

بِنَامِ خدا

# گزارش کارآموزی

موضوع:

PLC

محل کارآموزی:

شرکت آموزشی موج علم لاهیج

استاد راهنما:

جناب آقای دکتر علیرضا سحاب

کارآموز:

محمد باقری

مهندس برق الکترونیک

دانشگاه آزاد اسلامی واحد لاهیجان

## پیشگفتار

نخستین بار در اوایل سالهای دهه ۷۰ بود که کنترل کننده های قابل برنامه ریزی در صنایع موتوری-جایی که تابلوهای بزرگ رله ای مورد استفاده قرار می گرفت- به طور گسترده به خدمت گرفته شده و جایگزین این تابلوها گردید. آنها توانستند کنترل لازم را با اشغال فضای بسیار کمتری نسبت به مدارهای رله ای معادلشان فراهم کنند. همچنین در زمانهای کارکرد طولانی تر، بسیار مطمئن تر عمل می کردند. مهمتر از همه، آنها بی نهایت قابل انعطاف بودند و اگر نیازی به اصلاح یا تغییر ترتیب مولفه های یک سیستم کنترل پیش می آمد، به سادگی قابل اعمال بود. اکنون دیگر این امکام فراهم آمده بود که یک سیستم کنترل بدون کلیدزنی یا قطع کردن سیم و صرفاً توسط تغییر در برنامه تغییر یابد.

در ده سال اول تولید عام، بازار کنترل کننده های قابل برنامه ریزی در بریتانیا، شاهد رشدی معادل تقریباً از صفر تا بیش از چهل میلیون دلار فروش سالیانه بوده است. در زیر تعدادی از تولید کننده های معروف لیست شده اند:

Saab Allen Bradley Co

Siemens Ltd General Electric Fanuc

Square D GEC

Texas Instruments Gould Ltd

Telemecanique Klockner Moeller

Toshiba Mitsubishi Electric Ltd

Westinghouse MTE

آهنگ رو به رشد استفاده از کنترل کننده های قابل برنامه ریزی در فرآیندهای صنعتی، کارخانجات تولید کننده PLC را تشویق کرده است تا اقدام به ایجاد و توسعه خانواده ای از PLC های اساس میکروپروسسوری با سطوح متفاوتی از کارایی های عملیاتی، نمایند. هم اکنون گستره کنترل کننده های قابل برنامه ریزی در دسترس، از PLC های کوچک کامل و جامع با بیست ورودی - خروجی و پانصد مرحله برنامه ریزی تا سیستم های modular پیچیده را شامل می شود. این راهکار یعنی از پیش ساخته شدن مازولها و سپس اتصال آنها به PLC، امکان گسترش و بهبودی یک سیستم را با حداقل هزینه و اشکالات، فراهم آورد.

با توجه به این که در کشور ما نیز همانند اغلب کشورهای دنیا از PLC های Simens در اکثر صنایع بزرگ نظیر نیروگاهها و پتروشیمی ها استفاده می شود. این نوع PLC با توجه به آخرین محصولات موجود در بازار که از سری S5 و S7 می باشند و همچنین در دسترس بودن این نوع در محل کارآموزی مطالب این گزارش حول آن خواهد بود.

لعدم پدر و مادرم ...

## فهرست مطالب

فصل اول : معرفی PLC

فصل دوم: زبان های برنامه نویسی

فصل سوم: محیط نرم افزاری STEP7

فصل چهارم: پنوماتیک در PLC

فصل پنجم: معرفی سخت افزاری PLC S7-۳۰۰

فصل ششم: آماده سازی S7-۳۰۰

فصل هفتم: مثال های کاربردی و صنعتی

ضمایم:

ضمیمه ۱: لیست دستورات STL در S7-۴۰۰ و S7-۳۰۰

ضمیمه ۲: فهرست کدهای ماشین در PLC های زیمنس

ضمیمه ۳: فهرست خطاهای موجود در ISTACK

منابع

نیو اور

## فصل اول: معرفی PLC

از عبارت programmable logic controller به معنای کنترل کننده منطقی قابل برنامه ریزی گرفته شده است. PLC کنترل کننده نرم افزاری است، که در قسمت ورودی اطلاعات را به صورت باینری دریافت و آنها را طبق برنامه ای که در حافظه اش ذخیره شده پردازش می نماید. و در نتیجه عملیات را نیز از قسمت خروجی به صورت فرمان هایی به گیرنده ها و اجرا کننده های فرمان ارسال می نماید به عبارتی دیگر plc عبارت از یک کنترل کننده منطقی است که می تواند منطق کنترل را توسط برنامه برای آن تعریف نمود و در صورت نیاز به راحتی آن را تغییر داد.

### تفاوت plc با کامپیووتر:

استفاده از کامپیووتر مستلزم آموزش های نسبتا طولانی، صرف وقت و هزینه‌ی بسیار است. چنانچه کنترل فرآیندی مورد نظر باشد استفاده از کامپیووتر به مراتب پیچیده‌تر و در اغلب موارد عملاً ناممکن می شود. اما plc کنترل آن فرآیند را با کمترین هزینه و بهترین شکل ممکن در اختیار قرار می دهد.

استفاده از plc بسیار آسان بوده و نیاز به آموزش های مفصل ندارد. تمامی اجزای کامپیووتر در سیستم plc وجود دارد خروجی یک plc می تواند یک ترانزیستور، رله، دیود و یا هر چیز دیگری باشد. تفاوت عمدی با کامپیووتر این است که در plc ما نتیجه عمل را می بینیم ولی در کامپیووتر فقط اطلاعات را مشاهده می کنیم.

### : تاریخچه plc

در سال ۱۶۲۳ میلادی یک آلمانی ویلیام شیکار، با استفاده از چرخ دنده های ساعت ماشین محاسباتی کاملاً میکانیکی ساخت که عمل جمع و تفریق را انجام می داد این ماشین در سال ۱۶۴۰ توسط پاسکال کامل تر شد و در ۱۶۷۲ یک ریاضی دان آلمانی ماشینی ساخت که می توانست ۴ عمل اصلی را انجام دهد او، سیستم عدد باینری را پایه گذاری کرد. در سال ۱۹۵۳ میلادی ترانزیستور به وجود آمد و پس از آن دستگاه های شمارش ساده تر و کوچک تر شدند و در سال ۱۹۶۸ میلادی اولین plc توسط آمریکایی ها در شرکت جنرال موتور ساخته شده که به نام pc و یک کامپیووتر صنعتی بود. در سال ۱۹۷۲ میلادی آلمانی ها توسط شرکت زیمنس اولین plc ها را وارد بازار کردند. اولین plc که شرکت زیمنس ارائه کرد step-5 به نام simatic که به dose معروف و تحت dose بوده است. این محیط مشکلات بسیار زیادی داشته مخصوصاً در رفع عیوب و مشکلات.



بعد از ابداع سیستم عامل ویندوز برنامه را تحت  
ویندوز در آوردن که منجر به ساخت step7-  
۲۰۰,plc توسط آمریکایی ها شد پس از مدتی  
آلمانی ها ویندوز را تحت اختیار خود گرفتندو با  
ارتفاع S5 به ۴۰۰,۳۰۰,S7-۲۰۰ ساخته شد.

: plc بهترین

Siemen (۲) Alen bradlly (۱)

Omron (۳)



نمونه ای از دو PLC

## دلایل استفاده محدود plc در ایران :

۱) ارتباط مشکل با منابع تامین کننده خارجی

۲) عدم دسترسی به موقع به اطلاعات سیستم ها

۳) عدم پشتیبانی موثر سازندگان از تجهیزات فروخته شده خود

۴) هزینه بالای تجهیزات خارجی

۵) هزینه بالای آموزش در خارج از کشور

شرکت های داخلی نیز با توجه به مشکلات یاد شده و برای پر کردن خلا موجود اقدام به طراحی و ساخت چند نوع plc نمودند . plc های مذکور کلیه امکانات استاندارد plc های متداول را داشته . از شرکت های داخلی تولید کننده plc می توان شرکت contorlnic را نام برد که زبان برنامه نویسی آن C STL می باشد .

نکته: یکی از خصوصیات شرکت زیمنس تنوع محصولات آن است که ورژن ها و قیمت ها و کاربردهای مختلفی دارد، همچنین به دلیل آنکه می توانند ارتقا یابند، کاربرد بیشتری دارند.

## plc های مختلف زیمنس:

در طبقه بندی محصولات زیمنس plc ها در زیر مجموعه زماتیک قرار می گیرد. برخی از آن هایه صورت فشرده و متراکم (compact)، طراحی و ساخته شده اند به این معنا که منبع تغذیه و CPU و ورودی و خروجی به صورت یکپارچه در کنار هم، به یکدیگر متصل هستند و یک واحد تلقی می شوند. برخی دیگر به صورت مدولار هستند که بر خلاف نوع اول، کاربر می تواند مدول های دلخواه از آن خانواده وابسته به نیاز خود انتخاب و در کنار هم قرار دهدن. plc های زیمنس را می توان به ۵ خانواده زیر تقسیم نمود:

۱) زما تیک S5

۲) زماتیک S7 .

S7-۳۰۰

S7-۴۰۰

. Logo(۳

## C۷) زمانیک ۴

## ۵) زمانیک ۵۰۵

### موارد کاربرد Plc :

۱) کنترل ماشین یا هر وسیله برقی ۲) کنترل هر سیستم خط تولید ۳) کنترل مدار فرمان

مهمترين مشخصه اي که در انتخاب Plc باید در نظر گرفت :

- |                   |                   |                |                               |                    |                 |                      |
|-------------------|-------------------|----------------|-------------------------------|--------------------|-----------------|----------------------|
| ۱) تعداد وروديهای | ۲) تعداد خروجي ها | ۳) تعداد فلگها | ۴) تعداد شمارنده ها و تايمرها | ۵) نوع فلگ و تايمر | ۶) اندازه حافظه | ۷) سرعت اجراي برنامه |
|-------------------|-------------------|----------------|-------------------------------|--------------------|-----------------|----------------------|

### استاندارد IEC ۱۱۳۱ :

بدليل آنکه زبان برنامه نويسي Plc های مختلف دارای شباهت ها و تفاوت های گوناگون تر می باشد کمسيون جهانی فناوري برق که يك سازمان بين المللی برای ارایه ی استانداردهای جهانی در فناوري برق است در سال ۱۹۷۹ به بررسی جامع Plc ها شامل سخت افزار برنامه نويسي و ارتباط به عهده گرفت. هدف اين گروه تدوين روش های استانداردي بود که موارد فوق را پوشش می داد و توسط سازندگان Plc بكار گرفته شد و سرانجام ۱۲ سال بعد توانستند اين استاندارد را ارایه دهند. اما با اين وجود سازندگان Plc هنوز نتوانستند زبان Plc مورد نظر را با زبان IEC مطابقت دهند و تفاوت های زیادي مخصوصا با زبان S<sub>7</sub> وجود دارد که از جمله آن می توان به موارد زير اشاره نمود :

۱) تعداد دستورهای IEC کمتر از S<sub>7</sub> است.

۲) دستورهای IEC عمومی، قابل درک برای همه مليت ها می باشد.

۳) زمان اجرای دستورهای IEC معمولا طولاني تر است.

۴) دستورهای IEC فقط به صورت های FBD و LAD می باشد.

۵) بعضی از دستورهای IEC همانند شمارنده‌ها، تایمرها و غیره به گونه‌ای متفاوت با مشابه  $S_7$  خود می‌باشد.

۶) هنگام استفاده از دستورهای IEC به نوع داده حساس است و در صورت اشتباه خطأ می‌دهد.

: PC

به کامپیوترهای شخصی که نرم افزار مورد نیاز plc در آن نصب باشد و نیاز به PC Adaptor دارد.

: PC Adaptor

برای آنکه برنامه را از PC به PLC Download کنیم ، باید توسط یک واسطه به نام PC Adaptor انجام گیرد.

دو نوع PC Adaptor وجود دارد، یک نوع به درگاه سریال RS۲۳۲ و نوع دیگر به درگاه USB متصل می‌شود.

: RS۲۳۲ ✓

در plc ها معمولاً برای ارتباط با سایر قسمتها و ارسال و دریافت داده‌ها از خطوط سریال استفاده می‌شود. جهت ارتباط سریال استانداردهای وجود دارد که مهمترین آن RS۲۳۲ می‌باشد. این استاندارد ارتباط سریال در فوائل کوتاه است ، این استاندارد ، اتصالات الکتریکی و فیزیکی ، ارتباط بین سیگنالها و روند مبادله اطلاعات را تعریف می‌کند ، در ارتباط سریال سه ویژگی وجود دارد :

- ۱ - سرعت انتقال: یعنی تعداد بیت ارسالی در ثانیه و عرض پالس هر بیت.
- ۲ - سطح ولتاژهای منطقی: یعنی بیت ۱ و ۰ با چه ولتاژی نشان داده می‌شود.
- ۳ - نحوه همزمانی دادها تا گیرنده ، داده‌ها را به طور صحیح دریافت کند.

سُبْرَة

## فصل دوم: زبان های برنامه نویسی

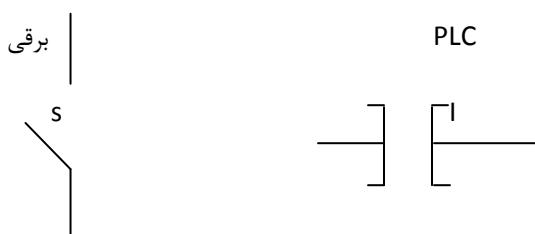
روش های نمایش برنامه های زیر در تمام plc ها مشترک می باشد:

### (1) دیاگرام نرده بانی : (LAD) LADPER DIAGRAM :

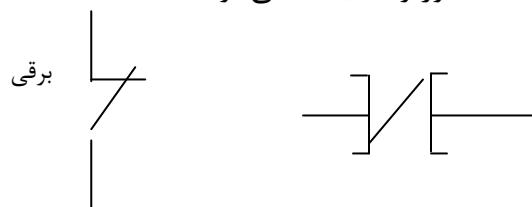
در سیستم انگلیسی با LAD و در سیستم آلمانی KOP نمایش می دهند این روش برنامه نویسی بیشتر برای مدارات برقی استفاده می گردد که بر اساس کلید های N.O و نرمالی N.C کار می کنند. کاربرد این زبان به حدی است که بعضی از شرکت ها تنها با این زبان کار می کنند .. ساده ترین زبان بوده و در این روش نقشه های برق صنعتی را ۹۰ درجه می چرخانیم و جای المان های برقی المان های plc را قرار می دهیم.

تیغه باز: تیغه باز می تواند start ، بی متال، سنسور، میکرو سوئیچ و یا هر چیز دیگری از جنس ورودی باشد بنابراین با حرف I نمایش داده می شود . همچنین می تواند از جنس خروجی هم باشد، مثل تیغه های کمکی کنتاکتور.

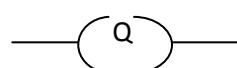
هر گاه کلید باز از جنس خروجی بود شکل آن تغییر نداشته فقط آن را با حرف Q نمایش می دهند.



تیغه بسته: کلیدهای بسته میکروسوئیچ ، رله ، کنتاکتور و یا stop می تواند باشد.



کلید خروجی: Q نمایش داده می شود و به شکل زیر است :



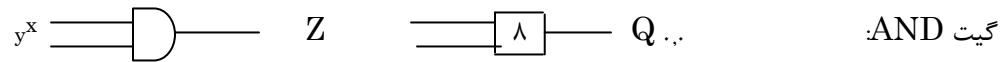
برای فهمیدن plc باید آدرس ورودی خروجی را به آن بدهیم. دو منطقه آدرس داریم Address Area و خود به دو قسمت input area و output area تقسیم می شود. آدرس ، بایت ها و بیت های یک بایت را به آن می دهیم. مثلا. I.. یعنی بایت صفر ورودی و ..Q. یعنی بیت صفرم بایت صفر ورودی .

نکته : plc در واقع منطق آن طراحی است بنابراین با روش سعی و خطاب به جواب می رسیم.

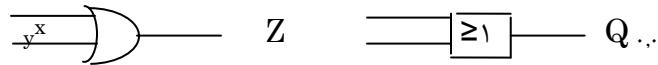
### روش دوم : (CSF) Control System Flouchart :

با CSF یا FBD (Function Block Diagram) نمایش داده می شود. در آلمانی با FUP نمایش داده می شود. این روش در الکترونیک کاربرد دارد. که با استفاده از کلیدها، عمل برنامه سازی صورت می گیرد در این سیستم ورودی ها از سمت چپ وارد و از سمت راست خروجی ها خارج می شود.

در این زبان بیشتر با گیت های منطقی کار می کنیم. گیت ها را به صورت مربع یا بلوك نشان می دهند.



: OR گیت



برای NOT یک ورودی در پایه آن علامت —O— قرار می دهند.

سومین زبان :

لیست بیانی :

### ( statements list)( STL)

در سیستم انگلیسی با STL و در سیستم آلمانی با AWL نمایش می دهند. در این روش برنامه نویسی که بیشتر در کامپیوتر کاربرد دارد کمیت ها بر حسب پارامترهایی مشخص می شود که مقادیر این پارامترها بر حسب کار تغییر می کند.

عبارات منطقی :

هر عبارت منطقی به ۲ قسمت تقسیم می شود.

۱) عملگرها : که نشان دهنده ی عملی است که باید انجام شود.

۲) عملوندها : نشان دهنده ی آن است که عمل روی چه چیز انجام شود.

عملگرها	مفهوم
A	AND
AN	AND NOT
O	OR
ON	OR NOT
=	خروجی Q
عملوندها	*****
I	ورودی ها
Q	خروجی ها
F	فلگ ها
C	شمارندها
T	تایمرها

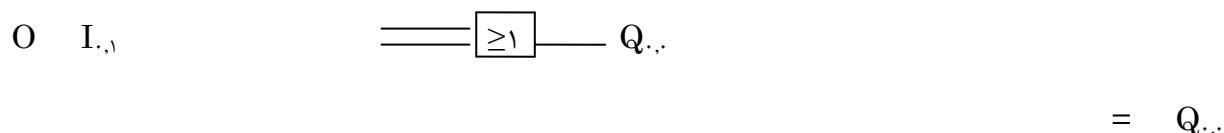
A I,. : دستور برنامه AND



$$= Q..$$

BE  $\Rightarrow$  Block End

O I,. : دستور برنامه OR



کاملترین روش برنامه سازی است ، زیرا زبان سیستم عامل است. همچنین تمام برنامه هایی که به زبان STL نوشته می شود ، نمی توان به زبان های دیگر نوشت. اما٪ ۹۰ ، STL به زبان CSF و٪ ۸۵ LAD به STL تبدیل می شوند، همچنین یک سری بلوک هایی وجود دارند که فقط به زبان STL نوشته می شوند . همانند FE ها، بلوک های تابع و غیره...

نکته ۱: همیشه قبل از شروع برنامه نویسی اول تجهیزات را چک کنید.

نکته ۲: بلوک OR نویز پذیر است اما AND نویز پذیر نیست.

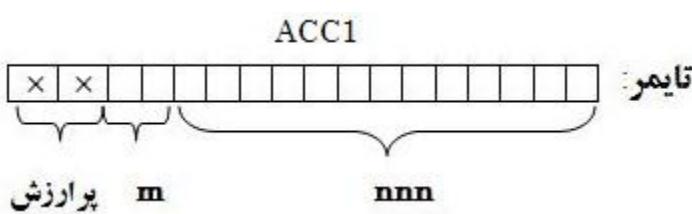
نکته ۳: قرارداد میکنیم که تمام تک ورودی ها را با AND نمایش دهیم.

#### آکامولاتور یا انباره :

به عنوان قسمتی از حافظه می باشد که جهت مبادله ای اعداد احتیاج به یک حافظه ای واسطه است که به آن انباره گفته می شود . به عبارتی دیگر انباره ها حافظه های کمکی CPU می باشند که در قسمت های مختلف PLC ما نند فلک ها ، مبادله ای اعداد بین آدرس های مختلف ، عملیات مقایسه ای ، خروجی ها و ورودی ها از آن استفاده می شود . در PLC های S5 معمولاً دو آکامولاتور وجود دارد که این آکامولاتور ها ۱۶ بیتی هستند

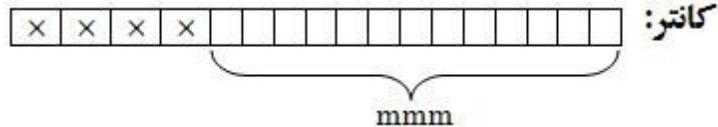
هر آکامولاتور دارای یک لیست با ارزش و یک لیست کم ارزش می باشد علت استفاده از آکامولاتورها ، افزایش سرعت در PLC ها می باشد.

نکته : مقدار TV ، تایمر و همچنین مقدار  $ACC_1$  بارگذاری می شود. مقدار عددی آنها بدین صورت در  $ACC_1$  بارگذاری می شود .



بیت	بیت	بیت های زمانی
۱۴	۱۳	۰۰۱ S
۰	۱	۰.۱ S
۱	۰	۱ S

### فلک :



فلک های plc همانند حافظه های Ram می باشند . فلک ها می توانند به صورت پایدار یا ناپایدار باشند . تعداد فلک های پایدار یا ناپایدار بستگی به نوع plc دارد. کاربرد دیگر فلک ها ، خلاصه سازی برنامه ها می باشد، که در این حالت می توان با حذف پرانتزهای متداخل از فلک ها استفاده کرد که سبب کاهش زمان اجرای برنامه می شود .

فلک به معنای پرچم یا نشانه می باشد در plc به معنای علامتی برای انجام شدن یا انجام نشدن عملی است.

### کاربرد فلک ها :

۱) خلاصه سازی برنا مه : پرانتز های متداخل را از بین می برد زیرا این پرانتزهای متداخل زمان اجرای برنامه را افزایش داده و plc را به stop می برد .

۲) به عنوان حافظه واسطه

### انواع فلک ها :

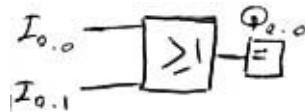
۱) فلک تقدم set بر

۲) فلک تقدم reset بر

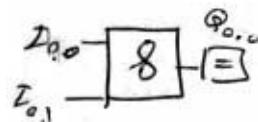
### دستور العملهای Step ۷

۱- دستورات منطقی Bit logic

oR این دستورات چک می کند ، که آیا وضعیت بیت آدرسی داده شده است یا نه و سپس RL0 را با وضعیت این بیت oR می کند .



AND (&) این دستور چک می کند که آیا وضعیت Bit آدرسی داده شده است یا نه و سپس آن را با وضعیت این بیت and می کند.



1--: این دستور شاخه ای به شاخه oR یا AND و.... اضافه می کند

0--: این دستور بیت لازمه را نات می کند. یعنی اگر  $0 \leftarrow 1$  او اگر  $1 \leftarrow 0$  می کند.

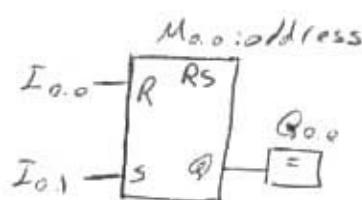
=--: خروجی را با این المان نشان داده می شود.

RS: فیلیپ فلاپ عمل ست و ری ست را وقتی که  $L_0 = 1$  باشد، اجرا می کند بنابراین اگر  $RLO = 0$  باشد، این دستور اجرا نمی گردد.

عمل ری ست کردن بیت آدرسی داده شده وقتی اتفاق می افته که ورودی  $R = 1$  و ورودی  $s = 0$  باشد.

عمل ست کردن بیت آدرس داده شده وقتی اتفاق می افته که ورودی  $R = 0$  و ورودی  $s = 1$  باشد.

اگر هر دو ورودی اشوند، نیز عمل ست انجام می شود



iMP نکته: در s1 فلگ را با M نمایش می دهند.

MP نکته : هر پایه به خروجی نزدیک تر باشه آن عمل تقدم دارد :

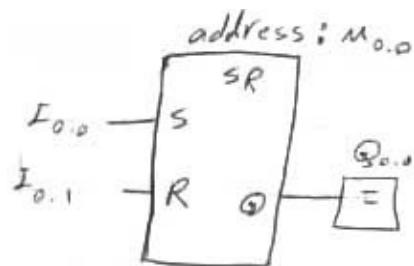
آدرس که مشخص می کند که ام بیت ست یا ری ست شود : address

S: فعال کردن ست

R: فعال کردن ری ست

Q: خروجی

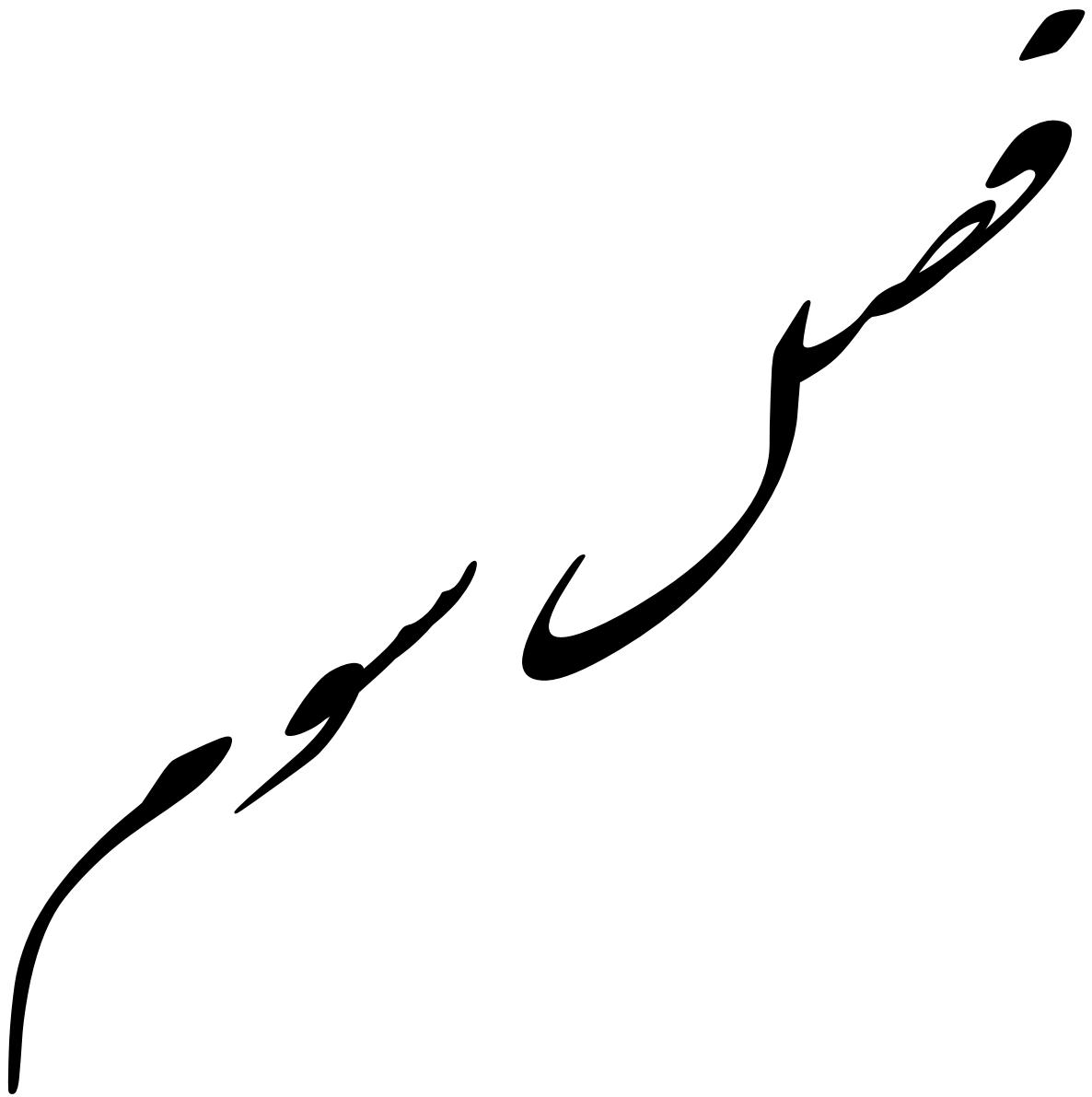
SR : این فیلیپ فلاپ هم زمانی عمل می کند که R باشد



انواع تایمر:

نوع تایمر	وضعیت تایمر	توضیح	کاربرد تایمر
-		-	-
s-pulsE		با یک شدن ورودی بکار می افتد و خروجی آن تا وقتی شرایط زیر برقرار باشے یک باقی می ماند : ۱) زمان t سپری نشده	به شرط داشتن ورودی TV : اگر بخواهیم خروجی کاملاً OUtT از ورودی شکل بگیرد

		باشد) ورودی ۱ باشد	
S-PEXT		با یک شدن ورودی بکار می افته و خروجی آنتا وقتی $t$ سپری نشده باقی می ماند	زمان TV برای ما هم باشند و در ابتدا خروجی یک باشد
S-ODT		با یک شدن ورودی بکار می افتد و خروجی آن ابتدا صفر بعد از $t$ بشرط داشتن ورودی یک می شود با صفر شدن ورودی صفر می شود	-TV ورودی : out_T در تأخیرها مثل راه اندازی موتور .تابع ورودی هم باشد
S-ODTS		با یک شدن ورودی بکار می افتد و خروجی آن ابتدا صفر سپس بعد از $t$ حتی اگر ورودی هم صفر باشد یک می شود	قبل از -TV+ ورودی : در تأخیرهای که حالت پایدار وجود داشته باشد
S-OffDT		با یک شدن ورودی بکار می افتد ولی زمان t از زمانی شروع می شود که ورودی صفر شود بعد از (t) آن صفر می شود	+TV ورودی : out_T در حالاتی که منجر ورودی به زمان دیگری نیز برای یک بودن خروجی نیاز باشد



## فصل سوم: محیط نرم افزاری STEP7

### نیازمندی های PC یا PG جهت نصب S7 :

- ۱) بسته به نوع STEP7 ، ویندوز ۹۸ یا ۲۰۰۰ یا XP می تواند مورد استفاده قرار گیرد .
- ۲) پردازنده باید ۸۰۴۸۶ یا بالاتر باشد .
- ۳) حافظه RAM مورد نیاز ، ۳۲ مگابایت است. سازنده توصیه می کند که ظرفیت RAM حداقل ۶۴ مگابایت باشد و حداکثر حافظه ای مجاز سیستم توسط کاربر انتخاب شود . تا برنامه عملکرد مناسبتری داشته باشد .
- ۴) دیسک سخت باید حداقل ۴۵۰ مگابایت فضای خالی جهت نصب step7 و ذخیره ای پروژه ها ، داشته باشد.
- ۵) برای ارتباط pc با plc به یک کارت MPL نیاز است . این کارت روی PG ها نصب است . دو نوع Adaptor وجود دارد، که یک نوع به پورت USB و نوع دیگر به RS۲۳۲ متصل می گردد .  
نکته : قابل ذکر است که هر چه سرعت پردازنده ، میزان RAM و فضای خالی دیسک سخت بالاتر باشد ، کارایی اجرای نرم افزار بالاتر خواهد رفت .

### نصب نرم افزار Step7 :

- پس از قرار دادن CD نصب Step7 در کامپیوتر، برنامه به صورت خودکار اجرا خواهد شد .
- در هنگام نصب نرم افزار Step7 به نکات زیر توجه کنید :
- ۱) ممکن است با آغاز عملیات نصب پیغامی مبنی بر فقدان یکی از المان های ویندوز نمایش داده شود و عملیات نصب لغو گردد . در چنین حالتی ابتدا المان مورد نظر را نصب کرده و سپس مجددا برای نصب اقدام نمایید .
  - ۲) در ابتدا دو بار Next را زده ، پنجره ای باز می شود که اجزای مختلف نرم افزار Step7 را نمایش می دهد .
  - ۳) در هنگام نصب اجزای مورد نظر را انتخاب نموده ، با انتخاب هر مورد میزان فضایی که روی دیسک اشغال می کند نشان داده می شود . برای کار با این نرم افزار فقط کافیست گزینه Step7 ، ورژن ۵/۳ و دو گزینه ای آخر تیک دارشوند .

۴) برای نصب مجوز نرم افزار، فلاپی مجوز را درون فلاپی درایو قرار می دهیم ، سپس درایوی را قصد داریم مجوز به آن منتقل شود انتخاب می کنیم . بعد از عبور از این مرحله مجوز به کامپیوتر منتقل خواهد شد .

۵) اگر مجوز نرم افزار Step ۷ را نداریم، باید در مرحله ی قبل گزینه ی No Install Authorization را انتخاب کنیم و از این مرحله عبور کنیم .

۶) در مرحله ای که برنامه ی نصب سریال را در خواست می کند در کادر ظاهر شده و سریال ۰۰۷۸۸۰۲۳۲۷ را وارد نمایید .

Step ۷ ، پنج زبان : انگلیسی ، آلمانی ، ایتالیایی ، فرانسوی و اسپانیایی را پشتیبانی می کند .

شما باید در هنگام نصب زبان مورد نظر را انتخاب کنید .

بعد از تکمیل مراحل نصب لازم است ، کامپیوتر دوباره راه اندازی شود . بعد از راه اندازی مجدد کامپیوتر در منوی Start یک گروه برنامه را با نام Simatic خواهید دید که اجزای نرم افزار Step ۷ در آن قرار دارد .

### مجوز Authorization

، یک نرم افزار حافظت شده است که با وجود مجوز بدون مشکل کار می کند در غیر این صورت مرتباً پیغامی مبنی بر نبود آن صادر می شود . به طور کلی مجوز Step ۷ به دو صورت می باشد :

#### ۱) مجوز اصلی :

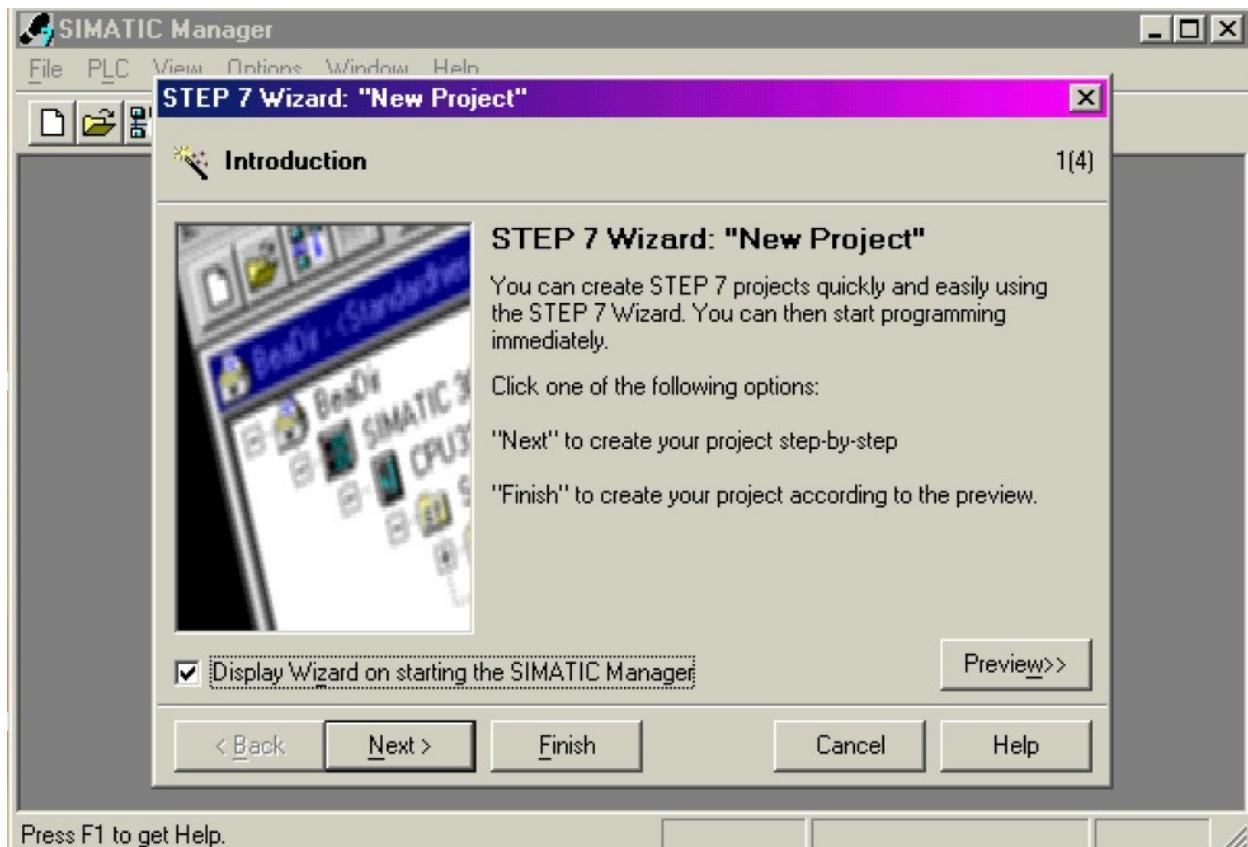
مجوز اصلی Step ۷ با کلید زردرنگ مشخص می شود .

#### ۲) مجوز اضطراری :

اگر به دلایلی مجوز اصلی از بین بود ، یه غیر فعال شود می توانید تا رفع مشکل مجوز اصلی از مجوز اضطراری استفاده کنید . این مجوز ۱۴ روزه است و با کلید قرمز رنگ مشخص می شود .

مجوزهای Step ۷ داخل یک فلاپی زردرنگ است و توسط برنامه ی Auto Mation Licence manager که در گروه Start Simatic از منوی قرار دارد قابل انتقال بر روی کامپیوتر می باشد . توجه کنید که در هنگام نصب این برنامه نیز نصب شود . برای نصب مجوز این برنامه را اجرا کرده ، در پنجره ی برنامه دو فهرست باز می شود که توسط آنها می توان ، درایو مبدا و مقصد را انتخاب کرد . در یکی از آنها درایو A را

انتخاب کنید ، بدین ترتیب مجوز های موجود در فلاپی نمایش داده خواهد شد . مجوز مورد نظرتان را انتخاب کرده و با فشردن دکمه های ->---<- به درایو مقصد منتقل نمایید



### محتويات نرم افزار ۷ : step

نرم افزار ۷ از برنامه ها و نرم افزارهای مختلفی تشکیل شده است که هر یک وظیفه‌ی خاصی را بر عهده دارند عبارتند از:

Hard ware configuration:-۱

از این برنامه جهت پیکربندی سخت افزار و تنظیم پارامترهای مازول ها استفاده می شود

PID Control Parameter Assignment-۲

از این برنامه جهت تنظیم پارامترهای بلوک های کنترل حلقه بسته استفاده می شود.

LAD , STL , FBD Programming S7 Blocks-۳

از این برنامه جهت برنامه نویسی بلوک های برنامه با یکی از زبان های STL , LAD , FBD استفاده میشود.

#### Netpro –Configuring Networks-۴

از این برنامه جهت پیکربندی شبکه های Profibus Ethernet , MPI صنعتی و همینطور برای ارتباط نقطه به نقطه استفاده می شود.

#### Symbol Editor-۵

از این برنامه جهت ایجاد و اصلاح سنبل ها استفاده می شود.

#### SIMATIC Manager-۶

از این برنامه که اصلی ترین برنامه نرم افزار ۷ step به شمار می رود جهت مدیریت پروژه های ۷ step استفاده می شود.

از طریق این برنامه می توانید به برنامه ها و ابزار ۷ step دسترسی پیدا کنید و قسمت های مختلف یک پروژه را تکمیل نمایید.

به کمک **somatic manager** می توانید:

۱-به صورت plc به صورت online دسترسی داشته باشد.

۲-پروژه ها و کتابخانه ها را مدیریت نمایید.

۳-برنامه نوشته شده را به plc و کارت حافظه ارسال کنید.

۴-به بلوک های برنامه ، ایستگاه های سخت افزاری و دیگر ابزارها دسترسی یابید.

#### آشنایی با :SIMATIC Manager

بعد از نصب نرم افزار ۷ step ، آیکن Desktop در صفحه ای Simatic Manager ظاهر می شود.می توانید

برای اجرای آن روی آیکن  کلیک نمایید و یا از منوی Start ، Simatic Manager ، Simatic انتخاب نمایید.

پس از اجرای آن یک پنجره باز می شود. دو روش ایجاد پروژه در آن وجود دارد:

۱-ایجاد پروژه با استفاده از Wizard

۲-ایجاد پروژه به صورت دستی

### (۱) ایجاد پروژه با استفاده از wizard

با اجرای Simatic Manager پنجره Wizard بطور خودکار باز می شود.

در پنجره مذکور روی دکمه Next کلیک کنید. در پنجره ای که ظاهر می شود Cpu و آدرس MPI مربوط به آن را انتخاب کنید. Next را می زنیم. در این پنجره به طور پیش فرض ob\_ انتخاب شده است.

پس از تعیین زبان، دکمه Next را می زنیم. در پنجره ای که باز شده نام پروژه را وارد کرده و روی دکمه کلیک می نماییم تا پروژه ایجاد شود. Finish

### ۲-ایجاد پروژه به صورت دستی

بعد از اجرای Simatic Manager در صورتی که پنجره Wizard به طور خودکار باز شده باشد، آن را میبندیم.

از منوی File ، New را انتخاب کرده ، در پنجره باز شده نام پروژه را در کادر وارد کرده ، سپس دکمه ok را فشار دهید.

بدنی ترتیب پنجره دو قسمتی پروژه ظاهر می شود. سپس روی صفحه ای راست، کلیک کرده و از قسمت Insert New Object را انتخاب ، سپس ۳۰۰ را انتخاب کرده و بعد روی Simatic کلیک کرده، در صفحه اصلی سخت افزار Hardware را معرفی کرده ، مشخصات plc ای که می خواهیم روی ریل بگذاریم باید دقیقاً معرفی کنیم.

از قسمت راست گزینه Simatic را می زنیم. گزینه ۳۰۰ Rack و بعد ریل را انتخاب می کنیم.

بعد از آن در قسمت اول قرار گرفته ، PC را انتخاب و به آن انتقال می دهیم.

روی ۳۰۰ PC کلیک کرده ، منبع تغذیه مورد نیاز که ۲ آمپری می طبشد را انتخاب می کنیم.

در قسمت دوم Cpu i ، Cpu ۳۱۳ را انتخاب می کنیم.

قسمت سوم جهت قرار دادن ماژول واسط است. در قسمت پنجره ای که در پایین آن باز شده، به قسمت روی آن دابل کلیک کرده تا آدرس را تغییر دهیم. در قسمت آدرس تیک سیستم را بر می داریم، می توانیم آدرس را تغییر دهیم.

برای کار با نرم افزار باید آدرس ورودی را از صفر تا ۲ قرار دهیم.

از قسمت File گزینه save and compare را زده، خود به خود save می کند.

و اگر به plc وصل باشیم، خودش را با آن مج می کند. تا مشاهده شود که آنچه را که معرفی کرده ایم، plc دارد یا نه.

سپس به صفحه‌ی سفید بر می گردیم، گزینه دیگری بنام C ۳۱۳ هم تشکیل شده است.

آن را تحریک کرده، از دو قسمت تشکیل شده S7 Program، سپس Blocks و ob را انتخاب می کنیم. در محیط برنامه نویسی، در قسمت View می توانیم زبان مورد نظرتان را انتخاب کنید. در قسمت چپ پنجره ای وجود دارد با چندین آیکن از جمله counter، F-F و Timer ها و ...

بعد از رسم برنامه‌ی مورد نظر اگر بخواهیم آن را به PLC دانلود کنیم، باید ابتدا PLC SimPLC را باز کرده، برای این کار به صفحه اصلی بازمی گردیم و آیکن را زده و پنجره‌ی باز می شود. در این پنجره با استفاده از Insert ← و QB IB سه گزینه وجود دارد. Runp، Run، Stop. به صفحه برنامه برگشته و برنامه را دانلود( ) می کنیم. برای دانلود از قسمت PLC دانلود را انتخاب کرده، دوباره به صفحه PLC Sim برگشته و برنامه را Runp می کنیم.

ورودی‌ها را تیک دار کرده و خروجی طبق برنامه تحریک می شود. سپس اگر بخواهیم نتیجه را روی صفحه برنامه نویسی مشاهده کنیم در صفحه برنامه نویسی آیکن را تحریک می کنیم. برای درست کردن Segment جدید کلیک راست کرده و Insert Network را می زنیم.

نکته: در قسمت FLAG ها SET، R-S، S-R مقدم است و در Reset مقدم است.

دانلود برنامه به حافظه PLC :

انتقال برنامه از PLC یا PG به حافظه دانلود گویند. این کار از طریق نرم افزار S7 و توسط یک واسطه به نام PC Adaptor انجام می شود.

به منظور دانلود برنامه بهتر است از وضعیت Stop استفاده شود. زیرا ممکن است در وضعیت Runp اگر بلوکی قبل از حافظه وجود داشته باشد و بخواهیم بلوک مشابه جدیدی را با پارامترهای متفاوت دانلود کنیم تداخل ایجاد شود و PLC به وضعیت Stop برود.

علاوه بر این بهتر است قبل از دانلود حافظه CPU باز نشانی شده باشد .

اگر در Simatic Manager ازمنوی PLC دانلود را انتخاب کنیم کل پروژه شامل پیکربندی سخت افزاری، پیکربندی بلوک ها و غیره بطور یکجا به حافظه PLC منتقل خواهد شد یاد آوری می شود که برنامه ابتدا به حافظه بارگذاری و سپس بخش اجرایی آن به حافظه کاری منتقل می شود.

شہزادہ

## فصل چهارم: پنوماتیک در PLC

شیرهای پنوماتیکی :

اجزای مختلف یک سیستم پنوماتیکی :

۱) منبع هوای فشرده

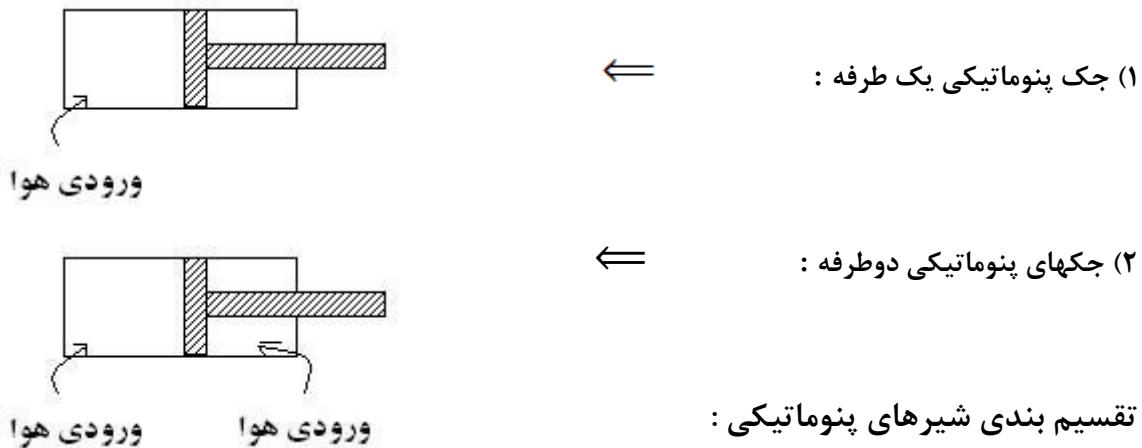
۲) قسمتهای انتقال

۳) شیرهای پنوماتیکی

۴) عملگرهای پنوماتیکی یا همان جکها

۵) سنسورها

جک های پنوماتیکی :



تقسیم بندی شیرهای پنوماتیکی :

۱) از بابت تعداد ورودیها و خروجی ها

۲) تعداد وضعیتهای کنترل

۳) نحوه تحریک

۴) ابعاد شیر

**نکته :** وضعیت، یعنی حالت باز یا بسته بودن فلش‌ها، جهت ورود و خروج فشار هوا را نشان می‌دهند.



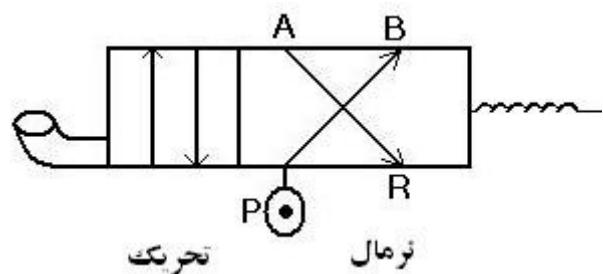
نمونه مدار ینوماتیک

### علامت اختصاری شیرها :

بدین منظور مستطیلی را در نظر می گیریم و آنرا به تعداد وضعیتهای خود تقسیم می کنیم  
در شیر ۴~~X~~<sup>۲</sup>: عدد ۴ نشان دهنده تعداد پورتهای ورودی و خروجی و ۲ نشان دهنده وضعیت های آن می باشد.

P نشان دهنده هوا است ، که از کن tact ہوا وارد می شود .

شیر دو خروجی A و B دارد که به سیلندر می رود . در حالت نرمال ، شیر در حالتی است که ، متر تاثیر می کند.



$$P \Rightarrow \begin{cases} A & \text{هو رسانی} \\ & \text{حریک} \end{cases}$$

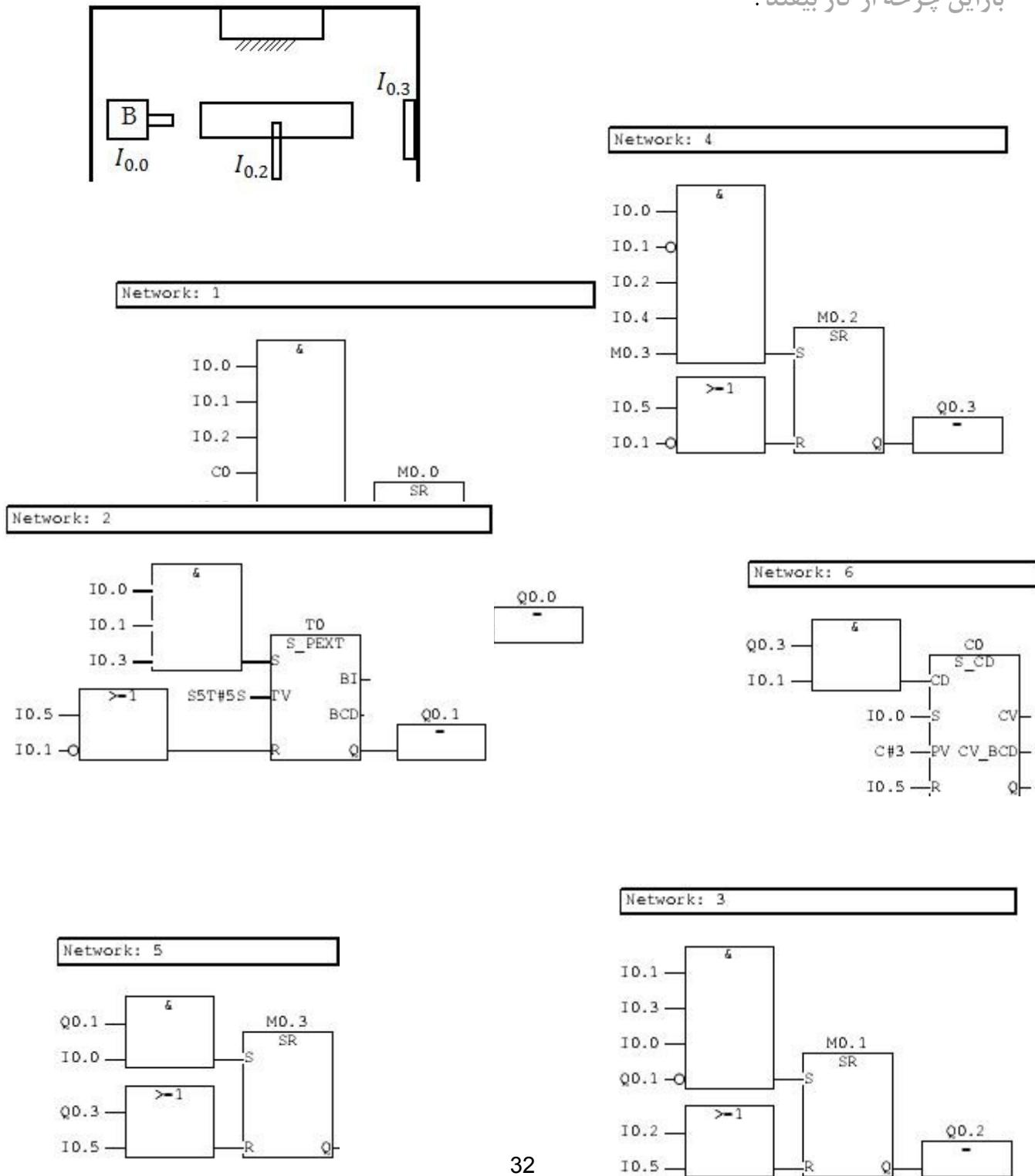
$$B \Rightarrow R \quad \text{تخلیه است}$$

$$\begin{cases} P \Rightarrow A & \text{هو رسانی} \\ & \text{تحریک} \end{cases}$$

$$A \Rightarrow R \quad \text{تخلیه است}$$

مثال: یک دستگاه رنگ پاش با ۴ سنسور را توسط PLC به صورت زیر کنترل می کنیم.

ابتدا با زدن Start و گذاشتن قطعه روی جک A، جک باز شده و سپس به مدت ۵ ثانیه رنگ پاشیده می شود. سپس جک A پایین آمده و قطعه توسط جک B به داخل جعبه انتقال می یابد و پس از ۶ بار این چرخه از کار بیفتد.





## فصل پنجم: معرفی سخت افزاری PLC S7-۳۰۰

### سخت افزار اصلی S7-۳۰۰ :

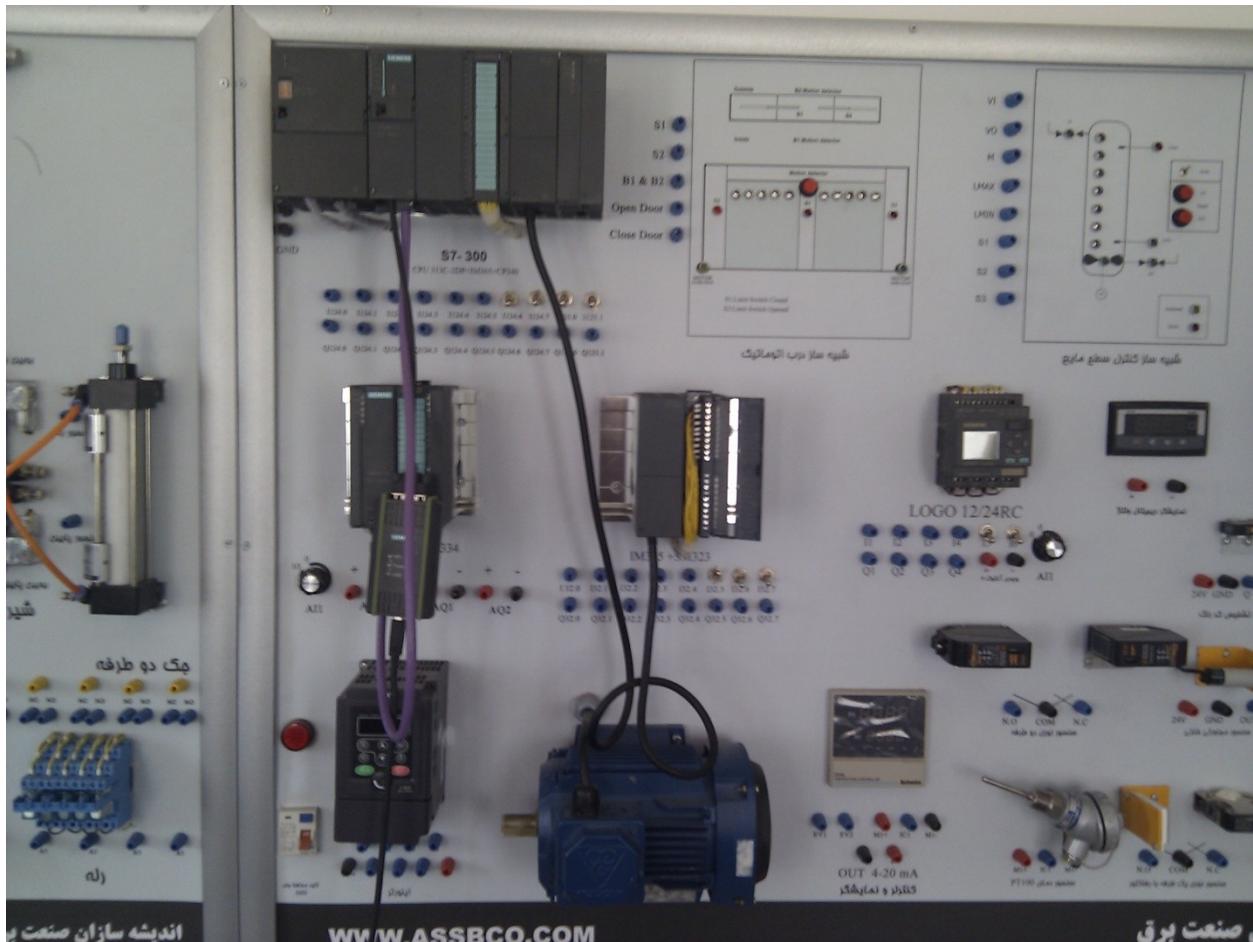
قبل از شروع پیکر بندی لازم است، در مرحله طراحی نوع و مشخصات سخت افزار مورد نیاز بررسی و تعیین شده باشد. بدین منظور برای اینکار به سوالات زیر باید پاسخ دهیم.

بررسی	نتیجه گیری
چه تعداد O/I داریم؟ توزیع فیزیکی سیگنالها چگونه است؟ آیا برای O/I ها به شبکه نیاز داریم؟ آیا قابلیت های خاص برای CPU مدنظر است؟ شرایط محیط نصب چگونه است؟ .....	CPU مشخص می گردد
چند نوع سیگنال آنالوگ وجود دارد؟ آیا برخی سیگنالها اهمیت بیشتری دارند؟ آیا کارت های O/I باید قابلیت خاصی داشته باشند؟ هر کارت چه تعداد ورودی و چه تعداد خروجی داشته باشد؟ .....	I/O کارت های
آیا کارت FM نیاز است؟ آیا کارت CP لازم است؟ .....	سایر مدول های مورد نیاز مشخص می شود.
تعداد کل مدول ها چقدر است؟ با احتساب اسلات رزرو آیا یک رک برای آنها کافی است؟ آیا لازم است نوع کارت های O/I را تغییر دهیم تا با گرفتن سیگنال بیشتر تعداد آنها کم شده و بتوان همه مدولها را روی یک رک جای داد؟ در صورت نیاز به رک اضافی چه تعداد رک و IM مورد نیاز است؟	رک اضافی مشخص می گردد.
جریان مصرفی مدولها بر اساس مشخصات فنی هر کدام از آنها چقدر است؟ چه منبع تغذیه ای برای تأمین جریان کل مدولها مناسب است؟	منبع تغذیه مشخص می گردد.

### پیکر بندی PLC از نوع S7-۳۰۰ :

## Rail یا Rack - ۱

محلى است که مازولهای S7-۳۰۰ روی آن قرار مى گيرند، نگهدارنده مدولها، تغذيه کننده مدولها و ايجاد ارتباط بين آنها مى باشد البته با وسائل جانبی.



ویژگی های رک : ۳۰۰ :

- ۱- يازده اسلات دارد
- ۲- به صورت رطلي است.
- ۳- فقط يك نوع دارد، که هم به عنوان رک اصلی و هم به عنوان رک اضافی استفاده مى گردد.
- ۴- فقط نقش نگهدارندگی برای مدول ها را دارد.

۵- ارتباط بین PS و Cpu با کانکتور خاصی برقرار می شود خود رک در ایجاد این ارتباط هیچ نقشی ندارد.

۶- ارتباط بین مدولها با کانکتور خود آن ها برقرار می شود.

۷- مدولها باید روی آن، کنار یکدیگر و بدون فاصله قرار گیرند.

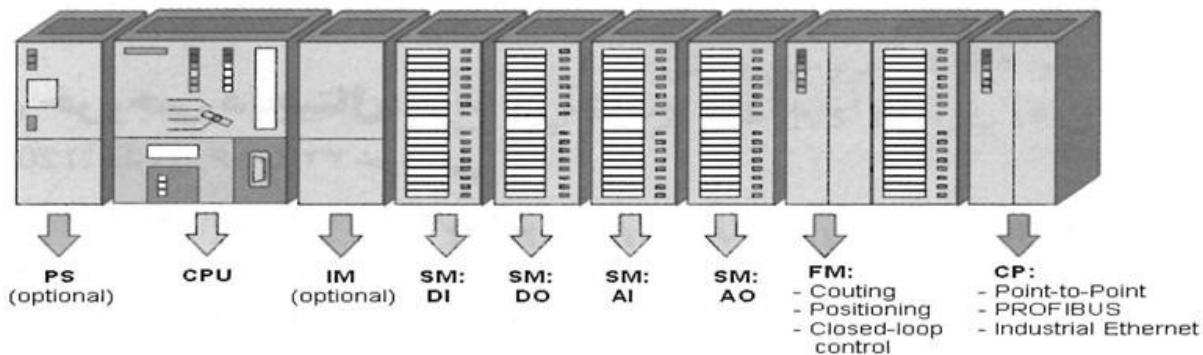
ترتیب مدولها در رک : ۳۰۰

روی رک نمی توان هر مدول را در اسلات دلخواهی قرار داد و ترتیب آن باید طبق جدول زیر رعایت شود.

مدولهای مجاز	شماره اسلات
Ps	۱
Cpu	۲
IM یا خالی	۳
FM - CP - SM	۱۱-۴

: S7-۳۰۰ بررسی مازول های

در شکل زیر مازول های تشکیل دهنده یک PLC سری S7-۳۰۰ نشان داده است .



۱- مازول منبع تغذیه (PS):

در جدول زیر انواع منابع تغذیه S7-۳۰۰ ذکر شده اند ، می توان به جای این منابع تغذیه هر نوع منبع تغذیه ۲۴ VDC دیگر هم استفاده کرد.

در واقع PS جریان مورد نیاز CPU و سایر مازول های PLC را تأمین می کند و ولتاژ آنها کاملاً ثبیت شده است.

## ۲- مژول CPU :

### سلکتور وضعیت :

این سلکتور ، به منظور وضعیت plc تعییه شده است، وضعیتهای قابل انتخاب عبارتند از:

MRES : در این وضعیت ، حافظه CPU بازنشانی می شود و CPU به وضعیت اولیه خود باز می گردد. بهتر است قبل از دانلود برنامه حافظه CPU بازنشانی شود.

✓ STOP : در این وضعیت ، برنامه اجرا نمی شود ، و دسترسی به I/O ها امکان پذیر نیست .

✓ RUN : در این وضعیت ، برنامه کاربر اجرا می شود و تمام ورودی و خروجی ها به روز می شوند. در این وضعیت فقط امکان خواندن از CPU و به عبارتی Upload وجود دارد.

✓ RUN-P : در این وضعیت ، برنامه کاربر اجرا می شود ، علاوه بر خواندن ، امکان نوشتمن و به عبارت دیگر Download نیز وجود دارد.

### توصیف LEDها :

SF : اگر خطای سیستم و یا خطای داخلی رخ دهد این LED روشن خواهد شد.

BATF : اگر باتری پشتیبان نصب نشده باشد یا اینکه خالی باشد این LED روشن خواهد شد.

DCAV : روشن بودن این LED نشان دهنده این است که ولتاژ ۵V مستقیم داخلی برقرار است.

FRCE : اگر در برنامه حداقل یک ورودی یا یک خروجی FORCE شده باشد ، این LED روشن خواهد شد.

RUN : اگر CPU در وضعیت RUN باشد این LED روشن خواهد شد.

STOP : اگر CPU در وضعیت STOP باشد این LED روشن خواهد شد ، همچنین اگر به دلایلی حافظه نیاز به بازنشانی داشته باشد این LED چشمک می زند.



PLC S7-300

سُبْرَة

## S7-۳۰۰ آماده سازی فصل ششم:

### محیط Hwconfig :

در PLC های سری S7-۳۰۰ ، سخت افزار PLC باید در محیط نرم افزاری پیکر بندی شود.

ابتدا در SIMATIC MANAGER یک پروژه جدید ایجاد کنید، سپس از منوی Insert ، ایستگاه (station) را انتخاب کنید.

با انتخاب آیکن آن به پنجره سمت چپ پروژه افزوده می شود. و به طور خودکار نام (۱) ۳۰۰ SIMATIC برای آن در نظر گرفته می شود، با کلیک روی آیکن Hardware در پنجره سمت راست پروژه ظاهر می شود. با دوبار کلیک روی آن برنامه دیگری به نام Hwconfig اجرا می شود. از این ابزار برای پیکر بندی سخت افزاری استفاده می شود. نوار عنوان پنجره باز شو شامل نام پروژه و ایستگاه انتخاب شده است. در سمت راست این پنجره فهرست کلیه قطعات سخت افزاری در پنجره ای با نام Hardware catalog مشاهده می شود ، اگر این پنجره وجود نداشت توسط منوی View >> catalog آن را فعال کنید. با انتخاب پروفایل Standard فهرست کلیه قطعات سخت افزاری نمایش داده می شود.

می توان یک پروفایل سفارشی ، شامل المانهایی مورد نیاز ایجاد نمود. بدین منظور از منوی option>>Edit catalog Profiles پایینی و ذخیره کردن آن به فهرست پروفایل ها اضافه می شود

### شروع پیکر بندی سخت افزاری (Hardware configuration ) :

فرض کنید ، می خواهیم سخت افزار یک PLC سری S7-۳۰۰ را پیکر بندی کنیم. در پنجره catalog ، روی آیکن SIMATIC ۳۰۰ کلیک می کنیم ، تا فهرست المانهای مربوط به S7-۳۰۰ نمایش داده شود. در پیکر بندی سخت افزار، Rack، اولین المانی است که باید انتخاب شود در پنجره catalog ، روی پوشه Rack-۳۰۰ کلیک نمایید تا فهرست رک های موجود نمایش داده شود ، در S7-۳۰۰ فقط یک نوع ریل داریم. با دوبار کلیک روی Rail وارد پنجره Hwconfig می شود. بعد از آن نوبت انتخاب مژوی ها و قرار دادن آنها در اسلات مناسب و صحیح است. منبع تغذیه ، در پنجره catalog ، روی پوشه PS-۳۰۰ کلیک نمایید ، فهرست کلیه منابع مربوط به S7-۳۰۰ ظاهر

می شوند ، منبع تغذیه مورد نظر را انتخاب و آن را وارد اسلات شماره ۱ می کنیم. CPU مورد نظر را از کاتالوگ انتخاب کنید و در اسلات شماره ۲ قرار دهید.

نکته: البته اگر روش Wizard پروژه را باز کرده باشید ، در ابتدا CPU مورد نظر انتخاب می شود.

اسلات شماره ۳ ، جهت قرار دادن ماژول واسطه رزرو شده است و نمی توان ماژول دیگری را در این اسلات قرار داد. به همین ترتیب ماژول های را از کاتالوگ انتخاب و در اسلات صحیح قرار دهید.

### بررسی سازگاری اجزاء (Consistency check) :

برنامه Hwconfig ، به طور خودکار در طول انجام عملیات پیکر بندی ، سازگاری اجزاء با یکدیگر را بررسی می کند در صورت لزوم خطاها را اعلام می کند ، با وجود این بهتر است پس از اتمام عملیات پیکر بندی سخت افزاری با استفاده از منوی Station >> Consistency check سازگاری اجزاء با یکدیگر را بازبینی نمایید.

### ذخیره پیکر بندی سخت افزاری:

جهت ذخیره کردن پیکربندی سخت افزاری ، دو گزینه در منوی Station قرار دارند که عبارتند :

✓ Save : با انتخاب این گزینه ، پیکربندی سخت افزاری ذخیره می شود ، بدون اینکه بلوکهای داده سیستم ایجاد شود.

✓ Save & Compile : با انتخاب این گزینه ، علاوه بر ذخیره پیکر بندی ، بلوکهای داده سیستم ایجاد و تحت نام System data ذخیره می شوند. بلوکهای داده سیستم یا SDB ها حاوی اطلاعات پیکربندی و پارامترهای هر ماپزول می باشند. بعد از Download ، این بلوکها در بخش حافظه کاری ، حافظه CPU ذخیره می شوند. اگر CPU دارای کارت حافظه ای از نوع Flash EPROM باشد ، با ذخیره SDB ها در آن ، در صورت قطع تغذیه و نبود باتری پشتیبان پیکر بندی از بین نخواهد رفت.

### CPU به سخت افزار نمودن پیکر بندی (Download) :

برای سلکتور plc Stop را در حالت Hwconfig می خواهیم . در محیط Hwconfig از منوی گزینه Download را انتخاب کنید. بدین ترتیب ، عملیات پیکر بندی سخت افزاری به پایان می رسد.

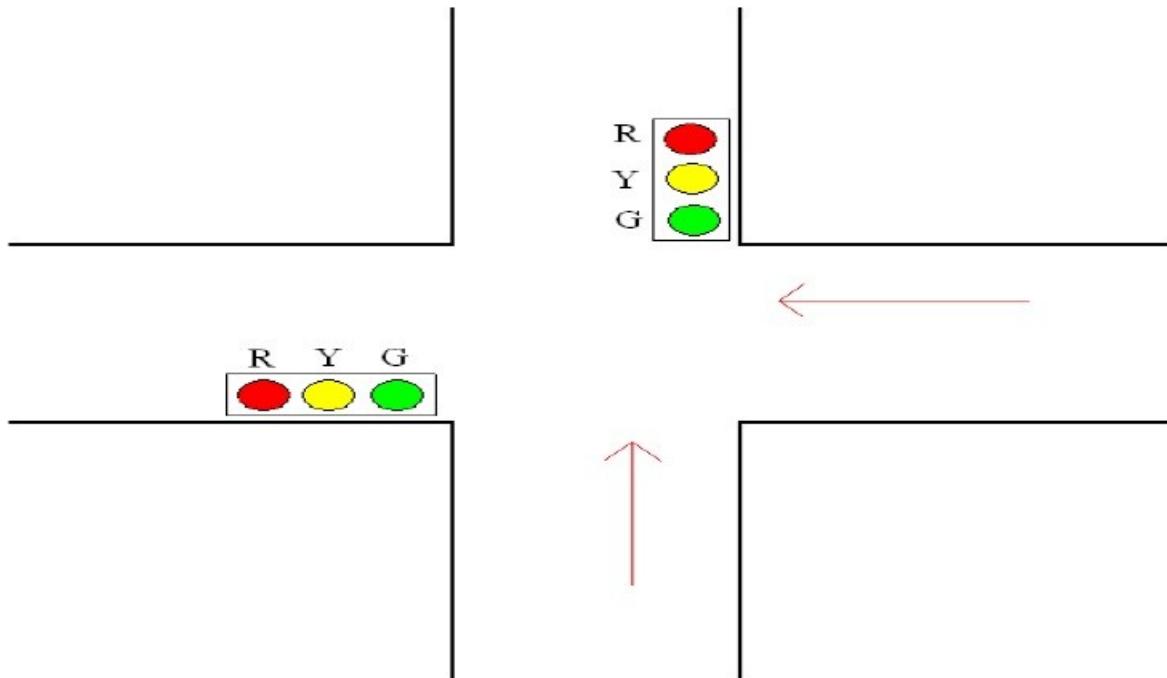
برای خواندن پیکربندی سخت افزاری از منوی گزینه Upload را انتخاب می کنیم.

مکالمہ

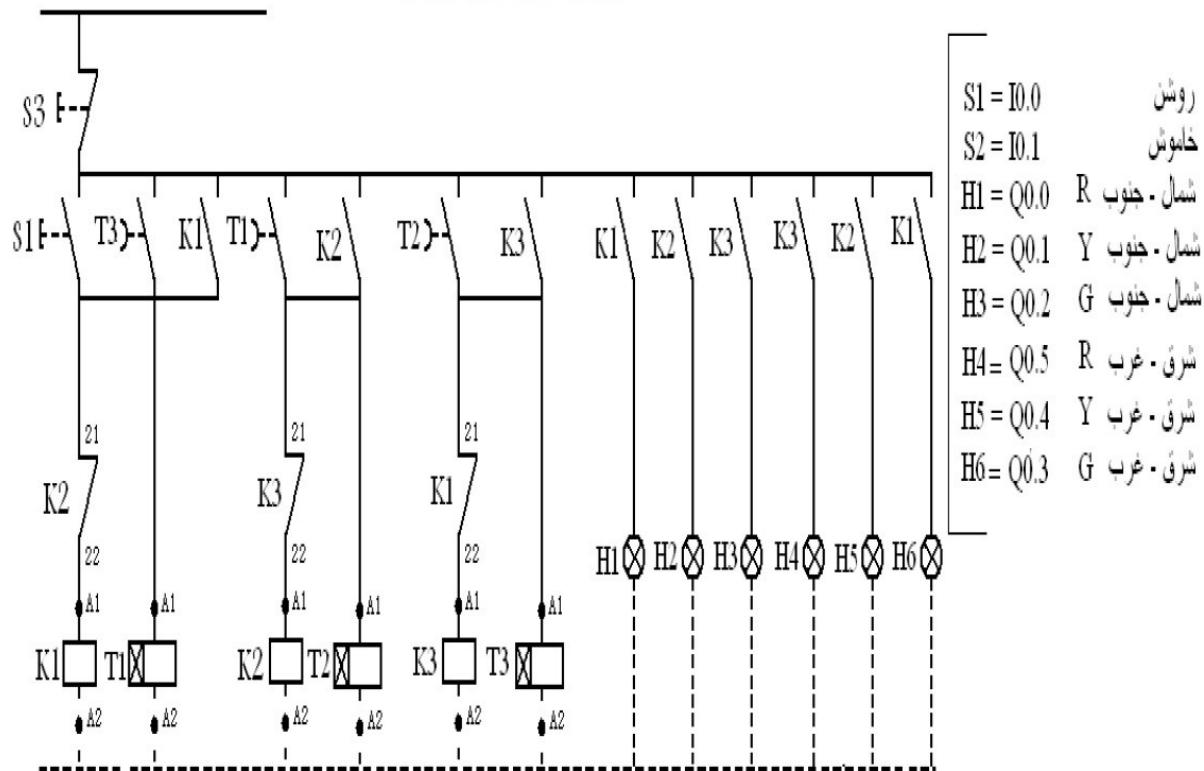
## فصل هفتم: مثال های کاربردی و صنعتی

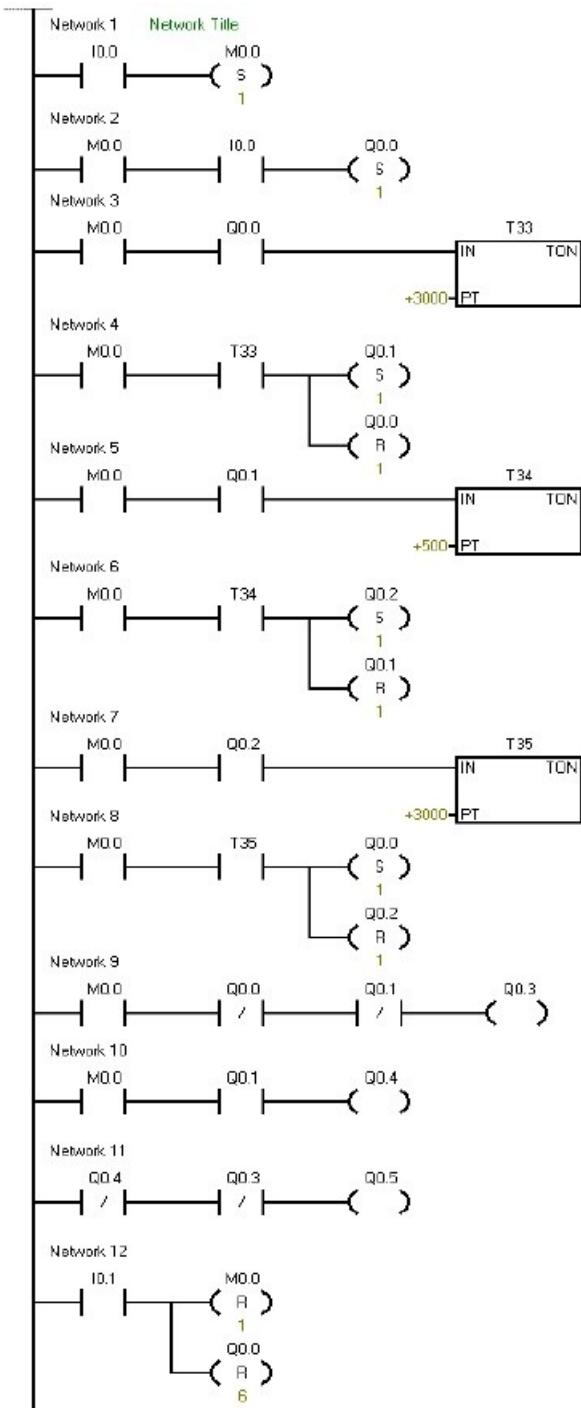
مثال ۱: در یک چهار راه سیستم چراغ راهنمایی به صورت زیر است:

مدت زمان چراغ قرمز ۳۰ ثانیه و مدت زمان چراغ زرد ۵ ثانیه و مدت زمان چراغ سبز ۳۰ ثانیه می باشد  
سیستم کنترل آن را طراحی کنید.



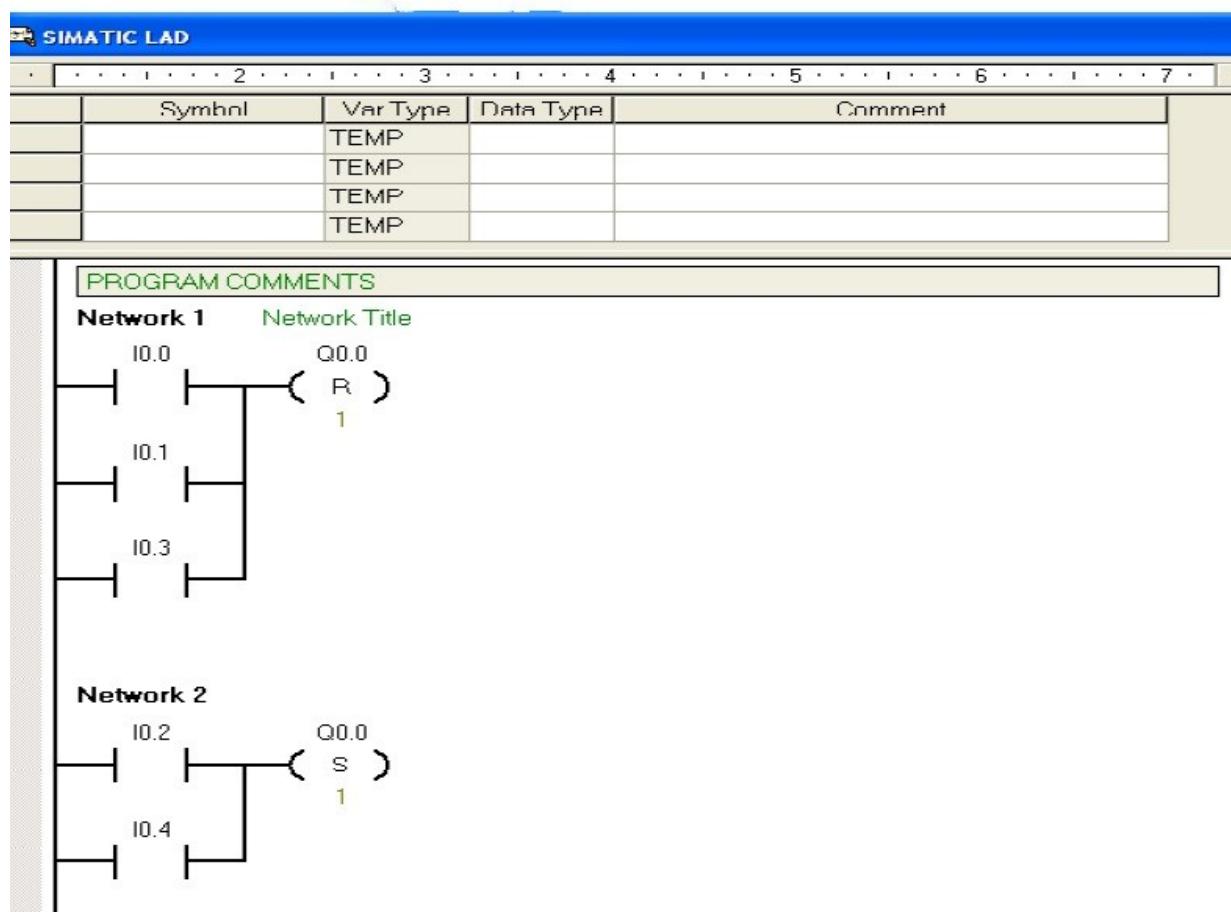
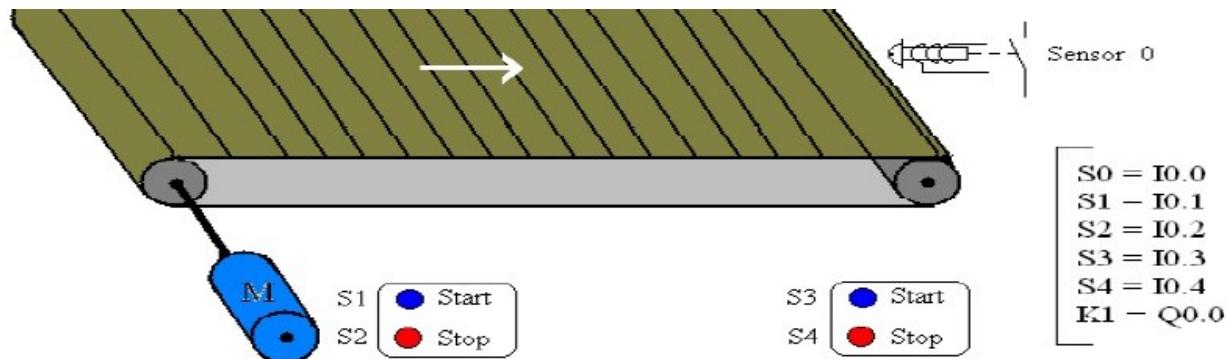
ابتدا سیستم رله ای آن را طراحی می کنیم:





برنامه آن به صورت رو به رو خواهد بود:

مثال ۲: در شکل زیر دو کلید فشاری S1 و S2 به ترتیب برای start و stop در سمت آغازین نوار نقاله وجود دارد همچنین در بخش انتهایی آن دو کلید S3 و S4 برای start و stop نوار تعییه شده است از طریق هر دو بخش آغازین و انتهایی می‌توان آن را start و stop نمود لازم به ذکر است که سنسور S0 برای توقف نوار به هنگام رسیدن جسم به انتهای نوار نقاله نصب شده است برنامه کنترلی آن را بنویسید.





## ضمیمه ۱: لیست دستورات STL در ۳۰۰ و ۴۰۰

English Mnemonics	Program Elements Catalog	Description
+	Integer math instruction	Add Integer Constant (16, 32-Bit)
=	Bit logic Instruction	Assign
)	Bit logic Instruction	Nesting Closed
+AR1	Accumulator	AR1 Add ACCU 1 to Address Register 1
+AR2	Accumulator	AR2 Add ACCU 1 to Address Register 2
+D	Integer math instruction	Add ACCU 1 and ACCU 2 as Double Integer (32-Bit)
*D	Integer math instruction	Multiply ACCU 1 and ACCU 2 as Double Integer (32-Bit)
/D	Integer math instruction	Divide ACCU 2 by ACCU 1 as Double Integer(32-Bit)
?D	Integer math instruction	Compare Double Integer (32-Bit) ==, <>, >, <, >=, <=
?I	Compare	Add ACCU 1 and ACCU 2 as Integer (16-Bit)
-I	Integer math instruction	Subtract ACCU 1 from ACCU 2 as Integer (16-Bit)
*I	Integer math instruction	Multiply ACCU 1 and ACCU 2 as Integer (16-Bit)
/I	Integer math instruction	Divide ACCU 2 by ACCU 1 as Integer (16-Bit)
?I	Integer math instruction	Compare Integer (16-Bit) ==, <>, >, <, >=, <=
?F	Compare	Add ACCU 1 and ACCU 2 as a Floating-Point Number (32-Bit IEEE-FP)
+R	Floating point Instruction	Subtract ACCU 1 from ACCU 2 as a Floating-Point Number (32-Bit IEEE-FP)
*R	Floating point Instruction	Multiply ACCU 1 and ACCU 2 as Floating-Point Number (32-Bit IEEE-FP)
/R	Floating point Instruction	Divide ACCU 2 by ACCU 1 as a Floating-Point Number (32-Bit IEEE-FP)
?R	Floating point Instruction	Compare Floating-Point Number (32-Bit) ==, <>, >, <, >=, <=
A	Bit logic Instruction	And
A(	Bit logic Instruction	And with Nesting Open
ABS	Floating point Instruction	Absolute Value of a Floating-Point Number (32-Bit IEEE-FP)
ACOS	Floating point Instruction	Generate the Arc Cosine of a Floating-Point Number (32-Bit)
AD	Word logic Instruction	AND Double Word (32-Bit)
AN	Bit logic Instruction	And Not
AN(	Bit logic Instruction	And Not with Nesting Open
ASIN	Floating point Instruction	Generate the Arc Sine of a Floating-Point Number (32-Bit)
ATAN	Floating point Instruction	Generate the Arc Tangent of a Floating-Point Number (32-Bit)
AW	Word logic Instruction	AND Word (16-Bit)
BE	Program control	Block End
BEC	Program control	Block End Conditional
BEU	Program control	Block End Unconditional
BLD	Program control	Program Display Instruction (Null)
BTD	Convert	BCD to Integer (32-Bit)
BTI	Convert	BCD to Integer (16-Bit)
CAD	Convert	Change Byte Sequence in ACCU 1 (32-Bit)
CALL	Program control	Block Call
CALL	Program control	Call Multiple Instance

English Mnemonics	Program Elements Catalog	Description
CALL	Program control	Call Block from a Library
CAR	Load/Transfer	Exchange Address Register 1 with Address Register 2
CAW	Convert	Change Byte Sequence in ACCU 1-L (16-Bit)
CC	Program control	Conditional Call
CD	Counters	Counter Down
CDB	Convert	Exchange Shared DB and Instance DB
CLR	Bit logic Instruction	Clear RLO (=0)
COS	Floating point Instruction	Generate the Cosine of Angles as Floating-Point Numbers (32-Bit)
CU	Counters	Counter Up
DEC	Accumulator	Decrement ACCU 1-L-L
DTB	Convert	Double Integer (32-Bit) to BCD
DTR	Convert	Convert Double Integer (32-Bit) to Floating-Point (32-Bit IEEE-FP)
ENT	Accumulator	Enter ACCU Stack
EXP	Floating point Instruction	Generate the Exponential Value of a Floating-Point Number (32-Bit)
FN	Bit logic Instruction	Edge Negative
FP	Bit logic Instruction	Edge Positive
FR	Counters	Enable Counter (Free) (free, FR C 0 to C 255)
FR	Timers	Enable Timer (Free)
INC	Accumulator	Increment ACCU 1-L-L
INVD	Convert	Ones Complement Double Integer (32-Bit)
INVI	Convert	Ones Complement Integer (16-Bit)
ITB	Convert	Integer (16-Bit) to BCD
ITD	Convert	Integer (16-Bit) to Double Integer (32-Bit)
JBI	Jumps	Jump if BR = 1
JC	Jumps	Jump if RLO = 1
JCB	Jumps	JCB SPBB Jumps Jump if RLO = 1 with BR
JCN	Jumps	Jump if RLO = 0
JL	Jumps	Jumps Jump to Labels
JM	Jumps	Jump if Minus
JMZ	Jumps	Jump if Minus or Zero
JN	Jumps	Jump if Not Zero
JNB	Jumps	Jump if RLO = 0 with BR
JNBI	Jumps	Jump if BR = 0
JO	Jumps	Jump if OV = 1
JOS	Jumps	Jump if OS = 1
JP	Jumps	Jump if Plus
JPZ	Jumps	Jump if Plus or Zero
JU	Jumps	Jump Unconditional
JUO	Jumps	Jump if Unordered
JZ	Jumps	Jump if Zero
L	Load/Transfer	Load
L DBLG	Load/Transfer	Load Length of Shared DB in ACCU 1
L DBNO	Load/Transfer	Load Number of Shared DB in ACCU 1

English Mnemonics	Program Elements Catalog	Description
L.DILG	Load/Transfer	Load Length of Instance DB in ACCU 1
L.DINO	Load/Transfer	Load Number of Instance DB in ACCU 1
L.STW	Load/Transfer	Load Status Word into ACCU 1
L.	Timers	Load Current Timer Value into ACCU 1 as Integer (the current timer value can be a number from 0 to 255, for example, L.T 32)
L.	Counters	Load Current Counter Value into ACCU 1 (the current counter value can be a number from 0 to 255, for example, L.C 15)
LAR1	Load/Transfer	Load Address Register 1 from ACCU 1
LAR1 <D>	Load/Transfer	Load Address Register 1 with Double Integer (32-Bit Pointer)
LAR1 AR2	Load/Transfer	Load Address Register 1 from Address Register 2
LAR2	Load/Transfer	Load Address Register 2 from ACCU 1
LAR2 <D>	Load/Transfer	Load Address Register 2 with Double Integer (32-Bit Pointer)
LC	Counters	Load Current Counter Value into ACCU 1 as BCD (the current timer value can be a number from 0 to 255, for example, LC.C 15)
LC	Timers	Load Current Timer Value into ACCU 1 as BCD (the current counter value can be a number from 0 to 255, for example, LC.T 32)
LEAVE	Accumulator	Leave ACCU Stack
LN	Floating point Instruction	Generate the Natural Logarithm of a Floating-Point Number (32-Bit)
LOOP	Jumps	Loop
MCR(	Program control	Save RLO in MCR Stack, Begin MCR
)MCR	Program control	End MCR
MCRA	Program control	Activate MCR Area
MCRD	Program control	Deactivate MCR Area
MOD	Integer math instruction	Division Remainder Double Integer (32-Bit)
NEGD	Convert	Twos Complement Double Integer (32-Bit)
NEGI	Convert	Twos Complement Integer (16-Bit)
NEGR	Convert	Negate Floating-Point Number (32-Bit, IEEE-FP)
NOP 0	Accumulator	Null Instruction
NOP 1	Accumulator	Null Instruction
NOT	Bit logic Instruction	Negate RLO
O	Bit logic Instruction	Or
O(	Bit logic Instruction	Or with Nesting Open
OD	Bit logic Instruction	OR Double Word (32-Bit)
ON	Bit logic Instruction	Or Not
ON(	Bit logic Instruction	Or Not with Nesting Open
OPN	DB call	Open a Data Block
OW	Word logic Instruction	OR Word (16-Bit)
POP	Accumulator	CPU with Two ACCUs
POP	Accumulator	CPU with Four ACCUs
POP	Accumulator	CPU with Two ACCUs
PUSH	Accumulator	CPU with Four ACCUs
R	Bit logic Instruction	Reset
R	Counters	Reset Counter (the current counter can be a number from 0 to 255, for example, R.C 15)
R	Timers	Reset Timer (the current timer can be a number from 0 to 255, for example, R.T 32)

English Mnemonics	Program Elements Catalog	Description
RLD	Shift/Rotate	Rotate Left Double Word (32-Bit)
RLDA	Shift/Rotate	Rotate ACCU 1 Left via CC 1 (32-Bit)
RND	Convert	Round
RND-	Convert	Round to Lower Double Integer
RND+	Convert	Round to Upper Double Integer
RRD	Shift/Rotate	Rotate Right Double Word (32-Bit)
RRDA	Shift/Rotate	Rotate ACCU 1 Right via CC 1 (32-Bit)
S	Bit logic Instruction	Set
S	Counters	Set Counter Preset Value (the current counter can be a number from 0 to 255, for example, S C 15)
SAVE	Bit logic Instruction	Save RLC in BR Register
SD	Timers	On-Delay Timer
SE	Timers	Extended Pulse Timer
SET	Bit logic Instruction	Set
SF	Timers	Off-Delay Timer
SIN	Floating point Instruction	Generate the Sine of Angles as Floating-Point Numbers (32-Bit)
SLD	Shift/Rotate	Shift Left Double Word (32-Bit)
SLW	Shift/Rotate	Shift Left Word (16-Bit)
SP	Timers	Pulse Timer
SQR	Floating point Instruction	Generate the Square of a Floating-Point Number (32-Bit)
SQRT	Floating point Instruction	Generate the Square Root of a Floating-Point Number (32-Bit)
SRD	Shift/Rotate	Shift Right Double Word (32-Bit)
SRW	Shift/Rotate	Shift Right Word (16-Bit)
SS	Timers	Retentive On-Delay Timer
SSD	Shift/Rotate	Shift Sign Double Integer (32-Bit)
SSI	Shift/Rotate	Shift Sign Integer (16-Bit)
T	Load/Transfer	Transfer
T	Load/Transfer	Transfer ACCU 1 into Status Word
TAK	Accumulator	Toggle ACCU 1 with ACCU 2
TAN	Floating point Instruction	Generate the Tangent of Angles as Floating-Point Numbers (32-Bit)
TARI	Load/Transfer	Transfer Address Register 1 to ACCU 1
TARI	Load/Transfer	Transfer Address Register 1 to Destination (32-Bit Pointer)
TARI	Load/Transfer	Transfer Address Register 1 to Address Register 2
TAR2	Load/Transfer	Transfer Address Register 2 to ACCU 1
TAR2	Load/Transfer	Transfer Address Register 2 to Destination (32-Bit Pointer)
TRUNC	Convert	Truncate
UC	Program control	Unconditional Call
X	Bit logic Instruction	Exclusive Or
X(	Bit logic Instruction	Exclusive Or with Nesting Open
XN	Bit logic Instruction	Exclusive Or Not
XN(	Bit logic Instruction	Exclusive Or Not with Nesting Open
XOD	Word logic Instruction	Exclusive OR Double Word (32-Bit)
XOW	Word logic Instruction	Exclusive OR Word (16-Bit)

## ضمیمه ۲: فهرست کدهای ماشین در PLC های زیمنس

**Machine Code Listing**

Machine Code								Operation	Oper- and		
B0		B1		B2		B3					
L	R	L	R	L	R	L	R				
0	0	0	0					HOPG			
0	1	0	0					CPW			
0	2	0 <sub>1</sub>	0 <sub>2</sub>					L	T		
0	3	0 <sub>1</sub>	0 <sub>2</sub>					TNS			
0	4	0 <sub>1</sub>	0 <sub>2</sub>					PR	T		
0	5	0	0					BEC			
0	6	0 <sub>1</sub>	0 <sub>2</sub>					PR =			
0	7	0 <sub>1</sub>	0 <sub>2</sub>					A =			
0	8	0	0					IA			
0	9	0	0					IAA			
0	A	0 <sub>1</sub>	0 <sub>2</sub>					CSW			
0	B	0 <sub>1</sub>	0 <sub>2</sub>					L	FB		
0	C	0 <sub>1</sub>	0 <sub>2</sub>					T	FB		
0	D	0 <sub>1</sub>	0 <sub>2</sub>					LC	T		
0	E	0 <sub>1</sub>	0 <sub>2</sub>					JO =			
0	F	0 <sub>1</sub>	0 <sub>2</sub>					LC =			
1	0	0	2					G			
1	0	0	3					BLD	130		
1	0	0	4					BLD	131		
1	0	0	5					BLD	132		
1	0	0	6					BLD	133		
1	0	0	7					BLD	255		
1	1	0 <sub>1</sub>	0 <sub>2</sub>					I			
1	2	0 <sub>1</sub>	0 <sub>2</sub>					L	FW		
1	3	0 <sub>1</sub>	0 <sub>2</sub>					T	FW		
1	4	0 <sub>1</sub>	0 <sub>2</sub>					SF	T		
1	5	0 <sub>1</sub>	0 <sub>2</sub>					EP =			
1	6	0 <sub>1</sub>	0 <sub>2</sub>					SFC =			
1	7	0 <sub>1</sub>	0 <sub>2</sub>					S =			
1	8	0 <sub>1</sub>	0 <sub>2</sub>					D			
1	C	0 <sub>1</sub>	0 <sub>2</sub>					SE	T		
1	D	0 <sub>1</sub>	0 <sub>2</sub>					SC	FB		

Machine Code								Operation	Oper- and		
B0		B1		B2		B3					
L	R	L	R	L	R	L	R				
1	E	0 <sub>1</sub>	0 <sub>2</sub>					SEC =			
1	F	0 <sub>1</sub>	0 <sub>2</sub>					= =			
2	0	0 <sub>1</sub>	0 <sub>2</sub>					C	D8		
2	1	2	0					DEP			
2	1	4	0					ACP			
2	1	6	0					DE < F			
2	1	8	0					E = F			
2	1	A	0					DE = F			
2	1	C	0					AC = F			
2	2	0 <sub>1</sub>	0 <sub>2</sub>					L	DIL		
2	3	0 <sub>1</sub>	0 <sub>2</sub>					T	DIL		
2	4	0 <sub>1</sub>	0 <sub>2</sub>					SH	T		
2	5	0 <sub>1</sub>	0 <sub>2</sub>					SH =			
2	6	0 <sub>1</sub>	0 <sub>2</sub>					SHH =			
2	7	0 <sub>1</sub>	0 <sub>2</sub>					SHHH =			
2	8	0 <sub>1</sub>	0 <sub>2</sub>					L	KB		
2	A	0 <sub>1</sub>	0 <sub>2</sub>					L	DR		
2	B	0 <sub>1</sub>	0 <sub>2</sub>					T	DR		
2	C	0 <sub>1</sub>	0 <sub>2</sub>					SC	T		
2	D	0 <sub>1</sub>	0 <sub>2</sub>					SP =			
2	E	0 <sub>1</sub>	0 <sub>2</sub>					SSU =			
2	F	0 <sub>1</sub>	0 <sub>2</sub>					OM =			
3	0	0	1	0 <sub>1</sub>	0 <sub>2</sub>	0 <sub>1</sub>	0 <sub>2</sub>	L	RC		
3	0	0	2	0 <sub>1</sub>	0 <sub>2</sub>	0 <sub>1</sub>	0 <sub>2</sub>	L	RT		
3	0	0	4	0 <sub>1</sub>	0 <sub>2</sub>	0 <sub>1</sub>	0 <sub>2</sub>	L	RF		
3	0	1	0	0 <sub>1</sub>	0 <sub>2</sub>	0 <sub>1</sub>	0 <sub>2</sub>	L	RS		
3	0	2	0	0 <sub>1</sub>	0 <sub>2</sub>	0 <sub>1</sub>	0 <sub>2</sub>	L	RY		
3	0	4	0	0 <sub>1</sub>	0 <sub>2</sub>	0 <sub>1</sub>	0 <sub>2</sub>	L	RH		
3	0	8	0	0 <sub>1</sub>	0 <sub>2</sub>	0 <sub>1</sub>	0 <sub>2</sub>	L	RW		
3	2	0 <sub>1</sub>	0 <sub>2</sub>					L	DW		
3	3	0 <sub>1</sub>	0 <sub>2</sub>					T	DW		
3	4	0 <sub>1</sub>	0 <sub>2</sub>					SP	T		



Machine Code								Operation	Operand
B0	B1	B2	B3						
L	R	L	R	L	R	L	R		
7	2	0 <sub>1</sub>	0 <sub>1</sub>			L		PB/PY*	
7	3	0 <sub>1</sub>	0 <sub>1</sub>			T		PB/PY*	
7	5	0 <sub>1</sub>	0 <sub>1</sub>			IU		PS	
7	6	0 <sub>1</sub>	0 <sub>1</sub>			DO=			
7	8	0	5	0	0	0 <sub>1</sub>	0 <sub>1</sub>	G	09
7	9	0	0			+F			
7	A	0 <sub>1</sub>	0 <sub>1</sub>			L		PW	
7	B	0 <sub>1</sub>	0 <sub>1</sub>			T		PW	
7	C	0 <sub>1</sub>	0 <sub>1</sub>			R		C	
7	D	0 <sub>1</sub>	0 <sub>1</sub>			IU		SB	
7	E	0	0			DI			
8	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>			A		F	
8	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>			O		F	
9	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>			S		F	
9	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>			=		F	
A	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>			AH		F	
A	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>			ON		F	
B	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>			R		F	
B	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>			A		C	
B	9	0 <sub>1</sub>	0 <sub>1</sub>			O		C	
B	A	0	0			A)			
B	B	0	0			DX			
B	C	0 <sub>1</sub>	0 <sub>1</sub>			AH		C	

Machine Code								Operation	Operand
B0	B1	B2	B3						
L	R	L	R	L	R	L	R		
B	0	0 <sub>1</sub>	0 <sub>1</sub>					ON	C
B	1	0	0					J	
C	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					A	I
C	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					A	Q
C	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					O	I
C	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					O	Q
D	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					S	I
D	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					S	Q
D	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					=	I
D	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					=	Q
E	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					AH	I
E	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					AH	Q
E	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					ON	I
E	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					ON	Q
F	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					R	I
F	0 <sub>1</sub>	0 <sub>1</sub>	0 <sub>1</sub>					R	Q
F	0	0 <sub>1</sub>	0 <sub>1</sub>					A	T
F	9	0 <sub>1</sub>	0 <sub>1</sub>					O	T
F	A	0 <sub>1</sub>	0 <sub>1</sub>					JC=	
F	B	0	0					O	
F	C	0 <sub>1</sub>	0 <sub>1</sub>					AH	T
F	D	0 <sub>1</sub>	0 <sub>1</sub>					ON	T
F	F	F	F					NOP 1	

\* PY in the case of 55-DOS programmers

#### Explanation of the Indices

a + byte address  
 b + bit address  
 c + parameter address  
 d + timer number  
 e + constant  
 f + block number  
 g + word address

h + number of shifts  
 i + relative jump address  
 k + register address  
 l + block length in bytes  
 m + jump displacement (16 bit)  
 n + value  
 o + counter number

### ضمیمه ۳: فهرست خطاهای موجود در ISTACK

Mnemonics for Control Bits and Interrupt Display	
Control Bit Mnemonics	Mnemonics for Cause of Error/Fault (= Error ID)
BSTSCH	Block shift requested
SCHTAE	Block shift active (function: KOMPAG)
ADRBAAU	Construction of address lists
SPABER	Compress operation aborted
CA-DA	Interprocessor communication flag output address list available
CE-DE	Interprocessor communication flag input address list available
REMAN	0: all timers, counters, and flags are reset on Cold Restart 1: the second half of timers, counters, and flags are reset on Cold Restart
STOPUS	"STOP" state (external request for example via the programmer)
STOPANZ	"STOP" display
NEUSTA	PC in Cold Restart
BATPLUF	Battery backup okay
BARE	Program check
BARBEND	Request for end of program check
UAFEHL	Incorrect interrupt display
AF	Interrupt enable
ASPNEP	Memory submodule is an EEPROM
ASPNR	Memory submodule is a RAM
KOPPNI	Block header cannot be interpreted
ASPNEEP	Memory submodule is an EEPROM
KENAS	No memory submodule
SYNPER	Synchronization error (blocks are incorrect)
NINEU	Cold Restart not possible
URLAD	Bootstrapping required
Other mnemonics:	
SD	System data (from address EA000)
Other mnemonics:	
STOPS	Interrupt display word
SUF	Substitution error
TRAF	Transfer error for data block statements; data word number > data block length
NNN	Statement cannot be interpreted in the SS-11SU (e.g., a 1505 statement)
STS	Operation interrupted by a programmer STOP request or programmed STOP statements
STUES	Block stack overflow: The maximum block call nesting depth of 16 (or 32 in the case of CPU 944) has been exceeded
FEST	Error in the CPU self-test routine
NAU	Power failure
QVZ	Time-out from I/Os: A nonexistent module has been referenced
KOLIP	Interprocessor communication flag transfer list is incorrect
ZTK	Scan time exceeded: The set maximum permissible program scan time has been exceeded
SYSPE	Error in DBT
PEU	I/Os not ready: power failure in the I/O expansion unit; connection to the I/O expansion unit interrupted
BAU	No terminator in the central controller
ASPFA	Battery failure
Other mnemonics:	
UAW	Interrupt display word
ANZ1/ANZ0	00: ACCUM 1 = 0 or 0 is shifted 01: ACCUM 1 > 0 or 1 is shifted 10: ACCUM 1 < 0
OVFL	Arithmetic overflow (+ or -)
ODER (OR)	OR memory (set by "O" operation)
STATUS	STATUS of the operand of the last binary statement executed
VKE	Result of logic operation
BRAB	First scan
KET...KES	Meeting stack entry 1 to 6 entered for A/ and C/ G: O: 1: A/
PCT	
REF-KEG	Statement register
SAZ	Step address counter
DB-ADR	Data block address
BS1-STR	Block stack pointer
NR	Block number (QB, PB, FB, SB, DB)
REL-SA2	Relative step address counter

### Meaning of ISTACK Displays (Continued)

Error/Fault	Fault/Error Cause (ID in ISTACK)	Cause	Remedy
Program scanning interrupted	SUF	Substitution error: A function block was called with an incorrect actual parameter.	Correct the function block call.
	TRAF	Transfer error: <ul style="list-style-type: none"> <li>- A data block statement has been programmed with data word number greater than the data block length.</li> <li>- A data block statement has been programmed without opening a DB first.</li> <li>- DB to be generated is too long for user memory (0 DB operation)</li> </ul>	Correct the program error.
	STS	<ul style="list-style-type: none"> <li>- Software stop by statement (STP, STS)</li> <li>- STOP request from programmer</li> <li>- STOP request from SINEC L1 master</li> </ul>	
	NNN	<ul style="list-style-type: none"> <li>- A statement cannot be decoded.</li> <li>- A parameter has been exceeded.</li> </ul>	Correct the program error.
	STUB8	Block stack overflow. <ul style="list-style-type: none"> <li>- The maximum block call nesting depth of 32 has been exceeded.</li> <li>- Interrupt-driven or time-driven program interrupts cyclic program during processing of an integrated function block and an integrated function block is also called in the interrupting program.</li> </ul>	Correct the program error.  Disable interrupts in the cyclic program before calling integrated function blocks.
	NAU	There has been a power failure.	
	QVZ	Time-out from I/Os: <ul style="list-style-type: none"> <li>- A peripheral byte that was not addressed has been referenced in the program.</li> <li>- An I/O module does not acknowledge.</li> </ul>	Correct the program error or replace the I/O module.
	ZYK	Scan time exceeded: The program scanning time is greater than the IAT monitoring time.	Check the program for continuous loops. If necessary, retrigger the scan time with OB 31 or change the monitoring time.

**Meaning of the ISTACK Displays**

Fault/Error	Fault/Error Cause (ID in ISTACK)	Cause	Remedy
Cold Restart is not possible.	NINEU SYNFEHV KOPFH	Faulty block: <ul style="list-style-type: none"> <li>• Compressing has been interrupted by a power failure.</li> <li>• Block transfer between programmer and PLC was interrupted by a power failure.</li> <li>• Program error (TIE/TNS/BMW)</li> </ul>	Perform an Overall Reset. Reload the program.
	KOLIF	DB 1 is programmed incorrectly.	Check the following: <ul style="list-style-type: none"> <li>• ID for interprocessor communication flag definitions ("MAS601");(see Section 12.1)</li> <li>• ID for the part of DB 1 to be interpreted ("DB 1");(see Section 11.3)</li> <li>• the end IDs in each case for interprocessor communication flag definitions or for the part of DB 1 to be interpreted</li> </ul>
	PEST	There is an error in the self-test routine of the CPU.	Replace the CPU.
Faulty submodule	ASPPA	The submodule ID is illegal: (AG 110S/135U/150U submodule)	Plug in the correct submodule.
Battery failure	BAU	There is no battery or the battery is low and the retentive feature is required.	Replace the battery. Perform an Overall Reset. Reload the program.
I/Os not ready	PEU	The I/Os are not ready: <ul style="list-style-type: none"> <li>• There has been a power failure in the expansion unit.</li> <li>• The connection to the expansion unit has been interrupted.</li> <li>• There is no terminator in the central controller.</li> </ul>	<ul style="list-style-type: none"> <li>• Check the power supply in the expansion unit.</li> <li>• Check the connection.</li> <li>• Install a terminator in the central controller.</li> </ul>
Program scanning interrupted	STOPs	The mode selector is on STOP.	Put the mode selector on RUN.

## منابع

- ۱) simens-simatic s5-115u. ۱۹۹۶. control and monitoring systems.
- ۲) Ian G. Warnock. ۱۹۸۸. programmable controllers operation and application. prentice-hall
- ۳) راهنمای جامع PLC کنترل کننده های منطقی برنامه پذیر، مهندس فربد قابوس، نشر آفرینگ، ۱۳۸۲
- ۴) راهنمای جامع STEP7 جلد اول، محمدرضا ماهر، شرکت صابکو، ۱۳۸۳
- ۵) عملکرد و کاربردهای PLC در اتوماسیون صنعتی، یان وارناک، ترجمه دکتر سید علی اکبر صفوي و مهندس حسین شجاعی، انتشارات نص، ۱۳۸۴.
- ۶) کنترویک، راهنمای کاربردی PLC
- ۷) مجموعه جزوات آموزشی خانم مهندس داداشی.