مجموعه‌ی آموزشی

PLC

گردآوری و تنظیم :

وحید کارگرمقدم

Edition 3

تقدیم به همسرم سرکار خانم آذر سزاوار ذاکران

فهرست مطالب

فصل اول

پیش نیاز

در این فصل مطالبی بیان می گردد که خواننده گرامی از قبل با آنها آشنایی دارد و در حقیقت پیش نیاز اتوماسیون صنعتی و برنامه نویسی PLC می باشد ، لذا بطور اجمال به این مفاهیم خواهیم پرداخت و کسب اطلاعات مشروح به عهده خواننده محترم واگذار می گردد . همچنین در پایان این فصل به شرح برخی از مفاهیم کامپیوتری می پردازیم که نیاز دانش آموزان برای موفقیت در آزمون فنی و حرفه ای می باشد .

مدار فرمان :

بطور کلی مدار فرمان عبارت است از مداری که فرامین کنترلی را برای مدار قدرت صادر می کند . این مدارات با توجه به ورودی مطلوب از طرف اپراتور ، خروجی را برای مدار قدرت ارسال می کنند . مدارات فرمان بدلیل پیچیدگی زیادی که دارند باید یکبار و بطور صحیح پیاده سازی شوند زیرا در صورت نقص ، عیب یابی آن مشکل بوده و معمولا باید برای رفع عیب ، مدار فرمان دوباره طراحی و پیاده سازی شود . در طراحی مدارات فرمان قطعات مختلفی سهیم هستند که مدار فرمان به کمک این قطعات مدار قدرت را کنترل می کند . مهمترین قطعات استفاده شده در مدارات فرمان کنتاکتورها ، رله ها و راه انداز های موتور هستند .

رله ها :

رله یک کلید الکتریکی است که با جریان کمی عمل می کند و کنتاکت های آن می تواند جریان زیادی را عبور دهند . رله در قرن نوزدهم برای استفاده در تلگراف اختراع شد و در قرن بیستم از رله در سیستم های کنترل استفاده گردید . در مقایسه با سایر ادوات الکتریکی رله ها بخاطر داشتن قسمت های فیزیکی متحرک ، سرعت و عمر محدودی دارند . رله های الکتریکی از نظر سرعت ، جریان عبوری ، اندازه ، قابلیت اطمینان و عمر مفید بر رله های الکترومکانیکی ارجحیت دارند . معمولا رله ها برای محدوده کمتر از 5 آمپر مناسب است .

کنتاکتور ها :

کنتاکتور یک کلید مغناطیسی می باشد که یکی از اجزای مهم در مدارات فرمان الکتریکی بحساب می آید . کنتاکتور با استفاده از خاصیت الکترومغناطیس ، مانند رله ها تعدادی کنتاکت را به یکدیگر وصل و یا از یکدیگر جدا می کند ، از این خاصیت جهت قطع و وصل و یا تغییر اتصال در مدارات استفاده می شود . استفاده از کنتاکتور بجای کلیدهای دستی دارای مزایایی می باشد که بشرح زیر است :

1 – کنترل و راه اندازی مصرف کننده یا سیستم از راه دور و چندین نقطه بطور همزمان

2- سرعت بالای قطع و وصل کنتاکتور و طول عمر بالای قطعات

3- امکان طراحی مدارات فرمان خودکار جهت کنترل سیستم های پیچیده

4- کنتاکتور دارای ضریب ایمنی بالا جهت بهره بردار یا اپراتور است

5- در هنگام قطع جریان برق و اتصال مجدد آن ، مصرف کننده یا سیستم خود بخود راه اندازی نمی شود و باید مجددا سیستم را استارت نمود

معمولا کنتاکتورها برای محدوده بیشتر از 15 آمپر مناسب است .

راه اندازهای موتور :

یک کنتاکتور است که قسمت محافظ در برابر اضافه بار به آن اضافه شده است که هر زمان موتور دچار اضافه بار گردید ، کنتاکتور عمل می کند و موتور را از زیر بار خارج می کند . رله ها و کنتاکتورها و راه اندازهای موتور بهم شبیه هستند زیرا تمام آنها دارای بوبین متحرک هستند که با تحریک آن یکسری اتصالات بهم مرتبط و یکسری از هم جدا می شوند .

موتورهای پله ای (Stepped Motor) :

موتورهای پله ای نمونه ای از موتورهای الکتریکی هستند که بدون استفاده از فیدبک امکان کنترل سرعت و تنظیم موقعیت حرکتی را در اختیار ما قرار می دهند . با تحریک ورودی توسط پالس موتور به اندازه چند درجه حول محور خود دوران می کند . در حقیقت یک موتور پله ای پالس الکتریکی را به حرکت مکانیکی تبدیل می کند . عملکرد اصلی یک موتور پله ای به شفت موتور اجازه می دهد تا به اندازه زاویه ای دقیق مطابق پالس های الکتریکی ارسالی به موتور بچرخد ، از آنجا که شفت موتور فقط به اندازه زاویه طراحی شده هنگام ارسال پالس الکتریکی حرکت می کند ، می توان با کنترل پالس های الکتریکی ارسالی موقعیت و سرعت را کنترل کرد . گشتاور نگهداری به موتورهای پله ای این اجازه را می دهد که موقعیت خود را هنگام توقف بطور محکم حفظ کنند . موتور پله ای عموما در موقعیت توقف بدون انرژی باقی می ماند و هنگامی که تغذیه موتور به کلی قطع شود بصورت مغناطیسی در موقعیت قبلی خود قفل می شود .

موتورهای خود فرمان (Servo Motor) :

موتورهای با قدرت بالایی هستند که برای جابجایی اوزان سنگین مورد استفاده قرار می گیرند و مستقیما توسط ولتاژ AC تغذیه می شوند که البته نوع DC آن نیز موجود می باشد .

شیرهای برقی :

از یک سیم پیچ مغناطیسی به اضافه یک میله متحرک تشکیل شده اند . هنگامیکه سیم پیچ برقدار می گردد ، میله را بطرف خود می کشد و بدین ترتیب مسیر شیر باز می گردد و راه برای روان شدن و گردش مایع فراهم می شود . زمانیکه سیم پیچ از تحریک می افتد نیروی کشش فنر میله را به حالت اولیه خود برمی گرداند و راه گردش مایع مسدود می شود .

لیمیت سوئیچ :

یک قطعه مکانیکی می باشد که از کنتاکت فیزیکی برای آشکارسازی حضور یا عدم حضور یک جسم استفاده می کند . هنگامیکه جسم هدف با محرک کنتاکت فیزیکی پیدا می کند ، محرک از موقعیت عادی خود به موقعیت کاری تغییر مکان می دهد این عمل مکانیکی کنتاکت های بدنه سوئیچ را فعال می کند و خروجی صادر می شود .

سنسور :

یک نوع مبدل سیگنال می باشد که سیگنال غیرالکتریکی ورودی را به سیگنال الکتریکی برای خروجی تبدیل می کند . سنسورها پارامترهای مختلف نظیر سرعت ، دما ، رطوبت ، جابجایی و غیره را به سیگنال الکتریکی تبدیل می کنند . در صنعت طیف وسیعی از سنسورها استفاده می شود که در ادامه به معرفی آنها خواهیم پرداخت .

ترموکوپل :

یک نوع سنسور دما است که از اتصال دو فلز غیر همجنس در یک انتخاب بدست می آید . اصول کار ترموکوپل بر مبنای اثر سیبک است (وقتی دو فلز غیرهمجنس از یک سمت بهم وصل شوند و محل پیوند حرارت داده شود ، در سمت دیگر اختلاف پتانسیل کوچکی بوجود می آید)

لودسل :

یک سنسور نیرو می باشد که نیرو یا وزن را به سیگنال الکتریکی تبدیل می کند . اساسا لودسل از یک مجموعه استرین گیج تشکیل شده است که معمولا چهار عدد هستند و بصورت مدار پل وتسون بهم اتصال دارند .

سنسور فتوالکتریک (مادون قرمز) :

این سنسور از یک پرتو نوری مدوله شده استفاده می کند که توسط هدف شکسته شده یا منعکس می گردد . سنسور فتوالکتریک قادر به تشخیص نور مدوله شده از نور محیط می باشد . منابع نور از طریق این سنسورها در محدوده سبز قابل رؤیت تا مادون قرمز نامرئی در طیف نوری بکار گرفته می شوند .

سنسور القایی :

این سنسور به کمک خاصیت الکترومغناطیس توانایی تشخیص فلز را در میدان دید خود دارد .

سنسور خازنی :

نسبت به تمام مواد (فلز و غیرفلز) حساس بوده و با حضور قطعه موردنظر در نزدیکی آن و تغییر ظرفیت خازنی ، سوئیچ می کند . این سنسور برای کنترل سطح مایعات بکار می رود .

پتانسیومتر :

یک سنسور موقعیت است . پتانسیومتر یک مقاومت متغیر است که با تغییر مکان بازوی آن مقدار مقاومت تغییر می کند . با اندازه گیری میزان مقاومت بین بازوی متحرک و یکی از سرهای ثابت مکان مشخص می گردد .

کد کننده (Encoder) :

ابزاری الکترومکانیکی است که مکان یا حرکت را تشخیص می دهد و شامل یک LED و یک سنسور نوری می باشد . با حرکت یک صفحه مشبک از جلوی LED یک سری پالس در خروجی سنسور نوری ایجاد می شود که به کمک این پالس ها فاصله محاسبه می شود .

واحدهای ورودی و خروجی آنالوگ :

بیشتر سیگنال های طبیعی تغییرات پیوسته دارند و سنسورهایی که کمیت های فیزیکی مانند فشار ، سرعت ، درجه حرارت را تشخیص می دهند خروجی آنالوگ ایجاد می کنند . برای پردازش این سیگنال ها در یک سیستم دیجیتال لازم است این اطلاعات به دیجیتال تبدیل شوند . مبدل های A/D سیگنال پیوسته را تبدیل به یک کد دیجیتال می کنند که معمولا این کد 8 بیتی می باشد . هرچه تعداد بیت خروجی A/D بیشتر باشد دقت تبدیل بیشتر است . ورودی و خروجی آنالوگ معمولا در بازه 10 – 0 ولت یا 20 – 0 میلی آمپر فعالیت می کند .

دیود نورانی LED :

دیودهای نورانی معمولا از بلور نیمه هادی گالیم _ آرسنیک ساخته می شوند . با افزایش جریان مستقیم ، تولید فوتون های نوری زیادتر شده و در نتیجه شدت نور تابشی این دیود افزایش می یابد . امروزه دیودهای نورانی برای نورهای قرمز ، زرد ، سبز و مادون قرمز ساخته شده اند . از جمله موارد مهم کاربرد دیودهای نورانی مادون قرمز ، مخابرات فیبر نوری است .

NO و NC :

برای هر کنتاکت دو حالت زیر متصور می باشد :

NC یا در حالت عادی بسته ، NO یا در حالت عادی باز

ترانس میتر :

ترانس میتر وسیله ای است که ولتاژ را تبدیل به جریان می کند .

مدارات منطقی :

مداراتی که در آن متغیرها دارای دو مقدار بوده و بوسیله عملگرهای منطقی بهم مرتبط می گردند را مدار منطقی می نامیم . مدارات منطقی به دو دسته کلی تقسیم می گردند : مدارهای ترکیبی (حلقه باز) ، مدارهای ترتیبی (حلقه بسته) .

مدارهای ترکیبی (حلقه باز) :

در این مدارها خروجی لحظه فعلی به ورودی در همان لحظه بستگی دارد ، بعبارت دیگر هر ورودی اعمال شده به سیستم ، خروجی متناظر خود را تولید می نماید . در مدارات حلقه باز اطلاعاتی از خروجی به ورودی داده نمی شود و کنترل حلقه باز زمانی دچار اختلال می شود که اختلال ناخواسته ای باعث شود خروجی ها از حد مطلوب خارج شوند ، در اینصورت ممکن است سیستم کلی از کنترل خارج شود . بعنوان مثالی از این مدارات می توان به مکانیزم کاری یک ماشین لباسشویی اشاره نمود .

مدارهای ترتیبی (حلقه بسته) :

در مدارات ترتیبی حالت فعلی خروجی علاوه بر وضعیت فعلی ورودی ها به وضعیت قبلی خروجی نیز بستگی دارد یعنی خروجی مدار که در لحظه های قبل بدست آمده و در یک واحد حافظه ذخیره گردیده است ، بر وضعیت فعلی خروجی اثر می گذارد . در این نوع کنترل برای جبران اثر اختلال ، خروجی سیستم اندازه گیری می شود و در صورتیکه خروجی از مقدار مطلوب فاصله داشته باشد تدابیر کنترلی مناسب برای جبران آن اعمال می شود . در مدارات ترتیبی عناصر حافظه وجود دارند که اطلاعات خروجی را برای استفاده ورودی در خود نگهداری می کنند . یکی از عناصر حافظه در مدارات ترتیبی فلیپ فلاپ ها هستند .

حافظه ها :

یک واحد حافظه ابزاری است که اطلاعات دودوئی جهت ذخیره شدن به آن منتقل و یا اطلاعاتی که برای پردازش لازم است از آن دریافت می شود . محلی که اطلاعات ، دستورالعمل ها و نتایج حاصل از عملیات منطقی یا حسابی روی داده ها ، بصورت اطلاعات کد شده برای مدت زمان آنی یا دائم در آنجا نگهداری می شود ، حافظه نامیده می شود . دو نوع حافظه در سیستم دیجیتال وجود دارد : حافظه با دستیابی تصادفی RAM ، حافظه فقط خواندنی ROM .

اطلاعات حافظه RAM بگونه ای است که هم می توان آنها را خواند و هم می توان آنها را تغییر و یا حذف نمود اما اطلاعات حافظه ROM فقط قابل خواندن است و نمی توان آن را تغییر داد . در حافظه از نوع RAM ، محتوای حافظه با قطع جریان برق از بین می رود . حافظه های PROM یک حافظه ROM است با این تفاوت که برنامه توسط برنامه نویس نوشته می شود و توسط پروگرامر PROM در حافظه PROM قرار می گیرد و دیگر

قابل تغییر نیست . حافظه های EPROM و EEPROM یک نوع حافظه ROM هستند با این تفاوت که برنامه ای که در حافظه قرار می گیرد را می توان تغییر داد . بدین صورت که حافظه را مدت مشخصی تحت تابش ماوراء بنفش قرار می دهیم ، اتصالات منطقی برنامه از بین می رود و EPROM آماده برنامه ریزی مجدد می شود و همچنین برای پاک کردن برنامه داخلی EEPROM از امواج الکتریکی استفاده می کنیم . حافظه ها از جنس نیمه هادی هستند .

واحدهای حافظه به سه دسته زیر تقسیم می شود :

1 – CPU که واحد پردازش کننده کامپیوتر است

2 – CU که واحد کنترل پردازنده است

3 – ALU که واحد محاسبات منطقی و ریاضی است

Boud Rate :

منظور از Boud Rate یا نرخ ارتباط ، سرعت ارتباط دو سیستم دیجیتالی است و انتقال تعداد بیت در یک ثانیه را گویند .

گذرگاه یا BUS :

BUS در لغت به معنی اتوبوس یا وسیله حمل و نقل عمومی بوده ، در اصطلاح کامپیوتری وسیله ای است که حمل و نقل عمومی داده ها را بر عهده دارد . در این گذرگاه قسمتی که حمل و نقل و جابجایی اطلاعات را بر عهده دارد ، دیتا باس Data Bus می نامند و قسمتی از مسیر عمومی که جابجایی آدرس ها را بر عهده دارد ، آدرس باس Address Bus گفته می شود . این گذرگاه مجموعه ای از خطوط سخت افزاری است که جهت انتقال داده ها بین اجزای یک سیستم کامپیوتری ، مورد استفاده قرار می گیرد . به عبارت دیگر گذرگاه ، یک مسیر مشترک است که بین بخشهای مختلف سیستم از جمله ریزپردازنده ، حافظه و درگاه های ورودی و خروجی و دیگر قسمت ها ارتباط برقرار می نماید . در سیستم های کامپیوتری ، گذرگاه توسط ریزپردازنده کنترل شده ، به انتقال انواع مختلفی از اطلاعات اختصاص می یابد . بعنوان مثال گروهی از خطوط ، داده ها را انتقال داده و گروه دیگر آدرس های محل استقرار اطلاعات را منتقل ساخته ، یک گروه دیگر سیگنال های کنترل را جهت حصول اطمینان از اینکه بخش های مختلف سیستم از مسیر مشترک خود بدون ایجاد تداخل استفاده می کنند ، عبور می دهند که به این بخش از گذرگاه کنترل باس Control Bus گویند . گذرگاه ها با تعداد بیت هایی که در هر لحظه می توانند انتقال دهند ، مشخص می شوند . بعنوان مثال یک کامپیوتر دارای 8 بیتی در هر لحظه 8 بیت از داده ها و یک کامپیوتر دارای گذرگاه 16 بیتی در هر لحظه 16 بیت از داده ها را انتقال می دهند .

پورت پارالل :

درگاهی است که اطلاعات را بصورت موازی و هشت بیت هشت بیت دریافت یا ارسال می کند . پورت پارالل می تواند هشت خط ارتباطی را بطور همزمان ارسال کند .

پورت سریال Com :

اطلاعات در پورت سریال بصورت بیتهای پشت سر هم دریافت می شوند . این پورت معمولا دارای 9 پین و یا 25 پین می باشد .

ALU :

حافظه دارای بخشی بنام ALU یا واحد محاسبات منطقی و ریاضی است . این بخش در حافظه مسئول انجام اعمال حسابی مانند جمع ، تفریق ، ضرب و تقسیم و اعمال منطقی مانند AND ، OR ، NOT می باشد .

فصل دوم

مفاهیم منطقی

در این فصل می خواهیم درباره مدارات منطقی و اساساً منطق دیجیتال بحث کنیم ، همچنین در این فصل از مبناهای اعداد صحبت خواهیم کرد و تبدیل مبناها را به یکدیگر آموزش خواهیم داد .

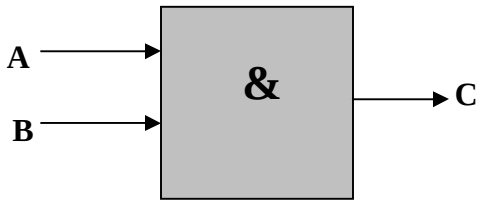
منطق دیجیتال :

در مدار منطقی دیجیتال از المان های الکترونیکی نظیر دیود و ترانزیستور استفاده می شود . از ترکیب چند المان توابع منطقی ایجاد شده که هرکدام منطق خاصی را پیروی می کنند . در این مدارها از دو اصطلاح صفر و یک بسیار استفاده می کنند ، مفهوم این دو اصطلاح بدین شرح است : در یک سیستم تنها چیزی که برای المان های الکتریکی قابل فهم است ، بود یا نبود ولتاژ است چون منطق دیجیتال از این خاصیت تبعیت می کند پس باید دو سطح از ولتاژ را برای درک سیستم تعریف نمود مثل 0 ولت و 24 ولت . در این سیستم سطح ولتاژ 24 ولت ، یک و سطح ولتاژ 0 ولت ، صفر تلقی می شود .

توابع منطقی دیجیتال دارای یک یا چند ورودی و یک خروجی می باشند که وضعیت خروجی متناسب با وضعیت ورودی می باشد . در مدارهای منطقی یا دیجیتال عناصری وجود دارد که توانایی انجام عملیات بر روی صفر و یکها را دارند که به آنها گیت (Gate) می گویند . هفت گیت منطقی دیجیتال موجود می باشد : AND ، NOT ، NAND ، OR ، NOR ، XOR ، XNOR

گیت منطقی AND :

در این تابع خروجی فقط زمانی که تمام ورودیها در وضعیت یک قرار دارند یک می شود . عملکرد این تابع مثل تیغه های باز سری می باشد .

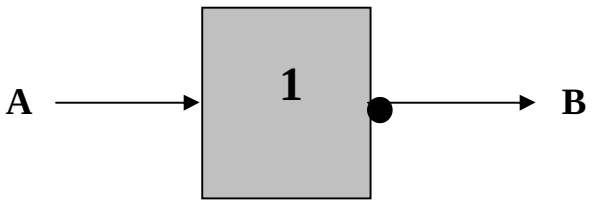


A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

جدول درستی

گیت منطقی NOT :

این تابع فقط یک ورودی دارد و همیشه وضعیت خروجی عکس وضعیت ورودی است . این تابع معادل کنتاکت بسته در مدار فرمان می باشد .

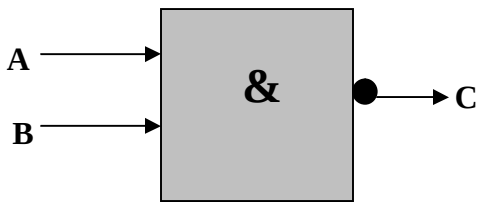


A	B
0	1
1	0

جدول درستی

گیت منطقی NAND :

خروجی این تابع فقط زمانی که همه ورودیها یک باشند در وضعیت صفر قرار می گیرد . در حقیقت این تابع ، عکس تابع منطقی AND عمل می کند .



A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

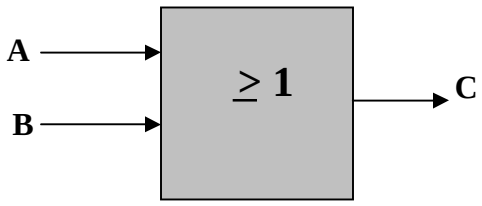
جدول درستی

گیت منطقی OR :

خروجی این تابع فقط زمانی که همه ورودی ها صفر باشند در وضعیت صفر قرار می گیرد . این تابع معادل تیغه های باز موازی می باشد .

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

جدول درستی

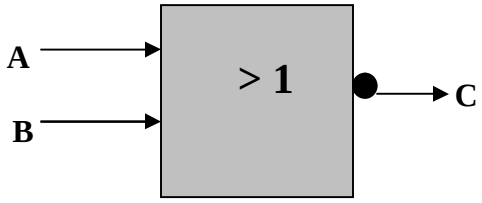


گیت منطقی NOR :

خروجی این تابع فقط زمانی که همه ورودی ها صفر باشند در وضعیت یک قرار می گیرد . در حقیقت این تابع ، عکس تابع منطقی OR عمل می کند .

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

جدول درستی

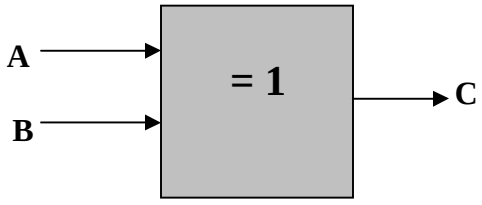


گیت منطقی XOR :

خروجی این تابع فقط زمانی که تنها یکی از ورودی ها یک باشد در وضعیت یک قرار می گیرد . این تابع همانند مدار کلید تبدیل عمل می کند . اصطلاحا به این تابع ، گیت فرد می گویند .

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

جدول درستی

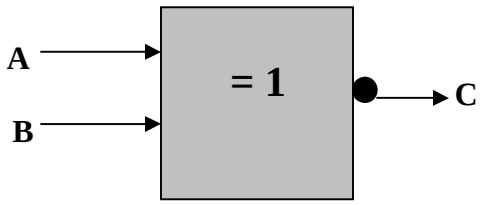


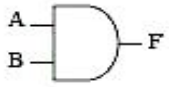
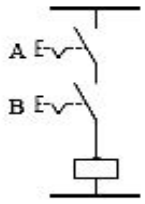
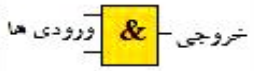
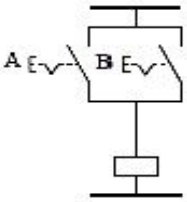
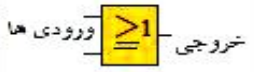
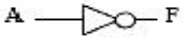
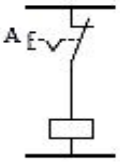
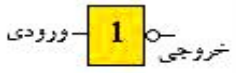
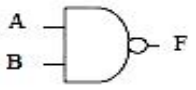
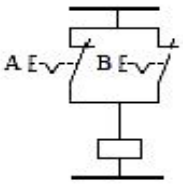
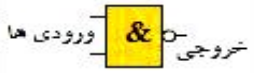
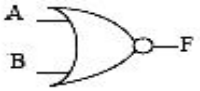
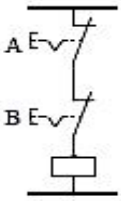
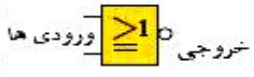
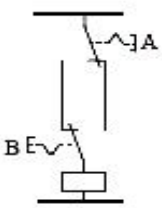
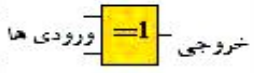
گیت منطقی XNOR :

خروجی این تابع فقط زمانی که دو ورودی در یک وضعیت باشند در وضعیت یک قرار می گیرد . در حقیقت این تابع ، عکس تابع منطقی XOR عمل می کند . اصطلاحا به این تابع ، گیت زوج می گویند .

A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

جدول درستی



عملگر منطقی	جدول صحت	علامت اختصاری	مدار کلیدی	شکل بلوکی
AND	خروجی / ورودی ها A B F 0 0 0 0 1 0 1 0 0 1 1 1			
OR	خروجی / ورودی ها A B F 0 0 0 0 1 1 1 0 1 1 1 1			
NOT	A F 0 1 1 0			
NAND	خروجی / ورودی ها A B F 0 0 1 0 1 1 1 0 1 1 1 0			
NOR	خروجی / ورودی ها A B F 0 0 1 0 1 0 1 0 0 1 1 0			
XOR	خروجی / ورودی ها A B F 0 0 0 0 1 1 1 0 1 1 1 0			

مفهوم بیت :

با ترکیب چند تابع منطقی سلول حافظه تشکیل می شود ، این بدان معنی است که وضعیت صفر و یا یک بودن ورودی یا خروجی را در خود حفظ می کند ، به این سلول حافظه یک بیت گفته می شود .

اعداد را می توان در مبناهای عددی مختلف نمایش داد . آشناترین مبنای اعداد ، مبنای ده می باشد . در مبنای ده کلیه اعداد با ترکیبی از اعداد 0 تا 9 حاصل می گردند . از دیگر مبناهای عددی رایج می توان به مبنای دو اشاره نمود ، همانند اعداد مبنای ده هر رقم یک عدد در مبنای دو دارای ارزش خاصی می باشد . در این مبنا تنها اعداد صفر و یک موجود می باشند ، مثلا عدد 01101 یک عدد پنج رقمی در مبنای دو می باشد . هر رقم در مبنای دو را یک بیت و هر هشت بیت را یک بایت و هر دو بایت را یک کلمه می نامند . جهت بدست آوردن معادل مبنای دو یک عدد دهدهی این عدد را بطور متناوب بر دو تقسیم می کنیم تا جاییکه خارج قسمت نهایی بر دو قابل تقسیم نباشد ، باقیمانده های بدست آمده را از انتها به ابتدا به ترتیب از چپ به راست بعد از آخرین خارج قسمت می نویسیم و اینگونه معادل دودویی اعداد بدست می آید :

$(41)_{10} = (\quad ? \quad)_{2}$

$41 \div 2 \longrightarrow$ خارج قسمت 20 و باقیمانده 1

$20 \div 2 \longrightarrow$ خارج قسمت 10 و باقیمانده 0

$10 \div 2 \longrightarrow$ خارج قسمت 5 و باقیمانده 0

$5 \div 2 \longrightarrow$ خارج قسمت 2 و باقیمانده 1

$2 \div 2 \longrightarrow$ خارج قسمت 1 و باقیمانده 0

چون خارج قسمت بر دو بخش پذیر نیست لذا طبق روش گفته شده معادل باینری عدد را می نویسیم :

$(41)_{10} = (101001)_{2}$

جهت تبدیل یک عدد از مبنای دو به مبنای ده می توان هر رقم را در ارزش مکانی خود ضرب نمود و سپس حاصلضربهای بدست آمده را با هم جمع نمود :

$(101001)_{2} = (\quad ? \quad)_{10}$

$101001 \longrightarrow 1 \times 32 + 0 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = 41$

از دیگر مبناهای عددی پرکاربرد مبنای 16 می باشد . یک عدد در مبنای 16 معادل یک عدد دودویی چهار رقمی است . جدول زیر مبنای شانزده و معادل دهدهی و دودویی آن را نشان می دهد .

مبنای شانزده	مبنای دو	مبنای ده
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

جهت تبدیل یک عدد دودویی به عدد مبنای شانزده کافی است از سمت راست اعداد را چهار رقم چهار رقم جدا نموده و سپس معادل مبنای شانزده آنها را جایگزین نماییم .

$$(10001101)_2 = (\ ?\ )_{16}$$

$$1000\ ,\ 1101 \Longrightarrow 8D$$

$$(10001101)_2 = (\ 8D\ )_{16}$$

جهت تبدیل یک عدد از مبنای شانزده به مبنای دو به جای هر عدد معادل دودویی چهار رقمی آن را جایگزین می کنیم .

$$(A3B)_{16} = (\ ?\ )_2$$

$$1010\ ,\ 0011\ ,\ 1011 \Longrightarrow A\ ,\ 3\ ,\ B$$

$$(A3B)_{16} = (\ 101000111011\ )_2$$

مبنای هشت	مبنای دو
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

جهت تبدیل یک عدد دودویی به عدد مبنای هشت کافی است از سمت راست اعداد را سه رقم سه رقم جدا نموده و سپس معادل مبنای هشت آنها را جایگزین نماییم .

$$(001101)_2 = (\ ?\ )_8$$

$$001\ ,\ 101 \Longrightarrow 1\ ,\ 5$$

$$(001101)_2 = (\ 15\ )_8$$

جهت تبدیل یک عدد از مبنای هشت به مبنای دو به جای هر عدد معادل دودویی سه رقمی آن را جایگزین می کنیم .

$$(23)_8 = (\ ?\ )_2$$

$$010\ ,\ 011 \Longrightarrow 2\ ,\ 3$$

$$(23)_8 = (\ 010011\ )_2$$

کد BCD : در BCD هر رقم در مبنای دهدهی بطور جداگانه به شکل دودویی کد می شود . هر رقم در چهار بیت کد می شود ، چون بزرگترین رقم دسیمال یعنی 9 در باینری چهار رقمی است .

اعداد صحیح Integer : INT عدد صحیح شانزده بیتی می باشد ، بیت پانزدهم نشان دهنده علامت عدد است . اگر صفر باشد عدد مثبت و اگر یک باشد عدد منفی می باشد . بازه این اعداد بین 32768 – تا 32767 + می باشد .

مکمل یک :

بدین صورت بدست می آید که کافیست تمام بیت های عدد موردنظر را NOT کنیم . به مثال زیر توجه کنید :

مکمل یک عدد 1001101 برابر است با 0110010

مکمل دو :

بدین صورت بدست می آید که کافیست تمام بیت های بعد از اولین بیت یک از سمت راست عدد را NOT کنیم . به مثال زیر توجه کنید :

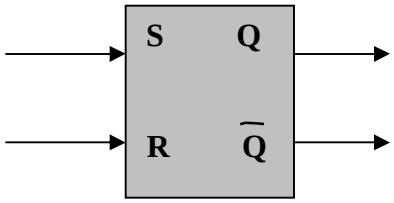
مکمل دو عدد 1010010 برابر است با 0101110

فلیپ فلاپ :

کوچکترین عنصر حافظه در یک مدار ترتیبی را فلیپ فلاپ می نامند . یک فلیپ فلاپ قادر است مادامیکه ورودیهاش تغییر نکرده و جریان تغذیه آن نیز قطع نشده باشد ، یک مقدار را بمدت نامحدود حفظ نماید . انواع مختلفی از فلیپ فلاپ وجود دارد که عبارتند از : فلیپ فلاپ نوع D ، فلیپ فلاپ نوع JK ، فلیپ فلاپ نوع RS ، فلیپ فلاپ نوع T . از آنجا که فلیپ فلاپ کاربردی در PLC فلیپ فلاپ RS می باشد ، به بررسی این فلیپ فلاپ خواهیم پرداخت .

فلیپ فلاپ RS :

جدول درستی و نمای شماتیکی این فلیپ فلاپ بصورت زیر است :



S	R	Q (t + 1)
0	0	Q(t) بدون تغییر
0	1	0
1	0	1
1	1	غیرقابل پیش بینی

* جدول بیان شده برای تجزیه و تحلیل و تعیین طرز کار فلیپ فلاپ مفید است . این جدول به هنگام معلوم بودن ورودی ها و حالت فعلی ، حالت بعدی را تعیین می کند .

Q(t) ⟶ حالت فعلی

Q(t+1) ⟶ حالت بعدی

* حالت یک فلیپ فلاپ با تغییر در ورودی کنترل عوض می شود ، این تغییر لحظه ای را تریگر گویند .

جدول کارنو :

برای ساده سازی توابع منطقی (جبر بول) ، از نقشه کارنو استفاده می شود . جدول کارنو یا نقشه کارنو ، یک روش هندسی برای ساده سازی توابع و مدارهای منطقی است .

- به هر 1024 بایت ، یک کیلوبایت می گویند .
- به هر 1024 کیلوبایت ، یک مگابایت می گویند .
- به هر 1024 مگابایت ، یک گیگابایت می گویند .
- به هر 1024 گیگابایت ، یک ترابایت می گویند .

فصل سوم

مفاهیم PLC

در این فصل مفاهیم اولیه PLC را بررسی می کنیم و زبان های برنامه نویسی و واحدهای تشکیل دهنده آن را معرفی خواهیم کرد . در حقیقت فصل حاضر مقدمه بحث اصلی این مجموعه که آموزش PLC است ، می باشد .

پیشرفت های چشمگیر فناوری نیمه هادی در زمینه ساخت ریزپردازنده و حافظه های با حجم بالا امکان ساخت کنترل کننده های منطقی الکترونیکی برنامه پذیر را فراهم آورد . در این کنترل کننده ها برخلاف کنترل کننده های مبتنی بر قسمت های الکترومکانیکی ، برای تغییر منطق کنترل کافی است بدون تغییری در سیم کشی یا قطعات ، فقط برنامه کنترل را تغییر دهیم . در اینصورت می توانیم از یک کنترل کننده منطقی برنامه پذیر هرجا که خواسته باشیم استفاده نماییم .

مزایای استفاده از کنترل کننده های منطقی برنامه پذیر :

1 – استفاده از PLC حجم تابلوهای فرمان را کاهش می دهد .

2 – استفاده از PLC موجب صرفه جویی فراوان در هزینه می گردد .

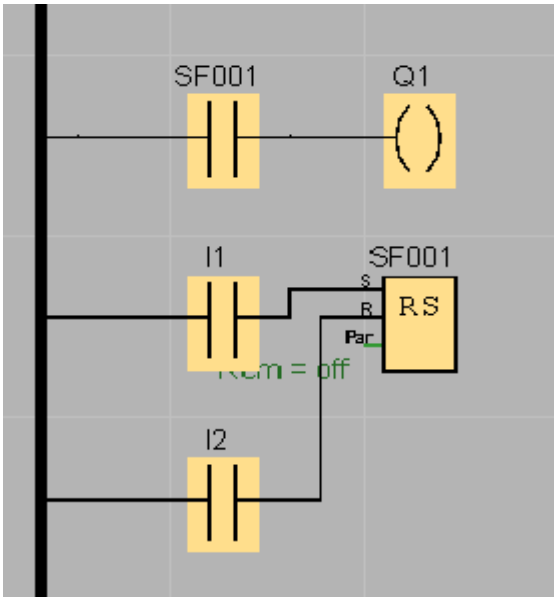
3 – PLC استهلاک مکانیکی ندارد بنابراین علاوه بر طول عمر بیشتر ، نیازی به سرویس و تعمیرات دوره ای ندارد .

4 – مصرف انرژی PLC بسیار کمتر از مدارهای رله ای است .

5 – PLC نویزهای صوتی و الکتریکی ایجاد نمی کند .

6 – عیب یابی مدارات کنترل با PLC سریع و آسان است و معمولا PLC خود دارای برنامه عیب یابی می باشد .

PLCها مبتنی بر میکروپروسسور هستند و با داشتن اجزایی مانند زمان سنج ، شمارنده و ثبات انتقالی ، کنترل فرآیندهای پیچیده را آسان می سازند . PLC ها کامپیوترهای تک منظوره ای هستند که از سه بخش تشکیل شده اند : ورودی ، حافظه و پردازش . اطلاعات ورودی از طریق سنسورها دریافت و در حافظه ذخیره می گردند ، این اطلاعات با توجه به فرامین موجود در حافظه پردازش شده و سپس خروجی ها به نحوی مناسب ساخته می شوند . هر PLC دارای زبان برنامه نویسی خاص خود بوده که رابط بین کاربر و سخت افزار PLC می باشد . مهمترین روشهای برنامه نویسی عبارتند از : روش نردبانی ، روش فلوچارتی ، روش لیست جملات .

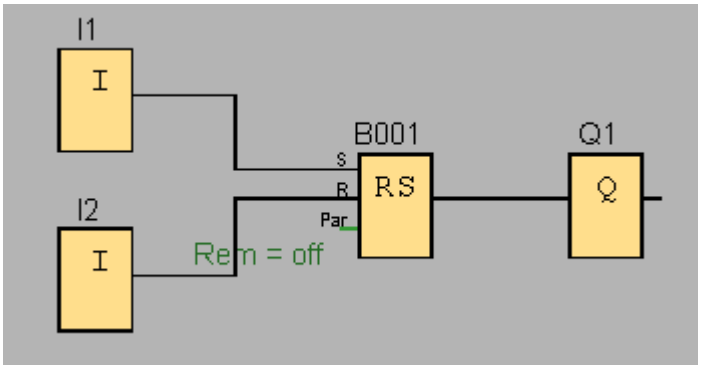


برنامه نویسی به روش نردبانی LAD :

از آنجا که تمام نقشه های کنترل و فرمان منطقی قبل از ظهور PLC ها بصورت نردبانی و یا چیزی شبیه به آن تهیه و طراحی می شد ، لذا سازندگان PLC این روش برنامه نویسی را بعنوان یکی از روش های ممکن برنامه نویسی انتخاب نمودند . در این روش آن دسته از عناصر نردبان که تابع یا عمل خاص و پیچیده ای را انجام می دهند برای سهولت با یک جعبه نمایش داده می شوند . دستورات نوشته شده به روش نردبانی بترتیب از چپ به راست و از بالا به پایین انجام می گردند . میتوان اینگونه بیان داشت که این زبان براساس نقشه های مدار فرمان ، طراحی شده است .

برنامه نویسی به روش فلوچارتی CSF یا نمایش جعبه ای تابع FBD :

در این روش برنامه بصورت بلوکی نوشته شده که در آن هر بلوک بیانگر یک عملگر می باشد ، بدین ترتیب برنامه های نوشته شده به روش FBD عبارتند از یک سری جعبه که به یکدیگر متصل گردیده اند . این روش معمولا بطور مستقل کاربرد چندانی ندارد و اغلب برای عیب یابی و یا شناخت منطق کنترل سیستم بسیار مفید است . این زبان براساس مدارهای الکترونیک و دیجیتال طراحی شده است .



برنامه نویسی به روش لیست جملات STL :

در این روش هر عمل منطقی توسط یک جمله یا عبارت مناسب نوشته می شود . نکته قابل توجه در این روش برنامه نویسی آن است که هر PLC دارای کد دستورات منحصر بفردی می باشد که این دستورات به نوع CPU بکار رفته بستگی دارد . این زبان براساس زبان برنامه نویسی کامپیوتر ایجاد شده است . زبان برنامه نویسی در حالت STL مثل زبان بیسیک یا اسمبلی بوده و نوشتاری است . روش STL نیازهای گرافیکی بسیار کمتری نسبت به دو روش قبل دارد ، لذا نوع و تعداد دستورات قابل درک و اجرا در این روش بیشتر از روش های LAD و FBD می باشد . به همین دلیل برنامه هایی که به روش LAD یا FBD نوشته می شود معمولا قابل تبدیل به STL می باشد ، درحالیکه عکس این قضیه همواره ممکن نیست . در برنامه نویسی به روش STL هر چند خط برنامه که عمل خاصی را انجام می دهد یک Segment می گویند .

```
Network 1      Network Title
Network Comment
A       I0.0
A       I1.0
=       Q3.4
```

عبارت یا Statement :

Statement یا هر خط از برنامه نوشته شده به روش STL ، سطری از برنامه است که معمولا دارای دو بخش زیر است :

1) عملگر یا Operation

2) عملوند یا Operand

عملگر یا Operation :

به عمل منطقی که در عبارت صورت می گیرد ، عملگر گفته می شود . عملگرهای مهم عبارتند از : AND ، OR ، NOT ، .....

عملوند یا Operand :

به قسمتی از عبارت گفته می شود که قرار است یک عمل منطقی (عملکرد) در مورد آن اجرا شود مانند ورودی ها ، خروجی ها ، ..... عملوند خود شامل دو بخش آدرس عملوند و نوع عملوند است . نوع عملوند ، همان ورودی ها ، خروجی ها و غیره هستند و آدرس عملوند ، محل عملوند را مشخص می نماید .

واحدهای تشکیل دهنده PLC :

در PLC های کوچک ، پردازنده ، حافظه نیمه هادی ، ماژول های I/O و منبع تغذیه در یک واحد جای داده شده اند . در PLC های بزرگتر ، پردازنده و حافظه در یک واحد ، منبع تغذیه در واحد دوم و واسطه های I/O در واحدهای بعدی قرار دارند . حافظه ثابت سیستم ، حاوی برنامه ای است که توسط کارخانه سازنده تعبیه شده است . این برنامه وظیفه ای مشابه سیستم عامل DOS در دستگاه های PC دارد که بر روی تراشه های خاصی بنام حافظه فقط خواندنی قرار گرفته است . اطلاعات حافظه تغییرپذیر بر روی تراشه های نیمه هادی ذخیره می شود که امکان برنامه ریزی ، تغییر و پاک کردن آنها توسط برنامه ریز میسر است .

پردازنده CPU :

این واحد اساسی ترین قسمت PLC می باشد . واحد پردازش ، یک ریزپردازنده است و مجموعه اعمال و محاسبات منطقی را انجام می دهد و ارتباط بین واحدهای مختلف را برقرار می سازد . در واحد پردازش عناصر دیگری مثل شمارنده ها و تایمرها تعریف شده اند . اغلب CPUها مجهز به یک باتری پشتیبان هستند ، بنابراین اگر تغذیه ورودی قطع شود و متعاقبا منبع تغذیه نتواند ولتاژ سیستم را تامین کند ، باتری پشتیبان برنامه ذخیره شده در RAM را حفظ می کند .

ماژول های ورودی و خروجی :

ماژول ورودی بصورت الکترونیکی چهار کار اصلی را انجام می دهد ، اولا این ماژول حضور یا عدم حضور سیگنال الکتریکی در تمام ورودی ها را بررسی می کند . ثانیا این ماژول سیگنال مربوط به وصل بودن را از نظر الکتریکی به سطح DC که توسط مدارات الکتریکی ماژول I/O قابل استفاده باشد ، تغییر می دهد . ثالثا این ماژول جداسازی الکترونیکی را با جداکردن خروجی ماژول ورودی از ورودی اش بصورت الکترونیکی انجام می دهد . در نهایت این ماژول سیگنالی را که توسط CPU سیستم PLC قابل تشخیص است ، ایجاد می کند . ماژول خروجی بگونه ای عکس ماژول ورودی عمل می نماید . یک سیگنال DC که از CPU ارسال می گردد ، در هر ماژول خروجی به سیگنال الکتریکی با سطح ولتاژ مناسب بصورت AC یا DC که توسط دستگاه ها قابل استفاده باشد ، تبدیل می گردد .

منبع تغذیه :

منبع انرژی الکتریکی که معمولا استفاده می شود ، منبع جریان متناوب 220 ولت با فرکانس 50 الی 60 هرتز می باشد . از آنجا که اغلب PLCها با ولتاژهای +5 ، - 5 و 24 ولت کار می نمایند لذا هر PLC باید مجهز به مدارهایی باشد که بتواند این تبدیل ولتاژها را انجام دهد . این تبدیل با استفاده از یک منبع تغذیه داخلی انجام می شود .

در یک تقسیم بندی ، PLC ها در دو غالب : PLC ها با کاربرد محلی و PLC ها با کاربرد وسیع تقسیم می گردند .

برنامه ریز PLC :

برای نوشتن برنامه در PLC از وسیله ای بنام PG (Programmer) برنامه ریز دستی استفاده می شود . امروزه برای نوشتن برنامه PLC عمدتاً از دستگاه کامپیوتر استفاده می شود زیرا کاربرها با کامپیوتر و دکمه های آن آشنایی کافی داشته و دستگاهی چند منظوره است که با نصب نرم افزار مربوط به PLC ، به راحتی مورد استفاده قرار می گیرد .

زمان مرور برنامه Scan Time :

در یک PLC ، بمدت زمانیکه طول می کشد تا ورودی خوانده شود ، عمل پردازش روی آن صورت بگیرد و نتیجه به خروجی منتقل شود ، زمان مرور برنامه می گویند و هرچه این زمان کمتر باشد سرعت عمل PLC بالاتر خواهد بود .

زمان مرور برنامه به عوامل زیر وابسته است :

1- تاخیر زمان ورودی ها
2- زمان اجرای برنامه
3- تاخیر زمان خروجی ها

کوپل کننده های نوری :

جهت حفاظت مدارات داخلی PLC و جلوگیری از نویزهائی که معمولا در محیط های صنعتی وجود دارند ، ارتباط ورودی ها با مدارات داخلی PLC توسط کوپل کننده های نوری (Opto Coupler) انجام می گردد . در داخل PLC ایزولاسیون الکتریکی توسط آپتوکوپلر انجام می شود .

بیت RLO :

هنگامی که PLC اجرای برنامه ای را آغاز می کند ، مقدار عملوند یا سطر اول برنامه را در بیت بخصوصی که به RLO موسوم است ، قرار می دهد و در اجرای سطر بعدی ، RLO را با عملوند بعدی مطابق برنامه ، ترکیب می کند و مجددا حاصل را در RLO قرار می دهد . این روند تا زمانی ادامه پیدا می کند که در سطری از برنامه به دستور هم ارزی (=) برسد . پس از انجام این عمل یعنی انتساب بیت RLO به عملوند موجود در سطر هم ارزی ، RLO مقدار خود را از دست داده ، پذیرای مقدار جدید می گردد . لذا مجددا مقدار عملوند سطر بعد از عمل هم ارزی در RLO قرار می گیرد و PLC ، این روند را تا پایان برنامه ادامه دهد . بیت RLO یک ثبات می باشد که نتیجه عملیات منطقی در آن قرار می گیرد . بنابراین ارزش بیت RLO به نتیجه عملیات منطقی سطر بستگی دارد . انجام برخی از دستورات به RLO وابسته (RLO Dependent) و برخی دیگر غیر وابسته اند (RLO Independent) .

وابستگی یک دستور به RLO بدین معنی است که جهت اجرا شدن آن باید بیت RLO سطر قبلی 1 باشد ، در غیر این صورت ، این دستور اجرا نمی شود . عدم وابستگی یک دستور به RLO بدین معنی است که این دستور صرف نظر از مقدار بیت RLO ، اجرا می شود .

طرح برنامه :

قبل از نوشتن برنامه مسلما نیازمند طرحی کلی از برنامه موردنظر هستیم . باید پس از طی مراحل مختلف طرح نهایی برنامه بدست آید و سپس بکمک طرح برنامه ، برنامه نوشته شود و روی PLC بارگذاری و اجرا گردد . در حقیقت در طراحی و اجرای یک فرآیند همانند برنامه نویسی به زبانهای کامپیوتر ، باید مراحلی را رعایت نماییم و در اجرای این مراحل گام به گام جلو برویم تا به هدف برسیم .

مراحل طراحی و اجرای یک فرآیند :

1 – تعریف صورت مسئله بصورت کاملا دقیق
2 – تعیین مراحل انجام فرآیند بصورت کاملا دقیق و مرتب
3 – تعیین وسایل و لوازم مورد استفاده از فرآیند و تهیه لیست تجهیزات
4 – اجرای طرح اولیه و کلی بدون درنظر گرفتن جزئیات
5 – خلاصه سازی مدار
6 – وارد جزئیات کار شدن و روند کاری عمل را طرح نمودن
7 – نوشتن برنامه به یکی از سه روش موجود
8 – استفاده از قوانین موجود در برنامه نویسی جهت ساده سازی مطالب برنامه
9 – استفاده از برنامه های موجود در حافظه ماشین جهت ساده سازی و جلوگیری از تکرار مطالب
10 – نوشتن برنامه در کامپیوتر و اجرای آن جهت آزمایش اولیه

با اجرای مراحل فوق ما می توانیم به راحتی و بدون هیچگونه مشکلی به طراحی و برنامه ریزی فرآیندهای گوناگون پرداخته و با تکرار و تمرین در این امر مهارت پیدا نماییم .

فصل چهارم

PLC
LOGO

در این فصل در مورد LOGO صحبت خواهیم کرد و سخت افزار و نرم افزار آن معرفی می شود . همچنین روش برنامه نویسی LOGO را بیان خواهیم داشت .

سخت افزار LOGO :

LOGO یک ماژول همه منظوره دیجیتال از تولیدات شرکت زیمنس آلمان می باشد که در سال 1996 به بازار جهانی عرضه شد . همانند هر سیستم دیجیتال مبتنی بر پردازنده ، این کنترل کننده نیز از واحدهای کوچکتری که در ارتباط تنگاتنگ با یکدیگر می باشند ، تشکیل شده است .

مهمترین واحدهای تشکیل دهنده یک LOGO عبارتند از :

1 – منبع تغذیه (تغذیه سخت افزار از طریق دو ترمینال به نامهای N و L1 انجام می شود. ولتاژ ورودی دستگاه به این ترمینال ها اتصال می یابد)

2 – ورودی ها

3 – خروجی ها (نوع خروجی ها روی سخت افزار مشخص می شود . این خروجی ها بصورت رله ای یا ترانزیستوری می باشند)

4 – دریچه اتصال ماژول با پوشش

5 – صفحه کنترل (این صفحه در مدل های RCO وجود ندارد)

6- صفحه نمایش (این صفحه در مدل های RCO وجود ندارد)

7 – نمایش وضعیت حالت های Stop\Run

8 – میله های قفل کننده مکانیکی

9 – حفره های قفل گذاری مکانیکی

11 – واحد پردازش مرکزی CPU

12 – اسلاید

چگونگی تشخیص مدل های LOGO از روی دستگاه :

برای تشخیص مدل های LOGO از روی دستگاه ، آشنایی با یکسری از اعداد و حروف که هریک بیانگر یک ویژگی از دستگاه می باشد ، لازم است که عبارتند از :

- عدد 12 : نشان دهنده نوع 12 VDC می باشد .
- عدد 24 : نشان دهنده نوع 24 VDC می باشد .
- عدد 230 : نشان دهنده نوع 230 VAC می باشد .
- حرف R : نشان دهنده خروجی های رله ای می باشد . در صورتیکه حرف R درج نشده باشد مفهوم آن اینست که خروجی ها بصورت ترانزیستوری هستند .
- حرف C : نشان دهنده دارابودن سوئیچ تایمر و زمان بندی هفتگی است .
- حرف O : نشان دهنده نسخه بدون نمایشگر است .

ماژول های خروجی ترانزیستوری : در LOGO با خروجی ترانزیستوری ، خروجی ها از لحاظ اتصال کوتاه و بار اضافی حفاظت شده اند و ولتاژ کمکی برای خروجی ها مورد نیاز نیست و LOGO آنرا تامین می کند . در حقیقت ماژول خروجی ترانزیستوری منحصرا در DC بکار گرفته می شود و هیچ حالت متحرک نداشته و عمل قطع و وصل با جرقه همراه نخواهد بود . سرعت سوییچ آن بسیار بالا خواهد بود .

ماژول های خروجی رله ای : در LOGO با خروجی رله ای ، خروجی ها می بایست از منبع تغذیه و ورودی های ایزوله باشند . این خروجی بیشتر برای حالت AC مناسب است اما در حالت DC نیز کاربرد دارد .

- ماکزیمم جریان کلیدزنی در خروجی ها 10A می باشد .

کابل Interface :

جهت برقراری ارتباط مابین LOGO و کامپیوتر و بمنظور انتقال اطلاعات شرکت سازنده ، کابل های واسطی را به همراه سخت افزار LOGO ارائه کرده است که به کابل Interface معروف می باشد .

نصب و سیم بندی LOGO :

جهت نصب و سیم بندی سخت افزار LOGO باید به نکات زیر توجه کرد :

1 – سیم بندی را بصورتی انجام دهید که در محل برخورد سیم ها بهم بی نظمی ایجاد نشود .

3 – رشته سیم ها نباید بیش از حد ممکن پیچ بخورند و حداکثر نیروی پیچشی 0.5 N/m می باشد .

4 – سیم بندی را برای مسافت های کوتاه تا حد ممکن محافظت کنید . اگر سیم بندی برای مسافت های طولانی استفاده شود ، در آنصورت لازم است از کابل های پوششی برای حفاظت بیشتر استفاده گردد .

5 – در هنگام سیم بندی باید مدارهای متناوب AC ، مدارهای جریان مستقیم DC با ولتاژ زیاد و با سیکل های سوئیچینگ سریع و سیم بندی های با ولتاژ کم به طور جداگانه نگهداری شوند .

6 – سیم بندی در مقابل اضافه ولتاژ ناگهانی مثل صاعقه ، محافظت شود .

طریقه نصب سخت افزار LOGO بر روی ریل :

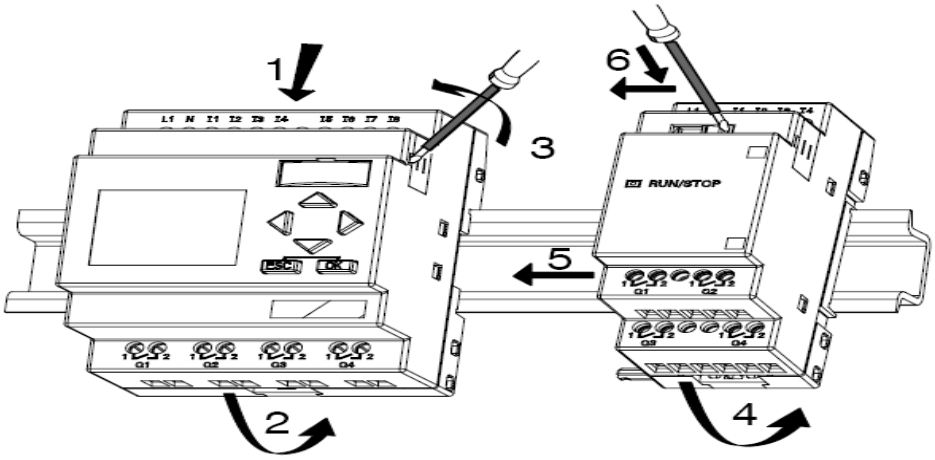
در شکل ، شما نحوه سوارکردن و نصب LOGO و یک ماژول دیجیتال را روی ریل مشاهده می کنید :

1 – ابتدا LOGO را روی ریل قرار دهید .

2 – سپس آن را به سمت پایین فشار داده تا کاملا در جای خود قرار گرفته و زائده اسلاید مانند در جای خود قرار گیرد .

3 – برای اتصال ماژول دیجیتال ، پوشش قسمت اتصال دهنده را بردارید .

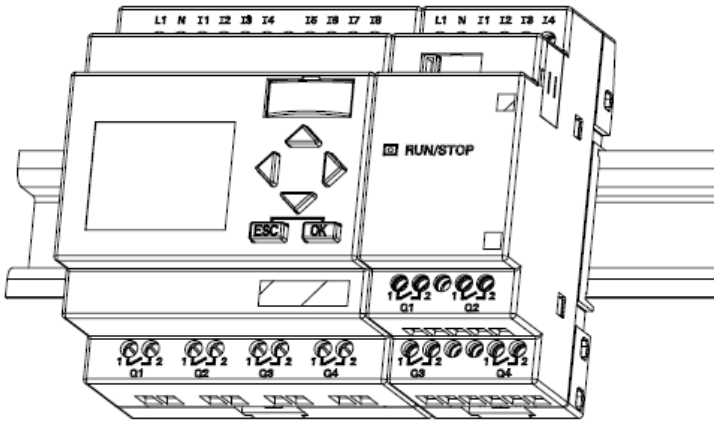
4 – ماژول دیجیتال را در طرف راست LOGO روی ریل قرار دهید .



5 – ماژول را بطرف چپ روی ریل حرکت دهید تا کاملا به LOGO بچسبد .

6 – با استفاده از یک پیچ گوشی ، زائده ماژول دیجیتال را بطرف محل اتصال دهنده LOGO فشار دهید تا کاملا این ارتباط بطور صحیح انجام گیرد .

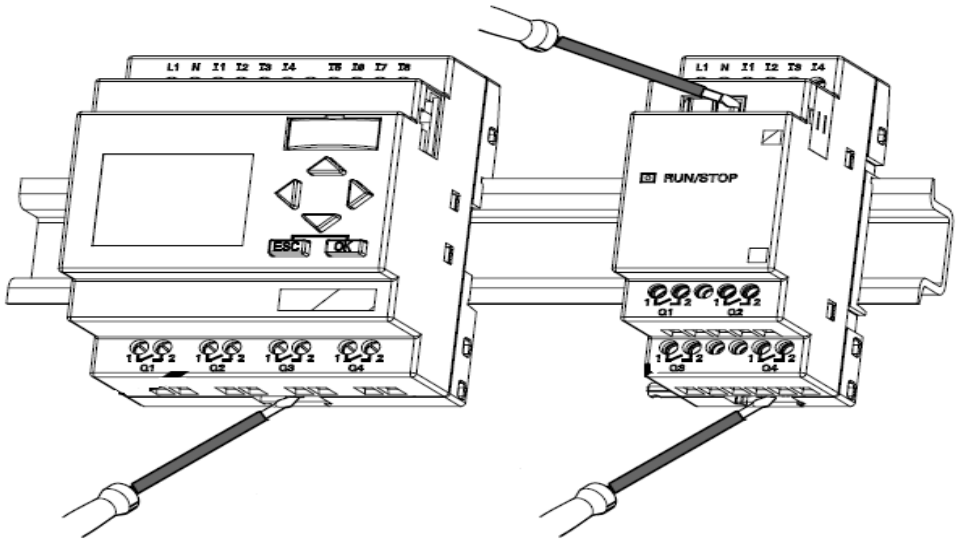
7 - هرگاه بخواهیم یک ماژول توسعه دیگری را به این مجموعه اضافه کنید ، مراحل 3 تا 6 را برای نصب ماژول انجام دهید .



توجه : رابط اتصال ماژول های اضافی در آخرین ماژول می بایست با استفاده از کاور مخصوص پوشانده شود .

طریقه جدا کردن LOGO از روی ریل :

برای جداکردن LOGO از روی ریل اگر فقط یک دستگاه روی ریل نصب شده باشد ، در آنصورت یک پیچ گوشی را داخل روزنه پایین LOGO قرار دهید و آن زائده را بطرف پایین فشار داده و بطرف بیرون حرکت دهید . سپس LOGO را بچرخانید تا از روی ریل جدا شده و آن را خارج کنید . برای جدا کردن ماژول وصل شده به LOGO با استفاده از یک پیچ گوشی ، کشوی ارتباط دهنده را بطرف پایین فشار داده و سپس آن را بطرف راست حرکت دهید تا آزاد شود . ماژول را بطرف راست کشیده تا یک فاصله مناسب بین ماژول و LOGO ایجاد شود . سپس یک پیچ گوشی داخل روزنه پایین ماژول کرده و آن را به طرف پایین فشار داده و بطرف بیرون حرکت دهید . ماژول توسعه را بچرخانید تا از روی ریل جدا شود .

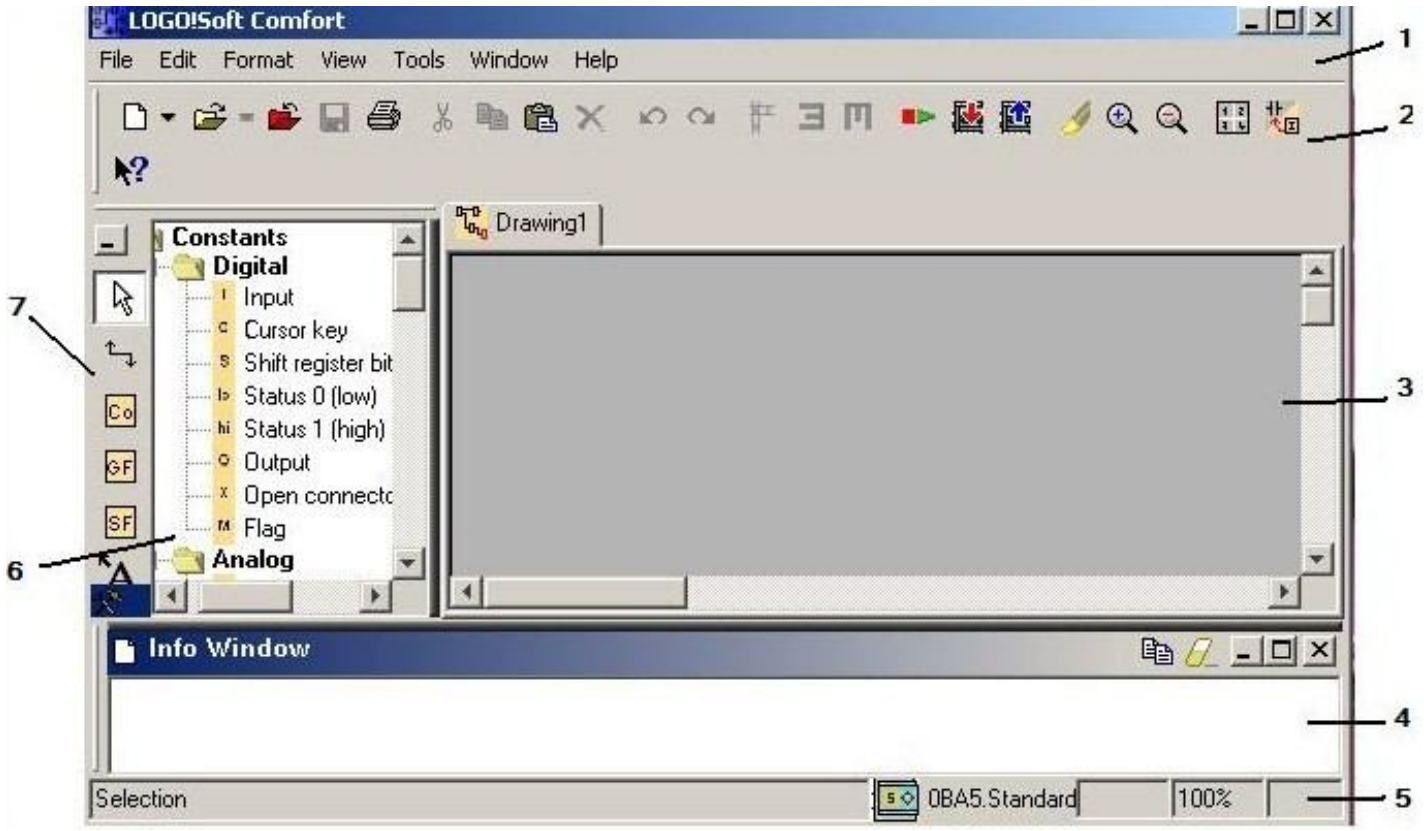


نرم افزار LOGO :

نرم افزار LOGO! Soft Comfort نرم افزاری جهت فراهم سازی محیط برنامه نویسی جهت کار با مینی PLCهای LOGO می باشد . از آنجا که برنامه نویسی از طریق صفحه کلید تعبیه شده بر روی سخت افزار ، کاری وقتگیر بوده و همچنین نوشتن برنامه های حجیم با صفحه کلید مزبور بسیار دشوار می نماید ، لذا شرکت سازنده ، نرم افزار موردنظر را که محیطی مناسب جهت نوشتن و ویرایش برنامه را در اختیار کاربر قرار و همچنین امکان تست برنامه های مربوط به پروژه های مختلف را فراهم می آورد ، ارائه کرده است . لذا ضروری است کاربر پس از شناخت سخت افزار سیستم، شناخت کاملی نیز از نرم افزار موردنظر بدست آورده تا بتواند با بکارگیری آن پروژه های موردنظر را پیاده سازی نماید . ویرایش جدید نرم افزار LOGO نسبت به نسخه های قبلی دارای توابع و دستورات جدیدی می باشد که برنامه نویسی را ساده تر و قابلیت های کار با دستگاه را افزایش می دهد . بطور کلی در محیط نرم افزار LOGO نوشتن برنامه به دو صورت نردبانی یا فلوچارتی صورت می گیرد که انتخاب هریک به نظر کاربر بستگی دارد . همچنین می توان برنامه نوشته شده را توسط دستور تبدیل به روش دیگری تبدیل کرد بدون اینکه مشکلی در عملکرد برنامه نوشته شده پیش آید .

محیط نرم افزار :

محیط برنامه نویسی LOGO در شکل نمایش داده شده است . شما می توانید به سادگی و البته بطور کامل از امکانات در این محیط استفاده کنید. امکانات لازم جهت برنامه نویسی بطور کامل در این محیط وجود دارد و آیکن اتصالات منطقی ، ابزار ترسیم و مرتب کردن آنها بطور کامل در این محیط وجود دارد .



قسمت های مختلف محیط نرم افزار LOGO عبارتند از :

1 – نوار منو

2 – نوار ابزار استاندارد

3 – رابط برنامه نویسی

4 – جعبه اطلاعات

5 – نوار وضعیت

6 – اتصالات و اتصال دهنده ها

7 – ابزار برنامه نویسی

نوار وضعیت :

نوار وضعیت به دو قسمت تقسیم می شود که در داخل آن پنج بخش مجزا وجود دارد که دارای اطلاعات مفیدی در مورد برنامه شما می باشد .

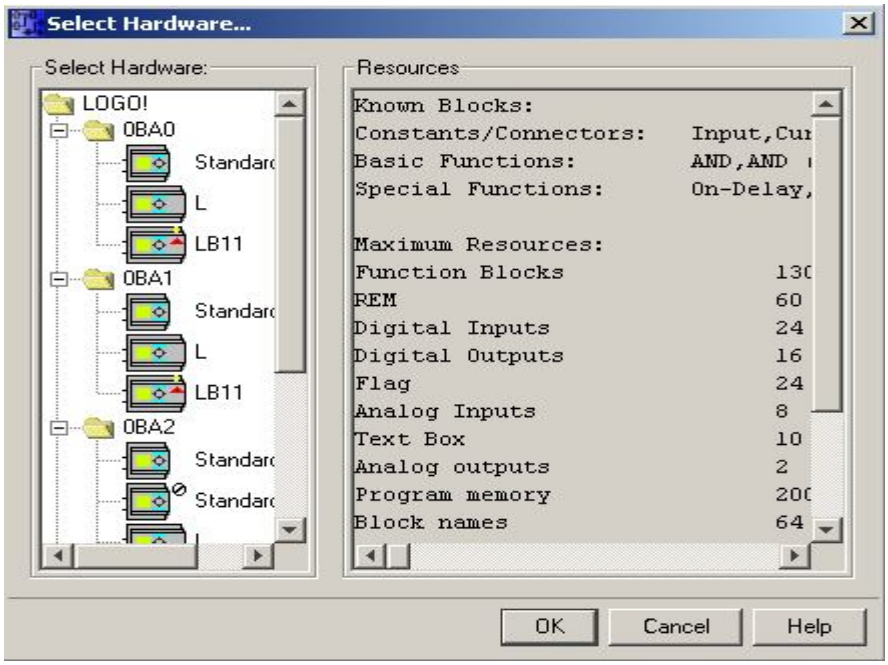
1 – پنجره اطلاعات Info Window :

پنجره اطلاعات موضوعاتی از قبیل پیام های خطای ایجاد شده در ابتدای شبیه سازی ، تاریخ و زمان پیام و نام برنامه مدار برای هر پیام ایجاد شده را نمایش می دهد . برای فراخوانی و نمایش اطلاعات در پنجره اطلاعات می توانید از کلید F2 استفاده کنید . همچنین برای باز و بسته کردن این پنجره می توانید از کلید F4 استفاده کنید . برای حرکت دادن این پنجره نیز می توانید آن را بوسیله موس انتخاب کرده و سپس کشیده و رها کنید و یا می توانید آن را خارج از محیط کار نرم افزار قرار دهید که در اینصورت به شکل یک پنجره جدا باز خواهد شد . برای ویرایش پیامها در پنجره اطلاعات می توان پیام های موردنظر را انتخاب کرده و سپس آنها را حذف کرد و یا توضیحاتی به دلخواه در آن قسمت نوشت . همچنین می توان از تمام یا قسمتی از اطلاعات داخل پنجره توسط  کپی گرفت و در محیط دیگری مثل word فراخوانی کرده و آن را ذخیره کرد . همچنین با استفاده از  نیز می توان اطلاعات پنجره را پاک کرد .

2 – جعبه اطلاعات :

این قسمت ابزار استفاده شده جاری را نمایش می دهد .

3 – انتخاب سخت افزار :



این قسمت نوع دستگاه LOGO انتخاب شده را نمایش می دهد . اگر شما بخواهید مشخصات LOGO را تغییر دهید می توانید بر روی شکل LOGO دوبار کلیک کنید و سپس تغییرات لازم را از منوی نمایش داده شده انجام دهید .

این منو به دو قسمت تقسیم می شود که در سمت چپ آن مدل های مختلف سخت افزار LOGO نمایش داده شده که برای انتخاب هریک کافیست روی مدل موردنظر کلیک کنید . در قسمت راست این منو امکانات و دستورات قابل پشتیبانی توسط دستگاه انتخاب شده نمایش داده می شود .

4 – فاکتور بزرگنمایی :

در این قسمت وضعیت بزرگ نمایی صفحه مشخص شده است .

5 – این قسمت صفحه مدار برنامه جاری را نمایش می دهد .

نوار ابزار :

امکاناتی که نوار ابزار در اختیار ما قرار می دهد دستوراتی هستند که بیشترین مورد استفاده را در برنامه نویسی دارند . این دستورات از طریق منوها نیز قابل دسترسی می باشند و شامل نوار ابزار استاندارد ، نوار ابزار برنامه نویسی و نوار ابزار سیمیلاتور می باشند .

نوار ابزار استاندارد شامل دستوراتی می باشد که سرعت عملکرد برنامه نویسی را افزایش می دهند .

گزینه هایی که نوار ابزار استاندارد را شامل می شوند و البته از منوها نیز قابل دسترسی هستند ، شامل دستورات زیر هستند که دستورات بیان شده زیر مهمترین دستورات و پرکاربردترین آنها می باشد .

- Align Automatically تنظیم خودکار : 

با انتخاب این گزینه دستورات و توابع خودکار بصورت افقی و یا عمودی تا حدودی در امتداد یکدیگر قرار می گیرند . برای فعال شدن این گزینه ابتدا باید توابعی که می بایست منظم شوند را انتخاب کنید و پس از استفاده از این گزینه محیط کار بطور خودکار منظم خواهد شد .

- Align Vertical آرایش بلوک های انتخاب شده بصورت عمودی : 

برای ردیف کردن بلوک های تابع انتخاب شده بصورت عمودی می توانید از این گزینه استفاده کنید .

- Align Horizontal آرایش بلوک های انتخاب شده بصورت افقی : 

برای ردیف کردن بلوک های تابع انتخاب شده بصورت افقی می توانید از این گزینه استفاده کنید .

- Switch Logo! Mode : 

این گزینه برای Run یا اجرا و Stop یا توقف عملیات LOGO می باشد .

- PC > LOGO : 

این گزینه برای دانلود کردن برنامه کامل شده بر روی سخت افزار LOGO مورد استفاده قرار می گیرد .

- LOGO > PC : 

برای انتقال برنامه از سخت افزار LOGO به کامپیوتر می توانید از این گزینه استفاده کنید .

- Select Line انتخاب خط : 

با انتخاب این گزینه در صورتیکه قبلا خطی را انتخاب کرده باشید محل اتصال دو سر اتصال یا خط رابط بین بلوک ها نمایش داده خواهد شد . از این گزینه می توانید در دیاگرام های پیچیده جهت جلوگیری از اشتباه استفاده کنید .

- Zoom in & Zoom out : 

برای بزرگ یا کوچک کردن صفحه کار می توانید از این گزینه ها استفاده کنید .

- Convert LAD<>FBD تبدیل زبان : 

این گزینه برای تبدیل برنامه نوشته شده از حالت LAD به FBD و برعکس مورد استفاده قرار می گیرد .

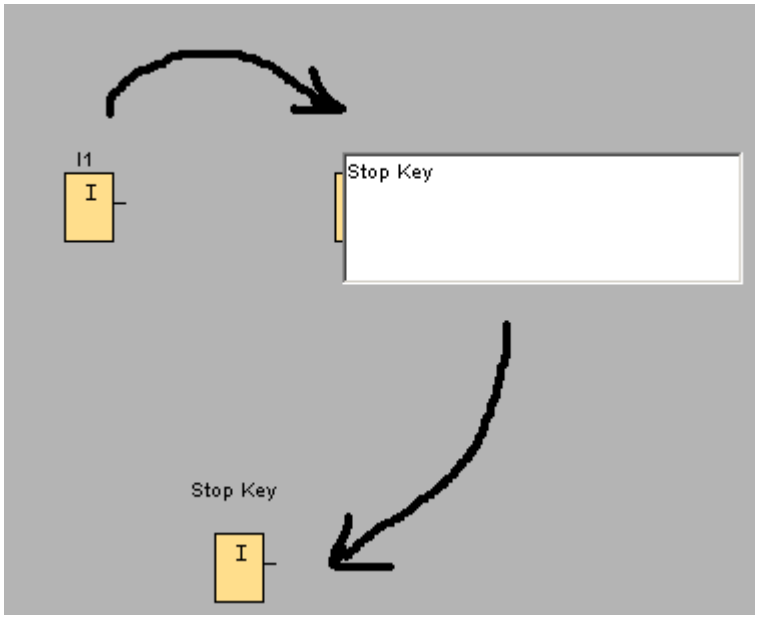
جعبه ابزار برنامه نویسی :

- ابزار انتخاب : 

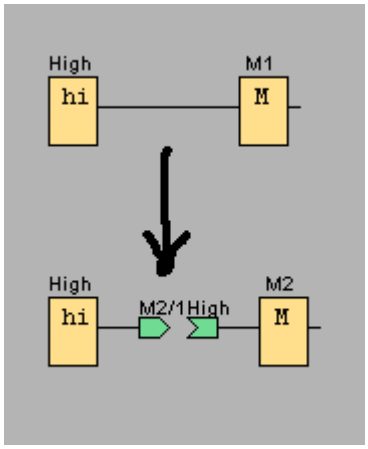
این ابزار برای انتخاب کردن و حرکت دادن بلوک های تابع ، خط های اتصال دهنده و توضیحات مورد استفاده قرار می گیرد . برای انتخاب بلوکی بصورت جداگانه می توانید بر روی آن کلیک کرده و آن را انتخاب کنید . برای انتخاب کردن چند بلوک می توانید یک مستطیل بوسیله موس در روی آن ایجاد کنید . همچنین می توانید از این گزینه برای جابجا کردن خطوط اتصال دهنده استفاده کنید .

- ابزار نوشتن : 

برای وارد کردن متن در صفحه ترسیم یا بر روی بلوک ، ابتدا آیکن متن را انتخاب کرده و در محل موردنظر کلیک کنید . در این صورت اگر بر روی صفحه کار کلیک کنید ، پنجره ای باز می شود که شما می توانید متن موردنظر خود را وارد کنید . بعد از وارد کردن متن و برای تثبیت شدن متن روی صفحه ترسیم کلیک کنید .



- ابزار اتصال  :



برای کامل کردن مدار باید بلوک ها بوسیله خطوط اتصال به یکدیگر متصل شوند تا ارتباط بین توابع برقرار شود . برای انجام دادن اینکار شما می توانید موس را حرکت داده و روی ترمینال یک بلوک قرار و فشار دهید و کلید چپ موس را نگه دارید و بطرف ترمینال بلوک مربوطه بکشید و سپس آن را رها کنید ، در اینصورت دو بلوک بهم متصل می شوند .

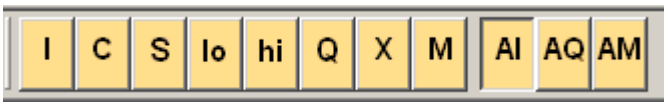
- ابزار قطع و وصل  :

این ابزار برای قطع اتصال بین دو بلوک استفاده می شود . با انتخاب این گزینه و در صورت انتخاب اتصالات بین دو بلوک ، باعث قطع شدن ارتباط بین دو بلوک می شود . از این گزینه همچنین می توانید برای اتصال خطوط قطع شده بهم یا به مرجع مشخص شده استفاده کنید . در صورتیکه بر روی ترمینالی که با علامت های  نمایش داده شده کلیک کنید ، این ترمینال به ترمینال مرجع متصل خواهد شد .

- اتصال دهنده ها  :

هرگاه شما بخواهید از بلوک های ورودی ، بلوک های خروجی ، تعیین حافظه یا ثبات ها برای ترسیم بر روی صفحه استفاده کنید ، می توانید این ابزار را انتخاب کنید . امکاناتی که این قسمت در اختیار شما می گذارد همانند اشکال زیر است که در دو حالت نردبانی و بلوکی نمایش داده شده است .

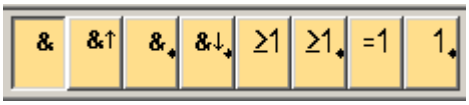
در حالت نردبانی



در حالت بلوکی

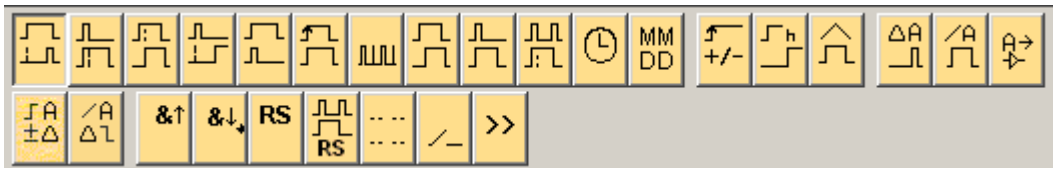
- توابع پایه : 

اگر شما بخواهید از بلوک های منطقی برای ترسیم استفاده کنید ، آیکن های این قسمت برای این منظور فراهم است . در این قسمت شما می توانید از دستورات منطقی استفاده کنید . ذکر این نکته ضروری است که این دستورات مخصوص حالت بلوکی می باشند و در حالت نردبانی آیکن قسمت توابع پایه غیرفعال می باشد .



- توابع ویژه : 

این بلوک برای دستیابی به توابع ویژه می باشد که جهت فعال کردن آن می توانید روی آیکن موردنظر کلیک کنید . امکاناتی که این قسمت در اختیار شما می گذارد ابزاری مانند تایمرها ، زمان سنج ها و غیره می باشد . این توابع در دو حالت نردبانی و بلوکی مشترک است ، تنها تفاوتی که بین این دو حالت وجود دارد اینست که در حالت نردبانی دو دستور AND با لبه بالارونده و NAND با لبه پایین رونده نسبت به حالت بلوکی به توابع ویژه اضافه شده است .



- سیمیلاتور : 

هرگاه شما ابزار شبیه سازی را انتخاب کنید ، نوار ابزار سیمیلاتور برای نظارت و کنترل کردن ورودی ها و خروجی ها در پایین صفحه کار ظاهر می شود . در این حالت با استفاده از این نوار ابزار شما می توانید عملکرد برنامه خود را آزمایش کنید . وقتی که گزینه سیمیلاتور را انتخاب می کنید خط های اتصال دهنده بلوک ها آبی رنگ می شوند ، حال اگر آیکن های مربوط به ورودی را فشار دهید در صورت درست بودن مدار ، نتیجه را روی همین نوار ابزار مشاهده خواهید کرد . همچنین رنگ خطوط اتصال دهنده بلوک های فعال شده به رنگ قرمز تغییر پیدا می کنند .

- Online Test : 

این گزینه جهت تست همزمان برنامه نوشته شده روی سخت افزار ، مورد استفاده قرار می گیرد .

نوار ابزار سیمیلاتور :

این نوار ابزار که با انتخاب گزینه مربوط به سیمیلاتور فعال می شود ، دارای امکاناتی از قبیل کنترل عملکرد سوئیچ های ورودی ، کنترل و قطع و وصل منبع تغذیه ، مانیتورینگ خروجی ، کنترل عملکرد سیمیلاتور و قسمت کنترل زمان می باشد که در ذیل مهمترین بخش های این نوار توضیح داده خواهد شد .

- نمایش ورودی ها :

ورودی ها بصورت سمبل کلید یا سوئیچ و شستی نمایش داده شده اند که نمای زیر نمایش این سمبل ها می باشد .

نمایش وضعیت کلید I1 در حالتی که باز است .

نمایش وضعیت کلید I1 در حالتی که بسته است .

نکته : تفاوت شاستی با سوئیچ :

یک شاستی زمانی فعال است که در حالت پایین نگه داشته شود ، اگر این شاستی رها شود اتصال برق نیز قطع می گردد اما پس از فشار دادن سوئیچ، سوئیچ در حالت فعال باقی می ماند تا زمانیکه دوباره فشار داده شود .

- نمایش ورودی های آنالوگ و فرکانس :

نمایش ورودی های آنالوگ و ورودی فرکانس کمی با هم تفاوت دارند . در اینجا شما می توانید مقدار ولتاژ آنالوگ یا فرکانس را با استفاده از یک نوار متغیر تنظیم کنید .

- کنترل منبع تغذیه :

این قسمت برای قطع و وصل کردن تغذیه همه ورودی ها کاربرد دارد . یکی از کاربردهای این قسمت ریست کردن برنامه در حال اجرا می باشد .


منبع تغذیه در حالت قطع


منبع تغذیه در حالت وصل

- نمایش خروجی ها :

وضعیت یک خروجی یا نشانگر حافظه بوسیله یک سمبل حباب نورانی نمایش داده می شود .

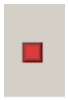
نمایش وضعیت خروجی در حالتیکه Q1 فعال باشد .

نمایش وضعیت خروجی در حالتیکه Q2 فعال نباشد .

کنترل سیمیلاتور :

این قسمت شامل شروع ، نگه داشتن و خاتمه عملیات کار سیمیلاتور می باشد .

شروع به کار سیمیلاتور



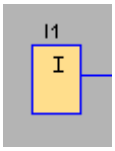
توقف عملیات سیمیلاتور

نگه داشتن عملیات سیمیلاتور

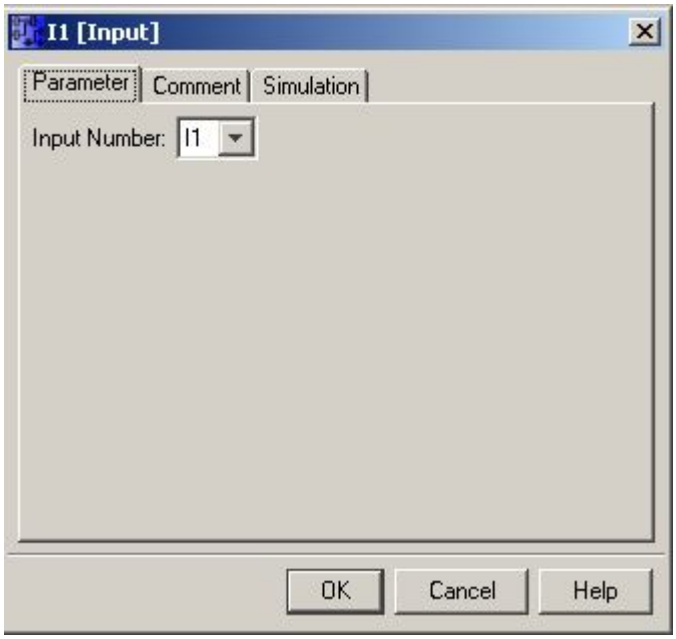
عناصر و اتصال دهنده های مرجع :

نوارهای ابزار که در این قسمت معرفی خواهد شد شامل توابع ثابت ، توابع پایه و توابع ویژه می باشند . با انتخاب این نوارهای ابزار انواع ورودی ، خروجی و ابزارهای دیگر در دسترس شما قرار می گیرد و این امکان را به شما می دهد تا در صورت نیاز در هر قسمت از برنامه به سادگی از آنها استفاده کنید . البته ذکر این نکته ضروری می باشد که در بعضی از قسمت ها تفاوت هایی در تعداد حالت های انتخابی برای حالت نردبانی و بلوکی وجود دارد .

- ورودی ها Inputs :

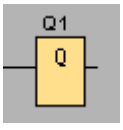


بلوک های ورودی نشان دهنده ترمینال های ورودی بر روی سخت افزار LOGO می باشند . هرگاه بخواهید پارامتر ورودی را انتخاب کنید یا اینکه توضیحاتی در مورد آن بنویسید ، می توانید بر روی بلوک ورودی دوبار کلیک کرده که در این حال پنجره ای باز خواهد شد که می توانید پارامتر و توضیحات مربوط به ورودی ها را تعیین کنید .



همانطور که در شکل مقابل مشاهده می کنید ، پنجره تنظیمات ورودی شامل سه قسمت Parameter ، Comment و Simulation می باشد که قسمت Parameter مربوط به تعیین شماره ورودی می شود . قسمت Comment برای درج توضیحات در مورد بلوک ورودی مورد استفاده قرار می گیرد . توضیحاتی که در این قسمت نوشته می شوند بعد از تایید ، در بالای بلوک ورودی قرار خواهند گرفت . قسمت Simulation مربوط به حالت های مختلف ورودی است که در این قسمت می توان از یک ورودی به چهار حالت کلیدی ، سوئیچ لحظه ای نرمال باز ، سوئیچ لحظه ای نرمال بسته و فرکانسی استفاده کرد .

- خروجی ها Outputs :

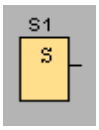


بلوک های خروجی نشان دهنده ترمینال های خروجی بر روی سخت افزار LOGO می باشند . البته قابل ذکر است که خروجی ها متناسب با نسخه های مختلف LOGO متغیر هستند .

- سطوح سیگنال ثابت Fixed Signal Level :

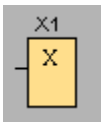
سطح سیگنال بلوک hi در خروجی بصورت یک یا فعال می باشد و می توان از این ویژگی برای ورودی بلوک های دیگر استفاده کرد . بلوک ها را برای نشان دادن سطح ولتاژ ثابت نمی توان استفاده کرد . سطح سیگنال بلوک lo در خروجی بصورت صفر می باشد یعنی خروجی این بلوک دائما غیرفعال یا صفر می باشد .

- بیت های شیفت رجیستر Shift Register Bit :



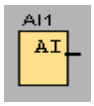
دستگاه LOGO بیت های شیفت رجیستر را از S1 تا S8 فراهم می سازد که فقط برای ویژگی خواندن در مدار برنامه تعیین شده اند و نیز فقط بصورت ورودی در برنامه استفاده می گردد . این بیت ها تنها با توابع ویژه شیفت رجیستر کنترل می شوند .

- اتصال یا رابط آزاد Open Connectors :



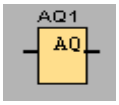
با استفاده از این دستور می توان خروجی یک تابع بکار نرفته را مسدود کرد تا با خروجی بلوک های دیگر اشتباه گرفته نشود .

- ورودی های آنالوگ Analog Inputs :



مدل های LOGO با علامت های 12/24RC ، 12/24RCO و 24RC پردازش و اجرای سیگنال های آنالوگ را ممکن می سازد و تا دو ورودی می تواند توسط این مدل ها استفاده گردد .

- خروجی های آنالوگ Analog Outputs :

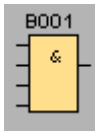


در نرم افزار LOGO تنها دو خروجی آنالوگ با نام های AQ1 و AQ2 قابل دسترسی می باشد . شما می توانید یک مقدار را توسط خروجی آنالوگ یا پرچم آنالوگ نمایش دهید .

- پرچم ها Flags :

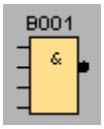
خروجی بلوک های پرچم همان سیگنال ورودی و یک بیت از حافظه می باشند که می توان از آن برای ذخیره کردن اطلاعات بصورت بیتی استفاده کرد . این بیت معادل کنتاکتور کمکی در مدارهای فرمان می باشد . پرچم M8 در اولین سیکل برنامه کاربر نشانده می شود . پس می توان این پرچم را بصورت یک پرچم شروع کننده یا Start Up در برنامه جاری استفاده کرد . همچنین بعد از اجرای اولین سیکل برنامه پرچم M8 بازنشانده می شود و در سیکل های بعدی هیچ واکنشی ندارد .

- گیت AND :



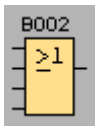
با این گیت در بخش های ابتدایی آشنا شدید و حال در اینجا گیت تابع AND را بکار می بریم . خروجی تابع AND فقط زمانی فعال است که همه ورودی ها فعال باشند . در این گیت پایه آزاد حکم یک یا فعال را دارد .

- گیت NAND :



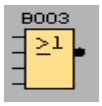
خروجی این گیت تنها زمانی در حالت صفر می باشد که همه ورودی ها در حالت یک یا فعال باشند ، در غیر اینصورت خروجی گیت در حالت یک می باشد . اگر یک پایه ورودی متصل نشده یا آزاد باشد آن پایه بطور خودکار فعال می باشد .

- گیت OR :



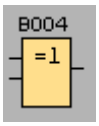
خروجی تابع OR زمانی در حالت فعال قرار می گیرد که حداقل یکی از ورودی ها دارای حالت یک باشد . اگر یکی از پایه های ورودی این بلوک متصل نشده باشد ، آن پایه بطور خودکار در وضعیت صفر قرار می گیرد .

- گیت NOR :



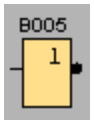
خروجی تابع NOR زمانی در وضعیت یک می باشد که همه ورودی ها حالت صفر را داشته باشند . در این تابع پایه آزاد حکم صفر را دارد .

- گیت XOR :



خروجی این تابع زمانی در وضعیت یک قرار می گیرد که تعداد فردی از ورودی ها دارای وضعیت یک باشد . پایه آزاد در این تابع حکم صفر را دارد .

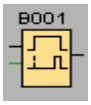
- گیت NOT :



هرگاه ورودی در وضعیت صفر باشد ، خروجی در وضعیت یک می باشد و بالعکس . اگر پایه ورودی این بلوک متصل نشده باشد در آن صورت آن پایه حکم یک را دارد .

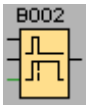
تایمرها Timers :

- تایمر تاخیر در وصل On-Delay :



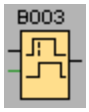
در این تابع خروجی زمانی فعال می شود که زمان تعریف شده Ta بعد از فعال شدن پایه Trg سپری شده باشد . نحوه عملکرد تابع فوق به این صورت می باشد که وقتی ورودی Trg از وضعیت صفر به یک تغییر پیدا کند ، زمان تنظیم شده Ta اجرا می شود . اگر ورودی Trg برای مدت طولانی در وضعیت یک باقی بماند ، در آن صورت بعد از سپری شدن زمان تنظیم شده Ta خروجی در وضعیت یک قرار می گیرد و اگر ورودی Trg قبل از سپری شدن زمان Ta در وضعیت صفر قرار گیرد ، زمان Ta نیز در وضعیت صفر قرار می گیرد . در صورت قطع شدن برق ، مدت سپری شده ریست می شود . خروجی این تابع تا زمانیکه ورودی Trg فعال باشد روشن است و با غیرفعال شدن ورودی Trg خروجی نیز غیرفعال می شود .

- تایمر تاخیر در قطع Off-Delay :

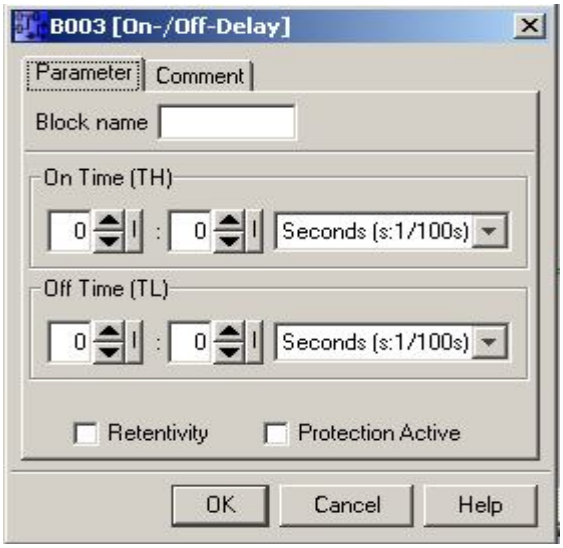


در این تابع خروجی با لبه بالارونده ورودی Trg روشن می شود و تا وقتیکه زمان تنظیم شده Ta سپری گردد ، روشن می ماند . در این تابع اگر ورودی Trg به وضعیت یک سوئیچ شود ، خروجی فورا روشن می شود و زمان تنظیم شده Ta با لبه پایین رونده ورودی Trg فعال می شود و بعد از سپری شدن زمان تنظیم شده ، خروجی Q خاموش می شود . اگر ورودی Trg بعد از فعال شدن زمان Ta به وضعیت یک سوئیچ شود و دوباره به وضعیت صفر سوئیچ شود ، زمان Ta نیز دوباره از اول شروع به شمردن می کند . ورودی R برای ریست زمان Ta و خروجی مورد استفاده قرار می گیرد . رله راه پله یک تایمر تاخیر در قطع می باشد .

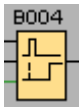
- تایمر تاخیر در وصل و قطع On/Off-Delay :



این تابع یک تایمر تاخیر در وصل و تاخیر در قطع می باشد . در این تابع خروجی بعد از سپری شدن اولین زمان TH روشن می شود و بعد از سپری شدن دومین زمان تنظیم شده TL ، خاموش می شود . وقتیکه وضعیت ورودی Trg از صفر به یک تغییر پیدا می کند ، زمان تنظیم شده TH اجرا می شود و اگر در حین شمارش زمان TH ، ورودی Trg دوباره به وضعیت صفر و سپس یک سوئیچ شود ، زمان TH ریست می شود . وقتیکه وضعیت ورودی Trg از یک به صفر تغییر پیدا کند ، زمان تنظیم شده TL اجرا می شود و بعد از سپری شدن این زمان خروجی خاموش می شود . البته قابل ذکر است که بعد از سپری شدن زمان TH خروجی Q روشن می شود . اگر قبل از اینکه زمان TL سپری شود ، ورودی Trg به حالت یک سوئیچ شود ، در این حالت زمان TL ریست می شود . در صورت قطع برق ، زمان TL و TH ریست می شود . برای تنظیم زمان TH و TL از قسمت تنظیم پارامترها در پنجره مشخصات این تابع اقدام می کنیم . برای دستیابی به این پنجره و تنظیم زمان TH و TL کافیست روی بلوک تایمر دوبار کلیک کنید سپس پنجره مشخصات بصورت روبرو باز خواهد شد .

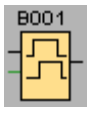


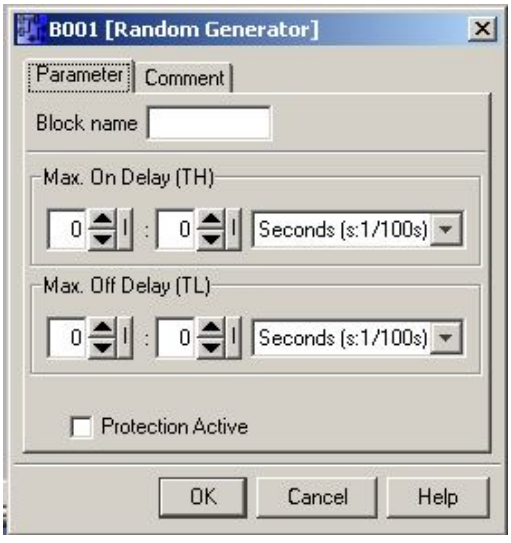
- تایمر تاخیر در وصل مستقل از ورودی Retentive On-Delay :



در این تایمر زمان Ta با لبه بالارونده ورودی Trg فعال شده و پس از سپری شدن زمان تنظیم شده ، خروجی Q روشن می شود . وقتی وضعیت ورودی Trg از صفر به یک تغییر پیدا می کند ، شمارش زمان Ta آغاز می شود . اگر مدت سپری شده به زمان T برسد ، خروجی Q در وضعیت یک قرار می گیرد . اگر ورودی Trg دوباره سوئیچ شود ، بر روی زمان Ta تاثیری ندارد . همچنین خروجی و Ta به حالت صفر بازنشانده نمی شوند تا وقتیکه ورودی R در وضعیت یک یا فعال قرار گیرد . اگر در حین شمارش زمان برق قطع گردد ، با دوباره وصل شدن برق زمان Ta ریست می شود .

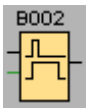
- مولد تصادفی Random Generator :



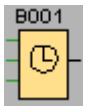


این تابع یک تایمر ترکیبی تاخیر در وصل و قطع است که زمان قطع و وصل را بطور تصادفی از تنظیم های مشخص شده برای آن تعیین می کند . اگر وضعیت ورودی En از صفر به یک تغییر پیدا کند ، یک زمان تصادفی بین صفر و TH تعیین می شود . اگر ورودی En برای مدت زمان تاخیر در روشن در وضعیت یک باقی بماند ، خروجی بعد از سپری شدن زمان تاخیر در روشن ، در وضعیت یک قرار می گیرد . اگر وضعیت ورودی En قبل از سپری شدن زمان تاخیر در روشن به وضعیت صفر سوئیچ شود ، تایمر ریست می شود . اگر وضعیت ورودی En دوباره به وضعیت صفر سوئیچ شود ، یک زمان تصادفی بین صفر و TH تعیین شده و شروع به اجرا کردن زمان می کند . اگر ورودی En به حالت صفر تغییر وضعیت دهد ، زمان تاخیر در خاموش اجرا می شود و در آن صورت خروجی بعد از پایان یافتن زمان تاخیر در خاموش در وضعیت صفر قرار می گیرد . اگر وضعیت

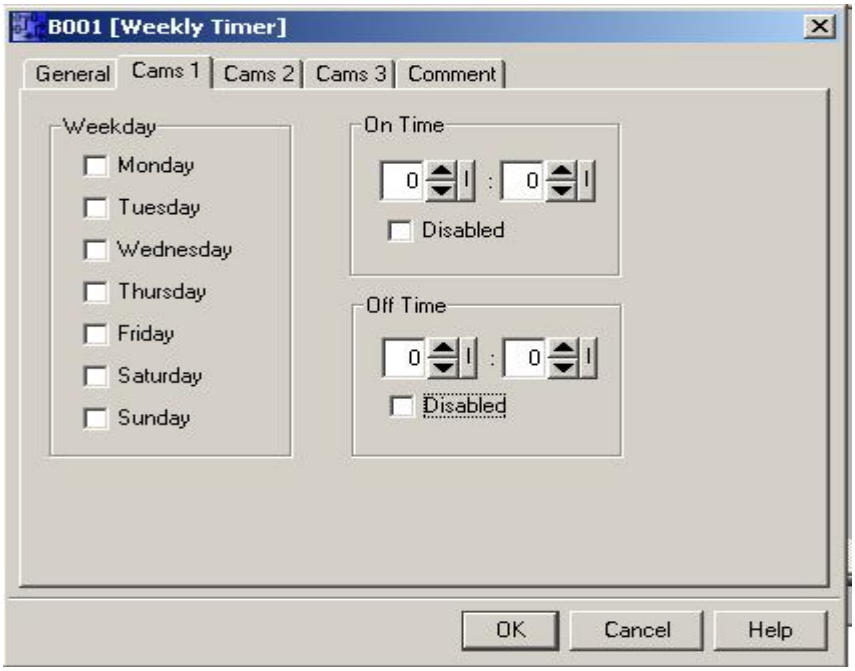
ورودی En قبل از سپری شدن زمان تاخیر در خاموش TL به وضعیت یک سوئیچ شود ، تایمر ریست می شود . برای تنظیم مولد تصادفی می توانید از پنجره تنظیمات با دوبار کلیک روی تایمر استفاده کنید .

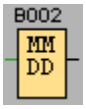
- کلید روشنایی راه پله Stairway Lighting Switch : 

خروجی با لبه بالارونده پالس ورودی Trg فعال می شود و بعد از سپری شدن زمان T که قابل تنظیم است ، خاموش می شود . اگر وضعیت ورودی Trg از صفر به یک تغییر پیدا کند ، خروجی فعال می شود و با لبه پایین رونده ورودی Trg زمان T شروع به شمردن می کند . این به آن معناست که بهتر است برای ورودی از یک شاسی استفاده شود . اگر شمارش زمان برابر مقدار تنظیم شده شود ، در آنصورت خروجی خاموش می شود . اگر ورودی Trg دوباره از وضعیت صفر به یک و پس از وضعیت یک به صفر تغییر کند و یا برق قطع شود ، در اینصورت زمان Ta بازنشانده می شود .

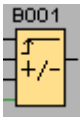
- تایمر هفتگی Weekly Timer : 

این تابع یک تایمر هفتگی می باشد که در ساعات و روزهای قابل تنظیم ، خروجی آن فعال خواهد شد . برای اینکار باید از پنجره تنظیم ، مشخصات هفته های فعال را انتخاب کنید ، سپس تاریخ موردنظر را وارد کنید .

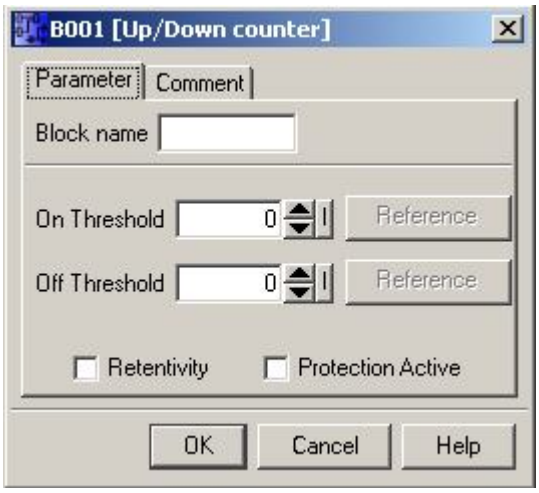


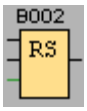
- تایمر سالیانه Yearly Timer : 

در این تابع خروجی توسط تاریخ روشن و خاموش می شود . یک تاریخ برای روشن شدن و یک تاریخ برای خاموش شدن از دوازده ماه تعیین می گردد . خروجی از یک تابع تنظیم شده روشن می شود تا تاریخ بعدی که نشان دهنده خاموش شدن خروجی می باشد .

- شمارنده بالا / پایین شمار Up\Down Counter : 

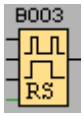
این تابع یک شمارنده می باشد که می تواند پالس هایی که به ورودی آن توسط سنسورها و غیره وارد می شود را شمارش و در مقدار تنظیم شده ، خروجی را فعال و غیرفعال کند . خروجی این شمارنده زمانی فعال می شود که مقدار شمارش بیشتر یا برابر مقدار تنظیم شده شود . جهت شمارش را می توانید با ورودی Dir تنظیم کنید . زمانیکه این ورودی فعال است ، شمارش بطرف پایین می باشد و زمانیکه فعال نیست ، شمارش بطرف بالا می باشد .



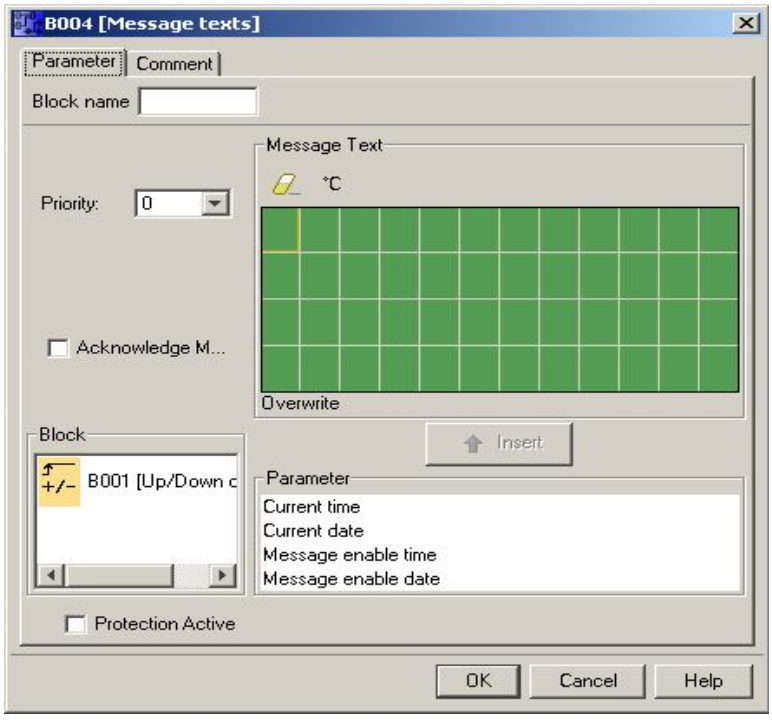
- رله نگهدارنده Latching Relay : 

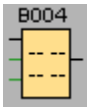
ورودی S خروجی Q را فعال می کند و ورودی R خروجی Q را غیرفعال می کند . یک رله نگهدارنده بصورت یک سلول حافظه دودویی می باشد . روشن بودن یک خروجی بستگی به وضعیت ورودی ها و وضعیت قبلی خروجی دارد . هرگاه ورودی های S و R هر دو باهم در وضعیت یک قرار بگیرند ، در آنصورت اولویت با پایه R می باشد . اگر بر روی بلوک RS دوبار کلیک کنید و در پنجره مشخصات بازشده گزینه Retentivity را انتخاب کنید ، در اینصورت وقتی برق قطع می شود با وصل مجدد برق وضعیت رله نگهدارنده در همان حالت قبلی که قبل از قطع شدن برق داشت ، باقی می ماند .

S	R	Q
0	0	بدون تغییر
0	1	0
1	0	1
1	1	0

- رله پالسی Pulse Relay : 

در این رله خروجی بوسیله یک پالس کوتاه ورودی ، ست و ریست می شود . این رله همانند کلید استپ استارت عمل می کند . هرگاه وضعیت ورودی Trg از صفر به یک تغییر پیدا کند ، خروجی Q نسبت به وضعیت ورودی تغییر وضعیت می دهد . در این رله خروجی با لبه بالارونده پالس اول روشن می شود و با لبه بالارونده پالس دوم خاموش می شود .



- Message Text : 

یک متن پیام تنظیم شده در مدت اجرا نمایش داده می شود . طریقه عملکرد تابع به این صورت است که هرگاه وضعیت ورودی En از صفر به یک تغییر پیدا کند ، متن پیام تنظیم شده در حالت اجرا نمایش داده می شود . اگر وضعیت ورودی En از یک به صفر تغییر پیدا کند در اینصورت متن پیام نمایش داده نمی شود .

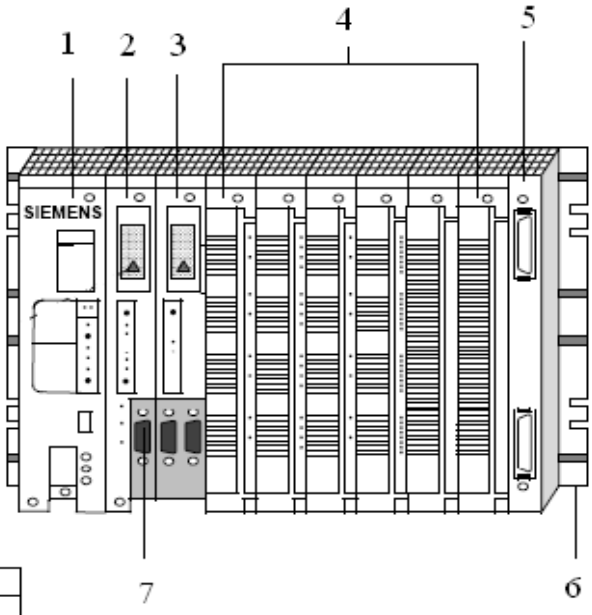
فصل پنجم

PLC
S5

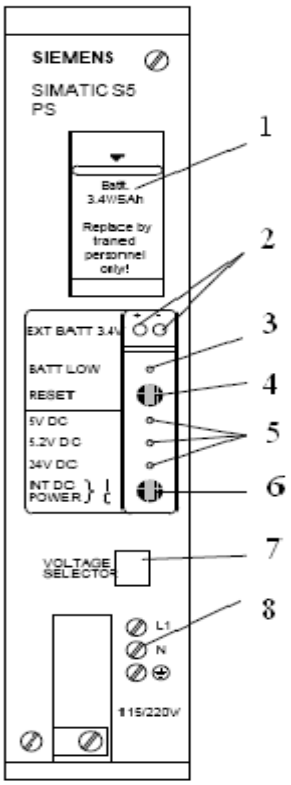
در این فصل در مورد PLC_S5 صحبت خواهد شد و سخت افزار و نرم افزار آن معرفی می شود . همچنین روش برنامه نویسی S5 را بیان خواهیم کرد .

PLC های خانواده S5 از شرکت زیمنس آلمان یکی از قدیمی ترین کنترل کننده های برنامه پذیر در حد وسیع بوده که در کشور ما نیز در صنایع گوناگون بصورت گسترده استفاده گردیده است . البته در سال های اخیر با توجه به تنوع تولید PLC و ساخت نمونه های با ظرفیت نرم افزاری بیشتر ، کمتر شاهد بکارگیری این نوع PLC در ساخت و راه اندازی سیستم های جدید صنعتی می باشیم . ولیکن از آنجا که به روز نمودن سیستم های کنترل قدیمی هزینه بسیار بالایی را به صنعت وارد می نماید ، لذا امروزه نیز PLC های خانواده S5 در صنایع ما مشغول بکار بوده و لذا هنوز نیاز به فراگیری نحوه کار با این نوع PLC احساس می گردد . با توجه به تعدد نوع PLC این خانواده و شباهت بسیار زیاد این نمونه ها ، در این مجموعه تاکید بر S5_115U خواهد بود .

اجزای سیستم :



1- ماژول منبع تغذیه PS
2- واحد پردازش مرکزی CPU
3- پردازنده های ارتباطی CP
4- ماژول های ورودی و خروجی I/O
5- ماژول های گسترش IM
6- بدنه اصلی Rack
7- ترمینال سریال



منبع تغذیه Power Suply Module :

ماژول منبع تغذیه ، ولتاژ تغذیه ورودی را به یک ولتاژ مورد نیاز برای قسمت های مختلف داخلی تبدیل می نماید . ولتاژهای مورد نیاز برای این سیستم : 230VAC ، 115VAC ، 24VDC می باشند . با توجه به نوع ماژول مورد استفاده می توان به جریان های ماکزیمم خروجی برابر با 3 ، 7 و 15 آمپر دست یافت . برای منابع تغذیه کمتر از 7 آمپر نیازی به استفاده از فن نمی باشد . ماژول تغذیه علاوه بر ایجاد ولتاژهای مورد نیاز PLC ، یک ولتاژ پشتیبان برای RAM سیستم با استفاده از یک باتری لیتیم ایجاد می نماید . یک باتری از نوع لیتیم ، حافظه برنامه ، پرچم های پایدار داخلی و مقادیر تایمرها و شمارنده ها را در زمان قطع تغذیه حفظ می نماید . بر روی ماژول تغذیه ، یک LED برای نمایش وضعیت باتری تعبیه شده ، که در صورت عدم وجود و یا خرابی باتری روشن می گردد . در شکل روبرو قسمت های مختلف یک منبع تغذیه نمایش داده شده است .

1- محفظه قرارگرفتن باطری Backup
2- ترمینال جهت ولتاژهای Backup خارجی ، این ولتاژ در زمان تغییر باتری که ولتاژ تغذیه نیز وجود ندارد ، اعمال می گردد .

3- نمایشگر خرابی باتری : LED موردنظر در سه حالت زیر روشن می گردد :

الف : زمانی که باتری وجود نداشته باشد .

ب : باتری بطور نادرست نصب شده باشد .

ج : ولتاژ باتری به کمتر از 2.8 ولت رسیده باشد .

4- کلید Reset : این کلید را می توان برای تصدیق سیگنال خطای بوجود آمده باتری در زمان نصب باتری ، بکار برد .

5- نمایشگر ولتاژهای عمل کننده : +5V ولتاژ تغذیه ماژول های ورودی و خروجی

+5.2V ولتاژ تغذیه پروگرامر

+24V ولتاژ تغذیه رابط انتقال سریال

6- کلید On / Off : با قطع این کلید بدون قطع شدن خط تغذیه ، ولتاژهای عمل کننده قطع می گردند .

7- انتخاب کننده ولتاژ خط 115VAC یا 230VAC

8- ترمینال های اتصال خط تغذیه

ولتاژ باتری های Back Up معمولا بین 2.8 تا 3.6 ولت می باشد .

واحد پردازش مرکزی :

با انتخاب نوع CPU می توان به قابلیت های نرم افزاری بیشتر ، حجم حافظه بیشتر و در نتیجه سرعت اجرای بالاتری دست یافت . قسمت های مختلف موجود بر روی CPU برای انجام اعمال زیر در نظر گرفته شده اند :

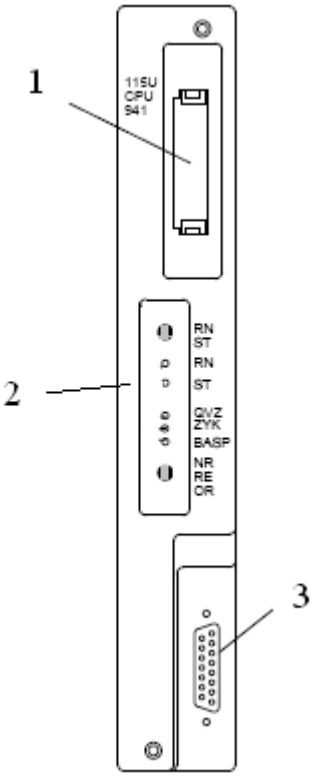
1- قراردادن ماژول اضافی حافظه

2- اتصال یک برنامه ریز PG یا کامپیوتر شخصی یا یک پانل کاری اپراتور

3- اتصال شبکه محلی LAN

4- اتصال PLC های ساخته شده توسط سایر کارخانه ها

5- تنظیم مود عملکرد PLC



شکل روبرو نمای ظاهری ماژول CPU را نمایش می دهد .

1- محل قرارگرفتن ماژول حافظه خارجی

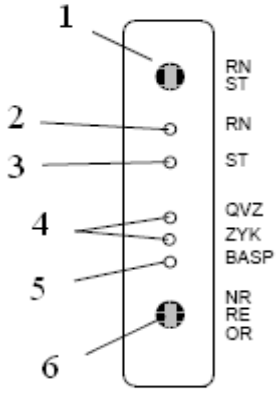
2- قسمت کنترل

3- ترمینال ارتباطی برای اتصال پروگرامر ، پانل ارتباطی کاربر یا ترمینال شبکه LAN

قسمت کنترل در ماژول CPU ، قسمت مهم این ماژول به حساب می آید زیرا این قسمت توسط LED های موجود بر روی خود ، وضعیت کاری CPU را نمایش می دهد . اگر خطا و مشکلی در کار CPU بوجود آید ، توسط پانل کنترل CPU مشخص می شود .

پانل کنترل :

پانل کنترل در شکل روبرو نمایش داده شده است و قسمتهای مختلف آن در ادامه بررسی می گردد .



1- سلکتور انتخاب حالت کاری Stop / Run

2- LED نمایش حالت Run

3- LED نمایش حالت Stop

4- LED نمایش حالت Error

5- LED نمایش ناتوانی خروجی ها

6- سوئیچ برای Restart سیستم

دو LED موجود بر روی پانل کنترل ، وضعیت CPU را نمایش می دهند . به جدول زیر توجه کنید :

وضعیت PLC	LED سبز	LED قرمز
CPU در حالت اجرای روتین راه اندازی سرد	ON	ON
حالت Stop	OFF	ON
حالت Start	ON	OFF
چک کردن برنامه بصورت پی در پی	OFF	OFF

حالات مختلف کاری PLC :

با قرار دادن سلکتور انتخاب حالت کاری در وضعیت Stop یا Run ، می توان سیستم را در حالت کاری اجرا یا توقف بکار گرفت . CPU بطور اتومات در تغییر وضعیت اجرا به توقف ، حالت Restart را اجرا می نماید . برنامه نوشته شده در حالت کاری Stop ، Scan نمی گردد . آخرین مقدار تایمرها ، شمارنده ها و پرچم ها در زمان رفتن به حالت Stop نگهداری می گردند . در حالت Stop همچنین ماژول های خروجی غیرفعال گشته و LED مربوط به نمایش ناتوانی خروجی روشن می گردد . پس از اینکه سیستم عامل مربوط به CPU ، برنامه های کاری مربوط به راه اندازی را به انجام رساند ، سیکل Scan و اجرای برنامه آغاز می گردد .

در حالت Run ، ورودی های موجود بر روی ماژول ورودی ، بطور متناوب و در یک سیکل مشخص خوانده شده و در PII ذخیره می شود . همچنین در این حال مقادیر پرچم های داخلی Update می گردند . برنامه کنترلی نوشته شده توسط کاربر ، این اطلاعات را به همراه مقادیر جاری تایمرها و شمارنده ها پردازش می نماید . برنامه کنترل نوشته شده توسط کاربر شامل یکسری جملات نوشته شده پی در پی می باشد . پردازنده ، این جملات را یک به یک از حافظه برنامه فراخوانده و پس از تجزیه و تحلیل ، آنها را اجرا می نماید . نتیجه این پردازش به مکان PIO انتقال می یابد .

PII : جدول پردازش ورودی در ناحیه حافظه CPU قرار دارد که وضعیت تمام سیگنال های ورودی در آنجا ذخیره می شود . قبل از اجرای برنامه ، CPU تمام ورودی ها را بررسی می کند و در قسمتی از حافظه بنام PII ذخیره می کند . در حین اجرای برنامه CPU بجای ورودی ها به این حافظه مراجعه می کند .

PIO : جدول پردازش خروجی شامل مقادیر خروجی حاصل از اجرای برنامه است . این مقادیر در انتهای هر دوره به خروجی واقعی فرستاده می شود . هرگاه در حین برنامه یک مقدار برای خروجی بدست آید ، در این قسمت حافظه نگهداری می شود و در هنگام اجرای برنامه ، CPU بجای خروجی ها به این حافظه مراجعه می کند .

زمان پاسخ دهی :

زمان پاسخ دهی عبارتست از مدت زمان بین تحریک یک ورودی تا پاسخ دهی خروجی متناظر در طی یک برنامه که ورودی بطور مستقیم و از طریق نرم افزار به خروجی متصل گشته است بعبارت دیگر زمان پاسخ دهی مدت زمانی است که طول می کشد تا PLC تمام برنامه های کاربر را پویش نماید و در این مدت تغییرات بوجود آمده در ورودی ها ، وارد مکان تصویر ورودی نمی گردد و خروجی ها نیز به حالتی که در پویش قبلی بودند باقی می مانند . این مدت زمان بطور معمول مجموعی از عوامل زیر می باشد :

1- تاخیر ذاتی ماژول های ورودی
2- مدت زمان Scan برنامه

برنامه نویسی PLC_S5 :

یک برنامه کنترل ، مجموعه دستورالعمل هایی است که به PLC فرمان هایی جهت کنترل پروسه صادر می نماید . لذا باید این برنامه به زبانی خاص و مطابق با دستورات قابل درک برای PLC نوشته شود . زبان برنامه نویسی که خانواده S5 از آن استفاده می نماید ، Step 5 نامیده می شود .

زبان برنامه نویسی Step 5 از سه گروه دستورالعمل مختلف تشکیل شده است :

- **دستورات اصلی (Basic Operation) :** این دستورات در بلوک های سازماندهی (OB) ، بلوک های برنامه (PB) ، بلوک های ترتیبی(SB) و بلوک های عملگر (FB) استفاده می گردند . به جز عملگرهای + (Addition) ، -- (Subtraction) و عملگرهای

سازماندهی ، سایر دستورالعمل های اصلی می توانند به عنوان ورودی و خروجی در برنامه های نوشته شده به روش های STL ، CSF ، LAD بکار گرفته شوند .

- **دستورات تکمیلی (Supplementary Operation) :** این دستورات شامل عملگرهای پیچیده ، عملگرهای جایگزینی ، تست ، انتقال و تبدیل می باشند .
- **عملگرهای سیستمی :** جهت دسترسی مستقیم به سیستم عامل می توان از آنها استفاده نمود .

فلگ ها یا پرچم ها :

محل هایی از حافظه هستند که جهت نگهداری وضعیت برخی نتایج و یا خروجی ها استفاده می شوند . فلگ ها به دو صورت پایدار (Retentive) و ناپایدار (Non_Retentive) موجود می باشند . در نمونه های پایدار ، اطلاعات موجود با قطع تغذیه از بین نمی روند ، در حالیکه در نمونه های ناپایدار با قطع تغذیه اطلاعات موجود از بین خواهد رفت . بطور معمول تعداد فلگ های پایدار و ناپایدار در یک PLC برابر می باشند . فلگ ها حافظه های موقتی هستند که CPU هنگام اجرای برنامه از آنها برای یادداشت نتایج منطقی و یا حالت سیگنال ها استفاده می نماید .

در آدرس دهی ورودی ها ، خروجی ها و فلگ ها ، ابتدا نوع عملوند را با بکار بردن حروف اختصاری مشخص کرده و سپس شماره بایت و پس از یک علامت dot ، شماره بیت ذکر می گردد . مثلا عبارت I 5.3 بیانگر بیت سوم از بایت پنجم ورودی می باشد . چون در این PLC خطوط انتقال اطلاعات هشت بیتی می باشند لذا ورودی ها ، خروجی ها و فلگ ها در دسته های هشت بیتی سازماندهی می شوند . در جدول زیر حروف اختصاری عملوندها بیان شده است :

حروف اختصاری	نوع عملوند
I	Input
Q	Output
F	Flag
T	Timer
C	Counter

عملگرهای عمومی :

این عملگرها را در فصل پیش در مورد Logo کاملا معرفی کردیم و در اینجا این عملگرها را به زبان STL معرفی خواهیم کرد :

عملگر AND :

در این مثال خروجی Q 3.5 زمانی یک خواهد شد که دو ورودی گیت AND یک باشند درغیر اینصورت خروجی صفر خواهد بود .

```
A   I1.1
A   I1.3
=   Q3.5
```

عملگر OR :

در این مثال خروجی Q 3.5 زمانی یک خواهد شد که حداقل یکی از دو ورودی گیت OR یک باشند در غیراینصورت خروجی صفر خواهد بود .

```
O   I1.1
O   I1.3
=   Q3.5
```

عملگر NOT_AND :

در این مثال خروجی Q 3.5 زمانی یک خواهد شد که I1.1 = 1 بوده و I1.6 = 0 شود .

```
A    I1.1
AN   I1.6
=    Q3.5
```

عملگر فلیپ فلاپ : این عملگر برای Set و Reset کردن فلگ ها و خروجی ها استفاده می گردد .

در این مثال در صورت فعال شدن I2.7 ، خروجی Q3.5 یک خواهد شد . با تغییر وضعیت ورودی فوق الذکر از یک به صفر تغییری در خروجی حاصل نخواهد شد . با یک شدن I1.4 خروجی صفر خواهد شد .

```
A    I2.7
S    Q3.5
A    I1.4
R    Q3.5
```

نکته : دستور S با دستور = فرق می کند ، اگر Set کنیم باید یک فکری هم برای Reset کنیم .

دستور NOP 0 :

این دستور مختص PLC های زیمنس می باشد . از این دستور زمانی استفاده می نماییم که بخواهیم از یک خروجی استفاده ننماییم . به عبارت دیگر این دستور ، یک دستور جایگزین بوده که هیچ عمل خاصی را انجام نمی دهد . از دیگر کاربردهای این دستور ، استفاده در خطوط برنامه STL به جای دستوراتی است که حذف می گردند . این عمل این اجازه را به نرم افزار S5 می دهد که بتوان خطوط برنامه STL را به سایر روش ها از جمله LAD یا CSF تبدیل نماید .

کاربرد پرانتزها در برنامه نویسی به روش STL :

در مواردی که دستور AND قبل از دستور OR قرار گرفته باشد ، نیازی به استفاده از پرانتز در برنامه نویسی به روش STL نیست ولی در مواردی که در آن دستور OR قبل از دستور AND قرار گرفته است ، استفاده از پرانتز الزامی می باشد . برای روشن شدن مطالب عنوان شده ، به مثال زیر توجه نمایید :

در این مثال عدم استفاده از پرانتز در حالتیکه دستور AND قبل از دستور OR قرار گرفته باشد ، مورد بررسی قرار می گیرد .

```
A    I1.0
A    I2.3
O
A    I3.1
A    I2.2
=    Q3.5
```

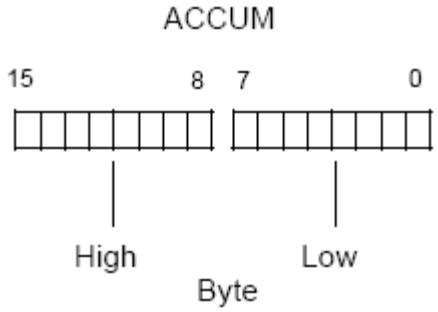
در این مثال استفاده از پرانتز در حالتیکه دستور OR قبل از دستور AND قرار گرفته باشد ، نشان داده می شود .

```
A(
O    I2.1
O    I4.1
)
A(
O    I2.4
O    I3.0
)
=    Q2.3
```

در هر برنامه فقط هفت پرانتز می توان باز کرد .

انبارک یا آکومولاتور :

جهت تبادل اطلاعات بین قسمت های مختلف نیاز به یک حافظه میانی است که عمل تبادل اطلاعات از طریق آن صورت پذیرد ، به این حافظه میانی آکومولاتور گفته می شود . آکومولاتور یا انباره یک ثبات 32 بیتی می باشد که جهت ذخیره و یا انتقال اعداد بکار می رود . در واقع آکومولاتور یک ثبات آزاد جهت کاربردهای عمومی می باشد . در حافظه PLC دو آکومولاتور به نام های AC1 و AC2 برای استفاده موجود است .
در شکل زیر ساختار آکومولاتور نمایش داده شده است :



دستور Load :

در این دستور اطلاعات از یک نقطه حافظه یا از منطقه تصویر ورودی ها به آکومولاتور یک منتقل می گردد . مقدار قبلی موجود در آکومولاتور یک به آکومولاتور دو منتقل می گردد . در این حال آخرین مقدار آکومولاتور دو از بین خواهد رفت .

دستور Transfer :

در این دستور اطلاعات از آکومولاتور یک ، به یک نقطه حافظه و یا منطقه تصویر خروجی منتقل می گردد . مقدار موجود در آکومولاتور یک در اثر عمل انتقال تغییری نخواهد کرد . در اثر انتقال اطلاعات به منطقه خروجی دیجیتال ، مقدار قبلی PIQ بصورت خودکار Update می گردد .

L IB12 ⟶ خواندن بایت ورودی 12
T QB36 ⟶ انتقال اطلاعات به بایت خروجی 36

نکته : بایت ورودی در زبان S5 با IBx مشخص می گردد که x شماره بایت ورودی می باشد . بعنوان مثال بایت ورودی یک که بر روی کارت های ورودی قرار دارد ، شامل هشت بیت از I1.0 تا I1.7 می باشد . یک کارت ورودی دیجیتال دارای چهار بایت ورودی می باشد .
نکته : بایت خروجی در زبان S5 با QBx مشخص می گردد که x شماره بایت خروجی می باشد . بعنوان مثال بایت خروجی چهار که بر روی کارت های خروجی قرار دارد ، شامل هشت بیت از Q4.0 تا Q4.7 می باشد . یک کارت خروجی دیجیتال دارای چهار بایت خروجی می باشد .

توابع زمانی یا تایمرها :

در بسیاری از مدارات یک خط تولید یا فرآیند ، احتیاج به مواردی همچون موارد زیر پیدا می کنیم :
1 – ایجاد وقفه در عملیات برای مدت زمان معین
2 – ایجاد فاصله زمانی مشخص بین انجام دو عملیات
3 – ایجاد زمان مناسب جهت انجام یک فرآیند
بنابراین به عنصری احتیاج پیدا می نماییم که بتواند مجموعه اعمال فوق را برای ما انجام دهد . به عنصری که این عمل را انجام می دهد ، تایمر یا رله زمانی گفته می شود . تایمرها از نظر نوع عملکرد دارای انواع مختلفی می باشند که در برنامه نویسی PLC به پنج گروه استاندارد تقسیم می شوند .

این پنج گروه بعبارت زیر است :

1- تایمر پله ای Pulse Timer (SP)
2- تایمر پله ای گسترده Extended Pulse Timer (SE)
3- تایمر با تاخیر در وصل On-Delay Timer (SR یا SD)
4- تایمر با تاخیر در قطع Off-Delay Timer (SF)
5- تایمر با تاخیر در وصل پایدار Stored On-Delay Timer (SS)

بارگذاری زمان تایمر :

به هنگام Start تایمر مقدار موجود در آکومولاتور یک بعنوان پارامتر یا زمان کاری تایمر در نظر گرفته می شود . لذا قبل از Start تایمر ، باید زمان موردنظر را در آکومولاتور یک بارگذاری نماییم . زمان موردنظر می تواند به یکی از اشکال زیر در نظر گرفته شود :

- KT : مقادیر ثابت برای زمان تایمر
- DW : مقادیر قرار داده شده در کلمات اطلاعاتی
- IW : کلمه ورودی
- QW : کلمه خروجی
- FW : کلمه پرچم یا فلگ

به مثال زیر که بارگذاری یک مقدار ثابت را نمایش می دهد ، توجه کنید :

L KT 040.2

L : دستور بارگذاری
KT : تعریف مقدار ثابت
40 : ضریب قابل تعریف برای تایمر
2 : ضریب زمانی
در مثال بیان شده ، ضریب بکار رفته می تواند مقادیری مطابق جدول زیر داشته باشد :

00 = 0	01 = 1	10 = 2	11 = 3
0.01s	0.1s	1s	10s

با توجه به مطالب فوق الذکر ، می توان زمان تخصیص یافته در مثال را محاسبه کرد :

Time = 40 × 1s = 40s

نکته : کمترین زمان ممکن قابل اندازه گیری توسط یک تایمر 0.01 ثانیه و حداکثر زمان قابل اندازه گیری 9990 ثانیه می باشد . در صورت نیاز به زمان های بیشتر می توان از ترکیب چند تایمر استاده نمود .

نحوه خواندن زمان جاری تایمرها :

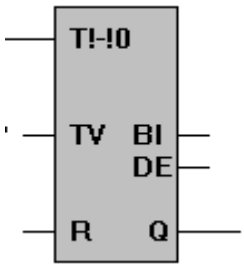
در اکثر PLC ها این قابلیت وجود دارد که بتوان زمان جاری تایمرها را خوانده و به خروجی های PLC انتقال داد . دستور L Tx برای خواندن زمان جاری تایمر x استفاده می شود . این مقدار را می توان برای استفاده نمایشگرهای دیجیتال خروجی بکار برد .

در زبان برنامه نویسی S5 ، برای ایجاد یک تایمر به روش STL از مجموعه جملات زیر استفاده می شود :

```
A     I1.0
L     KT005.2
SP    T1
L     T1
T     QW10
```

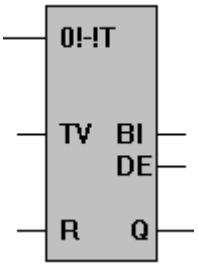
ورودی I1.0 بعنوان Start تایمر استفاده می گردد . این سه خط باید بترتیب ذکرشده ، استفاده گردند

خواندن زمان جاری تایمر و انتقال آن به کلمه خروجی 10



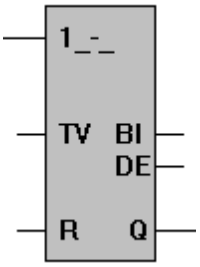
تایمر تاخیر در وصل SR یا SD :

این نوع تایمر در حقیقت مشابه رله زمانی در مدارهای فرمان الکتریکی که از کنتاکت باز آن استفاده می شود ، بوده و با یک شدن ورودی ، شروع به زمان گیری می نماید و بعد از زمان تاخیر ، خروجی تایمر یک خواهد شد . در این نوع تایمر عرض پالس ورودی باید از زمان تاخیر بیشتر باشد زیرا به محض صفر شدن ورودی ، خروجی تایمر نیز صفر می گردد .



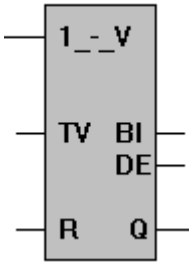
تایمر تاخیر در قطع SF :

در این نوع تایمر با یک شدن ورودی ، خروجی یک شده و بمحض صفر شدن ، ورودی زمان گیری آغاز و بعد از زمان تاخیر ، خروجی را صفرخواهد کرد . این تایمر در حقیقت با لبه نزولی شروع به زمان گیری و قطع مدار می کند .



تایمر پالس SP :

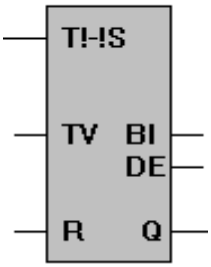
در این نوع تایمر با یک شدن ورودی ، خروجی بلافاصله یک شده و زمان گیری را آغاز و بعد از زمان موردنظر به شرط یک بودن ورودی ، خروجی را صفر می کند . بنابراین در این نوع تایمر ، عرض پالس ورودی باید بزرگتر از زمان تاخیر باشد .



تایمر توسعه یافته SE :

این تایمر عملکردی مشابه با تایمر SP دارد با این تفاوت که بعد از اعمال ورودی و یک شدن ، تایمر شروع به زمان گیری کرده و اگر ورودی صفر گردد ، تایمر به کار خود ادامه می دهد تا آنکه بعد از زمان تاخیر ، خروجی اش صفر گردد . بنابراین این تایمر با لبه صعودی فعال شده و عرض پالس ورودی تاثیری در عملکرد آن ندارد .

تایمر تاخیر در وصل پایدار SS :



این تایمر که ترکیبی از تایمر نوع SP و SE است بدین صورت عمل می کند که با یک شدن ورودی شروع به زمان گیری می نماید و بعد از زمان تاخیر ، خروجی آن یک می شود و به همین حالت باقی می ماند تا زمانیکه با یک ورودی تریگر ریست شود . بنابراین این تایمر با لبه صعودی پالس ورودی فعال شده و به پهنای پالس ورودی بستگی ندارد . همچنین اگر ورودی همچنان یک باقی بماند و تایمر ریست گردد ، مجدد حتما باید لبه صعودی به آن داده شود .

نکاتی در مورد تایمر :

- هر تایمر دارای سه ورودی و سه خروجی می باشد .
- حرف L در روش STL نشان از بارکردن زمان به تایمر است .
- هرچه ضریب کوچکتر انتخاب شود ، دقت تایمر بیشتر خواهد بود .
- از طریق پایه TV ، زمان تایمر بارگذاری می گردد .
- از طریق پایه DE ، می توان زمان باقیمانده نسبت به TV را به عددی در مبنای BCD به یک مکان حافظه منتقل نمود .
- از طریق پایه BI ، می توان زمان باقیمانده نسبت به TV را به صورت باینری به یک مکان حافظه منتقل نمود .

شمارنده ها :

در عملکرد بعضی از فرآیندها نیاز به مواردی داریم که اقلام یا اجناسی بدقت شمرده شوند یا یک فرآیند به دفعات مشخصی تکرار شود . عنصری که عمل شمارش را می تواند برای ما انجام دهد ، کانتر (شمارنده) نامیده می شود . شمارنده موجود در این نوع PLC شمارنده صعودی و نزولی بوده که مقدار جاری شمارنده نیز در اختیار کاربر قرار می گیرد .

انواع شمارنده :

- شمارنده بالارونده (صعودی) CU
- شمارنده پایین رونده (نزولی) CD

از آنجا که پس از Start ، شمارنده موجود در آکومولاتور یک بعنوان حد نهایی شمارش وارد شمارنده می گردد لذا می بایست قبل از Start شمارنده ، مقدار موردنظر را در یکی از فرم های زیر در آکومولاتور یک بار نمود :

KC : مقادیر ثابت برای شمارنده
IW : کلمه ورودی
QW : کلمه خروجی
FW : کلمه فلگ
DW : مقادیر قرار داده شده در کلمات اطلاعاتی

مثال زیر روش بارگذاری یک مقدار ثابت را در شمارنده نمایش می دهد :

L KC 37

L : دستور بارگذاری
KC : تعریف مقدار ثابت
37 : مقدار شمارش

نحوه خواندن مقدار جاری شمارنده ها :

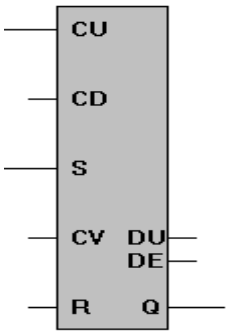
در اکثر PLC ها این قابلیت وجود دارد که بتوان مقدار جاری شمارنده ها را خوانده و به خروجی های PLC انتقال داد . دستور L Cx برای خواندن زمان شمارنده x استفاده می شود .

```
A    I1.0
CU   C4
```

ورودی I1.0 بعنوان پالس ورودی شمارش صعودی استفاده می گردد . هر لبه مثبت اعمالی به این ورودی باعث افزایش یک شماره در مقدار داخل شمارنده می گردد .

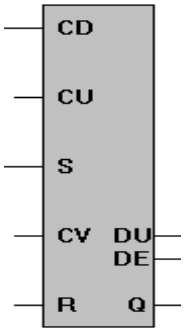
```
L    C4
T    QW10
```

خواندن مقدار جاری شمارنده و انتقال آن به کلمه خروجی 10



شمارنده بالاشمار CU :

در شمارنده بالاشمار با بارگذاری مبنای عددی موردنظر ، شمارش کانتر با Set کردن کانتر آغاز و ورودی CU که بصورت پالس است ، از عدد مبنا به بالا شروع به اضافه نمودن می کند تا حداکثر عدد شمارش که 999 می باشد ، ادامه می دهد. خروجی Q این نوع شمارنده با Set شدن شمارنده یک شده و همچنان یک باقی می ماند تا زمانیکه کانتر بالاشمار را Reset نماییم ، آنگاه خروجی صفر خواهد شد .



شمارنده پایین شمار CD :

در شمارنده پایین شمار ، با بارگذاری مبنای عددی موردنظر و Set کردن کانتر ، خروجی بلافاصله یک می شود و شمارش پالس های ورودی ، CD را از عدد مبنا به پایین کم می کند و بمحض رسیدن به عدد صفر ، خروجی بطور خودکار صفر می شود .

نکاتی در مورد شمارنده ها :

- در یک شمارنده می توان ورودی CD و CU را بطور همزمان استفاده نمود و هرگاه ورودی CU تحریک شود ، کانتر از عدد فعلی شمارنده کم می شود .
- از ترکیب شمارنده ها می توان محدوده شمارش را افزایش داد .
- مقدار اولیه ای که در شمارنده قرار می گیرد ، مبنای شمارش محسوب می شود . این مقدار از طریق پایه CV بارگذاری می شود .
- از طریق پایه DE ، مقدار CV بصورت BCD به مکان مورد نظر منتقل می شود .
- از طریق پایه DU(BI) ، مقدار CV بصورت باینری به مکان مورد نظر منتقل می شود .
- با فعال شدن پایه R ، مقدار CV و همچنین تمامی خروجی های کانتر صفر می شود .
- با فعال شدن پایه S ، مقدار اولیه شمارش کانتر از طریق پایه CV بارگذاری می شود .

مقایسه کننده ها :

در برخی موارد ، در مراحلی از انجام یک پروسه ، احتیاج به انجام عمل مقایسه بین دو کمیت شامل خروجی ، ورودی و فلگ خواهیم داشت که این عمل را توابع مقایسه ای یا مقایسه کننده ها برای ما انجام می دهند . این مقایسه کننده ها مقادیر موجود در دو آکومولاتور را مقایسه کرده و در اثر نتیجه مقایسه ، بیت RLO را مقداردهی می نماید . این دستورات مقدار داخلی آکومولاتورها را تحت تاثیر قرار نمی دهند .

عملگر	شرح عملگر
!= F	در این عملگر در صورتیکه AC1=AC2 باشد ، بیت RLO=1 می گردد
>< F	در این عملگر در صورتیکه مقدار دو آکومولاتور نامساوی باشد ، بیت RLO=1 می گردد
> F	در این عملگر در صورتیکه AC2>AC1 باشد ، بیت RLO=1 می گردد
>= F	در این عملگر در صورتیکه AC2>=AC1 باشد ، بیت RLO=1 می گردد
< F	در این عملگر در صورتیکه AC2<AC1 باشد ، بیت RLO=1 می گردد
<= F	در این عملگر در صورتیکه AC2<=AC1 باشد ، بیت RLO=1 می گردد

به مثال زیر توجه کنید :

```
L    IB0
L    IB2
!= F
=    Q0.0
```

عملکرد این دستورالعمل به گونه ای است که هر دو ورودی خود را بیت به بیت با هم مقایسه کرده و در صورت صحیح بودن عمل مقایسه ، خروجی یک را ظاهر خواهد نمود .

نکاتی در مورد مقایسه کننده ها :

- مقایسه گر می تواند دو بیت را نیز باهم مقایسه نماید و همچنین می توان اعداد ثابتی را به آن بار کرد تا عمل مقایسه را روی آنها انجام دهد .
- توجه داشته باشید که ورودی مقایسه گرها از یک نوع ، مثلا هر دو بایت یا هر دو Word باشند .
- در دستورات مقایسه ، برای بزرگتر یا کوچکتر بودن ، اولین عددی که بارگذاری می گردد در AC2 قرار می گیرد .

دستورات جمع و تفریق :

چنانچه در برنامه ای نیاز به جمع یا تفریق محتویات دو آکومولاتور داشته باشیم ، ابتدا این دو آکومولاتور را بوسیله دستور Load پر نموده و عمل جمع یا تفریق را مطابق زیر ، روی آن انجام و سپس نتیجه را به یک خروجی انتقال می دهیم .

تفریق دو آکومولاتور :

```
L    IB0
L    IB2
-F
T    QB0
```

جمع دو آکومولاتور :

```
L    IB0
L    IB2
+F
T    QB0
```

انواع بلوک های برنامه نویسی در PLC_S5 :

در نوشتن برنامه کنترل مربوط به فرآیندهای گسترده ، جهت جلوگیری از پراکندگی و در نتیجه از هم گسیختگی برنامه ، از روش برنامه نویسی ساخت یافته استفاده می شود . در این روش یک برنامه کنترل وسیع پس از تقسیم بندی به قسمت های کوچکتر ، در بلوک های مخصوصی نوشته می شوند و سپس ارتباطات مابین قسمت های مختلف برنامه از طریق یک برنامه سازماندهی انجام می پذیرد . برنامه های سازماندهی شده امکان عیب یابی برنامه را آسانتر کرده و بدلیل دسته بندی شدن برنامه به قسمت های مجزا ، دست یابی به اجزای مختلف برنامه در مدت زمان بسیار کوتاهی فراهم می گردد .

در PLC های خانواده S5 ، کل ساختار برنامه در قالب پنج نوع بلوک برنامه نویسی تعریف می گردد :

1 - بلوک های سازماندهی OB (Organization Block) :

از آنجا که در یک برنامه ساخت یافته نیازمند تعدادی بلوک اصلی برای سازمان دادن به برنامه کاربر و تعریف شرایط مختلف برای اجرای برنامه ها می باشیم ، لذا در این نوع PLC تعدادی بلوک سازماندهی جهت تعریف شرایط مختلف کاری در نظر گرفته شده اند . بلوک OB1 بدنه اصلی برنامه کاربر را شامل می گردد . به عبارت دیگر اولین خط از برنامه کاربر که می بایست مورد پردازش قرار گیرد ، در این بلوک نوشته می شود . از OB1 می توان برای سازمان دهی برنامه اصلی کاربر استفاده کرده و تمام پرش ها و شرط های لازم برای انجام سایر مراحل کاری را در این بلوک تعریف نمود .

2 – بلوک های برنامه PB (Program Block) :

PB ها شامل دستورات استفاده کننده جهت کنترل پروسه مربوطه می باشند . برنامه نویس پروسه کنترل خود را به قسمت های کوچکتر تقسیم کرده و هر قسمت را در یکی از بلوک های برنامه قرار می دهد . 256 بلوک برنامه از PB0 تا PB255 قابل دسترسی می باشد . در پایان هر بلوک برنامه ، یک دستور BE نمایانگر انتهای بلوک مربوطه می باشد .

3 – بلوک های ترتیبی SB (Sequence Block) :

یکی از موارد بسیار مهم در کنترل یک پروسه صنعتی ، استارت سیستم در حال توقف می باشد . در این موارد فعال شدن هر قسمت از پروسه با توجه به شرایط کاری خاصی انجام می پذیرد . در PLC خانواده S5 یک نوع از بلوک های برنامه نویسی بنام SB ها برای این منظور تعریف شده اند. در این بلوک ها تنها به شرط انجام و صحت نتیجه در بلوک های با شماره پایین تر ، بلوک با شماره بالاتر اجرا خواهد شد .

4 – بلوک های عملگر FB (Function Block) :

در کنترل برخی فرآیندها ، گاه لازم است توابعی بصورت مداوم و تکراری مورد استفاده قرار گیرند . بعنوان مثال ممکن است عمل ضرب دو عدد باینری در طی یک پروسه مرتبا مورد استفاده قرار گیرد . در چنین مواردی اگر بنا به تکرار برنامه ضرب در مکان های مختلف سطح برنامه کنترل باشد ، حجم برنامه بسیار زیاد شده و در نهایت بررسی و عیب یابی آن با مشکل مواجه خواهد شد . برای رفع این مشکل در برنامه نویسی مدرن از زیربرنامه ها استفاده می شود . بلوک های عملگر می توانند بصورت عباراتی همراه با متغیر قابل تعریف ، استفاده شده و یا تنها یک عمل ثابت را بدون پارامترهای قابل تنظیم بیان نمایند . در این PLC ، 256 بلوک عملگر از FB0 تا FB255 قابل تعریف می باشد .

هر FB از دو قسمت تشکیل شده است :

1 – سرخط بلوک (Block Header) که شامل نام و سایر مشخصات FB است .

2 – بدنه بلوک (Body Block) شامل توابع و دستوراتی است که باید در FB اجرا شود .

در تقسیم بندی کلی می توان FB ها را به دو دسته زیر تقسیم نمود :

1 – بلوک تابع ساز استاندارد (Standard FB) : این بلوک ها شامل توابعی هستند که در آنها اعمال منطقی نظیر ضرب ، جمع و ....تعریف شده است.

2 – بلوک تابع ساز انتسابی (Assignable FB) : در اجرای این نوع FB می توان عملوندها یعنی ورودی ها ، خروجی ها و ..... را در هر پروسه تعریف نمود و یا تغییر داد .

5 – بلوک های اطلاعاتی DB (Data Block) :

بلوک های اطلاعاتی برای ذخیره سازی داده های مورد استفاده در برنامه کاربر مورد استفاده قرار می گیرند . این اطلاعات که می توانند به فرمت های مختلف مورد استفاده قرار گیرند شامل پیغام ها ، آلارم ها و ..... می باشند که می توانند در طی پروسه مورد استفاده قرار گرفته یا بر

روی صفحات نمایشگر به نمایش درآیند . هر بلوک اطلاعاتی شامل تعدادی کلمه های اطلاعاتی می باشد که به آنها Data Word گفته شده و با علامت DW نمایش می دهند . DB های ایجاد شده را از لحاظ محل ذخیره سازی می توان به دو دسته کلی زیر تقسیم نمود :

1- DB هایی که حاوی پارامترهای ثابت فرآیند و یا خط تولید بوده ، اطلاعات آنها در EPROM یا EEPROM ذخیره می گردد .

2- DB هایی که حاوی اطلاعات موقتی بوده و برای مصارف کوتاه مدت و موقت استفاده می شوند. اینگونه DB ها در RAM ذخیره می گردد.

دستورات انجام عملیات دیجیتال :

این دستورات محتویات دو آکومولاتور را بیت به بیت ترکیب کرده و نتیجه را در آکومولاتور یک قرار می دهند .

عملگر	شرح عملگر
AW	انجام عمل AND منطقی بصورت بیت به بیت
OW	انجام عمل OR منطقی بصورت بیت به بیت
XOW	انجام عمل XOR منطقی بصورت بیت به بیت

دستورات افزایش و کاهش :

با استفاده از این دستورات می توان از عدد موجود در آکومولاتور ، حداکثر به اندازه 255 واحد کم و یا به آن افزود . میزان افزایش یا کاهش با توجه به عدد نوشته شده در مقابل دستور محاسبه می گردد .

عملگر	شرح عملگر
D	حداکثر 255 واحد معادل یک بایت از کلمه ورودی کم می کند
I	حداکثر 255 واحد معادل یک بایت به کلمه ورودی اضافه می کند

```
L   IW0
I   200
T   QW0
```

دستورات اعلام پایان برنامه :

در انتهای هر برنامه لازم است به نحوی به PLC اطلاع داده شود که برنامه به پایان رسیده است ، این عمل با استفاده از دستورات مربوط به پایان برنامه صورت می گیرد . این دستورات را همراه با توضیح و چگونگی عملکرد آنها در جدول زیر می بینید :

عملکرد	عملوند	توضیحات
BE	_	پایان برنامه ، برنامه صرفنظر از اینکه بیت RLO چه مقداری داشته باشد پایان می یابد
BEU	_	پایان برنامه بدون شرط ، برنامه صرفنظر از اینکه بیت RLO چه مقداری داشته باشد پایان می یابد
BEC	_	پایان برنامه با شرط ، چنانچه مقدار RLO یک باشد برنامه پایان یافته ، در غیر اینصورت اجرای برنامه ادامه می یابد

لازم به ذکر است که دستور BE تنها در انتهای برنامه استفاده می شود ، در صورتیکه دستورات BEU و BEC در طول برنامه مورد استفاده قرار می گیرند .

نکته :

- **OB21 :** هنگامیکه PLC از Stop به Start سوییچ می شود ، این بلوک اجرا می شود .
- **OB22 :** هنگامیکه منبع تغذیه به حالت On می رود ، این بلوک اجرا می شود .
- **OB34 :** نشان دهنده وضعیت باتری است که در صورت تضعیف یا ایراد باتری ، این بلوک اجرا می شود .
- **OB35 :** برای وقفه دوره ای است و مدت زمان فراخوانی آن در پیش فرض 100ms می باشد .
- **OB100 – OB102 :** OBهای راه اندازی می باشند . هنگام راه اندازی PLC ابتدا این OBها اجرا می گردند و برای موارد ایمنی نظیر قطع و وصل جریان برق استفاده می شوند .

فصل ششم

PLC
S7_200

پس از معرفی و بررسی PLCهای LOGO و S5 خانواده زیمنس ، اکنون و در این فصل در مورد S7_200 ، یکی دیگر از PLCهای خانواده زیمنس صحبت خواهیم کرد . در ابتدا سخت افزار و سپس نرم افزار و برنامه نویسی این PLC را بیان خواهیم داشت .

آشنایی با سخت افزار S7_200 :

S7_200 نیز همانند سایر کنترل کننده های منطقی از چندین قسمت اصلی تشکیل شده است :

1 – CPU یا واحد پردازنده مرکزی

2 – واحدهای حافظه

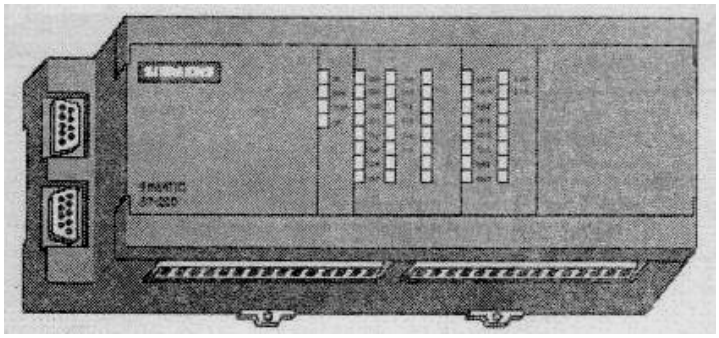
3 – واحدهای ورودی و خروجی

4 – منبع تغذیه

واحد پردازنده مرکزی CPU :

CPU یکی از مهمترین اجزای PLC محسوب می شود ، زیرا مرکز محاسبات و کنترل PLC بوده و دستورالعمل ها بوسیله این واحد پردازش و اجرا می شود . بعبارت دیگر این واحد را می توان مغز PLC دانست . قدرت و سرعت پردازش در یک کنترل کننده بستگی به سرعت پردازش CPU دارد . برای استفاده از CPU باید دستورالعمل های مناسبی به آن داده شود که این کار نیز با توجه به ساختار ماشین و پروسه کنترلی ، توسط کاربر وارد می شود . باید این نکته را نیز یادآور شد که CPU هوشمند نیست و خودش تصمیم نمی گیرد و فقط از مجموعه دستورالعمل هایی که توسط کاربر نوشته و در حافظه قرار می گیرد ، پیروی می کند .

تاکنون ده مدل S7_200 به بازار عرضه شده است که شامل سری های 21x و 22x می باشند که امروزه استفاده از CPU های سری 21x تقریبا منسوخ شده است . در سری 22x ، ابعاد کمتر و سرعت پردازش آن بیشتر از 21x است .



S7_200 21x

S7_200 22x

واحد CPU در S7_200 ، دارای کلیدهایی با عناوین Run\Stop\Term می باشد . علاوه بر این کلیدها ، تعدادی LED نیز در قسمت جلوی کارت CPU وجود دارد که به LED های وضعیت معروف می باشند . کار این LED ها ، نمایش حالت و وضعیت CPU می باشد. معمولا Run با رنگ سبز ، Stop با رنگ نارنجی و SF (خطاهای سیستمی) با رنگ قرمز نشان داده می شود .

مدهای کاری CPU :

حالت Run : در این حالت برنامه کاربر ، اجرا شده و CPU به ورودی و خروجی ها دسترسی دارد . در این وضعیت برنامه کاربر حالت ReadOnly دارد ، یعنی نمی توان برنامه جدیدی روی CPU ارسال نمود . برای امنیت و چنانچه خطایی در برنامه یا سخت افزار وجود داشته باشد ، CPU بصورت خودکار به حالت Stop می رود .

حالت Stop : در این حالت برنامه کاربر اجرا نمی شود و CPU در حالت توقف بوده و دسترسی به ورودی و خروجی ها وجود ندارد . در این وضعیت می توان برنامه را به PLC ارسال نمود یا از PLC برنامه آن را خواند .

حالت Term : این حالت هنگامی که کامپیوتر به PLC متصل است ، استفاده می شود . در این وضعیت امکان تغییر دادن مدهای کاری CPU توسط نرم افزار وجود دارد .

ترمینال های ورودی دیجیتال DI :

از این قسمت سیگنال ها به PLC ارسال می شوند . در اکثر PLCهای بزرگ ، سطح ولتاژ این سیگنال ها می بایست 24 VDC باشد . CPU از طریق این ماژول از اتفاقات محیط خارج از PLC مطلع می گردد . وضعیت سیگنال های ورودی توسط LEDهایی که در جلوی این مدول ها نصب شده اند ، قابل رویت است . تعداد ورودی های قابل اتصال به PLC از طریق ماژول ورودی ، بسته به نوع ورودی معمولا در رنج های 8 ، 16 و 32 تایی می باشد .

یادآوری : زمانیکه یک کلید در ورودی PLC بسته می شود ، CPU آن را معادل یک منطقی و زمانیکه همان کلید در ورودی PLC باز شود ، CPU آن را معادل صفر منطقی در نظر می گیرد .

ترمینال های خروجی دیجیتال DO :

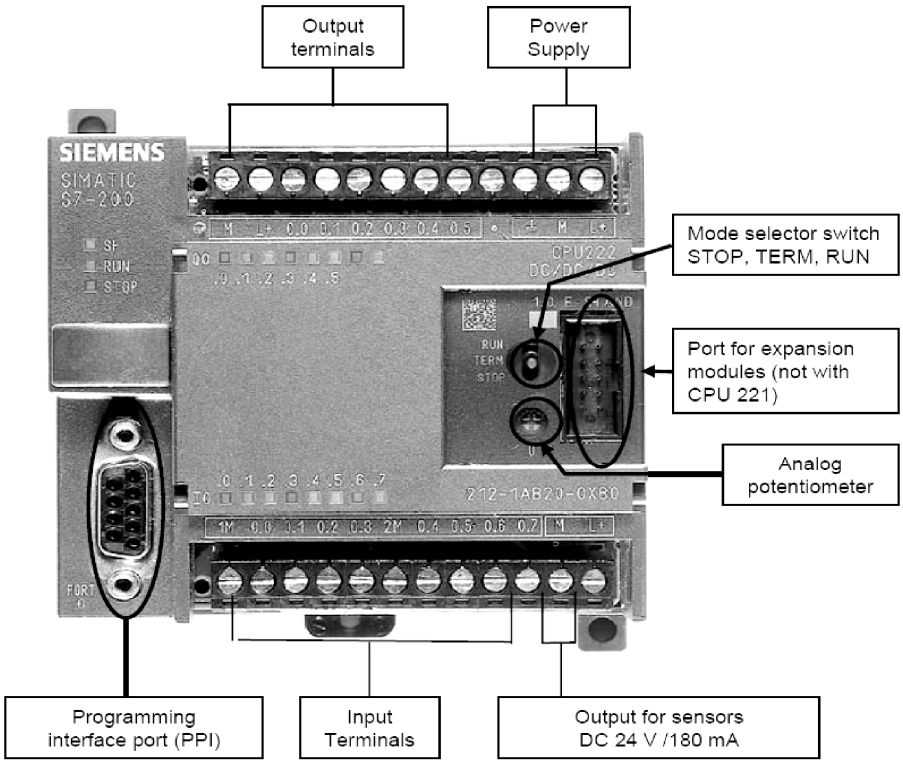
از این قسمت ، سیگنال ها به پروسه و محرک ها ارسال می شوند . این فرمان ها اغلب بصورت سیگنال های استاندارد 0 تا 24 VDC می باشند . وضعیت سیگنال های خروجی توسط دیودهای نوری که در جلوی این ماژول نصب شده اند قابل رویت است . باید این نکته را یادآور شد که کارت های خروجی معمولا در دو نوع رله ای و ترانزیستوری ساخته می شوند . در نوع رله ای ، فرمان CPU یک رله داخلی را فعال کرده و از طریق کنتاکت این رله ، خروجی فعال می شود . یکی از مزیت های خروجی رله ای این است که توسط آن می توان بارهایی با ولتاژهای بالا و یا سلفی را کنترل نمود . در نوع ترانزیستوری ، فرمان CPU یک ترانزیستور را فعال می کند و از طریق آن خروجی فعال می شود .

ترمینال های ورودی آنالوگ AI :

کمیت های آنالوگ بصورت پیوسته می باشند و نمی توان توسط ورودی های دیجیتال آن ها را اندازه گیری نمود . جهت اندازه گیری این کمیت ها در صنعت ، از ورودی های آنالوگ استفاده می شود . این نوع ورودی ها در واقع سطح سیگنال آنالوگ را به دیجیتال تبدیل می نمایند. کارت های ورودی آنالوگ دارای سخت افزار لازم جهت تبدیل سیگنال آنالوگ ولتاژ و جریان و تبدیل آنها به مقدار عددی می باشند . در این کارت ها معمولا مبدل A/D بکار رفته است . ماژول های آنالوگ سیگنال های پیوستی دریافتی از فرآیند را بمنظور پردازش داخلی در PLC به سیگنال دیجیتال تبدیل می نمایند . اکثر ماژول های آنالوگ مجهز به یک کلید DIP جهت متغیرهای خاص می باشند .

ترمینال های خروجی آنالوگ AO :

جهت ارسال سیگنال آنالوگ از PLC به سطح پروسه ، از خروجی های آنالوگ استفاده می شود . این خروجی ها در حقیقت سطح سیگنال داخلی PLC که یک سیگنال دیجیتال می باشد را به سیگنال آنالوگ تبدیل می نماید . در کارت های خروجی آنالوگ از مبدل های D/A استفاده شده است که یک عدد دیجیتال را به ولتاژ یا جریان تبدیل می نماید .



منبع تغذیه PS :

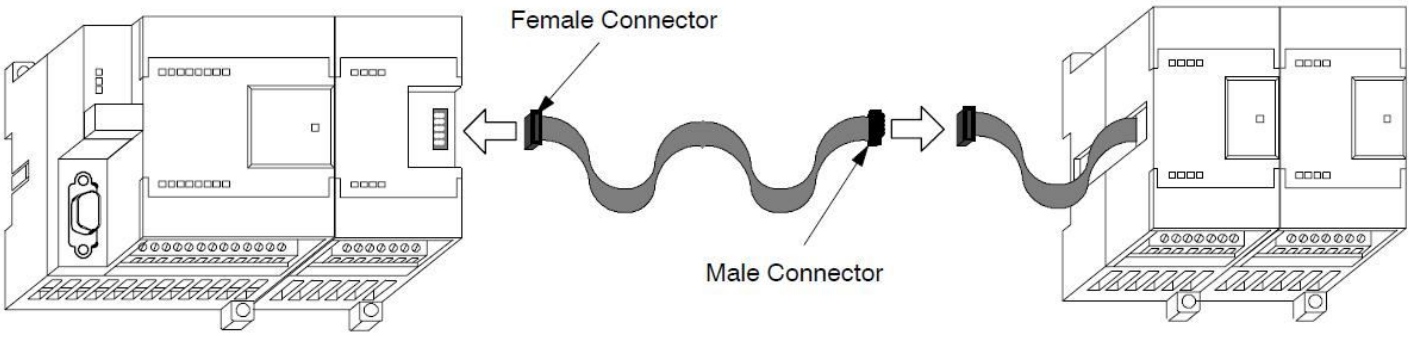
کار منبع تغذیه دریافت برق شهر و تامین ولتاژهای مورد نیاز PLC می باشد . طراحی مدار این کارت ها بصورت Switching بوده و از پایداری و راندمان بالایی برخوردار می باشند . البته منظور از این منبع تغذیه ، منبع تغذیه داخلی CPU نمی باشد . در بعضی مواقع که نیاز به یک منبع تغذیه با توان بالاتر می باشد ، می توان از این منبع تغذیه استفاده نمود .

پورت ارتباطی :

امکان ارتباط CPU را با شبکه های صنعتی و کامپیوتر و دیگر دستگاه های جانبی فراهم می کند . دقت نمایید که ارتباط بین PLC با کامپیوتر از طریق استاندارد RS232 می باشد . بعضی از انواع S7_200 دارای دو پورت ارتباطی می باشند . از این پورت ها جهت اتصال به پانل های نمایشی و پروگرامر استفاده می شود .

کانکتور ارتباطی :

در S7_200 از یک کانکتور جهت ارتباط بین CPU با واحدهای توسعه یافته استفاده می شود . از این کانکتور برای ارتباط بین واحد مرکزی CPU با واحدهای توسعه و یا ارتباط بین واحدهای توسعه بکار می رود . در صورت نیاز به اضافه نمودن واحدهای دیگر ورودی و خروجی به PLC از این کانکتور استفاده می شود . اگر در کنترل فرآیندهای صنعتی تعداد I/O ها افزایش یابد ، باید از ماژول های افزایشی استفاده نمود .



کارت های افزایشی :

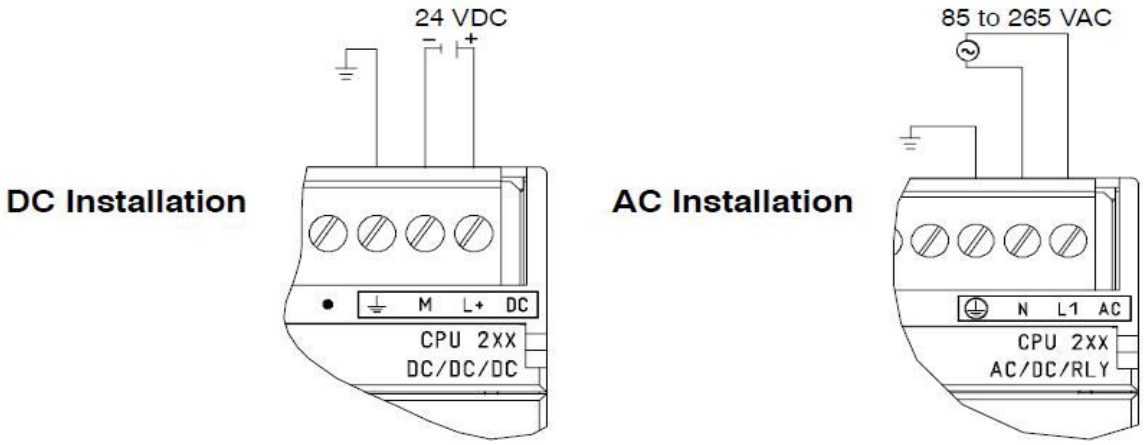
این کارت ها به ورودی و خروجی هایی که بصورت Onboard روی PLC وجود دارد ، اضافه می شوند . این کارت ها می توانند مانند PLC بر روی Rail نصب شوند . معمولا کارت های افزایشی در سمت راست PLC قرار می گیرند .

نصب S7_200 بر روی ریل های استاندارد :

S7_200 نیز همانند سایر کنترل کننده ها می تواند بر روی ریل های استاندارد قرار گیرد . جهت انجام این کار قلاب موجود در پشت PLC را روی ریل قرار داده ، سپس اسلاید موجود در پشت PLC را بطرف پایین کشیده و سپس طرف دیگر ریل را نیز وارد قلاب می کنیم ، در پایان کار اسلاید را رها می نماییم . در پشت S7_200 یک اسلاید تعبیه شده است که توسط این اسلاید PLC می تواند به ریل متصل شود . در ضمن جهت اتصال واحدهای گسترش یافته نیز می توان از ریل های استاندارد استفاده نمود .

اتصال تغذیه به PLC :

PLC های سری S7_200 در دو نوع AC و DC موجود می باشند . کاربرد نوع AC از DC بیشتر می باشد . در S7_200 دو ترمینال جهت اتصال تغذیه تعبیه شده است .



در PLC هایی که دارای تغذیه AC می باشند ، از برق شهر 220VAC جهت تغذیه استفاده می شود . این سری از PLC ها پس از دریافت برق شهر ، بر روی دو ترمینال دیگر که با نام های L و M مشخص شده است ، ولتاژ 24VDC را تولید می کنند . از این سطح ولتاژ جهت اتصال سنسورها و یا اتصال ورودی ها استفاده می شود . خروجی 24VDC که توسط PLC تولید می شود ، معمولا دارای سطح جریانی برابر 100 میلی آمپر می باشد .

نصب S7_200 روی پانل :

علاوه بر نصب S7_200 روی ریل های استاندارد ، روش دیگری نیز جهت نصب وجود دارد . S7_200 قابلیت نصب مستقیم بر روی یک پانل را دارا می باشد . در گوشه های S7_200 محل هایی جهت نصب مستقیم روی پانل تعبیه شده است . نصب S7_200 می تواند هم بصورت افقی و هم بصورت عمودی باشد .

اتصال PLC به کامپیوتر :

بعد از نصب PLC در محل موردنظر و اطمینان از سلامت تغذیه نوبت به پروگرام کردن برنامه به PLC می رسد . جهت انجام این کار ابتدا باید کابل PC/PPI را از یک طرف به کامپیوتر و از طرف دیگر به PLC متصل نمود . ارتباط کامپیوتر با PLC از طریق پورت سریال صورت می گیرد . جهت ارسال برنامه از کامپیوتر به PLC حتما باید تغذیه PLC وصل باشد یا بعبارت دیگر PLC روشن و در حالت STOP باشد . زمانی که برنامه ای به PLC ارسال می شود ، CPU بصورت خودکار به حالت STOP می رود .

انواع حافظه و مکان های حافظه در S7_200 : :

I : ورودی های فیزیکی — Q : خروجی های فیزیکی

AIW : ورودی های آنالوگ — AQW : خروجی های آنالوگ

ACC : آکومولاتور (انبارک قابل خواندن و نوشتن) — M : حافظه عمومی

T : حافظه برای مقدار تایمرها — C : حافظه برای مقدار کانترها و شمارنده ها

V : حافظه ویژه جهت جابجایی داده ها یا آدرس . این حافظه نقش چکنویس را دارد و اعداد و داده ها به عنوان متغیر در آن ذخیره می شود و به مجرد آنکه برق PLC قطع شود ، مقادیر صفر می شوند .

L : متغیر محلی است و فقط برای زیربرنامه هایی که در آنها تعریف می شوند ، قابل استفاده هستند .

SM : حافظه ویژه جهت کارهای خاص

برنامه نویسی S7_200 :

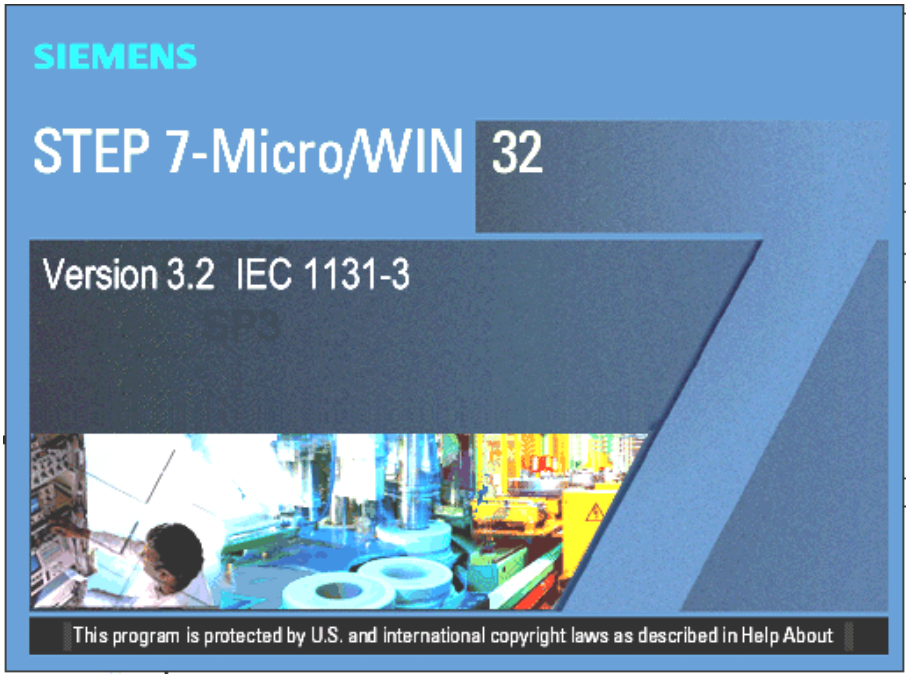
یک خط برنامه به روش STL از قسمت های زیر تشکیل شده است :

Address	Operand	Operation
آدرس	عملوند	عملکرد

نحوه آدرس دهی را با چند مثال بررسی می کنیم :

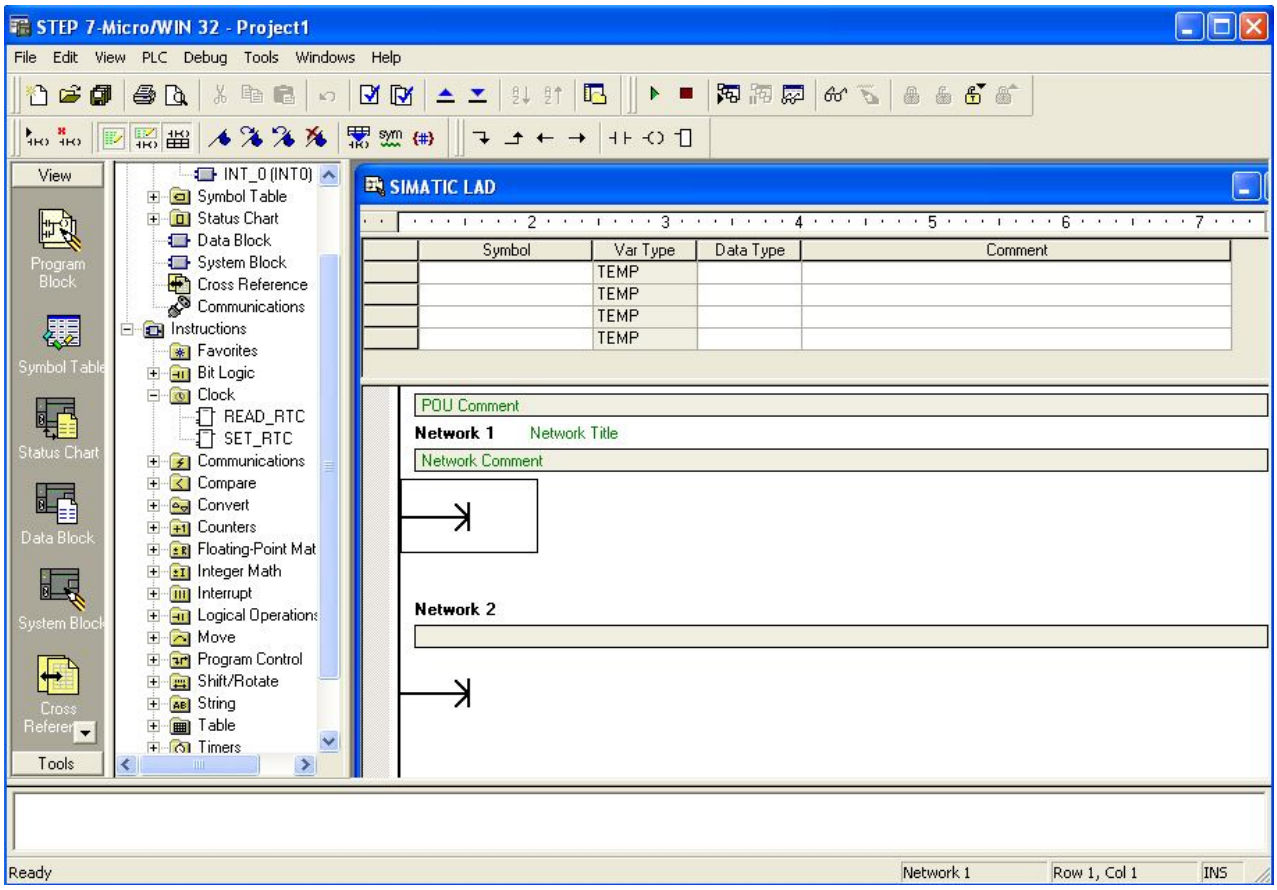
I 8.2	بیت دوم از بایت هشتم ورودی
Q 5.3	بیت سوم از بایت پنجم خروجی
M 1.3	بیت سوم از بایت اول فضای حافظه

در ادامه این فصل می خواهیم شما خواننده گرامی را با نرم افزار برنامه نویسی S7_200 آشنا کنیم . برای برنامه نویسی PLC های S7_200 از نرم افزار Step7_Micro/Win استفاده می شود . این نرم افزار جامع و کاربردی ، به برنامه نویس این امکان را می دهد که به هر سه زبان LAD ، CSF و STL برنامه نویسی کند . در ادامه کار با این نرم افزار و همچنین برنامه نویسی S7_200 را در کنار هم مطرح و به آموزش آن می پردازیم .



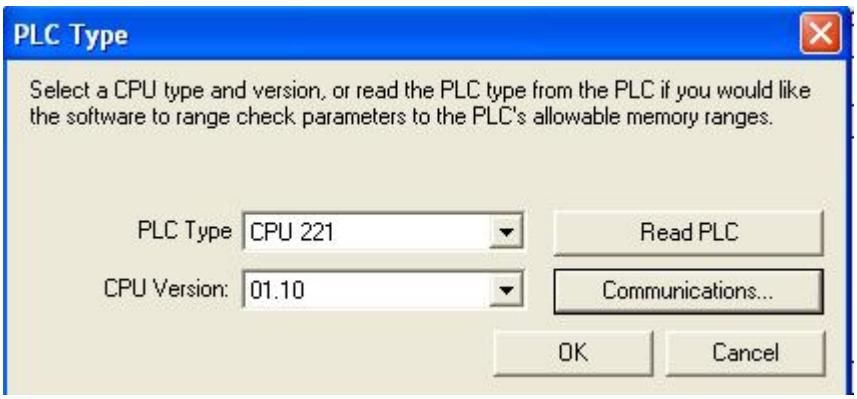
نرم افزار Step7_Micro/Win :

قبل از شروع به کار با نرم افزار بهتر است با محیط نرم افزار آشنا شویم . همانطور که در شکل زیر ملاحظه می کنید نرم افزار Micro/Win دارای قسمت های متعددی است که به بررسی آنها خواهیم پرداخت .

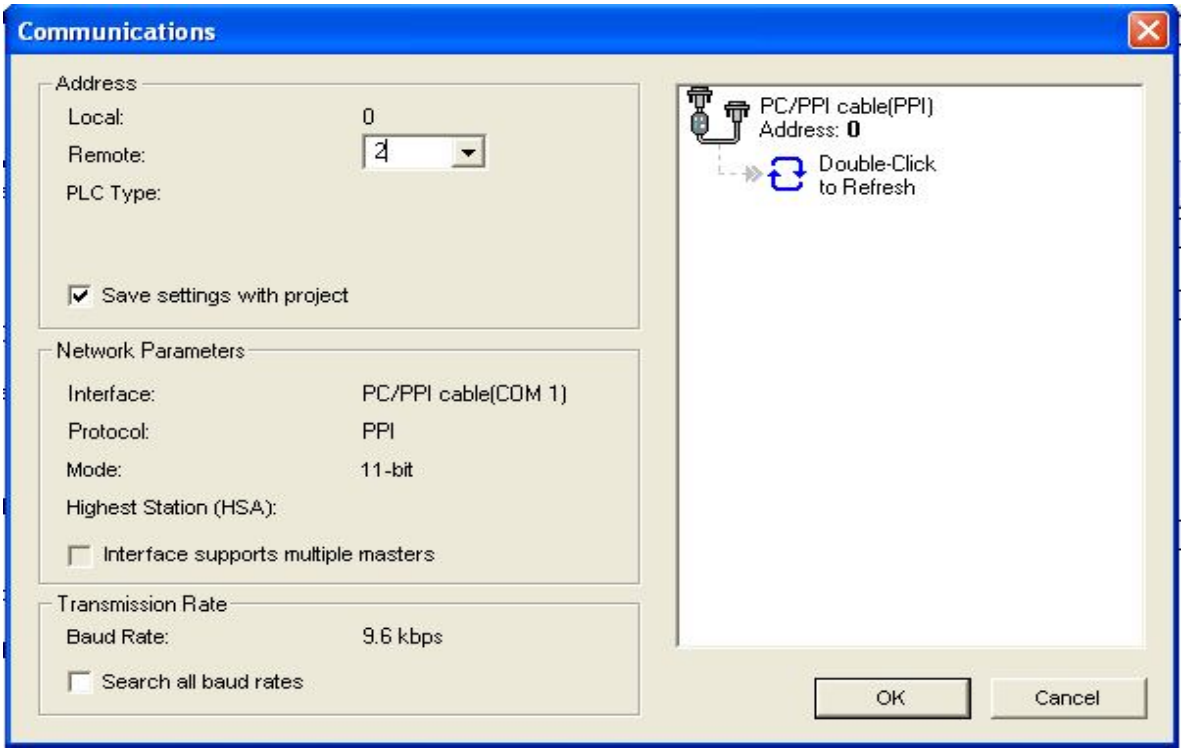


برای اینکه بتوانیم با نرم افزار Micro/Win کار کنیم باید عملکرد چند منوی اصلی را بدانیم . باید این نکته را متذکر شد که این نرم افزار ، محیط شبیه ساز برای تست برنامه ندارد و برای تست برنامه باید با PLC ارتباط برقرار کرد و تست برنامه بصورت Online صورت گیرد . برای این منظور مراحل زیر را انجام می دهیم :

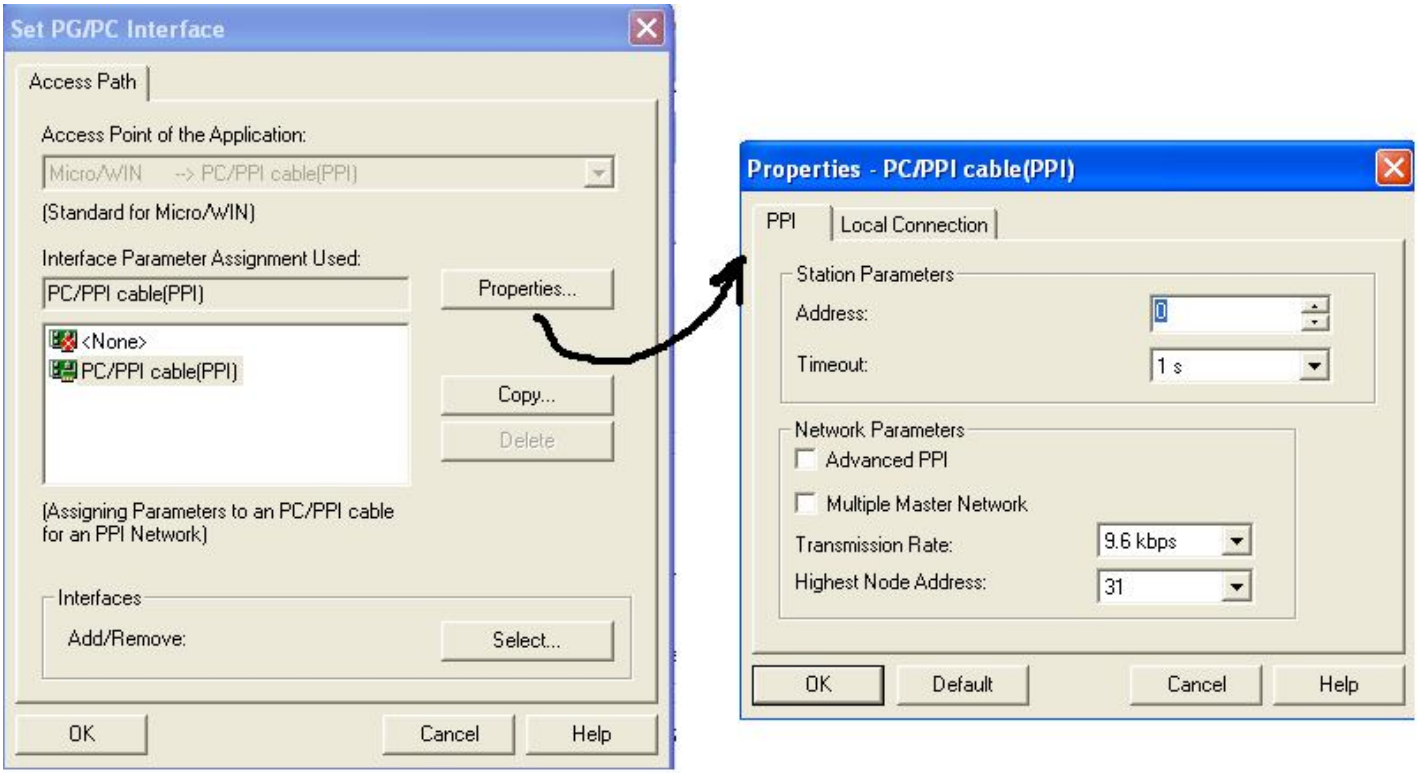
1 – ابتدا وارد منوی PLC شده و گزینه Type را انتخاب می کنیم . در صفحه PLC Type نوع CPU را وارد می کنیم .

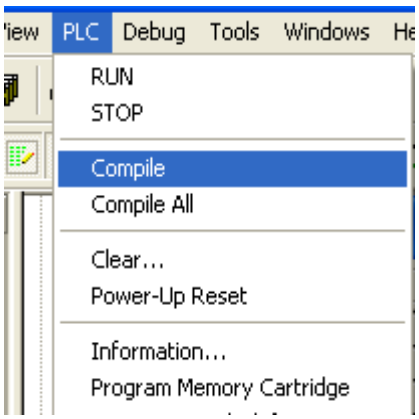


2 – جهت شناختن PLC ، گزینه Communication را کلیک می کنیم . صفحه Communication باز می شود . در این صفحه در قسمت Remote عدد 2 را می نویسیم و سپس بر روی Double Click to Refresh دوبار کلیک می کنیم .



اگر به هر دلیلی کابل را نشناخت ، از صفحه Communication قسمت PC/PPI را دوبار کلیک می کنیم تا وارد صفحه Set PG/PC Interface شویم. در این صفحه بر روی Properties کلیک می کنیم تا صفحه Properties باز شود . در صفحه باز شده ، در قسمت PPI مقدار Address برای PG همیشه صفر می باشد ولی می توان مقدار Time Out را تغییر دهیم .

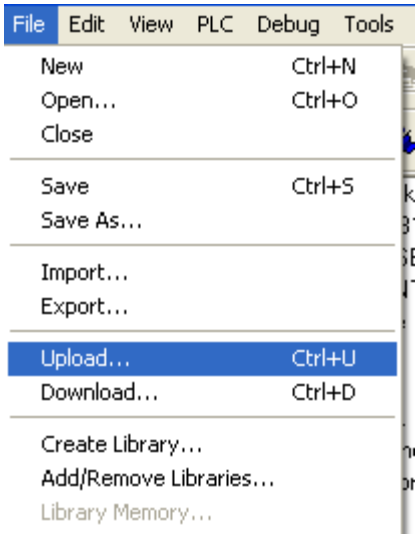




3- منوی PLC گزینه Run و Stop : در صورتیکه سوئیچ روی سخت افزار در وضعیت Term باشد ، از طریق نرم افزار می توان PLC را به حالت Stop یا Run ببریم .

4 – منوی PLC گزینه Compile و Compile All : قبل از Download کردن ، برنامه را باید Compile کنیم تا اگر از نظر ویرایشی برنامه صحیح نباشد ، مشخص شود . با انتخاب گزینه Compile قسمت های Program Block و Data Block چک می شود .

5- منوی PLC گزینه Clear : با انتخاب این گزینه می توانیم برنامه داخل PLC را پاک کنیم . روش دیگر پاک کردن برنامه آن است که یک صفحه خالی را در PLC بارگذاری کنیم .

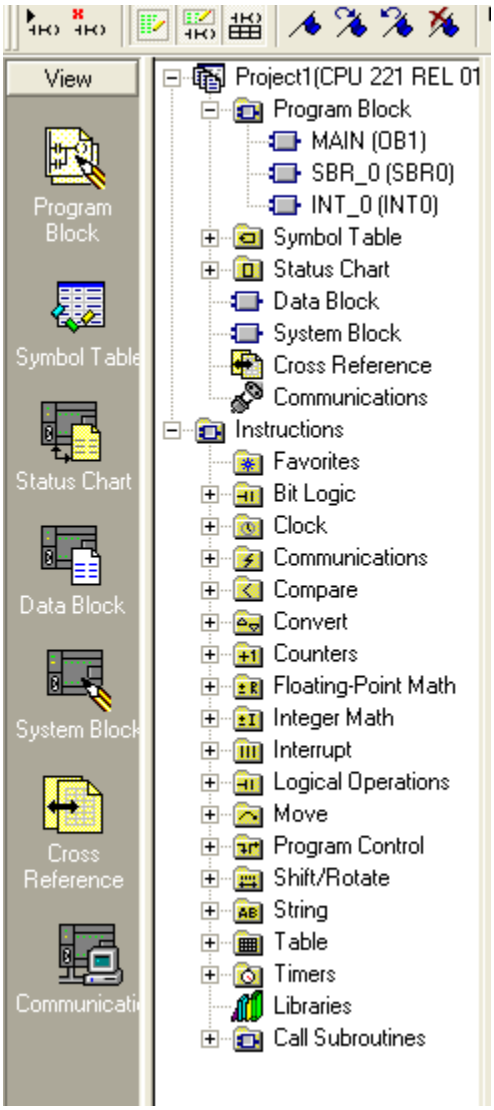


6- منوی File گزینه Upload و Download : هنگامیکه برنامه ای را نوشتیم ، جهت ارسال آن به PLC از گزینه Download استفاده می کنیم . همچنین اگر برنامه ای در PLC باشد و بخواهیم آن را روی PG ببینیم ، Upload را استفاده می کنیم .

7 - منوی PLC گزینه Information : در صفحه PLC Information در خصوص وضعیت PLC توضیح داده می شود که مثلا دارای چه مدل CPU است و یا در وضعیت Stop یا Run می باشد .

8 – منوی PLC گزینه Program Memory Catridge : اگر بخواهیم از کارت حافظه استفاده کنیم ، جهت ریختن برنامه از PLC به کارت حافظه و یا برعکس از این پنجره استفاده می کنیم .

9 – منوی View گزینه STL_LAD_FBD : زبان برنامه نویسی را عوض می کنیم .



صفحه دستورالعمل ها Information Tree :

ابزارها و امکانات برنامه نویسی و دستورالعمل ها در این قسمت قرار دارد . در صورتیکه قسمت Information Tree را بر روی صفحه برنامه نداشتیم ، می توانیم از طریق منوی View و گزینه Frame زیرمجموعه Information Tree را کلیک کنیم .

Navigation Bar :

این نوار ابزار در سمت چپ صفحه قرار می گیرد که از طریق منوی View گزینه Frame نیز قابل دستیابی است . نوار Navigation Bar شامل قسمت های مختلف زیر است که در ادامه به بررسی هر قسمت خواهیم پرداخت :

1- Program Block (بلوک برنامه نویسی)
2 – Status Chart (نمودار وضعیت)
3 – Cross Reference (ارجاع متقابل)
4 – System Block (بلوک سیستم)
5 – Data Block (بلوک اطلاعات)
6 – Symbol Tabel (جدول سمبل ها)

بلوک برنامه نویسی Program Block :

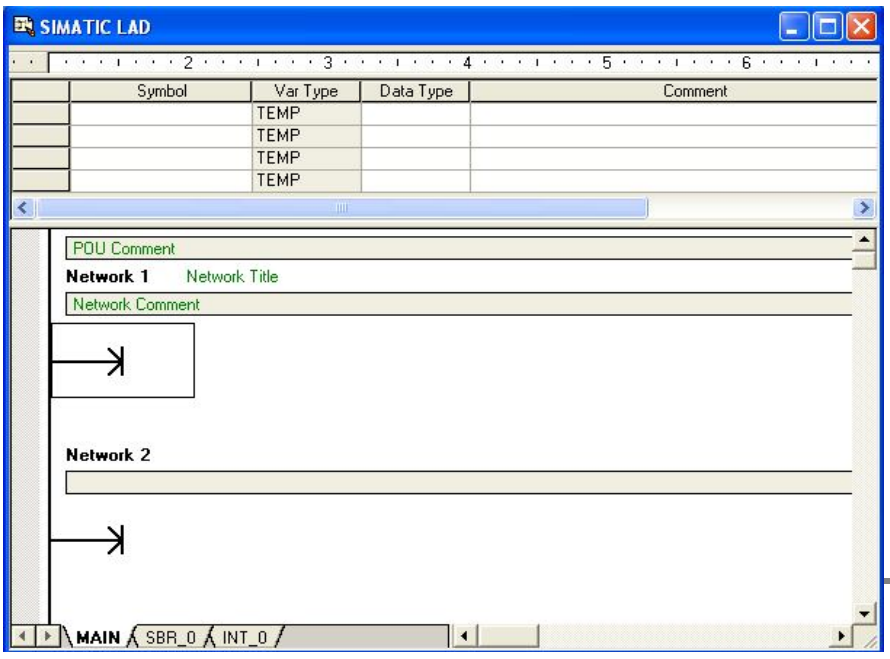
این بلوک محیط اصلی برنامه نویسی در S7_200 می باشد .

Main : همان محیط اصلی برنامه نویسی است که با OB1 نمایش می دهند .

SBR : زیر روال یا Subroutines می باشد .

INT : روال توقف ، وقفه یا Interrupt Routines می گویند .

برای وارد شدن به صفحه Program Block می توانیم بر روی شکل آن در نوار Navigation Bar کلیک کنیم . در صفحه Program Block در پایین صفحه سه انتخاب Main ، SBR و INT وجود دارد .



در صفحه Program Block عناوین مهم زیر را داریم :

- Network Number : برای مشخص کردن Network ها از شماره های مجزا استفاده می کنند . این شماره گذاری اتوماتیک انجام می شود .
- Network Title : اسم Network در کنار آن نوشته می شود ، شما می توانید نهایتا 256 کاراکتر برای هر اسم داشته باشید .
- Network Comments : توضیحات Network زیر اسم Network نشان داده می شود و قابلیت توضیحات جزئی تر Network را فراهم می کند . شما می توانید از 4096 کاراکتر برای هر توضیح Network استفاده کنید .

اکنون برای آنکه بتوانیم در محیط Program Block برنامه نویسی کنیم باید به بررسی صفحه دستورالعمل بپردازیم . دستورات برنامه نویسی از Bit Logic شروع شده و تا Call Subroutines ادامه دارد .

: Bit Logic

1 – Normally Open Contact

زمانی که مقدار ذخیره شده در آدرس n یک باشد ، بسته خواهد بود و جریان در آن مشاهده می شود .

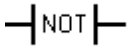
2 - Normally Closed Contact

زمانی که مقدار ذخیره شده در آدرس n صفر باشد ، بسته خواهد بود و جریان در آن مشاهده می شود .

3 – Output

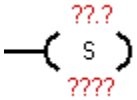
زمانی که جریان برقرار می شود ، خروجی فعال شده و بیتی که آدرس n به آن اشاره می کند ، یک می شود .

4 – Not



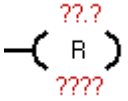
حالت برقراری جریان را معکوس می کند .

5 – Set



از S_bit شروع و تا n نقطه بعد از آن را Set می کند .

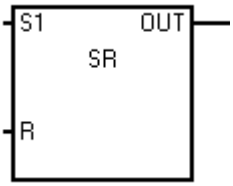
6 – Reset



از S_bit شروع و تا n نقطه بعد از آن را Reset می کند .

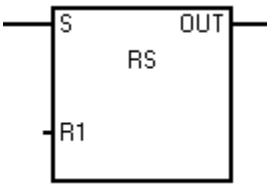
نکته : دستور Set و Reset نباید با هم در یک Network باشد و باید هرکدام در یک Network جداگانه باشند .

7 – Set Dominant Bistable (فلیپ فلاپ SR)



اگر پایه S و R با هم یک باشند خروجی یک می شود (اولویت با پایه S)

8 – Reset Dominant Bistable (فلیپ فلاپ RS)



اگر پایه S و R با هم یک باشند ، خروجی صفر می شود (اولویت با پایه R)

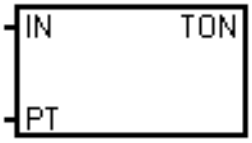
تایمرها Timer :

در S7_200 سه نوع تایمر وجود دارد که جمع تعداد کل آنها 256 عدد می باشد و با دقت 1ms (4 تایمر) ، 10ms (16 تایمر) و 100ms (236 تایمر) می باشد .

Timer Type	Resolution	Maximum Value	Timer Number
TONR	1 ms	32.767 s	T0, T64
	10 ms	327.67 s	T1-T4, T65-T68
	100 ms	3276.7 s	T5-T31, T69-T95
TON, TOF	1 ms	32.767 s	T32, T96
	10 ms	327.67 s	T33-T36, T97-T100
	100 ms	3276.7 s	T37-T63, T101-T255

اگر بخواهیم از تایمر TON با دقت 10ms استفاده کنیم ، می توانیم از تایمرهای شماره T33 الی T36 و یا از تایمرهای شماره T97 الی T100 استفاده کنیم . اگر بخواهیم از تایمر TONR با دقت 100ms استفاده کنیم از تایمرهای شماره T5 الی T31 و یا از تایمرهای شماره T69 الی T95 استفاده کنیم .

تایمر تاخیر در وصل TON :



زمانیکه ورودی تایمر TON فعال شود ، تایمر شروع به کار می کند و بمحض اینکه مقدار آن بزرگتر یا مساوی PT شد ، بیت تایمر فعال می گردد . تایمر و کنتاکت آن نباید هر دو با هم در یک Network قرار گیرند .

تنظیم زمان تایمر PT :

جهت تنظیم مقدار PT تایمر ، ابتدا باید دقت تایمر مشخص شده و سپس مدت زمانی را که می خواهیم تایمر کار کند را مشخص کرده و در فرمول زیر قرار می دهیم .

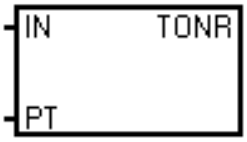
$$\mathbf{PT} = \frac{\text{زمان مورد نظر}}{\text{دقت تایمر}}$$

مثال : می خواهیم PT را برای یک تایمر بدست آوریم بشرطی که مقدار 2s و دقت تایمر 1ms باشد .

$$\mathbf{PT} = \frac{2s}{1ms} = \mathbf{2000}$$

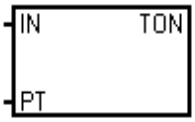
نکته : مقدار PT عدد صحیح است و اعداد کسری و اعشاری را در برنمی گیرد .

تایمر تاخیر در وصل پایدار TONR :



مانند TON است با این تفاوت که در تایمر TON در صورتی که IN تایمر مقدارش یک شود ، تایمر شروع به محاسبه می کند . اگر قبل از پایان زمان موردنظر تایمر ، مقدار IN صفر گردد و مجددا یک شود ، تایمر TON از ابتدا زمان را محاسبه ولی تایمر TONR از بقیه زمان باقیمانده ، زمان را محاسبه می کند . همچنین اگر تایمر TONR زمان خود را به اتمام رسانده و کنتاکت خود را در Network بعدی فعال نماید ، در صورتیکه مقدار IN تایمر صفر گردد ، مقدار کنتاکت تایمر TONR در Network های بعدی صفر نخواهد شد . جهت صفر کردن مقدار تایمر TONR و همچنین صفر کردن مقدار کنتاکت های آن در Network های بعدی ، باید از دستور Reset استفاده نمود .

تایمر تاخیر در قطع TOF :

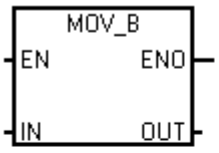


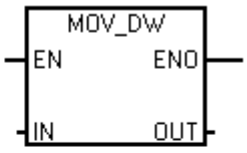
بلافاصله بعد از یک شدن IN تایمر ، خروجی یک می شود و پس از گذشت زمان تایمر ، خروجی صفر خواهد شد .

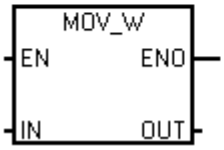
نکته : در تایمر TOF زمانی که مقدار IN همواره یک باشد ، تایمر زمان را ثابت نگه داشته و عمل نمی کند .

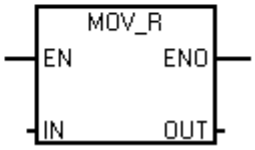
دستور Move :

Move به معنای حرکت است . انتقال اطلاعات و داده ها و اعداد از جایی به جای دیگر توسط دستور Move انجام می گیرد . Move دارای فرمت های مختلفی است مانند MOV_B ، MOV_W و MOV_DW . اگر ورودی EN فعال باشد ، مقدار موجود در ورودی های IN در آدرس خروجی OUT کپی می شود .









نکته : اگر از Move با فرمت بایت استفاده می کنیم باید مقدار IN و OUT هر دو با فرمت بایت باشد .

عملیات ریاضی :

بطور کلی عملیات ریاضی بر روی دو دسته بزرگ اعداد صورت می گیرد : اعداد صحیح ، اعداد حقیقی .

اعداد صحیح یا Integer شامل (..... , +1 , 0 , -1 , ...) می باشد . نوع داده INT همان عدد صحیح 16 بیتی است . این عدد می تواند بین 32768- تا 32768+ باشد .

اعداد حقیقی یا Real شامل (..... , 2.5 , 1 , 0 , .5- , 1.5- , ...) می باشد و دارای فرمت DW هستند .

عملیات ریاضی Integer شامل موارد زیر است :

جمع اعداد صحیح ← ADD_I

تفریق اعداد صحیح ← SUB_I

ضرب اعداد صحیح ← MUL_I

تقسیم اعداد صحیح ← DIV_I

نکته : از کاربردهای عملیات ریاضی می توان به شمارنده ها اشاره نمود ، مثلا بیشترین عددی که یک کانتر می تواند بشمارد 999 است . برای شمارش بیشتر باید حاصل چند کانتر را با هم جمع نمود .

مقایسه کننده ها Compare :

دو مقدار آنالوگ را با هم مقایسه می کند و همینطور دو تایمر ، دو کانتر ، دو عدد و ..... مقایسه کننده ها با فرمت های Byte ، Word ، Double Word دارد . توجه نمایید که دو مقداری را که می خواهیم با هم مقایسه کنیم ، فرمتشان چیست و با توجه به فرمتشان از مقایسه کننده مناسب استفاده کنیم . در ادامه به بررسی انواع مقایسه کننده ها می پردازیم :

1 - INT1 ==I INT2 : اگر مقدار INT2 برابر با مقدار INT1 باشد ، جریان برقرار می شود .

2 - INT1 <>I INT2 : اگر مقدار INT2 برابر با مقدار INT1 نباشد ، جریان برقرار می شود .

3 - INT1 >=I INT2 : اگر INT2 برابر یا بزرگتر از INT1 باشد ، جریان برقرار می شود .

4 - INT1 <=I INT2 : اگر INT2 کوچکتر یا برابر INT1 باشد ، جریان برقرار می شود .

5 - INT1 >I INT2 : اگر INT2 از INT1 بیشتر باشد ، جریان برقرار می شود .

6 - INT1 <I INT2 : اگر INT2 از INT1 کمتر باشد ، جریان برقرار می شود .

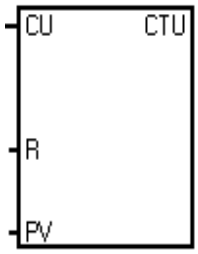
شمارنده ها :

درPLC های S7_200 دو نوع کانتر وجود دارد :

1- کانترهای معمولی شامل CTU , CTD , CTUD

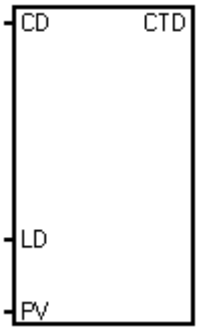
2- کانترهای سرعت بالا شامل PLS , HDEF , HSC

تعداد کانترهایی که در یک برنامه می توانید استفاده کنید بین 0 تا 255 عدد می باشد .



کانتر صعودی شمار CTU :

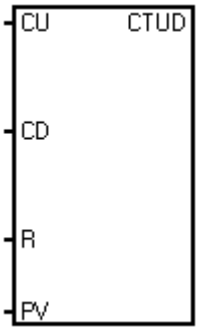
در این کانتر پایه CU ، ورودی کانتر برای Start است . پایه R ، برای Reset کانتر مورد استفاده قرار می گیرد و در پایه PV تعداد شمارش مقداردهی می گردد .



کانتر نزولی شمار CTD :

در این کانتر پایه CD ورودی کانتر برای Start است . پایه L برای Reset کانتر مورد استفاده قرار می گیرد و در پایه PV تعداد شمارش معکوس مقداردهی می گردد . در این کانتر عدد PV بطور معکوس شمارش می شود تا به مقدار صفر برسد ، یعنی هر زمان که CD یک پالس را ببیند یک عدد را محاسبه کرده و کم می کند تا به صفر برسد . LD نیز جهت Reset کردن کانتر می باشد . وقتی مقدار کانتر صفر شد ، کنتاکت آن عمل کرده و خروجی فعال می شود .

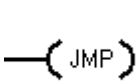
کانتر CTUD :

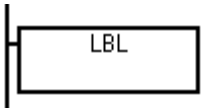


در این کانتر هم شمارش به سمت بالا و هم شمارش به سمت پایین با هم محاسبه می شود . پایه CU ورودی سنسور پالس دهنده به سمت بالا ، پایه CD ورودی سنسور پالس دهنده به سمت پایین ، پایه R برای Reset کانتر و در پایه PV تعداد شمارش مقداردهی می شود .

دستور JUMP و LBL :

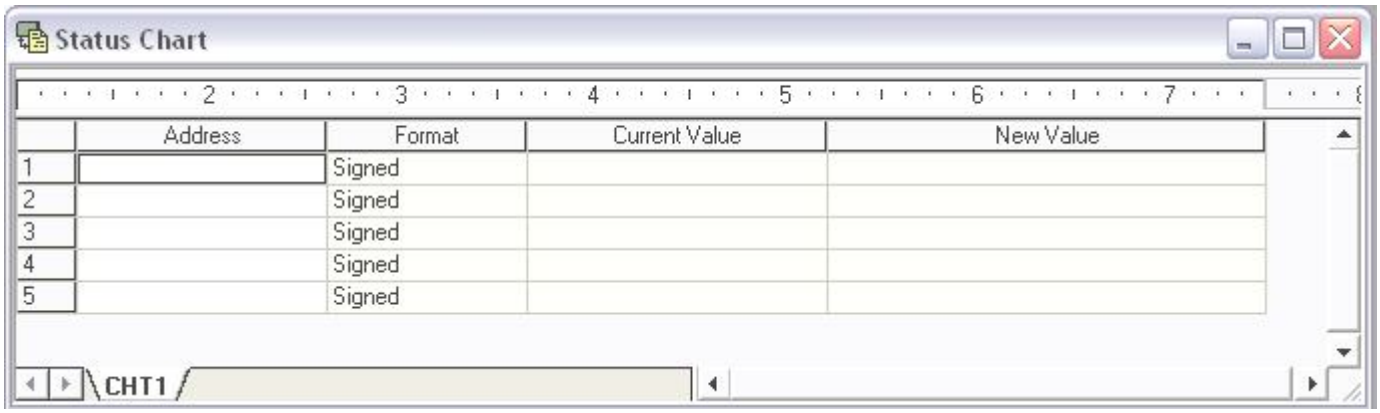
گاهی اوقات در برنامه لازم است با یک شدن ورودی ، چند سطر از برنامه غیرفعال شود و از چرخه برنامه خارج گردد و در صورت لزوم مجددا به چرخه برنامه بازگردد . برای این منظور در قسمت Program Control از دستور JUMP و LBL استفاده می کنیم .





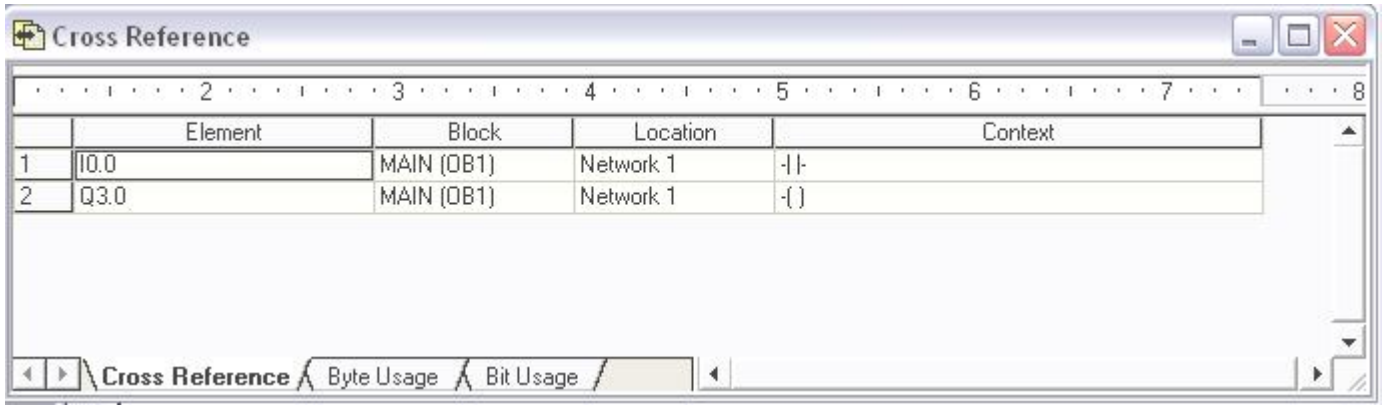
نمودار وضعیت Status Chart :

نمودار وضعیت شما را قادر می سازد که در حین اجرا شدن برنامه ملاحظه کنید مقادیر بکار رفته در برنامه چگونه تغییر می کنند ، همچنین وضعیت ورودی ها و خروجی ها چگونه است . باید توجه داشت که نمودار وضعیت بر روی PLC دانلود نمی شود . برای وارد شدن به Status Chart می توانیم از طریق منوی View گزینه Component قسمت Status Chart را انتخاب کنیم .



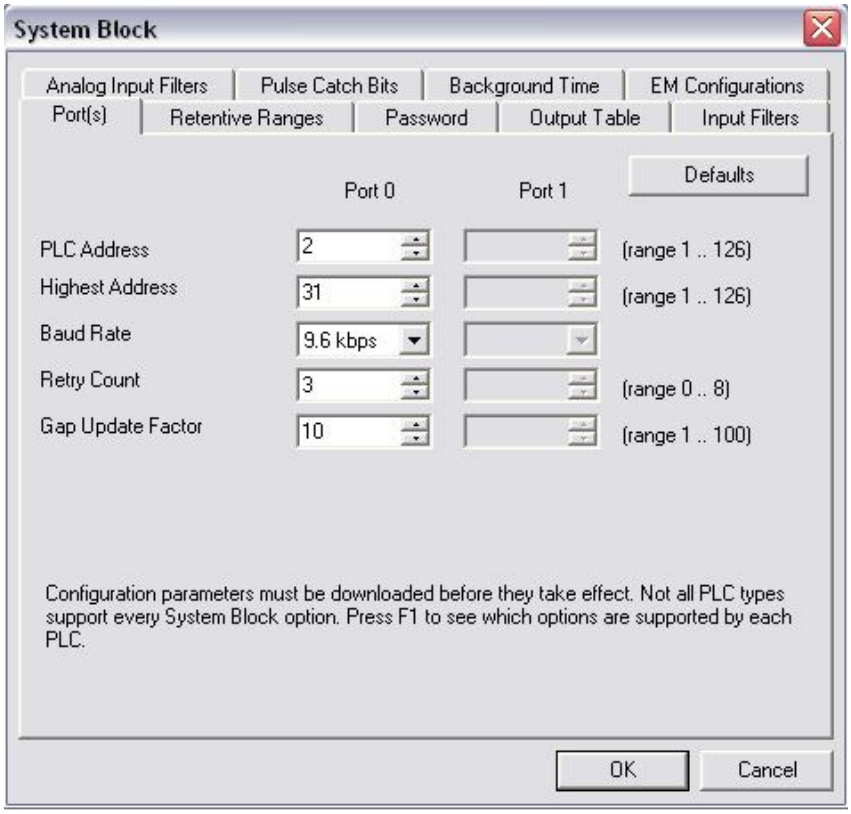
ارجاع متقابل Cross Reference :

حافظه های استفاده شده ، مقادیر ورودی و خروجی و همچنین صفحه های استفاده شده در PLC جهت جستجو راحتتر ، در این قسمت قرار می گیرند . پنجره ارجاع متقابل بر روی PLC دانلود نمی شود .



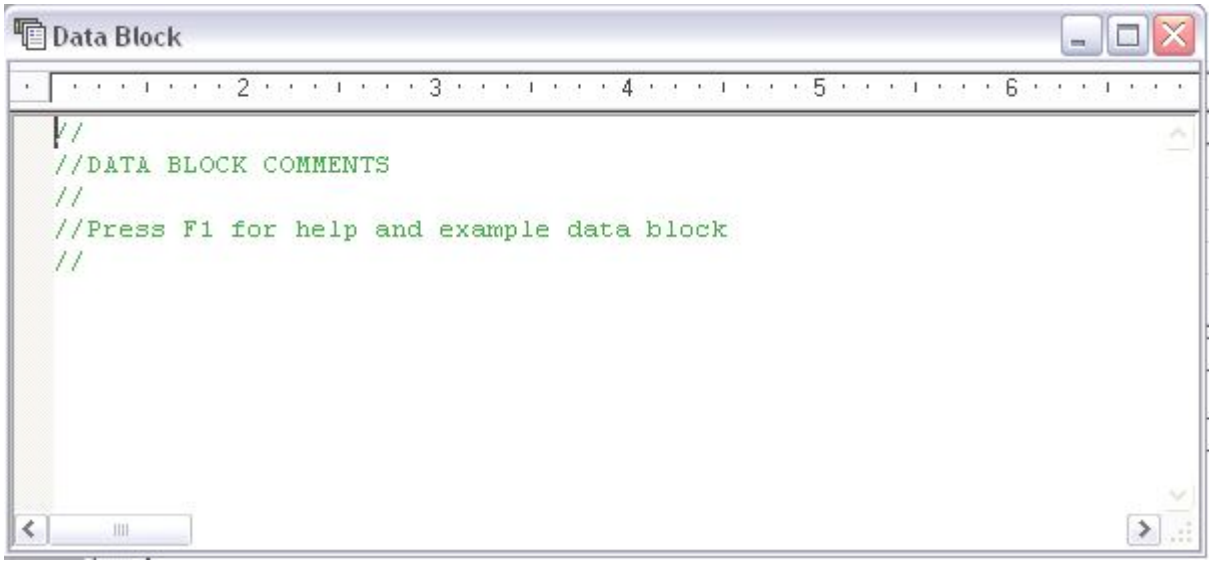
بلوک سیستم System Block :

این قسمت سخت افزار PLC را پیکربندی می کند .



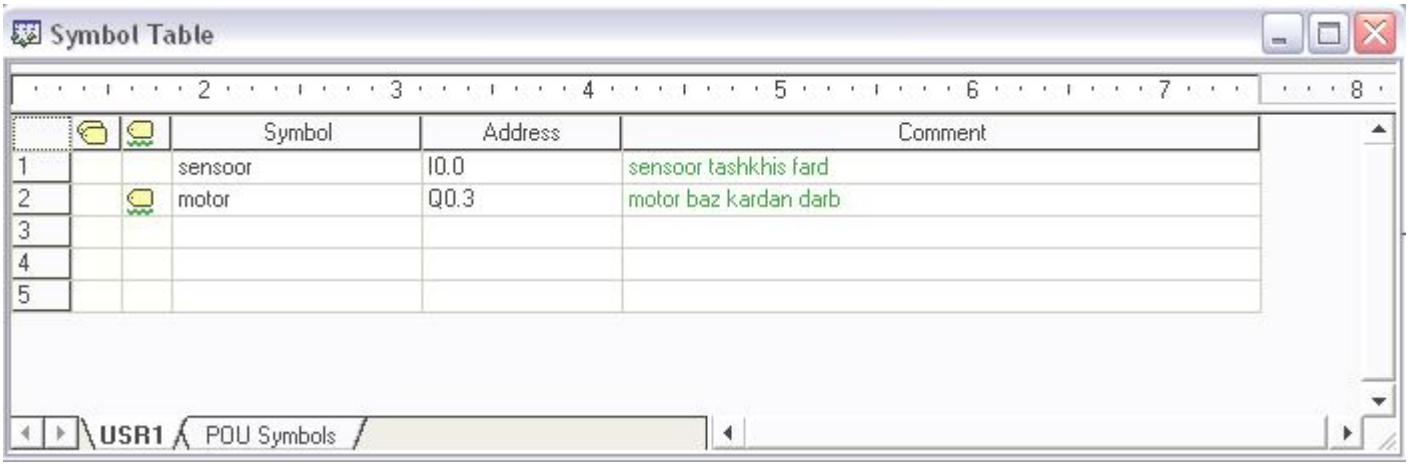
بلوک اطلاعات Data Block :

اطلاعات برنامه و شرایط اولیه در PLC را ذخیره می کند . بلوک اطلاعات از داده ها و توضیحات داده ها و اطلاعات تشکیل شده است . می توان یک مقدار اولیه را نیز برای حافظه ها و داده ها در نظر گرفت . توجه شود که بلوک اطلاعات بر روی PLC دانلود نمی شود .



جدول سمبل ها Symbol Table :

جدول سمبل ها ، برنامه نویس را قادر می سازد از آدرس های سمبلیک استفاده کند . سمبل ها ابزاری مناسب برای برنامه نویس ایجاد می کند و برنامه را برای دنبال کردن و چک کردن آسانتر می کند . باید توجه داشت که جدول سمبل ها بر روی PLC دانلود نمی شود .



در پایان این فصل به نکات مهم زیر اشاره می کنیم :

- ممکن است بعضی از خوانندگان گرامی خواستار ارائه مثال های متعدد برای دستورات S7_200 بوده اند که در این مجموعه بیان نشده است ، برای این دسته از عزیزان راهنمایی زیر برای ارائه مثال بیان می شود :

در نوار ابزار نرم افزار Micro/Win در قسمت Help گزینه ?What is this را انتخاب کنید . صفحه ای باز می شود که حاوی تمامی بلوک ها و گیت های برنامه نویسی است که با کلیک بر روی هر یک ، توضیحات کامل به همراه مثالی از آن دستور بیان شده است . توجه به این نکته لازم است که ارائه مثال برای تمامی دستورات بر حجم مجموعه می افزود که موجب کسالت فکری خواننده می شد لذا برای خواننده علاقمند راهنمایی فوق برای کسب اطلاعات بیشتر ارائه گردید .

- در این مجموعه تمامی دستورات و جزئیات برنامه نویسی S7_200 ارائه نشده است ، علت این امر وجود چندین کتاب جامع آموزش S7_200 که توسط همکاران عزیز نوشته شده است ، بود . لذا برای احترام به زحمت این عزیزان ، در این مجموعه تنها بخش کاربردی برای برنامه نویس مبتدی ارائه شد و از شما خواننده علاقمند دعوت می شود برای اطلاعات بیشتر به کتاب های جامع موجود مراجعه نمایید .

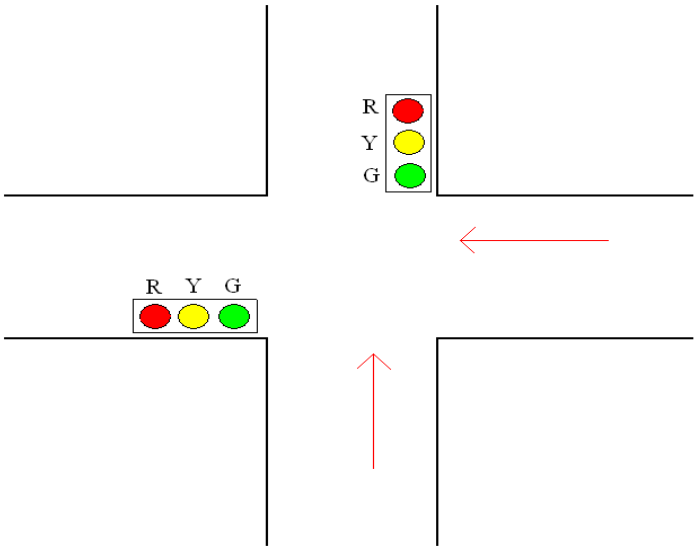
فصل هفتم

تمرینات

تمرین 1 :

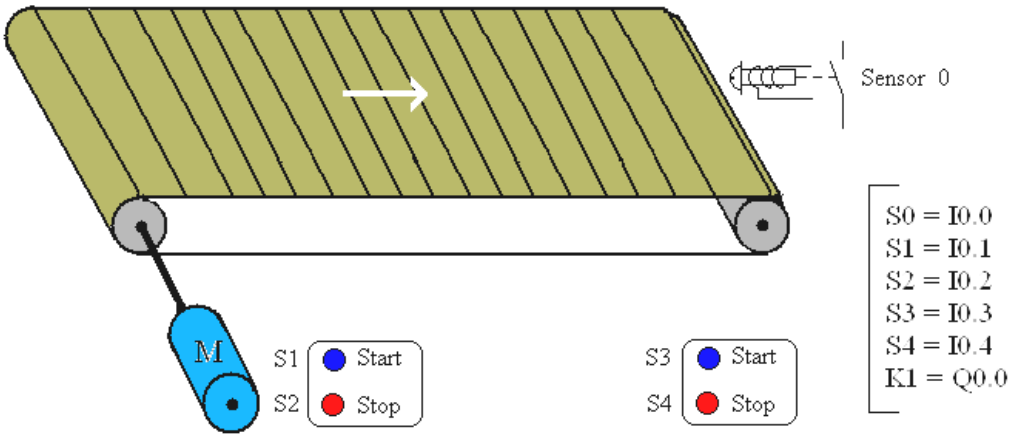
در یک چهارراه سیستم چراغ راهنمایی به صورت زیر است :

مدت زمان چراغ قرمز 30 ثانیه و مدت زمان چراغ زرد 5 ثانیه و مدت زمان چراغ سبز 30 ثانیه می باشد . سیستم کنترل آن را طراحی کنید .



تمرین 2 :

در شکل زیر دو کلید فشاری S1 و S2 به ترتیب برای استارت و استپ در سمت آغازین کانوایر وجود دارد ، همچنین در بخش انتهایی کانوایر دوکلید فشاری S3 و S4 برای استارت و استپ کانوایر تعبیه شده است . از طریق هر دو بخش آغازین و انتهایی کانوایر می توان آن را استپ یا استارت نمود . لازم به ذکر است که سنسور S0 برای توقف کانوایر هنگام رسیدن جسم به انتهای کانوایر نصب شده است . برنامه کنترلی این کانوایر را بنویسید .



تمرین 3 :

سیستم کنترل میز مسابقه سه نفره ای را به گونه ای طراحی کنید که اگر هر کدام از شاسی های S1، S2، S3 را که زودتر فشار داده شود ، چراغ مربوط به آن روشن شده و چراغهای دیگر عمل نکند .

تمرین 4 :

سیستمی را طراحی نمایید که با فشار دادن شاسی S1 کنتاکتور K1 مگنت کرده و در حالت مگنت باقی بماند ، زمانی که برای بار دوم شاسی S1 را فشار دادیم کنتاکتور K1 قطع شود .

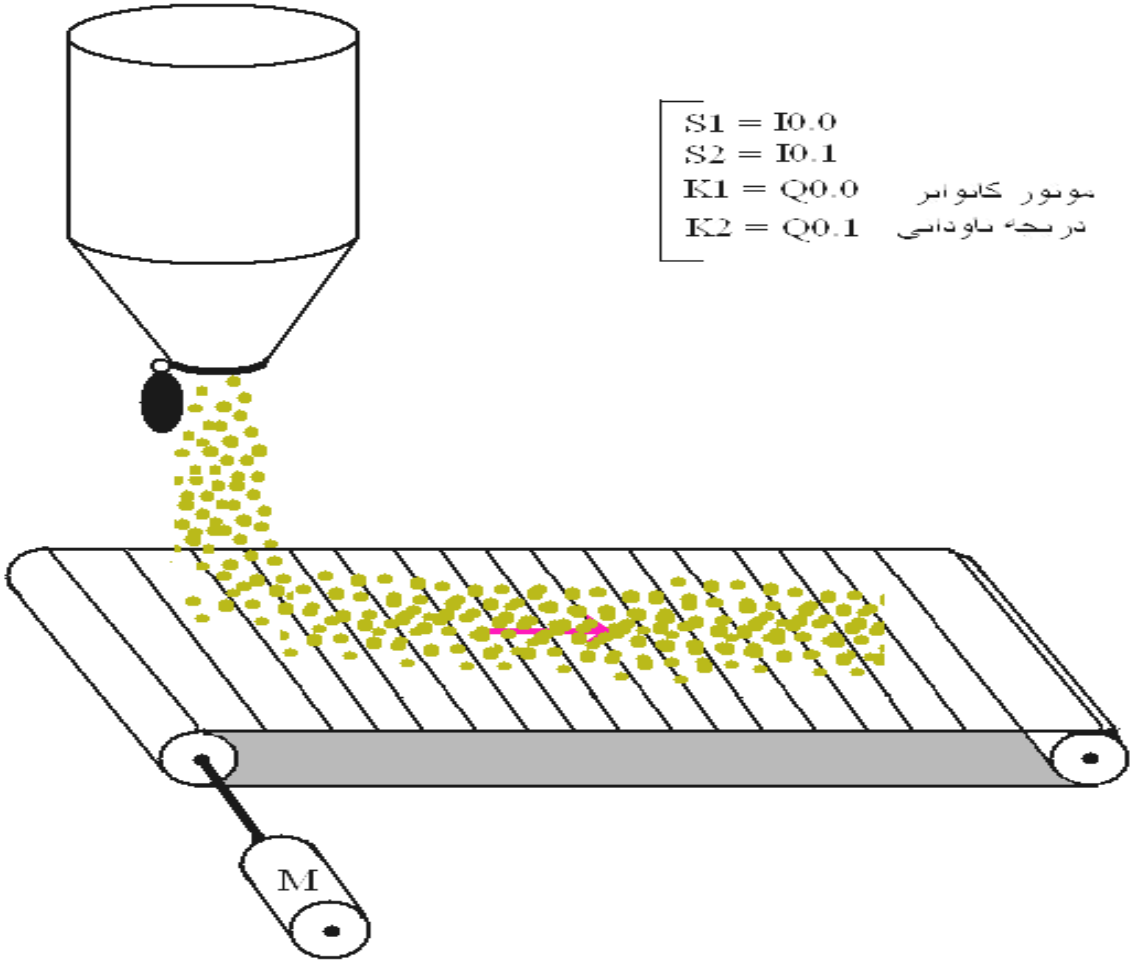
تمرین 5 :

در یک ماشین ابزار سیستم کنترل به صورت زیر است :

با فشار دادن شاسی S1 موتور بطور دائم کار می کند و با فشار دادن شاسی S0 موتور خاموش می گردد . با فشار دادن شاسی S2 موتور بطور لحظه ای کار می کند و هنگامی که شاسی S2 رها شود موتور خاموش می گردد . (سیستم کنترل لحظه ای و دائم)

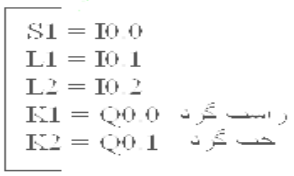
تمرین 6 :

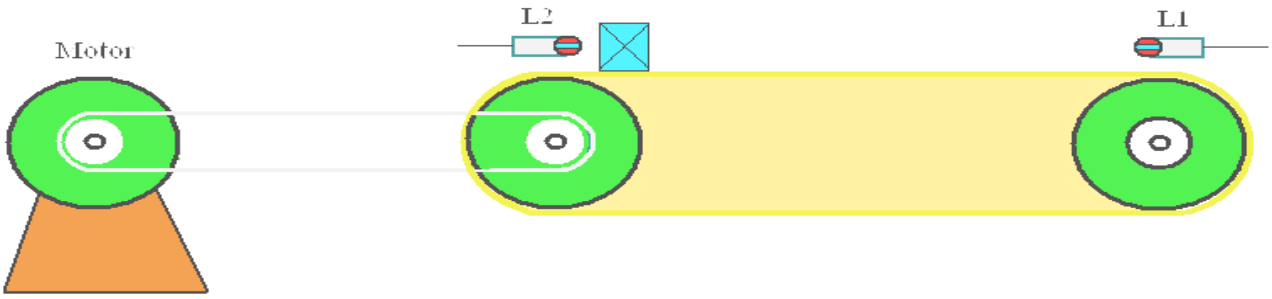
شکل زیر یک انتقال دهنده خرده سنگ از مخزن بر روی کانوایر می باشد . یک دریچه در قسمت ناودانی مخزن خرده سنگ وجود دارد و از آنجا خرده سنگ بر روی کانوایر ریخته می شود ، یک موتور حرکت کانوایر را کنترل می کند . اگر مکانیزم موتور کانوایر متوقف شود و یا مکانیزم عمل معیوب گردد ، دریچه ناودانی مخزن باید بسته شود . وقتی که دریچه مخزن انرژی می گیرد باز می گردد و با قطع انرژی آن بسته می شود شاسی فشاری S1 باعث خاموش شدن سیستم و شاسی S2 باعث روشن شدن سیستم می گردد .



تمرین 7 :

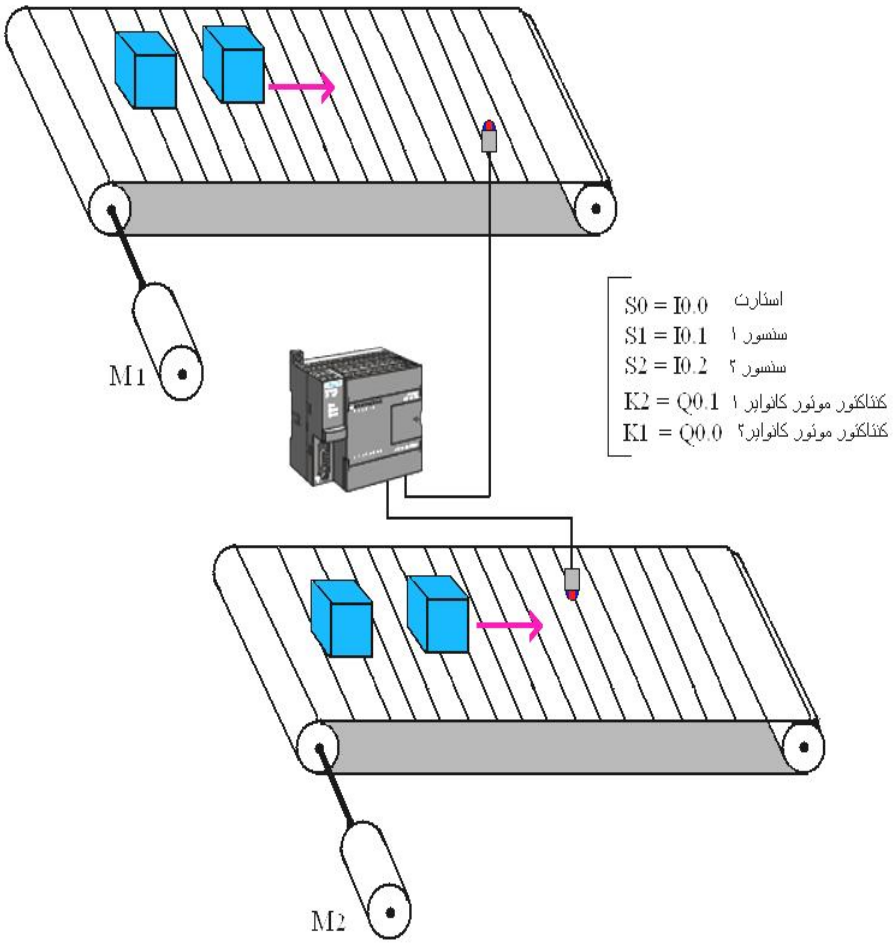
قطعه کاری بر روی یک کانوایر قرار دارد که می تواند در محدوده دو سنسور نوری L1 و L2 حرکت کند . وقتی که شاسی استارت فشار داده می شود ، در صورتی که سنسور L2 توسط قطعه کار تحریک شده باشد کانوایر توسط موتور به سمت جلو حرکت می کند . با حرکت کانوایر و برخورد آن به سنسور L1 کانوایر توسط موتور تغییر جهت می دهد .





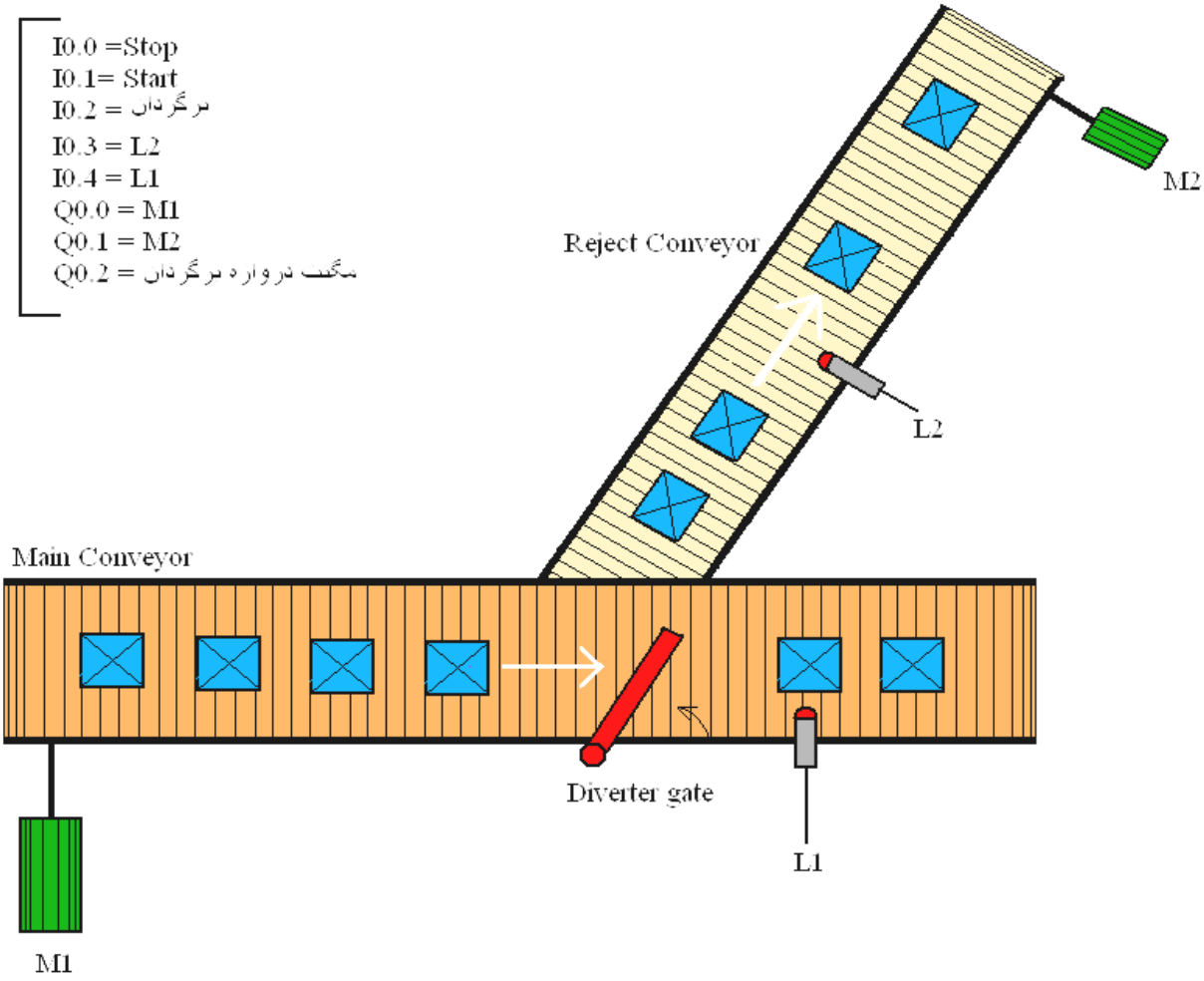
تمرین 8 :

در شکل زیر دو کانوایر که در ساخت یک فرآیند نقش دارند نشان داده شده است . هر کانوایر دارای یک سنسور می باشد که قطعه کارهای عبوری از کنار آن را شمارش می کند . برنامه هایی بنویسید که قطعه کارهای عبوری از هر یک از دو کانوایر را بطور اختصاصی شمارش نماید . زمانی که قطعات کانوایر 1 از 100 عدد گذشت کانوایر 1 خاموش شود و زمانی که سنسور کانوایر 2 از 200 عدد گذشت کانوایر 2 نیز خاموش شود . با زدن مجدد شاسی استارت مراحل از اول آغاز گردد .



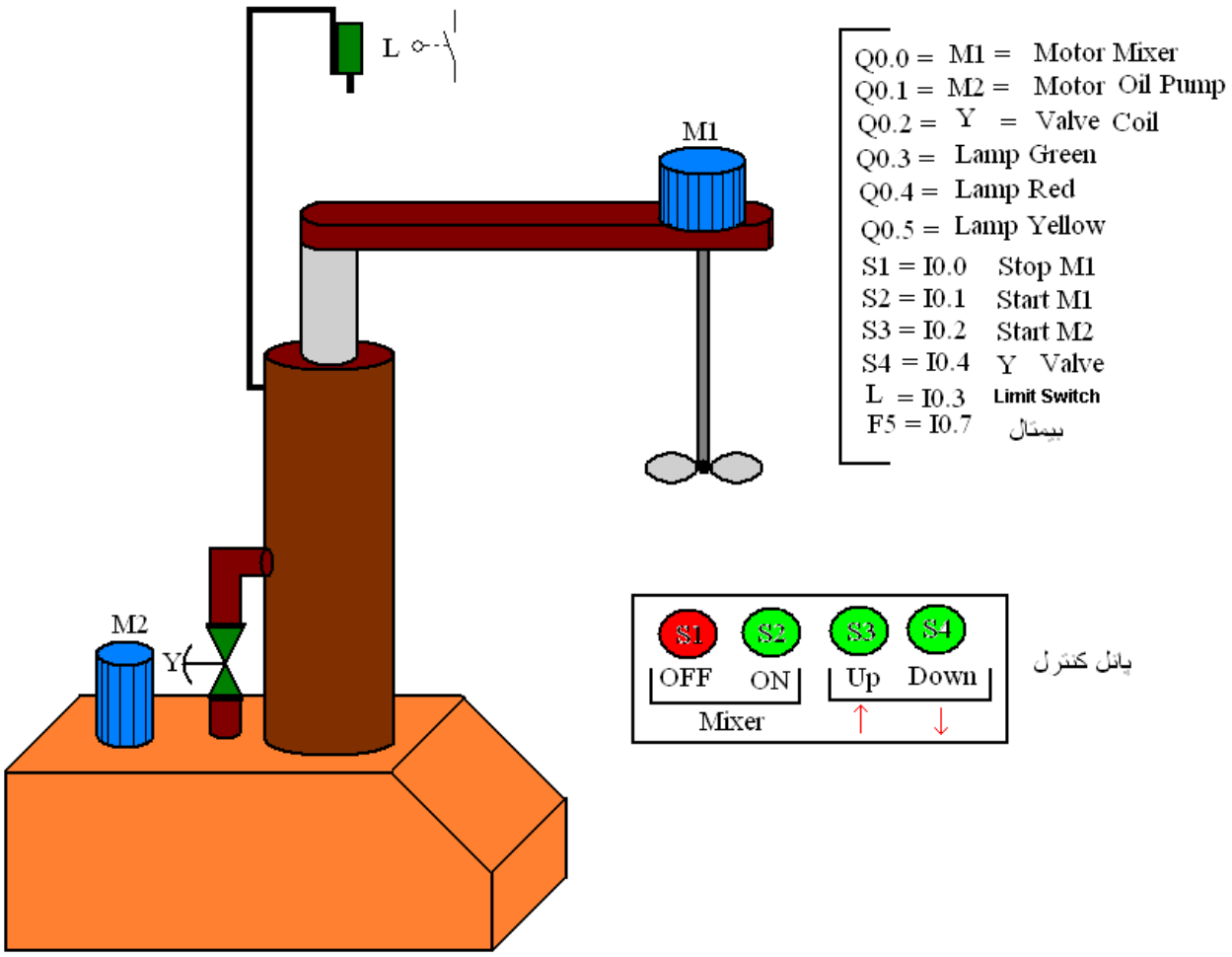
تمرین 9 :

در شکل زیر یک کانوایر اصلی (Main Conveyor) با یک دروازه برگردان (Diverter gate) برای قسمتهای معیوب به داخل کانوایر برگشتی (Reject Conveyor) را نشان می دهد . اگر در بازرسی یک قطعه کار معیوب باشد ، دروازه برگردان توسط اپراتور فعال شده و قطعات به داخل کانوایر برگشتی انتقال پیدا می کند . یک سنسور در کنار کانوایر اصلی و یک سنسور در کنار کانوایر برگشتی وجود دارد و قطعات عبوری را شمارش می کند . برنامه بنویسید که اگر قطعات عبوری از کانوایر برگشتی از 10 عدد و قطعات عبوری از کانوایر اصلی از 50 عدد گذشت سیستم کانوایر متوقف شود (سیستم عملکرد برگردان به این صورت است که با برق دار شدن برگردان عمل کرده و با قطع برق به حالت عادی باز می گردد)



تمرین 10 :

شکل زیر میکسر یک کارخانه صنایع رنگ سازی می باشد سیستم عملکرد میکسر به این صورت است که با زدن شاسی S2 موتور اصلی میکسر (M1) شروع به کار می کند . جهت بالا و پایین کردن پروانه میکسر از یک پمپ هیدرولیک استفاده شده است با روشن شدن موتور M2 پمپ هیدرولیک عمل کرده و پروانه میکسر تا محدوده میکروسوییچ L بالا رفته و با تحریک شدن میکروسوییچ L متوقف می شود . جهت پایین آوردن میکسر شیر برقی Y تعبیه شده که روغن را وارد مخزن اصلی می کند . سیستم کنترل آن را برنامه نویسی نمایید (در صورتی که موتور میکسر دچاراضافه بار گردید سیستم خاموش شده و چراغ نارنجی رنگ چشمک بزند)



تمرین 11 :

یک شمارنده ماشین برای شمارش تعداد ماشین وارد شده و خارج شده از یک پارکینگ با ظرفیت 25 اتوموبیل مورد نیاز است :

الف : یک ورودی تعداد اتوموبیل های وارد شده و ورودی دیگر تعداد اتوموبیل های خارج شده را می شمارد .

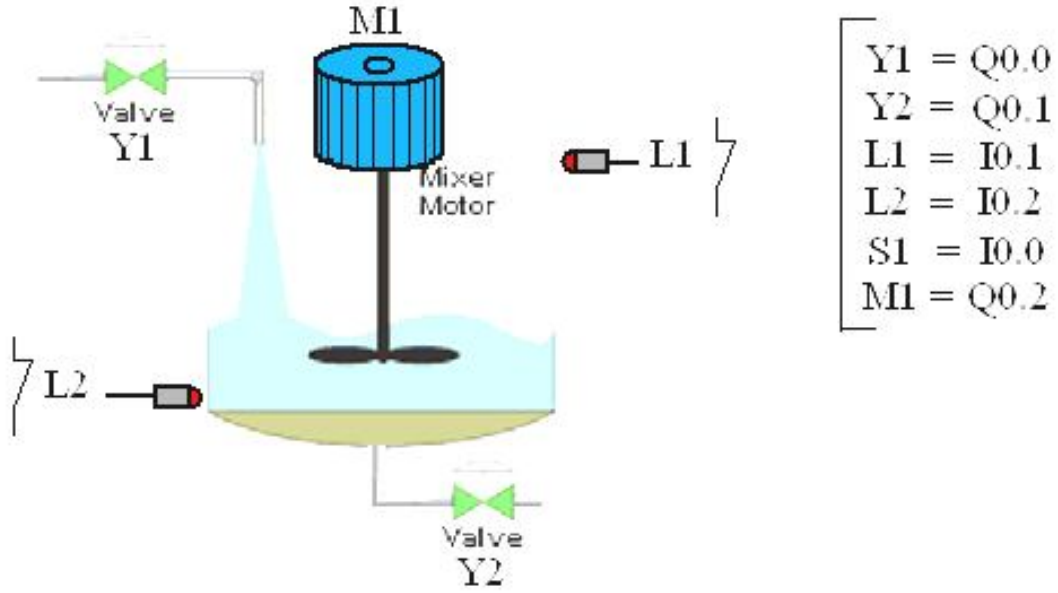
ب : وقتی تعداد اتوموبیل های داخل پارکینگ به 25 عدد رسید ، خروجی کانتر با نمایش عبارت ``Full`` پربودن پارکینگ را مشخص نماید .

ج : وقتی تعداد اتوموبیل های داخل پارکینگ کمتر از 25 عدد بود ، عبارت ``Vacancy`` بمعنی ظرفیت داشتن پارکینگ روشن شود .

د : یک کلید ورودی می تواند توسط متصدی پارکینگ ، پارکینگ را در موقعیت بسته نگهدارد .

تمرین 12 :

شکل زیر یک میکسر را نشان می دهد وقتی که شاسی استارت فشار داده شود سلنوئید Y1 فعال شده و مایع می تواند وارد مخزن شود . سنسورهای L1 و L2 سطح بالا و پایین مایع مخزن را مشخص می کنند و هر دو دارای کنتاکت NC می باشند (وقتی که مخزن خالی است L1 و L2 بسته هستند) زمانی که مخزن پر شد سنسور L1 سلنوئید Y1 را قطع و فرمان شروع به کار موتور میکسر را صادر می کند . موتور میکسر برای 30 ثانیه فعال بوده و سپس خاموش می گردد . وقتی که موتور خاموش شد سلنوئید Y2 فعال شده و مایع مخزن را تخلیه می کند پس از خالی شدن مخزن سنسور L2 به حالت عادی برگشته (بسته شده) و سلنوئید Y2 قطع می گردد . برنامه کنترل آن را بنویسید .



تمرین 13 :

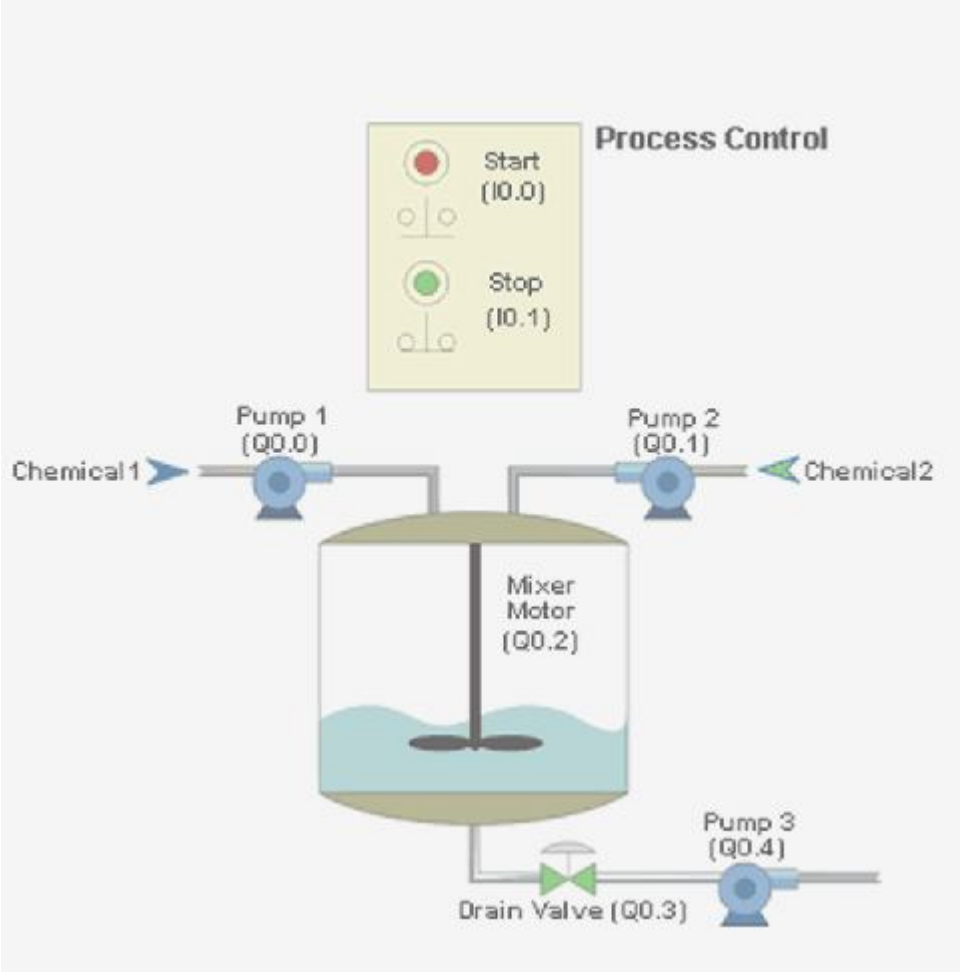
برنامه ای بنویسید مربوط به روشن و خاموش کردن چراغ راه پله های یک ساختمان بطوری که با وارد شدن به طبقه اول ساختمان با فشار دادن کلید چراغ طبقه اول روشن شود و در ابتدای ورودی طبقه دوم با فشار دادن کلید چراغ طبقه دوم روشن و چراغ طبقه اول خاموش شود و این روند تا طبقه آخر ادامه داشته و عکس آن از بالا به پایین نیز صادق باشد . همچنین در هر یک از طبقات در صورت انصراف بتوان برگشت . (سه طبقه)

تمرین 14 :

تایمری طراحی کنید که هم تاخیر در وصل باشد و هم تاخیر در قطع .

تمرین 15 :

شکل زیر یک مخزن به همراه میکسر مواد شیمیایی می باشد . سیستم عملکرد آن به این صورت است که با فشار دادن شاسی استارت ، پمپ یک شروع به کار کرده و یک مایع شیمیایی را وارد مخزن می کند پس از 20 ثانیه از شروع کار پمپ یک ، پمپ 2 نیز روشن شده و یک مایع شیمیایی دیگر را وارد مخزن می کند . پس از 10 ثانیه هر دو پمپ خاموش شده و موتور میکسر به مدت 15 ثانیه مواد شیمیایی را میکس می نماید سپس شیر خروجی باز شده و پمپ 3 به مدت 30 ثانیه روشن شده و مواد مخزن را تخلیه می کند برنامه آن را بنویسید .



تمرین 16 :

میخواهیم دو عدد موتور را بشرح زیر راه اندازی کنیم :

الف : با زدن کلید استارت ، موتور M1 شروع بکار نموده و بمدت 60 ثانیه روشن و سپس خاموش می شود .

ب : موتور M2 ، 15 ثانیه بعد از موتور M1 روشن شده و همراه با M1 خاموش شود .

فهرست منابع

منبع	مؤلف	مترجم	ناشر
کنترل موتورهای الکتریکی	Rexford & Giuliani	محمد طلوع خراسانیان	نشر طراح
عملکرد و کاربردهای PLC در اتوماسیون صنعتی	Ian Warnock	علی اکبر صفوی حسین شجاعی	مؤسسه علمی فرهنگی نص
خودآموز جامع پیشرفته PLC	عبداله بهرام پور		انتشارات سیمای دانش
کنترل کننده های قابل برنامه ریزی PLC	فرامرز خوش لفظ		انتشارات ابتدا
الکتروپنیوماتیک و کنترل کننده های منطقی قابل برنامه ریزی	حمیدرضا رستمی		جهان نو
اصول و کاربرد سنسورها	پیتر هاپتمن	نوید تقی زادگان لادن اجلالی مهران صباحی	انتشارات آشینا
کاربرد سنسورهای نوری در صنعت	جف تامسون	محمد شکیبا مهدی پیرمرادیان	انتشارات بال
سنسورها و ترانسدیوسرها	Ian Sinclair	محمد طلوع خراسانیان	نشر طراح
کنترل کننده های برنامه پذیر	ابراهیم حسینی سورنا مرآت		دیباگران تهران
مجموعه راحل های اتوماسیونی با مرجع کامل مینی PLC	هادی رضایی اسماعیل علی خانی		انتشارات مهکامه
کنترل کننده های منطقی قابل برنامه ریزی	داود رزم		دیباگران تهران
خودآموز PLC S7_200	محسن سید هاشم		جهان نو
مرجع کامل Simatic S7_200	اکبر اویسی فر		انتشارات سایه گستر
مبانی الکترونیک	سید علی میرعشقی		نشر شیخ بهایی
طراحی دیجیتال	موریس مانو	قدرت سپیدنام	انتشارات خراسان
خودآموز کامل PLC	آرمان مؤمنی		گروه نرم افزاری رزپارسیان
مرجع کامل PLC کنترل کننده های برنامه پذیر	فربد قابوسی		نشر آفرنگ
آموزش عملی PLC	محمد فدوی مزینانی		انتشارات امید مهر
جزوات آموزشی شرکت اندیشه سازان صنعت برق	مهدی توانا		شرکت اندیشه سازان صنعت برق
جزوات آموزشی شرکت قشم ولتاژ	محمدرضا ماهر نوشین سعیدی		شرکت قشم ولتاژ
برقکار صنعتی درجه 1	بهروز احمدی		جهان نو
تکنولوژی و کارگاه برق صنعتی درجه 2	احمد سهرابی		انتشارات گلپونه
کارگاه برق و مدار فرمان	جواد نیکوکار		نشر دانش پرور
استپ موتورها	عباس صمیمی فر		www.ir.micro.com
آشنایی با موتورهای پله ای	رضا فروغی		www.ir.micro.com
برقکار عمومی 2	خلیل افشارگلی محسن آقاحسینی		انتشارات سناباد
مدارات کاربردی برق صنعتی درجه 2	علی مسگری هادی قناد		انتشارات صفار
منابع لاتین شرکت Siemens	Micro System Simatic S7_200 Manual Siemens PLC_S5 Manual Siemens PLC_LOGO Manual		

فهرست منابع

منبع	مؤلف	مترجم	ناشر
خودکاری با PLC	سید حجت سبزپوشان		دانشگاه علم و صنعت
اتوماسیون صنعتی پیشرفته	صانع مجدانی		انتشارات سیمای دانش
مرجع کامل نرم افزار LOGO	محمد هادی رضایی		دانش پارسیان
جزوه آموزشی LOGO	طهماسبی		www.ir.micro.com
کارگاه مدار فرمان 2	اقبال طلب		دانشکده فنی میرزا کوچک خان
شرح ساده توابع LOGO	مهدی کهربایی		www.ir.micro.com
جزوه آمورش PLC به زبان نردبانی	فتح الله نظریان		www.eca.ir
Programmable Logic Controllers	W. Bolton	وحید کارگرمقدم	Elsevier Newnes
سنسور مادون قرمز	غلامرضا حسن زاده مجید شاهوردی		دانشگاه آزاد تبریز
سخت افزار و نرم افزار LOGO	شرکت نیکوپترو پرداز		www.npp.ir
مدار منطقی	بابک یاری محمد نیک روان		مدرسان شریف
جزوه آموزشی ابزار دقیق	دکتر طباطبایی		دانشگاه فردوسی
PLC Siemens	فرهنگ انصاری		www.electronic-blogfa.com
PLC500 عملیاتی	عباس خسروبیگی		دانشکده علوم دریایی محمودآباد
برنامه نویسی ساختاریافته PLC	محمدرضا کاکاوند		www.MRKakavand.tk
کنترلرهای منطقی برنامه پذیر	مقداد بنام		مرکز آموزش عالی مهاجر اصفهان
Automating Manufacturing Systems	Jack Hugh	وحید کارگرمقدم	
نحوه استفاده و آشنایی با PLC های زیمنس	محمد یادگار		شرکت ایران خودرو

تماس با مؤلف : moghadam _ plc@yahoo.com