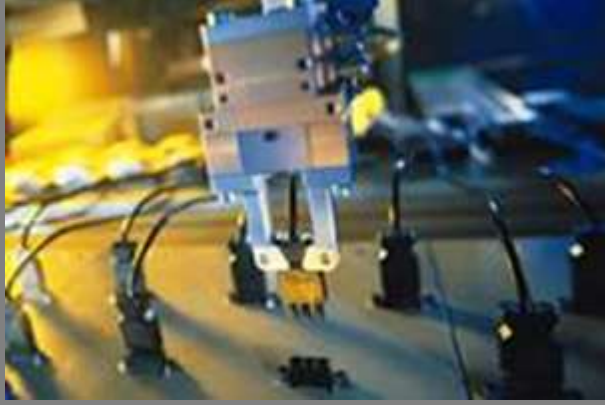


2014

PROGRAMLANABİLİR DENETLEYİCİLER

PROGRAMLANABİLİR DENETLEYİCİ UYGULAMALARI

Bu ders notu programlanabilir denetleyicilerin temel yapısı, programlanması, dijital ve analog giriş/çıkış bağlantılarının yapılması ve programlanmasını sağlayacak temel bilgiler ile bunları destekleyecek örnek programlardan oluşmuştur.



**T.C. EGE ÜNİVERSİTESİ
EGE MESLEK YÜKSEKOKULU**

MEKATRONİK PROGRAMI

"PROGRAMLANABİLİR DENETLEYİCİLER"

"PROGRAMLANABİLİR DENETLEYİCİ UYGULAMALARI"



DERS NOTU

Yrd.Doç.Dr. DilşadEngin

İçerik

| | |
|---|----|
| İçerik | 2 |
| PROGRAMLANABİLİR DENETLEYİCİLER..... | 5 |
| 1 PLC'nin İşlevleri | 5 |
| 2 Donanım..... | 5 |
| 2.1 Giriş / Çıkış Sistemi..... | 6 |
| 2.2 Bellek Yapısı | 13 |
| 2.2.1 Bellek Boyutu..... | 15 |
| 2.2.2 PLC Bellek Düzeni | 16 |
| 3 PLC Programlama Dilleri | 18 |
| 3.1 Merdiven Diyagram Programlaması | 19 |
| 3.1.1 Röle-Tipi Buyruklar | 20 |
| 3.2 Deyim Listesi ile Programlama | 22 |
| 3.2.1 Özel Merdiven Diyagramı Buyrukları: OPEN ve SHORT | 22 |
| 3.2.2 Parantez Kullanma | 22 |
| 3.2.3 İç içe parantez açmak | 23 |
| 3.2.4 MPS, MRD, MPP buyrukları | 24 |
| 3.2.5 Parantez kullanımına ilişkin ek notlar | 25 |
| 3.3 Fonksiyon Bloklarına Giriş | 28 |
| 3.3.1 Zamanlayıcı ve Sayıcı Fonksiyon Blokları | 28 |
| 3.4 Sayısal Komutlar | 30 |
| 3.4.1 Atama Buyruğu (:=) | 30 |
| 3.4.2 Karşılaştırma Buyrukları | 32 |
| 3.4.3 Aritmetik İşlemler | 34 |
| 3.4.4 Mantık Komutları..... | 36 |
| 3.4.5 Kaydırma (Shift) Buyrukları | 37 |
| 3.4.6 Dönüşüm Buyrukları | 38 |
| 3.4.7 Tek/Çift Sözcük Dönüşüm Buyrukları..... | 39 |
| 3.5 Program Buyrukları | 40 |
| 3.5.1 END Buyrukları | 40 |
| 3.5.2 NOP Buyruğu..... | 41 |
| 3.5.3 Dallanma (Jump) Buyrukları..... | 41 |
| 3.5.4 Altyordam (Subroutine) Buyrukları..... | 42 |
| 3.6 Temel Fonksiyon Blokları | 43 |
| 3.6.1 Fonksiyon Blokları Bit ve Sözcük Objeleri | 44 |
| 3.6.2 Zamanlayıcı (Timer) Fonksiyon Bloğu (%Tmi)..... | 45 |

| | | |
|-------|--|----|
| 3.6.3 | Yukarı/Aşağı Sayıcı (Counter) Fonksiyon Bloğu (%Ci) | 51 |
| 3.6.4 | Bit Kaydırma Yazacı (Shift Bit Register) Fonksiyon Bloğu (%SBRi)..... | 55 |
| 3.6.5 | Adım Sayacı (Step Counter) Fonksiyon Bloğu (%SCi)..... | 57 |
| 3.7 | İleri Fonksiyon Blokları | 58 |
| 3.7.1 | İleri Fonksiyon Bloğu İleri Fonksiyon Blokları için Programlama İlkeleri..... | 60 |
| 3.7.2 | LIFO/FIFO Yazaç Fonksiyon Bloğu (%Ri)..... | 61 |
| 3.7.3 | Vuru Genişlik Modülasyon Çıkışı (Pulse Width Modulation Output) Fonksiyon Bloğu, %PWM 65 | |
| 3.7.4 | Vuru Üretme Çıkış (Pulse Generation Output) Fonksiyon Bloğu, %PLS | 67 |
| 3.7.5 | Tambur Denetleyici (Drum Controller) Fonksiyon Bloğu (%DR) | 69 |
| 3.7.6 | Hızlı Sayıcı Fonksiyon Bloğu (Fast Counter Function Block), %FC..... | 73 |
| 3.8 | Saat Fonksiyonları | 75 |
| 3.8.1 | GZS Düzeltmesi Değeri | 75 |
| 3.8.2 | Takvim Blokları | 75 |
| 3.8.3 | Program aracılığıyla Zaman Tarihleme | 77 |
| 3.8.4 | Tarih ve Zamanı Ayarlamak | 78 |
| 3.9 | Dahili Analog Fonksiyonlar | 80 |
| 3.9.1 | Potansiyometreler..... | 80 |
| 3.9.2 | Analog Kanal | 80 |
| 3.10 | Sistem Objeleri ve Değişkenleri..... | 82 |
| 3.11 | Sistem Bitleri (%Si)..... | 84 |
| 3.12 | Sistem Sözcükleri (%SWi)..... | 87 |
| 3.13 | Uygulama Örnekleri | 89 |
| 4 | KAYNAKÇA..... | 92 |

PROGRAMLANABİLİR DENETLEYİCİLER

Programlanabilir denetleyiciler röleler ve elektromekanik adımlama aygıtlarının yerine geçen işlevleri gerçeklemek için programlanabilen mikroişlemci tabanlı özel amaçlı bilgisayarlardır. En azından, 1970'lerde ilk kez ortaya çıktıklarında tasarlanma amacı buydu. Programlanabilir Mantık Denetleyicisi (**P**rogrammable **L**ogic **C**ontroller) için PLC kısa adını kullandılar. O zamanlar tasarıma kapalı döngü uygulamaları için PID (oransal, tümevsel, türevsel) denetim yetkinliği eklenmiş ve üretici firmaların pazarlama bölümleri örneksel denetim etkinliğinin artırılmasını mantık yetkinliğinin artırılmasına tercih etmişlerdir.

1 PLC'nin İşlevleri

Programlanabilir mantık denetleyicilerinin temel görevi röleler, sayıcılar, zamanlayıcılar, tambur ardışılmalıyıcılar ve adım motorları gibi elektromekanik aygıtlar tarafından gerçekleştirilen işlevleri elektronik olarak yerine getirmektir. Piyasaya ilk sürüldüğü haliyle, ac ya da dc işaretlerin geçmesi için kapanan kontaklara sahip dış aygıtlardan kesikli (aç-kapa) işaretleri almaktaydı ve röle bobinlerini, solenoidleri, ışıkları, uyarma aygıtlarını, ve diğer elektriksel yükleri enerjilendirmek için elektriksel gerilimleri üreten çıkışlar yaratmaktaydı. PLC'ler halen bu işlemi çok iyi yerine getirmektedirler.

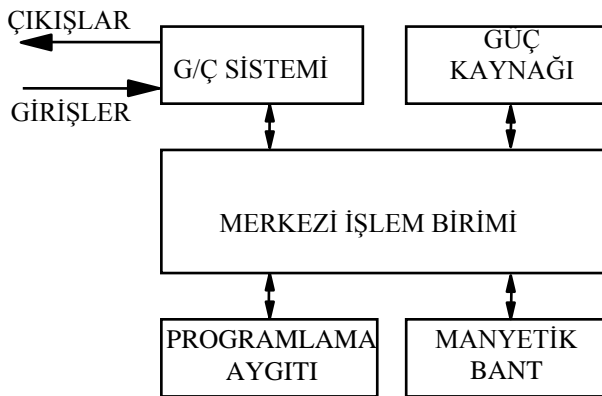
Bugün piyasada bulunan genişletilmiş şekliyle PLC, sensörler ve transdüserlerden sürekli değişen örneksel işaretleri kabul edebilir ve sürekli değişen örneksel işaretleri çıkış olarak üretebilir. Bu işaretler akım-basınç (I/P) dönüştürücüleri gibi regülatörler ve SCR'leri çalıştırabilir. Bazı PLC'ler ısılıçiftler ve gerilme göstergeleri gibi süreç ölçüm aygıtlarından doğrudan alınan düşük seviyeli sayısal girişleri, darbeleri ve milivolt düzeyindeki gerilimleri kabul edebilir.

İlk üretilen tasarımların makine denetimi uygulamalarında kullanılması amaçlanmıştı. Bugünkü tasarımlar endüstriyel süreç denetim uygulamalarında da kullanılabilir.

Yeni gelişmeler piyasaya sürüldükçe, PLC'nin görünüşü de değişmektedir. İlk programlanabilir mantık denetleyicileri nispeten küçük modüller halinde, panoya takılabilir boyutlardaydı. Bunlar yalnızca klavye ve ekrana sahipti. Katot ışınlu tüpler (CRT) 12 inçten daha büyük olmayan siyah-beyaz tüplerdi. Yıllar geçtikçe renkli süreç grafik ekranına sahip PLC'ler piyasaya sürüldü. Bugün süreç denetim uygulamaları için belirgin bir gelişme olan konsollar ve geniş ekranlar üretilmektedir. PLC tasarımı halen mantık denetimi ve ardışık işlemler içindir.

2 Donanım

Gelişmiş bir PLC'de bulunması gereken birimler mikroişlemci veya bilgisayar olarak adlandırılan merkezi işlem birimi (MİB), giriş/çıkış (G/Ç) birimi, bellek, bir programlama dili ve aygıtı, ve bir güç kaynağıdır. Şekil-1'de bir programlanabilir mantık denetleyicisinin bileşenleri görülmektedir.



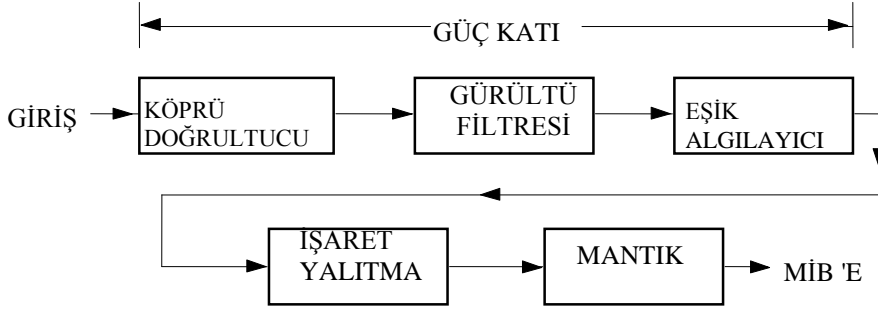
Şekil-1 Bir programlanabilir denetleyicinin bileşenleri.

Şekil-1'deki PLC sisteminde bir giriş/çıkış sistemi, bir güç kaynağı, bir manyetik bant saklama birimi, bir programlama aygıtı ve bir merkezi işlem birimi bulunmaktadır.

2.1 Giriş / Çıkış Sistemi

Giriş/Çıkış sistemi seviye, sıcaklık, basınç, akım, ve gerilim gibi sürecin fiziksel niceliklerini ölçmek ve iletmek için çeşitli iletişim (arabirim) devreleri kullanır. Algılanan durum veya ölçülen değerlere dayanarak, MİB vanalar, motorlar, pompalar ve alarmlar gibi çeşitli aygıtları bir makine ya da süreç üzerinde denetim sağlamak amacıyla denetler. PLC giriş ve çıkış işaretleri Tablo-1'de gösterilmiştir.

AC Giriş Modülleri Şekil-2'de bir AC giriş modülünün blok şeması görülmektedir. Üretilen PLC'ler arasında çok farklı giriş modülleri bulunmakla birlikte genelde tüm ac giriş modülleri bu şemadakine benzer şekilde çalışır. Giriş devresi iki bölümden oluşmaktadır: güç katı ve mantık katı. Devrenin güç ve mantık katları giriş güç katını mantık devrelerinden elektriksel olarak yalıtın bir devre ile bağlatırılmıştır. Bu elektriksel yalıtma normalde gürültülü olan endüstriyel ortamda çok önem taşır.



Şekil-2 AC giriş modülü blok şeması.

Bir giriş modülünün güç katı bir giriş aygıtından gelen gerilimi (220V ac) program denetim taramasında işlemcinin kullanabileceği bir dc mantık seviyesine çevirir. Köprü doğrultucu devresi gelen ac işareti bir dc seviyeye dönüştürür ve bu işaret bir filtreye beslenir. Bu filtre devresi giriş güç hattındaki elektriksel gürültüye karşı koruma sağlar ve 10 ile 25 ms'lik bir zaman gecikmesine de neden olur. Eşik devresi gelen işaretin belirlenen giriş nominal değerine uygun gerilim düzeyine ulaşmış olup olmadığını saptar. Giriş işareti eşik gerilimini aşar ve en azından filtre gecikmesi kadar bir süre bu gerilim değerinin üzerinde kalırsa, işaret geçerli bir giriş olarak kabul edilir.

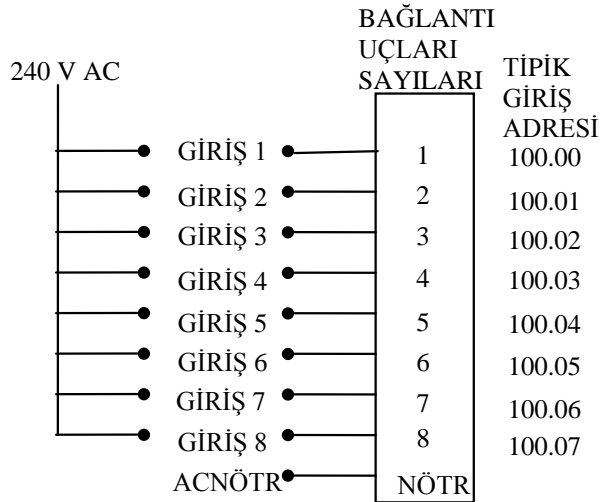
Tablo-1 Programlanabilir Mantık Denetleyici 'nin Giriş ve Çıkış İşaretleri

| Girişler | | |
|--|-----------------------|--|
| Kaynak | Giriş Çeşidi | Açıklama |
| Sınır anahtarları | 220 V ac / 250 V dc | Giriş filtreleme |
| Seçici anahtarlar | | |
| Basma butonu | | |
| Yaklaşım anahtarları | 220 V ac, 5 - 30 V dc | |
| Fotoelektrik anahtarlar | 12, 24, 48 V dc | |
| TTL sinyaller | 5 V dc | |
| Kodlayıcılar | 5, 12 veya 24 V dc | 50kHz'e kadar darbeleri sayılabilir |
| Dörtlü (Quadrature) tip | | |
| Arttırmalı tip | | |
| Optik ışın sayıcılar | | |
| Sıfır hız anahtarlı sekiz-bitlik Gray kodu | 12 - 24 V dc | İşaretler mutlak kodlayıcılardan alınmakta |
| Thumb wheel anahtarlar | 12 - 24 V dc | |
| ASCII | 12 - 24 V dc | Arabirim (iletişim) işaretleri çevre birimlerine gönderilmekte |
| Isılçiftler, RTD 'ler | Milivolt seviyeleri | Sıcaklık ölçme elemanları |

| Çıkışlar | | |
|--------------------------|----------------------------|---|
| Yük | Gerilim düzeyi | Açıklama |
| Röleler | 120 V ac 2.0 A | Triak çıkışı |
| Seloidler | 220 - 240 V ac | |
| TTL devreler | 5 V dc | |
| Mantık devreleri | | Reed Röleler |
| Anahtarlama röleleri | 0 - 250 V ac | Cıvalı röleler |
| Motor starterleri | 0 - 175 V dc | |
| Değişken Hızlı sürücüler | | |
| Servolar | ±10 V dc | Robotlar için eksenel hareket denetimi |
| Giriş / Çıkış | | |
| Örneksel devreler | 0 - 5V, 0 - 10V, 4 - 20 mA | Genellikle modül başına iki tane |
| Çeşitli | | |
| Saatli darbeler | | İşlemci tarafından takvim ve zaman karşılaştırmalarında kullanılır. |

Geçerli bir işaret saptandığında, ac seviyeden mantık seviyesine geçişte elektriksel yalıtımı sağlayan yalıtma devresinden geçirilir. Yalıtma devresinden alınan dc işaret mantık devresi tarafından kullanılır ve işlemci veri yolu için kullanılabilir hale getirilir. Saha aygıtı (güç) ve denetleyici (mantık) arasında elektriksel bağlantı olmayacak şekilde elektriksel yalıtma yapılır. Bu elektriksel ayırma arabirimin (veya denetleyicinin) mantık tarafının büyük sivri gerilim darbelerinden korunmasını sağlar. Güç ve mantık kısımlarının arasındaki bağlaşım optik bağlaştırmacı veya darbe transformatörü ile gerçekleşir.

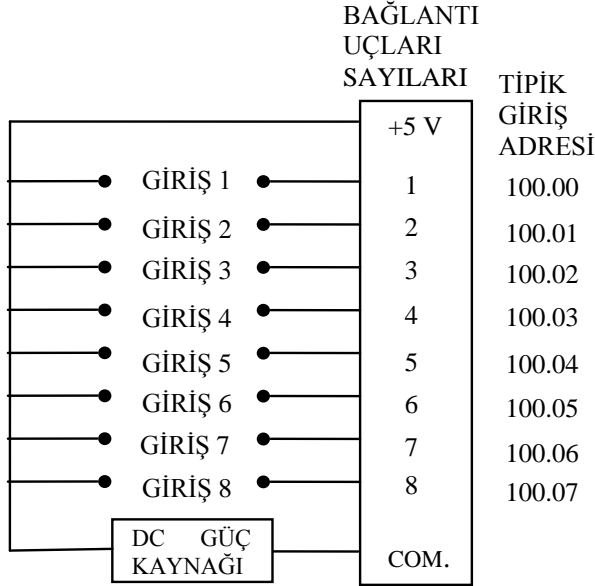
Giriş modüllerinin çoğu uygun giriş gerilim seviyesinin varlığını belirtmek için bir LED gösterge kullanılabilir. Şekil-3 'te bir ac giriş bağlantı şeması görülmektedir.



Şekil-3 Tipik giriş modülü (240 V AC).

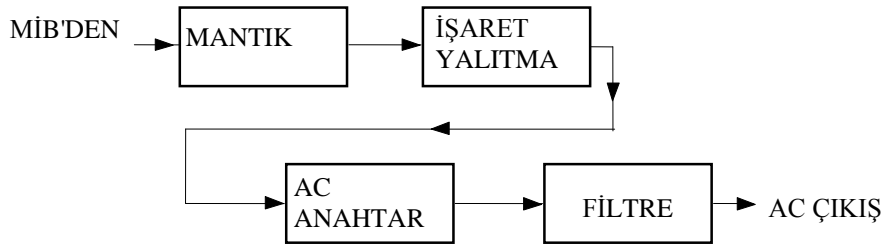
Transistör Transistör Mantığı (TTL) Giriş Modülleri

TTL giriş modülleri denetleyicinin katıhal denetimi ve algılama enstrümanlarını içeren TTL-uyumlu aygıtlardan alınan işaretleri kabul etmesini sağlar. TTL girişler aynı zamanda bazı 5V dc seviyeli denetim aygıtları ve bazı tür fotoelektrik algılayıcılar ile iletişim kurmak için de kullanılırlar. TTL arabirimi ac giriş modüllerine benzer bir yapıya sahiptir; ancak, filtrelemeden kaynaklanan giriş gecikme süresi daha kısadır. TTL giriş modülleri normalde belirli akım sınırlamalarına sahip +5V'luk dc güç kaynağı ile dışarıdan beslenirler. Şekil-4'te bir TTL giriş bağlantı şeması görülmektedir.



Şekil-4 TTL bağlantı şeması.

AC Çıkış Modülleri Şekil-5'te bir ac çıkış modülünün iç devresinin blok şeması görülmektedir. Üretilen PLC'ler arasında çok farklı AC çıkış modüllerine sahip olanlar vardır, fakat blok gösterim ac çıkışların temel çalışma ilkesini göstermektedir. Devre temelde bir yalıtma devresi ile bağlaştırılan mantık ve güç katlarından oluşmaktadır. Çıkış arabirimi çıkış aygıtını denetleyen gücün sağlandığı basit bir anahtar olarak düşünülebilir.



Şekil-5 AC çıkış modülü.

Normal çalışma sırasında, işlemci mantık programı tarafından belirlenen çıkış durumunu mantık devresine gönderir. Çıkış enerjilendirildiyse, işlemciden alınan işaret mantık kısmına beslenir ve gücü saha aygıtına anahtarlayacak yalıtma devresinden geçirilir.

Anahtarlama kısmı genellikle gücü anahtarlama için bir silisyum kontrollü doğrultucu (SCR) veya triak kullanılır. AC anahtar normalde tepe gerilimini maksimum anma değerinin altındaki herhangi bir değere sınırlamak ve aynı zamanda elektriksel gürültünün modülün çalışmasını etkilemesini engellemek için kullanılan bir RC (snubber) koruma ve çoğunlukla bir metal oksit varistör (MOV) tarafından korunur. Çıkış devresine konulacak bir sigorta ac anahtarın aşırı akımdan korunması için kullanılabilir. Eğer devrede sigorta bulunmuyorsa, kullanıcı tarafından konulması gerekir.

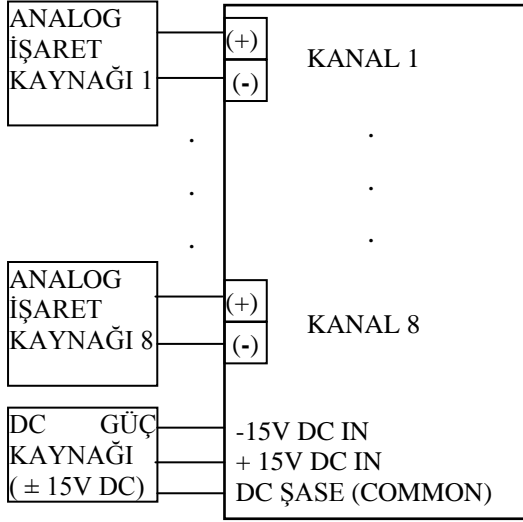
Giriş modüllerindeki gibi, çıkış modülünde de çalışma mantığını göstermek için LED göstergeler bulunabilir. Devrede sigorta bulunuyorsa, sigorta durumunu gösteren bir gösterge de bağlanabilir. AC

TTL Çıkış Modülü TTL çıkış modülü denetleyicinin yedi parçalı LED göstergeler, tümleşik devreler, ve çeşitli 5V dc mantık aygıtları gibi TTL uyumlu çıkış aygıtlarını sürmesini sağlar. Bu modüller genellikle özel akım değerlerinde dışarıdan +5V'luk dc güç kaynağına bağlanırlar. Bir TTL çıkış modülü için bağlantı şeması Şekil-7 'deki DC çıkış modülü bağlantı şemasıyla aynı olup farklı olarak kaynak gerilimi +5Vd.c.'dir.

Örneksel Giriş Modülleri Örneksel giriş arabirimi saha aygıtlarından gelen örneksel akım veya gerilim işaretlerini kabul etmek için gerekli devreyi içerir. Gerilim veya akım girişleri bir örneksel-sayısal çevirici (ADC) tarafından sayısal değere çevrilir. Örneksel işaretle orantılı dönüşüm değeri denetleyicinin veri yoluna aktarılır ve daha sonra kullanılmak üzere bir bellek yerinde saklanır.

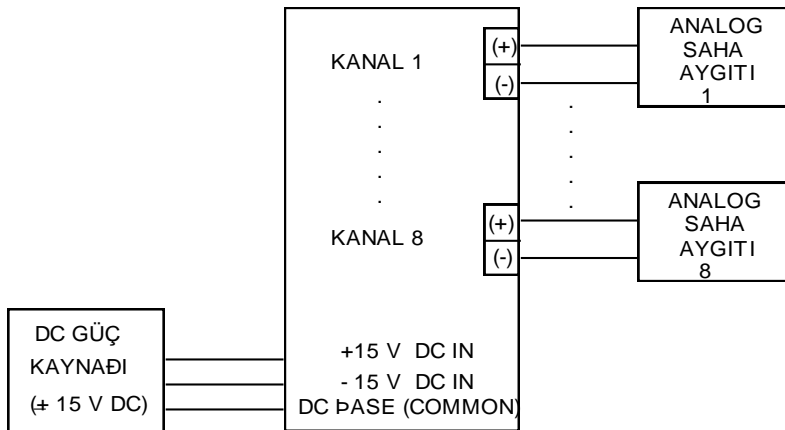
Örneksel giriş arabirimlerinin giriş empedansı, düşük güçteki (4-20 mA çıkışlı) saha aygıtlarıyla iletişim kurulabilmesi için çok yüksektir. Örneksel aygıttan gelen giriş hattında genellikle bükülü-çift iletkenler kullanılır. Bükülü-çift kablo düşük bir gürültü giriş işareti sağlar. Arabirimin giriş katı, modülü ek saha gürültüsünden korumak için süzme ve yalıtma devrelerine sahiptir.

Dönüşümden sonraki örneksel değer giriş işaretlerinin düşük ve yüksek (tam skala) değerlerini ifade eden 000'dan 999'a kadar İKO (BCD) olarak kodlanır. Şekil-8'de bir örneksel giriş modülünün bağlantısı görülmektedir.



Şekil-8 Örneksel giriş modülü bağlantı şeması.

Örneksel Çıkış Modülleri Örneksel çıkış modülü, bir örneksel saha aygıtını denetlemek için, MİB'den oransal gerilim veya akıma çevrilen sayısal veri alır. Sayısal veri bir DAC'a gönderilir ve çıkış artık örnekseldir. Çıkış devresi ve mantık devresi arasındaki yalıtma genellikle optik bağlaştırmacılarla sağlanır. Bu çıkış modülleri belirli akım ve gerilim değerlerine sahip bir dış güç kaynağı ile beslenir. Şekil-9'da bir örneksel çıkış modülünün aygıt bağlantısı görülmektedir.



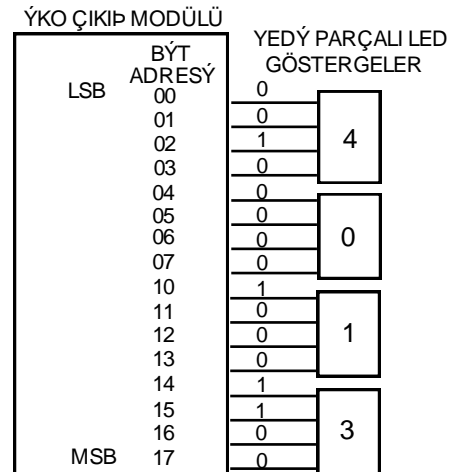
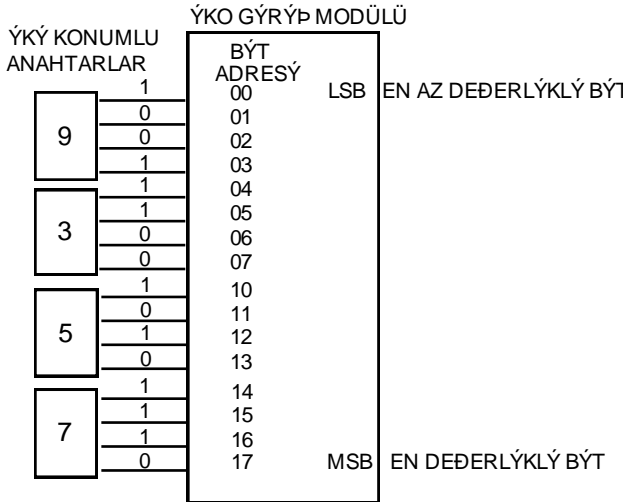
Şekil-9 Örneksel çıkış bağlantı şeması.

İkilik Kodlanmış Onluk (İKO≡BCD) Giriş Modülleri İkilik kodlanmış sayısal giriş modülü işlemci ve "thumbwheel" (iki konumlu) anahtar gibi giriş aygıtları arasında koşut iletişim sağlar. Bu tür modül genellikle parametreleri denetim programı tarafından kullanılan bellekteki özel veri yerlerine girmek için kullanılır. Bu parametreler zamanlayıcı ve sayıcı önkurma değerleri ve süreç denetim set noktası değerleridir.

Bu modüller genellikle 5V dc (TTL) ile 24V dc arasındaki gerilimleri kabul eder ve bir veya iki G/Ç (I/O) yazacına karşılık gelen 16 veya 32 giriş içeren bir modül içinde gruplanırlar. Veri işleme buyruklarına yazac giriş arabiriminden alınan veriye erişmek için kullanılırlar. Şekil-10'da bir İKO giriş modülü için aygıt bağlantısı görülmektedir. Bu modülde, iki konumlu anahtardaki her bit ("1"/"0") PLC 'deki bir kelime yerindeki bir bit yerini denetler.

İKO Çıkış Modülleri Bu sayısal modül işlemci ve yedi parçalı LED gösterge veya bir İKO alfasayısal gösterge gibi bir çıkış aygıtı arasında koşut iletişim sağlar. İKO çıkış modülü aynı zamanda düşük akımla çalışan TTL mantık yüklerine de gerilim sağlar. Yazac çıkış modülü genellikle 5V dc (TTL) ile 30V dc arasında gerilimler sağlar ve 16 veya 32 çıkış hattına (bir veya iki G/Ç yazacına) sahiptir.

Bilgi, işlemciden veri iletim veya G/Ç yazac komutu aracılığıyla gönderildiğinde, veri modülde tutulur ve çıkış devresinde kullanılır hale getirilir. Şekil-11'de bir yazac çıkış modül bağlantısı görülmektedir.

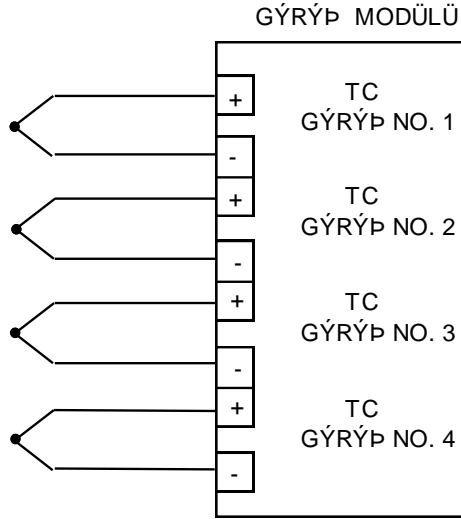


Şekil-10 Bir İKO giriş modülü bağlantı şeması.

Şekil-11 Bir İKO çıkış modülü bağlantı şeması.

Özel PLC Giriş / Çıkış Modülleri Belirli tür işaretleri veya veriyi verimli olarak işlemek için PLC'nin özel modüllere gereksinimi vardır. Bu özel arabirimler ısılıçift modülleri gibi giriş işaretlerini veya standart G/Ç modüllerini kullanarak iletişim kurulamayan diğer işaretleri düzenleyen modülleri içermektedir. Özel G/Ç modüllerinde arabirime zeka kazandıran mikroişlemci de kullanılabilir. Bu akıllı modüller MİB'den ve denetim programı taramasından bağımsız tüm süreç işlevlerini yerine getirebilirler.

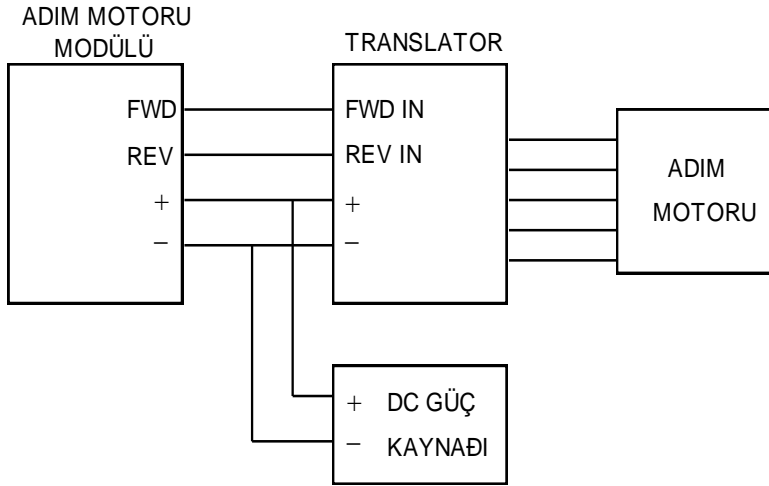
Isılıçift Giriş Modülü Bir ısılıçift giriş modülü girişleri doğrudan bir ısılıçiftten almak için tasarlanmıştır; aynı zamanda soğuk eklemdeki (referans eklemi) sıcaklık değişimlerinin etkisini de gidermektedir. Bu çeşit bir modülün çalışması çok düşük seviyedeki işaretlerin ısılıçiftten (J-tipi TC için en fazla sıcaklıkta yaklaşık 43mV) alınmasının dışında standart örneksel girişinkine benzerdir. Bu işaretler süzülür, yükseltilir ve A/D çevirici ile sayısallaştırılır ve sonra bir program buyruğuaracılığıyla MİB'ne gönderilir. Veri, sıcaklık denetimi ve/veya gösterimi için denetim programı tarafından kullanılır. Şekil-12 'de ısılıçift giriş bağlantı şeması görülmektedir.



Şekil-12 Isılçift (TC) modülü bağlantı şeması.

ASCII Giriş / Çıkış Modülü ASCII giriş/çıkış modülü çevre donanımı ve denetleyici arasında alfasayısal veri göndermek ve almak için kullanılmaktadır. ASCII giriş/çıkışa sahip çevre aygıtları arasında yazıcılar, video ekranları, sayısal göstere enstrümanları ve diğerleri bulunmaktadır. Bu özel G/Ç modülü, üretici firmaya bağlı olarak, kart üzerinde bellek ve beraberinde mikroişlemci içeren iletişim arabirim devresine sahiptir. Bilgi alışverişi genellikle bir RS-232, RS-422, veya bir 20 mA akım döngü iletişim bağlantısı ile yapılmaktadır.

ASCII modülü genellikle iletilecek veri bloklarını saklayabilen kendi RAM belleğine sahiptir. Çevre birimlerden gelen giriş verisi modül tarafından algılandığında, bir veri aktarım buyruğu ile PLC G/Ç veri hattı hızında PLC belleğine aktarılır. Tüm başlangıç iletişim parametreleri, eşlik (çift veya tek), durdurma bitlerinin sayısı, ve iletişim hızı vb., donanımla veya yazılımla seçilebilir.



Şekil-13 Adım motoru bağlantı şeması.

Adım Motoru Çıkış Modülü Adım motoru modülü, adım motoru uyumlandırıcısı için bir darbe dizisi üretir. Uyumlandırıcıya gönderilen darbeler motora adım sayısı, hız ve yön bilgilerini verir.

Adım motoru arabirimi denetim programından konum komutlarını alır. Bu komutlar genellikle program denetimi sırasında tanımlanır ve çıkış arabirimi ile başlangıç koşullarına getirildiğinde PLC programına göre çıkışta darbeler üretir. Hareket başladıktan sonra, çıkış modülü genellikle hareket tamamlanana kadar MİB 'den gelen komutları kabul etmeyecektir.

Darbelerin adım hızı seçilebilir adım darbe genişliğinde 1 ile 50 kHz arasında değişebilir. Bu modüller genellikle dışarıdan bir güç kaynağı ile beslenirler. Şekil-15'te bir adım motoru bağlantı şeması görülmektedir.

Denetim Döngüsü Modülü Denetim döngüsü modülü oransal-tümlevsel-türevsel (PID) denetim algoritmasının gerektiği kapalı-döngü denetimde kullanılır. Bazı üreticiler bu arabirime PID modül de demektedir, ve sürekli kapalı-döngü denetiminin gerektiği herhangi bir sürece uygulanır.

Bu modül tarafından yerine getirilen denetim algoritması aşağıdaki eşitlikle belirtilmektedir:

$$V_{\text{çıkış}} = K_p e + K_i \int e dt + K_d \frac{de}{dt}$$

$V_{\text{çıkış}}$ = çıkış denetim değişkeni

e = SD - SN = hata

K_p = oransal kazanç

K_i = tümlevsel kazanç

K_d = türevsel kazanç

Modül süreç değişkenini (SD) alır, kullanıcı tarafından seçilen set noktası (SN) ile karşılaştırır ve hata farkını hesaplar. Kullanıcı (operatör) veya kontrol mühendisi uygulamaya göre denetim parametrelerinin (K_p , K_i , ve K_d) değerlerini belirler.

PID modülüne PLC işlemcisi tarafından gönderilen bilgi öncelikle denetim parametreleri ve set noktalarıdır. Kullanılan PID modülüne göre, veri çıkış değişkeninin ($V_{\text{çıkış}}$) güncellendiği süre olan güncelleme zamanını ve hata işareti ile karşılaştırılan bir nicelik olan hata ölü bandını tanımlamak için gönderilebilir. Eğer hata işaret hatasından daha az ya da eşitse, güncelleme olmaz. Bazı modüllerde akış denetim döngülerinde kullanılmak üzere doğrusallaştırılmış skalalı bir çıkış elde etmek için süreç değişkeninin kare kökünü alan kare kök ayırıcı (square root extractor) bulunmaktadır.

2.2 Bellek Yapısı

Bellek PLC sistemin denetleme programını saklamak için kullanılır ve genellikle MİB ile aynı yere yerleştirilir. Bellekte saklanan bilgi giriş ve çıkış verisinin nasıl işleneceğini belirler.

Bellek elemanları bit adı verilen veri parçacıklarını saklarlar. Bir bitin iki durumu vardır - "1" veya "0", aç veya kapa, DOĞRU veya YANLIŞ, vb. Bellek birimleri devre kartları üzerine yerleştirilmiştir ve "K"nın katlarıyla yani binlerle belirtilirler. 1K pek çok durumda 1024 kelimelik ($2^{10} = 1024$) saklama yeridir. PLC bellek kapasitesi bin bitten daha azdan 64,000 kelimeye (64K kelime) kadar değişmektedir. Denetleme programının karmaşıklığı gereken bellek kapasitesini belirler.

Bellekte saklanan programlar ve veri dört terim kullanılarak tanımlanır: (1) Yürütme (İşletim Sistemi), (2) Genel Kayıt Belleği (Scratch Pad), (3) Denetleme Programı, ve (4) Veri Tablosu. *Yürütme* PLC 'nin işletim sistemini oluşturan programların geçici olarak saklandığı yerdir. Bu işletim sistemi denetleme programının işletimi, çevre birimleri ile iletişim, ve diğer sistem içdüzen işlevleri (sayma, kaydırma, dizinleme, vb.) gibi etkinliklerini yönetir. *Genel Kayıt Belleği* MİB'nin geçici denetleme veya hesaplamalar için gerekli az miktardaki veriyi saklamak için kullandığı geçici bir saklama alanıdır. Bazı işlemler için yüksek hızlı bir kayıt ortamı sağlayan bu alan veri işleme hızını arttırmaktadır. *Denetleme Programı* alanı kullanıcı tarafından girilen herhangi programlanmış buyrukların kaydedilmesi için kullanılır. *Veri Tablosu*, zamanlayıcı / sayıcı önayar (preset) değerleri ve ana program veya MİB'nin kullandığı yüklü sabitler veya değişkenler gibi denetleme programıyla ilişkili herhangi bir veriyi saklar. Aynı zamanda giriş/çıkış durum bilgisini de tutar.

Saklama ve erişim isteği Yürütme, Genel Kayıt Belleği, Denetleme Programı ve Veri Tablosu için aynı değildir; dolayısıyla, bunlar her zaman aynı bellek çeşitlerinde saklanmazlar. Örneğin, Yürütme içeriklerini kalıcı olarak saklayan bir bir belleğe gereksinir ve bu bellek içeriği kasten veya kazayla elektrik kesilmesiyle veya kullanıcı tarafından değiştirilemez. Diğer taraftan kullanıcı verilen herhangi bir denetleme programı ve/veya veri tablosunu değiştirmek isteyecektir.

Birkaç farklı bilgisayar bellek çeşidi olmasına rağmen, her zaman *geçici* veya *kalıcı* olarak

sınıflandırılabilirler. Geçici bellek kolayca değiştirilebilir ve pil yedekleme ve/veya programın kaydedilmiş bir kopyası tarafından desteklendiğinde çok sayıda programlama uygulaması için oldukça uygundur.

Kalıcı bellekte güç tamamen kesildiğinde dahi veri ve programlar saklı kalır. Bir yedekleme sistemine gereksinim duymaz. Kalıcı bellek genellikle değiştirilemez, ancak bazı kalıcı bellek çeşitleri değiştirilebilir.

Bellek Çeşitleri Kullanılacak bellek çeşidini seçerken bir sistem tasarımcısı programlamanın geçici ve kolay olmasıyla ilgilenir. Geçici olmalıdır çünkü bellek süreç denetleme programını tutar ve eğer bu program kaybolursa fabrikadaki üretim durur. Bellek kullanıcı ve PLC arasındaki herhangi bir etkileşimde yer aldığından bellek içeriğini değiştirmenin kolay olması önemlidir. Bu etkileşim başlangıçta sistem programlanması ve yanlışlardan arındırma (debugging) ile başlar ve zamanlayıcı ve sayıcı önkurma değerlerinin değiştirilmesi gibi çevrim içi (on-line) değişikliklerle devam eder.

Salt-Oku Bellek (ROM)

Salt-oku bellek normalde değiştirilemeyen veya değiştirilmeyecek olan sabit bir programı kalıcı olarak saklamak için tasarlanmıştır. Adını veri veya program saklandıktan sonra içeriğinin yalnızca okunabilir fakat yazılamaz/değiştirilemez olmasından alır. Tasarımlarından dolayı ROM'lar genellikle elektriksel gürültü veya güç kaybından kaynaklanan değişikliklere bağışlıdır. Yürütme veya PLC'nin işletim sistemi programı ROM'da saklanır.

PLC'ler nadiren denetim uygulamaları program belleği için ROM'u kullanırlar. Ancak, sabit verinin gerektiği uygulamalarda, ROM hız, maliyet ve güvenilirlik açısından üstünlüklere sahiptir. Genellikle ROM-tabanlı PLC programları fabrikada donanım üreticisi tarafından üretilir. Orijinal buyruk seti programlandıktan sonra kullanıcı tarafından değiştirilemez. Üretici programı yazarken ve hatalardan arındırırken bir Oku/Yaz-tabanlı denetleyici veya bilgisayar kullanır ve sonra son program ROM 'a girilir. ROM uygulama belleği yalnızca çok küçük PLC'lerde bulunmaktadır.

Rasgele Erişimli Bellek (RAM)

Rasgele erişimli bellek veri veya bilginin tek bir yere yazılabilmesi veya buradan okunabilmesi için tasarlanmıştır. İki çeşit RAM vardır: geçici RAM güç kaybı olduğunda içeriğini tutmaz ve kalıcı RAM (core) güç kaybı olduğunda içeriğini tutar. Geçici RAM güç kesildiğinde içeriğini korumak için pil yedekleme sistemine sahiptir. Çoğu programlanabilir denetleyiciler uygulama belleği için pil yedeklemeli RAM kullanmaktadır. RAM kolayca bir denetleme programı yaratılması ve değiştirilmesini sağladığı gibi veri girişine de izin vermektedir. Diğer bellek çeşitleriyle karşılaştırıldığında, RAM göreceli olarak hızlıdır. Pil yedeklemeli RAM 'in tek önemli sakıncası kritik bir zamanda pilin bitebilmesidir, fakat pek çok PLC uygulamalar için piller yeterlidir.

Bu tür bellek düzenlemesi hem geçici hem de kalıcı belleğin üstünlüklerine sahiptir.

Silinebilir Programlanabilir Salt-Oku Bellek (EPROM)

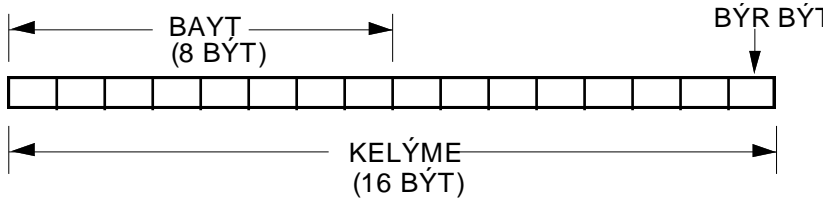
EPROM ultraviyole (UV) (morötesi) ışık kaynağı ile tamamen silindikten sonra yeniden programlanabilen özel bir PROM türüdür. EPROM tümdevre yongası iç bellek devrelerinin UV ışığa tutulabilmesi için tepesinde bir pencereye sahiptir. EPROM değiştirilene kadar içinde bir programı saklayan geçici bir saklama aygıtı olarak düşünülebilir. Bir denetleme programı çevrim içi değişiklikler ve/veya veri girişleri gerektiğinde uygun olmayacaktır. Ancak pek çok PLC'de EPROM denetleme programı belleği pil yedeklemeli RAM'e seçenek olmaktadır. Kalıcı saklama yeteneğiyle EPROM, kolayca değiştirilebilen RAM ile birleştirildiğinde uygun bir bellek sistemi oluşturmaktadır.

Bellek Birimleri (Bitler, Baytlar, Kelimeler) PLC bellekleri her biri "1" veya "0" şeklindeki tek bir bitlik bilgiyi saklayabilen saklama hücrelerinin iki boyutlu dizini olarak düşünülebilir. Bir tek bit adını "Binary digiT" ten almaktadır. Bir bit belleğin en küçük yapı taşıdır ve bilgiyi "1" ler ve "0" lar şeklinde saklar. Gerçekte her hücrede birler ve sıfırlar yoktur; her bir hücrede "gerilim var" bir biri temsil eder veya "gerilim yok" bir sıfırı temsil eder.

Pek çok durumda MİB'in tek bir bitten fazlasını işlemesi gerekir. Örneğin, belleğe ve bellekten veri

iletirken, sayıları saklarken ve kodları programlarken, bayt veya kelime denilen bir grup bite gereksinim duyulur. Bir bayt MİB tarafından bir defada işlenebilen en küçük bit grubudur. PLC 'lerde, bayt boyutu normalde 8 bittir, fakat kullanılan işlemciye bağlı olarak değişebilir.

Kelime uzunluğu genellikle 1 bayt veya daha fazladır. Örneğin, 16 bitlik bir kelimedede 2 bayt bulunur. Programlanabilir denetleyicilerde kelime uzunluğu 8, 16 ve 32 bittir. Bir PLC'nin yapısal bellek birimleri Şekil-14 'te görülmektedir.



Şekil-14 PLC'deki temel yapısal bellek birimleri.

2.2.1 Bellek Boyutu

Programlanabilir denetleyici tabanlı sistemlerin tasarımında bellek boyutu önemli bir etmendir. Doğru bellek boyutunu kesin olarak belirlemek donanım giderlerinden tasarruf sağlayabilir ve zamanın boşa harcanmasını önleyebilir. Bellek boyutunun doğru hesaplanması yeterli kapasitede olmayan veya genişletilemez bir PLC satın alma olasılığını ortadan kaldırır.

Pek çok denetleyicide bellek boyutu maksimum bir noktaya kadar genişletilebilir, fakat bazı küçük PLC'lerde genişletme yapılamaz. (Küçük PLC'ler 10 ile 64 giriş/çıkış aygıtını denetleyen birimler olarak tanımlanmaktadır.) 64 veya daha fazla G/Ç aygıtını idare eden programlanabilir denetleyiciler genellikle 1K, 2K, 4K, vb. katlarında genişletilebilirler. Daha büyük denetleyicilerde bellek boyutu genelde 64K 'dır.

Bir PLC'nin belirtilen bellek boyutu, belleğin bir kısmı denetleyiciler tarafından iç işlevler için kullanıldığından, kullanıcının yararlanabileceği bellek alanının yalnızca kaba bir ifadesidir. Diğer sorun PLC üreticileri tarafından genellikle 8 bit ve 16 bit olmak üzere iki farklı kelime uzunluğunun kullanılmasıdır. Şekil-15'te 2K belleğin 8 ve 16 bitlik sözcükler için karşılaştırılması görülmektedir. 16 bitlik belleğin iki kat saklama kapasitesine sahip olduğu açıktır.

| KELÝME | 8 BÝT | KELÝME | 16 BÝT |
|--------|-------|--------|--------|
| 0000 | | 0000 | |
| 0001 | | 0001 | |
| 0002 | | 0002 | |
| 0003 | | 0003 | |
| 0004 | | 0004 | |
| 0005 | | 0005 | |
| · | · | · | · |
| · | · | · | · |
| · | · | · | · |
| · | · | · | · |
| 2046 | | 2046 | |
| 2047 | | 2047 | |
| 2048 | | 2048 | |

Şekil-15 2K'lık belleğin 8 ve 16 bitlik kelimeler için karşılaştırılması.

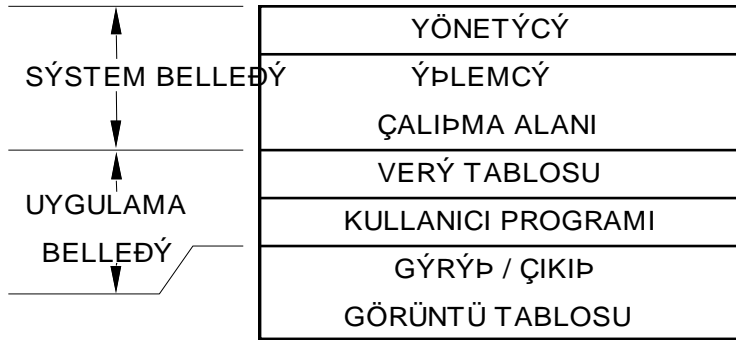
Bir uygulama için bellek boyutunun belirlenmesinde sistemdeki G/Ç noktalarının sayısından yola çıkılmaktadır. G/Ç noktalarının sayısını 10 kelimelik bellekle çarpmak iyi bir kuraldır. Örneğin, eğer sistem 100 G/Ç noktasına sahipse, program genellikle 1000 kelime veya daha az olacaktır. Program boyutu denetleme programının karmaşıklığından da etkilenmektedir. Eğer uygulamada veri yönetimi ve PID denetimi gibi karmaşık denetim algoritmaları kullanılacaksa, ek belleğe gerek duyulacaktır. Belleği doğru olarak boyutlandırmak için, PLC belleğin tüm düzenini anlamamız gerekir.

2.2.2 PLC Bellek Düzeni

PLC belleğinin düzeni *bellek haritası* olarak bilinir. Bir bellek haritası sistem belleği ve uygulama belleğinin yerini göstermek için kullanılır; denetleme programının yazılmasında PLC programlayıcısı için yardımcı olur. Şekil-16'da görülen bellek haritası şu birimleri içerir: yönetici, işlemci çalışma alanı, giriş/çıkış tablosu, veri tablosu ve kullanıcı programı.

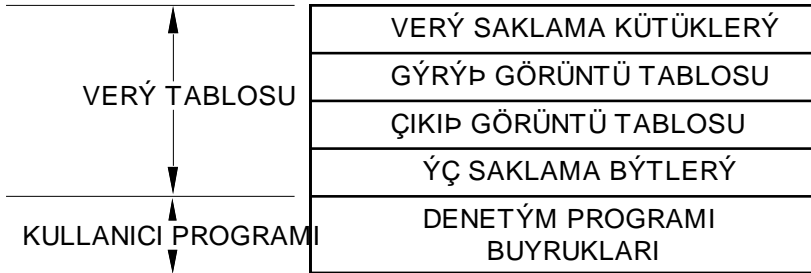
Programlayıcı yönetici veya işlemci çalışma alanına erişime sahip değildir. Bu iki alana sistem belleği denir. Bir uygulama için PLC belleğini boyutlandırırken bu alanı kullanılabilir programlama alanının bir parçası olarak hesaba katmamız gerekir. Diğer taraftan, veri tablosu ve kullanıcı program alanları programcı tarafından kullanılır ve uygulama belleği olarak adlandırılır. Dolayısıyla, PLC belleği üç ana bölümden oluşmaktadır: sistem belleği, uygulama belleği ve giriş/çıkış tablosu.

Bazı denetleyici üreticileri sistem belleğini belirlenmiş bellek boyutunun içine katmaktadır. 64K'lık belleğe sahip bir denetleyici 32K'lık sistem belleği ve 2K'lık G/Ç görüntü tablosuna sahip olabilir ve uygulama belleği olarak geriye yalnızca 30K'lık bellek kalabilir. Pek çok PLC satıcısı bellek boyutunu tanımlarken sistem belleğini katmamaktadır, dolayısıyla, bir PLC denetim sistemi tasarlanırken yalnızca uygulama belleği boyutunun düşünülmesi yeterli olmaktadır.



Şekil-16 PLC bellek haritası.

Uygulama Belleği Uygulama belleği denetleme programı buyrukları ve denetim işlevlerini yerine getirmek için MİB'nin kullandığı veriyi saklar. Uygulama belleği haritası Şekil-17'de görülmektedir. Uygulama belleği iki ana alana bölünebilir: *Veri Tablosu* ve *Kullanıcı Programı*. Programlama buyrukları kullanıcı program alanında saklanırken, tüm veri, veri tablosunda saklanır.



Şekil-17 Uygulama bellek haritası.

Veri tablosu işlevsel olarak dört alana bölünebilir:

1. Veri saklama yazaçları
2. Giriş görüntü tablosu
3. Çıkış görüntü tablosu
4. İç saklama bitleri

Veri Saklama Yazaçları

Giriş/çıkış tablosunda saklanan bilgi "1" veya "0" ile belirtilen AÇ/KAPA durum bilgisidir. Tek bir "1" veya "0" ile belirtilemeyen bir değere sahip herhangi bir niceliği göstermek için bit grupları veya

bellek sözcükleri kullanılmalıdır. Değer bilgisini saklayan bellek sözcüklerine saklama yazaçları denilmektedir. Genelde üç çeşit saklama yazacı vardır: giriş yazaçları, sabitleri saklama yazaçları ve çıkış yazaçları. Toplam yazaç sayısı denetleyici bellek boyutuna ve veri tablosunun nasıl düzenlendiğine bağlı olarak değişmektedir. Saklama yazaçlarında saklanan değerler ikilik ya da İKO (BCD) biçimindedir. Her yazaç genellikle yüklenebilir, değiştirilebilir veya programlama birimini veya özel bir veri giriş aygıtını kullanarak görüntülenebilir.

Giriş yazaçları giriş arabirimleri yoluyla, iki konumlu anahtarları, mil kodlayıcıları veya İKO giriş sağlayan aygıtlardan alınan sayısal veriyi saklamak için kullanılır. Örneksel işaretler de giriş yazaçlarında saklanması gereken sayısal veriyi sağlarlar. Çeşitli örneksel transmitterler tarafından üretilen akım veya gerilim işareti örneksel arabirim tarafından sayısala dönüştürülür. Bu örneksel değerlerden ikilik gösterimler elde edilir ve belirtilen giriş yazacında saklanır. Giriş yazacındaki değer giriş aygıtı tarafından belirlenir ve dolayısıyla, denetleyici içinde veya diğer veri giriş aygıtları aracılığıyla değiştirilemez.

Sabitleri saklama yazaçları, buyruklar tarafından programın ürettiği değişken değerler (örneğin, matematik işlemci, zamanlayıcı veya sayıcı) veya programlama birimi veya başka veri giriş yöntemleri aracılığıyla girilen sabit değerleri saklarlar.

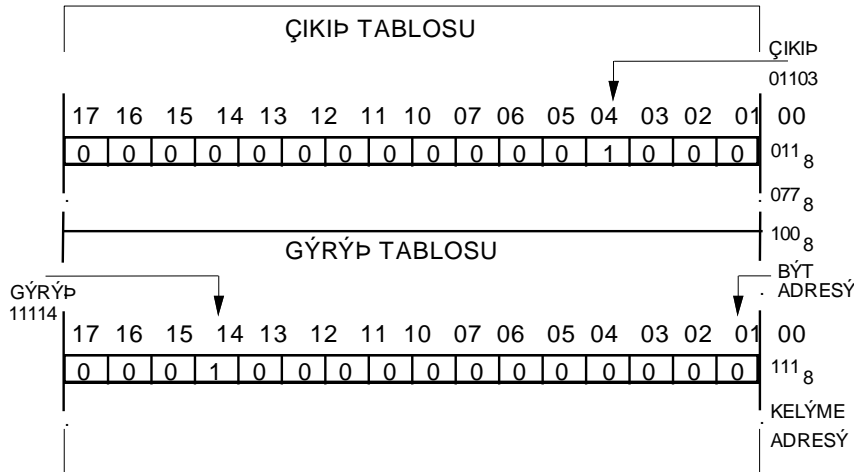
Çıkış yazaçları çeşitli çıkış aygıtlarını denetleyen sayısal veya örneksel değerleri saklamak için kullanılır. Çıkış yazaçlarının veri kabul eden bazı aygıtlar alfasayısal LED göstergeler, kaydediciler, örneksel ölçme aygıtları, hız denetleyiciler ve kontrol vanalarıdır.

G/Ç tabloları ve saklama yazaç alanlarının yanı sıra, bazı denetleyiciler veri tablosunun bir parçasını onluk, ASCII ve ikilik veriyi saklamak için ayırmıştır. Bu tablo alanı çözüm verisi, rapor üretme iletileri veya program işletimi sırasında saklanabilen veya getirilebilen diğer veri için kullanılır.

Giriş Görüntü Tablosu

Giriş görüntü tablosu giriş modüllerine bağlı süreçten alınan sayısal girişlerin durumunu saklayan bit dizimidir. Tablodaki bit sayısı en fazla giriş sayısına eşittir. En fazla 64 girişe sahip bir denetleyicinin 64 bitlik bir giriş tablosu olmalıdır. Her giriş, giriş tablosunda bağlı bulunduğu terminale karşılık gelen bir bite sahiptir. Eğer giriş ENERJİLİ ise, tabloda karşılık gelen biti de "1"dir. Eğer giriş ENERJİSİZ ise, tabloda karşılık gelen biti temizlenir ("0" ları).

Giriş tablosu bağlantılı giriş aygıtlarının durumunu yansıtmak için değişmektedir. Durum bilgisi aynı zamanda denetleme programı tarafından da kullanılmaktadır. Giriş tablosu Şekil-18' de verilmiştir.



Şekil-18 Giriş-Çıkış tabloları.

Çıkış Görüntü Tablosu

Çıkış tablosu çıkış arabirim devrelerine bağlı sayısal çıkış aygıtlarının durumunu denetleyen bir bit dizimidir. Çıkış tablosundaki bit sayısı en fazla çıkış sayısına eşittir. En fazla 512 çıkışa sahip bir denetleyicinin çıkış tablosunda 512 bit bulunmalıdır.

Bağlanmış her çıkış, çıkış tablosunda bağlı olduğu terminale tam olarak karşılık gelen bir bite sahiptir. MİB, çıkış tablosundaki bitleri denetleme programını yorumlarken ve G/Ç taraması sırasında günceller. Çıkış bitlerinin durumları bütün program işletildikten sonra değişir. Bir bit "1" olursa, bağlı çıkış da ENERJİLENİR. Bir bit temizlenirse veya "0" olursa, çıkışın da ENERJİSİ KESİLİR. Çıkış tablosu Şekil-18 'de verilmiştir.

İç Saklama Bitleri (Bellek Bitleri)

Pek çok denetleyici iç saklama bitleri için bir alan ayırmaktadır. Bu saklama bitlerine iç girişler veya iç bobinler de denilmektedir. İç çıkış, program mantığı tarafından denetlenen herhangi bir çıkış gibi çalışır; ancak, çıkış yalnızca iç mantık programlaması için kullanılır ve sürece giden bir çıkış doğrudan denetlemez. İç çıkışlar denetim programında İç çıkışlar, sayıcılar ve zamanlayıcıların "işlem tamam" bitlerini ve çeşitli iç mantık rölelerinin çıkışlarını kapsar. Denetleme programında bulunan bir adresteki her iç çıkış bitinin, aynı adreste bir saklama biti vardır. Denetim mantığı "DOĞRU" olduğunda, iç çıkış saklama biti "1" olur.

Uygulama belleğinin kullanıcı program bellek alanı denetim programını saklamak için kullanılır. Makine ya da süreci denetleyen tüm buyruklar burada saklanır. Gerçek ve iç G/Ç bitlerinin adresleri belleğin bu bölümünde tanımlanır. PLC çalışma modunda olduğunda ve denetim programı işletildiğinde MİB bu bellek yerlerini yorumlar ve veri tablosunda her gerçek veya iç G/Ç bitine karşılık gelen bitleri denetler.

Denetleyicinin bellek boyutu, kullanılabilir bellek miktarını belirler. Orta ve büyük çaplı PLC'lerde, kullanıcı program boyutuna göre ve en düşük veri saklama gereksinimlerini karşılayacak şekilde, veri tablosu boyutu değiştirilir. Küçük çaplı PLC'lerde ise, kullanıcı program boyutu sabittir.

Güç Kaynakları Güç kaynakları 240V ac gücü işlem için gerekli 5V ve 24V dc gerilime dönüştürür. Bir güç kaynağı belirli sayıdaki giriş ve çıkışlara yeterli güç sağlamalıdır. Erklenen giriş ve çıkış sayısına bağlı olarak yük değişecektir, fakat bir güç kaynağı maksimum koşulu idare edebilecek şekilde boyutlandırılmalıdır.

3 PLC Programlama Dilleri

Bir PLC özel amaçlı bir bilgisayardır ve belirli özel amacını yerine getirmek için programlanmalıdır. Programlama dili kullanıcının bir programlama aygıtı aracılığıyla PLC ile iletişim kurmasını sağlar. Bir PLC, programlamanın yapılması için bir tuş takımı ve sıvı kristal göstergeye sahip olmalıdır. Günümüzde PLC'lerin programlanması için bilgisayarların kullanılması yaygınlaşmıştır.

PLC üreticileri farklı programlama dilleri kullanmaktadır, fakat hepsi buyruklar aracılığıyla sistemi denetlemek için aynı mantığı kullanmaktadır.

Programlanabilir denetleyicilerde en çok kullanılan dört dil vardır:

1. Merdiven diyagramları
2. Boole kısa adları
3. İşlev blokları
4. İngilizce buyruklar

Bu diller iki ana sınıfta toplanabilir. İlk ikisi temel PLC dillerini oluştururken, işlev blokları ve İngilizce buyruklar yüksek seviyeli dillerdir. Temel PLC dilleri en ilkel denetim işlevlerini yerine getiren buyruk setinden oluşur. Bunlar rölelerin yerleştirilmesi, zamanlama, sayma, ardışık işlemler ve mantık işlevleridir. Ancak denetleyici modeline göre, buyruk seti diğer temel işlemleri gerçekleştirmek için genişletilebilir veya artırılabilir. Bu dillerde programlama bir sıvı kristal gösterge (LCD) ekran üzerinde bir grafik resim düzenleyerek yapılır. Bu grafik resim röle merdiven diyagramına çok benzemektedir. Bu çeşit bir dil röle mantığı ile çalışmış olan tasarım mühendisleri ve teknisyenlerinin kullanma ve yorumlamaya alışkın olduğu bir dildir. Diyagramlar devrelerde neler olduğunu belirtmek için gözlenebilir ve bilgisayar programları ile belgelenebilir. Belgeleme işlemi

girilen programın ASCII karşılığının bir çıktısı şeklinde olabildiği gibi, içindeki çeşitli aygıtların etiketlerini belirtmek için kısa adlarıyla beraber merdiven diyagramının bir çıktısı da alınabilir.

Programlama Boole mantığını kullanarak da yapılabilir. Boole mantığını kullananlar için bu anlaşılabilir bir dildir. Ancak röle ve merdiven diyagramlarla yıllarca uğraşmış mühendis ve teknisyenler her zaman Boole mantığına alışmamışlardır; Boole cebirini kullanarak mantık tasarımı öğrenen öğrenciler ise, bir mantık diyagramının ne olduğunu anlamakta zorlanmaktadır.

Yüksek seviyeli diller basit zamanlama, sayma ve aç/kapa denetimi haricinde daha güçlü buyrukları işletmek için kullanılmaktadır. Yüksek seviyeli diller örneksel denetim, veri işleme, rapor üretme ve temel buyruk setiyle gerçekleştirilemez diğer işlevler için kullanılmaktadır.

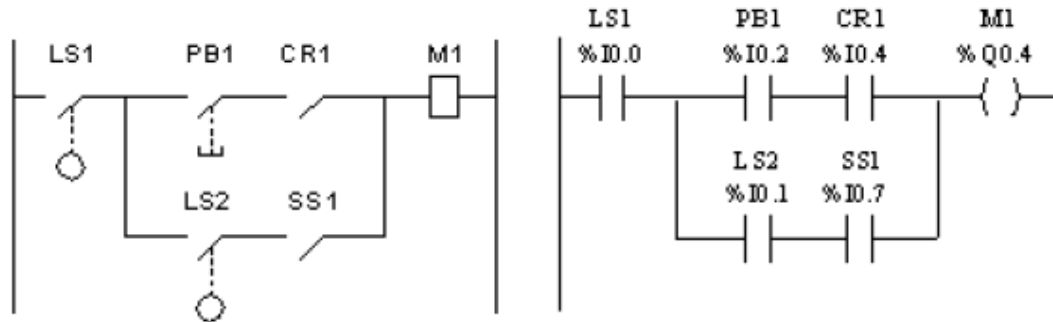
Bir PLC’de kullanılan dil gerçekte denetleyicinin kullanılabileceği çalışma sınırlarını belirtir. Denetleyicinin boyutu ve yeteneklerine bağlı olarak bir veya daha fazla dil kullanılabilir. Dil birleşimleri şunlar olabilir:

1. Yalnızca merdiven diyagramları
2. Yalnızca Boole mantığı
3. Merdiven diyagramları ve işlev blokları
4. Boole mantığı ve işlev blokları
5. Boole mantığı ve İngilizce buyruklar

3.1 Merdiven Diyagram Programlaması

“Merdiven diyagramı” adını elemanlarını bir merdivenin basamaklarına benzer, izleyen hatlar üzerinde gösteren şematik bağlantı çiziminden almaktadır.

Şekil-19’da şematik bağlantının bir merdiven diyagram gösterimi bulunmaktadır. Güç ve nötr yolları merdivenin kenar parmaklıklarını oluşturur ve bir mantık sistemini gerçekleyen elektromekanik aygıtların aktif elemanları merdivenin basamakları üzerinde gösterilmektedir. Basamakları soldan sağa ve yukarıdan aşağıya doğru izleyerek işlem sırası ortaya çıkarılabilir. Ek olarak, terminaller gösterilip işaretlenirse, gerçek enterkonnekte kablolar tanımlanabilir.



Şekil-19 Kumanda devresi ve merdiven diyagram programı.

Bir mantık devresini oluşturan bileşenler anahtarlar, kontaklar, bobinler ve devrelerdir. Bunlar şematik çizimde uygun yerlerdeki bileşenleri temsil eden simgelerle gösterilmektedir.

Bobinler ve kontaklar merdiven diyagram buyruk setinin temel simgelerini oluştururlar. Bir basamaktaki programlanmış kontak simgeleri çıkışın denetimini sağlamak için değerlendirilecek koşulları temsil eder. Tablo-2’de kontak simgeleri özetlenmiştir.

Programlandığında her bir kontak ve bobin bir adresle tanımlanır. Bu adresler ya bir iç çıkış ya da bağlı bir giriş veya çıkışın veri tablosundaki yerini belirtir. Bir kontak, ya bir girişi ya da bir iç çıkışı temsil etmesine bakılmaksızın o koşulun değerlendirilmesi gerektiğinde kullanılabilir.

Basamak kontaklarının biçimi istenilen denetim mantığına bağlıdır. Kontaklar belirli bir çıkışı elde etmek için seri, paralel veya seri/paralel olarak bağlanabilirler. Bir çıkışın enerjilendirilmesi için en azından bir soldan-sağa yoldaki kontakların kapalı olması gerekir. Tamamen kapalı bir yolda “mantık

devamlılığı vardır” denir. “Mantık devamlılığı” (en azından bir yolda) oluştuğunda, basamak koşulu “DOĞRU”dur. Yolun devamlılığı yoksa basamak koşulu “YANLIŞ”tır.

Tablo-2 Kontaklar

| İşlem Türü | Temel Simgesi | Buyruk | Tanım |
|---------------------------|---------------|--------|-----------------------------|
| Röle mantığı Kontaklar | | LD | Normalde açık kontak (NA) |
| | | LDN | Normalde kapalı kontak (NK) |
| | | LDR | Yükselen kenar kontak |
| | | LDF | Düşen kenar kontak |

3.1.1 Röle-Tipi Buyruklar

Bu buyruklar merdiven diyagramının en temel buyruklarıdır. İletkenle bağlantı yapılmış röle mantığı ile aynı yeterlikleri daha fazla esneklikle sağlamaktadır. Bu buyruklar öncelikle bellekte adreslenmiş özel bir bitin AÇ/KAPA durumunu inceleyerek bir iç veya dış çıkışın durumunu denetlerler.

Normalde Açık Kontak (NA)

Normalde açık kontak ilgili işaretin varlığının bir çıkışı enerjilendirmesi gerektiğinde kullanılır. Değerlendirildiğinde, ilgili adres bir AÇ (1) koşulu için incelenir. Bu adres bir dış giriş veya iç program bitinin durumunu belirtebilir. İncelendiğinde, ilgili adres ON (1) ise, o zaman normalde açık kontak kapanır ve mantık devamlılığını (güç akışını) sağlar. Eğer OFF (0) ise, o zaman normalde açık kontak programlanmış durumunu (açık) koruyacak ve mantık devamlılığını bozacaktır.

Normalde Kapalı Kontak (NK)

Normalde kapalı kontak ilgili işaretin olmamasının bir çıkışı enerjilendirmesi gerektiğinde kullanılır. İncelendiğinde, ilgili adres bir KAPA (1) koşulu için denetlenir. Bu adres bir dış giriş veya iç program bitinin durumunu belirtebilir. İncelendiğinde, ilgili adres OFF (0) ise, o zaman normalde kapalı kontak kapalı kalarak mantık devamlılığını koruyacaktır. İlgili adres ON (1) ise, o zaman normalde kapalı kontak açılacak ve mantık devamlılığını bozacaktır.

Yükselen Kenar Kontak (Rising Edge)

Yalnızca gerçek girişler için kullanılan bu buyruk girişteki DÜŞÜK seviyeden YÜKSEK seviyeye geçişte bir PLC tarama çevrimi süresince MANTIK 1 seviyesinde kalmaktadır. Bir yükselen kenar geçişinde işlem yapılacağı zaman programlanır.

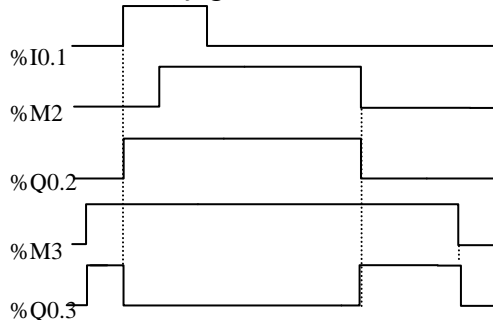
Düşen Kenar Kontak (Falling Edge)

Yalnızca gerçek girişler için kullanılan bu buyruk girişteki YÜKSEK seviyeden DÜŞÜK seviyeye geçişte bir PLC tarama çevrimi süresince MANTIK 1 seviyesinde kalmaktadır. Bir düşen kenar geçişinde işlem yapılacağı zaman programlanır.

Değil Buyruğu : N

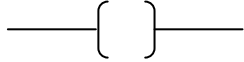
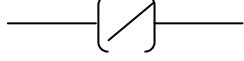
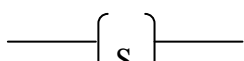
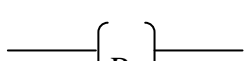
Bu buyruk önceki buyruğun Boole işlem sonucunun değilini alır. İşleneni yoktur. Sadece deyim listesinde bulunan bu buyruğun kullanımını aşağıdaki örnekte verilmiştir.

| | |
|-----|-------|
| LD | %I0.1 |
| OR | %M2 |
| ST | %Q0.2 |
| N | |
| AND | %M3 |
| ST | %Q0.3 |



Çıkış bitlerinin kontrolü için kullanılan merdiven mantık programlama sembolleri “bobin” olarak adlandırılmaktadır ve Tablo-3’te bu buyruklar ve sembolleri gösterilmiştir.

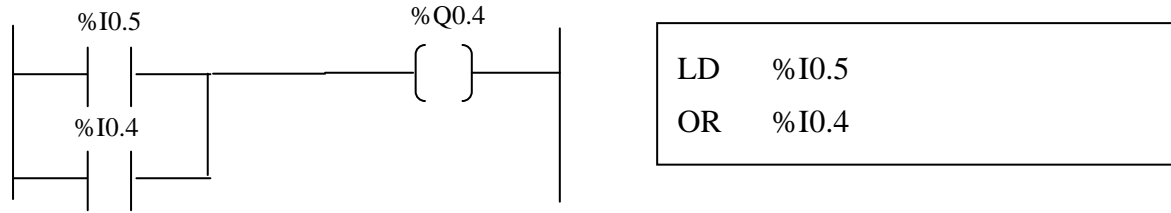
Tablo-3 Bobinler

| İşlem Türü | Temel Simgesi | Buyruk | Tanım |
|--------------|---|--------|-------------------------|
| Röle mantığı |  | ST | Çıkış rölesi |
| Bobinler |  | STN | Terslemeli çıkış rölesi |
| |  | S | Kilitlemeli çıkış (SET) |
| |  | R | Bırakma çıkışı (RESET) |

Çıkış Bobini (Rölesi)

Çıkış bobini buyruğu ya denetleyiciye bağlı bir çıkışı ya da iç denetim bitini denetlemek için programlanır. Herhangi bir basamak mantık devamlılığına sahipse, ilgili çıkış enerjilenir. Mantık devamlılığı bozulursa çıkışın enerjisi kesilir. Çıkış enerjili olduğunda aynı adresteki normalde açık kontak kapanır, normalde kapalı kontak açılır.

Çıkışın enerjisi kesilirse, herhangi bir normalde açık kontak açılır, normalde kapalı kontak kapanır. Şekil-20’de örnek bir merdiven basamağı gösterilmektedir. Bu örnekte, %I0.5 ya da %I0.4 girişlerinden biri DOĞRU ise, %Q0.4 çıkışı DOĞRU olacaktır.



Şekil-20 Program örneği

Tersleyen Çıkış

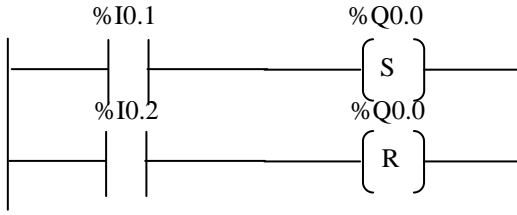
Tersleyen çıkış bobini ile programlandığında, herhangi bir basamak mantık devamlılığına sahipse, ilgili çıkışın enerjisi kesilir. Mantık devamlılığı bozulursa çıkış enerjilenir.

Kilitlemeli Çıkış (Latched/Set Output)

Kilitlemeli çıkış buyruğu çıkışı enerjilendiren kontakların durumu değişse bile çıkışın enerjili kalmasını sağlamak için kullanılır. Herhangi bir basamak yolu mantık devamlılığına sahipse, çıkış enerjilenir ve mantık devamlılığı veya sistem gücü kesilse bile enerjili kalır. Aynı adresteki bir bırakma (unlatch) çıkış buyruğu ile bırakılana kadar çıkış enerjili kalır. Pek çok denetleyici hem iç hem de dış çıkışları kilitlemeye izin verirken bazıları yalnızca içi çıkışların kilitlemesine izin vermektedir.

Bırakma Çıkışı (Unlatch/Reset Output)

Bırakma çıkışı buyruğu aynı adresteki kilitlemeli çıkışı resetlemek için programlanır. Mantık devamlılığı sağlandığında ilgili adres “0”lanır. Bırakma çıkışı kilitlemeli çıkışın otomatik olarak resetlenmesi için kullanılır. Şekil-22’de tutma ve bırakma bobinlerinin kullanımı görülmektedir. Bu örnekte, %I0.1 biti %Q0.0 çıkışını kurmak için DOĞRU olmalıdır. Bu bit YANLIŞ olursa, %I0.2 biti DOĞRU olana kadar çıkış rölesi enerjili kalacaktır. Çoğu PLC’de bırakma bobini kilitleme bobininden önce yerleştirilir çünkü program merdiven diyagramının başından sonuna doğru taranır ve giriş görüntü tablosu program taramasının sonunda güncellenir.



Şekil-22 Set ve reset röleleri.

3.2 Deyim Listesi ile Programlama

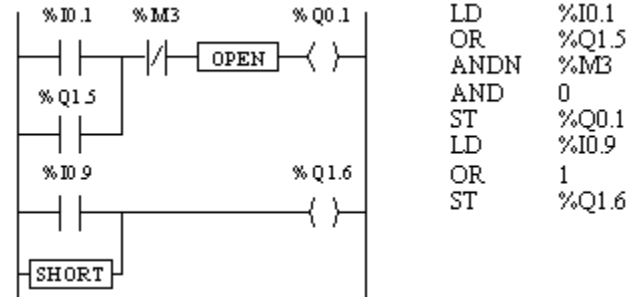
3.2.1 Özel Merdiven Diyagramı Buyrukları: OPEN ve SHORT

Merdiven diyagramı programlayıcısına kolaylık olsun diye hata bulma (debugging) ve arıza giderme (troubleshooting) işlemleri sırasında kullanılacak iki buyruk vardır. SHORT buyruğu merdiven basamağının (rung) sürekliliğini kısaltır. OPEN buyruğu merdiven basamağının sürekliliğini açar. Böylece merdiven basamağının mantığını değiştirmeyi sağlarlar.

Tablo-4 OPEN ve SHORT buyruklarının kullanımı

| Buyruk | List buyruğu | Tanım |
|--------|--------------|---|
| OPEN | AND 0 | En son mantık işleminin sonucunu dikkate almadan merdiven basamağının sürekliliğini açar. |
| SHORT | OR 1 | En son mantık işleminin sonucunu dikkate almadan merdiven basamağının sürekliliğini sağlar. |

Örnekler: Aşağıdaki program OPEN ve SHORT buyruklarının kullanımına örnektir.

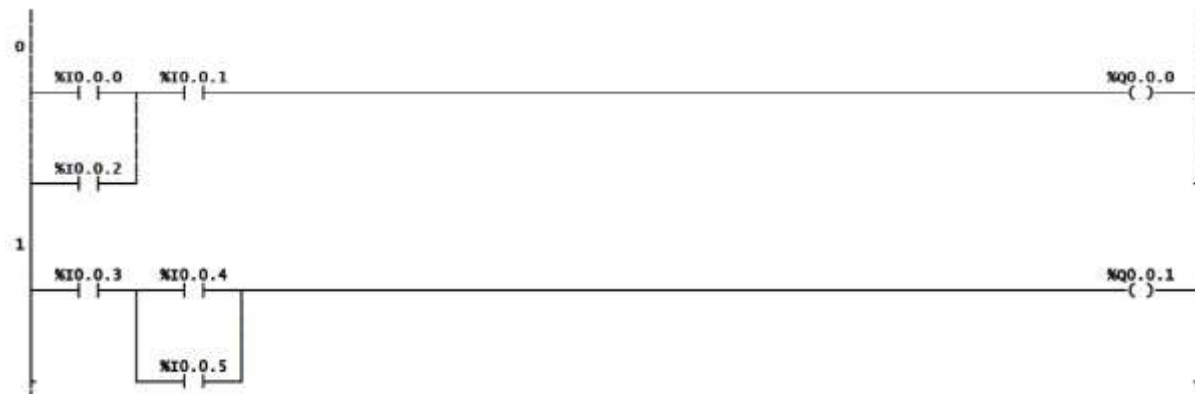


Şekil-23 OPEN ve SHORT buyruklarının kullanımına örnek program.

3.2.2 Parantez Kullanma

AND ve OR buyrukları parantez kullanabilir. Parantezler Merdiven diyagramlarındaki paralel dallar için kullanılır. Parantez açma işlemi AND ve OR buyrukları ile birlikte kullanılır. Parantezi kapama, her açık parantez için ayrı bir satırda kullanılması gereken bir buyruktur.

Örnek: AND (...)



```

0 LD    %I0.0.0
1 OR    %I0.0.2
2 AND   %I0.0.1
3 ST    %Q0.0.0
4 LD    %I0.0.3
5 AND(  %I0.0.4
6 OR    %I0.0.5
7 )
8 ST    %Q0.0.1

```

Örnek: OR (...)



```

0 LD    %I0.0.0
1 AND   %I0.0.1
2 OR(   %I0.0.2
3 AND   %I0.0.3
4 )
5 ST    %Q0.0.0

```

Şu deęiřtiriciler parantezlere atanabilir: N, F, R, [

- N deęilini alma. Örneęin AND (N veya OR (N
- R yükselen kenar. Örneęin AND (R veya OR (R
- F yükselen kenar. Örneęin AND (F veya OR (F
- [karşılařtırma

Ladder Diagram



List Instructions

```

0 LD    %I0.0.0
1 AND   %I0.0.1
2 OR(N  %I0.0.2
3 AND   %I0.0.3
4 )
5 ST    %Q0.0.0

```

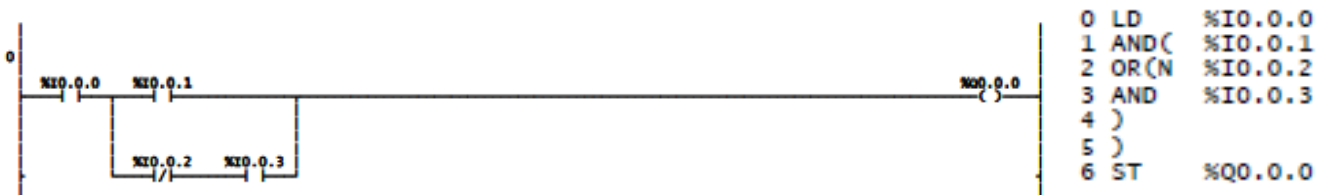
Şekil-24 Parantez içinde deęiřtiricilerin kullanımına örnek program

3.2.3 İç içe parantez açmak

8 taneye kadar iç içe parantez açmak mümkündür.

Not:

- Açılan her parantez kapatılmalıdır.
- %Li etiketleri ve SRi altyordamları parantez içindeki ifadelere yerleřtirilmemelidir. Atlama (JMP), altyordam (SRi) ve fonksiyon blokları için de aynı kural geçerlidir.
- Atama buyrukları (ST, STN, S ve R) parantez içinde programlanmamalıdır.



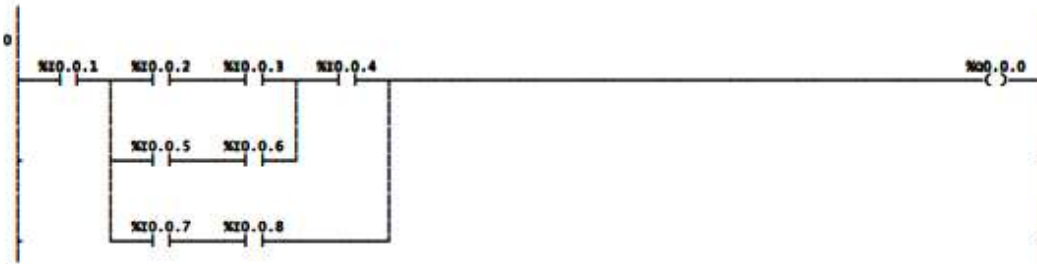
```

0 LD    %I0.0.0
1 AND(  %I0.0.1
2 OR(N  %I0.0.2
3 AND   %I0.0.3
4 )
5 )
6 ST    %Q0.0.0

```

(a)

Ladder Diagram



List Instructions

```

0 LD %IO.0.1
1 AND( %IO.0.2
2 AND %IO.0.3
3 OR( %IO.0.5
4 AND %IO.0.6
5 )
6 AND %IO.0.4
7 OR( %IO.0.7
8 AND %IO.0.8
9 )
10 )
11 ST %QO.0.0

```

(b)

Şekil-25 İç içe parantez kullanımına örnek programlar.

3.2.4 MPS, MRD, MPP buyrukları

Bu üç tip buyruk çıkışlara yönlendirmeyi gerçekleştirmek için kullanılır. Bu buyruklar 8 Boole ifadesine kadar depolama yapabilen ve yığın (stack) denilen depolama alanlarını kullanırlar.

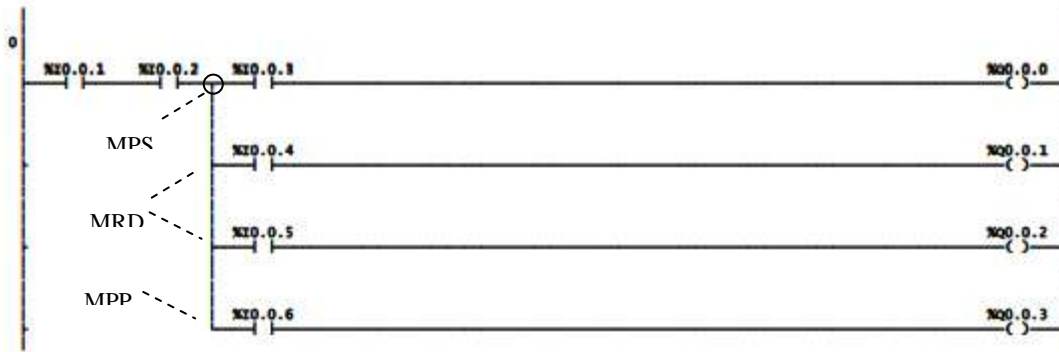
MPS (**M**emory **Pu**Sh) buyruğu akümülatörü (en son mantık işleminin sonucu) yığının en tepesine itip (yazıp) diğer değerleri aşağıya doğru kaydırır.

MRD (**M**emory **O**kuma) buyruğu yığının tepesini akümülatöre okur (kopyalama işlemi).

MPP (**M**emory **Po**P) buyruğu yığının tepesini akümülatöre taşıyıp diğer değerleri yığının tepesine doğru kaydırır.

Örnekler:

Ladder Diagram



List Instructions

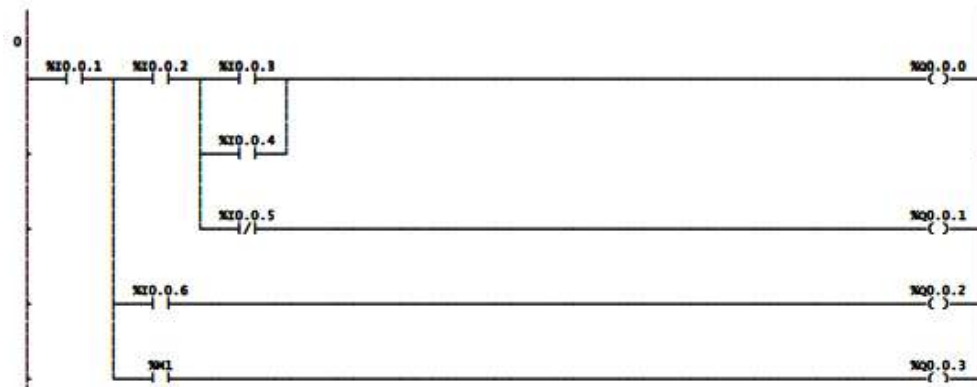
```

0 LD %IO.0.1
1 AND %IO.0.2
2 MPS
3 AND %IO.0.3
4 ST %QO.0.0
5 MRD
6 AND %IO.0.4
7 ST %QO.0.1
8 MRD
9 AND %IO.0.5
10 ST %QO.0.2
11 MPP
12 AND %IO.0.6
13 ST %QO.0.3

```

(a)

Ladder Diagram



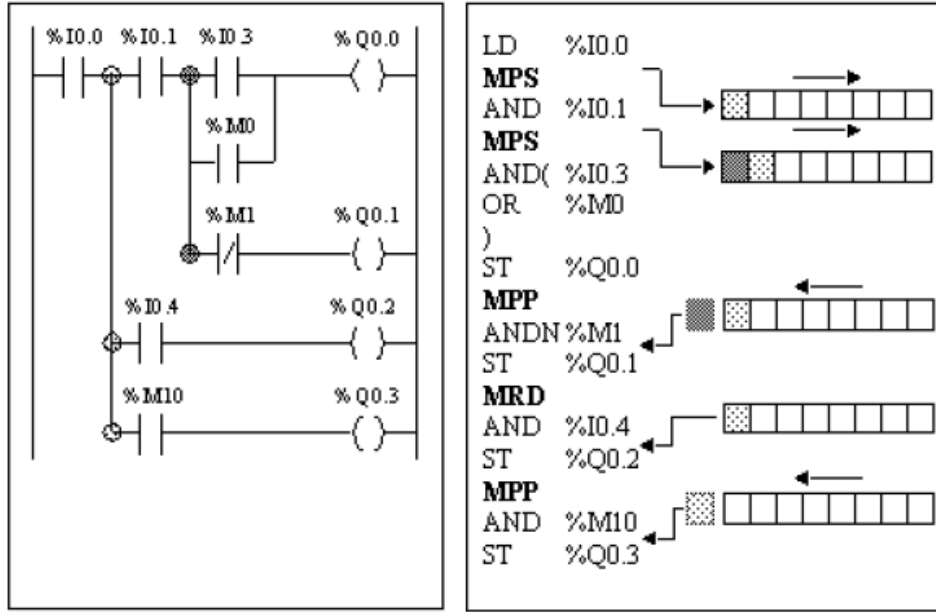
List Instructions

```

0 LD %IO.0.1
1 MPS
2 AND %IO.0.2
3 MPS
4 AND( %IO.0.3
5 OR %IO.0.4
6 )
7 ST %QO.0.0
8 MPP
9 ANDN %IO.0.5
10 ST %QO.0.1
11 MRD
12 AND %IO.0.6
13 ST %QO.0.2
14 MPP
15 AND %M1
16 ST %QO.0.3

```

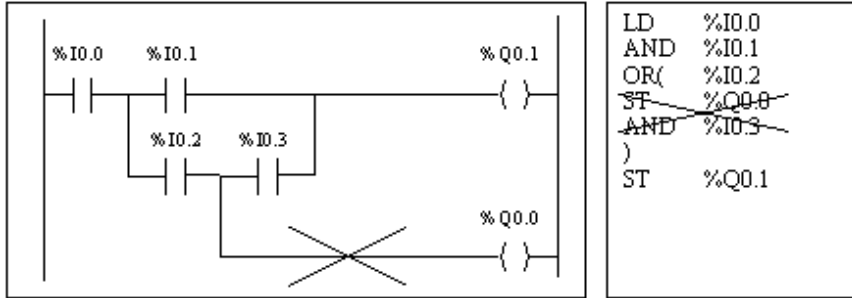
(b)



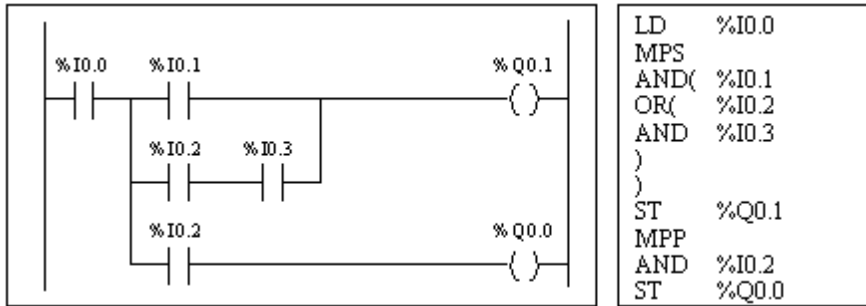
(c) Şekil-26 MPS, MRD ve MPP buyruklarının kullanımına (a,b) ve çalışmasına (c) örnek programlar.

3.2.5 Parantez kullanımına ilişkin ek notlar

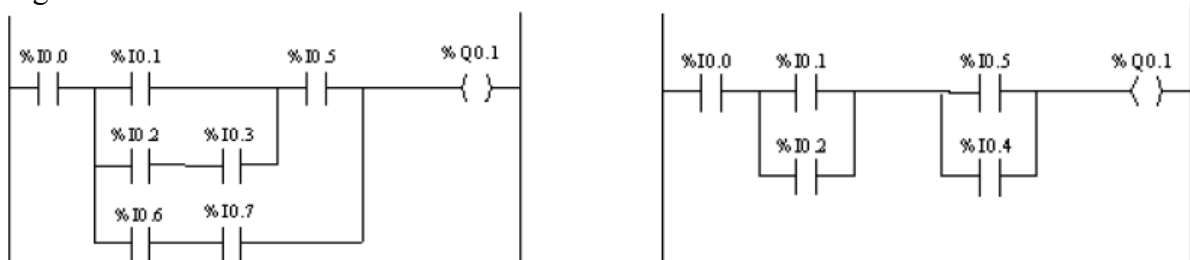
Çıkış atama işlemleri parantez içinde yapılamaz.



Şekil-27 Parantezin yanlış kullanımına örnek program. Yukarıdaki program aşağıda gösterildiği gibi yeniden yazılmalıdır.

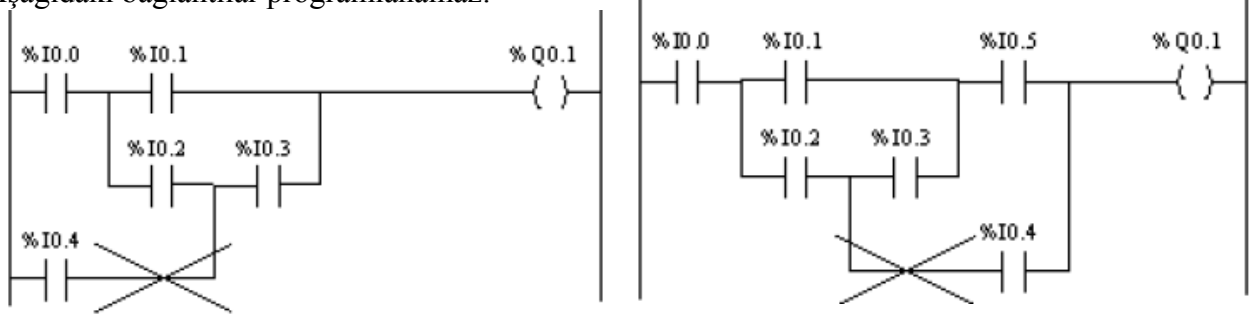


Şekil-28 Parantezin doğru kullanımına örnek yeniden yazılmış olan program. Eğer çok sayıda kontak paralel bağlanacaksa, aralarında uygun bir ağ oluşturularak veya ayrı ayrı programlanmalıdır.

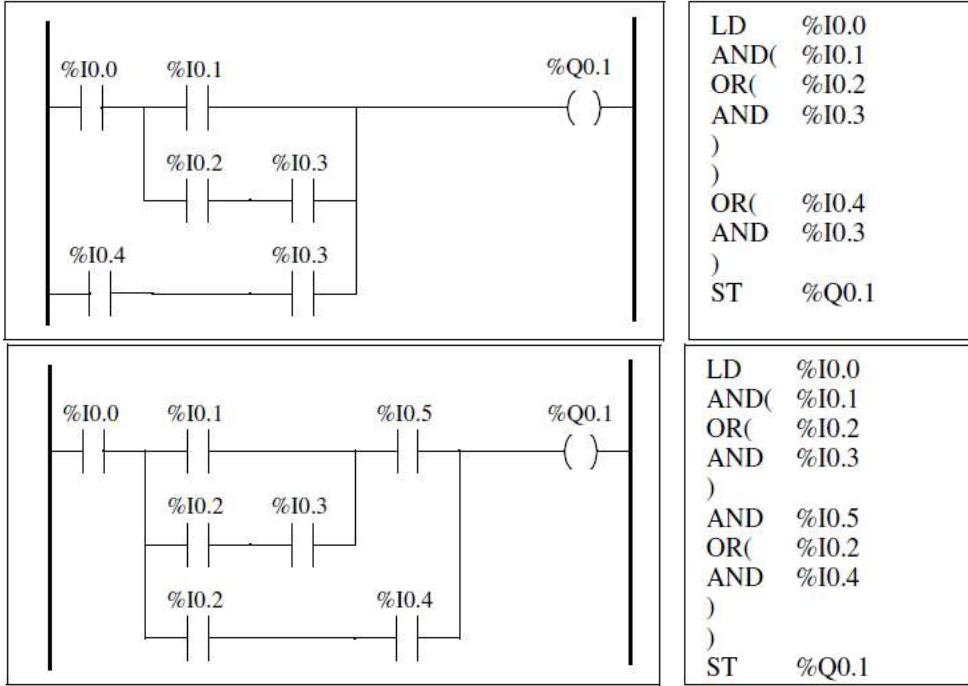


Şekil-29 Parantez kullanımına örnek programlar.

Aşağıdaki bağlantılar programlanamaz:



Şekil-30 Yanlış kontak bağlantısına örnek programlar.
Programlanabilmeleri için aşağıdaki gibi düzenlenmelidirler.



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

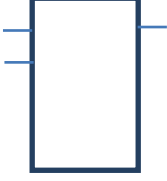
```
LD %I0.0
AND( %I0.1
OR( %I0.2
AND %I0.3
)
AND %I0.5
OR( %I0.2
AND %I0.4
)
)
)
ST %Q0.1
```

Şekil-31 Kontaktarın doğru bağlantıları ve parantez kullanımına örnek programlar.
Test buyruklarının özeti Tablo-5’te verilmiştir.

Tablo-5 Test buyrukları

| List buyruk | Temel Simgesi | Fonksiyon |
|-------------|---------------|---|
| LD | | Boole işlem sonucu, işlenen değeri ile aynıdır. |
| LDN | | Boole işlem sonucu, işlenen değerinin değiline eşittir. |
| LDR | | Boole işlem sonucu, işlenen değerinin yükselen kenarında 1 olur. |
| LDF | | Boole işlem sonucu, işlenen değerinin düşen kenarında 1 olur. |
| AND | | Boole işlem sonucu, önceki buyruğun Boole sonucu ile işlenenin durumu arasındaki AND mantığına |
| ANDN | | Boole işlem sonucu, önceki buyruğun Boole sonucu ile işlenenin değili arasındaki AND mantığına eşittir. |

Tablo-8 Fonksiyon blokları

| İşlem Türü | Temel Simgesi | Fonksiyon |
|---|---|--|
| Zamanlayıcılar, sayıcılar, yazaçlar ve diğerleri. |  | Her bir fonksiyon bloğu diğer grafik elemanlarına bağlantı sağlayan girişler ve çıkışları kullanır. Not: Fonksiyon bloklarının çıkışları birbirine bağlanmaz. |

3.3 Fonksiyon Bloklarına Giriş

3.3.1 Zamanlayıcı ve Sayıcı Fonksiyon Blokları

Tablo-8’de genel simgesi görülen zamanlayıcı ve sayıcılar donanımsal sayıcılar ve zamanlayıcılar ile aynı işlevleri sağlarlar. Bir çıkış aygıtının belirli bir zaman gecikmesinden sonra enerjilendirilmesi ya da enerjisinin kesilmesi için kullanılırlar. Zamanlayıcı ve sayıcı çıkış buyrukları iç çıkışlar olarak tanımlanırlar. Röle tipi buyruklar gibi, zamanlayıcı ve sayıcı fonksiyon blokları da merdiven diyagramının temel buyruklarını oluşturur.

Zamanlayıcı ve sayıcıların çalışmaları birbirine çok benzemektedir, çünkü her ikisi de sayma işlemi yapar. Zamanlayıcı sabit bir zaman tabanında sayma yapar (1ms, 10ms, 100ms, 1s, 1 dak). 3 saniyelik zaman gecikmesi için zamanlayıcı üç adet 1s’lik zaman aralığını (zaman tabanı) sayar. Sayıcı bir olayın gerçekleşmesini sayar. Zamanlayıcı ve sayıcı buyrukları sayılan sayma değerinin saklanması için bir mevcut değer yazacı (sözcük yeri, %TMi.V ile %Ci.V) ve önayar değerinin saklanması için de bir önayar yazacı (%TMi.P ile %Ci.P) kullanır. Önayar değeri olayların ya da zaman-tabanlı aralıkların en fazla sayısını belirtir.

3.3.1.1 Zamanlayıcı Fonksiyon Bloğu

Zamanlayıcı ENERJİLİ (TON)

ENERJİLİ zamanlayıcı tipi bir olayı zaman-gecikmeli gerçekleştirmek yada bir olayın süresini ölçmek için kullanılır. Bir basamağın mantık devamlılığı sağlandığında, zamanlayıcı zaman-tabanlı aralıkları saymaya başlar ve akümülatördeki zaman değeri önayar (%TMi.P) değerine eşit olana kadar saymaya devam eder. Mevcut değer yazacındaki değer önayar değerine eşit olduğunda, çıkış enerjilenir ve zamanlayıcı çıkışının kontağı (“işlem tamam” biti, Q) kapanır. Zamanlayıcı kontağı bir mantık işlemini yerine getirmek için programda normalde açık (NA) veya normalde kapalı (NK) kontak olarak kullanılabilir.

Zamanlayıcı ENERJİSİZ (TOF)

ENERJİSİZ zamanlayıcı tipibir olayı zaman-gecikmeli olarak gerçekleştirmek için kullanılır. Bir basamağın mantık devamlılığı yitirildiğinde (düşen kenarda), zamanlayıcı zaman-tabanlı aralıkları saymaya başlar ve akümülatördeki zaman değeri önayar değerine eşit olana kadar saymaya devam eder. Mevcut değer yazacındaki bu değer önayar değerine eşit olduğunda, çıkışın enerjisi kesilir ve zamanlayıcı çıkışının kontağı açılır. Zamanlayıcı kontağı bir mantık işlemini gerçekleştirmek için programda NA veya NK kontak olarak kullanılabilir. Zamanlayıcı önayar değerine ulaşmadan önce mantık devamlılığı sağlanırsa, akümülatör sıfırlanır (zamanlayıcı başa kurulu).

Tutmalı Zamanlayıcı ENERJİLİ (TP)

Tutmalı zamanlayıcı çıkış buyruğu mantık devamlılığının bozulması ya da gücün kesilmesi durumlarında zamanlayıcının akümülatördeki değerinin korunması istendiğinde kullanılır. Mantık devamlılığı sağlandığında zamanlayıcı akümülatördeki değerden başlayarak ön-kurma değerine kadar sayar. Mevcut değer yazacındaki değer önayar değerine ulaştığında, çıkış enerjilenir ve zamanlayıcı çıkışının kontağı açılır. Zamanlayıcı kontağı programda NA veya NK kontak olarak kullanılabilir.

3.3.1.2 Yukarı/Aşağı Sayıcı Fonksiyon Bloğu

Yukarı Sayıcı girişi veya buyruğu (CU)

Yukarı sayıcı giriş buyruğu sayılan olay gerçekleştiğinde sayıcı sözcüğünü (%Ci.V) bir artırır. Bir denetim uygulamasında sayıcının görevi belirli bir değere ulaşıldığında bir aygıtı kontrol etmektir. Örneğin, belirli bir noktadan geçen şişelerin sayısının belirlenmesi için bir sayıcı kullanılsın. Yukarı sayıcı her şişenin geçişinde oluşan “0”dan “1”e geçişi sayarak içeriğini bir artırır. Toplam değer sayıcının önayar değerine ulaştığında, bu sayıcının kontağı kapanır. Kontrol aygıtına bağlı olarak, sayıcı son sayma değerine ulaştığında sıfırlanır veya her “0”dan “1”e geçişte bir artmaya devam eder. Eğer bu sonraki durum söz konusu ise, akümülatörü sıfırlamak için bir reset buyruğu gereklidir.

Aşağı Sayıcı girişi veya buyruğu (CD)

Aşağı sayıcı giriş buyruğu sayılan olay gerçekleştiğinde sayıcı sözcüğünü bir azaltır. Bu olay her gerçekleştiğinde, akümülatördeki değer (%Ci.V) azalır. Aşağı sayıcı genellikle yukarı sayıcı ile birlikte kullanılarak bir aşağı/yukarı sayıcı oluşturulur.

Sayıcıcı Reset girişi veya buyruğu (R)

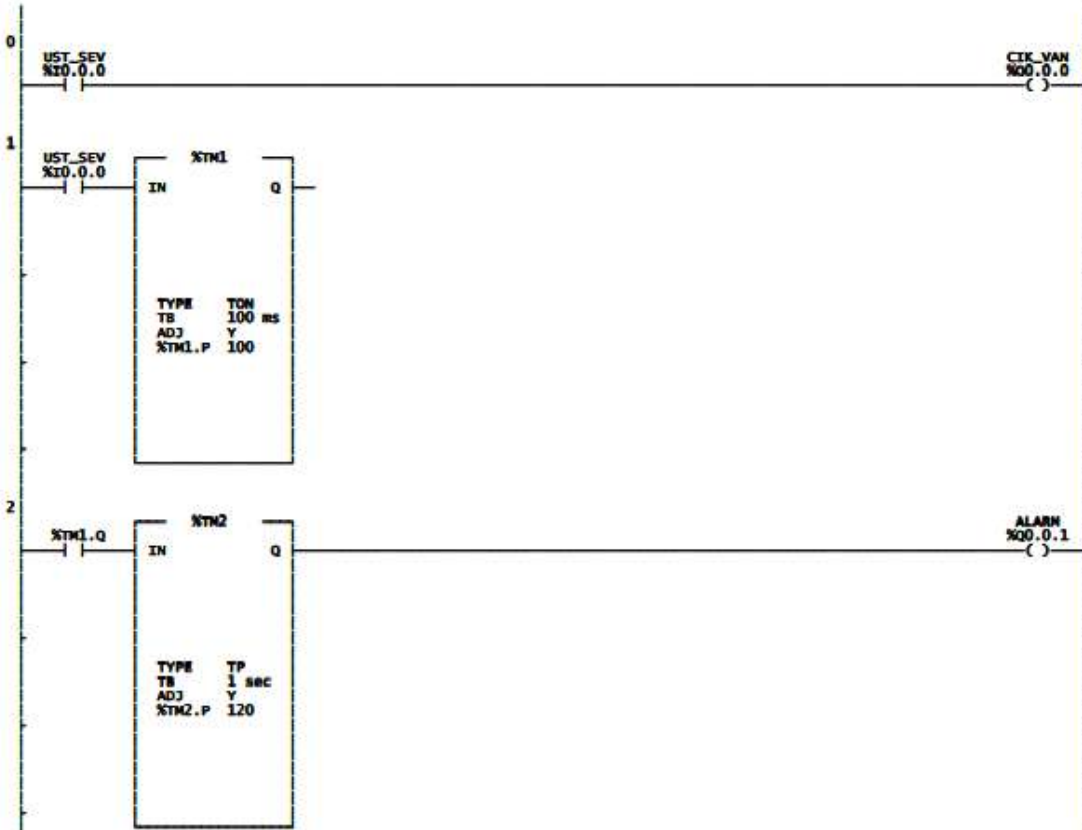
Sayıcıcı reset buyruğu CU son değerinin sıfırlanması için kullanılır. Programlandığında, Reset buyruğu yada girişinin basamak koşulu DOĞRU ise, sayıcı mevcut değeri (%Ci.V) sıfırlanır.

Sayıcıcı Set girişi veya buyruğu (S)

Sayıcıcı set buyruğu ile aşağıya doğru saydırılacak sayıcıya başlangıç değeri yüklenir. Programlandığında, set buyruğu ya da girişinin basamak koşulu DOĞRU ise, sayıcı önayar değeri (%Ci.P) mevcut değer sözcüğüne (%Ci.V) yüklenir.

ÖRNEK PROGRAM: Bir tanktaki üst seviyenin aşılmasıyla NK çıkış vanasını açan ve 10s sonra 3 dak. süreyle bir alarmı çalıştıran programın sembol tablosu, merdiven diyagramı ve deyim listesinin de yazıldığı zamanlayıcı örnek programı aşağıda verilmiştir.

Ladder Diagram



List Instructions

```
0 LD    %IO.0.0
1 ST    %Q0.0.0
2 BLK   %TM1
3 LD    %IO.0.0
4 IN
5 END_BLK
6 BLK   %TM2
7 LD    %TM1.Q
8 IN
9 OUT_BLK
10 LD   Q
11 ST   %Q0.0.1
12 END_BLK
```



Symbol Table

| Address | Symbol | Comment |
|---------|---------|---------|
| %IO.0.0 | UST_SEV | |
| %Q0.0.1 | ALARM | |
| %Q0.0.0 | CIK_VAN | |

3.4 Sayısal Komutlar

Sayısal komutlar genelde 16 bitlik sözcüklere uygulanır. Deyim listesinde köşeli parantezler arasına yazılırken merdiven diyagramda Tablo-7’de gösterilen işlem blokları ile programlanırlar.

Tablo-7 İşlem ve Karşılaştırma Blokları

| İşlem Türü | Temel Simgesi | Fonksiyon |
|---------------------|--|--|
| Karşılaştırma bloğu |  | İki işleneni karşılaştırır, sonuç doğruysa çıkış “1” olur. |
| İşlem bloğu |  | Aritmetik ve mantık işlemlerini gerçekleştirir. |

Veri Atama İşlemleri Bir yazacın içeriğini diğerine aktarmaya yarayan veri iletim buyrukları, kullanıcının erişimine izin verilmeyen alanlar dışında, bellek veri tablosundaki herhangi bir yeri adresleyebilir. Önceden yüklenmiş değerler otomatik olarak getirilebilir ve yeni bir yere yazılabilir. Bu yer bir zamanlayıcı ya da sayıcının önayar yazacı veya bir çıkış göstergesini denetleyen bir çıkış yazacı bile olabilir.

3.4.1 Atama Buyruğu (:=)

Atama buyruğu belirtilen adresin içeriğine bir sözcüğün veya diğer işlemlerin sonucunun yazılmasını sağlar. Bazı denetleyiciler matematik işlemler veya mantık işlemlerinin sonucunun tanımlanan bir sözcüğe yazılması için bu buyruğu kullanır.

Kullanımı:

Atama işlemi işlenen 2’yi (Operand 2:op2) işlenen 1’e (Operand 1:op1) yükler.

Atama buyruğunun sözdizimi (syntax) şöyledir:

| | | |
|----------|---|-----------|
| Op1:=Op2 | ⇔ | Op1 → Op2 |
|----------|---|-----------|

Atama işlemleri aşağıdakiler arasında yapılabilir:

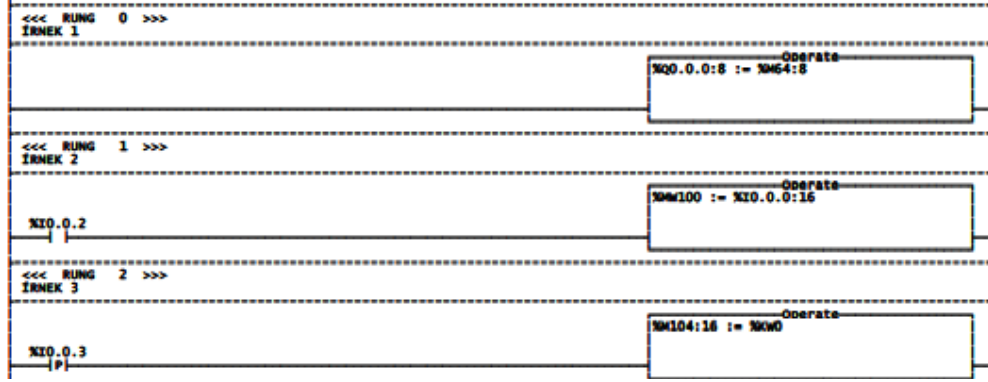
- Bit dizileri
- Sözcükler
- Çift sözcükler
- Floating (gerçek sayı) sözcük
- Sözcük tabloları
- Çift sözcük tabloları
- Floating sözcük tabloları

Bit Dizilerinin Atanması:

- Bit dizisi → bit dizisi (Örnek 1)
- Bit dizisi → sözcük (Örnek 2) veya çift sözcük (dizili:indexed)
- Sözcük veya çift sözcük (indexed) → Bit dizisi (Örnek 3)
- Mutlak değer (immediate value) → Bit dizisi

ÖRNEKLER

Ladder Diagram



List Instructions

```
-----  
(* ÖRNEK 1 *)  
0 LD 1  
1 [ %Q0.0.0:8 := %M64:8 ]  
-----  
(* ÖRNEK 2 *)  
2 LD %XIO.0.2  
3 [ %XIO.0.0:16 ]  
-----  
(* ÖRNEK 3 *)  
4 LDR %XIO.0.3  
5 [ %M104:16 := %KWO ]
```

Şekil-32 Bit dizisi atama örnekleri

Kullanım kuralları:

- Bit dizisi → sözcük atamaları için: Dizideki bitler sağdan başlayarak sözcüğe transfer edilir (dizideki ilk bit sözcükteki bit 0'a yazılır) ve transfere dahil olmayan sözcük bitleri (uzunluk ≤ 16) 0'lanır.
- Sözcük → Bit dizisi atamaları için: Sözcük bitleri sağdan başlayarak transfer edilir (sözcük biti 0 dizideki ilk bite yazılır).

Tablo-8 Bit dizisi atamaları sözdizimi

| Operatör | Sözdizimi | Operand 1 (Op1) | Operand 2 (Op2) |
|----------|---|--|--|
| := | [Op1 := Op2] Operand 1 (Op1), operand 2'nin (Op2) değerini alır | %MWi,%QWi, %SWi %MWi[MWi], %Mi:L, %Qi:L, %Si:L, %Xi:L | Anlık değer, %MWi, %KWi, %IW, %INWi, %QW, %QNWi, %SWi, %BLK.x, %MWi[MWi], %KWi[MWi], %Mi:L,%Qi:L, %Si:L, %Xi:L, %li:L |

Not: %BLK.x kısaltması herhangi bir fonksiyon bloğuna ait sözcüğü tanımlamak için kullanılmaktadır (örn. %C1.P).

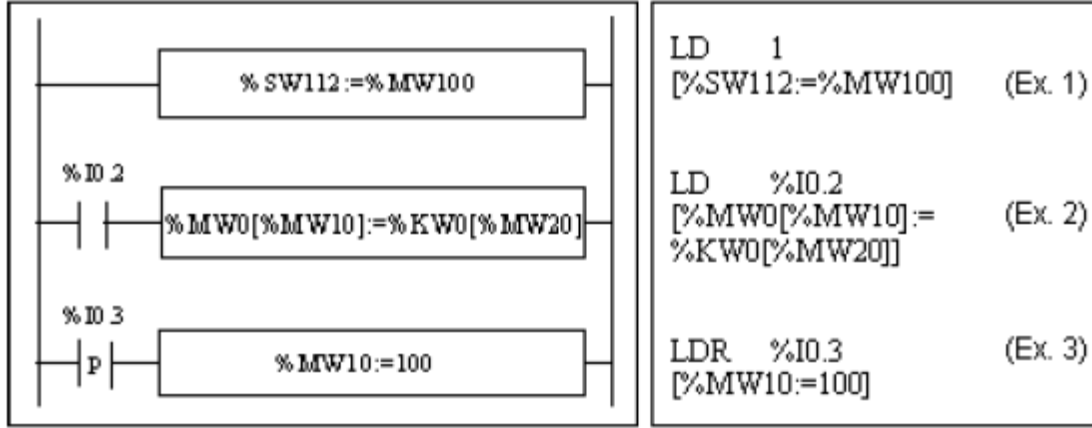
Sözcüklerin Atanması:

Sözcük veya çift sözcükler üzerinde atama işlemleri yapılabilir:

- Sözcük (dizili) → Sözcük (dizili veya değil)
- Çift sözcük (dizili) → Çift sözcük (dizili veya değil)
- Mutlak tam değer → Sözcük (Örnek 3) veya çift sözcük (dizili veya değil)
- Bit dizisi → Sözcük veya çift sözcük

- Floating point (dizili veya değil) → Floating point (dizili veya değil)
- Sözcük veya çift sözcük → Bit dizisi
- Mutlak floating point değeri → Floating point (dizili veya değil)

Şekil-33'te sözcük atama işlemlerine örnekler gösterilmektedir.



Şekil-33 Sözcük atama örnekleri

Tablo-9 Sözcük atamaları sözdizimi

| İşlemci | Sözdizimi |
|---------|--|
| := | [Op1:=Op2] İşlenen 1 (Op1) işlenen 2'nin (Op2) değerini alır. |

Tablo-10'da işlenenlerin ayrıntısı verilmiştir.

Tablo-10 Sözcük işlenenleri ve işlenenlerin ayrıntıları

| Tip | İşlenen 1 (Op1) | İşlenen 2 (Op2) |
|---------------------------------|--|---|
| Sözcük, çift sözcük, bit dizisi | %BLK.x, %MWi, %QWi, %QWai, %SWi, %Mwi[%MWi], %MDi, %MDi[%MWj], %Mi:L, %Qi:L, %Si:L, %Xi:L | Mutlak değer, %MWi, %KWi, %IW, %IWAi, %QWi, %SWi, %MWi[%MWi], %KWi[%MWi], %MDi, %MDi[%MWj], %KDi, %KDi[%MWj], %INW, %Mi:L, %Qi:L, %QNW, %Si:L, %Xi:L, %Ii:L |
| Floating point | %MFi, %MFi[[%MWj]] | Mutlak floating point değeri, %MFi, %MFi[[%MWj], %KFi, %KFi[[%MWj]] |

Not: %BLK.x kısaltması herhangi bir fonksiyon bloğuna ait sözcüğü tanımlamak için kullanılmaktadır (örn. R3.I). %Mi:L, %Si:L, %Xi:L bit dizileri için ilk bit dizisinin taban adresi 8'in katları olmalıdır (0, 8, 16,..., 96,...)

3.4.2 Karşılaştırma Buyrukları

Genelde, mantık diyagramı buyrukları kullanarak verinin işlenmesi, iki yazacın içeriklerinin karşılaştırılması şeklinde basit işlemleri içermektedir. Merdiven programlamada kullanılan buyruklar: eşit (=), büyük (>), küçük (<), büyük/eşit (≥), küçük/eşit (≤) ve eşit değil (<>). Bu karşılaştırmaların sonucu olarak bir çıkış denetlenebilir veya başka bir işlem gerçekleştirilebilir. Karşılaştırılan değerler

veya sözcükler karşılaştırma bloğu kullanılarak doğrudan girilebilir.

Bu buyruklar iki yazacın içeriğini belirtilen koşul için karşılaştırır. İşlem DOĞRU ise, çıkış bobini enerjilenir veya bir sözcük yazma işlemi yapılabilir.

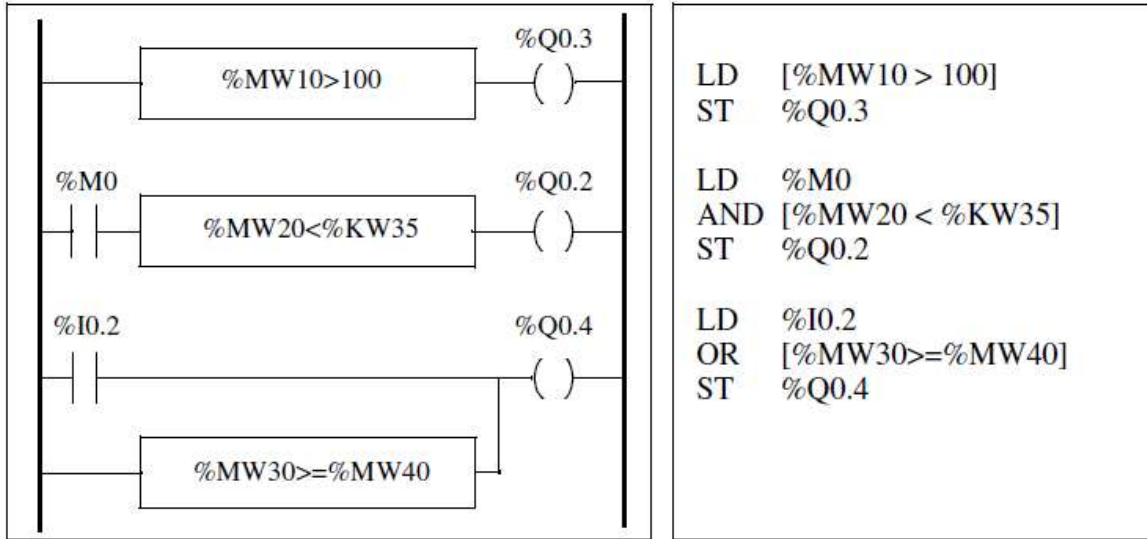
Tablo-11 Karşılaştırma buyrukları

| Buyruk | Sözdizimi |
|--------|--|
| > | İşlenen 1 (Op) işlenen 2'den (Op2) daha büyük mü diye test eder. |
| >= | İşlenen 1 (Op) işlenen 2'den (Op2) daha büyük veya eşit mi diye test eder. |
| < | İşlenen 1 (Op) işlenen 2'den (Op2) daha küçük mü diye test eder. |
| <= | İşlenen 1 (Op) işlenen 2'den (Op2) daha küçük veya eşit mi diye test eder. |
| = | İşlenen 1 (Op) işlenen 2'ye (Op2) eşit mi diye test eder. |
| <> | İşlenen 1 (Op) işlenen 2'den (Op2) farklı mı diye test eder. |

Yapı

Karşılaştırma, köşeli parantezler içinde LD, AND, ve OR buyruklarını takiben işletilir. Yapılan karşılaştırma işlemi doğru ise, sonuç "1"dir.

Şekil-34'te karşılaştırma örnekleri verilmiştir.



Şekil-34 Karşılaştırma buyrukları örneği.

Karşılaştırma buyrukları sözdizimi:

| İşlemci | Sözdizimi |
|------------------|--------------------|
| >, >=, <, <=, =, | LD[Op1 İşlem Op2] |
| <> | AND[Op1 İşlem Op2] |
| | OR[Op1 İşlem Op2] |

Tablo-12 İşlenenler

| Tip | İşlenen 1 (Op1) | İşlenen 2 (Op2) |
|----------------|--|---|
| Sözcükler | %MWi, %KWi, %INWi, , %IW, %IWAi, %QNW, %QWi, %QWAi, %SWi, %BLK.x | Mutlak değer, %MWi, %KWi, %INWi, %IW, %IWAi, %QNW, %QW, %QWAi, %SWi, %MWi[%MWi], %KWi[%MWi], %BLK.x |
| Çift sözcükler | %MDi, %KDi | Mutlak değer, %MDi, %MDi[%MWi], %KDi, %KDi[%MWi] |
| Floating point | %MFi, %KFi | Mutlak floating point değeri, %MFi, %MFi[[%MWj], %KFi, %KFi[[%MWj] |

Not: Karşılaştırma buyrukları parantezler içinde kullanılabilir.

Parantezler içinde karşılaştırma buyruklarının kullanılmasına örnek program:

```
0 LD      %M0
1 AND    ( [ %MF20 > 10.0 ]
2 OR     %I0.0.0
3 )
4 ST     %Q0.0.1
```

3.4.3 Aritmetik İşlemler

Aritmetik işlemler toplama, çıkarma, çarpma ve bölme gibi dört temel işlemin dışında bölmeden kalan, karekök, bir arttır ve bir azalt işlemlerini de içerir. Bu buyruklar bir, iki veya üç yazacın içeriklerini kullanarak gerekli işlemleri yaparlar.

Toplama (“ + ”)

“+” buyruğu iki farklı bellek alanındaki değerlerin toplanması işlemini yapar. Sonuç belirtilen bir sözcük adresinde saklanır. Toplama işlemi belirli yolların DOĞRU olması koşuluyla yapılacaksa, toplama basamağından önce giriş koşulları programlanmalıdır. Taşma olduğunda, taşma biti (%S18) tarafından belirtilir.

Çıkarma (“ - ”)

“-” buyruğu iki farklı yazacın çıkarma işlemini yapar. Toplamada olduğu gibi, çıkarma işlemine izin verecek bir koşul varsa, değerler girilmeden önce bu koşulun programlanması gerekir. Çıkarma işleminin sonucu negatif olduğunda bunu gösteren bite (%S17) sahiptir.

Çarpma (“ * ”)

“*” buyruğu çarpma işlemini yapar ve çarpma işlemine tabi tutulan yazaçların çarpım sonucunu saklamak için de bir yazaç kullanır. İşleme izin verecek bir koşul varsa, değerlerin girildiği çarpma işleminin yapıldığı basamaktan önce bu koşulun programlanması gerekir. Eğer sonuç, sonuç sözcüğünün kapasitesini aşarsa, %S18 sistem biti 1’e kurulur ve çarpma işlem sonucu anlamsızdır.

Bölme (“ / ”)

“/” buyruğu iki sayının bölünmesi işlemini gerçekleştirir. Bölmenin sonucu belirtilen bir yazaçta saklanır. Bölmeden kalanı elde etmek için **REM** (Remainder: Kalan) buyruğu kullanılır. Bölen 0 ise, bölme işlemi olanaksızdır ve %S18 biti 1’e kurulur. Sonuç yanlıştır.

Karekök alma (SQRT)

Karekök alma işlemi sadece pozitif değerlere uygulanır. Karekökün işleneni negatifse, %S18 biti 1'e kurulur ve sonuç yanlıştır.

İşleneni bir artırma (INC)

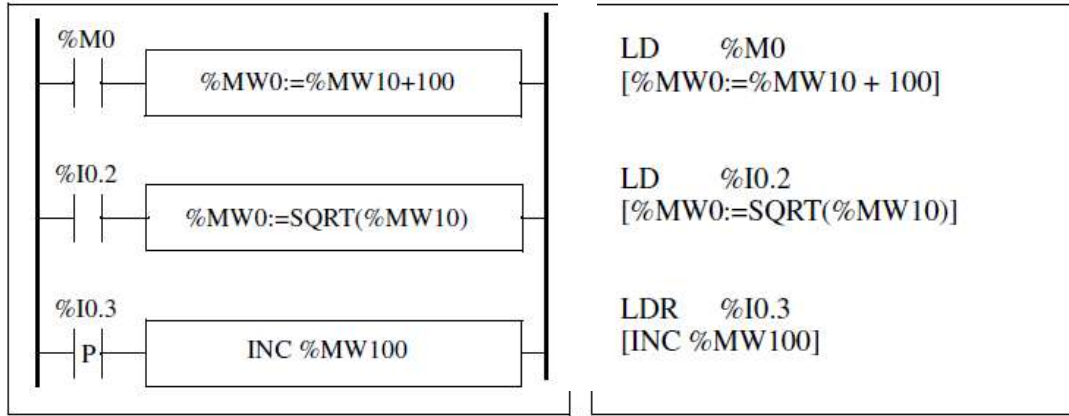
Bir işleneni gerçekleştirilen bu buyruk, programlanan koşullar DOĞRU ise işlenen sözcüğünün içeriğini bir artırır ve sonucu yine aynı sözcüğe yazar.

İşleneni bir azaltma (DEC)

Bir işleneni gerçekleştirilen bu buyruk, programlanan koşullar DOĞRU ise işlenen sözcüğünün içeriğini bir azaltır ve sonucu yine aynı sözcüğe yazar.

Yapı

Aritmetik işlemler Şekil-35'teki gibi gerçekleştirilebilir.



Şekil-35 Aritmetik buyruklar örneği.

Not: Sistem bitleri %S17 ve %S18'in idaresinden kullanıcı programı sorumludur. Bu bitler denetleyici tarafından "1"e kurulur ve yeniden kullanılabilmesi için program tarafından

Sözdizimi Tablo-13'te gösterildiği gibi kullanılan işleme göre değişir.

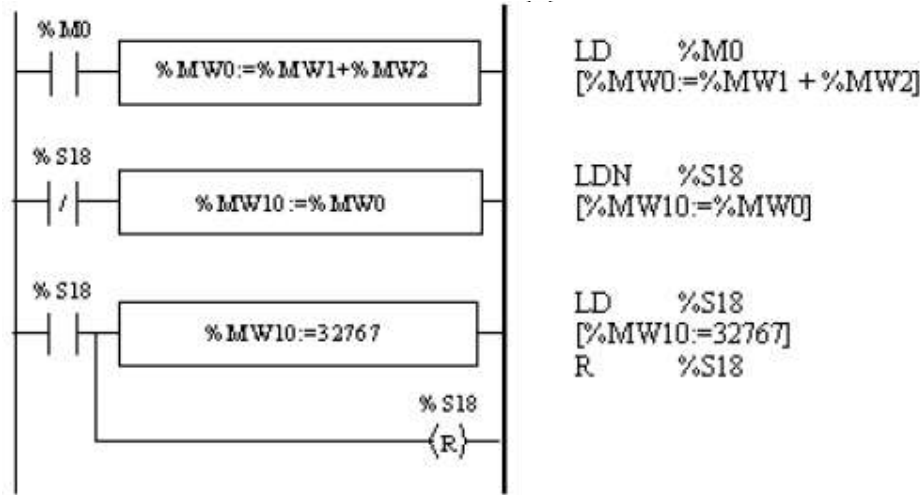
Tablo-13 Sözdizimi

| İşlemci | Sözdizimi |
|-----------------|------------------------|
| +, -, *, /, REM | [Op1 := Op2 İşlem Op3] |
| INC, DEC | [İşlem Op1] |
| SQRT (1) | [Op1:= SQRT (Op2)] |

Tablo-14 İşlenenler

| Tip | İşlenen 1 (Op1) | İşlenen 2 (Op2) |
|----------------|-------------------------|--|
| Sözcükler | %MWi, %QWi, %QWai, %SWi | Mutlak değer, %MWi, %KWi, %INW, %IW, %IWAi, %QNW, %QW, %QWai, %SWi, %BLK.x |
| Çift sözcükler | %MDi | Mutlak değer, %MDi, %KDi |

Örnek: Toplamada taşma



Şekil-36 Taşma bitinin sıfırlanması.

Eğer %MW1 = 23241 ve %MW2 = 21853, gerçek sonuç (45094) 16 bitlik tek bir sözcükte gösterilemez, %S18 biti “1”e kurular ve elde edilen sonuç (-20442) yanlıştır. Bu örnekte sonuç 32767’den büyüktür ve değeri 32767’de sabitlenir.

3.4.4 Mantık Komutları

Bu komutlar ile iki sözcük işleneni arasında veya tek bir sözcük işleneni üzerinde mantık işlemleri gerçekleştirilir.

AND: İki işlenenin bitleri arasındaki VE işlemi

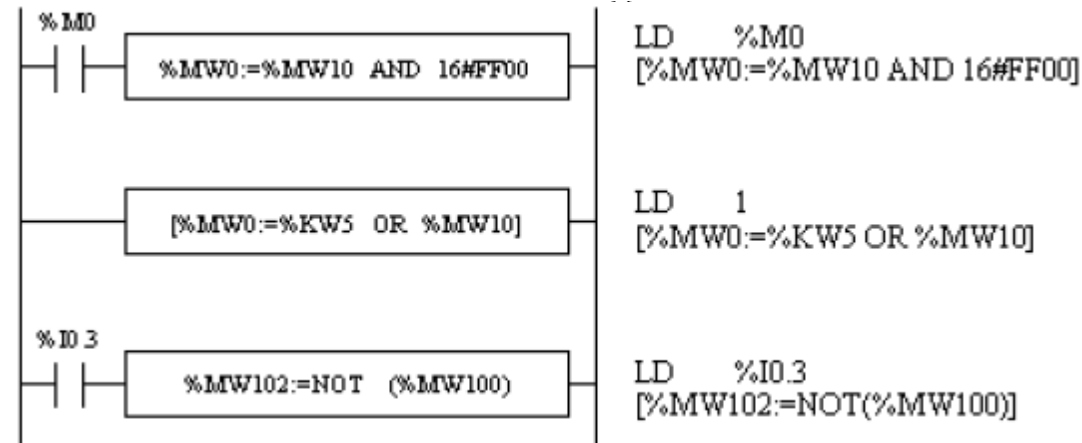
OR: İki işlenenin bitleri arasındaki VEYA işlemi

XOR: İki işlenenin bitleri arasındaki Özel VEYA işlemi

NOT: Bir işlenenin değili (bit olarak)

Yapı:

Mantık işlemleri aşağıdaki gibi gerçekleştirilir:



Şekil-37 Mantık buyrukları örneği.

Tablo-15 Mantık buyrukları sözdizimi ve işlenenleri

| İşlemci | Sözdizimi | İşlenen 1 (Op1) | İşlenen 2 ve 3 (Op2 & 3) |
|--------------|------------------------|-------------------------|--|
| AND, OR, XOR | [Op1 := Op2 işlem Op3] | %MWi, %QWi, %QWai, %SWi | Mutlak değer (1), %MWi, %KWi, %IW, %IWAi, %QW, %QWai, %SWi, %BLK.x |
| NOT | [NOT(Op2)] | | |

Not: (1) NOT işleminde Op2 mutlak değer olamaz.

Örnek:

Aşağıdaki örnek bir mantık AND buyruğu kullanımını göstermektedir:

[%MW15:=%MW32 AND %MW12]

3.4.5 Kaydırma (Shift) Buyrukları

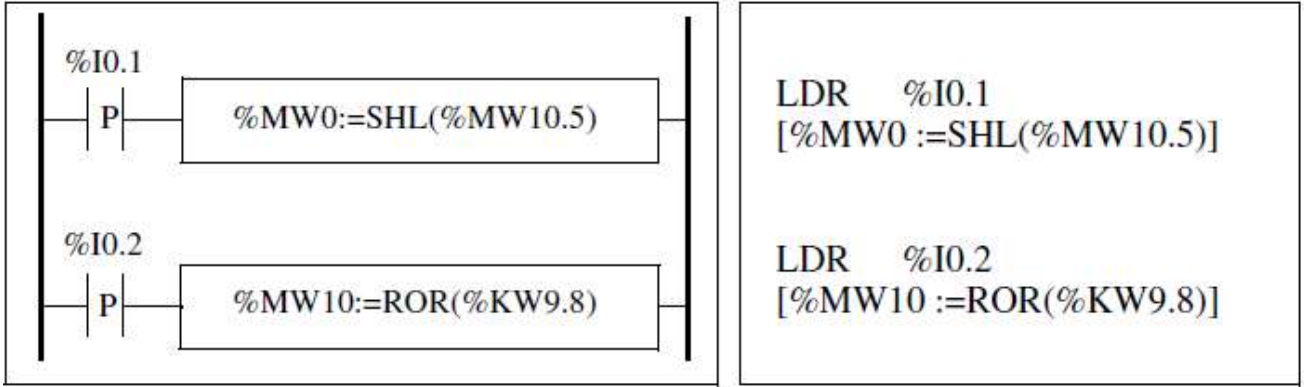
Açıklama: Kaydırma buyrukları bir işlenenin bitlerinin belirli bir sayıda sağa ya da sola kaydırma işlemi için kullanılır. Tablo-16’da kaydırma buyrukları sözdizimi ve kullanımını gösterilmiştir.

Tablo-16 Kaydırma buyrukları sözdizimi ve açıklamaları

| Buyruk | İşlev | |
|-----------------|--|--|
| Mantık kaydırma | | |
| SHL(op2,i) | Sola doğru i kadar konum kaydırma mantık işlemi. | |
| SHR(op2,i) | Sağa doğru i kadar konum kaydırma mantık işlemi. | |
| Döndürme | | |
| ROL(op2,i) | Sola doğru i kadar konum döndürme mantık işlemi. | |
| ROR(op2,i) | Sağa doğru i kadar konum döndürme mantık işlemi. | |

Yapı:

Kaydırma işlemleri aşağıdaki gibi yapılır:



Şekil-38 Kaydırma buyrukları program örnekleri.

Sözdizimi

Tablo-17’de görüldüğü gibi, sözdizimi kullanılan işleme göre değişir.

Tablo-17 Kaydırma buyrukları sözdizimi

| İşlemci | Sözdizimi |
|----------|--------------------------|
| SHL, SHR | [Op1 := İşlemci (Op2,i)] |
| ROL, ROR | |

Tablo-18 İşlenenler

| Tip | İşlenen 1 (Op1) | İşlenen 2 (Op2) |
|----------------|-------------------------|---|
| Sözcükler | %MWi, %QWi, %QWai, %SWi | %MWi, %KWi, %IW, %IWAi, %QW, %QWai, %SWi, %BLK.x |
| Çift sözcükler | %MDi | %MDi, %KDi |

3.4.6 Dönüşüm Buyrukları

Açıklama: Dönüşüm buyrukları farklı sayı gösterimleri arasındaki dönüşümü yapmak için kullanılır. Tablo-19’da Dönüşüm buyruklarının çeşitleri görülmektedir.

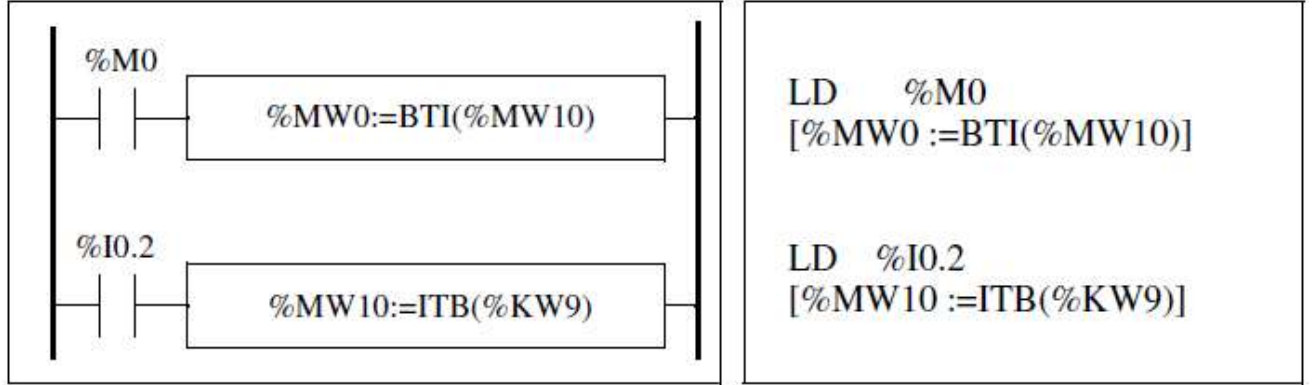
Tablo-19 Dönüşüm buyrukları

| Buyruk | İşlev |
|--------|-----------------------|
| BTI | BCD → İkilik dönüşümü |
| ITB | İkilik → BCD dönüşümü |

Not: Sistem biti %S17 kapasite aşımı için kullanılır.

Yapı:

Dönüşüm işlemleri aşağıdaki gibi yapılır:



Şekil-39 Dönüşüm buyrukları program örnekleri.

Sözdizimi

Sözdizimi Tablo-20’de görüldüğü gibidir.

Tablo-20 Dönüşüm buyrukları sözdizimi

| İşlemci | Sözdizimi |
|----------|------------------------|
| BTI, ITB | [Op1 := İşlemci (Op2)] |

Tablo-21 İşlenenler

| Tip | İşlenen 1 (Op1) | İşlenen 2 (Op2) |
|----------------|-------------------------|---|
| Sözcükler | %MWi, %QWi, %QWai, %SWi | %MWi, %KWi, %IW, %IWAi, %QW, %QWai, %SWi, %BLK.x |
| Çift sözcükler | %MDi | %MDi, %KDi |

Uygulama Örneği:

- BTI buyruğu BCD kodlanmış iki konumlu anahtarlar aracılığıyla denetleyicinin girişlerine uygulanan bir set noktası değerini işlemek için kullanılır.
- ITB buyruğu BCD kodlanmış göstergelerde nümerik değerleri (örneğin, hesaplama sonucu, bir fonksiyon bloğunun mevcut değeri) göstermek için kullanılır.

3.4.7 Tek/Çift Sözcük Dönüşüm Buyrukları

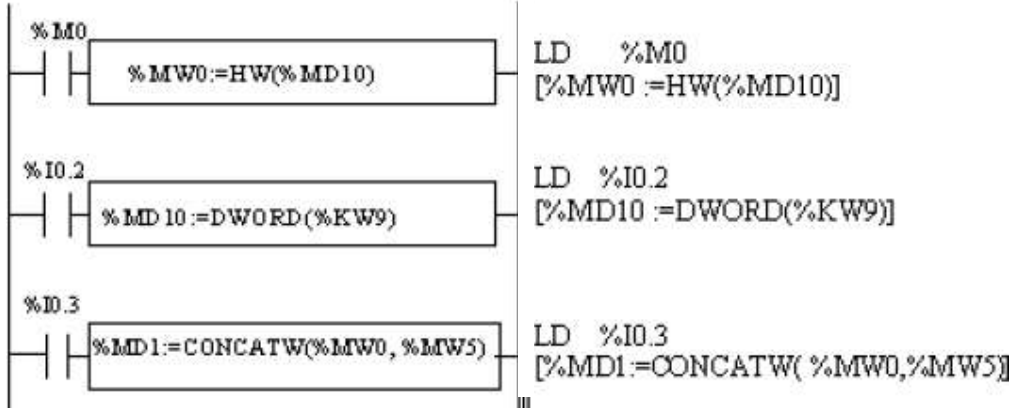
Açıklama: Tek/Çift sözcükler arası dönüşüm buyrukları Tablo-22’de verilmiştir.

Tablo-22 Sözcükler arası dönüşüm buyrukları sözdizimi

| Buyruk | İşlev |
|---------|--|
| LW | Çift sözcüğün LSB’sini sözcüğe çıkarır. |
| HW | Çift sözcüğün MSB’sini sözcüğe çıkarır. |
| CONCATW | İki sözcüğü bir çift sözcükte birbirine bağlar. |
| DWORD | 16 bitlik sözcüğü 32 bitlik bir çift sözcüğe dönüştürür. |

Yapı:

Dönüşüm işlemleri aşağıdaki gibi yapılır:



Şekil-40 Sözcük Dönüşüm buyrukları program örnekleri.

Sözdizimi

Sözdizimi Tablo-23'te görüldüğü gibidir.

Tablo-23 Dönüşüm buyrukları sözdizimi

| İşlemci | Sözdizimi | İşlenen 1 (Op1) | İşlenen 2 (Op2) | İşlenen 3 (Op3) |
|---------|---------------------------|-----------------|--------------------------|--------------------------|
| LW, HW | [Op1 := İşlem (Op2)] | %MWi | %MDi, %KDi | (-) |
| CONCATW | [Op1 := İşlem (Op2, Op3)] | %MDi | %MWi, %KWi, mutlak değer | %MWi, %KWi, mutlak değer |
| DWORD | [Op1 := İşlem (Op2)] | %MDi | %MWi, %KWi | (-) |

3.5 Program Buyrukları

Bu bölümde incelenecek olan buyruklar şunlardır:

- END Buyrukları
- NOP Buyrukları
- Dallanma (Jump) Buyrukları
- Altyordam (Subroutine) Buyrukları

3.5.1 END Buyrukları

Açıklama: END buyrukları bir program tarama çevriminin işletilmesinin sonunu tanımlar.

Üç farklı “end” buyruğu vardır:

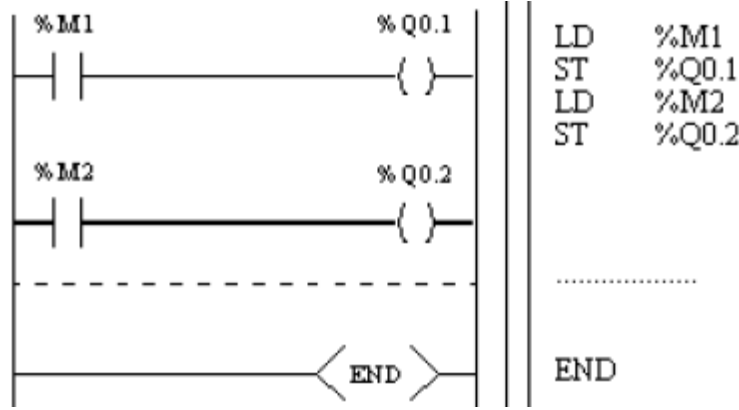
- END: Koşulsuz program sonu.
- ENDC: Önceki Boole işlem sonucunun “1” olması durumunda program sonu
- ENDCN: Önceki Boole işlem sonucunun “0” olması durumunda program sonu

Önceden tanımlı olarak (normal mod), program sonu etkinleştirildiğinde çıkışlar güncellenir ve yeni tarama çevrimi başlatılır.

Tarama periyodikse, program sonuna ulaşıldığında çıkışlar güncellenir ve yeni tarama başlatılır.

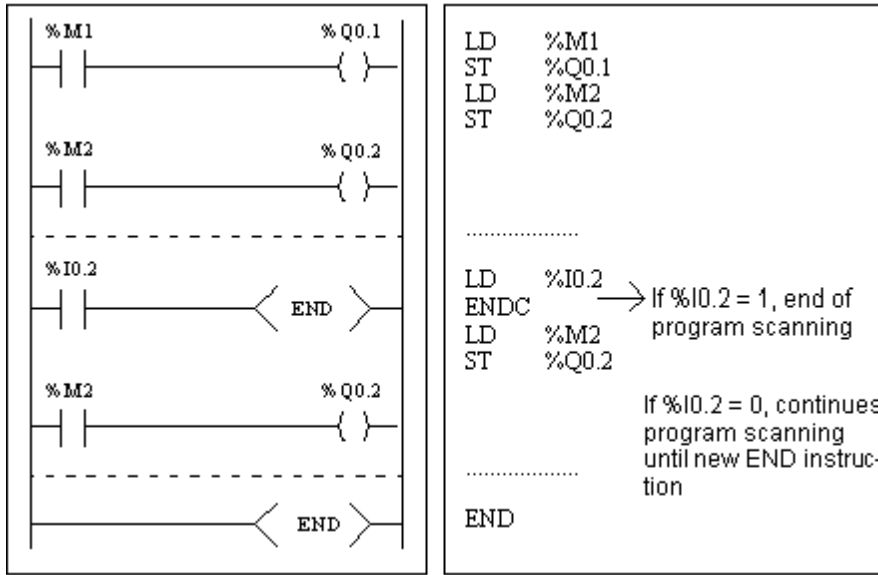
Örnekler

Şekil-41’de koşulsuz END buyruğunun kullanımı görülmektedir.



Şekil-41 Koşulsuz END buyruğunun kullanımı.

Şekil-42’de koşulsuz END buyruğunun kullanımı görülmektedir.



Şekil-42 Koşullu END buyruğunun kullanımı.

3.5.2 NOP Buyruğu

NOP (No Operation) buyruğu hiçbir işlem yapmaz. Daha sonra satır numaralarını değiştirmeden programa yeni buyruklar eklenmek istendiğinde satır “ayırarak” için kullanılır.

3.5.3 Dallanma (Jump) Buyrukları

Açıklama: Jump buyrukları programın işletilmesinin hemen kesilmesi ve bu program satırından sonra %Li (compact PLC’ler için i = 1 ile 16 arası ve diğerleri için 1 ile 63 arası) etiketli satırdan devam edilmesini sağlar.

JMP, JMPC ve JMPCN

Üç farklı “Jump” buyruğu bulunmaktadır:

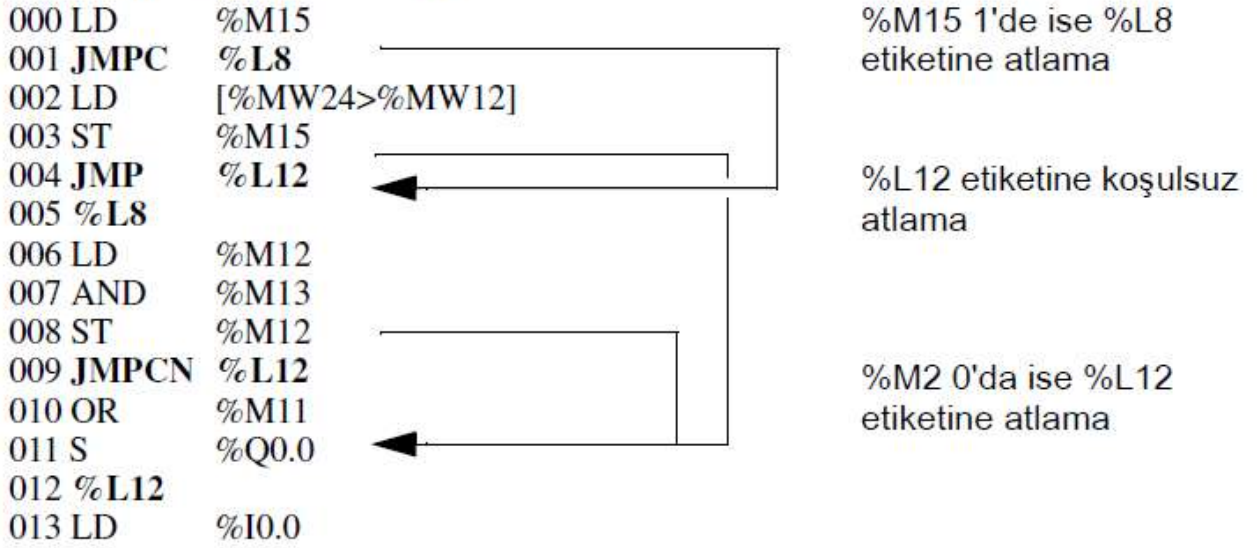
JMP: Koşulsuz program dallanma (jump)

JMPC: Önceki Boole işlem sonucunun “1” olması durumunda program dallanma

JMPCN: Önceki Boole işlem sonucunun “0” olması durumunda program dallanma

Örnekler

Jump Buyrukları Örnekleri



Şekil-43 Dallanma buyruklarının kullanımı.

Kurallar

- Dallanma buyrukları parantezler içinde kullanılamaz ve AND(, OR(buyrukları ve bir parantez kapama buyruğunun ")" arasına yerleştirilmemelidir.
- Etiket bir LD, LDN, LDR, LDF veya BLK buyruğundan önce yerleştirilmelidir.
- %Li etiketinin etiket numarası programda yalnızca bir defa tanımlanmalıdır.
- Program dallanma yukarı veya aşağı doğru olan programlama satırına gerçekleştirilir. Dallanma yukarı doğru olduğunda, programın tarama süresine dikkat edilmelidir. Uzatılmış tarama süresi güvenlik zamanlayıcısının (watchdog) tetiklenmesine neden olabilir.

3.5.4 Altyordam (Subroutine) Buyrukları

Açıklama: Altyordam buyrukları bir programın bir altyordamı yürütmesi ve sonra ana programa geri dönmesini sağlar.

SRn, SRn: and RET.

Altyordamlar üç adımdan oluşur:

- Önceki Boole işlem sonucu "1" ise, SRn buyruğu SRn etiketi ile gönderme yapılan altyordamı çağırır.
- Altyordama SRn: etiketi ile gönderme yapılır. TWDLCAA10DRF, TWDLCAA16DRF için n = 0 ile 15 arası ve tüm diğer denetleyiciler için 0 ile 63 arası.
- Altyordamın sonuna yerleştirilen RET buyruğu program akışını ana programa döndürür.

Örnekler

Altyordam buyrukları örnekleri.

000 LD %M15

001 AND %M5

002 ST %Q0.0

003 LD [%MW24>%MW12]

004 SR8

005 LD %I0.4

006 AND M13

007 -

008 -

009 -

010 END

011 SR8:

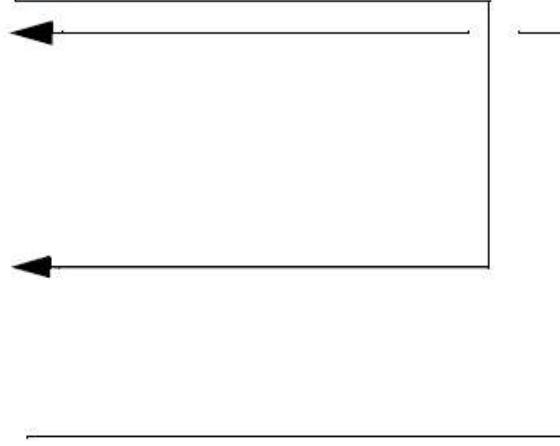
012 LD 1

013 IN %TM0

014 LD [%TM0.Q]

015 ST %M15

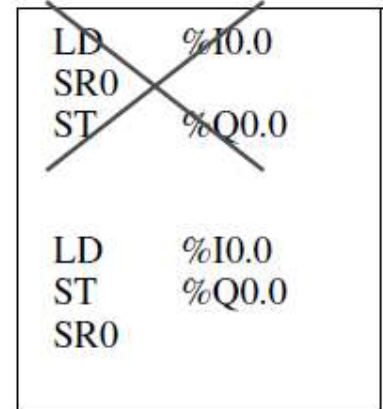
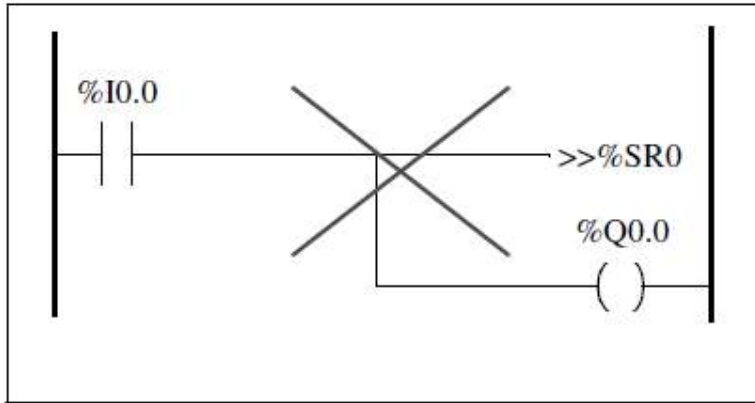
016 RET



Şekil-44 Altyordam buyruklarının kullanımı.

Kurallar

- Bir altyordam başka bir altyordamı çağırmamalıdır.
- Altyordam buyrukları parantezler içinde kullanılamaz ve AND(, OR(buyrukları ve bir parantez kapama buyruğunun ")" arasına yerleştirilmemelidir.
- Etiket, sadece bir LD veya bir Bool denkleminin (veya rung) başlangıcını işaret eden bir BLK komutundan önce yerleştirilebilir.
- Altyordamın çağırılmasının ardından bir atama buyruğu kullanılmamalıdır. Bunun nedeni altyordamın Boole akümülatörünün içeriğini değiştirebilmesidir. Dolayısıyla altyordamdan dönüşte, çağırılmadan öncekinden daha farklı bir değer alabilir. Aşağıdaki örnekte bunu inceleyebiliriz.



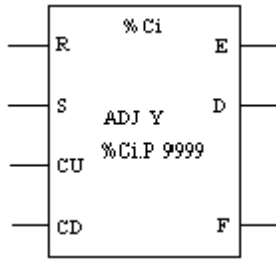
Şekil-45 Altyordam buyruklarının yanlış kullanımına örnekler.

3.6 Temel Fonksiyon Blokları

Açıklama: Fonksiyon blokları programlar tarafından kullanılan bit objeleri ve özel sözcük kaynaklarıdır. Temel fonksiyon blokları zamanlayıcılar, yukarı/aşağı sayma gibi basit işlevleri sağlarlar.

3.6.1 Fonksiyon Blokları Bit ve Sözcük Objeleri

Aşağıdaki gösterim bir yukarı/aşağı Sayıcı (Counter) Fonksiyon Bloğu örneğidir.



Şekil-46 Yukarı/aşağı Sayıcı Fonksiyon Bloğu.

Bit Objeleri

Bit objeleri blok çıkışlarına karşılık gelir. Bu bitler aşağıdaki yöntemlerden biri kullanılarak Boolean test buyrukları ile erişilebilir:

Girişler buyruklar formunda erişilebilir.

- Eğer dönüştürülebilir programlamada bloğa bağlandıysa doğrudan (örneğin, LD E) (Standart fonksiyon blokları programlama ilkelerine bkz.).
- Blok tipini tanımlayarak (örneğin, LD %Ci.E).

Sözcük Objeleri

Sözcük objeleri aşağıdaki şekilde tanımlı parametrelere ve değerlere karşılık gelir:

- Blok konfigürasyon parametreleri: Bazı parametreler program tarafından erişilebilir (örneğin, önceden seçilen parametreler) ve bazıları program tarafından erişilemez (örneğin, zaman tabanı).
- Mevcut değerler: Örneğin, %Ci.V, mevcut sayma değeri.

Erişilebilir Bit ve Sözcük Objeleri

Tablo-24'te program tarafından erişilebilir temel fonksiyon blokları bit ve sözcük objeleri tanımlanmıştır.

Tablo-24 Fonksiyon blokları bit ve sözcük objeleri

| Temel Fonksiyon Bloğu | Sembol | Aralık (i) | Obje Çeşitleri | Tanım | Adres | Yazma Erişimi |
|-----------------------|--------|------------|----------------|---------------------------------|--------|---------------|
| Zamanlayıcı | %Tmi | 0-127 | Sözcük | Mevcut Değer | %Tmi.V | yok |
| | | | | Önayar değeri | %Tmi.P | var |
| | | | Bit | Zamanlayıcı çıkışı | %Tmi.Q | yok |
| Yukarı/aşağı Sayıcı | %Ci | 0-127 | Sözcük | Mevcut Değer | %Ci.V | yok |
| | | | | Önayar değeri | %Ci.P | var |
| | | | Bit | Alttan taşma çıkışı (empty:boş) | %Ci.E | yok |
| | | | | Önayar değerine ulaşıldı (done) | %Ci.D | yok |
| | | | | Üstten taşma çıkışı (full:dolu) | %Ci.F | yok |

3.6.1.1 Standart Fonksiyon Blokları Programlama İlkeleri

Açıklama

Standart fonksiyon bloklarını programlamak için aşağıdaki yöntemlerin birini kullanın:

- Fonksiyon blok buyrukları (örneğin, BLK %TM2): Bu dönüştürülebilir merdiven diyagramı programlama yöntemi, işlemlerin blok üzerinde programın tek bir yerinde (merdiven basamağında) yapılmasına olanak sağlar.
- Özel buyruklar (örneğin, CU %Ci): Bu dönüştürülemez yöntem işlemlerin blokların girişlerinde programın farklı yerlerinde yapılmasına olanak sağlar (örneğin, 100.satır CU %C1, 174. satır CD %C1, 209. satır LD %C1.D).

Dönüştürülebilir Programlama

Dönüştürülebilir programlama için BLK, OUT_BLK ve END_BLK buyruklarını kullanın:

- BLK: Blok başlangıcını belirtir.
- OUT_BLK: Blok çıkışlarının doğrudan bağlanmasında kullanılır.
- END_BLK: Blok sonunu belirtir.

3.6.2 Zamanlayıcı (Timer) Fonksiyon Bloğu (%TMi)

Açıklama

Üç çeşit zamanlayıcı fonksiyon bloğu vardır:

Zaman gecikmeleri veya vuru periyotları programlanabilir ve belki TwidoSoft yazılımı kullanılarak değiştirilebilir.

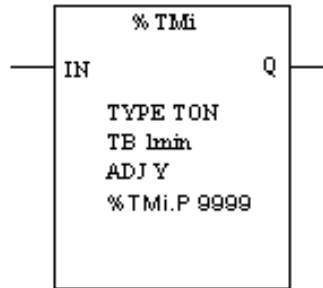
TON (Timer ON-Delay): Bu tip zamanlayıcı zaman gecikmeli enerjilendirme işlemlerinin yapılması için kullanılır.

TOF (Timer OFF-Delay): Bu tip zamanlayıcı zaman gecikmeli enerji kesme işlemlerinin yapılması için kullanılır.

TP (Timer - Pulse): Bu tip zamanlayıcı belirli bir zaman aralığında vuru üretmek için kullanılır.

Gösterim

Aşağıdaki Şekil-47 Zamanlayıcı Fonksiyon Bloğu gösterimidir.



Şekil-47 Zamanlayıcı Fonksiyon Bloğu.

Parametreler

Zamanlayıcı Fonksiyon Bloğu aşağıdaki parametrelere sahiptir:

Tablo-25 Zamanlayıcı Fonksiyon Bloğu parametreleri

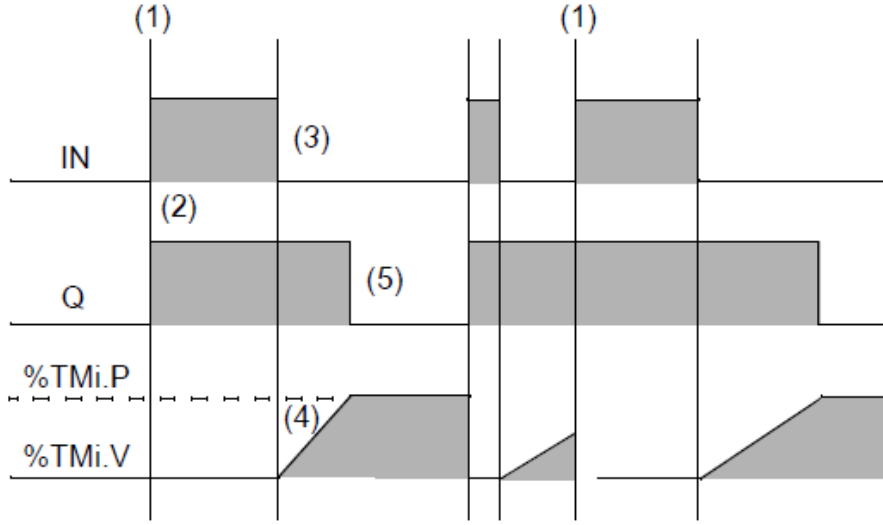
| Parametre | Etiket | Değer |
|----------------------------|--------------------|--|
| Zamanlayıcı no.su | %T _{Mi} | 0 - 63: TWDLCAA10DRF ve TWDLCAA16DRF 0 - 127 diğer denetleyiciler için. |
| Tip | TON | • Timer on-delay |
| | TOF | • Timer off-delay |
| | TP | • Vuru (monostable) |
| Zaman Tabanı (Time Base) | TB | 1 dak (önceden tanımlı), 1 s, 100 ms, 10 ms, 1 ms |
| Mevcut Değer | %T _{Mi.V} | Zamanlayıcı çalıştığında 0'dan %T _{Mi.P} 'ye kadar artan sözcük. Okunabilir ve test edilebilir, fakat program tarafından yazılamaz. %T _{Mi.V} Animasyon Tablosu Editörü kullanılarak değiştirilebilir. |
| Önayar Değeri | %T _{Mi.P} | 0-9999. Okunabilir, test edilebilir ve program tarafından yazılabilir sözcük. Önceden tanımlı değer 9999. Üretilen zaman gecikmesi %T _{Mi.P} ×TB'dir. |
| Animasyon Tablosu Editörü | Y/N | Y: Yes (Evet), %T _{Mi.P} önayar değeri Animasyon Tablosu Editörü kullanılarak değiştirilebilir. N:No(Hayır),%T _{Mi.P} önayar değeri değiştirilemez |
| İzin girişi (veya buyruğu) | IN | TON veya TP tipi zamanlayıcıları yükselen kenarda, TOF tipi zamanlayıcıyı düşen kenarda başlatır. |
| Zamanlayıcı çıkışı | Q | Atanmış %T _{Mi.Q} biti gerçekleştirilen fonksiyona (TON, TOF veya TP) göre "1"e kurulur. |

3.6.2.1 TOF Tipi Zamanlayıcı

Açıklama: TOF (Timer Off-Delay) tipi zamanlayıcı zaman gecikmeli enerjisi kesilecek olayları kontrol etmek için kullanılır. Bu gecikme TwidoSoft kullanılarak programlanabilir.

Zamanlama Diyagramı

Aşağıdaki zamanlama diyagramı TOF tipi zamanlayıcının çalışmasını göstermektedir.



Şekil-48 TOF tipi zamanlayıcının zamanlama diyagramı.

Çalışma:

Tablo-26'da TOF tipi zamanlayıcının çalışması açıklanmıştır.

Tablo-26 TOF tipi zamanlayıcının çalışması

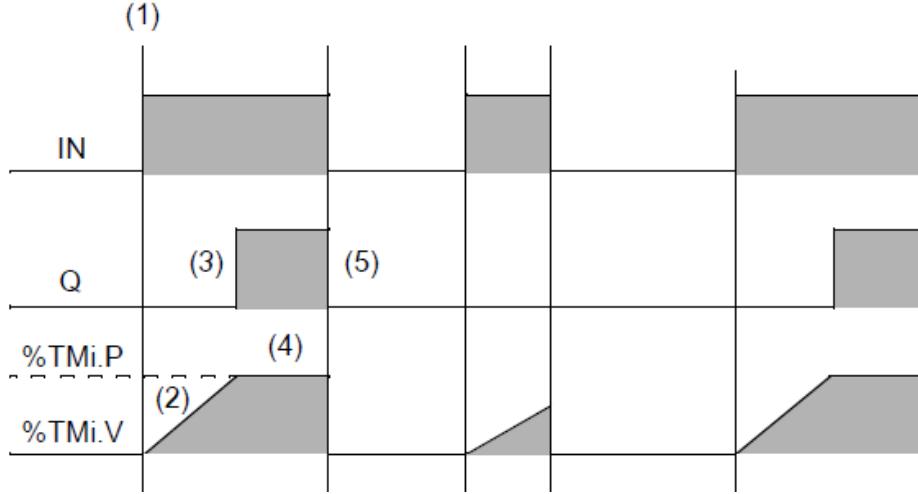
| Faz | Tanım |
|-----|--|
| 1 | %T.Mi.V mevcut değeri, zamanlayıcı çalışıyor bile olsa, IN girişindeki yükselen kenarda "0" a kurulur. |
| 2 | %T.Mi.Q çıkış biti IN girişinde yükselen kenar algılandığında "1" e kurulur. |
| 3 | Zamanlayıcı IN girişinin düşen kenarında çalışmaya başlar. |
| 4 | %T.Mi.V mevcut değeri %T.Mi.P'ye kadar zaman tabanının (TB) her vurusunda bir artmaya başlar. |
| 5 | %T.Mi.Q çıkış biti, mevcut değer %T.Mi.P'ye ulaştığında "0" lanır. |

3.6.2.2 TON Tipi Zamanlayıcı

Açıklama: TON (Timer On-Delay) tipi zamanlayıcı zaman gecikmeli enerjilenecek olayları kontrol etmek için kullanılır. Bu gecikme TwidoSoft kullanılarak programlanabilir.

Zamanlama Diyagramı

Aşağıdaki zamanlama diyagramı TON tipi zamanlayıcının çalışmasını göstermektedir.



Şekil-49 TON tipi zamanlayıcının zamanlama diyagramı.

Çalışma:

Tablo-27’de TON tipi zamanlayıcının çalışması açıklanmıştır.

Tablo-27 TON tipi zamanlayıcının çalışması

| Faz | Tanım |
|-----|--|
| 1 | Zamanlayıcı IN girişinin yükselen kenarında çalışmaya başlar. |
| 2 | %T.Mi.V mevcut değeri %T.Mi.P’ye kadar zaman tabanının (TB) her vurusunda bir artmaya başlar. |
| 3 | %T.Mi.Q çıkış biti, mevcut değer %T.Mi.P’ye ulaştığında “1”e kurulur. |
| 4 | IN girişi “1” olduğu sürece %T.Mi.Q çıkış biti “1” değerinde kalır. |
| 5 | IN girişinde bir düşen kenar algılandığında, mevcut değer %T.Mi.P’ye ulaşmamış bile olsa, zamanlayıcı durur ve %T.Mi.V “0”lanır. |

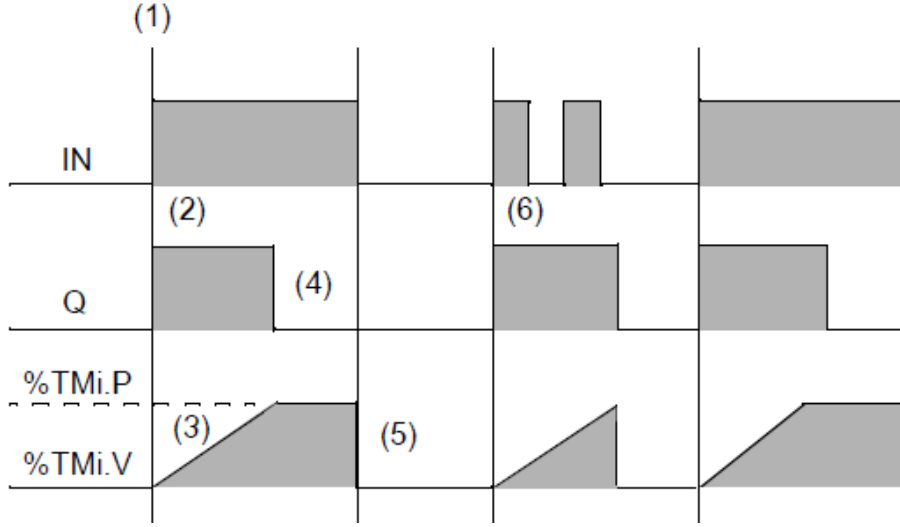
Not: Önayar Değeri ne kadar büyük olursa zaman gecikmesi doğruluğu o kadar yüksek olur.

3.6.2.3 TP Tipi Zamanlayıcı

Açıklama: TP (Timer-Pulse) tipi zamanlayıcı belirli bir zaman aralığında vuru üretmek için kullanılır. Bu gecikme TwidoSoft kullanılarak programlanabilir.

Zamanlama Diyagramı

Aşağıdaki zamanlama diyagramı TP tipi zamanlayıcının çalışmasını göstermektedir.



Şekil-50 TP tipi zamanlayıcının zamanlama diyagramı.

Çalışma:

Tablo-28’de TP tipi zamanlayıcının çalışması açıklanmıştır.

Tablo-28 TP tipi zamanlayıcının çalışması

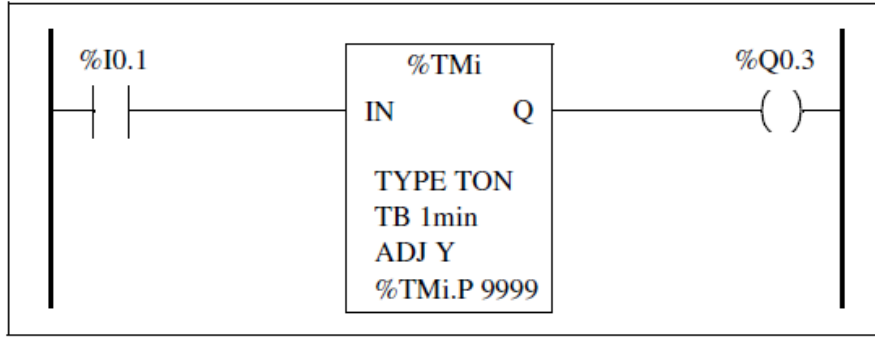
| Faz | Tanım |
|-----|--|
| 1 | Zamanlayıcı IN girişinin yükselen kenarında çalışmaya başlar. Zamanlayıcı çalışmaya başlamamış ise mevcut değer %TMI.V “0”a kurur. |
| 2 | Zamanlayıcı çalışmaya başladığında %TMI.Q çıkış biti “1”e kurur. |
| 3 | %TMI.V mevcut değeri %TMI.P’ye kadar zaman tabanının (TB) her vurusunda bir artmaya başlar. |
| 4 | %TMI.Q çıkış biti, mevcut değer %TMI.P’ye ulaştığında “0”lanır. |
| 5 | Mevcut değer %TMI.V %TMI.P’ye eşit olduğunda ve IN girişi 0’a döndüğünde, %TMI.V “0”lanır. |
| 6 | Bu zamanlayıcı sıfırlanamaz. Ancak mevcut değer %TMI.V %TMI.P’ye eşit ve IN girişi 0 olduğunda %TMI.V “0”lanır. |

Zamanlayıcıların Programlanması ve Konfigurasyonu

Açıklama: Zamanlayıcı fonksiyon blokları (%TMI) nasıl kullanıldıklarına bakılmaksızın aynı şekilde programlanırlar. Zamanlayıcı fonksiyonu (TON, TOF veya TP) konfigürasyon sırasında seçilir.

Örnekler

Aşağıdaki gösterimde zamanlayıcı fonksiyon bloğunun dönüştürülebilir ve dönüştürülemez programlama örnekleri bulunmaktadır.



Dönüştürülebilir programlama

Dönüştürülemez programlama

```

BLK   %TMI
LD    %I0.1
IN
OUT_BLK
LD    Q
ST    %Q0.3
END_BLK

```

```

LD    %I0.1
IN    %TMI
LD    %TMI.Q
ST    %Q0.3

```

Şekil-51 Zamanlayıcı programlama örneği.

Yapılandırma

Yapılandırma sırasında aşağıdaki parametreler girilmelidir:

- Zamanlayıcı tipi: TON, TOF veya TP
- Zaman tabanı: 1 dak, 1 s, 100 ms, 10 ms veya 1 ms
- Önayar değeri (%TMI.P): 0 ile 9999 arası
- Ayarla: İşaretli veya İşaretsiz

Özel Durumlar

Tablo-29’da zamanlayıcı fonksiyon bloğunun programlanması için özel durumlar listelenmiştir.

Tablo-29 Zamanlayıcı fonksiyon bloğunun programlanmasında özel durumlar.

| Özel Durum | Tanım |
|--|---|
| Soğuk yeniden başlatmanın etkisi (%S0=1) | Mevcut değeri 0 olmaya zorlar. %TMI.Q’yu 0’lar. Önayar değeri konfigürasyon sırasında tanımlanan değere eşitlenir. |
| Sıcak yeniden başlatmanın etkisi (%S1=1) | Zamanlayıcının mevcut değeri ve önayar değeri üzerinde etkisi yoktur. Güç kesildiğinde mevcut değer değişmez. |
| Denetleyicinin durmasının etkisi | Denetleyicinin durdurulması mevcut değeri dondurmaz. |
| Program dallanmasının etkisi | Zamanlayıcı bloğunun üzerinden dallanma zamanlayıcıyı dondurmaz. Zamanlayıcı önayar değerine (%TMI.P) ulaşana kadar artmaya devam eder. Bu noktada, zamanlayıcı bloğunun Q çıkışına atanan “işlem tamam” biti (%TMI.Q) durum değiştirir. Ancak, blok çıkışına doğrudan bağlanan ilgili çıkış etkinleştirilmez ve denetleyici tarafından taranmaz. |
| %TMI.Q biti ile test etme | %TMI.Q bitinin program içinde bir defa test edilmesi önerilir. |

| | |
|--|--|
| MCS/MCR master kontrol rölesi buyruklarının etkisi | İki master kontrol rölesi buyruğunun arasında programlanmış zamanlayıcı bloğu MCS buyruğu etkin olduğunda sıfırlanır. |
| %TMi.P'nin değiştirilmesinin etkisi | Bir buyruk kullanarak veya değeri değiştirerek önayar değerinin değiştirilmesi yalnızca zamanlayıcının bir sonraki devreye girişinde etkin olur. |

3.6.2.4 1 ms Zaman Tabanı Kullanan Zamanlayıcılar

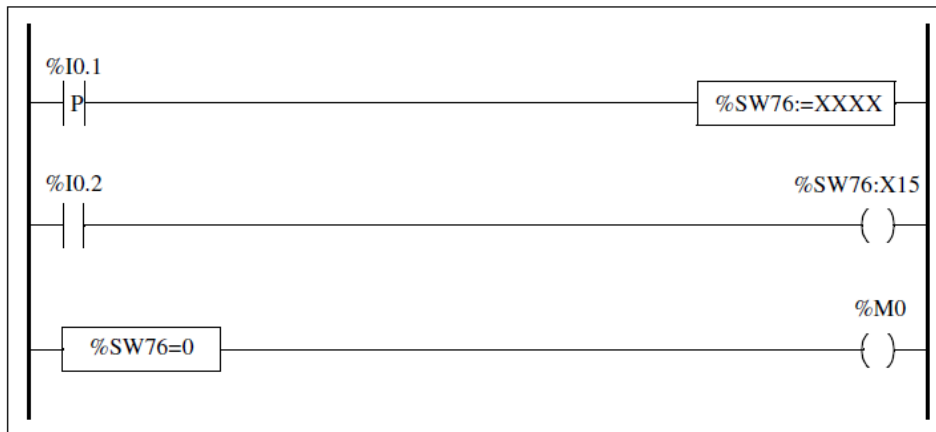
1 ms zaman tabanı ilk beş zamanlayıcı ile kullanılabilir. Dört sistem sözcüğü %SW76, %SW77, %SW78 ve %SW79, "kum saatleri" olarak kullanılabilir. Bu dört sözcüğün her biri pozitif değere sahiplerse sistem tarafından her milisaniyede bir azaltılırlar.

Bu sözcüklerin birini ard arda yükleyerek veya ara değerleri test ederek çoklu zamanlama yapılabilir. Bu dört sözcükten birinin değeri 0'dan küçükse, değiştirilmeyecektir. Bir zamanlayıcı karşılık gelen bit 15'i "1"e kurarak "dondurulabilir", ve sonra aynı bit "0"lanarak "çözülebilir".

Programlama Örneği

Aşağıdaki örnekte bir zamanlayıcının programlanması görülmektedir.

| | | |
|-------|---------------|--|
| LDR | %I0.1 | (%I0.1'in yükselen kenarında, zamanlayıcıyı başlatmak) |
| | [%SW76:=XXXX] | (XXXX = istenen değer) |
| LD | %I0.2 | (donmanın opsiyonel yönetimi, I0.2 girişi donar) |
| ST | %SW76:X15 | |
| LD | [%SW76=0] | (zamanlayıcı son reset) |
| ST | %M0 | |
| | | |



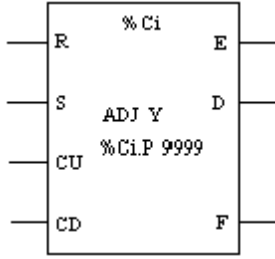
Şekil-52 Zamanlayıcı programlama örneği.

3.6.3 Yukarı/Aşağı Sayıcı (Counter) Fonksiyon Bloğu (%Ci)

Açıklama: Sayıcı fonksiyon bloğu (%Ci) olayların yukarı veya aşağı doğru sayılması için kullanılır. Bu iki işlem ardışık olarak yapılabilir.

Gösterim

Aşağıdaki Şekil-54 Yukarı/Aşağı Sayıcı Fonksiyon Bloğu gösterimidir.



Şekil-53 Yukarı/aşağı Sayıcı Fonksiyon Bloğu.

Parametreler

Yukarı/Aşağı Sayıcı Fonksiyon Bloğu aşağıdaki parametrelere sahiptir:

Tablo-30 Sayıcı Fonksiyon Bloğu parametreleri

| Parametre | Etiket | Değer |
|--|-----------|--|
| Sayıcı no.su | %Ci | 0 - 127 |
| Mevcut Değer | %Ci.V | Sözcük CU ve CD girişlerine (veya buyruklarına) göre artırılır veya azaltılır. Okunabilir ve test edilebilir, fakat program tarafından yazılamaz. %Ci.V Animasyon Tablosu Editörü kullanılarak değiştirilebilir. |
| Önayar Değeri | %Ci.P | $0 \leq \%Ci.P \leq 9999$. Okunabilir, test edilebilir ve program tarafından yazılabilir sözcük. Önceden tanımlı değer 9999. |
| Animasyon Tablosu Editörü ile düzenlemek | ADJ | Y: Yes (Evet), %Ci.P önayar değeri Animasyon Tablosu Editörü kullanılarak değiştirilebilir. N:No(Hayır),%Ci.P önayar değeri değiştirilemez |
| Reset girişi (veya buyruğu) | R | “1” durumunda: %Ci.V = 0. |
| Set girişi (veya buyruğu) | S | “1” durumunda: %Ci.V = %Ci.P. |
| Yukarı sayma girişi(veya buyruğu) | CU | Yükselen kenarda %Ci.V’yi bir artırır. |
| Aşağı sayma girişi (veya buyruğu) | CD | Yükselen kenarda %Ci.V’yi bir azaltır. |
| Aşağı sayma taşma çıkışı | E (Empty) | Aşağı sayıcıda %Ci.V 0’dan 9999’a değiştiğinde ilgili %Ci.E biti = 1 olur (%Ci.V 9999’a ulaştığında “1”e kurulur ve sayıcı aşağı saymaya devam ederse “0”lanır). |
| Önayar değerine erişildi | D (Done) | %Ci.V=%Ci.P olduğunda ilgili %Ci.D biti= 1 olur. |
| Yukarı sayma taşma çıkışı | F (Full) | Yukarı sayıcıda %Ci.V 9999’dan 0’a değiştiğinde ilgili %Ci.F biti= 1 olur (%Ci.V 0’a ulaştığında “1”e kurulur ve sayıcı yukarı saymaya devam ederse “0”lanır). |

Çalışma:

Tablo-31'de Yukarı/aşağı sayıcının çalışması açıklanmıştır.

Tablo-31 Yukarı/aşağı sayıcının çalışması

| Çalışma | Hareket | Sonuç |
|----------------------|--|---|
| Sayma | Yukarı sayma CU girişinde bir yükselen kenar algılanır (veya CU buyruğu etkin olur). | %Ci.V mevcut değeri bir artar. |
| | %Ci.V mevcut değeri %Ci.P önayar değerine eşit olur. | %Ci.D “önayar değerine ulaşıldı” çıkış biti “1” olur. |
| | %Ci.V mevcut değeri 9999'dan 0'a değişir. | %Ci.F çıkış biti (yukarı sayma taşması) “1” olur. |
| | Sayıcı yukarı saymaya devam ederse, | %Ci.F çıkış biti “0”lanır. |
| Aşağı sayma | Aşağı sayma CD girişinde bir yükselen kenar algılanır (veya CD buyruğu etkin olur). | %Ci.V mevcut değeri bir azalır. |
| | %Ci.V mevcut değeri 0'dan 9999'a değişir. | %Ci.E çıkış biti (aşağı sayma taşması) “1” olur. |
| | Sayıcı aşağı saymaya devam ederse. | %Ci.E çıkış biti “0”lanır. |
| Yukarı / Aşağı sayma | Yukarı ve aşağı sayma fonksiyonlarını eşzamanlı kullanmak (veya CU ve CD girişlerini etkinleştirmek) için, karşılık gelen CU ve CD girişleri eşzamanlı olarak kontrol edilmelidir. Sonra bu iki giriş ard arda taranır. Her ikisi de 1 ise, mevcut değer değişmeden kalır. | |
| Reset (Sıfırlama) | R girişi 1 durumuna kurulur (veya R buyruğu etkin olur). | %Ci.V mevcut değeri 0'lanır. %Ci.E, %Ci.D ve %Ci.F çıkışları 0 olur. Reset girişinin önceliği vardır. |
| Preset (Kurma) | S girişi “1”e kurulduğunda (veya S buyruğu etkin olduğunda) ve reset girişi 0 ise (R buyruğu etkin değilse). | %Ci.V mevcut değeri %Ci.P'ye eşit olur ve %Ci.D çıkışı “1”e kurulur. |

Özel Durumlar

Tablo-32 Sayıcı fonksiyon bloğunun programlanmasında özel durumlar.

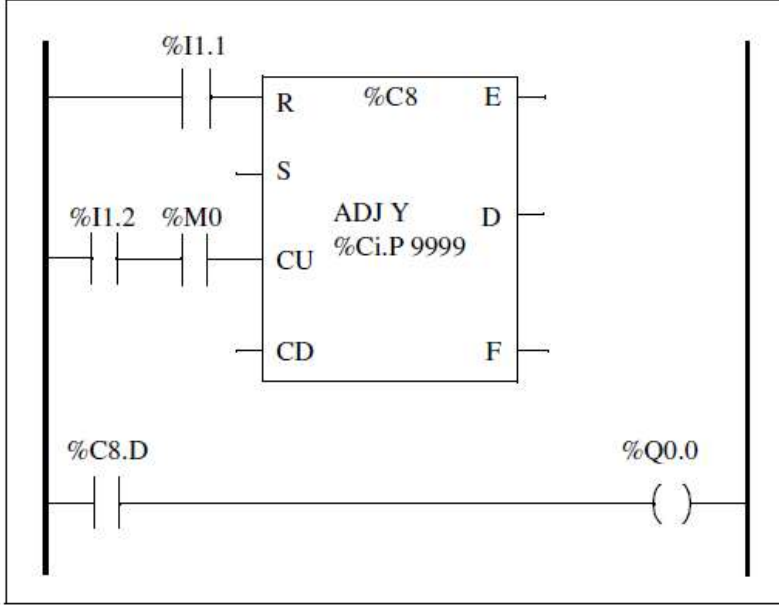
| Özel Durum | Tanım |
|--|---|
| Soğuk yeniden başlatmanın etkisi (%S0=1) | <ul style="list-style-type: none">Mevcut değer %Ci.V sıfırlanır.%Ci.E, %Ci.D ve %Ci.F çıkışları 0 olur.Önayar değeri konfigürasyon sırasında tanımlanan değere eşitlenir. |
| Sıcak yeniden başlatmanın etkisi (%S1=1) | Sayıcının mevcut değeri (%Ci.V) üzerinde etkisi yoktur. |
| %Ci.P'nin değiştirilmesinin etkisi | Bir buyruk kullanarak veya ayarlayarak önayar değerinin değiştirilmesi, blok uygulama tarafından işlendiğinde etkili olur. |

3.6.3.1 Sayıcıların Programlanması ve Yapılandırması

Açıklama: Aşağıdaki örnek 5000 parçaya kadar saymayı sağlayan bir sayıcıdır. %M0 dahili biti "1"e kurulduğunda %I1.2 girişindeki her vuru %C8 sayıcısını son değerine kadar bir (bit %C8.D=1). Sayıcı %I1.1 girişi ile sıfırlanır.

Örnekler

Aşağıdaki gösterimde sayıcı fonksiyon bloğunun dönüştürülebilir ve dönüştürülemez programlama örnekleri bulunmaktadır.



Ladder diyagramı

| | |
|---------|-------|
| BLK | %C8 |
| LD | %I1.1 |
| R | |
| LD | %I1.2 |
| AND | %M0 |
| CU | |
| END_BLK | |
| LD | %C8.D |
| ST | %Q0.0 |

Dönüştürülebilir programlama

| | |
|-----|-------|
| LD | %I1.1 |
| R | %C8 |
| LD | %I1.2 |
| AND | %M0 |
| CU | %C8 |
| LD | %C8.D |
| ST | %Q0.0 |

Dönüştürülemez programlama

Şekil-54 Yukarı sayıcı programlama örneği.

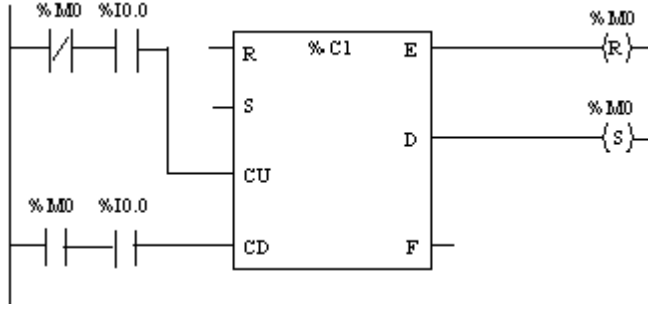
Yapılandırma

Yapılandırma sırasında aşağıdaki parametreler girilmelidir:

- Önarar değeri (%Ci.P): Bu örnekte 5000'e ayarlı

Yukarı/Aşağı Sayıcı Örneği

Aşağıdaki gösterim yukarı/aşağı sayıcı fonksiyon bloğu örneğidir.



Şekil-55 Yukarı/aşağı sayıcı programlama örneği.

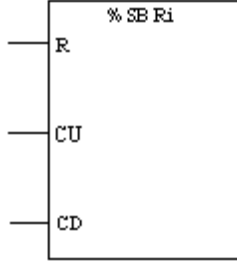
Bu örnekte, %C1.P = 4 ise, sayıcının mevcut değeri %C1.V 0'dan 3'e kadar artırılabilecektir, sonra 3'ten 0'a doğru azalacaktır. %I0.0=1 olduğunda %C1.V 0 ile 3 arasında salınacaktır.

3.6.4 Bit Kaydırma Yazacı (Shift Bit Register) Fonksiyon Bloğu (%SBRi)

Açıklama: Bit Kaydırma Yazacı fonksiyon bloğu (%SBRi) ikilik veri bitlerinin (0 veya 1) sağa veya sola kaydırılması sağlar.

Gösterim

Aşağıdaki Şekil-56 Bit Kaydırma Yazacı fonksiyon bloğu gösterimidir.



Şekil-56 Bit Kaydırma Yazacı fonksiyon bloğu

Parametreler

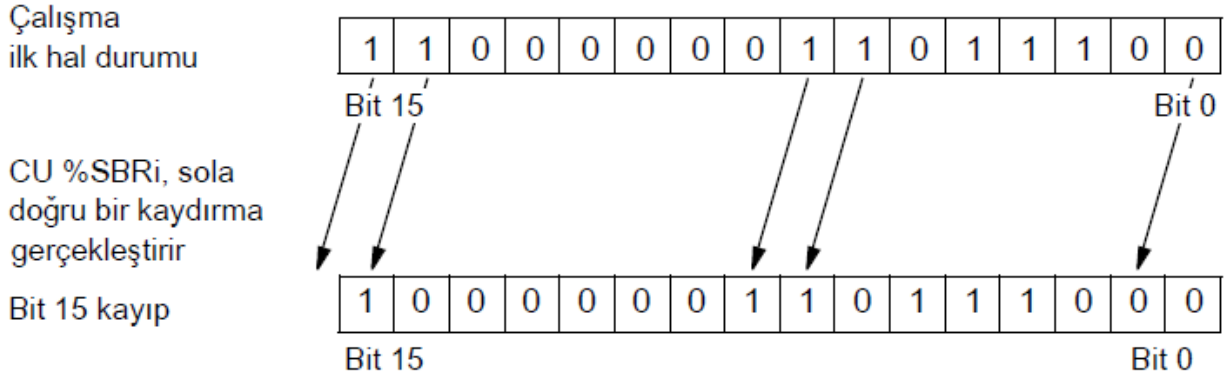
Bit Kaydırma Yazacı Fonksiyon Bloğu aşağıdaki parametrelere sahiptir:

Tablo-33 Bit Kaydırma Yazacı Fonksiyon Bloğu parametreleri

| Parametre | Etiket | Değer |
|-----------------------------------|---------|--|
| Yazaç no.su | %SBRi | 0 - 7 |
| Yazaç biti | %SBRi.j | Kayar yazacın 0 ile 15 (j = 0 ile 15) arası bitlerini bir test buyruğu ile test edilebilir ve bir atama buyruğu ile yazılabilir. |
| Reset girişi (veya buyruğu) | R | Yükselen kenarda, %SBRi.j'nin 0 ile 15 arası bitlerini 0'lar. |
| Set girişi (veya buyruğu) | S | "1" durumunda: %Ci.V = %Ci.P. |
| Sola kaydır girişi(veya buyruğu) | CU | Yükselen kenarda, bir yazaç bitini sola kaydırır. |
| Sağa kaydır girişi (veya buyruğu) | CD | Yükselen kenarda, bir yazaç bitini sağa kaydırır. |

Çalışma:

Aşağıdaki Şekil-57 Bit Kaydırma Yazacı kaydırma işleminin önce ve sonrasındaki bit örüntüsü gösterimidir.

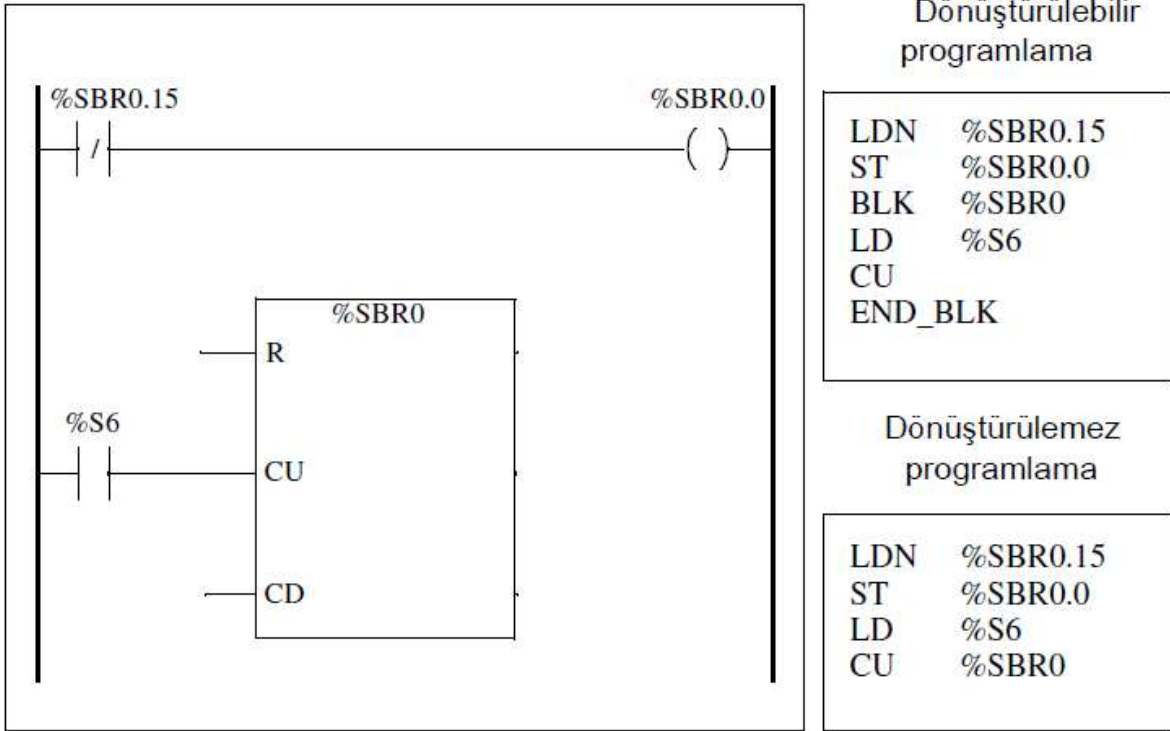


Şekil-57 Bit Kaydırma Yazacının bit örüntüsü

Bu çalışma şekli CD buyruğu kullanılarak sağa doğru (Bit 15'ten Bit 0'a) kaydırma isteğinde de geçerlidir. Bit 0 kaybolur.

Programlama

Aşağıdaki örnekte, CU girişine bağlı %S6 sistem biti aracılığıyla bir bit her saniyede sola doğru kaydırılırken Bit 0 ise Bit 15'in değerinin tersine eşit olmaktadır.



Şekil-58 Bit Kaydırma Yazacı program örneği

Özel Durumlar

Tablo-34'te Bit Kaydırma Yazacının programlanması için özel durumlar listelenmiştir.

Tablo-34 Bit Kaydırma Yazacının programlanmasında özel durumlar.

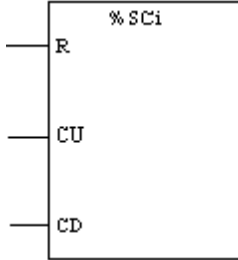
| Özel Durum | Tanım |
|--|--|
| Soğuk yeniden başlatmanın etkisi (%S0=1) | Yazaç sözcüğünün tüm bitleri sıfırlanır. |
| Sıcak yeniden başlatmanın etkisi (%S1=1) | Yazaç sözcüğünün bitleri üzerinde etkisi yoktur. |

3.6.5 Adım Sayacı (Step Counter) Fonksiyon Bloğu (%SCi)

Açıklama: Bir Adım Sayacı fonksiyon bloğu (%SCi) eylemlerin atanabileceği bir seri adım dizisi sağlar. Bir adımdan diğer adıma geçiş harici veya dahili olaylara dayanmaktadır. Bir adımın etkin olduğu her sefer ilgili bit "1"e kurur. Bir defada adım sayacının yalnızca bir adımı etkin olabilir.

Gösterim

Aşağıdaki Şekil-60 Adım Sayacı fonksiyon bloğu gösterimidir.



Şekil-59 Adım Sayacı fonksiyon bloğu

Parametreler

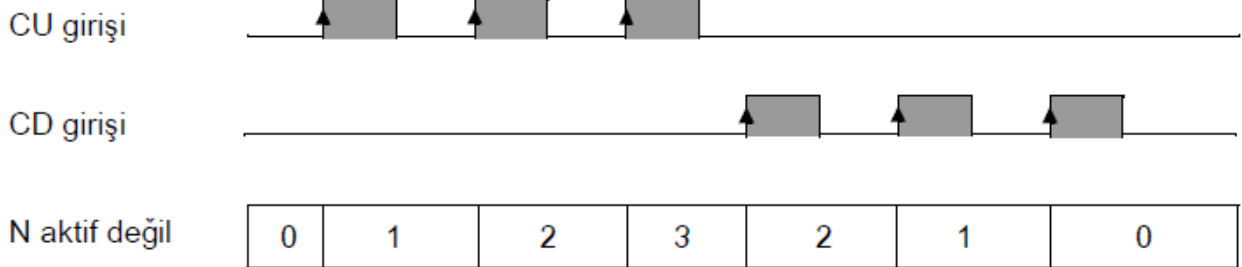
Adım Sayacı Fonksiyon Bloğu aşağıdaki parametrelere sahiptir:

Tablo-35 Adım Sayacı Fonksiyon Bloğu parametreleri

| Parametre | Etiket | Değer |
|-------------------------------|--------|---|
| Adım Sayacı no.su | %SCi | 0 - 7 |
| Adım Sayacı biti | %SCi.j | Adım Sayacı 0 ile 255 (j = 0 ile 255) arası bitleri bir LOAD mantık buyruğu ile test edilebilir ve bir atama buyruğu ile yazılabilir. |
| Reset girişi (veya buyruğu) | R | Yükselen kenarda, adım sayacını sıfırlar. |
| Arttırma girişi(veya buyruğu) | CU | Yükselen kenarda, adım sayacını bir adım arttırır. |
| Azaltma girişi (veya buyruğu) | CD | Yükselen kenarda, adım sayacını bir adım azaltır. |

Zamanlama Diyagramı

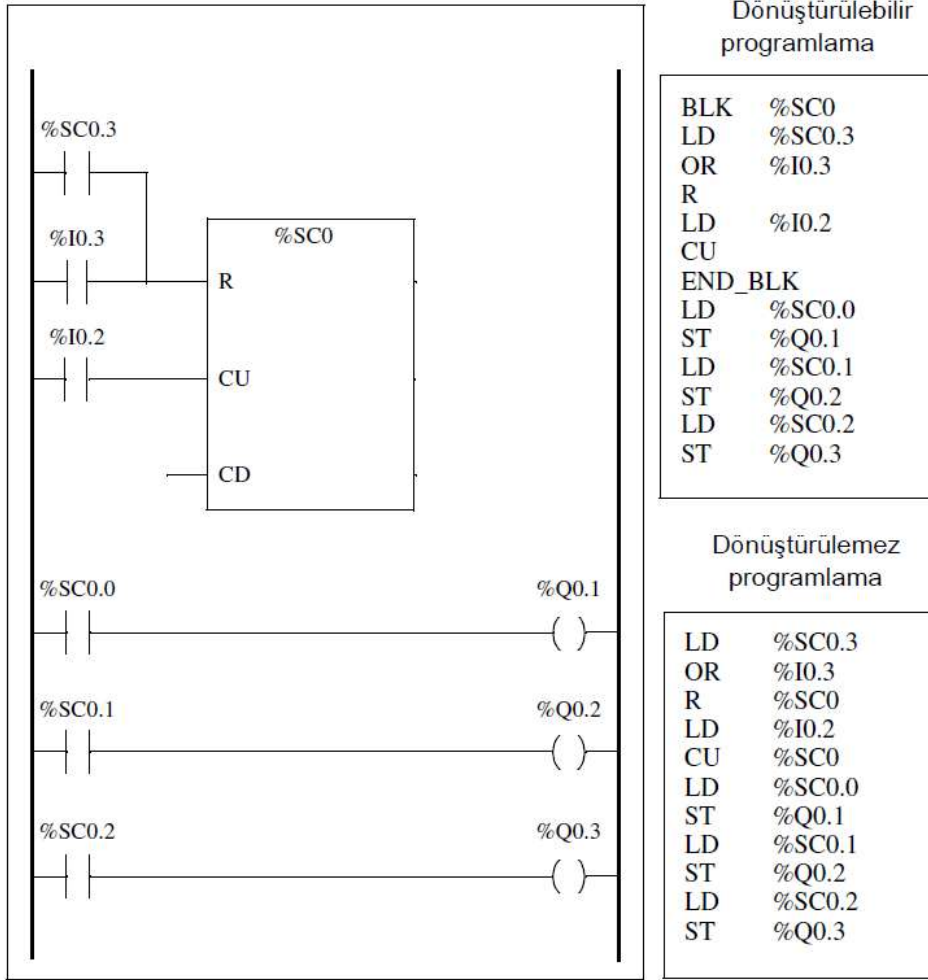
Aşağıdaki zamanlama diyagramı adım sayacı fonksiyon bloğunun çalışmasını göstermektedir.



Şekil-60 Adım Sayacı zamanlama diyagramı

Programlama

Aşağıdaki örnek bir Adım Sayacı fonksiyon bloğunun dönüştürülebilir ve dönüştürülemez programlanmasını göstermektedir. Step Counter 0 %I0.2 arttırılmakta, adım 3'e gelindiğinde %I0.3 girişi ile sıfırlanmaktadır. Adım 0 %Q0.1, adım 1 %Q0.2 çıkışını ve adım 2 %Q0.3 çıkışını kontrol eder.



Şekil-61 Adım Sayacı programlama örneği.

Özel Durumlar

Tablo-36'da Adım Sayacının programlanması için özel durumlar listelenmiştir.

Tablo-36 Adım Sayacının programlanması için özel durumlar.

| Özel Durum | Tanım |
|--|---|
| Soğuk yeniden başlatmanın etkisi (%S0=1) | Adım sayacı başlangıç durumuna getirilir. |
| Sıcak yeniden başlatmanın etkisi (%S1=1) | Adım sayacı üzerinde etkisi yoktur. |

3.7 İleri Fonksiyon Blokları

Açıklama: İleri fonksiyon blokları standart fonksiyon bloklarındakilere benzer atanmış sözcük ve bitlerini kullanırlar. İleri fonksiyon blokları şunlardır:

- LIFO/FIFO yazaçlar (registers) (%R)
- Tambur Denetleyiciler (Drum controllers) (%DR)
- Hızlı sayıcılar (Fast counters) (%FC)
- Çok Hızlı sayıcılar (Very fast counters) (%VFC)
- Vuru Genişlik Modülasyonlu çıkış (Pulse width modulation output) (%PWM)
- Vuru üretici çıkışı (Pulse generator output) (%PLS)
- Bit Kaydırma Yazacı (Shift Bit Register) (%SBR)
- Adım Sayacı (Shift counter) (%SC)
- Mesaj kontrol bloğu (Message control block) (%MSG)

Program Tarafından Erişilebilir Objeler

Aşağıdaki tabloda çeşitli ileri fonksiyon bloklarına ait program tarafından erişilebilir sözcükler ve bitlerin bir özeti bulunmaktadır. Tablodaki yazma erişiminin konfigürasyon sırasında “Ayarlanabilir” ayarının seçilmesine dayandığına dikkat edilmelidir. Bu ayarın seçilmesi TwidoSoft veya operatör arabirimi aracılığıyla sözcüklerin veya bitlerin erişimine izin verir veya izni vermez.

Tablo-37 İleri fonksiyon blokları

| İleri Fonksiyon Bloğu | İlgili Sözcükler ve Bitler | Adres | Yazma Erişimi | |
|-----------------------|----------------------------|----------------------------------|---------------|-----|
| %R | Sözcük | Yazaç girişi | %Ri.I | var |
| | Sözcük | Yazaç çıkışı | %Ri.O | var |
| | Bit | Yazaç çıkışı dolu | %Ri.F | yok |
| | Bit | Yazaç çıkışı boş | %Ri.E | yok |
| %DR | Sözcük | Mevcut adım no. | %DRi.S | var |
| | Bit | Mevcut adım son adıma eşit | %DRi.F | yok |
| %FC | Sözcük | Mevcut değer | %FCi.V | var |
| | Sözcük | Önayar değeri | %FCi.P | var |
| | Bit | İşlem tamam | %FCi.D | yok |
| %VFC | Sözcük | Mevcut değer | %VFCi.V | yok |
| | Sözcük | Önayar değeri | %VFCi.P | var |
| | Bit | Sayma yönü | %VFCi.U | yok |
| | Sözcük | Yakalama değeri | %VFCi.C | yok |
| | Sözcük | Eşik 0 değeri | %VFCi.S0 | var |
| | Sözcük | Eşik 1 değeri | %VFCi.S1 | var |
| | Bit | Taşma | %VFCi.F | yok |
| | Bit | Hesaplanan frekans | %VFCi.M | var |
| | Bit | Yansıma Çıkışı 0 İzni | %VFCi.R | var |
| | Bit | Yansıma Çıkışı 1 İzni | %VFCi.S | var |
| | Bit | Eşik Çıkışı 0 | %VFCi.TH0 | yok |
| | Bit | Eşik Çıkışı 1 | %VFCi.TH1 | yok |
| | Bit | Frekans Ölçümü Zaman Tabanı | %VFCi.T | var |
| %PWM | Sözcük | Vuru genişliğinin periyoda oranı | %PWMi.R | var |
| | Sözcük | Önayar değeri | %PWMi.P | var |
| %PLS | Sözcük | Vuru sayısı | %PLSi.N | var |
| | Sözcük | Önayar değeri | %PLSi.P | var |
| | Bit | Mevcut çıkış iznlenildi | %PLSi.Q | yok |
| | Bit | Vuru üretimi tamamlandı | %PLSi.D | yok |
| %SBR | Bit | Yazaç biti | %SBRi.j | yok |
| %SC | Bit | Adım sayacı biti | %SCi.j | var |
| %MSG | Bit | İşlem tamam | %MSGi.D | yok |
| | Bit | Hata | %MSGi.E | yok |

Not: Dönüştürülebilir fonksiyon blokları buyruklarının kullanımı List programlamanın düzgün çalışması için zorunlu değildir. Bazı buyruklar için dönüştürülebilir olmadan List dilinde programlama yapmak mümkündür.

3.7.1 İleri Fonksiyon Bloğu İleri Fonksiyon Blokları için Programlama İlkeleri

Tüm Twido uygulamaları Ladder Editor'da bile yazılımlar List programları olarak saklanırlar, dolayısıyla, Twido denetleyicilere List "makinelere" denir. "Dönüştürülebilir" terimi TwidoSoft'un List uygulamasını Ladder'a ve tekrar List'e dönüştürebilme yeteneğini ifade eder. Kendiliğinden tüm Ladder programları dönüştürülebilirdir.

Temel fonksiyon bloklarında olduğu gibi, ileri fonksiyon bloklarında da dönüştürülebilme kuralları dikkate alınmalıdır. List programlama dilinde dönüştürülebilir fonksiyon bloklarının yapısı aşağıdaki talimatları yerine getirmeyi gerektirir:

- BLK: Blok başlangıcını ve fonksiyon bloğunun giriş bölümünü belirtir.
- OUT_BLK: Fonksiyon bloğunun çıkış bölümünün başlangıcını belirtir.
- END_BLK: Blok sonunu belirtir.

Adanmış Girişler ve Çıkışlar

Hızlı Sayıcı, Çok Hızlı Sayıcı, PLS ve PWM ileri fonksiyonları atanmış giriş ve çıkışları kullanırlar, fakat bu bitler herhangi bir bloğun özel kullanımına ayrılmamıştır. Daha çok, bu atanmış kaynakların kullanımı yönetilmelidir..

Bu ileri fonksiyonları kullanırken, atanmış giriş ve çıkışların nasıl ayrılmış olduğunun yönetilmesi gerekir. TwidoSoft bu kaynakların yapılandırılmasına giriş/çıkış yapılandırma ayarlarını göstererek ve atanmış bir giriş veya çıkış halihazırda yapılandırılmış bir fonksiyon bloğu tarafından kullanılıyorsa uyarı vererek yardım eder.

Aşağıdaki tablolar atanmış giriş ve çıkışların ve özel fonksiyonların bağımlılığını özetlemektedir.

Tablo-38

Sayma işlevleri ile kullanıldığında:

| Girişler | Kullanımı |
|----------|--|
| %I0.0.0 | %VFC0: Yukarı/Aşağı idaresi veya Faz B |
| %I0.0.1 | %VFC0: Vuru girişi veya Faz A |
| %I0.0.2 | %FC0: Vuru girişi veya %VFC0 önayar girişi |
| %I0.0.3 | %FC1: Vuru girişi veya %VFC0 yakalama girişi |
| %I0.0.4 | %FC2: Vuru girişi veya %VFC1 yakalama girişi |
| %I0.0.5 | %VFC1 önayar girişi |
| %I0.0.6 | %VFC1: Yukarı/Aşağı idaresi veya Faz B |
| %I0.0.7 | %VFC1: Vuru girişi veya Faz A |

Sayma işlevleri ile veya özel fonksiyonlarla kullanıldığında:

| Çıkışlar | Kullanımı |
|----------|------------------------------|
| %Q0.0.0 | %PLS0 veya PWM0 çıkışı |
| %Q0.0.1 | %PLS1 veya PWM1 çıkışı |
| %Q0.0.2 | %VFC0 için yansıma çıkışları |
| %Q0.0.3 | |
| %Q0.0.4 | %VFC1 için yansıma çıkışları |
| %Q0.0.5 | |

Not: Atanmış G/Ç kullanımını değiştirmek için, obje tipini “kullanılmıyor” (not used) olarak ayarlayarak fonksiyon bloğunun yapılandırmasını değiştirin ve sonra uygulamanızdaki fonksiyon bloğu referanslarını kaldırın.

Adanmış Girişler ve Çıkışların Kullanımı

TwidoSoft'ta atanmış giriş ve çıkışların kullanımı için aşağıdaki kuralların uygulanması zorunludur.

- Atanmış G/Ç kullanan her fonksiyon bloğu yapılandırılmalı ve sonra uygulamada kaynak gösterilmelidir. Atanmış G/Ç yalnızca bir fonksiyon bloğu yapılandırıldığında o fonksiyon bloğu için ayrılmıştır ve bir program içinde kullanıldıysa, ayrılmamıştır.

- Bir fonksiyon bloğu yapılandırıldıktan sonra, atanmış giriş ve çıkışı uygulama veya başka bir fonksiyon bloğu tarafından kullanılamaz.

Örneğin, %PLS0'ı yapılandırırsanız, %DR0'da (drum controller) veya uygulama mantığında (ST %Q0.0.0 şeklinde) %Q0.0.0'ı kullanamazsınız.

- Uygulama veya diğer bir fonksiyon bloğu tarafından zaten kullanılan atanmış bir giriş veya çıkışa bir fonksiyon bloğu gereksinim duyarsa, bu fonksiyon bloğu yapılandırılmaz.

Örneğin, %FC0'ı bir yukarı sayıcı olarak yapılandırırsanız, %VFC0'ı %I0.0.2'yi yakalama girişi olarak kullanmak için yapılandıramazsınız.

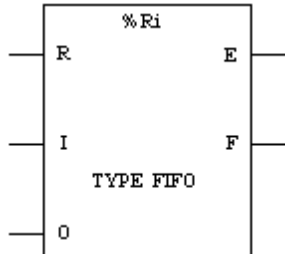
3.7.2 LIFO/FIFO Yazaç Fonksiyon Bloğu (%Ri)

Açıklama: Bir yazaç iki farklı şekilde 16 bitlik 16 sözcüğe kadar depolama yapabilen bir bellek bloğudur:

- Sıra (Queue: First In, First Out, İlk Giren İlk Çıkar): FIFO olarak adlandırılır.
- Yığın (Stack: Last In, First Out, Son Giren İlk Çıkar) LIFO olarak adlandırılır.

Gösterim

Aşağıdaki Şekil-63 Yazaç fonksiyon bloğu gösterimidir.



Şekil-62 Yazaç fonksiyon bloğu

Parametreler

Yazaç Fonksiyon Bloğu aşağıdaki parametrelere sahiptir:

Tablo-39 Yazaç Fonksiyon Bloğu parametreleri

| Parametre | Etiket | Değer |
|--------------------|----------------|--|
| Yazaç no.su | %Ri | 0 - 3 |
| Tipi | FIFO veya LIFO | Sıra veya Yığın. |
| Giriş sözcüğü | %Ri.I | Yazaç giriş sözcüğü. Okunabilir, test edilebilir ve yazılabilir. |
| Çıkış sözcüğü | %Ri.O | Yazaç çıkış sözcüğü. Okunabilir, test edilebilir ve yazılabilir. |
| Saklama girişi | I (In) | Yükselen kenarda, %Ri.I sözcüğünün içeriğini saklar. |
| Geri erişim girişi | O (Out) | Yükselen kenarda, yazacın bir veri sözcüğünü %Ri.O sözcüğüne yükler. |

| | | |
|-----------------------------|-----------|--|
| Reset giriři (veya buyruęu) | R | “1” durumunda, yazacı bařlangıç deęerlerine getirir. |
| Boř çıkıřı | E (Empty) | %Ri.E ilgili biti yazacın boř olduęunu belirtir. Test edilebilir. |
| Dolu çıkıřı | F (Full) | %Ri.F ilgili biti yazacın dolu olduęunu belirtir. Test edilebilir. |

LIFO, alıřması

Açıklama: LIFO iřleminde (Last In, First Out), girilen son veri öęesi ilk geri çağrılan veri olur.

alıřma:

Tablo-40'ta LIFO alıřması açıklanmıřtır.

Tablo-40 LIFO'nun alıřması

| Adım | Tanım | Örnek |
|------|---|---|
| 1 | I giriřindeki bir yükselen kenar veya I buyruęunun etkinleřtirilmesiyle bir saklama isteęi alındıęında, önceden yüklenmiř olan %Ri.I giriř sözcüęünün içerięi yığının tepesinde saklanır (řek. a). Yığın dolu olduęunda (çıkıř F=1), daha fazla yükleme yapılamaz. | <p>%Ri.I'nın içerięinin yığının tepesine yazılması.</p> |
| 2 | O çıkıřındaki bir yükselen kenar veya O buyruęunun etkinleřtirilmesiyle bir geri eriřim isteęi alındıęında, en yüksekteki veri sözcüęü (girilen en son sözcük) %Ri.O çıkıř sözcüęüne yüklenir (řek. b). Yazaç boř olduęunda (çıkıř E=1), daha fazla geri çağırma yapılamaz. %Ri.O çıkıř sözcüęü deęiřmez ve deęerini korur. | <p>Yığının en tepesindeki veriye eriřim.</p> |
| 3 | R giriřine “1” uygulanarak veya R buyruęu etkinleřtirilerek yığın her zaman sıfırlanabilir. Yığın göstergesinin gösterdięi eleman artık yığının tepesidir. | |

FIFO, alıřması

Açıklama: FIFO iřleminde (First In, First Out), girilen ilk veri öęesi ilk geri çağrılan veri olur.

alıřma:

Tablo-41'de FIFO alıřması açıklanmıřtır.

Tablo-41 FIFO'nun çalışması

| Adım | Tanım | Örnek |
|------|--|---|
| 1 | I girişindeki bir yükselen kenar veya I buyruğunun etkinleştirilmesiyle bir saklama isteği alındığında, önceden yüklenmiş olan %Ri.I giriş sözcüğünün içeriği sıranın tepesinde saklanır (Şek. a). Sıra dolu olduğunda (çıkış F=1), daha fazla yükleme yapılamaz. | <p>%Ri.I'nın içeriğinin sıranın tepesine yazılması.</p> <p>(a)</p> |
| 2 | O çıkışındaki bir yükselen kenar veya O buyruğunun etkinleştirilmesiyle bir geri erişim isteği alındığında, en alttaki veri sözcüğü %Ri.O çıkış sözcüğüne yüklenir ve yazacın içeriği sırada aşağıya doğru bir yer ilerler (Şek. b). Yazaç boş olduğunda (çıkış E=1), daha fazla geri çağırma yapılamaz. %Ri.O çıkış sözcüğü değişmez ve değerini korur. | <p>%Ri.O'ya yüklenen ilk veri ögesinin geri çağırılması.</p> <p>(b)</p> <p>% Ri.O</p> |
| 3 | R girişine "1" uygulanarak veya R buyruğu etkinleştirilerek sıra her zaman sıfırlanabilir. | |

Yazaçların Programlanması ve Yapılandırılması

Açıklama: Aşağıdaki örnek, depolama isteği (%I0.2) üzerine, %R2 yazacı dolu değilse (%R2.F = 0), bir bellek sözcüğünün (%MW34) içeriğinin bir yazaca (%R2.I) yüklenmesini göstermektedir. Yazaçtaki depolama isteği %M1 ile yapılmıştır. Erişim isteği %I0.3 girişiyle yapılmaktadır ve yazaç boş değilse (%R2.E = 0), %R2.O %MW20'ye yüklenir.

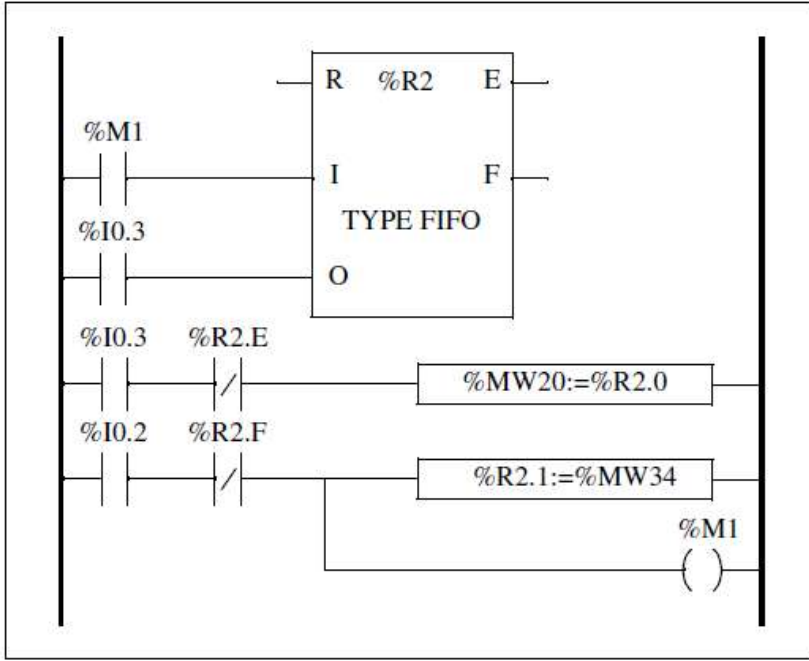
Yapılandırma

Yapılandırma sırasında girilecek tek parametre yazaç tipidir:

- FIFO (önceden tanımlı) veya
- LIFO

Programlama Örneği

Aşağıdaki gösterimde yazaç fonksiyon bloğunun dönüştürülebilir ve dönüştürülemez programlama örnekleri bulunmaktadır.



Ladder diyagramı

| | | | |
|----------------|-------|----------------|-------|
| BLK | %R2 | LD | %M1 |
| LD | %M1 | I | %R2 |
| I | %M1 | LD | %I0.3 |
| LD | %I0.3 | O | %R2 |
| O | %R2 | ANDN | %R2.E |
| END_BLK | %R2 | [%MW20:=%R2.0] | |
| LD | %I0.3 | LD | %I0.2 |
| ANDN | %R2.E | ANDN | %R2.F |
| [%MW20:=%R2.0] | %R2.E | [%R2.1:=%MW34] | |
| LD | %I0.2 | ST | %M1 |
| ANDN | %R2.F | | |
| [%R2.1:=%MW34] | %R2.F | | |
| ST | %M1 | | |

Dönüştürülebilir program

Dönüştürülemez program

Şekil-63 Yazaç fonksiyon bloğu ile programlama örneği.

Özel Durumlar

Tablo-42’de Yazaçların programlanması için özel durumlar listelenmiştir.

Tablo-42 Yazaçların programlanması için özel durumlar.

| Özel Durum | Tanım |
|---|--|
| Soğuk yeniden başlatmanın etkisi (%S0=1) | Yazaçın içeriğini başlangıç durumuna getirir. E çıkışı ile ilgili çıkış biti %Ri.E “1”e kurur. |
| Denetleyicinin durması üzerine sıcak yeniden başlatmanın etkisi (%S1=1) | Yazaçın mevcut değeri yada çıkış bitlerinin durumları üzerinde etkisi yoktur. |

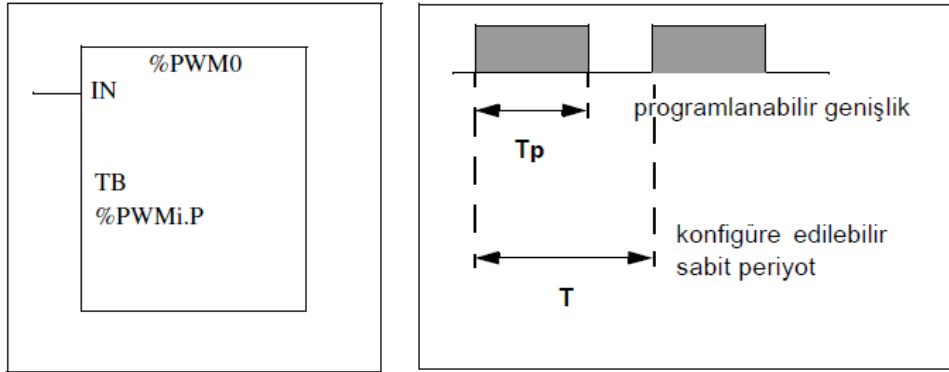
3.7.3 Vuru Genişlik Modülasyon Çıkışı (Pulse Width Modulation Output) Fonksiyon Bloğu, %PWM

Açıklama: Vuru Genişlik Modülasyonu (Pulse Width Modulation: %PWM) fonksiyon bloğu %Q0.0.0 veya %Q0.0.1 adanmış çıkış kanallarında değişken genişlik ve dolayısıyla değişken görev süresine sahip kare dalga üretir. Frekans sınırlaması nedeniyle röle çıkışlı denetleyicilerde bu iki kanal bu fonksiyonu desteklememektedir.

İki %PWM bloğu bulunmaktadır. %PWM0 %Q0.0.0 atanmış çıkışını ve %PMW1 %Q0.0.1 atanmış çıkışını kullanır. %PLS fonksiyon bloğu da aynı adanmış çıkışları kullandığı için çakışma olmaktadır, bu nedenle iki fonksiyon arasında seçim yapılmalıdır.

Gösterim

Aşağıdaki Şekil-64 %PWM fonksiyon bloğu gösterimidir.



Şekil-64 %PWM fonksiyon bloğu

%PWM Fonksiyon Bloğu aşağıdaki parametrelere sahiptir:

Tablo-43 %PWM Fonksiyon Bloğu parametreleri

| Parametre | Etiket | Tanım |
|---------------------------|---------|---|
| Zaman Tabanı | TB | 0.142 ms, 0.57 ms, 10 ms, 1 s (önceden tanımlı) |
| Periyot ön seçimi | %PwMi.P | $0 < \%PwMi.P \leq 32767$ (TB 10 ms veya 1 s için) $0 < \%PwMi.P \leq 255$ (TB 0.57 ms veya 0.142 ms için) 0 = Fonksiyon kullanımda değil |
| Görev Süresi (Duty Cycle) | %PwMi.R | Bu değer bir periyotta sinyalin 1 durumunda olma yüzdesini verir. Dolayısıyla, vuru genişliği T_p : $T_p = T \times (\%PwMi.R/100)$. Kullanıcı uygulaması %PwMi.R değerini yazar. Görev süresini kontrol eden bu sözcüktür. Önceden tanımlı değeri 0'dır ve 100'den büyük girilen değerler 100'e eşit olarak kabul edilir. |
| Vuru üretme girişi | IN | 1 durumunda, vuru genişlik modülasyonu sinyali çıkış kanalında üretilir. 0 durumunda, çıkış kanalı sıfırlanır. |

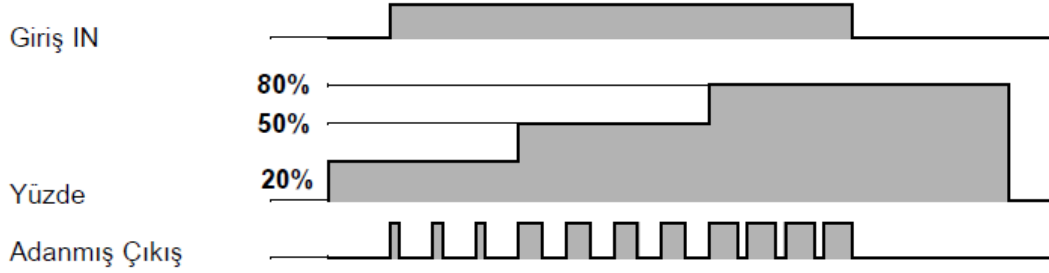
Periyot Aralığı

Önayar değeri ve zaman tabanı yapılandırma sırasında değiştirilebilir. Bu değerler sinyal periyodunu $T = \%PwMi.P * TB$ belirler. Düşük oranların elde edilebilmesi için %PwMi.P'nin büyük seçilmesi gerekir. Mevcut periyot aralıkları:

- 0.142 ms ile 36.5 ms arası, 0.142 ms'lik adımlarla (27.4Hz ile 7kHz arası)
- 0.57 ms ile 146 ms arası, 0.57 ms'lik adımlarla (6.84 Hz ile 1.75 kHz arası)
- 10 ms ile 5.45 dak. arası, 10 ms'lik adımlarla
- 1 s ile 9.1 saat arası, 1 s'lik adımlarla

Çalışma

Çıkış sinyalinin frekansı zaman tabanı TB ve önayar değeri %PwMi.P'nin yapılandırma sırasında seçimi ile ayarlanır. Programda %PwMi.R görev süresinin değiştirilmesi sinyalin genişliğini modüle eder. Aşağıdaki gösterimde PWM fonksiyon bloğunun değişen görev süreleri ile değişen vuru diyagramı bulunmaktadır.



Şekil-65 PWM fonksiyon bloğu vuru diyagramı.

Programlanma ve Yapılandırma

Bu örnekte, sinyal genişliğinin denetleyici girişleri %I0.0.0 ve %I0.0.1'in durumlarına göre nasıl değiştirildiği gösterilmektedir.

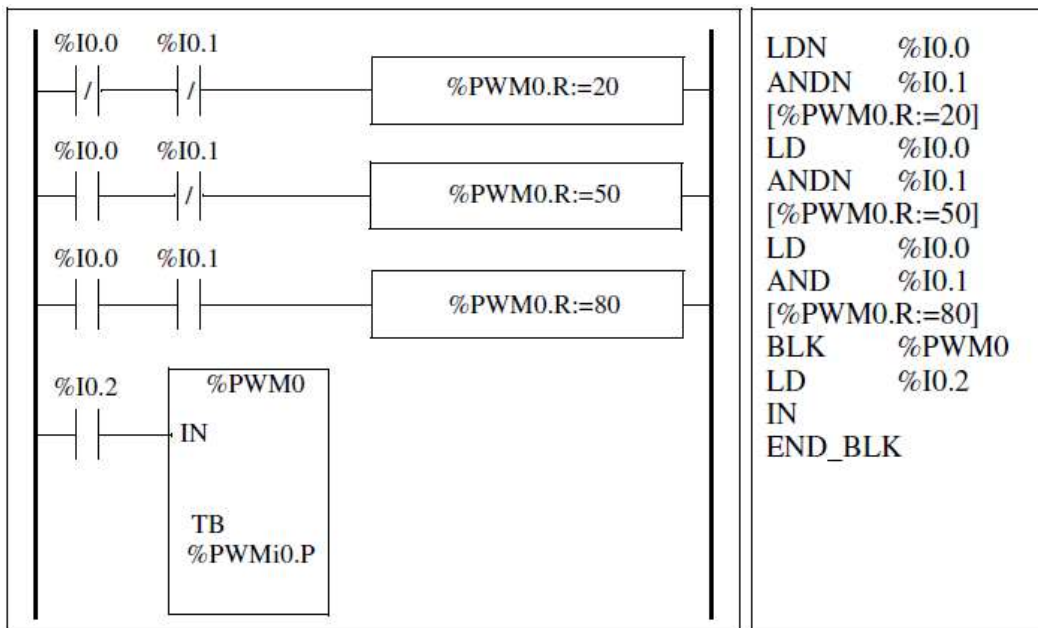
%I0.0.1 ve %I0.0.2 "0" a kurulursa, %PWM0.R oranı %20'ye ayarlanır, o zaman sinyalin "1" seviyesi süresi:

$\%20 \times 500 \text{ ms} = 100 \text{ ms}$ olur.

%I0.0.0 "0" a ve %I0.0.1 "1" e ayarlanırsa, %PWM0.R oranı %50 olur (süre 250 ms).

%I0.0.0 ve %I0.0.1 "1" e kurulursa, %PWM0.R oranı %80 olur (süre 400 ms).

Programlama Örneği:



Şekil-66 PWM fonksiyon bloğu program örneği.

Özel Durumlar

Tablo-44'te PWM fonksiyon bloğunun programlanması için özel durumlar listelenmiştir.

Tablo-44 PWM fonksiyon bloğunun programlanması için özel durumlar.

| Özel Durum | Tanım |
|--|---|
| Soğuk yeniden başlatmanın etkisi (%S0=1) | %PWMi.R oranını 0 yapar. Bun ek olarak %PWMi.P değeri yapılandırılmış değerine geri döner ve bu Animasyon Tablosu Editörü'nden veya Operatör Göstergesi'nden yapılan herhangi bir değişikliği de geçersiz kılar.. |
| Sıcak yeniden başlatmanın etkisi (%S1=1) | Hiçbir etkisi yoktur. |
| Çıkışların %PWM bloğuna atanmış olma gerçeğinin etkisi | Bir programlama aygıtı kullanarak %Q0.0.0 veya %Q0.0.1 çıkışını zorlamak (Zorlama) sinyal üretimini durdurmaz. |

3.7.4 Vuru Üretme Çıkış (Pulse Generation Output) Fonksiyon Bloğu, %PLS

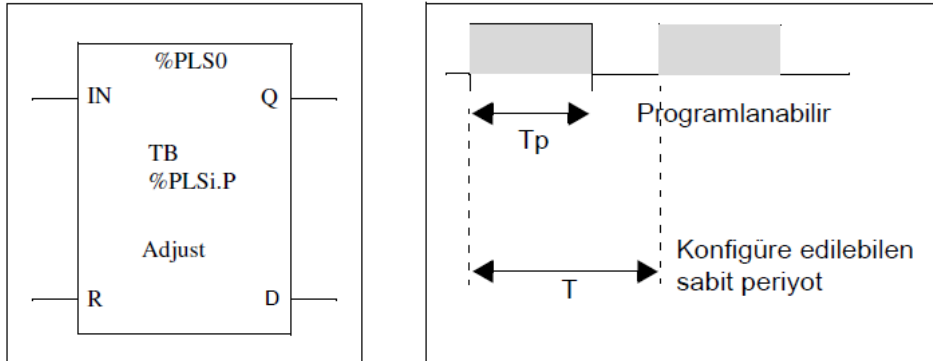
Açıklama: %PLS fonksiyon bloğu kare dalga sinyalleri üretmek için kullanılır. %Q0.0.0 veya

Not: Röle çıkışlı denetleyicilerde bu iki kanal %PLS fonksiyonunu desteklememektedir

%Q0.0.1 atanmış çıkış kanallarında iki %PLS fonksiyonu vardır. %PLS fonksiyon bloğu sadece bir sinyal genişliği veya %50 görev süresine olanak sağlar. Üretilen vuru adedi veya yürütülecek vuru dizisi sınırı seçilebilir. Bu değerler yapılandırma sırasında belirlenebilir ve/veya kullanıcı programı tarafından güncellenebilir.

Gösterim

Aşağıdaki Şekil-67 vuru üretme fonksiyon bloğu gösterimidir.



Şekil-67 %PLS fonksiyon bloğu.

- $TON = T/2$, 0.142ms ve 0.57ms zaman tabanları için.
 $= (\%PLSi.P * TB) / 2$
- $TON = [(\%PLSi.P) / 2] * TB$, 10ms ile 1s arası zaman tabanları için.

Özellikler

Aşağıdaki tablo PLS fonksiyon bloğunun karakteristiklerini içermektedir:

Tablo-45 %PLS Fonksiyon Bloğu parametreleri

| Parametre | Etiket | Tanım |
|-----------------------------|---------|---|
| Zaman Tabanı | TB | 0.142 ms, 0.57 ms, 10 ms, 1 s (önceden tanımlı) |
| Önayar periyodu | %PLSi.P | 0.142 ms ve 0.57 ms zaman tabanları için %PLS1.N'ye ulaşıldığında %PLS1 çıkışındaki vurular dudurulmaz. 1 < %PLSi.P <= 32767 (TB 10 ms veya 1 s için) 0 < %PLSi.P <= 255 (TB 0.57 ms veya 0.142 ms için) 0 = Fonksiyon kullanımında değil 10 ms ve 1 s zaman tabanları için görev süresinden iyi seviyede bir hassaslık isteniyorsa, P'nin tek olması durumunda %PLSi.P ≥ 100 yapılması önerilir. |
| Vuru Sayısı | %PLSi.N | T periyodunda üretilen vuru sayısı 0 < %PLSi.N <32767 ile sınırlıdır. Önceden tanımlı değeri 0'dır. Sınırsız sayıda vuru üretmek için %PLSi.N'nin "0" a ayarlanması gerekir. Vuru sayısı "Ayarlanabilir" ayarından bağımsız olarak her zaman değiştirilebilir. |
| Ayarlanabilir | Y/N | Y (Evet) seçilirse, %PLSi.P önayar değerini HMI veya Animasyon Tablosu Editörü aracılığıyla değiştirmek mümkündür. N (Hayır) önayar değeri erişimi olmadığı anlamına gelir. |
| Vuru üretme girişi | IN | 1 durumunda, vuru üretimi atanmış çıkış kanalında gerçekleştirilir. 0 durumunda, çıkış kanalı sıfırlanır. |
| Reset girişi (veya buyruğu) | R | 1 durumunda, %PLSi.Q ve %PLSi.D çıkışları 0'lanır. T periyodunda üretilen vuru sayısı da "0" olur. |
| Mevcut vuru üretme çıkışı | %PLSi.Q | 1 durumunda, yapılandırılan atanmış çıkış kanalında vurunun üretildiğini gösterir. |
| Vuru üretme "tamam" çıkışı | %PLSi.D | 1 durumunda, vuru üretiminin tamamlandığını belirtir. İstenen vuru adedine ulaşılmıştır. |

Not: İstenen vuru adedine ulaşıldığında %PLSx.D kurulur. IN girişini veya R girişini "1" e kurarak sıfırlanır.

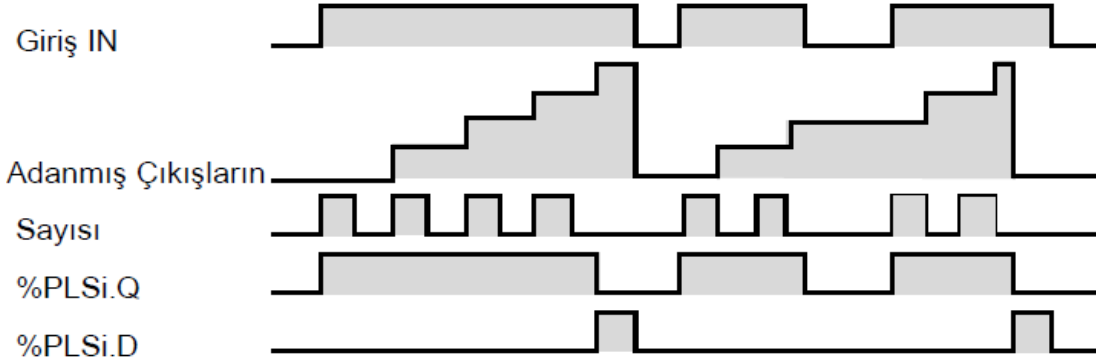
Periyot Aralığı

Önayar değeri ve zaman tabanı yapılandırma sırasında değiştirilebilir. Bu değerler sinyal periyodunu $T = \%PLSi.P * TB$ belirler. Mevcut periyot aralıkları:

- 0.142 ms ile 36.5 ms arası, 0.142 ms'lik adımlarla (27.4Hz ile 7kHz arası)
- 0.57 ms ile 146 ms arası, 0.57 ms'lik adımlarla (6.84 Hz ile 1.75 kHz arası)
- 20 ms ile 5.45 dak. arası, 10 ms'lik adımlarla
- 2 s ile 9.1 saat arası, 1 s'lik adımlarla

Çalışma

Aşağıdaki gösterimde %PLS fonksiyon bloğunun çalışması verilmiştir.



Şekil-68 %PLS fonksiyon bloğunun çalışmasını gösteren zamanlama diyagramı.

Özel Durumlar

Tablo-46'da %PLS fonksiyon bloğunun programlanması için özel durumlar listelenmiştir.

Tablo-46 %PLS fonksiyon bloğunun programlanması için özel durumlar.

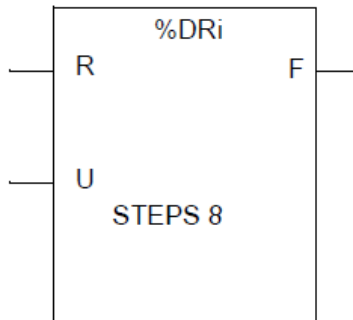
| Özel Durum | Tanım |
|--|--|
| Soğuk yeniden başlatmanın etkisi (%S0=1) | %PLSi.P değerini yapılandırılmış değerine ayarlar. |
| Sıcak yeniden başlatmanın etkisi (%S1=1) | Hiçbir etkisi yoktur. |
| Önayar (%PLSi.P) değerinin değiştirilmesi | Anında etki yapar. |
| Çıkışların %PLS bloğuna atanmış olma gerçeğinin etkisi | Bir programlama aygıtı kullanarak %Q0.0.0 veya %Q0.0.1 çıkışını zorlamak (Zorlama) sinyal üretimini durdurmaz. |

3.7.5 Tambur Denetleyici (Drum Controller) Fonksiyon Bloğu (%DR)

Açıklama: %DR fonksiyon bloğu harici olaylarla adım değiştiren elektromekanik tambur denetleyicilerin çalışma ilkesine benzer şekilde çalışır. Her adımda, kamın yüksek olan noktası denetleyici tarafından işletilen bir komut verir. Tambur denetleyicide ise bu yüksek noktalar her adımda "1" durumu ile sembolize edilir ve kontrol bitleri olarak bilinen %Qi.j çıkışlarına veya %Mi dahili bitlerine atanır.

Gösterim

Aşağıdaki Şekil-69 tambur denetleyici fonksiyon bloğu gösterimidir.



Şekil-69 %DR fonksiyon bloğu gösterimi.

Özellikler

Aşağıdaki tablo %DR fonksiyon bloğunun parametrelerini içermektedir:

Tablo-47 %DR Fonksiyon Bloğu parametreleri

| Parametre | Etiket | Değer |
|--------------------------------------|-----------|---|
| Yazaç no.su | %DRi | 0 – 3 (Kompakt Denetleyici), 0 – 7 (Modüler Denetleyici) |
| Mevcut adım no.su | %DRi.S | $0 < \%DRi.S < 7$. Okunabilir ve yazılabilir sözcük. Yazılan değer onluk mutlak değer olmalıdır. Yazıldığında, fonksiyon bloğunun bir sonraki işletiminde etki eder. |
| Adım sayısı | | 0 ile 8 arası |
| Adım 0'a dönme girişi (veya buyruğu) | R (Reset) | "1" durumunda, tambur denetleyiciyi adım 0'a getirir. |
| İlerle girişi (veya buyruğu) | U (Upper) | Yükselen kenarda, tambur denetleyicinin bir adım ilerlemesini sağlar ve kontrol bitleri güncellenir. |
| Çıkış | F (Full) | Mevcut adımın tanımlanan son adıma eşit olduğunu belirtir. İlgili %DRi.F biti test edilebilir (örneğin, %DRi.S = yapılandırılan adım sayısı – 1 ise, DRi.F=1 olur). |
| Kontrol bitleri | | Yapılandırma Editöründe tanımlanan adımlarla ilişkilendirilmiş çıkışlar veya dahili bitler (16 kontrol biti) |

Drum Controller Fonksiyon Bloğu %DRi Çalışması

Açıklama: Tambur Denetleyici şunları içerir:

Aşağıdaki tablodaki örnekte tambur denetleyicinin ana karakteristikleri özetlenmiştir.

- Sekiz adımda (0 ile 7 arası) ve 0 ile F arası numaralandırılmış 16 veri biti şeklinde düzenlenmiş bir sabit veri matrisi (kamlar).
- Bir yapılandırılmış çıkış (%Qi.j.k) veya bellek biti (%Mi) ile ilişkilendirilmiş kontrol bitleri listesi. Mevcut adım sırasında, kontrol bitleri bu adım için tanımlanmış ikilik durumları alırlar.

Aşağıdaki tablodaki bu örnek tambur denetleyicinin ana karakteristiklerini özetlemektedir.

Tablo-48 %DR Fonksiyon Bloğu çalışma örneği.

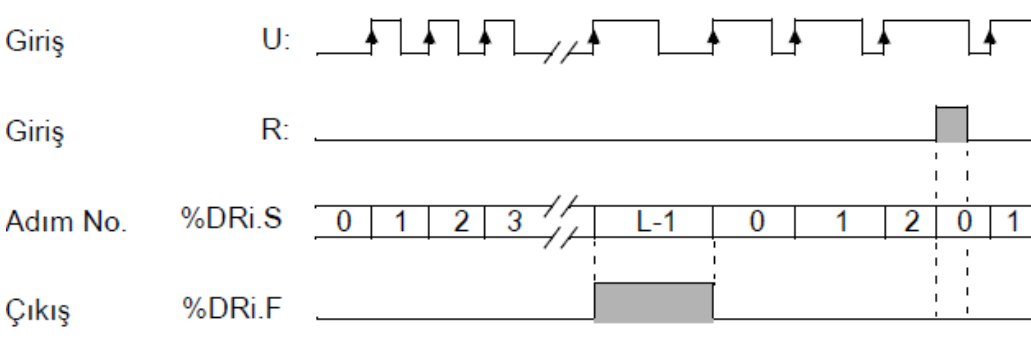
| Sütun | 0 | 1 | 2 | ... | D | E | F |
|-----------------|-------|-------|-------|-----|-------|-------|-------|
| Kontrol Bitleri | %Q0.1 | %Q0.3 | %Q1.5 | | %Q0.6 | %Q0.5 | %Q1.0 |
| Adım 0 | 0 | 0 | 1 | | 1 | 1 | 0 |
| Adım 1 | 1 | 0 | 1 | | 1 | 0 | 0 |
| ⋮ | | | | | | | |
| Adım 5 | 1 | 1 | 1 | | 0 | 0 | 0 |
| Adım 6 | 0 | 1 | 1 | | 0 | 1 | 0 |
| Adım 7 | 1 | 1 | 1 | | 1 | 0 | 0 |

Çalışma:

Yukarıdaki örnekte, adım 5 mevcut adımdır, kontrol bitleri %Q0.1, %Q0.3 ve %Q1.5 “1”e kurulmuştur; kontrol bitleri %Q0.6, %Q0.5 ve %Q1.0 “0” durumundadır. U girişindeki her yükselen kenarda (veya U buyruğunun her etkinleştirilmesinde) mevcut adım no.su bir artırılır. Mevcut adım program tarafından değiştirilebilir.

Zamanlama Diyagramı

Aşağıdaki diyagram tambur denetleyicinin çalışmasını göstermektedir.



Şekil-70 %DR fonksiyon bloğunun zamanlama diyagramı.

Özel Durumlar

Tablo-49’da tambur denetleyici bloğunun programlanmasındaki özel durumlar listelenmiştir.

Tablo-49 %DR fonksiyon bloğunun programlanması için özel durumlar.

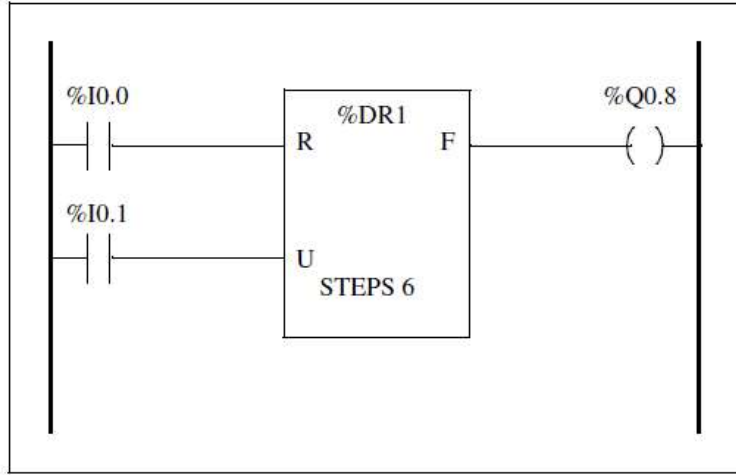
| Özel Durum | Tanım |
|--|--|
| Soğuk yeniden başlatmanın etkisi (%S0=1) | Tambur denetleyiciyi adım 0’a döndürür. |
| Sıcak yeniden başlatmanın etkisi (%S1=1) | Mevcut adımdan sonra kontrol bitlerini günceller. |
| Program dallanmasının etkisi | Tambur denetleyicinin artık taranmadığı gerçeği kontrol bitlerinin sıfırlanmadığı anlamına gelir. |
| Kontrol bitlerinin güncellenmesi | Sadece adımda bir değişiklik olduğunda veya sıcak veya soğuk başlatma durumunda gerçekleşir. |
| MCS/MCR master kontrol rölesi buyruklarının etkisi | MCS/MCR buyruklarının arasında tambur denetleyicinin bulunması, MSC buyruğundan önceki Boole işlem sonucu 0 ise, kontrol bitlerinin sıfırlanması anlamına gelir. |

Programlanma ve Yapılandırma

Açıklama: Aşağıdaki örnek tambur denetleyicinin programlama ve yapılandırmasını göstermektedir. %Q0.0 ile %Q0.5 arası ilk altı çıkış, %I0.1 girişi her seferinde “1”e kurulduğunda, sırayla etkinleştirilir. %I0.0 girişi çıkışları “0”lar.

Programlama Örneği

Aşağıdaki gösterimde tambur denetleyici fonksiyon bloğunun dönüştürülebilir ve dönüştürülemez programlama örnekleri bulunmaktadır.



Ladder diyagramı

| | | | |
|---------|-------|----|--------|
| BLK | %DR1 | LD | %I0.0 |
| LD | %I0.0 | R | %DR1 |
| R | %DR1 | LD | %I0.1 |
| LD | %I0.1 | U | %DR1 |
| U | %DR1 | LD | %DR1.F |
| OUT_BLK | %DR1 | ST | %Q0.8 |
| LD | F | | |
| ST | %Q0.8 | | |
| END_BLK | | | |

Dönüştürülebilir program Dönüştürülemez program

Şekil-71 %DR fonksiyon bloğunun programlama örneği..

Yapılandırma

Aşağıdaki bilgi yapılandırma aşamasında tanımlanır:

- Adım sayısı: 6
- Her tambur denetleyici adımı için çıkış durumları (kontrol bitleri).

Tablo-50 %DR fonksiyon bloğunun yapılandırılması.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Adım 1: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Adım 2: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Adım 3: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Adım 4: | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Adım 5: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Adım 6: | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Kontrol bitlerinin ataması.

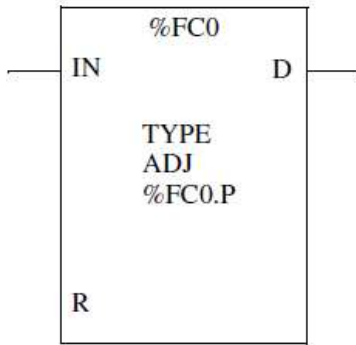
| | | | |
|----|-------|----|-------|
| 1: | %Q0.0 | 4: | %Q0.1 |
| 2: | %Q0.2 | 5: | %Q0.3 |
| 3: | %Q0.4 | 6: | %Q0.5 |

3.7.6 Hızlı Sayıcı Fonksiyon Bloğu (Fast Counter Function Block), %FC

Açıklama: Hızlı sayıcı fonksiyon bloğu (%FC) yukarı sayıcı veya aşağı olarak görev yapabilir. Dijital girişlerin yükselen kenarını 5kHz'e kadar frekanslarda sayabilir. Hızlı Sayıcılar, özel donanım kesmeleri tarafından idare edildiğinden maksimum frekans, uygulama ve donanım yapılandırmasına göre değişiklik gösterebilir.

Kompakt denetleyiciler maksimum üç adet hızlı sayıcı kullanmak üzere yapılandırılabilirken Modüler denetleyicilerde maksimum iki adet kullanılabilir. Hızlı Sayıcı fonksiyon blokları %FC0, %FC1 ve %FC2 sırasıyla %I0.0.2, %I0.0.3 ve %I0.0.4 adanmış çıkışlarını kullanırlar. Bu bitler harici kullanım için ayrılmamıştır. Bu adanmış kaynaklar için yapılan atamalar diğer fonksiyon bloklarının kullanımı ile birlikte düşünülmelidir.

Gösterim



Şekil-72 Hızlı Sayıcı fonksiyon bloğunun gösterimi.

Parametreler

Aşağıdaki tablo Hızlı Sayıcı fonksiyon bloğunun parametrelerini içermektedir:

Tablo-51 Hızlı Sayıcı Fonksiyon Bloğu parametreleri

| Parametre | Etiket | Tanım |
|-------------------|--------|--|
| Fonksiyon | TYPE | Yapılandırma sırasında tanımlanır, sadece yukarı yada aşağı sayıcı olarak ayarlanabilir. |
| Önayar değeri | %FCi.P | 1 ile 65535 arasında ayarlanabilen başlangıç değeri |
| Ayarlanabilir | Y/N | Y (Evet) seçilirse, %FCi.P önayar veya %FCi.V değerini Operatör Göstergesi veya Animasyon Tablosu Editörü aracılığıyla değiştirmek mümkündür. N (Hayır) önayar değeri erişimi olmadığı anlamına gelir. |
| Mevcut değer | %FCi.V | Mevcut değer yukarı veya aşağı sayıcı fonksiyonunun seçimine göre artırılır veya azaltılır. Yukarı sayma için mevcut değer sıfırlanır ve 65536'ya kadar sayar. Aşağı sayma için mevcut değer %FCi.P önayar değerine atanır ve aşağıya doğru 0'a kadar sayar. |
| İzin verme girişi | IN | 1 durumunda, mevcut değer fiziksel girişe uygulanan vurulara göre güncellenir. 0 durumunda, mevcut değer son değerinde tutulur. |
| Reset | %FCi.R | Bloğu başlangıç değerlerine getirmek için kullanılır. 1 durumunda, yukarı sayıcı olarak yapılandırılırsa mevcut değer "0"lanır; aşağı sayıcı olarak yapılandırılırsa %FCi.P değerine atanır. |

| | | |
|----------------------|--------|--|
| | | İşlem tamam biti %FCi.D önceden tanımlanan değerine geri döndürülür. |
| “İşlem tamam” çıkışı | %FCi.D | Yukarı sayıcı olarak yapılandırılırsa, %FCi.V %FCi.P değerine; aşağı sayıcı olarak yapılandırılırsa, %FCi.V 0’a ulaştığında, bu bit 1’e kurur. Sadece okunabilir bu bit yalnızca %FCi.R’yi “1” yaparak resetlenir. |

Özel Not: Ayarlanabilir olarak yapılandırılırsa, o zaman uygulama herhangi bir zaman %FCi.P önayar değerini ve mevcut değer %FCi.V’yi değiştirebilir. Ancak reset girişi etkin olduğunda veya %FCi.D çıkışındaki bir yükselen kenarda yeni bir değer dikkate alınır. Bu da tek bir vuru dahi kaybetmeden ardışık farklı sayımların yapılmasına izin verir.

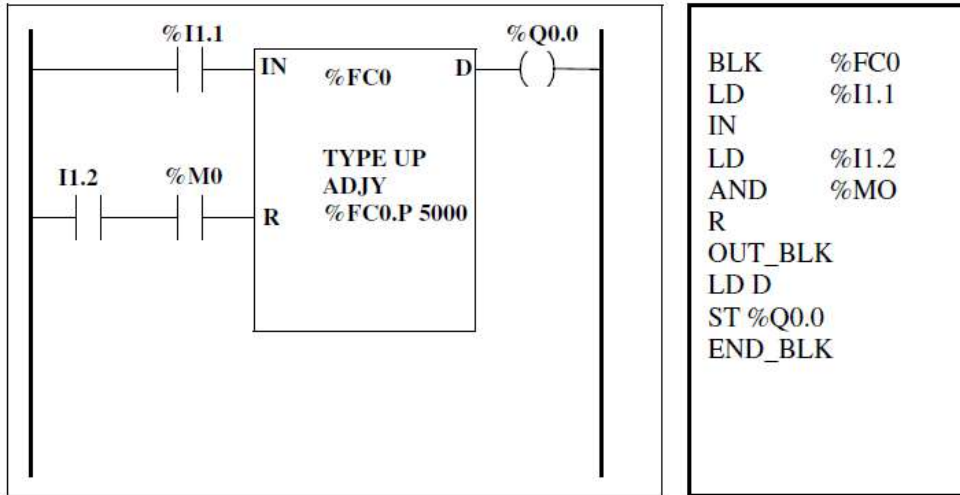
Çalışma

Yukarı sayıcı olarak yapılandırıldığında, atanmış girişte bir yükselen kenar algılandığında, mevcut değer bir artırılır. Önayar değeri %FCi.P’ye ulaşıldığında, işlem tamam çıkış biti %FCi.D “1”e kurur ve %FCi.V mevcut değerine sıfır yüklenir.

Aşağı sayıcı olarak yapılandırılırsa, atanmış girişte bir yükselen kenar algılandığında, mevcut değer bir azaltılır. Değer sıfır olduğunda, Tamam çıkış biti %FCi.D “1”e kurur ve %FCi.P önayar değeri mevcut değere yüklenir.

Yapılandırma ve Programlama

Bu örnekte, %I1.1 “1” olduğunda uygulama 5000’e kadar öğeleri saymaktadır. %FC0 için olan giriş atanmış %I0.0.2 girişidir. Önayar değerine ulaşıldığında, %FC0.D “1”e kurur ve %FC0.R biti %I1.2 ve %M0’ın “AND” işlem sonucu tarafından kontrol edilene kadar aynı değeri korur.



Şekil-73 Hızlı Sayıcı fonksiyon bloğunun programlanması.

Özel Durumlar

Tablo-52’de hızlı sayıcı bloğunun programlanmasındaki özel durumlar listelenmiştir.

Tablo-52 %FC fonksiyon bloğunun programlanması için özel durumlar.

| Özel Durum | Tanım |
|--|---|
| Soğuk yeniden başlatmanın etkisi (%S0=1) | Tüm %FC özelliklerini kullanıcı veya kullanıcı uygulaması tarafından yapılandırılmış değerlere geri döndürür. |
| Sıcak yeniden başlatmanın etkisi (%S1=1) | Hiçbir etkisi yoktur. |
| Denetleyicinin durmasının etkisi | %FC denetleyicinin durdurulduğu anda izinlenen parametre ayaalarıyla saymaya devam eder. |

3.8 Saat Fonksiyonları

Açıklama Twido denetleyiciler, Gerçek-Zaman Saati opsiyonu (GZS) gerektiren bir gün zamanı saati fonksiyonuna sahiptir ve aşağıdakileri sağlar:

- Takvim blokları, önceden tanımlı zamanlarda olayları kontrol etmek için kullanılır.
- Zaman/tarih damgalama zaman ve tarihleri, olaylara atamak ve olay Twido gün zamanı saatine, TwidoSoft Software menüsündeki Schedule Blocks (Takvim Blokları) seçilerek erişilebilir. Gün zamanı saati, bir program tarafından da ayarlanabilir. Eğer batarya, denetleyici kapatılmadan önce, en az altı saat boyunca şarj edilmiş ise, denetleyici kapatıldığında, saat ayarları 30 güne kadar çalışmaya devam eder. Gün zamanı saati, 24-saat formatındadır ve artık yılları da hesaba katar.

3.8.1 GZS Düzeltmesi Değeri

GZS Düzeltmesi değeri, GZS'nin doğru çalışması için gereklidir. Her bir GZS ünitesi, ünite üzerine yazılmış kendi düzeltme değerine sahiptir. Bu değer, TwidoSoft'ta Controller Operation diyalog kutusundaki Configure RTC (GZS'yi Konfigüre Et) opsiyonu kullanılarak yapılandırılabilir.

3.8.2 Takvim Blokları

Açıklama Takvim Blokları, önceden tanımlı bir ay, gün ve zamandaki aksiyonları kontrol etmek için kullanılır. Maksimum 16 takvim bloğu kullanılabilir ve herhangi bir program girişine ihtiyaç duymaz.

Not: Sistem biti %S51'i, Gerçek-Zaman Saati (GZS) opsiyonunun yerleştirilmiş olduğunu doğrulamak için kontrol ediniz. Takvim bloklarının kullanılması için GZS opsiyonu gereklidir.

Parametreler Aşağıdaki tablo, bir takvim bloğu için parametreleri listeler.

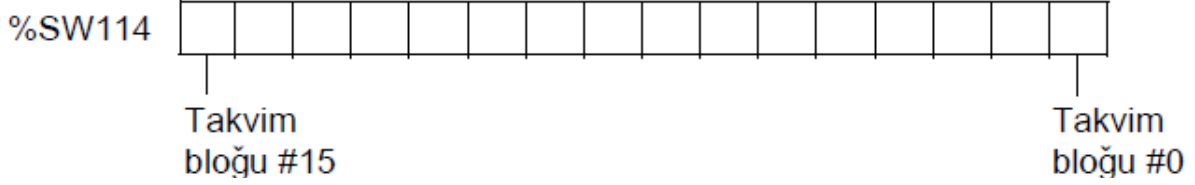
Tablo-53 Takvim bloğu parametreleri

| Parametre | Format | Fonksiyon/Aralık |
|-----------------------|-----------------|--|
| Takvim bloğu numarası | n | n = 0 ila 15 |
| Konfigüre edilmiş | Check kutusu | Seçilen takvim bloğu numarasını konfigüre etmek için bu kutuya check işareti koyunuz. |
| Çıkış biti | %Qx.y.z | Çıkış ataması, takvim bloğu tarafından aktive edilir: %Mi ve %Qj.k . Bu çıkış, güncel tarih ve zaman, aktif periyot başlangıcı ayarı ve aktif periyot sonu ayarı arasında olduğunda 1'e set edilir. |
| Başlangıç ayı | Ocak ila Aralık | Takvim bloğunun başlatılacağı ay. |
| Bitim ayı | Ocak ila Aralık | Takvim bloğunun sonlandırılacağı ay. |
| Başlangıç tarihi | 1-31 | Takvim bloğunun başlatılacağı ayın günü. |
| Bitim tarihi | 1-31 | Takvim bloğunun sonlandırılacağı ayın günü. |
| Başlangıç zamanı | hh:mm | Takvim bloğunun başlatılacağı gün zamanı; saatler (0 ila 23) ve dakikalar (0 ila 59). |
| Durma zamanı | hh:mm | Takvim bloğunun sonlandırılacağı gün zamanı; saatler (0 ila 23) ve dakikalar (0 ila 59). |
| Haftanın günü | Pazartesi-Pazar | Takvim bloğunun aktivasyonu için haftanın günü tanımlayan check kutuları. |

Takvim Bloklarını İzinlemek

Sistem sözcüğü %SW114'ün bitleri 16 takvim bloğunun her birinin çalışmasını izinler (bit, 1'e set edilir) veya iptal eder (bit 0'a set edilir).

%SW114'deki takvim bloklarının atanması:



Default olarak (veya bir soğuk yeniden başlatmadan sonra) bir sistem sözcüğünün tüm bitleri, 1'e set edilir. Program tarafından bu bitlerin kullanımı opsiyoneldir.

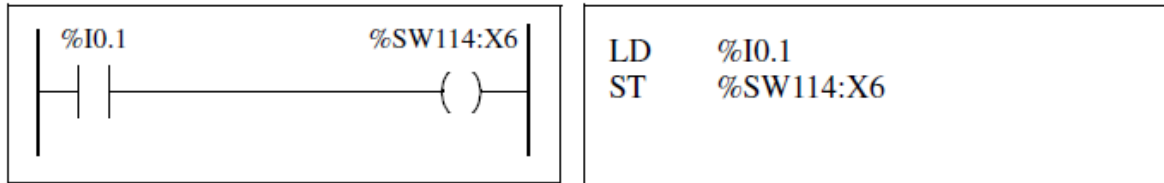
Takvim Bloklarının Çıkışı

Aynı çıkış (%Mi veya %Qj.k) birkaç blok tarafından atanmışsa, bu nesneye son olarak atanan, bu blokların herbirinin sonucunun "OR'lanmıştır" (aynı çıkış için birkaç "çalışma aralığı" olması mümkündür).

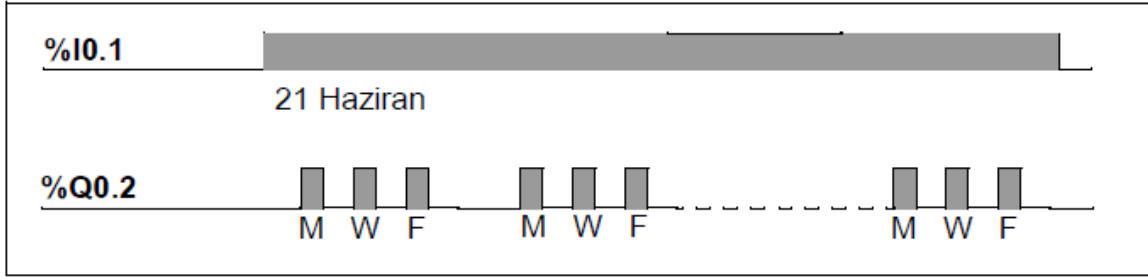
Örnek Aşağıdaki tablo, bir yaz ayı sulama programı örneği için parametreleri göstermektedir:

| Parametre | Değer | Tanım |
|------------------|---------------------------|---|
| Takvim bloğu | 6 | Takvim bloğu numarası 6 |
| Çıkış biti | %Qx.y.z | Çıkış %Qx.y.z'yi aktive et |
| Başlangıç ayı | June (Haziran) | Aktiviteyi, Haziran'da başlat |
| Bitim ayı | September (Eylül) | Aktiviteyi, Eylül'de durdur |
| Başlangıç tarihi | 21 | Aktiviteyi, Haziran'ın 21. gününde başlat |
| Bitim tarihi | 21 | Aktiviteyi, Eylül'ün 21. gününde durdur. |
| Haftanın günü | Monday, Wednesday, Friday | Aktiviteyi, Pazartesi, Çarşamba ve Cuma günleri çalıştır (run). |
| Başlatma zamanı | 21:00 | Aktiviteyi, 21:00'da başlat |
| Durma zamanı | 22:00 | Aktiviteyi, 22:00'da durdur |

Takvim bloğu, aşağıdaki program kullanılarak, bir anahtar veya %I0.1 girişine bağlı bir nem dedektörü aracılığıyla iptal edilebilir.



Aşağıdaki zamanlama diyagramı, %Q0.2 çıkışının aktivasyonunu gösterir.



3.8.3 Program aracılığıyla Zaman Tarihleme

Tarih ve zamanın her ikisi birden, sistem sözcükleri %SW50 ile %SW53'te mevcuttur. Bu yüzden, güncel tarih ve zaman ile anlık değerler veya ayar noktalarını (setpoints) içerebilen sözcükler %MWi (veya %KWi) arasında aritmetik karşılaştırmalar yaparak, denetleyici programı içinde zaman ve tarih damgalama gerçekleştirmek mümkündür.

Zaman/Tarih Damgalama

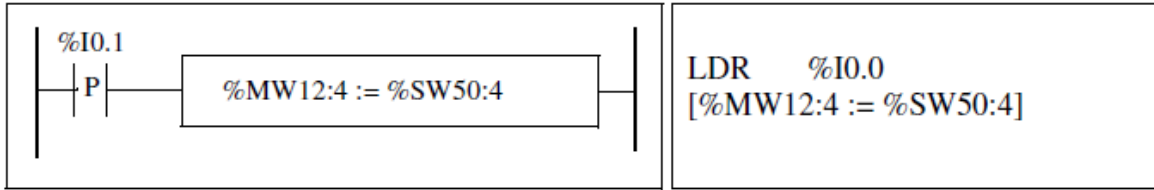
Açıklama Sistem sözcükleri %SW50 ile %SW53; bir çevresel (peripheral) cihaz üzerinde görüntüleme için veya cihaza iletim için yararlı olan güncel tarih ve zamanı BCD formatında içerir. Bu sistem sözcükleri, bir olayın zaman ve tarihini saklamak için kullanılabilir.

Not: Tarih ve zaman, opsiyonel Operatör Ekranı kullanılarak da ayarlanabilir.

Bir Olayı Tarihlemek Bir olayı tarihlemek için; atama işlemleri kullanmak, sistem sözcüklerinin içeriğini dahili sözcüklere transfer etmek ve daha sonra bu dahili sözcükleri işlemek yeterlidir (örneğin, birimi görüntülemek için EXCH komutu aracılığıyla iletim).

Programlama Örneği

Aşağıdaki örnek, %I0.1 girişi üzerindeki bir yükselen kenarın nasıl tarihleneceğini gösterir.



Bir olay saptandığında, sözcük tablosu aşağıdakileri içerir:

| Kodlama | En anlamlı byte | En az anlamlı byte |
|---------|-----------------|--------------------|
| %MW12 | Saniye | Haftanın günü (1) |
| %MW13 | Saat | Dakika |
| %MW14 | Ay | Gün |
| %MW15 | Yüzyıl | Yıl |

Not: (1) 0 = Pazartesi, 1 = Salı, 2 = Çarşamba, 3 = Perşembe, 4 = Cuma, 5 = Cumartesi, 6 = Pazar.

Sözcük Tablosu Örneği

19 Nisan 2002, Pazartesi, 13:40:30 için örnek veri:

| Word | Değer (hex) | Anlam |
|-------|-------------|---------------------------|
| %MW12 | 3000 | 30 saniye, 00 = Pazartesi |
| %MW13 | 1340 | 13 saat, 40 dakika |
| %MW14 | 0419 | 04 = Nisan, Ayın 19'u |
| %MW15 | 2002 | 2002 |

Son Durmanın Tarih ve Zamanı

Sistem sözcükleri %SW54 ila %SW57, son durmanın tarih ve zamanını içerir ve %SW58 sözcüğü, son durmanın nedenini BCD formatında gösteren kodu içerir.

3.8.4 Tarih ve Zamanı Ayarlamak

Açıklama Tarih ve zaman ayarları, aşağıdaki yöntemlerden biri kullanılarak güncellenebilir:

TwidoSoft

Set Time (Zamanı Ayarla) diyalog kutusunu kullanınız. Bu diyalog, **Controller** menüsünden **Controller Operations** (Kontrolör İşlemleri) seçilerek görüntülenen **Controller Operations** diyalog kutusunda mevcuttur.

%SW50 ila %SW53 sistem sözcüklerini veya sistem sözcüğü %SW59'u kullanınız.

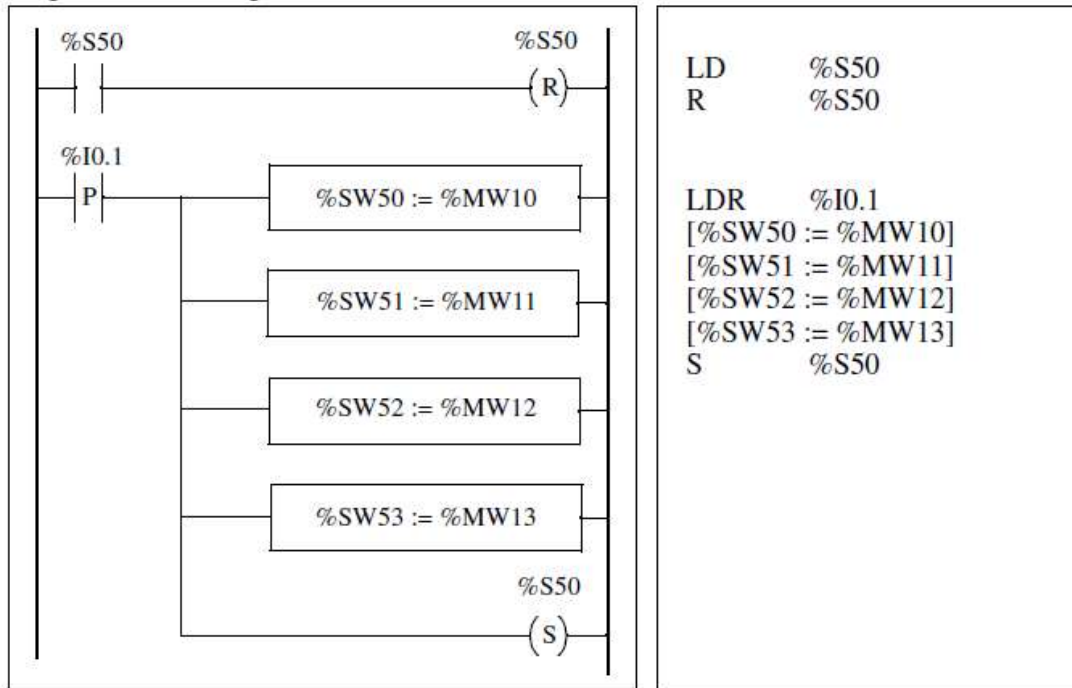
Tarih ve zaman ayarları, yalnızca denetleyici üzerinde GZS opsiyon kartuşu (TWDXCPRTC) kurulu olduğunda güncellenebilir.

%SW50 ila %SW53'ü Kullanmak

Sistem sözcükleri %SW50 ila %SW53'ü, tarih ve zamanı ayarlamakta kullanmak için, %S50 biti, 1'e kurulmalıdır. Bu işlem, aşağıdakilere yol açar:

- %SW50 ila %SW53 sözcüklerinin, dahili saat aracılığıyla güncellenmesini iptal eder.
- %SW50 ila %SW53 sözcüklerine yazılmış değerleri, dahili saate iletir.

Programlama örneği:



Sözcükler %MW10 ila %MW13; yeni tarih ve zamanı, BCD formatında içerecektir ve sözcükler %SW50 ila %SW53'ün kodlamasına karşılık gelecektir.

Sözcük tablosu, yeni tarih ve zamanı içermek zorundadır:

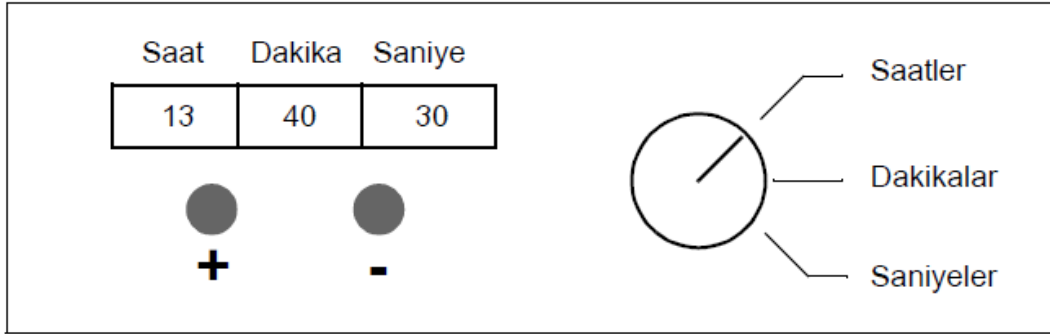
| Kodlama | En anlamlı byte | En az anlamlı byte |
|---------|-----------------|--------------------|
| %MW10 | Saniye | Haftanın günü (1) |
| %MW11 | Saat | Dakika |
| %MW12 | Ay | Gün |
| %MW13 | Yüzyıl | Yıl |

%SW59'u Kullanmak Tarih ve zamanı güncellemek için bir başka yöntem, sistem biti %S59'u ve tarih ayarlama sistem sözcüğü %SW59'u kullanmaktır.

%S59 bitini 1 yapmak, mevcut tarih ve zamanı, %SW59 sözcüğü aracılığıyla ayarlamaya olanak sağlar. %SW59, tarih ve zaman bileşenlerinin herbirini, bir yükselen kenarda artırır veya azaltır.

Uygulama Örneği

Aşağıdaki ön panel, dahili saatin (clock) saatini (hour), dakikalarını ve saniyelerini değiştirmek için yaratılmıştır.



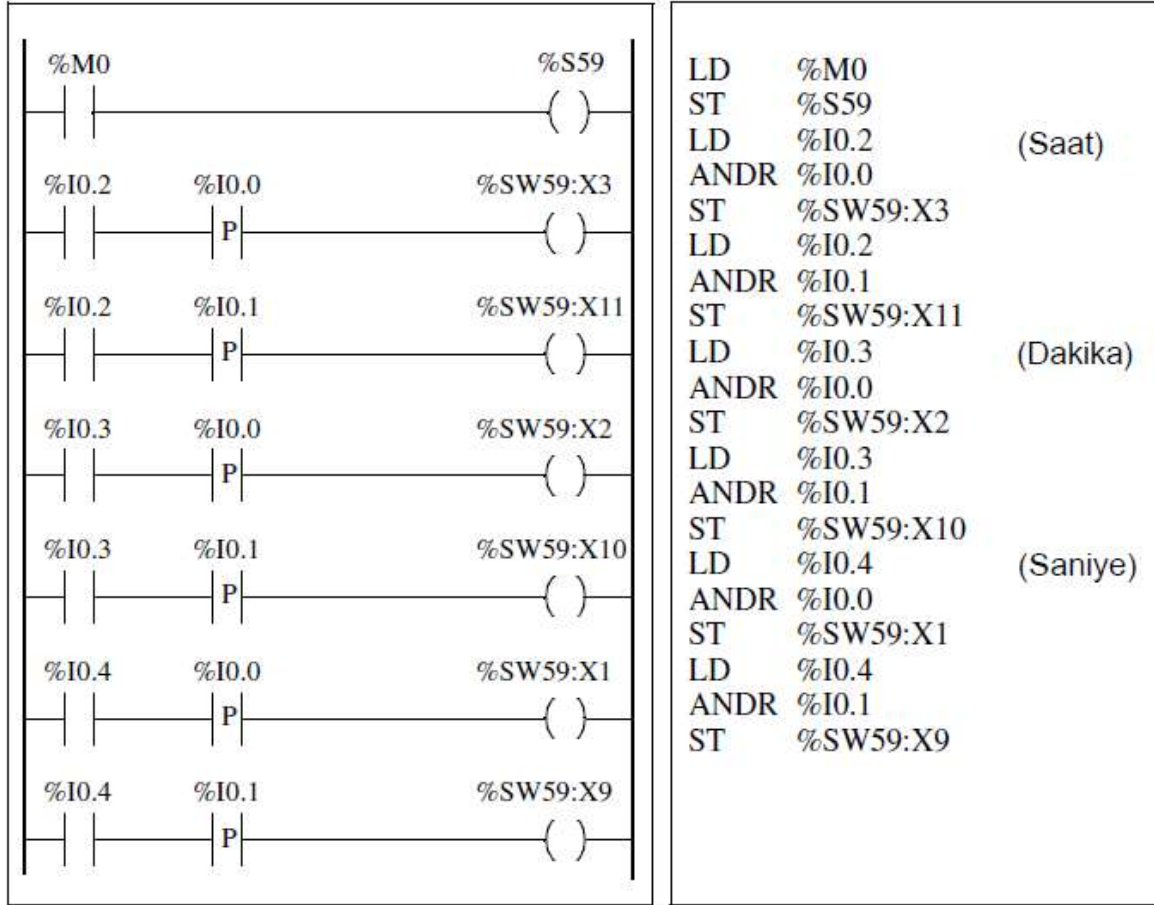
Kontrollerin tanımı:

Saatler/Dakikalar/Saniyeler anahtarı, sırasıyla %I0.2, %I0.3 ve %I0.4 girişlerini kullanarak, değiştirilecek zaman göstergesini seçer.

Push buton "+"; seçilen zaman göstergesini, %I0.0 girişini kullanarak artırır.

Push buton "-"; seçilen zaman göstergesini, %I0.1 girişini kullanarak azaltır.

Aşağıdaki program, panelden girişleri okur ve dahili saati ayarlar.



3.9 Dahili Analog Fonksiyonlar

3.9.1 Potansiyometreler

Açıklama Twido denetleyicilerin potansiyometre sayıları:

- TWDLCAA16DRF ve TWDLMDA20DRT denetleyicilerde bir potansiyometre
- TWDLCAA24DRF ve TWDLCAE40DRF denetleyicilerde iki potansiyometre

Programlama Potansiyometre 1 ve potansiyometre 2 tarafından sağlanan analog değerlere karşılık gelen, Potansiyometre 1 için, 0'dan 1023'e kadar, potansiyometre 2 için ise 0'dan 511'e kadar olan nümerik değerler, aşağıdaki iki sistem sözcüğü içinde tutulur:

- %IW0.0.0 potansiyometre 1 için (en soldaki)
- %IW0.0.1 potansiyometre 2 için (en sağdaki)

Bu sözcükler, aritmetik işlemlerde kullanılabilir. Bunlar, her tip ayarlama için; örneğin, bir zaman-gecikmesi veya sayıcı ön-ayarlaması, darbe üretici frekansının veya makine ön ısıtma süresinin ayarlanması için kullanılabilir.

3.9.2 Analog Kanal

Açıklama Tüm Modüler denetleyiciler (TWDLMDA20DTK, TWDLMDA20DUK, TWDLMDA20DRT, TWDLMD40DTK ve TWDLMD40DUK), bir dahili analog kanala sahiptir. Gerilim girişi aralığı, 0'dan 10V'a ve sayısallaştırılmış sinyal ise 0'dan 511'e kadardır. Analog kanal, sekiz örnek üzerinde ortaya çıkan basit bir ortalama alma düzeninden yararlanır.

İlke Bir analog/dijital çevirici, 0 ila 10 V'luk bir gerilim değerini, 0 ile 511 arasında bir dijital değere çevirir. Bu değer, sistem sözcüğü %IW0.0.1'de saklanır. Bu değer, tüm aralık boyunca doğrusaldır,

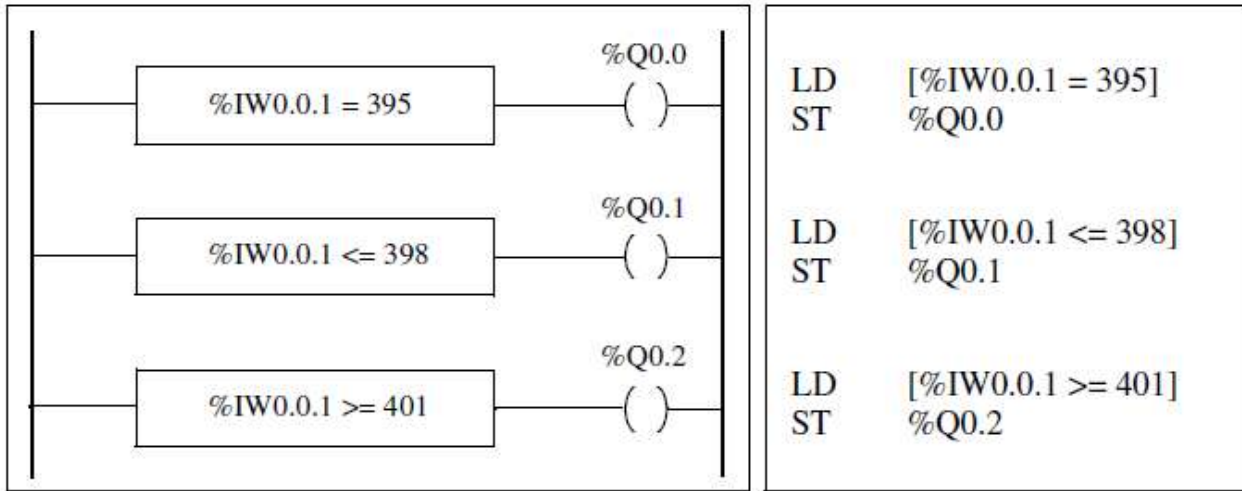
böylece her bir sayım yaklaşık olarak 20 mV'tur (10V/512). Giriş sinyalinin maksimum değerinin aşılp aşılmadığını saptamak için, okunan değer olarak 511 kullanılır.

Programlama Örneği

Bir fırının sıcaklığının kontrol edilmesi: Pişirme sıcaklığı, 350°C'ye ayarlanmıştır. $\pm 2.5^\circ\text{C}$ 'lik bir değişim, çıkışlar %Q0.1 ve %Q0.2'nin sırasıyla çekmesine neden olmaktadır. Pratik olarak, bu örnekte, analog kanalın 0'dan 511'e kadar olan olası ayarlama aralığının tamamı kullanılmıştır. Sıcaklık ayar noktaları için analog ayarlar:

| Sıcaklık (°C) | Gerilim | Sistem Word'ü %IW0.0.1 |
|---------------|---------|------------------------|
| 0 | 0 | 0 |
| 347.5 | 7.72 | 395 |
| 350 | 7.77 | 398 |
| 352.5 | 7.83 | 401 |
| 450 | 10 | 511 |

Yukarıdaki örneğe ait program:



3.10 Sistem Objeleri ve Değişkenleri

Aşağıdaki tabloda Operator Göstergesi tarafından gösterilebilen ve değiştirilebilen sistem objeleri ve değişkenleri erişim sıralarına göre listelenmiştir.

Tablo-54 Sistem objeleri ve değişkenleri

| Obje | Değişken/Özellik | Tanım | Erişim |
|--------------------------------------|------------------|------------------------|---------------------|
| Input (Giriş) | %Ix.y.z | Değer | Okuma/Zorlama |
| Output (Çıkış) | %Qx.y.z | Değer | Okuma/Yazma/Zorlama |
| Timer (Zamanlayıcı) | %TMX.V | Mevcut Değer | Okuma/Yazma |
| | %TMX.P | Önayar Değeri | Okuma/Yazma |
| | %TMX.Q | Done (İşlem Tamam) | Okuma |
| Counter (Sayıcı) | %Cx.V | MevcutDeğer | Okuma/Yazma |
| | %Cx.P | ÖnayarDeğeri | Okuma/Yazma |
| | %Cx.D | Done (İşlem Tamam) | Okuma |
| | %Cx.E | Empty (Boş) | Okuma |
| | %Cx.F | Full (Dolu) | Okuma |
| Bellek (Memory) Biti | %Mx | Değer | Okuma/Yazma |
| Bellek Sözcüğü | %MWx | Değer | Okuma/Yazma |
| Sabit (Constant) Sözcük | %KWx | Değer | Okuma |
| Bellek Çift Sözcük | %MDx | Değer | Okuma/Yazma |
| Sabit Çift Sözcük | %K Dx | Değer | Okuma |
| Bellek floating Sözcük | %MFx | Değer | Okuma/Yazma |
| Sabit floating Sözcük | %KFx | Değer | Okuma |
| Sistem Biti | %Sx | Değer | Okuma/Yazma |
| Sistem Sözcüğü | %SWx | Değer | Okuma/Yazma |
| Analog Giriş | %IWx.y.z | Değer | Okuma |
| Analog Çıkış | %QWx.y.z | Değer | Okuma/Yazma |
| Fast Counter (Hızlı Sayıcı) | %FCx.V | Mevcut Değer | Okuma |
| | %FCx.P | Önayar Değeri | Okuma/Yazma |
| | %FCx.D | Done (İşlem Tamam) | Okuma |
| Very Fast Counter (Çok Hızlı Sayıcı) | %VFCx.V | Mevcut Değer | Okuma |
| | %VFCx.P | Önayar Değeri | Okuma/Yazma |
| | %VFCx.U | Sayma Yönü | Okuma |
| | %VFCx.C | Yakalama (Catch)Değeri | Okuma |
| | %VFCx.S0 | | Okuma/Yazma |

| | | | |
|---|--|--|--|
| | %VFCx.S1 %VFCx.F %VFCx.M %VFC.T %VFC.R %VFC.S | Eşik 0 Değeri EşikDeğeri1 Taşma Frekans Tamam Zaman tabanı Reflex Çıkış İzni Reflex Giriş İzni | Okuma/Yazma Okuma Okuma/Yazma Okuma/Yazma Okuma/Yazma Okuma/Yazma |
| Giriş Network Sözcüğü | %INWx.z | Değer | Okuma |
| Çıkış Network Sözcüğü | %QNWx.z | Değer | Okuma/Yazma |
| Grafcet | %Xx | Adım Biti | Okuma |
| Pulse Generator (Vuru Üreteci) | %PLS.N %PLS.P %PLS.D %PLS.Q | Vuru Sayısı ÖnayarDeğeri Done (İşlem Tamam) Mevcut Çıkış | Okuma/Yazma Okuma/Yazma Okuma Okuma |
| Pulse Width Modulator | %PWM.R %PWM.P | Oran Önayar Değeri | Okuma/Yazma Okuma/Yazma |
| Drum Controller (Tambur Denetleyici) | %DRx.S %DRx.F | Mevcut Adım Numarası Full (Dolu) | Okuma Okuma |
| Step counter | %SCx.n | Adım Sayacı biti | Okuma/Yazma |
| Register (Yazaç) | %Rx.I %Rx.O %Rx.E %Rx.F | Input (Giriş) Output (Çıkış) Empty (Boş) Full (Dolu) | Okuma/Yazma Okuma/Yazma Okuma Okuma |
| Shift bit register | %SBR.x.yy | Yazaç Biti | Okuma/Yazma |
| Mesaj | %MSGx.D %MSGx.E | Done (İşlem Tamam) Error (Hata) | Okuma Okuma |
| AS-Interface slave input | %IA.x.y.z | Değer | Okuma/Zorlama |
| AS-Interface analog slave input | %IWA.x.y.z | Değer | Okuma |
| AS-Interface slave output | %QA.x.y.z | Değer | Okuma/Yazma/Zorlama |

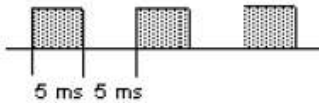
3.11 Sistem Bitleri (%Si)

Giriş

Aşağıdaki bölümde sistem bitlerinin bir genel tanıtımı ve nasıl kontrol edildikleri açıklanmıştır.

Açıklama: Tablo-55’te sistem bitlerinin ayrıntılı tanıtımı ve nasıl kontrol edildikleri açıklanmıştır.

Tablo-55 Bazı sistem bitleri ve tanımları

| Sistem Biti | Fonksiyon | Tanım | Başlangıç değeri | Kontrol |
|-------------|---------------------|--|------------------|------------|
| %S0 | Soğuk Başlama | Normalde 0’dır, “1”e kurulması aşağıdaki koşullarda gerçekleşir: <ul style="list-style-type: none">• Veri kaybıyla gücün yeniden gelmesi (pil hatası).• Kullanıcı programı veya Animasyon Tablosu Editörü• İşlemler Göstergesi. İlk tam tarama süresince bu bit “1”e kuruludur. Sonraki taramadan önce sistem tarafından sıfırlanır. | 0 | S veya U→S |
| %S1 | Sıcak Başlama | Normalde 0’dır, “1”e kurulması aşağıdaki koşullarda gerçekleşir: <ul style="list-style-type: none">• Veri yedeklemesiyle gücün yeniden gelmesi.• Kullanıcı programı veya Animasyon Tablosu Editörü• İşlemler Göstergesi. Tam tarama sonunda sistem tarafından sıfırlanır. | 0 | S veya U→S |
| %S4 | Zaman tabanı: 10ms | <p>Durum değiştirme hızı dahili saat tarafından ölçülür. Denetleyicinin taraması ile eşzamanlı değerlerdir.</p>  <p>Örnek: %S4</p> | - | S |
| %S5 | Zaman tabanı: 100ms | | | |
| %S6 | Zaman tabanı: 1s | | | |
| %S7 | Zaman tabanı: 1min | | | |
| %S8 | Bağlantı testi | Başlangıçta “1”e kurulu olan bu bit, denetleyici “yapılandırılmamış” durumunda olduğunda, bağlantıları test etmek için kullanılır: <ul style="list-style-type: none">• “1”e kurulduğunda çıkış resetlenir.• “0”a kurulduğunda bağlantı testi yetkilendirilir. | 1 | U |
| %S9 | Reset çıkışları | Normalde 0’dır. Program tarafından veya terminal aracılığıyla (Animasyon Tablosu Editörü) “1”e kurulur: <ul style="list-style-type: none">• “1” durumunda, denetleyici “RUN” modunda olduğunda çıkışlar “0” olmaya zorlanır.• “0” durumunda, çıkışlar normal olarak güncellenir. | 0 | U |
| %S10 | G/Ç hatası | Normalde “1”dir. Bu bir G/Ç hatası algılandığında sistem tarafından 0’a kurulabilir. | 1 | S |
| %S11 | Watchdog taşması | Normalde 0’dır. Program işletim süresi (tarama süresi) maksimum tarama süresini aştığında (yazılım watchdog) “1”e kurulur. Watchdog taşması denetleyicinin HALT’a (durmak) geçmesine neden olur. | 0 | S |
| %S12 | PLC RUN modunda | Bu bit denetleyicinin çalışma durumunu yansıtır. Denetleyici çalıştığında sistem bu biti “1”e kurar. Veya durdurma, init veya diğer bir durum için “0”lar. | 0 | S |

| | | | | |
|------|--|--|---|-----|
| %S13 | RUN'da ilk tarama | Normalde 0'dır. Bu bit denetleyici RUN moduna alındıktan sonraki ilk tarama süresince sistem tarafından "1"e kurulur. | 1 | S |
| %S17 | Kapasite aşıldı | Normalde 0'dır, sistem tarafından "1"e kurulur: <ul style="list-style-type: none"> • İşaretsiz aritmetik işlemi (kalan) sırasında elde taşması durumunda. • Döndürme veya kaydırma işlemi sırasında. Sistem çıkışı "1" durumundaki bit ile gösterir. Taşma riski olduğunda her işlemin arkasından kullanıcı programı tarafından test edilmeli ve sonra, taşma oluşursa kullanıcı tarafından sıfırlanmalıdır. | 0 | S→U |
| %S18 | Aritmetik taşma veya hata | Normalde 0'dır. Şu durumlarda 16 bitlik işlem gerçekleştirildiğinde sistem tarafından "1"e kurulur: <ul style="list-style-type: none"> • Tek uzunluktaki sözcükte, +32 767'den büyük veya - 32 768'den küçük sonuç. • Çift uzunluktaki sözcükte, +2 147 483 647'den büyük veya - 2 147 483 648'den küçük sonuç. • Gerçel sayı noktası kullandığında, + 3,402824E+38'den büyük veya - 3,402824E+38'den küçük sonuç. • 0 ile bölme. • Negatif sayının karekökü. • BTI veya ITB dönüşümü anlamlı değil: BCD değeri sınırların dışında. Taşma riski olduğunda her işlemin arkasından kullanıcı programı tarafından test edilmeli ve sonra, taşma oluşursa kullanıcı tarafından sıfırlanmalıdır. | 0 | S→U |
| %S19 | Tarama periyodu aşımı (periyodik tarama) | Normalde 0'dır, bu bit tarama periyodu aşımı (kullanıcı tarafından yapılandırılma sırasında tanımlanmış veya %SWO'da programlanmış periyottan daha uzun tarama süresi) durumunda sistem tarafından "1"e kurulur. Bu bit kullanıcı tarafından "0"lanır. | 0 | S→U |
| %S20 | İndeks taşması | Normalde 0'dır, bu bit indekslenmiş objenin adresi 0'dan küçük olduğunda veya objenin maksimum boyutu aşıldığında "1"e kurulur. Taşma riski olduğunda her işlemin arkasından kullanıcı programı tarafından test edilmeli ve taşma oluşursa kullanıcı tarafından sıfırlanmalıdır. | 0 | S→U |
| %S24 | Operatör Ekranı | Normalde 0'dır, bu bit kullanıcı tarafından "1"e kurulabilir. <ul style="list-style-type: none"> • 0 durumunda, Operatör Ekranı normal çalışıyor. • 1 durumunda, Operatör Ekranı "dondurulmuş"tur, mevcut göstergede kalır, yanıp sönme devre dışı, ve giriş anahtarını işlemi durdurulmuş. | 0 | U→S |
| %S31 | Olay Maskeleye | Normalde 0'dır, bu bit kullanıcı tarafından "1"e kurulabilir. <ul style="list-style-type: none"> • "0"a kurulduğunda, olaylar işletilebilir. • "1"e kurulduğunda, olaylar işletilemez ve sıraya atılır. Soğuk yeniden başlatmada kullanıcı ve sistem tarafından 0'lanabilir. | 0 | U→S |
| %S38 | Olayların olay sırasına yerleştirilme izni | Normalde 0'dır, bu bit kullanıcı tarafından "1"e kurulabilir. <ul style="list-style-type: none"> • "0"a kurulu: algılanır algılanmaz olaylar işletilebilir. • "1"e kurulu: olaylar, olaylar sırasına konulamaz. Soğuk yeniden başlatmada kullanıcı ve sistem tarafından 0'lanabilir. | 0 | U→S |

| | | | | |
|-------|---|---|---|-----|
| %S39 | Olaylar sırasının doyması | Normalde 0'dır, bu bit kullanıcı tarafından "1"e kurulabilir. <ul style="list-style-type: none"> • "0"a kurulu, tüm olaylar raporlanır. • En azından bir olay kaybolursa "1"e kurulur. Soğuk yeniden başlatmada kullanıcı ve sistem tarafından 0'lanabilir. | 0 | U→S |
| %S50 | %SW50 ila 53 sözcükleri aracılığıyla tarih ve zamanın güncellenmesi | Normalde 0'dır. Bu bit, program veya Operatör Ekranı tarafından 1 veya 0 yapılabilir. <ul style="list-style-type: none"> • 0 durumunda, tarih ve zaman okunabilir. • 1 durumunda, tarih ve zaman güncellenebilir. | 0 | U→S |
| %S51 | Gün zamanı saatinin durumu | Normalde 0'dır. Bu bit, program veya Operatör Ekranı tarafından 1 veya 0 yapılabilir. <ul style="list-style-type: none"> • 0 durumunda, tarih ve zaman ayarlanır. • 1 durumunda, tarih ve zaman, kullanıcı tarafından ayarlanmalıdır. Bu bit 1'e set edildiğinde, gün zamanı saati, geçerli değildir. Tarih ve zaman daha önce hiç yapılandırılmamış, batarya düşük seviyede veya denetleyici düzeltme sabiti geçersiz olabilir. 1 durumundan 0 durumuna geçiş, GZS'nin düzeltme sabitinin yazılmasını zorlar. | 0 | U→S |
| %S59 | %SW59 sözcüğü kullanılarak tarih ve zamanın güncellenmesi | Normalde 0'dır. Bu bit, program veya Operatör Ekranı tarafından 1 veya 0 yapılabilir. <ul style="list-style-type: none"> • 0 durumunda, tarih ve zaman değişmeden kalır. • 1 durumunda, tarih ve zaman, %SW59'daki kontrol bitlerine göre artırılır veya azaltılır. | 0 | U |
| %S69 | Kullanıcı STAT LED göstergesi | <ul style="list-style-type: none"> • 0 durumunda, STAT LED'i, kapalıdır. • 1 durumunda, STAT LED'i açıktır. | 0 | U |
| %S100 | TwidoSoft iletişim kablosu bağlantısı | TwidoSoft iletişim kablosunun bağlı olup olmadığını gösterir. <ul style="list-style-type: none"> • "1"e kurulu: Ya TwidoSoft iletişim kablosu bağlı değildir ya da TwidoSoft bağlanmıştır. • "0"a kurulu: TwidoSoft Uzaktan Bağlantı kablosu bağlı. | - | S |
| %S118 | Uzak uç G/Ç hatası | Normalde "1"e kurulu. Uzak bağlantıda bir G/Ç hatası algılandığında bu bit "0"a kurulur. | 1 | S |
| %S119 | Yerel G/Ç hatası | Normalde "1"e kurulu. Uzak bağlantıda bir G/Ç hatası algılandığında bu bit "0"a kurulur. %SW118 hatanın doğasını belirler. Hata ortadan kalktığında "1"e kurulur. | 1 | S |

Aşağıdaki tabloda kullanılan kısaltmalar verilmiştir.

Tablo-56 Kısaltmalar tablosu.

| Kısaltma | Tanım |
|----------|--|
| S | Sistem tarafından kontrol edilir. |
| U | Kullanıcı tarafından kontrol edilir. |
| U→S | Kullanıcı tarafından "1"e kurulur, sistem tarafından "0"lanır. |
| S→U | Sistem tarafından "1"e kurulur, kullanıcı tarafından "0"lanır. |

3.12 Sistem Sözcükleri (%SWi)

Giriş: Aşağıdaki bölümde sistem sözcüklerinin işlevi ve nasıl kontrol edildikleri açıklanmıştır.

Ayrıntılı Açıklama: Aşağıda sistem sözcüklerinin ayrıntılı tanıtımı ve nasıl kontrol edildiklerinin açıklandığı bir tablo (Tablo-57) verilmiştir.

Tablo-57 Bazı sistem sözcükleri ve tanımları

| Sistem Sözcüğü | Fonksiyon | Tanım | Kontrol |
|-----------------|---|--|---------|
| %SW0 | Denetleyici Tarama Periyodu (Periyodik Görev) | Yapılandırmada tanımlanan denetleyici tarama periyodunu Animasyon Tablosu Editörü'ndeki kullanıcı programı aracılığıyla değiştirir. | U |
| %SW6 | Denetleyici durumu | Denetleyici Durumu: 0 = NOCONFIG 2 = STOPP 3 = RUN 4 = HALT | S |
| %SW7 | Denetleyici durumu | <ul style="list-style-type: none">• Bit [0]: Yedekleme/yeniden yükleme (Backup/restore) ilerlemede:<ul style="list-style-type: none">○ Yedekleme/yeniden yükleme ilerliyor ise "1"e kurulur.○ Backup/restore tamamlandı veya devre dışı ise "0"a kurulur.• Bit [1]: Denetleyicinin yapılandırması OK:<ul style="list-style-type: none">○ Yapılandırma OK ise "1"e kurulur.• Bit [3..2] EEPROM durum bitleri:<ul style="list-style-type: none">○ 00 = Kartuş yok○ 01 = 32 Kb EEPROM kartuşu○ 10 = 64 Kb EEPROM kartuşu○ 11 = Gelecek kullanım için ayrıldı• Bit [4]: RAM'deki uygulama EEPROM'dakinden farklı:<ul style="list-style-type: none">○ RAM uygulaması EEPROM'dakinden farklıysa "1"e kurulur.• Bit [5]: RAM uygulaması kartuştakinden farklıysa:<ul style="list-style-type: none">○ RAM uygulaması kartuştakinden farklıysa "1"e kurulur.• Bit [6] kullanılmıyor (durum 0)• Bit [7]: Denetleyici rezerve edildi:<ul style="list-style-type: none">○ Rezerve edildiysen "1"e kurulur.• Bit [8]: Uygulama Yazma modunda:<ul style="list-style-type: none">○ Uygulama korunuyorsa "1"e kurulur.• Bit [9] kullanılmıyor (durum 0)• Bit [10]: İkinci seri port yüklendi:<ul style="list-style-type: none">○ Yüklendiyse "1"e kurulur.• Bit [11]: İkinci seri port tipi: (0 = EIA RS-232, 1 = EIA RS-485):<ul style="list-style-type: none">○ "0"a kurulu = EIA RS-232○ "1"e kurulu = EIA RS-485• Bit [12]: Dahili bellekteki uygulama geçerli:<ul style="list-style-type: none">○ Uygulama geçerliyse "1"e kurulur.• Bit [13] Kartuşta geçerli uygulama:<ul style="list-style-type: none">○ Uygulama geçerliyse "1"e kurulur.• Bit [14] RAM'de geçerli uygulama:<ul style="list-style-type: none">○ Uygulama geçerliyse "1"e kurulur.• Bit [15]: İşletim için hazır:<ul style="list-style-type: none">○ İşletim için hazırsa "1"e kurulur. | |
| %SW18- %SW19 | 100ms mutlak zaman sayıcı | Sayıcı iki sözcük kullanarak çalışır: <ul style="list-style-type: none">• %SW18 en az değerlikli sözcüğü temsil eder.• %SW19 en değerlikli sözcüğü temsil eder. | S ve U |

| %SW30 | Son tarama süresi | Son tarama çevriminin işletim süresini ms olarak gösterir. Not: Bu süre tarama çevriminin başlatılmasından (giriş isteği) sona kadar (çıkışların güncellenmesi) geçen süredir. | S | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|-------------------------|--|--------|--------|-----------|-------|-------|---------------|-------|-------|--------|-------|--------|--------|-------|--------|------|-------|--------|-----|-------|--------|----|-------|--------|-----|-------|--------|--------|---|
| %SW31 | Maksimum tarama süresi | Son soğuk başlatmadan itibaren en uzun denetleyici tarama süresinin işletim süresini ms olarak gösterir. Not: Bu süre tarama çevriminin başlatılmasından (giriş isteği) sona kadar (çıkışların güncellenmesi) geçen süredir. | S | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| %SW32 | Minimum tarama süresi | Son soğuk başlatmadan itibaren en kısa denetleyici tarama süresinin işletim süresini ms olarak gösterir. Not: Bu süre tarama çevriminin başlatılmasından (giriş isteği) sona kadar (çıkışların güncellenmesi) geçen süredir. | S | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| %SW48 | Olay sayısı | Son soğuk başlatmadan beri kaç olayın işletildiğini gösterir. Not: Uygulamanın yüklenmesi ve soğuk başlatmadan sonra "0" a kurulum, her olayın işletimiyle bir artırılır. | S | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| %SW49 %SW50 %SW51 %SW52 %SW53 | Takvim bloğu süresi | Takvim bloğu (GZS) fonksiyonu: Mevcut tarih ve zaman değerlerini (BCD değerlerini) içeren sözcükler: %SW49 xN Haftanın günü (N=0 Pazartesi için) %SW50 00SS Saniye %SW51 HHMM Saat ve Dakika %SW52 MMDD Ay ve Gün %SW53 CCYY Yüzyıl ve Yıl Bu sözcükler, %S50 biti 0'da iken, sistem tarafından kontrol edilir. %S50 biti 1'e set edildiğinde, bu sözcükler kullanıcı programı veya terminal tarafından yazılabilir. | S ve U | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| %SW54 %SW55 %SW56 %SW57 | Takvim bloğu fonksiyonu | Takvim bloğu (GZS) fonksiyonu: Son enerji kesintisinin veya denetleyici durmasının tarih ve zamanını (BCD) içeren sistem sözcükleri: %SW54 SS Saniye %SW55 HHMM Saat ve Dakika %SW56 MMDD Ay ve Gün %SW57 CCYY Yüzyıl ve Yıl | S | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| %SW59 | Mevcut tarihi ayarla | Mevcut tarihi ayarlar. Mevcut tarihi ayarlamak için iki set halinde 8-bit içerir. İşlem daima, bitin yükselen kenarında gerçekleştirilir. Bu sözcük, %S59 biti tarafından izinlenir. <table border="1"> <thead> <tr> <th>Artım</th> <th>Azalış</th> <th>Parametre</th> </tr> </thead> <tbody> <tr> <td>bit 0</td> <td>bit 8</td> <td>Haftanın günü</td> </tr> <tr> <td>bit 1</td> <td>bit 9</td> <td>Saniye</td> </tr> <tr> <td>bit 2</td> <td>bit 10</td> <td>Dakika</td> </tr> <tr> <td>bit 3</td> <td>bit 11</td> <td>Saat</td> </tr> <tr> <td>bit 4</td> <td>bit 12</td> <td>Gün</td> </tr> <tr> <td>bit 5</td> <td>bit 13</td> <td>Ay</td> </tr> <tr> <td>bit 6</td> <td>bit 14</td> <td>Yıl</td> </tr> <tr> <td>bit 7</td> <td>bit 15</td> <td>Yüzyıl</td> </tr> </tbody> </table> | Artım | Azalış | Parametre | bit 0 | bit 8 | Haftanın günü | bit 1 | bit 9 | Saniye | bit 2 | bit 10 | Dakika | bit 3 | bit 11 | Saat | bit 4 | bit 12 | Gün | bit 5 | bit 13 | Ay | bit 6 | bit 14 | Yıl | bit 7 | bit 15 | Yüzyıl | U |
| Artım | Azalış | Parametre | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 0 | bit 8 | Haftanın günü | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 1 | bit 9 | Saniye | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 2 | bit 10 | Dakika | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 3 | bit 11 | Saat | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 4 | bit 12 | Gün | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 5 | bit 13 | Ay | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 6 | bit 14 | Yıl | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 7 | bit 15 | Yüzyıl | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| | | | |
|-----------------|--------------------------------|---|--------|
| %SW60 | GZS düzeltmesi değeri | Gerçek Zaman Saati (GZS) düzeltmesi değeri | U |
| %SW76 ila %SW79 | Aşağı sayıcılar 1-4 | Bu 4 sözcük, 1 ms'lik zamanlayıcılar olarak hizmet eder. Değerleri pozitif ise, her ms'de sistem tarafından ayrı ayrı azaltılırlar. Bu, 4 aşağı sayıcıya, 1 ms ila 32767 ms'lik bir çalışma aralığında ms olarak aşağı sayma olanağı verir. Bit 15'i, 1'e set etmek, azaltımı durdurabilir | S ve U |
| %SW114 | Takvim bloklarını (GZS) izinle | Takvim bloklarının (GZS) çalışmasını, kullanıcı programı veya Operatör Ekranı aracılığı ile izinler veya iptal eder, Bit 0: 1 = takvim bloğu #0'ı izinler Bit 15: 1 = takvim bloğu #15'i izinler Başlangıçta tüm takvim blokları izinlenmiştir, ilk durum 0'dır. Hiçbir takvim bloğu yapılandırılmamışsa, o zaman varsayılan değer FFFF'dir. | S ve U |
| %SW118 | Ana denetleyici durumu sözcüğü | Ana denetleyicide saptanan hataları gösterir. Bit 9: 0= harici hata veya haberleşme hatası Bit 12: 0= GZS yerleştirilmemiş. Bit 13: 0= konfigürasyon hatası (G/Ç genişletmesi yapılandırılmış fakat mevcut değil veya arızalı). Bu sözcüğün diğer tüm bitleri, 1'e set edilir ve rezerve edilir. Arızalı olmayan bir denetleyici için, bu sözcüğün değeri FFFFh'dir. | S |

Aşağıdaki tabloda kullanılan kısaltmalar verilmiştir.

Tablo-58 Kısaltmalar tablosu.

| Kısaltma | Tanım |
|----------|--------------------------------------|
| S | Sistem tarafından kontrol edilir. |
| U | Kullanıcı tarafından kontrol edilir. |

3.13 Uygulama Örnekleri

- 1) NA başlatma butonu (BAS_BUT) ile bir motoru (MOTOR) çalıştırıp NK durdurma butonu (DUR_BUT) ile durduran programı merdiven diyagramı ve deyim listesi ile programlayın.
- 2) Yukarıdaki örnekte verilen programa aşağıdaki eklemeleri yapınız.
 - a. Motorun çalıştığını gösteren bir sinyal lambası (SIN_LAM) motor durduktan sonra da 10s yanmaya devam etsin.
 - b. Her 5 duruştan sonra motora 20s boyunca yol verilmesin.
- 3) NA başlatma butonuna (BAS_BUT) basıldığında sırasıyla 1. motor 10 s sonra, 2. motor 20 s sonra, 3. motor 25 s sonra devreye girsin. 35 s sonra 3. motor, 40 s sonra da 1. ve 2. motorlar devreden çıkıp tekrar başlatma butonuna basılmasını beklesin. Programı merdiven diyagramı ve deyim listesi ile yazınız.
- 4) Yukarıdaki örneği tek bir zamanlayıcı ile yapınız.
- 5) Bir akış hattındaki pompanın arızalanması durumunda akışkan pompalanmayacaktır. Bu durumda hatta akışın olup olmadığını algılayan basınç anahtarı açılarak pompanın enerjisini kesip bir alarmı 1 dakika süreyle çalıştıracaktır. Programı merdiven diyagramı ile yazınız.
- 6) Kapalı bir tanktaki kritik basınç değerinin aşılmasıyla emniyet vanasını açıp bir alarmı 20 s sonra, 2 dakika süreyle çalıştıran programı deyim listesiyle yazınız.
- 7) Bir hattaki akışın durmasıyla açılan basınç anahtarının ısıtıcıyı devreden çıkararak bir alarmı 2 dakika süreyle ve 30'ar saniye aralıklarla kesikli olarak çalıştırmasını sağlayan programı merdiven diyagramı ve deyim listesi ile yazınız.

8) Aşağıdaki kontrol sistemini sağlayan programı merdiven diyagramı ile yazınız.

S1 başlatma butonu sistemin çalışmaya başlamasına izin verir. Sistem basıncı 100mbar'a düştüğünde basınç anahtarı kapanır ve kompresör 2 dakika çalışır. Sistemdeki sıvı seviyesi 10m'nin altına düştüğünde seviye anahtarı kapanarak bir pompayı çalıştırır. Eğer sıcaklık 100°C'nin üstüne çıkarsa sıcaklık anahtarı kapanır, kompresör ve pompa çalışmaz. Acil durumlar için sistemin durdurulması gerektiğinde S2 durdurma butonu kullanılacaktır. Sistem basıncı 100 defa verilen değerin altına düşerse kompresör bir daha çalışmayacaktır.

9) 3 katlı bir garajda 1. kat maksimum 12, 2. kat maksimum 10 ve 3. kat maksimum 8 araba almaktadır. Katların giriş ve çıkışları birbirinden bağımsızdır. Katlar dolunca giriş kapıları kapanmaktadır. 3 kattaki arabaların toplamı 20 olunca bir lamba yanıp sönmeye başlayacaktır. Bütün katlar dolunca bir başka lamba sürekli yanacaktır. 3 kattaki arabaların toplamı bir dahili sözcükte saklanacaktır.

10) Aşağıdaki kontrol programını yazınız.

Bir oto parkın giriş ve çıkış olmak üzere iki kapısı vardır. Giriş kapısında içeriye kaç arabanın daha girebileceğini belirten gösterge vardır. Her iki kapıda ayrı iki optik sensör kullanılmaktadır. Ters kapıdan giriş ve çıkışın mümkün olmadığını ve aynı anda giriş ve çıkışın yapılmadığını varsayalım. Oto parkın kapasitesi 80 arabadır ve park dolduğunda giriş kapısındaki bariyer kapatılacaktır. NOT: İçeriye daha kaç arabanın gireceğini göstermek için %MW2 sözcüğünü kullanın. Giriş kapısı için %Q0.0 çıkışını, giriş kapısındaki sensör için %I0.0 ve çıkış kapısındaki için ise, %I0.1 girişini kullanınız.

11) Tambur denetleyici (drum controller) üzerinde toplam 4 adım bulunmaktadır. Bu adımların işleme süreleri ve enerjilendirilen çıkışlar aşağıdaki tabloda verilmiştir. DRUM denetleyici kullanarak bu ardışıklığı sağlayan programı yazınız.

| | %Q0.6 | %Q0.5 | %Q0.4 | %Q0.3 | %Q0.2 | %Q0.1 | süre |
|--------|-------|-------|-------|-------|-------|-------|------|
| Adım 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2s |
| Adım 1 | 0 | 0 | 1 | 1 | 0 | 0 | 2s |
| Adım 2 | 1 | 1 | 0 | 0 | 0 | 0 | 3s |
| Adım 3 | 1 | 1 | 1 | 1 | 1 | 1 | 3s |

%I0.2 ile DRUM 0 nolu adıma, %I0.3 ile 3 nolu adıma atlatılacaktır.

12) İki konumlu iki anahtarın konumuna göre 6 motorun kontrolü aşağıdaki gibidir. DRUM denetleyici kullanarak bu ardışıklığı sağlayan programı yazınız.

| | A | B | %Q0.0 | %Q0.1 | %Q0.2 | %Q0.3 | %Q0.4 | %Q0.5 |
|--------|---|---|-------|-------|-------|-------|-------|-------|
| Adım 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| Adım 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| Adım 2 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| Adım 3 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |

13) Şekilde görülen yıkama kazanı NA başlatma butonuna (BAS_BUT) basıldığında sırasıyla aşağıdaki işlemleri tekrarlar. Bu programı DRUM ile yazınız.

1. V1 vanası açılır.
2. Kazan dolunca V1 vanası kapanır ve 5s boyunca V3 vanası açılarak kazana deterjan eklenir.
3. Deterjan eklendikten sonra M1 motoru 10s çalışır.
4. 10s sonra M1 motoru durur, V2 vanası açılarak kazanın suyu boşalır.
5. Kazan tamamen boşaldıktan sonra deterjan eklenmeden 1, 3, 4. adımlar tekrarlanarak durulama işlemi yapılır.
6. Sistem başa dönüp tekrar başlatılmayı bekler.

Acil durdurma sistemi başlangıca döndürerek tekrar başlatılmayı bekler.

14) Twido PLC üzerinde bulunan 1 nolu potansiyometriyi kullanarak %TM0 adresli bir zamanlayıcı çıkışının 5 ile 10s arasında ayarlanabilir bir gecikmeyle aktif olmasını sağlayınız. Zamanlayıcı %I0.0 girişine bağlı bir sinyal ile çalışmaya başlayıp %Q0.0 çıkışını kontrol edecektir.

Not: 1 nolu potansiyometrenin ayarladığı sözcük adresi %IW0.0.0'dır ve değeri 0 ile 1023 arasında değişir. Zamanlayıcı için şu parametreleri seçiniz: TYPE: TON, TB: 10ms

15) Bir taşıma bandından gelen parçalar B0 optik sensörü ile algılanınca A kaldırma silindiri ile yukarı kaldırılmakta ve B itme silindiri ile yukarıdaki taşıyıcı bandın üzerine itilmektedir. B silindiri geriye geldikten sonra A silindiri aşağıya inecektir. Silindir konumları A silindiri için geride B1, ileride B2 ve B silindiri için geride B3, ileride B4 manyetik anahtarlarıyla algılanmaktadır. Her iki silindir de çift etkilidir. Bu sıralı kontrol programını deyim listesi ile yazın.

4 KAYNAKÇA

1. Hughes, Thomas A. Measurement and Control Basics, 1991. Instrument Society of America. ISBN: 1-55617-097-1. pp. 306
2. Aksoy, Saadettin. Programlanabilir Lojik Denetleyiciler ve Mühendislik Uygulamaları, 2004. Değişim Yayınları. ISBN: 975-8289-63-2. ss. 240
3. Twido programmable controllers Software Reference Guide, TWD USE 10AE eng Version 3.2, Schneider Electric
4. Twido programmable controllers Hardware Reference Guide TWD USE 10AE eng Version 3.2, Schneider Electric
5. Twido Programlanabilir Kontrolörler Donanım Referans Kılavuzu TWD USE 10AT Sürüm 1.0, Schneider Electric
6. Twido Programlanabilir Kontrolörler Donanım Referans Kılavuzu TWD USE 10AT Sürüm 1.0, Schneider Electric
7. Product data sheet Characteristics, TWDLCAE40DRF compact PLC base Twido - 100..240 V AC supply - 24 I 24V DC - 16 O, Schneider Electric