فصل پنجم

برنامه نویسی ساختار یافته(فانکشن و فانکشن بلاک ها)

در این فصل آشنا خواهید شد با ....

۱. نحوه عملکرد فانکشن ها و فانکشن بلاک ها
۲. برنامه نویسی با این بلوک ها

فانکشن ها Function :

همان طور که قبلاً نیز گفته شد برنامه نویسی به دو طریق قابل انجام است:

حالت اول که حالت خطی است برای برنامه های با حجم کم جوابگو است و اگر برنامه اندکی پیچیده شود عیب یابی و تست آن بسیار سخت می گردد. برای حل این مشکل برنامه نویسی به روش فانکشن ها پیشنهاد می گردد. اساس کار به این صورت است که کاربر ابتدا فانکشنی را می نویسد و در صورت نیاز آن را فراخوانی می نماید

فانکشن چیست ؟ مجموعه دستور العمل هایی است که کاربر برای رسیدن به هدف مشخصی آن را ایجاد می نماید.

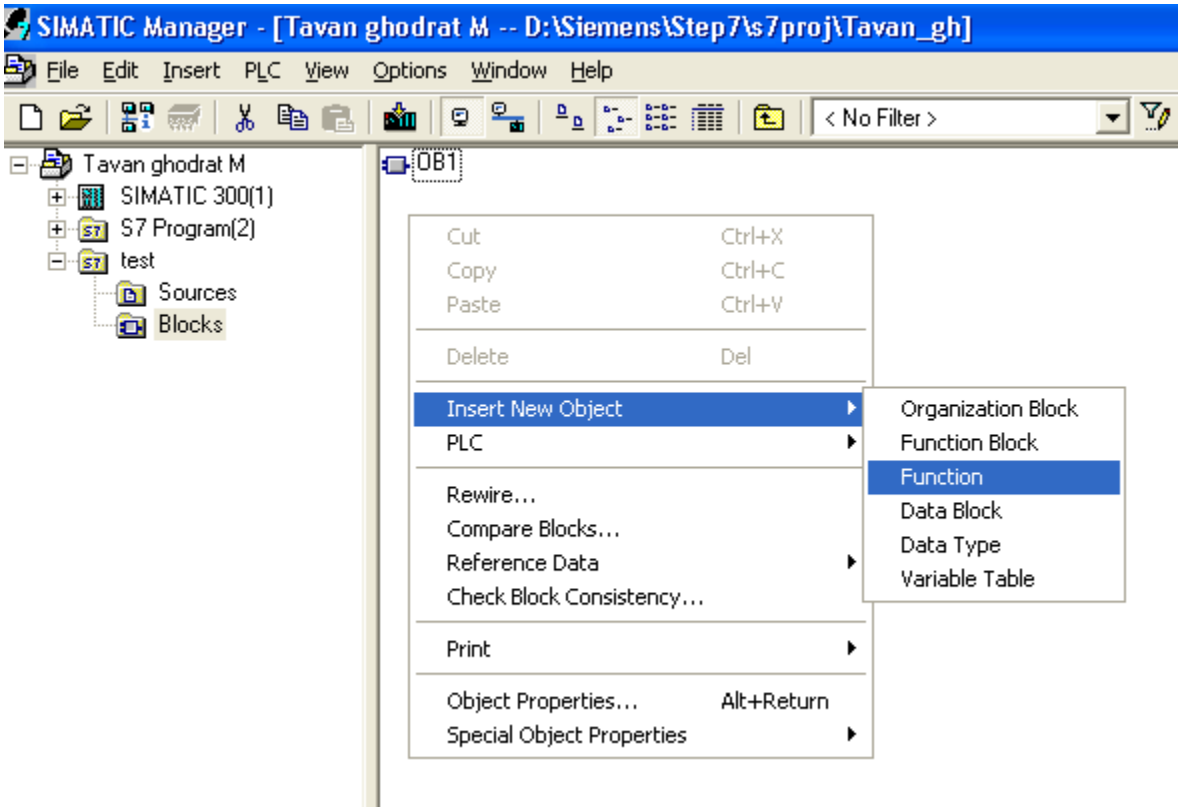
محیط برنامه نویسی فانکشن ها دقیقاً مشابه برنامه نویسی در OB ها می باشد و تمامی دستورات و بلوک های گفته شده نیز در این محیط قابل استفاده می باشند

برای مثال فرض کنید کاربری بخواهد یک مخلوط کن صنعتی را کنترل نماید و این مخلوط کن دارای دو ورودی و یک خروجی مایع باشد در این فرآیند باید دمای مخزن ، سرعت چرخش همزن، غلظت مایع ورودی، غلظت مایع خروجی و فشار داخلی مخزن کنترل گردد. حال اگر کاربر بخواهد تمام پارامتر های کنترلی را در یک OB وارد کند قطعاً در صورت بروز مشکل قادر به تشخیص محل خطا نخواهد بود و شاید مجبور به نوشتن برنامه ای با بیش از 200 Network شود

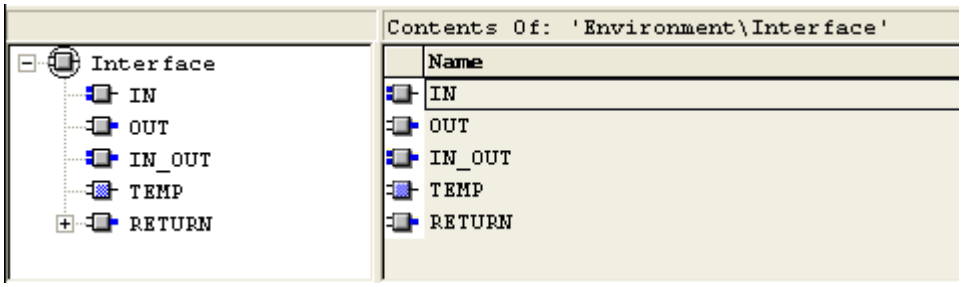
برای جلو گیری از این مشکل کاربر می تواند یک فانکشن بنویسد که فقط عملیات کنترل دما را انجام دهد و فانکشن دیگر عملیات کنترل غلظت و الی آخر. در نهایت نیز همه این فانکشن ها را فراخوانی کرده وسیستم کنترل نهایی را ایجاد می نماید. در صورت بروز مشکل به سراغ فانکشن مربوطه رفته و عیب را رفع می نماید

مراحل ایجاد فانکشن:

برای ایجاد یک فانکشن کافی است در صفحه اصلی برنامه راست کلیک نموده و گزینه Function را انتخاب نمائید مطابق شکل زیر اقدام نمائید



بعد از ایجاد نمودن یک FC کافیست تا روی آن دابل کلیک نمائید تا صفحه مربوطه باز شود تمامی این مراحل مشابه OB1 می باشد. از آنجائیکه کاربر در FC ابتدا برنامه را می نویسد و در OB فراخوانی می کند پس باید یک واسط هایی وجود داشته باشند که این ارتباط را بر قرار نمایند. این واسط ها یا از نوع ورودی اند یا خروجی یا هم ورودی و هم خروجی که در بالای صفحه قابل مشاهده هستند



IN: متغییرهایی که فقط به عنوان ورودی فانکشن قابل تعریف هستند

OUT: متغییر هایی که فقط به عنوان خروجی فانکش تعریف می شوند

IN_OUT: متغییر هایی هستند که به عنوان ورودی به بلوک فانکشن وارد می گردد و پس از انجام عملیات نتیجه در متغییر با همان نام ذخیره می گردد

Temp: متغییر های میانی می باشند که در روند برنامه نویسی FC از آن استفاده می گردد

RETURN: این گزینه برای نشان دادن مقدار بازگشتی می باشد فقط یک عدد از این متغییر می توان داشت که نام آن هم ثابت بوده و قابل تغییر توسط کاربر نیست با عملکرد این پایه در مباحث آنالوگ و بلوک FC105 آشنا شده اید

از آنجائیکه FC برای انجام امور مورد نیاز کاربر به کار می رود می توان هر یک از متغییر های فوق را با نام مورد نظر ذکر کرد در هنگام تعریف متغییر علاوه بر نام آن می توان نوع آن را نیز مشخص نمود سپس برای استفاده از آن به OB مربوطه رفته و از منوی سمت چپ بر روی FC_Blocks کلیک کرده و FC مورد نظر را انتخاب کرده و وارد می کنیم. حال می توان ورودی ها را آدرس دهی کرد

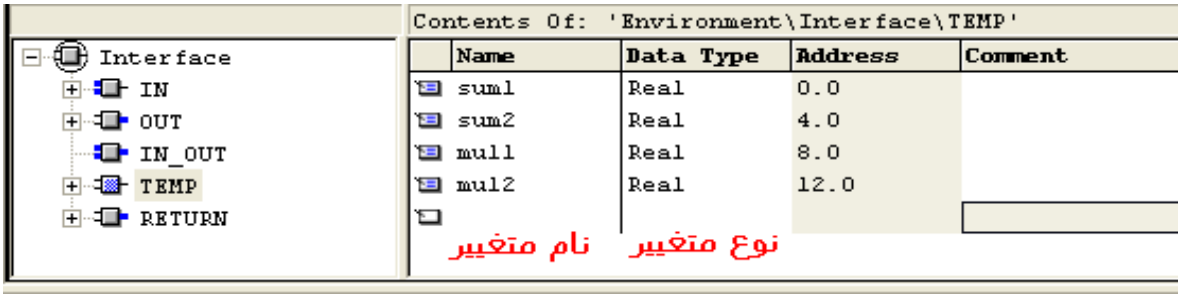
پس از اتمام کار باید OB و FC را با هم دانلود نمود در غیر اینصورت برنامه اجرا نخواهد شد اگر تغییر در FC داده شود باید در OB آن FC را حذف کرد و دوباره وارد نمود و گرنه تغییرات داده شده در FC اعمال نخواهد شد

مثال : با استفاده از FC ها برنامه ای بنویسید که حاصل ضرب 3 عدد را محاسبه نماید و عدد حاصل را بر حاصل جمع این 3 عدد تقسیم نماید

پاسخ: اعداد را با نام های NO1,NO2,NO3 تعریف کنید که به عنوان ورودی ها می باشند متغیر های میانی SUM و MUL به عنوان مقدار حاصل جمع و حاصل ضرب اعداد می باشند و در نهایت نیز ANS نیز به عنوان خروجی این FC می باشد که از تقسیم حاصل ضرب بر حاصل جمع اعداد بدست می آید در شکل زیر نحوه تعریف

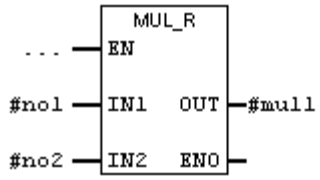
نمودن متغییر های میانی نشان داده شده است avg1 حاصل جمع دو عدد اول و avg2 حاصل جمع 3 عدد می باشد.

برای mul نیز همین روند صدق می کند برای بقیه متغییر ها نیز به همین روند اقدام می شود

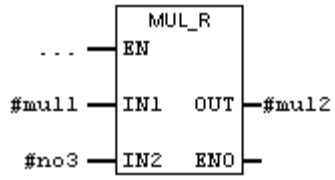


پس از تعریف متغییر ها نوبت به برنامه نویسی است. ابتدا باید سه عدد را جمع نموده و سپس آنها را ضرب کرده و در نهایت عمل تقسیم صورت پذیرد

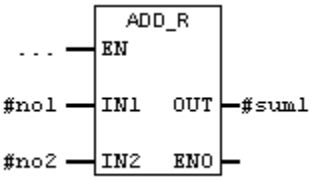
دو عدد اول با هم ضرب می شوند و حاصل در mul1 قرار می گیرد



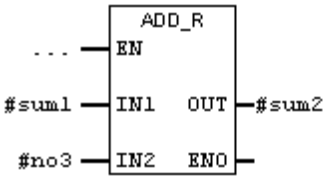
در این مرحله عدد سوم نیز ضرب می شود و حاصل در MUL2 قرار داده می شود



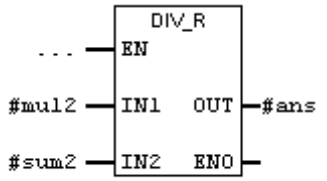
دو عدد اول با هم جمع می شوند و در sum1 قرار می گیرند



عدد سوم نیز جمع شده و در sum2 قرار داده می شود

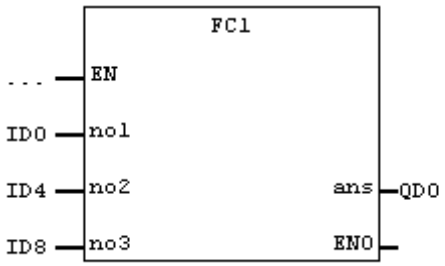


در انتها نیز عمل تقسیم ضرب اعداد بر حاصل جمع آنان انجام می گیرد



حال FC مربوط به این عملیات ریاضی نوشته شده است برای استفاده از آن باید به OB مربوطه رفته و آنرا همان گونه که گفته شد فراخوانی نمود حال بعد از فراخوانی کافیست آدرس ورودی ها و خروجی ها را وارد نمائید

در محیط شبیه سازی به این سه عدد مقدار دهید و خروجی را مشاهده نمائید. دقت کنید که اعداد از جنس Real می باشند

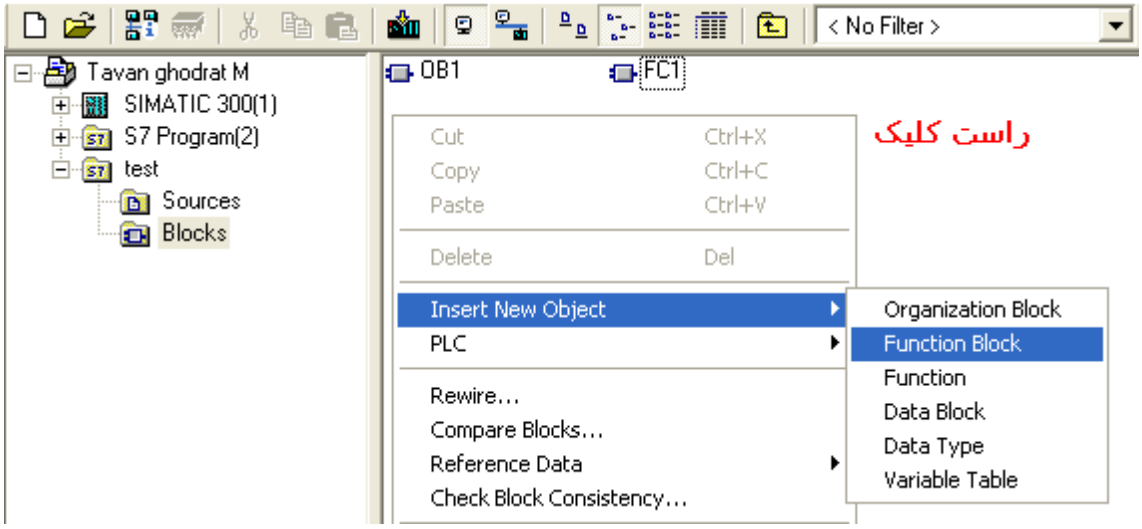


فانکشن بلاک ها (FB):

کاربر می تواند در داخل یک FC از FC های دیگر نیز استفاده نماید. دقت کنید اطلاعات در پایان سیکل زمانی اسکن یک FC صفر می شوند و دوباره اسکن برنامه از ابتدا آغاز می شود این عامل باعث می شود که FC قادر به حفظ حالت از سیکل قبلی نباشد. برای مثال اگر از یک بلوک SR استفاده گردد با فعال شدن پایه set خروجی فقط تا پایان سیکل، یک می ماند و با شروع سیکل بعدی صفر می شود در مورد تایمر و شمارنده‌ها این طور نیست

برای رفع این مشکل باید از فانکشن بلاک ها (FB) استفاده نمود. همان طور که در بخش دیتا بلاک ها (DB) اشاره شده است ، FB این مشکل را با استفاده از DB های اختصاصی رفع می کند. اصول کاری FB مانند FC ها می باشد با این تفاوت که در تعریف واسط ها (Interface) گزینه دیگری به نام STAT نیز اضافه می شود که مربوط به حالت هایی است که کاربر می خواهد آنها را در سیکل بعدی حفظ نماید

برای ایجاد یک FB کافیست در صفحه اصلی راست کلیک نموده و مسیر Insert New Object گزینه Function Blocks را انتخاب نمائید به شکل زیر توجه کنید



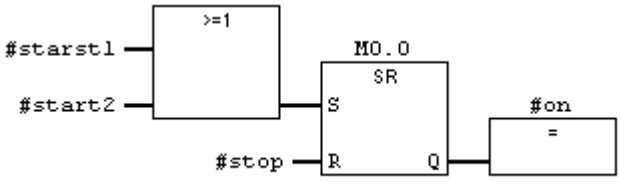
در هنگامی فراخوانی FB در OB باید نام یک DB را به آن اختصاص دهید که اگر این دیتا بلاک وجود نداشته باشد خود نرم افزار برای کاربر آن را ایجاد می نماید. دقت شود که DB ساخته شده از نوع اختصاصی و مخصوص همین FB می باشد

مثال: برای راه اندازی یک موتور دو عدد شستی در فواصل متفاوت برای راه اندازی موتور از دو نقطه در نظر گرفته شده است. از یک سنسور دما برای کنترل دما و یک انکودر برای کنترل سرعت استفاده شده است. یک شستی Stop نیز برای حالت اضطراری در نظر گرفته شده است. بر نامه ای بنویسید که این موتور توسط یکی از این دو شستی روشن شود و هرگاه دما بالاتر از 45.6 درجه سانتی گراد و سرعت آن از 1330rpm بیشتر شد موتور را خاموش کند با فشار کلید stop موتور در هر صورت باید خاموش شود روشن بودن موتور نیز توسط یک چراغ سیگنال مشخص می شود

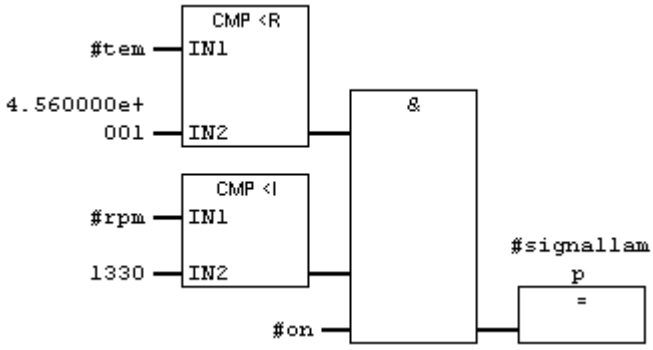
پاسخ: به علت وجود شستی در روند برنامه نویسی ، کاربر نیازمند حفظ حالت آن می باشد پس باید حتماً از FB استفاده کند در مرحله اول باید متغییر های واسط(Interface) تعریف شوند (به متغییر های پایدار دقت کنید)

متغیرهای ورودی : start1,start2,stop,tem,rpm دقت کنید که متغیر دما از نوع Real سرعت از نوع INT می باشد متغییر های پایدار : on , off و متغییر های خروجی : signal lamp

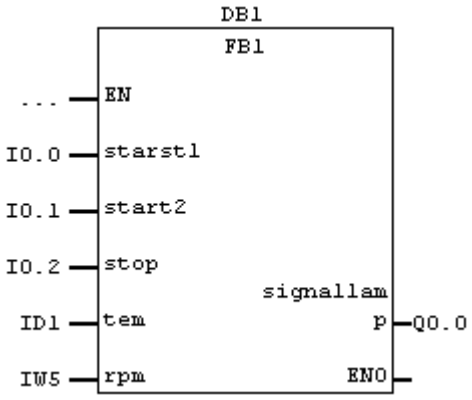
یکی از سوئیچ ها سیستم را راه اندازی می کند و در Network بعدی شرایط کنترلی بررسی می شوند و سیگنال خروجی اعمال می شود



بعد از نوشتن این FB کافیست به OB مربوطه رفته و آن را فراخوانی کنید



بعد از فراخوانی کافیست آدرس های مربوطه را وارد کنید. برای اجرای برنامه باید دقت شود که علاوه بر OB1، باید DB و FB مربوطه نیز دانلود شود وگرنه با خطای سیستمی CPU مواجه می شوید



اگر وارد DB مربوطه شوید می توانید متغییر های مربوطه را مشاهده و مقدار واقعی آن را مشاهده کنید. برای این منظور کافیست پس از باز کردن DB ، مانیتورینگ آن را فعال نمائید

DB Param - [@DB1 -- Tavan ghodrat M\test ONLINE]

Data block Edit PLC Debug View Window Help

	Address	Declaration	Name	Type	Initial value	@Actual val	Actual value
1	0.0	in	starst1	BOOL	FALSE	TRUE	FALSE
2	0.1	in	start2	BOOL	FALSE	TRUE	FALSE
3	0.2	in	stop	BOOL	FALSE	FALSE	FALSE
4	2.0	in	tem	REAL	0.000000e+000	45.0	0.000000e+000
5	6.0	in	rpm	INT	0	200	0
6	8.0	out	signall...	BOOL	FALSE	TRUE	FALSE
7	10.0	stat	on	BOOL	FALSE	TRUE	FALSE
8	10.1	stat	off	BOOL	FALSE	FALSE	FALSE

نوع واسط | نام متغییر | نوع متغییر | مقدار اولیه | مقدار واقعی