

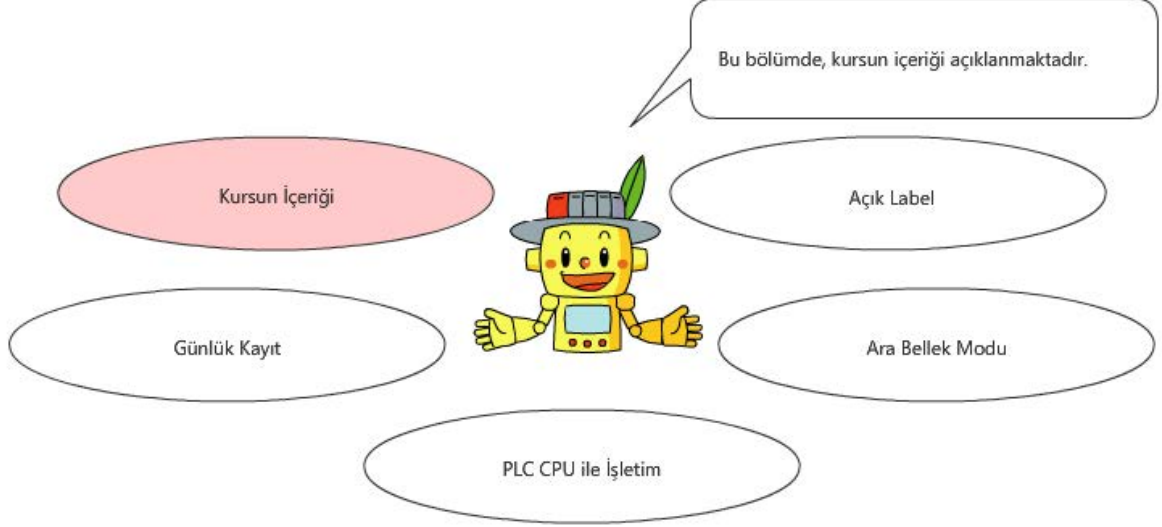
Servo Sistemi Kontrolörü

MELSEC iQ-R Series Motion Modülü Temel Bilgileri (RD78G(H) Pozisyonlama Kontrolü)

Bu eğitim kursu, MELSEC iQ-R Serisi Motion modülünü ilk kez kullanarak bir motion kontrol sistemi kuracak kullanıcılara yöneliktir.

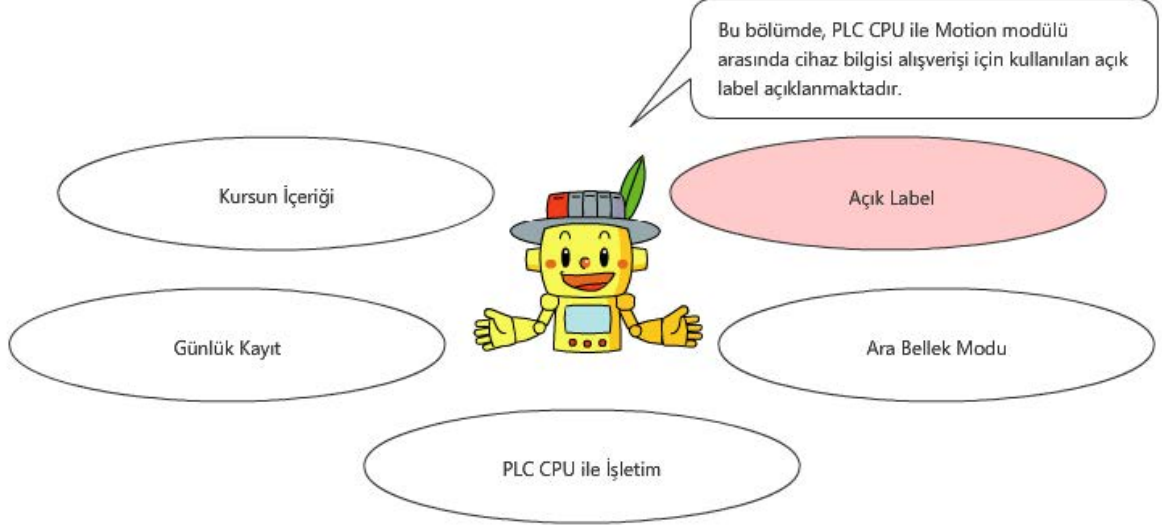
Sonraki sayfaya ilerlemek için sağ üst köşedeki İleri butonunu tıklayın.

Bu kursun amacı, MELSEC iQ-R Serisi Motion modülü kullanılarak Motion kontrol sisteminin pozisyonlama kontrolü hakkında bilgi edinilmesini ve anlaşılmasını sağlamaktır.



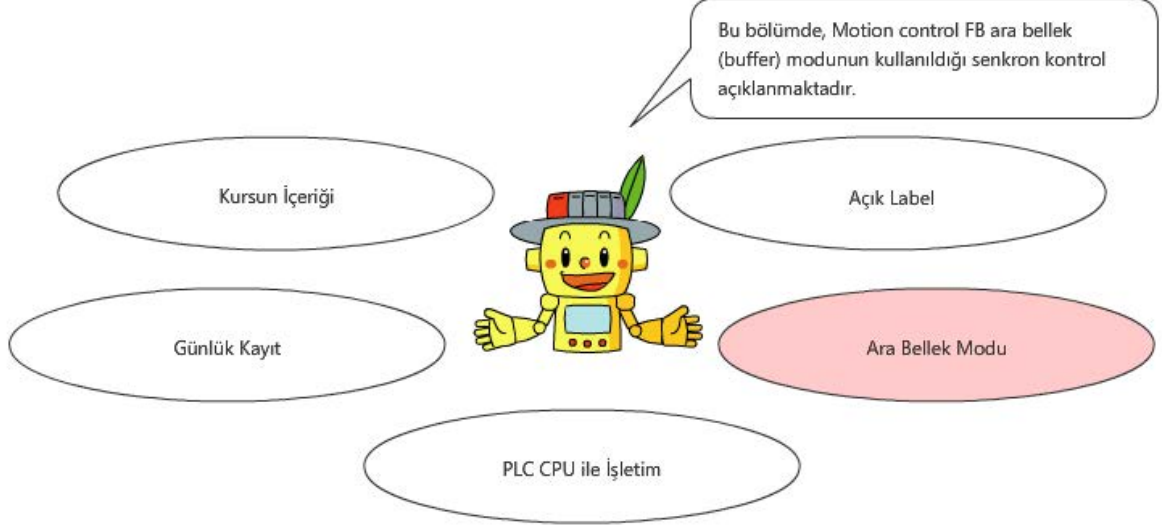
Bu kurs MELSEC iQ-R Serisi Motion Modülü Temel Bilgileri (RD78G(H) Başlangıç) eğitiminin devamı niteliğindedir. Bu kursu almadan önce Başlangıç kursunu bitirdiğinizden emin olun.

Bu kursun amacı, MELSEC iQ-R Serisi Motion modülü kullanılarak Motion kontrol sisteminin pozisyonlama kontrolü hakkında bilgi edinilmesini ve anlaşılmasını sağlamaktır.



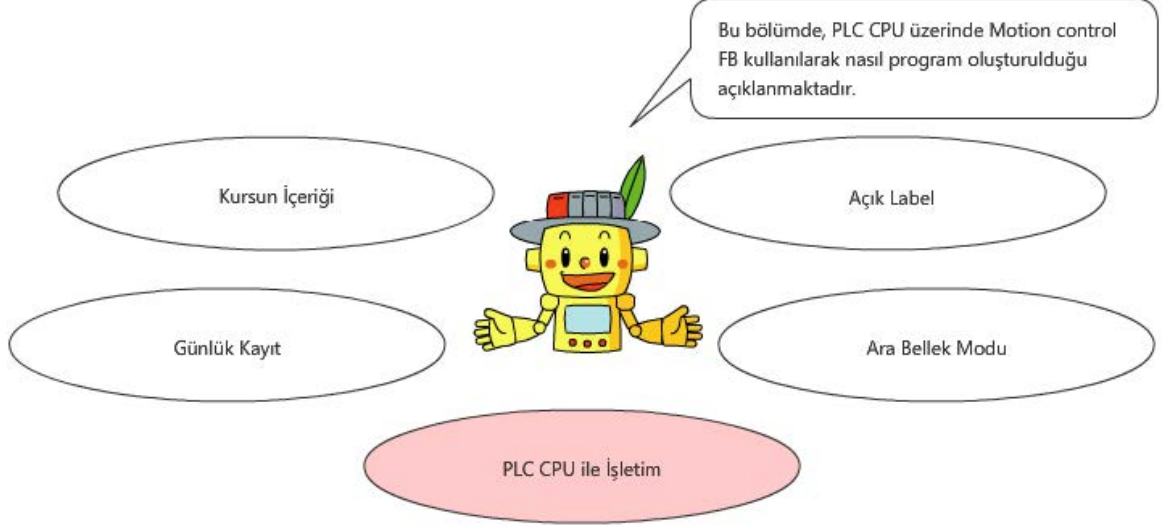
Bu kurs MELSEC iQ-R Serisi Motion Modülü Temel Bilgileri (RD78G(H) Başlangıç) eğitiminin devamı niteliğindedir. Bu kursu almadan önce Başlangıç kursunu bitirdiğinizden emin olun.

Bu kursun amacı, MELSEC iQ-R Serisi Motion modülü kullanılarak Motion kontrol sisteminin pozisyonlama kontrolü hakkında bilgi edinilmesini ve anlaşılmasını sağlamaktır.



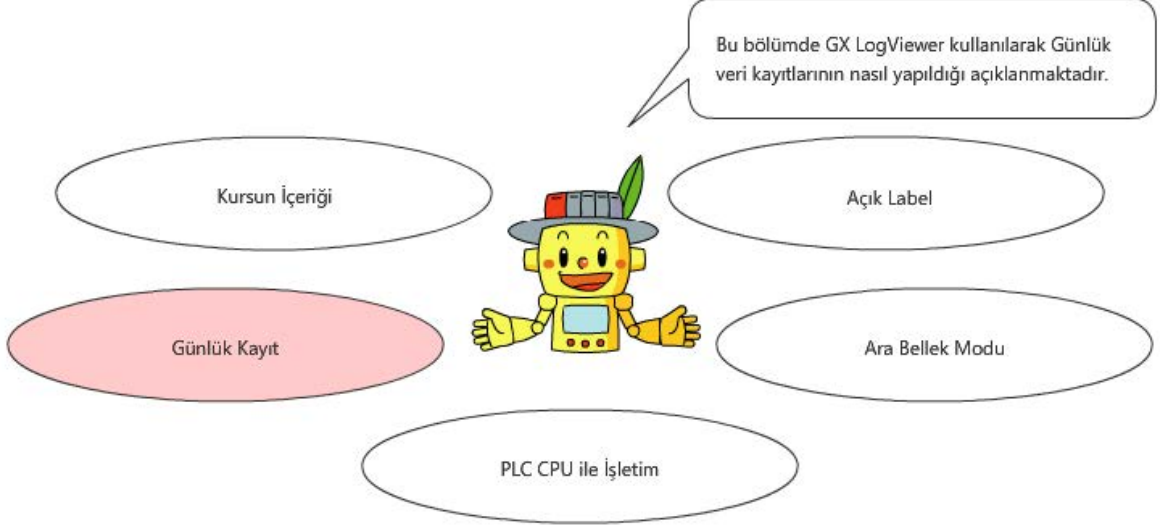
Bu kurs MELSEC iQ-R Serisi Motion Modülü Temel Bilgileri (RD78G(H) Başlangıç) eğitiminin devamı niteliğindedir. Bu kursu almadan önce Başlangıç kursunu bitirdiğinizden emin olun.

Bu kursun amacı, MELSEC iQ-R Serisi Motion modülü kullanılarak Motion kontrol sisteminin pozisyonlama kontrolü hakkında bilgi edinilmesini ve anlaşılmasını sağlamaktır.



Bu kurs MELSEC iQ-R Serisi Motion Modülü Temel Bilgileri (RD78G(H) Başlangıç) eğitiminin devamı niteliğindedir. Bu kursu almadan önce Başlangıç kursunu bitirdiğinizden emin olun.

Bu kursun amacı, MELSEC iQ-R Serisi Motion modülü kullanılarak Motion kontrol sisteminin pozisyonlama kontrolü hakkında bilgi edinilmesini ve anlaşılmasını sağlamaktır.



Bu kurs MELSEC iQ-R Serisi Motion Modülü Temel Bilgileri (RD78G(H) Başlangıç) eğitiminin devamı niteliğindedir. Bu kursu almadan önce Başlangıç kursunu bitirdiğinizden emin olun.

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

Bölüm 1 - Kursun İçeriği

Bu bölümde, kursun içeriği açıklanmaktadır.

Bölüm 2 - Açık Label

Bu bölümde, PLC CPU ile Motion modülü arasında cihaz bilgisi alışverişi için kullanılan açık Label açıklanmaktadır.

Bölüm 3 - Buffer Modu

Bu bölümde, Motion control FB buffer modunun kullanıldığı senkron kontrol açıklanmaktadır.

Bölüm 4 - PLC CPU ile İşletim





Bu bölümde, PLC CPU üzerinde Motion control FB kullanılarak nasıl program oluşturulduğu açıklanmaktadır.

Bölüm 5 - Günlük Kaydı

Bu bölümde GX LogViewer kullanılarak günlük veri kayıtlarının nasıl yapıldığı açıklanmaktadır.

Son Test

Toplam 4 kısım (7 soru)

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

■Güvenlik Önlemleri

Eğitim amacıyla gerçek ürünler kullanırken, kullanılacak ürünün kılavuzunda açıklanan "Güvenlik Önlemlerini" dikkatlice okuyunuz ve güvenlik ve uygun kullanım hususlarına çok dikkat ediniz.

■Bu Kurstaki Önlemler

Kursta gösterilen görüntüler, sürüme bağlı olarak gerçek yazılımınızdan farklı olabilir. Kursta aşağıdaki yazılım sürümleri kullanılmaktadır.

Her yazılımın en son sürümü için, Mitsubishi Electric FA web sitesini kontrol ediniz.

MELSOFT GX Works3	Ver.1.066U	Motion Control Setting function	Ver.1.012N
GX LogViewer	Ver.1.106K		
MELSOFT MR Configurator2	Ver.1.110Q veya daha yeni		

PLC CPU donanım yazılımı sürümü 44 veya daha yeni olmalıdır (RD78GH için 46 veya daha yeni).

Motion modülü donanım yazılımı sürümü 10 veya daha yeni olmalıdır.

Donanım yazılımı sürümünün nasıl güncelleneceği hakkında bilgi için, MITSUBISHI ELECTRIC FA web sitesine veya modülün konfigürasyon kılavuzuna başvurunuz.



simgesi başvuru kılavuzunu gösterir.

Bu kursta açıklanan kılavuzların içeriği aşağıdaki sürümlerin içeriğini yansıtır.

Sürümler farklılık gösterirse, içerik ve açıklama bölümleri biraz farklı olabilir.

Kılavuz adı	Kılavuz No.	Sürüm
MELSEC iQ-R Motion Module User's Manual (Startup)	IB-0300406	E
MELSEC iQ-R Motion Module User's Manual (Application)	IB-0300411	E
MELSEC iQ-R Motion Module User's Manual (Network)	IB-0300426	E
MELSEC iQ-R Programming Manual (Motion Module Instructions, Standard Functions/Function Blocks)	IB-0300431	E
MELSEC iQ-R Programming Manual (Motion Control Function Blocks)	IB-0300533	C
MELSEC iQ-R Structured Text (ST) Programming Guide Book	SH-081483	F
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)	SH-081266	Z
MELSEC iQ-R CPU Module User's Manual (Application)	SH-081264	AK

Aşağıda, kursun genel yapısı gösterilmektedir.

Bölüm 1 Kursun İçeriği

Bu bölümde, kursun içeriği açıklanmaktadır.



Bölüm 2 Açık Label

Bu bölümde, PLC CPU ile Motion modülü arasında cihaz bilgisi alışverişi için kullanılan açık label açıklanmaktadır.



Bölüm 3 Buffer Modu

Bu bölümde, Motion control FB buffer modunun kullanıldığı senkron kontrol açıklanmaktadır.



Bölüm 4 PLC CPU ile İşletim

Bu bölümde, PLC CPU üzerinde Motion control FB kullanılarak nasıl program oluşturulduğu açıklanmaktadır.



Bölüm 5 Günlük Kayıt

Bu bölümde, Motion modülünün işletimini kontrol etmek üzere GX LogViewer kullanılarak Günlük veri kayıtlarının nasıl yapıldığı açıklanmaktadır.

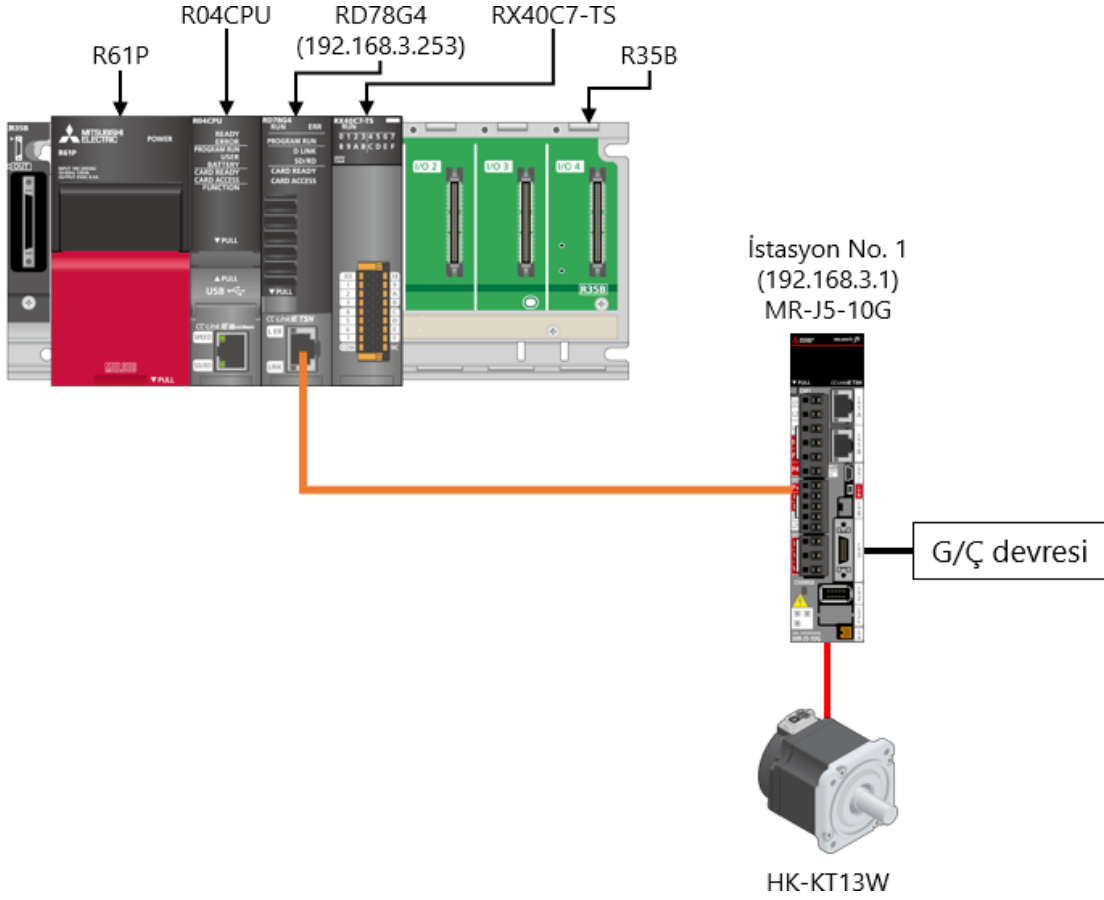
Bu kursta, Başlangıç kursunda kullanılanla aynı tek şaftlı bilyeli vida mekanizması kullanılmaktadır.



Hedef sistemin konfigürasyonu aşağıdaki gibidir.

Başlangıç kursunda kullanılan sistemdeki uzak giriş modülünü çıkarın ve RX40C7-TS giriş modülünü, programlanabilir kontrolörün ana ünitesindeki 1. yuvaya ekleyin.

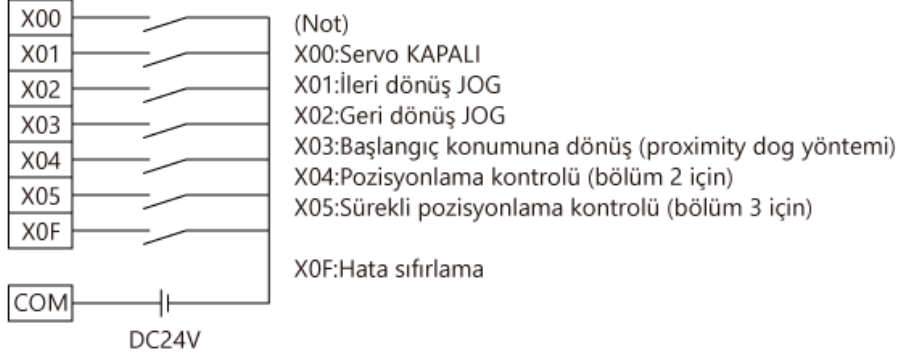
MR-J5-10G servo sürücünün istasyon numarası 1 olarak değiştirilmiş olup IP adresi de 192.168.3.1 olarak değiştirilmiştir.



Programlanabilir kontrolör ve servo sürücü için güç kaynağının kablo tesisatı ve servo sürücünün bağlantı yöntemi, Başlangıç kursunda açıklananla aynıdır.

Aşağıda, giriş modülünün harici devre kablo tesisatı gösterilmektedir.

RX40C7-TS



(Not) RX40C7-TS için I/O No. 0020H olduğundan, programda X20 ila X25 ve X2F kullanılmaktadır.

Bu kursun 1. bölümünde kullanılan sistemde açıklandığı üzere, Motion modülü programlanabilir kontrolörün giriş modülüyle kontrol edildiğinde, PLC CPU ve Motion modülünün cihaz bilgilerini alıp vermesi gerekir.

Aşağıdaki iki yöntem mevcuttur.

1. Açık labelların kullanımı.
2. Motion modülünün ara belleğinin (buffer memory) kullanımı.

Bu bölümde açık labellar kullanılarak veri alışverişinin nasıl yapıldığı açıklanmaktadır.

Bu bölümde ve bölüm 3'te kullanılacak örnek programı, aşağıdaki bağlantıyı tıklayarak indirin.

[RD78GBasic2_sample1.zip \(1.34MB\)](#)

[Nokta]

Buffer memory kullanırken, alınıp verilecek veriyi kullanıcı alanına kopyalayın (U0\G11478000 to G11997999).

(Program örneği)

<PLC CPU>

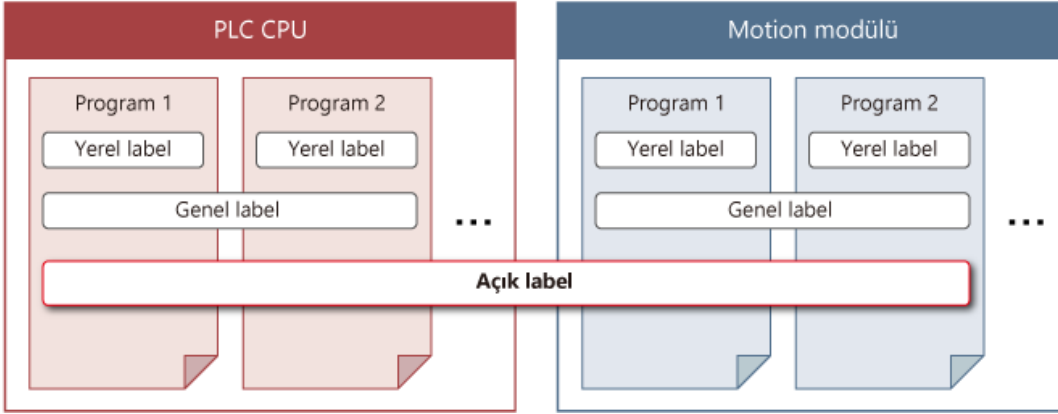
Pozisyonlama başlatma

—||—(U0\G11478000.0)

<Motion modülü>

```
MC_MoveAbsolute_1(  
  Execute:= G11478000.0 ,  
  :  
  :  
);
```

Açık label, hem Motion modülü hem de PLC CPU'da kullanılabilen ortak labeldir. Aşağıda, yerel label, genel label ve açık labelın uygulandığı alanlar gösterilmektedir.



(1) Açık labellar nasıl kaydedilir

Açık labelları, Motion modülünün genel labellarından kaydedin.

Motion Module Setting Function ekranının genel label düzenleyicisinde "Public Label" sütununun görünür olduğundan emin olun.

Genel label olarak kaydedilecek labelları "Enabled" şeklinde ayarlayın.

Bu işlem "Motion Control Attribute" sütununu etkinleştirir.

Her bir labelın PLC CPU'dan okunacağını mı, yoksa PLC CPU'ya yazılacağını mı seçin.

	Label Name	Data Type	Class	Initial	Constant	Japanese	English(Display Target)	Chinese	Remark	Public Label	Motion Control Attribute
1	G_bSVONCMD	Bit	VAR_GLOBAL				Servo ON			Enabled	WRITE (=> Motion)
2	G_leJogVelocity	FLOAT [Double Precision]	VAR_GLOBAL				JOG Velocity			Enabled	WRITE (=> Motion)
3	G_bJogFwd	Bit	VAR_GLOBAL				JOG Forward			Enabled	WRITE (=> Motion)
4	G_bJogBwd	Bit	VAR_GLOBAL				JOG Backward			Enabled	WRITE (=> Motion)
5	G_bJogBusy	Bit	VAR_GLOBAL				JOG Busy			Enabled	READ (Motion =>)
6	G_lePosition0	FLOAT [Double Precision]	VAR_GLOBAL				Position0 Address			Disabled	-
7	G_bHomingCMD	Bit	VAR_GLOBAL				Homing Command			Enabled	WRITE (=> Motion)
8	G_bHomingDone	Bit	VAR_GLOBAL				Homing Done			Enabled	READ (Motion =>)
9	G_bHomingReq	Bit	VAR_GLOBAL				Homing Request			Enabled	READ (Motion =>)
10	G_bPosCMD	Bit	VAR_GLOBAL				Positioning Command			Enabled	WRITE (=> Motion)
11	G_bPosDone	Bit	VAR_GLOBAL				Positioning Done			Enabled	READ (Motion =>)
12	G_bPosReq	Bit	VAR_GLOBAL				Positioning Start Request			Enabled	READ (Motion =>)
13	G_bErrorReset	Bit	VAR_GLOBAL				Error Reset			Enabled	WRITE (=> Motion)
14	G_bContPosCMD	Bit	VAR_GLOBAL				Continuous Positioning Command			Enabled	WRITE (=> Motion)
15	G_bContPosReq	Bit	VAR_GLOBAL				Continuous Positioning Start Request			Enabled	WRITE (=> Motion)
16	G_bContPosDone	Bit	VAR_GLOBAL				Continuous Positioning Done			Enabled	READ (Motion =>)
17											

[Nokta]

Açık labelın sütunu görünür değilse, tabloyu sağa kaydırın.

(2) Açık label olarak kaydedilebilen veri türleri
Aşağıdaki tabloda, açık label olarak kaydedilebilen veri türleri gösterilmektedir.

Değişken türü	Tür	Dizilim seçimi	Açık label ayarı	Yorumlar
Genel label	Basit tür	Hayır	○	Aşağıdaki labellar ve sınıf için ayarlar mümkün değildir. ■ Label <ul style="list-style-type: none"> • Dizi tipi label • Zamanlayıcı tipi label • Sayaç tipi label • Uzun sayaç tipi label • Güçlü zamanlayıcı labelı • Uzun güçlü zaman tipi label • Uzun zamanlayıcı tipi label
		Evet	△(Not 1,2)	
	Yapılandırılmış (Structured) veri türü	Hayır	△(Not 3)	
		Evet	△(Not 1,2,4,5)	
	FB (Motion control FB'ler dahil)	Hayır	×	
		Evet	×	
Program	-	-	×	
Program bloku yerel labelı	-	-	×	
Yapılandırılmış (Structured) veri türü	-	-	△(Not 3,5)	■ Sınıf <ul style="list-style-type: none"> • VAR_GLOBAL_CONSTANT sınıfı
Motion control FB Yapılandırılmış (Structured) verisi	-	-	△(Not 6,7)	

(Not)

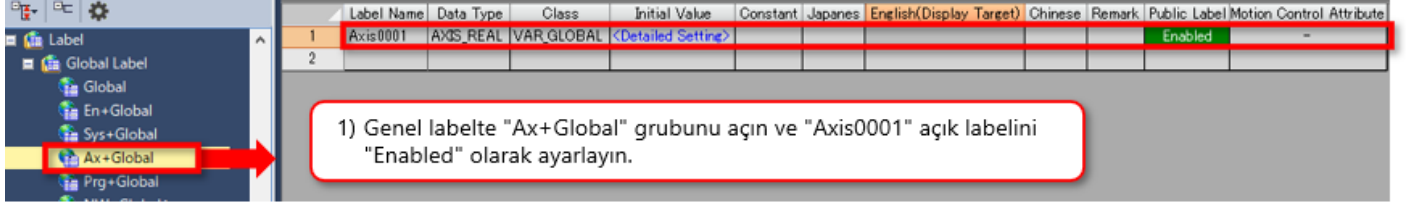
1. Açık label ayarı, bir dizilimin her ögesi için konfigüre edilemez.
2. Bit tipi dizilim kullanıldığında, açık label "Enabled" olarak ayarlanamaz. (Yapılandırılmış (Structured) bir veride, yalnızca ilgili üye "Enabled" olarak ayarlanamaz.)
3. Yapılandırılmış (Structured) veri türü olarak dizi türü kullanıldığında, üye "Enabled" olarak ayarlanamaz.
4. En fazla dört katmanı olan yapılandırılmış (Structured) veri açık hale getirilebilir.
5. Yapılandırılmış (Structured) veri dizilimi, yapılandırılmış (Structured) verinin üyesi olarak kullanıldığında, üye "Enabled" olarak ayarlanamaz.
6. CPU modülü tarafından PLCopen Motion control FB programında kullanılabilir.
7. Bir Motion control FB yapılandırılmış (Structured) verisinde dizi türü kullanıldığında, Motion control FB yapılandırılmış (Structured) veri türü kendi başına ayarlanamaz.

(3) Yapılandırılmış (Structured) veri nasıl açık label olarak kaydedilir

Sistemde hazırlanan yapılandırılmış (Structured) veri türünün üyelerini (eksen monitör verileri gibi) açık label olarak ayarlamak için, açık labeleri aşağıda gösterildiği gibi yapılandırılmış (Structured) verinin katmanına göre kaydedin.

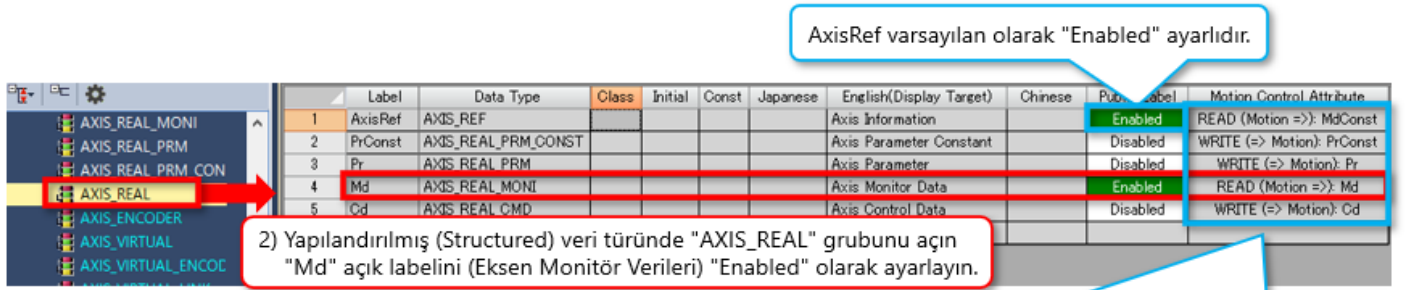
Bu kursta, gerçek sürücü ekseninin (Axis_Real) Pozisyon Ayarlama (SetPosition) ve Hız Ayarlama (SetVelocity) monitör verilerinin (Md) nasıl açık labeler olarak kaydedildiği açıklanmaktadır.

[AxisName.Md.SetPosition (Komut mevcut pozisyonu) ve AxisName.Md.SetVelocity (Komut mevcut hızı) nasıl açık labeler ayarlanır]



Label Name	Data Type	Class	Initial Value	Constant	Japanese	English(Display Target)	Chinese	Remark	Public Label	Motion Control Attribute
1	Axis0001	AXIS_REAL	VAR_GLOBAL	<Detailed Setting>					Enabled	-
2										

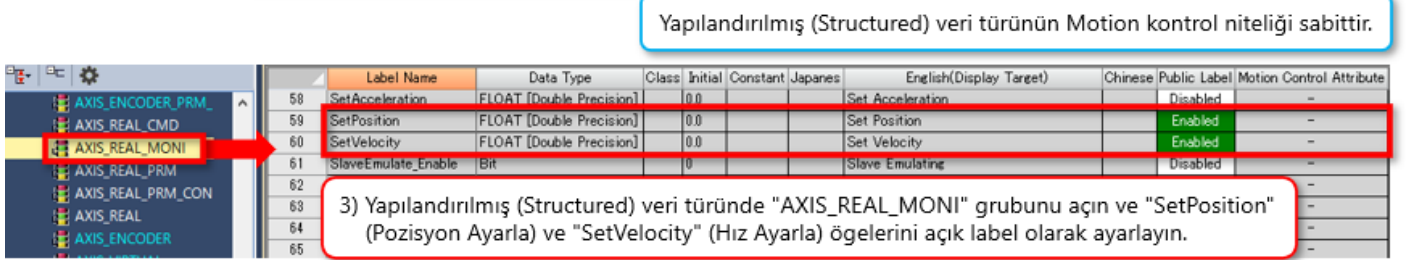
1) Genel labelte "Ax+Global" grubunu açın ve "Axis0001" açık labelini "Enabled" olarak ayarlayın.



Label	Data Type	Class	Initial	Const	Japanese	English(Display Target)	Chinese	Pub. Label	Motion Control Attribute
1	AxisRef	AXIS_REF				Axis Information		Enabled	READ (Motion =>): MdConst
2	PrConst	AXIS_REAL_PRM_CONST				Axis Parameter Constant		Disabled	WRITE (=> Motion): PrConst
3	Pr	AXIS_REAL_PRM				Axis Parameter		Disabled	WRITE (=> Motion): Pr
4	Md	AXIS_REAL_MONI				Axis Monitor Data		Enabled	READ (Motion =>): Md
5	Cd	AXIS_REAL_CMD				Axis Control Data		Disabled	WRITE (=> Motion): Cd

AxisRef varsayılan olarak "Enabled" ayarlıdır.

2) Yapılandırılmış (Structured) veri türünde "AXIS_REAL" grubunu açın "Md" açık labelini (Eksen Monitör Verileri) "Enabled" olarak ayarlayın.



Label Name	Data Type	Class	Initial	Constant	Japanese	English(Display Target)	Chinese	Public Label	Motion Control Attribute
58	SetAcceleration	FLOAT [Double Precision]	0.0			Set Acceleration		Disabled	-
59	SetPosition	FLOAT [Double Precision]	0.0			Set Position		Enabled	-
60	SetVelocity	FLOAT [Double Precision]	0.0			Set Velocity		Enabled	-
61	SlaveEmulate_Enable	Bit	0			Slave Emulating		Disabled	-
62									
63									
64									
65									

Yapılandırılmış (Structured) veri türünün Motion kontrol niteliği sabittir.

3) Yapılandırılmış (Structured) veri türünde "AXIS_REAL_MONI" grubunu açın ve "SetPosition" (Pozisyon Ayarla) ve "SetVelocity" (Hız Ayarla) öğelerini açık label olarak ayarlayın.

(4) Açık labelleri yansıtma

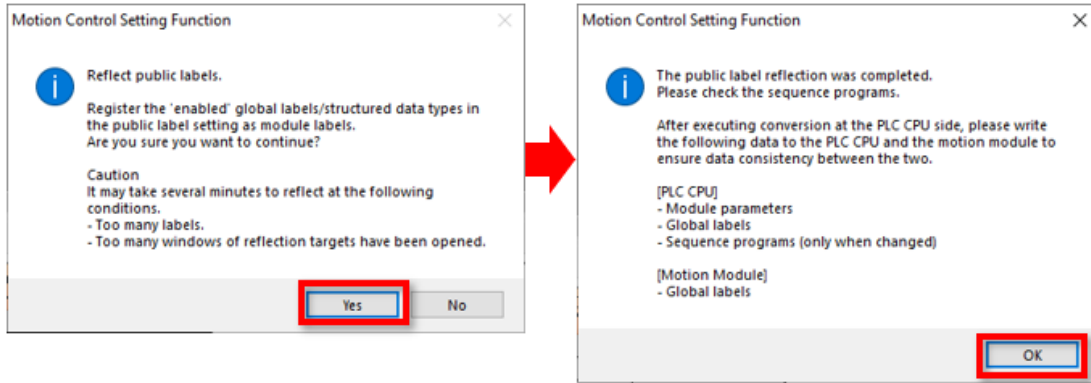
Menüde [Convert] → [Rebuild All] seçin.

Açık labelin serbest kapasitesi, çıkış penceresinde Bilgi olarak görüntülenir.

No.	Result	Data Name	Category	Content	Error Code
1	Information	Public Label	Free Volume	99.88[%] (32728 [Word] = 32768 [Word] - (Global: 40 [Word]))	-

Tüm sürecin yeniden oluşturulması başarıyla tamamlandığında, menüde [Convert] → [Reflect Public Labels] seçimlerini yapınız. Aşağıdaki açılır pencerede [Yes] düğmesini tıklayınız.

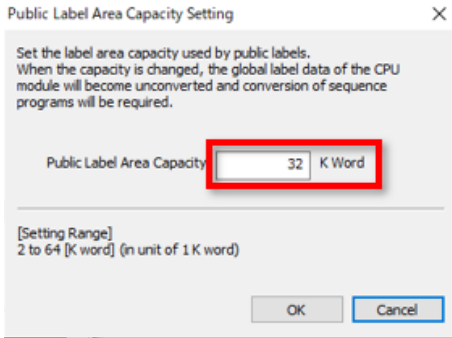
Açık labellerin başarıyla yansıtıldığını bildiren bir mesaj açıldığında, [OK] düğmesini tıklayınız.



(Not) Açık labelleri kaydetmek için kullanılabilen bellek kapasitesi varsayılan olarak 32K sözcüktür.

Kapasite 64K sözcüğe çıkarılabilir.

Kapasiteyi değiştirmek için, bellek boyutunu menüdeki [Convert] → [Public Label Capacity Setting] öğelerini kullanarak ayarlayınız.



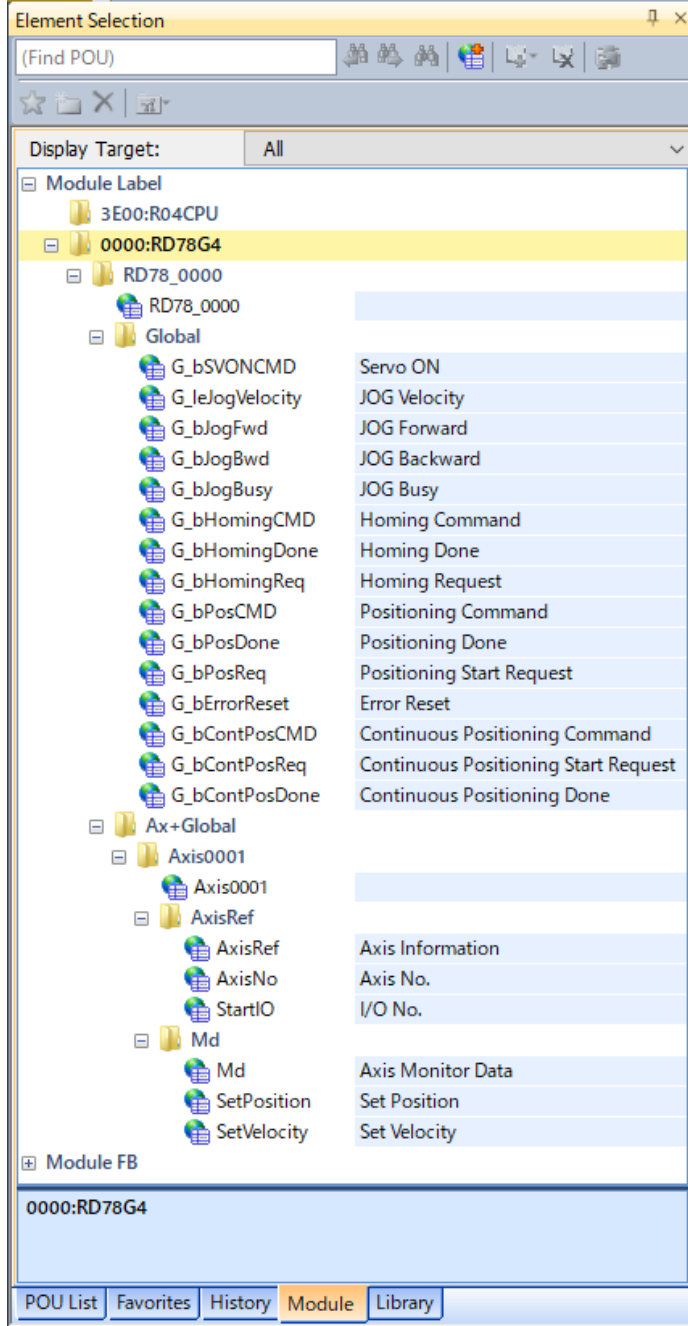
(5) PLC CPU tarafındaki labellerin kontrolü

Yansıtılan açık labeller PLC CPU tarafındaki modül labeline kaydedilir.

Modül labelini GX Works3 Element Selection penceresinden seçin ve açık labellerin [Module Label] içinde [0000:RD78G4] altına kaydedildiğini kontrol edin.

Açık label ayarını değiştirdikten sonra, mutlaka "Reflect Public Labels" işlemini tekrar yürütün.

PLC CPU'da açık labellar kullanırken, tüm programları yeniden oluşturun.

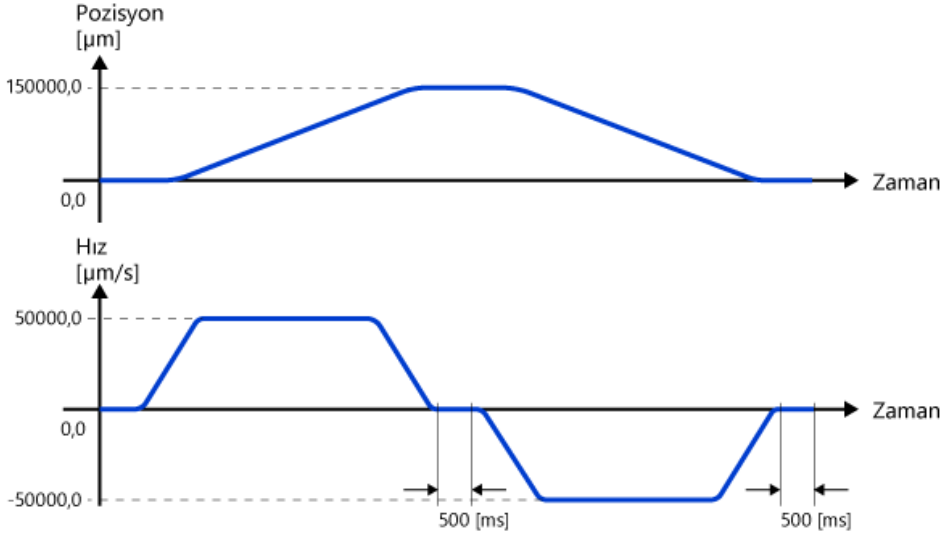


(1) Örnek program işletimi

Bu bölümde kullanılan örnek programın giriş sinyalleri aşağıdaki şekilde atanır.

Giriş	İşletim
X20	Servo kapalı (Not)
X21	İleri dönüş JOG işletimi
X22	Geri dönüş JOG işletimi
X23	Başlangıç konumuna dönüş
X24	Pozisyonlama kontrolü
X25	Sürekli pozisyonlama kontrolü (Bölüm 3)

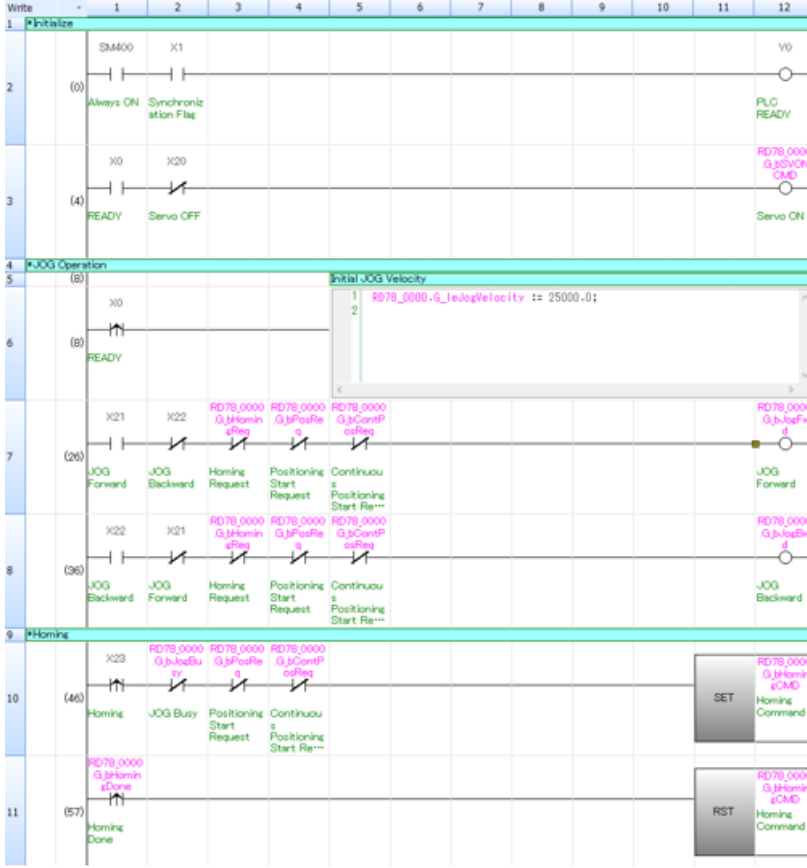
Aşağıda X24'ün çalışma düzeni gösterilmektedir: pozisyonlama kontrolü.



(Not) PLC CPU RUN olarak ayarlandığında bu örnek program servo AÇIK komutunu otomatik olarak yürütür. Başlama sinyali AÇIK haldeyken güç açıldığında, servo motor devreye girebilir.

(2) PLC CPU'nun programı

1) MAIN (ladder, tarama programı)



Y0 ilk olarak açılır.

X0 açıldığında, servo AÇIK yürütülür. Servo KAPALI komutunu yürütmek için X20'yi açın.

JOG hızının başlangıç değerini ayarlayın. Bu programda, inline ST kullanılmıştır. JOG hızını saklayan "G_JogVelocity" genel labelinin motion kontrol niteliği "WRITE (→Motion)" olarak ayarlandığı için, PLC CPU'da sayısal değerin ayarlanması gerekir.

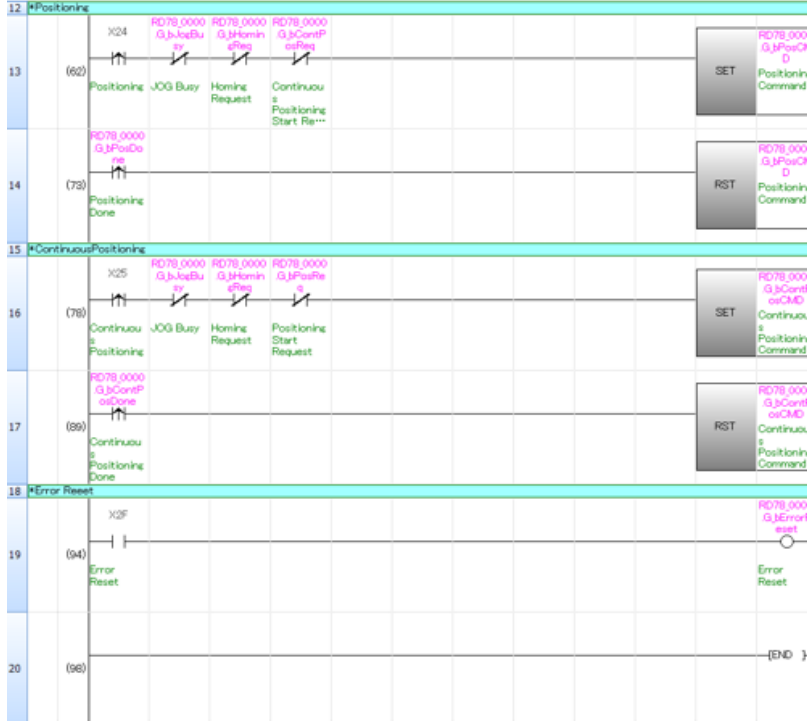
JOG işletiminin başlama sinyalini açar. İleri dönüş ve geri dönüşün aynı anda başlatılmasını engeller. Başka bir program yürütülürken JOG işletiminin başlatılmasını engellemek için bir ara kilit ayarlanır.

Başlangıç konumuna dönüşün başlangıcı (X23), G_bHomingCMD açık labelinde tutulur ve Motion modülüne, başlangıç konumuna dönüşün başlama koşulu olarak gönderilir. Başka bir program çalışırken başlangıç konumuna dönüşün başlatılmasını engellemek için bir ara kilit ayarlanır. Motion modülünün başlangıç konumuna dönüş tamamlama sinyalini açtığı alındığında, bu sinyalin yükselme eşliğinde G_bHomingCMD sıfırlanır.

(2) PLC CPU'nun programı

1) MAIN devam kısmı (ladder, tarama programı)

(Önceki sayfadan devam)



Pozisyonlama kontrolü başlatmanın yükselme eşiği (X24), G_bPosCMD'de tutulur ve pozisyonlama kontrolünün başlama koşulu olarak Motion modülüne gönderilir.

Başka bir program çalışırken pozisyonlama kontrolünün başlatılmasını engellemek için bir ara kilit ayarlanır.

Motion modülünün başlangıç konumuna dönüş tamamlama sinyalini açtığı alındığında, bu sinyalin yükselme eşiğinde G_bPosCMD sıfırlanır.

Sürekli pozisyonlama için başlatma programı bölüm 3'te açıklanmaktadır. Ayrıntılar için, 3.4'e başvurunuz.

X2F açıldığında hatalar sıfırlanır.

(2) PLC CPU'nun programı

2) MONITOR (ST, tarama programı)

Açık label olarak ayarlanan, eksen monitörüne ait SetPosition (Pozisyon Ayarla) ve SetVelocity (Hız Ayarla), D0 ve D2 word device'larda açık label olarak saklanır.

SetPosition ve SetVelocity çift hassasiyetli reel sayı tipinde olduğundan, PLC CPU tarafından kolayca işlenebilmeleri için imzalı çift word tipine dönüştürülürler. (Not)

Bu word device'lar bu kursta kullanılmıyor olsa da, diğer sekans (Sequence) programlarında ve GOT üzerinde veri görüntülemek amacıyla ve diğer amaçlarla kullanılırlar.

```
1 D0:D := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetPosition);  
2 D2:D := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetVelocity);  
3
```

"D0:D" ile imzalı çift word tipini belirtin.

(Not) Çift hassasiyetli reel sayı tipi, imzalı çift word tipine dönüştürüldüğünde, dönüştürülecek değer -2147483648 ila 2147483647 aralığının dışındaysa, bir hesaplama hatası oluşur.

(3) Motion modülünün programı

1) ServoON_JOG (normal yürütme tipi)

```

1 //-----Servo ON-----
2 MC_Power_1(
3     Axis      := Axis0001.AxisRef,
4     Enable    := TRUE,
5     ServoON   := G_bSVONCMD,
6     Busy      => bPowerBusy
7 );
8
9 //-----JOG Operation-----
10 //Initial Value Setting
11 IF (bPowerBusy) THEN
12     leJogAcceleration := 50000.0;
13     leJogDeceleration := 50000.0;
14     leJogJerk          := 0.0;
15 END_IF;
16
17 //JOG
18 MCv_Jog_1(
19     Axis      := Axis0001.AxisRef,
20     JogForward := G_bJogFwd,
21     JogBackward := G_bJogBwd,
22     Velocity   := G_leJogVelocity,
23     Acceleration := leJogAcceleration,
24     Deceleration := leJogDeceleration,
25     Jerk       := leJogJerk,
26     Busy       => G_bJogBusy
27 );
28

```

PLC CPU'dan servo AÇIK sinyalini (G_bSVONCMD) alır ve servo AÇIK komutunu yürütür.

MC_Power_1'in Busy (Meşgul) çıkışı açıldığında, JOG hızlanma, JOG yavaşlama ve JOG sarsıntı değerlerini labellere kaydeder.

PLC CPU'dan JOG başlama sinyalleri ve JOG hızını alır.

Meşgul çıkışını PLC CPU'ya geri döndürür.

(Not) Bu örnek programda, kullanılmayan veya başlangıç değerleri değiştirilmemiş FB'lerin G/Ç sinyalleri göz ardı edilmektedir.

- (3) Motion modülünün programı
2) Başlangıca dönüş (normal yürütme tipi)

```

1  //-----Homing Operation-----
2  //Initial Value Setting, Operation Start Request
3  IF G_bHomingCMD THEN
4      G_lePosition0 := 0.0 ;
5      G_bHomingReq := TRUE ;
6  ELSE
7      G_bHomingReq := FALSE ;
8  END_IF;
9
10 //Homing
11 MC_Home_1(
12     Axis      := Axis0001.AxisRef ,
13     Execute   := G_bHomingReq,
14     Position  := G_lePosition0 ,
15     Done      => bHomingDone ,
16     Busy      => G_bHomingBusy ,
17     CommandAborted => bHomingAborted ,
18     Error     => bHomingError
19 );
20
21 //Done Signal => PLC CPU
22 G_bHomingDone := bHomingDone OR bHomingAborted OR bHomingError;
23

```

PLC CPU'dan başlangıç konumuna dönüş komut sinyalini (G_bHomingCMD) alır.
Başlangıç konumu adresini labela kaydeder ve başlangıç konumuna dönüş isteğini (G_bHomingReq) açar.

G_bHomingCMD kapatıldığında G_bHomingReq'i kapatır.

Başlangıç konumuna dönüş başarıyla tamamlandıktan (Bitti çıkışı AÇIK), yürütme yarıda kesildikten (CommandAborted çıkışı AÇIK) veya bir hata oluştuğundan (Hata çıkışı AÇIK) sonra yürütme tamamlama sinyalini PLC CPU'ya döndürür.

- (3) Motion modülünün programı
3) Pozisyonlama (normal yürütme tipi)

```

1 //-----Positioning Operation-----
2 //Initial Value Setting, Operation Start Request
3 IF G_bPosCMD THEN
4     lePosition1      := 150000.0;
5     lePosVelocity    := 50000.0;
6     lePosAcceleration := 100000.0;
7     lePosDeceleration := 100000.0;
8     lePosJerk        := 200000.0;
9     G_bPosReq := TRUE;
10 ELSE
11     G_bPosReq := FALSE;
12 END_IF;
13
14 //Positioning1
15 MC_MoveAbsolute_1(
16     Axis      := Axis0001.AxisRef ,
17     Execute   := G_bPosReq ,
18     Position  := lePosition1 ,
19     Velocity  := lePosVelocity ,
20     Acceleration := lePosAcceleration ,
21     Deceleration := lePosDeceleration ,
22     Jerk      := lePosJerk ,
23     Direction := MC_DIRECTION__mcShortestWay ,
24     BufferMode := MC_BUFFER_MODE_mcAborting ,
25     Done      => bMoveAbs1Done ,
26     CommandAborted => bMoveAbs1Aborted ,
27     Error     => bMoveAbs1Error
28 );
29 //Dwell
30 TON_1(IN:= bMoveAbs1Done ,PT:= T#500ms ,Q=> bDwell1_out );
31 //Positioning2
32 MC_MoveAbsolute_2(
33     Axis      := Axis0001.AxisRef ,
34     Execute   := bDwell1_out ,
35     Position  := G_lePosition0 ,
36     Velocity  := lePosVelocity ,
37     Acceleration := lePosAcceleration ,
38     Deceleration := lePosDeceleration ,
39     Jerk      := lePosJerk ,
40     Direction := MC_DIRECTION__mcShortestWay ,
41     BufferMode := MC_BUFFER_MODE_mcAborting ,
42     Done      => bMoveAbs2Done ,
43     CommandAborted => bMoveAbs2Aborted ,
44     Error     => bMoveAbs2Error
45 );
46 //Dwell
47 TON_2(IN:= bMoveAbs2Done ,PT:= T#500ms ,Q=> bDwell2_out );
48
49 //Error Signal, Aborted Signal
50 bPosError := bMoveAbs1Error OR bMoveAbs2Error;
51 bPosAborted := bMoveAbs1Aborted OR bMoveAbs2Aborted;
52
53 //Done Signal => PLC CPU
54 G_bPosDone := bDwell2_out OR bPosError OR bPosAborted;

```

PLC CPU'dan pozisyonlama kontrolü başlama sinyalini (G_bPosCMD) alır.
Pozisyonlama için gereken veriler labele kaydedilir ve pozisyonlama kontrolü başlatma isteği (G_bPositioningReq) açılır.

G_bPosCMD PLC CPU tarafından kapatıldıktan sonra G_bPositioningReq'i kapatır.

MC_MoveAbsolute pozisyonlamayı yürütür.

MC_MoveAbsolute_1'in Bitti çıkışı açıldığında bekleme tetikleyen açılış geciktirme zamanlayıcısını başlatır.

Bekleme süresi geçtikten sonra, MC_MoveAbsolute pozisyonlamayı tekrar yürütür.

MC_MoveAbsolute_2'in Bitti çıkışı açıldığında bekleme tetikleyen açılış geciktirme zamanlayıcısını başlatır.

Bekleme süresi geçtikten veya MC_MoveAbsolute'un Hata çıkışı veya CommandAborted çıkışı açıldıktan sonra PLC CPU'ya yürütmenin tamamlama sinyalini döndürür.

- (3) Motion modülünün programı
4) ErrorReset (normal yürütme tipi)

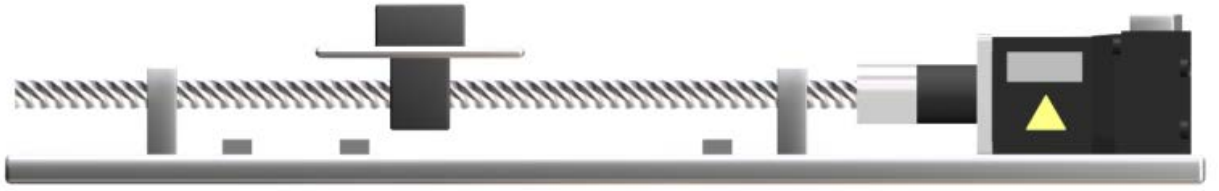
```
1 //Axis Error Reset
2 MC_Reset_1(
3     Axis      := Axis0001.AxisRef ,
4     Execute   := G_bErrorReset
5 );
6
7 //System Error Reset
8 MCv_MotionErrorReset_1(
9     Execute   := G_bErrorReset
10 );
11
```

PLC CPU'dan hata sıfırlama sinyalini (G_bErrorReset) alır ve eksen hatası sıfırlama ve sistem hatası sıfırlama işlemini yürütür.

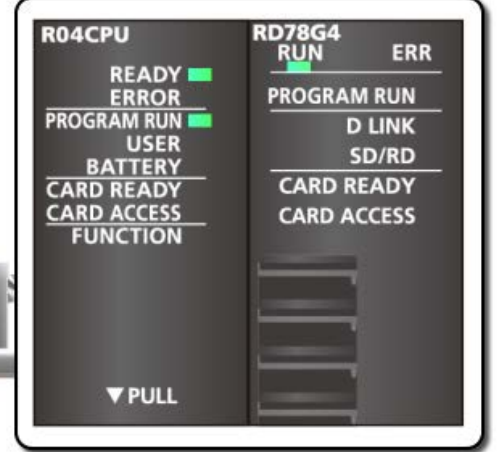
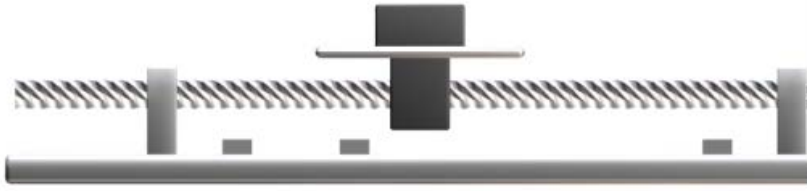
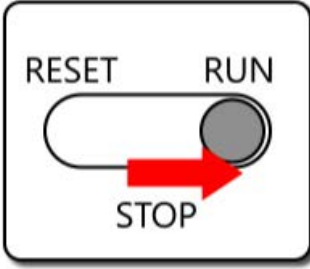
Programı ve parametreleri PLC CPU ve Motion modülüne yazın.

- 1) PLC CPU'daki tüm programlar yeniden oluşturulduktan sonra, tüm verileri PLC CPU'ya yazmak için GX Works3 araç çubuğunda [Online] → [Write to PLC] seçin.
- 2) Parametreler PLC CPU'ya yazıldığında, Motion modülüyle haberleşme etkin hale gelir.
Tüm verileri Motion modülüne yazmak için Motion Control Setting Function araç çubuğunda [Online] → [Write to Module] seçin.
- 3) Yazma işlemini bitirmek için PLC CPU'yu sıfırlayın.

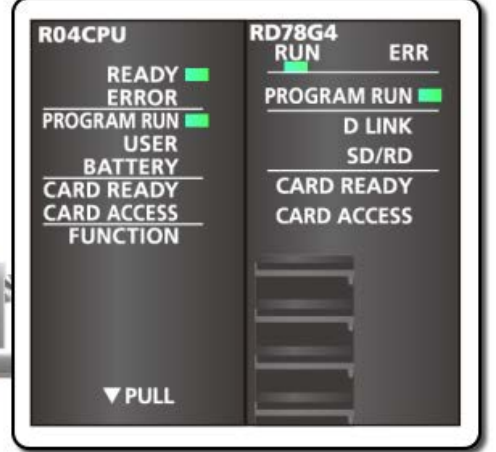
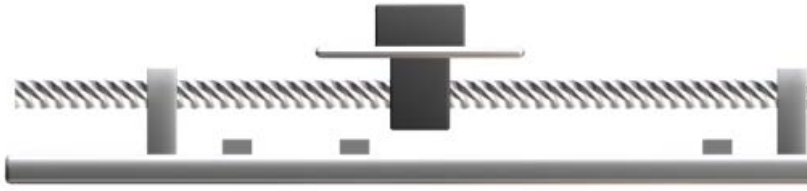
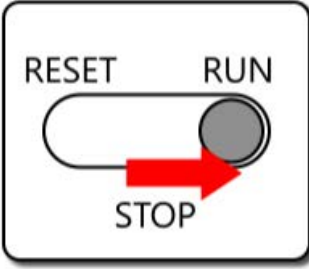
Pencerenin sol alt kısmındaki oynat düğmesini tıklayın.



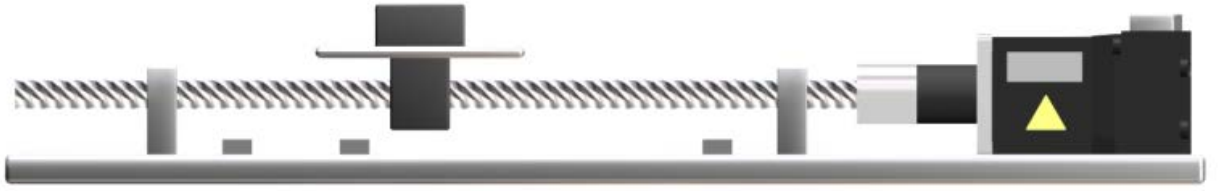
Örnek program çalışmasını kontrol edin.
İşletimi başlatmadan önce, PLC CPU ve Motion modülü programlarının yazıldığından emin olun.



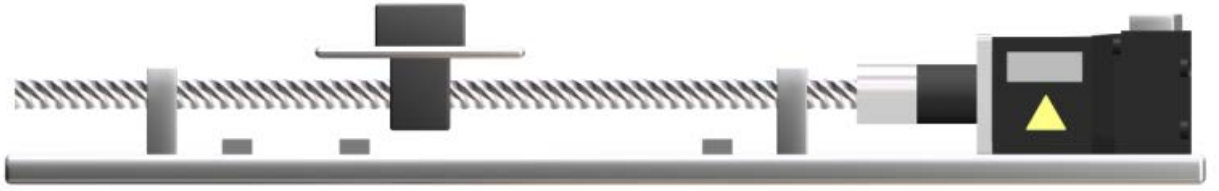
PLC CPU'nun RUN/STOP/RESET düğmesini, RUN durumuna getirin.
PLC CPU'nun READY lambası ve PROGRAM RUN lambası yanar.
Motion modülünün RUN lambası yanar.



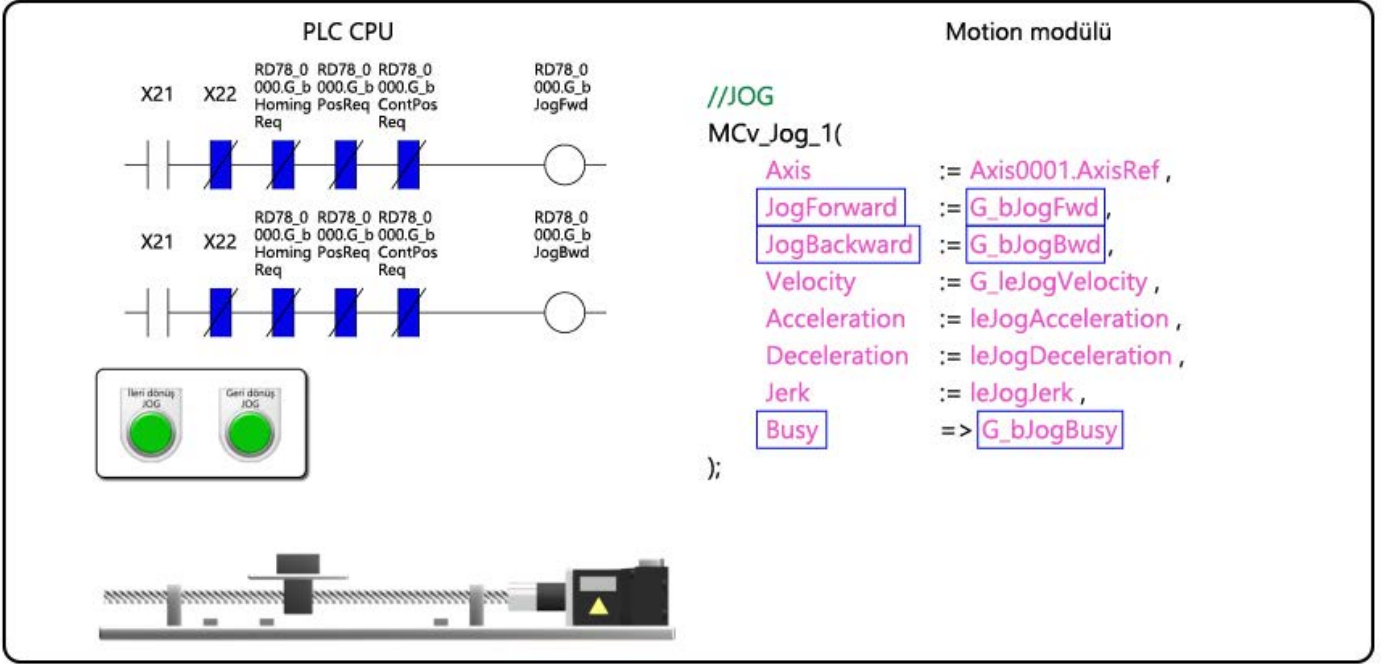
Motion modülünün PROGRAM RUN lambası yanana kadar bekleyin.
Servo sürücü üzerinde "r.01" görüntülenir. (Noktalar yanar.)
Servo motor, servo AÇIK durumuna girer.



Servo KAPALI komutunu yürütmek için X20'yi açın.
Servo sürücü üzerinde "r.01" görüntülenir. (Noktalar yanıp söner.)
Servo AÇIK komutunu tekrar yürütmek için X20'yi kapatın.



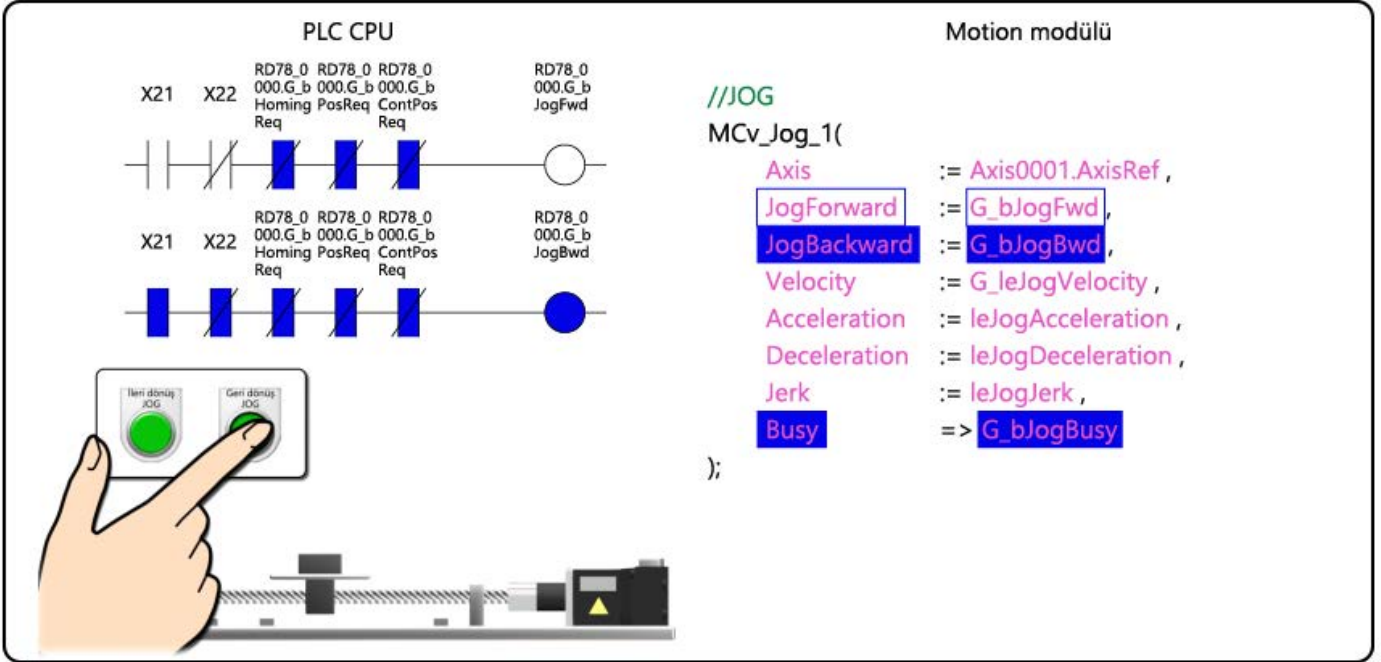
Ekseni, adres artırma yönünde ilerletmek için ileri dönüş JOG'unu (X21) açın ve durdurmak için kapatın.
Ekseni, adres azaltma yönünde ilerletmek için geri dönüş JOG'unu (X22) açın ve durdurmak için kapatın.



Program monitörünü kontrol edin.

X21 açıldığında, "RD78_0000.G_bJogFwd" açılır ve Motion modülü
tarafındaki "G_bJogFwd" açılır.

MCv_Jog_1'in JogForward girişi açıldığında, ileri dönüş JOG'u başlar.



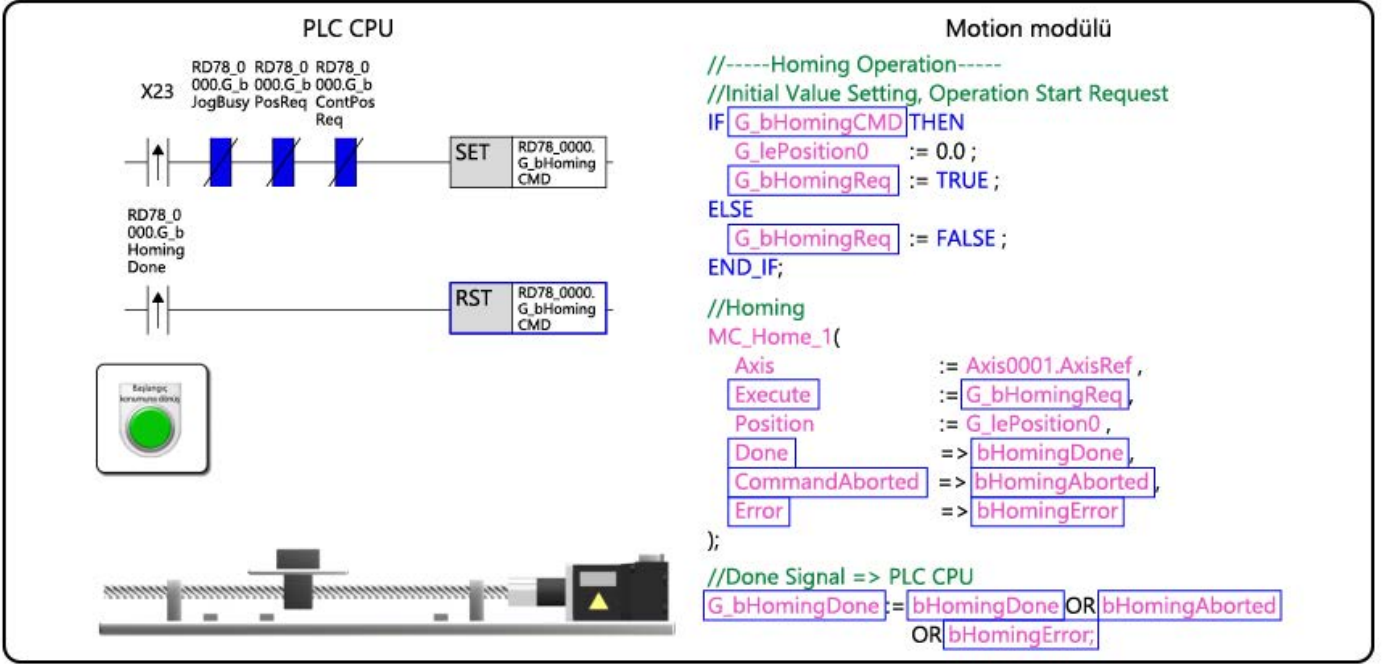
X22 açıldığında, "RD78_0000.G_bJogBwd" açılır ve Motion modülü tarafındaki "G_bJogBwd" açılır.
MCv_Jog_1'in JogBackward girişi açıldığında, geri dönüş JOG'u başlar.



Başlangıç konumuna dönüşü başlatmak için başlangıç konumuna dönüşü (X23) açın.

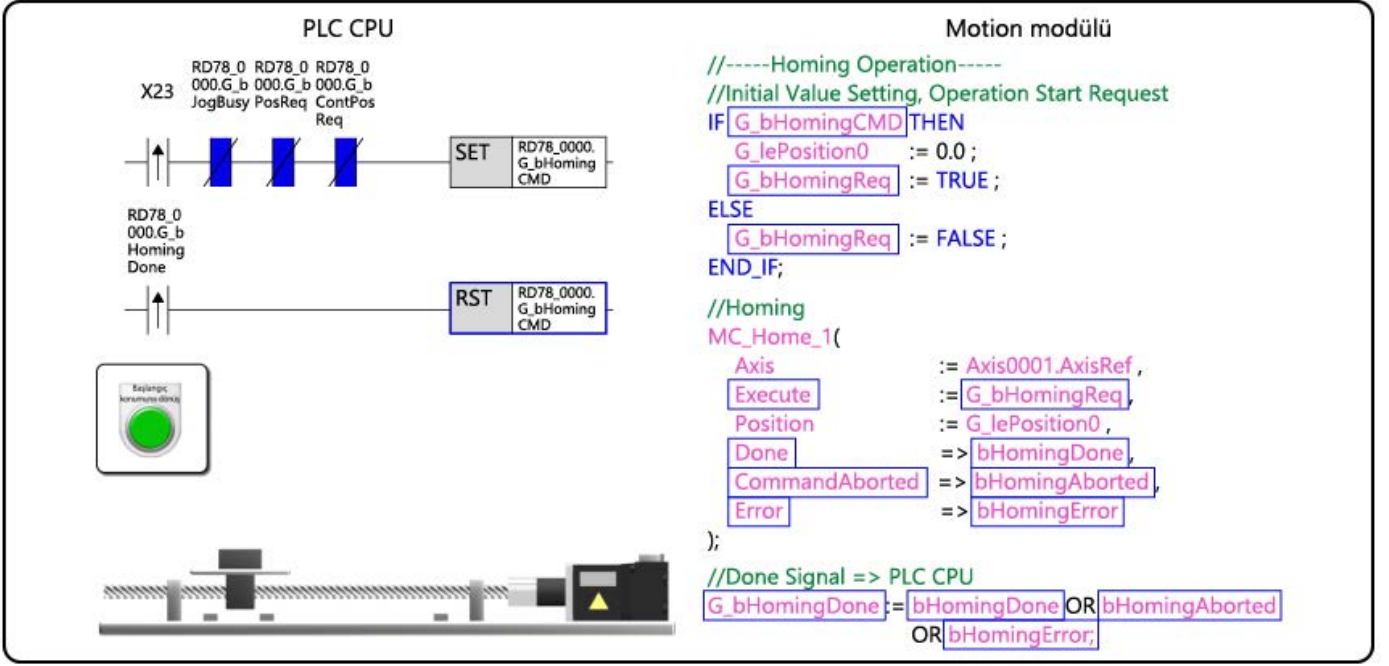
Başlangıç konumuna dönüşü proximity dog yöntemiyle yürütün
(Pr.PT45'ten 33 çıkarılır)

Eksen, dog'un biraz ilerisinde durur ve noktayı başlangıç konumu olarak ayarlar.



Program monitörünü kontrol edin.

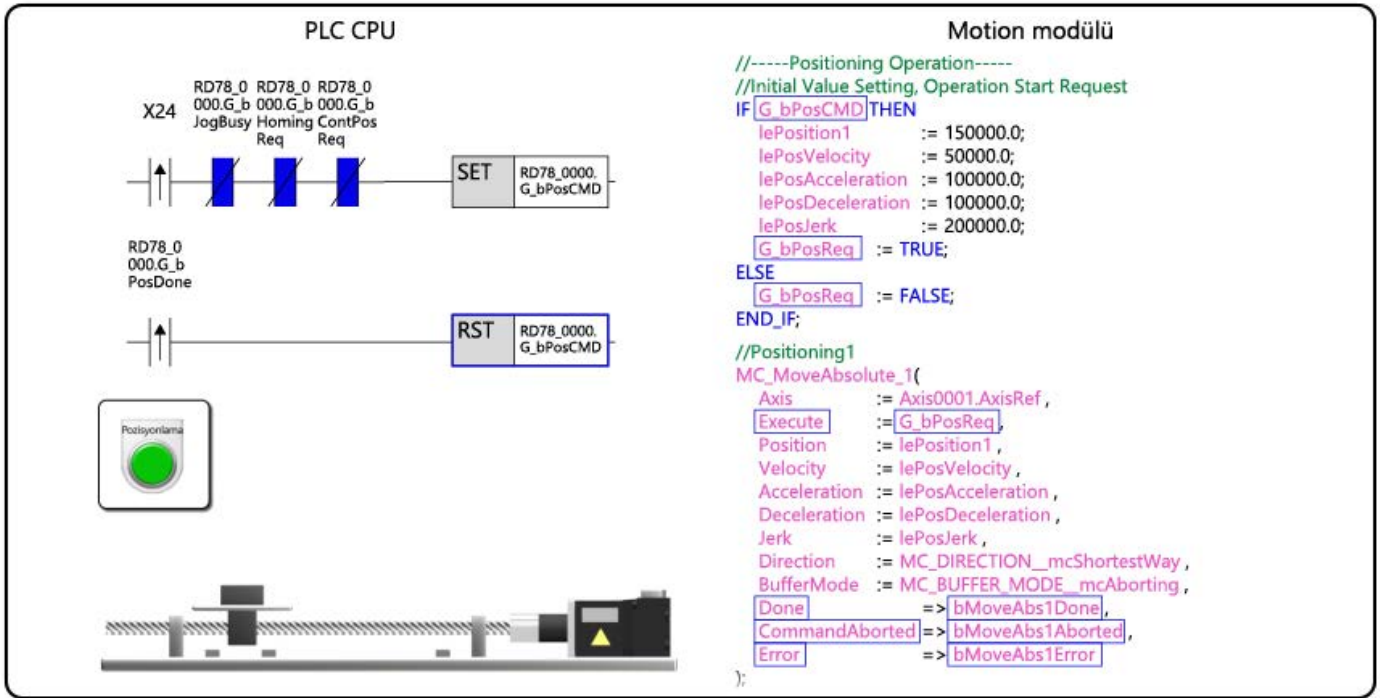
X23 açıldığında, "RD78_0000.G_bHomingCMD" ayarlanır.
Motion modülü tarafındaki "G_bHomingCMD" açılır ve MC_Home_1'in yürütme komutu olan "G_bHomingReq" açılır.



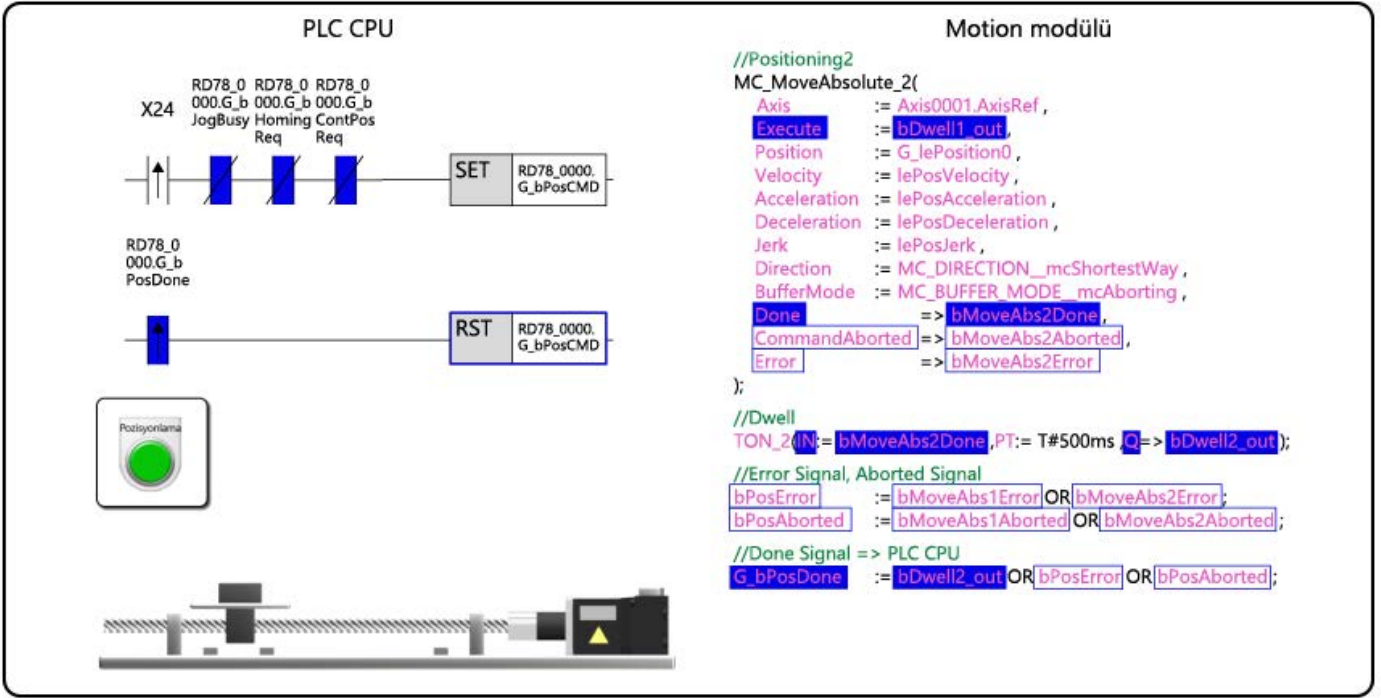
Başlangıç konumuna dönüş tamamlandığında, Bitti çıkışı ve "G_bHomingDone" açılır.
PLC CPU üzerindeki "G_bHomingCMD" ilk durumuna sıfırlanır.



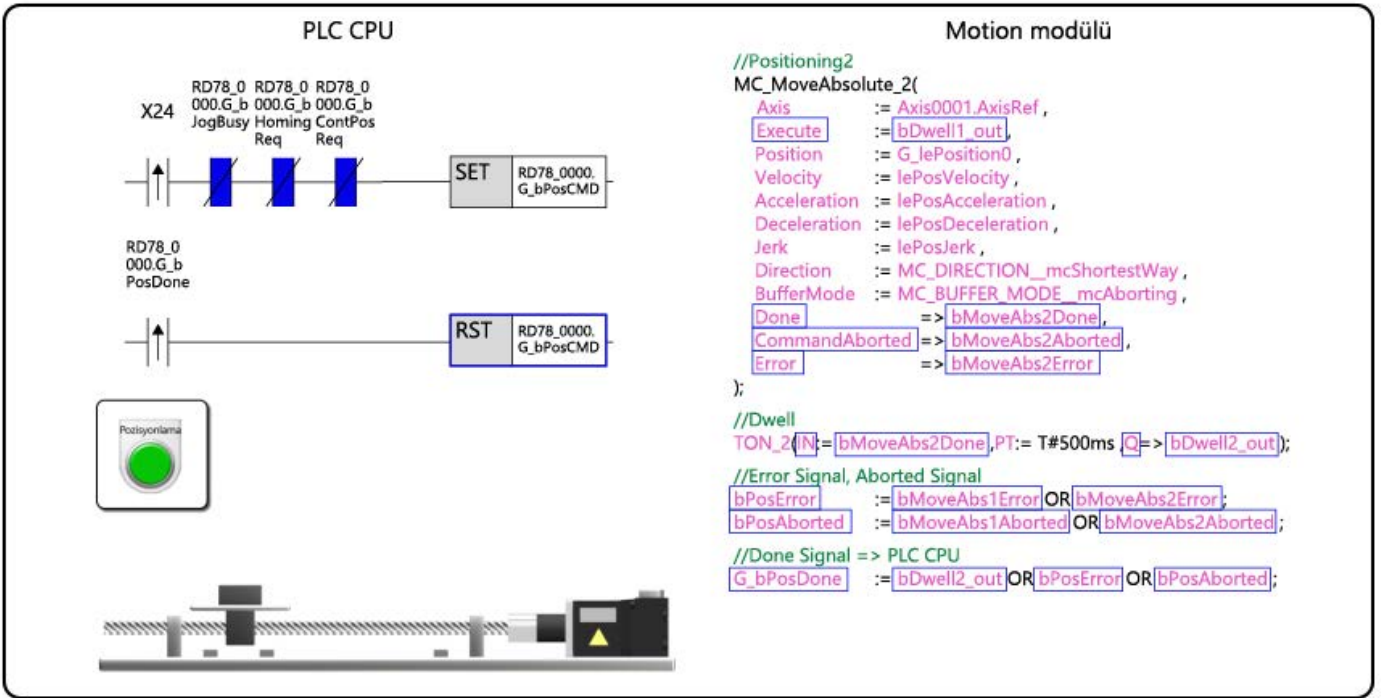
Pozisyonlama başlatmanın (X24) açılması ileri-geri hareketi başlatır. Eksen 150 mm ileri gider ve 0,5 saniye durur ve 150 mm geri gider ve 0,5 saniye durur.



Program monitörünü kontrol edin.
 X24 açıldığında, "RD78_0000.G_bPosCMD" ayarlanır.
 Motion modülü tarafındaki "G_bPosCMD" açılır ve
 MC_MoveAbsolute_1'in yürütme komutu olan "G_bPosReq" açılır.



İleri-geri hareket bittikten ve bekleme süresi geçtikten sonra,
 "G_bPosDone" açılır.
 PLC CPU üzerindeki "G_bPosCMD" ilk durumuna sıfırlanır.



Bu şekilde çalışma kontrolü tamamlanır.
Sonraki sayfaya gidin.

Bu bölümde, şunları öğrendiniz:

- Açık Label nedir?
- Açık Label Ayarı
- Program Örneği
- Yazma Programı
- Çalışma Kontrolü

Nokta

Açık Label nedir?	<ul style="list-style-type: none">• Açık label, hem Motion modülü hem de PLC CPU'da kullanılabilen ortak labeldir.
Açık Label Ayarı	<ul style="list-style-type: none">• Açık labelleri, Motion modülünün genel labellerinden kaydedin.• Her bir labelin PLC CPU'dan okunacağını mı, yoksa PLC CPU'ya yazılacağını mı seçin.• Sistemde hazırlanan yapılandırılmış (Structured) veri türünün üyelerini açık label olarak ayarlamak için, açık labelleri yapılandırılmış (Structured) verinin katmanına göre kaydedin.• Açık labelleri Motion modülünde ayarladıktan sonra, tüm programları yeniden oluşturun ve açık labelleri yansıtın.• Açık labeler PLC CPU tarafındaki modül labeline kaydedilir.
Program Örneği	<ul style="list-style-type: none">• Bu bölümde şu program örneği açıklanmıştır: pozisyonlama başlatma sinyali ve pozisyonlama tamamlama sinyalini alıp vermek için açık labeleri kullanan PLC CPU'nun ladder programı.
Yazma Programı	<ul style="list-style-type: none">• Verileri ilk olarak PLC CPU ve ardından Motion modülüne yazın.
Çalışma Kontrolü	<ul style="list-style-type: none">• Örnek programın çalışmasını videoda kontrol ettiniz.

Ara bellek (buffer) modu, Motion control FB'nin çoklu işletim FB'lerini başlatarak işlemleri sürekli olarak yürütür. Motion control FB'nin BufferMode girişiyle ayarlanabilir. Her eksen ve eksen grubu için aynı anda en fazla iki FB başlatılabilir.

(Örnek) MC_MoveAbsolute

```
MC_MoveAbsolute_1(
  Axis           := Axis0001.AxisRef ,
  Execute        := G_bPositioningReq ,
  ContinuousUpdate := FALSE ,
  Position       := lePosition1 ,
  Velocity       := lePosVelocity ,
  Acceleration   := lePosAcceleration ,
  Deceleration   := lePosDeceleration ,
  Jerk           := lePosJerk ,
  Direction      := MC_DIRECTION_mcShortestWay ,
  BufferMode      := MC_BUFFER_MODE__mcAborting ,
  Options        := 0 , //mcAccDec
  Done           => bMoveAbs1Done ,
  CommandAborted => bMoveAbs1Aborted ,
  Error          => bMoveAbs1Error
);
```

0 veya MC_BUFFER_MODE__mcAborting
 1 veya MC_BUFFER_MODE__mcBuffered
 2 veya MC_BUFFER_MODE__mcBlendingLow
 3 veya MC_BUFFER_MODE__mcBlendingPrevious
 4 veya MC_BUFFER_MODE__mcBlendingNext
 5 veya MC_BUFFER_MODE__mcBlendingHigh

- Yürütülmekte olan FB yarıda kesilir ve sonraki FB hemen yürütülür.
- Yürütülmekte olan FB'nin işletimi tamamlandıktan sonra, sıradaki FB yürütülür.
- Yürütülmekte olan FB için hedef hızlardan düşük olanı ve ara belleğe alınacak FB, anahtarlama hızı olarak ayarlanır.
- Yürütülmekte olan FB'nin hedef hızı anahtarlama hızı olarak ayarlanır.
- Ara belleğe alınacak olan FB'nin hedef hızı anahtarlama hızı olarak ayarlanır.
- Yürütülmekte olan FB için hedef hızlardan yüksek olanı ve ara belleğe alınacak FB, anahtarlama hızı olarak ayarlanır.

[Nokta]

Direction ve BufferMode girişi için, MC_BUFFER_MODE ve MC_DIRECTION ile başlayan ENUM enümeratörleri veya numaralar belirtin.

ENUM enümeratörlerinin ayrıntıları için, aşağıdaki kılavuza başvurun.



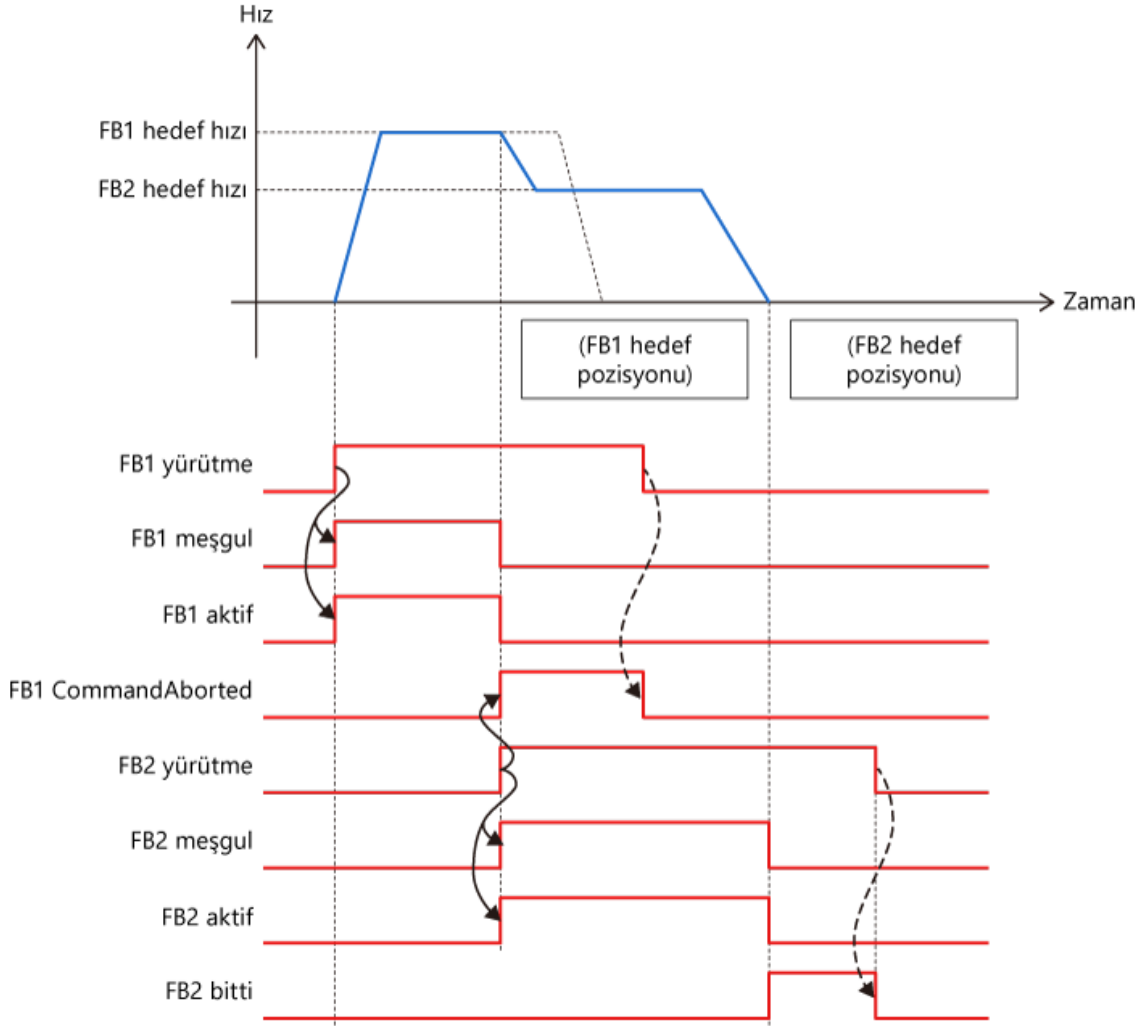
MELSEC iQ-R Programming Manual (Motion Control Function Blocks)

2 VARIABLES AND MOTION CONTROL FB

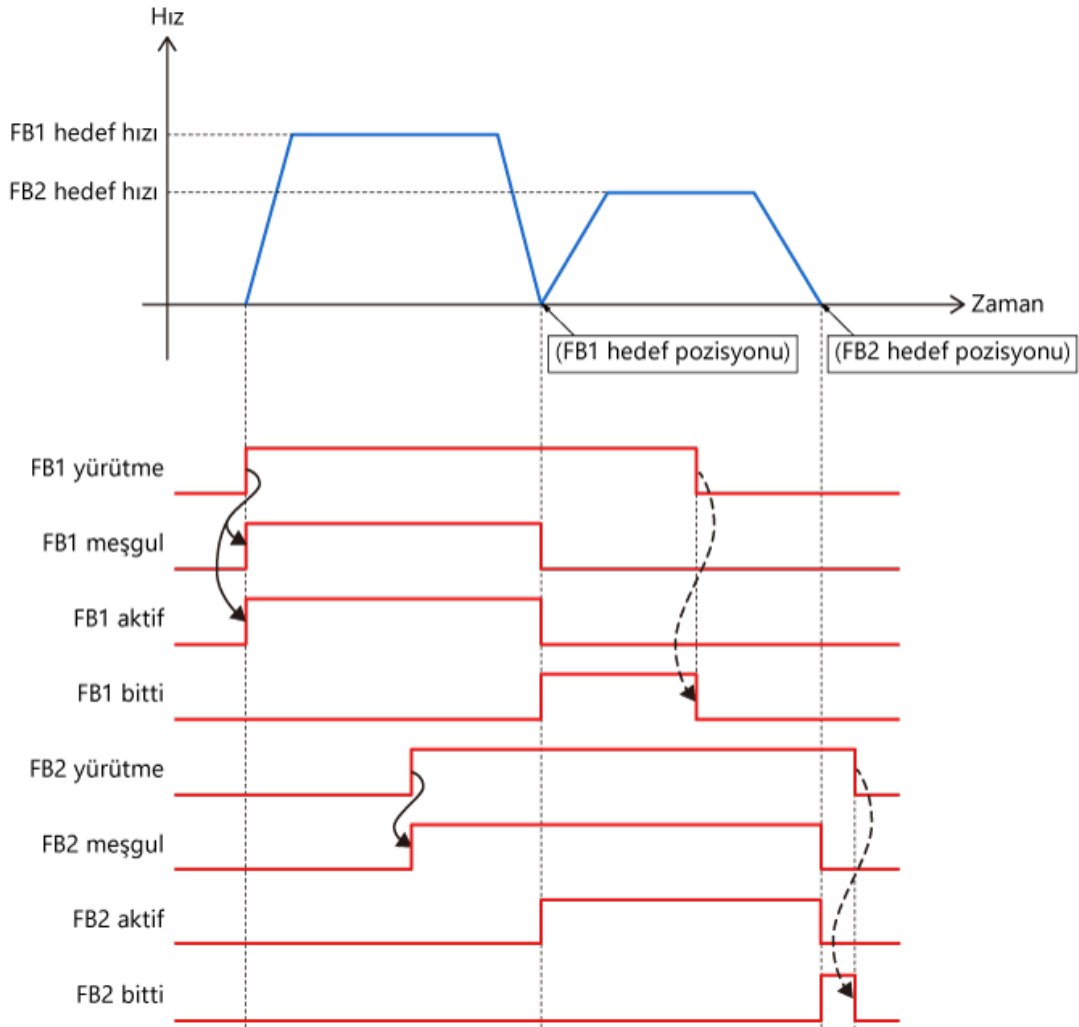
2.2 List of Enumerators

Aşağıdaki şemada, BufferMode 0:

mcAborting olarak ayarlandığındaki işletim gösterilmektedir. Yürütülmekte olan FB yarıda kesilir ve sonraki FB hemen yürütülür.



Aşağıdaki şemada, BufferMode 1: mcBuffered olarak ayarlandığındaki işletim gösterilmektedir. Yürütülmekte olan FB'nin işletimi tamamlandığında, sonraki FB yürütülür.



BufferMode, mcBlending*** olarak ayarlandığında, sonraki FB, yürütülmekte olan FB'nin hedef pozisyonuna ulaşıldığında sürekli olarak yürütülür.

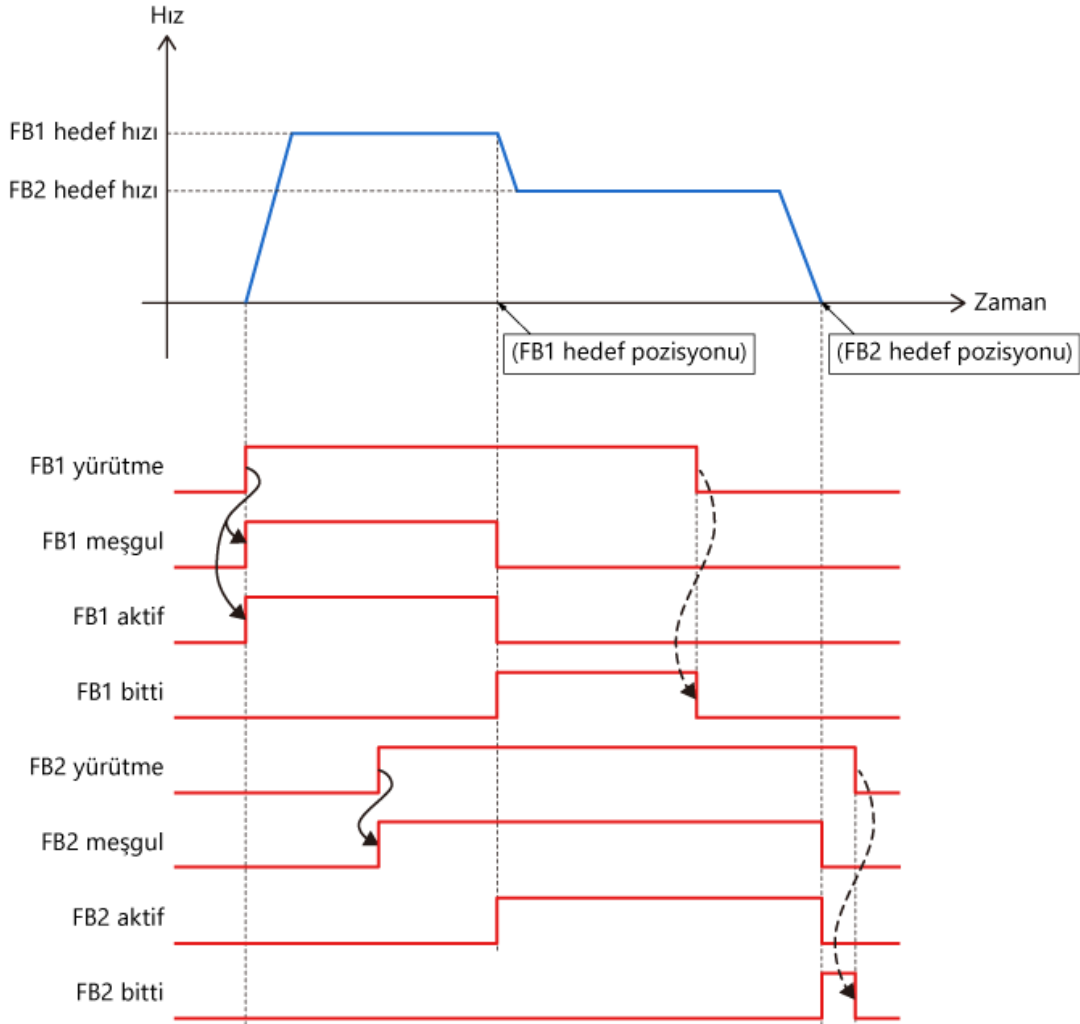
Aşağıdaki açıklamada, ilk yürütülecek FB, FB1 olup ara belleğe alınacak FB, FB2'dir.

(1) BlendingPrevious

Aşağıdaki şemada, BufferMode 3 olarak ayarlandığındaki işletim gösterilmektedir: mcBlendingPrevious.

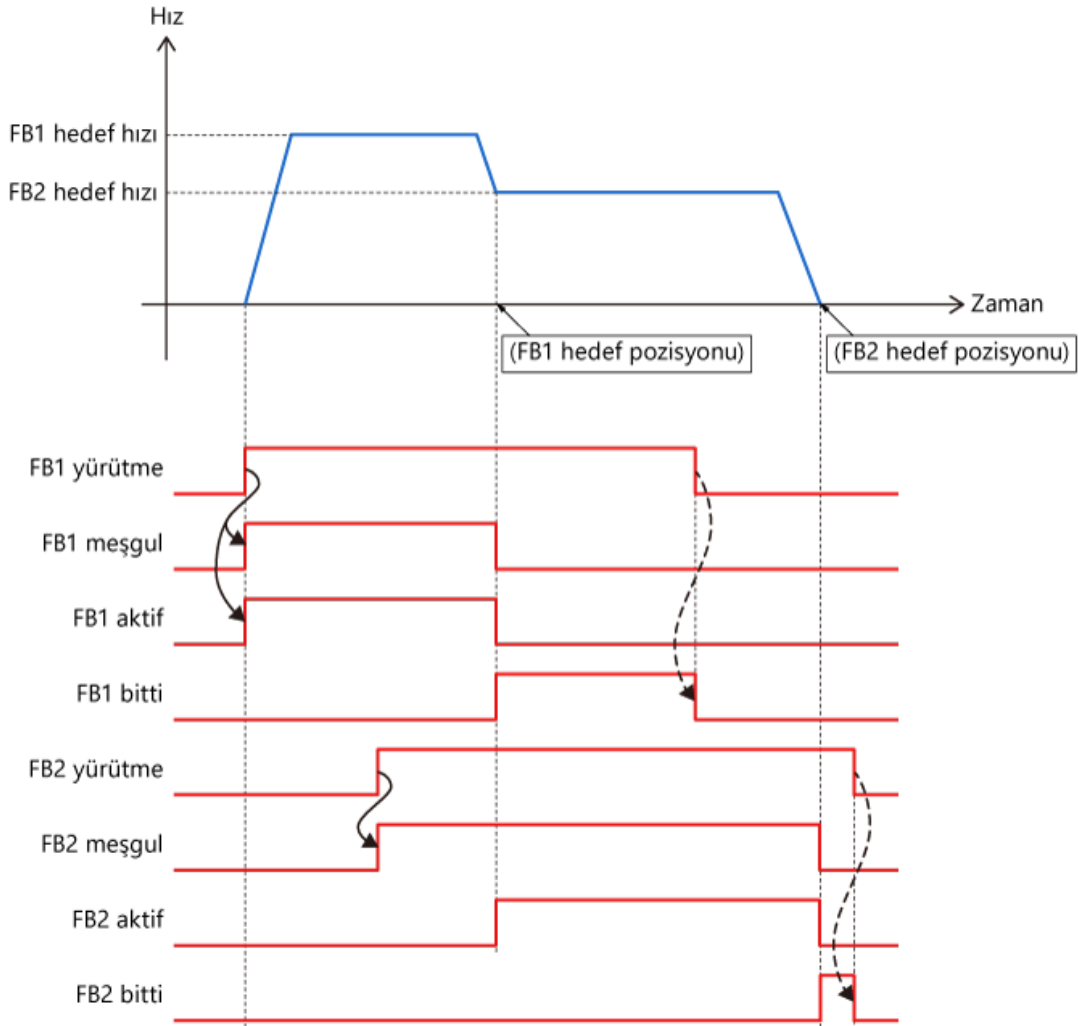
İşletim, FB1'in hedef pozisyonuna kadar FB1'in hedef hızında gerçekleştirilir.

İşletim FB2'ye geçtiğinde, hız FB2'nin hedef hızı olarak değiştirilir ve FB2'nin hedef pozisyonuna hareket eder.



(2) BlendingNext

Aşağıdaki şemada, BufferMode 4 olarak ayarlandığındaki işletim gösterilmektedir: mcBlendingNext. İşletim FB1'in hedef pozisyonuna ulaştığında, hız FB2'nin hedef hızı olarak değişir.



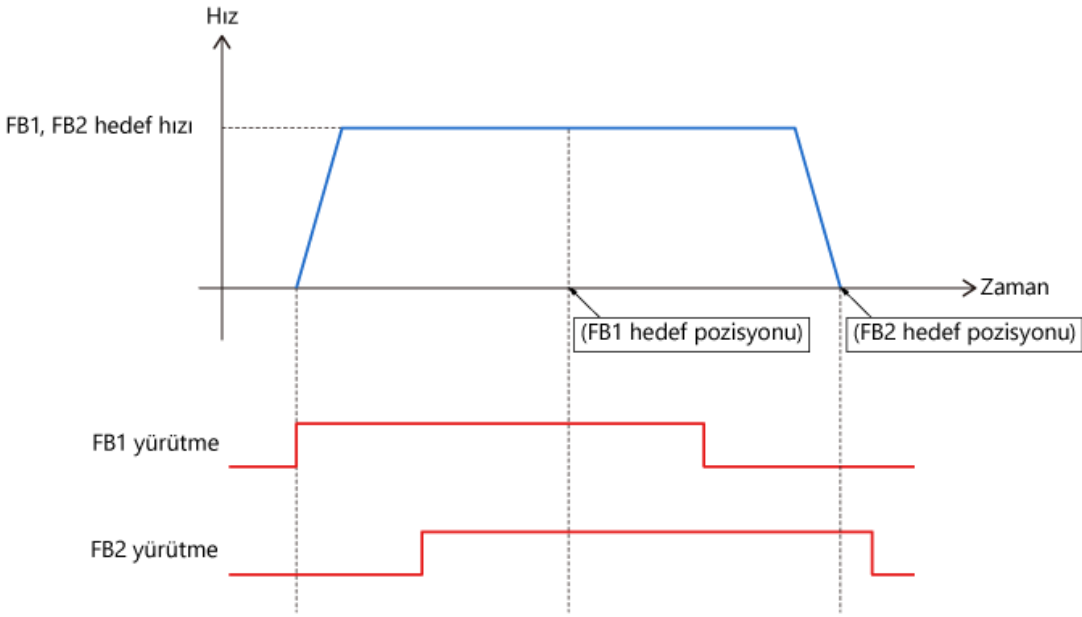
(3) BlendingLow, BlendingHigh

BufferMode 2: mcBlendingLow veya 5: mcBlendingHigh olarak ayarlandığındaki işletim, FB1 ve FB2'nin hangi hedef hızlarının daha yüksek olduğuna bağlı olarak değişiklik gösterir.

Ayar değeri	FB1 hedef hızı > FB2 hedef hızı	FB1 hedef hızı < MFB2 hedef hızı
2: mcBlendingLow	BlendingPrevious ile aynı işletim	BlendingNext ile aynı işletim
5: mcBlendingHigh	BlendingNext ile aynı işletim	BlendingPrevious ile aynı işletim

[Nokta]

Aşağıdaki şemada FB1 ve FB2'nin hızı aynı olduğunda, BlendingPrevious, BlendingNext, BlendingHigh, ve BlendingLow için hız dalga formu gösterilmektedir.



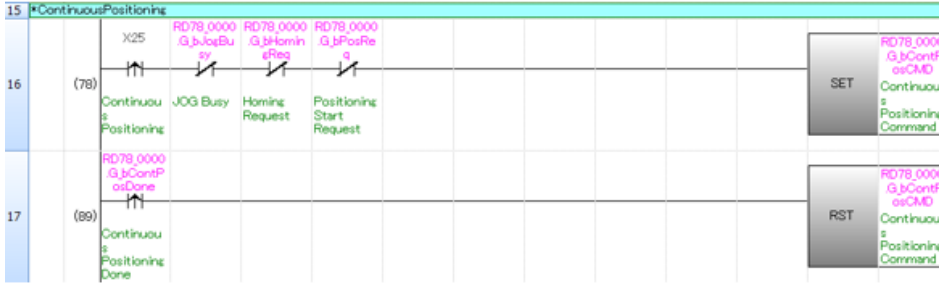
(1) Örnek program işletimi

Bu bölümde, Bölüm 2'de kullanılan örnek program kullanılmaktadır.

X25 ile başlayan bir programdaki ara bellek modu işletim farkını kontrol edin.

Bileşen	FB1 (MC_MoveAbsolute)	FB2 (MC_MoveAbsolute)
Pozisyonlama adresi	75000,0 [µm]	150000,0 [µm]
Hız	50000,0 [µm/s]	25000,0 [µm/s]
Hızlanma, yavaşlama	100000,0 [µm/s ²]	50000,0 [µm/s ²]
Sarsıntı	200000,0 [µm/s ³]	100000,0 [µm/s ³]

- (2) PLC CPU'nun programı
MAIN (ladder, tarama programı)



Sürekli pozisyonlama kontrolü başlatmanın yükselme eşiği (X25), G_bContPosCMD ile tutulur ve sürekli pozisyonlama kontrolünün başlama koşulu olarak Motion modülüne gönderilir. Başka bir program çalışırken pozisyonlama kontrolünün başlatılmasını engellemek için bir ara kilit ayarlanır. Motion modülünün sürekli başlangıç konumuna dönüş tamamlama sinyalini açtığı alındığında, bu sinyalin yükselme eşiğinde G_bContPosCMD sıfırlanır.

(3) Motion modülü programı

Sürekli Pozisyonlama (normal yürütme tipi)

```

1  //-----Continuous Positioning Operation-----
2  //Initial Value Setting, Operation Start Request
3  IF G_bContPosCMD THEN
4      lePosition1 := 75000.0;
5      lePosVelocity1 := 50000.0;
6      lePosAcceleration1 := 100000.0;
7      lePosDeceleration1 := 100000.0;
8      lePosJerk1 := 200000.0;
9      lePosition2 := 150000.0;
10     lePosVelocity2 := 25000.0;
11     lePosAcceleration2 := 50000.0;
12     lePosDeceleration2 := 50000.0;
13     lePosJerk2 := 100000.0;
14     G_bContPosReq := TRUE;
15 ELSE
16     G_bContPosReq := FALSE;
17 END_IF;
18
19 FB1 //Positioning1
20 MC_MoveAbsolute_1(
21     Axis := Axis0001.AxisRef ,
22     Execute := G_bContPosReq ,
23     Position := lePosition1 ,
24     Velocity := lePosVelocity1 ,
25     Acceleration:= lePosAcceleration1 ,
26     Deceleration:= lePosDeceleration1 ,
27     Jerk := lePosJerk1 ,
28     Direction := MC_DIRECTION_ncShortestWay ,
29     Done => bMoveAbs1Done ,
30     Active => bMoveAbs1Active ,
31     CommandAborted => bMoveAbs1Aborted ,
32     Error => bMoveAbs1Error
33 );
34 FB2 //Positioning2
35 MC_MoveAbsolute_2(
36     Axis := Axis0001.AxisRef ,
37     Execute := bMoveAbs1Active ,
38     Position := lePosition2 ,
39     Velocity := lePosVelocity2 ,
40     Acceleration:= lePosAcceleration2 ,
41     Deceleration:= lePosDeceleration2 ,
42     Jerk := lePosJerk2 ,
43     Direction := MC_DIRECTION_ncShortestWay ,
44     BufferMode := MC_BUFFER_MODE_ncBuffered ,
45     Done => bMoveAbs2Done ,
46     CommandAborted => bMoveAbs2Aborted ,
47     Error => bMoveAbs2Error
48 );
49 //Dwell
50 SET(bMoveAbs2Done,bDwell_In);
51 TON_1(IN:= bDwell_In ,PT:= T#100ms ,Q=> bDwell_out);
52
53 //Error Signal, Aborted Signal
54 bError := bMoveAbs1Error OR bMoveAbs2Error;
55 bAborted := bMoveAbs1Aborted OR bMoveAbs2Aborted;
56
57 //Done Signal => PLC CPU
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
59 //Reset Dwell_In
60 RST(G_bContPosDone,bDwell_In);
61

```

PLC CPU'dan sürekli pozisyonlama kontrolü başlama sinyali (G_bContPosCMD) alır. Sürekli pozisyonlama için gereken veriler labela kaydedilir ve pozisyonlama kontrolü başlatma isteği (G_bContPositioningReq) açılır.

G_bContPosCMD kapatıldığında G_bContPositioningReq'i kapatır.

FB1 (MC_MoveAbsolute_1) pozisyonlamayı yürütür.

FB2 (MC_MoveAbsolute_2), FB1(MC_MoveAbsolute_1) Aktif çıkışı tarafından başlatılır. Ardından, FB1 çalışırken FB2 ara belleğe alınır.

FB2'nin BufferMode girişini (MC_MoveAbsolute_2) değiştirin ve ara bellek modunun çalışmasını kontrol edin.

Açılış geciktirme zamanlayıcısını tetikleyen bekleme, FB2'nin Bitti çıkışı (MC_MoveAbsolute_2) tarafından başlatılır.

Bekleme süresi geçtikten veya MC_MoveAbsolute'un Hata çıkışı veya CommandAborted çıkışı açıldıktan sonra PLC CPU'ya yürütmenin tamamlama sinyali döndürür. Aynı zamanda, açılış geciktirme zamanının girişi sıfırlanır.

Pencerenin sol alt kısmındaki oynat düğmesini tıklayın.

```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis       := Axis0001.AxisRef ,  
37     Execute    := bMoveAbs1Active ,  
38     Position   := lePosition2 ,  
39     Velocity   := lePosVelocity2 ,  
40     Acceleration := lePosAcceleration2 ,  
41     Deceleration := lePosDeceleration2 ,  
42     Jerk       := lePosJerk2 ,  
43     Direction  := MC_DIRECTION__mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcBuffered ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Ara bellek modu çalışmasını kontrol edin.
Örnek programda "LinearInterpolation" açın ve ara bellek modunun çalışmasını kontrol etmek için MC_MoveAbsolute_2'nin BufferMode girişini değiştirin.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```



```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcBuffered ,  
45     Done      => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error     => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

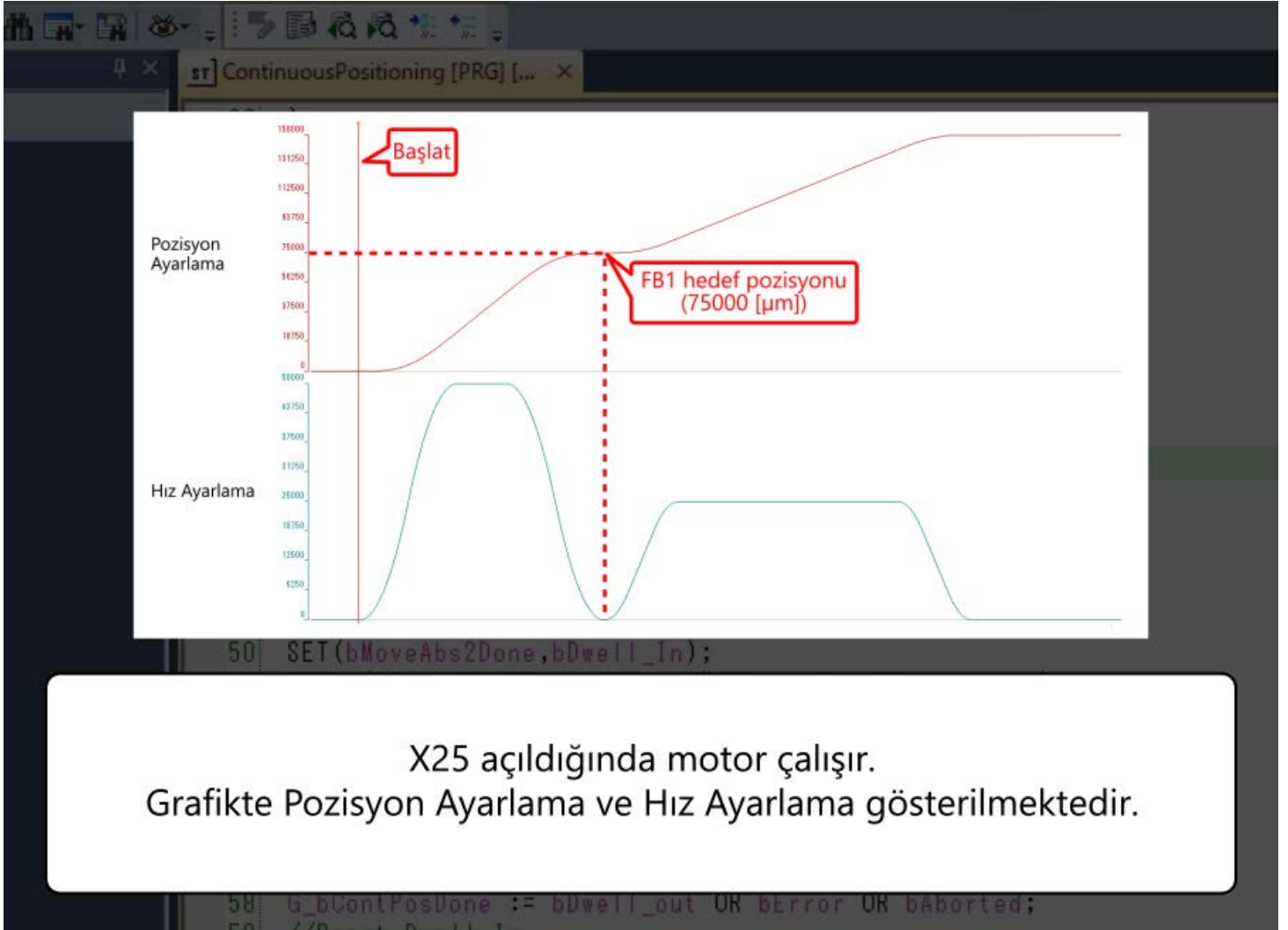
Program indirildiğinde mcBuffered ön ayarlıdır.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```

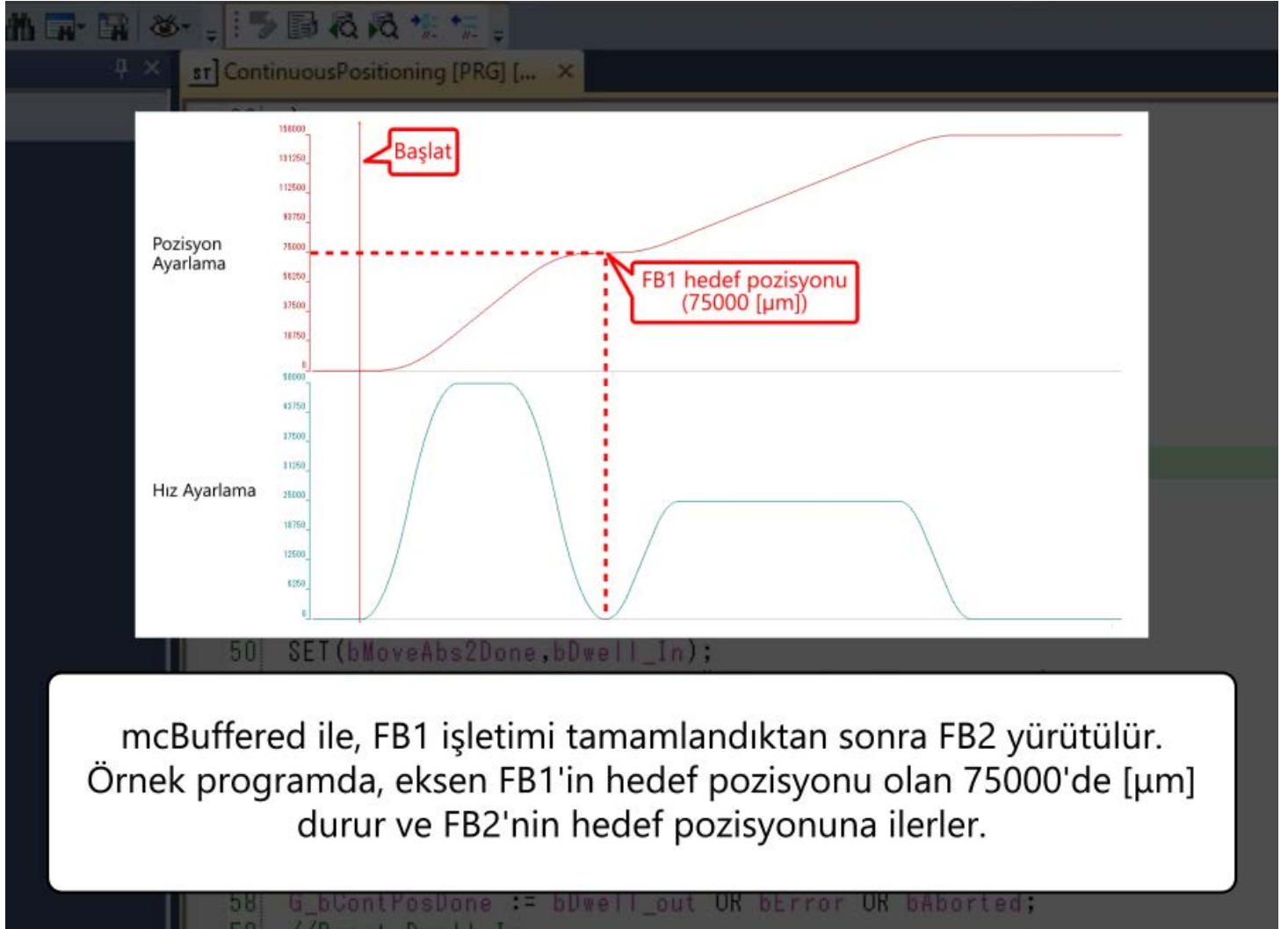
```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE_mcBuffered ,  
45     Done      => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error     => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

İlk olarak, mcBuffered çalışmasını kontrol edin.
MC_MoveAbsolute_2'nin BufferMode girişinin
"MC_BUFFER_MODE_mcBuffered" olarak ayarlandığını kontrol edin ve
programı Motion modülüne yazın.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```



X25 açıldığında motor çalışır.
Grafikte Pozisyon Ayarlama ve Hız Ayarlama gösterilmektedir.



mcBuffered ile, FB1 işletimi tamamlandıktan sonra FB2 yürütülür. Örnek programda, eksen FB1'in hedef pozisyonu olan 75000'de [μm] durur ve FB2'nin hedef pozisyonuna ilerler.

```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE_mcAborting ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Daha sonra, mcAborting işletimini kontrol edin. MC_MoveAbsolute_2'nin BufferMode girişini "MC_BUFFER_MODE_mcAborting" olarak değiştirin, tüm programları yeniden oluşturun ve Motion modülüne yazın.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```



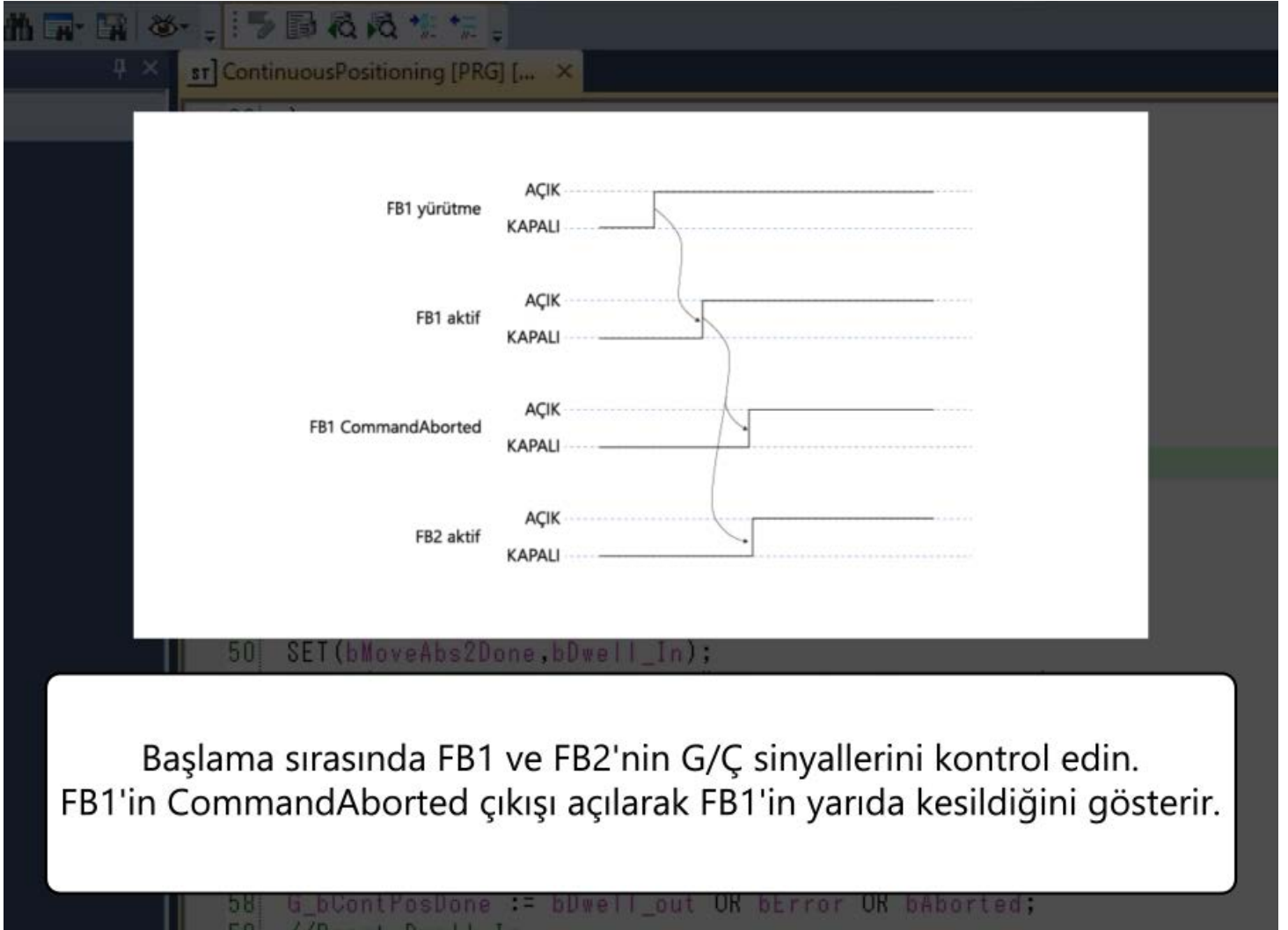
X25 açıldığında motor çalışır.
Grafikte Pozisyon Ayarlama ve Hız Ayarlama gösterilmektedir.

58 G_bContPosDone := bDwell_out OR bError OR bAborted;



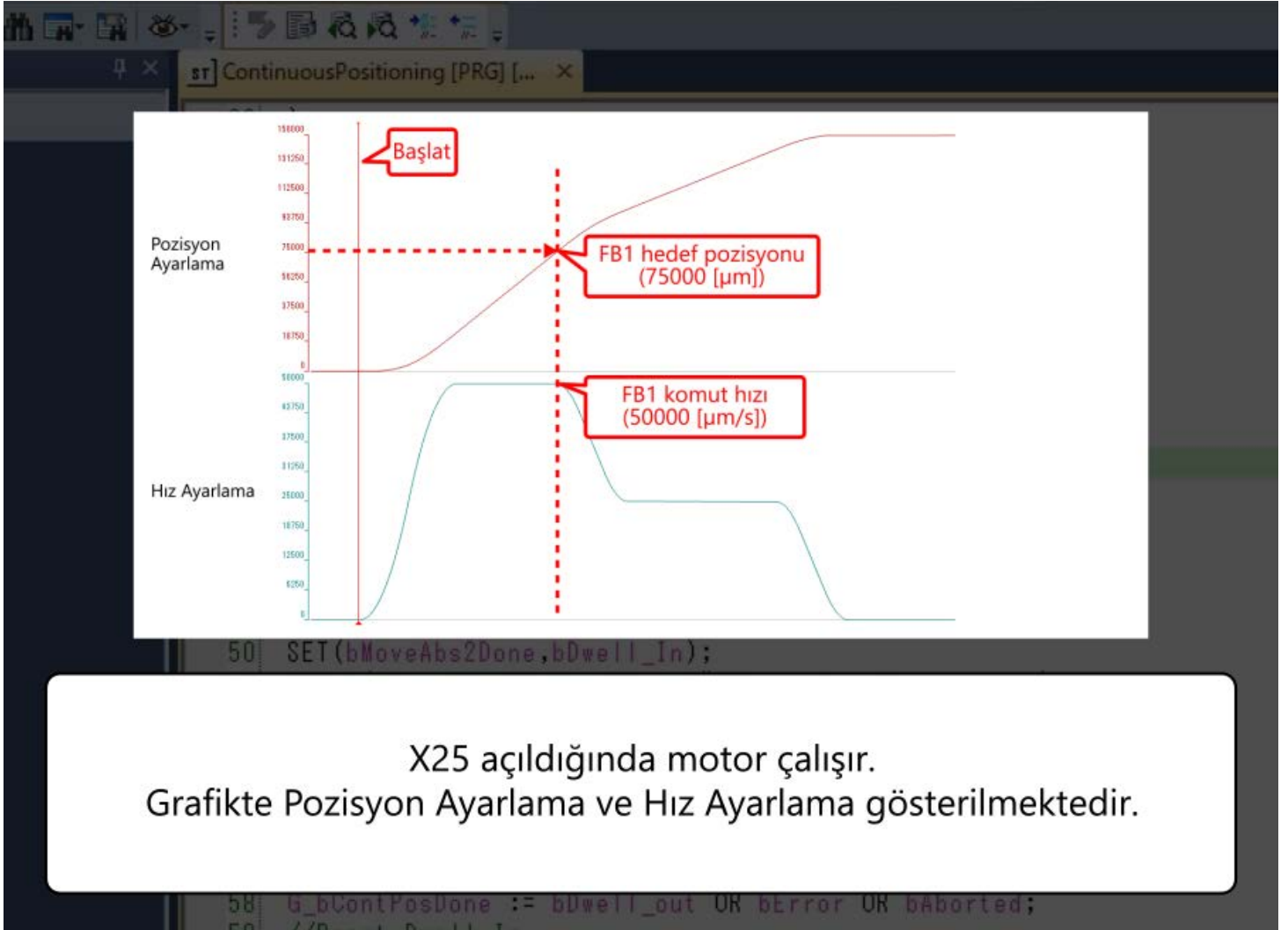
mcAborting ile, FB2 yürütme açıldığında FB1 yarıda kesilir. Örnek programda, FB1 işletimi başlattıktan hemen sonra FB2 yürütme açıldığından, işletim FB2 tek başına yürütüldüğü zamanla neredeyse aynıdır.

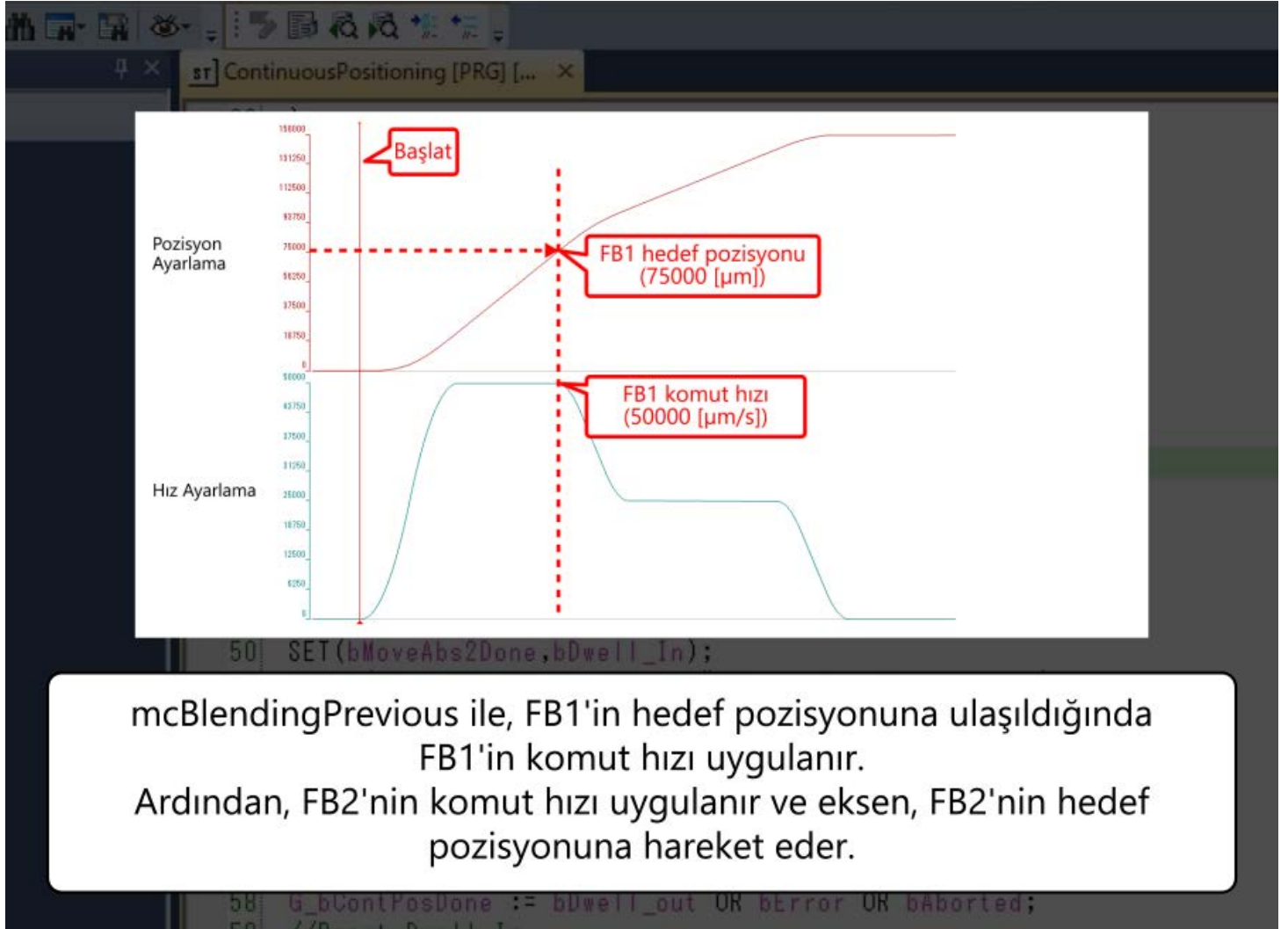
```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```




```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION__mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcBlendingPrevious ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Daha sonra, mcBlendingPrevious işletimini kontrol edin.
MC_MoveAbsolute_2'nin BufferMode girişini
"MC_BUFFER_MODE__mcBlendingPrevious" olarak değiştirin, tüm
programları yeniden oluşturun ve Motion modülüne yazın.



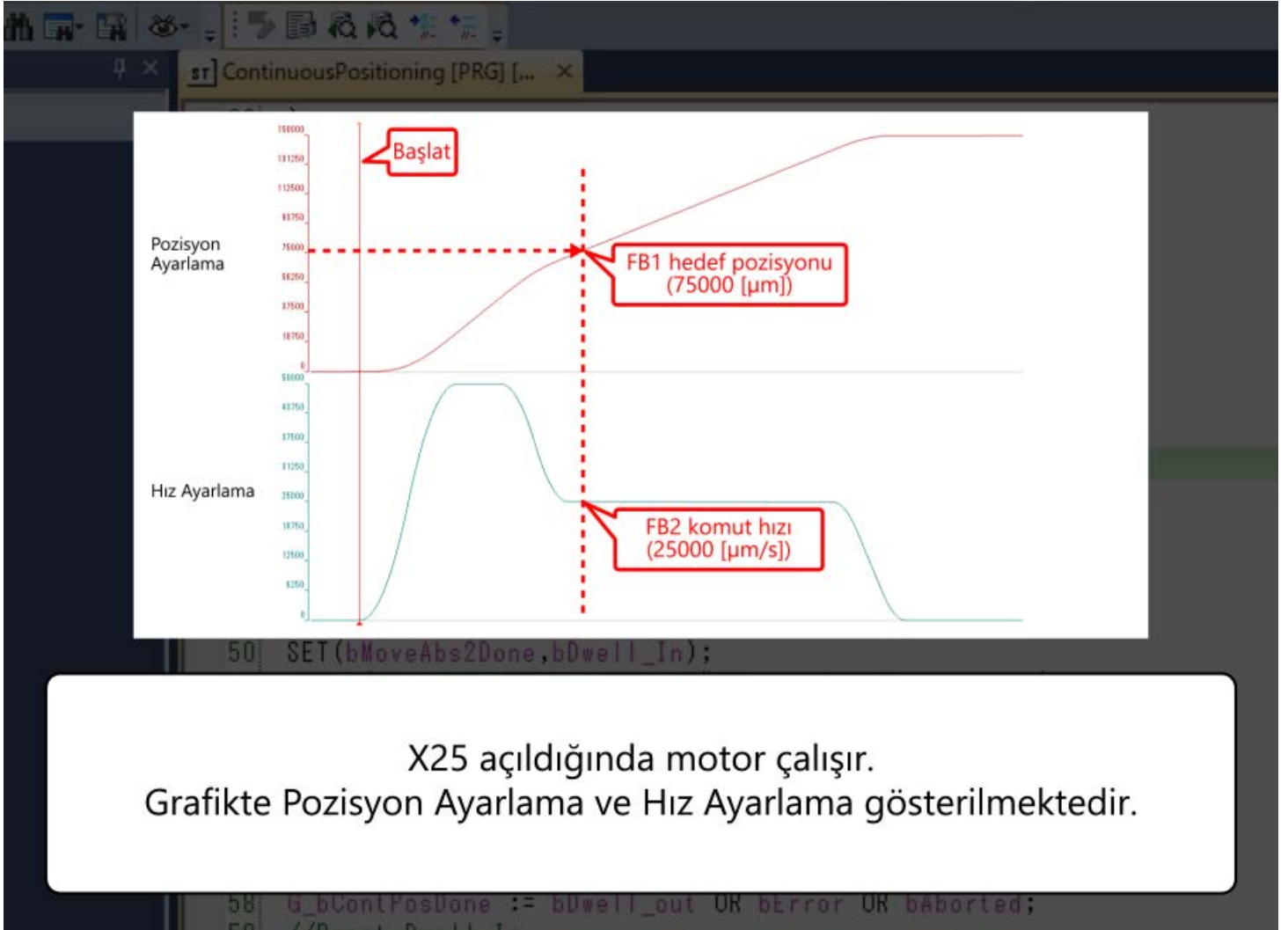


mcBlendingPrevious ile, FB1'in hedef pozisyonuna ulaşıldığında FB1'in komut hızı uygulanır. Ardından, FB2'nin komut hızı uygulanır ve eksen, FB2'nin hedef pozisyonuna hareket eder.

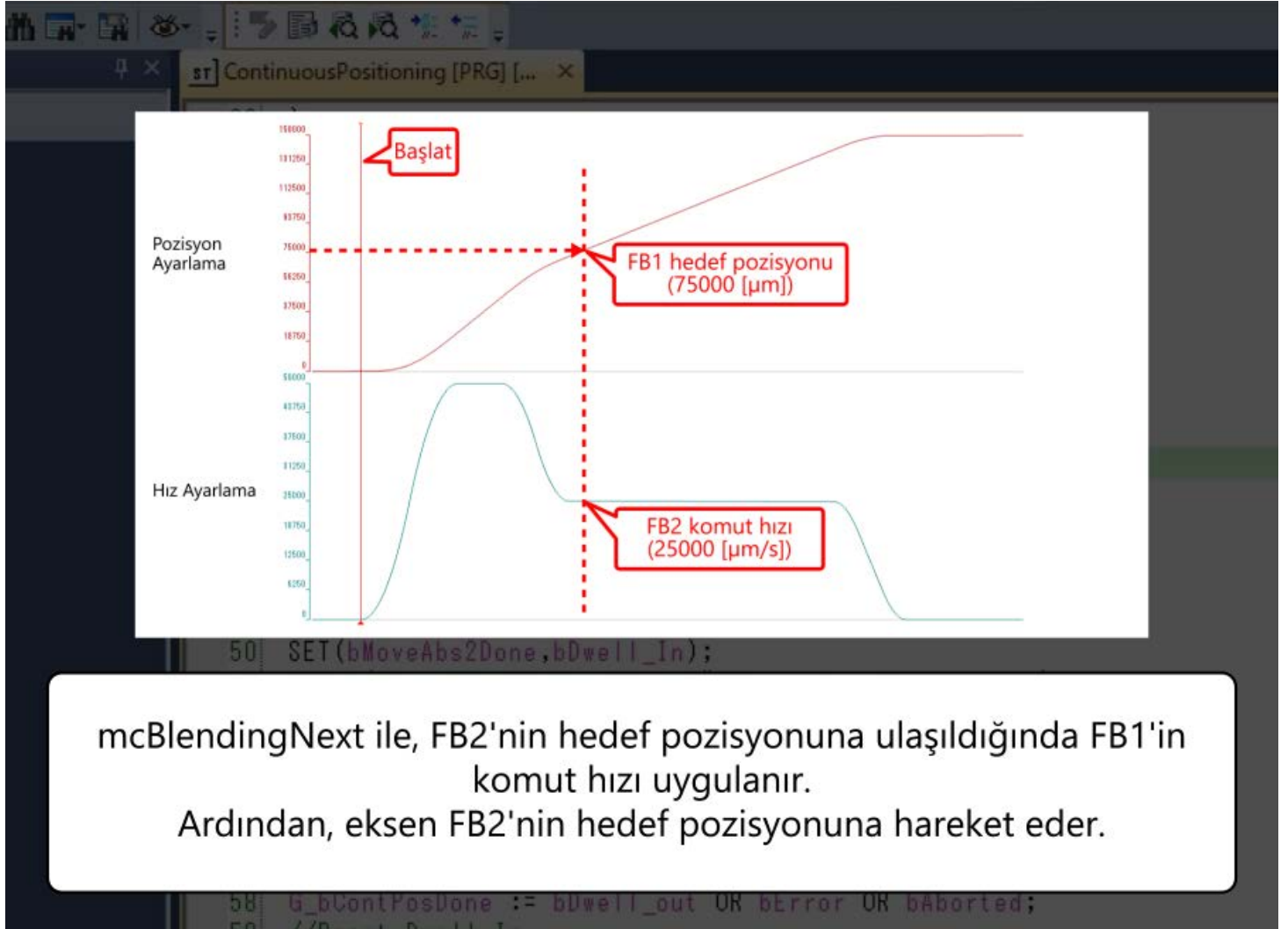
```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36   Axis      := Axis0001.AxisRef ,  
37   Execute   := bMoveAbs1Active ,  
38   Position  := lePosition2 ,  
39   Velocity  := lePosVelocity2 ,  
40   Acceleration:= lePosAcceleration2 ,  
41   Deceleration:= lePosDeceleration2 ,  
42   Jerk      := lePosJerk2 ,  
43   Direction  := MC_DIRECTION_mcShortestWay ,  
44   BufferMode := MC_BUFFER_MODE_mcBlendingNext ,  
45   Done       => bMoveAbs2Done ,  
46   CommandAborted => bMoveAbs2Aborted ,  
47   Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Daha sonra, mcBlendingNext işletimini kontrol edin.
MC_MoveAbsolute_2'nin BufferMode girişini
"MC_BUFFER_MODE_mcBlendingNext" olarak değiştirin, tüm
programları yeniden oluşturun ve Motion modülüne yazın.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```



X25 açıldığında motor çalışır.
Grafikte Pozisyon Ayarlama ve Hız Ayarlama gösterilmektedir.

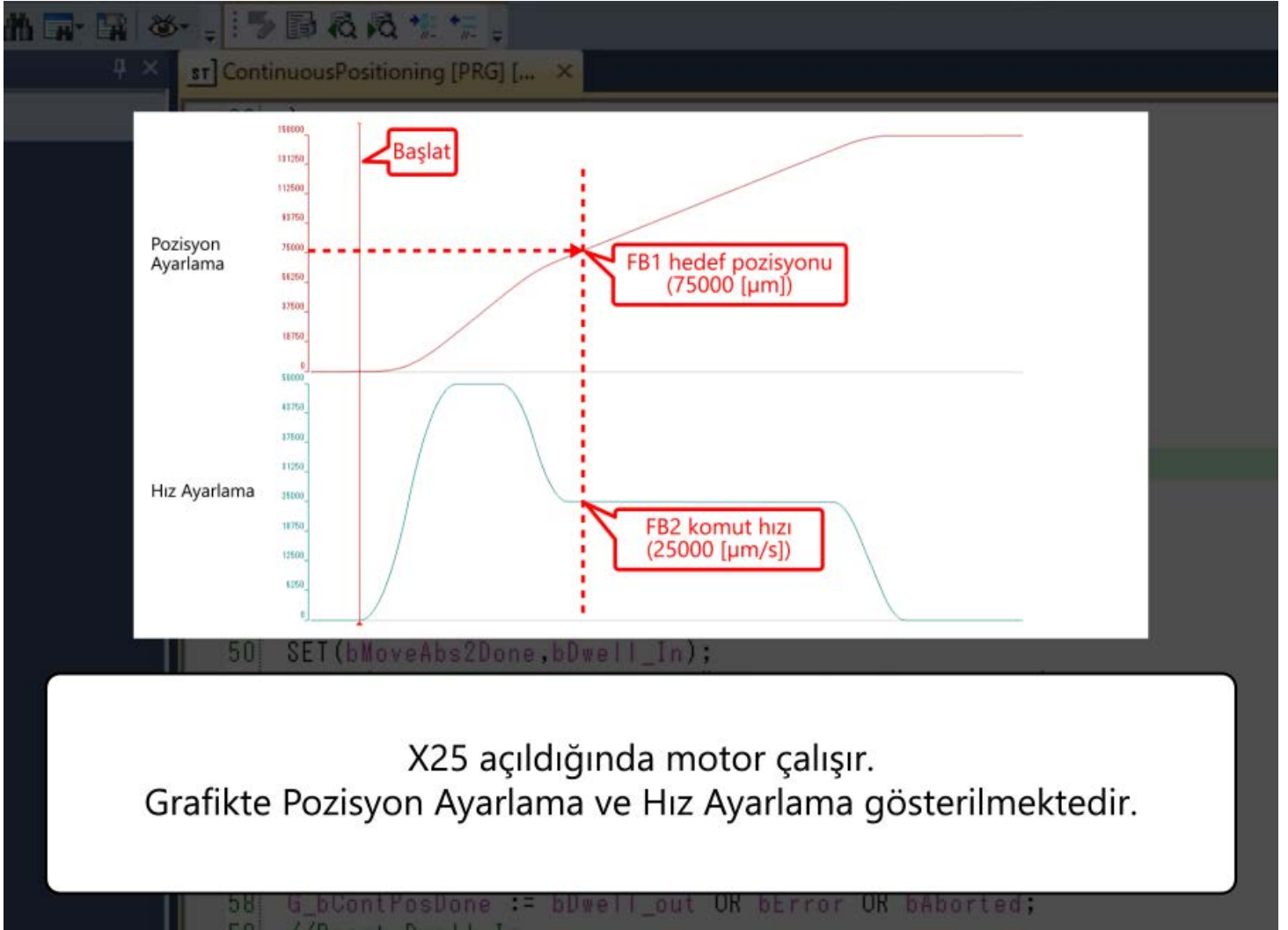


mcBlendingNext ile, FB2'nin hedef pozisyonuna ulaşıldığında FB1'in komut hızı uygulanır. Ardından, eksen FB2'nin hedef pozisyonuna hareket eder.

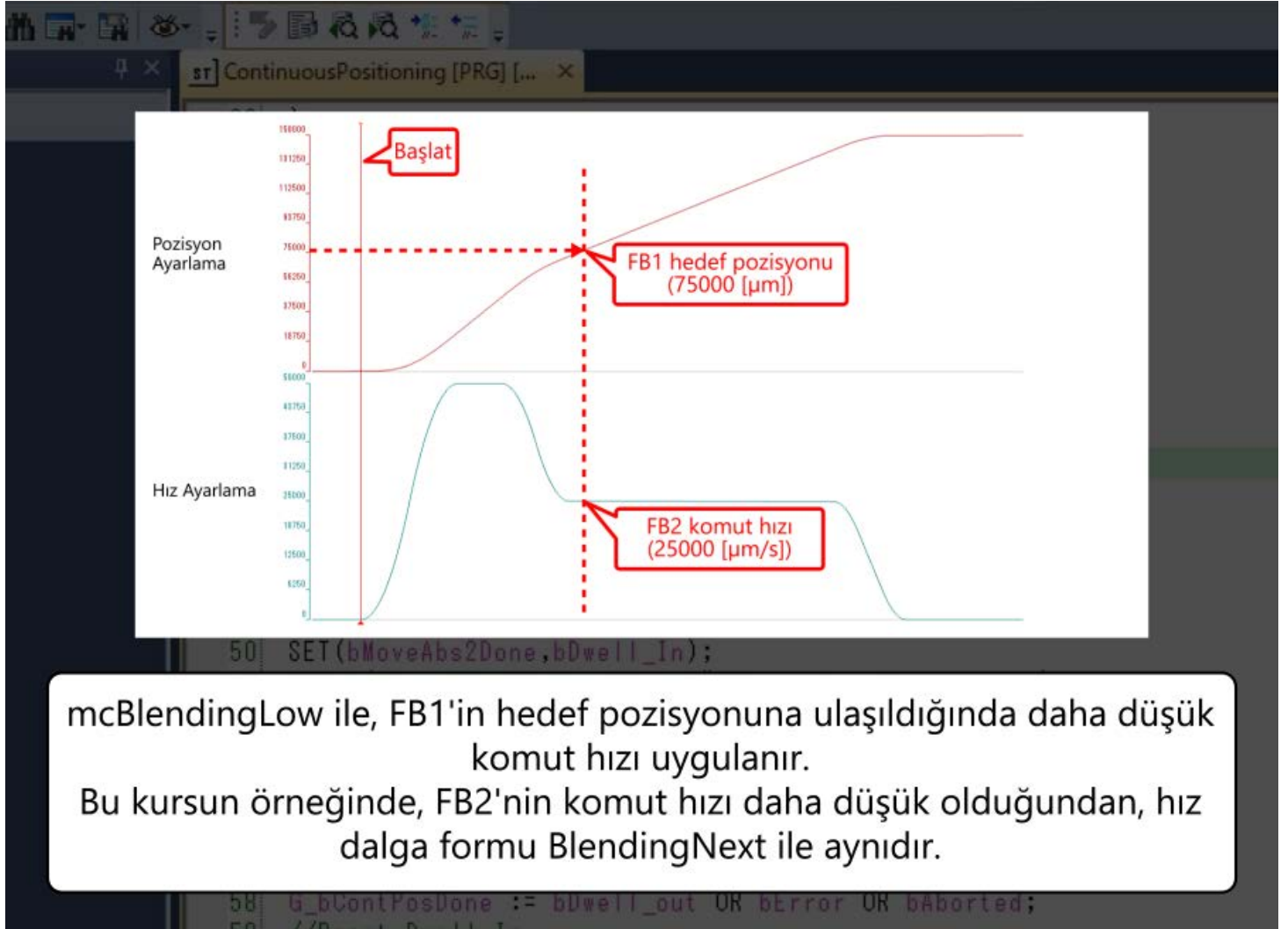
```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE_mcBlendingLow ,  
45     Done      => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error     => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Daha sonra, mcBlendingLow işletimini kontrol edin.
MC_MoveAbsolute_2'nin BufferMode girişini
"MC_BUFFER_MODE_mcBlendingLow" olarak değiştirin, tüm
programları yeniden oluşturun ve Motion modülüne yazın.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```



X25 açıldığında motor çalışır.
Grafikte Pozisyon Ayarlama ve Hız Ayarlama gösterilmektedir.



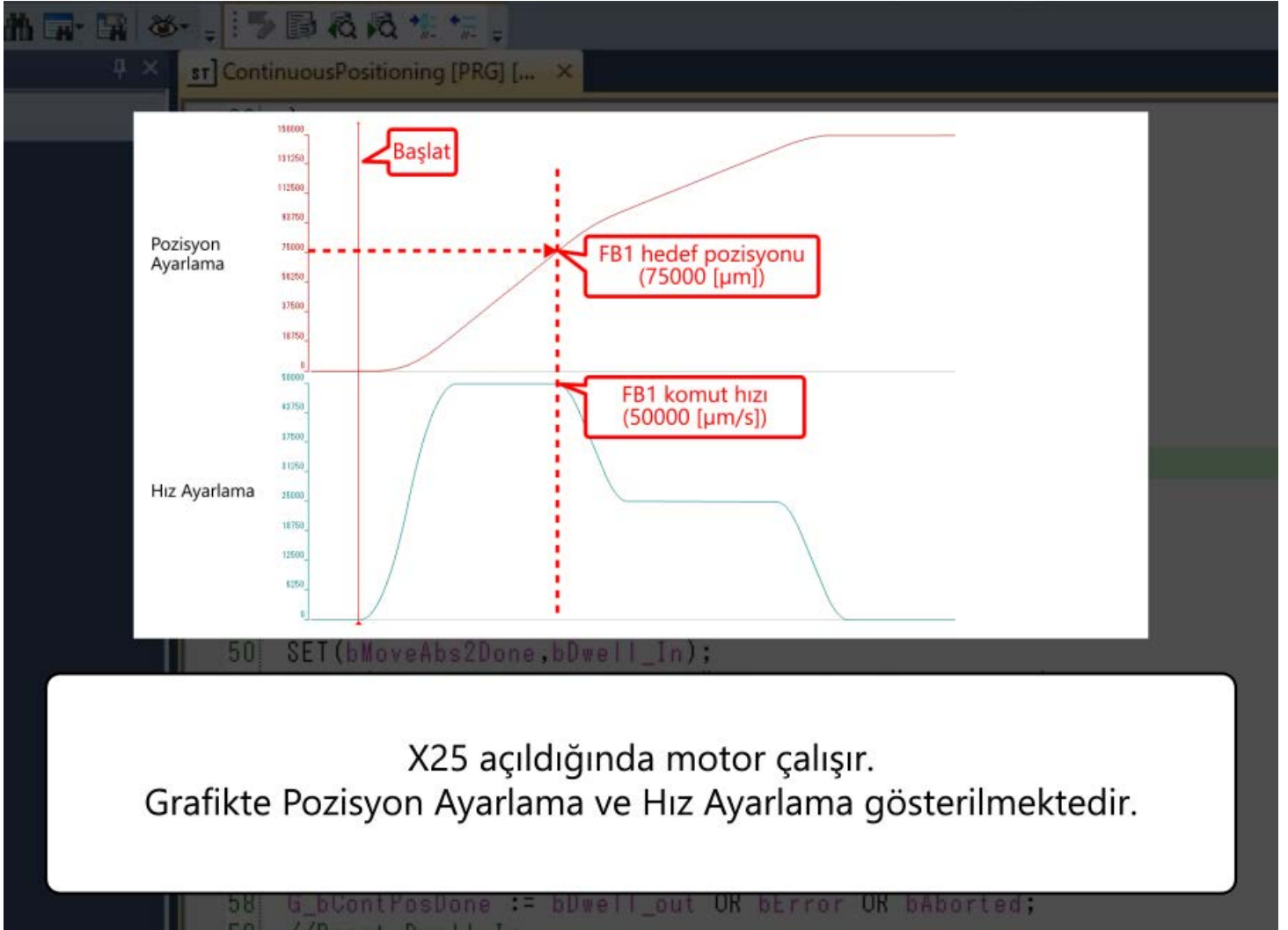
mcBlendingLow ile, FB1'in hedef pozisyonuna ulaşıldığında daha düşük komut hızı uygulanır.

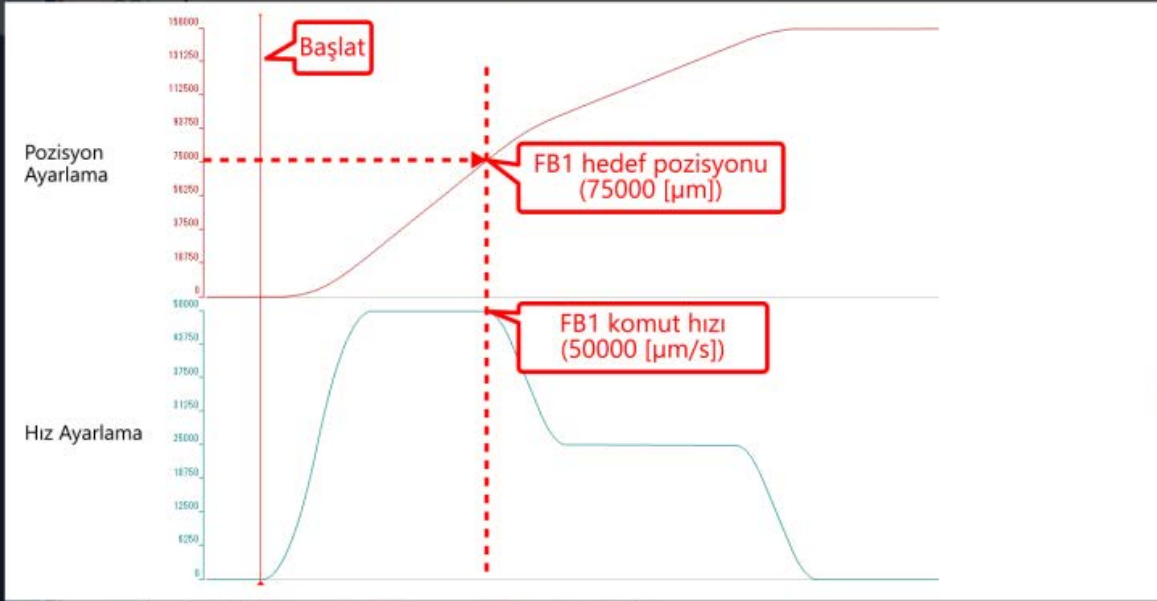
Bu kursun örneğinde, FB2'nin komut hızı daha düşük olduğundan, hız dalga formu BlendingNext ile aynıdır.

```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE_mcBlendingHigh ,  
45     Done      => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error     => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Son olarak, mcBlendingHigh işletimini kontrol edin.
MC_MoveAbsolute_2'nin BufferMode girişini
"MC_BUFFER_MODE_mcBlendingHigh" olarak değiştirin, tüm
programları yeniden oluşturun ve Motion modülüne yazın.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell
```



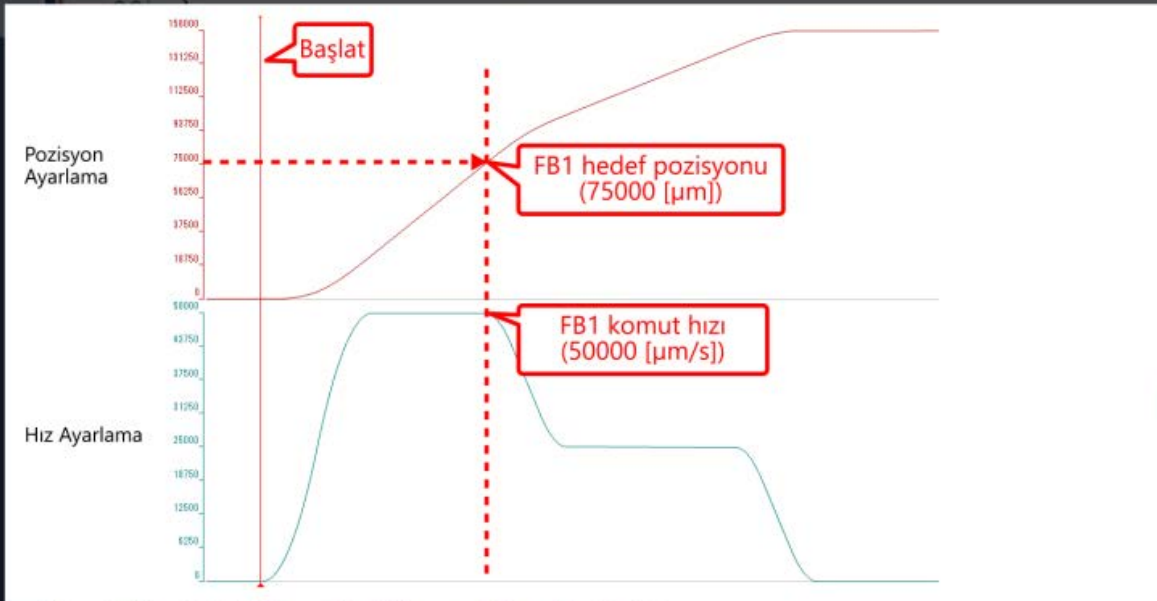


```
50 SET(bMoveAbs2Done,bDwell_In);
```

mcBlendingHigh ile, FB1'in hedef pozisyonuna ulaşıldığında daha yüksek komut hızı uygulanır.
Bu kursun örneğinde, FB1'nin komut hızı daha yüksek olduğundan, hız dalga formu BlendingPrevious ile aynıdır.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```

```
59 /
```



```
50 SET(bMoveAbs2Done,bDwell_In);
```

Bu şekilde BufferMode çalışma kontrolü tamamlanır.
Sonraki sayfaya gidin.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```

Bu bölümde, şunları öğrendiniz:

- İptal etme
- Ara Belleğe Alma
- Karıştırma
- Program Örneği
- Çalışma Kontrolü

Nokta

İptal etme	<ul style="list-style-type: none">• İşlem tipi FB çalışırken ve sonraki işletim tipi FB yürütüldüğünde, İptal Etme işlemi yürütülmekte olan FB'yi kesintiye uğratar ve sonraki FB'yi yürütür.
Ara Belleğe Alma	<ul style="list-style-type: none">• İşlem tipi FB çalışırken ve sonraki işletim tipi FB yürütüldüğünde, Ara Belleğe Alma işlemi yürütülmekte olan FB tamamlanana kadar bekler ve sonraki FB'yi yürütür.
Karıştırma	<ul style="list-style-type: none">• İşlem tipi FB çalışırken ve sonraki işletim tipi FB yürütüldüğünde, Karıştırma işlemi yürütülmekte olan FB'yi durdurmadan sonraki FB'yi yürütür.• Karıştırmada, dört adet hız anahtarlama yöntemi vardır: BlendingLow, BlendingHigh, BlendingPrevious ve BlendingNext.
Program Örneği	<ul style="list-style-type: none">• Ara bellek modunu, işletim FB'sinin BufferMode girişiyle seçin.
Çalışma Kontrolü	<ul style="list-style-type: none">• Her ara bellek modunun çalışmasındaki farkı videoda kontrol ettiniz.

Bu bölümde kullanılacak örnek programı, aşağıdaki bağlantıdan indirin. Program içeriği, bölüm 2 ve bölüm 3'te açıklanan örnek program ile aynıdır. Yalnızca programlama yöntemi farklıdır.

[RD78GBasic2_sample2.zip \(1.39 MB\)](#)

(1) FB kitaplığını indirin

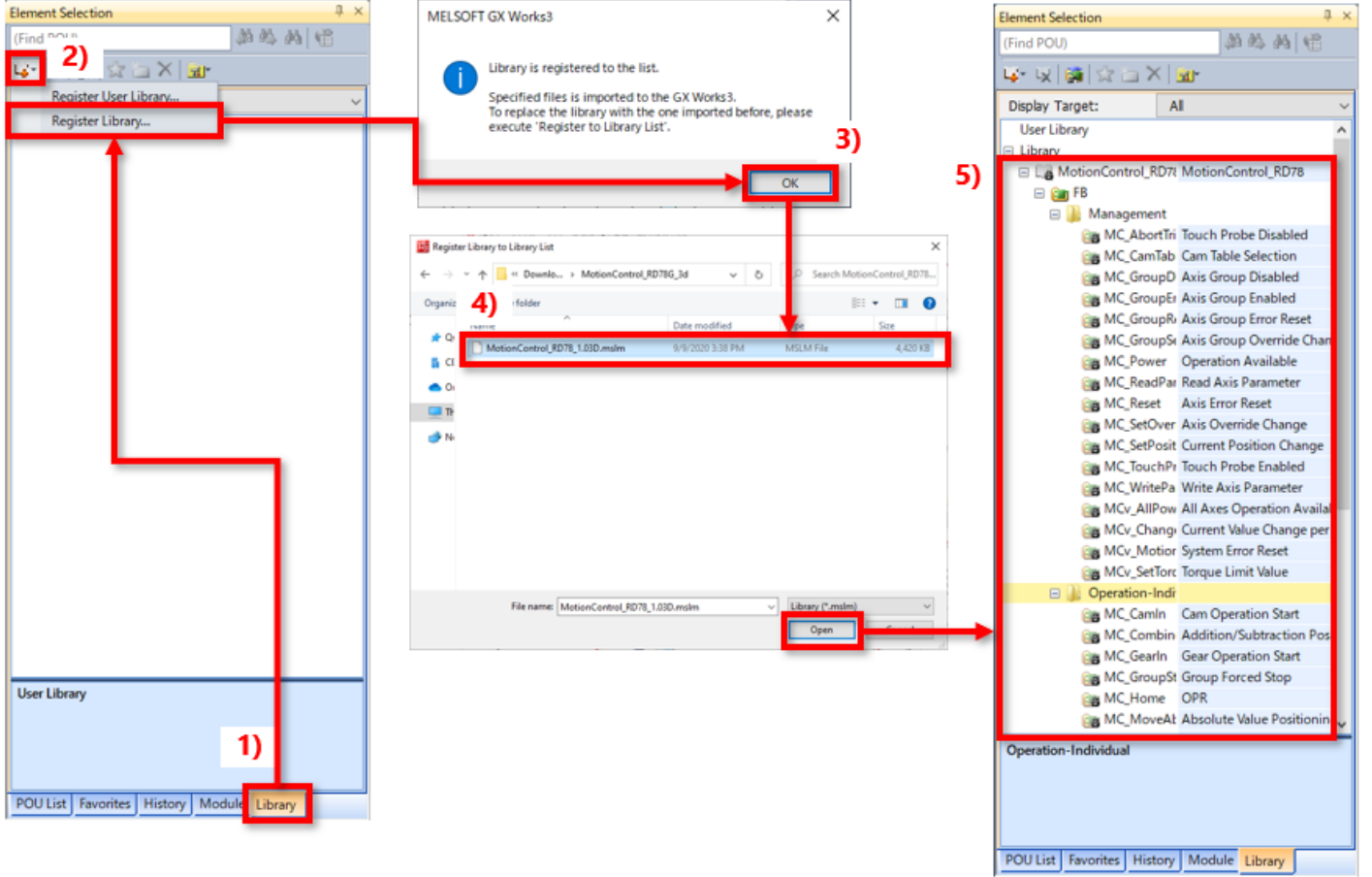
Motion modülü için FB kitaplığı GX Works3'e kaydedilerek Motion control FB, PLC CPU'nun programında kullanılabilir. FB kitaplığını aşağıdaki bağlantıdan indirin ve ZIP dosyasını istediğiniz hedefe çıkartın.

[MotionControl_RD78G_3d.zip\(4.29 MB\)](#)

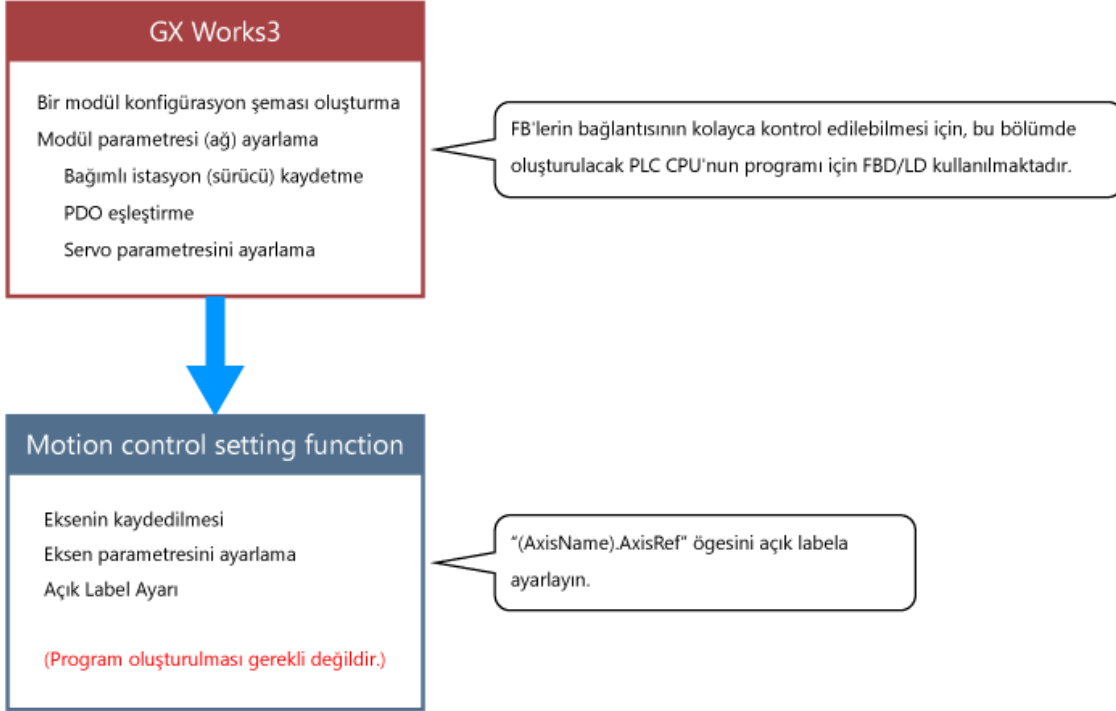
(Not) FB kitaplığının son versiyonu MITSUBISHI ELECTRIC FA Global web sitesinden indirilebilir.

(2) FB kitaplığının kaydedilmesi

- 1) GX Works3'te herhangi bir proje açın ve Element Selection penceresinde Kitaplık sekmesini açın.
- 2) Üst bölümdeki [Register to Library List] düğmesini tıklayın ve [Register Library] seçin.
- 3) "Library is registered to the list" (Kitaplık listeye kaydedildi) mesajı görüntülediğinde, [OK] düğmesini tıklayın.
- 4) FB kitaplık dosyasını "MotionControl_RD78_****.mslm" seçin ve [Open] ögesini tıklayın.
(**** versiyonu belirtir.)
- 5) Motion control FB, Element Selection (Öge Seçme) penceresindeki kitaplığa kaydedilir.



Proje oluřturma iřlemi, nceki kısımda aıklananla aynıdır.



MELSOFT GX Works3 E: Sample.gx3 - [Global [Global Label Setting]]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [...]

Element Selection (Find POU) Display Target: All User Library Library

Label Name	Data Type	English/Display Target	Access from External Device
G.JoetFW	Bit	JOG Forward	
G.JoetBk	Bit	JOG Backward	
G.JoetVelocity	Float (Double Precision)	JOG Velocity	
G.JoetBusy	Bit	JOG Busy	

Oynat düğmesini tıklayın.

Extended Display Do Not Show Always

System label is reserved to be registered. System label is reserved to be released. The system label is already registered to the

To execute the Reservation to Register/Release for the system label, reflection to the system label database is required. Please execute 'Reflect to System Label Database'. It is unnecessary to change reference side project when assigned device is changed in system label Ver.2.
* Only Q-R series/GOT 2000 series is available for system label Ver.2.
* To execute Online Program Change, execute Online Program Change and save.

Reservation to Register System Label
Reservation to Release System Label
Import System Label

Not Reflected: 0
Total: 0

Library POU... Favori... History Mod... Library

Output Progress R04 Host Row 1Column 1 CAP NUM

MELSOFT GX Works3 E: Sample.gx3 - [Global [Global Label Setting]]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [...]

Label Name	Data Type	English (Display Target)	Access from External Device
G.JogFw	Bit	JOG Forward	<input type="checkbox"/>
G.JogBw	Bit	JOG Backward	<input type="checkbox"/>
G.JogVelocity	FLOAT (Double Precision)	JOG Velocity	<input type="checkbox"/>
G.JogBusy	Bit	JOG Busy	<input type="checkbox"/>

Extended Display: Do Not Show Always

System label is reserved to be registered. System label is reserved to be released. The system label is already registered to the

To execute the Reservation to Register/Release for the system label, reflection to the system label database is required. Please execute 'Reflect to System Label Database'. It is unnecessary to change reference side project when assigned device is changed in system label Ver.2.
 * Only Q-R series/GOT 2000 series is available for system label Ver.2.
 * To execute Online Program Change, execute Online Program Change and save.

Reservation to Register System Label
 Reservation to Release System Label
 Import System Label

Not Reflected: 0
 Total: 0

Element Selection (Find POU) Display Target: All User Library Library

Library POU... Favori... History Mod... Library

Output Progress R04 Host Row 1 Column 1 CAP NUM

Bu videoda örnek olarak JOG işletimi için bir FB (MCv_Jog) programının nasıl oluşturulduğu gösterilmektedir.

MELSOFT GX Works3 E: Sample.gx3 - [Global [Global Label Setting]]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [...]

Label Name	Data Type	English/Display Target	Access from External Device
G.JogFW	Bit	JOG Forward	
G.JogBW	Bit	JOG Backward	
G.JogVelocity	FLOAT (Double Precision)	JOG Velocity	
G.JogBusy	Bit	JOG Busy	

Display Target: All

User Library

Library

Extended Display: Do Not Show Always

System label is reserved to be registered. System label is reserved to be released. The system label is already registered to the

To execute the Reservation to Register/Release for the system label, reflection to the system label database is required. Please execute 'Reflect to System Label Database'. It is unnecessary to change reference side project when assigned device is changed in system label Ver.2.
* Only Q-R series/GOT 2000 series is available for system label Ver.2.
* To execute Online Program Change, execute Online Program Change and save.

Reservation to Register System Label

Reservation to Release System Label

Import System Label

Not Reflected: 0 Total: 0

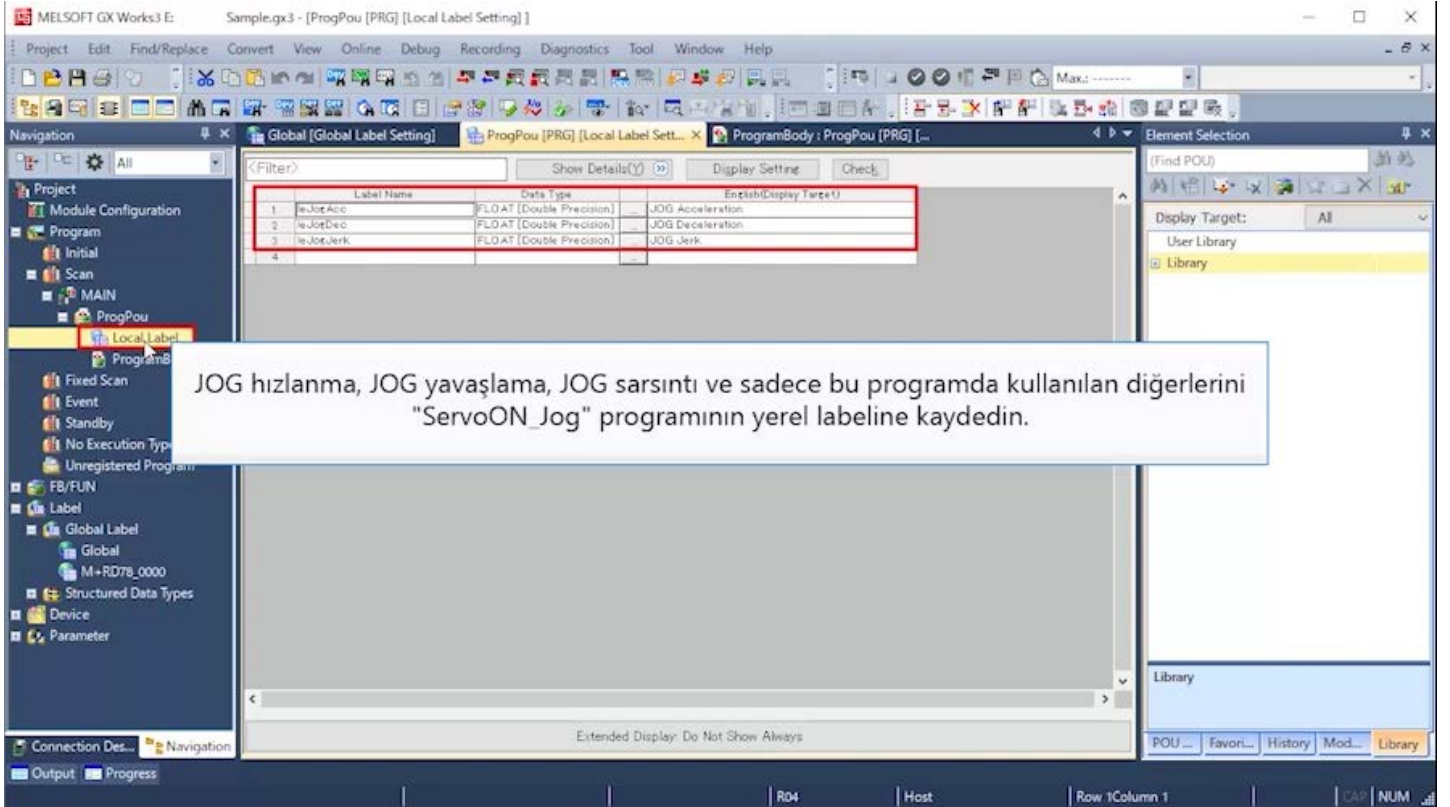
Library

POU... Favori... History Mod... Library

Output Progress

R04 Host Row 1 Column 1 CAP NUM

JOG komutu, JOG hızı ve Jog meşgul'ü genel labela kaydedin. (JOG hızının GOT gibi harici cihazlardan ayarlanabileceği varsayılarak, JOG hızı genel labela kaydedilir.)



JOG hızlanma, JOG yavaşlama, JOG sarsıntı ve sadece bu programda kullanılan diğerlerini "ServoON_Jog" programının yerel labeline kaydedin.

Label Name	Data Type	English (Display Target)
vJogAcc	FLOAT [Double Precision]	JOG Acceleration
vJogDec	FLOAT [Double Precision]	JOG Deceleration
vJogJerk	FLOAT [Double Precision]	JOG Jerk

The screenshot displays the MELSOFT GX Works3 E software interface. The main window shows a ladder logic editor with a grid. On the left, the 'Navigation' pane is open, showing a tree view of the project structure. The 'ProgramBody' folder is selected. On the right, the 'Element Selection' panel is open, displaying a list of instructions categorized into SEQUENCE INSTRUCTIONS, BASIC INSTRUCTIONS, and APPLICATION INSTRUCTIONS. A callout box with a white background and a black border points to the 'Library' button in the bottom right corner of the software interface. The callout box contains the text: [Library] ögesini tıklayın.

Navigation

- Project
- Module Configuration
- Program
 - Initial
 - Scan
 - MAIN
 - ProgPou
 - Local Label
 - ProgramBody
 - Fixed Scan
 - Event
 - Standby
 - No Execution Type
 - Unregistered Program
 - FB/FUN
 - Label
 - Global Label
 - Global
 - M+RD78_0000
 - Structured Data Types
 - Device
 - Parameter

Element Selection

(Find POU)

Display Target: All

SEQUENCE INSTRUCTIONS

- Contact instructions
- Association instructions
- Output instructions
- Shift instructions
- Master Control instructions
- Stop instruction

BASIC INSTRUCTIONS

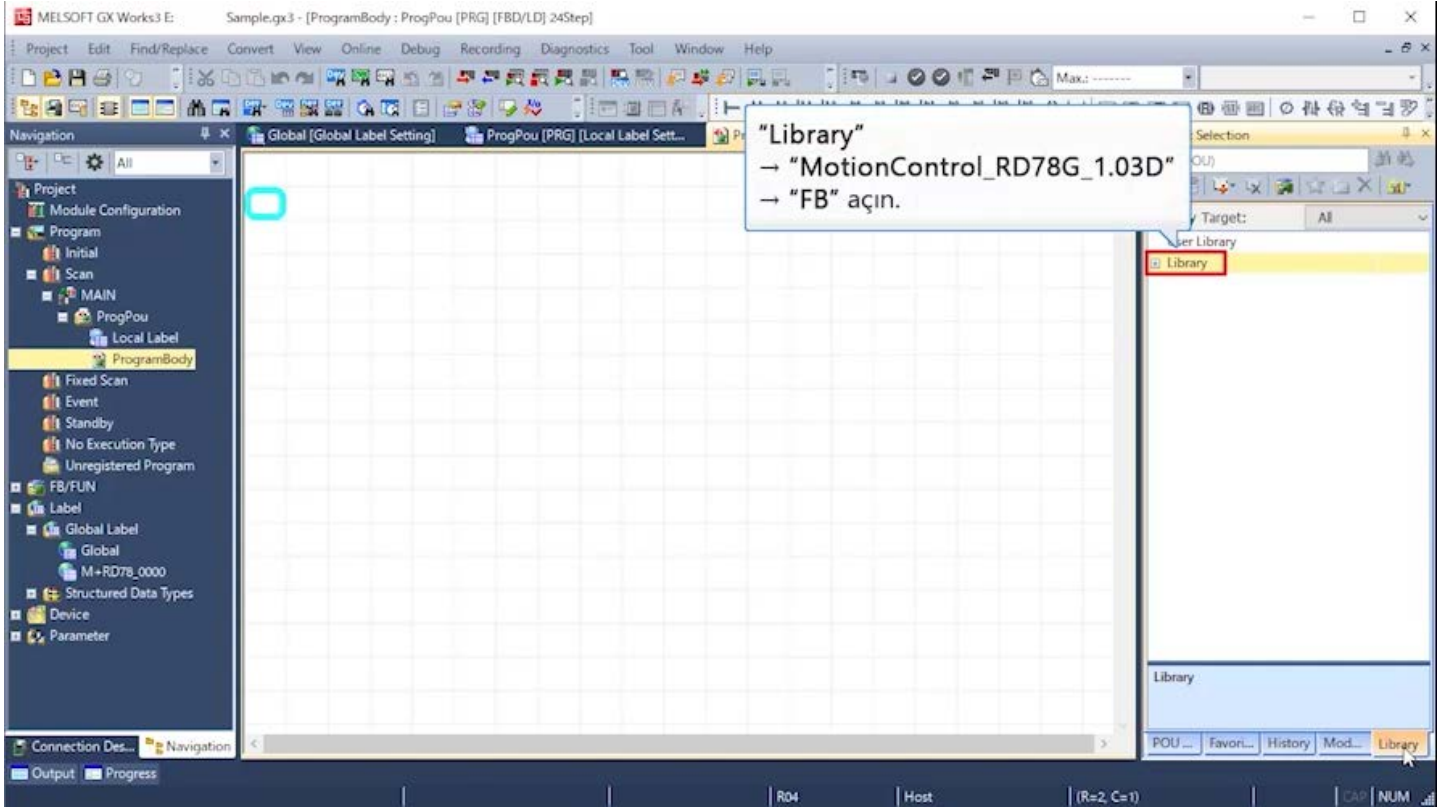
- Comparison Operation instructions
- Arithmetic Operation instructions
- Data transfer instructions
- Logical Operation instructions
- Data shift instructions
- Bit processing instructions
- Data Conversion instructions

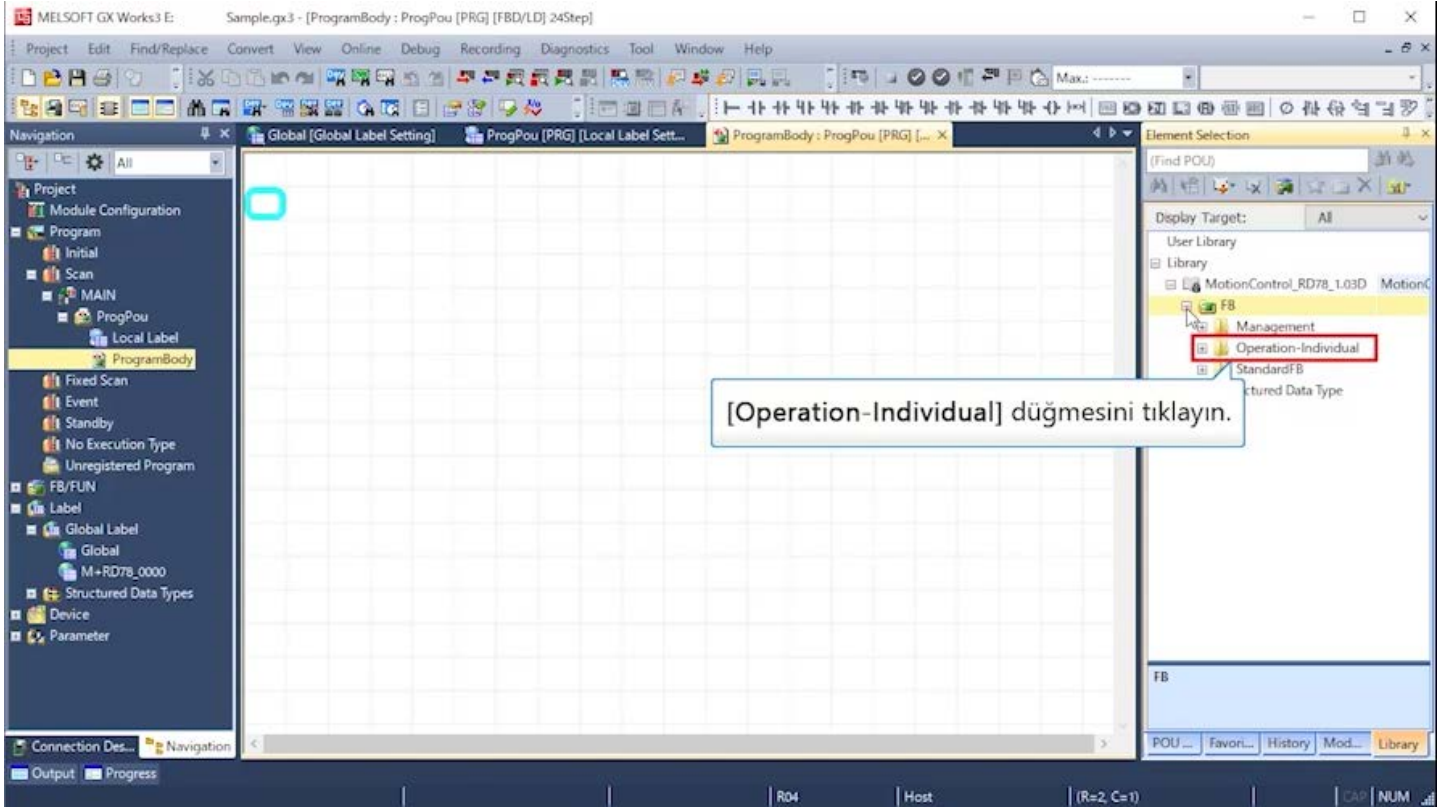
APPLICATION INSTRUCTIONS

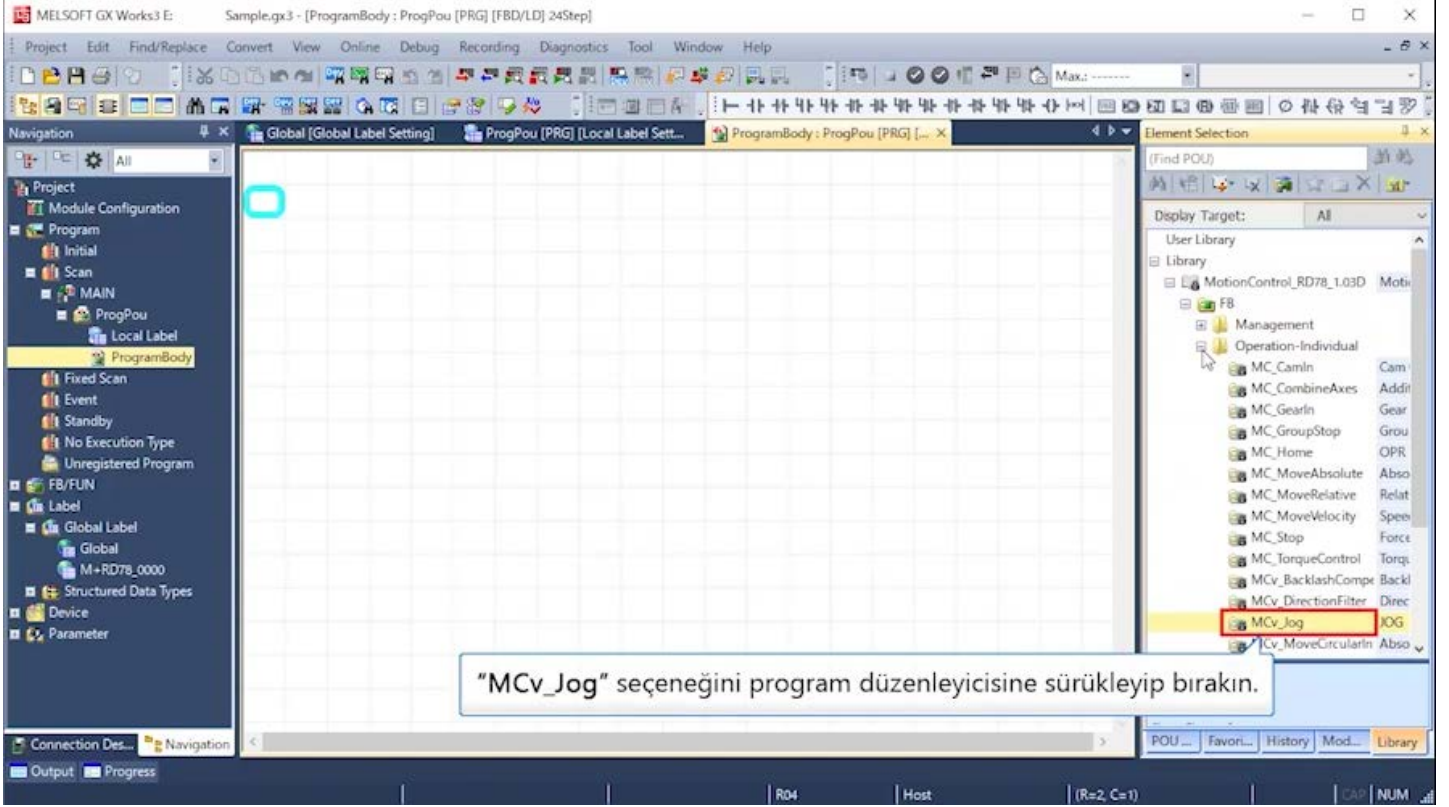
- Program execution control instructions
- Rotation instructions
- Data processing instructions

[Library] ögesini tıklayın.

POU... Favori... History Mod... Library







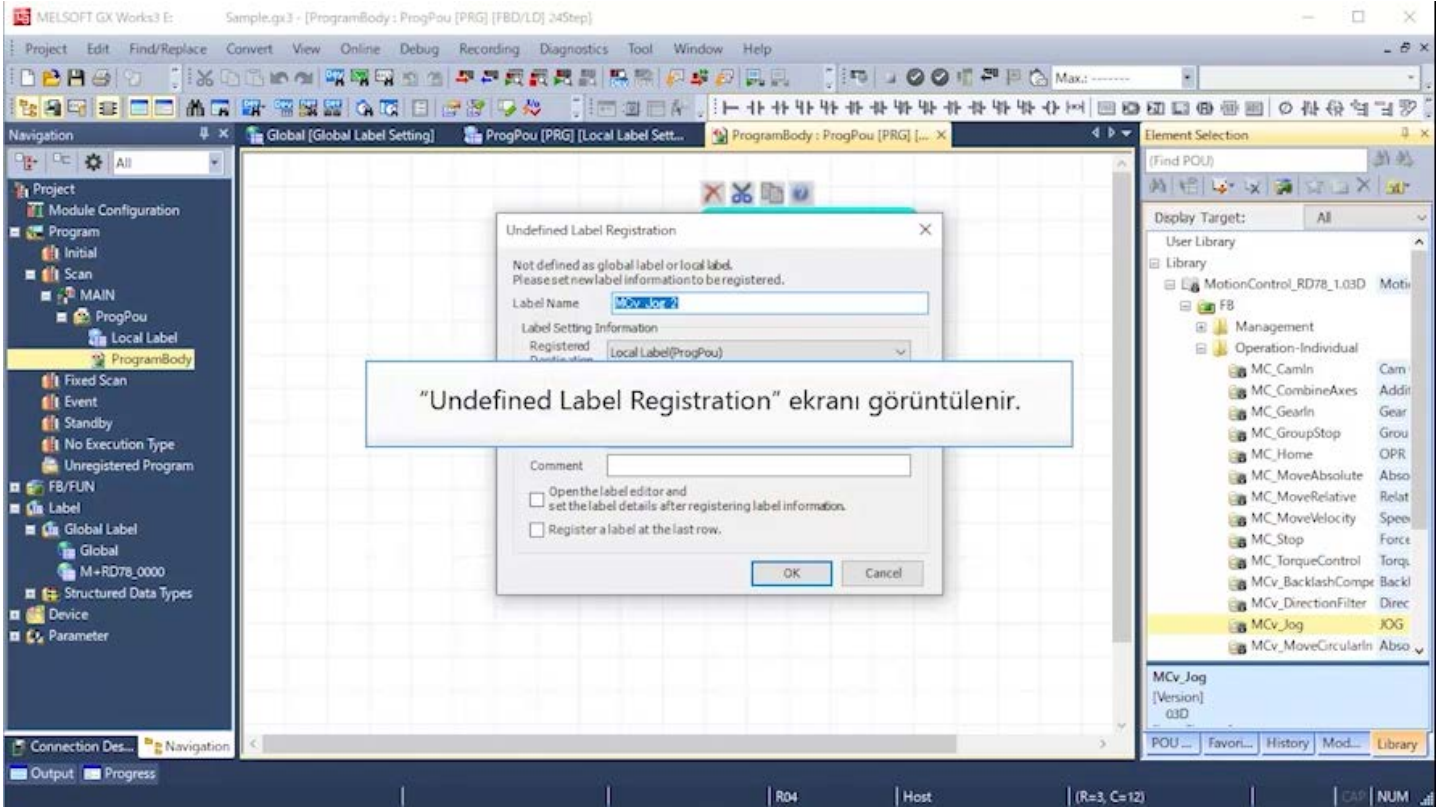
The screenshot displays the MELSOFT GX Works3 software interface. The main workspace is a grid. On the left, the 'Navigation' pane shows the project structure, with 'ProgramBody' selected. On the right, the 'Element Selection' window is open, showing a library of motion control functions. The 'MCv_Jog' function is highlighted with a red box. A text box at the bottom of the grid contains the instruction: "MCv_Jog" seçeneğini program düzenleyicisine sürükleyip bırakın.

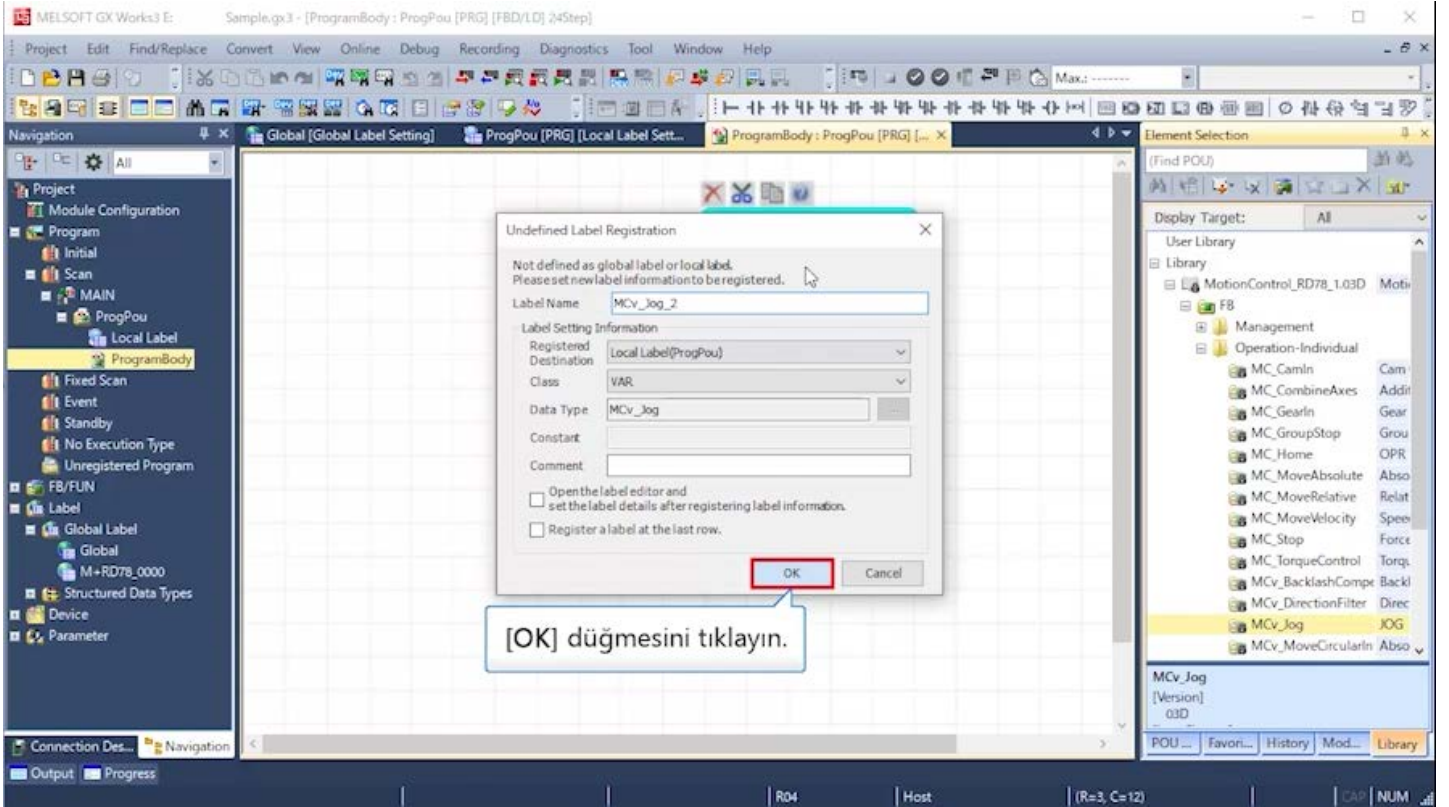
Navigation: Project, Module Configuration, Program, Initial, Scan, MAIN, ProgPou, Local Label, ProgramBody, Fixed Scan, Event, Standby, No Execution Type, Unregistered Program, FB/FUN, Label, Global Label, Global, M+RD78_0000, Structured Data Types, Device, Parameter.

Element Selection: (Find POU), Display Target: All, User Library, Library, MotionControl_RD78_1.03D, FB, Management, Operation-Individual, MC_CamIn, MC_CombineAxes, MC_GearIn, MC_GroupStop, MC_Home, MC_MoveAbsolute, MC_MoveRelative, MC_MoveVelocity, MC_Stop, MC_TorqueControl, MCv_BacklashComp, MCv_DirectionFilter, MCv_Jog, Cvv_MoveCircularIn.

POU... Favori... History Mod... Library

R04 Host (R=2, C=1) CAP NUM





The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with a grid. A red square indicates the target location for placing the block. A callout box with the text "LD ögesinin yerleştirileceği alana tıklayın." (Click on the area where the LD element will be placed.) points to this square. A blue-bordered block labeled "MCv_Jog_2" is being dragged towards the target. The block contains the following parameters:

Parameter	Value
JogForward	Done
JogBackward	Busy
Velocity	Active
Acceleration	CommandAborted
Deceleration	Error
Jerk	ErrorID
Options	
Axis	

The right-hand side of the interface shows the "Element Selection" panel with a tree view of the "User Library". The "MotionControl_RD78_1.03D" folder is expanded, showing various motion control blocks. The "MCv_Jog" block is selected, and its details are shown at the bottom of the panel.

MCv_Jog
[Version]
03D

POU... Favori... History Mod... Library

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a grid with a blue square symbol (LD) being placed. A callout box with the text "LD ögesinin simgesine tıklayın." (Click on the LD symbol) points to this symbol. The element selection pane on the right shows a library of Motion Control FB blocks, with "MCv_Jog" selected. The navigation pane on the left shows the project structure, including "ProgramBody" and "Local Label".

LD ögesinin simgesine tıklayın.

MCv_Jog_2

JogForward	Done
JogBackward	Busy
Velocity	Active
Acceleration	CommandAborted
Deceleration	Error
Jerk	ErrorID
Options	
Axis	

MCv_Jog
[Version]
03D

POU... Favori... History Mod... Library

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with a block labeled "MCv_Jog.2" (MDv_Jog). A callout box points to the LD (Load) input of the block with the text "Belirtilen LD ögesi yerleştirilir." (The specified LD element is placed). The right panel shows the User Library with the MCv_Jog block selected. The bottom status bar shows "R04 Host (R=3, C=4) CAP NUM".

Navigation: Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody: ProgPou [PRG] [...]

Element Selection: (Find POU) Display Target: All User Library Library MotionControl_RD78_1.03D Moti FB Management Operation-Individual MC_CamIn Cam MC_CombineAxes Addri MC_GearIn Gear MC_GroupStop Grou MC_Home OPR MC_MoveAbsolute Abso MC_MoveRelative Relat MC_MoveVelocity Speec MC_Stop Forcc MC_TorqueControl Torq MCv_BacklashComp Backl MCv_DirectionFilter Direc MCv_Jog JOG MCv_MoveCircularIn Abso

MCv_Jog [Version] 03D POU... Favori... History Mod... Library

R04 Host (R=3, C=4) CAP NUM

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with a function block labeled "MCv_Jog_2" being inserted. A callout box with the text "[???' ögesine çift tıklayın." (Double-click on the '???' element) is positioned over the function block. The software interface includes a menu bar, a toolbar, a navigation pane on the left, and an element selection pane on the right.

The navigation pane on the left shows the following structure:

- Project
- Module Configuration
- Program
- Initial
- Scan
- Event
- Standby
- No Execution Type
- Unregistered Program
- FB/FUN
- Label
- Global Label
- M+RD78_0000
- Structured Data Types
- Device
- Parameter

The element selection pane on the right shows the following structure:

- Library
- MotionControl_RD78_1.03D
- FB
- Management
- Operation-Individual
- MC_CamIn
- MC_CombineAxes
- MC_GearIn
- MC_GroupStop
- MC_Home
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_Stop
- MC_TorqueControl
- MCv_BacklashComp
- MCv_DirectionFilter
- MCv_Jog
- MCv_MoveCircularIn

The status bar at the bottom shows the following information:

- R04
- Host
- (R=3, C=8)
- CAP
- NUM

MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

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

Navigation Global [Global Label Setting] Bit cihazının label adını girin. ProgPou [PRG] Element Selection

Display Target: All

User Library

Library

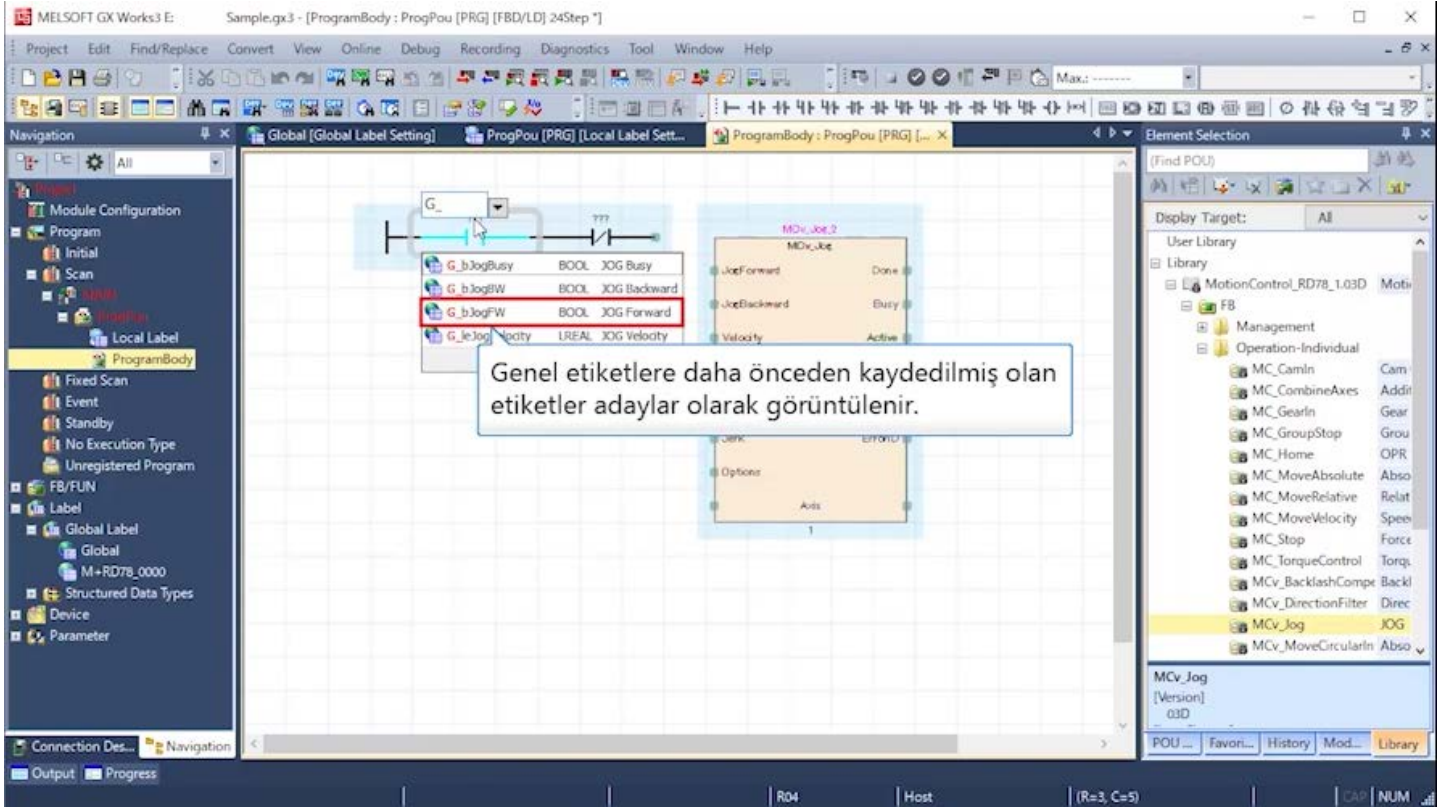
- MotionControl_RD78_1.03D Moti
 - FB
 - Management
 - Operation-Individual
 - MC_CamIn Cam
 - MC_CombineAxes Addi
 - MC_GearIn Gear
 - MC_GroupStop Grou
 - MC_Home OPR
 - MC_MoveAbsolute Abso
 - MC_MoveRelative Relat
 - MC_MoveVelocity Spee
 - MC_Stop Forcs
 - MC_TorqueControl Torq
 - MCv_BacklashComp Backl
 - MCv_DirectionFilter Direc
 - MCv_Jog JOG
 - MCv_MoveCircularIn Abso

MCv_Jog [Version] 03D

POU ... Favori... History Mod... Library

R04 Host (R=3, C=5) CAP NUM

G_bJogBusy	BOOL	JOG Busy
G_bJogBW	BOOL	JOG Backward
G_bJogFW	BOOL	JOG Forward
G_leJogVelocity	LREAL	JOG Velocity



The screenshot displays the MELSOFT GX Works3 E interface. The main workspace shows a ladder logic diagram with a network containing a normally open contact labeled 'G_bJogFW' and a coil labeled 'MDV_Jog_2'. A callout box points to the 'G_bJogFW' label in the library, stating: "Genel etiketlere daha önceden kaydedilmiş olan etiketler adaylar olarak görüntülenir." (Labels already registered in general labels are displayed as candidates).

The library on the right shows the 'MotionControl_RD78_1.03D' folder, which contains various function blocks. The 'MCv_Jog' block is highlighted, showing its parameters: [Version] 03D.

Label	Type	Description
G_bJogBusy	BOOL	JOG Busy
G_bJogBW	BOOL	JOG Backward
G_bJogFW	BOOL	JOG Forward
G_leJog_Velocity	LREAL	JOG Velocity

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with a normally open contact labeled 'G_bJogFW' and a coil labeled 'G_bJogBW'. A red box highlights the coil, and a callout box points to it with the text: "LD ögesini FB'nin JogForward girişine bağlayın." (Connect the LD element to the JogForward input of the FB).

The function block library on the right side of the screen shows the 'MCv_Jog' block. The inputs and outputs are listed as follows:

Input	Output
JogForward	Done
JogBackward	Busy
Velocity	Active
Acceleration	CommandAborted
Deceleration	Error
Jerk	ErrorID
Options	
Axis	

The 'MCv_Jog' block is highlighted in the library, and its details are shown in the bottom right corner:

MCv_Jog
[Version]
03D

MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

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

Navigation Global [Global]

LD ögesi JogForward girişine ve JogBackward girişine bağlandığında bunun gibi görünebilir.

Element Selection (Find POU)

Display Target: All

User Library

Library

- MotionControl_RD78_1.03D Moti
 - FB
 - Management
 - Operation-Individual
 - MC_CamIn Cam
 - MC_CombineAxes Addit
 - MC_GearIn Gear
 - MC_GroupStop Grou
 - MC_Home OPR
 - MC_MoveAbsolute Abso
 - MC_MoveRelative Relat
 - MC_MoveVelocity Spees
 - MC_Stop Forcs
 - MC_TorqueControl Torq
 - MCV_BacklashComp Backl
 - MCV_DirectionFilter Direc
 - MCV_Jog JOG
 - MCV_MoveCircularIn Abso

MCV_Jog [Version] 03D

POU... Favori... History Mod... Library

R04 Host (R=5, C=10) CAP NUM

MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [...]

Element Selection (Find POU)

Display Target: All

User Library

Library

- MotionControl_RD78_1.03D Moti
- FB
- Management
- Operation-Individual
 - MC_CamIn Cam
 - MC_CombineAxes Addi
 - MC_GearIn Gear
 - MC_GroupStop Grou
 - MC_Home OPR
 - MC_MoveAbsolute Abso
 - MC_MoveRelative Relat
 - MC_MoveVelocity Spee
 - MC_Stop Forcs
 - MC_TorqueControl Torq
 - MCv_BacklashComp Backl
 - MCv_DirectionFilter Direc
 - MCv_Jog JOG
 - MCv_MoveCircularIn Abso

MCv_Jog [Version] 03D

POU... Favori... History Mod... Library

R04 Host (R=5, C=10) CAP NUM

FBD ögesinin yerleştirileceği alana tıklayın.

The screenshot displays the MELSOFT GX Works3 E software interface. The main workspace shows a ladder logic diagram with a function block call. A callout box points to the 'G_LeJogVelocity' parameter, with the text: "G_LeJogVelocity" genel etiketini girin.

The function block is 'MCv_Jog' (JOG Velocity). Its parameters are:

- JobBackward: Boolean
- Busy: Boolean
- Active: Boolean
- CommandAborted: Boolean
- Error: Boolean
- ErrorID: Integer
- Deceleration: Real
- Jerk: Real
- Option: Integer
- Axis: Integer

The diagram shows two normally open contacts labeled 'G_LeJogPW' connected to the 'G_LeJogVelocity' parameter. The function block is labeled 'MCv_Jog' and has a 'Setting...' button. The status bar at the bottom indicates 'R04 Host (R=6, C=8)'. The bottom right corner shows 'CAP NUM'.

Genel labellere daha önceden kaydedilmiş olan labellar adaylar olarak görüntülenir.

MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [... x] Element Selection

(Find POU)

Display Target: All

FBD ögesini FB'nin Hız girişine bağlayın.

MDV_Jog_2

MDV_Jog

JogForward Done

JogBackward Busy

Velocity Active

Acceleration CommandAborted

Deceleration Error

Jerk ErrorID

Options

Axis

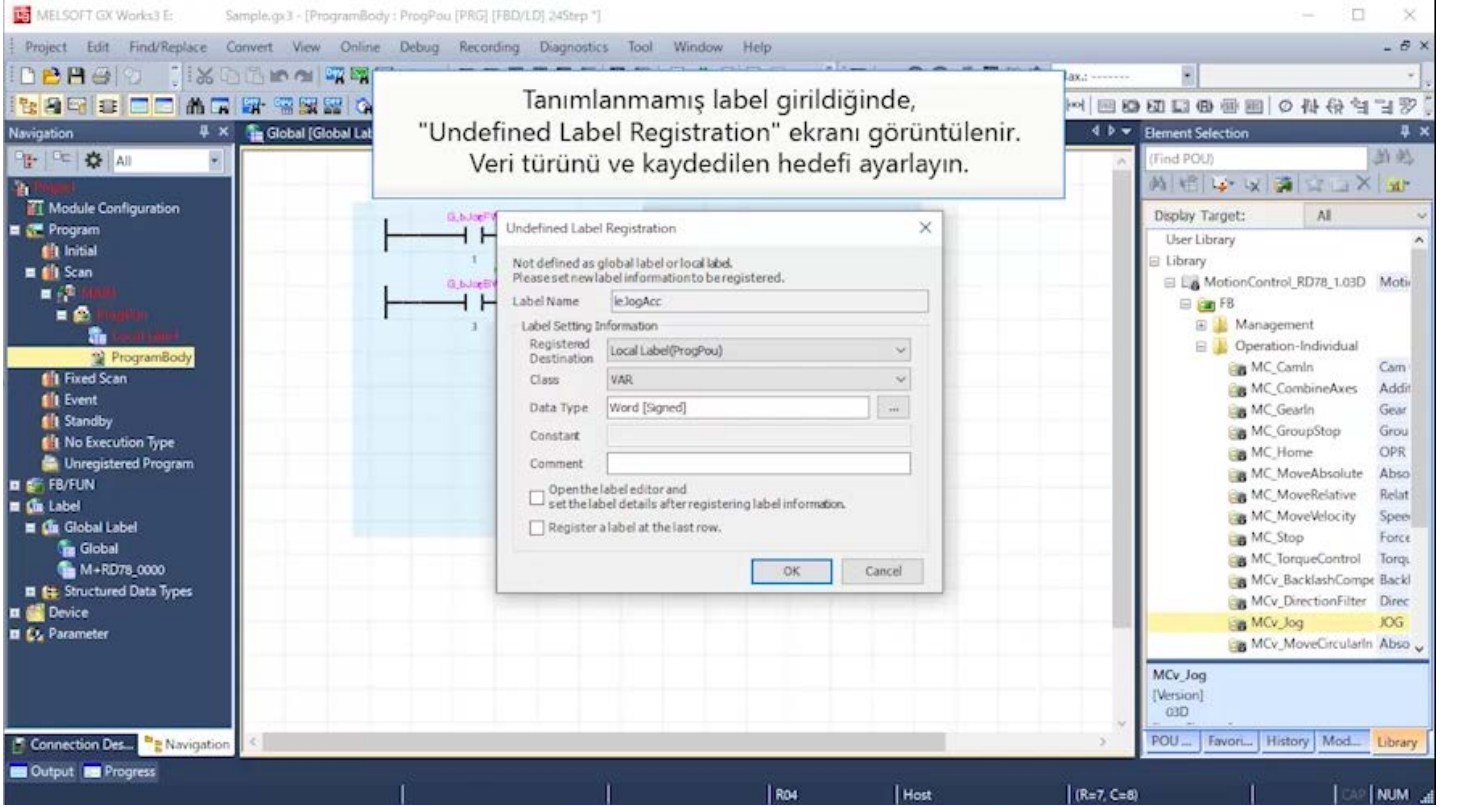
5

MCv_Jog [Version] 03D

POU ... Favori... History Mod... Library

R04 Host (R=6, C=8) CAP NUM

Tanımlanmamış label girildiğinde, "Undefined Label Registration" ekranı görüntülenir. Veri türünü ve kaydedilen hedefi ayarlayın.



The screenshot displays the MELSOFT GX Works3 E software interface. The main window shows a ladder logic diagram with a label 'G_b.JogAcc' highlighted. A dialog box titled 'Undefined Label Registration' is open, prompting the user to register the label. The dialog box contains the following information:

- Label Name: JogAcc
- Registered Destination: Local Label(ProgPou)
- Class: VAR
- Data Type: Word [Signed]
- Constant: (empty)
- Comment: (empty)
- Open the label editor and set the label details after registering label information. (checkbox)
- Register a label at the last row. (checkbox)

The background shows a ladder logic diagram with a label 'G_b.JogAcc' and a library tree on the right side of the screen. The library tree includes the following items:

- Library
- MotionControl_RD78_1.03D
- FB
- Management
- Operation-Individual
- MC_CamIn
- MC_CombineAxes
- MC_GearIn
- MC_GroupStop
- MC_Home
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_Stop
- MC_TorqueControl
- MCV_BacklashComp
- MCV_DirectionFilter
- MCV_Jog
- MCV_MoveCircularIn

MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [...]

Element Selection (Find POU)

Display Target: All

User Library

Library

- MotionControl_RD78_1.03D Moti...
- FB
- Management
- Operation-Individual
 - MC_CamIn Cam
 - MC_CombineAxes Addit...
 - MC_GearIn Gear
 - MC_GroupStop Grou
 - MC_Home OPR
 - MC_MoveAbsolute Abso
 - MC_MoveRelative Relat
 - MC_MoveVelocity Spee
 - MC_Stop Forcc
 - MC_TorqueControl Torq
 - MCv_BacklashComp Backl
 - MCv_DirectionFilter Direc
 - MCv_Log JOG

Connection Des... Navigation

Output Progress

R04 Host (R=5, C=16) CAP NUM

[Module] düğmesini tıklayın.

MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett...]

Module Configuration
Program
Initial
Scan
Motion Control
Motion Control Label
ProgramBody
Fixed Scan
Event
Standby
No Execution Type
Unregistered Program
FB/FUN
Label
Global Label
Global
M+RD78_0000
Structured Data Types
Device
Parameter

Global Label Setting

Global Label
M+RD78_0000

Module Label
Module FB

Module Label

POU ... Favori... History Module Library

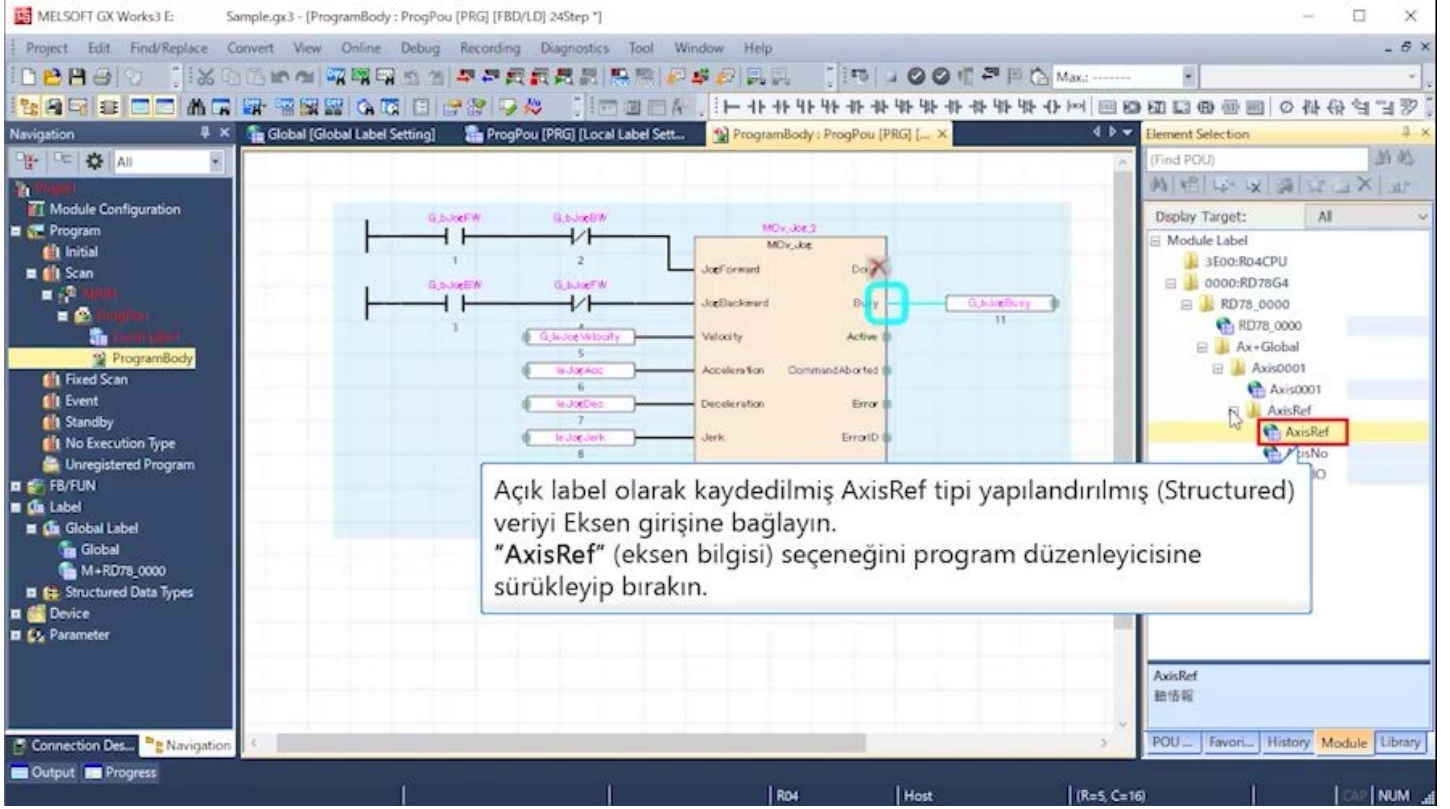
“Module Label”
→ “0000:RD78G4” → “RD78_0000”
→ “Ax+Global” → “Axis0001”
→ “AxisRef” açın.

MCv_Ax2
MCv_Ax2
JocForward
JocBackward
Velocity
Acceleration
Deceleration
Jerk
Options
Active
CommandAborted
Error
ErrorID
Axis

1
2
3
4
5
6
7
8
9
10

11

RD4 Host (R=5, C=16) CAP NUM



Açık label olarak kaydedilmiş AxisRef tipi yapılandırılmış (Structured) veriyi Eksen girişine bağlayın.
"AxisRef" (eksen bilgisi) seçeneğini program düzenleyicisine sürükleyip bırakın.

MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

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

Navigation Global [Global Label Setting] ProgPou [PRG] [Local Label Sett... ProgramBody : ProgPou [PRG] [... X]

Element Selection (Find POU)

Display Target: All

Module Label

- 3E00-R04CPU
- 0000-RD78G4
- RD78_0000
- Ax-Global
- Axis0001
- AxisRef
- AxisNo
- StartIO
- Md
- Module FB

MDV_Job2
MDV_kie

JobForward Done

JobBackward Busy

Velocity Active

Acceleration CommandAborted

Deceleration Error

Jerk ErrorID

Options

Axis

1 G_LoJobFW

2 G_LoJobBW

3 G_LoJobFW

4 G_LoJobVelocity

5 W_JobAcc

6 W_JobDec

7 W_JobJerk

8

9

10 RD78_0000 Axis0001 AxisRef

11

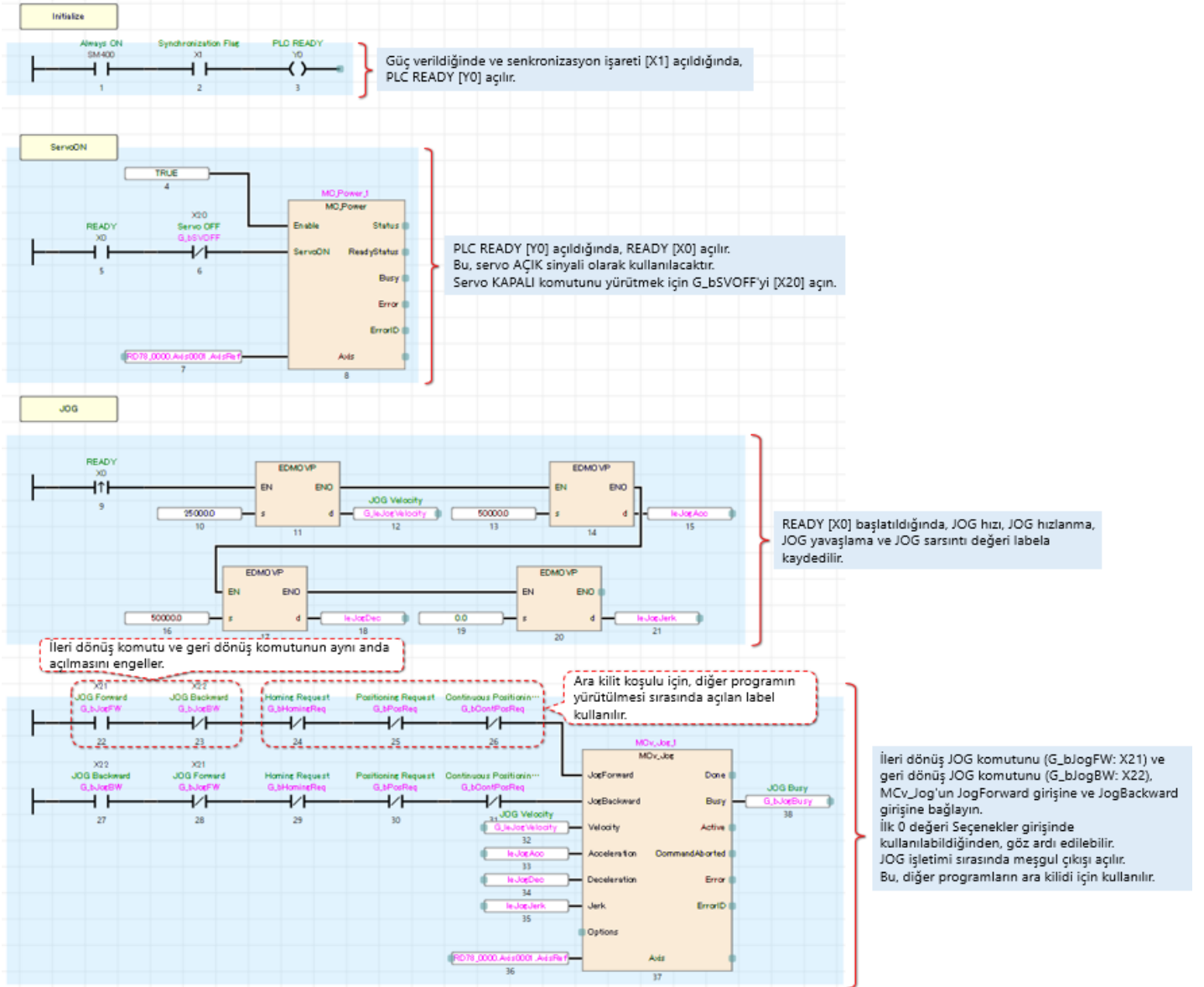
12 G_LoJobBusy

Bu şekilde programlama için girişler tamamlanır.
Sonraki sayfaya geçmek için > düğmesini tıklayın.

R04 Host (R=13, C=11) CAP NUM

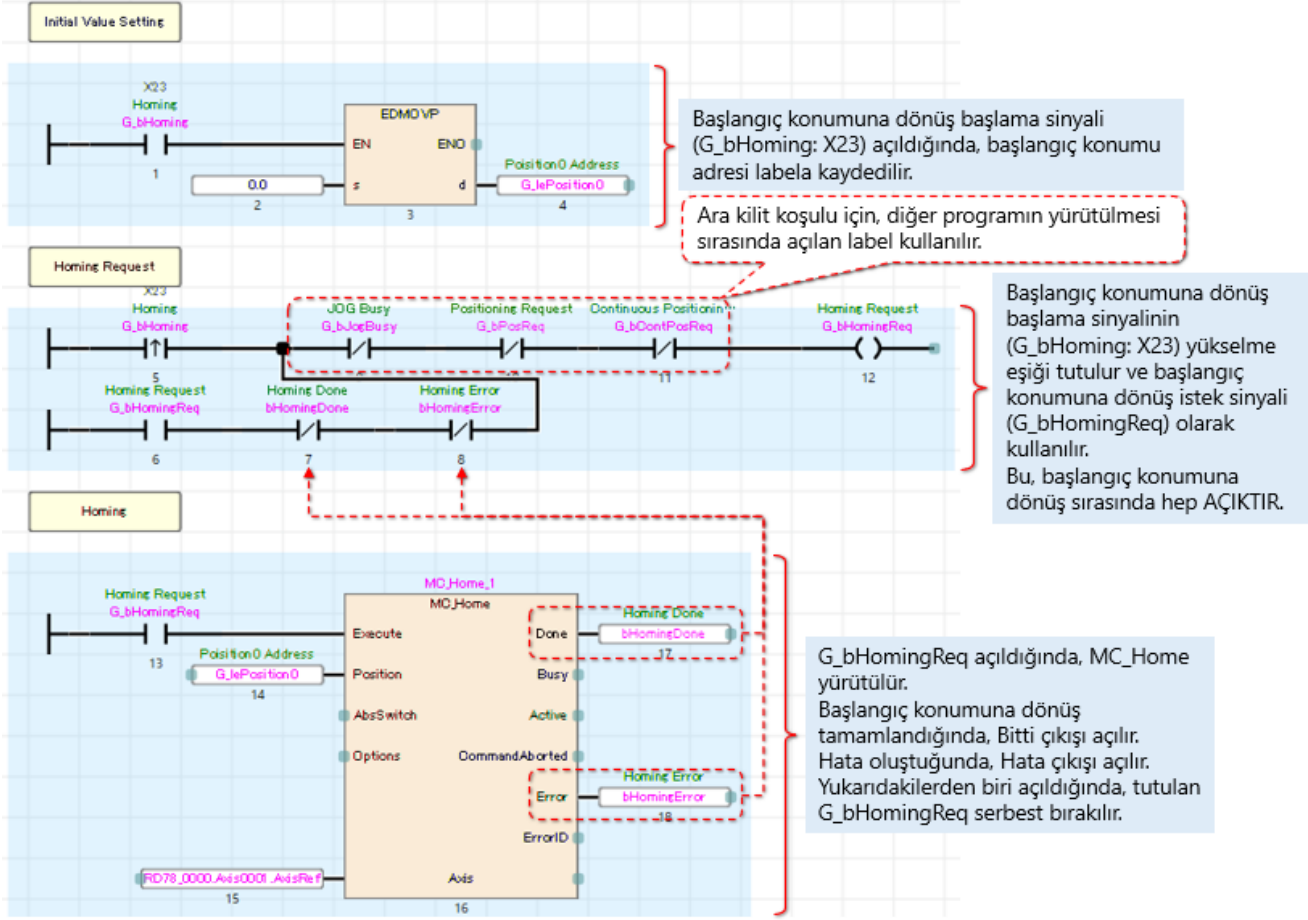
(1) ServoON_Jog

Bu program; PLC hazır AÇIK, servo AÇIK ve JOG işletimini yürütür.



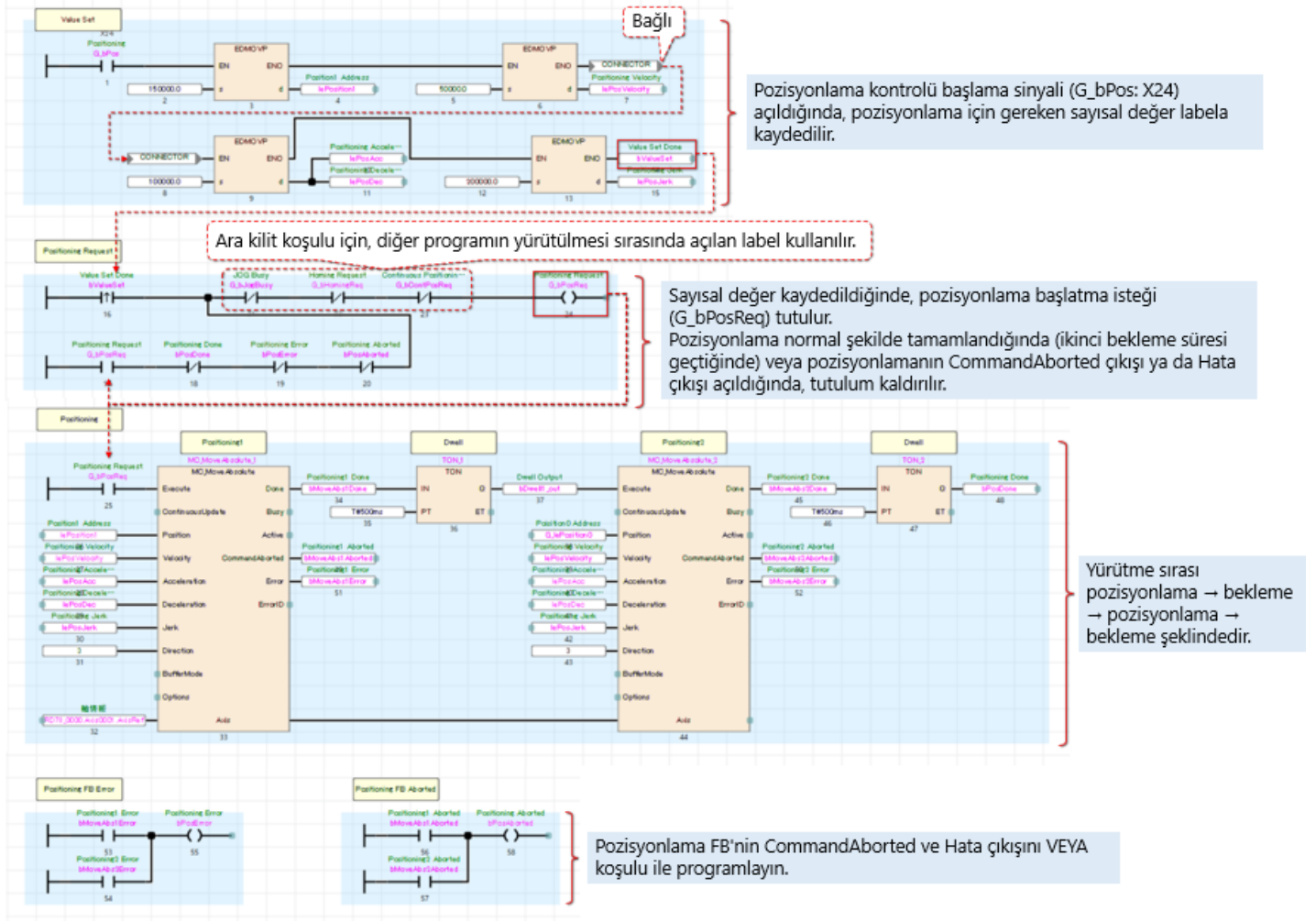
(2) Başlangıç dönüş

Bu program, başlangıç konumuna dönüşü gerçekleştirir.



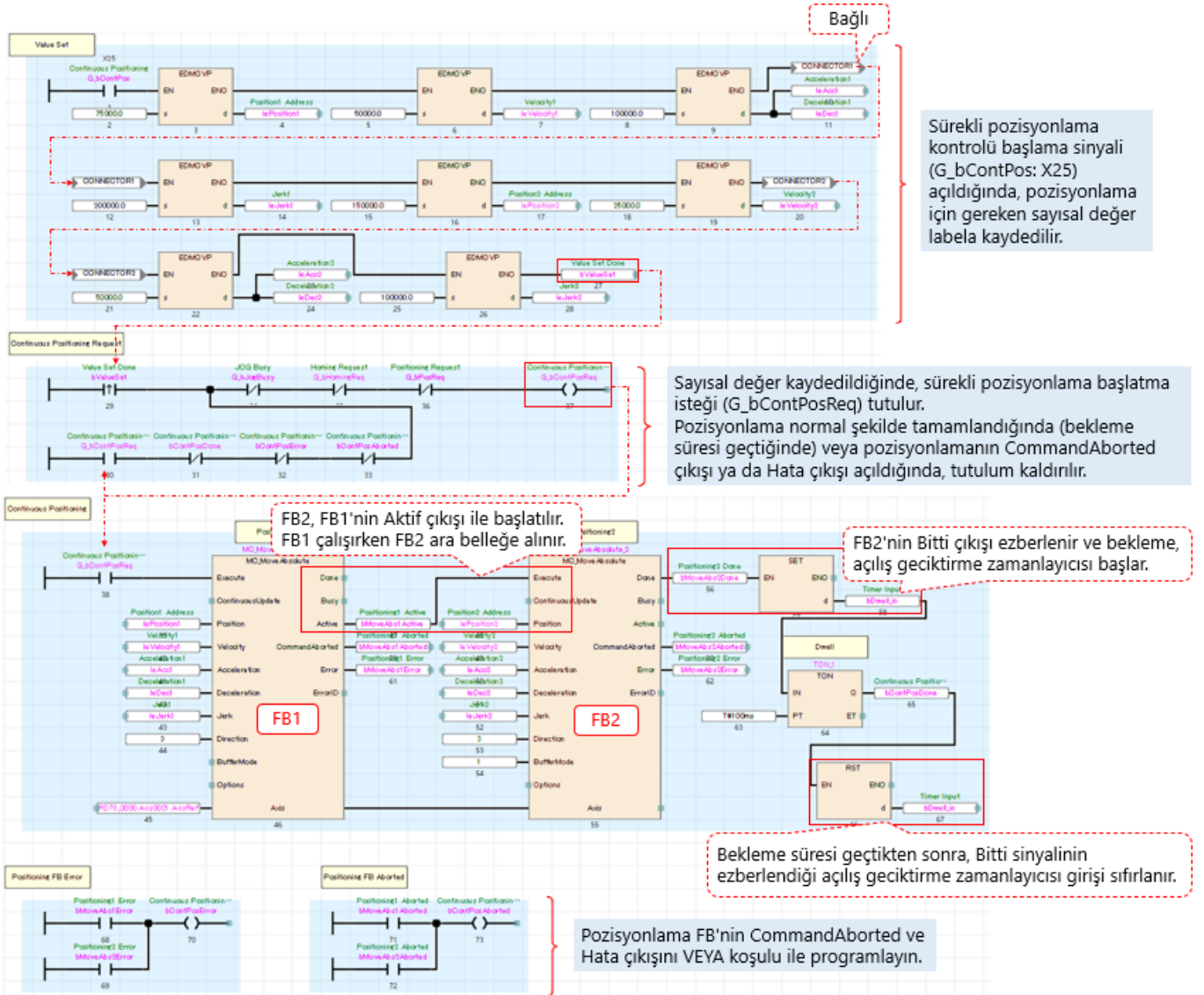
(3) Pozisyonlama

Bu program, tek eksenli pozisyonlama işletimini gerçekleştirir.



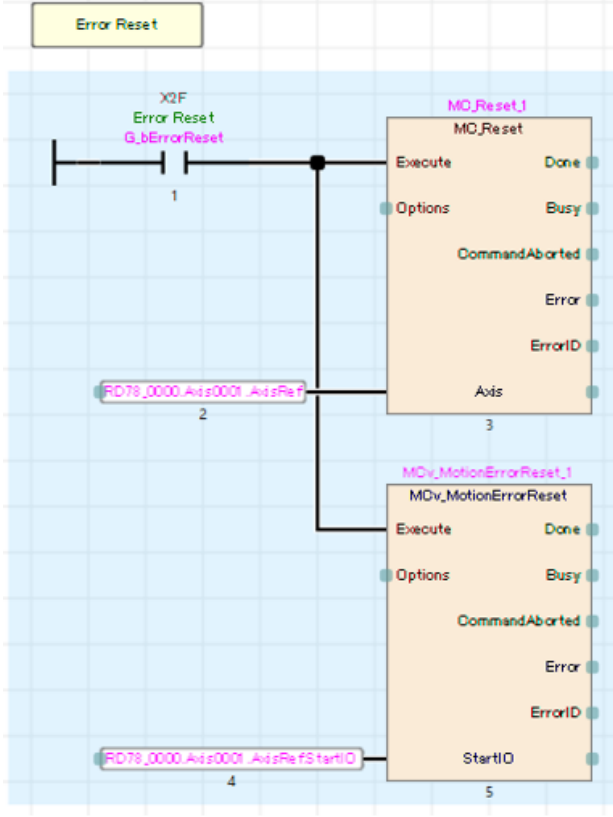
(4) ContinuousPositioning

Bu program, ara bellek modunu kullanarak sürekli pozisyonlamayı gerçekleştirir.



(5) ErrorReset

Bu program hata sıfırlamayı gerçekleştirir.



Hata sıfırlama sinyali (G_bErrorReset: X2F) açıldığında, eksen hatası sıfırlama (MC_Reset) ve sistem hatası sıfırlama (MCv_MotionErrorReset) yürütülür.

(6) Monitör

Bu program, PLC CPU'nun D0 ve D2'sine atanan eksen monitörü genel labeline ait SetPosition (Pozisyon Ayarla) ve SetVelocity (Hız Ayarla) değerlerini saklar.

SetPosition ve SetVelocity çift hassasiyetli reel sayı tipinde olduğundan, PLC CPU tarafından kolayca işlenebilmeleri için imzalı çift word tipine dönüştürülürler. (Not)

Bu word device'ları bu kursta kullanılmamaktadır.

Bunlar, diğer sekans (Sequence) programlarında ve GOT üzerinde görüntüleme amacıyla ve diğer amaçlarla kullanılmaktadır. Bu program ST ile açıklanmaktadır.

```
1 G_dSetPosition := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetPosition);  
2 G_dSetVelocity := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetVelocity);  
3
```

G_dSetPosition → D0
G_dSetVelocity → D2

(Not) Çift hassasiyetli reel sayı tipi, imzalı çift word tipine dönüştürüldüğünde, dönüştürülecek değer -2147483648 ila 2147483647 aralığının dışındaysa, bir hesaplama hatası oluşur.

Programı ve parametreleri PLC CPU ve Motion modülüne yazın.

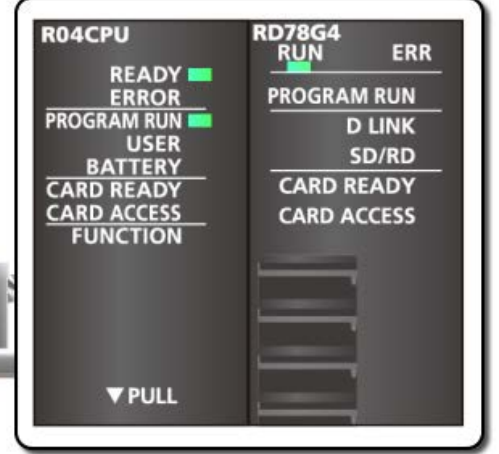
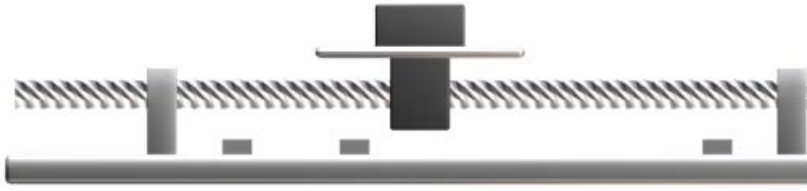
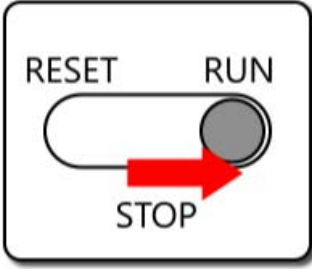
Program yalnızca CPU modülüne yazılır. Eksen parametresi ve açık label ayarlarının, Motion modülü tarafına yazılması gerekir.

- 1) PLC CPU'daki tüm programlar yeniden oluşturulduktan sonra, tüm verileri PLC CPU'ya yazmak için GX Works3 araç çubuğunda [Online] → [Write to PLC] seçin.
- 2) Parametreler PLC CPU'ya yazıldığında, Motion modülüyle haberleşme etkin hale gelir.
Tüm verileri Motion modülüne yazmak için Motion Control Setting Function araç çubuğunda [Online]→[Write to Module] seçin.
- 3) Yazma işlemini bitirmek için PLC CPU'yu sıfırlayın.

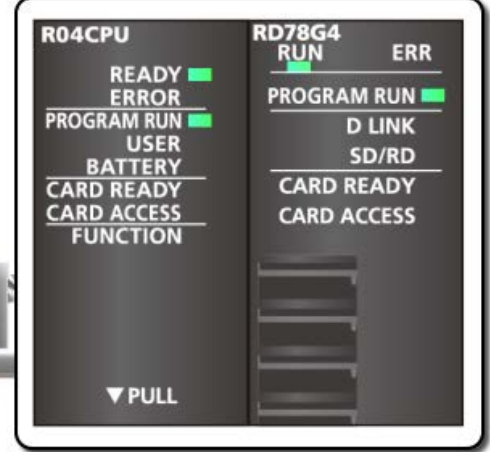
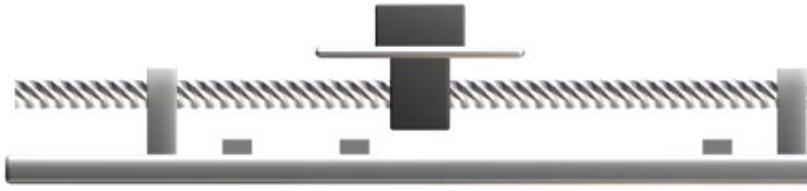
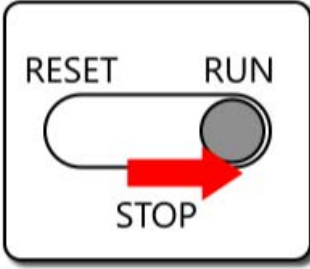
Pencerenin sol alt kısmındaki oynat düğmesini tıklayın.



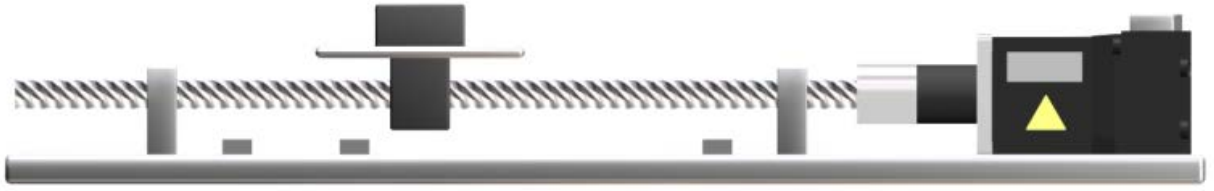
Örnek program çalışmasını kontrol edin.
Çalıştırmaya başlamadan önce, programların ve parametrelerin
PLC CPU ve Motion modülüne yazıldığından emin olun.



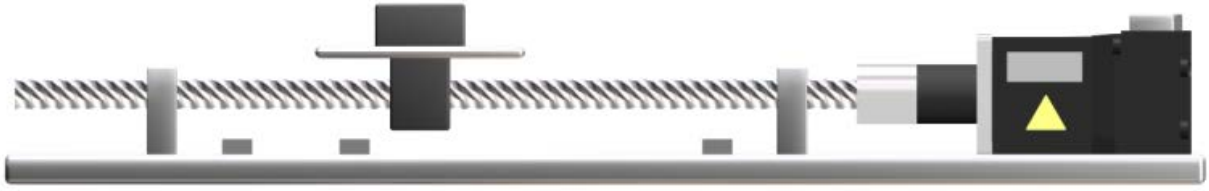
PLC CPU'nun RUN/STOP/RESET düğmesini, RUN durumuna getirin.
Programlanabilir kontrolörün READY lambası ve PROGRAM RUN
lambası yanar.
Motion modülünün RUN lambası yanar.



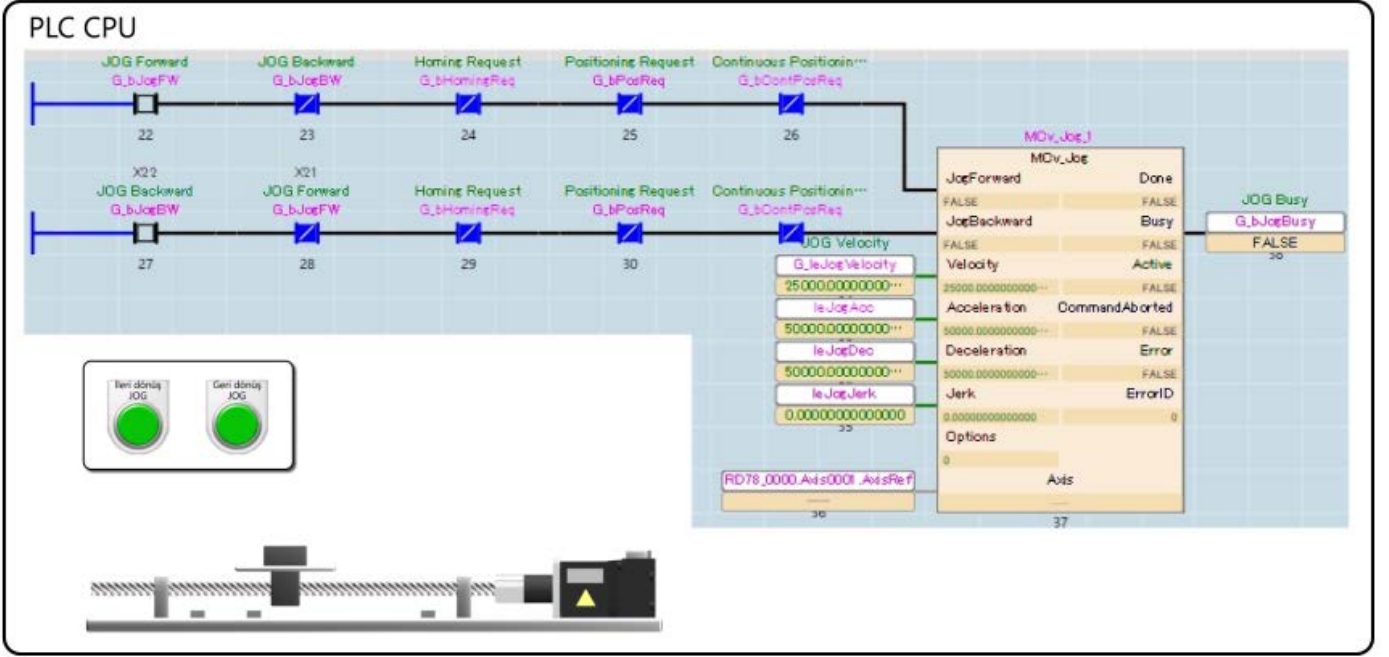
Motion modülünün PROGRAM RUN lambası yanana kadar bekleyin.
Servo sürücü üzerinde "r.01" görüntülenir. (Noktalar yanar.)
Servo motor, servo AÇIK durumuna girer.



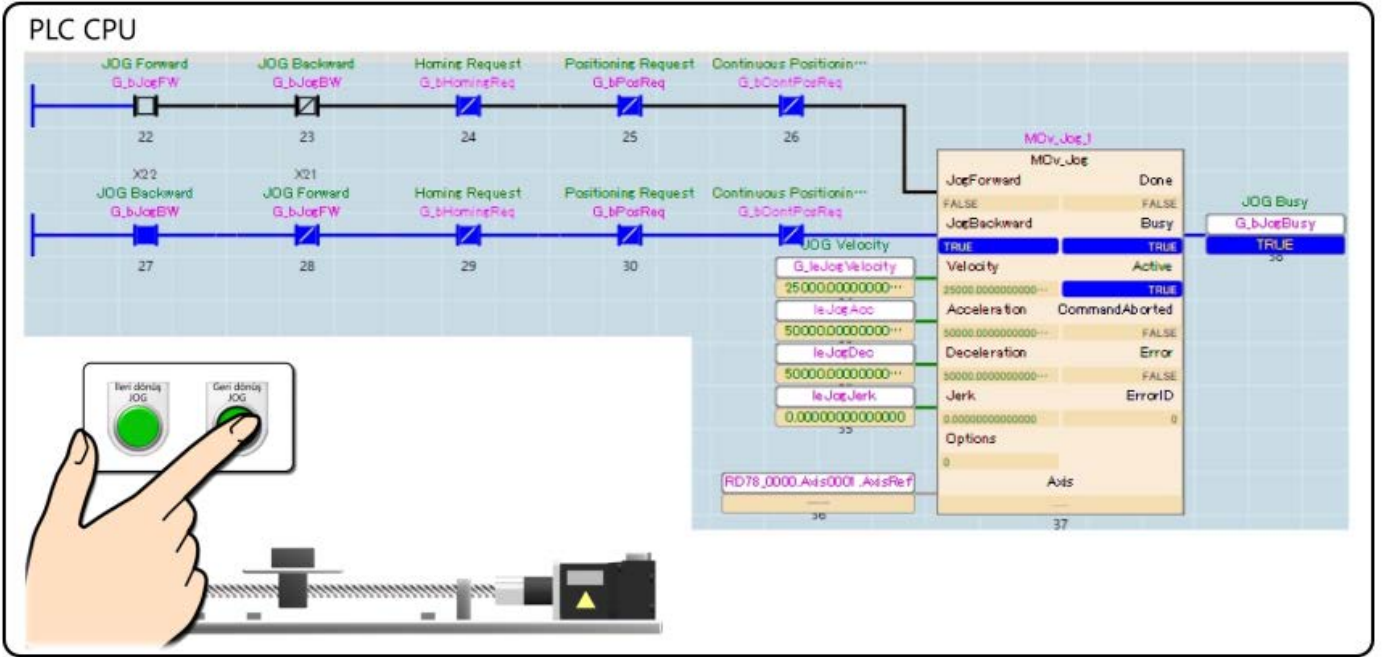
Servo KAPALI komutunu yürütmek için X20'yi açın.
Servo sürücü üzerinde "r.01" görüntülenir. (Noktalar yanıp söner.)
Servo AÇIK komutunu tekrar yürütmek için X20'yi kapatın.



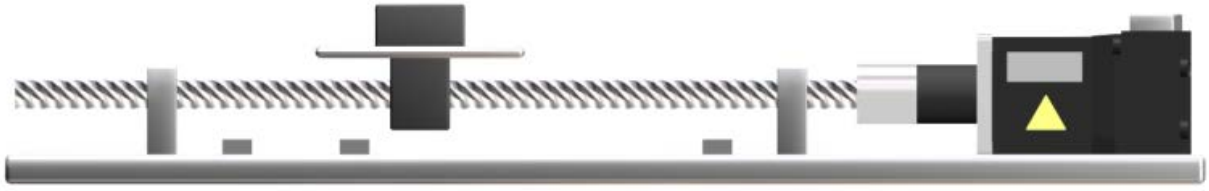
Adres artırma yönünde ilerlemek için ileri dönüş JOG'unu (X21) açın ve durdurmak için kapatın.
Adres azaltma yönünde ilerlemek için eksen geri dönüş JOG'unu (X22) açın ve durdurmak için kapatın.



Program monitörünü kontrol edin.
 X21 açıldığında, MCv_Jog_1'in JogForward girişi açılır.
 Normal dönüş JOG işletimi gerçekleşir.
 İşletim sırasında meşgul çıkışı ve "G_bJogBusy" açılır.



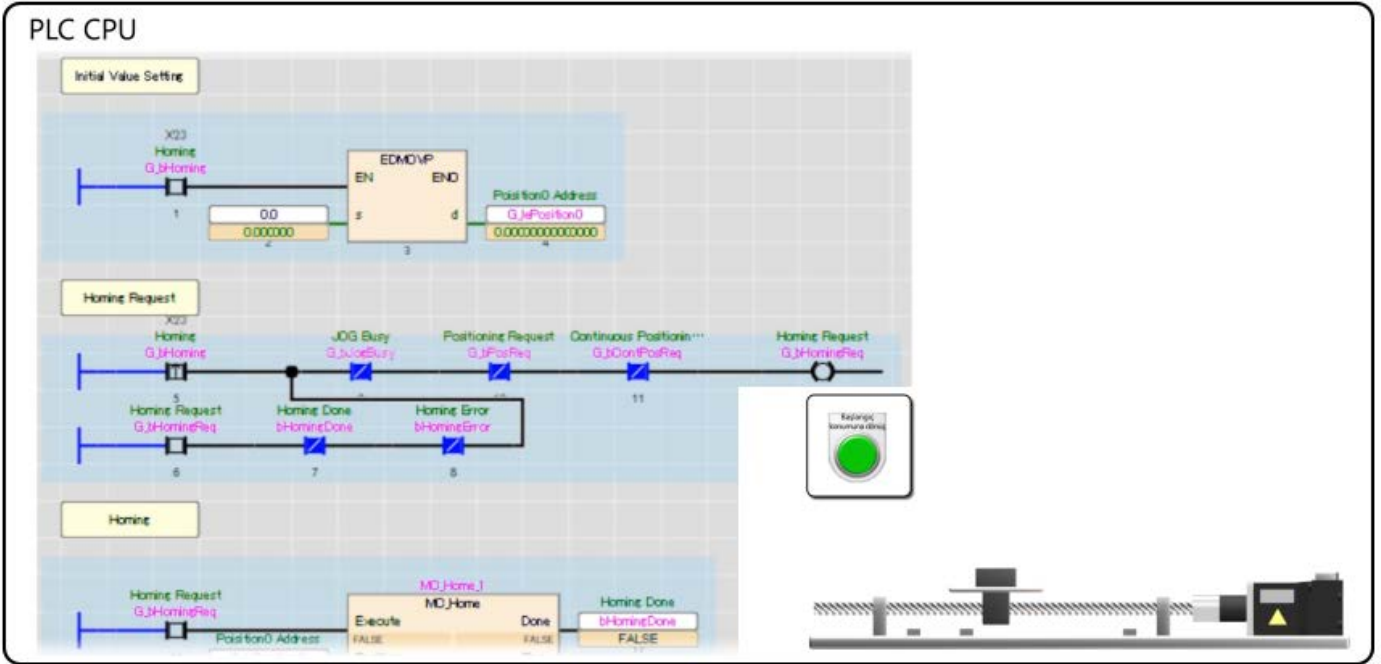
X22 açıldığında, MCv_Jog_1'in JogBackward girişi açılır.
Geri dönüş JOG işletimi gerçekleşir.
İşletim sırasında meşgul çıkışı ve "G_bJogBusy" açılır.



Başlangıç konumuna dönüşü başlatmak için başlangıç konumuna dönüşü (X23) açın.

Başlangıç konumuna dönüşü proximity dog yöntemiyle yürütün (Pr.PT45'ten 33 çıkarılır)

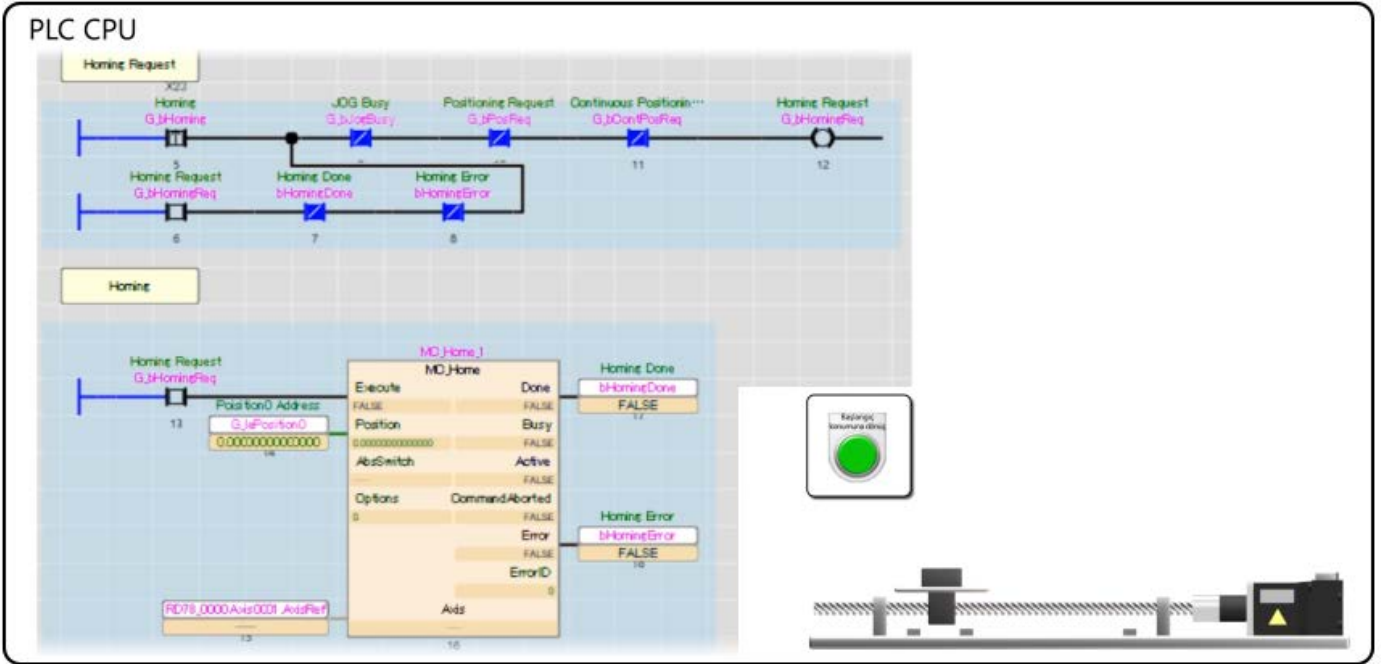
Eksen, dog'un biraz ilerisinde durur ve noktayı başlangıç konumu olarak ayarlar.



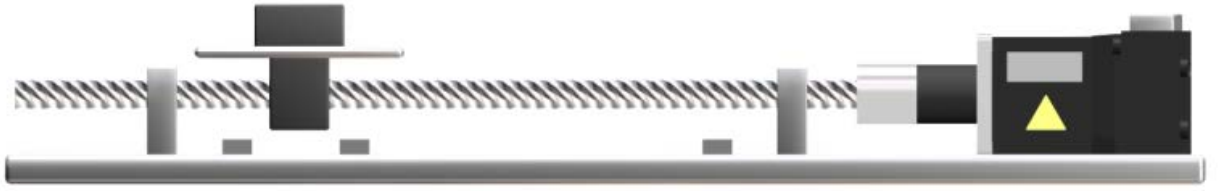
Program monitörünü kontrol edin.

X23 açıldığında, başlangıç konumu adresi labele kaydedilir.

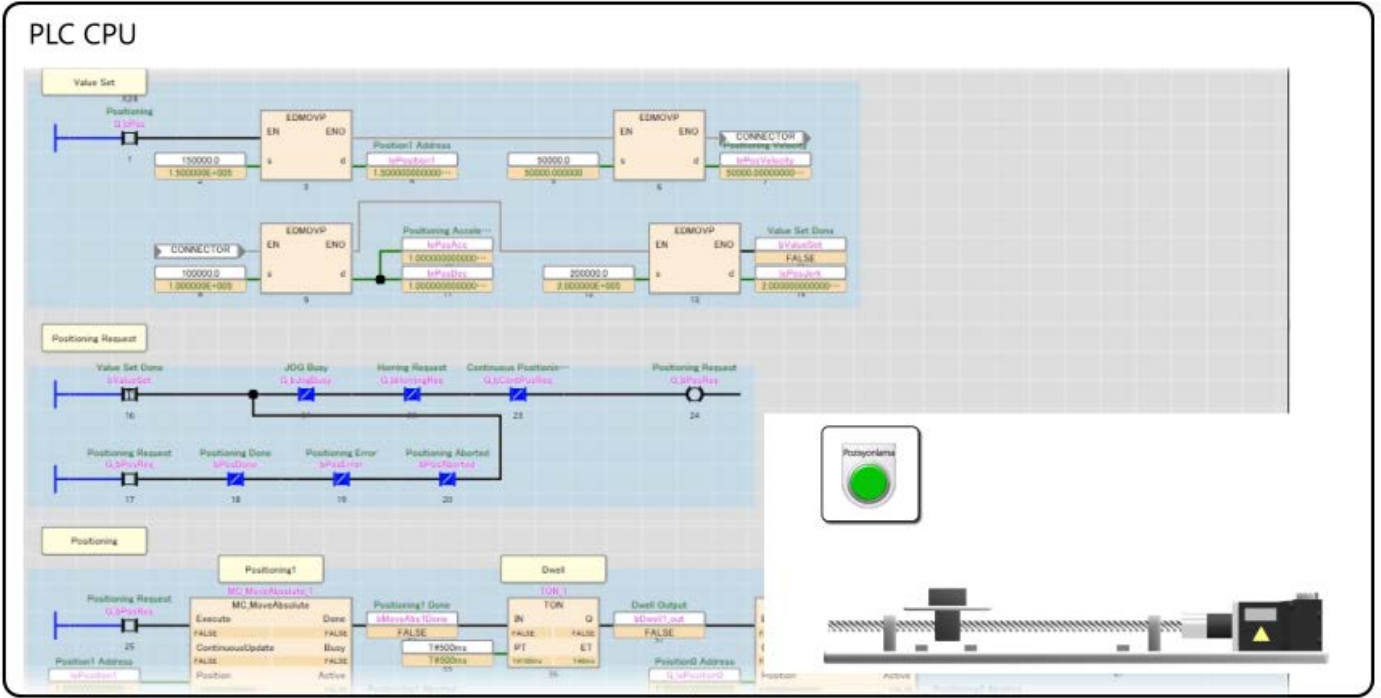
MC_Home_1'in yürütme komutu olan "G_bHomingReq" açılır ve tutulur.



Başlangıç konumuna dönüş işlemi başlar.
Başlangıç konumuna dönüş tamamlandığında, Bitti çıkışı ve "G_bHomingDone" açılır ve "G_bHomingReq" tutulumu iptal edilir.



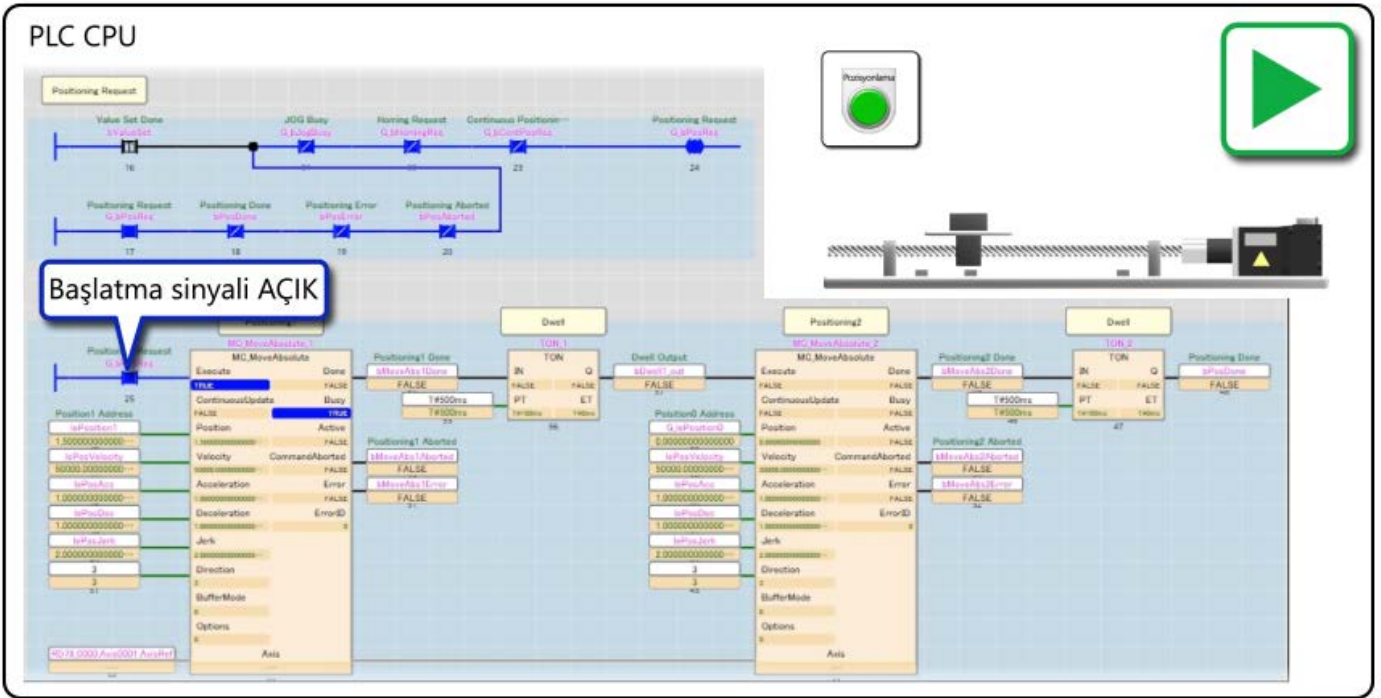
Pozisyonlama başlatmanın (X24) açılması ileri-geri hareketi başlatır. Eksen 150 mm ileri gider ve 0,5 saniye durur ve 150 mm geri gider ve 0,5 saniye durur.



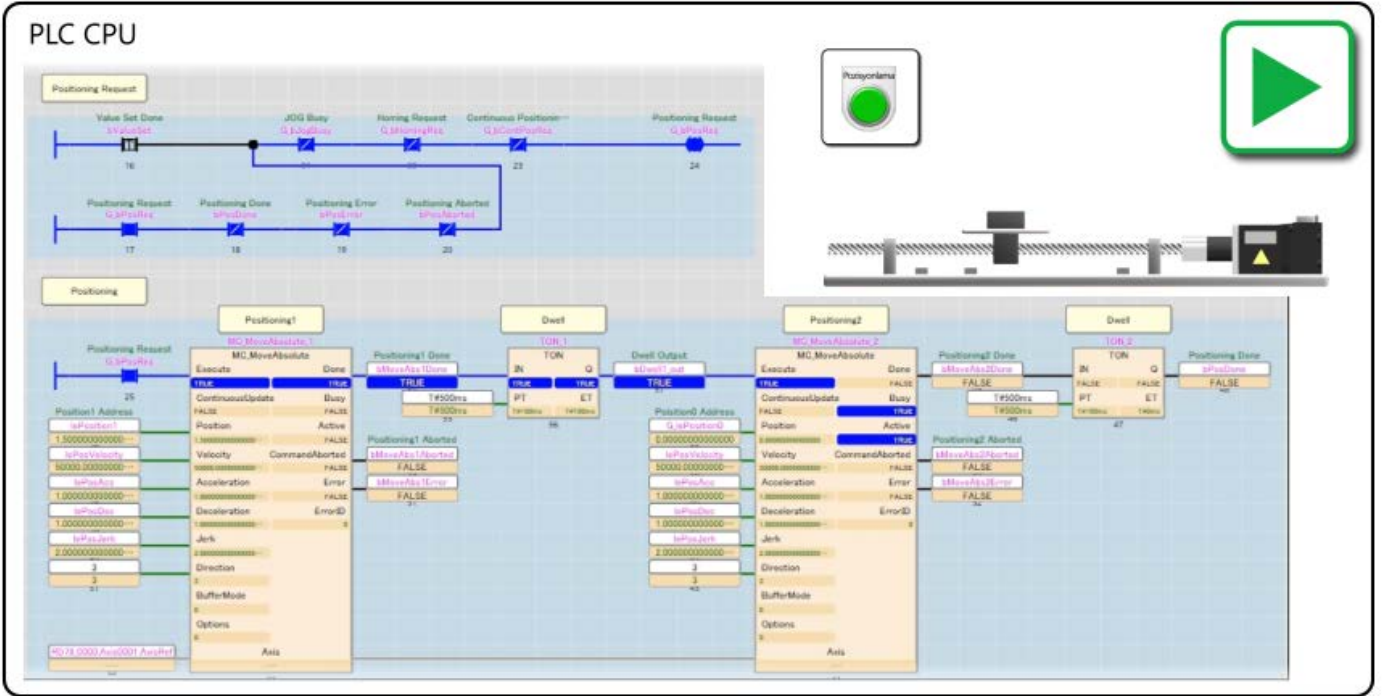
Program monitörünü kontrol edin.

X24 açıldığında, pozisyonlamaya ait veriler her label kaydedilir ve "bValueSet" açılır.

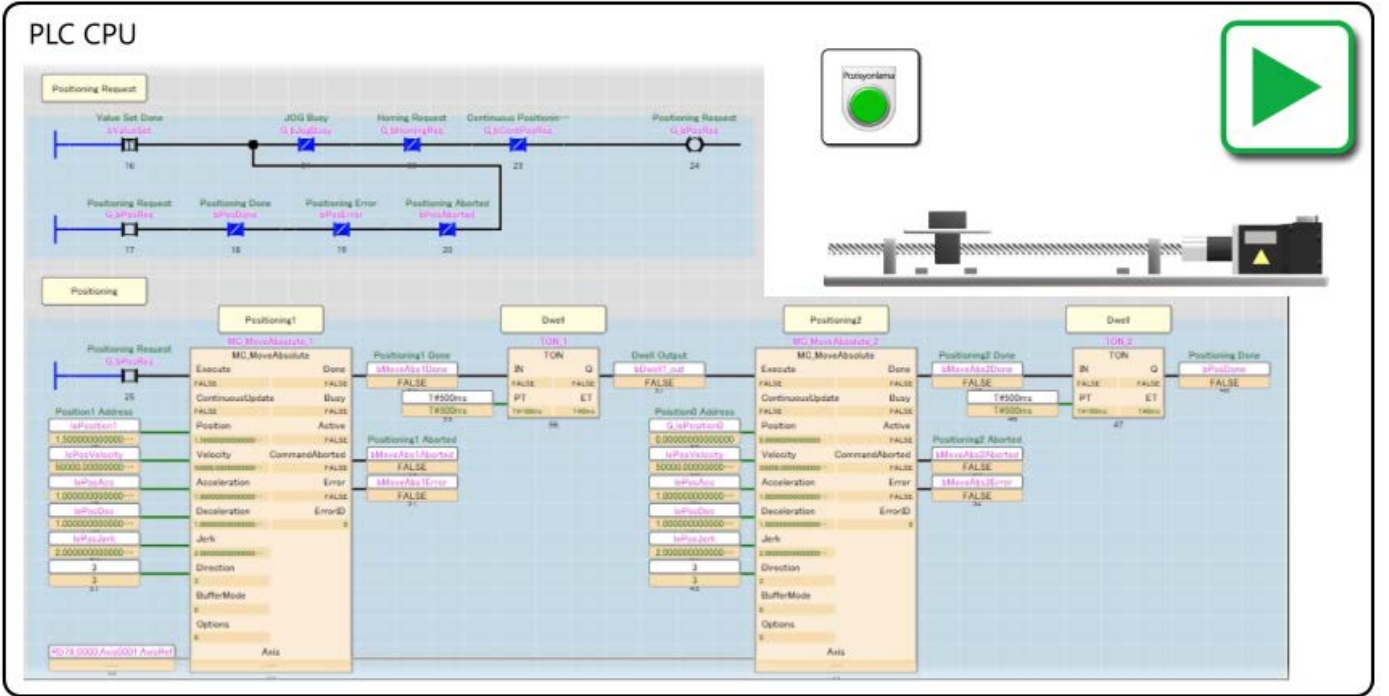
MC_MoveAbsolute_1'in yürütme komutu olan "G_bPosReq" açılır ve "bValueSet" yükselme eşliğinde tutulur.



"G_bPosReq" açıldığında, MC_MoveAbsolute_1 başlar ve servo motor çalışmaya başlar.



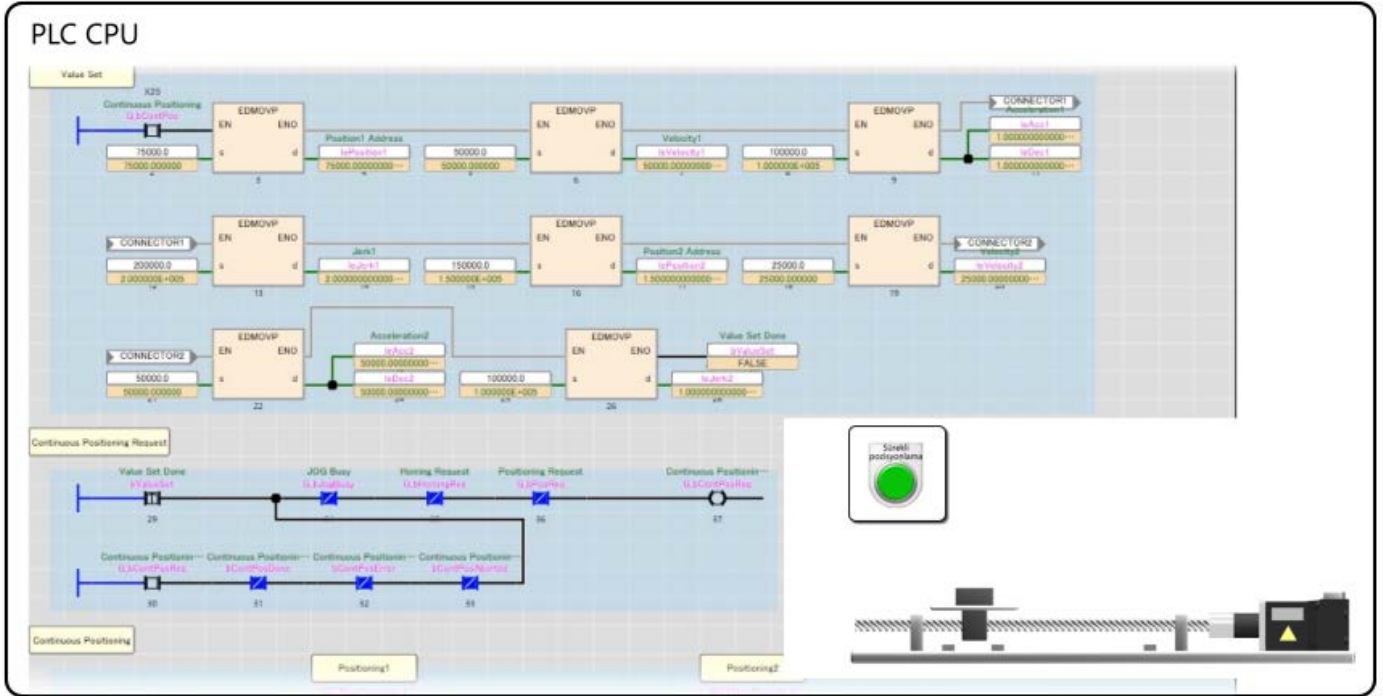
MC_MoveAbsolute_1 tarafından pozisyonlama tamamlandığında, bekleme olan TON_1 çalışır. 500 ms geçtiğinde, MC_MoveAbsolute_2 yürütülür ve servo motor çalışmaya başlar.



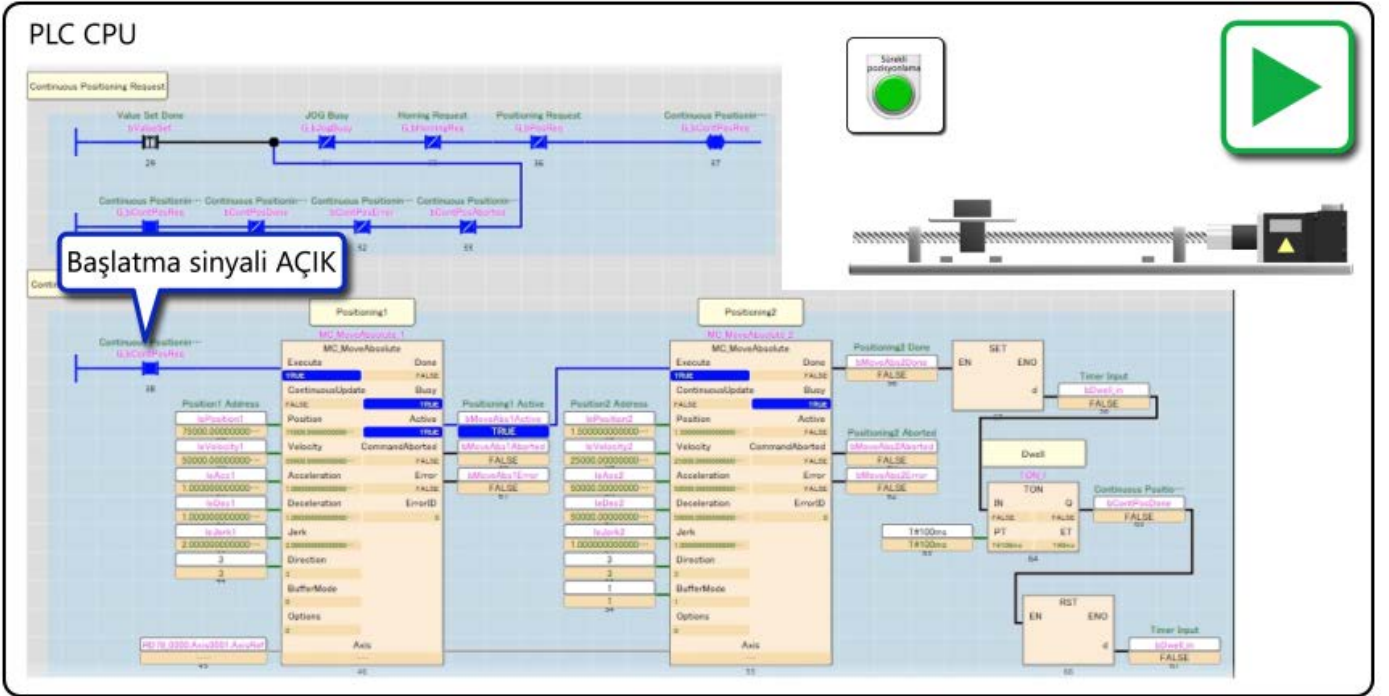
MC_MoveAbsolute_2 tarafından pozisyonlama tamamlandığında, bekleme olan TON_2 çalışır. 500 ms geçtiğinde, "G_bPosReq" tutulumu kaldırılır ve ilk duruma sıfırlanır.



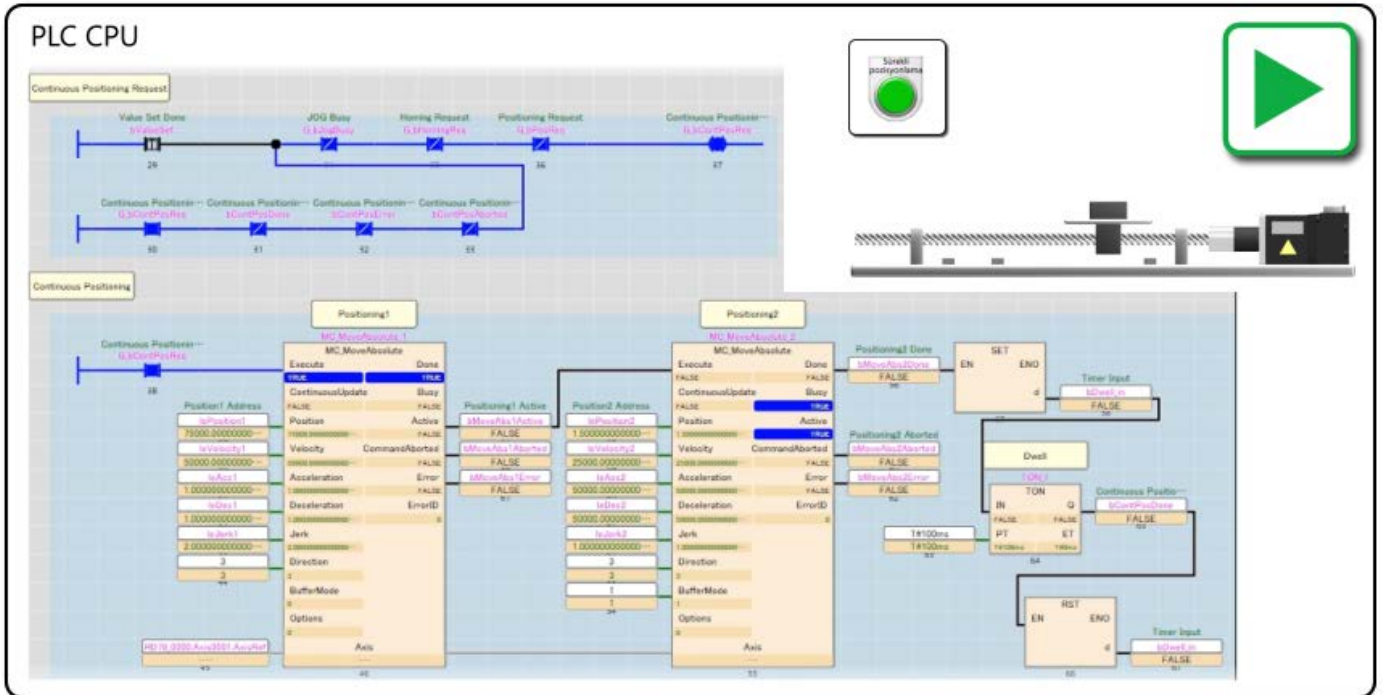
Ara bellek modunun işletimini başlatmak (mc_Buffered) için sürekli pozisyonlama başlatmayı (X25) açın.



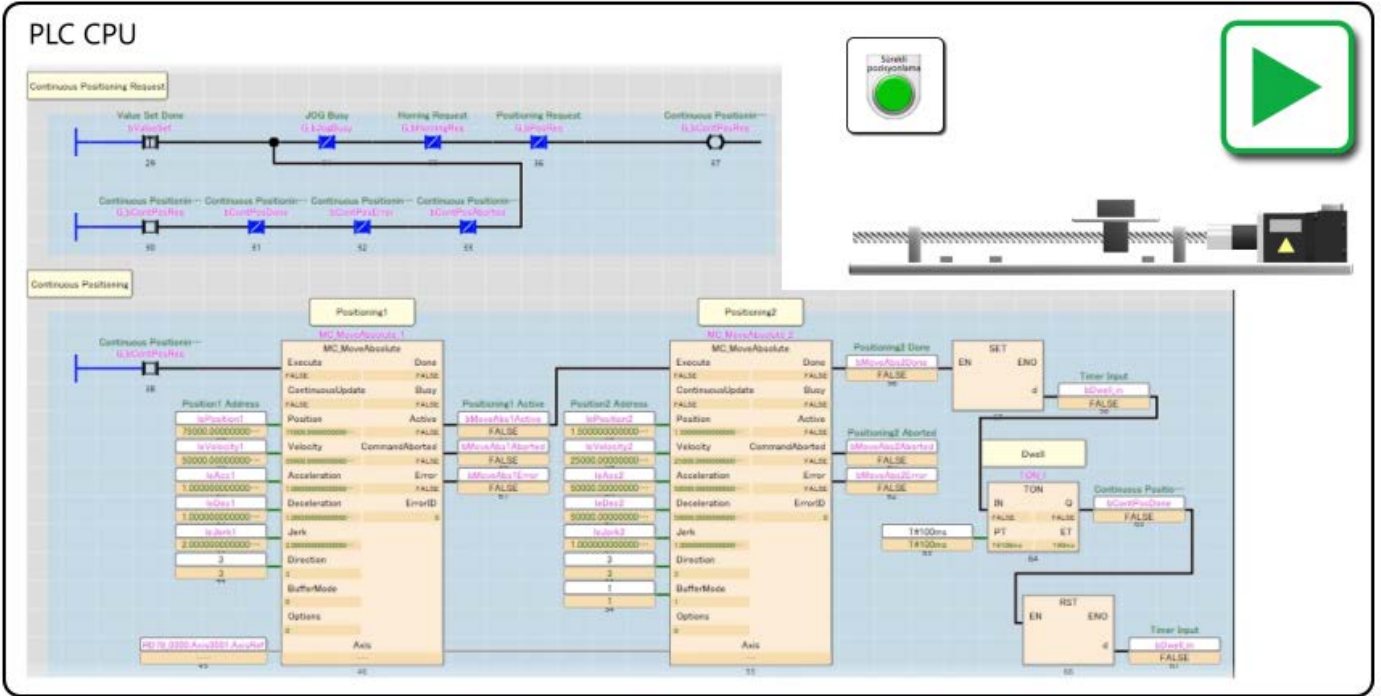
Program monitörünü kontrol edin.
 X25 açıldığında, pozisyonlamaya ait veriler her labela kaydedilir ve "bValueSet" açılır.
 MC_MoAbsolute_1'in yürütme komutu olan "G_bContPosReq" açılır ve "bValueSet" yükselme eşliğinde tutulur.



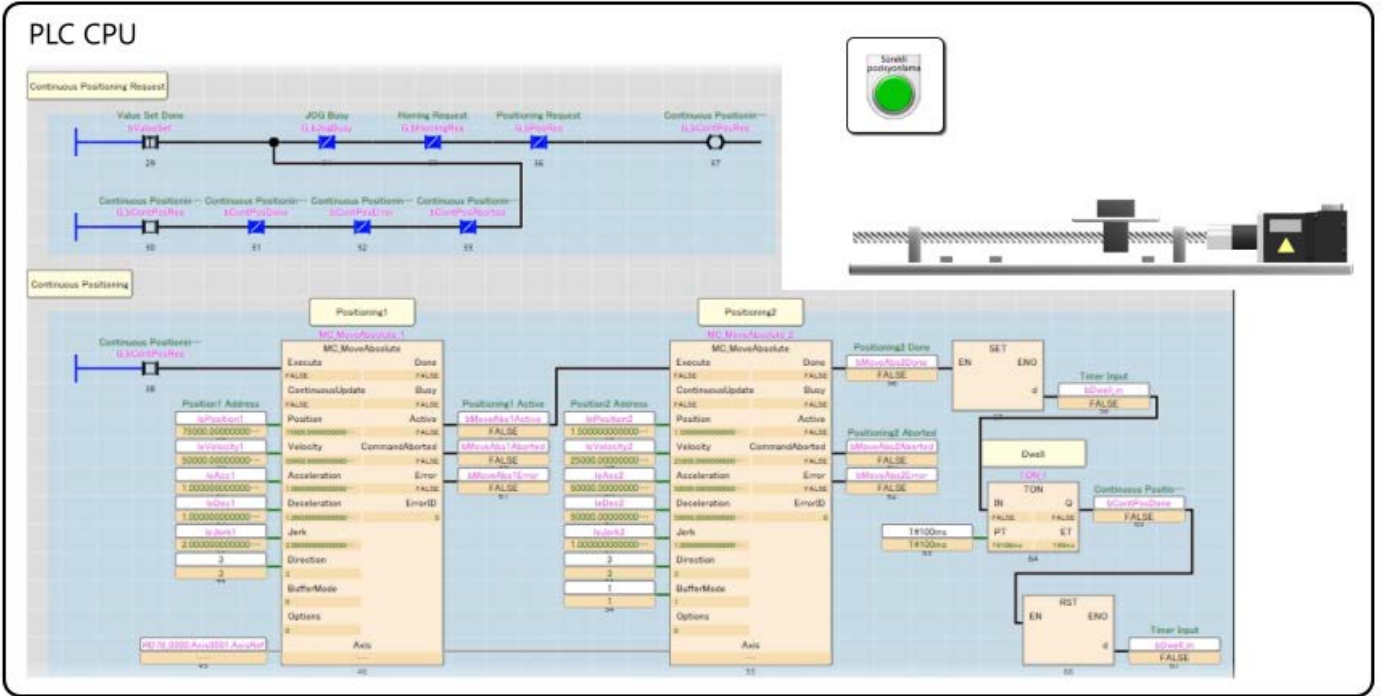
"G_bContPosReq" açıldığında, MC_MoveAbsolute_1 başlar ve servo motor çalışmaya başlar.
Bu sırada, Aktif çıkış MC_MoveAbosolute_2'nin yürütme komutu olduğundan, MC_MoveAbsolute_2 ara belleğe alınır.



MC_MoveAbsolute_1 işlemini tamamlandığında, ara belleğe alınan C_MoveAbsolute_2 yürütülür.
 MC_MoveAbsolute_2 işlemini tamamlandığında, bekleme olan TON_1 yürütülür.



100 ms geçtiğinde, "G_bContPosReq" tutulumu kaldırılır ve ilk duruma sıfırlanır.



Bu şekilde çalışma kontrolü tamamlanır.
Sonraki sayfaya gidin.

Bu bölümde, şunları öğrendiniz:

- Motion Modülü FB Kitaplığının Kaydedilmesi
- Proje Oluşturma
- Motion Control FB nasıl kullanılır
- Örnek Program Açıklaması
- Örnek Program Çalışma Kontrolü

Nokta

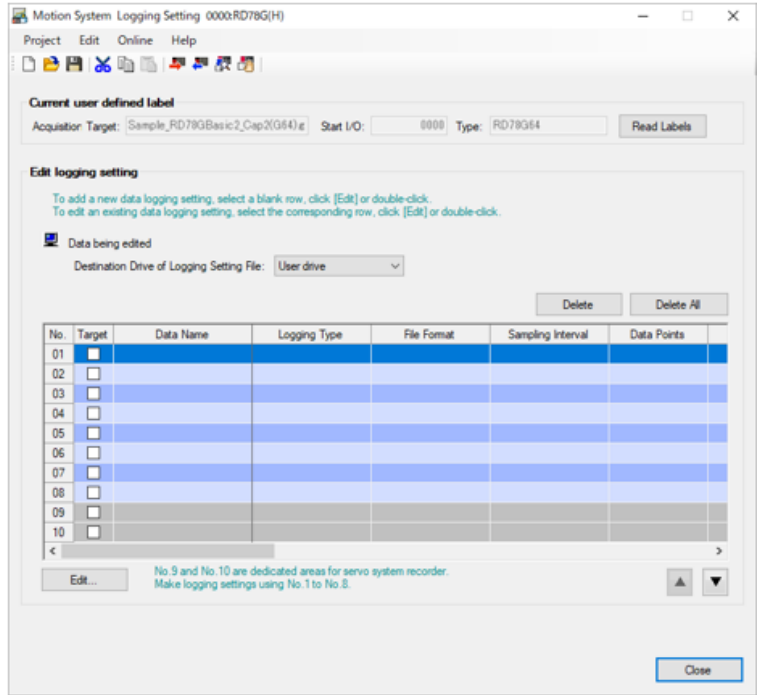
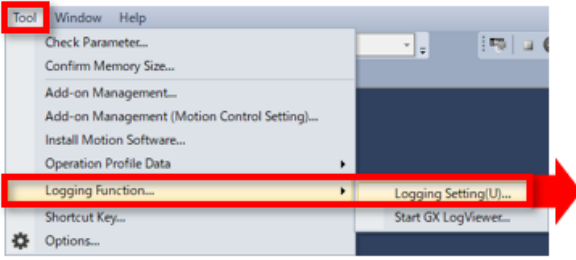
Motion Modülü FB Kitaplığının Kaydedilmesi	<ul style="list-style-type: none">• Motion control FB'yi PLC CPU'da kullanmak için FB kitaplığının GX Works3'e kaydedilmesi gerekir.
Proje Oluşturma	<ul style="list-style-type: none">• Eksen parametrelerini ve diğer ayarları, Motion modülüne programlama yaparken olduğu gibi yapılandırın.
Motion Control FB nasıl kullanılır	<ul style="list-style-type: none">• Motion kontrol FB, GX Works3'ün Öge Seçme penceresindeki Kitaplık sekmesinden sürüklenip bırakılarak program düzenleyiciye yerleştirilebilir.• Kontak ve labeli, FB'nin giriş/çıkışına bağlayın.
Örnek Program Açıklaması	<ul style="list-style-type: none">• Yalnızca PLC CPU kullanarak bölüm 2 ve bölüm 3'teki örnek programlara benzer bir program oluşturduunuz.
Örnek Program Çalışma Kontrolü	<ul style="list-style-type: none">• Örnek programın çalışmasını videoda kontrol ettiniz.

Bu bölümde, Motion modülüne ait verilerin günlüğe nasıl kaydedildiği ve grafikte görüntülediği açıklanmaktadır. Bu kursta, örnek olarak bölüm 2 ve bölüm 3'teki örnek programın pozisyonlama başlatma programının verileri günlüğe kaydedilecektir.

(Not) Bölüm 4'teki program, bu bölümde açıklanan işlemlerle günlüğe kaydedilemez.

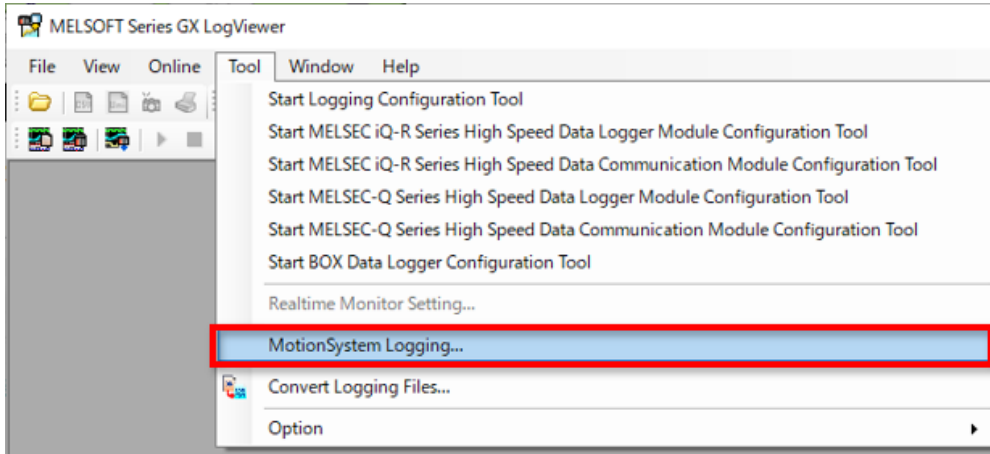
"CPU Module Logging Configuration Tool" kullanılmalıdır.

Motion Kontrol Ayar Fonksiyonu ekranının araç çubuğundan [Tool] → [Logging Function] → [Logging Setting] seçin. Motion sistemi günlük kaydı ayarlama aracı açılır.



[Nokta]

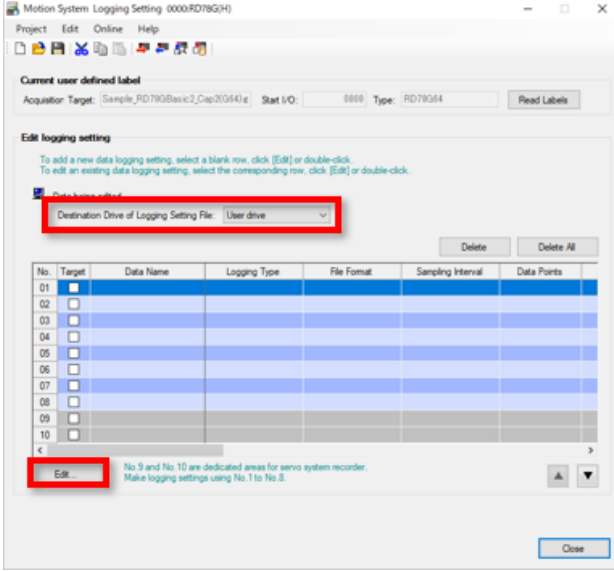
Motion sistemi günlük kaydı ayarlama aracı, GX LogViewer'da [Tool] → [MotionSystem Logging] seçimleri yapılarak başlatılabilir.



(1) Motion sistemi günlük kaydı ayarlama aracının günlük kaydını düzenleme ayar alanında, günlük kaydı verilerinin kayıt hedefini ayarlayın.

Ardından, [Edit] düğmesini tıklayın.

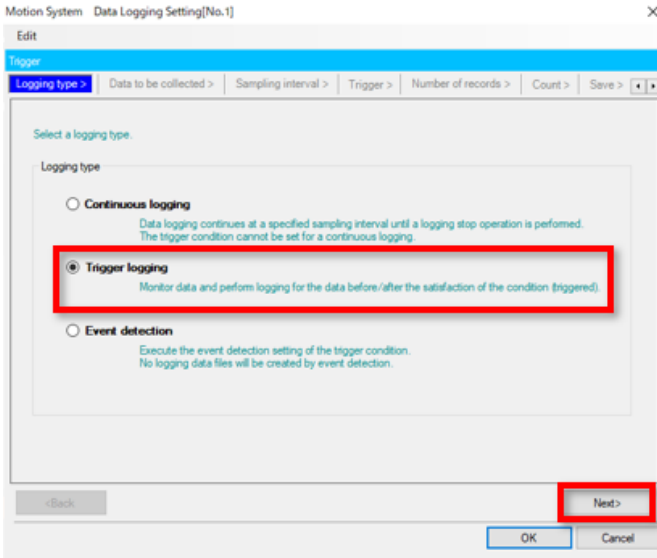
Veri Kaydı Ayarlama Ekranı görüntülenir.



(2) [Logging type] ayarını sürekli kaydetme, tetikleyiciyle kaydetme ve olay tespiti arasından seçin.

Bu kursta, tetikleyiciyle kaydetmenin ayrıntıları açıklanmaktadır.

Tetikleyiciyle kaydetme seçimini yapın ve [Next] düğmesini tıklayın.



(3) Günlüğe kaydedilecek verinin labeli [Data to be collected] bölümüne kaydedilir.

1) Genel label

Genel labeli veri adı alanına girin.
Veri türü alanında, labelin veri türünü seçin.

2) Yerel label

Veri adını "program adı/yerel label adı" biçiminde girin.
Veri türü alanında, labelin veri türünü seçin.

3) Yapılandırılmış (Structured) veri türü

[Label Input Assistant] düğmesini tıklayın ve listeden yapılandırılmış (Structured) veri türü üyesini seçin.
Bunu listeden seçin ve toplanacak veriye yansıtmak için [Register] düğmesini tıklayın.

Motion System Data Logging Setting[No.1] Edit

Trigger

Logging type > Data to be collected > Sampling interval > Trigger > Number of records > Count > Save >

Set the data to collect during logging

Data points to be collected: 7 /1024Points Items to be displayed: 7 /32

Label Input Assistant

No.	Display	Type	Data Name	Data Type	Size [Word]	Axis Name
0001	<input checked="" type="checkbox"/>	VAR	G_bPosReq	Bit	1	
0002	<input checked="" type="checkbox"/>	VAR	Positioning/bMoveAbs1Done	Bit	1	
0003	<input checked="" type="checkbox"/>	VAR	Positioning/bDwell1_out	Bit	1	
0004	<input checked="" type="checkbox"/>	VAR	Positioning/bMoveAbs2Done	Bit	1	
0005	<input checked="" type="checkbox"/>	VAR	G_bPosDone	Bit	1	
0006	<input checked="" type="checkbox"/>	VAR	Axis0001.Md.SetPosition	Float (double precision)	4	Axis0001
0007	<input checked="" type="checkbox"/>	VAR	Axis0001.Md.SetVelocity	Float (double precision)	4	Axis0001
0008	<input type="checkbox"/>	VAR				
0009	<input type="checkbox"/>	VAR				
0010	<input type="checkbox"/>	VAR				
0011	<input type="checkbox"/>	VAR				

<Back Next> OK Cancel

Bu kursta, örnek olarak aşağıdaki veriler günlüğe kaydedilmektedir.

Veri adı
G_bPosReq
Positioning/bMoveAbs1Done
Positioning/bDwell1_out
Positioning/bMoveAbs2Done
G_bPosDone
Axis0001.Md.SetPosition
Axis0001.Md.SetVelocity

Kayıt tamamlandığında [İleri] düğmesini tıklayın.

Label Input Assistant

Set the data to collect during logging.

Type: Real Drive Axis

Axis Name: Axis0001

Structured Data Type: Monitor Data

Data List

Name	Data Name (Label)
Md Object Data_VelActualValue	*.Md.Lo_VelActualValue
Md Jerk Limit Value	*.Md.JerkLimit
Md Control Cycle	*.Md.OperationCycle
Md Operation Setting at Overrun	*.Md.OverrunOperation
Md Current Position Restoration Status	*.Md.PosRestoration_Status
Md Execution Profile ID No.	*.Md.ProfileID
Md Set Acceleration	*.Md.SetAcceleration
Md Set Position	*.Md.SetPosition
Md Set Velocity	*.Md.SetVelocity
Md Slave Emulating	*.Md.SlaveEmulate_Enable
Md Start Permission at Homing Uncompleted	*.Md.StartableAtUnhomed
Md Deceleration at Stop	*.Md.StopMode_Deceleration
Md Stop Signal Signal Detection Method	*.Md.StopSignal_Detection

Register Close

- (4) [Sampling Interval] kısmında örnekleme aralığını ayarlayın.
Bu kursta, örnekleme için çalışma döngüsü No. 1 kullanın.

Örnekleme aralığını seçtikten sonra, [Next] düğmesini tıklayın.

The screenshot shows the 'Motion System Data Logging Setting[No.1]' window. The 'Sampling interval' tab is selected. The 'Operation cycle(S)' radio button is selected, and the 'Operation cycle selection' dropdown is set to 'Operation cycle No.1'. The 'Time' radio button is unselected. The 'Next' button is highlighted with a red box.

Motion System Data Logging Setting[No.1]

Edit

Trigger

Logging type > Data to be collected > Sampling interval > Trigger > Number of records > Count > Save >

Set the data sampling interval.

Sampling interval

Operation cycle(S)
Collect data at an interval of the operation cycle.
Operation cycle selection: Operation cycle No.1

Time
Collect data at a specified interval. Set the time of the data sampling interval by ms.
1000.000 [ms] (0.032 to 86400.000)

<Back

Next >

OK Cancel

(5) Günlük kaydına başlama koşulu [Trigger] kısmında ayarlanır.

Bu kursta, pozisyonlama başlatma sinyali olan başlatma biti, tetikleyici olarak kullanılmaktadır.

- 1) [With trigger] seçin.
- 2) Tetikleyici tipi için "OR combination" seçin.
- 3) Koşul listesinde No. 1 seçin ve düzenleme düğmesini tıklayın. Alt pencere açılır.
- 4) "Data Condition" seçin ve veri adı için "0001:G_bPosReq" seçin.
Koşul için "UP" seçin.
Seçim tamamlandığında, [OK] düğmesini tıklayın.
- 5) Orijinal ekrana geri döndükten sonra, [Next] düğmesini tıklayın.

Motion System Data Logging Setting[No.1]

Edit

Trigger

Logging type > Data to be collected > Sampling interval > **Trigger >** Number of records > Count > Save >

Set the trigger type and the trigger condition. Without trigger **With trigger**

Trigger Type: **OR combination** Description: Triggered when any of the set conditions is satisfied.

Condition List

Monitoring Data Points: 1 /32 Points **Edit** Delete Delete All

No.	Type	Content
1	Data Condition (OFF-ON)	G_bPosReq
2		
3		
4		
5		
6		
7		
8		

<Back **Next>** OK Cancel

OR combination(No.1)

Data Condition Triggered when the set condition is satisfied by comparing data with data, or data with constant.

Data Name	Condition	Data/Constant	Data Name/Constant Value
0001:G_bPosReq	UP		

Fixed Scan Triggered every specified cycle. [1] [ms] (1 to 95400000)

At System Start Triggered at system start.

At Trigger Instruction Execution(W) Triggered when LoggingRef Cd Trigger is set to TRUE.

Event History Triggered when the event of the corresponding event history occurs.
- Set the event code of the trigger target.
- Multiple event codes can be set by using a ':' (hyphen) or a ',' (comma).

OK Cancel

- (6) Örneklem noktalarının sayısı [Number of records] içinde ayarlanır. Bu kursta, kayıt sayısı (tetikleyiciden önce) "500" olarak ve kayıt sayısı (tetikleyiciden sonra) "19500" olarak ayarlanmıştır. Ayar tamamlandığında, [Next] düğmesini tıklayın.

Motion System Data Logging Setting[No.1]

Edit

Trigger

Logging type > Data to be collected > Sampling interval > Trigger > Number of records > Count > Save >

Set the No. of records output to file when triggered during trigger logging.

No. of records (before trigger): 500 [Record] (0 to 299999)

No. of records (after trigger): 19500 [Record] (1 to 300000)

Total No. of records: 20000 [Record] (1 to 300000)

<Back Next>

OK Cancel

- (7) Günlük kaydı sayısı [Count] kısmında ayarlanır. Bu kursta, sayı 1 olarak ayarlanmıştır. Ayar tamamlandığında, [Next] düğmesini tıklayın.

Motion System Data Logging Setting[No.1]

Edit

Trigger

Logging type > Data to be collected > Sampling interval > Trigger > Number of records > Count > Save >

Specify the trigger counts.

Count

Specified Count 1 (1 to 32767)

Execute trigger logging repeatedly for the specified count.
The operation will be 'Overwrite' when the maximum number of saved files is exceeded.

Specified Number of Saved Files

Execute the trigger logging repeatedly according to the maximum number setting of saved files.

<Back Next>

OK Cancel

- (8) Günlük kaydı verilerinin dosya biçimi ve kaydedilen dosya sayısı [Save] kısmında ayarlanır. Bu kursta, varsayılan değer (biçim: JSON, kaydedilen dosya sayısı: 1) ayarlanır. Ayar tamamlandığında, [Next] düğmesini tıklayın.

Motion System Data Logging Setting[No.1]

Edit

Trigger

Sampling interval > Trigger > Number of records > Count > Save > Start condition > Finish

Save setting for logging data file

Set the destination folder of the logging data file and the file format.

File format: JSON

Save in the same folder as the setting file

Destination drive:

Folder:

Maximum number of saved files

Set the maximum number of saved files of the destination drive.
Files are created until the capacity of the destination becomes full when 'Do not specify the maximum value' has been enabled.

1 (1 to 65535)

Do not specify the maximum value

Operation when the maximum number is exceeded

Overwrite Files with lower numbers are deleted

Stop Logging stops.

<Back

Next >

OK Cancel

- (9) Günlük kaydına başlama koşulu [Start condition] kısmında ayarlanır. Bu kursta, "Start by User Operation" ayarlanmıştır. Ayar tamamlandığında, [Next] düğmesini tıklayın.

Motion System Data Logging Setting[No.1]

Edit

Trigger

Sampling interval > Trigger > Number of records > Count > Save > Start condition > Finish

Select the condition to start the data logging

Logging Start Condition

Auto Start
The data logging starts automatically at the start of the motion system.
It is possible to stop the logging manually.

Unstoppable After Auto Start
The data logging starts automatically at the start of the motion system.
Unable to stop the logging manually.

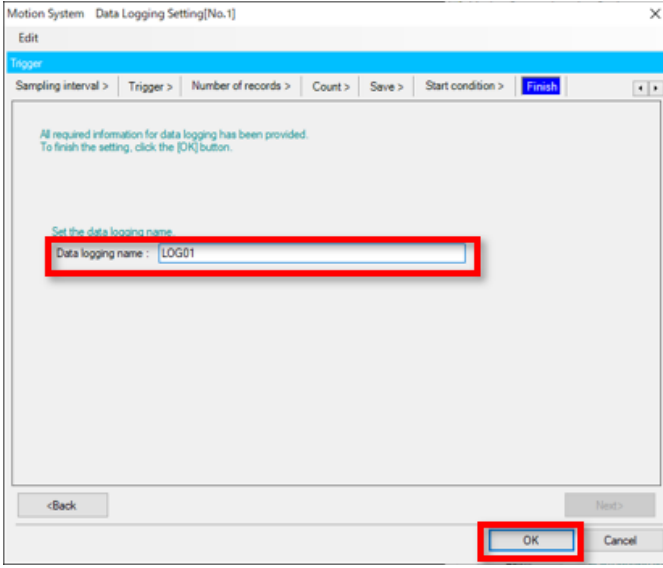
Start by User Operation
Execute the data logging start operation in the Logging Status and Operation window, etc.
It is also possible to stop the logging manually.

<Back

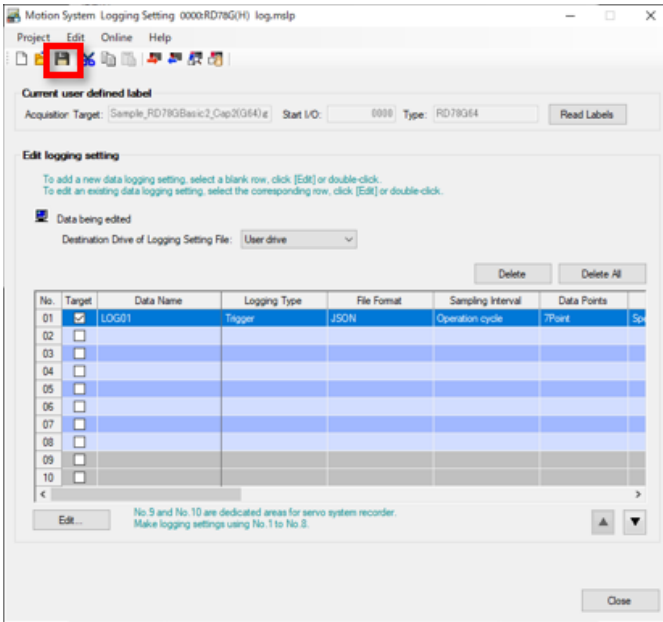
Next >

OK Cancel

- (10) Veri günlük kaydı adı [Finish] kısmında ayarlanır. Bu kursta, varsayılan değer (LOG01) ayarlanmıştır. Ayar tamamlandığında, [Next] düğmesini tıklayın.



- (11) Motion sistemi günlük kaydı ayarlama aracına geri dönün. Konfigüre edilmiş ayarlar kaydedilebilir. Kaydet simgesini tıklayın ve tercih ettiğiniz hedefe kaydedin.



Günlük kaydının ayarlama bilgisi yazılır.

Günlük kaydı ayarını yaz simgesini tıklayın, hedef belleği seçin ve [Write] düğmesini tıklayın.

Onay penceresi açılır.

[Yes] düğmesini tıklayın ve devam edin. Yazma tamamlandığında, [OK] düğmesini tıklayın ve ekranı kapatın.

Motion System Logging Setting - 0000:RD78G(H) log.mstp

Project Edit Online Help

Current user defined label
Acquisition Target: Sample_RD78GBasic2_Cap020641e Start I/O: 0000 Type: RD78G64 Read Labels

Edit logging setting
To add a new data logging setting, select a blank row, click [Edit] or double-click.
To edit an existing data logging setting, select the corresponding row, click [Edit] or double-click.

Data being edited
Destination Drive of Logging Setting File: User drive

No.	Target	Data Name	Logging Type	File Format	Sampling Interval	Data Points
01	<input checked="" type="checkbox"/>	LOG01	Trigger	JSON	Operation cycle	79Point
02	<input type="checkbox"/>					
03	<input type="checkbox"/>					
04	<input type="checkbox"/>					
05	<input type="checkbox"/>					
06	<input type="checkbox"/>					
07	<input type="checkbox"/>					
08	<input type="checkbox"/>					
09	<input type="checkbox"/>					
10	<input type="checkbox"/>					

Edit... No 9 and No.10 are dedicated areas for servo system recorder. Make logging settings using No.1 to No.8. Close

Write Logging Setting-0000:RD78G(H) (Host)

Online operation
 Read logging setting
 Write logging setting
 Delete logging setting

Target memory: User drive

Target logging setting data
Data being edited
If the servo system recorder add-on is enabled at system start, No.9 and No.10 will use logging settings of the SD memory card with priority. Refer to manuals for details.

Data Logging Setting
 No.01[LOG01]Trigger

Write Close

Motion System Logging Configuration Tool

Write the logging settings to the motion system.
< Connection Destination >
RD78G8

The following setting No., which has been included in the selected logging settings, already exists in the target memory of the write destination.

- No.01

Writing settings will overwrite the logging setting in the target memory.
Do you want to continue?

Yes No

Motion System Logging Configuration Tool

Completed writing the logging setting data.

< Caution >
Logging is not started just by writing logging settings.
Start logging in the 'Logging Status Display and Operation' window.
To display the 'Logging Status Display and Operation' window, select [Online] -> [Logging Status Display and Operation].

OK

5.2'de (9) "Start by User Operation" ayarlandığında [Logging Status and Operation] ekranını görüntülemek ve günlük kaydı başlatmak için [Logging Status and Operation] simgesini tıklayın.

Yürütülecek günlük veri kayıt adı seçildiğinde ve [Start] düğmesi tıklandığında, LoggingStatus durumu "Waiting for trigger" şeklinde değişir.

Bu durumda program yürütüldüğünde ve tetikleyici koşul karşılandığında (bu kurstaki örnekte X24 AÇIK hale getirildiğinde), durum "Triggered" olarak değişir.

Günlük kaydı tamamlandığında, durum "CollectionCompleted" iken "Saving" olarak değişir.

The image shows two screenshots of the Motion System software interface. The left screenshot is the 'Logging Setting' window, and the right screenshot is the 'Logging Status and Operation' window. A red arrow points from the 'Start' button in the right window to the 'Start' button in the left window. A red box highlights the 'Start' button in the right window. A red arrow points from the 'Start' button in the right window to the 'Logging Status' flowchart on the right.

The 'Logging Status and Operation' window shows the following data:

No.	Target	Data Name	Logging Type	File Format	Sampling Interval	Data Ports	Stopped
01	LOG01	Trigger	LOGON	Operation cycle	7Hz		<input checked="" type="checkbox"/>
02							<input type="checkbox"/>
03							<input type="checkbox"/>
04							<input type="checkbox"/>
05							<input type="checkbox"/>
06							<input type="checkbox"/>
07							<input type="checkbox"/>
08							<input type="checkbox"/>
09							<input type="checkbox"/>
10							<input type="checkbox"/>

The 'Logging Status' flowchart shows the following sequence of status changes:

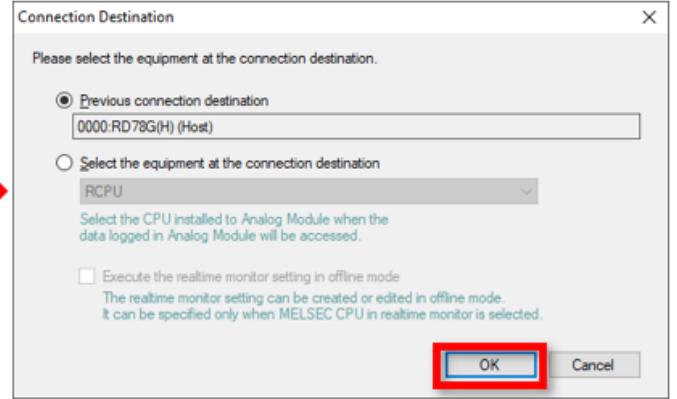
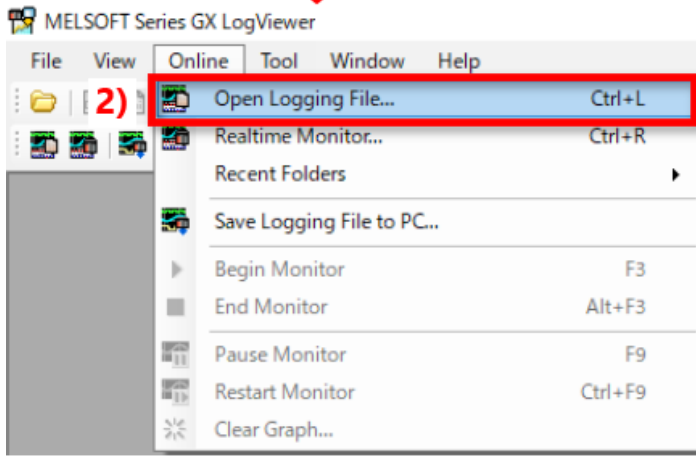
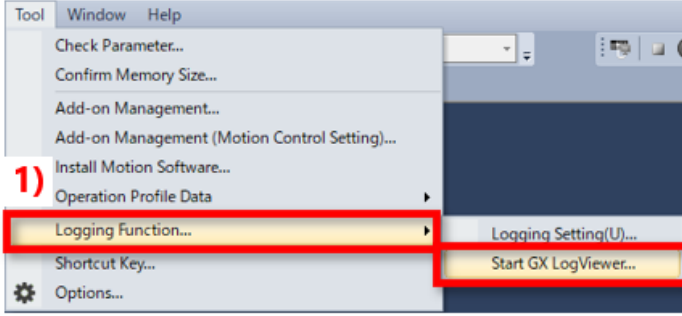
- Logging Status: Waiting for trigger
- Logging Status: Triggered
- Logging Status: Saving
- Logging Status: CollectionCompleted

Günlük kayıt verilerini okumak için GX LogViewer kullanılır.

Motion Kontrol Ayar Fonksiyonu ekranının araç çubuğundan [Tool] → [Logging Function] → [Start GX LogViewer] seçin.

GX LogViewer başlatıldığında, [Online] → [Open Logging File] seçin.

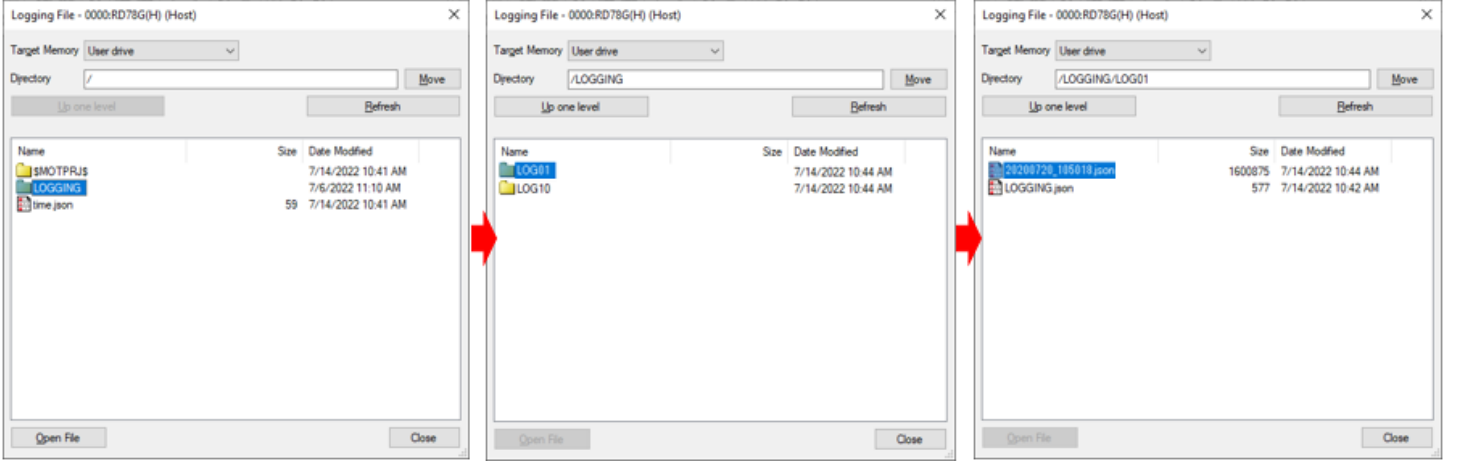
Bağlantı Hedefi ekranında "0000:RD78G(H) (Host)" seçin. (Not)



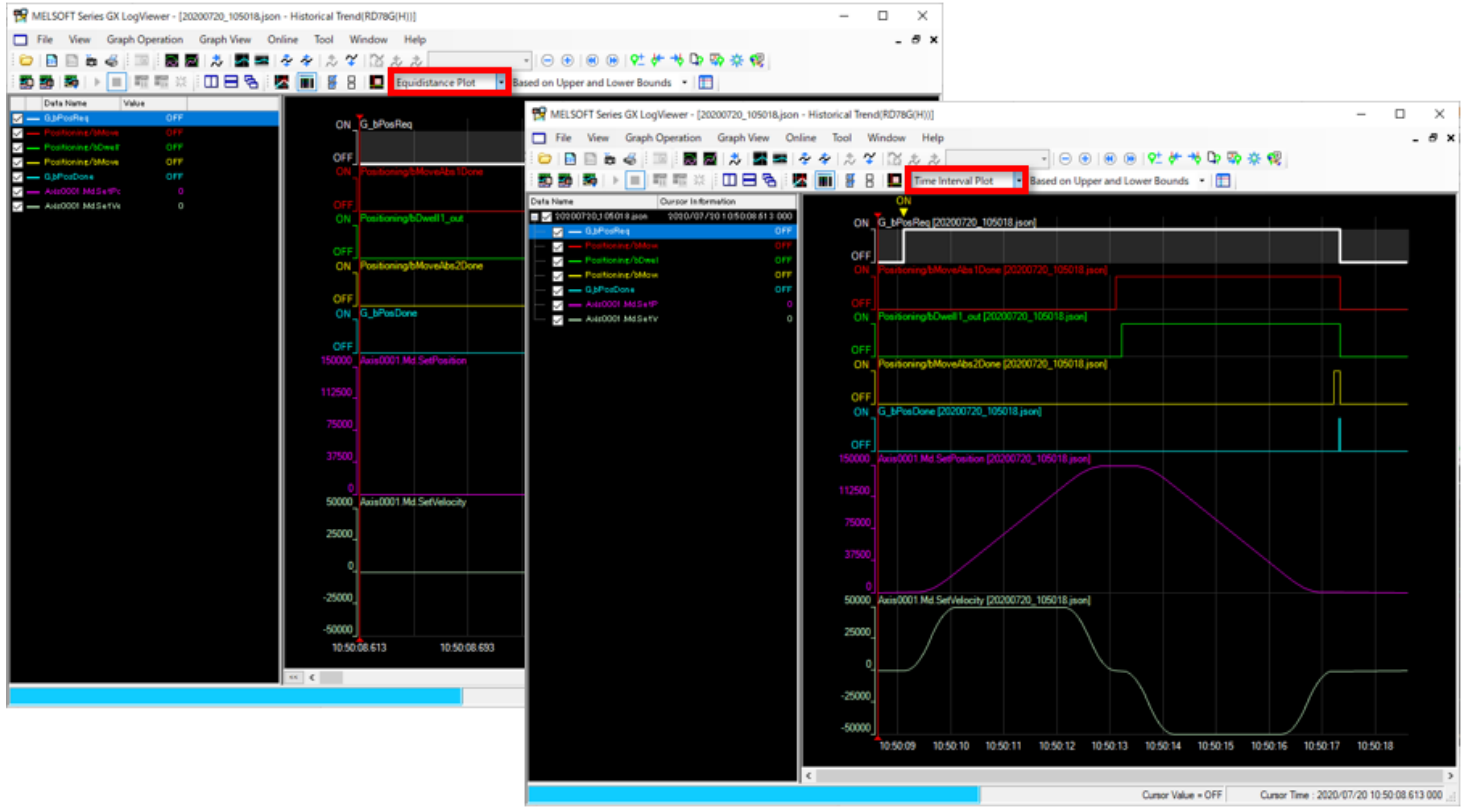
(Not) GX LogViewer daha önceden başlatıldıysa ve Motion modülü ile haberleşme zaten ayarlanmışsa, bu ekran görüntülenmez.

Okunacak günlük kayıt dosyasını seçin.

Bu bölüm örneğinde, "LOGGING" → "LOG1" → "(Logged date and time).json" kısmındaki kullanıcı sürücüsü seçilir. Dosya adını seçin ve [Open File] düğmesini tıklayın.



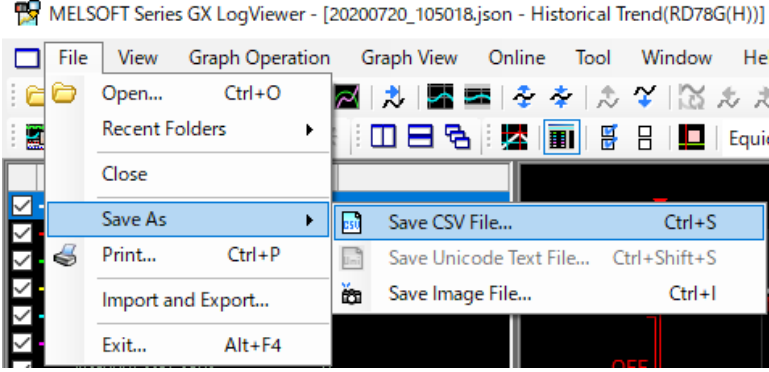
GX LogViewer'da kaydedilen dalga formu verileri görüntülenir. Çizim biçimi "Equidistance Plot" iken "Time Interval Plot" olarak değiştirilirse, kaydedilen dalga formunun tamamı görüntülenebilir.



Günlüğe kaydedilen dalga formu verileri, bir csv dosyası veya json dosyası olarak kaydedilebilir. (CSV biçiminde günlüğe alındığında, CSV dosyası olarak kaydedilebilir.)

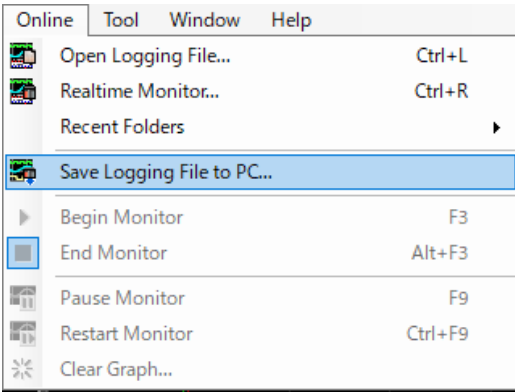
1) Csv dosyası olarak kaydederken

GX LogViewer araç çubuğundan [File] → [Save As] → [Save CSV File] seçin.



2) Json dosyası olarak kaydederken

GX LogViewer araç çubuğundan [Online] → [Save Logging File to PC] seçin.



Bu bölümde, şunları öğrendiniz:

- Günlük Kayıt Konfigürasyon Aracının Başlatılması
- Günlüğe Kaydedilecek Verileri Ayarlama
- Günlük Kayıt Ayarını Yazma
- Günlük Kaydını Başlatma
- Günlük Kayıt Verilerini Okuma
- Günlük Kayıt Verilerinin Kaydedilmesi

Nokta

Günlük Kayıt Konfigürasyon Aracının Başlatılması	<ul style="list-style-type: none">• Motion kontrol ayar fonksiyonundan motion sistemi günlük kayıt ayarlama aracını başlatın.
Günlüğe Kaydedilecek Verileri Ayarlama	<ul style="list-style-type: none">• Motion sistemi günlük kaydı ayarlama aracını izleyerek günlüğe kaydedilecek verileri, tetikleyici koşulları ve diğerlerini ayarlayın.
Günlük Kayıt Ayarını Yazma	<ul style="list-style-type: none">• Günlük kaydıdan önce günlük kaydı ayarlama verilerini Motion modülüne yazın.
Günlük Kaydını Başlatma	<ul style="list-style-type: none">• Günlük kaydı başlatma koşulu "Start by User Operation" olarak ayarlandığında, günlük kaydına başlamak için "Logging Status and Operation" ekranındaki başlatma düğmesini tıklayın.
Günlük Kayıt Verilerini Okuma	<ul style="list-style-type: none">• Günlük kayıt verilerini okumak için GX LogViewer kullanılır.
Günlük Kayıt Verilerinin Kaydedilmesi	<ul style="list-style-type: none">• Günlüğe kaydedilen dalga formu verileri, bir csv dosyası veya json dosyası olarak kaydedilebilir.

Açık labele ilgili doğru açıklamayı/açıklamaları seçin. (Birden fazla cevap seçebilirsiniz.)

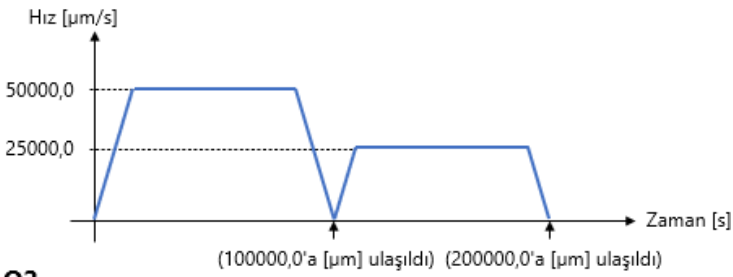
- Açık label, hem Motion modülü hem de PLC CPU'da kullanılabilen ortak labeldir.
- Açık label PLC CPU genel labelinden kaydedilir.
- Genel label, açık labele ayarlandığında, labelin PLC CPU'dan okunacağı mı, yoksa PLC CPU'ya yazılacağı mı seçilir.

İlk olarak FB1 yürütülür ve ardından FB2 yürütülür.

FB1 ve FB2'nin hedef pozisyonu ve hedef hızı aşağıdaki tabloda gösterildiği gibi olduğunda, sonrasında gerçekleştirilen ara bellek modunu seçin.

	Hedef pozisyon	Hedef hız
FB1	100000,0 [μm]	50000,0 [$\mu\text{m/s}$]
FB2	200000,0 [μm]	25000,0 [$\mu\text{m/s}$]

Q1



Q1

Uygun cevabı seçin.

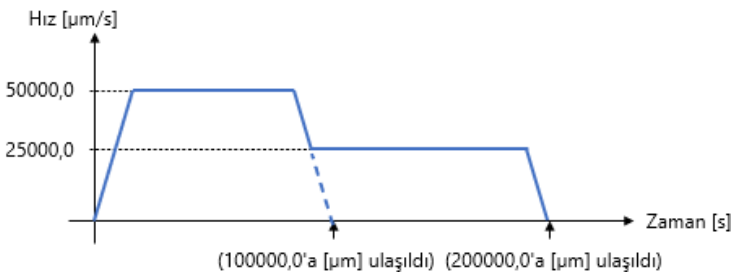


Q2

Uygun cevabı seçin.



Q2



Q1: 1: mcAborting
2: mcBuffered
3: mcBlendingNext
4: mcBlendingPrevious

Q2: 1: mcBlendingNext ve mcBlendingHigh
2: mcBlendingPrevious ve mcBlendingHigh
3: mcBlendingNext ve mcBlendingLow
4: mcBlendingPrevious ve mcBlendingLow

PLC CPU ile programlama için aşağıdakiler arasından doğru cevabı/cevapları seçin. (Birden fazla cevap seçebilirsiniz.)

PLC CPU'da Motion modülü için Motion control FB'yi kullanmak için FB kitaplığının GX Works3'e kaydedilmesi gerekir.

Motion control FB'yi GX Works3'ün proje ağaç şemasındaki program düzenleyiciye yerleştirin.

Motion modülü için ayarlanacak parametre yoktur.

Boşlukları doldurmak için uygun cevapları seçin.

- Günlüğe kaydedilecek verileri ayarlamak için (Q1) başlatın.
- Günlük kaydı yapmak için günlük kaydı verilerini (Q2) üzerine yazın.
- (Q3) günlük kayıt verilerini okumak ve dalga formunu kontrol etmek için kullanılır.

Q1

Uygun cevabı seçin.



Q2

Uygun cevabı seçin.



Q3

Uygun cevabı seçin.



Q1: 1: CPU module logging configuration tool
2: Motion sistemi günlük kaydı ayarlama aracı

Q2: 1: CPU
2: Motion modülü
3: Servo sürücü

Q3: 1: MR Configurator2
2: GX LogViewer

Açık labele ilgili doğru açıklamayı/açıklamaları seçin. (Birden fazla cevap seçebilirsiniz.)

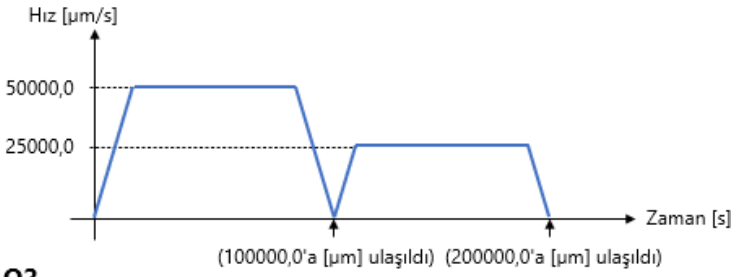
- Açık label, hem Motion modülü hem de PLC CPU'da kullanılabilen ortak labeldir.
- Açık label PLC CPU genel labelinden kaydedilir.
- Genel label, açık labele ayarlandığında, labelin PLC CPU'dan okunacağı mı, yoksa PLC CPU'ya yazılacağı mı seçilir.

İlk olarak FB1 yürütülür ve ardından FB2 yürütülür.

FB1 ve FB2'nin hedef pozisyonu ve hedef hızı aşağıdaki tabloda gösterildiği gibi olduğunda, sonrasında gerçekleştirilen ara bellek modunu seçin.

	Hedef pozisyon	Hedef hız
FB1	100000,0 [μm]	50000,0 [$\mu\text{m/s}$]
FB2	200000,0 [μm]	25000,0 [$\mu\text{m/s}$]

Q1



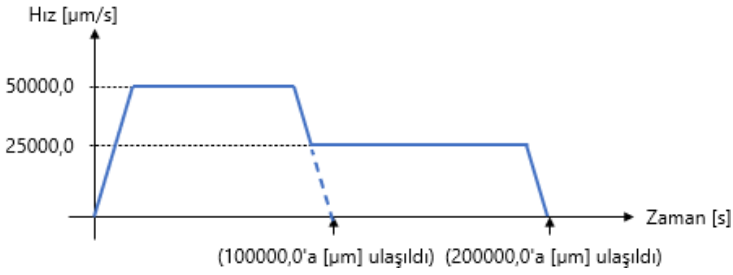
Q1

2: mcBuffered

Q2

3: mcBlendingNext ve mcBlendingLow

Q2



Q1: 1: mcAborting
2: mcBuffered
3: mcBlendingNext
4: mcBlendingPrevious

Q2: 1: mcBlendingNext ve mcBlendingHigh
2: mcBlendingPrevious ve mcBlendingHigh
3: mcBlendingNext ve mcBlendingLow
4: mcBlendingPrevious ve mcBlendingLow

PLC CPU ile programlama için aşağıdakiler arasından doğru cevabı/cevapları seçin. (Birden fazla cevap seçebilirsiniz.)



PLC CPU'da Motion modülü için Motion control FB'yi kullanmak için FB kitaplığının GX Works3'e kaydedilmesi gerekir.



Motion control FB'yi GX Works3'ün proje ağaç şemasındaki program düzenleyiciye yerleştirin.



Motion modülü için ayarlanacak parametre yoktur.

Boşlukları doldurmak için uygun cevapları seçin.

- Günlüğe kaydedilecek verileri ayarlamak için (Q1) başlatın.
- Günlük kaydı yapmak için günlük kaydı verilerini (Q2) üzerine yazın.
- (Q3) günlük kayıt verilerini okumak ve dalga formunu kontrol etmek için kullanılır.

Q1

2: Motion sistemi günlük kaydı ayarlama aracı



Q2

2: Motion modülü



Q3

2: GX LogViewer



Q1: 1: CPU module logging configuration tool
2: Motion sistemi günlük kaydı ayarlama aracı

Q2: 1: CPU
2: Motion modülü
3: Servo sürücü

Q3: 1: MR Configurator2
2: GX LogViewer

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	✓	✓	✓							

Toplam soru: **7**

Doğru cevaplar: **7**

Yüzde: **100 %**

Temizle

**"MELSEC iQ-R Serisi Motion Modülü Uygulaması (RD78G(H)/
Pozisyonlama Kontrolü)" 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