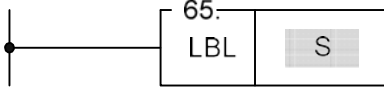


## Gelişmiş Fonksiyon Komutu

FUN 65 LBL	ETİKET	FUN 65 LBL
---------------	--------	---------------

### Ladder symbol



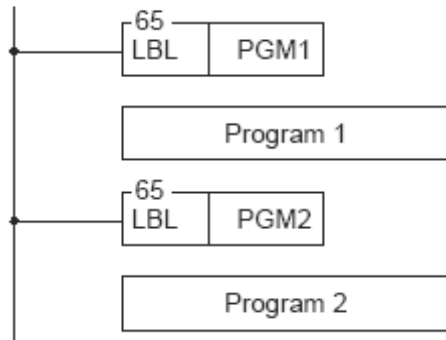
S : Alfa-numerik, 1~6 karakter

- Bu komut, program içerisinde belirli adresleri etiketlemekte kullanılır. CALL komutu ve kesme servisi JUMP komutunun çalışması için hedef adres sağlarlar. Aynı zamanda, programın değerlendirilmesi ve okunabilirliğini gerçekleştirmek amacıyla doküman için kullanılırlar..
- Bu komut, açıklama için veya akış prosedürünün kontrolünü gerçekleştirmek için program adresi işaretlemek gibi işlemler yapar. Komut kendi kendine herhangi bir hareket gerçekleştiremez; program bu komutu içersin veya içermesin program çalışmasının sonucu bu komutu etkilemeyecektir.
- Etiket ismi, 1 ~ 6 arasından alfanumerik karakterler tarafından oluşturulabilir ve aynı program içinde ikinci kez kullanılamaz. Aşağıdaki etiket isimleri interrupt fonksiyonunda kullanılmak üzere ayrılmışlardır. Bu "ayrılmış wordler", normal program etiketlerinde kullanılamazlar.

Ayrılmış Wordler	Tanım
X0+I~X15+I ( INT0~ INT15 ) X0□I~X15□I( INT0□~INT15□)	Harici girişli (X0~X15) interrupt servis rutini etiketler.
HSC0I ~HSC7I	Yüksek hızlı sayıcı HSC0~HSC7 interrupt servisi yordamı için etiketler.
1MSI ( 1MS ) , 2MSI ( 2MS ) , 3MSI ( 3MS ) , 4MSI ( 4MS ) , 5MSI ( 5MS ) , 10MSI ( 10MS ) , 50MSI ( 50MS ) , 100MSI ( 100MS )	Dahili zamanlayıcı interrupt servisi yordamının 8 çeşidi için etiketler.
HSTAI ( ATMRI )	Yüksek hızda sabitlenmiş zamanlayıcı interrupt servisi yordamı için etiket.
PSO0I ~ PSO3I	Pulse çıkış komutu bitmiş interrupt servisi yordamı için etiketler.

Interrupt servisi yordamı, sadece tablo üzerinde listelenmiş etiket isimlerinde kullanılabilirler, normal subrutineda ayrılmış etiket kullanımında yanlış varsa CPU hatası veya tanımlanamayan işlemler oluşabilir..

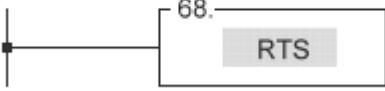
Aşağıdaki diyagramda gösterilen etiket, program açıklaması gibi kullanılmıştır (çağırma veya atlama hedef için bir etiket gibi davranmaz). Atlama kontrolündeki etiketlemenin uygulaması açıklaması için JMP komutuna bakınız. Etiketlemede alt program gibi hizmetler için detaylı bilgi CALL komutunda bulabilirsiniz.



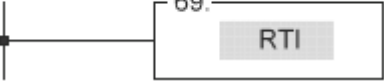
FUN 66 P JMP	JUMP	FUN 66 P JMP
<p style="text-align: center;"><u>Ladder symbol</u></p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div data-bbox="220 398 670 488"> </div> <div data-bbox="869 398 1204 432"> <p>LBL : Atlanacak program etiketi</p> </div> </div>		
<ul style="list-style-type: none"> <li>• Atlama kontrolü "EN"=1 veya "EN↑" (P komutu) 0→1 şeklinde değiştiğinde, PLC, işaretlenmiş etiketten sonraki yere atlayacak ve program bu kısımdan çalışmaya devam edecektir.</li> <li>• Bu komut, uygulamalar için özellikle uygundur burada programın bazı parçaları kesin durumlar altında çalışacaklardır. Bu, tüm programı çalıştırmazken tarama zamanını daha kısaltabilecektir.</li> <li>• Bu komut geri atlamaya izin verir (mesela; LBL'nin adresi, JMP komutunun adresinden önce gelir). Ancak, atlama eylemi sebebi watchdog zamanlayıcı tarafından kurulan sınırı aşan tarama zamanı ise dikkate alınmalıdır. WDT kesme oluşacak ve çalışma duracaktır.</li> <li>• Jump komutu, sadece ana programlar arasında atlamaya veya alt program alanı arasında atlamaya izin verir. Ana/alt program alanının ortasından atlayamazlar.</li> </ul> <div style="display: flex; justify-content: space-between; align-items: flex-start; margin-top: 20px;"> <div data-bbox="220 1120 670 1456" style="width: 45%;"> </div> <div data-bbox="805 1093 1444 1361" style="width: 50%;"> <p>Soldaki diyagramda, X0=1 olduğunda, program PATHB şeklinde adlandırılmış LBL pozisyonuna atlayacak ve program B çalışmaya devam edecektir. Bu yüzden, A programı atlanacak ve A programının komutunun hiçbiri yürütülmeyecektir. A programı ile birleştirilmiş bobinler ve registerların durumu değişmeden tutulacaktır (A kısmında program yok ise).</p> </div> </div>		

## Gelişmiş Fonksiyon Komutu

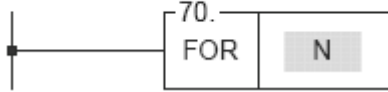
FUN 67 P CALL	CALL	FUN 67 P CALL
<p style="text-align: center;"><u>Ladder symbol</u></p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div data-bbox="255 369 699 470"> </div> <div data-bbox="861 392 1300 436"> <p>LBL: Çağırılacak alt program etiket ismi.</p> </div> </div>		
<ul style="list-style-type: none"> <li>• Çağırma komutu "EN"=1 veya "EN ↑" (P komutu) 0→1'e değiştiğinde, PLC aynı etiket ismini taşıyan subrutini çağrılacaktır. Alt program çalıştığında, program normal olarak çalışmaya devam edecek ama program karşılandığında RTS komutu sonra programın akışı CALL komutundan sonra adrese hemen geri dönecektir..</li> <li>• Tüm alt programlar, bir "alt program komutu olan RTS'den geri dönüş" komutu ile sonlanmalıdır. Aksi takdirde CPU duracak veya çalışma hatası oluşacaktır. Yine de, bir RTS komutu, sağdaki diagramda alt programlar SUB1 ~ 3 gibi alt programlar (çok girişli subrutinler şeklinde adlandırılmışlardır; giriş nokta çiftleri farklıdır, aynı yoldan geri dönerler) tarafından paylaşılabilirler.</li> <li>• Ana program, bir alt programı çağırdığında, alt program aynı zamanda en fazla (interrupt yordamı dahil) 5. seviyeye kadar diğer alt programları (dallanmış alt program çağrılmış) çağırabilirler.</li> </ul> <div style="display: flex; justify-content: space-around; align-items: center;"> <div data-bbox="183 1243 861 1579"> </div> <div data-bbox="853 683 1404 1332"> </div> </div>		
<ul style="list-style-type: none"> <li>• Interrupt servisi programları HSC0I~HSC7I, PSO0I~PSO3I, X0+I~X15+I/INT0~INT15, X0I~X15I/INT0□~INT15□, HSTAI/ATMRI, 1MSI/1MS, 2MSI/2MS, 3MSI/3MS, 4MSI/4MS, 5MSI/5MS, 10MSI/10MS, 50MSI/50MS, 100MSI/100MS) aynı zamanda bir çeşit alt programlardır. Aynı zamanda alt program alanına yerleştirilmişlerdir. Ancak, interrupt servis programının çağrılması, karşılıklı interrupt servisi programı gerçekleştirmek için donanımın sinyali tarafından tetiklenmesi durdurulmuştur (bu interrupt servisi programının çağrılması gibi adlandırılmıştır). Interrupt servis programı, yüksek öncelikli diğer interrupt tarafından kesilmiş veya alt programı çağırabilmelidir. Çünkü aynı zamanda bir subroutine, interrupt servisi programı veya alt programının 4 seviyesi tarafından kesilmiş veya çağrılabilir. Açıklama için RTI komutuna bakınız.</li> </ul>		

FUN 68 RTS	ALT PROGRAMDAN GERİ DÖNÜŞ	FUN 68 RTS
<p style="text-align: center;"><u>Ladder symbol</u></p> 		
<ul style="list-style-type: none"><li>• Bu komut, bir alt programın sonunu simgelemek için kullanılmıştır. Bu yüzden, sadece alt program alanı içerisinde görünürler. Giriş alanında kontrol sinyali yoktur, bu yüzden kontaklar veya seri bir bağlantı yolu yoktur. Bu komut kendini tutar ve güç hattına direk olarak bağlanır.</li><li>• PLC bu komutla karşılaştığında bir alt programın çalışması sayesinde tamamlanmıştır. Bu yüzden, CALL komutundan sonra hemen adrese dönecektir. Bu program önceden çalıştırılmıştır ve çalışmaya devam edecektir.</li><li>• Üç akış kontrol komutu MC, SKP, veya JMP'nin herhangi birinin komutla karşılaşırsa, bu komut çalıştırılmaz (var olmadığına dikkat edilecektir). Komut üstünde, alt program sebep olan ve subrutinede kullanılan RTS komutu çalışmaz sonra PLC çalışması duraklar ve M1933 1'e kurulur (akış hata bayrağı). Bu yüzden, ne olursa olsun akış sürer, RTS komutunun herhangi bir alt program ile eşlenebildiğinden emin olunmalıdır.</li><li>• RTS komutunun kullanımı için, CALL komutu için kullanılan komutlara bakınız.</li></ul>		

## Gelişmiş Fonksiyon Komutu

FUN 69 RTI	RETURN FROM INTERRUPT	FUN 69 RTI
<p style="text-align: center;"><u>Ladder symbol</u></p> 		
<ul style="list-style-type: none"><li>• Bu komutun fonksiyonu RTS'ye benzerdir. Yine de, RTS alt programın çalışmasını bitirmek için kullanılır ve RTI interrupt servisi programının çalışmasını bitirmek için kullanılır. RTS komutunun açıklamasına bakınız.</li><li>• RTI komutu, birden fazla interrupt servis programı tarafından paylaşılabilir. Kullanımı, bir çok alt program tarafından RTS'in paylaşılması ile aynıdır. Bunun için CALL komutunun açıklamasına bakınız.</li><li>• interrupt ve çağırma arasındaki fark çağırmanın alt program ismi (LBL)dir ve bu kullanıcı tarafından belirlenir ve etiket adı ve call komutu diğer alt program ve ana programda dâhil edilmiştir. Bu yüzden, PLC, CALL gerçekleştiğinde ve giriş "EN"=1 veya "EÑ (P komutu) 0→1 şeklinde değiştiğinde, PLC bu alt programı çağıracaktır. interrupt servis programının çalışması için, diğer daha az önemli işler durmakta ve donanım sinyalleri ile direk olarak kullanılmaktadır ve donanım sinyaline uygun interrupt servisi programı gerçekleşmektedir. Çalışmak taratılmış olmak ihtiyacı call komutu ile karşılaştırıldığında, interrupt, dış dünyaya cevapta daha gerçek zamanlıdır. Ek olarak, Interrupt servis programı etiket ismi ile çağrılmayabilir; bu yüzden, PLC tarafından sunulmuş çeşitli interruptlara uygun özel etiket ismi ile "ayrılmış wordler"i korumalıyız. Örneğin; Ayrılmış word X0+I, X0 giriş noktasında meydana gelmiş interruptta ayrılmıştır; Alt program X0+I'in etiketini içerdiği sürece, giriş noktası X0 interruptı ortaya çıktığında PLC diğer düşük program öncelikleri duracak ve program çalışması hemen X0+I gibi etiketlenmiş alt program adresine atlayacaktır.</li><li>• CPU daha yüksek önceliğe (donanım yüksek hızlı sayıcı interruptsı gibi) veya aynı öncelik (öncelik düzeyleri için bölüm 10'a bakınız) interrupt programı boşaltılırken bir interrupt oluşmuştur. PLC, tüm yüksek öncelikli prgramlar tamamlanana kadar bu interrupt için interrupt programı çalışmayacaktır..</li><li>• RTI komutu, interrupt servis komutunda gerçekleştirilmemiş veya ulaşılamamışsa ciddi CPU kapanmalarına sebep olabilir. Sonuç olarak, herhangi bir interrupt servsi programında çalıştırılmayacaktır.</li><li>• Interrupt kullanımı için örnek ve detaylı açıklama için bölüm 10'a bakınız.</li></ul>		

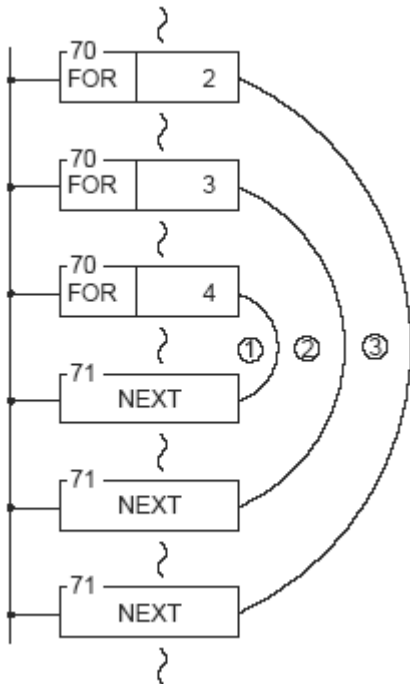
FUN 70 FOR	FOR	FUN 70 FOR
---------------	-----	---------------

Ladder symbol

N : Döngü çalışmasının zamanlama sayısı

Range	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
Operand	WX0   WX240	WY0   WY240	WM0   WM1896	WS0   WS984	T0   T255	C0   C255	R0   R3839	R3840   R3903	R3904   R3967	R3968   R4167	R5000   R8071	D0   D4095	1   16383
N	○	○	○	○	○	○	○	○	○	○	○	○	○

- Bu komutun giriş kontrolü yoktur. güç hattına direk olarak bağlanmıştır ve herhangi bir durum ile seri olarak bağlanamaz..
- FOR ve NEXT komutları içerisindeki programlar bir program döngüsü biçimindedir (program döngüsünün başlangıcı FOR'dan sonra NEXT komutu ve sondada NEXT'den önce kullanılır). PLC'de FOR komutu çalıştırılır, bu komuttan sonra N değeri kaydedilir sonra döngüdeki programların sonuna başlattan başlayarak N kadar ardışık çalışma yapılır. Döngü dışı atıldığında, NEXT komutundan sonra komut hemen çalışmaya devam eder.
- Döngü dallanmış yapıdadır, örn; döngü diğer döngülere dahildir, bir soğan gibi. 1 döngü 1 seviye şeklinde adlandırılmıştır ve maksimum 5 seviye olabilmektedir. FOR ve NEXT komutları çift olarak kullanılmalıdır. İlk FOR komutu ve son NEXT komutu dallanmış döngünün en dıştaki kısımlarıdır. İkinci FOR komutu ve ikinci NEXT komutu, ikinci düzeydir. Son FOR komutu ve ilk NEXT komutu döngünün en içteki seviyeleridir.




Sol diyagramdaki örnekte , 1 döngüsü  $4 \times 3 \times 2 = 24$  kez çalışmalıdır, 2 döngüsü  $3 \times 2 = 6$  kez çalıştırılır ve 3döngüsü 2 kez çalıştırılacaktır.

FOR komutu var ve uygun NEXT komutu yok ise, veya dallanmış döngüdeki FOR ve NEXT komutları çift çift kullanılmıyorsa veya FOR ve NEXT dizisi yanlış yerleştirilmiş ise bir söz dizimi hatası oluşacak ve bu program çalışmayacaktır.

Döngüdeki, JMP komutu döngü dışına atlamakta kullanılabilir. Ancak, girilmiş olan döngü bir kez alındığına dikkat edilmelidir. (ve FOR döngüsü çalıştırılmış), program akış atlamaları ne olursa olsun, Programın sonuna veya END komutuna ulaşmadan önce NEXT komutuna ulaşabilmelidir. Aksi takdirde FBs-PLC çalışması duracak ve hata mesajı vericektir.

N'nin efektif aralığı 1~16383 kezdır. FBs-PLC bu aralığın dışında 1 gibi davranacaktır. Eğer N'nin aralığı çok geniş ve döngü programı çok büyük ise bir WDT oluşabileceği dikkate alınmalıdır.

## Gelişmiş Fonksiyon Komutu

FUN 71 NEXT	END DÖNGÜSÜ	FUN 71 NEXT
<p style="text-align: center;"><u>Ladder symbol</u></p>  <p>The diagram shows a ladder logic symbol for the FUN 71 NEXT command. It consists of a vertical line on the left with a small circle at its top. A horizontal line extends from this circle to the left side of a rectangular box. The box is labeled '71.' at the top left and 'NEXT' in a shaded area in the center.</p>		
<ul style="list-style-type: none"><li>• Bu komut ve FOR komutu birlikte bir program döngü formudur. Bu komut kendi kendine giriş kontrolüne sahip değildir, güç hattına direk olarak bağlanmıştır ve herhangi bir durum ile seri olamaz..</li><li>• PLC henüz döngü girişine sahip olmadığında, ama NEXT komutu ulaştığında PLC herhangi bir hareket yapmayacak, nitekim bu komut çalıştırılmayacaktır.</li><li>• Bu komutun kullanımı için, önceki sayfadaki FOR komutu açıklamalarına bakınız.</li></ul>		