

**T.C.  
MİLLÎ EĞİTİM BAKANLIĞI**



**MEGEP**

**(MESLEKÎ EĞİTİM VE ÖĞRETİM SİSTEMİNİN  
GÜÇLENDİRİLMESİ PROJESİ)**

**ELEKTRİK ELEKTRONİK TEKNOLOJİSİ**

**PLC PROGRAMLAMA TEKNİKLERİ**

**ANKARA 2007**

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

# İÇİNDEKİLER

AÇIKLAMALAR .....	iv
GİRİŞ .....	1
ÖĞRENME FAALİYETİ-1 .....	3
1. YAZILIMI SİSTEME YÜKLEMEK .....	3
1.1. Bilgisayar Haberleşme Portları .....	3
1.1.1. Paralel Portlar .....	3
1.1.2. Seri Portlar .....	3
1.1.3. USB (Universal Serial BUS ) .....	4
1.2. Bilgisayar ile PLC Arasındaki Haberleşme Ayarları .....	5
1.2.1. Haberleşme Elemanları .....	5
1.2.2. PLC Ayarları .....	5
1.2.3. Com ve Port Ayarları .....	6
1.3. Kontrol Programının Plc'ye Yüklenmesi .....	8
UYGULAMA FAALİYETİ .....	9
PERFORMANS DEĞERLENDİRME .....	10
ÖĞRENME FAALİYETİ-2 .....	11
2. PLC İÇİN KONTROL PROGRAMININ YAZILMASI .....	11
2.1. PLC'de Programlama Mantığı ve Kontrol Programı Türleri .....	11
2.1.1. PLC'de Program İşleme Mantığı .....	11
2.1.2. Lineer (Doğrusal) Programlama Mantığı .....	12
2.1.3. Yapısal Programlama Mantığı .....	12
2.2. PLC Program Yazılım Dilleri .....	13
2.2.1. Kontak Plan (LADDER Plan) .....	13
2.2.2. Fonkiyon Plan (FBD) .....	13
2.2.3. Deyim Listesi (STL) .....	14
2.3. PLC Programlama Yazılımının Bilgisayara Kurulması .....	14
2.4. Program Menüleri ve Açıklanması .....	20
2.4.1. Araç Çubukları .....	21
2.4.2. File (Dosya) Menüsü .....	21
2.4.3. Edit Menüsü .....	23
2.4.4. View Menüsü .....	25
2.4.5. PLC Menüsü .....	28
2.4.6. Debug Menüsü .....	30
2.4.7. Tools Menüsü .....	32
2.4.8. Windows Menüsü .....	33
2.4.9. Help Menüsü .....	33
2.4.10. Kısayol Menüsü .....	34
2.5. Program Pencereleeri ve Genel Kullanımı .....	34
2.5.1. Programlama Editörü Penceresi .....	34
2.5.2. Symbol Table ( Sembol tablosu) Penceresi .....	35
2.5.3. Data Block ( Data bloğu) Penceresi .....	36
2.5.4. Status Chart (Durum Tablosu) Penceresi .....	36
2.5.5. Cross Reference (Çapraz Referans) Penceresi .....	37
2.5.6. System Block (Sistem bloğu) Penceresi .....	38
2.6. PLC Tipinin Seçilmesi .....	38

2.7. Hafıza Alanları ve Adreslenmeleri.....	39
2.7.1. Bit, Bayt ve Word kavramları.....	39
2.7.2. Girişler.....	41
2.7.3. Çıkışlar.....	43
2.7.4. M Hafıza Alanları.....	44
2.7.5. V Hafıza (Data) Alanları.....	45
2.7.6. L (Lokal) Hafıza Alanları.....	46
2.7.7. AC (Akümülatör) Hafıza Alanları.....	47
2.7.8. HC (Hızlı sayıcı HSC) Hafıza Alanları.....	47
2.7.9. S (Sıralayıcı kontrol rölesi SCR ) Hafıza Alanı.....	48
2.7.10. Analog Giriş ve Analog Çıkış Hafıza Alanları.....	49
2.7.11. Özel Hafıza Alanları.....	49
2.7.12. Zamanlayıcılar.....	50
2.7.13. Sayıcılar.....	52
2.7.14. Endirekt (dolaylı) Adresleme.....	53
2.8. Temel Programlama Komutları.....	53
2.8.1. Start, Stop ve Çıkışlar.....	53
2.8.2. Çıkış Atama ( = ) Komutu.....	57
2.8.3. LD ve LDN Kontak Yükleme Komutları.....	58
2.8.4. NOT Değilleme.....	58
2.8.5. A (VE), AN (VE DEĞİL) Komutları ve Uygulaması.....	59
2.8.6. O (VEYA), ON (VEYA DEĞİL) Komutları ve Uygulaması.....	59
2.8.7. Birden Fazla Komut Bloğu Arasında (ALD ) VE İşlemi ve Uygulaması.....	60
2.8.8. Birden Fazla Komut Bloğu Arasında (OLD) VEYA İşlemi ve Uygulaması.....	61
2.8.9. LPS, LPP, LRD ( Lojik yığın ) Komutları.....	62
2.8.10. SET, RESET Komutları ve Uygulaması.....	62
2.8.11. Giriş Sinyalinin Çıkan Kenarına Göre Çıkış Alma.....	64
2.8.12. Giriş Sinyalinin İnen Kenarına Göre Çıkış Alma ve Uygulaması.....	65
2.8.13. Zamanlayıcı (Timer) Komutu.....	66
2.8.14. Harici Olarak Zamanlayıcı Ayar Değerinin Atanması.....	68
2.8.15. Zamanlayıcı Uygulamaları.....	68
2.8.16. Sayıcı (Counter) Komutları.....	69
2.8.17. Harici Olarak Sayıcı Ayar Değerinin Atanması.....	72
2.8.18. Sayıcı Uygulamaları.....	72
2.8.19. Karşılaştırma Komutları ve Uygulaması.....	72
2.8.20. Sayı Sistemleri ile Taşıma ( MOVE) Komutları ve Uygulaması.....	75
2.8.21. Matematik İşlem, Fonksiyon Komutları ve Uygulaması.....	80
2.8.22. Mantık İşlemler, Kaydırma, Dönüştürme Komutları ve Uygulaması.....	86
2.8.23. Program Kontrol Komutları ve Uygulaması.....	93
2.9. Yapılan Programa Göre Kullanılan Giriş ve Çıkışların Simülasyonu.....	96
UYGULAMA FAALİYETİ.....	97
PERFORMANS DEĞERLENDİRME.....	98
UYGULAMA FAALİYETİ.....	99
PERFORMANS DEĞERLENDİRME.....	100
ÖLÇME VE DEĞERLENDİRME.....	101
ÖĞRENME FAALİYETİ-3.....	103

3. PROGRAM YEDEKLEME .....	103
3.1. Program Yedeklemenin Önemi.....	103
3.2. PLC’deki Programın Bilgisayara Çekilmesi.....	103
UYGULAMA FAALİYETİ.....	105
PERFORMANS DEĞERLENDİRME .....	106
MODÜL DEĞERLENDİRME.....	107
CEVAP ANAHTARLARI .....	108
ÖNERİLEN KAYNAKLAR .....	110
KAYNAKÇA .....	111

## AÇIKLAMALAR

<b>KOD</b>	<b>523EO0053</b>
<b>ALAN</b>	<b>Elektrik Elektronik Teknolojisi</b>
<b>DAL/MESLEK</b>	<b>Otomasyon Sistemleri</b>
<b>MODÜLÜN ADI</b>	<b>PLC Programlama Teknikleri</b>
<b>MODÜLÜN TANIMI</b>	PLC ile kumanda edilecek sistemin kontrol programının bilgisayar ortamında yazılması ve PLC'ye yüklenmesi için gerekli bilgi ve becerilerin kazandırıldığı öğrenme materyalidir.
<b>SÜRE</b>	40/32
<b>ÖN KOŞUL</b>	PLC'ye Giriş modülünü tamamlamış olmak
<b>YETERLİK</b>	PLC ile kontrol edilecek sisteme göre programı yaparak, PLC'ye yüklemek.
<b>MODÜLÜN AMACI</b>	<p><b>Genel Amaç</b> Gerekli ortam sağlandığında sistemin PLC ile kumanda ve kontrolünü TSE, İç Tesisleri Yönetmeliği ve şartnamelere uygun şekilde kurup çalıştırabileceksiniz.</p> <p><b>Amaçlar</b></p> <ol style="list-style-type: none"><li>1. Hazırlanan yazılımı gerekli bağlantıyı kurup parametre ayarlarını yaparak bilgisayardan PLC cihazına yükleyebileceksiniz.</li><li>2. Kurulacak sistemin kontrolünü sağlayacak PLC programını teknigine uygun olarak yapabileceksiniz.</li><li>3. PLC cihazına yüklenen programı bilgisayar ortamına çekebilecek, yedeklediğiniz programı sistemin yazılımından kaynaklanan arıza durumlarında tekrar yükleyebileceksiniz.</li></ol>
<b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b>	PLC deney seti veya PLC cihazı PLC programı yazılım editörü Bilgisayar Bilgisayar ile PLC arası haberleşme kablosu Program yükleme kartları
<b>ÖLÇME VE DEĞERLENDİRME</b>	Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. Öğretmen, modül sonunda size ölçme aracı (uygulama, soru-cevap) uygulayarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek değerlendirecektir.

# GİRİŞ

**Sevgili Öğrenci,**

Günümüzde kullanılan birçok cihazın kontrolü, eskiden mekanik olarak yapılmakta iken, günümüz teknolojisinde bu işlemler elektromekanik ve hatta elektronik ortamda yapılmaktadır.

Elektromekanik sistemlerin bakım ve işletim maliyetinin yüksek oluşu, bu sistemlerin mekanik kısımlarının terk edilmesine ve elektronik kontrolün cazibesinin artmasına sebep olmuştur. Artık cihazların standart işlem yapar durumdan sürekli yeni ihtiyaçlara cevap verir hâle gelmesi bir zorunluluk halini almıştır.

Cihazlara değişik işleri yaptırmak için farklı kontrol birimleri eklemek yerine tek bir kontrol birimiyle bu işlemlerin hepsini kumanda etmek; hem cihazların boyutu hem de fonksiyonelliği açısından önem kazanmıştır. Bu süreçte sayısal işlem yapabilen mikro işlemcili sistemler ön plana çıkmıştır.

Bu mikro işlemcili sistemlerden endüstriyel otomasyonda en yaygın olarak kullanılanları programlanabilir kontrol cihazlarıdır. Kısaca PLC olarak anılan bu cihazların bugünkü kullanma alanı yaygın olduğu gibi gelecekte de vazgeçilmez olacağı kesindir. PLC cihazlarının kurulması ve programlanması için bu konuda yetişmiş elemana ihtiyaç gittikçe artmaktadır.

Bu modül sonunda piyasa ihtiyaçlarına cevap verebilecek şekilde bir PLC cihazının programını yazabileceksiniz.





# ÖĞRENME FAALİYETİ-1

## AMAÇ

Hazırlanan yazılımı gerekli bağlantıyı kurup parametre ayarlarını yaparak bilgisayardan PLC cihazına yükleyebileceksiniz.

## ARAŞTIRMA

- Ø Bilgisayar ile PLC arasında veri akışının (haberleşme) nasıl sağlandığı hakkında internet ortamından, kütüphanelerden ve PLC kataloglarından araştırarak rapor haline getiriniz ve sınıfta sununuz.

## 1. YAZILIMI SİSTEME YÜKLEMELİK

### 1.1. Bilgisayar Haberleşme Portları

Bilgisayar ile çevre birimleri arasında veri alışverişi portlar aracılığı ile gerçekleştirilir. Bilgisayarlarda, verinin iletim biçimi bakımından, seri ve paralel portlar olmak üzere iki çeşit port bulunur.

#### 1.1.1. Paralel Portlar

Paralel portlar ile veriler paralel bir biçimde, yani bir seferinde bir bayt olarak iletilir. Port sekiz adet veri hattı içerir ve baytın her biti bayttaki diğer bitlerle hemen hemen aynı anda farklı bir hattan iletilir. Paralel portlar LPT1, LPT2 gibi isimlendirilir ve bilgisayarlar en az bir paralel port bulundurur. Paralel portlar şekilde görüldüğü gibi 25 pinlik bir dışi konnektör kullanırlar.



Resim 1.1: Paralel port

#### 1.1.2. Seri Portlar

Seri portlar ile veriler seri bir biçimde yani her seferinde tek bit olarak iletilir. Bunun sebebi portun her yön için tek bir veri hattına sahip olmasıdır. Seri iletişimde bitler peş peşe ve belirli zaman aralıkları ile gönderilir ya da alınır. Alıcı ve vericinin senkronize olması için bir bitlik verinin hatta kalma süresi önemlidir. Bu nedenle bilgisayarın çeşitli çevre cihazları ile seri iletişim hızı birbirinden farklı olabilir. Seri iletişim hızı **bit/sn** ya da **Baud** birimiyle ölçülür. Baud hız, bir saniyede iletilen bit sayısıdır.

Seri portların konektörleri 25 ve 9 pin olmak üzere 2 şekilde olur. 25 pinlik bir aygıtı 9 pinlik bir porta ya da 9 pinlik bir aygıtı 25 pinlik bir aygıta bağlamak gibi durumlarda kullanılabilecek adaptörler vardır. 25 pinlik konektörler artık kullanılmamaktadır.



**Resim 1.2: Seri portlar**

Seri portlar ile paralel portların bir kıyaslaması yapılması gerekirse; seri portlar ile bilgilerin iletilmesi daha güvenli olmasına rağmen iletişim hızı yavaştır. Paralel portlar ise seri porttan çok daha hızlıdır. Bununla birlikte güvenilir bir veri iletimi sağlamazlar. Özellikle kablo uzunluğu arttıkça verilerin kaybolma riski doğar.

Bilgisayarda bulunan PLC kontrol programını, PLC'ye yüklemek için seri portlar kullanılır.

### **1.1.3. USB (Universal Serial BUS )**

USB, bilgisayar ve telekomünikasyon endüstrisinde geliştirilmiş, iletişim standartlarında yeni bir bağlantı şeklidir. Amacı, geleneksel seri ve paralel portların yerini alarak çevre cihazlarla iletişimi evrenselleştirmektir. Bilgisayarlar, artık USB' siz üretilmemektedir. Çevre birimleri de aynı şekilde USB destekli üretilmektedir. Klavye, fare, oyun çubuğu, telefon, tarayıcı, yazıcı, mikrofon, hoparlör, disket sürücü, kamera, modem, CD-ROM sürücü vb. pek çok cihazla bilgisayar arasında USB bağlantı yapılmaktadır.



**Resim 1.3: USB Port**



**Resim 1.4: USB bağlantı kablosu**

## 1.2. Bilgisayar ile PLC Arasındaki Haberleşme Ayarları

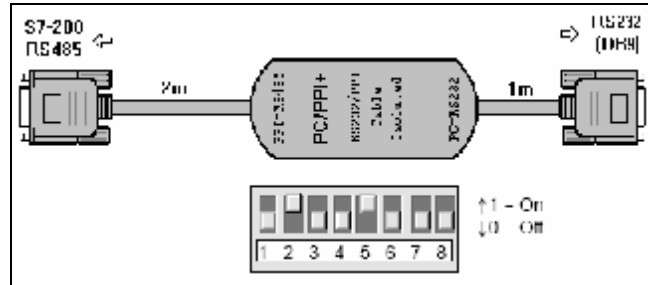
### 1.2.1. Haberleşme Elemanları

Bilgisayar ile S7 – 200 PLC’ler arasında haberleşmeyi sağlamak için Resim 1.5’te görülen PC / PPI master programlama kablosu kullanılır. Kablonun, üzerinde “PC” yazan RS 232 konnektörü bilgisayarın 9 pin seri portuna, üzerinde PPI yazan RS 485 konnektörü S7 – 200 PLC’nin haberleşme portuna bağlanır.



Resim 1.5: PC/PPI programlama kablosu

S7 – 200 PLC verileri 9600 bit/sn hızla gönderir ve alır. Bu hızı sağlamak için PC / PPI kablosu üzerindeki sviçlerin Şekil 1.1’deki konumda olmaları gerekmektedir.



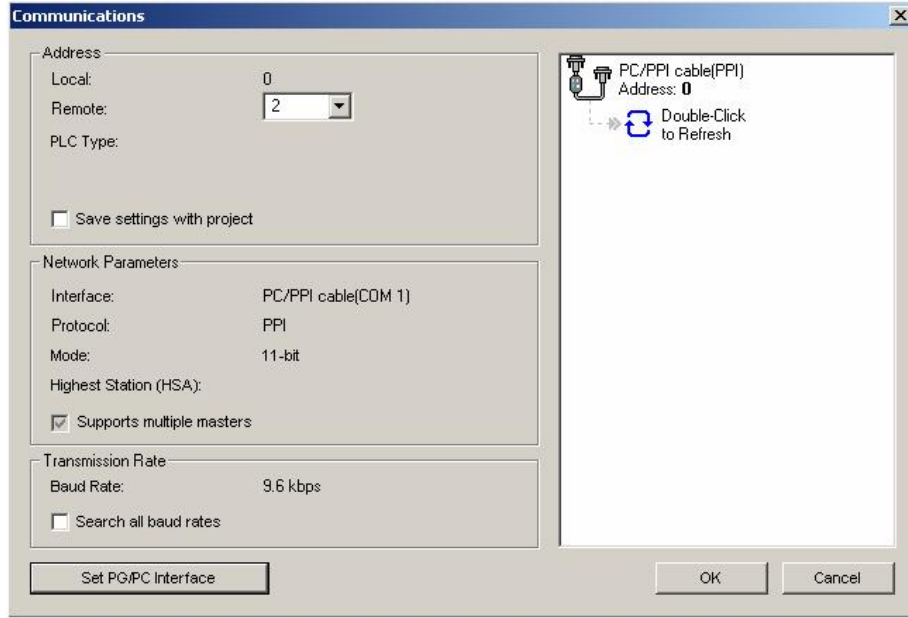
Şekil 1.1: PC/ PPI kablosu sviç konumları

### 1.2.2. PLC Ayarları

Bilgisayar ile PLC iletişim arasında iletişim kurmak için Communication iletişim kutusundan iletişim ayarlarını kontrol etmemiz gerekir. Bunun için View >component> Communication menü seçeneklerini kullanarak ya da araştırma çubuğundan Communication ikonuna tıklayarak Şekil 1.2’deki iletişim kutusunun açılmasını sağlamalıyız.

Communication iletişim kutusunda aşağıdaki ayarları doğrulamanız gerekmektedir.

- Ø PC / PPI kablo başlangıç adres değerinin 0 olduğunu doğrulayın
- Ø İletişim hızının 9.6 kbps olduğunu doğrulayın
- Ø Şebeke parametresinin COM1'e ayarlı olduğunu doğrulayın

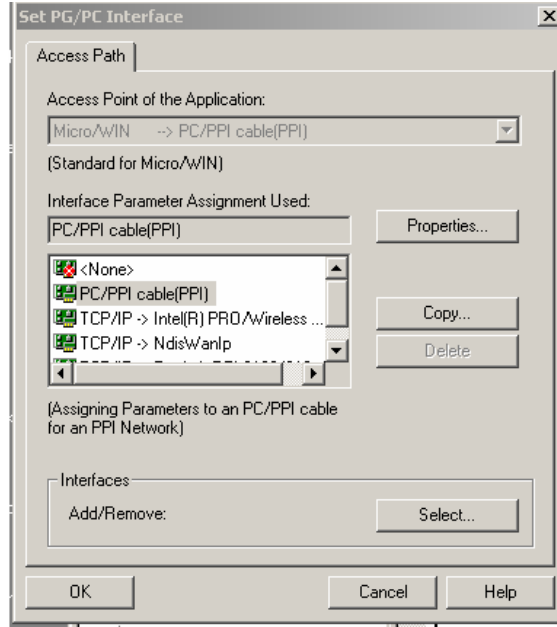


**Şekil 1.2: Communications iletişim kutusu**

Daha sonra Double - Click to refresh ikonuna çift tıklandığında Step – 7 Micro/ Win programının S7 -200 PLC cihazını araştırarak bulmasını sağlamanız gerekmektedir.

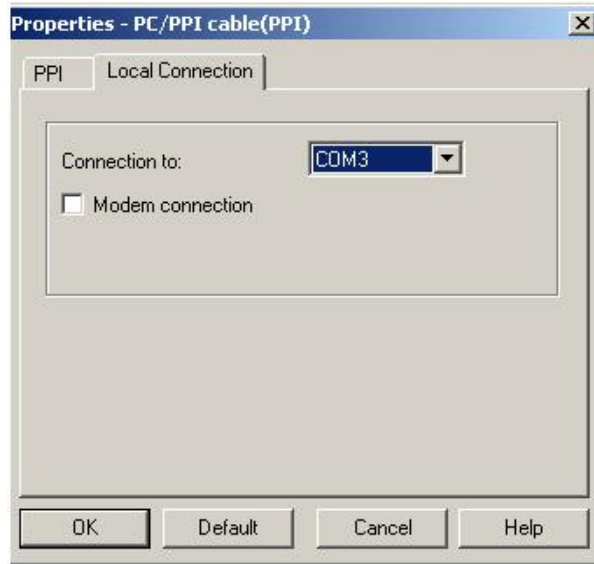
### **1.2.3. Com ve Port Ayarları**

Port ayarlarının yapılması için View>Component> Set PC/PPI interface menü seçeneklerini kullanarak ya da araştırma çubuğundan Set PC / PPI interface ikonuna tıklayarak Şekil 1.3'teki Set PC / PPI interface iletişim kutusunun ekrana gelmesini sağlayınız.



**Şekil 1.3: Set PG / PC Interface iletişim kutusu**

İletişim kutusundan PC / PPI cable seçeneğini seçtikten sonra properties düğmesine tıklayıp açılan Şekil 1.4'teki Properties iletişim kutusundan Local connection sekmesine tıklayarak COM port seçimi yapınız.

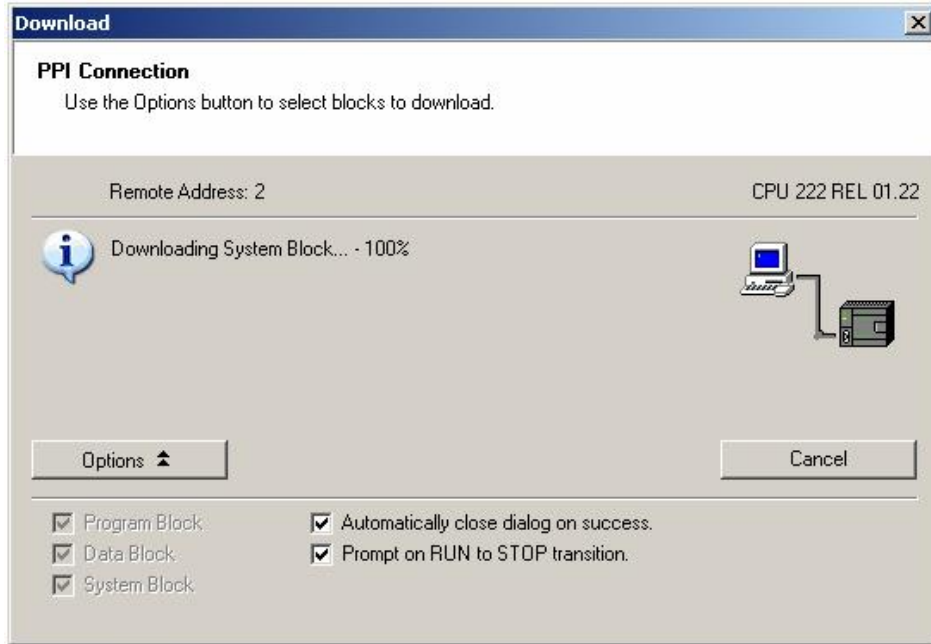


**Şekil 1.4: Properties iletişim kutusu**

### 1.3. Kontrol Programının Plc'ye Yklenmesi

Hazırlanmıř olan PLC kontrol programının PLC program hafızasına yklenmesi iin File mensnden Download seeneđine ya da standart ara ubuđundaki Download ikonuna tıklanarak řekil 1.5'teki Download iletiřim kutusunun ekrana gelmesi sađlanır.

PLC programı, program blođuna yazılan komutlardan, data blođuna V hafızası iin yazılmıř bařlangı deđerlerinden ve system blođunda yapılan ayarlardan meydana gelmektedir. Eđer data blođu hazırlamıř ve sistem blođunda ayarlamalar yapmıřsanız programın dođru alıřması iin bu bloklardaki bilgilerin PLC'ye yklenmesi gerekir. Options dđmesine tıklayarak ykleme istediđiniz blokları onayladıktan sonra download dđmesine tıklayarak programın PLC hafızasına yklenmesini sađlamıř oluruz.



řekil 1.5: Download iletiřim kutusu Set PG / PC Interface iletiřim kutusu

## UYGULAMA FAALİYETİ

Atölyede bulunan PLC cihazını bilgisayarınıza bağlayarak haberleşmesini sağlayarak daha önce hazırlanmış bir programı PLC cihazına yükleyiniz.

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"><li>Ø PPI kablosu sviç konumlarını ayarlayınız.</li><li>Ø Bilgisayar ile PLC cihazı arasındaki bağlantıyı PPI kablosu ile uygun porttan yapınız.</li><li>Ø PLC programlama editöründen gerekli ayarları yaparak haberleşme portunu seçiniz.</li><li>Ø PLC editöründe yazılan programın gerekli yönergeleri izleyerek PLC'ye aktarılmasını sağlayınız.</li></ul>	<ul style="list-style-type: none"><li>Ø Atölyenizde bulunan PLC cihazını öğretmeninize danışarak alınız.</li><li>Ø Bilgisayar ile PLC arasında haberleşme irtibatını sağlamak için gerekli kabloyu depodan alınız.</li><li>Ø Kablonun RS 232 konnektörünü bilgisayarınızın seri portuna takılmasına dikkat ediniz.</li><li>Ø Kablonun RS 485 konnektörünün PLC portuna takılmasına dikkat ediniz.</li><li>Ø Programın yüklenebilmesi için PLC'nin stop konumunda olmasına dikkat ediniz.</li><li>Ø Yaptığınız işlemleri rapor haline getirerek öğretmeninize sununuz.</li></ul>

## PERFORMANS DEĞERLENDİRME

DEĞERLENDİRME ÖLÇÜTLERİ		Evet	Hayır
1	Sviç konumlarını doğru olarak ayarlayabildiniz mi?		
2	PLC'yi PPI kablosu ile bilgisayara bağladınız mı?		
3	PLC programlama editöründen haberleşme için gerekli port seçimini yaptınız mı?		
4	Bilgisayarda kayıtlı bulunan herhangi bir kontrol programını PLC editöründe açabildiniz mi?		
5	Açtığınız programı PLC cihazına gönderebildiniz mi?		



# ÖĞRENME FAALİYETİ-2

## AMAÇ

Kurulacak sistemin kontrolünü sağlayacak PLC programını teknigine uygun olarak yapabileceksiniz.

## ARAŞTIRMA

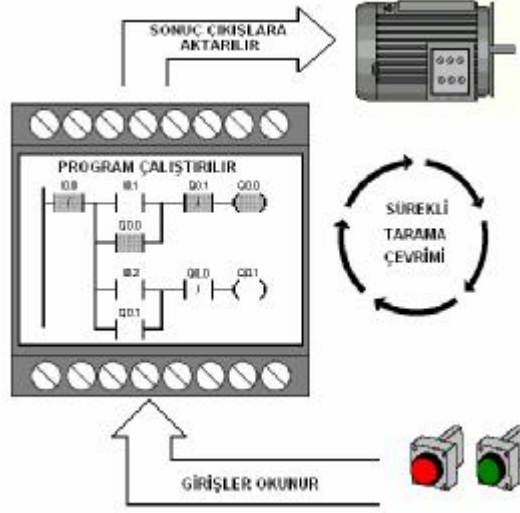
- Ø PLC programlama mantığı ve kontrol programı türleri hakkında internet ortamı ve kütüphanelerden araştırma yaparak yaptıklarınızı rapor haline getirerek sınıfta sununuz.
- Ø PLC program yazılım dilleri hakkında internet ortamı ve kütüphanelerden araştırma yaparak yaptıklarınızı rapor haline getirerek sınıfta sununuz.
- Ø PLC programlama komutlarını internet ortamı ve kütüphanelerden araştırma yaparak yaptıklarınızı rapor haline getirerek öğretmeninize sununuz.

## 2. PLC İÇİN KONTROL PROGRAMININ YAZILMASI

### 2.1. PLC’de Programlama Mantığı ve Kontrol Programı Türleri

#### 2.1.1. PLC’de Program İşleme Mantığı

PLC çalışma (RUN) moduna alındığında ilk önce çıkış hafızası sıfırlanır. Sonra girişlere bağlı bulunan sensör, buton, sınır anahtarı vb. elemanlardan alınan sinyaller okunarak giriş hafızasına kaydedilir ve PLC’nin program hafızasına yüklenen kontrol programı komutları sırası ile işletilir. Giriş değişkenlerine bağlı olarak elde edilen sonuçlar çıkış hafızasına aktarılır. Çıkış hafızasındaki bilgiler PLC çıkışlarına bağlanan iş elemanlarını çalıştıracak şekilde çıkışlara aktarılır ve tekrar girişler okunur. Tüm bu işlemlerin gerçekleşmesi için geçen zamana bir tarama çevrimi adı verilir. PLC’nin enerjisi kesilinceye kadar ya da STOP konumuna alınıncaya kadar tarama çevrimi sürekli olarak tekrar eder. Bir tarama çevriminin süresi PLC çalışma hızına, kullanılan komutların niteliğine ve kontrol programının uzunluğuna göre değişir. Tipik olarak bir tarama çevrimi süresi 3ms ile 10 ms arasındadır. Eğer bu süre çok uzun olursa girişlerde meydana gelen çok kısa süreli sinyal değişiklikleri algılanamayabilir.



Şekil 2.1: PLC Tarama çevrimi

### 2.1.2. Lineer (Doğrusal) Programlama Mantığı

Doğrusal programlama, bütün komutların aynı program alanına yazıldığı bir programlama biçimidir. Komut yazılış sırasına göre yürütülür ve bir çevrim boyunca bütün komutlar işleme girer. Bu programlama biçiminde program ana program ve alt program biçiminde düzenlenir. Alt programlar ya ana programın program sonu komutundan (END, MEND gibi ) sonra ya da özel bir alana yazılır.

Doğrusal programlamada, bir alt programlamadan başka bir alt programın çağrıldığı yapılar kullanılabilir. Ancak bu programlama kumanda sisteminin tasarımını ve izlenmesini zorlaştırdığı için genellikle tercih edilmezler.

Ana programlamada yazılan komutlar genellikle alt programlarda kullanılabilir. Siemens S7- 200 PLC sınıfının programlanması için geliştirilen “STEP 7-Micro/WIN V3.0” yeni versiyon yazılımında alt programlar, bu alt programlar için ayrılmış alanlara yazılmakta ve bu nedenle, daha önceki versiyonlarda kullanılan ana program sonu komutu ( MEND ) ve alt program komutu ( RET ), kesme alt program komutu ( RETI ) kullanılmamaktadır. Yine bu yazılımda S7–200 sınıfı yeni nesil işlemciler ( CPU 221, CPU 222, CPU 224 ve CPU 226 ) için alt program yapısal kullanıma uygun biçimde çalışmasını sağlayacak özellikler artırılmıştır.

### 2.1.3. Yapısal Programlama Mantığı

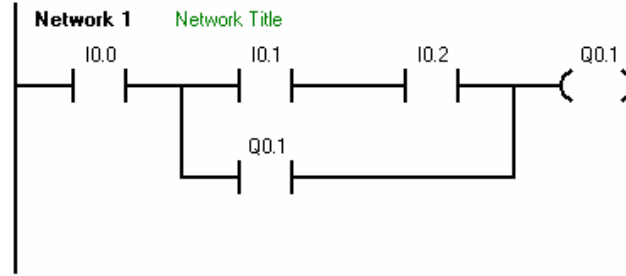
Yapısal programlama, büyük ölçekli programların işlevine göre parçalanarak ve aynı işlevi sağlayan işler için yalnız bir program parçasının kullanıldığı bir programlama biçimidir.

Siemens tarafından üretilen S5 sınıfı PLC'leri programlamak için kullanılan STEP 5 ve yeni nesil S7 – 300/400 sınıfı PLC'leri programlamak için STEP 7 yazılımları hem doğrusal hem de yapısal programlamaya uygun dillerdir. Örneğin STEP 5 dili verilen program organizasyon, program ve fonksiyon blokları çeşitli işlevleri sağlayan program parçalarını içerir. Bütün program blokları birer alt program gibi düşünülebilir. Kesme alt programları için de yine tanımlanmış özel organizasyon blokları kullanılır. Sistem programı organizasyon bloklarını işletir. Organizasyon bloğuna yazılan atlama komutları ile bir program çevriminde hangi blokların yürütüleceği belirlenir.

## 2.2. PLC Program Yazılım Dilleri

### 2.2.1. Kontak Plan (LADDER Plan)

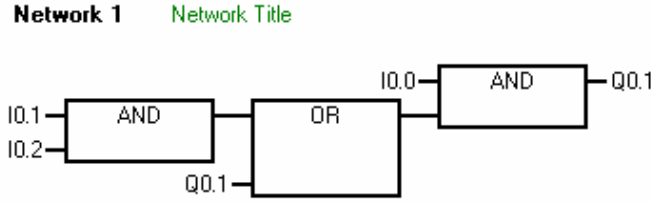
Ladder plan, röle ve kontaktörlerle yapılan klasik kumanda devrelerinin çizimlerine benzeyen grafiksel bir programlama şeklidir. Ladder plan gerçek elektrik devrelerinde olduğu gibi bir enerji kaynağından kontaklar aracılığıyla akan enerjiyi sembolize etmek şeklinde kullanıcıya kolay gelebilecek bir programlama mantığına sahiptir. Ladder programında sol tarafta gösterilen dikey çizgi enerji kaynağını gösterir. Kapalı kontaklar enerji akışına izin verirken açık kontaklar enerji akışına izin vermezler. Ladder plan yöntemi daha çok elektrik eğitimi almış kişiler ve yeni başlayanlar için uygundur. Şekil 2.2'de FBD yöntemiyle yazılmış program örneği görülmektedir.



Şekil 2.2: Kontak plan(LADDER) yöntemle program örneği

### 2.2.2. Fonksiyon Plan (FBD)

FBD yöntemi, lojik kapıların kullanımına dayanan ve şematik bir gösterim şekli sunan programlama şeklidir. Burada kullanılan lojik semboller kutular şeklinde gösterilir. Sembollerin sol tarafında giriş sinyalleri, sağ tarafında ise çıkış sinyalleri bulunur. Bu yöntem dijital elektronik eğitimi almış kişilerce daha rahat kullanılabilir. Şekil 2.3'te FBD yöntemiyle yazılmış program örneği görülmektedir.



**Şekil 2.3: Fonksiyon plan ( FBD) yöntemiyle program örneği**

### 2.2.3. Deyim Listesi (STL)

STL yönteminde PLC'nin türüne ve markasına göre aynı işlevi gören fakat yazılım şeklinde küçük farklılıklar olan komutlar kullanılır. Bir komut yapılan işlemi belirten Mnemonic ve üzerinde işlem yapılan hafıza alanlarını gösteren operantlardan oluşur. Bu yöntem cihazın, makina koduna en yakın gösterim şekli olduğundan çok geniş programlama imkanları sunar. STL yöntemi bilgisayar teknolojisine yatkın kişilere hitap eder.

STL, FBD ve LADDER yöntemiyle yazılan programlar hatasız yazılmış ve derlenmiş olmak şartıyla birbirinin stillerine dönüştürülebilir. Şekil 2.4'te STL yöntemiyle yazılmış program örneği görülmektedir.

Network 1	Network Title
LD	I0.0
LD	I0.1
A	I0.2
O	Q0.1
ALD	
=	Q0.1

**Şekil 2.4: Deyim Listesi ( STL) yöntemiyle program örneği**

## 2.3. PLC Programlama Yazılımının Bilgisayara Kurulması

Kurulacak sistemin PLC programını yazmak için bir PLC programlama editörüne ihtiyaç vardır. Piyasada bulunan her model PLC için kendine özgü programlama yazılımı mevcuttur ve satın alınan cihaz ile birlikte verilmektedir. Bu modül de S7-200 PLC'leri programlamak için kullanılan Step7 - Micro /Win32 V programı kullanılacaktır.

Step7-Micro/Win 32 programının çalışabilmesi için bilgisayarınızın en az aşağıdaki özelliklere sahip olması önerilir.

- Ø Windows 95 / NT4.0 ve daha üstü sürümlere sahip işletim sistemi
- Ø 50 MB Hard disk alanı
- Ø 16 MB Ram

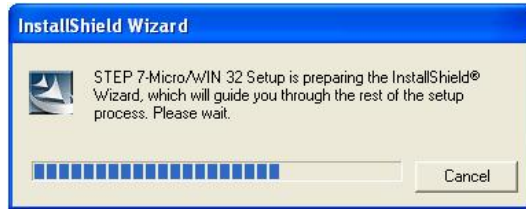
Step7-Micro/Win 32 V3.2 programını bilgisayarınıza kurmak için aşağıdaki adımları uygulamanız gerekmektedir.

- Ø **ADIM 1:** Step7 Micro/Win 32 V3.2 kurulum CD'sini CD sürücüsüne yerleştiriniz. Kurulum programı otomatik olarak çalışmaya başlayacak ve Şekil 2.5'te görülen dil seçimi iletişim kutusu ekrana gelecektir. Buradan seçilecek dil programın yüklenmesi sırasında bize rehberlik edecek kurulum dilidir. Türkçe kurulum seçeneği bulunmamaktadır. Açılır liste kutusundan dil seçimi yaptıktan sonra Tamam düğmesine tıklayınız.



Şekil 2.5: Step7-Micro/ Win kurulum adım1

- Ø **ADIM 2:** Bu adımda Şekil 2.6 görüldüğü gibi Step7 Micro/Win 32 kurulum sihirbazı otomatik olarak yüklenecektir.



Şekil 2.6: Step7-Micro/ Win kurulum adım2

- Ø **ADIM 3:** Bu adımda Next düğmesine tıklayarak kurulumla devam etmeniz gerekmektedir.



Şekil 2.7: Step7-Micro/ Win kurulum adım3

- Ø **ADIM 4:** Şekil 2.8’de görülen bu adımda lisans anlaşmanızın bulunup bulunmadığı sorgulanmaktadır. Eğer lisanslı bir program kullanıyorsanız Yes düğmesine tıklayarak kurulumla devam edebilirsiniz. Kullandığınız kurulum CD’si lisanssız ise programı kurmanız durumunda cezai sorumluluğu kabul etmiş olursunuz.



Şekil 2.8: Step7-Micro/ Win kurulum adım4

- Ø **ADIM 5:** Şekil 2.9’da görülen iletişim kutusunda User Name isimli metin kutusuna bir kullanıcı adı ve Company Name isimli metin kutusuna ise Firma / Şirket ya da okul adı yazıldıktan sonra Next düğmesi aktif hâle gelir. Next düğmesine tıklayarak diğer aşamaya geçebilirsiniz.



Şekil 2.9: Step7-Micro/ Win kurulum adım5

- Ø **ADIM 6:** Bu adımda Step7 Micro/Win 32 programının kurulacağı dizinin seçimi yapılmalıdır. Şekil 2.10'da görülen iletişim kutusunun Destination Folder bölümünde Programın, Program Files klasörü içine açılacak Siemens isimli alt klasör içine kurulacağı varsayılmaktadır. İsterseniz önerilen dizini kabul edebilir ya da Browse düğmesine tıklayarak kendinizin belirleyeceği bir klasör seçtikten sonra Next düğmesine tıklamanız gerekmektedir.



Şekil 2.10: Step7-Micro/ Win kurulum adım 6

- Ø **ADIM 7:** Bu adımda diller seçilerek programı çalıştıracak dosyalar için alt yapı oluşturulur. Birden fazla dil seçersek program kurulduktan sonra programda kullanılan dili Options iletişim kutusundan dilediğimiz zaman değiştirebiliriz. Böylece kurduğumuz program farklı dilleri bilen kişilerce de kullanılabilir. Program tarafından desteklenen tüm dilleri yüklemek için Şekil 2.11'de görülen iletişim kutusundaki **All Languages** seçeneğini seçtikten sonra Next düğmesine tıklayınız.



Şekil 2.11: Step7-Micro/ Win kurulum adım7

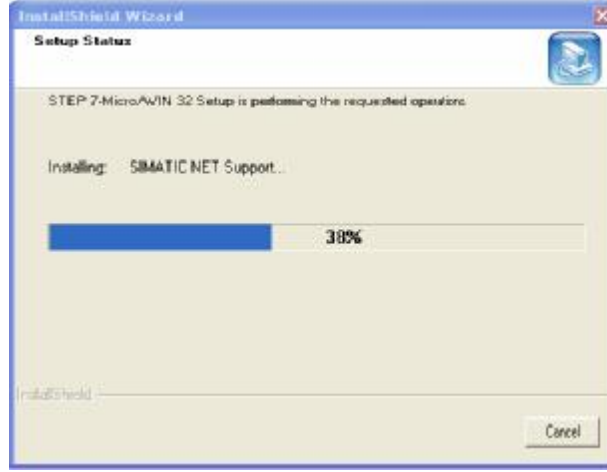
Eğer sadece sizin tercih edeceğiniz dillerde programı yüklemek istiyorsanız. Select Languages seçeneğini seçip Next düğmesine tıklamanız gerekmektedir. Bu durumda Şekil 2.12'deki iletişim kutusu ekrana gelecektir. Buradan istenilen dillerin önündeki onay kutularını işaretleyerek dil seçimi yaptıktan sonra Next düğmesine tıklayınız.



Şekil 2.12: Step7-Micro/ Win kurulum adım 7

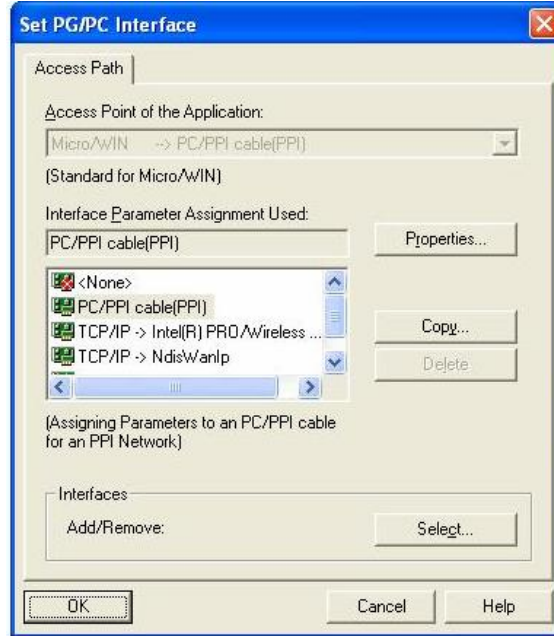
- Ø **ADIM 8:** Bu adımda programın ve bileşenlerinin kurulması işlemi otomatik olarak yapılır. Yükleme oranı %100 olduğunda kurulum tamamlanır ve bir sonraki adıma geçilir.





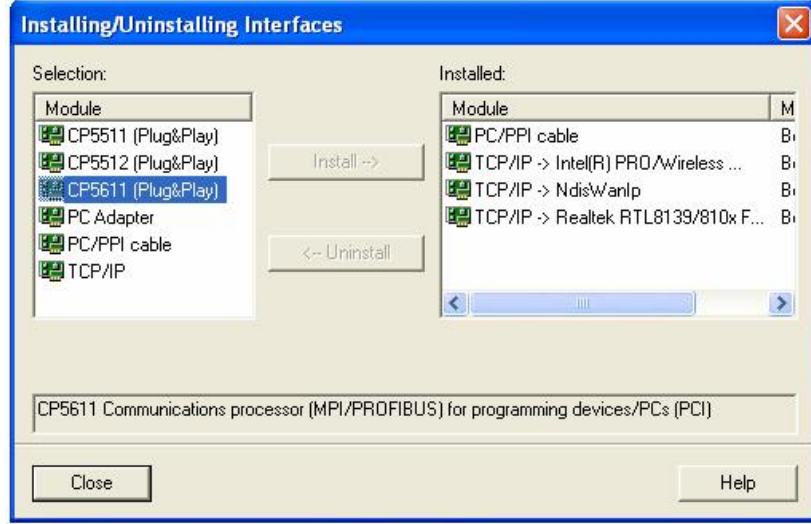
Şekil 2.13: Step7-Micro/ Win kurulum adım 8

- Ø **ADIM 9:** Bu adımda Şekil 2.14'te görülen iletişim kutusundan PLC ile bilgisayar arasında haberleşmeyi sağlayacak haberleşme devre elemanlarının seçimi yapılır. Temel uygulamalar için PC / PPI CABLE (bir noktadan diğer noktaya haberleşme kablosu) elemanının seçilmesi yeterli olacaktır. Sadece bu elemanı seçmek için üzerine fare sol tuşu ile tıkladıktan sonra Ok düğmesine tıklayınız.



Şekil 2.14: Step7-Micro/ Win kurulum adım9

Eğer diğer haberleşme devre elemanlarını da yüklemek istiyorsak Select... düğmesine tıklayarak Şekil 2.15'te görülen iletişim kutusunun ekrana gelmesi sağlanır.



Şekil 2.15: Step7-Micro/ Win kurulum adım7

- Ø **ADIM 10:** Buraya kadar olan adımların uygulanmasıyla programın bilgisayara yüklenmesi işlemi tamamlanmıştır. Son olarak bilgisayarınızın kapatılarak yeniden başlatılması gerekmektedir. Bu adımda Şekil 2.16'daki iletişim kutusundan “Bilgisayarımı yeniden başlatmak istiyorum” ya da daha sonra başlatma seçeneklerinden birini seçerek Finish düğmesine tıklayınız.



Şekil 2.16: Step7-Micro/ Win kurulum adım7

## 2.4. Program Menüleri ve Açıklanması

PLC programlama yazılımının menülerinde bulunan seçenekler hakkında bilgi sahibi olunması progracıya kolaylık sağlayacaktır. Step7-Micro/Win 32 V3.2 programında “File”, “Edit”, “View”, “PLC”, “Debug”, “Tools”, “Windows” ve “Help” menüleri bulunmaktadır.

Bu menülerde kullanıcının sıklıkla başvurduğu seçenekler programın araç çubuklarına kısayol ikonu olarak yerleştirilmiştir.

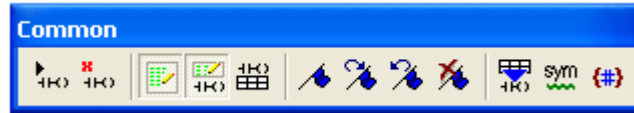
### 2.4.1. Araç Çubukları

- Ø **Standart Araç Çubuğu:** Şekilde görülen standart araç çubuğunda genel olarak menülerde sıklıkla başvuru alan seçenekler için oluşturulmuş kısayol ikonları bulunur.



Şekil 2.17: Standart araç çubuğu

- Ø **Common Araç Çubuğu:** Şekilde görülen Common araç çubuğunda Edit ve View menülerinde sıklıkla başvuru alan seçenekler için oluşturulmuş kısayol ikonları bulunur.



Şekil 2.18: Common araç çubuğu

- Ø **Debug Araç Çubuğu:** Şekilde görülen Debug araç çubuğunda PLC ve Debug menülerinde sıklıkla başvuru alan seçenekler için oluşturulmuş kısayol ikonları bulunur.



Şekil 2.19: Debug araç çubuğu

- Ø **Instruction Araç Çubuğu:** Bu araç çubuğunda Ladder ve FBD programlama editörlerinde çalışırken komutları temsil eden sembollere kolay ulaşım için gerekli kısayol ikonları bulunur

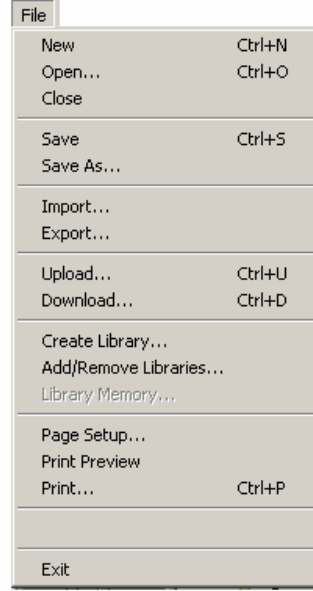


Şekil 2.20: Instruction araç çubuğu

### 2.4.2. File (Dosya) Menüsü

File menüsü üzerine tıklandığında Şekil 2.21'de görülen File menüsü seçenekleri ekrana gelir.

- Ø **New:** PLC cihazını programlayacak yazılımı yapmak için yeni bir sayfa açar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Open:** Daha önce hazırlanmış olan ve bilgisayara kaydedilmiş program dosyalarını açar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Close:** PLC'ye yüklenmek için hazırlanan aktif program penceresinin kapatılmasını sağlar. Bu seçenek tıklandığında, programda yapılan değişiklikler kaydedilmemiş ise "Kaydedilsin mi?" sorusu ekrana gelir.



**Şekil 2.21: File menüsü**

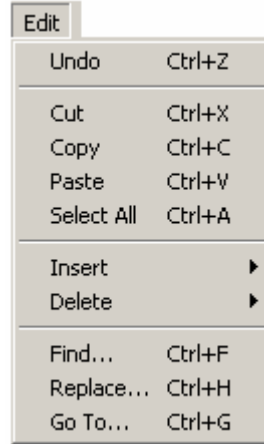
- Ø **Save:** PLC'nin programlanması için hazırlanmış uygulamanın bir dosya halinde bilgisayarakaydedilmesini sağlar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Save As:** PLC'nin programlanması için hazırlanmış uygulamada yapılan değişikliklerin farklı isimde dosyalara kaydedilmesini sağlar.
- Ø **Import:** Eğer PLC için hazırlanmış olan uygulama bilgisayara ASCII yazı karakteri kullanılarak kaydedilmiş ise dosyasının yüklenebilmesi için bu seçenek kullanılır. ASCII yazı karakteri kullanılarak kaydedilmiş program dosyası Open seçeneği kullanılarak açılmaz.
- Ø **Export:** PLC'nin programlanması için hazırlanmış yazılımın bilgisayara kaydedilmesi sırasında ASCII yazı karakteri kullanılarak belirtilen dosya içerisine yazılmasını sağlar. Bu şekilde kaydedilen dosyalar Import seçeneği ile açılabilir.

- Ø **Upload:** PLC proram hafızasına yüklenmiş olan yazılımı program editörüne çekmek için kullanılır. Eğer yazılım PLC hafızasına yüklenirken şifre verilmiş ise Upload seçeneğine tıklandığında şifre sorgulaması yapılacaktır. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Download:** Hazırlanmış olan programın PLC program hafızasına yüklenmesini sağlar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Page Setup:** Hazırlanan uygulamanın yazıcıdan alınması sırasında kullanılacak kağıt ölçülerinin girilmesini sağlar.
- Ø **Print Preview:** Ön izleme olarak adlandırılır ve yazıcıya gönderilecek sayfaların incelenmesini sağlar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Print:** Hazırlanan programın yazıcıdan yazdırılmasını sağlar. Yazıcıya gönderilecek programın ekranda açık olması zorunludur. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Exit:** Programın kapatılarak windows ortamına dönülmesini sağlar. Eğer programın son hali kaydedilmemişse bu seçeneğe tıklandığında “Değişiklikler kaydedilsin mi?” soru iletişim kutusu ekrana gelecektir.

#### 2.4.3. Edit Menüsü

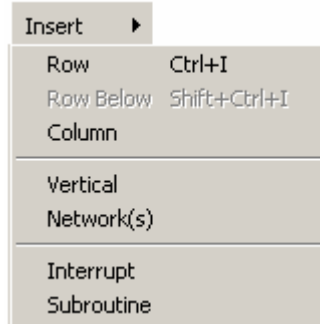
Edit menüsü üzerine tıklandığında Şekil 2.22’de görülen Edit menüsü seçenekleri ekrana gelir.

- Ø **Undo:** Programın yazımı sırasında hata yapılan durumlarda bir önceki adım veya adımlara geri dönüş için kullanılır. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Cut:** Network içerisinde bulunan seçili bir komutun veya seçilmiş bir networkün tamamının silinerek başka bir alana taşınmak üzere panoya alınmasını sağlar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Copy:** Network içerisinde bulunan seçili bir komutun veya seçilmiş bir networkün tamamının bir alana kopyalanmak üzere panoya alınmasını sağlar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Paste:** Cut ya da copy seçeneği ile panoya aktarılan bir komut ya da networkün istenilen alana yerleştirilmesini sağlar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Select All:** Program networklerinin tamamının seçilmesini sağlar.



Şekil 2.22: Edit menüsü

- Ø **Insert:** PLC programı hazırlanırken unutulmuş bir komut veya network için gerekli boş alanı oluşturmak için kullanılır. Satır (Row), sütun (Column) veya paralel kol (Vertical) oluşturarak yeni komutlar yazabilme ile Kesme programı (Interrupt) ve Alt program (Subroutine) sayfası açma imkanı sağlar.



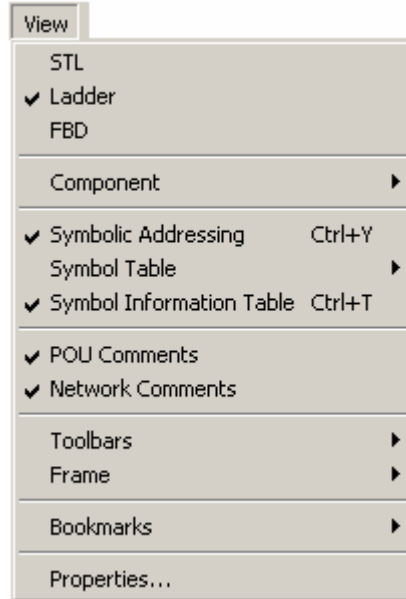
Şekil 2.23: Insert alt menüsü

- Ø **Delete:** Seçilen bir satır, sütun, network, Interrupt veya Subroutine sayfasının silinmesini sağlar.
- Ø **Find:** Hazırlanan programda herhangi bir giriş, çıkış veya dahili hafıza adresi ile kelime işlemci adreslerin programın hangi networklerinde kullanıldığının bulunmasını sağlar.
- Ø **Replace:** Hatalı yazılmış bit adreslerinin veya kelime işlemci adreslerin isimlerinin tümünden değiştirilmesine imkan sağlar.
- Ø **Go To:** Girilen bir adresin, kullanıldığı tüm program networklerindeki satırlarına giderek adreslerin doğruluğunu test etme imkanı sağlar.

#### 2.4.4. View Menüsü

View menüsü üzerine tıklandığında Şekil 2.24'te görülen View menüsü seçenekleri ekrana gelir.

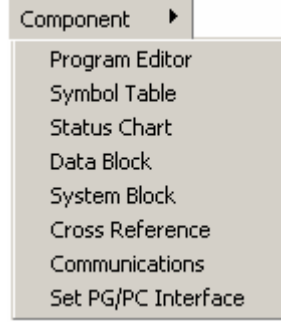
- Ø **STL:** PLC cihazını programlamak için hazırlanacak uygulamada, programlama mantığı olarak STL yöntemi kullanılmak isteniyorsa bu mantığı destekleyen STL program editörü penceresinin açılmasını sağlar.
- Ø **Ladder:** PLC cihazını programlamak için hazırlanacak uygulamada, programlama mantığı olarak Ladder Plan yöntemi kullanılmak isteniyorsa bu mantığı destekleyen Ladder Plan program editörü penceresinin açılmasını sağlar.
- Ø **FBD:** PLC cihazını programlamak için hazırlanacak uygulamada, programlama mantığı olarak FBD yöntemi kullanılmak isteniyorsa bu mantığı destekleyen FBD program editör penceresinin açılmasını sağlar.



Şekil 2.24: View menüsü

- Ø **Component:** Fare ile bu seçenek işaret edildiğinde Şekil 2.25'te görülen ek menü açılır. Bu ek menüde Bir PLC programının hazırlanmasında kullanılan pencerelerin isimleri bulunur. Bu bileşenlerin üzerine tıklayarak tasarım alanında görüntülenmesini sağlayabiliriz. Bu pencereler hakkında daha ayrıntılı bilgiyi bir sonraki bölümde bulabilirsiniz.
- Ø **Program Editör:** Program editör penceresinin ekranda görüntülenmesini sağlar. Wiew araştırma çubuğunda kısayol ikonu bulunmaktadır.

- Ø **Symbol Table:** Symbol tablosu penceresinin ekranda görüntülenmesini sağlar. Wiew araştırma çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Status Chart:** Durum tablosu penceresinin ekranda görüntülenmesini sağlar. Wiew araştırma çubuğunda kısayol ikonu bulunmaktadır.

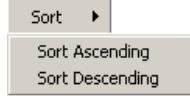


**Şekil 2.25: View menüsü**

- Ø **Data Block:** Data blok penceresinin ekranda görüntülenmesini sağlar. Wiew araştırma çubuğunda kısayol ikonu bulunmaktadır.
- Ø **System Block:** Sistem blok penceresinin ekranda görüntülenmesini sağlar. Wiew araştırma çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Cross Reference:** Çapraz referans tablosu penceresinin ekranda görüntülenmesini sağlar. Wiew araştırma çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Communication:** PLC ile bilgisayar arasındaki iletişim ayarlarının yapılmasını sağlayan iletişim penceresinin ekranda görüntülenmesini sağlar. Wiew araştırma çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Set PG / PC Interface:** PLC ile bilgisayar arasındaki iletişimi sağlayacak haberleşme elemanlarının yüklenmesi ve ayarlarının yapılmasını sağlayan iletişim penceresinin ekranda görüntülenmesini sağlar. Wiew araştırma çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Symbolic Adressing:** Programın yazılması sırasında kolaylık sağlaması ve projenin daha anlaşılır olması amacıyla kullanılan ve sembol tablosunda gösterilen sembolik adres isimlerinin programlama penceresinde görüntülenmesini ya da görünmemesini sağlar. Bu seçenek onaylı ise programlama alanında sembolik adres isimleri görünür. Eğer onaylı değilse mutlak adresler görülür.
- Ø **Symbol Information table:** Sembol tablosu hazırlanmış programlarda, programda kullanılan her networkün sonunda o network içinde kullanılan sembolik adreslerin bir sembol tablosu içinde görüntülenmesini sağlar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **POU Comment:** Program editörünün en başında bulunan ve programcı tarafından, hazırlanan programın kullanım amacı ile ilgili açıklayıcı bilgilerin yazılabildiği alanın görüntülenmesi ya da gizlenmesi için kullanılır. Standart araç çubuğunda kısayol ikonu bulunmaktadır.

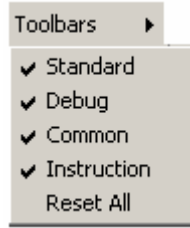


- Ø **Network Comments:** Her networkün başında bulunan ve programcı tarafından, o networkte program tarafından yapılan işlerle ilgili açıklayıcı bilgilerin yazılabildiği alanların görüntülenmesi ya da gizlenmesi için kullanılır. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Sort:** Bu seçenek wiew menüsünde, Sembol tablosu (Symbol table)ya da Durum tablosu ( Status chart ) penceresi aktifken görülür. Kullanım amacı ise belirtilen tablolardaki sembolik adres isimleri alfabetik olarak sıralamaktır.



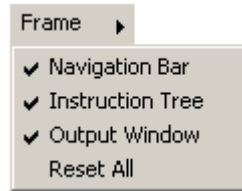
Şekil 2.26: Sort alt menüsü

- Ø **Sort Ascending:** Tablolardaki sembolik adres isimlerini A – Z ye sıralar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Sort Descending:** Tablolardaki sembolik adres isimlerini Z – A ya sıralar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Toolbar:** Menülerin hemen altında yer alan araç çubuklarının görüntülenmesini ya da gizlenmesini sağlamak için seçeneklerin bulunduğu alt menüdür. Toolbar alt menüsünün önünde onay işareti olan araç çubukları ekranda görüntülenir. **Reset All** seçeneğine tıklanırsa bütün araç çubukları ekranda görüntülenir.



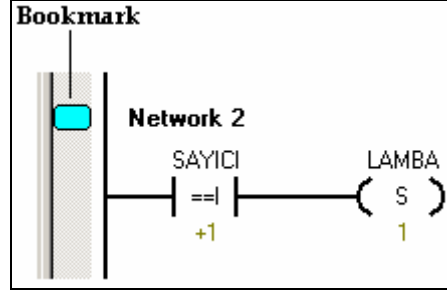
Şekil 2.27: Toolbar alt menüsü

- Ø **Frame:** Program ekranının solunda bulunan araştırma çubuğunun (Nevigation Bar) ortasında bulunan Komut listesi ağacının (Insruction tree) ve alt bölümünde bulunan durum penceresinin görüntülenmesi ya da gizlenmesi sağlayan seçeneklerin bulunduğu alt menüdür.

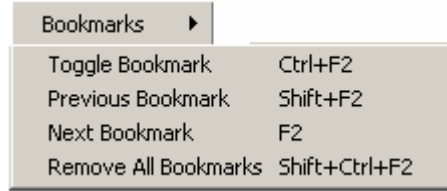


Şekil 2.28: Frame alt menüsü

- Ø **Bookmarks:** Çok uzun programlarda program satırları arasında kolayca gezinmek için program alanına Şekil 2.29'da görüldüğü gibi bookmark adı verilen işaretler yerleştirilebilir. Bookmarklar üzerinde bulunan network isminin yanına yerleştirilir. Bookmarklar arasında geçiş yapmak için bookmark alt menüsündeki seçenekler kullanılır.



Şekil 2.29: Bookmark



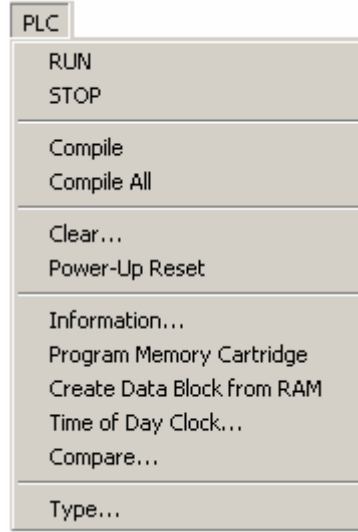
Şekil 2.30: Bookmark alt menüsü

- Ø **Tooggle Bookmark:** Üzerinde bulunan networkün bulunduğu noktaya bir bookmark işareti konulmasını sağlar. Common araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Previous Bookmark:** Bulunulan noktadan bir önceki bookmarkın bulunduğu networke gitmek için kullanılır. Common araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Next Bookmark:** Bulunulan noktadan bir sonraki bookmark'ın bulunduğu network'e gitmek için kullanılır. Common araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Remove All Bookmarks:** Program alanına yerleştirilen tüm bookmarkları kaldırmak için kullanılır. Common araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Properties:** Üzerinde çalışılan projenin başkaları tarafından kullanılmasını engellemek için parola verilerek gizlenmesini sağlar.

#### 2.4.5. PLC Menüsü

PLC menüsü üzerine tıklandığında Şekil 2.31'de görülen PLC menüsü seçenekleri ekrana gelir.

- Ø **RUN:** PLC üzerinde bulunan manuel çalışma anahtarı kullanılmadan programdan direkt olarak cihazın çalıştırılması sağlanır. Debug araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **STOP:** Çalışmakta olan PLC 'nin üzerindeki anahtarı kullanmadan durdurulmasını sağlar. Debug araç çubuğunda kısayol ikonu bulunmaktadır.
- Ø **Compile:** Ana programda (main), alt programlarda (Subroutine) ve kesme alt programlarında kullanılan komut ve adres isimlerinin doğru karakterlerle ve eksiksiz olarak yazılıp yazılmadığını test eder.



**Şekil 2.31: PLC menüsü**

- Ø **Compile All:** Ana programda (main), alt programlarda (Subroutine) ve kesme alt programlarında kullanılan komut ve adres isimlerinin doğru karakterlerle ve eksiksiz olarak yazılıp yazılmadığını test eder. Compile seçeneği ile yapılan test taramasından farklı olarak Sistem ve Data Blok adreslerinin içerisindeki bilgiler de hatalara karşı taranır.
- Ø **Clear:** PLC program hafızasına yüklenmiş olan en son uygulama programının, PLC'nin program hafızasından silinmesini sağlar. Clear komutundan sonra PLC hafızasında hiçbir bilgi kalmaz.
- Ø **Power Up Reset:** Çalışma sırasında PLC hata uyarısı verirse uyarı lambası yanar ve programın çalışması durdurulur. Bu seçeneğe tıklandığında PLC'nin çalışma durum bilgisi resetlenerek STOP modundan tekrar RUN moduna alınacak ve hata sıfırlanacaktır.
- Ø **Information:** Bu seçenek, PLC ve ek olarak bağlanmış modüllere ait bilgileri ekrana getirir. Hata denetimi yaparak da kullanılan modülün sağlam olup

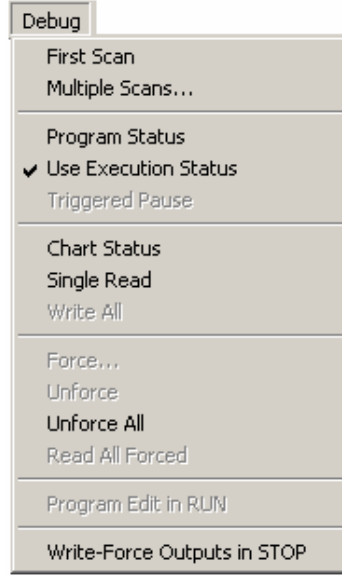
olmadığı test edilir. Ayrıca ek modül adreslerini ekrana getirerek yanlış adresleme yapılması önlenmiş olur.

- Ø **Program Memory Cartridge:** Bilgisayardan PLC'ye yüklenen program PLC'nin EEPROM belleğine kaydedilir. Ayrıca RAM alanında da yedekleme kayıt yeri bulunmaktadır. Eğer bu yedekleme alanında program saklanmak istenir ise Bu seçeneği kullanarak PLC enerji altında olmak şartıyla bu işlem gerçekleştirilebilir. Ayrıca PLC stop modunda iken PLC nin hafızasındaki programı hafıza kartına yüklemek için bu seçenek kullanılabilir.
- Ø **Create Data Block from RAM:** PLC'nin RAM hafızasının EEPROM hafızaya ve oradan da Data Blok'a kaydedilmesini sağlar.
- Ø **Time Of Day Clock:** PLC'nin gerçek zaman saati değerinin okunmasını sağlar. Program ile gün içerisinde farklı saatlerde değişik işlemler yaptırılmak isteniyor ise gerçek zaman saati uygulaması kullanılır.
- Ø **Compare:** PLC hafızasına yüklenen program ile bilgisayar ekranındaki program arasında fark olup olmadığını test etmek için kullanılır.
- Ø **Type:** Bilgisayar ile haberleşecek olan (kullanılan) PLC'nin tipini belirlemek ve adreslerini ayarlamak için kullanılır.

#### 2.4.6. Debug Menüsü

Debug menüsü üzerine tıklandığında Şekil 2.32'de görülen debug menüsü seçenekleri ekrana gelir.

- Ø **Firs Scan:** Programın çalışması sırasında bir tarama işlemi yaptırılarak hataların denetimi sağlanır. Bu işlem gerçekleştirilirken PLC stop modunda olmalıdır.
- Ø **Multiple Scans:** Programın çalışması sırasında istenilen tarama sayısı kadar program çalıştırır ve hata denetimi sağlanır. Bu işlem gerçekleştirilirken PLC stop modunda olmalıdır.
- Ø **Program Status:** PLC'nin çalışması sırasında meydana gelen değişiklikleri program penceresinde izlemek için kullanılır. Bu modda PLC çalışırken, PLC programının simülasyonu yapılır.



Şekil 2.32: Debug menüsü

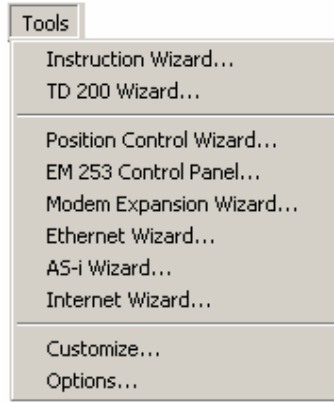
- Ø **Chart Status:** PLC'nin çalışması sırasında adreslerde meydana gelen çalışma durum bilgisini tabloya sıralar. Kullanılan adreslerin "1" veya "0" konumunda olduğunu ekranda gösterir. Bu işlem dahili kontakların konumlarının belirlenmesinde fayda sağlamaktadır.
- Ø **Single Read:** Programda, kullanılmış herhangi bir adresin çalışma durum bilgisi yenilenmiş ise durum tablosunda değişikliği ekrana getirmek amacıyla tek adres için kullanılır. PLC üzerindeki adresin durum bilgisi değişmiş ise yeniden okunması gerekir.
- Ø **Write All:** Durum tablosundan, PLC üzerinde istenilen adreslerin çalışma durum bilgileri değiştirilmek istenirse yazılan yeni değerlere göre çalışmanın yönlendirilmesini sağlar.
- Ø **Force:** PLC'ye yüklenen programda çalışması gereken herhangi bir adresin zorlanarak durdurulmasını veya tersi işlem yapılmasını sağlar. Zorlanmış konumda kilitli kalır.
- Ø **Unforce:** PLC üzerinde çalışma durum bilgisi kilitlenmiş herhangi bir adresin kilidinin açılarak normal çalışmaya dönmesini sağlar.
- Ø **Unforce All:** Kilitlenmiş tüm adreslerin hepsinin tekrar normal çalışmaya dönebilmelerini sağlar.
- Ø **Read All Forced:** Daha önceden zorlanarak kilitlenmiş adreslerin o anki durum bilgilerinin okunmasını sağlar.

- Ø **Program Edit in RUN:** PLC RUN konumunda iken projede değişiklik yapılması için kullanılır. Proje üzerinde yapılan değişiklikler RUN konumunda iken PLC'ye yüklenebilir. Program çalışırken yapılan değişiklikler tehlike yaratabilecekse gerekli önlemler alınmalıdır. Bu özellikler CPU 224 sürüm 1.1.0 daha yüksek model PLC'lerde uygulanabilir.
- Ø **Write-Force Outputs in STOP:** PLC Stop modunda iken hem analog hem de dijital adreslere değer yazma veya değiştirme imkanı sağlar.

#### 2.4.7. Tools Menüsü

Tools menüsü üzerine tıklandığında Şekil 2.33'te görülen Tools menüsü seçenekleri ekrana gelir.

- Ø **Instruction Wizard:** PLC için program hazırlanırken bazı uygulamalar karmaşık ve özel adresler kullanmayı gerektirir. Bu seçenek PID, NETR/NETW ve HSC 'lerin uygulama için gerekli adreslemelerinin kolay yapılmasını sağlayan sihirbazdır.



Şekil 2.33: Tools menüsü

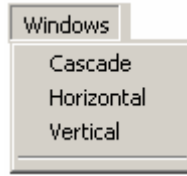
- Ø **TD 200 Wizard:** TD 200 operatör paneli için gerekli mesajların hazırlanmasına yardımcı olan sihirbazdır.
- Ø **Position Control Wizard:** PLC'ye bağlanarak kullanılabilen pozisyonlama modülünü ayarlamak ve programlamak için kullanılan sihirbazdır.
- Ø **EM 253 Control Panel:** Pozisyonlama modülünün işleyişini test etmek amacıyla kullanılan ve program tarafından sunulan kontrol panelidir.
- Ø **Modem Expansion Wizard:** PLC cihazının analog telefon hattına bağlanmasını sağlayan EM 241 modem modülünün ayarlarının kolayca yapılmasını sağlayan sihirbazdır.

- Ø **AS-i Wizard:** PLC'ye bağlanarak analog ve dijital giriş çıkış sayısının ciddi oranda artırılmasını sağlayan AS-i (CP 243 – 2) genişleme modülünün ayarlarının kolayca yapılmasını sağlayan sihirbazdır.
- Ø **Customize:** Program kısayol çubuklarının özelleştirilmesini sağlar.
- Ø **Options:** Program ekranındaki yazılım dili ve sembollerin standartlarının (yazı tipi, yazı rengi, sembol boyutları vb.) değiştirilmesini sağlar. Standart araç çubuğunda kısayol ikonu bulunmaktadır.

#### 2.4.8. Windows Menüsü

Windows menüsü üzerine tıklandığında Şekil 2.34'te görülen Windows menüsü seçenekleri ekrana gelir.

- Ø **Cascade:** Programda açık durumdaki uygulama pencerelerinin arka arkaya sıralanarak görüntülenmesini sağlar.

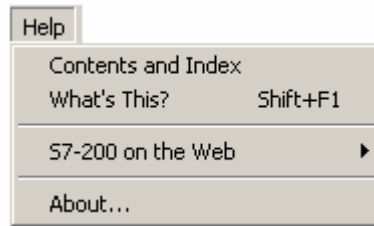


Şekil 2.34: Windows menüsü

- Ø **Horizontal:** Programda açık durumdaki uygulama pencerelerinin yatay olarak sıralanarak görüntülenmesini sağlar.
- Ø **Vertical:** Programda açık durumdaki uygulama pencerelerinin dikey olarak sıralanarak görüntülenmesini sağlar.

#### 2.4.9. Help Menüsü

- Ø **Contents and index:** Program ile ilgili yardım konularına kolayca ulaşmak için kullanılır. Bu kısımdan program ile ilgili her türlü bilgiye ulaşabilirsiniz.

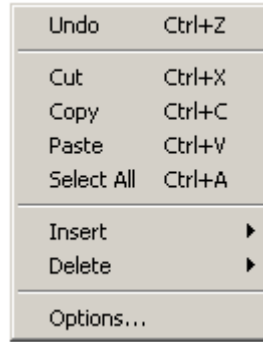


Şekil 2.35: Help menüsü

- Ø **What is this:** Bu seçeneğe tıklandığında fare işaretçisi ile birlikte bir soru işareti sembolü görülür. Soru işaretini program alanında hakkında bilgi almak istediğimiz nesnenin üzerine götürüp tıklayarak yardım alabilirsiniz.
- Ø **About:** Kullanılan Microwin yazılımı hakkında bize bilgi verir.

## 2.4.10. Kısayol Menüsü

Şekil 2.36’da görülen kısayol menüsü, programlama editöründe çalışırken fare sağ tuşuna tıklandığında ekrana gelir. Bu menüye program yazımı sırasında sıkça başvurduğumuz Edit menüsü seçenekleri yerleştirilmiştir. Ayrıca kısayol menüsünden Options iletişim kutusuna erişebilirsiniz.



Şekil 2.36: PLC menüsü

## 2.5. Program Pencereleri ve Genel Kullanımı

### 2.5.1. Programlama Editörü Penceresi

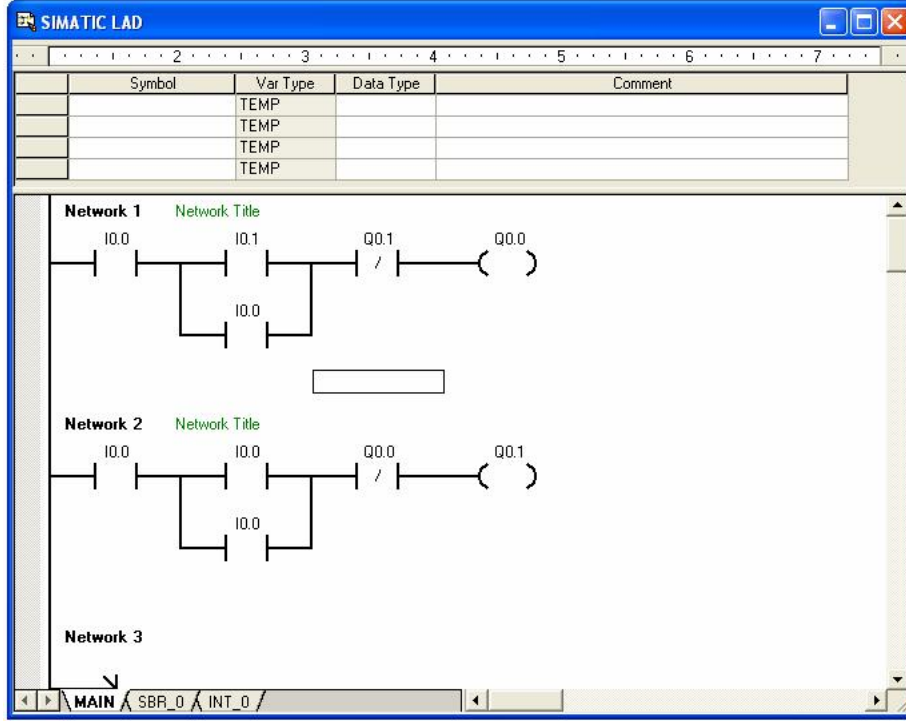
PLC programının yazılımı bu editörle gerçekleşir. Şekil 2.37’de görüldüğü gibi editör Network (devre) adı verilen bölümlere ayrılmıştır. Her network kontrol probleminin çözümünde bir adımı temsil eder. Böylece kontrol probleminin geliştirilmesi ve hataların tespiti daha kolay yapılabilir.

Step7 Micro/Win 32 V3.2 programlama editörü ile Ladder Plan, STL ve FBD yöntemlerinin herhangi biriyle program yazımı gerçekleştirilebilir. Hangi yöntemle yazmak istiyorsak View menüsünden seçmemiz gerekir. Seçilen yönteme bağlı olarak program editörünün görünümünde bir değişiklik olmaz. Ancak hangi yöntemi seçersek sadece o yönteme ait komut ya da sembolleri kullanabiliriz.

Programlama editörü yapısal programlamayı kolaylaştıracak şekilde düzenlenmiştir. Ana program, alt programlar ve kesme alt programları ayrı ayrı sayfalara yazılır. MAIN ana programın yazıldığı sayfadır. Bir kontrol programında sadece bir main sayfası olabilir. SBR\_0, SBR\_1, SBR\_2... şeklinde isimlendirilen sayfalara Subroutine alt programları yazılır. INT\_0, INT\_1, INT\_2... şeklinde isimlendirilen sayfalara Interrupt (kesme) alt programları yazılır. Böylece ayrı ayrı sayfalarda bulunan ana program ve alt programların denetimi kolayca yapılabilir. Alt programlar daima ana programdan çağrılarak işletilir. Kesme alt programları ise ana programdan yetkilendirilerek bir dış birim tarafından çalıştırılır.

Editörün üst kısmında bulunan tablo alt programlara parametre göndermek için kullanılır.





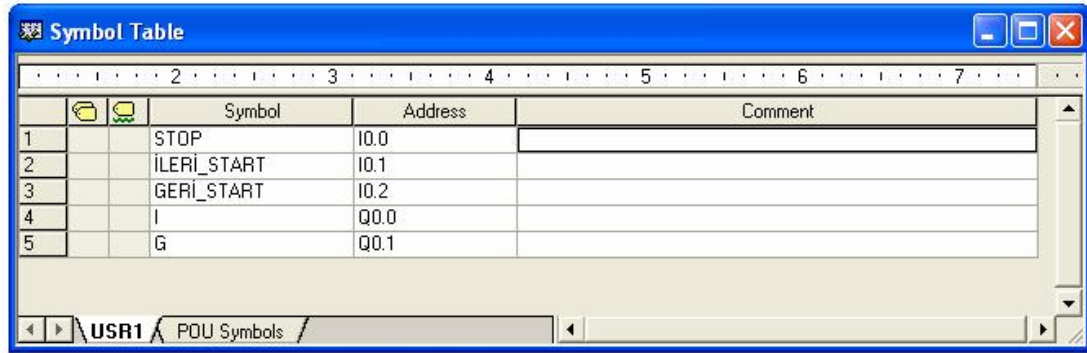
Şekil 2.37: Programlama editörü menüsü

## 2.5.2. Symbol Table ( Sembol tablosu) Penceresi

PLC 'de girişler, çıkışlar ve dahili elemanların birer adresleri vardır. Bu adreslerin ifade ediliş şekli PLC markalarına göre değişiklik gösterir. PLC üreticileri tarafından belirlenen bu adresler mutlak adreslerdir. Örneğin S7 - 200 PLC'lerde girişler I0.0, I0.1,I0.2 .. ve çıkışlar Q0.0,Q0.1,Q0.2... şeklinde adreslenir.

Büyük çaplı programlarda mutlak adreslerin kullanılması durumunda, programın yazımı, takibi ve hataların bulunması zorlaşır. Bu nedenle mutlak adreslere üstlendikleri görevi ifade edecek şekilde sembolik isimler verilebilir. Böylece program yazımı sırasında mutlak adresler yerine sembolik adresleri kullanarak programımızın anlaşılabilirliğini arttırmış oluruz.

Mutlak adreslere sembolik isimler atamak için Şekil 2.38'de görülen sembol tablosu penceresi kullanılır. Tablonun Symbol sütununa bizim tarafımızdan belirlenen sembolik isimler yazılır. Birden fazla adrese aynı isim verilemez. Address sütununa ise temsil ettikleri mutlak adresler yazılır. Comment isimli sütun açıklama sütunudur. Bu alana kullanılan adresin kullanım amacı ile ilgili açıklayıcı bilgi yazılabilir. Bu alan program tarafından dikkate alınmaz. Bu yüzden bu alanı doldurmak şart değildir.

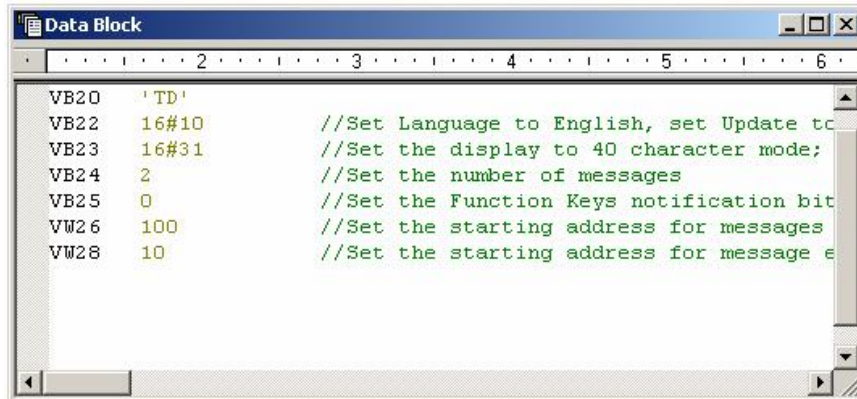


Şekil 2.38: Symbol table penceresi

### 2.5.3. Data Block ( Data bloğu) Penceresi

Şekil 2.39’da görülen Data blok editörü, PLC’nin sadece V hafıza alanlarına değişik uygulamalar için gerekli verileri yüklemek için kullanılır. Örneğin PLC’ye TD 200 operatör paneli bağlanacaksa, LCD ekranda görünmesi gereken mesajları Data block editörünü kullanarak PLC nin istenilen V hafıza adreslerine yükleyebiliriz.

Data block editörü ile V hafıza alanlarına binary, desimal, heksadesimal ve ASCII karakter formatında veri girişi yapılabilir.



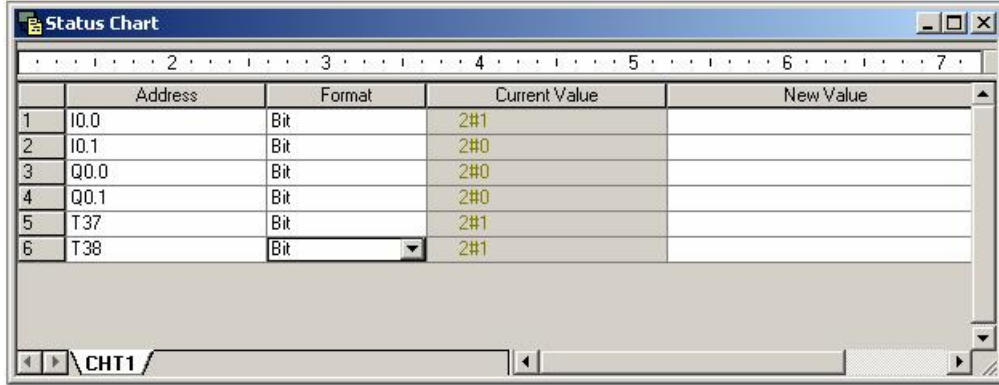
Şekil 2.39: Data Block penceresi

### 2.5.4. Status Chart (Durum Tablosu) Penceresi

Şekil 2.40 da görülen Durum tablosu, PLC hazırlanan programı işletirken program değişkenlerinin (adreslerinin) almış olduğu değerlerin izlenebilmesini ve değiştirilmesini sağlar. Böylece herhangi bir anda izlemek istediğiniz adreslerin almış oldukları değerleri görerek projenin test çalışmalarını kolaylıkla yapabilirsiniz, ayrıca adreslere sizin belirlediğiniz değerleri girerek program performansına etkisini gözleyebilirsiniz. Programınızın değişik kısımlarındaki değişik adresleri izlemek amacıyla birden çok durum tablosu oluşturabilirsiniz.

Şekilde görülen durum tablosunun Address sütununa izlemek istediğiniz adresleri, Format sütununa izlenmek istenen adresin kullandığı veri tipini girmeniz gerekir. PLC 'yi RUN moduna aldıktan sonra Current Value sütunundan adreslerin değerlerini izleyebilirsiniz. New Value sütundan ise seçtiğiniz bir adresin sizin istediğiniz değeri almasını sağlayabilirsiniz.

Programda kullan sabit değerlerin, akümülatör alanlarının ve Lokal hafıza alanlarının durumları bu pencereden izlenemez.

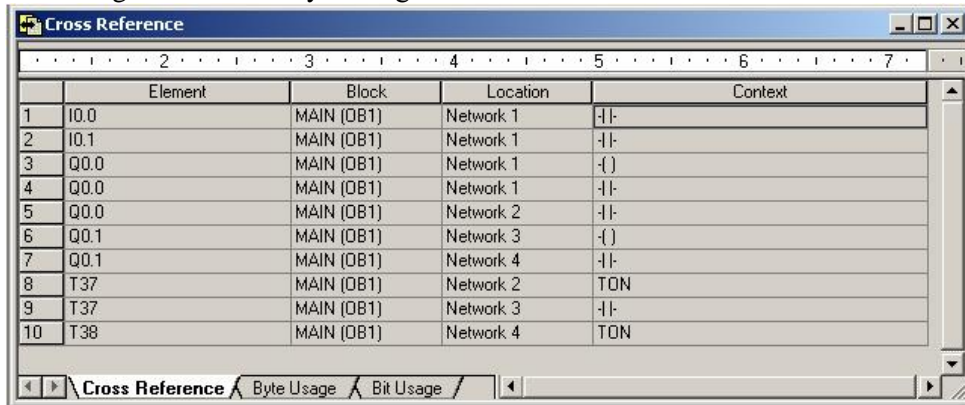


	Address	Format	Current Value	New Value
1	I0.0	Bit	2#1	
2	I0.1	Bit	2#0	
3	Q0.0	Bit	2#0	
4	Q0.1	Bit	2#0	
5	T37	Bit	2#1	
6	T38	Bit	2#1	

Şekil 2.40: Status chart penceresi

### 2.5.5. Cross Reference (Çapraz Referans) Penceresi

Çapraz referans tablosu programda kullandığınız mutlak ya da sembolik adreslerin programın hangi bölümlerinde yer aldığını listeler.

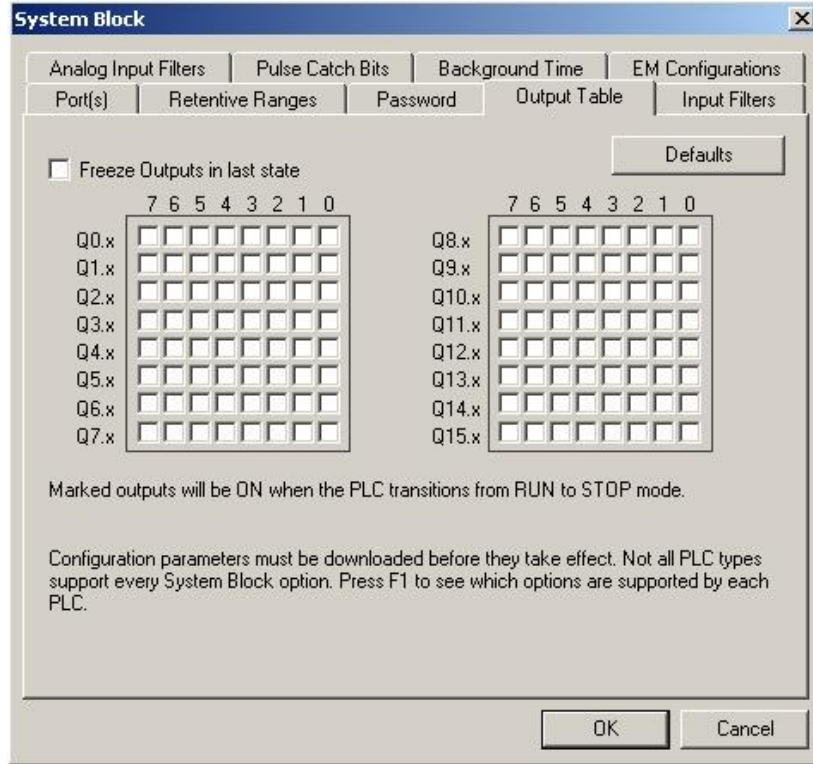


	Element	Block	Location	Context
1	I0.0	MAIN (OB1)	Network 1	-I-
2	I0.1	MAIN (OB1)	Network 1	-I-
3	Q0.0	MAIN (OB1)	Network 1	-O-
4	Q0.0	MAIN (OB1)	Network 1	-I-
5	Q0.0	MAIN (OB1)	Network 2	-I-
6	Q0.1	MAIN (OB1)	Network 3	-O-
7	Q0.1	MAIN (OB1)	Network 4	-I-
8	T37	MAIN (OB1)	Network 2	TON
9	T37	MAIN (OB1)	Network 3	-I-
10	T38	MAIN (OB1)	Network 4	TON

Şekil 2.41: Çapraz referans tablosu penceresi

### 2.5.6. System Block (Sistem bloğu) Penceresi

Şekil 2.42’de görülen Sistem bloğu penceresi PLC’nin çalışma şeklini etkileyecek değişik donanım seçeneklerinin ayarlanmasını sağlar. Yeterince bilgi ve tecrübe sahibi olmadan buradaki ayarlar rastgele değiştirilmemelidir.



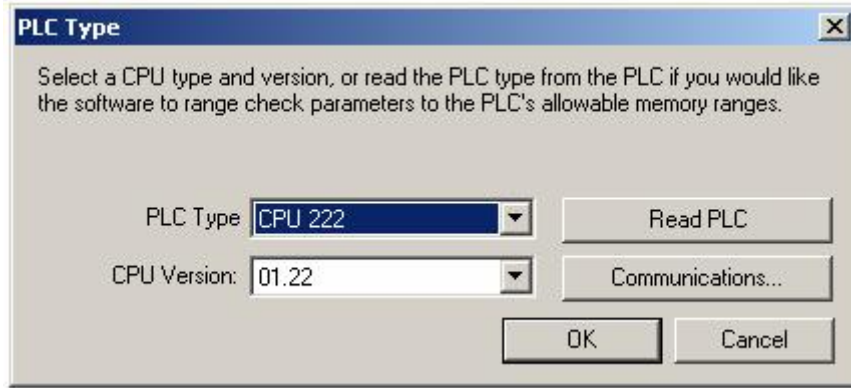
Şekil 2.42: System Block penceresi

### 2.6. PLC Tipinin Seçilmesi

Proje hazırlarken programda kullandığımız adreslerin, seçtiğimiz PLC ‘nin CPU tipi ile uyum içinde olması gerekir. Aksi takdirde hazırladığımız programı PLC hafızasına gönderirken sorunla karşılaşabiliriz.

PLC tipinin seçilmesi için PLC menüsünden Type... seçeneğine tıklanarak şekil 2.43’te görülen iletişim kutusunun ekrana gelmesi sağlanır. Buradaki açılır liste kutularından projede kullanacağımız PLC tipini ve CPU versiyonunu seçtikten sonra Read PLC butonuna tıklamamız gerekir.

**Communications** ... düğmesini tıklayarak iletişim ayarlarını kontrol edebilirsiniz. İletişim ayarları ile ilgili bilgiler için Öğrenme Faaliyeti 2 ‘ye bakınız.



Şekil 2.43: PLC tipinin seçme iletişim kutusu penceresi

## 2.7. Hafıza Alanları ve Adreslenmeleri

PLC'lerde veriler değişik kullanım amaçları için ayrılmış hafıza alanlarında bulunur. Herhangi bir hafıza alanındaki veriye erişmek için adresinin kullanılması gerekir. Adresler Bit, Bayt, Word ve Double Word olarak ifade edilebilir.

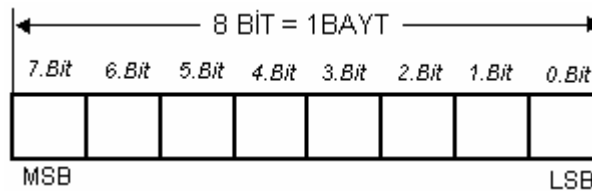
### 2.7.1. Bit, Bayt ve Word kavramları

- Ø **BİT:** Dijital sistemlerde kullanılan en küçük hafıza birimidir. Şekil 2.44 ile temsil edilen bir bitlik alan içerisinde 0 ya da 1 verisi depolanabilir. Bitsel işlem yapan komutlar bit adreslerini kullanırlar.



Şekil 2.44: Bir bitlik hafıza alanı

- Ø **BYTE:** 8 bitlik bir hafıza alanı bir bayt ile ifade edilir. En anlamlı veri ( sayı, harf vb.) bir baytlık alanda saklanır. Şekil 2.45 ile temsil edilen bir baytlık hafıza alanında desimal olarak 0..255 arasındaki işaretli sayılar, -128 ...+127 arasındaki işaretli sayılar ( tam sayılar ) ikilik sisteme çevrilerek depolanabilir. LSB en düşük değerlikli bit, MSB en yüksek değerlikli bittir. İşaretli sayıların depolanmasında MSB biti işaret biti olarak kullanılır. Bu bir 1 ise sayı pozitif, 0 ise sayı negatiftir.



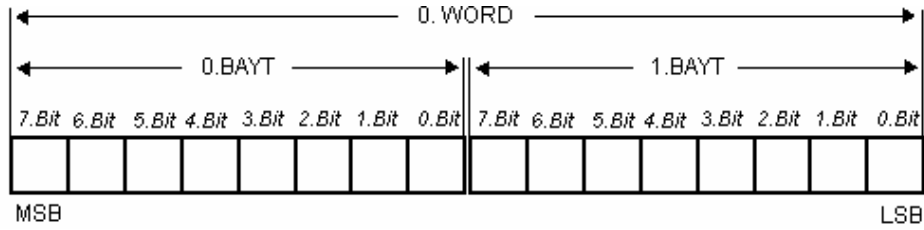
Şekil 2.45: Bir bayt hafıza alanı

Tablo 2.1’de bir baytlık alanda bulunabilecek sayı türleri ve değer aralıkları görülmektedir. S7 – 200 PLC lerde işaretli tam sayılar bir bayt alanından daha uzun hafıza alanlarında bulunurlar.

SAYI TÜRÜ	DEĞER ARALIĞI
BİNARY	00000000 ...11111111
İŞARETSİZ DESİMAL	0 ...255
HEKSADESİMAL	00 ... FF

**Tablo 2.1: Bir bayt alanında bulunabilecek sayı türleri**

- Ø **WORD:** 2 baytlık bir alan bir word ile ifade edilir. Diğer bir ifade ile bir word 16 bitlik bir hafıza alanını temsil eder. Bir wordlük veri alanını oluşturan baytlardan byte numarası küçük olan yüksek değerli, bayt numarası büyük olan ise düşük değerlidir.



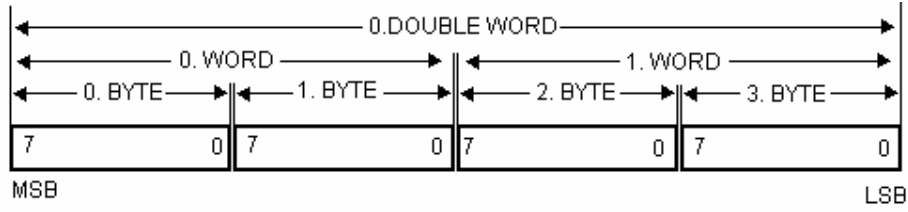
**Şekil 2.46: Bir word hafıza alanı**

Tablo 2.2’de bir wordlük alanda saklanabilecek sayı türleri ve değer aralıkları görülmektedir. Tablonun çok uzun olmasına neden olacağından Binary sayı aralıkları gösterilmemiştir.

SAYI TÜRÜ	DEĞER ARALIĞI
İŞARETSİZ DESİMAL	0 ...65536
İŞARETLİ DESİMAL (TAMSAYI)	-32768...+32767
HEKSADESİMAL	0000 ... FFFF

**Tablo 2.2: Bir Word alanında bulunabilecek sayı türleri**

- Ø **DOUBLE WORD:** 2 wordlük bir hafıza alanı bir double word ile ifade edilir. Diğer bir ifade ile bir double word 4 byte ya da 32 bitlik bir hafıza alanını temsil eder. Bir double wordlük veri alanını oluşturan wordlerden word numarası küçük olan yüksek değerli, word numarası büyük olan ise düşük değerlidir.



**Şekil 2.47: Bir double word hafıza alanı**

Aşağıdaki tabloda bir double word'luk alanda saklanabilecek sayı türleri ve değer aralıkları görülmektedir. Tablonun çok uzun olmasına neden olacağından Binary sayı aralıkları gösterilmemiştir.

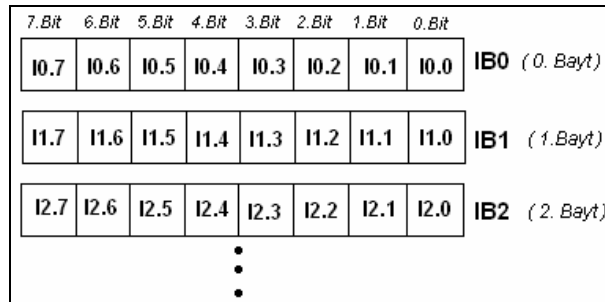
SAYI TÜRÜ	DEĞER ARALIĞI
İŞARETSİZ DESİMAL	0 ...4294967295
İŞARETLİ DESİMAL (TAMSAYI)	- 2147483648...+2147483648
HEKSADESİMAL	00000000 ... FFFFFFFF
REEL SAYILAR	+1.175495E-38 ...-1.175495E-38

**Tablo 2.3: Bir Word alanında bulunabilecek sayı türleri**

### 2.7.2. Girişler

PLC'ye giriş modülünden hatırlayacağınız gibi girişlere buton, sensör, sınır anahtarı, temassız algılayıcı vb. elemanlar bağlanır. Bu elemanlarda meydana gelen konum değişiklikleri bağlı bulundukları girişlere gerilim uygulanmasına ya da var olan gerilimin kesilmesine neden olur. Girişlerdeki gerilim değişikliği giriş adreslerinde lojik bir değişim olarak algılanır. Bu lojik değişim PLC programı tarafından okunarak değerlendirilir.

PLC'de girişler Şekil 2.48'deki gibi kendileri için ayrılmış 8 bitlik bayt alanları ile adreslenir. Bu adreslemede I (Input) giriş anlamına gelen alan belirteçidir.



**Şekil 2.48: PLC girişlerinin adreslenmesi**

PLC tipine göre girişler için ayrılmış bayt alanlarının sayısı değişiklik gösterir. Girişler için belli bir byte alanı ayrılmasına rağmen PLC üzerinde belli sayıda giriş bulunur. Diğer bir ifade ile giriş için ayrılan bayt alanlarının hepsi aktif değildir. Mevcut tesise zamanla yeni kumanda ve kontrol birimlerinin eklenmesi ile giriş sayısı yetersiz kalabilir.

Böyle durularda yeni giriş modülleri eklenir. PLC tipine göre eklenebilecek giriş modülü sayısı sınırlıdır. Her eklenen yeni modül bir bayt adresini kullanır.

Tablo 2.4’de bazı PLC’lerin üzerlerinde bulunan mevcut giriş sayıları ve adreslenebilir giriş aralıkları görülmektedir. Burada dikkat edilirse bazı PLC tiplerinde mevcut giriş sayısı bayt alanlarını tam olarak doldurmamıştır. Örneğin CPU 221 PLC de I0.0 ...I0.5 olmak üzere 6 giriş bulunmaktadır. Oysa ki bir baytlık alan için I0.6 ve I0.7 adreslerinde mevcut olması gerekir. Böyle bir PLC’ye yeni bir giriş modülü eklendiğinde yeni modülün PLC üzerinde boş kalan adreslere etkisi olmaz. I0.6 ve I0.7 adresli çıkışlar kullanılamaz. Aynı durum çıkışlar için de geçerlidir.

PLC TİPİ	CPU 221	CPU 222	CPU 224	CPU 226
<b>MEVCUT GİRİŞ ADRESLERİ</b>	I0.0 ... I0.5 ( 6 Giriş)	I0.0...I0.7 (8 Giriş)	I0.0 ... I1.5 (14 Giriş)	I0.0 ... I2.7 (24 Giriş)
<b>ADRESLENEBİLİR GİRİŞ ARALIĞI</b>	I0.0 ... I15.7	I0.0 ... I15.7	I0.0 ... I15.7	I0.0 ...I15.7

**Tablo 2.4**

Bitsel işlem yapan komutlar operant olarak girişlerin bit adreslerini kullanırlar. Giriş bit adreslerine erişmek için aşağıdaki format kullanılır.

**I[Bayt numarası].[Bit numarası] I0.4** 0. giriş baytının 4. biti

Ladder diyagramlarında giriş bit adresleri kontak olarak gösterilir. Normalde açık ya da kapalı olan kontağın temsil ettiği giriş bitindeki değişim kontağın kapanarak enerji akışına izin vermesini ya da açılarak enerji akışını kesmesini sağlar.

I0.0  
—| |— I0.0 giriş biti 1 olursa kontak kapanarak enerji akışı sağlar.

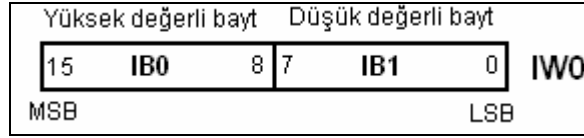
I0.3  
—| / |— I0.3 giriş biti 1 olursa kontak açılarak enerji akışını keser.

Fonksiyonel komutlar girişlere Bayt, Word ve Double word boyutunda erişim sağlar. Bu tür adresleme de aşağıdaki format kullanılır.

**I[Alan boyutu ].[Byte numarası] ( Alan boyutu B,W,D harfleriyle belirtilir)**  
**IB5 IW0 ID0**

Word uzunluğunda adres tanımlaması yaparken hafıza alan belirteci ve alan boyutu belirteci yazıldıktan sonra başlangıç bayt numarası yazılır. Yazılan bayt numarası alanı bir wordlük hafıza alanının yüksek değerli baytını temsil eder. Sıradaki bayt adresi ise bir wordlük alanın düşük değerlikli bayt alanıdır. Bu kural bütün hafıza alanları için geçerlidir. Şekil 2.49’da 16 bitlik bir giriş adres alanının tamamının bir word adresiyle ifade edildiği görülmektedir.

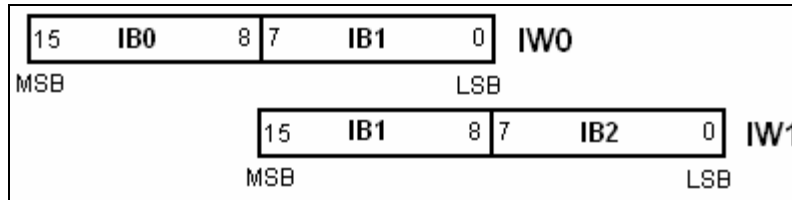




**Şekil 2.49: Bir word alanının oluşumu**

Eğer kullandığınız PLC'ye modül ekleyerek giriş sayısını arttırırsanız, yeni eklediğiniz girişlerin adreslendiği bayt alanı düşük değerlidir. Bu kural çıkış adresleri içinde geçerlidir.

Word uzunluğundaki adreslemeler IW0, IW2, IW4,...şeklinde 2 aralıklı olarak yapılmalıdır. Aksi takdirde tanımlanan word adresleri birbirlerinin bayt alanlarını kullanırlar. Bu durum programların işletilmesinde yanlış sonuçların alınmasına neden olabilir. Şekilde IW0 ve IW1 adresleri IB1 alanını ortak kullanmaktadır.



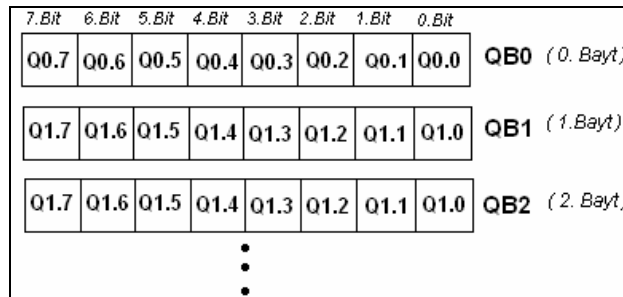
**Şekil 2.50: Word alanlarının adreslenmesi**

Aynı kurallar Double word uzunluğundaki adreslemeler için de geçerlidir. Double word uzunluğundaki adreslemeler VD0, VD4, VD8, ... şeklinde 4 aralıklı olarak yapılmalıdır.

### 2.7.3. Çıkışlar

PLC çıkışlarına Valf, lamba, Röle, Motor vb. çıkış elemanları bağlanır. Çıkışlar işletilen PLC programı tarafından kontrol edilir. Çıkışların akım verme kapasiteleri sınırlıdır. Bu nedenle kapasitenin üzerinde akım çeken yükler doğrudan PLC çıkışlarına bağlanmazlar.

PLC'de çıkışlar Şekil 2.51'deki gibi kendileri için ayrılmış 8 bitlik byte alanları ile adreslenir. Alan belirteci olarak çıkış anlamına gelen Q ( Quit ) harfi kullanılır.



**Şekil 2.51: PLC çıkışlarının adreslenmesi**

Girişlerde olduğu gibi çıkışlar için de belli bir bayt alanı ayrılmıştır. PLC'ler tiplerine göre üzerinde belli sayıda çıkış bulundurulur. Eğer tesiste oluşabilecek genişlemeler nedeniyle çıkış sayısı yetersiz kalırsa çıkış modülleri eklenerek çıkış kapasitesi artırılabilir. Tablo 2.5'te bazı PLC'lerin üzerlerinde bulunan mevcut çıkış sayıları ve adreslenebilir çıkış aralıkları görülmektedir.

PLC TİPİ	CPU 221	CPU 222	CPU 224	CPU 226
<b>MEVCUT ÇIKIŞ ADRESLERİ</b>	Q0.0 ... Q0.3 ( 4 Çıkış)	Q0.0...Q0.5 ( 6 Çıkış)	Q0.0 ... Q1.1 ( 10 Çıkış)	Q0.0 ... Q1.7 ( 16 Çıkış)
<b>ADRESLENEBİLİR ÇIKIŞ ARALIĞI</b>	Q0.0 ... Q15.7	Q0.0 ... Q15.7	Q0.0 ... Q15.7	Q0.0 ...Q15.7

**Tablo 2.5**

Çıkış adres bitlerine erişmek için kullanılacak adresin yazım formatı aşağıdaki gibidir.

**Q[Bayt numarası].[Bit numarası] Q 0.0 Q 1.5**

Ladder diyagramlarında çıkış bit adresleri çıkış sembolü ile gösterilir. Herhangi bir çıkış bitinin değeri lojik 1 ise PLC'nin o çıkışından gerilim alınır ve çıkışa bağlı alıcı enerjilenir. Çıkış bit adresleri aynı zamanda bir kontağa atanabilir. Normalde açık ya da kapalı olan kontağın temsil ettiği çıkış bitindeki değişim kontağın kapanarak enerji akışına izin vermesini ya da açılarak enerji akışını kesmesini sağlar.

—( <sup>Q0.0</sup> ) Q0.0 çıkış biti 1 olursa bu çıkışa bağlı alıcı enerjilenir.

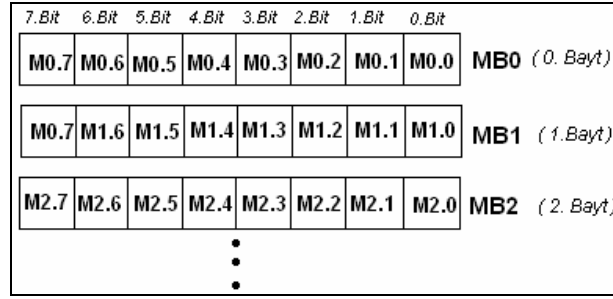
Çıkış hafızasına Bayt, Word ve Double word boyutunda erişmek için aşağıdaki format kullanılır.

**Q[Alan boyutu ].[Byte numarası] QB0 QW0**

#### 2.7.4. M Hafıza Alanları

PLC programlarının yazımı sırasında karmaşık fonksiyonlu işlemlerin kolaylıkla gerçekleştirilebilmesi için işlemlerin ara sonuçlarının saklanabileceği hafıza alanlarına ihtiyaç vardır. Bu alanlara M hafızası ya da durum tespit işaretçileri adı verilir. Durum tespit işaret alanlarını klasik kumanda devrelerinde kullanılan yardımcı röleler gibi düşünebiliriz. PLC bünyesinde belli bir bayt alanı M hafıza için ayrılmıştır.

M hafıza alanları Şekil 2.52'de görüldüğü gibi kendileri için ayrılmış 8 bitlik bayt alanları ile adreslenir. Hafıza alan belirteci M harfidir.



**Şekil 2.52: M hafıza alanlarının adreslenmesi**

Tablo 2.6 da bazı PLC modellerinin M hafıza adres aralıkları görülmektedir.

PLC TİPİ	CPU 221	CPU 222	CPU 224	CPU 226
<b>M HAFİZA ARALIĞI</b>	M0.0 ... M31.7 (32 BAYT)	M0.0 ... M31.7 (32 BAYT)	M0.0 ... M31.7 (32 BAYT)	M0.0 ... M31.7 (32 BAYT)

**Tablo 2.6**

M hafıza bitlerine erişmek için kullanılacak adresin yazım formatı aşağıdaki gibidir.

**M [Bayt numarası].[Bit numarası] M2.4**

Ladder diyagramlarında işlem ara sonucunun bir M hafıza bitine atama işlemi çıkış sembolü kullanılarak gerçekleştirilir. M hafıza bit adresleri aynı zamanda bir kontağa atanabilir. Normalde açık ya da kapalı olan kontağın temsil ettiği M hafıza bitindeki değişim kontağın kapanarak enerji akışına izin vermesini ya da açılarak enerji akışını kesmesini sağlar.

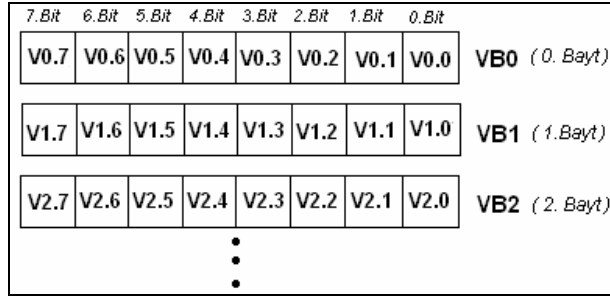
M hafızasına Bayt, Word ve Double word boyutunda erişmek için aşağıdaki format kullanılır.

**M[Alan boyutu].[Byte numarası] MW0 MD4**

### 2.7.5. V Hafıza (Data) Alanları

Data alanlarına değişken hafıza alanı ya da kısaca V hafızası adı verilir. V hafıza alanları program akışı sırasında oluşan ara sonuçları saklamak için kullanılabileceği gibi proses için gerekli değişkenleri ve sabitleri yazmak için de kullanılabilir.

V hafıza alanları Şekil 2.53'te görüldüğü gibi kendileri için ayrılmış 8 bitlik byte alanları ile adreslenir. Hafıza alan belirteci V harfidir.



**Şekil 2.53: V hafıza alanlarının adreslenmesi**

V hafıza alanının büyüklüğü kullanılan PLC'nin modeline göre değişmektedir. Tablo 2.7 de bazı PLC modellerinin V hafıza adres aralıkları görülmektedir.

PLC TİPİ	CPU 221	CPU 222	CPU 224	CPU 226 XM
<b>V HAFIZA ARALIĞI</b>	VB0 ... VB2047 (2047 BAYT)	VB0 ... VB2047 (2047 BAYT)	VB0 ... VB5119 (5119 BAYT)	VB0 ... VB10239 (10239 BAYT)

**Tablo 2.7**

V hafıza bitlerine erişmek için kullanılacak adresin yazım formatı aşağıdaki gibidir.

**V [Bayt numarası].[Bit numarası] V25.2**

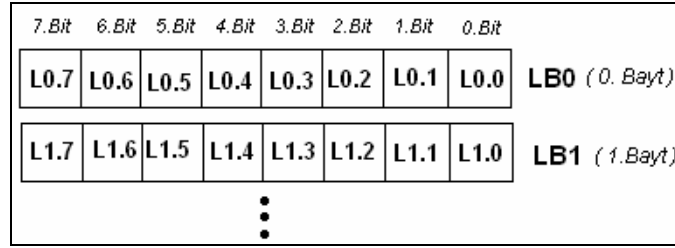
V hafızasına Bayt, Word ve Double word boyutunda erişmek için aşağıdaki format kullanılır.

**V[Alan boyutu ].[Byte numarası] VB100 VW47 VD5**

#### **2.7.6. L (Lokal) Hafıza Alanları**

Lokal hafıza, ana program ve her alt program için ayrı ayrı ayrılmış 64 bayt büyüklüğündeki hafıza alanlarıdır. Lokal hafıza alanlarının esas kullanım amacı alt programlara değişken parametreler göndermektir. Bunun yanında herhangi bir amaç içinde kullanılabilir. Program Ladder veya FBD editörleri ile yazılıyorsa Lokal hafıza alanlarının son 4 baytlık kısmı program tarafından kullanılır. Bu durumda programcı ilk 60 bayte'lık kısmını kullanabilir. STL editörü ile çalışılması durumunda 64 bayt'lık lokal hafıza alanının tamamı kullanılabilir.

L hafıza alanları Şekil 2.54'te görüldüğü gibi kendileri için ayrılmış 8 bitlik byte alanları ile adreslenir. Hafıza alan belirteci L harfidir.



Şekil 2.54: L hafıza alanlarının adreslenmesi

L hafıza bitlerine erişmek için kullanılacak adresin yazım formatı aşağıdaki gibidir.

**L [Bayt numarası].[Bit numarası] L0.0**

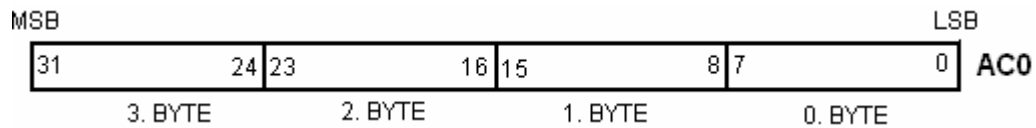
L hafızasına Bayt, Word ve Double word boyutunda erişmek için aşağıdaki format kullanılır.

**L[Alan Boyutu].[Byte numarası] LB50 LW5**

### 2.7.7. AC (Akümülatör) Hafıza Alanları

Akümlatörler, okuma ve yazma yapılabilecek hafıza alanlarıdır. Akümülatörler V, M, Q gibi bazı hafıza alanlarına veri aktarmak veya bu alanlardan veri almak için kullanılabilir. Ayrıca akümülatör hafıza alanlarını alt programlara parametre göndermek için de kullanabiliriz.

S7 – 200 PLC’lerde 32 bit ( Double Word) uzunluğunda 4 adet akümülatör bulunur. Akümülatör alanları AC0, AC1, AC2, AC3 şeklinde adreslenir. Akümülatör içeriklerine Byte, Word ve Double Word uzunluğunda erişebiliriz. Bir akümülatör alanına hangi uzunlukta erişileceği kullanıldığı komutun işlem uzunluğuna bağlıdır.



Şekil 2.54: AC hafıza alanlarının adreslenmesi

Şekil 2.54’te AC0 akümülatör hafıza alanı görülmektedir. Akümülatör adresine bayt uzunluğunda işlem yapan bir komut ile erişiliyorsa sadece 0.bayt, word uzunluğunda işlem yapan bir komut ile erişiliyorsa 0.bayt ve 1.bayte, double word uzunluğunda işlem yapan bir komut ile erişiliyorsa akümülatör alanının tamamı kullanılır.

### 2.7.8. HC (Hızlı sayıcı HSC) Hafıza Alanları

Hızlı sayıcılar 20 KHz – 30 KHz gibi yüksek frekanslı sinyallerin sayılmasında kullanılırlar. Normal sayıcılarda olduğu gibi yukarı sayıcı, aşağı sayıcı ve aşağı yukarı sayıcı

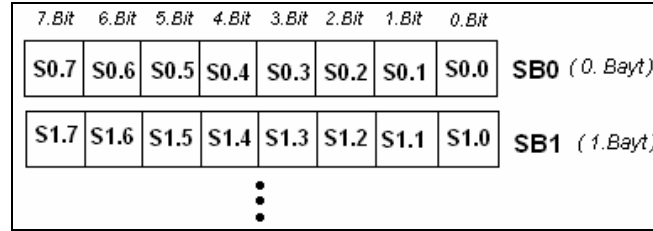
olarak kullanılabilirler. Hızlı sayıcıların giriş sinyali kare dalga olarak tek sinyal olabildiği gibi çift sinyal de olabilir. Hızlı sayıcıların sayma işlemini gerçekleştirebilmeleri için programlanmaları gerekir.

Hızlı sayıcıların saymış olduğu anlık değerler kendileri için ayrılmış double word uzunluğundaki hafıza alanlarında saklanır. Bu alanlara erişmek için hızlı sayıcı adresleri kullanılır. Yeni nesil S7 – 200 PLC’lerde 6 adet hızlı sayıcı bulunmaktadır. Hızlı sayıcı hafıza alanları HC0, HC1, HC2, HC3, HC4, HC5 şeklinde adreslenir. Hızlı sayıcı adreslerine yalnızca double word uzunluğunda işlem yapan komutlarla erişilebilir.

### 2.7.9. S (Sıralayıcı kontrol rölesi SCR ) Hafıza Alanı

SCR’ler birbirini takip eden işlem adımları şeklinde çalışan makineler için kolay programlama imkanı sağlar. SCR’ler kumanda programının lojik segmentler halinde ifade edilmesine olanak verir. Bu segmentler kullanılarak programın içerisine birbirini takip eden adımlardan oluşan sıralı bir akış yerleştirilebilir.

SCR’ler ile sıralı çalışan program adımları oluşturulurken S hafıza alanları kullanılır. Sıralayıcı hafıza alanları Şekil 2.55’te görüldüğü gibi kendileri için ayrılmış 8 bitlik bayt alanları ile adreslenir.



Şekil 2.55: S hafıza alanlarının adreslenmesi

S hafıza bitlerine erişmek için kullanılacak adresin yazım formatı aşağıdaki gibidir.

**S[Bayt numarası].[Bit numarası] S1.0**

S hafızasına Bayt, Word ve Double word boyutunda erişmek mümkündür. Bu tür adreslemede aşağıdaki format kullanılır.

**S[Alan boyutu ].[Byte numarası] SB0 SW5**

S7 – 200 CPU 221 ve daha üst model PLC’lerde **S0.0 ...S31.7** aralığında 32 baytlık S hafıza alanı bulunmaktadır.

### 2.7.10. Analog Giriş ve Analog Çıkış Hafıza Alanları

S7 – 200 PLC'lere bağlanacak analog giriş çıkış modülleri ile sıcaklık, basınç gibi analog değerler 16 bitlik dijital değerlere çevrilirken, 16 bitlik dijital değerler, dijital değerlerle orantılı bir akım veya gerilim değerlerine dönüştürülebilir.

Analog girişlerden elde edilen word uzunluğundaki bilgiye erişmek için analog giriş adresleri kullanılır. Analog girişler AIW0, AIW2, AIW4, AIW6 şeklinde adreslenir ve bu adreslerdeki bilgiye sadece word uzunluğunda işlem yapan komutlar ile erişilebilir.

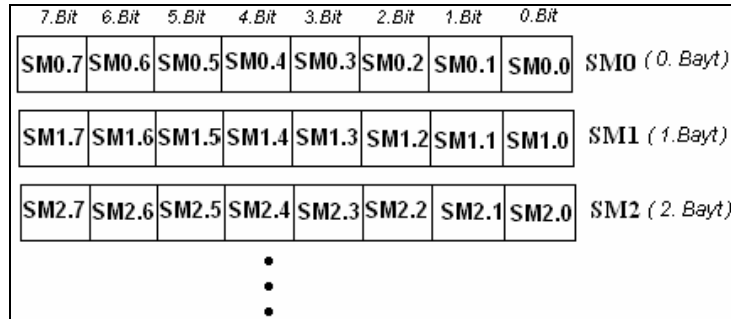
Analog çıkışlar ise AQW0, AQW2, AQW4 şeklinde adreslenir ve girişlerde olduğu gibi bu adreslere de word uzunluğunda işlem yapan komutlar ile erişilebilir.

Analog giriş ve çıkış adres sayısı analog giriş çıkış modülünde bulunan analog giriş ve analog çıkış sayısı kadardır.

### 2.7.11. Özel Hafıza Alanları

Özel hafıza alanları CPU ile program arasında iletişim sağlayacak çeşitli kontrol fonksiyonlarını gerçekleştirmeyi sağlar. Özel hafıza alanlarına genellikle bit düzeyinde erişim sağlanır ve her bir özel hafıza biti özel bir görevi üstlenir.

Özel hafıza alanları Şekil 2.56'da görüldüğü gibi kendileri için ayrılmış 8 bitlik byte alanları ile adreslenir. Hafıza alan belirteci olarak SM kullanılır.



Şekil 2.56: SM hafıza alanlarının adreslenmesi

Özel hafıza alanının büyüklüğü kullanılan PLC'nin modeline göre değişmektedir. Tablo 2.8 bazı PLC modellerinin SM hafıza adres aralıkları görülmektedir.

PLC TİPİ	CPU 221	CPU 222	CPU 224...226XM
SM ADRES ARALIĞI	SM0.0 ... SM179.7	SM0.0 ... SM299.7	SM0.0 ... SM549.7

Tablo 2.8

Özel hafıza bitlerinin her birinin özel görevi bulunduğundan her bir bitin görevinin tek tek açıklanması çok uzun ve karmaşık olacağından konuların anlatımında temel olarak bize yardımcı olacak bazı bitleri açıklamak yerinde olacaktır.

- Ø **SM0.0:** Bu bit daima lojik 1 değerindedir. Dolayısıyla ladder yöntemiyle yazılan programlarda SM0.0 bitinin atandığı kontak daima enerji akışına izin verir.
- Ø **SM0.1:** Bu bit ilk taramada lojik 1, ikinci taramada lojik 0 değerini alır. Sayıcıların ve kalıcı tip zaman rölelerinin reset edilmesinde kullanılabilir.
- Ø **SM0.3:** Enerji verildikten sonra ilk taramada 1, sonraki taramalarda 0 değerini alır.
- Ø **SM0.4:** 60 saniye periyodlu flaşördür. Bu bit periyodik olarak 30 saniye lojik 1, 30 saniye lojik 0 değerini alır.
- Ø **SM0.5:** 1 saniye periyodlu flaşördür. Bu bit periyodik olarak 0,5 saniye lojik 1, 30 saniye lojik 0 değerini alır.
- Ø **SM0.6:** Bu bit periyodik olarak bir taramada 1, diğerinde lojik 0 değerini alır.

Diğer özel hafıza (SM) bitlerinin üstlendikleri görevleri öğrenmek için Simatic **S7 – 200 kullanma klavuzuna bakabilirsiniz.**

SM hafıza alanlarına Bayt, Word ve Double word uzunluğunda erişilebilir.

#### 2.7.12. Zamanlayıcılar

PLC bünyesinde bulunan zamanlayıcılar klasik kumanda devrelerinde kullanılan zaman röleleri gibi görev yaparlar. Klasik zaman rölelerinin çalışma biçiminde farklı olarak PLC bünyesinde bir de toplamalı tip zamanlayıcı bulunur.

S7 – 200 PLC’lerde çalışma şekli bakımından üç tip zamanlayıcı bulunmaktadır.

- TON** : Çekmede gecikmeli ( düz)zamanlayıcı
- TOF** : Düşmede gecikmeli (ters) zamanlayıcı
- TONR:** Çekmede gecikmeli kalıcı tip (toplamalı tip) zamanlayıcı

PLC’lerde bulunan zamanlayıcılar belirli bir zaman aralıklarını sayarlar. Sayılan zaman aralığının değerine zaman tabanı denir. Zamanlayıcılarda sayılan zaman tabanı bakımından bir grupta daha yapılır. S7 – 200 PLC’lerde 1ms, 10ms ve 100 ms zaman tabanlı üç tip zamanlayıcı bulunmaktadır.

Bir zamanlayıcının aynı adresi taşıyan iki değişkeni bulunur.

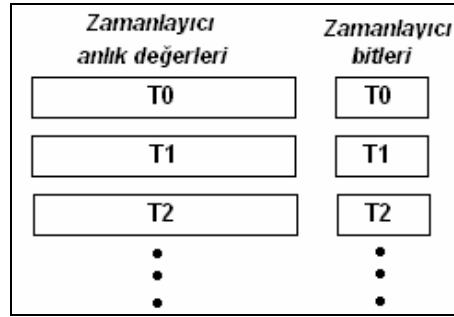
- Ø **Anlık değer:** Zamanlayıcı tarafından sayılmış olan süreyi gösterir. Bu değer 16 bit (word) uzunluğundaki bir zamanlayıcı hafıza adresinde saklanır ve bu değere sadece word uzunluğunda bir operant ile erişilebilir.



Ø **Zamanlayıcı Biti:** Zamanlayıcı Anlık değeri ile ayar değerinin karşılaştırma işlemi sonucunda zamanlayıcı bit adresi 1 ya da 0 olur. Ayar değeri zamanlayıcı komutunun bir parçası olarak girilir.


PLC programlarında zamanlayıcının anlık değerine mi, yoksa zamanlayıcı bitine mi ulaşılacağı kullanılan komutun kullandığı operant tipine bağlıdır.

Bir zamanlayıcı adresine erişmek için hafıza alanı belirteci olan T (Timer) harfinden sorma zamanlayıcı numarası yazılır. Yazılacak zamanlayıcı numarası kullanılacak PLC'nin tipine göre değişir. Yazılabilecek maksimum zamanlayıcı numarası kullanılan PLC de bulunan zaman rölesi sayısıdır. Bir zamanlayıcı numarası aynı program içinde birden fazla zamanlayıcı için kullanılmamalıdır.



Şekil 2.57: Zamanlayıcı adresleri

Ladder diyagramlarında zamanlayıcı bit adresleri normalde açık ya da normalde kapalı bir kontağa atanır. Zamanlayıcı anlık değeri ile ayar değeri birbirine eşit olduğunda zamanlayıcı biti 1 olur ve atandığı kontakın açılmasını veya kapanmasını sağlar.

T37  
 T37 zamanlayıcı biti 1 olduğunda kontak kapanarak enerji akışı sağlar.

Tablo 2.9 da S7 - 200 CPU 221 ... 226M tipi PLC'lerin çalışma şekli ve zaman tabanına göre zamanlayıcı adresleri verilmiştir.

ZAMANLAYICI TİPİ	ZAMAN TABANI	ZAMANLAYICI NUMARASI
TON VE TOF	1 ms	T32 ve T96 ( 2 adet)
	10 ms	T33 ...T36 ve T97...T100 ( 8 adet)
	100ms	T37 ...T63 ve T101...T255 ( 181 adet)
TONR	1 ms	T0 ve T64 ( 2 adet)
	10 ms	T1...T4 ve T65...T68 (8 adet)
	100ms	T5...T31 ve T69...T95 ( 54 adet)

Tablo 2.9

### 2.7.13. Sayıcılar

Endüstride, üretilen ürün sayılarının belirlenmesi ya da bir işlemin tekrar sayısının tespitinde sayıcılar kullanılır. Sayıcılar sayma girişlerine gelen sinyalin yükselen kenarlarını sayarlar.

S7 – 200 PLC’lerde çalışma şekli bakımından üç tip sayıcı bulunmaktadır.

**CTU:** Yukarı (Up) sayıcı

**CTD:** Aşağı (Down) sayıcı

**CTUD:** Aşağı /Yukarı (Up / Down) sayıcı

Bir sayıcının aynı adresi taşıyan iki değişkeni bulunur.

- Ø **Anlık Değer:** Sayıcı tarafından sayılmış değeri gösterir. Bu değer 16 bit (word) uzunluğundaki bir sayıcı hafıza adresinde saklanır ve bu değere sadece word uzunluğunda bir operant ile erişilebilir.
- Ø **Sayıcı Biti:** Sayıcı anlık değeri ile sayıcı ayar değerinin karşılaştırma işlemi sonucunda sayıcı bit adresi 1 ya da 0 olur. Ayar değeri sayıcı komutunun bir parçası olarak girilir.

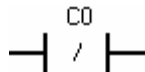
PLC programlarında sayıcının anlık değerine mi, yoksa sayıcı bitine mi ulaşılacağı kullanılan komuta bağlıdır.

Bir sayıcı adresine erişmek için hafıza alanı belirteci olan C (Counter) harfinden sonra sayıcı numarası yazılır. Yazılabilecek en büyük sayıcı numarası kullanılan PLC nin tipine göre değişir. Bir sayıcı numarası üç sayıcı tipinden herhangi biri için kullanılabilir. Ancak bir sayıcı numarası aynı program içinde birden fazla sayıcı için kullanılmamalıdır.

Sayıcı anlık değerleri			Sayıcı bitleri
15(MSB)	C0	0 (LSB)	C0
	C1		C1
	C2		C2
	⋮		⋮
	⋮		⋮
	⋮		⋮

Şekil 2.58: Sayıcı adresleri

Ladder diyagramlarında sayıcı bit adresleri normalde açık ya da normalde kapalı bir kontağa atanır. Sayıcı anlık değeri ile ayar değeri birbirine eşit olduğunda zamanlayıcı biti 1 olur ve atandığı kontağın açılmasını veya kapanmasını sağlar.

 C0 zamanlayıcı biti 1 olduğunda kontak açılarak enerji akışını keser.

Aşağıdaki tabloda S7 - 200 CPU 212, 214,215...226M tipi PLC'lerin sayıcı adres aralıkları görülmektedir.

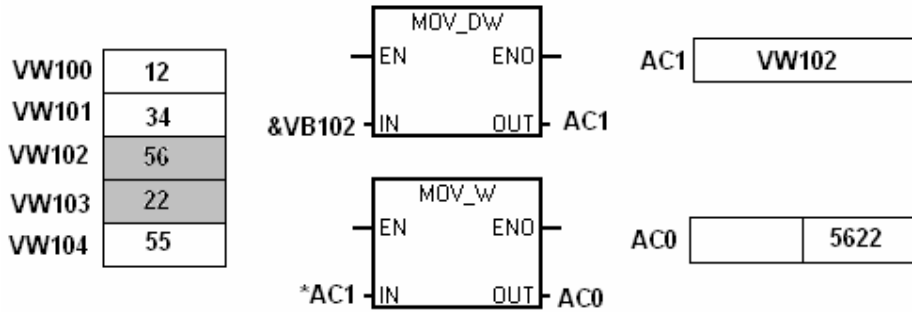
CPU TİPİ	SAYICI ADRESLERİ
CPU 212	C0.....C63 ( 64 adet)
CPU 214	C0.....C127 (128 adet)
CPU 215 ...226	C0.....C255 (256 adet)

**Tablo 2.10**

#### 2.7.14. Endirekt (dolaylı) Adresleme

Endirekt adreslemede bir hafıza alanındaki veriye erişmek için pointer (adres göstericisi) kullanılır. Pointer olarak kullanılan hafıza alanının içeriğinde bir veri değil başka bir alanın adresi bulunur. S7 – 200 PLC'lerde yalnızca V ve L hafıza alanları ile AC1, AC2 ve AC3 akümülatör alanları pointer olarak kullanılabilir. Pointer alanı double word uzunluğunda tanımlanır. Pointer kullanarak I,Q,V,M,S,T ve C hafıza alanlarına byte, word ve double word uzunluğunda erişilebilir. Tek tek bitlere ve AI, A, Q, HC, SM ve L hafıza alanlarına pointer kullanarak erişilenez.

Bir pointer oluşturmak için Şekil 2.59'da görüldüğü gibi MOV\_DW komutuyla endirekt olarak adreslenecek alanın adresini & işareti ile pointer alanına taşımanız gerekir. Bir pointerın gösterdiği adresin içeriğine erişmek için \* karakteri ile beraber pointer adresi yazılır.



**Şekil 2.59: Endirekt adresleme**

## 2.8. Temel Programlama Komutları

### 2.8.1. Start, Stop ve Çıkışlar

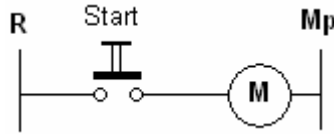
Klasik kumanda devrelerinde sistemin çalışmasını başlatmak ve durdurmak için kullanılan start ve stop butonları ile kumanda akışını yönlendiren sensör, sınır anahtarı vb. elemanlar PLC cihazının girişlerine bağlanır. Klasik kumanda devrelerinde motor vb. alıcılara yol vermek için kullanılan kontaktörler ise PLC çıkışlarına bağlanır. Klasik

kumanda sistemlerinde rölelerle birlikte karmaşık devre bağlantıları ile gerçekleştirilen işlemler PLC ile program yazılarak kolayca gerçekleştirilebilir.

Bu bölümde klasik kumanda devreleri ile Ladder dili kullanılan PLC programları arasındaki ilişki anlatılacaktır.

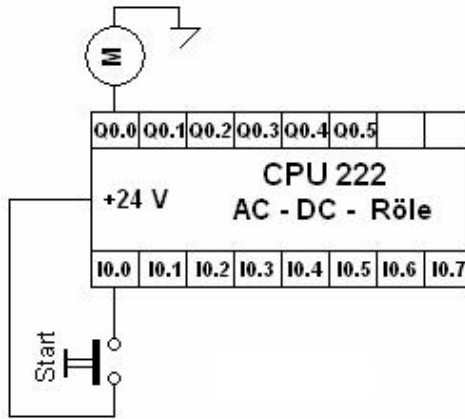
### Ø Start Butonu ile Bir Motorun Kesik Çalıştırılması

Şekil 2.60'daki klasik kumanda devresinde kullanılan start butonu ani temaslıdır. Butona basıldığında M kontaktörü enerjilenir ve yol verdiği motor çalışır. Butondan elimizi çektiğimizde M kontaktörünün enerjisi kesilir ve yol verdiği motorun çalışması durur.



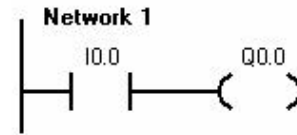
Şekil 2.60

Şekil 2.60'taki klasik kumanda devresini PLC ile gerçekleştirmek için start butonunun PLC girişlerinden herhangi birine, M kontaktörünün ise PLC çıkışlarından herhangi birine bağlanması gerekir. Burada kullanılan PLC CPU 222 AC –DC – Röle modeli olup çıkışlarından 220 V altında 2A'e kadar akım çekilebilmektedir. Bu işlemden sonra yapılması gereken PLC programı ile Start butonunun bağlı olduğu giriş adresi ile M kontaktörünün bağlı olduğu çıkış adresini istenilen şartlara göre ilişkilendirmektir.



Şekil 2.61

LADDER:



STL :

Network 1

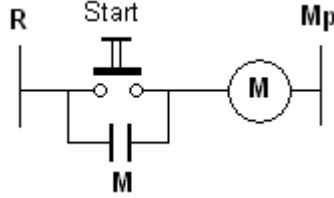
LD I0.0  
= Q0.0

Şekil 2.62

Şekil 2.62'de verilen PLC programında I0.0 girişine bağlı start butonuna basıldığında I0.0 giriş adresi lojik 1 değerini alır. I0.0 giriş adresi ile adreslenen normalde açık kontak kapanarak Q0.0 çıkış adres bitinin lojik 1 değerini almasını sağlar. Q0.0 çıkışı aktif olur ve M kontaktörü enerjilenerek yol verdiği motor çalışmaya başlar. Start butonundan elimizi çektiğimizde I0.0 giriş adresi lojik 0 değerini alır ve bu bitle adreslenen kontak açılır. Q0.0 çıkış adresi lojik 0 değerini alır. Bu durumda M kontaktörünün enerjisi kesilerek yol verdiği motorun çalışması durur.

### Ø Start butonu ile bir motorun sürekli çalıştırılması

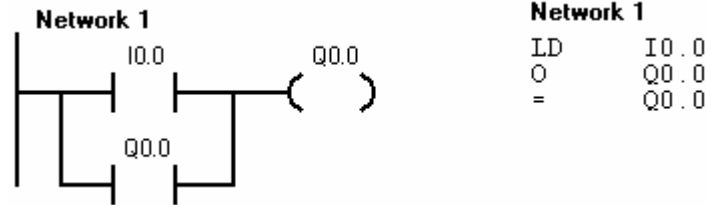
Şekil 2.63'teki klasik kumanda devresinde start butonuna basıldığında M kontaktörü enerjilenerek kontaklarını konum değiştirir. Start butonundan elimizi çektiğimizde M kontaktörü, kapanmış olan kontağı üzerinden enerjili kalmaya devam eder ve yol verdiği motor sürekli çalışır. Bu olaya mühürleme adı verilir.



Şekil 2.63

Şekil 2.63'teki klasik kumanda devresini PLC ile gerçekleştirmek için gerekli bağlantı Şekil 2.61 ile aynıdır. Burada mühürleme işlemi program tarafından gerçekleştirilir. Şekil 2.64'de görülen PLC programında I0.0 girişine bağlı start butonuna basıldığında bu giriş ile adreslenen kontak kapanarak Q0.0 çıkışını enerjilendirir. Q0.0 çıkış biti lojik 1 değerini alarak bu bitle adreslenen kontak kapanır ve mühürleme işlemini gerçekleştirir. Butondan elimizi çeksek bile çıkış enerjili kalmaya devam eder. PLC programlamada mühürleme işlemi, daha sonraki bölümlerde anlatılacak olan SET fonksiyonu ile de gerçekleştirilebilir.

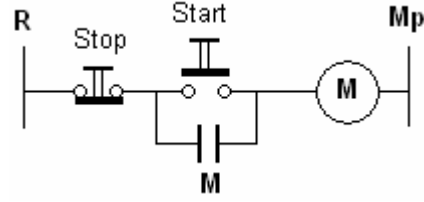
Bu devrede mühürleme işlemi ile çıkış sürekli enerjili kalır ve çıkışa bağlı alıcı durdurulamaz. Bu nedenle sistemi durdurmak için PLC girişine stop butonu bağlayarak programla ilişkilendirmemiz gerekir.



Şekil 2.64

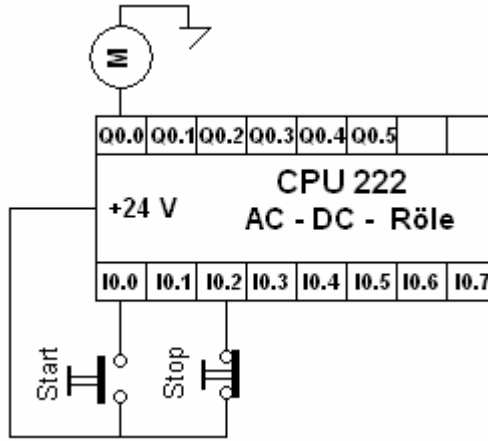
### Ø Start butonu ile sürekli çalıştırılan motorun stop butonu ile durdurulması

Şekil 2.65'teki klasik kumanda devresinde start butonuna basıldığında M kontaktörü enerjilenerek kendisini mühürler. Start butonundan elimizi çeksek dahi M kontaktörü enerjili kalmaya devam eder. Stop butonuna basıldığında M kontaktörünün enerjisi kesilerek sistemin çalışması sona erer.



Şekil 2.65

Şekil 2.65'teki devreyi PLC ile gerçekleştirmek için Şekil 2.66'da PLC'nin I0.2 girişine bir stop butonu bağlanmıştır. Sistemi durdurmak için PLC programında Şekil 2.67'deki gibi normalde kapalı bir kontak kullanırsak önemli bir hata yapmış oluruz. Çünkü I0.2 girişine bağlı stop butonu normalde kapalı pozisyonda olduğundan bağlı olduğu girişi sürekli enerji altında tutar. I0.2 giriş biti sürekli lojik 1 konumda olacağından ladder programındaki I0.2 giriş biti ile adreslenmiş normalde kapalı kontak konum değiştirilerek sürekli açık durumda kalacaktır. Bu durumda start butonu ile sistem çalıştırılmaz.



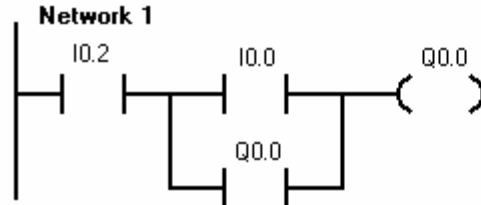
Şekil 2.66

Network 1

```
LDN    I0.2
LD      I0.0
O       Q0.0
ALD
=       Q0.0
```

Şekil 2.67

Bu sakıncayı önlemek için sistemi durdurmak amacıyla kullanılan I0.2 kontağının Şekil 2.68'deki gibi normalde açık olması gerekir. Eğer sistemi durdurmak için kullanılan PLC girişinin sürekli enerji altında kalmasını istemiyorsak durdurma işlemini start butonu kullanarak yapabiliriz.



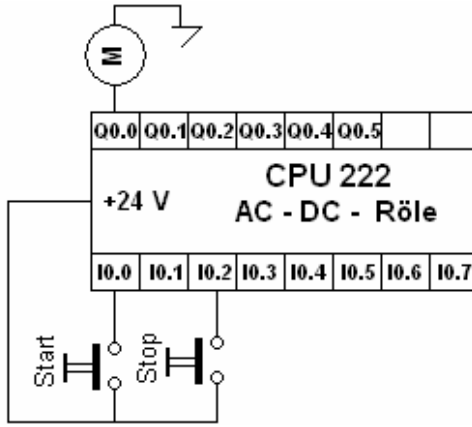
Şekil: 2.68

Network 1

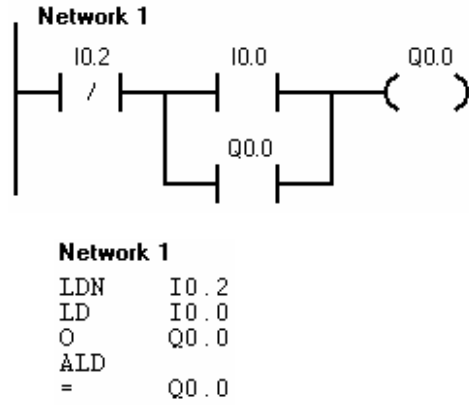
```
LD      I0.2
LD      I0.0
O       Q0.0
ALD
=       Q0.0
```

### 2.8.1.1. Start Butonu ile Durdurma

Şekil 2.68'deki PLC bağlantısında start butonuna basıldığında M kontaktörü enerjilenerek yol verdiği motorun sürekli çalışmasını sağlamaktadır. Sistemi durdurmak için I0.2 girişine normalde açık bir stop butonu bağlanmıştır. Sistemin çalışmasını sağlayacak PLC programını şekildeki gibi tasarlayabiliriz.



Şekil 2.69



Şekil 2.70

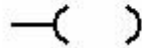
Şekil 2.70'teki PLC programında I0.0 girişine bağlı start butonuna basıldığında Q0.0 çıkışı enerjilenerek kendisini mühürler. I0.2 girişine bağlı normalde açık pozisyondaki stop butonuna basıldığında I0.2 kontağı açılarak sistemin çalışması durur. Bu tip durdurma tam emniyetli değildir. Çünkü stop butonunun I0.2 girişi ile olan bağlantısında bir kopukluk veya temassızlık oluşması durumunda sistem durdurulamaz.

### 2.8.2. Çıkış Atama ( = ) Komutu

PLC programlarında mantıksal sinyal akışının nihai sonucunu bir adres bitine aktarmak için çıkış komutu kullanılır. Sinyalin aktarıldığı bit lojik 1 değerini alır. Eğer sinyal akışı bir çıkış bitini lojik 1 yaparsa bu çıkışa bağlı alıcı enerjilenir. Sinyal akışı kesildiğinde çıkış ataması yapılan bit lojik 0 değerine döner.

LADDER:

Bit adresi

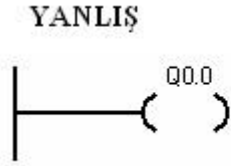


STL :

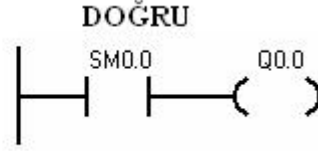
= Bit adresi

ÇIKIŞ KOMUTU OPERANTLARI: I, Q, M, SM, S, T, C, V, L hafıza alan bitleri

Ladder yöntemiyle yazılan PLC programlarında bir çıkış doğrudan enerji kaynağına bağlanamaz. Diğer bir ifade ile program bir çıkış atama komutu ile başlayamaz. Çıkıştan önce en az bir tane normalde açık ya da normalde kapalı kontak bulunmalıdır.



Şekil 2.71

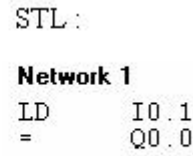
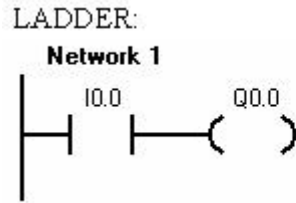


Şekil 2.72

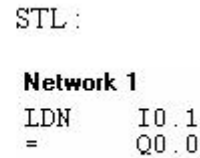
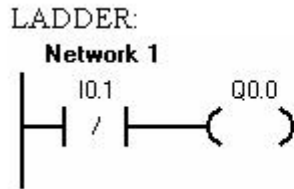
### 2.8.3. LD ve LDN Kontak Yükleme Komutları

- Ø **LD Komutu:** Normalde açık bir kontak ile hat ( komut bloğu) başlatılmasını sağlar. Kontakın üst kısmına operant yazılır. Operant herhangi bir hafıza alan biti olabilir. Yükleme komutu ile başlayıp bir sonraki yükleme komutuna kadar olan kesim, mantıksal olarak birbirine bağlı komut bloğudur.

OPERANTLAR: I, Q, M, SM, S, T, C, V, L hafıza alan bitleri



- Ø **LDN Komutu:** Normalde kapalı bir kontak ile hat başlatılmasını sağlar. Kontakın üst kısmına operant yazılır. Operant herhangi bir hafıza alan biti olabilir.

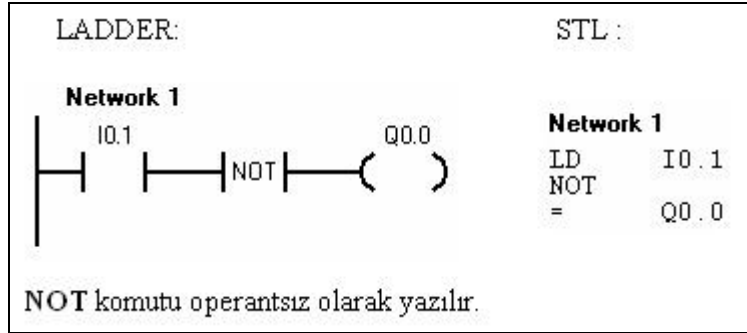


LD VE LDN KOMUTU OPERANTLARI: I, Q, M, SM, S, T, C, V, L hafıza alan bitleri

### 2.8.4. NOT Değilleme

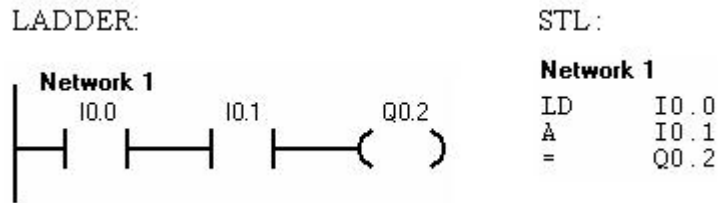
**NOT** komutu ladder diyagramlarında NOT kontağı ile gösterilir. NOT kontağı enerji akışını tersine çevirir. Girişinde enerji varsa çıkışında enerji olmaz, eğer girişinde enerji yoksa çıkışına enerji sağlar. Yapılan işlem lojik değil işlemidir. NOT kontağı enerji hattına doğrudan bağlanamaz. Aşağıda örnekte NOT kontağının kullanım şekli ve STL dilindeki kullanımı görülmektedir.



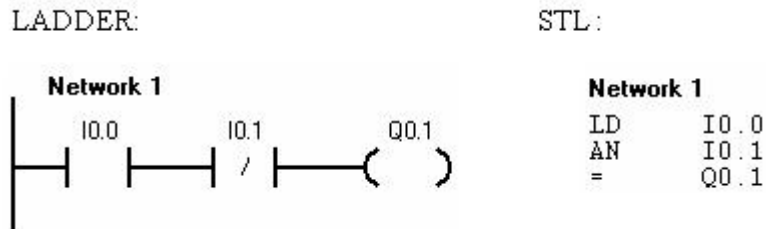


### 2.8.5. A (VE), AN (VE DEĞİL) Komutları ve Uygulaması

A Komutu ladder mantığına göre kendisinden önce gelen devre kesimine normalde açık bir kontağı seri bağlar.



AN Komutu ladder mantığına göre kendisinden önce gelen devre kesimine normalde kapalı bir kontağı seri bağlar.

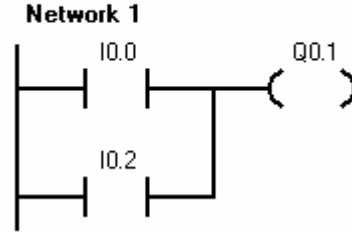


**A ve AN KOMUTLARI OPERANTLARI: I, Q, M, SM, S, T, C, V, L hafıza alan bitleri**

### 2.8.6. O (VEYA), ON (VEYA DEĞİL) Komutları ve Uygulaması

O Komutu ladder mantığına göre kendisinden önce gelen devre kesimine normalde açık bir kontağı paralel bağlar.

LADDER:

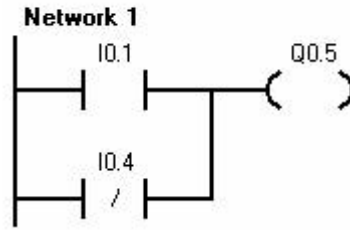


STL:

**Network 1**  
LD I0.0  
O I0.2  
= Q0.1

**ON** Komutu ladder mantığına göre kendisinden önce gelen devre kesimine normalde kapalı bir kontağı paralel bağlar.

LADDER:



STL:

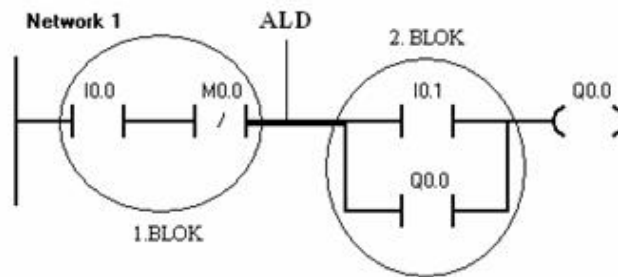
**Network 1**  
LD I0.1  
ON I0.4  
= Q0.5

**O ve ON KOMUTLARI OPERANTLARI: I, Q, M, SM, S, T, C, V, L hafıza alan bitleri**

### 2.8.7. Birden Fazla Komut Bloğu Arasında (ALD ) VE İşlemi ve Uygulaması

Ladder mantığına göre birden fazla kontağın kendi arasında seri veya paralel bağlanması ile oluşan gruba blok adı verilir. STL mantığına göre blok bir kontak yükleme (LD veya LDN) komutuyla başlayan ve bir sonraki kontak yükleme komutuna kadar olan kesimdir. Birden fazla komut bloğunu birbirine seri bağlamak için (VE işlemi) ALD komutu kullanılır.

LADDER:



STL:

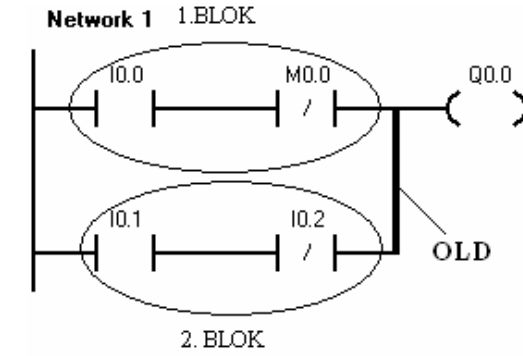
**Network 1**  
LD I0.1  
AN M0.0  
LD I0.1  
O Q0.0  
ALD  
= Q0.0

**ALD** komutu operantsız olarak yazılır.

## 2.8.8. Birden Fazla Komut Bloğu Arasında (OLD) VEYA İşlemi ve Uygulaması

Birden fazla komut bloğunu birbirine paralel bağlamak için OLD komutu kullanılır.

LADDER:

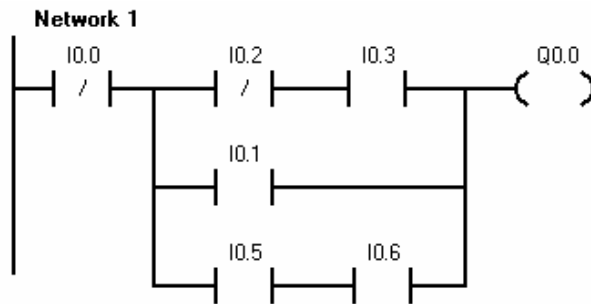


STL:

**Network 1**  
LD I0.0  
AN M0.0  
LD I0.1  
A I0.2  
OLD  
= Q0.0

OLD komutu operantsız olarak yazılır.

**UYGULAMA 1:** Şekilde ladder diyagramı verilen PLC programını STL dili ile yazınız?



**Çözüm:**

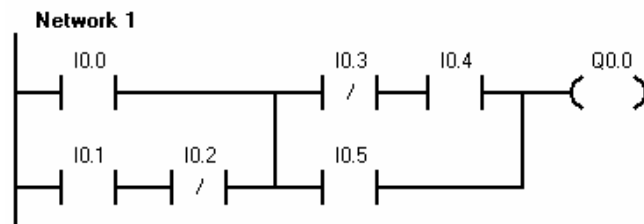
**Network 1**  
LDN I0.0  
LDN I0.2  
A I0.3  
O I0.1  
LD I0.5  
A I0.6  
OLD  
ALD  
= Q0.0

**UYGULAMA2:** Verilen STL dili programının Ladder diyagramını çiziniz?

STL:

**Network 1**  
LD I0.0  
LD I0.1  
AN I0.2  
OLD  
LDN I0.3  
A I0.4  
O I0.5  
ALD  
= Q0.0

**Çözüm:**

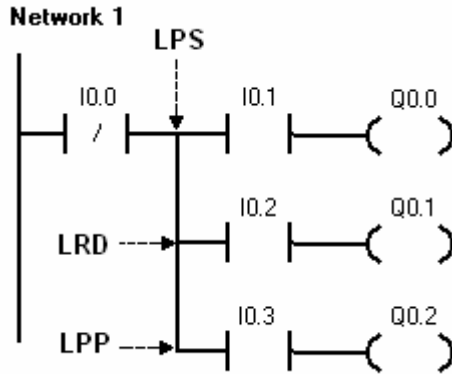


### 2.8.9. LPS, LPP, LRD ( Lojik yığın ) Komutları

- Ø **LPS Komutu:** Ara kol bağlantısı oluşturmak için kullanılır.
- Ø **LRD Komutu:** Bir çıkışa arakol oluşturmak için kullanılır.
- Ø **LPP Komutu:** En son çıkış kolu oluşturmak için kullanılır.

#### ÖRNEK:

LADDER:

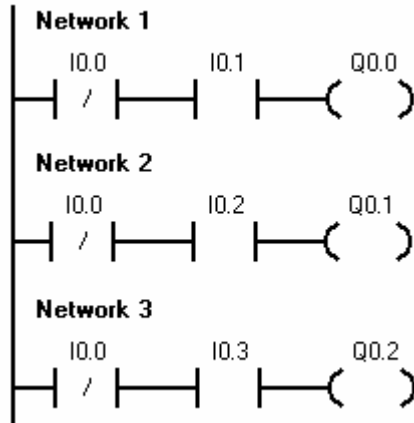


STL:

```
Network 1
LDN    I0.0
LPS
A      I0.1
=      Q0.0
LRD
A      I0.2
=      Q0.1
LPP
A      I0.3
=      Q0.2
```

Kolların farklı network'lere yerleştirilmesi durumunda lojik yığın komutlarına ihtiyaç kalmaz.

LADDER:



STL:

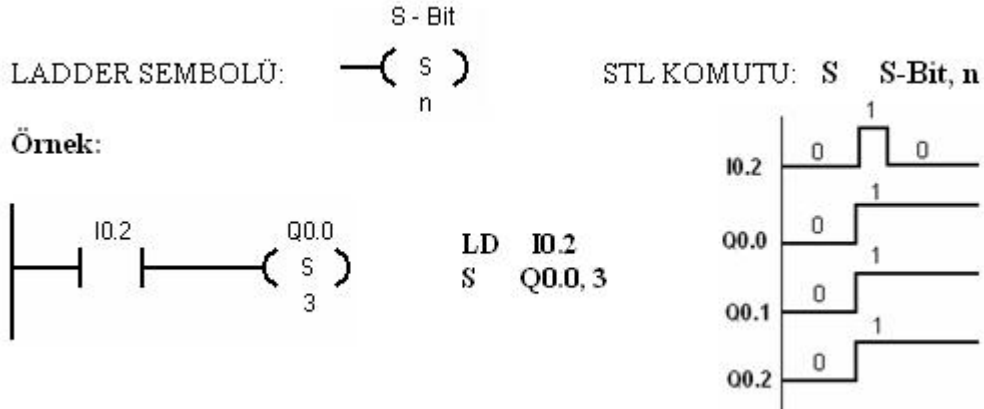
```
Network 1
LDN    I0.0
A      I0.1
=      Q0.0

Network 2
LDN    I0.0
A      I0.2
=      Q0.1

Network 3
LDN    I0.0
A      I0.3
=      Q0.2
```

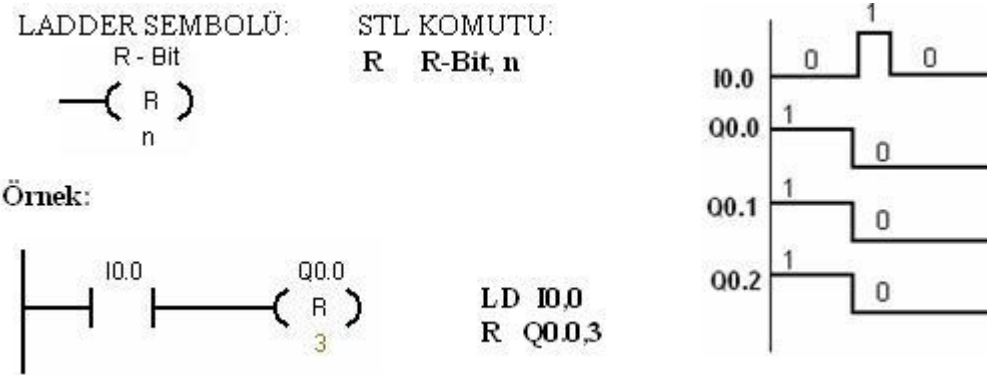
### 2.8.10. SET, RESET Komutları ve Uygulaması

- Ø **SET KOMUTU:** Bir hafıza alanının istenilen bitinden (S-Bit) itibaren n adet biti set (lojik 1) yapmak için kullanılır. Set yapılan çıkışlar hafızaya alınır ve reset yapılana kadar aktif durumda kalır.



Verilen ladder devresinde I0.2 girişi aktif (lojik1) olduğunda kontak kapanarak SET fonksiyonunu çalıştırır. SET fonksiyonu Q0.0 dan itibaren 3 biti, bir başka ifade ile Q0.0, Q0.1 ve Q0.2 çıkış bitlerini aktif yapar. PLC'nin bu çıkışlarına bağlı alıcılar çalışır. I0.2 girişi lojik 0 olduğunda kontak açılır. Set yapılan çıkışlar hafızaya alındığından aktif durumda kalmaya devam eder. PLC programlamada klasik mühürleme tekniği yerine kullanılır.

Ø **RESET KOMUTU:** Bir hafıza alanının istenilen bitinden (S-Bit) itibaren n adet biti reset (lojik 0) yapmak için kullanılır.



Verilen ladder devresinde I0.0 girişi aktif (lojik 1) yapıldığında kontak kapanarak Reset fonksiyonunu çalıştırır. Reset fonksiyonu Q0.0 dan itibaren 3 biti, bir başka ifade ile Q0.0, Q0.1 ve Q0.2 çıkış bitlerini pasif (lojik 0) yapar. PLC'nin bu çıkışlarına bağlı alıcıların çalışması durur. Reset komutunun etkisini görmek için bitlerin set yapılmış olması gerekir.

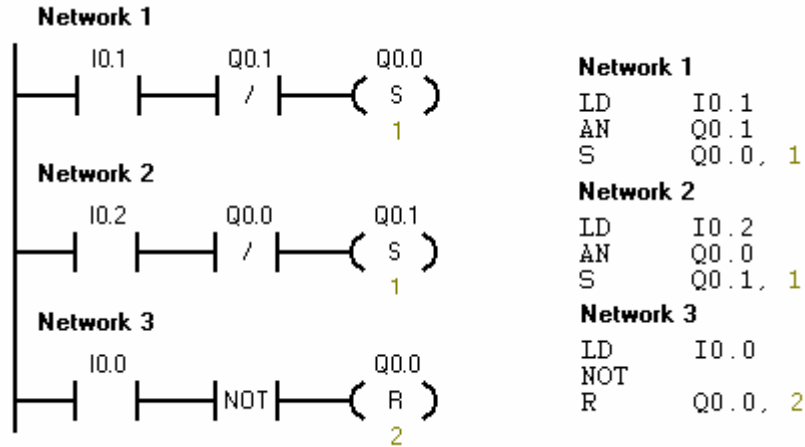
SET VE RESET KOMUTLARI OPERANTLARI:

S-Bit, R-Bit : **I,Q,M, SM,S,T,C,V,L hafıza alan bitleri**

n : **IB, QB, VB, MB, SMB, SB, LB, AC, \* VD, \* LD, \* AC, Sabit ( 0 ... 255)**

**UYGULAMA:** Üç fazlı bir asenkron motor ileri geri çalıştırılacaktır. IO.1 girişine bağlı start butonuna basıldığında motor ileri yönde, IO.2 girişine bağlı start butonuna basıldığında motor geri yönde çalışacaktır. Motor herhangi bir yönde çalışırken diğer yönde çalıştırılamayacaktır. IO.0 girişine bağlı stop butonuna basıldığında motor duracaktır.

**Çözüm:**

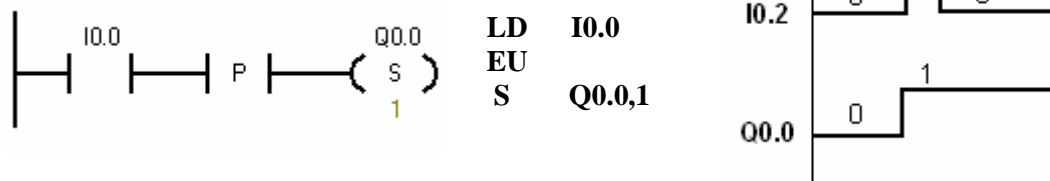


### 2.8.11. Giriş Sinyalinin Çıkan Kenarına Göre Çıkış Alma

Yükselen kenar ( Pozitif kenar) algılama kontağı girişine uygulanan sinyalin 0'dan 1 konumuna her geçişinde sadece bir tarama için enerji akışına izin verir. Pozitif kenar algılama kontağının STL eşdeğeri EU komutudur.

Ladder Sembolü: STL Komutu: **EU**

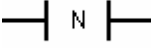
**Örnek:**



Verilen Ladder devresinde IO.0 girişi ani temaslı bir butonla aktif yapıldığında kontak kapanarak enerji akışı sağlar. Ancak grafikten anlaşılabileceği gibi pozitif kenar algılama kontağı IO.0 kontağından gelen sinyalin sadece yükselen kenarında enerji akışına izin vererek Q0.0 çıkış bitinin set yapılmasını sağlar.

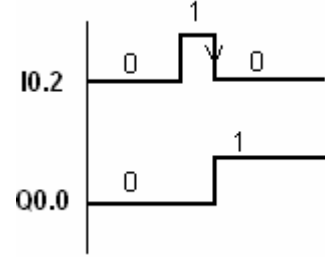
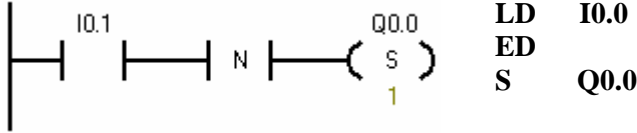
## 2.8.12. Giriş Sinyalinin İnen Kenarına Göre Çıkış Alma ve Uygulaması

Düşen kenar ( Negatif kenar) algılama kontağı girişine uygulanan sinyalin 1'den 0 konumuna her geçişinde sadece bir tarama için enerji akışına izin verir. Negatif kenar algılama kontağının STL eşdeğeri ED komutudur.

Ladder Sembolü: 

STL Komutu: **ED**

**Örnek:**

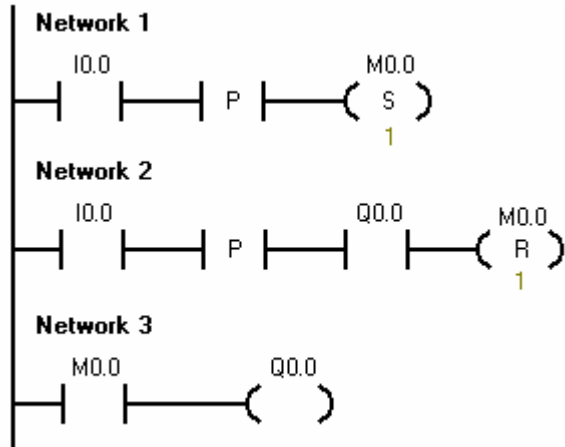


Şekildeki ladder devresinde I0.1 girişi ani temaslı bir butonla aktif yapıldığında kontak kapanarak enerji akışı sağlar. Ancak grafikten anlaşılabileceği gibi negatif kenar algılama kontağı I0.1 kontağından gelen sinyalin sadece düşen kenarında enerji akışına izin vereceğinden çıkış enerjilenmez. Butondan elimizi çektiğimiz anda negatif kenar algılama kontağı girişine uygulanan sinyalin düşen kenarında enerji akışına izin vererek Q0.0 çıkış bitinin set yapılmasını sağlar.

Pozitif ve Negatif kenar algılama komutları operant kullanmaz.

**UYGULAMA:** Q0.0 çıkışı ile kontrol edilen bir motor I0.0 girişine bağlı ani temaslı bir start butonu ile hem çalıştırılıp hem de durdurulacaktır.

**Çözüm:**



**Network 1**  
**LD I0.0**  
**EU**  
**S M0.0, 1**

**Network 2**  
**LD I0.0**  
**EU**  
**A Q0.0**  
**R M0.0, 1**

**Network 3**  
**LD M0.0**  
**= Q0.0**

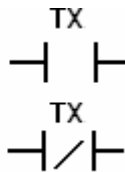
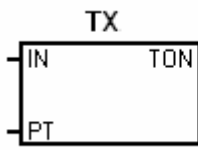
### 2.8.13. Zamanlayıcı (Timer) Komutu

PLC programlarında bir giriş sinyaline karşılık zamana bağımlı çıkış sinyali üretmek için zamanlayıcılar kullanılır. S7 – 200 PLC’lerde çalışma şekli bakımından 3 tip zamanlayıcı bulunmaktadır.

#### Ø Çekmede Geçikmeli Zamanlayıcı ( TON)

IN girişine enerji geldikten sonra PT girişi ile belirlenen zaman sonunda, zamanlayıcı biti ile adreslenen kontakları konum değiştiren zamanlayıcılardır. Klasik kumanda devrelerindeki düz zaman rölesi gibi çalışır.

LADDER:



STL:

TON TX,PT

Şekilde görülen TON tipi zamanlayıcıda;

- **IN:** Zamanlayıcı girişidir. IN girişine gelen sinyal 1 olduğunda zamanlayıcı zamanı saymaya başlar. Bu giriş doğrudan enerji hattına bağlanamaz. Zamanlayıcıyı başlatmak için mutlaka bir kontak kullanılmalıdır.
- **PT(Preset Time):** Zamanlayıcı ayar değeridir ve kullanıcı tarafından belirlenir. PT değeri desimal olarak 0....32767 aralığında herhangi bir değer olabilir. PT değerini belirlemek için bir hafıza alan adresi kullanılabilir.
- **TX:** Zamanlayıcı adresini belirtir. Zamanlayıcı tarafından sayılan süre anlık olarak bu adrese yazılır. Aynı adresi kullanan bir de zamanlayıcı biti bulunmaktadır. IN girişinin enerjilenmesi ile birlikte zamanlayıcı zaman sayma işlemini başlatır. TX adresine kaydedilen zamanlayıcı anlık değeri ile PT ayar değeri birbirine eşit olduğu anda zamanlayıcı biti ( TX ) ile adreslenen kontaklar konum değiştirir. Zaman sayma işlemi 32767’ye kadar devam eder. Eğer zamanlayıcının IN girişine gelen sinyal 0 olursa sayma değeri sıfırlanır.

#### Ø Çekmede Gecikmeli Kalıcı Tip ( toplamalı tip) Zamanlayıcı (TONR)

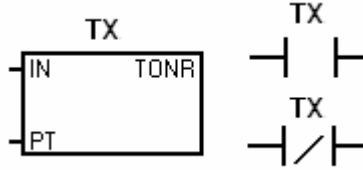
TONR tipi zamanlayıcı TON tipi zamanlayıcıdan farklı olarak enerjili kaldığı sürelerin toplamını sayar. Zamanlayıcıda IN girişi enerjilendiğinde zaman sayma işlemi başlar. Eğer PT değerine ulaşılmadan IN girişine gelen enerji kesilirse zaman rölesi anlık değeri zaman rölesi adresinde saklanır. Böylece enerji tekrar uygulandığında zamanlayıcı



sayma işlemine kaldığı yerden devam eder. Herhangi bir anda TX adresindeki zamanlayıcı anlık değeri ile PT değeri eşit olduğunda zamanlayıcı biti ile adreslenen kontaklar konum değiştirir.

TONR tipi zamanlayıcının değerini sıfırlamak için zamanlayıcı bitinin RESET komutuyla sıfırlanması gerekir.

LADDER SEMBOLÜ:



STL KOMUTU:

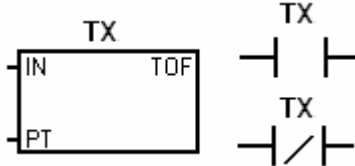
**TONR TX,PT**

### Ø Bırakmada Geçikmeli Zamanlayıcı ( TOF)

TOF tipi zamanlayıcı IN girişi enerjilendiğinde zamanlayıcı biti ile adreslenen kontakları hemen konum değiştirir. IN girişine uygulanan sinyal kesildiği anda zamanlayıcı sayma işlemi başlatır. TX adresine yazılan zamanlayıcı anlık değeri ile PT ayar değeri birbirine eşit olduğunda kontaklar normal konumlarına dönerler. Klasik kumanda devrelerinde kullanılan ters zaman rölesi gibi çalışır.

TOF tipi bırakmada gecikmeli zamanlayıcı her marka ve model PLC’de bulunmayabilir. TOF tipi zamanlayıcı komutu bulunmayan bir PLC’de program hazırlarken ihtiyaç duyulması halinde problem TON tipi zamanlayıcılarla çözülür.

LADDER SEMBOLÜ:



STL KOMUTU:

**TOF TX,PT**

ZAMANLAYICI KOMUTLARI OPERANTLARI:

IN: **I,Q,M, SM,S,T,C,V,L** hafıza alan bitleri ile adreslenen bir kontakta sonra bağlanır.

TX: **T0 ... T255** arasındaki zamanlayıcı adresleri (zamanlayıcı numaraları için sayfa 38’e bakınız)

PT: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C, \* VD, \* LD, \* AC, Sabit ( 0 ... 32767 )**

#### 2.8.14. Harici Olarak Zamanlayıcı Ayar Değerinin Atanması

Bir zamanlayıcı ile elde edilmek istenen zaman gecikme süresi, seçilen zamanlayıcının zaman tabanına ve PT ayar değerine bağlıdır. PT değeri belli bir zaman tabanı süresinden kaç tane sayılacağını gösterir. PT değeri aşağıdaki basit formülle kolayca bulunabilir.

$$PT = \frac{\text{Elde edilecek zaman süresi (ms)}}{\text{Zaman tabanı (ms)}}$$

**Örnek:** T37 adresli zamanlayıcı kullanarak 22 sn zaman süresi elde etmek için PT değeri ne olmalıdır?

22 sn = 22000ms ve T37 adresli zamanlayıcının zaman tabanı 100ms dir

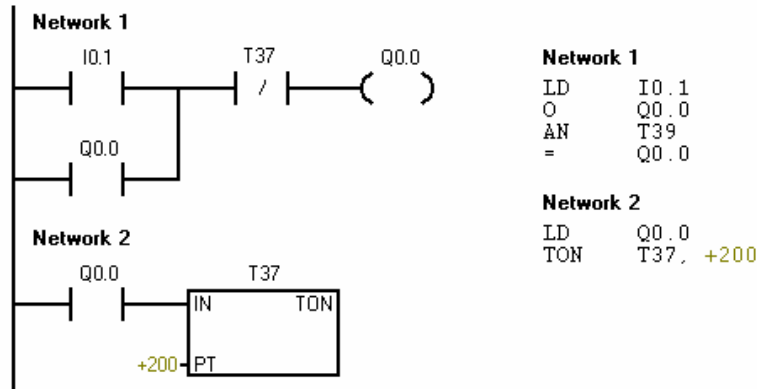
$$PT = \frac{22000 \text{ (ms)}}{100 \text{ (ms)}} = 220$$

PT değeri 220 olarak atanmalıdır.

#### 2.8.15. Zamanlayıcı Uygulamaları

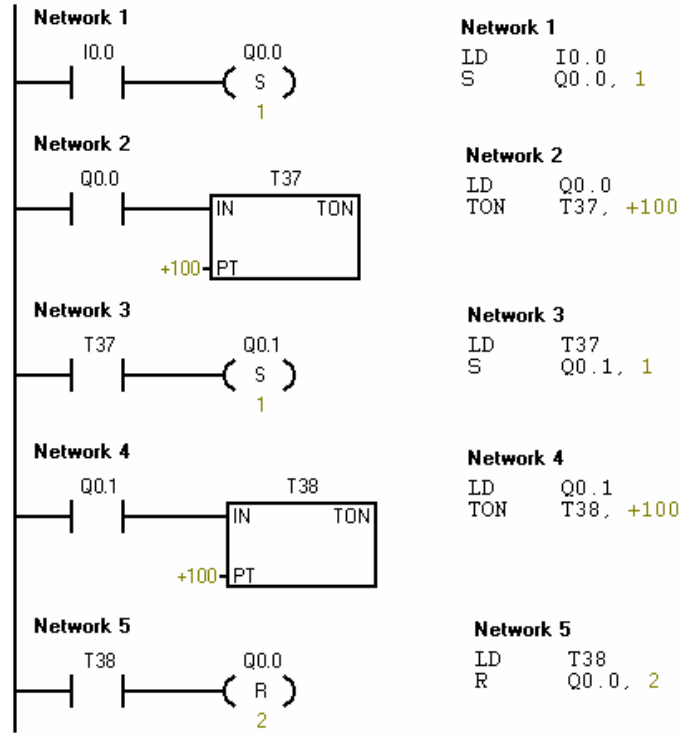
**UYGULAMA 1:** I0.1 girişine bağlı start butonuna basıldığında Q0.0 çıkışına bağlı motor çalışarak 20 sn sonra kendiliğinden duracaktır. Gerekli PLC programını yazınız?

**Çözüm:**



**UYGULAMA 2:** I0.1 girişine bağlı start butonuna basıldığında Q0.0 çıkışına bağlı 1.motor çalışmaya başlayacak, 10 sn sonra Q0.1 çıkışına bağlı 2.motor devreye girecektir. İki motor birlikte 20 sn çalıştıktan sonra her iki motor devreden çıkacaktır. Gerekli PLC programını yazınız?

### Çözüm:



### 2.8.16. Sayıcı (Counter) Komutları

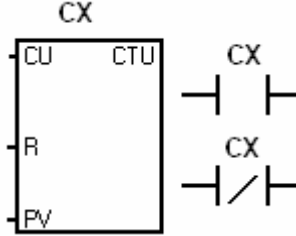
Endüstride, üretilen ürün sayılarının belirlenmesi ya da bir işlemin tekrar sayısının tespitinde sayıcılar kullanılır. Sayıcılar sayma girişlerine gelen sinyalin yükselen kenarlarını sayarlar.

S7 – 200 PLC’lerde çalışma şekli bakımından üç tip sayıcı bulunmaktadır.

#### Ø Yukarı Sayıcı ( CTU )

CTU sayıcıda CU sayma girişine uygulanan sinyalin her yükselen kenarlarında sayıcı anlık değeri 0 dan itibaren bir artar. Sayıcı anlık değeri CX ile belirtilen sayıcı adresinde saklanır. Herhangi bir anda sayıcı anlık değeri, PV sayma değerine eşit ya da büyük olduğunda sayıcı biti (CX) ile adreslenen kontaklar konum değiştirir.

LADDER SEMBOLÜ:



STL KOMUTU:

**CTU CX,PV**

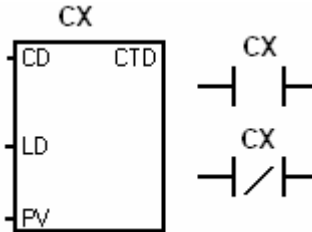
Şekilde görülen CTU tipi sayıcıda;

- **CU ( Count UP):** Yukarı sayma girişidir. CU girişine uygulanan bir yükselen kenar sinyali sayıcı anlık değerini bir arttırır. Bu giriş doğrudan enerji hattına bağlanamaz. Sayma sinyallerini üreten bir kontak ile kullanılır.
- **R (Reset):** Sayıcı resetleme( sıfırlama) girişidir. R girişine 1 uygulandığında CX adresinde bulunan sayıcı anlık değeri ve sayıcı biti sıfırlanır. Konum değiştirmiş olan sayıcı kontakları normal konumlarına döner. CU girişine sinyal uygulanmaya devam ediliyorsa sayıcı sıfırdan itibaren tekrar saymaya başlar.
- **PV (Preset Value):** Sayıcı ayar değeridir ve kullanıcı tarafından belirlenir. PV değeri desimal olarak 0....32767 aralığında herhangi bir değer olabilir. PV değerini belirlemek için bir hafıza alan adresi kullanılabilir.

#### Ø Aşağı sayıcı ( CTD )

CTD sayıcıda CD sayma girişine uygulanan sinyalin her yükselen kenarlarında sayıcı anlık değeri LD girişi ile yüklenen sayma değerinden itibaren bir azalır. Sayıcı anlık değeri CX ile belirtilen sayıcı adresinde saklanır. Herhangi bir anda sayıcı anlık değeri 0 olduğunda sayıcı biti (CX) ile adreslenen kontaklar konum değiştirir.

LADDER SEMBOLÜ:



STL KOMUTU:

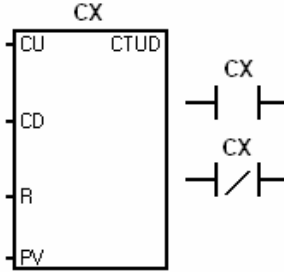
**CTD CX,PV**

- **CTD ( Count DOWN):** Aşağı sayma girişidir. CD girişine uygulanan bir yükselen kenar sinyali sayıcı anlık değerini bir azaltır. Bu giriş doğrudan enerji hattına bağlanamaz. Sayma sinyallerini üreten bir kontak ile kullanılır.
- **LD (Reset):** Yükleme girişidir. Bu girişe uygulanacak bir yükselen kenar sinyali, PV ile belirtilen sayıcı ayar değerini CX ile belirtilen sayıcı adresine yükler.
- **PV (Preset Value):** Sayıcı ayar değeridir ve kullanıcı tarafından belirlenir. PV değeri desimal olarak 0....32767 aralığında herhangi bir değer olabilir. PV değerini belirlemek için bir hafıza alan adresi kullanılabilir.

#### Ø Aşağı yukarı Sayıcı (CTUD)

CTUD sayıcıda CU (yukarı sayma) girişine uygulanan sinyalin her yükselen kenarında sayıcı anlık değeri bir artarken, CD (aşağı sayma) girişine uygulanan sinyalin her yükselen kenarında sayıcı anlık değeri bir azalır. Sayıcı anlık değeri CX ile belirtilen sayıcı adresinde saklanır. Herhangi bir anda sayıcı anlık değeri, PV sayma değerine eşit ya da büyük olduğunda sayıcı biti (CX) ile adreslenen kontaklar konum değiştirir. R girişi sayıcı resetleme( sıfırlama) girişidir. R girişine lojik 1 uygulandığında CX adresinde bulunan sayıcı anlık değeri ve sayıcı biti sıfırlanır. Konum değiştirmiş olan sayıcı kontakları normal konumlarına döner.

LADDER SEMBOLÜ



STL KOMUTU:

**CTD CX,PV**

SAYICI KOMUTLARI OPERANTLARI:

CU, CD, R : **I, Q, M, SM, S, T, C, V, L** hafıza alan bitleri ile adreslenen bir kontakta sonra bağlanır.

CX: **C0 ... C255** arasındaki sayıcı adresleri

PT: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C, \* VD, \* LD, \* AC, Sabit ( -32768 ... 32767 )**

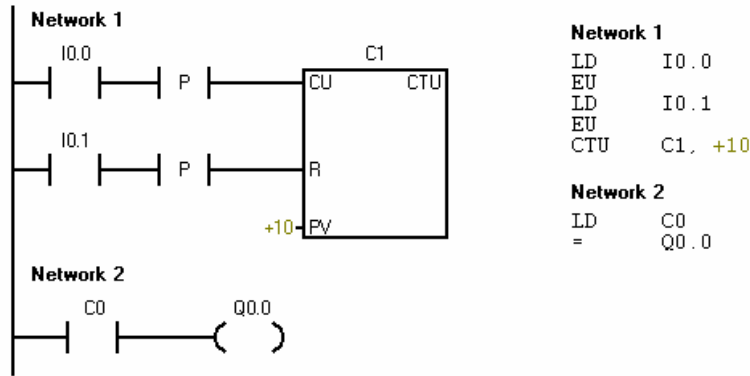
### 2.8.17. Harici Olarak Sayıcı Ayar Değerinin Atanması

Sayıcı ayar değeri PV kullanıcı tarafından atanır. PV ayar değeri desimal olarak girilir ve en fazla -32768 ... +32767 aralığında olabilir. PV değeri doğrudan sabit sayı olarak girilebilir ya da word uzunluğunda başka bir hafıza alan adresi operant olarak kullanılabilir.

### 2.8.18. Sayıcı Uygulamaları

**UYGULAMA:** I0.0 girişine bağlı sensörün önünden 10 parça geçtiğinde Q0.0 çıkışı aktif olacaktır, I0.1 girişine bağlı sensör algıladığında ise sayıcı sıfırlanacaktır. Gerekli PLC programını yazınız?

**Çözüm:**



### 2.8.19. Karşılaştırma Komutları ve Uygulaması

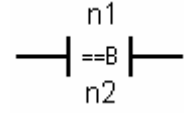
Karşılaştırma komutları iki veriyi büyüklük, küçüklük ve eşitlik ölçütlerine göre kıyaslayarak gerekli kumanda işaretlerinin üretilmesini sağlarlar. Karşılaştırma komutları ladder diyagramlarında bir karşılaştırma kontağı ile gösterilir.

#### Ø Bayt karşılaştırma komutları

Bayt uzunluğundaki ( 0 – 255 ) iki operant içeriği karşılaştırılır. Operantları bayt uzunluğundaki herhangi bir hafıza alan adresi ve sabit değer olabilir. Karşılaştırma kontaklarının devreye bağlantı mantığı normal kontaklar gibidir. Bu yüzden STL dili programlarında karşılaştırma kontakları LD (Başlangıç ), A ( Seri bağlı ), ve O ( Paralel) bağlı konumda olabilir.

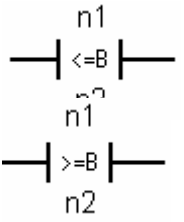
LADDER SEMBOL:

STL :



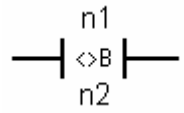
**LDB=** n1,n2  
**AB=** n1,n2  
**OB=** n1,n2

n1 değeri n2 değerine eşit olduğunda karşılaştırma kontağı kapanarak enerji akışına izin verir.



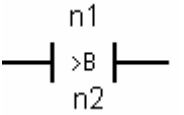
**LDB<=** n1,n2  
**AB<=** n1,n2  
**OB<=** n1,n2  
**LDB>=** n1,n2  
**AB>=** n1,n2  
**OB>=** n1,n2

n1 değeri n2 değerine eşit ya da n2 değerinden küçük olduğunda karşılaştırma kontağı kapanarak enerji akışına izin verir.



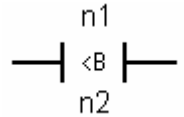
**LDB<>** n1,n2  
**AB<>** n1,n2  
**OB<>** n1,n2

n1 değeri n2 değerine eşit değilse karşılaştırma kontağı kapanarak enerji akışına izin verir.



**LDB>** n1,n2  
**AB>** n1,n2  
**OB>** n1,n2

n1 değeri n2 değerinden büyük olduğunda karşılaştırma kontağı kapanarak enerji akışına izin verir.



**LDB<** n1,n2  
**AB<** n1,n2  
**OB<** n1,n2

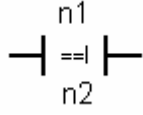
n1 değeri n2 değerinden küçük olduğunda karşılaştırma kontağı kapanarak enerji akışına izin verir.

#### Ø Tamsayı karşılaştırma komutları

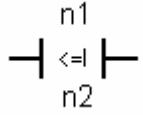
Bir word uzunluğundaki (-32768 ... +32767) tamsayıların karşılaştırılmasında kullanılırlar. Operantları word uzunluğundaki herhangi bir hafıza alan adresi ve sabit değer olabilir. Tamsayı karşılaştırma komutlarının Ladder ve STL gösterim biçimi alan belirteci dışında aynıdır.

LADDER

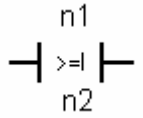
STL :



**LDW= n1,n2**  
**AW= n1,n2**  
**OW= n1,n2**



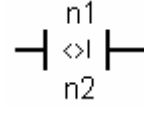
**LDW<= n1,n2**  
**AW<= n1,n2**  
**OW<= n1,n2**



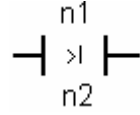
**LDW>= n1,n2**  
**AW>= n1,n2**  
**OW>= n1,n2**

LADDER :

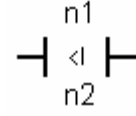
STL :



**LDW<> n1,n2**  
**AW<> n1,n2**  
**OW<> n1,n2**



**LDW > n1,n2**  
**AW > n1,n2**  
**OW > n1,n2**



**LDW < n1,n2**  
**AW < n1,n2**  
**OW < n1,n2**

Double word uzunluğundaki verilerin karşılaştırılması için alan belirteci D olan Double word karşılaştırma komutları, Double word uzunluğundaki alanda bulunan reel sayıları karşılaştırmak için ise alan belirteci R olan karşılaştırma komutu kullanılır.

KARŞILAŞTIRMA KOMUTLARINDA N1 ve N2 İÇİN OPERANTLAR:

BAYT KARŞILAŞTIRMA:

**IB, QB, VB, MB, SMB, SB, LB, AC, \* VD, \* LD, \* AC, Sabit**

WORD TAMSAYI KARŞILAŞTIRMA:

**IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C, \* VD, \* LD, \* AC, Sabit**

DOUBLE WORD TAMSAYI KARŞILAŞTIRMA:

**ID, QD, VD, MD, SMD, SD, LD, AC, HD, \* VD, \* LD, \* AC, Sabit**

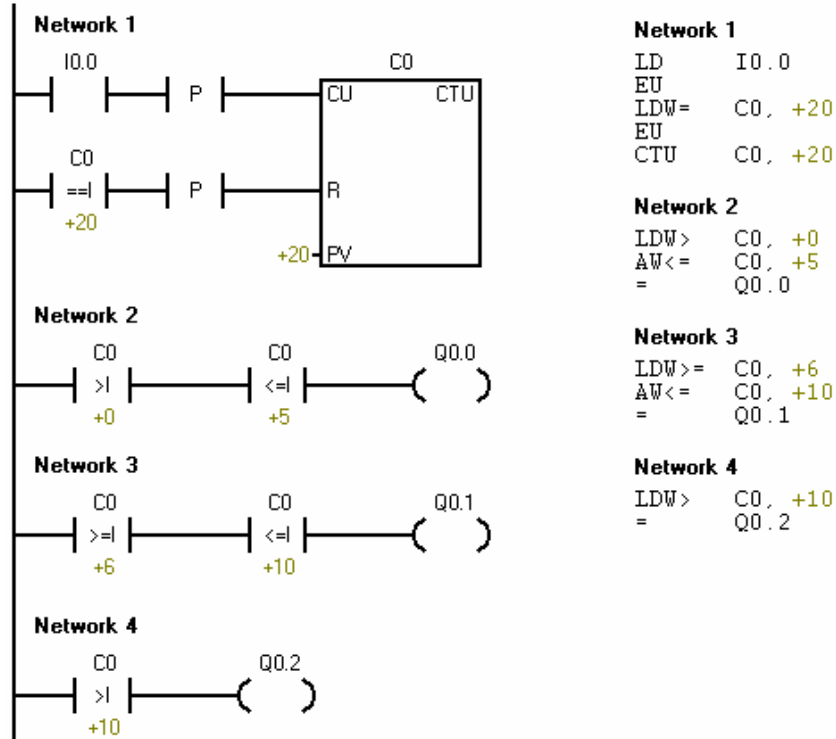
REEL SAYI KARŞILAŞTIRMA:

**ID, QD, VD, MD, SMD, SD, LD, AC, \* VD, \* LD, \* AC, Sabit**

**Uygulama:** I0.0 girişine bağlı butona her basılışta yukarı sayıcı değeri bir artacaktır. Sayıcı içeriği 0 ile 5 arasında (5 dahil) ise Q0.0 çıkışı, 6 ile 10 arasında ise Q0.1 çıkışı, eğer sayıcı içeriği 10'dan büyükse Q0.2 çıkışı aktif olacaktır. Sayıcı içeriği 20 olduğunda sayıcı sıfırlanacaktır.



### Çözüm:



## 2.8.20. Sayı Sistemleri ile Taşıma ( MOVE) Komutları ve Uygulaması

### 2.8.20.1. Sayı Sistemleri

- Ø **İkili ( Binary) Sayı Sistemi:** İkili sayı sisteminin tabanı 2 dir. Bu sayı sistemde sadece 0 ve 1 rakamları kullanılır. Sayısal elektronikte ( Bilgisayar, PLC, hesap makinesi vb.) veriler ikili sayı sistemi kullanılarak işlenir ve depolanır. İkili sayı sisteminde 0 rakamı sistemdeki gerilimin toprak seviyesinde (0 V) olduğunu, 1 rakamı ise sistemdeki gerilimin (+5V gibi) belli bir değerde olduğunu gösterir. PLC programlamada veriler binary sayı sistemi kullanılarak PLC ortamına aktarılabilir.
- Ø **Sekizli (Okta) Sayı Sistemi:** 0, 1, 2 ,3, 4, 5, 6, 7 rakamlarından oluşan 8 tabanlı sayı sistemidir. Bazı mikrobilgisayar sistemlerinde okta sayı sistemi kullanılmaktadır.
- Ø **Onlu ( Desimal) Sayı Sistemi:** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 rakamlarından oluşan 10 tabanlı sayı sistemidir. Günlük hayatta matematikte kullandığımız sayı sistemidir. PLC programlamada veriler onluk sayı sistemi kullanılarak PLC ortamına aktarılabilir.

Ø **Onaltılı( Hexadesimal) Sayı Sistemi:** 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11) , C (12) , D (13), E (14), F (15) rakamlarından oluşan 16 tabanlı sayı sistemidir. Bilgisayar sistemlerinde daha az bit kullanarak, daha çok veriyi işleyip saklayabilmek için geliştirilmiş sayı sistemidir. PLC programlamada veriler on altılı sayı sistemi kullanılarak PLC ortamına aktarılabilir.

#### Ø **Sayı Sistemlerinin Birbirine Dönüştürülmesi**

PLC programlarının yazılmasında binary, desimal ve heksadesimal sayılar kullanılır. Bir hafıza alanına veri yüklemek için bu üç sayı sisteminden herhangi birisi seçilebilir.

- **Binary Sayıların Heksadesimal Sayılara Dönüştürülmesi**

Yan yana gelmiş 4 bit binary sayı bir heksadesimal sayıyla gösterilir. Bu dönüşümde 8421 dizisinden yararlanılır. İkilik sayı 8421 dizisinin altına yazılır. Karşısına ikilik sayıya 1 denk gelen sayılar toplanır ve heksadesimal sayı elde edilir. 4 bitten daha uzun binary sayıları dönüştürmek için sayılar sağdan sola doğru 4'er gruplandırılır ve her grubun heksadesimal karşılığı bulunur. Eğer en soldaki grupta bulunan bit sayısı 4 den az ise 0 ilave edilerek tamamlanır.

**Örnek:**  $(1010)_2$  binay sayısını heksadesimal sayı sistemine dönüştürünüz?

<b>8 4 2 1</b>	8 ve 2 nin karşısında ikilik sayıdaki birler bulunmaktadır.
1 0 1 0	Öyleyse $8+2= 10 = (A)_{16}$ verilen sayının headesimal karşılığıdır.

<b>Örnek:</b> $(01101101)_2 = ( ? )_{16}$	<table border="0"><tr><td><b>8 4 2 1</b></td><td><b>8 4 2 1</b></td></tr><tr><td>0 1 1 0</td><td>1 1 0 1</td></tr><tr><td><b>6</b></td><td><b>D</b></td></tr></table>	<b>8 4 2 1</b>	<b>8 4 2 1</b>	0 1 1 0	1 1 0 1	<b>6</b>	<b>D</b>
<b>8 4 2 1</b>	<b>8 4 2 1</b>						
0 1 1 0	1 1 0 1						
<b>6</b>	<b>D</b>						
$(01101101)_2 = ( 6D )_{16}$							

- **Heksadesimal Sayıların Binary Sayılara Dönüştürülmesi**

Heksadesimal sayının her basamağı 4 bitlik binary sayıya çevrilir. Bu işlem için 8421 dizisinden yararlanılır. 8421 dizisinde, toplamaları heksadesimal sayının basamağındaki değeri veren sayıların altına 1'ler yerleştirilir. Dizinin kullanılmayan sayılarının altına sıfır yazılır.

**Örnek:**  $( 3A )_{16} = ( ? )_2$

<b>8 4 2 1</b>	<b>8 4 2 1</b>
0 0 1 1	1 0 1 0
$( 3A )_{16} = ( 00111010 )_2$	

- **Binary Sayıların Desimal Sayılara Dönüştürülmesi**

Binary sayının bitleri sağdan başlanarak 2'nin kuvvetleri ile çarpılır ve daha sonra çıkan sonuçlar toplanır.

**Örnek:**  $(10011010)_2 = ( ? )_{10}$

$$= 1.2^7 + 0.2^6 + 0.2^5 + 1.2^4 + 1.2^3 + 0.2^2 + 1.2^1 + 0.2^0$$

$$= 128 + 0 + 0 + 16 + 8 + 0 + 2 + 0 = (159)_{10}$$

	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
...	1	0	0	1	1	0	1	0
	128	0	0	16	8	0	2	0

$$128 + 0 + 0 + 16 + 8 + 0 + 2 + 0 = 159$$

- **Desimal Sayıların Binary Sayılara Dönüştürülmesi**

Desimal sayıları binary sayılara dönüştürmek için 2'ye bölme metodu uygulanır. Desimal sayı bölüm 0 oluncaya kadar sürekli 2'ye bölünür. Daha sonra kalanlar tersten yazılarak sayının binary karşılığı elde edilir.

**Örnek:**  $(27)_{10} = ( ? )_2$

Bölünen / Bölen	Bölüm	KALAN	
27 / 2	13	1	<div style="display: flex; align-items: center;"> <div style="border-left: 1px solid black; height: 100px; margin-right: 5px;"></div> <div style="font-size: 2em;">↑</div> </div>
13 / 2	6	1	
6 / 2	3	0	
3 / 2	1	1	
1 / 2	0	1	

$(27)_{10} = (11011)_2$

Pratik bir yöntem olarak sağdan sola doğru 2'nin kuvvetleri şeklinde giden sayı dizisinde, toplamı binary sayıya dönüştürülecek desimal sayıyı veren sayıların altına 1 yerleştirilir. Dizideki kullanılmayan sayıların altına 0 yerleştirilir.

	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
...	128	64	32	16	8	4	2	1	
				<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	

$$16 + 8 + 2 + 1 = 27$$



Bir word ve double word uzunluğundaki hafıza alanlarına sabit veri girişi aynı şekilde yapılır. Bayt alanlarından farklı olarak word alanlarına işaretli tam sayı, double word alanlarına işaretli tam sayı ve reel sayılar yerleştirilebilir.

**NOT:** Eğer sadece V hafıza alanlarına büyük miktarda sabit veri girişi yapılması gerekiyorsa bu işlem için Data Block penceresini kullanmanız gerekir. Eğer bu işlemi komutlarla yaparsanız programınız gereksiz bir şekilde uzar.

#### 2.8.20.2. Taşıma (MOVE) Komutları

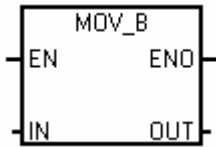
Bayt, Word ve Double Word uzunluğundaki sabit verilerin aynı uzunluktaki bir hafıza alanına ya da bir hafıza alanında bulunan Bayt, Word ve Double Word uzunluğundaki verinin aynı uzunluktaki başka bir hafıza alanına taşımak için kullanılan komutlardır.

Move komutları kaynak ve hedef olmak üzere iki operant kullanır. Taşınacak verinin bulunduğu alan kaynak, taşınacak verinin yazılacağı alan ise hedef operanttır. Move komutlarının IN girişlerine kaynak, OUT çıkışlarına ise hedef operant yazılır. Taşıma işleminin gerçekleşmesi için komutların EN girişlerinin enerjilenmesi gerekir. EN girişleri kesinlikle doğrudan enerji kaynağına bağlanamaz. Bu işlem bir kontak ile gerçekleştirilmelidir.

##### Ø Bayt Taşıma (MOV\_B) Komutu

EN girişi enerjilendiğinde, IN girişi ile belirtilen kaynak operanttaki bayt uzunluğunda veriyi OUT çıkışı ile belirtilen yine bayt uzunluğundaki hedef operant alanına taşır. Bu taşımada kaynak operant içeriğinde herhangi bir değişiklik olmaz. Taşıma komutunun icrası sırasında bir hata oluşursa ENO çıkışı lojik 0 değerini alır.

LADDER:



STL: **MOVB IN, OUT**

OPERANTLAR:

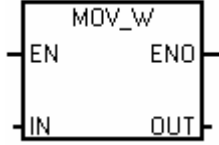
IN: IB, QB, VB, MB, SMB, SB, LB, AC, \* VD, \* LD, \* AC Sabit  
OUT: IB, QB, VB, MB, SMB, SB, LB, AC, \* VD, \* LD, \* AC

##### Ø Word Taşıma (MOV\_W) Komutu

EN girişi enerjilendiğinde, IN girişi ile belirtilen kaynak operanttaki Word uzunluğunda veriyi OUT çıkışı ile belirtilen yine word uzunluğundaki hedef operant alanına taşır. Bu taşımada kaynak operant içeriğinde herhangi bir değişiklik olmaz. Taşıma komutunun icrası sırasında bir hata oluşursa ENO çıkışı lojik 0 değerini alır.

LADDER:

STL: **MOVB IN, OUT**



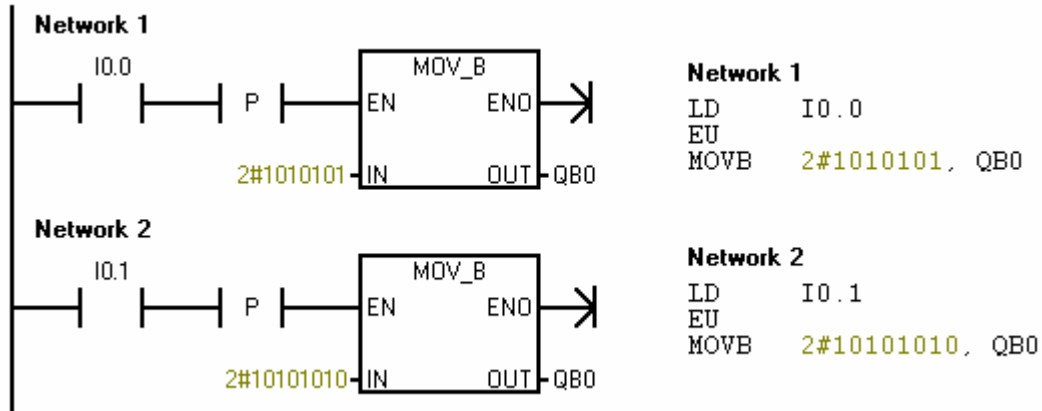
OPERANTLAR:

IN: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C, Sabit**  
OUT: **IW, QW, VW, MW, SMW, SW, LW, AQW, AC, T, C**  
\* **VD, \* LD, \* AC,**

Double word uzunluğundaki verileri taşımak için MOV\_D, double word uzunluğundaki alanda bulunan reel sayıları taşımak için ise MOV\_R komutu kullanılır. Bu komutların kullanım şekli ve operantları için S7 – 200 kullanım kılavuzuna bakınız.

**Uygulama:** I0.0 girişine bağlı start butonuna basıldığında çift numaralı çıkışlar (Q0.0, Q0.2, Q0.4, Q0.6 ) I0.1 girişine bağlı start butonuna basıldığında ise tek numaralı çıkışlar (Q0.1, Q0.3, Q0.5, Q0.7 ) aktif olacaktır.

**Çözüm:**



## 2.8.21. Matematik İşlem, Fonksiyon Komutları ve Uygulaması

PLC cihazlarında toplama, çıkarma, çarpma, bölme, karekök alma gibi işlevleri yerine getirebilen matematik işlem ve fonksiyon komutları bulunmaktadır. Bu fonksiyonların kullanılmasıyla kumanda ve kontrol sistemlerinde uzunluk hesabı, alan hesabı, hacim hesabı devir sayısı ölçümü gibi işlemleri yapmak mümkün olmaktadır.

Matematik işlem komutlarının icra edilmesi sonucunda yapılan işlemin sonucuna bağlı olarak SMB1 özel hafıza baytı alanında bulunan bazı durum bitleri set (lojik 1) olur. Bu durum bitleri kullanılarak işlem sonuçlarına göre program akışı değiştirilebilir.

- Ø **SM1.0:** Matematiksel işlem yapan komutların icrası sonucunda elde edilen sonuç sıfır (0) ise bu bit set olur. Sıfır biti olarak anılır.

- Ø **SM1.1:** Matematiksel işlem yapan komutların icrası sonucunda taşma meydana gelmesi veya geçersiz nümerik değer saptanması durumunda bu bit set olur. Taşma biti olarak anılır.
- Ø **SM1.2:** Matematiksel işlem yapan komutların icrası sonucunda elde edilen sonuç negatif bir sayı ise bu bit set olur. İşaret biti olarak anılır.
- Ø **SM1.3:** Bölme komutlarının icrası sırasında bir sayıyı sıfır ile bölme girişimi varsa bu bit set olur. Sıfıra bölme hatası biti olarak anılır.

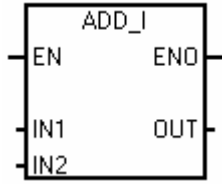
#### 2.8.21.1. Matematik İşlem Komutları

##### Ø Tamsayı Toplama (ADD\_I) Komutu

Word uzunluğundaki iki tam sayının toplanması için kullanılır. Toplanacak sayılar IN1 ve IN2 girişlerine operant olarak yazılır. Operantlar sabit sayılar ve word uzunluğundaki hafıza alan adresleri olabilir. Toplam sonucu OUT çıkışı ile belirtilen word uzunluğundaki hedef operant alanına yazılır. (  $IN1+IN2 = OUT$  )

Toplama işleminin sonucuna bağlı olarak SM1.0, SM1.1 ve SM1.2 özel hafıza bitleri etkilenir. Toplama işlemi sırasında bir hata oluşursa ENO çıkışı lojik 0 değerini alır.

LADDER:



STL: **MOVW IN1,OUT**  
**+I IN2,OUT**

OPERANTLAR:

IN1,IN2: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C, \*VD, \*LD, \*AC, Sabit**

OUT: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C, \*VD, \*LD, \*AC**

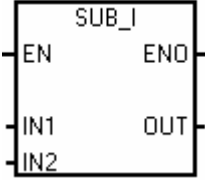
Double word uzunluğundaki tamsayıları toplamak için ADD\_DI, Double word uzunluğundaki alanda bulunan reel sayıları toplamak ise ADD\_R komutu kullanılır. Bu komutların kullanım şekli ve operantları için S7 – 200 kullanım kılavuzuna bakınız.

##### Ø Tamsayı Çıkarma (SUB\_I) Komutu

Word uzunluğundaki iki tam sayının çıkarılması için kullanılır. Çıkarılacak sayılar IN1 ve IN2 girişlerine operant olarak yazılır. Operantlar sabit sayılar ve word uzunluğundaki hafıza alan adresleri olabilir. Çıkarma işlemi sonucu OUT çıkışı ile belirtilen word uzunluğundaki hedef operant alanına yazılır. (  $IN1- IN2 = OUT$  )

Çıkarma işleminin sonucuna bağlı olarak SM1.0, SM1.1 ve SM1.2 özel hafıza bitleri etkilenir. Çıkarma işlemi sırasında bir hata oluşursa ENO çıkışı lojik 0 değerini alır.

LADDER:



STL: **MOVW IN1, OUT**  
**-I IN2, OUT**

OPERANTLAR:

IN1, IN2: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C,**  
**\*VD, \*LD, \*AC, Sabit**  
OUT: **IW, QW, VW, MW, SMW, SW, LW, AC, T, C,**  
**\*VD, \*LD, \*AC**

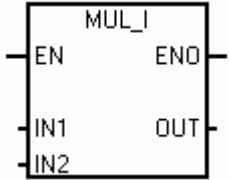
Double word uzunluğundaki tam sayıları çıkarmak için SUB\_DI, double word uzunluğundaki alanda bulunan reel sayıları çıkarmak ise SUB\_R komutu kullanılır.

### Ø Tamsayı Çarpma (MUL\_I) Komutu

Word uzunluğundaki iki tam sayının çarpılması için kullanılır. Çarpılacak sayılar IN1 ve IN2 girişlerine operant olarak yazılır. Operantlar sabit sayılar ve word uzunluğundaki hafıza alan adresleri olabilir. Çarpım sonucu OUT çıkışı ile belirtilen word uzunluğundaki hedef operant alanına yazılır. (  $IN1 * IN2 = OUT$  )

Çarpma işleminin sonucuna bağlı olarak SM1.0, SM1.1 ve SM1.2 özel hafıza bitleri etkilendir. Çarpma işlemi sırasında bir hata oluşursa ENO çıkışı lojik 0 değerini alır.

LADDER:



STL: **MOVW IN1, OUT**  
**\*I IN2, OUT**

OPERANTLAR:

IN1, IN2: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C,**  
**\*VD, \*LD, \*AC, Sabit**  
OUT: **IW, QW, VW, MW, SMW, SW, LW, AC, T, C,**  
**\*VD, \*LD, \*AC**

Double word uzunluğundaki tamsayıları çarpmak için MUL\_DI, double word uzunluğundaki alanda bulunan reel sayıları çarpmak için ise MUL\_R komutu kullanılır. Bu komutların kullanım şekli ve operantları için S7 – 200 kullanım kılavuzuna bakınız.

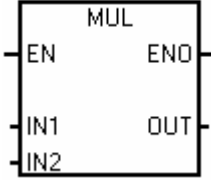
### Ø Double Tam Sayı Sonuçlu Tam Sayı Çarpma ( MUL ) Komutu

Bu komut IN1 ve IN2 girişlerinde bulunan word uzunluğundaki iki tam sayıyı çarpar ve sonucu OUT çıkışı ile belirtilen double word uzunluğundaki hafıza alanına yazar. (  $IN1 * IN2 = OUT$  )

Çarpma işleminin sonucuna bağlı olarak SM1.0, SM1.1 ve SM1.2 özel hafıza bitleri etkilendir. Çıkarma işlemi sırasında bir hata oluşursa ENO çıkışı lojik 0 değerini alır.



LADDER:



STL: **MOVW IN1, OUT**  
**MUL IN2, OUT**

OPERANTLAR:

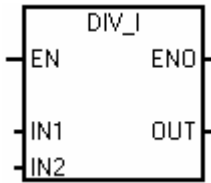
IN1, IN2: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C,**  
**\*VD, \*LD, \*AC, Sabit**  
OUT: **ID, QD, VD, MD, SMD, SD, LD, AIW, AC,**  
**\*VD, \*LD, \*AC**

### Ø Tam Sayı Bölme (DIV\_I) Komutu

Bu komut IN1 girişinde bulunan word uzunluğundaki tam sayıyı, IN2 girişinde bulunan ve yine word uzunluğundaki tam sayıya böler. Bölüm sonucu OUT çıkışı ile belirtilen word uzunluğundaki hafıza alanına yazılır. Bölme işlemi sonucunda elde edilen kalan dikkate alınmaz. ( herhangi bir alanda saklanmaz.)

Bölme işleminin sonucuna bağlı olarak SM1.0, SM1.1, SM1.2 ve SM1.3 özel hafıza bitleri etkilenir. Bölme işlemi sırasında bir hata oluşursa ENO çıkışı lojik 0 değerini alır.

LADDER:



STL: **MOVW IN1, OUT**  
**/I IN2, OUT**

OPERANTLAR:

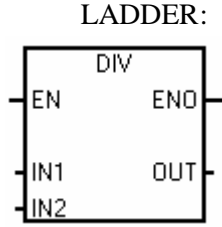
IN1, IN2: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C,**  
**\*VD, \*LD, \*AC, Sabit**  
OUT: **IW, QW, VW, MW, SMW, SW, LW, AC, T, C,**  
**\*VD, \*LD, \*AC**

Double word uzunluğundaki tam sayıların bölünmesi için DIV\_DI, double word uzunluğundaki alanda bulunan reel sayıların bölünmesi için ise DIV\_R komutu kullanılır. Bu komutların kullanım şekli ve operantları için S7 – 200 kullanım kılavuzuna bakınız.

### Ø Kalanlı Tam Sayı Bölme (DIV) Komutu

Bu komut IN1 girişinde bulunan word uzunluğundaki tam sayıyı IN2 girişinde bulunan yine word uzunluğundaki tam sayıya böler. Bölme işlemi sonucunda elde edilen bölüm ve kalan OUT çıkışı ile belirtilen double word uzunluğundaki hafıza alanına yazılır. Bölüm, double word uzunluğundaki alanın düşük değerli wordüne, kalan ise yüksek değerli wordüne yerleşir.

Bölme işleminin sonucuna bağlı olarak SM1.0, SM1.1, SM1.2 ve SM1.3 özel hafıza bitleri etkilenir. Bölme işlemi sırasında bir hata oluşursa ENO çıkışı lojik 0 değerini alır.



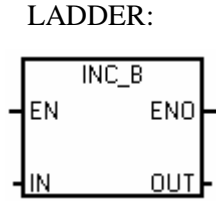
STL: **MOVW IN1, OUT**  
**DIV IN2, OUT**

OPERANTLAR:  
 IN1, IN2: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C, \*VD, \*LD, \*AC, Sabit**  
 OUT: **ID, QD, VD, MD, SMD, SD, LD, AC, \*VD, \*LD, \*AC**

#### 2.8.21.2. Arttırma ve Azaltma Komutları

##### Ø Bayt İçeriği Arttırma ( INC\_B) Komutu

Bu komut EN girişi her enerjilendiğinde, IN girişinde bulunan bayt uzunluğundaki operant içeriğini bir artırarak sonucu OUT çıkışı ile belirtilen bayt uzunluğundaki operant alanına yazar.



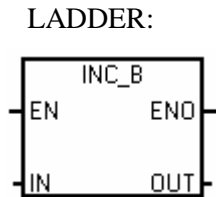
STL: **MOVW IN, OUT**  
**INCB OUT**

OPERANTLAR:  
 IN: **IB, QB, VB, MB, SMB, SB, LB, AC, \*VD, \*LD, \*AC, Sabit**  
 OUT: **IB, QB, VB, MB, SMB, SB, LB, AC, \*VD, \*LD, \*AC**

Word uzunluğundaki değerler için INC\_W, Double word uzunluğundaki değerler için INC\_DW komutları kullanılır. Bu komutların kullanım şekli ve operantları için S7 – 200 kullanım kılavuzuna bakınız.

##### Ø Bayt İçeriği Azaltma ( DEC\_B) Komutu

Bu komut EN girişi her enerjilendiğinde, IN girişinde bulunan bayt uzunluğundaki operant içeriğini bir azaltarak sonucu OUT çıkışı ile belirtilen bayt uzunluğundaki operant alanına yazar.



STL: **MOVW IN, OUT**  
**INCB OUT**

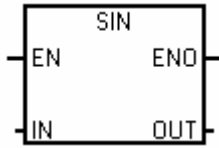
OPERANTLAR:  
 IN: **IB, QB, VB, MB, SMB, SB, LB, AC, \*VD, \*LD, \*AC, Sabit**  
 OUT: **IB, QB, VB, MB, SMB, SB, LB, AC, \*VD, \*LD, \*AC**

Word uzunluğundaki değerler için DEC\_W, double word uzunluğundaki değerler için DEC\_DW komutları kullanılır. Bu komutların kullanım şekli ve operantları için S7 – 200 kullanım kılavuzuna bakınız.

### 2.8.21.3. Nümerik Fonksiyon Komutları

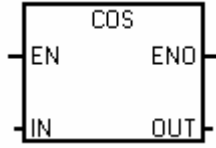
Ø **Sinüs, Kosinüs ve Tanjant Komutları:** Sinüs (SIN), Kosinüs (COS) ve Tanjant (TAN) komutları IN girişlerinde bulunan radyan cinsinden açısal değerin trigonometrik değerini hesaplayarak sonucu OUT çıkışı ile belirtilen double word uzunluğundaki hafıza alanına yazar. İşlemin sonucuna bağlı olarak SM1.0, SM1.1 ve SM1.2 özel hafıza bitleri etkilenir. Çıkarma işlemi sırasında bir hata oluşursa ENO çıkışı lojik 0 değerini alır.

LADDER:



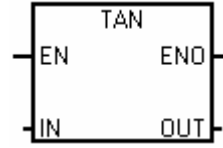
STL: SIN IN,OUT

LADDER:



STL: COS IN,OUT

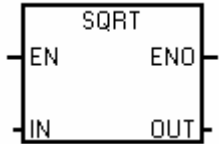
LADDER:



STL: TAN IN,OUT

Ø **Karekök (SQRT) Komutu:** IN girişinde bulunan herhangi bir reel sayının karekökünü alarak sonucu OUT çıkışı ile belirtilen double word uzunluğundaki hafıza alanına yazar. İşlemin sonucuna bağlı olarak SM1.0, SM1.1 ve SM1.2 özel hafıza bitleri etkilenir. Çıkarma işlemi sırasında bir hata oluşursa ENO çıkışı lojik 0 değerini alır.

LADDER:



STL: SQRT IN,OUT :

NÜMERİK FONKSİYON KOMUTLARI OPERANTLARI:

IN: ID, QD, VD, MD, SD, LD, AC, \* VD, \* LD, \* AC, Sabit sayılar

OUT: ID, QD, VD, MD, SMD, SD, LD, AC, \* VD, \* LD, \* AC

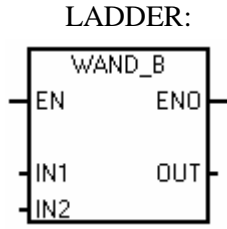
**NOT:** Bir sayının herhangi bir üssünü hesaplamak için doğal logaritma hesaplama (LN ) komutu ile doğal üssel fonksiyon hesaplama (EXP) komutlarından yararlanılır. Ayrıntılı bilgi için S7 – 200 kullanma klavuzuna bakınız.

Yukarıda sizlere matematik işlem ve fonksiyon komutları anlatılmıştır. Bu komutların özelliklerini öğrendiniz. Bu komutlarla ilgili örnek programaları internet ortamı ve değişik otomasyon kitaplarında bulup uygulama yapabilirsiniz.

## 2.8.22. Mantık İşlemler, Kaydırma, Dönüştürme Komutları ve Uygulaması

### 2.8.22.1. Mantık İşlem Komutları

- Ø **AND Bayt (WAND\_B) Komutu:** Bu komut IN1 ve IN2 girirlerinde bulunan byte uzunluğundaki iki değerin karşılıklı bitlerini lojik olarak AND işlemine tabi tutar ve sonucu OUT çıkışı ile belirtilen bayt uzunluğundaki operant alanına yazar. İşlem sonucu sıfır ise SM1.0 biti set olur.

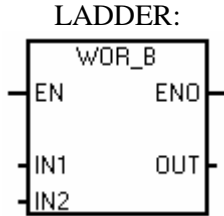


STL:  
**MOVB IN1, OUT**  
**ANDB IN2, OUT**

OPERANTLAR:  
IN1, IN2: **IB, QB, VB, MB, SMB, SB, LB, AC, \*VD, \*LD, \*AC, Sabit**  
OUT: **IB, QB, VB, MB, SMB, SB, LB, AC, \*VD, \*LD, \*AC**

Word uzunluğundaki değerler için **WAND\_W**, double word uzunluğundaki değerler için **WAND\_D** komutları kullanılır.

- Ø **OR Bayt (WOR\_B) Komutu:** Bu komut IN1 ve IN2 girirlerinde bulunan byte uzunluğundaki iki değerin karşılıklı bitlerini lojik olarak OR işlemine tabi tutar ve sonucu OUT çıkışı ile belirtilen bayt uzunluğundaki operant alanına yazar. İşlem sonucu sıfır ise SM1.0 biti set olur.

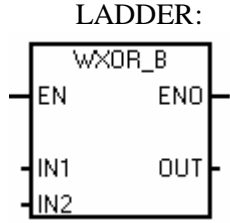


STL:  
**MOVB IN1, OUT**  
**ORB IN2, OUT**

OPERANTLAR:  
IN1, IN2: **IB, QB, VB, MB, SMB, SB, LB, AC, \*VD, \*LD, \*AC, Sabit**  
OUT: **IB, QB, VB, MB, SMB, SB, LB, AC, \*VD, \*LD, \*AC**

Word uzunluğundaki değerler için **WOR\_W**, double word uzunluğundaki değerler için **WOR\_D** komutları kullanılır.

- Ø **XOR Bayt (WXOR\_B) Komutu:** Bu komut IN1 ve IN2 girirlerinde bulunan byte uzunluğundaki iki değerin karşılıklı bitlerini lojik olarak XOR işlemine tabi tutar ve sonucu OUT çıkışı ile belirtilen bayt uzunluğundaki operant alanına yazar. İşlem sonucu sıfır ise SM1.0 biti set olur.



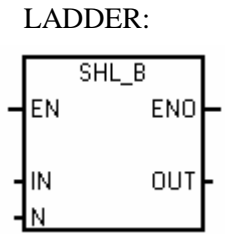
STL:  
**MOVB IN1, OUT**  
**B IN2, OUT**

OPERANTLAR:  
 IN,N: **IB,QB,VB,MB,SMB,SB,LB,AC,\*VD,\*LD,\*AC, Sabit**  
 OUT: **IB,QB,VB,MB,SMB,SB,LB,AC,\*VD,\*LD,\*AC**

Word uzunluğundaki değerler için WXOR\_W, double word uzunluğundaki değerler için **WXOR\_D** komutları kullanılır.

## 2.8.22.2. Kaydırma Komutları

Ø **Sola Kaydırma ( SHL\_B) Komutu:** Bu komut EN girişi aktif olduğunda IN girişi ile belirtilen bayt alanındaki bitleri N girişi ile belirtilen sayı kadar sola kaydırır ve sonucu OUT çıkışı ile belirtilen bayt alanına yazar. İşlem sonucuna göre SM1.0 ve SM1.1 özel hafıza bitleri etkilenir.

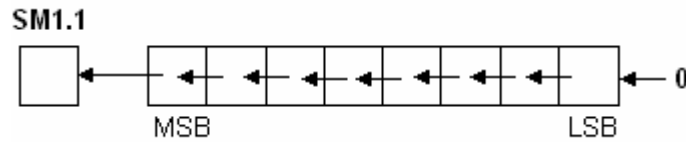


STL:  
**MOV IN, OUT**  
**SLB OUT, N**

OPERANTLAR:  
 IN,N: **IB,QB,VB,MB,SMB,SB,LB,AC,\*VD,\*LD,\*AC, Sabit**  
 OUT: **IB,QB,VB,MB,SMB,SB,LB,AC,\*VD,\*LD,\*AC**

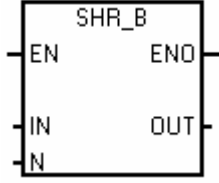
Şekilde görüldüğü gibi bir sola kaydırma işleminde bir baytlık alan içindeki MSB biti SM1.1 taşma bitine aktarılırken diğer tüm bitler bir sola kaydırılır. Sağda boşalan bitlere 0 yüklenir. Birden fazla kaydırma işlemi uygulandığında her kaydırmada MSB biti taşma bitine aktarılırken taşma bitindeki bir önceki değer yok olur.

Word uzunluğundaki alanda bulunan verileri sola kaydırmak için SHL\_W komutunun, double word uzunluğundaki alanda bulunan verileri sola kaydırmak için ise SHL\_DW komutunun kullanılması gerekir.



Ø **Sağa Kaydırma ( SHR\_B) Komutu:** Bu komut EN girişi aktif olduğunda IN girişi ile belirtilen bayt alanındaki bitleri N girişi ile belirtilen sayı kadar sağa kaydırır ve sonucu OUT çıkışı ile belirtilen bayt alanına yazar. İşlem sonucuna göre SM1.0 ve SM1.1 özel hafıza bitleri etkilenir.

LADDER:



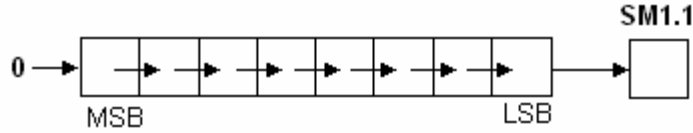
STL:

**MOV IN,OUT**  
**SRB OUT,N**

OPERANTLAR:

IN,N: **IB,QB,VB,MB,SMB,SB,LB,AC,\*VD,\*LD,\*AC, Sabit**  
OUT: **IB,QB,VB,MB,SMB,SB,LB,AC,\*VD,\*LD,\*AC**

Şekilde görüldüğü gibi bir sağa kaydırma işleminde bir baytlık alan içindeki LSB biti SM1.1 taşıma bitine aktarılırken diğer tüm bitler bir sağa kaydırılır. Solda boşalan bitlere 0 yüklenir. Birden fazla kaydırma işlemi uygulandığında her kaydırmada LSB biti taşıma bitine aktarılırken taşıma bitindeki bir önceki değer yok olur.

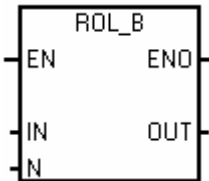


Word uzunluğundaki alanda bulunan verileri sağa kaydırmak için SHR\_W komutunun, double word uzunluğundaki alanda bulunan verileri sağa kaydırmak için ise SHL\_DW komutunun kullanılması gerekir.

### 2.8.22.3. Döndürme Komutları

Ø **Sola Döndürme ( ROL\_B ) Komutu:** Bu komut EN girişi aktif olduğunda IN girişi ile belirtilen bayt alanındaki bitleri N girişi ile belirtilen sayı kadar sola döndürür ve sonucu OUT çıkışı ile belirtilen bayt alanına yazar. İşlem sonucuna göre SM1.0 ve SM1.1 özel hafıza bitleri etkilenir.

LADDER:



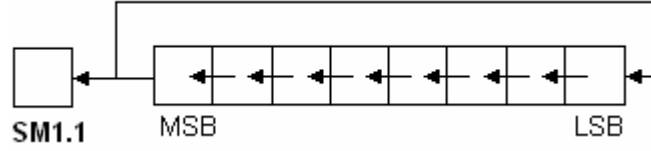
STL:

**MOV IN,OUT**  
**RLB OUT,N**

OPERANTLAR:

IN,N: **IB,QB,VB,MB,SMB,SB,LB,AC,\*VD,\*LD,\*AC, Sabit**  
OUT: **IB,QB,VB,MB,SMB,SB,LB,AC,\*VD,\*LD,\*AC**

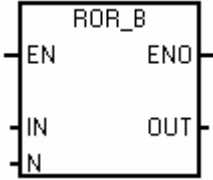
Şekilde bir bayt alanındaki bitlerin bir sola döndürülme işlemi görülmektedir. Döndürme işleminde bayt alanındaki bitler bir sola kayarken MSB biti hem taşıma bitine hem de en sağda boşalan LSB bitine kopyalanır. Birden fazla döndürme işlemi uygulandığında aynı işlemler tekrar edilir.



Word uzunluğudaki alanda bulunan verileri sola döndürmek için **ROL\_W** komutunun, double word uzunluğundaki alanda bulunan verileri sola döndürmek için ise **ROL\_DW** komutunun kullanılması gerekir.

Ø **Sağa Döndürme ( ROR\_B ) Komutu:** Bu komut EN girişi aktif olduğunda IN girişi ile belirtilen bayt alanındaki bitleri N girişi ile belirtilen sayı kadar sağa döndürür ve sonucu OUT çıkışı ile belirtilen bayt alanına yazar. İşlem sonucuna göre SM1.0 ve SM1.1 özel hafıza bitleri etkilenir.

LADDER:



STL:

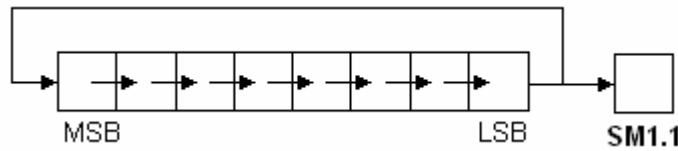
**MOV IN,OUT**  
**RLB OUT,N**

OPERANTLAR:

N1,IN2: **IB,QB,VB,MB,SMB,SB,LB,AC,\*VD,\*LD,\*AC, Sabit**

OUT: **IB,QB,VB,MB,SMB,SB,LB,AC,\*VD,\*LD,\*AC**

Şekilde bir bayt alanındaki bitlerin bir sağa döndürülme işlemi görülmektedir. Döndürme işleminde bayt alanındaki bitler bir sağa kayarken LSB biti hem taşıma bitine hem de en sağda boşalan MSB bitine kopyalanır. Birden fazla döndürme işlemi uygulandığında aynı işlemler tekrar edilir.

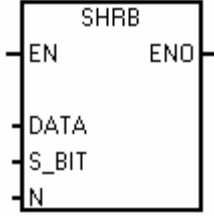


Word uzunluğudaki alanda bulunan verileri sağa döndürmek için **ROR\_W** komutunun, double word uzunluğundaki alanda bulunan verileri sola döndürmek için ise **ROR\_DW** komutunun kullanılması gerekir.

### 2.8.22.3. Kaydırmalı Kaydedici (Shift Register Bit ( SHRB ) Komutu

Bu komut EN girişi aktif olduğunda tanımlanan bir hafıza alanına bir bitlik (1 ya da 0) bilgiyi sağa ya da sola kaydırarak yazar. Komut ürün veya veri akışının sıralanması için kolay bir yöntem sunar.

LADDER:



STL:

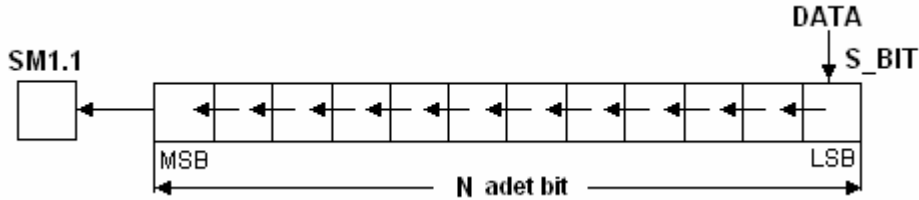
**SHRB DATA, S\_BIT, N**

OPERANTLAR:

N: **IB, QB, VB, MB, SMB, SB, LB, AC, \*VD, \*LD, \*AC, Sabit**

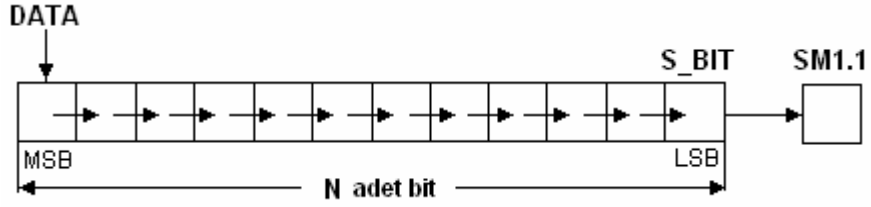
DATA, S\_BIT: **I, Q, M, SM, S, T, C, V, L hafıza alan bitleri**

- Ø **DATA:** Tanımlanan hafıza alanına kaydırılarak yazılacak bir bitlik verinin bulunduğu bit adresidir.
- Ø **S\_BIT:** Kaydırma işlemi yapılacak hafıza alanının en düşük değerli (LSB) bitidir.
- Ø **N:** Kaydırma hafıza alanının uzunluğunu ve kaydırma yönünü belirler. N değeri -128 ...+127 aralığında değer içeren bir bayt alanı ya da sabit olabilir. N değerinden anlaşılacağı gibi kaydırma alanı maksimum 16 bayt ya da 128 bitlik bir hafıza alanı olabilir. N değeri pozitif ise bir bitlik veri sola kaydırılarak yazılır. Eğer N değeri negatif ise bir bitlik veri sağa kaydırılarak yazılır.
- Ø **Sola kaydırarak yazma:** N değeri pozitif ise DATA girişinde bulunan 1 bitlik veri hafıza alanına sola kaydırılarak yazılır. EN girişi aktif olduğunda DATA girişinde bulunan bir bitlik veri, tanımlanan N bitlik hafıza alanının en düşük değerlikli (LSB) bitine yazılır. Hafıza alanının tüm bitleri bir sola kaydırılır. Hafıza alanının en yüksek değerlikli (MSB) biti SM1.1 taşıma bitine aktarılır.



- Ø **Sağa kaydırarak yazma:** N değeri negatif ise DATA girişinde bulunan 1 bitlik veri hafıza alanına sağa kaydırılarak yazılır. EN girişi aktif olduğunda DATA girişinde bulunan bir bitlik veri, tanımlanan N bitlik hafıza alanının en yüksek değerlikli (MSB) bitine yazılır. Hafıza alanının tüm bitleri bir sağa kaydırılır. Hafıza alanının en düşük değerlikli (LSB) biti SM1.1 taşıma bitine aktarılır.





Kaydırmalı kaydedici de DATA girişinde bulunan bir bitlik veri, EN girişi aktif olduğu anda tanımlanan hafıza alanına kaydırılarak yazılır. Yani bir tarama döngüsü içinde sadece bir kaydırma işlemi gerçekleşir. Eğer EN girişi sürekli aktif durumda olursa DATA girişinde bulunan bilgi defalarca kaydırılarak hafıza alanına yazılır. Bu durum komutun kullanım amacına aykırıdır. Program yazımlarında bu husus dikkate alınmalıdır.

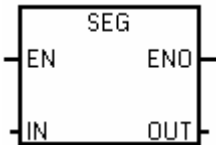
#### 2.8.22.4. Dönüştürme Komutları

PLC programlamada belli bir sayısal formatta elde edilen verilerin başka sayısal formata çevrilmesi işlemlerinde dönüştürme komutları kullanılır S7 – 200 PLC ‘ler tarafından desteklenen pek çok dönüştürme komutu bulunmaktadır. Bu modülde bazı dönüştürme komutları açıklanacaktır. Diğer dönüştürme komutları için kaynak kitaplara bakmanız tavsiye edilir.

**Ø Segment (SEG) komutu:** Bu komut IN girişinde bulunan bayt uzunluğundaki sayısal değeri 7 segmentli led göstergede görüntülenebilecek şekilde ikili sayı formatına dönüştürür.

Tabloda SEG komutunun IN girişinde yer alan heksadesimal olarak 0 – F arasındaki sayıların çıkış formatları görülmektedir.

LADDER:



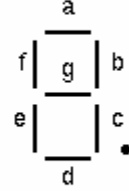
STL:

**SEG IN,OUT**

OPERANTLAR:

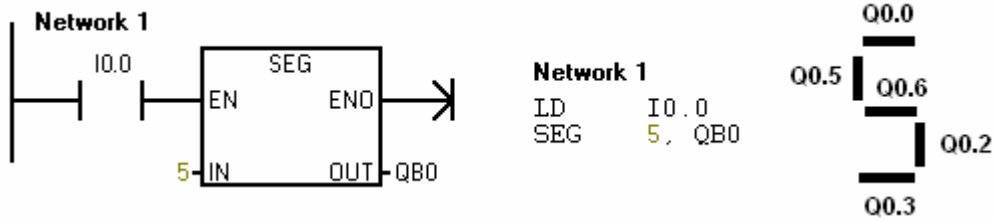
IN: IB, QB, VB, MB, SMB, SB, LB, AC, \* VD, \* LD, \* AC Sabit  
OUT: IB, QB, VB, MB, SMB, SB, LB, AC, \* VD, \* LD, \* AC

IN	Segment Display	OUT • g f e d c b a	IN	Segment Display	OUT • g f e d c b a
0	0	0 0 1 1 1 1 1 1	8	8	0 1 1 1 1 1 1 1
1	1	0 0 0 0 0 1 1 0	9	9	0 1 1 0 0 1 1 1
2	2	0 1 0 1 1 0 1 1	A	A	0 1 1 1 0 1 1 1
3	3	0 1 0 0 1 1 1 1	B	B	0 1 1 1 1 1 0 0
4	4	0 1 1 0 0 1 1 0	C	C	0 0 1 1 1 0 0 1
5	5	0 1 1 0 1 1 0 1	D	D	0 1 0 1 1 1 1 0
6	6	0 1 1 1 1 1 0 1	E	E	0 1 1 1 1 0 0 1
7	7	0 0 0 0 0 1 1 1	F	F	0 1 1 1 0 0 0 1



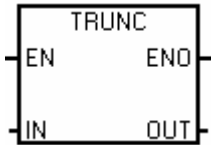
**Uygulama:** PLC nin çıkışlarına bağlanmış 7 segment display’de 5 rakanımı görmek için gerekli PLC programını yazınız?

**Çözüm:**



**Ø Kısaltarak Tam Sayıya Dönüştürme ( TRUNC ) Komutu:** Bu komut IN girişindeki reel sayının ondalık kısmını atarak tamsayıya dönüştürür ve sonucu OUT çıkışı ile belirtilen double word uzunluğundaki bir hafıza alanına aktarır. Örneğin 12.567 reel sayısı ondalık kısmı atılarak 12 tam sayısına dönüştürülür.

LADDER:



STL:

**TRUNC IN,OUT**

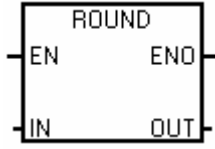
OPERANTLAR:

IN: **ID, QD, VD, MD, SMD, SD, LD, HC, AC, \* VD, \* LD**  
\*AC, Sabit

OUT: **ID, QD, VD, MD, SMD, SD, LD, HC, AC, \* VD, \* LD, \* AC**

**Ø Yuvarlayarak Tam Sayıya Dönüştürme ( ROUND ) Komutu:** Bu komut, IN girişinde bulunan reel sayıyı ondalık kısmındaki değere bağlı olarak bir üst ya da bir alt tam sayıya dönüştürür. Örneğin, 25.5 reel sayısı yuvarlanarak 26 tam sayısına dönüşür.

LADDER:



STL:

**TRUNC IN,OUT**

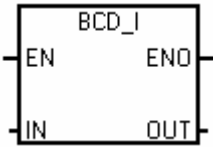
OPERANTLAR:

IN: **ID, QD, VD, MD, SMD, SD, LD, HC, AC, \* VD, \* LD**  
**\*AC, Sabit**

OUT: **ID, QD, VD, MD, SMD, SD, LD, HC, AC, \* VD, \* LD, \* AC**

- Ø **BCD'den Tam Sayıya ( BCD\_I) Komutu:** Bu komut IN girişinde bulunan word uzunluğundaki binary kodlanmış desimal (BCD) sayıyı tam sayıya çevirerek sonucu OUT çıkışı ile belirtilen double word uzunluğundaki bir hafıza alanına aktarır.

LADDER:



STL:

**MOVW IN, OUT**  
**BCDI OUT**

OPERANTLAR:

IN1,IN2: **IW, QW, VW, MW, SMW, SW, LW, AIW, AC, T, C,**  
**\*VD, \*LD, \*AC, Sabit**

OUT: **ID, QD, VD, MD, SMD, SD, LD, AC, \* VD, \* LD, \* AC**

### 2.8.23. Program Kontrol Komutları ve Uygulaması

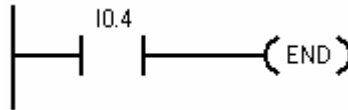
- Ø **Koşullu Son (END) Komutu:** End komutu önündeki şarta bağlı olarak program akışını keserek program taramasının Network 1'den itibaren tekrar başlatılmasını sağlar. END komutu yalnızca anaprogramda kullanılabilir. Alt program ve kesme alt programlarında kullanılamaz.

LADDER: —(END)

STL: **END**

OPERANT: **Yok**

Şekildeki örnekte I0.4 girişi aktif olduğunda program akışı kesilerek ilk komuttan itibaren yeniden başlar.



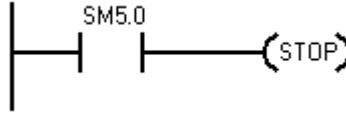
- Ø **STOP Komutu:** Stop komutu önündeki şarta bağlı olarak PLC'nin RUN modundan STOP moduna geçmesini sağlar. Böylece program işletimi tamamen kesilir. Program işletiminin yeniden başlatılması için PLC'nin tekrar RUN moduna alınması gerekir.

LADDER: —(STOP)

STL: STOP

OPERANT: Yok

Şekildeki örnekte SM5.0 özel hafıza biti 1 olduğunda kontak kapanarak STOP komutunu çalıştırır ve PLC stop moduna geçer. SM5.0 giriş çıkış hatası oluşması durumunda set olan özel hafıza bitidir.



Ø **Sıçrama ( JMP ) Komutu:** JMP komutu belli bir şarta bağlı olarak program akışının değiştirilmesi veya istenilen bir program kesiminin işletilmeden geçilmesini için kullanılır. Sıçramanın gerçekleşeceği konum n ile belirtilir.

LADDER: —(JMP)<sup>n</sup>

STL: JMP n

OPERANT: n: 0 – 255

Ø **Etiket (LBL) Komutu:** Bu komut JMP komutu ile yapılan sıçramanın hedefini gösterir. Bu iki komut birlikte kullanıldıkları zaman anlam kazanırlar.

LADDER: —[LBL]<sup>n</sup>

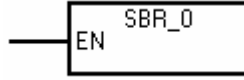
STL: STOP

OPERANT: n: 0 – 255



Ø **Alt Program Çağırma ( CALL) Komutu:** Bu komut SBR\_0, SBR\_1 ... sayfalarına yazılan alt programların ana programdan çağrılarak işletilmesi için kullanılır.

LADDER:

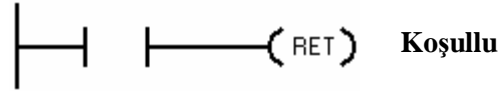


STL: **CALL, SBR\_No**  
OPERANT: **SBR\_0, SBR\_1, SBR\_2 ...**

Ladder editörü ile çalışırken bir alt programa dallanma komutunu program alanına yerleştirmek için komut ağacında bulunan Call Sabroutine bölümünden çağırmak istediğiniz alt programın üzerine çift tıklamanız gerekir. Komutun operantı haricen yazılmaz.

Ø **Alt Programdan Ana Programa Dönüş (RET) Komutu:** Bu komut CALL komutu ile çağrılarak işletilen alt programdan ( Sabroutine sayfasında) ana programa koşullu veya koşulsuz olarak geri dönüşü sağlar. Koşulsuz olarak yazılan RET komutu her zaman alt programın son komutudur. Belli bir koşula bağlı yazılan RET komutu istenilen koşul oluştuğunda alt programın tamamının işletilmesini beklemeden ana programa dönüşü sağlar.

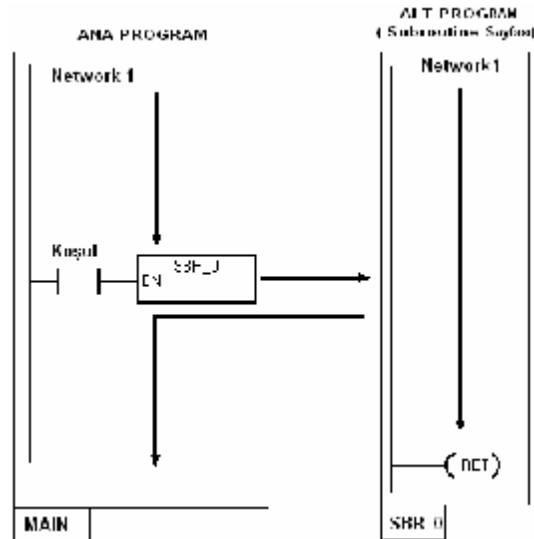
LADDER: —(RET) **Koşulsuz**



STL: **RET**


OPERANT: **Yok**

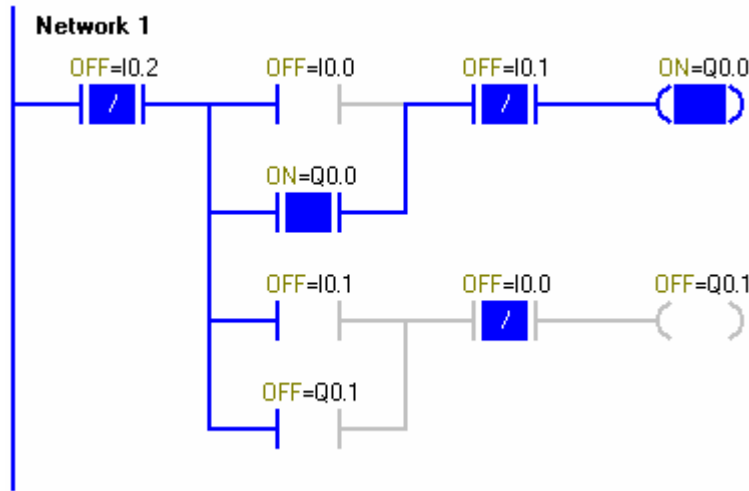
Şekilde ana program network 1 den itibaren işletilir. Alt program çağırma komutuna gelindiğinde koşulu belirten kontak enerji akışına izin veriyorsa program akışı SBR\_0 sayfasında bulunan alt programa geçer. Alt program sayfasındaki komutlar sırası ile işletilir. Alt programın son komutu olan RET komutunun işletilmesiyle program akışı alt program sayfasından ana program sayfasına geri döner ve alt program çağırma komutunun hemen akabindeki komuttan itibaren komutların işletilmesine devam edilir.



## 2.9. Yapılan Programa Göre Kullanılan Giriş ve Çıkışların Simülasyonu

Step 7 mic/Win 32 programının üst versiyonları ile hazırladığımız programın çalışmasını simülasyon şeklinde izleyerek projedeki hataları ve eksiklikleri kolayca bulabiliriz. Programın simülasyonunu görmek için öncelikle programın PLC hafızasına yüklenerek PLC'nin RUN moduna alınması gerekir.

PLC RUN modunda iken **Debug** menüsünden **Program status** seçeneğine tıklayarak ya da debug araç çubuğundan  ikonuna tıklamak suretiyle simülasyonu başlatabiliriz.



Şekilde bir PLC programının simülasyonu görülmektedir. Simülasyonda enerji akışının olduğu hatlar mavi renkle gösterilir. Enerji akışına izin veren kontakların içi mavi renkle boyalıdır ve adres kısımlarında ON ifadesi yer alır, diğerlerinde ise OFF ifadesi vardır. Ayrıca programda kullanılan zamanlayıcıların, sayıcıların anlık değerleri ile işlemsel komutların giriş ve çıkış değerleri gözlenebilir.

## UYGULAMA FAALİYETİ

Üç fazlı iki adet asenkron motor biri birinden bağımsız olarak ileri geri olarak çalışacaktır. Her iki motorda ileri ya da geri butonuna basıldığında 1 dk çalışıp duracaktır. Gerekli PLC programını ladder plan yöntemi ile yapınız?

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"><li>Ø Kontrol probleminin tanımlanması, ifade edilmesiyle kâğıda dökülmesi</li><li>Ø Sorunun çözümü için gerekli program veya fonksiyonların belirlenmesi</li><li>Ø Programın ladder diyagrama aktarılması</li><li>Ø 1. motorun ileri geri çalışması için gerekli ladder blokları kullanarak programı oluşturmak</li><li>Ø 2. motortun ileri geri çalışması için gerekli ladder bloklarını kullanarak programı oluşturmak</li><li>Ø Programın tamamlandıktan sonra gerekli kontrollerin yapılması</li></ul>	<ul style="list-style-type: none"><li>Ø Problemin tam anlaşılması için klasik kumanda devresini çiziniz.</li><li>Ø Atölyede bilgisayarınızı açarak programınız yüklü değilse öğretmeninizin gözetiminde CD den yükleyiniz.</li><li>Ø Problemin çözümü için Step-7 Micro/Win programını kullanınız.</li><li>Ø Ladder dilinde program komut sembollerinin doğru seçiniz.</li><li>Ø Çıkış işlemi yapılan komutların programda doğrudan enerji hattına bağlamayınız.</li><li>Ø Çıkış komutlarından önce mutlaka bir kontak kullanınız.</li><li>Ø Yaptıklarınızı rapor haline getirerek öğretmeninize sununuz.</li></ul>

## PERFORMANS DEĞERLENDİRME

DEĞERLENDİRME ÖLÇÜTLERİ		Evet	Hayır
1	Devrenin sinyal akış semasını doğru olarak çıkardınız mı?		
2	Devrenin PLC programını doğru olarak hazırlayabildiniz mi?		
3	Programı hatasız olarak PLC ye yüklediniz mi?		
4	Yapılan uygulamaya ait giriş ve çıkış elemanlarının simülasyonunu yaptınız mı?		



## UYGULAMA FAALİYETİ

Bir çıkış elemanı 1. butona basıldığında 1 dakika, 2. butona basıldığında 2.dakika, 3. butona basıldığında 3. dakika, 4 butona basıldığında 10 dakika çalışacak ve süre sonunda kendiliğinden duracaktır. Gerekli PLC programını ladder plan yöntemi ile yapınız?

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"><li>Ø Kontrol probleminin tanımlanması, ifade edilmesiyle kâğıda dökülmesi</li><li>Ø Sorunun çözümü için gerekli program veya fonksiyonların belirlenmesi</li><li>Ø Programın ladder diyagrama aktarılması</li><li>Ø Butonlara basıldığında zaman değerlerini zamanlayıcı için gerekli değer olarak atayacak çalışma için ladder bloklarını kullanarak programı oluşturmak</li><li>Ø Programın tamamlandıktan sonra gerekli kontrollerin yapılması</li></ul>	<ul style="list-style-type: none"><li>Ø Problemin tam anlaşılmasının için klasik kumanda devresini çiziniz.</li><li>Ø Atölyede bilgisayarınızı açarak programınız yüklü değilse öğretmeninizin gözetiminde CD den yükleyiniz</li><li>Ø Problemin çözümü için Step-7 Micro/Win programını kullanınız.</li><li>Ø Ladder dilinde program komut sembollerinin doğru seçiniz.</li><li>Ø Çıkış işlemi yapılan komutların programda doğrudan enerji hattına bağlamayınız.</li><li>Ø Çıkış komutlarından önce mutlaka bir kontak kullanınız.</li><li>Ø Yaptıklarınızı rapor haline getirerek öğretmeninize sununuz.</li></ul>

## PERFORMANS DEĞERLENDİRME

DEĞERLENDİRME ÖLÇÜTLERİ		Evet	Hayır
1	Devrenin sinyal akış semasını doğru olarak çıkardınız mı?		
2	Devrenin PLC programını doğru olarak hazırlayabildiniz mi?		
3	Programı hatasız olarak PLC'ye yüklediniz mi?		
4	Yapılan uygulamaya ait giriş ve çıkış elemanlarının simülasyonunu yaptınız mı?		

## ÖLÇME VE DEĞERLENDİRME

### A- OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki cümleleri doğru veya yanlış olarak değerlendiriniz.

- 1.( ) **STL**, klasik kumanda devrelerinin çizimlerine benzeyen grafiksel gösterime sahip bir PLC programlama dilidir.
- 2.( ) Programda kullanılan mutlak adreslere sembolik isimler vermek için **Data Block** penceresi kullanılır.
- 3.( ) **I0.3** bir giriş adresidir.
- 4.( ) Çıkış ( **Q** ) hafıza alanlarına sadece bit uzunluğunda erişim sağlanabilir.
- 5.( ) **VW0** adresinin yüksek değerlikli baytı **VB0** adresidir.
- 6.( ) Zamanlayıcılar 1ms, 10ms ve 100ms lik zaman sürelerini sayarlar.
- 7.( ) Sayıcılar **T0,T1,T2...** şeklinde adreslenir.
- 8.( ) Zamanlayıcı ve sayıcılarda **PV** ayar girişine yazılacak operant bayt uzunluğunda olmalıdır.
- 9.( ) **LD** komutu normalde kapalı bir kontakla hat başlatmak için kullanılır.
- 10.( ) **S** (SET) komutu ile aktif (lojik 1) yapılan hafıza bitlerini pasif (lojik 0) hâle getirmek için **R** (RESET ) komutu kullanılmalıdır.
- 11.( ) Alt programlar ( Subroutine) **SBR\_0, SBR\_1....** sayfalarına yazılır.

Aşağıda verilen cümlelerin tamamlanması için boş bırakılan yerlere uygun kelimeleri ve rakamları yazınız.

1. .... komutları IN girişlerinde bulunan operant içeriğini OUT çıkışında bulunan operant alanına taşırlar.
2. .... enerjili kaldığı sürelerin tamamını sayan zamanlayıcıdır.
3. Zaman tabanı 10 ms olan bir zamanlayıcı ile 12 sn zaman geçikmesi elde etmek için **PV** değeri ..... olmalıdır.
4. Birden fazla komut bloğunu paralel bağlamak için .....komutu kullanılır.
5.  $(01101011)_2$  binary sayısının desimal karşılığı (.....)<sub>10</sub> dir.

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete dönerek tekrar inceleyiniz.

Tüm sorulara doğru cevap verdiyseniz diğer faaliyete geçiniz.

## ÖLÇME VE DEĞERLENDİRME

### A- OBJEKTİF TESTLER (ÖLÇME SORULARI)

Aşağıdaki verilmiş olan cümleleri doğru veya yanlış olduklarını tespit ederek işaretleyiniz.

1. PLC ile iletişim sağlamak için bilgisayarın seri portu kullanılır.
2. Bilgisayarla PLC arasında haberleşmenin sağlanması için için PC / PPI master kablosuna ihtiyaç vardır.
3. S7 - 200 ile bilgisayar arasında iletişim hızı 20 kpbs dur.
4. Bilgisayar seri portuna bağlantı için RS 485 konnektörü kullanılır.
5. Kontrol programının PLC'ye yüklenebilmesi için **UPLOAD** seçeneği kullanılır.

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete dönerek tekrar inceleyiniz.

Tüm sorulara doğru cevap verdiyseniz diğer faaliyete geçiniz.

# ÖĞRENME FAALİYETİ-3

## AMAÇ

PLC cihazına yüklenen programı bilgisayar ortamına çekebilecek, yedeklediğiniz programı sistemin yazılımından kaynaklanan arıza durumlarında tekrar yükleyebileceksiniz.

## ARAŞTIRMA

- Ø PLC’de yapılan bir programı bilgisayara nasıl yedeklendiği hakkında internet ortamından ve kütüphanelerden araştırma yaparak rapor haline getiriniz. Sınıfta sununuz.

## 3. PROGRAM YEDEKLEME

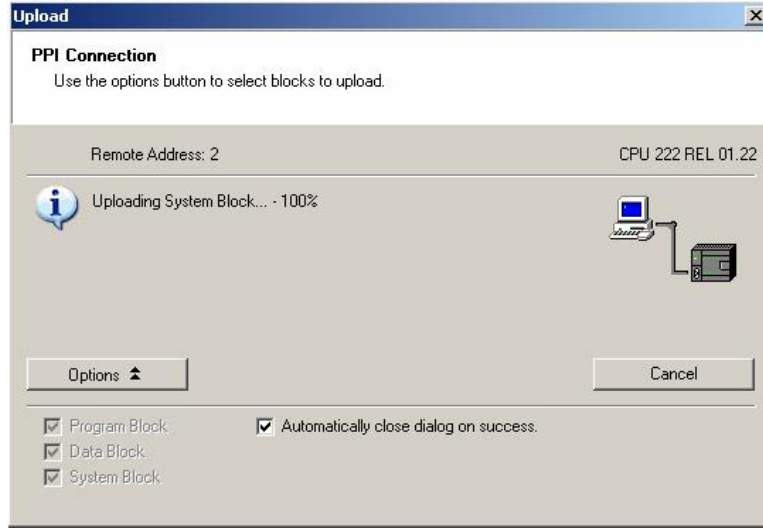
### 3.1. Program Yedeklemenin Önemi

Yedeklemenin önemini şöyle sıralayabiliriz.

- Ø PLC için yazılmış kontrol programları küçük değişikliklerle başka sistemler için kullanılabilir.
- Ø PLC arızası durumlarında hafızasında bulunan program silinirse yeniden program yazmadan yedeklenen program yüklenebilir.
- Ø Değişik amaçlar için yapılmış PLC programlar birleştirilerek değişik programlar yapılabilir.

### 3.2. PLC’deki Programın Bilgisayara Çekilmesi

S7-200 PLC’nin program hafızasına yüklenmiş olan kontrol programını programlama editörüne çekmek için öncelikle PC / PPI kablosu bağlantısı ile PLC ile bilgisayar arasındaki iletişimi sağlamanız gerekmektedir. Daha sonra File menüsünden Upload seçeneğine ya da standart araç çubuğundaki Upload ikonuna tıklanarak Şekil 3.1’de görülen Upload iletişim kutusunun ekrana gelmesi sağlanır. Options düğmesine tıklayarak ve projenin programlama editörüne çekmek istediğiniz bileşenlerin seçilmelidir. Upload düğmesine tıklayarak program PLC ‘den editöre çekilmiş olur. Çekmiş olduğunuz programı bilgisayarınızın harddiskine yedekleyebilirsiniz.



**Şekil 3.1: Upload iletişim kutusu**

## UYGULAMA FAALİYETİ

- Ø PLC cihazına yüklü bulunan herhangi bir programı PLC editörü kullanarak bilgisayara kaydediniz. Arşivleme için gerekli bilgileri ile birlikte CD veya disket ortamına kaydediniz.

İşlem Basamakları	Öneriler
<p>Ø PLC ile bilgisayar arasındaki bağlantıyı kurunuz.</p> <p>Ø PLC programlama editöründen haberleşme için gerekli ayarları yapınız.</p> <p>Ø Programı PLC'den bilgisayara alınız</p> <p>Ø Programı CD ya da disket ortamına kaydediniz.</p> <p>Ø Gerekli arşivleme bilgilerini kayıt ortamı üzerine etiketleyiniz.</p>	<p>Ø PLC'den program çekerken, marka ve modelini tespit ediniz.</p> <p>Ø Tesbit ettiğiniz marka ve modele uygun yazılımı temin ederek bilgisayarınıza yükleyiniz.</p> <p>Ø Tesbit ettiğiniz PLC marka ve modeline göre haberleşme kablosunu temin ediniz.</p> <p>Ø Arşivleme bilgilerini hatasız ve okunaklı olarak etiketleyiniz.</p>

## PERFORMANS DEĞERLENDİRME

DEĞERLENDİRME ÖLÇÜTLERİ		Evet	Hayır
1	Bağlantıyı doğru olarak kurdunuz mu?		
2	PLC programlama editöründen haberleşme için gerekli ayarları yaptınız mı?		
3	Programı PLC den bilgisayara kaydedebildiniz mi?		
4	Programı CD veya disket ortamına kaydettiniz mi?		
5	Arşivleme için gerekli bilgileri etiketlediniz mi?		



# MODÜL DEĞERLENDİRME

Aşağıda verilen cümlelerin tamamlanması için boş bırakılan yerlere uygun kelimeleri ve rakamları yazınız?

1. **STL**, klasik kumanda devrelerinin çizimlerine benzeyen .....gösterime sahip bir PLC programlama dilidir.
2. Programda kullanılan mutlak adreslere sembolik isimler vermek için .....penceresi kullanılır.
3. **I0.3** bir .....adresidir.
4. Çıkış ( **AC** ) hafıza alanlarına sadece ..... uzunluğunda erişim sağlanabilir.
5. **VW0** adresinin yüksek değerlikli baytı ..... adresidir.
6. Zamanlayıcılar 1ms, 10ms ve ..... lik zaman sürelerini sayarlar.
7. .... **T0,T1,T2**... şeklinde adreslenir.
8. Zamanlayıcı ve sayıcılarda **PV** ayar girişine yazılacak operant..... uzunluğunda olmalıdır.
9. **LD** komutu normalde .....bir kontakla hat başlatmak için kullanılır.
10. **S** (SET) komutu ile aktif (lojik 1) yapılan hafıza bitlerini pasif (lojik 0) hâle getirmek için ..... komutu kullanılmalıdır.
11. .... **SBR\_0, SBR\_1**.... sayfalarına yazılır.
12. PLC ile iletişim sağlamak için bilgisayarın ..... kullanılır.
13. Bilgisayarla PLC arasında haberleşmenin sağlanması için için ..... master kablosuna ihtiyaç vardır.
14. S7 - 200 ile bilgisayar arasında iletişim hızı ..... kpbs dur.
15. Bilgisayar seri portuna bağlantı için ..... konnektörü kullanılır.
16. Kontrol programının PLC'ye yüklenebilmesi için ..... seçeneği kullanılır.

Aşağıdaki soruların cevaplarını doğru veya yanlış olarak değerlendiriniz.

- 17.( ) Move komutları **IN** girişlerinde bulunan operant içeriğini **OUT** çıkışında bulunan operant alanına taşırlar.
- 18.( ) **TOFF** enerjili kaldığı sürelerin tamamını sayan zamanlayıcıdır.
- 19.( ) Zaman tabanı 10 ms olan bir zamanlayıcı ile 12 sn zaman geçikmesi elde etmek için **PV** değeri 12000 olmalıdır.
- 20.( ) Birden fazla komut bloğunu paralel bağlamak için **OLD** komutu kullanılır.
- 21.( )  $(01101011)_2$  binary sayısının desimal karşılığı  $(107)_{10}$  dir.

## DEĞERLENDİRME

Yaptığınız değerlendirme sonucunda eksikleriniz varsa öğrenme faaliyetlerini tekrarlayınız.

Modülü tamamladınız, tebrik ederiz. Öğretmeniniz size çeşitli ölçme araçları uygulayacaktır. Öğretmeninizle iletişime geçiniz.

# CEVAP ANAHTARLARI

## ÖĞRENME FAALİYETİ-1 CEVAP ANAHTARI

1	YANLIŞ
2	YANLIŞ
3	DOĞRU
4	YANLIŞ
5	DOĞRU
6	DOĞRU
7	YANLIŞ
8	YANLIŞ
9	YANLIŞ
10	DOĞRU
11	DOĞRU

1	MOVE
2	TONR
3	1200
4	OLD
5	107

## ÖĞRENME FAALİYETİ-2 CEVAP ANAHTARI

1	DOĞRU
2	DOĞRU
3	YANLIŞ
4	YANLIŞ
5	YANLIŞ

## MODÜL DEĞERLENDİRME CEVAPLARI

1	GRAFİKSEL
2	SEMBOL ADRES
3	GİRİŞ
4	DOUBLE WORD
5	VB0
6	100 ms
7	ZAMANLAYICI
8	WORD
9	AÇIK
10	RESET
11	SOUBROUTINE
12	SERİ PORT
13	PC/ PPI
14	9.6 kbps
15	RS 232
16	DOWNLOAD
17	DOĞRU
18	YANLIŞ
19	YANLIŞ
20	DOĞRU
21	DOĞRU

Cevaplarınızı cevap anahtarları ile karşılaştırarak kendinizi değerlendiriniz.

## ÖNERİLEN KAYNAKLAR

- Ø FESTO firmasının katalogları
- Ø SIEMENS firmasının katalogları
- Ø Hizmetiçi eğitim PLC kursu ders notları
- Ø [www. Programlama.com](http://www.Programlama.com) internet adresi
- Ø [www. Siemens.com](http://www.Siemens.com) internet adresi

## KAYNAKÇA

- Ø **SIMATIC S7 – 200 Kullanım Kılavuzu**, Siemens, Ağustos, 2003.
- Ø **SIMATIC S7 – 200 Programmable Controller**, System Manuel Siemens.
- Ø Mesleki Eğitim ve Teknoloji Merkezi (2002), **Programlanabilir Lojik Kontrolör Kullanım Kılavuzu**.
- Ø ÜSTÜN Behçet, **İleri Kumanda Teknikleri 1**, Bursa, 2001.
- Ø ÇETİN Recep, **S7 – 200 PLC’lerle Otomasyon**, Ankara, 2004.