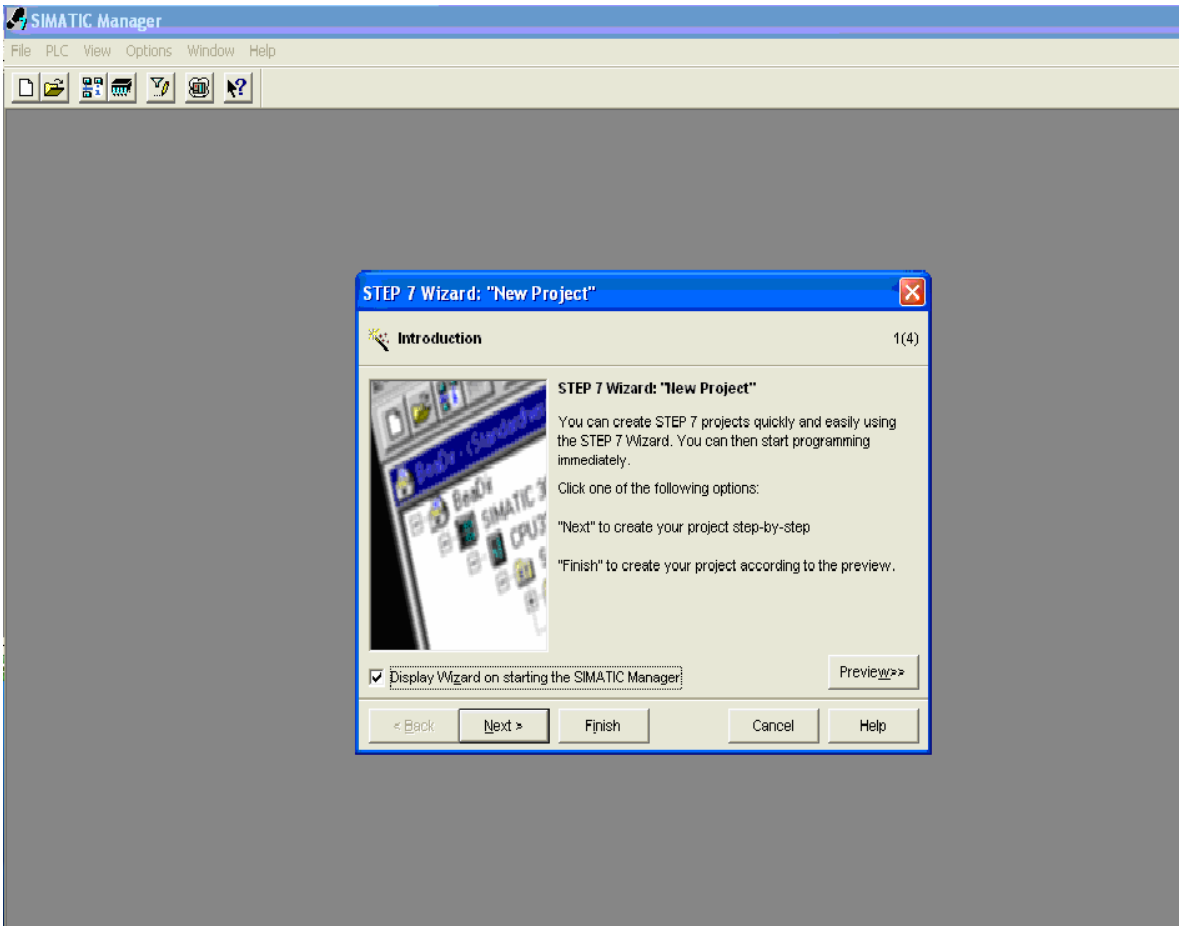
به نام خدا

۱. کار با (Simatic Manager) :

برای نصب و کار با نرم افزار STEP 7 Professional به یک وسیله برنامه ریزی یا کامپیوتر نیاز دارید . سیستم عامل های پیشنهادی Windows 2000 Professional و Windows XP Professional میباشند حدود 440MB فضا روی هارد دیسک برای نصب کامل این نرم افزار نیاز است.

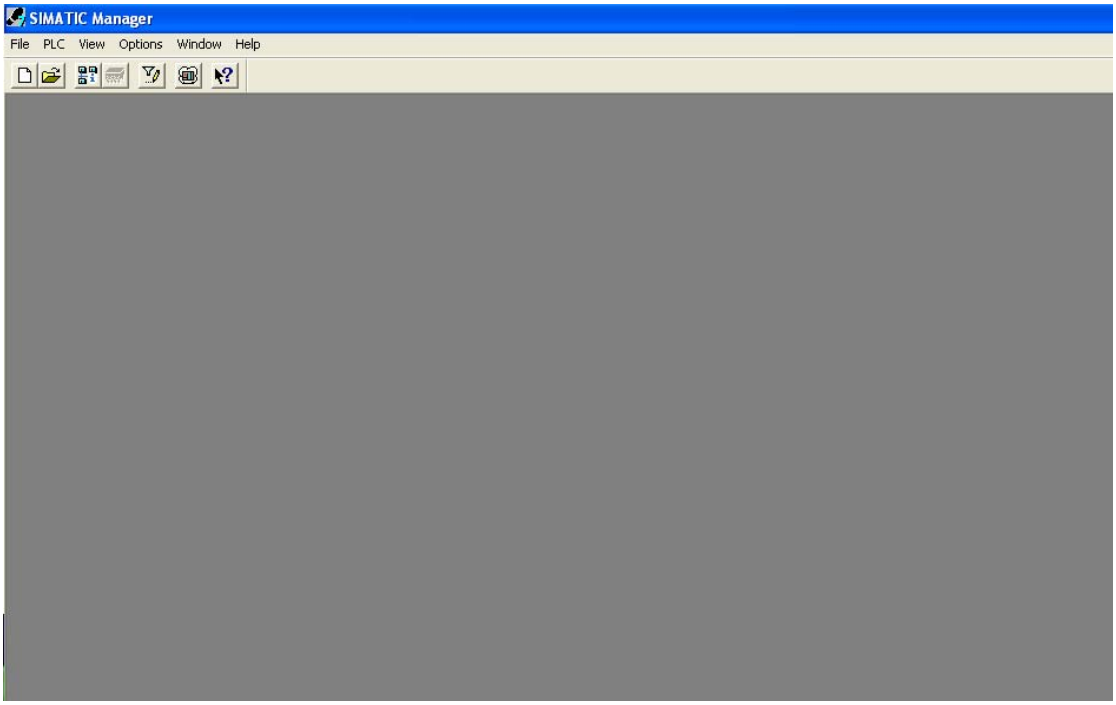
برای برقراری ارتباط با PLC توسط PC به آداپتور مخصوص نیاز است تا با اتصال به پورت RS232 کامپیوتر و پورتMPI روی PLCارتباط به شکل سخت افزاری نیز برقرار گردد.

پس از نصب نرم افزار ، آیکون  روی Desktop ایجاد میشود.با اجرای آن پنجره زیر ظاهر میگردد:

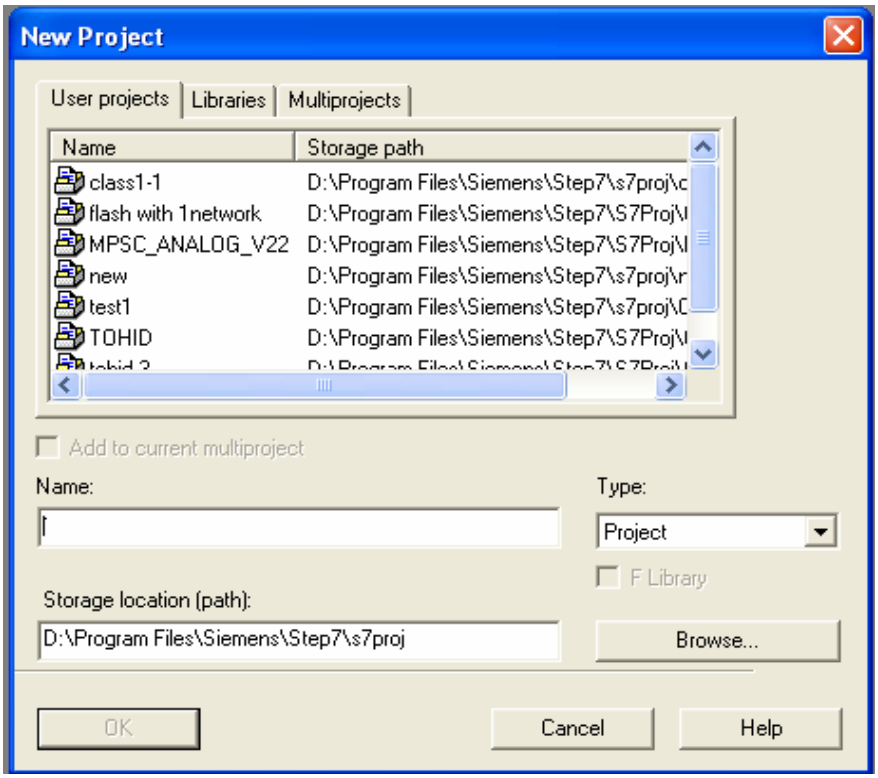


همانطور که مشاهده میشود پنجره Wizardبرای ایجاد پروژه جدید به صورت اتوماتیک باز میشود .ایجاد پروژه جدید از این طریق با محدودیت هایی روبروست.

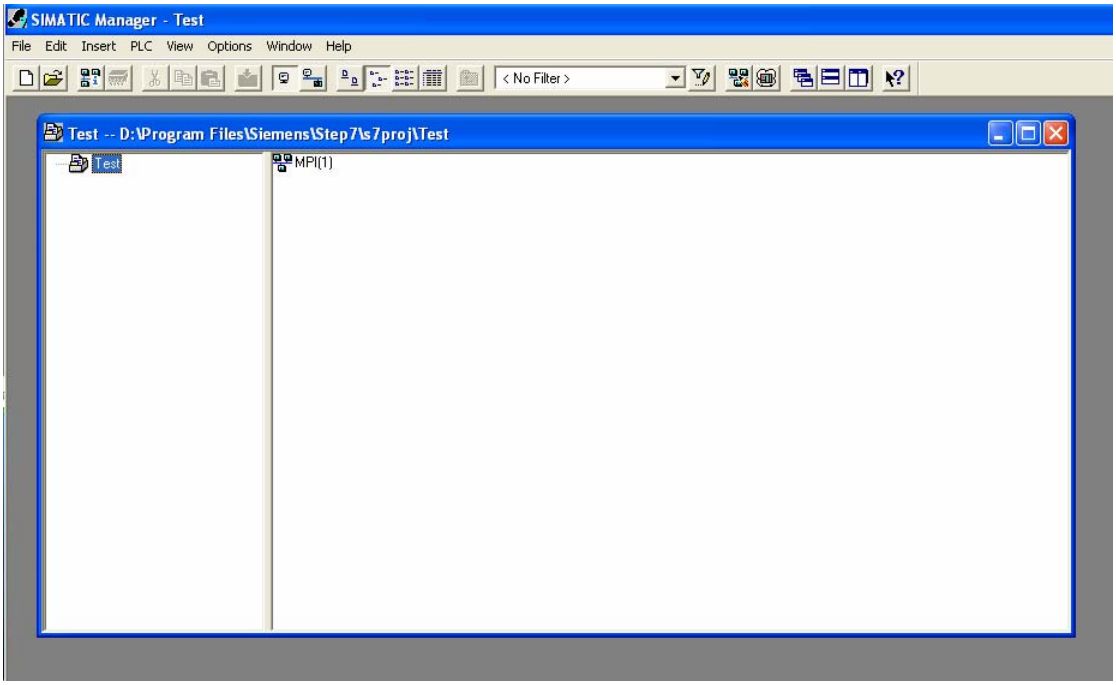
از این جهت پنجره مربوطه را بسته و خود به ایجاد یک پروژه به صورت کامل اقدام مینماییم.



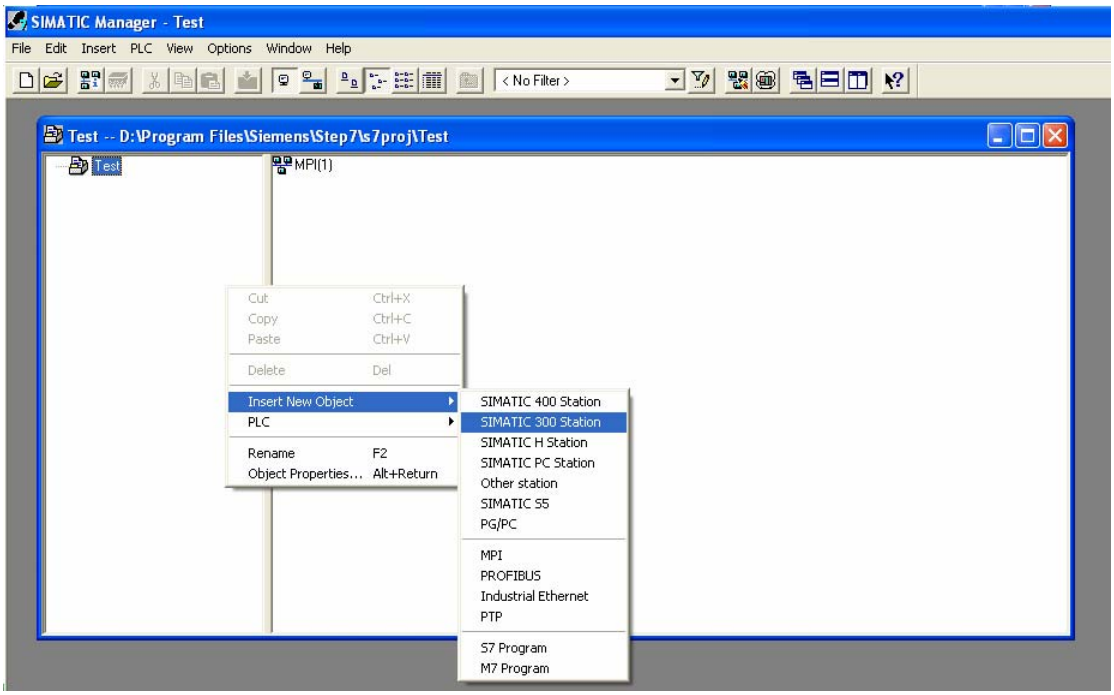
از مسیر File / new پروژه جدیدی را باز میکنیم .



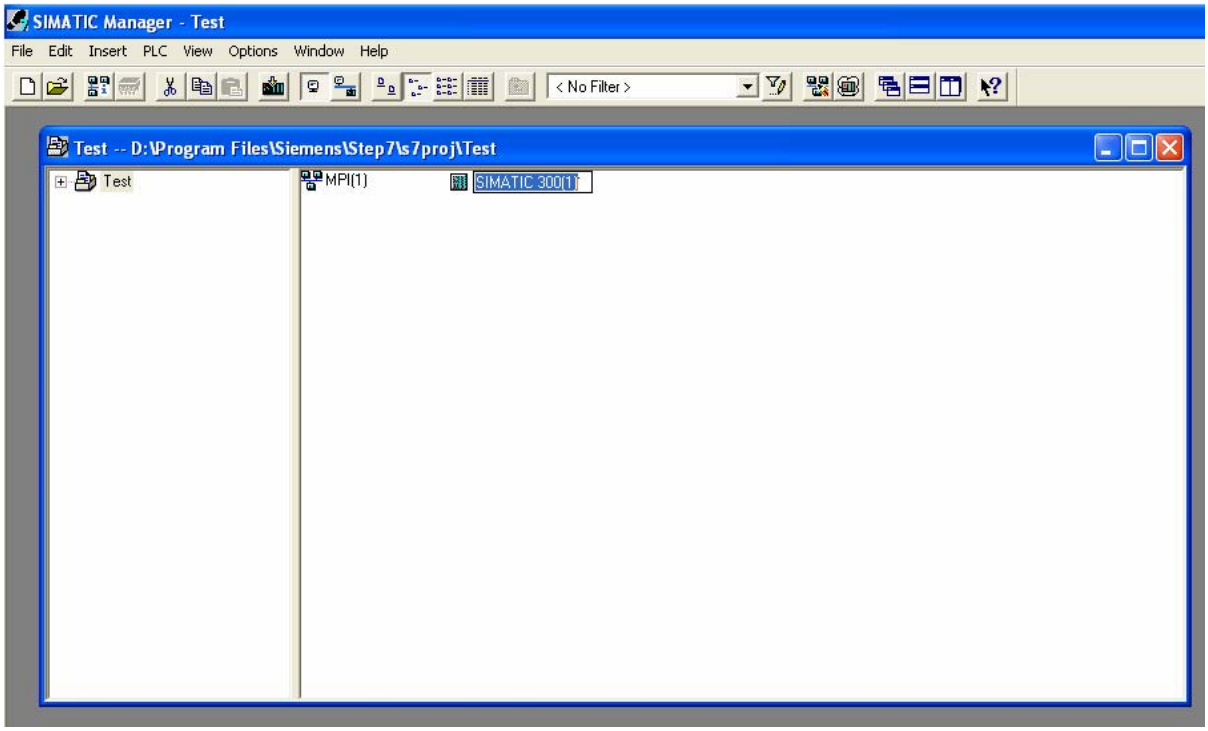
نام پروژه را در پنجره Name تایپ میکنیم . همچنین همانطور که مشاهده میشود نام و مسیر ذخیره پروژه های قبلی نیز دردسترس است .برای پروژه فعلی نیز میتوانید مسیری غیر از مسیر در نظر گرفته شده در پنجره Storage Location را انتخاب کنید.
به عنوان مثال نام پروژه را TEST گذاشته و ok میکنیم:



روی فضای خالی پنجره پروژه جدید کلیک راست کرده واز گزینه Insert new object ، مورد SIMATIC 300 Station را انتخاب میکنیم.



Station مربوطه در پنجره پروژه ایجاد میشود.



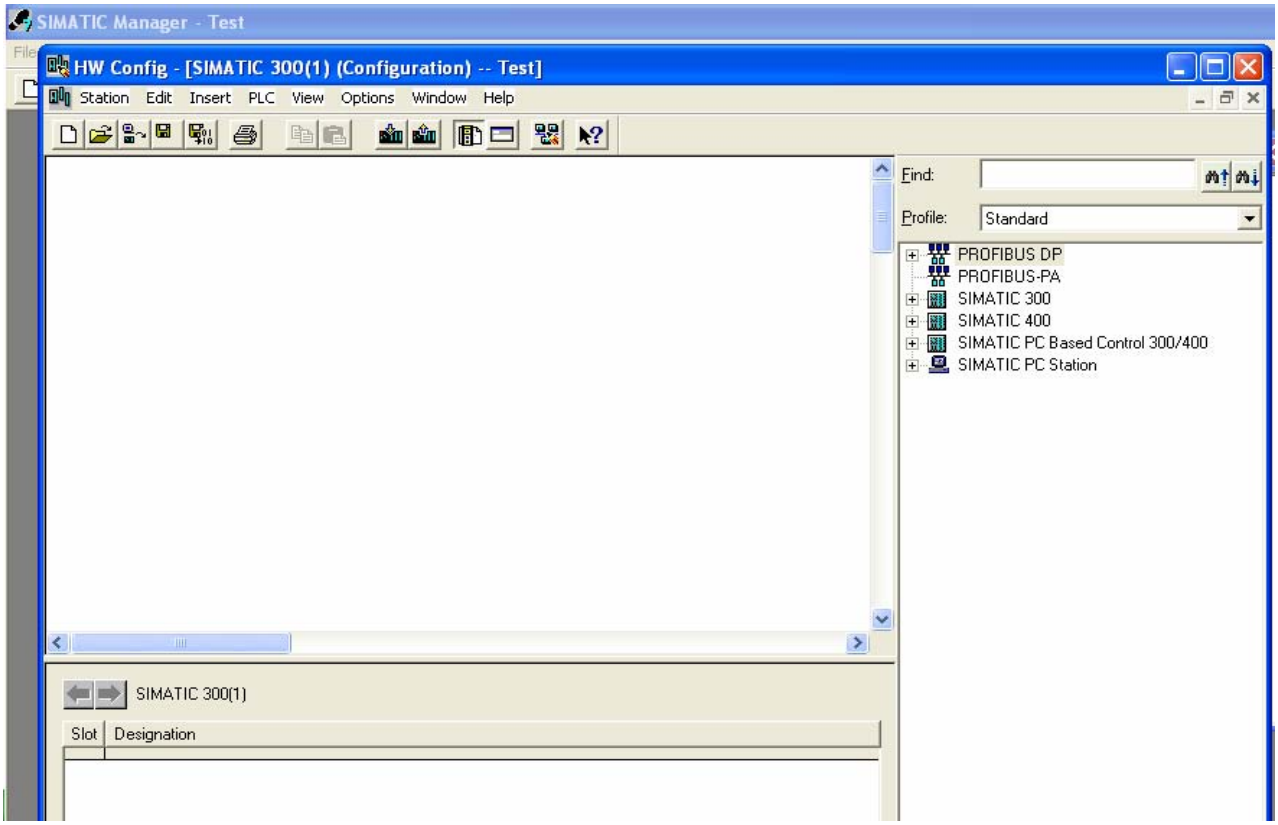
حال میبایست اجزای های سخت افزاری مورد استفاده در پروژه را به صورت کامل تعریف کنیم . بدین منظور روی Station مربوطه دو بار کلیک کرده تا آیکون Hardware ظاهر گردد.با باز کردن آیکون مذکور وارد محیط نرم افزار.HW config شده و درآنجا می توان تمامی سخت افزار مورد نیاز از جمله رک ، منبع تغذیه ،پروسسور ،انواع ماژولها و...را ازطریق کاتالوگ موجود در این نرم افزار تعیین کرد.

برای آوردن یا برداشتن کاتالوگ موجود در سمت چپ از آیکون 

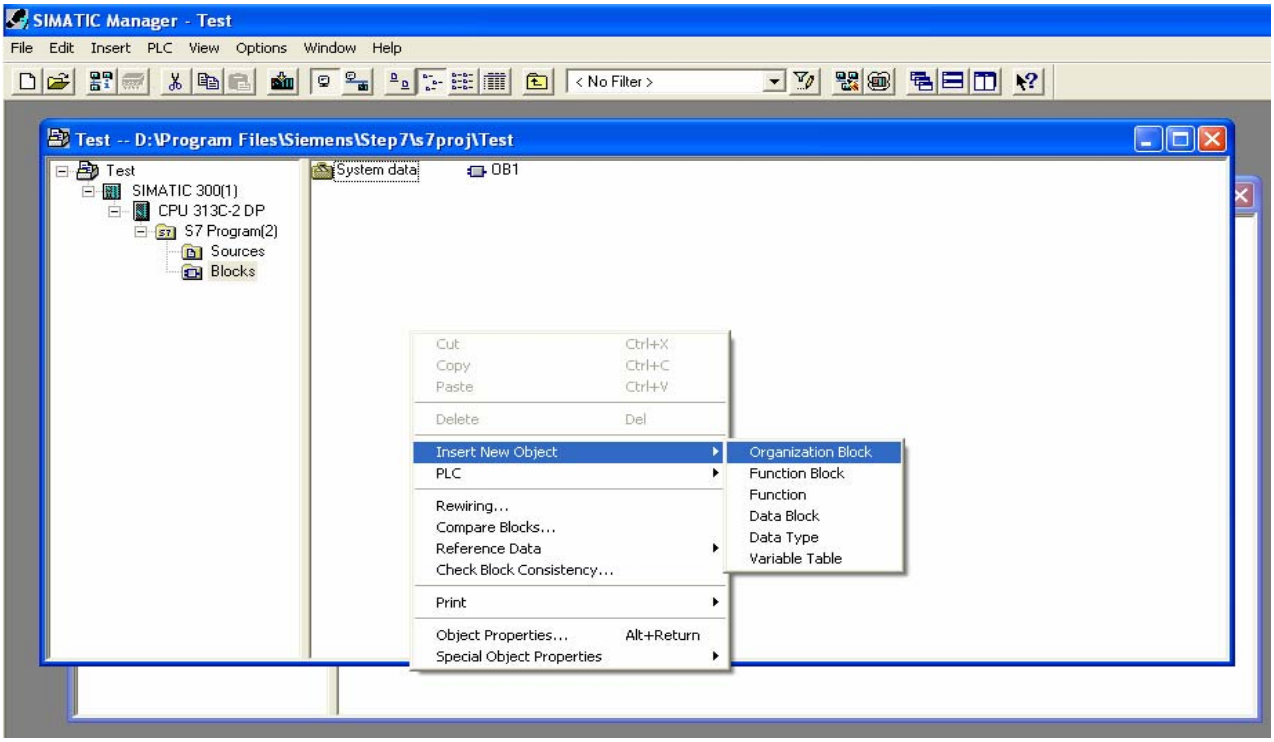
استفاده کرد.

در شکل صفحه بعد محیط این نرم افزار را میتوانید مشاهده نمایید.

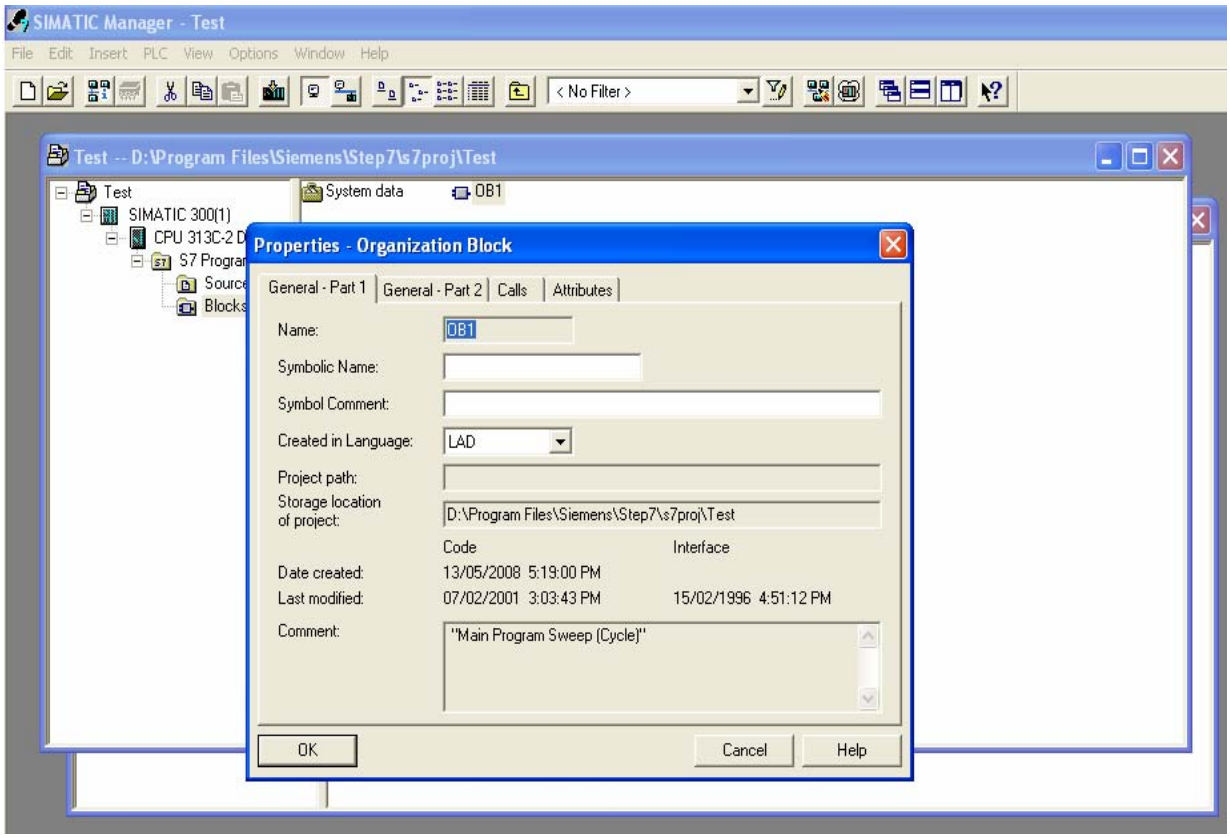
پس از تعیین سخت افزار آنرا در PLC دانلود نمایید. سپس از محیط .HW config خارج شده و وارد پنجره مربوط به پروژه در محیط Simatic manager شوید.



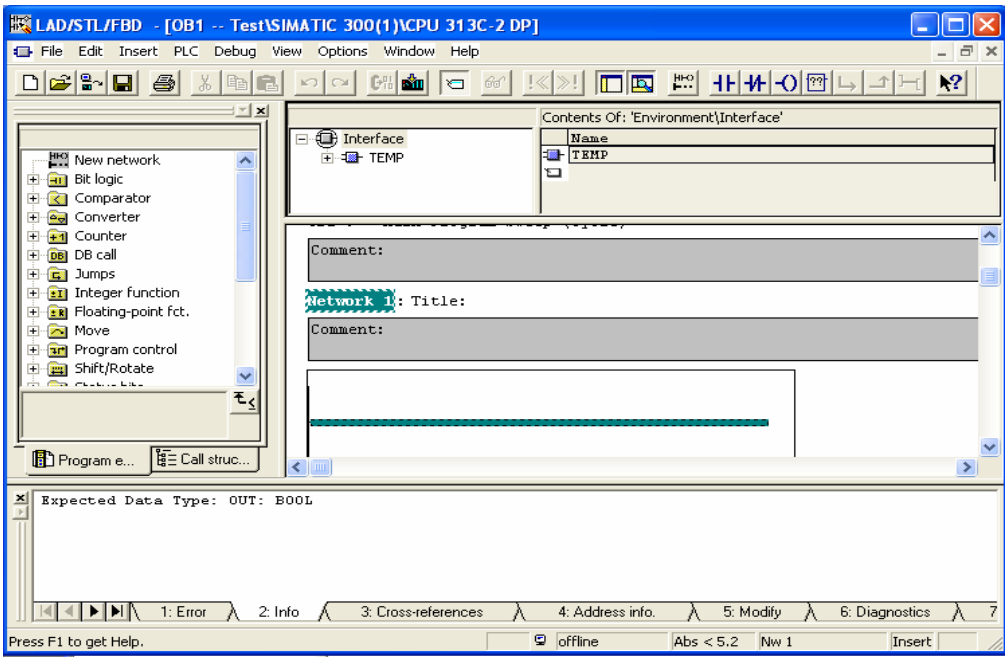
در کنار آیکون Hardware آیکون مدل پروسسور انتخابی را میتوان مشاهده کرد.با وارد شدن در آن درقسمت S7program ودرزیر شاخه block میتوانید محیط مربوط به فایل های پروژه رامشاهده نمایید که فعلا شامل OB1 و System data است.بعدا و در صورت نیاز به OB یا فانکشنهای دیگر میتوان با زدن دکمه راست ماوس و در گزینه Insert new objects بلاک مربوطه را انتخاب نمود.



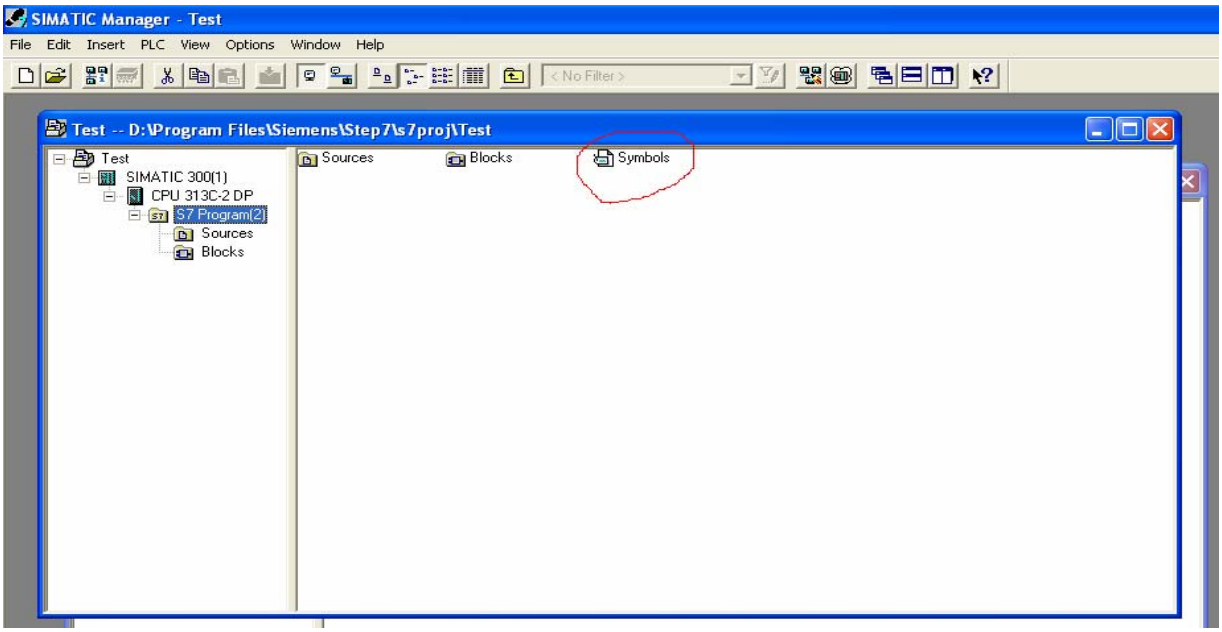
OB1 محیطی است که پروسسور هنگامی که در وضعیت اجرا (Run) قرار دارد آنرا به صورت سیکلی اسکن میکند و پردازش را روی دستورات آن اعمال میکند . بنابراین OB1 شامل برنامه اصلی ما خواهد بود.با کلیک روی آیکون OB1 پنجره ای مشابه پنجره زیر باز میشود که در آن میتوان یکی از سه زبان برنامه نویسی LADوSTLوFBD را انتخاب و همچنین نام سمبلیک برای برنامه گذاشت.(symbolic name)



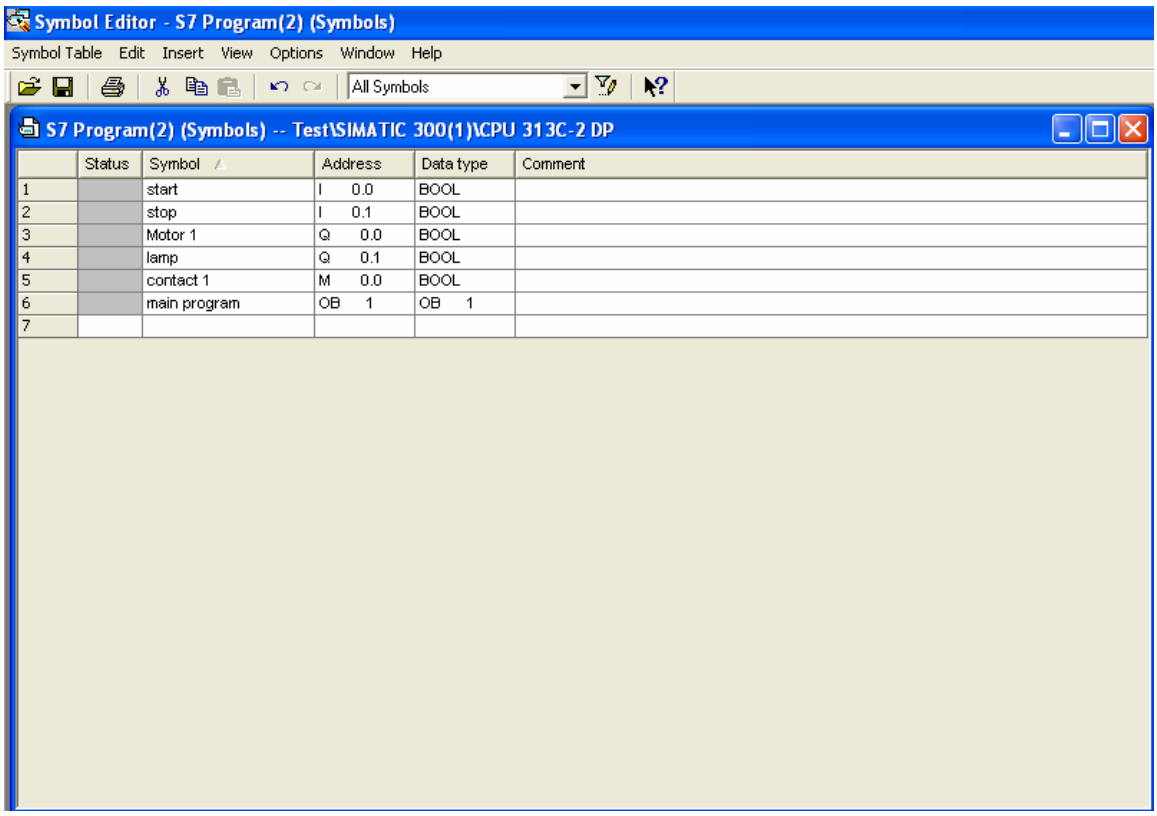
محیط برنامه نویسی به زبان انتخاب شده با نام OB1 باز شده و آماده استفاده است.



برای تعریف سمبل در پروژه میتوانید در صفحه اول پروژه از قسمت S7 Program روی آیکون symbols کلیک کرده تا محیط Symbol editorباز شود.



تمامی آدرسها میتوانند با یک نام سمبلیک(نمادین) در اینجا مشخص شوند.پس از بستن آن ،تمامی آدرسهای دارای سمبل را میتوانید با نام تعریف شده در محیط برنامه ببینید.

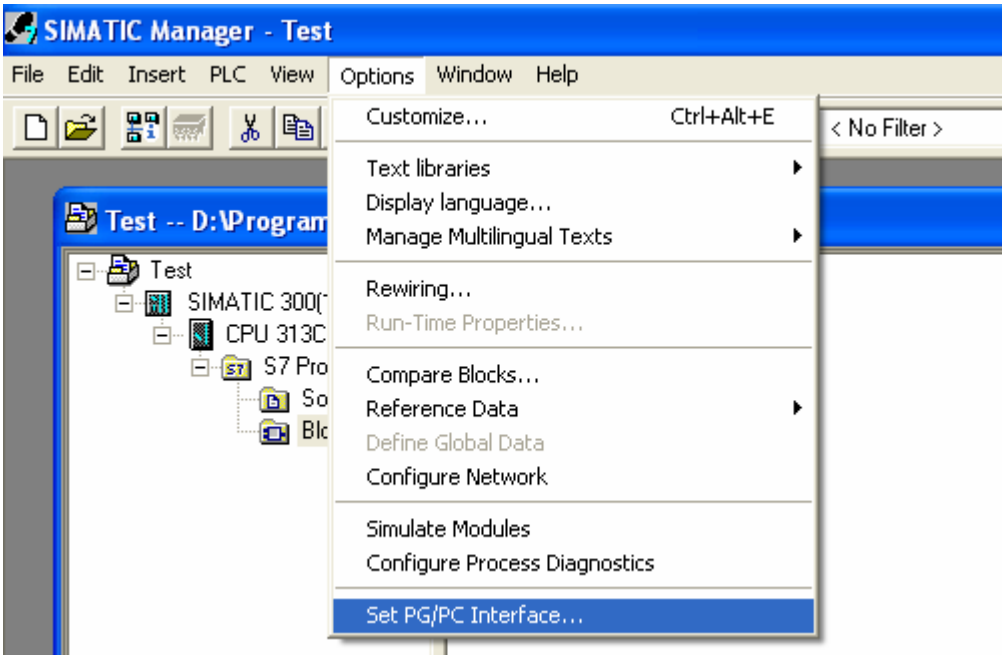


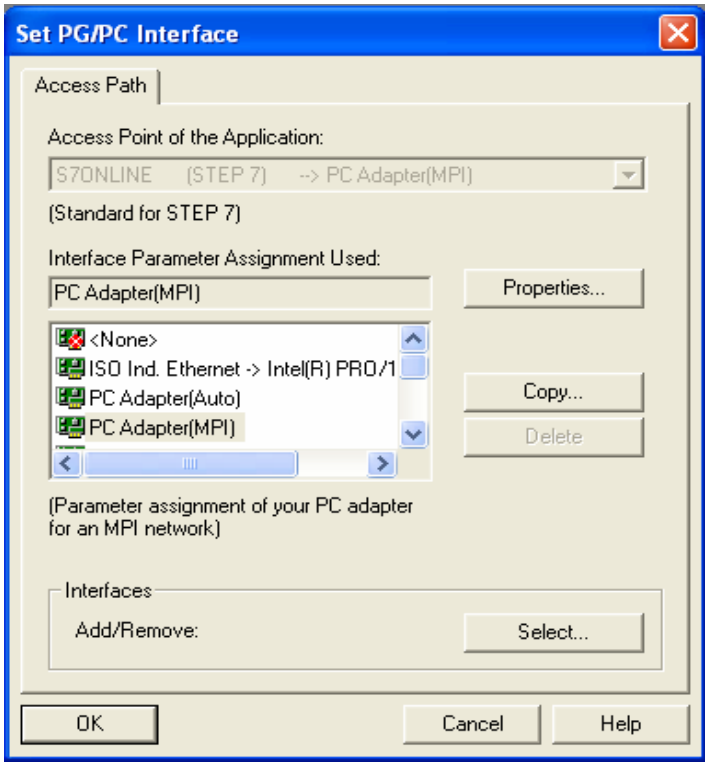
	Status	Symbol	Address	Data type	Comment
1		start	I 0.0	BOOL	
2		stop	I 0.1	BOOL	
3		Motor 1	Q 0.0	BOOL	
4		lamp	Q 0.1	BOOL	
5		contact 1	M 0.0	BOOL	
6		main program	OB 1	OB 1	
7					

پس از نوشتن برنامه آن را save نمایید سپس با استفاده از دکمه Download میتوان برنامه را در PLC دانلود کرد.توجه نمایید که اگر برنامه PLCSIM (برنامه سیمولاتور) باز باشد برنامه در آن دانلود میشود.

در صورتیکه بخواهیم برنامه را در PLC واقعی دانلود کنیم میبایست آداپتور مخصوص که بین خروجی سریال کامپیوتر و پورت MPI روی PLC ارتباط برقرار میکند را نصب و تنظیمات سخت افزاری را با نرم افزار Set PC/PG Interface انجام میدهیم .

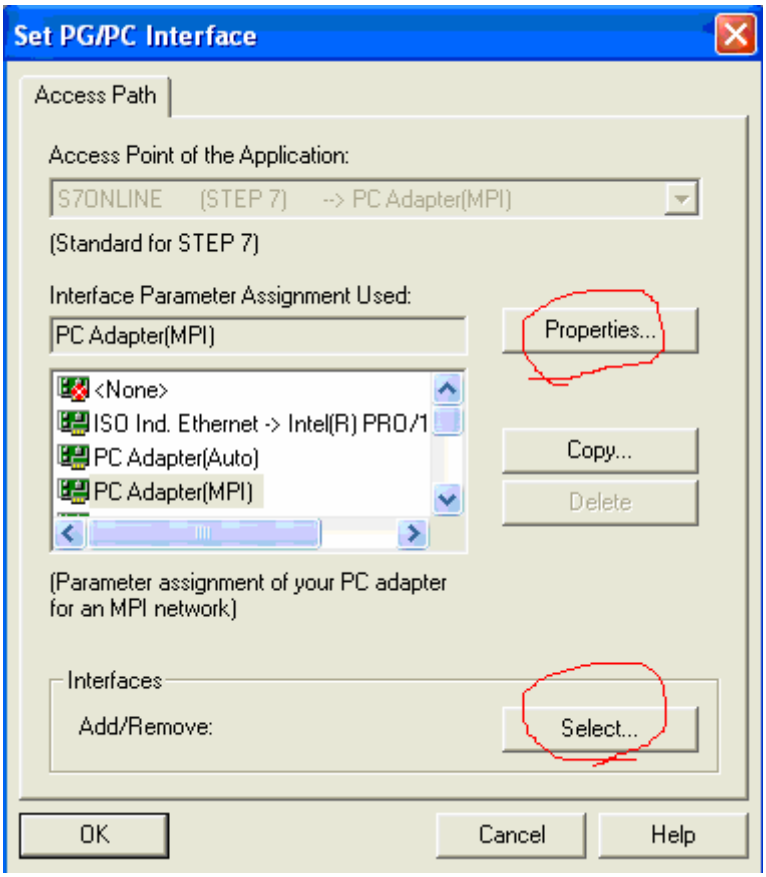
اکنون میتوانید برنامه را روی کنترلر دانلود نمایید.





در قسمت SET PC/PG باید گزینه PC Adaptor(MPI) را انتخاب کرده و ok نمایید.

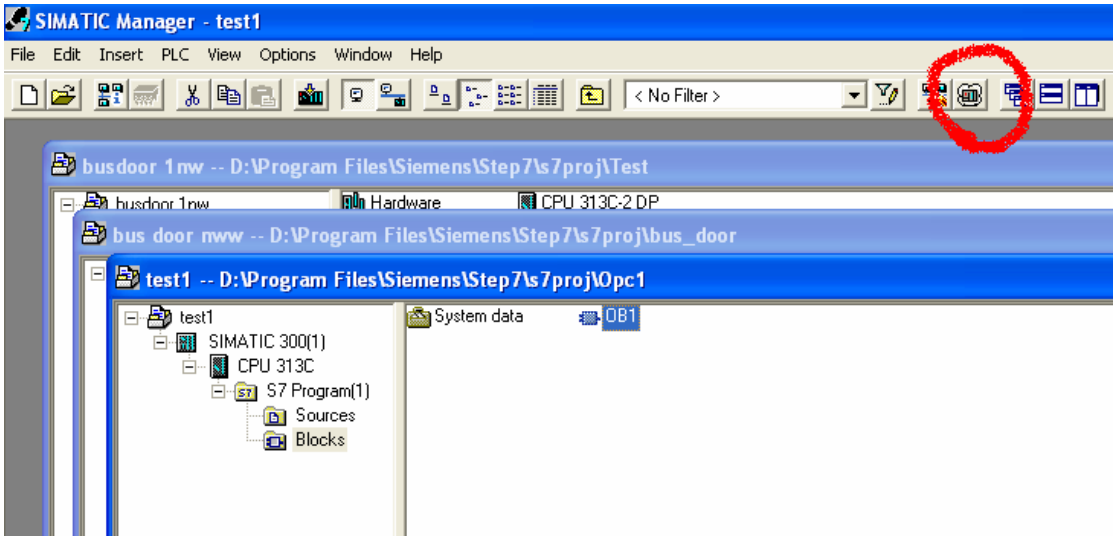
در صورتیکه آپشن نامبرده در پنجره فوق نبود روی دکمه select کلیک کرده ، از پنجره باز شده این آپشن را انتخاب و install کنید.



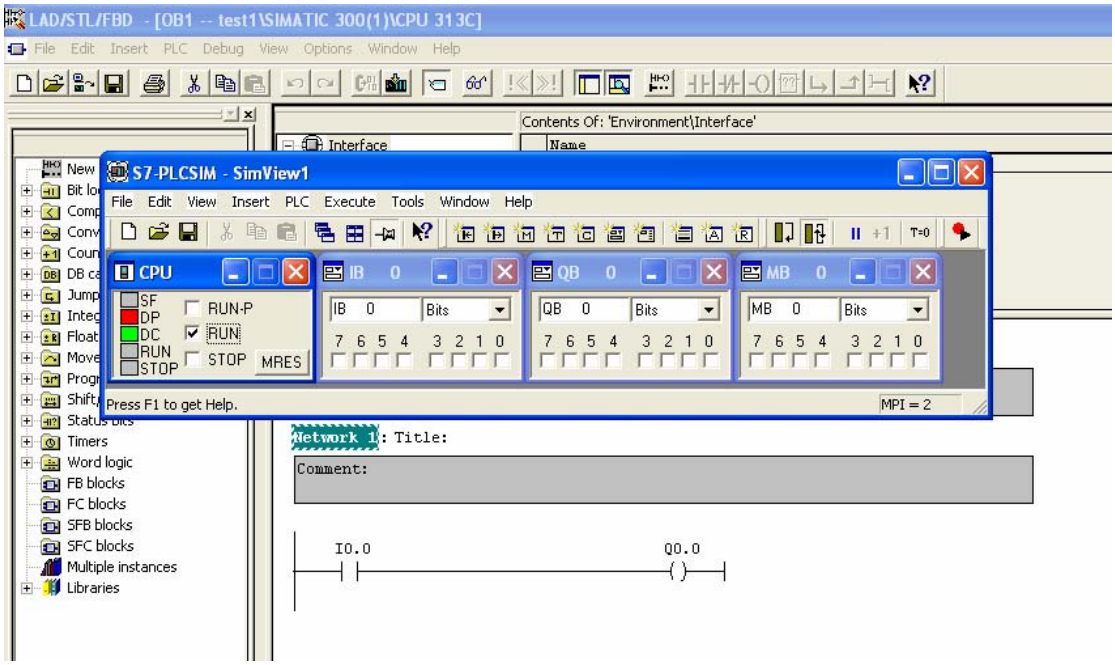
نکته دیگری که میبایست مورد توجه قرار گیرد اینست که پس انتخاب گزینه PC Adapter(MPI) و به قسمت properties رفته ودر بخش local connection ، پورت انتخابی درست انخاب شده باشد و هم چنین عدد Transmission rate با عدد انتخاب شده روی خود PC Adaptor (در کنار آن توسط dip سوییچ انتخاب میشود!) یکسان باشد.

نکته: در مورد PC Adapter های با پورت USBمیبایست حتما درایور آن را از طریق CDهمراه آن روی کامپیوتر نصب نمایید.

در صورت نیاز و قبل از دانلود کردن برنامه در PLC واقعی ،میتوان در محیط SIMATICآنرا شبیه سازی کرده و ورودی و خروجـی هـا را مـشاهده کرد. برای ورود به محیط سیمولیشن از نرم افزار PLCSIM در محیط SIMATIC استفاده میکنیم.

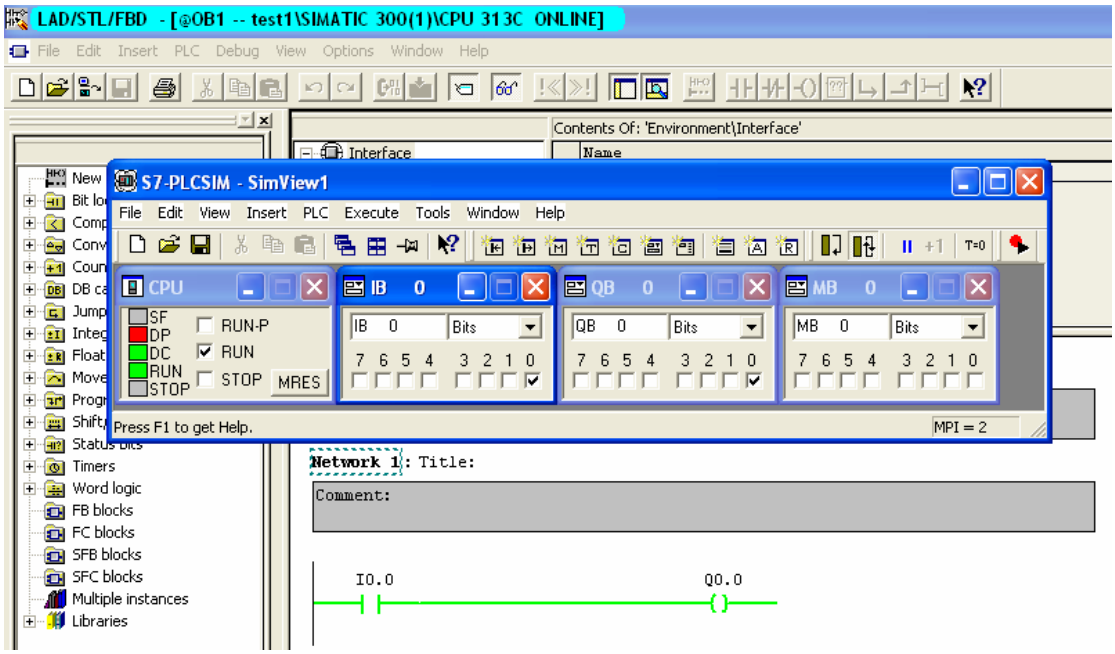


پس از کلیک کردن روی آیکون بالا ، وارد محیط سیمولیشن (PLC مجازی) میشویم.همانطور که مشاهده میشود مانند PLCواقعی دارای نمایشگر وضعیت ،دکمه های RUNوRESETو... است.



حال برنامه نوشته شده را توسط دکمه دانلود در این PLC ریخته و آنرا RUN میکنیم. میتوانید با تحریک ورودی های تعریف شده خروجی های مورد نظر را مطابق برنامه ریخته شده ملاحظه کنید.

استفاده از PLCSIM به برنامه نویس کمک میکند تا بدون نیاز به سخت افزار (کنترلر واقعی) مشکلات برنامه را رفع نماید.



برای مشاهده وضعیت اجزاء مختلف برنامه به صورت online ،کافیست روی آیتم "عینک" در قسمت بالایی برنامه کلیک نمایید.

در قسمت بعد، با دستورات پایه برنامه نویسی آشنا خواهیم شد.

۲. عملیات بیتی با (Bit Logic) :

دستوراتی که برای انجام عملیات بروی حافظه بیتی در زبان LADDER در محیط Step7 وجود دارد به قرار زیرند:

---| |--- Normally Open Contact

---| / |--- Normally Closed Contact

---|NOT|--- Invert Power Flow

---( ) Output Coil

---(#)--- Midline Output

---(S) Set Coil

---(R) Reset Coil

RS Reset-Set Flip Flop

SR Set-Reset Flip Flop

---(N)--- Negative RLO Edge Detection

---(P)--- Positive RLO Edge Detection

---(SAVE) Save RLO into BR Memory

NEG Address Negative Edge Detection

POS Address Positive Edge Detection

دستورات مربوط به کنتاکت های باز و بسته صرفا برای نمایش ورودی های سیستم اند که با چیدمان آنها در محیط برنامه نویسی LADDER میتوان عملیات منطقی مانند AND،OR و ... را ایجاد نمود. معادل این کار را میتوان با دستورات زیر در محیط برنامه نویسی STL انجام داد:

A AND

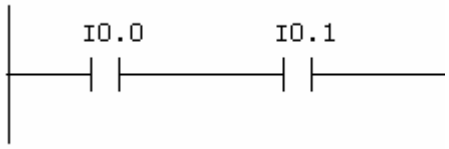
AN AND not

O OR

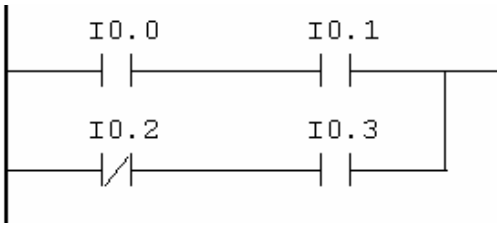
ON OR not

برای مثال:

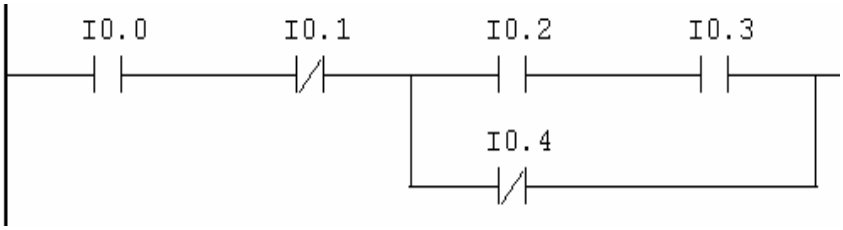
```
A       I       0.0
A       I       0.1
=       Q       0.0
```



نکته: دستور AND بر OR مقدم است.به مثالهای زیر دقت کنید:




```
A       I       0.0
A       I       0.1
O
AN      I       0.2
A       I       0.3
=       Q       0.0
```




```
A       I       0.0
AN      I       0.1
A(
A       I       0.2
A       I       0.3
ON      I       0.4
)
=       Q       0.0
```

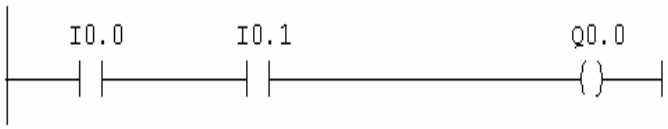
دستور **Invert Power Flow** :

همانطور که از نامش پیداست وظیفه NOT کردن سیگنال را به عهده دارد.

دستور **Output Coil** :

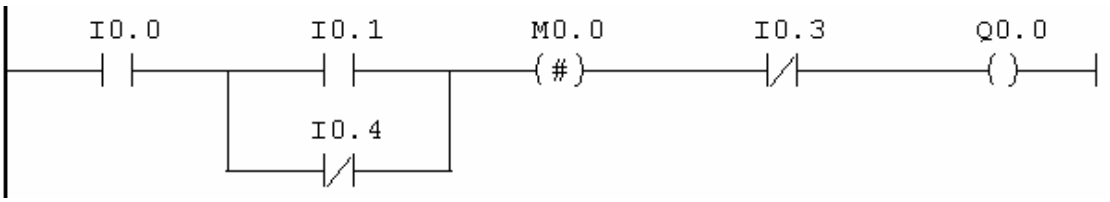
برای ریختن حاصل یک عملیات منطقی در یک آدرس(بیت) به صورت مستقیم از این دستور و نماد استفاده میشود:

```
A     I     0.0
A     I     0.1
=     Q     0.0
```



دستور **Midline Output** :

برای گرفتن خروجی میان یک Network از این دستور استفاده میشود، بدین ترتیب میتوان در برخی از موارد در استفاده ازnetwork اضافی جلوگیری کرد.مثال:

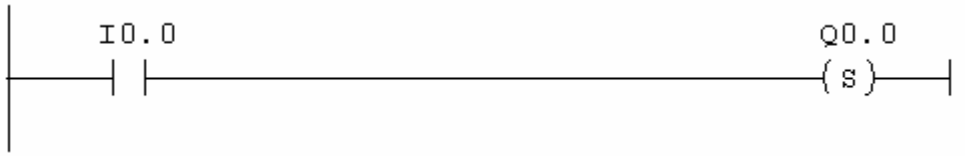



```
A     I     0.0
A(
O     I     0.1
ON    I     0.4
)
=     M     0.0
A     M     0.0
AN    I     0.3
=     Q     0.0
```

دستورات **Set & Reset Coil**:

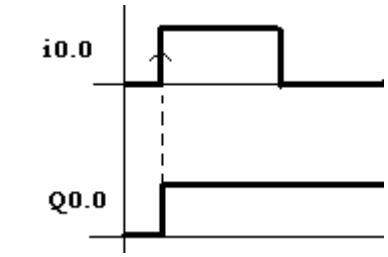
برای ست یا ری ست کردن یک بیت به کار میرود .در واقع با دیدن لبه بالارونده در network مربوطه مقدار یک بیت را به صورت دائم به یک وضعیت تغییرمیدهد. در صورتیکه دستور ست به کار برده باشید مقدار بیت را به یک و در صورت استفاده از ری ست به صفر تغییر میدهد.

مثال:




```
A       I       0.0
S       Q       0.0
```

همانطور که در شکل میبینید این دستور به صورت پیش فرض با لبه بالا رونده کار میکند.



Reset-Set & Set-Reset Flip Flop:

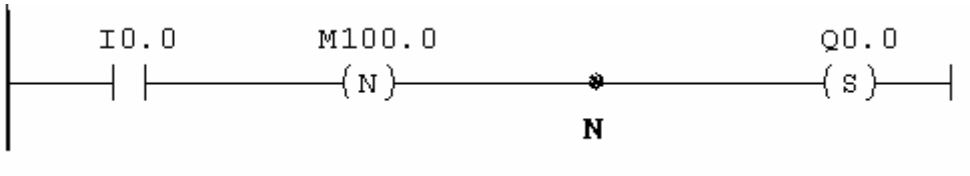
یک فلیپ فلاپ ساده است که در نوع RS ری ست بر ست و در تیپ SR عمل ست بر ری ست مقدم است.

Negative RLO Edge Detection & Positive RLO Edge Detection:

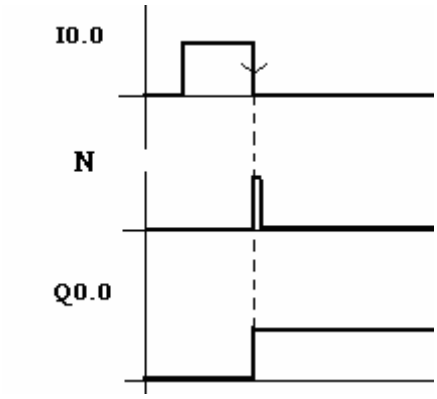
در صورتیکه سیگنالی به آنها برسد ، لبه بالا رونده یا پائین رونده آنرا تشخیص داده و یک پالس با عرض بسیار کم به نشانه دیده شدن لبه مربوطه در خروجی خود میدهند.

هنگام استفاده از این المان نیاز است که به آن آدرسی از حافظه دهید .برای این منظور میتوانید از آدرس یک بیت فلگ که از آن در برنامه استفاده نکرده و نخواهید نمود، بهره ببرید.این فلگ برای انجام عمل مربوط به این المان استفاده خواهد شد.

دستور معادل این المان در زبان STL ، دو دستورFN و FPمیباشد که یکی برای لبه پایین رونده و دیگری برای لبه بالا رونده است.




```
A     I     0.0
FN    M   100.0
S     Q     0.0
```

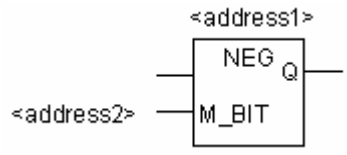


Save RLO into BR Memory:

مقدار RLOرا درBR Binary Result Bitذخیره میکند .

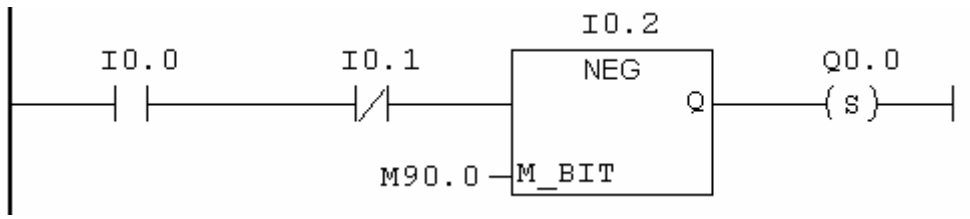
Address Negative Edge Detection & Address Positive Edge Detection:

به دو آدرس احتیاج دارد:



نتیجه منطقی قبل خود را (RLO) در آدرس ۲ ذخیره و سپس با دیده لبه پایین رونده بیت آدرس ۱(چونNEGست) خروجی میدهد.

مثال:



```
A     I      0.0
AN    I      0.1
A(
A     I      0.2
BLD   100
FN    M     90.0
)
S     Q      0.0
```

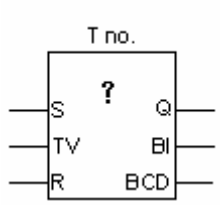
برای اطلاع بیشتر از هرکدام از این المانها میتوانید در محیط نرم افزار Step7 در پنجره سمت چپ محیط OB1 روی آن کلیک کرده و دکمه F1 رافشار دهید.

انواع داده (Elementry Data Types) :

Type and Description	Size in Bits	Format Options	Range and Number Notation (lowest to highest value)_	Example
BOOL(Bit)	1.	Boolean text	TRUE/FALSE	TRUE
BYTE (Byte)	8	Hexadecimal number □	B#16#0 to B#16#FF	L B#16#10 L byte#16#10
WORD (Word)	16	Binary number □ Hexadecimal number □ BCD Decimal number unsigned	2. 0 to 2#1111_1111_1111_1111 W#16#0 to W#16#FFFF □ C#0 to C#999 B#(0.0) to B#(255.255)	L 2#0001_0000_0000_0000 □ L W#16#1000 L word#16#1000 L C#998 L B#(10,20) L byte#(10,20)
DWORD (Double word)	32	Binary number □ □ Hexadecimal number Decimal number unsigned	2#0 to 2#1111_1111_1111_1111 1111_1111_1111_1111 DW#16#0000_0000 to DW#16#FFFF_FFFF B#(0,0,0,0) to B#(255,255,255,255)	2#1000_0001_0001_1000_ 1011_1011_0111_1111 □ L DW#16#00A2_1234 L dword#16#00A2_1234 L B#(1, 14, 100, 120) L byte#(1,14,100,120)
INT (Integer)	16	Decimal number signed	-32768 to 32767	L 1
DINT (Integer, 32□bits)	32	Decimal number signed	L#-2147483648 to L#2147483647	L L#1
REAL (Floating-point number)	32	IEEE Floating-point number	Upper limit: 3.402823e+38 Lower limit: 1.175 495e-38	L 1.234567e+13
S5TIME (SIMATIC time)	16	S7 time in steps of 10 ms (default)	S5T#0H_0M_0S_10MS to S5T#2H_46M_30S_0MS and S5T#0H_0M_0S_0MS	L S5T#0H_1M_0S_0MS L S5TIME#0H_1H_1M_0S_0MS
TIME (IEC time)	32	IEC time in steps of 1 ms, integer signed	-T#24D_20H_31M_23S_648MS to T#24D_20H_31M_23S_647MS	L T#0D_1H_1M_0S_0MS L TIME#0D_1H_1M_0S_0MS
DATE (IEC date)	16	IEC date in steps of 1□day	D#1990-1-1 to D#2168-12-31	L D#1996-3-15 L DATE#1996-3-15
TIME_OF_DAY (Time)	32	Time in steps of 1□ms	TOD#0:0:0.0 to TOD#23:59:59.999	L TOD#1:10:3.3 L TIME_OF_DAY#1:10:3.3
CHAR (Character)	8	ASCII characters	'A','B' etc.	L 'E'

تایمر ها (S5 Timers):

در این PLCها ، پنج نوع تایمر وجود دارد:



S PULSE

S PEXT

S ODT

S ODTS

S OFFDT

تمامی این تایمرها دارای ۶ پایه هستند و تفاوت آنها تنها به لحاظ عملکرد میباشد .این پایه ها عبارتند از :

SET(S)

TIME VALUE(TV)

RESET(R)

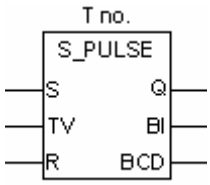
REMAINING TIME ,INT. FORMAT(BI)

REMAINING TIME, BCD FORMAT(BCD)

OUTPUT(Q)

در ادامه نحوه عملکرد این تایمرها به طور مختصر به همراه نمودارهای زمانی براساس سیگنال پایه های تایمر را خواهیم دید.

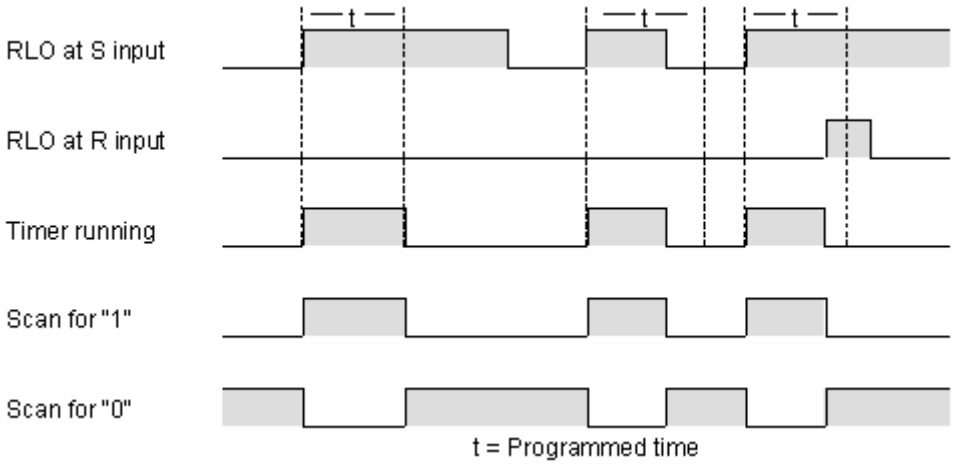
تایمر S PULSE:



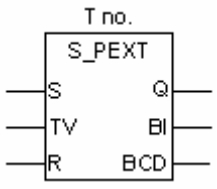
با دیده شدن لبه بالا رونده در پایه S تایمر شروع به کار میکندوخروجی آن یک میشود. پس از گذشت زمان داده شده خروجی تـایمر صـفر میگـردد و برای عمل مجدد به لبه بالارونده روی پایه S نیاز دارد.

توجه فرمایید که اگر حین عمل تایمر سیگنال را از پایه ست بردارم،تایمر از کار میافتد.همچنین در صورت دیده شدن لبه بالارونده در پایـه R تـایمر از کار افتاده ،خروجی آن صفر میشود(حتی اگر سیگنال پایه S برقرار باشد!).

در زیر نمودارهای تایمینگ ین نوع تایمر را ملاحظه می فرمایید:

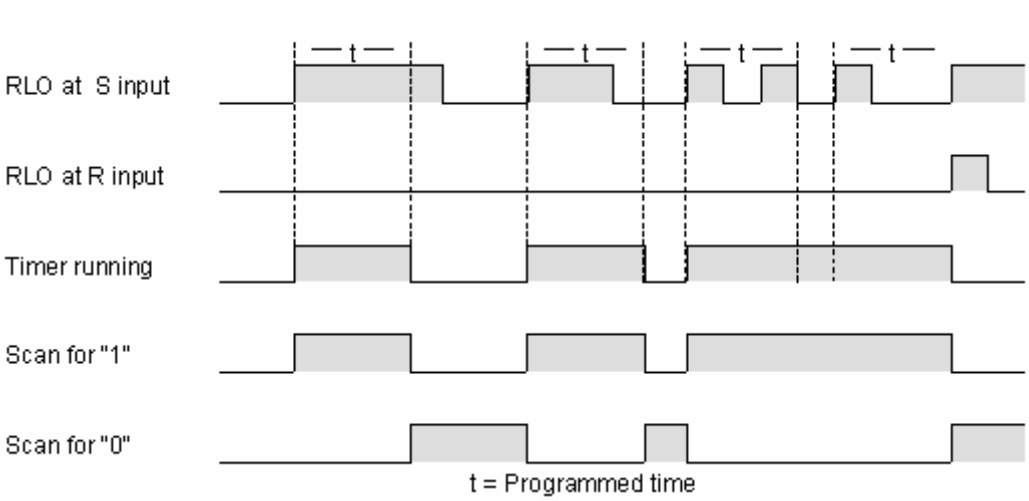


تایمر (Extended pulse)S PEXT:

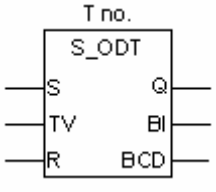


در این تایمر نیز مشابه تیپ قبلی با دیده شدن لبه بالا رونده روی پایه S تایمر شروع به کارکرده،خروجی اش یک میشود و پس از گذشت زمان تعیین شده خاموش وخروجی آن صفر میشود.
نکته: در این تایمر اگر سیگنال پایه S حین کار تایمر صفر شود روی کار آن اثر نخواهد گذاشت.

در هر مرحله از کار تایمر به محض دیده شدن لبه بالارونده در پایه R ،تایمر وخروجی صفر میشوند.

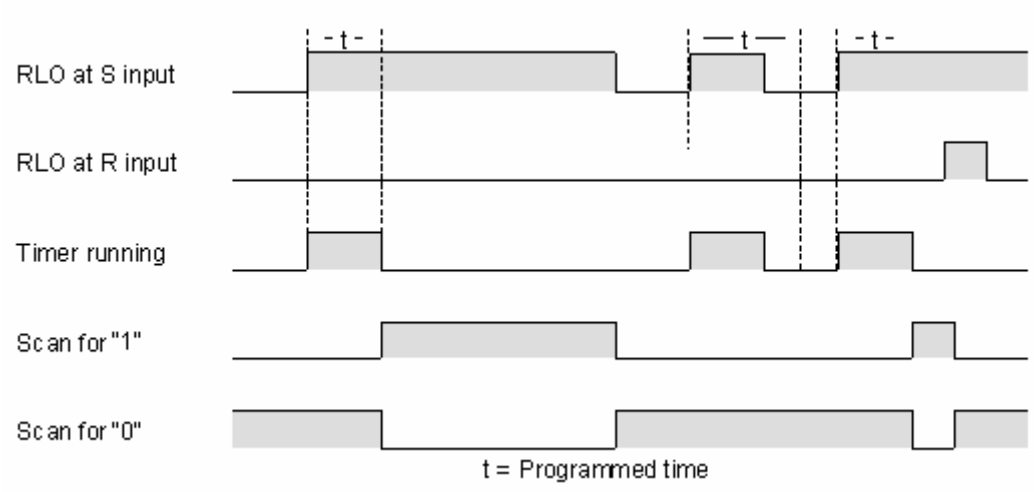


تایمر(S ODT) ON DELAY TIMER:

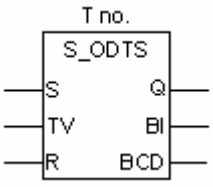


با آمدن لبه بالا رونده روی پایه S تایمر شروع به کار و پس از گذشت زمان تعیین شده، تایمر خاموش و خروجی یک میشود. خروجی تا زمانی که سیگنال ست وجود دارد یک باقی میماند.

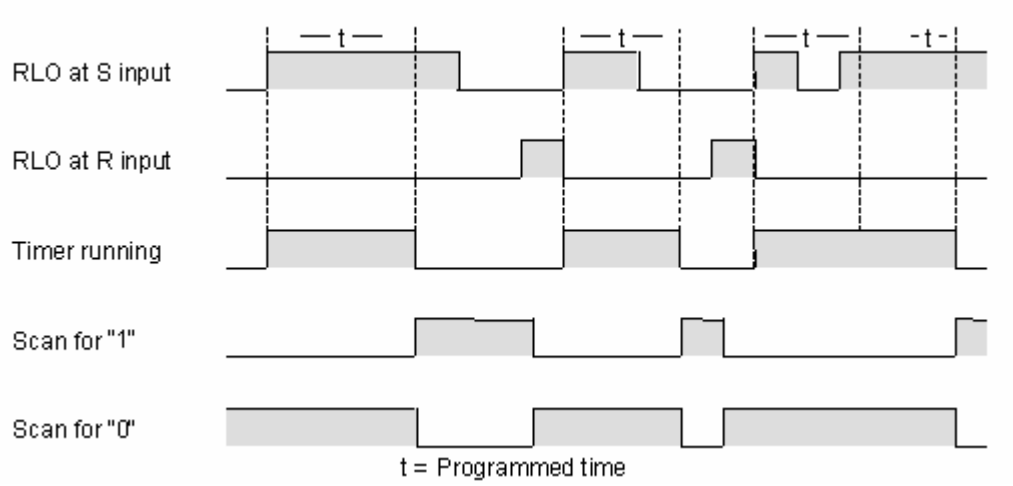
لازم به ذکر است که اگر حین کار تایمر ،سیگنال از پایه ست حذف شود تایمر خاموش میشود .



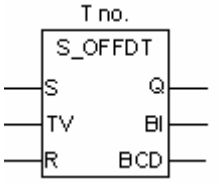
تایمر(S ODTS) ON DELAY TIMER :



در این تایمر که عملکردش مشابه تیپ قبلی است خروجی پس از گذشت زمان تایمر یک شده ،ولی با صفر شدن سیگنال روی S صفر نمیشود بلکه نیاز به یک لبه بالا رونده روی پایه R دارد.همچنین اگر سیگنال روی پایه ست هنگام کار تایمر برداشته شود، عملکرد تایمر مختل نخواهد شد و به کار خود ادامه میدهد.

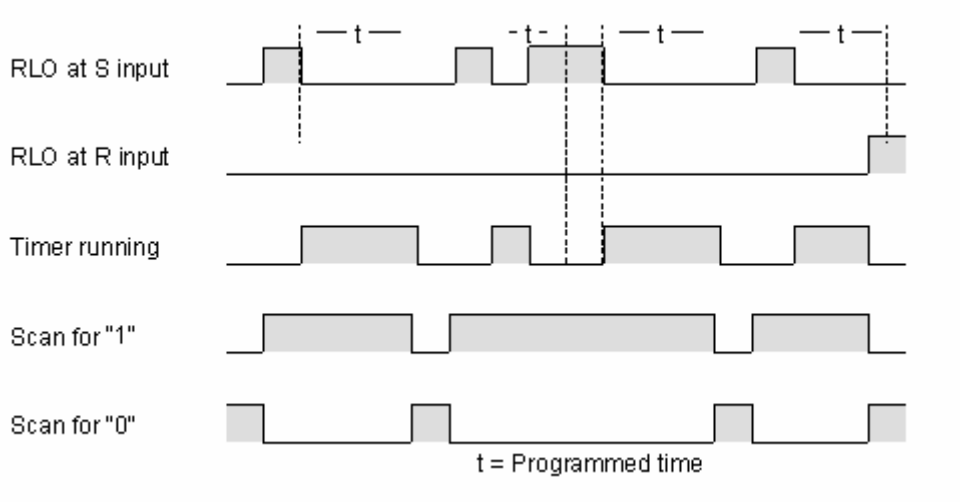


تایمر (S OFFDT) OFF Delay timer:



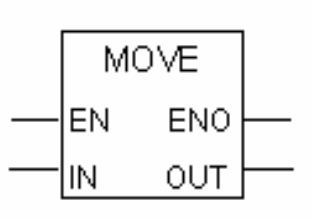
با دیده شدن لبه بالا رونده روی پایه ست ،ابتدا خروجی یک شده سپس به محض قطع شدن سیگنال ست،یعنی دیده شدن لبه پایین رونده روی S تایمر فعال گردیده پس ازگذشت مدت زمان معلوم تایمر خاموش و خروجی نیز با آن صفر میگردد .

لبه بالارونده روی پایه R میتواند تایمر را خاموش و خروجی را صفر کند.



دستور (Move):

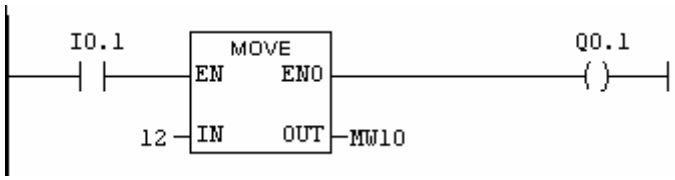
برای انتقال محتویات یک بخش از حافظه یا دیتای جدید به بخشی دیگر، از دستورMOVE استفاده میشود:



پارتهایی که در ورودی و خروجی گذاشته میشوند، میبایست به لحاظ فرمت دیتا با هم یکی باشند وگر نه سبب ازدست رفتن بخشی از اطلاعات میشود.برای مثال دیتا با فرمت بایت در حافظه مقصدی که فرمتی کمتر از بایت دارد ریخته نشود.

در این دستور،با رسیدن لبه بالا رونده به پایه EN عمل انتقال انجام و در صورتیکه این عمل بدون اشکال انجام گردد خروجی ENO فعال میگردد.

در مثال زیر با تحریک ورودی I0.1 عدد صحیح ۱۲ در Memory word10 ریخته میشودسپس خروجی Q0.1 به معنای انجام شدن دستور یک میشود:



معادل STL:

```
      A     I     0.1
      JNB   _001
      L     12
      T     MW    10
      SET
      SAVE
      CLR
_001: A     BR
      =     Q     0.1
```

کانتر ها (COUNTER) :

سه نوع کانتر در PLCهای S7 زیمنس داریم:

S CUD

S CD

S CU

پایه های مشترک این کانترها عبارتند از:

SET(S)

RESET(R)

PRESETTING VALUE(PV)

OUTPUT(Q)

COUNTER VALUE,INT. FORMAT(CV)

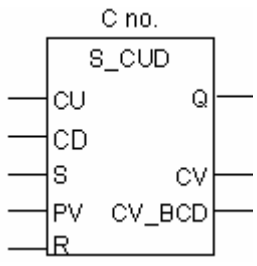
COUNTER VALUE ,BCD FORMAT(CV-BCD)

در کانترهای CD که شمارش معکوس (به سمت صفر) را انجام میدهند علاوه بر پایه های بالا یک پایه CD نیز داریم. در کانترهای CU پایه CU و در CUD ها دو پایه CDوCU وجود دارد.

در این کانتر ها،به محض دیده شدن لبه بالا رونده در پایه ست ،مقدار PVدر CVریخته شده و خروجی یک میشود. به طور کلی با رسیدن پالس به پایه CD مقدار CV یک واحد کم و در پایه CU مقدار CV یک واحد افزایش خواهد داشت. کانتر به محض صفر شدن مقدار CV خاموش و خروجی اش صفر میشود.

باید توجه داشت که در هر مرحله از کار کانتر که لبه بالارونده پالسی به پایه R برسد کانتر از کار میافتد و خروجی اش صفر خواهد شد.

سمبل کانتر S CUD در محیط برنامه نویسیLADDER:

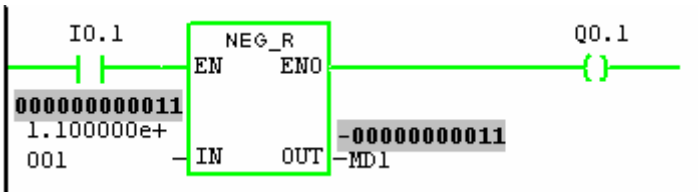


دستورات (COMPARATORS, INT. FUNCTIONS,CONVERTERS):

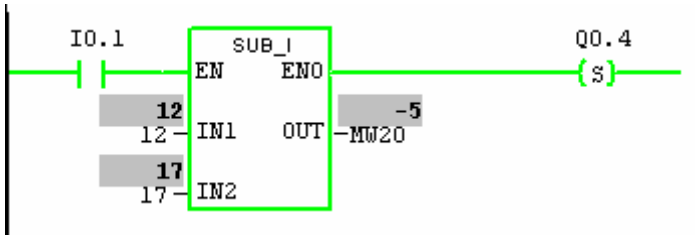
این دستورات شامل انواع فرمانهای مقایسه ای (بزرگتر، کوچکتر،کوچکترومساوی، مساوی و...) برای فرمتهای .INT،.DINT وReal ، توابع تبدیل انواع فرمتهای حافظه به هم و ... مانند BCD به INT یا منفی کردن عدد صحیح موجود در بخشی از حافظه، گرد کردن اعداد حقیقی و ...، همچنین برخی توابع ریاضی ساده مانند جمع ، تفریق و ... در حوزه فرمت های INT و DINT میباشند.

متذکر میشود که برای دیدن اطلاعات کاملتر هر دستوردر محیط نرم افزار S7 روی آن کلیک و دکمه F1 را فشار دهید.

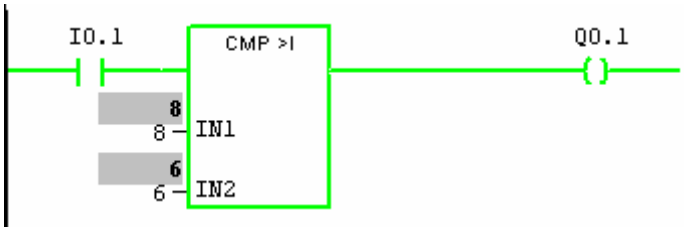
برای مثال در زیر تابع منفی ساز عدد حقیقی را به صورت اجرا شده ملاحظه مینمایید(عدد ۱۱در ورودی به ۱۱- تبدیل شده):



یا در مثال زیر تابع تفریق دو عدد صحیح را به صورت اجرا شده میبینید (INT1 – INT2=حاصل):



ویا در مثال زیر دو عدد صحیح با هم مقیسه و با توجه به نوع بلوک انتخابی(دستور"بزرگتر از") خروجی داده شده:



۳.مسئله ها :

۱- با فشردن کلید START ،لامپ L1 روشن و با رها کردن کلید خاموش شود.(مدارYes)

۲- با فشردن کلیدSTART، لامپ L1 خاموش وبا عدم تحریک کلید لامپ روشن باشد.(مدارNOT)

۳- تنها با تحریک هر دو کلید S1 و S2 لامپ L1 روشن گردد.(AND)

۴- با تحریک یکی از کلید های S1یا S2 و یا هردوی آنها لامپ L1روشن شود.(OR)

۵- مدارهای NOR،NANDرا رسم کنید.

۶- مدار XOR را رسم کنید.

۷- چنانچه کلید S1 به همراه تنها یکی از کلید های S2یا S3 تحریک شده باشد، سیلندرA بیرون رود.
(سیلندرAدوطرفه و شیر یک سر بوبین استفاده شود.)

۸- مادامیکه کلید S1 تحریک شده باشد، سیکل رویرو تکرار شود: A+ B+ A- B-

(سیلندرهای AوB دو طرفه واز شیر یکسر بوبین استفاده کنید.برای داشتن فیدبک از موقعیت جکها از سنسورهای روی جک یا میکرو سوییچهای موجود استفاده نمایید.)

۹- ما دامیکه کلید S1 تحریک شده باشد سیلندرهای A و B (هر دو جک دوطرفه هستند.) برای انجام عمل فید و سوراخکاری سیکل کاری مطابق روبرو دارند: A+ A- B+ B-

(از شیرهای یکسر بوبین و سنسورها و میکروسوییچهای موجود استفاده نمایید.)

۱۰- با زدن کلید S1 سیلندر A بیرون رود. با تحریک مجدد S1(در هرلحظه) سیلندر A به داخل بازگردد.

۱۱- با زدن START ،سیلندر A(جک دوطرفه) بیرون رفته و پس از گذشت ۵ ثانیه به داخل باز گردد.(توسط شیر دو سر بوبین).توجه شود که اگر دست روی کلید ماند ،سیکل دوباره تکرار نشود و تنها با تحریک مجدد دوباره انجام شود.

۱۲- مادامیکه کلید START تحریک شده باشد،لامپ L1 با فرکانس یک هرتز چشمک بزند.(Flasher 1Hz)

۱۳- با زدن ۵ مرتبه کلید START لامپ L1 با فرکانس یک هرتز شروع به چشمک زدن کند.

۱۴- با زدن START سیکل روبرو انجام شود: A+ B+ B- B+ B- B+ B- A-

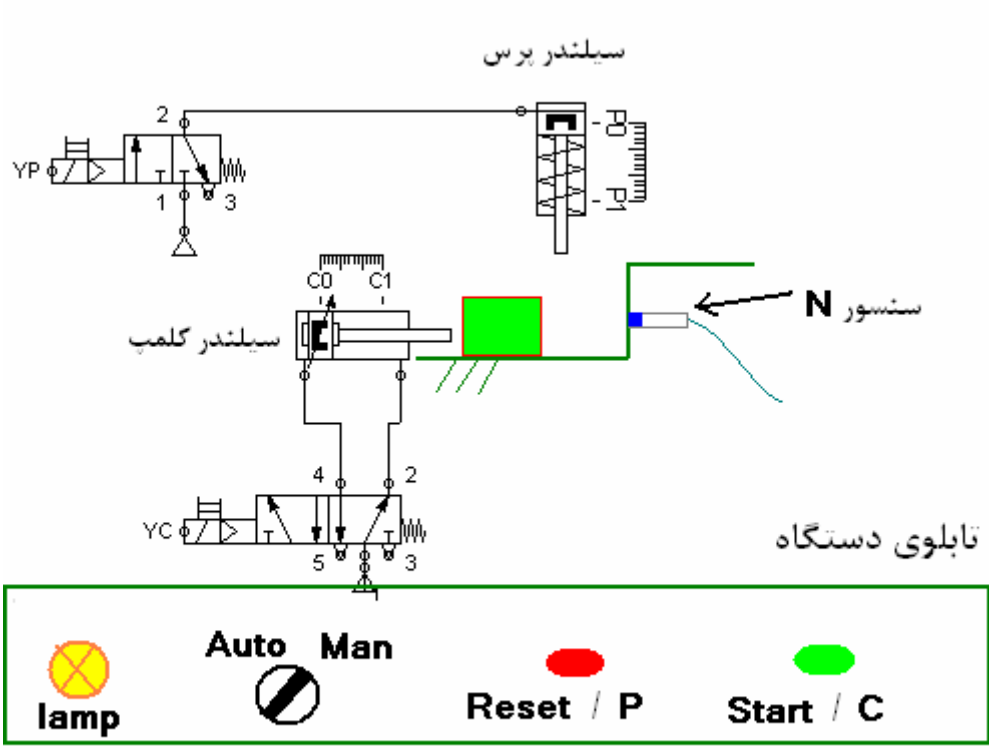
(جکهای دو طرفه ،شیر سیلندر A یک سر بوبین و برای سیلندرB دو سر بوبین استفاده نمایید.)

۱۵- در صورت تحریک start به مدت ۵ ثانیه سیلندر A سه مرتبه مثبت و منفی شده ،پس از آن منتظر فرمان بماند.

۱۶- دستگاه پرس کوچکی به شرح زیر داریم.برنامه آنرا نوشته و تست نمایید.

یک کلید خارکی (سلکتوری) برای انتخاب حالت اتوماتیک و یا دستی رو ی تابلوی دستگاه قرار دارد.

- حالت دستی: با زدن شستی C ،سیلندر کلمپ بیرون رفته و قطعه را میگیرد. حال اپراتور میتواند با استفاده از شستی P هر تعداد بار پرس که لازم است با زمان دلخواه انجام دهد.
- حالت اتوماتیک: با زدن کلید Start، در صورتیکه سنسورNتحریک شده باشد ، سیلندر کلمپ قطعه را گرفته و سیلندر پرس قطعه را ۴ مرتبه پرس میکند.در هر بار پرس، وقتی سیلندر پرس پایین میاید، ۱ثانیه توقف داشته سپس به داخل باز میگردد. در حالت اتومات، با تحریک Reset پروسه پرس اتوماتیک ری ست شده و دستگاه آماده استارت دوباره میشود.در مدت زمانی که در حالت اتوماتیک قرار داریم ،لامپ با فرکانس ۲ هرتز چشمک میزند.
- نکته: اگر هنگام کار پرس در مد اتوماتیک ،کلید خارکی به مد دستی برده شود، سیکل اتوماتیک تا انتها طی شود سپس در مد دستی قرار گیرد(چشمک زن خاموش میشود).



شکل مسئله ۱۶

۱۷- با سه بار تحریک S1 لامپ L1و با سه با دیگر تحریک آن لامپ L2نیز روشن شودو اگر سه بار دیگر تحریک شود،هردو لامپ خاموش شود.

یادداشت