

**PLC**

**Programlama Uygulamaları  
(Merdiven Şeması/MELSEC iQ-R  
Serisi)**

Bu kurs, MELSEC iQ-R serisi programlanabilir kontrolörlerin temellerini anlayan ve programlamanın bir sonraki adımını öğrenmek isteyen kullanıcılara yöneliktir.

L(CTS)00691TUR

Bu kurs, Programlama İlkeleri (Merdiven Şeması) kursunu tamamlayan veya eşdeğer bilgiye sahip olan kullanıcılara yöneliktir. Bu kurs, MELSEC iQ-R serisi programlanabilir kontrolörler için verimli programlama ve hata ayıklama, aygıtların ileri kullanımı ve etiket programlama hakkında bilgiler sağlar.

Bu kurs için ön koşul olduğundan, belirtilen kursları halihazırda tamamlamış olmanız ya da bu düzeyde bir bilgiye sahip olmanız gerekmektedir.

- MELSEC iQ-R Serisi Temel Bilgileri
- Programlama İlkeleri

Bu kursun içeriği aşağıdaki gibidir.

#### Bölüm 1 - Verimli programlama

Verimli programlamaya yönelik yöntemler ve ayarlar

#### Bölüm 2 - İleri programlama

Aygıtların ileri kullanımı ve etiket programlama

#### Bölüm 3 - Verimli hata ayıklama

Mühendislik yazılımının verimli hata ayıklamaya yönelik işlevleri

#### Son Test

Geçme notu: %60 veya üzeri gereklidir

Sonraki sayfaya git	>	Sonraki sayfaya git.
Önceki sayfaya dön	<	Önceki sayfaya dön.
İstenen sayfaya ulaş	TOC	"İçindekiler Tablosu" görüntülenerek istediğiniz sayfaya ulaşabilmenizi sağlar.
Eğitimden çık	X	Eğitimden çık.



**Güvenlik önlemleri**

Mevcut ürünleri kullanarak öğrendiğinizde, lütfen ilgili kılavuzlardaki güvenlik önlemlerini dikkatlice okuyun.

**Bu kurstaki önlemler**

Kullandığınız yazılım sürümünde görüntülenen ekranlar bu kurstakilerden farklı olabilir. Bu kursta aşağıdaki yazılım sürümü kullanılır:

- GX Works3 Sürüm 1.044W

## **Bölüm 1** Verimli programlama

Bu bölümde, mühendislik yazılımının verimli programlamaya yönelik ayarları ve label programlamanın temelleri açıklanmaktadır.

- 1.1 Farklı sistemlerdeki programların daha kolay kullanımı
- 1.2 Bellek alanının aygıt kullanım durumuna göre ayarlanması
- 1.3 Uygulamalar ile ilgili label adlarının kullanılması
- 1.4 Program okunurluğunun iyileştirilmesi

## 1.1

## Farklı sistemlerdeki programların daha kolay kullanımı

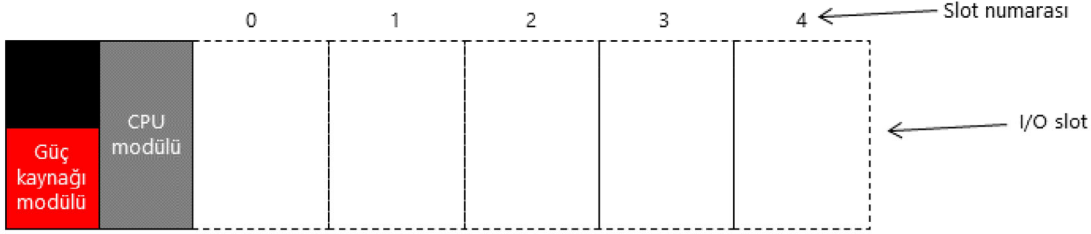
Bu bölümde, farklı sistemlerdeki programların daha kolay kullanımı için verimli I/O sayısı ataması açıklanmaktadır.

### 1.1.1

### Otomatik I/O numarası ataması

I/O numaraları, temel üniteye kurulu modüllere, CPU modülüne en yakın yuvadan başlanarak sıralı olarak atanır. I/O numaraları 16 noktalı birimler halinde (0 – F) atanır.

Atanacak mevcut noktaların (kapladığı alan) sayısı, modül tipine (16, 32, 64 vb.) bağlı olarak farklılık gösterir.



Aşağıdaki örnekte beş adet 16 noktalı modül kuruludur.

	0	1	2	3	4
	16 nokta	16 nokta	16 nokta	16 nokta	16 nokta
	00	10	20	30	40
	-	-	-	-	-
	0F	1F	2F	3F	4F

The table shows the I/O numbering for five 16-point modules. The first row shows the number of points covered by each module (16 nokta). The second row shows the starting I/O number (00, 10, 20, 30, 40). The third row shows the ending I/O number (-). The fourth row shows the I/O numbers in hexadecimal (0F, 1F, 2F, 3F, 4F). The columns are labeled 'Kapladiğı alan' and 'I/O sayısı'.

## 1.1.1

## Otomatik I/O numarası ataması

16, 32 ve 64 dolu noktaya sahip modüller aynı anda kullanıldığında I/O numaraları aşağıdaki şekilde atanır:

		0	1	2	3	4
Güç kaynağı modülü	CPU modülü	16 nokta	32 nokta	64 nokta	32 nokta	16 nokta
		00	10	30	70	90
		-	-	-	-	-
		0F	2F	6F	8F	9F

Kurulu modüllerin arasında boş bir slot varsa boş slotta da I/O numarası atanır.

		0	1	2	3	4
Güç kaynağı modülü	CPU modülü	16 nokta	32 nokta	64 nokta	16 nokta	16 nokta
		00	10	30	70	80
		-	-	-	-	-
		0F	2F	6F	7F	8F

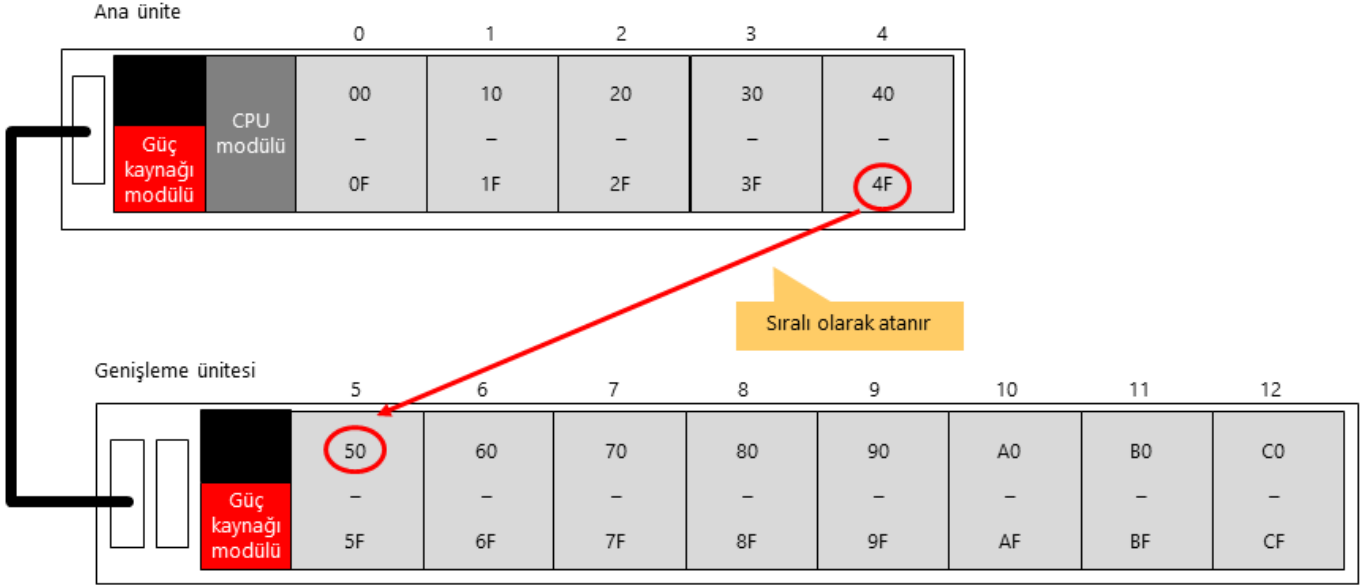
Boş slot

Boş slotta varsayılan olarak 16 nokta atanır. Atanan noktaların sayısı, 0 – 1024 nokta (16 noktalı birimde) aralığında parametre ayarıyla değiştirilebilir.

## 1.1.1

## Otomatik I/O numarası ataması

Genişleme ünitesinin I/O numaraları, ana ünitenin son I/O numarasının ardından otomatik olarak atanır.



## 1.1.2

## Sabit I/O numarası ataması

I/O numaraları manuel olarak atandığında, atanmış I/O numaraları, modül konfigürasyonu değişse bile sabit kalır ve değişmez. Bu durum, modül konfigürasyonundan bağımsız olarak, aynı kontrol için aynı kontrol programının kullanılabilmesi anlamına gelir.

### Otomatik atama

Modüller eklenmeden önce

	CPU modülü	Giriş modülü	Çıkış modülü	Akıllı işlem modülü	
Güç kaynağı modülü		64 nokta	64 nokta	16 nokta	
		X00 – X3F	Y40 – Y7F	X/Y80 – X/Y8F	

Modüller eklendikten sonra  
(32 noktalı bir giriş modülü ve 16 noktalı bir çıkış modülü eklenir.)

	CPU modülü	Giriş modülü	Giriş modülü	Çıkış modülü	Çıkış modülü	Akıllı işlem modülü	
Güç kaynağı modülü		64 nokta	32 nokta	64 nokta	16 nokta	16 nokta	
		X00 – X3F	X40 – X5F	Y60 – Y9F	YA0 – YAF	X/YB0 – X/YBF	

Modüller eklendikten sonra I/O numaraları yeniden atanır.

## 1.1.2

## Sabit I/O numarası ataması

### Sabit I/O numaraları için manuel atama

Modüller eklenmeden önce

		Giriş modülü	Çıkış modülü	Akıllı işlem modülü	
Güç kaynağı modülü	CPU modülü	64 nokta	64 nokta	16 nokta	
		X00 – X3F	Y40 – Y7F	X/Y80 – X/Y8F	

Modüller eklendikten sonra  
(32 noktalı bir giriş modülü ve 16 noktalı bir çıkış modülü eklenir.)

		Giriş modülü	Eklenen modüller		Akıllı işlem modülü	
Güç kaynağı modülü	CPU modülü	64 nokta	Giriş modülü 32 nokta	Çıkış modülü 16 nokta	16 nokta	
		X00 – X3F	X90 – XAF	Y40 – Y7F	Y80 – YBF	X/Y80 – X/Y8F

Modüllerin eklenmesinden bağımsız olarak aynı I/O numaraları atanır.

Mevcut modüllerin I/O numaraları aynı kaldığından, yalnızca eklenen modüller ile ilgili programların eklenmesi veya değiştirilmesi gerekir.

### 1.1.3

## Modül konfigürasyon şeması kullanılarak otomatik I/O numarası ataması

Modül konfigürasyonu, MELSOFT GX Works3 yazılımının modül konfigürasyon şeması kullanılarak ayarlanabilir. Bir modül modeli seçin ve modülü yerleştirmek için bir slot üzerine sürükleyip bırakın. I/O numaraları, şemada gösterildiği gibi CPU modülüne en yakın modülden başlanarak modüllere sıralı olarak atanır. Modülün başlangıç I/O numaralarını görüntülemek için yerleştirilmiş bir modül seçin.

The screenshot displays the MELSOFT GX Works3 Module Configuration interface. On the left, a rack configuration diagram shows slots for POW, CPU 0, and modules 1, 2, 3, and 4. A red box highlights slot 1, and a red arrow points to the 'Element Selection' window. A yellow callout box above the rack diagram contains the text: "Bir modül modelini sürükleyip bırakın." (Drag and drop a module model).

The 'Element Selection' window on the right shows a list of modules. The 'RY42NT2P' module is selected and highlighted with a red box. A red arrow points from this module to the 'Input the Configuration Detailed Information' window. A yellow callout box below this window contains the text: "Seçili modülün başlangıç I/O numarası" (Start I/O number of the selected module).

Module Model	Points	Type
RY40PT5B(S2S)	16	points(Source type)
RY40PT5P	16	points(Source type)
RY41NT2H	32	points(Sink type/High-Speed)
RY41NT2P	32	points(Sink type)
RY41PT1P	32	points(Source type)
RY41PT2H	32	points(Source type/High-Speed)
RY42NT2P	64	points(Sink type)
RY42PT1P	64	points(Source type)

Field	Value
Start XY	0040
Points	64 Points



## 1.1.4

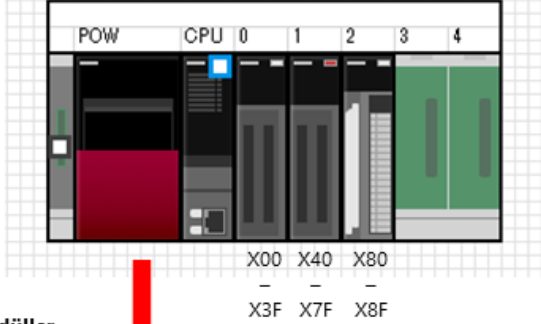
### Modül konfigürasyon şeması kullanılarak manuel I/O numarası ataması

Aşağıdaki örnekte, GX Works3'ün modül konfigürasyon şemasını kullanarak I/O numaralarının nasıl manuel olarak atanacağı açıklanmaktadır.

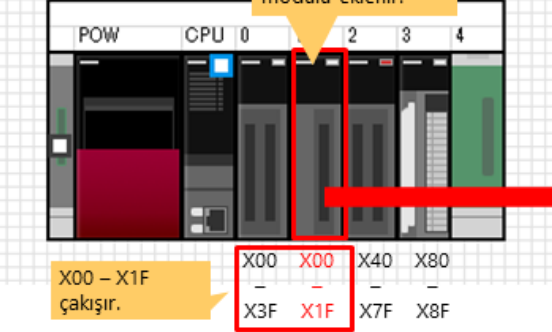
Modül konfigürasyonu, modül konfigürasyon şemasına yeni bir modül eklenerek değiştirildiğinde, mevcut modüllerin yerleşimi değiştirilmişse eklenen modülün I/O numarası mevcut numaralarla çakışır. I/O numaralarını çakışmayacak şekilde düzenleyin ve modül konfigürasyonunu belirleyin.

Modül konfigürasyonu, I/O numaraları çakışır haldeyken belirlenirse bir hata mesajı görüntülenir. Bu noktada, otomatik atama görüntülenen hata mesajı penceresinden yürütülebilir.

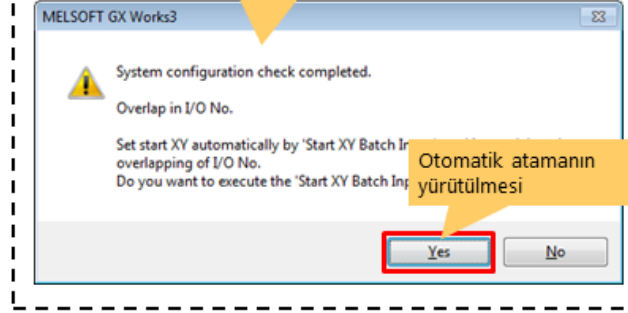
#### Modüller eklenmeden önce



#### Modüller eklendikten sonra



Modül konfigürasyonu, I/O numaraları çakışır haldeyken belirlenirse bir hata mesajı görüntülenir.



32 noktalı giriş modülünün başlangıç I/O sayısını 0000'dan 0090'a değiştirin.

Input the Configuration Detailed Information	
RX41C4	
Start XY	0090
Points	32 Points

Modül konfigürasyonunun belirlenmesi

## 1.2

## Bellek alanının aygıt kullanım durumuna göre ayarlanması

### 1.2.1

### Device/label bellek alanı ayarları

CPU modülleri için kullanılan aygıt (device) noktalarının sayısı, CPU modülünün tipine göre farklılık gösterir. Device noktalarının ilk sayısı, CPU modülünün device alanı kapasitesine dayalı olarak atanır.

R04CPU'nun device alanının kapasitesi 40K word.

Kullanılmayan alanların azaltılması, device noktalarının sayısını ilk değerden artırır.

Aşağıdaki şekilde, bellek alanını ayarlamaya yönelik parametrelerin bulunduğu "Device/Label Memory Area Setting" (Aygıt/Etiket Bellek Alanı Ayarı) penceresi gösterilmektedir.

Device alanının kapasitesi, label alanının veya dosya depolama (file storage) alanının kapasitesi azaltıldığında artar. Bunlara ek olarak, device/label bellek alanının tamamının kapasitesi, ilave bir SRAM kullanılarak artırılabilir.

Item	Setting
<b>Device/Label Memory Area Setting</b>	
Extended SRAM Cassette Setting	Not Mounted
Device/Label Memory Area Capacity Setting	
Device Area	
Device Area Capacity	40 K Word
Label Area	
Label Area Capacity	30 K Word
Latch Label Area Capacity	2 K Word
File Storage Area Capacity	128 K Word
Device/Label Memory Configuration Confirmation	<Confirmation>
Device/Label Memory Area Detailed Setting	
Device Setting	<Detailed Setting>
Latch Type Setting of Latch Type Label	Latch (1)

Device alanının kapasitesi

## 1.2.2

## Device ayarları

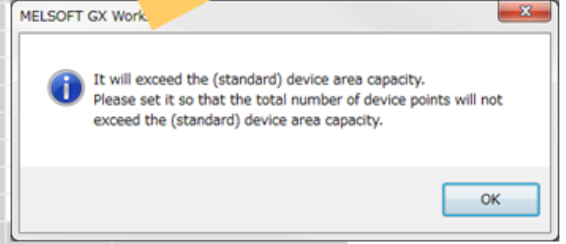
Her bir device a atanan device noktalarının sayısı, "Device Setting" penceresinde değiştirilebilir. Bazı devicelerin ilk değeri 0 noktadır. Bu gibi deviceleri kullanırken nokta sayısını atayın.

**Device noktalarının sayısı:**  
Her bir device tarafından kullanılan noktaların sayısını ayarlayın.

- İlk değerler önceden atanır
- Beyaz hücrelerde bulunan değerler değiştirilebilir
- Device noktalarının sayısını 16 noktalı birimler halinde ayarlayın
- 1K nokta 1024 nokta anlamına gelir

Device noktalarının toplam sayısı CPU modülünün kapasitesini aşarsa ayarın değiştirilmesi gerektiğini belirten bir mesaj görüntülenir.

**Device noktalarının toplam sayısı:**  
Device noktalarının sayısı otomatik olarak word birimine dönüştürülür.



**Device noktalarının maksimum sayısı = CPU modülünün kapasitesi**

Örneğin, R04CPU'nun CPU modülü kapasitesi 40K word.

"Device Setting" penceresi

Item	Symbol	Points	Device	Latch (1)	Latch (2)
Input	X	12K	0 to 21		
Output	Y	12K	0 to 1228		
Internal Relay	M	12K	0 to 1228	No Setting	No Setting
Link Relay	B	8K	0 to 1FFF	No Setting	No Setting
Link Special Relay	SB	2K	0 to 7FF	No Setting	No Setting
Annunciator	F	2K	0 to 2047	No Setting	No Setting
Edge Relay	V	2K	0 to 2047	No Setting	No Setting
Step Relay	S	0			
Timer	T	1K	0 to 1023		
Long Timer	LT	1K	0 to 1023		
Detecting Timer	ST	0			
		0			
		512	0 to 511		
		512	0 to 511		
		18K	0 to 18431		
		8K	0 to 1FFF		
		2K	0 to 7FF		
		8K	0 to 8191		
				No Setting	
Total Device			38.4K Word		
Total Word Device			34.5K Word		
Total Bit Device			62.0K Bit		

## 1.3

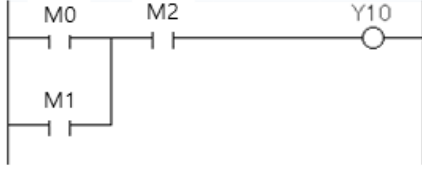
## Uygulamalar ile ilgili label adlarının kullanılması

### 1.3.1

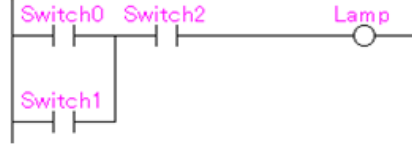
### Label kullanımının avantajları

Kontrol programlarında kullanılan device adları, bir harf ve bir sayıdan oluşmalıdır. Örneğin, "M0" ve "D5". Bir label adı, "StartSwitch" gibi bir uygulama ile ilgili olduğunda işlem hedefi belirgin hale gelir. Label adları uygulamalara göre serbestçe ayarlanabilir. Kullanıcıların, labelların kullanıldığı alanlar için device numaralarını dikkate almaları gerekmez.

Device adlarının kullanıldığı program



Labelların kullanıldığı program



Labellar, kullanım kapsamına göre aşağıdaki iki kategoriye ayrılır.

- **Global Labels**  
Global labellar, bir projedeki tüm programlar için kullanılabilir.
- **Local labels**  
Local labellar, yalnızca kayıtlı oldukları programda kullanılabilir.

Labellar gerçek (fiziksel) devicelar (X, Y) için kullanılırken device adları, GX Works3 kullanılarak global labellara atanmalıdır.

## 1.3.2

## Labellarin veri tipi

Kullanilacak deęerlerin aralıęını tanımlamak üzere her bir label için bir veri tipi belirtilmelidir. Veri tipleri, ařaęıda gsterildięi gibi bit ve tam sayı ięerir.

Veri tipi		Veri aralıęı
Bit tipi		Bit devicelerin aęık/kapalı durumu ve yrtme sonuęlarının doęru/yanlıř durumu
Tam sayı tipi	Word (unsigned)	0 – 65.535
	Word (signed)	-32.768 – 32.767
	Double word (unsigned)	0 – 4.294.967.295
	Double word (signed)	-2.147.483.648 – 2.147.483.647

Tam sayı tipini kullanırken, veri aralıęına gre word veya double word tipini seęin ve negatif deęerleri kullanma gereklilięine gre signed veya unsigned tipi seęin. Bir label adı belirlerken GX Works3 kullanarak labelın veri tipini belirtin.

Label Name	Data Type
Switch0	Bit
Data0	Word [Unsigned]/Bit String [16-bit]
Data1	Double Word [Signed]

Label deęer aralıęını belirtin.

Label ayarı penceresi

### 1.3.3

## Veri tiplerini temsil eden label adları

Transfer kaynağında ve hedefte farklı veri tiplerinin kullanılması, dönüşüm hatasına veya beklenmeyen bir sonuca neden olabilir. Aşağıda bu gibi bir duruma ilişkin bir program örneği yer almaktadır.

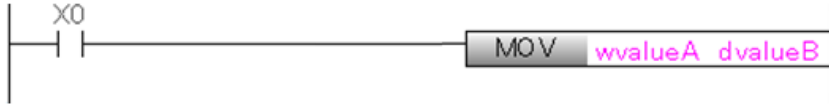


Double word değerleri bir word tipi labelına aktarılamaz. Ancak veri tipi label adıyla tanımlanamaz.

Bu nedenle, veri tipini temsil eden ön ekler, veri tiplerinin görsel olarak tanımlanabilmesi için label adlarına eklenebilir. Bu etiket adlandırma türü Macar notasyonu olarak bilinir.

Veri tipi	Veri aralığı	Ön ek	Ön ek'in uzun hali
Bit tipi	Bit devicelerinin açık/kapalı durumu ve yürütme sonuçlarının doğru/yanlış durumu	b	<b>bit</b>
Tam sayı tipi	Word (unsigned)	u	<b>unsigned word</b> (işaretsiz sözcük)
	Word (signed)	w	<b>signed word</b> (işaretlili sözcük)
	Double word (unsigned)	ud	<b>unsigned double-word</b> (işaretsiz çift sözcük)
	Double word (signed)	d	<b>signed double-word</b> (işaretlili çift sözcük)

Bu sayfanın üst kısmındaki program örneği, Macar notasyonu kullanılarak aşağıdaki gibi yazılabilir:



Macar notasyonu kullanılarak, veri tipindeki tutarsızlıklar program yazma sürecinde belirlenebilir.

Kursun geri kalanında, örneklerdeki label adları Macar notasyonu kullanılarak yazılmıştır.

## 1.3.4

## Hazırlanmış labelların kullanımı

Modül konfigürasyonu, modül konfigürasyon şemasında ayarlandığında, modül kurulum konumlarına karşılık gelen modül sinyallerini veya ayar değerlerini temsil eden labellar (modül labelları) otomatik olarak kaydedilir.

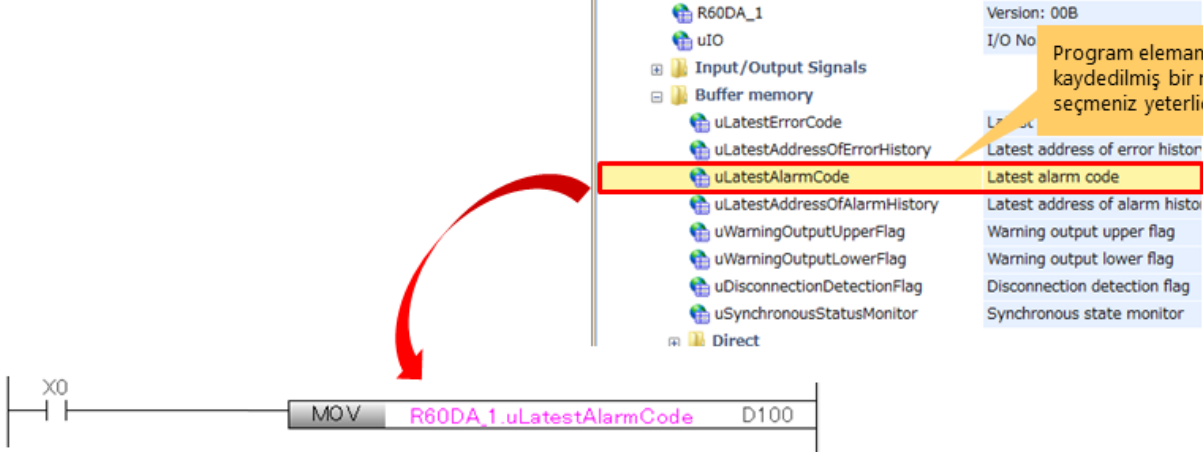
Device adları kullanarak programlama için sinyallere karşılık gelen device numaraları ve buffer memory adresleri kılavuzlardan kontrol edilmelidir ve bunu yapmak zaman alır. Programlama süresi modül labelları kullanılarak kısaltılabilir çünkü bu yöntemde kullanıcıların yalnızca listeden labelları seçmesi gerekir.

### Device adları kullanıldığında



Kurulum konumunu ve buffer adresini kontrol edin ve tanımlayın.

### Modül labelları kullanıldığında



## 1.3.5

## Labellara sabit sayılar atama

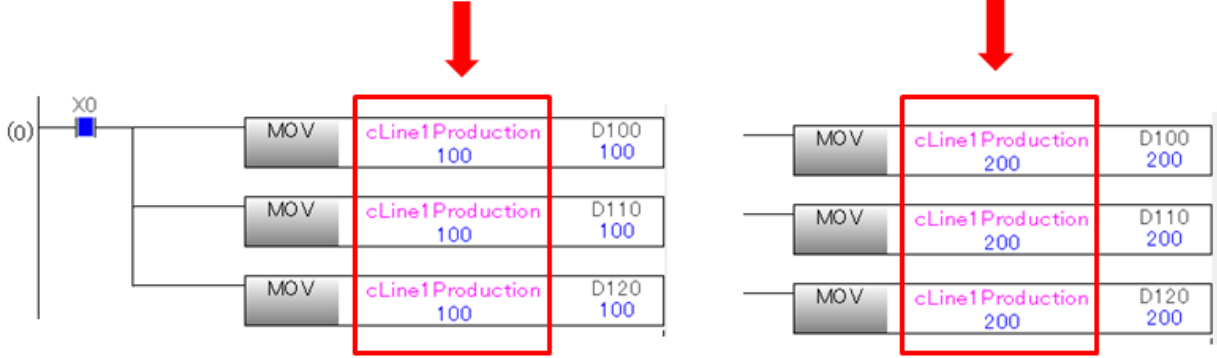
Labellara sabit sayılar atanabilir.

Labellara sabit sayılar atandığında, değerler programı modifiye etmeden değiştirilebilir.

Birden fazla label için kullanılan sabit, toplu olarak değiştirilebilir.

"cLine1Production" etiketine 100 sabitini atayın.

200 sabitini atayın.



Bir labela sabit sayı atamak için labelın uygulamasını belirten sınıfı, label ayarlama penceresinde değiştirin.

Local labellar için "VAR\_CONSTANT" sınıfını seçin.

Label Name	Data Type	Class	Initial Value	Constant
uData	Word [Unsigned]/Bit String [16-bit]	VAR_CONSTANT		100

Label uygulamasını belirtin.



## 1.4

## Program okunurluğunun iyileştirilmesi

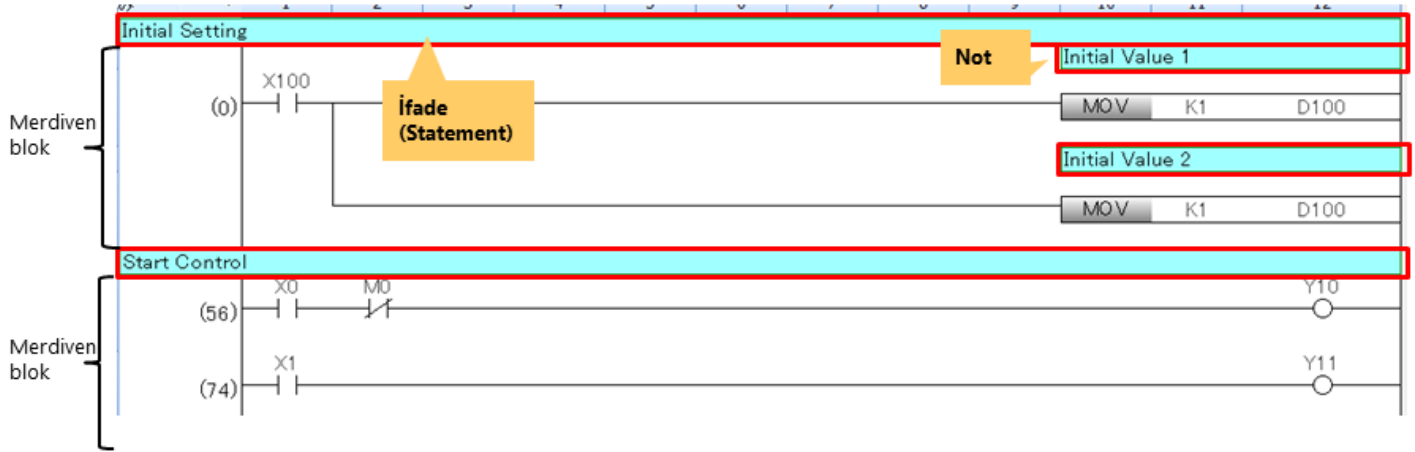
Bir programa, işlem detayları ve aygıt adları gibi yorumlar (comment) eklenebilir. Yorumlar (Comment), programın nasıl çalıştığının anlaşılmasına yardımcı olur.



Yorum tipi, programın elemanına veya aralığına göre seçilebilir.

Yukarıdaki program örneğinde, device uygulamalarının veya bağlı I/O devicelerinin label ve tiplerinin anlaşılması için "device/label comments" eklenmiştir.

Ayrıca yorum tipinde, işlem akışının anlaşılmasına yardımcı olmak için merdiven blokuna eklenen bir "ifade" ve bobinler ile uygulama yönergelerinin detaylarının anlaşılmasına yardımcı olmak için "notlar" bulunur.



Bu bölümde aşağıdakileri öğreniniz:

- I/O numarası ataması
- Bellek alanını ayarlama
- Label ile programlama
- Programlardaki yorumlar

#### Önemli noktalar

I/O numarası ataması	<ul style="list-style-type: none"><li>• I/O numaraları, CPU modülüne en yakın olandan başlanarak yuvalara otomatik olarak atanır</li><li>• Modüllere manuel olarak sabit I/O numarası atandığında, programlar farklı sistemlerde kullanılabilir</li></ul>
Device ayarları ve bellek alanı ayarları	<ul style="list-style-type: none"><li>• Device noktalarının sayısı, CPU modülüne bağlı olarak farklılık gösterir</li><li>• Kullanılmayan bellek alanları azaltılarak device noktalarının sayısı artırılabilir</li><li>• Her bir device nokta sayısı, kullanım durumuna göre değiştirilebilir</li></ul>
Label ile programlama	Label kullanıldığında işlem hedefi belirgin hale gelir
Yorumlar (Comment)	Yorumlar eklendiğinde, işlem detayları ve akışı daha kolay anlaşılır

## Bölüm 2 İleri programlama

Bu bölümde, devicelerin ve label programlamanın ileri kullanımı açıklanmaktadır.

- 2.1 Word device ın bit birimi cinsinden kullanımı
- 2.2 Device ı yalnızca kontak durumu değıştiğinde açma
- 2.3 Zamanlayıcının ölçtüğü süreyi tutma
- 2.4 Zamanlayıcının ölçüm birimini değıştirme
- 2.5 Birden fazla device ın kullanımı (index register)
- 2.6 Birden fazla değerin kullanımı (array)
- 2.7 Birden fazla değerin kullanımı (structure)
- 2.8 Device durumunu tutma (latch)
- 2.9 Device durumunu tutma (file register)
- 2.10 Önceden belirlenen işlevler ve operasyonları bulunan devicelerin kullanımı
- 2.11 Gerçek sayılarla hesaplama

## 2.1

## Word device in bit birimi cinsinden kullanımı

Data register gibi word devicelar, normalde word birimi cinsinden kullanılır ancak bit birimi cinsinden kullanılmaları da mümkündür.

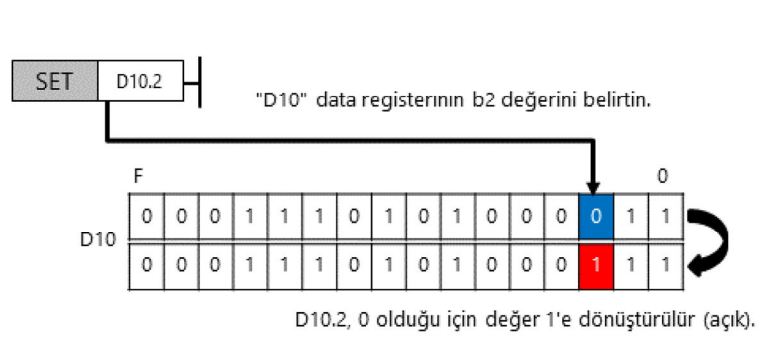
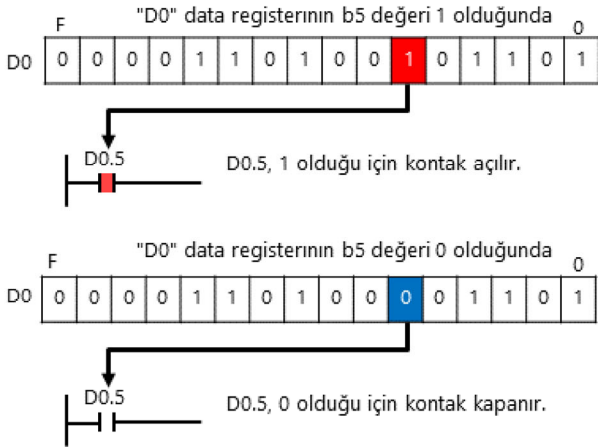
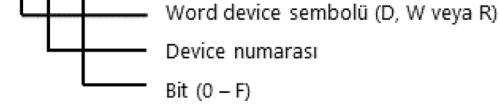
Bit birimi, data registerdaki (D) belirli bir biti belirtmek için kullanılır.

Örnek) Data register (D)

0	0	1	0	0	1	1	0	1	1	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Bit belirtim formatı

D □.□



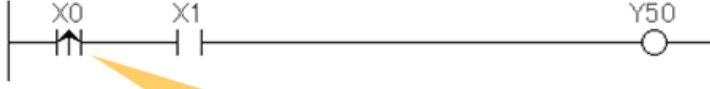
Labelları kullanmak için "uData.2" ve "uData.5" gibi bir tanımlama yapın.

## 2.2

## Device ı yalnızca kontak durumu deęiřtięinde açma

Kontaęın yükselen kenarında veya düşen kenarındaki yalnızca bir tarama için açılan bir sinyal belirlenebilir. Bu işlev, giriş koşulu tarafından yükselme ve düşüşün kontrol edilmesinde kullanışlıdır.

### Kontaęın yükselen kenar belirtimi

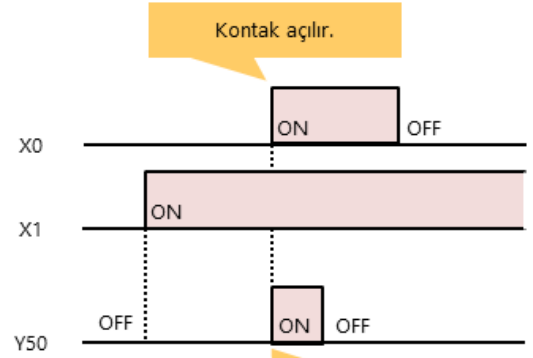


Sinyal, "X0" kontaęının açıldıęı yalnızca bir tarama için açılır.

### Kontaęın düşen kenar belirtimi

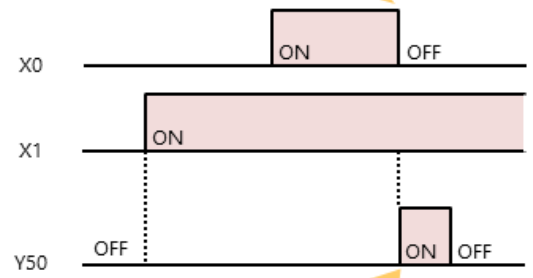


Sinyal, "X0" kontaęının kapandıęı yalnızca bir tarama için açılır.



Yalnızca bir tarama için açık

Kontak kapanır.



Yalnızca bir tarama için açık

## 2.3

## Zamanlayıcının ölçtüğü süreyi tutma

Bu bölümde, zamanlayıcı devicelerinden biri olan ve ölçülen süreyi tutabilen retentive timer açıklanmaktadır.

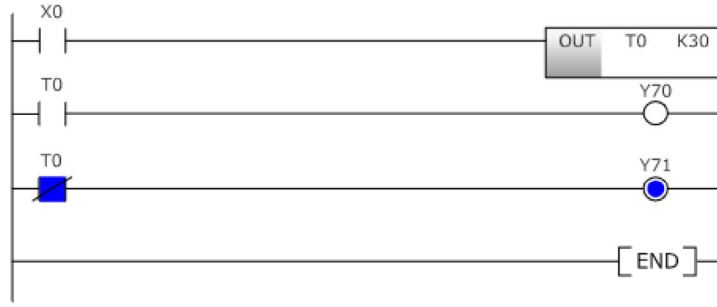
### 2.3.1

### Zamanlayıcı ile kalıcı zamanlayıcı arasındaki fark

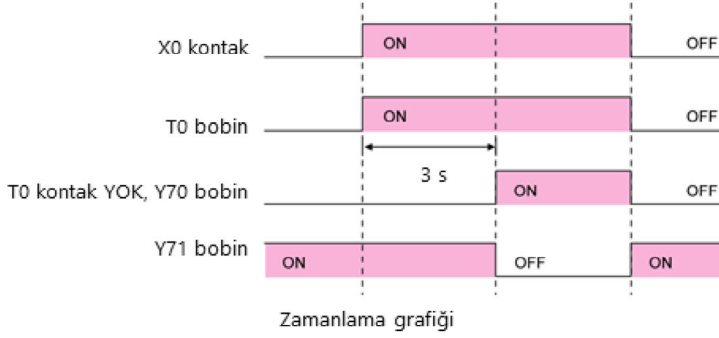
Kalıcı zamanlayıcıyı açıklamadan önce, zamanlayıcının nasıl çalıştığına bir göz atalım.

Zamanlayıcı, bobin açıldığında ölçüme başlar. Belirli bir süre geçtikten sonra zaman dolar ve kontak açılır. Bobin kapandığında ölçülen süre "0" olarak ayarlanır. Zamanlayıcının aygıt sembolü "T"dir.

Zamanlayıcının nasıl çalıştığını görmek için sağ taraftaki giriş anahtarını kullanın.



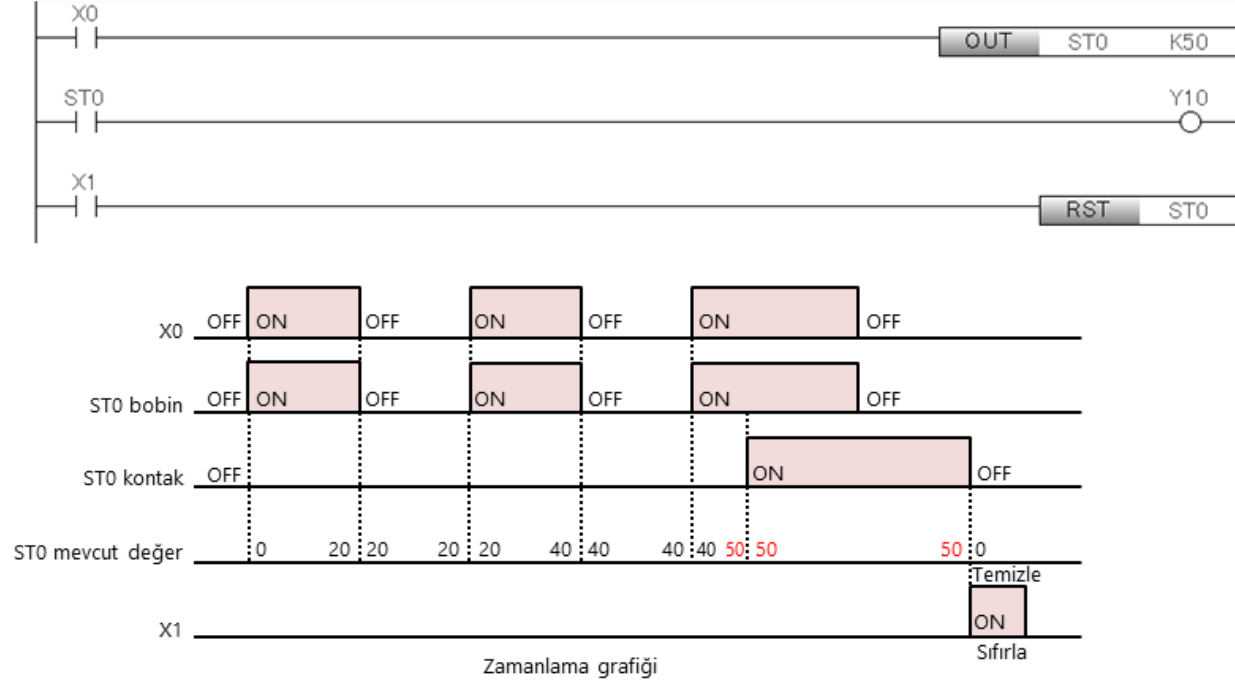
X0'ın açılmasından itibaren üç saniye geçtikten sonra, Y70 açılır ve Y71 kapanır.



## 2.3.1

## Zamanlayıcı ile kalıcı zamanlayıcı arasındaki fark

Kalıcı zamanlayıcı, toplam operasyon süresinin ölçülmesinde kullanışlıdır. Kalıcı zamanlayıcı, bobin açıldığında ölçüme başlar. Belirli bir süre geçtikten sonra zaman dolar ve kontak açılır. Bobin kapandığında ölçülen süre sıfırlanmaz. Bobin tekrar açıldığında, ölçüm tutulan değerden itibaren yeniden başlar. Kalıcı zamanlayıcının aygıt sembolü "ST"dir.



## 2.3.2

## Kalıcı zamanlayıcı için program örneği

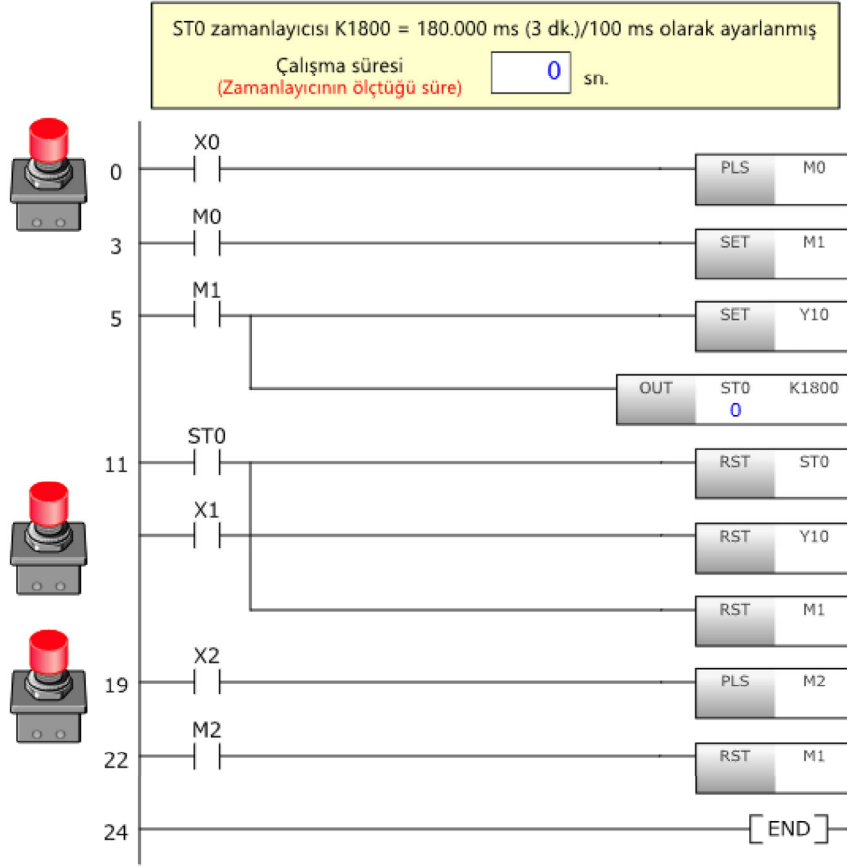
Giriş anahtarları (X0 – X2) kullanılarak çalıştırılan bir makineyi simüle ederek, kalıcı zamanlayıcının nasıl çalıştığına bir göz atalım.  
\*Kalıcı zamanlayıcı (ST0), 100 ms'lik artırımlarla ayarlanır.

1. X0 açıldığında çalışma başlar.
2. X2 açıldığında çalışma duraklar ve mevcut değer tutulur.
3. X0 tekrar açıldığında çalışma tekrar başlar.
4. X1 açıldığında çalışma sonlanır ve mevcut değer sıfırlanır.



X0 – X2: Giriş anahtarları

Y10: Başlangıç sinyali





## 2.3.3

### Kalıcı zamanlayıcı ayarı

Kalıcı zamanlayıcı tarafından kullanılan noktaların sayısı varsayılan olarak "0"dır. Kalıcı zamanlayıcıyı kullanmadan önce, GX Works3 kullanarak CPU parametresinin "Device Setting" (Aygıt Ayarı) kısmında nokta sayısını ayarlayın.

Aşağıdaki örnekte, kalıcı zamanlayıcı için 64 nokta (ST0 – ST63) ayarlanmıştır.

Item	Symbol	Device		Local Device			Latch (1)	Latch (2)
		Points	Range	Start	End	Points		
Input	X	12K	0 to 2FFF					
Output	Y	12K	0 to 2FFF					
Internal Relay	M	12K	0 to 12287				No Setting	No Setting
Link Relay	B	16K	0 to 3FFF				No Setting	No Setting
Link Special Relay	SB	16K	0 to 3FFF					
Annunciator	F	2K	0 to 2047				No Setting	No Setting
Edge Relay	V	2K	0 to 2047				No Setting	No Setting
Step Relay	S	0						
Timer	T	1K	0 to 1023				No Setting	No Setting
Long Timer	LT	1K	0 to 1023				No Setting	No Setting
Retentive Timer	ST	64	0 to 63				No Setting	No Setting
Long Retentive Time LST		0					No Setting	No Setting
Counter	C	512	0 to 511				No Setting	No Setting
Long Counter	LC	512	0 to 511				No Setting	No Setting
Data Register	D	18K	0 to 18431				No Setting	No Setting
Link Register	W	8K	0 to 1FFF				No Setting	No Setting
<b>Link Special Register SW</b>		2K	0 to 7FF					
Latch Relay	L	8K	0 to 8191					No Setting
Total Device			39.9K Word			0.0K Word		
Total Word Device			34.6K Word			0.0K Word		
Total Bit Device			84.2K Bit			0.0K Bit		

## 2.3.4

### Bir zamanlayıcıyı belirtmek için label kullanımı

Bir label, zamanlayıcı device kullanıyorsa veri tipini "Timer" (Zamanlayıcı) olarak ayarlayın.

Label Name	Data Type
uTimer1	Timer
uTimer2	Retentive Timer

Kalıcı zamanlayıcının önceki bölümde açıklandığı gibi kullanılabilmesi için device ayarı gereklidir. Ancak bu ayar labellar için gerekli değildir.

Ölçüm birimi ve süre, zamanlayıcı tipine göre farklılık gösterir.

- Yüksek hızlı zamanlayıcı (kısa ölçüm birimleri)
- Düşük hızlı zamanlayıcı (uzun ölçüm birimleri)
- Uzun süreli ölçümler yapabilen uzun zamanlayıcı

Yukarıdaki zamanlayıcılardan her birinin kalıcı zamanlayıcı işlevi vardır.

Tip	Ölçüm birimi	Program örneği	Operasyon	Mevcut değer hatırlanması
Düşük hızlı zamanlayıcı	100 ms (varsayılan)		T0 düşük hızlı zamanlayıcısı 5 saniye ölçer.	16 bit
Yüksek hızlı zamanlayıcı	10,00 ms (varsayılan)		T1 yüksek hızlı zamanlayıcısı 0,5 saniye ölçer.	
Düşük hızlı kalıcı zamanlayıcı	100 ms (varsayılan)		ST0 düşük hızlı kalıcı zamanlayıcısı 5 saniye ölçer.	
Yüksek hızlı kalıcı zamanlayıcı	10,00 ms (varsayılan)		ST1 yüksek hızlı kalıcı zamanlayıcısı 0,5 saniye ölçer.	
Uzun zamanlayıcı	0,001 ms (varsayılan)		LT0 uzun zamanlayıcısı 0,005 saniye ölçer.	32 bit
Uzun kalıcı zamanlayıcı			LST0 uzun kalıcı zamanlayıcısı 0,005 saniye ölçer.	

Başlangıçtaki ölçüm birimleri, düşük hızlı zamanlayıcı için 100 ms, yüksek hızlı zamanlayıcı için 10 ms ve uzun zamanlayıcı için 0,001 ms'dir.

Ölçüm birimini değiştirme hakkında bilgi için sonraki sayfaya bakın.

## 2.4

## Zamanlayıcının ölçüm birimini deęiřtirme

Zamanlayıcının ölçüm birimi, CPU parametresinin "Timer Limit Setting" (Zamanlayıcı Limit Ayarı) kısmında deęiřtirilebilir.

<b>Timer Limit Setting</b>	
Low Speed Timer/Low Speed Retentive Timer	100 ms
High Speed Timer/High Speed Retentive Timer	10.00 ms
Long Timer/Long Retentive Timer	0.001 ms

## 2.5

### Birden fazla device ın kullanımı (index register)

İndeks kaydı (Z), kontrol-hedef aygıtının aygıt numarasını dolaylı bir şekilde belirtmek (değiştirmek) için başka bir aygıtla birlikte kullanılır. İndeks kaydı, bir gruptaki birden fazla device ı tanımlayabildiği için programların basitleştirilmesinde kullanışlıdır.

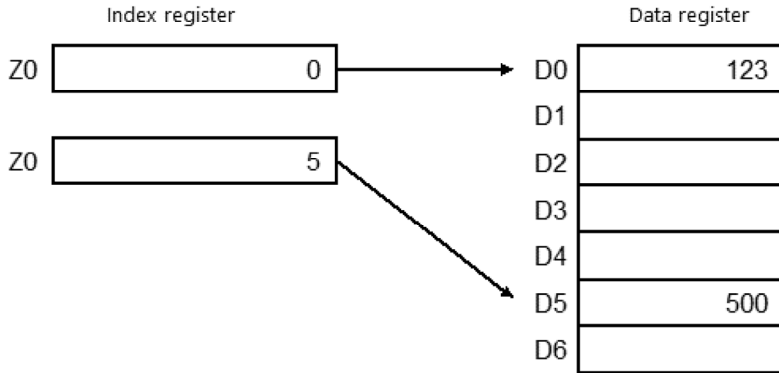
- İndeks kaydı, bir device sembolü ve bir device numarası ile tanımlanır
- Gerçek kontrol hedef device numarası = Device sembolü (device numarası + indeks kaydı)
- İndeks kaydı için device noktalarının sayısı varsayılan olarak 20 noktadır (Z0 – Z19)

### 2.5.1

#### İndeks kaydı için uygulama örneği

Bir device "D0Z0" olarak tanımlandığında bu, D(0+Z0) anlamına gelir.

Örnek) Z0 değeri 0 olduğunda device numarası D0'dır.  
Z0 değeri 5 olduğunda device numarası D5'tir.



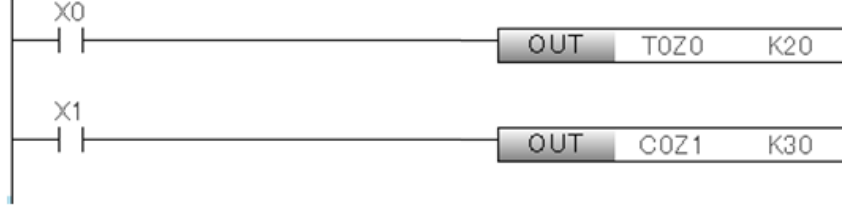
## 2.5.2

## İndeks kaydı ile modifiye edilebilen devicelar

İndeks kaydı ile modifiye edilebilen devicelar arasında aşağıdakiler yer alır:

Bit device	X, Y, M, L, S, B, F
Word device	T, C, D, R, W
Sabit	K, H
İşaretçi (Pointer)	P

Not) Zamanlayıcılarda ve sayılarda kullanılan kontaklar ve bobinler için yalnızca "Z0" veya "Z1" indeks kaydı kullanılabilir.



Z0 değeri 1 olduğunda T1 süreyi ölçer.

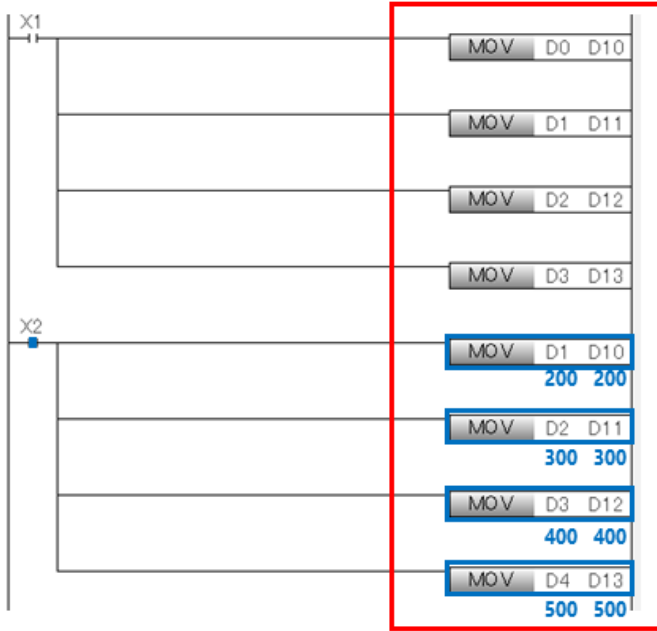
Z1 değeri 5 olduğunda C5 sayımı gerçekleştirir.

## 2.5.3

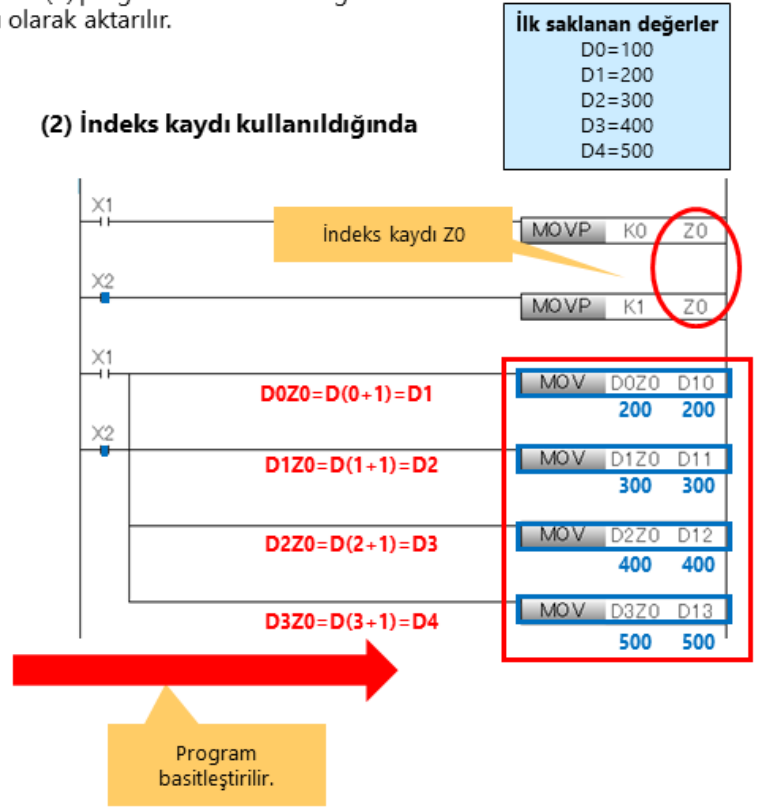
## Programların indeks kaydı kullanılarak basitleştirilmesi

Aşağıda gösterilen programlarda, X1 veya X2 açıldığında "D0 – D4" aralığındaki değerler "D10 – D13" aralığına aktarılmaktadır. (1) ve (2) programları aynı sonucu getirir. (1) programında veriler doğrudan aktarılır. (2) programında veriler, indeks kaydı üzerinden dolaylı olarak aktarılır.

### (1) İndeks kaydı kullanılmadığında



### (2) İndeks kaydı kullanıldığında



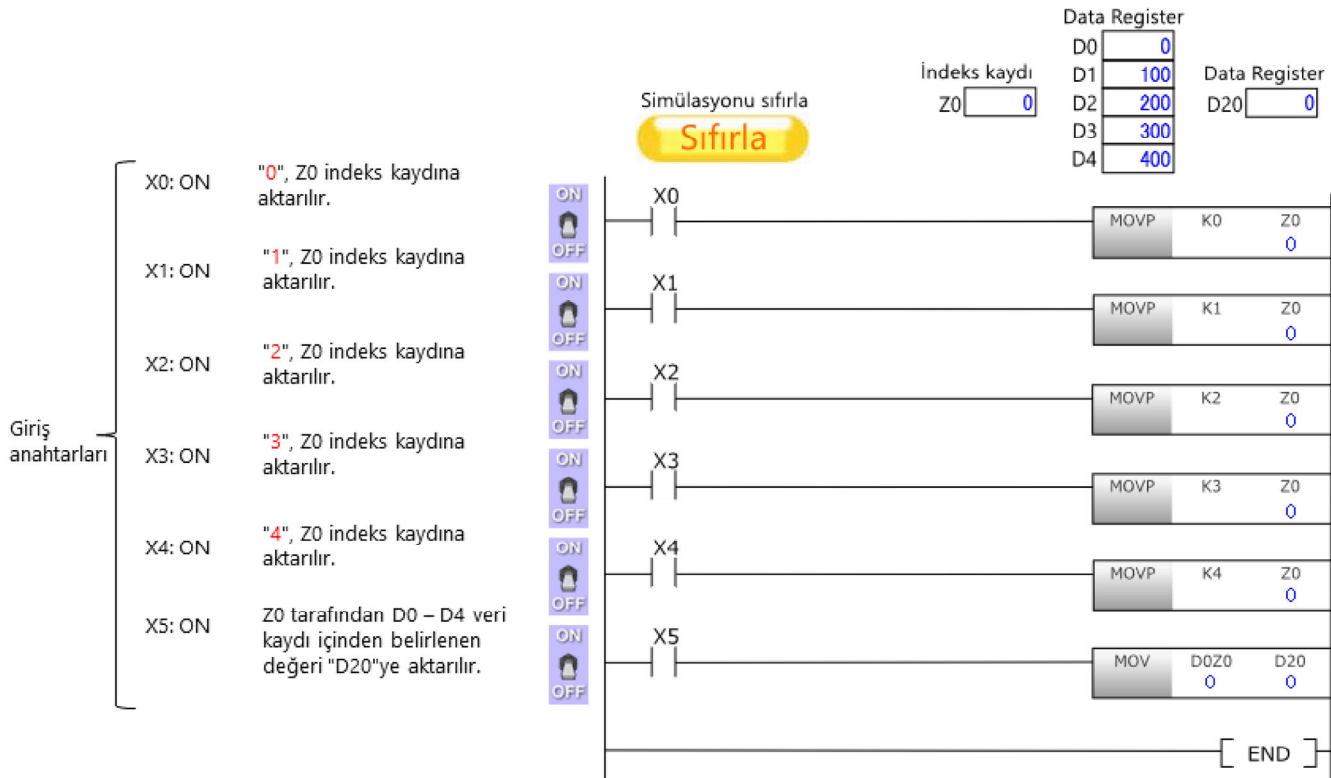
## 2.5.4

## İndeks kaydı için program örneği

Z0 indeks kaydının çalışma şekli, X0 – X5 giriş anahtarları açılarak simüle edilebilir.

\*K0 – K400, D0 – D4 data registerında halihazırda saklanmıştır.

Her bir device alanında saklanan değerleri kontrol etmek için X0 – X5 giriş anahtarlarını açın.








## 2.6

## Birden fazla deęerin kullanımı (array)

Array kullanılarak, birden fazla deęer bir label adı tarafından kullanılabilir.

Ařađıdaki örnekte, bir otomobil üretim fabrikasındaki üretim hacmi verileri hedefe göre saklanmıřtır.

Hedef			
Üretim hacmi	35 birim	75 birim	65 birim

Hedefe göre üretim hacmi verileri bir labela atanır.

Array kullanılmadıęında, her bir hedef için label adları oluřturulmalıdır.

Ancak array kullanarak, birden fazla hedefin üretim hacmi, tek bir label adına atanabilir ve bu label adında saklanabilir.

Array kullanılmadıęında

```
uProductionA  
uProductionB  
uProductionC
```

Array kullanıldıęında

```
uProduction
```

Array içindeki ayrı labelar, eleman numaraları kullanılarak belirtilir. Eleman numaraları [0]'dan bařlar.

```
uProduction [0]
```

Label adı

Eleman numarası

Hedef (satır)

Country A	[0]	35
Country B	[1]	75
Country C	[2]	65

Ařađıdaki program örneęinde, Country A için planlanan üretim hacmi, bařka bir etikete aktarılmıřtır.

```
MOV    uProduction[0]    uShowProductionPlan
```



## 2.7

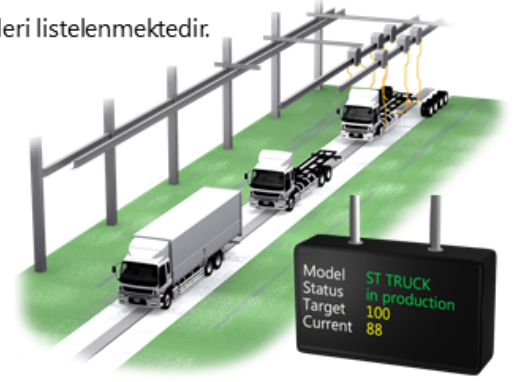
## Birden fazla değerin kullanımı (structure)

Structure kullanılarak, birden fazla ilgili label bir label adı tarafından kullanılabilir.

Aşağıdaki örnekte, bir otomobil üretim hattının durumu Andon'da (görüntü panosu) görüntülenmektedir.

Aşağıdaki tabloda, görüntülenen öğelere karşılık gelen label adları, değerler ve veri tipleri listelenmektedir.

Öğe	Etiket adı	Değer	Labelın veri tipi
Model	sModel	'ST TRUCK'	Dizi (String) tipi
Operasyon durumu	bStatus	'in production'	Bit tipi
Bugünün hedef üretim hacmi	uPlanQty	'100' birim	Tam sayı tipi (word, unsigned)
Mevcut üretim hacmi	uActualQty	'88' birim	Tam sayı tipi (sözcük, işaretsiz)



Birden fazla üretim hattı bulunan bir fabrika için yapı kullanılmıyorsa her bir hat için label adının değiştirilmesi gerekir.

Aşağıda, üretim hattı adlarının eklendiği label adlarına örnekler verilmektedir.

İlk üretim hattı

```
s1stLineModel  
b1stLineStatus  
u1stLinePlanQty  
u1stLineActualQty
```

İkinci üretim hattı

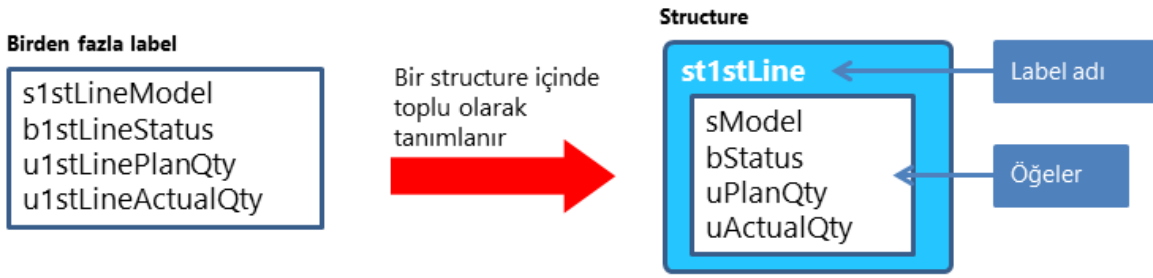
```
s2ndLineModel  
b2ndLineStatus  
u2ndLinePlanQty  
u2ndLineActualQty
```



Üretim hatlarının sayısı arttıkça, kullanılacak label sayısı da artar. Sonuç olarak, program daha uzun ve daha zor okunur hale gelir.

Structure, bir label adının bir üretim hattı ile ilgili birden fazla labelı temsil etmesini sağlar.

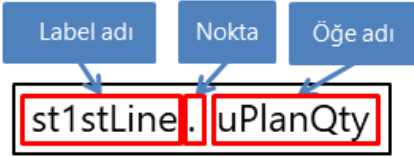
Dolayısıyla structure, fiziksel nesnelere veya devicelere, ekipman ve iş parçaları gibi cisimlerle ilgili koşullar ve özelliklerin toplu olarak organize edilmesi, saklanması ve kullanılması için kullanılır.



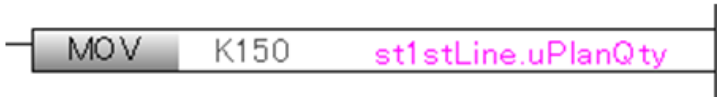
Structure labeları, **structure** (yapı) sözcüğünü temsil eden "st" ön ekini alır.

Structure tarafından tanımlanan ayrı labeler öğeler olarak adlandırılır. Her bir öğenin veri tipleri farklı olabilir.

Bir structure içindeki her bir öğeyi belirlemek için yapı etiketinin sonuna, aralarında sınırlayıcı olarak nokta (.) kullanarak üye adını ekleyin.



Aşağıdaki program örneğinde, ilk üretim hattı için structure tipi labelın bir öğesine bir sabit atanmıştır.



## 2.8

### Device durumunu tutma (latch)

Bu bölümde, CPU modülü çalışmayı durdurduğunda device değerlerini tutan latch işlevi açıklanmaktadır. Örneğin, izin verilen anlık elektrik kesintisi süresini aşan bir elektrik kesintisi gerçekleştiğinde programlanabilir kontrolör, operasyon durduğunda tutulan verileri kullanarak sıra kontrolünü yeniden başlatabilir.

Latch işlevi kullanılmıyorsa aşağıdaki durumlarda device değerleri temizlenir ve ilk değerlere (bit aygıtları için kapalı ve sözcük aygıtları için "0") sıfırlanır:

- Güç kesintisi
- RUN/STOP/RESET (ÇALIŞTIRMA/DURDURMA/SIFIRLAMA) anahtarıyla sıfırlama
- İzin verilen anlık elektrik kesintisi süresini aşan elektrik kesintisi

### 2.8.1

#### Devicelerde latch ayarlama

CPU parametresinin "Device Setting" (Aygıt Ayarı) penceresinde latch aralığını ayarlayın. Aşağıda, D0 – D128 veri kaydı için bir latch ayarı örneği verilmektedir.

Latch (1)		Latch (2)		
No.	Device	Points (Decimal)	Start	End
1	D	129	0	128
2				

Latch olarak ayarlanacak device

Latch deviceinin başlangıç numarası

Latch device in bitiş numarası

## 2.8.2

## Latch tipleri ve temizleme yöntemleri

Temizleme yöntemlerine göre iki tip latch (latch (1) ve latch (2)) bulunur.

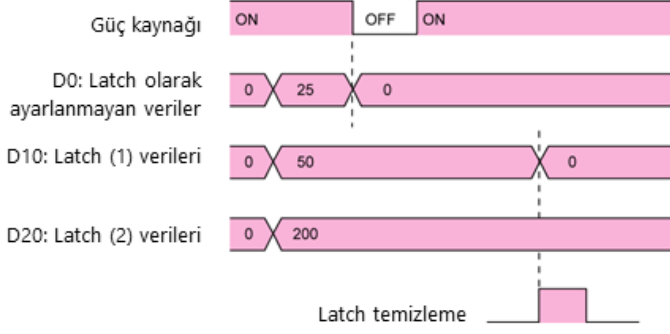
- Latch (1)

Latch olarak ayarlanan veriler, GX Works3'ün CPU belleği operasyon işlevi\* kullanılarak temizlenebilir. Kurulum tesisinde latch olarak ayarlanan verilerin temizlenmesi gerektiğinde latch 1'i kullanın.

- Latch (2)

Latch olarak ayarlanan veriler, programdaki bir yönerge kullanılarak temizlenebilir. Kurulum tesisinde latch olarak ayarlanan veriler temizlenmeyecekse latch 2'yi kullanın.

Aşağıda, latch temizleme işleminin zamanlama grafiği yer almaktadır.



No.	Device	Points (Decimal)	Start	End
1	D	129	0	128
2				

\*GX Works3 menüsünden [Online] → [CPU Memory Operation] ögesini seçerek işlevi çalıştırın.

## 2.8.3

### Labellarda latch ayarlama

Bir labelda latch ayarlamak için label ayarlama penceresinde "RETAIN" ifadesini içeren bir sınıf adı seçin. Local labellar için "VAR\_RETAIN" sınıfını seçin.

Label Name	Data Type		Class
uData	Word [Unsigned]/Bit String [16-bit]	...	VAR_RETAIN ▼
		...	▼

## 2.9

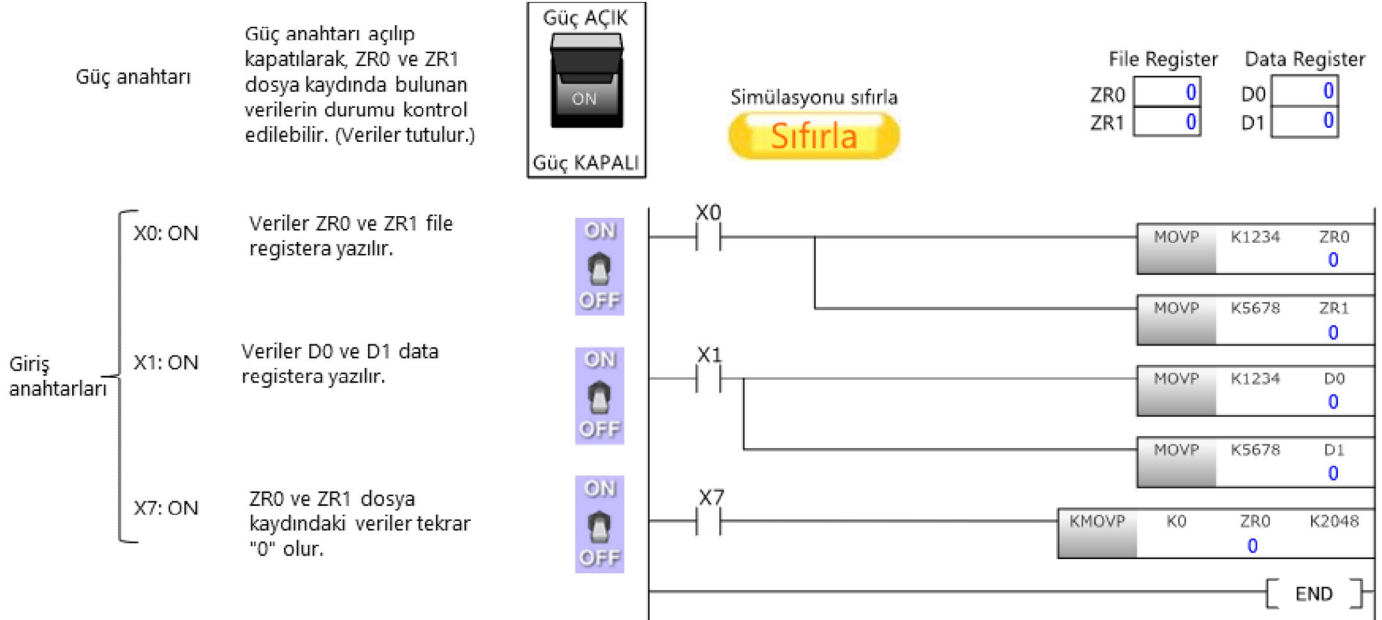
## Device durumunu tutma (file register)

- File register, data registerını (D) genişletmek için kullanılan bir word devicedir.
- File register, data registera kıyasla daha büyük miktarda veri bulundurabilir
- File register, CPU modülünün veri belleğinde veya bir bellek kartında saklanır
- File registerda saklanan veriler, sistem gücü kesildiğinde veya CPU modülü sıfırlandığında bile tutulur. File register, standart değerler gibi önceden tanımlanmış değerlerin saklanması için kullanışlıdır
- Verileri temizlemek için file register aralığına 0 yazın veya belleği GX Works3'ten temizleyin
- Device sembolü "ZR"dir

### 2.9.1

### Merdiven programının çalışması

File registerın çalışma şekli, giriş anahtarlarını ve güç anahtarını açarak simüle edilebilir.



## 2.9.2

## File register ayarı

Bu bölümde, bir local file register her bir programın saklama hedefi olarak belirleyen ayar açıklanmaktadır. CPU parametresinde "File Register Setting" ögesini seçin ve "Use File Register of Each Program" ögesini seçin.

Item	Setting
<b>File Register Setting</b>	
Use Or Not Setting	Use File Register of Each Program
Capacity	
File Name	

Programlanabilir kontrolöre veri yazılabilmesi için file register her bir programa yazılmalıdır.

The screenshot shows the "Online Data Operation" software interface. The main window has a menu bar with "Display", "Setting", and "Related Functions". Below the menu bar are several icons for "Write", "Read", "Verify", and "Delete". There are also buttons for "Parameter + Program(E)", "Select All", "Open/Close All(I)", and "Deselect All(N)". A legend indicates "CPU Built-in Memory", "SD Memory Card", and "Intelligent Function Module". The main area is a table with columns for "Module Name/Data Name", "Detail", and "Title". The "File Register" row is highlighted in red. A red arrow points from the "Detail" button of the "File Register" row to the "File Register Detail Setting" dialog box. The dialog box has two radio buttons: "Whole Range" and "Specify Range". The "Specify Range" option is selected, and the "ZR" field is set to "0" and "32767". There are "Default", "OK", and "Cancel" buttons. A yellow callout box with a speech bubble says "Yazılacak file register aralığını ayarlayın." (Set the file register range to be written).

Module Name/Data Name	Detail	Title
Memory Card Parameter		
Remote Password	<input type="checkbox"/>	
Global Label	<input type="checkbox"/>	
Global Label Setting	<input type="checkbox"/>	
Program	<input type="checkbox"/>	Detail
MAIN	<input type="checkbox"/>	
Device Memory	<input type="checkbox"/>	
MAIN	<input type="checkbox"/>	Detail
File Register	<input type="checkbox"/>	Detail
MAIN	<input type="checkbox"/>	2018/05/08 10:35:52 Not Calculated
Common Device Comment	<input type="checkbox"/>	
COMMENT	<input type="checkbox"/>	2018/05/08 10:35:52 Not Calculated



## 2.10

### Önceden belirlenen işlevler ve operasyonları bulunan devicelerin kullanımı

CPU modülünde kullanılan özel röle ve özel kaydın önceden belirlenen işlevleri ve operasyonları bulunur. Özel röle (SM), bit bilgileri (açık/kapalı) için kullanılan dahili röle olup özel kayıt (SD) ise sözcük bilgileri için kullanılan dahili kayıttır.

### 2.10.1

### Özel röle (Special relay) ve özel kayıt tipleri (Special register)

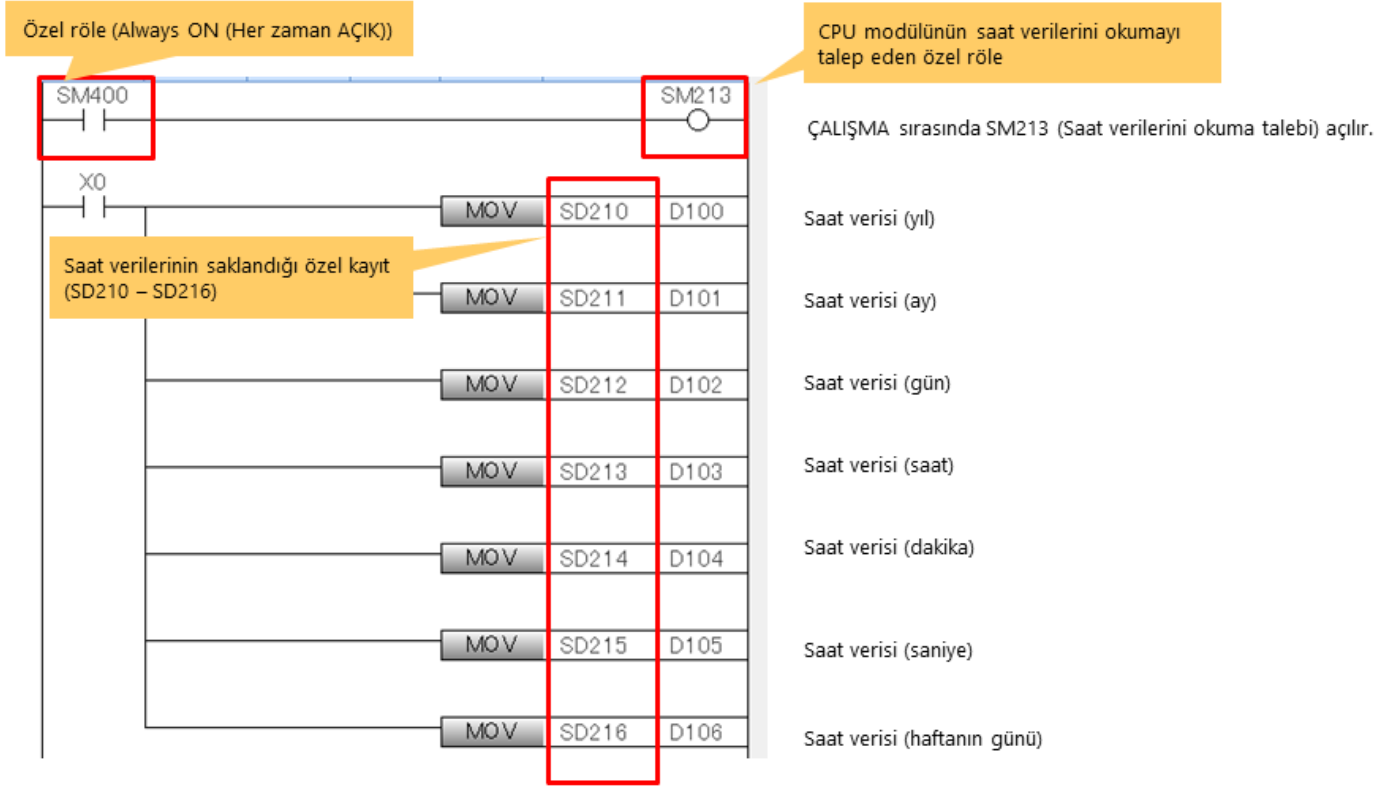
Özel röle ve özel kayıt, bilgi tiplerine göre kategorize edilir. Başlıca tipler aşağıda listelenmektedir. Kullanıcı programlarında, özel röle ve özel kayıt, kontrol için belirleme koşulları olarak kullanılır. Bunlar aynı zamanda, GX Works3'ün aygıt izleyicisinde gerçekleştirilebilen operasyon izleme işlemi için de kullanılır.

Diagnostic bilgileri
CPU modülünün diagnostic sonuçları
Diagnostic hataları ve hata kodları
Sistem bilgileri
CPU modülünün sistem bilgileri
CPU modülünün işletim durumu, saat verileri ve diğer bilgiler
Sistem saati
Temel zamanlama elemanları olarak kullanılan saat sinyalleri ve sayma değerleri
Çeşitli saat sinyalleri (belirtilen aralıklarda her zaman ON (AÇIK), ON/OFF (AÇIK/KAPALI) ve diğer sinyaller)

## 2.10.2

## Özel röle ve özel kayıt için program örneği

Aşağıda, özel röle ve özel kayıt kullanılarak CPU modülünün saat verilerini okumaya ilişkin bir program örneği verilmektedir.



## 2.10.3

## Özel röle ve özel kayıt için label kullanımı

Özel röle ve özel kayıt, CPU modülünde modül labelları olarak hazırlanır. Bunlar, kılavuzdaki device numaraları kontrol edilmeden, sadece ilgili label adları seçilip yerleştirilerek kullanılabilir.

The screenshot shows the 'Element Selection' dialog box in the SIMATIC Manager. The 'Module Label' section is expanded, showing a list of labels for the '3E00:R04CPU' module. The 'bAlways\_ON' label is highlighted in yellow. A red arrow points from the 'bAlways\_ON' label in the list to the 'RCPU.stSM.bAlways\_ON' label in the main configuration window. A yellow callout box points to the 'bAlways\_ON' label with the text 'SM400'ün modül labelı (Always ON (Her zaman AÇIK))'.

Label Name	Description
bAlways_ON	Always ON <SM400>
bAlways_OFF	Always OFF <SM401>
bAfter_RUN1_Scan_ON	After RUN ON at 1scan only <SM402>
bAfter_RUN1_Scan_OFF	After RUN ON at 1scan only <SM403>

## 2.11

## Gerçek sayılarla hesaplama

### 2.11.1

### Gerçek sayıların uygulanması

- "Gerçek sayılar", ondalık hanesi bulunan sayısal değerlerdir
- Kontrol programlarında genellikle tam sayılar kullanılır. Bununla birlikte, trigonometri işlevi ve üstel işlem gibi ileri operasyon kontrolüne yönelik programlarda gerçek sayılar kullanılır çünkü bu gibi programlarda, ondalık hanesi bulunan sayısal değerlerin kullanılması gerekir
- CPU modülünde kullanılan gerçek sayıların sayısal değerleri "floating-point data" olarak adlandırılır

#### Not:

- Bir gerçek sayı, sayıdan bağımsız olarak **daima iki ardışık word device kullanır** (32 bit bellek alanı)\*
- Kontrol programlarında, gerçek sayıları kullanan **özel çalışma yönergeleri** (toplama, çıkarma, çarpma, bölme ve özel işlevler gibi) hazırlanır. Tam sayılar ile gerçek sayılar arasında dönüşüme ilişkin yönergeler de hazırlanır

\*Daha çok sayıda anlamlı basamak içeren daha hassas bir hesaplama gerekiyorsa dört word device kullanın.

- İki word device kullanan gerçek sayılar, tek hassasiyetli gerçek sayılar olarak adlandırılır
- Dört adet word register kullanan gerçek sayılar, çift hassasiyetli gerçek sayılar olarak adlandırılır

Bu kursta tek hassasiyetli gerçek sayılara odaklanılmıştır.

## 2.11.2

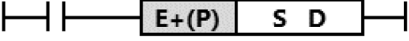
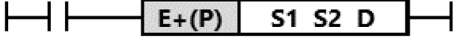
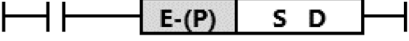
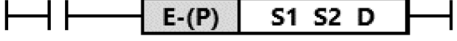
## Gerçek sayılar için notasyon

"E", gerçek sayıları temsil etmek için kullanılır.

### Bir sabiti gerçek sayılarla ifade etme

Bir sabiti yazmak için "E" ile başlayın.

Normal ifade	Sayısal bir değeri olduğu gibi yazın. (Örnek) 10,2345 değerini "E10.2345" olarak yazın.
Üstel ifade	Sayısal bir değeri "(sayısal değer) $\times 10^n$ " şeklinde yazın. (Örnek) 1234,0 değerini "E1.234+3" olarak yazın.

Yönerge sembolü	Merdiven örneği	
E+ (Tek hassasiyetli gerçek sayıları toplama)	 Gerçek sayı işlemi " $D + S = D$ " yürütülür.	 Gerçek sayı işlemi " $S1 + S2 = D$ " yürütülür.
E- (Tek hassasiyetli gerçek sayıları çıkarma)	 Gerçek sayı işlemi " $D - S = D$ " yürütülür.	 Gerçek sayı işlemi " $S1 - S2 = D$ " yürütülür.

S (source): İşlemden önceki veriler (sabit, aygıt numarası)

D (Destination): İşlemden sonraki veri hedefi (device numarası)

P: Yükselen kenarda yürütülecek yönerge (kapalıdan açığa)

S1 ve S2: İşlem yapılacak iki veri ögesi.

#### Not:

Tam sayılar ve gerçek sayılar bir işlemde karıştırılamaz.

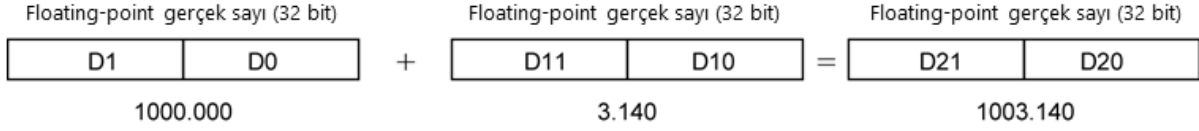
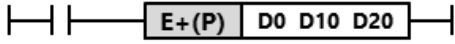
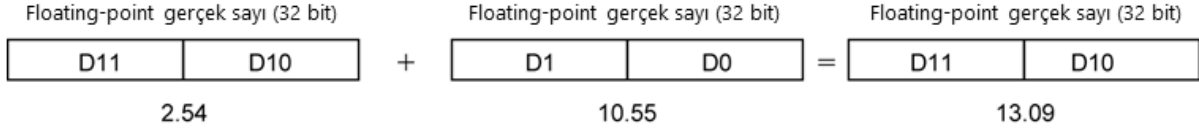
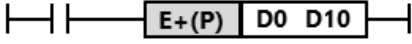
Tek hassasiyetli gerçek sayı işlemlerinde, işlem ifadesinde yer alan S, S1 ve S2 tek hassasiyetli gerçek sayılar olmalıdır.

Tek hassasiyetli gerçek sayılar D içinde saklanır.

### 2.11.3

### Çalıştırma yönergeleri (toplama ve çıkarma)

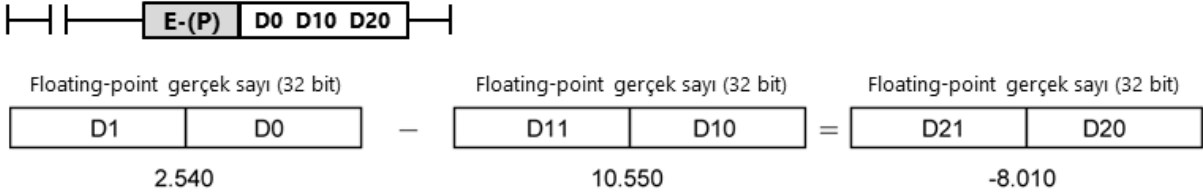
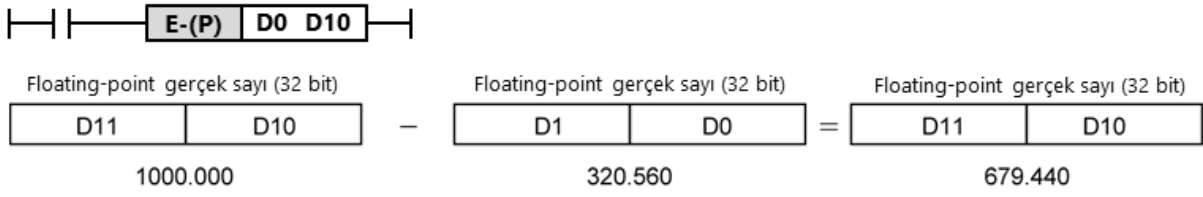
Toplama yönergeleri için uygulama örneği



## 2.11.3

## Çalıştırma yönergeleri (toplama ve çıkarma)

Çıkarma yönergeleri için uygulama örneği





## 2.11.4

## Çalıştırma yönergeleri (çarpma ve bölme)

Yönerge sembolü	Merdiven örneği
E* (Tek hassasiyetli gerçek sayıları çarpma)	 Gerçek sayı işlemi "S1 * S2 = D" yürütülür.
E/ (Tek hassasiyetli gerçek sayıları bölme)	 Gerçek sayı işlemi "S1 / S2 = D" yürütülür.

S1, S2 (kaynak): İşlem yapılacak iki veri ögesi.

D (destination): İşlemden sonraki veri hedefi (device numarası)

P: Yükselen kenarda yürütülecek yönerge (kapalıdan açığa)

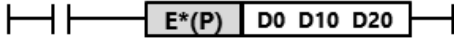
### Not:

Tek hassasiyetli gerçek sayı işlemlerinde, işlem ifadesinde yer alan S1 ve S2 tek hassasiyetli gerçek sayılar olmalıdır. Tek hassasiyetli gerçek sayılar D içinde saklanır.

## 2.11.4

## Çalıştırma yönergeleri (çarpma ve bölme)

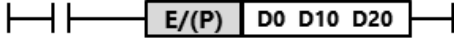
### Çarpma yönergesi için uygulama örneği



Floating-point gerçek sayı (32 bit)      Floating-point gerçek sayı (32 bit)      Floating-point gerçek sayı (32 bit)



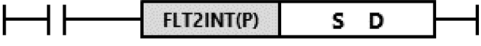

D1	D0	×	D11	D10	=	D21	D20
1000.000			25.590			25590.000	

### Bölme yönergesi için uygulama örneği



Floating-point gerçek sayı (32 bit)      Floating-point gerçek sayı (32 bit)      Floating-point gerçek sayı (32 bit)

D1	D0	÷	D11	D10	=	D21	D20
1000.000			25.590			39.078	

Yönerge sembolü	Merdiven örneği	
INT2FLT (Tam sayıyı tek hassasiyetli gerçek sayıya dönüştürme)	<p>Bir tam sayı (16 bit) bir gerçek sayıya (32 bit) dönüştürülür.</p>  <p>S (16 bit) dönüştürülür ve D içinde saklanır.</p>	<p>Bir tam sayı (32 bit) bir gerçek sayıya (32 bit) dönüştürülür.</p>  <p>S (32 bit) dönüştürülür ve D içinde saklanır.</p>
FLT2INT (Tek hassasiyetli gerçek sayıyı tam sayıya dönüştürme)	<p>Bir gerçek sayı (32 bit) bir tam sayıya (16 bit) dönüştürülür.</p>  <p>S dönüştürülür ve D (16 bit) içinde saklanır.</p>	<p>Bir gerçek sayı (32 bit) bir tam sayıya (32 bit) dönüştürülür.</p>  <p>S dönüştürülür ve D (32 bit) içinde saklanır.</p>

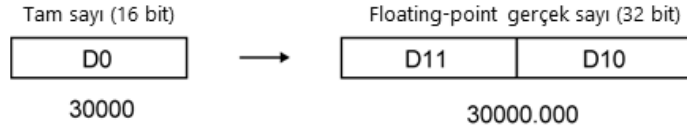
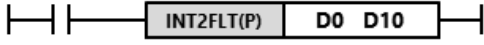
S (source): İşlemden önceki veriler (sabit, device numarası)

D (destination): İşlemden sonraki veri hedefi (device numarası)

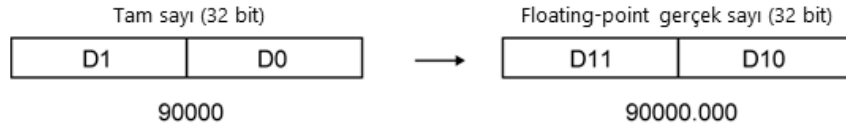
## 2.11.5

## Tam sayılar ile gerçek sayılar arasında dönüşüme ilişkin yönergeler

Tam sayıyı (16 bit) gerçek sayıya (32 bit) dönüştürme yönergesi için uygulama örneği



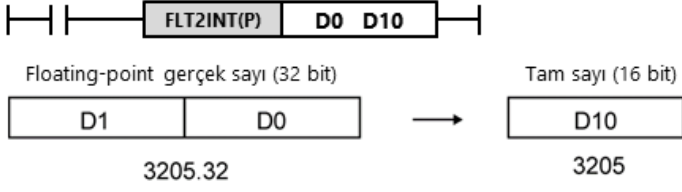
Tam sayıyı (32 bit) gerçek sayıya (32 bit) dönüştürme yönergesi için uygulama örneği



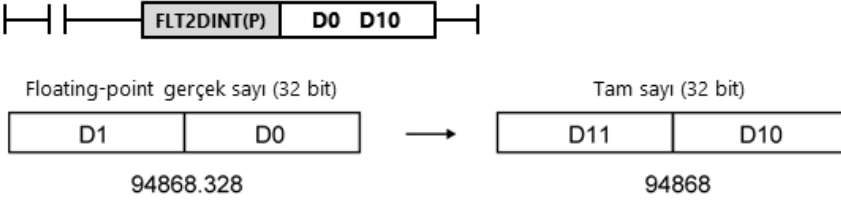
## 2.11.5

## Tam sayılar ile gerçek sayılar arasında dönüşüme ilişkin yönergeler

Gerçek sayıyı (32 bit) tam sayıya (16 bit) dönüştürme yönergesi için uygulama örneği



Gerçek sayıyı (32 bit) tam sayıya (32 bit) dönüştürme yönergesi için uygulama örneği



## 2.11.6

### Gerçek sayıları temsil eden etiketlerin kullanımı

Gerçek sayılar için label kullanmak üzere, labelları ayarlama penceresinde veri tipini "Single Precision" (Tek Hassasiyetli) veya "Double Precision" (Çift Hassasiyetli) olarak ayarlayın.

Label Name	Data Type
eData	FLOAT [Single Precision]
leData	FLOAT [Double Precision]

Bu bölümde aşağıdakileri öğrendiniz:

- Word devicein bit belirtimi
- Kontaklar için yükselen veya düşen kenar belirtimi
- Kalıcı zamanlayıcı
- Zamanlayıcının ölçüm birimi belirtimi
- İndeks kaydı
- Array
- Structure
- Latch
- File register
- Özel röle, özel kayıt (Special relay, special register)
- Gerçek sayılarla hesaplama

#### Önemli noktalar

Kalıcı zamanlayıcı	Ölçülen süre, bobin kapansa bile tutulur ve ölçüm bobin tekrar açıldığında devam eder.
Zamanlayıcının ölçüm birimi	Zamanlayıcının ölçüm birimi, parametre içinde değiştirilebilir.
İndeks kaydı	Bir grup içinde birden fazla device tanımlanabilir.
Array	Birden fazla değer bir label adı tarafından kullanılabilir.
Structure	Birden fazla ilgili label bir label adı tarafından kullanılabilir.
Latch	<ul style="list-style-type: none"> <li>• CPU modülü çalışmayı durdurduğunda latch olarak ayarlanan device değerleri tutulur</li> <li>• Latch olarak ayarlanan devşce değerleri, CPU bellek operasyonu tarafından veya bir program yönergesi kullanılarak temizlenir</li> </ul>
File register	<ul style="list-style-type: none"> <li>• File register, data registera kıyasla daha büyük miktarda veri bulundurabilir</li> <li>• CPU modülü çalışmayı durdurduğunda device değerleri tutulur</li> <li>• Device değerleri, CPU bellek operasyonu tarafından veya device aralığına 0 yazarak temizlenebilir</li> </ul>
Özel röle, özel kayıt (Special relay, special register)	Tanımlama bilgileri ve sistem bilgileri gibi CPU modülünün dahili durumuna ilişkin bilgiler bu aygıtlarda halihazırda saklanmıştır.
Gerçek sayı	<ul style="list-style-type: none"> <li>• En az iki word device kullanır (32 bit)</li> <li>• Özel çalıştırma yönergeleri sağlanır</li> <li>• Tam sayılar ve gerçek sayılar bir işlemde karıştırılmaz</li> </ul>

## **Bölüm 3** Verimli hata ayıklama

Bu bölümde, GX Works3'ün verimli hata ayıklamaya yönelik işlevleri açıklanmaktadır.

- 3.1 Program aralığını geçici olarak değiştirme
- 3.2 Device değerlerini değiştirirken operasyonu kontrol etme
- 3.3 Program operasyonunu simüle etme



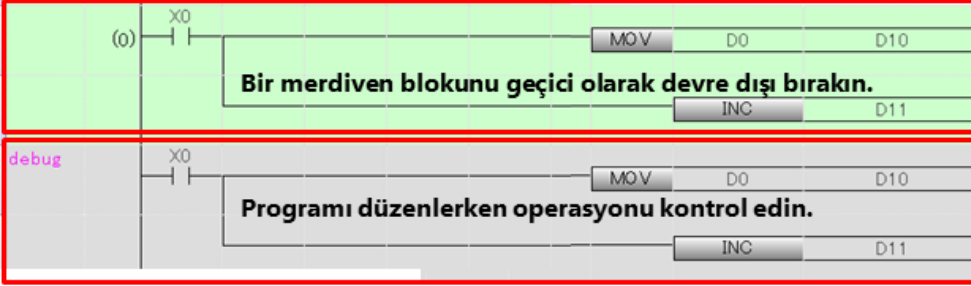
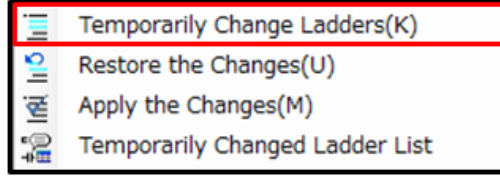
## 3.1

# Program aralığını geçici olarak değiştirme

Hata ayıklama için geniş bir program aralığı düzenlenirken, tüm değişikliklerin geri alınması oldukça zordur. Kullanıcılar, istenen merdiven blokunu geçici olarak devre dışı bırakarak, hata ayıklama için programın orijinalini değiştirmeden bir kopyasını kullanabilirler.

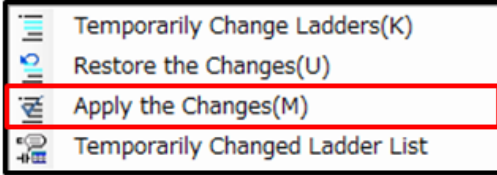
(Temporarily change ladders (Merdivenleri geçici olarak değiştir) işlevi)

Merdiven bloku değiştirildikten ve operasyon bu işleyle kontrol edildikten sonra, sorun yoksa değişiklikler uygulanır veya bir sorun varsa değişiklikler geri yüklenir.

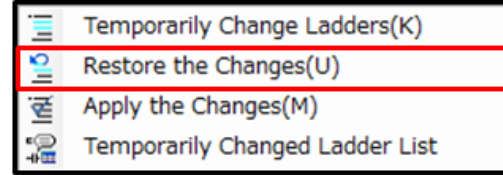


Devre dışı bırakılan merdiven blokunu kopyalayın.

Sorun bulunmadığında



Sorun bulunduğunda

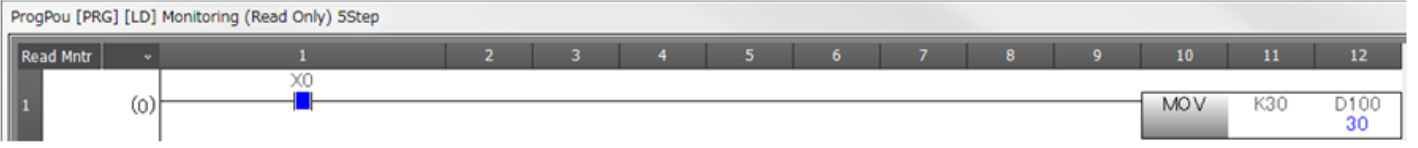


## 3.2

## Device değerlerini değiştirirken operasyonu kontrol etme

Oluşturulmuş bir program çalıştırılırken, bit devicelerin açık/kapalı durumu ve word devicelerinde saklanan değerler, program düzenleyicisinde görüntülenebilir. (Monitor (izleme) işlevi)

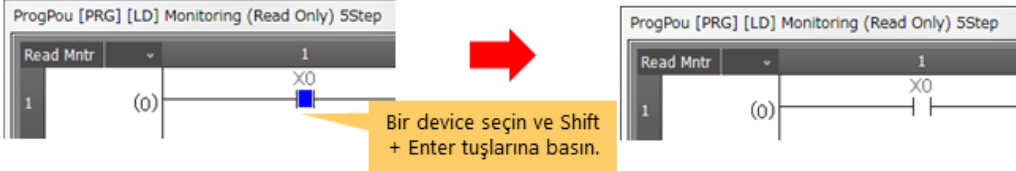
Kullanıcılar, izleme işleviyle programın işletim durumunu kolayca kontrol edebilirler.



Aygıtın mevcut değerleri izleme sırasında zorla değiştirilebilir. (Mevcut değer değişikliği)

Mevcut değer değişikliği işleviyle, programın tamamı düzenlenmeden veya program gerçek sistemde çalıştırılmadan değişiklik yapılabilir.

Bit devicelerin durumu program düzenleyicisinde açılabilir.

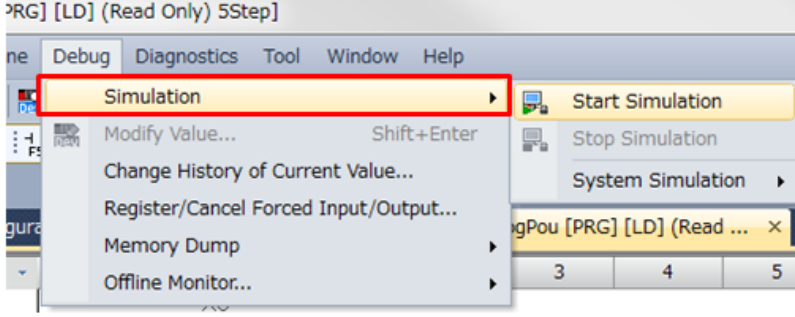


"Watch" (izle) penceresi kullanılarak, izlenecek word deviceleri kaydedilebilir ve bu devicelerin mevcut değerleri değiştirilebilir.

Watch 1[Watching]			
Deviceleri kaydedin.		Değerleri girin.	
Name	Current Value	Display Format	Data Type
X0	FALSE	BIN	Bit
D0	100	Decimal	Word [Signed]

Gerçek sistemde yeni oluşturulan bir program çalıştırılırsa beklenmeyen bir hata görülebilir.

Program operasyonu, gerçek bir programlanabilir kontrolör kullanmadan simüle edilebilir. (Simulation (Simülasyon) işlevi)



Simülasyon işleviyle program operasyonu, program gerçek bir programlanabilir kontrolörde çalışıyormuş gibi kontrol edilebilir.

Bu bölümde aşağıdakileri öğreniniz:

- Merdiven blokunun geçici olarak değiştirilmesi
- Program izleme ve mevcut değerin değiştirilmesi
- Program simülasyonu

Önemli noktalar

Merdiven blokunun geçici olarak değiştirilmesi	Bir merdiven bloku geçici olarak devre dışı bırakılabilir ve hata ayıklama için, orijinal program değiştirilmeden programın bir kopyası kullanılabilir.
İzleme	<ul style="list-style-type: none"><li>• Programın çalışma şekli görselleştirilebilir.</li><li>• Program çalışması, mevcut device değerleri zorla değiştirilirken kontrol edilebilir.</li></ul>
Simülasyon	Program operasyonu, gerçek bir programlanabilir kontrolör kullanmadan simüle edilebilir.

Böylece bu e-egitim kursunun sonuna geldik.

Aşağıda bu kursun özeti yer almaktadır.

- **Verimli programlama**

- Farklı sistemlerdeki programları kolayca kullanmak için modüllere sabit I/O numaraları atayın
- Bellek alanlarını, device kullanım durumuna göre ayarlayın
- Programlamayı daha kolay ve operasyonu daha anlaşılır hale getirmek için label kullanın
- Program okunurluğunu iyileştirmek için yorumlar (Comment) ekleyin

- **İleri programlama**

- Ölçülen süreyi tutmak için kalıcı zamanlayıcı kullanın
- Değerleri toplu olarak kullanmak için indeks kaydı, array veya structure kullanın
- Device durumunu tutmak için latch işlevi ve file register kullanın
- CPU modülünün dahili durumunu saklayan özel röle ve özel kayıt sağlanır
- Bir gerçek sayı, iki veya dört word device ile temsil edilir. Tam sayılar ve gerçek sayılar bir işlemde karıştırılmaz

- **Verimli hata ayıklama**

Kullanıcılar, GX Works3 kullanarak aşağıdakileri yapabilirler:

- Orijinal programı değiştirmeden bir programda hata ayıklama
- Programın çalışma şeklini görselleştirme
- Program operasyonunu simüle etme

Sonraki adıma geçmek için, bir programı, kolayca yeniden kullanılabilmesi için katmanlara ve bileşenlere ayıran "yapılandırma" işlemi hakkındaki kursları alın.

- Verimli Programlama
- Verimli Programlama (Uygulama) (yayınlanacak)

Artık **Programlama Uygulamaları (Merdiven Şeması/MELSEC iQ-R Serisi)** kursundaki tüm dersleri tamamladığınızdan, son teste girmeye hazırsınız. Ele alınan konulardan herhangi birini tam anlamadıysanız, lütfen bu konuları gözden geçirmek için bu fırsatı değerlendirin.

**Bu Son Testte toplam 14 soru (35 madde) yer almaktadır.**

Son testi istediğiniz sayıda uygulayabilirsiniz.

### Puan sonuçları

Doğru cevap sayısı, soru sayısı, doğru cevapların yüzdesi ve başarılı/başarısız sonucu puan sayfasında görüntülenir.

		1	2	3	4	5	6	7	8	9	10	11	12	
Yeniden Dene	Test 1	✓	✗	✗	✓									Toplam soru: <b>28</b>
	Test 2	✓	✓	✓	✓									Doğru cevaplar: <b>22</b>
	Test 3	✓												Vüde: <b>79 %</b>
	Test 4	✓												
	Test 5	✓	✓											
Yeniden Dene	Test 6	✓	✗	✗	✗									
	Test 7	✓	✓	✓	✓									
	Test 8	✓	✓	✓	✓	✓								
	Test 9	✓												
Yeniden Dene	Test 10	✗												

Testi geçmek için, doğru cevapların **%60** olması gerekir.

Modüllerin I/O sayısı noktasının ataması hakkında aşağıdaki ifadelerden hangisi doğrudur?

S1

- I/O noktaları her bir modüle manuel olarak atanabilir. Böylece, modül konfigürasyonu değiştirildiğinde programın modifiye edilmesi gerekmez
- Otomatik olarak atanan I/O noktaları değiştirilemez

Device noktası ayarı hakkında ařağıdaki ifadelerden hangisi doğrudur?

S1

- Device kullanılmasa bile her bir device için en az bir nokta atanmalıdır
- Noktalar, kullanılan nokta sayısına uygun olarak atanabilir



Labellar hakkında ařağıdaki ifadelerden hangileri doğrudur? (Birden fazla yanıtı vardır)

S1

Label kullanımı, işlem hedefinin tanımlanmasına yardımcı olur ve programlamayı kolaylaştırır

Modül sinyallerini ve ayar değerlerini temsil eden labellar sağlanır

Programın okunurluğunu iyileştirmek için elemanlara yorumlar (comment) eklenebilir

Labellara sabit sayılar atanabildiği için değerler program modifiye edilmeden değiştirilebilir

Retentive timer açıklandığı aşağıdaki metni tamamlayın.

Retentive timer, (S1) açıldığında ölçüme başlar (bobin (S2) duruma geçer).

Retentive timer, giriş koşulu (S3) duruma geçtiğinde bile ölçülen süreyi tutar ve giriş koşulu tekrar (S4) duruma geçtiğinde ölçüme tutulan değerden devam eder.

Retentive timer, ölçülen süre ayar değerine ulaştığında zaman aşımına uğrar ve bu noktada (S5) açılır.

S1	-- Select --	S2	-- Select --
S3	-- Select --	S4	-- Select --
S5	-- Select --	S6	-- Select --

Aşağıdaki işlemi yürüten kontrol programını tamamlayın.

- Giriş sinyali X0 veya X1'in açık kalma süresini ölçmek için retentive timer (ST0) kullanın
- X0 veya X1'in açık kalma süresi 30 saniyeye ulaştığında Y70 bobinini ve zaman aşımı göstergesini açın
- X2 açıldığında, retentif zamanlayıcının (ST0) kontağını kapatın ve ölçülen süreyi (mevcut değer) temizleyin

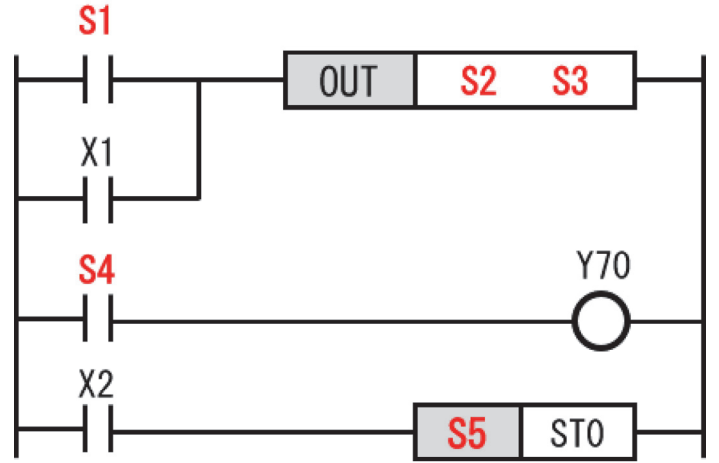
S1 -- Select --

S2 -- Select --

S3 -- Select --

S4 -- Select --

S5 -- Select --



[ + ]

Aşağıdaki kontrol programındaki koşullardan her birinin altında, X0 açıldığında D20 veri kaydında saklanan değeri seçin.

S1) Z2'de saklanan değer "0" olduğunda

S2) Z2'de saklanan değer "1" olduğunda

S3) Z2'de saklanan değer "2" olduğunda

S4) Z2'de saklanan değer "3" olduğunda

S1

-- Select --

S2

-- Select --

S3

-- Select --

S4

-- Select --



Data registerda  
saklanan değerler

D0	100
D1	200
D2	400
D3	500

[ + ]

Bir array bir elemanın nasıl belirtileceği hakkında aşağıdaki ifadelerden hangisi doğrudur?

S1

- Label adının sonuna bir eleman sayısı eklenir
- Doğrudan bir device numarası belirtilir

Structure hakkında ařağıdaki ifadelerden hangisi doğrudur?

S1

- Structure, fiziksel nesnelere veya cisimlerle ilgili koşulları ve özellikleri toplu halde organize etmek ve saklamak için kullanılır
- Structure kullanılarak, büyük miktarda verinin işleme kısaca açıklanabilir
- Bir structure içinde tanımlanan elemanlar aynı veri tipine sahip olmalıdır

Latch işlevi hakkında aşağıdaki ifadelerden hangisi doğrudur?

S1

- Devicelerin başlangıçta değerleri tutma işlevi bulunur
- Mühendislik yazılımı kullanılarak, değerleri tutmak için parametre ayarı yapılması gereklidir

Dosya kaydını açıklayan aşağıdaki metni tamamlayın.

Dosya kaydı, veri kaydını (D) genişletmek için kullanılan bir sözcük aygıtıdır ve device sembolü **(S1)** şeklindedir. File register, datda registra göre **(S2)** kapasiteye sahiptir ve saklanan veriler, sistem kapatıldığında veya CPU modülü sıfırlandığında bile **(S3)**.

File register kullanmak için mühendislik yazılımı kullanılarak parametre ayarı yapılması **(S4)**.

**S1**

-- Select --

**S2**

-- Select --

**S3**

-- Select --

**S4**

-- Select --



Özel röle veya özel kayıt hakkında aşağıdaki ifadelerden hangisi doğrudur?

S1

- CPU modülünün dahili durumu özel röle ve özel kayıta halihazırda saklanmıştır ve bu devicelar, bir kontrol programında belirleme koşulları olarak kullanılır
- Özel röle ve özel kayda özel işlevler serbestçe atanabilir

Gerçek sayıları (tek hassasiyetli) açıklayan aşağıdaki metni tamamlayın.

- Bir gerçek sayı **(S1)** sözcük aygıtı kullanır ve **(S2)** bitlik bir bellek alanında saklanır.
- Gerçek sayıların sayısal değer verileri **(S3)** olarak adlandırılır. Örneğin 2,035, bir kontrol programında **(S4)** olarak tanımlanır.
- Gerçek sayıların kullanıldığı bir çalışma yönergesinde tam sayılar ile gerçek sayıların karıştırılması **(S5)**.

<b>S1</b>	-- Select --	<b>S2</b>	-- Select --
<b>S3</b>	-- Select --	<b>S4</b>	-- Select --
<b>S5</b>	-- Select --		

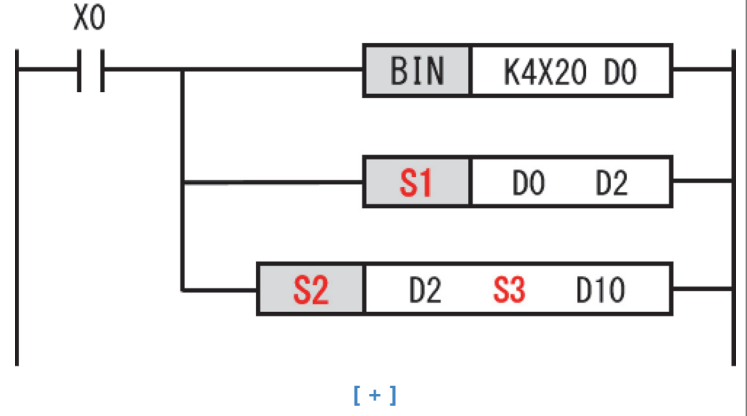
Aşağıdaki işlemi yürüten kontrol programını tamamlayın.

- X0 açık olduğunda X20'den X2F'ye veriler (BCD verileri) okunur ve D0'da saklanır.
- D0'daki değer gerçek sayıya dönüştürülür ve dönüştürülen değer D2'de saklanır.
- D2'deki değer 3,14 ile çarpılır ve sonuç D10'da saklanır.

S1

S2

S3



Kontrol programında hata ayıklama hakkında aşağıdaki ifadelerden hangisi doğrudur?

S1

- Mühendislik yazılımının işlevi kullanılarak program çalışması güvenli bir şekilde simüle edilebilir.
- Bir programda hata ayıklamak için program gerçek sistemde yürütülmelidir.

Son Testi tamamladınız. Sonuç alanınız aşağıda gösterildiği gibidir.  
Son Testi sonlandırmak için bir sonraki sayfaya ilerleyin.

	1	2	3	4	5	6	7	8	9	10
Son Test 1	✓									
Son Test 2	✓									
Son Test 3	✓									
Son Test 4	✓	✓	✓	✓	✓	✓				
Son Test 5	✓	✓	✓	✓	✓					
Son Test 6	✓	✓	✓	✓						
Son Test 7	✓									
Son Test 8	✓									
Son Test 9	✓									
Son Test 10	✓	✓	✓	✓						
Son Test 11	✓									
Son Test 12	✓	✓	✓	✓	✓					
Son Test 13	✓	✓	✓							
Son Test 14	✓									

Toplam soru: **35**

Doğru cevaplar: **35**

Yüzde: **100 %**

Temizle

**Programlama Uygulamaları (Merdiven Şeması/MELSEC iQ-R Serisi) kursunu tamamladınız.**

Bu kursa katıldığınız için teşekkür ederiz.

Derslerden keyif almış olmanızı ve bu kursta edindiğiniz bilgilerin gelecekte faydalı olmasını umarız.

Kursu istediğiniz kadar çok gözden geçirebilirsiniz.

**İncele**

**Kapat**