

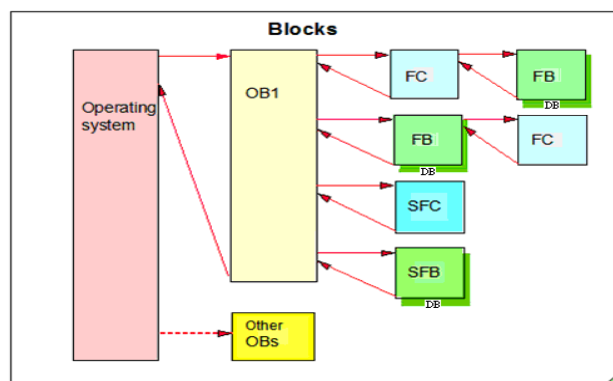
- معرفی برنامه نویسی ساختار یافته و معرفی بلوکهای FC,^۱ FB,^۲ SFC,^۳ SFB^۴ و DB^۵
- بررسی نحوه ایجاد بلوک FC و نحوه فراخوانی آن در بلوکها دیگر
- بررسی نحوه ایجاد بلوک FB و بلوک DB اختصاصی^۶ برای آن و نحوه فراخوانی بلوک FB در بلوکها دیگر
- بررسی نحوه ایجاد DB اشتراکی^۷ و نحوه استفاده از دادههای آن در بلوکها

مقدمه: برنامه نویسی ساختار یافته

در برنامه نویسی برای PLC، می توان برنامه یک پروژه را بر حسب محتوای آن تقسیم بندی کرده و هر قسمت از برنامه را در یک بلوک جداگانه نوشت. سپس بلوکها را بر اساس نیاز در برنامه بلوکهای دیگر فراخوانی کرد. بلوکهای FC و FB بلوکهای هستند که توسط برنامه نویس ایجاد، برنامه ریزی و در بلوکهای دیگر فراخوانی می شوند. فراخوانی FBها در بلوکهای دیگر نیاز به DB اختصاصی است که دادههای مورد نیاز برنامه فراخوانی شده در آن قرار می گیرند. علاوه بر این دادههای این DBها در بخش های دیگر از برنامه قابل استفاده هستند. بلوکهای FC، بلوک DB اختصاصی ندارند دادههای حافظههای محلی این بلوکها فقط در موقع فراخوانی این بلوک معتبر بوده و در خارج از آن اعتباری ندارند

در نرم افزار Simatic Manager بلوکهای به نامهای SFC و SFB وجود دارند که عمل کرد آنها شبیه کار FC و FB هستند این نوع بلوکها را طراح نرم افزار از قبل برای منظورهایی خاص آماده کرده و در اختیار کاربران قرار داده است. برنامههای موجود در این بلوکها شامل برنامههای عمومی مورد نیاز در صنعت اتوماسیون و برنامههای مربوط به بکار گیری ماژولهای خاص سازنده PLC هستند.

برای برنامه ریزی PLC، بلوکهای دیگری به نام OB^۸ها وجود دارند که از پیش توسط طراح نرم افزار برای منظورهایی خاصی آماده شده اند. محتوای این بلوکها توسط کاربر برنامه ریزی و توسط Operating System در زمانهای خاص فراخوانی می شوند. در شکل (۷-۱) جایگاه فراخوانی همه بلوکها در برنامه نشان داده شده است.



شکل (۷-۱)

^۱ Function
^۲ Function Block
^۳ System Function
^۴ System Function Block
^۵ Data Block
^۶ Instance
^۷ Share
^۸ Organization Block

برای تقسیم‌بندی برنامه‌ی یک پروژه، می‌توان برنامه‌های عملیاتی که اجرای آنها تکرار می‌شوند، و همچنین برنامه‌هایی که غیرتکراری بوده ولی بر حسب نوع عمل کرد دسته‌بندی شده اند را در بلوک‌های FC و FB مجزا نوشته و آنها را در برنامه بلوک-های دیگر فراخوانی کرد.

برای عیب‌یابی برنامه طراحی شده و یا عیب‌یابی تجهیزات ابزار دقیق موجود در Plant، می‌توان هر یک از بلوک‌های FC و FB مربوطه را جداگانه پایش^۱ کرد.

در برنامه نویسی برای PLC می‌توان داده‌های یک برنامه را بر اساس مضامین مختلف گروه بندی کرده، و هر گروه از این داده را در یک DB^۲ اشتراکی^۲ بطور مجزا وارد کرد. با ایجاد این نوع بلوک‌ها، می‌توان هر یک از آنها را در قسمت‌های مختلف برنامه باز و از داده‌های آنها در آن قسمت از برنامه استفاده کرد. همچنین در صورت نیاز می‌توان، در موقع اجرای برنامه، داده‌های این بلوک‌ها را پایش کرد.

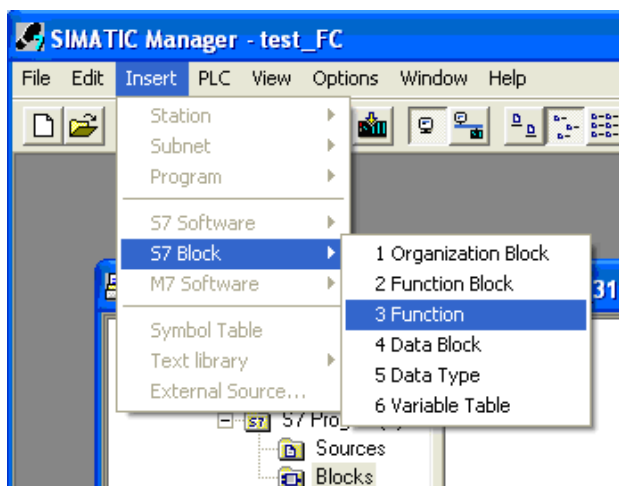
۷-۱) نحوه ایجاد یک بلوک FC، و نحوه فراخوانی آن در بلوک دیگر

مقدمه

در اینجا می‌خواهیم در یک بلوک مناسب برنامه‌ای نوشته شود، که آن برنامه یک عدد مثبت دو رقمی با کد BCD را از ورودی بلوک گرفته و مجذور آنرا بصورت کد BCD سه رقمی به خروجی بلوک منتقل کند، بطوریکه اگر نتیجه مجذور بیشتر از ۹۹۹ شد عدد ۹۹۹ را بجای نتیجه به خروجی منتقل، و وضعیت پیش آمده را توسط یک بیت داده در خروجی بلوک نشان دهد. بلوک مناسب برای این برنامه بلوک FC است که در زیر نحوه ایجاد آن، نحوه نوشتن برنامه در آن و نحوه فراخوانی آن در بلوک دیگر آمده است.

۷-۱-۱) ایجاد بلوک FC.

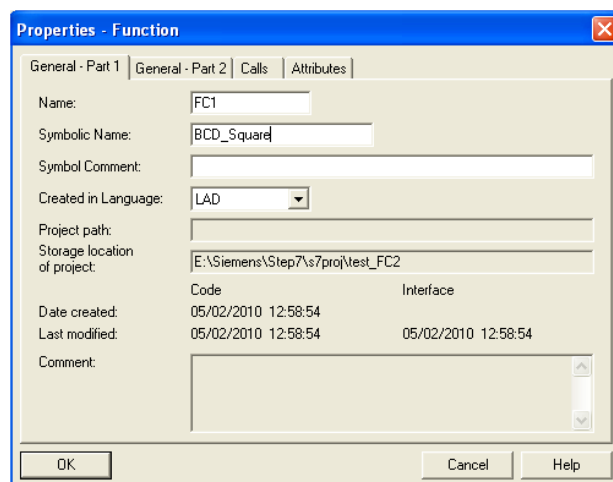
- برای ایجاد یک FC از صفحه Simatic manager، در پنجره پوشه‌های اشیای پروژه، پوشه بلوک انتخاب شده، سپس از مسیر نشان داده شده در شکل (۷-۲) گزینه Function انتخاب می‌شود.



شکل (۷-۲)

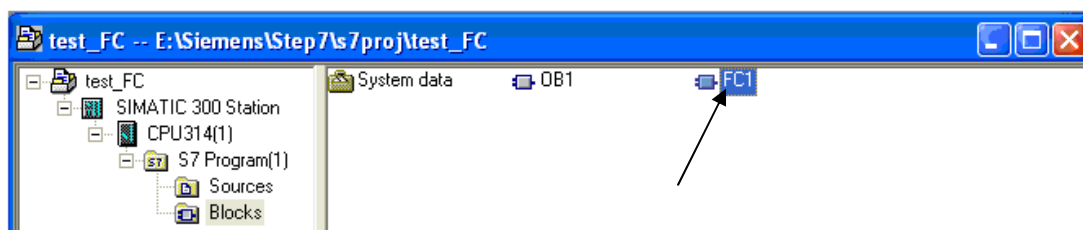
^۱ Monitor
^۲ Share

- با انتخاب گزینه Function، صفحه Properties- Function به شکل (۳-۷) باز می‌شود. در این صفحه می‌توان نام تابع را بصورت سیمبول (که در این مثال BCD_Square است) در محل مربوطه وارد و زبان برنامه‌نویسی مورد نظر را (که در اینجا Ladder است) انتخاب کرد. در پایان با انتخاب دکمه OK این صفحه بسته می‌شود.



شکل (۳-۷)

- پس از بسته شدن صفحه Properties- Function، آیکن Function ایجاد شده (FC)، در پنجره بلوک‌ها بطوریکه که در شکل (۴-۷) نشان داده شده است، ظاهر می‌شود. با دو بار کلیک راست کردن بر روی این آیکن (FC) ایجاد شده محیط برنامه نویسی آن باز می‌شود.



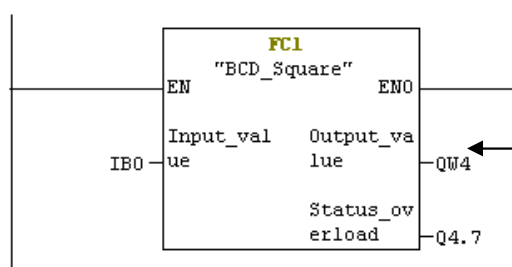
شکل (۴-۷)

۷-۱-۲) نوشتن برنامه در بلوک FC، برای نوشتن برنامه در بلوک FC لازم است متغیرهایی که در برنامه داخل بلوک به کار گرفته شوند در جدول بالای محیط برنامه نویسی FC وارد شوند. متغیرهای موجود در جدول شامل ورودی (IN)، خروجی (OUT)، ورودی- خروجی (IN_OUT)، حافظه موقت (TEMP) و متغیر برگشتی (RETURN) هستند. که در این مثال از همه این متغیرها استفاده نمی‌شود.

متغیرهای استفاده شده در برنامه FC مورد مثال، شامل یک متغیر ورودی (بنام Input_Value) و دو متغیر خروجی (بنام- های Output_Value، Status_Overload) و چهار حافظه موقت (بنامهای Temp_BCD، Temp_int، Temp_square، Temp_Overload) بوده که لازم است وارد جدول شوند. متغیرهای وارد شده در جدول در شکل (۷-۵) نشان داده شده است.

۷-۱-۳) **فراخوانی بلوک FC**، برای اینکه برنامه FC ایجاد شده در برنامه بلوک دیگر که در این مثال بلوک OB1 است فراخوانی شود، لازم است ابتدا از پنجره بلوک‌ها، در صفحه Simatic Manager، بلوک OB1 باز و سپس FC آماده شده در آن فراخوانی شود. در این مثال می‌خواهیم FC1 ایجاد شده یک مرتبه در OB1 فراخوانی شود. بلوک FC1 ایجاد شده، در پنجره Program element در زیر گروه FC قابل دسترسی است. برای فراخوانی این بلوک در بلوک OB1، می‌توان آن را مانند المانهای دیگر در Network ها نصب کرد. در این مثال بلوک FC1 ایجاد شده، بصورت نشان داده شده در شکل (۷-۷) در OB1 به کار گرفته می‌شود. پس از آماده شدن برنامه بلوک OB1، این برنامه Save و به PLC منتقل شود. **توجه شود** در موقع انتقال این بلوک لازم است بلوک FCهای که در آن فراخوانی می‌شوند حتما قبل یا همزمان با آن به PLC منتقل شود. در غیر این صورت PLC متوقف و LED مربوط به خطا (در روی ماژول CPU بصورت قرمز رنگ) روشن می‌شود. اگر PLC در حال Stop باشد با پس یا پیش شدن زمان انتقال بلوک‌ها خطائی رخ نمی‌دهد.

OB1 : Title:
Network 1 : Title:



شکل (۷-۷)

پس از انجام تمام کارهای فوق برنامه را آزمایش و نتایج آنرا بررسی کنید.

د) در برنامه OB1 یک بار دیگر بلوک FC1 را با آدرس‌های IB1، QW2 و Q2.7 فراخوانی کرده عملکرد آنرا پس از انتقال به PLC بررسی کنید

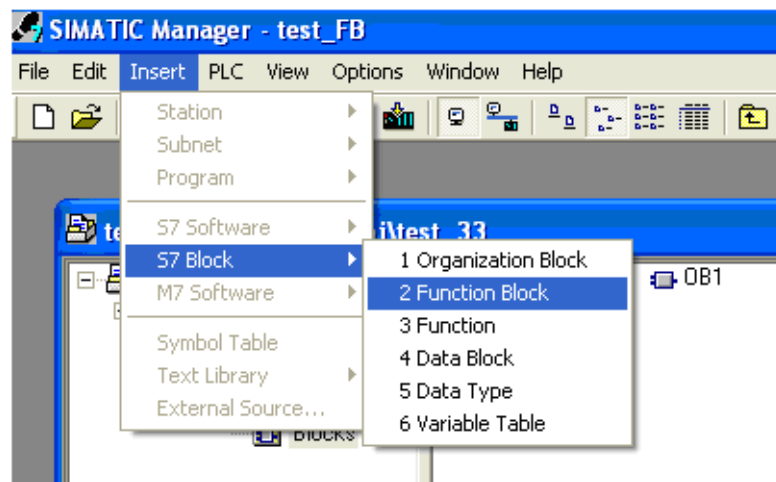
۷-۲) نحوه ایجاد یک FB و نحوه فراخوانی آن در بلوک دیگر

مقدمه

در اینجا می‌خواهیم در یک بلوک مناسب برنامه‌ای نوشته شود که تا آن برنامه دو داده بیتی را از ورودی بلوک دریافت و یک داده بطول Word را به خروجی منتقل کند، بطوریکه به ازاء هر لبه بالا رونده یکی از دو ورودهای بیتی، ۵ واحد به خروجی Word (که فرمت داده آن بصورت کد BCD خواهد بود) اضافه و با یک شدن ورودی بیتی دوم این بلوک مقدار خروجی Word آن صفر شود. بلوک مناسب برای این برنامه FB است که در زیر نحوه ایجاد آن، نحوه نوشتن برنامه در آن و نحوه فراخوانی آن در بلوک دیگر آمده است.

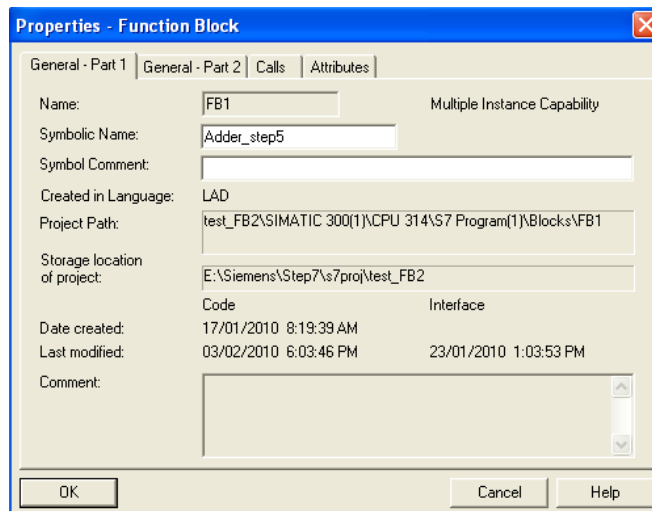
۷-۲-۱) ایجاد Function Block (FB).

- برای ایجاد یک FB در صفحه Simatic manager، در پنجره پوشه‌های اشیاء پروژه، ابتدا پوشه بلوک انتخاب شده. سپس از مسیر نشان داده شده در شکل (۷-۹) گزینه Function Block انتخاب می‌شود.



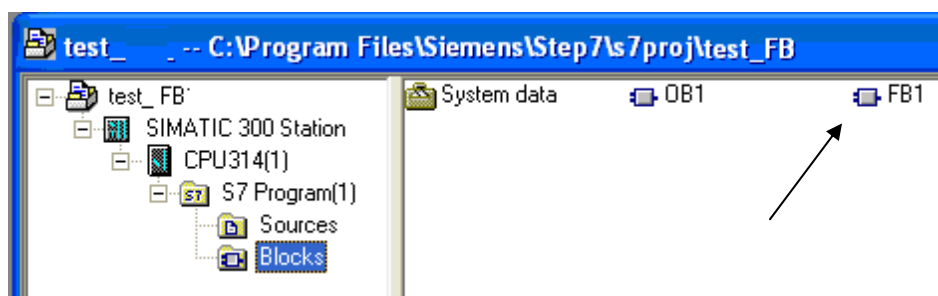
شکل (۷-۹)

- با انتخاب گزینه Function Block، صفحه Properties- Function Block که در شکل (۷-۱۰) نشان داده شده است، باز می-شود. در این صفحه می-توان نام بلوک را بصورت سیمبول (که در اینجا Adder_step5 است) در محل مربوطه وارد و زبان برنامه-نویسی مورد نظر را (که در اینجا Ladder است) انتخاب کرد. در پایان با انتخاب دکمه OK این صفحه بسته خواهد شد



شکل (۷-۱۰)

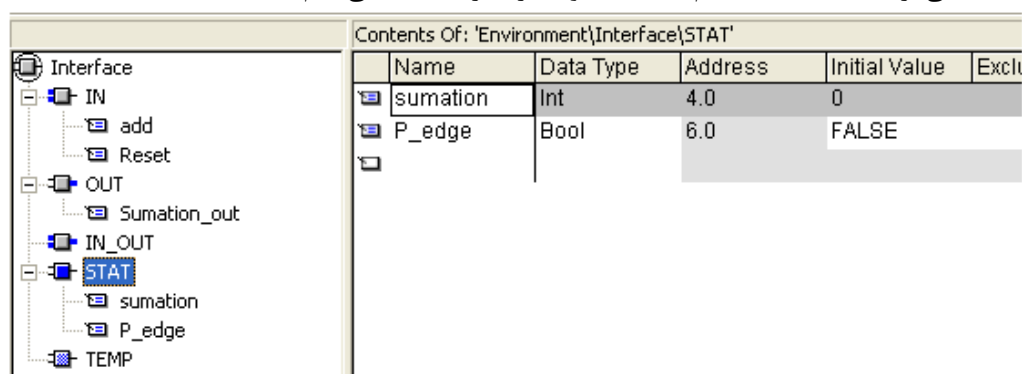
با بسته شدن صفحه فوق، آیکن Function Block ایجاد شده (FB) در صفحه Simatic manager در پنجره بلوک‌ها ظاهر می-شود. FB1 ایجاد شده در شکل (۷-۱۱) در پنجره بلوک‌ها نشان داده شده است. با دو بار کلیک راست کردن بر روی آیکن (FB)، محیط برنامه نویسی مربوط به آن باز می-شود.



شکل (۷-۱۱)

۷-۲-۲) **نوشتن برنامه در FB:** برای نوشتن برنامه در FB، ابتدا لازم است متغیرهای مورد نیاز در برنامه‌ی این را بلوک در جدول بالای محیط برنامه نویسی این بلوک وارد و مقادیر اولیه آن‌ها تنظیم شوند. متغیرهای موجود در جدول شامل ورودی (IN)، خروجی (OUT)، ورودی-خروجی (IN_OUT)، حافظه نگهدارنده داده (Stat) و حافظه موقت (TEMP) هستند. که در این مثال از همه این متغیرها استفاده نمی‌شود.

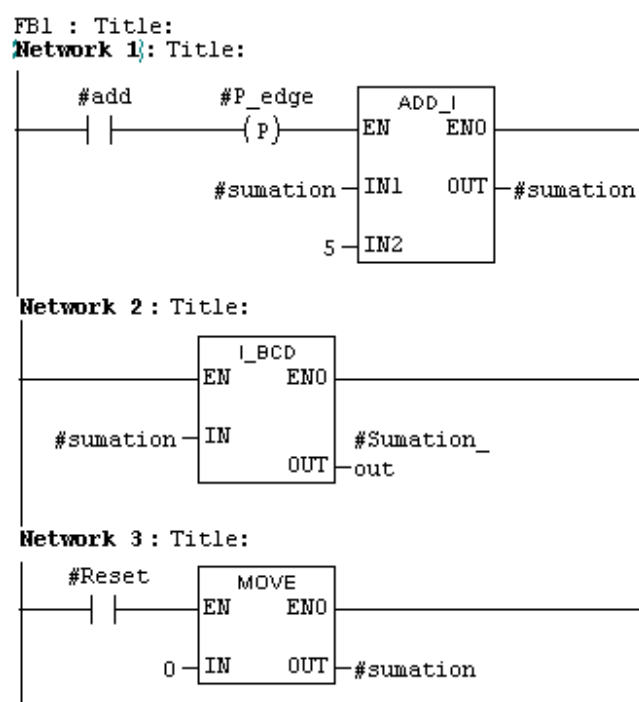
متغیرهای مورد نیاز در برنامه توضیح داده شده در مقدمه این بخش که باید وارد بلوک شوند در شکل (۷-۱۲) نشان داده شده اند. این متغیرها شامل، دو ورودی بیتی (بنام‌های add و Reset) یک خروجی از نوع Word (بنام sumation_out) و دو حافظه Stat که یکی برای داده Integer (بنام sumation) و دیگری برای داده بیتی (بنام P_edge) هستند.



Name	Data Type	Address	Initial Value	Excl
sumation	Int	4.0	0	
P_edge	Bool	6.0	FALSE	

شکل (۷-۱۲)

پس از آماده کردن جدول متغیرها برای بلوک FB، برنامه مربوطه در آن نوشته می‌شود. برنامه مورد نظر در این مثال در شکل (۷-۱۳) ارائه شده است،



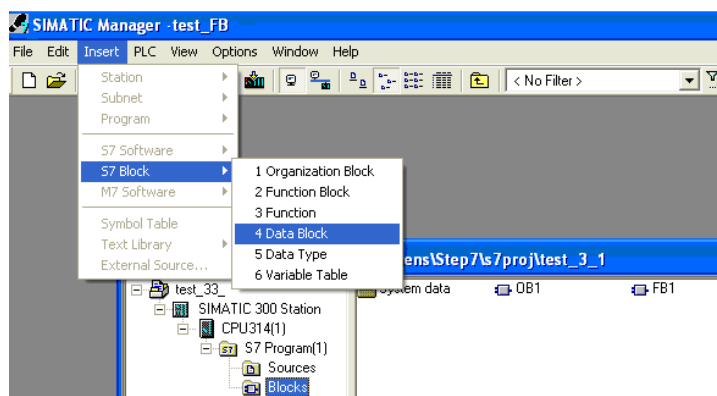
شکل (۷-۱۳)

پس از آماده نمودن برنامه FB، لازم است برنامه Save و به PLC منتقل شود.

توضیح: در برنامه FB می‌توان بجای سیمبول متغیرهای Stat و Temp از آدرس‌های متناظر آنها که در جدول مربوطه آمده-اند استفاده کرد. برای این کار، از حرف L بصورت پیشوند در جلوی آدرس‌های مربوطه که در جدول متغیرها آمده‌اند، استفاده می‌شوند. بطور مثال می‌توان بجای Sumation از آدرس LW4 و بجای P-edge از آدرس L6.0 استفاده کرد.

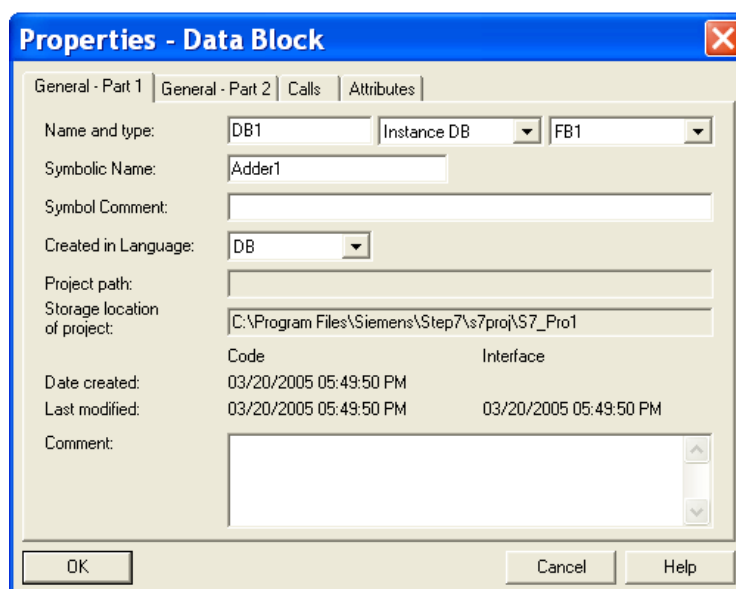
۷-۲-۳) **ایجاد DB اختصاصی**، برای فراخوانی یک FB در بلوکهای دیگر نیاز به دیتا بلوک است. در مثال ارائه شده در این بخش می‌خواهیم FB ایجاد شده دو مرتبه به منظور اجرای دو عملیات با دو دیتا مختلف فراخوانی شود. برای برآورده شدن این منظور لازم است دو دیتا بلوک مجزا با روند زیر ایجاد شود.

الف- برای ایجاد هریک از DB ها، ابتدا در صفحه Simatic manager ، در پنجره پوشه‌های اشیاء پروژه، پوشه بلوک انتخاب شده سپس از مسیر نشان داده شده در شکل (۷-۱۴) گزینه Data Block انتخاب می‌شود.



شکل (۷-۱۴)

ب- با انتخاب گزینه فوق صفحه Properties Data Block به صورت نشان داده شده در شکل (۷-۱۵) باز می‌شود. در این صفحه می‌توان نام بلوک را بصورت سیمبول (که در اینجا Adder1 است) در محل مربوطه و نوع دیتا بلوک را Instance که در این مثال مختص FB1 است، انتخاب کرد. در پایان با انتخاب دکمه OK این صفحه بسته خواهد شد.



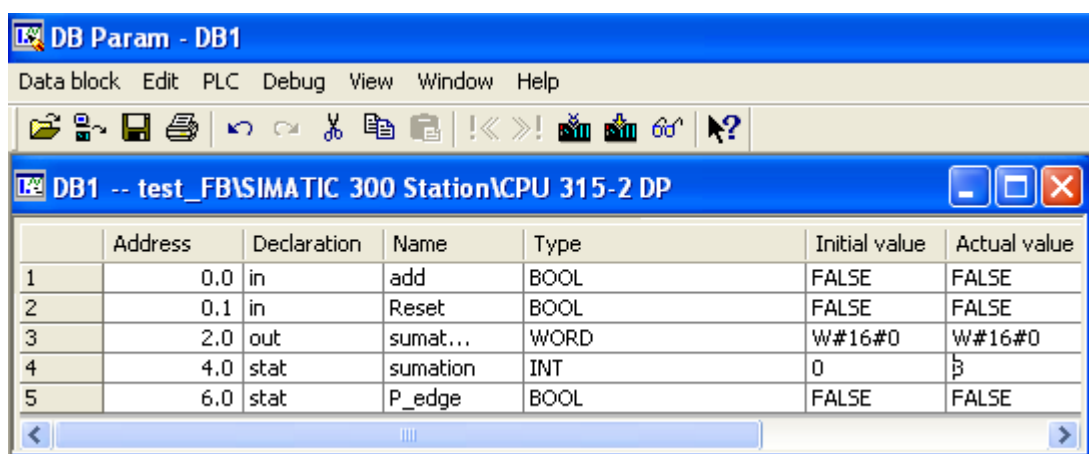
شکل (۷-۱۵)

با بسته شدن صفحه فوق، اکنون Data Block ایجاد شده بصورت DB1 در پنجره بلوک‌ها در صفحه Simatic manager ظاهر می‌شود که عدد یک شماره دیتا بلوک ایجاد شده است.

با روند فوق دیتا بلوک دیگری با نام سیمبولیک Adder2 برای پروژه آماده شود. در پایان لازم است هر دو دیتا بلوک ایجاد شده، از پوشه بلوکها از صفحه Simatic Manager انتخاب و به PLC منتقل شود. **توجه شود** انتقال این نوع بلوکها، باید حتماً همزمان یا قبل از FB مربوطه، به PLC منتقل شوند. در غیر این صورت PLC دچار خطا شده و متوقف می‌شود.

توضیح: اگر لازم شود مقادیر داده‌های مربوط به دیتا بلوکها تغییر یابند، برای اینکار می‌توان هر یک از این دیتا بلوکها را از پوشه بلوکها (در صفحه Simatic Manager) باز و تغییرات لازم را در ستون Actual value جدول آنها اعمال کرد. برای فعال کردن ستون Actual value، باید از منوی View گزینه Data View انتخاب شود.

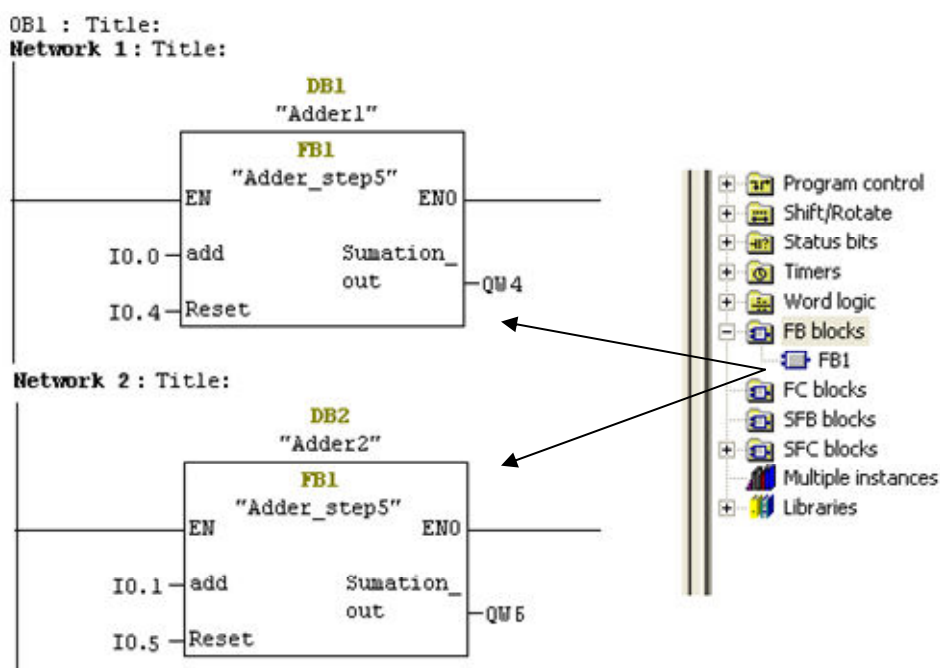
در این مثال می‌خواهیم برای متغیر summation در DB1 مقدار 3 و در DB2 مقدار 1 اعمال شود. در شکل (۷-۱۶) بلوک باز شده DB1 نشان داده شده است. توجه شود پس از هر تغییرات در یک دیتا بلوک، لازم است دیتا بلوک تغییر داده شده Save و مجدداً به PLC منتقل شود



	Address	Declaration	Name	Type	Initial value	Actual value
1	0.0	in	add	BOOL	FALSE	FALSE
2	0.1	in	Reset	BOOL	FALSE	FALSE
3	2.0	out	sumat...	WORD	W#16#0	W#16#0
4	4.0	stat	summation	INT	0	3
5	6.0	stat	P_edge	BOOL	FALSE	FALSE

شکل (۷-۱۶)

ه) **فراخوانی بلوک FB**، برای اینکه برنامه FB ایجاد شده در برنامه بلوک دیگر فراخوانی شود، ابتدا در صفحه Simatic Manager از پنجره بلوکها، بلوک مورد نظر باز و سپس FB ایجاد شده در آن فراخوانی می‌شود. در این مثال می‌خواهیم FB1 ایجاد شده دو مرتبه در OB1 فراخوانی شود. برای این کار لمان FB1 ایجاد شده که در پنجره Progamelement در زیر گروه FB قابل دسترس است، در دو Network جداگانه در بلوک OB1 نصب می‌شوند. در شکل (۷-۱۷) برنامه آماده شده در بلوک OB1 که در آن بلوک دو مرتبه FB1 فراخوانی شده ارائه شده است. این فراخوانی در شبکه یک به همراه DB1 و در شبکه دو به همراه DB2 انجام شده است.

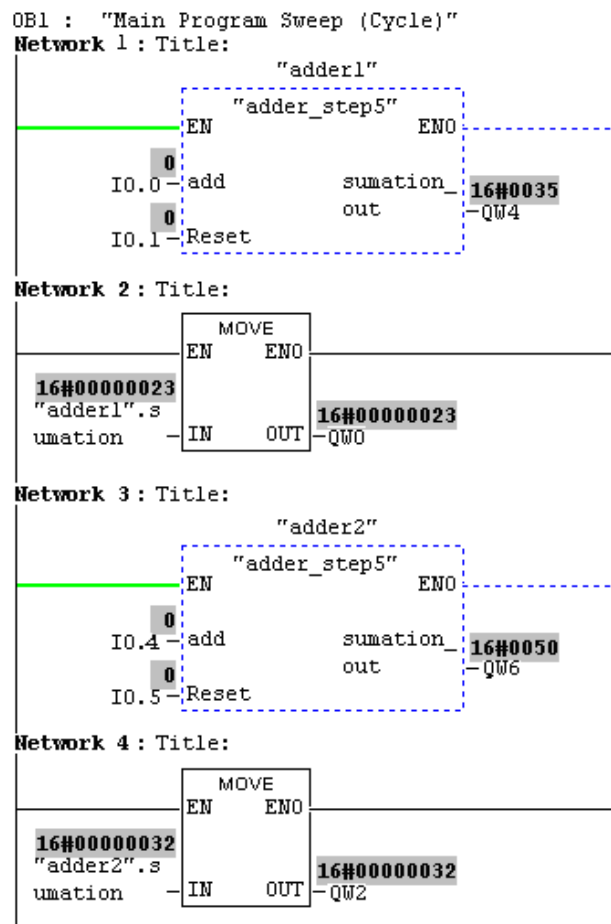


در نهایت بلوک OB1 آماده شده Save و سپس به PLC منتقل می‌شود.

توضیح، حتما لازم است همزمان و یا قبل از انتقال OB1 به PLC، بلوک‌های FB و DB بکار گرفته در برنامه OB1 به PLC منتقل شوند. برای انتقال همه بلوک‌ها بطور همزمان می‌توان در صفحه Simatic Manager از پوشه بلوک‌ها همه بلوک‌ها را همزمان انتخاب و به PLC منتقل کرد. اگر PLC در حال Stop باشد با تقدم و یاخیر در انتقال بلوک‌های خطائی رخ نمی‌دهد. پس از انجام تمام مراحل فوق پروژه را آزمایش و در صورت وجود اشکال احتمالی آنرا رفع و نتایج پروژه را بررسی کنید.

۷-۲-۴) بررسی استفاده از داده‌های دیتا بلوک‌ها در خارج از FBها

برای بررسی استفاده از داده‌های دیتا بلوک در خارج از FBها دو Network جدید بر طبق شکل (۷-۱۷) به برنامه اضافه کرده سپس اجرای این برنامه را بررسی و به سؤال زیر پاسخ دهید.

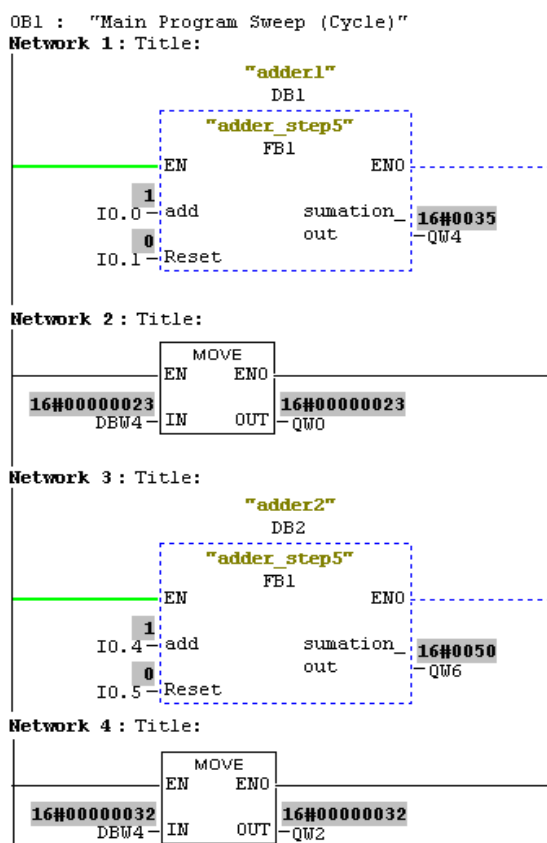


شکل (۷-۱۷)

سؤال: خروجی‌های QW6 و QW0 معادل کدام متغیرهای DB بلوک‌های adder2 و adder1 هستند و چرا؟

توضیح: متغیرهای یک DB را می‌توان بصورت سیمبول و یا غیر سیمبول در خارج از بلوک آدرس دهی کرد. **توجه شود** در صورتی می‌توان از آدرس دهی بصورت سیمبول استفاده کرد که DB مربوط هم نام سیمبول داشته باشد. برای مثال بطوریکه در برنامه OB1 در شکل (۷-۱۷) مشاهده می‌شود در شبکه ۲ بجای آدرس DB1.DBW4 از شکل آدرس‌دهی سیمبولیک آن، یعنی بجای DB1 از سیمبول adder1 و بجای DBW4 از سیمبول sumation استفاده شده است.

توضیح ۲: آدرس‌های شبکه‌های دو و چهار شکل (۷-۱۷) را مانند شکل (۷-۱۸) تغییر داده و کار برنامه را مجدداً بررسی کنید. بطوریکه مشاهده می‌شود، ورودی‌های بلوک‌های MOVE در شبکه‌های ۲ و ۴ این شکل مانند هم هستند، ولی مقادیر خروجی‌های این دو شبکه باهم متفاوت هستند. دلیل آن این است که هر موقع FB فراخوانی شود DB مربوطه به آن باز می‌شود. با باز شدن DB مربوطه می‌توان بدون ذکر آدرس DB بلوک باز شده، از دیتای آن (در بلوک فراخوانی شده FB) استفاده کرد، تا زمانی که DB جدیدی باز نشود (در اینجا یعنی تا FB دیگری اجرا نشود) می‌توان از داده‌های DB باز شده استفاده کرد.

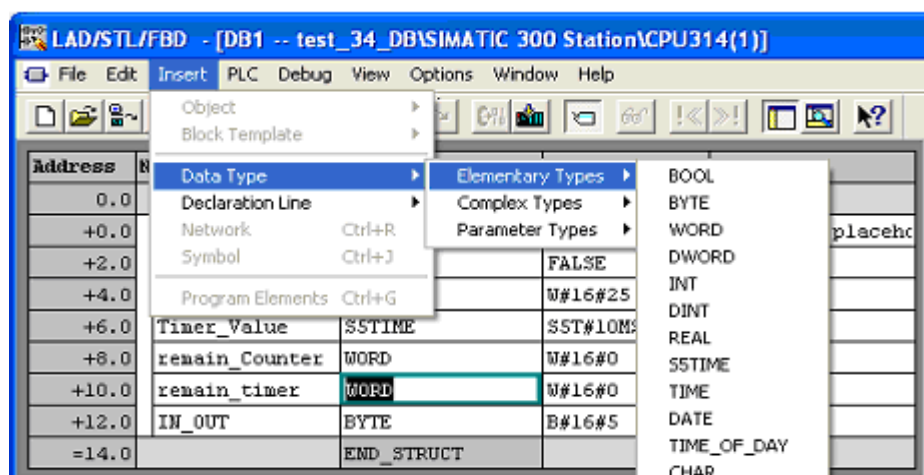


شکل (۷-۱۸)

۷-۳ ایجاد Data Block (DB) اشتراکی^۱ و نحوه بکارگیری آن در بلوک‌ها:

الف) ایجاد DB اشتراکی، برای ایجاد DB از نوع Shared، همانند روند ایجاد DB از نوع Instance که در بخش ۷-۲-۳ آمده است عمل می‌شود، با این تفاوت که در صفحه Data block properties بجای Instance عنوان Shared انتخاب می‌شود. در اینجا، به منظور تمرین استفاده از Data Block در برنامه، دو دیتا بلوک ایجاد می‌شوند. پس از ایجاد دو دیتا بلوک، برای وارد کردن داده‌ها در آنها، از صفحه Simatic Manager از پنجره بلوک‌ها هر یک از آنها (بصورت Declaration View) باز و متغیرهای مورد نیاز برای اجرای برنامه در جدول آنها وارد می‌شوند. یکی از ستون‌های جدول دیتا بلوک مربوط به نوع داده (Data Type) است. در شکل (۷-۱۹) مسیر انتخاب نوع داده‌ها برای این ستون نشان داده شده است.

^۱ Share Data Block



شکل (۷-۱۹)

در شکل‌های (۷-۲۰) و (۷-۲۱) دیتاهای وارد شده در جدول دو دیتا بلوک نمایش داده شده‌اند. در این شکل‌ها، DB1 بصورت Declaration View و DB2 بصورت Data View نمایش داده شده‌اند. در نمایش Declaration View می‌توان متغیرها را وارد جدول کرده و در نمایش Data View می‌توان مقادیر Actual Value آنها را تغییر داد. این دو View از منوی View قابل انتخاب هستند.

پس از آماده کردن هر یک از دیتا بلوک‌ها آنها را save و به PLC منتقل کنید.

LAD/STL/FBD - [DB1 -- test_34_DB\SIMATIC 300 Station\CPU314(1)]				
File Edit Insert PLC Debug View Options Window Help				
Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DB_VAR	INT	0	Temporary placeh
+2.0	Set_Counter	BOOL	FALSE	
+4.0	offset_counter	WORD	W#16#25	
+6.0	Timer_Value	S5TIME	S5T#10MS	
+8.0	remain_Counter	WORD	W#16#0	
+10.0	remain_timer	WORD	W#16#0	
+12.0	IN_OUT	BYTE	B#16#5	
=14.0		END_STRUCT		

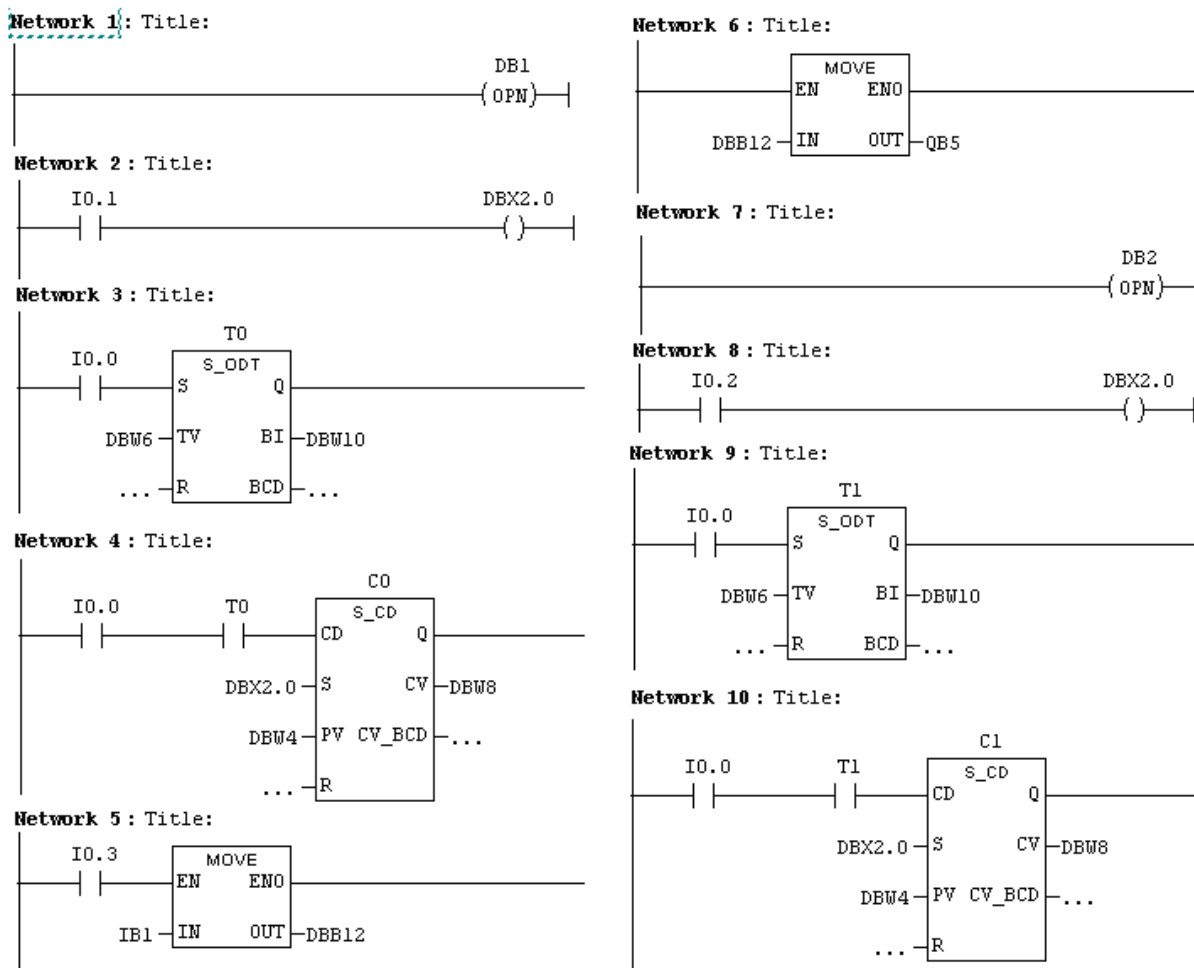
شکل (۷-۲۰) Declaration view

LAD/STL/FBD - [DB2 -- test_34_DB\SIMATIC 300 Station\CPU314(1)]					
File Edit Insert PLC Debug View Options Window Help					
Address	Name	Type	Initial value	Actual value	Comments
0.0	DB_VAR	INT	0	0	Tempo
2.0	Set_Counter	BOOL	FALSE	FALSE	
4.0	Offset_counter	WORD	W#16#50	W#16#40	
6.0	Time_value	S5TIME	S5T#30MS	S5T#2S	
8.0	remain_Count	WORD	W#16#0	W#16#0	
10.0	remain_time	WORD	W#16#0	W#16#0	

شکل (۷-۲۱) Data View

ب) استفاده از DB اشتراکی در برنامه :

در برنامه ارائه شده در شکل (۷-۲۲) نحوه استفاده از دو دیتا بلوک (DB1 و DB2) آمده است، این برنامه را آماده کرده، سپس آنرا save و به PLC منتقل کنید. پس از انتقال برنامه عملکرد آنرا بررسی و به سؤال‌های آمده در زیر پاسخ دهید؟



شکل (۷-۲۲)

سؤال ۱: دیتای DB1 در کدام شبکه‌های این برنامه بکار گرفته شده است؟

سؤال ۲: دیتای DB2 در کدام شبکه‌های این برنامه بکار گرفته شده است؟

سؤال ۳: مقادیر Time Value زمان‌سنج‌های T_0 و T_1 در این برنامه چقدر هستند؟

سؤال ۴: مقادیر Pre Value شمارنده‌های C_0 و C_1 در این برنامه چقدر هستند؟

سؤال ۵: از صفحه DB1 Simatic Manager بلوک را باز کرده، سپس با یک کردن ورودی I0.3 و با تغییر مقادیر IB1 بررسی کنید کدام یک از داده‌ها مقدار آن تغییر می‌کند.