

PLC

Verimli Programlama

Bu kursta, programlamanın daha verimli hale getirilmesi için yapısal programlama kullanılması açıklanmaktadır. Bu kurs, basit ladder diyagramı kullanarak programlama yapanları hedef alır.

Giriş**Kursun amacı**

Bu kurs boyunca, yapısal programlamanın avantajlarını öğreneceksiniz. Bu kursun daha verimli şekilde programlama yapmanıza yardımcı olacağını umuyoruz.

Bu kurs, takip edilen kurs için ön koşuldur:

- Programlama Basic (Temel Programlama)

Bu kursun içeriği aşağıdaki gibidir.

Bölüm 1 - Yapısal programlama neden gerekli?

Bu bölümde, daha fazla verimlilik için yapısal programlamanın gerekliliği açıklanmaktadır.





Bölüm 2 - Yapısal programlar neden bu kadar verimli?

Bu bölümde, yapısal programlamanın avantajları açıklanmaktadır.

Son Test

Geçer not: %60 veya üstü

Giriş**Bu e-Learning aracının kullanımı**

Sonraki sayfaya git		Sonraki sayfaya git.
Önceki sayfaya dön		Önceki sayfaya dön.
İstenen sayfaya ulaş		"İçindekiler Tablosu" görüntülenererek istediğiniz sayfaya ulaşabilmenizi sağlar.
Eğitimden çık		Eğitimden çıkın.

Giriş**Kullanırken dikkat edilecekler****Güvenlik önlemleri**

Gerçek ürünleri kullanarak öğreniyorsanız lütfen ilgili kılavuzlardaki güvenlik önlemlerini dikkatlice okuyun.

Bu kurstaki önlemler

Kullandığınız yazılım sürümünde görüntülenen ekranlar bu kurstakilerden farklı olabilir.

Bu kursta şu yazılım sürümü kullanılmaktadır:

- GX Works3 Sürüm 1.032J

Bölüm 1 Yapısal programlama neden gerekli?

Bu bölümde, daha verimli mühendislik için yapısal programlamanın gerekliliği açıklanmaktadır.

- 1.1 FA sektörünün güncel durumu
- 1.2 Programlama verimliliğine duyulan ihtiyaç
- 1.3 Yapısal programlarla verimli programlama
- 1.4 MELSOFT GX Works3 ile yapısal programlama

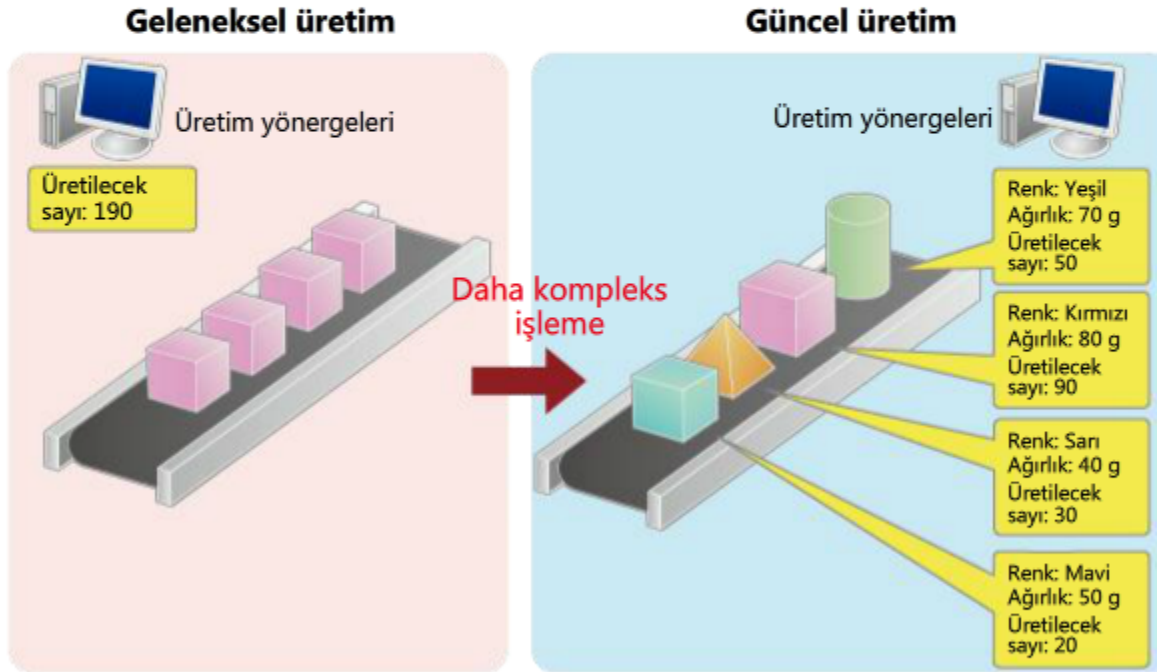
1.1

FA sektörünün güncel durumu

Günümüzdeki FA sektöründe, işleme giderek gelişip kompleks hale geldikçe üretim ekipmanları tarafından kullanılan veriler de artıyor. Ayrıntılar aşağıda açıklanmıştır.

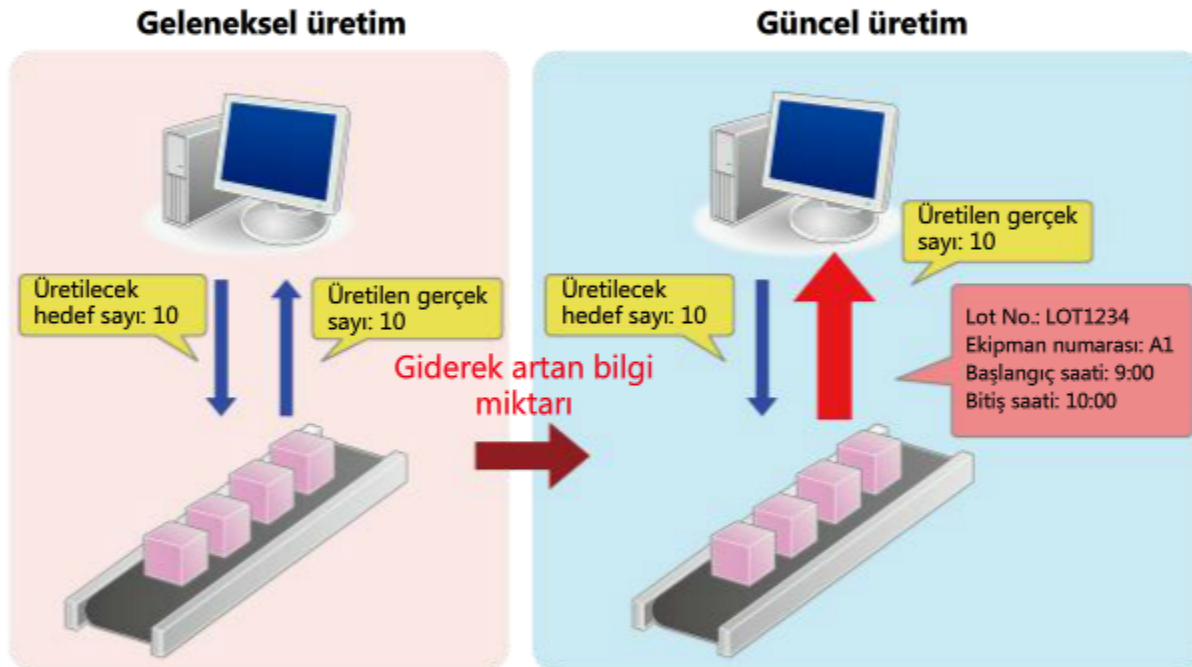
1. Daha kompleks işleme

Daha önce, tek bir üretim yönergesine göre tek bir ürün tek bir üretim hattında üretiliyordu. Ancak ürün çeşitlendirmenin ve üretim sisteminin düzenlenmesinin bir sonucu olarak birden çok ürünün tek bir üretim hattında üretilmesi giderek yaygınlaşıyor. Bu nedenle işleme giderek daha kompleks hale geliyor.



2. Giderek artan bilgi miktarı

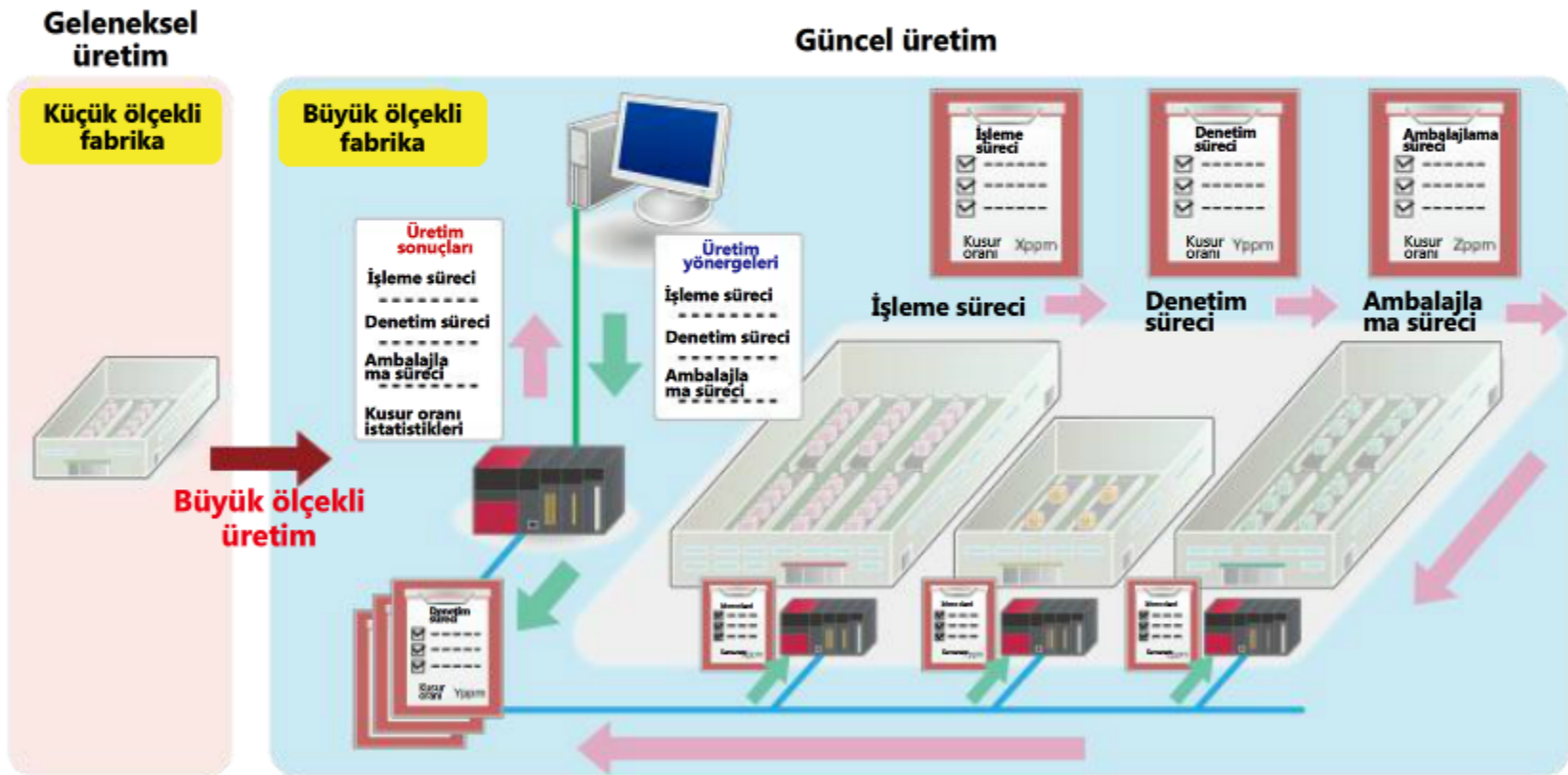
Daha önce, işletme seviyesindeki bilgisayarlar ve üretim katındaki ekipmanlar arasında paylaşılan bilgiler, basit iş yönergeleri ve üretim sonuçlarıyla sınırlıydı. Günümüzde ise izlenebilirlik sağlamak için üretim günlükleri de bu üretim kontrolü bilgilerine dahil edildi. Böylece üretim tesislerinde kullanılan bilgi miktarı da arttı.



1.1 FA sektörünün güncel durumu

3. Büyük ölçekli sistem yapılandırmaları

Büyük ölçekli fabrikaların kurulması ve geliştirilmesi giderek yaygınlaştı. Bu büyük ölçekli fabrikalarda, her üretim süreci için ağlar yapılandırılıyor ve her sürecin bilgileri işletme seviyesindeki bilgisayarlar tarafından toplanıyor. Bu ortamlarda kullanılan bilgi miktarı, geleneksel küçük ölçekli fabrikalardakine nazaran çok daha fazladır.

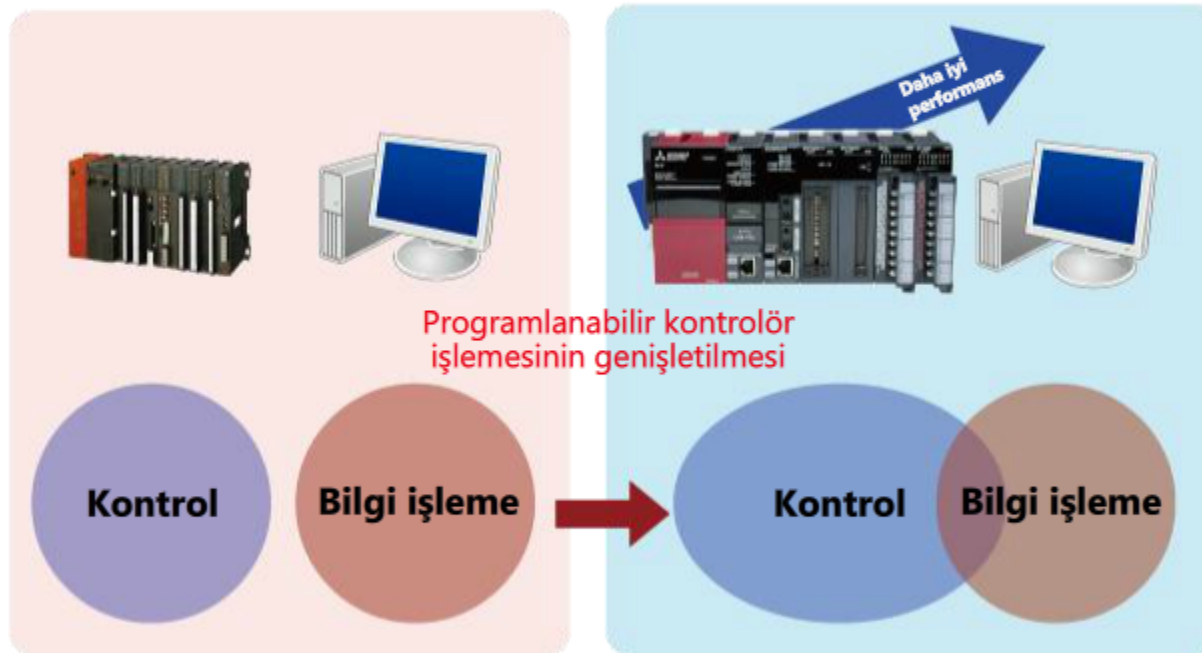


FA ortamlarında kullanılan bilgiler arttıkça, programlanabilir kontrolörler tarafından gerçekleştirilen işleme tipleri de değişti.

Önceleri, programlanabilir kontrolörler esasen kontrol için kullanılırken kişisel bilgisayarlar bilgi işleme için kullanılıyordu. Son zamanlarda ise programlanabilir kontrolörlerin hem depolama kapasitesi hem de işleme hızı arttı. Programlanabilir kontrolörler, FA ortamlarında kullanılmak üzere tasarlanmıştır ve dolayısıyla kişisel bilgisayarlara göre daha güvenilir ve sağlamdır. Bu nedenle bilgi işleme konusunda da kullanılmalarına doğru bir eğilim olmuştur.

Geleneksel üretim

Güncel üretim



Programlanabilir kontrolörün işleme alanı genişlediği için, daha kompleks işlemin yanı sıra sıralı kontrol için de artık kontrol programları kullanılıyor; böylece adım sayısı da artıyor. Bunun bazı örnekleri, işleme metin verileri, veri tabanı işlemleri ve ağ kontrolü işlemleridir.

Basit ladder diyagramları kullanılarak oluşturulan geleneksel programlarda, tüm işleme fonksiyonları net bir ayırım olmadan açıklanır. Düzenlendiğinde veya daha sonra yeni bir fonksiyon eklendiğinde, birçok adım içeren kompleks programlar aşağıdaki dezavantajlara sahiptir.

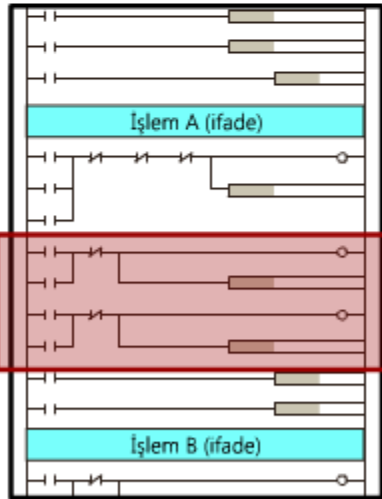
Mevcut programların kullanılması kolay değildir



Bir işlemin başlangıcının ve sonunun belirlenmesi zordur.

Kullanılabilir bölümler, yorumlara ve ifadelere bakılarak aranmalıdır.

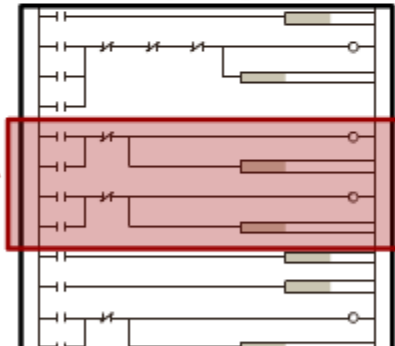
Mevcut program



İşlemin başlangıcı

İşlemin sonu

Mevcut programları kullanan yeni program



Animasyonu tekrar baştan başlatmak için sağ üst köşedeki ögesini tıklayın.

Program geliştirme tesislerinde, yeni programların geliştirilmesinde mühendislik süresinin azaltılması için benzer program kullanımı ve sık kullanılan programın standardizasyonu/paylaşımı gibi verimli programlama uygulamaları gerekir.

1.3

Yapısal programlarla verimli programlama

Geleneksel, basit ladder programlarıyla ilgili çekince, anlaşılmasının ve kullanımlarının zor olmasıdır. Yapısal programlar, programlama verimliliğini artırıp bu sorunları ortadan kaldıracaktır.

Aşağıdaki animasyonu, programların program parçalarına ayrılması konseptini açıklamaktadır.

Programlar, tek tek program parçaları seçilerek kullanılabilir.




Mevcut program



Yalnızca program parçasının seçilmesi gerekir

Mevcut programları kullanan yeni program



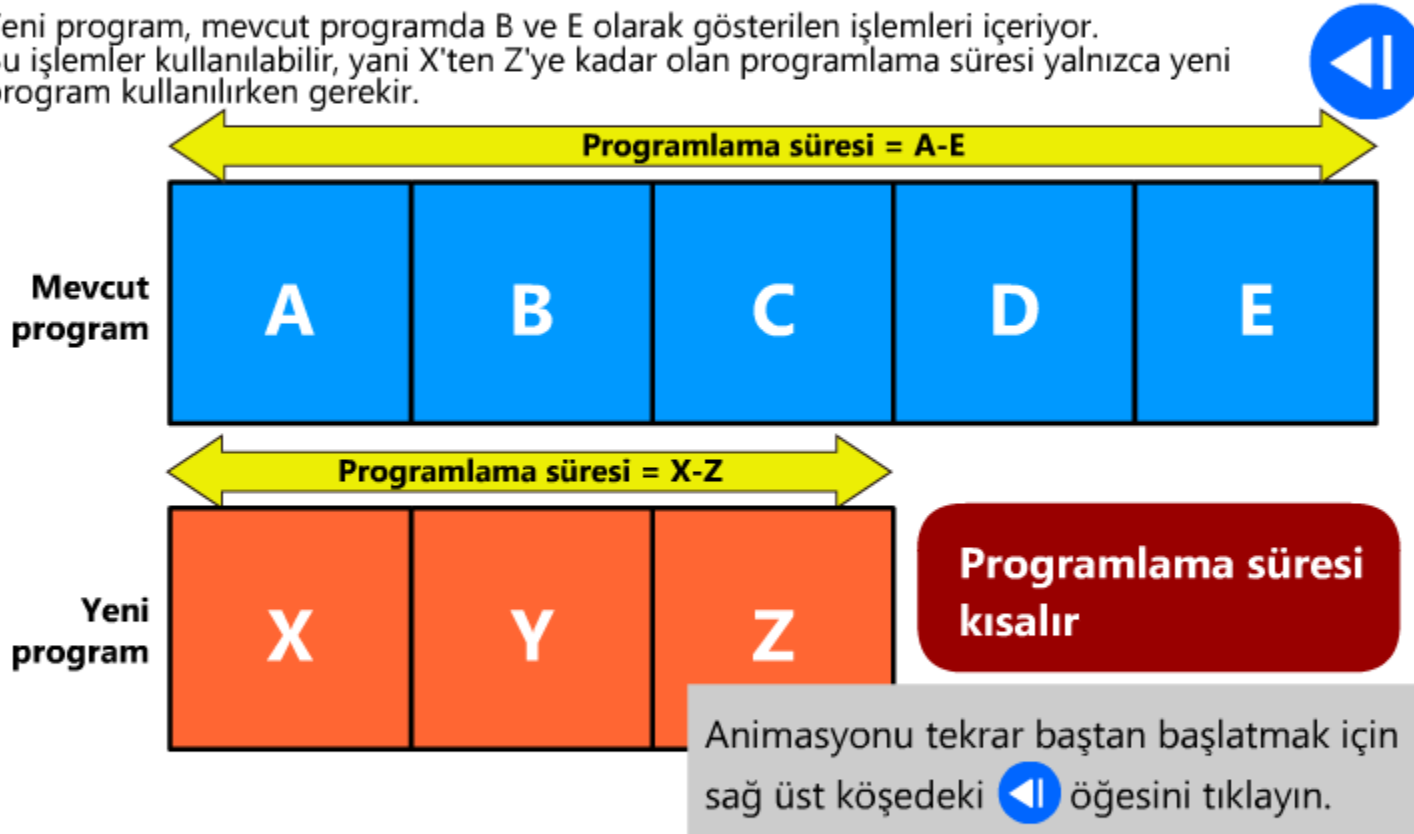
Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

Program kullanımı programlama verimliliğini artırır. Ayrıntılar sonraki bölümlerde açıklanmaktadır.

1.3.1 Program kullanımının avantajları

Mevcut programların mümkün olduğu kadar çok kullanılması programlama süresini azaltır.

Yeni program, mevcut programda B ve E olarak gösterilen işlemleri içeriyor. Bu işlemler kullanılabilir, yani X'ten Z'ye kadar olan programlama süresi yalnızca yeni program kullanılırken gerekir.



Bu sayede, mevcut programların kullanımı yeni programların programlanması ihtiyacını ortadan kaldırarak genel programlama süresini azaltır. Mevcut program kullanımını en üst düzeye çıkarmak, programlama verimliliğini artırır.

Yapısal programlar oluşturmanın ardındaki temel konsept, sonraki bölümlerde açıklanmaktadır.


1.3.2

Kullanılabilir bölümlerin verimli şekilde seçilmesi

Aşağıdaki animasyonda, hiyerarşik organizasyon konsepti açıklanmaktadır. Hiyerarşik organizasyon kullanımı, kullanılabilir bölümlerin daha verimli şekilde seçilmesini sağlar.

POU'ların bu şekilde kademelenmesi, programların küçük yığınlar halinde ve aynı zamanda yüksek bir seviyede görüntülenmesini sağlar, böylece kullanılabilir bölümler daha verimli şekilde seçilebilir.



Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

1.3.2

Kullanılabilir bölümlerin verimli şekilde seçilmesi


Bu örnekte, hiyerarşik olarak düzenlenmiş bir programın bölümlerinin kullanılmasının ne kadar verimli olabileceği gösterilmektedir.

Mevcut program

Mevcut programları kullanan yeni program



Birden çok bölüm kullanma

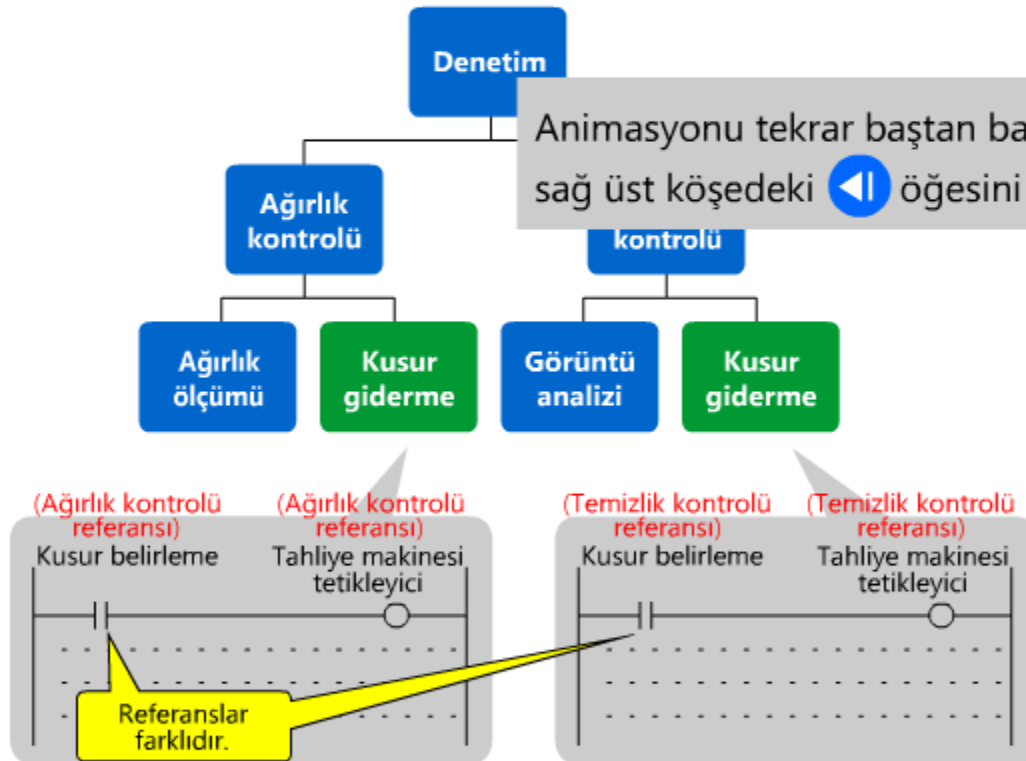
Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

1.3.3

Aygıtları yeniden atamaya gerek kalmadan program kullanımı

Yapısal programların program kullanımını nasıl kolaylaştırdığını öğrendiniz. Yapısal programlamayla, aygıtlar yerine etiketler kullanılır. Etiket kullanımı, programcıların yeni programla aygıt çatışmaları konusunda endişelenmeden program oluşturmalarını sağlar.

Aygıtlar için yineleme kontrolleri gerekmez.



1.3.4

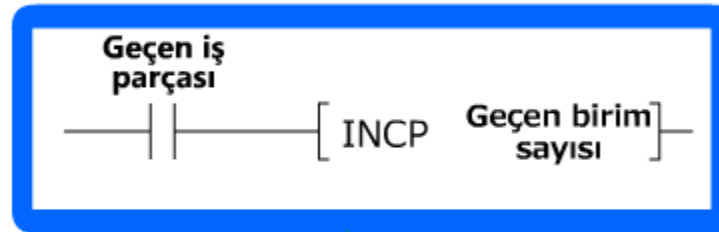
Spesifik işleme için etiket kullanarak programlama

Etiket kullanımı, programların yorumlar ve aygıtlarla uğraşmak zorunda kalmadan oluşturulmasını sağlar.


Etiket kullanımı, programların aygıtlar konusunda endişelenmeden işlem adlarıyla oluşturulmasını sağlar.



Etiket kullanımı



Sezgisel ve anlaşılması kolay

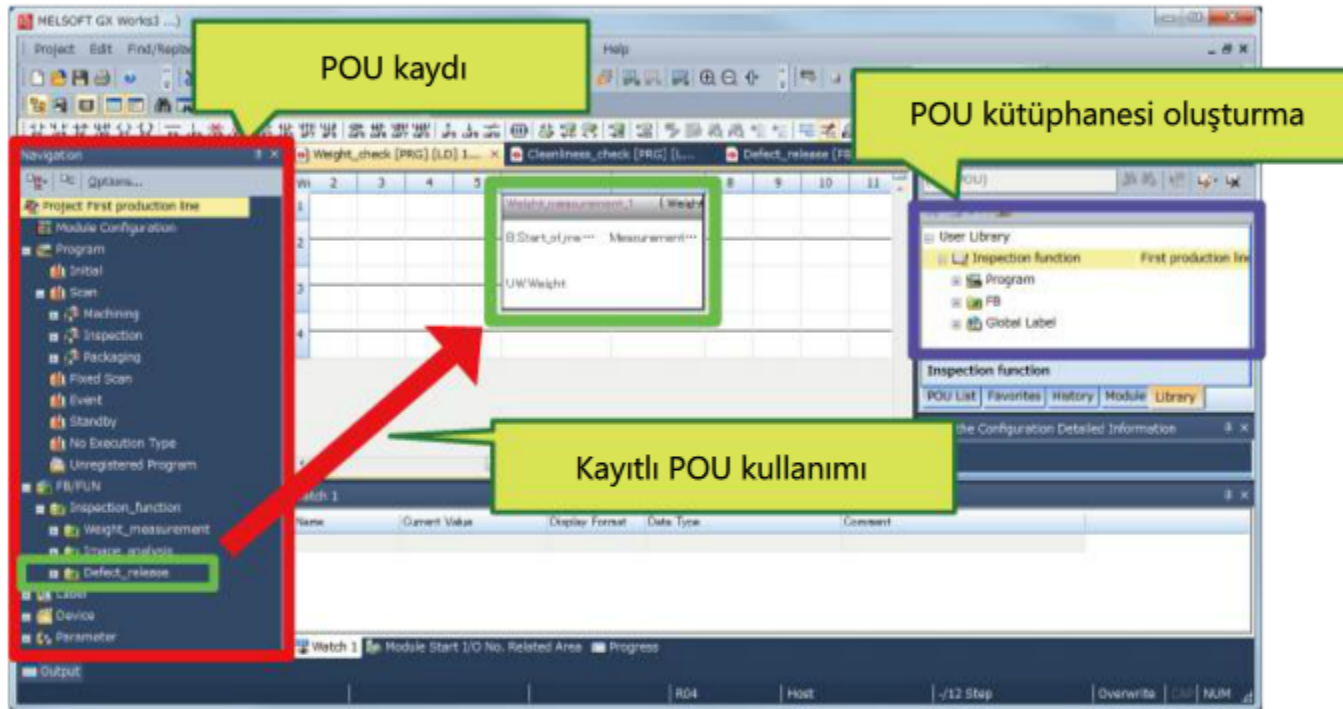
Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

1.4

MELSOFT GX Works3 ile yapısal programlama

Mühendislik yazılımı MELSOFT GX Works3, yapısal program oluşturmayı destekler.

MELSOFT GX Works3, kolay seçim için mevcut Program Organization Unit (POU) (Program Organizasyon Birimi) öğelerini göstererek program kullanımını kolaylaştırır.



MELSOFT GX Works3

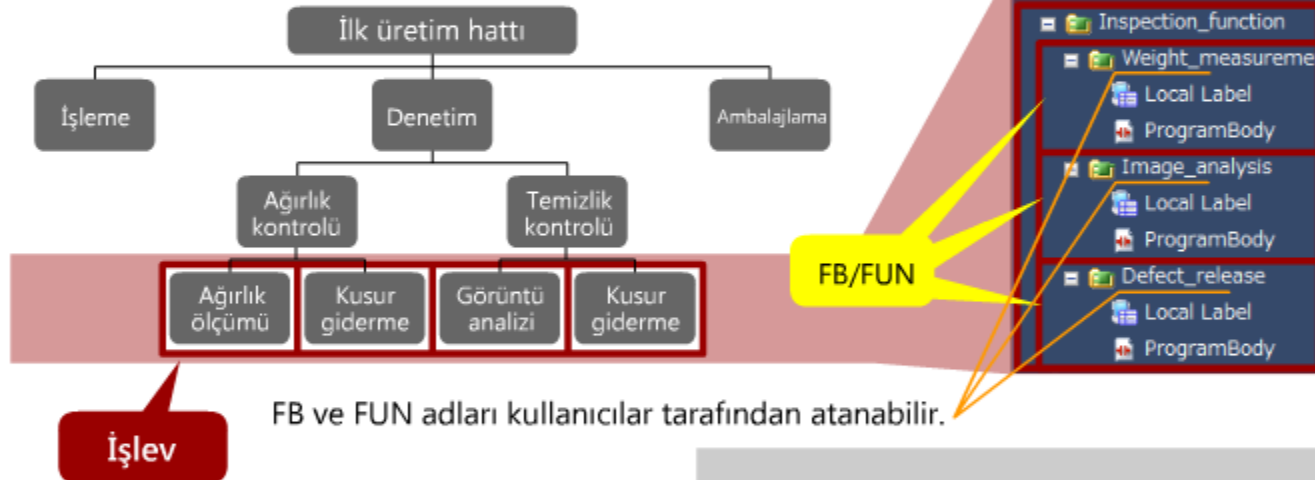
1.4.1


GX Works3'te hiyerarşik program organizasyonunun sunumu

Aşağıdaki animasyon, üretim hattı örneği kullanarak, GX Works3'te hiyerarşik program organizasyonunun nasıl sunulduğunu açıklamaktadır.

Ağırlık kontrolü işlemi, ağırlık ölçümü ve kusur giderme işlevlerini içerir.

İşlemden yer alan ağırlık ölçümü, görüntü analizi ve kusur giderme işlevleri, FB/FUN seviyesinin altındaki alt seviyelerde gösterilir.



Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

1.4.2 Program organization unit (POU) (program organizasyon birimi) kullanımı

Bu bölümde, GX Works3'te POU kullanma prosedürü açıklanmaktadır. Proje penceresinden FB/FUN seçilir ve programda düzenlenir.


Project First production line

- Module Configuration
- Program
 - Initial
 - Scan
 - Machining
 - Inspection
 - Weight_check
 - Local Label
 - ProgramBody
 - Cleanliness_check
- Packaging
- Fixed Scan
- Event
- Standby
- No Execution Type
- Unregistered Program
- FB/FUN
 - Inspection_function
 - Weight_measurement
 - Local Label
 - ProgramBody
 - Image_analysis
 - Defect_release



1	2	3	4	5	6	7	8	9	10	11	12	
						Weight_measurement_1						
						(Weight_measurement)						
						B:Start_of_measurement						
						Measurement_result UW						
						UW:Weight						
												END

1	2	3	4	5	6	7	8	9	10	11	12	
<	Weight	K100							MOV	K1	Measurement_result	
=	Weight	K100										
>	Weight	K100										

Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

1.4.3

Kütüphane kullanımı

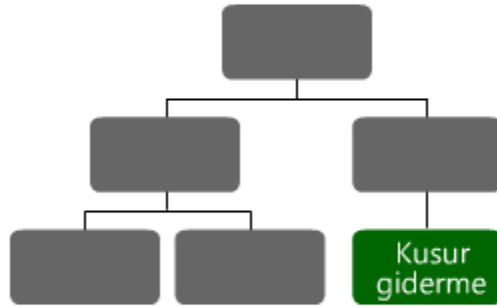
POU, kütüphane işlevi kullanılarak projeler arasında paylaşılabilir. Kanıtlanmış POU'lardan oluşan kütüphaneler oluşturularak program kalitesinin tutarlı olması sağlanır.


POU'lar, programın üzerine sürüklenip bırakılarak kullanılabilir.



2	3	4	5	6	7
				Defect_release 2 (Defect_release)	

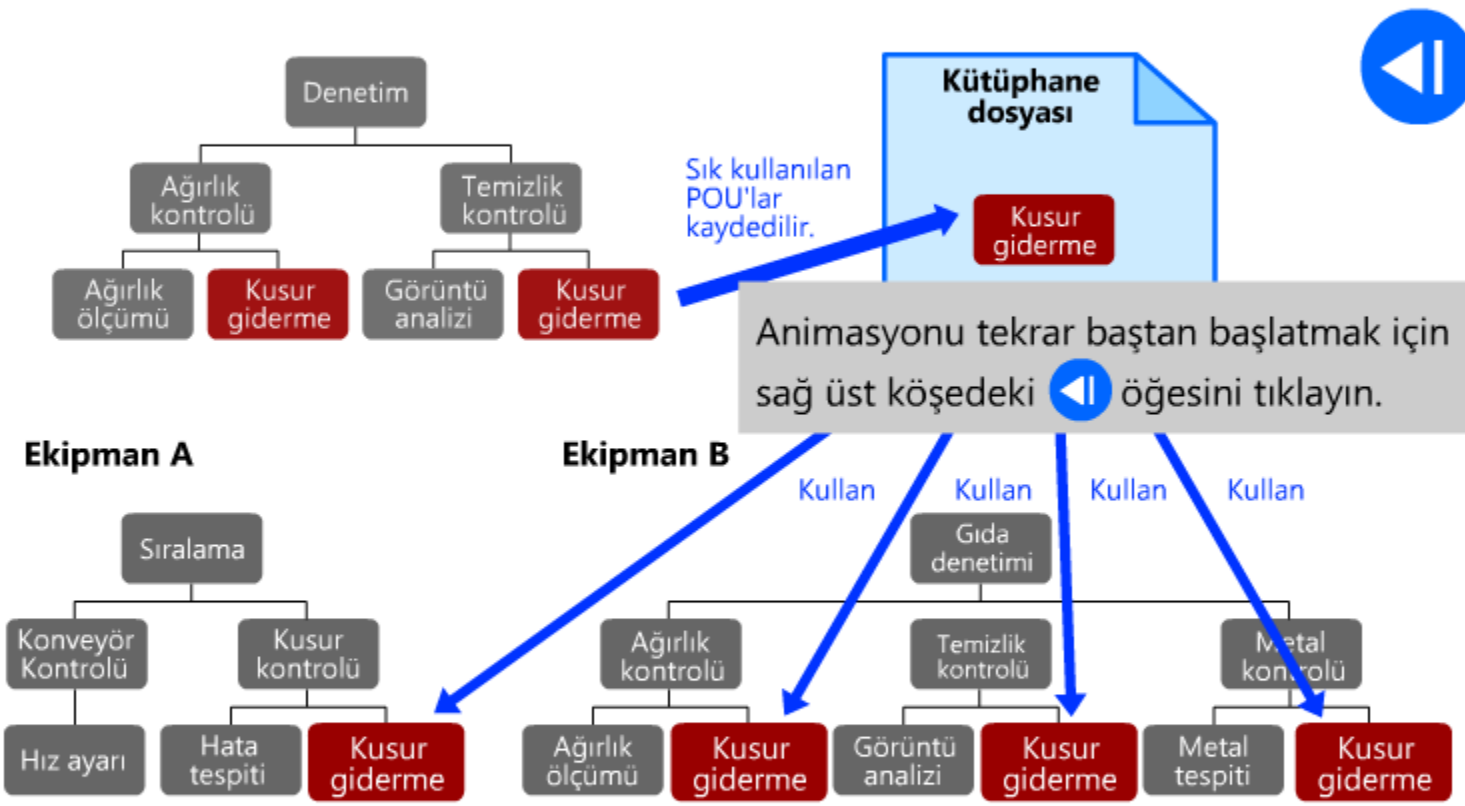
User Library	
Inspection function	First production line
Program	
FB	
Defect_release	
Image_analysis	
Weight_measurement	
Global Label	



Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

1.4.3 Kütüphanelerin etkili kullanımı

Sık kullanılan POU'lardan oluşan kütüphanelerin oluşturulması çok etkilidir. Aşağıdaki denetim programlarında, "kusur giderme" FB/FUN kütüphaneye kaydedilecek en uygun öğedir.



1.4.4


Önceden yapılmış POU kullanımı

GX Works3, birçok tipik programa yönelik POU'larla önceden yapılandırılmıştır. Bu önceden yapılmış POU'lar, modül yapılandırmasına ve MELSOFT Library'e (MELSOFT Kütüphanesi) uygun olarak otomatik olarak kaydedilmiş modül FB'leri içerir.

Mitsubishi Electric tarafından sağlanan MELSOFT Library'de kullanıcılar için faydalı birçok FB bulunur.

FB'ler, Mitsubishi Electric tarafından üretilen modülleri veya diğer üreticilerin ürünlerini kullanmanıza yardımcı olur.



Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

MELSOFT Library için, lütfen bölgenizdeki Mitsubishi Electric temsilcisine başvurun.


1.4.5 Verimli etiket girişi

Tahmini metin özelliği, etiketleri daha verimli şekilde girmenize yardımcı olur.

Bir etiket girmeye başladığınızda, girilen ilk birkaç karakterle eşleşen kayıtlı etiketlerden oluşan bir liste otomatik olarak görünür.

Ardından istenen etiket yalnızca seçerek girilir. Bu şekilde giriş hatası yapmadan aynı etiketi birden çok yere girmenizi sağlar.

Write	1	2	3	4	5	6	7	8	9
1	Start_conve...								
2	(10)								

Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

Kayıtlı etiketleri kullanarak programlama

Bu bölümün içindekiler şunlardır:

- FA sektörünün güncel durumu
- Basit Ladder kullanan programlar ile ilgili çekinceler
- Yapısal programların konseptleri ve avantajları
- MELSOFT GX Works3 ile yapısal programlama

Dikkat edilecek önemli noktalar:

Program kullanımının avantajları	Program kullanımı, gereken yeni geliştirme işlerini azaltır. Mevcut program kullanımını en üst düzeye çıkarmak, programlama verimliliğini artırır.
Programların program parçalarına bölünmesi	Kullanılabilir bölümler, programları program parçalarına bölerek ve bu parçalara anlamlı adlar verilerek netleştirilebilir.
Hiyerarşik organizasyon	Programların program parçalarına bölünmesinin yanı sıra, programlar için hiyerarşik bir organizasyonun kullanılması da kullanılabilir program bölümlerinin seçilmesini kolay ve verimli hale getirir.
Etiket kullanımının avantajları	<ul style="list-style-type: none">• Etiket kullanımı, aygıt çatışması kontrollerine ve aygıtların yeniden atanmasına olan ihtiyacı ortadan kaldırır ve program kullanımını daha verimli kılar• Programlama, spesifik işlemler için anlamlı adlar kullanıldığında daha sezgisel olur
MELSOFT'taki yapısal programlar	<ul style="list-style-type: none">• MELSOFT, yapısal programların kolayca anlaşılabilir görsel bir düzenleyicide oluşturulmasını sağlar• MELSOFT programlama verimliliğinin artırılmasına yardımcı olur

Bölüm 2 Yapısal programlar neden bu kadar verimli?

Bu bölümde, yapısal programlamanın avantajları açıklanmaktadır. Programlama süreci, aşağıdaki aşamalara ayrılarak açıklanacaktır.

- Program düzenleme: Programı oluşturma ve düzenleme
- Değerlendirme testi: Programın doğru çalıştığını onaylama
- Arama ve düzeltme: Hataları tanımlama ve düzeltme

2.1 Program kullanarak programlama süresini kısaltma

2.2 Yeniden aygıt atama sürecini kaldırma

2.3 Kazara yapılan değişiklikleri önleme

2.1

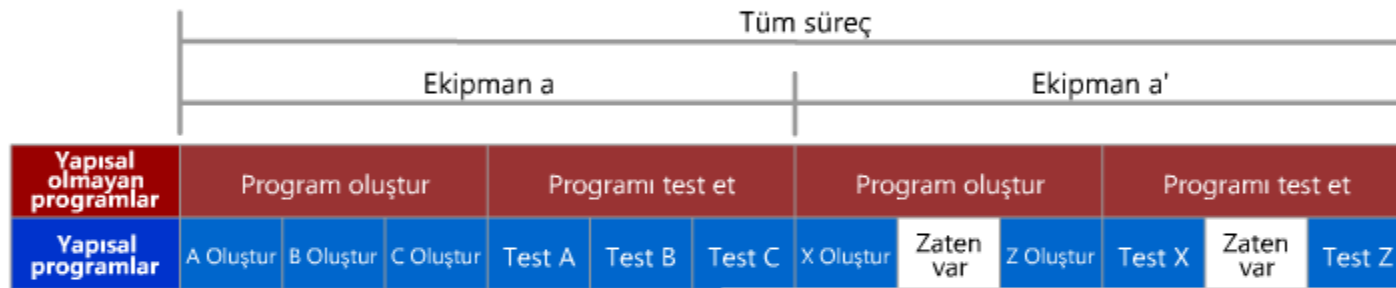
Program kullanarak programlama süresini kısaltma


Mevcut programların kullanılması, programlama süresini kısaltır ve programlama verimliliğini artırır.

Yapısal programlar kullanarak verimli programlama



Programlama süreci



Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

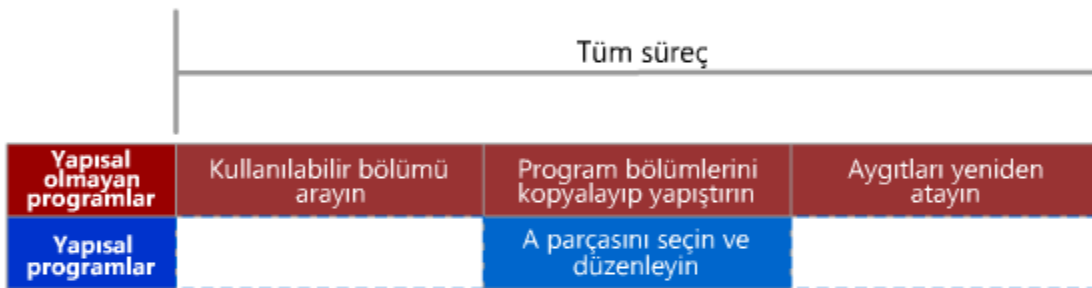
A Oluştur	B Oluştur	C Oluştur	Test A	Test B	Test C	X Oluştur	Z Oluştur	Test X	Test Z	Kaldırıldı
-----------	-----------	-----------	--------	--------	--------	-----------	-----------	--------	--------	------------


Yapısal programların kullanılması, yeni programda yeniden aygıt atama gerekliliğini ortadan kaldırır.

Yapısal programlar kullanarak verimli programlama



Programlama süreci



Animasyonu tekrar baştan başlatmak için sağ üst köşedeki  ögesini tıklayın.

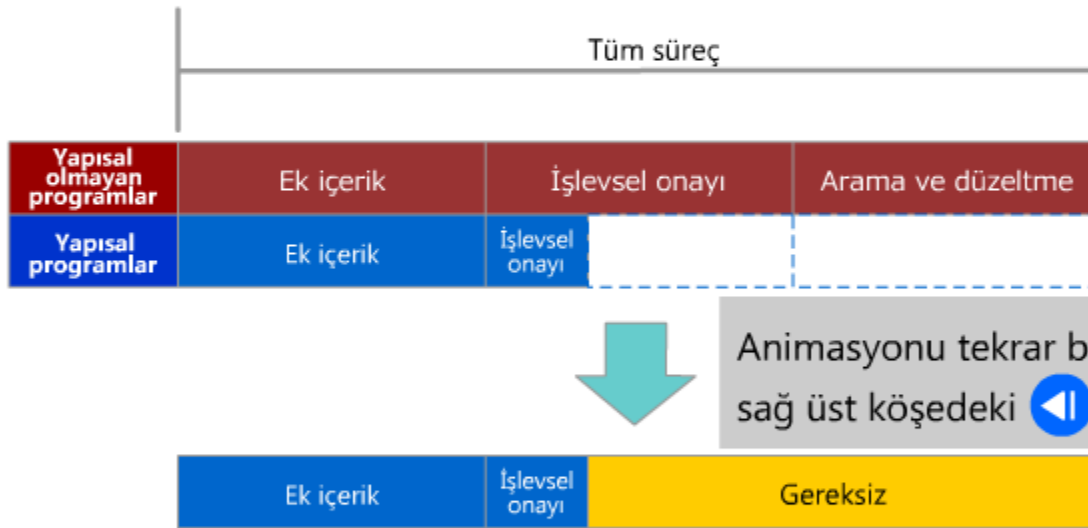


Programlara işlev eklemekten kaynaklanabilecek kazara yapılan deęişiklikler, yapısal program kullanılarak önlenabilir.

Yapısal programlar kullanarak verimli programlama



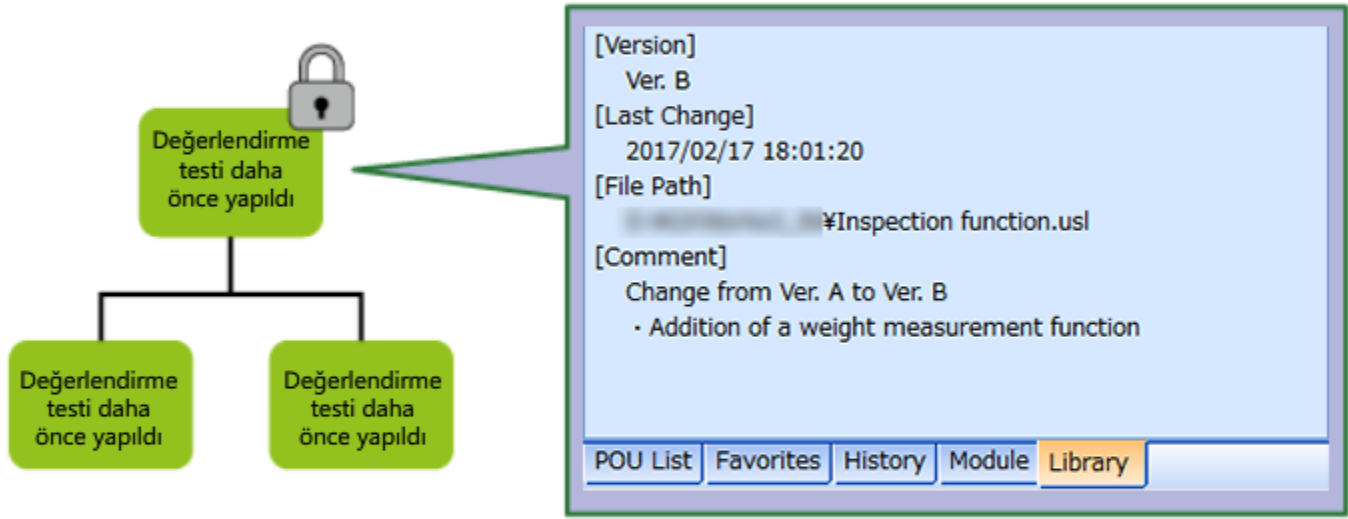
Programlama süreci



2.3.1 Güvenli POU yönetimi

POU'lar, MELSOFT GX Works3 kullanılarak güvenli bir şekilde yönetilebilir.

Kazara yapılan değişiklikler, tek tek POU'lar kilitlenerek tamamen ortadan kaldırılabilir. Her POU, sürüm, güncelleme tarihi ve yorumlar gibi kendi bilgilerine sahip olabilir. Değişiklik geçmişi yorumlara kaydedilebilir.



Bu bölümün içindekiler şunlardır:

- POU kullanımının avantajları
- Etiket kullanımının avantajları
- Yapısal program kullanımının avantajları

Dikkat edilecek önemli noktalar:

Daha az değerlendirme testi	Yapısal program kullanımı, yalnızca program oluşturmak için gereken süreyi azaltmaz, aynı zamanda değerlendirme testi için gereken süreyi de azaltır.
Kazara yapılan değişiklikleri önleme	Yapısal programlarla, değişiklikler tek tek POU'larda yapılır, böylece diğer program parçalarında kazara değişiklik yapılması önlenir.
POU kilitleme	Değerlendirme testi tamamlandıktan sonra POU'nun kilitlemesi, güvenli POU yönetimi sağlar.

Bu kursta, yapısal programlamanın, programlamanın daha verimli hale getirilmesi için kullanılması açıklanmıştır. Bu kursu tamamladınız.

Bu kursta ele alınanların özeti aşağıdadır.

- Yapısal programlama, büyük ölçekli sistemlerin üretim eğilimleri ve artan veri miktarıyla birlikte giderek karmaşıklaşan programların verimli bir şekilde oluşturulmasına yarayan bir yöntemdir
- Yapısal programların kullanımı kolaydır; bu da yeni geliştirme miktarını azaltır ve programlama verimliliğini artırır
- Programların program parçalarına bölünmesi, programların işleme ve işleve göre ayrılmasını sağlar. Program parçaları için hiyerarşik bir organizasyonun ve anlamlı adların kullanılması, kullanılacak program bölümlerini netleştirir
- Hiyerarşik organizasyon kullanımı, kullanılabilir bölümlerin daha verimli şekilde seçilmesini sağlar
- Etiket kullanımı, POU'ların yeni programla aygıt çatışmaları konusunda endişelenmeden kullanılabilmesini sağlar
- MELSOFT GX Works3, yapısal programlamayı destekler ve programlama verimliliğini artırır
- Kütüphane oluşturmak, sık kullanılan program parçalarının paylaşımını/standardizasyonunu ve programların kalitesinin tutarlı olmasını sağlar
- Program kullanımının yanı sıra, mevcut kütüphaneler de yeni geliştirme miktarını azaltmak için kullanılabilir

Yapısal programlama gerçekten uygulanırken, aşağıdakilere bakın.

MELSOFT GX Works3 ile programlama

- "GX Works3 (Ladder)" (GX Works3 (Ladder)) e-Learning kursu
- MELSOFT GX Works3 kullanım kılavuzları

Program yapısı ve söz dizimi

- MELSEC iQ-R Serisi programlama kılavuzları

Artık **PLC Verimli Programlama** 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 9 soru (10 madde) yer almaktadır.

Son testi istediğiniz sayıda uygulayabilirsiniz.

Testin puanlanması

Cevabı seçtikten sonra, **Cevapla** düğmesini tıkladığınızdan emin olun. Cevapla düğmesini tıklamadan ilerlemeniz durumunda 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: **5**

Toplam soru: **5**

Yüzde: **100%**

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

Devam Et

İncele

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

Test

Son Test 1



Yapısal programlar

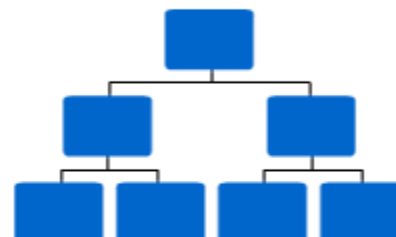
Yapısal programlar için doğru resmi seçin.

- A
- B
- C

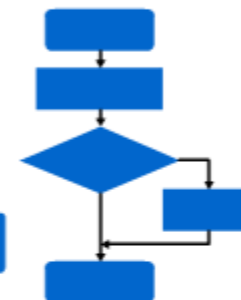
A



B



C



Cevapla

Geri

Etiket kullanımı

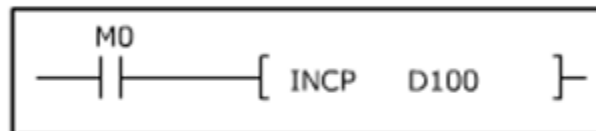
Etiket kullanarak oluşturulmuş programı seçin.

- A
- B
- C

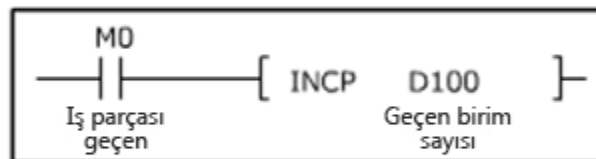
A



B



C



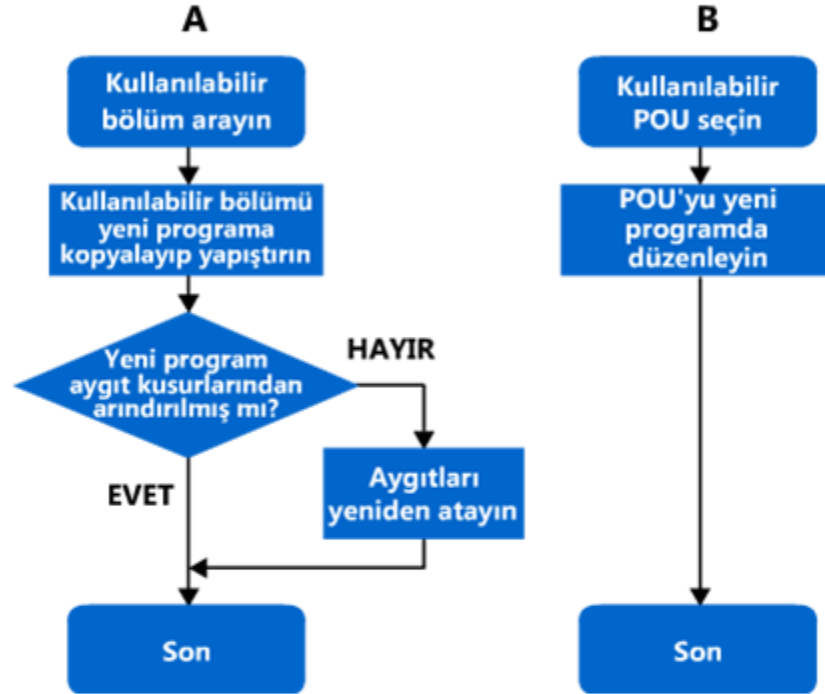
Cevapla

Geri

Program kullanımı süreci

Etiket kullanılarak oluşturulmuş yapısal programların kullanım sürecini doğru şekilde gösteren akış grafiğini seçin.

- A
- B



Cevapla

Geri

Yapısal programların amacı

Yapısal programların amacını anlatan doğru açıklamayı seçin. (Birden çok cevap)

- Program işleme hızını artırmak
- Program verimliliğini artırmak
- Program derleme süresini azaltmak
- Programların görselleştirilmesini iyileştirmek

Cevapla

Geri

GX Works3'te yapısal programlama

GX Works3'te yapısal program oluşturma ile ilgili doğru açıklamayı seçin.

- Yapısal programlar, yazılım kurulduktan sonra hemen oluşturulabilir.
- GX Works3'te yapısal program oluşturmak için eklenti satın almak gerekir.

Cevapla

Geri

Programların program parçalarına bölünmesi

Programların, program parçalarına bölünmesini açıklayan tüm ifadeleri seçin. (Birden çok cevap)

- Programları önceden belirlenmiş, eşit adım sayısına bölün
- Programları bireysel işlem ve işlevlere bölün
- Süreçlere anlamlı adlar atayın
- Farklı bölümlere atlamak için işaretçi kullanın
- Program organizasyon birimleri (POU), programların program parçalarına bölünmesinin sonucudur

Cevapla

Geri

Kütüphanelerin avantajları

Programların kütüphanelere kaydedilmesinin avantajlarını anlatan doğru açıklamayı seçin. (Birden çok cevap)

- Sık kullanılan POU'ların kaydedilmesi, programların verimli kullanılmasını sağlar
- POU kullanımını engeller
- POU'lar birçok kişi arasında paylaşılabilir
- Standartlaştırılmış programların kaydedilmesi ve tekrar kullanılması, tutarlı program kalitesi sağlar

Cevapla

Geri

Kütüphane, Modül FB'leri ve MELSOFT Library Kullanımı

Modül işlev bloğu ve MELSOFT Library gibi kütüphanelerin kullanımı ile ilgili doğru açıklamayı seçin.

- Önceden hazırlanmış kütüphaneler kullanıldığında, dahili operasyonun doğrulanması gerekmez
- Modül FB'leri, modül modeline uygun şekilde oluşturulmalıdır

Cevapla

Geri

Test**Son Test 9**

Önceden Yapılandırılmış Kütüphaneler

Mitsubishi Electric tarafından sağlanan kütüphaneyi açıklayan seçeneği seçin.

S1 Modül FB'leri

S2 MELSOFT kütüphanesi

S1

S2

Cevapla

Geri

Test**Test Puanı**

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

Doğru cevaplar: **9**

Toplam soru: **9**

Yüzde: **100%**

[Devam](#)[İncele](#)

Tebrikler. Testi geçtiniz.

PLC Verimli Programlama 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