به نام خدا

دستورکار کنترل صنعتی

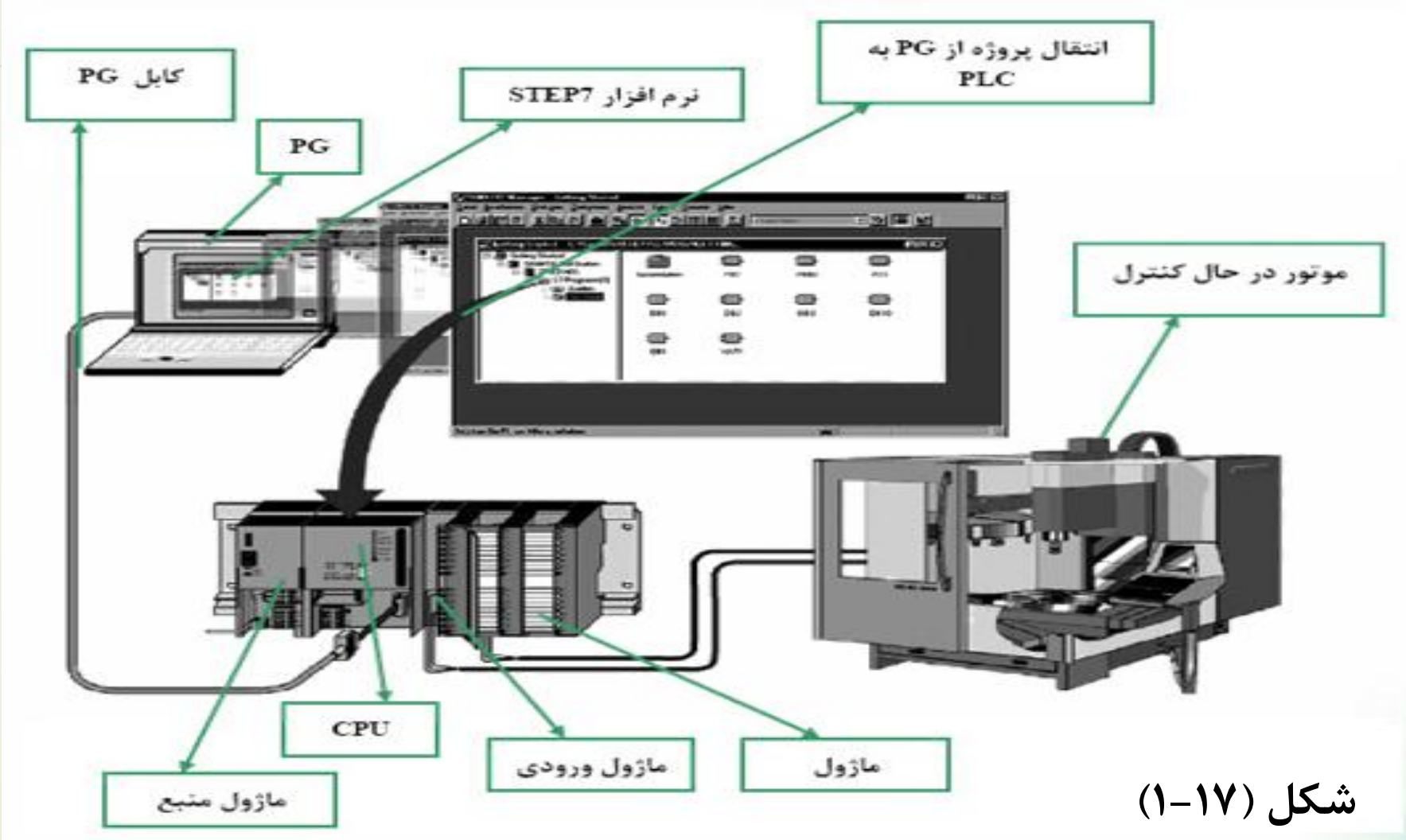
(فصل هفتم)

دکتر فلاح

نحوه کار با step 7

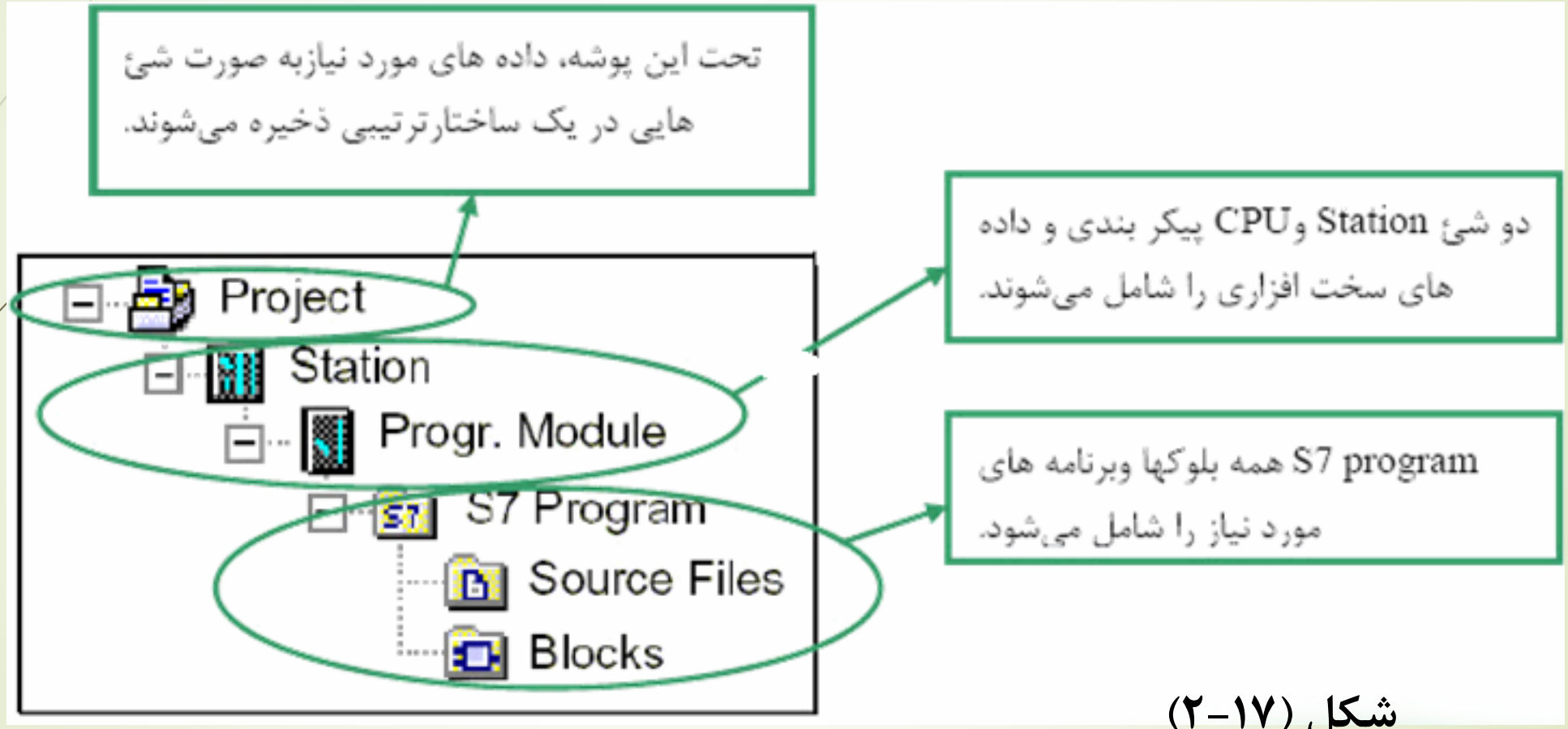
به وسیلهٔ نرم افزار S7 می توان همه داده ها و برنامه ها را اهم از داده های مربوط به پیکره بندی سخت افزاری ، پارامترهای اختصاص یافته به ماژول ها ، داده های مربوط به شبکه های ارتباطی و برنامه ها ی نوشته شده برای ماژول های قابل برنامه نویسی در ساختار یک پروژه ذخیره و سازمان دهی نمود. شکل (۱۷-۱) نمایی از یک PLC با راه اندازی برنامه S7 به همراه یک واسط برنامه نویسی را نشان می دهد. که در آن PG یک واسط برنامه ریزی می باشد. اساس استفاده از محیط نرم افزاری STEP 7 بدین صورت است که کاربر، پوشه ای به عنوان پروژه در آن اجرا می کند و همه مشخصات سخت افزاری و نرم افزاری PLC مورد استفاده و برنامه نویسی ماژول های قابل برنامه نویسی آن را برای طراح اتوماسیون در شی هایی تحت آن پوشه در PG ذخیره می کند . و این اطلاعات از طریق کابلی که PG را به PLC وصل می کند ، منتقل می شود.شکل (۱۷-۲) ساختار کلی یک پروژه و اجزای آن را پس از ایجاد در محیط S7 نشان می دهد.

نحوه کار با step 7



شکل (۱۷-۱)

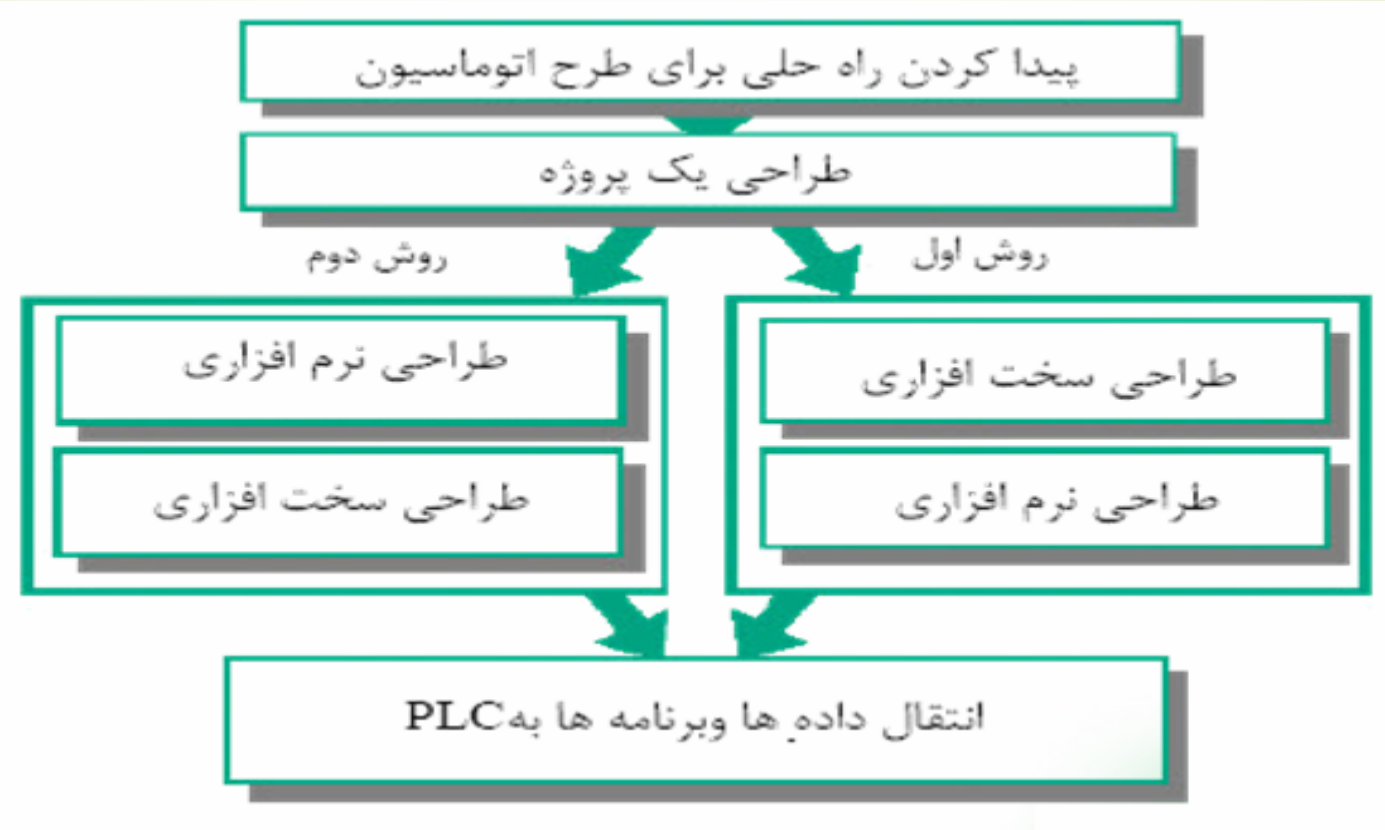
نحوه کار با step 7



شکل (۱۷-۲)

نحوه کار با step 7

مراحل طراحی یک پروژه کنترلی به شکل (۱۷-۳) می باشد.



شکل (۱۷-۳)

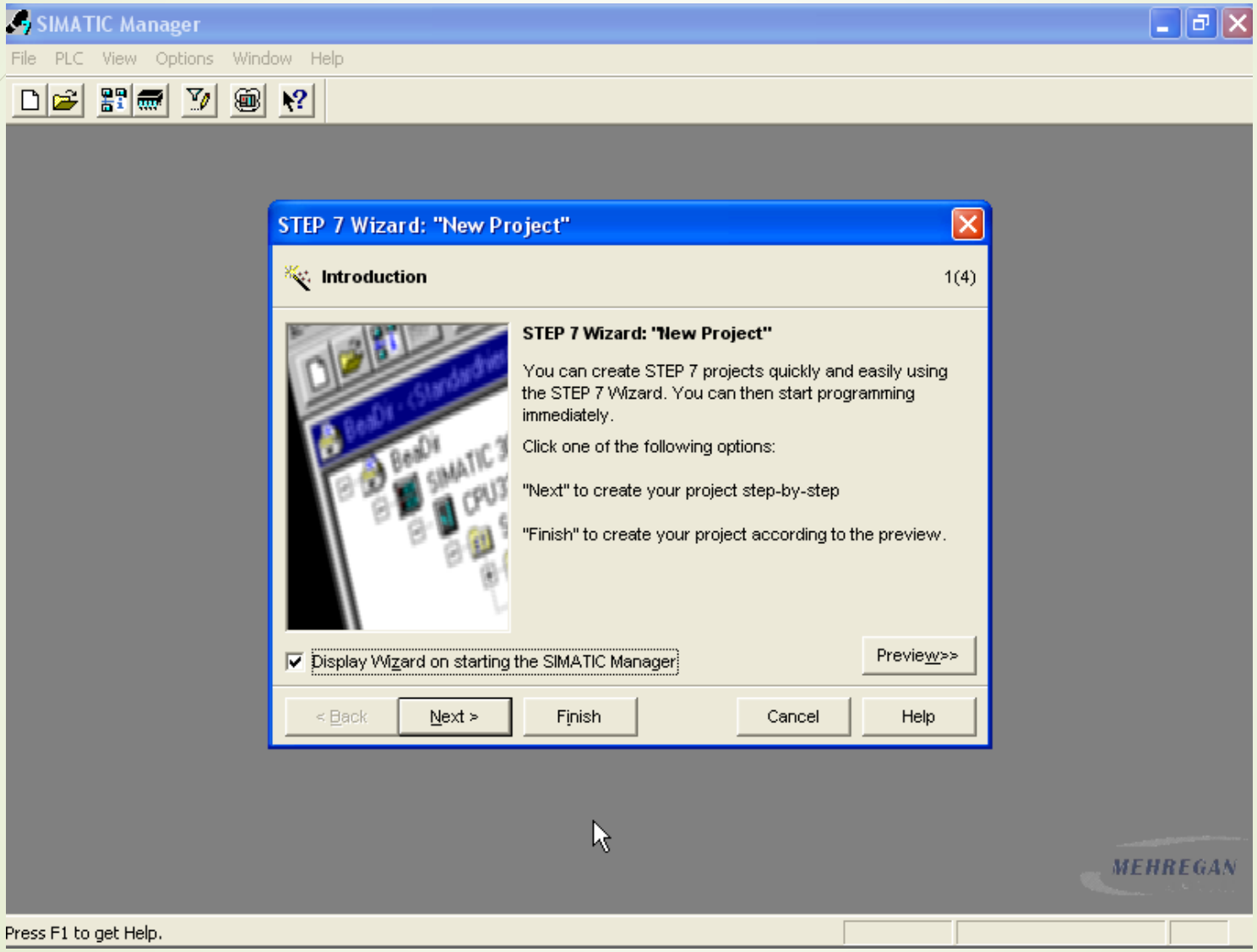
نحوه کار با step 7

منظور از طراحی یک پروژه انتخاب Station ، Cpu و در صورت لزوم دیگر ماژول های قابل برنامه نویسی نظیر FM ، CP و انتخاب OB های مورد نیاز است. برای طرح یک اتوماسیونکه دارای ورودی ها و خروجی های زیادی است ، بهتر است طراحی سخت افزار PLC را قبل از نرم افزار انجام دهیم . مزیت این انتخاب در این است که در پیکره بندی سخت افزاری S7 به طور خودکار ورودی ها و خروجی ها را آدرس دهی می کند، ولی اگر در ابتدا پروژه را طراحی نرم افزاری کنیم ، باید همه آدرس دهی های مربوطه را خود تعیین کنیم . در حالی که ممکن است برخی از آدرس های انتخابی به دلایلی برای S7 قابل استفاده نباشند.

ایجاد یک پروژه با استفاده از Wizard به روش مستقیم :

پس از ورود به محیط نرم افزاری S7 ، Step7 Wizard به صورت پیش فرض به شکل (۱۷-۴) اجرا خواهد شد.

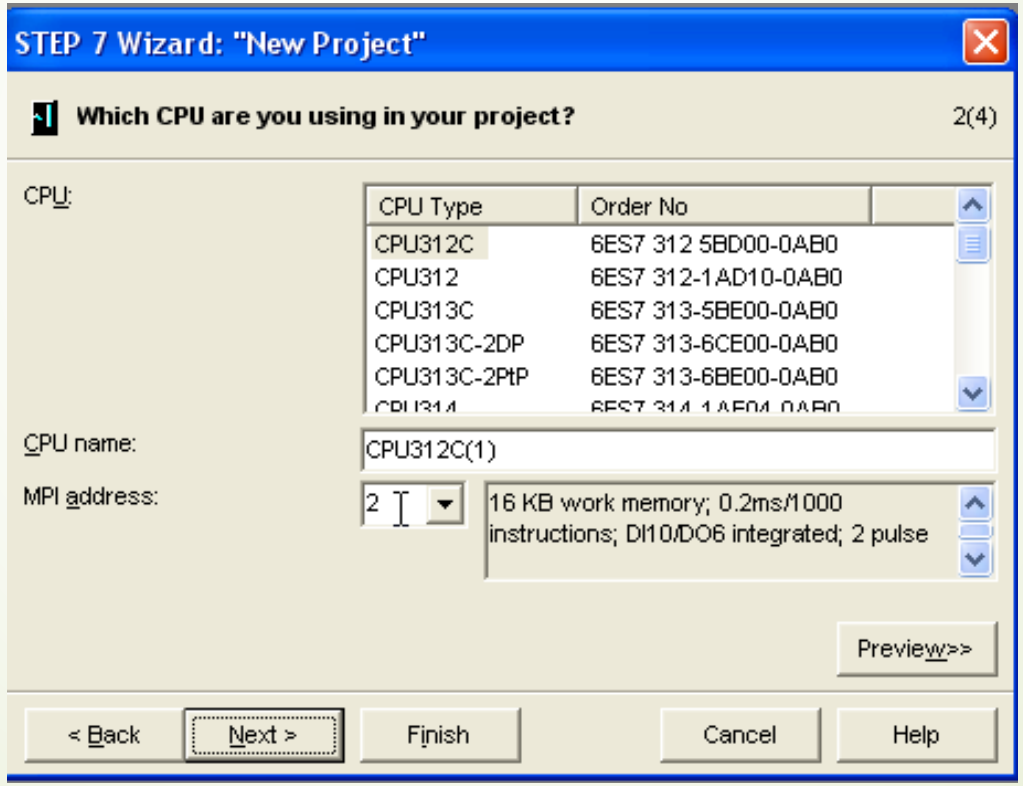
نحوه کار با step 7



شکل (۱۷-۴)

نحوه کار با step 7

با کلیک کردن روی **Next** به ترتیب چند کادر ارتباطی جهت انتخاب اجزای اصلی پروژه به وجود می آیند. اولین کادر ارتباطی در شکل (۱۷-۵) آمده است.



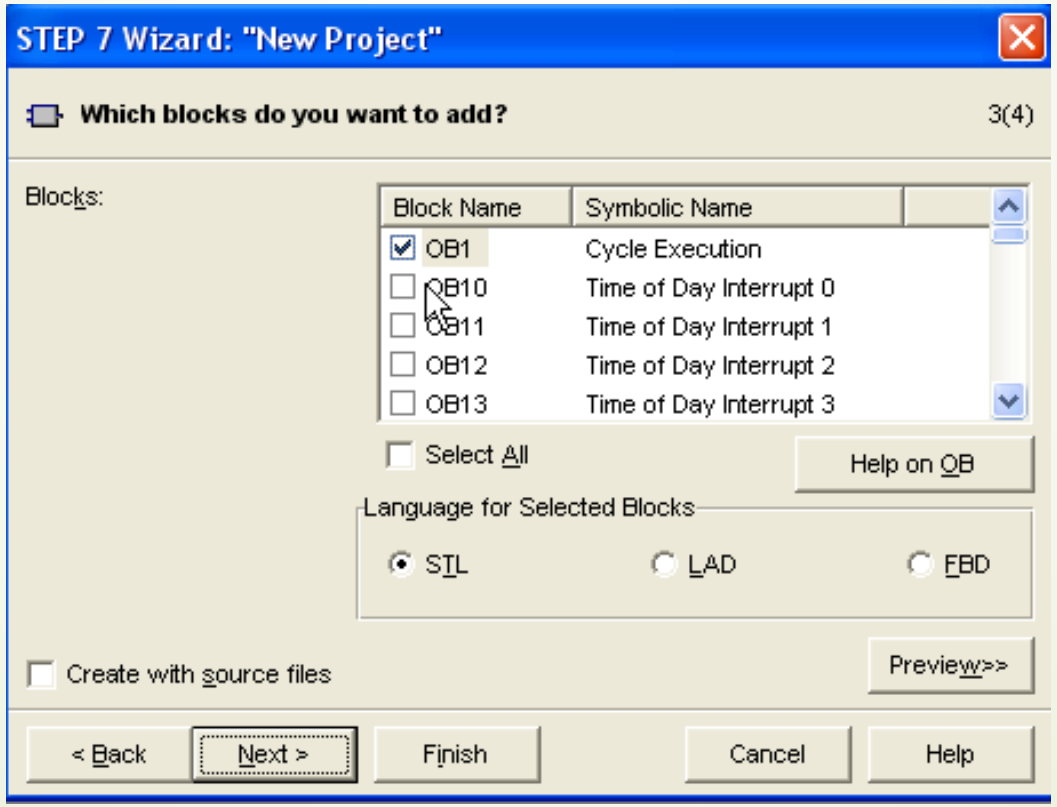
شکل (۱۷-۵)

نحوه کار با step 7

در این کادر می توان نوع Cpu و آدرس MPI آن را تعیین نمود. دلیل انتخاب Cpu قبل از برنامه نویسی این است که مشخصات هر Cpu از قبیل ساختار حافظه و فضای آدرس دهی آن که مختص به همان Cpu است، باید در برنامه نویسی لحاظ شود.آدرس واسط ارتباط چند نقطه ای برای ارتباط Cpu با ابزار برنامه ریزی مورد نیاز است، این آدرس به طور پیش فرض ۲ می باشد و معمولا نیازی به تغییر آن نیست ، مگر آنکه چندین Cpu از طریق گذرگاه MPI شبکه شده باشند. که در این صورت چون هر Cpu به یک آدرس MPI منحصر به فرد نیاز دارد. می توان آدرس Cpu در گذرگاه MPI را در اینجا تعیین نمود ، با کلیک کردن روی Next عملیات انجام شده در این مرحله تایید می شود و کادر ارتباطی بعدی باز می شود ، که در شکل (۱۷-۶) آمده است. این کادر ارتباطی برای انتخاب OB های مورد نیاز، استفاده می شوند ، چون برای همه برنامه ها به OB1 نیاز است و ساختار اصلی برنامه کاربر در این بلوک نوشته می شود، به طور پیش فرض انتخاب می شود. زبان برنامه نویسی را نیز در این کادر می توان انتخاب کرد. البته در پنجره برنامه نویسی ، هر بلوک قابل برنامه نویسی

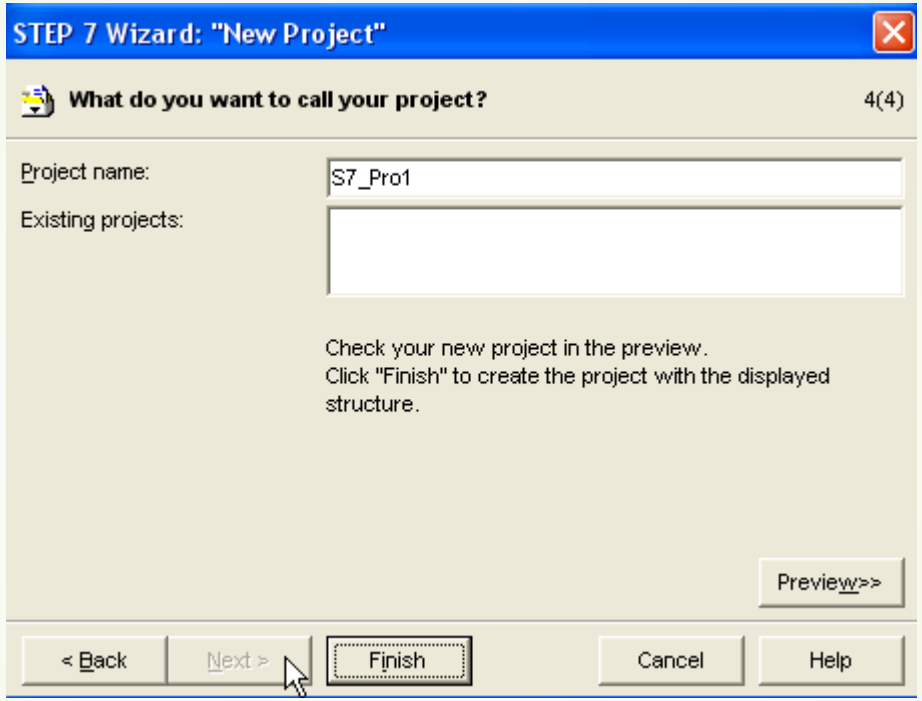
نحوه کار با step 7

تغییر زبان برنامه نویسی قابل امکان است.با کلیک کردن بروی **Next** تنظیمات انجام شده **Save** می شود و کادری به شکل (۱۷-۷) باز می شود.



شکل (۱۷-۶)

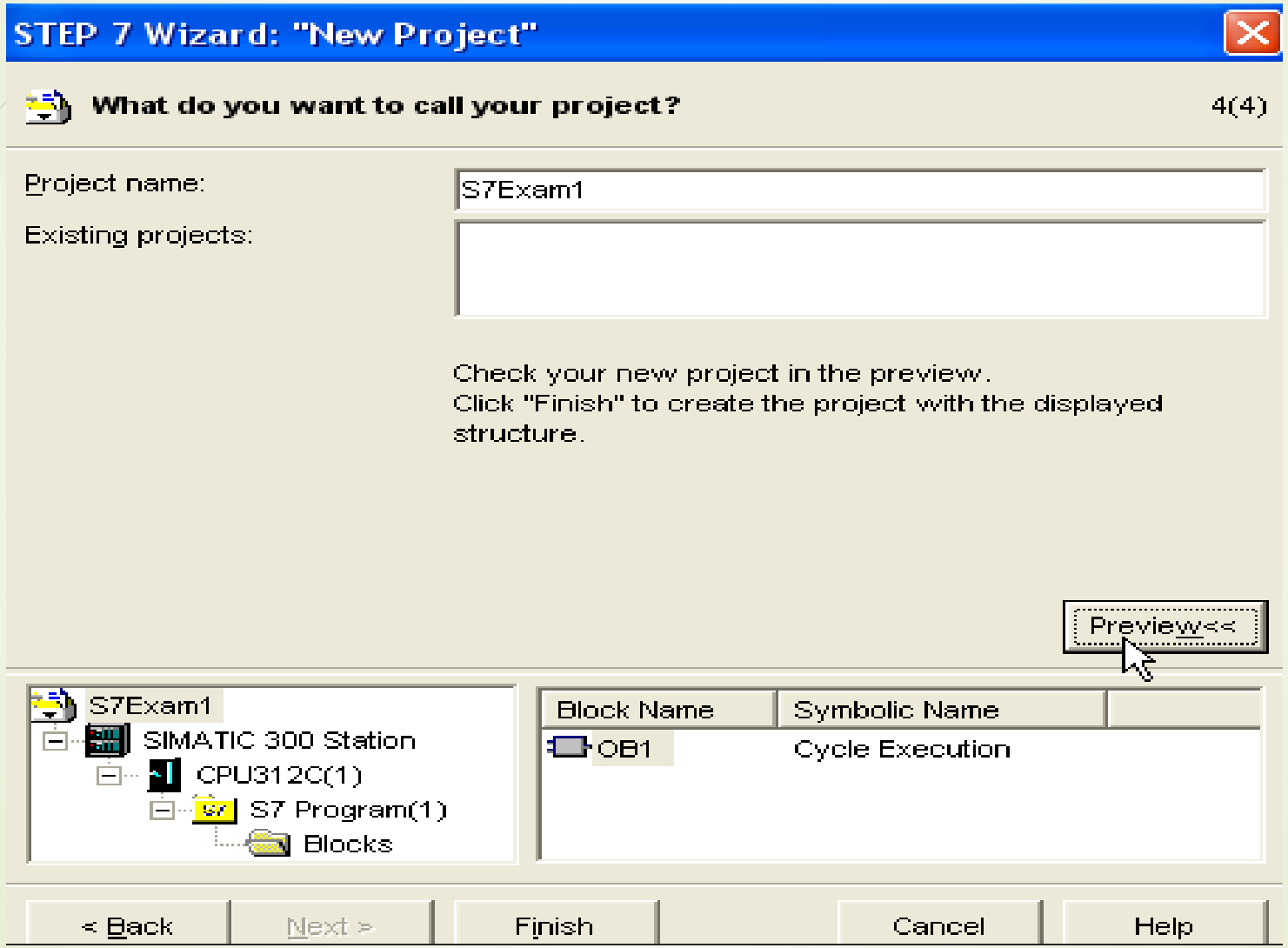
نحوه کار با step 7



شکل (۱۷-۷)

در قسمت **Project name** نام پروژه را نوشته و اگر برروی **Preview** کلیک کنیم . ساختار شکل گیری پروژه از ابتدا تا مرحله بعدی را نمایش می دهد. در شکل (۱۷-۸) این نمایش، نشان داده شده است.

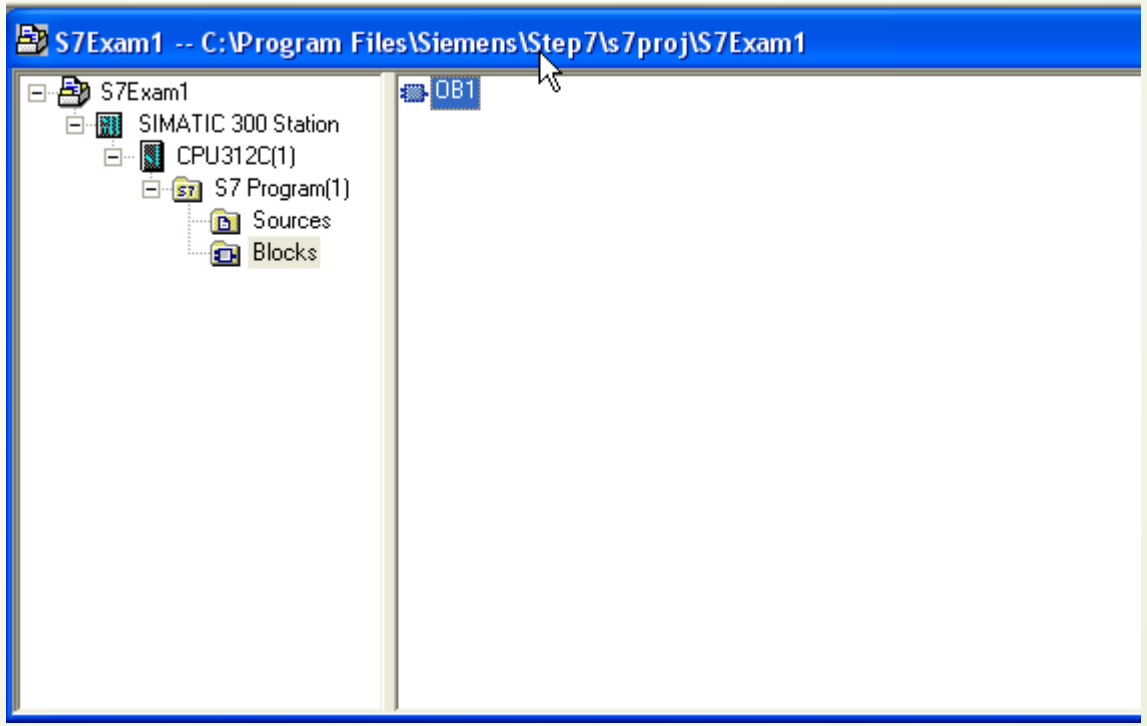
نحوه کار با step 7



شکل (۱۷-۸)

نحوه کار با step 7

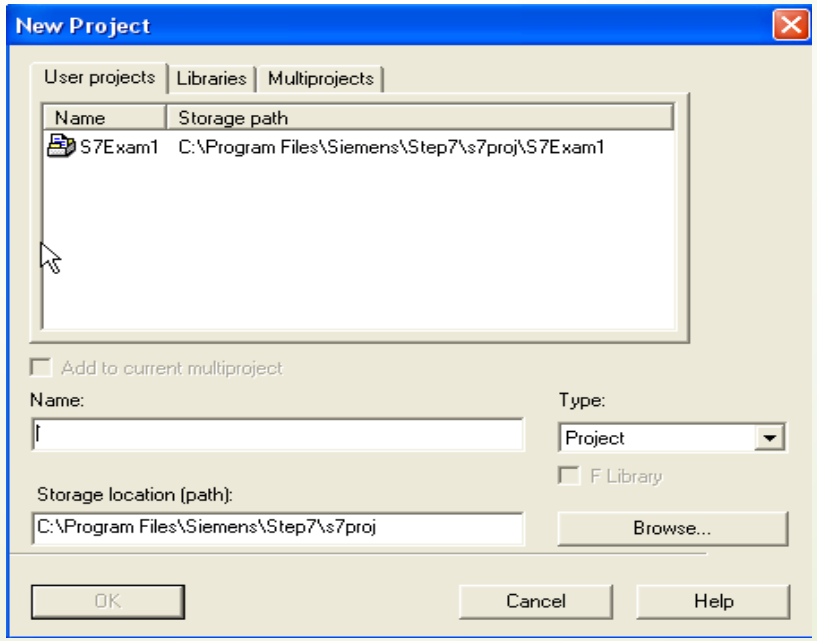
حال اگر روی finish کلیک کنیم ، پروژه تشکیل می شود، به این ترتیب به پنجره ای به شکل (۱۷-۹) می رسیم . این پنجره را ، پنجره پروژه می نامیم.



شکل (۱۷-۹)

نحوه کار با step 7

همانطور که در مراحل قبل ملاحظه کردید، Step 7 Wzard در بدو ورود به برنامهٔ S7 ظاهر می شود، ولی می توان این تنظیم پیش فرض را در اولین کادر ارتباطی Wizard غیر فعال نمود. در این صورت می بایست هر شی از پروژه را به طور دستی فعال کرد. به منظور ایجاد یک پروژه در نوار منو ، گزینه ی file > new را اجرا کرده ، به این ترتیب کائر ارتباطی نام گذاری پروژه به شکل (۱۷-۱۰) باز می شود.



شکل (۱۷-۱۰)

نحوه کار با step 7

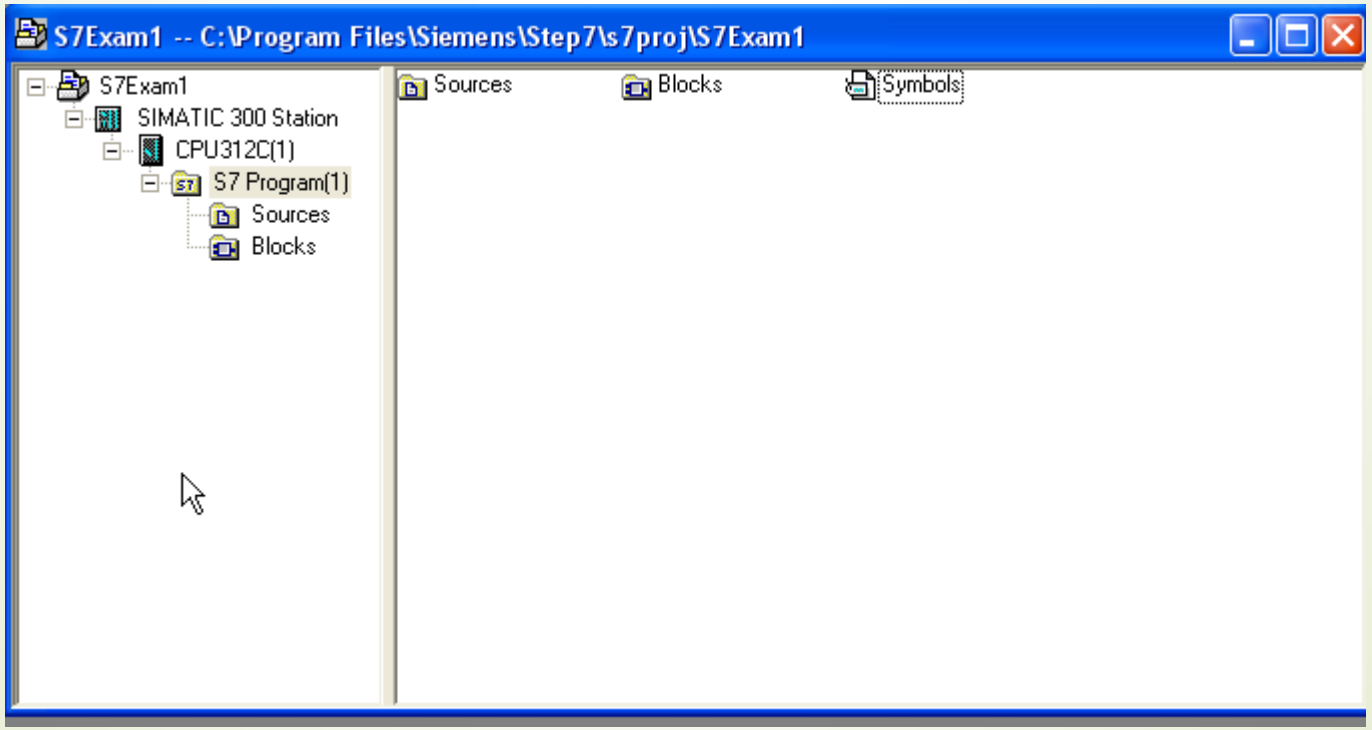
در قسمت Name نام پروژه را نوشته و با زدن گزینه Ok ، پروسه را ادامه می دهیم. اکنون در حالی که پنجره پروژه فعال است ، فرمان Station < insert را اجرا می کنیم و از زیر منو ها ی موجود در Station ، PLC مورد نظر را انتخاب می کنیم، که در اینجا PG/PC را انتخاب می کنیم ، به این ترتیب شی Station ایجاد می شود، بار دیگر پروژه را فعال کرده و این بار فرمان Program < Insert را اجرا می کنیم. بنابراین انتخاب ، شی S7 Program یا M7 Program اجرا می شود، که با استفاده از آن می توان نرم افزار پروژه را طراحی کرد. حال برای اضافه کردن جزییات نیز می توان از Insert استفاده کرد. به عنوان مثال برای ایجاد بلوک های S7 ، ابتدا S7 Program را علامت دار کرده و سپس گزینه S7 < Insert Block و از زیر منوی آن بلوک مورد نیاز را انتخاب می کنیم.

جدول نماد ها ، جدول معرفی متغییر های بلوک :

همانطور که گفته شد S7 اشیاء پروژه را در ییک ساختار ترتیبی در پنجره پروژه

نحوه کار با step 7

نمایش می دهد. این پنجره دارای دو بخش می باشد. کادر سمت چپ در شکل (۱۷-۱۱) ساختار پروژه را نمایش می دهد و با انتخاب و هر شی سمت چپ ، احزای آن در صفحه سمت راست نمایش داده می شود.



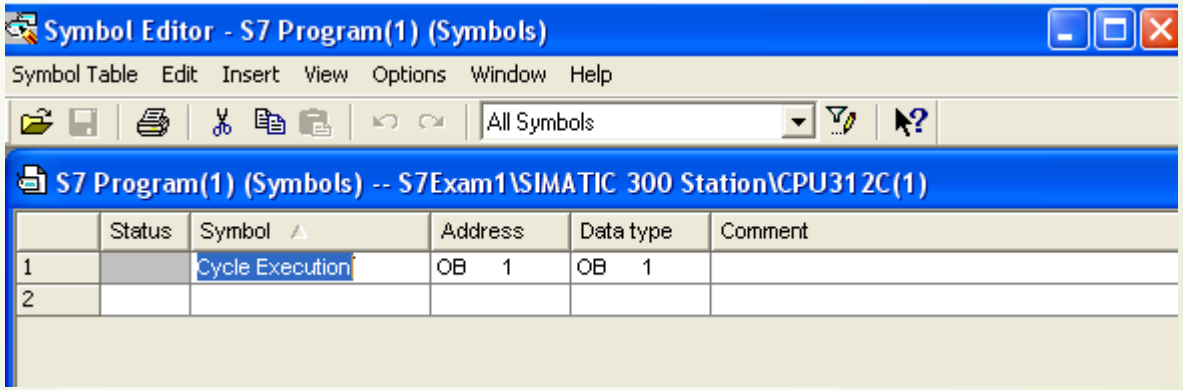
شکل (۱۷-۱۱)

نحوه کار با step 7

به عنوان مثال روی شی S7 Program1 کلیک کنید. اجرای آن به ترتیب Symbls ، Blocks ، Sources در سمت راست نشان داده می شود. همان طور که می دانید در برنامه نویسی S7 می توان از آدرس های نمادین به جای آدرس های مطلق استفاده کرد. آدرس های نمادین به دو دسته مشترک و محلی تقسیم بندی می شوند، آدرس های نمادین مشترک در جدول نماد ها تعیین می شوند و در هر جای برنامه کاربر ، توسط همه بلوک ها قابل استفاده می باشند و در همه بلوک ها یک مقصد مشترک را آدرس دهی می کنند و به همین دلیل منحصر به فرد می باشند.آدرسهای نمادین محلی در جدول متغیرهای بلوک تعیین می شوند و تنها برای همان بلوک شناخته شده هستند. به همین دلیل ممکن است یک نماد در بلوک های مختلف معنای مختلفی داشته باشد، آدرس های نمادین مشترک در داخل گیومه نشان داده می شوند و قبل از آدرسهای نمادین محلی علامت نامبر ساین قرار می گیرد. البته نیازی به قرار دادن علامت ها توسط خود کاربر نیست.

نحوه کار با step 7

زیرا هنگام استفاده از این آدرس ها در برنامه نویسی ، کنترلر ساختار دستور ، این کار را انجام می دهد. پس از ایجاد شی S7 یا M7 ، شی Symbols که همان جدول نمادها است به طور اتوماتیک ایجاد می شود. برای هر ماژول قابل برنامه نویسی ، تنها یک جدول نمادها اختصاص داده می شود. و آن جدول فقط برای آن ماژول قابل دسترس است . بنابراین اگر نیاز به استفاده یک جدول برای چندین Cpu باشد، می بایست آن جدول نمادها، شی Symbols را در پنجره پروژه دوباره کلیک کنیم . بدین ترتیب پنجره جدول به شکل (۱۷-۱۲) باز می شود. این نمادها می توانند به فضاهای آدرس دهی اختصاص می یابند.



شکل (۱۷-۱۲)

نحوه کار با step 7

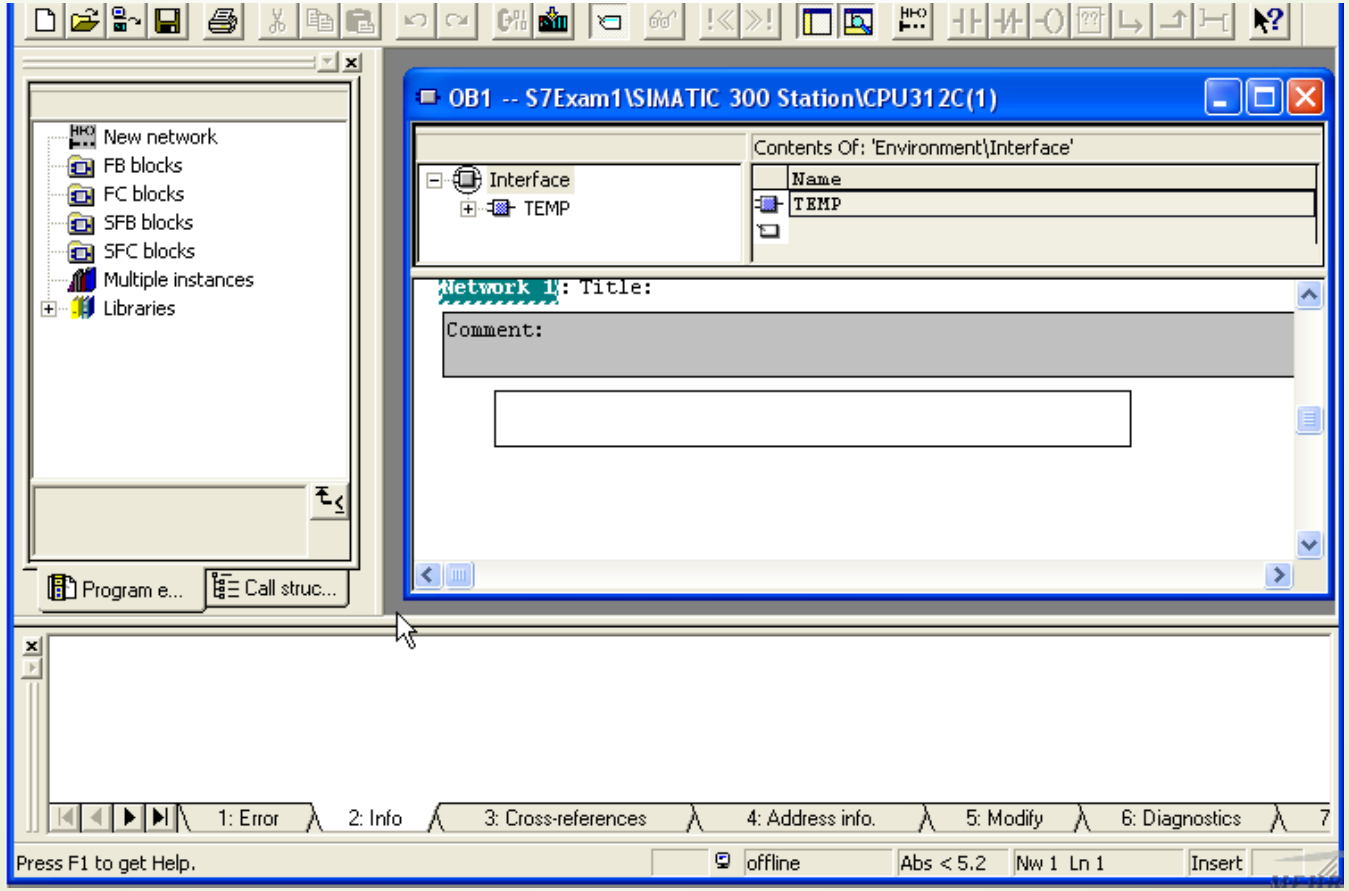
نمادهای قسمت های مختلف بلوک داده را نباید در جدول نمادها تعریف کرد و این آدرس ها در جدول معرفی متغیرهای آن بلوک تعیین می شوند. در ستون اول هر ردیف از جدول نمادها که همان ستون Symbol است نمادهایی که نباید بیشتر از ۲۴ کاراکتر باشد تایپ می کنیم و در ستون دوم آن آدرس مطلق مربوطه را وارد می کنیم و نوع داده به طور اتوماتیک به جدول اضافه می شود . در ستون Comment می توان در حداکثر ۸۰ کاراکتر در مورد نماد توضیح داد. این توضیحات در هنگام نوشتن برنامه به زبان STL جلوی دستورات آورده می شوند.

برنامه نویسی به زبان های FBD ، STL و LAD :

بلوک های OB ، FC و FB قابل برنامه نویسی به زبان های FBD ، STL و LAD توسط کاربر می باشند ، پس از باز کردن یکی از بلوک ها، پنجره برنامه نویسی باز می شود. برای مثال بلوک OB1 را باز می کنیم و وارد محیطی به شکل (۱۷-۱۳) می شویم. هر پنجره برنامه نویسی ارز سه بخش تشکیل شده است.

نحوه کار با step 7

در قسمت فوقانی هر پنجره ، جدول معرفی متغیرهای آن بلوک قرار دارد، نیمه پایینی صفحه برای نوشتن برنامه بکار می رود و آن پنجره را فضای برنامه نویسی گویند.



شکل (۱۷-۱۳)

نحوه کار با step 7

بخش سوم پنجره برنامه نویسی ، پنجره المان های برنامه است که در یکی از طرفین قرار دارد. در صورتی که پنجره المان های برنامه بروی صفحه نمایش دیده نشود . می توان با اجرای فرمان Program elements < Insert و یا انتخاب آیکن overview on/off پنجره المان ها باز خواهد شد. با استفاده از این پنجره می توان به المان های مورد نیاز برای برنامه نویسی به زبان LAD و FBD ، مانند دستورات منطقی ، تایمرها ، شمارنده ها ، مقایسه کننده ها ، مبدل های فرمت اعداد و پرش ها دسترسی پیدا کرد.به همین دلیل در برنامه نویسی به این دو زبان به مراتب از آن استفاده می شود. اغلب به دلیل پیچیدگی یک پروژه اتوماسیون ، برنامه های آن در چندین بخش نوشته می شوند. که هر کدام از آن ها هدف خاصی را دنبال می کنند. می توان هر بخش را در یک Network نوشت. برای ایجاد یک Network جدید فرمان Network < Insert را اجرا می کنیم و یا روی آیکن New Network در نوار ابزار کلیک می کنیم . اجرای فرمان ها ی View < LAD یا FBD یا STL در هر پنجره برنامه نویسی ، زبان برنامه نویسی را به ترتیب به

نحوه کار با step 7

STL ، LAD و FBD تغییر می دهد. اگر از منو View بروی گزینه Display with رفته و از آن ، زبانهٔ Symbol Information را علامت دار کنیم . توضیحات نوشته شده در ستون Comment جدول نمادها جلوی خطوط دستورات برنامه آورده می شود. هر Network را می توان در قسمت Title آن عنوان بندی کرد و در قسمت توضیحات مورد نیاز برای آن Network را نوشت. برای آشنایی بیشتر با برنامه نویسی با هر یک از سه زبان گفته شده در محیط S7 یک مثال برای هر سه حالت بیان کنیم.

فرض کنید می خواهیم راه اندازی یک موتور را برنامه ریزی کنیم که با فشردن کلید فشاری Start key در صورتی که سویتچ Emergency stop روشن نباشد ، موتور راه اندازی می شود. بنابراین می توان از یک کلید NO با نماد Start key و یک کنتاکت NC با نماد Emergency stop و یک Coil خروجی با نماد Motor on و یک کلید NO به عنوان لتچ ورودی با خروجی Motor on برای

نحوه کار با step 7

نوشتن برنامه استفاده کرد. پس از ایجاد پروژه، OB1 را باز کرده و به منظور ایجاد جدول نمادها فرمان Symbols table < options را از نوار منو اجرا می کنیم و جدول نمادها ظاهر می شود و به کمک آن می توان نمادهای مورد نیاز را به ورودی و خروجی اختصاص داد. البته توجه شود که در این مثال به دلیل محدود بودن ورودی و خروجی ها ، طراحی نرم افزار پروژه ، قبل از سخت افزار آن مشکلی به وجود نمی آورد. ولی لازم است برا یورودی ها و خروجی ها آدرس های مطلق مناسبی را به دلخواه اختصاص دهیم . پس از کامل نمودن جدول آن را ذخیره می کنیم . این جدول به شکل (۱۷-۱۴) پر می شود.

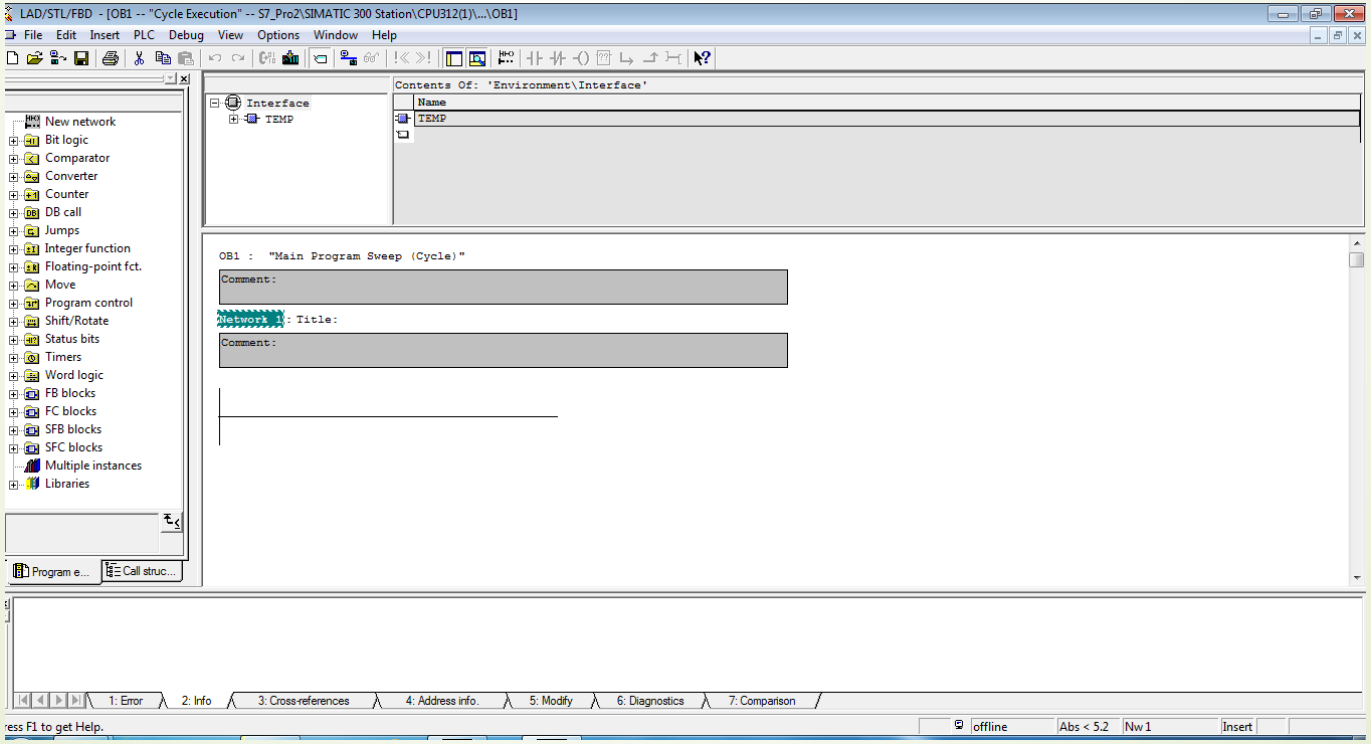
S7 Program(1) (Symbols) -- S7Exam1\SIMATIC 300 Station\CPU312C(1)

	Status	Symbol	Address	Data type	Comment
1		Start_Key	I 0.0	BOOL	Push-button for start motor
2		Emergency Stop	I 0.1	BOOL	Switch for stop the motor
3		Motor On	I 0.2	BOOL	Activates

شکل (۱۷-۱۴)

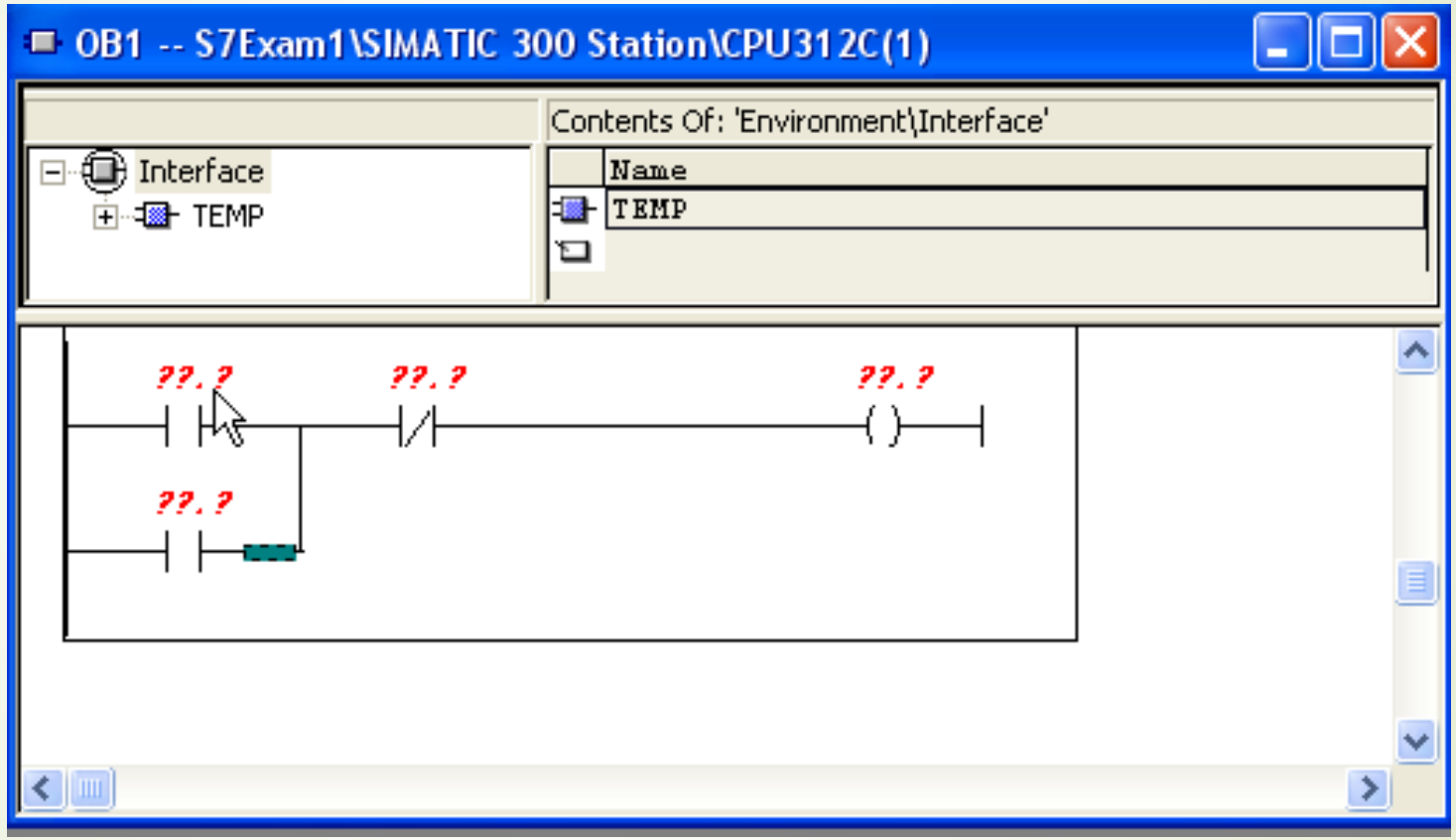
نحوه کار با step 7

در ابتدا یک برنامه به زبان LAD می نویسیم . در ابتدا پروژه را ایجاد کرده و زبان برنامه نویسی آن را روی LAD تنظیم می کنیم و صفحه پروژه به شکل (۱۷-۱۵) تبدیل می شود.



شکل (۱۷-۱۵)

نحوه کار با step 7



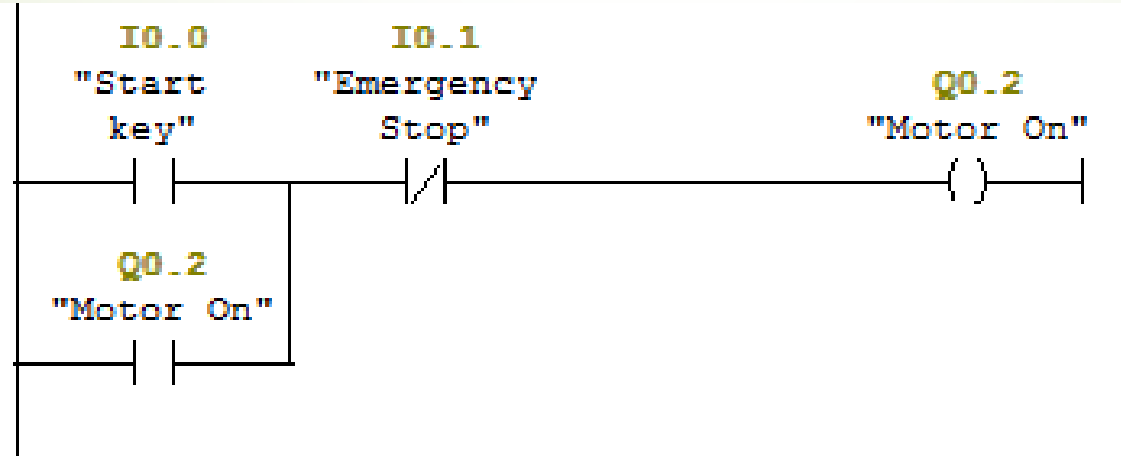
شکل (۱۷-۱۶)

نحوه کار با step 7

همانطور که در شکل (۱۷-۱۵) دیده می شود ، طبق توضیحات قبلی المان های اصلی برنامه نویسی در جدول سمت راست آن آمده است . برنامه نویسی به زبان LAD مانند نقشه کشی مدارات کنترل و فرمان الکتریکی است و دستورات برنامه از چپ به راست و از بالا به پایین اجرا می شوند . خط عمودی برنامه به عنوان یک منبع تغذیه مجازی است و المان های برنامه روی خط ورودی برنامه قرار می گیرند، با اجرای برنامه جریان انرژی از چپ به راست اعمال می شود. برای نوشتن برنامه لازم است خط ورودی برنامه فعال باشد. برای این منظور روی این خط کلیک می کنیم بدین ترتیب المان های برنامه نویسی موجود در نوارافزار فعال می شود. برای قرار دادن یک کنتاکت NO روی آیکن این المان در نوارافزار کلیک می کنیم و یا از پوشه Bit logic المان مورد نظر را با درک کردن بروی خط ورودی برنامه قرار می دهیم.با استفاده از این روش یک کنتاکت NO و یک Coil خروجی را نیز روی خط عمودی برنامه کلیک می کنیم و سپس یک شاخه قرار می دهیم. که شکل کلی مدار به شکل (۱۷-۱۶) می شود.

نحوه کار با step 7

حال روی علامت سوال هر متغیر کلیک راست کرده و گزینه **Insert symbol** را کلیک می کنیم. برای آدرس دهی برای متغیرهایی که از قبل تعریف کرده ایم با کلیک روی آن ها آدرس مورد نظر را فرآخوانی می کنیم. اگر از منوی **View** > **Display with** < **Symbol Representation** فعال باشد، آدرس نمادین را برای آن المان قرار می دهد و اگر غیر فعال باشد ، از آدرس دهی مطلق برای نمایش آدرس المان مورد نظر استفاده می کند. شکل برنامه با فعال بودن گزینه مذکور به شکل (۱۷-۱۷) نشان داده می شود.



شکل (۱۷-۱۷)

نحوه کار با step 7

حال برنامه را به زبان STL و FBD بازنویسی می کنیم.

جهت نوشتن به زبان STL ابتدا زبان برنامه نویسی را روی STL تنظیم می کنیم و سپس کد برنامه را به شکل (۱۷-۱۸)می نویسیم.

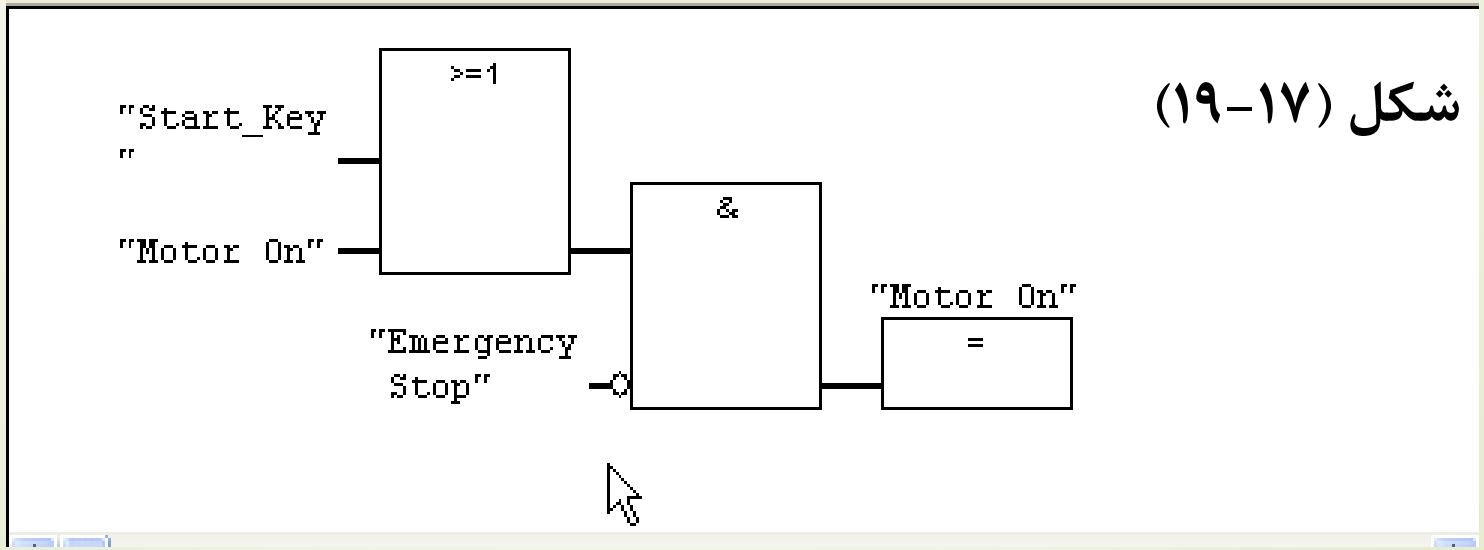
```
A(
O       "Start_Key"
O       "Motor On"
)
AN      "Emergency Stop"
=       "Motor On"
```

شکل (۱۷-۱۸)

در این کد O به معنی OR کردن ،A به منای AND ، = به معنای انتصاب و AN به معنای نقیض AND می باشد . حال اگر بخواهیم برنامه فوق را به زبان FBD بنویسیم ، زبان برنامه نویسی روی FBD تنظیم می کنیم ، برای OR کردن دو متغیر Start key و Motor On از بلوک استفاده می کنیم.حال یک بلوک AND

نحوه کار با step 7

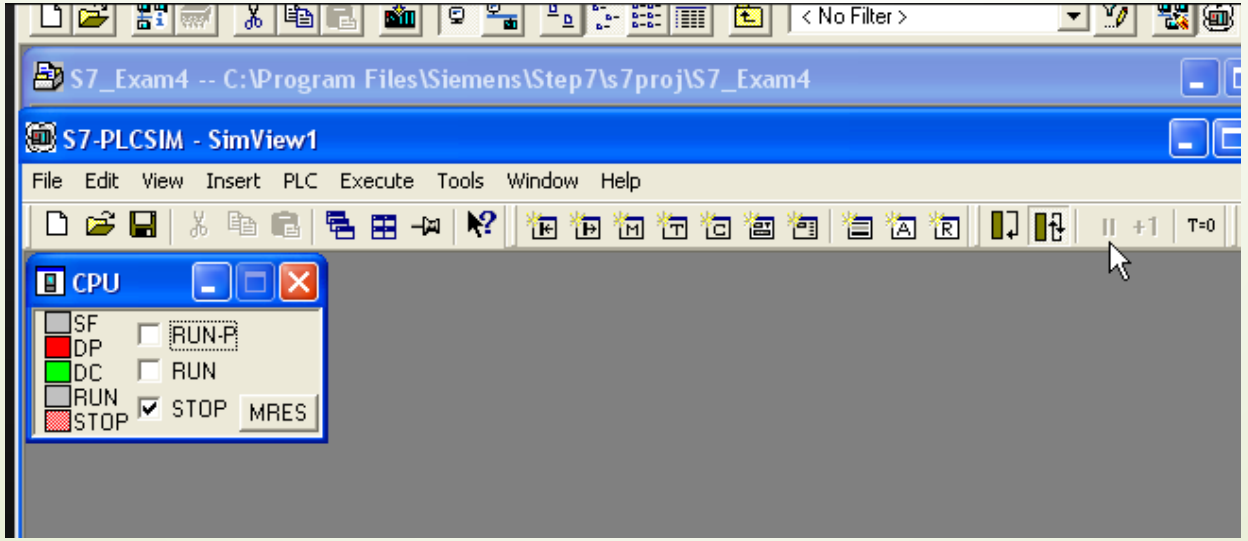
قرار داده و حاصل را OR را با متغیر Emergency Stop ، AND می کنیم و حال یک بلوک تخصیص مساوی در حالی ک بلوک AND فعال است ، اضافه می کنیم ، که در خروجی بلوک AND قرار می گیرد و آن را Motor On می نامیم. چون ورودی دوم AND باید نقیض Emergency Stop باشد، از این رو روی پورت ورودی این بلوک کلید کرده تا فعال شود. سپس روی علامت نقیض ساز کلیک کرده تا ورودی مورد نظر نقیضش شود. به طور کلی شکل برنامه کامل آن به صورت زیر خواهد بود.



شکل (۱۷-۱۹)

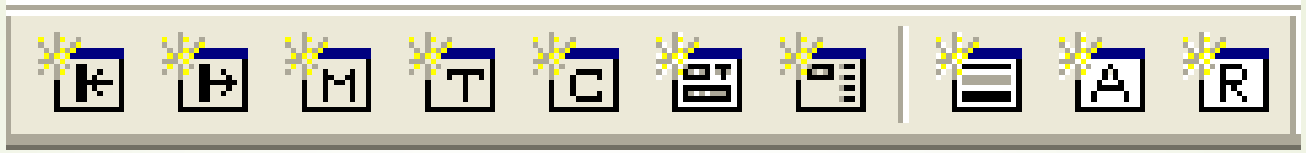
نحوه کار با step 7

برای شبیه ساز ی برنامه ابتدا برنامه نوشته شده در OB1 را اجرا می کنیم. سپس بر روی Simulation On/Off کلیک می کنیم، این آیکن مربوط به برنامه PLC_SIM می باشد، که حتما باید به همراه سیماتیک نصب شده باشد و در صورت غیرفعال بودن این آیکن حتما باید این برنامه به طور جداگانه نصب شود. به این ترتیب پنجره شبیه ساز ی به شکل (۱۷-۲۰) باز می شود ، در پنجره شبیه سازی ورودی ها ، خروجی ها ، تایمرها ، شمارنده ها و ... دسترسی دارید که در شکل (۱۷-۲۱) نشان داده شده است.



شکل (۱۷-۲۰)

نحوه کار با step 7



شکل (۱۷-۲۱)

همانطور که پنجره **Cpu** در شکل (۱۷-۲۰) نشان داده شده است می توان شبیه سازی را **Run** یا **Stop** کرده ، البته توجه به این نکته ضروری است که اگر شبیه سازی **Run** باشد نمی توان تغییرات و عملیات دانلود به برنامه اعمال کرد. به محیظی که برنامه نوشته اید ، دوباره بازگردید و برنامه را ذخیره کنید . سپس برروی **Download** کلیک کرده ، به این ترتیب برنامه برروی **PLC** مجازی بارگزاری می شود، اکنون در محیط شبیه سازی بروی گزینه **Run** کلیک کرده به این ترتیب شبیه سازی آغاز می شود. حال ماژول های ورودی و خروجی را در محیط شبیه سازی فراخوانی کنید. حال با توجه به نحوه برنامه نویسی و آدرس دهی متغیرهای آن اگر بیت صفر ام از بایت صفرام ورودی را فعال کنیم ، بیت صفرام از بایت صفر ام خروجی فعال می شود، حال اگر بیت صفرام از بایت صفرام ورودی را

نحوه کار با step 7

غیر فعال کنیم، باز خروجی روشن می ماند ، ولی اگر کلید Emergency Stop را فعال کنیم یعنی بیت اول از بایت ورودی ، خروجی خاموش می شود. با کلیک کردن برروی آیکن Monitor On/Off می توانید وضعیت کنتاکت های برنامه را در ورودی شماتیک برنامه مشاهده کنید.