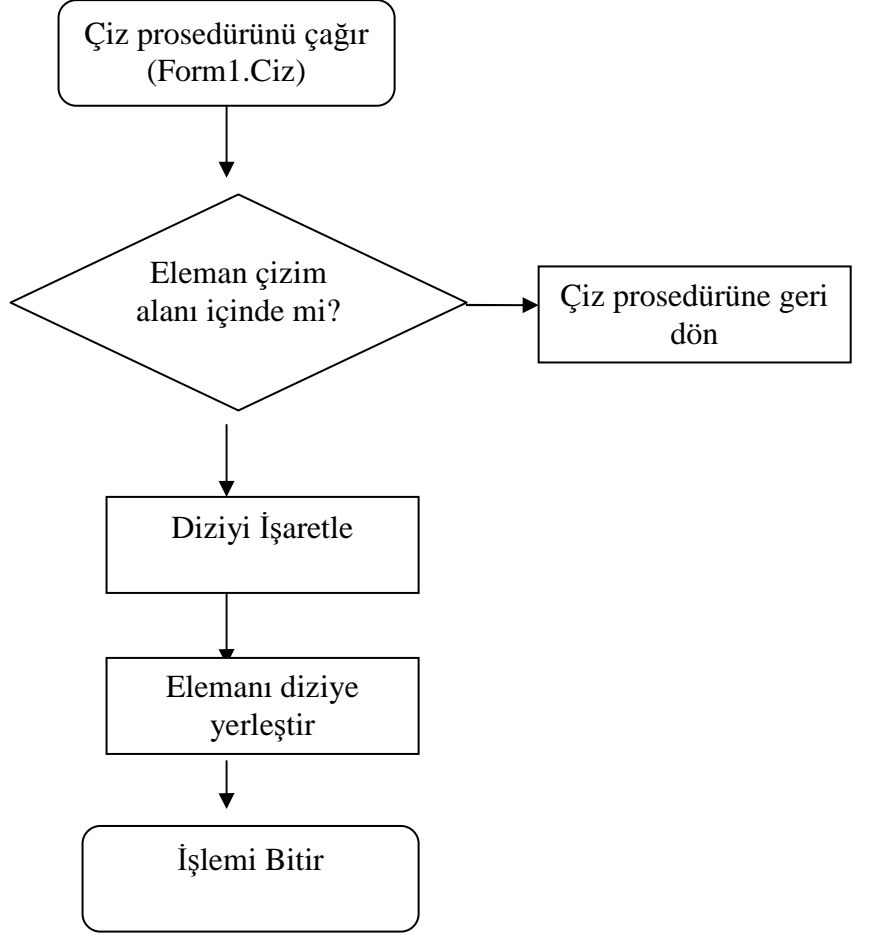
0.5 ve üzerini yukarı, altını ařaęı yuvarlar. Yerleřtirmelerde ondalıkları yuvarlamaya yarayan yuvarla prosedürüdür.

- **Yerleřtir** : (procedure TForm1.yerlestir(Sender : TObject);)

Bu prosedür diziler üzerinde sadece iřaretleme yapar ve daha sonra çizim iřlemleri yapılır. Bu prosedürün içinde **ciz procedure'ı** çağırılır. Ciz prosedürü; elemanın çizim alanının içinde olup olmadığını gösterir. İndeysse diziyi iřaretler ve elemanı diziyeye yerleřtirir.

Bu procedure'ın çağırıldığı alt procedure'lar veya function'lar :

- cizimAlaniIcinde
- Form1.Ciz
- dosyayaIsle
- sahneyiCiz



**Şekil 6.6 :** Yerleştir prosedürünün akış şeması

- **Edit :** procedure TForm1.Edit(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);

Yerleştirilmiş olan komponentler üzerinde değişiklik yapma (isim, durum) ya da komponenti seçip başka bir yere taşıma, silme...) işlemlerini yapar.

Bu procedure'ın çağırdığı alt procedure'lar veya function'lar :

- yuvarla
- dosyayaIsle
- sahneyiCiz

- **procedure TForm1.Timer1Timer(Sender: TObject);**

Ele alınan komponentin (aktifEleman) Mouse ile sürüklenebilmesi için Top ve Left koordinatlarının sürekli olarak Mouse'un koordinatlarına eşitlenerek tazelenmesi gerekir. Bu Timer, bu iş içindir.

- **procedure TForm1.Timer3Timer(Sender: TObject);**

Timer3, sökülen komponentin taşınması içindir. Koordinat bilgilerinin sürekli güncellenmesi için kullanılır.

- **procedure TForm1.Ciz(a : integer; b : integer; c : integer);**

a ve b : dizinin indis değerleri (Bu indis değerlerinde koordinatlar hesaplanacak)

c : komponent (Tanımlanan constant değerleri içerir)

Bu prosedür; komponentleri (Açık kontak, kapalı kontak, paralel bağlantı..) çizmeye yarar ve bazı komponentlerin üzerlerine durum tabelalarını da ekler.

- **procedure TForm1.simDeitir1Click(Sender: TObject);**

PopUpMenu nesnesinin (Form'un üzerindeki küçük nesnelere biri) prosedürdür. İlk procedure'da aynı isimde başka bir komponent var mı diye kontrol eder, varsa Mesaj kutusu açılıyor ( MESSAGEDlg ...). Evet tıklanırsa komponentin değeri diğerlerine eşitleniyor.

- **procedure TForm1.BobiniSil;**

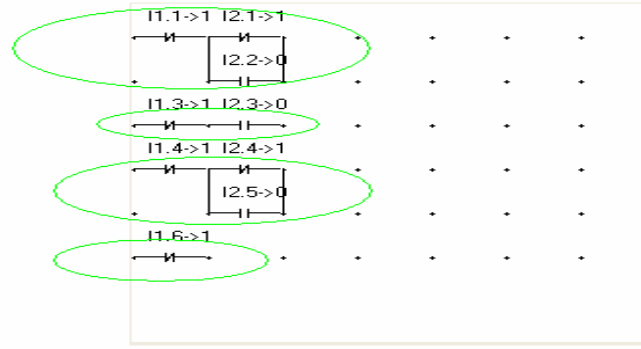
Bobinleri kaldırmaya yarar.

Bu procedure'ın çağırdığı alt procedure'lar veya function'lar :

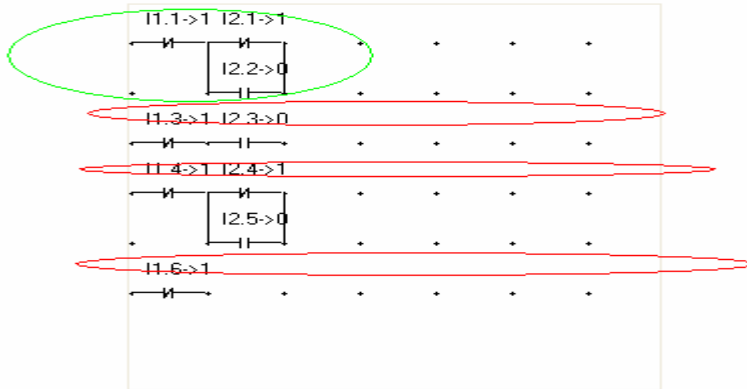
- networkSayisiniBul

- **procedure networkSayisiniBul;**

Adından da anlaşıldığı gibi kaç adet network olduğunu bulur. Çalışma mantığı şudur : **dikeyDolu[]** dizisini kontrol ederek birbirinden bağımsız satırları bulur. Birbirinden bağımsız her bir satır network demektir.



**Şekil 6.7 :** PLC Eğitim Simulatorü ekranında network(basamak) gösterimi(yeşil renk)



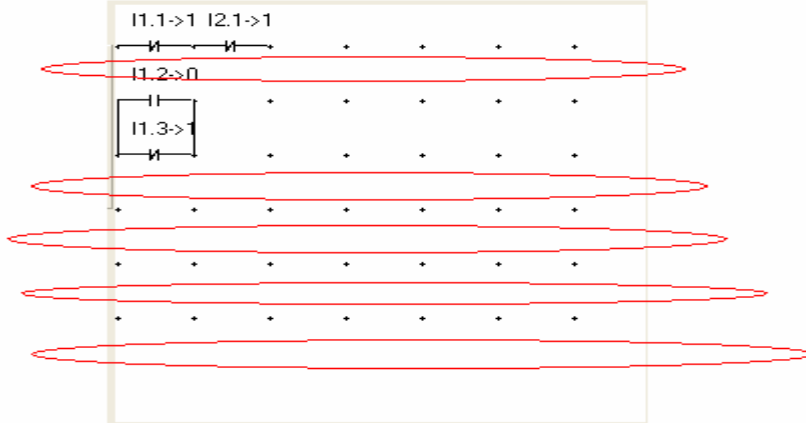
**Şekil 6.8 :** PLC Eğitim Simulatorü ekranında boşlukların gösterimi (kırmızı renk)

- **procedure PaneliGoster**(network : integer; network\_adi : integer);

Bu procedure da hangi çıkışın hangi girişi etkilemesini istediğimizi belirttiğimiz panelin (sağ taraftaki **Panel3**) açılmasını sağlar. Tabii bu panelin açılması için bazı değerler gereklidir. Hangi network üzerinde uğraştığımız ve kaç adet network olduğu. Çünkü bir network kendinden “sonra” gelen network’leri tetikleyebilir. Mesela 2. network üzerinde uğraşıyorsak ve 5 network varsa 3,4 ve 5 numaralı *CheckBox*’lar aktif (enabled) olmalıdır.

- **procedure SatirDolu**(satir : integer) : Bool;

**Satir** parametresi ile verilen satırda komponent olup olmadığına bakar yani bir satırın boş olup olmadığına bakar. Satırda tek bir komponent olsa dahi True değeri döner. Network sayısını ve network’lerin başlangıç / bitiş satırlarını bulmakta kullanılır.



**Şekil 6.9** : SatirDolu prosedürü ile ekran üzerindeki network sayısının bulunması

Kırmızı ile gösterilen yerler boşluklardır. Bu boşlukların olur olmaz hepsini network olarak algılamaması için bulunduğu her boşluğun üstünde komponent olup olmadığını kontrol etmelidir. Satırda en az bir komponent olması yeterlidir. Procedure da zaten

bu mantıkta işlemektedir. 3, 4 ve 5'in üzerinde hiç komponent yoktur. Dolayısıyla 2 adet network bulunmaktadır.

### 6.4.3 Bir örnek ile procedure ve function'ların çağrılmaları

Programı çalıştırdık. Mouse ile gittik, sol taraftaki toolbar'dan bir "Açık Kontak" (Image3) aldık. İlk olarak Image3.Click procedure'ı çalıştır (sıra 3062 - Image nesnesi yaratılır, sürüklemek için. Onun resim ve boyut özellikleri vs. verilir. Mouse imlecini takip etmesi için de Timer1.Timer çalıştırılır (sıra 2956). Daha sonra götürüp nesneyi Image1 (çizim alanı) içine bıraktığımızda Form1.Yerleştir procedure'ı çalışır.

Aynı şekilde başka komponentler de eklenebilir :

Eklenen komponentlerin üzerlerinde tıklanırsa Form1.Edit meydana gelir (sıra 2515). Form1.Edit komponentler üzerinde düzenleme yapılmasına olanak tanır.

**sağ tıklarınca :**

PopUp Menu devreye girer.

Pop Up Menüden

- **0 tıklanırsa** : procedure TForm1.N01Click(Sender: TObject); meydana gelir (sıra 3273).
- **1 tıklanırsa** : procedure TForm1.N11Click(Sender: TObject); meydana gelir (sıra 3334).
- **İsim Değiştir tıklanırsa** : procedure TForm1.simDeitir1Click(Sender: TObject); meydana gelir (sıra 3796).

**Sol tıklanırsa** : Komponent seçilip taşınabilir .

- **Geri Al** Butonu tıklanırsa: yapılanları geri alır.
- **İleri Al** Butonu tıklanırsa : yapılanları ileri alır.
- **Denklemi Oluştur ve Sonucu Hesapla** butonu tıklanırsa : çalıştır procedure'ını çağırır.
- **Temizle** butonuna tıklanırsa tüm alanı siler ve her şeyi sıfırlar.

## 6.5 PLC Eğitim Simulator Programının Genel Yapısı

Nesnelerin yerlerinin bilinmesi, değerlerin ( akım geçiriyor mu, geçiriyor mu) tutulması ve bağlantıların (özellikle de paralel) algılanması için iki boyutlu diziler kullanıldı. Bu dizilerin tamamı 7x7 boyutlarındadır.

Kontroller; diziler ile yapılmıştır. Belirli bir çizim alanı belirlenip o çizim alanının dışına çizim yaptırmaya çalışılmıştır. Hangi komponentlerin konulduğunu bir dizi tutarken, bir başka dizi de (tüm diziler iki boyutlu) paralel bağlantı yapılmış olan yerleri tutacak şekilde tasarlanmıştır.

Simulator ekranının sol bölümünden alınan komponentlere iki kez tıklandığı zaman yerine yerleşmektedir.

Simulator yazılımında en çok kullanılan diziler :

**Yatay** : İlgili yerde elemanın cinsi ( açık kontak, kapalı )

**Yatay Dolu** : İlgili yerde eleman olup olmadığı

**Dikey Dolu** : İlgili yerde paralel bağlantı olup olmadığı

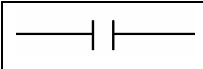
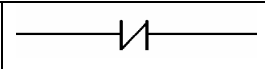
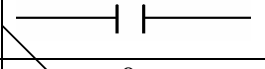
**Durum** : İlgili yerin akım geçirip geçirmediği belirtir.

Bunlar işlevsel olarak önemli oldukları için öncelikli olarak kullanılmıştır. Bunların haricinde “**isim**” dizisi de komponentlerin isimlerini tutar.

“**cikisinGiriseEtkisi**” , iki boyutlu bir dizidir. **Boolean** tiptedir. Adında anlaşılacağı gibi , herhangi bir çıkışın herhangi bir girişi etkileyip etkilemediğini tutmaktadır.

**cikisinGiriseEtkisi[1,2] := True;** // → 1 numaralı Network'ün çıkışı 2 numaralı network'ün girişini etkilemektedir.

**Örnek: (Çizim alanı : 7 x7)**

0	1						
							
							
	0						

**Şekil 6.10** : Çizim alanında dizilerin gösterimi

Yatay [ 1,1 ] : 1 (Açık kontak)

DikeyDolu [ 3,1 ] : True

DikeyDolu [ 2,1 ] : True

Yatay [ 2,1 ] : 0 (Kapalı kontak)

Yatay [ 2,2 ] : 1 (Açık kontak)

YatayDolu [ 1,1 ] : True

YatayDolu [ 2,1 ] : True

YatayDolu [ 2,2 ] : True

Durum [ 1,1 ] : 0

Durum [ 2,1 ] : 1

Durum [ 2,2 ] : 0

## 6.6 Toolbar'dan Bir Komponentin Seçilmesi

Bir komponenti seçmek için soldaki toolbara gidilir. Toolbar'dakiler düğme değil, "image" nesnelidir. Bunlar tıklandığında olan olayları görmek için tasarım halindeki formun (kod penceresinde F12' basarak ulaşılabilir) üzerine çift tıklanabilir. Bu Image'ler tıklandığında şu olaylar meydana gelir :



- Bir Image nesnesi yaratılır (a)
- Yaratılan nesneye gerekli özellikler verilir. (a.Left :=.....; v.b )
- Daha sonra en önemli aşama gelir : “komponentin sürüklenmesi”
- Komponentin sürüklenmesi, mantık olarak çok basittir. “Sürekli olarak”, sürüklenecek Image’in “top” ve “left” değerleri, Tmouse nesnesinin top ve left değerlerine eşitlenmelidir. Böylece mouse’u nereye sürüklersek Image’da oraya gelir. Bütün bu iş, **Timer1Timer** procedure’ında yapılır.
- Mouse’un X ve Y koordinat (Top, Left) değerlerine ulaşabilmemiz için bir Tmouse nesnesi yaratmak gerekir. (m:=Tmouse.Create)
- Daha sonra Timer1’i aktif hale getiririz. (Timer1.Enabled := True;)

## 6.7 Toolbar’dan alınan bir komponentin çizim alanına yerleştirilmesi

( procedure TForm1.yerlestir(Sender : TObject);)

Genel olarak;

- Önce çizim alanı içinde olup olmadığı kontrol edilir :

If ( cizimAlaniIcinde = True ) Then

begin

    \*

end

else

end;

- \* alanda şu kontrolleri yapar : Yerleştirilen komponentin bir dikey kablo (paralel bağlantı) olup olmadığına bakar. Dikey kablo ise dikeyDolu, diğer nesnelere ise yatayDolu, yatay, isim ve durum dizilerine işlenecektir. Ayrıca yerleştirme yapılmak istenen yerin boş olup olmadığı kontrol edilir. Boşsa yerleştirilir. (Bu procedure’da sadece işaretlemeler yapılır)

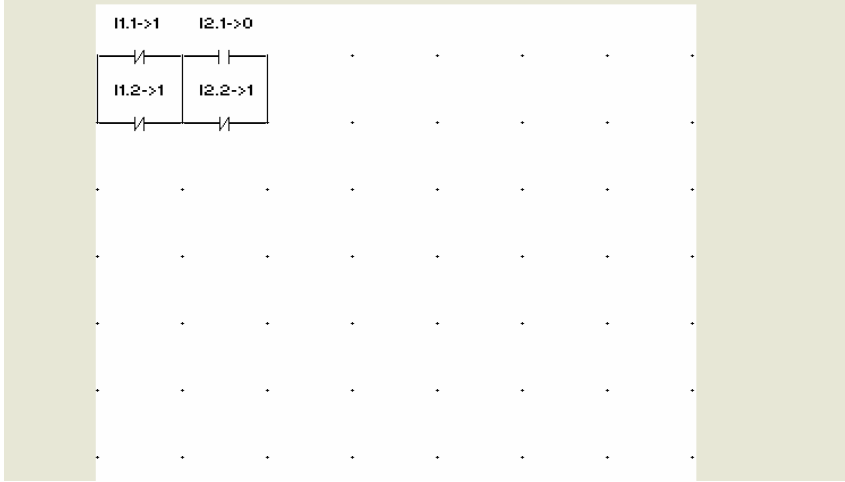
## 6.8 Yerleşik komponentler üzerindeki işlemler (başka bir yere taşıma, silme, adını veya durumunu değiştirme.....)

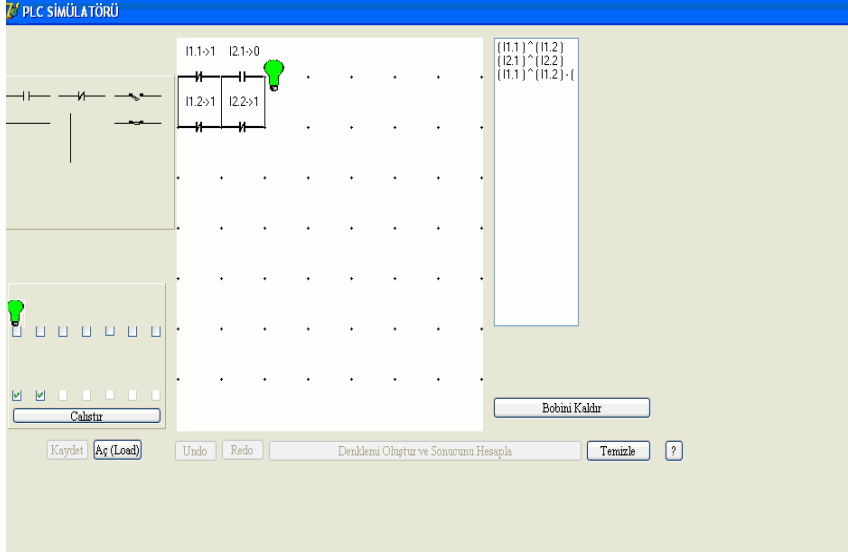
Edit, (TForm1.Edit)

Öncelikle mouse'un hangi tuşuna bastığımız, daha sonra da elimizde sökülü bir komponent (çizim alanından sökülmiş, toolbar'dan değil) olup olmadığı kontrol eder. Diğer tüm işlemler yok edilecek / yeri değişecek komponentin Top, Left gibi değerlerini güncellemek ve ilgili dizi (yatayDolu, yatay, durum v.b) değerlerini güncellemektir.

## 6.9 DENEMELER (UYGULAMALAR)

### Örnek 1 :





DikeyDolu [ 1,1 ] : True;      DikeyDolu [ 2,1 ] : True  
 DikeyDolu [ 3,1 ] : True  
 Yatay [ 1,1 ] : Kapalı Kontak;      Yatay [ 1,2 ] : Kapalı Kontak  
 Yatay [ 2,1 ] : Açık Kontak;      Yatay [ 2,2 ] : Kapalı Kontak

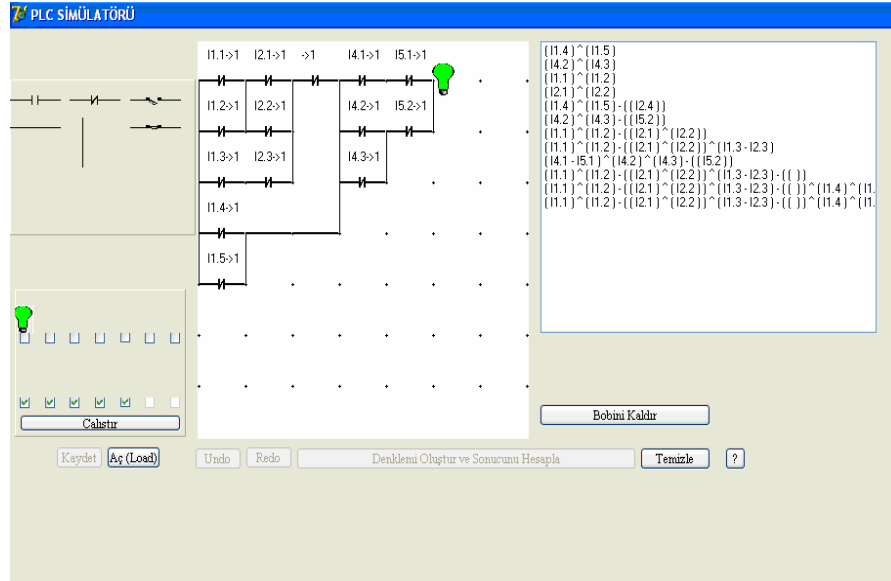
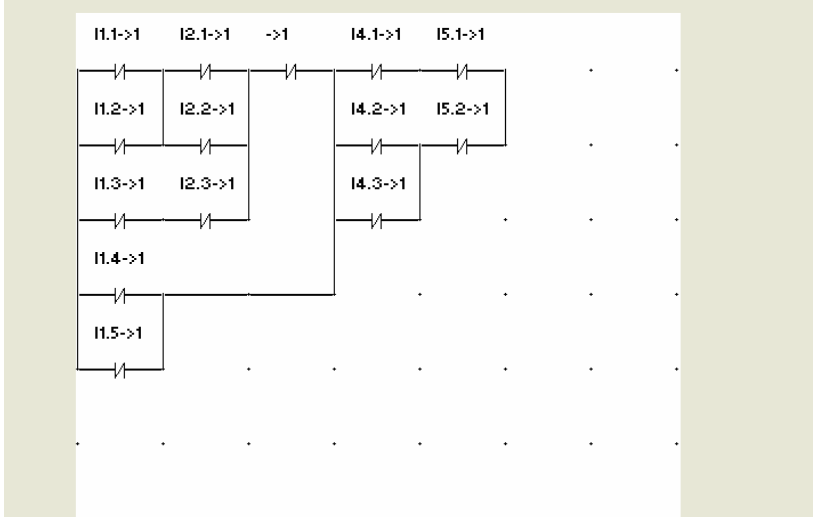
Bu örnekte; DikeyDolu, yatay ve durum dizileri kullanılmıştır. Durum dizileri hesaplama işlemleri için gereklidir.

İlk başta; paralelliklerin belirlenmesi için 1 ve 2 node ları arasındaki dikey çizgilere bakılır. Bu dikey çizgilerin mevcut olması gerekir. Eğer bu iki dikey çizgi arasında yatay çizgiler mevcutsa bir paralellikten bahsetmek mümkündür (paralellik için DikeyDolu çizgilerin başlangıç ve bitiş noktalarında component olması şarttır) ve denklem sondan başa doğru olarak hesaplanır. Paralellikler (koyu renkte) belirlendikten ve hesaplandıktan sonra seri bağlantılar şeklinde denklemin sonucu hesaplanır.

**Denklem :**

1	2	3
(1,2)	(2,3)	
(1,2)	(2,3)	

## Örnek 2 :



DikeyDolu [ 1,1 ] : True;      DikeyDolu [ 2,1 ] : True; DikeyDolu [ 3,1 ] : True

DikeyDolu [ 4,1 ] : True;      DikeyDolu [ 6,1 ] : True

Yatay [ 1,1 ] : Kapalı Kontak; Yatay [ 1,2 ] : Kapalı Kontak;

Yatay [ 2,1 ] : Kapalı Kontak; Yatay [ 2,2 ] : Kapalı Kontak;

Yatay [ 1,3 ] : Kapalı Kontak; Yatay [ 2,3 ] : Kapalı Kontak

Yatay [ 1,4 ] : Kapalı Kontak; Yatay [ 1,5 ] : Kapalı Kontak;

Yatay [ 4,2 ] : Kapalı Kontak; Yatay [ 4,3 ] : Kapalı Kontak;

Yatay [ 5,2 ] : Kapalı Kontak

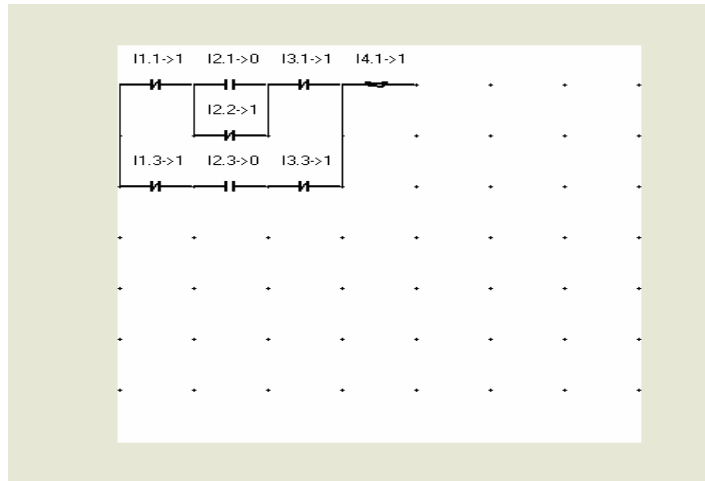
Bu örnekte; DikeyDolu, yatay ve durum dizileri kullanılmıştır. Durum dizileri hesaplama işlemleri için gereklidir.

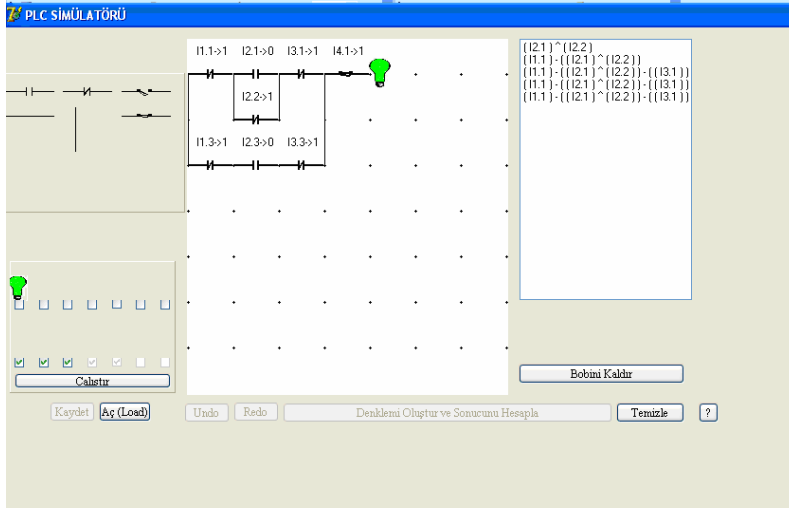
**Denklem :**

1	2	3	4	5	6	7
(1,2)	(2,3)	(3,4)	(4,6)			
(1,2)	(2,3)		(4,5)	(5,6)		
(1,3)			(4,5)			
(1,2)	(2,4)					
(1,2)						

Koyu renkte yazılan işlemler; paralelliklerin olduğu alanları belirtir. Bu paralellik işlemlerinin sonuçları hesaplandıktan sonra diğer işlemler ve seri bağlantılara bakılır. Denklem sonucunda eğer işlemin sonucu 1 ise lamba yanar yani akım geçişi olur, aksi durumda akım geçmezse lamba yanmaz. . Bu örnekteki denklemin sonucu 1 olduğu için çıkıştaki lambadan akım geçeceği için lamba yanar.

**Örnek 3 :**





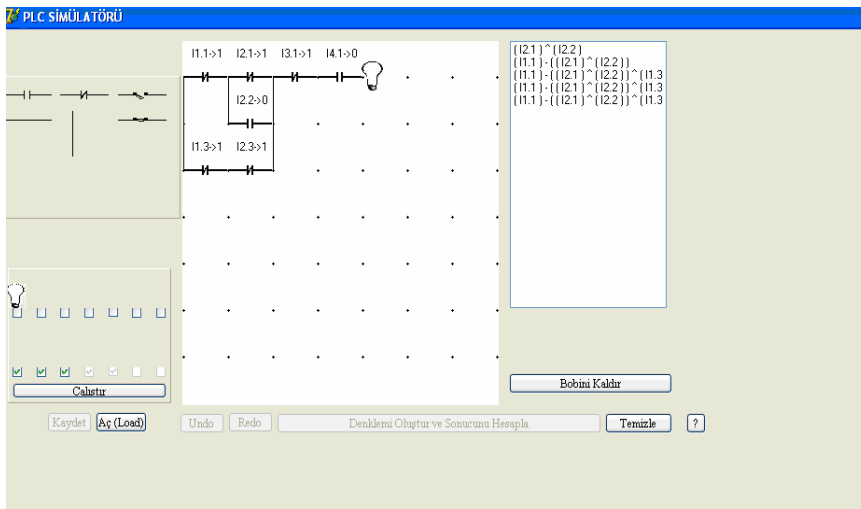
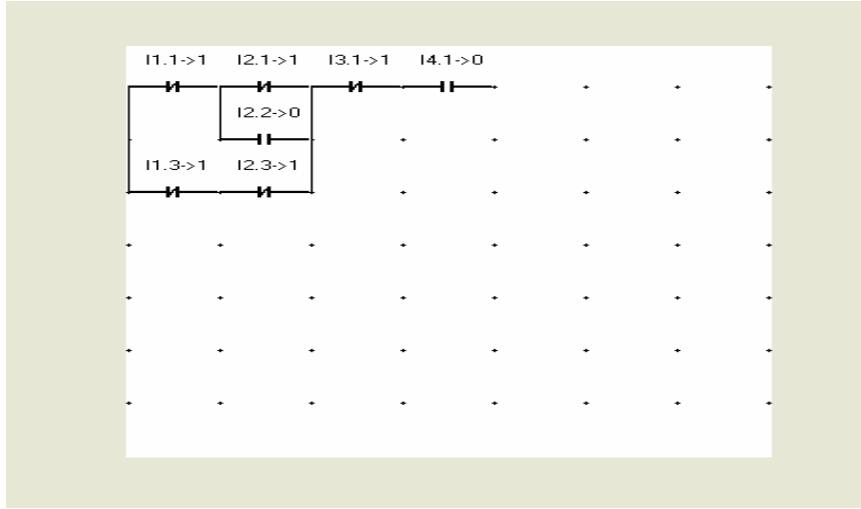
Bu örnekte; DikeyDolu, yatay ve durum dizileri kullanılmıştır. Durum dizileri hesaplama işlemleri için gereklidir.

**Denklem :**

1	2	3	4	5	6
(1,2)	(2,3)	(3,4)	(4,5)		
	(2,3)				
(1,4)					

Koyu renkte yazılan işlemler; paralelliklerin olduğu alanları belirtir. Bu paralellik işlemlerinin sonuçları hesaplandıktan sonra diğer işlemler ve seri bağlantılara bakılır. Denklem sonucunda eğer işlemin sonucu 1 ise lamba yanar yani akım geçişi olur, aksi durumda akım geçmezse lamba yanmaz. Bu örnekteki denklemin sonucu 1 olduğu için çıkıştaki lambadan akım geçeceği için lamba yanar.

## Örnek 4 :



Bu örnekte; DikeyDolu, yatay ve durum dizileri kullanılmıştır. Durum dizileri hesaplama işlemleri için gereklidir.

Koyu renkte yazılan işlemler; paralelliklerin olduğu alanları belirtir. Bu paralellik işlemlerinin sonuçları hesaplandıktan sonra diğer işlemler ve seri bağlantılara bakılır. Bu örnekteki denklemin sonucu 0 olduğu için çıkıştaki lambadan akım geçmez ve lamba yanmaz.

## Denklem :

1	2	3	4	5	6
(1,2)	(2,3)	(3,5)			
	(2,3)				
(1,3)					

## Örnek 5 :

PLC SİMÜLATÖRÜ

Network Çıktılarının Etkilediği Girişler

Network	Etkilediği Girişler	
N1	Yok !	değiştir
N2	Yok !	değiştir
N3	Yok !	değiştir
N4	Yok !	değiştir
N5	Yok !	değiştir
N6	Yok !	değiştir
N7	Yok !	değiştir

Kaydet Aç (Load) Undo Redo Denklemi Oluştur ve Sonucunu Hesapla Temizle ?

PLC SİMÜLATÖRÜ

Network Çıktılarının Etkilediği Girişler

Network	Etkilediği Girişler	
N1	Yok !	değiştir
N2	Yok !	değiştir
N3	Yok !	değiştir
N4	Yok !	değiştir
N5	Yok !	değiştir
N6	Yok !	değiştir
N7	Yok !	değiştir

Kaydet Aç (Load) Undo Redo Denklemi Oluştur ve Sonucunu Hesapla Temizle ?

Network1 - [[1.1]] - [[1.2]]  
 Network1 - [[1.1]] - [[1.2]] - [[1.3]]  
 Network2 - [[1.2]] ^ [[1.3]]  
 Network2 - [[1.2]] ^ [[1.3]] - [[1.2]] - [[1.2]]  
 Network3 - [[1.4]] ^ [[1.5]]  
 Network3 - [[1.4]] ^ [[1.5]]  
 Network3 - [[1.4]] ^ [[1.5]] - [[1.4]] ^ [[1.5]]  
 Network3 - [[1.4]] ^ [[1.5]] - [[1.4]] ^ [[1.5]] - [[1.3.4]]  
 Network4 - [[1.6]] - [[1.6]]  
 Network4 - [[1.6]] - [[1.6]] - [[1.6]]

Bu örnekte ise; farklı networklerdeki işlemlerin sonuçları (network 1, network 2, network 4, network 6, network 7) olmak üzere 5 network üzerinde yukarıdaki ekran görüntüsünde gösterilmiştir.



## Örnek 6 :

PLC SİMÜLATÖRÜ

Network Çıktılarının Etkilediği Girişler

Network	Etkilediği Girişler	
N1	Yok !	<a href="#">değiştir</a>
N2	Yok !	<a href="#">değiştir</a>
N3	Yok !	<a href="#">değiştir</a>
N4	Yok !	<a href="#">değiştir</a>
N5	Yok !	<a href="#">değiştir</a>
N6	Yok !	<a href="#">değiştir</a>
N7	Yok !	<a href="#">değiştir</a>

Bobinleri Kaldır

Kaydet Aç (Load) Undo Redo Denklemi Oluştur ve Sonucunu Hesapla Temizle ?

PLC SİMÜLATÖRÜ

Network Çıktılarının Etkilediği Girişler

Network	Etkilediği Girişler	
N1	Yok !	<a href="#">değiştir</a>
N2	Yok !	<a href="#">değiştir</a>
N3	Yok !	<a href="#">değiştir</a>
N4	Yok !	<a href="#">değiştir</a>
N5	Yok !	<a href="#">değiştir</a>
N6	Yok !	<a href="#">değiştir</a>
N7	Yok !	<a href="#">değiştir</a>

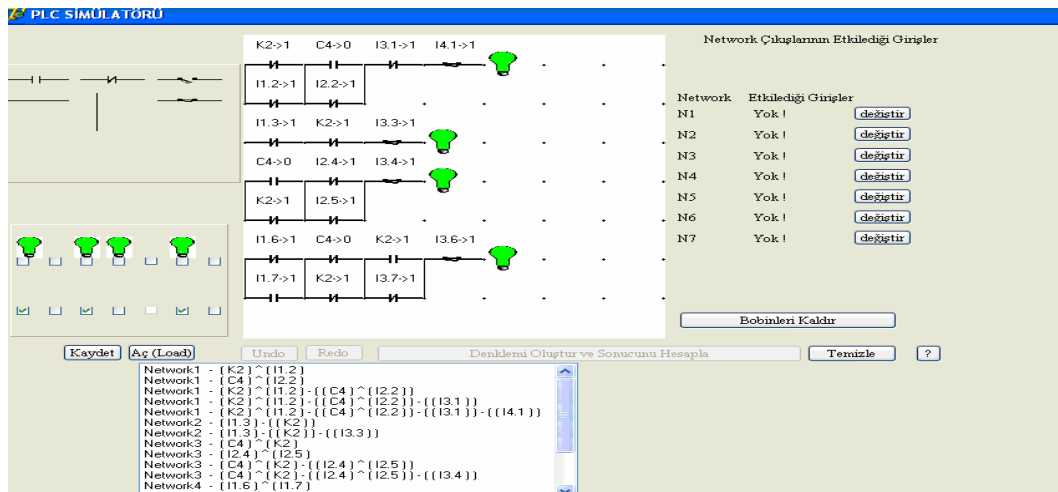
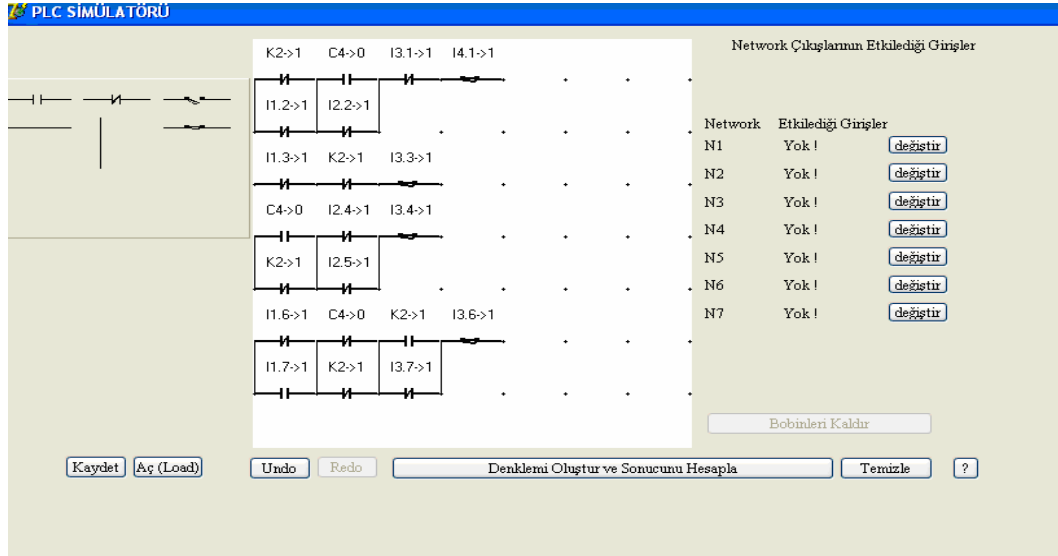
Bobinleri Kaldır

Kaydet Aç (Load) Undo Redo Denklemi Oluştur ve Sonucunu Hesapla Temizle ?

Network1 - ((I1.1)^(I1.2))  
 Network1 - ((I1.1)^(I1.2)-((I2.1)))  
 Network1 - ((I1.1)^(I1.2)-((I2.1))-((I3.1)))  
 Network1 - ((I1.1)^(I1.2)-((I2.1))-((I3.1))-((I4.1)))  
 Network2 - ((I3.1)-((I2.3)))  
 Network2 - ((I3.1)-((I2.3))-((I3.3)))  
 Network3 - ((I4.1)^(I1.5))  
 Network3 - ((I4.1)^(I1.5)-((I2.4)^(I2.5)))  
 Network3 - ((I4.1)^(I1.5)-((I2.4)^(I2.5))-((I3.4)))  
 Network4 - ((I6.1)^(I1.7))  
 Network4 - ((I6.1)^(I1.7))-((I2.6))

Bu örnekte; farklı networklerdeki işlemlerin sonuçları (network 1, network 3, network 4, network 6) olmak üzere 4 network üzerinde yukarıdaki ekran görüntüsünde gösterilmiştir.

## Örnek 7 :



Bu örnekte; farklı networklerdeki işlemlerin sonuçları (network 1, network 3, network 4, network 6) olmak üzere 4 network üzerinde yukarıdaki ekran görüntüsünde gösterilmiştir.

Sol taraftaki panelden değiştirme yapınca, eğer aynı adda başka componentler de varsa onları da düzeltiyor. Açık kontakta kapalı, kapalıysa açık yapıyor(şekilleri değiştiriyor). Röleleri de aynı şekilde... Ayrıca 0->1, 1->0 dönüşümlerinde de, aynı addaki diğer componentlerin de resimlerini değiştiriyor.

## 7. SONUÇLAR

Endüstriyel üretim süreçlerinde bilgisayarların etkin olarak kullanımı PLC'lerle başlamıştır. PLC'lerin kullanılmaya başlanmasından sonra çok karmaşık kontrol süreçleri endüstriyel bilgisayar da denebilecek bu elemanlarla gerçekleştirilmeye başlanılmıştır. Klasik kontrol panosu içinde bulunan elektromekanik elemanların (röleler , zaman röleleri , sayıcılar) hemen hepsi PLC içinde bulunmaktadır. Bu elemanların kullanılması çok masraflı olacağından ve bu düzeneklerin kullanımında zaman kaybı fazla olacağından bu düzeneklerin yerini PLC'ler almıştır. Yarıiletken cihazların, analog kontrolörlerin ve hatta mikrobilgisayarların yerine başarılı bir şekilde kullanılmaya başlanmıştır.

Programlanabilir lojik kontrolörler endüstriyel donanımlı bilgisayarlardır. PLC'ler kolayca programlanabilecek ve herhangi bir donanım değişikliğine ihtiyaç olmaksızın uygulama gereksinimlerdeki değişikliklere cevap verebilecek şekilde geliştirilmişlerdir.

PLC'lerin yetenekleri hızlı bir şekilde büyümektedir. PLC uygulamalarındaki eğitim görme gereksinimi teknik okul, üniversite ve endüstri alanlarında artmaktadır. Kıyasıya rekabetin yaşandığı günümüzde, endüstriyel kuruluşların varlıklarını sürdürebilmeleri için verimli, esnek ve üretim maliyetlerinin düşük olması gereklidir. Klasik kontrol ve kumanda sistemlerin kullanımıyla bu hedefe ulaşmak oldukça zor görülmektedir.

Endüstriyel otomasyon sektöründe yerlerini korumak isteyen işletmelerde Programlanabilir Lojik Kontrolörlerin kullanımı her geçen gün artmaktadır. Başlangıçta tek bir makineyi kontrol elemanı olarak üretilen PLC'ler günümüz endüstrisinin kullandığı çok amaçlı bir kontrol elemanıdır. PLC'ler diğer kontrol elemanları ve bilgisayar ağ sistemleriyle uyum içinde çalışmaktadır.

Bu tezde genel amaçlı bir PLC eğitim simulatörünün gerçekleştirilmesi ve Ladder diyagram tekniğiyle görsel ortamda bir programın yazılması amaçlanmış ve gerçekleştirilmiştir. Ladder diyagram tabanlı PLC programlamanın temel

fonksiyonlarının belirlenmesi, her bir fonksiyonun sađlanabilmesi iin gerekli alt program ve mantıkların oluřturulması, Ladder diyagramların iřlem bloklarının grsel nesnelere olarak tanımlanması, srkle bırak tekniđiyle program yazma arayznn geliřtirilmesi, tanımlanmış sayıda giriř ıkıř iin simulasyon ortamında verilecek deđerlerin gsterilmesi iřlemlerini sađlayacak yazılım Delphi programlama dilinde gerekleřtirilmiřtir.

## KAYNAKLAR

- [1] “Programmable Logic Controllers Principles and Applications” , Fourth Edition, p.p 62, John W. Webb, Ronald A.Rels, Prentice Hall 1999.
- [2] “Telemecanique TSX Micro PLC (Programmable Controller System Manual” , Edition 2003
- [3] Bayazıt Hasan, “Uygulamalı PLC Programlama ve Operatör Panel Konfigürasyonu”, sayfa :14, Nobel Yayınları, 2004.
- [4] “Programmable Logic Controllers”, Frank D. Petruzella, Macmillian/ Mc Graw-Hill 1992
- [5] “Telemecanique TSX Nano PLC (Programmable Controller System Manual” , Edition 2003
- [6] “Simatic S7-200 Programmable Controller System Manual” , Edition 03/2000
- [7] “Telemecanique TSX Micro PLC (Programmable Controller System Manual” , Edition 2002
- [8] “Simatic S7-200 Applications” “Tipsad Trick”, Status 10/95
- [9] “SIMATIC HMI OP7, OP17 Operator Panel”, Equipment Manual, Release 04/99
- [10] “Communication User’s Manual” , Siemens, Release- 11/97
- [11] Recep Çetin, S7-200 İle PLC Otomasyon, 2005.
- [12] “Otomasyon için Komponent ve Sistem Katalogu”, Omron
- [13] “Mitsubishi Programmable Controller Programming Manual”, March 1987
- [14] Kurtalan, S., “PLC ile Endüstriyel Otomasyon”, Birsen Yayın Evi, İstanbul,1999.
- [15] “Programmable Logic Controllers and Their Engineering Applications” Alan J. Crispin, McGraw-Hill Book Company 1990
- [16] “Analog and Digital Control Systems”, Leonard Sokoloff, Prentice Hall, 1998
- [17] <http://www.plcprogramlama.com/plcs.htm> (Erişim tarihi : 17.01.2007)
- [18] <http://www.mikroelektronika.co.yu/english/product/books/PLCbook> (Erişim Tarihi : 01.05.2007)

- [19] [www.automation.siemens.com/simatic/industriessoftware/html\\_76/plcsim.htm](http://www.automation.siemens.com/simatic/industriessoftware/html_76/plcsim.htm)  
(Eriřim Tarihi : 03.04.2007)
- [20] “Telemecanique TSX Micro PLC (Programmable Controller System Manual”,  
Edition 2002
- [21] “Simatic S7-200 Programmable Controller System Manual”, Edition 07/2003
- [22] <http://www.plcdev.com/catalog> (Eriřim Tarihi : 11.12.2006)
- [23] Esin E. Murat, “C Builder ile Grafik ve Çizim Uygulama Paketi Tasarımı ve  
Geliřtirilmesi”, Bileřim Yayınevi, Temmuz 2005
- [24] Esin E. Murat, PC Tabanlı Proses Kontrol İin Genel Amalı Bir Trke  
SCADA Ortamı, GYTE Arařtırma Fonu, 96-B-03-04-01-2000
25. Delphi 7, Nihat Demirli Yksel İnan, Prestige Yayınları, Ankara 2003
26. Visual C#. Net 2003, Nihat Demirli Yksel İnan, Prestige Yayınları, Ankara  
2003
27. Watson Karli, Beginning Visual C#, Wiley Publishing, 2004
28. Schildt Herbert, Herkes iin C#, Mc Graw Hill, Alfa Yayınları, Eyll 2002
29. [www.knowledgestorm.com](http://www.knowledgestorm.com) (Eriřim Tarihi : 07.02.2007)
30. [www.ticalc.org/archives/files/fileinfo/234/23490.html](http://www.ticalc.org/archives/files/fileinfo/234/23490.html) (tiplc.zip) (Eriřim Tarihi  
: 24.04.2007)
31. <http://www.plcsimulator.net> (Eriřim Tarihi : 25.04.2007)
32. [www.smartdraw.com](http://www.smartdraw.com) (Eriřim Tarihi : 14.02.2007)
33. [www.directindustry.com](http://www.directindustry.com) (Eriřim Tarihi : 19.03.2007)

## ÖZGEÇMİŞ

1974 yılında Konya'da doğdu. İlk, orta ve lise öğrenimini İstanbul'da tamamladı. Doğu Akdeniz Üniversitesi Elektrik-Elektronik Mühendisliği bölümünden 1999 yılında mezun olduktan sonra aynı yıl Maltepe Üniversitesi Mühendislik-Mimarlık fakültesinde Araştırma Görevlisi olarak çalışmaya başladı ve 2002 yılında İşletme Yüksek Lisans (MBA) programını bitirdi. 2004 yılında askerlik hizmetini tamamladıktan sonra 2005 yılında Maltepe Üniversitesi Meslek Yüksek Okulunda Öğretim Görevlisi olarak çalışmaya başladı ve aynı yıl Maltepe Üniversitesi Bilgisayar Yüksek Lisans programına başlayarak eğitim ve akademik hayatını birlikte yürütmeye başladı.

Akademik hayatının ilk yıllarından itibaren PLC konusunda çalışmaya başlamış ve bu tez ile birlikte PLC Simulatorü konusunda bir görsel bir yazılım ve kaynak hazırlamıştır. PLC Simulatör konusunu seçmesindeki neden; günümüzde PLC'lerin yetenekleri hızlı bir şekilde büyümesi ve PLC uygulamalarındaki eğitim görme gereksinimi teknik okul, üniversite ve endüstri alanlarında artmasıdır.