

Sysmac Studio'da Fonksiyon Bloğu ve Fonksiyon Oluşturma

- Genel Bilgi
- Fonksiyon Bloğu Oluşturma
- Fonksiyon Oluşturma
- Fonksiyon ile Fonksiyon Bloğu Arasındaki Farklar

1. Genel Bilgi

Öncelikle Fonksiyon Blokları ve Fonksiyonların özelliklerinden bahsedilecektir.

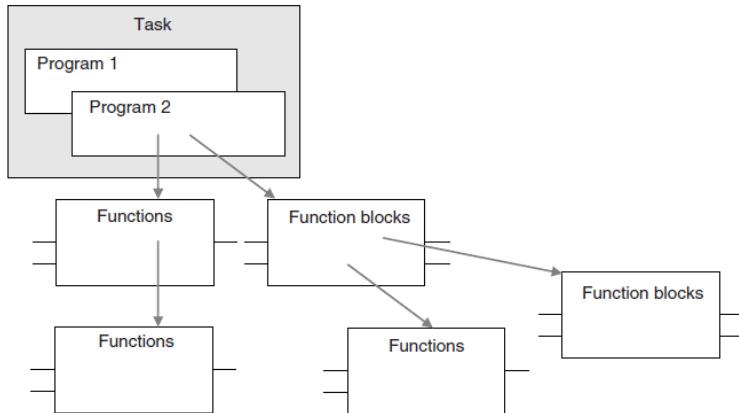
1.1 Fonksiyon Blokları

- Fonksiyon blokları programdan veya diğer fonksiyon bloklarından çağrılabilmektedir.
- Hem kullanıcı tanımlı hem de sistem tanımlı fonksiyon blokları bulunmaktadır.
- Fonksiyon bloklarının algoritmasında herhangi programda bulunan fonksiyon, fonksiyon blokları ve kullanıcı tanımlı fonksiyon, kullanıcı tanımlı fonksiyon blokları kullanılabilir.

1.2 Fonksiyonlar

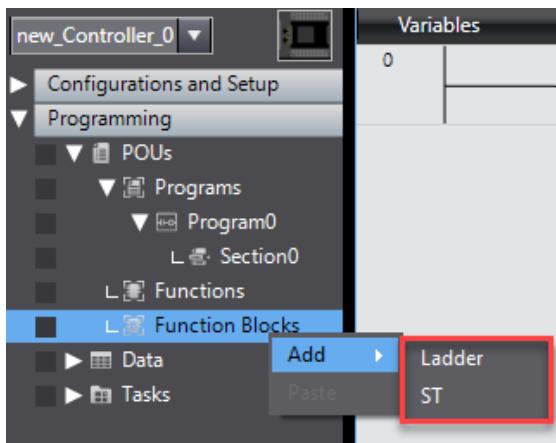
- Fonksiyonlar programdan, fonksiyon bloklarından veya diğer fonksiyon bloklarından çağrılabilmektedir.
- EN girişi yürütme koşulunu belirtmektedir. Sadece TRUE olduğunda fonksiyon yürütülmektedir.
- Hem kullanıcı tanımlı hem de sistem tanımlı fonksiyonlar bulunmaktadır.
- Fonksiyonların algoritmasında programda bulunan fonksiyonlar blokları ve kullanıcı tanımlı fonksiyon blokları kullanılamamaktadır.

Item	POU type	Function blocks	Functions
Execution method		Called from a program or another function block.	Called from a program, function, or function block.
Algorithm	Any instructions	Supported.	Not supported.
	User-defined functions	Supported.	Supported.
	User-defined function blocks	Supported.	Not supported.
Execution condition		Executed each period. Specify the execution condition with an input variable.	Specify the execution condition with the EN input.

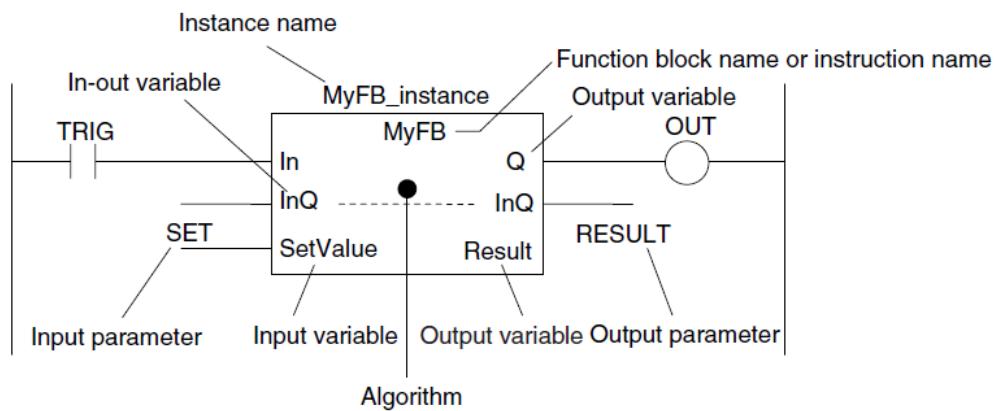


2. Fonksiyon Bloğu Oluşturma

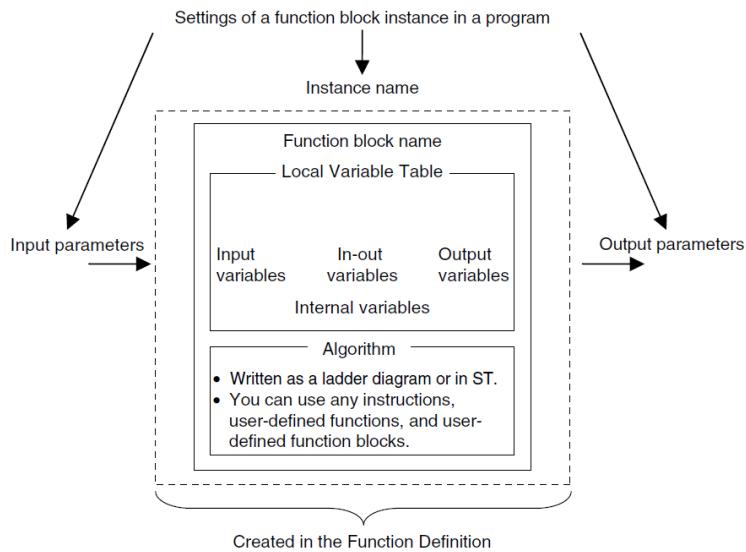
Fonksiyon bloğu Ladder olarak veya ST olarak oluşturulabilir.



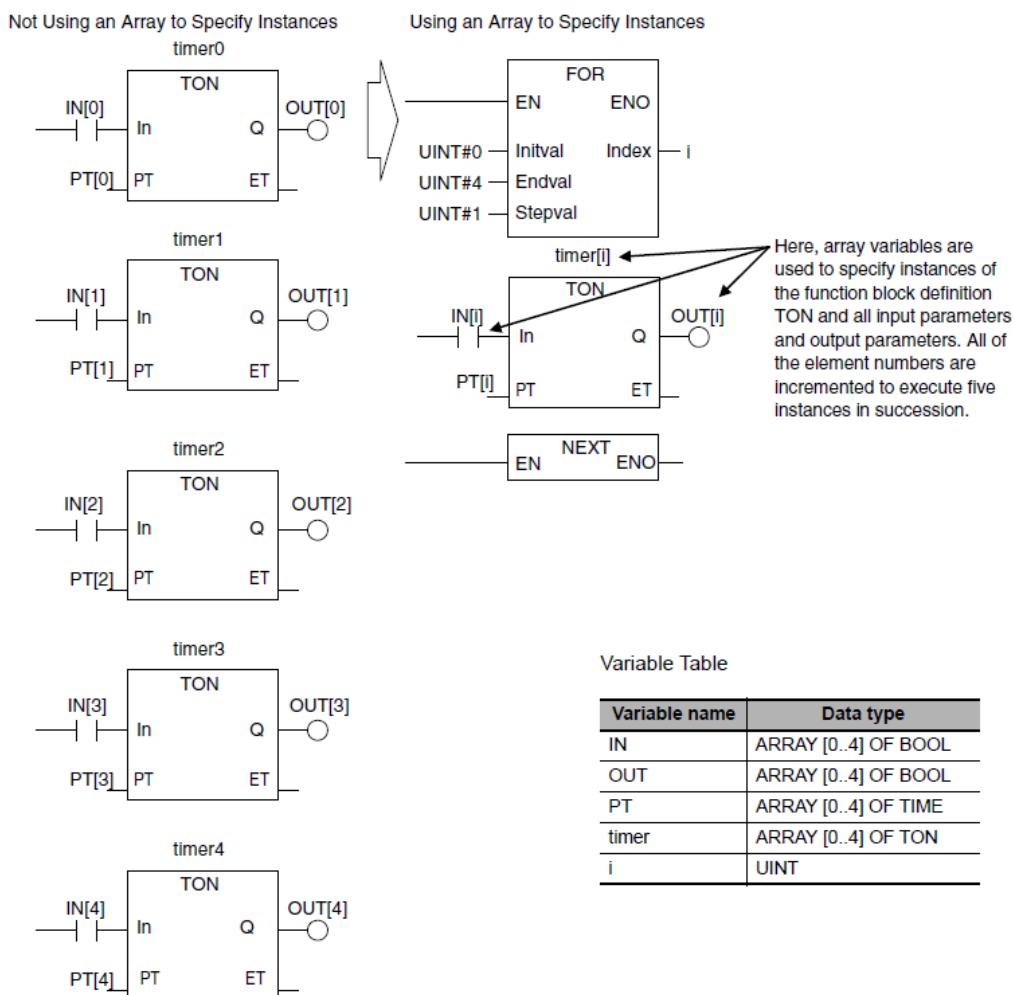
Ladder olarak oluşturulan fonksiyon bloğunun yapısı aşağıda görüldüğü gibidir.



Fonksiyon bloğu ismiyle çağrılmaktadır. Instance ismi programda kullanılan ismidir ve fonksiyon bloklarını birbirinden ayırarak farklı alanlar kaplamasını sağlar.



Instance ismi Array olarak da tanımlanabilmektedir.



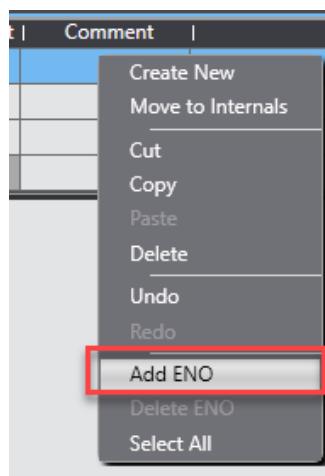
Fonksiyon bloğunda kullanılacak değişkenler aşağıda görülen Variables kısmına tanımlanmaktadır.

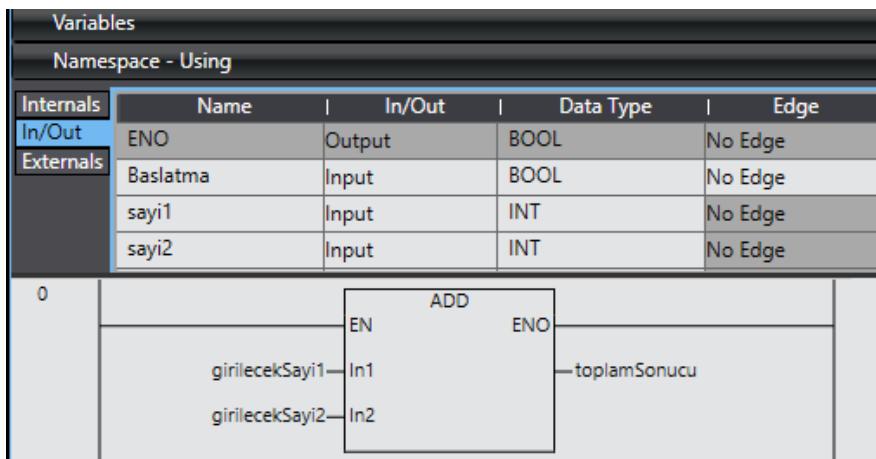
Fonksiyon bloğu çağrıldığında çağrılan kısımdan girilmesi gereken input ve outpular In/Out kısmından girilerek Input, Output ya da In/Out olarak kullanımına göre seçilmelidir.

Variables									
Namespace - Using									
Internals	Name	Data Type	Initial Value	AT	Retain	Constant	Comment		
In/Out	Empty. Click here to add item.								
Externals									

Variables									
Namespace - Using									
Internals	Name	In/Out	Data Type	Edge	Initial Value	Retain	Constant	Comment	
In/Out	input	Input	BOOL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>		
Externals	output	Input	BOOL	No Edge		<input type="checkbox"/>	<input type="checkbox"/>		
		Input							
		In/Out							
		Output							

Fonksiyon bloğu oluştururken ENO çıkışınıda eklemek gerekmektedir. İlk girilen Input Ladder'a bağlı olarak gelmektedir. Bunun için oraya giriş yapılması gereken inputun girilmemesi faydalı olacaktır.

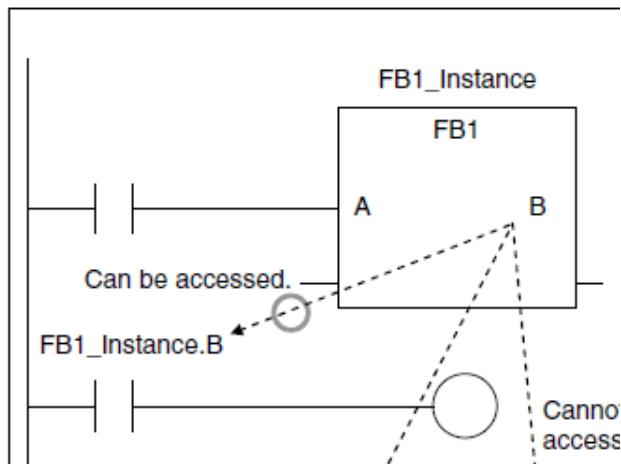




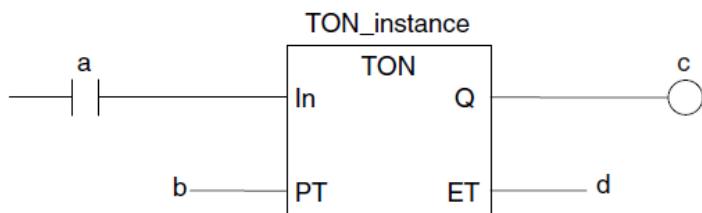
Değişkenlerin maksimum tanımlanma sayıları bulunmaktadır. Bunlar aşağıdaki tablodan görülebilir.

Input variables	1-64
Output variables	1-64
In-out variables	0-64
Internal variables	Sınır yok
External variables	Sınır yok
EN	Fonksiyon bloklarında bulunmaz.
ENO	0 veya 1

Fonksiyon bloğunda bulunan değişkenlere program içinden ulaşılabilir. Programda Instance ismi.değişken ismi şeklinde ulaşılabilir.



Fonksiyon bloğunun ST dilinde çağrılmaları için programda bulunan TON fonksiyon bloğunun kullanımı örnek verilebilir.



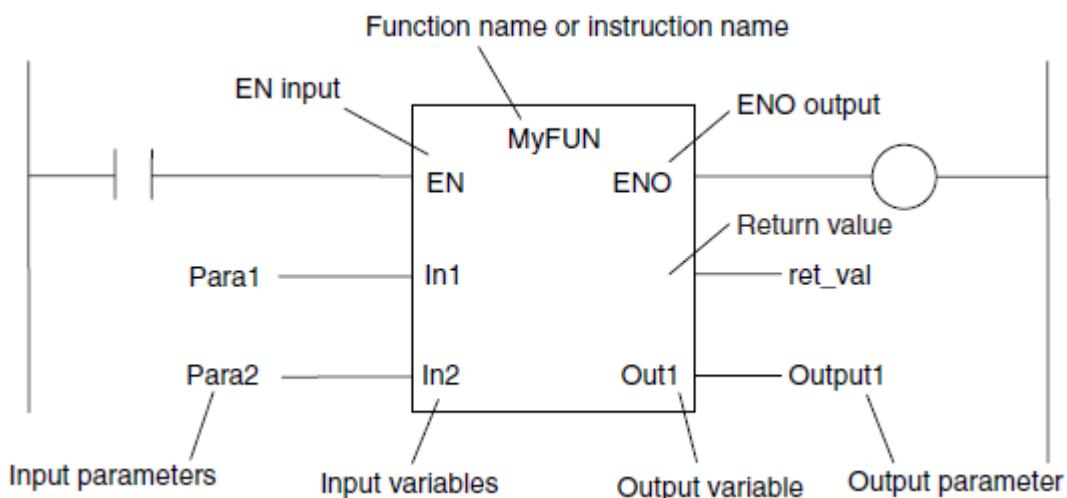
Function Blocks Expressed in ST:

Instance name
 TON_instance(*In*:=*a*, *PT*:=*b*, *Q*=>*c*, *ET*=>*d*);
 TON_instance(*In*:=*a*, *PT*:=*b*, *Q*=>*c*); (*The *ET* output is omitted here.*)
 TON_instance(*a*,*b*,*c*,*d*); (*Input and output variables are omitted here.*)

Klavizeden “F” ile fonksiyon bloğu çağrılmaktadır.

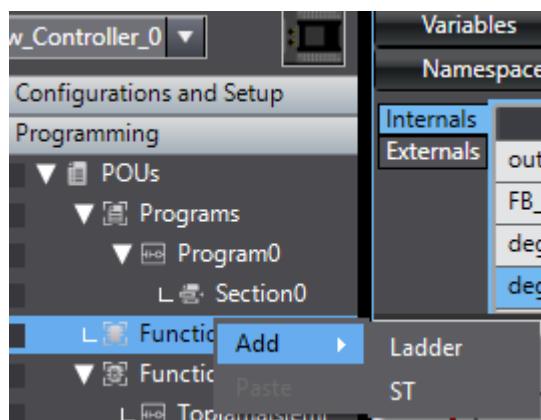
3. Fonksiyon Oluşturma

Ladder'da oluşturulan fonksiyon yapısı aşağıdaki gibidir.

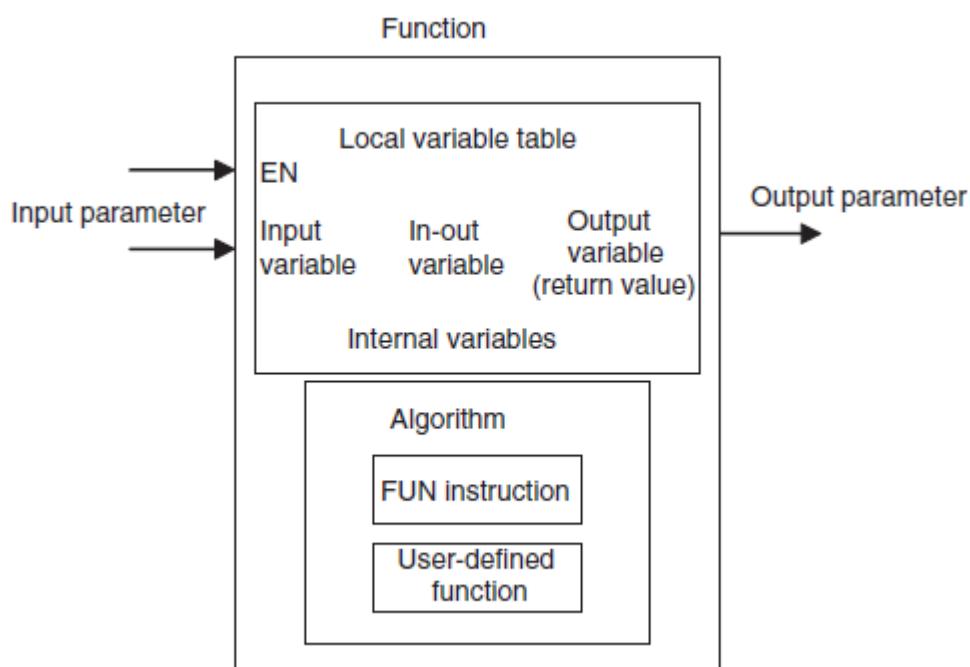


Fonksiyonlara programda kullanılırken isim verilmeyz. Fonksiyon oluştururken verilen isim yeterlidir.

Fonsiyonlar da Ladder ve ST tipinde oluşturulabilir.



Girişine otomatik olarak EN girişi eklenmiş durumdadır. EN girişi olmak zorundadır.



Fonksiyon bloğunda olduğu gibi değişken sayılarında sınırlama olabilmektedir. Aşağıdaki tablodan görülebilir.

Input variables	0-64
Output variables	0-64
In-out variables	0-64
Internal variables	Sınır yok
External variables	Sınır yok
EN	1
ENO	0 veya 1
Return value	1

EN ve ENO parametreleriyle ilgili karışıklık olmaması için aşağıdaki tablo incelenebilir.

POU		EN	ENO
FB	User-defined functions	Cannot be used. A building error occurs if you try to define EN in the variable table from the Sysmac Studio.	Can be used or omitted. You define ENO as an output variable in the Sysmac Studio.
	Instruction	All FB instructions do not use EN.	Some instructions use ENO, and others do not.
FUN	User-defined functions	Required. When you create a function, the Sysmac Studio automatically adds EN to the variable table by default.	Can be used or omitted. You define ENO as an output variable in the Sysmac Studio.
	Instruction	All FUN instructions use EN.	Some instructions use ENO, and others do not.

4. Fonksiyon ve Fonksiyon Bloğu Arasındaki Farklar

Kısaca özetlemek gerekirse;

- Fonksiyonlar instance ismine ihtiyaç duymaz ama fonksiyon bloklarında girilmek zorundadır.
- Fonksiyonlar bellek alanı kaplamazlar ve bu yüzden de sınırsız kullanılabilirler ama fonksiyon blokları alan kapladığı için kullanımlarında sınır vardır.
- Fonksiyonlar durum hatırlaması gerektirmeyen basit hesaplamalar için uygundur, fonksiyon blokları da hatırlanma gereken timer gibi işlemler için kullanılır.
- Fonksiyonlarda değişkenler için “AT” kısmı yoktur.