



PLC PROFESYONEL

S7-300 PRO 1



ENTEK EĐİTİM TEKNOLOJİLERİ

PLC NEDİR ?

Programlanabilir Lojik Kontrolörler probleme bağlı olmaksızın üretilmiş kumanda ve kontrol elemanlarıdır.

Bütün kumanda problemlerinin çözümünde mantık işlemleri, bellek fonksiyonları, zaman ve sayıcı gibi elemanlara ihtiyaç vardır. Bunlar PLC'lerde üretici firmalar tarafından hazır olarak sunulmuş durumdadır.

Basit bir programlama ile bütün bu imkanlar problemin çözümünde bir araya getirilebilirler.

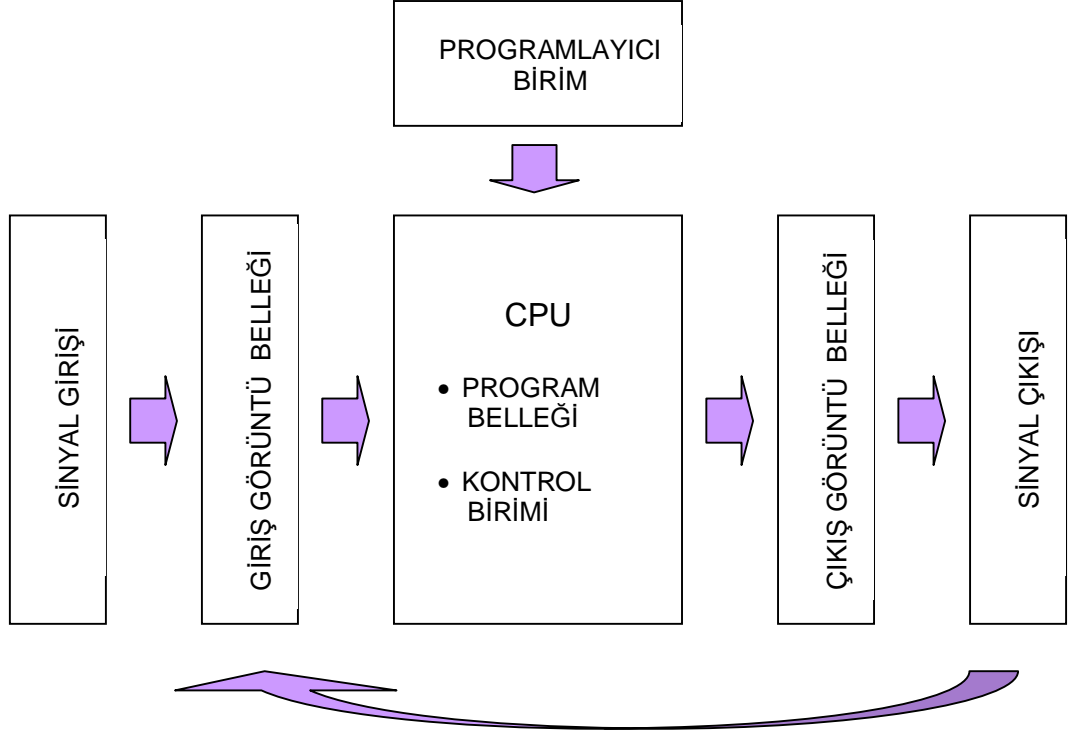
PLC ile yapılan çözümde kumanda devresi yazılımla sağlandığından daha kolay ve güvenilirlerdir.

- Daha az yer tutar ve daha az arıza yaparlar
- Yeni bir uygulamaya daha çabuk adapte olurlar
- Kötü çevre şartlarından kolay etkilenmezler
- Daha az kablo bağlantısı isterler
- Hazır fonksiyonları kullanma imkanı vardır
- Giriş ve çıkışların durumları izlenebilir

S7-300 PLC'ler

- Orta performanslı işler için geliştirilmiş modüler PLC sistemi
- Farklı farklı otomasyon problemlerine cevap verebilecek nitelikte zengin ürün çeşidi
- Proseste bir geliştirme gerektiğinde kolay ve sorunsuz olarak genişleme olanağı
- MPI, Profibus ve Endüstriyel Ethernet gibi haberleşme ağlarına bağlanabilme olanağı
- Programlama aşamasında geniş bir komut kümesine destek sağlaması
- SCL, Graph gibi üst düzey programlama teknikleriyle programlayabilme

PLC NASIL ÇALIŞIR ?



S7-300 PLC'lerin girişi 24 VDC veya 120/230 VAC'dir. 24 V ile çalışan S7-300 20.4V ile 28.8 V'luk gerilimler arasındada çalışabilmesine rağmen buna izin verilmemelidir. Girişe uygulanacak gerilim buton, sınır anahtarı, sıcaklık, seviye, basınç sensörleri gibi anahtarlama elemanları üzerinden alınır.

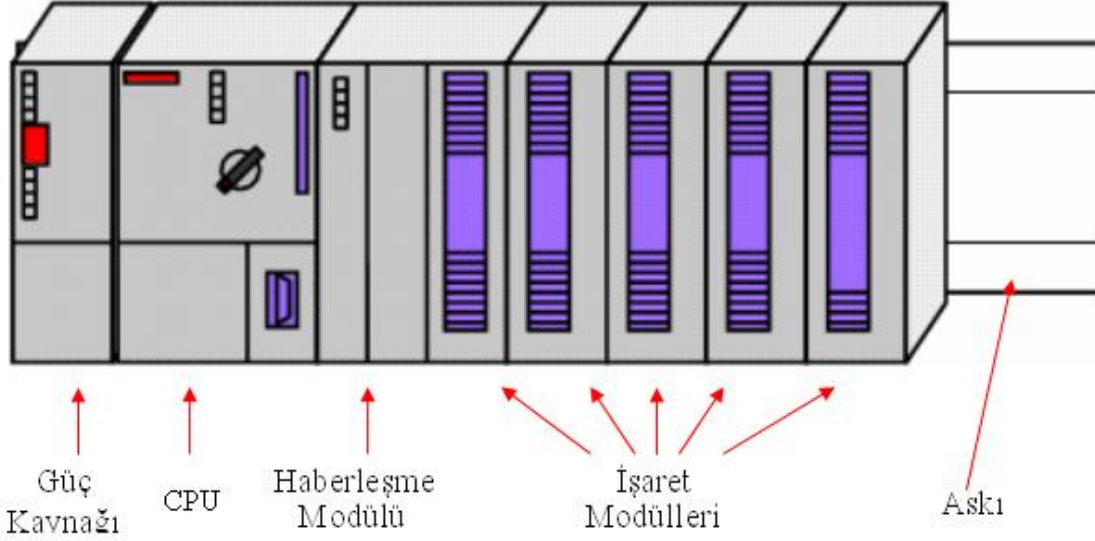
Giriş bilgisi giriş görüntü belleğine yazılır ve buradan merkezi işlem ünitesine gönderilir. Merkezi işlem ünitesindeki bellek iki kısımdan oluşmuştur :

- ROM bellek : Cihazın kendine ait olan bilgilerin tutulduğu ve sadece okunabilen bellektir
- RAM bellek : İşletilmesi istenilen programı barındıran ve silinebilen bellektir. Ram bellek iki kısımdan oluşur. Programın yazıldığı yükleme alanı (Load Memory) ve programın çalıştırıldığı çalışma alanı (Work Memory). Yazımlanan program Micro Memory Card'a yüklenir.

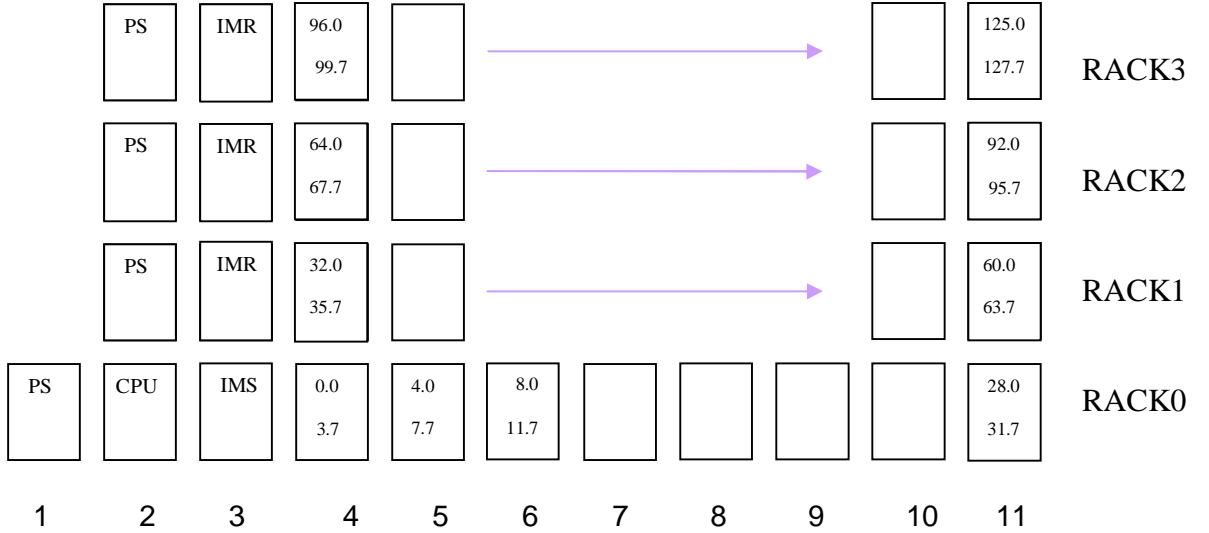
Giriş görüntü belleğinden alınan bilgi işletilecek programa bildirilir ve program yürütülür. Elde edilen veriler çıkış görüntü belleğine yazılır. Veriler çıkış sinyali olarak çıkış katına aktarıldığı gibi tekrar giriş görüntü belleğine giriş bilgisi olarak gönderilir.

Çıkış sinyalleri kontrol edilen sisteme ait kontaktör, röle, selenoid gibi çalışma elemanlarını sürer. PLC transistör çıkışlı ise DC 0.5A, Triyak çıkışlı ise AC 1 A ve Röle çıkışlı ise AC/DC 2 A'in geçmesine izin verir.

MODÜLER YAPI



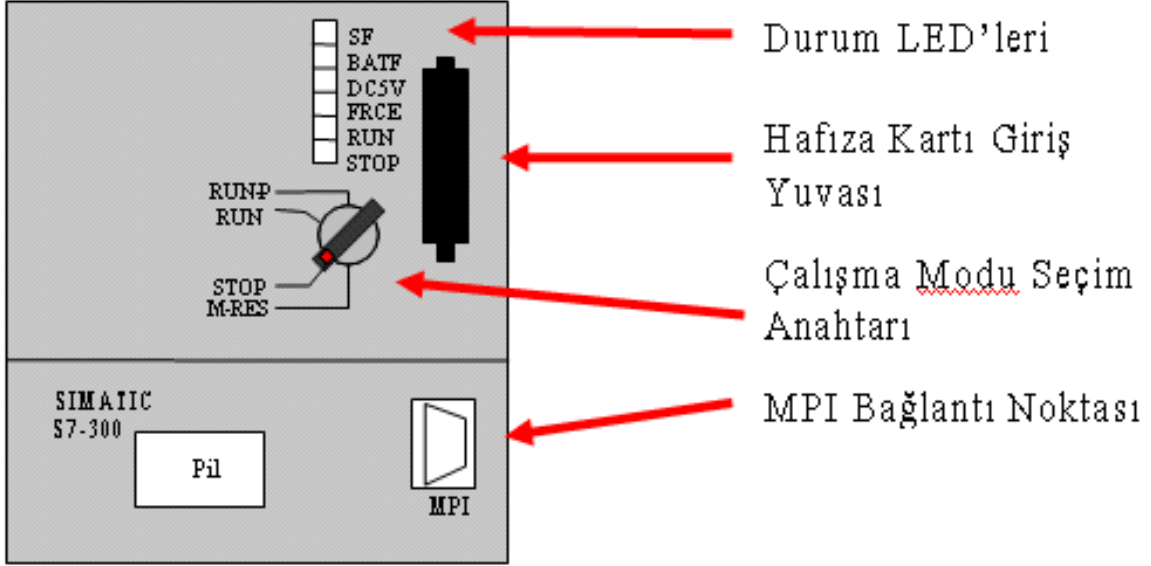
1. RACK YAPILANDIRMASI



- PS : Güç kaynağı (Power Supply)
CPU : Merkezi işlem birimi (Central Process Unit)
IMS : Gönderici ara birim (Interface Modül Sender)
IMR : Alıcı birim modülü (Interface Modül Receive)

S7-300 PLC'lere 32 modül eklenebilir. Her raya 8 adet sinyal modülü eklenebilir. Maksimum 4 ray kullanılabilir. Her sinyal modülü 32 bitlik veri içerir. Toplam 1024 bitlik veri işlenebilir. Rayların kendi aralarında haberleşmesini sağlamak için haberleşme birimine ihtiyaç vardır. Eğer sadece 0 nolu ray kullanılacaksa 3.slot boş bırakılır.

2. CPU YAPISI



(S7-300 IFM 314)

MRES :Hafıza Reset

STOP :Durma Modu

RUN : Çalışma Modu, bilgisayar tarafından sadece okuma anlamında ulaşım mümkün.

RUN-P : Çalışma modu, bilgisayar tarafından hem okuma hem de yazma anlamında erişme mümkün.

SF: Grup Hatası, CPU'da yada modüllerde bir hata var

BATF: Pil hatası,pilin gerilim seviyesi düşük yada pil yok.

DC5V:5 Volt DC sinyali

FRCE :Bir yada daha fazla giriş yada çıkış zorlanıyor

RUN :CPU çalışmaya başlarken yanıp söner, çalışma modunda ise sürekli yanar.

STOP :Durma modunda sürekli yanar. Hafızayı sıfırlama işlemi süresince yavaş bir şekilde yanıp söner,işlemin bitmesiyle beraber hızlı olarak yanıp söner.

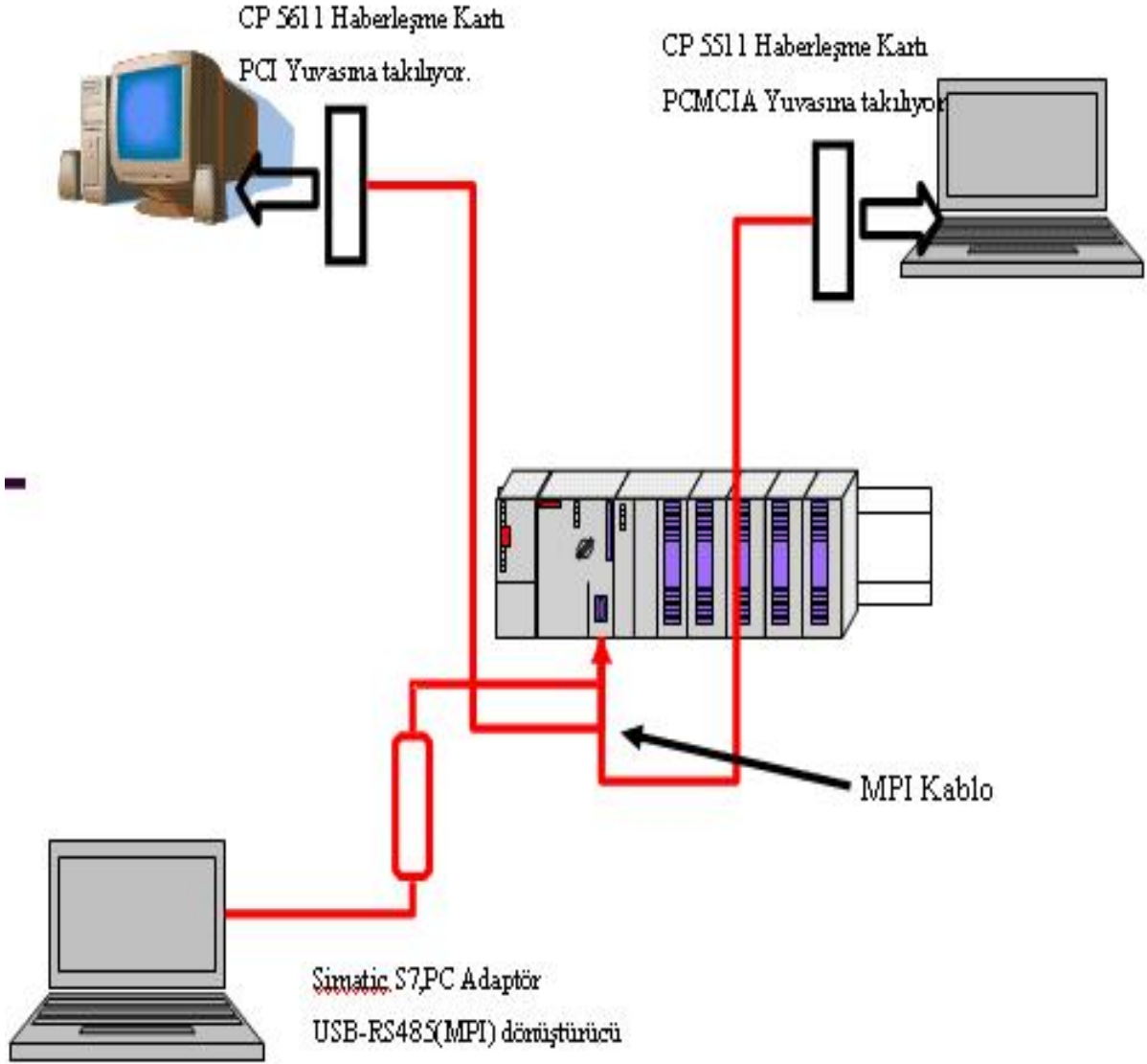
Eski tip CPU'larda 3.6V'luk lityum pil bulunur. Ömrü 5 yıldır. Ancak 2 yılda bir değiştirilmesi önerilir. Programı korur, hardware ayarlarını korur ve gerçek zaman saatini işletir.

Yeni tip CPU'larda pil yerine süper kondansatör kullanılır. 10 hafta kadar bilgilerin muhafaza edilmesini sağlar.



| SIEMENS | S7-200 (216) | S7-300 (312C) | S7-400 (416) | S7-300 (314 IFM) |
|---|---------------------|------------------------|---------------------|-----------------------------|
| Dijital giriş, çıkış sayısı | 128 bit | 1024 bit (10DI-6DO) | 128 Kbit | 1024 bit |
| Analog giriş, çıkış sayısı | 20 byte | 128byte | 8192 byte | 64 byte |
| Sayıcı sayısı | 256 | 128 | 512 | 64 |
| Zaman elemanı sayısı | 256 | 128 | 512 | 128 |
| Durum tespit işareti sayısı (Yardımcı röle) | 256 | 2048 | 16384 | 2048 |
| Çalışma Hafızası | 8 KByte | 16 Kbyte | 512 Kbyte | 24 Kbyte |
| Çalışma hızı | 0.8 ms | 0.3msn | 0.08 msn | 0.3 msn |
| Program işleme şekli | Yapısal | Yapısal | Yapısal | Yapısal |

4. PG/PC- PLC HABERLEŞMESİ



PG/PC ile PLC arasındaki haberleşmeyi sağlamak için farklı çözümler söz konusudur.

ISA Yuvası üzerinden (Ör : MPI-ISA Card)

PCI Yuvası üzerinden (Ör : CP5611)

PCMCIA Yuvası üzerinden (Ör : CP5511)

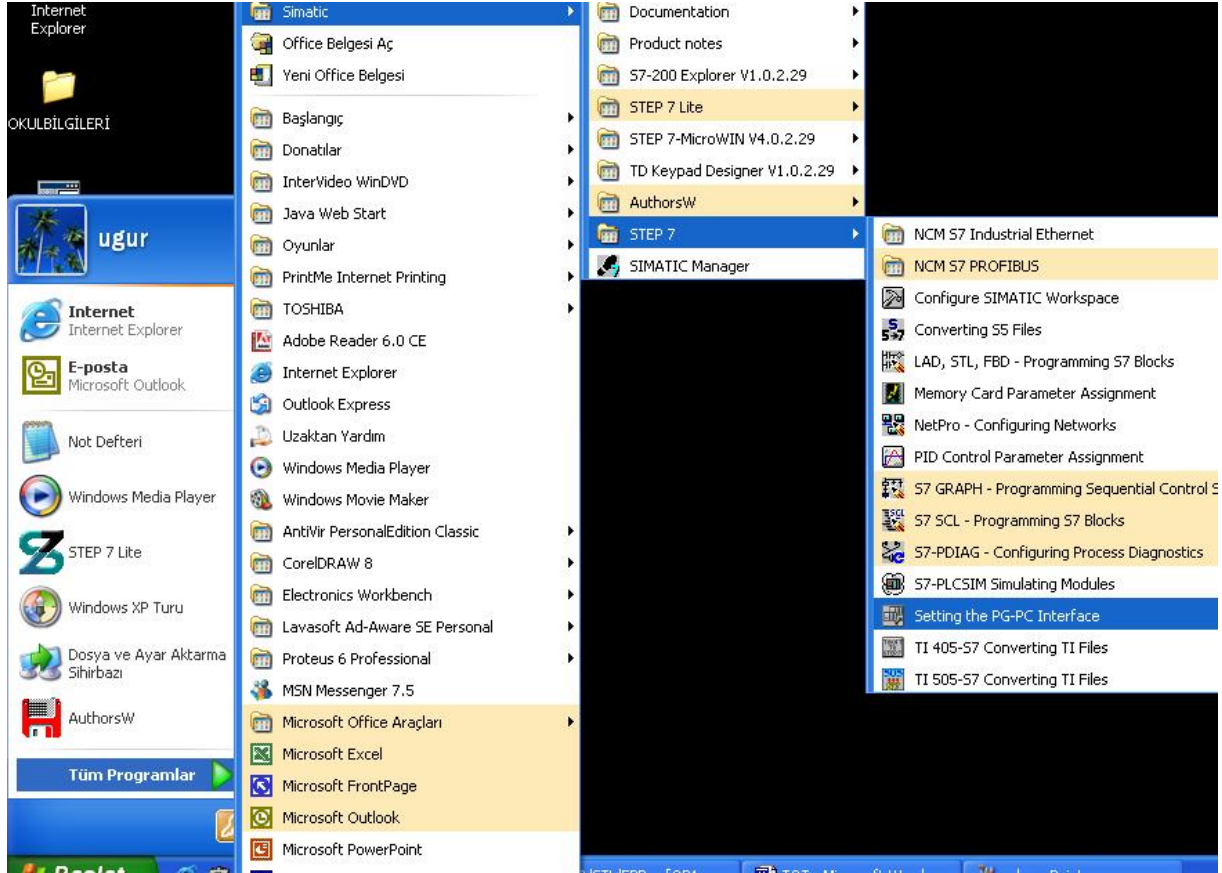
USB çıkışı üzerinden (Ör : Simatic S7,PC Adaptör)

5. PC – PLC HABERLEŞMESİNİN SAĞLANMASI

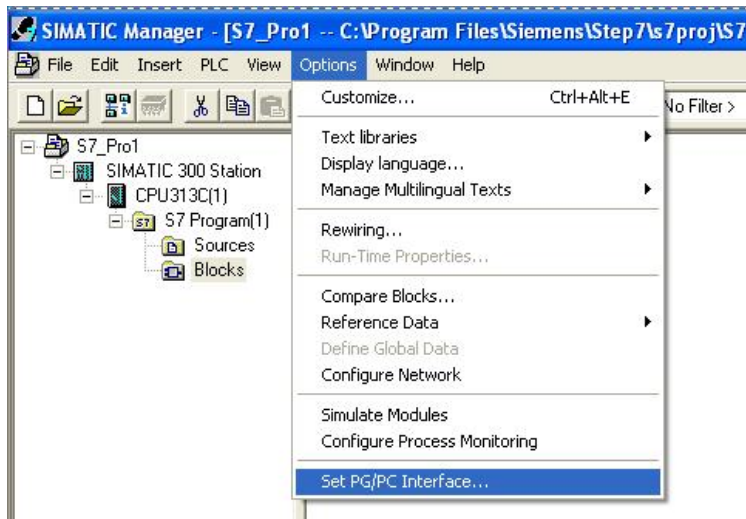
1. Adım

PC-PLC bağlantısı için bu iki cihazın haberleşmesini sağlayan MPI adaptörü ayarlarının yapılmalıdır. Bu işlem STEP 7 ile beraber gelen PG-PC Interface programı vasıtasıyla yapılır.

Başlat à SIMATIC à STEP 7 à Setting PG-PC Interface



veya



Manager programı
açıldıktan sonra
Options menüsünün
altında Set PG/PC
Interface... seçilebilir.

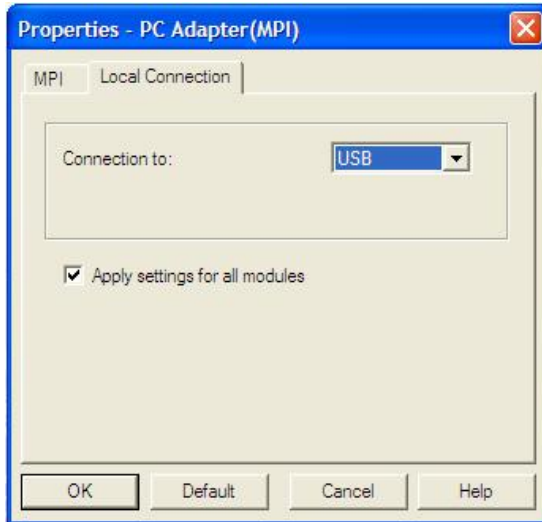


2. Adım

PC ile PLC arasındaki haberleşme protokolü olan MPI'ın özelliklerini belirlemek üzere ilk olarak "PC Adapter(MPI)" seçilir ve "Properties" e basılır.

Bu pencerede, kullanılan arayüzün PC ile haberleşmek için kullandığı fiziksel ortam belirlenir.

Seçilmiş olan haberleşme arayüzü (USB) yüklenmiş olarak belirecektir.



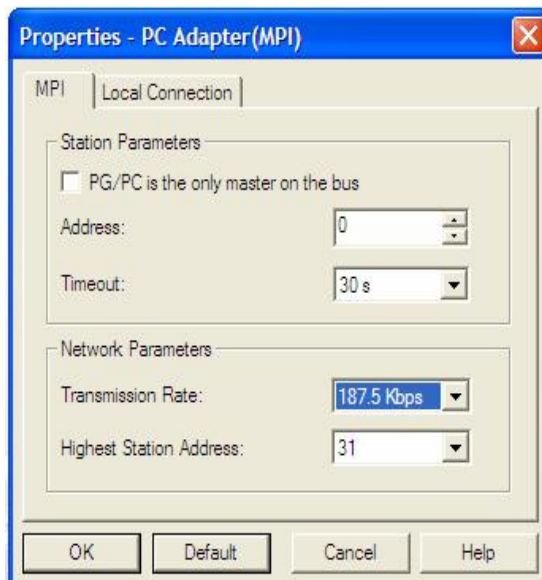
MPI address: PC'nin, MPI ağına bağlandığında alacağı adres belirlenir.

Timeout: MPI ağında bir hata oluştuğunda ağın ne kadar süreyle izleneceğini belirler. Mesela ağda haberleşme yoğunluğundan dolayı cevap paketlerinde bir gecikme olduğunda ayarlanan süre kadar PC cevabın gelmesini bekler. Alabileceği değerler 10 s., 30 s., ve 100 s.'dir

Transmission Rate: Ağda kullanılacak haberleşme hızı belirlenir.

Alabileceği değerler 1.5 Mbps., 187.5 Kbps., 19.2 Kbps.'dir

Highest Station Address: Ağa bağlı olan cihazlara verilebilecek en yüksek adres girilir. Alabileceği değerler 15, 31, 63, 126'dır.



STEP 7 MANAGER EDITÖRÜ İLE ÇALIŞMAK

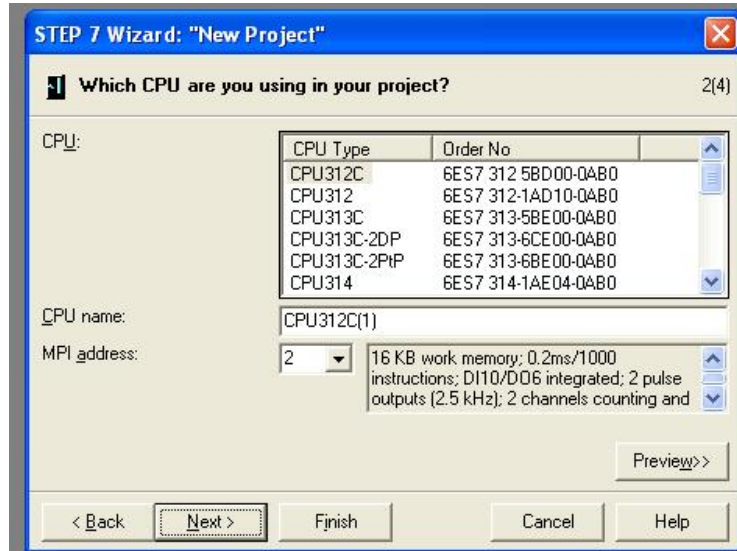


Masaüstünde bulunan kısa yoldan veya Başlat_Tüm Programlar_Simatic_Simatic Manager yolu takip edilerek program başlatılabilir.



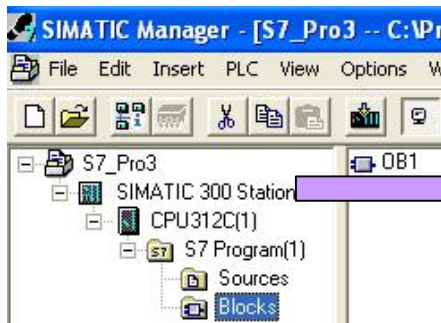
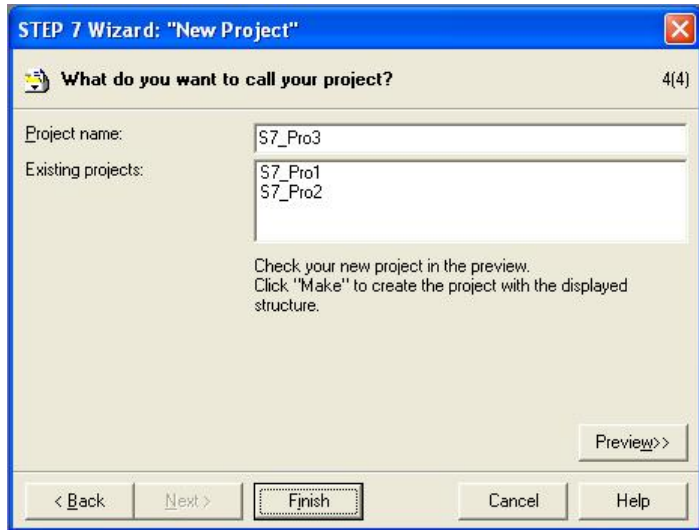
Açılış sihirbazı çalışır. Finish seçilirse daha önceki projede tanımlanan donanım mevcut kabul edilir ve S7 Pro.. adı ile proje açılır. Finish yerine Next seçilerek sonraki basamağa geçilebilir.

Bu aşamada çalışılacak olan CPU 'nun tanımlanması gerekmektedir.

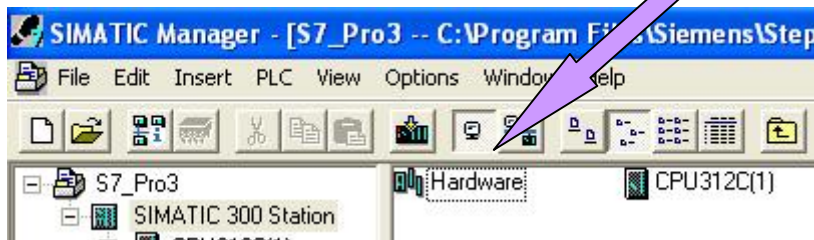


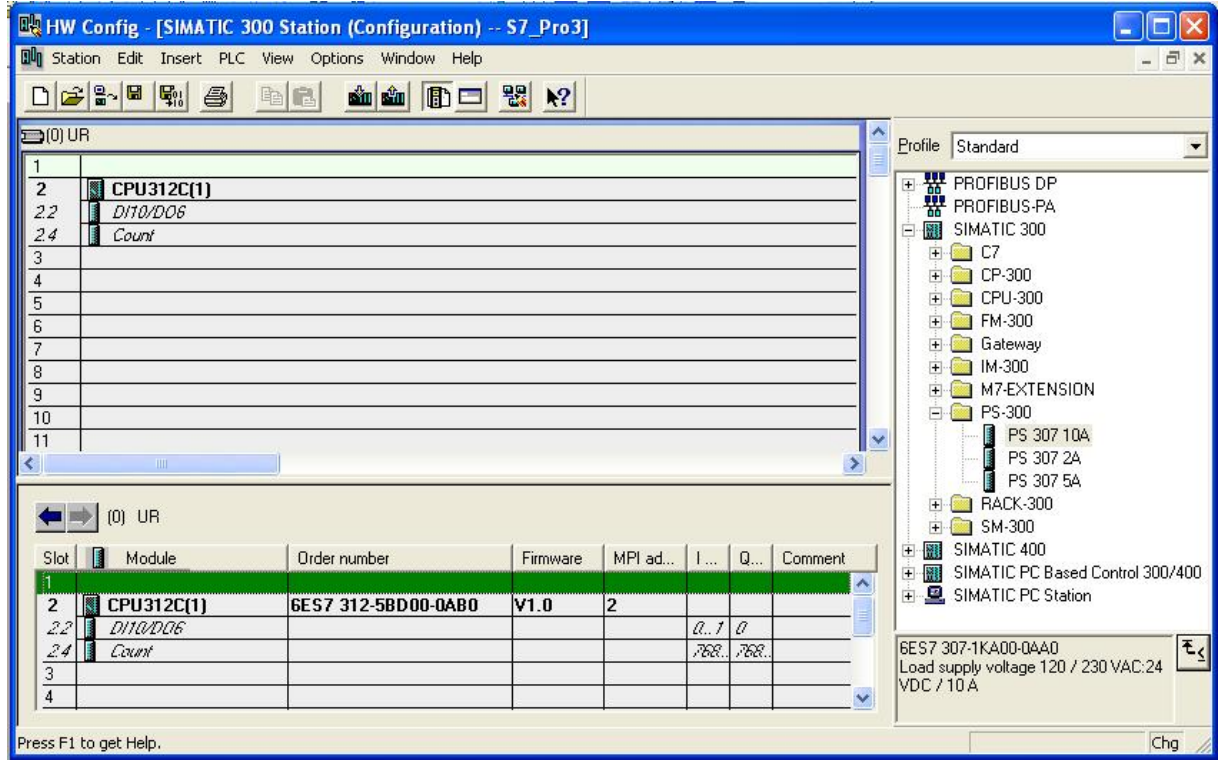


Çalışılacak olan Organizasyon Blokları ve çalışılacak olan programlama dili seçilmelidir. Next tuşuna basıldığında oluşturulacak olan projeye bir isim verilmesi istenir.




Seçilen CPU ile birlikte çalışacak olan donanımı oluşturmak için açılan programın ana sayfasında SIMATIC 300 Station seçilir. Açılan menüden hardware seçilmelidir.





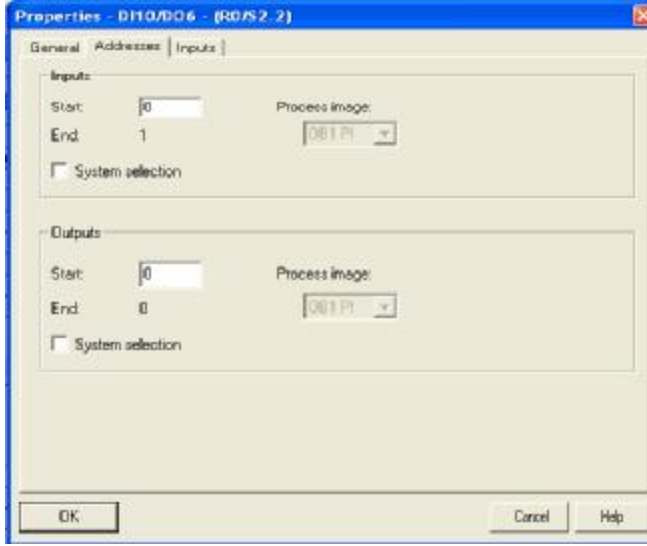
Rack'ın 1. slotunda Power suply bulunur. Şayet siemens ürünü bir PS kullanılıyorsa PS-300 seçeneğinden ilgili ürün çift tıklanarak veya mause'la sürüklenerek seçilmelidir. Sonraki slotta CPU yer alır. 3. slot haberleşme modülü için rezervedir. Başka modül tanımlanamaz. Diğer modüller 4 ile 11. slotlara tanımlanmalıdır.

Donanım tanımlanırken aynı ürün seçme kataloğunda aynı üründen birden fazla tanımlı ise CPU'nun portunu kapatan kapağın altında CPU versiyonu ve dış yüzeyinde ise sipariş kodu bulunur. Donanım tanımlanırken bu değerlere dikkat edilemelidir.

Hardware seçim işlemi tanımlandığında Station_Save and Compile seçilmeli ve donanım Download edilmelidir. Download işlemi PLC menüsünün altından veya  kısa yol tuşundan yapılabilir. Manager programı hatalı donanım seçildiğinde herhangi bir uyarı vermez. Sadece CPU seçilmemişse ve donanım sıralaması hatalı yapılmışsa uyarı verir. Donanımı test etmek için Station _ Consistency Check seçilir. Step7 Lite editör programında ise hatalı donanım seçilmesi durumunda program donanımın tanımlamasını kabul etmez.

CPU'nun özelliklerini görmek için kursorü CPU'nun üzerine getiriniz ve mause sağ tıklayınız. Açılan menüden Object Properties seçiniz.

Order Number ile belirtilen hanede yazılan kod ürünün sipariş kodudur.



Simatic Manager'da PLC'nin giriş ve çıkış adreslerinin değiştirilmesi mümkündür. Örneğin giriş adresleri değiştirilmek istendiğinde, adreslerin bulunduğu slot üzerine gelinerek mouse sağ tıklanır. Açılan menüden Object Properties seçilir. Açılan menüde Address secilir.

Inputs kısmında bulunan System selection kutucuğu seçili olmaktan çıkarılır ve start yazan haneye ilk giriş adresinin byte numarası yazılır. Donanımın Save and Compile edilerek CPU'ya download edilmesi ile yeni adres tanımlamaları işlerlik kazanacaktır.

PLC ÇALIŞMA DİLİNİN SEÇİMİ



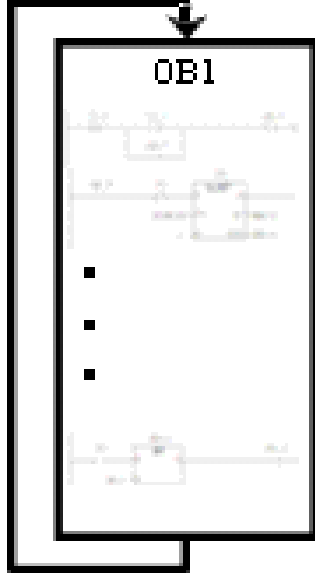
Options _ Custumize seçildiğinde Lanuage menüsünden Almanca İngilizce veya kurulmuşsa diğer diller açılır. Seçim yapılmalıdır. Eğer bir değişiklik yapılmaz ise kurulum dili ile program çalışmaya devam eder.

PROGRAMLAMA TEKNİKLERİ

1. Doğrusal Programlama

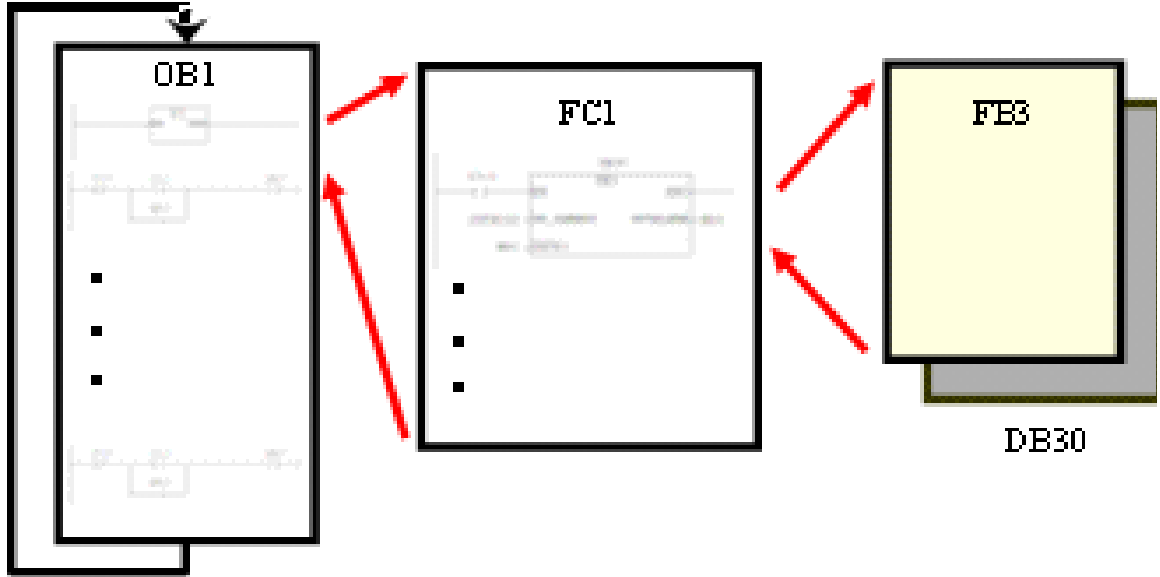
Proje bünyesinde PLC için geliştirilen program tek bir blok üzerinden programlanır. Alt program gibi yapılar kullanılmaz. Çözüm tek bir program parçası ile üretilir.

Linear Programlama



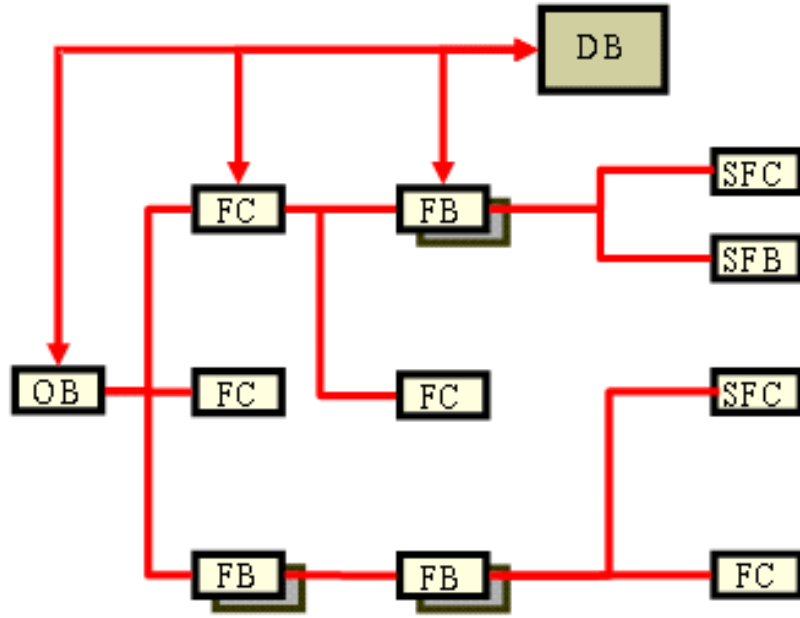
Proje bünyesinde PLC için geliş program tek bir blok üzerinden program Alt program gibi yapılar kullanılmaz. Ç tek bir program parçası ile üretilir.

2. Yapısal Programlama



Bu programlama tekniğinde program uygun alt parçalara bölünür. Ana Program ise bunları çağıran bir yapıda programlanır. Hem projenin tasarımı hem de işletilmesi aşamasında kullanıcılara kolaylıklar sağlar.

YAPISAL PROGRAMLAMA ELEMANLARI



OB: Organizasyon Blokları program modüllerinin hangi sırayla işleneceğini belirler. İşletim sistemi tarafından çağrılan bloklardır. OB1 ana programın koşturulduğu organizasyon bloğudur. OB'unun çevrim süresi S7-300 PLC'lerde 150 ms'dir. Bu süre aşırsa işletim sistemi OB80'i çağırır. PLC stop konumuna geçer. CPU'nun tipine göre organizasyon bloklarının sayısı değişebilir.

FC: Fonksiyon yapısal programlama mantığı içerisinde gelişmiş bir alt program gibi davranan yapılardır. Kompleks program parçalarını küçük, takip edilebilir yapılara bölmek için kullanılır.

FB: Fonksiyon blokları FC'a benzemekte olup hafızaya sahiptirler ve her biri için DB atanmıştır

SFC/SFB: Sistem Fonksiyon ve Sistem Fonksiyon Blokları CPU ile birlikte gelen hazır yapılardır.

DB: Data Blokları veri saklanmak için kullanılan yapılardır ki program deyimleri içermezler. Örneğin program içerisinde editörde yapılan açıklamalar (comment) DB'da saklanır. Özel ve Genel olmak üzere iki farklı tipi mevcuttur.

PROGRAM BLOKLARINI OLUŞTURMA

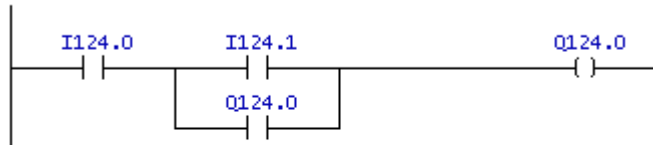
Organizasyon Bloğunun (OB1) içinde FC, FB vs. gibi alt programlar oluşturmak için Insert_S7 Block... menüsünden yararlanılabileceği gibi sağ taraftaki pencere üzerinde iken mausun sağ tuşuna basıldığında Insert new object seçeneğinde yeni FC,FB,DB.. elemanlarının açılmasını sağlar. Açılan pencerede isim verilmesi istenir.



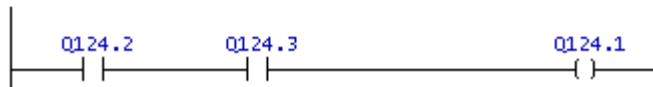
PROGRAM GÖSTERİMİ

Bir kumanda yada kontrol sisteminin çözümüne ilişkin sözel yada matematiksel kuralların PLC program belleğine aktarılması özel bir programlama dili ve derleyicisi aracılığıyla yapılır. PLC programlama dilleri, komut ile programlama ve grafiksel programlama olarak iki ana başlık altında toplanabilir. *LAD, STL, FBD*. STL komut ile programlama tekniğine diğer ikisi ise grafiksel programlama tekniğine girer. Bunlar aşağıdaki gibidir.

Network 1: Title:



Network 2: Title:



LADDER DIAGRAM (MERDİVEN DIAGRAMI)

Network 1: Title:

```

A      I      124.0
AC
O      I      124.1
O      Q      124.0
)
=      Q      124.0

```

Network 2: Title:

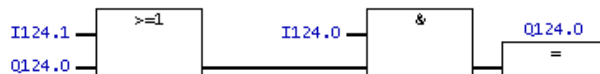
```

A      Q      124.2
A      Q      124.3
=      Q      124.1

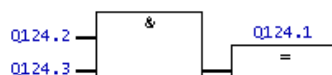
```

STATEMENT LIST (KOMUT DİZİNİ)

Network 1: Title:



Network 2: Title:



FUNCTION BLOCK DIAGRAM (FONSİYON BLOK DIAGRAMI-LOJİK KAPI GÖSTERİMİ)

SAYI FORMATLARI

Bit

Ör: I0.0, I0.1, Q4.0, M5.9

Byte(B)

Ör: MB9, IB3, QB4

Integer(I)

Ör: MW10, MW12, IW0

Double Integer(DI)

Ör: MD10, MD14

Floating Point

Ör: MD10, MD14

1 Byte = 8 Bit

1 Integer = 16 Bit

1 Integer = 2 Byte

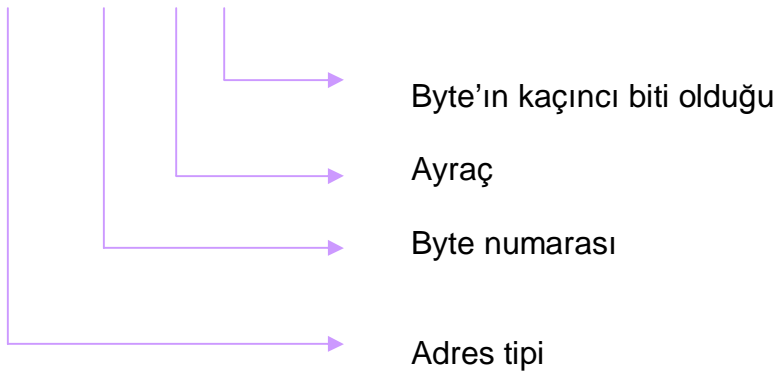
1 Double Integer = 32 Bit

1 Double Integer = 4 Byte

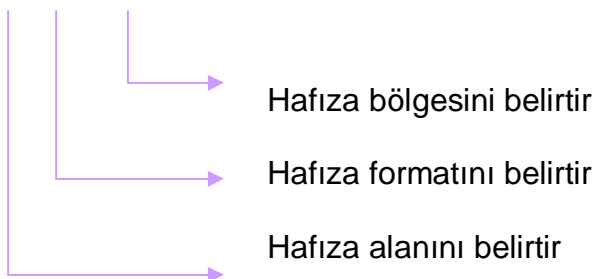
| | İşaretsiz Tam Değer | İşaretili Tam Değer |
|-----------------------|-------------------------------------|---|
| Byte | 0...255 | -127...128 |
| Integer | 0...65535 (0...FFFF) | -32,768...32,767 (8000...7FFF) |
| Double Integer | 0...4,294,967,295 (0...FFFFFFFF) | -2,147,483,648... 2147,483,647 (8000 0000...7FFF FFFF) |

| | MSB | | | | | | LSB | |
|-----|-----|---|---|---|---|---|-----|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| I 0 | | | | | | | | |
| I 1 | | | | | | | | |
| I 2 | | | | | | | | |
| I 3 | | | | | | | | |
| I 4 | | | | | | | | |
| I 5 | | | | | | | | |
| I 6 | | | | | | | | |
| I 7 | | | | | | | | |
| . | | | | | | | | |
| . | | | | | | | | |

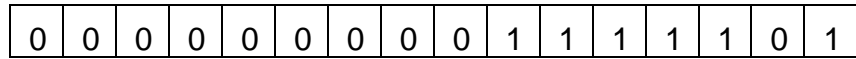
I 0 . 4



MB20

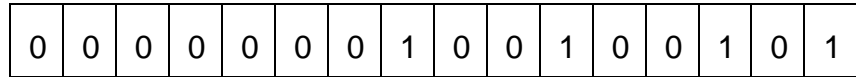


Integer



$$\begin{aligned} &= 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 64 + 32 + 16 + 8 + 4 + 0 + 1 \\ &= 125 \end{aligned}$$

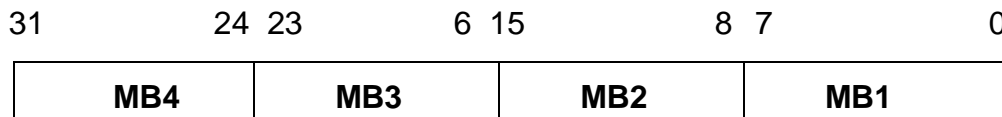
BCD



İşaret
0000 → +
1111 → -

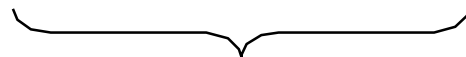
MSB

LSB



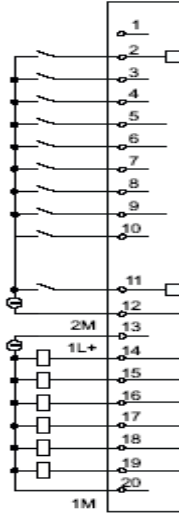
MW1

MW2

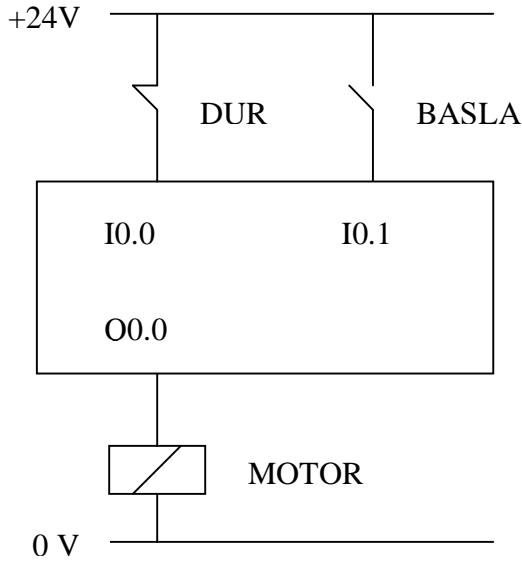


MD1

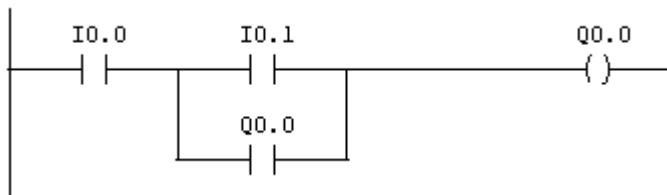
CPU 312C MODELİ PLC'İN KABLAJI



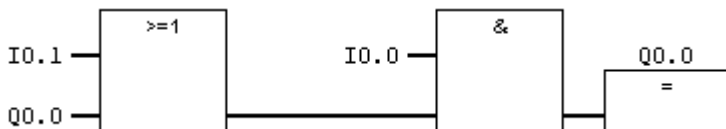
PLC'de L+ ile ifade edilen klemenslerin tümüne +24VDC, M ile gösterilen klemenslerin tümüne 0V ve topraklama işareti olan noktalarda muhakkak toprağa bağlanmalıdır. PLC üzerinde birden daha fazla giriş olmasının nedeni PLC'nin CPU'sunun enerji ihtiyacı dışında girişlerin ve çıkışlarında beslenmesi gereğidir. Şayet girişler beslenmezse sahadan gelen bilgiler CPU'ya ulaşmayacak, çıkışlar beslenmediğinde de CPU'nun çıkışa gönderdiği bilgiler çıkış adreslerine ulaşmayacaktır.



Örnek olarak sürekli çalışma devresini incelersek iki giriş ve bir çıkış bilgisine ihtiyaç bulunmaktadır. Bu girişler PLC'de sırası ile giriş adreslerine bağlanır. Çıkışta kullanılan ve alıcımızı çalıştıracak olan rölede ilk çıkışımız olan Q0.0 adresine bağlanır. Elektriksel bağlantının tamamlanması ile birlikte yazılım yapılmalı ve PLC'ye yüklenerek aktif hale getirilmelidir.

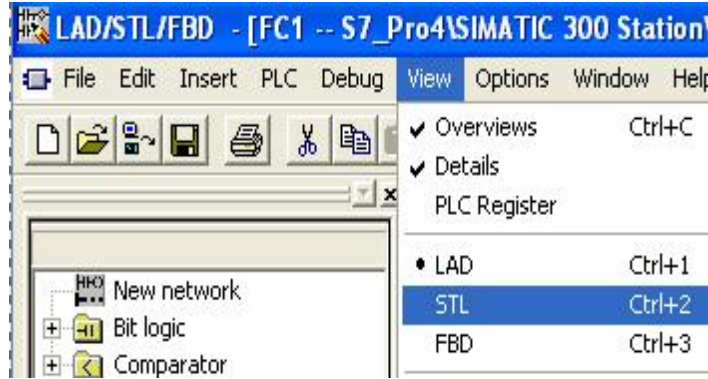
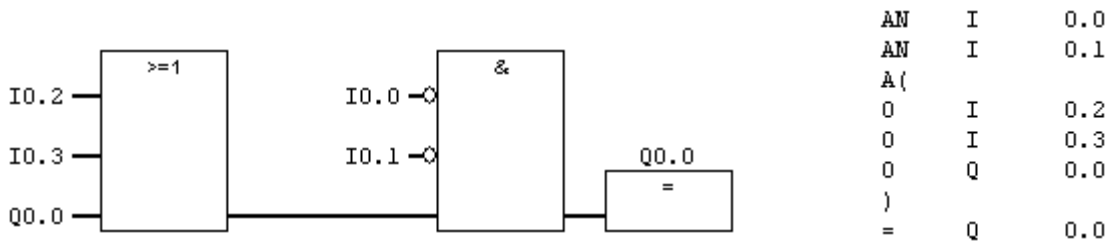
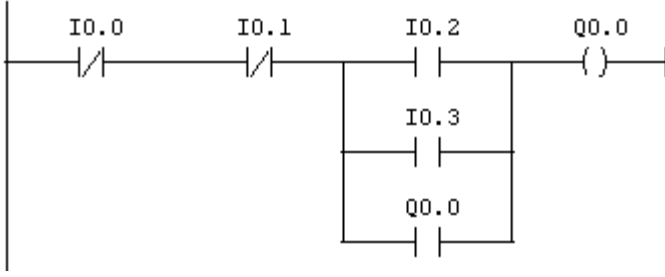


| | | |
|----|---|-----|
| A | I | 0.0 |
| A(| | |
| O | I | 0.1 |
| O | Q | 0.0 |
|) | | |
| = | Q | 0.0 |



ÖRNEK:

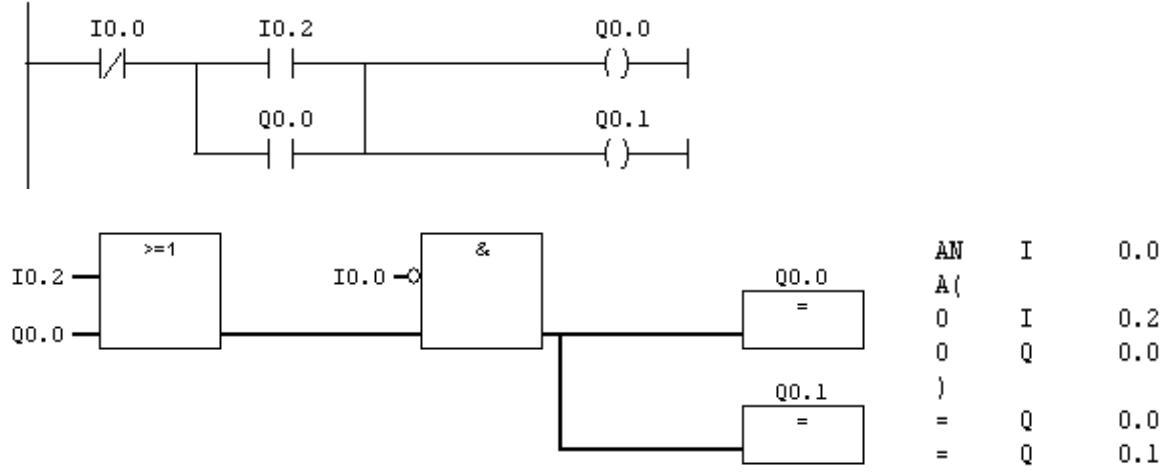
Motor iki ayrı yerden çalıştırılabilir ve iki ayrı yerden durdurulabilir.



NOT: Ladder diyagramda yapılan yazılımları Statement List (STL) veya Function Block Diagram (FBD) programlama diline çevirme işlemi programın yazıldığı pencerede View menüsü altından yapılır. Ayrıca kısa yol tuşları da kullanılabilir. STL ile yazılan programlar Ladder'a ve FBD'a, Ladder yazılım FBD'a ve FBD yazılım Ladder'a sorunsuz dönüşebilir. Ancak Ladder ve FBD'da yazılan programlar STL'ye dönüşemeyebilir.

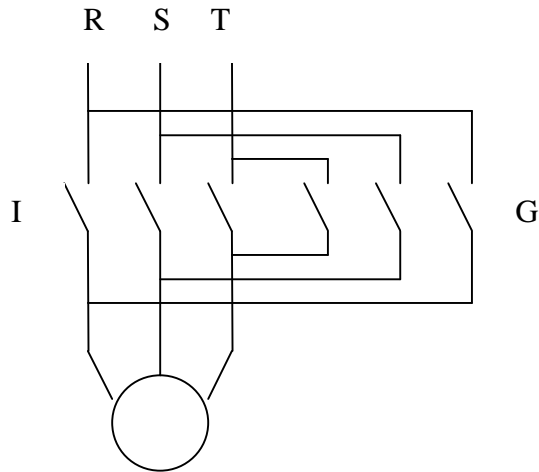
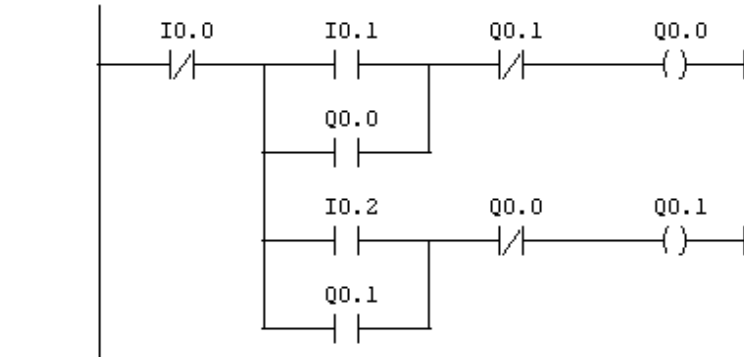
ÖRNEK:

Start butonuna basılınca iki motor aynı anda çalışsın stopla durdurulsun.



ÖRNEK :

3 fazlı asenkron motorun ileri geri çalışması için gerekli yazılımı yapınız



NOT:

Üç fazlı asenkron motorları devir yönlerini değiştirmek için güç devresinde iki fazın yeri değiştirilmelidir.

ÖRNEK :

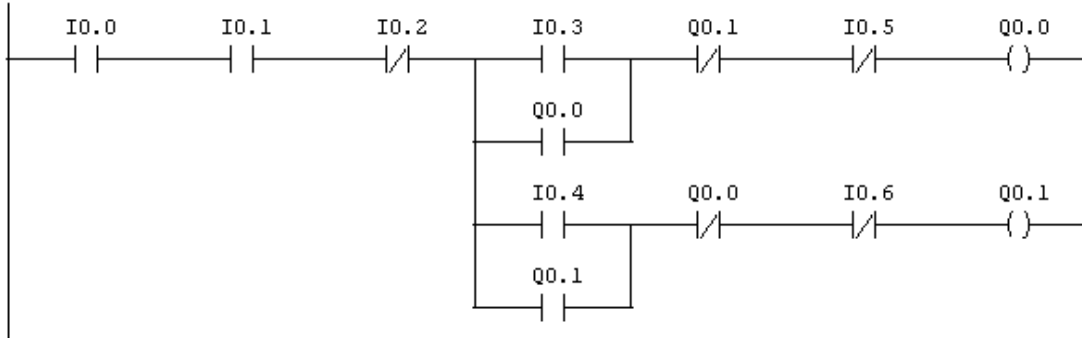
Yukarıdaki örnekte çalıştırılan motor otomatik kapıda kullanılmaktadır. Kapının geri ve ileri hareketlerini sınırlayan switchler bulunmaktadır. Motor istenildiğinde durdurulabilmekte ve herhangi bir acil durum oluştuğunda devrenin enerjisi farklı bir noktadan kesilebilmektedir. Ayrıca motoru korumak amaçlı A.A röleside kullanılmaktadır. Gerekli yazılımı yapınız.

OB1 : "Main Program Sweep (Cycle)"

Comment:

Network 1: Title:

Comment:



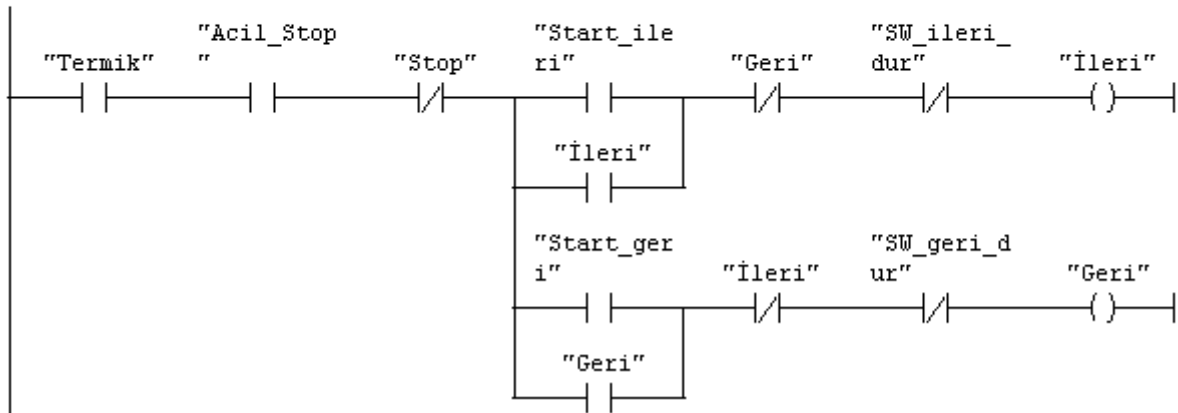
Aynı devre aşağıdaki şekilde gösterilirse daha açıklayıcı olacaktır

OB1 : 3 FAZLI ASENKRON MOTOR İLERİ GERİ ÇALIŞMA DEVRESİ

Konveyör sistemde kullanılan motor

Network 1: Tüm devre

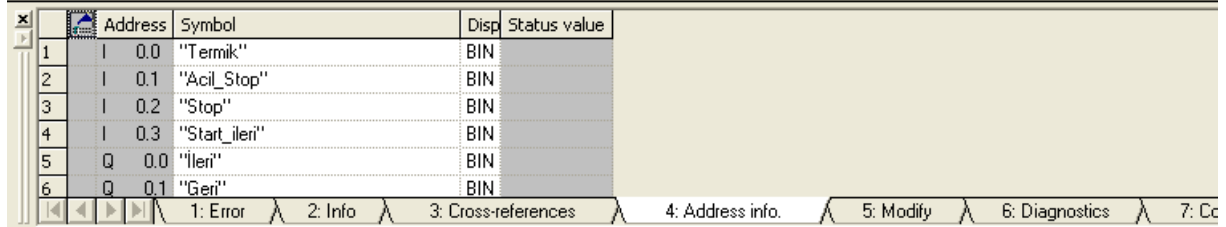
30.03.2006 tarihinde devreye alınmıştır.
Motor gücü 3 KW



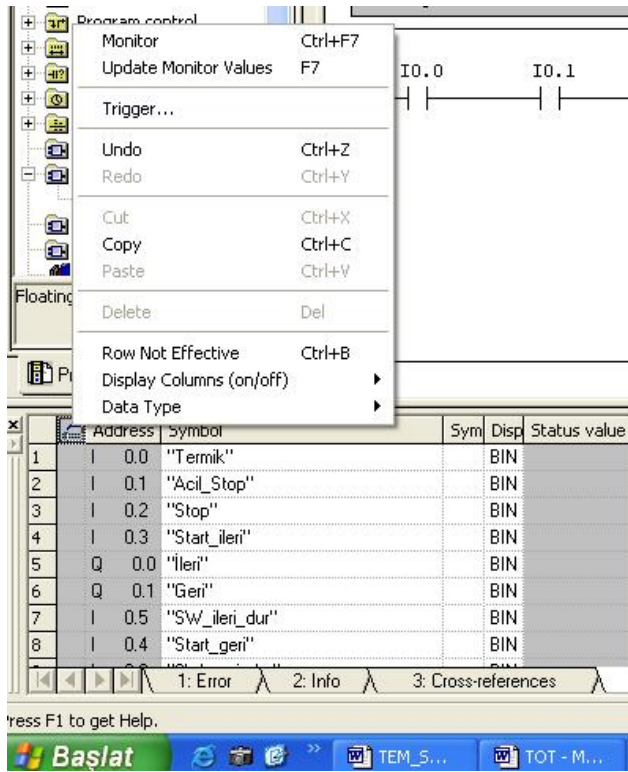
Programa ad ve açıklama eklemek mümkün olduğu gibi her network'ede ad ve açıklama eklenebilir.

View _ Display with _Comment

Seçilirse açıklama eklenebilir



| | Address | Symbol | Disp | Status value |
|---|---------|---------------|------|--------------|
| 1 | I 0.0 | "Termik" | BIN | |
| 2 | I 0.1 | "Acil_Stop" | BIN | |
| 3 | I 0.2 | "Stop" | BIN | |
| 4 | I 0.3 | "Start_ileri" | BIN | |
| 5 | Q 0.0 | "Ileri" | BIN | |
| 6 | Q 0.1 | "Geri" | BIN | |



"View" menüsü altında "details" seçildiğinde ekranın alt kısmında yukarıdaki pencere açılacaktır. "Address info." seçildiğinde atanmış adresler dizinlenecektir. "Symbol" kısmına uygun görünen semboller atanır.

Sembol tablosunun üzerine kursor getirilip mausun sağ tuşu seçildiğinde yandaki menü açılır. Monitör seçilirse PLC çalışırken adresleri almış olduğu değerler izlenebilir.

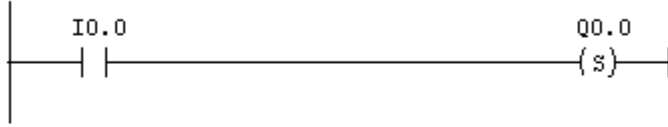
Sembollerin aktif olması için yine "Display with" menüsünden "Symbolic Representation" seçilmelidir. Hem adres hem semboller aynı anda görünsün isteniyorsa "Symbol information" da seçili olmalıdır.

KURMA-SİLME (SET-RESET) İŞLEMLERİ

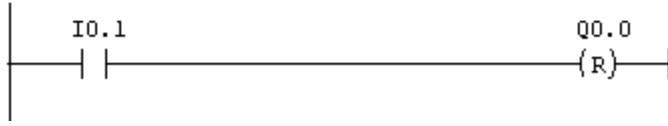
Kurma komutu, bir bitlik adres alanının içeriğini "1" yapar. Silme komutu ise bir bitlik adres alanının içeriğini "0" a çeker.

LADDER DİAGRAMI

Network 1: Title:



Network 2: Title:



KOMUT DİZİNİ

Network 1: Title:

| | | |
|---|---|-----|
| A | I | 0.0 |
| S | Q | 0.0 |

Network 2: Title:

| | | |
|---|---|-----|
| A | I | 0.1 |
| R | Q | 0.0 |

FONKSİYON BLOK DİAGRAMI

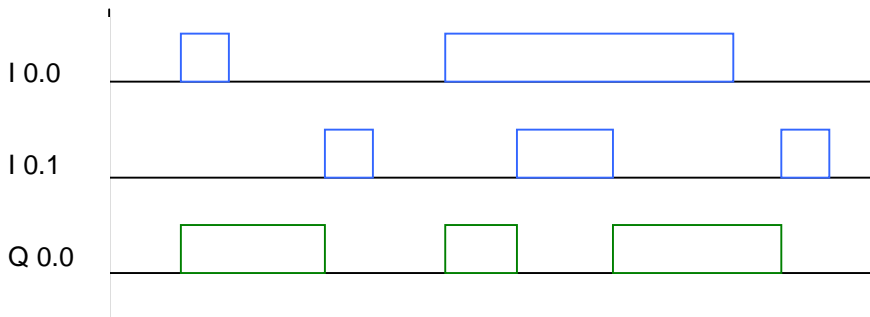
Network 1: Title:



Network 2: Title:

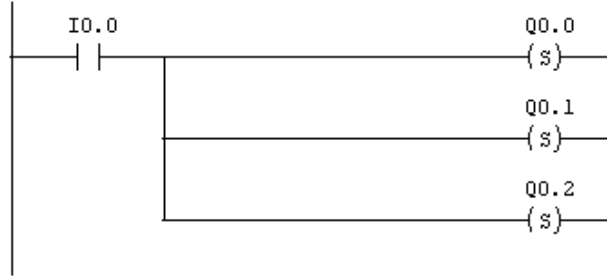


ZAMAN DİAGRAMI

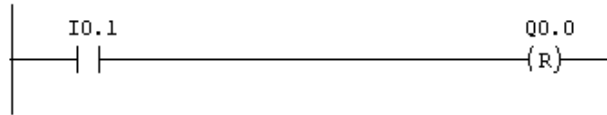


ÖRNEK :

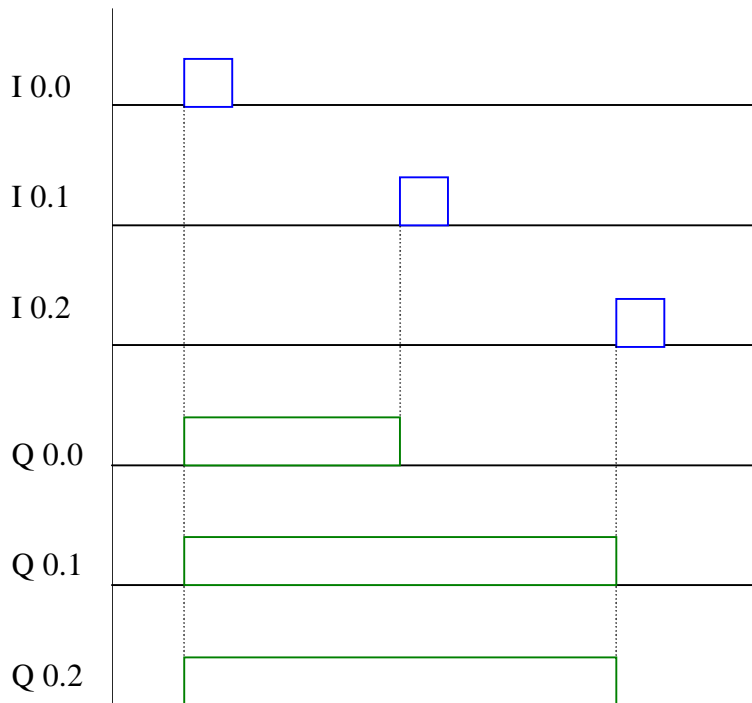
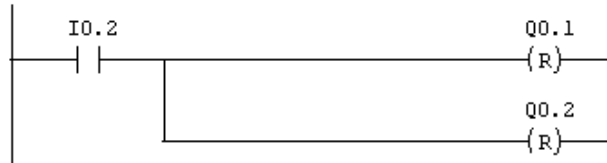
Start butonuna basıldığında 3 lambada yanacaktır. Dur1'e basıldığında 1. Dur 2'ye basıldığında 2.ve 3. motorlar duracaktır. Ladder diagramını ve durum grafiğini çiziniz.



Network 2 : Title:



Network 3 : Title:

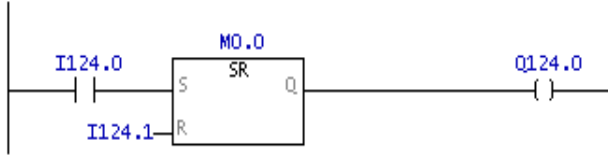


KURMA VEYA SİLME BASKIN İKİ KARARLI İŞLEM ELEMANLARI

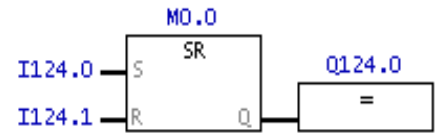
Kurma baskın iki kararlı (RS) elemanın her iki girişi de 1 yapıldığında çıkışı 1, silme baskın iki kararlı (SR) elemanın her iki girişi de 1 yapıldığında çıkışı 0 olur. Diğer giriş değerlerinde silme kurma komutları ile eşdeğerdir.

RESET BASKIN

Network 1 : Title:

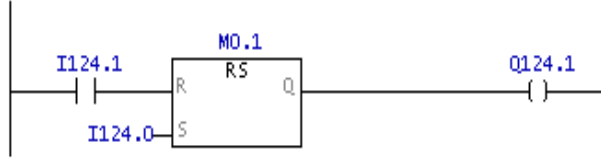


Network 1 : Title:

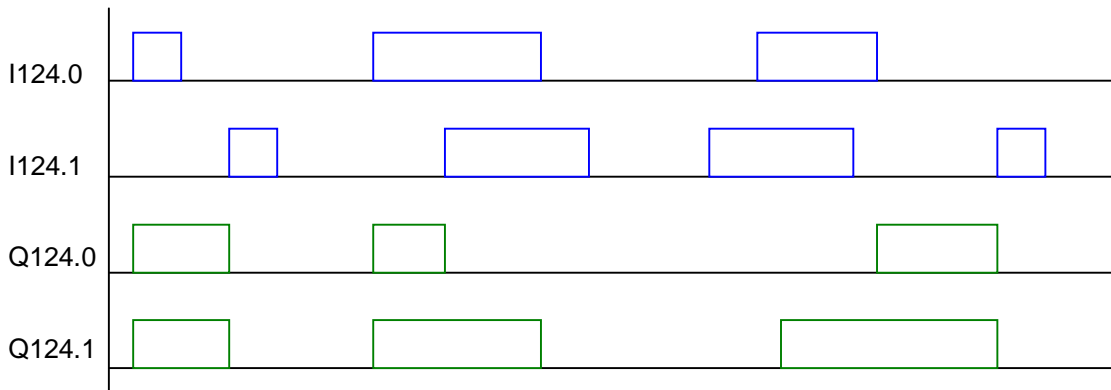
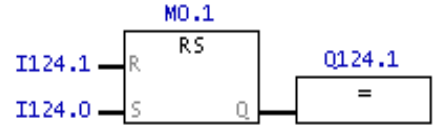


SET BASKIN

Network 2 : Title:

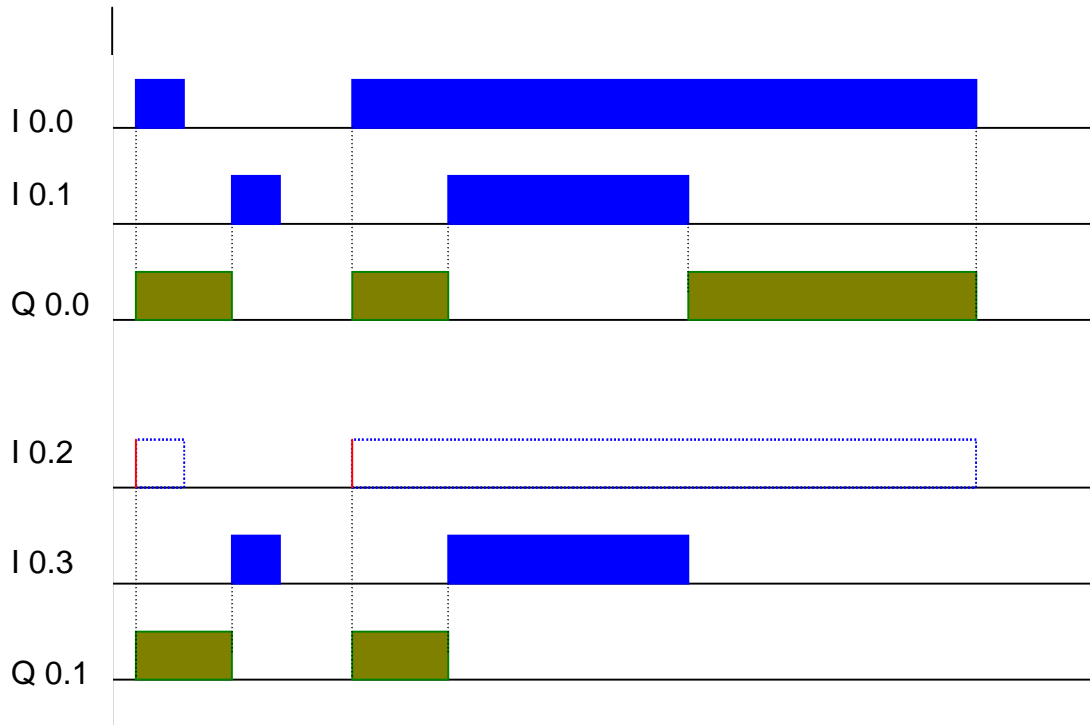
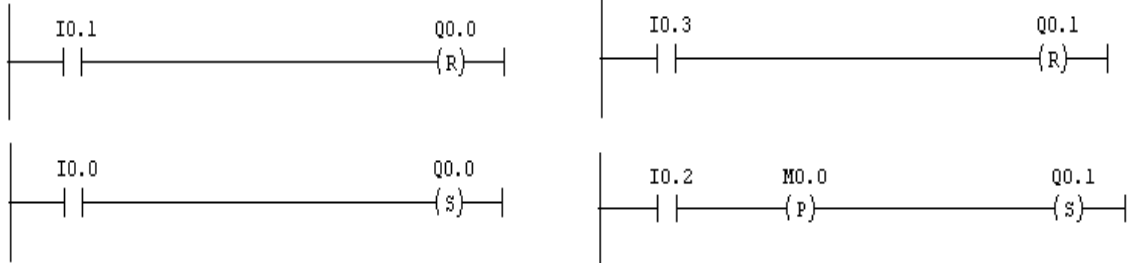


Network 2 : Title:

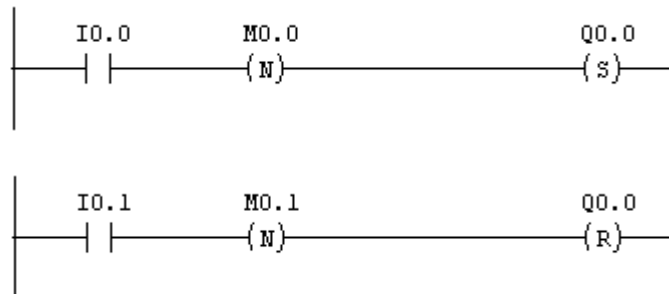


ÇIKAN KENAR VE DÜŞEN KENAR ALGILAMA KOMUTLARI

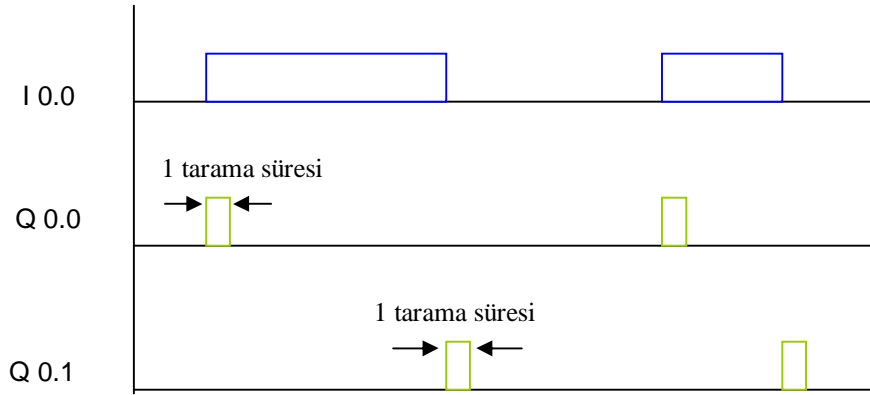
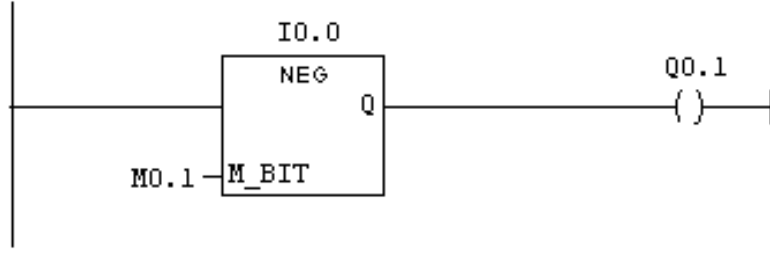
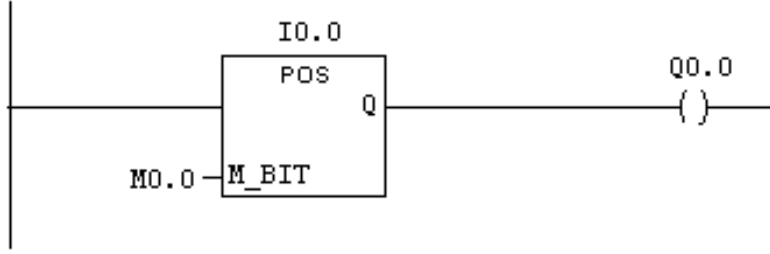
POZİTİF KENAR ALGILAMA



NEGATİF KENAR ALGILAMA

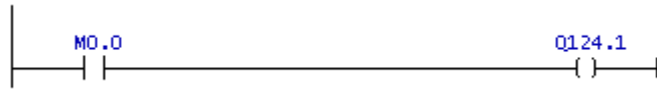
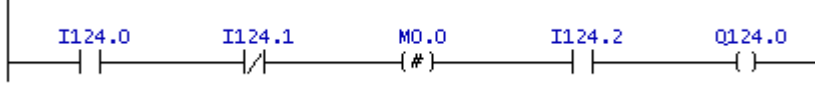


2. İşaret Kenar Değerlendirmesi



Lojik İfadenin İçeriğini Bir Hafıza Alanında Saklama

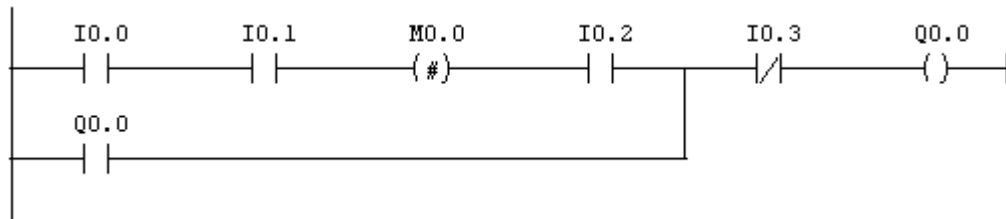
İkili mantıksal işlemlerin ara sonuçlarını ikili sonuç komutu kullanmadan saklamaya yarayan komuttur. Bu şekilde aynı ikili alt işlemi bir alt satırda yeniden yapılması gerekmez.



PROBLEM:

1. motorun çalışma şartı 1.2. ve 3. start butonlarına aynı anda basılmasıdır. Motor sürekli çalışacaktır. Çalışma stop butonu ile sonlanmaktadır.

2. motor ise sadece 1.ve 2. butonlara basılarak çalışmakta ve butonlardan birine basılmaması durumunda 2. motor durmaktadır.



SAYI SİSTEMLERİ

A) ONLU (DECİMAL)SAYI SİSTEMİ

Sayı tabanı 10'dur. 0,1,2,3,4,5,6,7,8,9 sayıları onluk sayı sistemini oluşturur.

Herhangi bir sayıyı formülize edersek:

$$D = 1985 = 1.10^3 + 9.10^2 + 8.10^1 + 5.10^0$$

B) İKİLİ (BİNARY-DUAL) SAYI SİSTEMİ

'0' ve '1' rakamları ile temsil edilen, taban değeri '2' olan ve iki olasılıklı durumları ifade etmek amacıyla kullanılan sayı sistemidir. İkili sayı sisteminde her bir basamak 'bit' olarak (**B**inayr **D**igit) adlandırılır.

$$B = 101101101$$

MSB LSB

İkili sayı sisteminde en sağdaki basamağa "en düşük anlamlı bit"-DAB (Least Significant Bit-LSB), En soldaki basamağa da "en yüksek anlamlı bit"-YAB (Most Significant Bit-MSB) denir

C) ONALTILIK (HEXADECİMAL) SAYI SİSTEMİ

İkili sayı sistemlerinde çok fazla basamak kullanılır. Buda hatayı ve işlem süresini uzatır. İkili sayı sisteminin daha kolay gösterilmesini sağlamak için onaltılık sayı sistemi kullanılır. Onaltılık sayı sisteminde 0 ile 9 arasındaki rakamlar ile A,B,C,D,E,F harfleri kullanılmaktadır.

SAYISAL DÖNÜŞÜMLER

| ONLU | İKİLİ | ONALTILI |
|------|-------|----------|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | A |
| 11 | 1011 | B |
| 12 | 1100 | C |
| 13 | 1101 | D |
| 14 | 1110 | E |
| 15 | 1111 | F |

ONLU SAYI SİSTEMİNDEKİ SAYILARIN İKİLİ VE ONALTILI SAYI SİSTEMLERİNE DÖNÜŞÜMÜ

Örnek:

$(39)_{10}$ sayısının ikili sayı sistemine çevirelim

| Bölünen | Bölüm | Kalan | |
|---------|-------|-------|-----|
| 39 / 2 | 19 | 1 | LSB |
| 19 / 2 | 9 | 1 | ↑ |
| 9 / 2 | 4 | 1 | ↑ |
| 4 / 2 | 2 | 0 | ↑ |
| 2 / 2 | 1 | 0 | MSB |

$$\begin{array}{l} \text{└─} \\ \text{└─} \\ \text{└─} \\ \text{└─} \\ \text{└─} \end{array} \rightarrow 100111 \quad (39)_{10} = (100111)_2$$

Örnek :

$(423)_{10} = (?)_{16}$

| | | | |
|----------|----|----|-----|
| 423 / 16 | 26 | 7 | |
| 26 / 16 | 1 | 10 | → A |
| 1 | | 1 | → 1 |

$(423)_{10} = (1A7)_{16}$

İKİLİ SAYI SİSTEMİNDEKİ SAYILARIN ONLU SAYILARIN ONLU VE ONALTILI SAYI SİSTEMLERİNE DÖNÜŞÜMÜ

Örnek :

$(11001)_2 = (?)_{10}$

$$= 1.2^4 + 1.2^3 + 0.2^2 + 0.2^1 + 1.2^0$$

$$= 16 + 8 + 0 + 0 + 1$$

$$= 25$$

$$(11001)_2 = (25)_{10}$$

Örnek :

$(10111101110000111101)_2 = (?)_{16}$

NOT: İkili sayı sisteminden 16'lık sayı sistemine geçilirken sayılar sağdan başlanarak dörderli gruplara ayrılır. Eğer son grup eksik kalırsa başına sıfır eklenir.

1011 1101 1100 0011 1101

B D C 3 D

$$(10111101110000111101)_2 = (BDC3D)_{16}$$

KODLAMA VE KODLAR

Görülebilien okunabilien yazı, sayı, ve işaretlerin değıştirilmesi işleme “kodlama” denir. Sayısal karakterlerin kodlanmasına “sayısal kodlama” (BCD kodları) denilirken , alfabetik ve sayısal karakterlerin kodlanmasını içeren kodlama yöntemlerine “alfasayısal kodlar” denilir.

Sayısal Kodlar

- BCD Kodu
- Gray Kodu

BCD KODU (BİNARY CODED DECİMAL CODE) - 8421 KODU

10'luk sistemdeki bir sayının, her bir basamağının ikilik sayı sistemindeki karşılığının yazılması ile ortaya çıkan kodlama yöntemine denir.

Örnek:

(263)₁₀ sayısını BCD kodu ile kodlayalım.

$$\begin{array}{ccc} 2 & 6 & 3 \\ 0010 & 0110 & 0011 \end{array} \quad (263)_{10} = (001001100011)_{BCD}$$

Örnek :

(100110010010)_{BCD} sayısını onlu sisteme çevirelim.

$$\begin{array}{ccc} 1001 & 1001 & 0010 \\ 9 & 9 & 2 \end{array} \quad (100110010010)_{BCD} = (992)_{10}$$

ZAMANLAMA İŞLEMLERİ

5 tip zamanlayıcı mevcuttur.

- S_ODT(SD) Gecikmeli Zamanlayıcı
- S_ODTD(SS) Gecikmeli ve Mühürlemeli Zamanlayıcı
- S_OFFDT(SF) Düşen Kenara Göre Gecikmeli Zamanlayıcı
- S_PULSE(SP) Darbe Zamanlı Zamanlayıcı
- S_PEXT(SE)Uzatılmış Darbe Zamanlı Zamanlayıcı

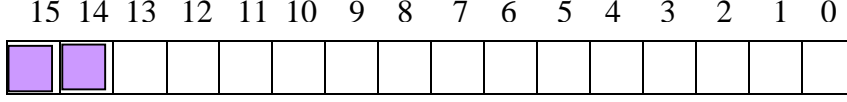
S7 300 ailesine ait CPU'larda S7-200'lerden farklı olarak CPU'daki zamanlayıcılar, zamanlayıcı tiplerine göre belli sabit sayılarda olacak şekilde ayrılmamıştır. Projedeki ihtiyaca bağlı olarak istenilen tipte zamanlayıcı CPU'nun izin verdiği zamanlayıcı sayısını aşmamak şartıyla istenildiği kadar kullanılabilir.

- Bunlardan dördü yükselen kenar, bir türü ise düşen kenar ile tetiklenir.
- Zamanlayıcılar tetikleme prensibine göre çalışırlar ve içerikleri belirlenen zamandan 0'a doğru geriye akar.
- Her yeni gelen tetikleme sinyali ile zamanlayıcı içeriğine belirlenen zaman yeniden yüklenir ve yeniden geriye doğru akmaya başlar.
- Her bir zamanlayıcı için CPU'nun hafızasında 16 bitlik bir bellek adresi ayrılmıştır.
- Zamanlayıcı adedi CPU'nun tipine bağlıdır.

| S7 300 CPU Tipi | Zamanlayıcı Adedi |
|------------------------|--------------------------|
| CPU 312 | 0...63 |
| CPU 313 | 0...127 |
| CPU 314 | 0...127 |
| CPU 315 | 0...127 |
| CPU 315-2DP | 0...127 |
| CPU 318-2DP | 0...511 |

ZAMANLAYICI DEĞERİ(TV)

Her bir zamanlayıcı için CPU'nun hafızasında 16 bitlik bir bellek adresi ayrılmıştır.



Zamanlayıcı için ayrılmış 16 bitlik adres alanında 0 ile 11 no.lu bitler arasına BCD formatında zaman değeri, 2 ve 13 no.lu bitlere ise Zaman Çarpanı(Zaman Tabanı) yazılır. 14 ve 15 no.lu bitler ise kullanılmamaktadır. Buna göre zamanlayıcılar ile 10 ms.'den 2 saat 46 dakika ve 30 saniyeye kadar bir süre için gecikme yaratılabilir.

| 13.BİT | 12.BİT | ZAMAN ÇARPANI |
|--------|--------|---------------|
| 0 | 0 | 0.01sn |
| 0 | 1 | 0.1sn |
| 1 | 0 | 1sn |
| 1 | 1 | 10sn |

Zamanlayıcı değerini iki farklı şekilde atamak mümkündür

1. Hexadecimal formatta atama yapma

W#16#klmn

k : Zaman çarpanı

lmn : BCD formatında zaman değeri

1 saat,10 dakika ve 20 saniye için TV yerine W#16#3422
7 saniye 20 milisaniye için TV değeri olarak W#16#0702 y

2. S5 Time formatında atama yapma

S5T#wH_xM_yS_zMS

H: Saat **M:** Dakika **S:** Saniye **MS:** Milisaniye

1 saat,10 dakika ve 20 saniye için TV yerine S5T#1H10M20S yazılmalıdır.
7 saniye 20 milisaniye için TV değeri olarak S5T#7S20MS yazılmalıdır

1. S_ODT(SD) – GECİKMELİ ZAMANLAYICI

Gecikmeli Zamanlayıcı, Kurma(S) girişinin yükselen kenarı ile zaman geriye doğru akar ve öngörülen TV değeri dolduğunda Q çıkışı 1 olur. Kurma girişi 0 olana kadar(R=0 koşulunda) veya Silme(R) girişinin 1 olmasıyla çıkış 1 olmaya devam eder. Her zaman Silme girişinin önceliği vardır.

Geçerli Adresler

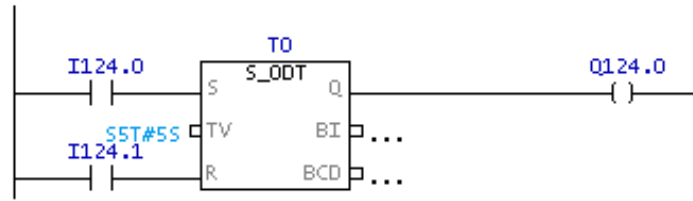
S(bit) : I, Q, M, D, L, T, C

TV : I, Q, M, D, L veya sabit

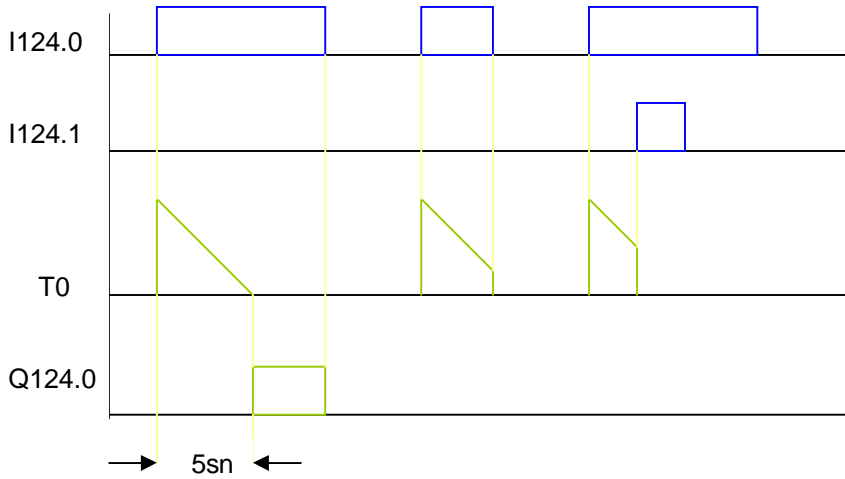
R(bit): I, Q, M, D, L, T, C

T....Timer

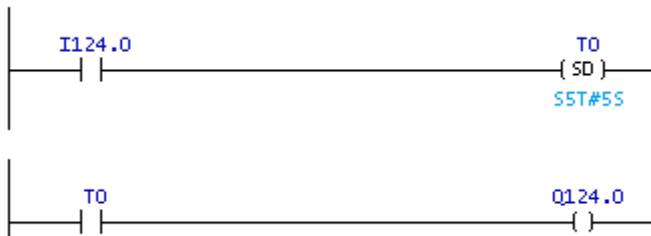
C....Counter



Zaman Grafiği



SD



2. S_ODTS(SS) – GECİKMELİ VE MÜHÜRLEMELİ ZAMANLAYICI

Gecikmeli ve Mühürlemeli Zamanlayıcı, Kurma(S) girişinin yükselen kenarı ile zaman geriye doğru akar ve öngörülen TV değeri dolduğunda Q çıkışı 1 olur.

Kurma girişi 0 olsa bile zaman geriye doğru akmaya devam eder. Öngörülen TV değeri dolduğunda Q çıkışı mühürlenir ve kurma girişinden yeni bir tetiklemeye kapanır. Çıkış silme(R) girişine 1 sinyali gelene kadar 1 olmaya devam eder.

Silme(R) girişinin önceliği vardır.

Geçerli Adresler

S(bit) : I, Q, M, D, L, T, C

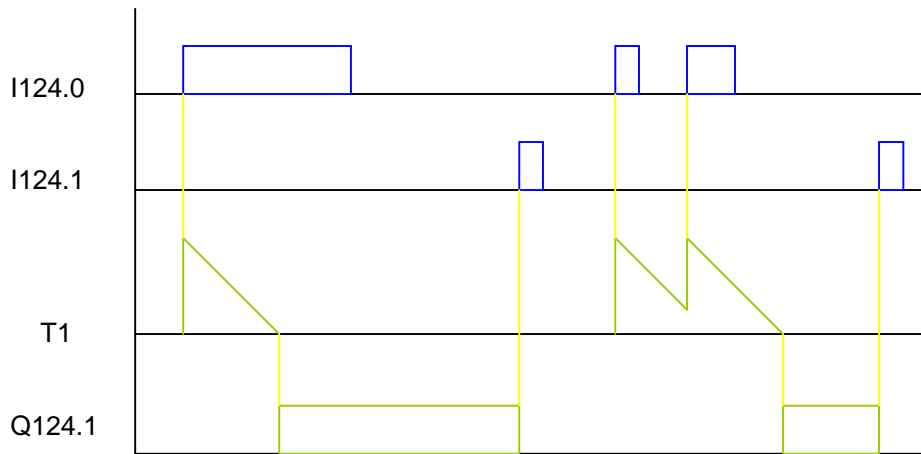
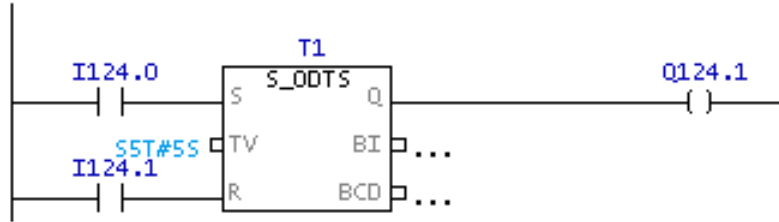
TV : I, Q, M, D, L veya sabit

R(bit) : I, Q, M, D, L, T, C

Q(bit) : I, Q, M, L, D

BI(Int) : I, Q, M, D, L

BI_BCD(Int) : I, Q, M, D, L



3. S_OFFDT(SF) – DÜŞEN KENARA GÖRE GECİKMELİ ZAMANLAYICI

Düşen Kenara Göre Gecikmeli Zamanlayıcının Kurma(S) girişinin yükselen kenarı ile Q çıkışı 1 olur, düşen kenarı ile zaman geriye doğru akmaya başlar ve öngörülen zaman değeri dolduğunda Q çıkışı 0 olur. Zamanlayıcının içeriği 0 olmadan kurma girişi yeniden 1 olması halinde zamanlayıcının içeriği sabit kalır. Kurma girişi yeniden 0 olmasıyla da zamanlayıcı geriye doğru kaymaya devam eder.

Silme girişinin önceliği vardır.

Geçerli Adresler

S(bit) : I, Q, M, D, L, T, C

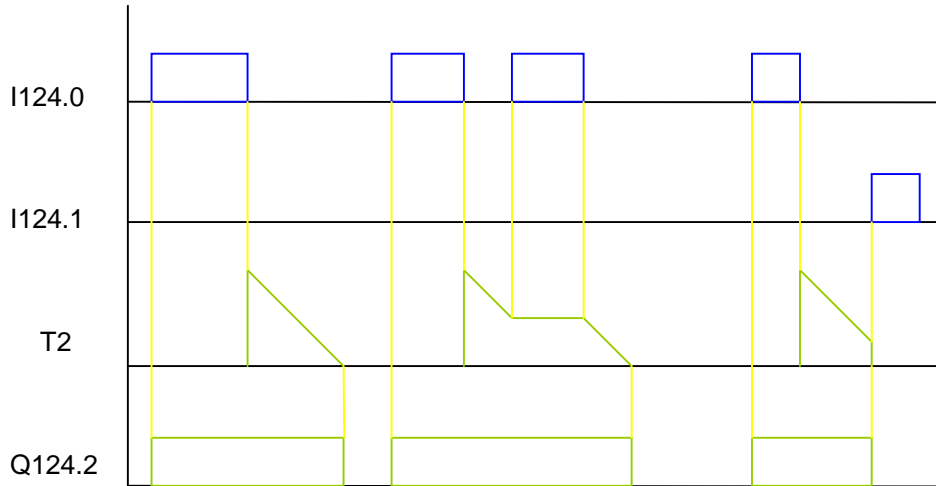
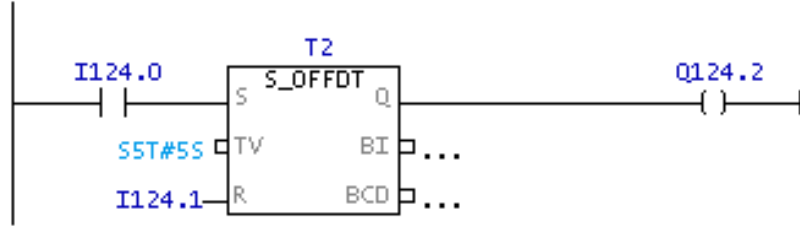
TV : I, Q, M, D, L veya sabit

R(bit): I, Q, M, D, L, T, C

Q(bit) : I, Q, M, L, D

BI(Int) : I, Q, M, D, L

BI_BCD(Int) : I, Q, M, D, L



4. S_PULSE(SP) – DARBE ZAMANLI ZAMANLAYICI

Darbe Zamanlı Zamanlayıcının Kurma(S) girişinin yükselen kenarı ile zaman geriye doğru akar ve Q çıkışı 1 olur. Öngörülen TV değeri dolduğunda yada kurma girişi 0 olduğunda Q çıkışı 0 olur. Zamanın geriye doğru akması için kurma girişinin 1 olması gerekir.

Silme girişinin önceliği vardır.

Geçerli Adresler

S(bit) : I, Q, M, D, L, T, C

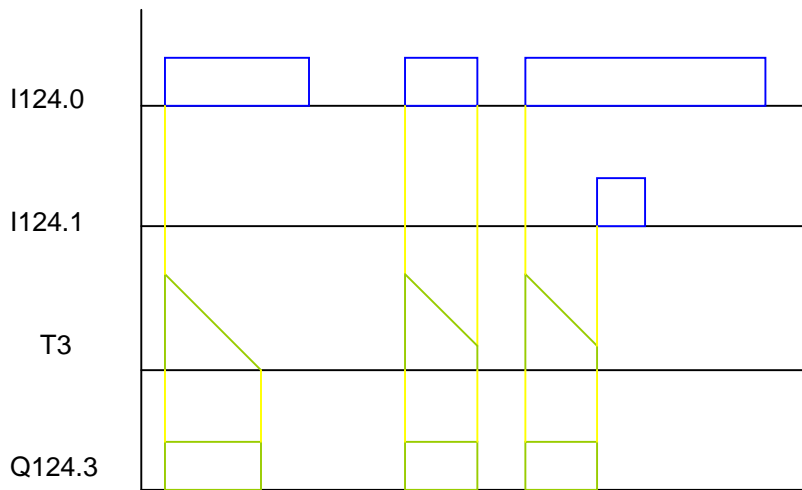
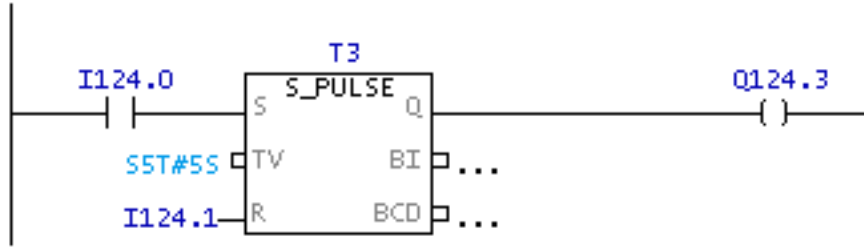
TV : I, Q, M, D, L veya sabit

R(bit): I, Q, M, D, L, T, C

Q(bit) : I, Q, M, L, D

BI(Int) : I, Q, M, D, L

BI_BCD(Int) : I, Q, M, D, L



5. S_PEXT(SE) – UZATILMIŞ DARBE ZAMANLI ZAMANLAYICI

Darbe Zamanlı Zamanlayıcının Kurma(S) girişinin yükselen kenarı ile zaman geriye doğru akar ve Q çıkışı 1 olur. Zamanlayıcı değeri(TV) dolduğunda yada kurma girişi 0 olduğunda Q=0 olur. S girişi 0'a düştüğünde hala zaman geriye doğru akmaya devam ediyor ise Q çıkışı öngörülen zamanın sonunda sıfıra düşer.

Geçerli Adresler

S(bit) : I, Q, M, D, L, T, C

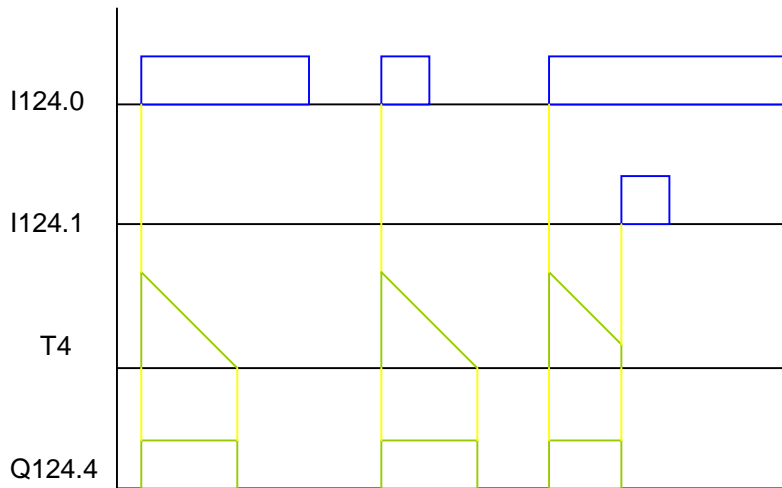
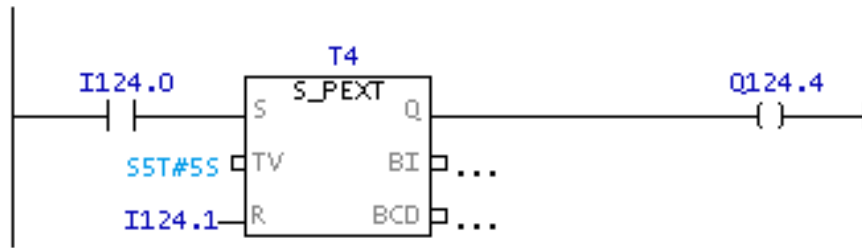
TV : I, Q, M, D, L veya sabit

R(bit): I, Q, M, D, L, T, C

Q(bit) : I, Q, M, L, D

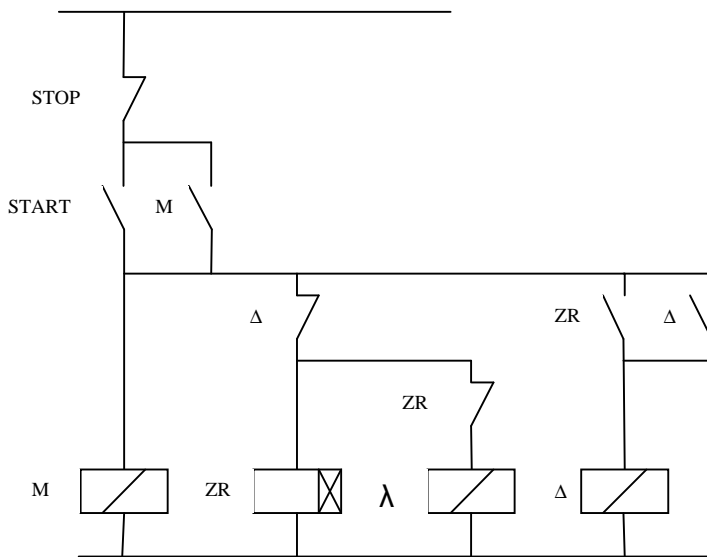
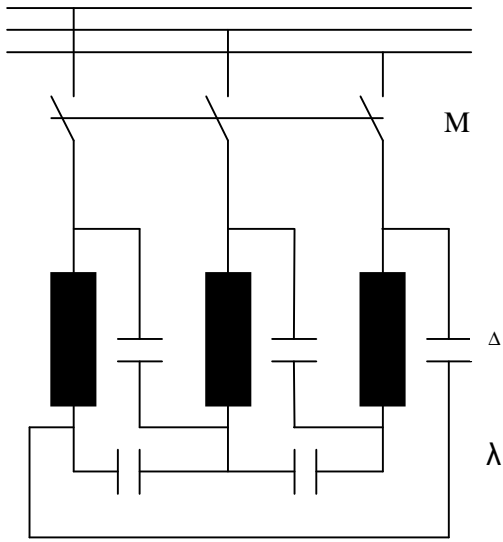
BI(Int) : I, Q, M, D, L

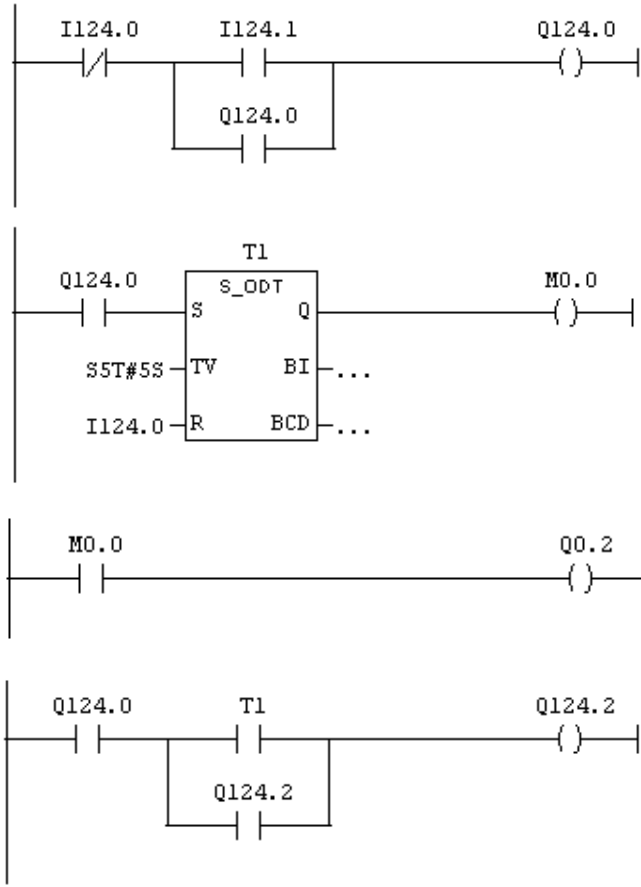
BI_BCD(Int) : I, Q, M, D, L



ÖRNEK :

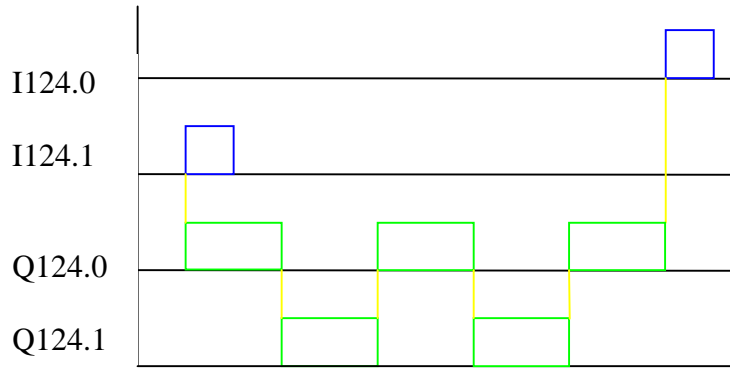
3 fazlı asenkron motorların gücü 5 kW'tan büyük olduğu zaman direk çalıştırılmazlar. İlk çalışma anında aşırı akım çekeceklerinden dolayı kalkınamazlar. Bu yüzden gücü 5 kW'tan büyük olan motorları farklı yöntemlerle kalkındırmak gerekir. En sık karşılaşılan yöntem yıldız üçgen yolvermedir. Motor yıldız çalıştırılarak üçgen çalışmaya oranla 1/3 oranında daha düşük akım çekerek, düşük momentle çalışmaya başlar. Ayarlanan süre sonunda üçgen çalışmaya geçilerek tam gerilim altında motorun çalışması sağlanır.





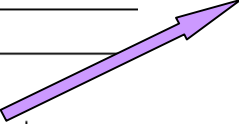
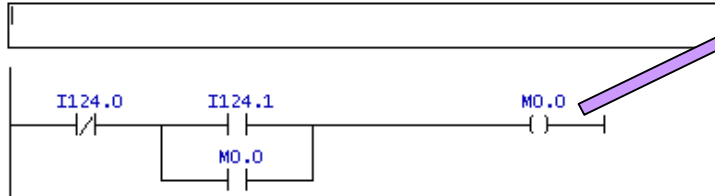
ÖRNEK :

Start butonuna basılınca kırmızı ışık yanacaktır. Kırmızı 5 sn sonra sönecek ve sarı ışık yanacak. 10sn sonra sarı sönecek ve tekrar kırmızı yanacak. 5 sn sonra tekrar sarı yanacak. Çalışma bu şekilde periyodik olarak devam edecektir. Çalışma stop butonu ile sonlanacaktır. İş akış diagramını çiziniz ve ladder digramını oluşturunuz



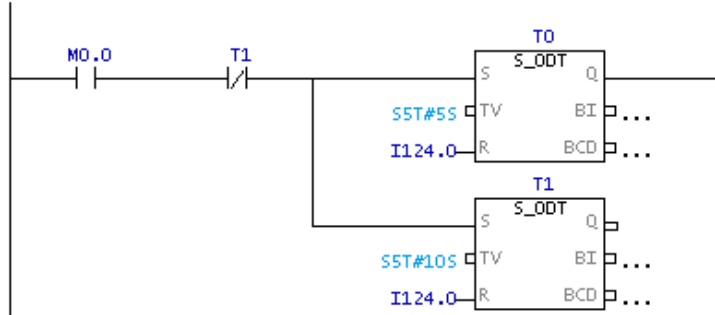
Merker –
Flag-
Bayrak
Rölesi

Network 1: Title:



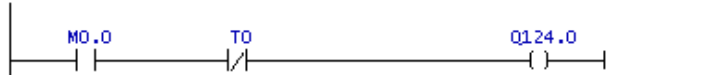
Network 2: Title:

Comment:



Network 3: Title:

Comment:



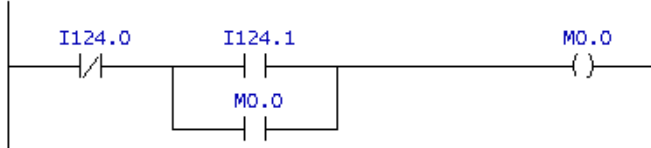
Network 4: Title:

Comment:

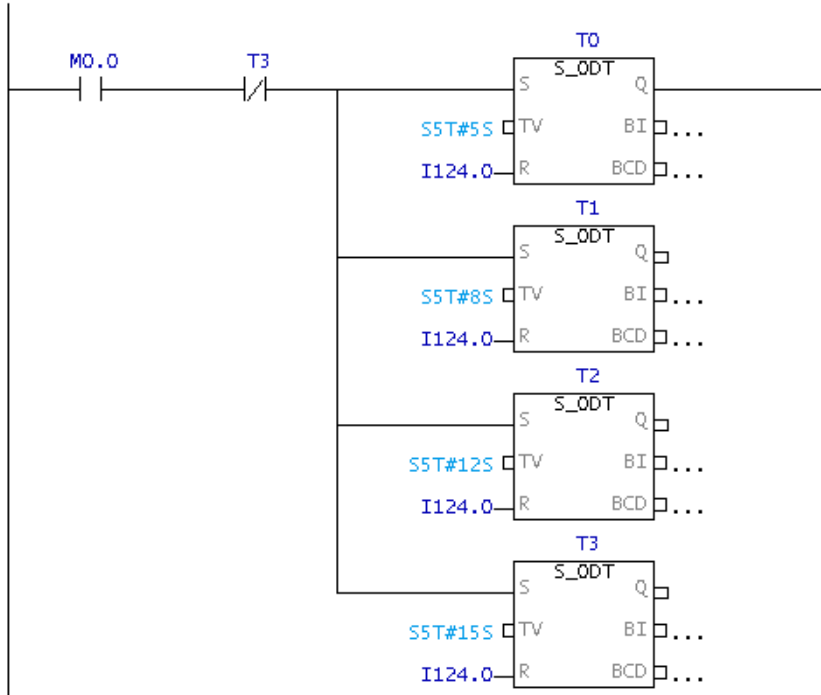


ÖRNEK :

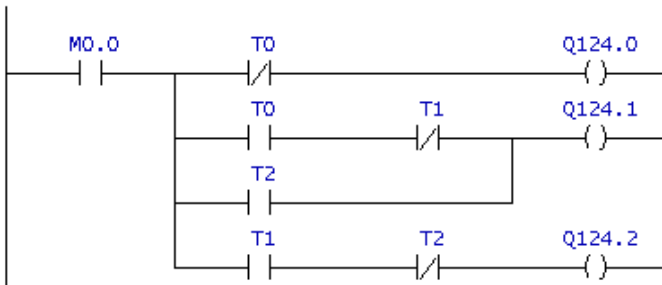
Start butonuna basılınca kırmızı ışık yanacaktır. Kırmızı 5 sn sonra sönecek ve sarı ışık yanacak. Sarı ışık 3 sn yanacak ve sönecek . Sarı sönünce yeşil 5 sn yanacaktır. Yeşil sönünce sarı 3 sn yanacak ve sönecektir. Sarı sönünce kırmızı yanacak ve çalışma periyodik olarak devam edecektir.Çalışma stop butonu ile sonlanacaktır.Ladder digramını oluşturunuz



Network 2 : Title:



Network 3 : Title:



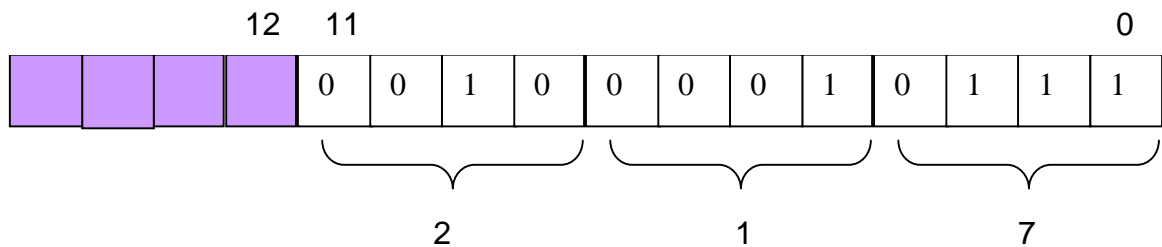
SAYMA İŞLEMLERİ

Genel Özellikler

- Sayma işlemleri için 3 farklı tür sayıcı kullanılır.
 - § İleri Sayıcı(CU)
 - § Geri Sayıcı(CD)
 - § İleri-Geri Sayıcı(CUD)
- Her bir sayıcı için CPU'nun hafızasında Sayıcı Değeri olarak adlandırılabilen 16 bitlik bir bellek adresi ayrılmıştır.
- Bir sayıcı 0'dan 999'a kadar sayabilir.
- Sayıcılar ileri yada geri sayma girişlerine gelen sinyalin çıkan kenarına göre içeriğini değiştirirler.

Sayıcı adedi CPU tipine bağlıdır.

| S7 300 CPU Tipi | Sayıcı Adedi |
|-----------------|--------------|
| CPU 312 | 0...128 |
| CPU 313 | 0...255 |
| CPU 314 | 0...255 |
| CPU 315 | 0...255 |
| CPU 315-2DP | 0...255 |
| CPU 318-2DP | 0...511 |



16 bitlik Sayıcı Deęeri adres alanında 0 ile 11 no.lu bitler arasına BCD formatında sayıcı içerięi kaydedilir. 12, 13, 14 ve 15 no.lu bitler ise kullanılmamaktadır.

Mesela sayıcının içerięi 217 deęeri kurulmak istendięinde PV girişine C#217 yazılmalıdır.

1. İLERİ SAYICI (CU)

İleri Sayıcı, *Sayıc İleri(CU)* girişine gelen işaretin çıkan kenarı ile 16 bitlik Sayıcı Deęerinin içerięini 1 arttırır. Sayıcı Deęerinin 0'dan farklı olması durumunda sayıcı Q *çıkışını* 1 yapar. *Kurma(S)* girişi gelen sinyalin çıkan kenarı ile de Sayıcı Deęerine *PV(Preset Value)* girişine yazılmış olan deęer aktarır. *Silme(R)* girişi ise dięer iki girişten farklı olarak kendisine baęlı olan sinyalin 1 olduęu süre boyunca etkin olur ve Sayıcı Deęerine 0 yazar. Sayıcı Deęeri tamsayı(Integer) formatında *BI çıkışına*, BCD formatında ise *BI_BCD çıkışına* aktarılır.

Geçerli Adresler

CU(bit):I, Q, M, D, L

S(bit) : I, Q, M, D, L

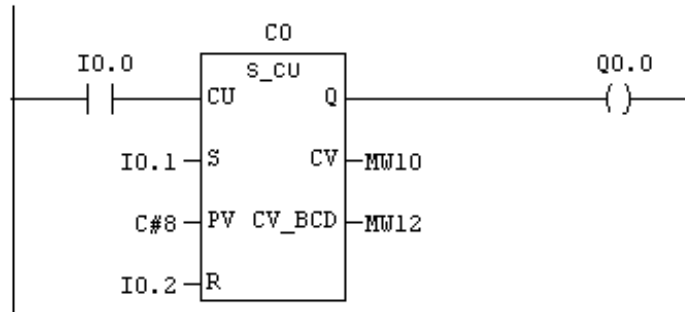
PV : I, Q, M, D, L veya sabit

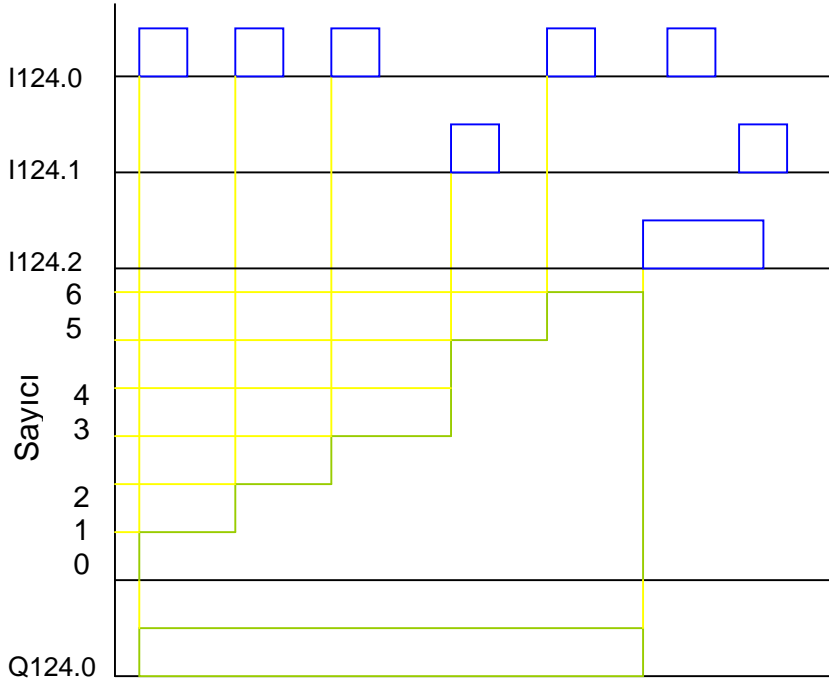
R(bit): I, Q, M, D, L

Q(bit) : I, Q, M, L, D

BI(Int) : I, Q, M, D, L

BI_BCD(Int) : I, Q, M, D, L





2. GERİ SAYICI(CD)

Geri Sayıcı, *Sayıcı Geri(CD)* girişine gelen işaretin çıkan kenarı ile 16 bitlik Sayıcı Değerinin içeriğini 1 azaltır. Sayıcı Değerinin 0'dan farklı olduğu sürece sayıcı Q çıkışını 1 yapar. *Kurma(S)* girişi gelen sinyalin çıkan kenarı ile de Sayıcı Değerine *PV(Preset Value)* girişine yazılmış olan değer aktarır. *Silme(R)* girişi ise diğer iki girişten farklı olarak kendisine bağlı olan sinyalin 1 olduğu süre boyunca etkin olur ve Sayıcı Değerine 0 yazar. Sayıcı Değeri tamsayı(Integer) formatında *BI çıkışına*, BCD formatında ise *BI_BCD çıkışına* aktarılır.

Geçerli Adresler

CU(bit): I, Q, M, D, L

S(bit) : I, Q, M, D, L

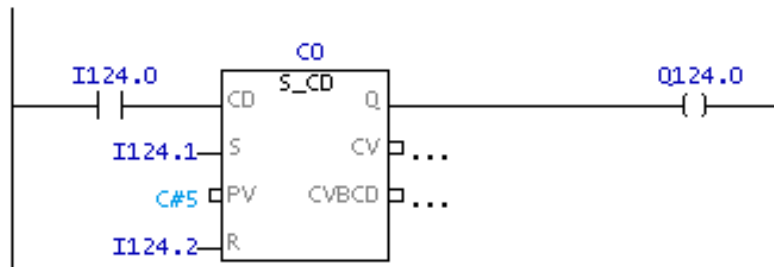
PV : I, Q, M, D, L veya sabit

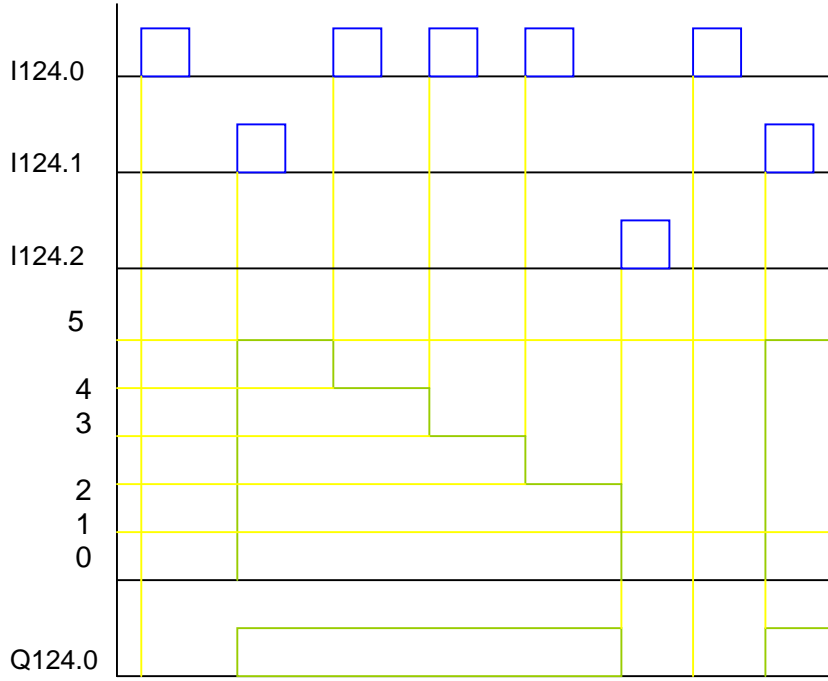
R(bit): I, Q, M, D, L

Q(bit) : I, Q, M, L, D

BI(Int) : I, Q, M, D, L

BI_BCD(Int) : I, Q, M, D, L



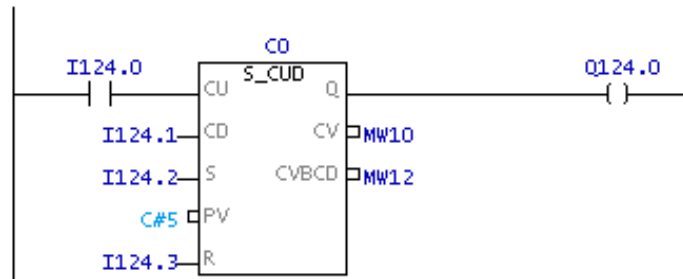


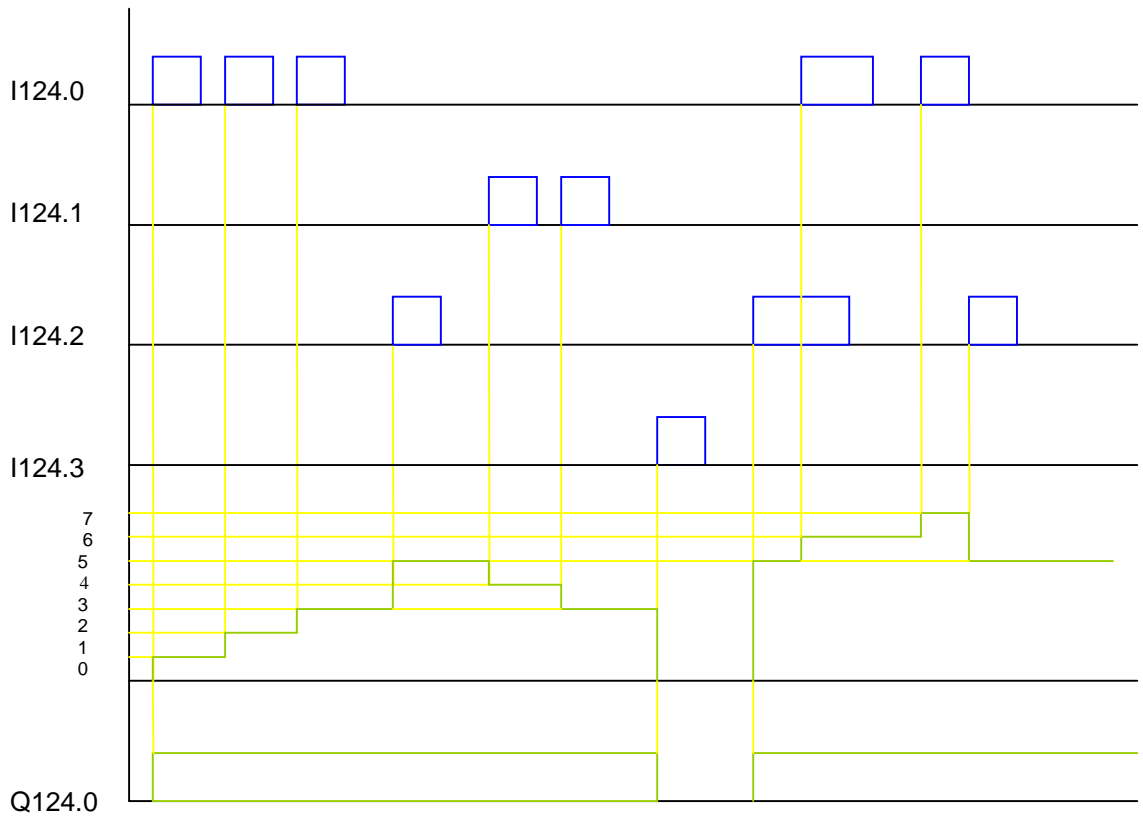
3. İLERİ-GERİ SAYICI (CUD)

İleri-Geri Sayıcı, *Sayıcı İleri(CU)* girişine gelen işaretin çıkan kenarı ile Sayıcı Değerinin içeriğini 1 arttırır, *Sayıcı Geri(CD)* girişine gelen sinyalin çıkan kenarı ile de Sayıcı Değerini 1 azaltır. Sayıcı Değerinin 0'dan farklı olması durumunda sayıcı *Q çıkışını* 1 yapar. *Kurma(S)* girişi gelen sinyalin çıkan kenarı ile de Sayıcı Değerine *PV(Preset Value)* girişine yazılmış olan değer aktarır. *Silme(R)* girişi ise diğer iki girişten farklı olarak kendisine bağlı olan sinyalin 1 olduğu süre boyunca etkin olur ve Sayıcı Değerine 0 yazar. Sayıcı Değeri tamsayı(Integer) formatında *BI çıkışına*, BCD formatında ise *BI_BCD çıkışına* aktarılır.

Geçerli Adresler

CU(bit): I, Q, M, D, L S(bit) : I, Q, M, D, L
 PV : I, Q, M, D, L veya sabit R(bit): I, Q, M, D, L
 Q(bit) : I, Q, M, L, D BI(Int) : I, Q, M, D, L
 BI_BCD(Int) : I, Q, M, D, L





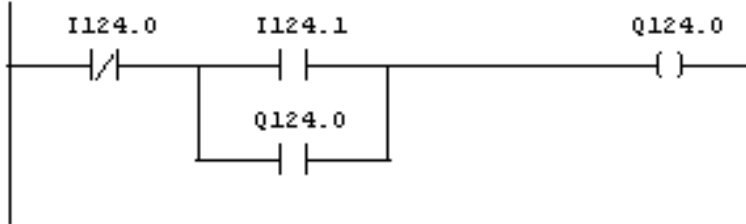
ÖRNEK :

FCL : Title:

3 FAZLI ASEKRON MOTOR TAŞIYICI BANDI ÇALIŞTIRMAKTADIR. TAŞIYICI BAND ÜZERİNDEN 5 KUTU GEÇİNCE DOLU BİLGİSİ VEREN KIRMIZI LAMBA YANACAKTIR.

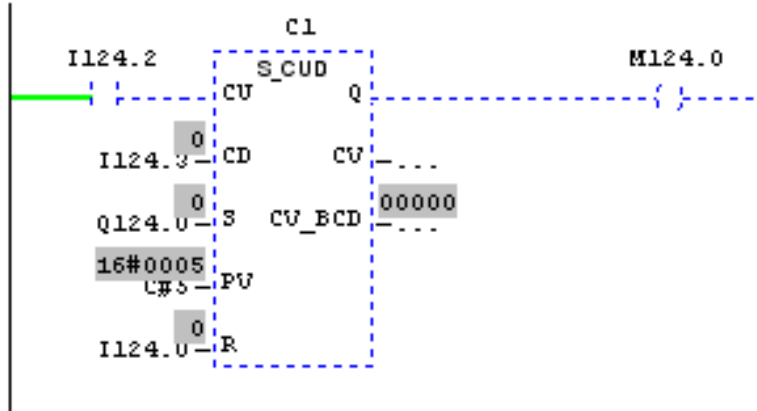
Network 1 : TAŞIYICI BANT

SİSTEMDE İLK ÇALIŞACAK MOTORDUR. MOTOR AYNI ZAMANDA DEVRENİN ÇALIŞMAŞARTINI DA OLUŞTURMAKTADIR.



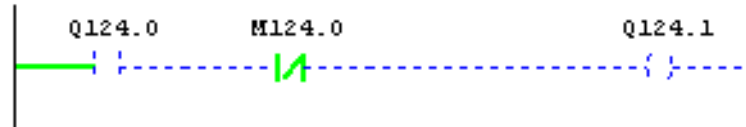
Network 2: SAYMA İŞLEMİ

MOTORUN ÇALIŞMASI SAYICIMIZI SET EDER. "CV" VEYA "SET" GİRİŞLERİNİN "1" OLMASINA BAĞLI OLARAK SAYICI SAYI İÇERİĞİ "PV" OLUR. SAYICI "Q" ÇIKIŞI AKTİF HALE GEÇER. "CV"VE "CD" GİRİŞLERİ SAYICI İÇERİĞİNİ BELİRLER. SAYICI İÇERİĞİ "0" OLDUĞUNDA ÇIKIŞ PASİF OLACAKTIR.



Network 3 : DOLU BİLGİSİ

SAYICI İÇERİĞİNİN SIFIR OLMASINA BAĞLI DOLU BİLGİSİ ELDE EDİLECEKTİR.



KARŞILAŞTIRMA KOMUTLARI

Genel Özellikleri

Çeşitli boyutlardaki veriler büyüklük, küçüklük veya eşitlik ölçütlerine göre karşılaştırma komutları kullanılarak değerlendirilir.

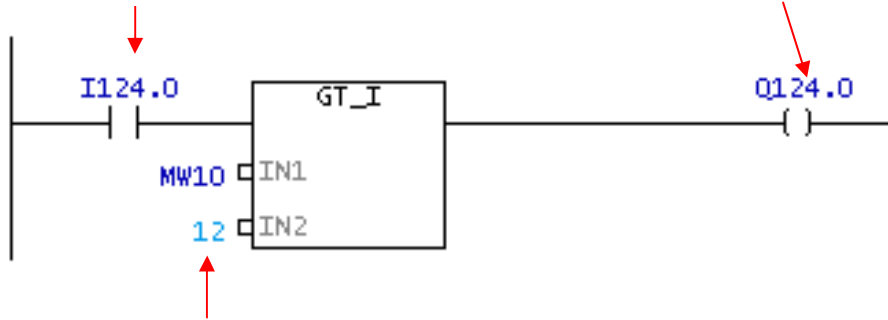
- Bu komutlar kullanılarak 16 bitlik tamsayı(INT), 32 bitlik tamsayı(DINT) ve 32 bitlik gerçek sayıları(FLOATING-POINT) karşılaştırılabilir.
- 6 farklı karşılaştırma yapmak mümkündür.

| Comparator | Equal | Eşit (==) |
|------------|-----------------------|-----------------|
| EQ_I | Equal | Eşit (==) |
| NE_I | Not equal | Eşit değil (<>) |
| GT_I | Greater Than | Büyük (>) |
| LT_I | Less Than | Küçük (<) |
| GE_I | Greater Than or Equal | Büyük eşit(>=) |
| LE_I | Less Than or Equal | Küçük eşit(<=) |

- Karşılaştırma işleminin sonucu olumlu ise Q çıkışı 1 olur.

Blok Girişi: Karşılaştırma komutunun hangi şartlarda çalışması isteniyorsa onunla ilgili mantıksal ifade bu kısma yazılır.

Blok Çıkışı: Karşılaştırma işleminin olumlu sonuçlanması halinde yapılacak olan işler bu kısma yazılır.



IN1 ve IN2: Karşılaştırılmak istenilen sayısal değerler bu iki girişe yazılır. Burası uygun bir hafıza alanı yada sabit bir değer olabilir.

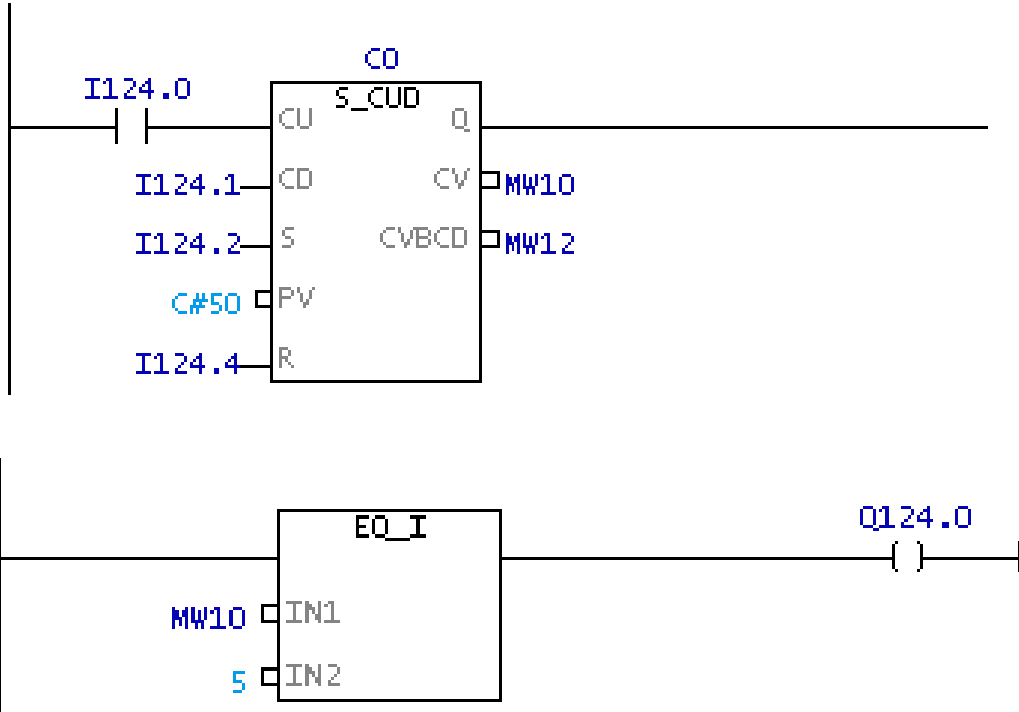
Geçerli Adresler

Blok Girişi(bit): I, Q, M, D, L
IN1 : I, Q, M, D, L veya sabit
IN2 : I, Q, M, D, L veya sabit

Blok Çıkışı(bit) : I, Q, M, L, D

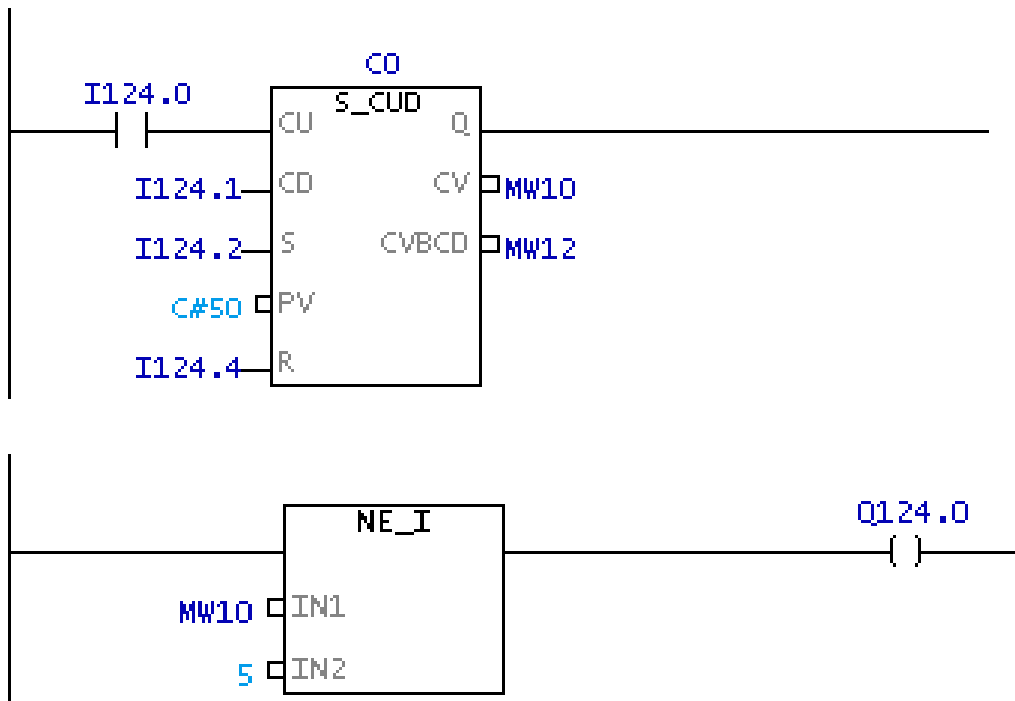
1. EŞİT-EQUAL (EQ_I)

IN1 = IN2 ise Q=1 olur



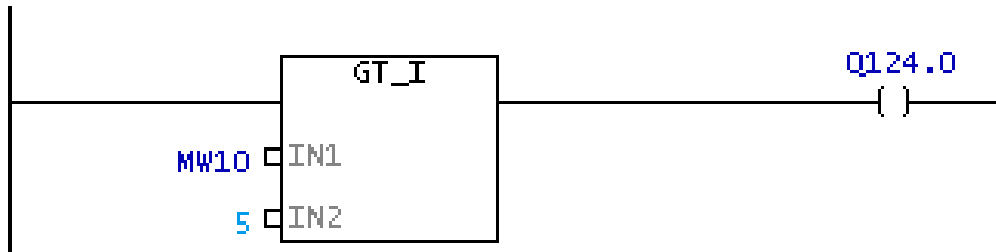
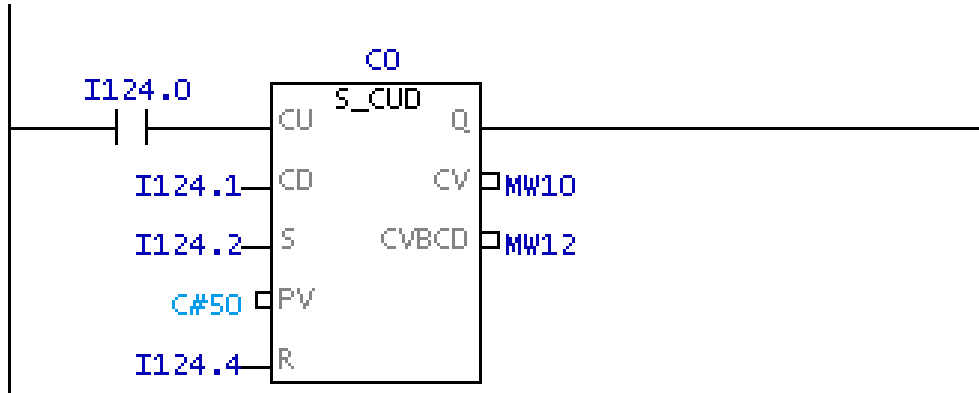
2. EŞİT DEĞİL – NOT EQUAL (NE_I)

IN1 ≠ IN2 ise Q=1 olur



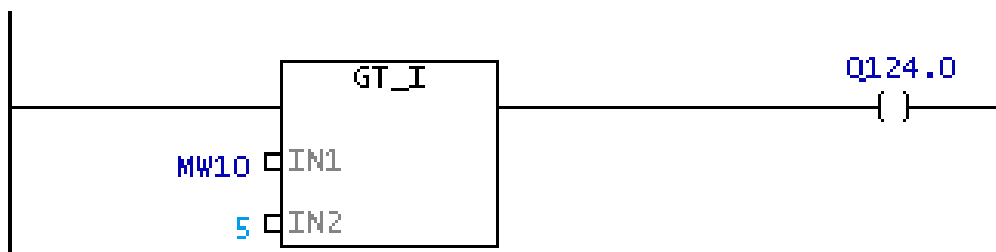
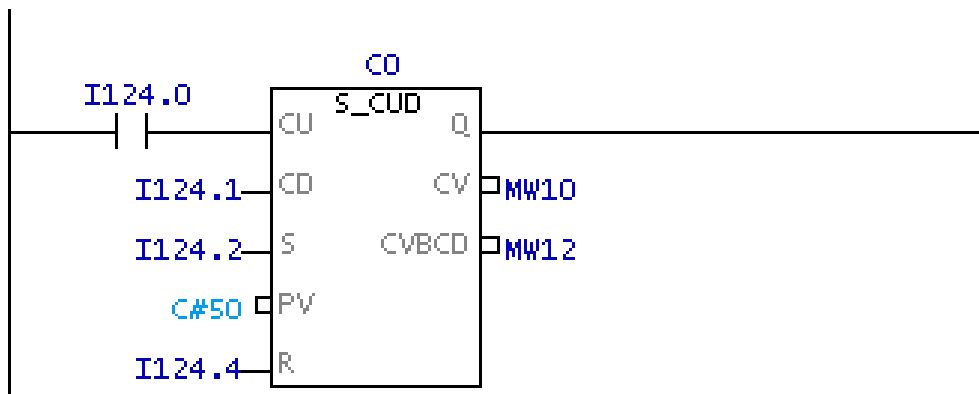
3. BÜYÜK – GREATER THAN (GT_I)

IN1 > IN2 ise Q=1 olur



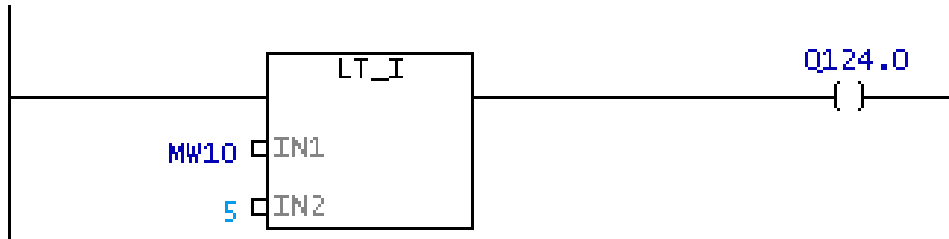
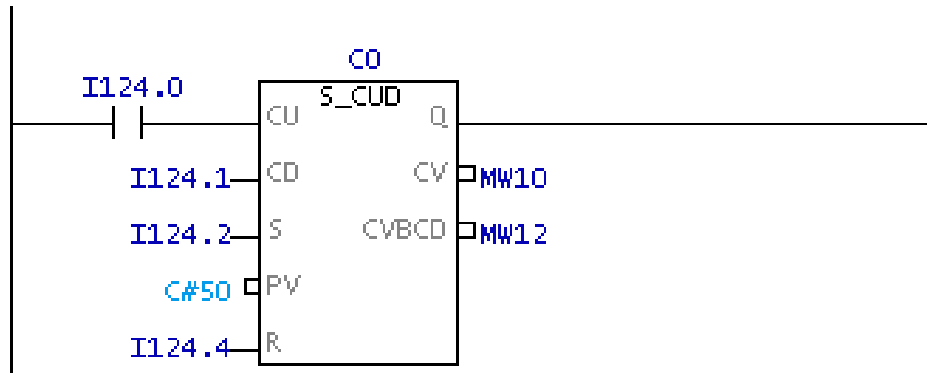
4. BÜYÜK EŞİT – GREATER THAN OR EQUAL (GE_I)

IN1 >= IN2 ise Q=1 olur



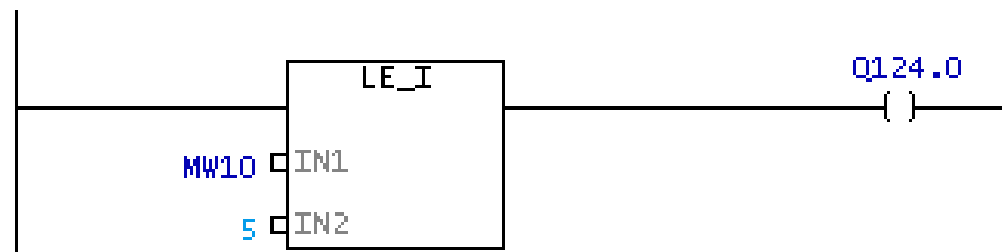
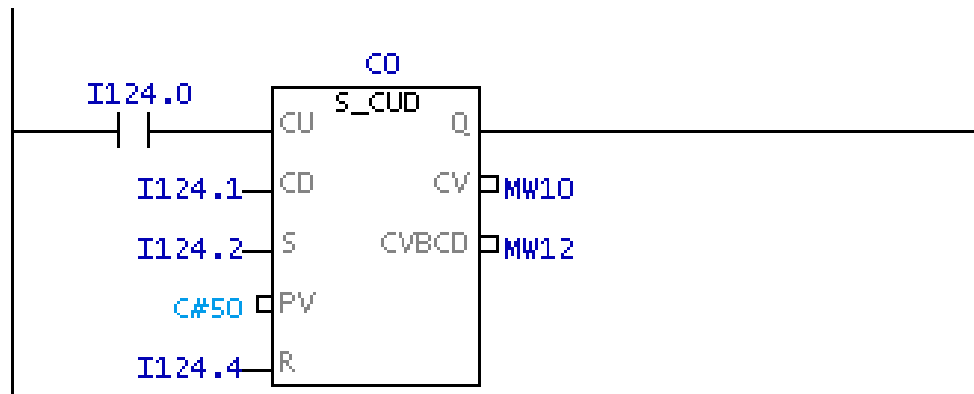
5. KÜÇÜK- LESS THAN (LT_I)

$IN1 < IN2$ ise $Q=1$ olur



6. KÜÇÜK EŞİT- LESS THAN OR EQUAL (LE_I)

$IN1 \leq IN2$ ise $Q=1$ olur

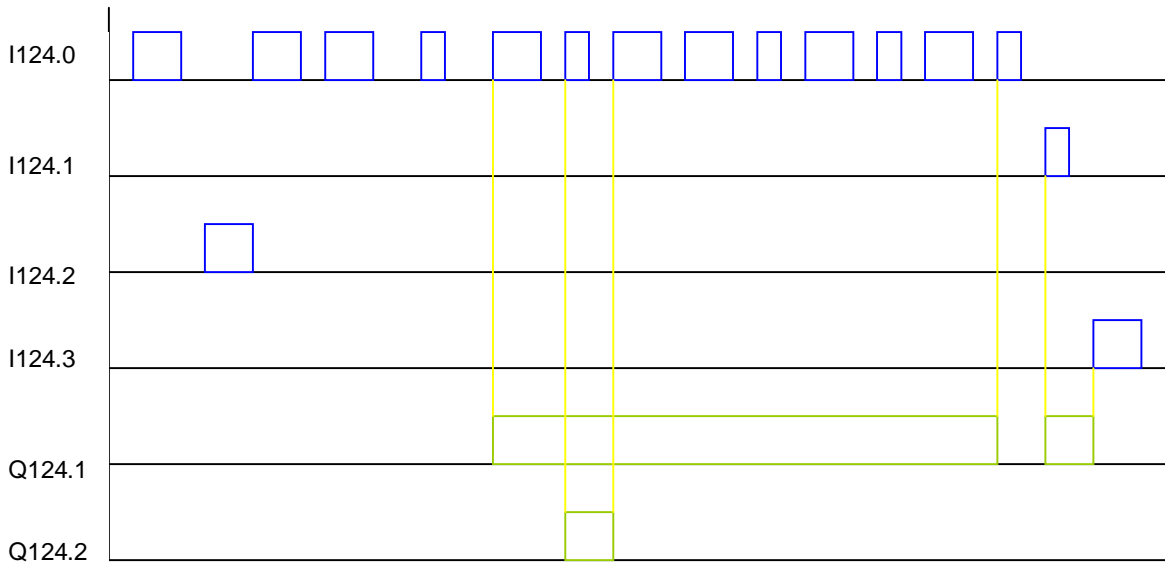
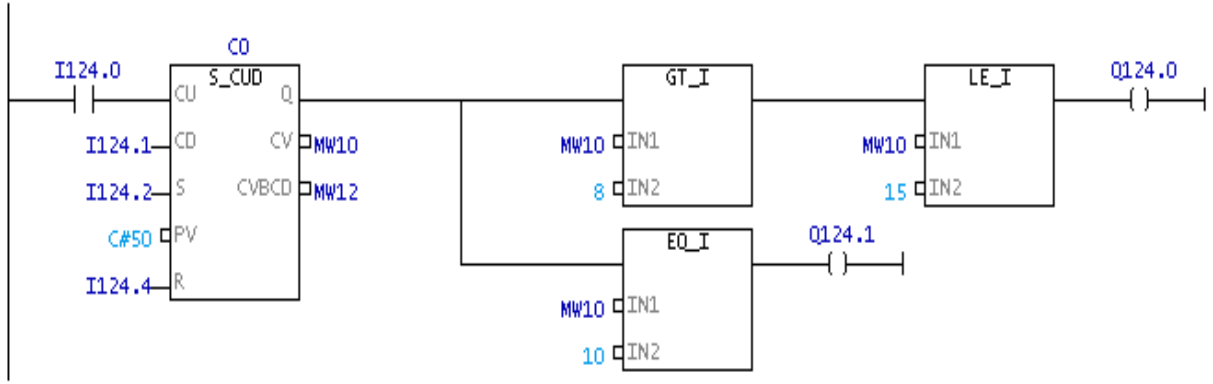


KARŞILAŞTIRMA KOMUTLARININ SAYICILARLA BERABER KULLANILMASI

S7 300 ailesine ait olan CPU'lardaki sayıcılar S7 200 ailesine ait CPU modellerindeki sayıcılara göre çalışma prensibi olarak farklılık göstermektedirler. S7 200 ailesine ait CPU'larda sayıcılar hem sayma işlemini gerçekleştirirken hem de karşılaştırma işlemini gerçekleştirmektedirler. Öyle ki sayıcının değeri PV girişine girilen değere eşit yada büyük ise sayıcının Q çıkışı lojik 1 değerini almaktadır.

Fakat S7 300'lerde sayıcı çıkışı sayıcı değerinin sıfırdan farklı olması durumuna bağlıdır. Sayıcının içeriğinin S7 200'lerdeki gibi karşılaştırılması söz konusu değildir

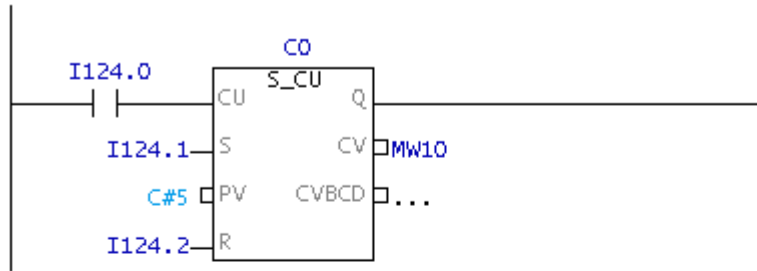
Bu bağlamda sayıcıları karşılaştırma komutları ile beraber kullanmak gerekir.



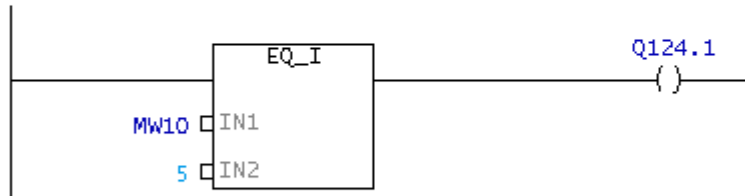
ÖRNEK :

Bir turnikenin 5. bilgisi sayısında kırmızı lamba yanması, 7'den büyük ve 12'den küçük değerler için ise siren ötürmesi isteniyor. Ladder diagramı çiziniz.

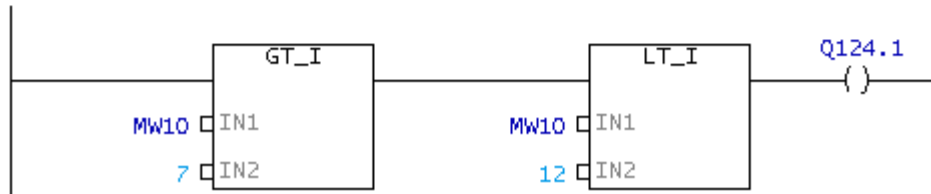
Network 1 : Title:



Network 2 : Title:



Network 3 : Title:



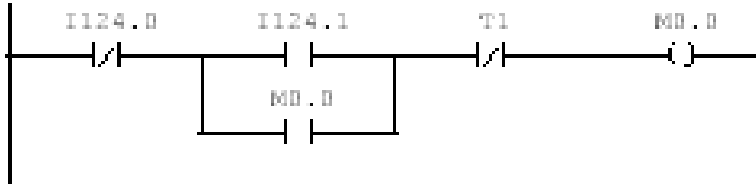
ÖRNEK :

PC1 : SAYICI VE ZAMANLAYICI

STARTA BUTONUNA BASILINCA TAŞIYICI BANT ÇALIŞIYOR. BANT ÜZERİNDEKİ ÜRÜNLER SENSÖR TARAFINDAN SAYILIYOR.20 ADET ÜRÜN OLUNCA KONVEYÖR BANT DURUYOR VE DOLU BİLGİSİ VEREN LAMBA YANIYOR. LAMBA 15 SANİYE YANDIKTAN SONRA TÜM SİSTEM DURUYOR.

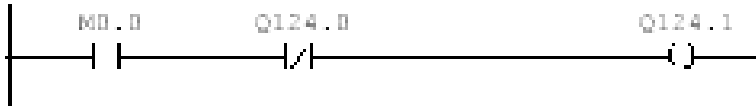
Network 1 : Title:

STARTA BASILDIĞINDA MB.0 EMERJİLENİR. TÜM DEVRENİN EMERJİLİ OLMASI SAĞLANIR



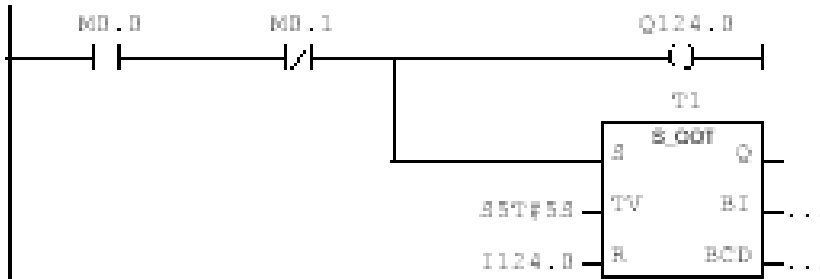
Network 2 : Title:

Comment:



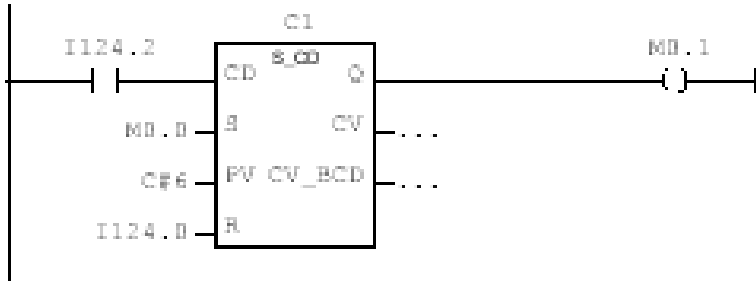
Network 3 : Title:

Comment:



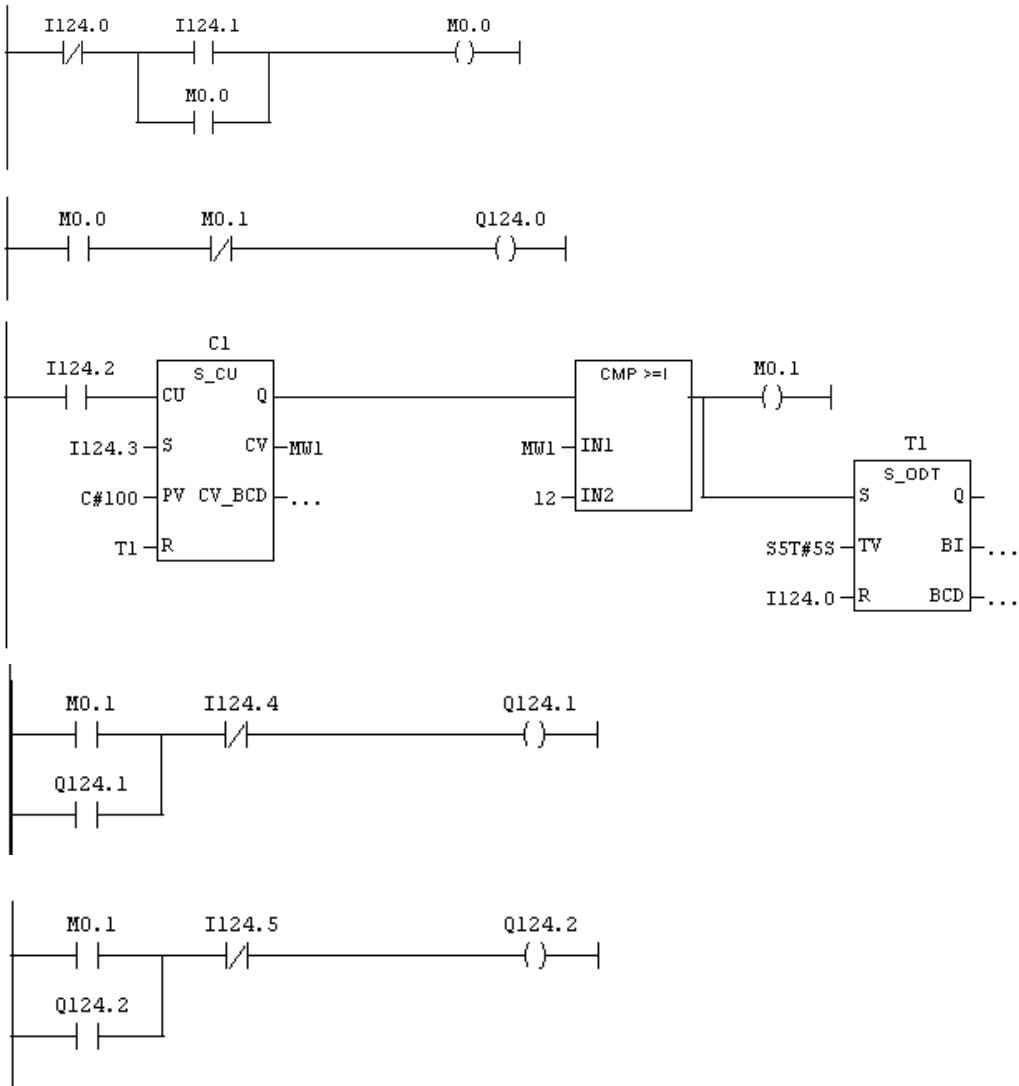
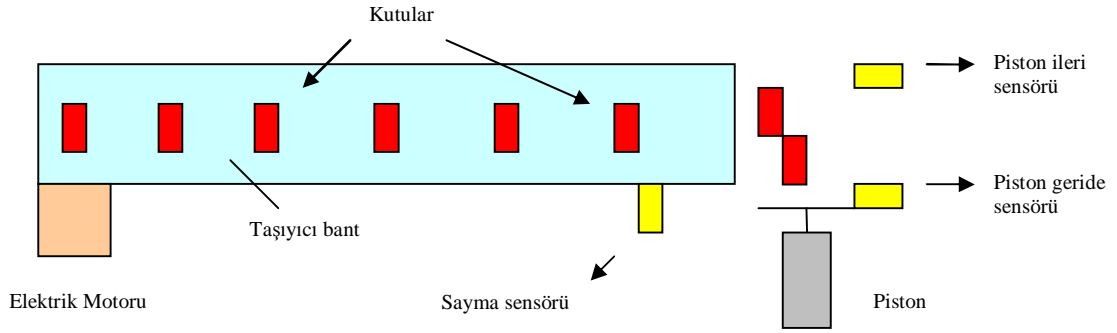
Network 4 : Title:

Comment:

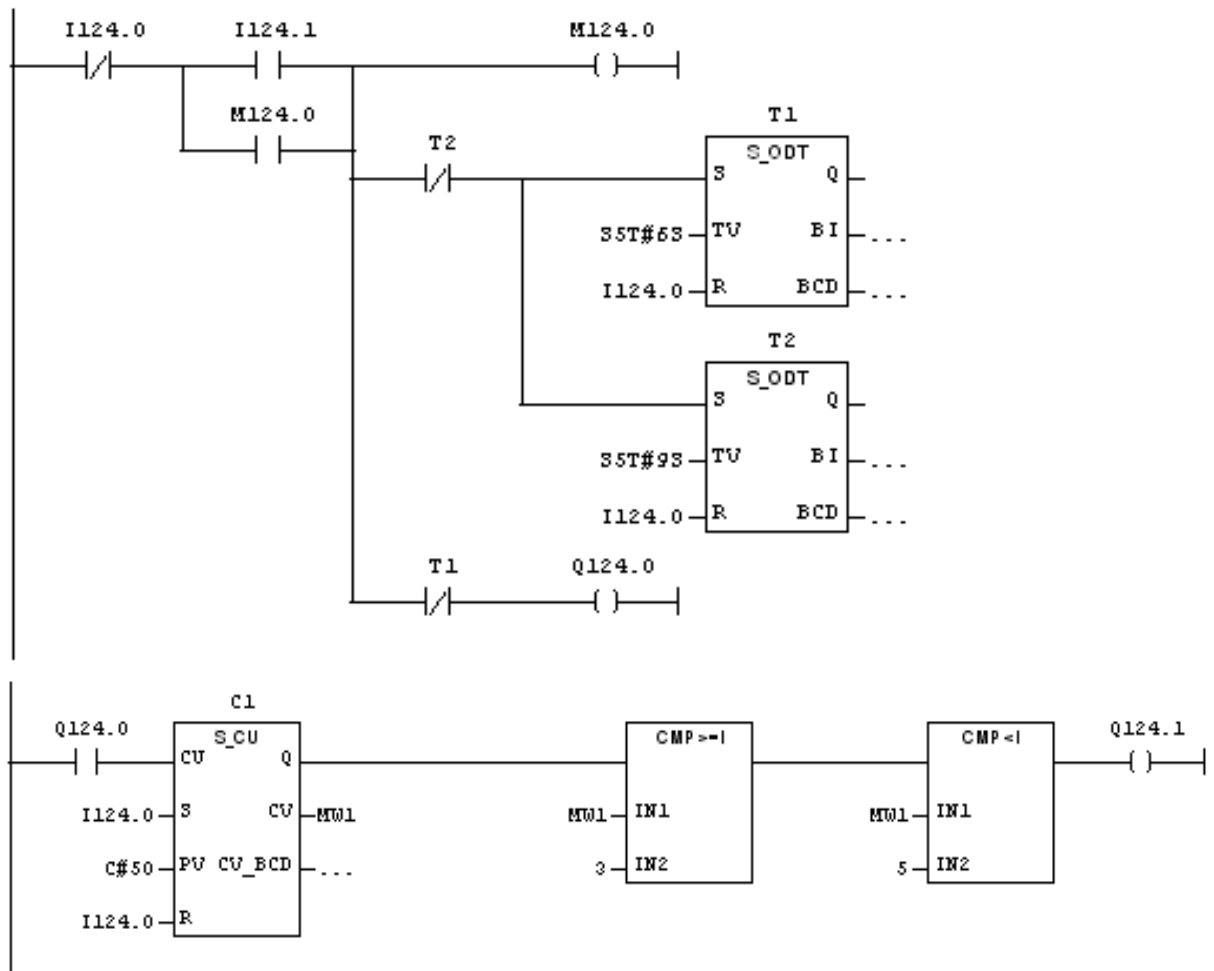
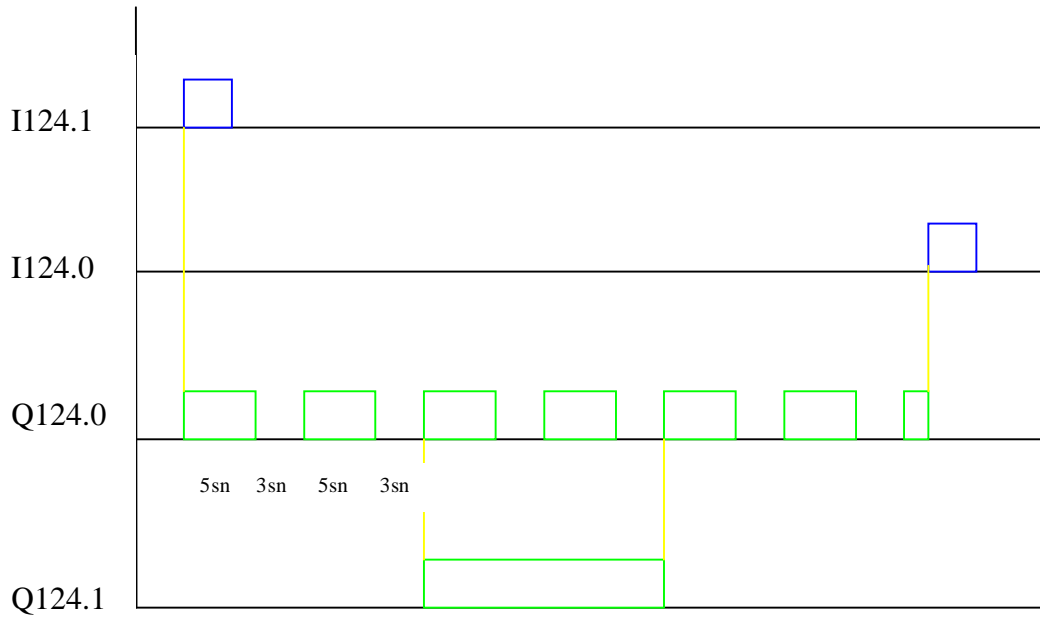


ÖRNEK :

Starta basılınca taşıyıcı bant çalışıyor. Taşıyıcı bant üzerindeki kutular sensör tarafından algılanıyor.12 adet kutu geçince taşıyıcı bant geçici olarak 5sn duruyor. Bu esnada taşıyıcı bandın durmasıyla piston ileri çıkarak kutuları ileri itiyor ve geri geliyor. Band tekrar çalışarak işlemi gerçekleştirilmeye devam ediyor. Çalışma stop butonu ile sonlanıyor. İşlem Kumanda sistemini PLC'de oluşturunuz



ÖRNEK :



DEĞİŞKEN TABLOSU (VARIABLE TABLE – VAT)

Programa bağlı kalmadan, bir PLC’de bulunan değişkenleri statik veya dinamik olarak izlemek ve gerekirse değerini değiştirmek için kullanılan tablolardır.

VAT tabloları CPU’ya download edilmezler ancak bağlanırlar. Sayıları 65535 adet olabilir. VAT tablosunda kullanılan özel bir menü bulunur.



Değişken tetikleme durumlarını ayarlar . (Variable Trigger)



Değişikliğin izlenmesini sağlar . I-O adreslemelerinde kullanılır. (Monitor Variable)



Değişikliği yapar. I-O adreslemelerinde kullanılır. (Modify Variable)

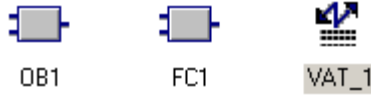


Memory, Timer, Counter, Data gibi elemanlarda kullanılır (Update Monitor Values)

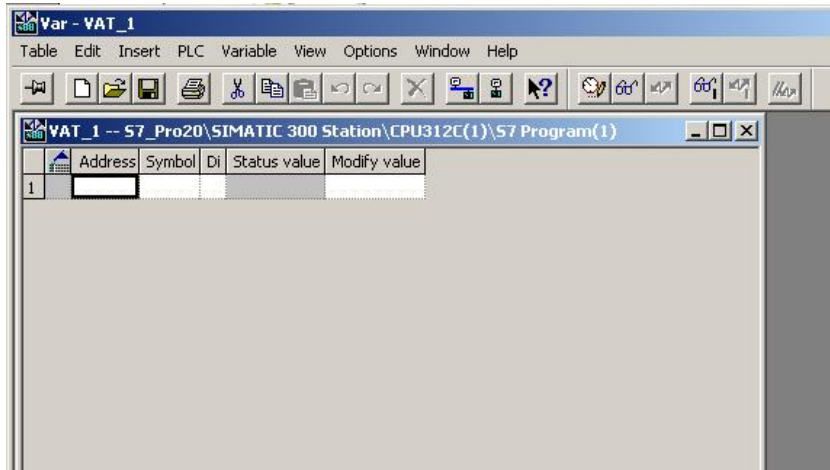


Tekrar çalışma durumunda değişikliği yeniler (Update Modify Values)


- 1) Block’ların bulunduğu yere “VAT_1” sembolü ile VAT tablosu eklenir




- 2) VAT1 seçildiğinde aşağıdaki şekilde ekrana gelecektir.



- 3) İzlenecek ve değeri değiştirilecek eleman adresleri tanımlanmalıdır

- 4)  simgesini seçilerek izleme açılır.
- 5) Hangi adrese müdahale edilecekse "Modify Value" kısmına yeni değer girilir.
- 6) Giriş ve çıkış adresleri modify edilecekse çevrime bağlı olarak bazı ayarlamalar yapmak gerekmektedir.

- Giriş adresleri modify edilecekse :

Variable Trigger () seçildiğinde

"Trigger point of modifying" "Beginning of Scan Cycle"

"Trigger Condition of Cycle"..... "Every Cycle" ayarlanmalıdır.

- Çıkış adresleri modify edilecekse :

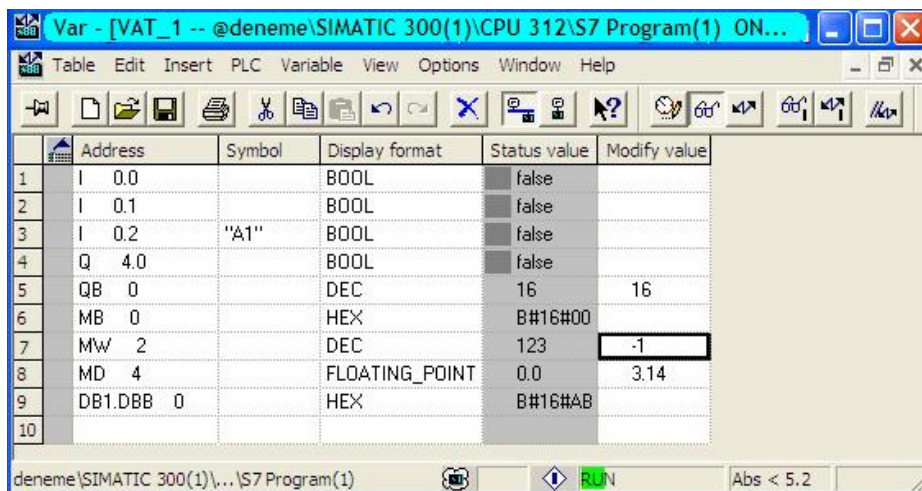
Variable Trigger () seçildiğinde

"Trigger point of modifying" "End of Scan Cycle",

"Trigger Condition of Cycle"..... "Every Cycle"

Counter , Timer gibi giriş ve çıkış değişkenlerinde Trigger ayarı yapılmaz

- 7)  kısa yolu seçilerek değişiklik gerçekleştirilir.

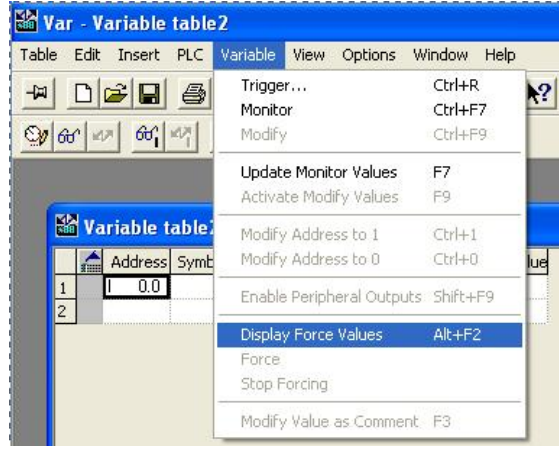


The screenshot shows the SIMATIC Manager Variable Table window. The table has the following columns: Address, Symbol, Display format, Status value, and Modify value. The data is as follows:

| | Address | Symbol | Display format | Status value | Modify value |
|----|-----------|--------|----------------|--------------|--------------|
| 1 | I 0.0 | | BOOL | false | |
| 2 | I 0.1 | | BOOL | false | |
| 3 | I 0.2 | "A1" | BOOL | false | |
| 4 | Q 4.0 | | BOOL | false | |
| 5 | QB 0 | | DEC | 16 | 16 |
| 6 | MB 0 | | HEX | B#16#00 | |
| 7 | MW 2 | | DEC | 123 | -1 |
| 8 | MD 4 | | FLOATING_POINT | 0.0 | 3.14 |
| 9 | DB1.DBB 0 | | HEX | B#16#AB | |
| 10 | | | | | |

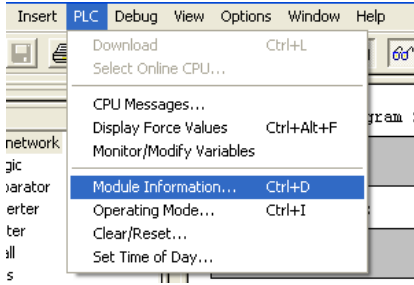
The status bar at the bottom shows the program is running (RUN) and the absolute error is less than 5.2 (Abs < 5.2).

BİR GİRİŞİ VEYA ÇIKIŞI SÜREKLİ ENERJİLİ TUTMAK (FORCE)



Açılan tabloda force edilecek adres veya adresler girilir. Force edilir.

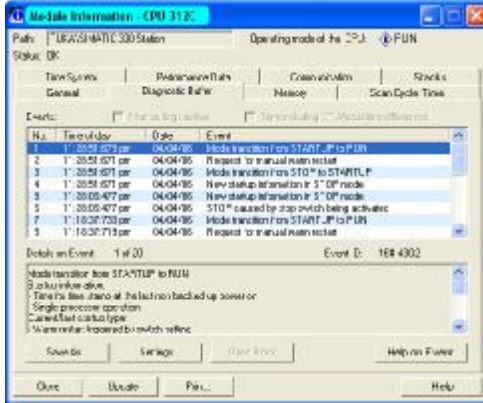
MODÜLE AİT BİLGİLER



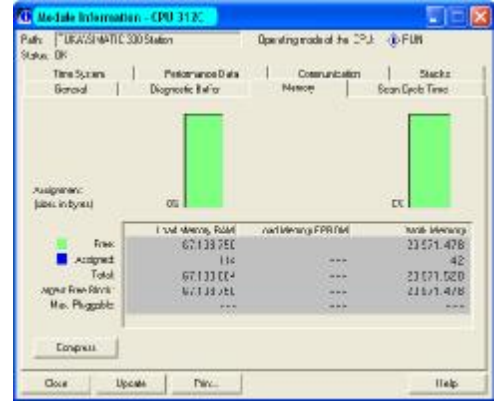
Module information modül bilgileridir. Etkin olarak kullanılan bilgiler aşağıda örneklenmiştir.

1. Diagnostic buffer : İşlem sırası
2. Memory : Kullanılan hafıza alanı
3. Scan Cycle Time : Çevrim süresi
4. Performans Data : CPU'nun özellikleri

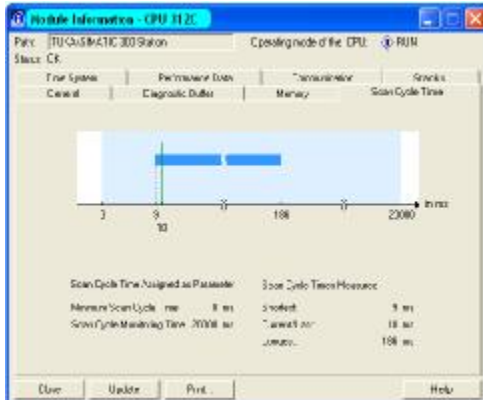
1.



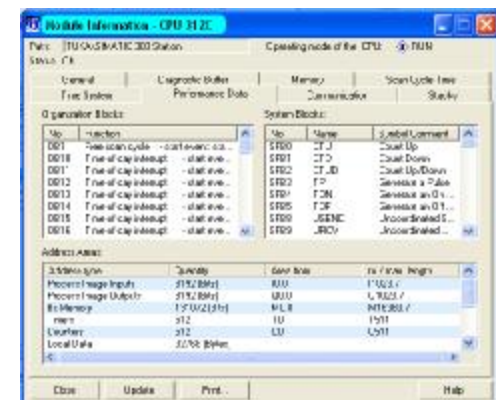
2.



3.



4.



PROGRAMIN KAYDEDİLMESİ VE GEREKTİĞİNDE AÇILMASI

Programlama pencereleri kapatılmalı , Simatic Manager'ın ana safasına dönülmelidir. Bu sayfada :

FILE _ ARCHIVE seçilmelidir. Programa yeni bir ad verilebilir. Kayıt yeride belirlenerek program kaydedilmelidir. Archive'in özelliği programı "Zip" formatında sıkıştırarak saklamasıdır.

Program açılmak istendiğinde "Retrive" seçeneği kullanılmalıdır.