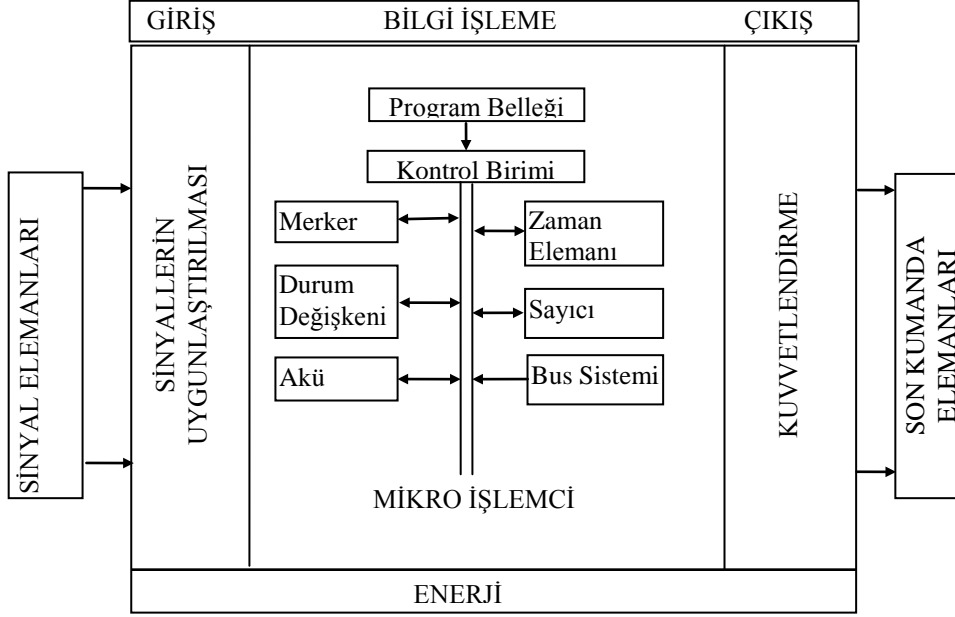


## PLC' İN ÇALIŞMASI VE FONKSİYONU



Bir kumanda cihazı temel prensip olarak üç bölümden oluşur. Birincisi merkezi işlem bölümüdür. İkincisi bütün sinyal elemanlarının bağlandığı giriş bölümü, üçüncüsü de son kumanda elemanlarının bağlandığı çıkış bölümüdür.

Genelde her PLC ilk önce prizden aldığımız 220 volt'u CPU için 5 Volta indirgeyen bir modüle sahiptir. Giriş ve çıkış bölümleri için genelde 24 Volt kullanılmakla beraber bu değer 220 Volta kadar değişmektedir.

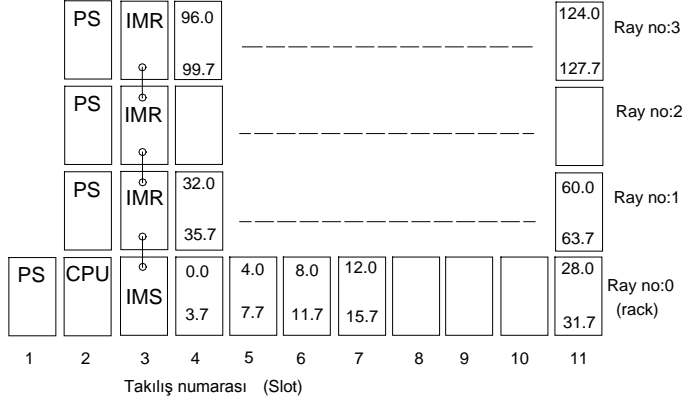
Girişler Opto-Coupler (opto izolatör) ile CPU' dan galvanik olarak ayrılmıştır. Aynı şekilde çıkışlar da Opto-Coupler veya röle ile CPU' dan galvanik olarak ayrılmıştır. Çıkışların kullanımda aşırı yüklenmesi durumunda üretici firmanın uyarılarına dikkat edilmesi gerekir.

Daha sonraki konularda açıklanacağı gibi, sensörlerden gelen sinyallerin durumu ( 0/1 şeklinde) dikkate alınarak büyük bir tabloya benzetebileceğimiz bütün sinyallerin durumunu gösteren proses giriş resmi (PAE) oluşturulur. Aynı şekilde çıkışlar için de proses çıkış resmi (PAA) mevcuttur.

Bellekteki programın işlenmesinde mikroişlemci bir adres sayıcısı sayesinde her seferinde bir bellek adresinden bir emri alır ve onu işler. İşlem sona erince adres sayıcısı 1 adım yükseltmek suretiyle ikinci bir emir alınır. Bu işlem, normal şartlarda hiç ara vermeksizin devam eder.

Aynı zamanda mikroişlemci SİSTEM BUS'u sayesinde zaman elemanı, sayıcı, proses giriş resmi, durum tespit işaretleri vb. elemanlarla haberleşir ve onları programın işlenmesinde kullanır. Bunu yaparken de devamlı olarak bu elemanların sinyal durumlarının ( 0/1) ne olduğunu sorgular.

S7-300 sistemlerine maksimum 32 modül eklenebilir. Her montaj rayı 8 modül taşıyabilir. 4 ayrı montaj rayı montaj edilebilir. Rayların kendi aralarında haberleşmesini sağlamak amacıyla haberleşme birimine ( IM ) ihtiyaç vardır.



PS =Güç kaynağı (Power Supply)

CPU =Merkezi işlem birimi (Central Process Unit)

IMS =Gönderici arabirim modülü(Interface Modul Sender)

IMR =Alıcı arabirim modülü (Interface Modul Receive)

4.slot numarasından itibaren sinyal (Dijital, analog – giriş, çıkış) , haberleşme ve fonksiyon modüllerinin takılabileceği alandır. Her modül için bir DWORD' luk ( 32 bit )alan rezerve edilmiştir.

Sadece 0. montaj rayı kullanılacaksa 3. slot boş bırakılacaktır.

1,2 ve 3 nolu raylarda 2 nolu slot boşdur. ( PS yine 1.slot'a takılmıştır.)

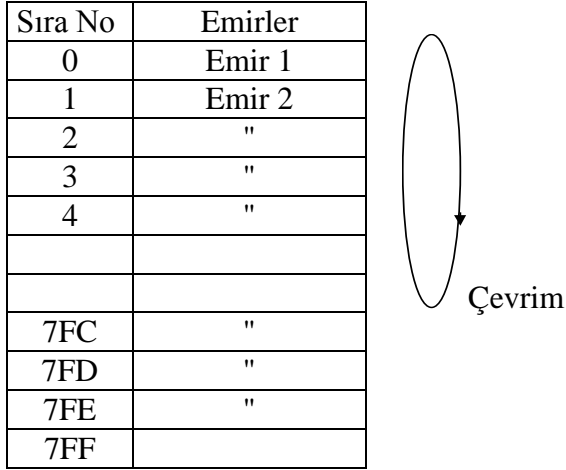
## PROGRAM İŞLEME ŞEKİLLERİ

### 1. Lineer Program işleme

Emirler program belleğinde buldukları sıraya göre işlenirler. Program sona erdiğinde aynı işlem tekrarlanır. Yani sürekli bir çevrim söz konusudur.

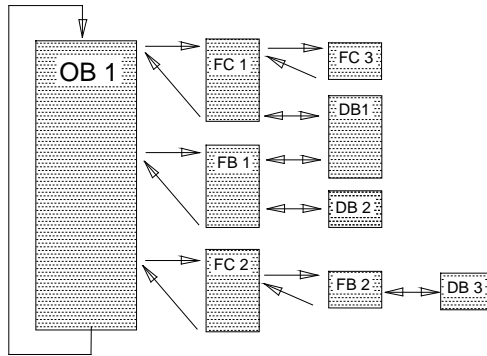
Bir programdaki bütün emirlerin bir kez işlenmesi için gerekli zamana çevrim süresi adı verilir. Bu süre programın içerdiği emirlerin sayısına ve türlerine bağlıdır. Dikkat edilmesi gereken husus, bu sürenin mümkün olduğunca kısa tutulmasıdır.

Lineer program işleme genelde basit ve kapsamlı olmayan programlar için kullanılır.



### 2. Yapısal Program işlenmesi

Özellikle kapsamlı problemlerde kullanılır. Program mantıklı olarak fonksiyonuna göre küçük modüllere ayrılır ( FB, FC ler). Programlarda kullanılacak verilerin saklandığı modüller ( DB ) ve bütün bu alt programların üzerinde bu programları belirli bir sıraya göre çağırarak bir organizasyon programı ( OB 1 ) oluşturulur.



**S7-300 de PROJE OLUŞTURMA**

Not: Aşağıdaki anlatımlarda kullanılan

"X" : Mausun sol tuşu ile tek tıklamayı

→ : Bir alt menüye geçişi ifade etmektedir.

Markalama ise mausun sol tuşu ile tek tıklanarak o elemanın farklı bir renge boyanması demektir.

**3. Donanım tanıtılmadan ( IFM Modüller için geçerlidir)**

A- Yeni bir klasör açmak

**Bilgisayarım XX →, ( C ) X, Yeni →, Klasör X** ( Yeni klasör yerine isim yazılır )

B- SIMATIC MANAGER dosyası açmak

a) **Simatik manager XX, Datei X, Neu →, Projekt X** (Proje penceresine yeni isim yazılır )

b) "**Ablegeort**" penceresine ( C:\ ..... ) kendinizin verdiği klasör adı yazılacak ve onaylanacak.

c) Bizim yazdığımız dosya adında yeni bir pencere açıldı.

1- **Einfügen →, station →, Simatic 300 station X**, (Dosya adımızın başında (+) oluştu), ( + ) X,

2- + **Simatic 300 Station** açıldı

**Einfügen →, Program →, S7 Program X**,

3- Listemizin altına ( + **S7 Program (1)** ) açıldı

( + ) X,

4- Listemize "**Quellen**" ve "**Bausteine**" eklendi

**Bausteine XX**, ( Yan taraftaki pencereye OB1 yazıldı. Programımızı sadece OB1 içerisinde yazacaksak OB1 çift tıklanarak açılır ve program buraya yazılır. )

5- Diğer blokları kullanmamız gerekiyor ise, (blokların ne zaman ve nasıl kullanılacağı ileri konularda anlatılacaktır.

**Einfügen →, S7 Baustein →**, Eklenmek istenen blok ( FB, FC, DB vs ) üzerine X,

6- Açılan pencerede gerekli ayarlamalar ( Modül numarası, programcıya ait bilgiler, program yazım şekli v.b.) yapıp tamam tuşuna basılacak.

7- Yan penceremize yeni oluşturduğumuz bloklar açıldı. Hangi blokta çalışmak istiyorsak o blok üzeri çift tıklanarak çalışma sayfamız açılmış olur. ( Blokların açılmasında önce çağrılan, sonra çağırılan blok açılmalıdır.)

Yukarıdaki dosya açma şekli IFM modellerde geçerlidir. IFM olmayan modellerde PLC' e ait donanımın tanıtılması gereklidir. Bunun için;

**4. Donanım tanıtılarak**

A- Yeni bir klasör açmak

**Bilgisayarım XX →, ( C ) X, Yeni →, Klasör X** ( Yeni klasör yerine isim yazılır ) Bunun amacı bütün çalışmaların aynı klasör altında toplanmasını, kolay taşınmasını ve silinmesini sağlamak içindir.

B- SIMATIC MANAGER dosyası açmak

a) **Simatik manager XX, Datei X, Neu →, Projekt X** (Proje penceresine yeni isim yazılır "DENE")

b) "**Ablegeort**" penceresine ( C:\ ..... ) kendinizin verdiği klasör adı yazılacak ve onaylanacak

- c) Bizim yazdığımız dosya adında (DENE) yeni bir pencere açıldı.
- 1) **Einfügen →, station →, Simatic 300 station X**, (Dosya adımızın başında (+) oluştu) ( + ) X,
  - 2) **Simatic 300 üzeri X**,
  - 3) **Bearbeiten →, Objekt öfnen X**,
  - 4) **"Hardware Konfiguieren"** penceresi açıldı **Ansicht →, Katalog X**,
  - 5) + **Simatic 300 X**,
  - 6) + **Rack 300 X**,
  - 7) **Profilschine XX**,
  - 8) Açılan pencerenin 1 No'lu alanına güç kaynağı ile ilgili bilgiler yazılacak. 1. No'lu alan markalanarak  
 + **PS 300 X**,  
**PS 307 5A XX**, (Bu değer 1 no'lu alana eklendi.)
  - 9) 2. No'lu alana CPU ile ilgili bilgiler yazılacak.  
 + **CPU 300 X**,  
**CPU 314 IFM XX**, (Elimizdeki CPU özelliklerine göre "5AE01-0AB0" ) (Bu değer 2 No'lu alana eklendi.)
  - 10) 3. No'lu alan daha sonra eklenebilecek (Haberleşme "IM") modüller için boş bırakılacaktır.
  - 11) 4. No'lu alana PLC özelliğine göre giriş ve çıkış v.b. modüller eklenebilir fakat CPU 314 IFM bunlar yazılım olarak tanıtılmıştır. Eğer gerekliyse;  
 + **SM 300 X** den istenilen özellikteki modüller bulunarak çift tıklanır.
  - 12) Gerekli bilgiler eklendikten sonra hafızaya alınır ve CPU ya transfer edilir.  
**Station → Speichern X**, veya hafıza ikonundan.  
**Zielsystem →Laden** veya **laden ikonundan**  
**"Hardware konfiguieren"** penceresi kapatılarak **"Simatic Manager"** penceresine dönülür.
  - 13) + **Simatic 300 station X**,  
 + **CPU 314 IFM (1) X**,  
 + **S7 program (1) X**,  
 + **Bausteine üzerine X**,
  - 14) Yan pencerede OB1 açıldı diğer blokları açmak için  
**Einfügen →,S7 Baustein →**, ilgili blok ismi tıklanacak.
- Yukarıda anlatılan yöntemler projenin programlama cihazında, Off-line konumunda ve programcının donanımı kendisinin kurması ve bunu PLC'e yüklemesi durumunda takip edilecek yöntemlerdir.
- Ancak PLC'de yüklü bir projenin olması durumunda, projeye girilmesi, değişiklik yapılması, donanımın öğrenilmesi gibi işlemlerin yapılabilmesi için PLC'deki projenin programlama cihazına alınması gerekir. Örneğin bir işletmede çalışan bir sisteme ait PLC'e müdahale etmek gerektiğinde. Bunun için yapılması gereken işlemler.

1- **Simatik manager XX, Datei X, Neu →, Projekt X** (Proje penceresine yeni isim yazılır )

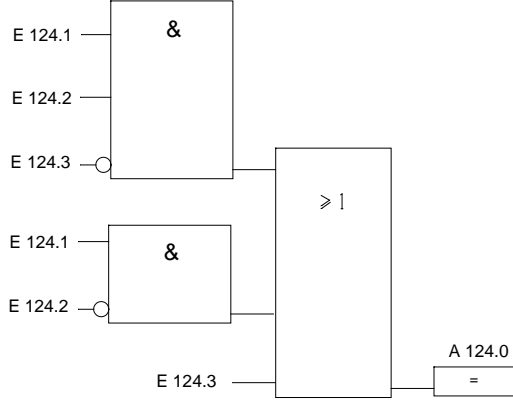
- 2- "**Ablegeort**" penceresine ( C:\ ..... ) kendinizin verdiđi klasör adı yazılacak ve onaylanacak
- 3- Bizim yazdığımız dosya adında yeni bir pencere açıldı.
- 4- **Einfügen** →, **station** →, **Simatic 300 station X**, markalanır ve sağ pencerede "**Hardware**" çift tıklanır.
- 5- Oluşan "**Hardwarekonfigurieren**" penceresinde **Zielsystem** → **Laden in PG** gelen pencerede bizim verdiğimiz proje adı seçilir ve onaylanır.
- 6- Oluşturulan proje bir ağ sisteminin elemanı olabilir, bunun için montaj rayı numarası, slot numarası, MPI adresi seçilmelidir. Biz henüz bir ağ sistemi oluşturmadığımız için verilen adresler kontrol edilerek onaylanır.
- 7- Öğrenmek istediğimiz PLC sistemine ait bütün modüller adresleri ile birlikte belirlenmiş olur. Modüllere ait ayrıntılı bilgi **Ansicht** → **Detailsicht** den alınabilir.

## MANTIK KAPILARI KOMBİNASYONLARI

Kumanda programlarında sadece "VEYA", "VE", "DEĞİL" kapılarının kullanılması söz konusu değildir. Gerekli fonksiyonlar çeşitli kapıların bir araya getirilmesi ile gerçekleştirilmektedir. Bu durumda sonucun doğru olarak alınabilmesi için "VE" den önce "VEYA" ve "VEYA" dan önce "VE" problemleri oluşmaktadır.

### "VEYA" kapısından önce "VE" kapısı

#### İşlev Şeması



#### Deyim listesi

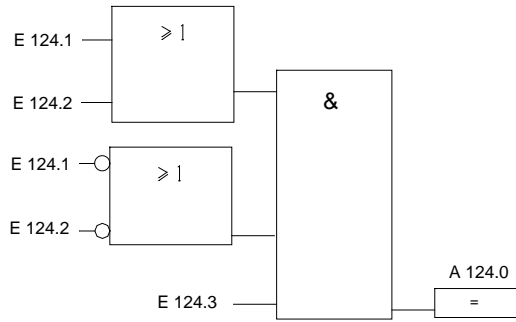
```

U E 124.1
U E 124.2
UN E 124.3
0
U E 124.1
UN E 124.2
0 E 124.3
= A 124.0

```

### "VE" kapısından önce "VEYA" kapısı

#### İşlev Şeması



#### Deyim listesi

```

U(
O E 124.1
O E 124.2
)
U(
ON E 124.1
ON E 124.2
)
U E 124.3
= A 124.0

```

Yukarıdaki çözümler incelendiğinde görülecektir ki, bazı mantık kapıları kombinasyonlarının "PLC " tarafından işlenmesinde yanlış sonuçlar doğmaktadır. Bu da temel prensip olarak "VE" den önce "VEYA" olması durumundadır.

Böyle bir durumda PLC'in programı sürekli bir çevrim olarak işlediğini hatırlamalıyız. Emirler sırayla işlenmekte ve elde edilen sonuçlar her seferinde bir yere yazılmaktadır. Bu yazılan alanlar emir listesinde parantez işlemlerine dönüşmektedir.

## PROGRAMIN PLC DE ÇALIŞMA ŞEKLİ

### GİRİŞ SİNYALLERİNİN SORGULANMASI

PLC programını çalıştırmaya başladığında kendisine donanım olarak bağlı bulunan bütün giriş sinyallerini okur ve onları dahili bir hafıza alanına kaydeder ( "0" veya "1" ). Bu işleme giriş resminin okunması denir. ( PEA )

Bütün emirleri bir kez işleyip yeniden sorgulama yapıncaya kadar geçen sürede giriş sinyallerinde meydana gelen değişiklikler dikkate alınmaz.

### PROGRAMIN ÇALIŞTIRILMASI

Alınan giriş sinyalleri belirlenen mantığa göre satır satır işlenerek, çıkışa atanması gereken değerlere ulaşılır.

### ÇIKIŞ SİNYALLERİNİN ATANMASI

Emirlerin tamamı işlenip program içerisinde çıkış yapılması gereken atamalar belirlendikten sonra bu bölümde atamalar yapılır.

Yeni bir atama yapıncaya kadar daha önce yapılan atamalar durumlarını korurlar. Bu işleme çıkış resminin yazılması denir. ( PAA )

## SEMBOİK ADRESLEME

Adreslerin sembollerle gösterildiği adresleme şeklidir. Program; sistemin belirlediği rakamsal değerler yerine, bizim belirlediğimiz isimlerle takip edilebilir.

Sembolik adreslerle yapılan bir program farklı PLC' lere kolay adapte edilir.

S7 programında iki tür sembolik çalışma mümkündür.

**1-Global Sembol:** Sembol tablosunda tanımlanan, S7 Programının bütün modülleri için geçerli olan bir semboldür. Sembol uzunluğu max. 24 işaret olabilir. Türkçe karakterler ve rakamla başlayan semboller kullanılamaz. Bir programda maksimum 1600 sembol kullanılabilir.

Sembol tablosunu başka bir editöre göndermek veya oradan almak mümkündür.

	B	M	K	SEMBOİK	ADRES	DATA TİPİ	AÇIKLAMA
0				Start	E 124.0	BOOL	Start butonu
1				Stop	E 124.1	BOOL	Stop butonu
2				Deger	MW 12	WORD	Karşılaştırma değeri

U #Start

UN #Stop

S #Motor

L #Deger

= #Çıkış

Diyez ( # ) işareti Yapılan adreslemenin sembolik olduğunu gösterir.

- B;M;K sütunları; ilişkilendirme sütunlarıdır ( B : WIN CC, M : Alarm, K : Haberleşme )
- Data tipi sütunu adres sütununda verilen değere göre otomatik olarak doldurulur.
- Açıklama sütunu verilen sembole ait açıklamaları içerir. Maksimum 80 karakter olabilir.

**2-Lokal Sembol:** Program modüllerinin deklarasyon tablosunda tanımlanan ve sadece o modül için geçerli olan bir tanımlamadır. (".....") ile gösterilir Ayrıntılı bilgi ilerde verilecektir.

**Not:** Program başka bir yere taşınmak veya kopyalanmak istendiğinde sembol tablosunun ayrıca taşınıp aynı projeye yerleştirilmesi gerekir.

**Örnek:1 LAMBA KUMANDASI**

Bir salona ait aydınlatma lambası iki ayrı anahtar ( giriş ve çıkış kapısında birer anahtar) ile yapılmak istenmektedir. Anahtarlardan herhangi birine basılması ile lamba yanacak, diğer anahtara basıldığında sönecektir. (vaviyen anahtar bağlantısı)

**ATAMA LİSTESİ**

Operand	Sembol	Açıklama
E124.0	S1	1. Anahtar
E124.1	S2	2. Anahtar
A124.0	K	Lamba

Atama listesi; programda kullanılacak giriş çıkış sinyallerinin hangi amaçla ve hangi sembolle kullanılacağını belirlediği yerdir.

Problemin çözümü için gerçeklik tablosu oluşturup, bu tablo üzerinden programımızı yazabiliriz

E2	E1	A	
0	0	0	
0	1	1	$E1.\overline{E2}$
1	0	1	$\overline{E1}.E2$
1	1	0	

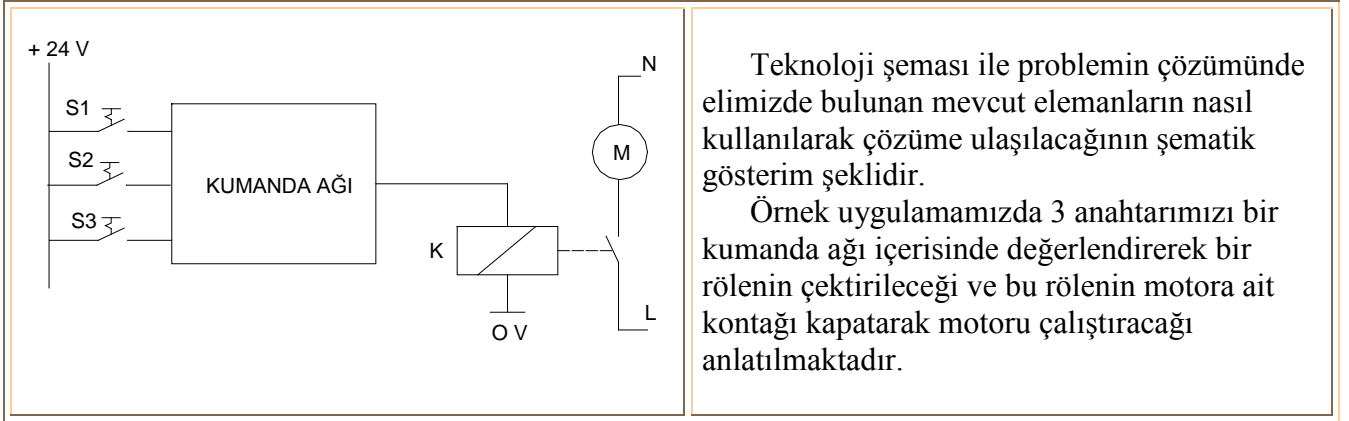
Satır sayısı  $2^2 = 4$   
( 2 giriş değişkenleri Sayısı )

Programın yazılması  
 $A = E1.\overline{E2} + \overline{E1}.E2$

PLC programımız		Sembolik adresleme ile programın yazılımı	
U	E 124.0	U	S1
UN	E 124.1	UN	S 2
O		O	
UN	E 124.0	UN	S1
U	E 124.1	U	S2
=	A 124.0	=	K

**Örnek:2 MOTOR KUMANDASI**

Bir motorun kumandası röle üzerinden üç ayrı anahtar yapılmak istenmektedir. Fonksiyon olarak anahtarlardan herhangi birine basılması veya bırakılması durumunda motorun çalışma durumunu değiştirmesi istenmektedir. Örneğin, anahtarlardan birine basıldığında motor çalışacak diğer birine basıldığında duracaktır. İlk iki anahtarın konumunda bir değişiklik olmaksızın üçüncüsüne basıldığında motor tekrar duracaktır.(Ara vaviyen)

**Teknoloji Şeması****Atama listesi**

Operand	Sembol	Açıklama
E124.1	S1	1. Anahtar
E124.2	S2	2. Anahtar
E124.3	S3	3. Anahtar
A124.0	K	Motoru kumanda eden kontaktör

Atama listesi; programda kullanılacak giriş çıkış sinyallerinin hangi amaçla ve hangi sembolle kullanılacağı belirlendiği yerdir.

Problemin çözümü için gerçeklik tablosu oluşturup, bu tablo üzerinden programımızı yazabiliriz

E3	E2	E1	A	
0	0	0	0	
0	0	1	1	$E1.\overline{E2}.\overline{E3}$
0	1	0	1	$\overline{E1}.E2.\overline{E3}$
0	1	1	0	
1	0	0	1	$\overline{E1}.\overline{E2}.E3$
1	0	1	0	
1	1	0	0	
1	1	1	1	$E1.E2.E3$

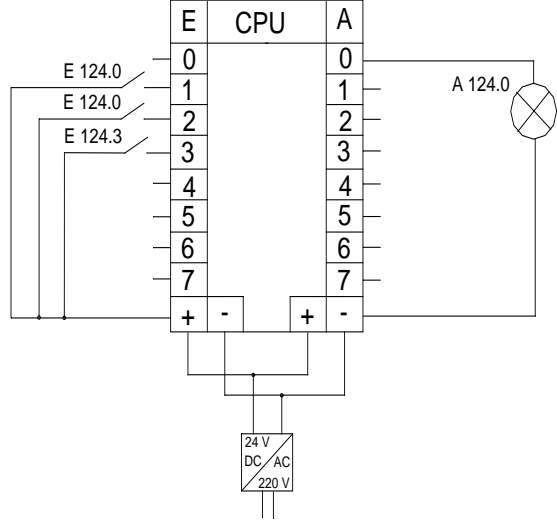
Satır sayısı  $2^3 = 8$   
( 3 giriş değişkenleri Sayısı )

Programın yazılması

$$A = E1.\overline{E2}.\overline{E3} + \overline{E1}.E2.\overline{E3} + \overline{E1}.\overline{E2}.E3 + E1.E2.E3$$

özüm için çıkışın "1" olduğu durumlar birbiri ile "VEYA" olarak bağlanırlar. Bu durumda "VE" kapısı "VEYA" kapısından öncedir.

PLC programı		Sembolik Adresleme	PLC bağlantı şeması												
U E124.1	UN E124.2	UN E124.3	O	UN E124.1	U E124.2	UN E124.3	O	UN E124.1	UN E124.2	U E124.3	O	U E124.1	U E124.2	U E124.3	= A124.0
U S1	UN S2	UN S3	O	UN S1	U S2	UN S3	O	UN S1	UN S2	U S3	O	U S1	U S2	U S3	= K



### Örnek:3 HAVALANDIRMA CİHAZLARININ İZLENMESİ

Bir kapalı otoparkın havalandırması için 4 adet havalandırma cihazı çalıştırılmaktadır. Havalandırma cihazları hava kirliliğine göre otomatik olarak devreye girmektedir. Bu cihazların izlenmesi PLC ile yapılmaktadır. Havalandırma cihazlarının hepsi veya 3 tanesi çalışıyor ise, havalandırma yeterli olmakta ve bu durum yeşil bir lamba ile gösterilmektedir. İki cihazın çalışması durumunda sarı lamba, bir veya lambaların hiçbirinin yanmaması durumunda kırmızı lamba yanacaktır.

TEKNOLOJİ ŞEMASI		ATAMA LİSTESİ		
		Operand	Sembol	Açıklama
		E 124.0	E 1	1.cihaz için sinyal verici
		E 124.1	E 2	2.cihaz için sinyal verici
		E 124.2	E 3	3.cihaz için sinyal verici
		E 124.3	E 4	4.cihaz için sinyal verici
		A 124.0	A 1	Kırmızı lamba
		A 124.1	A 2	Sarı lamba
		A 124.2	A 3	Yeşil lamba

GERÇEKLİK TABLOSU						
E 4	E 3	E 2	E 1	A 1	A 2	A 3
0	0	0	0	1	0	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	0	1

**Örnek: 4** Bir kimyasal reaksiyon kabı belli ısı ve basınç altında çalışmaktadır. Bu reaksiyon kabında ısı ve basınç ölçümü için ısı ve basınç ölçer vardır. Isı ve basıncın ayarlanması; bir ısıtıcı (H) soğuk su beslemesi (K) ve bir emniyet valfi (S) ile yapılmaktadır. Reaksiyon kabı kumanda elemanlarının çalışma şartları şöyledir.

Emniyet valfi (S)	:Basınç çok yüksekse
Soğuk su girişi (K)	:Isı çok yüksekse
Isıtıcı (H)	:Isı çok düşükse veya basınç çok düşük ve ısı normale
Karıştırıcı (U)	:Soğuk su girişi veya ısıtıcı çalışıyorsa
Reaksiyon işlemi üç şekilde çalışmaktadır.	
Başlangıç	:Basınç çok düşük(A <sub>N</sub> )
Normal	:Basınç normal (N <sub>B</sub> )
Alarm	:Basınç çok büyük (A <sub>L</sub> )

ATAMA TABLOSU		
OPERAND	SEMBOL	AÇIKLAMA
E124.0	P <sub>G</sub>	Basınç büyük
E124.1	P <sub>K</sub>	Basınç küçük
E124.2	T <sub>G</sub>	Isı yüksek
E124.3	T <sub>K</sub>	Isı küçük
A124.0	U	Karıştırıcı
A124.1	S	Emniyet valfi
A124.2	K	Soğuk su besleme
A124.3	H	Isıtıcı
A124.4	A <sub>N</sub>	Başlangıç
A124.5	N <sub>B</sub>	Normal işletim
A124.6	A <sub>L</sub>	Alarm

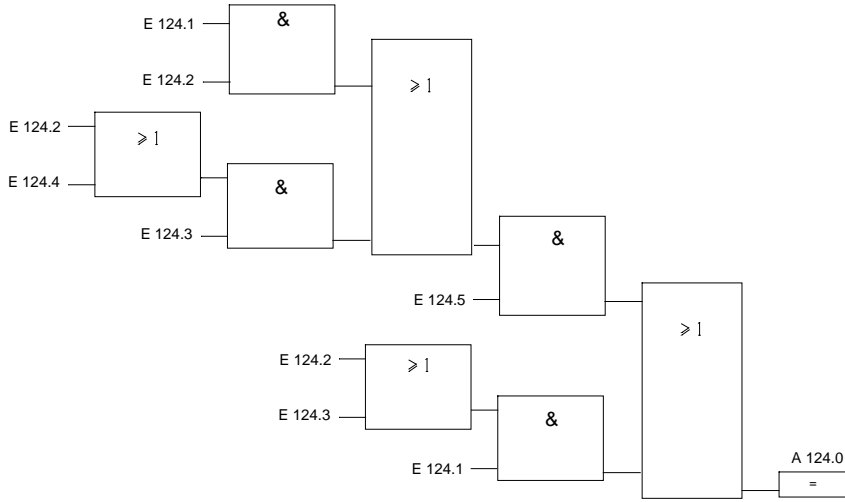
PLC Programı		Teknoloji şeması
Güvenlik valfi	O A124.3	
U E124.0	= A124.0	
Başlangıç	U E124.1	
Soğuk Su Girişi	= A124.1	
U E124.2	= A124.2	
Isıtıcı	UN E124.0	
U E124.3	UN E124.1	
O	= A124.5	
U E124.1	Alarm	
UN E124.2	U E124.0	
UN E124.3	= A124.6	
Karıştırıcı	BE	
O A124.2		

## DURUM TESPİT İŞARETLERİ ( MERKER / FLAG / YARDIMCI RÖLE )

Çeşitli kumanda programlarının ve özellikle de uzun fonksiyonların yazılımı sırasında karmaşa oluşmaktadır. Parantez seviyeleri çoğalmakta ve programı anlamak zorlaşmaktadır. Durum tespit işaretleri elektronik hafıza elemanları (R-S flip-flop) olup, bir sinyalin durumunu "0" veya "1" olarak saklanmasını sağlarlar.

### 2 Seviyeli Parantezli İşlev Şeması

#### PLC Programı



#### Deyim listesi

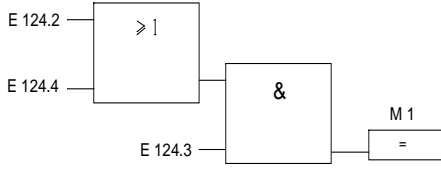
```

U(
U E124. 1
U E124. 2
O
U(
O E124. 2
O E124. 4
)
U E124. 3
)
U E124. 5
O
U(
O E124. 2
O E124. 3
)
U E124. 1
=A 124. 0

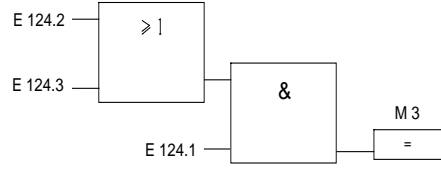
```

Yukarıdaki şekilde yazılmış programlar arıza aranması durumunda sinyal durumlarının tespitinde problem yaratırlar. Bunun yerine programdaki ara sonuçların durum tespit işaretlerine atanması büyük kolaylık sağlar. Durum tespit işaretleri ile çıkışlar arasında büyük benzerlik vardır. Aradaki en büyük fark, durum tespit işaretlerinin kuvvetlendiriciye sahip olmamaları veya dışarıya direkt olarak verilememeleridir. Ayrıca bir çıkış değişkenine atanmaları gerekir. Şimdi yukarıdaki işlev şemasını parçalara ayırarak sonuçları durum tespit işaretlerine yükleyelim.

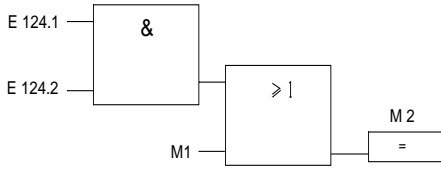
Netzwerk 1



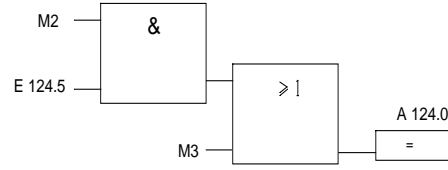
Netzwerk 3



Netzwerk 2



Netzwerk 4



Yeni elde ettiğimiz işlev şemasını deyim listesi olarak ifade edersek;

Atama listesi	Deyim Listesi			
M1=M 0.1	U (	U E 124.1	U (	U M 0.2
M2=M 0.2	O E 124.2	U E 124.2	O E 124.2	U E 124.5
M3=M 0.3	O E 124.4	O M 01	O E 124.3	O M 0.3
	)	= M 0.2	)	= A 124.0
	U E 124.3		U E 124.1	
	= M 0.1		= M 0.3	

PLC'in STOP durumuna getirilmesiyle veya elektrik kesilmesi sonrası durum tespit işaretlerindeki bilgilerin ne olacağı kullanılan cihaza bağlıdır. **Bir çok PLC' de durum tespit işaretleri remanent (kalıcı) veya remanent olmayan (kalıcı olmayan) diye ikiye ayrılmaktadırlar.** Remanent durum tespit işaretleri cihaz içerisindeki bir pil sayesinde sinyal durumunu korumaya devam eder. Diğerleri ise gerilimin gitmesi durumunda sakladıkları bilgileri kaybederler. Sayıcı ve zaman elemanları da kalıcı ve kalıcı olmayan diye ayrılırlar.

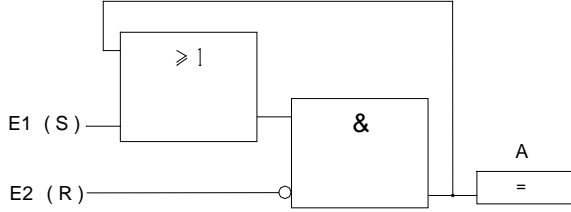
Örneğin; bir asansör yukarı çıkarken elektrik kesildi ve yeniden geldiğinde "0". katta imiş gibi davranmasını; veya yıldız üçgen ile kalkınan bir motorda yıldız kalkınmayı tamamlamış bir durumda elektrik kesilirse elektrik geldiğinde üçgenden kalkınmasını istemeyiz.

PLC' de remanent ayarlarının yapılması

**SIMATIC 300→CPU 314 IFM→sağ tuş→Objekteigenschaften→Remanenz**

**RS-HAFIZA ELEMANI**

E1 = Set girişi,  
E2 = Reset girişidir.

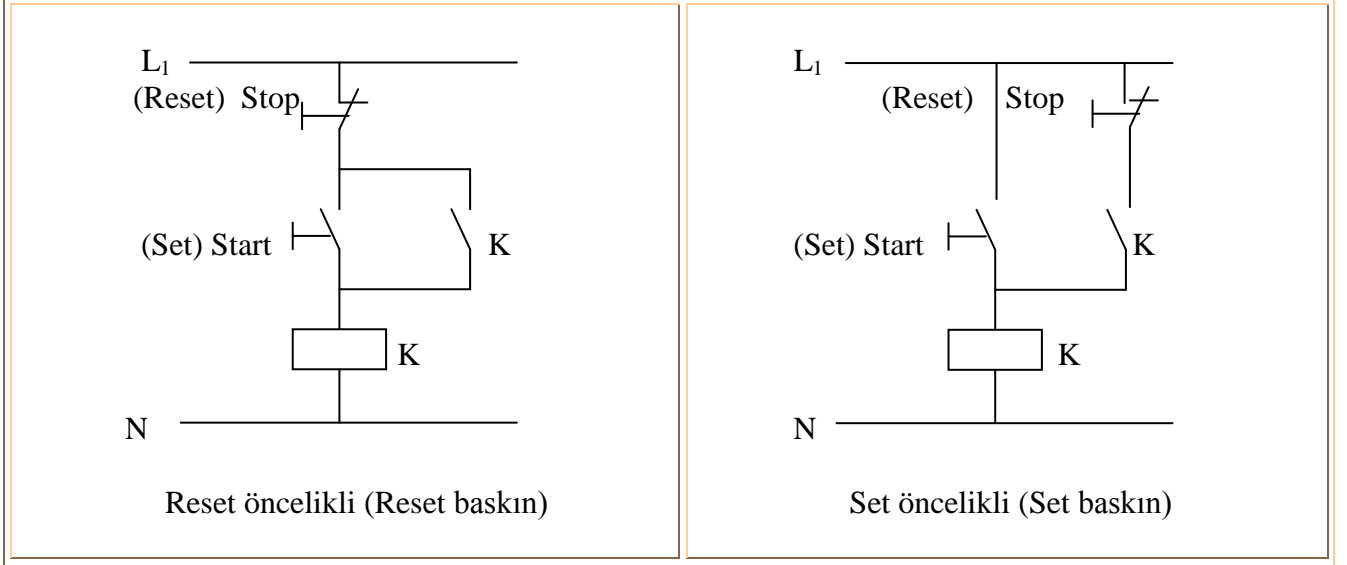


RS-Hafıza elemanları kumanda problemlerinde çok sık olarak kullanılmaktadır. Bu nedenle bütün PLC üreten firmalar belli bir sayıda RS-elemanı intern olarak hazırlamışlardır. Zaten PLC' nin kolaylığı da burada başlamaktadır. Bu elemanlar daha önce ara sonuçları aktardığımız elemanlar olarak öğrendiğimiz durum tespit işaretleri ile hardware olarak tamamen aynıdır. Sadece software olarak kullanılışlarında farklılık vardır. Başka bir şekilde durum tespit işaretlerinin yeni bir özelliğini tanıdığımızı söyleyebiliriz. İşlev şeması ve deyim listesi olarak bu elemanların programlanması aşağıdaki gibidir.

	Reset öncelikli	U E 1 S M1 U E 2 R M1
RS hafıza elemanın İşlev şeması olarak programlanması		RS hafıza elemanın Deyim listesi olarak programlanması
	Set öncelikli	U E 1 R M1 U E 2 S M1
RS hafıza elemanın İşlev şeması olarak programlanması		RS hafıza elemanın Deyim listesi olarak programlanması

Görüldüğü gibi "eşittir (=)" yerine "S" ve "R" emirleri gelmiştir. Tabi yukarıdaki işlev şemasına göre deyim listesi tam doğru değildir. Çünkü durum tespit işaretlerinin çıkış kuvvetlendiricisi yoktur. Bunların çıkış olarak dışarıya verilebilmesi için bir çıkış değişkenine atanmaları gerekir. Fakat yukarıdaki açıklamalar mantık olarak anlamaya yöneliktir.

Hafıza elemanları elektrik kumanda devrelerinde kullanılan kilitleme (mühürleme) devreleridir. Aşağıda iki değişik tipte gösterilmiştir.  
(Her iki devrede set ve reset butonlarına aynı anda basılırsa ne olur?)



### Örnek:5 ASENKRON MOTORUN ÇALIŞTIRILMASI

3 fazlı bir asenkron motor "S1" butonu ile çalıştırılacak "S0" butonu ile de durdurulacaktır. Ayrıca motor "F2" aşırı akım rölesi ile korunacaktır. Gerekli PLC programını yazınız.

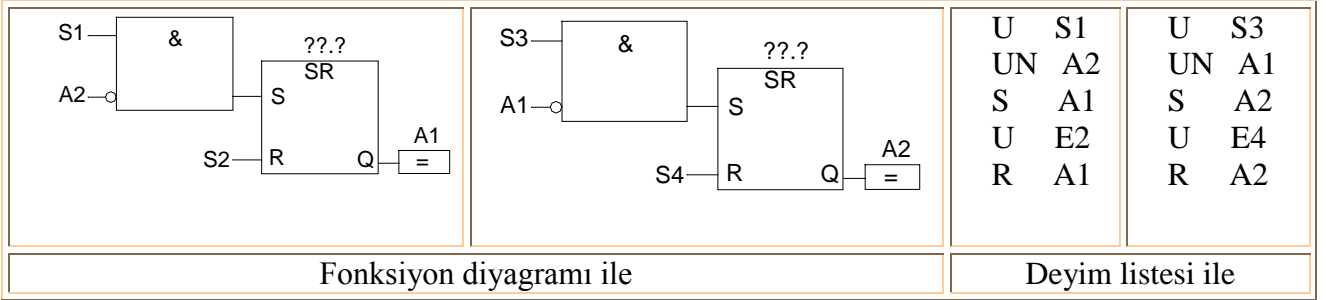
Atama Listesi		
Operand	Sembol	Açıklama
E 124.0	S0	Stop anahtarı (NK) (Stop butonu kablo kopmalarına karşı önlem almak için NK seçilmelidir.
E 124.1	S1	Start Butonu (NA)
E 124.7	F2	Aşırı akım rölesi(NK) (Aşırı akım rölesi kontağı kablo kopmalarına karşı önlem almak için NK seçilmelidir
A 124.0	K	Kontaktör
NA :Normalde açık      NK :Normalde kapalı		

U	E 124 . 1
S	M 0 . 0
U (	
ON	E 124 . 0
ON	E 124 . 7
)	
R	M 0 . 0
U	M 0 . 0
=	A 124 . 0
BE	

## BELLEK ELEMANLARININ KARŞILIKLI OLARAK KİLİTLENMESİ

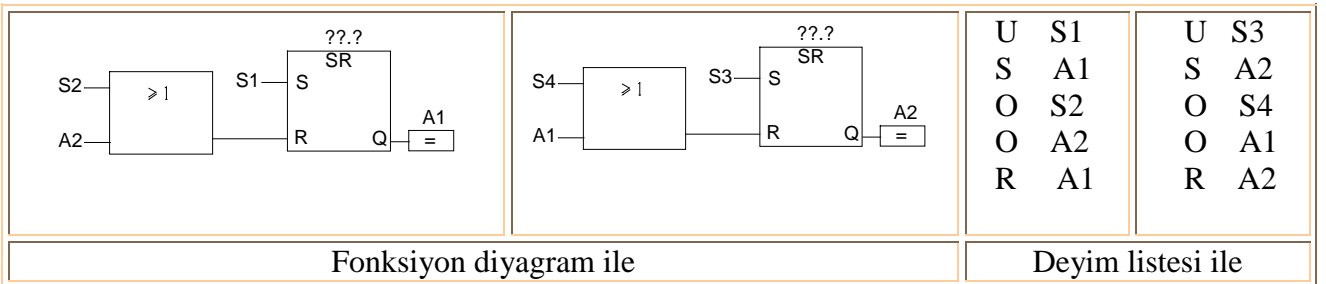
Bellek elemanlarının karşılıklı olarak kilitlemesi kumanda problemlerinde her zaman karşılaşılan, göz önünde tutulması gereken bir prensiptir. Bir bellek elemanın kilitlemesi, bu elemanın ancak belirli şartlar altında SET yapılabilmesi demektir. Kilitleme SET kısmında olabileceği gibi RESET kısmında da olabilir.

### SET Kısımında Kilitleme



VE kapıları sayesinde SET işleminin yapılabilmesi için mutlaka diğer çıkış değişkeninin "0" değerine sahip olması gerekir.

### RESET Kısımında Kilitleme



Bu sistemde ise "S1" veya "S3" sinyalinin gelmesi ilgili bellek elemanın SET yapılmasını sağlıyor. Fakat PLC seri olarak çalıştığı için, diğer çıkış değişkenin "1" değerine sahip olması durumunda SET yapılan bellek elemanı yeniden RESET yapılacaktır.

RESET kısmında yapılacak bir kilitleme sadece RESET ağırlıklı sinyal elemanlarında mümkün olmakla birlikte çok sık kullanılmaktadır.

Başka bir kilitleme şekil de bellek fonksiyonlarının kesin olarak belirlenmiş bir sırayla SET ve RESET yapılmalıdır. Bu şekilde bir bellek elemanın SET yapılabilmesi için öncekinin SET yapılmış olması gerekmektedir. Ayrıca her SET işleminden sonra bir önceki bellek elemanı RESET yapılmaktadır.

### Sıra Halinde SET Girişinde Kilitleme:

		<pre> U S1 S A1 U S2 R A1 </pre>	<pre> U S3 U A1 S A2 U S4 R A2 </pre>
Fonksiyon diyagram ile		Deyim listesi ile	

### Sıra Halinde RESET Girişinde Kilitleme:

		<pre> U S1 S A1 U S2 R A1 </pre>	<pre> U S3 S A2 U S4 UN A1 R A2 </pre>
Fonksiyon diyagram ile		Deyim listesi ile	

**Örnek:6** ASENKRON MOTORUN İLERİ ve GERİ YÖNDE ÇALIŞTIRILMASI

3 fazlı bir asenkron motor "S1" butonu ile ileri yönde "S2" butonu ile geri yönde çalıştırılacaktır. "S0" butonuna basılınca da duracaktır. Bir yönde çalışırken motor durdurulmadan diğer yönde çalışmayacaktır. Ayrıca motor "F2" aşırı akım rölesi ile korunacaktır.

Atama Listesi		
Operand	Sembol	Açıklama
E 124.0	S0	Stop anahtarı (NK)
E 124.1	S1	İleri çalıştırma butonu(NA)
E 124.2	S2	Geri çalıştırma Butonu(NA)
E 124.7	F2	Aşırı akım rölesi(NK)
A 124.0	K1	İleri yön rölesi
A 124.1	K2	Geri yön rölesi

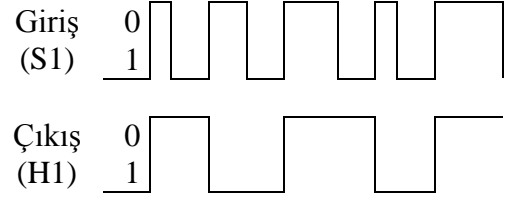
NA :Normalde açık  
NK :Normalde kapalı

PLC programı	Sembolik adresleme
U E 124 . 1 Motorun ileri yönde çalışması	U S1
UN A 124 . 1	UN K2
S M 0 . 0	S M 0 . 0
U (	U (
ON E 124 . 0	ON S0
ON E 124 . 7	ON F2
)	)
R M 0 . 0	R M 0 . 0
U E 124 . 2 Motorun geri yönde çalışması	U S2
UN A 124 . 0	UN K1
S M 0 . 1	S M 0 . 1
U (	U (
ON E 124 . 0	ON S0
ON E 124 . 7	ON F2
)	)
R M 0 . 1	R M 0 . 1
U M 0 . 0	U M 0 . 0
= A 124 . 0	= K1
U M 0 . 1	U M 0 . 1
= A 124 . 1	= K2

**Örnek:7** Bir butonun kısa süreli uyarılması ile bir lamba yanmalı, butonun yeniden uyarılması ile lamba sönmelidir.

ATAMA LİSTESİ		
Operand	Sembol	Açıklama
E 124.0	S1	Açma kapama butonu
A 124.0	H1	Lamba

Fonksiyon diyagramı



```
UN A 124 . 0
U E 124 . 0
S A 124 . 0
```

```
U A 124 . 0
U E 124 . 0
R A 124 . 0
```

Program bu şekli yazıldığında çalışmayacaktır.

```
UN A 124 . 0
U E 124 . 0
S M 0 . 0
```

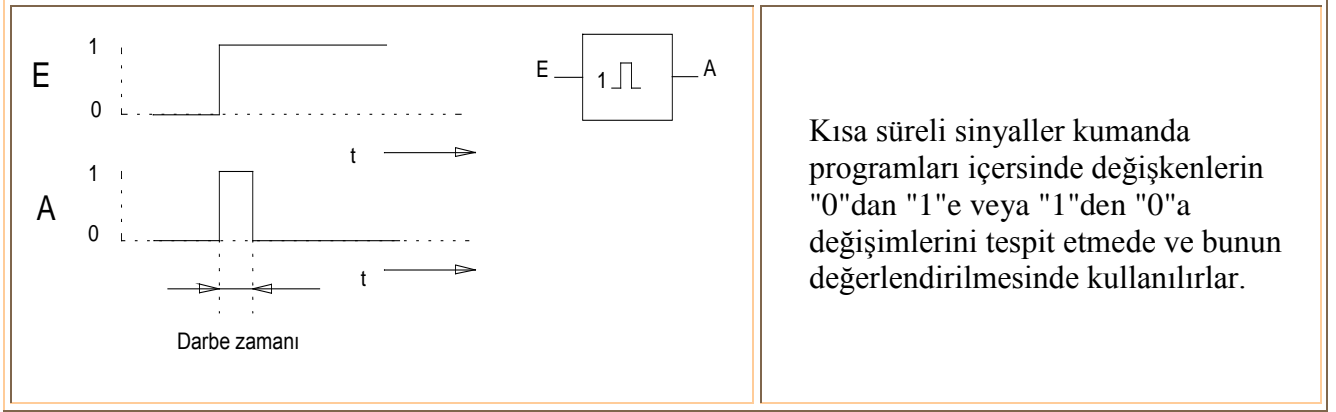
```
U A 124 . 0
U E 124 . 0
R M 0 . 0
```

```
U M 0 . 0
= A 124 . 0
```

Program bu şekli yazıldığında yine bizim istediklerimizi yapmayacaktır. Çünkü butona elimizi basıp çekinceye kadar PLC programımızı çok kez çalıştıracaktır. Dolayısı ile butondan elimizi çekme zamanına bağlı olarak lamba bazen yanık, bazen sönmüş kalacaktır. (Nasıl önlem alınır?)

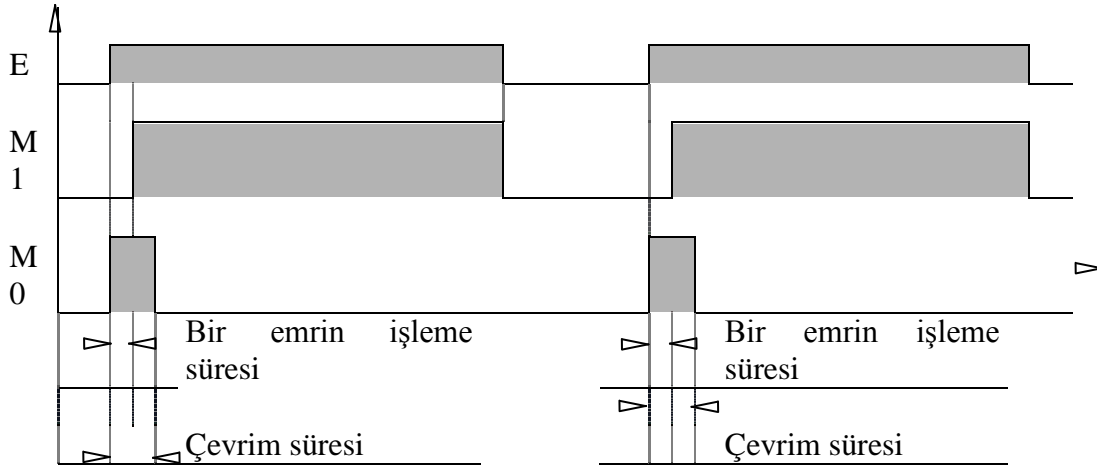
## DARBE VERİCİLER

Darbe vericiler kumanda tekniğinde uzun süreli giriş sinyallerinden bir darbe oluşturulmasında kullanılırlar. Kontaktör tekniğinde de aynı düşünce mevcuttur. Kontaktörün çekmesi veya düşmesi sırasında çok kısa bir sinyal elde edilebilir. Bu fonksiyon bazı PLC üreten firmalarda intern olarak gerçekleştirilir. Hatta darbe süresi bile ayarlanabilir.

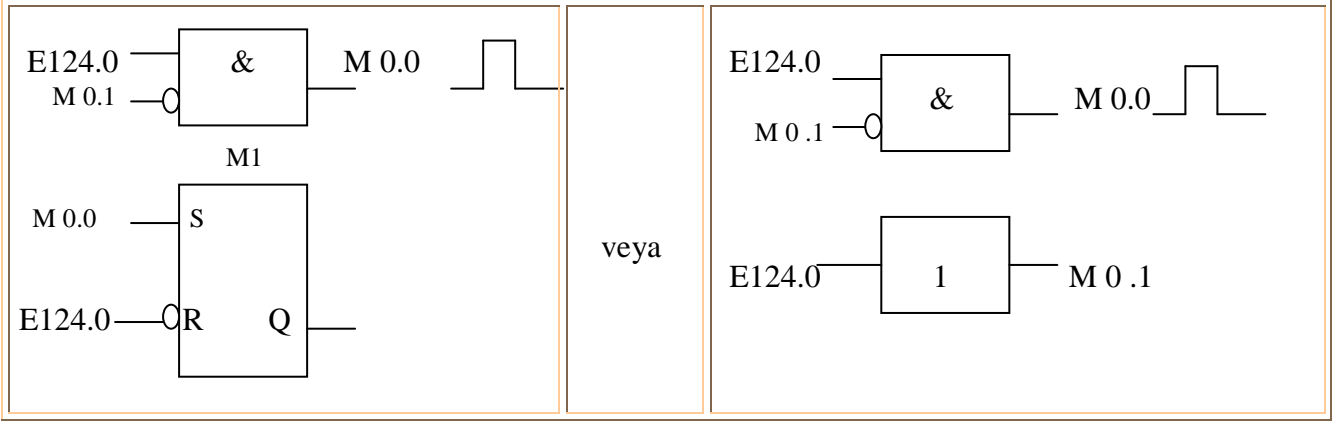


## POZİTİF BİR DEĞİŞİMİN TESPİT EDİLMESİ

Aşağıdaki tabloda da kısa süreli sinyal vericilerin daha iyi anlaşılması için çevrim sayısına göre değişkenlerin durumları belirlenmiştir.

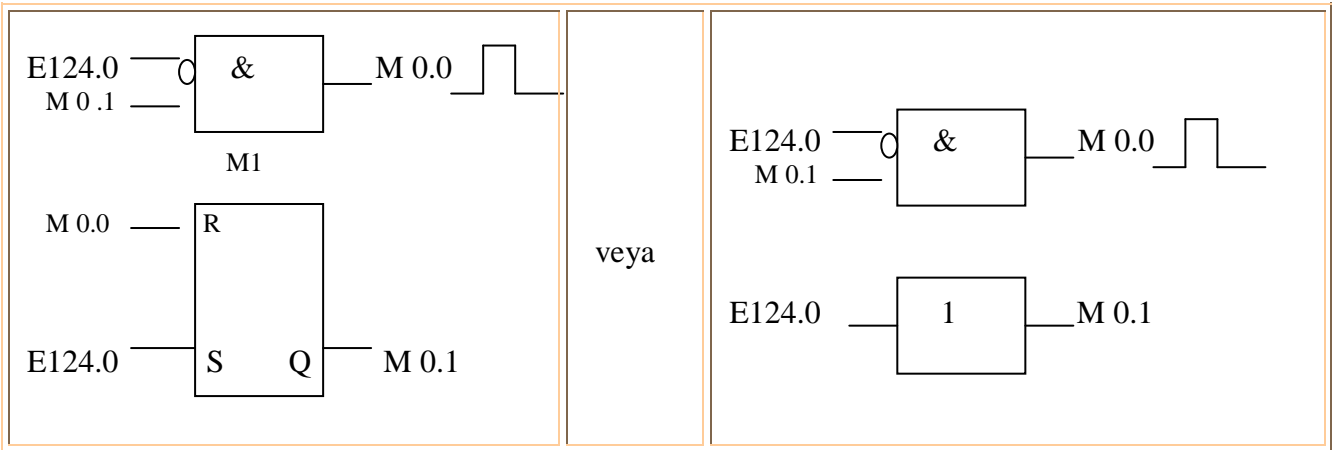


### Pozitif Değişim



U E124.0 UN M0.1 = M0.0 S M0.1 UN E124.0 R M0.1	veya	U E124.0 UN M0.1 = M0.0 U E124.0 = M0.1	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><th>Çevrim</th><td>1 2 3 4 . . . . n-1 n</td></tr> <tr><th>Değişken</th><td></td></tr> <tr><th>E</th><td>0 1 1 1 . . . . 1 0</td></tr> <tr><th>M 0.0</th><td>0 1 0 0 . . . . 0 0</td></tr> <tr><th>M 0.1</th><td>0 1 1 1 . . . . 1 0</td></tr> </table>	Çevrim	1 2 3 4 . . . . n-1 n	Değişken		E	0 1 1 1 . . . . 1 0	M 0.0	0 1 0 0 . . . . 0 0	M 0.1	0 1 1 1 . . . . 1 0	Siemens'te hazır darbe U E124.0 FP M0.1 = M0.0
Çevrim	1 2 3 4 . . . . n-1 n													
Değişken														
E	0 1 1 1 . . . . 1 0													
M 0.0	0 1 0 0 . . . . 0 0													
M 0.1	0 1 1 1 . . . . 1 0													

### Negatif değişim

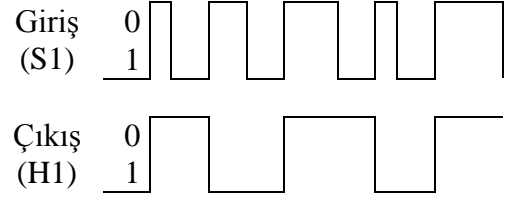


UN E124.0 U M0.1 = M0.0 R M0.1 U E124.0 S M0.1	veya	UN E124.0 U M0.1 = M0.0 U E124.0 = M0.1	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><th>Çevrim</th><td>1 2 3 4 . . . . n-2 n-1 n</td></tr> <tr><th>Değişken</th><td></td></tr> <tr><th>E</th><td>0 1 1 1 . . . . 1 0 0</td></tr> <tr><th>M 0.0</th><td>0 0 0 0 . . . . 0 1 0</td></tr> <tr><th>M 0.1</th><td>0 1 1 1 . . . . 1 0 0</td></tr> </table>	Çevrim	1 2 3 4 . . . . n-2 n-1 n	Değişken		E	0 1 1 1 . . . . 1 0 0	M 0.0	0 0 0 0 . . . . 0 1 0	M 0.1	0 1 1 1 . . . . 1 0 0	Siemens'te hazır darbe U E124.0 FN M0.1 = M0.0
Çevrim	1 2 3 4 . . . . n-2 n-1 n													
Değişken														
E	0 1 1 1 . . . . 1 0 0													
M 0.0	0 0 0 0 . . . . 0 1 0													
M 0.1	0 1 1 1 . . . . 1 0 0													

**Örnek:8** Bir butonun kısa süreli uyarılması ile bir lamba yanmalı, butonun yeniden uyarılması ile lamba sönmelidir.

ATAMA LİSTESİ		
Operand	Sembol	Açıklama
E 124.0	S1	Açma kapama butonu
A 124.0	H1	Lamba

Fonksiyon diyagramı



```

U   E   124.0
FP  M   0.3
=   M   0.2   Bir çevrim süresi kadar darbe

```

```

UN  A   124.0
U   M   0.2
S   M   0.0
U   A   124.0
U   M   0.2
R   M   0.0

```

```

U   M   0.0
=   A   124.0

```

**Örnek:9 BİR BUTON DÖRT LAMBA KUMANDASI**

Bir buton sıra ile 4 lamba yakacaktır. Butona birinci kez basıldığında 1. lamba, ikinci kez basıldığında 2. lamba, üçüncü kez basıldığında 3. lamba, dördüncü kez basıldığında 4. lamba yanacaktır.5. kez basıldığında bütün lambalar sönecektir.

Atama Listesi		
Operand	Sembol	Açıklama
E 124.0	S0	Start (NA)
A 124.0	H1	1.Lamba
A 124.1	H2	2.Lamba
A 124.2	H3	3.Lamba
A 124.3	H4	4.Lamba

<p><b>DARBE</b></p> <p>U E 124.0 FP M0.1 = M0.0</p> <p><b>RESET</b></p> <p>U M 0.0 U M 10.3 = M 10.5</p> <p><b>4.LAMBA</b></p> <p>U M 0.0 U M 10.2 S M 10.3 U M 10.5 R M10.3</p> <p><b>3.LAMBA</b></p> <p>U M 0.0 U M 10.1 S M 10.2 U M 10.5 R M10.2</p>	<p><b>2.LAMBA</b></p> <p>U M 0.0 U M 10.0 S M 10.1 U M 10.5 R M10.1</p> <p><b>1.LAMBA</b></p> <p>U M 0.0 S M 10.0 U M 10.5 R M10.0</p> <p><b>ÇIKIŞLAR</b></p> <p>U M 10.0 = A124.0 U M 10.1 = A124.1 U M 10.2 = A124.2 U M 10.3 = A124.3 BE</p>
--	---

**Örnek:10 BİR BUTON DÖRT LAMBA KUMANDASI**

Bir buton sıra ile 4 lamba yakacaktır. Butona birinci kez basılıp çekildiğinde 1. lamba, ikinci kez basılıp çekildiğinde 2. lamba, üçüncü kez basılıp çekildiğinde 3. lamba, dördüncü kez basılıp çekildiğinde 4. lamba yanacaktır.

5. kez basılıp çekildiğinde 4.lamba 6. kez basılıp çekildiğinde 3. lamba 7. kez basılıp çekildiğinde 2. lamba 8. kez basılıp çekildiğinde 1.lamba sönecektir.9. kez basılıp çekildiğinde ise 1.lamba yanacak yani işlem başa dönecektir.

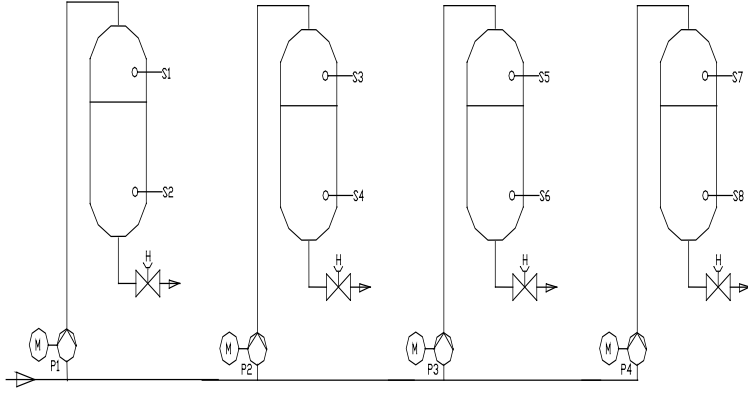
Atama Listesi				
Operand	Sembol	Açıklama		
E 124.0	S0	Start (NA)		
A 124.0	H1	1.Lamba		
A 124.1	H2	2.Lamba		
A 124.2	H3	3.Lamba		
A 124.3	H4	4.Lamba		
<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> <p><b>DARBE</b></p> <p>U E 124 . 0</p> <p>FP M 0 . 1</p> <p>= M 0 . 0</p> <p><b>1.LAMBA</b></p> <p>U M 0 . 0</p> <p>S M 10 . 0</p> <p>U M 0 . 0</p> <p>UN M 10 . 1</p> <p>U M 10 . 6</p> <p>R M 10 . 0</p> <p><b>2.LAMBA</b></p> <p>U M 0 . 0</p> <p>U A 124 . 0</p> <p>S M 10 . 1</p> <p>U M 0 . 0</p> <p>UN M 10 . 2</p> <p>U M 10 . 6</p> <p>R M 10 . 1</p> <p><b>3.LAMBA</b></p> <p>U M 0 . 0</p> <p>U A 124 . 1</p> <p>S M 10 . 2</p> <p>U M 0 . 0</p> <p>UN M 10 . 3</p> <p>U M 10 . 6</p> <p>R M 10 . 2</p> </td> <td style="width: 50%; vertical-align: top;"> <p><b>4.LAMBA</b></p> <p>U M 0 . 0</p> <p>U A 124 . 2</p> <p>S M 10 . 3</p> <p>U M 0 . 0</p> <p>U M 10 . 6</p> <p>R M 10 . 3</p> <p><b>RESET</b></p> <p>U M 10 . 0</p> <p>U M 10 . 1</p> <p>U M 10 . 2</p> <p>U M 10 . 3</p> <p>S M 10 . 6</p> <p>UN M 10 . 0</p> <p>U M 0 . 0</p> <p>R M 10 . 6</p> <p><b>ÇIKIŞLAR</b></p> <p>U M 10 . 0</p> <p>= A 124 . 0</p> <p>U M 10 . 1</p> <p>= A 124 . 1</p> <p>U M 10 . 2</p> <p>= A 124 . 2</p> <p>U M 10 . 3</p> <p>= A 124 . 3</p> <p><b>BE</b></p> </td> </tr> </table>			<p><b>DARBE</b></p> <p>U E 124 . 0</p> <p>FP M 0 . 1</p> <p>= M 0 . 0</p> <p><b>1.LAMBA</b></p> <p>U M 0 . 0</p> <p>S M 10 . 0</p> <p>U M 0 . 0</p> <p>UN M 10 . 1</p> <p>U M 10 . 6</p> <p>R M 10 . 0</p> <p><b>2.LAMBA</b></p> <p>U M 0 . 0</p> <p>U A 124 . 0</p> <p>S M 10 . 1</p> <p>U M 0 . 0</p> <p>UN M 10 . 2</p> <p>U M 10 . 6</p> <p>R M 10 . 1</p> <p><b>3.LAMBA</b></p> <p>U M 0 . 0</p> <p>U A 124 . 1</p> <p>S M 10 . 2</p> <p>U M 0 . 0</p> <p>UN M 10 . 3</p> <p>U M 10 . 6</p> <p>R M 10 . 2</p>	<p><b>4.LAMBA</b></p> <p>U M 0 . 0</p> <p>U A 124 . 2</p> <p>S M 10 . 3</p> <p>U M 0 . 0</p> <p>U M 10 . 6</p> <p>R M 10 . 3</p> <p><b>RESET</b></p> <p>U M 10 . 0</p> <p>U M 10 . 1</p> <p>U M 10 . 2</p> <p>U M 10 . 3</p> <p>S M 10 . 6</p> <p>UN M 10 . 0</p> <p>U M 0 . 0</p> <p>R M 10 . 6</p> <p><b>ÇIKIŞLAR</b></p> <p>U M 10 . 0</p> <p>= A 124 . 0</p> <p>U M 10 . 1</p> <p>= A 124 . 1</p> <p>U M 10 . 2</p> <p>= A 124 . 2</p> <p>U M 10 . 3</p> <p>= A 124 . 3</p> <p><b>BE</b></p>
<p><b>DARBE</b></p> <p>U E 124 . 0</p> <p>FP M 0 . 1</p> <p>= M 0 . 0</p> <p><b>1.LAMBA</b></p> <p>U M 0 . 0</p> <p>S M 10 . 0</p> <p>U M 0 . 0</p> <p>UN M 10 . 1</p> <p>U M 10 . 6</p> <p>R M 10 . 0</p> <p><b>2.LAMBA</b></p> <p>U M 0 . 0</p> <p>U A 124 . 0</p> <p>S M 10 . 1</p> <p>U M 0 . 0</p> <p>UN M 10 . 2</p> <p>U M 10 . 6</p> <p>R M 10 . 1</p> <p><b>3.LAMBA</b></p> <p>U M 0 . 0</p> <p>U A 124 . 1</p> <p>S M 10 . 2</p> <p>U M 0 . 0</p> <p>UN M 10 . 3</p> <p>U M 10 . 6</p> <p>R M 10 . 2</p>	<p><b>4.LAMBA</b></p> <p>U M 0 . 0</p> <p>U A 124 . 2</p> <p>S M 10 . 3</p> <p>U M 0 . 0</p> <p>U M 10 . 6</p> <p>R M 10 . 3</p> <p><b>RESET</b></p> <p>U M 10 . 0</p> <p>U M 10 . 1</p> <p>U M 10 . 2</p> <p>U M 10 . 3</p> <p>S M 10 . 6</p> <p>UN M 10 . 0</p> <p>U M 0 . 0</p> <p>R M 10 . 6</p> <p><b>ÇIKIŞLAR</b></p> <p>U M 10 . 0</p> <p>= A 124 . 0</p> <p>U M 10 . 1</p> <p>= A 124 . 1</p> <p>U M 10 . 2</p> <p>= A 124 . 2</p> <p>U M 10 . 3</p> <p>= A 124 . 3</p> <p><b>BE</b></p>			

**Örnek:11** Bir motor iki buton ile kumanda edilecektir.Herhangi bir butona basılınca motor çalışacak,yine herhangi bir botuna basılınca duracaktır.

Butonlardan birinin kontakları birbirine yapışırsa diğer buton görevine devam edecektir.

Atama Listesi		
Operand	Sembol	Açıklama
E 124.0	S1	Açma kapama butonu (NA)
E 124.1	S1	Açma kapama butonu (NA)
A 124.0	M	Motor kontaktörü

Bir buton olsaydı	İki buton olunca	İki buton olunca
<pre> U E 124.0 FP M 0.1 = M 0.0  UN A 124.0 U M 0.0 S M 0.2 U A 124.0 U M 0.0 R M 0.2  U M 0.2 = A 124.0 BE </pre>	<pre> O E 124.0 O E 124.1 = M 10.0  U M 10.0 FP M 0.1 = M 0.0  UN A 124.0 U M 0.0 S M 0.2 U A 124.0 U M 0.0 R M 0.2  U M 0.2 = A 124.0 BE </pre>	<pre> U ( O E 124.0 O E 124.1 ) FP M 0.1 = M 0.0  UN A 124.0 U M 0.0 S M 0.2 U A 124.0 U M 0.0 R M 0.2  U M 0.2 = A 124.0 BE </pre>

**Örnek: 12 SU TANKI KUMANDASI****Teknoloji Şeması:**

Dört adet tank manuel olarak boşaltılabilmektedirler. Doldurma işlemi ortak bir depodan dört ayrı pompa sayesinde yapılmaktadır. Her tankta tankın boş veya dolu olduğunu bildiren sinyal vericiler mevcuttur. Her pompayı çalıştıran motorların güçleri farklıdır. P1 = 3 Kw, P2 = 2 kW, P3 = 7 kW, P4 = 5 kW

Herhangi bir tankın boş olduğunu bildiren sinyal geldiğinde doldurma işleminin başlaması istenmektedir. Fakat toplam güç olarak 10 kW aşılmamalıdır.

**Atama Listesi :**

Operand	Sembol	Açıklama					
E 124.0	S1	Dolu sinyali Tank 1	Problemin çözülebilmesi için hangi kombinasyonlarda 10 kW sınırının aşılabileceğinin bilinmesi ve bu durumlarda başka bir tankın çalışmasının engellenmesi veya kilitlenmesinin sağlanmasıdır.				
E 124.2	S3	Dolu sinyali Tank 2					
E 124.4	S5	Dolu sinyali Tank 3					
E 124.6	S7	Dolu sinyali Tank 4					
E 124.1	S2	Boş sinyali Tank 1					
E 124.3	S4	Boş sinyali Tank 2					
E 124.5	S6	Boş sinyali Tank 3					
E 124.7	S8	Boş sinyali Tank 4					
A 124.1	P1	Pompa Tank 1	Bunu bir tablo ile araştıralım.				
A 124.2	P2	Pompa Tank 2					
A 124.3	P3	Pompa Tank 3					
A 124.4	P4	Pompa Tank 4					
			Pompalar	P1	P2	P3	P4
			Kombinasyonlar	P2 & P3	P1 & P3	P1 & P2 , P4	P3

Her pompanın ayrı ayrı ne zaman çalışıp ne zaman duracağını şartları belirlendi, böylece örneğin "P1" in çalışma şartı deponun boş olduğunu bildiren "S2" sinyali gelmiş ise "P1" çalışacak ve depo dolu sinyali "S1" gelince pompa duracaktır.	U	S2	Ayrıca bu pompa P2 & P3 pompalarının birlikte çalışıyor olması durumunda da çalışmayacaktır. Bu şartı da programa ekleyecek olursak	U	S2
	S	P1		S	P1
	U	S1		O	S1
	R	P1		U	P2
				U	P3
				R	P1

Programın tamamını bu mantıkla gerçekleştirecek olursak

POMPA 1	POMPA 2	POMPA 3	POMPA 4	Yandaki program bu şekli ile yazıldığında çalışmayacaktır. Pompa sürekli olarak set ve reset yapılacaktır. Set şartı olduğunda pompa çalışıyor gibi görünmesine rağmen sürekli olarak durumu değişmektedir.
U S2	U S4	U S6	U S8	
S P1	S P2	S P3	S P4	
U S1	U S3	O S5	U S7	
O	O	O P4	O	
U P2	U P1	O	U P3	
U P3	U P3	U P1	R P4	
R P1	R P2	U P2		
		R P3		

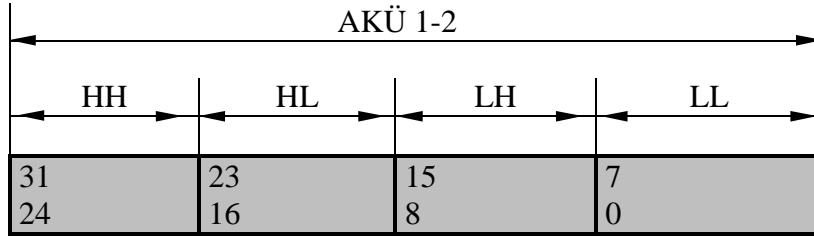
Bu sorunun çözümü için merkerler kullanılmalı ve çıkışlar programın sonunda atanmalıdır.

POMPA 1	POMPA 2	POMPA 3	POMPA 4	ÇIKIŞLAR
U E 124.1	U E 124.3	U E 124.5	U E 124.7	U M 1.0
S M 1.0	S M 2.0	S M 3.0	S M 4.0	= A 124.1
U E 124.0	U E 124.2	O E 124.4	O A 124.3	U M 2.0
O	O	O A 124.4	O E 124.6	= A 124.2
U A 124.2	U A 124.1	O	R M 4.0	U M 3.0
U A 124.3	U A 124.3	U A 124.1		= A 124.3
R M 1.0	R M 2.0	U A 124.2		U M 4.0
		R M 3.0		= A 124.4

## YÜKLEME (L) VE TRANSFER (T) OPERASYONLARI

Yükleme ve transfer emirleri word işlemeyi kavramak için önemli operasyonlardır. Çünkü bu emirler kumandanın giriş ve çıkış değişkenleri, durum tespit işaretleri ve data bellek alanları gibi çeşitli modülleri arasında veri alışverişini sağlamaktadırlar. Bu veri değişimi direkt olmamaktadır. Bunun için akümülatörler kullanılmaktadır.

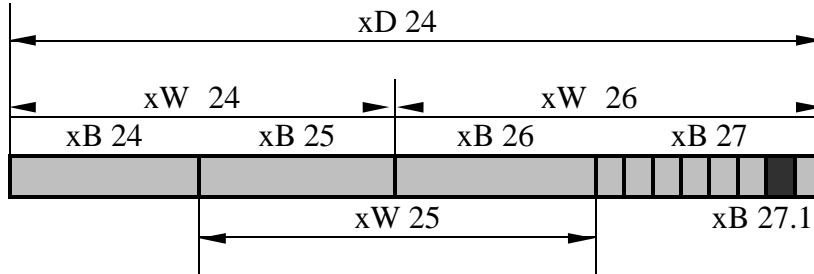
**Akü'lerin tanımlanması:** Akümülatörler PLC'lerde geçici bilgi saklamak amacıyla kullanılan özel registerlerdir. Akümülatör esas register "**Akü 1**" ve yardımcı register "**Akü 2**" olmak üzere iki bölümden oluşan ve mikroişlemci tarafından özel olarak tanınan 32 bit'lik bir hafıza alanıdır.



LL = Akü 1 ve 2' nin düşük word' unun düşük bayt'ı  
 LH = Akü 1 ve 2' nin düşük word' unun yüksek bayt'ı  
 HL = Akü 1 ve 2' nin yüksek word' unun düşük bayt'ı  
 HH = Akü 1 ve 2' nin yüksek word' unun yüksek bayt'ı

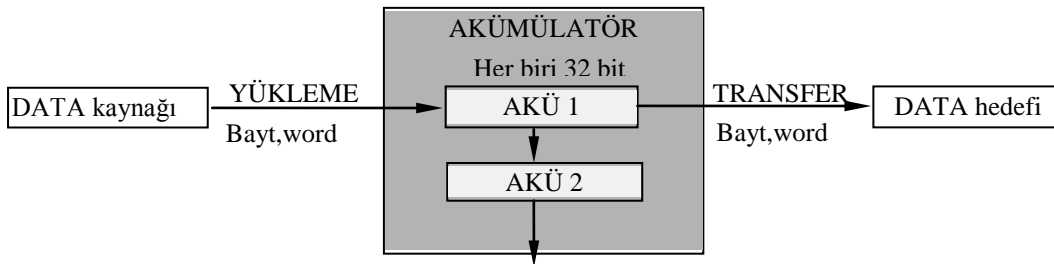
Aynı tanımlamalar 32 bitlik tüm veri alanları için geçerlidir.

### Word veya Doppelword' da baytların atanması



x = E ( Eingang = Giriş kanalı )  
 x = A ( Ausgang = Çıkış kanalı )  
 x = M ( Merker = Hafıza elemanı )

"SIEMENS PLC" lerde BAYT numaraları soldan sağa doğru, bit numaraları sağdan sola doğru artar !



Karşılaştırma, aritmetik ve lojik kombinasyon operasyonları için Akü 2 de gereklidir. ilk önce

yüklenen operand Akü 1'e gelir. ikinci operand için 1. Aküdeki operand 2. Aküye kaydırılır ve ikinci operand Akü 1'e yüklenir. Daha sonra istenilen operasyon gerçekleştirilir ve sonuç Akü 1'dedir. Her yeni yüklemde Akü 2 deki değer silinir.

**YÜKLEME:** Verilerin bir kaynaktan akümülatöre getirilmesidir. Bu kaynak giriş-çıkış değişkenleri, durum tespit işaretleri, data alanlarıdır. Yüklenecek veri bayt ,word veya doppelwort olabilir. Aynı şekilde sabit değerler, zaman elemanları, sayıcılar da akümülatöre yüklenebilir. Yüklenen veri Akü 1 içerisinde en sağdaki bitten itibaren yerleşir.

Sayıcı ve zaman elemanlarının yüklenmesinde iki yöntem vardır:

- L : Sayısal (binary) olarak
- LC : BCD kodlanmış olarak

Önemli yükleme emirlerinden bazıları

Yapılacak iş	Operand	AWL kodu
"- 5" sayısını 16 bitlik tam sayı olarak yükle	+ / -	L -5
"17" ve "129" sabit sayılarını iki bayt olarak yükle ( Akü 1 LL = 0001 0001, Akü 1 HL = 1000 0001 )	B#( )	L B#(17,129)
" +7 " sayısını 32 bit tam sayı olarak yükle	L#	L L#+7
8 bit heksadesimal sayı "E3" ü yükle	B#16#	L B#16#E3
16 bit heksadesimal sayı "3AC5" i yükle	W#16#	L W#16#3AC5
32 bit heksadesimal sayı "C0D2 A3FF" i yükle	DW#16#	L DW#16#C0D2_A3FF
"10011" dual sayısını yükle	2#	L 2#10011
32 bit kayan virgüllü sayı "1,25*10 <sup>2</sup> " yi yükle	...	L 1.25 E2
"A" işaretini yükle	'.'	L 'A'
"ABC" işaret dizisini yükle (max. 4 = 32 bit )	'...'	L 'ABC'
16 bit sayaç sabitesini yükle: Start değeri = "725"	C#	L C#725
S5 zaman sabitesini yükle: "1 saat 22 dk 15 sn"	T#	L T#0D1H22M15S
Tarih yükle: "2 Ağustos 1998"	D#	L D#1998-8-2
"START" değişkeni göstergesini yükle	P#	L P#START

Global datamodül <b>genişliğini</b> Akü 1'e yükle	L DBLG
Global datamodül <b>numarasını</b> Akü 1'e yükle	L DBNO
Özel (instanz) datamodül <b>genişliğini</b> Akü 1'e yükle	L DILG
Özel (instanz) datamodül <b>numarasını</b> Akü 1'e yükle	L DINO

**TRANSFER:** Akümülatördeki verilerin bir data hedefine atanmasıdır. Bu data hedefi giriş-çıkış değişkenleri, durum tespit işaretleri ve data alanları olabilir. Transfer edilecek veri bayt ,word veya doppelwort olabilir.

Yükleme ve transfer emirlerinin işlev şeması olarak yazılımı S7 sistemlerinde mümkün olmasına rağmen, deyim listesi olarak yazılması daha uygundur. Bu sayede emir kapasitesinin çoğunluğunun kullanıldığı problemlerde yavaş yavaş deyim listesi yazılım şekline dönülmesi gerektiğinin görülmesi gerekir.

Yükleme ve transfer etme operasyonları daima 32 bit genişliğindeki register "AKÜ 1" üzerinden

gerçekleşir. Yüklenen bilgiler "AKÜ 1" içerisinde daima sağa yaslı olarak düzenlenirler. 32 bitten daha az kapsamlı iseler geri kalan bitler sıfırla tamamlanır.

Yükleme ve transfer operasyonları şart'a bağlı çalışmazlar. Program işleme sırası o satıra geldiğinde "L" ve "T" emirleri işlevlerini yerine getirirler.

Yazılım:

**L EW 124** → Önce giriş word' u 124 (124 ve 125. Bayt' lar) değişkenleri "Akü 1" yükleniyor. Yüklemeye anında "Akü 1" içerisindeki değer "Akü 2" ye kaydırılır. "Akü 2" içerisindeki değer silinir. Yüklemeye yapılan kaynakta herhangi bir değişiklik meydana gelmez.

**T AW 124** → Transfer işleminde o anda "Akü 1" içerisinde bulunan değer, belirlenen hedef adrese ( çıkış kanalı 124) kopyalanır. Transfer esnasında "Akü 1" içerisinde herhangi bir değişiklik olmaz.

Yüklemeden önce	AKÜ 2	YYYYYYYY	YYYYYYYY	YYYYYYYY	YYYYYYYY
	AKÜ 1	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
Yükledikten sonra	AKÜ 2	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX
	AKÜ 1	00000000	00000000	EB 125	EB 124

S7 de **Online** durumunda AKÜ ve diğer register içeriklerini görmek mümkündür. Bunun için;

**Extras→Einstellungen→AWL** (Standart AKÜ 1'in içeriğidir)

## ZAMAN FONKSİYONU

Zaman fonksiyonunun oluşturulması kumanda tekniğinin temel lojik fonksiyonlardandır. **Zaman elemanlarının görevi bir giriş (start) sinyaline, zamana bağlı çıkış sinyali olarak cevap vermektir.** Üretici firmaya göre çeşitli sayı ve fonksiyonlara sahip zaman elemanları (TIMER) hazır bir şekilde kullanıma sunulmuştur. Kullanım için basit bir programlama yeterlidir.

Zamanlayıcılar ile aşağıdaki program fonksiyonları yapılabilir.

- Bekleme süreleri
- Gözetim süreleri
- Zaman ölçümü
- Palsler oluşturma

Aşağıdaki zaman elemanları **Siemens** firmasının PLC ailesi **SIMATIC S7'** e göre açıklanmıştır. Fakat kullanım veya programlama farklı olmakla birlikte, benzer fonksiyonları diğer firmalarda da bulmak mümkündür.

Zaman elemanlarının intern olarak nasıl çalıştığı normal şartlarda kullanıcıyı ilgilendirmemesi gerekir. Çünkü kullanım için önemli olan zaman elemanın fonksiyonudur.

Basit PLC cihazlarında analog zaman elemanları kullanılır. Kondansatörlerin doldurulması ve boşaltılması bu elemanların temel prensibidir. Fakat genelde **Dijital zaman elemanları kullanılmaktadır. Bu prensipte intern oluşturulan sayma darbeleri bir geri sayıcıya iletilir.** Bu durumda bir zaman elemanın start yapılması sayıcıya belirli bir sayının yüklenmesi demektir. Sayıcı intern sayma darbeleri ile sifıra ulaştığında zaman bitmiştir. Programın seriel işlenmesi bunlardan etkilenmemektedir.

Ayrıca çok önemli bir nokta, kullandığımız zaman elemanlarının statik veya dinamik olup olmadıklarını bilmektir. Zaman elemanı diyagramlarına baktığımızda, giriş sinyali RESET yapılması sırasında ve sonrasında "1" olmakla beraber, RESET ten sonra zaman elemanın çıkışı "0" kalmaktadır. Bunun nedeni Siemens firmasının ürettiği PLC' nin dinamik zaman elemanlarına sahip olmasıdır. Yani **bir zaman elemanın start yapılabilmesi için giriş sinyalinin pozitif veya negatif bir değişim göstermesi gerekir.**

## ZAMANIN AYARLANMASI

S7 de zaman değeri, rezerve edilmiş olan CPU-RAM bölümüne iki şekilde aktarılır.

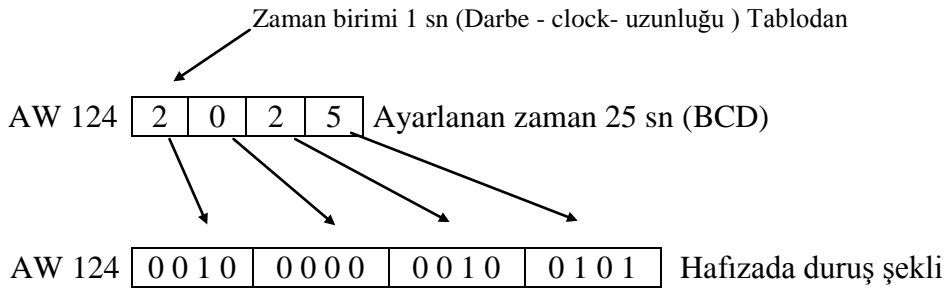
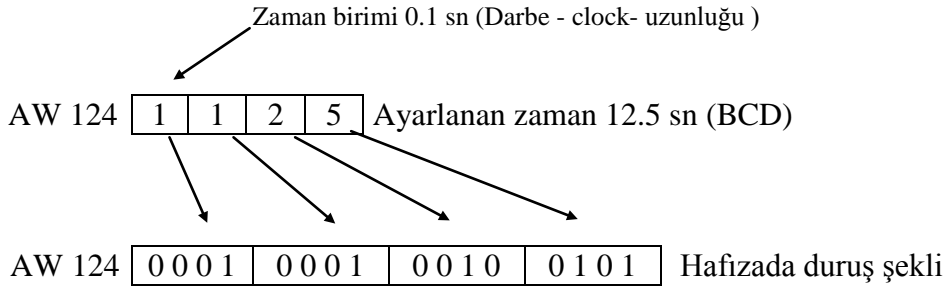
	Operand	Açıklama	Örnek
S5 formatı	S5T#aH_bcM_cdS_efgMS	Minimum süre :10 milisaniye Maksimum süre : 2saat 46 dak.30 sn	İstenilen: 2 dak. 21 sn L S5T#2m21s
Word formatı	W#16#wxyz	w = Zaman birimi 0 = 10 msn 1 = 100 msn 2 = 1 sn 3 = 10 sn xyz = BCD Format olarak zaman değeri	İstenilen: 2 dak. 21 sn (örn: 141 sn) L W#16#2141

Zaman değeri wort olarak tanımlandı ise, zamanın hesaplanması ve hesap sonucunun örn. Bir merkerwort' a atılması olanağı da vardır. S5 formatı ile de zaman değeri üzerinde işlem ( merkerword'a atılabilir ) yapılabilir,ancak o değere tam hakim olmak zordur. Zaman değeri dual formatta ve zaman birimi zaman değerine bağlı olarak değişmektedir. Wort olarak tanımlanan zaman değerine tam olarak hakim olmak mümkündür.

Zaman değeri istenilen anda bir hafıza alanına alındıktan sonra oradan çağırılması sadece basit bir "L MW xxx" emri ile yapılabilir.

S7 300 zamanlayıcılarında en az 10 msn, en çok 2 saat 46 dak 30 sn zaman değeri verilebilir

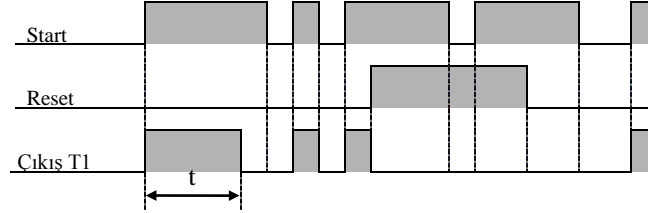
Zaman elemanlarını bit veya sayısal olarak sorgulayabiliriz. Eğer zaman elemanın sahip olduğu durumu bit olarak öğrenmek istiyorsak bu VE veya VEYA fonksiyonlarıyla yapılabilir.



## S7-300 de ZAMAN ELEMANLARININ ÇEŞİTLERİ

### "SI" Darbe (Pals)

Giriş sinyalinin pozitif bir değişim göstermesiyle zaman elemanı çalışmaya başlar. Çıkış da hemen "1" değerini alır. Süre bitiminde veya giriş sinyalinin sıfır olması durumunda çıkış da hemen sıfır olur. Bu ayrıntılar diyagramlara bakılarak daha iyi anlaşılabilir. Bir zaman elemanın doğru olarak kullanılabilmesi



için özelliklerinin tam olarak bilinmesi gerekir.

<p style="text-align: center;"><b>T1</b></p>	<p>(S5 Formatında)</p> <pre> U E 124.0 L S5T#3s SI T1 U E 124.1 R T1  U T1 = A 124.0 </pre>	<p>(Word olarak)</p> <pre> U E 124.0 L W#16#1050 SI T1 U E 124.1 R T1  U T1 = A 124.0 </pre>
İşlev Şeması	Deyim listesi	

**Örnek:13 İKİ BUTON İLE BİR MOTORUN KUMANDASI**

Bir motor iki buton ile kumanda edilecektir. Herhangi bir butona basılınca motor çalışacak, yine herhangi bir butona basılınca duracaktır.

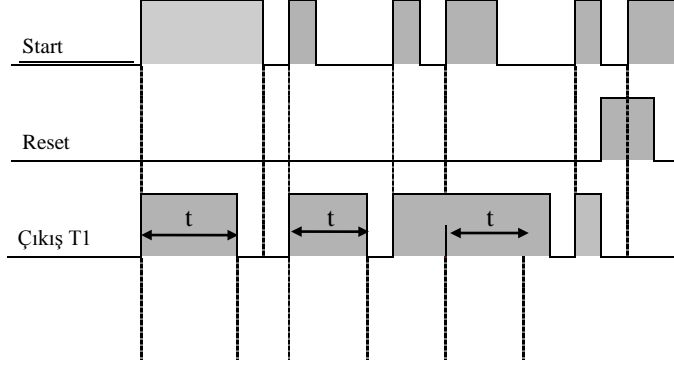
Butonlardan birinin kontakları birbirine yapışırsa 2 saniye sonra yapışan kontak devre dışı olacak ve diğer buton görevine devam edecektir. Birinci butonun arızalandığını L1 lambası ikinci butonun arızalandığını L2 lambası gösterecektir.

ATAMA LİSTESİ		
Operand	Sembol	Açıklama
E 124.0	S1	Açma kapama butonu (NA)
E 124.1	S1	Açma kapama butonu (NA)
A 124.0	M	Motor kontaktörü
A125.0	L1	1.Buton arıza lambası
A125.1	L2	2.Buton arıza lambası

U E 124 . 0 UN M 2 . 0 O U E 124 . 1 UN M 2 . 1 = M 0 . 1	U E 124 . 0 UN T1 S M 2 . 0 UN E 124 . 0 R M 2 . 0  U E 124 . 1 UN T2 S M 2 . 1 UN E 124 . 1 R M 2 . 1	U M 0 . 1 FP M 0 . 2 = M 0 . 0  UN A 124 . 0 U M 0 . 0 S M 0 . 3 U A 124 . 0 U M 0 . 0 R M 0 . 3  U M 0 . 3 = A 124 . 0 Lamba  U M 2 . 0 = A 125 . 0 1.Buton arızası  U M 2 . 1 = A 125 . 1 2.Buton arızası BE
U E 124 . 0 1.Buton L S5T#2S için SI T1 süre		
U E 124 . 1 2.Buton L S5T#2S için SI T2 süre		

## "SV" Uzatılmış darbe

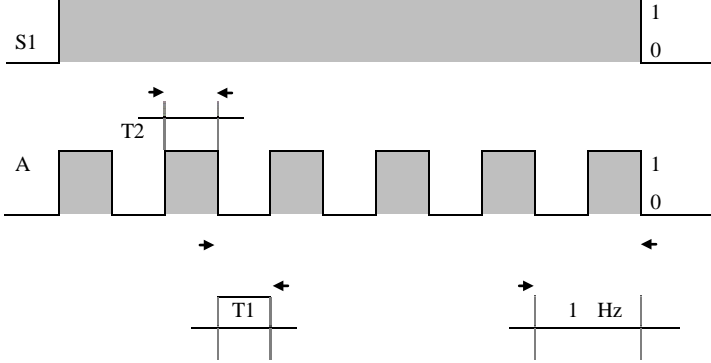
Start sinyalinin bir değişim göstermesiyle zaman elemanı çalışmaya başlar. Çıkış hemen "1" değerini alır. Zaman elemanı çalışmaya devam ederken start sinyali yeni bir değişim gösterirse zaman yeniden start yapılır. Zaman elemanlarının durumunu bit olarak sormak için VE veya VEYA kapıları kullanılabilir. İstenirse zaman elemanın bit olarak durumu bir durum tespit işaretine veya çıkış değişkenine de atanabilir.



<p style="text-align: center;"><b>T1</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">S_VIMP</th> </tr> </thead> <tbody> <tr> <td style="width: 50%;">E 124.0 — S</td> <td style="width: 50%;">DUAL — ???</td> </tr> <tr> <td>S5T#3S — TW</td> <td>DEZ — ???</td> </tr> <tr> <td>E 124.1 — R</td> <td>Q — A 124.0</td> </tr> </tbody> </table>	S_VIMP		E 124.0 — S	DUAL — ???	S5T#3S — TW	DEZ — ???	E 124.1 — R	Q — A 124.0	<pre> U   E 124.0 L   S5T#3s SV  T1 U   E 124.1 R   T1 U   T1 =   A 124.0 </pre>
S_VIMP									
E 124.0 — S	DUAL — ???								
S5T#3S — TW	DEZ — ???								
E 124.1 — R	Q — A 124.0								
<b>İşlev Şeması</b>	<b>Deyim Listesi</b>								

**Örnek:14** 1 Hz' lik bir takt üretici oluşturulmak istenmektedir. Çıkış değişkeni (lamba), "S1" anahtarına basılmasıyla hemen yanmaya başlamalıdır.

#### Fonksiyon Diyagramı



#### Atama Listesi

S1 = E 124.0 Açma / Kapama anahtarı  
A = A 124.0 Lamba

Böyle bir uygulamada hangi tür zaman elemanın ve ne şekilde kullanılacağına karar vermek çok önemlidir.

Zaman elemanın seçiminde dikkate almamız gereken en önemli özellik giriş sinyalinin gelmesi ile birlikte çıkışın alınması gerekliliği ve sürenin start sinyali ile başlamasıdır. Bunu sağlayan "SI" ve "SV" zaman elemanlarıdır.

Böyle bir program çözümünde yanma ve sönme zaman değerlerini de farklı yapabileceğimiz en kolay mantık yanma için ayrı, sönme için ayrı bir zaman elemanı kullanmaktır.

```
U E 124.0
UN T 2
L S5T#1S
SI T 1
```

//

```
U E 124.0
UN T 1
L S5T#1S
SI T 2
```

//

```
U T 1
= A 124.0
```

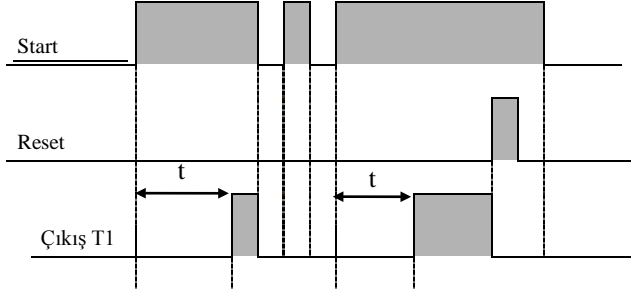
"SI" tipi zaman elemanı ile yapılan çözümde giriş sinyali gittiğinde lamba hemen söner, "SV" tipi zaman elemanında giriş sinyali gittiğinde lamba ayarlanan süre kadar yanmaya devam eder.

Diğer zaman elemanları ile de farklı özelliklerde takt üretici elde edilebilir.

Not: S7 – 300 sabit frekanslarda clock üreticine (Taktmerker'e) sahiptir

**"SE" Girişin uzatılması (Gecikmeli start)**

Bu zaman elemanın özellikleri de ilgili diyagramdan öğrenilebilir. Start sinyali alındığında süre başlar. Çıkış önce sıfırdır, ayarlanan süre bitiminde "1" olur ve start sinyali kaybolana kadar "1" de kalır. Start sinyali daha önce yok olursa, çıkış "0" da olur.

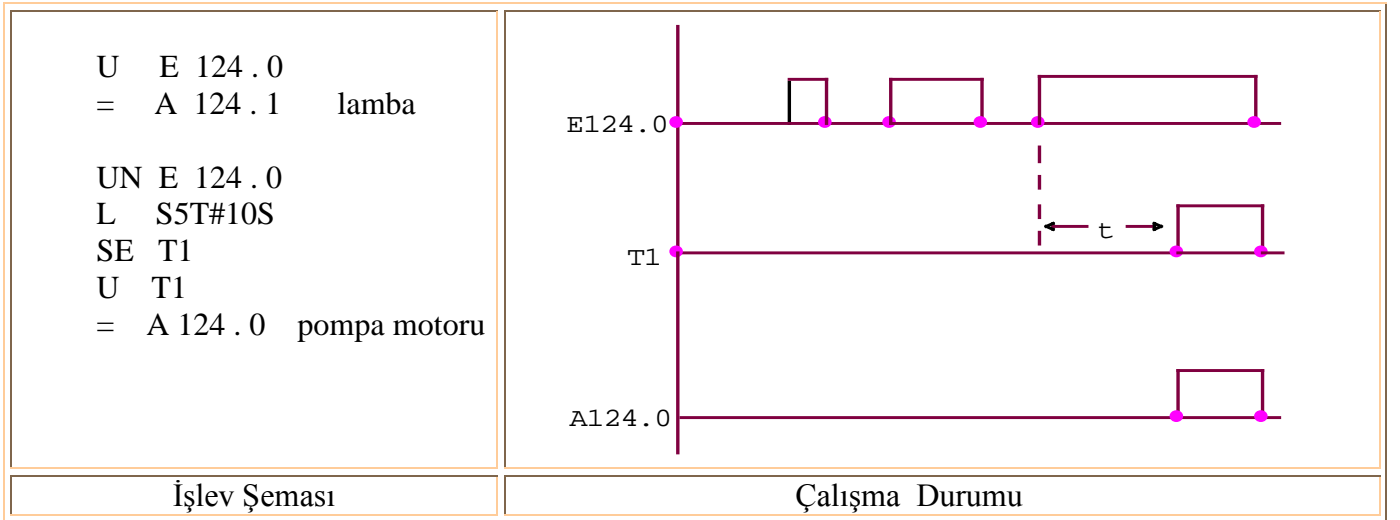
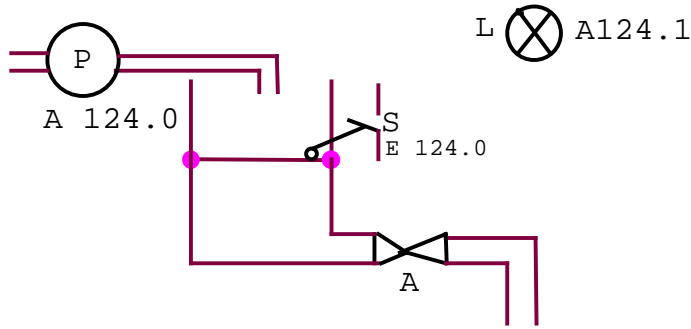


<p style="text-align: center;"><b>T1</b></p> <pre> S_EVERZ E 124.0 — S   DUAL — ??? S5T#3S — TW  DEZ  — AW 124 E 124.1 — R   Q    — ??? </pre>	<pre> U E 124.0 L S5T#3S SE T 1 U E 124.1 R T 1 // LC T 1 T AW 124 </pre>
<b>İşlev Şeması</b>	<b>Deyim Listesi</b>

**Örnek :15 SU POMPASI KUMANDASI**

Şekildeki P pompası S seviye butonu ile kumanda edilmektedir. Su seviyesi yükseldiğinde seviye butonu kapanacak P pompası duracak. Su aşağıda A vanasından boşalacaktır. Su seviyesi düşünce S seviye butonu açılırsa P pompası ayarlanan süre sonunda çalışacaktır. (örneğin 10 sn) Havuzdaki suyun dalgalanması ile S seviye butonu açılıp kapanabilir,kumanda bundan etkilenmeyecektir.S seviye butonunun kapalı olduğunu L lambası bildirecektir.

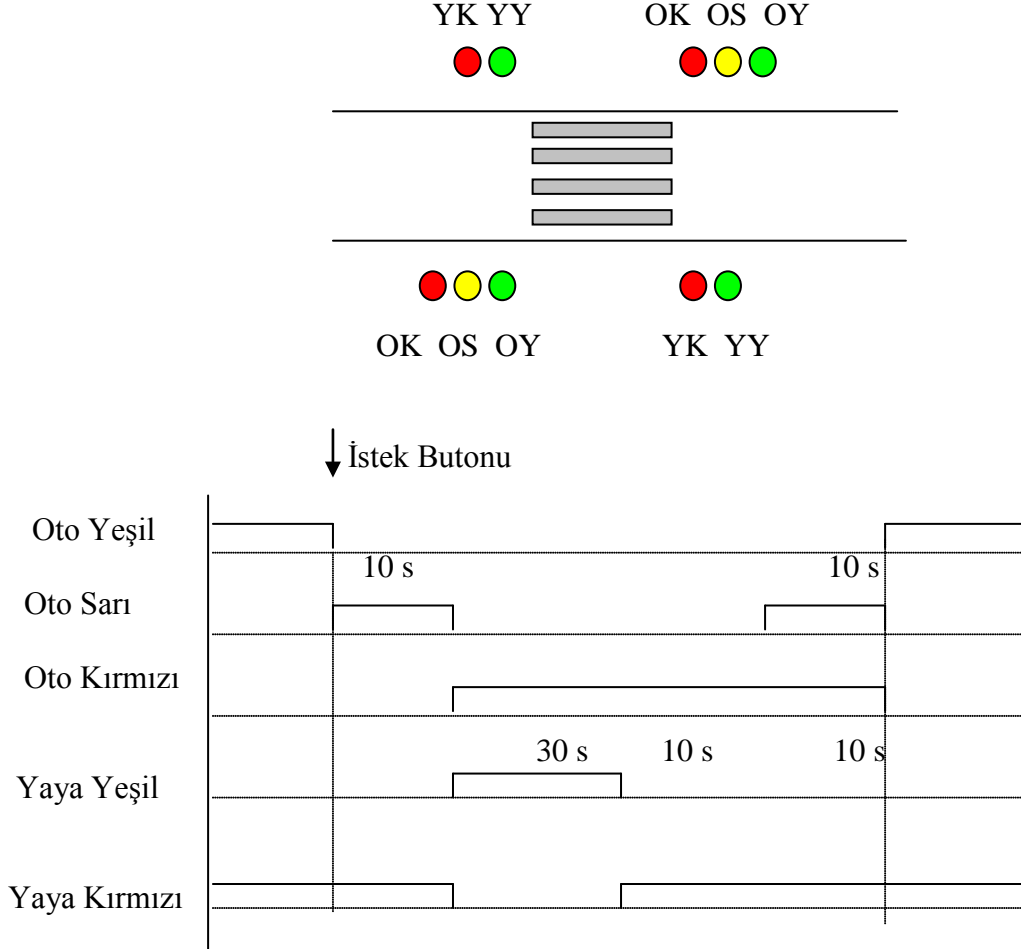
Teknoloji Şeması



**Örnek:16 TRAFİK LAMBASI KUMANDASI**

Şekildeki yaya geçidinde bulunan trafik lambaları PLC ile kumanda edilecektir.

Trafik lambalarının başlangıç durumu araçlar için yeşil ve yayalar için kırmızıdır. Yayalar karşıdan karşıya geçmek istediğinde “ S ” istek düğmesine basacak, otolar için trafik lambası önce yeşilden sarıya sonra kırmızıya dönecektir ve aşağıdaki diyagrama göre çalışacaktır.



- \*Sarı lamba araçlar için 10 sn
- \*Yeşil lamba araçlar için 30 sn
- \*Kırmızı lamba araçlar için 10 sn
- \*Yeşil lamba yayalar için 30 sn
- \*Kırmızı lamba yayalar için 30 sn yanık kalmaktadır.

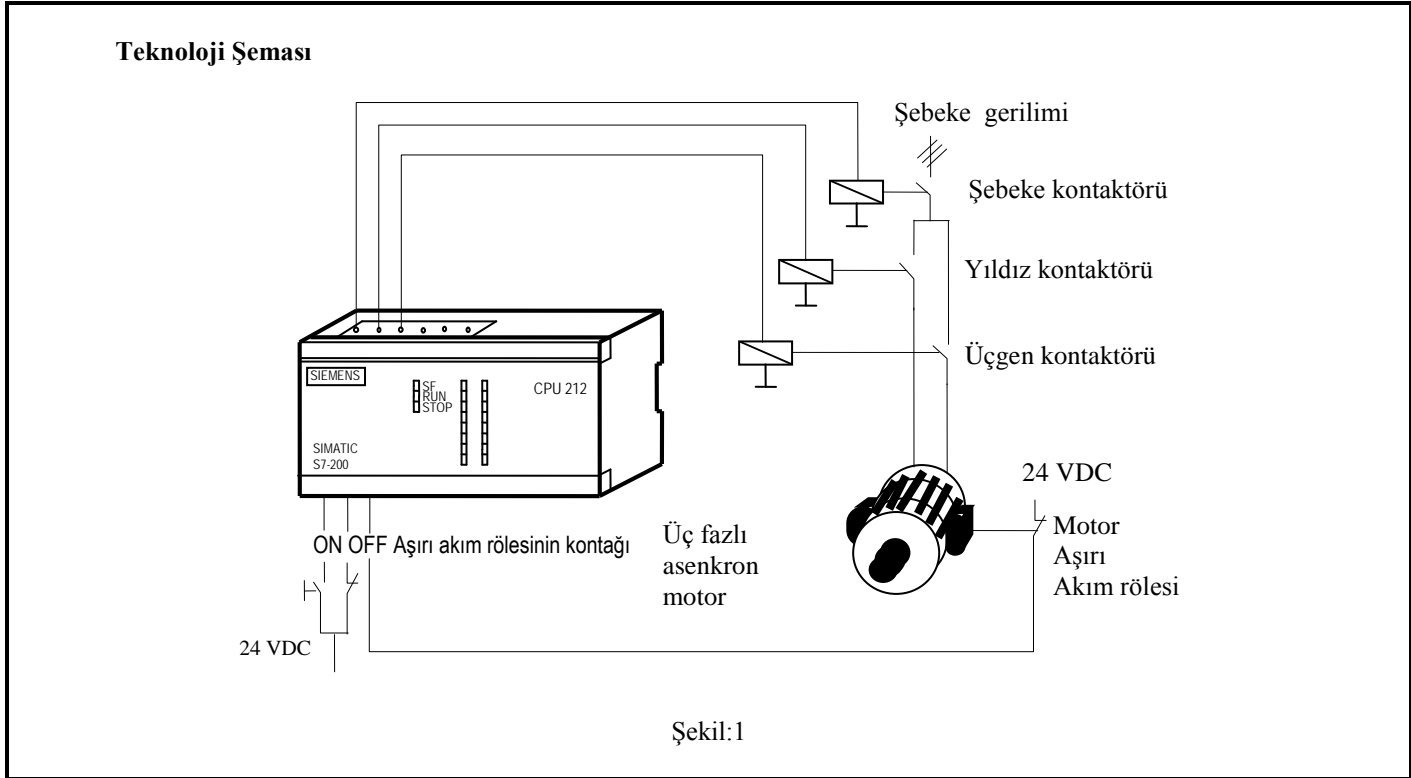
ATAMA LİSTESİ		
Operand	Sembol	Açıklama
E 124.0	S	Yaya istek butonu
A 124.0	S1	Otolar için yeşil trafik lambası
A 124.1	M	Otolar için sarı trafik lambası
A124.2	L1	Otolar için kırmızı trafik lambası
A124.3	L2	Yayalar için yeşil trafik lambası
A124.3	L3	Yayalar için kırmızı trafik lambası

<pre> UN M 0.0 = A 124.0  U( O E 124.0 O M 0.0 ) UN T 4 = M 0.0  U M 0.0 L S5T#10S SE T 1 NOP 0 NOP 0 NOP 0 NOP 0  U M 0.0 U( ON T 1 O T 3 ) = A 124.1  U M 0.0 U T 1 = A 124.2 </pre>	<pre> U A 124.2 UN T 2 = A 124.3  U M 0.0 U T 2 L S5T#10S SE T 3 NOP 0 NOP 0 NOP 0 NOP 0  U M 0.0 U( ON T 1 O T 2 ) ON M 0.0 = A 124.4  U M 0.0 U T 3 L S5T#10S SE T 4 NOP 0 NOP 0 NOP 0 NOP 0 BE </pre>
--	--

**Örnek:17 BİR ASENKRON MOTORUN YILDIZ ÜÇGEN YOL VERİLMESİ**

Üç fazlı bir asenkron motor otomatik yıldız üçgen yol verilecektir. S<sub>1</sub> butonuna basılınca motor yıldız yol alacak, 5s sonra otomatik olarak üçgen çalışacaktır. S<sub>2</sub> butonuna basılınca motor duracaktır. Ayrıca motor aşırı akım rölesi ile korunacaktır.

ATAMA LİSTESİ		
Operand	Sembol	Açıklama
E 124.0	S0	Stop (NK)
E 124.1	S1	Start (NA)
E 124.2	F2	Aşırı akım rölesi (NK)
A 124.0	K1	Şebeke kontaktörü
A 124.1	K2	Yıldız kontaktörü
A 124.2	K3	Üçgen kontaktörü



## Yıldız üçgen yol verme

## SEBEKE KONTAKTÖRÜ

```

U   E 124 . 1
S   M  0 . 0
ON  E 124 . 0
ON  E 124 . 2
R   M  0 . 0
U   M  0 . 0
=   A 124 . 0

```

## ZAMANLAYICI

```

U   A 124 . 0
L   S5T#5S
SE  T0

```

## YILDIZ KONTAKTÖRÜ

```

U   E 124 . 1
S   M  0 . 1
ON  E 124 . 0
ON  E 124 . 2
O   T0
R   M  0 . 1
U   M  0 . 1
=   A 124 . 1

```

## ÜÇGEN KONTAKTÖRÜ

```

U   T0
S   M  0 . 2
ON  E 124 . 0
ON  E 124 . 2
R   M  0 . 2
U   M  0 . 2
=   A 124 . 2

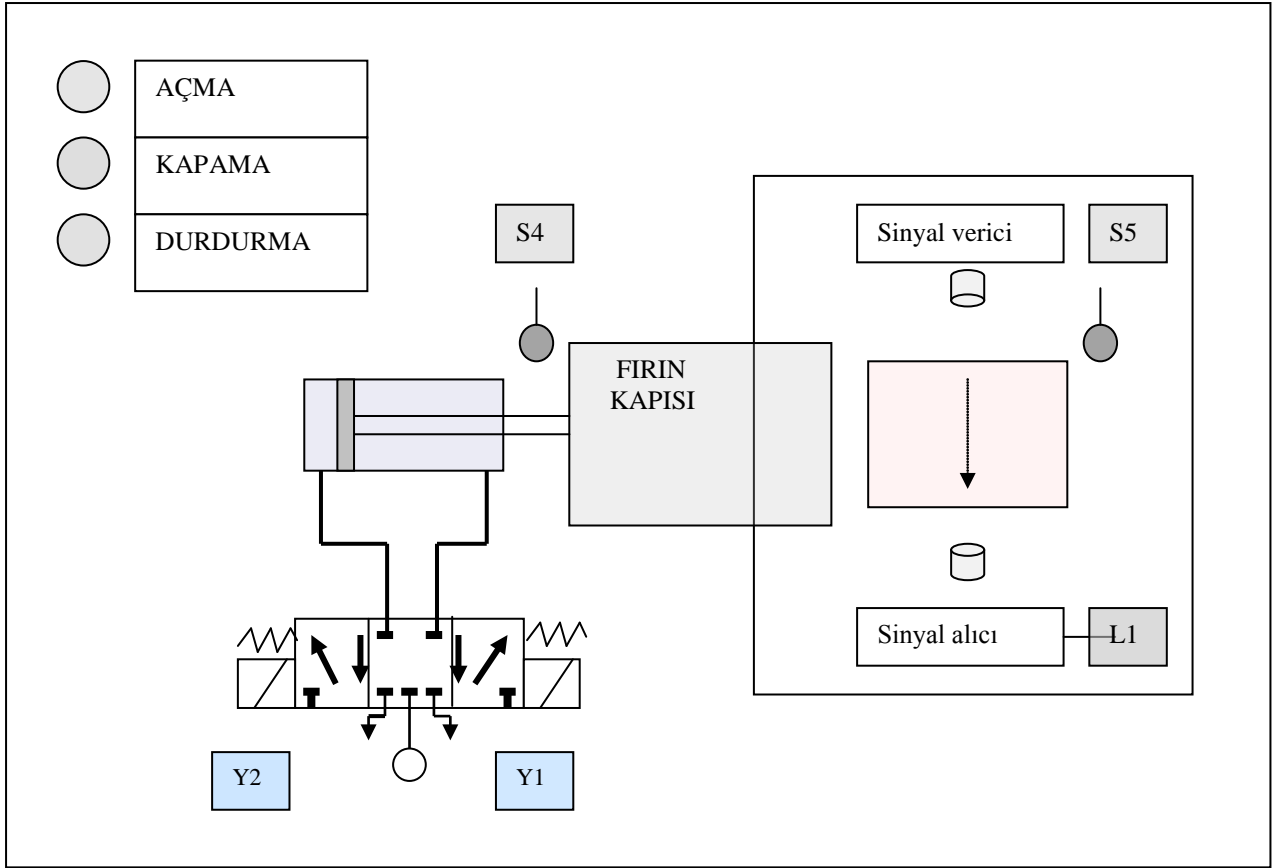
```

**Örnek :18 FIRIN KAPISI KUMANDASI**

Bir fırın kapısı açma , kapama ve durma olmak üzere üç fonksiyonlu bir silindirle kumanda edilmek istenmektedir. Başlama durumunda kapı kapalıdır.

- \* S1 butonuna basılması ile kapı açılmaya başlamalıdır ve S4 sınır anahtarı ile açma işlemi sona ermelidir.
- \* Kapı en açık duruma ulaştıktan sonra 6 saniyelik bir zamanın geçmesiyle veya S2 butonu ile kapatılacaktır.
- \* Kapının kapanması S5 sınır anahtarı ile sona erecektir.
- \* Kapama işlemi optik sinyal vericimizin uyarılması ile hemen durdurulmalı ve optik sinyal vericimizin önündeki engelin gitmesi ile kapama işlemi tekrar devam etmelidir.
- \* Kapının iki hareketi karşılıklı olarak kilitlenmelidir

\*\*\* Burada kullanılan valf çift bobin kumandalı 5/3 yay merkezlemeli orta konumlu bir valftir.



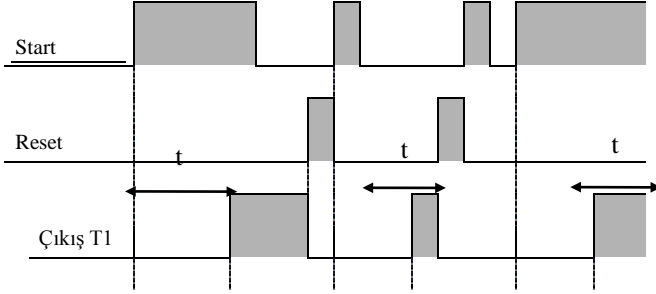
ATAMA LİSTESİ		
Operand	Sembol	Açıklama
E 124.1	S1	Açma (NA)
E 124.2	S2	Kapama (NA)
E 124.3	S3	Durdurma
E 124.4	S4	Sınır anahtarı (NK)
E 124.5	S5	Sınır anahtarı (NK)
E 124.6	L1	Sinyal verici (NK)
A 124.0	Y1	Açma
A 124.1	Y2	Kapama

## Fırın kapısı kumandası

U S1 S M0.0 O M0.1 ON S3 ON S4 R M0.0  U M0.0 = Y1  O T1 O S2 S M0.1 O Y1 ON S3 ON S5 R M0.1	U M0.1 U L1 = Y2  UN S4 L S5T#6S SE T1 BE
--	--

**"SS" Hafızalı gecikmeli start**

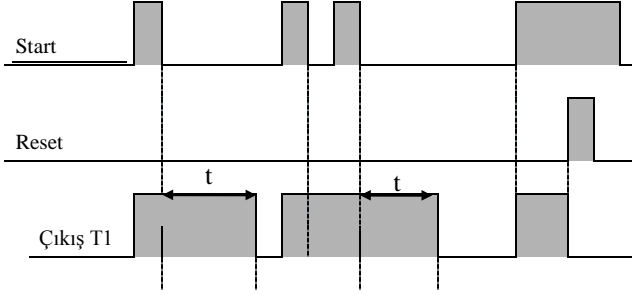
SE deki özellikler ile aynıdır fakat start sinyali gelip gitse bile ayarlanan süre sonunda çıkış "1" olur. Start sinyali ayarlanan süre bitiminden önce tekrar gelirse, o zaman süre yeniden başlatılır. Çıkış ancak RESET edilerek tekrar "0" yapılabilir.



<p style="text-align: center;"><b>T1</b></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tbody> <tr> <td colspan="2" style="text-align: center;">S_SEVERZ</td> <td></td> <td></td> </tr> <tr> <td>E 124.0</td> <td>S</td> <td>DUAL</td> <td>MW 16</td> </tr> <tr> <td>W#16#2003</td> <td>TW</td> <td>DEZ</td> <td>???</td> </tr> <tr> <td>E 124.1</td> <td>R</td> <td>Q</td> <td>A 124.0</td> </tr> </tbody> </table>	S_SEVERZ				E 124.0	S	DUAL	MW 16	W#16#2003	TW	DEZ	???	E 124.1	R	Q	A 124.0	<pre> U E 124.0 L W#16#2003 SS T 1 U E 124.1 R T 1 L T 1 T MW 16 U T 1 = A124.0 </pre>
S_SEVERZ																	
E 124.0	S	DUAL	MW 16														
W#16#2003	TW	DEZ	???														
E 124.1	R	Q	A 124.0														
İşlev Şeması	Deyim Listesi																

**"SA" Çıkışın uzatılması (Gecikmeli stop)**

Diğer zaman elemanları ile benzer özelliklere sahiptir. Fonksiyonu diyagramlardan öğrenilebilir. Giriş sinyalinin "1" olmasıyla çıkışta "1" olur. Fakat giriş sinyali "0" olduktan sonra



çıkış sinyali ayarlanan süre kadar "1" olmaya devam eder. Reset sinyali olmadan yeni değer yüklenmez.

<p><b>T1</b></p>	<pre> U E 124.0 L S5T#3S SA T 1 U E 124.1 R T 1 </pre>
İşlev Şeması	Deyim Listesi

Bir çok kumanda problemlerinde verilen zamanın bitip bitmediği değil, ayrıca ne kadar zaman geçtiğinin önemi vardır. Zaman elemanlarının sayısal değeri ikili (binâr) veya BCD sayı olarak sorgulanabilir. örneğin bu yolla zamanı bir göstergeye aktarabiliriz. Böylece zamanın sürekli olarak takibi mümkündür.

L T 1	1 nolu zaman elemanı içeriğinin dual olarak AKÜ 1'e yüklenmesi
T MW 16	AKÜ 1 içeriğinin ( bir önceki satırda zaman elemanı içeriği yüklenmişti) MW 16'ya yüklenmesi
LC T 4	4 nolu zaman elemanı içeriğinin BCD olarak AKÜ 1'e yüklenmesi
T AW 124	AKÜ 1 içeriğinin ( bir önceki satırda zaman elemanı içeriği yüklenmişti) çıkış word'una transfer edilmesi

**Örnek :19** Bir motor çalışma emri verildikten 2.5 s sonra dönmeye başlaması ve dur emri verildikten 2.5 s sonra durması gerekmektedir.

**Start :E 124.0      Motor:A 124.0**

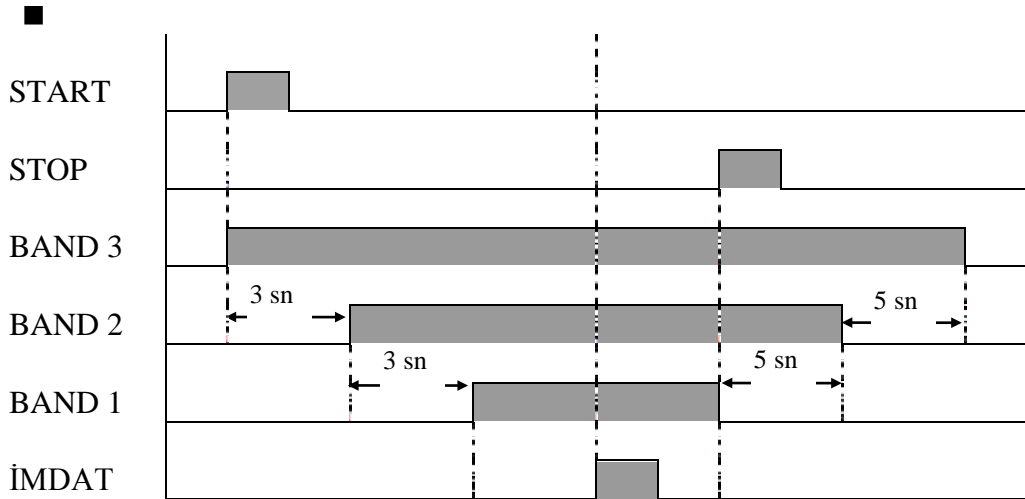
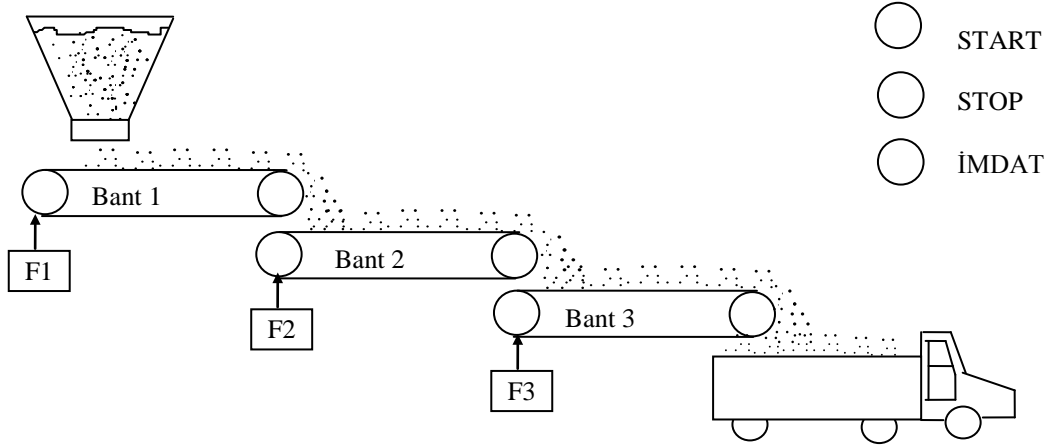
```
U  E 124 . 0
L  W#16#1025
SI  T1
SA  T2
UN  T1
U   T2
=   A 124 . 0
```

Burada zaman değeri word formatında verilmiştir. (LW#16#1025)  
Zaman değeri S5 formatında da verilebilir. (L S5t#2,5S)

**Örnek: 20** Taşıma bantları kumandası

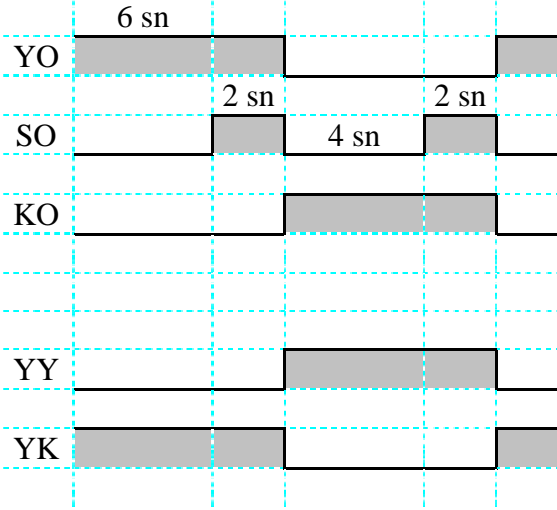
Bir doldurma tesisinde ardışık olarak çalışan üç bant yardımıyla kamyonlar doldurulacaktır.

- Start butonuna basıldığında 3. Bant hemen, 2. Bant 3 sn sonra ve 1. Bant ise 6 sn sonra çalışacaktır.
- Stop butonuna basıldığında 1. Bant hemen, 2. Bant 5 sn sonra ve 3. Bant 10 sn sonra duracaktır.
- Acil stop butonuna basıldığında bütün bantlar hemen duracaktır.
- Ayrıca her bant birer termik ile korunmaktadır.
  - Birinci banda ait termik attığında 1. Bant,
  - İkinci banda ait termik attığında 1 ve 2. Bantlar,
  - Üçüncü banda ait termik attığında her üç bantda hemen duracaktır.



SEMBOL	OPERAND	AÇIKLAMA
K1	A 124.0	1.Bant motoru
K2	A 124.1	2.Bant motoru
K3	A 124.2	3.Bant motoru
F1	E 124.5	1.Motor için termik (NK)
F2	E 124.6	2.Motor için termik (NK)
F3	E 124.7	3.Motor için termik (NK)
S0	E 124.0	İMDAT (NK)
S1	E 124.1	Start (NA)
S2	E 124.2	Stop (NK)

**Uygulama :21** Yaya geçidi olan bir yoldaki trafik lambaları "S0" anahtarı ile kontrol edilecektir. "S0" anahtarı kapatıldığında ( Gündüz çalışması ) trafik lambaları şekildeki diyagrama göre çalışacaktır."S0" anahtarı açıldığında (Gece çalışması) lambalar sönecektir,sarı lamba 1 Hz lik bir sinyal ile yanıp sönecektir.



### ATAMA LİSTESİ

OPERAND	SEMBOL	AÇIKLAMA
E 124.0	S0	Çalıştırma anahtarı
A 124.0	K0	Otolar için kırmızı lamba
A 124.1	S0	Otolar için sarı lamba
A 124.2	Y0	Otolar için yeşil lamba
A 125.0	YK	Yayalar için kırmızı lamba
A 125.1	YY	Yayalar için yeşil lamba

## Trafik lambası kontrolü

## Otolar için yeşil lamba

```

UN A124 . 0
U E124 . 0
UN T0
L ST5#6S
SV T0
O A124 . 0
O T5
ON E 124 . 0
R T0
U T0
= A124 . 2

```

## Otolar için yeşil ile beraber sarı lamba

```

U T0
L ST5#4S
SS T1
ON T0
O T5
ON E 124 . 0
R T1

```

## Otolar için kırmızı lamba

```

UN T0
U E124 . 0
UN A124 . 2
L ST5#6S
SV T3
UN E124 . 0
R T3
U T3
= A124 . 0

```

## Otolar için kırmızı ile beraber sarı lamba

```

U T3
L ST5#4S
SE T2
ON T3
ON E124 . 0

```

## Sarı lamba

```

O T1
O
UN E124 . 0
U T10
O T2
= A124 . 1

```

```

U A124 . 1
L ST5#4S
SE T5

```

## Yayalar için kırmızı lamba

```

U A124 . 2
= A125 . 0

```

## Yayalar için YEŞİL lamba

```

U A124 . 0
= A125 . 1

```

## 1 Hz'lik clock generatörü

```

UN T11
UN E124 . 0
L S5T#500MS
SV T10

```

```

UN T10
L S5T#500MS
SI T11
BE

```

## SAYMA FONKSİYONU KUMANDA TEKNİĞİNDE SAYMA

Belirli bir miktarın belirlenmesi birçok durumda darbelerin sayılması ile gerçekleştirilmektedir. Bunun için de bu darbeler bir sayıcıya gönderilir. ikili veya BCD olarak sorgulayabileceğimiz sayıcı durumu da bize belirlemek istediğimiz miktarı verir. Bu şekilde aşağıdaki problemlerle karşı karşıya kalabiliriz:

- Bir miktarın sayılması
- Bir istek miktarına göre sahip olunan değer eşit, küçük veya büyük olup olmadığının araştırılması
- Çeşitli miktarlar arasında farkın belirlenmesi vb.

Sayıcılar bazı ek elemanların kullanılması ile daha değişik fonksiyonları da gerçekleştirebilirler.

## SAYMA FONKSİYONLARI Sayıcıların Programlanması

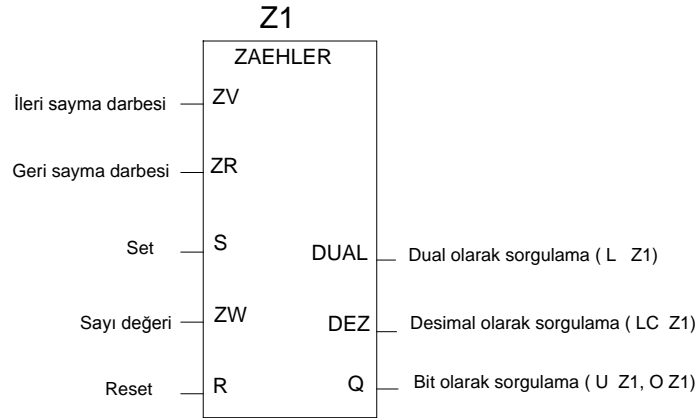
Temel olarak PLC sistemleri üç çeşit sayma fonksiyonuna sahiptirler.

a) -CPU' da Software olarak hazırlanmış ve her zaman çağırılacak sayıcılar. Bunlar belirli bir emir listesine sahiptirler ve kullanımları bu emirlere bağlı kalınarak yapılır. örneğin S 7 de sayıcılar CPU cinsine göre Z 1...Z 64 adını alırlar. Bu sayıcılar her çevrimde sadece bir sinyal olmak üzere ileri veya geri sayma yapabilirler. işleme süresi PLC' nin hızına bağlıdır.

b) -Durum tespit işaretleri (merker) ile de sayıcı yapmak mümkündür. ileri veya geri sayma işlemi toplama veya çıkarma emirleri ile yapılmaktadır. Bu tür sayıcılar bir çevrimde birden fazla extern darbeyi ileri veya geri sayabilir.

c) -Bazı sayıcılar da özel bir modüle yerleştirilip, hızlı sayıcılar olarak adlandırılmaktadırlar. Darbeler de direkt CPU' ya bağlanırlar. Bu sayıcılar çevrim süresinden daha kısa olan sinyalleri sayabilecek durumdadırlar.

Aşağıdaki şekilde software olarak hazırlanmış bir sayıcıyı tam olarak gösterebiliriz.



Görüldüğü gibi zaman elemanları ile sayıcılar arasında büyük benzerlik vardır. Bunun nedeni ikisinin de hardware olarak aynı olmalarıdır. Sayma yaptığımızda takt sinyalleri dışarıdan gelir. Zaman elemanları için ise intern bir clock generator mevcuttur.

**İLERİ SAYMA :** Giriş sinyali olarak giriş, çıkış veya durum tespit işaretleri kullanılabilir. Giriş değişkenindeki pozitif bir değişim (0'dan 1'e sayıcı değerini "1" arttırır. S7 de sayıcı değerini maksimum 999 yapabilir. Bundan sonra durumunda bir değişim olmaz.

**GERİ SAYMA :** Pozitif veya negatif bir değişim ile sayıcı değerini "1" azaltır. S7 de sayıcının sahip olabileceği minimum değer "0"dır. Bundan sonra sayıcı devamlı "0" olarak kalır. Yani negatif sayılara gitmez.

**SET :** Bu girişteki sinyalin pozitif değişim göstermesiyle sayıcıya bir ön sayı, yani fonksiyonları da sayma işleminin hangi sayıyla başlayacağı verilmiş ve sayıcı aktif hale getirilmiş olur. Bu ön sayı da "ZW" kısmında yazılı olan sayıdır. Bu sayı iki şekilde verilebilir.

- direkt bir sayı olarak, örneğin "L 15" veya
- indirekt bir adres olarak, örneğin EW 12.(giriş word'u 12)

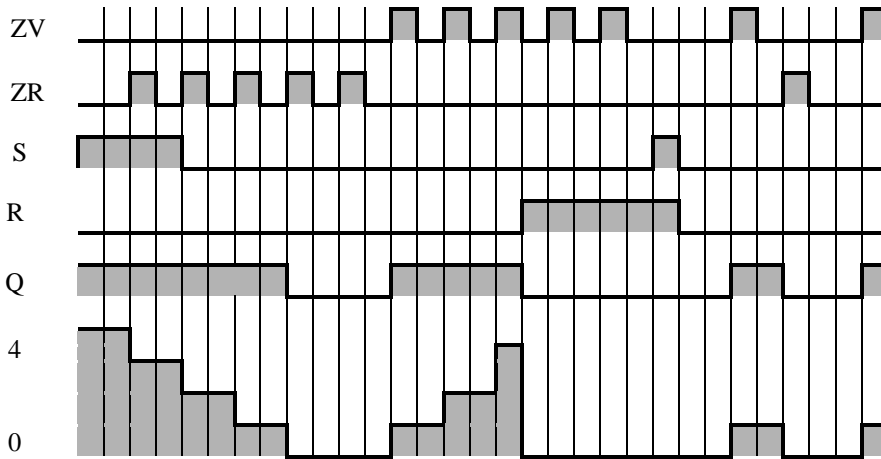
**RESET :** Buradaki bir 1 sinyali ile sayıcının değeri 0'a getirilir. Sayma veya SET işlemi RESET süresince yapılamaz.

**FREIGABE :** Set sinyalinin "1" olması şartı ile "FR" girişinden "1" sinyali verildiğinde sayıcı set değerine yeniden setlenir. ( set sinyali "0" dan "1" e yükselmiş gibi ) Freigabe sadece AWL yazılım için geçerlidir.

## SAYICI DURUMUNUN SORGULANMASI

### Bit olarak

Sayıcının değeri bit olarak VE veya VEYA mantığıyla sorulabilir. Sayıcı bit olarak "1" değerini ancak sayılan değer henüz "0" değilse alır. Başka bir şekilde, sayılan değer "0" ise bit olarak sonuç da "0"dır.



### Sayısal ( Nümerik ) olarak

Sayıcı içerisindeki sayısal değer (sayılan değer) "DU" çıkışında ikili (binâr 16 bitten oluşan bir kelime) sayı olarak veya "DE" çıkışında "BCD" olarak sorgulanabilir. Bunun için sayıcının içeriği önce akümülatöre gönderilir. Buradan da başka bir word veya byte adresine transfer edilir.

İşlev şeması	Deyim Listesi
<p style="text-align: center;"><b>Z1</b></p> <p style="text-align: center;"><b>ZAEHLER</b></p> <p>E 124.0 — ZV</p> <p>E 124.1 — ZR</p> <p>E 125.0 — S      DUAL — MW 126</p> <p>C#5 — ZW      DEZ — MW 128</p> <p>E 125.1 — R      Q — A 124.0</p>	<pre> U E 124.0 ZV Z 1 U E 124.1 ZR Z 1 U E 125.0 L C#5 S Z 1 U E 125.1 R Z 1 L Z 1 T MW 126 LC Z 1 T MW 128 U Z 1 = A 124.0 </pre>

Ayrıca S7 ler direkt taktmerkere sahiptir.

**S7 de Taktmerker oluşturma:** S7 Manager de proje çağrılır

**Simatic Station →Hardware** çift tıklanır→ **“Hardwarekonfiguration”** penceresi açılır. Burada CPU çift tıklanır→ **CPU Parametern** penceresi açılır→ **Register- Zyklus / Taktmerker** tıklanır→ **Taktmerker** start edilir ve merkerbyte tanıtılır.( Her merkerbyte taktmerker olarak kullanılabilir, Örneğin MB 20).

Yapılan ayarların CPU ya gönderilmesi için **“Menüleiste→Zielsystem→ Laden in Baugruppe.**

Böylece CPU uygun şekilde parametrelendirilmiş olur ve 1:1pals oranında Takt (darbe) verebilir. Takt frekansları sabit olarak aşağıdaki tabloda olduğu gibi tanımlanmıştır.

Merkerbyt kullanımı		
Bit 7	0.5 Hz	Yavaş yanıp sönen ışık
Bit 6	0.625 Hz	
Bit 5	1.0 Hz	Yanıp sönen ışık
Bit 4	1.25 Hz	
Bit 3	2.0 Hz	Hızlı yanıp sönen ışık
Bit 2	2.5 Hz	
Bit 1	5.0 Hz	
Bit 0	10 Hz	Titreyen ışık

O halde yukarıdaki alıştırma şöyle çözülebilir.  
(Önce CPU daki taktmerker aktif hale getirilir)

U E 124.0  
U M 20.5  
= A 124.0

**Örnek:22** Bir yaya geçidindeki trafik ışıkları gündüz diyagramda görüldüğü gibi gece ise 0.5 Hz lik taktla sarı lambanın yanıp sönmeleriyle çalışmaktadır.

Gece gündüz ayırımı bir anahtarla sağlanmaktadır.

Sayıcı kullanarak çözümü gerçekleştiriniz.

M0.2 M0.1 M0.0 Adım değiştirici çıkışları

SAYICI ÇIKIŞI			0	1	2	3	4	5	6	7	LAMBALAR					TAKT
M2	M1	M0									K	S	Y	K	Y	
0	0	0	1								X				X	1
0	0	1		1							X				X	2
0	1	0			1						X	X		X		3
0	1	1				1							X	X		4
1	0	0					1						X	X		5
1	0	1						1					X	X		6
1	1	0							1				X	X		7
1	1	1								1				X		8

#### ATAMA LİSTESİ

OPERAND	SEMBOL	AÇIKLAMA
E124.0	S0	Gece / Gündüz ayırma anahtarı
A125.0	H1	Otolar için kırmızı lamba
A125.1	H2	Otolar için sarı lamba
A125.2	H3	Otolar için yeşil lamba
A125.3	H4	Yayalar için kırmızı lamba
A125.4	H5	Yayalar için yeşil lamba
A124.0	A1	1.Adım
A124.0	A2	2.Adım
A124.1	A3	3.Adım
A124.2	A4	4.Adım
A124.3	A5	5.Adım
A124.4	A6	6.Adım
A124.5	A7	7.Adım
A124.6	A8	8.Adım

Burada adım değiştirici çıkışları sayıcının saydığı gözetlemek amacıyla çıkışlara atanmıştır. Normal durumda (Merker) yardımcı rölelere atanır. Çıkışları boşuna işgal etmemesi için.

**Çözüm**

```

UN E124.0 //TAKT OLUŞTURMA
U E124.0
UN T2
L S5T#1S
SV T1
UN T2
L S5T#1S
SI T2
//////////
U T2
ZV Z1
U(
ON E124.0
O M0.3
)
R Z1
L Z1
T MB0 //SAYICININ İÇERİĞİNİ MB 0' a
////////// ATAMA

```

**1. ADIM**

```

U E124.0
UN M0.0
UN M0.1
UN M0.2
=A 124.0

```

**2. ADIM**

```

U M0.0
UN M0.1
UN M0.2
=A 124.1

```

**3. ADIM**

```

UN M0.0
U M0.1
UN M0.2
=A 124.2

```

**4. ADIM**

```

U M0.0
U M0.1
UN M0.2
=A 124.3

```

**5. ADIM**

```

UN M0.0
UN M0.1
U M0.2
=A 124.4

```

**6. ADIM**

```

U M0.0
UN M0.1
U M0.2
=A 124.5

```

**7. ADIM**

```

UN M0.0
U M0.1
U M0.2
=A 124.6

```

**8. ADIM**

```

U M0.0
U M0.1
U M0.2
=A 124.7

```

**OTO KIRMIZI**

```

O A124.0
O A124.1
O A124.2
=A125.0

```

**OTO SARI**

```

O A124.2
O A124.7
O

```

**//GECE ÇALIŞMASI****İÇİN**

```

UN E124.0
U T1
=A125.1

```

**OTO YEŞİL**

```

O A124.3
O A124.4
O A124.5
O A124.6
=A125.2

```

**YAYA KIRMIZI**

```

O A124.2
O A124.3
O A124.4
O A124.5
O A124.6
O A124.7
=A125.3

```

**YAYA YEŞİL**

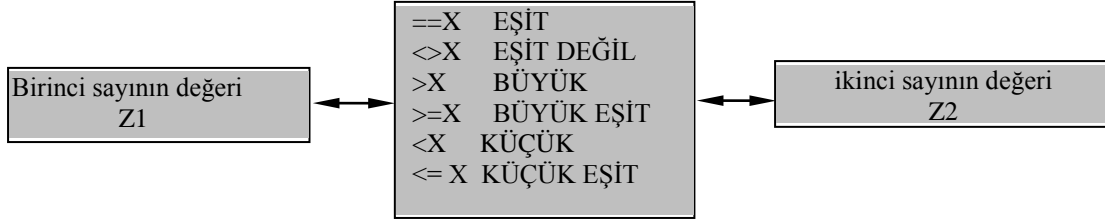
```

O A124.0
O A124.1
=A125.4
BE

```

## KARŞILAŞTIRMA OPERASYONLARI

Karşılaştırma operasyonlarının kullanılması yükleme ve transfer emirlerinin iyi anlaşılması durumunda gayet basittir. İlgili yazılım şeklinin kullanılmasıyla aşağıdaki gerçekleştirilebilir



Akü 2'nin içeriği Akü 1'in içeriği ile karşılaştırılır. Örneğin ">=D" Akü 2'nin içeriği büyük veya eşit mi Akü 1'e diye sorgulanır. Eğer Akü 2, Akü 1 den büyük ise lojik işlem sonucu ( VKE ) "1" olur

**DİKKAT:** Sadece aynı tip değişkenler karşılaştırılabilir

16 bit tam sayılar	→ x = I ( Kıyaslama Akü1-L ile Akü-2 L arasındadır.)
32 bit tam sayılar	→ x = D
Kayan virgüllü sayılar	→ x = R

	İşlem sırası	Z1 → (Akü 2)	+8	+8	+8		-8	-8	-8
		Z2 → (Akü 1)	+7	+8	+9		-7	-8	-9
L MW10	">=I" satırına geldiğinde "Akü 1" içinde MW12, "Akü 2" içinde MW10 vardır. Bu programa göre eğer MW10 içeriği, MW12 içeriğinden büyük veya eşit ise M 2.0 setlenecek, küçükse resetlenecek	==I	0	1	0		0	1	0
L MW12		<>I	1	0	1		1	0	1
>=I		>=I	1	1	0		0	1	1
S M2.0		>I	1	0	0		0	0	1
<I		<=I	0	1	1		1	1	0
R M2.0		<I	0	0	1		1	0	0

Karşılaştırma işlemleri sonucunda bilinmesi gereken önemli bir nokta ise, operasyon sonucunun bir bit ile değerlendirilmesidir.

**Örnek 1:** Giriş kanalından girilen sayı 10 dan büyük veya eşit, 20 den küçük veya eşit ise çıkış lamba yansın.

```
L EW124
L 10
>=I
= M10.0
//
L EW124
L 20
<=I
= M10.1
//
U M10.0
U M10.1
= A124.0
BE
```

Giriş kanalında okunan sayı tek tek istenen değerlerle kıyaslanmış ve kıyaslama sonuçları bir bit'e atanmıştır. Bu bitler değerlendirerek atama sağlanmıştır. Bu uygulamada kıyaslama sonuçları bir bit'e atanmadan da doğrudan değerlendirilebilir.

```
U(
L EW124
L 10
>=I
)
U(
L EW124
L 20
<=I
)
= A124.0
BE
```

**Örnek: 23** Giriş kanalından girilen sayı 10 dan büyük veya eşit, 20 den küçük veya eşit ise çıkış lamba yansın.

```
L EW124
L 10
>=I
= M10.0
//
L EW124
L 20
<=I
= M10.1
//
U M10.0
U M10.1
= A124.0
BE
```

Giriş kanalında okunan sayı tek tek istenen değerlerle kıyaslanmış ve kıyaslama sonuçları bir bit'e atanmıştır. Bu bitler değerlendirerek atama sağlanmıştır. Bu uygulamada kıyaslama sonuçları bir bit'e atanmadan da doğrudan değerlendirilebilir.

```
U(
L EW124
L 10
>=I
)
U(
L EW124
L 20
<=I
)
= A124.0
BE
```

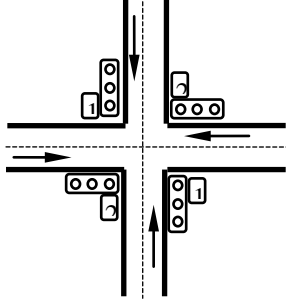
**Örnek :24** EB 124 ve EB125 giriş kanallarından bit örneklerinin birbirleri ile karşılaştırılması gerekmektedir.İkisi de aynı ise A 124.0 çıkışında “1” sinyali görülecektir.EB 124 girişindeki değer EB 125 dekinden büyük ise ,A 124.1 çıkışında “1” sinyali görülmesi gerekmektedir.

L EB124 L EB 125 ==I =A 124.0 <I =A 124.1	.	
--	---	--

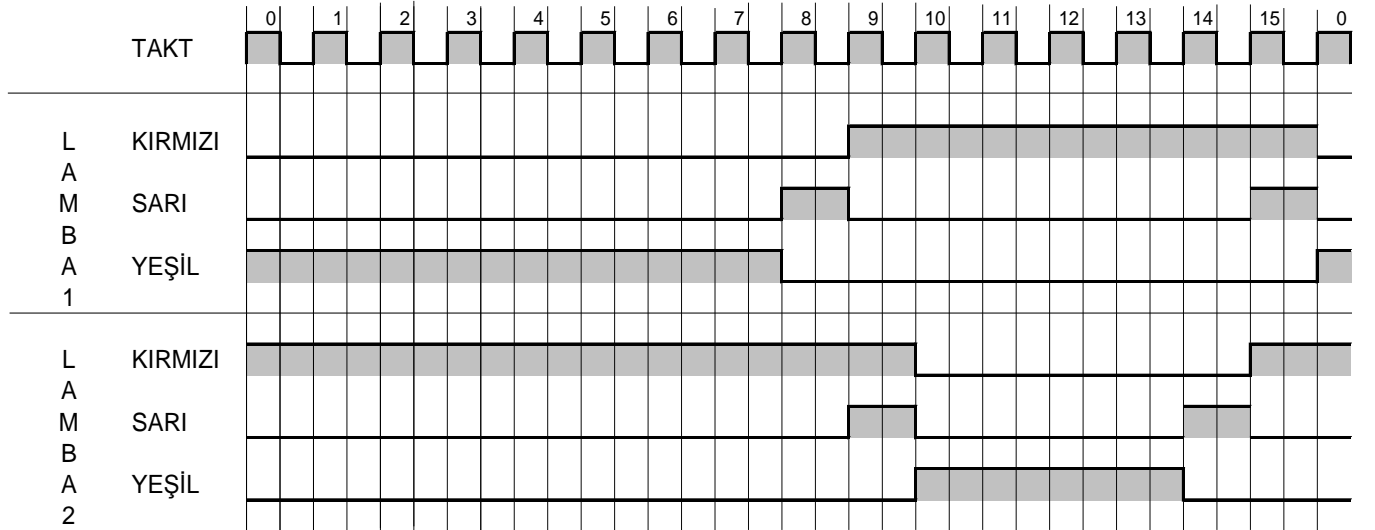
**Uygulama 25:** Trafik ışıkları

Bir kavşaktaki trafik ışıkları gündüz aşağıdaki diyagram gibi, gece ise; her iki sarı lamba 1Hz lik taktla yanıp sönecektir.

Darbelerle bir sayıcı ileriye saydırılacak ve sayıcı durumu kıyaslanarak lambaların ne zaman yanıp ne zaman söneceği belirlenecektir..

**Atama Listesi**

<u>Sembol</u>	<u>Adres</u>	<u>Açıklama</u>
SO	E 124.0	Gece/Gündüz
1L1	A 124.0	Lamba 1 Kırmızı
1L2	A 124.1	Lamba 1 Sarı
1L3	A 124.2	Lamba 1 Yeşil
2L1	A 125.0	Lamba 2 Kırmızı
2L2	A 125.1	Lamba 2 Sarı
2L3	A 125.2	Lamba 2 Yeşil



<pre>// Takt generatörü U( ON E 124.0 O E 124.0 ) UN T 2 L S5T#1S SV T 1 // UN T 1 L S5T#1S SI T 2  // Sayıcının programlanması  U T 2 ZV Z 1 U( ON E 124.0 O M 1.7 ) R Z 1  // Lamba 1 Kırmızı  L Z 1 L 9 &gt;=I = M 0.0  L Z 1 L 16 &lt;=I = M 0.1</pre>	<pre>// Lamba 1 Sarı  L Z 1 L 8 &gt;=I = M 0.2  L Z 1 L 9 &lt;I = M 0.3  L Z 1 L 15 &gt;=I = M 0.4  L Z 1 L 16 &lt;I = M 0.5  // Lamba 1 Yeşil  L Z 1 L 8 &lt;I = M 0.6  // Lamba 2 Kırmızı  L Z 1 L 10 &lt;I = M 0.7  L Z 1 L 15 &gt;=I = M 1.0</pre>	<pre>// Lamba 2 Sarı  L Z 1 L 9 &gt;=I = M 1.1  L Z 1 L 10 &lt;I = M 1.2  L Z 1 L 14 &gt;=I = M 1.3  L Z 1 L 15 &lt;I = M 1.4  // Lamba 2 Yeşil  L Z 1 L 10 &gt;=I = M 1.5  L Z 1 L 14 &lt;I = M 1.6  // Sayıcının resetlenmesi  L Z 1 L 16 ==I = M 1.7  // Lamba 1 Kırmızı  U M 0.0 U M 0.1 = A 124.0</pre>	<pre>// Lamba 1 Sarı  U M 0.2 U M 0.3 O U M 0.4 U M 0.5 O UN E 124.0 U T 2 = A 124.1  // Lamba 1 Yeşil  U M 0.6 U E 124.0 = A 124.2  // Lamba 2 Kırmızı  U M 0.7 U E 124.0 O U M 1.0 U E 124.0 = A 124.3  // Lamba 2 Sarı  U M 1.1 U M 1.2 O U M 1.3 U M 1.4 O UN E 124.0 U T 2 = A 124.4  // Lamba 2 Yeşil  U M 1.5 U M 1.6 = A 124.5 BE</pre>
--	--	--	---

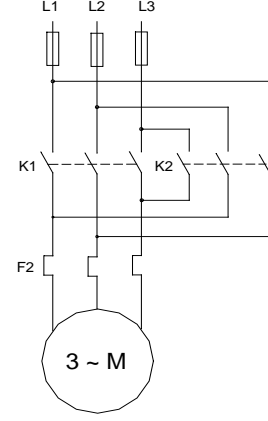
**Örnek:26** ASENKRON MOTORUN İLERİ ve GERİ YÖNDE ÇALIŞTIRILMASI

3 fazlı bir asenkron motor "S1" butonu ile ileri yönde "S2" butonu ile geri yönde çalıştırılacaktır. "S0" butonuna basılınca da duracaktır. Bir yönde çalışırken motor durdurulmadan diğer yönde çalışmayacaktır. Ayrıca motor "F2" aşırı akım rölesi ile korunacaktır.

Çözümü (S7 300 PLC ile) yapısal programlama ile yapınız.

Atama Listesi		
Operand	Sembol	Açıklama
E 124.0	S0	Stop anahtarı (NK)
E 124.1	S1	İleri çalıştırma butonu(NA)
E 124.2	S2	Geri çalıştırma Butonu(NA)
E 124.7	F2	Aşırı akım rölesi(NK)
A 124.0	K1	İleri yön rölesi
A 124.1	K2	Geri yön rölesi

NA :Normalde açık  
NK :Normalde kapalı



PLC programı	FC 1
OB1 'in Programlanması	.....
U E 124 . 1 Motorun ileri yönde çalışması	U E 124 . 1 Motorun ileri yönde çalışması
UN A 124 . 1	UN A 124 . 1
CC FC1	S M 0 . 0
U E 124 . 2 Motorun geri yönde çalışması	U (
UN A 124 . 0	ON E 124 . 0
CC FC2	ON E 124 . 7
BE	)
	R M 0 . 0
	U M 0 . 0
	= A 124 . 0
	BE
Açıklama:	FC2
Eğer,	...
(U E 124 . 1) ve (UN A 124 . 1) varsa, FC 1 fonksiyon bloğuna git,motoru ileri yönde çalıştır ve program bitiminde OB1'e dön.	U E 124 . 2 Motorun geri yönde çalışması
Eğer,	UN A 124 . 0
(U E 124 . 2) ve (UN A 124 . 0) varsa, FC 2 fonksiyon bloğuna git,motoru geri yönde çalıştır ve program bitiminde OB1'e dön.	S M 0 . 1
	U (
	ON E 124 . 0
	ON E 124 . 7
	)
	R M 0 . 1
	U M 0 . 1
	= A 124 . 1
	BE
CC: Şartlı çağırma komutu	
CALL: Şartsız çağırma komutu	

**STATUS WORD**

Genellikle PLC ile yapılan bir işlemin somut sonucu değil, belli bir değer aşılıp aşılmadığı, sonucun negatif veya pozitif mi olduğu, bir toplama veya çarpma işleminde artanın olup olmadığı önemlidir.

15	9	8	7	6	5	4	3	2	1	0	
"0"			BIE	A1	A0	OV	OS	OR	STA	VKE	/ER

Statusword'ün yapısı

Bu iş için CPU özel bir 16 bit Register' e sahiptir. Sadece 9 bit' i kullanılan bu registerin içeriğine STATUS WORD denir. Bu bitlerin anlamları aşağıdaki gibidir.

Bit	İşaret	Fonksiyon / açıklama
0	/ER	Lojik işlem başlangıcında '1', Lojik işlemin sonucu bir yere atandığında '0' olur. ('VE', 'VEYA' gibi lojik işlemlerde '1', '=', 'SET', 'RESET' yada sıçrama emirlerinde '0' olur.)
1	VKE	Lojik veya karşılaştırma işlemlerinin sonuçlarını kaydeder. VKE'yi set, reset, negasyon yapabiliriz. VKE ile zaman, sayıcı, şartlı sıçrama gibi operasyonları yaptırabiliriz.
2	STA	Adreslenmiş bir BİT değerini kaydeder. İşlemler tarafından sorgulanmaz, sadece program değişkenlerinin durumunu göstermeye yarar.
3	OR	VEYA' dan önce VE kapısı oluşumunu gösterir.
4	OV	Aritmetik veya virgüllü sayı karşılaştırma işlemlerinde hata oluştuğunda "1" olur. Hata ortadan kalktığında tekrar "0" olur. Sorgulama ve şartlı sıçrama emri olarak kullanılabilir. ( U OV, SPO gibi )
5	OS	"OV = 1" olduğunda "OS" setlenir, "OV = 0" olduğunda resetlenmez. Hata oluşan blokun bitmesi veya "PLC" emrinin işletilmesi gerekir. Sorgulama, şartlı sıçrama, modül çağırma ve bitirme emri olarak kullanılabilir. ( U OS, PLC ... gibi )
6	A0	Aritmetik işlemlerin sonuçları
7	A1	Aritmetik işlemlerin sonuçları(Aşağı tablolardaki gibi)
8	BIE	Word kapı bağlantı işleminin sonucunu dual olarak, bir başka dual kapı bağlantısı ile işlemeyi mümkün kılar.

**Bit operasyonunda Durum bitlerinin etkileşimi**

Emir	/ER	VKE	STA	OR	AÇIKLAMA
= M 10.0	0	X	X		
U E124.0	1	1	1	0	E124.0 = 1
UN E124.1	1	1	0	0	E124.1 = 0
O	1	1	1	1	
O E124.2	1	1	0	0	E124.2 = 0
ON E124.3	1	1	1	0	E124.3 = 1
= A124.0	0	1	1	0	A124.0 = 1
R A124.1	0	1	0	0	A124.1 = 0
S A124.2	0	1	1	0	A124.2 = 1
U E 125.0	1	X	X		

**Dijital operasyonlarda Durum bitlerinin etkileşimi**

Emir	A0	A1	OV	OS	AÇIKLAMA
T MW10	X	X	X	X	
L +12	X	X	X	X	
L +15	X	X	X	X	
-I	1	0	0	0	Sonuç negatif
L +20000	1	0	0	0	
*I	0	1	1	1	Taşma var ( "OV" ve "OS" =1 )
L +20	0	1	1	1	
+I	0	1	0	1	"OV" = 0 oldu, "OS" = 1 kaldı
T MW30	0	1	0	1	
L MW30	1	1	0	1	

**Aritmetik operasyonlarda "A1" ve "A0" bitlerinin durumu**

Taşma olmayan tam sayı operasyonlarında		
A1	A0	Açıklama
0	0	Sonuç = 0
0	1	Sonuç < 0
1	0	Sonuç > 0
Taşma olan tam sayı operasyonlarında		
A1	A0	Açıklama
0	0	+I ve +D operasyonları ve taşma negatif alanda
0	1	*I ve *D operasyonları ve taşma negatif alanda, +I, -I, +D, -D, NEGI, NEGD operasyonları ve taşma pozitif alanda
1	0	*I, *D, /I, /D operasyonları ve taşma pozitif alanda, +I, -I, +D ve -D operasyonları ve taşma negatif alanda,
1	1	"0" ile bölmede /I, /D
Taşma olan virgüllü sayı operasyonlarında		
A1	A0	Açıklama
0	0	Devreden sayılı bölme ( .3333... )
0	1	Taşma negatif alanda
1	0	Taşma pozitif alanda
1	1	Geçersiz virgüllü sayı operasyonu

**Karşılaştırma operasyonlarında "A1" ve "A0" bitlerinin durumu**

A1	A0	Açıklama
0	0	Akü 2 = Akü 1
0	1	Akü 2 < Akü 1
1	0	Akü 2 > Akü 1
1	1	Geçersiz operasyon

**Kayıdırma ve döndürme operasyonlarında "A1" ve "A0" bitlerinin durumu**

A1	A0	Açıklama
0	0	Kayan bit = 0
1	0	Kayan bit = 1

**Word lojik operasyonlarında "A1" ve "A0" bitlerinin durumu**

A1	A0	Açıklama
0	0	Sonuç = 0
1	0	Sonuç <> 0

9 bit'i kullanılan status word'ü de diğer veriler gibi Akü'ye yüklemek veya belli bir değeri status word'a transfer etmek mümkündür. ( L STW, T STW ) Online durumunda status word'un içeriği de izlenebilir.

**Not:** S7-300 CPU larda /ER, STA, OR bitleri yüklenmez, yükleme sonrasında sürekli "0" değerini verirler

<p><b>SET:</b> "VKE" yi doğrudan "1" yapar. "SET" emrinden sonra gelen emir ilk sorgulama emri olur. Sadece "AWL" de kullanılabilir.</p> <p><b>CLR:</b> "VKE" yi doğrudan "0" yapar. "CLR" emrinden sonra gelen emir ilk sorgulama emri olur. Sadece "AWL" de kullanılabilir.</p> <p><b>NOT:</b> "VKE" yi ters çevirir."1" ise "0", "0" ise "1" yapar.</p> <p><b>SAVE:</b> "VKE" yi "BIE" ye kopyalar.</p>	SET	1	"VKE" yi "1" yaptı
	= M 0.1	1	"VKE" "1" olduğu için atama "1" oldu
	CLR	0	"VKE" yi "0" yaptı
	= M 0.2	0	"VKE" "0" olduğu için atama "0" oldu
	CLR	0	"VKE" yi "0" yaptı
	U E 124.1	1	Önceki satırları değerlendirmes ilk emir İşlemi
	= M 0.3	1	görür, "1" ise çıkış"1", "0" ise çıkış 0"0 olur.
	SAVE	1	"E124.0" "1" ise "VKE" "1" olur ve bunu "BIE"ye kaydeder
	CLR	0	
	"		
"			
U BIE	1	Kaydedilen "BIE"nin durumunu "VKE" ye atar.	
= M 0.4	1	"BIE"nin durumuna göre atama yapılır.	
NOT	0	"VKE" "1" idi onu ters çevirir yani "0" yapar.	
= M 0.5	0		

## SIÇRAMA OPERASYONLARI

Sıçrama operasyonları, başka bir noktadaki işlemi yapabilmek üzere, fonksiyon veya fonksiyon modülü içerisindeki doğrusal akışı kesmeye yararlar. Bir sıçrama işleminin Operand'ı daima bir adres veya hedeftir ( hedef = sembolik sıçrama adresi ).

Hedef maksimum 4 işaretten oluşur ve ilk işaret bir harf olmalıdır ( Türkçe karakterler olmamalıdır). Sıçrama adresindeki iki nokta üst üste ( : ) işareti hedeflenen adresi tanımlar.

	SP	...	.....
	(Sıçra)	(şart)	(hedef)
... :	...		
(Hedef)	(emir)		

### Şartsız sıçramalar ( SPA ):

Lineer program işlenmesi sırasında "**SPA**" emri görüldüğünde hiçbir şarta bakılmaksızın belirlenen hedefe sıçrar.

Sıçrama hedefi ileri veya geriye olabilir, maksimum sıçrama kapasitesi; ileriye "32767", geriye "32768" satırdır.

```

U   E124.0
U   E124.1
SPB Ali
L   MB 10
INC 1
T   MB 10
SPA Veli
Ali:L   0
T   MB 10
Veli:U  E125.0

```

"E 124.0" ve "1" in lojik işlem sonucu "1" ise, program "Ali" satırına sıçrayıp emirleri oradan itibaren işlemeye devam edecektir. Eğer işlem sonucu "1" değil ise, "SPA Veli" satırına kadar emirler işlenip orada hiçbir şarta bakmaksızın 'Veli' satırına sıçrayacaktır.

Ayrıca S7 sistemlerinde programlanmış çoklu sıçrama emirleri kullanılabilir. Bunun için "Akü 1 LL" Bayt'ında bulunan sayı ile belirlenmiş hedefe sıçranabilir. Maksimum 256 sıçrama hedefi bildirilebilir.

L EB 124	
SPL Ma	( Akü 1LL) > 3 <sup>1</sup> ise, sıçrama hedefi 'Ma <sup>2</sup> ' dir
SPA Ma_1	( Akü 1LL) = 0 ise, sıçrama hedefi 'Ma_0' dir
SPA Ma_2	( Akü 1LL) = 1 ise, sıçrama hedefi 'Ma_1' dir
SPA Next	( Akü 1LL) = 2 ise, sıçrama hedefi 'Next' dir
SPA Ma_3	( Akü 1LL) = 3 ise, sıçrama hedefi 'Ma_3' tür
Ma:SPA Next	
Ma_1: Emir1	
Emir2	
SPA Next	
Ma_2: Emir1	
Emir2	
SPA Next	
Ma_3: Emir1	
Emir2	
SPA Next	
Next: Emir1	
Emir2	

- Aynı hedefe birçok sıçrama emri verilebilir.
- Hedefler bir sıra takip etmek zorunda değildir.

<sup>1</sup> "SPL" emrinden sonra gelen "SPA" sayısı

<sup>2</sup> "Bu hedef her zaman "SPA" lardan sonra gelen ilk sıçrama hedefi olmalıdır.

**Örnek 27:**Giriş EB 124 den girilen değerler ile şu işlemler yapılacaktır.

EB 124-----→0 ( E 125.0 dan 1 sinyali verilirse A 125.0 çalışsın)  
 EB 124-----→1 ( E 125.1 den 1 sinyali verilirse ve  
 ( E 125.2 den 1 sinyali verilirse A 125.1 çalışsın)  
 EB 124-----→2 ( E 125.3 den 1 sinyali verilirse A 125.2 1 sn. aralıklarla yanıp sönsün)  
 EB 124-----→3 ( E 125.4 den 1 sinyali verilirse veya  
 ( E 125.5 den 1 sinyali verilirse A 125.3 çalışsın)  
 EB 124-----→4 ( E 125.7 den 1 sinyali verilirse A 125.7 çalışsın)

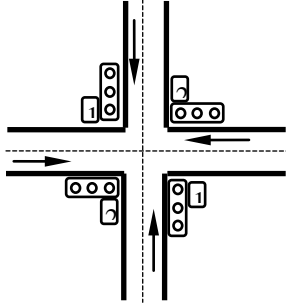
**Çözüm:**

L EB 124  
 SPL M0  
 SPA M01  
 SPA M02  
 SPA M03  
 SPA M04  
 M0: BEA  
 M01:U E125.0  
 =A 125.0  
 BEA  
 M02:U E125.1  
 U E125.2  
 =A 125.1  
 BEA  
 M03:U E125.3  
 UN T2  
 L S5T#1S  
 SV T1  
 UN T1  
 L S5T#1S  
 SI T2  
 U T1  
 =A 125.2  
 BEA  
 M04:O E125.4  
 O E125.5  
 =A 125.3  
 BEA  
 M05:U E125.7  
 =A 125.7  
 BE

**Örnek 28: Trafik ışıkları**

Bir kavşaktaki trafik ışıkları gündüz aşağıdaki diyagram gibi, gece ise; her iki sarı lamba 0.5 Hz lik taktla yanıp sönecektir.

İntern darbelerle bir sayıcı ileriye saydırılacak ve sayıcı durumu kıyaslanarak lambaların ne zaman yanıp ne zaman söneceği belirlenecektir..



2L3

A 125.2

Lamba 2 Yeşil

**Atama Listesi**

Sembol	Adres	Açıklama	Gece/Gündüz
SO	E 124.0		Gece/Gündüz
1L1	A 124.0	Lamba 1 Kırmızı	
1L2	A 124.1	Lamba 1 Sarı	
1L3	A 124.2	Lamba 1 Yeşil	
2L1	A 125.0	Lamba 2 Kırmızı	
2L2	A 125.1	Lamba 2 Sarı	

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	
	TAKT	■		■		■		■		■		■		■		■		■	
L	KIRMIZI										■	■	■	■	■	■	■	■	■
A																			
M	SARI									■									■
B																			
A	YEŞİL	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
1																			
L	KIRMIZI	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
A																			
M	SARI										■								■
B																			
A	YEŞİL											■	■	■	■	■	■	■	■
2																			

**Çözüm 1:**

Bir önceki uygulamada olduğu gibi intern darbelerle ileri sayan bir sayıcı oluşturulacak ve içeriği Akü' ye yüklenecektir. Akü içeriğinin durumuna yani sayıcının durumuna bağlı olarak çoklu sıçrama emri ile program yazılacaktır.

U E 124.0  
SPB GECE

U M 20.5  
ZV Z 1  
L Z 1  
L 15  
>I  
R Z 1  
L Z 1  
T AB 125

SPL M0 //SAYICI > 16  
SPA M1 //SAYICI 1 --> 8  
SPA M1 “  
SPA M1 “  
SPA M1 “  
SPA M1 “  
SPA M1 “  
SPA M1 “  
SPA M1 “  
SPA M2 //SAYICI 8 --> 9  
SPA M3 //SAYICI 9 --> 10  
SPA M4 //SAYICI 10 --> 14  
SPA M4 “  
SPA M4 “  
SPA M4 “  
SPA M5 //SAYICI 14 --> 15  
SPA M6 //SAYICI 15 --> 16

M0: SPA SON //SAYICI >16  
M1: SET //SAYICI 1 --> 8

S A 124.2  
S A 124.5  
R A 124.0  
R A 124.1  
R A 124.6  
R A 124.7  
SPA SON

M2: SET //SAYICI 8 --> 9  
S A 124.1  
R A 124.2  
SPA SON

M3: SET //SAYICI 9 --> 10  
S A 124.0  
S A 124.6  
R A 124.1  
SPA SON

M4: SET //SAYICI 10 --> 14  
S A 124.7  
R A 124.5  
R A 124.6  
SPA SON

M5: SET //SAYICI 14 --> 15  
S A 124.6  
R A 124.7  
SPA SON

M6: SET //SAYICI 15 --> 16  
S A 124.1  
S A 124.5  
R A 124.6

SON: BEA

GECE: L 0

T AW 124  
U M 20.5  
R Z 1  
= A 124.1  
= A 124.6  
BE

## Çözüm 2:

Bir önceki uygulamada olduğu gibi intern darbelerle ileri sayan bir sayıcı oluşturulacak ve içeriği Akü' ye yüklenecektir. Akü içeriğinin durumuna yani sayıcının durumuna bağlı olarak çoklu sıçrama emri ile program yazılacaktır.

U E 124.0  
SPB GECE

U M 20.5

ZV Z 1

L Z 1

L 15

>I

R Z 1

L Z 1

T AB 125

SPL M0 //SAYICI > 16

SPA M1 //SAYICI 1 --> 8

SPA M1 “

SPA M1 “

SPA M1 “

SPA M1 “

SPA M1 “

SPA M1 “

SPA M1 “

SPA M2 //SAYICI 8 --> 9

SPA M3 //SAYICI 9 --> 10

SPA M4 //SAYICI 10 --> 14

SPA M4 “

SPA M4 “

SPA M4 “

SPA M5 //SAYICI 14 --> 15

SPA M6 //SAYICI 15 --> 16

M0: SPA SON //SAYICI >16

M1:L W#16#2400

T AW124 // 1L Yeşil 2L Kırmızı

BEA

M2:L W#16#2400

T AW124 //1L Sarı 2L Kırmızı

BEA

M3:L W#16#2200

T AW124 //1LSarı 2L Kırmızı

2L Sarı

BEA

M4:L W#16#6100

T AW124 // 2L Yeşil 1L Kırmızı

BEA

M5:L W#16#8100

T AW124 // 2L Sarı 1L Kırmızı

BEA

M6:L W#16#4100

T AW124

BEA

SON: BEA

GECE: L 0

T AW 124

U M 20.5

R Z 1

= A 124.1

= A 124.6

BE

**Şartlı sıçramalar:**

Sıçrama işlemi sorgulanabilecek bir şarta bağlı olarak gerçekleştirilir. Bu şartlar; Mantık işleminin sonucuna, aritmetik veya karşılaştırma işleminin sonucuna, Status Wort'taki BIE, OV, OS bitlerinin durumuna bağlıdır.

VKE=1 ise sıçra	SPB y
VKE=0 ise sıçra	SPBN y
VKE=1 ise sıçra ve VKE'i BIE'ye kurtar	SPBNB y
VKE=0 ise sıçra ve VKE'i BIE'ye kurtar	SPBB y
BIE=1 ise sıçra	SPBI y
BIE=0 ise sıçra	SPBIN y
OV=1 ise sıçra	SPO y
OS=1 ise sıçra	PLC y
Sonuç=0 ise sıçra	SPZ y
Sonuç<>0 ise sıçra	SPN y
Sonuç<0 ise sıçra	SPP y
Sonuç>0 ise sıçra	SPM y
Sonuç<=0 ise sıçra	SPMZ y
Sonuç>=0 ise sıçra	SPPZ y
Sonuç geçerli değil ise sıçra	SPU y

Sıçramalar sadece bir program modülü içerisinde kullanılabilirler. Başka bir program modülündeki adrese gitmek mümkün değildir.

Sıçrama hem ileri hem de geriye doğru olabilirler. Maksimum sıçrama genişliği, program kodunun -32768 ve +32767 Word işareti kadardır. Bu sayılar emir sayıları ile aynı değildir, çünkü bunlar emirden emire bir, iki veya üç Wort' luk deyimler olabilirler.

Özellikle geriye doğru yapılan sıçrama operasyonlarında çevrim süresinin aşılmasına (sonsuz döngüye girilmemesine) dikkat edilmelidir.

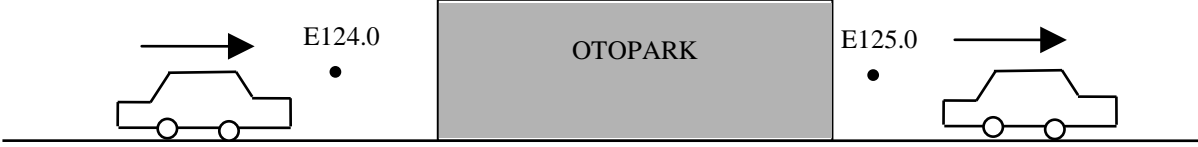
U E 124.0  
SPB atla  
S A 124.0  
R A 124.1  
BEA  
atla:S A 124.1  
R A 124.0  
BE

**Özel durum:** Yandaki programda E 124.0 girişi "0" olduğunda "VKE" değeri "0" olacağı için "atla" hedefine sıçrayamayacaktır. Bu durumda yine "VKE" nin "0" olmasından dolayı "S A124.0" emrinin işlememesi gerekir. Ancak bu S7 lerde özel bir durumdur. "SPB, SPBB, SPBN" gibi emirler "VKE" yi "1" yapar. Normal şartlarda E 124.0 girişinin "0" olduğunda setlenmemesi gereken "A 124.0" değeri setlenmektedir.

**Örnek 29: Otopark**

100 araba kapasiteli bir otoparkın doluluk durumu izlenmek istenmektedir. Otoparkın giriş ve çıkışına birer sensör konarak giren ve çıkan arabalar sayılmaktadır.

Otoparkın doluluk durumu; boş, boş değil, yarı dolu, %90 dolu, tam dolu olduğu ışıklı uyarılarla izlenebilmelidir.



BOŞ ⊗	BOŞ DEĞİL ⊗	YARI DOLU ⊗	%90 DOLU ⊗	DOLU ⊗
A124.0	A124.1	A124.2	A124.3	A124.4

Program çok farklı yöntemlerle yapılabileceği gibi biz konumuz gereği kıyaslama ve sıçrama operasyonları ile çözelim.

Bir sayıcı programlanacak, giren arabalar sayıcıyı ileri, çıkan arabalar sayıcıyı geri saydıracaktır. Yine sayıcı durumu kıyaslanacak ve yanması gereken lamba programına sıçranarak çıkış ataması yapılacaktır.

Bu programda program bitirme emirlerini de dikkatli kullanmak gerekir.

**Çözüm:****NW1: Çıkış kanalının boşaltılması**

L 0  
T AW 124

**NW2: Sayıcının programlanması**

U E 124.0  
ZV Z 1  
U E 125.0  
ZR Z 1  
U E 124.7  
R Z 1

**NW3: “OTOPARK BOŞ” kıyaslama**

L Z 1  
L 0  
==I  
= A 124.0  
BEB

**NW3: “OTOPARK BOŞ DEĞİL” kıyaslama**

L Z 1  
L 0  
<>I  
= A 124.1

**NW3: “OTOPARK YARI DOLU” kıyaslama**

L Z 1  
L 5  
<=I  
SPB yari

**NW3: “OTOPARK % 90” kıyaslama**

L Z 1  
L 9  
<=I  
SPB dolu

**NW3: “OTOPARK TAM DOLU” kıyaslama**

L Z 1  
L 10  
>=I  
SPB tam

yari: S A 124.2  
BEA

dolu: S A 124.3  
BEA

tam: S A 124.4  
BE

## B. DURUM GRAFİĞİ

Durum grafiği SPS programlamada bir çözüm yöntemidir. İçerisinde bellek, zaman, sayma vb. çeşitli fonksiyonlar içeren ve prosesten kaynaklanan belirli bir ardışık işlemin söz konusu olmadığı kumandalarda çözümün bulunması oldukça zordur. Ayrıca kişinin kendine ait yöntemler kullanmasının da sakıncaları vardır. Bu sakıncalar:

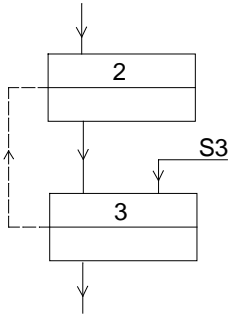
- Programın hata içerme oranı daha yüksektir.
- Arıza arama zordur.
- Dokümantasyon yeterince açıklayıcı, anlamayı kolaylaştırıcı değildir.

Çalışmanın bu bölümünde amaç programın yazılabilmesi için yardımcı bir taslağın oluşturulmasıdır. Bu yardımcı taslağın esası da proseste çalışma sırasında meydana gelebilecek durumların bulunmasıdır. Ortaya çıkabilecek durumlar bir durum grafiği olarak düzenlenirler. Bu durum grafiğinden gerekli program yazılmaktadır. Eğer meydana gelecek durumlar kesin olarak biliniyorsa, bu çeşit bir kumanda ardışık kumanda adını alır.

Bu şekilde yapılmış programlar genelde kişiye ait yöntemlerle yazılmış programlara göre daha uzundur. Fakat yukarıda değinilmiş olan problemler yoktur veya daha azdır.

Daha önce incelediğimiz konularda çözümü bulabilmek için bir gerçeklik tablosu düzenlemiştik. Çıkış sinyalleri direkt olarak giriş sinyallerinin o anki durumlarına bağlı idi. Aynı düşünce bu bölümde de uygulanabilir. Fakat dikkat edilmesi gereken nokta bu sefer çıkış sinyallerinin giriş sinyallerinin o andaki durumuna ve daha önce var olan sinyallere bağlı olacaktır. Yani hafızaya alınmış sinyaller de göz önüne alınacaktır.

Sonuç olarak yapacağımız taslak ile herhangi bir zamanda prosesin hangi konumda olacağını saptamalıyız. Bunun için de hangi şartlar (sinyaller) altında hangi konuma gireceğimizi ve hangi konumdan çıkacağımızı belirlememiz gerekir.

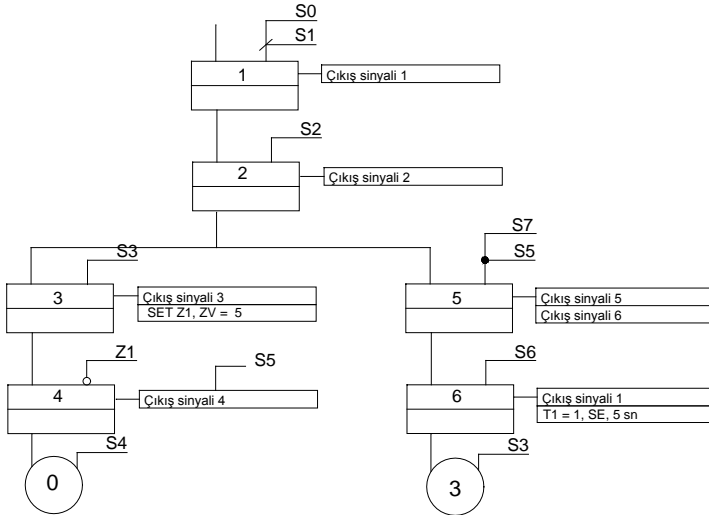
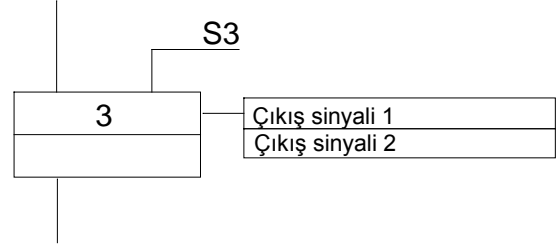


Durum grafiği içerisinde çeşitli durumlar dikdörtgen şeklindeki kutularla gösterilmektedir. Kutu içerisinde yatay bir çizgiyle oluşturulmuş iki bölüm vardır. Yukarıdaki bölüme durum numarası yazılır. Alttaki bölümde ise açıklayıcı kısa bir yazı bulunabilir. Etki çizgileri önceki ve sonraki durumlara gidişi gösterir.

Önceki durumlara gidişi göstermek amacıyla kullanılan etki çizgisi ve gidiş yönünü gösteren oklar normal şartlarda gösterilmez.

Üç numaralı duruma ulaşılabilmesi için "2" numaralı durumun aktif olması ve "B3" sinyalinin gelmesi gerekmektedir. "3" numaralı duruma gelmemizi sağlayan iki çizgiye de etki çizgisi adı verilir. Bunlar "VE" fonksiyonuna sahiptirler.

Yani ikisinin de "1" sinyali vermeleri durumunda "3" numaralı duruma ulaşılır. Her duruma ulaşıldığında belirli sonuç sinyalleri oluşturulabilir. Bu çıkış sinyalleri, çıkış değişkenleri, durum tespit işareti, zaman veya sayma fonksiyonları olabilir. Bunlar durum sembolünün hemen yanına bir etki çizgisiyle dikdörtgen olarak eklenirler. Eğer bu emirler birden fazla ise, dikdörtgenler alt alta olmak üzere çoğaltılabilirler. Ayrıca emirlerin yerine getirilmesi için başka şartlar varsa, bunlar dikdörtgenlerin üzerine eklenebilirler.



Durum grafiklerinde çeşitli ayrımlar olabilir. Yani proses birkaç kola ayrılabilir. Aşağıda görüldüğü gibi 2 numaralı durumdan sonra süreç ikiye ayrılmaktadır. Bu şekildeki ayrımların bulunduğu problemlerde durum grafiği açısından dikkat edilmesi gereken önemli bir sorun yoktur. Fakat durum grafiğinin bir programa çevrilmesinde kilitlemeler kullanılmalıdır. Çünkü iki ayrı duruma da girmemizi sağlayan sinyaller aynı anda gelebilirler. Böylece normalde bir süreç sadece bir durumda bulunabilecekken, aynı anda iki durum gerçekleşmiş olur.

Eğer durum grafiği içerisinde daha önce var olan bir duruma yeniden girilecekse ( bir döngü olayı var ise ), bunun yeniden bir dikdörtgen ile gösterilmesi yanılgılara neden olabilir. Böyle bir durum daire ile gösterilmelidir.

Bu şekilde yapılacak bir çözümde herhangi bir anda sadece tek bir durum aktif olabilir. Bu elde ettiğimiz taslağın programa dönüştürülmesinde kullanılacak SPS' nin operand haznesine göre değişik yöntemler kullanılabilir. Bu yöntemlerden bir tanesi her durumu bir bellek elemanına (RS-elemanı) atamaktır. Durum sayısı kadar bellek elemanı kullanılacaktır.

Kullanılacak durum tespit işareti durum sembolü içersine de yazılabilir.

Aşağıdaki 3 numaralı durum için durum grafiğinin programa dönüştürülmesi

DURUM GRAFİĞİ	"FUP" PROGRAMI	"AWL" PROGRAMI
		<pre> U M2 U E1 U E2 S M3 U M4 R M3 </pre>

SET girişinde çözümden görüldüğü gibi bir önceki adım ve iki etki çizgisi "VE" olarak bağlanmıştır. Bu üç şart gerçekleştiğinde "3." adım SET yapılacaktır. Bir önceki adım da "3." adımın SET yapılmasından sonraki çevrimde RESET yapılacaktır.

	<pre> O M1 O M2 O M4 = A1 </pre>	<p>Bir durum grafiğinin SPS programına çevriminde önce RS elemanlar kullanılır. Daha sonra da emirler bu RS elemanlarıyla dışarıya verilecektir. Örneğin "A1" çıkış değişkeninin 1,2 ve "4." adımlarda "1" değerini alması gerekiyorsa, bir VEYA kapısıyla çıkış ataması yapılacaktır.</p>
--	----------------------------------	--

Durum grafikleri ile program oluştururken önemli bir adım "0. Durum"un programlanmasıdır.

SPS'nin ilk açılmasıyla birlikte şartsız olarak sıfırınca durumun SET yapılması gerekir. Duruma göre de diğer bütün durumların RESET yapılması gerekebilir. Bu bir başlangıç darbesi ile yapılabilir. Bu başlangıç darbesi bazı SPS üreticisi firmalarca software olarak hazırlanmış durumdadır. Eğer böyle bir darbe yok ise de bu kolayca gerçekleştirilebilir.

```

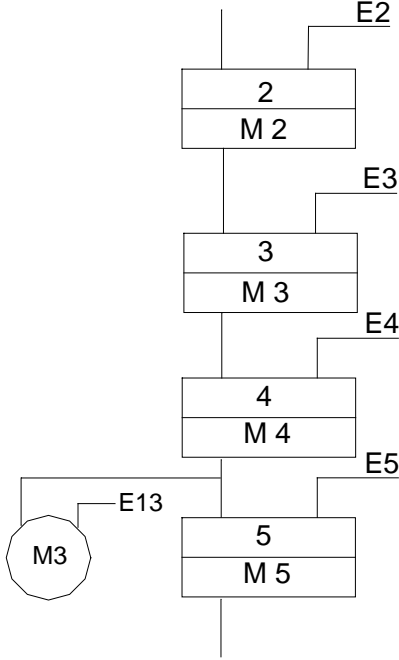
UN Mx
= My      x,y Durum tespit işareti numaraları
S Mx

```

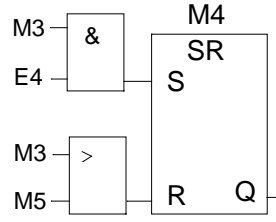
Yukarıdaki kısa programı incelersek, ilk çevrimde "My", "1" değerini alır ve hemen arkasında "Mx" "SET" yapılır. İkinci çevrimde "Mx", "SET" yapıldığı için "My", "0" değerini alır. Bu SPS tekrar kapatılıp açılana kadar böyle kalır. Tabii ki böyle bir program yazılımında mutlaka remanent (kalıcı) olmayan durum tespit işareti kullanılmalıdır. Tekrar hatırlayacak olursak, remanent durum tesbit işaretleri SET yapıldıktan sonra SPS tekrar kapatılıp açılabilir bir pil sayesinde sinyal durumlarını korurlar.

## DURUM GRAFİĞİ İÇERİSİNDE DÖNGÜLER

Durum grafiğinin durum tespit işaretleri kullanılarak programa dönüştürülmesinde arka arkaya bulunan iki durum arasında oluşturulacak döngülerde problemler ortaya çıkar.

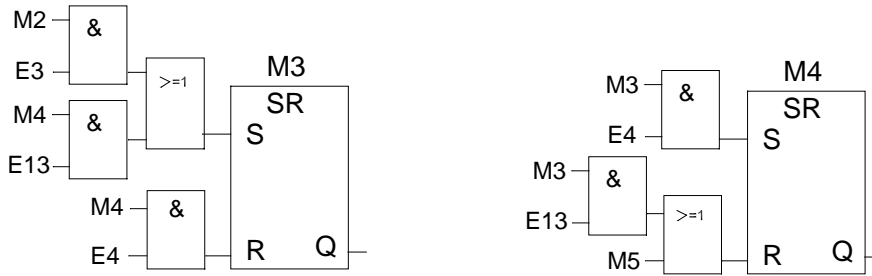


"3" ve "4" numaralı durumlar gerekli sinyallerin gelmesiyle sürekli olarak SET ve RESET yapılabilirler. "3" numaralı durum "4" numaralı durumun SET yapılması için hazırlık niteliği taşımakta ve diğer taraftan "4" numaralı durumu RESET yapmaktadır. Yani "M3", "M4"ün aynı anda hem SET hem de RESET bölümünde bulunmaktadır. Bu aynı şekilde "M3" için de geçerlidir.

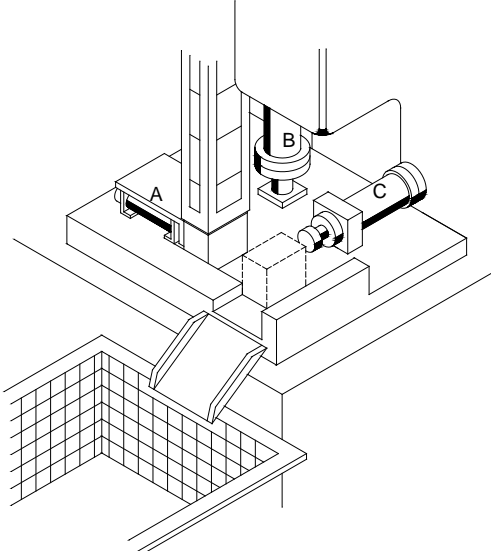


Yukarıdaki gibi yapılacak bir çözüm kendi kendini kilitlediği için doğru değildir. "M4" çok kısa bir süre SET, daha sonra da RESET yapılmaktadır.

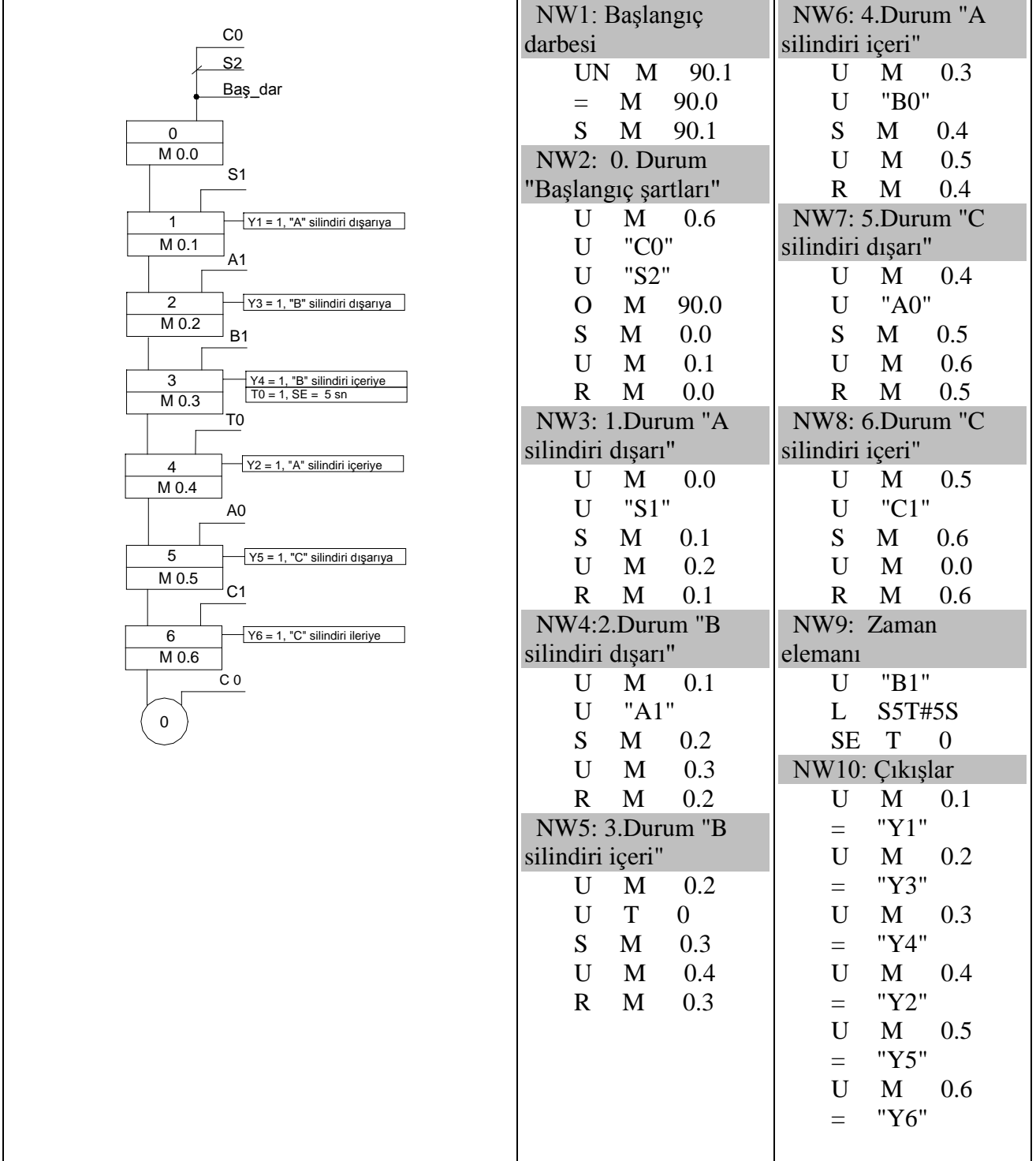
Çözümün doğru olarak gerçekleştirilmesi için SET ve RESET yaparken ilgili şartların hepsinin sorulması gerekir.



**Örnek:** Şekildeki damgalama düzeneğinde; Magazinde iş parçası var ise, "A" silindiri iş parçalarını magazinden alıp sıkıştırır. "B" silindiri damgalama işini yerine getirir. Damgalama için "B" silindirinin 5 sn aşağıda beklemesi gerekir. "B" silindiri içeri girdikten sonra "A" silindiri parçayı serbest bırakır ve "C" silindiri damgalanmış iş parçasını sepete atar.



SEMBOL	ADRES	AÇIKLAMA
S0	E125.0	Start
S1	E125.1	Stop
S2	E125.2	Magazin sorgulama
A0	E124.0	A silindiri içerde
A1	E124.1	A silindiri dışarıda
B0	E124.2	B silindiri içerde
B1	E124.3	B silindiri dışarıda
C0	E124.4	C silindiri içerde
C1	E124.5	C silindiri dışarıda
Y1	A124.0	A silindiri içerde
Y2	A124.1	A silindiri dışarıya
Y3	A124.2	B silindiri içerde
Y4	A124.3	B silindiri dışarıya
Y5	A124.4	C silindiri içerde
Y6	A124.5	C silindiri dışarıya



<b>NW1: Başlangıç darbesi</b>	UN M 90.1 = M 90.0 S M 90.1	<b>NW6: 4.Durum "A silindiri içeri"</b>	U M 0.3 U "B0" S M 0.4 U M 0.5 R M 0.4
<b>NW2: 0. Durum "Başlangıç şartları"</b>	U M 0.6 U "C0" U "S2" O M 90.0 S M 0.0 U M 0.1 R M 0.0	<b>NW7: 5.Durum "C silindiri dışarı"</b>	U M 0.4 U "A0" S M 0.5 U M 0.6 R M 0.5
<b>NW3: 1.Durum "A silindiri dışarı"</b>	U M 0.0 U "S1" S M 0.1 U M 0.2 R M 0.1	<b>NW8: 6.Durum "C silindiri içeri"</b>	U M 0.5 U "C1" S M 0.6 U M 0.0 R M 0.6
<b>NW4: 2.Durum "B silindiri dışarı"</b>	U M 0.1 U "A1" S M 0.2 U M 0.3 R M 0.2	<b>NW9: Zaman elemanı</b>	U "B1" L S5T#5S SE T 0
<b>NW5: 3.Durum "B silindiri içeri"</b>	U M 0.2 U T 0 S M 0.3 U M 0.4 R M 0.3	<b>NW10: Çıkışlar</b>	U M 0.1 = "Y1" U M 0.2 = "Y3" U M 0.3 = "Y4" U M 0.4 = "Y2" U M 0.5 = "Y5" U M 0.6 = "Y6"

**Örnek : GARAJ KAPISI KUMANDASI**

Şekildeki garaj kapısı kumandası PLC ile yapılacaktır.Çalışma şartları aşağıda belirtildiği gibidir.

\*A anahtarının kapatılması ile garaj kapısı çalışmaya hazır duruma gelecek.

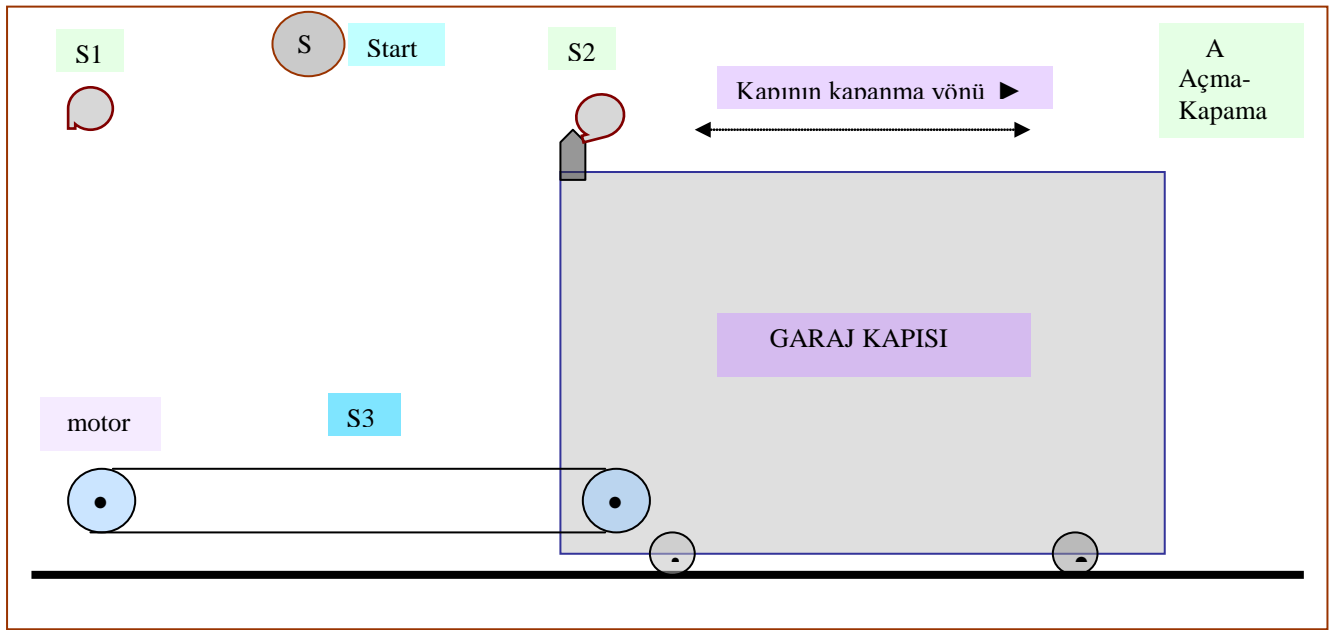
\*S start butonuna basıldığında motor ileri yönde çalışacak ve kapı açılacak,kapı S1 sınır anahtarına dokunduğunda duracak.

\*S start butonuna tekrar basıldığında motor geri yönde çalışacak ve kapı kapanacak ,kapı S2 sınır anahtarına dokunduğunda motor duracak.

\*Kapı açılırken veya kapanırken elektrik kesilip,tekrar geldiğinde ve S start butonuna basıldığında kapı açılacaktır.

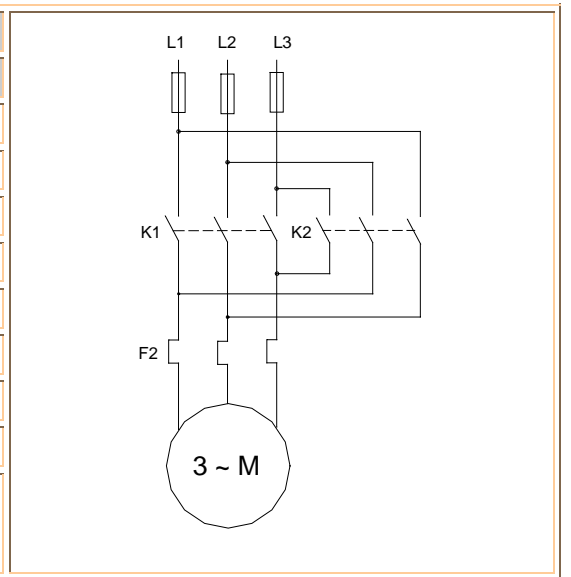
\*Eğer herhangi bir araç kapının ortasında kalırsa,S3 sensörü algılayacak ve kapı, araç gidinceye kadar kapanmayacaktır.

\*Ayrıca elektrik motoru aşırı akım rölesi ile korunacaktır.

**Teknoloji Şeması**

Atama Listesi		
Operand	Sembol	Açıklama
E 124.0	A	Açma kapama anahtarı
E 124.1	S	Sart butonu (NA)
E 124.2	S1	Sınır anahtarı
E 124.3	S2	Sınır anahtarı
E 124.4	S3	Sensör (NA)
E 124.5	F2	Aşırı akım rölesi(NK)
A 124.0	K1	İleri yön rölesi
A 124.1	K2	Geri yön rölesi

NA :Normalde açık  
NK :Normalde kapalı



## Deyim Listesi (AWL)

Kapının açılması

U E 124.0

U E 124.1

UN M 0.1

S M 0.0

U(

O E 124.2

ON E 124.5

)

R M 0.0

U M 0.0

= A 124.0

Kapının Kapanması

U(

U E 124.0

U E 124.1

UN M 0.0

S M 0.1

U(

O E 124.3

ON E 124.5

)

R M 0.1

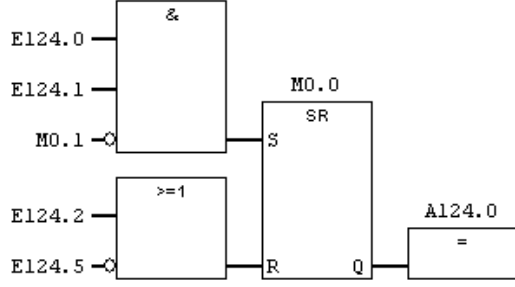
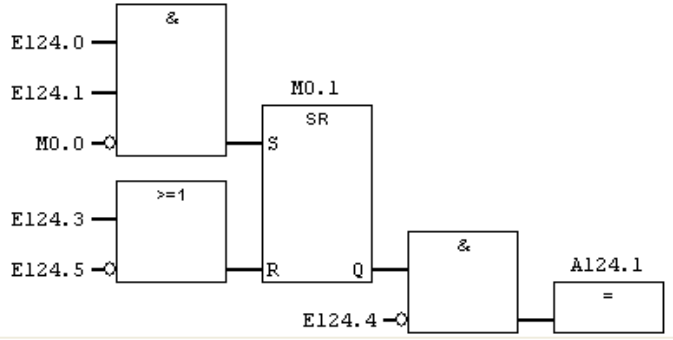
U M 0.1

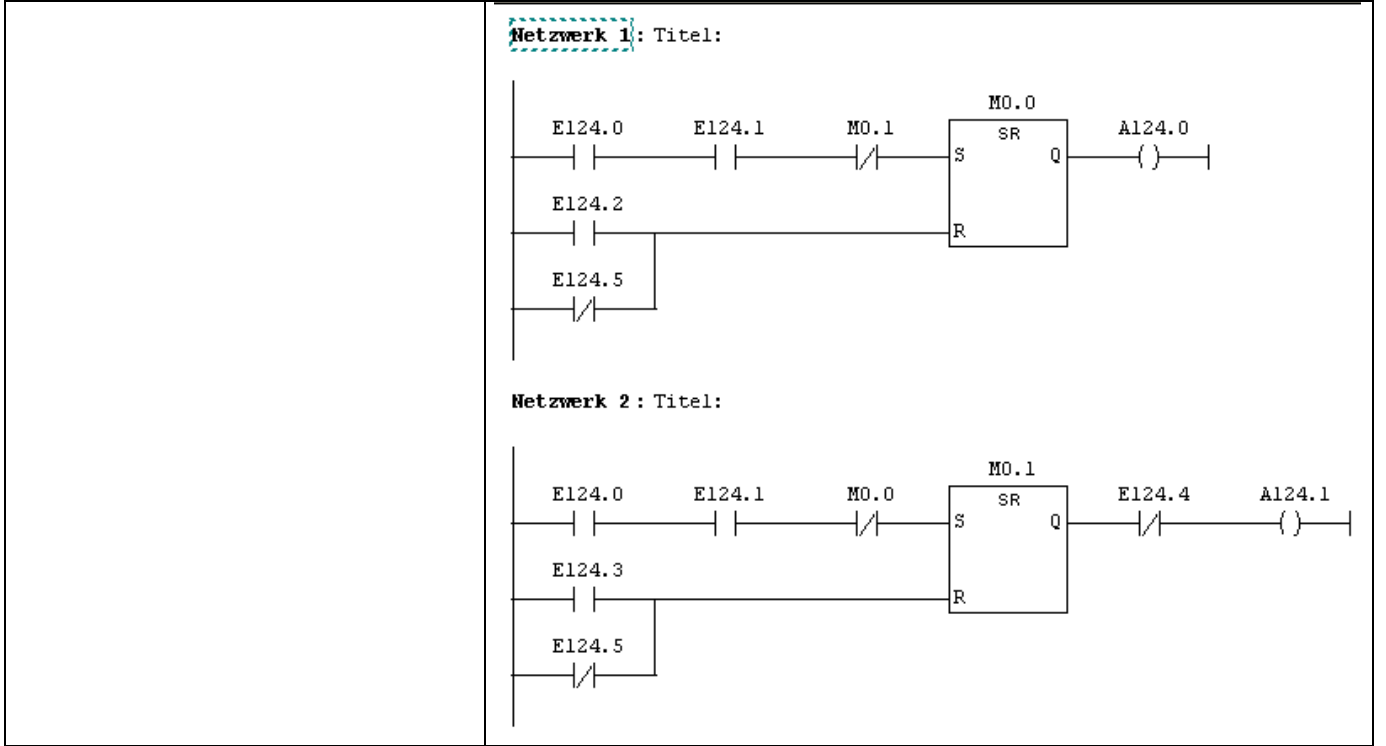
)

UN E 124.4

= A 124.1

## Fonksiyon (FUP)

**Netzwerk 1:** Titel:**Netzwerk 2:** Titel:

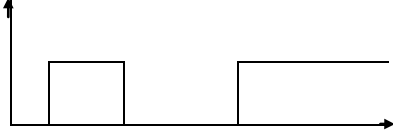


## ANALOG DEĞERLERİN İŞLENMESİ

### SİNYAL ÇEŞİTLERİ

Kumanda tekniğinde sinyaller ikili ,dijital ve analog olmak üzere üç çeşittir. Şu ana kadar işlenen konularda ikili ve dijital sinyaller kullanıldı. Programlanabilir denetleyiciler belirli giriş ve çıkış modüllerinin mevcut olması durumunda analog sinyalleri de okur, işler ve dışarıya verebilirler.

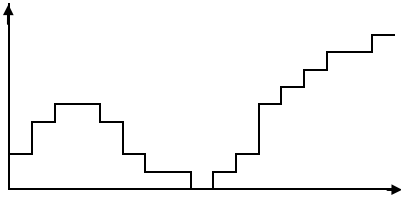
#### İkili sinyal



İkili sinyalde sinyal durumu '0' veya '1' olabilir.

'0' : Kapı kapalı → 0 Volt  
'1' : Kapı açık → 24 Volt  
Su sıcaktır veya soğuktur.

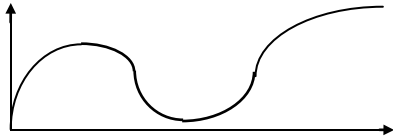
#### Dijital sinyal



Dijital sinyalde, sinyal durumu; "0" ile "1" arasındaki veya kapının açık ile kapalı olması arasındaki durumlarda belli zaman aralıklarında sorgulanır.

Rakamsal gösterimli termometre ile su sıcaklığının tespiti.

#### Analog sinyal



Analog sinyalde ise aradaki durumlar sonsuz sayıda sorgulanmaktadır. Yani kapının ne kadar açık olduğu. İbre göstergeli termometre ile su sıcaklığının tespiti

Örneğin :

Isı : -50 → +150 C<sup>0</sup>  
Debi : 0 → 200 l/dk  
Devir : 0 → 1500 dev/dk

## ANALOG SİNYAL GİRİŞ VE ÇIKIŞI

SPS sadece elektriksel yani gerilim veya akım olarak sinyalleri okuyabilir ve dışarıya verebilirler. Elektriksel olmayan sinyaller mevcut ise bunların SPS dışında elektrik sinyallerine dönüştürülmesi gerekir. Yani yukarıda verilen örnekteki ısı, debi, devir sayısı gibi değerler elektrik sinyallerine dönüştürüldükten sonra işlenmesi gereklidir.

Örneğin; bir motorun devir sayısı 0 ile 1500 dev/dk arasında değişiyor ise; motor dönmediğinde 0 volt, 1500 dev/dk ile döndüğünde ise 10 volt gerilim elde edilmelidir. Ara değerlerde de o oranda gerilim alınmalıdır.

Bir analog sinyalin değerlendirilebilmesi için sinyalin hangi değiştirilebilir parametrede olduğunun bilinmesi gerekir. Bir elektrik sinyali için iki sinyal parametresi mevcuttur.

- Gerilim değeri; 'U'
- Akım değeri; 'I'

#### Sinyallerin çevrilmesi

Analog sinyaller SPS tarafından direkt olarak okunamaz, çünkü SPS sadece '0' ve '1' leri tanır. Bunun için sinyal dönüştürücüleri vardır. Giriş gerilimine orantılı olarak bir dijital değer atayan analog - dijital çevirici (ADU), dijital çıkış değerlerini analog gerilim sinyallerine dönüştüren dijital - analog dönüştürücüler (DAU) kullanılır.

**S7-300 CPU' larda analog işleme**

S7-300 Sistemlerinde analog değer işlenmesi ( Dijitalize edilmiş analog değer okumak, dijital değerleri analog değer olarak dışarı vermek ) direkt olarak 'Periferi = Çevre birimleri' üzerinden yapılır. Yani S5 sistemlerinde olduğu gibi fonksiyon modülleri (FB 250= Analog giriş, FB 251=Analog çıkış) üzerinden değil, analog olarak okunan değer direkt PAE ye yazılır, ataması yapılacak değer direkt PAA dan gerçekleştirilir.

- L PEW 128** → Periferi giriş word' u 128 den okunan analog değer dijital olarak Akü 1' e yüklenmesi.  
**T PAW 128** → Akü 1 deki dijital değer Periferi çıkış word' u 128 den analog olarak dışarı verilmesi

Bir elektrik sinyali dijital sinyallere dönüştürülürken dönüştürüldüğü bit sayısı çözünürlüğü verir.  
 Örneğin; 0 → 10 volt değeri, 2 bit ile dönüştürülüyor ise

Bit durumu			İfade ettiği Voltaj değeri
0	0	→	0.....2.5 V
0	1	→	2.5.....5 V
1	0	→	5.....7.5 V
1	1	→	7.5...10 V

"0 → 10" volt dört birimde değerlendirilir, her bitlik değişim 2.5 volt' a karşılık gelir.

S7 300 sistemlerinde dönüştürme işlemi program modülüne göre, işlem 8 ile 15 bit arasında yapılır.

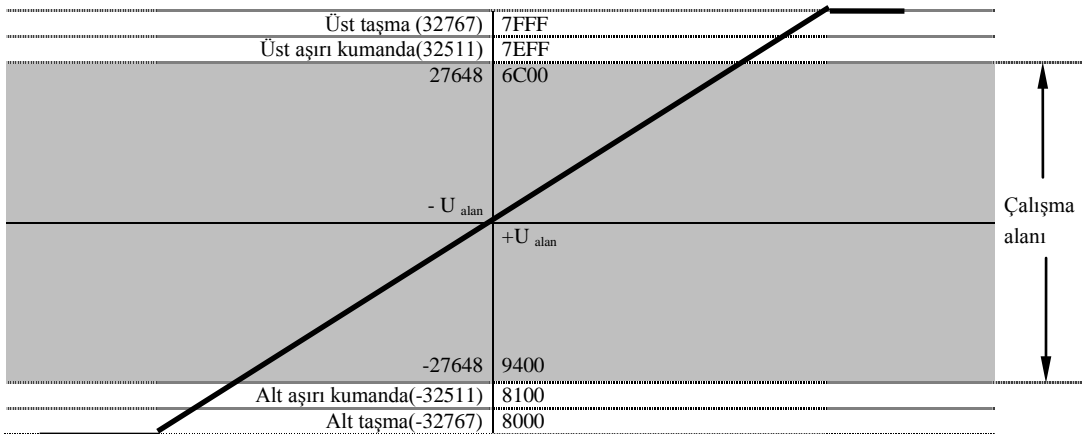
Bit olarak işlem	Analog değer		Birimler( Artan ve eksilen)		Adım uzunluğu
	High - Byte	Low - Byte	Desimal	Heksadesimal	
8	VZ XXX XXXX	X - - - - -	- 32768...+32767	8000...7FFF	128
9	VZ XXX XXXX	XX - - - - -	- 32768...+32767	8000...7FFF	64
10	VZ XXX XXXX	XXX - - - - -	- 32768...+32767	8000...7FFF	32
11	VZ XXX XXXX	XXXX - - - -	- 32768...+32767	8000...7FFF	16
12	VZ XXX XXXX	XXXX X - - -	- 32768...+32767	8000...7FFF	8
13	VZ XXX XXXX	XXXX XX - -	- 32768...+32767	8000...7FFF	4
14	VZ XXX XXXX	XXXX XXX -	- 32768...+32767	8000...7FFF	2
15	VZ XXX XXXX	XXXX XXXX	- 32768...+32767	8000...7FFF	1

Adres n

Adres n+1

VZ = Ön işaret bit'i: 0= Pozitif, 1= Negatif

Yukarıdaki çizelgede görüldüğü gibi CPU 314 IFM' lerde çözünürlük 12 bit ( 11 + VZ ) üzerinden gerçekleştirilir ve görüntüleme işi A124.0 → A124.7 (high-byte) ve A125.4 → A125.7 (low-nibble) dual çıkışlarında yapılır.



Bu eğri ölçme alanına bağlı olmaksızın, tüm program grupları için geçerlidir.

+32768	→	+11,7589 V
+27648	→	+ 10 V
0	→	0 V
- 27648	→	- 10 V
- 32768	→	- 11,7589 V

15 bit + VZ ile işlem gören bir CPU'da en düşük değerlikli bit değişimi  $10 : 27648 = 0,000361689$  V dur  
11 bit + VZ ile işlem gören bir CPU'daki her bitlik bir değişim  $0,000361689 \times 16^3 = 0.005787$  V olur.

Analog giriş kanalından okunan değer 16 bit tam sayıdır. Bizimde analog çıkış kanalına göndereceğimiz sayı 16 bit tam sayı olmalıdır.

#### Analog program modülü kanal adresleri:

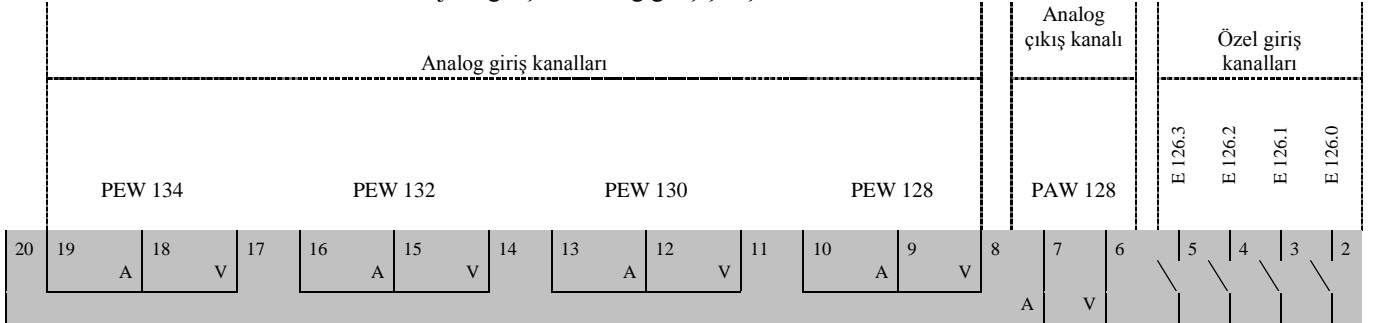
314 IFM modüllerde kanal adresleri aşağıdaki gibidir. (Diğer CPU 'lardaki kanal adresleri el kitaplarından temin edilebilir.)

1. giriş kanalı : PEW 128
2. giriş kanalı : PEW 130
3. giriş kanalı : PEW 132
4. giriş kanalı : PEW 134

1. çıkış kanalı : PAW 128

Giriş kanallarından okunan elektriksel büyüklüklerin gerilim ( V ) veya akım ( I ) cinsinden alınabilmesini sağlamak amacıyla, analog kart üzerindeki bağlantıların değiştirilmesi gerekir.

CPU 314 IFM SPS lerde özel dijital giriş ve analog giriş çıkış kanalları



Donanım ayarları yapılan analog karta ait yazılım ayarlarının da yapılması gerekir. Bunun için **Hardware Configuration** tablosunda montaj rayına eklenen her analog kart satırının üzerinde sağ tuş ile özelliklere girilir. Orada giriş ve çıkışlara ait kullanacağımız kartın özelliklerine göre kataloglardan bulduğumuz aşağıdaki ayarlar yapılır.

Ölçme tipi; Voltaj "V", akım "I", direnç "R", termoelement v.b

Ölçme aralığı; "V" için; +/- 80mV, +/- 250mV, +/-500mV, +/- 1V, +/- 2.5V, +/- 5V, +/- 10V  
"I" için; +/- 3.2mA, +/- 10mA, +/- 20mA, 0..20mA, 4..20mA,  
"R" için; 150 ohm, 300 ohm, 600 ohm  
"Termoelement" için; Tip N, E, J, L, K

<sup>3</sup> Son 4 bit perdelenmiştir, her bitlik perdeleme bir önceki bit'in değerinin 2 kat artmasını sağlar.

**Örnek :** Analog program modülünün 1. Kanalından bir analog değer okunacak (örneğin 2 volt ) ve dijitalleştirilmiş şekilde dual çıkış kanallarına gönderilecektir. Ayrıca dual girişlerden verilecek bir değer ( Örneğin 0010\_0000\_0110 ) analog çıkış kanalına gönderilecektir. Çıkış kanallarındaki değerleri hesaplayınız.

L PEW 128	Periferi giriş word'u 128 dan ayarlanan 2 voltluk gerilimin dijitalleştirilerek Akü 1' e yüklenmesi
T AW 124	Akü 1 deki değerın dual çıkış kanalına transfer edilmesi (Tam 2 volt' a ayarlanmışsa görülmesi gereken değer 0001_0101_1001 dir)
L EW 124	Dual giriş kanalındaki (0010_0000_0110) dijital değerın Akü 1' e yüklenmesi
T PAW128	Akü 1' deki dijital değerın Analog' a dönüştürülerek periferi çıkış kanalına transfer edilmesi ( $\cong 3$ volt)

**Örnek:** Bir tankın doluluk oranı izlenecektir. Eğer tank %80 ve üzerinde dolu ise A124.0, %20 ve altında dolu ise A124.7 lambaları, doluluk oranı bu iki değer arasında ise A124.4 lambası yanacaktır.

Tankın doluluk oranı '0' ile '10' volt gerilim veren bir sensörden alınmaktadır.

```

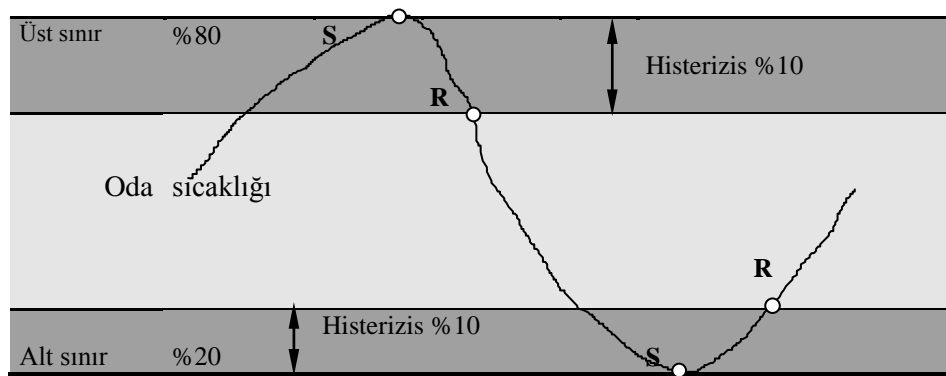
L PEW 130
L 22118 // Tank % 80 dolu ise (27648 / 0.80 = 22118)
>=I
= A 124.0
L PEW 130
L 5530 // Tank %20 dolu ise (27648 / 0.20 = 5530)
<=I
= A 124.7
UN A 124.0
UN A 124.7
= A 124.4

```

**Örnek:** Bir odanın sıcaklığı klima sistemi ile kontrol edilmektedir. Oda sıcaklığı istenen değerın üzerine çıktığında soğutucu, istenen değerın altına indiğinde ısıtıcı çalışacak belirlenen değerlere geldiğinde duracaktır.

Odanın sıcaklığı, 0 ile 10 volt arasında çıkış veren bir sensör ile algılanmaktadır.

Oda sıcaklığına bağlı olarak üst ve alt çalışma sınırları aşağıdaki diyagrama göre olacaktır.



SEMBOL TABLOSU			
SEMBOL	ADRES	TİP	AÇIKLAMA
ger_deg	PEW 128	WORD	Oda sıcaklığı
ısıtıcı	A 124.0	BOOL	Isıtıcı
soğutucu	A 124.7	BOOL	Soğutucu

DB1			
ADRES	SEMBOL	TİP	BAŞ.DEĞERİ
0.0	ust_sin	INT	22118
2.0	alt_sin	INT	5530
4.0	his	INT	2764

NW1: Soğutucunun set edilmesi

```
L "ger_deg"
L DB1.DBW 0
>=I
S "soğutucu"
```

NW2: Soğutucunun reset edilmesi

```
L DB1.DBW 0
L DB1.DBW 4
-I
L "ger_deg"
>I
R "soğutucu"
```

NW3: Isıtıcının set edilmesi

```
L "ger_deg"
L DB1.DBW 2
<=I
S "isitici"
```

NW3: Isıtıcının reset edilmesi

```
L DB1.DBW 2
L DB1.DBW 4
+I
L "ger_deg"
<I
R "isitici"
BE
```