

EGE ÜNİVERSİTESİ

EGE MYO

MEKATRONİK PROGRAMI

- List Programlarına Genel Bakış
- List Buyruklarının Çalışması
- List Dili Buyrukları
- Parantezlerin Kullanımı
- Yiğın Buyrukları (MPS, MRD, MPP)

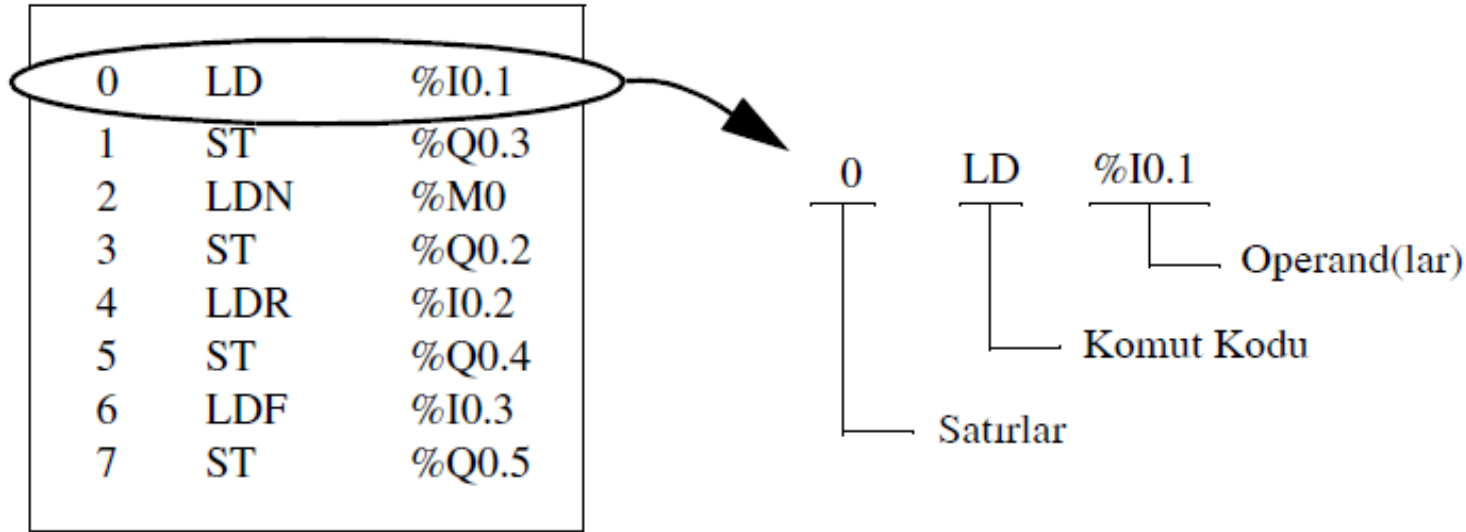
BUYRUK LİST DİLİ

List Programlarına Genel Bakış

- **Açıklama** List dilinde yazılmış bir program, denetleyici tarafından ardışıl olarak icra edilen bir buyruk dizisinden oluşur. Her bir List buyruğu, tek bir program satırı ile temsil edilir ve üç öğeden oluşur:
 - Satır Numarası (Line Number)
 - Buyruk Kodu
 - İşlenen(ler)

Bir List Program Örneği

- Aşağıdaki, bir List program örneğidir.



- Satır Numarası** Satır numaraları, bir buyruk girildiğinde otomatik olarak üretilir. Boş satırlar ve Açıklama satırlarının numarası yoktur.
- Buyruk Kodu** Buyruk kodu, işlenen(ler) kullanılarak gerçekleştirilecek işlemi tanılayan bir operatör için bir semboldür. Tipik operatörler, Boole ve nümerik işlemleri belirtir.

- Örneğin, yukarıdaki örnek programda, LD, LOAD buyruğunun kısaltmasıdır.
- LOAD komutu, %I0.1 işleneninin değerini, akümülatör denen bir dahili yazaca yükler.
- Temelde iki tip kod vardır:
 - Test komutları
 - Bunlar, bir aksiyonu gerçekleştirmek için gerekli koşulları kurar veya test eder.
 - Örneğin, LOAD (LD) ve AND.
 - Aksiyon komutları
 - Bunlar setup koşullarının sonuçlarına göre, aksiyonları gerçekleştirir. Örneğin, STORE (ST) ve RESET (R) gibi atama komutları.

- **İşlenen** İşlenen, programın bir buyruk içerisinde manipüle edebileceği (kendi çıkarları doğrultusunda değiştirebileceği) bir numara, adres veya semboldür. Örneğin, yukarıdaki örnek programda, %I0.1 işleneni, denetleyicinin bir girişinin değerine atanmış bir adrestir. Bir buyruk, buyruk kodunun tipine bağlı olarak, 0'dan 3 adede kadar işlenene sahip olabilir.
- İşlenenler, aşağıdakileri temsil edebilir:
 - Sensörler, basma butonları ve röleler gibi denetleyici giriş ve çıkışları.
 - Zamanlayıcılar ve sayıcılar gibi ön-tanımlı sistem fonksiyonları.
 - Aritmetik, mantık, karşılaştırma ve nümerik işlemler.
 - Bitler ve sözcükler gibi denetleyici dahili değişkenleri.

List Komutlarının Çalışması

- **Açıklama** List buyrukları, yalnız bir açık (explicit) işlenene sahiptir, diğer işlenen dolaylıdır.
- Dolaylı işlenen, Boole akümülatörün değeridir. Örneğin, LD %I0.1 buyruğunda, %I0.1, açık işlenendir. Dolaylı işlenen, akümülatörde saklanmaktadır ve %I0.1'in değeri, akümülatörün değeri olur.
- **Çalışma** Bir List buyruğu, akümülatörün ve açık işlenenin içeriklerine bağlı olarak belirli bir işlem gerçekleştirir ve akümülatörün içeriğini sonuçla değiştirir. Örneğin, AND %I1.2 işlemi, akümülatörün içeriği ve 1.2 girişi arasında bir mantık AND işlemi gerçekleştirir ve akümülatörün içeriğini, bu sonuçla değiştirir.

- Bütün Boole komutları (Load, Store ve Not haricinde), iki işlenenle çalışır. Bu iki işlenenin değeri, Doğru (True) veya Yanlış (False) olabilir ve programın icrası, tek bir değer üretir; Doğru veya Yanlış. Load (Yükleme) buyrukları, işlenenin değerini, akümülatörde saklı tutar, Store komutları, akümülatörün değerini, işlenene transfer eder. Not komutunun, açık işleneni yoktur ve akümülatörün durumunu evriğine (tersine) çevirir.
- **Desteklenen List Komutları**
- Aşağıdaki tablo, desteklenen List buyruk tiplerinin bir özetidir.



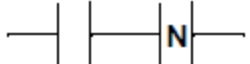
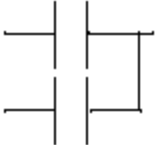
Komut Tipi	Örnek	Fonksiyon
Bit komutu	LD %M10	%M10 dahili bitini okur
Blok komutu	IN %TM0	%TM0 zamanlayıcısını başlatır
Word komutu	[%MW10 := %MW50+100]	Toplama işlemi
Program komutu	SR5	Altrutin #5'i çağırır
Grafcet komutu	-* -8	Adım #8

List Dili Buyrukları

- **Açıklama** List dili, aşağıdaki buyruk tiplerini içerir:
 - Test buyrukları
 - Aksiyon buyrukları
 - Fonksiyon bloğu buyrukları
- Bu kısım, List programlama için Twido buyrukları tanımlar ve tanımlar.

- **Test Buyrukları** Aşağıdaki tablo, List dilindeki test buyruklarını tanımlamaktadır.

Ad	Eşdeğer grafik eleman	Fonksiyon
LD		Bool sonuç, operandın durumu (status) ile aynıdır.
LDN		Bool sonuç, operandın durumunun evriği (değili) ile aynıdır.
LDR		Bool sonuç, operandın 0'dan 1'e (yükselen kenar) geçişinin saptanması ile 1 olur.
LDF		Bool sonuç, operandın 1'den 0'a (düşen kenar) geçişinin saptanması ile 1 olur.
AND		Bool sonuç, bir önceki komutun Bool sonucu ile operandın durumu arasındaki AND lojiğine eşittir.

ANDN		Bool sonuç, bir önceki komutun Bool sonucu ile operandın durumunun evriği arasındaki AND lojiğine eşittir.
ANDR		Bool sonuç, bir önceki komutun Bool sonucu ile operandın yükselen kenarının saptanması (1= yükselen kenar) arasındaki AND lojiğine eşittir.
ANDF		Bool sonuç, bir önceki komutun Bool sonucu ile operandın düşen kenarının saptanması (1= düşen kenar) arasındaki AND lojiğine eşittir.
OR		Bool sonuç, bir önceki komutun Bool sonucu ile operandın durumu arasındaki OR lojiğine eşittir.

Ad	Eşdeğer grafik eleman	Fonksiyon
AND(Lojik AND (8 parantez düzeyi (level))
OR(Lojik OR (8 parantez düzeyi)
XOR, XORN, XORR, XORF		Exclusive OR
MPS MRD MPP		Bobinlere geçme (switching)
N	-	Evrikleme (NOT)

Aksiyon Buyrukları

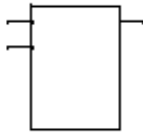
- Aşağıdaki tablo, List dilindeki aksiyon buyruklarını tanımlamaktadır.

Ad	Eşdeğer grafik eleman	Fonksiyon
ST	—()—	İlgili operand, test bölgesi sonucunun değerini alır.
STN	—(/)—	İlgili operand, test bölgesi sonucunun evriğinin değerini alır.
S	—(S)—	İlgili operand, test bölgesi sonucu 1 ise, 1'e set edilir.
R	—(R)—	İlgili operand, test bölgesi sonucu 1 ise, 0'a set edilir.
JMP	->>%Li	Koşulsuz olarak, akış-yukarı veya akış-aşağıdaki etiketlenmiş bir satıra gidilir.

SRn	->>%SRi	Numarası verilmiş bir altrutine gidilir.
RET	<RET>	Bir altrutinden dön (return).
END	<END>	Program sonu.
ENDC	<ENDC>	Bool sonucun 1 olması durumunda, koşullu bir programın sonu.
ENDCN	<ENDCN>	Bool sonucun 0 olması durumunda, koşullu bir programın sonu.

Fonksiyon Bloğu Buyrukları

- Aşağıdaki tablo, List dilindeki fonksiyon bloklarını tanımlamaktadır.

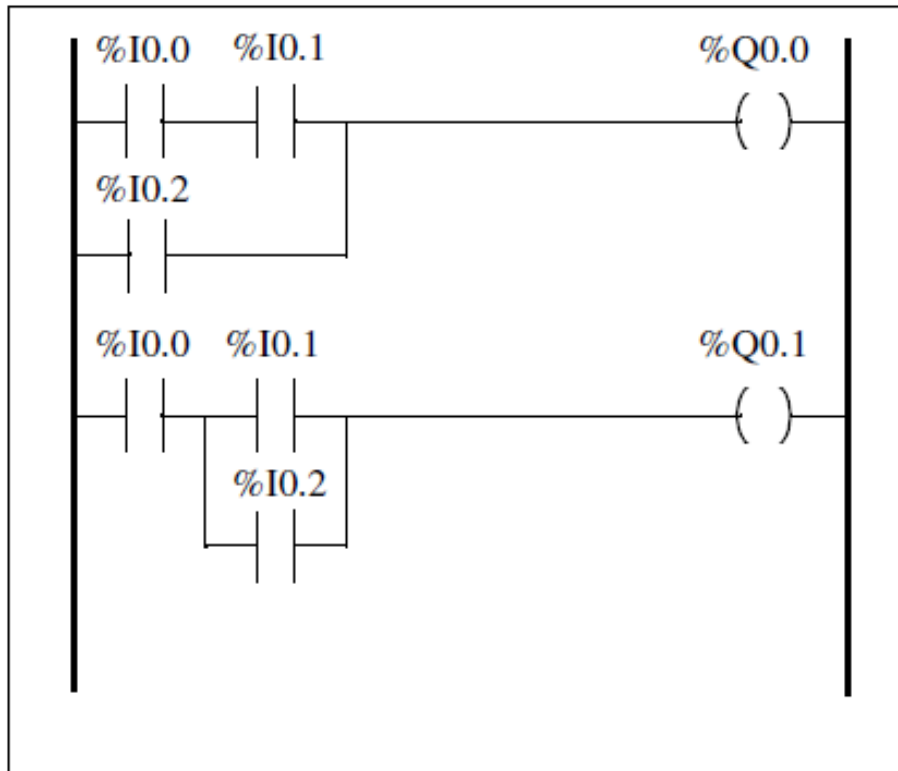
Ad	Eşdeğer grafik eleman	Fonksiyonlar
Zamanlayıcılar, sayıcılar, register'lar, vb.		Her bir fonksiyon bloğu için, bloğu kontrol etmeye yarayan komutlar bulunmaktadır. Blok giriş ve çıkış'larının doğrudan bağlantısını yapmak için, yapılandırılmış bir form kullanılır. Not: Fonksiyon bloklarının çıkışları, birbirine bağlanamaz (düşey kısa devreler).

Parantezlerin Kullanımı

- **Açıklama** Ladder diyagramlarındaki paralel kollarını (branches) işaret etmek üzere, AND ve OR mantık buyruklarında parantezleri kullanınız. Parantez açma ve kapama, buyruklar ile aşağıdaki gibi ilişkilendirilmiştir:
 - Parantez açma, AND ve OR buyruğuyla ilişkilendirilmiştir.
 - Parantez kapama, her bir açık parantez için gerekli bir buyruktur.

AND Buyruğunu Kullanan Örnek

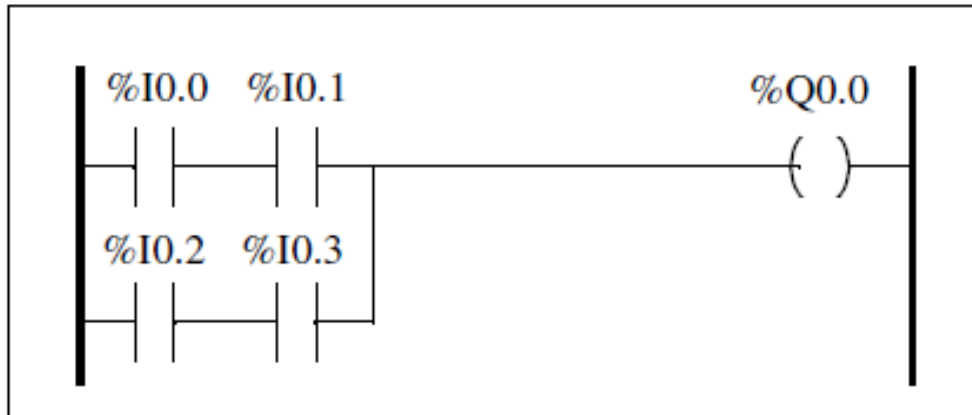
- Aşağıdaki diyagramlar, AND buyruğunun parantez ile kullanımına örnek olarak verilmiştir: AND(...).



```
LD    %I0.0
AND   %I0.1
OR    %I0.2
ST    %Q0.0
```

```
LD    %I0.0
AND(  %I0.1
OR    %I0.2
)
ST    %Q0.1
```

- **OR Buyruğunu Kullanan Örnek**
- Aşağıdaki diyagramlar, OR buyruğunun parantez ile kullanımına örnek olarak verilmiştir: OR(...).



```
LD    %I0.0
AND   %I0.1
OR(   %I0.2
AND   %I0.3
)
ST    %Q0.0
```

- **Değiştiriciler (Modifiers)** Aşağıdaki tablo, parantezlere uygulanabilen değiştiricileri listelemektedir.

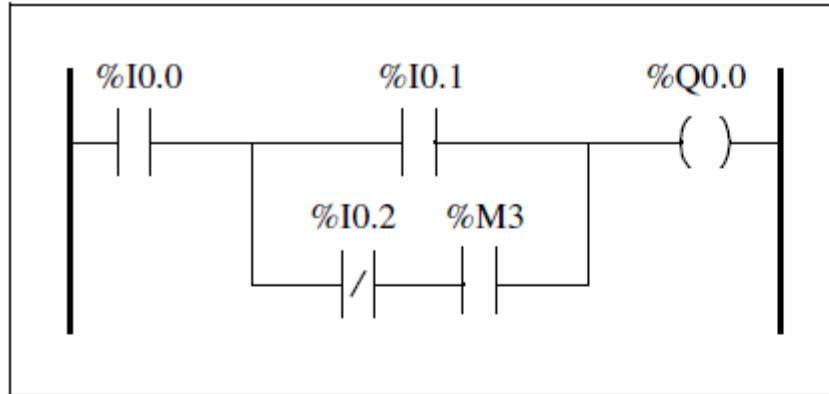
Değiştirici	Fonksiyon	Örnek
N	Evrikleme (Negation)	AND(N veya OR(N
F	Düşen kenar	AND(F veya OR(F
R	Yükselen kenar	AND(R veya OR(R
[Karşılaştırma	Bkz. <i>Karşılaştırma Komutları</i> , s. 236

İççe Parantezler

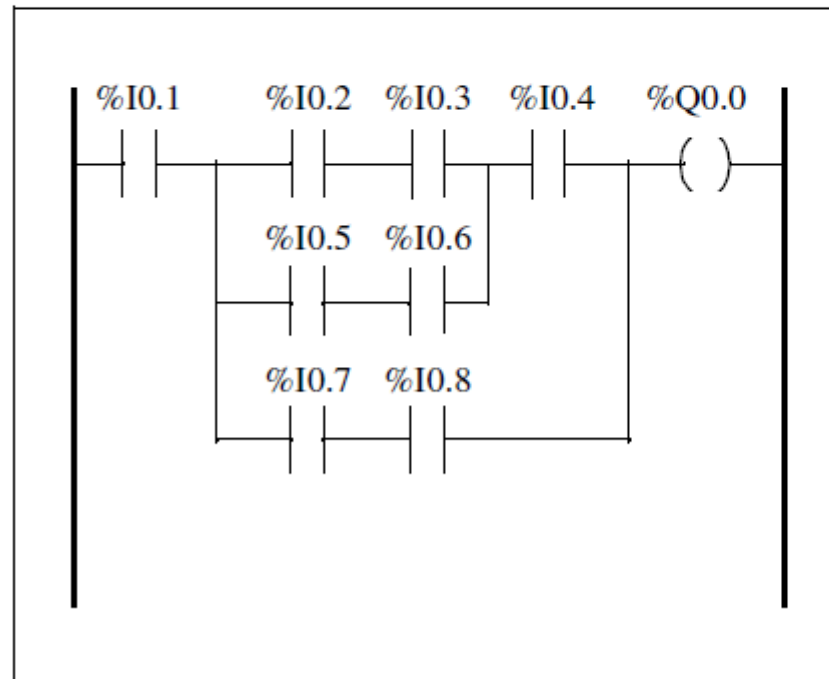
- İççe sekiz adede kadar parantez kullanmak mümkündür.
- İççe parantezler kullanılırken, aşağıdaki kurallara uyulmalıdır:
 - Her bir açık paranteze karşılık bir kapalı parantez bulunmalıdır.
 - Etiketler (%Li:), altyordamlar (SRi:), jump (atlama) buyrukları (JMP) ve fonksiyon bloğu buyrukları, parantezler arasındaki ifadelerin içine yerleştirilmemelidir.
 - Store (saklama) buyrukları ST, STN, S ve R, parantezler arasında kullanılamaz.
 - Yığın buyrukları MPS, MRD ve MPP, parantezler arasında kullanılamaz.

• İççe Parantez Örnekleri

- Aşağıdaki diyagramlar, iççe parantez örnekleridir.



```
LD    %I0.0
AND(  %I0.1
OR(N  %I0.2
AND   %M3
)
)
ST    %Q0.0
```



```
LD    %I0.1
AND(  %I0.2
AND   %I0.3
OR(   %I0.5
AND   %I0.6
)
AND   %I0.4
OR(   %I0.7
AND   %I0.8
)
)
ST    %Q0.0
```

Yığın Komutları (MPS, MRD, MPP)

- **Açıklama** Yığın buyrukları, bobinlere (çıkışlara) yönlendirmeyi gerçekleştirmek için kullanılır.
- MPS, MRD ve MPP buyrukları, 8 adet Boole ifadesine kadar depolama yapabilen yığın (stack) adı verilen geçici bir depolama alanı kullanır.

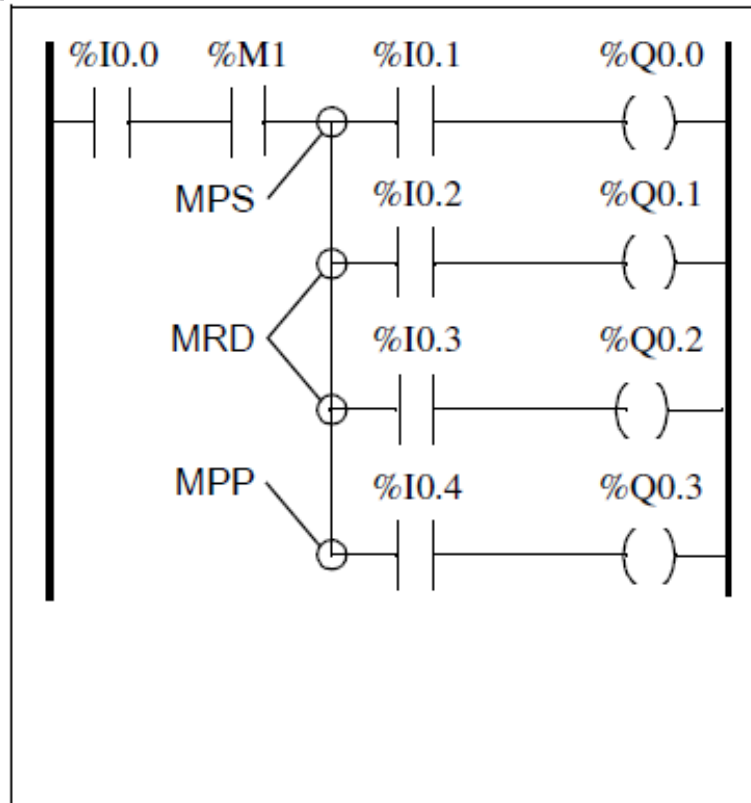
Not: Bu buyruklar, parantezler arasındaki ifadeler içinde kullanılamaz.

Yığın Komutlarının Çalıştırılması

- Aşağıdaki tablo, üç yığın buyruğunun çalışmasını tanımlamaktadır.

Komut	Tanım	Fonksiyon
MPS	Yığına değer yazımı (Memory Push onto stack)	En son lojik komutun değerini (akümülatörün içeriği) yığının en tepesine koyar (push) ve diğer değerleri, yığının altına doğru kaydırır.
MRD	Yığından değer okunması	Yığının en tepesindeki değeri, akümülatöre okur.
MPP	Yığından değer alımı (Memory Pop from stack)	Yığının en tepesindeki değeri, akümülatöre kopyalar (pop) ve diğer değerleri, yığının tepesine doğru kaydırır.

- **Yığın Buyruklarına Örnekler**
- Aşağıdaki diyagramlar, yığın buyruklarını kullanan örneklerdir.



LD	%I0.0
AND	%M1
MPS	
AND	%I0.1
ST	%Q0.0
MRD	
AND	%I0.2
ST	%Q0.1
MRD	
AND	%I0.3
ST	%Q0.2
MPP	
AND	%I0.4
ST	%Q0.3

- **Yığın Çalışmasına Örnekler**
- Aşağıdaki diyagramlar, yığın buyruklarının nasıl çalıştığını gösterir.

