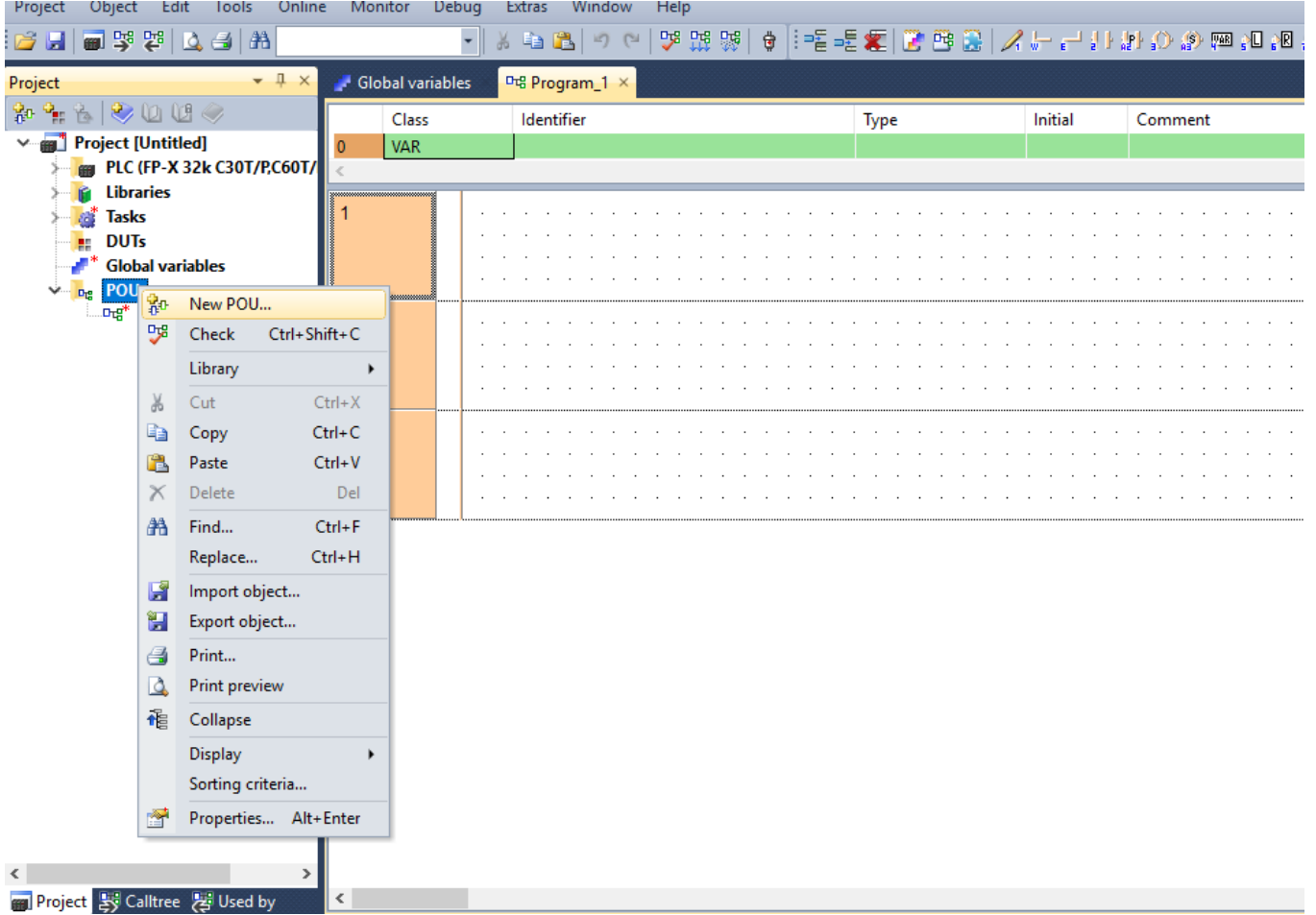


# SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

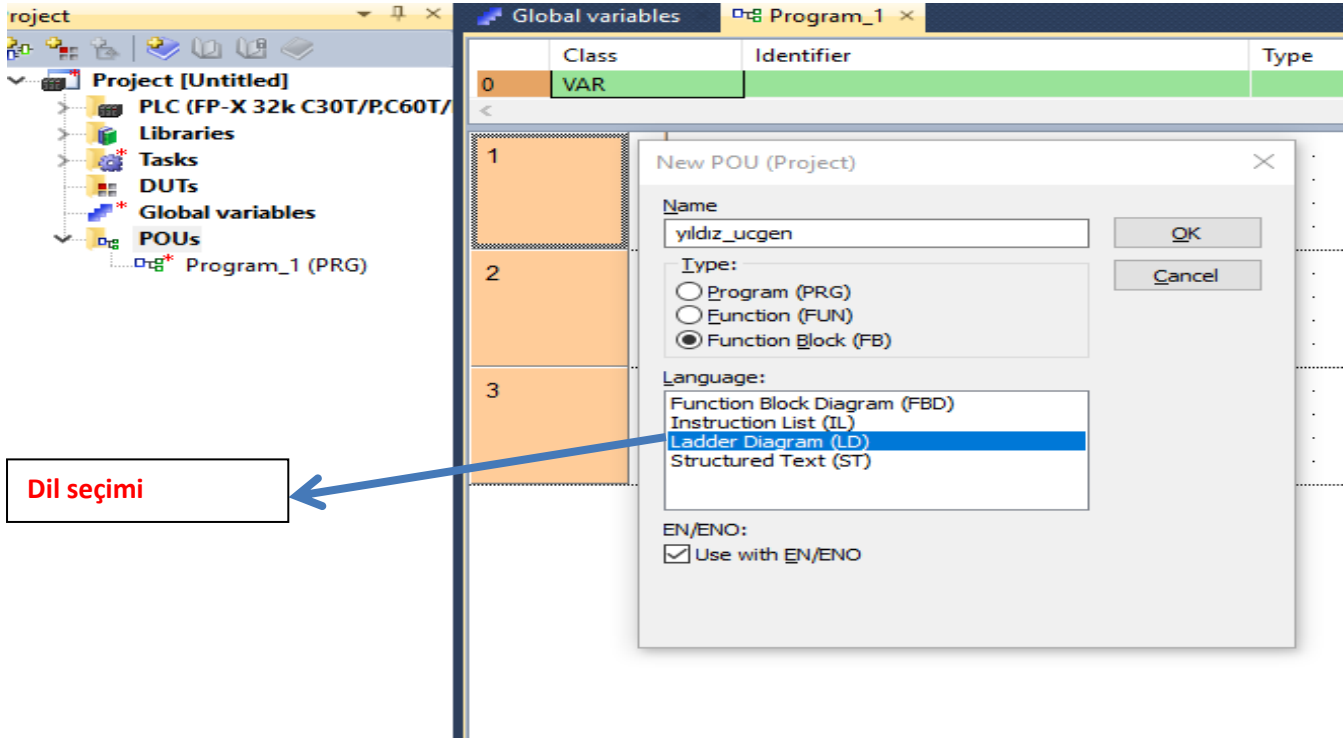
## FPWIN PRO 7 de Fonksiyon Blok Oluşturma

1)Programı açtıktan sonra sol tarafta bulunan menüden 'POUs' ikonu üstüne sağ tık yaparak New POU seçilir.

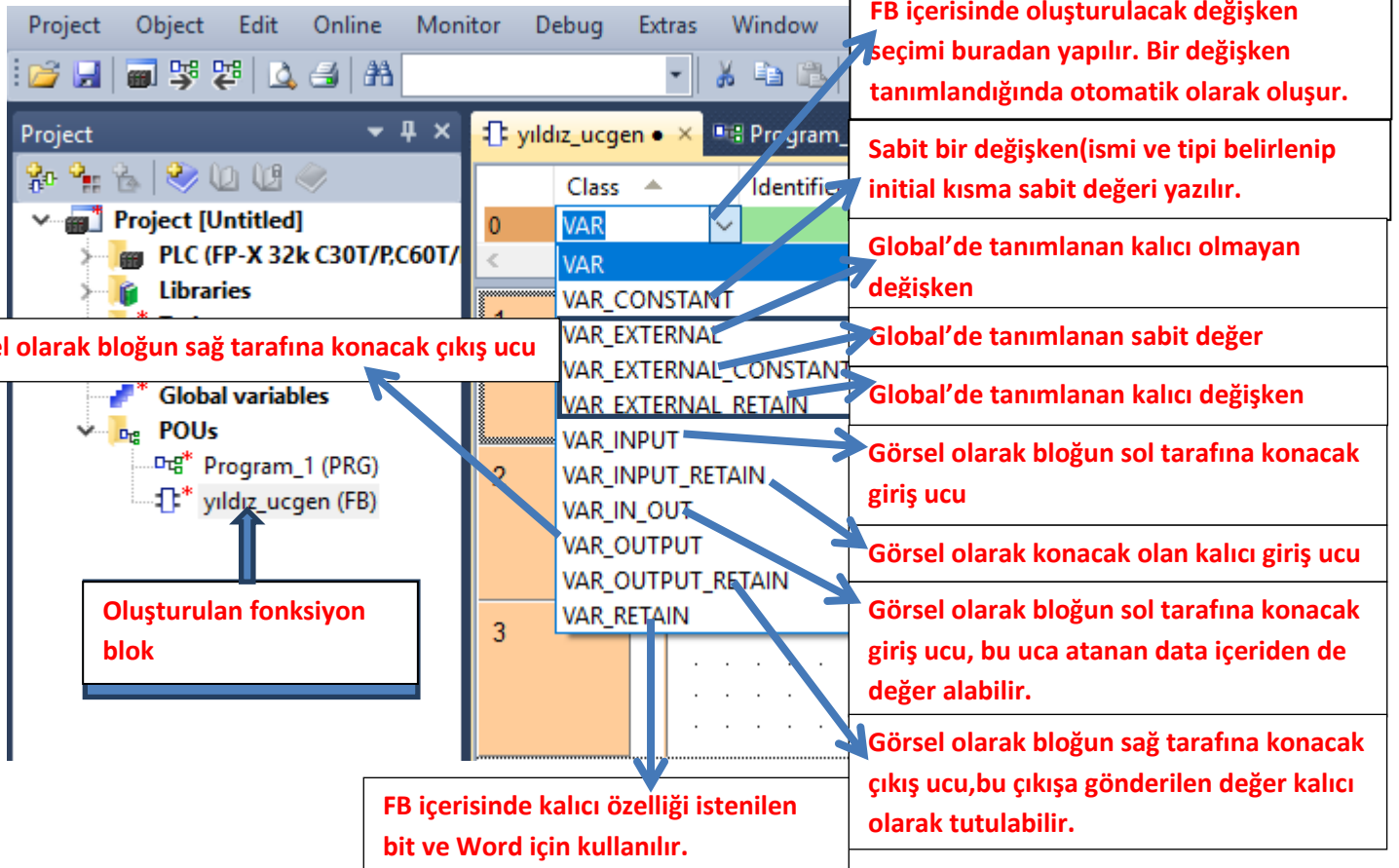


2)Karşımıza böyle bir ekran çıkacaktır.Bizim burada 'yıldız\_ucgen' olarak tanımladığımız kısma fonksiyonunuzun ismini verebilirsiniz.Type kısmından ise FB seçilerek devam edilir.Kullandığımız programlama dilini 'Language' kısmından seçebiliriz. Oluşturulacak fonksiyon blokta en altta seçili EN-ENO uçlarının olup olmayacağı seçilebilir.

# SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ



3)Yaptığımız ayarlamalardan sonra fonksiyon bloğunun proje sayfası açılır.



4)FB de kullanılacak değişken tipleri seçilir.

Oluşturulacak fonksiyon bloğunda kullanılacak değişken tipleri tek tek seçilmelidir.

# SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

**Bir dizi aynı tip değişkeni gruplandırır.**

**Bit özelliği**

**Bazı fonksiyonlara özel tarih formatı**

**Bazı fonksiyonlara özel tarih ve saat formatı**

**32 bit DINT decimal formatı +/- yazılabilir.**

**32 bit doubleworld hexadecimal formatı**

**16 bit int decimal formatı +/- yazılabilir.**

**32 bit noktalı sayı formatı**

**Karakter formatı 32 karakter**

**Bazı fonksiyonlara özel saat formatı**

**Bazı fonksiyonlara bir günlük saat formatı (00.00.00-23.59.59)**

**32 bit DINT sadece pozitif sayı formatı**

**16 bit INT sadece pozitif sayı formatı**

**16 bitlik bir alandır hexadecimal formatında**

Class	Identifier	Type	Initial	Comment
0	VAR			

Type class	Type
<input checked="" type="radio"/> Elementary data type	ARRAY
<input type="radio"/> Data unit types	BOOL
<input type="radio"/> Function blocks	DATE
	DATE_AND_TIME
	DINT
	DWORD
	INT
	REAL
	STRING[32]
	TIME
	TIME_OF_DAY
	UDINT
	UINT
	WORD

## 5)FB blok oluşturulamaya başlarız.

Class	Identifier	Type	Initial	Comment
0	VAR_INPUT	start	BOOL	FALSE
1	VAR_INPUT	zaman	INT	0
2	VAR_OUTPUT	motor	BOOL	FALSE
3	VAR_OUTPUT	yıldız	BOOL	FALSE
4	VAR_OUTPUT	uçgen	BOOL	FALSE
5	VAR			

**FB blok oluşturulamaya başlandığında önceden tanımlı bit ve Wordleri kullanabiliriz.**

## 6)

# SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

The screenshot displays the SAVIOR software interface. At the top, there are two tabs: 'yıldız\_ucgen' and 'Program\_1'. Below the tabs is a table with the following columns: Class, Identifier, Type, Initial, and Comment.

Class	Identifier	Type	Initial	Comment
0	VAR_INPUT	start	BOOL	FALSE
1	VAR_INPUT	zaman	INT	0
2	VAR_OUTPUT	motor	BOOL	FALSE
3	VAR_OUTPUT	yıldız	BOOL	FALSE
4	VAR_OUTPUT	üçgen	BOOL	FALSE
5	VAR			

Below the table is a ladder logic diagram with two rungs. Rung 1 contains a normally open contact labeled 'start' and a coil labeled 'motor'. Rung 2 is empty. A red text box with a blue arrow pointing to the 'motor' coil contains the following text:

**Fonksiyon blok içerisine timer eklemek istediğimizde eğer fonksiyon bloğunu birden fazla yerde kullanıyorsak sonunda FB ile ifade edilen timer**

On the right side of the interface, there is an 'Instructions' panel. It shows a tree view of instructions. The 'Timer instructions' category is expanded, showing a list of timer instructions: F137\_STMR, F183\_DSTM, TM\_100ms\_FB, TM\_10ms\_FB, TM\_1ms\_FB, and TM\_1s\_FB. A blue arrow points from the text box to the 'TM\_10ms\_FB' instruction.

- Fonksiyon bloğunu birden fazla kullanacaksa bit ve Word isimlerini dahili olarak kullanmamalıyız.
- Örn: R0 yerine start,DT yerine deger gibi.

# SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

Table showing variable declarations:

	Class	Identifier	Type	Initial	Comment
1	VAR_INPUT	zaman	INT	0	
2	VAR_OUTPUT	motor	BOOL	FALSE	
3	VAR_OUTPUT	yıldız	BOOL	FALSE	
4	VAR_OUTPUT	üçgen	BOOL	FALSE	
5	VAR	tm1	TM_1s_FB		

Timer ı fonsiyon blok içerisine aldığımızda bir isim vermemiz gerekir.

İsim kısmına tm1 yazıp Enter a basıldığında üstteki pencere açılır ve herhangi birşey yapmadan OK tuşuna basınız.

Modify 'tm\_1' dialog box:

- Location: <Header> yıldız\_ucgen
- Class: VAR
- Identifier: tm\_1
- Type: TM\_1s\_FB
- Initial:
- Commer:
- Deactivat:

Compile/check messages dialog box:

- <Interrupt 5>
- <Interrupt 6>
- <Interrupt 7>
- <Periodic interrupt>
- <Program\_1 (PRG, LD)>
- <Program\_1: Header>
- <Program\_1: Body>
- <ROM\_Download Program code (21 steps)>
- 0 errors
- 0 warnings

Minimize dialog after display:

Buttons: Show, -> Error, -> Warning, Cancel, Close

Ladder Logic Diagram:

- Input: zaman
- Output: motor
- Output: yıldız
- Output: üçgen
- Timer: tm\_1 (TM\_1s\_FB)
- Timer inputs: start (T), SV (EV)
- Timer output: cikis

1. FB içerisine program yazıldıktan sonra compile(derleme) işlemi yapılır. Herhangi bir hata bulunmuyor ise kapatıp proje sayfasına geçebiliriz.

# SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

The screenshot shows the SIMATIC Manager software interface. The left pane displays the project structure under 'Project [Untitled]'. The main editor window shows the 'Program\_1' project with a table of variables and a ladder logic editor.

Class	Identifier	Type	Initial	Comment
0	VAR			

An arrow points from the 'yıldız\_ucgen (FB)' entry in the project tree to the main editor window. A text box contains the following text:

**Ana proje (Program\_1) sayfasına geçildikten sonra POU's kısmında Program\_1 altında yaptığımız fonksiyon bloğu görebiliriz.**

The screenshot shows the SIMATIC Manager software interface with the function block editor open. The left pane shows the project structure. The main editor window shows the 'yıldız\_ucgen' function block being edited.

Class	Identifier	Type	Initial	Comment
0	VAR	motor1	yıldız_ucgen	

The function block editor shows the following connections:

- motor1 (VAR) is connected to the EN input of the yıldız\_ucgen function block.
- start (VAR) is connected to the ENO output of the yıldız\_ucgen function block.
- zaman (VAR) is connected to the motor input of the yıldız\_ucgen function block.
- üçgen (VAR) is connected to the yıldız input of the yıldız\_ucgen function block.

An arrow points from the 'yıldız\_ucgen (FB)' entry in the project tree to the function block editor. A text box contains the following text:

**Fonksiyon bloğumuza bir isim vermemiz gerekir.**

**Program\_1 altında bulunan Yıldız\_ucgen fonksiyon bloğu üzerine gelip Mouse ile seçtikten sonra Enter butonuna basarsak blok mouse'un ucuna almış oluruz. Böylelikle program üzerinde herhangi bir noktaya bırakabiliriz.**

# SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

Class	Identifier	Type	Initial	Comment
0	VAR	motor1	yıldız_ucgen	
1	VAR			

Compile/check messages

<yıldız\_ucgen: Header>  
<Program\_1: Body>  
Warning in NW1: Warning: 'R0' is an explicit user address. It is recommended to  
Warning in NW1: Warning: 'Y0' is an explicit user address. It is recommended to  
Warning in NW1: Warning: 'Y1' is an explicit user address. It is recommended to  
Warning in NW1: Warning: 'Y2' is an explicit user address. It is recommended to  
<yıldız\_ucgen: Body>  
<ROM\_Download Program code (74 steps)>  
0 errors  
4 warnings

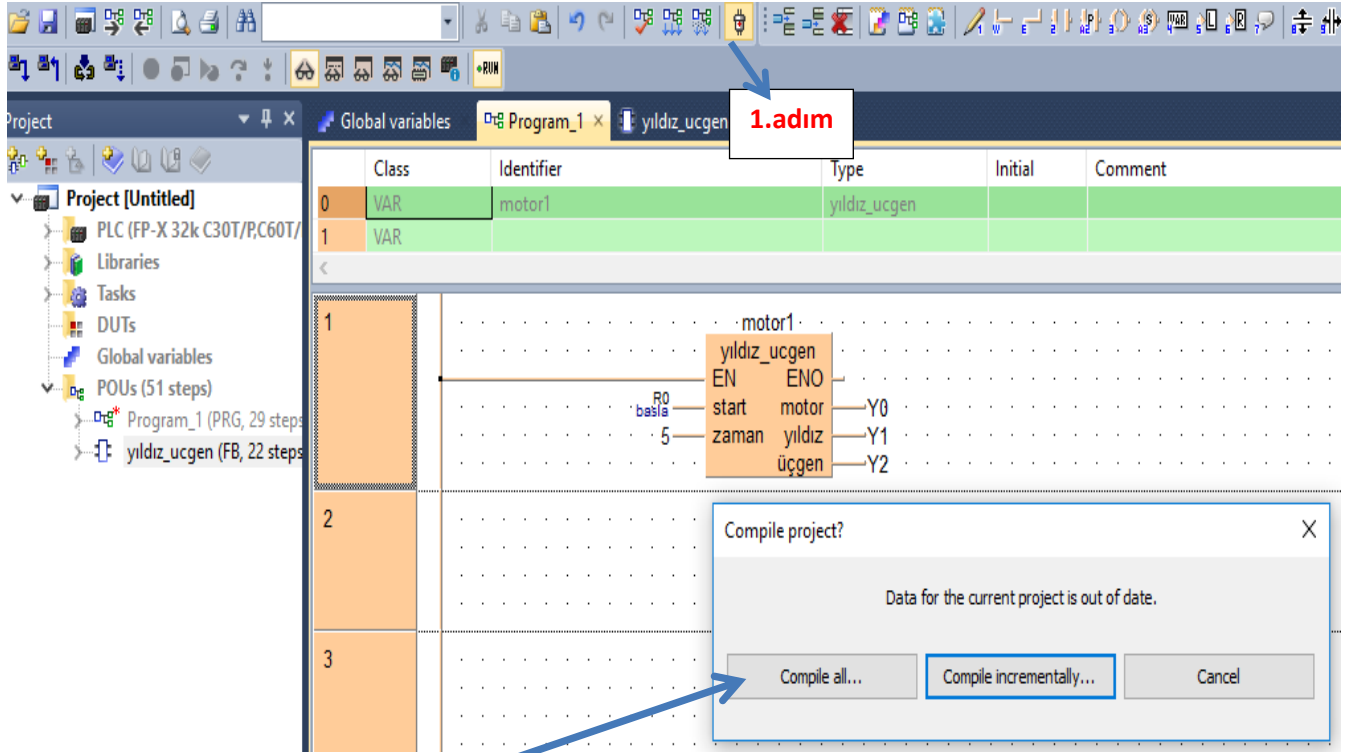
Minimize dialog after display

Show -> Error -> Warning Cancel Close

**Fonksiyon bloğun giriş ve çıkış adreslerini tanımladıktan sonra compile işlemi yapılmalı.**

**Compile işlemi yapıldıktan sonra bir sorun yoksa programı plc ye yükleyebiliriz.**

# SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ



1.adım

Class	Identifier	Type	Initial	Comment
0	VAR	motor1	yildiz_ucgen	
1	VAR			

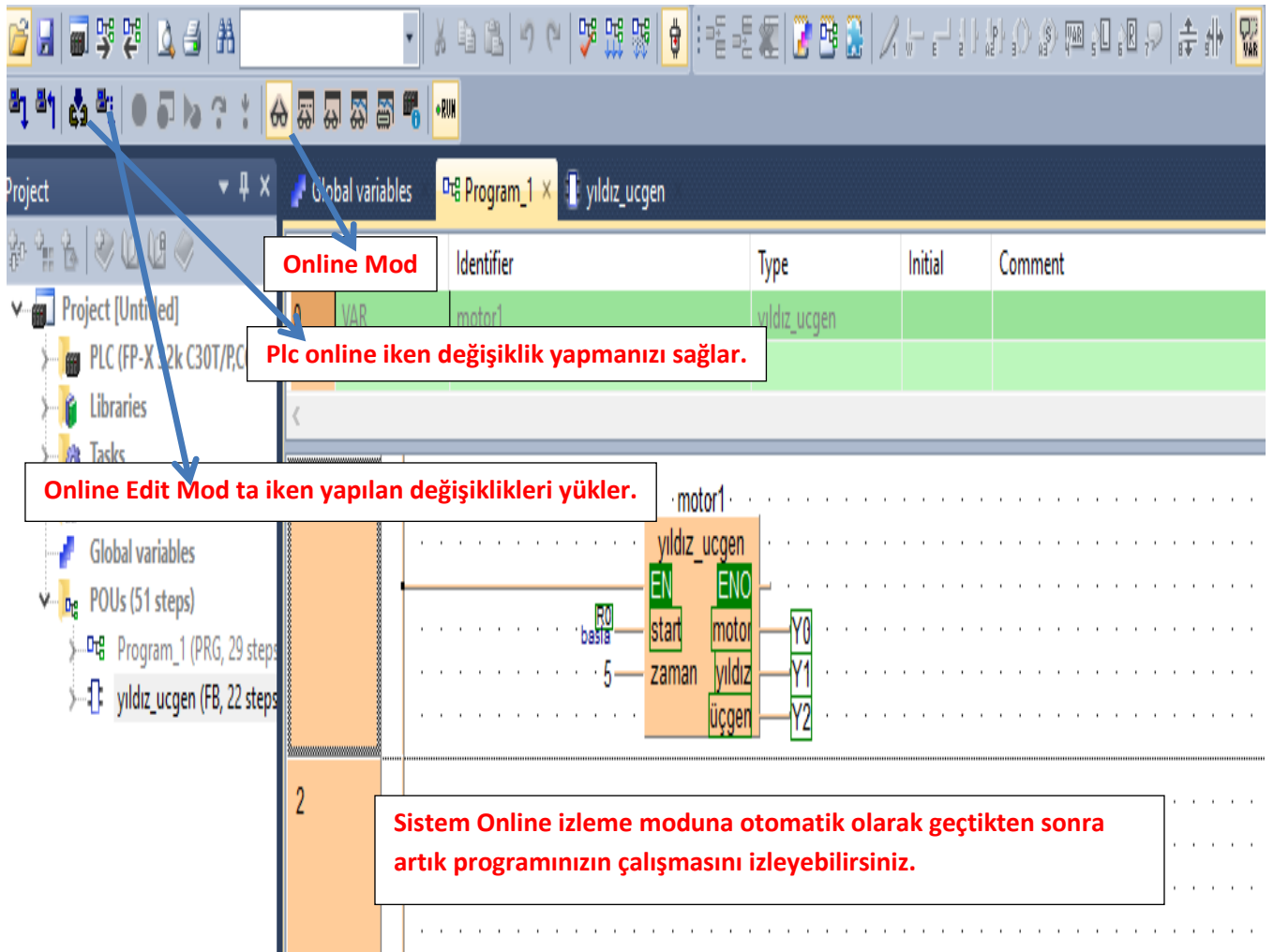
Compile project?

Data for the current project is out of date.

Compile all... Compile incrementally... Cancel

Compile all diyerek devam edelim.

Böylece programımız PLC ye yüklenmiş oldu.



Online Mod

Plc online iken değişiklik yapmanızı sağlar.

Online Edit Mod ta iken yapılan değişiklikleri yükler.

Sistem Online izleme moduna otomatik olarak geçtikten sonra artık programınızın çalışmasını izleyebilirsiniz.



# SAVİOR OTOMASYON TEKNİK DESTEK BİRİMİ

The screenshot shows the Siemens SIMATIC Manager interface. The top menu bar includes Project, Object, Edit, Tools, Online, Monitor, Debug, Extras, Window, and Help. The toolbar contains various icons for file operations, editing, and simulation. The Project tree on the left shows a project named 'Project [Untitled]' with a PLC (FP-X 32k C30T/P/C60T/), Libraries, Tasks, DUTs, Global variables, and POU (51 steps). The main workspace displays a ladder logic diagram with two function blocks (FB) for motor control. The first FB is named 'motor1' and the second is named 'motor2'. Both FBs are of type 'yıldız\_ucgen'. The first FB has inputs 'start' (motor) and 'zaman' (yıldız) and outputs 'Y0', 'Y1', and 'Y2'. The second FB has inputs 'start' (motor) and 'zaman' (yıldız) and outputs 'Y0', 'Y1', and 'Y2'. A text box in the bottom left corner contains the following text: 'Eğer fonksiyon bloğu birden fazla kullanmak isterseniz sadece kopyala yapıştır yaptıktan sonra ismini değiştirmeniz yeterli olacaktır.'

Class	Identifier	Type	Initial	Comment
0	VAR	motor1	yıldız_ucgen	
1	VAR	motor2	yıldız_ucgen	

1

motor1

yıldız\_ucgen

EN ENO

start motor —Y0

zaman yıldız —Y1

üçgen —Y2

2

motor2

yıldız\_ucgen

EN ENO

start motor —?

zaman yıldız —?

üçgen —?

Eğer fonksiyon bloğu birden fazla kullanmak isterseniz sadece kopyala yapıştır yaptıktan sonra ismini değiştirmeniz yeterli olacaktır.