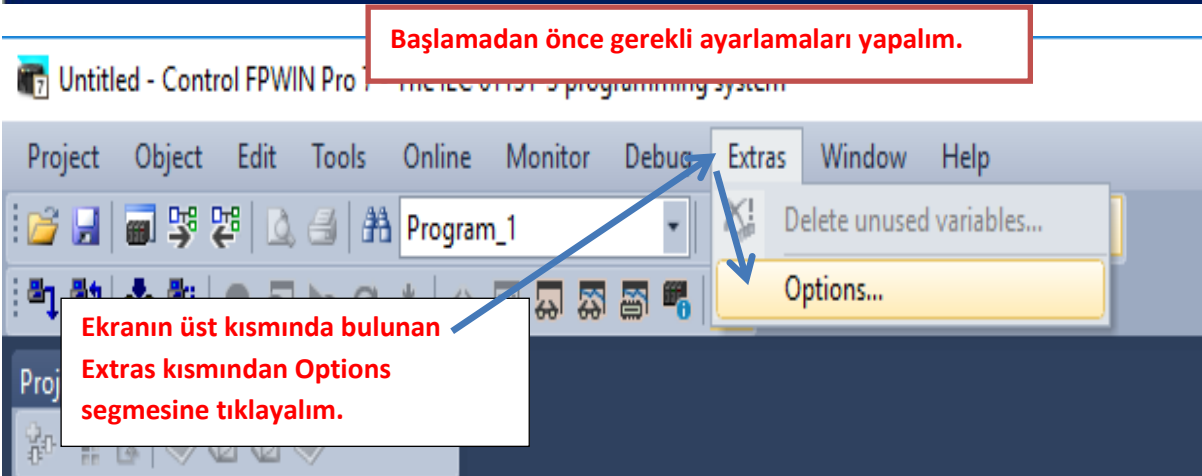


SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

FPWINPRO_7 Servo motor linear fonksiyon bloğu

(LINEAR FONKSİYON BLOĞU: Bu komutun kullanım amacı, iki tane servo motoru aynı blok üzerinden eş zamanlı olarak kontrol etmektir. Örneğin; başlangıçta farklı konumlarda olan iki motorumuz olsun bu motorların aynı zamanda aynı yerde olmasını istersek bu komutu kullanabiliriz ve buna benzer uygulamalarda kullanılabilir bir diğer örnek olarak da dikdörtgen, kare, üçgen vb. hareket uygulamalarında kullanılabilir.)



SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

Options

×

Array elements to show 10

Number of recent projects on project menu 50

Hide start page if project is open

Automatically save the project information in the PLC after downloading the program code and PLC configuration, if configuration memory is available and activated

Automatically change to PROG mode, upload the PLC configuration, compile, and change to RUN mode while downloading the program code and PLC configuration

Automatically start monitoring of programs if the project is consistent

Ask for download of the program code and PLC configuration after going online if the project is not consistent

OK Cancel Default Apply Help

Daha sonra çıkan pencerenin sol tarafında bulunan Program options kısmında General tuşundan bu iki segme işaretlenir.

Options

×

Initialization of retain variables

Initialize all retain variables by program code. Recommended default.

Initialize all retain variables by PLC configuration. Recommended default.

Keep values of global retain variables which have been assigned an explicit address. Initialize all other retain variables by program code.

Default string length (characters) 32

Indexed function block instantiation

Path\file name to store compile files

Stop check at number of errors 20

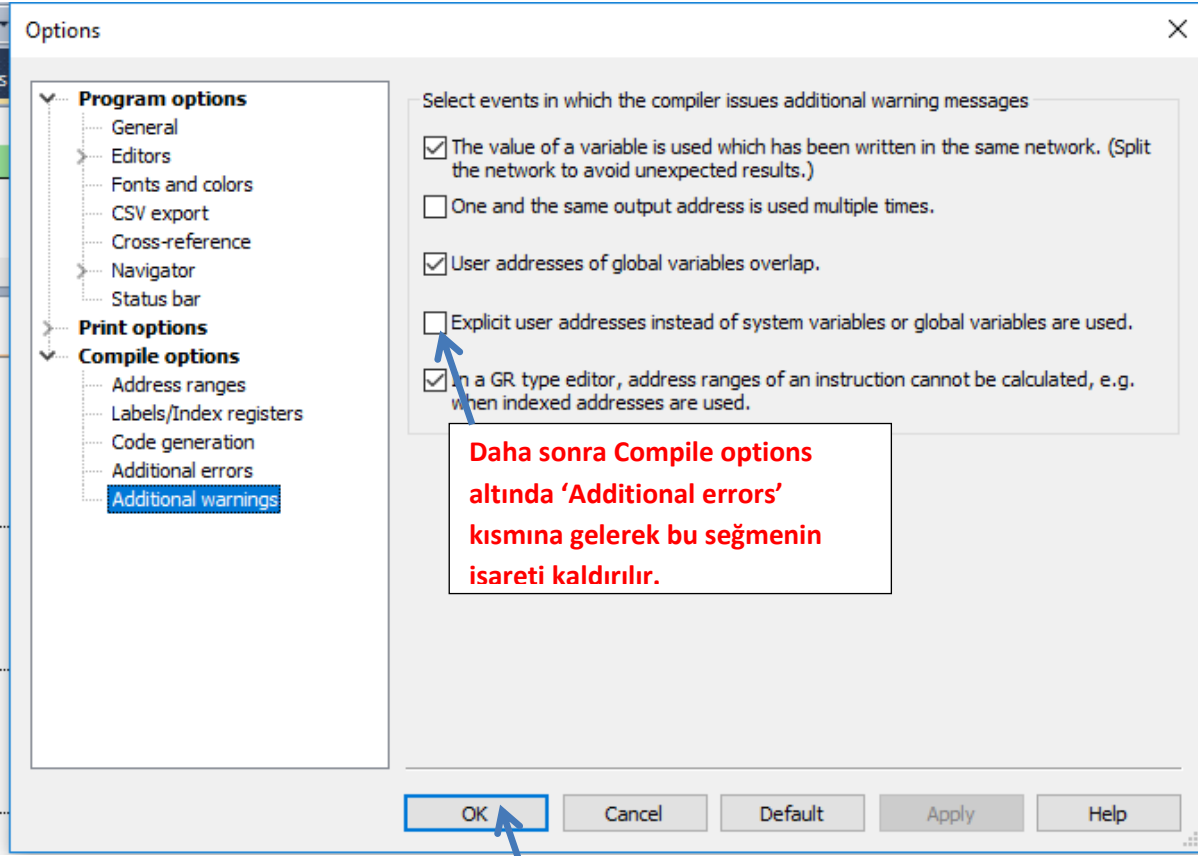
Stop check at number of warnings 20

OK Cancel Default Apply Help

PLC de kalıcı adreslerdeki değerlerin her program yüklenmesinde silinmemesini sağlar.

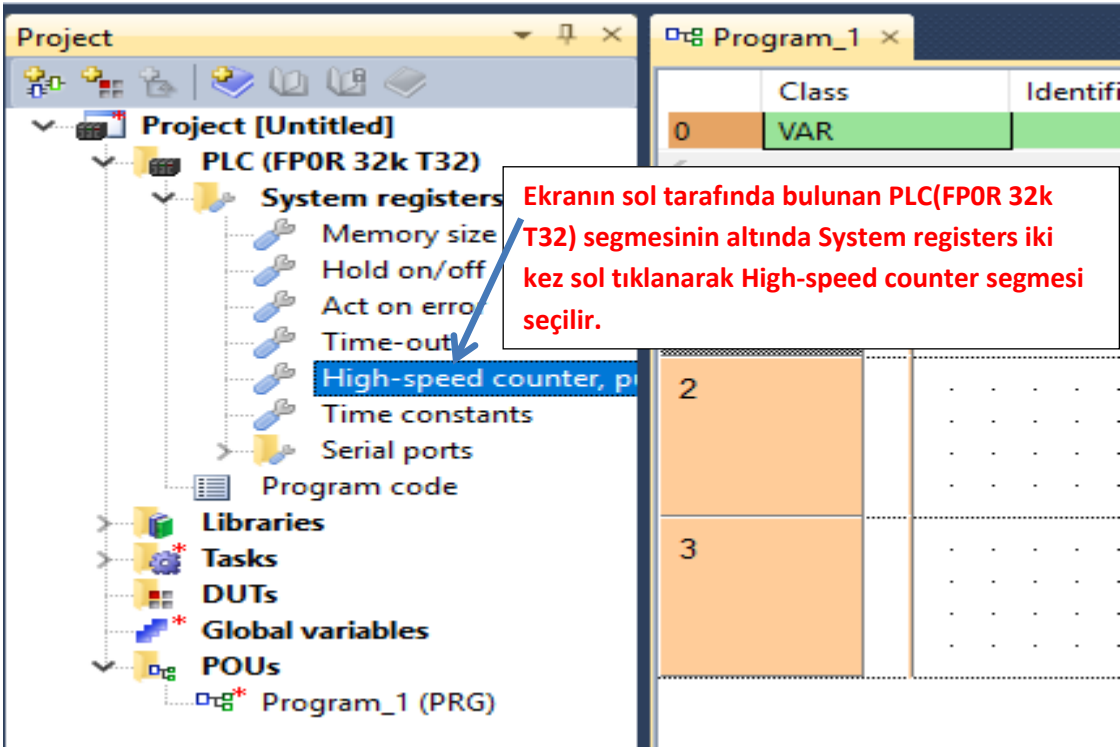
Daha sonra Compile options altında 'Code generation' kısmına gelerek bu segme de işaretlenir.

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ



Yapılan bu ayarlamalardan sonra Ok deyip çıkalım.

Kullanacağımız pulse çıkış ayarlarını yapalım



SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

Böyle bir pencere açılacaktır.

İki motor kullanacağımız için iki tane farklı kanal seçimi yapmalıyız.

0.kanalı seçelim.

Burada ki X0 ise JOG1 bloğunda kullanılır.

Burada ki X4 ile PLC home ye gönderilebilir.

Buradaki kanal seçimimizde baştaki 3 tanesinden birisi seçilebilir. Biz en üstekini tercih ettik.

404/405	Interrupt input: X0 -> Interrupt 0	Unused	Unused	Specifies input as interrupt trigger.
404/405	Interrupt input: X1 -> Interrupt 1	Unused	Unused	Specifies input as interrupt trigger.
404/405	Interrupt input: X2 -> Interrupt 2	Unused	Unused	Specifies input as interrupt trigger.
404/405	Interrupt input: X3 -> Interrupt 3	Unused	Unused	Specifies input as interrupt trigger.
404/405	Interrupt input: X4 -> Interrupt 4	Unused	Unused	Specifies input as interrupt trigger.
404/405	Interrupt input: X5 -> Interrupt 5	Unused	Unused	Specifies input as interrupt trigger.
404/405	Interrupt input: X6 -> Interrupt 6	Unused	Unused	Specifies input as interrupt trigger.
404/405	Interrupt input: X7 -> Interrupt 7	Unused	Unused	Specifies input as interrupt trigger.

1.kanalı seçelim.

Böylece 2 farklı kanal seçimi yapmış olduk.

High-speed coun...errupt input • Program_1

No	Item name	Data
400	High-speed counter: Channel 0	Unused
400	High-speed counter: Channel 1	Unused
400	High-speed counter: Channel 2	Unused
400	High-speed counter: Channel 3	Unused
401	High-speed counter: Channel 4	Unused
401	High-speed counter: Channel 5	Unused
402	Pulse output: Channel 0	Pulse output (Y0-Y1)
402	Pulse output: Channel 1	Pulse output (Y2-Y3)
402	Pulse output: Channel 2	Unused
402	Pulse output: Channel 3	Unused
403	Pulse-catch input: X0	Pulse output (Y2-Y3), home input (X5)
403	Pulse-catch input: X1	Pulse output (Y2-Y3), home input (X5), position control trigger input (X1)
403	Pulse-catch input: X2	PWM output (Y2)
403	Pulse-catch input: X3	Disable
403	Pulse-catch input: X4	Disable
403	Pulse-catch input: X5	Disable
403	Pulse-catch input: X6	Disable
403	Pulse-catch input: X7	Disable

Seçtikten sonra buradaki çarpı işaretine tıkladığımızda kaydedilsin mi diye sorar. Evet diyerek kapatabiliriz.

Save

Save changes of "High-speed counter, pulse-catch input, interrupt input"?

Evet Hayır İptal

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

Ekranımız bu şekilde açılacaktır.

The screenshot displays the SAVIOR software interface. The main window is titled "Program_1" and contains a table with the following columns: Class, Identifier, Type, Initial, and Comment. The table has four rows, with the first row highlighted in green and containing the text "VAR". The right side of the interface features an "Instructions" panel with a "Category" dropdown menu and "Filter settings: <ALL> / <ALL> / <ALL>". Below this is a list of instruction categories, including "Communication instructions", "Comparison instructions", "Conversion instructions", "Counter instructions", "Data table instructions", "Date and time instructions", "Edge detection instructions", "GT panel instructions", "High-speed counter instruct", "Input, output and unit acces", "Memory device instructions", "Pointer instructions", "Process control instructions", "Program execution control i", "Pulse output instructions", "Selection instructions", "SFC control instructions", "Signal processing instructor", "Size information instructions", "Special instructions", "String instructions", "System register instructions", and "Timer instructions".

Ekranın sağ tarafında bulunan Instructions segmesinden bu kısma mouse ile tek tek tıklayarak Category seçilir.

Daha sonra Pulse output instructions içine girilir.

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

Sağdaki gibi bir ekran açılacaktır.

Biz Linear_FB modunu kullanacağımız için bu segmeyi mouse ile tutup ekrana sürükleyip bırakalım.

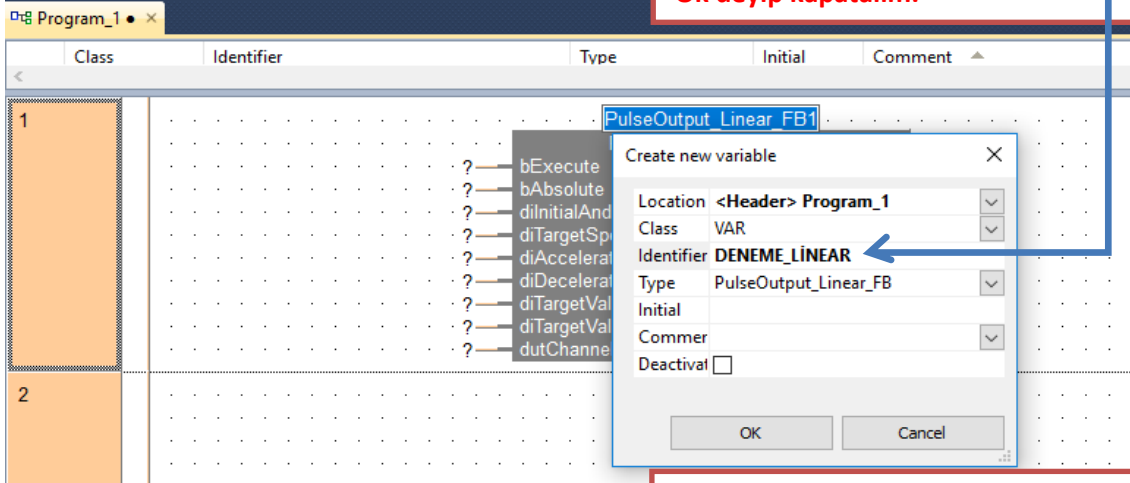
Görüldüğü gibi ekranımıza bu şekilde gelecektir.

PulseOutput_Linear_FB

?	bExecute	bError	?
?	bAbsolute	rInitialAndFinalSpeed_X	?
?	dInitialAndFinalSpeed	rTargetSpeed_X	?
?	dTargetSpeed	rInitialAndFinalSpeed_Y	?
?	dAccelerationTime	rTargetSpeed_Y	?
?	dDecelerationTime	dutAdditionalOutputs	?
?	dTargetValue_X		
?	dTargetValue_Y		
?	dutChannelConfiguration_X_Y		

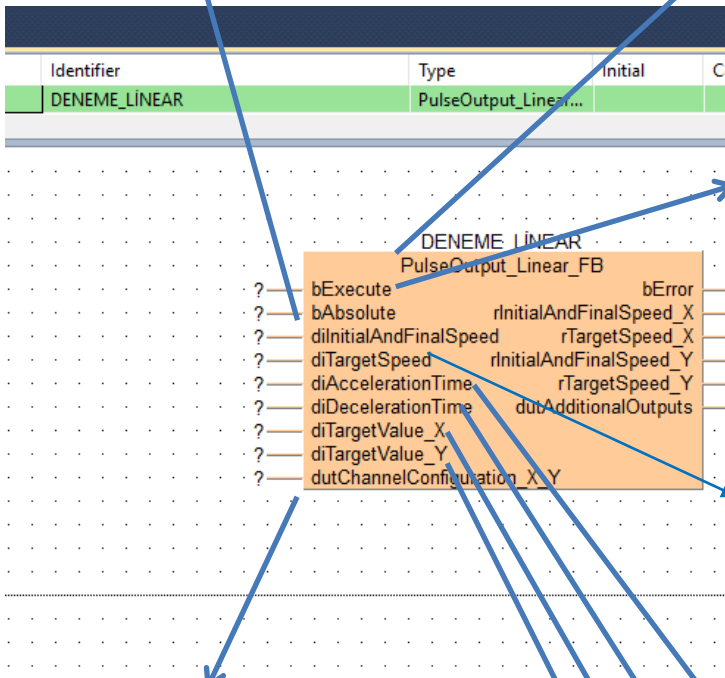
SAVİOR OTOMASYON TEKNİK DESTEK BİDİMİ

Daha sonra görüldüğü gibi herhangi bir isim verelim.
Ok deyip kapatalım.



Başlangıç ve bitiş hız değeri atanır. Maximum ve minimum değerlerine dikkat edilmelidir. Kullanacağımız bacak üstüne mouse ile geldiğimizde max-min aralığını görebiliriz. Buradaki hız değeri belli bir değerde sabitlenmeli ve Target Speed hızından küçük olmalıdır.

Şimdi ise Linear bloğu üzerinde bulunan parametreleri açıklayalım.



Aktif olduğu bloğumuz da aktiftir.

False olur veya bu bacak slinirse pals çıkışı Incremental Modda çalışır. True olursa Absolute modda çalışır. Incremental modda her tetikte target değeri kadar gidilir. Absolute modda ilk tetikte target kısımdaki değer kadar gidilir. İkinci tetik de çıkış üretilmez, mevcut Target değerinden daha büyük veya daha küçük değer yazılarak servonun ileri ya da geri gitmesi sağlanır.

Hedef hız buraya yazılır. Blok üstüne mouse ile geldiğimizde max-min aralığını görebiliriz. Bu kısma girdiğimiz hız değeri baz alınarak bloğun sağ tarafında bulunan TargetSpeed hızları ve InitialAndFinalSpeed hızları PLC tarafından belirlenir. Bu hızlar da girilen TargetValue değerlerine göre birbirine farklılıklar gösterir.

Kullanacağımız Data Unity Type adresini gösterir. Dut ile ilgili bütün ayarlamaları buradan yapabiliriz. Bu kısma belirlediğimiz bir isim verebiliriz.

Başlarken hızlanma rampa süresi. Blok üstüne mouse ile geldiğimizde max-min aralığını görebiliriz.

Motor yavaşlama rampa süresi. Blok üstüne mouse ile geldiğimizde max-min aralığını görebiliriz.

Hedef değer olarak ifade edilir. Motorumuz buraya girdiğimiz değer kadar gidip durmaktadır. Kanal_0(X)

Hedef değer olarak ifade edilir. Diğer motorumuz da buraya girdiğimiz değer kadar gidip durmaktadır. Kanal_1(Y)

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

Buraya herhangi bir karakter yazdığımızda PLC bunu Real formatta atamaktadır. Bizim girdiğimiz kalkış rampa değerine göre PLC kendisi belirlemektedir.

Bool olarak bir çıkış atanabilir.

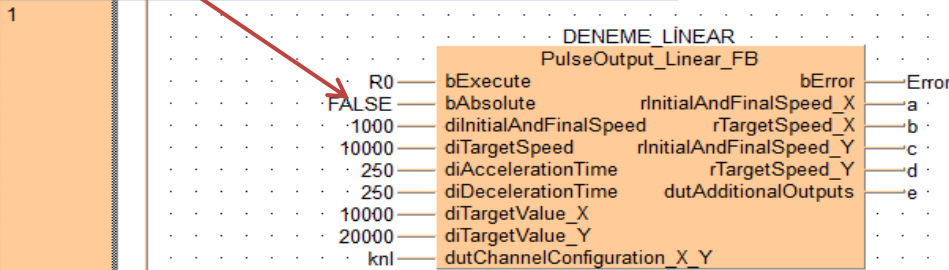
Buraya herhangi bir karakter yazdığımızda PLC bunu Real formatta atamaktadır. Bizim girdiğimiz kalkış rampa değerine göre PLC kendisi belirlemektedir.

Buraya herhangi bir karakter yazdığımızda PLC bunu Real formatta atamaktadır. Girilen TargetSpeed ve TargetValue değerine göre PLC kendisi buraya değer atamaktadır.

Buraya herhangi bir karakter yazdığımızda PLC bunu Real formatta atamaktadır. Girilen TargetSpeed ve TargetValue değerine göre PLC kendisi buraya değer atamaktadır.

Şimdi gerekli parametreleri girelim ve Servo-On çıkışını bağlayalım.

TRUE olduğunda geri çalışma aktif yapılır.



Y2 çıkışı Servo-On.

R9010 butonu ile yapılır. Daima TRUE durumdadır.

Bloğumuzda girilen False, True gibi ifadeleri büyük veya küçük harflerle yazabiliriz. Bir sorun oluşturmaz.

Görüldüğü gibi parametrelerimiz girilmiştir.

Daha sonra Linear bloğu ile ilgili diğer ayarlara bakalım.

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

Global variables • Program_1 • x

Class	Identifier	Type	Initial	Comment
VAR	knl	PulseOutput_Channel		

Programın üst kısmında bulunan 'knl' diye atadığımız kısımdan Dut ayarları için Initial içine girelim.

DENEME_LINEAR
PulseOutput_Linear_FB

R0 — bExecute bEr
FALSE — bAbsolute rInitialAndFinalSpeed
1000 — diInitialAndFinalSpeed rTargetSpeed
10000 — diTargetSpeed rInitialAndFinalSpeed_Y
250 — diAccelerationTime rTargetSpeed_Y
250 — diDecelerationTime dutAdditionalOutputs
10000 — diTargetValue_X
20000 — diTargetValue_Y
knl — dutChannelConfiguration_X_Y

sys_bTrue R9010: Normally TRUE Y2 Servo On

Data unit initial values

Bu şekilde ekran açılacaktır.

Data unit type:
PulseOutput_Channel_Configuration_I

Channel	INT	0	FP-SIGMA:	0, 2	FP-X/NHR:	0, 1	FP-X/NH 16K C14T:	0, 1, 2,	FP-X/NH 32K C30T, C60T:	0, 1, 2, 3	FP-X/NH 32K C60T: +4
bOutput_Pulse_ForwardTrue	BOOL	FALSE	if neither bOutput_Pulse_ForwardTrue nor bOutput_Pulse_ForwardFalse: Forward or reverse pulses (FP0, FP-e: Only pulses)								
bOutput_Pulse_ForwardFalse	BOOL	FALSE	if neither bOutput_Pulse_ForwardTrue nor bOutput_Pulse_ForwardFalse: Forward or reverse pulses (FP0, FP-e: Only pulses)								
bAccelerationSteps60	BOOL	TRUE	FP-SIGMA, FP-X: Number of acceleration/deceleration steps: 60 (else 30 steps)								
bDutyRatio25	BOOL	TRUE	FP-SIGMA, FP-X: Duty ratio (for pulse duration and period): 25% (else 50%)								
bFrequencyRange_48Hz_100kHz	BOOL	FALSE	FP-SIGMA, FP-X: Frequency range for initial and target speed: 48Hz-100kHz (if neither bFrequencyRange_48Hz_100kHz nor bFrequencyRange_191Hz_100kHz)								
bFrequencyRange_191Hz_100kHz	BOOL	TRUE	FP-SIGMA, FP-X: Frequency range for initial and target speed: 191Hz-100kHz (if neither bFrequencyRange_48Hz_100kHz nor bFrequencyRange_191Hz_100kHz)								
bPulseWidth80µs	BOOL	FALSE	FP0, FP-e Home, Trapezoidal: 80µs (else 50%)								
iDutyRatioIn10PercentSteps	INT	0	FP0, FP-e Jog: 1-9: duty ratio in increments of 10%, 0: fixed pulse width of 80µs								
bEnableHomeOnlyAfterNearHomeDeceleration	BOOL	FALSE	FPOR: Type 1 (else type 0), FP-SIGMA, FP-X: Type 2 (else type 1)								
iHomeInputDeviationCounterClearSignalOutputTime_in_0p5ms	INT	0	FPOR, FP-SIGMA, FP-X: 0 to 200 [x0.5ms]								
bCalculationOnly	BOOL	FALSE	FPOR: Jog, Trapezoidal: Output operation calculation only (else pulse output)								
bTrapezoidalMaximumTargetSpeed50kHz	BOOL	FALSE	FPOR: Output operation: Type 1: The target speed can be up to the maximum speed 50kHz, acceleration and deceleration are determined by								
bExecuteInInterrupt	BOOL	FALSE	FPOR: Jog positioning, trapezoidal: Execute in or called from interrupt program (else in main program)								
bJogWithNoCounting	BOOL	FALSE	Only pulse outputs without counting, no target value match. FP-SIGMA, FP-X: bReverse=TRUE: Reverse pulses, bReverse=FALSE: Forward pulses								
bContinueAfterDone	BOOL	FALSE	FP-SIGMA circular pulse output: 0=Execution stops when target value has been reached, 1=Continue values can be set using 'sys_bisCircular'								

Buradaki özellikleri açıklayalım.

Channel => Kullanılan PLC modeline göre kanal sayısı değişebilir. Biz 0 ve 1. kanalı kullandığımız için görüldüğü gibi 0 değeri vardır.

bOutput_Pulse_ForwardTrue => Direction modda çalıştırıldığında ileri yön dijital çıkış için TRUE yapılır.

bOutput_Pulse_ForwardFalse => Direction modda çalıştırıldığında geri

Eğer bu komutlara hiç dokunmadan fabrika ayarında bırakırsak CW-CCW modunda çalıştırmış oluruz.

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

yön dijital çıkış için TRUE yapılır.

bAccelerationSteps60 => FP-X ve FP-Σ PLC modellerinde kalkış rampasında ki adım aralığını gösterir.TRUE olduğunda 60 FALSE olduğunda 30 olur.(Fabrika ayarında bırakabiliriz)

bDutyRatio25 => FP-X ve FP-Σ PLC modellerinde pulse çıkış PWM çalışma oranını gösterir. Eğer bu şekilde fabrika ayarında bırakırsak pals oranının %25 lik kısmı duty olur.False yaparsak pals oranının %50 lik kısmı duty olur. (Fabrika ayarında bırakabiliriz)

bFrequencyRange_48Hz_100kHz => FP-X ve FP- Σ PLC modellerinde TRUE yapıldığında 48Hz ile 100kHz arasında çalışmaya izin verir. Bu komutu TRUE yapıp 48Hz den düşük değer girildiğinde hata verecektir.

bFrequencyRange_191Hz_100kHz => FP-X ve FP- Σ PLC modellerinde TRUE yapıldığında 191Hz ile 100kHz arasında çalışmaya izin verir.Bu komutu TRUE yapıp 191Hz den düşük değer girildiğinde hata verecektir.

bPulseWidth80µs => FP0 ve FP-e modellerinde Home ve Trapezoidal kullanırken Pulse genişliği çalışma süresi 80µs kadardır.Eğer TRUE yaparsak 40 µs olur. .(Fabrika ayarında bırakabiliriz)

iDutyRatioIn10PercentSteps => FP0 ve FP-e modellerinde JOG bloğunda kullanılır.0 girildiğinde pulse genişliği 80 mikrosaniyeye sabitlenir.(Fabrika ayarında bırakabiliriz)

bEnableHomeOnlyAfterNearHomeDeceleration =>Home bloğu ile ilgilidir. Eğer bu komutu aktif edersek servo motorumuz X4 girişini görse bile durmaz. diCreepSpeed komutunu kullanarak motor yavaşladıktan sonra X4 fiziksel girişini görüp home tamamlanmaktadır. (Fabrika ayarında bırakabiliriz)

iHomeInputDevitionCounterClearSignalOutputTime_in_0p5ms => Bu komutu kullanmak için kullandığımız kanala göre PLC çıkışları vardır.Bunlar:CH0: Y8, CH1 : Y9, CH2: YA, CH3:YB bu şekildedir.Örneğin 0. Kanalı kullanıyorsak Y8 çıkışına fiziksel olarak sürücümüzün ilgili yerine bağlamalıyız. Girdiğimiz sürenin yarısı kadar zamanda silme işlemini gerçekleştirir .(Fabrika ayarında bırakabiliriz)

bCalculationOnly => FP0R PLC modelinde Jog ve Trapezoidal modda geçerlidir.Sadece hesaplama işlemlerinde kullanılır.Bu mod sadece PulseOutput(Y0-Y1) kanal girişinde çalışmaktadır. (Fabrika ayarında bırakabiliriz)

bTrapezoidalMaximumTargetSpeed50Hz => Bu komut Trapezoidal modda çalıştırılır.Bu özelliği TRUE yaptığımızda 'Execute' girişini kesmeden TargetSpeed'den hız ayarı değişikliği yapabiliriz.FALSE olarak bırakılırsa girdiğimiz ilk hız değeri ile motor hedefine varır.(Fabrika ayarında bırakabiliriz.)

bExecuteInInterrupt =>Eğer interrupt içinde trapezoidal veye jog pozisyon modu kullanacaksak bu komutu aktif etmeliyiz. (Fabrika ayarında bırakabiliriz.)

bJogWithNoCounting => Eğer PLC cihazımızın kustuğu puls değerini göstermek istersek bu girişi aktif etmeliyiz. Örneğin 0.kanal için DDT90400 adresinden gösterilebilir. (Fabrika ayarında bırakabiliriz).

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

bContinueAfterDone => FP- Σ PLC modelinde dairesel enterpolasyon kullanıldığı durumlarda kullanılır.
(Fabrika ayarında bırakabiliriz)

NOT: BURADA GİRİLEN TRUE-FULSE ifadeleri büyük veya küçük harf ile yazılabilir. Herhangi bir sorun oluşturmaz.

Data unit initial values

Data unit type:
PulseOutput_Channel_Configuration_I

iChannel	INT	0	FP-SIGMA: 0, 2 FP-X\NHR: 0, 1 FP-X\NH 16K C14T: 0, 1, 2, FP-X\NH 32K C30T, C60T: 0, 1, 2, 3 FP-X\NH 32K C60T: + 4, 5 FPOR: 0, 1, 2, 3 FP0:
bOutput_Pulse_ForwardTrue	BOOL	FALSE	if neither bOutput_Pulse_ForwardTrue nor bOutput_Pulse_ForwardFalse: Forward or reverse pulses (FP0, FP-e: Only pulses)
bOutput_Pulse_ForwardFalse	BOOL	FALSE	if neither bOutput_Pulse_ForwardTrue nor bOutput_Pulse_ForwardFalse: Forward or reverse pulses (FP0, FP-e: Only pulses)
bAccelerationSteps60	BOOL	TRUE	FP-SIGMA, FP-X: Number of acceleration/deceleration steps: 60 (else 30 steps)
bDutyRatio25	BOOL	TRUE	FP-SIGMA, FP-X: Duty ratio (for pulse duration and period): 25% (else 50%)
bFrequencyRange_48Hz_100kHz	BOOL	FALSE	FP-SIGMA, FP-X: Frequency range for initial and target speed: 48Hz-100kHz (if neither bFrequencyRange_48Hz_100kHz nor bFrequencyRange_191Hz_100kHz: 1.5Hz-9.8kHz)
bFrequencyRange_191Hz_100kHz	BOOL	TRUE	FP-SIGMA, FP-X: Frequency range for initial and target speed: 191Hz-100kHz (if neither bFrequencyRange_48Hz_100kHz nor bFrequencyRange_191Hz_100kHz: 1.5Hz-9.8kHz)
bPulseWidth80µs	BOOL	FALSE	FP0, FP-e Home, Trapezoidal: 80µs (else 50%)
iDutyRatioIn10PercentsSteps	INT	0	FP0, FP-e Jog: 1-9: duty ratio in increments of 10%, 0: fixed pulse width of 80µs
bEnableHomeOnlyAfterNearHomeDeceleration	BOOL	FALSE	FPOR: Type 1 (else type 0), FP-SIGMA, FP-X: Type 2 (else type 1)
HomeInputDeviationCounterClearSignalOutputTime_in_0p5ms	INT	0	FPOR, FP-SIGMA, FP-X: 0 to 200 [x0.5ms]
bCalculationOnly	BOOL	FALSE	FPOR: Jog, Trapezoidal: Output operation calculation only (else pulse output)
bTrapezoidalMaximumTargetSpeed50kHz	BOOL	FALSE	FPOR: Output operation: Type 1: The target speed can be up to the maximum speed 50kHz, acceleration and deceleration are determined by the maximum speed 50kHz
bExecuteInInterrupt	BOOL	FALSE	FPOR Jog positioning, trapezoidal: Execute in or called from interrupt program (else in main program)
bJogWithNoCounting	BOOL	FALSE	Only pulse outputs without counting, no target value match. FP-SIGMA, FP-X: bReverse=TRUE: Reverse pulses, bReverse=FALSE: Forward pulses.
bContinueAfterDone	BOOL	FALSE	FP-SIGMA circular pulse output: 0=Execution stops when target value has been reached, 1=Continue values can be set using 'sys_bIsCircularInterpolationOverwritingPossible'



Şuan için herhangi bir özellik kullanmadığımdan hiçbirşeye dokunmadan ekranı kapatabiliriz.

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

The screenshot shows a PLC programming software interface. At the top, there is a table for global variables:

Class	Identifier	Type	Initial	Comment
VAR	kn1	PulseOutput_Chan...		

Below the table, there is a ladder logic diagram. The first step (1) shows a function block call for 'DENEME_LINEAR'. The function block is 'PulseOutput_Linear_FB'. The inputs and outputs are as follows:

- Inputs: R0 (bExecute), FALSE (bAbsolute), 1000 (diInitialAndFinalSpeed), 10000 (diTargetSpeed), 250 (diAccelerationTime), 250 (diDecelerationTime), 10000 (diTargetValue_X), 20000 (diTargetValue_Y), kn1 (dutChannelConfiguration_X_Y).
- Outputs: bError (Error), rInitialAndFinalSpeed_X (a), rTargetSpeed_X (b), rInitialAndFinalSpeed_Y (c), rTargetSpeed_Y (d), dutAdditionalOutputs (e).

The second step (2) shows a normally open contact labeled 'R9010: Normally TRUE' connected to a coil labeled 'Y2 Servo On'.

The screenshot shows the 'Monitor' menu of the PLC programming software. The menu is open, and the 'Special flags and registers' option is selected. A sub-menu is displayed, showing the following options:

- PLC status information
- Basic error messages
- Operating values
- Serial communication
- Ethernet communication
- Advanced Ethernet communication
- Analog values
- Clock/Calendar functions
- High-speed counters 0-5
- High-speed counters 6-10,A,B
- Pulse outputs**
- Step ladder
- PLC link 0 status
- PLC link 1 status
- PLC link settings

A blue arrow points from the 'Special flags and registers' option in the main menu to the 'Pulse outputs' option in the sub-menu. A text box on the right side of the image contains the following text:

Şimdi ise bu bloğu daha iyi anlamak için enkoder değerini gösterelim ekranın üst kısmında bulunan Monitör segmesinden Special flags and registers bölümünden Pulse outputs kısmına girelim.

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

Bu ekran açılacaktır.

ElapsedValue kısmında bulunan adres ile enkoder değerini gösterebiliriz.

Adres değerlerini öğrendikten sonra bu segmeden programımıza dönelim.

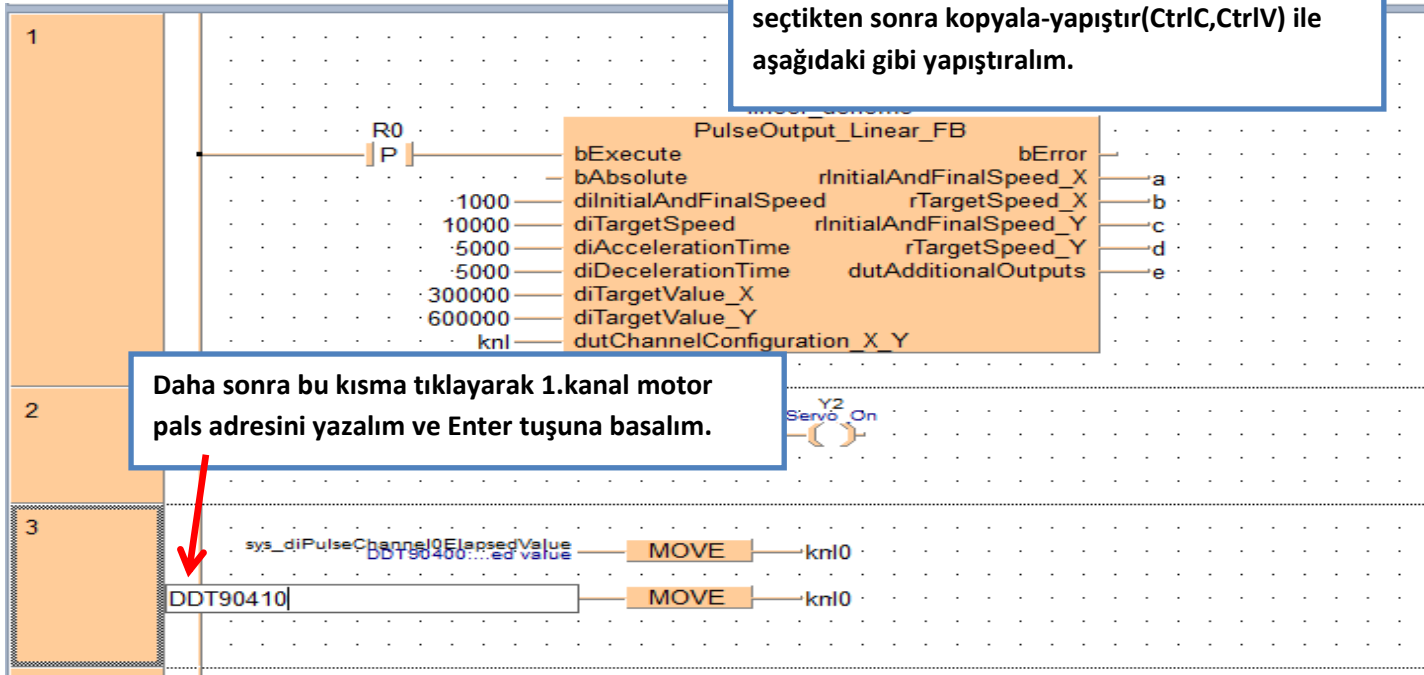
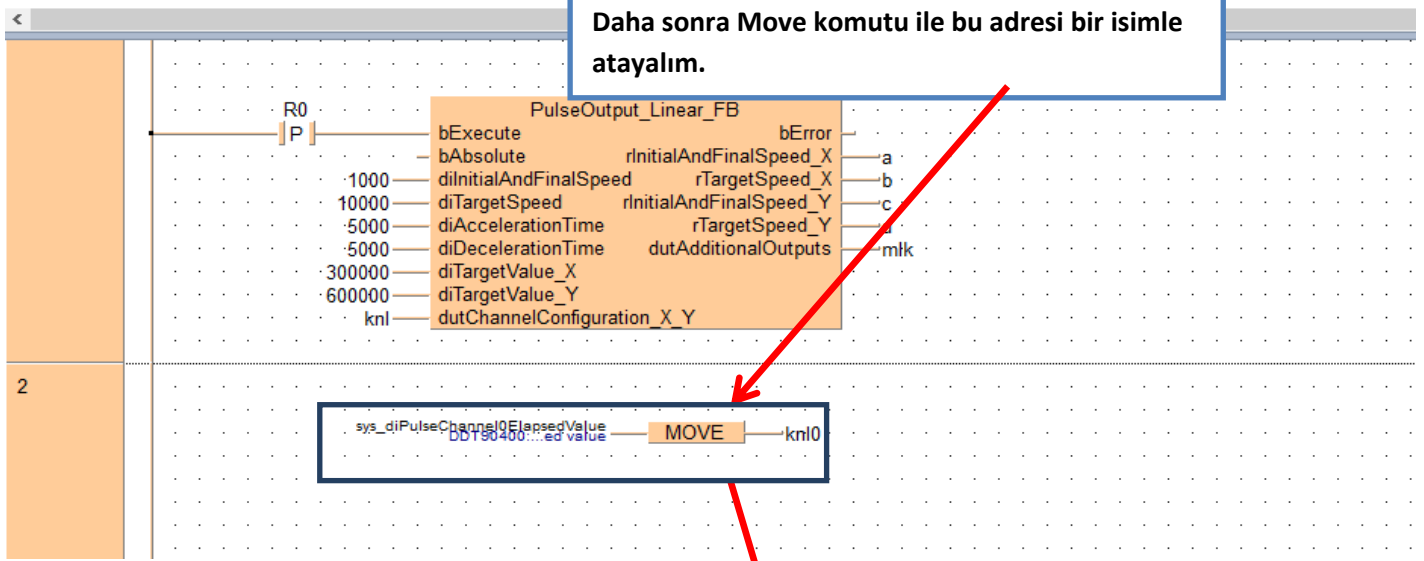
Adres değerini programda göstermek için bu segmeye tıklanır, mouse ucuna gelen kutucuk programda istediğimiz alana getirilip bırakılır. Daha sonra adres değeri bu alana girilir ve Enter tuşuna basılır.

Class	Identifier	Type	Initial	Comment
7	VAR	knl	PulseOutput_Chan...	

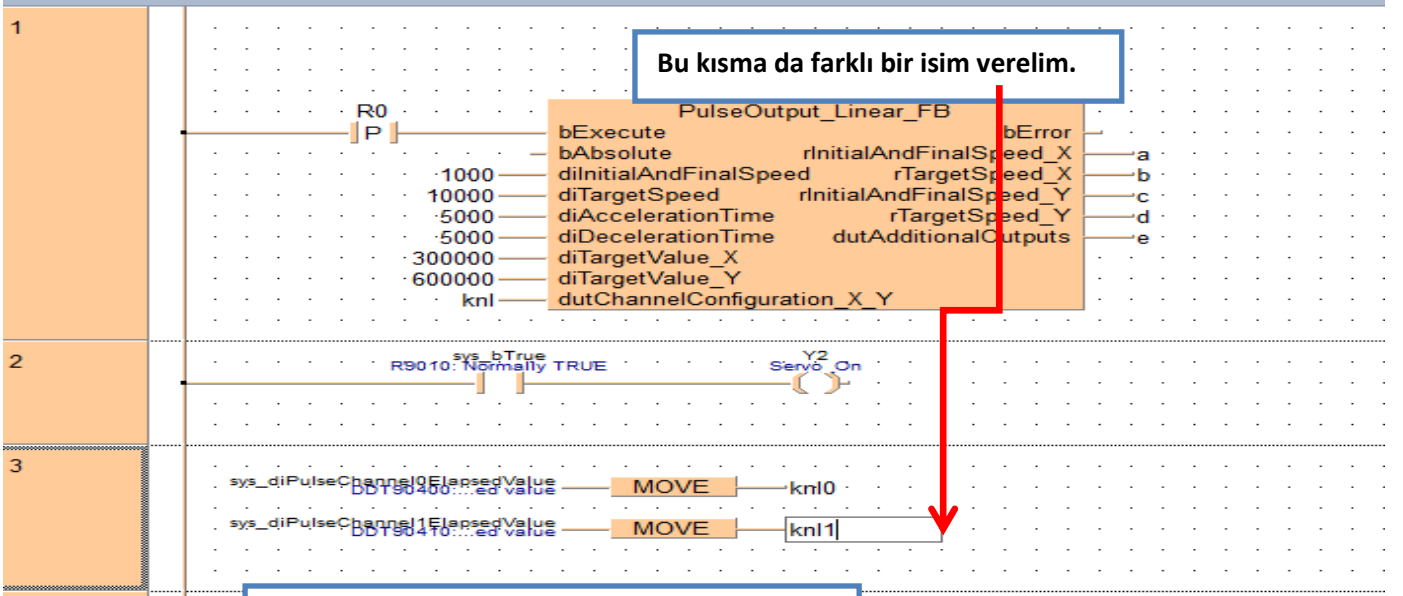
Adres değerini programda göstermek için bu segmeye tıklanır, mouse ucuna gelen kutucuk programda istediğimiz alana getirilip bırakılır. Daha sonra adres değeri bu alana girilir ve Enter tuşuna basılır.

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

7



SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ



Belirlediğimiz bu isimlerin data tiplerini belirleyelim.

Program_1 • x Global variables

	Class	Identifier	Type	Initial	Comment
2	VAR	error	PulseOutput_Chan...		
3	VAR	a	REAL	0.0	
4	VAR	b	REAL	0.0	
5	VAR	c	REAL	0.0	
6	VAR	d	REAL	0.0	
7	VAR	kn10	DINT	0	
8	VAR	kn11	DINT	0	
9	VAR	e	PulseOutput_Linear...		
10	VAR				

1

Bu kısımdan mouse ile tutup aşağı doğru seçerek bu pencereyi açabiliriz.

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

Class	Identifier	Type	Initial
2	VAR	error	PulseOutput_Chann...
3	VAR	a	REAL
4	VAR	b	REAL
5	VAR	c	REAL
6	VAR	d	REAL
7	VAR	knI0	DINT
8	VAR	knI1	DINT
9	VAR	e	
10	VAR		

Bu kısımdan Data tipini belirleyebiliriz.

Bizim okuyacağımız enkoder değeri DDT tipinde olduğu için bu kısımdan DINT seçilir.

Her ikisini de bu şekilde seçtikten sonra OK deyip kapatalım.

Derleme işlemi buradan yapılır.

Derleme işleminden sonra görüldüğü gibi hatamız bulunmadığına göre Close deyip kapatabiliriz.

Şimdi programımızı yükleme işlemini yapıp çalıştığını görelim.

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

1

Program_1 x Global variables

Class Identifier

1

linear_deneme

Control FPWIN Pro 7

? The program code in FPWIN Pro is different from the program code in the PLC.
The PLC configuration in FPWIN Pro is different from the PLC configuration in the PLC.
Do you want to download the program code and the PLC configuration?

2

2.adım Evet diyelim

Evet Hayır

3

sys_diPulseChannel0ElapsedValue — MOVE — knl0

sys_diPulseChannel1ElapsedValue — MOVE — knl1

Görüldüğü gibi R0 girişine tetik vermediğimiz için motor pals değerleri şu anda '0' değerini göstermektedir.

Görüldüğü gibi programımız yüklenmiş ve online olunmuştur.

Bu değer sabit yapılmalı ve Target speed değerinden küçük olmalıdır.

Type Initial Comment

1

R0

PulseOutput_Linear_FB

bExecute bAbsolute rInitialAndFinalSpeed_X a = 0.0

1000 diInitialAndFinalSpeed rTargetSpeed_X b = 0.0

10000 diTargetSpeed rInitialAndFinalSpeed_Y c = 0.0

5000 diAccelerationTime rTargetSpeed_Y d = 0.0

5000 diDecelerationTime dutAdditionalOutputs e

300000 diTargetValue_X

600000 diTargetValue_Y

knl dutChannelConfiguration_X_Y

2

R9010: Normally TRUE Servo_On

3

sys_diPulseChannel0ElapsedValue = 0 — MOVE — knl0 = 0

sys_diPulseChannel1ElapsedValue = 0 — MOVE — knl1 = 0

SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

The screenshot displays a Ladder Logic (LAD) program with three rungs. Rung 1 contains a normally open contact labeled 'R0' connected to the 'bExecute' input of a 'PulseOutput_Linear_FB' function block. The block's inputs are configured as follows: 'bAbsolute' is set to '1000', 'diInitialAndFinalSpeed' is '10000', 'diTargetSpeed' is '5000', 'diAccelerationTime' is '5000', 'diDecelerationTime' is '300000', 'diTargetValue_X' is '600000', 'diTargetValue_Y' is '600000', and 'kn1' is 'dutChannelConfiguration_X_Y'. The block's outputs are: 'rInitialAndFinalSpeed_X' is 'a = 447.21359', 'rTargetSpeed_X' is 'b = 4472.1362', 'rInitialAndFinalSpeed_Y' is 'c = 894.42719', and 'rTargetSpeed_Y' is 'd = 8944.2725'. Rung 2 shows a normally open contact 'R010: Normally TRUE' connected to the 'sys_bTrue' input of a 'Servo_On' coil. Rung 3 contains two 'MOVE' instructions: 'sys_ciPulseChannel0ElapsedValue = 300000' moving to 'kn0 = 300000' and 'sys_ciPulseChannel1ElapsedValue = 600000' moving to 'kn1 = 600000'. A table of global variables is visible on the right side of the screen.

Daha sonra R0 tetiklenerek motorumuzu Linear_FB modunda çalıştırabiliriz.

Görüldüğü gibi iki motorun hızları farklı oldu. TargetValue değerleri birbirinden farklı olduğu için hız değerleri de buna bağlı olarak PLC tarafından ayarlandı.

İki motorumuz da aynı zamanda hedef değerlerine ulaşmıştır.

Böylece FPWINPRO 7 de LINEAR_FB bloğunu çalıştırmış olduk.