به نام خدا

برنامه نویسی ساختار یافته‌ی

PLC

S5

کنترل کننده‌های منطقی قابل برنامه‌ریزی

زبان S5

گردآورنده:

محمدرضا کاکاوند

فهرست

مقدمه ۳
بخش اول – مفاهیم ٤
۱- مبناهای عددی ٤
۲- منطق دودویی ٥
۳- فلیپ فلاپ‌ها ٥
٤- اجزای حافظه ٥
٥- سیتم‌های کنترل ٦
بخش دوم - ساختار PLC ۷
سخت‌افزار ۷
PII ۸
PIO ۸
فلگ‌ها تایمرها شمارنده‌ها ۸
بخش سوم – نوشتن برنامه با زبان S5 ..... ۹
عملکرد(Operation) ۹
عملوند(Operand) ۹
نحوه‌ی آدرس دهی عملوندها ۱۰
روش نمایش STL ۱۱
سیکل زمانی اجرای برنامه ۱۱
برنامه‌نویسی سازمان یافته ۱۱
دستورالعمل‌های PLC ۱٤
ضمیمه ۲۲
منابع ۲٤

مقدمه

با رشد استفاده از قطعات الکترونیکی به جای قطعات مکانیکی استفاده از تابلوهای برق و مدارات پیچیده‌ی فرمان کم‌کم منسوخ گشت. PLC را شاید بتوان محصول همین تغییر و تحول دانست وسیله‌ای که امروزه در اکثر محیط‌های صنعتی نقش مهمی بازی می‌کند.

PLC دارای معانی مختلفی در زمینه‌ی برق می‌باشد. از جمله انتقال اطلاعات در خطوط انتقال نیرو که بیشتر در نیروگاه‌ها مورد استفاده است. امّا آنچه در این‌جا مورد نظر است دستگاهی است که با یک ریزپردازنده، حافظه و سایر قطعات که به تفصیل خواهد آمد، خطوط و پروسه‌های صنعتی را تحت کنترل در خواهد آورد. این سه حرف مخفف عبارت Programmable Logic Controller می باشد. که معنی کنترل‌کننده‌های منطقی برنامه پذیر را می‌دهد.

PLC نوع ساده‌ای از کامپیوتر است که برنامه‌ای خاص را دریافت و اجرا می‌کند. با توجه به این‌که امروزه به شدت در صنعت مورد استفاده قرار می‌گیرد در این نوشته سعی شد به صورت اجمالی مفاهیم مهم و اصلی آن مطرح شود اما آنچه تحت این عنوان در مراکز آموزش فنی و حرفه‌ای و سایر آموزشکده‌ها تدریس می‌شود مشتمل بر اعداد باینری و تبدیل اعداد و مفاهیم مربوط اصول اولیه‌ی رایانه است که در این‌جا با توجه به مخاطب آشنا به رایانه و اینترنت از آن صرف‌نظر شده است.

این جزوه می‌تواند برای دریافت مدرک بخشی از مهارت و حتی مهارت سازمان فنی و حرفه‌ای می تواند مورد استفاده قرار گیرد. که امیدوارم رضایت خاطر شما را کسب کند. در آخر از همه اساتید، صاحب‌نظران ، دانشجویان و کارآموزان عزیز خواهشمند است نظر اصلاحی خود را درباره مطالب این جزوه با نشانی MRK@MRKakavand.tk با من در جریان بگذارند.

محمّدرضا کاکاوند

بهمن ۱۳۸۵

بخش اول – مفاهیم

۱– مبناهای عددی: در محاسبات عددی روزمره از اعداد ۰–۹ استفاده می‌کنیم. در ریاضی این اعداد را در مبنای ۱۰ می نامیم (ده‌دهی). اما در کامپیوتر و دستگاه‌های دیجیتالی از اعداد در مبنای ۲ استفاده می‌شود که شامل ۰ و ۱ می‌شود.

هر رقم در مبنای دو (دودویی) یک بیت است. که صفر برای خاموش و یک برای روشن بودن آن بیت مورد استفاده است. n بیت را به ۲^n حالت مختلف می‌توان نشان داد. برخی از اعداد ده‌دهی(decimal) در مبنای دودویی(binary) :

$5_{(10)} = 0101_{(2)}$ $9_{(10)} = 1001_{(2)}$ $8_{(10)} = 1000_{(2)}$

البته مبانی دیگری چون هشت‌تایی (Octal) و یا شانزده (Hexadecimal) هم وجود دارند. و البته برای هر عددی می‌توان مبنای خاص آن عدد را تعریف کرد و روابط مربوط به آن را همانند مبنای ۲ استخراج کرد. اعداد کد شده در مبنای ۲ را BCD گویند.

برای رسانده عددی از مبنای بالاتر به مبنای پائین‌تر عدد را بر مبنای پایینی تا جای ممکن تقسیم می‌کنیم ، سپس از آخرین خارج‌قسمت به سمت اولین باقی‌مانده اعداد را به ترتیب می‌نویسیم. این عدد در مبنای پائین‌تر است. به عنوان مثال:

$41 \div 2 = 20 + \underline{1}$
$20 \div 2 = 10 + \underline{0}$
$10 \div 2 = 5 + \underline{0}$
$5 \div 2 = 2 + \underline{1}$
$2 \div 2 = \underline{1} + \underline{0}$

$41 = (101001)_2$

برای بردن عدد به مبنای بالاتر با توجه به مبنای آن عدد، ارزش مکانی هر رقم را با رساندن مبنا به توان آن مکان معین و سپس با ضرب در رقم و جمع کل ارقام حاصل شده رقم در مبنای ۱۰ ظاهر می‌شود. به عنوان مثال:

$$(101001)_2 = 2^0\times1 + 2^1\times0 + 2^2\times0 + 2^3\times1 + 2^4\times0 + 2^5\times1 = 41$$

۲- منطق دودویی: برای هر عدد در مبنای دودویی سه نوع عملیات منطقی اصلی وجود دارد:

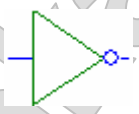
الف) AND که با ضرب بولی سازگار است.

⊙	۰	۱
۰	۰	۰
۱	۱	۱

ب) OR که با جمع بولی سازگار است.

+	۰	۱
۰	۰	۱
۱	۱	۱

پ) NOT که صفر را یک و یک را صفر می‌کند.



x	۰	۱
x′	۱	۰

۳- فلیپ فلاپ‌ها: سلول‌های دودویی که فقط ۱ بیت اطلاعات ذخیره می‌کند. هر فلیپ فلاپ دارای دو ورودی S و R و دو خروجی Q و Q′ می‌باشد.

٤- اجزای حافظه: هر Bit (binary digit) یک ۰ یا ۱ در یک سیستم دیجیتال است. بیت کوچکترین واحد اطلاعات است. هر ۸ بیت یک بایت (Byte) را تشکیل می‌دهند. و هر ۱۰۲٤ بایت یک کیلوبایت را تشکیل می‌دهند. و همین ترتیب مگا، گیگا و ترا بایت را هم می‌توان محاسبه کرد. هر بایت می‌تواند یک حرف الفبا یا یک رقم(۰-۹) ارائه دهد.

Bus: مجموعه‌ای از سخت افزار که وسیله انتقال اطلاعات را به عهده دارد.

Memory: حافظه که انواع مختلفی دارد، و برای ذخیره‌ی اطلاعات به کار می‌رود.

RAM: (حافظه با دسترسی تصادفی) در هر لحظه می‌توان اطلاعات را از هر جای آن فراخواند. و به چند نوع فقط خواندنی(ROM)، اغلب خواندنی(RMM)و اغلب نوشتنی(RWM) تقسیم می‌شود.

۵- سیستم‌های کنترل: برای کنترل یک دستگاه از سیستم‌های کنترل استفاده می‌شود. در گذشته سیستم‌های کنترل اکثراً دستی بود و نیاز به نیروی انسانی و صرف انرژی بالایی داشته‌است. روند گذار از کلید به کنتاکتور، بردهای الکترونیکی، میکروپرسسورها و سرانجام PLC که در این‌جا مورد بحث و بررسی هستند.

ساختار سیستم‌های کنترل:

الف)سیستم‌های کنترل حلقه باز(Open Loop): در این نوع سیستم ساختار تحت کنترل توسط یک سیگنال خارجی کنترل می‌شود که عموماً بر اساس تخمین و ترجیح کاربر است مثل میزان تمیزی لباس‌ها در یک ماشین لباسشویی.

ب)سیستم‌های کنترل حلقه بسته(Closed Loop): در این نوع سیستم کنترل دقیق‌تر بوده توسط سنسورها و دستگاه‌های پردازشگر انجام می‌پذیرد.

انواع سیستم‌های کنترل:

الف) سیستم‌های کنترل سخت‌افزاری: شامل مداراتی هستند که با استفاده از رله‌ها، دیودها، ترانزیستورها ساخته می‌شوند. برنامه‌ی کنترل در این سیستم‌ها به راحتی قابل تغییر نمی‌باشد.

ب) سیستم‌های کنترل نرم‌افزاری: شامل حافظه‌ای برای ذخیره برنامه کنترل در آن هستند. مهمترین مزیت آن‌ها در این است که نحوه‌ی کنترل در آن را بدون نیاز به تغییر در سخت افزار سیستم می‌توان عوض کرد.

بخش دوم – ساختار PLC

می‌توان PLC را کنترل کننده‌ای نرم‌افزاری دانست که اطلاعات را از ورودی به صورت Binary دریافت نموده و طبق برنامه‌ای که در حافظه دارد پردازش می‌کند و نتیجه را از طریق خروجی خود به اجرا کننده‌های فرمان می‌سپارد. برای این منظور در PLC واحدی به نام CPU وجود دارد و می‌تواند با استفاده از منطق برنامه نویسی تعریف شده برای آن ورودی‌های دریافتی را کنترل و سپس پردازش کند و سپس خروجی‌ها را فعال یا غیرفعال نماید.

استفاده از کامپیوتر به جای PLC بسیار وقت گیر و مستلزم آموزش‌های خاص و دستگاه‌های ویژه است که عملاً ناممکن می‌نماید. PLC توسط شرکت‌های مختلفی چون LG, OMRON, AEG, SIEMENS و... ساخته شده است و هر شرکتی زبان برنامه نویسی خاص خود را تعریف نموده است. از انواع داخلی هم PLC شرکت کنترونیک را می توان نام برد. اما از آنجا که PLCهای شرکت زیمنس فراگیری بیشتری دارد، زبان Step 5 (S5) که مربوط به شرکت نامبرده می‌باشد برای آموزش انتخاب شد. لازم به ذکر است که هم‌اکنون درPLCهای جدید زبان S7 مورد بهره‌برداری است که از لحاظ مفاهیم کلی با S5 قابل فراگیری است.

سخت افزار:

۱- منبع تغذیه PS (Power Supply): 24V DC or 110-220V AC
۲- پردازشگر مرکزی CPU (Central Processing Unit)
۳- ترمینال‌های ورودی Input Module که به دونوع دیجیتال و آنالوگ است.
٤- ترمینال‌های خروجی Output Module که به دونوع دیجیتال و آنالوگ است.
۵- حافظه
٦- ارتباط پردازشگرها CP (Communication Processor)
۷- رابط IM (Interface Module)

PII: قبل از اجرای برنامه CPU تمام ورودی‌ها را بررسی می‌کند و در قسمتی از حافظه به نام Process Image Input (PII) ذخیره می‌کند. در حین اجرای برنامه CPU به جای ورودی‌ها به این حافظه مراجعه می‌کند.

PIO: (Process Image Output) هرگاه در حین برنامه یک مقدار برای خروجی بدست آید، در این قسمت حافظه نگهداری می‌شود. و در هنگام اجرای برنامه CPU به جای خروجی‌ها به این حافظه مراجعه می‌کند.

فلگ‌ها، تایمرها و شمارنده‌ها: فلگ Flag محلی از حافظه است که جهت نگهداری بعضی از نتایج یا خروجی‌ها استفاده می‌شود. جهت شمارش از شمارنده‌ها Counter و جهت زمان‌سنجی از تایمرها Timer استفاده می‌شود. در آینده مطالب بیشتری در این باب خواهیم آورد.

بخش سوم – نوشتن برنامه به زبان S5

برنامه را به سه صورت زیر می‌توان نوشت:

۱- عبارتی STL (Statement List)

۲- نردبانی LAD (Ladder)

۳- فلوچارتی CSF (Control System Flowchart)

البته روش دیگری به نام GRAPH5 هم وجود دارد که در این‌جا عنوان نمی‌شود.

در نمایش نردبانی برنامه به صورت نماد اتصال و سیم پیچ‌های مدارفرمان رله‌ای نشان داده می‌شود لذا ساختار برنامه شبیه مدارهای فرمان رله‌ای می‌باشد. در نمایش فلوچارتی از نمادهای مستطیل شکل (بلوک) استفاده می‌شود. این طرز نمایش بیشتر در هنگام طراحی برنامه استفاده می‌شود. در هربلوک عمل منطقه‌ای نشان داده می‌شود. در روش عبارتی از دستورات و جملات نوشتاری برای نوشتن برنامه بهره می‌گیرند. هر عبارت یا statement دارای دو بخش است:

الف- عملکرد (Operation): به عمل منطقی‌ای که در عبارت صورت می‌گیرد عملکرد گفته می‌شود. عملکردهای مهم در جدول زیر نمایش داده شده است.

در زبان برنامه‌نویسی	در زبان محاوره‌ای		در ریاضی
A	«و»	AND	ترکیب عطفی
O	«یا»	OR	ترکیب فصلی
AN	نقیض «و»	AND NOT	نقیض ترکیب‌عطفی
ON	نقیض «یا»	OR NOT	نقیض ترکیب‌فصلی
=	مساوی	ASSIGN TO	ترکیب هم‌ارزی

ب- عملوند(Operand): به قسمتی از عبارت که یک عمل منطقی روی آن انجام شود. مانند ورودی‌هاI ، خروجی‌هاQ ، فلگ‌هاF و... که نحوه‌ی آدرس دهی آن‌ها در آینده بیان می‌شود. عملوندهای مورد استفاده در زبان S5 در زیر آمده است:

I	Inputs	ورودی‌ها، از پروسه‌ی تحت کنترل به PLC
Q	Outputs	خروجی‌ها، ازPLC به پروسه‌ی تحت کنترل
F	Flags	فلگ‌ها، حافظه‌ای جهت نگهداری مقادیر میانی حاصل از عملیات باینری
D	Data	دیتا، حافظه‌ای جهت نگهداری مقادیر میانی حاصل از عملیات دیجیتالی
T	Timers	زمان‌سنج‌ها، حافظه‌ای جهت تخصیص زمان‌سنج‌ها
C	Counters	شمارنده‌ها، حافظه‌ای جهت تخصیص شماش‌گرها
K	Constants	ثابت‌ها

نحوه‌ی آدرس‌دهی عملوندها: هر کدام از ورودی‌ها، خروجی‌ها و فلگ‌ها در دسته‌های ۸ بیتی (تک بایتی) سازمان‌دهی می‌شوند. و در آدرس‌دهی ابتدا باید آدرس بایت مربوط و سپس آدرس بیت تعیین شود. در زیر مثال‌هایی از این باب آورده شده است:

اگر بایت ورودی هشتم (IB 8) به صورت زیر باشد:

0	0	1	0	1	1	0	1
۷	٦	٥	٤	۳	۲	۱	۰

پس I 8.5 (یعنی بیت پنجم از بایت هشتم ورودی) مقداری برابر ۱ خواهد داشت.

و اگر بایت ششم فلگ (FY 6) به صورت زیر باشد:

0	1	1	0	0	1	1	1
۷	٦	٥	٤	۳	۲	۱	۰

F 6.1 (یعنی بیت اول از بایت ششم ورودی) مقداری برابر ۰ دارد.

به همین صورت بایت‌های خروجی (QB) نیز نامگذاری می‌شوند.

روش نمایشSTL: به هر دستور یک رشته (خط) برنامه یا Statement گفته می‌شود. هر دستور یا خط برنامه معمولاً یکی از ترکیب‌های منطقی ریاضی (= , AND , OR , NOT و...) را در بر دارد. و همچنین کنترل فلگ‌ها و فلیپ فلاپ‌ها را نیز بر عهده دارند. در برنامه‌نویسی به روش STL هر چند خط برنامه که عمل خاصی را انجام می‌دهد یک Segment می‌گویند. یک برنامه می‌تواند شامل یک Segment یا بیشتر باشد.

هر برنامه با یک « ; » شروع شده و با«BE» Block End خاتمه می‌یابد. برنامه نوشته شده با CSF یا LAD را می‌توان به STL تبدیل نمود. اما برنامه‌ی نوشته شده با STL را لزوماً نمی‌توان به LAD یا CSF تبدیل کرد.

سیکل زمانی اجرای برنامه(Cycle time): ریز پردازنده از سطر اول برنامه شروع به خواندن و اجرای دستورات می‌کند. تا به دستور BE برسد. مدت زمان لازم برای این‌کار را سیکل زمانی اجرای برنامه می‌گویند. برای تسریع در اجرای برنامه و کاهش این سیکل زمانی می‌توان پردازنده‌ای با سرعت بالا به کار برد. یا برنامه را سازماندهی کرد(Structure Programming).

برنامه نویسی سازمان یافته: در نوشتن برنامه‌های پیچیده که معمولاً طولانی هستند. برنامه‌های فرعی را در بخش‌های جداگانه می‌نویسند و سپس آن‌ها را در برنامه‌ی اصلی به کار می‌برند. هر کدام از این قسمت‌ها را در یک بلوک می‌نویسند.

۱- بلوک‌های برنامه PB(Program Block): تشکیل‌دهنده‌ی برنامه‌ی کنترل یک فرآیند می‌باشند که از شماره‌ی ۰-۲۵۵ شماره گذاری شده‌اند. کاربرد برنامه را به تشخیص خود در هر بلوک PB می‌نویسیم و آخر برنامه BE می‌زنند.

۲- بلوک‌های ترتیبی SB(Sequence Block): در کنترل‌های ترتیبی مثل راه‌اندازی بخش‌های خط تولید، استفاده می‌شود.

۳- بلوک‌های تابع ساز FB(Function Block): توابعی که در طول برنامه بارها مورد استفاده هستند و در خود برنامه تعریف نشده‌اند مثل ضرب دو عدد باینری؛ که از شماره‌ی ۰-۲۵۵ شماره‌گذاری شده‌اند.

هر FB از دو بخش تشکیل شده‌است:

الف) سر خط بلوک(Block Header): شامل نام و سایر مشخصات FB

ب) بدنه‌ی بلوک(Block Body): شامل توابع و دستوراتی که باید در FB اجرا شود. علاوه بر دستورات S5 یک سری دستورات مخصوص Supplementary نیز موجود است که تنها در FBها اجرا می‌شود.

دو نوع FB وجود دارد:

۱)Standard FB: که در آن‌ها اعمال منطقی نظیر ضرب و تقسیم و... تعریف شده‌است و آن‌ها به صورت بسته‌های نرم افزاری در اختیار کاربر قرار می‌گیرد.

۲)Assignable FB: در اجرای این نوع FB می‌توان عملوندها را در هر پروسه تعیین نمود، تعریف کرد، یا تغییر داد.

لازم به ذکر است که FBها فقط به طریقه‌ی STL قابل برنامه نویسی هستند.

٤- بلوک‌های اطلاعاتی DB(Data Block): ۲۵٦ بلوک برای ذخیره‌ی اطلاعاتی که هنگام اجرای برنامه مورد استفاده‌اند همچون پیغام‌ها، هشدارها و... در نظر گرفته شده‌اند.

سه نوع اطلاعات در بلوک‌های DB وجود دارند:

۱- اطلاعات DATA

۲- متن TEXT

۳- الگوی بیت BIT PATTERN: تعدادی بیت ۰و۱ که به‌صورت بایتی و یا کلمه‌ای است و عمل سیگنالینگ یا روشن و خاموش کردن خروجی را بر عهده دارد.

می‌توان در هر بلوکی اطلاعات DBها را فرا خوانی نمود. برای فراخوانی سطر صدم از DB 50 به صورت زیر عمل می‌کنیم:

C DB 50 نام بلوک

L DW 100 نام سطر

اطلاعات ذخیره شده در DB ها به یکی از فرمت‌های زیر هستند:

۱- KH: برای اعداد در مبنای ۱۶ ($16Bits: 0000_{(H)} \rightarrow FFFF_{(H)}$)

۲-KF: برای اعداد در مبنای ۱۰ (16Bits: -32768 → 3278)

۳- KT: برای اعداد ثابت TV(زمان)(14Bits: 001.0 → 999.3)

٤- KC: برای شمارنده‌ها(12Bits: 000 → 999)

٥- KY:١٦بیت دو بایت چپ و راست تقسیم می‌شوند.DL و DR که کاملاً مجزا از هم هستند.

٦- KM:$16Bits: (00...0)_{(16Bits)} \rightarrow (11...1)_{(16Bits)}$

۷- KG:اعداد اعشاری و اعداد بسیار بزرگ و بسیار کوچک (32Bits:Dmord)

۸- KM: برای متون.

٥- بلوک سازماندهی OB(Organization Block): ساختار برنامه را مشخص می‌کنند.هر OB با یک شماره‌ی خاص مشخص می‌شود:

OB 1: در شروع هر سیکل برنامه، سیستم عامل اولین سطر این بلوک را اجرا می‌کند. و آخرین سطر آن پایان بخش برنامه است. در واقع مشخص کننده‌ی ساختار برنامه است.

OB 21: هنگامی که PLC از STOP به START سوئیچ می‌شود، این بلوک رخ می‌دهد.

OB 22: هنگامی که Power، On می‌شود این برنامه رخ می‌دهد.

OB 34: نشان‌دهنده‌ی وضعیت باتری است. که در صورت تضعیف و یا وقوع ایراد در آن، تا رفع اشکال مکرراً تکرار خواهد شد.

نکته: برنامه‌های کمتر از ۵۰۰ سطر را می‌توان در OB 1 نوشت.

دستورالعمل‌های PLC :

۱- اصلی(Basic): توابعی که در تمام بلوک‌ها قابل اجرا هستند. بغیر از جمع (+F) و تفریق (-F) تمام دستورها می‌توانند به عنوان ورودی و خروجی در روش‌های برنامه‌نویسی STL ، LAD و CSF به کار روند.

۲- تکمیلی(Supplemental): توابع ترکیبی نظیر دستورات جابجایی، توابع Shift و نیز دستورات تبدیلی می‌باشند. که فقط در FB و در حالت STL قابل اجرا هستند.

۳- سیستم(System): دستوراتی که مستقیماً روی سیستم عامل PLC تاثیر دارد و مخصوص برنامه نویسان حرفه‌ای است.

خواندن صفر (Scanning for zero): برای خواندن عدد0 در ورودی از دستور AN (AND NOT) استفاده کرد. که (0یا1) در ورودی به صورت (1یا0) در می‌آید. در LAD ⊣/⊢ و در CSF —|NOT|— .

کنتاکت در حالت عادی باز(NO): وقتی دکمه‌ی فشاری(Push Button) فشرده یا کلیدی روشن گردد در ورودی ۱ ظاهر می‌شود و بر عکس. ⊣ ⊢

کنتاکت در حالت عادی بسته(NC): وقتی دکمه‌ی فشاری(Push Button) فشرده یا کلیدی روشن گردد در ورودی ۰ ظاهر می‌شود و بر عکس (که از دستور ANاستفاده می‌کنیم)⊣/⊢

حالا با ذکر چند مثال برنامه‌نویسی با زبان S5 را آغاز می‌کنیم.

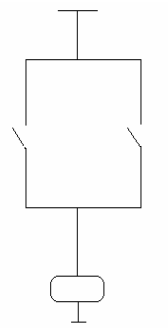
مثال: برنامه ای بنویسید که با دو کلید A وB که به صورت سری به هم وصلند خروجی را روشن و خاموش نماید.

حل:

```
A   I 0.1
A   I 0.2
=   Q 0.0
BE
```

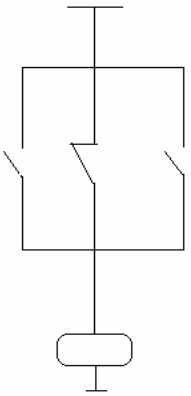
مثال: برای دو کلید A و B که بصورت موازی به هم وصل شده‌اند خواهیم نوشت:

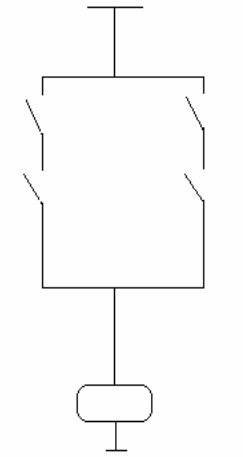
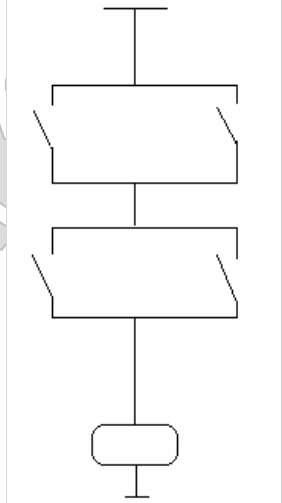
حل:

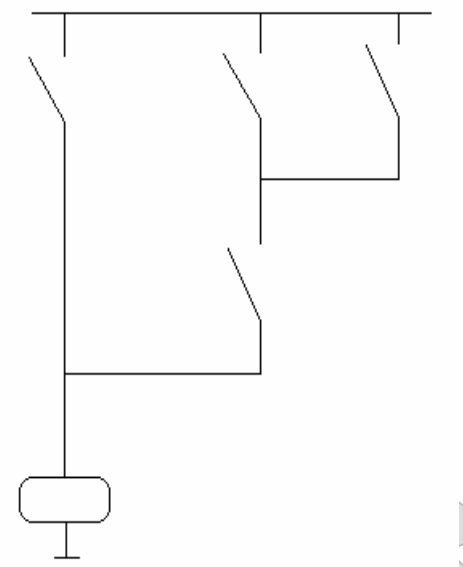


```
O   I 0.0
O   I 0.5
=   Q 0.0
BE
```

مثال: برای مدارهای شکل زیر می‌توان نوشت:

۱		```A I 0.0 AN I 0.1 = Q 0.2 BE```
۲		```O I 0.1 ON I 0.2 O I 0.0 = Q 0.1 BE```

۳		AN I 0.0 A I 0.2 AN I 0.3
٤		A I 1.1 A I 1.2 O A I 1.0 A I 1.3 = Q 3.1
٥		A(O I 1.4 O I 1.5) A(O I 2.0 O I 2.1) = Q 3.0 BE

O I 0.0 O A I 0.1 A(O I 1.4 O I 1.4) = Q 2.1 BE		٦

در دو مثال آخر کاربرد پرانتز (کمانک) در برنامه‌نویسی مشخص شد. که برایOR قبل از AND لازم است. امّا برای AND قبل از OR نیازی نیست. البته این گونه مدارات بدون پرانتز و با استفاده از فلگ‌ها قابل نوشتن است.

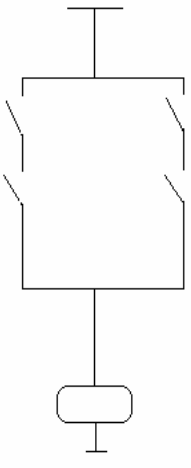
فلگ (Flag) یا پرچم: هر فلگ یک بیت از حافظه‌ی PLC است که آن‌را می‌توان معادل خروجی مجازی دانست. این بیت مانند هر بیت حافظه می‌تواند دومقدار 0 یا 1 را بگیرد. با این تفاوت که فلگ‌ها حافظه‌های موقتی هستند.

آدرس دهی فلگ‌ها همانند ورودی‌ها و خروجی‌هاست که به تفصیل به آن اشاره شد. به عنوان مثال F 8.0 مقدار بیت هشتم از بایت صفر است.

کاربرد فلگ‌ها در برنامه‌هایی است که OR قبل از AND دارد. با حذف پرانتزها می‌توان از فلگ استفاده کرد. البته گاهی ممکن است برنامه طولانی‌تر شود. مثلاً مثال بالا را با فلگ‌ها بازنویسی می‌کنیم:

مثال:

```
O   I 1.4
O   I 1.5
=   F 6.0
O   I 2.0
O   I 2.1
=   F 6.1
```



```
A    F 6.0
A    F 6.1
=    Q 3.0
BE
```

بیت RLO (Result of Logic Operation):PLC در اجرای هر خط برنامه مقدار حاصل از اعمال منطقی را در بیتی به‌نام RLO قرار می‌دهد و در اجرای سطر بعدی این مقدار با عملوند بعدی طبق برنامه ترکیب و مقدار حاصل در RLO جایگزین می‌شود. این عمل تا رسیدن به خط دستور هم ارزی (=) ادامه پیدا می‌کند. در این هنگام RLO مقدار خود را از دست داده و پذیرای مقدار جدید می‌شود.

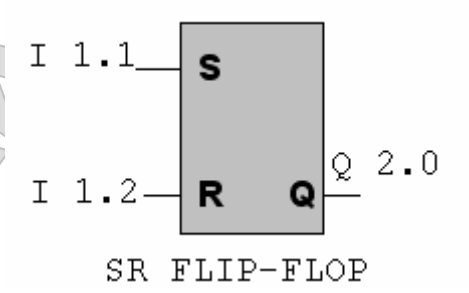
سِت(set) **و ری‌سِت**(reset) در فلگ‌ها و خروجی‌ها: *فلیپ فلاپ* شامل دو ورودی Set و Reset می‌باشد. دو نوع فلیپ فلاپ وجود دارد:

۱- فلیپ فلاپ SR

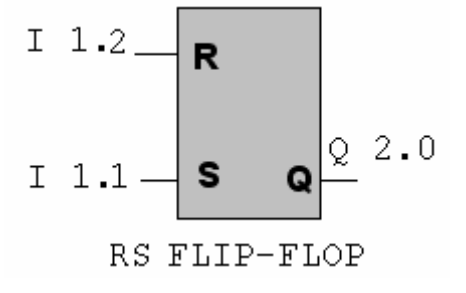
۲- فلیپ فلاپ RS

که تفاوت آن‌ها در ارجحیّت ورودی‌های سِت و ری‌سِت است:

```
A    I 1.1
S    Q 2.0
A    I 1.2
R    Q 2.0
BE
```




```
A    I 1.2
R    Q 2.0
A    I 1.1
S    Q 2.0
BE
```



در فلیپ فلاپ‌های SR هنگامی که ورودی R در حالت 0 باشد کافی است در یک لحظه ورودی S در حالت 1 قرار بگیرد تا خروجی بصورت پایدار 1 شود. این

وضعیت مادامی که R بصورت 0 است باقی خواهد ماند. در این فلیپ فلاپ اگر هر دو ورودی برابر 1 باشد ارجحیت با دستور دوم است. چرا که دستور دوم ناقض دستور اول است و PLC دستورات را سطر به سطر اجرا می‌کند. با این بیان می‌توان اصل کلی زیر را نتیجه گرفت: «هر دستوری که به خط پایان برنامه BE نزدیک باشد از نظر اجرا ارجح‌تر است.»

دستور NOP 0(در روش STL): هر گاه بخواهیم از خروجی یک فلیپ فلاپ یا قسمتی از برنامه هیچ استفاده‌ای نکنیم. از دستور NOP 0 استفاده می‌کنیم. باید متذکر شد که این دستور مختص PLCهای زیمنس است.

مثال:

```
A   I 2.3
S   Q 3.4
A   I 2.4
R   Q 3.5
NOP 0
```

می‌توان خروجی یک فلیپ فلاپ را یک فلگ قرار داد.

مثال:

```
A   I 0.1
S   F 2.7
A   I 0.7
R   F 2.7
A   F 2.7
=   Q 3.4
BE
```

دستور پرش غیرشرطی(JU): همان‌طور که گفته شد نتیجه‌ی عملکرد دستورات هر خط در بیت خاصی به نام RLO ذخیره می‌شود. دستورات می‌توانند به بیت RLO وابسته باشند یا نباشند. که اگر وابسته نباشند غیرشرطی خواهند بود. JU بدون وجود هرگونه شرطی، پرش یا انتقال را انجام می‌دهد. این پرش ممکن است از

یک بلوک به بلوک دیگر، یا از یک سطر بلوک به سطر دیگر همان بلوک انجام گیرد. وابسته بودن بدین معنی است که با وجود RLOای مساوی 1 دستور اجرا می‌شود.

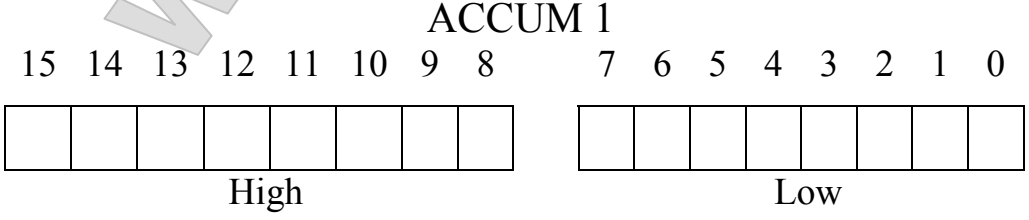
دستور پرش شرطی (JC): این دستور وابسته به بیت RLO است. و همانند JU عمل پرش را انجام می‌دهد. اما بصورت وابسته به بیت RLO .

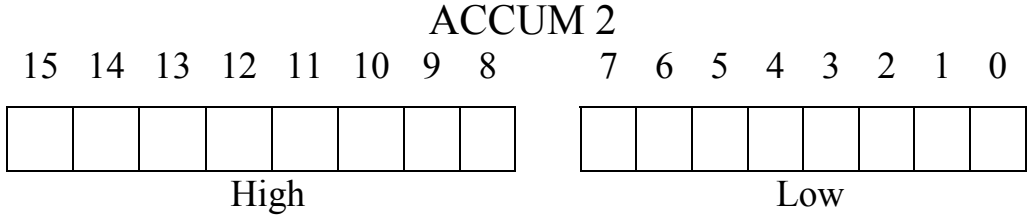
مثال: برنامه‌ای بنویسید که با فشردن یک کلید (فعال نمودن یک کلید) PB 18 و در صورت غیرفعال نمودن همان کلید PB 19 را اجرا نماید.

حل: با اندکی تفکر درمی‌یابیم که چنین برنامه‌ای را باید در OB 1 نوشت. زیرا همانطور که گفته شد ساختار کلی سیستم در این بلوک شکل می‌گیرد. همچنین باید از دستور پرش شرطی استفاده کرد.اگر فرض کنیم کلید گفته شده در صورت سوال I 0.0 باشد:

```
A   I 0.0
JC  PB 18
AN  I 0.0
JC  PB 19
BE
```

دستورهای بارگذاری و انتقال: جهت مبادله‌ی مقادی ورودی‌ها، خروجی‌ها یا فلگ‌ها نیاز به یک حافظه‌ی واسطه می‌باشد. که از قسمتی که به آن آکومولاتور (Accumulator) یا انباره (انبارک) می‌گویند، وجود دارد. این حافظه از نوعRegister و ۱۶ بیتی هستند. که معمولاً شامل ۱۶ بیت یا دو بایت با ارزش بالا(High) و ارزش پایین(Low) می‌باشد.





دستور L(Load): برای بارگذاری اطلاعات از این دستور استفاده می‌کنیم. به این ترتیب محتویات یک بایت اعم از کلمه یا اعداد فراخوانی و در انبارک جایگزین می‌شود.

L IB 4 , L KD 5 , L KH 3 , L FY 5 , ...

اگر PLC ما دو انبارک داشته باشد، با دستور L IW 4 شانزده بیت موجود در کلمه‌ی ورودی شماره‌ی ٤ (یعنی بایت ٤ و ٥) را به ACCUME 1 می‌فرستد. اگر در همین حالت L IW 6 اجرا شود اطلاعات ACCUME 1 به ACCUME 2 رفته و IW 6 به ACCUME 1 منتقل می‌شود. و باز اگر در همین حالت IW 12 اجرا شود محتویات ACCUME 2 به دور ریخته شده و ACCUME 1 جایگزین آن می‌شود. و در ACCUME 1 ، IW 12 جایگزین می‌شود.

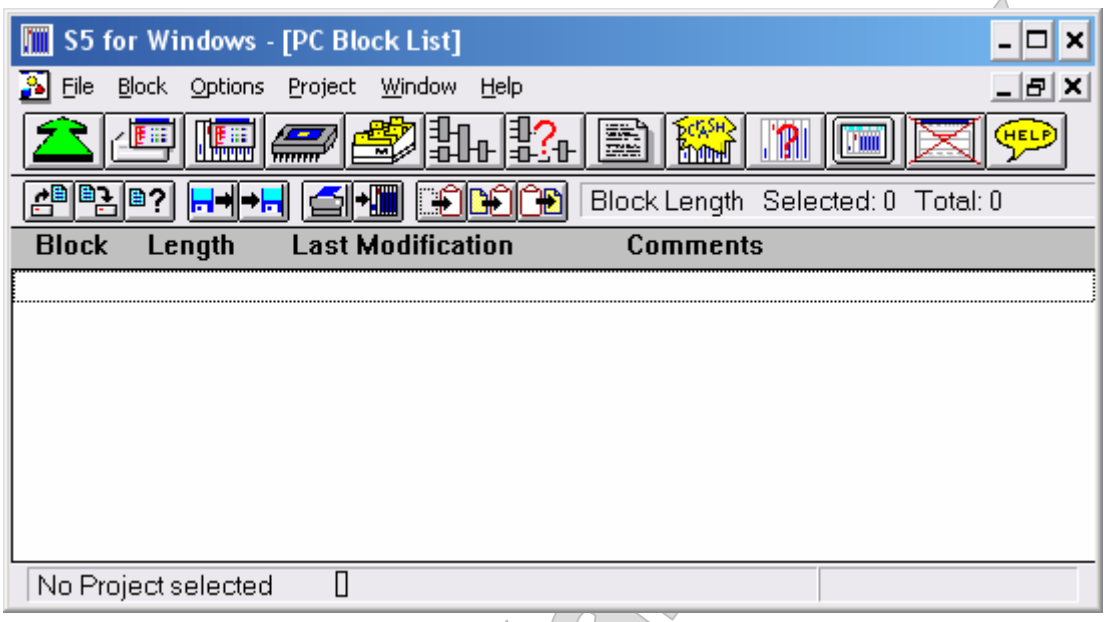
دستور T (Transfer): برای انتقال اطلاعاتی که در انبارک‌ها موجود است به خروجی‌ها یا فلگ‌ها از این دستور استفاده می‌شود.

T QW 8 , T FW 52

با اجرای دستور T QW 8 محتویات ACCUME 1 به کلمه‌ی خروجی۸ کپی می‌شود (منتقل نشده و از بین نمی‌رود). دستورات L و T به RLO وابسته نیستند و لذا غیر شرطی خواهند بود.

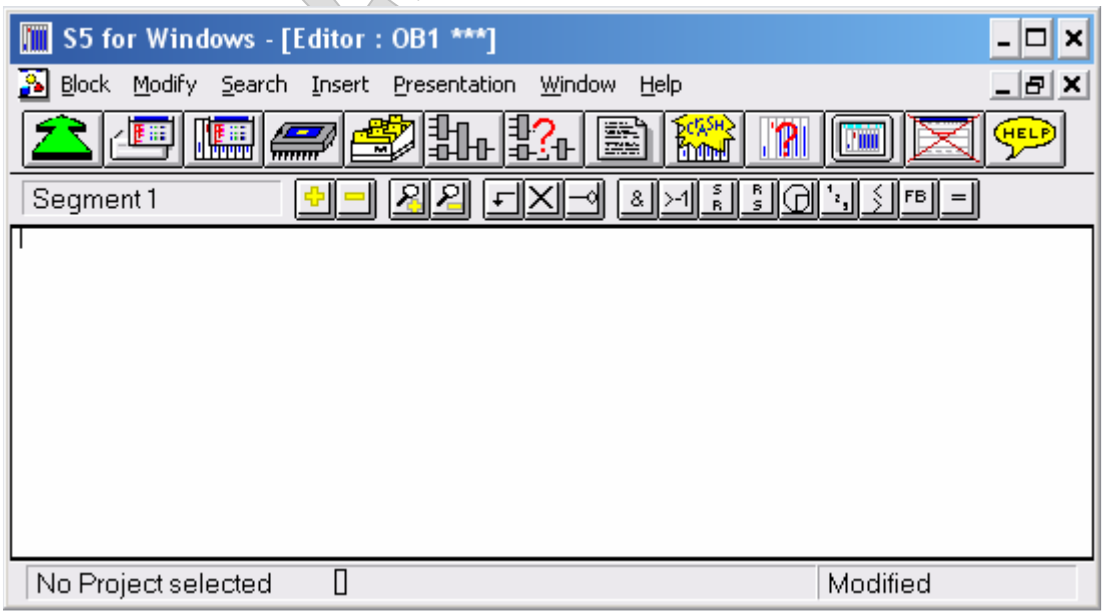
ضمیمه

نرم افزار s5w-demo ساختار سخت‌افزار و نرم‌افزار یک سیستم PLC زیمنس را شبیه سازی می‌کند. این نرم افزار به روی سه عدد فلاپی ذخیره شده است و می‌توان به جای فلاپی از نسخه‌ی لوح فشرده‌ی آن استفاده کرد.

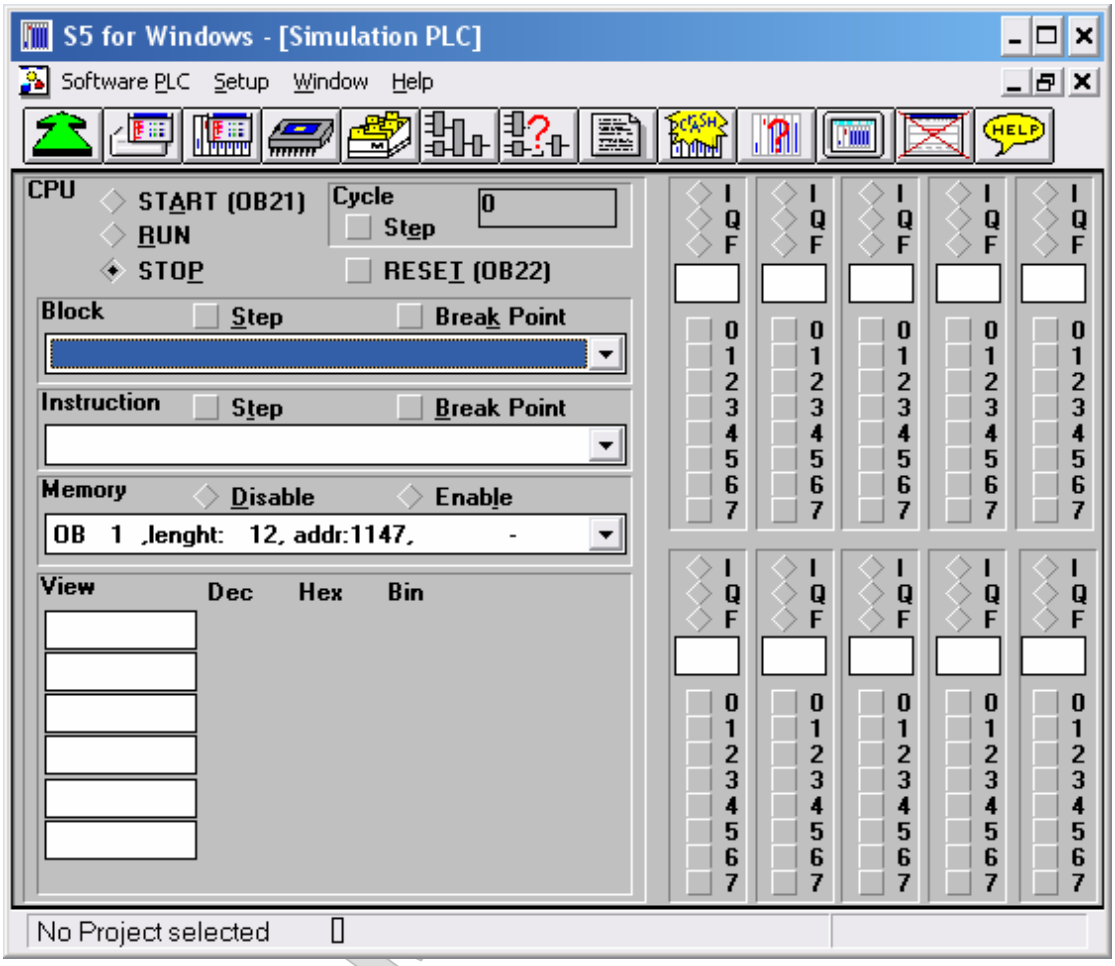


استفاده از این نرم‌افزار بسیار ساده است اما نکاتی به اجمال برای راهنمایی اولیه در زیر خواهد آمد:

منوی Block شامل گزینه‌هایی برای دستکاری بلوک‌هاست. برای ویرایش هر بلوک پس از ایجاد آن پنجره‌ی فوق تغییر می‌کند. از منوی Presentation می‌توان شیوه‌ی نمایش بلوک مورد نظر را انتخاب کرد(STL، CSF، LAD،...)



پس از نوشتن تغییرات لازم با فشردن دکمه‌ی  پنجره‌ی شبیه‌ساز PLC باز می‌شود در این‌جا ورودی‌ها خروجی‌ها و فلگ‌ها مشاهده می‌شوند و می‌توان عملکرد برنامه‌ی نوشته شده را روی PLC آزمود.



استفاده از این نرم‌افزار بخصوص برای کارآموزان که دسترسی به PLC ندارند توصیه می‌شود.

برای استفاده از نرم‌افزار فوق و یادگیری مهارت‌های استفاده از آن احتمالاً نیاز به راهنمایی Help خود نرم‌افزار خواهید داشت. از منویی به همین نام و یا با فشردن کلید F1 ظاهر می‌شود.

منابع

۱. قابوسی، فربد. ۱۳۸۵. مرجع کامل PLC . انتشارات آفرنگ.
۲. رزم، داود. ۱۳۸٤. مرجع PLC . انشارات دیباگران تهران.
۳. مانو، موریس، ترجمه دکتر سپیدنام. ۱۳۸٤. انتشارات خراسان.