



PLC

GX Works2 İleri Düzey Bilgileri

Bu kurs, "verimlilik", "kalite", "proje yönetimi" ve "güvenlik önlemleri" konularında sorunlar yaşayan tasarım sahasının gelişim ortamını iyileştirmeye yönelik işlevler hakkında bilgi sağlamaktadır. Kursun hedef kitlesi halihazırda MELSOFT GX Works2'yi kullanan sıra programlayıcılarıdır.

Giriş**Kursun Amacı**

Bu kurs, "verimlilik", "kalite", "proje yönetimi" ve "güvenlik önlemleri" konularında sorunlar yaşayan tasarım sahasının gelişim ortamını iyileştirmeye yönelik işlevler hakkında bilgi sağlamaktadır. Kursun hedef kitlesi halihazırda MELSOFT GX Works2'yi kullanan sıra programlayıcılarıdır.

Giriş**Kursun Yapısı**

Bu kursun içeriği aşağıdaki gibidir.
Bölüm 1'den başlamanızı tavsiye ederiz.

Bölüm 1 - GX Works2'yi Kullanarak Gelişim Ortamını İyileştirme

Tasarım sahasının karşılaştığı sorunlar ve bu sorunları çözmek için gereken gelişim ortamı hakkında bilgi sahibi olacaksınız.

Bölüm 2 - Programlama

Programlama için kullanılan işlevleri öğreneceksiniz.

Bölüm 3 - Hata Ayıklama

Hata ayıklama için kullanılan işlevleri öğreneceksiniz.

Bölüm 4 - Proje Yönetimi ve Güvenlik Önlemleri

Sistem operasyonunun ardından gelişim ve bakım aşamalarındaki proje yönetimi ve güvenlik önlemleri hakkındaki işlevleri öğreneceksiniz.

Son Test

Geçer not: %60 veya üzeri

Giriş**Bu e-Eğitim Aracının Kullanımı**

Sonraki sayfaya git		Sonraki sayfaya gidin.
Önceki sayfaya dön		Önceki sayfaya dönün.
İstenen sayfaya ulaş		"İçindekiler Tablosu" görüntülenerek istediğiniz sayfaya ulaşabilmenizi sağlar.
Eğitimden çık		Eğitimden çıkın. "İçindekiler" ekranı gibi pencereler ve eğitim kapatılacaktır.

Güvenlik tedbirleri

Gerçek ürünleri kullanmayı öğrendiğinizde, lütfen ilgili kılavuzlardaki güvenlik önlemlerini dikkatlice okuyun.

Bu kurstaki önlemler

- Kullandığınız yazılım sürümünde görüntülenen ekranlar bu kurstakilerden farklı olabilir.

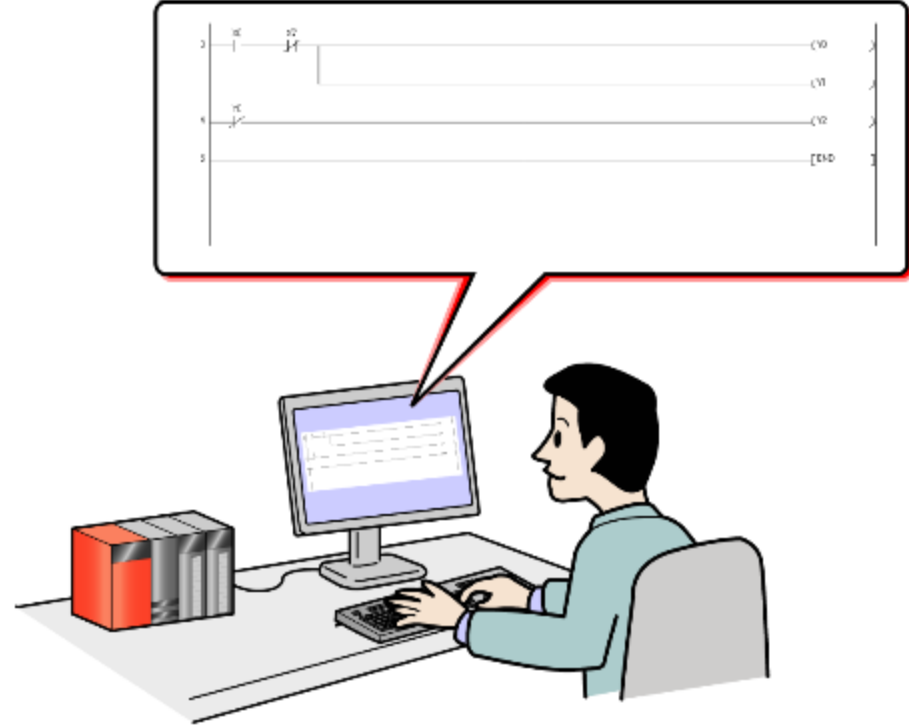
Bölüm 1 GX Works2'yi Kullanarak Gelişim Ortamını İyileştirme

Bölüm 1'deki öğrenme adımları

Bu kursun hedef kitlesi sıra programlarının gelişimi üzerinde çalışan programlayıcılardır. **MELSOFT GX Works2**'deki "**Verimlilik**", "**kalite**", "**proje yönetimi**" ve "**güvenlik**" konularına ilişkin tasarım sahası sorunlarını çözmeye yönelik mükemmel işlevleri nasıl kullanacağınızı öğreneceksiniz.

Bölüm 1'de, tasarım sahasının karşılaştığı sorunları ve bu sorunları çözmek için gereken gelişim ortamı hakkında bilgi sahibi olacaksınız.

- 1.1 Tasarım Sahasının Karşılaştığı Sorunlar
 - 1.1.1 Verimliliği İyileştirme
 - 1.1.2 Kalite İyileştirme
 - 1.1.3 Proje Yönetimi
 - 1.1.4 Güvenlik Önlemleri
- 1.2 Öğrenme Prosedürü



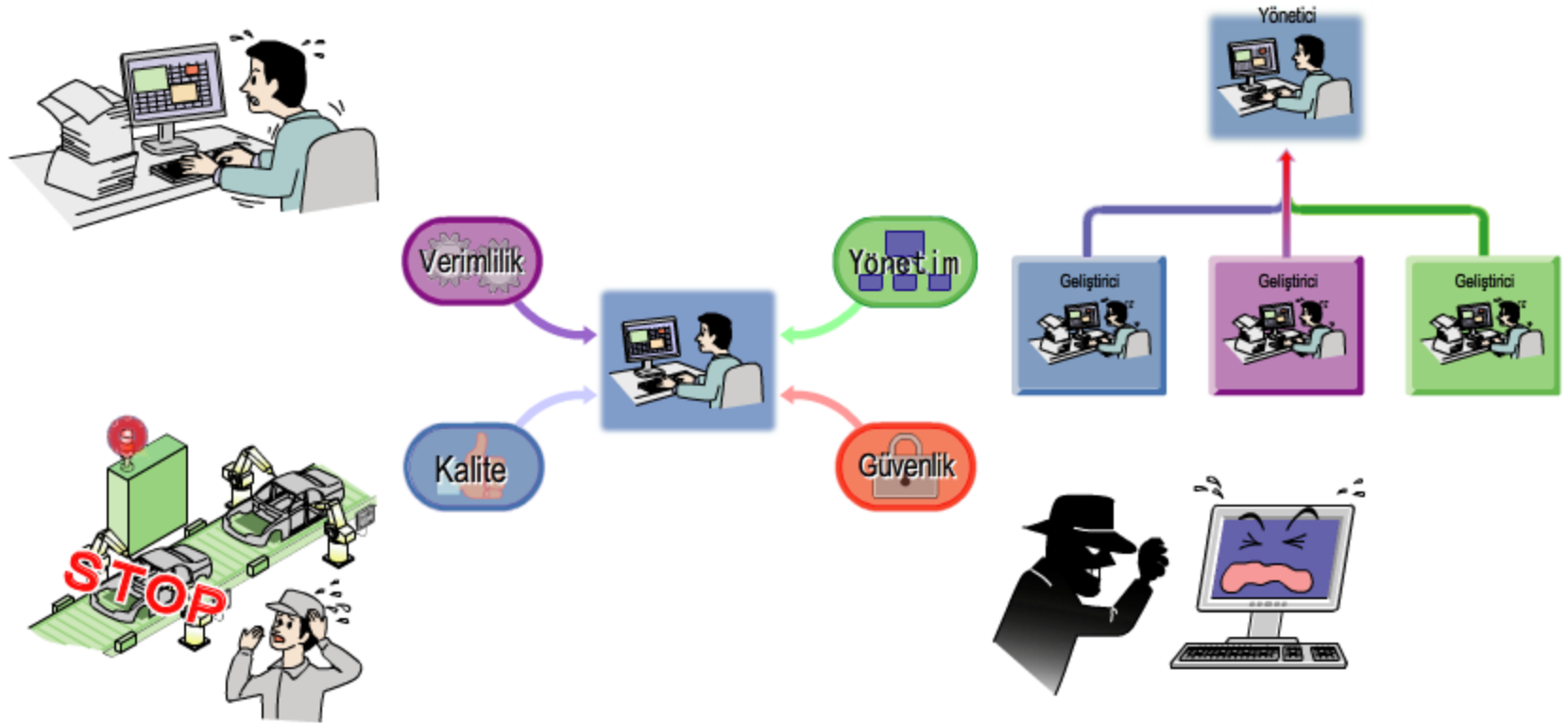
1.1

Tasarım Sahasının Karşılaştığı Sorunlar

Tasarım sahasının bir yandan gelişim maliyetlerini düşürmek amacıyla verimliliği iyileştirirken bir yandan da geliştirilen tüm programların yüksek kaliteli olmasını sağlaması gerekmektedir.

Ayrıca tasarım sahası ekip gelişimine imkan tanıyan ve sorun çıkması halinde hızlıca iyileşme sağlayan bir proje yönetimi yürütmelidir.

Sıra programlarının önemli bilgiler ve veriler içermesi nedeniyle güvenlik önlemlerinin de alınması gerekir.



1.1.1 Verimliliği İyileştirme

Tasarım sahasının büyük ölçekli ve kompleks işleme gerçekleştirebilmesi için sıra programları geliştirmesi gerekmektedir.

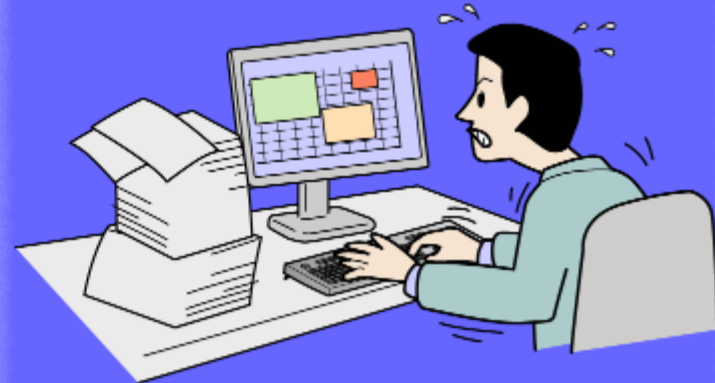
Ancak, gelişim maliyeti, gelişim sürecinin uzunluğu ile paralel olarak artar.

Bu nedenle, gelişim maliyetini düşürebilmek için program gelişim verimliliğini iyileştirmeniz gerekmektedir.

GX Works2, bu sorunu çözmek için aşağıdaki işlevleri sunmaktadır:

- Etiket
- İşlev bloğu
- Aygıtın ilk değeri ve aygıt belleği*
- Satır içi yapılandırılmış metin
- Örnek yorumdan içe aktarma

* MELSEC-F serisi ilk aygıt değerlerini yapılandıramaz.



1.1.2 Kalite İyileştirme

Sıra programındaki bir hata sistemin durmasına, arızalanmasına veya sistemde bir kaza meydana gelmesine neden olur ve böylesi bir durum üretimi aksatabilir; kar ve güvenilirlik kaybına sebep olur. Hata barındırmayan kalite programları geliştirmeniz gerekmektedir.

GX Works2, bu sorunu çözmek için aşağıdaki işlevleri sunmaktadır:

- Etiket
- İşlev bloğu
- Aygıtın ilk değeri ve aygıt belleği*
- Yorum
- Örnek yorumdan içe aktarma
- İzleme
- Çapraz başvuru
- Örnekleme izi*
- İşlemsel koşullu aygıt testi*
- Adım yürütme işlevi*
- I/O sistem ayarı



* MELSEC-F serisi ilk aygıt değeri ayarlarını, örnekleme izini, işlemsel koşullu aygıt testini ve adım yürütme işlevini kullanamaz.

1.1.3 Proje Yönetimi

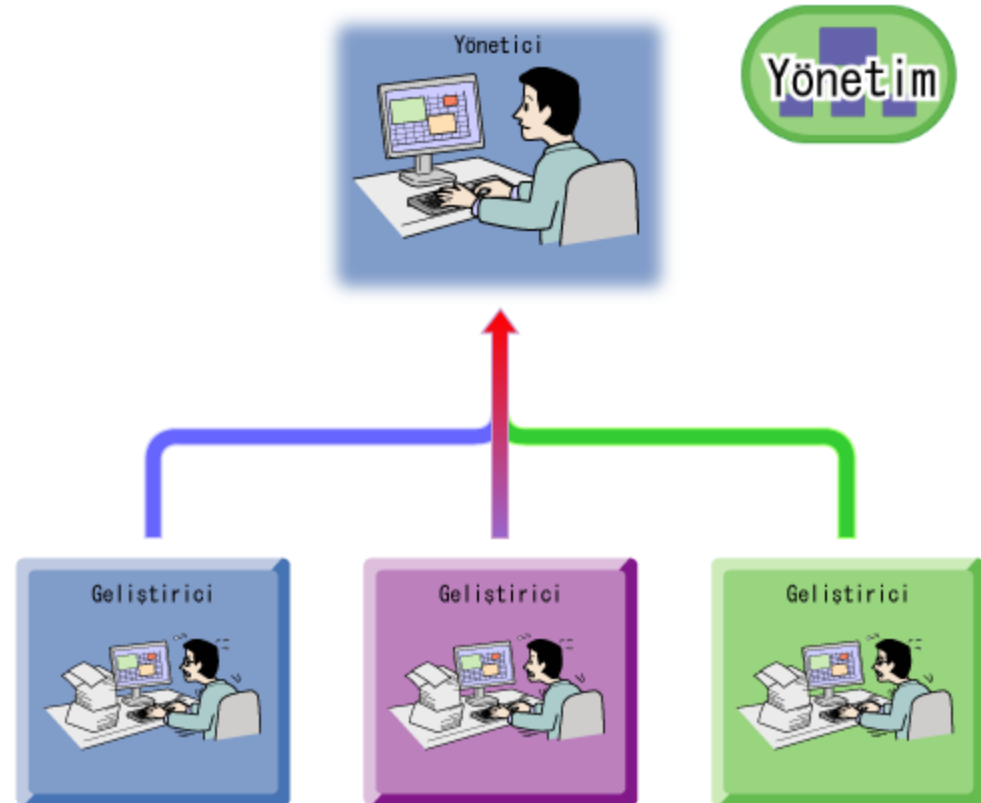
Büyük ölçekli sıra programı genellikle bir programlayıcı ekibi tarafından geliştirilir.

Bireysel yönetim yetkisi ve veri gizliliği bakımından, erişilebilir veri aralığı ve kullanılabilir işlevler için sınırlar belirlenmelidir.

Buna ek olarak, programlanabilir kontrolör arızasından kaynaklanan program kaybı durumlarında hızlı kurtarma sağlamak için program sürüm yönetimi ve periyodik program yedeklemesi uygulamanız gerekmektedir.

GX Works2, bu sorunları çözmek için aşağıdaki işlevleri sunmaktadır:

- Güvenlik
- Düzeltme geçmişi
- Kişisel bilgisayarla doğrulama



1.1.4

Güvenlik Önlemleri

Sıra programı, stratejik açıdan önemli bilgiler ve veriler içerir.

Programa ait bilgi birikimi ve verilerin programın dışına **sızması** işletme üzerinde yıkıcı bir etkiye sahip olabilir.

Programın **yetkisiz değiştirilmesi** sistemin durması gibi üretim sorunlarına neden olabilir.

Bu tür sorunların ortaya çıkmasını engellemek için uygun güvenlik önlemleri alınmalıdır.

GX Works2, bu sorunları çözmek için aşağıdaki işlevi sunmaktadır:

- Güvenlik



1.2 Öğrenme Prosedürü

Bu kursta, gerçek sistem gelişimine yönelik aşağıdaki prosedürü izleyerek GX Works2 işlevleri hakkında bilgi sahibi olacaksınız.

(1) Programlama Bölüm 2




(2) Hata ayıklama Bölüm 3





(3) Proje yönetimi ve güvenlik önlemleri Bölüm 4


<Simgelerin açıklaması>

Bölüm 2, 3 ve 4'teki sayfalarda görüntülenen simgeler aşağıda gösterildiği şekilde ilgili gelişme işlevlerine karşılık gelmektedir.

 Verimlilik Program verimliliğini geliştirmek için kullanılan işlev

 Kalite Program kalitesini geliştirmek için kullanılan işlev

 Yönetim Proje yönetimi için kullanılan işlev

 Güvenlik Güvenlik tedbirleri için kullanılan işlev

Bölüm 2 Programlama

Bölüm 2'deki öğrenme adımları

Bölüm 2'de programlama için kullanılan işlevler hakkında bilgi sahibi olacaksınız. GX Works2, verimli programlama sağlamak ve programların kalitesini iyileştirmek için birçok farklı işlev sunmaktadır.

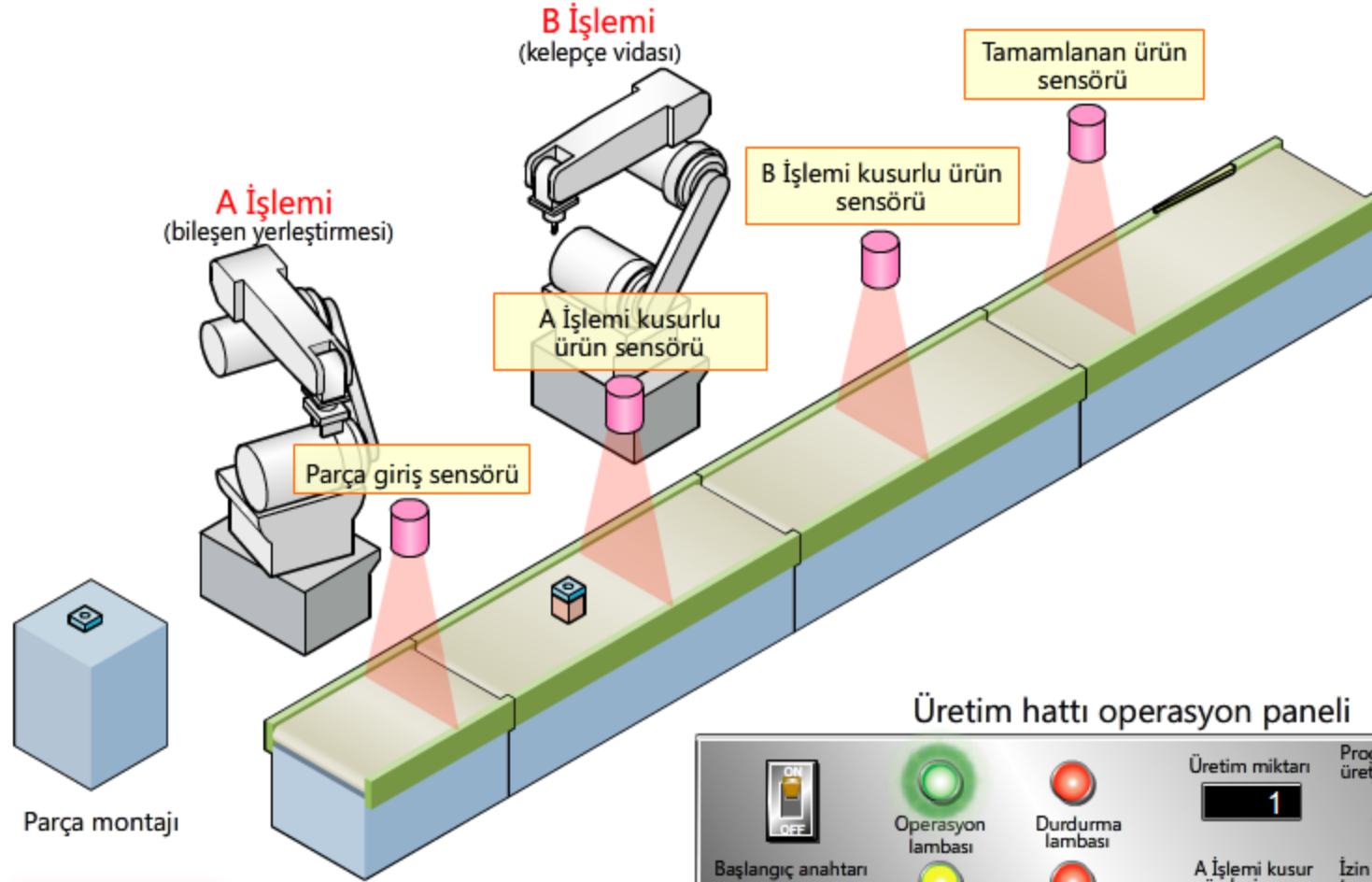
- 2.1 Aygıt Adını Uygulamaya İlişkin bir Adla Değişirme
 - 2.1.1 Etiket Tipleri
 - 2.1.2 Etiket Uygulamaları ve Kayıtlı Değerlerin Tipleri
 - 2.1.3 Etiket Kaydı
 - 2.1.4 Etiketlerin Aygıtlara Otomatik Atanması
- 2.2 Sürekli Kullanılan Merdiven Bloklarının Değişiklik için İşlev Blokları Olarak Düzenlenmesi
 - 2.2.1 İşlev Bloklarının Oluşturulması ve Yerleştirilmesi
 - 2.2.2 İşlev Bloğu Kitaplığının Kullanımı
- 2.3 Programı Düzeltmeden İlk Aygıt Değerlerini Değişirme
- 2.4 Merdiven Programlarının Basitleştirilmesi
- 2.5 Anlaması ve Okuması Kolay Programların Oluşturulması
 - 2.5.1 Her bir Merdiven Bloğu için Yorum Yazma
 - 2.5.2 Her bir Çıkış (Bobin, Uygulama Yönergesi) için Yorum Yazma
- 2.6 Özel Röleler/Kayıtlar İçeren Programların Okunmasını Basitleştirme

Bölüm 2 Bu Kursta Öğrenme için Kullanılan Sisteme Giriş

Verimlilik

Kalite

Bu kursta GX Works2 işlevlerini nasıl kullanacağınıza yardımcı olmak için aşağıdaki varsayımsal sistem kullanılmaktadır.



▶ Normal üretim

▶ A İşlemi kusur yüzdesi hatası

▶ B İşlemi kusur yüzdesi hatası

"Back to Top" (Yukarıya Dön) düğmesine basılması halinde ilk operasyon seçimine dönlür.

Yukarıya Dön

Üretim hattı operasyon paneli

			Üretim miktarı	Programlanan üretim hacmi
Başlangıç anahtarı	Operasyon lambası	Durdurma lambası	1	20
			A İşlemi kusur yüzdesi	İzin verilen A İşlemi kusur yüzdesi
Üretim miktar sıfırlaması	Ulaşılan üretim miktarı	A İşlemi kusur yüzdesi hata lambası	0	10
			B İşlemi kusur yüzdesi	İzin verilen B İşlemi kusur yüzdesi
		B İşlemi kusur yüzdesi hata lambası	0	5

2.1

Aygıt Adını Uygulamaya İlişkin bir Adla Değiştirme

Verimlilik

Kalite

Sıra programında kullanılan **aygıt** "M0" veya "D5" gibi bir harf ve ardından bir rakamdan oluşan bir ad verilir.

Böylece aygıt adı aygıtın uygulamasına ilişkin herhangi bir ipucu vermez.

Büyük ölçekli bir programda birçok aygıt tipi kullanılır ve bu da programlama sırasında aygıtların uygulamasını tespit etmek için sistem tasarımı belgelerini sürekli olarak kontrol etmeniz gerektiği anlamına gelmektedir. Bu, çalışma verimliliğini düşürür ve aygıtların seçimindeki hatalardan dolayı program kalitesini olumsuz yönde etkiler.

Bu sorunları çözmek için **"etiketler"** kullanılabilir.

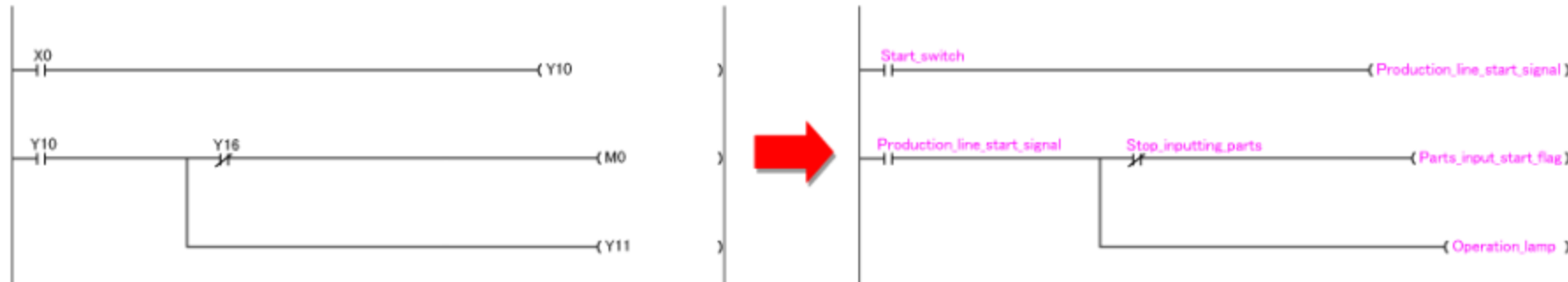
Aygıt adını kullanmak yerine, **"Üretim hattı başlangıç sinyali"** gibi gerçek uygulamayı belirten bir ad (etiket) kullanabilirsiniz.

Bu tür bir ad için alfasayısal karakterlere ek olarak Japonca (hiragana, katakana ve kanji) karakterler de kullanılabilir.

<BELİRLENEN yönerge uyarınca üretim hattı başlangıç sinyalini içeren "M0"ı AÇIK konumuna getirmeye yönelik ifade>

Aygıt	—[SET M0]—
Etiket	—[SET Üretim hattı başlangıç sinyali]—

Bu tür bir etiket kullanmak; okuması kolay bir program oluşturmak, program gelişimini iyileştirmek ve aygıt giriş hatalarını önlemek için etkilidir.



2.1.1 Etiket Tipleri

Verimlilik

Kalite

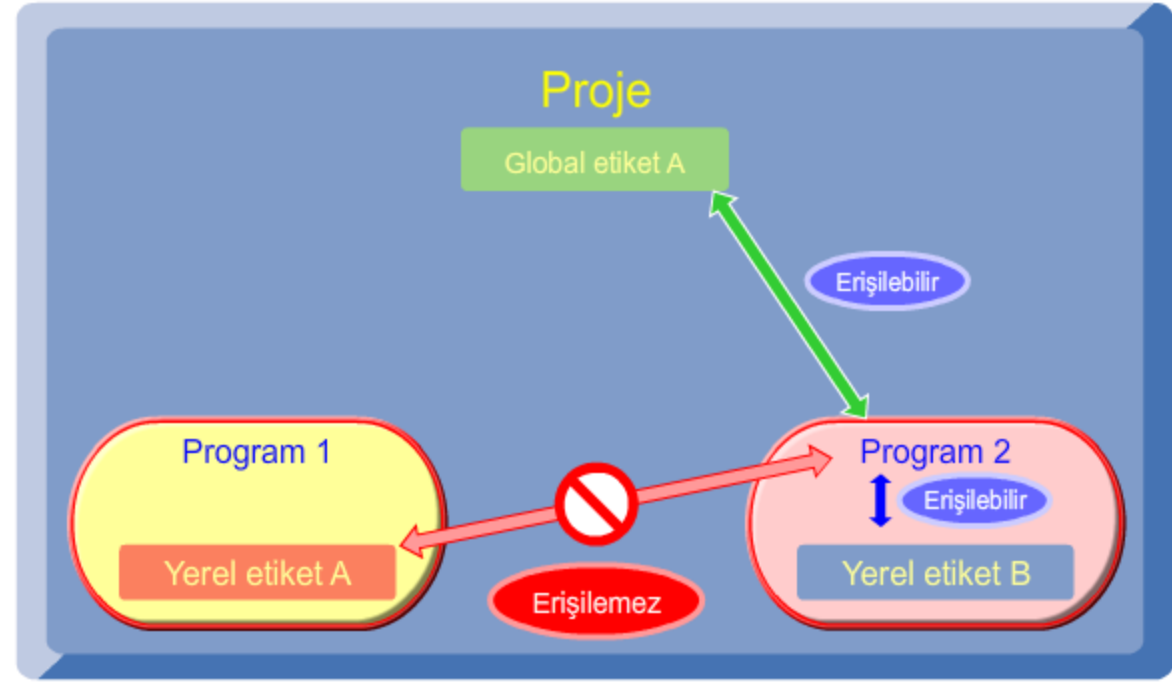
İki etiket tipi mevcuttur: "**global etiket**" ve "**yerel etiket**".

<Global etiket>

Global etiketler tüm proje için kullanılır ve bu etiketlere söz konusu projedeki herhangi bir program ile erişilebilir.

<Yerel etiket>

Yerel etiketler belirli bir programda kullanılır ve bu etiketlere yalnızca etiketin kayıtlı olduğu program tarafından erişilebilir.



"Program 1" ve "Program 2" düğmedir. İki etiket tipine ulaşip ulaşamadığını görmek için herhangi bir programa basın.

2.1.2 Etiket Uygulamaları ve Kayıtlı Değerlerin Tipleri

Verimlilik

Kalite

Bir etiketi kaydederken, etiket uygulaması ile "**Class**" (Sınıf) ve "**Data type**" (Veri tipi) kullanılarak kayıt edilebilecek değer tipini belirleyin.

<Sınıf>

Sınıf, etiketin kullanım aralığını ve uygulamasını belirtir.

Seçilebilecek sınıflar etiket tipine göre farklılık gösterir.

Sınıf	Kullanım amacı	Etiket ayar alanı		
		Global etiket	Program yerel etiketi	İşlev bloğu yerel etiketi
VAR_GLOBAL	Bir projedeki programlar ve işlev blokları tarafından kullanılacak genel etiket	<input type="radio"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
VAR_GLOBAL_CONSTANT	Bir projedeki programlar ve işlev blokları tarafından kullanılacak, sabit içeren genel etiket	<input type="radio"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
VAR	Etiketin kullanılacağı programlar ve işlev blokları tarafından kullanılabilen etiket	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>
VAR_CONSTANT	Etiketin kullanılacağı programlar ve işlev blokları tarafından kullanılabilen, sabit içeren etiket	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>
VAR_RETAIN	Etiketin kullanılacağı programlar ve işlev blokları tarafından kullanılabilen etiket (mandal tipi)	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>
VAR_INPUT	Etiketin kullanılacağı işlev bloğunun girişi için kullanılan etiket * Değer, bir program bileşeninde değiştirilemez.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>
VAR_OUTPUT	Etiketin kullanılacağı işlev bloğunun çıkışı için kullanılan etiket	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>
VAR_IN_OUT	Etiketin kullanılacağı işlev bloğunun giriş ve çıkışı için kullanılan etiket * Değer, bir program bileşeninde değiştirilebilir.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>

2.1.2

Etiket Uygulamaları ve Kayıtlı Değerlerin Tipleri

Verimlilik

Kalite

<Veri tipi>

Veri tipi, etikette kayıtlı değer tipine işaret eder.

Bir etikete atanan veri tipi, etiket ve ilgili aygıtta saklanabilecek değerlerin tipini ve aralığını ifade eder.

Merdiven programları ile kullanılabilir veri tipleri aşağıda listelenmiştir.

Veri tipi	Açıklama	Bit uzunluğu	Kayıtlı değerlerin aralığı
Bit	ON (Açık) veya OFF (Kapalı) kayıtlıdır. "M" aygıtına karşılık gelir	1 bit	1: ON (Açık), 0: OFF (Kapalı)
Sözcük (işaretli)	Kesir içermeyen bir tamsayı kaydedilir.	16 bit	-32768 ila 32767
Çift sözcük (işaretli)	"D" aygıtına karşılık gelir	32 bit	-2147483648 ila 2147483647
FLOAT (tek hassasiyet)	Kesirli gerçek bir sayı kaydedilir.	16 bit	-2^{128} ila -2^{-126} , 0, 2^{-126} ila 2^{128}
FLOAT (çift hassasiyet)	"D" aygıtına karşılık gelir	32 bit	-2^{1024} ila -2^{-1022} , 0, 2^{-1022} ila 2^{1024}
Dize	Bir karakter dizesi kaydedilir. "D" aygıtına karşılık gelir	Değişken	Maksimum 255 karakter
Zamanlayıcı	Belirlenen zamana ulaşıldığında açılır. "T" zamanlayıcı aygıtına karşılık gelir	-	-
Retentif zamanlayıcı	Belirlenen zamana ulaşıldığında açılır. "ST" retentif zamanlayıcı aygıtına karşılık gelir	-	-
Sayaç	Belirlene sayıya ulaşıldığında açılır. "C" sayaç aygıtına karşılık gelir	-	-
İşaretçi	Alt rutin başlangıç konumu kaydedilir. "P" işaretçi aygıtına karşılık gelir	-	-

2.1.3 Etiket Kaydı



MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [Local Label Setting MAIN [PRG]]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

[PRG]write MAIN (194)Step * Global Label Setting Global1 Local Label Setting MAIN [P...]

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
 - Global1
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
- FB_Pool
- Structured Data Type
- Local Device Commen

- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

	Class	Label Name	Data Type	Constant
1	VAR	Production_qty_atained	Bit	...
2	VAR	Parts_input_start_flag	Bit	...
3	VAR	Start_reject_pct_calc	Bit	...
4				...
5				...
6				...
7				...
8				...
9				...
10				...
11				...
12				...
13				...
14				...
15				...
16				...
17				...
18				...
19				...
20				...
21				...
22				...
23				...
24				...
25				...
26				...

Etiket kaydı için ayarları tamamlayın.
Devam etmek için esine tıklayın.

2.1.4 Etiketlerin Aygıtlara Otomatik Atanması

Verimlilik

Kalite

Bir programın otomatik olarak dönüştürülmesi sınıf ve veri tipine göre uygun aygıtı etikete atar. Bir etiket kullanırken, hangi aygıtın atandığının bilinmesine gerek yoktur.

Etikete atanan aygıtların aralığını değiştirmek için "**Device/Label Automatic-Assign Setting**"i (**Aygıt/Etiket Otomatik Ataması**) kullanın.

* Bu ekran, MELSEC-Q VE MELSEC-F serileri için otomatik atama aygıt ayar penceresidir. Ekran MELSEC-F serisinde farklılık gösterebilir.

<"Device/Label Automatic-Assign Setting"i (**Aygıt/Etiket Otomatik Atama Ayarı**) **Başlatma**> GX Works2 menüsünden, **[Tool]** – **[Device/Label Automatic-Assign Setting]** (**[Araç]** - **[Aygıt/Etiket Otomatik Atama Ayarı]**) öğesini seçin.

Device/Label Automatic-Assign Setting

Set a device range to automatically assign to labels.
Labels will be assigned from its way down the displayed device list when multiple devices are selected.

	Device	Digit	Assign Selection	Assignment Range		Total Points	PLC Parameter Device Setting Range
				Start	End		
Word Device							
VAR Range	D	10	<input checked="" type="checkbox"/>	6144	12287	6144	0 -- 12287
	W	16	<input type="checkbox"/>				0 -- 1FFF
	R	10	<input type="checkbox"/>				
VAR_RETAIN Range Latch(1)	D Latch	10	<input type="checkbox"/>			0	
	W Latch	16	<input type="checkbox"/>				
	ZR Latch	10	<input type="checkbox"/>				
Bit Device							
VAR Range	M	10	<input checked="" type="checkbox"/>	4096	8191	4096	0 -- 8191
	B	16	<input type="checkbox"/>				0 -- 1FFF
VAR_RETAIN Range Latch(1)	L Latch	10	<input type="checkbox"/>			0	
	B Latch	16	<input type="checkbox"/>				
Pointer							
VAR Range	P	10	<input checked="" type="checkbox"/>	2048	4095	2048	2048 -- 4095
Timer							
VAR Range	T	10	<input checked="" type="checkbox"/>	64	2047	1984	0 -- 2047
VAR_RETAIN Range Latch(1)	T Latch	10	<input type="checkbox"/>			0	
Retentive Timer							
VAR Range	ST	10	<input type="checkbox"/>			0	
VAR_RETAIN Range Latch(1)	ST Latch	10	<input type="checkbox"/>			0	
Counter							
VAR Range	C	10	<input checked="" type="checkbox"/>	512	1023	512	0 -- 1023
VAR_RETAIN Range Latch(1)	C Latch	10	<input type="checkbox"/>			0	

Latch(1) : Able to clear the value by using a latch clear.
Latch(2) : Unable to clear the value by using a latch clear. Clearing will be executed by remote operation or program.

(Caution)
1. Label-nonassigned devices, of the automatically assigned ones while compiling, will be allotted the device that displayed at the lowest of the selected ones. Ex)-Device will be assigned to ZR when D and ZR are selected.
2. Changing the assignment target device may also change the processing speed since the arithmetic processing speed for R and ZR is difference from other devices.

OK Cancel

Robot control - [[PRG]Write MAIN 194 Step]

Tool Window Help

- IC Memory Card
- Check Parameter...
- Options...
- Key Customize...
- Device/Label Automatic-Assign Setting...**
- Block Password...
- Confirm Memory Size...
- Set TEL Data/Connect via Modem
- LCPU Logging Configuration Tool...
- Ethernet Adapter Module Configuration Tool...
- Built-in I/O Module Tool
- Check Intelligent Function Module Parameter
- Intelligent Function Module Tool
- Language Selection...

2.2

Art Arda Kullanılan Merdiven Bloklarının Değişiklik için İşlev Blokları Olarak Düzenlenmesi

Verimlilik

Kalite

Büyük ölçekli bir programda, bazı merdiven blokları art arda şekilde kullanılabilir.

Veya bazı merdiven blokları farklı programlarda kullanılabilir.

Gerektiği her zaman aynı merdiven bloğunu girmeniz gerekiyorsa çalışma verimliliği iyileştirilemez.

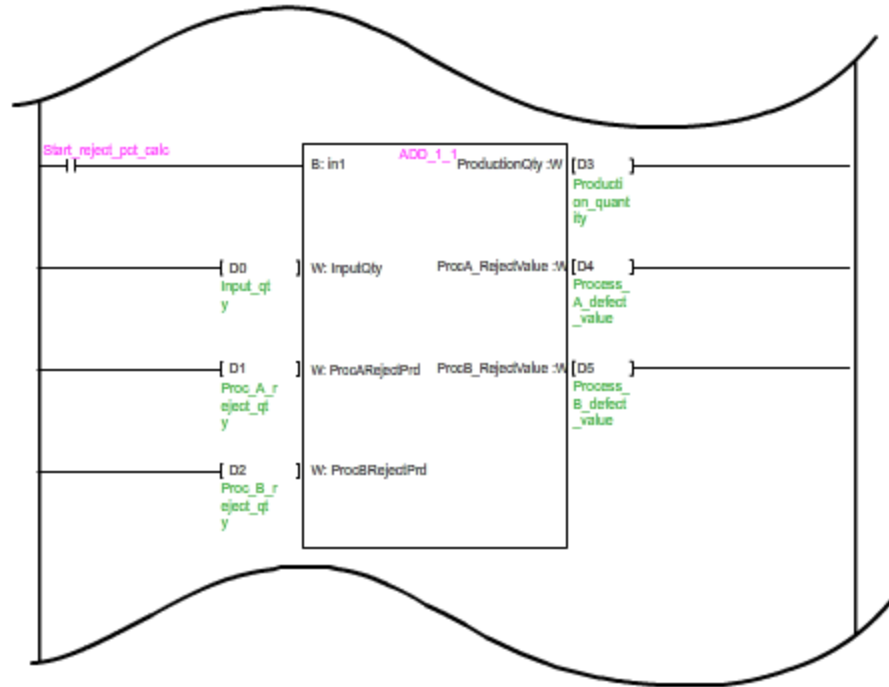
Buna ek olarak, genel merdiven bloğunda bir arıza tespit edilirse, her bir bloğun belirlenmesi ve düzeltilmesi gerekir. Bu, önemli bir zaman kaybına yol açar.

Bu tür sorunları çözmek için **"Function Block (FB)"**yi (**İşlev Bloğu (İB)**) kullanın.

Art arda kullanılan bir merdiven bloğu, her programda kullanılabilen **işlev bloğu olarak düzenlenebilir**. Bu işlem çalışma verimliliğini önemli ölçüde artırır.

Genel blokta bir arıza tespit edilirse, tüm yapmanız gereken işlev bloğunu düzeltmektir.

Bu yalnızca zaman tasarrufu sağlamaz; aynı zamanda blokların birini düzeltmemeden kaynaklanan riskin oluşmasını engeller.



Animasyonu baştan izlemek için düğmeye basın.



2.2.1 İşlev Bloklarının Oluşturulması ve Yerleştirilmesi

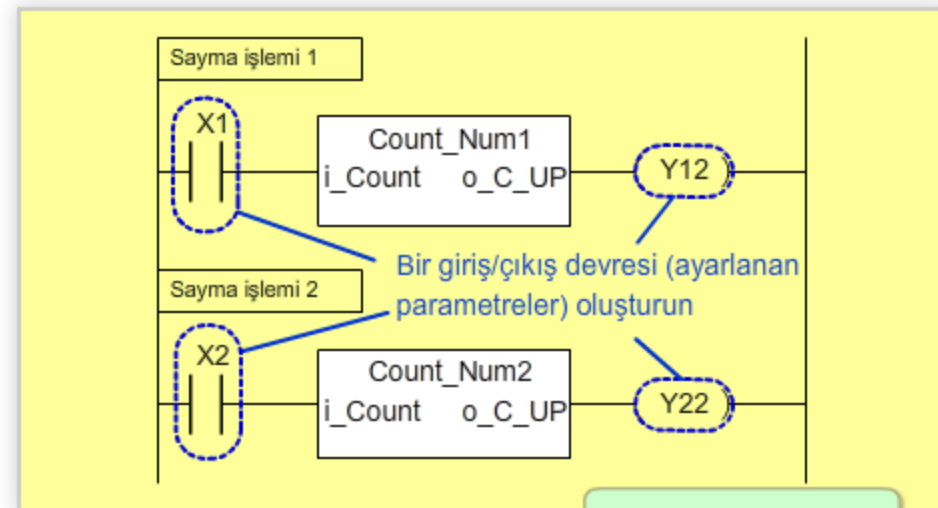
Verimlilik

Kalite

İşlev bloğu farklı programlarda kullanılacağı için işlev bloğunda gerçek aygıtlar (X, Y ve D gibi) kullanılamaz. Bölüm 2.1'de öğrendiğiniz üzere bu aygıtlar **etiketlerle** değiştirilmelidir.

Merdiven bloklarının işlev bloğu olarak düzenlenmesi prosedürü animasyonla gösterilmiştir.

1. Bir işlev bloğunda düzenlenecek programı hazırlayın.
2. Programı giriş ve çıkış olarak ayırın ve işlev bloğu yapmak için dahili aygıtları dahili etiketlerle değiştirin.
3. Sürükleyip bırak işlevini kullanarak oluşturulan işlev bloğunu programa yerleştirin.
4. Programa yerleştirdiğiniz işlev bloğunun giriş ve çıkış etiketlerine parametreler atayın.
5. İşlev bloğundan önce ve sonra parametreyi giriş etiketine aktaran bir giriş devresi ve çıkış etiketinden parametre alan bir çıkış devresi oluşturun.



İlk akışa dönmek için düğmeye basın



2.2.1 İşlev Bloklarının Oluşturulması ve Yerleştirilmesi



MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Write MAIN (188)Step *]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG]Write MAIN (188)Ste... Global Label Setting Global1 Local Label Setting MAIN [PRG] Function/FB Lat

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
 - Global1
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - ADD_1
 - Program
 - Local Label
 - Structured Data Type

User Library

Connection Destination

Start reject pct calc

ADD_1.1

Bin1

ProductionQty:W [D3 Product on quantity]

[D0 Input qty] W:InputQty ProcARrejectValue:W [D4 Process A defect value]

[D1 Proc A reject qty] W:ProcARrejectProd ProcBRrejectValue:W [D5 Process B defect value]

[D2 Proc B reject qty] W:ProcBRrejectProd

(135) [= D100 Schedule Production volume D3 Product on quantity]

İşlev bloğu oluşturmayı tamamlayın.
Devam etmek için düğmesine tıklayın.

2.2.2 İşlev Bloğu Kitaplığının Kullanımı

Verimlilik

Kalite

Artık bir işlev bloğunu nasıl oluşturacağınızı öğrendiniz.

Mitsubishi Electric, CPU'lar, analog giriş/çıkış modülleri, ağlar ve konumlandırma modülleri gibi çeşitli kontrol programlarını işlev blokları olarak düzenlemiştir ve bunlara **"FB Library"**den **(İB Kitaplığı)** ücretsiz olarak erişebilirsiniz.

İB kitaplığının kullanılması, geleneksel olarak kullanıcıların sorumluluğunda olan modül kontrol programlarının geliştirilmesi ihtiyacını ortadan kaldırır. Bilgi sahibi olmadığınız modüller dahi kolaylıkla tanıtılabilir.

İB kitaplığını **MITSUBISHI ELECTRIC FA Website** yer alan **"MELSOFT Library Download" (MELSOFT Kitaplık İndirme)** bölümünden indirebilirsiniz.

<Önce>



<İB kitaplığını kullanırsanız>



* MELSEC-F serisi için İB kitaplığı bulunmamaktadır.

Desteklenen modeller indirme sayfasında listelenmiştir. İndirmeden önce lütfen desteklenen modeli kontrol edin.

2.3 Programı Düzeltmeden İlk Aygıt Değerlerini Değiştirme

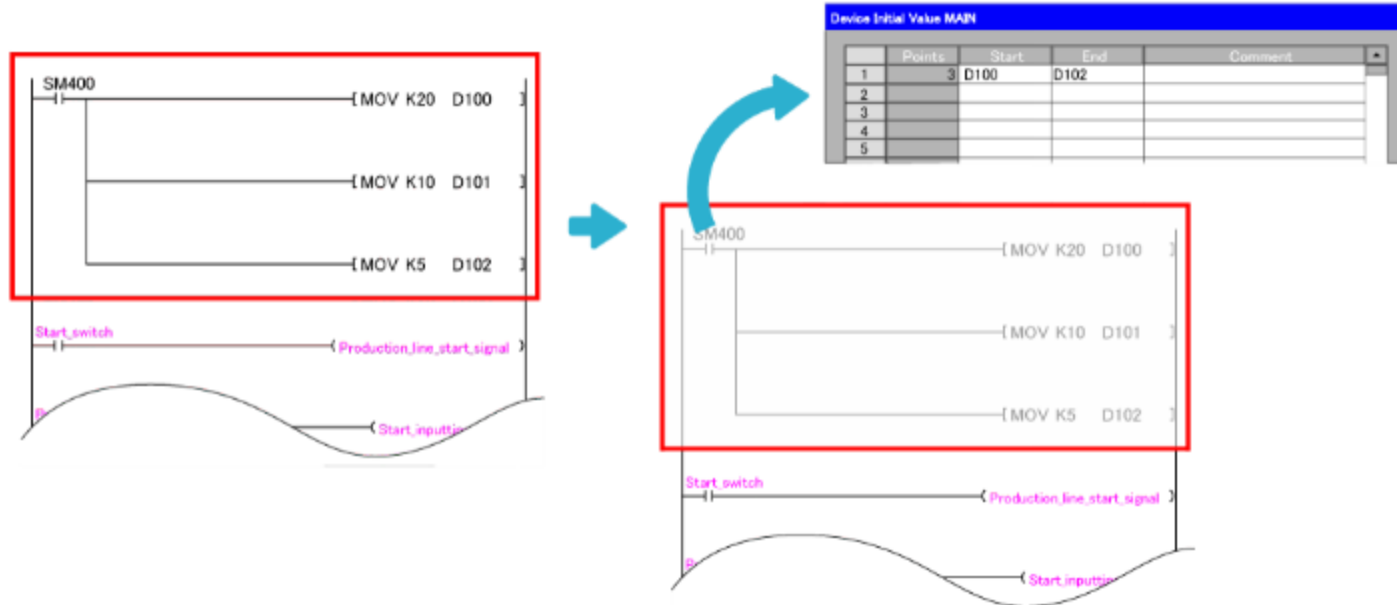
Verimlilik

Kalite

Bir aygıtın ilk değeri veya sabiti normalde ana program işlemlerinden önce MOV yönergesi kullanılarak ayarlanır. Bu durumda program, program operasyonu her değiştirildiğinde sistem uygulamasına göre doğrudan düzeltilmelidir. Bu yöntem hem zaman alır hem de düzeltme yapma sırasında düzeltme hataları veya arıza riskini taşır.

Bu tür sorunları çözmek için "**Device Initial Value**"yu (**İlk Aygıt Değeri**) kullanın. GX Works2 işlevinin kullanılması, ilk aygıt değerinin uygun biçimde yönetilmesini sağlar ve program düzeltmeleri yapma ihtiyacını ortadan kaldırır; böylece size programları daha verimli bir şekilde oluşturma olanağı tanır. Buna ek olarak, ilk aygıt değeri ayar programı ihtiyacını ortadan kaldırır ve program hacmi (bellek kullanımı) azalır ve arıza riski düşürülür.

İlk aygıt değeri için ilk değerleri ayarlamaya yönelik aygıt aralıklarını belirleyin. Gerçek ilk değerler **aygıt belleğinde** saklanır ve belirlenen aygıt aralığına atanır. Sistem uygulamalarına göre hazırlanmış aygıt belleği alanları ile ilk aygıt değerleri yalnızca atanacak aygıt belleğinin değiştirilmesi ile değiştirilebilir.



2.3

Programı Düzeltmeden İlk Aygıt Değerlerini Değiştirme

Verimlilik

Kalite

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN (166)Step *]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG]Write MAIN (166)Ste... Global Label Setting Global1 Local Label Setting MAIN [PRG]

Project

- Global1
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Type
 - Local Device Comment
- Device Memory
 - MAIN
 - ProdQty1
- Device Initial Value
 - MAIN

Project

User Library

Connection Destination


English Simple

(0) Start switch (Production_line_start_signal)

(14) Production_line_start_signal Stop_inputting_parts (Parts_input_start_flag)

(24) Production_line_start_signal (Stop_lamp)

(28) Parts_input_start_flag Parts_input_sensor

İlk aygıt değerlerini ve aygıt belleğini ayarlamayı tamamlayın.
Devam etmek için  düğmesine tıklayın.

2.4

Merdiven Programlarının Basitleştirilmesi

Verimlilik

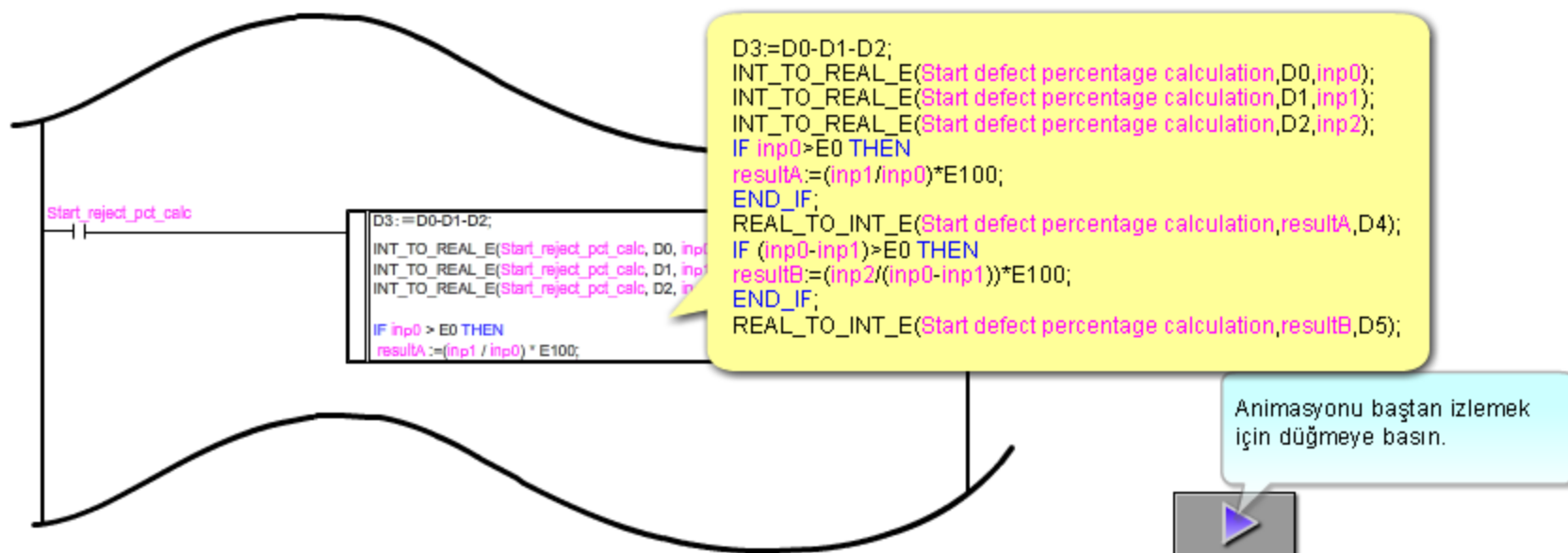
Kalite

Büyük ölçekli merdiven programında, aygıtların bağlantısı, yönergeler ve yönetilen hatlar oldukça karmaşık olabilir ve bu nedenle programın gerçekleştirdiği işlemi anlamak zor hale gelebilir. Özellikle, tek bir hat üzerinde tamamlanan formül gibi sayısal hesaplamalar yönerge kombinasyonu kullanılarak programlanmalıdır.

Bu tür sorunları çözmek için **"Inline Structured Text"**i (**Satır İçi Yapılandırılmış Metin**) kullanın. Merdiven programı kısmen **structure text (ST) language (yapı metni (YM) dili)** ile yazılmış bir programla değiştirilir. YM dili, bilgisayar yazılım programlaması için kullanılan C diline benzer bir sıra kontrolü programlama dilidir. Sayısal hesaplamalar formüller kullanılarak yazılabilir; böylece C hakkında bilgi sahibi olmayan programlayıcılar dahi satır içi yapılandırılmış metni kullanabilir.

Aşağıdaki şekilde sisteme yönelik merdiven programının bir kısmının satır içi yapılandırılmış metinle değiştirildiği bir örnek gösterilmektedir.

Karmaşık olan ikinci programın artık kolayca anlaşılabilir hale geldiğini görebilirsiniz.



2.4

Merdiven Programlarının Basitleştirilmesi



MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 114 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG]Write MAIN 114 Step Global Label Setting Global1 Local Label Setting MAIN [PRG]

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Global1
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Type
 - Local Device Commen
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

```

( 75) Start reject pct_calc
      INT_TO_REAL_E(Start_reject_pct_calc,D2,inp2);
      IF inp0>E0 THEN
        resultA:=(inp1/inp0)*E100;
      END_IF;
      REAL_TO_INT_E(Start_reject_pct_calc,resultA,D4);
      IF (inp0-inp1)>E0 THEN
        resultB:=(inp2/(inp0-inp1))*E100;
      END_IF;
      REAL_TO_INT_E(Start_reject_pct_calc,resultB,D5);

( 87) [= D100 D3 ] Production_qty_ attained
      Schedule Producti
      on produc on quant
      tion vol ity

      Production_qty_ attained_lamp

( 94) Production_qty_ attained
      D4 D101 Proc_A_reject_pct_error_lamp
      Process Process
      A defect A defect
      value thresho
      id

      D5 D102
  
```

Satır içi yapılandırılmış metin ayarını tamamlayın.
Devam etmek için ögesine tıklayın.

English Simple Q03UDE Host Station (83 N...

2.5 Anlaması ve Okuması Kolay Programların Oluşturulması

Verimlilik

Kalite

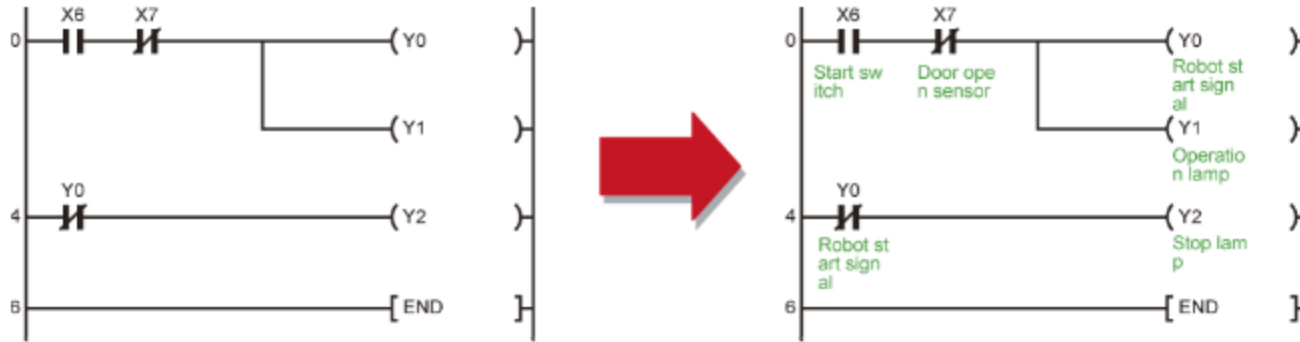
Büyük ölçekli bir programdaki kontrol detaylarını yalnızca programa bakarak anlamak sizin için zor olabilir. Bunun sonucunda aşağıdaki sorunlar meydana gelebilir:

- Program hataları yaparsınız (örn. hatalı önergelerin veya aygıtların girilmesi).
- Program hatalarının nedenlerini bulamazsınız.
- Programlama işini üstlenen kişi kontrol detaylarını anlayamayabilir.

Bu tür sorunları çözmek için **"Comment"**i (**Yorum**) kullanın.

Kontrol bilgisi ve aygıt adları gibi notlar programa eklenebilir; böylece kontrol detayları kolayca anlaşılabilir.

Programı yalnızca sizin için değil diğer kişiler için de kolayca anlaşılabilir hale getirmek için mümkün olan durumlarda programları oluşturmak amacıyla bu yorumlar girilmelidir.



GX Works2 aşağıdaki yorumların girilmesine olanak tanır:

Yorum tipi	Yorum aralığı
Aygıt yorumu	Yorum, bir aygıta eklenebilir. Bu yorum her bir aygıtın uygulamasını ve bağlı I/O aygıtının tipini belirtir.
İfade	Yorum, bir merdiven bloğuna eklenebilir. Bu yorum işlem akışının kolayca anlaşılmasını sağlar.
Not	Yorum, programdaki bobin/uygulama yönergesine eklenebilir. Bu yorum, bobin (çıkış) içeriklerinin ve uygulama yönergesinin kolayca anlaşılmasını sağlar.

2.5.1 Her bir merdiven bloğu için yorum yazma



MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 145 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

[PRG]Write MAIN 145 Step

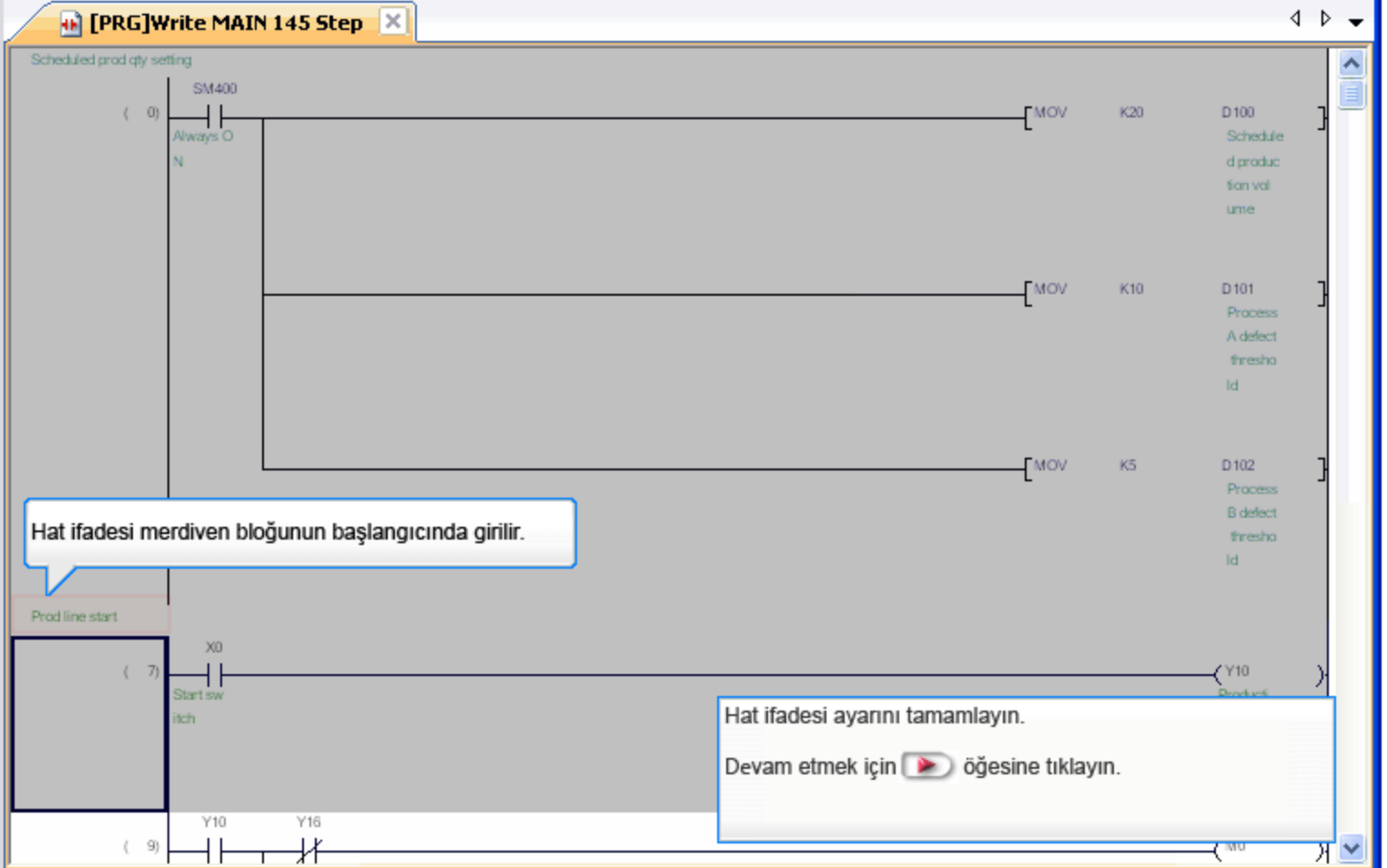
Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination



Hat ifadesi merdiven bloğunun başlangıcında girilir.

Hat ifadesi ayarını tamamlayın.
Devam etmek için ögesine tıklayın.

2.5.2

Her bir çıkış (bobin, uygulama yönergesi) için yorum yazma

Verimlilik

Kalite

MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Write MAIN 181 Step]


Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG]Write MAIN 181 Step

Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

Not ayarını tamamlayın.
Devam etmek için  öğesine tıklayın.

Bobin (M0) konumuna bir not girilmiştir.

<Start inputing parts>

M0 Parts in out start flag

Prod line start

X0 Start switch

Y10 Product on line

Y16 Stop

Y11

MOV K10 D101 Process A defect thresho ld

MOV K5 D102 Process B defect thresho ld

English Simple Q03UDE Host Station (37 Nl.5)

2.6

Özel Röleler/Kayıtlar İçeren Programların Okunmasını Basitleştirme

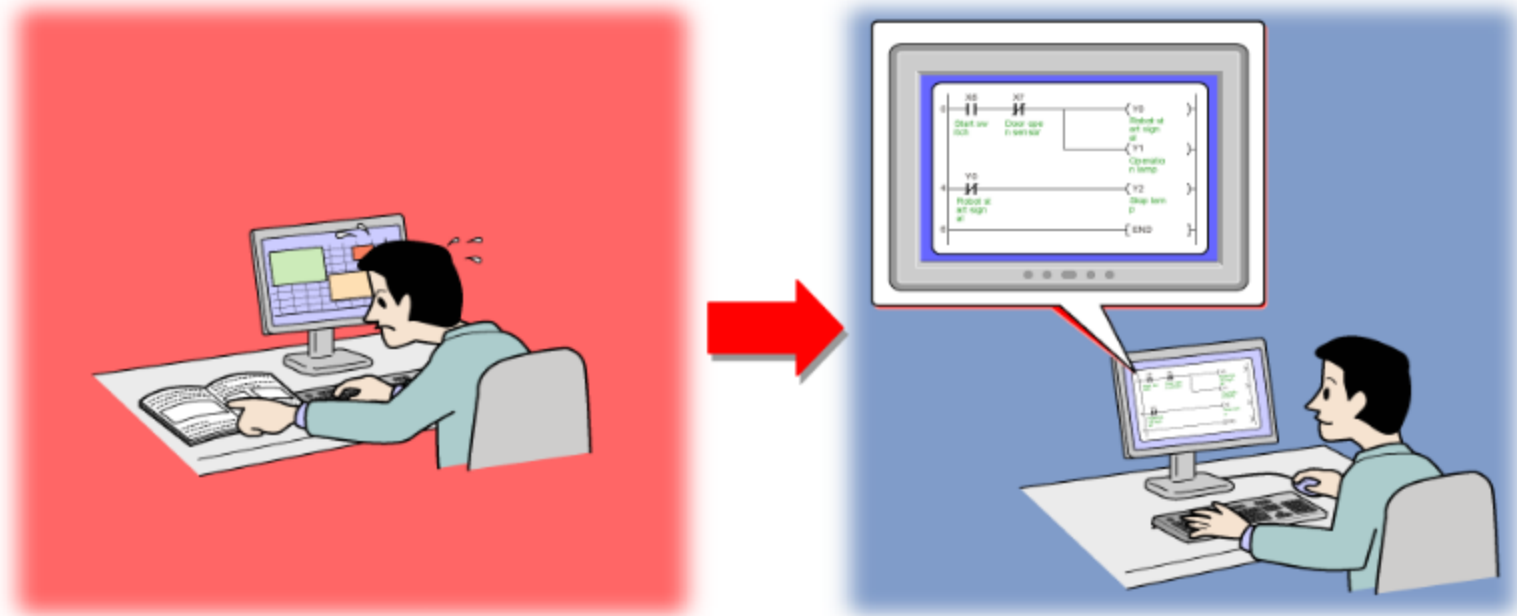
Bir programda öze röleler, özel kayıtlar ve/veya akıllı işlev modülü aygıtları kullanılıyorsa, bu aygıtların tümüne ait uygulamaları ve işlevleri anlamak zor olabilir. Programı elinizde bulunan kılavuzla okumanız gerekmektedir. Her bir aygıtta yorum eklenmesi halinde programı okumak daha kolay olsa da, birçok aygıt kullanılması durumunda yorumların eklenmesi önemli ölçüde zaman ve çaba gerektirecektir.

Bu tür sorunları çözmek için "**Sample Comment**"i (**Örnek Yorum**) kullanın.

GX Works2 özel rölelerin, özel kayıtların ve akıllı işlev modülü aygıtlarının uygulamalarını ve işlevlerini açıklayan örnek yorumlar sunar.

Bu örnek yorumların kullanımı programın okunmasını kolaylaştırmak için aygıtlara yorum ekleme işlemini çok daha basit hale getirir.

Örnek yorumlar gerektiği takdirde değiştirilebilir.



2.6

Özel Röleler/Kayıtlar İçeren Programların Okunmasını Basitleştirme

MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Write MAIN 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG]Write MAIN 194 Step Device Comment COMMENT

Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

English Simple Q03UDE Host Station (15 N...

(0) SM400 Always ON N

[MOV K20 D100 Schedule d produc tion vol ume]


[MOV K10 D101 Process A defect thresho ld]

[MOV K5 D102 Process B defect thresho ld]

(22) X0 Start sw itch (Y10 Producti on line start si)

(34) Y10 Producti on line start si Y16 Stop imp utting p arts

"Always ON" (Daima AÇIK) örnek yorumunun SM400'e uygulanıp uygulanmadığını görmek için programı kontrol edin.

Örnek yorumun otomatik ayarını tamamlayın.
Devam etmek için  ögesine tıklayın.

Bölüm 3 Hata Ayıklama

Bölüm 3'teki öğrenme adımları

Bölüm 3'te hata ayıklama için kullanılan işlevler hakkında bilgi sahibi olacaksınız. GX Works2 hataları düzeltmek için birçok farklı izleme ve hata (böcek) ayıklama aracı sunar. Bu izleme ve hata ayıklama araçlarını kullanarak hatasız ve kaliteli programlar oluşturun.

- 3.1 Yalnızca Hedef Aygıtları ve Etiketleri İzleme
- 3.2 Aygıtların ve Etiketlerin Kullanım Durumunu Kontrol Etme
- 3.3 Zaman İçerisinde Aygıt Değeri Değişimine Dair Bilgileri Toplama
- 3.4 Programı Düzeltmeden Aygıt Değerlerini Değiştirme
- 3.5 Adım Adım Program Operasyonunun Hatasını Ayıklama
 - 3.5.1 Adım Yürütme İşlevini Kullanma
- 3.6 Harici Aygıtın Operasyonunu Simüle Etme
 - 3.6.1 I/O Sistemi Ayarı için Aygıt Değerlerini Girme
 - 3.6.2 I/O Sistemi Ayarı için Zamanlama Grafiği Biçimini Kullanma

3.1

Yalnızca Hedef Aygıtları ve Etiketleri İzleme

Kalite

Bir program birçok yönerge ve aygıt kullanır.

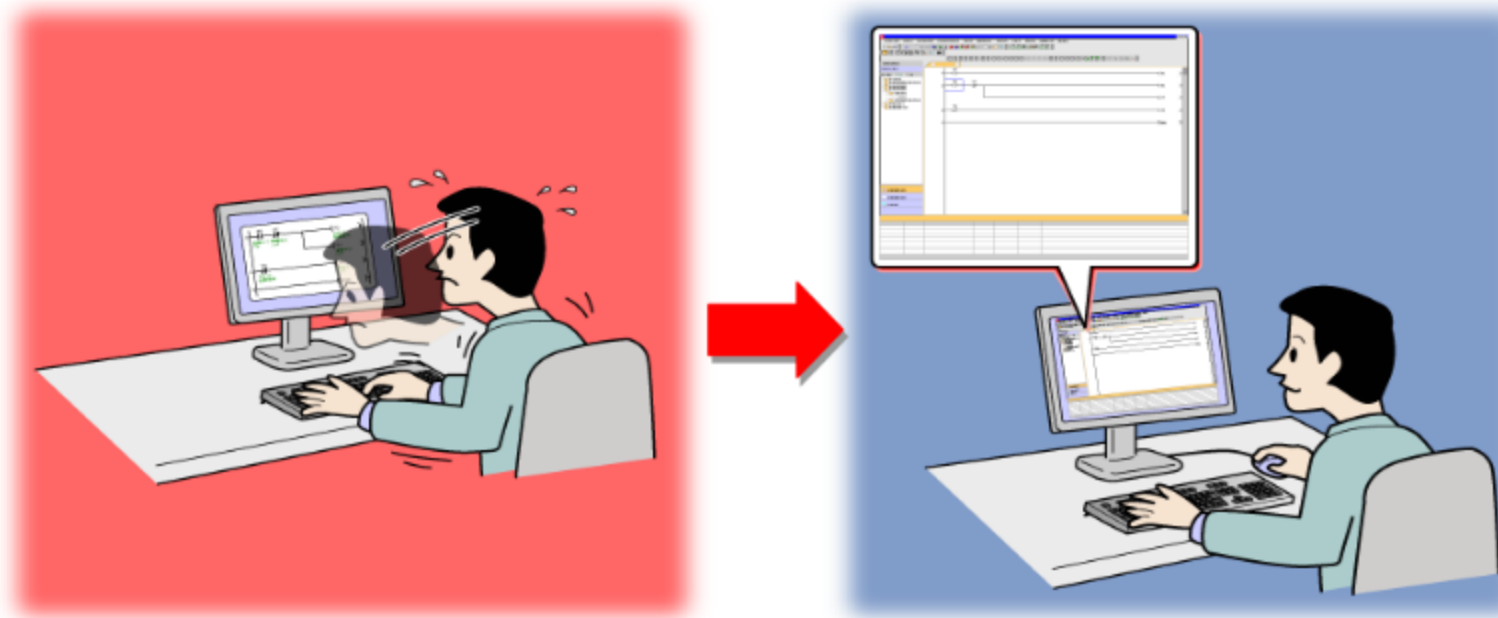
Buna ek olarak bir programın uzun dikey uzunluğu, kişisel bilgisayar monitörü ekranında tek seferde yalnızca bir kısmının görüntülediği anlamına gelir.

Bu nedenle, merdiven monitörü tek başına tüm operasyonu izleyemez.

Bu tür sorunları çözmek için **"Watch"**u (**Gözcü**) kullanın.

Bu işlev yalnızca önceden belirlediğiniz aygıtları ve etiketleri izlemeniz için kullanılabilir.

Her aralığın izlendiğinden emin olmak için iki veya daha fazla Gözcü oluşturulabilir.





3.1 Yalnızca Hedef Aygıtları ve Etiketleri İzleme

MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG]Monitor Executing M...

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
- Program
 - MAIN
 - Program

Prod qty calc

(74) M1

Start re ject pct calc

D1 D2 D6

GX Simulator 2

Tool Options

Switch

RESET RUN

LED

MODE

RUN

ERR.

USER

Watch 1

Aygıtın Watch 1 (Gözcü 1) ile kaydedilen mevcut değerleri izlenir.

Device/Label	Current Value	Data Type	Class	Device	Comment
X0	0	Bit		X0	Start switch
X1	0	Bit		X1	Parts input sensor
X2	0	Bit		X2	
X3	0	Bit		X3	

Watch (Gözcü) kaydı ve izlemesini tamamlayın.

Devam etmek için ögesine tıklayın.

English Simple Q03UDE Simulation (84 N...

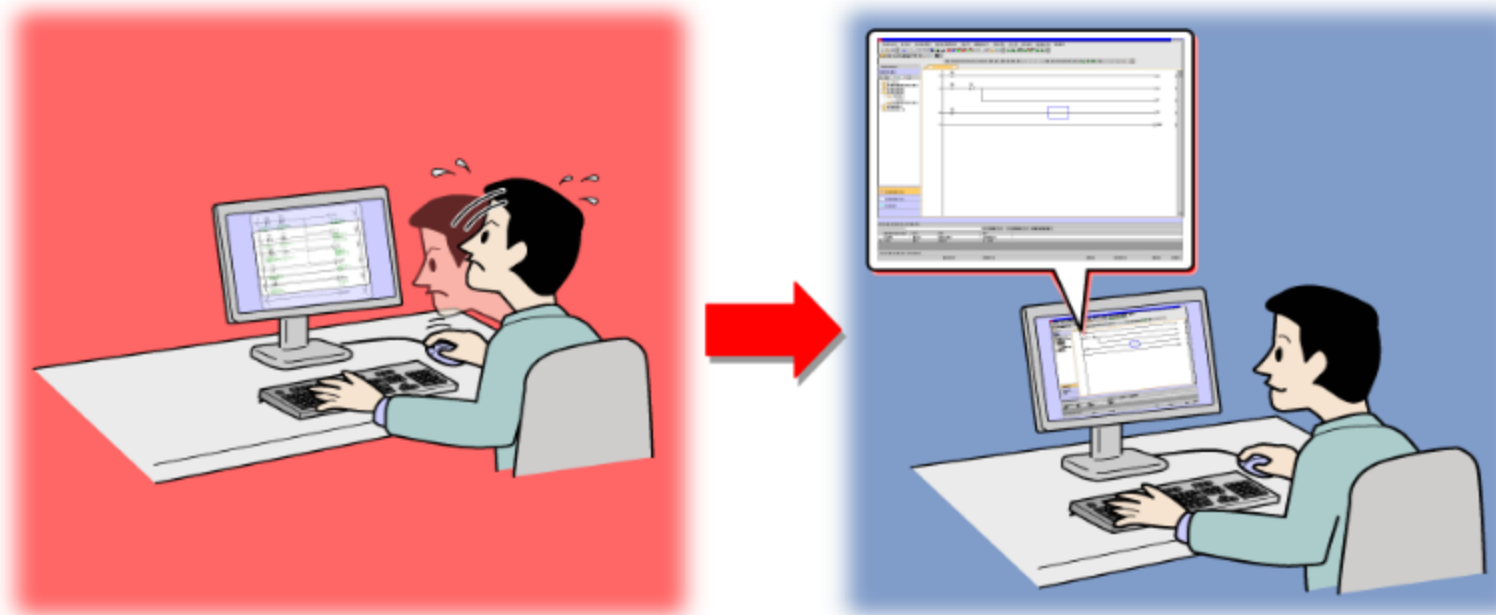
3.2 Aygıtların ve Etiketlerin Kullanım Durumunu Kontrol Etme

Kalite

Bir program farklı konumlarda aynı aygıtları ve etiketleri kullanır. Konumlar arasında karşılaştırma suretiyle bu aygıtların ve etiketlerin kullanım durumunu kontrol etmek isteyebilirsiniz.

Bunu yapmak için "**Cross reference**"ı (**Çapraz Referans**) kullanın.

Bu işlem arama koşullarını eşleştirerek aygıtların ve etiketlerin konumlarını listeler; böylece bu aygıt ve etiketleri karşılaştırabilir ve hatalı kullanıma karşı kontrol edebilirsiniz.

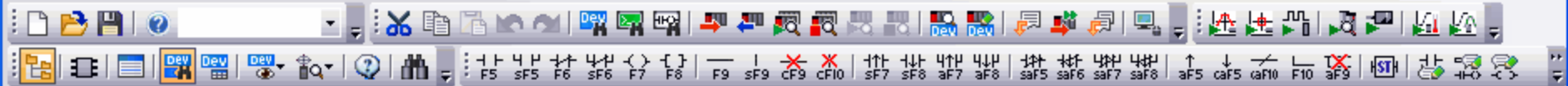


3.2 Aygıtların ve Etiketlerin Kullanım Durumunu Kontrol Etme

Kalite

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



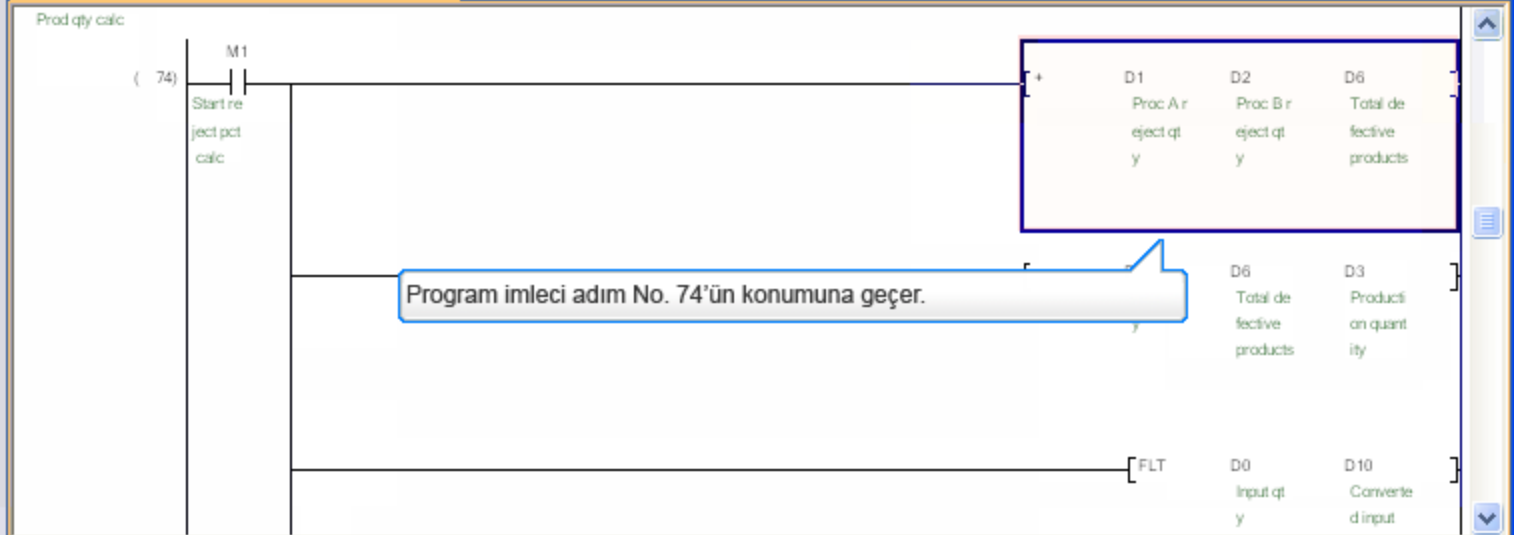
Navigation

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
- Program
 - MAIN

Project

[PRG]Write MAIN 194 Step



Cross Reference

Cross Reference Information

Condition Setting

Device/Label D1

Find

Print...

Print Preview...

Device/Label	Device	Instruction	Ladder Symbol	Position	Data Name
Filtering Condition	Filtering Con...	Filtering Condit...		Filtering Condition	
D1	D1	+P	-[]-	Step No.63	
D1	D1	+	-[]-	Step No.84	
D1	D1	FLT	-[]-	Step No.92	

5: device/cross reference information of label "D1"

Cross Reference'ı (Çapraz Referans) kullanarak tamamlayın.
Devam etmek için öğesine tıklayın.

English

Simple

Q03UDE

Host Station

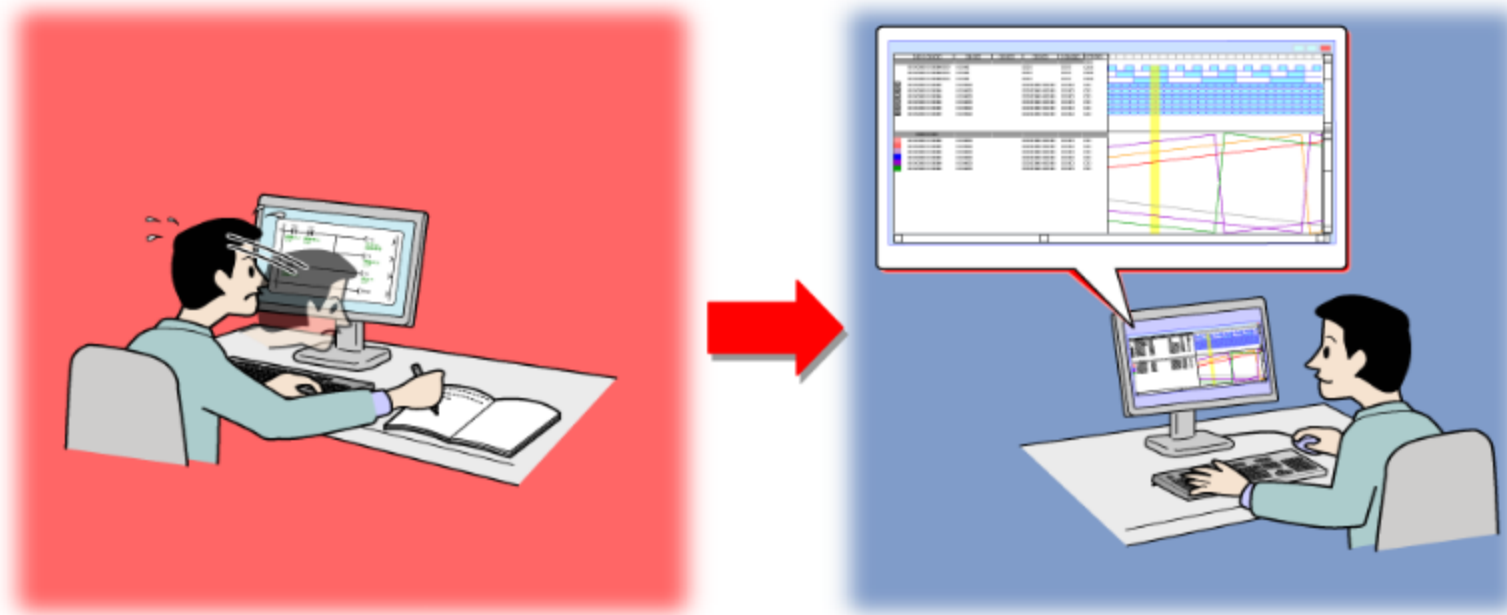
(84 N...

3.3 Zaman İçerisinde Aygıt Değeri Değişimine Dair Bilgileri Toplama

Kalite

Bir arıza halinde meydana gelen değişiklikleri kontrol etme suretiyle, aygıt ve etiket değerlerindeki değişikliklerin tasarım aralığında olup olmadığını kontrol etmek veya sorun giderme işlemi gerçekleştirme isteyebilirsiniz.

Bu durumda **"Sampling Trace"**i (**Örnekleme İzlemesi**) kullanın. (Yalnızca MELSEC-Q ve MELSEC-L serisi için)
Bu işlev zaman içerisinde aygıt ve etiket değerlerindeki farklılıkları izlemek ve kaydetmek için kullanılabilir. Kayıt dosyalara kaydedilebilir.



3.3 Zaman İçerisinde Aygıt Değeri Değişimine Dair Bilgileri Toplama



MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [Sampling Trace]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



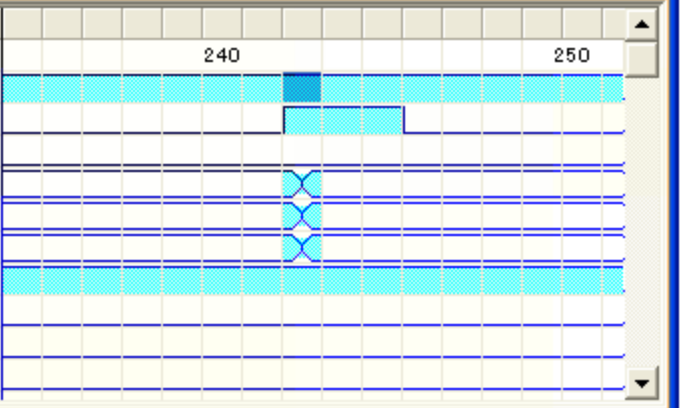
Navigation [PRG]Monitor Executing MAIN (Rea... Sampling Trace

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
- FB_Pool
- Structured Data Type
- Local Device Commen

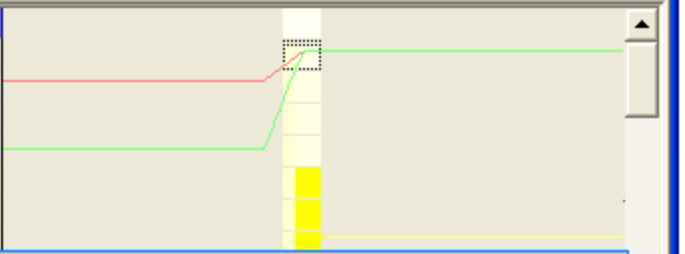
- Device Memory

Device/Label	Device	Comment	Data Type
M0	M0	Parts input st	Bit
M1	M1	Start reject pc	Bit
M2	M2	Production qt	Bit
<input checked="" type="checkbox"/> D3	D3	Production qu	Word(Signed)
<input checked="" type="checkbox"/> D4	D4	Process A de	Word(Signed)
<input checked="" type="checkbox"/> D5	D5	Process B de	Word(Signed)
X0	X0	Start switch	Bit
X1	X1	Parts input se	Bit
X2	X2	Proc A defect	Bit
X3	X3	Proc B defect	Bit



Trend Graph

<input checked="" type="checkbox"/>	D3	D3	Production qu	Word(Signed)	D 1
<input checked="" type="checkbox"/>	D4	D4	Process A de	Word(Signed)	D 1
<input checked="" type="checkbox"/>	D5	D5	Process B de	Word(Signed)	D 1



Örnekleme izi ayarını tamamlayın.
Devam etmek için öğesine tıklayın.



3.4 Programı Düzeltmeden Aygıt Değerlerini Değiştirme

Kalite

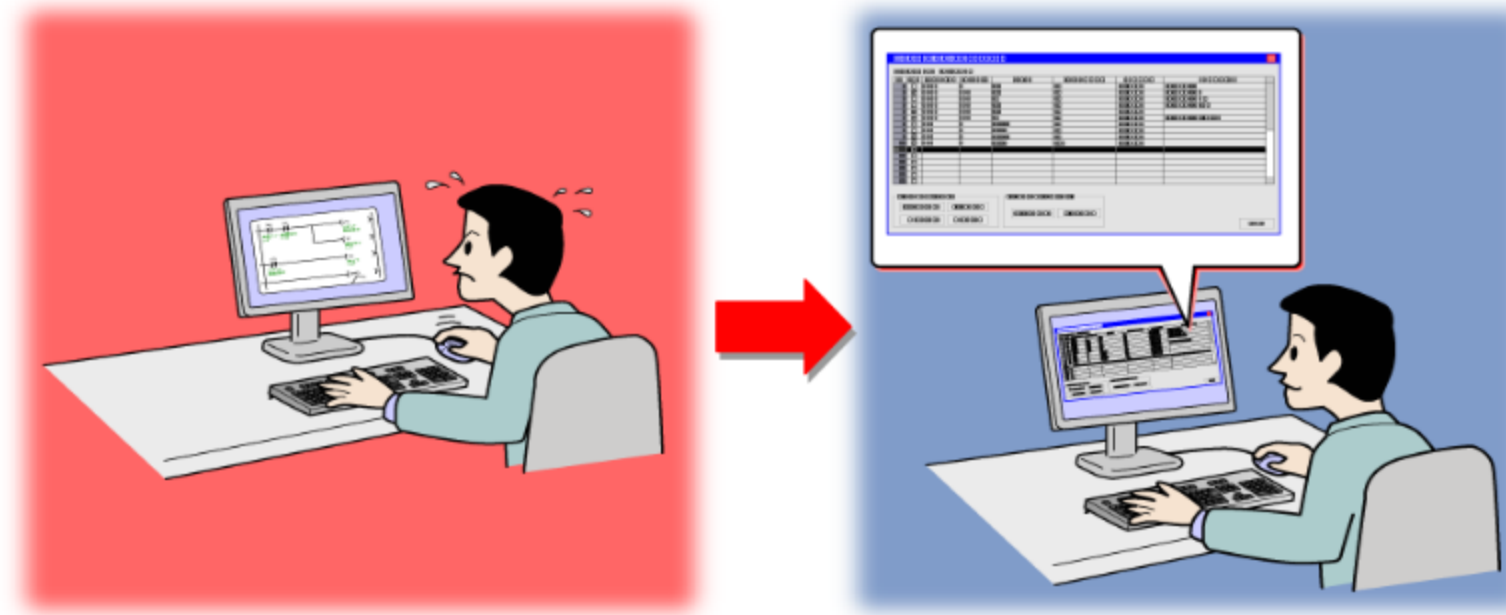
Hata ayıklama işlemini gerçekleştirirken, program operasyonundaki farklılığı kontrol etmek için aygıt değerlerini zorla değiştirmek isteyebilirsiniz.

Ancak bu, aygıt değeri her değiştiğinde programı değiştirmeniz gerektiği anlamına gelir ve bu işlem önemli ölçüde zaman ve çaba gerektirir.

Buna ek olarak, değiştirilen programı orijinal ayarlarına geri ayarlamayı unutursanız, başka bir arıza ortaya çıkabilir.

Bu durumda **"Execuational Conditioned Device Test"**i (**İşlemsel Koşullu Aygıt Testi**) kullanın. (Yalnızca MELSEC-Q ve MELSEC-L serisi için)

Bu işlev, programı değiştirmenize gerek kalmadan, önceden belirlenmiş adım numarasının yürütülmesinin ardından aygıt değerlerini değiştirir.



3.4 Programı Düzeltmeden Aygıt Değerlerini Değiştirme



MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step

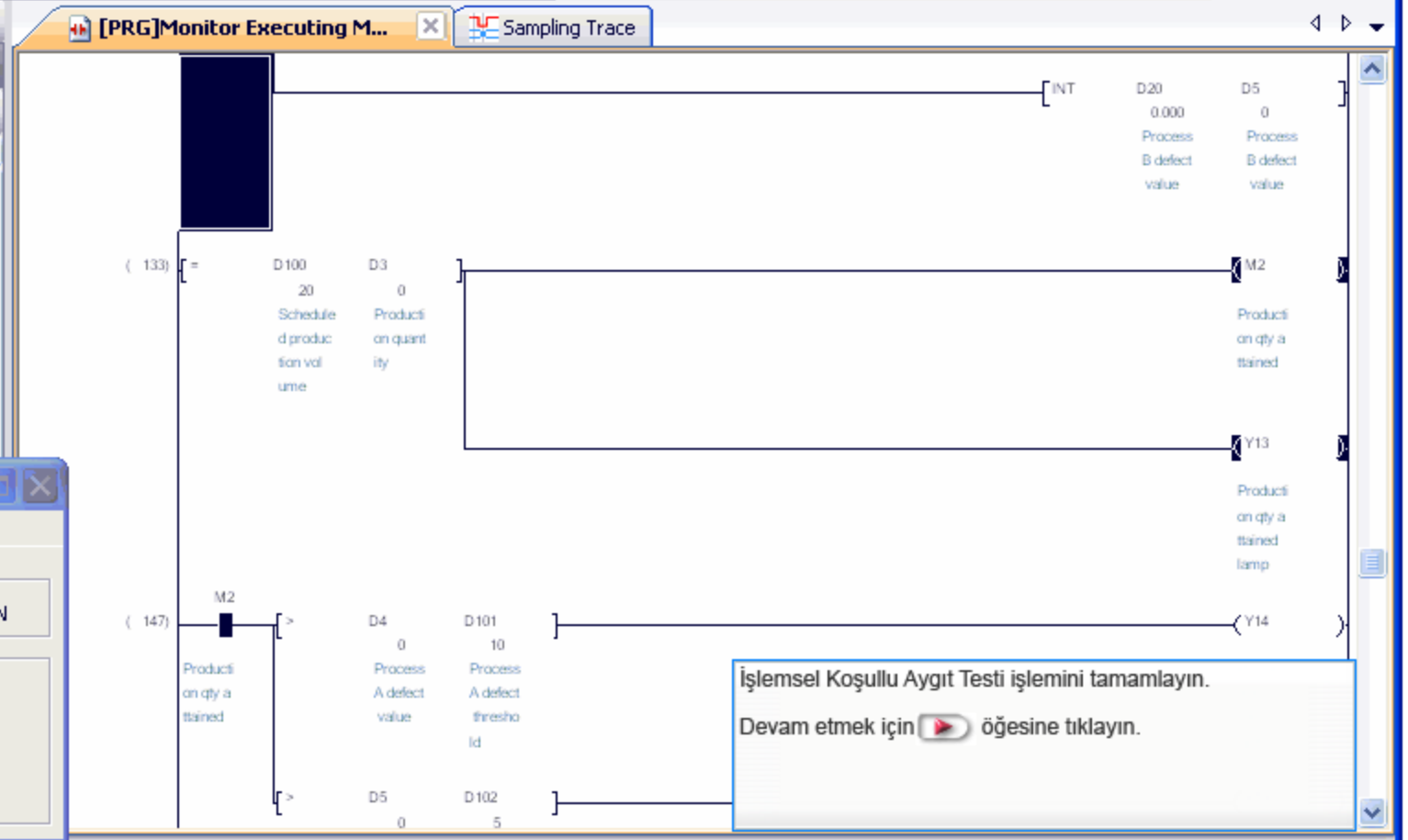
Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
- FB_Pool



GX Simulator 2

Tool Options

Switch

RESET STOP RUN

LED

MODE

RUN

ERR.

USER

İşlemsel Koşullu Aygıt Testi işlemini tamamlayın.
Devam etmek için ögesine tıklayın.



3.5 Adım Adım Program Operasyonunun Hatasını Ayıklama

Kalite

Hata ayıklama sırasında, her adımda yönerge uygulamasını onaylamak veya aygıt değerindeki değişiklikleri kontrol etmek isteyebilirsiniz.

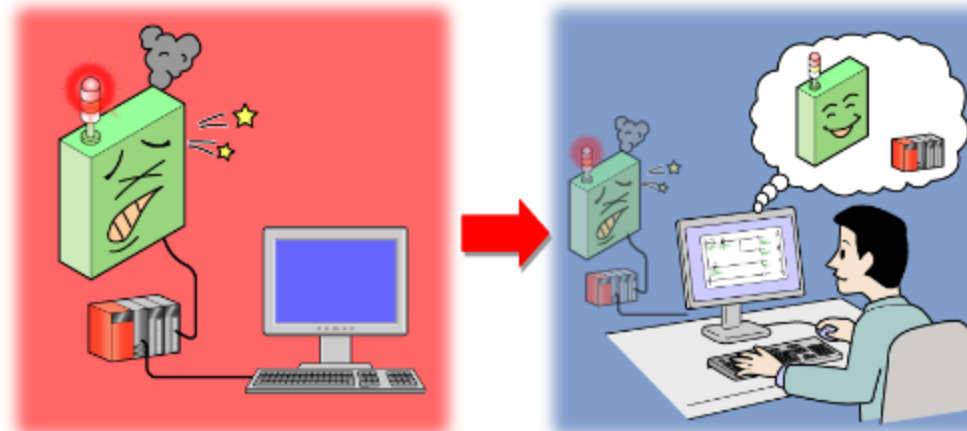
Ancak, adım adım hata ayıklama hızlı program yürütme hızından (tarama süresi) dolayı zor olabilir.

Bu sorunu çözmek için "**Step Execution Function**"ı (**Adım Yürütme İşlevi**) kullanın. (Yalnızca MELSEC-Q ve MELSEC-L serisi için)

Bu işlev tek seferde bir adımda program yürütme sağlar ve böylece adım adım hata ayıklama işlemini uygular. Adım yürütme işlevini **simulation function (simülasyon işlevi)** ile birlikte kullanın. (Adım yürütme işlevi gerçek bir PLC'de hata ayıklama işlemi için kullanılamaz.)

Aşağıdaki işlemler, adım yürütme işlevi tarafından kullanılabilir.

İşlev	Açıklama
Kesme yürütme	Bu işlev belirlenen kesme koşulları karşılanana kadar programı yürütür. Kesme koşulları karşılandığında, program yürütme durdurulur. Kesme koşullarını belirlemek için bir kesme noktası ve kesme aygıtı kullanın.
Adım yürütme	Bu işlev programı adım adım yürütür.
Kısmi yürütme	Bu işlev programı yalnızca belirlenen konumdan itibaren yürütür.



3.5.1 Adım Yürütme İşlevini Kullanma

Kalite

Sırasıyla hata ayıklama başlangıç konumu ve başlangıç koşulu olarak bir **kesme noktası** ve **kesme aygıtı** belirleyin. Ayrıca, program yürütmesini geçici olarak engellemek istediğiniz bir **geçme aralığı** belirleyebilirsiniz. (Yalnızca MELSEC-Q ve MELSEC-L serisi için)

Kesme yürütme başlatıldıktan sonra kesme koşulları karşılandığında, program yürütme kesilir. Sonrasında, **adım yürütme işlevi** ile program operasyonunu adım adım yürütürken, arızayı tespit etmek için aygıt değeri değişikliklerini kontrol edin.

<Kesme noktası>

Program yürütme işlemini kesmek istediğiniz yerde bir kesme noktası belirleyin.

Bunu, adım cinsinden belirleyin.

Tüm projede en fazla 64 kesme noktası belirlenebilir.

<Kesme aygıtı>

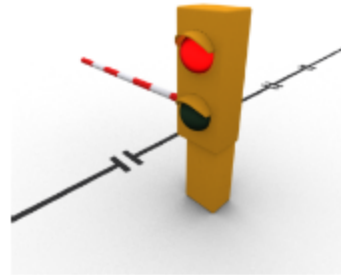
Aygıt veya etiket değeri belirlenen koşulu karşıladığında, program yürütmenin kesildiği bir kesme aygıtı belirleyin.

En fazla 16 bit ve/veya sözcük aygıtı belirlenebilir.

<Geçme aralığı>

Adım yürütme sırasında programın yürütülmeyeceği aralığı merdiven bloğu cinsinden belirleyin.

Tüm projede en fazla 16 aralık belirlenebilir.



<Geçme aralığı uygulaması>

Geçme aralığı kullanılarak arıza noktası daraltılabilir.

Kesme yürütmesini geçme aralığı belirlenmiş ve belirlenmeden gerçekleştirin.

Arıza yalnızca kesme aralığı ayarı serbest bırakıldığında meydana geliyorsa, bu durum aralığın bir arıza içerdiğine işaret eder.

3.5.1 Adım Yürütme İşlevini Kullanma

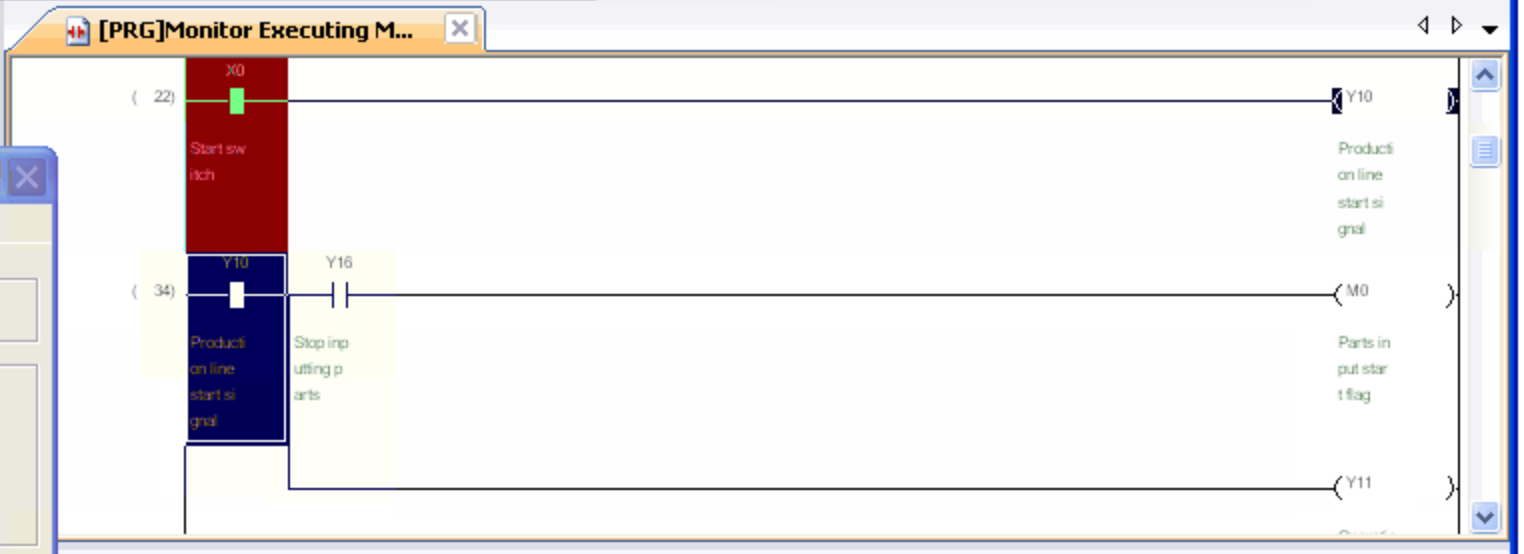


MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG]Monitor Executing M...

Project



GX Simulator2

Tool Options

Switch
 RESET STOP RUN

LED

MODE	
RUN	
ERR.	
USER	

Break Device

Devam Et

Combination
 Judge each break device (OR condition) Judge all break devices (AND condition)

Enable/Disable	Comparative Source (Device/Label)	Condition	Comparative Target (Value/Device/Label)	Comparative Type
<input checked="" type="checkbox"/>	Y10	Rising	-	Bit

Skip Range Break Point Break Device

Simulation controls: Stop, Play, Next, Previous, and a slider.

3.6 Harici Aygıtın Operasyonunu Simüle Etme

Kalite

Harici bir aygıt operasyonu ile birlikte hata ayıklama, simülasyon işlevi gibi harici bir aygıtı kullanılmadığı gelişim ortamında mümkün değildir.

Bu sorunu çözmek için harici bir aygıtın operasyonunu simüle eden bir hata ayıklama programı konvansiyonel olarak eklenir. Ancak bu, bir simülasyon programı oluşturmak için yalnızca önemli ölçüde zaman ve çaba gerektirmez; aynı zamanda operasyonun değiştirilmesi sırasında programın da değiştirilmesini gerektirir.

Bu sorunu çözmek için "**I/O System Setting**"i (**I/O Sistemi Ayarı**) kullanın.

Bu işlev hata ayıklama programı olmadan harici bir aygıtın operasyonunu simüle edebilir.

Harici bir aygıtın operasyonu ayar penceresinden kolayca belirlenebilir veya değiştirilebilir. Bu nedenle, konvansiyonel program oluşturma/değiştirme işlemine gerek kalmaz.

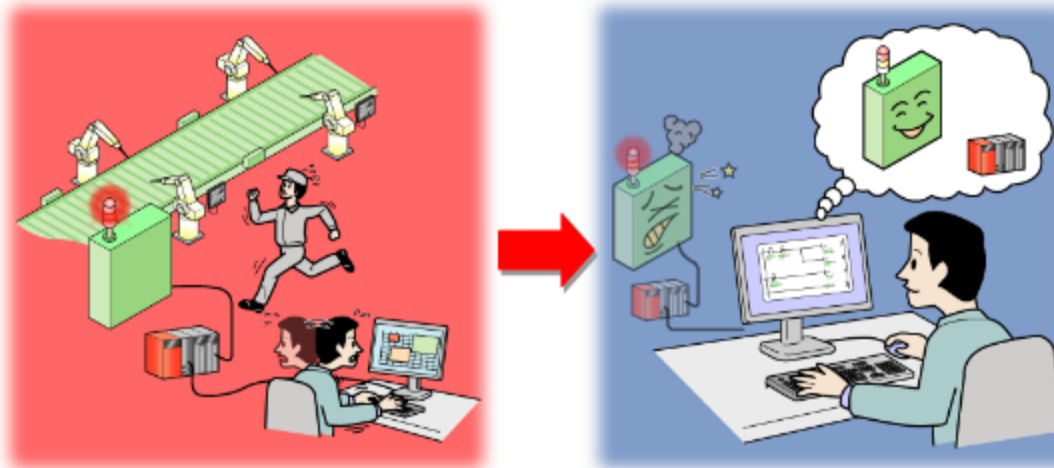
Harici bir aygıtın operasyonu aşağıda belirtilen iki yöntemle belirlenebilir:

<Aygıt değerlerini belirleme>

Belirlenen aygıt değeri, belirlenen koşullar karşılandıktan sonra zamanlayıcı ile ayarlanmış zamanda değiştirilebilir.

<Zamanlama grafiği biçiminde ayarlama>

Belirlenen zamanlama grafiği biçiminde belirlenen aygıt değeri, belirlenen koşullar karşılandığında yürütülebilir.



3.6.1 I/O Sistemi Ayarı için Aygıt Değerlerini Girme



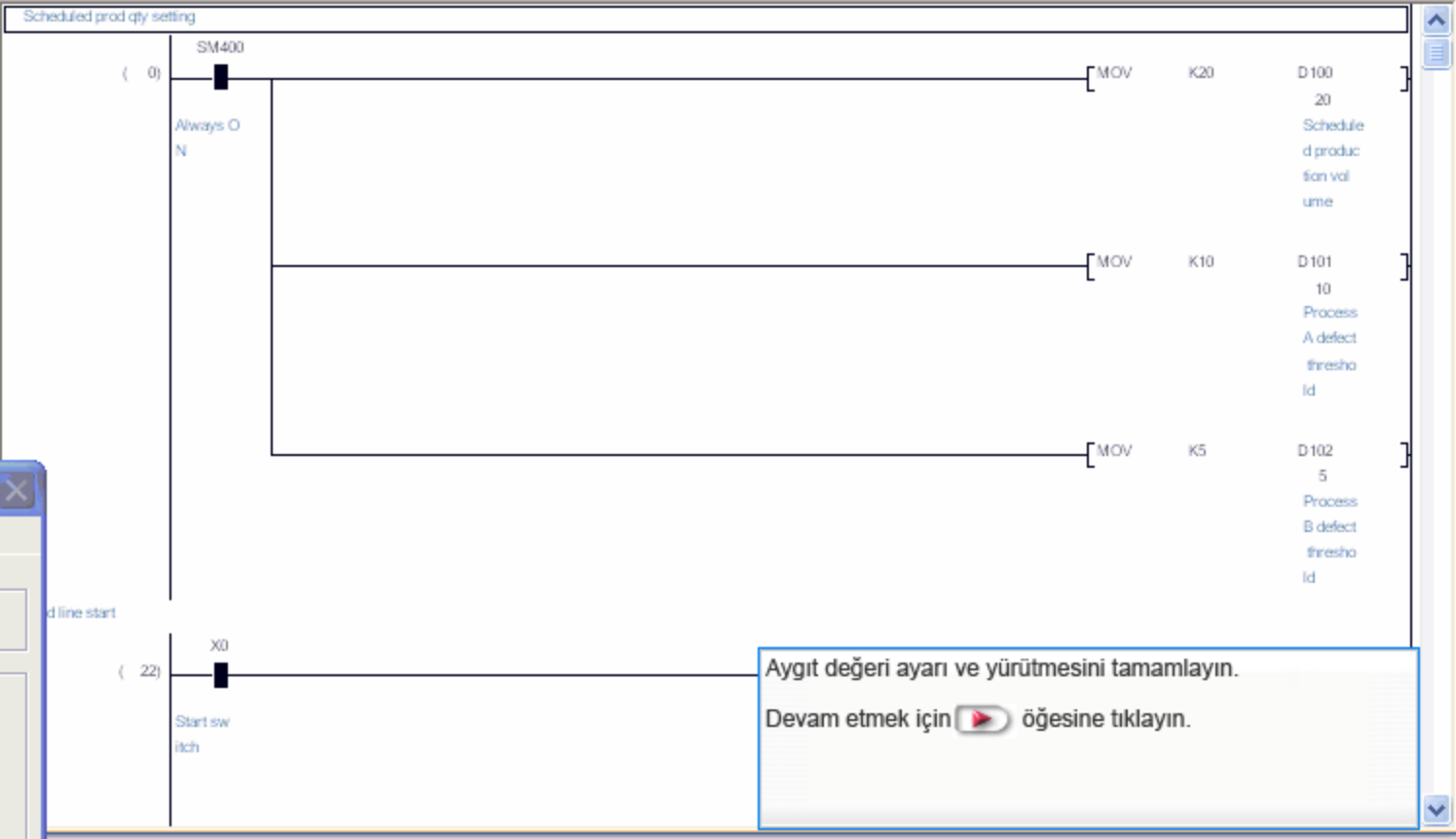
MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG]Monitor Executing M...

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Type



GX Simulator 2

Tool Options

Switch

RESET STOP RUN

LED

MODE	■
RUN	■
ERR.	■
USER	■

Aygıt değeri ayarı ve yürütmesini tamamlayın.
Devam etmek için ögesine tıklayın.

3.6.2 I/O Sistemi Ayarı için Zamanlama Grafiği Biçimini Kullanma



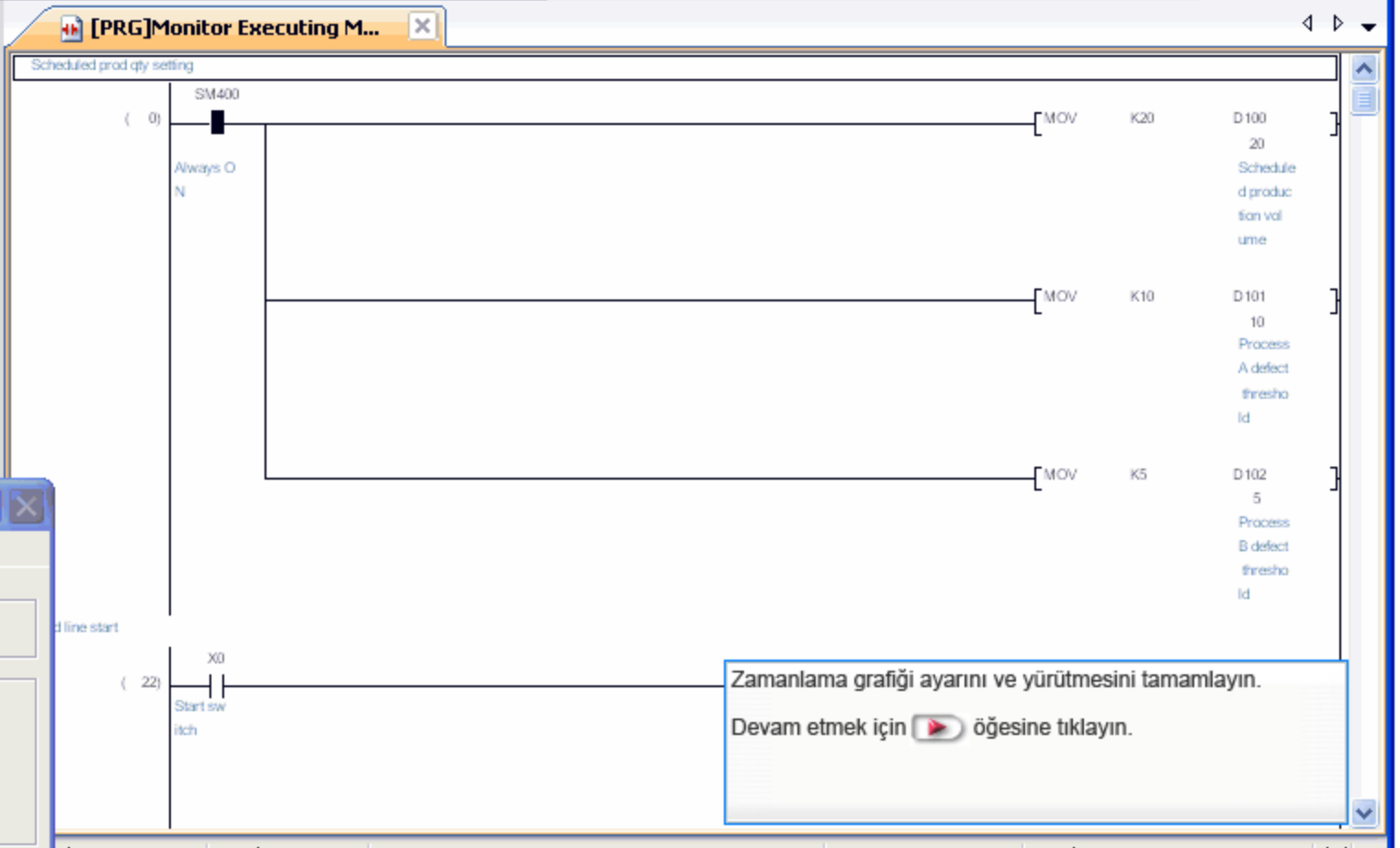
MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG]Monitor Executing M...

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Type



GX Simulator 2

Tool Options

Switch

RESET RUN

LED

MODE	■
RUN	■
ERR.	■
USER	■

Zamanlama grafiği ayarını ve yürütmesini tamamlayın.
Devam etmek için ögesine tıklayın.

Bölüm 4 Proje Yönetimi ve Güvenlik Önlemleri

Bölüm 4'teki öğrenme adımları

Bölüm 4'te, proje yönetimi ve güvenlik önlemleri için kullanılan işlevler hakkında bilgi sahibi olacaksınız.

- 4.1 Bilgi Birikiminin Sızmasını ve Programların Yetkisiz Değiştirilmesini Önleme
 - 4.1.1 Erişilebilir Verileri Her Bir Kullanıcıya Göre Sınırlama
- 4.2 Proje Yedeklemesi ve Sürüm Yönetimi
- 4.3 Programlanabilir Kontrolör ve Kişisel Bilgisayara Kayıtlı Programların Karşılaştırılması

4.1

Bilgi Birikiminin Sızmasını ve Programların Yetkisiz Değiştirilmesini Önleme

Güvenlik

Yönetim

Sıra programı stratejik açıdan önemli bilgi birikimi ve veri içerir.

Programa ait bilgi birikimi ve verilerin programın dışına sızması işletme üzerinde yıkıcı bir etkiye sahip olabilir.

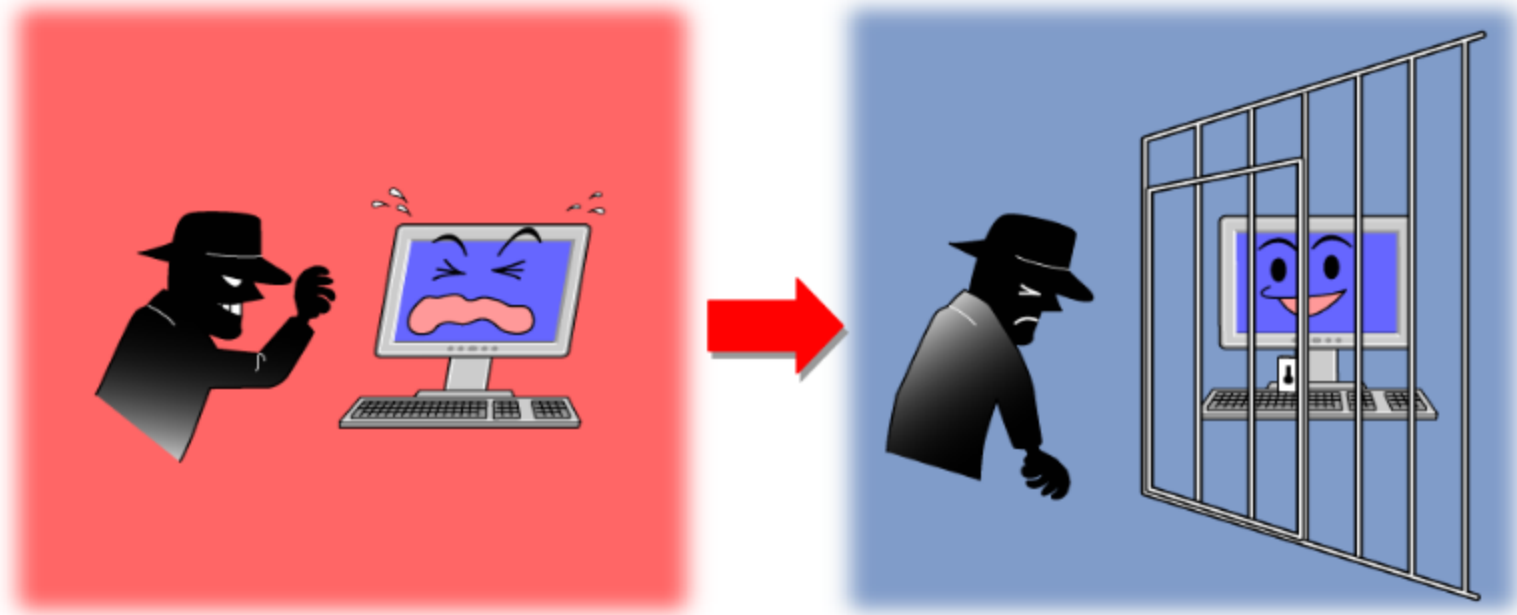
Programın yetkisiz değiştirilmesi sistemin durması gibi üretim sorunlarına neden olabilir.

Bu tür sorunları çözmek için **"Security"**yi (**Güvenlik**) kullanın.

Bu işlev, parola ile koruma altına alınmış her bir projeye erişebilecek kullanıcıları kısıtlar.

Ayrıca, her bir kullanıcının erişebileceği veya kullanabileceği veri ya da işlev aralığını kısıtlar.

Böylece bu işlev yetkisiz kullanıcıların programlara göz atmasını veya programları düzenlemesini engeller.



4.1.1

Erişilebilir Verileri Her Bir Kullanıcıya Göre Sınırlama

Güvenlik

Yönetim

Büyük ölçekli bir sıra programı genellikle çalışmayı paylaşan iki veya daha fazla programlayıcı tarafından geliştirilir. Ekip gelişimi durumunda, erişilebilir veri ve kullanılabilir işlevler aralığı, her bir programlayıcının çalışma aralığı ve beceri seviyesine göre ve her bir programcı tarafından kullanılan verilerin gizliliğine göre uygun biçimde yönetilmelidir. Bu erişim yönetimi güvenlik **erişim seviyelerinin** ayarlanması ile uygulanabilir.

<Erişim seviyesi>

Projedeki veriler için operasyon ayrıcalıkları her bir kullanıcı için belirlenebilir.

Aşağıda yer alan beş erişim seviyesi belirlenebilir.

Erişim seviyesi		Operasyon yetkisi	
Yüksek ↑ ↓ Düşük	Yöneticiler	Yönetici seviyesi	Tüm işlevleri kullanma yetkisine sahiptir.
	Geliştiriciler (Seviye3)	Geliştirici seviyesi	Güvenlik ayarları, veri erişimi ve bazı operasyonlar kısıtlanmıştır.
	Geliştiriciler (Seviye2)		
	Geliştiriciler (Seviye1)		
Kullanıcılar	Operatör seviyesi	Yalnızca proje verisine göz atma kullanılabilir. PLC CPU'sundan okuma mümkün değildir.	

4.1.1

Erişilebilir Verileri Her Bir Kullanıcıya Göre Sınırlama



MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Read MAIN (Read Only) 194 Step]

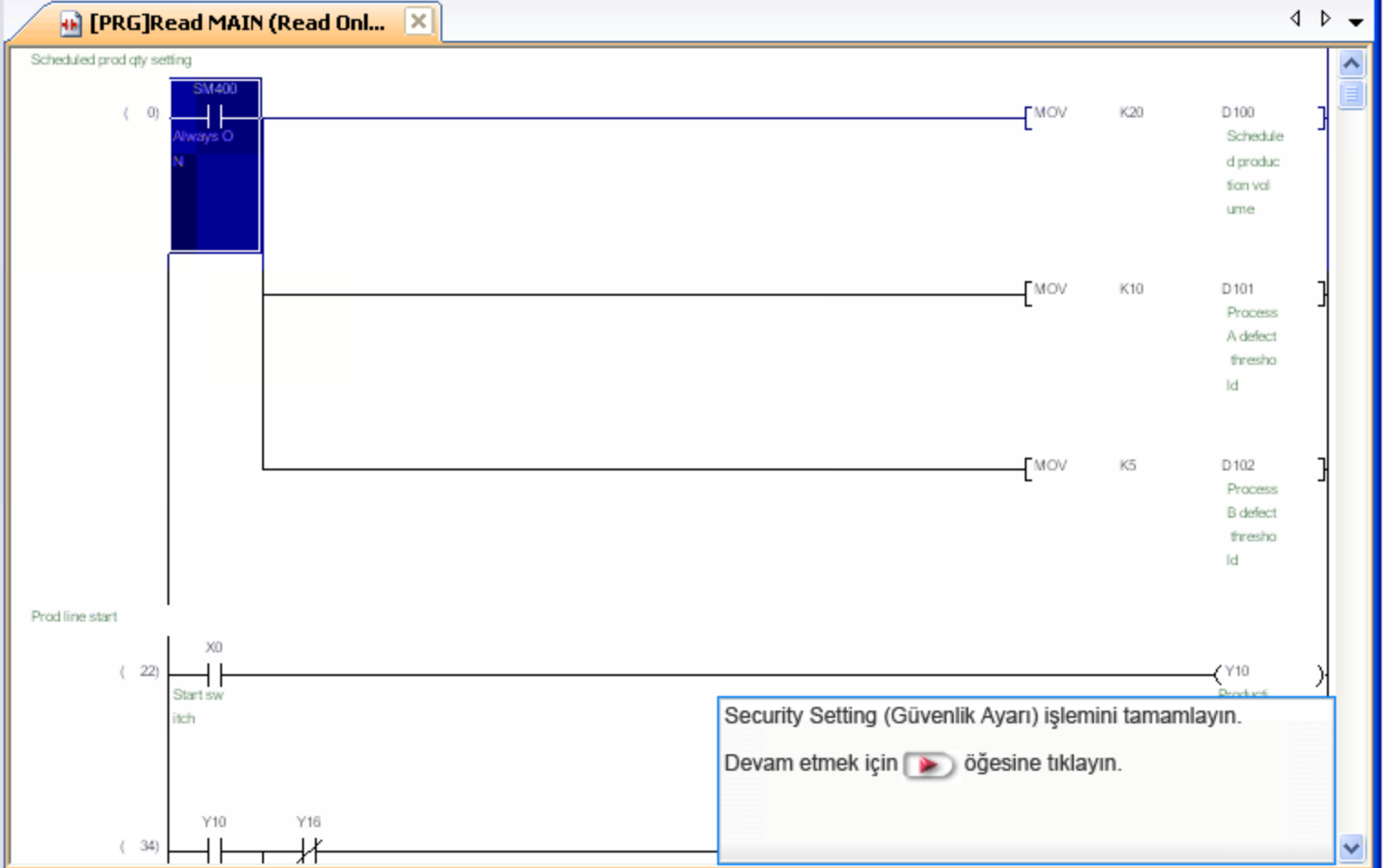
Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value



Security Setting (Güvenlik Ayarı) işlemini tamamlayın.
Devam etmek için öğesine tıklayın.



4.2

Proje Yedeklemesi ve Sürüm Yönetimi

Yönetim

Programlanabilir kontrolör CPU'sunun arızalanması nedeniyle önemli bir program kaybedilebilir.

Yedekleme programı bulunmuyorsa, hızlı geri kazanım çok mümkün değildir.

Bir yedekleme programı bulunuyor olsa dahi, sürüm yönetimi güvencesi olmadan, sürümün kaybedilen programla aynı olup olmadığını belirlemek güçtür.

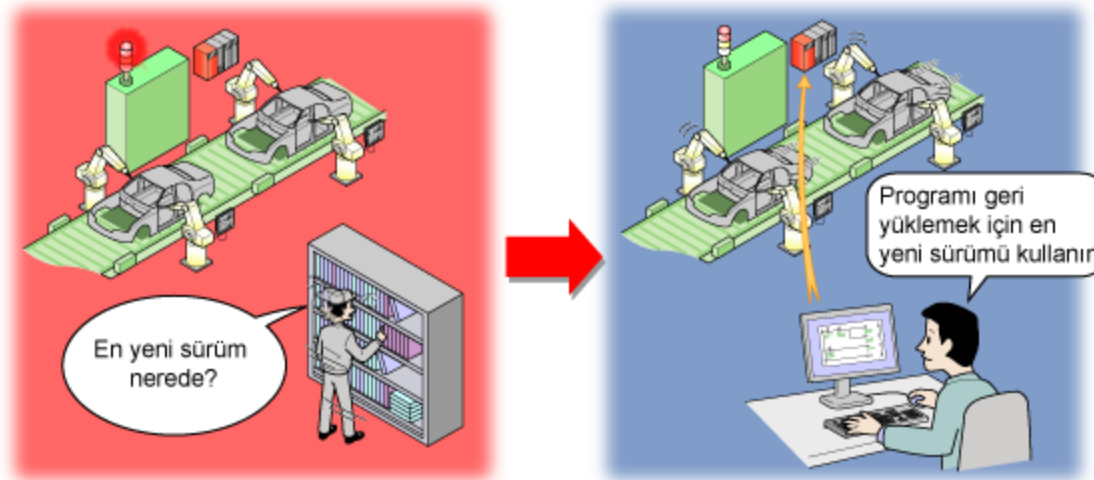
Beklenmedik durumlara karşı hazırlıklı olmak için periyodik yedekleme yapmak ve sürüm yönetiminin yapıldığından emin olmak önemlidir.

Bu sorunları çözmek için "**Change history**"yi (**Değişim geçmişi**) kullanın.

Bu işlev projeye ait en fazla 100 değişim geçmişi dizisini (geçmiş numarası, tarih/saat, kullanıcı, başlık, yorum) okuyabilir.

Aynı zamanda, kayıt sırasındaki proje verisi de yedeklenir.

Değişim geçmişi işlevi tarafından sağlanan sürüm yönetimi, kaybedilen programları geri yüklemenize, program sürümlerini doğrulamanıza ve böylece beklenmedik durumlarda kolay geri kazanım sağlamanıza olanak tanır.

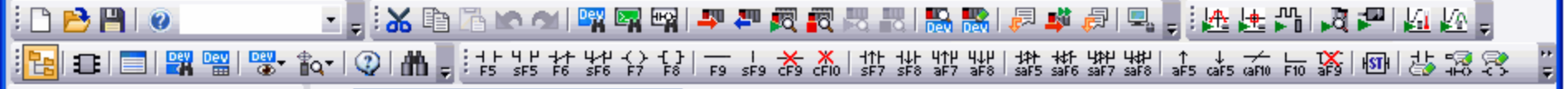


4.2

Proje Yedeklemesi ve Sürüm Yönetimi

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control (Revision No.3 : Revision B) - [[PRG]Write MAIN 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

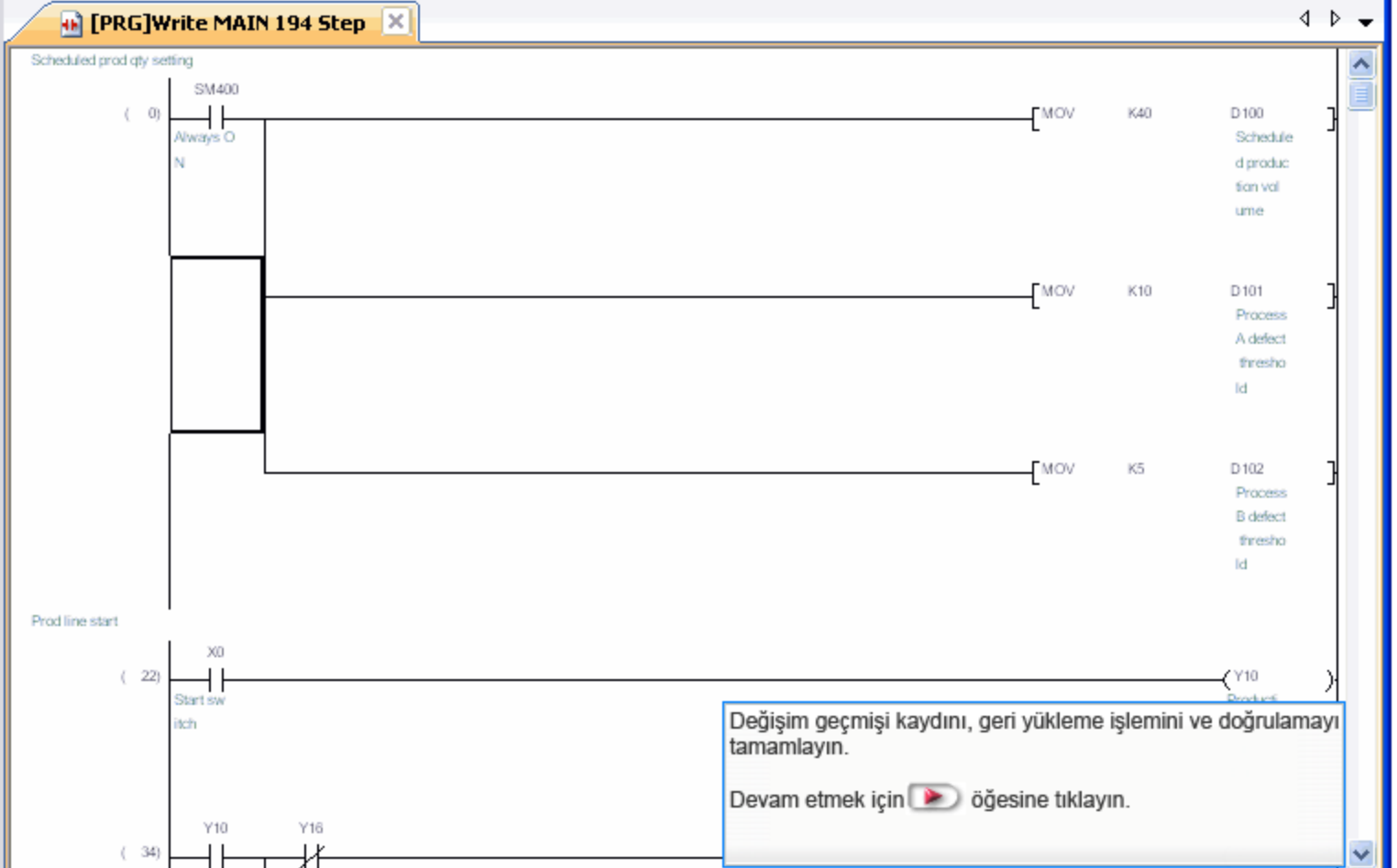
Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
- FB_Pool
- Structured Data Types
- Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

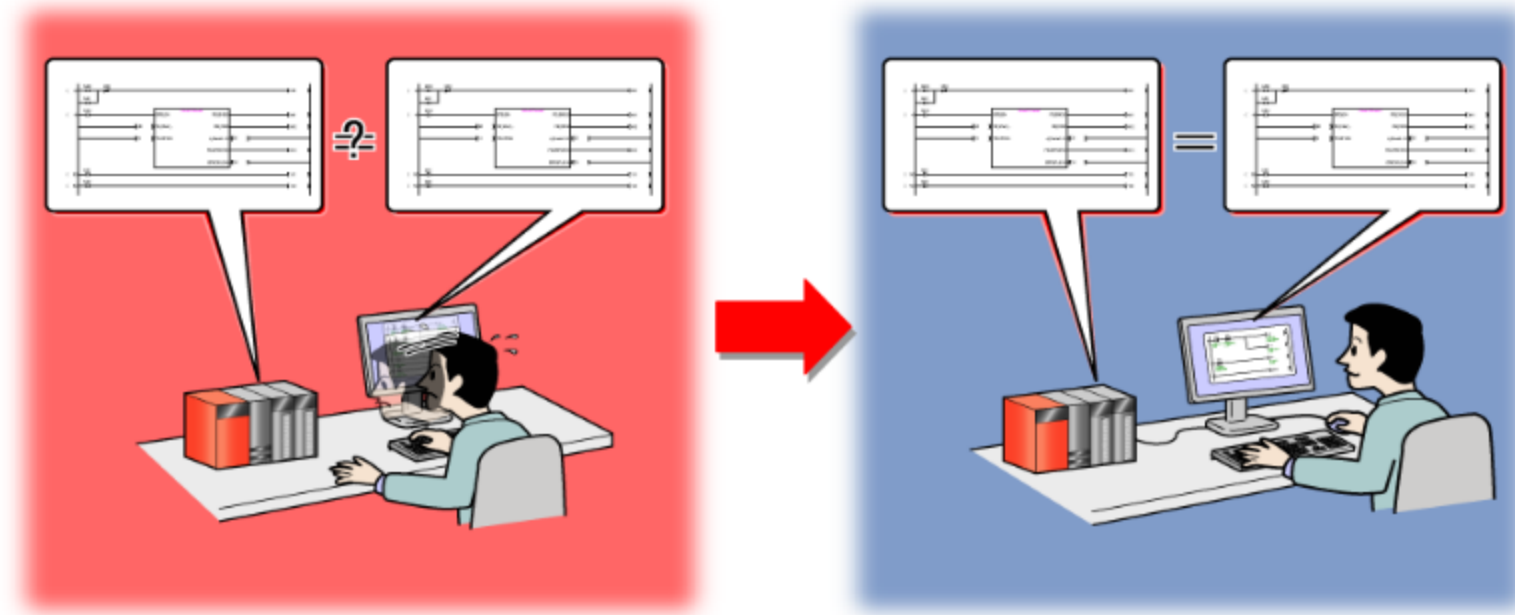


4.3 Programlanabilir Kontrolör ve Kişisel Bilgisayara Kayıtlı Programların Karşılaştırılması

Yönetim

Normalde, programlar gelişim ortamında kişisel bilgisayara kaydedilir ve ayrıca PLC'ye yazılırlar. Bu iki program her zaman aynı değildir. Aynı olup olmadıklarını görmek için yalnızca görsel kontrol yapmak hatalara neden olabilir.

Bu sorunu çözmek için **"Verify with PLC"**'yi (**PLC ile doğrula**)" kullanın. Bu işlem GX Works2 tarafından açılan programın PLC'ye yazılan programla eşleştiğini doğrulayabilir.



4.3 Programlanabilir Kontrolör ve Kişisel Bilgisayara Kayıtlı Programların Karşılaştırılması



MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Write MAIN 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation [PRG]Write MAIN 194 Step Verify Result [PLC Verify]

Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

Scheduled prod qty setting

(0) SM400
Always O
N

MOV K20 D100
Schedule
d produc
tion vol
ume

MOV K10 D101
Process
A defect
thresho
ld

MOV K5 D102
Process
B defect
thresho
ld

Prod line start

(22) X0
Start sw
itch

(34) Y10 Y16
Product

Çevrimiçi program doğrulamasını tamamlayın.
Devam etmek için öğesine tıklayın.

English Simple Q03UDE Host Station (16 N...

Test**Son Test**

Artık **PLC GX Works2 İleri Düzey Bilgileri** 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 8 soru (8 madde) yer almaktadır.

Son testi istediğiniz sayıda uygulayabilirsiniz.

Testin puanlanması

Cevabı seçtikten sonra, **Cevapla** düğmesini tıkladığınızdan emin olun. Cevapla düğmesini tıklamadan ilerlerseniz cevabınız kaybolur. (Cevaplanmamış soru olarak değerlendirilir.)

Puan sonuçları

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

Doğru cevaplar : 2

Toplam soru : 9

Yüzde : 22%

Testi geçebilmek için, soruların **%60**'ını doğru cevaplamanız gerekir.

Devam Et

İncele

Tekrar Dene

- Testten çıkmak için **Devam Et** düğmesini tıklayın.
- Testi incelemek için **İncele** düğmesine basın. (Doğru cevap kontrolü)
- Testi tekrar yapmak için **Tekrar Dene** düğmesini tıklayın.

Test**Son Test 1**

Aşağıdaki işlevlerden hangisi art arda kullanılan merdiven bloklarını paylaşılabilir bileşenler olarak kullanarak verimli programlama sağlar? (Birini seçin.)

- Satır içi yapılandırılmış metin
- Etiket
- İşlev bloğu

Cevapla

Geri

Test**Son Test 2**

Aşağıdaki işlevlerden hangisi aygıt adlarını uygulamaları ile ilgili adlara değiştirerek okunması kolay programlar oluşturur? (Birini seçin.)

- Aygıt yorumu
- Etiket
- Not

Cevapla

Geri

Test**Son Test 3**

Aşağıdaki işlevlerden hangisi her bir merdiven bloğu için işleme bilgisi sağlayarak okunması kolay programlar oluşturur? (Birini seçin.)

- Aygıt yorumu
- Hat ifadesi
- Not

Cevapla

Geri

Test

Son Test 4

Aşağıdakilerden hangisi "Verify with PLC" (PLC ile doğrula) işlevinin doğru açıklamasıdır? (Birini seçin.)

- Düzenlenen programı deęişim gemişinde kayıtlı programla karşılaştırır.
- Düzenlenen programı kişisel bilgisayara kayıtlı seçilen programla karşılaştırır.
- Düzenlenen programı PLC CPU'suna yazılan programla karşılaştırır.

Cevapla

Geri

Test

Son Test 5

Aşağıdakilerden hangisi "I/O System Setting" (I/O Sistem Ayarı) işlevinin doğru açıklamasıdır? (Birini seçin.)

- Hata ayıklama sırasında kişisel bilgisayarda harici I/O ekipmanının operasyonunu simüle eder.
- Hata ayıklama sırasında kişisel bilgisayardan harici I/O ekipmanının operasyonunu uzaktan kumanda eder.
- Hata ayıklama sırasında kişisel bilgisayarda PLC CPU'sunun operasyonunu simüle eder.

Cevapla

Geri

Test

Son Test 6

Aşağıdakilerden hangisi "Change history" (Değişim geçmişi) işlevinin doğru açıklamasıdır? (Birini seçin.)

- İlerde bağımsız bir biçimde geri yüklenebilmesi için GX Works2'nin operasyonunu adım adım kaydeder.
- İleride doğrulama ve geri yükleme sağlamak için projenin geçmiş bilgileri ve yedeklemelerini kaydeder.

Cevapla

Geri

Test**Son Test 7**

Aşağıdaki işlevlerden hangisi hata ayıklama sırasında programı değiştirmeden yalnızca aygıt değerlerinin değiştirilmesi için kullanılabilir? (Birini seçin.)

- Kesme yürütme
- I/O sistem ayarı
- İşlemsel koşullu aygıt testi

Cevapla

Geri

Test**Son Test 8**

Projenin A ve B olmak üzere iki program içerdiğini ve sizin "etiket" işlevini kullandığınızı düşünün. Aşağıda belirtilen etiket tiplerinden hangisi B programı tarafından erişilebilir? (İki tanesini seçin.)

- Global etiket
- A programı için yerel etiket
- B programı için yerel etiket

[Cevapla](#)[Geri](#)

Test TEST PUANI

Son Testi tamamladınız. Sonuçlarınız aşağıdaki alanda gösterilmektedir.
Son Testi sonlandırmak için, sonraki sayfaya geçin.

Doğru cevaplar : 0

Toplam soru : 8

Yüzde : 0%

Devam Et

İncele

Yeniden Dene

Testte başarısız oldunuz.

PLC GX Works2 İleri Düzey Bilgileri 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