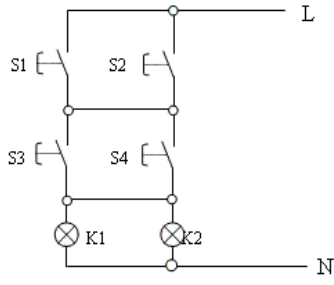


Örnek : Aşağıdaki devreyi PLC için programlayınız.

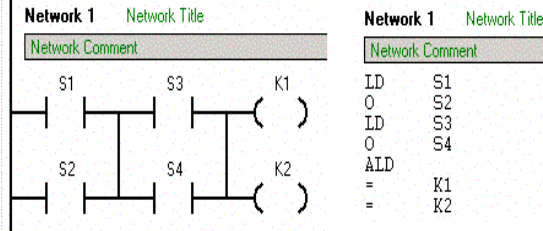


Çözüm :

Bağlantı planlarında verilen sembolik ifadelerle PLC mutlak adresleri sembol tablolarında ilişkilendirilmelidir. Bu nedenle aşağıdaki sembol tablosu oluşturulmalıdır.

View	Symbol	Address
Program Block	S1	I0.1
	S2	I0.2
	S3	I0.3
	S4	I0.4
Symbol Table	K1	Q0.1
	K2	Q0.2

Ladder

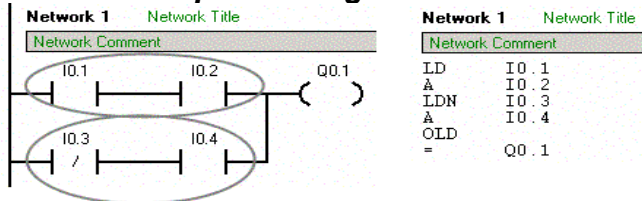


STL

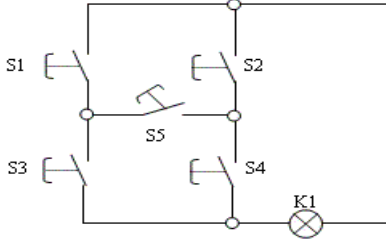
“ALD “ Komutu LD ile başlayan iki program parçasını birbirine seri olarak bağlar.

2- PARALEL BLOK BİRLEŞTİRME İŞLEMİ (OLD) :

İki bloğun birbirine paralel bağlanmasında ‘OLD’ (Or Load) komutu kullanılır. Başka bir ifadeyle OLD komutu, LD veya LDN ile başlayan program bloklarını birbirine paralel bağlar.



Örnek 1 : Aşağıdaki devreyi PLC için programlayınız.



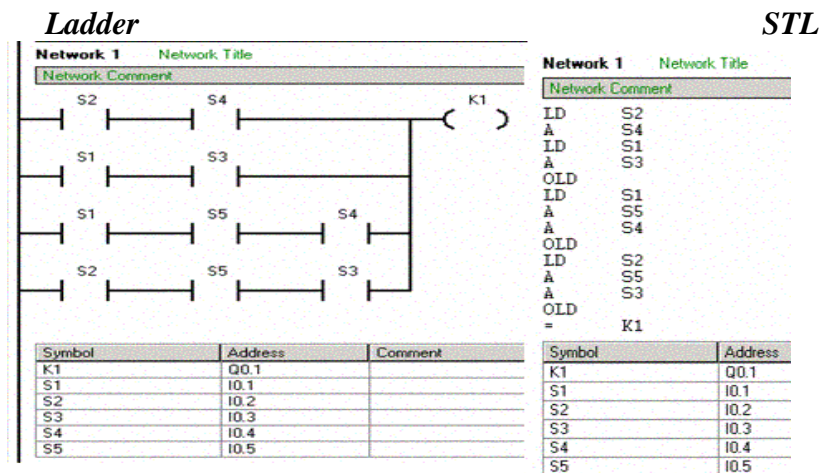
Birden fazla kontağın seri veya paralel bağlanması sonucunda blok meydana gelir. Meydana gelen bloklar da kendi aralarında seri veya paralel bağlı olabilirler. Birbirine seri bağlı iki blok ALD komutu ile birbirine bağlanır. Bunun gibi birbirine paralel bağlı iki blok için ise OLD komutu kullanılır. Bu şekilde blokları birbirine bağlayan özel komutlar her PLC bünyesinde bulunmayabilir.

K1 çıkışının aktif olabilmesi için değişik ihtimaller mevcuttur. Bu ihtimalleri şu şekilde sıralayabiliriz :

- **K1 çıkışı S2 ve S4 giriş elemanlarının her ikisinin de birlikte uyarılması durumunda aktif olur. Yani $K1 = S2 \times S4$**
- **K1 çıkışı S1 ve S3 giriş elemanlarının her ikisinin de birlikte uyarılması durumunda aktif olur. Yani $K1 = S1 \times S3$**
- **K1 çıkışı S1 ve S5 ve S4 giriş elemanlarının birlikte uyarılması durumunda aktif olur. Yani $K1 = S1 \times S5 \times S4$**
- **K1 çıkışı S2 ve S5 ve S3 giriş elemanlarının birlikte uyarılması durumunda aktif olur. Yani $K1 = S2 \times S5 \times S3$**

Tüm ihtimalleri bir arada yazdığımızda aşağıdaki gibi bir sonuç oluşur :

$$K1 = S2 \times S4 + S1 \times S3 + S1 \times S5 \times S4 + S2 \times S5 \times S3$$



SEMBOLİK ADRESLEME :

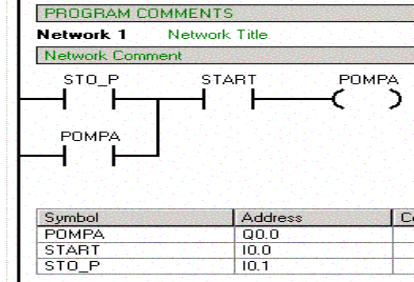
Sembolik adresleme CPU'daki adrese daha anlaşılır ve açıklayıcı bir metin (sembol) tanımlama işlemidir. Böylece pratik olarak takibi zor olan mutlak adreslemenin yerine(I0.1, Q0.3, T38, v.b.) programın izlenmesini

kolaylaştıran gerçek isimler yerleştirilmiş olur (start, pompa motoru, zamanlayıcı v.b.).

Sembolik programlama için önceden bir sembol tablosu oluşturmak gerekir. Genel anlamda bu tablonun düzenlenmesi şu şekildedir:

View		Symbol		Address	
1	START	10.0			
2	STO_P	10.1			
3	POMPA	Q0.0			
4					
5					

Sembolik programlama örneği :

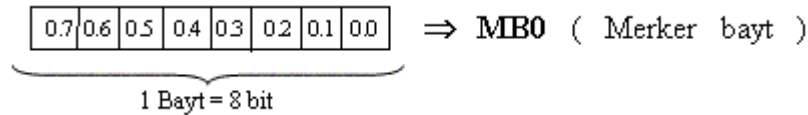


YARDIMCI RÖLE (M)

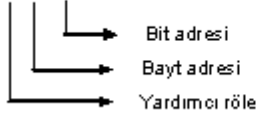
Yardımcı rölelere **durum tespit işaretçileri** de denilmektedir. Durum tespit işaretçileri, elektronik hafıza birimleri olup bir sinyalin durumunu **0** veya **1** olarak saklanmasını sağlarlar.

PLC programlarının yazılımı sırasında uzun ve karmaşık fonksiyonlar bir kargaşa oluşturmaktadır. Fonksiyonda parantez seviyeleri çoğaldıkça programı anlamak zorlaşmaktadır. Bu durumda ara sonuçları atayabileceğimiz durum tespit işaretlerine (yardımcı röle) ihtiyaç vardır (M).

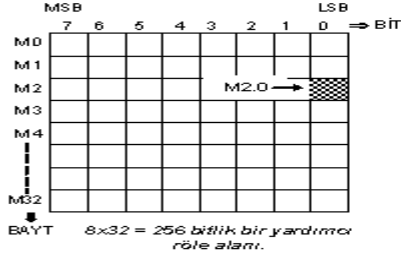
- Yardımcı röleler bit, byte (bayt) veya word olarak adreslenebilirler.



M 0.2 şeklindeki yardımcı röle ifadesinin anlamı şöyledir :



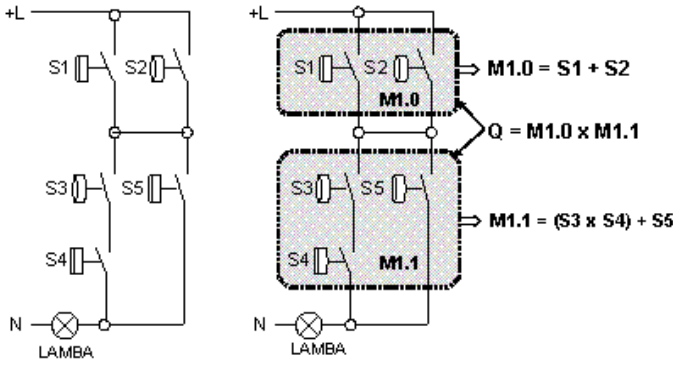
M 0.2 ifadesi; sıfıncı baytın ikinci biti olarak anlaşılmalıdır. Örneğin S7-200 CPU 214 PLC için 0.0 ilâ 31.7 arasında olmak üzere toplam 32 adet yardımcı röle için ayrılmış bayt alanı bulunmaktadır. Her bir bayt içinde 0-7 arasında toplam 8 adet yardımcı röle bulunduğu göre $32 \times 8 = 256$ adet yardımcı röle bulunmaktadır. Kullanıcı bunlardan herhangi birini seçmekte serbesttir.



- **Set** ve **reset** komutları ile PLC içerisinde hafıza birimi olarak da çalışabilirler.
- Her PLC' de değişik sayılarda yardımcı röle bulunur. Yardımcı röleler hem normalde kapalı hem de normalde açık kontaklara sahiptirler.
- PLC'nin 'stop' konumuna getirilmesiyle veya elektrik kesilmesi sonrası durum tespit işaretlerindeki bilgilerin ne olacağı kullanılan PLC cihazına bağlıdır.

- Durum tespit işaretçileri (yardımcı röle) bir kuvvetlendiriciye sahip olup, harici yüklerin sürülmesinde mutlaka bir çıkış değişkenine atanmaları gerekmektedir. *Çıkış yüklerinin direkt olarak sürülmesinde kullanılmazlar.*
- Durum tespit işaretlerinin oluşturulmasında yardımcı röleler kullanılır. Bunun için PLC içinden bir yardımcı röle seçmeliyiz. S7-200 PLC içinde yardımcı röleler **M** harfi ile ifade edilirler. Başk bir PLC için bu adres alanı değişik bir harf veya rakamla ifade edilebilir. Bunun için ilgili kullanım klavuzundan yararlanılmalıdır.

Örnek : Elektrik bağlantı plânı verilen devreyi yardımcı rölelerden yararlanarak PLC için programlayınız.



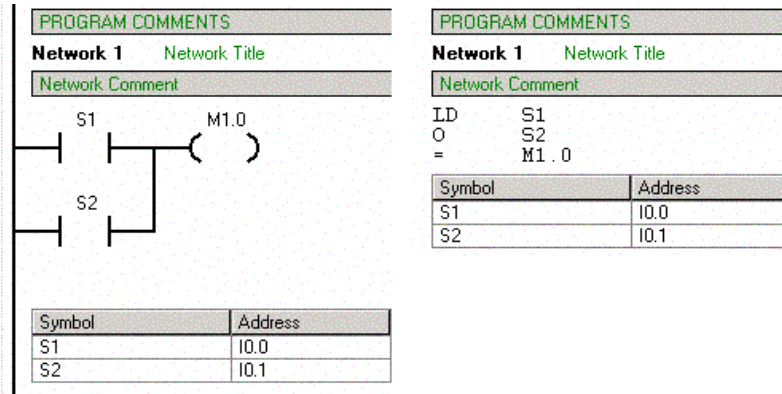
Programlamanın daha anlaşılır olması bakımından mutlak adresler yerine sembolik adresleme şekli kullanılacaktır. Bunun için sembolik ifadelerin mutlak adreslerle ilişkilendirilmiş olması gerekir.

Sembol tablosu

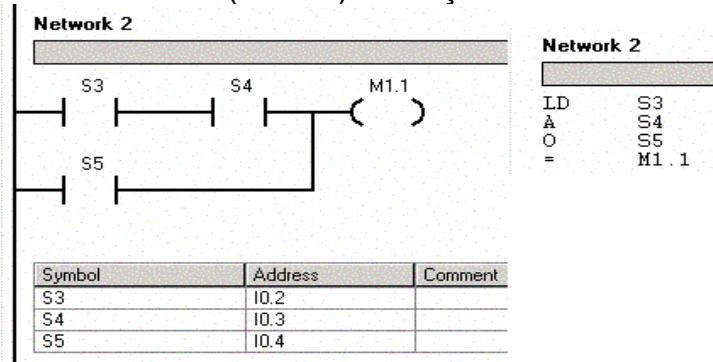


		Symbol	Address
1		S1	I0.0
2		S2	I0.1
3		S3	I0.2
4		S4	I0.3
5		S5	I0.4
6		LAMBA	Q0.0
7			

Network 1 // Öncelikle devre elemanlarının birbirlerine ne şekilde bağlı oldukları tespit edilir. Örneğin S1 ve S2 elemanları birbirlerine paralel bağlı olduklarından, S1+S2 şeklindeki bağlantıya ait ifade PLC içersinden serbest olarak seçilebilen bir durum tespit işaretçisine atanır. Örneğin "M1.0 = S1+S2 " gibi.

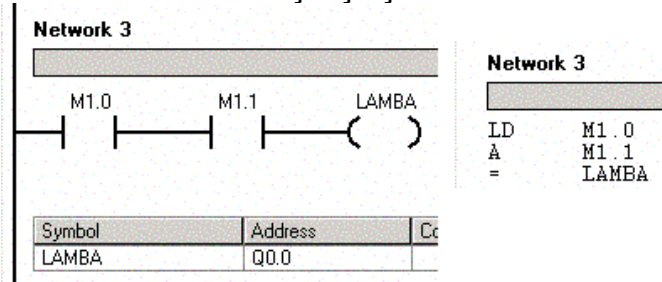


Network 2 // S3,S4,S5 elemanlarından oluşan devre bölümü de PLC içersinden serbest olarak seçilebilen bir durum tespit işaretçisine atanır. S3 ve S5 elemanları birbirine seri ve S4 elemanı da bu seri devreye paralel bağlı durumda olduklarından devre “M1.1 = (S3XS4)+S5” şeklinde ifade edilebilir.



Network 3 // Bundan böyle S1+S2 şeklindeki ifadenin sonucu M1.0 durum tespit işaretçisi içinde saklıdır. Aynı şekilde (S3XS4)+S5 şeklindeki ifadenin sonucu da M1.1 içinde saklı durumdadır.

M1.0 ve M1.1 şeklindeki her iki durum tespit işaretçisi de birbirlerine seri bağlı olduklarından Q çıkışı için $Q = M1.0 \times M1.1$ eşitliği yazılabilir.



SMB0: Durum Bitleri

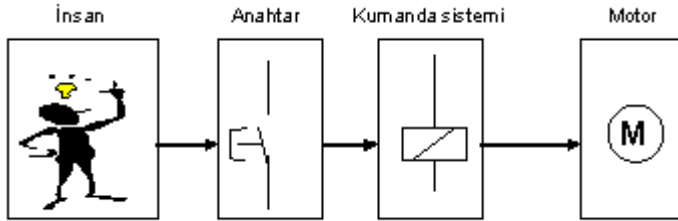
Tablo D-1’de açıklandığı gibi, SMB0 içerisinde S7-200 tarafından her taramada güncellenen 8 adet bit yer alır.

Tablo D-1 Özel Hafıza Baytı SMBO (SM0.0 ila SM0.7)

SM Bit	Açıklama (Salt Oku)
SM0.0	Bu bit her zaman "1"dir.
SM0.1	Bu bit ilk taramada "1"dir. Başlangıç ayarlarının yapılması için kullanılabilir.
SM0.2	Bu bit, kalıcı veri kaybında sadece bir tarama için "1"dir. Bu bit bir hata durumu göstergesi olarak veya özel bir başlatma mekanizması kontrolü için kullanılabilir.
SM0.3	Bu bit, enerji verildikten sonra ilk tarama için "1" olur. Makina ısıtması için kullanılabilir.
SM0.4	Bu bit, 1 dakika periyotlu flaşördür. Yani, 30 saniye süreyle on, 30 saniye süreyle off olur.
SM0.5	Bu bit, 1 saniye periyotlu flaşördür. Yani, 0.5 saniye süreyle on, 0.5 saniye süreyle off olur.
SM0.6	Bu bit bir tarama süresince "1", onu takip eden taramada "0" olan ve böylece tekrarlayan bir bittir. Tarama sayısı sayacı olarak kullanılabilir.
SM0.7	Bu bit, konum şalterinin pozisyonunu gösterir. Off durumu TERM'e, On durumu RUN'a işaret eder. Bu biti kullanarak Freeport modu ve PPI modu arasında geçiş kontrolü sağlanabilir.

KLÂSİK KUMANDA DEVRELERİNİ PROGRAMLAMA

Kumanda : Geri beslemesiz sistemlerin tümü kumanda yapısındadır. Örneğin bir anahtar elemanı ile bir motorun çalıştırılması bir kumandadır.



Bir anahtar elemanı ile bir motorun çalıştırılması bir kumandadır.

Trafik sinyalizasyon şebekesini düşünelim : Bunlar, bir zaman ayarlayıcı (timer) ile yanıp sönerler. Belli bir zaman süresince kırmızı yanar, trafiği durdurur. Başka bir zaman süresince yeşil yanarak trafiği açar. Böyle bir sistem trafiği iyi düzenleyemez. Trafik akışı olmasa da veya yoğun olsa da, bu ışıklar aynı zaman aralıklarında yanıp sönerler. O halde trafik sinyalizasyon şebekeleri bir zamana bağımlı kumanda biçimidir.

Kumanda sistemlerini aşağıdaki gibi sınıflandırabiliriz :

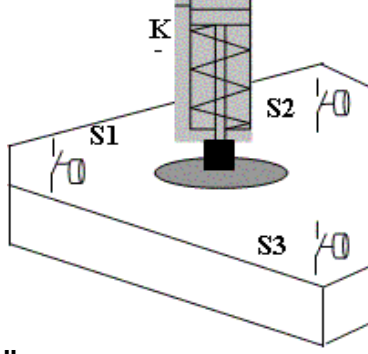
- 1- Kombinasyonel lojik kumandalar.
- 2- Ardışık kumandalar.
 - a-) Zaman bağımlı ardışık kumandalar.
 - b-) İşleme bağımlı ardışık kumandalar.

1- KOMBİNASYONEL LOJİK KUMANDALAR

Kombinasyonel kumandalar, giriş değişkenlerinin durumuna göre çıkışın sahip olabileceği durumları inceler. Bu kumanda sistemlerinin çözümünde doğruluk tablolarının çıkarılması büyük yarar sağlar.

Kombinasyonel çözümlerde giriş ve çıkış değişkenleri belirlenerek doğruluk tablosunda ilgili haneye yazılır.

Her bir giriş değişkeninin sahip olabileceği durumlar, doğruluk tablosunun ilgili hanesine kaydedilir. Bundan sonra girişlerin durumuna göre hangi çıkış ya da çıkışlar aktif (1) ise çıkış hanesinde bu durum '1' olarak işaretlenir. Bundan sonra çıkış hanesinde '1' bulunan satırlara ait lojik ifadeler çarpımların toplamı şeklinde yazılır. Yani '0' bulunan satırlar normalde kapalı kontak biçiminde '1' bulunan satırlar ise normalde açık kontak biçiminde ele alınarak çizilir.

ÜÇGEN PRES :

Üçgen bir masanın her 3 köşesinde buton elemanları bulunmaktadır. K1 presi, ancak herhangi 2 butona birlikte basılması durumunda çalışacaktır. Sadece tek bir butonun uyarılması durumunda veya üçünün birden uyarılması durumunda pres çalışmamalıdır.

İstenenler :

1- Sistemin çözümünü S7200 PLC için gerçekleştiriniz.

Çözüm :

1- Doğruluk tablosunun oluşturulması :

Verilen probleme ait S1, S2, S3 şeklinde 3 giriş değişkeni bulunduğundan

;

$n = 3$ olur. 2^n formülüne göre, $2^3 = 8$ değişik durum vardır.

Buna göre doğruluk tablosunu oluşturalım:

Herhangi iki butona birlikte basıldığında presin çalışması istendiğine göre 3 buton da birbirlerine seri "AND" fonksiyonu şeklinde bağlı olmalıdırlar.

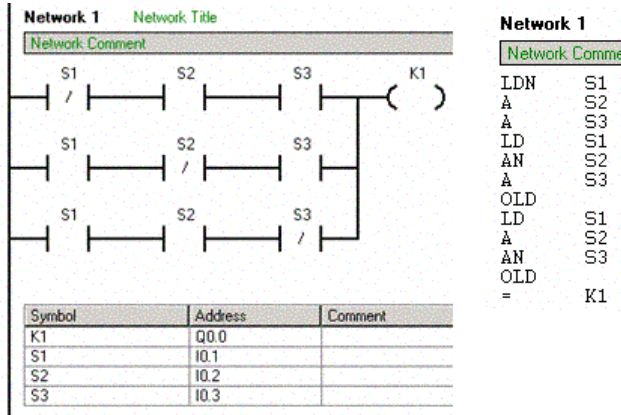
Kontak elemanlarından biri normalde kapalı olmalıdır ki diğer ikisi uyarıldığında çıkış aktif olabilsin. Başka bir ifadeyle, butonlardan ikisi mutlaka normalde açık ve birisi de kapalı olmalıdır.

	GİRİŞ			ÇIKIŞ
	S1	S2	S3	K1
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	0

Doğruluk tablosunun çıkış hanesinde istenilen fonksiyonu veren çıkışlar "1" olarak işaretlenir. Bu durumda doğruluk tablosu çıkış hanesindeki sıfırlar bizi ilgilendirmemektedir.

Sistemin matematik ifadesi : Doğruluk tablosunun 4.6. ve 7. satırları istenilen çalışmaya uygundur. Diğer satırlardaki durumlar dikkate alınmaz. 4.6. ve 7. satırlar çarpımların toplamı şeklinde ifade edildiğinde ;

$$K1 = \underbrace{\bar{S1} \cdot S2 \cdot S3}_{4.\text{satır}} + \underbrace{S1 \cdot \bar{S2} \cdot S3}_{6.\text{satır}} + \underbrace{S1 \cdot S2 \cdot \bar{S3}}_{7.\text{satır}}$$

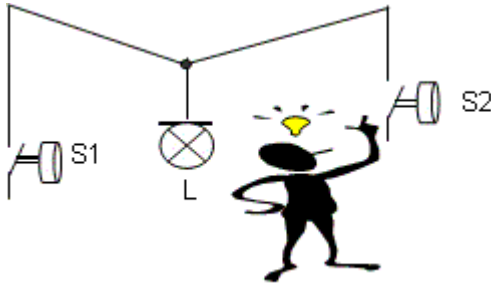


Örnek 1: Bir L lambası S1 anahtarından yakılmış ise S2' den söndürülebilir. Veya S2' den yakılmış ise S1 anahtarından söndürülebilir. Sistemdeki her iki anahtar da kalıcı tipten olacaktır.

İstenenler :

- Doğruluk tablosunu oluşturunuz.
- Doğruluk tablosundan çıkan sonuca göre PLC programını yazınız.

Durum planı :



Problemin kavranması :

- Devrede A ve B şeklinde iki giriş değişkeni bulunmaktadır.
- Devredeki her bir anahtar L lambasını hem yakma, hem de söndürme fonksiyonuna sahiptir.
- Sistem elektrik tesislerinde sıkça olarak kullanılan vavien tesisidir.

Çözüm :

Doğruluk tablosunun oluşturulması :

Sistemde S1 ve S2 şeklinde iki giriş değişkeni bulunduğundan,

$n = 2$ olur.

$2^n = 2^2 = 4$ değişik durum vardır.

Buna göre doğruluk tablosunu oluşturalım :

	GİRİŞ		ÇIKIŞ
	S1	S2	L
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

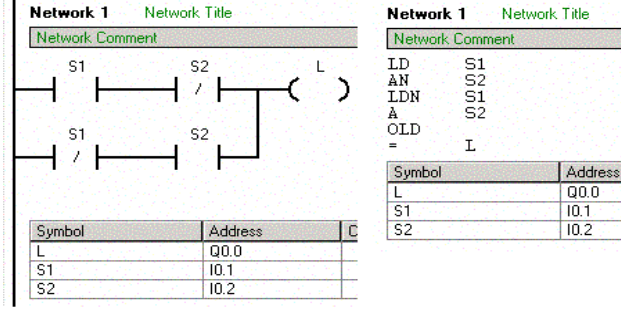
1. satırda her iki anahtar da uyarısız olduğundan lamba yanmaz. $L = 0$

2. satırda sadece S2 anahtarı uyarılı durumda ve lamba yanar. $L = 1$
3. satırda sadece S1 anahtarı uyarılı durumda ve lamba yanar. $L = 1$
4. satırda her iki anahtar da uyarılı ve lamba yanmaz. $L = 0$

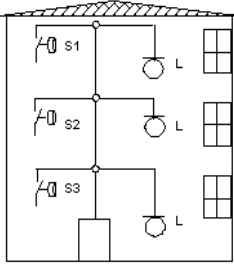
Sistemin matematik ifadesi ve programlama :

Doğruluk tablosunun çıkış hanesinde '1' olan satırlar bizim istediğimiz çalışmayı gerçekleştirecek olan satırlardır. Bunlar 2. ve 3. satırlardır. O halde bu satırlar çarpımların toplamı şekline çevirelim:

$L = S1 \cdot S2 + S1 \cdot \bar{S2}$ elde edilir. (X-OR fonksiyonu)



Örnek 2 : Üç katlı bir bina da merdiven aydınlatmasının her kattan ayrı ayrı yakılıp söndürülmesi isteniyor. Anahtarların her birisi kalıcı tip olup, her hangi bir kattan yakılan lamba, herhangi 2 kattan söndürülebilecektir.



İstenenler :

1. Sistemin doğruluk tablosunu çıkartınız.
2. Sistemin matematik ifadesini yazınız.
3. Sistemi PLC için programlayınız.

Cözüm :

- S1 ile lamba yakıldığında, S2 veya S3 ile sönsün.
- S2 ile lamba yakıldığında, S1 veya S3 ile sönsün.
- S3 ile lamba yakıldığında, S1 veya S2 ile sönsün.

Doğruluk tablosunun oluşturulması :

Verilen problemde A,B ve C şeklinde üç giriş değişkeni bulunduğundan ; $n = 3$ olur.

Bu durumda 2^n formülüne göre; $2^3 = 2^3 = 8$ değişik durum söz konusudur.

Buna göre doğruluk tablosunu oluşturalım :

GİRİŞ			ÇIKIŞ
S1	S2	S3	L
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Doğruluk tablosu incelendiğinde şu sonuca varmak mümkündür. Tabloda yatay satırları temsil eden ifadeler birbirleri ile çarpım durumunda ve sütunlar ise toplam durumundadır. Buna göre her üç anahtarın birbirlerine seri bağlı

olması gereklidir. O halde seri bağlı üç anahtardan herhangi birine basılması durumunda (lojik1) lambanın yanabilmesi için diğer iki anahtarın normalde kapalı olması gereği hemen anlaşılmalıdır.

Sistemin matematik ifadesi :

L lambasının istenilen şekilde çalışabilmesi için değişik kombinasyonlar mevcuttur. Lambanın istenilen şekilde çalışmasını sağlayan kombinasyonları, doğruluk tablosunun çıkış (L) hanesinde '1' olarak işaretlediğimizde problem çözülmüş olacaktır.

“Doğruluk tablosunun çıkış hanesinde istenilen çalışmayı sağlayabilen dört adet '1' bulunmaktadır. Doğruluk tablosunun çıkış bölümünde '1' bulunan satırları formüle edersek aşağıdaki sonuca varabiliriz :
'1' bulunan satırlar yukardan aşağıya doğru sırayla ;

1. adım : $\textcircled{R} S1.S2.S3$

Topla

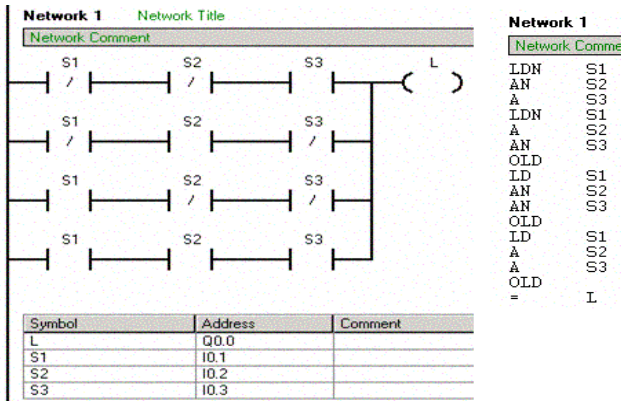
2. adım : $\textcircled{R} S1.S2.S3$

3. adım : $\textcircled{R} S1.S2.S3$

1. adım : $\textcircled{R} S1.S2.S3$

Yukarıdaki satırlara ait ifadeleri toplarsak aşağıdaki sonucu elde ederiz:

$$L = S1.S2.S3 + S1.S2.S3 + S1.S2.S3 + S1.S2.S3$$



3- ÇIKIŞ ATAMALARINA AİT KOMUTLAR :

Çıkış rölelerine ait komutları aşağıdaki gibi sınıflandırabilirsiniz

- Çıkış

???

— ()

- Set

???

— (S)

????

- Reset

???

— (R)

????

ÇIKIŞ : Bobinin bağlı olduğu hatta enerji geldiği zaman çıkış rölesi çeker ve adres '1' olur.

Bobinden önce 'NOT' kontağı kullanılarak çıkışın ters olarak çalışması sağlanabilir.

- İş elemanları olarak değerlendirebileceğimiz pnömatik silindir, motor, valf, aydınlatma gereci vb. elemanlarla PLC arasındaki bağlantıyı çıkış rölesi sağlar. PLC'nin tür ve kapasitesine göre 8, 16, 32 adet çıkış rölesi bulunabilir.
- PLC içerisinde dahili ve harici olmak üzere iki kontak tipi bulunur. Harici kontaklar yük devresine doğrudan bağlanabilir. Ancak dahili çıkış kontağı sadece program içerisinde başka fonksiyonları etkilemek için kullanılabilir.
- Çıkış rölesinde de giriş rölesinde olduğu gibi normalde açık ve normalde kapalı kontaklar bulunur.
- Çıkış röleleri de program içerisinde belli adreslere sahiptir. Program içerisinde çıkış röleleri bu adreslerden çağrılarak işlem yapılır. Adresi belli olmayan bir çıkış rölesi komutunu PLC kabul etmez.

RS (TUTMA KONTROLÜ) KOMUTLARI VE UYGULAMASI

HAFIZA : Çıkış sinyallerinin girişi etkilemesidir. Bu da bellek fonksiyonunun belirgin bir özelliğidir.

Hafıza, bir geçici durum sinyalinin sürekli hale getirilmesi işlemidir. Bir sinyalin hafızaya alınabilmesi için sinyalin süreklilik özelliği bulunmamalıdır. Yani sinyal geçici bir duruma sahip olmalıdır. Örneğin, normalde açık bir buton elemanının bir an için uyarılması geçici bir durum sinyali (impuls) oluşturur. Bunun yanında kalıcı tip anahtar fonksiyonları için hafıza işlemine gerek duyulmaz.

Otomasyon sistemlerinin PLC ile gerçekleştirilmesinde hafıza tekniği iki biçimde gerçekleştirilebilir :

- 1- AND, OR vb. “genel fonksiyonlarından” yararlanarak oluşturulan hafıza.
- 2- RS hafıza fonksiyonlarını kullanmak suretiyle oluşturulan hafıza.

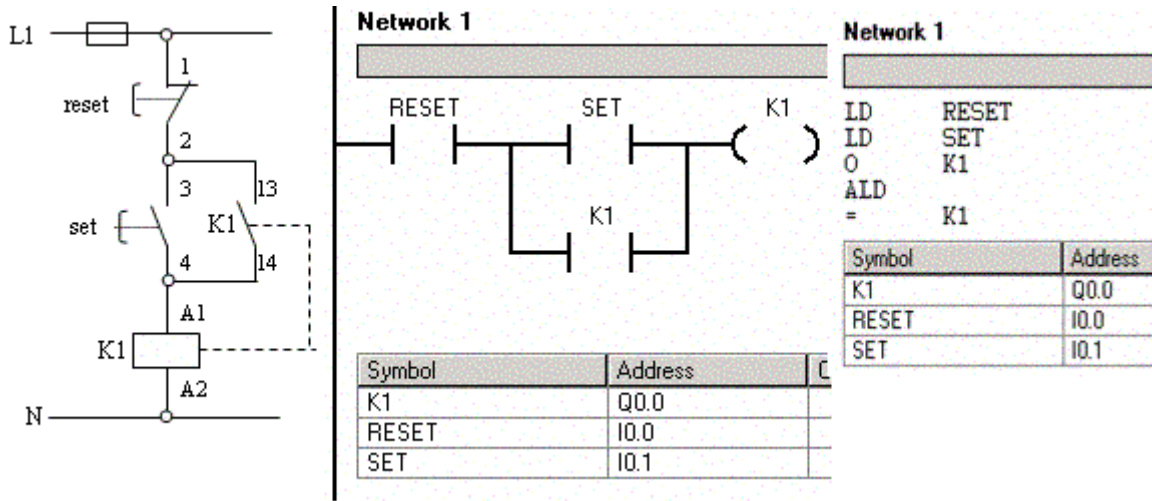
HAFIZA İŞLEMİNİN (TUTMA KONTROLÜ) GENEL FONKSİYONLARLA GERÇEKLEŞTİRİLMESİ

- 1- Reset ağırlıklı (durdurma ağırlıklı) hafıza tekniği.
- 2- Set ağırlıklı (çalıştırma ağırlıklı) hafıza tekniği.

1. Reset ağırlıklı hafıza tekniği :

Bu teknikte, set ve reset butonları aynı anda uyarıldığında reset işlemi baskın olduğundan sistem kumanda almaz (reset baskın).

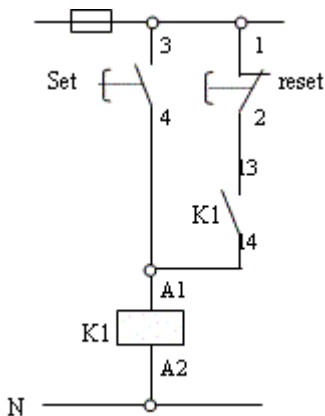
Konvansiyonel devre şeması

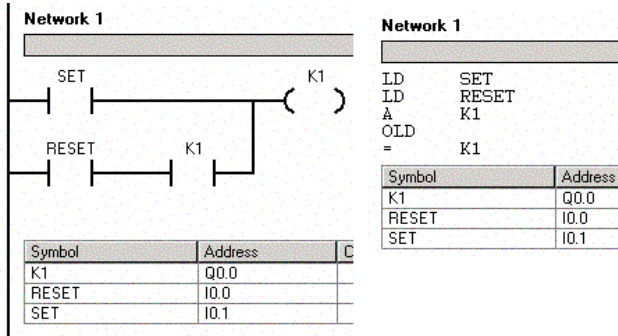


2. Set ağırlıklı hafıza tekniği :

Bu teknikte, set ve reset butonları aynı anda uyarıldığında set elemanı baskın olduğundan sistem kumanda alarak çalışır (set baskın).

Konvansiyonel devre şeması

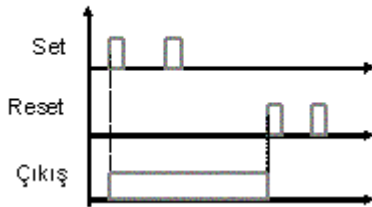




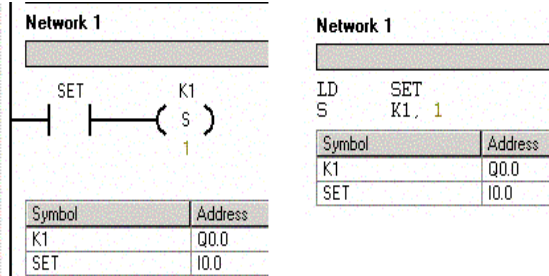
BELLEK ELEMANI (RS) İLE TUTMA KONTROLÜ

RS hafıza elemanları kumanda problemlerinde çok sık olarak kullanılmaktadır. Bu nedenle bütün PLC üreten firmalar, belli sayıda bir RS elemanını intern olarak hazırlamışlardır. RS hafıza elemanı, geçici durum sinyallerinin hafızaya alınarak sürekli hale getirilmesini sağlar. Elektrik kumanda tekniğinde mühürlemeli devre bağlantılarına eşdeğerdir.

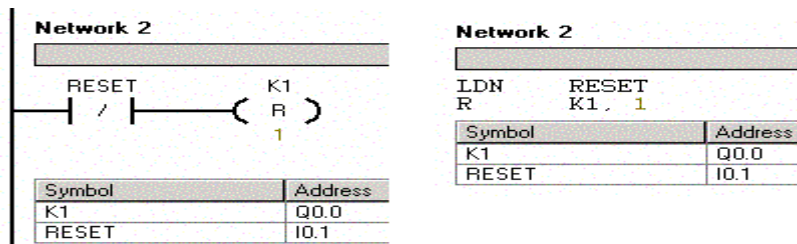
Çalışma diyagramı :



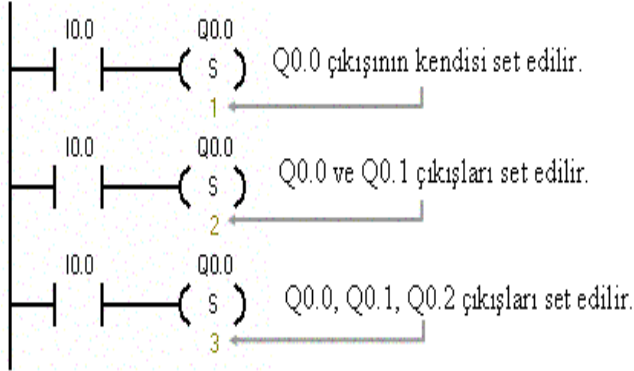
Yukarıdaki çalışma diyagramına uygun PLC yazılımı aşağıda olduğu gibidir. “SET” butonelemanı bir an için uyarıldığında “K1” çıkış biti sürekli “<ST1:METRICCoVERTERProductID="1"” w:st="on">1”</ST1:METRICCoVERTER> olur.



Bu durum RESET sinyali uygulanmasına kadar devam eder. Reset sinyali PLC giriş adresine normalde kapalı kontaklardan geldiğinden sinyalin tersi alınmalıdır. Bu buton uyarıldığında K1 çıkış biti reset olur.



Bir' den fazla çıkışın set edilmesi :



Bir' den fazla çıkışın Reset edilmesi :

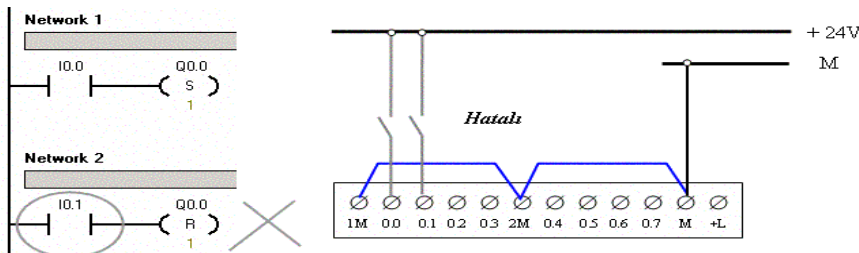


PLC programlamada durdurma işlemleri, röle tekniğinde olduğu gibi normalde kapalı kontak formunda gerçekleştirilmelidir. Kumanda programında herhangi bir arızadan dolayı sistemi durduramama gibi bir ihtimal bulunmamalıdır. Örneğin, herhangi bir şekilde durdurma (stop) elemanını kablosunun kopması durumunda sistem kendini otomatik olarak durdurabilmelidir.

Bellek fonksiyonları ile oluşturulmuş kumanda ve kontrol programında, durdurma işlemi iki biçimde gerçekleştirilebilir :

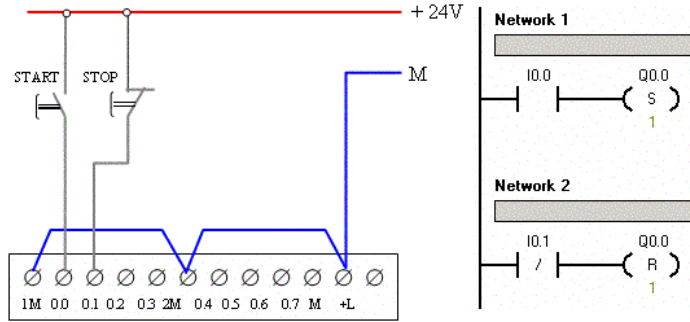
- 1- **Normalde açık elemanlarla.**
- 2- **Normalde kapalı elemanlarla.**

Bellek elemanlarının durdurma (reset) işlemi normalde açık kontaklarla yapılabilir. Sonuçta durdurma işlemi gerçekleşir. Ancak bu yöntemin en sakıncalı tarafı, durdurma elemanı giriş ya da çıkışında meydana gelebilecek iletken kopması durumunda çalışmakta olan sistem durdurulamaz.



SONUÇ: Bellek elemanlarının normalde açık buton elemanları ile durdurulması kablo kopmalarına karşı güvenilir değildir. Bu nedenle uygulanmamalıdır.

Otomasyon sistemlerinde, durdurma ve koruma işlemlerinin özel bir yeri ve tekniği olmalıdır. Bellek fonksiyonları ile gerçekleştirilen kumandalarda stop elemanı PLC ünitesine normalde kapalı kontaklar ile ulaştırılmalıdır.



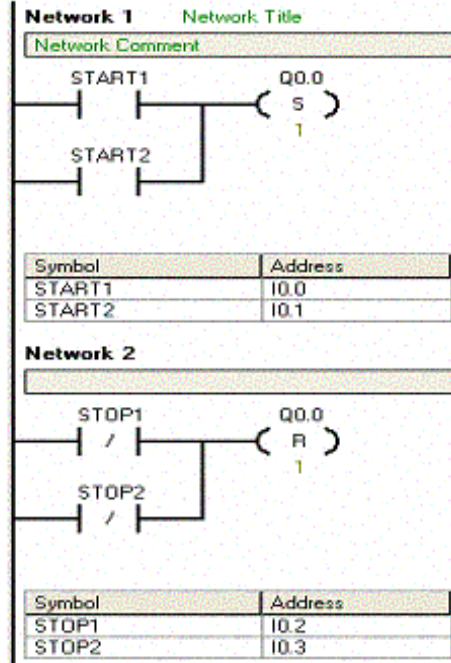
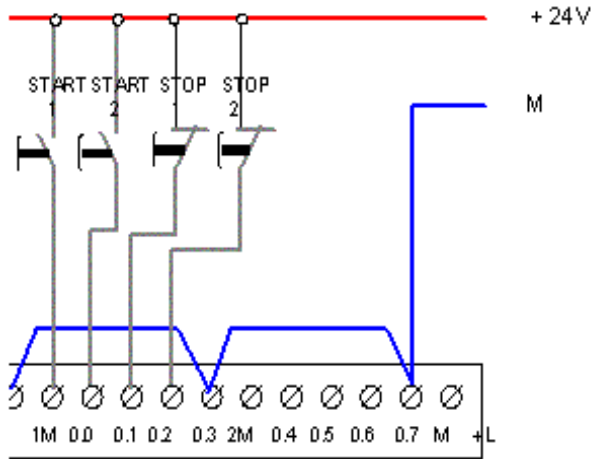
SONUÇ: Bellek fonksiyonunun normalde kapalı buton elemanları ile durdurulması kablo kopmalarına karşı güvenilirdir. Bu nedenle uygulanmalıdır.

Bazı durumlarda set edilmiş bir bellek elemanı (RS) birden fazla noktadan durdurulması gerekir. Örneğin bir motor iki ayrı yerden çalıştırılıp, iki ayrı noktadan da durdurulması rastlanan bir durumdur.

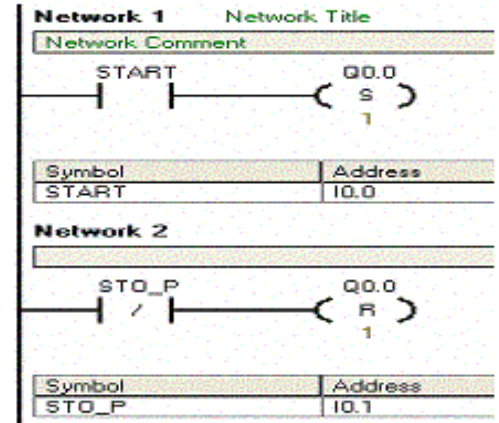
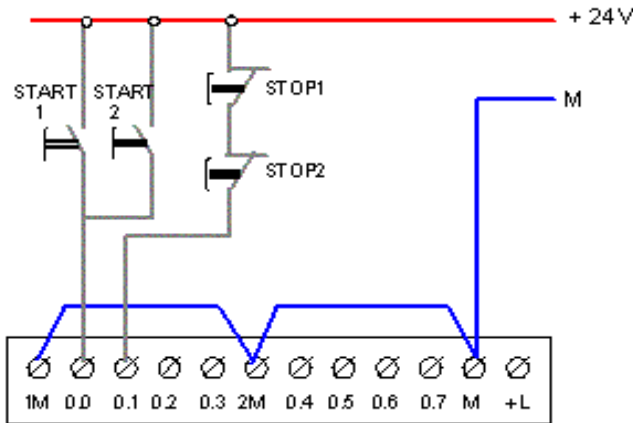
Bazı durumlarda set edilmiş bir bellek elemanı (RS) birden fazla noktadan durdurulması gerekir. Örneğin bir motor iki ayrı yerden çalıştırılıp, iki ayrı noktadan da durdurulması rastlanan bir durumdur.

1. yöntem:

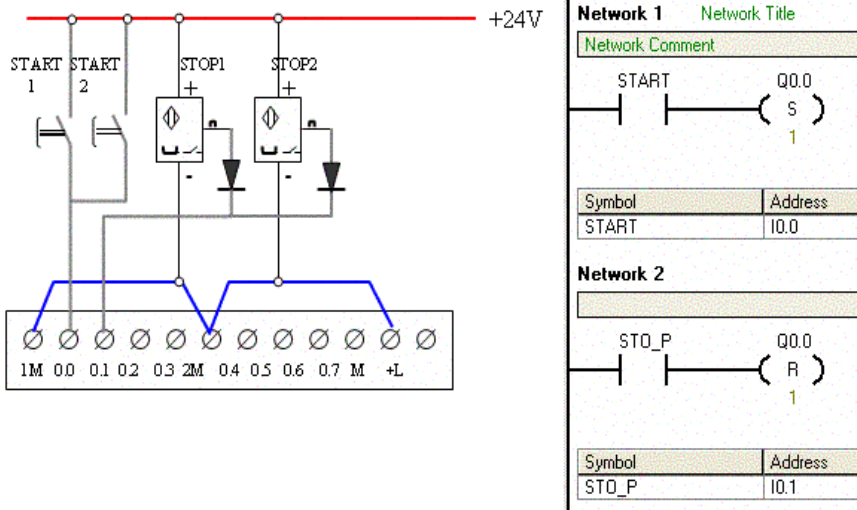
Start ve stop elemanları aynı aynı PLC adreslerine girilir. Gerekli paralele bağlantılar program içerisinde yapılır. Toplam 4 input kullanıldığından bir sonraki yöntemlere göre ekonomik değildir.



Bu yöntemde tüm start elemanları dışarıda paralel bağlanır. Bu şekilde PLC input ünitesine sadece tek bir adresten giriş yapılır (I0.0). Aynı şekilde tüm stop elemanları dışarıda seri bağlanıp aynı şekilde tek bir giriş adresine bağlanır (I0.1). Bu şekilde toplam 2 input kullanıldığından bir önceki yöntemle nazaran 2 input az kullanılmış olur.



Bu yöntemde söz konusu motor iki farklı yerden çalıştırılabilir. Yine iki farklı yerden durdurulmak istenmektedir. Ancak durdurma elemanları stop butonu yerine PNP çıkışlı sensörlerdir. Görüldüğü üzere sensörler dışarıda paralel bağlanıp tek bir input adresine bağlanmışlardır. Sensörlerin birbirleri üzerinden kısa devre akımı dolaştırıp bozulmalarını önlemek için sinyal çıkışlarına mutlaka birer diyot bağlanmalıdır



RS hafıza fonksiyonlarını kullanmak suretiyle SET öncelikli veya RESET öncelikli hafıza fonksiyonları gerçekleştirilebilir. Ancak bundan evvel **ÖNCELİK** kavramının PLC' de ne anlam taşıdığıнын bilinmesi gereklidir.

Bilindiği gibi PLC, program işlemede seriyel bir yol izler. Yani, programda komutların taranarak işlenmesi, yukarıdan aşağıya ve soldan sağa doğru gerçekleşir. Program, END komutunun bulunduğu satıra kadar işlendikten sonra, tekrar yeni bir çevrim için ilk komutun yazıldığı satıra (başa) döner. Buna bir tarama çevrimi denilir. İlk taramadaki programın bir kopyası işlem görüntüleri birimi (PI) denilen bir RAM alanına kopyalanır. Yani giriş ve çıkışlarla ilgili değerleri hemen çıkışlara aktarmaz.

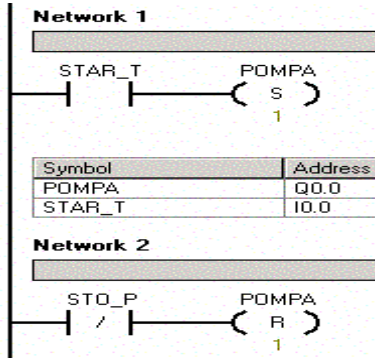
İkinci bir çevrimde program tekrar yukarıdan aşağıya ve soldan sağa olmak üzere taranır. END komutunun bulunduğu satıra gelindiğinde, programın o anki kopyası tekrar giriş görüntüleri birimine kopyalanır. Ancak bu alanda, bir önceki tarama çevriminden dolayı bulunan programa ait kopya ötelenerek çıkışlara ya da ilgili alanlara gönderilir. Yani, çıkış röleleri çalıştırılır. Zamanlayıcı ve sayıcılara ait zaman değerleri işlenir. Dikkat edilirse bir giriş sinyalinin çıkışta izlenmesi ancak bir tarama çevrimi sonra gerçekleşir.

- Bellek fonksiyonunun Reset ağırlıklı olarak yazılması :
- Bellek fonksiyonunun Set ağırlıklı olarak yazılması :

Yazılımda en son işlenen komut hakim komuttur. Hem start hem de stop elemanının aynı anda uyarılması durumunda ;

STAR_T = 1 STO_P = 1 olacaktır.

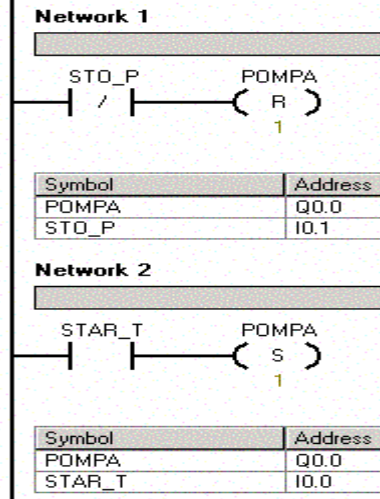
Bu durumda çıkış biti yukarıda set edilirken, aşağıda reset edilecektir. En son durum, çıkış bitinin reset durumu olduğundan, yazılım reset ağırlıklıdır denilir.



Yazılımda en son işlenen komut hakim komuttur. Hem start hem de stop elemanının aynı anda uyarılması durumunda ;

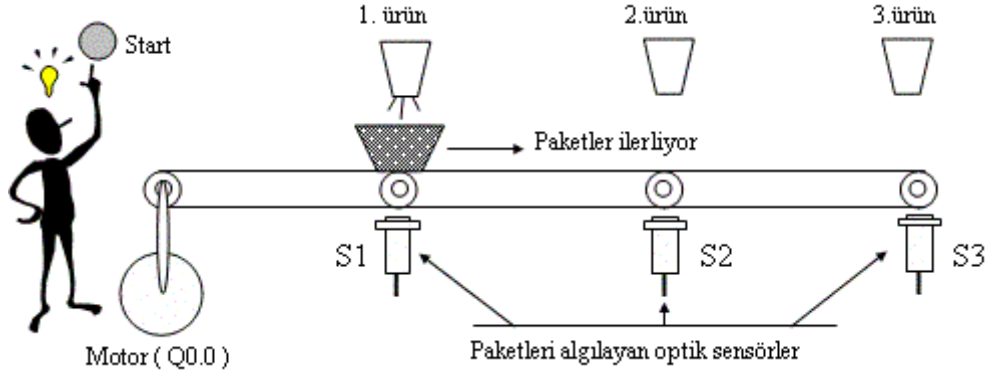
STAR_T = 1 STO_P = 1 olacaktır.

Bu durumda çıkış biti yukarıda Reset edilirken, aşağıda Set edilecektir. En son durum, çıkış bitinin Set durumu olduğundan, yazılım Set ağırlıklıdır denilir.

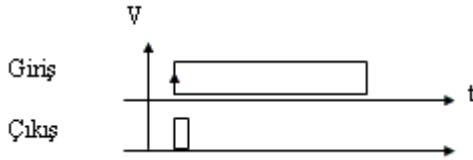


UYGULAMA : Bir bant sistemi üzerinde ilerleyen paketlere 3 değişik toz ürün doldurulacaktır. Bunun için start verici eleman uyarılarak, bant motorunun çalışması sağlanacaktır.

- Paket S1 sensörü algılama alanı içine girdiğinde bant motoru stop edecektir
- Bir miktar ürün pakete boşaltılacaktır. Operatör aynı start butonunu ikinci kez uyararak bant motorunu tekrar çalıştıracaktır.
- Paket bu defa S2 sensörü tarafından algılandığında tekrar ve ikinci ürün, aynı şekilde bir sonraki işlemle 3. ürünün dolması sağlanacaktır.
- Bu işlemler aynı şekilde devam etmelidir.

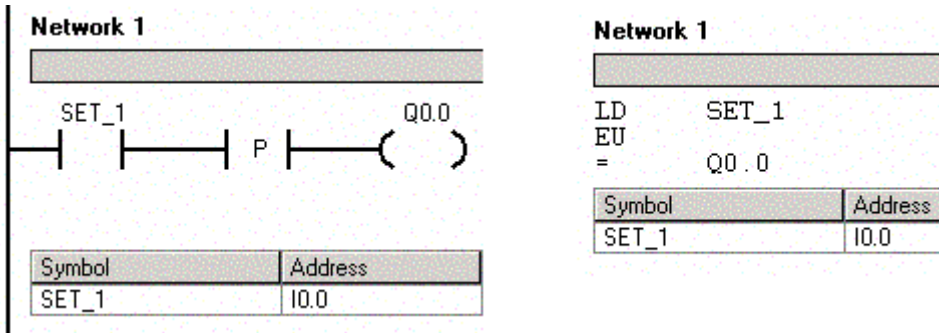


Bu komut, 0'dan '1' konumuna her geçişte, sadece bir tarama için güç akışına izin verir. Çalışma grafiğinden de görüldüğü gibi bir butonun kapatılmasıyla (pozitif değişim) çıkışta sadece bir tarama çevrimi süresince sinyal elde edilir. Bu sinyalin süresi tarama çevrimiyle ilgili olduğundan çok kısadır.

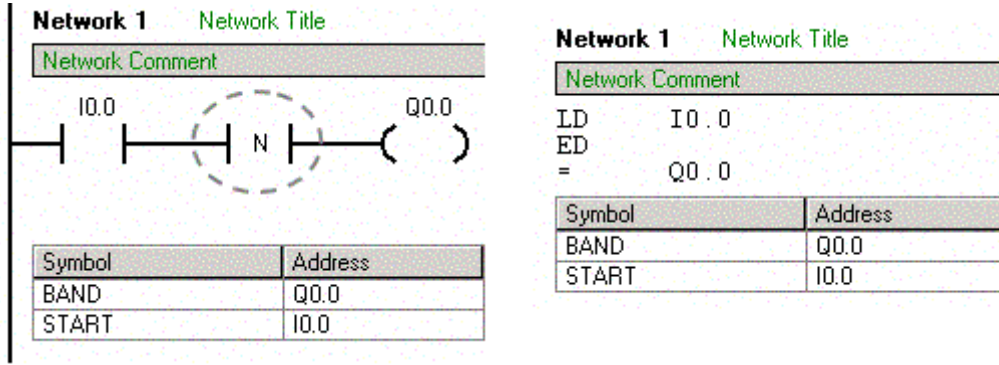


Uygulama 1 :

PLC cihazımızın I0.0 giriş adresine normalde açık bir buton elemanı bağlayalım: Butonu sürekli uyarılı (kapalı) tutup, bu esnada Q0.0 adresli çıkışın sadece bir tarama çevrimi süresince "ON" olduğunu görmeliyiz.



Bu komut 1'den 0 konumuna her geçişte, sadece bir tarama için güç akışına izin verir. Çalışma grafiğinden de görüldüğü gibi normalde açık bir buton kapatıldığında çıkışta bir sinyal gözlenmez. Ancak buton elemanı açıldığında (negatif değişim) sadece bir tarama çevrimi süresince bir çıkış sinyali elde edilir. Bu sinyalin süresi tarama çevrimiyle ilgili olduğundan çok kısa sürelidir. Çıkıştaki impuls sinyali, giriş sinyalinin düşen kenarında olduğundan dolayı negatif bir değişimin tespit edilmesinde kullanılır.



UYGULAMA:

Paketleri algılayan optik sensörler

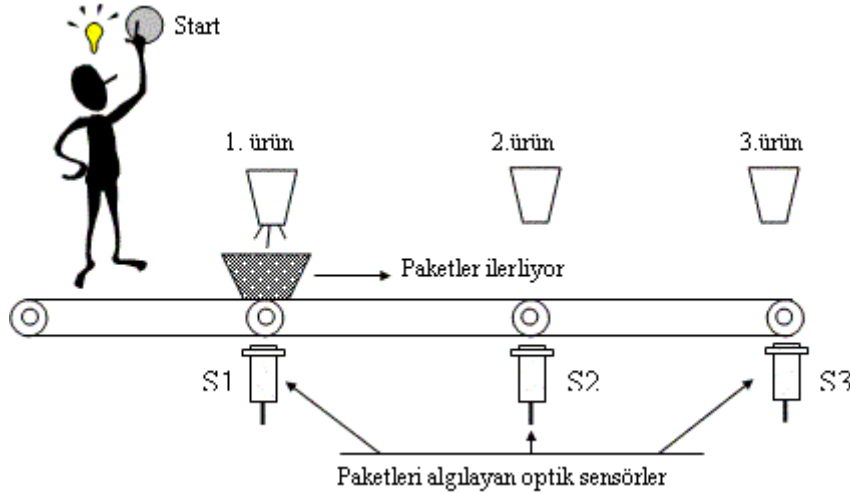
S1 S2 S3

Paketler ilerliyor

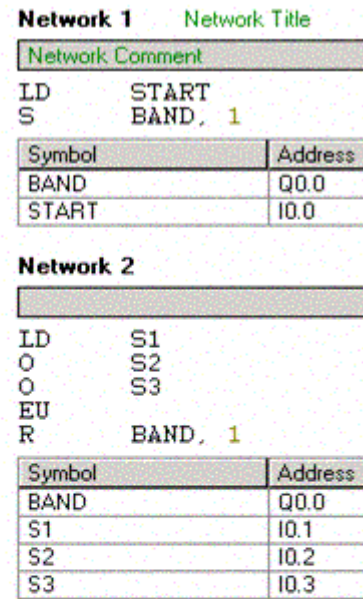
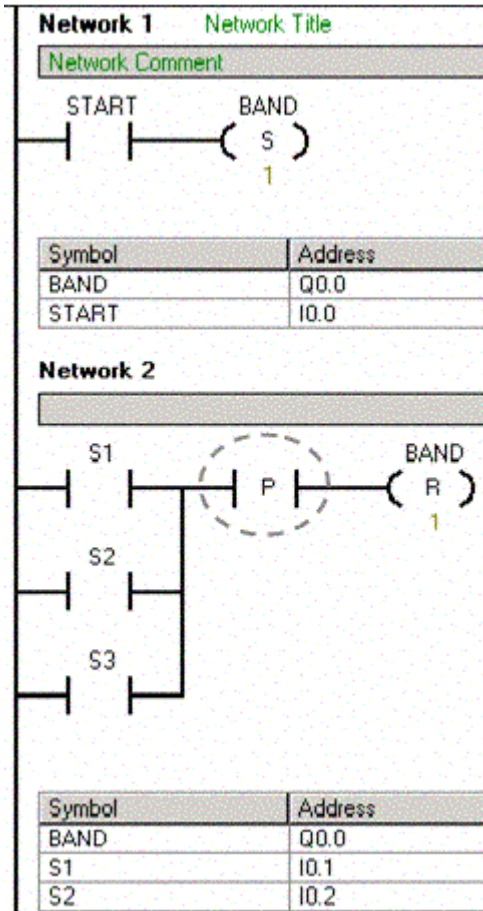
Start

1. ürün 2. ürün 3. ürün

Bir bant sistemi üzerinde ilerleyen paketlere 3 değişik toz ürün doldurulacaktır. Bunun için start verici eleman uyarılarak, bant motorunun çalışması sağlanacaktır.



- Paket S1 sensörü algılama alanı içine girdiğinde bant motoru stop edecektir
- Bir miktar ürün pakete boşaltılacaktır. Operatör aynı start butonunu ikinci kez uyararak bant motorunu tekrar çalıştıracaktır.
- Paket bu defa S2 sensörü tarafından algılandığında tekrar ve ikinci ürün, aynı şekilde bir sonraki işlemle 3. ürünün dolması sağlanacaktır.
- Bu işlemler aynı şekilde devam etmelidir.



Birbiriyle aynı anda devrede olmaması gereken sistemler vardır. Örneğin; 3 fazlı asenkron motorlarda devir yönünün değişiminde her iki yön rölesinin aynı anda devrede bulunması hali istenmeyen bir durum olup, faz çakışması ile sonuçlanmaktadır. O halde bu iki röle aynı anda devrede bulunmamalıdır.

Kilitleme, gerçek anlamda yasaklama kavramından yararlanılarak oluşturulmuş sistemdir. **Aynı anda devrede bulunması istenmeyen diğer çalışma organına ait giriş ya da çıkış sinyalinin sıfır değeri sorgulanır.**

İki giriş sinyalinden birisi verildikten sonra, ikinci giriş sinyalinin yasaklandığı sistemlere "kilitlemeli sistemler" adı verilmektedir.

Kilitlemelerin görevi, otomasyon sistemlerinde çeşitli istenmeyen durumları engellemektir. Örneğin, bir yönde dönmekte olan bir elektrik motorunun aynı anda aksi yönde dönmeye zorlamak, gerek elektriksel gerekse mekanik yönden arzu edilmeyen ve sistemde hasarlara yol açabilecek bir durum oluşturur.

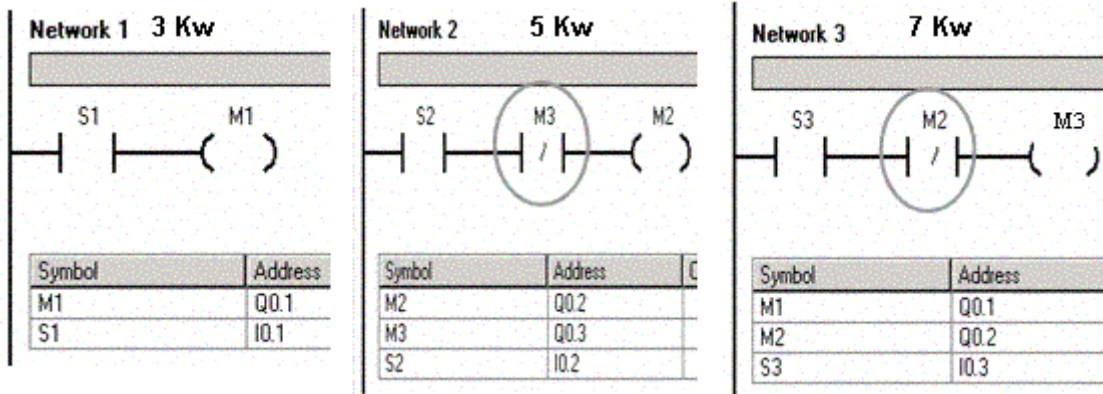
Örnek 2: Bir yükleme tesisinde üç adet motor farklı yükleri kaldırmaktadır.

M1 = 3 Kw, M2 = 5 Kw ve M3 = 7 Kw. Ancak tesisin toplam güç kapasitesi 10 Kw ile sınırlandırılmıştır. Tesis toplam 10 Kw sınırını aşmayacak her türlü kombinasyonda çalışmalıdır.

Problemin kavranması :

- Vinç motorları normalde açık yay geri dönüşlü bir butona (S1, S2, S3) basılı olduğu sürece çalışmalıdır. (Kesik çalıştırma)
- Sistemde ayrıca bir reset (stop) butonu bulunmayacaktır.
- Örneğin, 3 Kw ve 5 Kw'lık motorlar devrede iken 7 Kw gücündeki motor devreye alınmak istendiğinde bu motora ait sinyal etkisiz olacaktır.
- Aynı şekilde 7 Kw gücündeki motor devrede iken, 5 Kw devreye alınmak istendiğinde bu motor kumanda sinyali almasına rağmen devreye girmeyecektir. Aksi halde 10 Kw sınırı aşılmış olmaktadır.

Cözüm önerisi: 10 KW sınırını aşan tek ihtimal M2 ve M3 motorlarının aynı anda devrede bulunmasıdır. Bu nedenle her iki motora ait (M2 ve M3) devresinde kilitlemeler kullanılmalıdır.



Otomasyon sistemlerinin PLC ile gerçekleştirilmesinde birbirini ters etkileyen sistemlerin tasarımında, işin ciddiyeti yönüyle iki tür güvenlik önlemi aynı anda alınmalıdır.

- 1- Yazılım kilitlemesi. (software kilitleme)
- 2- Donanım kilitlemesi (Hardware kilitleme)

1- YAZILIM KİLİTLEMESİ (SOFTWARE KİLİTLEME)

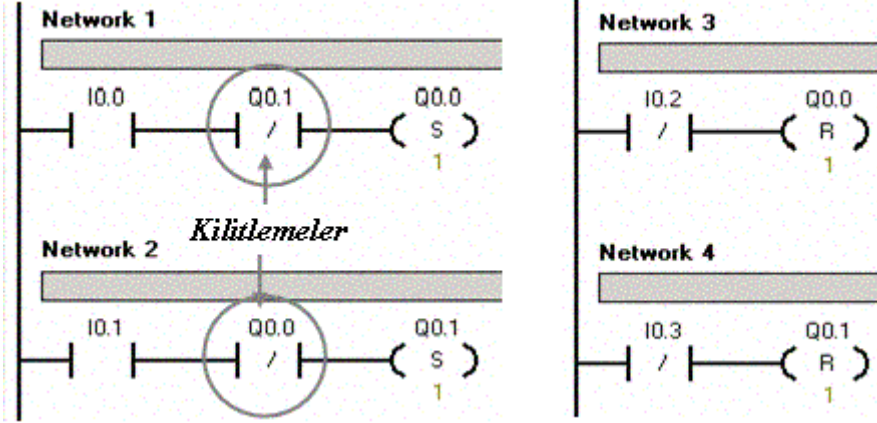
Otomasyon sistemleri PLC ile tasarlanırken kontaklarla oluşturulan mühürleme tekniğinden ziyade bellek elemanları (RS) kullanılır. Bu nedenle kilitleme konusunu bellek elemanları ile oluşturacağız. Bir sistem çalışırken diğer bir sistemin o anda çalışması sakınca oluşturuyorsa bu durumda bellek elemanları karşılıklı olarak kilitlemelidir.

Yazılım kilitlemesini iki biçimde uygulayabiliriz:

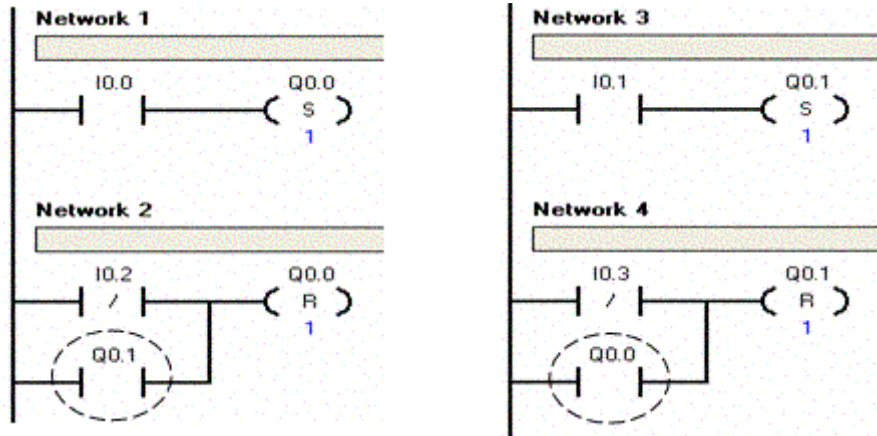
- a- Bellek elemanının (RS) set kısmında kilitleme.
- b- Bellek elemanının (RS) reset kısmında kilitleme.

Kilitlemeler

Bu kitleme işlemi bellek elemanının set girişinde yapıldığından bu isim verilmiştir. Set girişinde oluşturulan “AND” fonksiyonu sayesinde, set işleminin yapılabilmesi için diğer çalışma organına ait çıkışın, mutlaka sıfır değerine sahip olması gerekir. Yani bir çıkış aktiken diğer çıkışın aktif olması engellenmektedir.



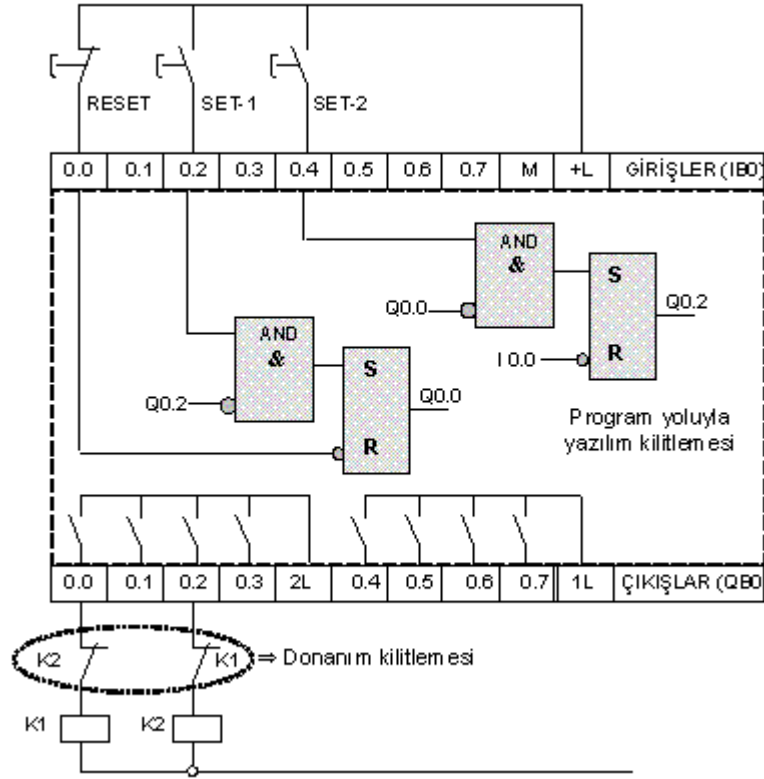
Bu kitleme işlemi bellek elemanının set girişinde yapıldığından bu isim verilmiştir. Bu şekilde Q0.0 = 1 iken, Q0.1 çıkışı sürekli reset durumundadır. Aynı şekilde Q0.1 = 1 iken, Q0.0 çıkışı sürekli reset durumundadır. Aynı anda çalışması istenmeyen sistemlerde kullanılabilir bir yöntemdir.



2- DONANIM KİLİTLEMESİ (HARDWARE KİLİTLEME)

Bu kitleme yönteminde istenmeyen sinyaller, PLC dışında yapılacak olan bağlantılarla giderilir. Elektrik kumanda devresinde meydana gelen kısa devreler sonucunda oluşan kontak yapışmalarında, yazılım kilitlemeleri yetersiz kalır. Bu

yetersizliği ortadan kaldırmak amacıyla PLC dışında, kumanda elemanlarının (röle, kontaktör vb.) kontakları üzerinden de kilitlemeler gerçekleştirilmelidir.



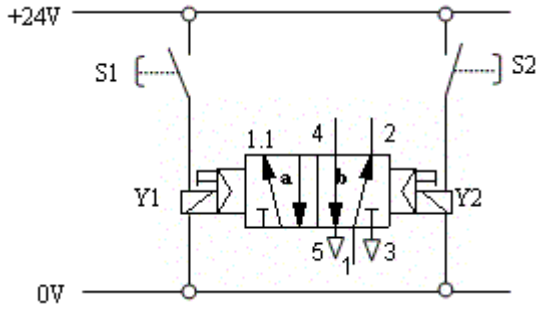
Ciddi arızalara sebep olabilecek durumlarda, gerek yazılım içinde gerekse donanım olarak her iki kilitleme şekli mutlaka kullanılmalıdır.

Unutulmamalıdır ki sadece yazılım yoluyla yapılan kilitlemeler, kontak yapışması durumunda etkisiz kalacaktır.

Şekilde böyle bir prensip uygulama görülmektedir. Buna göre Q0.0 ve Q0.2 çıkışlarının aynı anda çalışması istenmemektedir

Yazılımda gerekli kilitlemeler yapılmasına karşın, donanım yönüyle de PLC dışında gerekli kilitleme bağlantıları yapılmıştır. Bu şekilde herhangi bir kısa devre halinde kontak yapışması oluşmuşsa bu durumda söz konusu kontak açık durumda bulunacaktır. Bu şekilde diğer bir çıkışı çalıştırmak mümkün olmayacaktır.

UYGULAMA : A + B + B – A - şeklindeki ardışık kumanda zincirinde, çift etkili silindirler 5/2 tek bobin uyarılı ve yay geri dönüşlü yönlendirme valfleri ile kumanda edilmek isteniyor. Sistemin kumandasını PLC için hafızalı sıralayıcı yöntemini kullanarak gerçekleştiriniz.

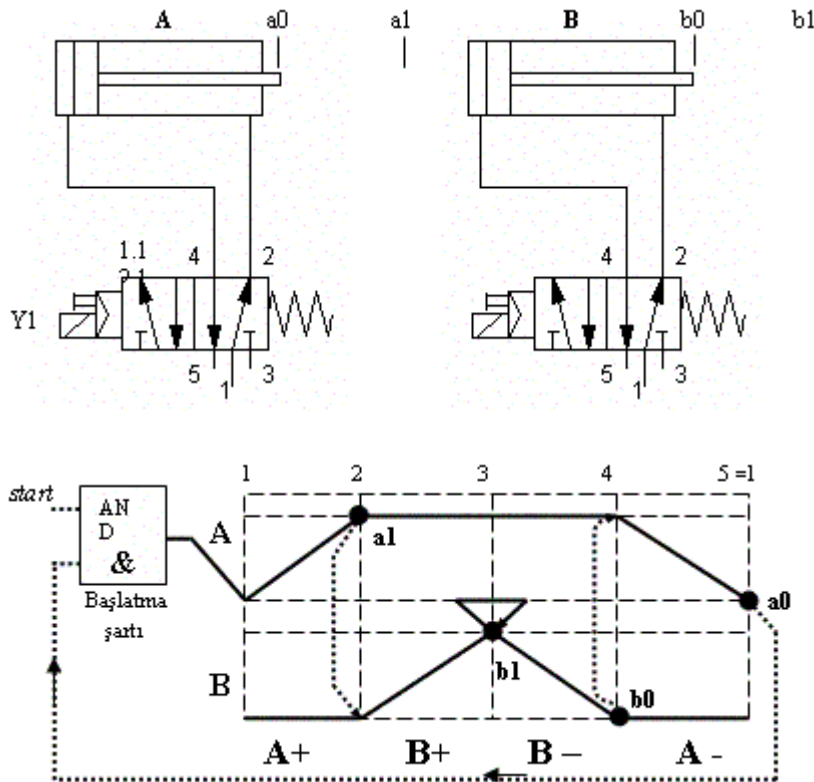


Elektropnömatik sistemlerde ardışık kumanda zincirinin PLC ile kumanda edilmesinde 5/2 tek bobin uyarılı ve yay geri dönüşlü yönlendirme valflerinin kullanılması durumunda, şu prensip çözüm zinciri uygulanmalıdır :

Prensip çözüm zinciri :

- Çalışmayı ilk defa başlatmak için start şartı belirlenir.
- Adım sayısı kadar yardımcı röle kullanılır. (M1.1, M1.2, M1.3, vb.)
- Bir adımdan diğer bir adıma geçmeyi sağlayan şartlar (sinyal vericiler) yol-adım diyagramları oluşturularak belirlenir.
- *Bir önceki adım gerçekleşmişse ve bir adımdan diğer bir adıma geçmeyi sağlayan sinyal verici eleman uyarılmışsa yardımcı röle set edilir.*
- En son adımın tamamlanması ile beraber tüm yardımcı röleler reset edilir.
- Yardımcı röleler çıkışlara atanır.

Pnömatik devre şeması :



"A+B+B- A-" şeklindeki ardışık kumanda zincirinin PLC ile kumandası :

Pnömatik devre şemasında görüldüğü gibi A+B+B-A- şeklindeki ardışık kumanda zinciri, tek bobin uyarılı ve yay geri dönüşlü yönlendirme valfleri ile gerçekleştirilmektedir. Tüm elektropnömatik sistem çözümlerinde, çözüm yöntemi her ne olursa olsun mutlaka bir başlangıç şartı bulunmalıdır. Yol-adım diyagramından görüldüğü gibi, ilk çalışma adımı gerçekleştirilmeden önce (A+) kumanda zincirinin, en son adımını gerçekleştirip gerçekleştirmediği sorgulanmalıdır. A+B+B-A- şeklindeki bir çalışma için en son adım A silindirin geri son konumunda (A -) olmasıdır. Bu durumda "a0" sinyal verici eleman uyarılıdır. Çalışmayı başlatma için "start" elemanı uyarılı ve en son adım sinyal vericisi (a0) uyarılmış ise ilk adım çalışması gerçekleşmelidir.

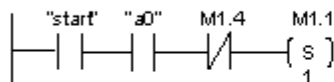
Yay geri dönüşlü valfler, bobin enerjisi var olduğu sürece konumlarını muhafaza ederler. Bu nedenle kumanda zinciri içerisinde valf bobini sinyalleri, RS hafıza fonksiyonlarıyla hafızaya alınmak suretiyle konumlarını koruyabilirler.

A+B+B-A- şeklindeki bir çalışmanın programlanması sembolik adresleme ile gerçekleştirilecekse aşağıdaki sembol tablosu oluşturulmalıdır :

Sembol tablosu :

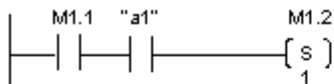
Sembol	Adresi	Açıklama
Start	I0.0	başlatma butonu
a0	I0.1	A silindiri geri son konum algılayıcısı
a1	I0.2	A silindiri ileri son konum algılayıcısı
b0	I0.3	B silindiri geri son konum algılayıcısı
b1	I0.4	B silindiri ileri son konum algılayıcısı
Y1	Q0.0	Y1 valf bobini
Y2	Q0.1	Y2 valf bobini

NETWORK 1 // Start elemanı uyarılmışsa ve en son adım sinyali mevcutsa (a0) ve M1.4 yardımcı rölesi görevini tamamlamışsa A+ hareketi için bir yardımcı röleyi hafızaya al.



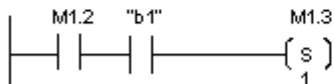
LD	"START"
A	"a0"
AN	M1.4
S	M1.1, 1

NETWORK 2 // Bir önceki adım sinyali mevcutsa (M1.1) ve A silindiri ileri son konumuna ulaşmışsa (a1 uyarılı ise) B+ hareketi için M1.2 yardımcı rölesini set et.



LD	M1.1
A	"a1"
S	M1.2, 1

NETWORK 3 // Bir önceki adım sinyali mevcutsa (M1.2) ve B silindiri ileri son konumuna ulaşmışsa (b1 uyarılı ise) B- hareketi için M1.3 yardımcı rölesini set et.



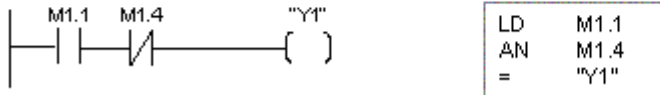
LD	M1.2
A	"b1"
S	M1.3, 1

NETWORK 4 // Bir önceki adım sinyali mevcutsa (M1.3) ve B silindiri geri son konumuna ulaşmışsa (b0 uyanlı ise) A- hareketi için M1.4 yardımcı rölesini set et. Bu adım ardışık kumanda zincirinin en son adımı olduğundan daha önce set edilmiş olan 4 adet yardımcı röleleri aynı anda reset et.

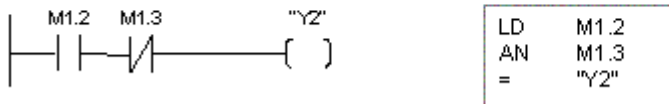


Not : Programlamada 4 adet yardımcı rölenin aynı anda reset edilmesi işleminde tüm yardımcı röleler ayrı ayrı reset edilmek yerine, "R" komutunun altına yazılan 4 rakamı ile M1.1' den M1.4' e kadar olan tüm yardımcı röleler (4 adet) aynı anda reset edilebilmektedir. Bu yöntem, S7-200 PLC cihazları için geçerlidir. Diğer PLC cihazları için ayrı ayrı resetleme işlemi yapılmalıdır.

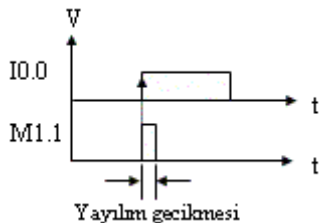
NETWORK 5 // **A + A - hareketi :** Daha önceden hafızaya alınmış olan yardımcı röle sinyalleri, bu bölümde ilgili çıkışlara atanır. A silindirini ileri çıkartacak olan valf bobini Y1'dir. M1.1 yardımcı rölesi set edildiğinde "Y1" enerjilenerek A silindiri ileri son konumunu alır. Y1 valf bobini bu konumunu M1.4 yardımcı rölesi enerjilenene dek korur. M1.4 enerjilendiğinde normalde kapalı olan kontağını açar. Böylece Y1 enerjisiz kalarak A- hareketi sağlanmış olur.



NETWORK 6 // **B + B - hareketi :** B silindirini ileri çıkartacak olan valf bobini Y2'dir. M1.2 yardımcı rölesi set edildiğinde "Y2" enerjilenerek B silindiri ileri son konumunu alır. Y2 valf bobini bu konumunu M1.3 yardımcı rölesi enerjilenene dek korur. M1.3 enerjilendiğinde normalde kapalı olan kontağını açar. Böylece Y2 enerjisiz kalarak B- hareketi sağlanmış olur.



3 fazlı bir asenkron motor normalde açık, uyarıldığında kapanan tek bir butonla hem çalıştırılıp (set), hem de durdurulmak (reset) isteniyor:



Butona bir kez basılıp çekildiğinde motor çalışacak, aynı butona ikinci bir kez basılıp çekildiğinde ise motor duracaktır. Ayrıca motor termik bir röle üzerinden aşırı akımlardan korunacaktır.

Çözüm :

Problemin birden fazla çözümü mevcuttur. Örneğin S7-PLC içerisinde kullanıma hazır olarak bulunan, yükselen kenar tetikleme sinyali komutunu (EU) kullanmak suretiyle çözüme ulaşılabilir.

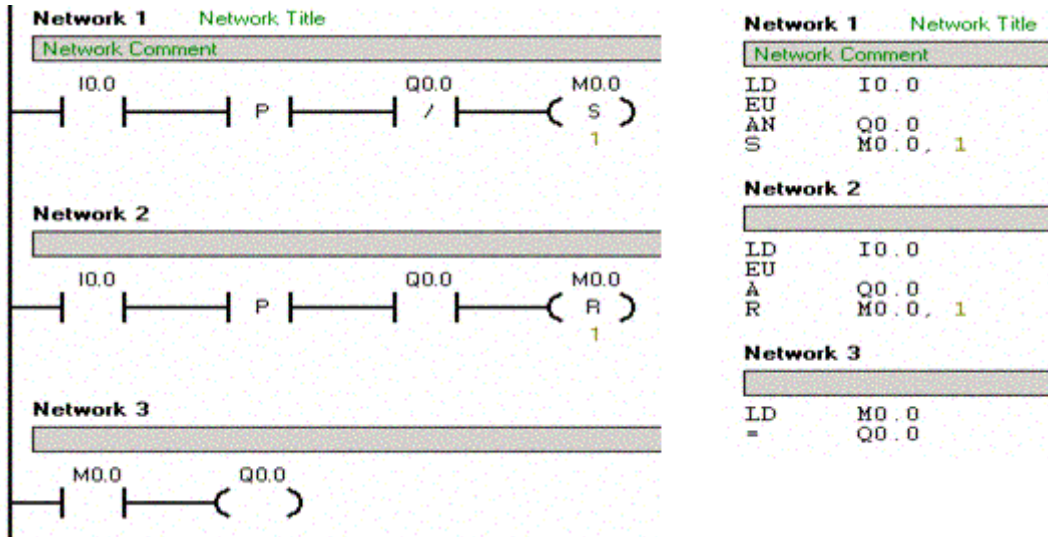
Problemin çözümü için, kesinlikle elektrik kumanda devreleri mantığından uzaklaşmış olmak gereklidir. Çünkü I0.0 buton elemanı, Q0.0 çıkışını hem set, hem de reset etmelidir. Bu nedenle set ve reset işlem komutları alt alta yazıldığında, bir çevrimde set işlemi, ikinci veya sonraki herhangi bir çevrimde ise reset işlemi gerçekleşmelidir. Bu nedenle set ve reset işlemlerini farklı çevrimlere yaymak gereklidir.

Pozitif yükselen kenar komutu (EU), I0.0 girişi kapalı kaldığı sürece sadece tek bir çevrim için sinyal üretir. Bundan sonraki çevrimlerde sinyal üretmez.

I0.0 giriş adresi =1 yapıldığında ilk çevrimde sadece set komutu türütülür. Q0.0 çıkış adresi = 1 olur.

I0.0 giriş adresi ikinci bir kez uyarıldığında, ilk çevrimden dolayı açılan Q0.0 kontağı M0.0 bitini tekrar set edemez.

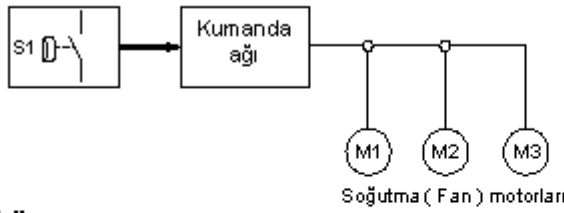
Ancak alt satırda kapanan Q0.0 kontağı bu defa reset işlemi yaptırır.



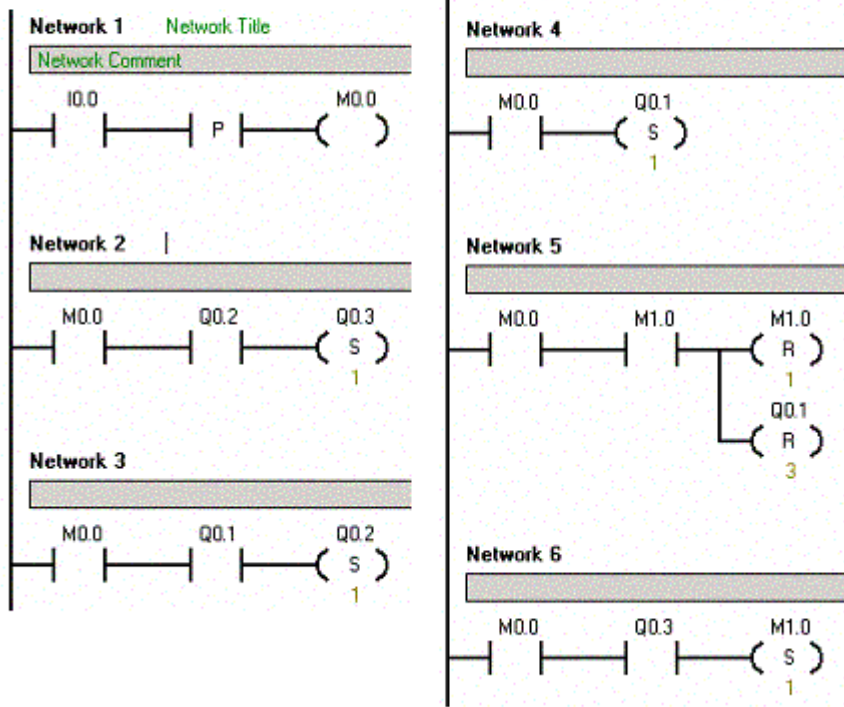
Normalde açık fakat uyarıldığında kapanan bir buton ile 3 adet soğutucu motoru (fan) şu şekilde çalıştırılmak isteniyor:

S1 butonu bir kez uyarıldığında birinci fan motoru (M1) sürekli çalışmaya başlıyor. S1 butonu ikinci kez uyarıldığında ikinci fan motoru (M2), buton üçüncü kez uyarıldığında ise üçüncü fan motoru (M3) sürekli çalışmaya başlıyor. S1 dördüncü kez uyarıldığında çalışmakta olan tüm motorlar duracaktır (reset).

Uygulama planı :



Çözüm :



Zaman fonksiyonunun oluşturulması kumanda tekniğinde temel lojik fonksiyonlardandır. Zaman elemanlarının görevi; bir giriş sinyaline karşı zamana bağımlı çıkış sinyali üretmektir. PLC içinde zamanlayıcılar numaralandırılmışlardır. Üretici firmaya göre çeşitli sayı ve fonksiyonlara sahip zaman elemanları (timer) hazır bir şekilde kullanıma sunulmuştur. Kullanım için sadece basit bir programlama yeterlidir.

Zaman elemanlarının intern olarak nasıl çalıştığı normal şartlarda programcıyı ilgilendirmez. Çünkü kullanım için önemli olan zaman elemanının fonksiyonudur. PLC cihazlarında genelde dijital zaman elemanları kullanılmaktadır. Bu prensibe göre intern olarak oluşturulan sayma darbeleri bir geri sayıcıya iletilir.

Öncelikle zamanlayıcıya start sinyalinin verilmesiyle sayma değeri yüklenmiş olur. Zamanlayıcı, intern sayma darbeleri ile sıfıra ulaştığında zaman bitmiştir.

PLC içersindeki zamanlayıcıyı kullanabilmek için zamanlayıcı ile ilgili parametrelerin tanımlanması gereklidir.

ZAMANLAYICI PARAMETRELERİ:

- **Zamanlayıcı numarası** : PLC içersindeki kaç numaralı zamanlayıcının kullanılacağı belirtilmelidir. Örneğin, S7-200 CPU 212 tip PLC iç bünyesinde

üretici firma tarafından hazır olarak kullanıma sunulmuş 0-63 arasında toplam 64 adet zamanlayıcı bulunmaktadır. Programlamada bunlardan hangisinin kullanılacağı belirtilmelidir. Ayrıca zamanlayıcılar, bir program içerisinde birden fazla programlanmamalıdır. Örneğin bir program içinde iki adet zamanlayıcı kullanmak gerektiğinde, her iki zamanlayıcıya da aynı numaralar verilmemelidir. Örneğin birinci zamanlayıcı T38 ise ikincisi T39 veya başka bir adres numarası olmalıdır.

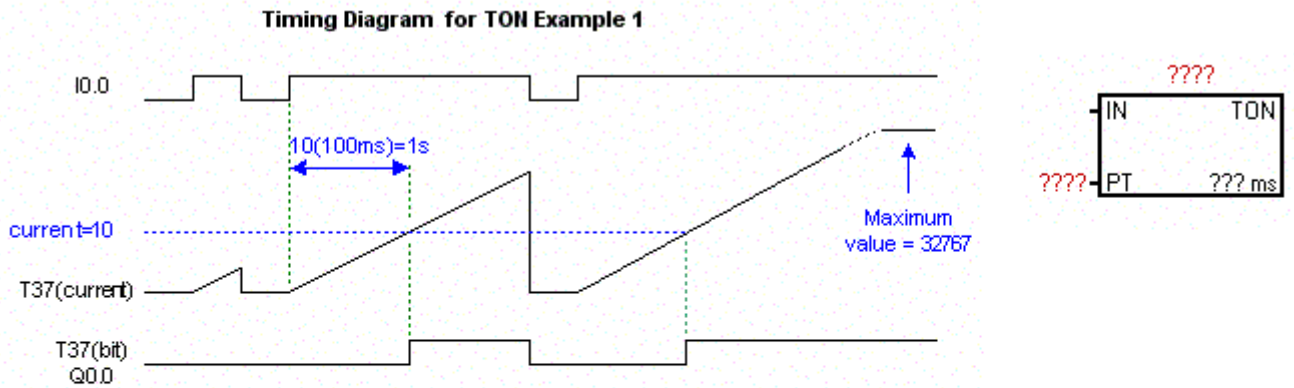
- **Zamanlayıcının türü** : PLC marka ve modeline göre cihazların iç bünyelerinde sahip oldukları zamanlayıcılar değişik olabilir. Ancak temelde her PLC cihazında bulunabilecek iki tip zamanlayıcı mevcuttur.

1- Kapamada gecikmeli zamanlayıcı.

2- Çekmede gecikmeli zamanlayıcı.

- **Zaman değeri** : Zamanlayıcıların programlanmasında milisaniye, saniye veya dakika olarak zaman değerlerinden herhangi birisi girilmelidir. Bu kullanıcıya ait bir seçimdir.
- **Reset** : Zamanlayıcının ne şekilde reset edileceği zamanlayıcının türüne göre değişebilir. Örneğin, kapama gecikmesi yapan bir zamanlayıcıda ayrıca bir reset girişi bulunmaz. Resetleme işlemi, zamanlamayı başlatma sinyalinin kesilmesiyle yapılır. Buna karşılık açma gecikmesi yapan bir zamanlayıcıda ise ayrıca bir reset girişi bulunmaktadır. Bu hususların daha önceden bilinmesi gereklidir.

Aşağıda kapamada gecikmeli zamanlayıcının programlanabilmesinde gerekli olan çalışma grafiği verilmiştir.

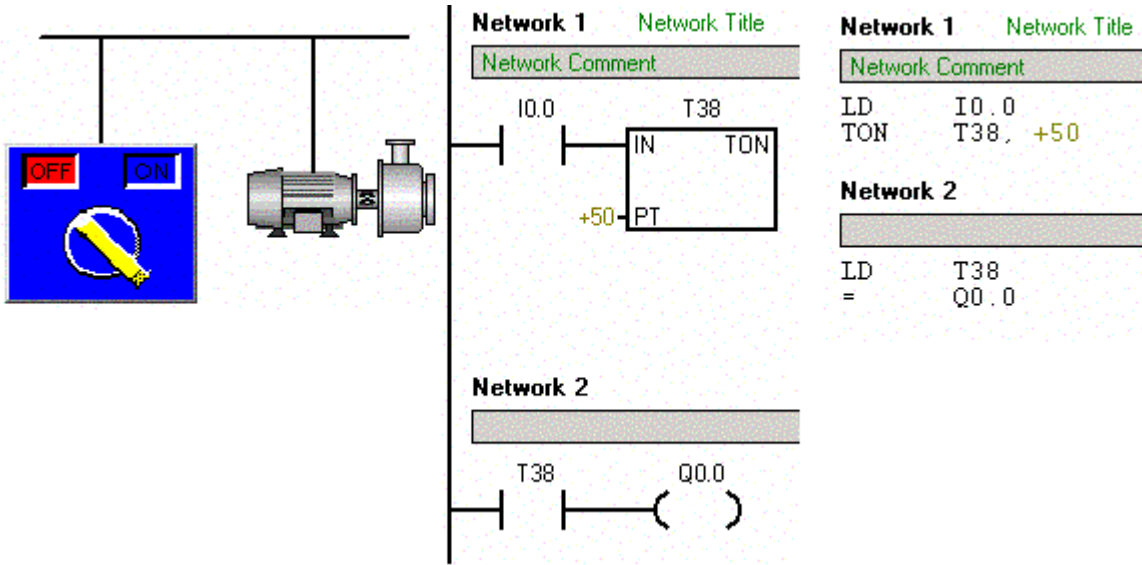


TON, TOF tipi zamanlayıcılar :

1 ms	32.767 s	T32, T96
10 ms	327.67 s	T33-T36, T97-T100
100 ms	3276.7 s	T37-T63, T101-T255

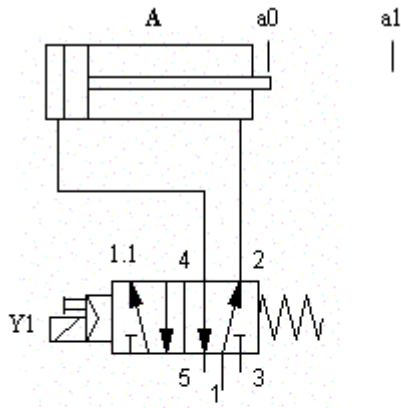
TON zamanlayıcı uygulaması -1:

PLC I0.0 giriş adresine bağlı şalter uyarıldıktan 5 saniye sonra motor çalışacaktır. Şalter kapalı kaldığı sürece motor çalışmasını sürdürmelidir. Şalter=0 yapıldığında motor durmalıdır.



TON zamanlayıcı uygulaması –2 :

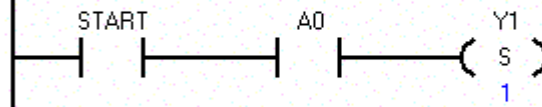
I0.7 girişi impuls darbesiyle uyarılacak. A silindiri A+ hareketini yaparak dışarı hareket ederek bir koltuğa 3 saniye botunca kuvvet uyguladıktan sonra geri dönecektir. I0.7 girişi sürekli uyarılı tutulduğu sürece bu çevrim tekrarlanacaktır.



	Symbol	Address
1	START	I0.7
2	A0	I0.0
3	A1	I0.1
4	Y1	Q0.1
5		

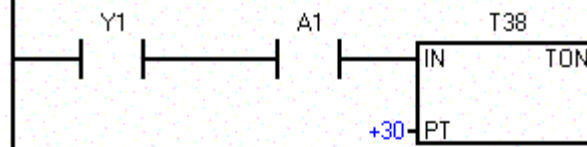
Network 1 Network Title

Network Comment



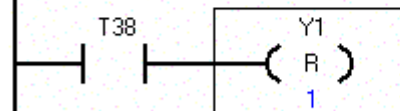
Symbol	Address	Comment
A0	I0.0	
START	I0.7	
Y1	Q0.1	

Network 2



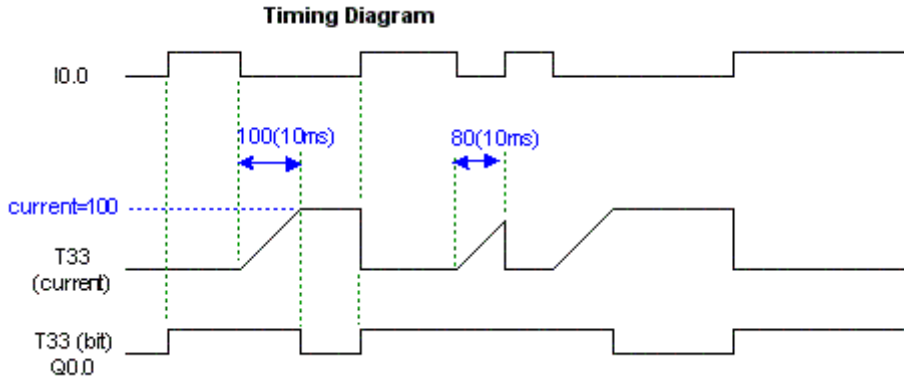
Symbol	Address	Comment
A1	I0.1	
Y1	Q0.1	

Network 3



Symbol	Address
Y1	Q0.1

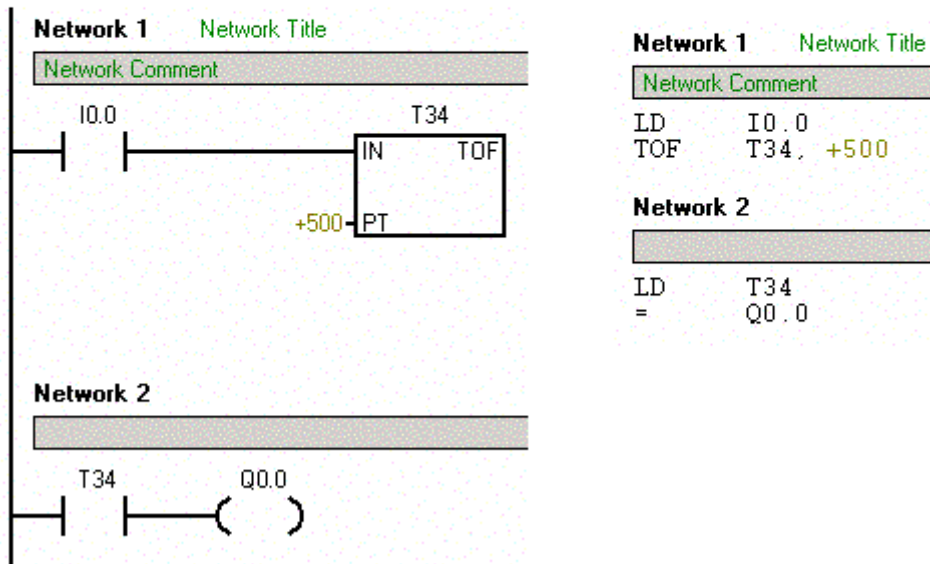
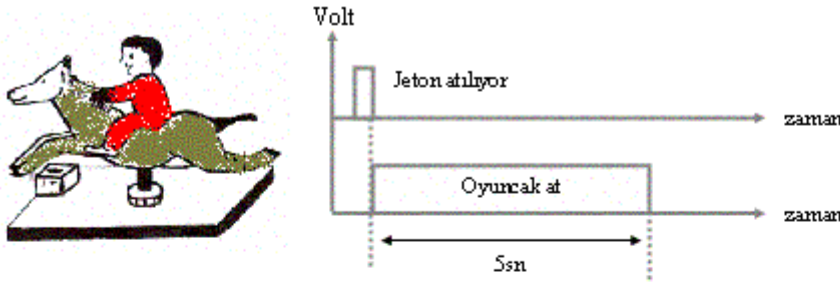
TOF TİPİ ZAMANLAYICI PARAMETRELERİ



TOF zaman parametreleri:

1 ms	32.767 s	T32, T96
10ms	327.67 s	T33-T36, T97-T100
100 ms	3276.7 s	T37-T63, T101-T255

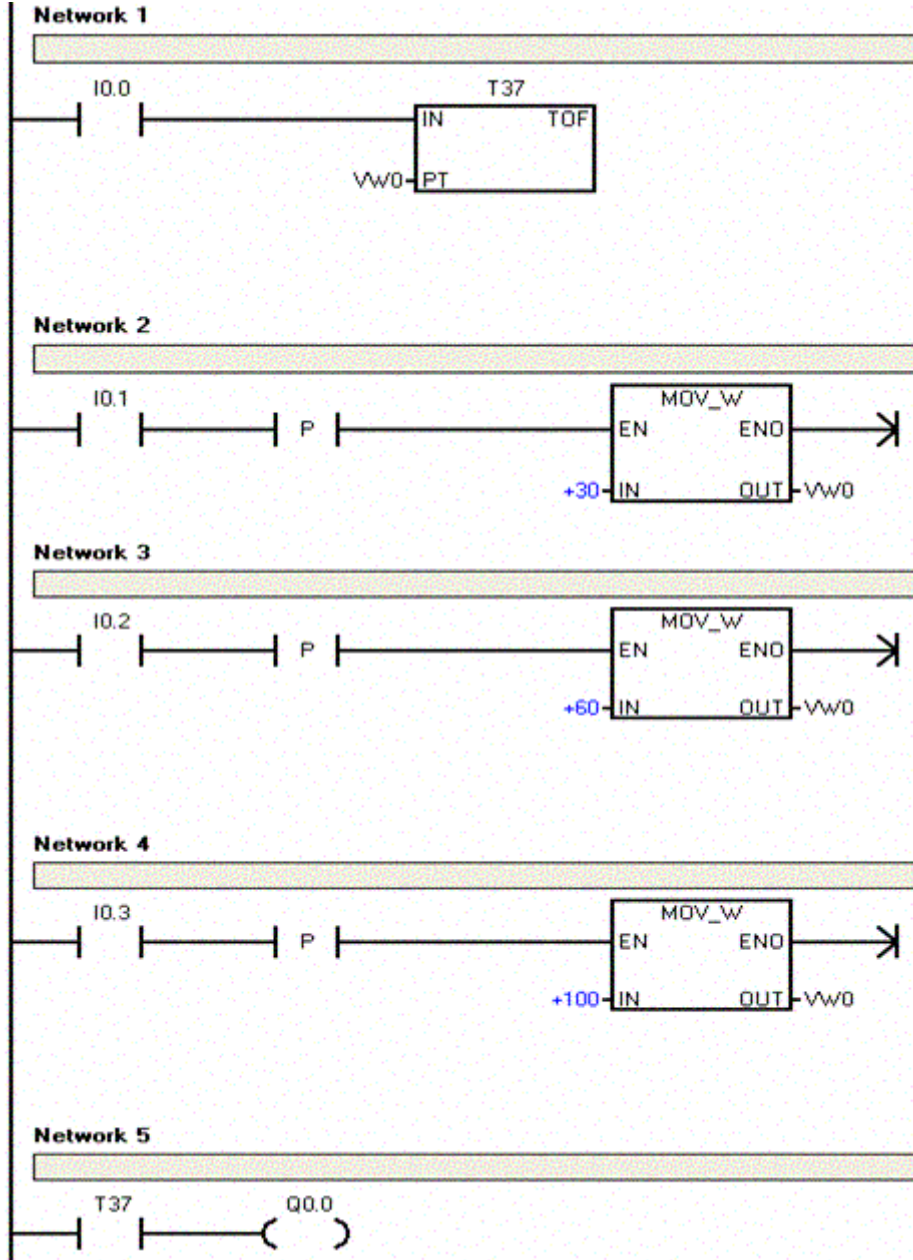
UYGULAMA-1 : Bir lunaparktaki oyuncak atın şu şekilde çalışması isteniyor: Jeton atılıyor. Oyuncak at 5 saniye süreyle çalışıp duruyor.



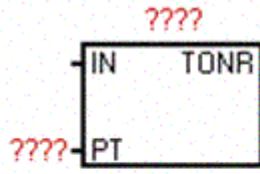
UYGULAMA-1 : Bir band sisteminin aşağıdaki gibi çalışması istenmektedir.

- PLC IO.0 adres girişi uyarıldığında Q0.0 çıkışına bağlı motor hemen çalışacaktır.

- PLC nin 3 input girişine değişik zaman değerleri atanacaktır.
- Örneğin ;
 I0.1 = 1 ise, 3 saniye
 I0.2 = 1 ise, 6 saniye
 I0.3 = 1 ise, 10 saniye süreyle band motoru çalışıp durmalıdır.

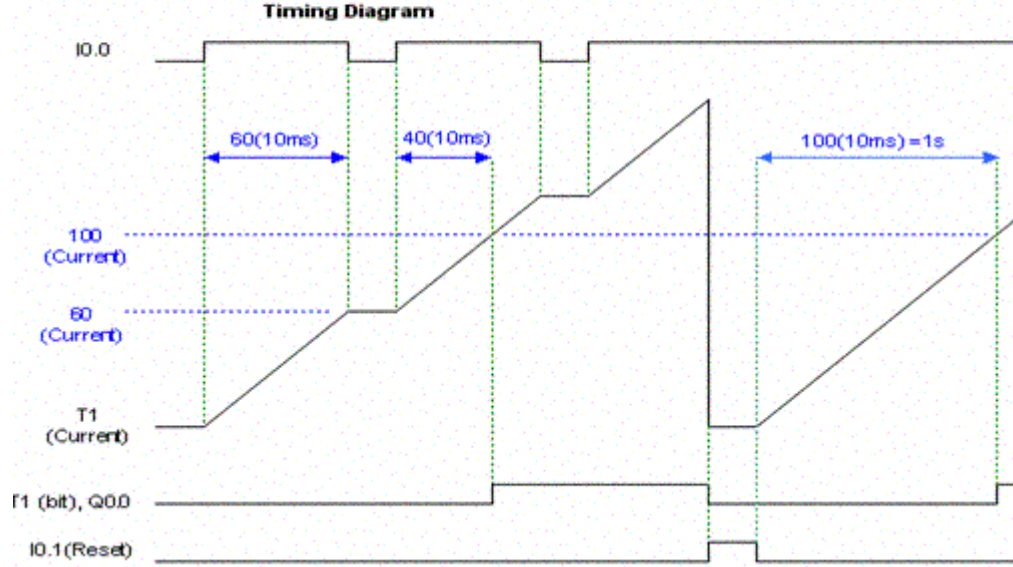


TONR TİPİ ZAMANLAYICI PARAMETRELERİ

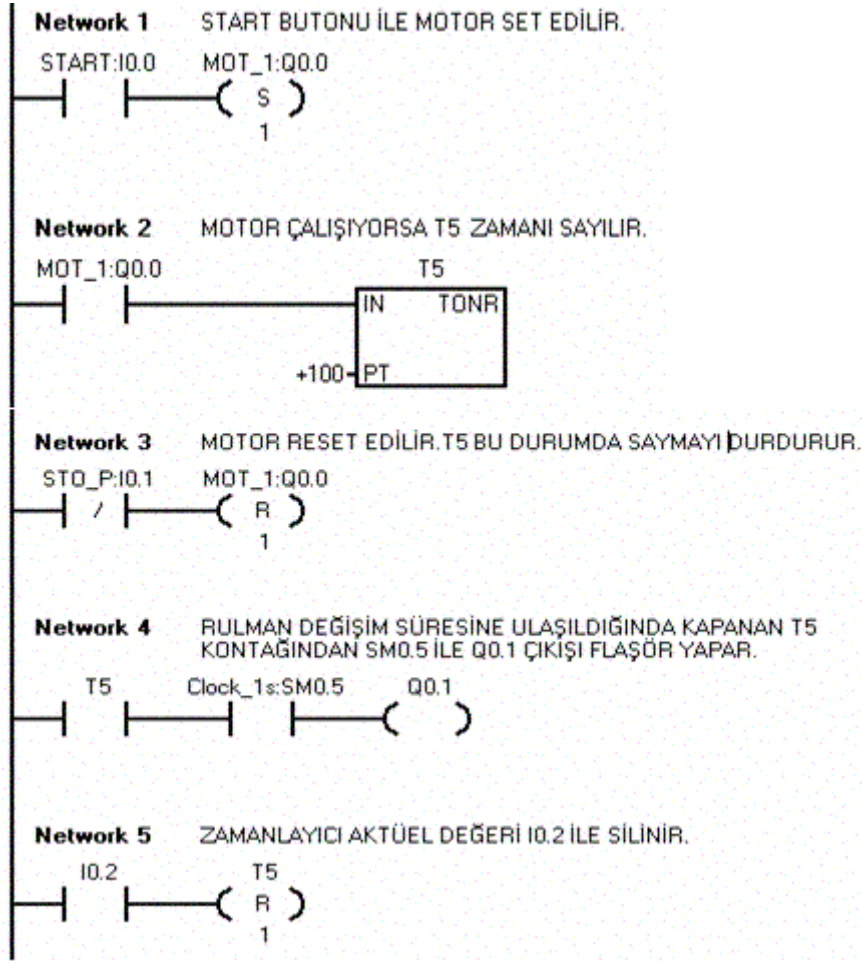


TONR zaman parametreleri :

1 ms	32.767 s	T0, T64
10 ms	327.67 s	T1-T4, T65-T68
100 ms	3276.7 s	T5-T31, T69-T95



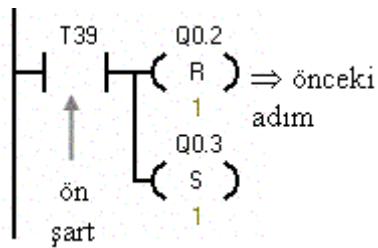
Uygulama: 3fazlı asenkron motorun çalışma süresine bağlı olarak rulmanları belli bir süreden sonra değiştirilecektir. Bu nedenle sadece motorun çalıştığı süreler tespit edilmelidir. Rulman değişim süresine gelindiği flaşör yapan bir lamba ile bildirilmelidir.



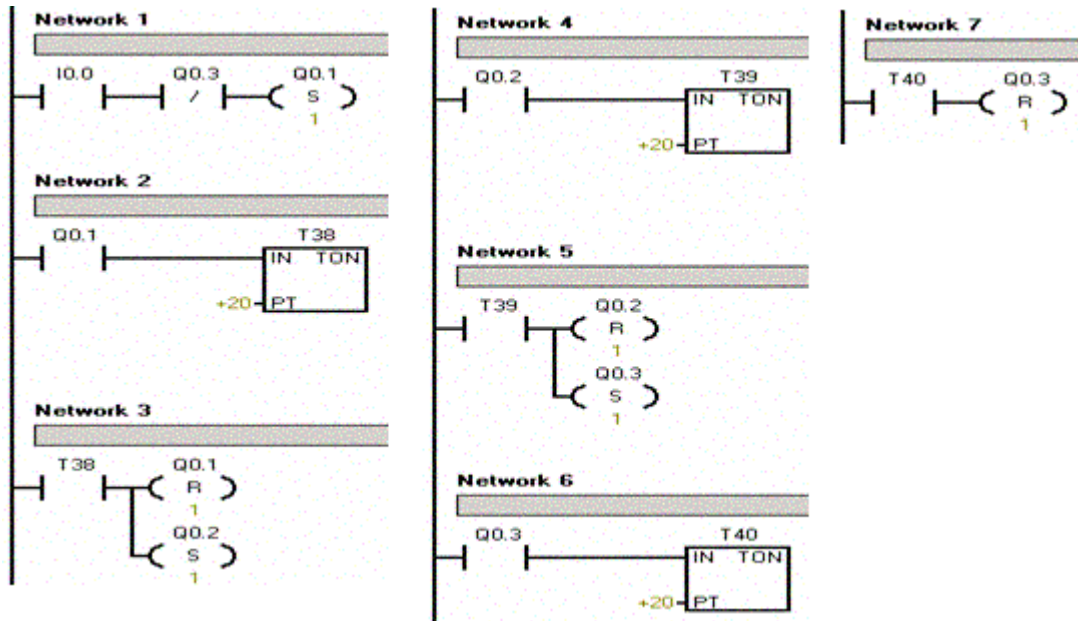
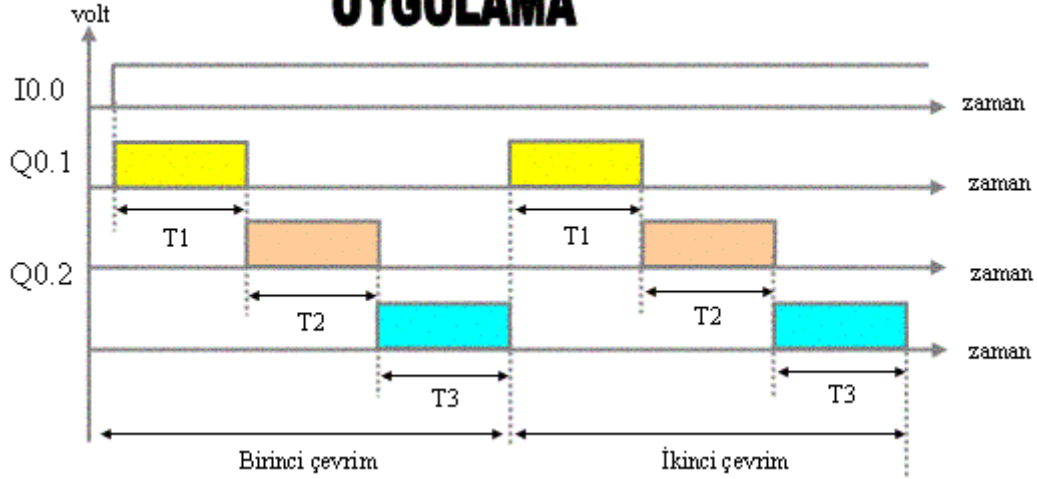
ZAMANA BAĞIMLI ARDIŞIK KUMANDA TEKNİĞİ VE UYGULAMALARI

Zamana bağımlı ardışık kumanda tekniğinde, bir sonraki işleme geçilmeden önce bir önceki işlemin gerçekleşip gerçekleşmediği sorgulanmalıdır. Eğer bir önceki işlem gerçekleşmişse ve ilgili zamanlayıcıya ait kontak kapanmışsa bir sonraki işlem gerçekleştirilir.

Aşağıda zamana bağımlı ardışık kumanda prensibi görülmektedir.



UYGULAMA



Endüstride üretilen ürünlere ait miktarların belirlenmesi çoğu zaman darbelerin sayılması yöntemi ile gerçekleştirilmektedir. Bunun için sayma darbeleri bir sayıcıya uygulanır. Sayıcı içindeki sayma darbeleri 16 bitlik (1 word) bir alandır.

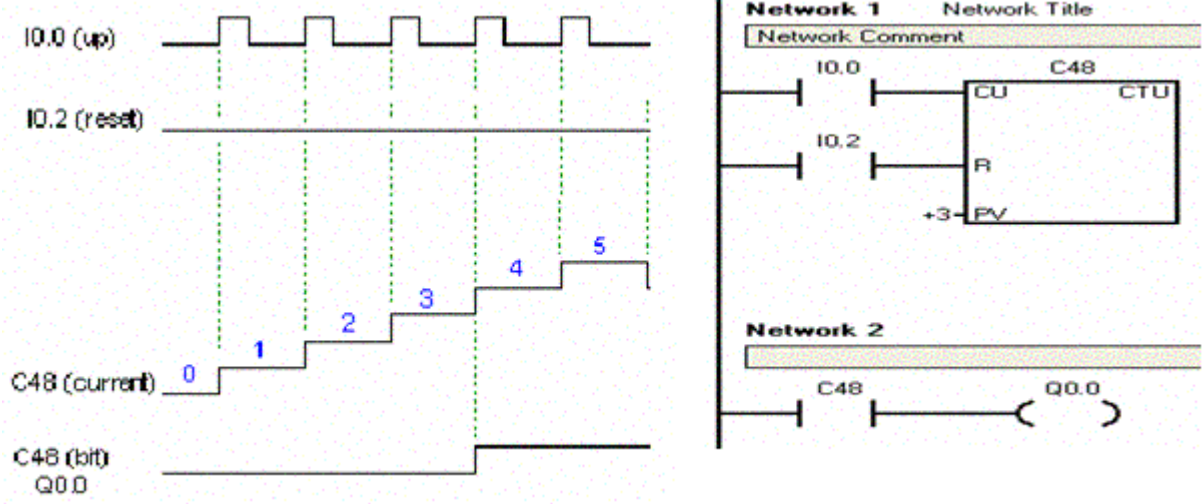
Sayıcı ; her 0'dan 1'e geçiş olgusunda sayan bir elemandır.

S7 200 PLC de 3 tip sayıcı bulunmaktadır.

- CTU : Yukarı sayıcı
- CTD : Aşağı sayıcı
- CTDU : Aşağı-Yukarı sayıcı

SIEMENS S7-200 CPU 221, 222, 224 model PLC cihazlarında kullanıma hazır 0-255 arasında toplam 256 adet sayıcı bulunur.

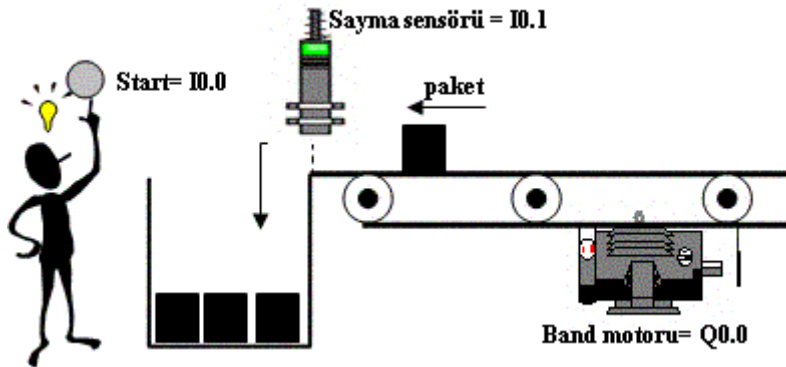
Yukarı Say komutu (CTU) sayma (CU) girişinin her yükselen kenarında anlık değeri bir artırır. Anlık değer (CV) ayar değerine (PV) eşit veya büyükse çıkış biti (Q) 1 olur. Reset girişi (R) geldiğinde sayıcı sıfırlanır.

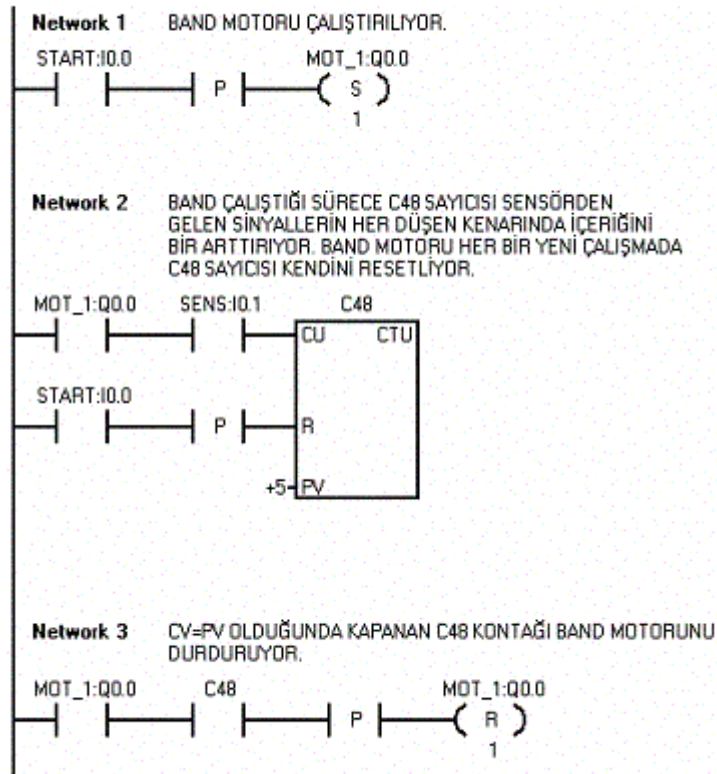


İstenilen değere ulaşıldığında sayıcı iç biti aktif olur. Örneğin $C48=1$ olduğunda ne yapılacağı tamamen kullanıcıya ait bir durumdur.

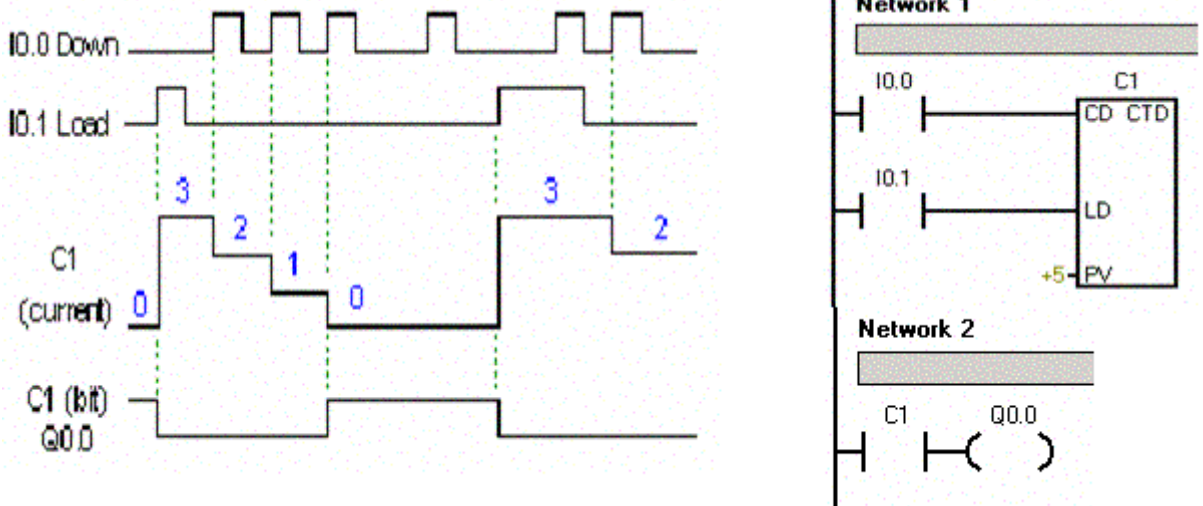
UYGULAMA-1: $PV=CV$ ise durdur.

Daha önce başlatılmış bulunan bir süreç, sayıcının istenilen değere ulaşmasıyla sona erdirilebilir. Start butonu ile band motoru çalıştırılıyor. Sayıcı sayma değeri ayar değerine eşit olduğunda ($CV=PV$) band motoru durduruluyor



Çözüm:

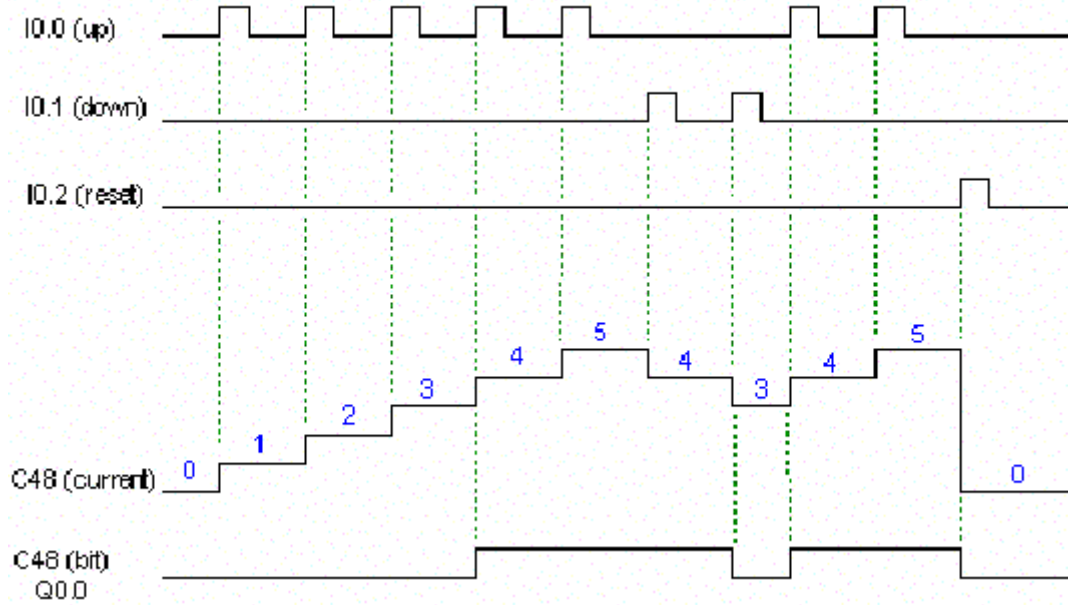
Aşağı Say komutu (CTD), yukarı sayma girişinin (CD) her yükselen kenarında (0'dan 1'e dönüşümünde) anlık sayma değerini bir azaltır. Cxx anlık değeri 0'a eşitse Cxx biti set olur. LD (Load) girişi geldiğinde sayıcı biti sıfırlanır ve anlık değer PV değerine eşit yapılır. Sıfıra ulaşıldığında sayma işlemi durur (ve Cxx biti set olur).



Yukarı/Aşağı Say komutu (CTUD), yukarı sayma girişinin (CU) her yükselen kenarında yukarı sayar, aşağı sayma girişinin (CD) her yükselen kenarında ise aşağı sayar. Sayıcının anlık değeri Cxx o ana kadarki sayılan değeri saklar. Sayma işlemi yapıldığı anda anlık değer ile ayar değeri PV karşılaştırılır.

Maksimum değere (32767) erişildiğinde yeni bir yukarı sayma girişi anlık değerin minimum değere dönmesine neden olur (-32768). Aynı şekilde, minimum değere ulaşıldıktan sonraki aşağı sayma giriş sinyali anlık değerin maksimum değer (32767) olmasına neden olur.

Anlık değer Cxx ayar değeri PV'ye eşit veya büyükse Cxx biti set olur. Diğer durumda sıfırdır. Reset (R) girişi geldiğinde veya Reset komutu işlendiğinde sayıcı sıfırlanır. CTUD sayıcısı PV değerine ulaştığında sayma işlemi durur.



Tip	Çalışma Şekli	Sayıcı Biti	İlk Taramada
CTU	CU anlık değeri artırır. Anlık değer 32767'ye kadar artırılabilir.	Sayıcı biti şu durumda 1 olur: Anlık değer \geq Ayar değeri	Sayıcı biti sıfırdır. Anlık değer saklanabilir. ¹
CTUD	CU anlık değeri artırır. CD anlık değeri azaltır. Anlık değer sayıcı resetleninceye kadar artırılıp azaltılabilir.	Sayıcı biti şu durumda 1 olur: Anlık değer \geq Ayar değeri	Sayıcı biti sıfırdır. Anlık değer saklanabilir. ¹
CTD	CD anlık değeri 0 oluncaya kadar azalır.	Sayıcı biti şu durumda 1 olur: Anlık değer = 0	Sayıcı biti sıfırdır. Anlık değer saklanabilir. ¹

