کاملترین مرجع کاربردی

PLC S7 SIEMENS

(سطح مقدماتی)

مهندس محمدرضا ماهر
مهندس احمد فرجی

Step 7 | S7 300 | LAD/STL/FBD | Hwconf

DVD

تاریخچه سیستم های کنترل – جایگاه PLC در سیستم های کنترل– ساختار و اجزای PLC – نکات طراحی و انتخابPLC– نکات نصب PLC– آشنایی با سنسورها و عملگرها– آشنایی با کنترلرهای زیمنس و انواع PLC های S7 – نصب و استفاده از نرم افزار Step7 – پیکربندی سخت افزار S7 300 توسط HWconfig– مفاهیم برنامه نویسی – برنامه نویسی با دستورات LAD و FBD – نحوه کار با Modify/ Force – ارتباط Online با PLC و ...

کاملترین مرجع کاربردی

PLC S7

(سطح مقدماتی)

اهدایی
نشر نگارندهٔ دانش

مهندس محمدرضا ماهر
مهندس احمد فرجی

انتشارات نگارنده دانش
تهران - بهار ۹۰

سرشناسه	: ماهر، محمدرضا- ۱۳۴۲
عنوان و نام پدیدآور	: کاملترین مرجع کاربردی *PLC S7* / نویسندگان محمدرضا ماهر، احمد فرجی
مشخصات نشر	: تهران: نگارنده دانش،۱۳۸۹
مشخصات ظاهری	: ۷۳۶ ص.: مصور، جدول، نمودار.
شابک	: ۹۷۸-۶۰۰-۹۱۹۰۰-۱-۳
وضعیت فهرست نویسی	: فیپا
موضوع	: استپ (زبان برنامه‌نویسی کامپیوتر)
موضوع	: کنترل‌کننده‌های برنامه‌پذیر
شناسه افزوده	: فرجی، احمد
رده بندی کنگره	: *TJ* ۲۲۳ک۹م۲۶ ۱۳۸۹
رده بندی دیویی	: ۶۲۹/۸۹۵
شماره کتابشناسی ملی	: ۵۴۴۳۷۲۲

کاملترین مرجع کاربردی PLC S7 (سطح مقدماتی)

نوشتهٔ	مهندس محمدرضا ماهر مهندس احمد فرجی
ویراستهٔ	مهندس علی کلانتری
حروفچینی	واحد تولید
لیتوگرافی	ترام‌سنج
چاپ و صحافی	اپل
چاپ اول	۱۳۹۰
تعداد صفحات، قطع	۷۳۶ ، وزیری
تعداد	۱۱۰۰ نسخه
بها به همراه یک *DVD*	۱۵۰۰۰ تومان

انتشارات نگارنده دانش

تهران، میدان انقلاب، خیابان جمالزاده جنوبی، کوچه دانشور، شمارهٔ ۱۰، واحد۱

تلفن: ۶۶۴۲۵۵۹۵ ، تلفکس: ۶۶۹۴۰۴۲۹

صندوق پستی ۹۱۶- ۱۳۱۴۵

Negarandedanesh.blogfa.com

هرگونه تکثیر یا کپی‌برداری از تمام یا بخشی از مطالب این کتاب بدون اجازه کتبی ناشر و مؤلف ممنوع است.

بنام یگانه منشاء دانش و بینش

پیشگفتار

حدود پنج سال از زمان انتشار دو جلد کتاب راهنمای جامع Step7 تألیف اینجانب که توسط انتشارات شرکت صابکو به چاپ رسید، گذشته است. در کتاب‌های فوق نواقص و کاستی‌های زیادی وجود داشت که متأسفانه ناشر بدون توجه به نظرات مولف، این کتاب‌ها را تجدید چاپ نمود. تماس‌های زیاد خوانندگان کتاب‌های ذکر شده و در خواست‌های متعدد شرکت‌کنندگان در کلاس‌های آموزشی اینجانب، موجب بازنگری و تدوین این کتاب‌ها در یک مجموعه سه جلدی به‌صورت کامل‌تر شامل "سطح مقدماتی، سطح پیشرفته و سطح تکمیلی" و با رفع نواقص قبلی، و چاپ این مجموعه توسط ناشر جدید شد.

با وجود مشغله کاری زیاد ، این کار بدون کمک و یاری همکاران متخصص امکان‌پذیر نبود. خوشبختانه *آقای مهندس احمد فرجی* (مدیر شرکت نامداران نوآور صنعت اهواز) که از متخصصین عرصه PLC هستند و قبلاً کتاب S7-Graph و مجموعه پروژه‌های کاربردی با PLC از ایشان توسط انتشارات نگارنده دانش به چاپ رسیده است، درخواست یاری بنده را پذیرفتند و تدوین بخش‌هایی از کتاب حاضر توسط ایشان با دقت و حوصله فراوان انجام شد.

اما این مجموعه فقط حاصل تلاش مؤلفین نیست و دوستان دیگری نیز در تدوین آن ما را یاری کردند. آقای *مهندس عبدالمجید خدری* ویرایش کلی و آقای *حمید کتیرایی* ویرایش بخش سنسورها و عملگرها (فصل ۵) را انجام دادند. سرکار *خانم الهام نوریان* نیز در کار تدوین و ویرایش کمک شایانی نمودند. زحمات همگی ایشان جای تشکر و قدردانی بسیار دارد. برای ایشان آرزوی موفقیت روز افزون داریم.

چاپ این کتاب را نیز به انتشارات نگارنده دانش با مدیریت آقای *مهندس علی کلانتری* که سه کتاب قبلی مؤلفین را به چاپ رساندند سپرده‌ایم. با دقت نظری که ایشان دارند اطمینان داریم که به یاری خدا اثری با کیفیت و ماندگار عرضه خواهد گردید.

این کتاب ابتدا به تشریح جایگاه PLC در سیستم‌های اتوماسیون پرداخته، مزایا و معایب آنرا با سایر سیستم‌ها بیان می‌کند سپس به شرح اجزای PLC می‌پردازد. سپس بحث را با بیان اصول طراحی و نکات نصب PLC ادامه می‌دهد و قبل‌ازورود به مبحث اصلی، سنسورها و عملگرها را (که به‌عنوان ورودی و خروجی PLC محسوب می‌شوند) در حد مختصر مرور می‌کند. این مطالب در فصل‌های ۱ تا ۵ ذکر شده است و PLC خاصی مورد بحث قرار نگرفته است. از فصل ۶ به صورت تخصصی سیستم‌های کنترل زیمنس و PLCهای این سازنده که در صنایع کشور ما کاربرد وسیعی دارند، مورد

بحث قرار گرفته است. سپس سخت افزار نوع S7-300 با ذکر نکات نصب و پیکربندی توسط Step7 تشریح گردیده است. برنامه‌نویسی با زبان‌های LAD و FBD در ادامه بیان شده است. مطالب مربوط به سخت‌افزار S7-400، برنامه‌نویسی ساختار یافته با FC و FB، برنامه‌نویسی و کار با سیگنال‌های آنالوگ،بحث وقفه‌ها و نحوه کنترل لوپ با PLC به کتاب سطح پیشرفته موکول شده است. در این کتاب سعی شده دستورات برنامه‌نویسی با مثال‌های متعددی تشریح شوند تا مطلب بر خواننده کاملاً قابل لمس باشد و تمریناتی نیز برای کار اضافی روی دستورات طراحی شده است.

امید به اینکه نتیجه تلاش انجام شده مورد رضایت خوانندگان عزیز قرار گیرد. مایه بسی خوشوقتی است که نظرات خود را از طریق ایمیل اینجانب reza.maher@gmail.com یا ایمیل آقای مهندس فرجی ahmad.fa62@gmail.com یا از طریق ناشر با ما در میان بگذارید.

در پایان توصیه می‌کنم خوانندگان محترم از تالار گفتمان اتوماسیون صنعتی که اخیراً در وب سایت اینجانب به آدرس www.indacts.com راه‌اندازی شده است دیدن فرمایند و پرسش‌ها، پاسخ‌ها، نظرات و تجربیات خود را با عموم متخصصین اتوماسیون داخلی به اشتراک بگذراند.

محمدرضا ماهر

فروردین ۱۳۹۰

فهرست

● **فصل ۱: نگاهی به سیستم‌های کنترل و جایگاه PLC** ۱

چکیده مطالب ۴

اصطلاحات و تعاریف ۵

۱-۱ مقدمه ۸

۱-۲ نگاهی به تاریخچه سیستم‌های کنترل ۸

۱-۳ مزایا و معایب سیستم‌های کنترل مدرن ۲۷

۱-۴ مقایسه PLC با سایر سیستم‌های کنترل و مزایا و معایب آن ۲۸

۱-۴-۱ مقایسه PLC با سیستم‌های رله کنتاکتی ۲۸

۱-۴-۲ مقایسه PLC با سیستم‌های DDC ۳۰

۱-۴-۳ مقایسه PLC با سیستم‌های DCS ۳۰

۱-۵ معرفی سازندگان PLC ۳۴

۱-۶ پرسش و تحقیق ۴۲

۱-۷ تست‌های خودآزمایی ۴۲

● **فصل ۲: اجزای PLC و عملکرد آنها** ۴۳

چکیده مطالب ۴۵

اصطلاحات و تعاریف ۴۶

۲-۱ مقدمه ۴۷

۲-۲ ماژول تغذیه ۴۹

۲-۲-۱ عملکرد منبع تغذیه ۴۹

۲-۲-۲ ویژگی‌های منبع تغذیه ۴۹

۲-۳ ماژول پردازشگر (CPU) ۵۱

۲-۳-۱ ساختار کلی CPU ۵۱

۲-۳-۲ حافظه CPU ۵۲

۲-۳-۳ عملکرد CPU ۵۴

۲-۴ ماژول ورودی دیجیتال ۵۶

۲-۴-۱ عملکرد ۵۶

۲-۴-۲ بخش‌های داخلی ماژول ۵۷

۲-۴-۳ ماژول دیجیتال با ورودی AC ۵۸

۲-۴-۴ ماژول دیجیتال با ورودی DC ۵۹

۲-۴-۵ ورودی‌های DC از نوع Sink /Source ۶۰
۲-۵ ماژول خروجی دیجیتال ۶۰
۲-۵-۱ عملکرد ۶۰
۲-۵-۲ بخش‌های داخلی ماژول ۶۱
۲-۵-۳ خروجی رله‌ای ۶۲
۲-۵-۴ خروجی‌های ترانزیستوری ۶۳
۲-۵-۵ خروجی Triac (خروجی AC) ۶۳
۲-۵-۶ خروجی Sink / Source ۶۴
۲-۶ ماژول ورودی آنالوگ ۶۴
۲-۷ ماژول خروجی آنالوگ ۶۸
۲-۸ پارامترهای کارت‌های ورودی و خروجی ۶۹
۲-۹ ماژول های مربوط به I/O های خاص ۷۱
۲-۱۰ ماژول ارتباطاتی Communication ۷۱
۲-۱۱ سایر اجزای PLC ۷۳
۲-۱۱-۱ ماژول‌های توسعه ۷۳
۲-۱۱-۲ وسیله‌برنامه‌ریزی (Programming Device) ۷۴
۲-۱۲ پرسش و تحقیق ۷۴
۲-۱۳ تست‌های خودآزمایی ۷۴

● **فصل ۳: اصول طراحی و انتخاب PLC و اجزای آن** ۷۷
چکیده مطالب ۷۹
اصطلاحات و تعاریف ۸۰
۳-۱ مقدمه ۸۲
۳-۲ شناخت فرآیند گام اول در فاز طراحی ۸۳
۳-۳ بررسی نقشه‌های P&ID ۸۴
۳-۴ استخراج لیست I/O ۸۶
۳-۵ انتخاب کارت‌های I/O ۸۷
۳-۶ انتخاب کارت‌های خاص ۸۸
۳-۷ بررسی چیدمان کارت‌های I/O و نیاز به Remote I/O ۸۸
۳-۸ انتخاب کارت‌های شبکه ۸۸
۳-۹ انتخاب CPU ۸۹
۳-۱۰ انتخاب رک ۹۱
۳-۱۱ انتخاب منبع تغذیه ۹۱

۳-۱۲ انتخاب پنلPLC ۹۳
۳-۱۳ کلید اضطراری ۹۶
۳-۱۴ انتخاب کابل با طول مناسب ۹۶
۳-۱۵ حفاظت در مقابل نویز ۹۸
۳-۱۶ پرسش و تحقیق ۹۹
۳-۱۷ تست‌های خودآزمایی ۹۹

● **فصل ۴: اصول نصب PLC** ۱۰۱
چکیده مطالب ۱۰۳
اصطلاحات و تعاریف ۱۰۴
۴-۱ مقدمه ۱۰۵
۴- ۲ نکات نصب پنل PLC ۱۰۵
۴-۳ نکات نصب منبع تغذیه ۱۰۶
۴-۴ نکات نصب اجزای PLC ۱۰۷
۴-۴-۱ نکات کلی نصب ۱۰۷
۴-۴-۲ نصب کارت‌های ورودی و خروجی ۱۰۹
۴-۴-۳ نکات مربوط به تغذیه I/Oها ۱۱۱
۴-۵ نصب تجهیزات حفاظتی ۱۱۳
۴-۶ سیم‌کشی ۱۱۵
۴-۷ زمین‌کردن ۱۱۷
۴-۸ نصب کلیدهای Emergency ۱۱۸
۴-۹ نصب MCR ۱۱۸
۴-۱۰ پرسش و تحقیق ۱۲۰
۴-۱۱ تست‌های خودآزمایی ۱۲۰

● **فصل ۵: آشنایی با وسایل ورودی و خروجی PLC** ۱۲۳
چکیده مطالب ۱۲۵
اصطلاحات و تعاریف ۱۲۶
۵-۱ مقدمه ۱۲۷
۵-۲ سیگنال‌های ورودی دیجیتال ۱۲۸
۵-۳ کلیدها و شستی‌های اپراتوری ۱۲۹
۵-۴ سنسورهای دیجیتال ۱۳۰
۵-۴-۱ لیمیت‌سوئیچ ۱۳۲

۵-۴-۲ سنسور القایی ۱۳۵
۵-۴-۳ سنسورهای خازنی ۱۴۱
۵-۴-۴ سنسورهای آلتراسونیک ۱۴۵
۵-۴-۵ سنسور فتوالکتریک ۱۴۷
۵-۴-۶ مقایسه ویژگی سنسورهای دیجیتال ۱۵۰
۵-۴-۷ سوئیچ‌های فرآیندی ۱۵۱
۵-۴-۷-۱ Pressure Switch ۱۵۱
۵-۴-۷-۲ Level Switch ۱۵۲
۵-۴-۷-۳ Temperature Switch ۱۵۳
۵-۴-۷-۴ Flow Switch ۱۵۴
۵-۵ سیگنال‌های خروجی دیجیتال ۱۵۴
۵-۵-۱ محرک‌های منطقی ۱۵۵
۵-۵-۲ سلونوئیدها ۱۵۵
۵-۵-۳ ولوها ۱۵۵
۵-۵-۴ کنتاکتور ۱۵۶
۵-۵-۵ سایر تجهیزات ۱۵۷
۵-۶ سیگنال‌های آنالوگ ورودی ۱۵۸
۵-۶-۱ سنسورهای اندازه‌گیری دما ۱۵۹
۵-۶-۲ سنسورهای اندازه‌گیری فشار ۱۶۲
۵-۶-۳ سنسورهای اندازه‌گیری ارتفاع سطح ۱۶۶
۵-۶-۴ سنسورهای اندازه‌گیری فلو ۱۶۸
۵-۷ سیگنال آنالوگ خروجی ۱۷۳
۵-۷-۱ ولوهای کنترلی ۱۷۳
۵-۷-۲ درایو کنترل‌دور موتورهای الکتریکی ۱۷۴
۵-۸ سیگنال‌های خاص ۱۷۵
۵-۹ پرسش و تحقیق ۱۷۵
۵-۱۰ تست‌های خودآزمایی ۱۷۵

● **فصل ۶: سیستم‌های کنترل زیمنس و جایگاه S7** ۱۷۹
چکیده مطالب ۱۸۱
اصطلاحات و تعاریف ۱۸۲
۶-۱ مقدمه ۱۸۵
۶-۲ سیستم‌های PLC ۱۸۶

۶-۲-۱ PLCهای خانواده S5 ..... ۱۸۷
۶-۲-۲ PLCهای خانواده S7 ..... ۱۸۹
۶-۲-۳ PLCهای خانواده TI ۱۹۲
۶-۲-۴ کنترل کننده‌های Quadlog ۱۹۳
۶-۲-۵ مینی PLCهای LOGO ۱۹۴
۶-۳ سیستم‌های PC based ۱۹۴
۶-۴ سیستم‌های کنترل فرآیند Process Control ۱۹۶
۶-۴-۱ Teleperm ۱۹۶
۶-۴-۲ PCS7 ..... ۱۹۶
۶-۴-۳ کنترل کننده‌های APACS ۱۹۷
۶-۵ سیستم‌های کنترل خاص ۱۹۸
۶-۵-۱ سیستم کنترل‌های TDC ۱۹۸
۶-۵-۲ سیستم کنترل SIMADYN D ۱۹۸
۶-۵-۳ سیستم‌های SINUMERIK ۱۹۹
۶-۵-۴ سیستم‌های SIMOTION ۱۹۹
۶-۶ پرسش و تحقیق ۲۰۰
۶-۷ تست‌های خودآزمایی ۲۰۰

● **فصل ۷: نرم‌افزارهای زیمنس و جایگاه Step 7** ۲۰۱
چکیده مطالب ۲۰۳
اصطلاحات و تعاریف ۲۰۴
۷-۱ مقدمه ۲۰۵
۷-۲ نرم‌افزارهای اصلی Programming ۲۰۵
۷-۲-۱ STEP 7 Basic ۲۰۵
۷-۲-۲ Step7 ..... ۲۰۶
۷-۲-۳ Step7 Proffessional ۲۰۷
۷-۲-۴ Step7 Lite ۲۰۷
۷-۲-۵ Step7 Microwin ۲۰۸
۷-۲-۶ STEP5 ..... ۲۰۹
۷-۲-۷ TISOFT ۲۰۹
۷-۲-۸ LOGO Soft Comfort ۲۰۹
۷-۲-۹ 4mation ۲۰۹
۷-۳ نرم افزارهای تکمیلی Programming ۲۱۰

۷-۳-۱ Graph 5/II ۲۱۰
۷-۳-۲ S7-Graph ۲۱۱
۷-۳-۳ S7-HiGRAPH ۲۱۲
۷-۳-۴ S7-SCL ۲۱۳
۷-۳-۵ CFC ۲۱۳
۷-۳-۶ SFC ۲۱۴
۷-۳-۷ SWR ۲۱۵
۷-۳-۸ H-System ۲۱۵
۷-۳-۹ F-System ۲۱۶
۷-۳-۱۰ Safety Matrix ۲۱۷
۷-۳-۱۱ M7 ProC/C++ ۲۱۷
۷-۳-۱۲ PLCSIM ۲۱۷
۷-۳-۱۳ Standard PID Control ۲۱۸
۷-۳-۱۴ Modular PID Control ۲۱۹
۷-۴ نرم‌افزارهای عیب‌یابی و سرویس ۲۱۹
۷-۴-۱ PDIAG ۲۱۹
۷-۴-۲ TeleService ۲۲۰
۷-۴-۳ PRODAVE ۲۲۰
۷- ۵ نرم‌افزارهای مانیتورینگ ۲۲۰
۷-۵-۱ winCC ۲۲۱
۷-۵-۲ WinCC Flexible ۲۲۲
۷-۵-۳ winAC ۲۲۲
۷-۵-۴ Protool ۲۲۳
۷-۵-۵ PC Access ۲۲۳
۷-۶ نرم‌افزارهای مربوط به شبکه‌های صنعتی ۲۲۳
۷-۶-۱ Simatic NET ۲۲۳
۷-۶-۲ iMAP ۲۲۴
۷-۶-۳ PDM ۲۲۴
۷-۶-۴ نرم‌افزارهای ارتباطی S5 ..... ۲۲۴
۷-۷ نرم‌افزارهای مربوط به Motion Control ۲۲۵
۷-۷-۱ S7-Technology ۲۲۵
۷-۷-۲ D7-Sys ۲۲۵
۷-۷-۳ Drive ES ۲۲۶

۷-۸ برخی دیگر از نرم‌افزارهای مرتبط با اتوماسیون ۲۲۶
۷-۸-۱ DOCPRO ۲۲۶
۷-۸-۲ HARDPRO ۲۲۶
۷-۸-۳ FuzzyControl++ ۲۲۶
۷-۸-۴ Neurosystem ۲۲۷
۷-۸-۵ Premium Studio ۲۲۸
۷-۸-۶ PCS7 ۲۲۸
۷-۹ پرسش و تحقیق ۲۲۸
۷-۱۰ تست‌های خودآزمایی ۲۲۹

● **فصل ۸: پیش‌نیازهای کار با نرم‌افزار Step 7** ۲۳۱
چکیده مطالب ۲۳۳
اصطلاحات و تعاریف ۲۳۴
۸-۱ مقدمه ۲۳۵
۸-۲ نیاز به Step7 در مراحل مختلف کار اتوماسیون ۲۳۵
۸-۳ نسخه‌های مختلف Step7 ۲۳۷
۸-۴ نیازمندی‌های نصب نرم‌افزار STEP 7 ۲۳۸
۸-۴-۱ سیستم عامل ۲۳۸
۸-۴-۲ مشخصات سخت‌افزاری ۲۳۹
۸-۵ نحوه نصب نرم‌افزار Step7 V5.5 ۲۴۰
۸-۶ مجوز نرم‌افزار Step 7 (Authorization) ۲۴۱
۸-۷ تنظیم ارتباط بین PLC و کامپیوتر ۲۴۳
۸-۸ انواع روش‌های ارتباط بین PLC و کامپیوتر ۲۴۴
۸-۸-۱ ارتباط بین کامپیوتر و پورت روی CPU ۲۴۵
۸-۸-۲ ارتباط بین کامپیوتر و کارت شبکه نصب شده روی PLC ۲۴۶
۸-۹ وسایل رابط بین PLC و کامپیوتر ۲۴۶
۸-۹-۱ PC Adapter ۲۴۶
۸-۹-۲ کارت PCMCIA برای لپ‌تاپ ۲۴۹
۸-۹-۳ کارت‌های قابل نصب روی مادربورد کامپیوتر ۲۴۹
۸-۱۰ استفاده از لپ‌تاپ خاص زیمنس PG ۲۵۱
۸-۱۱ پرسش و تحقیق ۲۵۲
۸-۱۲ تست‌های خودآزمایی ۲۵۲

● **فصل ۹: آشنایی با محیط نرم‌افزار Step 7** ۲۵۳
چکیده مطالب ۲۵۵
اصطلاحات و تعاریف ۲۵۶
۹-۱ مقدمه ۲۵۷
۹-۲ شروع کار با Simatic Manager ۲۵۷
۹-۲-۱ استفاده از Wizard جهت ایجاد پروژه ۲۵۸
۹-۲-۲ ایجاد پروژه به صورت دستی ۲۵۹
۹-۳ محیط پیکربندی سخت‌افزار ۲۶۳
۹-۴ محیط برنامه‌نویسی به زبان‌های LAD/FBD/STL ۲۶۴
۹-۵ Configure Network ۲۶۵
۹-۶ Simulator (PLC SIM) ۲۶۶
۹-۷ تنظیمات ارتباطی ۲۶۷
۹-۸ منوهای اصلی Simatic Manager ۲۶۸
۹-۸-۱ منوی File ۲۶۹
۹-۸-۲ منوی Edit ۲۷۱
۹-۸-۳ منوی Insert ۲۷۲
۹-۸-۴ منوی PLC ۲۷۳
۹-۸-۵ منوی View ۲۷۴
۹-۸-۶ منوی Options ۲۷۶
۹-۸-۷ منوی Window ۲۷۸
۹-۸-۸ منوی Help ۲۷۸
۹-۸-۹ نوار ابزار ۲۷۸
۹-۹ آشنایی با بخش Library و Sample ۲۷۹
۹-۱۰ فلوچارت کار با Step7 ..... ۲۸۰

● **فصل ۱۰: آشنایی با محیط پیکربندی سخت‌افزار** ۲۸۱
چکیده مطالب ۲۸۳
اصطلاحات و تعاریف ۲۸۴
۱۰-۱ مقدمه ۲۸۵
۱۰-۲ آشنایی با محیط HWconfig ۲۸۶
۱۰-۳ منوهای HWconfig ۲۸۷
۱۰-۳-۱ منوی Station ۲۸۷
۱۰-۳-۲ منوی Edit ۲۹۱

۱۰-۳-۳ منوی Insert ۲۹۲
۱۰-۳-۴ منوی PLC ۲۹۲
۱۰-۳-۵ منوی View ۲۹۵
۱۰-۳-۶ منوی Options ۲۹۵
۱۰-۴ آشنایی با کاتالوگ ۲۹۷
۱۰-۴-۱ نحوه وارد کردن اجزا ۲۹۷
۱۰-۴-۲ آشنایی با اجزای سخت افزاری کاتالوگ ۲۹۸
۱۰-۴-۳ پروفایل‌های کاتالوگ ۳۰۱
۱۰-۴-۴ Update نمودن کاتالوگ در H.Wconfig ۳۰۲
۱۰-۵ پرسش و تحقیق ۳۰۴

● **فصل ۱۱: نصب، پیکربندی و تنظیمات سخت‌افزار S7-300** ۳۰۵
چکیده مطالب ۳۰۷
اصطلاحات و تعاریف ۳۰۸
۱۱-۱ مقدمه ۳۰۹
۱۱-۲ Rack 300 ..... ۳۱۰
۱۱-۲-۱ عملکرد ۳۱۱
۱۱-۲-۲ ویژگی‌های رک S7-300 ..... ۳۱۲
۱۱-۲-۳ انواع رک ۳۰۰ ..... ۳۱۳
۱۱-۲-۴ نصب رک ۳۱۴
۱۱-۲-۵ پیکربندی رک در محیط HW Config ۳۱۵
۱۱-۳ Power Supply ۳۱۶
۱۱-۳-۱ عملکرد ۳۱۶
۱۱-۳-۲ انواع منبع تغذیه S7-300 ..... ۳۱۷
۱۱-۳-۳ ویژگی‌های منبع تغذیه ۳۱۷
۱۱-۳-۴ پیکربندی PS در محیط HWConfig ۳۱۸
۱۱-۴ CPU ۳۱۹
۱۱-۴-۱ عملکرد ۳۲۰
۱۱-۴-۲ انواع CPUهای 300 ..... ۳۲۰
۱۱-۴-۳ ویژگی‌های CPU ۳۲۳
۱۱-۴-۴ نصب و سیم‌کشی CPU ۳۲۷
۱۱-۴-۵ پیکربندی CPU در محیط HW Config ۳۲۹
۱۱-۵ کارت‌های ورودی دیجیتال ۳۳۲

۱۱-۵-۱ عملکرد کارت دیجیتال ورودی ۳۳۲
۱۱-۵-۲ انواع کارت‌های DI ۳۳۳
۱۱-۵-۳ مشخصات فنی کارت‌های DI ۳۳۳
۱۱-۵-۴ روش نصب کارت DI در رک ۳۳۵
۱۱-۵-۵ پیکربندی کارت DI در Hwconfig ۳۳۷
۱۱-۵-۶ قابلیت‌های ویژه در کارت‌های DI ۳۴۰
۱۱-۵-۷ کارت‌های DI خاص در S7-300 ..... ۳۴۱
۱۱-۶ کارت‌های خروجی دیجیتال DO ۳۴۳
۱۱-۶-۱ عملکرد ۳۴۳
۱۱-۶-۲ انواع کارت DO ۳۴۳
۱۱-۶-۳ مشخصات فنی کارت‌های DO ۳۴۵
۱۱-۶-۴ روش نصب و سیم‌کشی کارت DO ۳۴۶
۱۱-۶-۵ پیکربندی کارت DO در Hwconfig ۳۴۷
۱۱-۶-۶ قابلیت‌های ویژه در کارت‌های DO ۳۴۸
۱۱-۷ کارت‌های ورودی و خروجی دیجیتال (ترکیبی DI/DO) ۳۵۰
۱۱-۸ کارت‌های آنالوگ ورودی ۳۵۲
۱۱-۸-۱ عملکرد ۳۵۲
۱۱-۸-۲ انواع کارت‌های آنالوگ ورودی (AI) ۳۵۴
۱۱-۸-۳ مشخصات فنی کارت‌های آنالوگ ورودی ۳۵۷
۱۱-۸-۴ نصب کارت‌های آنالوگ ورودی ۳۵۸
۱۱-۸-۵ پیکربندی کارت AI در Hwconfig ۳۶۰
۱۱-۹ کارت‌های آنالوگ خروجی ۳۶۲
۱۱-۹-۱ عملکرد ۳۶۲
۱۱-۹-۲ انواع کارت‌های AO ۳۶۳
۱۱-۹-۳ مشخصات فنی کارت‌های آنالوگ خروجی ۳۶۴
۱۱-۹-۴ نصب کارت آنالوگ خروجی ۳۶۴
۱۱-۹-۵ پیکربندی کارت آنالوگ خروجی در Hwconfig ۳۶۶
۱۱-۱۰ کارت‌های ورودی و خروجی آنالوگ (ترکیبی AI/AO) ۳۶۷
۱۱-۱۱ کارت‌های خاص خانواده SM ۳۶۸
۱۱-۱۱-۱ Dummy Module (DM) ۳۶۸
۱۱-۱۱-۲ Simulator module (SM 374) ۳۷۰
۱۱-۱۱-۳ Position Decoder Module SM 338 ..... ۳۷۱
۱۱-۱۲ کارت‌های شبکه Communication Processor (CP) ۳۷۲

۱۱-۱۳ Function Module (FM) ۳۷۴
۱۱-۱۴ Interface Module (IM) ۳۷۶
۱۱-۱۴-۱ عملکرد ۳۷۶
۱۱-۱۴-۲ انواع IM در S7-300 ..... ۳۷۷
۱۱-۱۴-۳ روش‌های برقراری اتصال از طریق IM: ۳۸۰
۱۱-۱۴-۴ روش پیکربندی IM در hwconfig ۳۸۹
۱۱-۱۵ پیکربندی سیستم C7 ..... ۳۸۲
۱۱-۱۶ پرسش و تحقیق ۳۸۴
۱۱-۱۷ تست‌های خودآزمایی ۳۸۴

● **فصل ۱۲: مفاهیم پایه در عملکرد PLC** ۳۸۷
چکیده مطالب ۳۸۹
اصطلاحات و تعاریف ۳۹۰
۱۲-۱ مقدمه ۳۹۱
۱۲-۲ مدهای کاری PLC ۳۹۱
۱۲-۲-۱ مد Stop ۳۹۱
۱۲-۲-۲ مد Startup ۳۹۴
۱۲-۲-۳ مد RUN ۳۹۶
۱۲-۲-۴ جایگزینی RUN-P به‌جای RUN ۳۹۹
۱۲-۲-۵ مد HOLD ۴۰۰
۱۲-۲-۶ اولویت مدهای کاری CPU ۴۰۰
۱۲-۳ بخش‌های حافظه ۴۰۰
۱۲-۳-۱ Load Memory ۴۰۱
۱۲-۳-۲ Work Memory ۴۰۴
۱۲-۳-۳ System Memory ۴۰۵
۱۲-۴ راه‌اندازی مجدد CPU ۴۰۶
۱۲-۴-۱ Cold Restart ۴۰۶
۱۲-۴-۲ Warm Restart ۴۰۷
۱۲-۴-۳ Hot Restart ۴۰۷
۱۲-۵ ری‌ست کردن CPU ۴۰۸
۱۲-۵-۱ عملکرد Reset ۴۰۸
۱۲-۵-۲ Reset با سوئیچ MRES ۴۰۸
۱۲-۵-۳ Reset از طریق نرم‌افزار ۴۰۹

۱۲-۶ تاثیر ری‌ست و راه اندازی روی بخش‌های مختلف حافظه ۴۱۰
۱۲-۷ تاثیر عملکرد CPU روی LEDهای آن ۴۱۰
۱۲-۸ پرسش و تحقیق ۴۱۲
۱۲-۹ تست‌های خودآزمایی ۴۱۲

● فصل ۱۳: مفاهیم پایه برنامه‌نویسی PLC ۴۱۵
چکیده مطالب ۴۱۷
اصطلاحات و تعاریف ۴۱۸
۱۳-۱ مقدمه ۴۱۹
۱۳-۲ سیستم‌های عددی مورد استفاده در PLC ۴۱۹
۱۳-۲-۱ سیستم عددی دسیمال (Decimal System) ۴۱۹
۱۳-۲-۲ سیستم عددی باینری (Binary System) ۴۱۹
۱۳-۲-۳ سیستم عددی BCD ۴۲۱
۱۳-۲-۴ سیستم عددی Hex (Hexadecimal) ۴۲۲
۱۳-۳ انواع Data Type (نوع داده) ۴۲۵
۱۳-۳-۱ Elementary Data Types (داده‌های پایه) ۴۲۵
۱۳-۳-۲ Complex Data Types ۴۳۶
۱۳-۳-۳ Parameter Types ۴۴۰
۱۳-۴ نحوه آدرس‌دهی متغیرهای حافظه ۴۴۰
۱۳-۴-۱ نکات کلی آدرس‌دهی ۴۴۰
۱۳-۴-۲ آدرس‌دهی کارت‌های دیجیتال ۴۴۳
۱۳-۴-۳ آدرس‌دهی کارت‌های آنالوگ ۴۴۷
۱۳-۴-۴ آدرس‌دهی Peripheral ۴۴۹
۱۳-۴-۵ آدرس‌دهی متغیرهای حافظه ۴۵۰
۱۳-۴-۶ آدرس‌دهی کانتر و تایمر ۴۵۱
۱۳-۴-۷ آدرس دهی مطلق و سمبلیک ۴۵۱
۱۳-۵ نکته پایانی ۴۵۳
۱۳-۶ پرسش و تحقیق ۴۵۳
۱۳-۷ تست‌های خودآزمایی ۴۵۳

● فصل ۱۴: شروع برنامه‌نویسی با LAD/STL/FBD ۴۵۵
چکیده مطالب ۴۵۷
اصطلاحات و تعاریف ۴۵۸

۱۴-۱ مقدمه ۴۵۹
۱۴-۲ انواع روش‌های برنامه‌نویسی ۴۵۹
۱۴-۳ بلاک‌های برنامه‌نویسی ۴۶۱
۱۴-۳-۱ بلاک‌های OB ۴۶۱
۱۴-۳-۲ FC (فانکشن) ۴۶۴
۱۴-۳-۳ FB (فانکشن‌بلاک) ۴۶۴
۱۴-۳-۴ بلاک‌های سیستمی ۴۶۵
۱۴-۳-۵ بلاک‌های دیتا Data Block ۴۶۶
۱۴-۳-۶ UDT (User Defined Data Type) ۴۶۷
۱۴-۴ زبان‌های برنامه‌نویسی ۴۶۷
۱۴-۵ ایجاد بلاک‌ها در Simatic Manager ۴۶۹
۱۴-۶ محیط برنامه‌نویسی LAD /STL/FBD ۴۷۰
۱۴-۶-۱ آشنایی با بخش‌های مختلف LAD/STL/FBD ۴۷۰
۱۴-۶-۲ منوها و نوارابزار ۴۷۵
۱۴-۷ ایجاد سمبل‌ها ۴۸۰
۱۴-۸ تست برنامه با سیمولاتور ۴۸۲
۱۴-۹ گروه دستورات برنامه‌نویسی LAD و FBD ۴۸۳
۱۴-۱۰ پرسش و تحقیق ۴۸۴
۱۴-۱۱ تست‌های خودآزمایی ۴۸۴

● **فصل ۱۵: دستورات برنامه‌نویسی Bit Logic** ۴۸۷
چکیده مطالب ۴۸۹
۱۵-۱ مقدمه ۴۹۰
۱۵-۲ دستورات Bit Logic در زبان LAD ۴۹۲
۱۵-۳ دستورات Bit Logic در زبان FBD ۵۲۵
۱۵-۴ تمرین ۵۳۳

● **فصل ۱۶: برنامه‌نویسی با دستورات تایمر** ۵۳۵
چکیده مطالب ۵۳۷
۱۶-۱ مقدمه ۵۳۸
۱۶-۲ ورودی و خروجی‌های مشترک تایمرها ۵۳۹
۱۶-۳ انواع تایمرها و عملکرد آنها ۵۴۰
۱۶-۴ دستورات بیتی تایمر ۵۶۰

۱۶-۵ بررسی ماندگاری تایمرها ۵۶۱
۱۶-۶ بررسی خروجی های BCD و BI تایمر ۵۶۲
۱۶-۷ تمرین ۵۶۴

● **فصل ۱۷: دستورات برنامه‌نویسی کانترها** ۵۶۷
چکیده مطالب ۵۶۹
۱۷-۱ مقدمه ۵۷۰
۱۷-۲ انواع کانترها و ورودی و خروجی‌های آن‌ها ۵۷۱
۱۷-۳ اصول عملکرد کانتر ۵۷۲
۱۷-۴ بررسی خروجی‌های CV و CV_BCD کانتر ۵۷۷
۱۷-۵ دستورات بیتی کانتر ۵۷۸
۱۷-۶ تمرین ۵۷۹

● **فصل ۱۸: دستورات برنامه‌نویسی مقایسه‌گرها** ۵۸۱
چکیده مطالب ۵۸۳
۱۸-۱ مقدمه ۵۸۴
۱۸-۲ انواع مقایسه‌گر ۵۸۴
۱۸-۳ عملکرد مقایسه‌گرها ۵۸۵
۱۸-۴ تمرین ۵۹۴

● **فصل ۱۹: دستورات محاسباتی** ۵۹۷
چکیده مطالب ۵۹۹
۱۹-۱ مقدمه ۶۰۰
۱۹-۲ دستورات محاسباتی Integer ۶۰۱
۱۹-۲-۱ دستورات محاسباتی اعداد صحیح ۱۶ بیتی ۶۰۱
۱۹-۲-۲ دستورات محاسباتی اعداد صحیح ۳۲ بیتی ۶۰۳
۱۹-۳ دستورات محاسباتی Real ۶۰۴
۱۹-۴ تمرین ۶۱۲

● **فصل ۲۰: دستورات تبدیل** ۶۱۳
چکیده مطالب ۶۱۵
۲۰-۱ مقدمه ۶۱۶

۲۰-۲ مبدل‌های فرمت اعداد به یکدیگر ۶۱۷
۲۰-۲-۱ تبدیل BCD به Integer و بالعکس ۶۱۷
۲۰-۲-۲ تبدیل BCD به Double Integer و بالعکس ۶۲۲
۲۰-۲-۳ تبدیل Integer به Double Integer ۶۲۳
۲۰-۲-۴ تبدیل Integer به Real ۶۲۴
۲۰-۳ مبدل‌های متمم یک و دو (قرینه و معکوس‌سازی) ۶۲۵
۲۰-۳-۱ دستور متمم یک (معکوس بیت‌های عدد) ۶۲۵
۲۰-۳-۲ دستور متمم‌دو (قرینه‌ی عدد) ۶۲۶
۲۰-۴ مبدل‌های Roundکننده اعداد اعشاری ۶۲۷
۲۰-۵ تمرین ۶۲۸

● **فصل ۲۱: دستور انتقال Move** ۶۲۹
چکیده مطالب ۶۳۱
۲۱-۱ مقدمه ۶۳۲
۲۱-۲ عملکرد دستور MOVE ۶۳۲
۲۱-۳ نکات کار با دستور MOVE ۶۳۳
۲۱-۴ تمرین ۶۴۰

● **فصل ۲۲: دستورات شیفت و چرخش** ۶۴۱
چکیده مطالب ۶۴۳
۲۲-۱ مقدمه ۶۴۴
۲۲-۲ دستورات شیفت ۶۴۵
۲۲- ۳ دستورات چرخش ۶۴۹
۲۲-۴ تمرین ۶۵۱

● **فصل ۲۳: سایر دستورات برنامه‌نویسی LAD/FBD** ۶۵۳
چکیده مطالب ۶۵۵
۲۳-۱ مقدمه ۶۵۶
۲۳-۲ دستورات Word Logic ۶۵۶
۲۳-۳ دستورات Jump ۶۵۹
۲۳-۴ دستورات Program Control ۶۶۲
۲۳-۵ دستور باز کردن DB ۶۶۵
۲۳-۶ دستورات Status Bits ۶۶۶
۲۳-۷ تمرین ۶۶۶

● **فصل ۲۴: استفاده از Monitor/Modify و Force** ۶۶۷
چکیده مطالب ۶۶۹
۲۴-۱ مقدمه ۶۷۰
۲۴-۲ Monitor / Modify ۶۷۰
۲۴-۲-۱ ابزارهای Monitor/Modify ۶۷۰
۲۴-۲-۲ Monitor/Modify با استفاده از جدول VAT ۶۷۲
۲۴-۲-۳ نکات کار با Monitor/Modify ۶۷۷
۲۴-۳ انجام Force ۶۷۸
۲۴-۳-۱ نحوه دسترسی به پنجره Force ۶۷۹
۲۴-۳-۲ تفاوت Force و Modify ۶۷۹
۲۴-۳-۳ Force با استفاده از جدول VAT ۶۸۰
۲۴-۴ پرسش و تحقیق ۶۸۳

● **فصل ۲۵: ارتباط Online با PLC** ۶۸۵
چکیده مطالب ۶۸۷
۲۵-۱ مقدمه ۶۸۸
۲۵-۲ Download / Upload ۶۸۸
۲۵-۲-۱ Download و Upload در محیط Simatic Manager ۶۹۰
۲۵-۲-۲ Download و Upload در محیط HW Config ۶۹۲
۲۵-۲-۳ Download و Upload در محیط LAD/STL/FBD ۶۹۳
۲۵-۲-۴ Download و Upload در محیط NetPro ۶۹۳
۲۵-۳ Simatic Manager در نمایش Online ۶۹۴
۲۵-۴ محیط پیکربندی سخت‌افزار در نمایش Online ۶۹۶
۲۵-۵ محیط برنامه‌نویسی در نمایش Online ۷۰۱
۲۵-۶ پرسش و تحقیق ۷۰۳
پیوست ۷۰۷
منابع و مراجع ۷۱۳

فصل ۱

نگاهی به سیستم‌های کنترل و جایگاه PLC

۱-۱ مقدمه
۱-۲ نگاهی به تاریخچه سیستم‌های کنترل
۱-۳ مزایا و معایب سیستم‌های کنترل مدرن
۱-۴ مقایسه PLC با سایر سیستم‌های کنترل و مزایا و معایب آن
۱-۴-۱ مقایسه PLC با سیستم‌های رله کنتاکتی
۱-۴-۲ مقایسه PLC با سیستم‌های DDC
۱-۴-۳ مقایسه PLC با سیستم‌های DCS
۱-۵ معرفی سازندگان PLC
۱-۶ پرسش و تحقیق
۱-۷ تست‌های خودآزمابی

در این فصل با سیر تحول سیستم‌های کنترل آشنا خواهید شد. مزایا و معایب PLC نسبت به سایر سیستم‌ها و معرفی سازندگان PLC از دیگر مطالب این بخش هستند. این بخش صرفاً جهت ارتقاء اطلاعات عمومی خواننده است و مطالب آن پیش‌نیاز بخش‌های بعدی محسوب نمی‌شود.

چکیده مطالب

- اولین کاربرد صنعتی سیستم‌های کنترل در قرن ۱۷ میلادی بود که به‌عنوان سیستم کنترل مکانیکی شناخته می‌شود.
- در قرن ۱۸ اولین پایه‌های تئوری کنترل پی‌ریزی شد و در اوایل قرن ۱۹ رشد چشمگیری یافت.
- در دهه ۱۹۵۰ سیستم‌های کنترل الکترونیکی عرضه شدند.
- در دهه ۱۹۶۰ سیستم‌های کنترل مبتنی بر کامپیوتر موسوم به DDC ارائه گردید.
- در ۱۹۷۰ اولین PLCها برای جایگزینی مدارات فرمان رله کنتاکتوری پا به عرضه وجود گذاشتند .
- در ۱۹۷۶ اولین سیستم های DCS برای جایگزینی سیستم های DDC عرضه شدند.
- شرکت‌های Modicon و Allen Bradley و General Electric و Siemens و ABB از سازندگان پیشگام در زمینه PLC هستند.
- در دهه ۱۹۸۰ استفاده از Remote I/O برای اتصال ورودی و خروجی‌ها از طریق شبکه به PLC رونق گرفت.
- در دهه ۱۹۹۰ تکنولوژی فیلدباس برای استفاده در سیستم‌های کنترل ارائه گردید.
- زبان‌های برنامه‌نویسی PLC در ۱۹۹۳ توسط IEC1131 استاندارد گردید.
- استانداردسازی ارتباط بین کنترلرها و سیستم‌های مانیتورینگ در ۱۹۹۶ با ارائه OPC صورت گرفت.
- در ۱۹۹۶ سیستم‌های PC Based برای استفاده به‌عنوان کنترلر مطرح گردید که نسبت به PLC در برخی جنبه‌ها قوت و در برخی جنبه‌ها ضعف دارد.

اصطلاحات و تعاریف

فرآیند Process

به مجموعه‌ای که برای تولید محصول یا محصولات مشخصی طراحی شده است فرآیند می‌گویند. در یک فرآیند، دستگاه‌ها و ماشین‌آلات مختلفی نصب می‌شوند که هر کدام نقش خاصی را در تولید محصول نهایی بازی می‌کنند.

دیجیتال Digital

به کمیتی که فقط دارای دو وضعیت است دیجیتال گفته می‌شود. سیگنال دیجیتال سیگنالی است که به‌صورت 0 یا 1 (قطع یا وصل) ظاهر می‌شود و وضعیت دیگری ندارد.

آنالوگ Analoge

به کمیتی که مقدار آن می‌تواند در بازه خاصی تغییر کند آنالوگ گفته می‌شود، آنالوگ بر خلاف دیجیتال فقط دو وضعیت ندارد. سیگنال آنالوگ سیگنالی است که دارای مقادیر مختلف الکتریکی است بعنوان مثال یک ولتاژ متغیر یا یک جریان متغیر را می‌توان یک سیگنال آنالوگ نامید.

Continuous Process

به فرآیندهایی که در آن بیشتر سیگنال‌ها به‌صورت آنالوگ هستند گفته می‌شود.

Discrete Process

به فرآیندهایی که در آن بیشتر سیگنال‌ها به‌صورت دیجیتال هستند گفته می‌شود.

ترانسمیتر Transmitter

ترانسمیتر وسیله‌ای است که یک پارامترهای متغیر فیزیکی (مثل دما، فشار، فلو و...) یا پارامتر متغیر شیمیایی(مثل...H2S,CO2,) یا متغیر الکتریکی (مثل جریان و ولتاژ و ...) را پس از اندازه‌گیری به سیگنال جریانی یا ولتاژی استاندارد تبدیل می‌نماید. به‌عنوان مثال یک ترانسمیتر فشار که روی یک مخزن نصب شده است تغییرات فشار مخزن را که از ترانسدیوسر دریافت می‌کند به سیگنال 4-20 mA تبدیل می‌نماید. البته برخی از ترانسمیترها سیگنال الکتریکی غیر استاندارد را به سیگنال الکتریکی استاندارد تبدیل می‌نمایند، مثلاً ترانسمیتر دما می‌تواند سیگنال خروجی ترموکوپل یا RTD را به سیگنال جریانی تبدیل نماید.

سیگنال ترانسمیترها یکی از مهمترین ورودی‌های آنالوگ سیستم کنترل محسوب می‌گردد.

ترانسمیتر هوشمند Smart Transmitter

ترانسمیتری است که علاوه بر تبدیل مقدار متغیر اندازه‌گیری شده، قادر است آن را شخصاً پردازش کند یا آنرا به کدهای مناسب جهت انتقال به شبکه‌های صنعتی تبدیل نماید. این ترانسمیترها قابلیت اتصال به‌وسیله کالیبراتور Hand Held Communicator را دارا هستند.

سنسور Sensor

سنسور وسیله تبدیل کمیت فیزیکی یا کمیت شیمیایی به کمیت الکتریکی است و مفهومی گسترده‌تر از ترانسمیتر دارد زیرا کمیت‌های دیجیتال را نیز دربر می‌گیرد. یک سوئیچ فشار و یک ترانسمیتر فشار هر دو از خانواده سنسور محسوب می‌شوند. سنسورها به‌عنوان چشم و گوش و حواس یک فرآیند تلقی می‌شوند.

محرک Actuator

Actuator که به فارسی محرک یا عملگر نیز ترجمه شده است وسیله‌ای است که فرمان سیستم کنترل به آن ارسال شده و در آن فرمان کنترلی به یک حرکت مکانیکی تبدیل می‌شود. در برخی محرک‌ها فرمان الکتریکی ابتدا به نیوماتیک یا هیدرولیک تبدیل شده سپس حرکت مکانیکی را ایجاد می‌کنند. ولو کنترلی یک محرک است و سیستم کنترل می‌تواند به آن فرمان باز و بسته‌شدن بدهد.

فیدبک Feedback

منظور از فیدبک که به‌معنی بازخورد ترجمه شده است، نتیجه واکنش سیستم نسبت به یک فرمان خاص است . فیدبک به کنترلر برگشت داده می‌شود تا بتواند فرمان مناسب جدید را تولید کند. به‌عنوان مثال در یک کوره می‌توان با سنسور دما، دمای کوره را اندازه‌گیری کرد و به سیستم کنترل ارسال نمود. سیستم کنترل پس از اعمال فرمان افزایش دمای کوره نتیجه را از برگشت سیگنال سنسور دما بررسی می‌نماید.

لوپ کنترلی

لوپ یا حلقه کنترلی شامل فرآیند، سیستم کنترل، محرک و فیدبک است. به‌عنوان مثال در یک کوره‌ی گازسوز با کنترل شیر ورودی گاز دما تغییر می‌کند. این دما توسط سنسور اندازه‌گیری و به سیستم کنترل داده می شود تا فرمان را اصلاح نماید. این مجموعه یک حلقه را تشکیل می‌دهد که به آن لوپ کنترلی گفته می‌شود.

لوپ PID

منظور همان لوپ کنترلی است که در آن از یک روش کنترل به نام PID استفاده می‌شود. برای پاسخ خوب و مناسب آن لازم است سه پارامتر مختلف P(بهره)، I (انتگرال‌گیر)و D (مشتق‌گیر)در سیستم کنترل تنظیم گردد.

فیلد Field

در یک فرآیند به ناحیه‌ای که در آن سنسورها و محرک‌ها نصب می‌شوند ناحیه فیلد گفته می‌شود.

I/O

منظور Input / Output است که به اختصار I/O خوانده می‌شود. مفهوم آن از دیدگاه سیستم کنترل قابل توصیف است. آنچه از فیلد به سیستم کنترل وارد می‌شود به‌عنوان ورودی است و آنچه با فرمان ارسالی از کنترلر کنترل می‌گردد به‌عنوان خروجی محسوب می‌گردد.

شبکه صنعتی Industrial Network

به ارتباط سریال که برای ارتباط بین دو یا چند وسیله به‌کار می‌رود و در آن از پروتکل خاص صنعتی استفاده می‌گردد، شبکه صنعتی گفته می‌شود.

پروتکل Protocol

پروتکل از نظر لغوی به‌معنای قرار داد است. وقتی از پروتکل‌های شبکه صحبت می‌شود منظور قراردادی است که در طراحی و معماری یک شبکه از آن استفاده شده است. به‌عنوان مثال تعریف صفر و یک که به‌صورت ولتاژی باشد یا جریانی و سطح حداقل و حداکثر ولتاژ یا جریان یا نحوه خطایابی از مواردی است که قرارداد می‌شود و در پروتکل شبکه از آن استفاده می‌گردد.

مانیتورینگ HMI

HMI مخفف Human Machine Interfaceاست که به رابط انسان و ماشین ترجمه شده است. منظور سیستم کامپیوتری است که اپراتور از آن برای مشاهده وضعیت فرآیند یا اعمال برخی فرامین استفاده می‌کند. از اینرو اصطلاح سیستم مانیتورینگ نیز برای آن به‌کار می‌رود.

اتوماسیون Automation

اتوماسیون به‌معنای کنترل و هدایت یک دستگاه به‌صورت خودکار است، بدین معنی که ابزارهای کنترلی مثل کامپیوتر جایگزین قدرت تفکر و نیروی انسانی می‌شوند. مسئله اتوماسیون زمانی مطرح می‌شود که نیازهایی مانند انجام کاری به‌صورت مکرر، نیاز به نظارتی مستمر و دقیق، انجام فعالیتی خطرآفرین و یا نیاز به کارهایی با دقت یا سرعت فوق‌العاده زیاد وجود دارد.

۱-۱ مقدمه

از تولد اولین کنترل‌کننده برنامه‌پذیر که اختصاراً PLC خوانده شد زمان زیادی نمی‌گذرد. PLC که مخفف Programmable Logic Controller است، افق جدیدی را به روی سیستم‌های کنترل گشود و آنرا دستخوش تحولات زیادی کرد ولی این تحول صرفاً به‌واسطه‌ی PLC نبود. بسیاری از پیشرفت‌هایی که در زمینه‌های دیگر علمی صورت گرفت به PLC رنگ و جلوه دیگری بخشید و عرصه را برای سایر سیستم‌های کنترل مدرن باز کرد. از جمله آنها می‌توان به شبکه‌های صنعتی، ترانسمیترهای هوشمند، تکنولوژی فیلدباس، نرم‌افزارها و سخت‌افزارهای جدید کامپیوتری اشاره کرد.

در این فصل با نگاهی گذرا به سیستم‌های کنترلی قدیمی به بررسی سیستم‌هایی که از دهه ۶۰ میلادی به بعد عرضه شدند پرداخته‌شده است و ضمن تشریح سیستم‌های کنترلی مختلف سعی شده است جایگاه PLC در میان آنها روشن گردد. نهایتاً ضمن مقایسه PLC با سایر سیستم‌های کنترل، سازندگان مطرح PLC معرفی گردیده‌اند. مطالب این فصل صرفاً جهت بالا بردن دانش عمومی خواننده است و پیش‌نیاز مطالب فصل‌های بعد محسوب نمی‌شود.

۱-۲ نگاهی به تاریخچه سیستم‌های کنترل

نیاز بشر به سیستم‌های کنترل

جمله معروفی هست که می‌گوید: *احتیاج مادر اختراع است.* از دیر باز بشر همواره برای تامین نیازهای خود به دنبال ساخت ابزار جدید بوده است و حتی در شرایطی که از هر نظر امکانات زندگی خود را در حد مطلوب فراهم کرده است، باز عطش سیری‌ناپذیر او برای دستیابی به رفاه ، ثروت و امکانات بیشتر ساخت وسایل و ابزارآلات جدید را رقم زده است.

این احساس نیاز منجر به سیر تکاملی ساخت وسایل جدید گردید و نهایتاً انقلاب صنعتی را رقم زد. در این ایام تصور انسان بر این بود که هر چه بیشتر از منابع و مواد خام و نیروی کار استفاده کند امکانات وسیع‌تری را برای خود فراهم می‌سازد اما به‌تدریج و به‌ویژه در قرون اخیر به این نتیجه رسید که تکیه بر منابع و نیروی کار کافی نیست و لازم است توانمندی‌های ذهنی خود را نیز افزایش دهد. تلاش در جهت افزایش دانش و توانمندی فکر، منجر به نتایجی شگفت‌آور شد. وسایلی ساخته شد که از نظر مواد و منابع بسیار کوچک و کم حجم بودند ولی توانایی‌های وسیعی را با خود به همراه آوردند، تراشه‌های سیلیکونی از این دسته بودند و انقلاب میکروپروسسوری را رقم زدند. ارزش این‌گونه وسایل به‌خاطر دانشی بود که برای ساخت آنها به‌کار رفت نه به‌خاطر مواد اندکی که در آنها استفاده شد. از این مرحله انسان به مفهوم واقعی دانش‌گرا شد و روند حرکت به سوی دانش برای تولید ثروت و رفاه بیشتر سرعت گرفت.

سیستم‌های کنترل از جمله مثال‌های بارز در این زمینه هستند. سیستم‌های مبتنی بر میکروپروسسور بستری را فراهم کرد که انسان بدون نیاز به منابع بیشتر بتواند دانش و تجربه خود را به این سیستم‌ها وارد نماید و سیستم‌های خبره یا هوشمند را تولید کند. این سیستم‌ها می‌توانند بدون دخالت انسان تصمیم‌گیری کنند و فرآیند را به سمت مطلوب و بهینه هدایت نمایند.

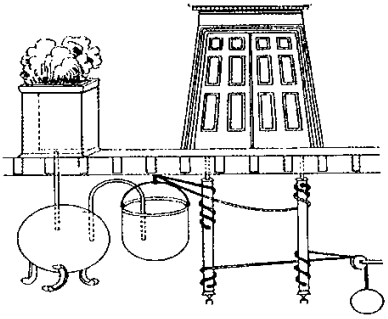
دلایلی که استفاده‌کننده را به‌سمت کاربری سیستم‌های پیشرفته کنترلی ترغیب می‌کند زیاد است، کاربران صنعتی دریافتند که با استفاده از این سیستم‌ها کار عیب‌یابی آنها در زمان کمتری انجام می‌شود و زمان‌های توقف فرآیند به حداقل کاهش می‌یابد. صاحبان صنایع دریافتند که با استفاده از سیستم‌های کنترلی می‌توانند محصولات خود را از نظر کیفیت و

کمیت بهبود ببخشند و هزینه‌های خود را نیز کاهش دهند. مدیران دریافتند که با استفاده از این سیستم‌ها به‌سهولت می‌توانند کل فرآیند را زیر نظر بگیرند و گزارش‌های دقیق‌تری دریافت کنند، مدل‌های تولید را بهبود ببخشند و طرح‌های توسعه را به‌سهولت اجرا کنند.

این موارد و مزایای بسیاردیگر منجر به توسعه روز افزون سیستم‌های کنترل گردید که هنوز رو به بهبود و تکامل است.

قدیمی‌ترین سیستم‌های کنترل

سیستم‌های کنترل اتوماتیک صرفاً سیستم‌های مدرنی که امروزه در اطراف ما به وفور یافت می‌شوند نیستند، سیستم‌های کنترل اتوماتیک از دیرباز وجود داشته‌اند. اینکه اولین سیستم چه سیستمی بوده و توسط چه کسی ابداع شده است به درستی مشخص نیست، ولی یافته‌ها نشان می‌دهد که بشر حتی سال‌ها قبل از میلاد مسیح نمونه‌هایی از کنترل خودکار را پیاده‌سازی نموده است. به‌نظر می‌رسد یکی از اولین سیستم‌های خودکار در کنترل باز و بسته کردن در معبدها به‌کار می‌رفته و مردم عادی را شگفت‌زده می‌نموده است. سیستم به‌گونه‌ای بوده که وقتی آتش در معبد روشن می‌شده درهای معبد باز و با خاموش کردن آن درهای معبد بسته می‌شده است.

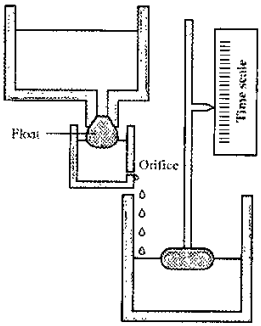


شکل ۱-۱ سیستم باز وبسته شدن خودکار در معبد– قبل از میلاد

با روشن شدن آتش، هوا در لوله‌ی بسته‌ی زیر آن گرم شده و انبساط هوا آب را از مخزن به داخل سطل می‌راند و وزن آن را زیاد می‌کند تا اینکه بر وزنه تعادلی غلبه کرده و در را باز می‌کند.

وقتی آتش خاموش شود، هوا در لوله به‌تدریج سرد شده و انقباض آن آب را از سطل به مخزن مکش می‌کند، بدین ترتیب وزنه بر وزن سطل غلبه کرده و در را می‌بندد.

نمونه دیگری که ازحدود چند دهه قبل از میلاد سراغ داریم ساعت آبی است که با تغییر سطح آب نمایشگر زمان جابه‌جا می‌شده است.



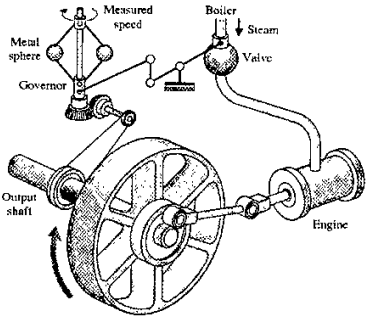
شکل ۱-۲ ساعت آبی

اختراعاتی از این دست از زمان‌های قدیم کم و بیش عرضه شده‌اند. با این حال اولین سیستم‌های حلقه بسته مجهز به فیدبک در قرن ۱۵ میلادی عرضه شدند که از معروف‌ترین آنها می‌توان به سیستم کنترل فشار دیگ بخار که توسط دنیس پاپن ابداع گردید اشاره نمود.

سیستم‌های کنترل مکانیکی

اولین کاربرد صنعتی سیستم‌های کنترل در قرن ۱۷ میلادی توسط جیمز وات ابداع گردید. این سیستم که به گاورنر توپ گردان[1] موسوم است در واقع یک کنترل‌کننده سرعت به‌شمار می‌رود.

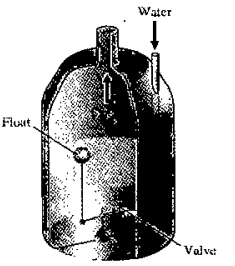
همانطور که در شکل ۱-۳ دیده می‌شود، چرخش توپ‌های گردان منجر به تغییر وضعیت شیر بخار می‌گردد. هر چه سرعت چرخش توپ بیشتر شود، بخار کمتری وارد سیستم می‌شود و سرعت سیستم کنترل می‌گردد.



شکل ۱-۳ گاورنر توپ گردان–کنترل سرعت ماشین بخار که توسط جیمز وات اختراع شد.

1. Flyball governor

سیستم تنظیم‌کننده سطح نیز در قرن ۱۷ عرضه شد. شناور سطح مخزن با فرمان به ولو ورودی، میزان سطح را کنترل می‌نماید. در شکل ۱-۴ ساختمان این وسیله نشان داده شده است.



شکل ۱-۴ ساختمان دستگاه تنظیم‌کننده‌ی سطح

با وقوع انقلاب صنعتی در اروپا، کوره‌ها، بویلرها، موتورهای بخار پیشرفته و رگولاتورهای شناور طراحی شد که کنترل آنها توسط سیستم‌های ساده امکان‌پذیر نبود. این سیستم‌های کنترل همگی به‌صورت مکانیکی بودند.

قرن ۱۸ با تلاش برای بهبود سیستم‌های کنترلی که تا آن زمان ساخته شده بود آغاز شد. بهبود پایداری و کاهش نوسانات سیستم، مسئله اصلی دانشمندان بود. در قرن ۱۸ اولین پایه‌های تئوری کنترل پی‌ریزی شد، تلاش‌های بسیاری در جهت یافتن قواعد سیستم، مدل‌کردن آنها به زبان ریاضی و یافتن راه‌کارهای بهبود آنها صورت گرفت. ماکسول از جمله دانشمندان آن دوره است که توانست یک چارچوب نظری برای سیستم کنترل سرعت جیمز وات ارائه نموده و آنرا با عبارات ریاضی به‌خوبی بیان نماید.

تئوری سیستم‌های کنترل روزبه‌روز توسعه یافت و در اوایل قرن ۱۹ رشد چشمگیری نمود. دانشمندانی همچون نایکوئیست از افراد شاخص این دوره به‌شمار می‌روند. در همین ایام بود که برای اولین بار لغت سرومکانیزم که معرف سیستم‌های کنترل مجهز به فیدبک بود در ادبیات کنترل به‌کار گرفته شد.

جنگ جهانی دوم علیرغم خسارت‌های جانی و مالی زیادی که به بشریت تحمیل کرد، کمک شایانی به پیشرفت سیستم‌های کنترلی نمود. هزینه‌های زیادی با اهداف نظامی صرف این سیستم‌ها گردید تا ادوات جنگی عملکرد بهتری از نظر سرعت و دقت داشته باشند. این امر عملاً منجر به شکوفایی مهندسی کنترل گردید.

در فاصله زمانی سال‌های ۱۹۴۰ تا ۱۹۴۵، یافته‌ها و تحقیقات انجام شده در زمینه کنترل محرمانه تلقی می‌شد. پس از جنگ محدودیت‌ها برداشته شد و حجم وسیعی از اطلاعات در اختیار دانشمندان قرار گرفت، اطلاعاتی که پایه و اساس پیشرفت‌های بعدی بود.

در دهه ۵۰ میلادی اولین کامپیوترهای بزرگ ساخته شدند، این کامپیوترهای بزرگ که به Main Frame *معروف شدند* اگرچه کند بودند ولی با استفاده از آنها تحلیل الگوریتم‌های پیچیده و بهینه‌سازی سیستم‌های کنترل امکان‌پذیر گردید. تا این زمان هنوز کامپیوتر به‌عنوان یک ابزار کنترل‌کننده به‌کار گرفته نشده بود و از آن فقط برای تحلیل سیستم‌ها استفاده می‌شد.

سیستم‌های کنترل نیوماتیکی

قبل از به‌کارگیری کامپیوتر در کار کنترل، سیستم‌های کنترلی دیگری نیز استفاده می‌شد که سیستم کنترل نیوماتیکی از این جمله است. این سیستم‌ها با تغذیه فشار هوای ابزاردقیق کار می‌کردند و مجهز به کنترل‌کننده‌های PID [1] نیوماتیکی بودند. در این کنترل‌کننده‌ها حتی اعمال محاسباتی ساده مانند جمع، ضرب، تقسیم و تفریق نیز به‌صورت نیوماتیکی پیاده‌سازی می‌شد. عملکرد لوپ‌های کنترلی که به‌صورت نیوماتیکی کنترل می شد به‌صورت منفرد مورد رضایت بود.

سیستم‌های کنترل الکترونیکی

بعد از اختراع ترانزیستور در سال ۱۹۴۷ میلادی، صنعت الکترونیک با سرعت زیادی رشد نمود. با توسعه تکنولوژی ساخت مدارات الکترونیکی، سیستم‌های کنترلی الکترونیکی عرضه شد. مدارات آنالوگ زیاد با استفاده از عناصر الکترونیکی ساخته شد و به‌جای سیستم‌های کنترل مکانیکی و نیوماتیکی به‌کار گرفته شد. این مدارات الکترونیکی به‌صورت مستقل طراحی و برای هدف مورد نظر به یکدیگر متصل می‌شدند و برای کنترل تک حلقه‌ای [2] به‌کار گرفته شد. این سیستم‌ها سرعت و دقت بیشتری نسبت به سیستم‌های نیوماتیکی داشتند و از نظر اقتصادی نیز ارزانتر بودند. پیشرفت روز افزون در ساخت قطعات الکترونیک و تولد میکروپروسسورها در فاصله زمانی کم، این سیستم‌ها را نیز منسوخ نمود.

سیستم‌های کنترل میکروپروسسوری

با ساخت مدارات مجتمع IC، کنترل‌کننده‌های پیشرفته و قابل برنامه‌ریزی و با هزینه‌ای کمتر عرضه شد. در دهه ۷۰ میکروپروسسورها وارد دنیای کنترل شدند و کار کنترل لوپ‌ها را به‌دست گرفتند.

این پیشرفت‌ها سیستم‌های کنترل را دچار تحول اساسی کرد. میکروپروسسور جامه جدیدی به سیستم‌های کنترل پوشاند و سیستم‌های کنترل مکانیکی و نیوماتیکی و الکترونیکی قدیمی را کنار زد.

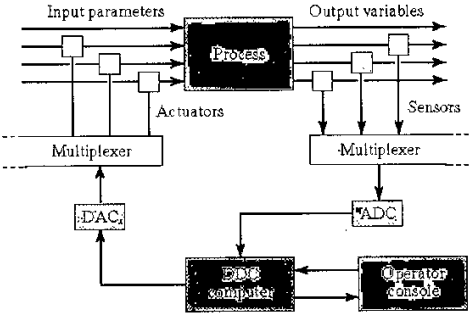
لازم است اشاره شود که قبل از این اتفاقات، در دهه ۱۹۶۰ استاندارد سیگنال‌های آنالوگ 4 الی 20mA ارائه شد و وسایل ابزاردقیق یعنی ترانسمیترها و کنترل‌ولوها بر اساس این استاندارد ساخته و عرضه شدند. این اتفاقات زمینه‌ساز تحولات بعدی سیستم‌های کنترل در سال‌های بعد گردید.

سیستم‌های کنترل DDC (اولین کنترل‌کننده‌های کامپیوتری)

با کاربرد میکروپروسسور به‌عنوان کنترلر، زمینه برای استفاده از کامپیوتر به‌عنوان قسمت اصلی سیستم کنترل فراهم شد. اولین سیستم‌های کنترل کامپیوتری در ۱۹۶۲ با نام DDC [3] عرضه شد که برای مانیتور کردن و کنترل فرآیند از آنها استفاده گردید. کامپیوتر می‌توانست سیگنال‌های استاندارد ترانسمیترها را دریافت نموده و آنها را نمایش دهد، بعلاوه می‌توانست براساس فانکشن برنامه‌نویسی شده‌ی PID، فرامینی را تولید کند تا پس از تبدیل به سیگنال‌های استاندارد به کنترل‌ولوها اعمال شود. در سیستم DDC ورودی و خروجی‌های آنالوگ از طریق واسطهای مناسب به کامپیوتر مرتبط می‌گردند. فانکشن‌های کنترلی به‌صورت متمرکز در کامپیوتر اجرا می‌شود. این تمرکز، مدیریت کامل روی لوپ‌ها را

1. Proportional Integral Derivative
2. Single Loop
3. Direct Digital Control

امکان‌پذیر ساخته و در صورت نیاز ارتباط بین آنها را به‌سادگی فراهم می‌کند. در شکل ۱-۵ نمایی از این سیستم نشان داده شده است.



شکل ۱-۵ شماتیک سیستم DDC

این سیستم به‌دلیل سهولت کار کنترل و مانیتورینگ و انعطاف‌پذیری آن برای اجرای الگوریتم‌های مختلف کنترلی با استقبال مواجه شد. ولی عیب بزرگ این سیستم این بود که در صورت بروز مشکل در سیستم کنترل مرکزی، کل فرآیند دچار اختلال می‌شد و همه لوپ‌های کنترلی از کار می‌افتاد، درحالی‌که در سیستم‌های قدیمی‌تر مانند سیستم‌های نیوماتیکی چنین مشکلی وجود نداشت.

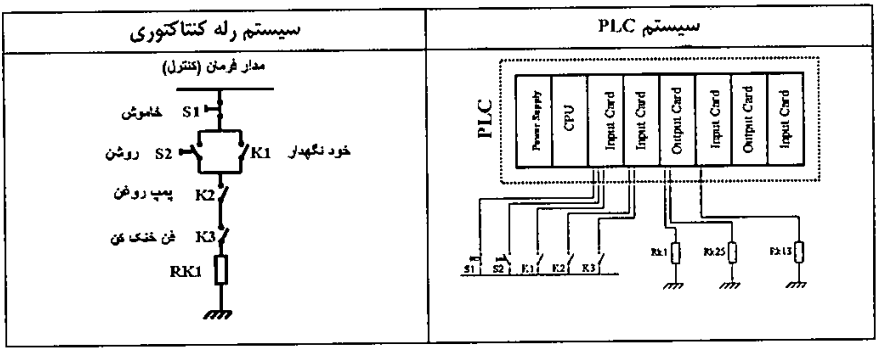
سیستم‌های کنترل PLC و DCS

برای رفع معایب سیستم‌های DDC، متخصصین شروع به چاره‌اندیشی کردند و ایده ایجاد سیستم غیر متمرکز موسوم به DCS [1] مطرح گردید. در ۱۹۷۶ اولین سیستم‌های کنترل غیر متمرکز ارائه شد. در این سیستم‌ها از چندین پردازشگر استفاده می‌شود که هر کدام لوپ‌های کنترلی مستقلی را کنترل می‌کنند و در صورت بروز مشکل در یک پردازشگر، کنترل سایر لوپ‌ها از دست نمی‌رود. به این ترتیب سیستم DDC از عرصه کنترل کنار گذاشته شد و سیستم‌های DCS روز بروز توسعه پیدا کرد. هم‌اکنون نیز در برخی صنایع از DCS به‌عنوان سیستم اصلی کنترل استفاده می‌شود.

اما PLC نیز تقریباً همزمان با DCS (دقیقاً چند سال قبل از آن) متولد شد. با وجود اینکه PLC و DCS دو سیستم مختلف با کاربری متفاوت هستند، ولی با این حال این دو هنوز در صنایع به موازات هم به‌کار می‌روند و هیچ‌کدام نتوانسته است نقش دیگری را به‌طور کامل ایفا نماید.

هدف اولیه از طراحی PLC، جایگزینی پردازشگر میکروپروسسور به‌جای سیستم‌های رله کنتاکتوری بود. از آنجایی که در سیستم‌های رله کنتاکتوری کار بر روی سیگنال‌های دیجیتال مد نظر بود، بنابراین PLC با هدف پردازش این سیگنال‌ها ارائه شد، درحالی که DCS با هدف کنترل لوپ‌های آنالوگ عرضه گردید.

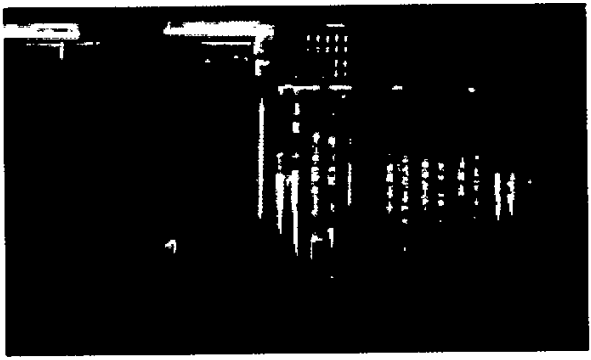
1. Distributed Control System



شکل ۱-۶ جایگزینی PLC به‌جای مدار فرمان رله کنتاکتوری

ساخت اولین PLC در دنیا

تصور کنید که برای کنترل یک سیستم، پنج تابلوی فرمان در کنار تابلوی قدرت نصب شده است. کابل‌های مختلف به تابلوها متصل و سیم‌کشی‌های پیچیده بین اجزای آنها انجام شده است. برای ردیابی این سیستم چندین نقشه وجود دارد. اگر در چنین سیستمی اشکالی پیش آید یا نیاز به تغییر منطق کنترل باشد چقدر کار مشکل خواهد بود؟ بعلاوه هزینه سرویس و نگهداری قطعات الکترومکانیکی نظیر رله و کنتاکتور در دراز مدت تا چه حد زیاد خواهد بود؟



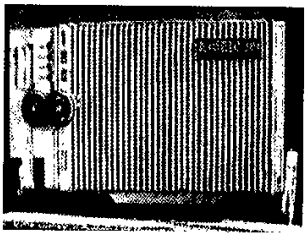
شکل ۱-۷ نمونه تابلو فرمان رله کنتاکتوری

این مشکلی بود که در سال ۱۹۶۸ بخش Hydramatic جنرال موتورز General Motors به‌دنبال حل آن بود و از شرکت‌های مختلف درخواست کرد که پیشنهاد خود را برای جایگزینی سیستمی به‌جای رله کنتاکتورها ارائه کنند.
برنده این پیشنهاد، شرکت Bedford بود. راه حل این شرکت ساخت یک PLC بود که پس از ساخته‌شدن، مدل 084 نام گرفت زیرا این پروژه هشتاد و چهارمین پروژه شرکت Bedford بود.

یکی از افرادی که در پروژه ساخت اولین PLC برای جنرال موتورز فعالیت داشت Richard Morely بود. از این رو برخی او را پدر PLC معرفی می‌کنند.

شکل ۱-۸ از چپ به راست
Morley, Boissevain, Scwerk, Landau.

Bedford پایه‌گذار شرکت Modicon بود که نام آنرا براساس MOdular DIgital CONtroller انتخاب کرد. چند سال بعد یکی از مهندسین این شرکت به‌نام Michael Greenberg اولین PLC به‌معنای صنعتی را با نام 184 عرضه نمود.



شکل ۱-۹ اولین PLC صنعتی – Modicon 184

برند Modicon در اصل آمریکایی است و در سال ۱۹۷۷ به‌Gould Electronics فروخته شد، سپس AEG آلمانی و نهایتاً اشنایدر الکتریک Schneider Electric فرانسه آنرا در اختیار گرفتند. هم‌اکنون PLCهای مدل Quantom و Premium توسط Modicon ساخته و عرضه می‌گردد.

سازندگان پیشکسوت PLC

پیشکسوتان ساخت PLC که در سال‌های ۱۹۶۸ تا ۱۹۷۳ اولین PLC خود را عرضه کردند به‌ترتیب عبارتند از:

- Modicon
- Allen Bradley

- General Electric
- Siemens
- Omron

کمپانی Allen Bradley در سال ۱۹۶۹ شرکت Information Instruments را که از شرکت‌کنندگان در مناقصه جنرال موتورز بود خریداری نموده و شروع به‌کار روی این تکنولوژی جدید کرد. این شرکت اولین PLC خود را در ۱۹۷۰ عرضه کرد. شرکت زیمنس نیز اولین PLC خود را در همین سال ارائه نمود.

از سازندگان دیگری که بعداً به این زمینه وارد شدند می‌توان شرکت‌های زیر را نام برد:

- AEG
- ABB
- Merlin Green
- Toshiba
- Mitsubishi
- Fuji Electric
- Hitachi

استقبال فراگیر از کاربرد PLC

با ظهور PLC ابتدا کسی باور نمی‌کرد که یک دستگاه کوچک بتواند جایگزین آن همه مدارات رله کنتاکتی شود، ولی در عمل وقتی یک برنامه ساده توانست حجم زیادی از سیم‌کشی‌ها و تجهیزات را حذف کند و عملکرد قابل اطمینانی ارائه دهد، کاربرد آن مورد رضایت صنعتگران واقع شد. استفاده گسترده PLC در صنعت از ۱۹۷۲ آغاز گردید و با رشدی سریع ادامه یافت.

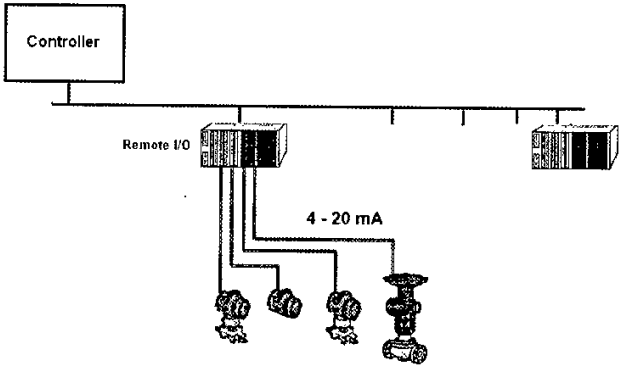
در PLCهای اولیه سیگنال‌ها همگی دیجیتال بودند چون هدف اولیه فقط جایگزینی رله‌ها و کنتاکتورهای کمکی بود ولی در ادامه روند توسعه، قابلیت استفاده از سیگنال‌های آنالوگ نیز به آن اضافه شد. به این ترتیب PLC توانست حتی کنترل و مانیتورینگ لوپ‌های کنترلی را بر اساس سیگنال‌های آنالوگ انجام دهد. PLC به‌دلیل ساختار متمرکز آن، گزینه خوبی برای لوپ‌های کنترلی زیاد نبود و نتوانست عملکرد DCS را ایفا کند.

ورود شبکه‌های صنعتی به عرصه کنترل

توسعه سیستم‌های کنترل از ۱۹۷۰ تاکنون روبه‌رشد بوده ودربرخی موارد این رشد یک جهش بزرگ را به همراه داشته است.

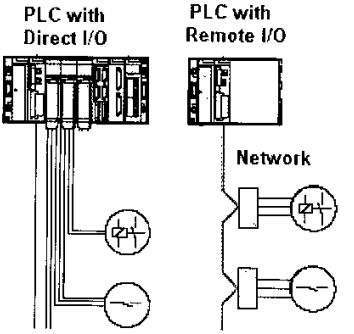
یکی از جهش‌های بزرگ در دهه ۸۰ میلادی اتفاق افتاد و آن استفاده از شبکه‌های صنعتی برای جمع‌آوری I/O ها و انتقال به سیستم‌های کنترل PLC و DCS بود. در این زمان پروتکل‌های مختلف که بر اساس انتقال سریال طراحی شده بود به‌صورت استاندارد عرضه شد.

در این روش واسطهایی بنام Remote I/O مورد استفاده قرار گرفت که می‌توانست در فیلد یا کنار کنترلر نصب شود و سیگنال‌های I/O را به بسته‌های سریال دیتا تبدیل و روی کابل شبکه به‌صورت سریال قرار دهد. از این طریق یک کابل شبکه می‌توانست جایگزین کابل‌های موازی بین کنترلر و فیلد گردد.



شکل ۱-۱۰ جایگزینی شبکه صنعتی به‌جای کابل کشی‌های موازی به سیستم کنترل

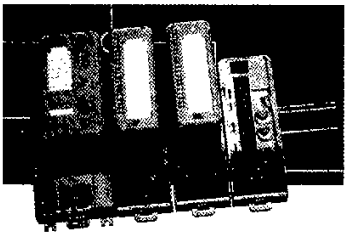
چنین سیستمی مزایای بزرگی را به‌همراه داشت که مهمترین آن کاهش هزینه سیستم به‌دلیل کاهش کابل و مسیر کابل بود. این کاهش، وقتی کاملاً چشمگیر بود که فاصله زیادی بین سیستم کنترل تا فیلد وجود داشت به‌صورتی که در برخی گزارش‌ها از کاهش ۵۰ درصدی هزینه سیستم صحبت به میان می‌آمد. مزیت دیگر این سیستم، دیجیتال بودن و کاهش تأثیر نویز روی آن بود. این سیستم علاوه بر PLC در DCS نیز مورد توجه قرار گرفت و آنها را نیز متحول کرد. در شکل۱-۱۱ یک سیستم PLC در حالت اتصال مستقیم ورودی خروجی‌ها و در حالت استفاده از Remote I/O نشان داده شده است.



شکل ۱-۱۱ مقایسه اتصال I/O به PLC به‌صورت مستقیم نسبت به استفاده از Remote I/O

یکی دیگر از مزایای بزرگ استفاده از شبکه‌های صنعتی استاندارد، ویژگی عدم وابستگی به سازنده خاص یا اصطلاحاً Vendor independent بودن است. این ویژگی موجب می‌شود که بتوان در سطح فیلد از هر نوع Remote I/O ساخت هر سازنده دلخواه استفاده کرد یا آنها را با مدل‌های سازندگان دیگر جایگزین نمود.

به تدریج PLCها و DCSها به ماژول‌های شبکه مجهز شدند و کارت‌ها و پورت‌هایی برای ارتباط با شبکه‌های صنعتی مانند Modbus و Profibus و .... روی آنها تعبیه گردید.



شکل ۱-۱۲ PLC همراه با کارت‌های شبکه صنعتی

کاربرد تکنولوژی فیلدباس در کنترل

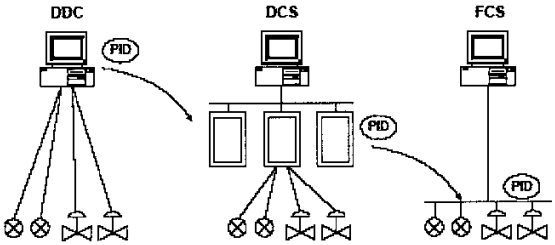
اما سیر تحول سیستم‌های اتوماسیون به اینجا نیز ختم نشد. در دهه ۹۰ میلادی سیستم‌های جدیدی با عنوان FCS[1] عرضه شدند. این تحول مدیون پیشرفتی بود که در ساخت ترانسمیترها و کنترل‌ولوها اتفاق افتاد.

با پیشرفت پردازشگرهای دیجیتال، از سال ۱۹۸۰ سنسورهای هوشمند[2] با قابلیت پردازش ارائه گردید و این اندیشه قوت گرفت که به‌جای اتصال ترانسمیترها و محرک‌ها به Remote I/O می‌توان آنها را مستقیماً به شبکه وصل نمود. بدیهی است وقتی وسایل فیلد دارای پردازشگر باشند می‌توان برخی فانکشن‌های ساده کنترلی را به خود آنها محول کرد و لزومی به اجرای این فانکشن‌ها توسط یک کنترلر مرکزی نیست. این امر علاوه بر اینکه بار کنترلر اصلی را کاهش می‌دهد قابلیت اطمینان سیستم را به‌دلیل استفاده از وسایل هوشمند توزیع شده افزایش می‌دهد.

شکل ۱-۱۳ یک نمونه کنترل‌ولو و چند نمونه ترانسمیتر هوشمند

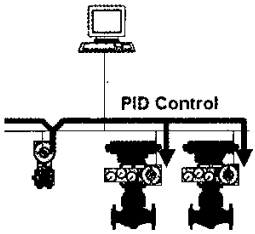
به‌عنوان مثال یک لوپ PID را در نظر بگیرید. در سیستم‌های DDC این لوپ توسط کنترلر مرکزی و در سیستم‌های DCS توسط کنترلرهای توزیع شده کنترل می‌شود؛ در حالی‌که در سیستم FCS این لوپ در خود وسیله پردازش می‌گردد. شکل ۱-۱۴ نشان می‌دهد که چگونه فانکشن کنترل PID با پیشرفت سیستم اتوماسیون جابه‌جا شده و به سطح فیلد نزدیک شده است.

1. Fieldbus Control System
2. Smart



شکل ۱-۱۴ جایگاه لوپ PID در سیستم‌های کنترل مختلف

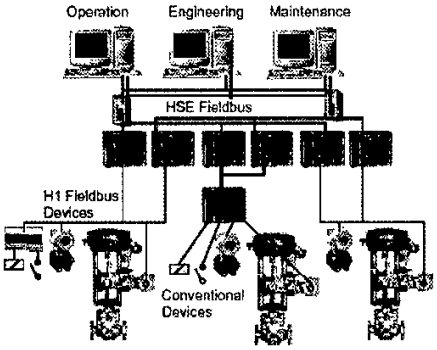
وسایل مبتنی بر FCS توان کنترل لوپ را به‌صورت داخلی و بدون نیاز به کمک از کنترلر اصلی شبکه (PC ، PLC) دارا هستند. در شکل ۱-۱۵ ترانسمیتر پس از اندازه‌گیری سیگنال و مقایسه آن با مقدار مبنا فانکشن PID را شخصاً اجرا نموده و فرمان لازم را به ولوهای کنترلی ارسال می‌کند.



شکل ۱-۱۵ لوپ کنترلی بین ترانسمیتر و کنترل ولو

بنابراین FCS شبکه کنترل متمرکز (DDC) را به‌صورت غیرمتمرکز (DCS) همراه با وسایل هوشمند در می‌آورد. بایستی توجه داشت که در برخی شبکه‌های DCS کار کنترل بین چند کنترلر تقسیم شده ولی باز همه این کنترلرها در یک یا چند پانل قرار گرفته‌اند. ارتباط بین کنترل‌کننده‌ها با فیلد توسط کابل‌کشی‌های موازی با سیگنال‌های 4-20mA صورت گرفته است. درحالی‌که در سیستم‌های FCS نه تنها کنترل به‌صورت DCS است بلکه ارتباط نیز از طریق کابل دو رشته شبکه صورت می‌گیرد.

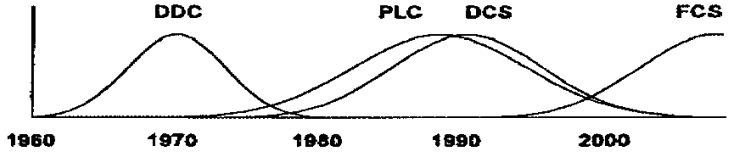
در برخی کاربردها طراحان سیستم کنترل ترجیح می‌دهند از سیستمی ترکیبی استفاده کنند، به‌عبارت دیگر ممکن است PLC یا DCS را همراه با Remote I/O و FCS به‌طور همزمان به‌کار ببرند. نیازهای فرآیند ممکن است طراح را به سمت چنین سیستمی سوق دهد. به‌عنوان مثال سیگنال‌های بسیار مهم و حساس ممکن است توسط کابل‌کشی مستقیم به کارت‌های کنار PLC متصل شوند و سایر سیگنال‌ها از طریق شبکه دریافت شوند. همچنین ممکن است در شرایطی از نظر اقتصادی جایگزینی کل سیستم موجود با سیستم FCS به صرفه نباشد. شکل ۱-۱۶ یک سیستم مختلط را نشان می‌دهد.



شکل ۱-۱۶ سیستم کنترل ترکیبی (مختلط)

با توجه به نکاتی که ذکر شد سیر تحول سیستم‌های اتوماسیون را می‌توان به‌صورت زیر خلاصه کرد:

- ۱۹۴۰ سیستم‌های کنترل و مانیتورینگ نیوماتیکی
- ۱۹۶۰ سیستم های الکتریکی مبتنی براستاندارد 4-20mA (سیستمهای متمرکزDDC)
- ۱۹۷۲ سیستم‌های مبتنی بر PLC
- ۱۹۷۶ سیستم‌های کنترلی غیر متمرکز (DCS)
- ۱۹۹۴ سیستم‌های کنترلی مبتنی بر فیلدباس (FCS)
-



شکل ۱-۱۷ ظهور سیستم‌های مختلف کنترل

اگر به‌طور خاص سیر تحول PLC مورد نظر باشد، می‌توان آنرا به‌صورت جدول زیر خلاصه نمود:

جدول ۱-۱ سیر تحول PLC

۱۹۶۸ مفاهیم اولیه توسعه یافت
۱۹۶۹ ارائه اولین سخت افزار با ۱ کیلوبایت حافظه و ۱۲۸ ورودی و خروجی و دستورات پردازش بیت
۱۹۷۱ اولین کاربرد صنعتی
۱۹۷۲ اضافه شدن دستورات کانتر و تایمر

۱۹۷۳ اضافه شدن دستورات محاسباتی و جابه‌جایی دیتا (MOVE) و برقراری ارتباط بین PC با PLC
۱۹۷۴ توسعه حافظه به ۱۲ کیلوبایت و تعداد ورودی و خروجی به ۱۰۲۴
۱۹۷۵ قابلیت کنترل PID
۱۹۷۸ برقراری ارتباط بین دو یا چند
۱۹۸۳ توسعه حافظه به 4 MB و تعداد ورودی و خروجی به 8192
۱۹۹۳ استانداردسازی زبان‌های برنامه‌نویسی

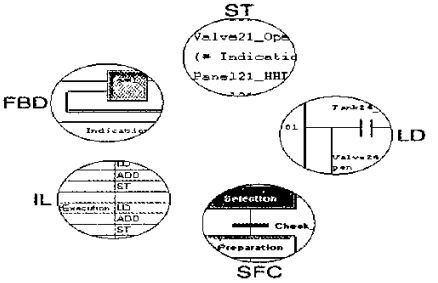
استانداردسازی PLC ها

با پیدایش PLC ها و رضایت صاحبان صنایع از عملکرد آنها، بازار PLC با تقاضای روز افزونی مواجه شد و شرکت‌های مختلفی شروع به ساخت PLC کردند. در ابتدا هیچ‌گونه هماهنگی بین این سازندگان وجود نداشت و طراحی سخت‌افزار و نرم‌افزار برنامه‌نویسی PLCها به‌صورت سلیقه‌ای صورت می‌گرفت. استاندارد IEC برای ساماندهی این اوضاع وارد عرصه شد. سال ۱۹۷۹ یک گروه متخصص در IEC کار بررسی جامع PLC ها را بعهده گرفت. هدف این گروه تدوین روش‌های استانداردی بود که بتواند سخت افزار، برنامه‌نویسی و ارتباطات PLC را پوشش داده و توسط سازندگان PLC بکار گرفته شود. این کار که با مشارکت سازندگان بزرگ PLC دنبال می‌شد حدود ۱۲ سال به‌طول انجامید و نهایتاً پس از بحث‌های موافق و مخالفی که صورت گرفت استاندارد IEC1131 شکل گرفت. در سال ۱۹۹۰ بخش‌های ۱ و۲ و در سال ۱۹۹۳ بخش ۳ و در سال ۱۹۹۵ بخش ۴ این استاندارد تدوین گردید.

تدوین و تصویب این استاندارد گامی بزرگ در رشد سیستم‌های اتوماسیون بود. قالب و چارچوبی که توسط IEC1131 تعیین گردید موجب هماهنگی در ساخت PLC بین سازندگان مختلف گردید. IEC نمی‌گوید ازچه پردازشگری با چه سرعتی و چه میزان حافظه‌ای استفاده شود، ولی مدلی ارائه می‌دهد که هر نوع سخت‌افزاری بایستی بتواند در آن مدل قرار گیرد و تست رضایت بخشی ارائه دهد. مهم‌ترین بخش استاندارد IEC بخش سوم آنست که به IEC1131-3 موسوم می باشد. در این بخش ۵ روش برنامه‌نویسی استاندارد شده و PLC هر سازنده بایستی بتواند همه یا بخشی از آنها را پشتیبانی کند. این زبان‌ها عبارتند از:

- روش دیاگرام نردبانی[1] که LD خوانده می‌شود و ترسیم آن مشابه نقشه‌های مدارات فرمان است.
- روش بلوک دیاگرامی که شبیه گیت‌های مدارات منطقی طراحی می‌شود وبه FBD [2] موسوم است.
- روش کدنویسی سطح پایین شبیه اسمبلی که IL [3]خوانده می‌شود.
- روش ساختاریافته شبیه برنامه‌نویسی‌های سطح بالا مانند پاسکال و C که به ST [4]موسوم است.
- روش گرافیکی برای کنترل ترتیبی که به آن SFC [5] گفته می‌شود.

1. Ladder Diagram
2. Function Block Diagram
3. Instruction List
4. Structrured Text
5. Sequential Function Chart



شکل ۱-۱۸ زبان‌های برنامه‌نویسی PLC در استاندارد IEC

علاوه بر استاندارد شدن روش‌های برنامه‌نویسی، سایر موارد مرتبط با برنامه‌ریزی مانند انواع متغیرها و نحوه تعریف آنها، بلاک‌های کنترلی مانند فانکشن‌ها و فانکشن‌بلاک‌ها و بسیاری از نکات دیگر مربوط به برنامه‌نویسی نیز در این استاندارد تعریف شده است.

سازمان PLCopen

PLCopen یک سازمان غیر انتفاعی است که بدون وابستگی به شرکت یا سازنده خاصی به‌طور تخصصی در زمینه برنامه‌نویسی سیستم‌های کنترل کار می‌کند. این سازمان استاندارد IEC61131-3 را پشتیبانی می‌نماید و تلاش دارد تا روز بروز آنرا گسترش داده و بهبود بخشد.

این سازمان در حال حاضر دارای شش کمیته تخصصی است که هر کدام در زمینه خاصی فعالیت می‌کنند. بسیاری از سازندگان سیستم‌های کنترل در این سازمان عضو هستند از جمله:

- Allen Bradley
- ABB
- Siemens
- Schneider Electric
- Fuji Electric
- Mitsubishi Electric
- HIMA
- Yokogawa
- GE
- Phoenix Contact
- Beckhof
- Omron

استانداردسازی ارتباط بین PLCها و سیستم‌های مانیتورینگ

گام بزرگ دیگری که به سیستم‌های کنترل سروسامان بیشتری بخشید در سال ۱۹۹۶ برداشته شد و آن ارائه OPC[1] بود. تا این زمان امکان ارتباط بین سیستم‌های کنترل مختلف با یک نرم‌افزار استاندارد وجود نداشت. هر سازنده نرم‌افزار خاص خود را برای ارتباط با سیستم کنترل پیشنهاد می‌کرد. به‌عنوان مثال اگر در جایی از PLC ساخت زیمنس استفاده می‌شد و

1. OLE For Process Control

قرار بود که نرم‌افزار سیستم مانیتورینگ از سازنده‌ای غیر از زیمنس باشد ارتباط بین این نرم‌افزار با سیستم کنترل زیمنس با دشواری زیاد انجام می‌گرفت. برای سازندگان دیگر نیز وضعیت به همین منوال بود.

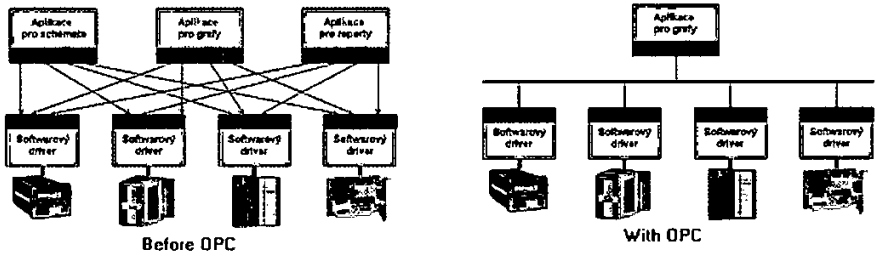
برای رفع این مشکلات، ایده استفاده از یک محیط استاندارد مطرح گردید. سیستم عامل ویندوز در آن سال‌ها به‌خوبی تست شده بود و بسیاری از نرم‌افزارهای مانیتورینگ و مهندسی روی ویندوز نصب می‌گردید.

بر اساس این ایده در سال ۱۹۹۶ برخی کمپانی‌های بزرگ که در زمینه اتوماسیون صنعتی فعالیت می‌کردند کنسرسیومی تشکیل دادند تا نسبت به تهیه استانداردی برای ارتباط سیستم‌های کنترل فرآیند با سیستم‌های مانیتورینگ HMI وکنترل و نظارت از راه دور SCADA[1] اقدام کنند و OPC از همینجا شکل گرفت.

در واقع این قدم مهمی بود که بعد از استاندارد IEC61131 برای استانداردسازی کنترل فرآیند برداشته شد.

سازنده سیستم کنترل ،نرم‌افزاری را تحت عنوان OPC ارائه می‌دهد که با نصب آن، فایل‌های کتابخانه‌ای dll[2] به محیط ویندوز اضافه می‌گردد. با این روش هر نرم‌افزار استاندارد دیگری که روی ویندوز نصب شود به این کتابخانه دسترسی خواهد داشت. به‌عنوان مثال فرض کنید سیستم کنترل زیمنس قرار است با نرم افزار مانیتورینگ غیر زیمنس مانند Citect کار کند. با نصب OPC زیمنس فایل‌های DLL در دسترس نرم افزار Citect قرار خواهد گرفت. بعلاوه در این شرایط می‌توان از هر نرم‌افزار استاندارد دیگر نیز با PLC زیمنس ارتباط برقرار کرد مثلاً با ویژوال بیسیک یا فرترن یا پاسکال و امثال آنها.

اکنون فرض کنید که تعدادی PLC از سازندگان مختلف روی یک شبکه متصل هستند و قرار است که مانیتورینگ آنها با یک نرم‌افزار انجام شود. کافی است OPC سازندگان PLC را روی کامپیوتر مورد نظر نصب کنیم؛ پس از آن با هر نرم‌افزار استاندارد دلخواه همه PLCها می‌توانند ارتباط بگیرند.



شکل ۱-۱۹ ساختار سیستم مانیتورینگ با کنترلرهای مختلف قبل و بعد از به‌کارگیری OPC

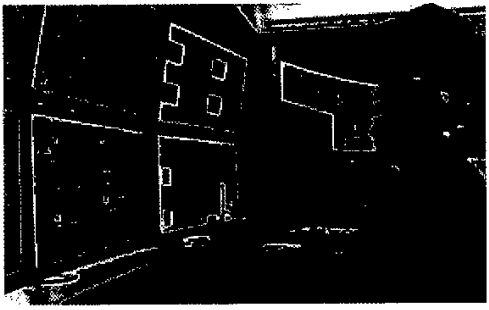
سیستم‌های اسکادا

بررسی سیر تحول سیستم‌های کنترل بیشتر از دیدگاه توسعه PLC و فناوری‌های مربوط به آن تشریح گردید ولی ناگفته نماند که در طول این تحول سیستم‌های دیگری نیز در زمینه کنترل مطرح شده‌اند.

سیستم کنترل نظارتی SCADA از این دسته است.

1. Supervisory Control and Data Acquisition
2. Dynamic Link Library

این سیستم در جایی به‌کار گرفته می‌شود که لازم است بخش‌های مختلف سیستم توسط یک سیستم مرکزی مانیتور و نظارت و در صورت لزوم کنترل شوند. به این سیستم مرکزی اصطلاحاً مرکز دیسپاچری یا دیسپاچینگ گفته می‌شود. سیستم‌های آبرسانی، گازرسانی و برق از جمله مواردی هستند که اسکادا در آنها به‌کار گرفته می‌شود.



شکل ۱-۲۰ سیستم SCADA در مرکز کنترل دیسپاچری

یک سیستم اسکادا برای کنترل و نظارت گاز رسانی را در نظر بگیرید که قرار است ایستگاه‌های مختلف تقویت فشار گاز را زیر نظر بگیرد. هر ایستگاه گاز مجهز به یک کنترل محلی است که RTU[1] خوانده می‌شود. RTU می‌تواند یک PLC باشد. RTUهای ایستگاه‌های گاز اطلاعات مورد نیاز را برای سیستم مرکزی ارسال می‌نمایند که این کار می‌تواند به‌صورت wireless انجام شود از اینرو بعضاً به سیستم اسکادا سیستم تله متری نیز اطلاق می‌گردد. اطلاعات مربوط به مقدار فشار و مصرف و وضعیت هر ایستگاه به سیستم مرکزی مخابره شده و در آنجا مانیتور می‌گردد. سیستم مرکزی در عین حال می‌تواند فرمان نیز صادر کند. این فرامین به ایستگاه‌ها مخابره شده و توسط RTUها اجرا می‌گردد. این فرامین نسبت به فرامین محلی اولویت دارند و گاه ممکن است فرامین محلی را نقض کنند. از سیستم مرکزی می‌توان یک ایستگاه محلی را از سرویس خارج کرد یا مقادیر مبنای کنترل آنرا تغییر داد. لوپ‌های کنترلی در داخل PLCهای محلی بسته می‌شوند و سیستم اسکادا بر آنها نظارت دارد. پس در یک سیستم اسکادا علاوه بر تجهیزات فیلد و تجهیزات کنترل محلی تجهیزات مخابراتی نیز به‌کار می‌رود و سیستم مانیتورینگ و ارتباطی قدرتمندی نیز در مرکز دیسپاچری مورد نیاز است. امروزه توسعه تکنولوژی‌های مرتبط با شبکه‌های صنعتی و مخابرات این سیستم‌ها را متحول کرده و هر روز با تغییرات جدیدی در سخت‌افزار و نرم‌افزار آنها روبه‌رو هستیم.

سسیتم‌های PC Based

در سیستم‌های PC Based ، کامپیوتر نقش سیستم کنترل را ایفا می‌کند. این سیستم اگرچه به‌ظاهر مشابه سیستم‌های DDC قدیمی است ولی از نظر سخت‌افزار و محیط نرم‌افزاری به‌کلی با آن متفاوت است.

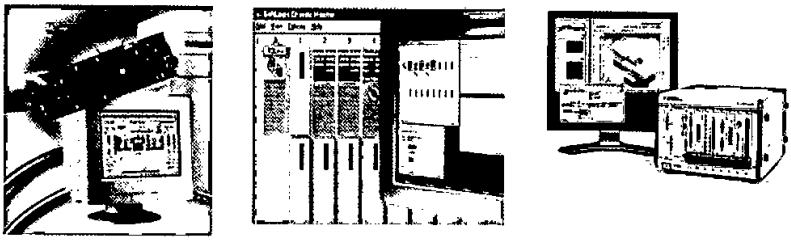
PC based به دو دسته تقسیم می‌شود:

- PC Based نرم‌افزاری
- PC Based سخت‌افزاری

1. Remote Terminal Unit

در نوع اول کامپیوتر می‌تواند با نصب نرم‌افزارهای خاص، الگوریتم‌های مختلف کنترلی را انجام دهد. اگر چه کامپیوتر ممکن است نسبت به کامپیوترهای معمولی مجهز به سخت‌افزارهای بهتری باشد ولی در اصل تفاوتی با آنها ندارد. در این حالت برای ارتباط کامپیوتر با I/Oهای فیلد از شبکه استفاده می‌شود. اگر شبکه از نوع پروفی‌باس باشد نصب کارت شبکه پروفی‌باس روی کامپیوتر الزامی است.

در نوع دوم یعنی نوع سخت‌افزاری یک کنترلر صنعتی یا به‌عبارت دیگر یک CPU که از نظر مدل مشابه CPU به‌کار رفته در PLC است،روی مادربورد نصب می‌گردد و کار کنترل به‌عهده این CPU خواهد بود. در اینجا نیز ارتباط کامپیوتر با ورودی و خروجی‌های موجود در فیلد از طریق شبکه انجام می‌شود و اگر CPU مجهز به پورت شبکه نباشد لازم است کارت شبکه صنعتی نیز روی کامپیوتر نصب گردد.



Softcontrol
ساخت ABB

SoftLogix
ساخت Allen Bradley

PXI
ساخت National Instruments

شکل ۱-۲۱ نمونه‌هایی از سیستم PC Based

از مزایای سیستم PC Based می‌توان به موارد زیر اشاره کرد:

- از نظر اقتصادی نسبت به کنترل‌کننده‌هایی مانند PLC کم‌هزینه‌تر است.
- می‌توان با استفاده از ابزارهای نرم‌افزاری سطح بالا و قوی فانکشن‌های پیچیده کنترلی را به‌راحتی پیاده‌سازی کرد.
- نیاز به مانیتورینگ جداگانه نیست و از PC که نقش کنترلر دارد نیز می‌توان به‌عنوان HMI استفاده کرد.

از معایب آن می‌توان موارد زیر را برشمرد:

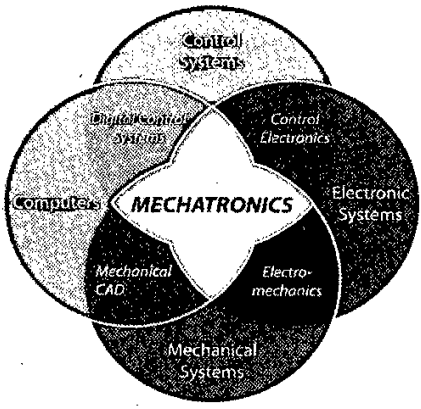
- استفاده از سیستم عامل‌های شناخته شده مانند ویندوز که نسبت به سیستم عامل‌های ناشناخته PLC ها و DCS ها آسیب‌پذیرتر هستند.
- قابلیت اعتماد یا Reliability سخت‌افزار PC Based نسبت به PLC کمتر است چون در PLC همه اجزا حتی منبع تغذیه به‌صورت صنعتی طراحی شده‌اند و در محیط‌هایی که شرایط سخت دارند به‌راحتی کار می‌کنند، ولی در PC Based ممکن است همه‌ی اجزا دارای چنین قابلیتی نباشند.
- برگشت PC به کار نرمال در زمان قطع و وصل تغذیه طولانی‌تر از PLC است .

- استفاده همزمان از نرم‌افزارهای دیگر می‌تواند روی عملکرد کنترلر تاثیرگذار باشد و در اجرای برنامه کنترل تأخیر ایجاد کند.

وضعیت کنونی سیستم‌های کنترل و افق‌های آینده

روند پیشرفت‌ها همچنان ادامه دارد، CPUهای سیستم‌های کنترل به‌صورت Onboard به ارتباطات شبکه مجهز شده‌اند. ارتباطات بدون سیم حتی در سطح ادوات فیلد پیاده‌سازی شده است و امکان سرویس‌های از راه دور نظیر ارتباط با سیستم کنترل از طریق خط تلفن یا اینترنت فراهم گردیده است. ربات‌ها در صنعت به‌طور وسیع به‌کار گرفته شده‌اند، دانش اپراتوری به دل سیستم‌های کنترل راه یافته و بلاک‌های کنترلی مبتنی بر تجربه و دانش اپراتوری تهیه شده‌اند. منطق فازی و شبکه‌های عصبی توسعه یافته و در کنار همه این موارد، دانش کنترل نیز در تئوری و کاربرد دستخوش تحولات و پیشرفت‌های بسیار شده است.

امروزه نه تنها الکترونیک و مخابرات و کنترل به هم گره خورده‌اند و هرگونه نوآوری در یکی از زمینه‌ها دیگری را نیز تحت تأثیر قرار می‌دهد بلکه سایر علوم نیز از این زنجیره خارج نیستند. تحولاتی که در مکانیک و هیدرولیک و نیوماتیک و دیگر رشته‌ها از جمله نانو تکنولوژی رخ می‌دهد نیز بر روند توسعه اتوماسیون صنعتی و کنترل تأثیر می‌گذارد. شناخت این وابستگی‌ها و کاربرد آ نها منجر به تاسیس رشته جدیدی به نام مکاترونیک گردیده است. واژه مکاترونیک ابتدا توسط تسورو موری ژاپنی در ۱۹۶۹ به‌کار گرفته شد. این رشته که در سال ۱۳۸۳ به رشته‌های آموزش عالی کشور اضافه شد، تلفیقی است از رشته‌های مختلف مکانیک و الکترونیک و کنترل و کامپیوتر و که رباتیک زیر شاخه‌ای از آن محسوب می‌شود.



شکل ۱-۲۲ مکاترونیک و علوم مرتبط با آن

اکنون که در دهه ۲۰۱۰ میلادی قرار داریم، روز بروز شاهد عرضه سیستم‌های کنترلی مجهزتر و تواناتر هستیم. پیشرفت حیرت‌آور تکنولوژی در سال‌های اخیر هر روز پنجره‌های جدیدی از دانش را به روی بشر گشوده است. اینکه تکنولوژی نانو، توسعه سیستم‌های بدون سیم، افزایش سرعت و اطمینان شبکه‌های صنعتی، سایر ایده‌ها و فناوری‌های

جدید در آینده چه تحولاتی ایجاد خواهد کرد بر کسی روشن نیست. آنچه روشن است این است که بشر هر چه بیشتر به عمق اقیانوس بیکران دانش فرو می‌رود، بیش از همه به میزان جهل و نادانی خود پی می‌برد.

۱-۳ مزایا و معایب سیستم‌های کنترل مدرن

در این قسمت به اختصار به برخی از مزایا و معایب سیستم‌های مدرن نسبت به سیستم‌های قدیمی اشاره می‌شود؛ ولی از آنجا که محوریت موضوع این کتاب PLC است در ادامه سیستم‌های PLC با سایر سیستم‌های کنترلی به تفصیل مقایسه شده‌اند.

مزایای سیستم‌های کنترل مدرن

به‌طور خلاصه می توان به موارد زیر اشاره کرد:

- کاهش تجهیزات از نظر حجم و تعداد
- کاهش هزینه‌های نصب
- کاهش هزینه‌های سرویس و نگهداری
- سهولت عیب‌یابی سیستم
- سهولت توسعه سیستم
- امکان انجام عملیات مهندسی در حین کار
- امکان کنترل دقیق پارامترهای مرتبط با کیفیت محصول
- امکان مانیتورینگ به‌صورت متمرکز یا گسترده
- امکان انتقال تجربه و دانش اپراتور به سیستم
- و...

معایب سیستم‌های کنترل مدرن

برای سیستم‌های مدرن نسبت به سیستم‌های قدیمی معایب چندانی نمی‌توان ذکر کرد با این وجود به موارد زیر می‌توان اشاره نمود:

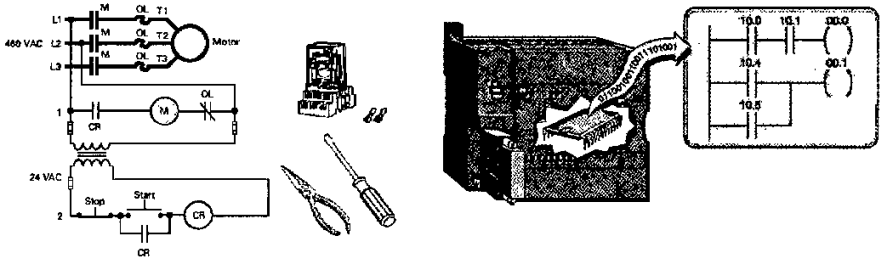
- نیاز به افراد متخصص
- عدم امکان ساخت و تعمیر بسیاری از اجزای سیستم
- نیاز مداوم به ارتقاء سیستم به‌دلیل عدم تولید محصولات قبلی توسط سازنده. سیاست بسیاری از سازندگان سیستم‌های مدرن در جهت فروش بیشتر سیستم‌های جدید است از اینرو تولید سیستم‌های قدیمی را متوقف کرده و مشتری را ناچار به ارتقاء سیستم می‌کنند.

۱-۴ مقایسه PLC با سایر سیستم‌های کنترل و مزایا و معایب آن

۱-۴-۱ مقایسه PLC با سیستم‌های رله کنتاکتی

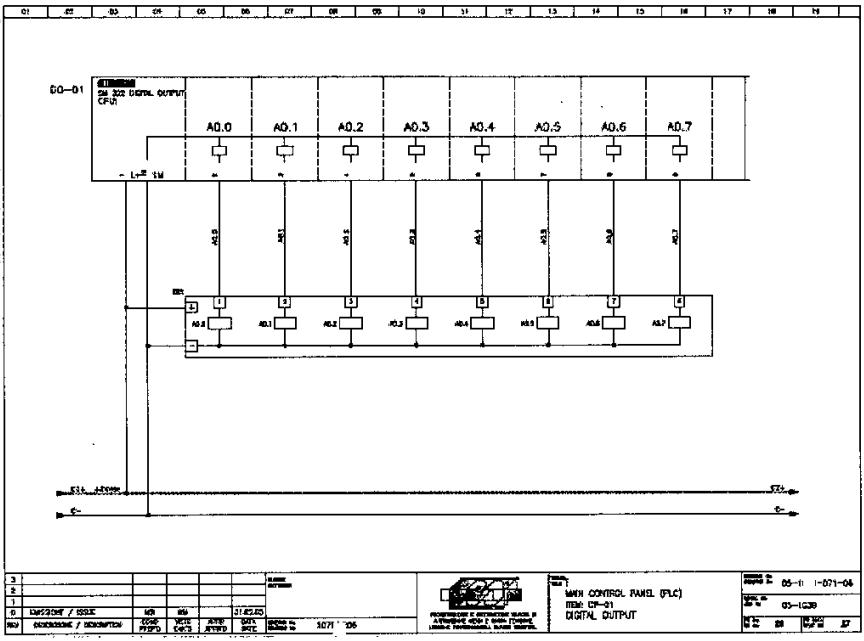
از آنجا که هدف اصلی از طراحی PLC جایگزینی آن به‌جای مدارات رله کنتاکتوری بوده است، این دو سیستم را با جزئیات مربوطه مورد مقایسه قرار می‌دهیم. به نکات زیر توجه شود:

- **سیم کشی کمتر:** سیستم PLC نسبت به رله‌کنتاکتی، سیم‌کشی کمتری نیاز دارد زیرا منطق کنترل به‌صورت نرم‌افزاری پیاده‌سازی می‌شود؛ در حالی‌که در سیستم رله‌کنتاکتی منطق کنترل توسط سیم‌کشی و با اتصال المان‌های سخت‌افزاری ساخته می‌شود. به‌عنوان مثال اگر نیاز باشد که سیستمی با فشار دادن یک شستی روشن و با فشار دادن مجدد همان شستی خاموش شود، پیاده‌سازی این منطق با سیم‌کشی کار پیچیده‌ای است در حالی‌که با برنامه‌نویسی به سهولت انجام می‌شود.
- **تجهیزات کمتر:** برخی المان‌های سخت‌افزاری سیستم قدیمی، مانند تایمرها و کانترها و رله‌های کمکی در سیستم PLC حذف شده و توسط نرم‌افزار پیاده‌سازی می‌گردند.



شکل ۱-۲۳ مقایسه PLC با مدار فرمان قدیمی

- **فضای کمتر:** به‌دلیل کاهش کابل‌کشی و کاهش تعداد عناصر، تابلوی PLC حجم و فضای بسیار کمتری نسبت به تابلوهای فرمان رله‌کنتاکتوری اشغال می‌کند.
- **عیب‌یابی ساده‌تر:** عیب‌یابی سیستم PLC بسیار ساده‌تر از عیب‌یابی سیستم رله‌کنتاکتی است. در سیستم قدیمی، نقشه‌های زیادی برای مدار فرمان وجود دارند که برای یافتن عیب بعضاً بایستی مسیر سیم‌کشی‌ها را بر اساس این مدارک جستجو و بررسی کرد؛ ولی در PLC نقشه‌ها بسیار ساده هستند. به‌عبارت دیگر نقشه خاص و پیچیده‌ای وجود ندارد.شکل ۱-۲۴ نقشه سیم‌کشی از یک کارت Digital Output به خروجی‌ها را نشان می‌دهد. سادگی این نقشه از شکل مشخص است. در این نقشه فقط مسیر سیم‌کشی از کارت تا خروجی لازم است ترسیم شود زیرا قبل از آن هیچ Wiring خاصی وجود ندارد و برنامه PLC است که به ترمینال‌های این کارت فرمان می‌فرستد.



شکل ۱-۲۴ مقایسه نمونه نقشه سیم‌کشی سیستم PLC

برای کارت‌های ورودی نیز به همین شکل کار ساده است. کافی است مسیر نقشه از سنسور تا کارت ترسیم گردد. ارتباط بین ورودی و خروجی‌ها نرم‌افزاری است و با برنامه انجام می‌پذیرد. بررسی منطق کنترل نیز توسط ابزارهای نرم‌افزاری دنبال می‌شود که بسیار ساده‌تر از ردیابی سیم‌کشی‌هاست.

- **عدم نیاز به سرویس و نگهداری:** نگهداری سیستم PLC کم‌هزینه‌تر از سیستم‌های قدیمی است. عناصر سخت‌افزاری مانند رله و کنتاکتور نیاز به سرویس و مراقبت دارند و بعد از کارکرد مشخصی بایستی تعویض شوند، در حالی‌که اجزای PLC نیازمند این گونه سرویس‌ها و تعویض‌ها نیستند.
- **سهولت توسعه سیستم:** توسعه سیستم مبتنی بر PLC بسیار ساده‌تر از توسعه سیستم رله‌کنتاکتی است زیرا در سیستم قدیمی ممکن است برای تغییر یا توسعه سیستم نیاز به تغییر سیم‌کشی‌ها و انجام سیم‌کشی جدید باشد. در حالی‌که در PLC تغییر و حذف و اضافه در برنامه، بدون دستکاری سیم‌کشی بیرونی یا با تغییرات مختصر در سیم‌کشی انجام می‌گردد.
- **طراحی ساده‌تر:** در فاز طراحی، سیستم‌های قدیمی کار بیشتری نیاز دارند و مستندات و نقشه‌های بیشتری بایستی برای آنها تهیه کرد ولی در سیستم PLC کار ساده‌تر است.

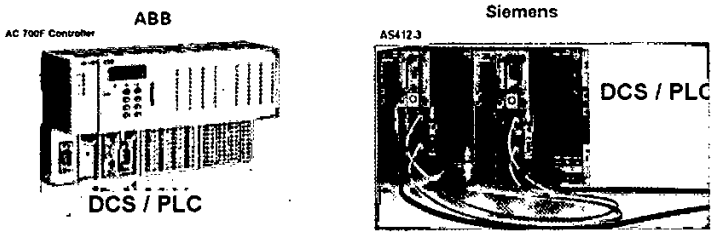
- **امکان اجرای محاسبات پیچیده:** انجام محاسبات پیچیده ریاضی در سیستم‌های قدیمی امکان‌پذیر نیست ولی در PLC به‌سادگی قابل اجراست.
- **تاخیر کمتر:** در سیستم‌های قدیمی وقتی از رله‌ها و کنتاکتورها برای پیاده‌سازی منطق کنترل استفاده می‌شود، اعمال فرمان به سیستم با تأخیر همراه خواهد بود. المان‌های الکترومکانیکی واسط، منجر به تأخیر می‌شوند. بایستی توجه داشت که این تاخیر در PLC وجود ندارد. البته اگر در سیستم قدیمی المان‌های تاخیر دهنده بین مسیر نباشند، پاسخ سیستم قدیمی سریعتر از PLC خواهد بود. به‌عنوان مثال اگر قرار باشد با فشردن یک شستی کنتاکتوری برقدار شود، در سیستم قدیمی فوراً این اتفاق می‌افتد ولی در PLC چون بایستی وضعیت شستی خوانده شود و پس از اجرای برنامه فرمان ارسال گردد با تاخیر کوتاهی به اندازه زمان اجرای برنامه روبرو خواهیم شد. اگر حجم برنامه زیاد باشد این تاخیر بیشتر خواهد بود ولی به‌طور معمول کمتر از ۱۰۰ میلی ثانیه است. البته باید توجه نمود که در سیستم قدیمی، معمولاً فرمان از یک شستی به یک مصرف‌کننده داده نمی‌شود و المان‌های واسط زیادی در این بین استفاده می‌گردد. از اینرو در این سیستم‌ها میزان تاخیر نسبت به PLC بیشتر خواهد بود.
- **هزینه کمتر:** از نظر اقتصادی با در نظر گرفتن فاکتورهایی مانند تعداد عناصر سخت‌افزاری، سرویس و نگهداری و زمان‌های توقف سیستم جهت عیب‌یابی، هزینه سیستم قدیمی بسیار بیشتر از هزینه سیستم PLC است.

۱-۴-۲ مقایسه PLC با سیستم‌های DDC

اگر چه PLC برای حذف سیستم‌های DDC که بر مبنای پردازش سیگنال‌های آنالوگ کار می‌کرد به میدان نیامد ولی اگر با این دیدگاه که PLC نیز قادر است آنالوگ‌ها را پردازش کند و به‌صورت متمرکز شبیه DDC لوپ‌های کنترلی را مدیریت و کنترل کند نگاه کنیم، می‌توان گفت که استفاده از PLC نسبت به DDC مزیت دارد. سیستم‌های کامپیوتری به اندازه سیستم PLC قابل اعتماد نیستند. اگر سخت‌افزار کامپیوتر حتی به‌صورت صنعتی طراحی شده باشد، باز به دلیل سیستم عامل‌های شناخته شده کامپیوتر، امکان بروز مشکلاتی مانند ویروس‌های رایانه‌ای و برخی مشکلات مربوط به سیستم عامل‌ها و اختلال در کار کامپیوتر وجود دارد. در PLC، از آنجایی که سیستم عامل ناشناخته است امکان نفوذ به آن فقط توسط متخصصین سازنده امکان‌پذیر است.

۱-۴-۳ مقایسه PLC با سیستم‌های DCS

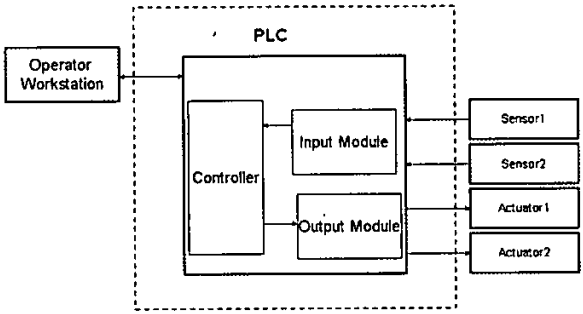
DCS برای حذف DDC و رفع مشکلات آن عرضه شد، از اینرو کاربری آن با کاربری PLC متفاوت است. پرسشی که هنوز در ذهن بسیاری از افراد علاقه‌مند به اتوماسیون صنعتی وجود دارد این است که چه تفاوت‌های اساسی بین PLC و DCS وجود دارد. این سوال به‌ویژه وقتی مطرح می‌شود که می‌بینیم با پیشرفت تکنولوژی، توانایی سیستم‌های DCS و PLC بسیار به هم نزدیک شده است، به‌صورتی که برخی سازندگان برای هر دو سیستم یک نوع CPU ارائه کرده‌اند. شکل ۱-۲۵ نمونه‌هایی از سیستم دو منظوره را نشان می‌دهد. از اینرو با توجه به ظاهر سیستم نمی‌توان تشخیص داد که سیستم کنترل PLC است یا DCS . برای یافتن این سوال بایستی به پارامترهای مختلف مانند نحوه عملکرد، نحوه شبکه‌بندی و نحوه طراحی منطق برنامه دقت کرد.



شکل ۱-۲۵ سیستم کنترل‌های قابل استفاده به‌عنوان PLC یا DCS

مواردی که می‌توان برای تشخیص تفاوت PLC و DCS به آنها اشاره کرد عبارتند از:

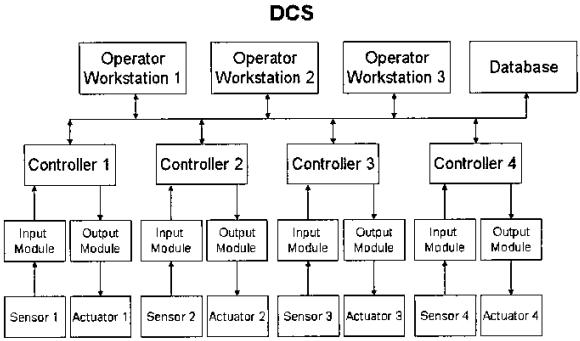
- **تفاوت معماری:** معماری PLC بر اساس یک پردازشگر است که ورودی‌ها و خروجی‌ها با آن در ارتباط هستند. به شکل ۱-۲۶ توجه کنید.



شکل ۱-۲۶ شماتیک کلی معماری PLC

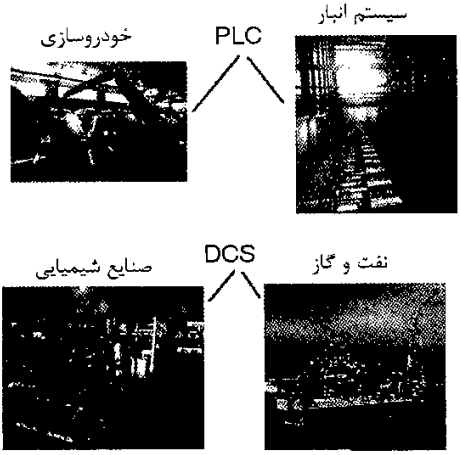
در سیستم DCS بر خلاف PLC پردازش متمرکز نیست و از کنترلرهای مجزایی که با یکدیگر شبکه شده‌اند استفاده شده است. این کنترلرها در عین حال از طریق شبکه به سیستم‌های اپراتوری متصل هستند و مقادیر مبنا را از آنها دریافت کرده و اطلاعات فرآیند را به آنها ارسال می‌نمایند.

بنابر این برخلاف PLC، یک سیستم کنترل غیر متمرکز(DCS) دارای پردازشگرهای مستقلی است که کار کنترل بین آنها تقسیم شده است. هر کدام از این پردازشگرها قابلیت کنترل چند لوپ را دارند. ورودی و خروجی‌ها از فیلد به کنترلر مربوطه اتصال می‌یابد. این موضوع در شکل ۱-۲۷ نشان داده شده است.



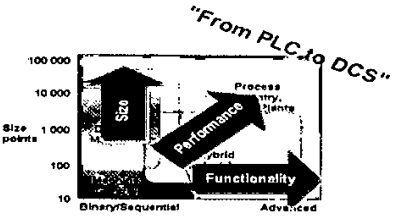
شکل ۱-۲۷ شماتیک کلی معماری DCS

- **تفاوت کاربری:** DCS و PLC سال‌ها در کنار یکدیگر برای دو کاربری متفاوت استفاده شده و هنوز استفاده می‌شوند. ماهیت متفاوت فرآیندها اقتضا می‌کند که در برخی از آنها DCS و در برخی دیگر PLC به‌عنوان سیستم کنترل اصلی به‌کار رود. به‌طور کلی در کنترل فرآیندهایی که لوپ کنترلی و سیگنال‌های آنالوگ زیاد دارند و اصطلاحاً Continuous Process نامیده می‌شوند بهترین سیستم کنترل DCS است. در فرآیندهایی که سیگنال دیجیتال زیاد دارند و کنترل عمدتاً به‌صورت on/off است و اصطلاحاً Discrete Process نامیده می‌شوند بهترین سیستم کنترل PLC است. در شکل ۱-۲۸ برخی از کاربردهای PLC و DCS نشان داده شده است.



شکل ۱-۲۸ برخی از کاربردهای PLC و DCS در صنایع

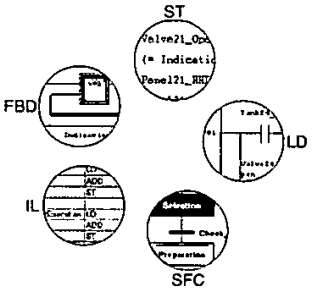
- **تفاوت عملکردی:** زمانی این دو سیستم دو کاربری کاملاً مجزا داشتند، از PLCها برای کنترل سیگنال‌های Discrete و از DCS برای لوپ‌های کنترلی استفاده می‌شد. امروزه با پیشرفت تکنولوژی، این دو سیستم می‌توانند در نقاطی هم‌پوشانی داشته باشند. یعنی از PLC برای کنترل لوپ نیز استفاده شود و DCS سیگنال‌های دیجیتال را نیز پردازش کند. ولی این نوع استفاده با کاستی‌هایی همراه است.



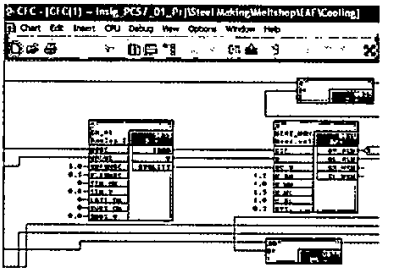
شکل ۱-۲۹ حوزه عملکرد PLC و DCS

وقتی از PLC برای کنترل لوپ استفاده می‌شود و تعداد لوپ‌ها زیاد باشد سیستم ریسک‌پذیر خواهد بود و همانطور که قبلاً اشاره شد، بروز مشکل در CPU منجر به اختلال در کار تمام لوپ‌ها خواهد شد. این مشکل در DCS وجود ندارد. اکنون به‌طور عکس تصور کنید که DCS قرار است کار PLC را انجام دهد. این امر به‌ظاهر از نظر طراحی منطقی مشکلی ندارد ولی مشکل آنجاست که در DCS به‌دلیل استفاده از روش‌های سطح بالا در برنامه‌نویسی، حجم کد تولیدی نسبت به کد ایجاد شده در زبان‌های سطح پایین که در PLC استفاده می‌شود بیشتر خواهد بود و به همین علت سرعت واکنش DCS نسبت به PLC کندتر است.

- **روش‌های برنامه‌نویسی:** برنامه‌نویسی در DCS به‌صورت چارت گرافیکی‌است‌و با ابزارهایی‌نظیر CFC[1] انجام می‌گیرد که سطح بالا محسوب می‌شود؛ در حالی‌که در PLC زبان‌های برنامه‌نویسی سطح پایین نظیر دیاگرام نردبانی به‌کار می‌رود. اگر در DCS از زبان برنامه‌نویسی سطح پایین استفاده شود قابلیت‌هایی مانند امکانات مانیتورینگ از دست می‌رود و اگر در PLC از زبان برنامه نویسی سطح بالا مانند CFC استفاده شود، بار پردازش آن افزایش خواهد یافت.



روش‌های برنامه‌نویسی PLC



روش برنامه‌نویسی DCS

شکل ۱-۳۰ مقایسه روش‌های برنامه‌نویسی PLC با روش برنامه‌نویسی DCS

1. Continuous Function Chart

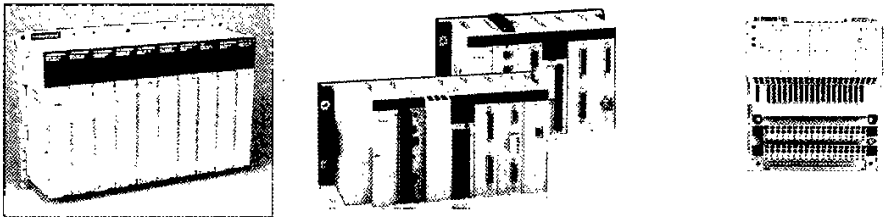
- **امکانات مانیتورینگ:** طراحی گرافیک و ابزارهای سیستم مانیتورینگ در DCS ساده‌تر است. در DCS بسیاری از امکانات توسط سیستم فراهم شده ولی در استفاده از PLC معمولاً امکانات مانیتورینگ از ابتدا توسط کاربر طراحی می‌شوند که کار مهندسی بیشتری را طلب می‌کند.
- **هزینه:** هزینه سیستم DCS بسیار بیشتر از هزینه سیستم PLC است.

۱-۵ معرفی سازندگان PLC

از دهه ۱۹۷۰ تاکنون سازندگان زیادی اقدام به ساخت و عرضه PLC نموده‌اند. شرکت‌های Modicon، Siemens، ABB ، Allen Bradley، Omron، Mitsubishi، GE Fanuc، HIMA، LG از این جمله‌اند. برخی از شرکت‌های معروف در زمینه ساخت PLC در ادامه به اختصار معرفی می‌شوند.

Modicon

Modicon اولین سازنده PLC در دنیاست و هنوز نیز در زیرمجموعه Schneider Electric به ساخت و عرضه PLC ادامه می‌دهد. برخی از مدل‌های جدید Modicon در شکل۱-۳۱ نشان داده شده‌اند، PLCهای مدل Premium و Quantom مدولار بوده و برای کاربردهای بزرگ به‌کار می‌روند. نوع Momentum یکپارچه است و برای کاربردهای کوچک طراحی شده است. برنامه‌نویسی این PLCها با نرم‌افزار UnityPro انجام می‌شود.



شکل۱-۳۱ PLC های ساخت Modicon

Allen Bradley

Allen Bradley از پیشگامان ساخت PLC در دنیاست و اولین PLC خود را ۱۹۷۰ میلادی عرضه کرده است. این شرکت مدل‌های مختلفی از PLC را برای کاربردهای کوچک تا بزرگ ارائه نموده است، از جمله:

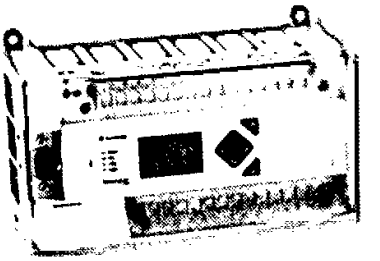
Pico Controller

یک مینی PLC کوچک به‌صورت یکپارچه با تعداد ورودی و خروجی خیلی کم است. برنامه‌ریزی آن از طریق صفحه نمایش روی آن یا از طریق نرم‌افزار Picosoft امکان‌پذیر است.

شکل ۱-۳۲ PLC مدل Pico

Micrologix

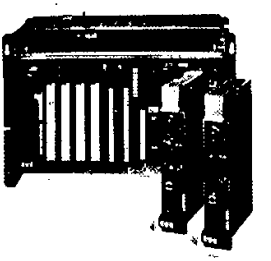
این مینی PLC به‌صورت یکپارچه است و برای کاربردهای کوچک طراحی شده است. تعداد ورودی و خروجی و توانایی‌های آن کم ولی بیش از نوع Pico می‌باشد. برنامه‌نویسی آن با نرم‌افزار RSlogix انجام می‌شود.



شکل ۱-۳۳ PLC مدل Micrologix

PLC-5

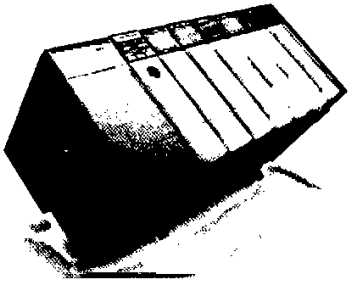
این PLC برای تعداد ورودی و خروجی زیاد با ساختاری محکم و کاملاً صنعتی طراحی و ارائه شده است. برنامه‌نویسی آن با نرم‌افزار RSlogix 5 انجام می‌شود.



شکل ۱-۳۴ PLC مدل PLC-5

SLC-500

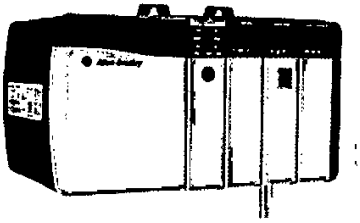
این PLC با ساختار مدولار برای ورودی و خروجی‌های متوسط (کمتر از PLC5) با قابلیت‌های مناسب طراحی و عرضه شده است. برنامه‌نویسی آن با نرم‌افزار RSlogix 500 انجام می‌شود.



شکل ۱-۳۵ PLC مدل SLC-500

ControlLogix 5000

این PLC برای تعداد ورودی و خروجی زیاد با ساختاری انعطاف‌پذیر و پیشرفته طراحی و ارائه شده است. برنامه‌نویسی آن با نرم‌افزار RSlogix 5000 انجام می‌شود.



شکل ۱-۳۶ PLC مدل Contrllogix5000

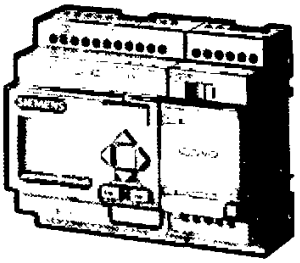
Siemens

SIEMENS

شرکت زیمنس نیز از شرکت‌هایی است که اولین PLC خود را حدود ۱۹۷۲ عرضه نموده و از سازندگان معروف در این زمینه محسوب می‌شود. زیمنس مدل‌های متنوعی از انواع PLC برای کاربری‌های مختلف می‌سازد که نمونه‌هایی از آنها را در شکل‌های زیر مشاهده می‌کنید. توضیحات کامل در مورد انواع PLCهای زیمنس در فصل های بعد آمده است.

LOGO

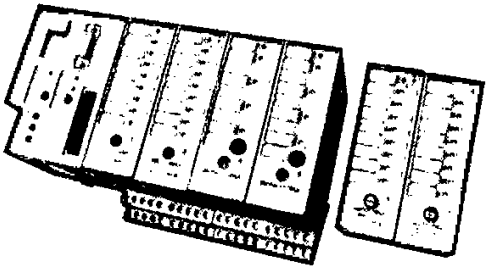
یک مینی PLC برای کاربردهای کوچک است که برنامه‌نویسی آن توسط صفحه کلید یا از طریق نرم‌افزار LogoSoft انجام می‌شود.



شکل ۱-۳۷ PLC LOGO

S5

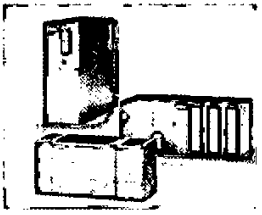
خانواده PLC های S5 قدیمی هستند و در حال حاضر تولید نمی‌شوند. S5 در مدل‌های مختلف از نوع یکپارچه تا انواع مدولار عرضه می‌شده است و برنامه‌نویسی آن با نرم‌افزار Step5 انجام می‌شود.



شکل ۱-۳۸ PLC S5

S7

خانواده جدید PLCهای زیمنس هستند که به S7-200 و S7-300 و S7-400 و S7-1200 تقسیم می‌شوند. برنامه‌نویسی آنها با نرم‌افزار Step7 انجام می‌شود.



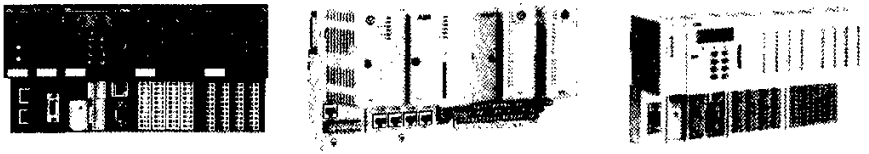
شکل ۱-۳۹ PLC S7

ABB

شرکت ABB نیز یکی از قدیمی‌ترین شرکت‌های سازنده PLC است که ابتدا نام آن BBC بود و سپس با ترکیب BBC و ASEA شرکت ABB تاسیس شد.

این شرکت، PLCهای مختلفی برای کاربردهای متفاوت عرضه می‌کند. نوع AC800M به‌صورت مدولار است و مدل‌های مختلفی دارد که از نظر سرعت و حجم حافظه و توانایی متفاوت هستند. نوع AC700F نیز مدولار بوده و می‌تواند به‌عنوان PLC یا DCS به‌کار رود.

نرم‌افزاری که برای برنامه‌نویسی استفاده می‌شود Control Builder نام دارد.

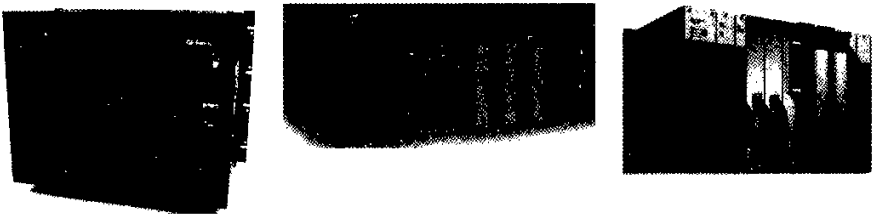


AC500 AC800M AC700F

شکل ۱-۴۰ نمونه PLCهای ABB

GE Fanuc

GE Fanuc یکی از سازندگان معروف PLC است که PLCهای خود را در مدل‌های متنوعی ارائه نموده است. مدل Series 90-70 برای کاربردهای وسیع و بزرگ و مدل Series 90-30 برای کاربردهای متوسط ارائه شده‌اند.

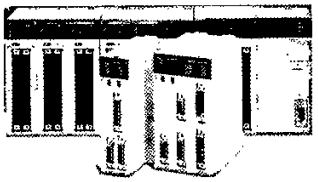


GE FANUC Series 6 GE FANUC Series 90-30 GE FANUC Series 90-70

شکل ۱-۴۱ مدل‌های مختلف PLC ساخت GE Fanuc

Omron

شرکت OMRON نیز یکی از پیشکسوتان ساخت PLC است که تاکنون کنترل‌کننده‌های متنوعی را عرضه نموده است. برخی از نمونه‌های خانواده Sysmac این سازنده در شکل ۱-۴۰ نشان داده شده است. این PLCها با نرم‌افزار CX-Programmer برنامه‌ریزی می‌شوند.


SYSMAC CS

SYSMAC CJ

Zen

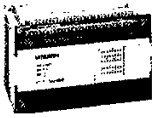
شکل ۱-۴۲ نمونه PLC های ساخت شرکت Omron

برای کاربردهای کوچک مینی PLC این شرکت که به نام Zen معروف می‌باشد، استفاده می‌شود. برای برنامه‌نویسی آن به‌جز صفحه نمایش خود PLC می‌توان از نرم‌افزار Zensupportsoft نیز استفاده کرد.

Mitsubishi

میتسوبیشی نیز از پیشگامان ساخت PLC است که هم‌اکنون PLC های مدولار، مینی PLC و میکرو PLC در مدل‌های مختلف عرضه می‌کند. نرم‌افزار برنامه‌نویسی این PLCها، GX Developer می‌باشد.

System Q PLC

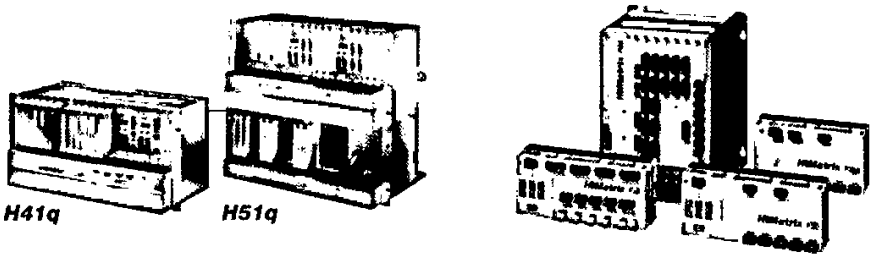

FX0

Micro PLC Alpha

شکل ۱-۴۳ نمونه PLCهای ساخت شرکت Mitsubishi

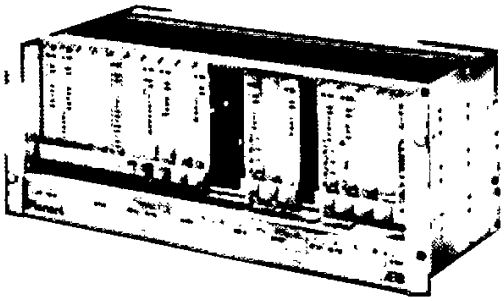
HIMA

HIMA یکی از سازندگانی است که با عرضه سیستم‌های کنترلی خاص برای Safety (ایمنی) معروف شده است. PLC مدولار مدل H41q/H51q و مدل یکپارچه HIMatrix برای ایمنی تا سطح SIL3 است که با نرم‌افزار ELOP II برنامه‌ریزی می‌شود.



شکل ۱-۴۴ مدل‌های مختلف PLC شرکت HIMA

مدل Planar4 برای سطح ایمنی بالا (SIL4) طراحی شده و منطق کنترل به‌صورت سخت افزاری پیاده‌سازی می‌شود.



شکل ۱-۴۵ PLC مدل Planar4 ساخت HIMA

LG

LS Industrial Systems

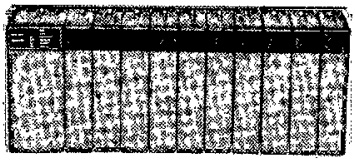
LG از شرکت‌هایی است که در سال‌های اخیر به جمع تولیدکنندگان PLC پیوسته است و مدل‌های مختلفی به‌صورت مدولار و یکپارچه عرضه می‌کند. PLC مدل XGT این شرکت با نرم‌افزار XG5000 و مدل Glofa با نرم‌افزار GMWIN برنامه‌ریزی می‌شود.

XGT GLOFA

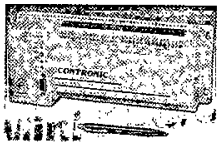
شکل ۱-۴۶ مدل‌های مختلف PLC شرکت LG

Contronic

کنترونیک یک شرکت ایرانی سازنده PLC است. PLC جدید این شرکت PLC500 نام دارد که به‌صورت مدولار است و توسط نرم‌افزار MULTIPROG-wt برنامه‌نویسی می‌شود. نوع یکپارچه آن Mini PLC/C است که با نرم افزار Ladwin برنامه نویسی می‌گردد.



PLC500 NSeries

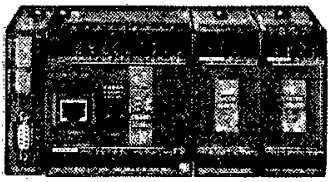


Mini PLC/C

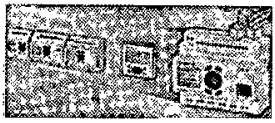
شکل ۱-۴۷ نمونه PLC های ساخت شرکت کنترونیک

برخی دیگر از سازندگان PLC

سازندگان PLC به موارد فوق محدود نمی‌شوند ولی ارائه لیست کامل آنها در این کتاب ضرورتی ندارد. لذا علاقه‌مندان جهت اطلاعات بیشتر می‌توانند به سایت اینترنتی شرکت‌های مختلف فعال در زمینه‌ی اتوماسیون صنعتی مراجعه نمایند. نمونه‌ای از PLCهای برخی دیگر از این سازندگان در شکل زیر نمایش داده شده است.



FATEK



Moeller

Keyence

شکل ۱-۴۸ مدل‌های مختلف PLC برخی از سازندگان دیگر

۱-۶ پرسش و تحقیق

۱- ویژگی‌های PLCهای ساخت سازندگان مختلف را با یکدیگر مقایسه کنید.
۲- سیستم اسکادای به‌کار رفته در پست برق چگونه است؟
۳- چگونه چند لوپ کنترلی را می‌توان روی یک سیستم FCS پیاده‌سازی کرد؟

۱-۷ تست‌های خودآزمایی

۱- اولین سیستم کنترلی که در صنعت به‌کار گرفته شد کدام سیستم می‌باشد؟
الف) مکانیکی ب) نیوماتیکی ج) رله‌کنتاکتوری د) الکترونیکی
۲- DCS جایگزین کدام‌یک از سیستم‌های زیر گردید؟
الف) رله کنتاکتوری ب) PLC ج) DDC د) هر سه مورد
۳- مزایای استفاده از شبکه‌های صنعتی به‌جای کابل‌کشی‌های موازی چیست؟
الف) حذف نویز ب) کاهش سیم‌کشی ج) کاهش هزینه د) هر سه مورد
۴- در کدام‌یک از سیستم‌های زیر امکان کنترل لوپ در وسیله نصب شده در فیلد وجود دارد؟
الف) PLC ب) DCS ج)FCS د) DDC
۵- هدف از ارائه‌ی OPC چه بود؟
الف) استانداردسازی PLC ب) یکپارچه‌سازی مانیتورینگ
ج) استاندارد سازی DCS د) هر سه مورد
۶- حوزه‌ی کاری SCADA کدامیک از موارد زیر است؟
الف) سیستم‌های آبرسانی ب) سیستم‌های گازرسانی ج) پست‌های برق د) هر سه مورد
۷- در سیستم PC Based ارتباط بین CPU و تجهیزات سطح فیلد چگونه برقرار می‌گردد؟
الف) سیم‌کشی مستقیم ب) توسط ترانسمیترها ج) شبکه‌ی صنعتی د) هیچ‌کدام
۸- در پروسه‌هایی که سیگنال آنالوگ زیادی وجود دارد استفاده از سیستم . . . . راه‌حل مناسبی است.
الف) PLC ب)DDC ج)DCS د) رله‌کنتاکتوری
۹- در صورت استفاده از DCS به‌جای PLC کدام اشکال در سیستم به‌وجود می‌آید؟
الف) امکانات مانیتورینگ از دست می‌رود. ب) پاسخ سیستم کند می‌شود.
ج) در صورت اشکال در کنترل‌کننده مرکزی کل سیستم متوقف می‌شود. د) هر سه مورد
۱۰- روش برنامه‌نویسی . . . استفاده از زبان‌های سطح بالا و . . . استفاده از زبان‌های سطح پایین است.
الف) DCS – PLC ب) PLC – DCS ج) DDC – PLC د)PLC -DDC

فصل ۲

اجزای PLC و عملکرد آنها

۲-۱ مقدمه

۲-۲ ماژول تغذیه

۲-۲-۱ عملکرد منبع تغذیه

۲-۲-۲ ویژگی‌های منبع تغذیه

۲-۳ ماژول پردازشگر (CPU)

۲-۳-۱ ساختار کلی CPU

۲-۳-۲ حافظه CPU

۲-۳-۳ عملکرد CPU

۲-۴ ماژول ورودی دیجیتال

۲-۴-۱ عملکرد

۲-۴-۲ بخش‌های داخلی ماژول

۲-۴-۳ ماژول دیجیتال با ورودی AC

۲-۴-۴ ماژول دیجیتال با ورودی DC

۲-۴-۵ ورودی‌های DC از نوع Sink /Source

۲-۵ ماژول خروجی دیجیتال

۲-۵-۱ عملکرد

۲-۵-۲ بخش‌های داخلی ماژول

۲-۵-۳ خروجی رله‌ای

۲-۵-۴ خروجی‌های ترانزیستوری

۲-۵-۵ خروجی Triac (خروجی AC)

۲-۵-۶ خروجی Sink / Source

۲-۶ ماژول ورودی آنالوگ

۲-۷ ماژول خروجی آنالوگ

۲-۸ پارامترهای کارت‌های ورودی و خروجی

۲-۹ ماژول های مربوط به I/O های خاص

۲-۱۰ ماژول ارتباطاتی Communication

۲-۱۱ سایر اجزای PLC

۲-۱۱-۱ ماژول‌های توسعه

۲-۱۱-۲ وسیله‌برنامه‌ریزی (Programming Device)

۲-۱۲ پرسش و تحقیق

۲-۱۳ تست‌های خودآزمایی

در این فصل اجزای PLC و عملکرد آنها به‌طور اجمال و در حد نیاز شرح داده شده است. برای درک مباحث فصل‌های بعد مطالعه این بخش ضروری است.

فصل ۲

چکیده مطالب

- اجزای اصلی PLC عبارتند از: ماژول تغذیه، CPU، ماژول I/Oهای معمولی، ماژول I/Oهای خاص، ماژول شبکه
- اجزای PLC می‌تواند به‌صورت یکپارچه[1] یا به‌صورت مجزا[2] باشد.
- ماژول تغذیه باید بتواند علاوه بر تثبیت ولتاژ نقش ایزوله‌سازی و حفاظت را نیز ایفا کند.
- خروجی ماژول تغذیه 24VDC است ولی ورودی آن می‌تواند 110/220 AC یا 24DC باشد.
- CPU میکروپروسسور کوچکی است که نقش اصلی آن اجرای برنامه و مدیریت اجزای داخلی از طریق باس‌های داخلی است.
- در CPU چهار نوع باس وجود دارد: Data Bus , Control Bus , I/O Bus , Address Bus.
- CPU دارای Clock داخلی است که فرکانس آن تعیین کننده سرعت پردازش است.
- CPU دارای حافظه داخلی ROM است که سیستم عامل روی آن ذخیره شده است.
- CPU به‌صورت داخلی دارای حافظه RAM است که برنامه و دیتا روی آن ذخیره می‌شود. این حافظه را می توان با کارت بیرونی افزایش داد. حافظه بیرونی می‌تواند از جنس RAM یا Flash باشد.
- ماژول‌های ورودی و خروجی دیجیتال دارای انواع AC و DC است.
- در کارت‌های دیجیتالDC با توجه به مسیر جریان بین سنسور و کانال می‌توان به‌صورت Sink یا Source ارتباط برقرار کرد.
- خروجی دیجیتال می‌تواند به‌صورت رله‌ای یا ترانزیستوری باشد که رله‌ای برای تامین جریان بالاتر و ترانزیستوری برای سرعت بالاتر مناسب است.
- سیگنال‌های آنالوگ ورودی می‌توانند به‌صورت ولتاژی ،میلی‌ولتی ، جریانی یا اهمی باشند. در کارت ورودی آنالوگ از این سیگنال‌ها نمونه‌برداری شده سپس توسط مبدل A/D به دیتای دیجیتال تبدیل و در اختیار CPU قرار می‌گیرند.
- سیگنال‌های آنالوگ خروجی می‌توانند به‌صورت ولتاژی یا جریانی باشند. دیتای CPU با ارسال به کارت توسط مبدل D/A به سیگنال ولتاژ یا جریان تبدیل و به خروجی ارسال می‌گردد.
- PLC بایستی امکان ارتباط با ورودی و خروجی‌های خاص که از طریق کارت‌های معمولی قابل استفاده نیستند مانند پالس‌های سریع را داشته باشد. برای این منظور از I/O های خاص استفاده می‌گردد.
- PLC برای ارتباط با وسیله (Programming) و سایر ارتباطات بیرونی بایستی مجهز به پورت‌های ارتباط سریال بوده و ماژول‌های شبکه را نیز پشتیبانی کند.

1. Comapct
2. Modular

اصطلاحات و تعاریف

Discrete
به اصطلاحات و تعاریف فصل ۱ مراجعه شود.

Analog
به اصطلاحات و تعاریف فصل ۱ مراجعه شود.

Multiplexer
مالتی پلکسر وسیله‌ای است که امکان ارتباط انتخابی ورودی‌های متفاوت را با خروجی، متناسب با سیگنال پایه کنترل مهیا می‌نماید. به‌عبارت دیگر متناسب با سیگنالی که به پایه کنترل آن داده می‌شود، در هر لحظه یکی از ورودی‌ها را به خروجی منتقل می‌کند.

Data
منظور از دیتا، رشته عددی درفرمت باینری (صفر و یک) می باشد که برای دیتاهای دیجیتال طول آن به اندازه یک بیت و برای دیتاهای آنالوگ به اندازه ۱۶ بیت می‌باشد. در این حالت کلیه‌ی اطلاعات و مقادیر با رشته‌ای از صفر و یک منطقی نشان داده می‌شود. معمولاً در سیستم‌های کامپیوتری فقط امکان پردازش دیتا وجود دارد و در صورتی‌که نیاز به پردازش سیگنال الکتریکی وجود داشته باشد از کارت‌های مبدل سیگنال الکتریکی به دیتا استفاده می‌شود. به این کارت‌ها کارت‌های ورودی گفته می‌شود.

Switching Power Supply
منبع تغذیه سوئیچینگ کنترل سطح ولتاژ خروجی را از طریق روشن و خاموش شدن ترانزیستور قدرت انجام می دهد.

Rack
رک وسیله‌ای است که اجزای PLC روی آن نصب می‌شوند. برخی از انواع رک علاوه بر نگه داشتن ماژول‌ها ارتباط بین آنها را نیز از طریق باس برقرار می‌کنند.

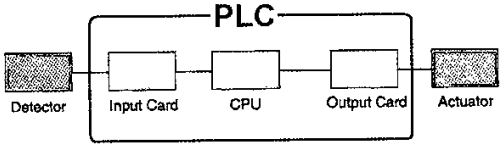
ایزوله‌سازی
منظور ایزوله‌سازی جدا کردن دو مدار الکتریکی به‌صورتی است که بروز شرایط غیرنرمال و خطر آفرین در یک مدار دیگری را تحت تاثیر قرار ندهد.

Optocoupler
کوپل نوری بین دو مدار الکتریکی به‌صورتی که در یک مدار ولتاژ به نور تبدیل شده (توسط LED) و این نور فتو ترانزیستوری را در مدار دوم به‌کار می اندازد .

فصل ۲

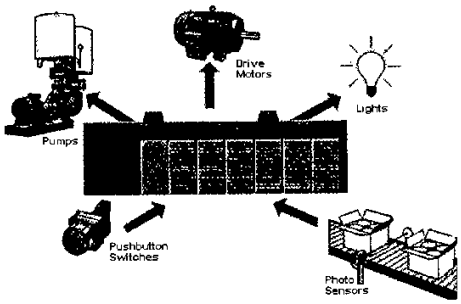
۲-۱ مقدمه

با مطالبی که به‌صورت مختصر در فصل قبل آورده شد، خواننده با یک برداشت کلی از PLC مواجه شده است. ذهنیتی که تا اینجا ایجاد گردیده قاعدتاً این است که PLC دارای سه قسمت اصلی شامل پردازشگر[1]، بخش ورودی[2] و بخش خروجی[3] می‌باشد.



شکل ۲-۱ سه بخش اصلی یک PLC

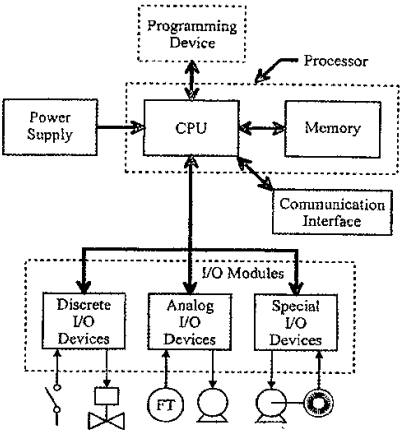
وظیفه‌ی CPU پردازش دیتاهای مربوط به فرآیند، مطابق برنامه‌ی نوشته شده در آن و صادر نمودن فرامین لازم جهت تجهیزات تحت کنترل می‌باشد. اطلاعات فرآیند معمولاً به‌صورت سیگنال الکتریکی از سنسورها و سوئیچها و ترانسمیترهای مختلف دریافت می‌گردد. از آنجایی که برای CPU فقط امکان پردازش اطلاعات به‌صورت دیتا وجود دارد، بنابراین باید سیگنال الکتریکی دریافت شده از سنسورها و سوئیچها به دیتا تبدیل شده و در اختیار CPU قرار گیرد که این عمل توسط کارت ورودی انجام می‌پذیرد. پس از پردازش برنامه توسط CPU، نیاز به ارسال فرمان‌های لازم به ادوات تحت کنترل می‌باشد. فرمان‌های تولید شده توسط CPU به‌صورت دیتا می‌باشد اما تجهیزات تحت کنترل معمولاً با سیگنال‌های الکتریکی تغذیه و راه‌اندازی می‌شوند. بنابراین لازم است که دیتاهای ایجاد شده توسط CPU ابتدا به سیگنال الکتریکی مناسب تبدیل شده و سپس به تجهیزات تحت کنترل ارسال شود که این کار توسط کارت‌های خروجی انجام می‌پذیرد.



شکل ۲-۲ ارتباط PLC با ورودی و خروجی‌ها

1. CPU
2. Input
3. Output

این ذهنیت اگر چه صحیح است ولی کامل نیست. یک PLC علاوه بر موارد فوق، دارای قسمت‌های اصلی دیگری نیز هست. در این فصل هدف آن است که اجزای اصلی PLC مورد بحث قرار گیرد، ولی جزئیات هر کدام از بخش‌ها مد نظر نیست. برای برخی از اجزا با کمی تفصیل بیشتر و برای برخی دیگر این موضوع در حد اشاره‌ای گذرا خواهد بود.
به‌طور کلی می‌توان معماری نشان داده شده در شکل ۲-۳ را به‌عنوان ساختار پایه یک سیستم PLC ارائه نمود.



شکل ۲-۳ معماری PLC

همانطور که دیده می شود، اجزای اصلی PLC عبارتند از:

- پردازشگر (CPU)
- حافظه[1]
- بخش تغذیه[2]
- بخش ارتباطات[3]
- بخش ارتباط با وسیله برنامه‌ریزی[4]– که می‌تواند در زیر مجموعه بخش ارتباطات نیز جای بگیرد.
- بخش ورودی و خروجی‌های Discrete (دیجیتال)
- بخش ورودی و خروجی‌های آنالوگ
- بخش ورودی و خروجی‌های خاص (به‌عنوان مثال برای ارتباط با سنسورهای سریع)

الزامی وجود ندارد که اجزای فوق به‌صورت مستقل و مجزا باشند تا نیاز به مونتاژ آنها وجود داشته باشد. PLC می‌تواند با ساختاری یکپارچه که در اصطلاح Compact نامیده می‌شود یا ساختاری با اجزای مستقل که اصطلاحاً Modular خوانده می‌شود ارائه شود .

1. Memory
2. Power Supply
3. Communication
4. Programming

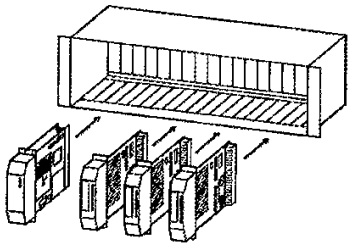
در نوع ماژولار بخش‌های فوق به‌صورت کاملا مجزا هستند و به یکدیگر به‌صورت دستی یا از طریق Rack متصل می‌شوند.

در نوع یکپارچه بخشی یا همه اجزا به‌صورت یکپارچه هستند و PLC به یکی از حالات زیر در دسترس است:

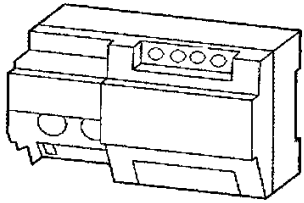
۱- CPU با بخش‌های ورودی و خروجی به‌صورت یکپارچه بوده ولی بخش تغذیه و ارتباطات مجزاست.

۲- CPU با بخش‌های ورودی و خروجی و تغذیه به‌صورت یکپارچه بوده ولی بخش ارتباطات مجزاست.

۳- CPU همراه با تمام بخش‌های دیگر به‌صورت یکپارچه می‌باشد.



Modular PLC



Compact PLC

شکل ۲-۴ PLC نوع یکپارچه و نوع Modular

در این فصل محور بحث بر اساس اجزای ماژولار است و هر کدام از بخش‌های اصلی PLC به‌طور مستقل مورد بحث قرار می‌گیرد ولی این بحث تناقضی با نوع Compact ندارد. در نوع یکپارچه می‌توان تصور کرد که در یک ساختار یکپارچه بخش‌های مختلف تفکیک شده و مورد بررسی قرار می‌گیرد.

۲-۲ ماژول تغذیه

منبع تغذیه معمولاً به‌عنوان اولین ماژول PLCهای Modular تلقی می‌شود. عملکرد و ویژگی‌های آن در این قسمت مورد بحث قرار می‌گیرد.

۲-۲-۱ عملکرد منبع تغذیه

وظیفه‌ی آن تأمین ولتاژ و جریان مورد نیاز CPU و سایر اجزای PLC می‌باشد. منبع تغذیه نه تنها وظیفه‌ی تغذیه تثبیت شده پردازشگر، حافظه و سایر ماژول‌های ورودی و خروجی و ارتباطات را به‌عهده دارد، بلکه باید بتواند تجهیزات PLC را در مقابل سایر اجزای سیستم (شبکه برق) محافظت کند.

۲-۲-۲ ویژگی‌های منبع تغذیه

از نظر فیزیکی منبع تغذیه می‌تواند به‌صورت مجزا یا به‌صورت یکپارچه با سایر اجزا باشد. در نوع یکپارچه فقط کافیست منبع تغذیه را به ولتاژ بیرونی متصل کنیم. نیازی به هیچگونه سیم‌کشی یا اتصال از منبع تغذیه به نقاط دیگر نیست؛ زیرا این اتصال بصورت داخلی انجام شده است. در نوع ماژولار ممکن است با یکی از دو حالت زیر مواجه شویم:

- منبع تغذیه همراه با سایر اجزا توسط یک باس ارتباط می‌یابد. در این حالت خروجی منبع تغذیه توسط باس به سایر اجزا منتقل می‌گردد.
- بین منبع تغذیه و سایر اجزا هیچ باس ارتباطی وجود ندارد. در این حالت برای اتصال خروجی منبع تغذیه به سایر اجزا سیم‌کشی بین آنها ضروری خواهد بود.

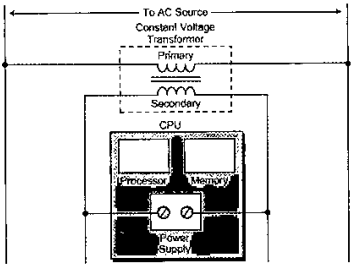
از نظر الکتریکی مهمترین ویژگی میزان جریان خروجی آن است و از این جنبه دارای انواع مختلف است مانند نمونه‌های 2A , 5A , 10 A , 20 A و ... .

منبع تغذیه‌های PLC معمولاً از نوع Switching Power Supply هستند. ولتاژ ورودی منبع تغذیه می‌تواند 24VDC , 120 VAC , 220VAC باشد. نوع AC متداول‌تر و پرکاربردتر از نوع DC است. منبع تغذیه با ولتاژ ورودی DC نیز در برخی موارد کاربرد دارد مانند:

- در شرایطی که فقط ولتاژ DC برای سیستم کنترل در دسترس باشد.
- در شرایطی که برای اطمینان بالاتر از تغذیه‌های مطمئن DC مانند باتری‌شارژرها که در شرایط قطع تغذیه AC باز برقرار هستند، استفاده شود.

در صورت بروز نوسان در سطح ولتاژ یا فرکانس ورودی، منبع تغذیه باید بتواند با تولرانس قابل قبول به تغذیه PLC ادامه دهد. به‌عنوان مثال اگر ورودی 220VAC به اندازه 10%-/+ تغییر کند یعنی بین 198-242 VAC باشد، لازم است منبع تغذیه بدون هیچ مشکلی بتواند ولتاژ و جریان مورد نیاز اجزا را تامین کند. اگر ولتاژ ورودی در چند سیکل بیش از تولرانس مجاز تغییر کند، بیشتر منابع تغذیه فرمان Shutdown را به CPU ارسال می‌نمایند. در برخی فرآیندها توقف با خسارات زیاد همراه است به همین علت تمهیدات مختلفی را برای ایجاد یک تغذیه مطمئن در نظر می‌گیرند. در کاربردهای حساس استفاده از دو منبع تغذیه که اصطلاحاً افزونه[1] خوانده می‌شوند توصیه می‌گردد . این دو منبع تغذیه به‌طور همزمان روی باس حضور دارند، در عین حال برای جلوگیری از جریان گردشی بین آنها تجهیزات لازم (مانند دیود) روی باس یا در خود منبع تغذیه در نظر گرفته شده است.

اما نباید با تمرکز روی خروجی منبع تغذیه از ورودی آن غافل شد. اگر ورودی AC با تغییرات و نوسانات زیاد همراه باشد یا هارمونیک را منتقل کند، تغذیه نامناسبی به منبع تغذیه عرضه خواهد شد. برای رفع این اشکال یک روش استفاده از ترانسفورماتور با خروجی ولتاژ ثابت است که به CVT موسوم است. این ترانس در صورت بروز تغییرات زیاد ولتاژ در ورودی که می‌تواند ناشی از راه‌اندازی یا توقف تجهیزات سنگین مانند موتورهای بزرگ باشد، ولتاژ خروجی را ثابت نگه می‌دارد. به‌عنوان مثال با 15% تغییر ولتاژ در ورودی، خروجی در حد 1% تغییر می‌کند.



شکل ۲-۵ ترانس تثبیت کننده ولتاژ CVT

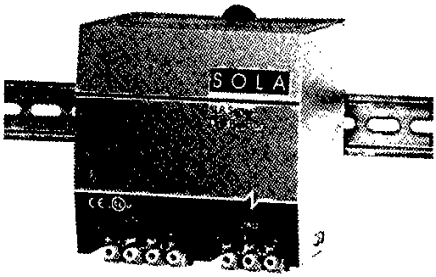
1. Redundant

فصل ۲

CVT اگرچه تثبیت کننده ولتاژ است ولی در صورت وجود هارمونیک‌های زیاد در موج ولتاژ ورودی، آنها را به خروجی منتقل می‌کند. برای رفع این مشکل از ترانس‌های تثبیت‌کننده Sola CVS[1] که می‌توانند موج سینوسی مناسب در خروجی تحویل دهند استفاده می‌شود.

امروزه همه نیازهای فوق توسط برخی منابع تغذیه فراهم شده است و نیازی به تهیه ترانس مجزا نیست بلکه با تهیه یک منبع تغذیه مناسب می‌توان به یک تغذیه مطمئن دست یافت و اهداف زیر برآورده خواهد شد:

- تثبیت ولتاژ[2] در صورت نوسان ولتاژ خط
- حذف هارمونیک‌های خط
- ایزوله‌سازی
- حفاظت در مقابل اضافه ولتاژهای گذرا



شکل ۲-۶ منبع تغذیه CVS

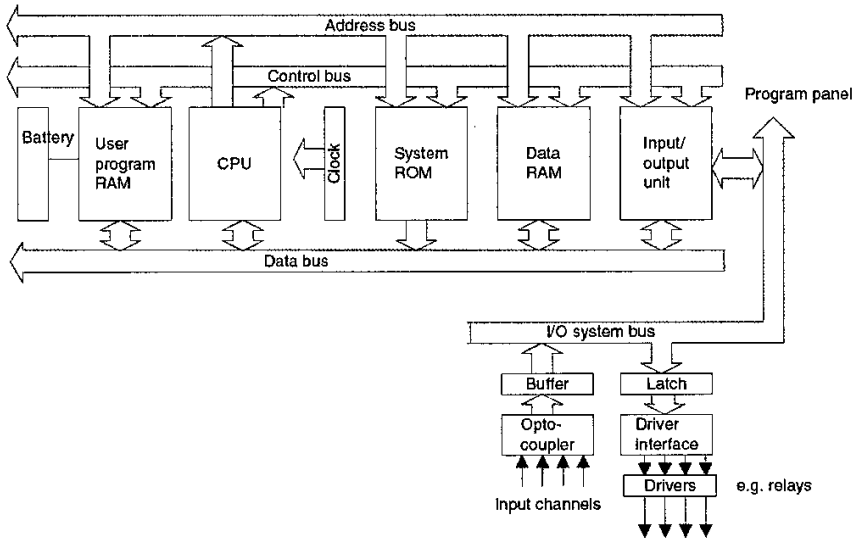
۲-۳ ماژول پردازشگر (CPU)

دریک PLC، ماژول CPU از میکروپروسسور کوچکی تشکیل شده که به‌طور خاص برای کاربرد صنعتی طراحی گردیده است. ممکن است PLC دارای بیش از یک میکروپروسسور باشد که به این سیستم Multiprocessing گفته می‌شود. در این سیستم معمولاً یک میکروپروسسور کار کنترل اصلی فرآیند را انجام می‌دهد و میکروپروسسور دیگر برای اجرای توابع پیچیده کنترلی به‌کار گرفته می‌شود که نتایج پردازش آن به میکروپروسسور اول ارسال می‌گردد.

۲-۳-۱ ساختار کلی CPU

سخت‌افزار CPU دارای بخش‌های مختلفی است. شکل ۲-۷ شماتیک اجزا و ارتباطات داخلی CPU را نشان می‌دهد.

1. Constant Voltage Source
2. Constant Voltage Transformer



شکل ۲-۷ شماتیک اجزا و ارتباطات داخل CPU

همانطور که در این ساختار شماتیک دیده می شود، CPU علاوه بر اجرای برنامه کنترل مدیریت باس‌های ارتباطی که با ورودی و خروجی‌ها مرتبط هستند یا باس‌هایی که مرتبط با بخش‌های حافظه هستند را نیز به عهده دارد.

CPU مجهز به Clock است که فرکانس آن معمولاً بین 1 تا 8 مگا هرتز می‌باشد. این Clock معرف سرعت پردازش CPU می‌باشد، بعلاوه برای سنکرون‌سازی (همزمان‌سازی) تمام عناصر داخلی PLC به‌کار می‌رود.

فاکتور دیگری که CPU بر اساس آن دسته‌بندی می‌شود تعداد بیت قابل پردازش است که از این جهت به ۸ بیتی، ۱۶ بیتی و ۳۲ بیتی تقسیم‌بندی می‌شود. مسلماً هرچه تعداد بیت قابل پردازش بیشتر باشد سرعت CPU بالاتر است.

CPU برای ارتباطات خود از مسیرهای داخلی استفاده می‌کند که به آنها Bus گفته می‌شود. در CPU چهار نوع باس وجود دارد:

- Data BUS : که برای تبادل دیتا بین اجزای مختلف به‌کار می‌رود.
- Address BUS :که برای ارسال آدرس محل دسترسی به دیتای حافظه استفاده می‌شود.
- Control BUS: برای ارسال سیگنال‌هایی که برای کنترل داخلی هستند (مانند فرمان سنکرون‌سازی) به‌کار می‌رود.
- I/O BUS : برای ارتباط بین ماژول ورودی و خروجی با پورت‌های I/O روی CPU استفاده می‌شود.

۲-۳-۲ حافظه CPU

حافظه CPU را می‌توان به دو بخش تقسیم نمود: حافظه داخلی و حافظه بیرونی.

حافظه داخلی

این حافظه همانطور که در شکل۲-۷ دیده می شود، شامل موارد زیر است:

- **System ROM**:[1] این حافظه فقط خواندنی است و قابل نوشتن و پاک شدن نمی‌باشد. سیستم عامل CPU در این حافظه ذخیره شده است.
- **RAM**:[2] این حافظه قابل خواندن و نوشتن است و سرعت دسترسی به آن بالاست. در شکل ۲-۷ دیده می‌شود که دو حافظه RAM در ساختار CPU وجود دارد. برنامه کاربر معمولاً در حافظه RAM می‌نشینند و دیتاها نیز در حافظه RAM ذخیره می‌گردد. بدیهی است این دو نوع اطلاعات بایستی بتوانند به‌سهولت قابل جایگزینی و پاک شدن باشند و در عین حال سرعت دسترسی به آنها بالا باشد به همین علت از نوع RAM برای آنها استفاده شده است. حافظه RAM در صورت قطع شدن تغذیه CPU، پاک می‌شود مگر آنکه باتری پشتیبان که در شکل فوق نشان داده شده است، وجود داشته باشد.

حافظه بیرونی

حافظه بیرونی معمولاً برای اهداف نشان داده شده در جدول ۲-۱ استفاده می‌شود.

جدول ۲-۱ اهداف استفاده از حافظه بیرونی CPU

هدف از به‌کارگیری حافظه بیرونی	نوع حافظه مورد استفاده
امکان ذخیره‌سازی حجم برنامه و دیتای بیشتر	RAM یا EPROM[3] یا EEPROM[4]
امکان حفظ برنامه در شرایط قطع تغذیه	EPROM یا EEPROM

در PLC های کوچک حافظه RAM داخلی قابل افزایش نیست. معمولاً در PLCهایی که کمتر از I/O 64 را پشتیبانی می‌کنند امکان افزایش حافظه وجود ندارد. (عملاً نیازی نیز به این افزایش وجود ندارد و برنامه کنترل و دیتا های مورد نیاز برای یک سیستم کوچک روی همین حافظه قابل ذخیره‌سازی است.)

در PLCهای متوسط و بزرگ، حافظه CPU را می‌توان به یکی از روش‌های زیر افزایش داد:

- کارت حافظه بیرونی از جنس RAM: که با قطع تغذیه پاک می‌شود. این کارت صرفاً امکان ذخیره سازی حجم بیشتر برنامه را فراهم می‌سازد.
- کارت حافظه بیرونی از جنس EPROM با قطع تغذیه پاک نمی‌شود. در حال حاضر استفاده از EPROM و EEPROM قدیمی منسوخ شده و نوع جدید Flash EPROM متداول گردیده است. در نوع جدید نوشتن و پاک کردن دیتا روی کارت حافظه در شرایطی که کارت روی CPU قرار دارد نیز امکان‌پذیر است. در انواع قدیمی این کار فقط توسط دستگاه برنامه‌ریزی ممکن بود.

جدول ۲-۲ انواع حافظه را به اختصار معرفی می‌نماید.

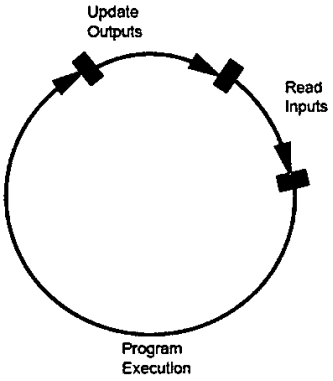
1. Read Only Memory
2. Random Access Memory
3. Erasable Programmable ROM
4. Electrically Erasable Programmable ROM

جدول ۲-۲ انواع حافظه در یک نگاه

نوع حافظه	کاربرد	عملکرد
RAM	حافظه داخلی یا حافظه بیرونی	نوعی از حافظه که دسترسی به آن سریع است ولی درصورت قطع شدن تغذیه آن کاملاً پاک می شود؛ از اینرو به آن volatile memory می‌گویند.
ROM	حافظه داخلی	حافظه فقط خواندنی که قابل پاک شدن نیست. اطلاعات آن در صورت قطع تغذیه پاک نمی‌شود.
EPROM	حافظه بیرونی	این نوع حافظه پس از برنامه‌ریزی شبیه ROM عمل خواهد کرد و با قطع تغذیه پاک نمی‌شود. تفاوت آن با ROM این است که با نور (UV) ultraviolet قابل پاک شدن و دوباره برنامه‌ریزی است. این کار چندین دقیقه زمان می‌برد.
EEPROM	حافظه بیرونی	این نوع حافظه مشابه EPROM است با این تفاوت که پاک کردن و دوباره برنامه-ریزی کردن آن با ولتاژ الکتریکی انجام می‌شود؛ از اینرو کار با آن ساده‌تر از EPROM است. تعداد دفعات پاک کردن و نوشتن روی آن اگرچه زیاد ولی محدود است. EEPROM به E2PROM نیز موسوم است. در نوع مدرن EEPROM می‌تواند به‌صورت Online در حالی که به CPU متصل است برنامه‌ریزی شود و نیازی به استفاده از وسیله برنامه‌ریزی بیرونی نیست. این نوع به Flash EPROM موسوم است.

۲-۳-۳ عملکرد CPU

در PLC، وظیفه CPU اجرای تکراری و مداوم یک سیکل‌کاری است که اصطلاحاً سیکل اسکن[1] خوانده می‌شود. در این سیکل، CPU اطلاعات ورودی را خوانده، برنامه را اجرا کرده و فرمان لازم را به خروجی‌ها ارسال می‌نماید.

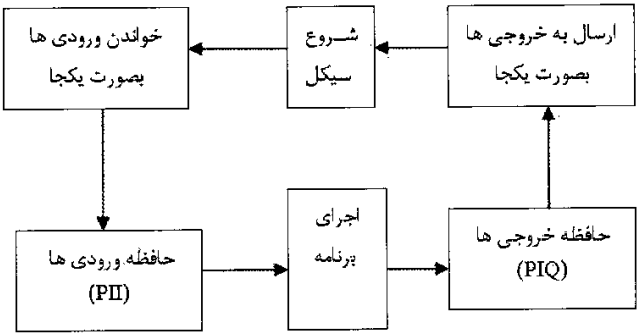


شکل ۲-۸ رفتار CPU در سیکل اسکن

1. Scan Cycle

ابتدا دیتاهای مربوط به سیگنال‌های ورودی که در ناحیه Data RAM قرار دارد خوانده شده و برنامه کاربری توسط CPU از Program RAM خوانده و اجرا می‌شود. فرامین ایجاد شده برای خروجی‌ها از ناحیه خروجی‌های Data RAM به کانال‌های خروجی ارسال می‌گردد. این کار دائماً به‌صورت سیکلی تکرار می گردد. همانطور که در شکل ۲-۷ صفحات قبل مشاهده می‌شود، وضعیت خروجی‌ها در طول سیکل Latch یعنی حفظ می شود تا اینکه فرمان جدید سیکل بعدی آن را update کند.

نکته‌ی قابل توجه آن است که ورودی‌ها به‌طور یکجا خوانده شده و در حافظه ورودی‌ها جای می‌گیرند و خروجی‌ها نیز به‌طور یکجا از حافظه ارسال می‌شوند. این عملکرد باعث می‌شود که تاخیر خواندن ورودی و نوشتن خروجی‌ها به حداقل برسد. اگر قرار بود که این کار بیت به بیت انجام گردد برای PLCهای بزرگ که با هزاران I/O سروکار دارند تاخیر بسیار زیادی بوجود می آمد.

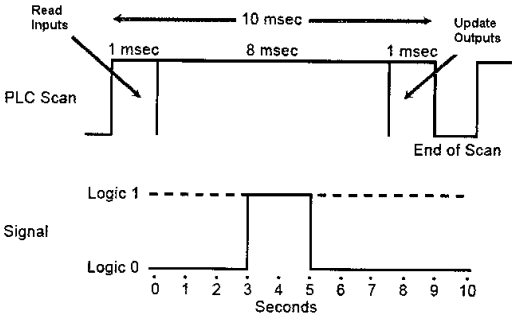


شکل ۲-۹ رفتار CPU در خواندن ورودی‌ها و ارسال فرمان به خروجی‌ها

زمان سیکل اسکن به حجم برنامه و دستورات به‌کار رفته بستگی دارد ولی به‌طور معمول از ۱۰ میلی‌ثانیه تا ۱۰۰ میلی‌ثانیه طول می‌کشد. علاوه بر برنامه، موارد دیگری نیز روی زمان سیکل‌اسکن تاثیر دارد که از جمله آنها می‌توان به ارتباط Online با وسیله برنامه‌ریزی اشاره کرد. در این شرایط CPU همزمان با اجرای برنامه بایستی اطلاعات را در اختیار وسیله مانیتورینگ بیرونی نیز قرار دهد که منجر به افزایش سیکل خواهد شد.

بر این اساس CPU قادر است ورودی‌هایی که زمان تغییر آنها کمتر از یک سیکل اسکن باشد را بخواند و به خروجی‌ها با تاخیر حداقل یک سیکل فرمان دهد.

شکل ۲-۱۰ حالتی را نشان می‌دهد که یک سیگنال ورودی سریعاً و در کمتر از یک سیکل تغییر می‌کند ولی CPU نمی‌تواند آنرا ببیند.



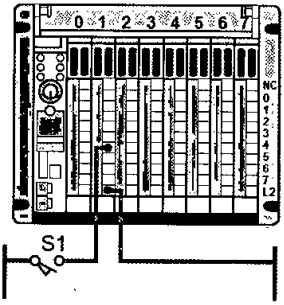
شکل ۲-۱۰ وضعیتی که تغییرات سیگنال سریعتر از سیکل اسکن است.

۲-۴ ماژول ورودی دیجیتال

۲-۴-۱ عملکرد

CPU در PLC با ولتاژهای پایین مانند 5 v که برای TTL [1] است کار می‌کند. دیتاهایی که در CPU آنالیز می‌شوند در همین سطح ولتاژ قرار دارند. ماژول ورودی سیگنال‌های دیجیتال را دریافت نموده و آنها را به دیتا تبدیل و در اختیار CPU قرار می‌دهد.

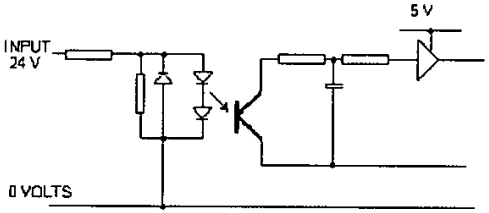
سیگنال دیجیتال یا Discrete سیگنال غیر پیوسته‌ای است که فقط دو وضعیت On و Off (قطع و وصل) دارد. سنسورها، سوئیچ‌ها و شستی‌های اپراتوری به‌عنوان ورودی دیجیتال به این ماژول متصل می‌گردند.



شکل ۲-۱۱ ورودی دیجیتال PLC

1. Transistor – Transistor Logic

ترمینال‌های این ماژول از بخش الکترونیکی سیستم به‌صورت اپتیک (نوری) ایزوله می‌شوند تا در صورت بروز پالس‌های ولتاژی گذرا یا اتصالی در سمت ورودی، مدار الکترونیکی آسیب نبیند.



شکل ۲-۱۲ ایزوله‌سازی ترمینال کانال ورودی PLC

۲-۴-۲ بخش‌های داخلی ماژول

در نگاه دقیق‌تر بخش ورودی دارای دو قسمت اصلی است:

Power Section: سیگنال دریافتی از فیلد به این بخش متصل می‌گردد که بسته به نوع کارت و نوع سنسور دارای سطح ولتاژهای متفاوتی می‌تواند باشد. متداول‌ترین سطح ولتاژ ۲۴ ولت DC است که به این بخش متصل می‌گردد.

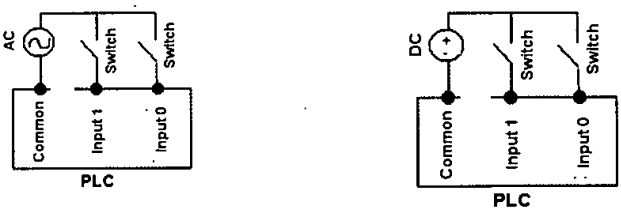
Logic Section : در این بخش سیگنال دریافتی به 5V DC تبدیل شده و به پردازشگر ارسال می‌گردد.

دو بخش فوق توسط Optocoupler از یکدیگر ایزوله شده‌اند.

ورودی‌های دیجیتال از نظر ولتاژ ورودی متنوع هستند و مهم‌ترین آنها در جدول ۲-۳ آمده است. همانطور که مشاهده می‌شود جنس ورودی دیجیتال می‌تواند AC یا DC باشد.

جدول ۲-۳ انواع سیگنال دیجیتال ورودی

نوع سیگنال	سطح سیگنال
DC	• 12-24 Vdc • 10-60 Vdc • 48 Vdc • 5 Vdc (TTL)
AC	• 100-120 Vac • 12-24 Vac • 200-240 Vac • 24 Vac



ورودی دیجیتال از نوع AC

ورودی دیجیتال از نوع DC

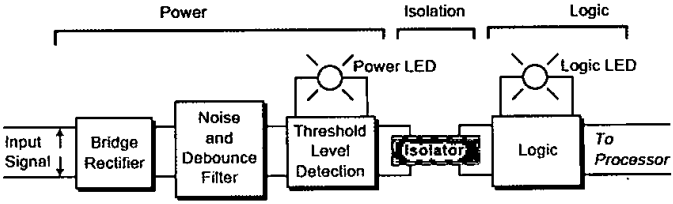
شکل ۲-۱۳ شماتیک اتصال ورودی‌های دیجیتال AC و DC

نکات قابل توجه

- ورودی‌ها به‌طور معمول High Impedance هستند؛ به‌عبارت دیگر جریان بسیار اندکی مصرف می‌کنند.
- ولتاژهای DC معمولاً سطح کمتری نسبت به ولتاژ AC دارند مانند ۱۲ و ۲۴ ولت.
- ولتاژهای DC نسبت به ولتاژهای AC از ایمنی بیشتری برخوردار می‌باشند.
- ورودی‌های DC سریعتر از نوع AC قابل شناسایی هستند. مثلاً در نوع AC برای شناخت یک سیگنال 50HZ به‌صورت قابل قبول نیاز به زمان 20 میلی‌ثانیه(1/50 ثانیه زمان) وجود دارد.
- ورودی و خروجی‌های DC پرکاربردتر از نوع AC هستند.
- در نوع AC نسبت به نوع DC ایمنی بیشتری در مقابل نویز وجود دارد، از اینرو ولتاژ AC برای مسافت‌های زیاد و در محیط‌هایی با نویز مغناطیسی بالا مناسب‌تر است.
- استفاده از ولتاژ AC ساده‌تر و کم هزینه‌تر از نوع DC است.
- در PLC های کوچک که به‌صورت Compact عرضه می‌شوند معمولاً نوع سیگنال ورودی و سطح ولتاژ آن برای تمام کانال‌ها یکسان و معمولاً DC 24 V است.

۲-۴-۳ ماژول دیجیتال با ورودی AC

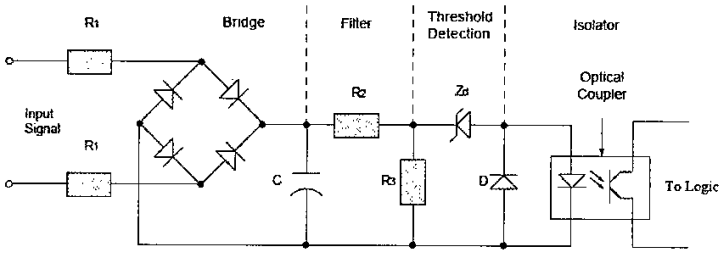
همانطور که در شکل ۲-۱۴ مشاهده می‌شود، سیگنال AC قبل از رسیدن به بخش Logic، در بخش Power از قسمت‌های مختلفی عبور می‌کند.



شکل ۲-۱۴ ورودی AC ماژول دیجیتال

وظیفه بخش Power آن است که سیگنال AC را به DC تبدیل کرده و با فیلتر کردن نویز در صورتی‌که در حد مجاز قرار داشته باشد به بخش ایزوله‌سازی تحویل دهد.

جزئیات بخش Power در شکل ۲-۱۵ آمده است.



شکل ۲-۱۵ بخش Power در ماژول ورودی دیجیتال AC

بخش یکسو ساز: سیگنال AC با ورود به بخش Power Section ابتدا توسط پل دیودی یکسو شده و به DC تبدیل می‌گردد.

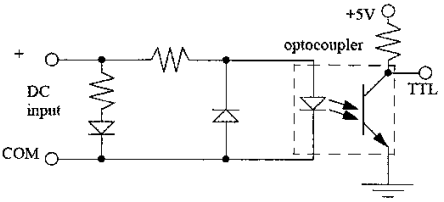
بخش فیلتر: در این بخش، سیگنال یکسو شده فیلتر می‌گردد تا ضمن حفاظت، اثرات نویز روی ورودی AC را حذف کند. این فیلتر تأخیری در حدود 9-25 m sec ایجاد می‌کند.

بخش آشکارساز آستانه: اگر سیگنال دریافتی در طول مدت ذکر شده (تاخیر بخش فیلتر) در حد آستانه مجاز باقی ماند به‌عنوان یک ورودی معتبر شناخته می‌شود.

بخش ایزوله‌سازی: سیگنال معتبر دریافتی توسط Optocoupler به‌صورت نوری ایزوله شده سپس وارد بخش Logic می‌گردد. این ایزوله‌سازی،بخش Logic را از ضربه‌های‌ولتاژی که ممکن است در سمت AC رخ دهد مصون نگه می‌دارد.

۲-۴-۴ ماژول دیجیتال با ورودی DC

برای سیگنال‌های DC در بخش Power نیاز به یکسوساز نیست. این سیگنال پس از حفاظت و ایزوله‌سازی به بخش Logic ارسال می‌شود. برای یک سیگنال معتبر، حدود مشخصی وجود دارد. به‌عنوان مثال برای ورودی 24 VDC بازه ولتاژ می‌تواند بین 3V- الی 30V باشد و در این ناحیه سیگنال معتبر خواهد بود.



شکل ۲-۱۶ ورودی دیجیتال با ورودی DC

۲-۴-۵ ورودی‌های DC از نوع Sink /Source

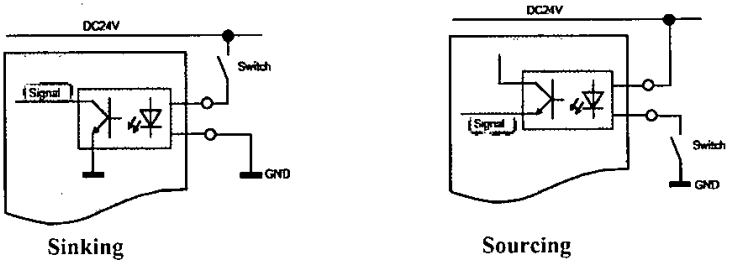
با توجه به مسیر جریان بین سنسور و کانال کارت ورودی DC این کارت به دو نوع زیر تقسیم می‌شود:

- Sinking Input
- Sourcing Input

این تقسیم‌بندی برای نوع AC وجود ندارد. برای شناخت Sink و Source بایستی به مسیر جریان بین سنسور و کارت توجه کرد.

- Sinking Input Module : اگر مسیر جریان از سمت سنسور به کارت باشد، سنسور به‌عنوان Source و کارت به‌عنوان Sink است.
- Sourcing Input Module : اگر مسیر جریان از سمت کارت به سمت سنسور باشد، کارت به‌عنوان Source و سنسور به‌عنوان Sink است.

روش اول متداول‌ترین روش است؛ یعنی سنسور به‌عنوان Source و کارت به‌عنوان Sink است.

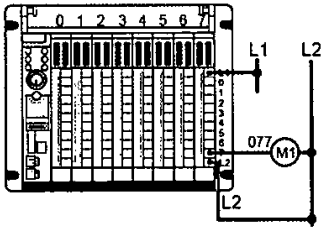


شکل ۲-۱۷ ورودی‌های دیجیتال Sink و Source

۲-۵ ماژول خروجی دیجیتال

۲-۵-۱ عملکرد

دیتای ارسالی توسط CPU، در این ماژول به سیگنال الکتریکی مناسب تبدیل شده و در اختیار ادوات تحت کنترل PLC قرار می‌گیرد. رله‌ها، کنتاکتورها، سلونوئیدها، چراغ‌های سیگنال از جمله وسایل خروجی دیجیتال هستند که به این کارت متصل می‌شوند.



شکل ۲-۱۸ خروجی دیجیتال PLC

انواع ولتاژهای خروجی کارت‌های دیجیتال خروجی در جدول ۲-۵ آمده است.

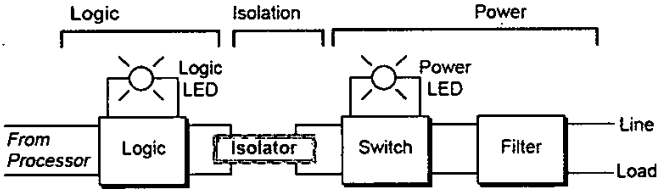
جدول ۲-۵ انواع ولتاژهای خروجی دیجیتال

نوع سیگنال	سطح سیگنال
DC	• 24 Vdc • 12-48 Vdc • 5 Vdc (TTL)
AC	• 120 Vac • 12-48 Vac • 230 Vac

یک ماژول خروجی دیجیتال می‌تواند چندین کانال داشته باشد که از نظر ولتاژ خروجی همگی مشابه هستند. از نظر جریان‌خروجی نیز،جریان‌های‌مختلفی‌توسط‌کارت قابل ارائه می‌باشد که ممکن است در هر کارت با کارت دیگر متفاوت باشد.

۲-۵-۲ بخش‌های داخلی ماژول

این ماژول نیز شبیه نوع ورودی شامل دو بخش Power Section و Logic Section است. PLC فرمان را به‌صورت ولتاژ 5 VDC به بخش Logic می‌فرستد، این ولتاژ پس از ایزوله‌سازی نوری به بخش Power منتقل شده و با تبدیل به سطح ولتاژ بالاتر (نظیر ۲۴ ولت DC) به خروجی منتقل می‌شود.



شکل ۲-۱۹ بخش‌های داخلی خروجی دیجیتال

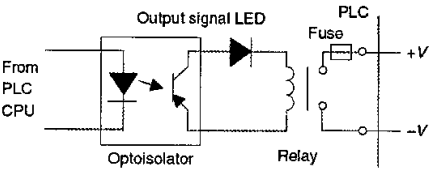
سوئیچینگ (قطع و وصل) این کارت می‌تواند به‌صورت ترانزیستوری یا رله‌ای باشد. از این نظر خروجی به انواع زیر تقسیم بندی می شود:

- **خروجی رله‌ای** که برای DC و AC قابل استفاده است.
- **خروجی ترانزیستوری** که برای DC استفاده می‌شود
- **خروجی Triac** که برای AC استفاده می‌شود.

به نوع رله‌ای Dry Contactc و به نوع Triac و ترانزیستوری Switched Contact می‌گویند.

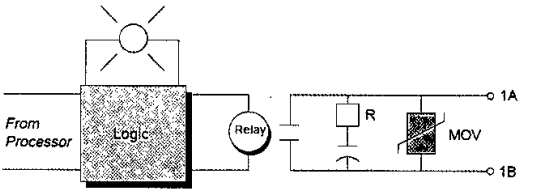
۲-۵-۳ خروجی رله‌ای

در این نوع کارت‌ها از یک رله برای قطع و وصل خروجی استفاده می‌شود. این رله که برای ولتاژ DC یا AC قابل استفاده است، در قطع و وصل خروجی تاخیر ایجاد می‌کند (حداقل ۱۰ میلی‌ثانیه). این رله بتدریج فرسوده می‌شود و پس از تعداد قطع و وصل مشخصی که ممکن است البته چند میلیون بار باشد، نیاز به تعویض دارد. مزیت رله ایزوله‌سازی است و خروجی را از مدار بیرونی جدا می‌سازد. خروجی‌های رله‌ای به Dry Contact موسوم هستند.



شکل ۲-۲۰ خروجی رله‌ای دیجیتال

نوع رله‌ای نیز به بخش حفاظت و فیلترسازی مجهز است که در شکل ۲-۲۱ نشان داده شده است. مدار حفاظتی توسط اسنوبر و نیز وریستور MOV یعنی metal oxide varistor پیش‌بینی شده است که پیک‌های ولتاژی را گرفته و آنها را محدود می‌سازد. بعلاوه، این مدار حفاظتی، نویزهای رسیده از سمت خروجی را دفع می‌کند. برای حفاظت در برابر جریان زیاد، استفاده از فیوز در خروجی توصیه می‌شود.



شکل ۲-۲۱ حفاظت خروجی دیجیتال

مزیت رله ایزوله‌سازی و امکان استفاده برای سیگنال DC و AC است، بعلاوه با انتخاب رله مناسب امکان استفاده از جریان‌های بالا برای مصرف کننده‌های خاص نیز امکان‌پذیر خواهد بود.

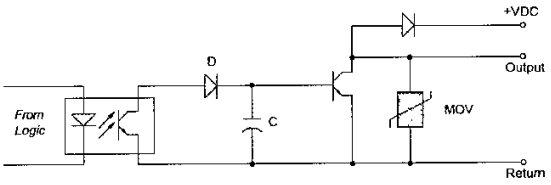
در هنگام استفاده از خروجی رله‌ای به نکات زیر توجه شود:

- هر رله دارای یک ولتاژ و جریان نامی است.
- اگر ولتاژ کمتر از نامی اعمال شود ممکن است خروجی فعال نشود.
- اگر ولتاژ بیش از نامی به رله اعمال شود فرسودگی آن زودتر اتفاق می‌افتد. اگر ولتاژ خیلی زیاد باشد امکان جرقه شدید و سوختن رله وجود دارد.
- اگر جریان بیش از نامی از رله عبور کند آنرا به‌شدت گرم می‌کند و رله زودتر خراب می‌شود. (رله جوش می‌خورد یا ذوب می‌شود.)

۲-۵-۴ خروجی‌های ترانزیستوری

ترانزیستور برای نوع DC استفاده می‌شود. در این نوع، سرعت عملکرد بالا و کمتر از یک میلی‌ثانیه است. جریان خروجی محدود و معمولاً ماکزیمم 1 A می باشد.

خروجی DC، ساختاری شبیه شکل ۲-۲۲ دارد. خروجی Logic پس از ایزوله‌سازی به بخش Power انتقال می‌یابد و پس از تثبیت ولتاژ به Power Transistor داده می‌شود تا آنرا به کانال خروجی سوئیچ کند. حفاظت MOV در اینجا نیز در نظر گرفته شده است و استفاده از فیوز نیز شبیه قبل توصیه می‌گردد.



شکل ۲-۲۲ خروجی DC دیجیتال

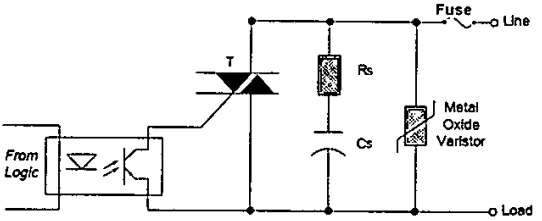
وقتی که مصرف‌کننده یک بار القایی مانند سلونوئید یا موتور باشد، در زمان روشن شدن میدان مغناطیسی ایجاد می‌کند. پس از اینکه خروجی Off شد، میدان مغناطیسی (back EMF) جریان برگشتی به سمت کانال خروجی PLC می‌فرستد. برای حفاظت ترانزیستور از شوک به‌وجود آمده، یک دیود هرزگرد در دو سر مصرف کننده قرار می‌دهند. جهت این دیود به‌صورتی است که در شرایطی که خروجی روشن است از آن جریان نمی‌گذرد.

توجه :اگر به اشتباه به نوع ترانزیستوری ولتاژ AC با همان دامنه DC وصل شود، در زمانی که سیگنال مثبت است (یعنی نصف سیکل) کار می‌کند ولی ممکن است دیده نشود؛ یعنی به‌ظاهر عملکرد درست باشد.

۲-۵-۵ خروجی Triac (خروجی AC)

این خروجی برای نوع AC کاربرد دارد و سرعت آن نسبت به رله‌ای بالاتر است. شکل ۲-۲۳ بلوک‌دیاگرام تبدیل را برای یک کارت خروجی AC نشان می‌دهد. در این کارت، سیگنال دریافتی از بخش Logic پس از ایزوله‌سازی وارد بخش Power می‌شود. این بخش برای AC سازی از Triac یا از SCR [1] برای سوئیچ کردن ولتاژ استفاده می‌کند. پس از آن مدار حفاظتی توسط اسنوبر و نیز وریستور MOV [2] پیش‌بینی شده است که پیک‌های ولتاژی را گرفته و آنها را محدود می‌سازد. همچنین این قسمت، نویزهای رسیده از سمت خروجی را دفع می‌کند. برای حفاظت در برابر اضافه جریان، در مسیر خط خروجی معمولاً فیوز نیز پیش‌بینی می‌شود. اگر این فیوز در داخل کارت وجود نداشته باشد استفاده کننده بایستی از فیوز بیرونی در مسیر اتصال کارت به خروجی استفاده کند. مشخصات فیوز لازم است طبق مشخصات ذکر شده توسط سازنده باشد.

1. Silicon Controlled rectifier
2. Metal oxide varistor



شکل ۲-۲۳ خروجیAC دیجیتال

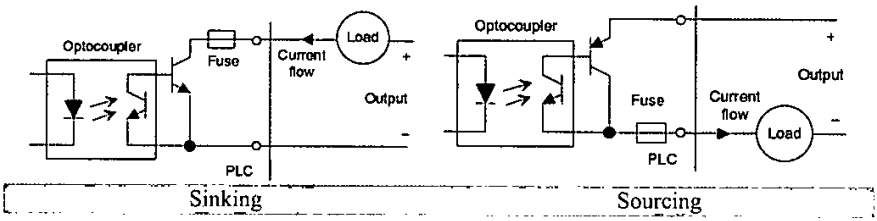
توجه : اگر به اشتباه به نوع Triac ولتاژ DC وصل شود، روشن می‌شود و به‌ظاهر کار می‌کند ولی دیگر خاموش نمی‌شود؛ مگر اینکه تغذیه PLC قطع و وصل گردد.

۲-۵-۶ خروجی Sink / Source

شبیه آنچه برای ورودی‌های دیجیتال ذکر شد، از دیدگاه مسیر جریان در اینجا نیز دو نوع کارت خروجی وجود دارد:

- **نوعSink:** در این نوع مسیر جریان از سمت مصرف کننده به سمت کانال کارت است.
- **نوع Source:** در این حالت مسیر جریان از سمت کارت به سمت مصرف کننده است.

این دو از نظر نوع ترانزیستور متفاوت هستند.

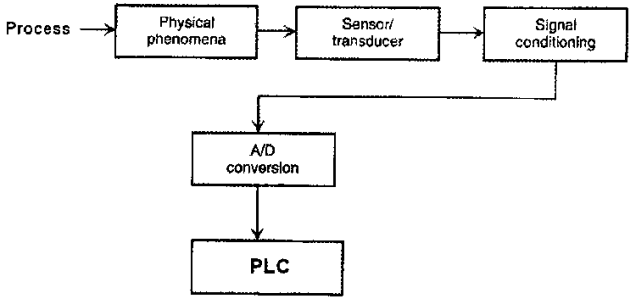


شکل ۲-۲۴ خروجی‌های Sink و Source دیجیتال

۲-۶ ماژول ورودی آنالوگ

سیگنال‌های ورودی آنالوگ برخلاف نوع Discrete (دیجیتال)، مقادیر پیوسته داشته و دو وضعیتی نیستند. پارامترهای فرآیندی مانند دما، فشار، فلو، ارتفاع سطح که در یک بازه مشخص تغییر می‌کنند، از جمله مهمترین سیگنال‌های آنالوگ ورودی برای PLC محسوب می‌شوند.

پارامتر فرآیندی توسط سنسور یا Transducer به سیگنال الکتریکی تبدیل شده، سپس توسط ترانسمیتر با تبدیل به سیگنال الکتریکی استاندارد به کارت ورودی آنالوگ ارسال می‌شود. سیگنال الکتریکی پس از تبدیل به دیتا در اختیار CPU قرار داده می‌شود. معمولاً ترانسدیوسر و ترانسمیتر در یک دستگاه عرضه می‌شوند.

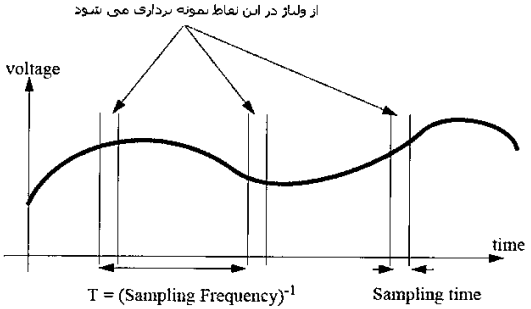


شکل ۲-۲۵ نحوه تبدیل سیگنال آنالوگ ورودی

سیگنال الکتریکی می‌تواند از جنس ولتاژ، جریان یا اهم باشد که البته هر کدام دارای بازه‌های خاص استاندارد است. برخی از مهمترین سیگنال‌های آنالوگ عبارتند از:

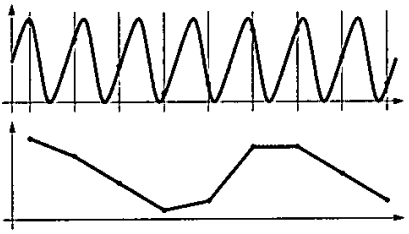
- 4–20 mA
- 0 to +1 volts DC
- 0 to +10 volts DC
- 1 to +5 volts DC
- +/-5 volts DC
- +/-10 volts DC

سیگنال الکتریکی دریافتی توسط کارت اگرچه پیوسته است ولی به‌صورت گسسته خوانده می‌شود. نمونه‌برداری طبق فرکانس تبدیل کارت انجام می‌شود. شکل ۲-۲۶ نمونه‌برداری از یک سیگنال آنالوگ ولتاژی را نشان می‌دهد.



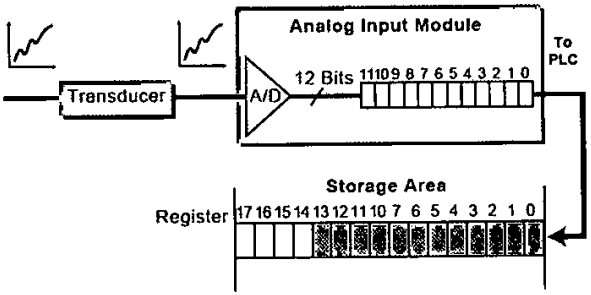
شکل ۲-۲۶ نمونه برداری از آنالوگ ورودی در کارت آنالوگ

نقاط نمونه‌برداری شده به‌عنوان روند تغییرات سیگنال از دیدگاه PLC محسوب می‌گردد. بدیهی است اگر فرکانس (سرعت) نمونه‌برداری خیلی کم باشد سیگنال دریافتی روند واقعی تغییرات را نشان نخواهد داد.



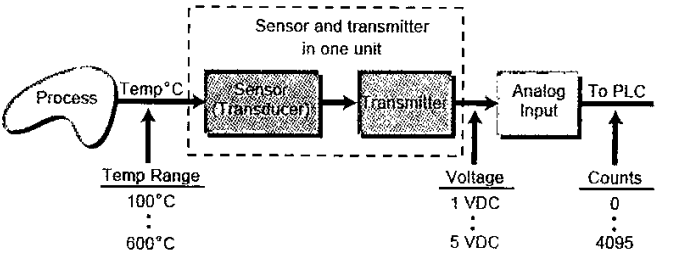
شکل ۲-۲۷ وقتی فرکانس نمونه‌برداری آنالوگ کم باشد.

سیگنال نمونه‌برداری شده توسط کارت به یک مبدل آنالوگ به دیجیتال به‌نام A/D داده می‌شود تا پس از تبدیل به دیتای دیجیتال (یعنی تعدادی 1 , 0) به CPU ارسال شود. مبدل می‌تواند ۸ بیتی تا ۱۶ بیتی باشد، در نوع ۸ بیتی باشد تغییرات سیگنال با هشت بیت 0 و 1 بیان می‌شود که یک بیت آن معرف علامت سیگنال است. برای به‌دست آوردن دقت بیشتر بهتر است از کارتی استفاده شود که نتیجه تبدیل را در بیت‌های بیشتری ذخیره می‌کند. شکل ۲-۲۸ این تبدیل را به‌صورت ۱۲ بیتی نشان می‌دهد. همانطور که مشاهده می‌شود، نتیجه تبدیل به PLC ارسال شده تا در یک حافظه ذخیره گردد.



شکل ۲-۲۸ تبدیل و ذخیره‌سازی آنالوگ ورودی

عدد دریافت شده در PLC به‌صورت ۱۲ بیت صفر و یک است که می‌توان آنرا به‌صورت یک عدد صحیح معادل‌سازی کرد، شکل ۲-۲۹ این موضوع را بهتر نشان می‌دهد.



شکل ۲-۲۹ مثالی از تبدیل دمای اندازه گیری شده فرآیند

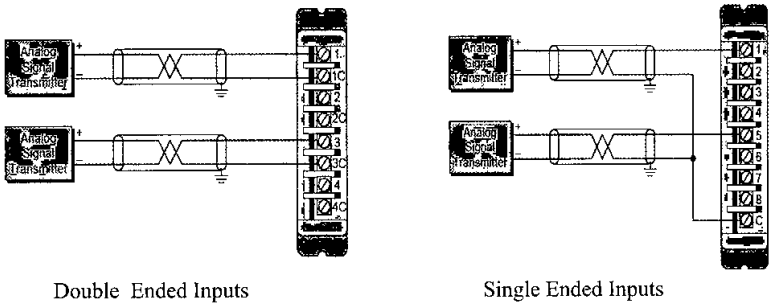
سازندگان مختلف این عدد را به‌صورت‌های مختلف تفسیر می‌کنند. برخی سازندگان حداقل و حداکثر را بر اساس کل بیت‌های تبدیل شده در نظر می‌گیرند. به‌عنوان مثال در حالتی که تبدیل به‌صورت ۱۲ بیتی است حداقل آن مقدار صفر است(معادل ۱۲ بیت صفر) و حداکثر آن ۴۰۹۵ است (معادل ۱۲ بیت که همگی یک باشند). به این ترتیب یک سیگنال آنالوگ که بین ۱ تا ۵ ولت تغییر می‌کند پس از تبدیل بین صفر تا ۴۰۹۵ تغییر خواهد کرد.

برخی دیگر از سازندگان تعداد بیت تبدیل شده را به‌طور کامل برای بازه سیگنال منظور نمی‌کنند و حدودی را برای Over Range و Overflow نیز تعیین می نمایند. نمونه این روش را در فصل‌های بعد برای PLC مدل S7 زیمنس خواهیم دید.

نکات قابل توجه در مورد آنالوگ ورودی

- در سیگنال دیجیتال اگر نویزی روی خط ورودی به کارت بیفتد و سطح ولتاژ را کمی تغییر دهد اهمیتی ندارد زیرا سطح سیگنال معرف فقط یک وضعیت است ولی در سیگنال آنالوگ اگر نویزی روی سیگنال سوار شود، توسط مبدل A/D تبدیل می‌گردد و به غلط معرف تغییر پارامترهای فرآیندی خواهد بود. برای جلوگیری از این تأثیر لازم است تمهیدات لازم برای حفاظت سیگنال نسبت به نویز پیش‌بینی شود. از مهمترین کارها انتقال سیگنال با کابل شیلددار است که شیلد آن زمین شده باشد.
- کارت‌های آنالوگ معمولاً بیش از یک ورودی را می‌پذیرند. بسیاری از سازندگان کارت‌های آنالوگ ۲ و ۴ و ۸ و حتی ۱۶ ورودی عرضه می‌کنند. از آنجا که معمولاً فقط یک مبدل A/D داخل کارت وجود دارد نیاز به یک Multiplexer می‌باشد که به‌ترتیب کانال‌ها را روی مبدل A/D بفرستد.
- قبل از ورود سیگنال به مبدل A/D لازم است عمل تقویت، فیلترسازی و حفاظت انجام شده باشد.
- کانال‌های ولتاژی کارت آنالوگ معمولاً High impedance هستند (در حد مگا اهم) بنابراین تأثیر نامطلوبی روی سیگنال اندازه‌گیری شده نمی‌گذارند.
- کانال‌های جریانی کارت آنالوگ دارای امپدانس پایین (250 – 500 ohm) است تا امکان انتقال جریان از ترانسمیتر به کانال وجود داشته باشد.
- ورودی‌های ولتاژی آنالوگ می‌توانند از یک Com مشترک در کارت استفاده کنند که به آنها Single ended گفته می‌شود، این نوع ورودی‌ها نویزپذیری زیادی دارند. اگر هر ورودی دارای Com خاص باشد به آنها

Double Ended یا Differntial گفته می‌شود و مشکل نویزپذیری ندارد ولی تعداد کانال بیشتری از کارت را اشغال می‌کند.



شکل ۲-۳۰ ورودی آنالوگ ولتاژ با پایه مشترک و پایه مجزا

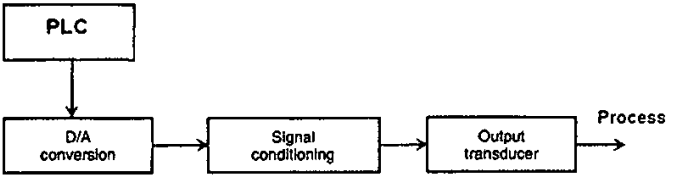
۲-۷ ماژول خروجی آنالوگ

سیگنال‌های خروجی آنالوگ دارای یک وضعیت پیوسته هستند. در نوع Discrete (دیجیتال) مصرف‌کننده دو وضعیتی است (ON/OFF) ولی در اینجا مصرف کننده نیاز به سیگنال پیوسته دارد. به‌عنوان مثال یک ولو کنترلی که بین 0 تا 100 درصد باز و بسته می‌شود، با یک سیگنال آنالوگ خروجی که به‌عنوان مثال می‌تواند بین ۴ الی ۲۰ میلی‌آمپر باشد از سمت PLC کنترل می‌گردد.

سیگنال‌های آنالوگ خروجی از جنس جریان یا ولتاژ هستند که بازه‌های استانداردی برای آنها وجود دارد، مانند:

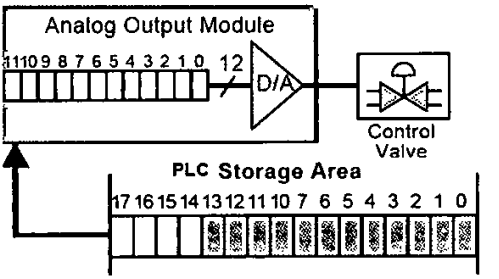
- 4–20 mA
- 0–20 mA
- 0 to +5 volts DC
- 0 to +10 volts DC
- +/- 2.5 volts DC
- +/- 5 volts DC
- +/- 10 volts DC

اصول عملکرد کارت آنالوگ خروجی برعکس کارت آنالوگ ورودی است، یعنی PLC فرمان خود را به‌صورت رشته‌ای از صفر و یک دیجیتال (دیتا) به کارت ارسال می‌کند. کارت از طریق یک مبدل دیجیتال به آنالوگ (D/A)، دیتا را تبدیل به جریان یا ولتاژ کرده و به مصرف‌کننده ارسال می‌نماید.



شکل ۲-۳۱ نحوه تبدیل سیگنال آنالوگ خروجی

عملکرد کارت آنالوگ خروجی ساده‌تر و سریعتر از آنالوگ ورودی است. این کارت به‌سهولت یک عدد را به ولتاژ یا جریان تبدیل می‌کند و به زمان‌بندی و مسائلی مانند نمونه‌برداری نیاز ندارد. شکل ۲-۳۲ ارسال یک عدد ۱۲ بیتی و تبدیل آن در کارت آنالوگ خروجی را نشان می‌دهد.



شکل ۲-۳۲ تبدیل فرمان آنالوگ PLC

نکات قابل توجه

- کارت آنالوگ خروجی می‌تواند چندین کانال داشته باشد؛ انواع متداول ۲ یا ۴ یا ۸ کاناله هستند. از آنجا که معمولاً یک مبدل D/A در کارت تعبیه شده است، خروجی این مبدل توسط یک Multiplexer به کانال انتقال می‌یابد.
- اتصال می‌تواند به‌صورت Single Ended یا Differntial به کارت انجام شود.
- کانال‌های کارت آنالوگ نه تنها از یکدیگر ایزوله هستند، بلکه از PLC نیز ایزوله شده‌اند. از اینرو در صورت بروز اضافه ولتاژ در سمت خروجی، کارت حفاظت خواهد شد.

۲-۸ پارامترهای کارت‌های ورودی و خروجی

در مشخصات فنی کارت‌های ورودی و خروجی پارامترهای مختلف الکتریکی، مکانیکی و محیطی ذکر می‌شود. هر سازنده ویژگی‌های محصول خود را با این پارامتر ها ارائه می‌دهد. در اینجا به اجمال مهمترین پارامترهای کارت‌های ورودی و خروجی دیجیتال را ذکر می‌کنیم که بسیاری از آنها در نوع آنالوگ نیز مشترک هستند.

Input Voltage Rating این پارامتر نوع سیگنال ورودی (که می‌تواند AC یا DC باشد) و دامنه آنرا مشخص می‌کند. معمولاً میزان انحراف مجاز نیز در این قسمت ذکر می‌شود. این انحراف، حدود ±10–15%. دامنه مجاز محسوب می‌شود. به‌عنوان مثال برای کارت دیجیتال ورودی AC 120، ولتاژ ورودی می‌تواند بین 108 تا 132 ولت تغییر کند و در این بازه معتبر است.

Input Current Rating حداقل جریانی که بایستی وسیله به کانال ورودی کارت بفرستد تا مدار الکترونیکی ورودی بتواند کار تبدیل را انجام دهد. به این پارامتر به‌طور غیر مستقیم حداقل توان مورد نیاز نیز گفته می‌شود.

Input Threshold Voltage آستانه‌ای که از آن به بعد سیگنال به‌عنوان وضعیت ON شناخته می‌شود.

Input Delay سیگنال ورودی دیجیتال بایستی در مدت زمان مشخصی در حد آستانه باقی بماند تا به‌عنوان ورودی معتبر شناخته شود. به این زمان تأخیر ورودی گفته می‌شود. این تأخیر برای حذف نوسان ولتاژ و حالات گذرای ناشی از قطع و وصل سوئیچ ها مورد نیاز است.

به‌طور معمول این تأخیر برای ورودی‌های AC بین 9-25 ms و برای ورودی‌های DC بین 1-3 ms است. بنابراین اگر در طول مدت فوق وضعیت سیگنال تغییر کند به PLC گزارش نمی‌شود.

Output Voltage Rating نوع و دامنه ولتاژ خروجی کارت را مشخص می‌کند. AC یا DC بودن سیگنال و میزان تولرانس مجازدرهمین‌پارامترآورده می‌شود که حدود ±10–15%. است. به‌عنوان مثال خروجیVDC 48 می‌تواند بین VDC 42-56 کار کند.

Output Current Rating ماکزیمم جریانی که یک کانال خروجی می‌تواند به‌طور ایمن تحویل مصرف‌کننده بدهد.

Output Power Rating ماکزیمم توان قابل تحویل توسط کارت خروجی در شرایطی که تمام کانال‌های آن فعال باشند. این توان به‌صورت ولت آمپر (VA) بیان می‌شود.

Current Requirements جریان مورد نیاز کارت وقتی که تمام کانال‌های آن فعال باشند. از این پارامتر برای محاسبه منبع تغذیه PLC استفاده می‌گردد.

Surge Current (Max). جریان گذرا که به آن جریان هجومی[1] Inrush Current نیز گفته می‌شود. برای بارهای القایی، در هنگام سوئیچینگ این جریان کشیده می شود. ویژگی آنرا با دامنه و زمان مشخص می‌کنند (مثلاً 20A برای 0.1 S). اگر جریان گذرای فوق بیش از حد تعیین شده در این پارمتر باشد، کارت ممکن است آسیب ببیند. بسیاری از کارت‌ها دارای مدار حفاظت داخلی هستند، اگر این حفاظت وجود نداشته باشد بایستی از دیود هرزگرد در دو سر مصرف کننده یا دیود زنر یا مدارات RC بین کارت و بار استفاده شود.

OFF-State Leakage Current ماکزیمم جریانی که در وضعیت OFF بودن سیگنال از Triac یا ترانزیستور عبور می‌کند و معمولاً در حد چند میکروآمپر تا چند میلی‌آمپر است. زیاد بودن این جریان ممکن است در شرایطی که از وسیله حساس استفاده می‌شود خطای کوچکی را ایجاد کند. به‌عنوان مثال وقتی توسط ولتمتر ولتاژ کانال را اندازه می‌گیریم کمی خطا داشته باشد.

Output-ON Delay میزان تأخیر از زمانی که PLC به ورودی Logic کارت خروجی فرمان ON (وصل) می‌فرستد تا زمانی که خروجی از وضعیت OFF به وضعیت ON درآید.

Output-OFF Delay. میزان تأخیر از زمانی‌که PLC به ورودی Logic کارت خروجی فرمان Off می‌فرستد تا زمانی که خروجی از وضعیت ON به وضعیت OFF درآید.

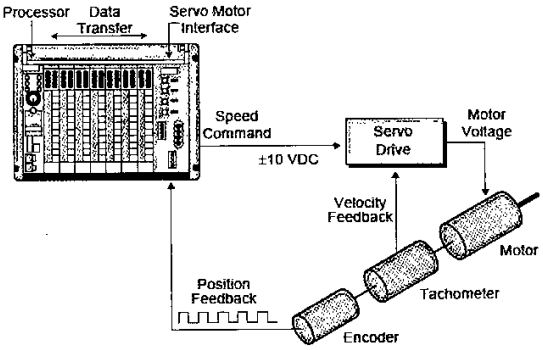
---

1. Inrush Current

Electrical Isolation	ماکزیمم مقدار ولتاژی که مدار ایزوله بین بخش Logic و بخش Power می‌تواند تحمل کند.
Output Fuse Rating	مشخصات فیوز برای کانال‌های خروجی که توسط سازنده به‌طور دقیق محاسبه و ذکر می‌شود و علاوه بر جریان، زمان قطع نیز ذکر می‌شود.
Points Per Module	تعداد کانال‌های کارت
Wire Size	ماکزیمم سایز سیم قابل اتصال به کانال را مشخص می‌کند. سازنده، سایز دقیق سیم مورد استفاده را تعیین نمی‌کند این کار بایستی توسط کاربر محاسبه شود.
Ambient Temperature Rating	ماکزیمم دمای هوای اطراف کارت را مشخص می‌کند تا کارت بتواند به‌راحتی گرمای ایجاد شده در داخل خود را دفع نماید.
Humidity Rating	ماکزیمم میزان رطوبت را در هوای اطراف کارت مشخص می‌نماید.

۲-۹ ماژول‌های مربوط به I/O های خاص

PLC بایستی امکان ارتباط با ورودی و خروجی‌های خاص که از طریق کارت‌های معمولی قابل استفاده نیستند را داشته باشد، این I/O ها کاربرد خاص دارند و خیلی متداول نیستند. حدود ۵ تا ۱۰ درصد I/O ها ممکن است از این نوع باشند. از نمونه این کارت‌ها می‌توان به مواردی که قابلیت دریافت سیگنال‌ها یا پالس‌های سریع مانند پالس‌های انکودر[1] را دارند اشاره نمود. کارت‌هایی که برای کنترل لوپ و کنترل موقعیت و کنترل فازی به‌کار می‌روند نیز از این جمله هستند. این کارت‌ها به‌طور مستقل از CPU می‌توانند کار پردازش را به‌طور کامل انجام داده و فرامین لازم را تولید و ارسال کنند. شکل ۲-۳۳ ماژول کنترلی از انواع فوق را نشان می‌دهد که مستقیماً برای کنترل درایو سرو موتور استفاده می‌شود.



شکل ۲-۳۳ ماژول خاص PLC برای کنترل سرو درایو

۲-۱۰ ماژول ارتباطاتی Communication

برای مقاصد زیر باید PLC مجهز به پورت ارتباطی سریال باشد:

- ارتباط با وسیله پروگرامینگ (برنامه‌ریزی)

1. endcoer

- ارتباط با سایر وسایل مانند PLCهای دیگر یا سیستم کامپیوتر مانیتورینگ
- در نوع جدید ارتباط با فیلدباس

برخی از پورت‌های ارتباطی بیان شده به‌صورت On board روی PLCتعبیه می‌شوند. به‌ویژه در PLCهای جدید، پورت‌های شبکه روی CPU اضافه شده‌اند ولی در هر صورت کارت شبکه ارتباطی نیز برای ارتباطات مختلف شبکه توسط سازنده عرضه می‌شود.

پورت ارتباطی برای Programming

برای ارتباط با وسیله برنامه‌ریزی، توسط سازندگان PLC پورت ارتباطی روی خود CPU تعبیه شده است که می‌توان آنرا به کامپیوتر متصل نمود. برخی از سازندگان این پورت را با استاندارد باز عرضه کرده‌اند. در این موارد با ارتباط سریال مانند RS232 می‌توان به‌سهولت و بدون نیاز به واسط خاصی بین کامپیوتر و PLC ارتباط برقرار کرد. برخی دیگر از سازندگان، این پورت را با پروتکل انحصاری خود عرضه نموده‌اند که در این حالت برای ارتباط کامپیوتر با PLC نیاز به کارت خاص جهت نصب روی کامپیوتر یا نیاز به آداپتور برای ارتباط پورت‌های کامپیوتر با پورت PLC وجود دارد.

ارتباط با سایر PLCها

برای این منظور معمولاً نیاز به کارت ارتباطی با پروتکل استانداردی است که همه سازندگان آن را ساپورت کنند. مانند:

- اترنت صنعتی
- پروفی‌باس
- مدباس

ارتباط بین PLC و سیستم مانیتورینگ

برای این منظور نیز از پروتکل‌های استاندارد شبکه استفاده می‌شود. یکی از معروف‌ترین و مرسوم‌ترین آنها شبکه اترنت‌صنعتی با پروتکل TCP/IP یا ISO است که سازندگان مختلف آنرا پشتیبانی می‌کنند. در این‌حالت با قراردادن کارت اترنت در کنار PLC و اتصال آن به شبکه‌ای که کامپیوتر مانیتورینگ نیز به آن متصل است، امکان ارتباط فراهم می‌گردد.

ارتباط بین PLC و وسایل فیلد

برخی از سازندگان، امروزه CPUهای PLC را به پورتی که می‌تواند به شبکه‌های فیلدباس برای اتصال وسایل فیلد ارتباط یابد مجهز کرده‌اند و برخی دیگر برای این منظور کارت مجزایی عرضه نموده‌اند. از شبکه‌های معروف فیلدباس می‌توان به موارد زیر اشاره کرد:

- Foundation Fieldbus
- Profibus-PA
- ASI
- DeviceNet

فصل ۲

۲-۱۱ سایر اجزای PLC

به‌جز موارد ذکر شده اجزای دیگری را نیز می‌توان در ارتباط با PLC ذکر نمود از جمله:

- ماژول های توسعه
- وسیله برنامه‌ریزی[1]

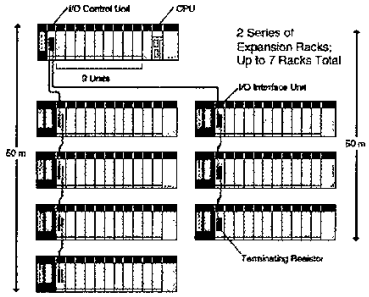
۲-۱۱-۱ ماژول‌های توسعه

اگر تعداد کارت‌های ورودی و خروجی یا سایر ماژول‌ها زیاد باشد، فقط تعداد محدودی را می‌توان روی باس اصلی متصل به CPU نصب کرد. برای اتصال سایر ماژول‌ها دو راه حل وجود دارد:

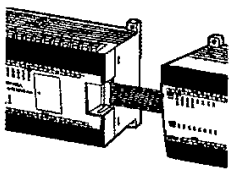
- نصب ماژول‌ها روی Rack اضافی
- نصب ماژول‌ها روی Remote I/O

رک اضافی

این رک توسط کارت‌های خاصی به رک اصلی مرتبط می شود. یک کارت روی رک اصلی و یک کارت روی رک اضافی قرار می‌گیرد و این دو توسط کابل و کانکتور خاصی به یکدیگر مرتبط می‌شوند. هر ماژول که روی رک اضافی نصب می‌شود شبیه ماژول‌هایی که روی رک اصلی نصب می‌شوند دیتا را از طریق باس به CPU منتقل می‌کند.
در مواردی که PLC به‌صورت یکپارچه باشد نیز امکان توسعه I/O از طریق رک اضافی وجود دارد.



رک اضافی در PLC ماژولار



رک اضافی در PLC یکپارچه

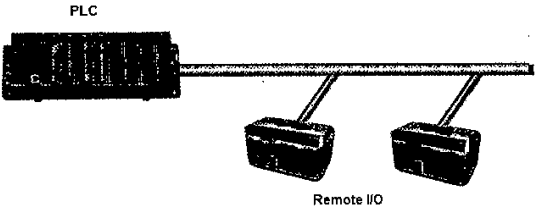
شکل ۲-۳۴ رک اضافی در PLCهای یکپارچه و ماژولار

Remote I/O

این ماژول توسط کابل شبکه (مانند شبکه پروفی‌باس) با CPU ارتباط برقرار می‌کند. ماژول‌های I/O که روی Remote I/O نصب می‌گردند دیتای خود را به ماژول رابط شبکه ارسال می‌کنند. ماژول رابط شبکه دیتا را طبق پروتکل شبکه

1. Programming Device

بسته‌بندی و کدگذاری می‌کند و به‌صورت سریال از طریق کابل شبکه به CPU ارسال می‌کند. سرعت انتقال اطلاعات نسبت به رک اضافی کندتر است.



شکل ۲-۳۵ استفاده از Remote I/O

۲-۱۱-۲ وسیله برنامه‌ریزی

برای برنامه ریزی CPU و تنظیمات و پیکربندی سخت‌افزار و شبکه، وسیله برنامه‌ریزی مورد نیاز است. برخی سازندگان کامپیوترهای خاصی را برای این منظور ارائه می‌نمایند که سخت‌افزار و نرم‌افزارهای لازم روی آن نصب شده است. با این حال معمولاً می‌توان از طریق کامپیوترهای معمولی نیز برنامه‌ریزی را انجام داد. برخی سازندگان ارتباط بین کامپیوتر و CPU را بدون نیاز به رابط یا سخت افزار خاصی فراهم کرده‌اند. به‌عنوان مثال می‌توان با ارتباط دادن پورت سریال RS232 کامپیوتر به پورت سریال CPU مستقیماً این ارتباط را برقرار کرد. برخی دیگر از سازندگان پورت برنامه‌ریزی CPU را به‌صورت انحصاری طراحی کرده‌اند و استفاده کننده مجبور به برقراری ارتباط از طریق سخت‌افزار یا رابط خاصی خواهد بود.

۲-۱۲ پرسش و تحقیق

- بازه تغییرات خروجی مبدل A/D کارت‌های آنالوگ سازندگان مختلف را با یکدیگر مقایسه کنید.
- کاربرد سیگنال‌های دیجیتال 5V DC TTL در کجاست؟

۲-۱۳ تست‌های خودآزمایی

۱- وظیفه‌ی CPU در یک PLC چیست؟

الف) تبدیل دیتا به سیگنال الکتریکی
ب) تبدیل سیگنال الکتریکی به دیتا
ج) پردازش برنامه
د) هر سه مورد

۲-وظیفه‌ی کارت ورودی در PLC چیست؟

الف) ارسال سیگنال به تجهیزات تحت کنترل
ب) تبدیل سیگنال الکتریکی به دیتا
ج) پردازش برنامه
د)تبدیل دیتا به سیگنال الکتریکی

۳-وظیفه‌ی کارت خروجی در یک PLC چیست؟

الف) دریافت ورودی‌ها
ب) تبدیل سیگنال الکتریکی به دیتا
ج) پردازش برنامه
د)تبدیل دیتا به سیگنال الکتریکی

فصل ۲

۴- وظیفه‌ی ..... تأمین ولتاژ و جریان مورد نیاز CPU و سایر اجزای PLC می‌باشد.

الف) CPU ب) کارت ورودی ج) منبع تغذیه د) کارت خروجی

۵- به‌طور خلاصه استفاده از ترانس در مسیر تغذیه AC منبع تغذیه PLC برای مقاصد زیر است:

الف) تثبیت ولتاژ در صورت نوسان ولتاژ خط ب) حذف هارمونیک‌های خط

ج) ایزوله‌سازی د) هر سه مورد

۶- وظیفه‌ی I/O Bus در یک CPU چیست؟

الف) ارتباط CPU با ورودی، خروجی‌ها ب) همزمان‌سازی ج) فرمان داخلی د) تبادل آدرس

۷- ..... از حافظه‌های CPU می‌باشد که سیستم عامل CPU در آن ذخیره می‌شود.

الف) RAM ب) EPROM ج) ROM د) EEPROM

۸- ..... از حافظه‌های داخلی CPU بوده که در صورت قطع تغذیه‌ی آن، اطلاعات موجود درآن پاک می‌شود.

الف) RAM ب) EPROM ج) ROM د) EEPROM

۹-در مورد کار CPU کدام مورد صحیح نمی‌باشد؟

الف) پردازش برنامه ب) دریافت ورودی‌ها و ارسال خروجی‌ها

ج) اجرای سیکل کاری مشخص (سیکل‌اسکن) د) تبدیل سیگنال آنالوگ به دیجیتال

۱۰-در مورد کارت (ماژول) ورودی دیجیتال کدام مورد صحیح است؟

الف) ورودی‌های دیجیتال مانند سوئیچ‌ها و سنسورهای دیجیتال به آن متصل می‌شوند.

ب) فقط دو حالت قطع و وصل را تشخیص می‌دهد.

ج) سیگنال الکتریکی دریافتی را به دیتا تبدیل می‌کند.

د) هر سه مورد

فصل ۳

اصول طراحی و انتخاب PLC و اجزای آن

۳-۱ مقدمه
۳-۲ شناخت فرآیند گام اول در فاز طراحی
۳-۳ بررسی نقشه‌های P&ID
۳-۴ استخراج لیست I/O
۳-۵ انتخاب کارت‌های I/O
۳-۶ انتخاب کارت‌های خاص
۳-۷ بررسی چیدمان کارت‌های I/O و نیاز به Remote I/O
۳-۸ انتخاب کارت‌های شبکه
۳-۹ انتخاب CPU
۳-۱۰ انتخاب رک
۳-۱۱ انتخاب منبع تغذیه
۳-۱۲ انتخاب پنلPLC
۳-۱۳ کلید اضطراری
۳-۱۴ انتخاب کابل با طول مناسب
۳-۱۵ حفاظت در مقابل نویز
۳-۱۶ پرسش و تحقیق
۳-۱۷ تست‌های خودآزمایی

در این فصل اصول انتخاب و طراحی PLC به اجمال و بر اساس مهمترین نکات بیان شده است. مطالعه این فصل به افرادی که در زمینه طراحی سیستم کنترل و انتخاب اجزای PLC فعالیت دارند توصیه می‌شود.

چکیده مطالب

- قدم‌های اولیه در فاز طراحی بسیار مهم هستند. هر تغییر کوچک در مراحل اول، تغییرات بزرگ در نتیجه نهایی را به‌دنبال خواهد داشت.
- قدم‌های فاز طراحی لازم است چندین بار برای انتخاب‌های مختلف تکرار شده و نتایج نهایی با هم مقایسه شوند.
- ابتدا بایستی فرآیند شناخته شود و شرح عملکرد و ترتیب عملیات آن مستند شود تا بر اساس آن مدارک دیگر قابل استخراج باشد.
- لیست I/Oها از اولین مدارک مورد نیاز برای طراحی است که از شرح عملکرد و لیست ابزاردقیق و نقشه‌های P&ID استخراج می‌گردد. I/Oهای خاص در لیست علامت‌گذاری می‌شوند.
- براساس لیست I/Oها، تعداد و نوع کارت‌های ورودی و خروجی تعیین می‌شود. اتصال مستقیم I/Oها یا استفاده از Remote I/O در این مرحله می‌تواند بررسی شود.
- کارت‌های شبکه مورد نیاز برای مانیتورینگ یا تبادل دیتا یا سایر نیازها مورد بررسی قرار می‌گیرد.
- CPU با توجه به تعداد I/O و سایر نیازهای فرآیند مشخص می‌گردد.
- با توجه به مجموع کارت‌های مورد نیاز، منبع تغذیه انتخاب می شود.
- پنل PLC با توجه به اجزای تعیین شده در مراحل قبل طراحی می‌گردد.
- نوع کابل و مسیر کابل با توجه به حداکثر افت مجاز تعیین می‌شود.
- در صورت استفاده از Remote I/O طول کابل با توجه به سرعت شبکه تعیین می‌شود.
- نکات مربوط به حفاظت در مقابل نویز و طراحی سیستم زمین مناسب مد نظر قرار می‌گیرد.
- تمام مراحل برای رسیدن به نتیجه مطلوب بارها بازنگری می‌گردد.

فصل ۳

اصطلاحات و تعاریف

PE

Protective Earth

ارت حفاظتی است که برای زمین کردن تابلوهای برق و بدنه ماشین آلات و بدنه پنل‌ها به‌کار می‌رود.

CE

Clean Earth

ارت کنترل و ابزاردقیق است که به‌طور مستقل برای زمین کردن PLC و اجزای آن و شیلد کابل‌های ابزاردقیق و شیلد کابل‌های شبکه از آن استفاده می‌شود.

P&ID

Piping and Instrumentation Diagram

نقشه‌ای که در آن مسیر فرآیند همراه با ابزاردقیق نصب شده روی آن و لوپ‌های کنترلی و آلارم‌های مورد نیاز نشان داده می‌شود.

Fail Safe

به اصطلاحات و تعاریف فصل ۶ مراجعه شود.

High Availability

به اصطلاحات و تعاریف فصل ۶ مراجعه شود.

NEMA

National Electrical Manufacturers Association

مؤسسه استانداردی است که در ایالات متحده در زمینه برق فعالیت می‌کند.

MCR

Master Control relay

رله حفاظتی است که در حالت نرمال برق‌دار است، اگر تغذیه آن قطع شود منجر به قطع تغذیه ورودی‌ها و خروجی‌های PLC می‌گردد.

Outdoor – Indoor

Indoor وقتی تجهیز در فضای بسته مانند اتاق کنترل نصب شود.

Outdoor وقتی تجهیز در فضای آزاد و باز نصب شود.

Warranty

وارانتی قراردادی است که معمولا" بین سازنده و خریدار - گذاشته می‌شود و سازنده تاکید می‌کند که محصول او در مدت زمان مشخص که معمولا" از دوره گارانتی بیشتر است با همان مشخصات اولیه کار می‌کند و در صورت بروز مشکل برای محصول، هزینه تعمیر یا تعویض قطعات را به‌عهده می‌گیرد و به این طریق به مصرف کننده اطمینان می‌دهد که در طول این مدت نیازی به انجام هزینه تعمیرات ندارد.

تفاوت وارانتی و گارانتی در این است که در دوران گارانتی در صورت بروز مشکل در محصول، سازنده آن را تعویض یا تعمیر می‌کند یا هزینه مشتری را بر می‌گرداند؛ ولی در دوران وارانتی سازنده صرفاً تعمیر یا تعویض قطعات را به‌عهده می‌گیرد ولی وسیله را پس نمی‌گیرد.

فصل ۳

۳-۱ مقدمه

امروزه هم تنوع سازندگان PLC زیاد است و هم PLCهای ساخته شده توسط یک سازنده بسیار متنوع است. این تنوع زیاد، طراح را برای انتخاب یک PLC مناسب با مشکل مواجه می‌کند. به‌ویژه وقتی که طراح با یک فرآیند بزرگ یا یک ماشین پیچیده سر و کار دارد.

از اینرو فاز طراحی یکی از فازهای مهم در پیاده‌سازی سیستم‌های کنترل محسوب می‌شود. طراحی صرفاً مربوط به انتخاب PLC مناسب نیست، بلکه انتخاب همه اجزای سیستم کنترل است که برخی از آنها عبارتند از:

- انتخاب سنسورها و عملگرهای مناسب
- تعیین محل مناسب برای نصب سنسورها و عملگرها
- انتخاب پنل مناسب
- تعیین محل مناسب برای نصب پنل
- تعیین محل مناسب برای نصب سیستم مانیتورینگ
- مشخص کردن تعداد I/O مورد نیاز
- بررسی نیاز یا عدم نیاز به Remote I/O
- مشخص کردن CPU مورد نیاز
- مشخص کردن منبع تغذیه مورد نیاز
- مشخص کردن تجهیزات حفاظتی مورد نیاز
- تعیین چیدمان اجزا در داخل پنل
- تعیین نوع کابل
- مشخص کردن مسیر کابل
- مطالعه سیستم زمین و تعیین زمین مناسب

پارامترهای فوق برای سیستم‌های معمولی متداول هستند. در سیستم‌های خاص، بررسی‌های دقیق‌تر و پیچیده‌تری مورد نیاز است مانند:

- در شرایطی که فرآیند ایجاب می‌کند اجزای سیستم کنترل به‌صورت High Availability باشند.
- در شرایطی که فرآیند ایجاب می‌کند اجزای سیستم کنترل به‌صورت Fail Safe باشند.

در این بخش هدف آن است که خواننده با کلیات نکاتی که مربوط به طراحی و انتخاب سیستم‌های کنترل معمولی است آشنا شود. انتخاب نوع سنسور و عملگر و نیز بررسی سیستم‌های خاص که نیاز به افزونگی یا سطح ایمنی بالا دارند خارج از چارچوب این کتاب است.

مطالعات فاز طراحی ممکن است منجر به انتخاب PLC به‌عنوان سیستم کنترل نشود. به‌عنوان مثال وقتی لوپ‌های کنترلی زیادی در فرآیند وجود داشته باشد، سیستم کنترل DCS به‌جای PLC مطرح می‌شود که این مورد موضوع بحث ما نیست. در اینجا با فرض اینکه نیاز به سیستم کنترل غیر متمرکز نیست و هدف رسیدن به مشخصات مناسب برای PLC است بحث را دنبال می‌کنیم.

نکته‌ای که باید به آن توجه نمود این است که در هر قدم از فاز طراحی نتایجی به‌دست می‌آید که الزاماً قطعی نیستند. با تغییر کوچکی در پارامترها نتایج کاملاً متفاوت خواهند بود. هر تغییر کوچک در یک فاز منجر به تغییرات بزرگتر در

فازهای بعدی می‌شود از اینرو قدم‌های اولیه بسیار مهم هستند و اگر بعداً دچار تغییر شوند نیاز به بازنگری اساسی در کل طرح ارائه شده وجود دارد. تغییرات در قدم‌های آخر ممکن است با بازنگری جزئی قابل حل باشد.

برای به‌دست آوردن بهترین طرح لازم است کار طراحی از قدم اول تا آخر با انتخاب‌های مختلفی انجام شود و نتایج نهایی به‌دست آمده با یکدیگر مقایسه شوند.

۳-۲ شناخت فرآیند گام اول در فاز طراحی

برای پیاده‌سازی سیستم‌های کنترل امروزه با دو نوع فرآیند روبرو هستیم:

۱. فرآیندی که با سیستم کنترل قدیمی در حال کار است.

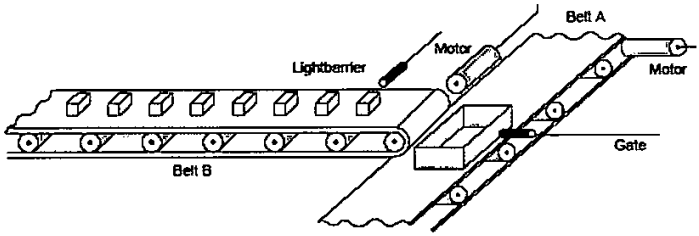
۲. فرآیند جدیدی که قرار است از ابتدا پیاده‌سازی شود.

در نوع اول، مطالعه فرآیند و نیازهای آن کار پیچیده‌ای نیست. اگر نیاز به بهبود یا اصلاح فرآیند باشد نیازمند مطالعات جدید خواهیم بود.

در نوع دوم کار مطالعه و شناخت فرآیند بسته به نوع ماشین، ابعاد فرآیند و میزان پیچیدگی آن متفاوت است. در فرآیندهای بزرگ و پیچیده کار به‌صورت تیمی انجام می‌شود. در حین کار تیمی، مهندس طراح اتوماسیون می‌تواند به شرح عملکرد سیستم دست پیدا کند که این نقطه آغاز کار است.

بدیهی است شرح عملکرد سیستم، صرفاً توالی عملیات مورد نیاز در فرآیند را برای تولید محصول نهایی بیان می‌کند و به‌صورت کلی نوشته می‌شود و سیستم اتوماسیون بر اساس آن پایه‌گذاری می‌گردد.

به‌عنوان یک مثال ساده می‌توان شکل ۳-۱ را به‌عنوان یک فرآیند ساده بسته‌بندی در نظر گرفت.



شکل ۳-۱ بخشی از یک فرآیند بسته‌بندی

متن زیر بخشی از شرح عملکرد این سیستم کوچک را توصیف می‌کند.

نمونه شرح عملکرد سیستم ساده بسته‌بندی

وقتی اپراتور سیستم را استارت کرد، ابتدا نوار A (Belt A) روشن شود تا جعبه خالی روبروی نوار B قرار گیرد. در این حالت نوار A متوقف و نوار B (Belt B) روشن شود. وقتی تعداد ۸ بسته محصول وارد جعبه خالی شد، نوار B از کار بیفتد و نوار A روشن شود تا جعبه پر شده از نوار خارج شده و جعبه خالی جدید روبروی نوار B قرار گیرد و تا زمانی که اپراتور فرمان Stop صادر نکرده است این سیکل تکرار شود. اگر اپراتور فرمان Stop صادر کرد، سیکل فعلی کامل شود یعنی جعبه خالی موجود پر و جابه‌جا شود ولی سیکل جدید شروع نشود. اگر اپراتور فرمان قطع اضطراری داد هر دو نوار بلافاصله متوقف شوند حتی اگر سیکل نیمه تمام باشد.

پس از اینکه عملکرد فرآیند به‌طور دقیق طراحی شد، نقشه‌ها و مدارکی که تعداد سنسورها و عملگرها و موقعیت کلی آنها را نشان می‌دهد تهیه می‌شود. از جمله این مدارک که مبنای کارهای بعدی در انتخاب سیستم کنترل است می‌توان به موارد زیر اشاره کرد:

- مدارک P&ID که بیان کننده شرح فرآیند و وسایل ابزار دقیق مورد نیاز هستند.
- Instrument List که تعداد و مشخصات سنسورها و عملگرها را نشان می‌دهد.

۳-۳ بررسی نقشه‌های P&ID

از نقشه‌های P&ID می‌توان اطلاعات مختلفی را استخراج کرد:

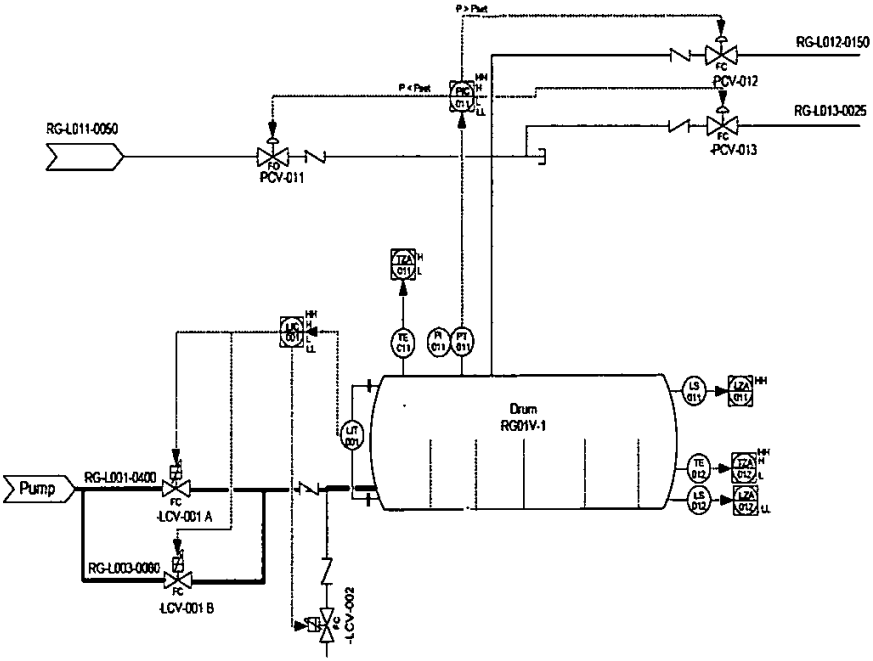
- عملیات فرآیند و تجهیزات فرآیندی به‌کار رفته در آن
- محل تقریبی نصب سنسورها و عملگرها
- لوپ‌های کنترلی مورد نیاز
- شرایط آلارم و تریپ‌های سیستم

بررسی نقشه‌های P&ID و علائم به‌کار رفته در آنها را بایستی در کتاب‌های ابزاردقیق جستجو کرد و تشریح آنها موضوع بحث این کتاب نیست. صرفاً برای آشنایی بیشتر خواننده، یک نمونه نقشه P&ID کوچک را مورد بررسی قرار می‌دهیم. برای شناخت علائمی که در این نقشه‌ها به‌کار می‌رود به پیوست ۲ مراجعه نمایید. شکل ۳-۲ نمونه‌ای از یک نقشه P&ID را نشان می‌دهد که در آن دو لوپ کنترلی برای یک مخزن ترسیم شده است.

- **لوپ فشار**: سیگنال فشار از ترانسمیتر به کنترلر داده شده و کنترلر بر اساس آن به ولوهای کنترلی فرمان می‌دهد.
- **لوپ سطح**: سیگنال ارتفاع مخزن از ترانسمیتر سطح به کنترلر منتقل شده و کنترلر بر اساس آن به ولوهای هیدرولیکی فرمان می‌دهد.

علاوه بر لوپ‌های کنترلی فوق، سیگنال‌هایی نیز برای آلارم وجود دارد.

- **آلارم سطح**: سوئیچ‌های سطح، وضعیت را به کنترلر گزارش می‌کنند تا آلارم‌های سطح پایین و سطح بالا توسط آن تولید شود.
- **آلارم دما**: سنسور دما که می‌تواند یک Pt100 باشد، میزان دما را به کنترلر اعلام می‌کند تا آلارم‌های دمای بالا و پایین را ایجاد نماید.



شکل ۳-۲ نمونه‌ای از یک نقشه P&ID

حروف‌و علائم به‌کار رفته در نقشه طبق استاندارد ISA می‌باشد. برای فهم نقشه فوق به راهنمای جدول ۳-۱ توجه کنید.

جدول ۳-۱ راهنمای علائم به‌کار رفته در نقشه شکل ۳-۲

راهنمای شکل ها	
وسیله نصب شده در فیلد (مانند ترانسمیتر یا ترموکوپل و ...)	○
وسیله نصب شده در اتاق کنترل (مانند PLC)	▭
Check Valve	⟋
Hydraulic Cylinder Type Valve	⧓
	⧓
خطوط سیگنال الکتریکی	----------
خطوط لوله و Piping فرایند	————

راهنمای علائم	
TE	Temperature Element
PT	Pressure Transmitter
LS	Level Switch
LIC	Level Indicator and Control
LCV	Level Control Valve
PIC	Pressure Indicator Control
PCV	Pressure Control Valve
TZA	Temperature Alarm
LZA	Level Alarm

در اینجا هدف آموزش P&ID نیست، بلکه هدف نحوه استخراج اطلاعات مورد نیاز طراحی از آن می‌باشد. به‌عنوان مثال نقشه فوق اطلاعات زیر را در اختیار قرار می‌دهد:

- تعداد دو ورودی دیجیتال برای Level Switch ها در سمت PLC مورد نیاز است.
- تعداد چهار ورودی آنالوگ برای ترانسمیترهای فشار و ترانسمیتر سطح و سنسورهای دما در سمت PLC مورد نیاز است.
- تعداد دو خروجی دیجیتال در PLC برای کنترل ولوهای هیدرولیکی مورد نیاز است.
- تعداد دو خروجی آنالوگ در PLC برای کنترل ولوهای تناسبی مورد نیاز است.
- کنترلر بایستی توانایی پردازش حداقل دو لوپ کنترلی را داشته باشد.
- همه سنسورها و ترانسمیترها روی مخزن نصب شده‌اند، بنابراین می‌توان سیگنال آنها را از طریق Remote I/O نیز انتقال داد تا کابل‌کشی کمتر باشد.

۳-۴ استخراج لیست I/O

پس از مشخص شدن شرح عملکرد سیستم و ترسیم نقشه‌های اولیه مانند P&ID و تهیه لیست Instrument، نوبت به استخراج لیست I/Oها می‌رسد. در این لیست مشخصات زیر ارائه می‌گردد. در این لیست علاوه بر سنسورها، ترانسمیترها و عملگرها، کلیدها و شستی‌های اپراتوری نیز وارد می‌شود.

عنوان سیگنال که لازم است نام‌گذاری به اختصار و با مفهوم باشد.

نوع سیگنال که دیجیتال است یا آنالوگ، ورودی است یا خروجی، برای این منظور می‌توان از حروف AI,AO,DI,DO استفاده کرد.

آدرس در PLC این ستون برای استفاده بعدی است و در این مرحله خالی می‌ماند.

مبدا سیگنال برای ورودی‌ها مبدا می‌تواند سنسور باشد، در این‌صورت کد وسیله که در نقشه P&ID آمده برای آن در این ستون ذکر می‌شود.

برای خروجی‌ها ، مبدا PLC است.

مقصد سیگنال برای ورودی‌ها ، مقصد PLC است.

برای خروجی‌ها، کد مربوط به عملگر از روی نقشه P&ID ذکر می‌شود.

بازه سیگنال برای سیگنال‌های آنالوگ بازه را مشخص می‌کند، مانند 4-20 mA یا 1-5V .

برای سیگنال‌های دیجیتال یک عدد مانند 24VDC یا 120AC که مقدار نامی سیگنال را نشان می‌دهد، وارد می‌گردد.

این‌مشخصات‌باتوجه‌به‌نوع‌سنسوریاترانسمیتر یا عملگر که قبلاً در لیست Instrument آمده وارد می‌شود.

نوع اتصال این ستون در این مرحله خالی می‌ماند. پس از مشخص شدن کل سیستم و نحوه جمع‌آوری I/O ها که به‌صورت مستقیم یا از طریق شبکه است این ستون پر می‌شود.

برای سیگنال‌هایی که به‌طور مستقیم سیم‌کشی می‌شوند کلمه Direct و برای سیگنال‌هایی که از طریق Remote I/O ارتباط می‌یابند کلمه Network را می‌توان وارد نمود.

سایر سایر اطلاعات مورد نیاز به‌عنوان توضیح اضافی در این ستون وارد می‌شود، مثلاً سیگنال‌های سریع را می‌توان در این ستون مشخص نمود.

جدول ۳-۲ نمونه اطلاعات فوق‌الذکر را در I/O List نشان می‌دهد.

جدول ۳-۲ نمونه I/O List تفصیلی

Item No	Signal Description	Signal Type	PLC Address	Source	Destination	Connection Type	Signal Range
1	Flow of Natural Gas	AI		FT401	PLC		4-20 mA
2	Air Temperature	AI		TE422			Pt100
3	Main Control Valve	AO		PLC	PCV51		0-10V
4	Pressure Switch	DI		PS273	PLC		24 VDC
5	Gas Shuttof valve	DO		PLC	SV101		24 VDC
6							

فصل ۳

پس از اینکه تمام I/Oها در این لیست آورده شد، خلاصه آن به‌صورت جمع‌بندی شده در لیستی مانند جدول ۳-۳ آورده می‌شود. این لیست مبنای انتخاب کارت‌های I/O در قدم بعدی است.

جدول ۳-۳ نمونه I/O List جمع‌بندی شده

Item No	Signal Type	Quantity	Remarks
1	AI 4-20mA	121	
2	AI Pt100	18	
3	AO 4-20mA	19	
4	DI 24VDC	190	
5	DO 24VDC/0.5A	56	

۳-۵ انتخاب کارت‌های I/O

انتخاب کارت‌های I/O بر مبنای لیست I/O صورت می‌گیرد، ولی ابتدا لازم است به جمع مقادیر I/O List درصدی به‌عنوان Spare (رزرو) اضافه شود تا اگر بعداً تعدادی وسیله به سیستم اضافه شد یا اگر نیاز بود برخی کانال‌ها جابه‌جا شود این کار به‌سهولت امکان‌پذیر باشد. در بسیاری از طرح‌ها این عدد را ۲۰٪ در نظر می‌گیرند.

با توجه به نوع سیگنال و انواع کارت‌های موجود می‌توان تعداد کارت مورد نیاز را برای یک یا چند مدل کارت بررسی کرد. کارت‌های ورودی و خروجی از نظر تعداد کانال و قابلیت‌های خاص متفاوت هستند.

مثال قبلی را برای یک مدل کارت بررسی می‌کنیم. با تقسیم تعداد کل سیگنال‌ها بر تعداد کانال کارت می‌توان تعداد کارت مورد نیاز را به‌دست آورد.

جدول ۳-۴ نمونه Module List

Item No	Signal Type	Quantity	20% Spare	Total Qty	Channels per Module	Number of Requiered Modules
1	AI 4-20mA	121	24	145	8	19
2	AI Pt100	18	4	22	8	3
3	AO 4-20mA	19	4	23	8	3
4	DI 24VDC	190	38	228	32	8
5	DO 24VDC/0.5A	56	11	67	16	5

نکات قابل توجه

- اگر ویژگی‌های خاصی برای برخی ورودی و خروجی‌ها مورد نیاز باشد، ممکن است نیاز به کارت خاص و جداگانه‌ای باشد.
- اگر در فازهای بعدی مشخص شود که نیاز به نصب Remote I/O در فیلد وجود دارد، جدول ۳-۴ دچار تغییر خواهد شد.

۳-۶ انتخاب کارت‌های خاص

برخی از سیگنال‌ها را نمی‌توان به کارت‌های I/O معمولی اتصال داد. به‌عنوان مثال اگر در یک خط بسته‌بندی، قطعات تولید شده با سرعت زیاد از روبروی سنسور عبور کنند ممکن است PLC نتواند آنها را به‌درستی شمارش کند. در این حالت برای شمارش سریع، کارت ورودی خاصی مورد نیاز است.

۳-۷ بررسی چیدمان کارت‌های I/O و نیاز به Remote I/O

نکات زیر قابل توجه است

- به‌دلایل اقتصادی و فنی، در برخی طرح‌ها کارت I/O در کنار PLC قرار نمی‌گیرد و همه I/Oها روی Remote I/O قرار داده می‌شود.
- Remote I/O را می‌توان در کنار پنل PLC یا در سایت و در نزدیکی تجهیزات نصب کرد. در نوع اخیر کابل‌کشی کمتر است ولی سیستم ریسک‌پذیر است زیرا با قطع شدن کابل شبکه، ممکن است سیگنال همه‌ی I/O ها از دست برود.
- Remote I/Oها از نظر تعداد ورودی و خروجی متفاوت و متنوع هستند. برخی از آنها به‌صورت یکپارچه و برخی دیگر ماژولار بوده و کارت‌های I/O روی آنها نصب می‌شود.
- معمولاً وقتی سرعت و قابلیت اطمینان بالا مورد نیاز است، کارت در کنار PLC نصب می‌شود.

در صورتی‌که نیاز به Remote I/O مورد تایید باشد، لازم است لیست کارت‌های I/O قبلی مورد بازنگری قرار گیرد. در حالتی که Remote I/Oها در فیلد نصب شوند بازنگری اساسی خواهد بود زیرا سیگنال‌های نزدیک به هم را بایستی از طریق یک Remote I/O منتقل نمود.

۳-۸ انتخاب کارت‌های شبکه

در سیستم‌های اتوماسیون امروزی، وجود کارت شبکه روی PLC از ضروریات سیستم محسوب می‌شود. مهمترین و متداول‌ترین کاربرد این کارت برای مانیتورینگ است که در حال حاضر بیشتر از شبکه اترنت صنعتی برای این منظور استفاده می‌شود. به‌دلیل پیشرفت‌های زیادی که در تکنولوژی ساخت تجهیزات این شبکه اتفاق افتاده است، می‌توان برخلاف گذشته، به این شبکه با دیدگاه یک شبکه مطمئن و Real Time نگاه کرد؛ از اینرو علاوه بر مانیتورینگ در کاربردهای حساسی همچون تبادل دیتا بین PLCها نیز قابل استفاده است.

با توجه به موارد فوق به‌نظر می‌رسد که در فاز طراحی برای هر PLC لازم است این کارت شبکه دیده شود. اگر چه در موارد خاصی ممکن است شبکه‌های دیگر مانند مدباس یا پروفی‌باس مورد نیاز باشد ولی کاربرد آنها به اندازه کاربرد اترنت وسیع و فراگیر نیست به‌همین جهت در قدم‌های اولیه طراحی که هنوز ساختار کل سیستم مشخص نشده تعیین دقیق نوع این شبکه‌ها معمولاً امکان‌پذیر نیست.

نتیجه این که در ساختار PLC فضای کافی برای کارت شبکه منظور شود و حداقل یک کارت شبکه اترنت نیز دیده شود. با توجه به نتایجی که نهایتاً در انتهای طراحی به‌دست می‌آید، می‌توان این مرحله را مجدداً بازنگری و اصلاح نمود.

۳-۹ انتخاب CPU

یکی از مهمترین قدم‌های فاز طراحی انتخاب CPU مناسب است. آنچه مسلم است این است که انتخاب صحیح نیاز به چندین بار بررسی و بازنگری دارد. توصیه می‌شود که وقتی تمام قدم‌های فاز طراحی به اتمام رسید باز به این قدم برگردید و از انتخاب صحیح خود مطمئن شوید. انتخاب نادرست CPU یا انتخابی که مبتنی بر آینده‌نگری نباشد، مشکلات زیادی را در آینده برای سیستم ایجاد خواهد کرد و هزینه‌های زیادی را از این بابت تحمیل خواهد نمود.

فاکتورهای اولیه‌ای که در انتخاب CPU معمولاً به آنها توجه می‌شود عبارتند از:

- تعداد ورودی و خروجی (I/O)
- میزان حافظه
- سرعت پردازش

و به‌طور دقیق‌تر بایستی فاکتورهای زیر را نیز به آن اضافه کرد:

- تعداد ورودی و خروجی‌های خاص
- تعداد پورت یا کارت شبکه مورد نیاز
- امکان افزایش حافظه
- امکان توسعه یا ارتقا سخت‌افزار سیستم

و به‌طور خاص:

- امکان افزونگی
- امکان Fail Safe

معمولاً در این مرحله یک یا چند سازندهPLC انتخاب می‌شوند و محصولات آنها با توجه به توانایی‌های مورد انتظار مقایسه می‌شوند. ولی بایستی دقت نمود که بررسی در این مرحله بایستی به‌صورت فنی اقتصادی و با ملاحظه همه شرایط انجام شود. اگر از نظر فنی CPU سازنده مورد نظر همه انتظارات را پوشش می‌دهد باز بایستی به فاکتورهای دیگر نیز توجه نمود مانند:

- سطح پشتیبانی فنی قابل ارائه توسط سازنده در محل یا از راه دور
- امکان دسترسی به قطعات یدکی در زمان کم
- امکان آموزش
- مستندات و نقشه‌ها و مدارک راهنما
- پشتیبانی در مورد ارتقا نرم‌افزارهای سیستم
- مدت زمان گارانتی و وارانتی سیستم

ممکن است با در نظر گرفتن این فاکتورها برخی سازندگان از لیست انتخاب حذف شوند. به هر حال پس از نهایی‌شدن لیست اسامی سازندگان، به بررسی و انتخاب CPU مورد نظر در بین محصولات آنها می‌پردازیم.

اولین فاکتور مهم در انتخاب CPU مناسب، تعداد ورودی و خروجی‌هاست. این تعداد لازم است با احتساب توسعه یعنی حداقل ۲۰ درصد اضافه جمع‌بندی شده باشد. با توجه به جمع کل I/O مورد نیاز می‌توان CPU را به‌صورت اولیه انتخاب نمود. به‌طور کلی CPUها نیز براساس تعداد I/O به نوع کوچک و متوسط و بزرگ تقسیم می‌شوند.

جدول ۳-۵ تقسیم‌بندی CPU براساس تعداد I/O

حدود تعداد I/O	سایز CPU
<= 100	کوچک
<=1000	متوسط
>1000	بزرگ

این تقسیم‌بندی به‌طور اتوماتیک سایر پارامترها را نیز تحت تأثیر قرار می‌دهد از جمله:

- CPUهای کوچک معمولاً یکپارچه و CPUهای بزرگ معمولاً ماژولار هستند.
- CPUهای کوچک حجم حافظه کم و CPUهای بزرگ حافظه زیاد دارند.
- در CPUهای کوچک معمولاً حافظه قابل توسعه نیست ولی درCPUهای بزرگ توسعه حافظه امکان‌پذیر است.
- CPUهای کوچک سرعت پردازش کمتری نسبت به CPUهای بزرگ دارند.

به‌عبارت دیگر وقتی طراح فقط با فاکتور تعداد ورودی و خروجی به انتخاب CPU می‌پردازد به‌طور خودکار سایر فاکتورها نیز پوشش داده می‌شود. یعنی CPUای که تعداد زیادی I/O را ساپورت می‌کند معمولاً از نظر سرعت پردازش نیز بالاست و حافظه آن نیز زیاد است و جای نگرانی چندانی وجود ندارد. با این وجود توصیه می‌شود که حافظه و سرعت و سایر امکانات نیز بررسی شود. بررسی پارامتر سرعت آنجایی اهمیت دارد که کاربر با توجه به منطق کنترل مورد نیاز فرآیند تا حدودی می‌تواند برآوردی از حجم برنامه داشته باشد. در این شرایط، کاربر می‌تواند محاسبه کند که اجرای یک سیکل‌اسکن توسط CPU چقدر طول می‌کشد و با توجه به سرعت تغییر ورودی‌ها یا زمان مورد نیاز برای فرمان به خروجی‌ها، آیا تاخیر سیکل‌اسکن مجاز است یا خیر؟ فاکتور دیگری که روی تأخیر سیکل‌اسکن تأثیر می‌گذارد ارتباطات شبکه است. معمولاً در انتهای سیکل‌اسکن زمانی نیز صرف ارتباطات CPU می‌شود که نهایتاً می‌تواند کند شدن بیشتر دریافت ورودی‌ها یا ارسال فرمان به خروجی‌ها را در بر داشته باشد.

زمان تأخیر یک مسئله نسبی است. در بسیاری از فرآیندها زمان کمتر از ۱۰۰ میلی‌ثانیه قابل قبول است ولی اگر شرایط فرآیند به‌گونه‌ای است که نیاز به فرمان سریع به برخی یا تمام خروجی‌ها در زمانی کمتر از ۱۰ میلی‌ثانیه وجود دارد یا نیاز باشد که ورودی‌ها در این زمان خوانده شوند، در این شرایط توصیه می‌شود که انتخاب CPU با دقت و وسواس بیشتری صورت گیرد.

۳-۱۰ انتخاب رک

انتخاب رک معمولاً در PLCهای نوع ماژولار مفهوم دارد. این موضوع در دو حالت زیر قابل بررسی است:

۱. وقتی سازنده چند نوع رک برای نصب کارت‌ها معرفی کرده باشد که از نظر تعداد اسلات متفاوت باشند.

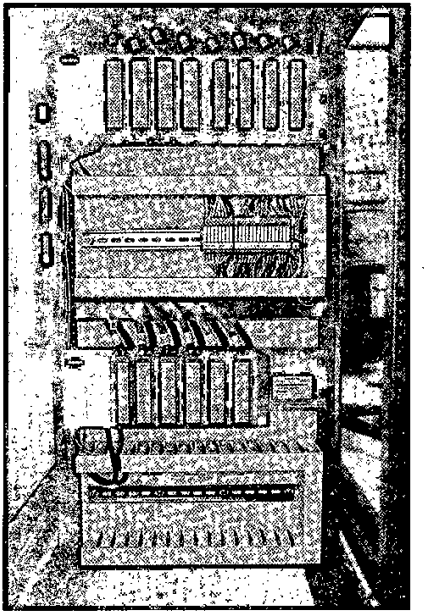
۲. وقتی تعداد کارت‌ها آنقدر زیاد باشد که روی یک رک قابل نصب نباشند و نیاز به رک اضافی باشد.

در حالت اول رک مورد نظر را متناسب با تعداد کارت‌ها و با در نظر گرفتن حداقل ۲ اسلات رزرو برای آینده انتخاب می کنیم.

در حالت دوم ابتدا بررسی می‌کنیم که آیا قرار دادن رک اضافی برای I/Oها از نظر فنی اقتصادی بهینه است یا بهتر است از Remote I/O استفاده شود.

لازم به ذکر است که برخی سازندگان برای بالا بردن قابلیت اطمینان سیستم، امکان نصب دو منبع تغذیه روی یک رک را فراهم ساخته‌اند که در این حالت رک مورد استفاده با رک‌های معمولی متفاوت می‌باشد.

فصل ۳



شکل ۳-۳ نمونه‌ای از نصب رک اضافی در پنل

۳-۱۱ انتخاب منبع تغذیه

برای انتخاب منبع تغذیه مناسب لازم است ولتاژهای قابل دسترس برای ورودی آن و نیز جریان مورد نیاز برای خروجی آن با توجه به مصارف ماژول‌ها از قبل مشخص شده باشد: در فاز طراحی، منبع تغذیه آخرین ماژولی است که انتخاب می‌شود. ابتدا باید لیست تمام ماژول‌های مورد نیاز را با در نظر گرفتن امکان توسعه سیستم مشخص نمود، سپس با توجه به بدترین

شرایط، منبع تغذیه را تعیین کرد. منظور از بدترین شرایط وضعیتی است که تمام خروجی‌های یک ماژول در وضعیت On (وصل) باشند یا تمام ورودی‌های یک ماژول ورودی در وضعیت On (وصل) قرار گیرند. ممکن است در حین کار چنین وضعیتی پیش نیاید ولی بایستی در طراحی دیده شود. به‌عنوان مثال اگر کنتاکتور راستگرد و چپگرد موتور به دو خروجی ماژول PLC متصل هستند هیچگاه این دو خروجی همزمان فعال نخواهند بود. با این وجود با توجه به احتمال تغییر سیم‌کشی و جابه‌جایی کانال‌ها بایستی فرض کرد که تمام کانال‌های کارت به‌طور همزمان فعال شوند. برای درک بهتر موضوع به مثال ۳-۱ توجه کنید.

مثال ۳-۱: در یک فرآیند، تعداد ۵۰ سیگنال دیجیتال ورودی و تعداد ۳۰ سیگنال دیجیتال خروجی وجود دارد. هر ماژول ورودی می‌تواند ۱۶ ورودی را بگیرد و هر ماژول خروجی می‌تواند به ۸ خروجی متصل شود.

پس از انتخاب هر ماژول، میزان جریان آنرا در حالت ظرفیت کامل از کاتالوگ مربوطه استخراج کرده و با جریان مصرفی سایر ماژول‌ها جمع می‌زنیم. مقادیر ارائه شده در جدول ۳-۶ به‌صورت فرضی است.

جریان مصرفی ماژول نباید به اشتباه بر اساس ولتاژ 5 VDC در محاسبات وارد شود. با توجه به اینکه منبع تغذیه 24 VDC است جریان ماژول بایستی بر مبنای 24V DC محاسبه گردد.

جدول ۳-۶ محاسبه جریان مصرفی ماژول های PLC

	تعداد سیگنال	تعداد ماژول مورد نیاز	جریان هر ماژول در حالت بار کامل	جریان کل ماژول‌ها
ورودی دیجیتال	50	4	10mA	40 mA
خروجی دیجیتال	30	4	200 mA	800 mA
خروجی آنالوگ	20	3	**200 mA**	**600 mA**
			جمع جریان ماژول‌ها	**1440 mA**
			جریان جذبی CPU	**100 mA**
			جمع کل	**1540 mA**

با توجه به محاسبات انجام شده در جدول ۳-۶، مشخص است‌که کل جریان مصرفی PLC برابر 1.5 A می‌باشد. بنابراین حداقل یک منبع تغذیه 2A مورد نیاز خواهد بود.

لازم است توجه شود که:

- مقادیر فوق فقط برای مصارف ماژول‌هاست. برای اتصال خروجی‌ها نیاز به محاسبات دیگر نیز وجود دارد. تأمین جریان خروجی‌ها بعضاً می‌تواند از منبع تغذیه جداگانه باشد.
- در شرایطی که تجهیزات بزرگ تولیدکننده نویز روی خط AC وجود دارند، بهتر است بین ورودی AC و منبع تغذیه، ترانس ایزوله به‌کار رود. این ترانس بایستی دارای ولت آمپر (VA) مناسب، متناسب با بار باشد. توصیه می‌شود که پس از محاسبه توان مصرفی، ظرفیت ترانس را ۲۵ درصد بیشتر انتخاب کنید.

۳-۱۲ انتخاب پنل PLC

انتخاب پنل PLC بایستی مطابق با استاندارد باشد. NEMA یکی از استانداردهای متداول در این زمینه است. بخش‌هایی از این استاندارد که مرتبط با سیستم‌های کنترل است موارد زیر می‌باشد. این استانداردها در DVD همراه کتاب ارائه شده‌اند:

- ICS 1, General Standards for Industrial Control and Systems
- ICS 6, Enclosures for Industrial Control Systems

در این استاندارد پارامترهای مهم نظیر دما، رطوبت، لرزش، نویز و ایمنی تعریف شده است. جدول ۳-۷ انواع پنل تعریف شده در NEMA را نشان می‌دهد.

جدول ۳-۷ پنل‌های تعریف شده در استاندارد NEMA

NEMA Type	کاربرد
Type 1 (Surface Mount)	برای استفاده Indoor جایی که شرایط بهره‌برداری غیر معمول وجود ندارد.
Type 1 (Flush Mount)	شبیه نوع قبل ولی وقتی نصب پنل روی فریم ماشین مد نظر است.
Type 3	برای نصب به‌صورت Outdoor که حفاظت در مقابل وزش باد و گرد وخاک و ریزش باران و برف و یخ زدگی بیرونی مد نظر است.
Type 4	برای نصب Indoor یا Outdoor جایی که حفاظت در مقابل گرد و غبار و پاشش آب مد نظر است.
Type 4 X	مشابه Type4 به اضافه حفاظت در مقابل خوردگی
Type 7	برای نصب Indoor در شرایطی که گازهای خطرناک و قابل انفجار وجود دارند. این پنل در مقابل بروز جرقه‌های ناشی از قطع و وصل قطعات الکتریکی در اتمسفر اطراف پنل حفاظت دارد.
Type 9	برای نصب indoor به‌صورت حفاظت شده در مقابل ورود ذرات خطرناک و قابل اشتعال
Type 12	برای نصب indoorبه‌صورت حفاظت شده در مقابل ذرات و مایعات کثیف غیر خورنده
Type 14	برای نصب indoor به‌صورت حفاظت شده در مقابل ذرات و پاشش آب و روغن و Coolantهای غیر خورنده

نکات دیگری که در ارتباط با پنل PLC بایستی به آن توجه داشت عبارتند از:

- نصب پنل بایستی در موقعیتی باشد که درهای آن بتواند به‌طور کامل برای دسترسی‌های لازم باز شود.
- عمق پنل به‌صورتی باشد که پس از بسته شدن در فاصله کافی بین آن و قطعات داخل پنل وجود داشته باشد.
- پنل بایستی مجهز به کلید قطع اضطراری باشد. این کلید در نقطه‌ای روی درب پنل نصب شود که دسترسی به آن آسان باشد.
- پنل در فاصله مناسب نسبت به تجهیزات تولید کننده نویز نصب شود. از جمله این تجهیزات می‌توان به دستگاههای جوشکاری ، کوره‌های القایی ودرایوها و موتورهای بزرگ اشاره کرد.
- موقعیت نصب پنل دارای لرزش بیش از حد مشخص شده توسط سازنده نباشد.
- سیستم خنک کنندگی پنل به‌صورتی باشد که دمای داخل آن از ماکزیمم حد مجاز که معمولاً ۶۰ درجه است فراتر نرود. نوع تهویه با توجه به توان جذب شده توسط اجزا می‌تواند مشخص شود. معمولاً حدود ۳۰ درصد از توان خروجی منبع تغذیه به گرما تبدیل می‌شود. بهترین آن نوع heat exchanger مدار بسته است.

خنک کنندگی پنل

موضوع خنک‌کنندگی پنل یکی از نکات قابل توجه در فاز طراحی است که روی انتخاب ابعاد پنل، فاصله‌ی بین اجزا و نوع سیستم خنک‌کنندگی تأثیر دارد.

به‌طور کلی بر اساس درجه IP کابینت به دو دسته تقسیم می‌شود :

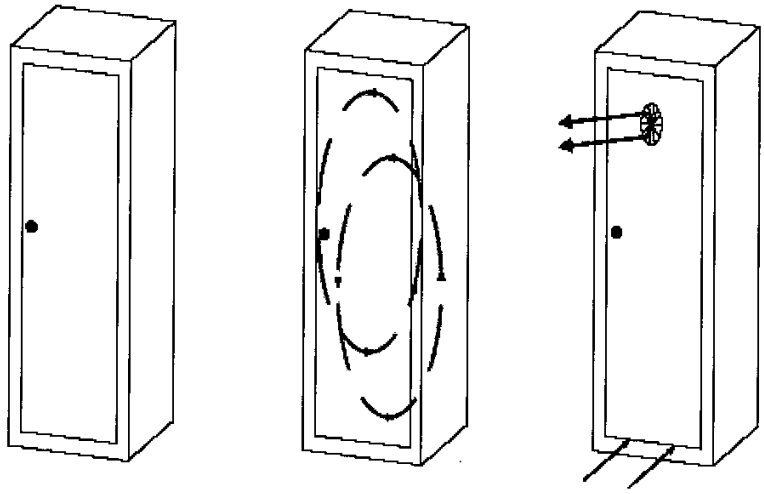
- **کابینت بسته:** که برای درجه حفاظتی بالا مانند IP54 استفاده می‌شوند.
- **کابینت باز:** که با درجه حفاظتی پایین مانند IP20 هستند.

خنک‌کنندگی در کابینت نوع بسته به سه روش می‌تواند انجام گیرد:

۱. با گردش هوای طبیعی داخل پنل از طریق سطح فلزی دیواره‌های پنل. بدیهی است این روش برای پنل‌ها با اتلاف حرارتی کم قابل استفاده است.

۲. با گردش هوا توسط فن داخل پنل که تبادل حرارتی با سطوح فلزی را بالا می‌برد ولی برای اتلاف حرارتی بالا مناسب نیست.

۳. با استفاده از Heat Exchanger که برای پنل‌های با گرمای زیاد و IP بالا مناسب است.



شکل ۳-۴ روش‌های خنک‌کنندگی در پنل‌های کاملاً بسته

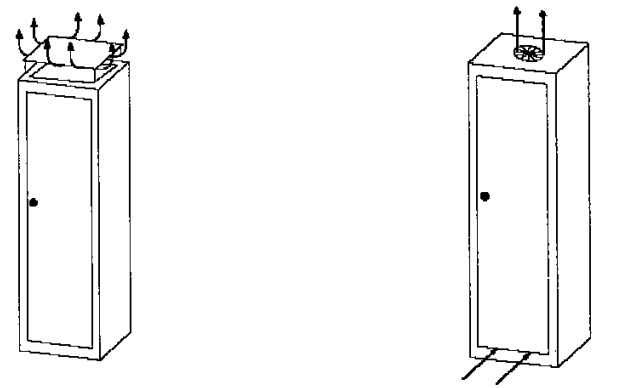
با توجه به شکل ۳-۵، خنک‌کنندگی در کابینت نوع باز به دو روش می‌تواند انجام گیرد:

۱. خنک‌کنندگی طبیعی از طریق قسمت باز بالای پنل

۲. خنک‌کنندگی از طریق فن نصب شده در بالای پنل

به‌طور معمول سیستم‌های خنک‌کننده داخل پنل، هوای خنک را از سمت پایین پنل به بالا می‌کشند و با عبور هوا از داخل یا کنار اجزای گرم‌شده، گرما را جذب و از پنل خارج می‌نمایند.

در هر دو روش فوق لازم است دریچه‌ای برای امکان ورودی هوا به پنل از پایین وجود داشته باشد. این دریچه‌ها به فیلتر مجهز هستند تا گرد و غبار به داخل پنل مکش نشود.



شکل ۳-۵ روش‌های خنک کنندگی در پنل‌های باز

عملکرد سیستم خنک کننده بایستی به‌گونه‌ای باشد که دمای داخل پنل از ۶۰ درجه سانتی‌گراد بالاتر نرود. در محاسبات شرایط زیر را به‌عنوان پیش‌فرض در نظر می‌گیرند:

- در هر لحظه فقط ۶۰ درصد ورودی‌ها On هستند.
- در هر لحظه فقط ۳۰ درصد خروجی‌ها On هستند.
- دمای هوای خنک‌کننده ورودی ماکزیمم به ۴۰ درجه ممکن است برسد.

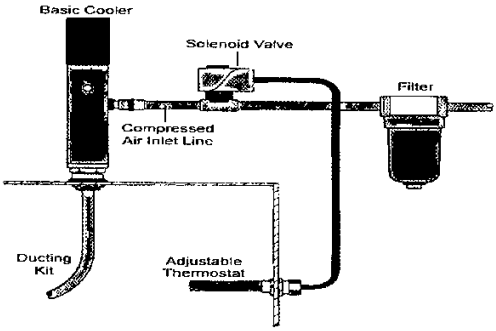
حالتی‌که تمام ورودی و خروجی‌ها on باشند و دمای هوا از ۴۰ درجه بیشتر باشد، یک حالت خاص محسوب شده و به‌عنوان شرایط معمول در نظر گرفته نمی‌شود. اگر احتمال بروز این حالت خاص باشد لازم است به‌منظور خنک‌کنندگی بیشتر، فضای بیشتری بین اجزا منظور شده و فن مناسب برای ایجاد خنک‌کنندگی در این شرایط نصب گردد.

دبی هوایی که برای خنک‌کنندگی بایستی از پنل عبور کند، بر اساس میزان گرمای تولید شده در پنل محاسبه می‌گردد.

هوای ورودی که توسط فن به داخل پنل کشیده می‌شود، ابتدا بایستی از فیلتر بگذرد تا ذرات ریز معلق در هوا به پنل وارد نشوند. ورود گرد و غبار به داخل پنل و نشستن آن روی قطعات باعث می‌شود که خنک‌کنندگی و تخلیه گرمایی آن‌ها بخوبی انجام نشود. بنابراین باز گذاشتن درب پنل برای خنک‌کنندگی بیشتر روش درستی نخواهد بود.

در شرایط گرمای شدید به‌ویژه اگر رطوبت نیز بالا باشد، به‌جای فن بایستی از سیستم‌های Air Condition تهویه مطبوع استفاده نمود که هوای خنک و تمیز را از داخل پنل عبور دهند.

در برخی شرایط نیز از هوای فشرده تمیز برای خنک‌کنندگی در شرایط خاص استفاده می‌کنند. شکل ۳-۶ این طرح را نشان می‌دهد. در صورتی‌که ترموسوئیچ داخل پنل عمل کند، مسیر هوای ورودی به پنل باز می‌شود.



شکل ۳-۶ خنک‌کنندگی پنل با هوای فشرده فیلتر شده

۳-۱۳ کلید اضطراری

کلید قطع اضطراری به تعداد کافی در نقاط مختلف فیلد و اتاق کنترل در نظر گرفته شود. کلید قطع اضطراری نباید با PLC سری شود بلکه بایستی مستقیماً و بدون واسطه بتواند خروجی‌ها را غیر فعال کند. معمولاً علاوه بر کلید قطع اضطراری، یک رله MCR [1] نیز در نظر گرفته می‌شود که در شرایط خاص بتواند فقط تغذیه خروجی‌ها یا تغذیه ورودی‌ها را قطع کند بدون اینکه خود PLC بی‌برق شود. توضیحات بیشتر در این زمینه در فصل‌های بعد آمده است.

۳-۱۴ انتخاب کابل با طول مناسب

کابل‌های I/O و همچنین کابل‌های شبکه بایستی با طول مشخصی کشیده شوند. افزایش بیش از حد طول، منجر به مشکل در دریافت یا ارسال I/O توسط PLC خواهد شد.

کابل I/O

اگر کابل‌کشی از فیلد به PLC به‌صورت Direct و بدون شبکه باشد، معمول این است که از هر سنسور یا عملگر کابل مجزایی با تعداد رشته مناسب (که برای بیشتر سیگنال‌ها دو رشته است) کشیده می‌شود. این کابل مستقیماً به پنل PLC متصل نمی‌شود بلکه ابتدا وارد جعبه ترمینالی که در فیلد نصب شده می‌گردد و سپس از آنجا توسط کابل‌های چند زوج [2] به پنل PLC کشیده می‌شود. این موضوع طرح را اقتصادی‌تر خواهد کرد. برای پرهیز از پدیده‌ی تداخل، توصیه می‌شود که سیگنال‌هایی با یک سطح ولتاژ کنار هم از یک کابل چند زوج عبور کنند. توضیحات بیشتر در این خصوص در فصل بعد آمده است.

1. Master Control Relay
2. Multicore

برای محاسبه حداکثر طول کابل لازم است به فاکتورهای زیر توجه شود:

- امپدانس کابل چقدر است؟
- جریان عبوری از کابل چقدر است؟
- حداقل ولتاژ معتبر برای کانال کارت یا وسیله چقدر است؟

با داشتن اطلاعات فوق می‌توان حداکثر طول کابل را محاسبه نمود.

مثال ۳-۲ : یک سنسور توسط کابل $0.5\ mm^2$ به کارت DI 24VDC متصل شده است. طبق مشخصات کارت حداقل ولتاژ برای شناسایی وضعیت ON برابر 13 VDc است. اگر جریان عبوری 50mA باشد و امپدانس کابل 0.25 ohm/m باشد (با فرض اینکه ترمینال‌ها هیچ مقاومتی ندارند) طول کابل چقدر می‌تواند باشد؟

حل: با توجه به اینکه منبع تغذیه 24 VDC است و با توجه به اینکه خود سنسور نیز افت ولتاژی در حد ۲ تا ۳ ولت ایجاد می‌کند، حداکثر افت ولتاژ مجاز روی خط برابر است با:

$$24 - 13 - 3 = 9\ V$$

و ماکزیمم مقاومت قابل قبول در مسیر برابر است با:

$$R = V / I \qquad R = 9 / 0.05 = 180\ Ohm$$

با توجه به امپدانس کابل، حداکثر طول مجاز کابل برابر است با:

$$180 / 0.25 = 720\ m$$

این طول، حداکثر طول است و نباید به‌عنوان عدد نرمال منظور شود. بهتر است حدود ۲۵ درصد از این طول را کاهش داده و به‌عنوان مبنای حداکثر قرار داد یعنی در این مثال حدود ۵۴۰ متر.

مثال ۳-۳ : یک سلونوئید ولو که با حداقل ولتاژ 22VDC کار می‌کند، برای تحریک نیاز به جریان حداقل 0.5A دارد. کابل به‌کار رفته $1.5\ mm^2$ است که مقاومت آن $14\ ohm / km$ می باشد. حداکثر طول مجاز کابل چقدر است؟

حداکثر افت مجاز روی کابل برابر است با:

$$24 - 22 = 2\ V$$

حداکثر مقاومت مجاز مسیر برابر است با:

$$2\ V / 0.5\ A = 4\ ohm$$

حداکثر طول کابل برابر است با:

$$4 / 14 = 0.286\ km = 286\ m$$

استفاده از Remote I/O

محاسبات کابلی که از سنسور تا R.I/O کشیده می‌شود مشابه محاسبات فوق است، ولی طول کابل شبکه که از R.I/O تا PLC کشیده می‌شود بایستی با توجه به نوع شبکه محاسبه شود. در برخی شبکه‌ها طول کابل شبکه تابع سرعت انتقال دیتاست، هرقدر سرعت بالاتر باشد طول کمتری قابل استفاده خواهد بود. به‌عنوان مثال در شبکه پروفی‌باس در سرعت

حداکثر (12Mbps)، حداکثر طول کابل ۱۰۰ متر است که با استفاده از چند تقویت کننده تا حداکثر ۱۰۰۰ متر قابل افزایش است. اگر به‌جای کابل مسی از فیبر نوری بین R.I/O تا PLC استفاده شود این طول به چندین کیلومتر می‌رسد.
با توجه به تنوع زیاد شبکه‌های فیلد باس، طول کابل شبکه را بایستی پس از انتخاب نوع شبکه و با توجه به مشخصات فنی آن شبکه انتخاب نمود.

۳-۱۵ حفاظت در مقابل نویز

نویز یکی از عوامل مزاحم در سیستم‌های کنترل است. نویز به‌ندرت باعث آسیب به کارت‌ها و سخت‌افزار PLC می‌شود. وجود نویز موجب می‌شود که کنترلر یک سیگنال را اشتباه بخواند یا فرمان کنترل با تغییر به خروجی اعمال شود. به‌طور کلی هر عاملی که موجب ایجاد تغییر در سیگنال‌های ورودی/خروجی شود را نویز می‌نامند. نویز عموماً در اثر وجود میدان‌های مغناطیسی ناخواسته در اطراف سیم‌های حامل سیگنال الکتریکی به‌وجود آمده و بر سیگنال الکتریکی تأثیر می‌گذارد. سیگنال‌های آنالوگ نسبت به نویز حساس‌تر از سیگنال‌های دیجیتال می باشند.
دروازه‌های ورود نویز به کنترلر می‌تواند موارد زیر باشد:

- ورودی AC منبع تغذیه
- ورودی و خروجی‌های دیجیتال و آنالوگ
- شبکه

حتی اگر پنل PLC دور از منابع نویز قرار گیرد، باز نویز می‌تواند از طریق کابل‌های متصل به PLC وارد سیستم شود. برای ورودی AC، منبع تغذیه بایستی مجهز به فیلترهای حذف نویز باشد. برای سایر انواع نویز، نکات مهمی در ارتباط با کابل‌کشی برای مقابله با نویز قابل توجه است مانند:

- استفاده از کابل مناسب (شیلد دار)
- استفاده از مسیر کابل مناسب (سینی‌گذاری و لوله‌کشی)
- استفاده از سیستم زمین مناسب

اگر در فاز طراحی نکات فوق به‌خوبی دیده نشود، مثلاً کابل‌های سیگنال و کابل‌های قدرت در یک مسیر به موازات هم و کنار هم کشیده شوند یا سیستم زمین مناسب نباشد، سیستم در حین بهره‌برداری با مشکلات زیادی مواجه خواهد شد. اشکالات گذرا که عیب‌یابی آنها بسیار دشوار است، معمولاً از منابعی مانند نویز ناشی می‌شوند.
توصیه‌های زیر در ارتباط با محدودسازی نویز قابل توجه است:

- از منبع تغذیه مناسب که مجهز به ایزوله‌سازی و فیلتر است استفاده کنید تا نویزهای سمت تغذیه AC به کنترلر و ورودی و خروجی‌های آن منتقل نشود.
- محل نصب تجهیزات را با بیشترین فاصله ممکن نسبت به منابع تولیدکننده نویز مانند موتورهای بزرگ و کوره‌های الکتریکی و امثال آنها طراحی کنید.
- از کابل شیلددار استفاده کنید که یک سمت آن زمین شود.
- مسیر کابل‌ها را به‌درستی طراحی کنید. اگر مسیر کابل‌های سیگنال و کابل‌های قدرت یکسان است، آنها را روی سینی‌های مجزا قرار داده و فاصله مجاز بین سینی‌ها را رعایت نمایید.

- دو سیستم زمین، یکی برای PE و دیگری برای CE در نظر بگیرید که اگر چه هر دو نهایتاً به یک شبکه زمین متصل هستند ولی دو مسیر مجزا دارند. تجهیزات کنترل به زمین CE متصل می‌شوند.

۳-۱۶ پرسش و تحقیق

اگر در فرآیندی نیازهای زیر وجود داشته باشد، از بین چند سازنده معتبر PLC کدامیک را نهایتاً انتخاب و چه سیستمی را برای این منظور پیشنهاد می‌کنید؟ مشخصات اجزای اصلی را تعیین کنید.

- تعداد ۱۰۰۰ سیگنال آنالوگ ورودی
- ۲۰ سیگنال آنالوگ خروجی
- ۲۰۰۰ دیجیتال ورودی
- ۵۰۰ دیجیتال خروجی
- تعداد ۲۰ لوپ کنترلی
- زمان خواندن ورودی‌ها و ارسال فرمان به خروجی‌ها کمتر از ۵۰ میلی‌ثانیه
- هیچ ورودی و خروجی خاصی مورد نیاز نیست.
- Remote I/O استفاده نمی‌شود.

فصل ۳

۳-۱۷ تست‌های خودآزمایی

۱- در کدام‌یک از سیستم‌های زیر، طراحی و پیاده‌سازی سیستم کنترل آسان‌تر است؟

الف) فرآیندی که با سیستم کنترل قدیمی در حال کار است. ب)فرآیند جدیدی که قرار است از ابتدا پیاده‌سازی شود.

ج) فرآیندی که ترکیبی از موارد فوق باشد. د) موارد الف و ب

۲- به‌منظور طراحی یک سیستم کنترل، قدم اول چیست؟

الف) دستیابی به شرح عملکرد سیستم ب) طراحی نقشه‌ها ج) برنامه‌نویسی د) انتخاب قطعات

۳- مدارک P&ID بیان‌کننده چه اطلاعاتی راجع به فرآیند می‌باشند؟

الف) شرح فرآیند ب) شرح وسایل ابزار دقیق مورد نیاز

ج) روند پروسه د) موارد الف و ب

۴- علامت PT در نقشه‌های P&ID بیانگر کدام مورد زیر است؟

الف) ترانسمیتر سطح ب) ترانسمیتر فشار ج) ترانسمیتر دما د) ترانسمیتر فلو

۵- انتخاب کارت‌های I/O با توجه به کدام مورد زیر انجام می‌پذیرد؟

الف) نقشه‌های P&ID ب) شرح پروسه ج) لیست I/O د) هیچ‌کدام

۶- انتخاب منبع تغذیه بر اساس کدام حالت صورت می‌پذیرد؟

الف) حالتی‌که همه‌ی ورودی‌ها وصل باشند. ب)حالتی‌که همه‌ی خروجی‌ها وصل باشند.

ج) حالتی‌که نصف ورودی/خروجی‌ها وصل باشند. د) حالتی‌که همه‌ی ورودی/خروجی‌ها وصل باشند.

۷- وقتی نیاز به سرعت و اطمینان بالا وجود داشته باشد، کارت‌های I/O :

الف) روی R.IO در فیلد قرار می‌گیرند. ب)روی R.IO در کنار PLC قرار می‌گیرند.

ج) روی خود PLC قرار می‌گیرند. د) موارد ب و ج

۸- امروزه استفاده از کدام کارت شبکه در کنار PLC متداول‌تر است؟

الف) کارت شبکه‌ی پروفی‌باس ب)کارت شبکه‌ی مدباس

ج) کارت شبکه ASI د) کارت شبکه اترنت صنعتی

۹- کدام مورد زیر در انتخاب CPU نقش مهمی ندارد؟

الف) تعداد I/O ها ب) ولتاژ I/O ها ج) سرعت پردازش د) حافظه‌ی مورد نیاز

۱۰- در مورد سیستم خنک‌کنندگی پنل PLC کدام مورد صحیح است؟

الف) از ورود گرد و غبار به داخل پنل جلوگیری نماید.

ب) دمای داخل پنل از ۷۰ درجه بیشتر نشود.

ج) در صورت نیاز، به‌منظور خنک‌سازی بتوان درب پنل را باز نمود.

د) هر سه مورد

۱۱- در باره نقش کلید قطع اضطراری کدام مورد صحیح است؟

الف) این کلید باید بتواند فرمان قطع به PLC بدهد.

ب) این کلید باید بتواند ورودی‌ها را مستقل از PLC قطع کند.

ج) این کلید باید بتواند ورودی و خروجی‌ها را مستقل از PLC قطع کند.

د) این کلید باید بتواند ورودی و خروجی‌ها را همراه با PLC قطع کند.

۱۲- کدام مورد در محاسبه‌ی طول کابل تأثیرگذار است؟

الف) امپدانس کابل ب) افت ولتاژ روی کابل ج) جریان عبوری از کابل د) هر سه مورد

۱۳- در مورد نویز کدام مورد صحیح نیست؟

الف) سیگنال‌های الکتریکی را تحت تأثیر قرار می‌دهد.

ب) باعث اشتباه در خواندن مقادیر سیگنال‌ها توسط CPU می‌گردد.

ج) تأثیر آن بر سیگنال‌های دیجیتال نسبت به سیگنال‌های آنالوگ بیشتر است.

د) در اثر وجود میدان‌های مغناطیسی اطراف سیم‌های انتقال سیگنال به‌وجود می‌آید.

۱۴- برای حذف نویز کدام مورد باید انجام پذیرد؟

الف) استفاده از کابل شیلددار ب) استفاده از زمین حفاظتی (PE)

ج) سینی‌گذاری و لوله‌کشی د) هر سه مورد

فصل ۴

اصول نصب PLC

۴-۱ مقدمه

۴- ۲ نکات نصب پنل PLC

۴-۳ نکات نصب منبع تغذیه

۴-۴ نکات نصب اجزای PLC

۴-۴-۱ نکات کلی نصب

۴-۴-۲ نصب کارت‌های ورودی و خروجی

۴-۴-۳ نکات مربوط به تغذیه I/Oها

۴-۵ نصب تجهیزات حفاظتی

۴-۶ سیم‌کشی

۴-۷ زمین‌کردن

۴-۸ نصب کلیدهای Emergency

۴-۹ نصب MCR

۴-۱۰ پرسش و تحقیق

۴-۱۱ تست‌های خودآزمایی

مطالعه این بخش برای افرادی که در کار نصب و اجرای پروژه‌های اتوماسیون فعالیت دارند مفید خواهد بود ولی بایستی به این نکته مهم توجه شود که مطالب این بخش حاوی توصیه‌های کلی در ارتباط با نصب PLC است. ضرورت دارد برای هر PLC خاص کاربر طبق دستورالعمل سازنده نصب PLC را انجام دهد.

چکیده مطالب

- در نصب پنل دور بودن از ماشین و فرآیند و منابع نویز و نصب در شرایط آب و هوایی مناسب و موقعیتی فاقد لرزش در نظر گرفته شود.
- بهترین محل برای نصب منبع تغذیه بالای پنل است تا گرمای ناشی از آن مستقیماً و بدون تاثیر روی سایر تجهیزات از پنل خارج شود.
- PLC در موقعیتی در پنل نصب شود که دسترسی به آن برای عیب‌یابی ساده باشد.
- تجهیزات PLC به شکلی نصب شوند که خنک‌کنندگی آنها به‌درستی انجام گیرد.
- در اطراف PLC فضای کافی برای امکان خنک‌کنندگی بهتر در نظر گرفته شود.
- بین کارت‌های I/O و داکت‌ها فاصله مناسب برای سهولت سیم‌کشی منظور شود.
- در کارت‌های I/O نوع AC تغذیه I/O با تغذیه AC ورودی منبع تغذیه مشترک باشد.
- در کارت های I/O نوع DC تغذیه I/O از تغذیه DC خروجی منبع تغذیه مجزا باشد.
- در مسیر تغذیه ورودی و خروجی‌ها حفاظت و موج‌گیر قرار دهید.
- در خروجی‌هایی که برای بارهای AC کوچک متصل هستند حفاظت اسنوبر کافیست.
- در خروجی‌هایی که به بارهای AC بزرگ متصل هستند علاوه بر اسنوبر از MOV استفاده شود.
- در خروجی‌هایی که به بارهای DC متصل هستند دیود معکوس به‌کار ببرید.
- در انجام سیم‌کشی ورودی‌ها و خروجی‌ها را از یکدیگر جدا کنید و ترجیحاً نوع دیجیتال و آنالوگ را نیز از یکدیگر تفکیک نمایید.
- بین کابل‌های سیگنال و کابل‌های قدرت فاصله قرار دهید.
- شیلد کابل را در ورودی پنل PLC زمین کنید.
- از کلیدهای قطع اضطراری و رله‌های MCR به تعداد کافی استفاده کنید.

فصل ۴

اصطلاحات و تعاریف

Duct

مسیری که در داخل پنل برای عبور کار تعبیه می شود داکت نام دارد .

Arrester

موج‌گیر است که به موازت مدار نصب شده و اجازه عبور اضافه ولتاژهای گذرا را نمی‌دهد.

Crosstalk

تأثیر میدان مغناطیسی یک زوج سیم روی زوج سیم دیگر Crosstalk گفته می‌شود.

Pull Cord

سیمی که به‌موازات نوار نقاله‌ها کشیده می‌شود در شرایط اضطراری با کشیدن این سیم نوار متوقف می‌شود.

۴-۱ مقدمه

در این فصل نکات کلی مرتبط با نصب PLC مورد توجه قرار می‌گیرد.
ممکن است پنل PLC به‌صورت مونتاژ شده سفارش داده شده باشد و صرفاً نیاز به سیم‌کشی داشته باشد، ولی آنچه در این فصل مورد توجه قرار گرفته این است که قطعات و اجزا به‌صورت مجزا تهیه شده و نیاز به مونتاژ ، نصب و سیم‌کشی وجود دارد. در این فصل به نکات نصب در ارتباط با موارد زیر می‌پردازیم:

- نصب پنل PLC
- نصب PLC
- نصب I/Oها و ترمینال‌های آنها
- نصب منبع تغذیه
- نصب حفاظت‌ها
- سیم‌کشی
- اتصال زمین

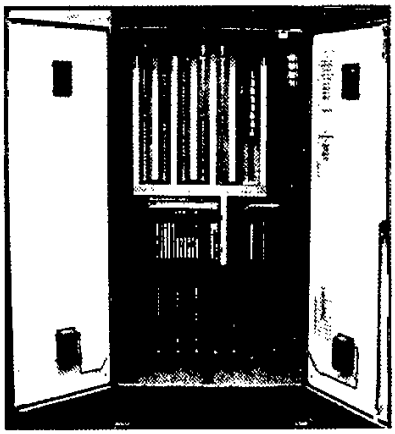
فصل ۴

۴-۲ نکات نصب پنل PLC

در فاز طراحی لازم است نوع پنل متناسب با شرایط نصب، به‌درستی انتخاب شده باشد. با این وجود لازم است قبل از نصب پنل به نکات زیر توجه شود:

- نصب پنل بایستی در موقعیتی باشد که درهای آن بتواند به‌طور کامل برای دسترسی‌های لازم باز شود.
- در محل نصب، لرزش و ارتعاش و شوک‌های دائمی وجود نداشته باشد.
- شرایط محل نصب متفاوت با طراحی پنل نباشد، به‌عنوان مثال پنلی که برای نصب indoor طراحی شده را به‌صورت Outdoor نصب نکنید. از اینرو قبل از نصب پنل شرایط محیط از نظر گرد و غبار، رطوبت، پاشش آب، درجه حرارت، وجود گازهای خورنده، وجود گازهای قابل اشتعال و ... را بررسی نموده و با مشخصات پنلی که آماده نصب است مطابقت دهید.
- پنل در معرض تابش مستقیم نور خورشید نباشد.
- پنل تا حد امکان دور از ماشین و فرآیند نصب شود.

پنل در فاصله مناسب نسبت به تجهیزات تولیدکننده نویز مانند موتورهای بزرگ، پنل تجهیزات فشار قوی، پنل درایو و امثال آنها نصب شود.



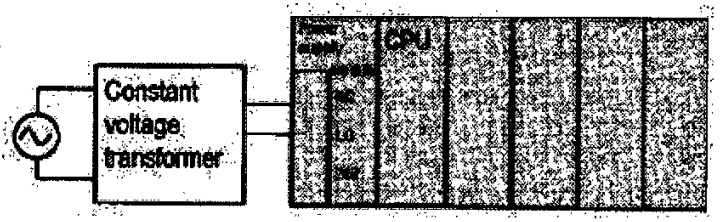
شکل ۴-۱ نمونه پنل PLC

- محل نصب پنل به‌گونه‌ای که دسترسی‌های غیر مجاز وجود نداشته باشد یا محدود شود.
- اگر روی در، سوئیچ و نشان دهنده نصب شده است، سیم‌های متصل به در از لوله قابل انعطاف عبور کنند.
- در صورتی که سیستم مانیتورینگ بر روی پنل نصب شده است، موقعیت آن برای کار اپراتوری به‌صورت ایستاده مناسب باشد.
- فاصله پنل از دیوار و همچنین فاصله پنل‌های مقابل نسبت به یکدیگر مناسب باشد.
- سیستم خنک‌کنندگی پنل مناسب باشد. این سیستم بایستی از گرمای غیرمجاز داخل جلوگیری کند در عین حال نباید با درجه IP مورد نیاز برای پنل تناقض داشته باشد. به‌عنوان مثال در IP54 نمی‌توان از دریچه و فن استفاده نمود.
- در قسمت ورودی کابل به کابینت طراحی به‌گونه‌ای باشد که بعد از اتمام کار نصب و کابل‌کشی منافذ ورودی کاملاً بسته شده و امکان ورود جانوران موذی همچون موش به داخل پنل نباشد.

۴-۳ نکات نصب منبع تغذیه

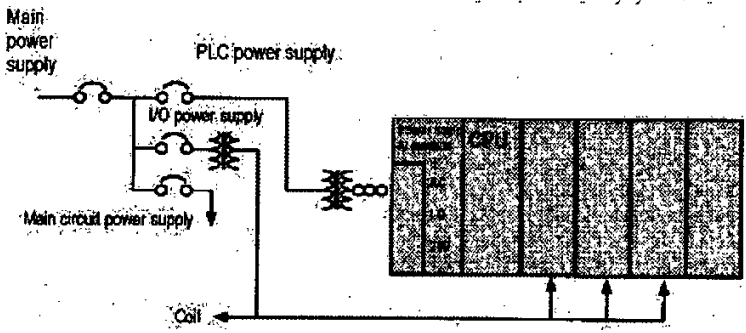
- منبع تغذیه بیش از بقیه اجزا گرما تولید می‌کند. از آنجا که جهت مسیر هوای خنک‌کننده رو به بالاست، منبع تغذیه نباید در زیر سایر تجهیزات PLC نصب شود. بهترین موقعیت برای نصب منبع تغذیه در بالای پنل است. فاصله بین منبع تغذیه و بالای پنل حداقل ۲۵ سانتی‌متر باشد. در عین حال منبع تغذیه می‌تواند در مجاور سایر اجزای PLC قرار گیرد.
- خطوط تغذیه ورودی، ترانس ایزوله و تثبیت ولتاژ، کلید اصلی قطع تغذیه ورودی و وسایل برقگیر تغذیه ورودی در بالای پنل و نزدیک به منبع تغذیه نصب شوند. موقعیت مناسب این تجهیزات که در کنار یکدیگر قرار می‌گیرند موجب می‌شود که خط AC که از بالا وارد پنل می‌شود کوتاه کشیده شود و در نتیجه تأثیر نویز حاصل از آن روی کنترلر به حداقل برسد.

- اگر برای خنک‌کنندگی پنل از فن استفاده شود، بایستی در نزدیکترین نقطه نسبت به تجهیزات تولیدکننده گرما یعنی در بالای پنل که منبع تغذیه قرار دارد نصب شود. از آنجا که فن هوای بیرون را به داخل پنل می‌کشد، لازم است در مسیر ورود هوا به پنل فیلترهایی تعبیه شود تا از ورود ذرات معلق در هوا به پنل جلوگیری کند.
- اگر نوسانات ولتاژ در تغذیه ورودی وجود دارد از ترانس تثبیت ولتاژ استفاده کنید.



شکل ۴-۲ استفاده از ترانس تثبیت ولتاژ

- منابع تغذیه از نوع سوئیچینگ، نویز تولید می‌کنند. بهتر است نوعی از منابع تغذیه که نویز کمتری دارند را به‌کار ببرید. اگر منبع تغذیه دارای نویز زیادی است، از ترانس ایزوله استفاده کنید.
- تغذیه PLC را از تغذیه I/O جدا کنید.



شکل ۴-۳ جداسازی تغذیه PLC از تغذیه I/O ها

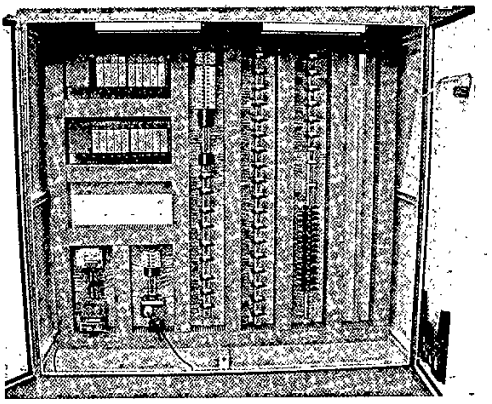
- ورودی AC منبع تغذیه به فیوز مناسب مجهز باشد.

۴-۴ نکات نصب اجزای PLC

۴-۴-۱ نکات کلی نصب

- در حین نصب و سیم‌کشی، روی PLC را بپوشانید تا قطعات و ذرات ناشی از سوراخکاری پنل و سیم‌کشی و... وارد آن نشوند.
- PLC را در پنلی که در آن تجهیزات فشار قوی نصب شده قرار ندهید.

- CPU در ارتفاع مناسبی نصب شود تا دسترسی به آن آسان باشد. این ارتفاع می‌تواند برای حالت کار ایستاده یا نشسته مطابق شکل ۴- ۴ باشد.

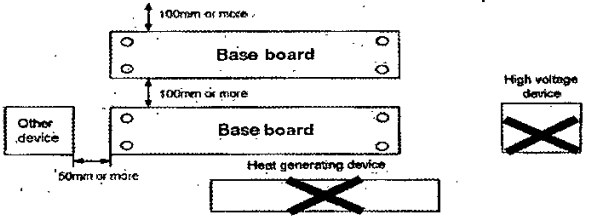


وضعیت نصب PLC برای کار در حالت ایستاده

وضعیت نصب PLC برای کار در حالت نشسته

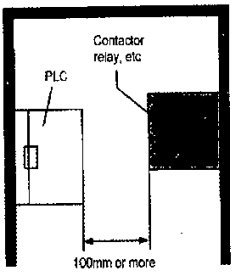
شکل ۴-۴ ارتفاع مناسب برای نصب PLC در پنل

- حداقل فضای بین PLC و داکت یا هر تجهیز دیگر حداقل ۵ سانتی‌متر باشد تا هم خنک‌کنندگی درست انجام شود و هم تعویض قطعات ساده‌تر باشد.
- اگر چند رک PLC در یک پنل نصب می‌شود یا وسایل دیگری که گرما و نویز تولید می‌کنند در یک پنل قرار می‌گیرند، حداقل بین آنها ۱۰ سانتی‌متر فضا قرار دهید.



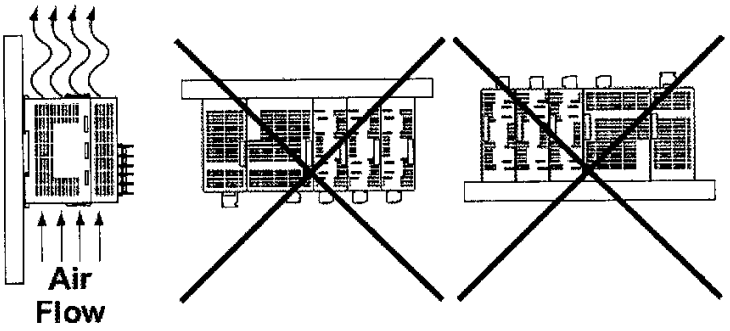
شکل ۴-۵ رعایت فاصله‌های مجاز بین PLC و دیگر تجهیزات

- PLC را در کنار کنتاکتورهای بزرگ که قطع و وصل آنها شوک و لرزش ایجاد می‌کند نصب نکنید؛ فاصله بین کنتاکتور و رله‌های متوسط و کوچک حدود ۱۰ سانتی‌متر باشد.



شکل ۴-۶ رعایت فاصله بین PLC و رله و کنتاکتورها

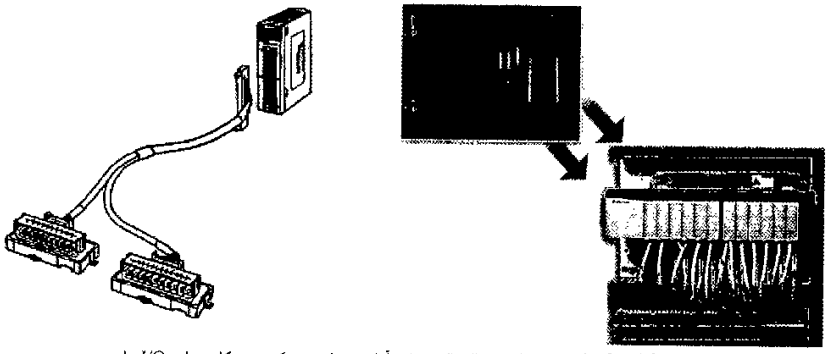
- نصبPLC به‌صورتی باشد که خنک‌کنندگی اجزا به‌خوبی انجام شود.



شکل ۴-۷ نصب غلط و نصب صحیح با توجه به جهت عبور هوای خنک کننده

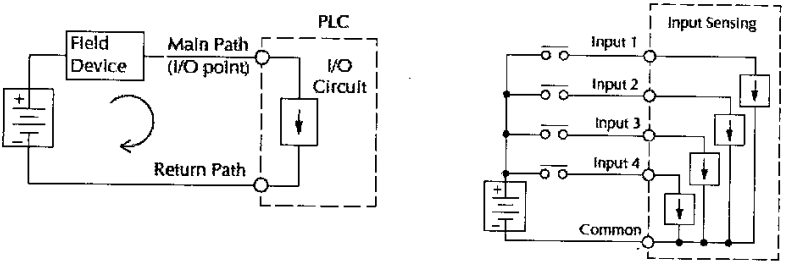
۴-۴-۲ نصب کارت‌های ورودی و خروجی

- فاصله بین کارت‌های I/O و داکت‌ها حداقل ۵ سانتی‌متر باشد.
- برای سهولت سیم‌کشی و عیب‌یابی بهتر است به‌جای سیم‌کشی مستقیم کارت‌های I/O از Terminal Blockهای آماده استفاده شود.



شکل ۴-۸ استفاده از ترمینال باکس‌های آماده به‌جای سیم‌کشی به کارت‌های I/O ها

- برای کاهش تأثیر نویز توصیه می‌شود سیم‌های ورودی و خروجی و نیز نوع DC و AC را به‌صورت مجزا دسته‌بندی کرده و از داکت عبور دهید. برای سهولت کار دسته‌بندی فوق، تا حد امکان کارت‌های مشابه را در کنار یکدیگر قرار داده و اگر ممکن است بین هر دو گروه یک اسلات خالی فاصله بیندازید.
- بهتر است برای هر I/O دو ترمینال در نظر گرفته شود. اگرچه در بسیاری کارت‌ها مسیر برگشت جریان مشترک است و در این حالت یک ترمینال برای هر I/O کافیست، ولی استفاده از کارت نوع اول و اتصال مجزا مشکلات کمتری از نظر نویز دارد.

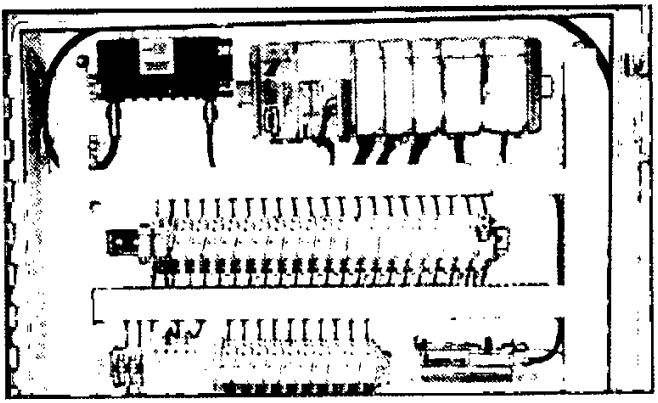


I/O های مجزا

I/O ها با Comm مشترک

شکل ۴-۹ اتصال ورودی‌ها با پایه مشترک و مجزا

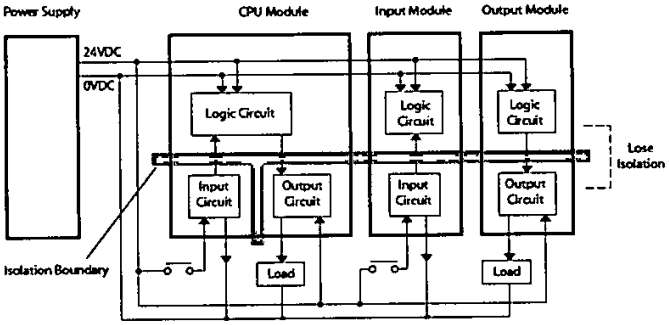
- فضای کافی بین ترمینال‌ها و داکت در نظر گرفته شود تا سیم‌کشی و عیب‌یابی ساده‌تر باشد.



شکل ۴-۱۰ رعایت فاصله بین داکت و ترمینال‌ها

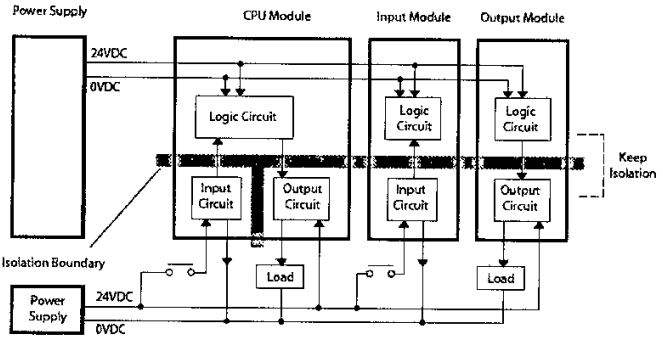
۴-۴-۳ نکات مربوط به تغذیه I/O ها

- تغذیه بخش Power و بخش Logic کارت‌های ورودی و خروجی DC از یکدیگر ایزوله باشند. در نوع AC این ایزوله‌سازی مورد نیاز نیست. در نوع DC اگر از یک منبع تغذیه برای دو بخش مزبور استفاده شود، ایزوله‌سازی از بین می‌رود.



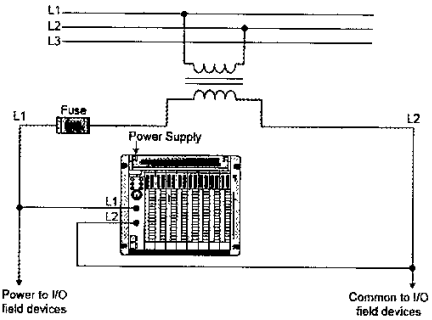
شکل ۴-۱۱ استفاده از تغذیه مشترک برای PLC و I/O های DC

با استفاده از دو منبع تغذیه ایزوله‌سازی برقرار خواهد شد.



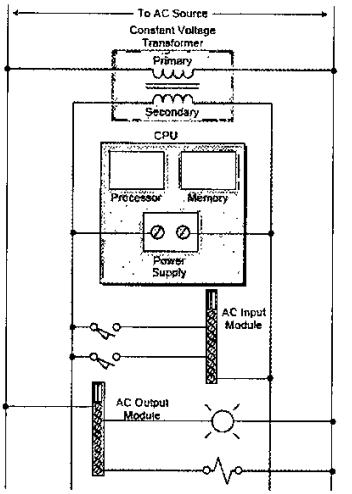
شکل ۴-۱۲ استفاده از تغذیه مجزا برای PLC و I/Oهای DC

- بهتر است تغذیه AC که به منبع تغذیه متصل می‌شود و تغذیه AC که برای سیگنال‌های I/O از نوع AC به‌کار می‌رود مشترک باشند تا هم تداخل سیگنال خطوط به حداقل برسد و هم از بروز فالت در سیگنال‌های ورودی اجتناب شود. وقتی سیگنال AC ورودی منبع تغذیه تثبیت شده ولی AC مربوط به I/Oها تثبیت نشده باشد، مشکل فوق پیش خواهد آمد. به هر حال وقتی AC Source مشترک باشد با این مسائل مواجه نخواهیم شد بعلاوه وقتی سطح ولتاژ ورودی AC از حد مجاز کمتر شود، منبع تغذیه می‌تواند به CPU سیگنال بدهد تا از خواندن سیگنال ورودی AC یا ارسال فرمان به خروجی AC اجتناب کند.



شکل ۴-۱۳ استفاده از تغذیه مشترک برای PLC و I/Oهای AC

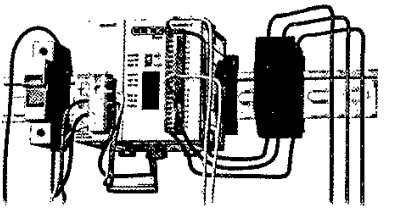
- در برخی شرایط که توان مورد نیاز I/Oهای AC بالاست، ممکن است در طرح فوق نیاز به ترانس با ولت آمپر بالا باشد. برای این منظور می‌توان از طرح نشان داده شده در شکل ۴-۱۳ که در آن I/Oها از سمت اولیه تغذیه می‌شوند استفاده کرد تا توان مورد نیاز برای ترانس کمتر باشد. اگر سطح سیگنال در سمت اولیه تغییرات زیادداشته باشد، این‌طرح مناسب نیست‌وبهتر است‌ازخروجی‌ترانس CVT به PLC و به I/O ها برق‌رسانی شود.



شکل ۴-۱۴ استفاده از تغذیه مشترک برای PLC و I/Oهای AC با اتصال مجزا به اولیه و ثانویه ترانس

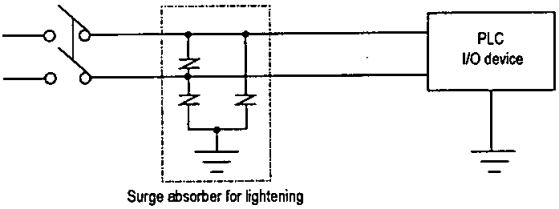
۴-۵ نصب تجهیزات حفاظتی

- در مسیر ورودی AC متصل به منبع تغذیه فیوز مناسب نصب شود.



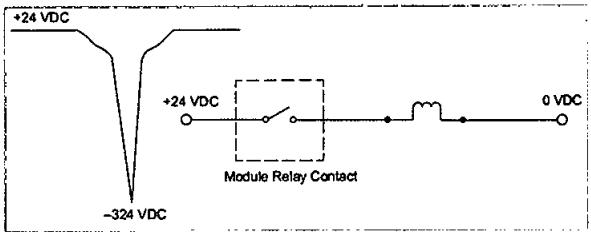
شکل ۴-۱۵ حفاظت در مسیر ورودی تغذیه

- کارت‌های ورودی و خروجی سازندگان معمولاً مجهز به فیوز نیستند. استفاده از فیوز تندکار با جریان کمتر از فیوز اصلی تغذیه I/Oها توصیه می‌شود. راه دیگر برای تشخیص مشخصه جریان فیوز آن است که از مشخصات فنی کارت، حداکثر جریان هر کانال را پیدا کرده و فیوز هر کانال را با جریانی کمی پایین‌تر از آن انتخاب کنیم.
- برای حفاظت کارت‌های I/O از پیک‌های گذرای ناشی از رعد و برق، از ارستر مانند شکل ۴-۱۶ در مسیر تغذیه ورودی و خروجی‌ها استفاده نموده و زمین برقگیر را از زمین PLC جدا کنید.



شکل ۴-۱۶ حفاظت برقگیر در مسیر تغذیه I/Oها

- در استفاده از خروجی‌های رله‌ای وقتی بار سلفی باشد بایستی توجه کرد که در شرایط قطع و وصل رله، ولتاژهای گذرا در سمت بار تولید می‌گردد که دامنه‌ای بیش از ولتاژ کاری نرمال به‌ویژه در نوع DC دارند. در نوع AC به‌دلیل متناوب بودن سیگنال، میزان موج برگشتی بسته به این است که در چه سطحی از ولتاژ مدار قطع شود.



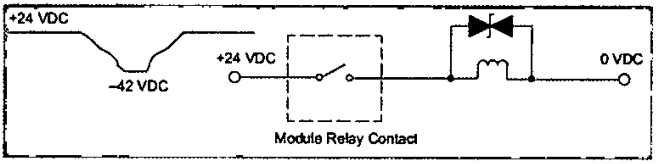
شکل ۴-۱۷ برگشت موج سیگنال برای بار سلفی DC

برای جلوگیری از آسیب این موج‌های گذرا به رله، لازم است در دو سر بار Surge Suppresser نصب شود. اگر اتصال این موج‌گیر در دو سر بار ممکن نباشد، لازم است در دو سر ترمینال رله که به خروجی متصل است نصب گردد.

از موج‌گیرهای مناسب که برای نوع DC و نوع AC قابل استفاده هستند می‌توان به موارد زیر اشاره کرد:

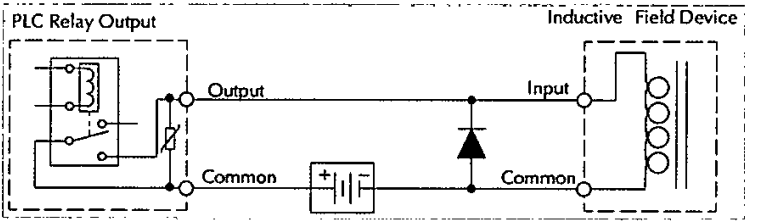
۱. TVS یا Transient Voltage Suppressors
۲. MOV یا Metal Oxide Varistors

به‌عنوان مثال پیک ولتاژی شکل ۴-۱۷ با استفاده با TVS به‌صورت نشان داده شده در شکل ۴-۱۸ در می‌آید.



شکل ۴-۱۸ محدود سازی برگشت موج سیگنال DC با استفاده از TVS

- برای خروجی‌های AC استفاده از TVS یا MOV ضروری است ولی برای خروجی‌های DC می‌توان از دیود هرزگرد مانند شکل ۴-۱۹ نیز استفاده نمود. این مدار برای نوع AC قابل استفاده نیست.



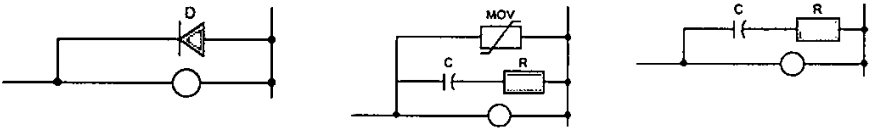
شکل ۴-۱۹ استفاده از دیود هرزگرد برای حفاظت خروجی DC

فصل ۴

دیود فوق لازم است در نزدیکترین نقطه به بار نصب شود و قابلیت تحمل ولتاژ تا ۱۰۰ ولت و جریان تا ۳ آمپر را به‌عنوان حداقل داشته باشد.

قبل از تحریک رله مطمئن شوید که جهت دیود فوق درست نصب شده، در غیر این‌صورت منجر به بروز اتصال کوتاه خواهد شد.

- برای بارهای القایی کوچک که به کارت‌های خروجی AC متصل هستند، حفاظت می‌تواند به‌صورت اسنوبر (مدار RC) باشد. ولی برای بارهای AC بزرگ، اسنوبر کافی نیست و نیاز به MOV یا TVS می‌باشد. شکل ۴-۲۰ این حفاظت‌ها را برای بارهای مختلف نشان می‌دهد.



حفاظت برای بارهای DC | حفاظت برای بارهای AC بزرگ | حفاظت برای بارهای AC کوچک

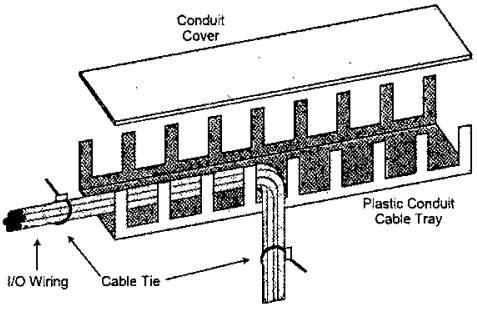
شکل ۴-۲۰ انواع تجهیزات حفاظتی برای خروجی‌های مختلف

۴-۶ سیم‌کشی

- سیم‌کشی با سیم مناسب طبق توصیه سازنده انجام شود.

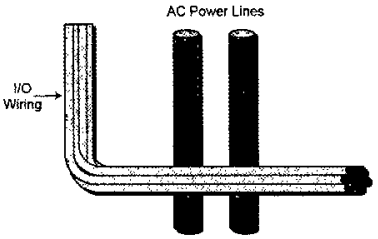
- بهترین روش سیم‌کشی این است که سیگنال‌های هم نوع را گروه‌بندی کرده تا پدیده Crosstalk به حداقل برسد. توصیه می‌شود تا حد امکان گروه‌بندی‌های زیر مراعات شود:

۱. ورودی‌های دیجیتال AC
۲. خروجی‌های دیجیتال AC
۳. ورودی‌های دیجیتال DC
۴. خروجی‌های دیجیتال DC
۵. سیگنال‌های آنالوگ ورودی
۶. سیگنال‌های آنالوگ خروجی



شکل ۴-۲۲ دسته‌بندی مناسب سیم های I/O

- سیگنال‌های ولتاژ پایین DC مانند TTL و آنالوگ از سیگنال‌های دیجیتال AC جدا باشند. بهتر است به‌طور کلی سیگنال‌های AC از کل DCها جدا باشند.
- خطوط تغذیه ورودی در داکت مجزا قرار گیرند و از خطوط I/O جدا باشند.
- بین سیم‌های I/O و کابل‌های قدرت حداقل ۱۰ سانتیمتر فاصله باشد.
- اگر در جایی مسیر خطوط I/O با خطوط تغذیه AC یکدیگر را قطع می‌کنند بهتر است به‌صورت عمودی باشد تا تأثیر نویز به حداقل برسد.

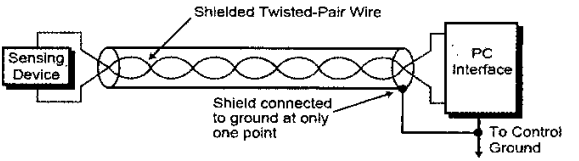


شکل ۴-۲۳ روش مناسب برای عبور دادن سیم‌های I/O از روی سیم های قدرت

- سیم‌ها بایستی شماره‌گذاری شده و در محل اتصال به ترمینال، شماره‌های آنها مشخص باشد. شماره‌گذاری کار عیب‌یابی را ساده‌تر می‌سازد. شماره‌گذاری طبق نقشه انجام می‌شود.
- سیم دارای طول اضافی داخل داکت باشد.

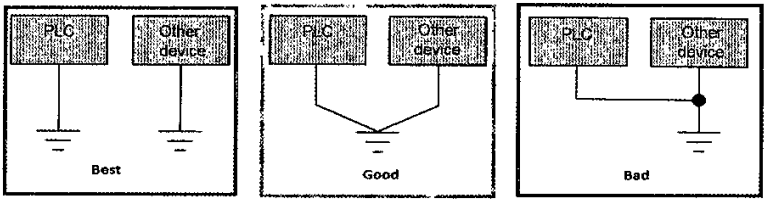
۴-۷ زمین کردن

- شیلد را در سمت PLC زمین کنید. برای این کار در ورودی کابل I/O به پنل روپوش آن را بردارید و با بست مناسب شیلد را به میله ارت داخل پنل متصل کنید.



شکل ۴-۲۴ اتصال شیلد به زمین فقط در سمت PLC

- اگر در کابل‌کشی از سینی یا لوله فلزی استفاده می‌شود آن را زمین کنید.
- برای زمین‌سازی سه روش زیر وجود دارد. استفاده از زمین مجزا برای برق و ابزاردقیق بهترین حالت است.

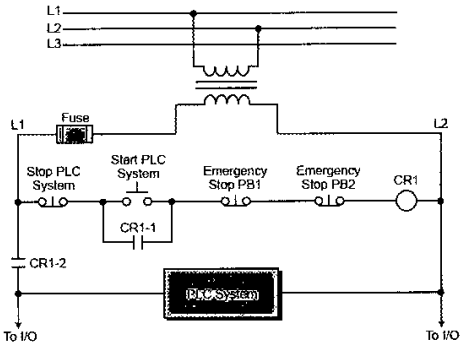


شکل ۴-۲۵ مقایسه سیستم‌های زمین مجزا و مشترک

- برای اتصال PLC به زمین از کابل حداقل $2\ mm^2$ استفاده کنید.
- نوع هادی و اتصالات زمین طبق استاندارد باشد. National Electric Code (NEC) Article 250 یکی از این استانداردهاست.
- سیم زمین در حداقل طول ممکن باشد تا اهم کمتری داشته باشد.
- بدنه پنل زمین شود.
- رک‌های داخل پنل زمین شوند.
- نقاط اتصال زمین باید تمیز باشد و هیچ پوشش رنگ یا لایه غیر رسانایی وجود نداشته باشد.

۸-۴ نصب کلیدهای Emergency

- کلید قطع اضطراری بایستی به‌موازات PLC سیم‌کشی شود. به‌عبارت دیگر حتی در صورتی که کنترلر دچار وضعیت Stop باشد باز بایستی کلید قطع اضطراری بتواند سیستم را قطع کند. Pull Cordهای کنار نوارهای نقاله و لیمیت‌سوئیچ‌های انتهای مسیر سیستم‌های متحرک از جمله موارد اضطراری هستند که تحریک آنها به‌طور مستقیم بایستی بتواند به موتورها و ولوها و سایر وسایل فرمان بدهد. برای رسیدن به درجه ایمنی بالا، هر ماشینی که توسط PLC کنترل می‌شود بایستی دارای کلید قطع اضطراری مستقلی باشد که بتواند ماشین را غیر فعال کند. این سوئیچ اضطراری به PLC سیم‌کشی نمی‌شود.
- در برخی کاربردها، تحریک کلید قطع اضطراری علاوه بر تغذیه ماشین، تغذیه PLC و تغذیه ورودی و خروجی‌ها را نیز قطع می‌کند. شکل ۴-۲۶ این حالت را نشان می‌دهد.



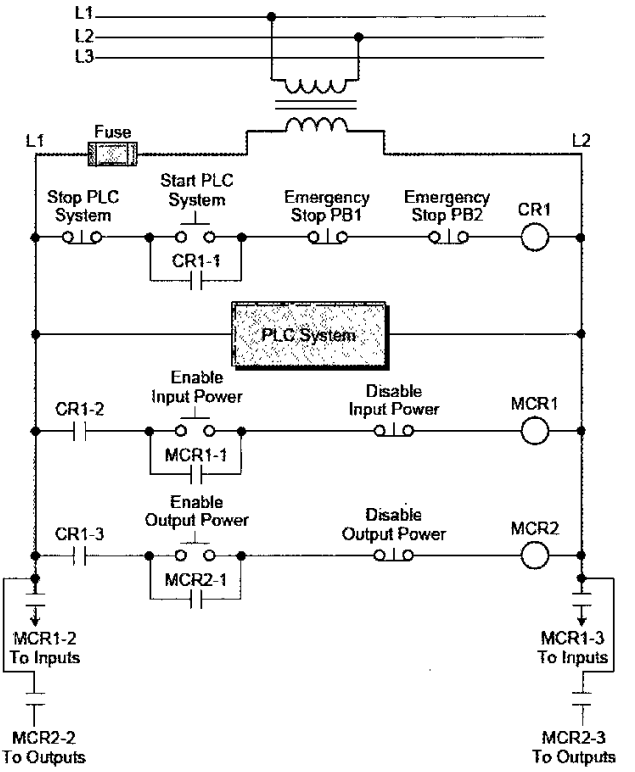
شکل ۴-۲۶ جایگاه کلید قطع اضطراری در مدار PLC

۹-۴ نصب MCR

در برخی کاربردهای دیگر، کلید قطع اضطراری تغذیه I/O را قطع می‌کند ولی تغذیه PLC برقرار است. برای این منظور از یک رله اصلی که به MCR [1] یا SCR [2] موسوم است استفاده می‌شود. کلید اضطراری به این رله سیم‌کشی می‌شود. اگر این رله تحریک شود به‌طور معمول منجر به قطع تغذیه ورودی و خروجی‌ها می‌گردد.

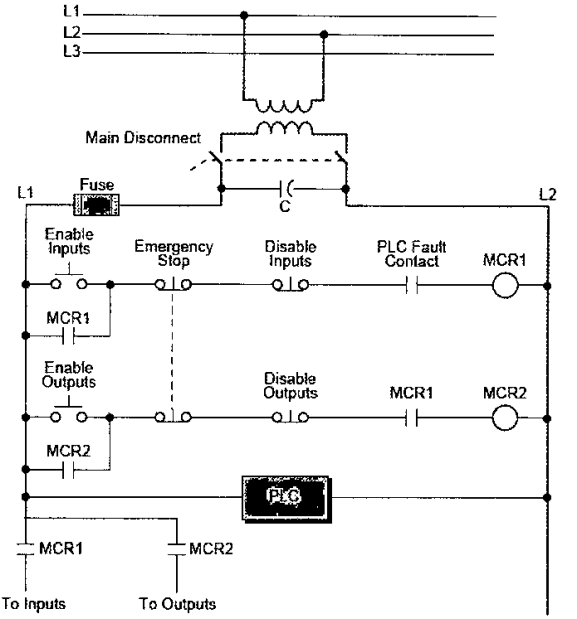
در مدار شکل ۴-۲۷ دو رله MCR در نظر گرفته شده است که یکی برای تغذیه ورودی‌ها و دیگری برای تغذیه خروجی‌هاست. MCR در صورتی‌که تغذیه‌اش قطع شود (de energizing)، منجر به قطع مدار مربوطه می‌شود. در صورت فعال‌شدن کلید قطع اضطراری، این دو رله تحریک می‌شوند و تغذیه ورودی‌ها و خروجی‌ها را قطع می‌کنند، ولی تغذیه PLC قطع نمی‌شود.

1. Master Control Relay
2. Safety Control Relay



شکل ۴-۲۷ استفاده از MCR برای قطع تغذیه I/Oها بدون قطع تغذیه PLC

- به‌طور انتخابی در صورت نیاز می‌توان سیم‌کشی را به‌صورتی انجام داد که با عمل‌کردن کلید اضطراری، تغذیه خروجی‌ها قطع شود ولی تغذیه ورودی‌ها و تغذیه PLC برقرار باشد.
- طراحی می‌تواند به‌گونه‌ای باشد که برخی فالت‌های CPU نیز رله MCR را تحریک کند. برای این منظور همانطورکه در شکل ۴-۲۸ دیده می‌شود یک کنتاکت PLC Fault در نظر گرفته شده است. وقتی PLC نرمال است این کنتاکت بسته است و در شرایط فالت باز شده و در نتیجه MCR تغذیه I/O را قطع می‌کند.



شکل ۴-۲۸ عملکرد MCR در صورت بروز فالت روی CPU

۴-۱۰ پرسش و تحقیق

- انواع سیستم های زمین را بررسی و با یکدیگر مقایسه کنید.

۴-۱۱ تست‌های خودآزمایی

۱- بهترین موقعیت نصب PLC کجاست؟

الف) نزدیک خطوط تغذیه AC　　ب) در اتاق کنترل

ج) نزدیک ماشین　　د) دور از ماشین و فرآیند

۲- بطور معمول کنترلرهای نصب شده داخل پنل بایستی در مقابل کدام شرایط دمایی مقاوم باشند؟

الف) ۶۰ درجه سانتی‌گراد بیرون پنل　　ب) ۵۰ درجه سانتی‌گراد بیرون پنل

ج) ۶۰ درجه سانتی‌گراد داخل پنل　　د) ۵۰ درجه سانتی‌گراد داخل پنل

۳- بهترین مکان جهت نصب منبع تغذیه در پنل کجاست؟

الف) پایین PLC و تجهیزات　　ب) بالای PLC و تجهیزات

ج) کنار PLC و تجهیزات　　د) بیرون از پنل

۴- در نصب کارت‌های ورودی و خروجی، به‌منظور کاهش اثر نویز کدام مورد توصیه می‌شود؟
الف) سیم‌های مربوط به کارت‌های AC و DC را دسته‌بندی و از یکدیگر جدا نمود.
ب) سیم‌های کارت‌های دیجیتال را در یک گروه قرار داد.
ج) سیم‌های کارت های آنالوگ را در یک گروه قرار داد.
د) هر سه مورد
۵- در مورد تغذیه کارت‌های I/O کدام مورد صحیح است؟
الف) تغذیه‌ی بخش Logic و بخش Power کارت مشترک باشد.
ب) تغذیه‌ی بخش Logic و بخش Power جدا از هم باشند.
ج) تغذیه‌ی بخش Logic و بخش Power از هم ایزوله باشند.
د) موارد ۲ و ۳ صحیح است.
۶- بهترین روش زمین نمودن کابل‌های متصل به PLC کدام روش است؟
الف) استفاده از زمین مشترک با سایر تجهیزات
ب) استفاده از زمین مشترک و کابل مشترک با سایر تجهزات
ج) استفاده از زمین جداگانه برای PLC و سایر تجهیزات
د) استفاده از زمین مشترک و سیم جدا از سایر تجهیزات
۷- تفاوت کلید Emergency و رله‌ی MCR در چیست؟
الف) کلید Emergency تغذیه‌ی PLC را قطع می‌کند ولی MCR تغذیه‌ی PLC و I/O ها را قطع می‌کند.
ب) کلید Emergency تغذیه‌ی I/Oها را قطع می‌کند ولی MCR تغذیه‌ی PLC و I/O ها را قطع می‌کند.
ج) کلید Emergency تغذیه‌ی PLC و I/Oها را قطع می‌کند ولی MCR تغذیه‌ی PLC را قطع می‌کند.
د) کلید Emergency تغذیه‌ی PLC و I/Oها را قطع می‌کند ولی MCR تغذیه‌ی I/Oها را قطع می‌کند.

فصل ۴

فصل ۵

آشنایی با وسایل ورودی و خروجی PLC
(سنسورها و عملگرها)

۵-۱ مقدمه
۵-۲ سیگنال‌های ورودی دیجیتال
۵-۳ کلیدها و شستی‌های اپراتوری
۵-۴ سنسورهای دیجیتال
۵-۴-۱ لیمیت‌سوئیچ
۵-۴-۲ سنسور القایی
۵-۴-۳ سنسورهای خازنی
۵-۴-۴ سنسورهای آلتراسونیک
۵-۴-۵ سنسور فتوالکتریک
۵-۴-۶ مقایسه ویژگی سنسورهای دیجیتال
۵-۴-۷ سوئیچ‌های فرآیندی
۵-۴-۷-۱ Pressure Switch
۵-۴-۷-۲ Level Switch
۵-۴-۷-۳ Temperature Switch
۵-۴-۷-۴ Flow Switch
۵-۵ سیگنال‌های خروجی دیجیتال
۵-۵-۱ محرک‌های منطقی
۵-۵-۲ سلونوئیدها
۵-۵-۳ ولوها
۵-۵-۴ کنتاکتور
۵-۵-۵ سایر تجهیزات
۵-۶ سیگنال‌های آنالوگ ورودی
۵-۶-۱ سنسورهای اندازه‌گیری دما
۵-۶-۲ سنسورهای اندازه‌گیری فشار
۵-۶-۳ سنسورهای اندازه‌گیری ارتفاع سطح
۵-۶-۴ سنسورهای اندازه‌گیری فلو
۵-۷ سیگنال آنالوگ خروجی
۵-۷-۱ ولوهای کنترلی
۵-۷-۲ درایو کنترل‌دور موتورهای الکتریکی
۵-۸ سیگنال‌های خاص
۵-۹ پرسش و تحقیق
۵-۱۰ تست‌های خودآزمایی

در این فصل با مهمترین وسایلی که به‌عنوان ورودی یا خروجی به PLC متصل می‌شوند آشنا می‌شوید. این وسایل سنسورهای و عملگرهای دیجیتال و آنالوگ هستند. سعی شده مطالب به‌صورت اجمال و در حد آشنایی ذکر شود از اینرو توصیه می‌شود که خواننده برای شناخت دقیق‌تر رفتار PLC با وسایل فیلد، این فصل را به‌دقت مطالعه کند.

چکیده مطالب

- سنسورها همچون چشم و گوش و حواس یک فرآیند هستند. تصمیم‌گیری و فرمان PLC بر اساس وضعیت آنها انجام می‌شود.
- کلیدها و شستی‌های اپراتوری یکی از مهمترین ورودی‌های دیجیتال محسوب می‌شوند.
- سنسورهای دیجیتال به دو گروه تماسی (مانند لیمیت‌سوئیچ) و غیرتماسی (مانند پروکسیمیتی سوئیچ) تقسیم می‌شوند.
- سوئیچ‌های دما، فشار، فلو و سطح از جمله سنسورهای دیجیتال هستند.
- سلونوئید ولوها، کنتاکتورها، رله‌ها و وسایل هشدار دهنده از جمله خروجی‌های دیجیتال هستند.
- ترانسدیوسر مبدل کمیت فیزیکی به سیگنال الکتریکی و ترانسمیتر مبدل این سیگنال به نوع استاندارد برای انتقال به کنترلر است.
- معمولاً ترانسدیوسر و ترانسمیتر در یک وسیله عرضه می‌شوند که به مجموع آنها ترانسمیتر اطلاق می‌شود.
- ترموکوپل و RTD و ترمیستور از سنسورهای اندازه‌گیری دما هستند.
- فشار به روش‌های مختلفی اندازه‌گیری می‌شود که از جمله آنها می‌توان به انواع استرین‌گیج، خازنی، مغناطیسی، رلوکتانسی و LVDT اشاره کرد.
- ارتفاع سطح می‌تواند توسط اختلاف فشار یا تغییر ظرفیت خازن یا توسط امواج صوتی یا امواج رادارای اندازه‌گیری شود.
- فلو می‌تواند توسط نوع توربینی، نوع اختلاف فشار، نوع القایی، نوع Vortex، نوع mass flow و آلتراسونیک اندازه‌گیری شود.
- ولوهای کنترلی و درایوهای موتورهای الکتریکی از مهمترین وسایل آنالوگ خروجی هستند.

فصل ۵

اصطلاحات و تعاریف

Orifice

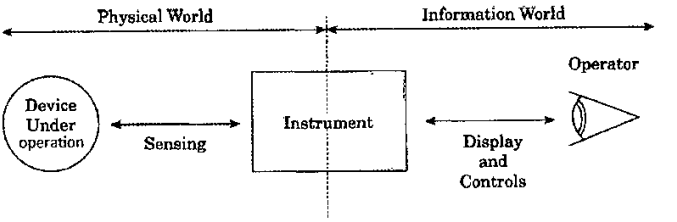
صفحه‌ای دایره‌ای شکل که در وسط لوله سیال قرار می‌گیرد و وسط آن سوراخی وجود دارد که مسیر سیال را باریک می‌کند. باریک شدن مسیر اختلاف فشاری را در دو طرف ایجاد می‌کند . از این پدیده در اندازه‌گیری استفاده می‌شود. به شکل ۵-۷۳ رجوع کنید.

Venturi

لوله‌ای که قطر آن در وسط کمتر از قطر آن در دو سمت ورودی و خروجی است و شبیه Orifice منجر به اختلاف فشار ورودی و خروجی شده و در اندازه‌گیری به‌کار می رود. به شکل۵-۷۳ رجوع کنید.

۵-۱ مقدمه

شناخت وسایل ابزاردقیق برای کسی که می‌خواهد با PLC کار کند، برنامه بنویسد، یا عیب‌یابی انجام دهد، یک ضرورت اجتناب ناپذیر است. سنسورها همچون چشم و گوش و حواس یک فرآیند هستند که کنترلر تصمیم‌گیری خود را بر اساس سیگنال‌های دریافتی از آنها انجام می‌دهد . از اینرو بایستی به‌خوبی عملکرد آنها را درک کرد، سپس به برنامه‌ریزی کنترل فرآیند اقدام نمود.



شکل ۵-۱ نقش وسایل ابزاردقیق در یک فرآیند

بررسی جزئیات وسایل ابزاردقیق که به‌عنوان ورودی یا خروجی PLC تلقی می‌شوند خارج از چارچوب این کتاب که با محوریت PLC تدوین شده می‌باشد و لازم است خواننده به کتب مرجع ابزاردقیق مراجعه نماید. در این فصل فقط به اجمال نگاهی داریم به انواع متداول وسایلی که در یک فرآیند به‌عنوان ورودی و خرجی PLC محسوب می‌شوند.

همانطور که در فصل‌های قبل اشاره شد، به منظور کنترل فرایند، اطلاعات مربوط به فرایند از طریق سنسورها به‌طور مستقیم یا از طریق شبکه وارد PLC می‌شود. کارت ورودی سیگنال الکتریکی دریافتی را به دیتا تبدیل نموده و سپس دیتا را در اختیار CPU قرار می‌دهد، CPU با آنالیز دیتا طبق برنامه نوشته شده فرامینی را برای خروجی ها صادر می‌نماید. فرمان صادر شده توسط CPU به‌صورت دیتاست که در کارت خروجی به سیگنال الکتریکی تبدیل شده و به تجهیزات تحت کنترل ارسال می‌گردد.

در این بخش انواع سیگنال‌ها را به چهار گروه اصلی زیر تقسیم‌بندی نموده و مهمترین وسایل مربوط به هر دسته را به اجمال توضیح می‌دهیم:

- سیگنال‌های ورودی دیجیتال
- سیگنال‌های خروجی دیجیتال
- سیگنال‌های ورودی آنالوگ
- سیگنال‌های خروجی آنالوگ

لازم به ذکر است که برخی از سیگنال‌ها در گروه‌بندی فوق جای نمی‌گیرند. به‌عنوان مثال پالس‌های سریع که فرکانس زیاد دارند توسط کارت‌های دیجیتال معمولی قابل دریافت نیستند، این سیگنال‌ها آنالوگ نیز محسوب نمی‌شوند و برای دریافت آنها نیاز به کارت‌های خاصی است. به‌عنوان مثال انکودر وسیله‌ای است که برای اندازه‌گیری موقعیت یا سرعت شافت موتور استفاده می‌شود. خروجی آن پالس‌هایی است که در فرکانس بالا قابل دریافت با کارت ورودی دیجیتال نیست.

فصل ۵

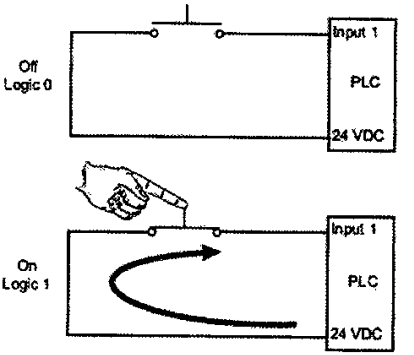
۵-۲ سیگنال‌های ورودی دیجیتال

مفهوم سیگنال‌های دیجیتال یعنی دو کمیتی بودن، یعنی اینکه یا سیگنال وجود دارد یا ندارد. بر همین اساس شستی‌ها و سنسورهای دیجیتال تنها دارای دو حالت می باشند:

الف) حالت قطع: در این حالت، معمولاً سیگنالی به کارت ارسال نمی‌گردد.

ب) حالت وصل: در این حالت، مقدار نامی سیگنال خروجی سنسور به کارت ارسال می‌گردد.

به‌عنوان مثال در شکل ۵-۲ از یک عدد شستی نرمال باز که از یک طرف به منبع ولتاژ 24 V DC و از طرف دیگر به کارت ورودی در PLC متصل شده است، استفاده شده است. تا زمانی که شستی فشرده نشده است، هیچ ولتاژی برای کارت ارسال نمی‌گردد؛ این حالت معادل صفر منطقی در نظر گرفته می‌شود. اگر شستی فشرده گردد، 24 V DC برای کارت ارسال می‌گردد که معادل یک منطقی است.



شکل ۵-۲ صفر و یک منطقی در یک مدار الکتریکی

به‌طور کلی سیگنال‌های دیجیتال را می‌توان از ادوات زیر دریافت نمود:

- شستی‌ها و کلیدهای اپراتوری
- سنسورها
- تیغه کمکی رله‌ها و کنتاکتورها

سیگنالی که از این وسایل به PLC ارسال می‌شود ممکن است در حالت عادی صفر یا یک باشد که بستگی به نوع کنتاکت سنسور دارد:

- اگر کنتاکت درحالت عادی قطع است به آن Normaly Open یا اختصاراً NO گفته می‌شود. این کنتاکت به PLC در حالت عادی صفر و اگر عمل کرد یک می‌فرستد.
- اگر کنتاکت درحالت عادی وصل است به آن Normaly Closed یا اختصاراً NC گفته می‌شود. این کنتاکت به PLC در حالت عادی یک و اگر عمل کرد صفر می‌فرستد.
- در برخی وسایل هر دو کنتاکت تعبیه شده است که انتخاب NO یا NC بسته به نوع کاربرد می‌تواند متفاوت باشد.

۵-۳ کلیدها و شستی‌های اپراتوری

به‌منظور صادر نمودن فرامین توسط اپراتور، از شستی یا کلید استفاده می‌گردد. مثلاً در یک سیستم اتوماسیون صنعتی اپراتور می‌تواند از طریق فشردن یک شستی، سیستم را روشن یا خاموش نماید.

کلیدهای اپراتوری

این ادوات دارای حالات مختلف قطع و وصل بوده و به‌گونه‌ای طراحی شده‌اند که در هر کدام از حالات قطع و وصل که قرار گیرند، تا زمانی که کاربر (اپراتور) حالت آنرا تغییر ندهد، در همان وضعیت باقی می‌مانند. در شکل ۵-۳ نمایی از یک عدد کلید را مشاهده می‌نمایید.

شکل ۵-۳ کلید سلکتوری

برخی کلیدها دو وضعیتی هستند و به یک کانال کارت دیجیتال سیم‌کشی می‌شوند. برخی کلیدها سه‌وضعیتی هستند، در این حالت دو کانال دیجیتال ورودی را اشغال می‌کنند. به‌عنوان مثال یک کلید سه‌وضعیتی می‌تواند برای Forward (حرکت به‌جلو) یا Reverese (حرکت معکوس) یا Off بودن موتور به‌کار رود. به این نوع کلیدها Selector Switch نیز گفته می‌شود.

پوش‌باتون‌ها

عملکرد این ادوات مانند کلیدها بوده ولی دارای یک حالت کاری نرمال هستند که پس از اینکه اپراتور آنها را رها نمود به حالت نرمال خود برمی‌گردند. در شکل ۵-۴ نمایی از یک عدد پوش‌باتون را مشاهده می‌نمایید.

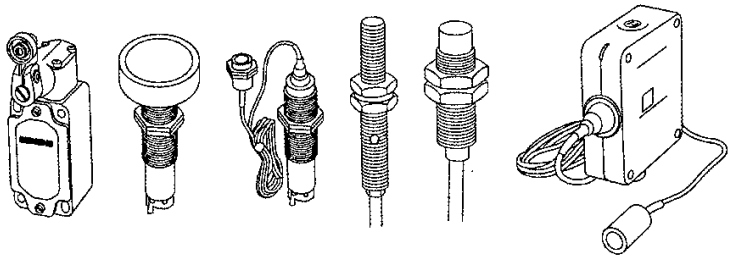
برخی از پوش‌باتون‌ها عمل وصل و برخی دیگر عمل قطع را انجام می‌دهند. به‌عنوان مثال معمولاً برای روشن شدن یک سیستم از نوع نرمال باز که عمل وصل را انجام می‌دهد و برای خاموش کردن سیستم از نوع نرمال بسته که عمل قطع را انجام می‌دهد، استفاده می‌شود.

شکل ۵-۴ پوش باتون

فصل ۵

۵-۴ سنسورهای دیجیتال

سنسورهایی که دارای سیگنال خروجی دیجیتال هستند در صنعت بسیار پرکاربرد می‌باشند. به‌عنوان مثال وقتی فشار یا دما یا ارتفاع سطح به حد معینی برسد یا اگر فلوی سیال قطع شود یا اگر جسمی در مقابل سنسور قرار گیرد، این سنسورها تغییر وضعیت داده و یک سیگنال دیجیتال را قطع یا وصل می‌کنند. شکل ۵-۵ نمونه‌هایی از سنسورهای دیجیتال را نشان می‌دهد.



شکل ۵-۵ نمونه‌ای از سنسورهای دیجیتال

عملکرد این سنسورها با یکدیگر متفاوت است. برخی به‌صورت مکانیکی قطع و وصل می‌شوند (مانند لیمیت‌سوئیچ)، برخی دیگر با تغییر میدان مغناطیسی یا تغییر خازن محیط اطرافشان تحریک می‌شوند (مانند پروکسیمیتی سوئیچ) و برخی دیگر با قطع و وصل نور کار می‌کنند (مانند نوع فتوالکتریک).

به‌صورت کلی سنسورهای دیجیتال را می‌توان به دو نوع تماسی و غیر تماسی تفکیک کرد.

تقسیم‌بندی سنسورها از نظر تماسی یا غیرتماسی

به لحاظ طرز عملکرد، این سنسورها را می‌توان به انواع زیر تقسیم نمود:

- سنسورهای تماسی: شامل انواع لیمیت‌سوئیچ مکانیکی
- سنسورهای بدون تماس: شامل
  - القایی
  - خازنی
  - ۱) پراکسیمیتی
  - ۲) آلتراسونیک
  - ۳) فتوالکتریک

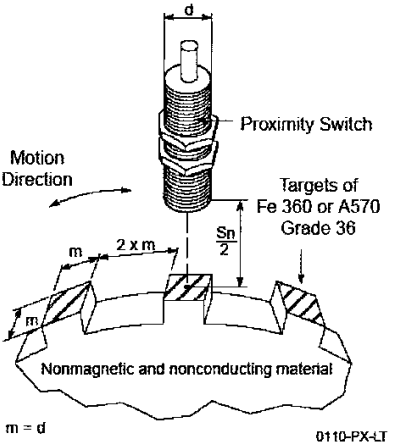
الف) سنسورهای تماسی

سنسورهایی هستند که مستقیماً با شیء مورد نظر در پروسه تماس داشته و از طریق تماس مکانیکی، متوجه‌ی وجود یا عدم وجود شیء می‌گردند، مانند لیمیت‌سوئیچ.

تو رفتگی است که فقط قسمت برآمده توسط سنسور قابل حس است. ماکزیمم سرعتی که محور می‌تواند بچرخد و سنسور لبه‌ها را تشخیص دهد (که به آن ماکزیمم فرکانس سوئیچینگ گفته می‌شود) چقدر است؟ این فرکانس به پارامترهای زیر بستگی دارد:

- سطح جسم مقصد چه اندازه است؟
- فاصله سنسور تا جسم چقدر است؟
- سرعت چرخش چقدر است؟
- سطح مقطع سنسور چقدر است؟

محاسبه فرکانس سوئیچینگ براساس پارامترهای فوق در استاندارد IEC60947-5-2 آمده است.



شکل ۵-۱۹ سنجش سرعت چرخش محور با سنسور پروکسیمیتی القایی

نکته‌ای که باید به آن توجه کرد این است که اگر فرکانس پالس دریافتی از سنسور بالا باشد بایستی کارت ورودی دیجیتال نیز قابلیت سوئیچ با این فرکانس را داشته باشد و سیکل CPU نیز کمتر از زمان تغییر پالس دریافتی باشد. معمولاً در فرکانس‌های بالا نیاز به سخت‌افزار خاصی در کنار CPU برای دریافت این پالس‌ها وجود دارد.

فصل ۵

مزایا و معایب سنسورهای پراکسیمیتی القایی

مزایای این سنسورها عبارتند از:

- عدم تأثیر رطوبت در آن
- عدم تأثیر محیط‌های آلوده و گرد و خاک در عملکرد آن
- عدم سایش مکانیکی (به‌دلیل عدم وجود قسمت مکانیکی متحرک)
- تشخیص آن وابسته به رنگ جسم نیست

معایب این سنسورها عبارتند از:

- فقط اجسام فلزی را حس می‌کند.
- رِنج عملکرد آن نسبت به سایر سنسورها کمتر است.
- تحت تأثیر میدان‌های الکترومغناطیسی قوی قرار می‌گیرد.

ولتاژ عملکرد:

- 10-30VDC
- 15-34VDC
- 10-65VDC
- 20-320VDC
- 20-265VAC

هیسترزیس سنسور

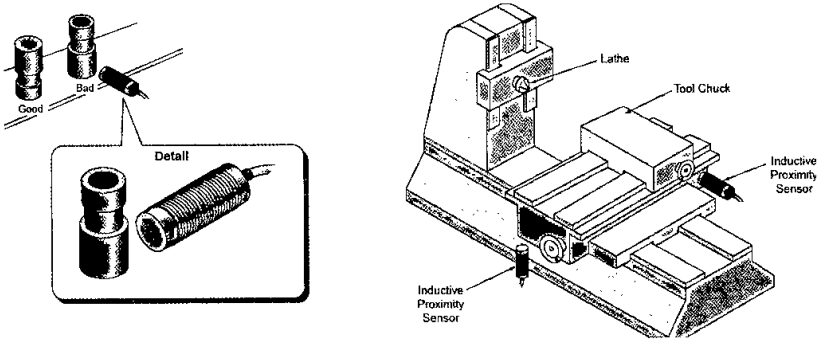
سنسور دارای یک ناحیه هیسترزیس است و تغییرات را در آن ناحیه نمی‌بیند. این ویژگی باعث می‌شود که اگر دستگاه دارای لرزش باشد منجر به قطع و وصل سیگنال نگردد.

کاربردهای عملی سنسورهای القایی

از جمله کاربردهای عملی سنسورهای القایی می‌توان به موارد زیر اشاره نمود:

- ماشین ابزار
- تشخیص قطعات فلزی
- تشخیص مرتب بودن قطعات

در شکل ۵-۲۰ نمونه‌هایی از کاربرد سنسور القایی را مشاهده می‌کنید.



تشخیص موقعیت ماشین ابزار — جداسازی محصولات

شکل ۵-۲۰ نمونه کاربردهای سنسور پروکسیمیتی القایی

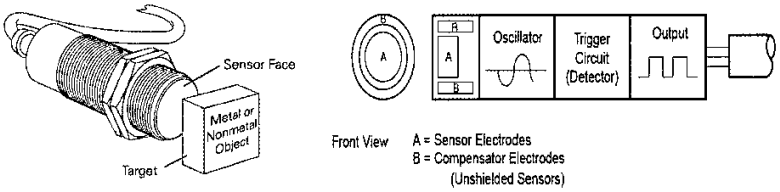
۵-۴-۳ سنسورهای خازنی

سنسور خازنی یک تکنولوژی بدون تماس مناسب برای تشخیص فلزات، غیر فلزات، جامدات و مایعات می‌باشد. *سنسور خازنی بهترین گزینه برای تشخیص اجسام غیر فلزی است*؛ زیرا این سنسورها دارای مشخصات پایین‌تر و هزینه‌ی بالاتر نسبت به سنسورهای القایی می‌باشند، از اینرو در کاربردهای تشخیص اجسام فلزی، استفاده از سنسورهای القایی بر سنسورهای خازنی ترجیح دارد. در کل می‌توان بیان نمود که سنسورهای القایی قابلیت اطمینان بیشتری نسبت به سنسورهای خازنی داشته و تکنولوژی ساخت آنها نیز ارزان‌تر است.

ساختمان سنسورهای خازنی

سنسورهای خازنی معمولاً از قسمت‌های زیر تشکیل شده‌اند:

۱) صفحه‌ی فلزی خازنی
۲) اسیلاتور
۳) مدار تریگر کننده
۴) مکانیزم قطع و وصل خروجی سنسور

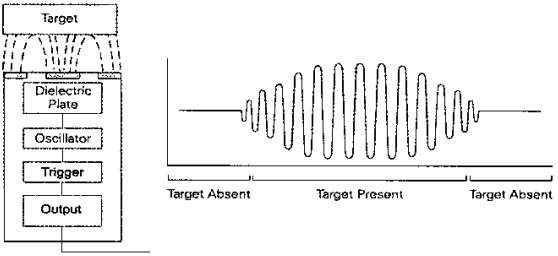


شکل ۵-۲۱ ساختمان سنسور پروکسیمیتی خازنی

عملکرد سنسور خازنی

پراب پشت سنسور، یک صفحه‌ی فلزی خازنی می‌باشد. هنگامی که تغذیه‌ی سنسور برقرار شود، یک میدان الکترواستاتیکی تولید می‌گردد. هرگاه جسمی در این میدان الکترواستاتیکی وارد شود، ظرفیت خازنِ سنسور دچار تغییر شده و سنسور نسبت به این تغییر عکس العمل نشان می‌دهد. هنگامی که جسم خارج از میدان الکترواستاتیکی قرار دارد، اسیلاتور غیرفعال است. چنانچه جسم نزدیک میدان شود، یک کوپلینگ خازنی بین جسم هدف و صفحه فلزی (پراب) به‌وجود می‌آید. هنگامی‌که ظرفیت خازنی به آستانه‌ی تعیین شده برسد، اسیلاتور فعال شده و خروجی مدار سنسور را فعال می‌کند. در شکل ۵-۲۲ نحوه عملکرد سنسور به منظور تشخیص جسم هدف، مشاهده می‌گردد.

فصل ۵



شکل ۵-۲۲ عملکرد سنسور پروکسیمیتی خازنی

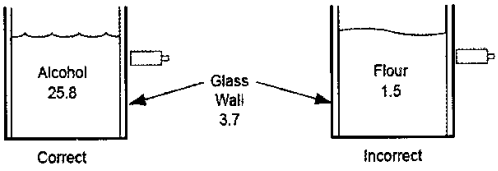
رنج فاصله‌ی عملکردی سنسور

رنج فاصله‌ی عملکردی سنسور برای تشخیص اجسام، به اندازه‌ی جسم هدف، ثابت دی‌الکتریک و فاصله‌ی جسم هدف از سنسور بستگی دارد. ثابت دی‌الکتریکی جزو مشخصات مواد بوده و در همه‌ی مواد وجود دارد. جدول ۵-۲ ثابت دی‌الکتریک را برای برخی از اجسام نشان می‌دهد.

جدول ۵-۲ ثابت دی‌الکتریک برخی از اجسام

Material	Constant	Material	Constant
Acetone	19.5	Perspex	3.2-3.5
Acrylic Resin	2.5-4.5	Petroleum	2.0-2.2
Air	1.000264	Phenol Resin	4-12
Alcohol	25.8	Polyacetal	3.6-3.7
Ammonia	15-25	Polyamide	5.0
Aniline	6.9	Polyester Resin	2.8-8.1
Aqueous Solutions	50-80	Polyethylene	2.3
Bakelite	3.6	Polypropylene	2.0-2.3
Benzene	2.3	Polystyrene	3.0
Carbon Dioxide	1.000985	Polyvinyl Chloride Resin	2.8-3.1
Carbon Tetrachloride	2.2	Porcelain	4.4-7
Celluloid	3.0	Powdered Milk	3.5-4
Cement Powder	4.0	Press Board	2-5
Cereal	3-5	Quartz Glass	3.7
Chlorine Liquid	2.0	Rubber	2.5-35
Ebonite	2.7-2.9	Salt	6.0
Epoxy Resin	2.5-6	Sand	3-5
Ethanol	24	Shellac	2.5-4.7
Ethylene Glycol	38.7	Shell Lime	1.2
Fired Ash	1.5-1.7	Silicon Varnish	2.8-3.3
Flour	1.5-1.7	Soybean Oil	2.9-3.5
Freon R22 & 502 (liquid)	6.11	Steel	
Gasoline	2.2	Styrene Resin	2.3-3.4
Glass	3.7-10	Sugar	3.0
Glycerine	47	Sulphur	3.4
Marble	8.0-8.5	Teflon	2.0
Melamine Resin	4.7-10.2	Toluene	2.3
Metal		Transformer Oil	2.2
Mica	5.7-6.7	Turpentine Oil	2.2
Nitrobenzine	36	Urea Resin	5-8
Nylon	4-5	Vaseline	2.2-2.9
Oil Saturated Paper	4.0	Water	80
Paraffin	1.9-2.5	Wood, Dry	2-7
Paper	1.6-2.6	Wood, Wet	10-30

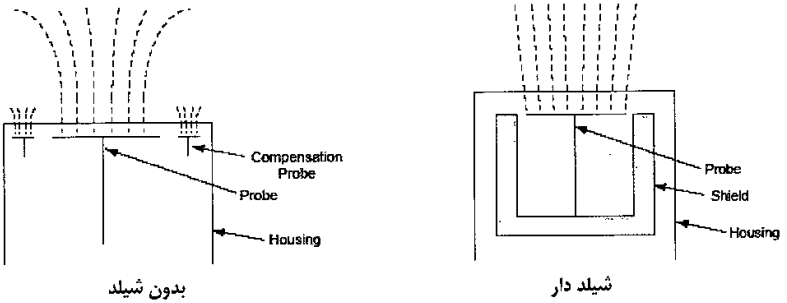
باید توجه نمود که اجسام با ثابت دی‌الکتریک بزرگتر، زودتر توسط سنسور شناسایی می‌گردند. به‌عنوان مثال فرض کنید در دو ظرف شیشه‌ای یکی الکل و در دیگری آرد وجود دارد. اگر سنسور خازنی را مانند شکل ۵-۲۳ در بیرون مخزن نصب کنیم ارتفاع الکل را تشخیص می‌دهد ولی ارتفاع آرد را خیر؛ زیرا ضریب دی‌الکتریک الکل از شیشه بزرگتر است ولی ضریب دی‌الکتریک آرد از شیشه کمتر است. توصیه می‌شود هر کاربرد ترکیبی ابتدا تست و سپس سنسور مناسب برای آن انتخاب و نصب گردد.



شکل ۵-۲۳ کاربرد پروکسیمیتی خازنی برای حس کردن سطح مخزن

سنسور خازنی شیلددار و بدون شیلد

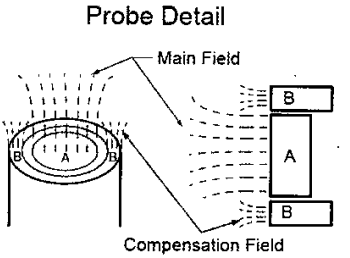
مشابه آنچه برای سنسور القایی ذکر شد در نوع خازنی نیز ممکن است بدنه سنسور دارای شیلد فلزی باشد که در این حالت تمرکز سنسور فقط در روبروی آن است. در نوع بدون شیلد، سنسور حساس‌تر است و در لبه‌ها نیز جسم را تشخیص می‌دهد.



شکل ۵-۲۴ سنسور پروکسیمیتی خازنی شیلددار و بدون شیلد

فصل ۵

نوع بدون شیلد در لبه‌ها مجهز به پراب جبران‌ساز است. این پراب باعث می‌شود تا خطای ناشی از ذرات گرد و غبار و وجود روغن و آب حذف شود. خازنِ قسمت‌های جبران‌ساز برای فاصله‌های بسیار نزدیک تغییر می‌کند. زمانی که فقط خازن اصلی تغییر کند به‌معنی وجود جسم است ولی اگر خازن اصلی و خازن ناحیه جبران‌ساز با هم تغییر کنند به‌معنی وجود گرد و غبار و ... است و در این حالت خروجی سنسور تغییر نخواهد کرد.



شکل ۵-۲۵ ساختار سنسور پروکسیمیتی خازنی بدون شیلد

مزایا و معایب سنسورهای خازنی

مزایای این سنسورها عبارتند از:

- تشخیص فلزات، غیر فلزات، جامدات و مایعات
- تشخیص جسم هدف از میان بعضی اجسام مانند جعبه
- عمر مفید بالا
- طریقه نصب به صورت های مختلف

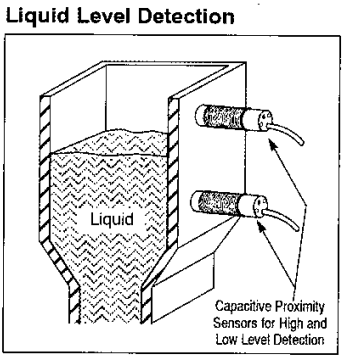
معایب این سنسورها عبارتند از:

- فاصله‌ی کوتاه حس‌کنندگی(که بسته به نوع ماده دارد ولی در بسیاری از حالات کمتر از ۲/۵ سانتی‌متر است)
- حساسیت خیلی زیاد نسبت به شرایط محیطی از قبیل رطوبت که می‌تواند روی خروجی سنسور تأثیر بگذارد.

کاربردهای عملی سنسورهای خازنی

- حس نمودن سطح مواد (level)
- خطوط Filling (پر نمودن محصولات) با امکان تشخیص جعبه خالی از جعبه پر
- تشخیص وجود قطعات پلاستیکی در داخل مواد مختلف
- تشخیص پالت حمل مواد
- تشخیص محصولات خراب که نسبت به محصولات سالم شکل نامنظم داشته باشند.

در شکل ۵-۲۶ کاربرد سنسور خازنی در تشخیص سطح (Level) را مشاهده می‌نمایید.



شکل ۵-۲۶ نمونه کاربردهای سنسور پروکسیمیتی خازنی

۵-۴-۴ سنسورهای آلتراسونیک

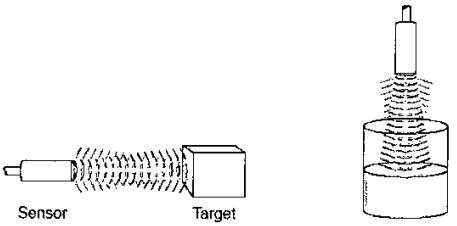
از جمله سنسورهای دیجیتال که به‌منظور تشخیص یک جسم به‌کار می‌رود می‌توان به سنسورهای آلتراسونیک اشاره نمود. این سنسورها براساس ارسال و دریافت صوت عمل‌می‌نمایند؛ از اینرو به آنها سنسورهای آلتراسونیک به معنی مافوق صوت گفته می‌شود.

اصول عملکرد

این سنسورها یک پالس صوتی تولید می‌نمایند که پس از برخورد به اجسام درون میدان موج، این پالس بازتاب پیدا می‌نماید. صوت بازتاب شده (اکو) توسط سنسور دریافت می‌گردد. تشخیص صوت، یک سیگنال در خروجی سنسور ایجاد می‌کند. سیگنال خروجی این سنسورها می‌تواند به‌صورت آنالوگ یا دیجیتال باشد.

این سنسورها توانایی تشخیص اغلب اجسام (فلزات و غیر فلزات، اجسام شفاف، اجسام کدر، مایع و جامد) را که دارای بازتاب‌پذیری مناسب (کافی) هستند دارا می‌باشد ولی برای اجسامی که صدا را جذب می‌کنند مانند فوم ، لباس و ... مناسب نمی باشند.

فصل ۵



شکل ۵-۲۷ سنسور التراسونیک

ساختمان سنسور آلتراسونیک

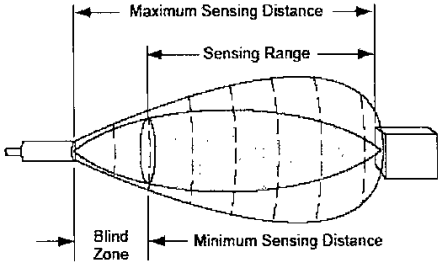
سنسورهای آلتراسونیک معمولاً از قسمت‌های زیر تشکیل می‌شوند:

- ترانسدیوسر/ گیرنده
- مقایسه‌گر
- مدار آشکارساز
- خروجی نیمه‌هادی

در شکل ۵-۲۸ قسمت‌های مختلف یک سنسور آلتراسونیک را مشاهده می‌نمایید.

شکل ۵-۲۸ ساختار سنسور التراسونیک

مقایسه‌گر، زمان رفت و برگشت صوت را محاسبه می‌کند. با معلوم بودن سرعت صوت، فاصله جسم تا سنسور نیز قابل تشخیص است که به‌صورت سیگنال آنالوگ ارسال می‌شود. در حالت دیجیتال با برگشت صوت، یک سیگنال دیجیتال به PLC ارسال می‌گردد. فاصله بین سنسور و منبع صوت دارای یک حداکثر و یک حداقل است. همانطور که در فاصله بیش از حداکثر جسم قابل آشکارسازی نیست، اگر فاصله از حداقل مجاز هم کمتر باشد جسم تشخیص داده نمی‌شود. به این ناحیه نقطه کور می‌گویند.



شکل ۵-۲۹ حدود آشکار سازی در سنسور التراسونیک

مزایا و معایب سنسورهای آلتراسونیک

مزایای این سنسورها عبارتند از:

- سنسورهای آلتراسونیک جهت حس نمودن فواصل زیاد در حدود 15m مناسب می باشد.
- پاسخ این سنسورها، به رنگ سطح یا بازتاب‌پذیری نوری جسم وابسته نیست.
- سنسورهای آلتراسونیک با خروجی دیجیتال دارای دقت حس نمودن عالی در تکرارهای مکرر می‌باشند.

معایب این سنسورها عبارتند از:

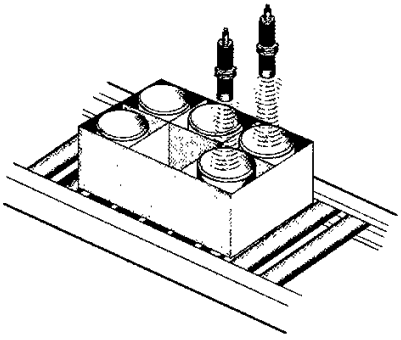
- سنسورهای آلتراسونیک باید یک سطح کاملاً چهار گوش جهت دریافت صحیح اکوی صوت را در مقابل خود ببینند.
- این سنسورها اگرچه در برابر اختلال مقاوم هستند، اما در برابر بعضی نویزها مانند صدای هیس در شیلنگ هوا و شیرهای اطمینان برقی که از صدای زیاد ایجاد می‌گردند آسیب پذیر هستند.
- تغییر در محیط مانند دما، فشار، رطوبت و . . . در پاسخ سنسور تأثیرگذار می‌باشد.

کاربردهای عملی سنسورهای آلتراسونیک

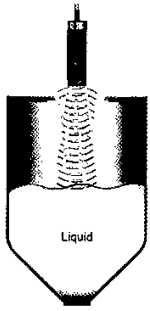
ازجمله کاربردهای عملی سنسورهای آلتراسونیک می‌توان به موارد زیر اشاره نمود:

- تشخیص فاصله
- تشخیص ارتفاع
- کنترل سطح
- تشخیص قطعات

در شکل۵-۳۰ نمونه هایی از کاربرد سنسورهای آلتراسونیک را مشاهده می‌نمایید.



تشخیص وجود شیشه در جعبه



تشخیص سطح مایع

شکل ۵-۳۰ نمونه کاربردهای سنسور التراسونیک

فصل ۵

۵-۴-۵ سنسور فتوالکتریک

این سنسورها با حس نمودن تغییر در میزان نور دریافتی، توسط یک آشکارساز نوری عمل می‌نمایند. تغییر در نور اجازه می‌دهد که سنسور بتواند حضور یا عدم حضور جسم، اندازه، شکل و رنگ آن و .... را تشخیص دهد. سنسورهای فتوالکتریک دارای قابلیت تشخیص دقیق جسم، بدون استفاده از کنتاکت‌های فیزیکی می‌باشند. این سنسورها قادر به تشخیص یک جسم هدف از فاصله‌ی (1 in) 2.45 cm تا صد متر یا بیشتر می‌باشند. این سنسور نیز دارای یک ناحیه کور کوچک است که اگر جسم از حدی به سنسور نزدیک‌تر باشد دیده نمی‌شود.

سنسورهای فتوالکتریک دارای انواع مختلفی می‌باشند که درهرکدام از یک روش منحصر به فرد برای سنسور استفاده شده است.

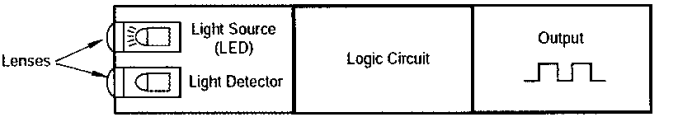
در شکل ۵-۳۱ یک نمونه سنسور فتوالکتریک را مشاهده می‌نمایید.

شکل ۵-۳۱ نمونه سنسور فتو الکتریک

ساختمان سنسور فتو الکتریک

یک سنسور فتوالکتریک شامل پنج قسمت اصلی زیر می‌باشد:

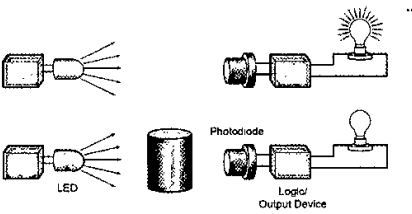
- منبع نور
- آشکارساز نور (دریافت کننده‌ی نور)
- لنز
- مدار منطقی
- خروجی



شکل ۵-۳۲ ساختمان سنسور فتوالکتریک

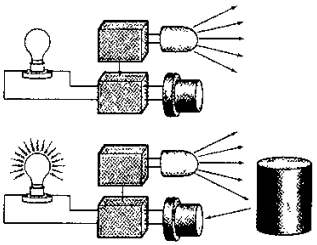
عملکرد

منبع نور، به طرف جسم، نوری را می‌فرستد. دریافت کننده‌ی نور مستقیماً یا از روی بازتاب نورِ تولیدی توسط منبع نور، متوجه‌ی حضور یا عدم حضور جسم در برابر سنسور می‌شود. تشخیص جسم در مقابل سنسور سبب به‌وجود آمدن سیگنال خروجی جهت سنسور می‌گردد.



شکل ۵-۳۳ سنسور فتوالکتریک با گیرنده و فرستنده مجزا

شکل فوق نوعی را نشان می‌دهد،که فرستنده و گیرنده نور مجزا هستند و به آن اصطلاحاً Transmitted Beam گفته می‌شود. در این نوع سنسور، در هر دو طرف نیاز به سیم‌کشی وجود دارد. نوع دیگری سنسور فتوالکتریک که به Retro reflective موسوم است وجود دارد که فرستنده و گیرنده، مانند شکل ۵-۳۴ در یک وسیله تعبیه شده است. در این نوع سنسور، سیم‌کشی در یک نقطه انجام می‌شود.



شکل ۵-۳۴ سنسور فتوالکتریک با گیرنده و فرستنده در یک طرف

در هر دو نوع فوق، برای جلوگیری از پخش شدن نور در فضا، به‌ویژه وقتی فاصله زیاد است از لنزهای خاصی در جلوی فرستنده و گیرنده استفاده می‌کنند که نور را متمرکز می‌نماید.

مزایا و معایب سنسور فتوالکتریک

مزایای این سنسورها عبارتند از:

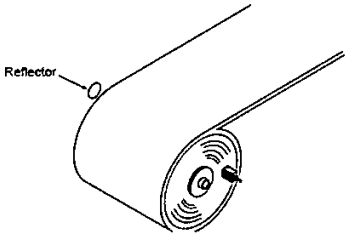
- امکان استفاده برای فاصله‌های زیاد.
- سرعت سوئیچینگ بالا از ۳۰ میکروثانیه تا ۳۰ میلی‌ثانیه.
- اطمینان بالا به‌ویژه وقتی شمارش سریع قطعات مورد نیاز است.
- میدان‌های مغناطیسی روی آنها بی‌اثر است.

معایب این سنسورها عبارتند از:

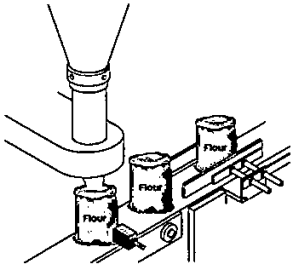
- در نوعی که فرستنده و گیرنده جداست تنظیم دقیق مورد نیاز است.
- قطعات کوچک توسط سنسور قابل حس کردن نیستند.
- دارای نقطه کور هستند. در نوع Retro reflective وجود جسم در فاصله‌ای کم نسبت به سنسور تشخیص داده نمی‌شود. ناحیه کور معمولاً بسته به نوع سنسور حدود ۲/۵ تا ۱۵ سانتی‌متر است.
- کثیف‌شدن چشمه‌های فرستنده و گیرنده مانع دید سنسور می‌شود.

نمونه‌ای از کاربردهای سنسور فتوالکتریک

در شکل ۵-۳۵ نمونه‌ای از کاربردهای سنسور فتوالکتریک نشان داده شده است.



کنترل قطر یک رول



آشکارسازی وجود کیسه

شکل ۵-۳۵ نمونه کاربردسنسور فتوالکتریک

۵-۴-۶ مقایسه ویژگی سنسورهای دیجیتال

جدول ۵-۳ ویژگی‌های سنسورهای مختلف دیجیتال را نشان می‌دهد.

جدول ۵-۳ ویژگی‌های سنسورهای مختلف دیجیتال

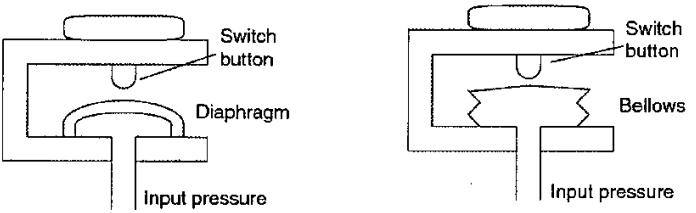
Sensor	• مزایا	• معایب	• نمونه کاربرد
Limit Switch	• توانایی عبور جریان زیاد از کنتاکت‌های آن • قیمت ارزان • کاربرد آسان	• تماس فیزیکی با شیء مورد نظر • سرعت پاسخ خیلی پایین • استهلاک بالا به دلیل تماس با اجسام	• به‌عنوان Inter lock • محدودکننده‌ی حرکت
Inductive	• مقاومت در برابر محیط های ناملایم (شرایط سخت) • توانایی پیش‌بینی با قابلیت تشخیص بالا • عمر زیاد • نصب آسان	• محدودیت فاصله • فقط تشخیص قطعات فلزی	• ماشین‌های صنعتی • ماشین ابزار
Capacitive	• قابلیت تشخیص • اجسام غیر فلزی	• حساسیت زیاد نسبت به تغییرات محیطی	• حس نمودن Level (سطح)
Ultrasonic	• تشخیص همه مواد	• حد تفکیک • تکرار پذیری	• جلوگیری از برخورد درهای برقی • کنترل سطح • ترمز
Photoelectric	• توانایی حس نمودن (تشخیص) همه انواع مواد • عمر زیاد • محدوده‌ی حس نمودن بالا • سرعت پاسخ بالا	• در معرض آلودگی قرار داشتن لنز سنسور • محدوده‌ی عملکرد آن بستگی به رنگ و بازتاب‌پذیری جسم مورد نظر دارد.	• در سیستم‌های بسته بندی • تشخیص مواد • تشخیص قطعات

۵-۴-۷ سوئیچ‌های فرآیندی (دما، فشار، فلو، سطح)

این سوئیچ‌ها برای حس‌کردن پارامترهایی مانند فشار، دما، سطح، فلو و امثال آنها در یک فرآیند به‌کار می‌روند. برخی از سنسورهای تماسی یا غیر تماسی که قبلاً تشریح شد را می‌توان در موارد خاصی استفاده کرد، ولی برخی از سنسورهای فرآیندی ساختار خاص دارند. در اینجا متداول‌ترین نوع آنها را شرح می‌دهیم.

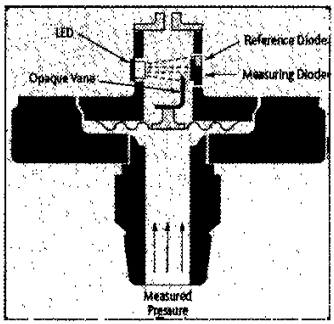
۵-۴-۷-۱ Pressure Switch

کنتاکت این سوئیچ در شرایطی که فشار از حد معینی بالاتر رود عمل می‌کند و می‌تواند مدار را قطع یا وصل نماید. ساختار این سوئیچ به شکل‌های متفاوتی طراحی شده است. دو نوع متداول آن Diaphragm و Bellow می‌باشد. این دو نوع، از ماده انعطاف‌پذیری ساخته شده است که با فرآیند مستقیماً تماس دارند. فشار می‌تواند در این قسمت تغییر حالت ایجاد کند و آنرا باز کند. وقتی فشار از حد معینی بیشتر شد، باز شدن این قسمت یک سوئیچ را تحریک می‌کند که می‌توان از کنتاکت‌های باز یا بسته آن استفاده نمود. شکل ۵-۳۶ این موضوع را نشان می‌دهد.



شکل ۵-۳۶ سوئیچ فشار با قسمت قابل ارتجاع

شکل ۵-۳۷ سنسور فشار از نوع نوری را نشان می‌دهد که در آن افزایش فشار باعث می‌شود که مانع متصل به قسمت انعطاف‌پذیر جابه‌جا شود و جلوی گیرنده نور قرار بگیرد.



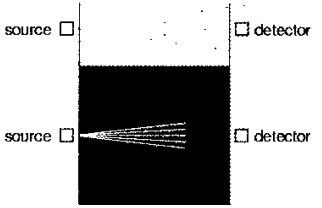
شکل ۵-۳۷ سوئیچ فشار نوری

فصل ۵

Level Switch ۲-۷-۴-۵

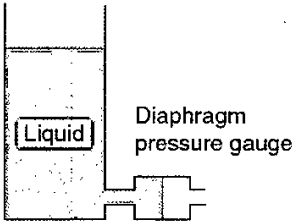
این سوئیچ وقتی سیال یک مخزن در ارتفاع مورد نظر قرار گرفت عمل می‌کند. انواع مختلفی برای آن عرضه شده است که عبارتست از:

- **سنسور پروکسمیتی خازنی:** همانطور که قبلاً ذکر شد می‌توان از سنسور خازنی برای برخی مواد و سیالات استفاده کرد. به شکل ۵-۲۶ رجوع کنید.
- **سنسور نوری:** برای برخی مواد و سیالات مانند شکل ۵-۳۸ وقتی ارتفاع به حد معینی برسد، مسیر نور قطع شده و سنسور عمل می‌کند.



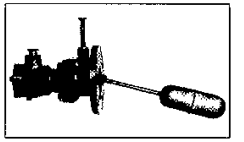
شکل ۵-۳۸ سنسور سطح از نوع نوری

- **سنسور فشار:** نوع دیگر در واقع یک Pressure Switch است که در پایین مخزن نصب می‌شود. بدیهی است هر چه ارتفاع سیال بیشتر باشد، فشار وارد بر دیافراگم بیشتر شده و اگر ارتفاع از حد معینی بیشتر شد عمل می‌کند.



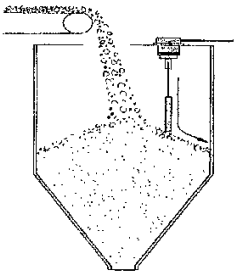
شکل ۵-۳۹ سنسور سطح از نوع سنجش فشار

- **نوع شناور:** وقتی ارتفاع به حد مشخصی رسید جابه‌جایی شناور باعث می‌شود که سنسور عمل نماید.



شکل ۵-۴۰ سنسور سطح از نوع شناور

- **نوع Conductive**: این نوع براساس سنجش هدایت الکتریک عمل می‌کند. وقتی پراب آن با ماده تماس پیدا کند با تغییر هدایت الکتریکی باعث فعال‌شدن سنسور می‌گردد.



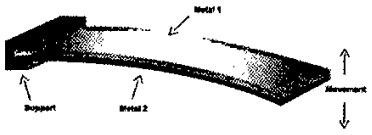
شکل ۵-۴۱ سنسور سطح از نوع کنداکتیو

- **نوع خازنی**: پس از تماس با جسم، با تغییر خازن بین پراب و جداره فلزی مخزن یا با تغییر خازن بین دو پراب آن سنسور عمل می‌کند.

شکل ۵-۴۲ سنسور سطح از نوع خازنی

Temperature Switch ۳-۷-۴-۵

این سوئیچ با تغییر دما فعال شده و انواع مختلفی برای آن عرضه شده است. شکل زیر نوع بی‌متال را نشان می دهد. بی‌متال از دو فلز با ضریب انبساطی مختلف ساخته شده که در اثر گرما خم شده و باعث فعال‌شدن سنسور می‌گردد.



شکل ۵-۴۳ سوئیچ دما از نوع بی‌متال

فصل ۵

Flow Switch ۴-۷-۴-۵

این سوئیچ در مسیر سیالات استفاده می‌شود و در صورت قطع‌شدن جریان عبور سیال (دبی) فعال می‌شود. این سنسور (سوئیچ) انواع مختلفی دارد که به برخی اشاره می‌شود:

- **نوع شناوری:** در این نوع، با ورود سیال به داخل فلوسوئیچ یک شناور جابه‌جا شده و یک کنتاکت را قطع یا وصل می‌کند که به مفهوم وجود فلو (عبور جریان سیال) است. با قطع‌شدن فلو، شناور پایین آمده و کنتاکت قطع می‌شود.
- **نوع پدالی:** با برخورد سیال به پدال سوئیچ، کنتاکت الکتریکی وصل می‌شود. در صورت عدم وجود فلو، پدال به حالت اولیه بر می‌گردد و کنتاکت قطع می‌شود.
- **نوع مقاومتی:** دو پراب دارد که در داخل سیال قرار می‌گیرد. وقتی با تغییر فلو یا قطع شدن فلو مقاومت بین پراب‌ها تغییر کند، سنسور فعال می‌شود.

فلو سوئیچ شناوری

فلو سوئیچ پدالی

شکل ۵-۴۴ فلو سوئیچ

۵-۵ سیگنال‌های خروجی دیجیتال

PLC پس از پردازش برنامه‌ی کاربر باید بتواند دستوراتی را به ادوات تحت کنترل ارسال نماید. این دستورات به‌صورت سیگنال‌های الکتریکی به تجهیز تحت کنترل ارسال می‌گردد. نوع سیگنال ارسالی می‌تواند دیجیتال یا آنالوگ باشد. در این بخش ادواتی که می‌توانند سیگنال خروجی دیجیتال دریافت نمایند را تشریح می‌کنیم.

۵-۵-۱ محرک‌های منطقی[1]

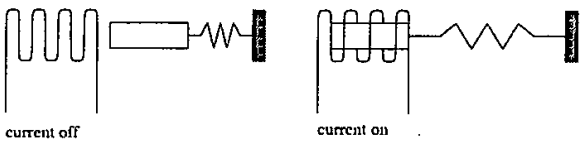
محرک‌ها ایجاد کننده حرکت در سیستم‌های مکانیکی می‌باشند. این عمل، اغلب با تبدیل انرژی الکتریکی به برخی از حرکت‌های مکانیکی صورت می‌پذیرد. از جمله محرک‌هایی که به منظور عملکرد خود نیاز به سیگنال الکتریکی دیجیتال دارند، می‌توان به سلونوئید، سلونوئید ولو و کنتاکتور اشاره نمود.

علاوه بر محرک‌های مکانیکی، ادوات الکتریکی نیز وجود دارند که به‌عنوان مصرف کننده‌ی دیجیتال شناخته می‌شوند، از جمله:

- هیتر
- لامپ
- رله‌های فرمان

۵-۵-۲ سلونوئیدها

اصول عملکرد سلونوئیدها بر اساس حرکت یک هسته‌ی آهنی (پیستون) که می‌تواند داخل یک سیم‌پیچ حرکت نماید بنا شده است. این موضوع در شکل ۵-۴۸ نشان داده شده است. در شرایط نرمال، پیستون توسط یک فنر خارج از سیم‌پیچ قرار گرفته است. هنگامی که ولتاژی به سیم پیچ اعمال شده و جریان الکتریکی در آن جاری شود؛ یک میدان مغناطیسی در اطراف سیم‌پیچ به‌وجود می‌آید که بر پیستون اثر گذاشته و آنرا به داخل سیم‌پیچ می‌کشد. پیستون می‌تواند جهت ایجاد یک نیروی خطی استفاده گردد. از جمله کاربردهای سلونوئید می‌توان به ولوهای پنوماتیکی و بازکننده‌های درب اتومبیل اشاره نمود. اکثر سلونوئیدهای صنعتی دارای ولتاژ تغذیه‌ی 24 VDC بوده و جریانی در حد چند میلی‌آمپر دریافت می‌کنند، از اینرو اغلب می‌توان آنها را مستقیماً به کارت‌های خروجی دیجیتال در PLC متصل نمود.



شکل ۵-۴۵ عملکرد سلونوئید

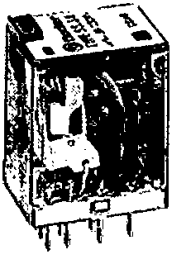
فصل ۵

۵-۵-۳ ولوها

جریان سیالات و هوا می‌تواند توسط ولوهای کنترلی و ولوهای OFF/ON کنترل شود. ولوها خود به انواع مختلفی تقسیم‌بندی می‌شوند. آن دسته از ولوها که به‌عنوان یک مصرف‌کننده‌ی دیجیتال شناخته می‌شوند، سلونوئید ولوها می‌باشند. سلونوئید ولو دارای یک سیم‌پیچ و یک سیلندر مخصوص است. هنگامی که به سیم‌پیچ ولتاژی اعمال گردد، در اثر میدان مغناطیسی به‌وجود آمده در آن، سیلندر به طرف داخل سیم‌پیچ کشیده شده و مسیر عبور هوا یا سیال باز می‌شود.

1. Logical Actuators

رله‌های فرمان: اصولی‌شبیه‌کنتاکتورداشته‌ولی‌جهت‌جریان‌های‌پایین‌طراحی‌شده‌اند. اکثر آنها دارای ولتاژ تغذیه 24 V DC و جریان مصرفی در حد میلی‌آمپر بوده و از اینرو قابلیت اتصال مستقیم به کارت‌های خروجی دیجیتال در PLC را دارا می‌باشند. از آنها معمولاً جهت ایزوله‌سازی سیگنال‌های ورودی به PLC استفاده می‌شود. همچنین در شرایطی که ولتاژ تجهیز تحت کنترل با ولتاژ کارت خروجی دیجیتال در PLC یکسان نباشد، از این رله‌ها به‌عنوان واسط استفاده می‌شود. در این‌حالت‌سیگنال‌های‌خروجی PLC به آنهاداده‌شده‌وازکنتاکت‌های‌این رله‌ها سیگنال مورد نیاز جهت تجهیز دریافت می‌گردد.



شکل ۵-۵۱ نمونه رله

۵-۶ سیگنال‌های آنالوگ ورودی

بر خلاف سیگنال‌های دیجیتال که ماهیتی گسسته بین دو مقدار صفر منطقی و یک منطقی دارند، سیگنال‌های آنالوگ ماهیتی پیوسته داشته و از سنسورهایی می‌آیند که به منظور حس نمودن یک کمیت فیزیکی با ماهیت پیوسته استفاده شده‌اند. برخی از کمیت‌های مهم فیزیکی که ماهیتی پیوسته دارند عبارتند از:

- دما
- فشار
- ارتفاع سطح (Level)
- فلو (جریان مایعات)

روش‌های اندازه‌گیری کمیت‌های فوق که از پرکاربردترین کمیت‌های آنالوگ هستند در ادامه توضیح داده می‌شود. کمیت‌های دیگر مانند وزن، موقعیت، رطوبت و ... نیز به‌صورت آنالوگ اندازه‌گیری می‌شوند که تشریح آنها خارج از چارچوب این کتاب است.

به‌منظور اندازه‌گیری و حس نمودن کمیت‌های آنالوگ از وسایلی استفاده می‌شود که خروجی آنها یک سیگنال الکتریکی آنالوگ می‌باشد. جنس سیگنال خروجی این سنسورها عموماً از نوع ولتاژ یا جریان یا مقاومت می‌باشد، هر وسیله اندازه‌گیری آنالوگ از دو بخش تشکیل می شوند:

- **ترانسدیوسر:** مبدلی است که کمیت مورد اندازه‌گیری را حس کرده و نتیجه را به سیگنال‌های الکتریکی تبدیل می‌نماید.
- **ترانسمیتر:** وسیله‌ای است که سیگنال دریافتی از ترانسدیوسر را دریافت کرده و آنرا به سیگنال الکتریکی استاندارد جهت ارسال به کنترلر تبدیل می‌کند. لازم است توجه شود که هر ولتاژ و جریانی را نمی‌توان به PLC داد و لازم است سیگنال ارسالی به PLC در بازه استاندارد باشد. امروزه مفهوم ترانسمیتر فراتر رفته و

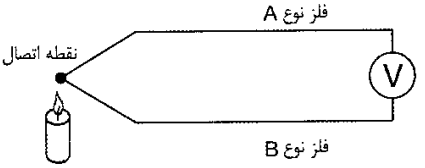
ترانسدیوسر را نیز دربر می‌گیرد، این دو در یک وسیله مونتاژ شده و با عنوان ترانسمیتر عرضه می‌شوند. به‌عنوان مثال سنسور فشارسنج و ترانسمیتر مربوطه در یک مجموعه تحت عنوان ترانسمیتر فشار ارائه می‌گردد که هم می‌تواند فشار را اندازه‌گیری نماید و هم آنرا به‌صورت سیگنال الکتریکی استاندارد به کنترلر تحویل دهد.

۵-۶-۱ سنسورهای اندازه‌گیری دما

معروف‌ترین و پرکاربردترین سنسورهای اندازه‌گیری دما، ترموکوپل و RTD می‌باشند. در ادامه به تشریح این دو نوع سنسور خواهیم پرداخت.

ترموکوپل

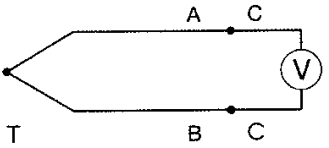
ترموکوپل از دو فلز غیر همجنس که در یک نقطه به یکدیگر متصل شده‌اند تشکیل می‌گردد.



شکل ۵-۵۲ عملکرد ترموکوپل

عملکرد ترموکوپل

با توجه به میزان دما در نقطه‌ی اتصال، ولتاژی در دو سر آزاد ترموکوپل ایجاد می‌گردد که معرف دمای محل اتصال است. هرگاه فاصله‌ی ترموکوپل از تجهیز اندازه‌گیری یا از PLC زیاد باشد و بخواهیم آنرا به PLC یا تجهیز مورد نظر متصل نماییم، ناگزیر به استفاده از کابل انتقال می‌باشیم. اگر جنس کابل انتقال که به ترموکوپل وصل می‌شود با جنس فلزات ترموکوپل یکسان باشد، مقدار اندازه‌گیری شده توسط ترموکوپل صحیح بوده و می‌توان از آن استفاده نمود.
ولی اگر جنس کابل انتقال با جنس فلزات ترموکوپل یکسان نباشد، در محل اتصال کابل با ترموکوپل، دو ترموکوپل جدید نیز ایجاد خواهند شد که در نهایت اندازه‌گیری ترموکوپل را دچار خطا خواهند نمود.



شکل ۵-۵۳ نقاط اتصال ترموکوپل

برای رفع این مشکل می‌توان از جبران‌ساز استفاده نمود. بدین ترتیب که دما در نقطه‌ی اتصال کابل انتقال به ترموکوپل محاسبه و از دمای اندازه‌گیری شده توسط ترموکوپل کم می‌شود. (جبران‌سازی خارجی)

فصل ۵

انواع ترموکوپل: ترموکوپل‌ها با توجه به جنس فلزات مورد استفاده در آن به‌صورت نشان داده شده در جدول ۵-۴ تقسیم‌بندی می‌شوند.

جدول ۵-۴ انواع ترموکوپل

Thermocouple	Conductor		Temperature	Voltage Range
Type	Positive	Negative	Range (°C)	(mV)
E	Chromel	Constantan	-270° to 1.000°	-9.835 to 76.358
J	Iron	Constantan	-210° to 1.200°	-8.096 to 69.536
K	Chromel	Alumel	-270° to 1.372°	-6.548 to 54.874
T	Copper	Constantan	-270° to 400°	-6.258 to 20.869
S	Platinum-10% Rhodium	Platinum	-50° to 1.768°	-0.236 to 18.698
R	Platinum-13% Rhodium	Platinum	-50° to 1.768°	-0.226 to 21.108

تبدیل سیگنال ترموکوپل

از آنجایی که در استفاده از ترموکوپل، معمولاً ناگزیر به استفاده از جبران‌سازی و انجام تنظیمات خاص بوده و سیگنال خروجی ترموکوپل بسیار نویزپذیر می‌باشد؛ در برخی کاربردهای صنعتی سیگنال ترموکوپل را به یک سیگنال الکتریکی استاندارد تبدیل می‌کنند. این عمل توسط ترانسدیوسرهای مخصوصی انجام می‌پذیرد. با استفاده از این ترانسدیوسرها می‌توان سیگنال خروجی ترموکوپل را به سیگنال جریانی یا ولتاژی تبدیل نمود.

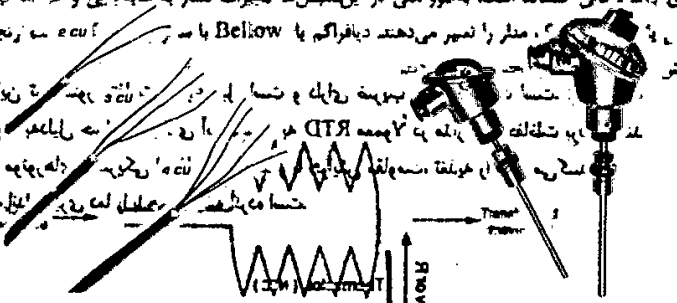
شکل ۵-۵۴ مبدل سیگنال ترموکوپل به سیگنال جریانی

RTD [1]

این سنسورها از سنسورهای مقاومتی دما می‌باشند، یعنی با تغییر دما میزان مقاومت آنها تغییر می‌کند. رنج تغییرات در RTDها تقریباً به‌صورت خطی می‌باشد و این موضوع باعث بالا رفتن دقت آنها در مقایسه با ترموکوپل در دماهای پایین شده است. توجه شود که اثر تغییرات دما در ترموکوپل به‌گونه‌ای است که در دماهای بالا ترموکوپل به دقت لازم می‌رسد. البته باید توجه نمود که رنج کاری RTD دماهای پایین‌تری را نسبت به ترموکوپل پوشش می‌دهد. (در حدود 200 c تا 850 c)

1. Resistance Temperature Detectors

RTD ها در سه نوع دو سیمه، سه سیمه و چهار سیمه ساخته می‌شوند.

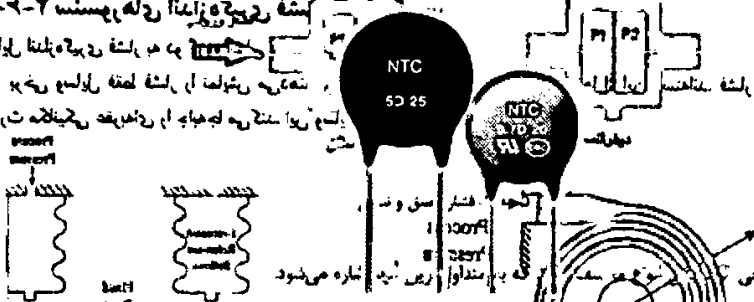


شکل ۵-۵۵ RTD و اتصالات آن

RTD ها می‌توانند مستقیماً به کارت‌های آنالوگ خاص خودشان متصل شوند، در عین حال که می‌توان با استفاده از ترانسدیوسر سیگنال RTD را به یک سیگنال استاندارد الکتریکی از نوع جریان یا ولتاژ تبدیل و سپس آنرا به کارت ورودی آنالوگ اعمال نمود.

ترمیستور

ترمیستورها نوعی مبدل نیمه‌هادی برای اندازه‌گیری دما هستند. ترمیستورها به دو نوع تقسیم می‌شوند:



شکل ۵-۵۶ ترمیستور

نوع NTC

مقاومت‌های نیمه هادی به‌کار رفته در این ترمیستور از اکسیدهای فلزی و دارای ضریب حرارتی منفی است. از اینرو به آنها NTC[1] می‌گویند. یعنی هر چه دما بالاتر رود مقدار مقاومت کاهش می‌یابد. به‌عنوان نمونه در 0°C مقاومت ۱۰ kΩ و در 100°C مقاومت 200Ω است. همانطور که مشاهده می‌شود، رنج تغییرات مقاومت در یک بازه

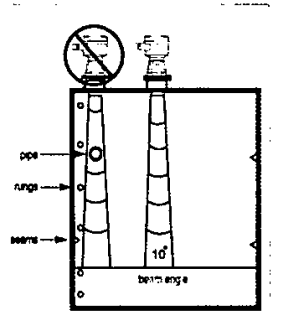
1. Negative Temperature Coefficient

سنسور نوع التراسونیک

در این نوع سنسورها یک پالس صوتی تولید شده و به طرف محلی که ارتفاع سطح آن می‌خواهد اندازه‌گیری شود ارسال می‌گردد. هنگامی که پالس صوتی با سطح کمیت مورد اندازه‌گیری برخورد نماید، بازتاب پیدا می‌نماید. صوت بازتاب شده (اکو) توسط سنسور دریافت می‌گردد. زمان برخورد یک اشعه صوتی با کمیت مورد اندازه‌گیری و بازتاب آن، نسبت مستقیم با فاصله‌ی سنسور از کمیت مورد نظر دارد. سیگنال خروجی این سنسور متناسب با میزان ارتفاع سطح اندازه‌گیری شده می‌باشد.

همانند سایر سنسورهای آنالوگ، این نوع سنسورها نیز معمولاً دارای سیگنال خروجی 4 mA الی 20 mA و یا ولتاژی (0 V الی 10 V) می‌باشند.

در شکل ۵-۷۰ نحوه قرارگیری صحیح یک سنسور آلتراسونیک در یک مخزن را مشاهده می‌نمایید.



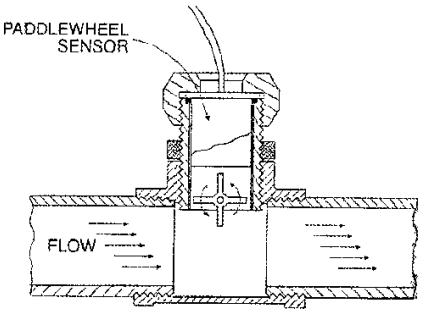
شکل ۵-۷۰ سنسور سطح از نوع التراسونیک

۵-۶-۴ سنسورهای اندازه‌گیری فلو

به منظور اندازه‌گیری فلوی عبوری مایعات، از تجهیزی به نام فلو ترنسمیتر استفاده می‌گردد. این تجهیز از یک سنسور اندازه‌گیری فلو و یک ترانسمیتر تشکیل شده است. روش‌های اندازه‌گیری فلو دارای تنوع زیادی بوده و هر کدام از فلو ترانسمیترهای موجود در صنعت بر اساس یکی از این روش‌ها اقدام به اندازه‌گیری فلو می‌نماید.

نوع پره چرخان

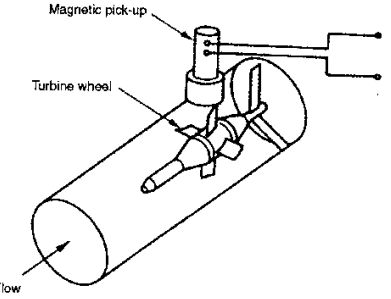
در این نوع، عبور فلو یک پره را می‌چرخاند که منجر به تولید پالس می‌شود. هر چقدر فلو بیشتر باشد فرکانس پالس بیشتر است. یک مدار واسط این فرکانس متغیر را به ولتاژ DC با سطح متغیر تبدیل می‌کند.



شکل ۵-۷۱ سنسور فلو از نوع پره‌ای

نوع توربینی

در این نوع، توربینی در مسیر سیال قرار می‌گیرد که پره‌های آن با عبور فلو می‌چرخد. با استفاده از یک سنسور پروکسمیتی از بیرون می‌توان پالس‌هایی را دریافت کرد که فرکانس آنها بستگی به فلوی عبوری دارد. می‌توان این پالس‌ها را به ولتاژ DC تبدیل کرده و به PLC انتقال داد. در برخی نمونه‌ها از نور برای سنجش سرعت چرخش توربین استفاده می‌شود.

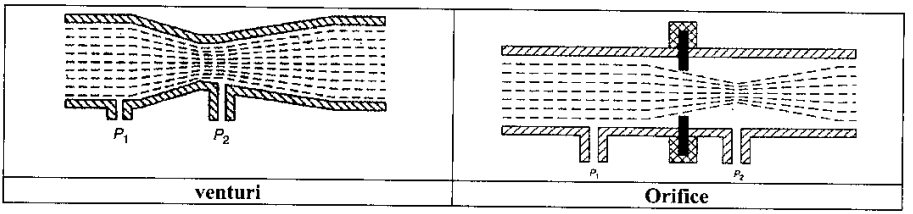


شکل ۵-۷۲ سنسور فلو از نوع توربینی

فصل ۵

نوع اختلاف فشار

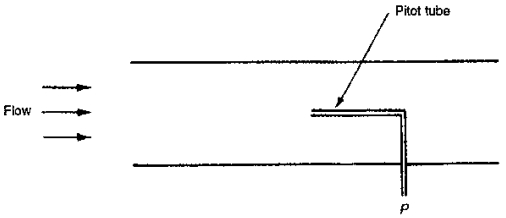
با استفاده از Orifice یا Venturi، مسیر سیال را را تنگ‌تر می‌کنند. این عمل منجر به اختلاف فشار بین دو طرف می‌شود که این اختلاف فشار با فلوی عبوری تناسب دارد.



شکل ۵-۷۳ اندازه‌گیری فلو با ایجاد اختلاف فشار دو طرف

Pitot Tube

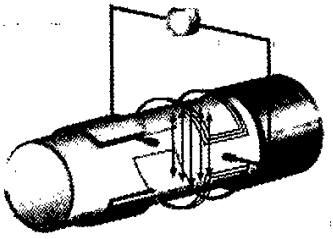
در این روش با قرار دادن انشعاب لوله‌ای شکل در مسیر فلو مانند شکل زیر فشار حس می‌شود که بااندازه گیری اختلاف فشار استاتیک و فشار مورد نظر می‌توان فلو را اندازه‌گیری نمود.



شکل ۵-۷۴ سنسور فلو از نوع Pitot Tube

نوع القایی

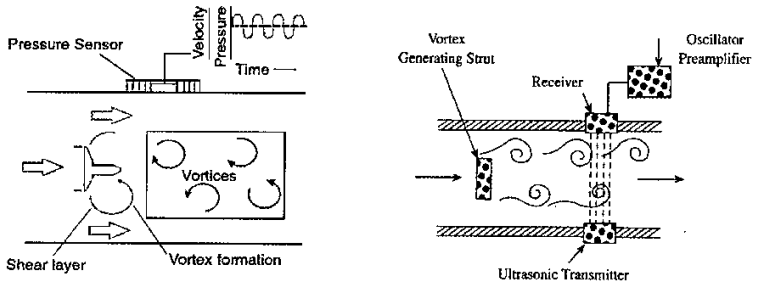
در این روش عبور سیال رسانا از داخل یک سیم‌پیچ که در دور لوله سیال تعبیه شده است منجر به تولید ولتاژ در دوسر آن می‌شود. هرچه سرعت فلو بیشتر باشد ولتاژ القایی بیشتر است. این روش هیچ قطعه یا پره‌ای که با سیال تماس پیدا کند ندارد و مشکلات سایش و خوردگی و ... در آن مطرح نیست، از اینرو دقت بالایی دارد. تنها عیب این روش آن است که برای سیالات رسانا قابل استفاده است.



شکل ۵-۷۵ سنسور فلو از نوع القایی

نوع گردابی Vortex

این روش برای اندازه‌گیری فلوی گاز و بخار و مایعات استفاده می‌شود. وقتی سیال در مسیر خود به مانعی برخورد می‌کند، پس از ترک مانع در اطراف آن حلقه‌های گردابی ایجاد می‌شود. هرچه سرعت فلو بیشتر باشد تعداد حلقه‌ها در واحد زمان بیشتر است.



شکل ۵-۷۶ سنسور فلو از نوع Vortex

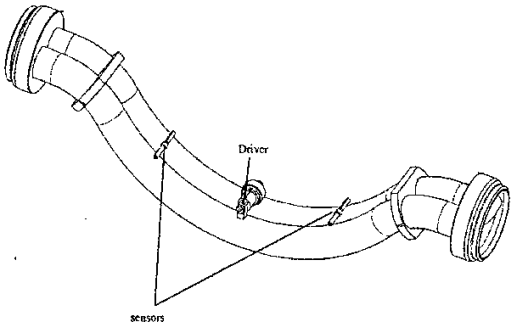
این گردابی یا آشفتگی در سیال که اصطلاحاً Turbulance خوانده می‌شود، به‌طریق الکتریکی قابل اندازه‌گیری است. با عبور دادن صوت یا نور تغییرات فشار به‌صورت پالس ظاهر می‌شود که می‌تواند به ولتاژ متغیر DC تبدیل گردد.

نوع Mass Flow

این روش برای اندازه‌گیری دقیق فلو با خطای بسیار کم به‌کار می‌رود. دما و فشار و چگالی سیال تأثیری در اندازه‌گیری ندارد. این روش خود به دو دسته نوع تقسم می‌شود:

Coriolis Mass Flow Meter

در این دوش در مسیر عبور فلو دو لوله خمیده به شکل U یا شکل‌های دیگر قرار می‌دهند. جنس این لوله‌ها متنوع است، مثلاً می‌تواند از جنس استیل باشد. خاصیت الاستیکی دو لوله باعث می‌شود که بتوانند تا حدی جابه‌جا شوند. بین دو لوله یک لرزاننده مغناطیسی وجود دارد که هر دو را می‌لرزاند (Driver در شکل ۵-۷۷). سنسورهایی نیز در سمت ورودی و خروجی روی لوله‌ها تعبیه شده‌اند.



شکل ۵-۷۷ سنسور فلو از نوع کوریولیس

فصل ۵

در حالتی‌که فلو وجود ندارد، لرزش هر دو لوله با هم هماهنگ و متقارن است؛ این بدان معنی است که سیگنالی که هر دو سنسور می‌فرستند هیچ اختلاف فازی با هم ندارد.

عدم تقارن وقتی فلو وجود دارد. تقارن دو لوله وقتی فلو وجود ندارد.

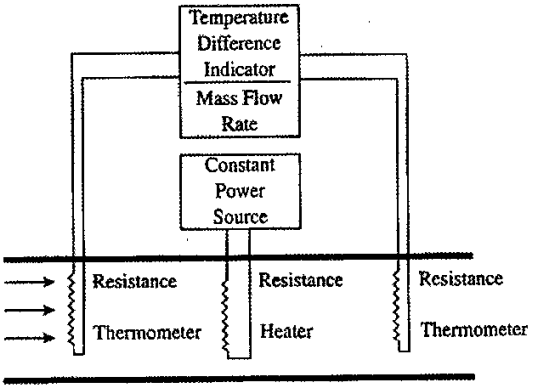
شکل ۵-۷۸ عملکرد کوریولیس

وقتی فلو جریان می‌یابد، لرزش دو لوله شکل ناهماهنگ پیدا می‌کند که جهت خمیدگی آنها نامتقارن است. هرچه فلو بیشتر باشد این عدم تقارن بیشتر است، این پدیده براساس نیروی موسوم به Coriolis می‌باشد. شکل ۵-۷۸ لحظه‌ای از لرزش دو لوله را در حین عبور فلو نشان می‌دهد.

در این حالت سیگنالی که دو سنسور می‌فرستند با هم اختلاف‌فاز پیدا می‌کند. هرقدر جریان فلو بیشتر باشد، این اختلاف فاز بیشتر است. مدار الکترونیکی، این اختلاف‌فاز را به سیگنال استاندارد ولتاژی یا جریانی تبدیل کرده و به PLC ارسال می‌کند.

Thermal Mass Flow

در این روش در دو طرف مسیر عبور سیال از دو RTD حساس استفاده می‌شود. یکی از RTDها با عبور جریان سیال گرم می‌شود و RTD سمت مقابل دما را اندازه‌گیری می‌کند. بدیهی است وقتی فلو وجود ندارد بیشترین دما حس می‌شود. وقتی جریان فلو برقرار می‌شود، منجر به سرد شدن RTD منبع حرارت می‌گردد. هرچه فلو بیشتر باشد، کاهش دما بیشتر توسط RTD سمت مقابل حس می‌شود. این تغییرات به سیگنال الکتریکی استاندارد تبدیل شده و در اختیار PLC قرار می‌گیرد.

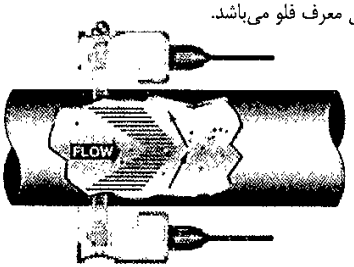


شکل ۵-۷۹ سنسور فلو از نوع حرارتی

در برخی نمونه‌ها در سمت گرم‌کننده کنترلری در نظر گرفته می‌شود که هرقدر مقاومت در اثر فلو سرد شد، جریان بیشتری از آن عبور کند تا دما ثابت بماند. در این سیستم، تغییرات جریان هیتر معرف دبی سیال خواهد بود.

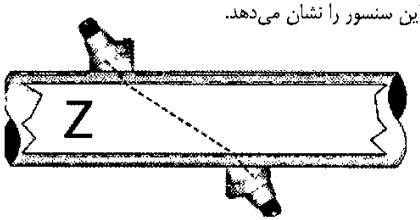
نوع Ultrasonic

نمونه‌ای از سنسورهای آلتراسونیک نیز وجود دارد که بر اساس پدیده دوپلر کار می‌کند. تغییر و شیفت فرکانس بین صوت ارسالی و صوت دریافتی در سمت مقابل معرف فلو می‌باشد.



شکل ۵-۸۰ سنسور فلو از نوع التراسونیک دوپلر

در نوع دیگری از سنسورهای آلتراسونیک، زمان گذر صوت اندازه‌گیری می‌شود که به نوع Transit Time موسوم است. شکل ۵-۸۱ این عملکرد این سنسور را نشان می‌دهد.



شکل ۵-۸۱ سنسور فلو از نوع التراسونیک زمانی

فصل ۵

۵-۷ سیگنال آنالوگ خروجی

برخی از ادوات صنعتی به‌منظور عملکرد خود نیاز به سیگنال‌های الکتریکی پیوسته (آنالوگ) دارند. از جمله این ادوات می‌توان به ولوهای کنترلی و درایوهای کنترل‌دور موتورها اشاره نمود. سیگنال الکتریکی آنالوگ مورد نیاز این تجهیزات به صورت ولتاژی یا جریانی می‌باشد.

۵-۷-۱ ولوهای کنترلی

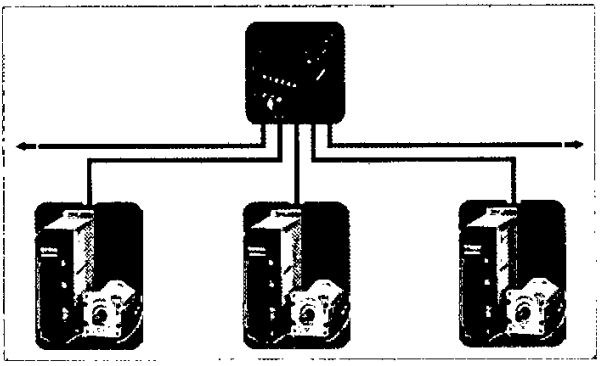
همانگونه که قبلاً اشاره گردید وظیفه ولوها باز و بسته نمودن مسیر عبور سیال یا هوا در سیستم‌های مختلف صنعتی از جمله سیستم‌های هیدرولیک و پنوماتیک می‌باشد. برخلاف سلونوئیدولوها که یک مسیر را کاملاً باز یا کاملاً بسته می‌نمودند، گروهی از ولوها وجود دارند که می‌توان میزان باز وبسته نمودن مسیر توسط آنها را کنترل نمود؛ به این نوع ولوها کنترل‌ولو می‌گویند. در شکل ۵-۸۶ یک نمونه کنترل‌ولو را مشاهده می‌نمایید.

شکل ۵-۸۲ نمونه ولو کنترلی

کنترل‌ولوها به‌منظور باز نمودن تدریجی دریچه، نیاز به سیگنال الکتریکی پیوسته (آنالوگ) دارند. اکثر کنترل‌ولوهای صنعتی با سیگنال 4 mA الی 20 mA کار می‌کنند.

۵-۷-۲ درایو کنترل‌دور موتورهای الکتریکی

با پیشرفت تکنولوژی ساخت نیمه‌هادی‌های قدرت، درایوهای کنترل سرعت موتورهای الکتریکی به سرعت جایگزین مدارات قدیمی‌شدند. از جمله درایوهای پرکاربرد حال حاضر در صنعت می‌توان به درایو کنترل دور موتور سه‌فاز القایی اشاره نمود. این درایوها توسط کنترل همزمان ولتاژ و فرکانسِ کار موتور، اقدام به کنترل همزمان سرعت و گشتاور موتور می‌نمایند. اکثر این درایوها دارای ترمینال‌های مخصوصی جهت اعمال سیگنال الکتریکی آنالوگ به‌منظور تنظیم فرکانس توسط درایو می‌باشند؛ که این سیگنال معمولاً یک سیگنال ولتاژی مانند 0 V الی 10 V می‌باشد. در بسیاری از درایوهای صنعتی نوین امکان اتصال به شبکه‌های صنعتی مانند پروفی‌باس نیز وجود دارد.



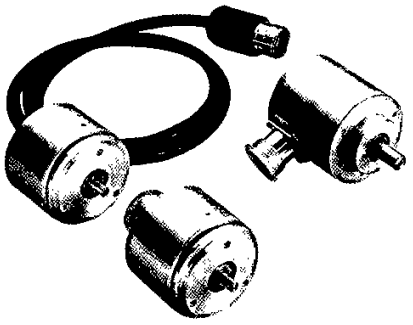
شکل ۵-۸۳ نمونه اتصال PLC به درایو

۵-۸ سیگنال‌های خاص

انکودر

انکودر از جمله تجهیزاتی است که در گروه‌بندی انجام شده برای سیگنال‌های الکتریکی جزو هیچ‌کدام از گروه‌ها قرار نمی‌گیرد. در عین حال با توجه به سیگنال خروجی آن، می‌توان انکودر را نزدیک به سنسورهای دیجیتال در نظر گرفت.
انکودر وسیله‌ای است که به ازای هر دور چرخش می‌تواند تعداد پالس معینی ایجاد نماید. از اینرو می‌توان آنرا بر روی محور موتور یا هر محور گردان وصل نموده و شمارش دور یا کنترل موقعیت را انجام داد.

از نظر سیگنال خروجی، سیگنال خروجی انکودر یک سیگنال دیجیتال محسوب می‌گردد که سریعاً قطع و وصل می‌شود. این سیگنال در برخی از نمونه‌های انکودر سیگنال 5V DC و در برخی از نمونه‌ها سیگنال 24V DC می‌باشد.



شکل ۵-۸۴ انکودر

۵-۹ پرسش و تحقیق

- مزایاو معایب و کاربردهای انواع سنسورهای آنالوگ را بر حسب کمیت مورد اندازه‌گیری با یکدیگر مقایسه کنید.

۵-۱۰ تست‌های خودآزمایی

۱- کدام گزینه برای یک سنسور صحیح است؟

الف) سنسور یک کمیت فیزیکی را به یک کمیت الکتریکی قابل اندازه‌گیری تبدیل می‌کند.

ب) سنسور اطلاعات مربوط به فرآیند را حس نموده و به کنترل‌کننده گزارش می‌کند.

ج) معمولاً سنسور به خودی خود فرمانی صادر نمی‌کند و اطلاعات را به کنترل‌کننده منتقل می‌کند.

د) هر سه مورد

۲- کدام تعریف برای یک سیگنال دیجیتال درست است؟

الف) دو وضعیت قطع و وصل دارد. ب) به‌صورت پیوسته است.

ج) فقط به‌صورت پالس است. د) هیچ‌کدام

۳- کدام گزینه جزو تجهیزات دیجیتال محسوب نمی‌شود؟

الف) شستی‌ها ب) کلیدها ج) لیمیت سوئیچ د) RTD

فصل ۵

۴- کدام گزینه برای یک کنتاکت باز (NO) صحیح است؟

الف) در شرایط نرمال باز است و در شرایط عملکردی نیز باز است.

ب) در شرایط نرمال بسته است و در شرایط عملکردی باز می‌شود.

ج) در شرایط نرمال باز است و در شرایط عملکردی بسته می‌شود.

د) در شرایط نرمال بسته است و در شرایط عملکردی نیز بسته است.

۵- تفاوت کلید و پوش‌باتون در چیست؟

الف) کلید ثابت است ولی پوش باتون با رها شدن به حالت نرمال برمی‌گردد.

ب) پوش باتون ثابت است ولی کلید با رها شدن به حالت نرمال برمی‌گردد.

ج) کلید و پوش باتون ثابت هستند و حالت خود را حفظ می‌کنند.

د) کلید و پوش باتون با رها شدن به حالت نرمال خود بر می‌گردند.

۶- لیمیت‌سوئیچ جزو کدام گروه از سنسورها قرار می‌گیرد؟

الف) آنالوگ – تماسی ب) دیجیتال – تماسی

ج) آنالوگ – غیرتماسی د) دیجیتال – غیرتماسی

۷- کاربرد لیمیت سوئیچ در چه مواردی است؟

الف) تشخیص وجود جسم ب) تشخیص حرکت جسم

ج) محدود کننده‌ی حرکت د) هرُ سه مورد

۸- کدام مورد جزو مزایای لیمیت‌سوئیچ نمی‌باشد؟

الف) استفاده‌ی آسان ب) عملکرد ساده و قابل مشاهده

ج) مقاومت زیاد در برابر شرایط محیطی مختلف در صنعت د) فرسایش قسمت‌های متحرک

۹- در مورد سنسور القایی کدام گزینه صحیح است؟

الف) این سنسور همه اجسام را تشخیص می‌دهد. ب) از فواصل زیاد نیز می‌تواند اجسام را تشخیص دهد.

ج) فقط فلزات را تشخیص می‌دهد. د) در شرایط محیطی سخت نمی‌تواند به‌کار رود.

۱۰- سنسور القایی بر چه اساسی کار می‌کند؟

الف) میدان الکتریکی ب) میدان مغناطیسی

ج) میدان الکترواستاتیکی د) هر سه مورد

۱۱- سنسورهای القایی بدون شیلد نسبت به نوع شیلددار دارای حساسیت . . . هستند.

الف) بیشتر ب) کمتر ج) یکسان د) نصف

۱۲- استفاده از سنسور خازنی برای تشخیص اجسام . . . مناسب‌تر است.

الف) فلزی ب) غیر فلزی ج) همه اجسام د) فقط اجسام فلزی

۱۳- ساخت سنسور خازنی نسبت به سنسور القایی. . .

الف) آسان‌تر است. ب) یکسان است. ج) سخت‌تر است. د) هیچ‌کدام

۱۴- توسط سنسور خازنی، اجسام با ثابت دی‌الکتریک بزرگتر . . .

الف) زودتر شناسایی می‌شوند. ب) دیرتر شناسایی می‌شوند.

ج) شناسایی نمی‌شوند. د) هیچ‌کدام

۱۵- مکانیزم عملکرد سنسور آلتراسونیک بر چه اساسی است؟
الف) میدان مغناطیسی ب) امواج صوتی ج) امواج راداری د) مادون قرمز
۱۶- کدام تعریف برای ناحیه کور در یک سنسور درست است؟
الف) در این ناحیه اجسام به‌خوبی توسط سنسور شناسایی می‌شوند.
ب) ناحیه‌ای است که فاصله‌ی جسم با سنسور از حد مجاز بیشتر باشد.
ج) ناحیه‌ای است که فاصله جسم با سنسور از حد مجاز کمتر باشد.
د) هیچ‌کدام
۱۷- مکانیزم عملکرد سنسور فتوالکتریک بر چه اساسی است؟
الف) میدان مغناطیسی ب) امواج صوتی ج) ارسال نور د) مادون قرمز
۱۸- در صورتی‌که فشار از حد معینی بیشتر شود، این سنسور (شستی فرآیندی) عمل نموده و یک سیگنال دیجیتال را قطع یا وصل می‌کند.
الف) Pressure Switch ب) Flow Switch ج) Thermo Switch د) Level Switch
۱۹- در صورتی‌که دما از حد معینی بیشتر شود، این سنسور (شستی فرآیندی) عمل نموده و یک سیگنال دیجیتال را قطع یا وصل می‌کند.
الف) Pressure Switch ب) Flow Switch
ج) Thermo Switch د) Level Switch
۲۰- در مورد سوئیچ‌های فرآیندی کدام مورد صحیح است؟
الف) سوئیچ توسط تغییر در یک کمیتِ فرآیند عمل می‌نماید.
ب) سوئیچ توسط اپراتور فعال و غیرفعال می‌گردد.
ج) فقط در صورت تغییر در فشار، سوئیچ عمل می‌نماید.
د) هر سه مورد
۲۱- جریان سیالات یا هوا توسط کدام مورد زیر می‌تواند کنترل شود؟
الف) ولوهای ON/OFF ب) ولوهای کنترلی ج) سلونوئید ولوها د) هر سه مورد
۲۲- رله‌های فرمانی و کنتاکتورها جزو کدام گروه زیر قرار می‌گیرند؟
الف) ورودی دیجیتال ب) خروجی دیجیتال ج) ورودی آنالوگ د) خروجی آنالوگ
۲۳- مهم‌ترین وظایف رله‌های فرمان در یک سیستم PLC کدام مورد است؟
الف) ایزوله‌سازی سیگنال ورودی به PLC ب) واسط سیگنال خروجی PLC و مصرف‌کننده
ج) موارد ۱و۲ د) هیچ‌کدام
۲۴- به‌منظور اطلاع اپراتور از روشن و خاموش بودن اجزاء مختلف یک سیستم صنعتی از . . . استفاده می‌شود.
الف) آژیر و بیزر ب) لامپ سیگنال ج) رله د) کنتاکتور
۲۵- در مورد سیگنال الکتریکی آنالوگ ورودی به PLC کدام مورد صحیح است؟
الف) از سنسورهایی که کمیت‌های فیزیکی پیوسته را اندازه‌گیری می‌کنند، دریافت می‌شود.
ب) به‌صورت سیگنال الکتریکی پیوسته می‌باشد.
ج) با تغییر در میزان کمیت مورد اندازه‌گیری توسط سنسور، این سیگنال نیز تغییر می‌کند.
د) هر سه مورد

فصل ۵

۲۶- در مورد ساختمان ترموکوپل کدام مورد صحیح است؟

الف) از دو فلز همجنس مجزا تشکیل شده است.

ب) از دو فلز غیرهمجنس مجزا تشکیل شده است.

ج) از دو فلز همجنس متصل به‌یکدیگر تشکیل شده است.

د) از دو فلز غیرهمجنس متصل به‌یکدیگر تشکیل شده است.

۲۷- در دماهای پایین دقت کدام سنسور بیشتر است؟

الف) ترمیستور ب) RTD ج) ترموکوپل د) PT100

۲۸- از مهمترین مشکلات کار با ترموکوپل به کدام مورد می‌توان اشاره نمود؟

الف) جبران‌سازی ب) نویزپذیری ج) تنظیمات خاص در PLC د) هر سه مورد

۲۹- منحنی پاسخ سنسور در برابر تغییرات دما در کدام نوع از سنسورهای زیر به‌صورت خطی است؟

الف) RTD ب) NTC ج) PTC د) ترموکوپل

۳۰- از سنسورهایی که برای اندازه‌گیری ارتفاع سطح استفاده می‌شوند، کدامیک دارای دقت بالاتری است؟

الف) آلتراسونیک ب) راداری ج) اختلاف فشار د) خازنی

۳۱- کدام تجهیز از جمله مصرف‌کننده‌های آنالوگ محسوب می‌شود؟

الف) موتور سه‌فاز ب) موتور تکفاز ج) درایو کنترل سرعت موتور د) سلونوئید ولو

فصل ۶

سیستم‌های کنترل زیمنس و جایگاه S7

۶-۱ مقدمه

۶-۲ سیستم‌های PLC

۶-۲-۱ PLCهای خانواده S5

۶-۲-۲ PLCهای خانواده S7

۶-۲-۳ PLCهای خانواده TI

۶-۲-۴ کنترل کننده‌های Quadlog

۶-۲-۵ مینی PLCهای LOGO

۶-۳ سیستم‌های PC based

۶-۴ سیستم‌های کنترل فرآیند Process Control

۶-۴-۱ Teleperm

۶-۴-۲ PCS7

۶-۴-۳ کنترل کننده‌های APACS

۶-۵ سیستم‌های کنترل خاص

۶-۵-۱ سیستم کنترل‌های TDC

۶-۵-۲ سیستم کنترل SIMADYN D

۶-۵-۳ سیستم‌های SINUMERIK

۶-۵-۴ سیستم‌های SIMOTION

۶-۶ پرسش و تحقیق

۶-۷ تست‌های خودآزمایی

در این فصل ضمن معرفی انواع مختلف سیستم‌های کنترل زیمنس به جایگاه خانواده S7 در بین آنها پرداخته شده است. به‌جز S7 سایر موارد جهت اطلاعات عمومی خواننده است ولی شناخت S7 پیش‌نیاز بحث‌های بعدی است و لازم است مطالعه شود.

چکیده مطالب

- کنترلرهای زیمنس را می‌توان به چهار دسته PLC و PC Based و Process Controller و SpecialController تقسیم کرد.
- در دسته PLC انواع S5 و S7 و C7 و TI و Quadlog و Logo جای می‌گیرند.
- TI و Quadlog در اصل از خود زیمنس نیستند و برند آنها توسط زیمنس خریداری شده است.
- S5 در حال حاضر توسط زیمنس تولید نمی‌شود.
- S7 به چهار دسته S7-200 و S7-1200 و S7-300 و S7-400 تقسیم می‌شود.
- S7-200 و S7-1200 برای کاربردهای کوچک مناسب هستند.
- S7-300 برای کاربردهای متوسط و S7-400 برای کاربردهای بزرگ طراحی شده است.
- S7-300F و S7-400F در کاربردهایی که ایمنی زیاد لازم است به‌کار می‌روند.
- S7-400H دارای دو CPU است که یکی اصلی و دیگری رزرو است و کاربرد آن برای فرآیندهایی است که نباید به‌دلیل فالت روی یک CPU فرآیند دچار توقف شود.
- S7-400FH سیستمی است با ویژگی S7-400H و قابلیت Fail Safe که به آن سیستم Fault Tolerant نیز می‌گویند.
- C7 با ترکیب S7-300 با پنل اپراتوری به‌صورت یکپارچه عرضه شده است.
- winAC و S7-mEC از خانواده کنترلرهای PC-Based هستند.
- ازسیستم‌های کنترل فرآیند که بیشتر نقش DCS را دارند می‌توان Teleperm و APACS و PCS7 را نام برد.
- برای کاربردهای سریع و دقیق کنترلرهای خاصی مانند SIMADYN عرضه شده است.

اصطلاحات و تعاریف

Module

ماژول کارت الکترونیکی است که در کنار سایر اجزای سیستم قرار می‌گیرد و برای هدف خاصی طراحی شده است. به‌عنوان مثال یک کارت الکترونیکی که قادر است اطلاعات سنسورها را به‌صورت صفر و یک دریافت کند یک ماژول خوانده می‌شود.

Modular

به سیستمی که اجزای آن از ترکیب ماژول‌های جداگانه ساخته شده‌اند ماژولار گفته می‌شود. این سیستم معمولاً انعطاف‌پذیر است و کاربر می‌تواند ماژول دلخواه را روی آن نصب کند، بعلاوه در صورت بروز اشکال روی یک ماژول می‌توان همان کارت را تعویض کرد و نیازی به تعویض کل سیستم نیست.

Compact

به سیستمی که اجزای آن به‌صورت یکپارچه طراحی شده است compact گفته می‌شود. این سیستم معمولاً انعطاف‌پذیری بسیار کمی دارد و توسعه آن با نصب ماژول اضافی به‌صورت محدود امکان‌پذیر است. در صورت بروز اشکال روی یک سیستم Compact کل سیستم بایستی تعویض شود.

Rack

رک به قسمتی از سیستم کنترل گفته می‌شود که اجزای سیستم روی آن نصب می‌گردد. به‌عنوان مثال به ریل آلومنیومی که در برخی سیستم‌های کنترل به‌کار می‌رود و کارت‌های الکترونیکی روی آن قرار می‌گیرند رک گفته می‌شود.

Fault

فالت که در فارسی به آن اشکال یا خطا نیز گفته می‌شود شرایطی است که سیستم در اجرای کار نرمال خود با مشکلی مواجه می‌شود.

Redundant

Redundant که در فارسی افزونه ترجمه شده است به سیستمی گفته می‌شود که برای اجرای عملیات مورد نظر دارای دو قسمت است، یکی سیستم اصلی و دیگری سیستم پشتیبان. کار کنترل توسط سیستم اصلی انجام می‌شود و در این شرایط سیستم پشتیبان آماده به‌کار است ولی در کار کنترل دخالت نمی‌کند. به محض بروز اشکال روی سیستم اصلی، سیستم پشتیبان وارد عمل شده و ادامه کار کنترل را به‌دست می‌گیرد. PLCهای افزونه دارای دو CPU برای این منظور هستند.

Fail Safe

سیستمی که در شرایط بروز فالت فرآیند را به سمت شرایط ایمن هدایت می‌کند Fail Safe گفته می‌شود. به‌عنوان مثال در سیستم کنترل Fail Safe می‌توان تعیین کرد که اگر کنترلر اصلی دچار مشکل شد، فرمان‌هایی که به Actuatorها ارسال شده در چه حالتی قرار گیرد.

Fault Tolerant

سیستمی است که هر دو قابلیت Redundancy و Fail Safe را دارد.

Availability

به معنای میزان در دسترس‌بودن است که در فارسی به دسترس‌پذیری نیز ترجمه شده است. این فاکتور بر اساس درصد بیان می‌شود و هر چقدر به 100% نزدیک باشد نشانگر دسترس‌پذیری بالاتر است. به‌عنوان مثال اگر سیستمی در طول یک هفته ۱۶۸ ساعته فقط ۱۴۰ ساعت در دسترس باشد و ۲۸ ساعت دیگر به‌دلیل نیاز به سرویس یا تعمیر قابل استفاده نباشد، میزان در دسترس بودن آن 140/168 است که معادل 83.3% است. در سیستم‌هایی که با عنوان High Availability یا دسترس‌پذیری بالا مطرح می‌شوند این عدد به‌صورت به ۱۰۰ درصد نزدیک است و به‌صورت تعدادی از ۹ بیان می‌شود مانند 99.999%. هر چه تعداد ۹ها بیشتر باشد، دسترس‌پذیری بیشتر است. در اتوماسیون صنعتی با استفاده از سیستم‌های کنترلی که دارای افزونگی هستند و سخت‌افزار پشتیبان دارند می‌توان دسترس‌پذیری بالا را ایجاد نمود.

Reliability

به معنای قابلیت اعتماد یا قابلیت اطمینان است. میزان اطمینان از اینکه یک سیستم یا یک تجهیز به‌طور رضایت‌بخشی بتواند وظایف خود را در شرایط طراحی شده‌اش انجام دهد. این میزان اطمینان بر اساس طول زمان عملکرد بدون مشکل وسیله نسبت به طول زمانی که عملکرد وسیله به‌دلیل فالت دچار مشکل می‌شود سنجیده می‌شود. به‌عنوان مثال وقتی یک سیستم الکترونیکی جایگزین یک سیستم رله کنتاکتوری می‌شود قابلیت اعتماد بالاتری را فراهم می‌سازد، زیرا در سیستم الکترونیکی اشکالات کمتری نسبت به یک سیستم الکترونیکی رخ می‌دهد.

در سیستم‌های Fail Safe که در صورت بروز اشکال نیاز به تضمین شرایط ایمن وجود دارد، قابلیت اعتماد[1] فاکتور مهمی است. برای این منظور باید از سخت‌افزارهای خاص استفاده کرد و چیدمان و آرایش آنها را به نحو مناسب انجام داد.

SIL

مخفف Safety Integrity Level است و منظور از آن درجه ایمنی فرآیند است که طبق استاندارد IEC61508 از SIL1 تا SIL4 تعریف می‌شود. هرچقدر این عدد بزرگتر باشد نیاز به سطح ایمنی بالاتر وجود دارد.

Drive

منظور از درایو وسیله‌ای است که برای تغییر دور و گشتاور موتورهای الکتریکی به‌کار می‌رود. برای موتورهای AC درایو به VFD مخفف Varriable Frequency Drive موسوم است.

Encoder

وسیله‌ای است که برای اندازه‌گیری موقعیت یا سرعت از آن استفاده می شود. اگر انکودر با یک محور گردان لینک شود با چرخش محور پالس تولید می‌کند هر چقدر سرعت بیشتر شود فرکانس پالس بالاتر می‌رود.

1. Reliability

ماشین CNC

computer numerical controlled ماشینی است که با کنترل عددی کار می‌کند منظور از کنترل عددی این است که موقعیت‌های حرکت یک ماشین می‌تواند به‌صورت عددی به سیستم کنترل داده شود تا ماشین را در آن قسمت قرار دهد. به‌عنوان مثال در یک ماشین تراش اتوماتیک می‌توان ابعاد x,y,z را برای قسمت متحرک تعریف کرد تا قطعه به‌صورت مناسب تراشکاری شود.

Motion Control

شاخه‌ای از اتوماسیون صنعتی است که در آن از تجهیزات مناسب برای کنترل موقعیت و سرعت ماشین‌آلات استفاده می‌شود. از کاربردهای آن می‌توان کارگاه‌های نورد، رباتیک و ماشین CNC را نام برد.

Unix

سیستم عاملی است که شبیه DOS و Windows می‌تواند روی کامپیوتر به‌کار گرفته شود. این سیستم عامل تحول زیادی را در رشد اینترنت ایجاد نموده است.

PCI Slot

PCI مخفف Peripheral Component Interconnect است و معرف باس کامپیوتر است که تجهیزات سخت‌افزاری توسط آن به پردازشگر ارتباط می‌یابند. قبلاً باس دیگری با عنوان ISA که استاندارد متفاوتی داشت استفاده می‌شد که امروزه کاربرد آن منسوخ شده است. ویژگی PCI این است که آدرس‌دهی به کارت به‌طور اتوماتیک انجام می‌شود.

TÜV

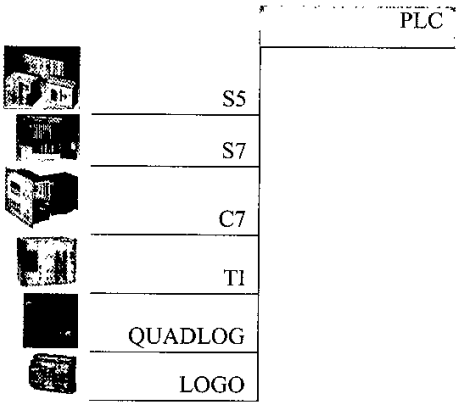
یک موسسه آلمانی است که نام آن برگرفته از سه کلمه آلمانی Technischer Überwachungs-Verein است و معادل انگلیسی آن Technical Inspection Association می‌باشد. این موسسه به‌طور خاص روی سیستم‌های ایمنی safety کار می‌کند. این موسسه با هدف حفاظت انسان در محیط‌های خطرناک، تجهیزات به‌کار رفته در این شرایط را بازررسی و تست می‌کند و در صورت رضایتبخش بودن به آنها گواهینامه می‌دهد.

۶-۱ مقدمه

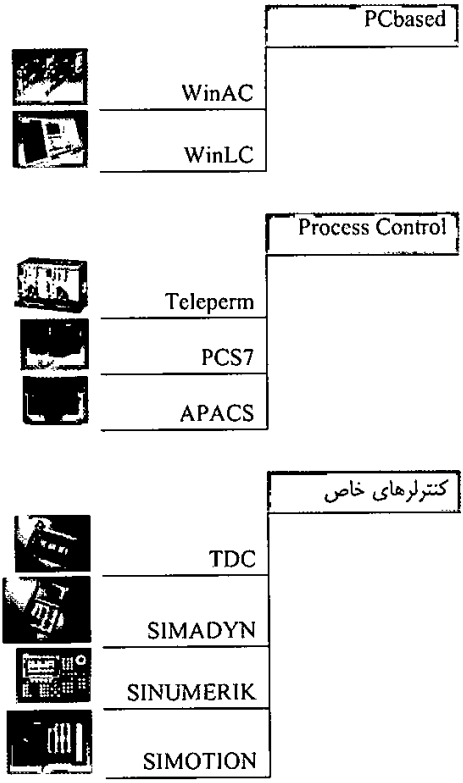
همانطور که در فصل اول اشاره شد، زیمنس یکی از اولین سازندگان سیستم‌های کنترل مدرن در دنیا محسوب می‌شود. این شرکت صرفاً به ساخت PLC اکتفا نکرده و سیستم‌های کنترل دیگری نیز جهت کنترل فرآیند یا کنترل ماشین‌های خاص عرضه کرده است. از آنجا که در فصل‌های بعدی مطالب به‌طور خاص در مورد PLCهای سری S7 زیمنس مطرح می‌گردند، لذا در این فصل با نگاهی اجمالی به انواع سیستم‌های کنترل زیمنس جایگاه S7 در بین آنها بررسی می‌شود.
سیستم‌های کنترل ارائه شده توسط زیمنس را می‌توان به چهار گروه اصلی تقسیم کرد:

- **سیستم‌های PLC:** از این گروه می‌توان به PLCهای S5و، S7و C7 و TI اشاره کرد.
- **سیستم‌های PC Based:** سیستم‌هایی هستند که بدون نیاز به PLC کار کنترل فرآیند را توسط کامپیوتر انجام می‌دهند. WinAC از این گروه به‌شمار می‌رود.
- **سیستم‌های کنترل پروسه:**[1] سیستم‌هایی هستند که بیشتر برای کنترل فرآیند و لوپ‌های کنترلی به‌صورت DCS به‌کار می‌روند. Teleperm، PCS7و APACS از این گروه هستند.
- **سیستم‌های کنترل خاص:** این سیستم‌ها برای کاربردهای خاص مانند کنترل عددی CNC و کنترل حرکت[2] و کنترل خاص درایوها استفاده می‌شوند. TDC و SIMADYN نمونه‌هایی برای کنترل درایو هستند.

همانطور که دیده می‌شود S7 فقط در گروه اصلی جای می‌گیرد . خواننده می‌تواند فقط مطالب مربوط به این گروه را مطالعه کند. سایر موارد صرفاً جهت افزایش اطلاع عمومی اشاره شده‌اند.
شکل ۶-۱ گروه‌های اصلی کنترلرهای مختلف زیمنس را در یک نگاه نشان می‌دهد.



1. Process Automation
2. Motion Control



شکل ۶-۱ سیستم‌های کنترل زیمنس در یک نگاه

سیستم‌های فوق در ادامه به اجمال معرفی شده‌اند.

۶-۲ سیستم‌های PLC

سیستم‌های PLC زیمنس متنوع هستند. به‌طور کلی می‌توان آنها را به‌صورت زیر طبقه‌بندی کرد:

- PLC های خانواده S5
- PLC های خانواده S7 که C7 را نیز دربر می‌گیرد.
- PLC های خانواده T1
- QUADLOG ،Safety PLCهای
- مینی PLCهای LOGO

۶-۲-۱ PLCهای خانواده S5

خانواده S5 از قدیمی‌ترین PLCهای زیمنس هستند که در حال حاضر تولید آنها متوقف شده است. این PLC یکی از اولین PLCهایی است که به‌طور وسیع در کشور ما مورد استفاده قرار گرفته است. از دهه ۱۳۶۰ به بعد صنایعی را در کشور می‌بینیم که سیستم کنترل آنها بر مبنای این PLC طراحی و پیاده‌سازی شده است.

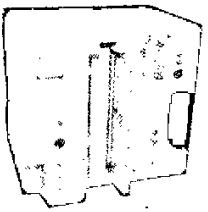
امروزه به‌دلیل مشکل تامین قطعات یدکی این PLC و برخی از کمبودها و نواقص آن، در برخی صنایع PLC S5 تدریجاً کنار گذاشته شده و با انواع جدید در حال جایگزینی است.

خانواده S5 اعضای مختلفی دارد که ویژگی‌های آنها با یکدیگر متفاوت است. کنترل‌کننده‌های S5 را می‌توان بصورت زیر تقسیم‌بندی نمود:

S5-90U/S5-95U

این PLCها به‌صورت یکپارچه می‌باشند و برای کاربردهای کوچک طراحی شده‌اند. برنامه‌نویسی آنها با Step5 انجام می‌شود. در شکل ۶-۲ دو نمونه از این PLCها را مشاهده می‌نمایید.

S5-90U



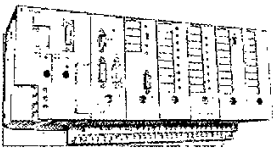
S5-95U

شکل ۶-۲ نمونه PLCهای S5-90U/S5-95U

S5-100U و S5115U

این PLCها به‌صورت مدولار و برای کاربردهای کوچک تا متوسط طراحی شده‌اند. برنامه‌نویسی آنها با Step5 انجام می‌شود.

S5-115U



S5-100U

شکل ۶-۳ نمونه PLCهای S5-100U/S5-115U

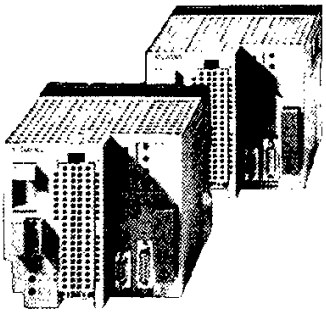
S5-155U و S5-135U

این PLCها برای کاربردهای بزرگ و وسیع که تعداد I/O زیاد دارند و نیاز به سرعت و امکانات بالا می‌باشد طراحی شده‌اند. برنامه‌نویسی آنها با Step5 انجام می‌شود.

شکل ۶-۴ نمونه PLC مدل S5-135U

S5-95F / S5-115F

این PLCها برای کاربردهایی که نیاز به ایمنی زیاد می‌باشد طراحی شده‌اند و اصطلاحاً به آن سیستم Fail Safe گفته می‌شود. حرف F در انتهای کد PLC معرف این مطلب است. برای برنامه‌نویسی و تنظیمات آنها علاوه بر Step5 نرم‌افزار COM95F/ COM115Fنیز مورد نیاز است.



S5-95F

شکل ۶-۵ نمونه PLC مدل Fail Safe از خانواده S5

S5-115H /S5-155H

این نوع PLCها در مواردی که نیاز به Avaliability بالا باشد مورد استفاده قرار می‌گیرند. منظور از دسترس‌پذیری بالا این است که در صورت بروز اشکال روی یکی از اجزای سیستم از جمله روی یک CPU کل فرآیند دچار مشکل نشود. برای همین منظور از دو CPU که روی دو رک مستقل قرار گرفته‌اند و یکی از آنها رزرو دیگری است، استفاده می‌شود.

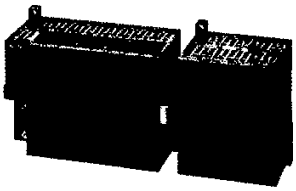
برای برنامه‌نویسی و تنظیمات آنها علاوه بر نرم‌افزار Step5 نرم‌افزار COM115H/COM155H نیز مورد نیاز است.

۶-۲-۲ PLCهای خانواده S7

PLC های این خانواده از پرکاربردترین PLCهای مورد استفاده در صنایع داخلی کشور می‌باشند، سیستم اتوماسیون صنعتی در بسیاری از پروژه‌های جدید در زمینه‌های نفت و گاز، پتروشیمی، فولاد، مس، آلومنیوم، خودروسازی و .... بر مبنای PLCهای این خانواده طراحی و پیاده‌سازی شده است. خانواده S7 به گروه‌های زیر تقسیم می‌گردد:

- S7-200
- S7-300
- S7-400
- S7-1200
- C7
- S7-200

این PLC به‌صورت یکپارچه (Compact) است ولی می‌توان با استفاده از ماژول‌های توسعه (Expansion) ورودی و خروجی‌های آن را تا حدی افزایش داد. از نظر کاربری در پروژه‌های کوچک که تعداد I/O کم است و توانایی زیادی از PLC مورد انتظار نیست به‌کار می‌روند. برنامه‌نویسی S7-200 توسط نرم‌افزار Step7 Microwin انجام می‌گیرد.

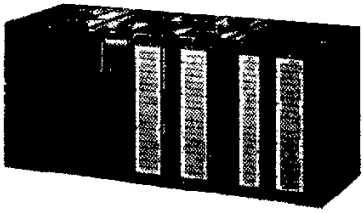


شکل ۶-۶ نمونه PLC از خانواده S7-200

S7-300

این PLC ماهیت مدولار دارد و کارت‌های مختلف I/O در کنار آن قابل نصب است. اگرچه برخی از مدل‌های آن به‌صورت یکپارچه همراه با I/O عرضه شده‌اند ولی به‌سهولت می‌توان با اضافه‌کردن کارت‌های دلخواه، تعداد ورودی و خروجی آنها را نیز تا حدی افزایش داد. از نظر کاربری، PLCهای این خانواده برای پروژه‌های در حد متوسط که I/O قابل توجه (ولی نه بسیار زیاد) دارند مورد استفاده قرار می‌گیرند.

در این PLCها، برنامه‌نویسی و پیکربندی و تنظیمات مربوط به شبکه توسط نرم‌افزار Step7 انجام می‌شود.

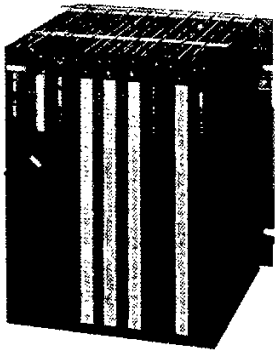


شکل ۶-۷ نمونه PLC از خانواده S7-300

S7-400

این PLC ها نیز به‌صورت مدولار بوده و برای پروژه‌هایی که تعداد I/O در آنها بسیار زیاد است یا توانایی‌های ویژه‌ای نظیر سرعت پردازش بالا مورد نیاز است به‌کار می‌روند.

در این PLCها نیز برنامه‌نویسی و پیکربندی و تنظیمات مربوط به شبکه توسط نرم‌افزار Step7 انجام می‌شود.



شکل ۶-۸ نمونه PLC از خانواده S7-400

S7-300F / S7-400F

این PLC ها همانطور که از نامشان پیداست، برای سیستم‌های Fail Safe به‌کار می‌روند؛ یعنی در مواردی که فرآیند خطرناک و در صورت بروز خطا امکان انفجار یا آتش‌سوزی وجود داشته باشد استفاده می‌شوند. برای پیکربندی و برنامه‌نویسی این سیستم علاوه بر Step7 نرم‌افزار F-system نیز مورد نیاز است.

شکل ۶-۹ نمونه PLC از خانواده S7-300Fail Safe

S7-400H

پایه آن همان S7-400 است و در جایی که High Availability مورد نیاز است به‌کار می‌رود، مانند:

۱. پروسه‌ای که اگر متوقف شود منجر به خسارت زیاد می‌شود مثلاً محصول گرانقیمتی از بین می‌رود.

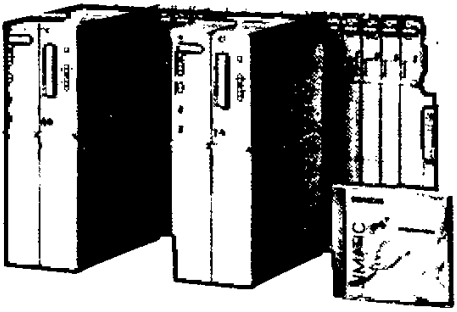
۲. جایی که هزینه راه‌اندازی مجدد سیستم پس از رفع عیب بالاست.

۳. در مواردی که راه‌اندازی مجدد CPU و انجام تنظیمات آن زمان‌بر است، مثلاً در جایی که PLC لوپ‌های PID را کنترل می‌نماید.

به این سیستم Redundant نیز گفته می‌شود و دارای دو CPU مشابه است که یکی به‌عنوان Master و دیگری به‌عنوان Standby کار می‌کنند. در صورتی‌که Master دچار مشکل شود Standby وارد عمل شده و کنترل فرآیند را به‌عهده می‌گیرد.

در طول مدت تغییر وضعیت از Master به Standby که در زمان بسیار کوتاهی اتفاق می‌افتد خروجی‌ها ثابت می‌مانند تا مشکلی در فرآیند پیش نیاید.

برای برنامه‌ریزی و پیکربندی این سیستم علاوه بر Step7 باید پکیج H-System نیز نصب گردد یا از نسخه Step7 Proffesional استفاده شود.

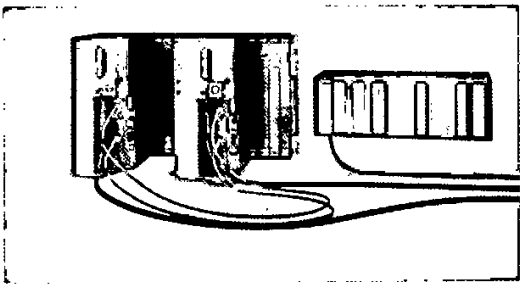


شکل ۶-۱۰ نمونه PLC از خانواده S7-400H

S7-400FH

این سیستم خواص سیستم‌های H و F را دارست و جایی که Availability بالا[1] و Reliability بالا نیاز است به‌کار می‌رود. طبق استانداردهای ایمنی برای پروسه‌های خطرناک، درجه ایمنی موسوم به SIL تعریف می‌شود و بر اساس آن انتخاب سیستم صورت می‌گیرد. از نمونه کاربردهای سیستم FH می‌توان کاربرد در SIL3 را مثال زد.

برای پیکربندی و برنامه‌نویسی این سیستم علاوه بر Step7 و H-system نرم‌افزار F-system نیز مورد نیاز است.

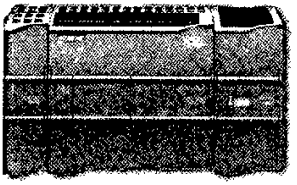


شکل ۶-۱۱ نمونه PLC از خانواده S7-400FH

۱. دسترسی‌پذیری بالا

S7-1200

میکرو PLC جدیدی است که اگرچه ورودی و خروجی محدودی را ساپورت می‌کند ولی مجهز به امکانات پیشرفته‌تری نسبت به سایر اعضای خانواده S7 است که از جمله آنها می‌توان مجهز بودن CPU به پورت شبکه اترنت را نام برد. در شکل ۶-۱۲ یک نمونه از این PLCها نشان داده شده است.



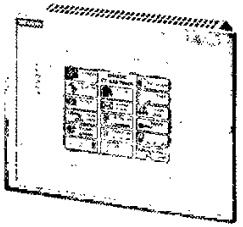
شکل ۶-۱۲ نمونه PLC از خانواده S7-1200

این PLC در دو مدل یکپارچه و مدولار عرضه شده و برنامه‌نویسی آن با Step7 V10 انجام می‌شود که با Step7 به‌کار رفته برای S7-300 و S7-400 متفاوت است.

PLCهای سری C7

این PLC ها دراصل از خانواده S7-300 هستند ولی به‌صورت یکپارچه و همراه با صفحه نمایش اپراتوری عرضه شده‌اند. به‌عبارت دیگر با ترکیب PLC و OP، سیستم کنترل و مانیتورینگ به صورت یکجا ارائه می‌گردد که اقتصادی‌تر از خرید مجزای این دو سیستم است. از جمله موارد کاربری C7 می‌توان به نصب روی درب پانل موجود در اتاق‌های کنترل اشاره نمود، که در این موارد C7 علاوه بر انجام کار کنترل عمل مانیتورینگ را نیز انجام می‌دهد.

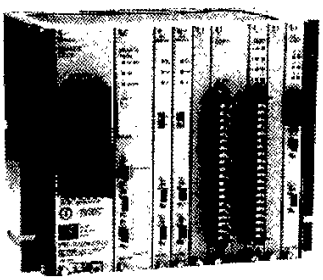
برنامه‌نویسی کنترل در آنها توسط نرم‌افزار Step7 مشابه سیستم‌های S7-300 انجام می‌گیرد. صفحات نمایش با نرم‌افزار Protool یا در نوع جدیدتر توسط نرم‌افزار winCC Flexible انجام می‌شود.



شکل ۶-۱۳ نمونه PLC از خانواده C7

۶-۲-۳ PLCهای خانواده TI

این PLCها ساخت شرکت Texas Insrtument هستند. زیمنس این شرکت را در سال ۱۹۹۱خریداری کرده و کنترلرهای آن با نام زیمنس عرضه می‌شوند. این PLCها خانواده‌ای جدا محسوب می‌گردند و اجزای آنها به‌صورت مدولار است. برای برنامه‌نویسی آنها از نرم‌افزار TISOFT استفاده می‌شود.



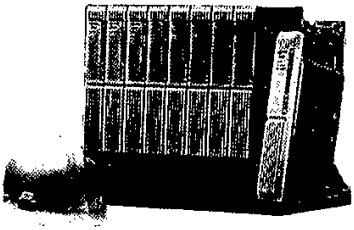
شکل ۶-۱۴ نمونه PLC از خانواده Simatic 505

۶-۲-۴ کنترل کننده‌های Quadlog

QUADLOG نوعی PLC است که سازنده اصلی آن شرکت Moore می‌باشد ولی امروزه محصولات آن با عنوان Siemens-Moore عرضه می‌گردد. در حال حاضر این نوع PLC تولید نمی شود و زیمنس آنرا با مدل‌های S7-400 جایگزین کرده است.

Quadlog برای سیستم‌هایی که نیاز به ایمنی بالا دارند طراحی شده است، از اینرو در دسته Safety PLCها قرار می‌گیرد.

سیستم‌های ESD قطع اضطراری، سیستم‌های کنترل بویلر، سیستم‌های اطفاء حریق از جمله کاربردهایی هستند که نیاز به Safety PLC دارند. در این سیستم‌ها بایستی اجزای سیستم کنترل به‌گونه‌ای کار کنند که در صورت بروز اشکال روی یکی از آنها فرآیند به سمت شرایط ایمن هدایت گردد، به همین علت به آنها سیستم‌های Fail Safe (ایمن در برابر خطا) نیز گفته می‌شود.



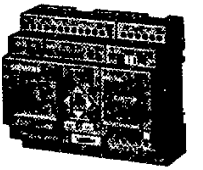
شکل ۶-۱۵ کنترلر QUADLOG

Quadlog از نظر درجه ایمنی می‌تواند تا SIL3 را طبق استاندارد IEC 61508 پشتیبانی کند و دارای تاییدیه‌های مختلف از استاندارد IEC و TUV و موسسات معتبر دیگر است است. تاییدیه این سیستم آنرا در حد سیستم افزونه سه‌گانه و چهارگانه جای می‌دهد که در همان حد قابل اطمینان Reliable است ولی پیچیدگی آنها را ندارد. نرم‌افزار مهندسی که

برای پیکربندی و برنامه‌نویسی این سیستم به‌کار می‌رود 4mation نام دارد که طبق استاندارد IEC61131 پنج روش برنامه‌نویسی را ساپورت می‌کند.

۶-۲-۵ مینی PLCهای LOGO

مینی PLC کوچک و ارزان قیمتی است که برای کارهای کنترلی کوچک مانند اتوماسیون ساختمان، آسانسورها، سیستم‌های تهویه مطبوع و . . . عرضه شده است. این PLC به‌صورت است و برنامه‌ریزی آن توسط کلیدهای روی آن انجام می‌شود. همچنین می‌توان توسط کامپیوتر نیز آنرا برنامه‌ریزی نمود. برای برنامه‌ریزی از طریق کامپیوتر باید نرم‌افزار LOGO! Soft Comfort نصب گردد.



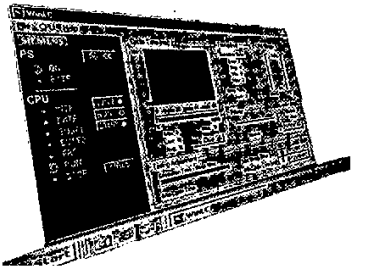
شکل ۶-۱۶ نمونه PLC از خانواده LOGO

۶-۳ سیستم‌های PC based

در سیستم PC Based بدون وجود PLC، کامپیوتر نقش PLC را به‌عهده گرفته است. این کنترلرها که عنوان winAC به آنها اطلاق شده است به دو گروه تقسیم می‌شوند:

Software PLC

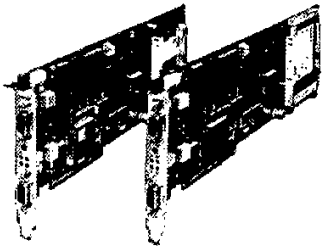
در این نوع، کامپیوتر به‌صورت نرم‌افزاری نقش PLC را ایفا می‌کند. برای این منظور نرم‌افزار WinLC مخفف Windows Logic Controller همراه با برخی نرم‌افزارهای دیگر مانند Simatic NET OPC نصب و استفاده می‌گردد. WinLC به‌صورتی طراحی شده که مشابه PLC به‌صورت سیکلی برنامه کنترل را اجرا می‌نماید. برای اتصال کامپیوتر به I/Oها از کارت CP5611 یا CP5613 که به شبکه پروفی‌باس متصل می‌گردند استفاده می‌شود.



شکل ۶-۱۷ WinLC از خانواده PC Based

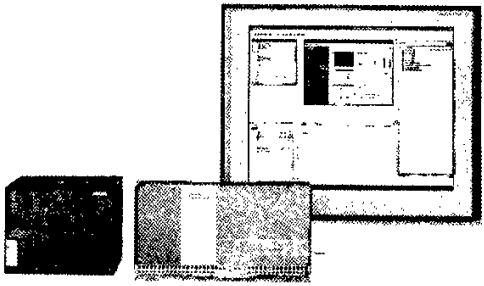
Slot PLC

در این نوع، کارت CPU که می‌تواند از نوع CPU 416-2 یا CPU 416 باشد در اسلات PCI کامپیوتر نصب می‌شود. این کارت از طریق شبکه پروفی‌باس با I/Oها ارتباط برقرار می‌سازد.



شکل ۶-۱۸ CPU کنترلر برای خانواده PC Based

تذکر: سیستم کنترل جدیدی توسط زیمنس با عنوان S7-mEC عرضه شده است. کلمه mEC مخفف modular Embeded Controller می باشد. در این سیستم کنترلر، HMI و امکانات PC به‌صورت یکجا جمع شده است.



شکل ۶-۱۹ نمونه سیستم S7-mEC

این سیستم مشابه WinAC است ولی علاوه بر آن PLC نیز به‌صورت سخت افزاری استفاده شده است و پنل اپراتوری قابل لمس[1] روی آن تعبیه شده است. همه‌ی اجزای این سیستم روی پلتفرم سیستم عامل ویندوز به‌راحتی با یکدیگر لینک شده‌اند. سیستم HMI می‌تواند با سایر نرم‌افزارها لینک شود. بعلاوه کنترلر نیز با نرم‌افزارهای تحت ویندوز به‌سهولت ارتباط برقرار می‌کند، به‌عنوان مثال می‌توان دیتاهای PLC را به محیط Office منتقل نمود. کنترلر این سیستم از نوع S7-300 است و ورودی و خروجی‌های آن مانند سیستم 300 قابل افزایش می‌باشد. برنامه‌نویسی با Step7 و طراحی مانیتورینگ با winCC Flexible انجام می‌شود.

1. Touch Punnel

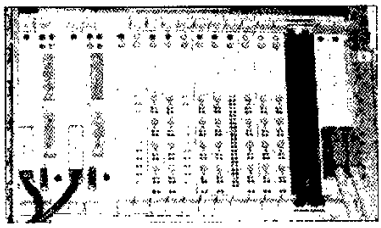
۶-۴ سیستم‌های کنترل فرآیند[1]

۶-۴-۱ Teleperm

Telepermمحصولی از بخش PG زیمنس (Power Generation) است که در زمینه تولید برق و تجهیزات نیروگاهی فعالیت می‌کند. انواع قدیمی Teleperm مدل‌های A و B و C و M است.

Teleperm XP که به اختصار TXP خوانده می‌شود محصول جدید این بخش از زیمنس است. TXP مبتنی بر سیستم عامل UNIX است که جایگزین مدل قبلی آن یعنی Teleperm M شده است.

این سیستم DCS به‌طور عمده برای اتوماسیون نیروگاهی طراحی و به‌کار گرفته شده است. برای برنامه‌نویسی این سیستم از نرم‌افزار FUP Editor استفاده می‌شود.

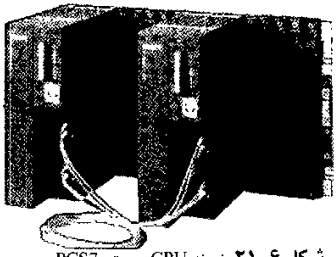


شکل ۶-۲۰ Teleperm XP

۶-۴-۲ PCS7

PCS7 که نام آن برگرفته از Process Control System است محصولی از شاخه Automation & Drive شرکت زیمنس یعنی Simatic می‌باشد که به‌عنوان یک سیستم DCS مطرح است. این سیستم در سال ۱۹۹۶ عرضه شده و سیستم‌های DCS قدیمی زیمنس که با نام Teleperm M شناخته می‌شدند توانایی ارتقا به آنرا دارا می‌باشند. کنترلرهای به‌کار رفته در PCS7 از انواع پیشرفته S7-400 هستند.

اجزای سیستم PCS7 به‌گونه‌ای است که می‌تواند نیازهای مختلفی که در سطوح مختلف اتوماسیون صنعتی وجود دارد را پوشش دهد. منظور از سطوح مختلف اتوماسیون، سطح فیلد، سطح کنترل، سطح اپراتوری و سطح مدیریتی می‌باشد. این یکپارچه‌بودن اجزای سخت‌افزاری اصطلاحاً TIA خوانده می‌شود که مخفف Totally Integrated Automation است.



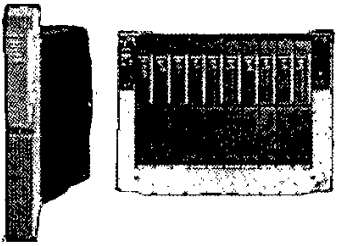
شکل ۶-۲۱ نمونه CPU سیستم PCS7

1. Process Controls

۶-۴-۳ کنترل کننده‌های APACS

+APACS سیستم کنترلری است که می‌تواند به‌عنوان PLC یا DCS به‌کار رود و با انتخاب سخت‌افزار مناسب در عین حال می‌تواند به‌طور همزمان هر دو نقش را ایفا کند.

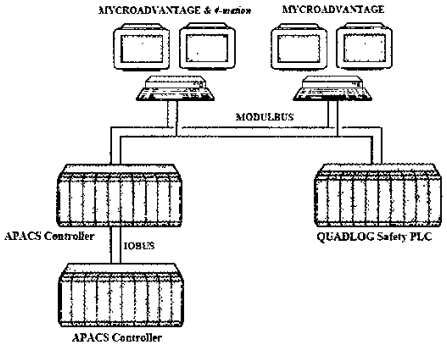
این سیستم همراه با QUADLOG توسط شرکت Moore ساخته شده است. این دو سیستم بخوبی با هم لینک می‌شوند و نرم‌افزار مهندسی آنها مشابه است. محصولات Moore با عنوان Siemens-moore عرضه می‌شد که شرکت زیمنس در حال جایگزینی آنها با S7-400 است.



شکل ۶-۲۲ کنترلر APACS

این کنترلر به‌صورت کاملاً ماژولار بوده و ماژول‌های کنترل، ماژول‌های I/O و ماژول‌های ارتباطی شبکه اصلی‌ترین اجزای آن هستند.

APACS به‌سهولت با QUADLOG لینک شده و می‌تواند برای کار کنترل یکپارچه به‌کار رود. در شکل ۶-۲۳، Quadlog برای Burner Management System و APACS برای Combustion Control استفاده شده است.



شکل ۶-۲۳ نمونه کاربرد APACS همراه با QUADLOG

برای پیکربندی و برنامه‌نویسی APACS از نرم‌افزار 4-mation استفاده می‌گردد.

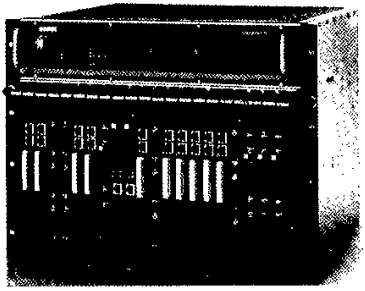
۶-۵ سیستم‌های کنترل خاص

۶-۵-۱ سیستم کنترل‌های TDC

این کنترلرها به‌صورت تخصصی در اتوماسیون مبتنی بر درایو که در برخی صنایع مانند کارخانجات فولاد کاربرد زیاد دارد طراحی شده است. TDC مخفف Technology and Drives Control است.

این سیستم مجهز به فانکشن (برنامه) محاسباتی از پیش نوشته‌ای است که منطق‌های پیچیده محاسباتی برای کاربرد مورد نظر در آنها به‌کار رفته است و کاربر را از طراحی و نوشتن این برنامه‌ها بی‌نیاز می‌کند.

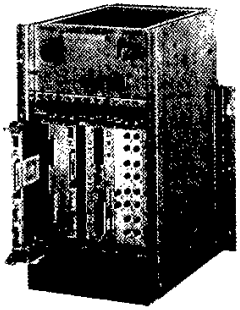
سخت‌افزار این سیستم نیز به‌طور خاص طراحی شده است. CPU قدرتمند و کارت‌های ورودی و خروجی که علاوه بر ورودی و خروجی‌های معمولی می‌توانند سیگنال‌های انکودر را نیز دریافت کنند از ویژگی‌های سخت‌افزار این سیستم است. روی این سیستم می‌توان چندین CPU نصب کرد که به‌راحتی با یکدیگر تبادل دیتا کنند. همه این CPUها می‌توانند از یک حافظه مشترک سراسری استفاده کنند که به Global Data Memory یا به اختصار GDM موسوم است.



شکل ۶-۲۴ نمونه سیستم کنترل TDC

۶-۵-۲ سیستم کنترل SIMADYN D

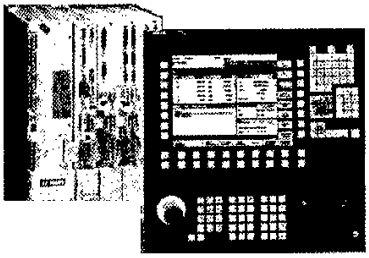
این سیستم به‌طور خاص برای کاربردهایی که نیاز به پاسخ بسیار سریع یا دقت محاسباتی بالا دارند (مانند کنترل درایوها) با ساختاری مدولار طراحی شده است که می‌تواند با تکنولوژی multi-processor کار کند. از جمله کاربردهای آن می‌توان به کنترل درایوهای کارخانجات نورد که نیاز به دقت و سرعت پاسخ زیاد دارند را نام برد.



شکل ۶-۲۵ کنترلر SIMADYN D

۶-۵-۳ سیستم‌های SINUMERIK

کنترل‌کننده عددی Numeric است که در ماشین‌های CNC استفاده می‌شود. این کنترل‌کننده خاص، دارای مدل‌های مختلف مانند 802 و 810 و 840 است که ویژگی‌ها و کاربرد متفاوت دارند.



شکل ۶-۲۶ کنترلر Sinumerik 840

۶-۵-۴ سیستم‌های SIMOTION

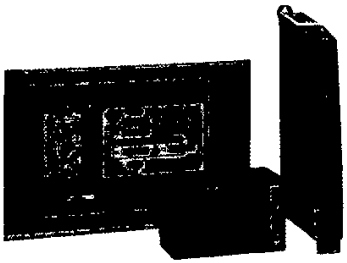
پس از سیستم‌های Sinumerik سیستم کنترل SIMOTION برای کنترل حرکت‌های پیچیده طراحی شده است؛ از اینرو آنرا می‌توان برای کنترل ماشین‌ابزار[1] و هم برای ماشین‌های تولید به‌کار برد. SIMOTION به‌راحتی با درایوهای کنترل سرعت لینک شده و دارای فانکشن‌های مختلف برای کنترل موقعیت است. نمونه کاربرد این کنترل‌کننده در ماشین‌های بسته‌بندی، ماشین‌های چاپ، پرس‌ها، ماشین‌های به‌کار رفته در صنایع چوب، سنگ و شیشه می‌باشد.

این سیستم به چند مدل تقسیم می‌شود:

SIMOTION C: که بر پایه S7-300 طراحی شده است.

SIMOTION P: که به‌صورت PC Based طراحی شده است.

SIMOTION D: که به‌صورت یکپارچه و برای نصب روی درایوهای SINAMICS استفاده می‌شود.



شکل ۶-۲۷ نمونه کنترلرهای SIMOTION

1. Machine Tools

۶-۶ پرسش و تحقیق

- گروه‌بندی سیستم‌های کنترل ساخت یکی از سازندگان دیگر را بررسی کنید.
- عدد انتهای کد SIL دقیقاً معرف چه شرایطی است؟

۶-۷ تست‌های خودآزمایی

۱- کد F در انتهای نام برخی از CPUها بیانگر کدام خصوصیت CPU می‌باشد؟

الف) افزونه ب) ایمن در برابرخطا ج) یکپارچه د) هیچ‌کدام

۲- در این سیستم از دو CPU که یکی از آنها رزرو دیگری است استفاده می‌شود.

الف) ایمن در برابر خطا ب) افزونه ج) یکپارچه د) هیچکدام

۳- کدام‌یک از PLCهای زیر دارای پورت اتصال به شبکه‌ی اترنت می‌باشد؟

الف) S7-300 ب) S7-400 ج) S7-1200 د) LOGO

۴- کدام‌یک از سیستم‌های زیر قابلیت انجام کنترل و مانیتورینگ را همزمان دارا می‌باشد؟

الف) S7-300 ب) S7-400 ج) C7 د) LOGO

۵- کدام کنترلر دارای حوزه‌ی عملکرد وسیع بوده و در سیستم PCS7 نیز استفاده می‌شود؟

الف) S7-300 ب)S7-1200 ج) C7 د) S7-400

۶- این PLCها برای کاربردهایی که نیاز به ایمنی زیادی دارند ساخته شده است.

الف) S7-300 ب) Quadlog ج) C7 د) S7-1200

۷- کدام کنترلر به‌صورت خاص برای ماشین‌های CNC طراحی شده است؟

الف) Sinumerik ب) Simadyn ج) Simotion د) TDC

فصل ۷

نرم‌افزارهای زیمنس و جایگاه Step7

۷-۱ مقدمه

۷-۲ نرم‌افزارهای اصلی Programming

۷-۲-۱ STEP 7 Basic

۷-۲-۲ Step7

۷-۲-۳ Step7 Proffessional

۷-۲-۴ Step7 Lite

۷-۲-۵ Step7 Microwin

۷-۲-۶ STEP5

۷-۲-۷ TISOFT

۷-۲-۸ LOGO Soft Comfort

۷-۲-۹ 4mation

۷-۳ نرم افزارهای تکمیلی Programming

۷-۳-۱ Graph 5/II

۷-۳-۲ S7-Graph

۷-۳-۳ S7-HiGRAPH

۷-۳-۴ S7-SCL

۷-۳-۵ CFC

۷-۳-۶ SFC

۷-۳-۷ SWR

۷-۳-۸ H-System

۷-۳-۹ F-System

۷-۳-۱۰ Safety Matrix

۷-۳-۱۱ M7 ProC/C++

۷-۳-۱۲ PLCSIM

۷-۳-۱۳ Standard PID Control

۷-۳-۱۴ Modular PID Control

۷-۴ نرم‌افزارهای عیب‌یابی و سرویس

۷-۴-۱ PDIAG

۷-۴-۲ TeleService

۷-۴-۳ PRODAVE

۷-۵ نرم‌افزارهای مانیتورینگ

۷-۵-۱ winCC

۷-۵-۲ WinCC Flexible

۷-۵-۳ winAC

۷-۵-۴ Protool

۷-۵-۵ PC Access

۷-۶ نرم‌افزارهای مربوط به شبکه‌های صنعتی

۷-۶-۱ Simatic NET

۷-۶-۲ iMAP

۷-۶-۳ PDM

۷-۶-۴ نرم‌افزارهای ارتباطی S5

۷-۷ نرم‌افزارهای مربوط به Motion Control

۷-۷-۱ S7-Technology

۷-۷-۲ D7-Sys

۷-۷-۳ Drive ES

۷-۸ برخی دیگر از نرم‌افزارهای مرتبط با اتوماسیون

۷-۸-۱ DOCPRO

۷-۸-۲ HARDPRO

۷-۸-۳ FuzzyControl++

۷-۸-۴ Neurosystem

۷-۸-۵ Premium Studio

۷-۸-۶ PCS7

۷-۹ پرسش و تحقیق

۷-۱۰ تست‌های خودآزمایی

در این فصل با نگاه کلی به نرم‌افزارهای مختلف زیمنس، جایگاه Step7 و اجزای آن بیان شده است.

فصل ۷

چکیده مطالب

عنوان	کاربرد
STEP7 Basic	برای S7-1200
STEP7 Lite	برای S7-300 و C7
STEP7	برای S7-400 و S7-300 و C7
STEP7 Proffessional	شامل Step7 و برخی نرم افزارهای Optional
STEP7 Micro/win	برای S7-200
STEP5	برای S5
TISOFT	برای Ti 505
LogoSoft	برای LOGO
4 mation	برای Quadlog و APACS
Graph 5/II	کنترل ترتیبی برای S5
S7-GRAPH	کنترل ترتیبی برای S7
S7-HIGRAPH	کنترل ترتیبی برای S7با دیاگرام حالت
S7-SCL	برنامه‌نویسی سطح بالا مشابه پاسکال برای S7 و PCS7
CFC	برنامه‌نویسی گرافیکی به‌صورت چارت برای S7 و PCS7
SFC	کنترل ترتیبی به‌صورت گرافیکی برای S7 و PCS7
F-SYSTEM	برای سیستم‌های Fail Safe
H-SYSTEM	برای افزونگی سخت‌افزاری در S7
SWR	برای افزونگی نرم‌افزاری در S7
SAFETY MATRIX	ابزار سطح بالا برای طراحی منطق کنترل Fail Safe
M7 ProC/C++	برای M7
PLCSIM	سیمولاتور S7
Standard PID Control	طراحی PID در S7
Modular PID Control	طراحی PIDهای پیچیده در S7
PDIAG	برای نمایش فالت‌های فرآیند در HMI
TeleService	امکان سرویس و دسترسی به PLC از راه دور
PRODAVE	نمایش اطلاعات PLC در HMI
winCC	طراحی مانیتورینگ HMI
WinCC Flexible	طراحی مانیتورینگ پنل‌های اپراتوری جدید
winAC	سیستم PC Based Control
Protool	طراحی مانیتورینگ، پنل‌های اپراتوری
PC Access	مانیتورینگ S7-200 از طریق OPC
Simatic NET	OPC برای S7
iMAP	پیکربندی و عیب‌یابی شبکه Profinet
PDM	پیکربندی و عیب‌یابی شبکه Profibus-PA و HART
Com –S5	نرم‌افزار ارتباطی S5
S7-Technology	خاص سیستم‌های CPU31x-T
D7-Sys	برای TDC و SIMADYN
Drive ES	پیکر بندی درایو
DOCPRO	مستندسازی پروژه Step7 و PCS7
HARDPRO	ابزار کمکی پیکربندی سخت افزار S7-300
FuzzyControl++	پیاده‌سازی منطق فازی با S7
Neurosystem	پیاده‌سازی شبکه عصبی با S7
Premium Studio	مجموعه نرم‌افزارهای کنترل و مانیتورینگ
PCS7	برای سیستم DCS جدید زیمنس (PCS7)

اصطلاحات و تعاریف

Runtime

منظور از محیط Runtime نرم‌افزار مانیتورینگ حالتی است که نرم افزار فعال بوده و پارامترهای فرآیند در حال نمایش هستند و اپراتور از طریق این محیط می‌تواند فرامین لازم را صادر نماید.

Engineering

منظور Engineering انجام کارهای پیکربندی و طراحی برنامه‌های کنترل و صفحات مانیتورینگ توسط نرم‌افزار است. عیب‌یابی نیز از امور Engineering محسوب می‌شود.

Fault

Fault که به فارسی خطا ترجمه می‌شود به مفهوم هر گونه شرایط غیر نرمال و غیر عادی در عملکرد یک وسیله یا یک سیستم است.

Diagnostic

Diagnostics به مفهوم تشخیص عیب است. در شرایط بروز Fault ابزارهای شناسایی و تشخیص خطا به عنوان Diagnostics محسوب می‌شوند.

Motion Control

به مفاهیم و اصطلاحات فصل ۶ رجوع شود.

فصل ۷

۷-۱ مقدمه

در این فصل به معرفی مهم‌ترین نرم‌افزارهای زیمنس که مربوط به سیستم کنترل و اتوماسیون هستند می‌پردازیم. این نرم‌افزارها را می‌توان به‌صورت زیر تقسیم بندی نمود:

- نرم‌افزارهای اصلی Programming [۱]سیستم‌های کنترل
- نرم‌افزارهای تکمیلی برای Programming سیستم‌های کنترل (ابزارهای اختیاری)
- نرم‌افزارهای عیب‌یابی و سرویس
- نرم‌افزارهای مانیتورینگ
- نرم‌افزارهای Simotion Control
- سایر موارد

نرم‌افزار Step7 که مخصوص PLCهای S7-300 و S7-400 است در گروه اول جای می‌گیرد. از اینرو خواننده محترم می‌تواند صرفاً مطالب این قسمت را مطالعه کرده و از بررسی سایر موارد که جهت افزایش اطلاعات عمومی است صرفنظر کند.

۷-۲ نرم‌افزارهای اصلی Programming

این گروه از نرم‌افزارها برای برنامه‌نویسی و تنظیمات سیستم‌های کنترل استفاده می‌شوند. هر کدام از آنها یک یا چند سیستم را ممکن است پوشش دهد. این نرم‌افزارها عبارتند از :

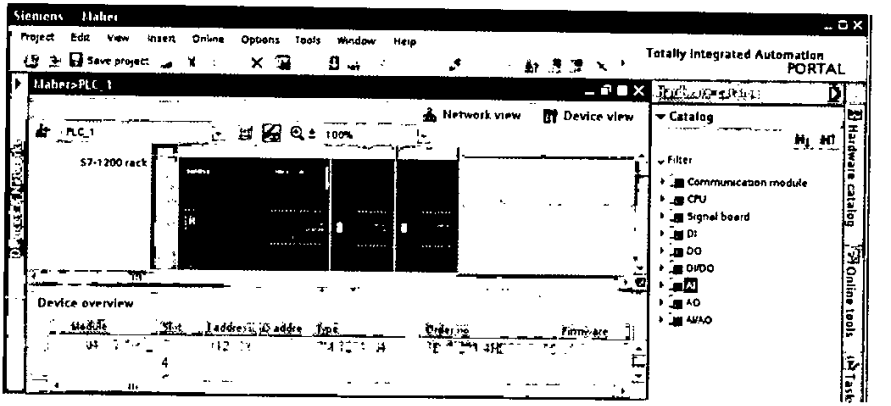
- STEP7 Basic — برای S7-1200
- STEP7 Lite — برای S7-300 و C7
- STEP7 — برای S7-400 و S7-300 و C7
- STEP7 Proffessional — شامل Step7 و برخی نرم‌افزارهای Optional
- STEP7 Micro/win — برای S7-200
- STEP5 — برای S5
- TISOFT — برای Ti 505
- LogoSoft — برای LOGO
- 4 mation — برای Quadlog و APACS

۷-۲-۱ STEP 7 Basic

این نرم‌افزار برای برنامه‌ریزی میکرو PLCهای S7-1200 به‌کار می‌رود. آخرین نسخه ارائه شده از این نرم‌افزار تا زمان تألیف این کتاب، Step7 basic V10.5 می‌باشد که روی سیستم عامل ویندوز XP و ویندوز Vista قابل نصب می‌باشد. با این نرم افزار علاوه بر برنامه‌نویسی PLCهای S7-1200 می‌توان برخی پنل‌های اپراتوری را نیز پیکربندی

۱. برنامه‌نویسی

کرد. در واقع نسخه محدودی از WinCC Flexible نیز همراه با Step7 V10.5 ارائه شده است. این نرم‌افزار برای S7-200 و S7-300 و S7-400 قابل استفاده نیست.



شکل ۷-۱ نرم‌افزار Step7 V10.5 برای برنامه‌نویسی S7-1200

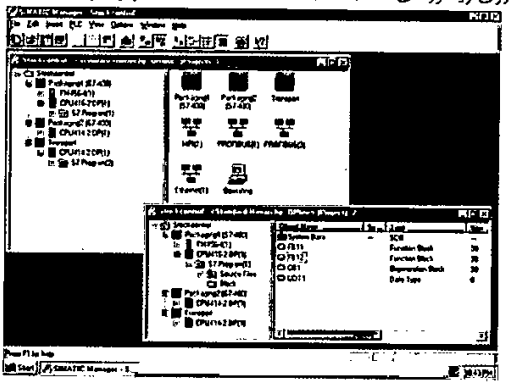
Step7 ۲-۲-۷

نرم‌افزار اصلی Programming سیستم‌های S7-300 و S7-400 و C7 است.

تا زمان تألیف این کتاب آخرین نسخه Step7 V5.5 می‌باشد که در DVD همراه کتاب ارائه شده است. این نرم‌افزار روی سیستم عامل‌های زیر قابل نصب است:

- Windows XP Professional SP2 or SP3
- Windows Server 2003 SP2 / R2 SP2 Standard Edition
- Windows 7 32 Bit Ultimate, Professional and Enterprise

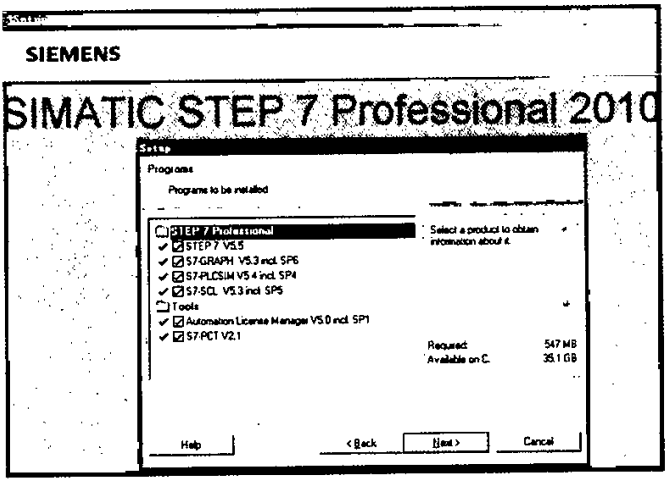
امکان نصب V5.5 روی ویندوزهای 95 , 98 , NT , 2000, Vista و ویندوز ۶۴ بیتی Win7 وجود ندارد.



شکل ۷-۲ نرم افزار Step7 خاص S7-300 و S7-400 و C7

Step7 Proffessional ۳-۲-۷

این نرم‌افزار مجموعه‌ای از چند نرم‌افزار مختلف است و علاوه بر Step7 برخی از ابزارهای تکمیلی آن را نیز دربر دارد.



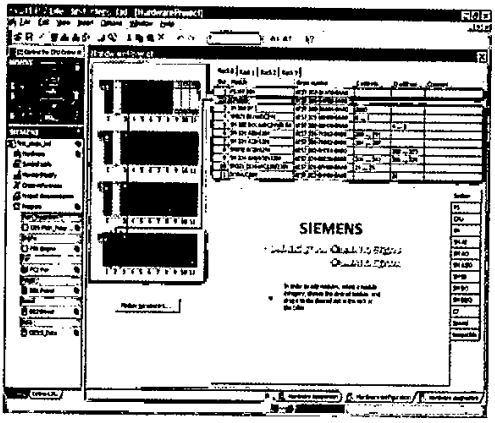
شکل ۷-۳ نرم‌افزار Step7 Proffessional

تا زمان تألیف این کتاب آخرین نسخه ارائه شده برای آن STEP 7 Professional 2010 می باشد که شامل نرم‌افزارهای زیر است:

- STEP 7 V5.5
- S7-SCL V5.3 SP5
- S7-GRAPH V5.3 SP6
- S7-PLCSIM V5.4 SP4

Step7 Lite ۴-۲-۷

این نرم‌افزار نسخه سبکی از Step7 است که توسط آن فقط می‌توان پیکربندی S7-300 و C7 را انجام داد و برای S7-400 قابل استفاده نیست. بعلاوه فقط ماژول‌های ورودی و خروجی محدودی را ساپورت می‌کند. با این نرم‌افزار می‌توان به سه زبان اصلی از زبان های IEC1131-3 یعنی LAD/FBD/STL برنامه نویسی کرد ولی نرم‌افزارهایی که برای دو زبان دیگر IEC1131 ارائه شده مانند SCL و Graph، روی آن قابل استفاده نیست.



شکل ۷-۴ نرم‌افزار Step7 Lite

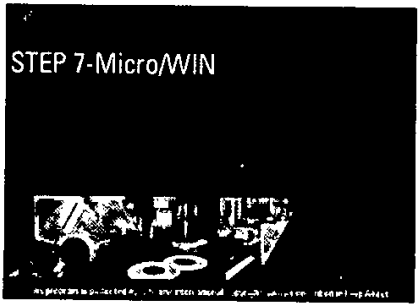
تا زمان تألیف این کتاب آخرین نسخه STEP 7 Lite V3.0 است که روی ویندوزهای زیر قابل نصب است:

- MS Windows XP Home SP2 or SP3
- MS Windows XP Professional SP2 or SP3

Step7 Microwin ۷-۲-۵

این نرم‌افزار برای برنامه‌نویسی S7-200 به‌کار می‌رود و می‌توان به‌طور همزمان آنرا با Step7 نصب کرد.
آخرین نسخه STEP 7-Micro/WIN V4.0 است که قابل نصب روی ویندوزهای زیر می‌باشد:

- Microsoft Windows2000 SP3 and higher
- Microsoft Windows XP Home
- Microsoft Windows XP Proffessional



شکل ۷-۵ نرم‌افزار Step7 Microwin

STEP5 ۶-۲-۷

برای برنامه‌نویسی PLCهای خانواده S5 به‌کار می‌رود. آخرین نسخه آن STEP 5 V7.2 است که در سال 2001 عرضه شده است و تحت ویندوز XP قابل نصب و اجراست.

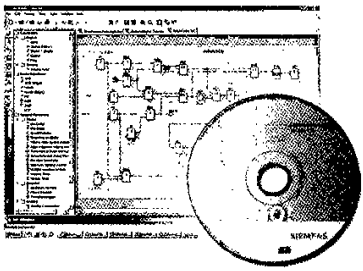
TISOFT ۷-۲-۷

برای برنامه‌نویسی PLCهای سری 505 و سری 500 ساخت Texas Instrument که امروزه با نام زیمنس عرضه می‌شوند، استفاده می‌گردد. نسخه TISOFT2 Release 6.3 که قبل از سال ۲۰۰۰ میلادی عرضه شده تحت DOS اجرا می‌گردد.

LOGO Soft Comfort ۸-۲-۷

مینی PLC های Logo را می‌توان توسط این نرم‌افزار برنامه‌نویسی کرد. آخرین نسخه ارائه شده تا این زمان V6.1 است که روی سیستم عامل‌های زیر قابل نصب است:

- Windows 98SE
- Windows NT 4.0
- Windows ME
- Windows XP (only 32-bit)
- Windows 2000
- Windows Vista (only 32-bit)
- Mac OS X 10.4
- Linux (tested with SUSE Linux 8.1)

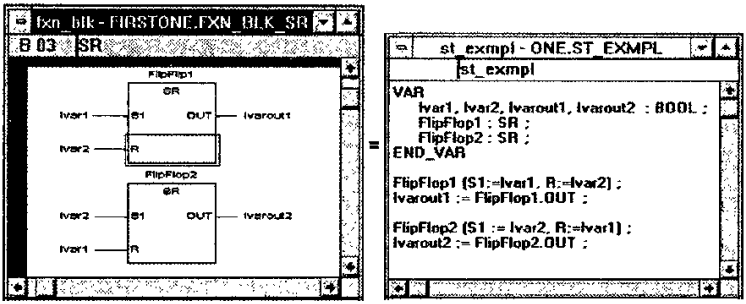


شکل ۷-۶ نرم‌افزار LOGO Soft Comfort

4mation ۹-۲-۷

این نرم‌افزار برای برنامه‌نویسی و تنظیمات کنترلرهای APACS و QUADLOG که ساخت شرکت Moore هستند به‌کار می‌رود.

آخرین نسخه عرضه شده تا این زمان process suite(4mation included) version 7 می باشد.



شکل ۷-۷ نرم‌افزار 4mation

۷-۳ نرم‌افزارهای تکمیلی Programming

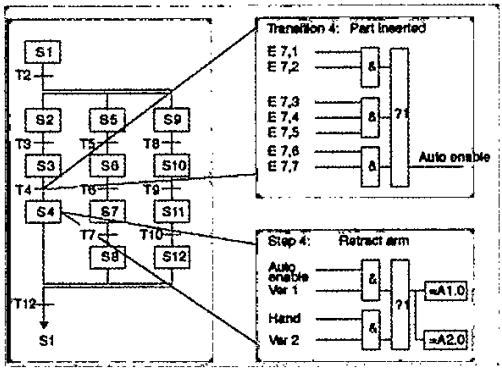
این نرم‌افزارها به‌تنهایی قابل نصب و استفاده نیستند و لازم است قبلاً نرم‌افزار اصلی برنامه‌نویسی (مانند Step7) نصب شده باشد تا بتوان آنها را نصب و اجرا نمود. لیست زیر نرم‌افزارهای تکمیلی برنامه‌نویسی را نشان می‌دهد:

- Graph 5/II
- S7-GRAPH
- S7-HIGRAPH
- S7-SCL
- CFC
- SFC
- F-SYSTEM
- H-SYSTEM
- SAFETY MATRIX
- M7 ProC/C++
- PLCSIM
- Standard PID Control
- Modular PID Control

۷-۳-۱ Graph 5/II

نرم‌افزار کنترل ترتیبی برای S5 و مطابق استاندارد IEC1131-3 می‌باشد. کنترل ترتیبیSequential Control یکی از نیازهای کاربران اتوماسیون صنعتی است. منظور از کنترل ترتیبی مدیریت عملیات یک فرآیند به‌صورت مراحل متوالی می‌باشد به‌گونه‌ای که تا هر مرحله انجام نشده است مرحله بعد شروع نشود. بر طبق استاندارد IEC61131-3 بایستی نرم‌افزار PLCها با روش SFC توسط سازنده ارائه شود و Graph برای همین منظور است.

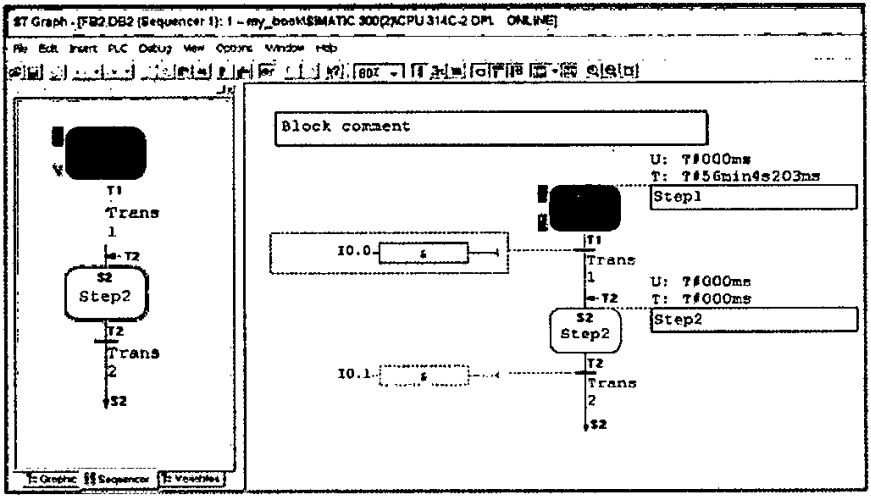
فصل ۷



شکل ۷-۸ ساختار محیط نرم‌افزار Graph 5/II

S7-Graph ۲-۳-۷

نرم افزارکنترل ترتیبی مشابه Graph 5 است که روی Step7 نصب می‌شود و یکی از اجزای Step7 Proffessional است.



شکل ۷-۹ محیط نرم‌افزار S7-Graph

این نرم افزار، یک نرم‌افزار گرافیکی به‌منظور کنترل فرآیندهای ترتیبی می‌باشد و با زبان SFC مندرج در استاندارد IEC 1131-3 تطابق دارد. ساختار این نرم‌افزار به‌گونه‌ای است که در آن، برنامه‌ی کاربر به مرحله‌های مختلفی تقسیم‌بندی می‌گردد. در هر مرحله می‌توان دستوراتی را اختصاص داد که در زمان فعال بودن مرحله اجرا گردند. همچنین

جهت عبور از هر مرحله به مرحله‌ی بعد می‌توان شرایطی را تعیین نمود که در صورت برقرار شدن شرایط مورد نظر، عبور از مرحله‌ی فعلی به مرحله‌ی بعدی صورت پذیرد.

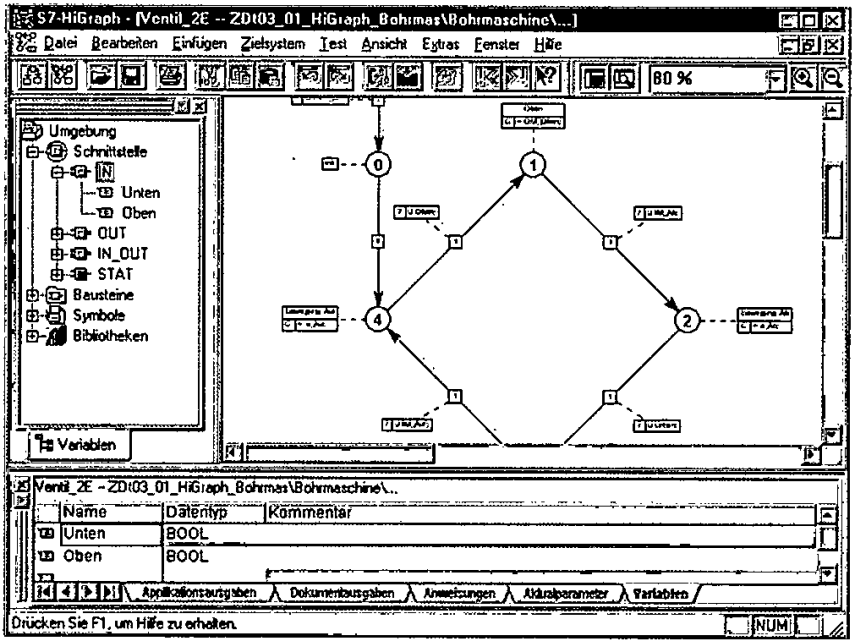
S7-Graph پس از کامپایل به کدهای STL تبدیل می‌گردد.

علاقمندان می‌توانند برای توضیحات بیشتر به کتاب برنامه نویسی به زبان S7-Graph تالیف احمد فرجی- علیرضا سینا مراجعه نمایند.

S7-Hi GRAPH ۳-۳-۷

این نرم‌افزار ابزار دیگری برای کنترل ترتیبی و پیاده‌سازی دیاگرام حالت است. ماشین‌هایی که سیکل مشخص و تکراری در عملیات فرآیندی را طی می‌کنند، می‌توانند به‌سهولت توسط این ابزار برنامه‌نویسی شوند. دستگاه‌های ماشین‌ابزار اتوماتیک از این جمله هستند که حالت‌های مختلف آنها به‌صورت گرافیکی ترسیم شده و فرمان‌های موتور، ولو و سایر عملگرها در هر حالت از ماشین تعریف می‌گردد. از اینرو این نرم‌افزار فرآیندگراست و عملکرد سیستم از روی آن به‌راحتی قابل درک و قابل ردیابی است.

خطوط مونتاژ، پرس‌ها، ماشین‌های ابزار و نوارنقاله‌ها از جمله مواردی هستند که کنترل آنها با HiGraph به‌سادگی پیاده‌سازی می‌شوند. این نرم‌افزار پس از Step7 نصب می‌شود و از طریق محیط Step7 قابل فراخوانی است.

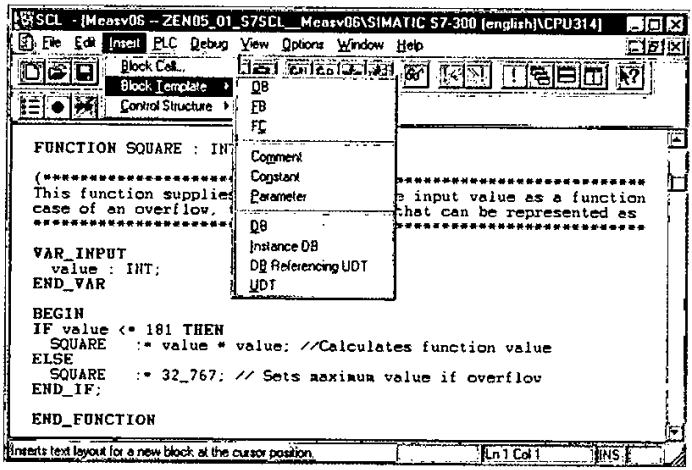


شکل ۷-۱۰ محیط نرم‌افزار Higraph

فصل ۷

S7-SCL [1] ۴-۳-۷

این نرم‌افزار زبان ST مندرج در استاندارد IEC1131-3 را پشتیبانی می‌کند. برنامه‌نویسی در آن به‌صورت ساختار یافته انجام می‌شود و دستورات و برنامه‌نوسی آن بسیار شبیه به زبان برنامه‌نویسی پاسکال است. این نرم‌افزار را می‌توان به‌صورت مستقل تهیه کرد و در عین حال در نسخه Step7 Proffessional موجود است. در هر صورت ابتدا بایستی Step7 و سپس این نرم‌افزار نصب گردد.



شکل ۷-۱۱ محیط نرم‌افزار S7-SCL

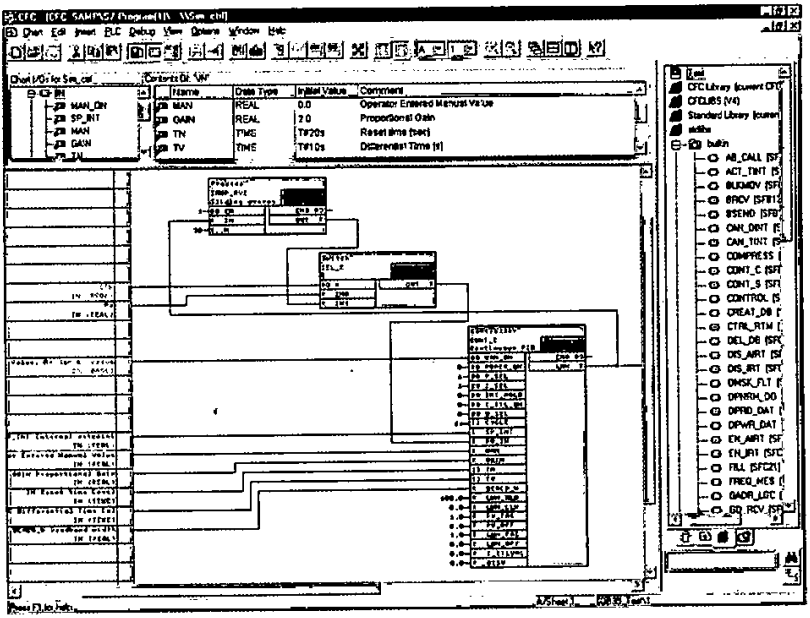
توضیحات کامل این نرم‌افزار در جلد سطح تکمیلی کتاب ارائه خواهد شد.

CFC ۵-۳-۷

Continuous Function Chartکه به اختصار CFC خوانده می‌شود، یک روش گرافیکی سطح بالا برای برنامه‌نویسی است که به‌طور خاص در PCS7 به‌کار می‌رود. این ابزار یکی از اجزای نرم‌افزار PCS7 محسوب می شود که در عین حال می‌تواند همراه با Step7 نیز به‌کار رود.

در CFC بلاک‌ها به‌صورت گرافیکی به یکدیگر متصل شده و منطق کنترل را شکل می‌دهند. CFC پس از کامپایل به کدهای STL تبدیل و به کنترلر دانلود می‌شود.

1. Structured Control Language



شکل ۷-۱۲ محیط نرم‌افزار CFC

علاقه‌مندان می‌توانند جهت توضیحات بیشتر در مورد CFC به کتاب مرجع کاربردی PCS7 تألیف محمدرضا ماهر مراجعه نمایند.

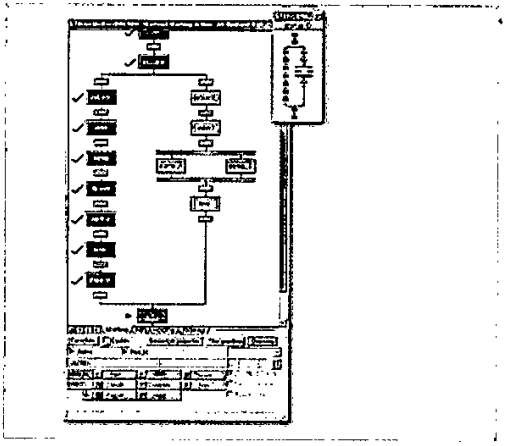
SFC ۶-۳-۷

SFC مخفف Sequential Function Chart ابزاری برای کنترل ترتیبی است که از اجزای نرم‌افزار PCS7 محسوب شده و مطابق استاندارد IEC61131-3 می‌باشد.

SFC به‌صورت گرافیکی است و توالی عملیات کنترل به‌ترتیب در آن مشخص می‌شود. SFC پس از کامپایل به کدهای STL تبدیل می‌گردد.

SFC نسبت به Graph توانایی‌های بیشتری را داراست ازجمله اینکه قابلیت مانیتورینگ مراحل انجام کار برای اپراتور بدون نیاز به محیط برنامه نویسی مهیا می‌باشد.

فصل ۷

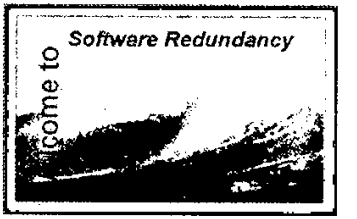


شکل ۷-۱۳ محیط نرم‌افزار SFC

علاقه‌مندان می‌توانند جهت توضیحات بیشتر در مورد SFC به جلد اول کتاب مرجع کاربردی PCS7 تألیف محمدرضا ماهر مراجعه نمایند.

SWR ۷-۳-۷

با استفاده از این نرم‌افزار می‌توان بین دو سیستم S7-300 یا S7-400 معمولی افزونگی برقرار کرد؛ یعنی با استفاده از دو سیستم معمولی S7-300 یا S7-400 یک سیستم افزونه[1] ایجاد نمود. بلاک‌های مخصوص افزونگی پس از نصب این نرم‌افزار به کتابخانه Step7 منتقل می‌شود. این نرم‌افزار بایستی روی Step7 نصب گردد و به تنهایی قابل استفاده نیست.

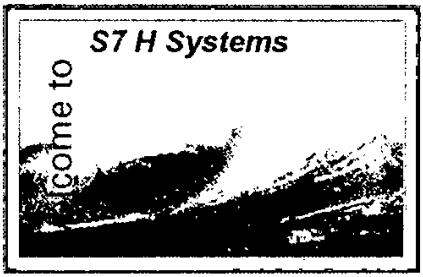


شکل ۷-۱۴ نرم‌افزار SWR

H-System ۸-۳-۷

نرم‌افزاری تکمیلی است که برای پیکربندی و برنامه‌نویسی سیستم‌های S7-400H مورد نیاز است. با نصب این نرم‌افزار، اجزای سخت‌افزاری S7-400H و برخی بلاک‌های مورد نیاز آن قابل دسترسی خواهد بود. نرم‌افزار H-System در نسخه Step7 Proffessional موجود است ولی اگر Step7 از نوع معمولی باشد فاقد آن می‌باشد.

1. Redundant

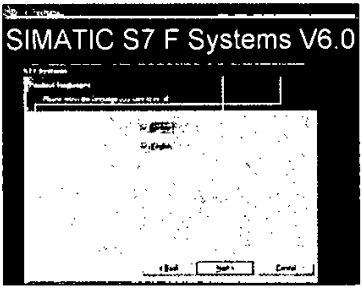


شکل ۷-۱۵ نرم‌افزار H-System

F-System ۹-۳-۷

نرم‌افزار تکمیلی است که برای پیکربندی و برنامه‌نویسی سیستم‌های Fail Safe مانند S7-300F و S7-400F مورد نیاز است.

با نصب این نرم‌افزار، بلاک‌های خاص Safety (ایمنی) به کتابخانه Step7 اضافه شده و زبان‌های برنامه‌نویسی F-LAD و F-FBD نیز قابل استفاده خواهند بود. این دو زبان برای S7-300F به‌کار می‌روند. برای S7-400F از CFC استفاده می‌شود که با نصب F-System بلاک‌های Safety به کتابخانه CFC اضافه می‌گردند.



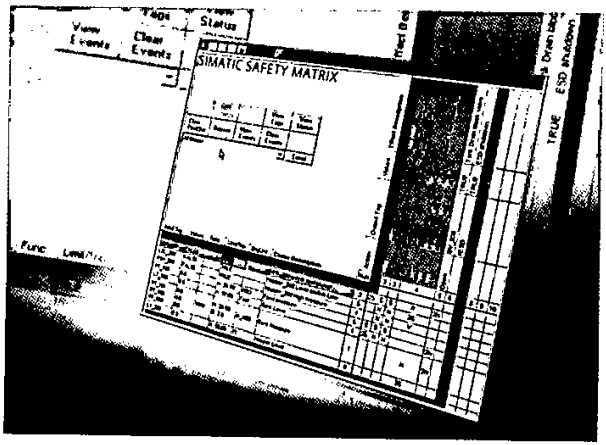
شکل ۷-۱۶ نرم‌افزار F-System

تذکر ۱: در سیستم‌های S7-400FH هر دو نرم‌افزار H-system و F-system مورد نیاز هستند.

تذکر ۲: در کاربردهای Fail Safe به‌ویژه وقتی از شبکه پروفی‌باس استفاده می‌شود علاوه بر F-System نرم‌افزارهای دیگر مانند F-Configuration Package و Distributed Safety نیز مورد نیاز هستند.

فصل ۷

Safety Matrix ۱۰-۳-۷



شکل ۷-۱۷ نرم‌افزار Safety Matrix

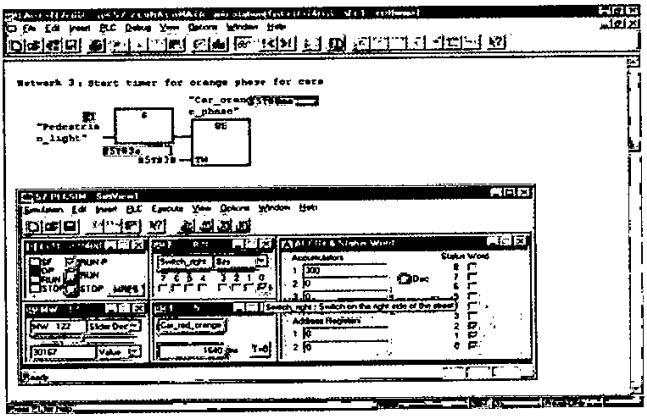
این نرم‌افزار برای سیستم‌های Fail Safe به‌کار می‌رود. ابزار سطح بالاست و به کمک آن جداول Cause and Effect تعریف می‌شود؛ به این معنی که تعیین می‌گردد در صورت بروز فالت چه عملی (Action) بایستی انجام گیرد. پس از تکمیل ماتریس فوق و کامپایل آن، برنامه Safety به‌صورت CFC تولید می‌گردد. برای استفاده از Safety Matrix لازم است قبلاً نرم‌افزارهای Step7 و F-System و CFC نصب شده باشد.

M7 ProC/C++ ۱۱-۳-۷

این ابزار به‌منظور برنامه‌نویسی PLCهای M7 به زبان C و C++ استفاده می‌گردد.

PLC SIM ۱۲-۳-۷

این نرم‌افزار به‌عنوان شبیه‌ساز PLCهای S7-300 و S7-400 به‌کار می‌رود. بسیاری از برنامه‌ها را می‌توان قبل از ارسال به PLC با آن تست و اشکالات احتمالی را برطرف نمود.

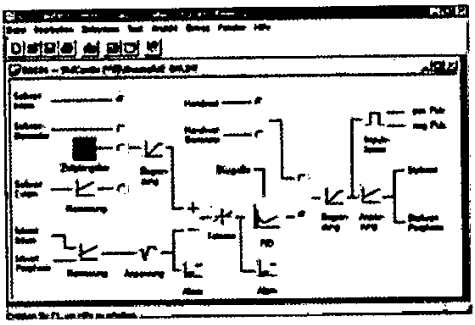


شکل ۷-۱۸ نرم‌افزار PLCSIM

این نرم‌افزار را اگرچه می‌توان به‌صورت مستقل تهیه و نصب کرد ولی در صورت دسترسی به نسخه Step7 Proffessional، در لیست نرم‌افزارهای انتخابی آن PLC SIM را نیز می‌بینیم. در هر صورت ابتدا بایستی نرم‌افزار اصلی Step7 نصب شده و سپس این نرم افزار نصب گردد.

Standard PID Control ۱۳-۳-۷

ابزار کمکی برای طراحی کنترل‌کننده‌های PID به‌صورت نرم‌افزاری می‌باشد که برای PLCهای S7-300 ، S7-400 و C7 استفاده می‌گردد.



شکل ۷-۱۹ نرم‌افزار Standard PID Control

Modular PID Control ۱۴-۳-۷

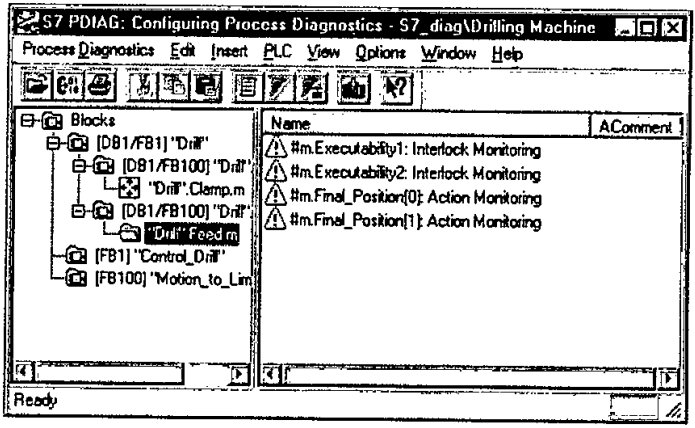
برای طراحی لوپ‌های کنترلی پیچیده استفاده می‌شود. این نرم‌افزار دارای فانکشن‌ها و بلاک‌های از قبل طراحی شده می‌باشد.

فصل ۷

۴-۷ نرم‌افزارهای عیب یابی و سرویس

PDIAG ۱-۴-۷

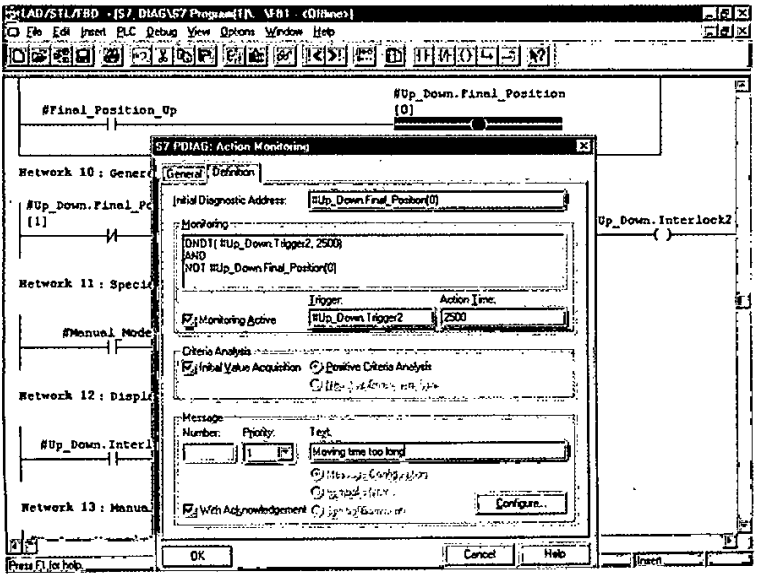
از Fault هایی که در یک سیستم رخ می‌دهد، درصد کمی مربوط به سیستم کنترل است. اکثر خطاها مربوط به فرآیند هستند. PDIAG که از دو کلمه Process Diagnostics گرفته شده، ابزار مناسبی است که با آن برنامه‌نویسی و مانیتورینگ خطاها به‌سادگی امکان‌پذیر می‌شود. این نرم‌افزار روی Step7 نصب شده و با به‌کار بردن فانکشن‌های آن در محیط برنامه‌نویسی، مشخصات و جزئیات خطاهای فرآیندی برای سیستم HMI ارسال می‌گردد تا برای اپراتور نمایش داده شود.



شکل ۷-۲۰ محیط نرم‌افزار S7-PDIAG

ابزار تکمیل کننده آن ProAgent است که پیام‌های تولید شده توسط PDIAG را در HMI نمایش می‌دهد. با استفاده از این دو ابزار با هرگونه تغییر در برنامه‌نویسی در سمت کنترل، تغییرات به‌طور اتوماتیک در HMI ایجاد شده و نیاز به کار مهندسی جداگانه‌ای نیست.

شکل ۷-۱۹ نحوه استفاده از PDIAG را در محیط برنامه‌نویسی نشان می دهد.



شکل ۷-۲۱ نحوه استفاده از PDIAG در محیط برنامه‌نویسی

TeleService ۲-۴-۷

با استفاده از این نرم‌افزار (و البته همراه با نصب سخت‌افزارهای خاص) امکان سرویس از راه دور فراهم می‌گردد. توسط این نرم‌افزار، اطلاعات یک Plant می‌تواند به سیستم دیگری در نقطه دور دست منتقل گردد و از طریق آن عیب‌یابی یا مانیتورینگ فرآیند انجام شود. سازنده سیستم می‌تواند سرویس مناسب به‌خصوص در دوران warranty ارائه نماید.
توسط این نرم‌افزار و با به‌کارگیری بلاک‌های خاص، سیستم می‌تواند به‌طور اتوماتیک SMS یا فکس یا ایمیل ارسال نماید. بعلاوه امکان تبادل دیتا بین دو Plant نیز فراهم می‌شود. Teleservice می‌تواند از طریق خط تلفن یا شبکه اترنت ارتباط بگیرد.

PRODAVE ۳-۴-۷

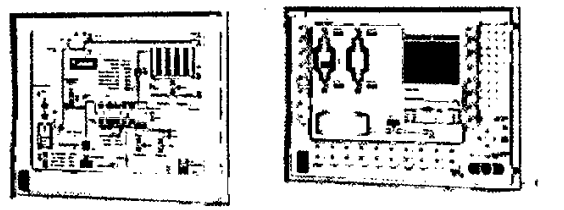
این نرم‌افزار می‌تواند از طریق شبکه‌های MPI یا Profibus یا Ethernet به PLC مرتبط شده و تمامی اطلاعات PLC را در محیط Runtime نمایش دهد. توسط این نرم‌افزار علاوه بر ورودی، خروجی، آدرس‌های حافظه، تایمر و کانتر، می‌توان اطلاعات PLC مانند شماره سفارش و ورژن سخت‌افزار و ... را نیز مانیتور نمود.

۷- ۵ نرم‌افزارهای مانیتورینگ

از مهمترین مزایای PLC می‌توان به امکان استفاده از مانیتورینگ صنعتی همراه با آن اشاره نمود. منظور از مانیتورینگ صنعتی نمایش گرافیکی پروسه به همراه جزئیات آن بر روی صفحات نمایش می‌باشد. از جمله قابلیت‌های مهم مانیتورینگ صنعتی می‌توان به موارد زیر اشاره نمود:

- امکان کنترل فرآیند با فرامین و نظر بهره‌بردار
- امکان نمایش گرافیکی پروسه
- امکان ایجاد و مدیریت آلارم‌ها
- امکان آرشیوسازی و بازیابی اطلاعات مربوط به رخدادها و آلارم‌های پیش آمده
- امکان رسم منحنی‌ها و جداول مختلف مربوط به پروسه

شرکت زیمنس به منظور انجام عمل مانیتورینگ نرم‌افزارها و سخت‌افزارهای مختلفی ارائه نموده است. از جمله سخت‌افزارهای مخصوص مانیتورینگ می‌توان به OP (TP) اشاره نمود. OP [1] به معنی پنل اپراتوری می‌باشد که توسط آن اپراتور می‌تواند دستوراتی را به فرایند تحت کنترل اعمال نماید. OPها خود دارای CPU بوده و مستقل از کامپیوتر عمل می‌نمایند. مدل لمسیOP به TP [2] مشهور است.



شکل ۷-۲۲ نمایی از TP

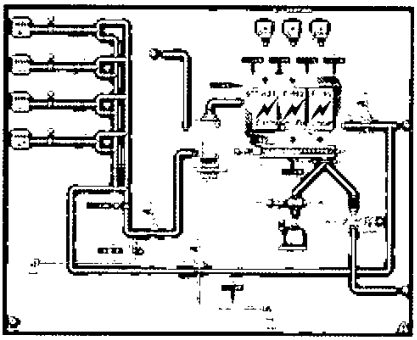
علاوه بر استفاده از OP، می‌توان از کامپیوتر نیز به‌عنوان سخت‌افزار مانیتورینگ استفاده نمود. در هر صورت علاوه بر سخت‌افزار، نیاز به نرم‌افزار مخصوصی جهت ایجاد صفحات گرافیکی، برنامه‌نویسی، و سایر موارد مربوط به مانیتورینگ می‌باشد. در ادامه به بررسی انواع نرم‌افزارهای مخصوص مانیتورینگ می‌پردازیم.

winCC ۱-۵-۷

با این نرم‌افزار سیستم مانیتورینگ معروف زیمنس یعنی WinCC طراحی می‌گردد. تعریف تگ‌ها، صفحات گرافیکی، آلارم‌ها، رخ دادها و اطلاعاتی که بایستی آرشیو شوند در محیط Engineering انجام شده و سپس به‌صورت Runtime برای اپراتور نمایش داده می‌شود. این نرم‌افزار دارای کتابخانه‌ای است که گرافیک بسیاری از اجزای صنایع مختلف در آن موجود است و کاربر جهت ایجاد صفحات گرافیکی می‌تواند از این اجزا استفاده نماید. به‌منظور انجام برنامه‌نویسی مربوط به مانیتورینگ، استفاده از زبان‌های برنامه‌نویسی C یا VB نیز در این محیط برنامه‌نویسی امکان‌پذیر می‌باشد.

تا زمان نگارش این کتاب آخرین نسخه عرضه شده WinCC 7.0 می‌باشد.

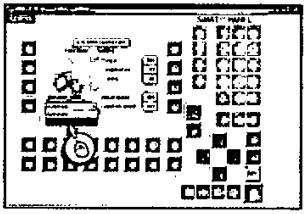
1. Operator Panel
2. Touch Panel



شکل ۷-۲۳ نمونه صفحه مانیتورینگ طراحی شده با نرم افزار wincc

WinCC Flexible ۷-۵-۲

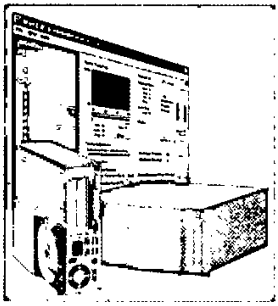
این نرم‌افزار برای پنل‌های اپراتوری جدید زیمنس به‌کار می‌رود و توسط آن گرافیک، آلارم، رخ‌داد و ارتباطات OP یا TP قابل تعریف است.



شکل ۷-۲۴ صفحه Touch Panel طراحی شده با WinCC Flexible

winAC ۷-۵-۳

این نرم‌افزار برای سیستم‌های PC based که در آن کامپیوتر نقش PLC را بازی می‌کند، به‌کار می‌رود.



شکل ۷-۲۵ نرم‌افزار و سخت‌افزار WinAC

فصل ۷

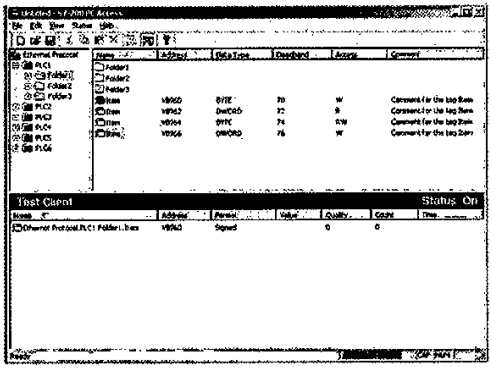
Protool ۴-۵-۷

OP و TPها قبلاً توسط این نرم‌افزار پیکربندی می‌شدند. در انواع جدید OP (TP) به‌جای Protool از winCC Flexible استفاده می‌شود.

شکل ۷-۲۶ نرم‌افزار Protool

PC Access ۵-۵-۷

این نرم‌افزار OPC مربوط به S7-200 می‌باشد و با استفاده از آن می‌توان اطلاعات S7-200 را تحت نرم‌افزارهای مانیتورینگ یا هر نرم‌افزار استاندارد دیگر در محیط ویندوز مانیتور نمود.



شکل ۷-۲۷ نرم‌افزار PC Access

۶-۷ نرم‌افزارهای مربوط به شبکه‌های صنعتی

Simatic NET ۱-۶-۷

نرم‌افزاری است که با نصب آن امکانات مختلفی برای کار با کارت‌های شبکه اترنت و پروفی‌باس فراهم می‌گردد. این نرم‌افزار به‌عنوان OPC زیمنس عرضه شده است و درایورهایی را در ویندوز می‌سازد که برای تمام نرم‌افزارهای دیگری که روی ویندوز نصب می‌شوند قابل دسترس خواهد بود.

iMAP ۲-۶-۷

نرم‌افزاری است که برای پیکربندی گرافیکی شبکه صنعتی Profinet به‌کار می‌رود. با استفاده از آن پیاده‌سازی این شبکه برای تبادل دیتا بین اجزای یک Plant ساده‌تر انجام می‌شود؛ بعلاوه امکانات و Diagnostic آن منجر به سهولت عیب‌یابی می‌شود.

PDM ۳-۶-۷

نرم‌افزار Process Device Manager که به اختصار PDM خوانده می‌شود برای پیکربندی و عیب‌یابی اجزای شبکه فیلدباس به‌کار می‌رود و توسط آن می‌توان Profibus-PA و HART را پیکربندی نمود. این نرم‌افزار یکی از ابزارهای PCS7 است.

۴-۶-۷ نرم‌افزارهای ارتباطی S5

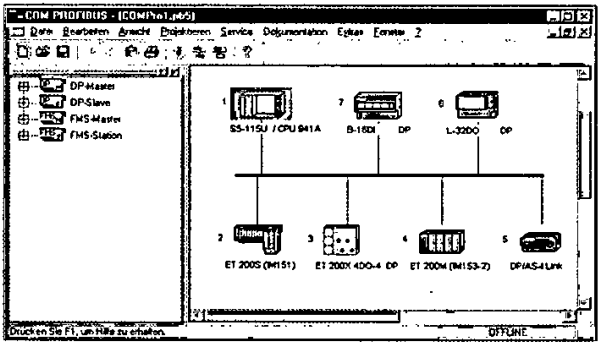
این نرم‌افزارها با پیشوند Com شروع می‌شوند و انواع مختلفی دارند که هر کدام برای کاربرد خاصی طراحی شده است، از جمله:

COM 246 / COM 247

برای تنظیمات کارت کنترل موقعیت S5 که به IP موسوم است به‌کار می‌رود.

Com profibus

برای پیکربندی و تنظیمات PLCهای خانواده S5 با شبکه پروفی‌باس که قبلاً در زیمنس Sincc L2 خوانده می‌شد، به‌کار می‌رود. توسط این نرم‌افزار می‌توان ET200ها را به‌عنوان Slave به PLCهای S5 معرفی نمود.



شکل ۷-۲۸ محیط نرم‌افزار Com profibus

Com5431

برای تنظیمات و پیکربندی کارت‌های شبکه پروفی‌باس CP 5431 از خانواده S5 به‌کار می‌رود. به‌عنوان مثال وقتی لازم است یک OP از طریق Profibus به S5 متصل شود، یا وقتی لازم است بین دو PLC S5 از طریق پروفی‌باس تبادل دیتا صورت گیرد این برنامه مورد نیاز می‌باشد.

Com 1430 /Com1437

ابزار پیکربندی و تنظیمات کارت‌های شبکه اترنت صنعتی است. Com1430 مربوط به TCP/IP که با آدرس IP کار می‌کند و COM1437 مربوط به MAP اترنت صنعتی است که با آدرس MAC کار می‌کند. به‌عنوان مثال برای ارتباط سیستم مانیتورینگ با S5 این نرم‌افزار مورد نیاز است.

COM115H/COM155H

این نرم‌افزار برای تنظیمات و برنامه‌نویسی PLCهای Redundant و Fault tolerant خانواده S5 (به‌عنوان مثال برای ارتباط S5-155H با CPU948RL) به‌کار می‌رود.

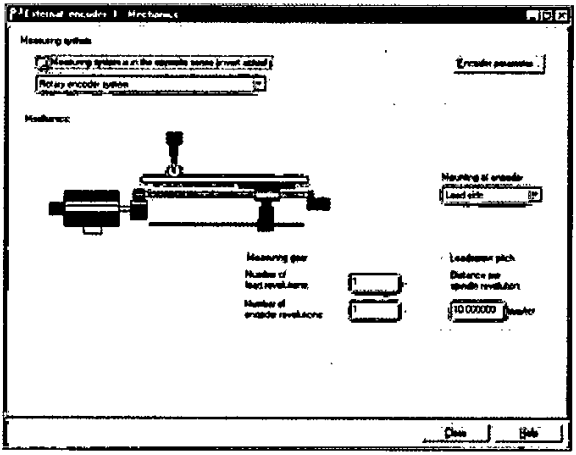
COM95F/ COM115F

این نرم‌افزار برای تنظیمات و برنامه‌نویسی PLCهای Fail Safe خانواده S5 به‌کار می‌رود.

۷-۷ نرم‌افزارهای مربوط به Motion Control

S7-Technology ۱-۷-۷

برخی از CPUهای خانواده S7-300 که با کد CPU31x-T مشخص می‌شوند برای کاربرد خاص مربوط به تکنولوژی Motion Control طراحی شده‌اند. پیکربندی و برنامه‌نویسی آنها با S7-Technology انجام می‌گیرد.



شکل ۷-۲۹ محیط نرم‌افزار S7-Technology

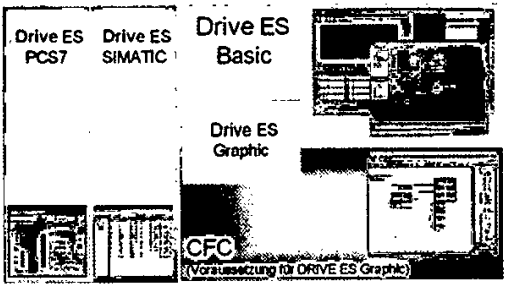
D7-Sys ۲-۷-۷

نرم‌افزاری است که دارای بلاک‌ها و امکانات زیادی برای کارهای ریاضی و کنترل‌های پیچیده و شبکه‌های ارتباطی است. این نرم‌افزار به‌صورت خاص در Motion Control استفاده می‌شود. از کاربردهای آن می‌توان به سیستم کنترل‌های TDC و SIMADYN اشاره کرد.

پس از نصب این نرم‌افزار، بلاک‌های آن در محیط Step7 و CFC و SFC قابل دسترس خواهند بود.

Drive ES ۳-۷-۷

نرم‌افزاری که با استفاده از آن، پیکربندی، ارتباطات و تبادل دیتا درایوهای زیمنس به‌سهولت در محیط Step7 انجام می‌گیرد. این نرم‌افزار دارای Optionهای مختلفی است که در شکل ۷-۳۰ نشان داده شده است.



شکل ۷-۳۰ نرم‌افزار Drive ES

۷-۸ برخی دیگر از نرم‌افزارهای مرتبط با کنترل و اتوماسیون

DOCPRO ۱-۸-۷

این نرم‌افزار برای مستندسازی پروژه به‌کار می‌رود. پس از اینکه کار پیکربندی و تنظیمات سخت‌افزار و شبکه به پایان رسید و برنامه‌نویسی کامل شد، می‌توان توسط این نرم‌افزار کل اطلاعات را به‌صورت مناسب در قالب‌های از پیش تعیین شده قرار داد و با فرمت مناسب چاپ نمود.

HARDPRO ۲-۸-۷

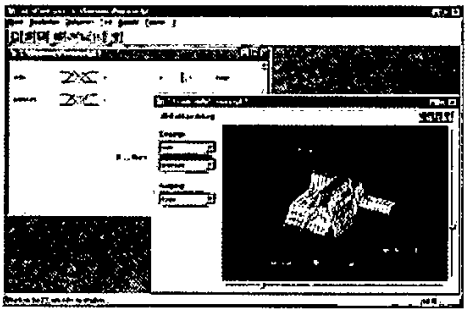
یک سیستم پیکربندی سخت‌افزاری برای PLC های S7-300 است که به کاربر کمک می‌کند تا پیکربندی سخت‌افزار را برای سیستم‌هایی که وظایف کنترلی پیچیده را انجام می دهند ساده‌تر انجام دهد.

FuzzyControl++ ۳-۸-۷

این نرم‌افزار برای پیاده‌سازی منطق فازی در PLC ارائه شده است. منطق فازی در سیستم کنترلی قابل استفاده است که منطق کنترل آن توسط معادلات ریاضی قابل توصیف نیست و دانش اپراتوری مبنای کنترل است. این دانش به قواعد فازی تبدیل شده و مورد استفاده قرار می‌گیرد. با نصب نرم‌افزار FuzzyControl++ امکان تعریف قواعد فازی و سایر نیازهای منطق فازی وجود دارد. PLC ورودی‌ها را از پروسه دریافت و در اختیار کامپیوتری که این برنامه روی آن نصب شده قرار می‌دهد، سپس FuzzyControl براساس پایگاه دانش خود فرامین لازم را تولید و در اختیار PLC قرار می‌دهد تا به خروجی‌ها اعمال گردد.

علاقمندان می‌توانند برای توضیحات بیشتر به کتاب پیاده‌سازی منطق فازی در PLC تألیف محمدرضا ماهر- غلامرضا آقاجری مراجعه نمایند.

فصل ۷

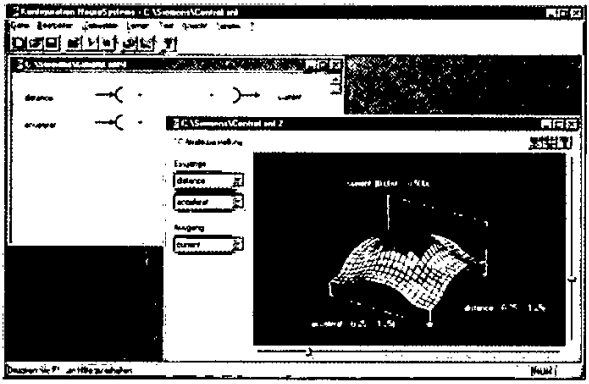


شکل ۷-۳۱ نرم‌افزار Fuzzy Control ++

Neurosystem ۴-۸-۷

در منطق فازی امکان توصیف رفتار سیستم با دستورات ساده IF ... THEN راه حل ساده‌ای است که با صرف زمان کوتاهی طراحی سیستم میسر می‌شود. ولی با وجود این مزیت، در بسیاری از کاربردها منطق فازی با محدودیت مواجه می‌شود؛ زیرا یافتن دانش مربوط به توصیف رفتار سیستم در میان خروارها اطلاعات زمان بسیار زیاد و دقت زیادی را طلب می‌کند.

در این شرایط شبکه عصبی می‌تواند راه حل مناسبی باشد زیرا شبکه عصبی توانایی یادگیری از روی مجموعه داده‌ها را دارد، به‌عبارت دیگر می‌تواند در برخی شرایط که داده‌ای از آن در پایگاه دانش وجود ندارد تصمیم‌گیری کند. پیاده‌سازی شبکه عصبی توسط نرم‌افزارهای خاصی انجام می‌شود که Neurosystem از این جمله است.



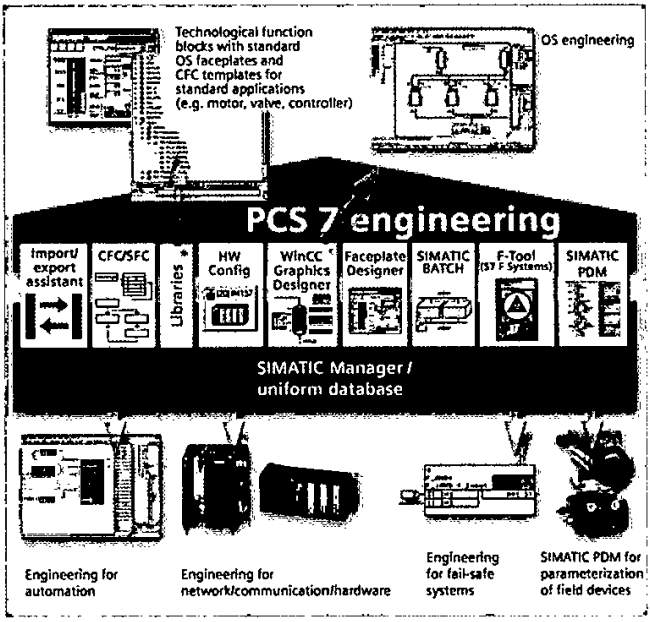
شکل ۷-۳۲ نرم‌افزار Neurosystem

Premium Studio ۵-۸-۷

مجموعه‌ای از نرم‌افزارهای مختلفی است که برای کاربردهای مختلف کنترل و مانیتورینگ استفاده می‌شود. نرم‌افزارهایی که برای S7، C7، Simatic Net، Simatic HMI و Sinumerik و Drive استفاده می‌شوند، در این مجموعه وجود دارند.

PCS7 ۶-۸-۷

نرم‌افزاری است که توسط آن سیستم DCS جدید زیمنس که PCS7 نام دارد پیکربندی و پیاده‌سازی می‌شود. PCS7 حاوی همه ابزارهای لازم برای کار مهندسی و اپراتوری است. نرم‌افزار Step7 یکی از اجزای PCS7 است که توسط آن پیکربندی سخت‌افزار و شبکه صورت می‌گیرد. برنامه کنترل در PCS7 با CFC و SFC نوشته می‌شود. نرم‌افزار WinCC نیز که برای مانیتورینگ سیستم است در ابزارهای PCS7 وجود دارد. علاقمندان می‌توانند برای جزئیات بیشتر به کتاب مرجع کاربردی PCS7 تالیف محمدرضا ماهر مراجعه نمایند.



شکل ۷-۳۳ نرم‌افزار PCS7

۷-۹ پرسش و تحقیق

- طراحی یک سیستم DCS توسط نرم‌افزار PCS7 چگونه است؟
- سیستم عصبی فازی در چه صنایعی می‌تواند کاربرد داشته باشد؟

فصل ۷

۷-۱۰ تست‌های خودآزمایی

۱- برنامه‌ریزی PLCهای S7-1200 توسط کدام‌یک از نرم‌افزارهای زیر امکان‌پذیر است؟
الف) STEP 7-micro/win ب) STEP 7 Basik
ج) STEP 7 Lite د) STEP 7

۲- کدامیک از نرم‌افزارهای زیر جهت پیکربندی S7-400 به‌کار می‌رود؟
الف) STEP 7-micro/win ب) STEP 7 Basik
ج) STEP 7 Lite د) STEP 7

۳- نرم‌افزار..... مجموعه‌ای‌از چندنرم‌افزار مختلف است و علاوه بر Step7 برخی از ابزارهای تکمیلی آن را نیز در بر دارد.
الف) STEP5 ب) STEP 7 Lite
ج) STEP 7 د) Step7 Proffessional

۴- کدامیک از نرم‌افزارهای زیر جهت برنامه‌نویسی پروسه‌های ترتیبی به‌کار می‌رود؟
الف) CFC ب) SCL ج) S7-GRAPH د) هر سه مورد

۵- در این نرم‌افزار توسط توابع از قبل آماده شده برنامه‌نویسی انجام می‌پذیرد.
الف) CFC ب) SCL ج) S7-GRAPH د) هر سه مورد

۶- نرم‌افزار..... زبان ST مندرج در استاندارد IEC1131-3 را پشتیبانی می‌کند. برنامه‌نویسی و دستورات آن شبیه زبان برنامه‌نویسی پاسکال است.
الف) CFC ب) SCL ج) S7-GRAPH د) SFC

۷- نرم‌افزار..... به‌منظور برنامه‌نویسی پروسه‌های ترتیبی استفاده شده و کاربرد اصلی آن در PCS7 است.
الف) CFC ب) SCL ج) S7-GRAPH د) SFC

۸- نرم‌افزار..... به‌عنوان شبیه‌ساز PLCهای S7-300 و S7-400 به‌کار می‌رود.
الف) PDIAG ب) TeleService ج) PLC SIM د) هیچ‌کدام

۹- با استفاده از این نرم‌افزار می‌توان بین دو سیستم S7-300 یا S7-400 معمولی افزونگی برقرار کرد.
الف) SWR ب) F SYSTEM ج) H SYSTEM د) موارد الف و ج

۱۰- با استفاده از این نرم‌افزار امکان سرویس از راه دور (از طریق خط تلفن) فراهم می‌گردد.
الف) PDIAG ب) TeleService ج) PRODAVE د) M7-SYS RT

۱۱- ..... ابزار مناسبی است که با آن برنامه‌نویسی و مانیتورینگ خطاها به‌سادگی امکان‌پذیر می‌شود.
الف) PDIAG ب) TeleService ج) PRODAVE د) M7-SYS

۱۲- نرم‌افزار..... می‌تواند از طریق شبکه‌های MPI یا Profibus یا Ethernet به PLC مرتبط شده و تمامی اطلاعات PLC را در محیط Runtime نمایش دهد.
الف) PDIAG ب) TeleService ج) PRODAVE د) M7-SYS

۱۳- کدامیک از نرم‌افزارهای زیر به‌منظور برنامه‌ریزی پنل‌های اپراتوری (OP) به‌کار می‌رود؟
الف) Wincc ب) Wincc fliexible ج) PC Access د) WinAC

۱۴- نرم‌افزار..... برای پیکربندی و عیب‌یابی اجزای شبکه فیلدباس به‌کار می‌رود.
الف) PDIAG ب) Simatic NET ج) iMAP د) PDM

فصل ۸

پیش‌نیازهای کار با نرم‌افزار Step7

۸-۱ مقدمه

۸-۲ نیاز به Step7 در مراحل مختلف کار اتوماسیون

۸-۳ نسخه‌های مختلف Step7

۸-۴ نیازمندی‌های نصب نرم‌افزار STEP 7

۸-۴-۱ سیستم عامل

۸-۴-۲ مشخصات سخت‌افزاری

۸-۵ نحوه نصب نرم‌افزار Step7 V5.5

۸-۶ مجوز نرم‌افزار Step 7 (Authorization)

۸-۷ تنظیم ارتباط بین PLC و کامپیوتر

۸-۸ انواع روش‌های ارتباط بین PLC و کامپیوتر

۸-۸-۱ ارتباط بین کامپیوتر و پورت روی CPU

۸-۸-۲ ارتباط بین کامپیوتر و کارت شبکه نصب شده روی PLC

۸-۹ وسایل رابط بین PLC و کامپیوتر

۸-۹-۱ PC Adapter

۸-۹-۲ کارت PCMCIA برای لپ‌تاپ

۸-۹-۳ کارت‌های قابل نصب روی مادربورد کامپیوتر

۸-۱۰ استفاده از لپ‌تاپ خاص زیمنس PG

۸-۱۱ پرسش و تحقیق

۸-۱۲ تست‌های خودآزمایی

در این فصل سیستم عامل و سخت‌افزار مناسب برای نصبStep7 و تنظیمات لازم برای کار با آن مورد بحث قرار گرفته است.

چکیده مطالب

- آخرین نسخه عرضه شده برای Step7 تا زمان تألیف این کتاب نسخه Step7 V 5.5 می‌باشد.
- در فاز طراحی می‌توان از Step7 به‌صورت Offline برای بررسی پیکربندی سخت‌افزار و شبکه و طراحی برنامه کنترل استفاده کرد و تست‌های اولیه را با سیمولاتور انجام داد.
- در فاز تست و راه‌اندازی می‌توان از Step7 برای چک‌کردن صحت اتصالات استفاده کرد.
- در فاز بهره‌برداری از Step7 برای عیب‌یابی و نمایش پارامترها استفاده می‌شود.
- نسخه Step7 V5.5 فقط روی ویندوزهای XP و 2003 و Win 7 (32 bit) قابل نصب است.
- سخت‌افزار کامپیوتر باید دارای حداقل مشخصات لازم باشد.
- در حین نصب یا پس از نصب بایستی کلیدهای Authorization نرم‌افزار نصب گردد، در غیر این‌صورت به‌صورت Trial فقط تا ۱۴ روز کار می‌کند.
- برای ارتباط بین PLC با PC لازم است رابط‌های خاص تهیه شود.
- رابط می‌تواند یک آداپتور یا یک کارت شبکه باشد که روی کامپیوتر نصب می‌شود.
- اگر از PG زیمنس استفاده شود آداپتور یا کارت شبکه مورد نیاز نیست.
- تنظیمات رابط بایستی به‌درستی در کنترل‌پنل ویندوز انجام شود.

فصل ۸

اصطلاحات و تعاریف

DP

DP مخفف Decentralized Peripheral است و معرفی یکی از پروتکل‌های شبکه پروفی‌باس می‌باشد که در سال ۱۹۹۲ عرضه شده است.

Profinet

پروتکل جدیدی است که بر مبنای شبکه اترنت صنعتی است که با اطمینان بالا کار می‌کند. امروزه Remote I/O هایی برای اتصال به این شبکه ارائه شده است.

MPI

Multipoint Interface پروتکل انحصاری زیمنس است که برای پروگرامینگ PLCها از آن استفاده می‌شود.

PCMCIA

Personal Computer Memory Card International Association معرف کارت‌های کوچکی است که در لپ‌تاپ قابل نصب است.

CP

Communication Processor معرف کارت شبکه زیمنس است.

۸-۱ مقدمه

همانطور که در فصل‌های قبل اشاره شد، برای کار با PLCهای خانواده S7 نیاز به نرم افزار Step7 می‌باشد. نام Step7 به همه نرم‌افزارهای زیر به‌نحوی اطلاق می‌شود که هر کدام برای کار با PLC خاصی از خانواده S7 طراحی شده است:

۱.	STEP7 Micro/win	برای S7-200
۲.	STEP7 Basic	برای S7-1200
۳.	STEP7 Lite	برای S7-300 و C7
۴.	STEP7	برای S7-400 و S7-300 و C7
۵.	STEP7 Professional	شامل Step7 و برخی نرم‌افزارهای Optional

فصل ۸

از آنجا که در این کتاب محور بحث براساس PLCهای S7-300 و S7-400 است، هرجا کلمه Step7 را به‌کار می بریم منظور مورد ۴ یا ۵ از لیست فوق است.

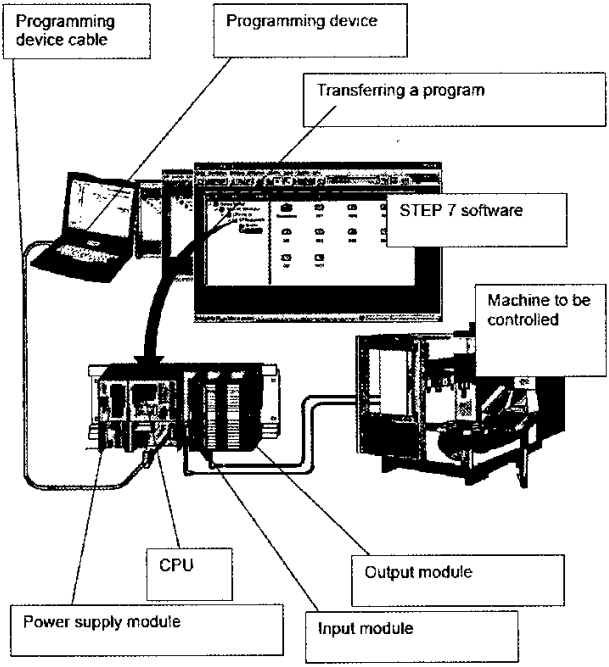
نرم افزار Step7 فقط برای برنامه‌نویسی PLC نیست، توسط این نرم‌افزار می‌توان:

- پیکربندی سخت‌افزار PLC را تعریف کرد.
- شبکه‌بندی PLC را مشخص نمود.
- برنامه‌نویسی انجام داد.
- برنامه را قبل از ارسال به PLC اصلی با شبیه‌ساز تست کرد (در نسخه Professional).
- سیستم PLC را قبل از برنامه‌نویسی از نظر سخت‌افزار و صحت اتصالات چک کرد.
- اطلاعات سخت‌افزار و شبکه و برنامه را از کامپیوتر به PLC انتقال داد (Download).
- اطلاعات از قبل بار شده به PLC را به کامپیوتر منتقل نمود (Upload).
- PLC را Stop یا Run یا Reset نمود.
- وضعیت سیگنال‌ها را در حین کار مشاهده کرد.
- در صورت بروز فالت سیستم را عیب‌یابی نمود.
- آرشیوسازی و تهیه مستندات
- و . . .

۸-۲ نیاز به Step7 در مراحل مختلف کار اتوماسیون

با توجه به نکات فوق می‌بینیم که یک مهندس اتوماسیون در فازهای مختلف کاری (طراحی، تست و راه‌اندازی، بهره‌برداری) نیاز به استفاده از این نرم‌افزار خواهد داشت.

ابتدا شکل ساده زیر که PLC را در کنار ماشین و کامپیوتر مهندسی نشان می‌دهد در نظر بگیرید.



شکل ۸-۱ ارتباط PLC با سایر اجزا

با توجه به اجزای نشان داده شده در شکل ۸-۱، نیازمندی‌ها در مراحل مختلف کاری به‌شرح زیر است:

فاز طراحی

در هنگام طراحی معمولاً نیازی به اینکه PLC یا ماشین در کنار PC یا PG موجود باشند نیست. فقط لازم است که قبل از شروع کار، فرآیند به‌خوبی مطالعه شده، ورودی و خروجی‌ها مشخص باشند و منطق سیستم کنترل معلوم شده باشد. بهتر است سخت‌افزار PLC نیز انتخاب شده باشد. با چنین معلوماتی می‌توان کار طراحی را با استفاده از Step7 به‌صورت Offline یعنی بدون اتصال به PLC انجام داد. منظور از کار طراحی Offline بررسی چیدمان سخت‌افزار روی Rack یا Remote I/O و انجام شبکه‌بندی و طراحی برنامه است.

در این مرحله می‌توان برنامه را توسط سیمولاتور (شبیه‌ساز) نرم‌افزاری تست کرد. سپس اشکالات اولیه را رفع نمود و تغییرات لازم را انجام داد.

Commissioning [1]

در این مرحله ماشین یا تجهیز نیز به جمع قبلی می‌پیوندد. تغذیه PLC وصل می‌شود و ابتدا فقط اطلاعات سخت‌افزار توسط نرم‌افزار Step7 به PLC دانلود می‌گردد. با ابزارهای موجود در Step7 می‌توان وضعیت سخت‌افزار و سیم‌کشی‌ها را چک کرد. به‌عنوان مثال اگر یک سنسور در سایت به‌طور دستی فعال شود و در نرم‌افزار Step7 وضعیت On شدن آن در کانال مورد نظر دیده شود، نشانگر این است که سنسور و کابل و سیم‌کشی‌ها صحیح هستند.

پس از تست سخت‌افزار، برنامه به PLC ارسال می‌شود. برنامه به‌صورت عملی و ابتدا در حالتی که هنوز از تجهیز بهره‌برداری نمی‌شود، تست می‌گردد که به این مرحله تست سرد[2] نیز می‌گویند. سیگنال‌ها به‌تدریج (نه یک‌دفعه) وارد مدار شده بخش‌های برنامه قدم به قدم تست می‌گردد. پس از آن تست گرم ماشین شروع می‌شود، یعنی از تجهیز به‌صورت آزمایشی بهره‌برداری می‌شود تا سایر ورودی و خروجی‌هایی که در تست سرد فعال نبودند تست گردند. برای انجام تست‌های فوق وجود Step7 روی PC یا PG و ارتباط Online با PLC ضروری است.

فصل ۸

Operation [3]

پس از تکمیل مراحل تست و اعمال تغییرات لازم در برنامه PLC، کار عادی فرآیند شروع می‌شود. در اینجا نیازی به PG یا PC و نرم‌افزار Step7 نیست؛ اگرچه باید برای نیازهای احتمالی، ادوات فوق در دسترس باشند.

Troubleshooting [4]

در صورتی‌که مشکلی در کار بهره‌برداری از فرآیند پیش بیاید که ناشی از اجزای سیستم کنترلی باشد، مجدداً به PG یا PC و نرم‌افزار Step7 نیاز پیدا می‌شود. این برنامه با امکانات مختلفی که در آن تعبیه شده می‌تواند به شناخت عیب و رفع آن کمک نماید.

۸-۳ نسخه‌های مختلف Step7

تاکنون نسخه های مختلفی برای نرم‌افزار Step7 عرضه شده است، ابتدا Version 3 سپس Version 4 و پس از آن Version 5 عرضه شده است. نسخه‌های متداول موجود همگی از خانواده Version 5 هستند. آخرین نسخه‌ای که تا زمان تألیف این کتاب عرضه شده Step7 V5.5 می‌باشد که در سال ۲۰۱۰ میلادی ارائه گردیده و در DVD همراه کتاب در اختیار خوانندگان قرار می‌گیرد.

کلیه مطالب گفته شده در این کتاب مربوط به نرم افزار Step 7 V5.5 می‌باشد.

قابلیت‌های اضافه شده در نسخه‌های جدید Step 7

از نسخه‌ی 5.1 نرم‌افزار Step 7 قابلیت‌های زیادی به نرم‌افزار اضافه گردیده است که مهمترین آنها به شرح ذیل می‌باشند:

---

۱. تست و راه‌اندازی

2. Cold Test

۳. بهره‌برداری

۴. عیب‌یابی

- قابلیت پیکربندی سخت‌افزار در مد کاری RUN که اصطلاحاً به آن Cir (Configuration in Run) گفته می‌شود. در پروسه‌هایی که هیچ توقفی نباید ایجاد گردد، توسط این قابلیت می‌توان در مد کاری RUN پیکربندی سخت‌افزار را تغییر داد. مثلاً می‌توان یک ماژول جدید به PLC اضافه نمود. در این حال وقفه‌ای کمتر از ۱ ثانیه به پروسه اعمال می‌گردد. در طول مدت وقفه، ورودی‌ها و خروجی‌ها آخرین حالت خود را حفظ می‌کنند. Cir جهت CPUهای S7-400 از Firmware 3.1 به بعد امکان‌پذیر است.
- از Step 7 V5.4 به بعد دو روش جهت نمایش زمان و تاریخ قرار داده شده است.
- از Step 7 V5.4 به بعد می‌توان دستیابی به پروژه‌ها و کتابخانه‌ها را توسط اختصاص پسورد به آنها محدود نمود. جهت این کار باید نرم‌افزار Simatic logon V1.3 sp1 روی سیستم نصب باشد.

Step 7 Professional

در این نسخه علاوه بر Step 7 V5.5، نرم‌افزارهای دیگری که قبلاً به صورت اختیاری[1] عرضه می‌شدند، یکجا ارائه می‌گردند. این نرم‌افزارها عبارتند از:

- S7- PLC Sim
- S7- GRAPH
- S7- PDIAG
- S7- SCL

لازم به ذکر است که برنامه‌ها و پروژه‌هایی که با نسخه‌های پایین ایجاد شده‌اند بدون مشکل در نسخه‌های جدیدتر باز می‌شوند ولی حالت برعکس همیشه امکان‌پذیر نیست.

۸-۴ نیازمندی‌های نصب نرم‌افزار STEP 7

به منظور نصب صحیح و استفاده از نرم‌افزار Step7 لازم است ملزومات سخت‌افزاری و نرم‌افزاری زیر رعایت گردند.

۸-۴-۱ سیستم عامل

هر کدام از نسخه‌های مختلف نرم‌افزار Step7 با نسخه‌های خاصی از سیستم عامل ویندوز سازگاری دارند.

برای نسخه Step7 5.5 فقط سیستم عامل‌های زیر می‌تواند استفاده شود:

- MS Windows XP Professional SP2 or SP3
- MS Windows Server 2003 SP2 / R2 SP2 Standard Edition as workstation
- MS Windows 7 32 Bit Ultimate, Professional and Enterprise (Standard Installation).

همانطور که مشاهده می‌شود امکان نصب Step7 V5.5 روی ویندوزهای Windows 95 / 98 / Me / NT / Windows 2000وجود ندارد و نصب این نرم‌افزار روی ویندوز Vista تست نشده است. جدول ۸-۱ سازگاری نسخه 5.5 و 5.4 با سیستم عامل‌های مختلف ویندوز را نشان می‌دهد.

1. Optional

جدول ۸-۱ سیستم عامل های مناسب برای Step7

STEP 7 Basic Packages		Microsoft Windows Operating Systems									
Product Name	Version	Win 2000 SP4	Win XP SP1	Win XP SP2	Win XP SP3	Win 2003	Win 2003 SP1	Win 2003 SP2	Vista	Vista SP1	Win 7
STEP 7	V5.5	–	–	X	X	–	–	X	(–)	(–)	X
STEP 7	V5.4 SP5	–	–	X	X	–	–	X	X	X	–
STEP 7	V5.4 SP4	–	–	X	X	–	–	X	X	X	–
STEP 7	V5.4 SP3.1	X	–	X	–	–	–	X	X	–	–
STEP 7	V5.4 SP3	X	–	X	–	–	–	X	X	–	–
STEP 7	V5.4 SP2	X	X	X	–	X	X	X	–	–	–
STEP 7	V5.4 SP1	X	X	X	–	X	X	–	–	–	–
STEP 7	V5.4	X	X	X	–	X	X	–	–	–	–
STEP 7 Professional	2010	–	–	X	X	–	–	X	(–)	(–)	X
STEP 7 Professional	2006 SR6	–	–	X	X	–	–	X	X	X	–
STEP 7 Professional	2006 SR5	–	–	X	–	–	–	X	X	X	–
STEP 7 Professional	2006 SR4	X	–	X	–	–	–	X	X	–	–
STEP 7 Professional	2006 SR3	X	–	X	–	–	–	X	X	–	–
STEP 7 Professional	2006 SR2	X	X	X	–	X	X	–	–	–	–
STEP 7 Professional	2006 SR1	X	X	X	–	X	X	–	–	–	–
STEP 7 Professional	2006	X	X	X	–	X	X	–	–	–	–

Legend: X : combination permitted
– : combination not permitted
(–) : combination not tested

۸-۴-۲ مشخصات سخت‌افزاری

جدول ۸-۲ حداقل مشخصات سخت‌افزاری لازم به منظور نصب نرم‌افزار Step7 V5.5 را نشان می‌دهد. به منظور نصب و استفاده صحیح از نرم‌افزار، لازم است مشخصات ذکر شده به‌عنوان حداقل رعایت گردد.

جدول ۸-۲ حداقل ملزومات سخت‌افزاری جهت نصب Step7 V5.5

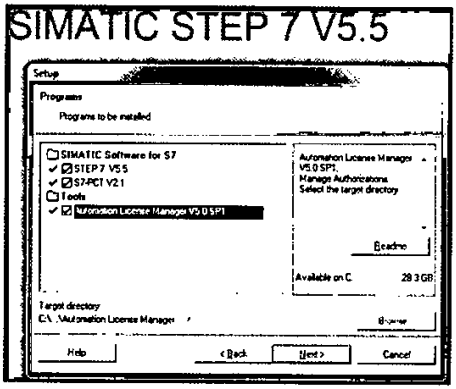
Operating System	Minimum Requirements for:		
	Processor	Expanded memory configuration	Graphics
MS Windows XP Professional	600MHz	512 MB *)	XGA 1024x768 16-bit color depth
MS Windows Server 2003	2.4 GHz	1 GB	XGA 1024x768 16-bit color depth
MS Windows 7 Professional	1 GHz	1GB **)	XGA 1024x768 16-bit color depth
MS Windows 7 Ultimate	1 GHz	1GB **)	XGA 1024x768 16-bit color depth
MS Windows 7 Enterprise	1 GHz	1GB **)	XGA 1024x768 16-bit color depth

*) At least 1 GB expanded memory configuration is recommended
**) At least 2 GB expanded memory configuration is recommended

۸-۵ نحوه نصب نرم‌افزار Step7 V5.5

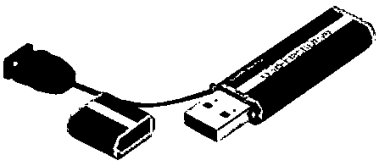
با در نظر گرفتن نکاتی که تا اینجا ذکر شد نصب نرم‌افزار Step7 پیچیدگی خاصی ندارد. فایل Setup را اجرا کرده و قدم به قدم جلو می‌رویم.

وقتی پنجره نشان داده شده در شکل ۸-۲ ظاهر شد، همه گزینه‌ها را انتخاب کرده و مسیر نصب نرم‌افزار که در پایین پنجره در قسمت Target Directory آمده است را چک نموده و در صورت لزوم تغییر می‌دهیم.



شکل ۸-۲ نصب Step7 V5.5

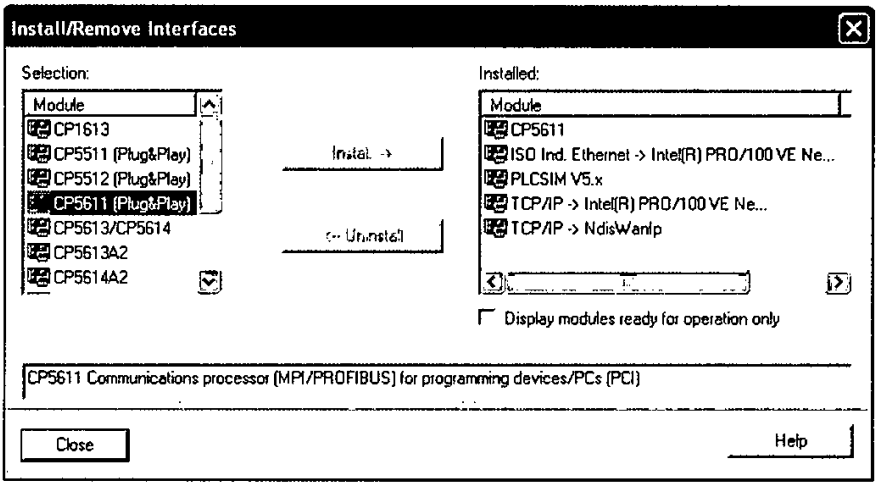
برای کار با Step7 نیاز به نصب مجوز خاصی[1] می‌باشد. این مجوز برای نسخه‌های قدیمی روی فلاپی‌دیسک تهیه شده و در اختیار کاربران قرار می‌گرفت. اما از نرم‌افزارStep7 V5.4 SP3 به بعد مجوز مورد نظر بر روی یک USB Memory Stick ارائه می‌گردد. در هنگام نصب نرم‌افزار اگر پیامی مبنی بر نصب License نشان داده شد، در صورتی‌که USB حاوی License موجود است آنرا وارد کرده و گزینه Yes را انتخاب کنید. در غیر اینصورت یا اگر تصمیم به استفاده از نسخه آزمایشی ۱۴ روزه دارید، گزینه NO را انتخاب نمایید.



شکل ۸-۳ USB حاوی License

در پایان نصب با پنجره نشان داده شده در شکل ۸-۴ مواجه می‌شویم که تنظیم رابط بین کامپیوتر و PLC را سوال می‌کند. در صورت عدم آشنایی می‌توان آنرا Cancel کرد. این تنظیم بعداً نیز قابل انجام است. نحوه انجام این تنظیمات در ادامه ارائه خواهد شد.

1. License Key

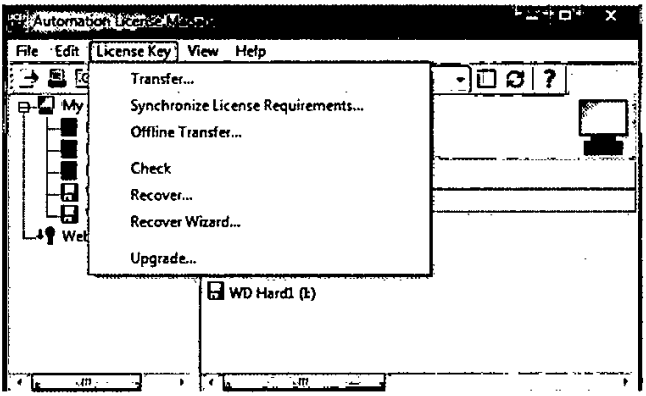


شکل ۸-۴ تنظیم ارتباط PLC با کامپیوتر

فصل ۸

۸-۶ مجوز نرم‌افزار Step 7 (Authorization)

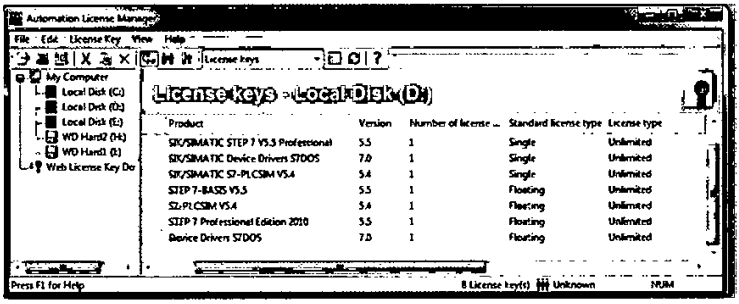
هنگامی‌که نرم‌افزار Step7 نصب می‌گردد، در حین نصب از وجود License سوال می‌کند. اگر License در آن مرحله نصب نشود، نرم‌افزار به مدت ۱۴ روز دارای مجوز استفاده رایگان می‌باشد. اگر License موجود است، می‌توان آنرا پس از اتمام نصب از طریق نرم‌افزار Automation License Manager نیز نصب نمود. این نرم‌افزار همراه با Step7 نصب می‌گردد. با اجرای Automation License Manager برنامه پنجره نشان داده شده در شکل ۸-۵ را خواهیم داشت. از منوی License Key می‌توان انتقال کلید از USB به درایو مورد نظر را انجام داد و فرقی نمی‌کند که کدام درایو انتخاب شود.



شکل ۸-۵ نرم‌افزار نصب License

در صورت نیاز می‌توان کلید را از هارد کامپیوتر موجود دوباره به USB منتقل کرد تا آنرا در کامپیوتر دیگری نصب نمود. توصیه می‌شود وقتی کلید به هارد منتقل شد، از هارد یا پارتیشن مربوطه Image گرفته شود تا در شرایط اضطراری قابل بازیافت باشد.

پس از اینکه کلید به هارد کامپیوتر منتقل شد می‌توان آنرا در پنجره‌ای مانند شکل ۸-۶ دید.



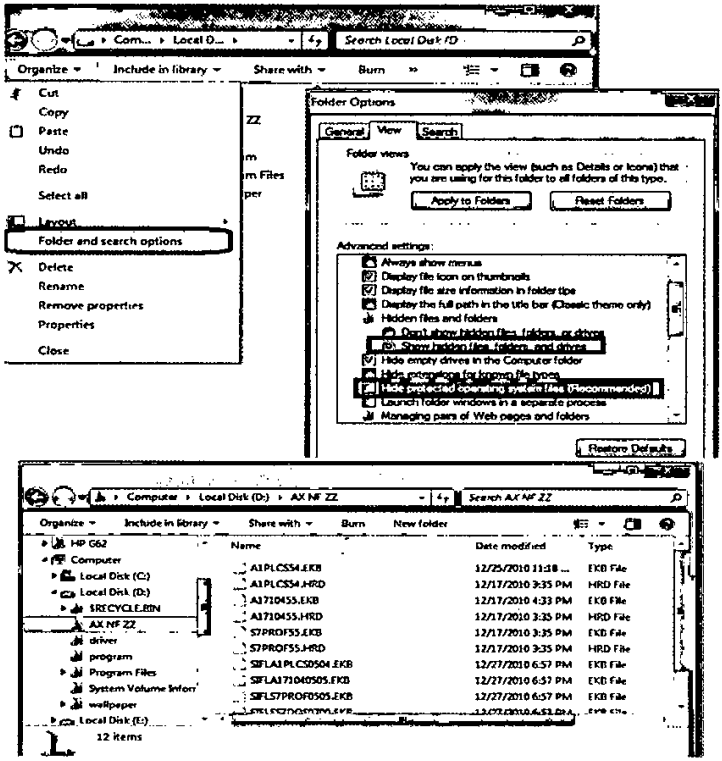
شکل ۸-۶ مشاهده کلیدهای نصب شده

کلیدها به انواع زیر تقسیم می‌شوند:

- Single که برای استفاده فقط روی یک کامپیوتر است.
- Floating که قابل استفاده روی شبکه می‌باشد.
- Rental : کاربرد محدود در قالب قرارداد اجاره‌ای دارد.
- Trial : کلیدی که فقط تا ۱۴ روز اعتبار دارد.
- Upgrade : وقتی قبلاً کلید نرم‌افزار خریداری شده و سپس نرم‌افزار ارتقا پیدا کند، می‌توان از این نوع کلید استفاده کرد.

کلیدهای فوق توسط این برنامه در یک فولدر مخفی بنام AX NF ZZ می‌نشیند. برای مشاهده آن تنظیم پنجره ویندوز را مانند شکل ۸-۷ تغییر دهید.

پس از آن، Licenseهای نصب شده را در فولدر AX NF ZZ خواهید دید.



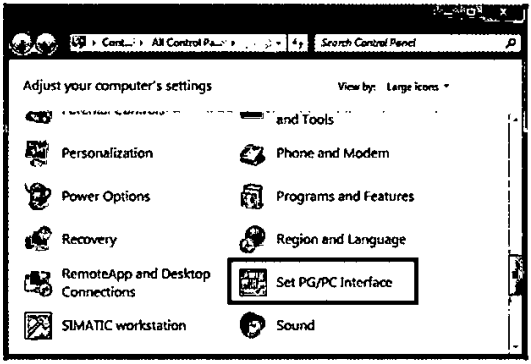
شکل ۸-۷ فولدر حاوی کلیدها

۸-۷ تنظیم ارتباط بین PLC و کامپیوتر

پس از نصب Step7 لازم است تعیین کنیم که از چه روشی قرار است کامپیوتر با PLC ارتباط برقرار نماید. این کار همانطور که گفته شد می‌تواند در آخرین مرحله نصب در پنجره‌ای که باز می‌شود انجام شود. با این وجود می‌توان این تنظیم را حتی پس از نصب نرم افزار به دو روش زیر تعیین کرد یا تغییر داد:

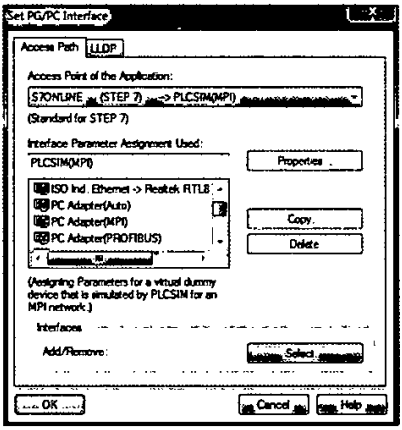
۱- از طریق منوهای نرم‌افزار Step7 که در فصل بعد تشریح می‌شود.

۲- از طریق گزینه Set PG/PC در Control Panel.



شکل ۸-۸ کنترل‌پنل ویندوز

با کلیک روی Set PG/PC پنجره‌ای مانند شکل ۸-۹ باز می‌شود که تنظیمات ارتباط بین PLC و کامپیوتر از طریق آن قابل انجام است.



شکل ۸-۹ برنامه Set PG/PC Interface

برای شناخت تنظیمات لازم است ابتدا با انواع روش‌های ارتباط بین PLC و PC آشنا شویم. از آنجا که هر روز آداپتورها و رابط‌های جدیدی عرضه می‌شوند در اینجا به مواردی که تا زمان تألیف این کتاب عرضه شده‌اند موجود هستند اشاره می‌گردد.

۸-۸ انواع روش‌های ارتباط بین PLC و کامپیوتر

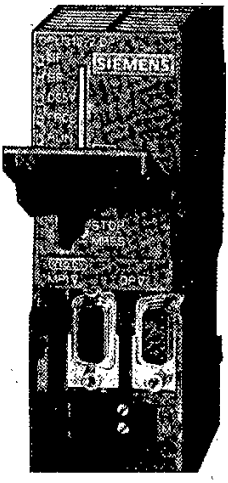
ارتباط بین کامپیوتر و PLC به دو نوع اصلی تقسیم می‌شود:

- ارتباط بین کامپیوتر و پورت روی CPU
- ارتباط بین کامپیوتر و کارت شبکه نصب شده روی PLC

۸-۸-۱ ارتباط بین کامپیوتر و "پورت" روی CPU

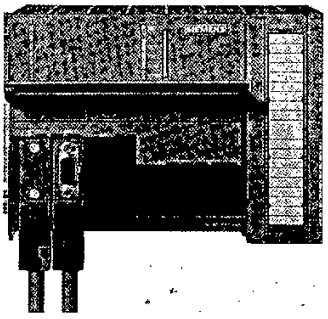
CPUهای خانواده S7-300 و S7-400 بسیار متنوع هستند و پورت‌های روی آنها می‌تواند شامل موارد زیر باشد:

MPI: این پورت روی همه CPUهای 400و 300 موجود است. MPI اگر چه براساس RS485 است ولی به‌صورت یک پروتکل انحصاری توسط زیمنس طراحی شده است و مهمترین کاربرد آن در Programming است. اگر توسط رابطهایی که در ادامه معرفی می‌شوند کامپیوتر را به این پورت متصل کنیم آنرا شناسایی کرده و از همان ابتدا می‌توان با PLC ارتباط برقرار کرد.



شکل ۸-۱۰ پورت MPI روی CPU (پورت سمت چپ)

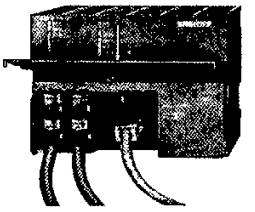
DP: این پورت معرف شبکه پروفی‌باس است که فقط روی برخی از CPUهای 400 و 300 موجود است. شبکه پروفی‌باس انحصاری نیست و دارای پروتکل باز است که می‌توان هر وسیله پروفی‌باس از هر سازنده‌ای را به آن متصل کرد.



شکل ۸-۱۱ CPU با پورت MPI و DP

کاربرد اصلی این پورت برای ارتباط با Remote I/Oهاست. در عین حال می‌توان از آن جهت ارتباط با کامپیوتر از طریق رابط خاص نیز استفاده کرد. به‌طور معمول این پورت از ابتدا فعال نیست، بنابراین در اولین دانلود نمی‌توان از آن استفاده کرد. لازم است ابتدا پورت را در محیط Step7 فعال نمود و اولین دانلود را با MPI انجام داد تا پورت DP فعال شود. پس از آن می‌توان کامپیوتر را با رابط خاص به این پورت متصل نمود.

IE: این پورت معرف شبکه اترنت صنعتی است که پروتکل استاندارد دارد و می‌توان وسایل شبکه اترنت یا پروفی‌نت را به آن‌متصل کرد. حرفPN روی‌CPU معرف‌این پورت است که فقط روی برخی از CPUهای 400 و 300 دیده می شود.



شکل ۸-۱۲ CPU با پورت MPI و DP و IE

به‌سادگی می‌توان پورت LAN کامپیوتر یا لپ‌تاپ را به این پورت متصل نمود و نیاز به رابط خاصی نیست ولی مشروط بر اینکه قبلاً این پورت تنظیم شده و توسط MPI به PLC معرفی شده باشد. بنابراین ممکن است این روش در اولین دانلود قابل استفاده نباشد.

۸-۸-۲ ارتباط بین کامپیوتر و کارت شبکه نصب شده روی PLC

با نصب کارت شبکه اترنت یا کارت شبکه پروفی‌باس در کنار CPU، می‌توان از آن برای ارتباط با کامپیوتر استفاده نمود. اگر کارت پروفی‌باس روی PLC موجود باشد، نیاز به کارت شبکه پروفی‌باس در سمت کامپیوتر وجود دارد تا بتواند با آن ارتباط برقرار کند ولی اگر کارت اترنت روی PLC موجود باشد، در سمت کامپیوتر نیاز به کارت خاصی نیست و با پورت LAN معمولی می‌توان به آن متصل شد.

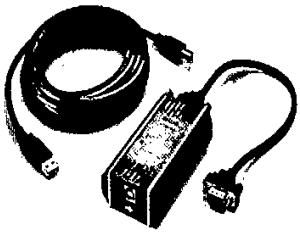
امروزه استفاده از کارت اترنت نسبت به پروفی‌باس متداول‌تر است. لازم به ذکر است که در سیستم‌های 400 اولین دانلود می‌تواند از طریق اترنت صنعتی انجام شود ولی در PLCهای 300 این کار عمدتاً امکان‌پذیر نیست. در این CPUها ابتدا بایستی با MPI اولین دانلود را انجام داد تا کارت اترنت فعال‌شود، سپس دانلود از طریق اترنت قابل استفاده خواهد بود.

۸-۹ وسایل رابط بین PLC و کامپیوتر

۸-۹-۱ PC Adapter

این وسیله به پورت MPI که روی همه CPUهای S7-300 و S7-400 موجود است و پورت انحصاری زیمنس محسوب می‌شود، قابل اتصال است. در عین حال می‌توان آنرا به پورت DP در CPUهایی که این پورت را دارند نیز متصل نمود.

PC Adapter از نظر ارتباط با کامپیوتر در دو مدل عرضه شده است: یکی نوع USB و دیگری نوع COM می‌باشد. امروزه نوع Com کمتر استفاده می‌شود و نوع USB متداول‌تر است.

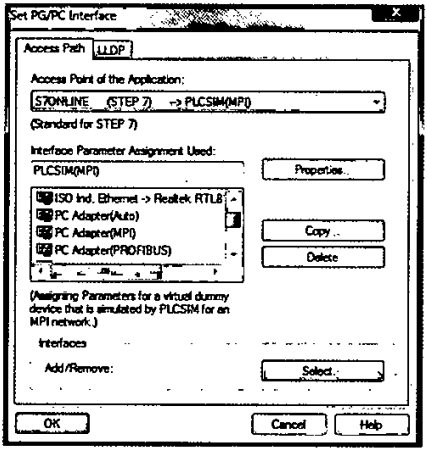


شکل ۸-۱۳ USB Adapter

فصل ۸

نوع USB برخلاف COM دارای هیچ دیپ‌سوئیچی برای تنظیم سرعت نیست ولی بر خلاف Com بایستی نرم‌افزار درایور آن نصب شود تا توسط کامپیوتر قابل شناسایی باشد.

برای تنظیم نوع USB در پنجره Set PG/PC همانطور که در شکل ۸-۱۴ دیده می‌شود، سه گزینه وجود دارد.



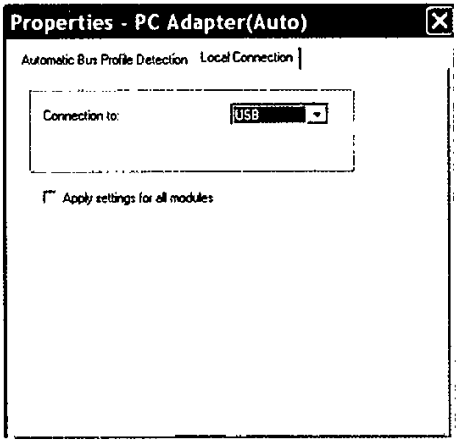
شکل ۸-۱۴ تنظیمات ارتباط آداپتور

گزینه‌های تنظیم ارتباط آداپتور

- PC Adapter (MPI) : وقتی از این تنظیم استفاده می‌شود که آداپتور به پورت MPI متصل باشد.
- PC Adapter (Profibus): وقتی از این تنظیم استفاده می‌شود که آداپتور به پورت پروفی‌باس متصل باشد. همانطور که قبلاً ذکر شد در اولین ارتباط نمی‌توان از این پورت استفاده کرد و لازم است ابتدا شبکه پروفی‌باس توسط Step7 تعریف شده و از طریق MPI دانلود شده باشد تا این پورت فعال گردد. پس از آن می‌توان آداپتور را به این پورت متصل و استفاده نمود. در این‌حالت آداپتور خود را با سرعت تنظیم شده برای پروفی‌باس (به شرط اینکه بیش از 1.5 Mbps نباشد) تطبیق می دهد.
- PC Adapter (Auto): این تنظیم هر دو حالت MPI و Profibus را اتوماتیک شناسایی می‌کند.

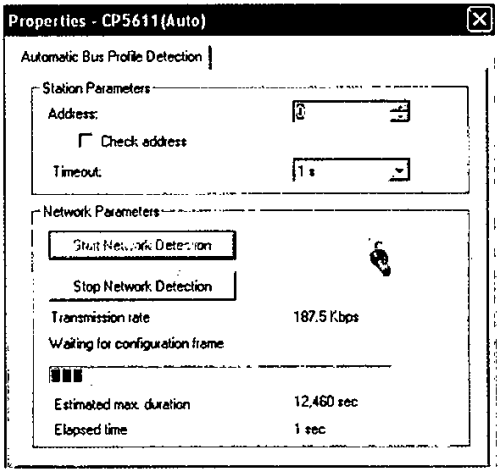
توجه: اگر در ارتباط بین PC Adapter و PLC مشکل پیدا کردید موارد زیر را چک کنید:

- مطمئن شوید که نرم‌افزار درایور USB نصب شده و کامپیوتر ری‌استارت شده باشد.
- مطمئن شوید که اتصال آداپتور به‌طور صحیح برقرار شده است. وقتی آداپتور به پورت CPU وصل می‌شود هر دو چراغ Power و MPI آن روشن و وقتی به کامپیوتر وصل می‌شود، چراغ USB آن باید روشن باشد.
- با انتخاب گزینه Auto روی قسمت Properties در شکل ۸-۱۴ کلیک کرده و در قسمت Local Connection مطمئن شوید که گزینه USB به‌درستی انتخاب شده است.



شکل ۸-۱۵ تنظیم پورت USB برای آداپتور

- برای تست در قسمت Automatic Bus Profile Connection روی گزینه Start network detection کلیک کرده و اگر شکل ۸-۱۶ ظاهر شد ارتباط سالم است.



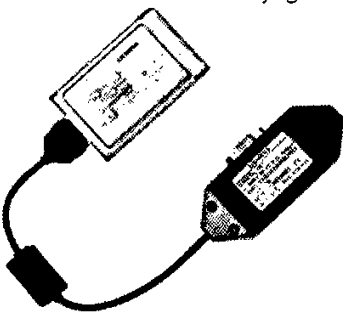
شکل ۸-۱۶ چک کردن ارتباط آداپتور

۸-۹-۲ کارت PCMCIA برای لپ‌تاپ

کارت‌های متنوعی برای نصب در اسلات PCMCIA لپ‌تاپ عرضه شده‌اند. برخی MPI و پروفی‌باس را ساپورت می‌کنند و برخی دیگر مربوط به شبکه اترنت صنعتی هستند. نمونه‌هایی از این کارت‌ها عبارتند از:

الف - کارت CP5511 و کارت CP5512

این دو کارت برای ارتباط با MPI یا DP عرضه شده است که می‌توان آنها را به پورت روی CPU یا به پورت کارت شبکه پروفی‌باس نصب شده روی PLC متصل کرد.



شکل ۸-۱۷ CP5511

ب- کارت CP1512

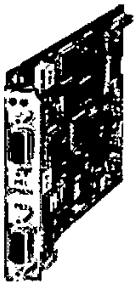
این کارت برای ارتباط با شبکه اترنت صنعتی عرضه شده است که می‌توان آنرا به پورت روی CPU یا به پورت کارت شبکه اترنت نصب شده روی PLC متصل کرد.

شکل ۸-۱۸ CP1512

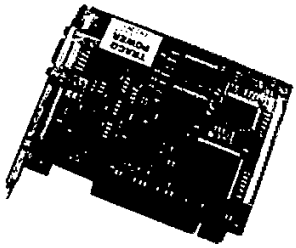
۸-۹-۳ کارت‌های قابل نصب روی مادربورد کامپیوتر

الف- کارت‌های MPI/DP

کارت‌های نشان داده شده در شکل ۸-۱۹ نمونه‌هایی از کارت‌های شبکه پروفی‌باس هستند که روی مادربورد کامپیوتر نصب می‌شوند. این کارت‌ها MPI را نیز ساپورت می‌کنند و می‌توان آنها را به CPU یا کارت شبکه پروفی‌باس نصب شده روی PLC متصل نمود.



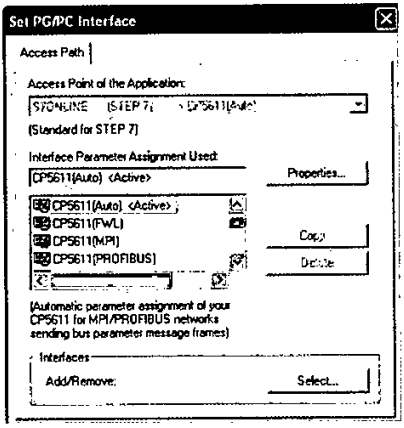
CP5624



CP5611

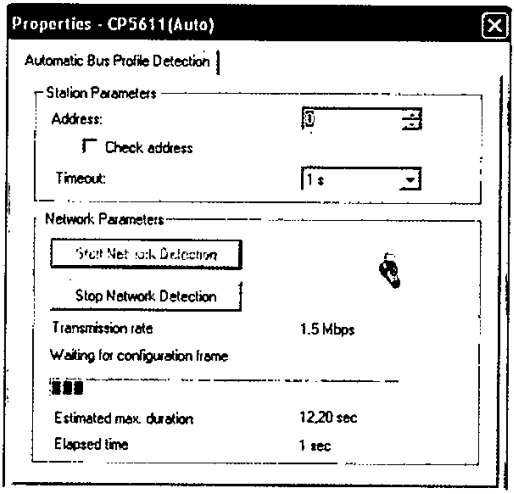
شکل ۸-۱۹ CP پروفی‌باس جهت نصب در مادربورد

CP5611 برخلاف CP5624 و CP5623 نیاز به نصب درایور ندارد و بصورت Plug and Play می‌باشد. پس از نصب این کارت روی مادربورد به‌طور اتوماتیک تشخیص داده شده و در پنجره Set PG/PC ظاهر می‌شود.



شکل ۸-۲۰ تنظیم ارتباط CP5611

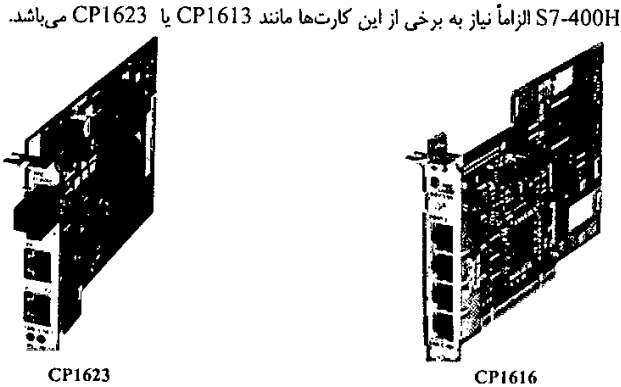
با تنظیم آن روی حالت Auto در بخش Properties می‌توان از صحت ارتباط مطمئن شد. شکل ۸-۲۱ چک کردن ارتباط این کارت را با پورت پروفی‌باس روی CPU نشان می‌دهد.



شکل ۸-۲۱ چک کردن ارتباط CP5611

ب- کارت‌های اترنت

شکل ۸-۲۲ نمونه‌هایی از این کارت‌ها را نشان می‌دهد. لازم است توجه شود که با پورت LAN معمولی و بدون نیاز به این کارت‌ها نیز می‌توان بین کامپیوتر و PLC ارتباط برقرار نمود. در برخی از کاربردهای خاص مانند مانیتورینگ سیستم‌های افزونه S7-400H الزاماً نیاز به برخی از این کارت‌ها مانند CP1613 یا CP1623 می‌باشد.

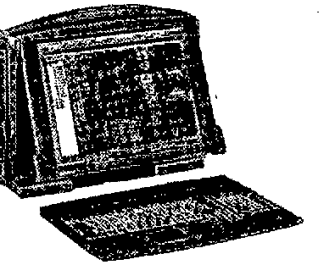


شکل ۸-۲۲ کارت اترنت جهت نصب روی مادربورد

۸-۱۰ استفاده از لپ‌تاپ خاص زیمنس PG

به‌منظور تسهیل در برنامه‌ریزی PLCهای زیمنس، این شرکت اقدام به ارائه لپ‌تاپ صنعتی به نام PG نموده است. اگر به‌جای کامپیوتر از PG استفاده شود نیازی به استفاده از مبدل‌های فوق نیست. PGهای زیمنس دارای پورت خروجی که مستقیماً به PLC وصل می‌گردد هستند.

Field PG



Power PG

شکل ۸-۲۳ PG های زیمنس

۸-۱۱ پرسش و تحقیق

چرا برای مانیتورینگ سیستم‌های افزونه S7-400H استفاده از کارت CP1613 یا CP1623 الزامی است؟

۸-۱۲ تست‌های خودآزمایی

۱- در مورد فاز طراحی یک سیستم کنترل کدام گزینه صحیح است؟

الف) نیاز به قرارگیری PLC در کنار فرآیند می‌باشد.

ب) توسط نرم‌افزار Step7 به‌صورت Online طراحی صورت می‌پذیرد.

ج) توسط نرم‌افزار Step7 به‌صورت Offline طراحی صورت می‌پذیرد.

د) بدون استفاده از نرم‌افزار Step7 طراحی صورت می‌پذیرد.

۲- اصلی‌ترین کاربرد پورت MPI کدام گزینه است؟

الف) تبادل دیتا بین چند PLC ب) تبادل دیتا در مانیتورینگ

ج) برنامه‌ریزی PLC د) ارتباط با Remote I/O

۴- در PLCهای S7-300، برای بار اول دانلود از طریق کدام پورت انجام می‌گیرد؟

الف) پورت MPI ب) پورت DP ج) پورت PN د) هر سه مورد

۵- کاربرد اصلی پورت DP کدام گزینه است؟

الف) تبادل دیتا بین چند PLC ب) تبادل دیتا در مانیتورینگ

ج) برنامه‌ریزی PLC د) ارتباط با Remote I/O

۶- در مورد PC adapter کدام گزینه صحیح است؟

الف) می‌توان از طریق آن ارتباط بین PLC و شبکه اترنت را ایجاد نمود.

ب) کاربرد اصلی آن در شبکه پروفی‌باس است.

ج) آنرا می‌توان به پورت MPI و DP موجود در CPU متصل نمود.

د) هر سه مورد

۷- در صورت استفاده از PG ...

الف) نیازی به استفاده از PC adapter نمی‌باشد. ب) نیاز به استفاده از PC adapter می‌باشد.

ج) نیاز به استفاده از CP5611 می‌باشد. د) نیاز به استفاده از CP1616 می‌باشد.

فصل ۹

آشنایی با محیط نرم‌افزار Step7

۹-۱ مقدمه

۹-۲ شروع کار با Simatic Manager

۹-۲-۱ استفاده از Wizard جهت ایجاد پروژه

۹-۲-۲ ایجاد پروژه به صورت دستی

۹-۳ محیط پیکربندی سخت‌افزار

۹-۴ محیط برنامه‌نویسی به زبان‌های LAD/FBD/STL

۹-۵ Configure Network

۹-۶ (PLC SIM) Simulator

۹-۷ تنظیمات ارتباطی

۹-۸ منوهای اصلی Simatic Manager

۹-۸-۱ منوی File

۹-۸-۲ منوی Edit

۹-۸-۳ منوی Insert

۹-۸-۴ منوی PLC

۹-۸-۵ منوی View

۹-۸-۶ منوی Options

۹-۸-۷ منوی Window

۹-۸-۸ منوی Help

۹-۸-۹ نوار ابزار

۹-۹ آشنایی با بخش Library و Sample

۹-۱۰ فلوچارت کار با Step7

در این فصل ضمن معرفی منوهای اصلی Step7 نحوه شروع کار با آن بیان شده است.

چکیده مطالب

- پس از نصب Step7 آیکن Simatic Manager روی Desktop در دسترس است که با آن برنامه Step7 اجرا می‌گردد.
- با اجرای Step7 پنجره ویزارد ظاهر می‌شود که توسط آن می‌توان پیکربندی پروژه را انجام داد.
- می‌توان با Cancel کردن ویزارد پروژه را به‌صورت دستی ایجاد نمود.
- پروژه‌ای که با ویزارد ایجاد می‌شود از نظر سخت‌افزاری ممکن است مطابق با سخت‌افزار نصب شده نباشد و نیاز به اصلاح داشته باشد. اگر پروژه به‌صورت دستی ایجاد شود این مشکل وجود ندارد.
- می‌توان در یک پروژه چندین PLC را معرفی کرد.
- همه ابزارهای لازم برای پیکربندی سخت‌افزار، برنامه‌نویسی و پیکربندی شبکه در Simatic Manager وجود دارند.
- برای یک پروژه می‌توان ابتدا سخت‌افزار یا برنامه را وارد کرد.
- اگر سخت‌افزار وارد شود برنامه‌نویسی بر اساس آدرس‌های آن انجام شده و نیاز به تغییرات بعدی در برنامه برای اصلاح آدرس‌ها نیست.

فصل ۹

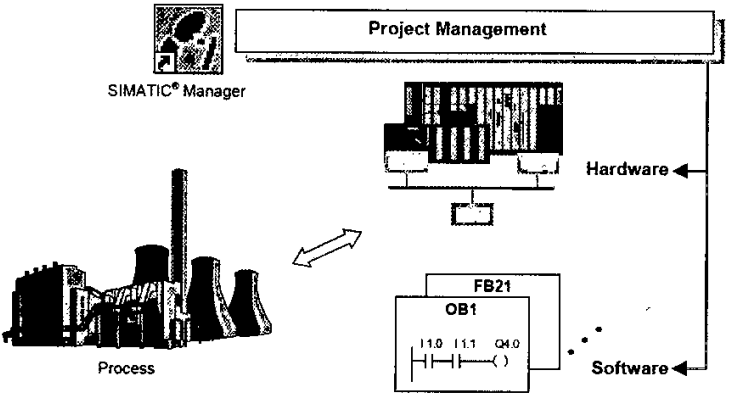
اصطلاحات و تعاریف

Wizard

ابزاری است که توسط آن کاربر می‌تواند به‌سهولت به امکانات نرم‌افزار مورد نظر دسترسی پیدا کند .

۹-۱ مقدمه

برای انجام اتوماسیون یک فرآیند توسط PLC مدل S7-300 یا S7-400، از نرم‌افزار Step7 استفاده می‌کنیم. آیکن این نرم‌افزار پس از نصب با عنوان Simatic Manager روی صفحه Desktop ظاهر می‌شود. با استفاده از این نرم‌افزار پیکربندی سخت‌افرار سیستم کنترل را انجام داده سپس به برنامه‌نویسی می‌پردازیم.



شکل ۹-۱ نقش Simatic Manager برای پیاده‌سازی اتوماسیون فرآیند

فصل ۹

به‌طور خلاصه اهم کارهایی که توسط Simatic Manager برای یک پروژه اتوماسیون بایستی انجام داد عبارتند از:

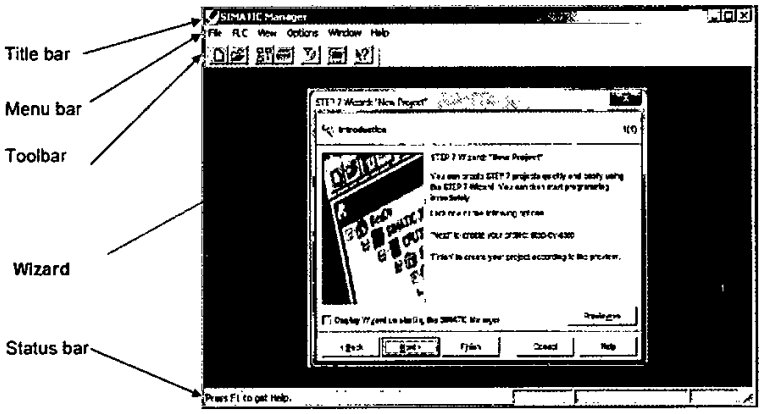
پیکربندی سخت‌افزار: یعنی تعریف تمام اجزای سخت‌افزاری به‌ترتیبی که نصب شده‌اند و انجام تنظیمات آن‌ها به شرحی که در فصل‌های بعدی خواهد آمد.

پیکربندی شبکه: یعنی تعریف شبکه‌های مورد نیاز مانند اترنت یا پروفی‌باس و تنظیمات آن‌ها که این موضوع در کتاب‌های جداگانه تشریح خواهد شد.

برنامه‌نویسی: یعنی پیاده‌سازی منطق کنترل با زبان‌های برنامه‌نویسی موجود در Step7.

۹-۲ شروع کار با Simatic Manager

به‌منظور پیکربندی سخت‌افزاری و تنظیمات آن و انجام برنامه‌نویسی، باید ابتدا نرم‌افزارSimatic Manager را اجرا نمود. آیکن این نرم‌افزار پس از نصب Step7 بر روی دسکتاپ قابل مشاهده می‌باشد. جهت اجرا، روی آن دوبار کلیک نمایید تا نرم‌افزار اجرا شده و کادر Wizard مطابق شکل ۹-۲ نمایش داده می‌شود.



شکل ۹-۲ نمایی کلی از نرم‌افزار Simatic Manager

پروژه

کلیه‌ی اطلاعات مربوط به پیکربندی سخت‌افزاری و برنامه‌ی کاربر تحت عنوان یک پروژه ایجاد و ذخیره می‌گردند. یک پروژه‌ی جدید را می‌توان با استفاده از Wizard یا به‌صورت دستی ایجاد نمود.

۹-۲-۱ استفاده از Wizard جهت ایجاد پروژه

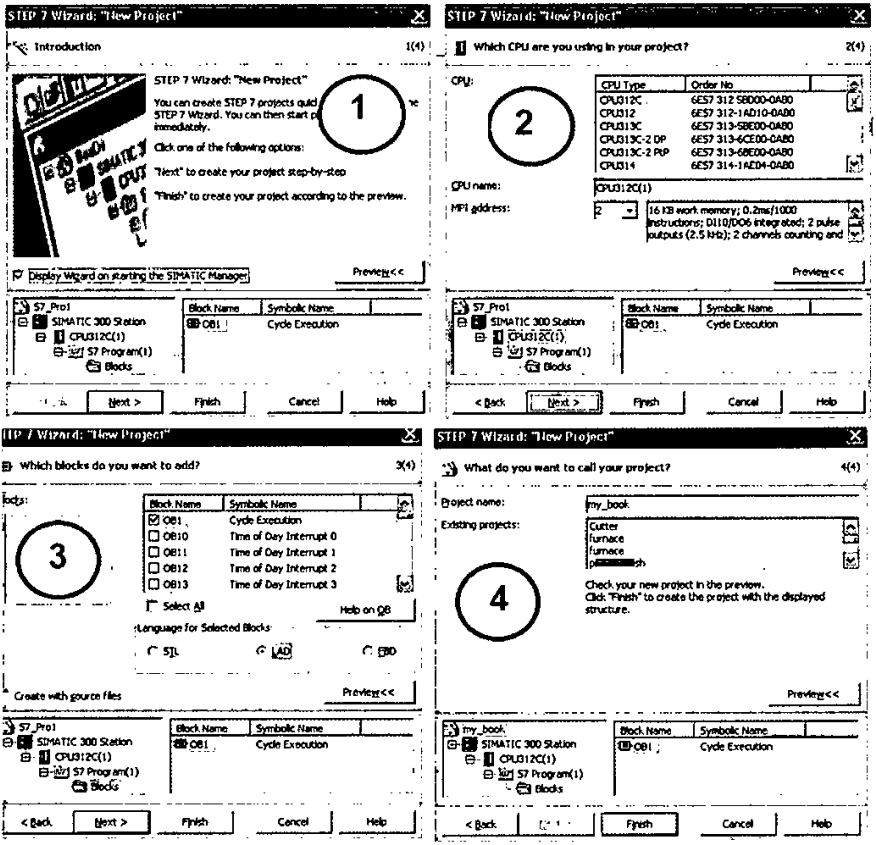
با استفاده از Wizard می‌توان یک پروژه‌ی جدید را به‌صورت سریع با انجام تنظیمات جزئی به‌وجود آورد. ولی ممکن است به‌طور کامل منطبق با ساختار سخت‌افزار مورد نیاز نباشد و لازم باشد بعد از ایجاد پروژه تغییراتی به‌صورت دستی در آن اعمال نمود. مراحل کلی انجام Wizard به‌صورت زیر می‌باشد:

۱. باز شدن کادر اصلی Wizard
۲. انتخاب CPU و آدرس MPI
۳. انتخاب بلاک‌ها و زبان برنامه‌نویسی
۴. انتخاب نام پروژه

لازم به ذکر که است در هر مرحله، جهت عبور به مرحله‌ی بعد باید روی گزینه‌ی Next کلیک نمود.

نکته‌ای که باید توجه نمود این است که با استفاده از Wizard انجام تنظیمات کمتری نسبت به ایجاد پروژه به‌صورت دستی قابل انجام می‌باشد. البته پروژه‌ای که توسط Wizard ایجاد می‌شود را می‌توان به صورت دستی ویرایش نموده و تنظیمات مورد نیاز را انجام داد. در شکل ۹-۳ مراحل ایجاد پروژه توسط Wizard مشاهده می‌گردد.

در مرحله ۱ کار خاصی لازم نیست؛ بنابراین روی Next کلیک می‌کنیم. در مرحله ۲ لیست همه CPUهای 300 و 400 ظاهر می‌شود، CPU مورد نظر را انتخاب کرده به مرحله بعد می‌رویم. در مرحله ۳ بلاک برنامه‌نویسی اصلی که OB1 نام دارد به‌طور پیش‌فرض انتخاب شده است. درصورت لزوم سایر بلاک‌های برنامه‌نویسی را انتخاب می‌کنیم و به مرحله بعد می‌رویم. در مرحله ۴ نام دلخواه را برای پروژه انتخاب می‌نماییم. بدین ترتیب پروژه توسط ویزارد ساخته می‌شود.



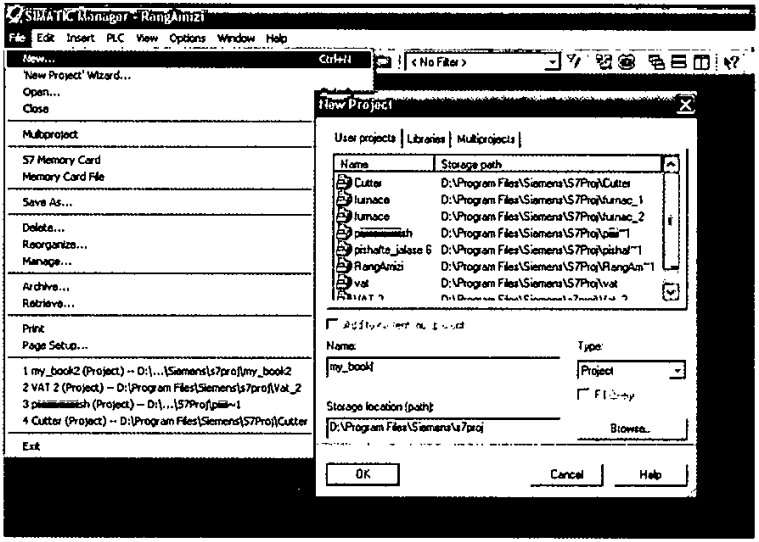
شکل ۹-۳ مراحل چهارگانه ایجاد پروژه توسط Wizard

لازم به ذکر است که اگر گزینه پایین پنجره ویزارد (مرحله ۱ شکل ۹-۳) فعال باشد هر بار که برنامه Simatic Manger اجرا می‌شود ویزارد ظاهر خواهد شد. اگر این گزینه غیر فعال شود با اجرای Simatic Manager آخرین پروژه‌ای که قبل از بستن برنامه در دفعات قبل باز بوده نمایش داده خواهد شد.

۹-۲-۲ ایجاد پروژه به‌صورت دستی

جهت ایجاد یک پروژه به‌صورت دستی، در کادر باز شده‌ی Wizard گزینه‌ی Cancel را انتخاب می‌کنیم. جهت ایجاد یک پروژه‌ی جدید می‌توان از منوی File، گزینه New را انتخاب و یا بر روی آیکن آن در Toolbars کلیک نمود. در

این‌صورت می‌توان در کادر باز شده که در شکل ۹-۴ نشان داده شده است، در قسمت Name، اسم پروژه را نوشته و بر روی گزینه OK کلیک نمود.



شکل ۹-۴ ایجاد پروژه به‌صورت دستی

پس از ایجاد پروژه کارهای زیر باید انجام شود:

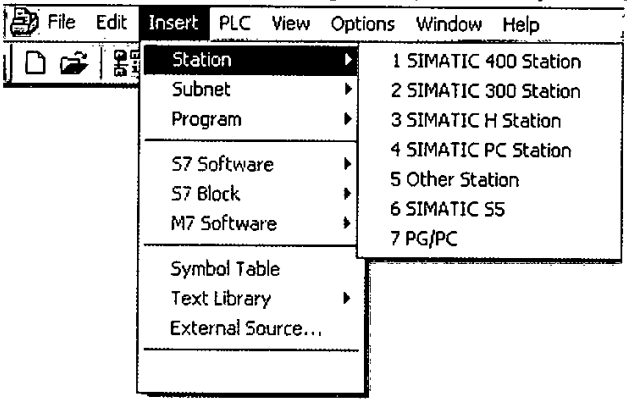
- پیکربندی سخت‌افزاری
- برنامه‌نویسی
- پیکربندی شبکه

هر کدام از موارد فوق را می‌توان به‌عنوان قدم اول پس از ایجاد پروژه تلقی نمود. باید توجه شود که اگر ابتدا پیکربندی سخت‌افزاری صورت پذیرد، آدرس‌های کارت‌های ورودی و خروجی مشخص شده و می‌توان از آنها در برنامه‌نویسی استفاده نمود،اما اگر پیکربندی صورت نگیرد، برنامه با توجه به آدرس‌های فرضی توسط کاربر نوشته شده و پس از مشخص شدن آدرس‌های اصلی، باید آنها را اصلاح نمود. به‌طور کلی می‌توان گفت در مواردی که سخت‌افزار پروژه مشخص است و برنامه‌ی نوشته شده فقط جهت آن پروژه کاربرد دارد، بهتر است ابتدا پیکربندی سخت‌افزار صورت پذیرد و سپس برنامه‌نویسی انجام شود. در مواردی که ممکن است از یک برنامه در پروژه‌های مختلف استفاده شود، می‌توان بدون انتخاب سخت‌افزار، برنامه را نوشته و سپس در هر پروژه با توجه به آدرس کارت‌ها برنامه را اصلاح نمود. در کاربردهای متداول، ابتدا پیکربندی سخت‌افزاری صورت پذیرفته و سپس برنامه نوشته می‌شود.

با توجه به نکات فوق در ادامه ابتدا سخت افزار (Station) را وارد کرده سپس برنامه را ایجاد می‌کنیم.

Station

یک ایستگاه کاری در پروژه می‌باشد که در آن می‌توان یک PLC S7-300 یا PLC S7-400 را پیکربندی و برنامه‌نویسی نمود. در یک پروژه می‌توان چندین ایستگاه کاری داشت که در هر یک از آنها یک PLC تعریف شده باشد. همه‌ی Stationهای موجود در یک پروژه را می‌توان از طریق شبکه‌های صنعتی به هم متصل نمود. جهت قرار دادن Station از منوی Insert گزینه‌ی Station را انتخاب می‌کنیم.



شکل ۹-۵ گزینه‌های موجود در زیرمنوی Station

فصل ۹

همانطور که در شکل ۹-۵ مشخص است، در منوی Station گزینه‌های زیر قابل انتخاب هستند:

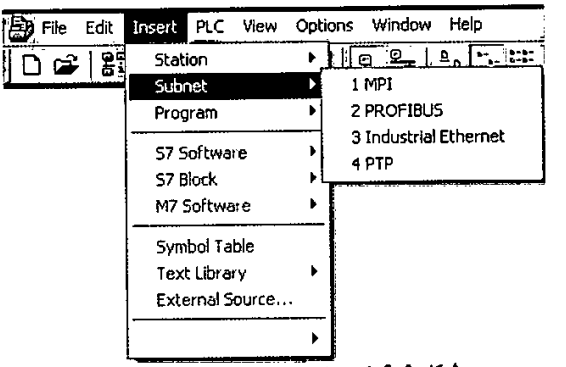
- Simatic 400 Station : یک ایستگاه کاری S7-400 ایجاد می‌نماید.
- Simatic 300 Station: یک ایستگاه کاری S7-300 ایجاد می‌نماید.
- Simatic H Station: به‌منظور پیکربندی و برنامه‌نویسی PLCهای S7-400H استفاده می‌شود. البته برای این منظور باید پکیج H-system نیز نصب شده باشد.
- Simatic PC Station: به‌منظور پیکربندی کارت‌هایی که روی کامپیوتر نصب می‌گردند و تنظیمات کامپیوتر وقتی از OPC یا سیستم PC based استفاده می‌شود به‌کار می‌رود.
- Other Station: برای سایر محصولاتی که ساخت شرکت زیمنس یا هر شرکت دیگری هستند به‌کار می‌رود. به‌عنوان مثال اگر PLC آلن‌بردلی با S7 شبکه باشد می‌توان آن را به‌عنوان یک Other Station معرفی کرد.
- Simatic S5: می‌تواند یک ارتباط S5 زیمنس را با PLC S7 تعریف نماید، ولی بایستی توجه کرد که تنظیمات S5 یا برنامه‌نویسی آن توسط Step7 امکان پذیر نیست.
- PG/PC: به‌منظور معرفی ارتباط PC یا PG که برای امور مهندسی استفاده می‌شود به کار می‌رود. لازم به ذکر است که به‌طور معمول برای انجام عملیات دانلود و آپلود و ارتباط با PLC هیچ نیازی به تعریف این Station نیست. فقط در برخی کاربردهای شبکه این کار ضرورت پیدا می‌کند که در کتاب شبکه تشریح خواهد شد.

در منوی Station مطابق با توضیحات بیان شده، یکی از گزینه‌های فوق‌الذکر انتخاب می‌گردد.

Subnet

از طریق این زیرمنو می‌توان یک شبکه صنعتی را وارد پروژه کرد. اگر در این مرحله شبکه وارد نشود مشکلی نخواهد بود، در پنجره‌های بعدی نرم‌افزار به‌سادگی می‌توان این کار را بعداً انجام داده و وارد نمود. شبکه‌های قابل انتخاب عبارتند از:

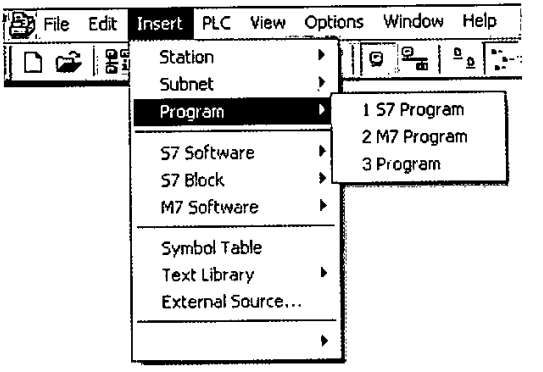
- MPI: برای شبکه‌بندی بین PLCهای S7 استفاده می‌شود.
- Profibus: شبکه‌ای استاندارد است که می‌توان وسایل مختلف از زیمنس یا سایر سازندگان را به آن متصل کرد.
- Industrial Ethernet: شبکه‌ی استاندارد که به‌طور عمده برای شبکه‌بندی PLC (S7 و غیر S7) به‌کار می‌رود. وسایل شبکه Profinet نیز به اترنت صنعتی متصل می‌شوند.
- PTP: شبکه استاندارد برای ارتباطاتی نظیر مدباس است.



شکل ۹-۶ گزینه‌های موجود در زیرمنوی Subnet

Program

توسط این زیر منو می‌توان بدون پیکربندی سخت‌افزاری، بلاک‌های مختلف برنامه‌نویسی را ایجاد نمود. همانطور که قبلاً ذکر شد این روش توصیه نمی‌شود زیرا هنوز آدرس‌های سخت‌افزار مشخص نشده و منجر به ضرورت بازنگری برنامه در مراحل بعد برای اصلاح آدرس‌ها خواهد شد.



شکل ۹-۷ گزینه‌های موجود در زیرمنوی Program

در این زیر منو، گزینه‌های زیر قابل انتخاب می باشند:

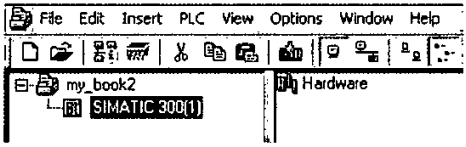
S7-Program: به منظور قرار دادن بلاک‌های برنامه‌نویسی جهت PLCهای S7-300 و S7-400.

M7-Program: به منظور برنامه‌نویسی کنترل‌کننده‌های M7.

معمولا برنامه اصلی بایستی در زیر Station وارد شود. استفاده از S7 Program به‌عنوان شاخه اصلی پروژه بیشتر برای ذخیره‌سازی Backup برنامه به‌کار می‌رود.

۹-۳ محیط پیکربندی سخت‌افزار

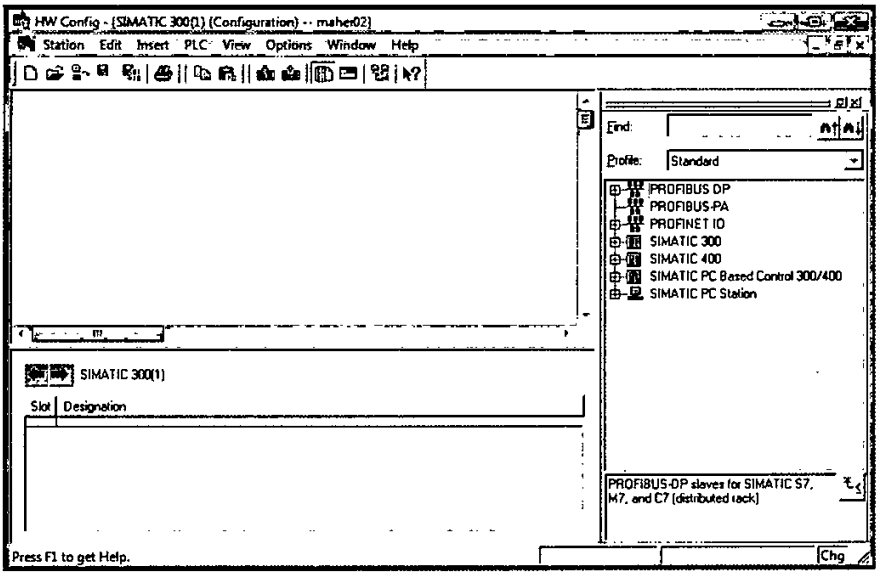
پس از ایجاد Station مورد نظر، می‌توان مطابق شکل۹-۸ روی گزینه‌ی Hardware کلیک نمود.



شکل ۹-۸ اجرای محیط پیکربندی سخت‌افزاری

در این‌صورت محیط پیکربندی سخت‌افزاری[1] ظاهر می‌گردد.

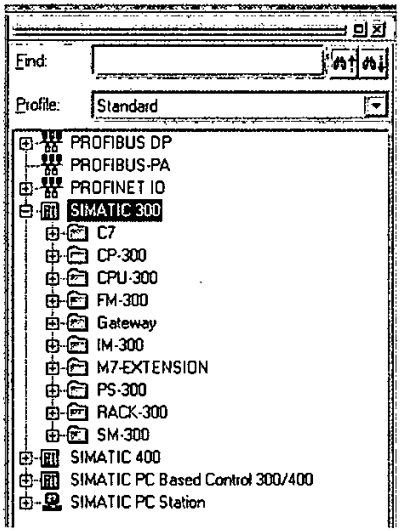
این محیط، در شکل ۹-۹ نشان داده شده است. اجزای مورد نیاز در بخش سفید رنگ سمت چپ وارد می‌شود.



شکل ۹-۹ محیط HW Config

1. H.W config: Hardware configuration

در قسمت سمت راست برنامه، کاتالوگ سخت‌افزارهای مختلف قرار دارد. در هر پیکربندی سخت‌افزاری، قطعات مورد نیاز PLC از طریق این کاتالوگ انتخاب می‌شوند.



شکل۹-۱۰ کاتالوگ محصولات در نرم‌افزار HW Config

همانگونه که مشاهده می‌گردد، در این کاتالوگ محصولات مختلفی به شرح ذیل قرار دارند:

- Profibus DP
- Profibus PA
- Profinet IO
- Simatic 300
- Simatic 400
- Simatic PC Based control 300/400
- Simatic PC Station

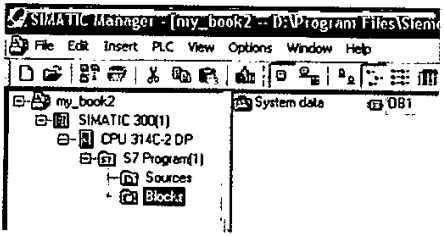
با کلیک روی علامت + کنار هر کدام از بخش‌های فوق، می‌توان به جزئیات بیشتری از آن بخش دست یافت و در نهایت با انتخاب قسمت‌های مختلف، پیکربندی را انجام داد.

اگر وسیله مورد نظر جدید باشد و در کاتالوگ یافت نشود، لازم است کاتالوگ Update شود. روش کار در فصل بعد تشریح شده است.

۹-۴ محیط برنامه‌نویسی به زبان‌های LAD/FBD/STL

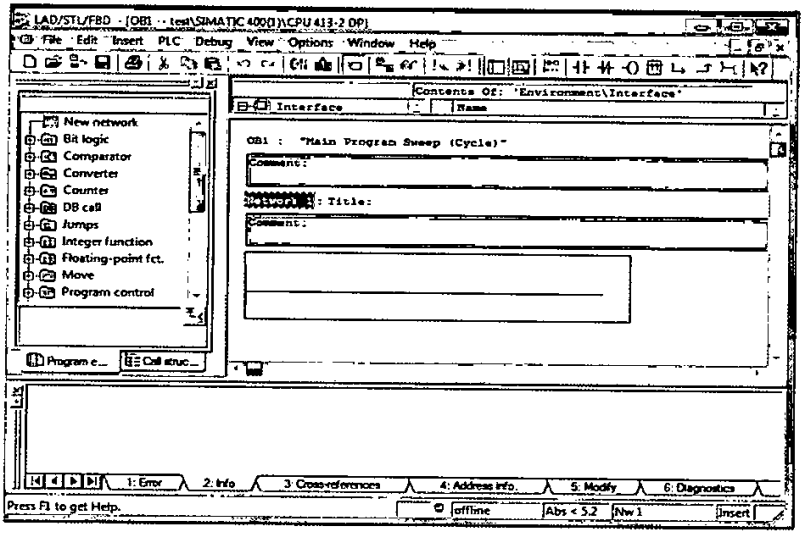
وقتی پیکربندی سخت‌افزاری انجام می‌پذیرد، می‌توان در ساختار درختی پروژه، در سمت چپ محیط Simatic manager که در شکل ۹-۱۱ نشان داده شده است، بلاک برنامه‌نویسی را مشاهده کرد. به‌طور پیش‌فرض OB1 که

بلاک اصلی برنامه‌نویسی PLC است در این مسیر ساخته می‌شود. در عین حال می‌توان آنرا به‌صورت دستی با کلیک راست موس ایجاد نمود.



شکل۹-۱۱ روش ایجاد OB1

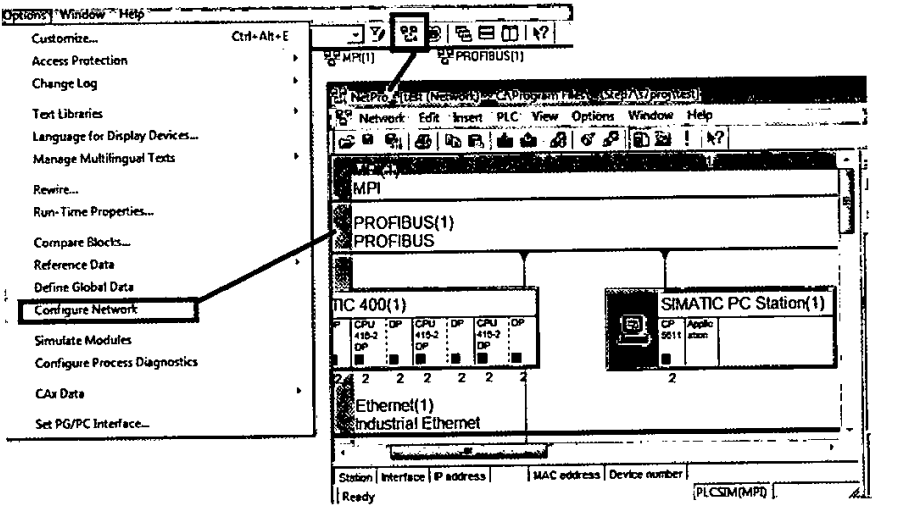
با دوبار کلیک کردن روی OB1، این محیط توسط برنامه LAD/STL/FBD باز می‌شود. برنامه‌ی کاربر در این محیط نوشته می‌شود. در فصل ۱۴ کار با این محیط کاملاً تشریح می‌گردد.



شکل۹-۱۲ محیط برنامه‌نویسی

۹-۵ Configure Network

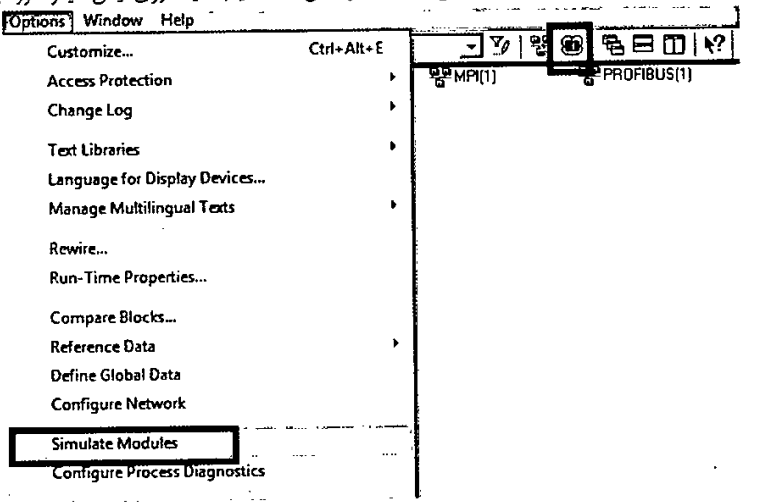
با استفاده از منوی Option و انتخاب گزینه Configure Network یا با استفاده از آیکن نشان داده شده در شکل ۹-۱۳ برنامه NetPro باز می‌شود. از این برنامه برای پیکربندی شبکه‌های صنعتی پروفی‌باس و اترنت و مدباس استفاده می‌گردد. توضیحات مفصل در کتاب‌های شبکه آمده است.



شکل ۹-۱۳ مسیر بازکردن برنامه NetPro

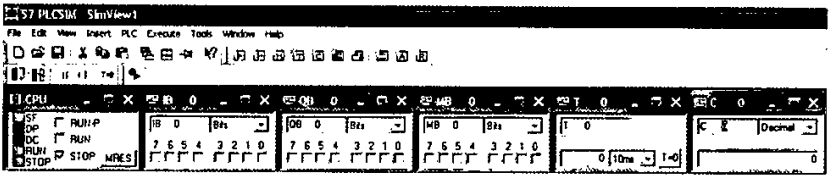
۹-۶ (PLC SIM) Simulator

همانگونه که قبلاً ذکر گردید، Simulator یک شبیه‌ساز نرم‌افزاری می‌باشد که قادر است عملکرد CPU واقعی را شبیه‌سازی کند. این شبیه‌ساز را می‌توان از مسیر نشان داده شده در شکل ۹-۱۴ یا با کلیک روی آیکن سیمولاتور اجرا نمود.



شکل ۹-۱۴ مسیر فراخوانی Simulator

پس از اجرای سیمولاتور محیط نرم‌افزار PLC Sim به‌صورت نشان داده شده در شکل ۹-۱۴ نمایان می‌گردد.



شکل ۹-۱۵ محیط نرم‌افزار PLCSim

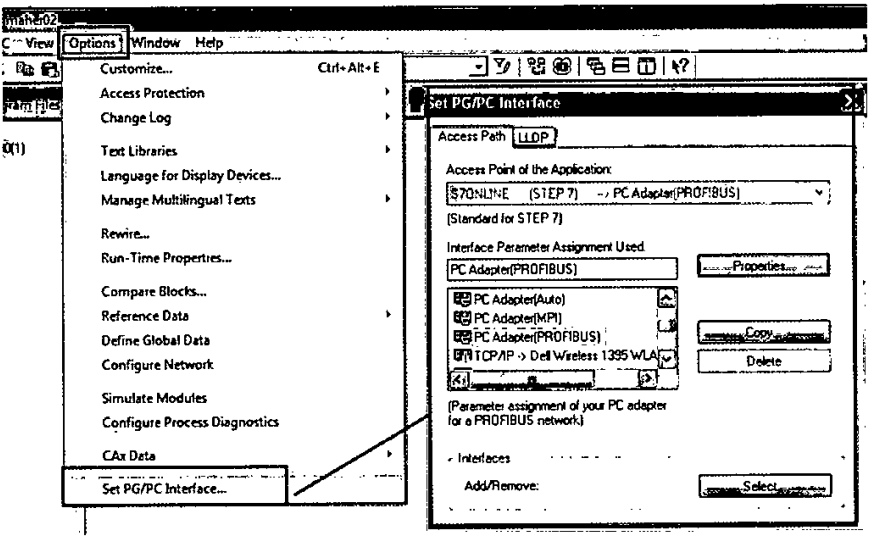
ماژول CPU همواره به‌طور پیش‌فرض در محیط این نرم‌افزار وجود دارد. CPU دارای سه وضعیت کاری STOP، RUNو RUN-P می‌باشد. همچنین دکمه‌ی MRES به منظور انجام عمل ریست قرار داده شده است. جهت شبیه‌سازی نمودن ورودی‌ها/ خروجی‌ها/ حافظه‌های داخلی/ کانتر و تایمر می‌توان توسط آیکن‌های نشان داده شده در شکل ۹-۱۶ نسبت به قرار دادن ماژول‌های فوق عمل نمود.

شکل ۹-۱۶ آیکن‌های ایجاد ورودی/خروجی/ مموری/ تایمر/کانتر و . . .در سیمولاتور

فصل ۹

۹-۷ تنظیمات ارتباطی

هنگامی که پیکربندی سخت‌افزاری صورت گرفت یا برنامه‌ی کاربر در کامپیوتر یا PG نوشته شد، باید برنامه‌ی نوشته شده و تنظیمات انجام شده به PLC واقعی انتقال پیدا نماید (دانلود شود). اما قبل از اینکه بتوان برنامه را از کامپیوتر به PLC دانلود نمود، باید تنظیمات ارتباطی بین آنها صورت پذیرد. برای این منظور می‌توان از کنترل پنل ویندوز به شرحی که در فصل قبل ذکر شد استفاده کرد. در عین حال در خود نرم افزار Simatic Manager از مسیر Option→Set PG/PC Interface نیز این کار قابل انجام است.



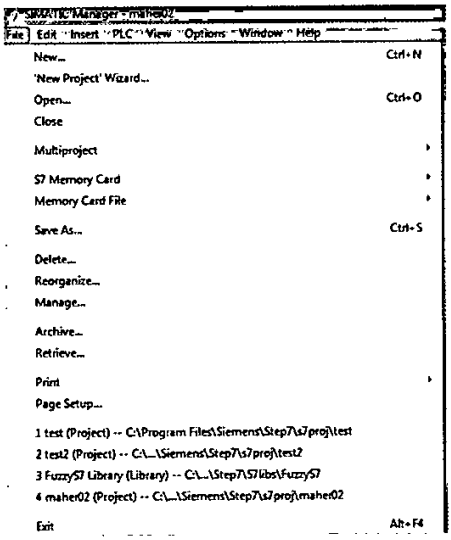
شکل ۹-۱۷ پنجره Set PG/PC Interface

۹-۸ منوهای اصلی Simatic Manager

در محیط نرم‌افزار Simatic Manager منوهایی به شرح ذیل قرار دارند. تشریح تمام این منوها در این فصل که هنوز خواننده با برخی مفاهیم اصلی آشنا نشده امکان‌پذیر نیست و صرفاً به‌عنوان آنها اشاره می‌گردد:

- File: در این منو دستوراتی جهت انجام روی پروژه وجود دارد.
- Edit: دستوراتی جهت انجام اعمال ویرایشی نظیر Copy، Cut، Paste، Properties و غیره در این قسمت وجود دارند.
- Insert: جهت قرار دادن Station، Subnet و بلاک‌ها استفاده می‌شود.
- PLC: جهت مواردی از قبیل ارسال و دریافت اطلاعات به PLC و از آن استفاده می‌شود.
- View: نحوه‌ی نمایش آیکن‌ها و اعمالی نظیر فیلترگذاری بر روی آنها در این بخش قابل تنظیم است.
- Options: برخی از دستورات و قابلیت‌هایی که به‌صورت اختیاری، کاربر می‌تواند از آنها استفاده کند در این قسمت قرار دارند.
- Window: در این قسمت می‌توان تنظیماتی جهت نمایش پنجره‌ها در Simatic Manger انجام داد.
- Help: یک راهنمای نرم‌افزاری جهت استفاده از نرم‌افزار Simatic Manager به زبان لاتین می‌باشد.

۹-۸-۱ منوی File



شکل۹-۱۸ زیرمنوهای منوی File

منوهای اصلی در منوی File عبارتند از:

فصل ۹

New

جهت ایجاد یک پروژه جدید استفاده می‌گردد.

New Project Wizard

جهت ایجاد یک پروژه جدید با استفاده از Wizard

Open

جهت باز نمودن پروژه‌ای که قبلاً ایجاد و ذخیره شده است.

Close

جهت بستن پروژه‌ای که باز است.

Multiproject

وقتی لازم است چندین Station در یک پروژه تعریف شود یا باز گردد استفاده می‌شود.

S7 Memory Card

وقتی لازم باشد کارت حافظه با PG یا کارت خوان زیمنس (USB Prommer) باز شود از این منو استفاده می‌شود.

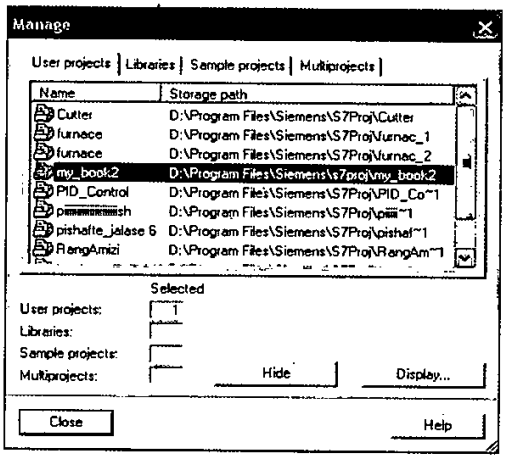
Memory Card File
برای سیستم‌های WinLC استفاده می‌گردد.

Save As
جهت ذخیره‌سازی پروژه با نام دیگر استفاده می‌شود.

Delete
جهت حذف نمودن کامل یک پروژه استفاده می‌گردد.

Reorganize
دیتابیس و فایل‌های مربوط به پروژه را سازماندهی مجدد می‌کند و فضاهای به‌وجود آمده ناشی از پاک‌کردن Objectها را حذف می‌کند. با انجام این عمل حافظه مورد نیاز جهت ذخیره‌سازی پروژه کاهش می‌یابد.

Manage
توسط آن می‌توان یک پروژه را مخفی ساخت یا یک پروژه را که قبلاً مخفی شده بود به حالت قابل نمایش درآورد. اگر گزینه Manage انتخاب گردد، کادر نشان داده شده در شکل ۹-۱۹ نمایش داده می‌شود.



شکل ۹-۱۹ کادر Manage

در این کادر می‌توان پروژه‌ی مورد نظر را انتخاب و توسط گزینه‌ی Hide آنرا مخفی نمود. همچنین توسط گزینه‌ی Display می‌توان یک پروژه را از حالت مخفی بودن خارج ساخت.

Archive
توسط این گزینه می‌توان کل پروژه کاربر شامل اطلاعات سخت‌افزار، شبکه و برنامه را به‌صورت یک فایل فشرده آرشیوسازی نمود. فایل فشرده شده را می‌توان روی هارد کامپیوتر یا CD ذخیره نمود تا در صورت نیاز به‌عنوان پشتیبان از

آن استفاده گردد. این فایل به‌صورت پیش‌فرض دارای فرمت Zip می‌باشد، ولی از مسیر Option > Customize می‌توان فرمت آنرا عوض نمود.

Retrieve

این گزینه عکس عمل آرشیوسازی را انجام می‌دهد و توسط آن می‌توان پروژه‌ای را که به‌صورت فایل آرشیو ذخیره شده است، از حالت آرشیو خارج نموده و محتویات آنرا مشاهده نمود.

Print

برای چاپ کردن اطلاعات موجود در پروژه استفاده می‌شود.

شکل۹-۲۰ محتویات زیرمنوی Print

اگر گزینه Object List انتخاب شود، محتویات پنجره Simatic Manager چاپ می‌گردد. همچنین می‌توان در ساختار درختی پروژه، روی سخت‌افزار یا بلاک‌های برنامه‌نویسی کلیک نموده و از زیرمنوی Print گزینه Object Contents را انتخاب نمود. در این صورت محتویات قسمت انتخاب شده چاپ می‌گردد.

فصل ۹

Page Setup

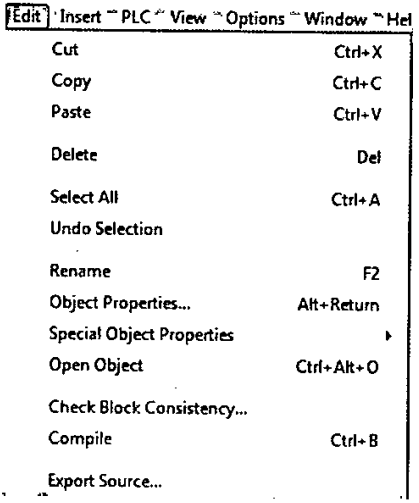
این گزینه به‌منظور انجام تنظیمات بر روی کاغذ چاپ به کار می‌رود.

Exit

جهت خروج از نرم‌افزار Simatic Manager استفاده می‌گردد.

۹-۸-۲ منوی Edit

در این منو امکاناتی مانند Copy / Paste / Cut و تنظیم برخی ویژگی‌ها وجود دارد که نیاز به توضیح ندارند. گزینه Properties برای مشاهده یا تغییر تنظیم Objectهایی مانند بلاک‌ها به‌کار می‌رود و گزینه Special Object Properties برای تنظیمات خاص مانند تنظیمات مرتبط با winCC کاربرد دارد که در کتاب‌های دیگر مورد بحث قرار می‌گیرد.

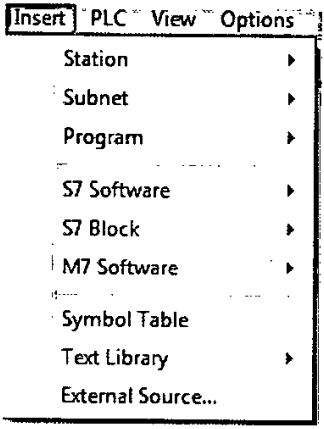


شکل۹-۲۱ محتویات منوی Edit

۹-۸-۳ منوی Insert

همانطور که قبلاً ذکر شد از طریق این منو می‌توان Objectهای مختلف نظیر Station یا S7 Program یا Subnet و ... را به پروژه وارد کرد.

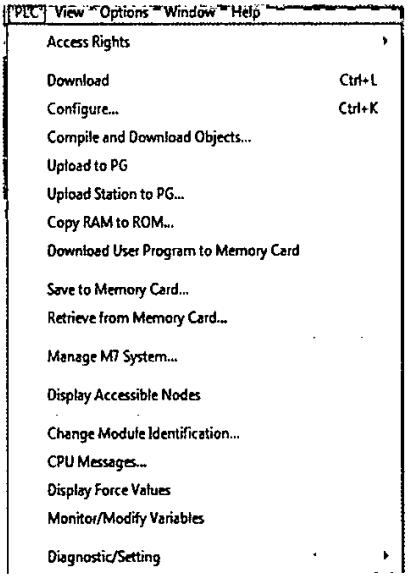
لازم به ذکر است که بدون نیاز به این منو در محیط Simatic Manager نیز می‌توان در هر نقطه با کلیک راست موس Object مورد نظر را وارد نمود.



شکل۹-۲۲ محتویات منوی Insert

۹-۸-۴ منوی PLC

در این منو موارد بسیاری که فعلاً تشریح آنها امکان پذیر نیست دیده می‌شود. برخی موارد صرفاً معرفی می‌شوند:



شکل۹-۲۳ محتویات منوی PLC

فصل ۹

Access Right

وقتی PLC مجهز به پسورد باشد از طریق این منو می‌توان با وارد کردن پسورد برای دفعات بعد آنرا غیر فعال نمود.

Download

برای انتقال بخشی یا تمام اطلاعات از کامپیوتر به PLC استفاده می‌شود.

Compile and Download Objects

تمام اطلاعات را ابتدا کامپایل سپس دانلود می‌کند.

Upload Station to PG

اطلاعات از PLC به کامپیوتر منتقل می‌شود.

Copy RAM to ROM

برای CPUهایی که ROM داخلی دارند محتویات RAM را به آن کپی می‌کند.

Download user program to Memory Card
برای انتقال برنامه به کارت فلش نصب شده روی CPU به‌کار می‌رود.

Save to Memory Card
کل پروژه را به‌صورت زیپ شده به کارت حافظه منتقل می‌کند (Backup).

Retrieve From Memory Card
عکس عمل قبلی است؛ یعنی پروژه زیپ شده روی کارت حافظه را باز کرده به کامپیوتر منتقل می‌کند.

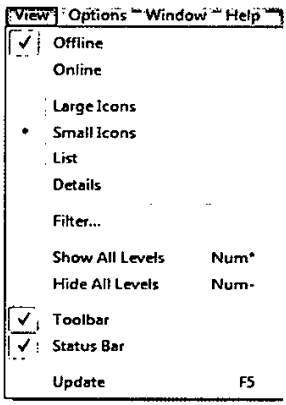
Display Accessible Nodes
PLC هایی که از طریق کامپیوترِ دیده می‌شوند را نشان می‌دهد.

Display Force Value
برای Force کردن ورودی و خروجی‌ها به‌کار می‌رود.

Monitor/Modify Value
برای مشاهده آدرس‌های حافظه PLC یا اختصاص مقدار به آنها به‌کار می‌رود.

Diagnostic/Setting
برای عیب‌یابی یا استارت و استپ CPU و تنظیمات دیگر کاربرد دارد.

۹-۸-۵ منوی View



شکل ۹-۲۴ زیر منوهای منوی View

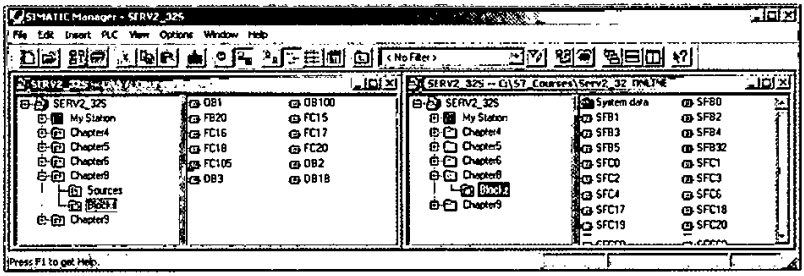
گزینه‌های موجود در این منو عبارتند از:

Offline

مشاهده‌ی پروژه‌ی ذخیره شده در کامپیوتر در حالت عدم اتصال به PLC.

Online

مشاهده‌ی پروژه‌ی درون PLC در حالت اتصال به PLC،همانطور که در شکل ۹-۲۵ نشان داده شده است.



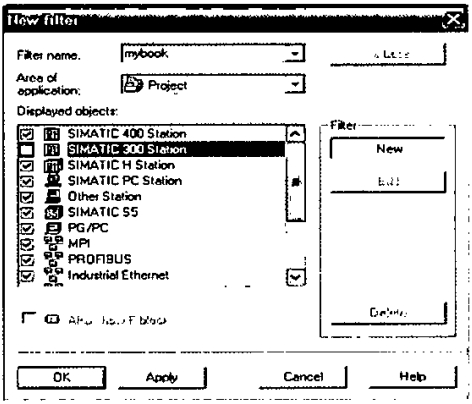
شکل ۹-۲۵ منوی View در حالت Online و Offline

Details - List - Small Icon – Large Icons

نحوه‌ی نمایش آیکن‌ها به‌صورت: بزرگ، کوچک، لیست شده، همراه با جزئیات.

Filter

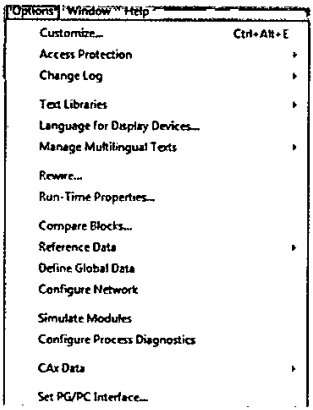
با انتخاب فیلتر می‌توان روی Objectهای مختلف موجود در پروژه فیلتر گذاشت تا نمایش داده نشوند. همانطور که در شکل ۹-۲۶ مشاهده می‌گردد، در قسمت Filter Name یک اسم دلخواه را وارد نموده و جهت عدم نمایش قسمت‌های دلخواه علامت √ را از کنار آنها حذف می‌کنیم. مثلاً در شکل ۹-۲۶ Simatic 300 Station در پنجره سمت راست درختی نمایش داده نخواهد شد.



شکل۹-۲۶ نمایی از پنجره فیلتر

۹-۸-۶ منوی Options

برخی از این زیرمنوها عبارتند از:



شکل ۹-۲۷ منوی Options

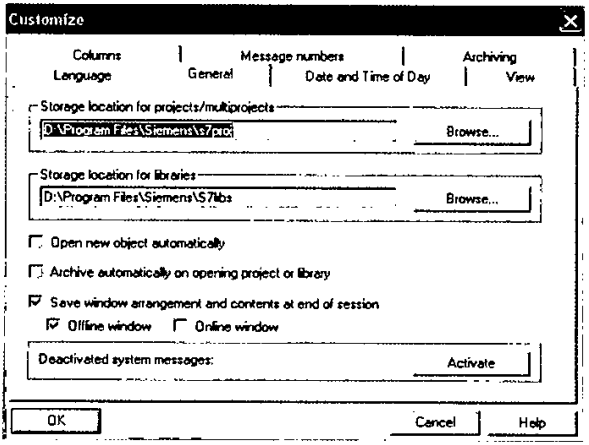
Customize

در این قسمت کاربر می‌تواند تنظیمات مختلف مربوط به پروژه را از حالت پیش‌فرض خارج نموده و تغییرات مورد نظر خود را اعمال نماید. این قسمت خود دارای پنجره‌های زیر می‌باشد:

General

برای انتخاب محل ذخیره‌سازی پروژه و موارد دیگر نظیر باز نمودن اتوماتیک یک پروژه‌ی جدید، آرشیوسازی اتوماتیک، ذخیره‌سازی ترتیب قرارگیری

اجزاء در صفحه و فعال نمودن پیام‌های غیرفعال شده توسط گزینه Do not Display this Message



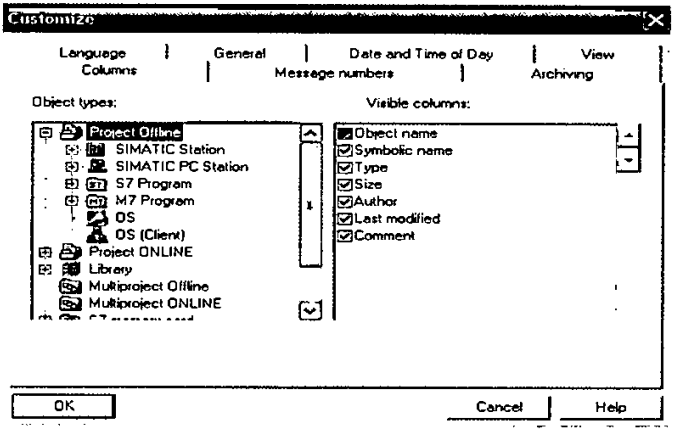
شکل ۹-۲۸ پنجره General از زیرمنوی Customize

View

به منظور اعمال تنظیمات جهت نمایش Online، مانند رنگ زمینه، نمایش مد کاری در سمبول ماژول و ... به‌کار می‌رود.

Columns

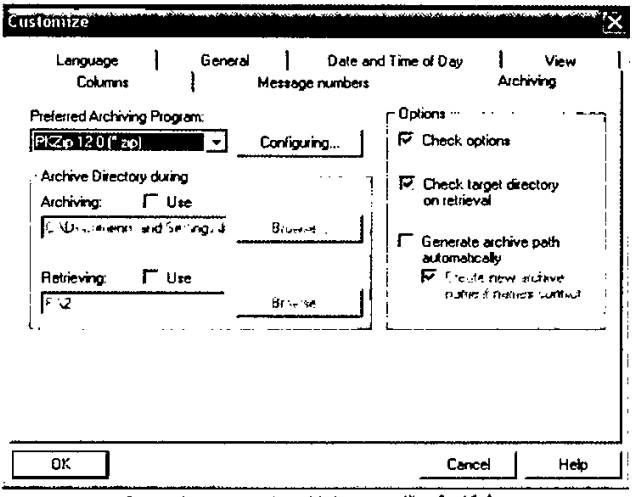
جهت تنظیم نمایش اطلاعات هر Object به‌کار می‌رود.



شکل ۹-۲۹ پنجره Columns از زیرمنوی Customize

Archiving

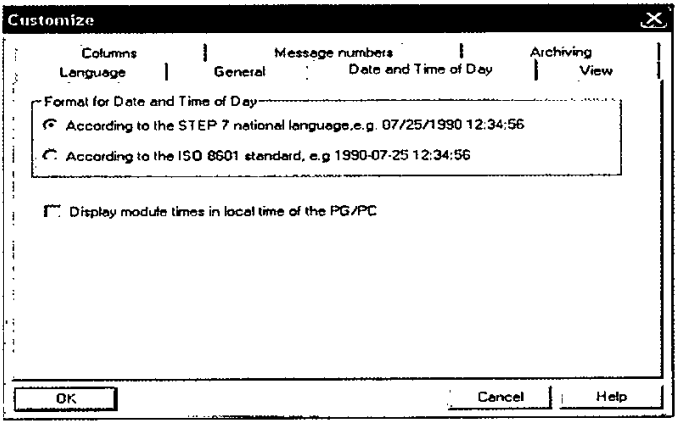
انتخاب فرمت فایل آرشیوشده توسط گزینه File > Archive، در این قسمت امکان‌پذیر می‌باشد.



شکل ۹-۳۰ پنجره Archiving از زیرمنوی Customize

Date and Time of Day

در این قسمت تعیین فرمت تاریخ و زمان، تنظیم می‌گردد.



شکل۹-۳۱ پنجره Date and Time of Day از زیرمنوی Customize

Message numbers

نحوه‌ی تعیین تعداد پیام‌ها برای یک پروژه‌ی جدید در این قسمت انجام می‌پذیرد.

۹-۸-۷ منوی Window

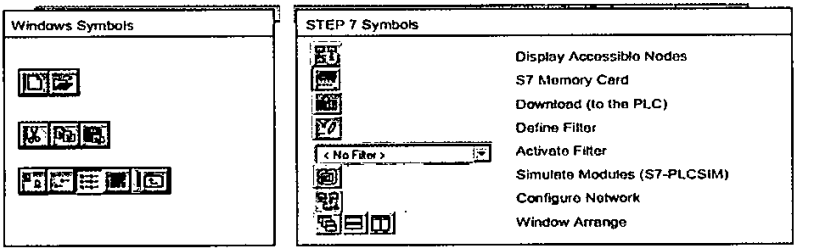
برای تنظیم پنجره نمایش یک یا چند پروژه به‌طور همزمان استفاده می‌شود.

۹-۸-۸ منوی Help

امکانات راهنمای Step7 در این منو قرار دارد.

۹-۸-۹ نوار ابزار

در نوار ابزار بالای پنجره نیز آیکن‌های زیر وجود دارد که به‌جای استفاده از گزینه‌های منو می‌توان از آنها استفاده نمود. این آیکن‌ها برخی مربوط به ویندوز و برخی مربوط به Step7 هستند.



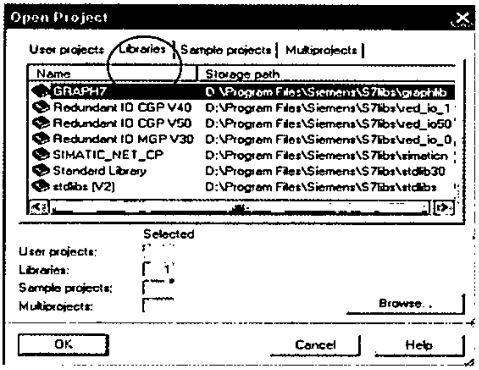
شکل۹-۳۲ آیکن‌های نوار ابزار Step7

تذکر : سایر گزینه‌های منوها یا آیکن‌هایی که‌در اینجا اشاره نشدنددرفصل‌های‌بعدی‌متناسب با کاربرد آنها تشریح می‌گردند.

۹-۹ آشنایی با بخش Library و Sample

Libraries

در نرم‌افزار Simatic Manager تعدادی بلاک آماده و از قبل برنامه‌نویسی شده به منظور انجام وظایف خاص قرار داده شده است، که کاربر می‌تواند از آنها در برنامه‌ی خویش استفاده نماید. این بلاک‌ها در قسمتی به نام Libraries قرار گرفته‌اند. جهت دستیابی به این بخش می‌توان از منوی File گزینه Open را انتخاب نموده و در پنجره نشان داده شده در شکل ۹-۳۳ گزینه‌ی Libraries را انتخاب نمود.

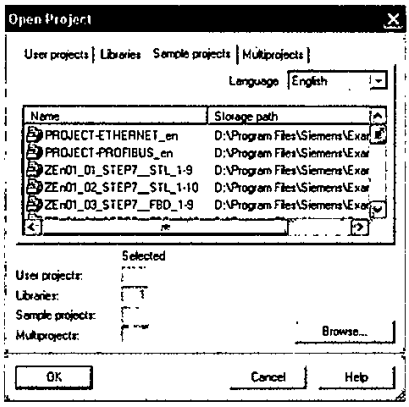


شکل ۹-۳۳ مسیر انتخاب Sample projects

فصل ۹

Sample projects

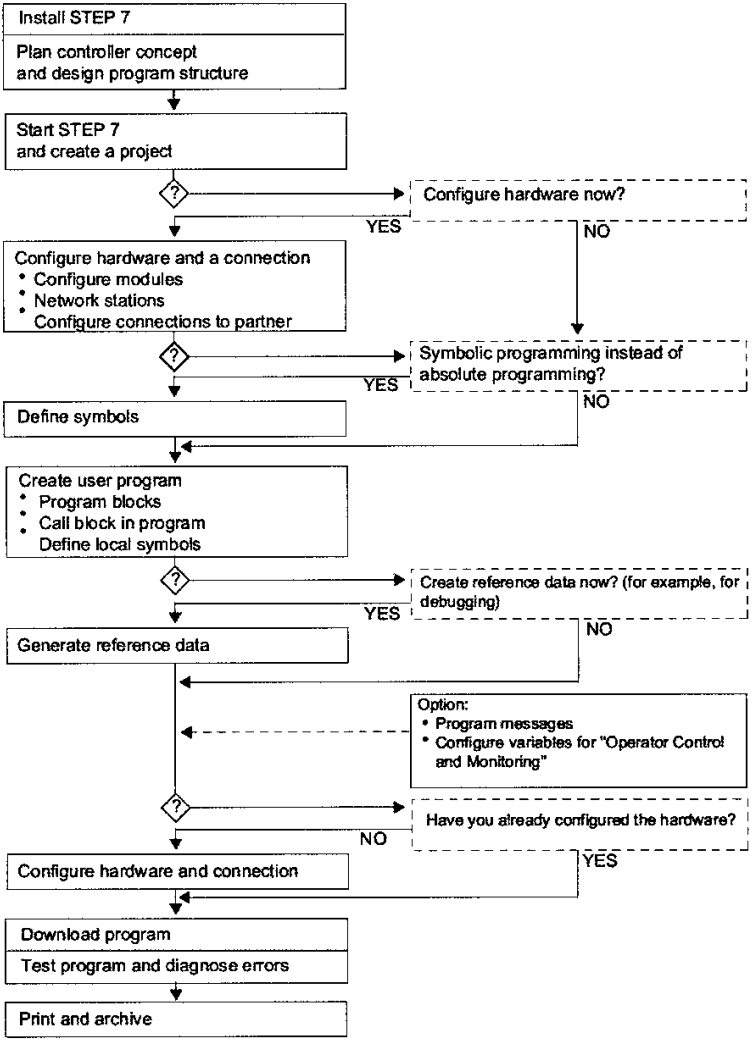
در نرم‌افزار Simatic Manager تعدادی پروژه آماده و برنامه‌نویسی شده به‌عنوان نمونه قرار داده شده است. کاربر می‌تواند ضمن اجرای آنها با نکات مربوط به پیکربندی سخت‌افزاری، برنامه‌نویسی و تنظیمات شبکه در یک پروژه‌ی صنعتی آشنا گردد. جهت دستیابی به این بخش می‌توان از منوی File گزینه Open را انتخاب نموده و در پنجره نشان داده شده در شکل ۹-۳۳ گزینه‌ی Sample projects را انتخاب نمود.



شکل ۹-۳۴ مسیر انتخاب Sample projects

۹-۱۰ فلوچارت کار با Step7

فلوچارت زیر مراحل کار با Step7 را نشان می‌دهد. جزئیات این مراحل در فصل‌های بعدی خواهد آمد.



شکل ۹-۳۵ مراحل کار با Step7

فصل ۱۰

آشنایی با محیط پیکربندی سخت‌افزار (HwConfig)

۱۰-۱ مقدمه

۱۰-۲ آشنایی با محیط HWconfig

۱۰-۳ منوهای HWconfig

۱۰-۳-۱ منوی Station

۱۰-۳-۲ منوی Edit

۱۰-۳-۳ منوی Insert

۱۰-۳-۴ منوی PLC

۱۰-۳-۵ منوی View

۱۰-۳-۶ منوی Options

۱۰-۴ آشنایی با کاتالوگ

۱۰-۴-۱ نحوه وارد کردن اجزا

۱۰-۴-۲ آشنایی با اجزای سخت افزاری کاتالوگ

۱۰-۴-۳ پروفایل‌های کاتالوگ

۱۰-۴-۴ Update نمودن کاتالوگ در H.Wconfig

۱۰-۵ پرسش و تحقیق

در این فصل محیط نرم‌افزار HwConfig که برای پیکربندی اجزای سخت افزار کاربرد دارد، معرفی شده است. مطالعه این بخش قبل از شروع برنامه‌نویسی ضروری است.

چکیده مطالب

- Hwconfig نرم‌افزاری از زیر مجموعه‌های Step7 است که توسط آن پیکربندی سخت‌افزار انجام می‌شود.
- در این محیط نه تنها پیکربندی سخت‌افرار PLCهای S7-300 و S7-400 قابل انجام است، بلکه اجزای دیگر نظیر کامپیوترهایی که برای مانیتورینگ یا OPC به‌کار می‌روند را نیز می‌توان پیکربندی نمود.
- در محیط Simatic Manager با انتخاب Station مورد نظر و باز کردن آن می‌توان وارد محیط Hw config شد.
- در محیط Hw config کاتالوگی وجود دارد که تمام اجزای PLC و PC و شبکه‌های صنعتی مانند Profibus و Profinet در آن موجود هستند.
- در صورتی که نرم‌افزار بروز نباشد ممکن است لازم باشد کاتالوگ Update شود تا اجزای سخت‌افزاری جدید به آن اضافه گردند.

فصل ۱۰

اصطلاحات و تعاریف

GSD

فایل درایور اجزای Profibus است که با نصب آن تجهیزات مورد نظر وارد کاتالوگ برنامه می‌شوند.

۱۰-۱ مقدمه

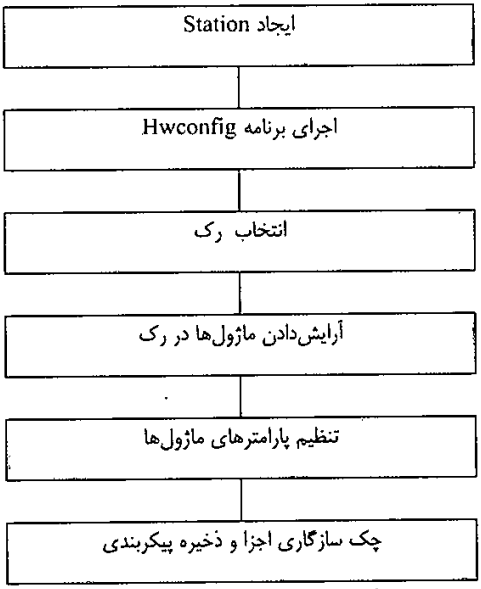
همانطور که در فصل قبل اشاره شد، ابزاری که نرم‌افزار Step7 برای پیکربندی سخت‌افزار به‌کار می‌برد Hwconfig نام دارد. با استفاده از آن می‌توان انواع PLCهای S7 را پیکربندی نمود. در PLCهای قدیمی مانند S5 پیکربندی سخت‌افزار از طریق نرم‌افزار انجام نمی‌شد بلکه با استفاده از دیپ‌سوئیچ‌ها تنظیمات سخت‌افزاری صورت می‌گرفت، ولی در PLC های جدید این تنظیمات با نرم‌افزار صورت می‌گیرد و پس از دانلود CPU می‌تواند اجزای تعریف شده و تنظیمات آنها را بشناسد. شایان ذکر است توسط Hwconfig در برخی موارد لازم است PC Station نیز پیکربندی شود. به‌عنوان مثال تعریف کارت شبکه‌ای که روی PC نصب شده و اختصاص آدرس به آن از جمله این تنظیمات است.

به‌طور کلی وقتی از پیکربندی سخت‌افزار PLC صحبت می‌کنیم منظور موارد زیر است:

- تعیین چیدمان اجزای PLC طبق سخت‌افزار نصب شده
- آدرس‌دهی به کارت‌ها برای استفاده در برنامه‌نویسی
- انجام تنظیمات برای برخی کارت‌ها که دارای دیپ‌سوئیچ هستند به‌منظور تطابق تنظیمات در برنامه با کارت نصب شده
- فعال کردن یا غیر فعال کردن برخی از قابلیت‌های کارت‌ها

غالباً اگر چیدمان کارت‌ها در نرم‌افزار متفاوت با چیدمان آنها در نرم‌افزار باشد، یا بین برخی تنظیمات سخت‌افزاری در عمل و در نرم‌افزار تفاوت وجود داشته باشد، CPU با فالت مواجه خواهد شد.

به‌طور کلی مراحلی که ابتدا برای پیکربندی سخت‌افزار انجام می‌شود در شکل ۱۰-۱ نشان داده شده است.

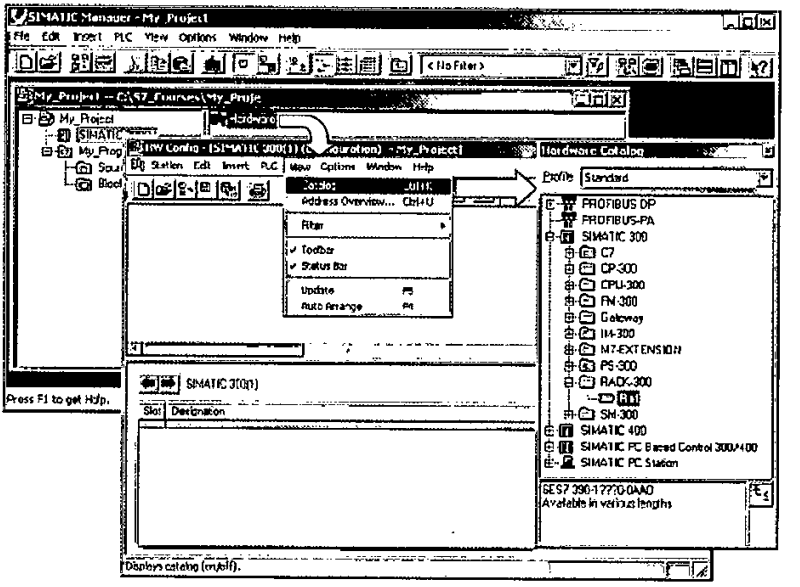


شکل ۱۰-۱ مراحل پیکربندی سخت افزار

فصل ۱۰

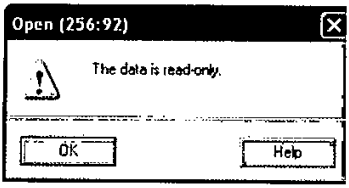
۱۰-۲ آشنایی با محیط HW Config

برنامه HW Config از داخل Simatic Manager فراخوان می‌شود و نیازی نیست که به‌طور مستقل اجرا شود. فرض کنید که یک Station 300 در محیط پروژه در Simatic Manager وارد کرده باشیم. اگر روی آن کلیک کنیم در روبروی آن آیکن Hardwarc مانند شکل ۱۰-۲ ظاهر می‌شود که با دوبارکلیک نمودن روی آن پنجره HW Config باز می‌گردد.



شکل ۱۰-۲ محیط Hwconfig

در برخی حالات وقتی پنجره HW Config باز می‌شود با پیغام read only مواجه می‌شویم؛ یعنی امکان تغییر در پیکربندی نیست و فقط می‌توان وضعیت آنها را دید. این موضوع نشان دهنده آن است که اطلاعات سخت‌افزار در محیط دیگری همزمان باز هستند. به‌عنوان مثال اگر برنامه Netpro که برای پیکربندی شبکه است باز باشد و در آن تغییراتی داده شده و هنوز ذخیره نشده باشد، با شرایط read only برای HW Config مواجه خواهیم شد.

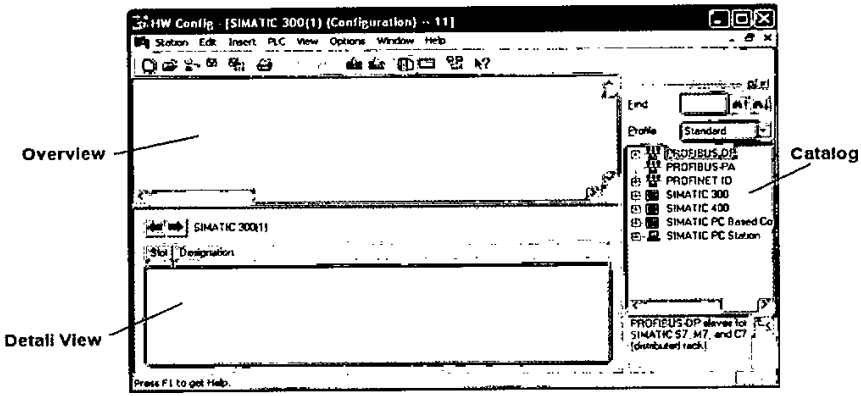


شکل ۱۰-۳ پیغام خطا ناشی از تداخل کار HW Config و Netpro

پس از باز شدن پنجره مشاهده می‌کنیم که این پنجره دارای سه بخش اصلی زیر است:

- بخشOverview: در این قسمت سخت‌افزار وارد شده به‌صورت کلی و بدون اطلاعات جزئی نشان داده می‌شود.
- بخش Detail View : در این قسمت سخت‌افزار وارد شده با اطلاعات دقیق و جزئی‌تر نشان داده می‌شود.
- بخش Catalog : کاتالوگ حاوی اجزای مختلف مورد نیاز پروژه است.

بخش کاتالوگ ممکن است غیرفعال شده و نمایش داده نشود ولی دو بخش دیگر همواره موجود هستند.

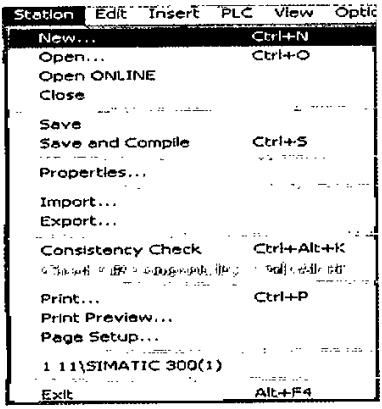


شکل ۱۰-۴ بخش‌های مختلف پنجره HW Config

۱۰-۳ منوهای HW Config

۱۰-۳-۱ منوی Station

زیرمنوهایی که در این منو وجود دارند در شکل ۱۰-۵ نشان داده شده است.

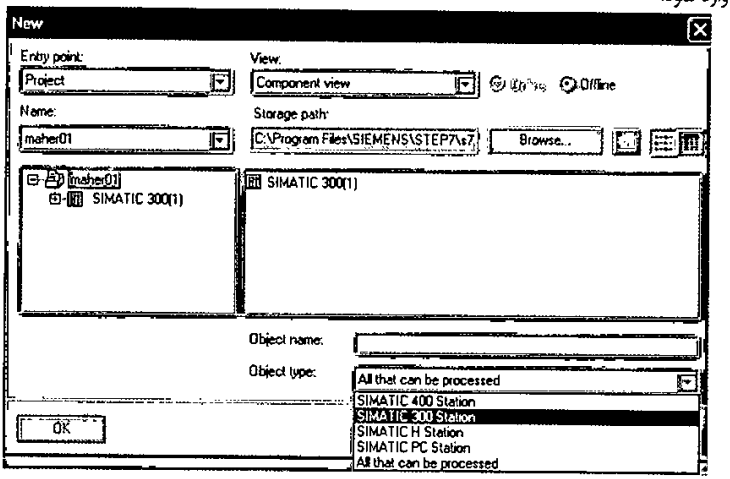


شکل ۱۰-۵ منوی Station

فصل ۱۰

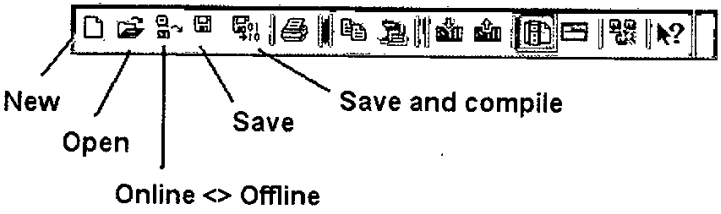
New : برای ایجاد یک پیکربندی سخت‌افزار جدید به‌کار می‌رود. بطور معمول نیاز نیست که از این مسیر سخت‌افزار تعریف شود بلکه همانطور که ذکر شد در محیط Simatic Manager بهتر است Station مورد نظر را وارد کرده و باز کنیم.

با این وجود با کلیک روی New پنجره‌ای به شکل ۱۰-۶ باز می‌شود که می‌توان Station مورد نظر را با نام دلخواه به پروژه وارد نمود.



شکل ۱۰-۶ استفاده از زیرمنوی New برای ایجاد پیکربندی جدید

Open : برای باز کردن سخت‌افزاری که قبلاً پیکربندی شده است به‌کار می‌رود. این گزینه نیز معمولاً مورد نیاز نیست چون غالباً سخت‌افزار از محیط Simatic Manager با دوبار کلیک نمودن روی آیکن Hardware باز می‌شود.
به‌جای استفاده از Open می‌توان روی آیکن نشان داده شده در شکل ۱۰-۷ کلیک کرد.



شکل ۱۰-۷ آیکن‌های مرتبط با منوی Station در نوار ابزار

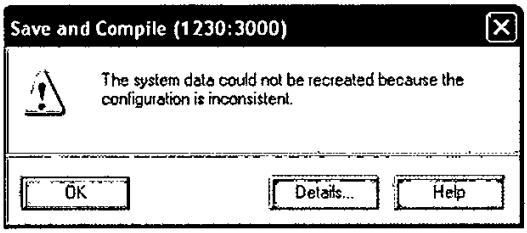
Open Online

با انتخاب این گزینه در صورتی‌که کامپیوتر به PLC متصل باشد پنجره سخت‌افزار به‌صورت Online ظاهر می‌شود. پنجره Online بیشتر برای مشاهده وضعیت و عیب‌یابی سخت‌افزار به‌کار می‌رود که بعداً تشریح خواهد شد. به‌جای استفاده از این گزینه می‌توان روی آیکن نشان داده شده در شکل ۱۰-۷ کلیک کرد.

Save : توسط این گزینه یا با استفاده از آیکن آن در منوبار که در شکل ۱۰-۷ نشان داده شده، اطلاعات پیکربندی سخت‌افزار ذخیره می‌گردد.

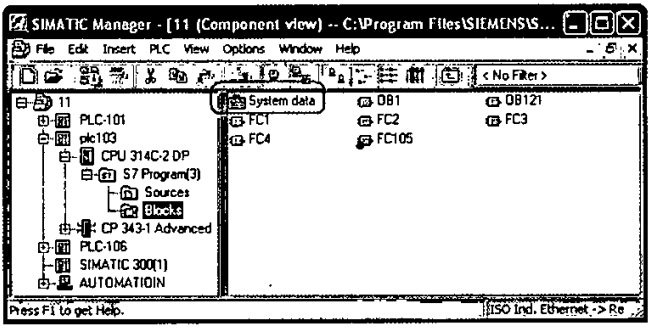
Save and Compile توسط این گزینه یا با استفاده از آیکن منوبار که در شکل ۱۰-۷ نشان داده شده است، اطلاعات پیکربندی سخت‌افزار ذخیره و کامپایل می‌گردد.

تفاوت آن با Save در این است که پس از کامپایل اگر خطایی در پیکربندی سخت‌افزار وجود داشته باشد گزارش داده می‌شود. شکل ۱۰-۸ نمونه‌ای از پیکربندی که با پیغام خطا در هنگام کامپایل مواجه شده است را نشان می‌دهد.



شکل ۱۰-۸ گزارش وجود خطا در پیکر بندی پس از ذخیره و کامپایل

اگر Save and Compile موفقیت‌آمیز باشد پس از بازگشت به Simatic Manager می‌بینیم که آیکن System Data در فولدر Blocks ساخته شده است، ولی با Save به تنهایی این آیکن ساخته نمی‌شود.

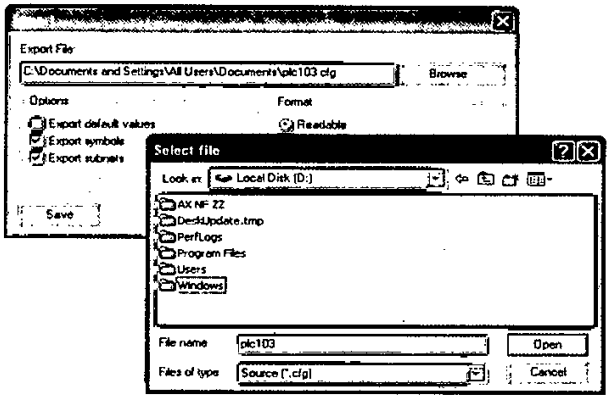


شکل ۱۰-۹ ایجاد System data پس از ذخیره و کامپایل بدون خطا

Properties : ویژگی‌های کلی سخت‌افزار مانند نام Station و ارتباطات شبکه و ... را نشان می‌دهد.

Export : اطلاعات سخت‌افزار را به‌صورت یک فایل با پسوند Cfg ذخیره می‌کند. لازم به ذکر است که به‌طور معمول اطلاعات سخت‌افزار در پروژه Siamtic Manager همراه با سایر اطلاعات برنامه‌نویسی و شبکه ذخیره می‌شود. در

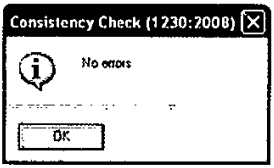
صورتی که نیاز باشد که صرفاً اطلاعات سخت‌افزار در یک فایل ذخیره شود تا بتوان آنرا در پروژه‌های دیگر استفاده نمود، از این گزینه استفاده می‌گردد.



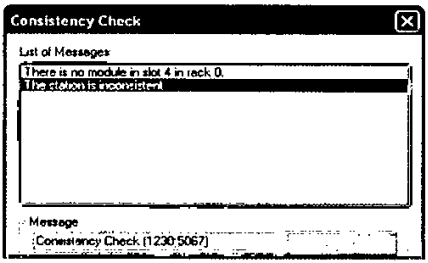
شکل ۱۰-۱۰ انتقال نتیجه پیکربندی سخت افزار به فایل cfg

Import : عکس عمل Export را انجام می‌دهد، یعنی اطلاعات سخت‌افزاری موجود در یک فایل cfg را به محیط HW Config وارد می‌کند.

Consistency Check : چک سازگاری اجزای سخت‌افزاری را انجام و نتیجه را گزارش می‌دهد. دو گزارش نمونه در شکل ۱۰-۱۱ نشان داده شده است. گزارشی که در موقع بروز خطا توسط این گزینه نمایش داده می‌شود همان گزارش detail مربوط به Save and Compile می‌باشد.



سازگاری بدون مشکل

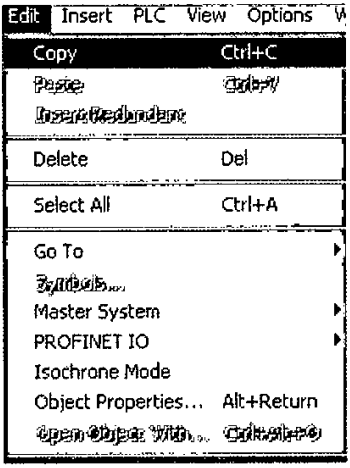


اشکال در سازگاری

شکل ۱۰-۱۱ چک سازگاری در دو حالت مختلف

۱۰-۳-۲ منوی Edit

از طریق این منو علاوه بر امکانات متداول نظیر Copy / Paste / Delete ، برخی امکانات خاص نیز وجود دارد که به برخی از آنها اشاره می‌شود.



شکل ۱۰-۱۲ منوی Edit

Insert Redundant: این گزینه فقط برای سیستم‌هایی که به‌صورت نرم‌افزاری در آنها افزونگی پیاده‌سازی می‌شود کاربرد دارد. توضیحات افزونگی در کتاب جداگانه‌ای خواهد آمد.

GoTo: وقتی اجزای سخت‌افزاری موجود در HWconfig زیاد باشند از طریق Goto می‌توان آنها را سریعتر پیدا کرد.

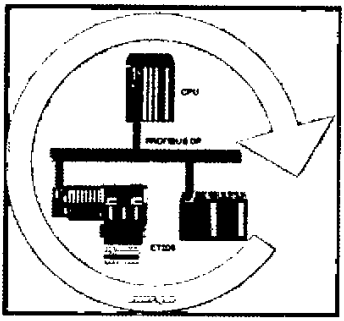
Symbols: این گزینه برای کارت‌های I/O فعال می‌شود و توسط آن می‌توان به آدرس کانال‌ها اسامی با مفهوم اختصاص داد.

Master System: این گزینه برای شبکه پروفی‌باس کاربرد دارد.

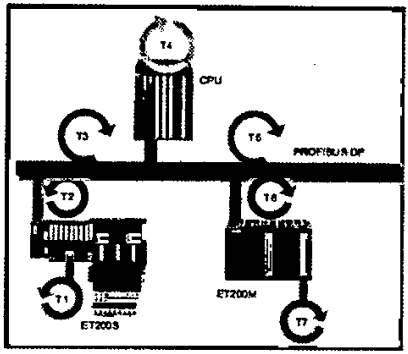
rofinet I/O : برای پیکربندی شبکه پروفی‌نت کاربرد دارد.

Isochrone Mode: یک مد کاری خاص برای برخی اجزای سخت‌افزاری است. اجزایی که دارای این قابلیت هستند می‌توانند با هم سنکرون (هم‌زمان) شوند و اختلاف زمانی بین آنها به حداقل برسد. به‌عنوان مثال ممکن است یک ورودی با سرعت زیاد در کارت آنالوگ تبدیل شود ولی ارسال آن به CPU از طریق شبکه یا به‌طور مستقیم تأخیر داشته باشد. در نوع Isochrone این تأخیرها از بین می‌رود و زمان یکسان و مناسبی برای همه اجزا منظور می‌گردد.

فصل ۱۰



با قابلیت Isochrone

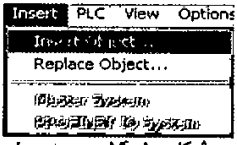


بدون قابلیت Isochrone

شکل ۱۰-۱۳ قابلیت Isochrone در اجزای سخت‌افزاری

۱۰-۳-۳ منوی Insert

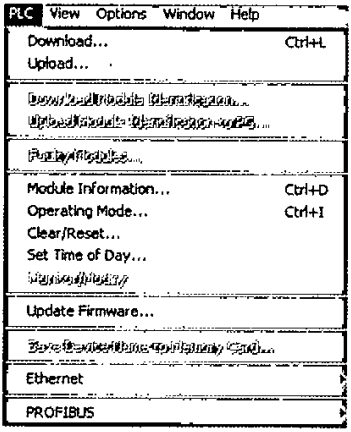
با استفاده از گزینه‌های موجود در این منو می‌توان اجزای سخت‌افزاری را وارد نمود یا اجزای وارد شده قبلی را با نوع دیگر جایگزین نمود.



شکل ۱۰-۱۴ منوی Insert

۱۰-۳-۴ منوی PLC

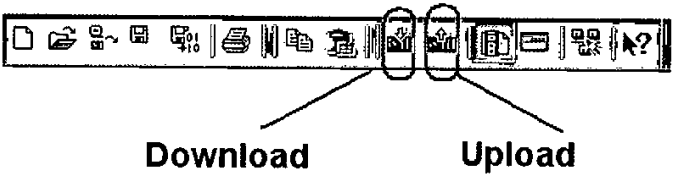
یکی از مهمترین منوهای HW Config برای کارهای Online است.مفاهیم‌برخی‌ازگزینه‌های‌آن‌دراین‌مرحله برای خواننده کاملاً روشن نخواهد شد مگر آنکه قسمت‌های بعدی را مطالعه و بحث را کامل کند. مهمترین گزینه‌های این منو عبارتند از:



شکل ۱۰-۱۵ منوی PLC

Download: توسط این گزینه یا از طریق آیکن شکل ۱۰-۱۶ می‌توان اطلاعات سخت‌افزار را به PLC دانلود نمود، البته به‌شرط اینکه از نظر سازگاری[1] مشکلی نداشته باشد. به‌همین دلیل توصیه می‌شود همواره قبل از دانلود، اطلاعات سخت‌افزاری را Save and Compile کرده تا از عدم وجود خطا مطمئن شوید.

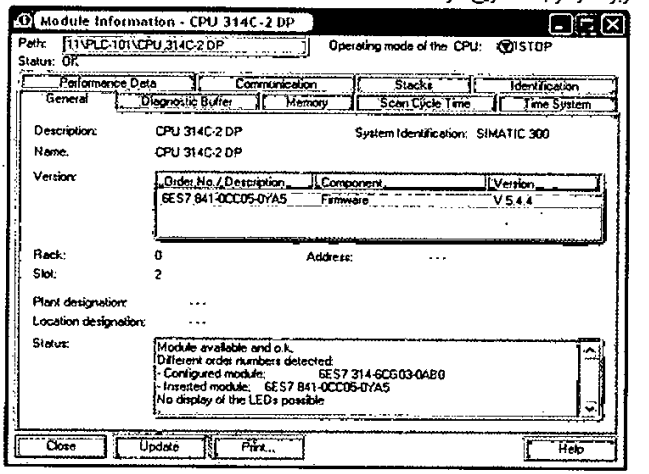
در قسمت‌های بعد توضیح بیشتری در مورد رفتار دانلود داده خواهد شد. در اینجا صرفاً اشاره می‌کنیم که دانلود سخت‌افزار به PLCای که در حال کار است منجر به توقف آن برای لحظاتی خواهد شد. زیرا CPU سخت‌افزار را فقط در لحظه راه‌اندازی چک می‌کند و لازم است یک‌بار Stop و مجدداً Run شود.



شکل ۱۰-۱۶ آیکن‌های دانلود و آپلود در نوار ابزار

Upload : برخلاف Download عمل می‌کند، یعنی اطلاعات سخت‌افزاری که قبلاً به PLC دانلود شده را از CPU به کامپیوتر منتقل کرده و در محیط HW Config جای می‌دهد. برای اینکه عمل انجام پذیرد باید پنجره HW Config خالی و قبلاً اطلاعاتی در آن وارد نشده باشد.

Module Information: اگر ابتدا سخت‌افزار مورد نظر را در پنجره انتخاب کرده سپس روی این گزینه کلیک کنیم در حالت Online وضعیت ماژول را خواهیم دید. شکل ۱۰-۱۷ این گزینه را برای یک CPU نشان می‌دهد. این گزینه برای عیب‌یابی بسیار کاربرد دارد و بعداً تشریح خواهد شد.



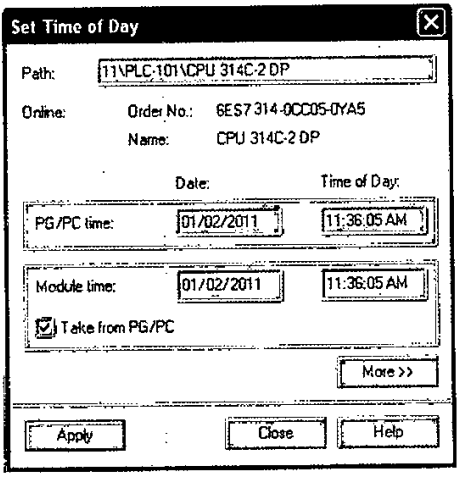
شکل ۱۰-۱۷ پنجره Module Information

1. Consistency

Operating Mode: توسط این گزینه می‌توان مدکاری PLC را تغییر داد، مثلاً آنرا متوقف کرد یا به راه انداخت. مدهای کاری PLC در فصل ۱۲ آمده است.

Clear Reset : توسط این گزینه می‌توان حافظه CPU را ریست نمود. توضیح اینکه حافظه چه بخش‌هایی دارد و کدام قسمت‌ها پاک می‌شوند در فصل ۱۲ آمده است.

Set Time of Day: توسط این گزینه می‌توان تاریخ و زمان CPU را تنظیم نمود. با کلیک روی آن، پنجره نشان داده شده در شکل ۱۰-۱۸ ظاهر می‌شود. در قسمت PG/PC time تاریخ کامپیوتر و در قسمت Module time تاریخ PLC نمایش داده می‌شود. با کلیک روی گزینه Apply می‌توان تاریخ و زمان کامپیوتر را به PLC منتقل نمود.



شکل ۱۰-۱۸ پنجره تنظیم تاریخ و زمان CPU

اگر گزینه Take From PG/PC غیر فعال شود، می‌توان تاریخ و زمان دلخواه را برای CPU تنظیم نمود.

Monitor / Modify

این گزینه برای کارت‌های ورودی و خروجی کاربرد دارد، توسط آن می‌توان وضعیت سیگنال‌ها را مشاهده کرد یا مقداری را به آنها تحمیل نمود. توضیحات بیشتر را در فصل ۲۴ ببینید.

Update Firmware

توسط این گزینه می‌توان Version مربوط به CPU را تغییر داد. نحوه تغییر ورژن در کتاب پیشرفته آمده است.

Ethernet

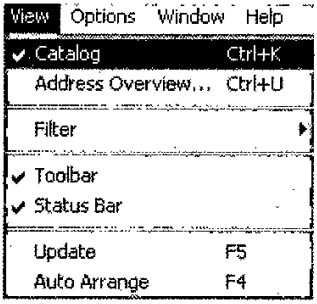
برای شناسایی و عیب‌یابی وسایلی که متصل به شبکه اترنت صنعتی هستند به‌کار می‌رود.

Profibus

برای شناسایی و عیب‌یابی وسایلی که به شبکه پروفی‌باس متصل هستند به‌کار می‌رود.

۱۰-۳-۵ منوی View

توسط امکانات موجود در منوی View می‌توان نحوه نمایش پنجره‌ها را در HW Config تغییر داد. گزینه‌های مهم آن عبارتند از:



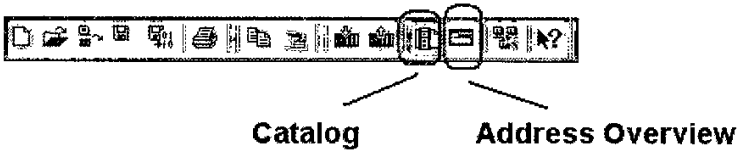
شکل ۱۰-۱۹ منوی view

Catalog

از طریق این گزینه می‌توان پنجره کاتالوگ را برای نمایش فعال یا غیرفعال نمود. این کار از طریق آیکن نشان داده شده در شکل ۱۰-۲۰ نیز امکان‌پذیر است. بخش کاتالوگ در ادامه تشریح شده است.

Address Overview

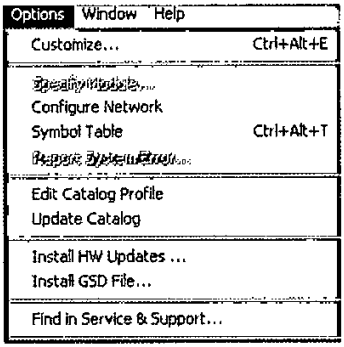
پس از اتمام پیکربندی سخت‌افزار می‌توان آدرس‌های اختصاص داده شده به کارت‌ها را از طریق این گزینه یا آیکن نشان داده شده در شکل ۱۰-۲۰ مشاهده کرد.توضیحات لازم را در فصل بعد در بحث آدرس دهی ببینید.



شکل ۱۰-۲۰ برخی از آیکن‌های منوی View در نوارابزار

۱۰-۳-۶ منوی Options

منوی Options امکانات انتخابی را لیست می‌کند که برخی از آنها با بیرون از محیط HW Config ارتباط دارد. مهمترین گزینه‌ها عبارتند از:



شکل ۱۰-۲۱ منوی Options

Customize: برای تغییر برخی تنظیمات محیط HWConfig به‌کار می‌رود.

Configure Network: با کلیک روی این گزینه یا آیکن نشان داده شده در شکل ۱۰-۲۲، برنامه Netpro که مخصوص پیکربندی شبکه است اجرا خواهد شد.

شکل ۱۰-۲۲ آیکن Netpro در نوار ابزار

بحث Netpro در کتاب مرجع کاربردی شبکه پروفی‌باس تألیف محمدرضا ماهر به‌طور کامل تشریح شده است.

Symbole Table: این گزینه جدول سمبل‌ها که بیرون از محیط HW Config است را باز می‌کند. در جدول سمب ها نه تنها اسامی اختصاص داده شده به ورودی و خروجی بلکه نام سایر متغیرهای حافظه و بلاک‌ها نمایش داده می‌شود.

Edit Cataloge Profile: کاتالوگ را می‌توان به‌طور دلخواه تغییر داد. به‌عنوان مثال کاربر می‌تواند یک کاتالوگ دلخواه برای خود ایجاد کند که فقط اجزای مورد نظرش در آن قرار داشته باشند. تشریح این مطلب در ادامه در بحث کاتالوگ آمده است.

Update Catalog: برای شبکه پروفی‌باس در برخی شرایط ممکن است با وجود اینکه درایور وسیله پروفی‌باس نصب شده ولی برنامه HW Config نتواند آنرا شناسایی کند و اطلاعات مورد نظر در کاتالوگ وارد نشود. با استفاده از این گزینه مشکل فوق برطرف می‌گردد.

Install HW update: اگر سخت‌افزار مورد نیاز در کاتالوگ وجود نداشته باشد با استفاده از این گزینه می‌توان کاتالوگ را از طریق سایت اینترنتی زیمنس به‌روز کرد. توضیحات در ادامه آمده است.

Install GSD File: این گزینه برای معرفی محصولات پروفی‌باس زیمنس یا شرکت‌های دیگر که در کاتالوگ موجود نیستند استفاده می‌شود.

۱۰-۴ آشنایی با کاتالوگ

۱۰-۴-۱ نحوه وارد کردن اجزا

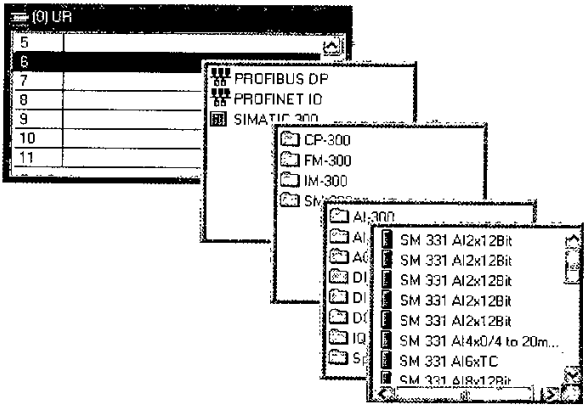
اجزای سخت‌افزاری مورد نیاز را به سه روش می‌توان در برنامه HW Config وارد کرد:

۱. با استفاده از منوی Insert و سپس انتخاب Insert Module

۲. با کلیک راست در قسمت Overview و انتخاب Insert Module

۳. با استفاده از کاتالوگ

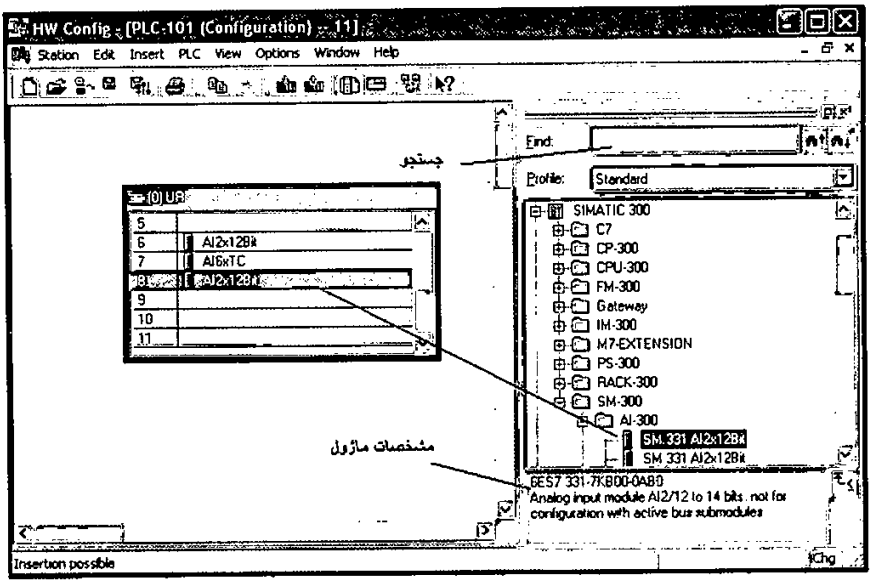
مزیت روش ۱ و ۲ این است که متناسب با نوع Station، سخت‌افزاری که می‌تواند استفاده شود را نشان می‌دهد ولی در روش سوم فیلتری وجود ندارد. در این روش، همه اجزا لیست می‌شوند و کاربر سخت‌افزار دلخواه را انتخاب می‌کند. به‌عنوان مثال در روش اول اگر Station از نوع S7-300 باشد، اجزای S7-400 در لیست ظاهر نمی‌شوند یا وقتی در یک اسلات رک Insert می‌کنیم فقط کارت‌هایی که در آن اسلات قابل نصب هستند نمایش داده می‌شوند؛ ولی در روش دوم همه اجزا در لیست هستند و متناسب با نوع Station تغییر نمی‌کنند.



شکل ۱۰-۲۳ نحوه وارد کردن اجزای سخت افزاری از طریق منوی Insert

مزیت روش سوم این است که اطلاعات سخت‌افزار به‌صورت کامل لیست می‌شود و می‌توان خلاصه ویژگی‌های کارت‌ها را نیز با کلیک روی آنها قبل از وارد کردن به محیط چک کرد.

فصل ۱۰

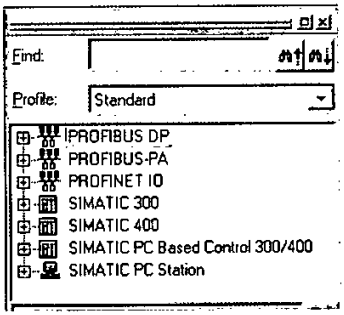


شکل ۱۰-۲۴ نحوه وارد کردن اجزای سخت افزاری از طریق کاتالوگ

مزیت دیگر روش سوم امکان جستجو است که در قسمت بالای کاتالوگ وجود دارد. با وارد کردن نام یا کد سفارش یا بخشی از این موارد می‌توان ماژول مورد نظر را در کاتالوگ جستجو نمود.

۱۰-۴-۲ آشنایی با اجزای سخت‌افزاری کاتالوگ

کاتالوگ همانطور که در شکل ۱۰-۲۵ دیده می‌شود، به چند گروه زیر تقسیم می شود:

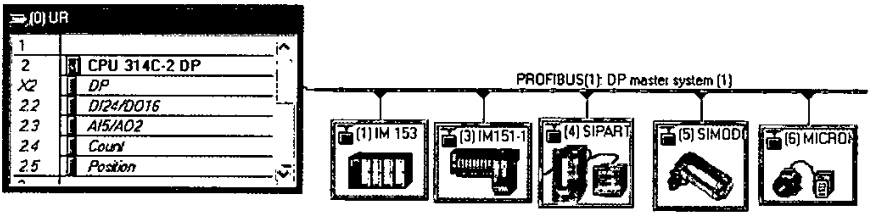


شکل ۱۰-۲۵ دسته بندی اصلی اجزای کاتالوگ

Profibus DP: به‌طور پیش‌فرض در این کاتالوگ اجزای سخت‌افزاری شبکه پروفی‌باس زیمنس وجود دارد.

می‌توان وسایل پروفی‌باس سازندگان دیگر را نیز با نصب فایل GSD از منوی Option به این قسمت اضافه کرد. از وسایل موجود در این کاتالوگ در صورتی می‌توان استفاده نمود که قبلاً در محیط HWC onfig شبکه Profibus DP فعال شده باشد.

شکل ۱۰-۲۶ یک PLC S7-300 را همراه با تعدادی وسیله Profibus DP نشان می‌دهد. جزئیات شبکه پروفی‌باس را در کتاب *مرجع کاربردی پروفی‌باس تألیف محمدرضا ماهر* ببینید.

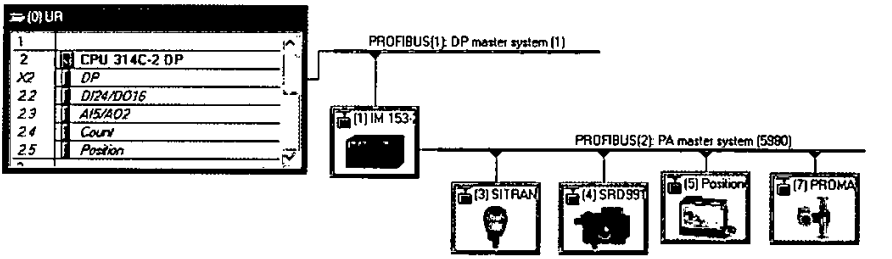


شکل ۱۰-۲۶ پیکربندی PLC S7-300 همراه با وسایل Profibus-DP

Profibus PA: اجزای این بخش مربوط به پروتکل دیگری از پروفی‌باس است که PA نام دارد و با نوع DP متفاوت است. بیشتر وسایل PA ترانسمیتر یا کنترل ولو هستند.

به‌طور پیش‌فرض این بخش خالی است و سخت‌افزاری در آن وجود ندارد. اگر نرم‌افزار Simatic PDM زیمنس نصب شود یا اگر GSD وسایل مورد نظر نصب گردد، اجزای این بخش در دسترس خواهند بود.

در صورتی می‌توان از این اجزا در پیکربندی استفاده نمود که قبلاً شبکه Profibus PA فعال شده باشد. شکل ۱۰-۲۷ یک PLC S7-300 را همراه با وسایل PA نشان می‌دهد.

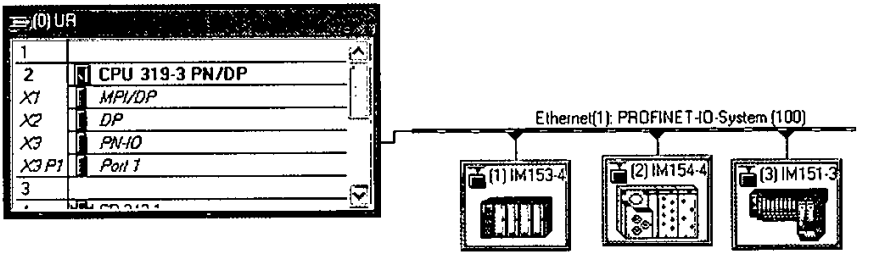


شکل ۱۰-۲۷ پیکربندی PLC S7- 300 همراه با وسایل Profibus-PA

Profinet I/O: اجزای این بخش مربوط به شبکه Profinet است که اگرچه توسط موسسه پروفی‌باس ارائه گردیده ولی ساختار فیزیکی اجزای آن بر پایه اترنت صنعتی است که سرعت بالاتری نسبت به پروفی‌باس دارد.

به‌طور پیش‌فرض چند وسیله Profinet در کاتالوگ فوق موجود هستند. در صورتی‌که قبلاً شبکه پروفی‌نت فعال شده باشد، می‌توان از این وسایل در پیکربندی استفاده نمود.

شکل ۱۰-۲۸ یک PLC S7-300 را همراه با وسایل Profinet نشان می‌دهد.



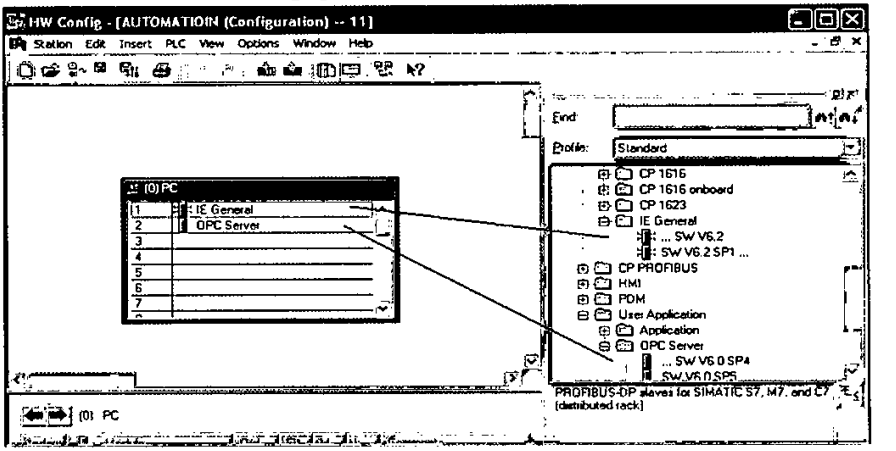
شکل ۱۰-۲۸ پیکربندی PLC S7- 300 همراه با وسایل Profinet

SIMATIC 300: در این قسمت کلیه سخت‌افزارهای مورد استفاده در S7-300 و C7 موجود است. در فصل بعد جزئیات کامل پیکربندی سخت‌افزار S7-300 را با استفاده از این کاتالوگ خواهیم دید.

SIMATIC 400: در این قسمت کلیه سخت‌افزارهای مورد استفاده در S7-400 موجود است. نحوه پیکربندی سخت‌افزار S7-400 با استفاده از این کاتالوگ در کتاب سطح پیشرفته کتاب آمده است.

SIMATIC PC Based Control: در این بخش اجزای مربوط به سیستم‌های WinAC و WinLC آمده است. جزئیات این بخش خارج از بحث کتاب است.

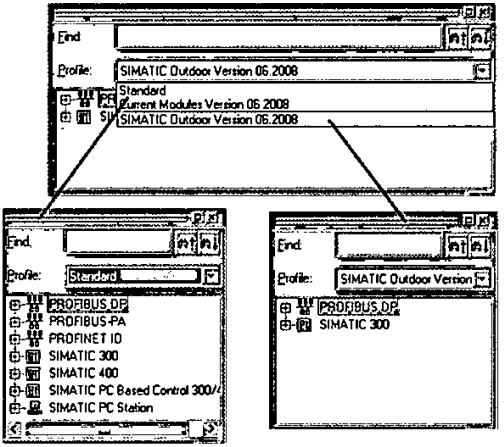
SIMATIC PC Station: در این بخش اجزای سخت‌افزاری و نرم‌افزاری قابل نصب روی PC معرفی شده است. انواع کارت‌های شبکه قابل نصب روی کامپیوتر و برخی نرم‌افزارها در این لیست وجود دارند. به‌عنوان مثال شکل ۱۰-۲۹ پیکربندی یک کامپیوتر را که قرار است به‌عنوان OPC server کار کند نشان می‌دهد.



شکل ۱۰-۲۹ پیکر بندی PC Station

۱۰-۴-۳ پروفایل‌های کاتالوگ

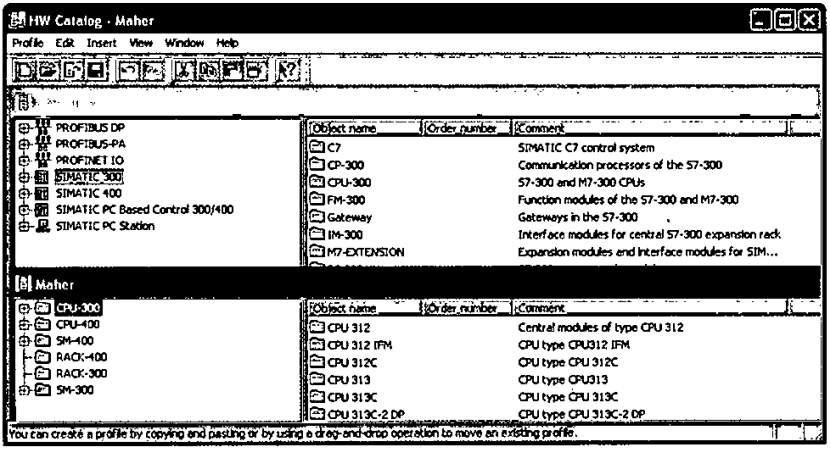
در قسمت‌بالای‌پنجره‌کاتالوگ‌بخش Profile رامشاهده‌می‌کنیم‌که‌با کلیک برروی آن پروفایل‌های مختلف مشاهده می‌شود.



شکل ۱۰-۳۰ پروفایل‌های کاتالوگ

پروفایلِ Standard، همه انواع سخت‌افزارها را نشان می‌دهد ولی با انتخاب پروفایل Outdoor فقط سخت‌افزارهایی که قابلیت نصب Outdoor دارند در کاتالوگ ظاهر می‌شوند.

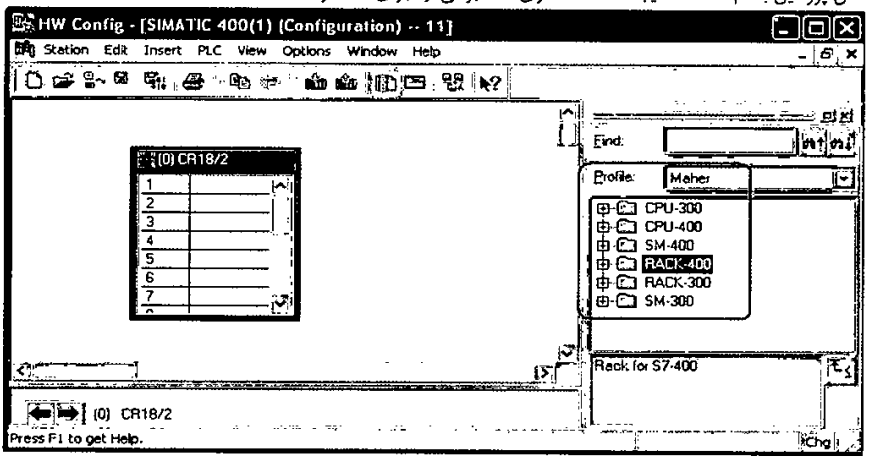
همانطور که قبلاً اشاره شد، کاربر می‌تواند از طریق منوی Option با انتخاب گزینه Edit Catalog Profile یک پروفایل شخصی ایجاد کند، در این‌صورت پنجره نشان داده شده در شکل ۱۰-۳۱ باز می‌شود. در این هنگام، در قسمت بالای پنجره پروفایل‌های قبلی ظاهر می‌شوند، اجزای دلخواه را از این بخش انتخاب نموده و توسط موس آنها را به پنجره پایینی Drag می‌کنیم. سپس پروفایل را به اسم دلخواه ذخیره می‌نماییم.



شکل ۱۰-۳۱ نحوه ایجاد پروفایل دلخواه

فصل ۱۰

پس از بازگشت به محیط اصلی HW Config دربخش کاتالوگ پروفایل جدید ایجاد شده را مشاهده می‌کنیم. در این مثال پروفایل با نام Maher ایجاد شده که حاوی فقط برخی از اجزای 300 و 400 است.



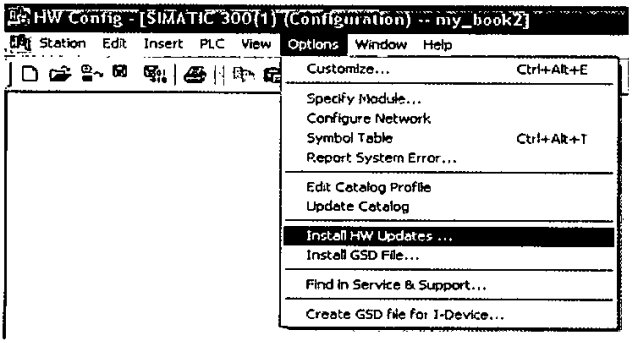
شکل ۱۰-۳۲ پروفایل ایجاد شده توسط کاربر در کاتالوگ

۱۰-۴-۴ Update نمودن کاتالوگ در H.Wconfig

به مرور زمان ممکن است محصولات سخت‌افزاری جدید توسط شرکت زیمنس ارائه گردد. از آنجایی که این محصولات، در لیست محصولات موجود در کاتالوگ نرم‌افزارهای قدیمی‌تر وجود ندارند، لذا هنگام پیکربندی سخت‌افزاری، کاربر با مشکل عدم دسترسی به آنها مواجه خواهد شد. بنابراین لازم است هرازگاهی لیست محصولات موجود در کاتالوگ HW Config به‌روز گردد.

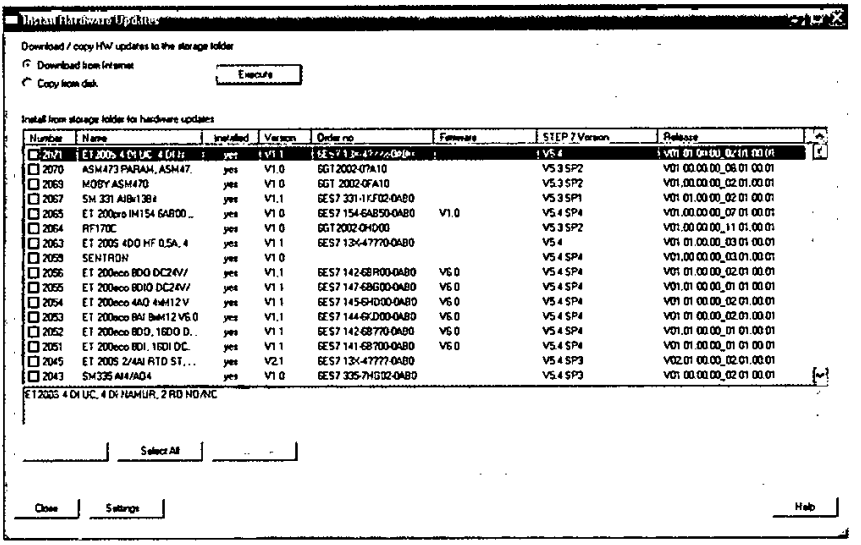
مراحل Update نمودن کاتالوگ به شرح زیر می‌باشد.

۱. از منوی option انتخاب گزینه‌ی Install HW updates را انتخاب کنید.



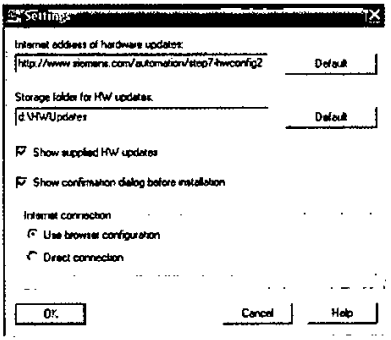
شکل ۱۰-۳۳ مرحله ۱ از مراحل Update کاتالوگ در HW Config

۲. از پنجره باز شده، در قسمت Download/Copy HW updates to the storge folder اگر فایل‌های update بر روی کامپیوتر ذخیره شده باشند، گزینه‌ی Copy from disk را انتخاب نموده و در غیر این‌صورت گزینه‌ی Download from Internet انتخاب شود.



شکل ۱۰-۳۴ پنجره Install HW updates

اگر فایل‌ها از طریق اینترنت دانلود شوند، می‌توان آنها را روی کامپیوتر نیز ذخیره نمود. انتخاب مسیر مورد نظر جهت ذخیره فایل‌ها در کامپیوتر در قسمت Settings و فیلد Storge for HW updates قابل تنظیم می‌باشد.



شکل ۱۰-۳۵ پنجره Settings

سپس از پنجره باز شده Copy Hardware updates می‌توان فایل‌هایی که نیاز است Update آنها نصب شود را انتخاب نمود. پس از کلیک بر روی گزینه‌ی Copy، این فایل‌ها در قسمت Install from storge for Hardware updates در پنجره‌ی Install Hardware updates نمایش داده می‌شوند. در این قسمت در ستون Number می‌توان ماژولی را که می‌خواهیم فایل update آن نصب شود را انتخاب نموده و گزینه‌ی Install را بفشاریم. در ستون Installed مشخص می‌شود که آیا فایل Update نصب شده یا خیر. در این بخش گزینه‌های زیر موجود می‌باشند:

- YES ← فایل Update نصب شده است
- NO ← فایل Update نصب نشده است
- Supplied: در ورژن نرم‌افزار جاری، این فایل موجود بوده است

پس از تکمیل مراحل گفته شده، ماژول‌های جدید در لیست کاتالوگ ظاهر می‌گردند.

۱۰-۵ پرسش و تحقیق

چه سخت‌افزارهایی از خانواده S7 قابلیت Isochrone دارند و نحوه پیکربندی آنها چگونه است؟

فصل ۱۱

نصب، پیکربندی و تنظیمات سخت افزار S7-300

۱۱-۱ مقدمه
۱۱-۲ Rack 300
۱۱-۲-۱ عملکرد
۱۱-۲-۲ ویژگی‌های رک S7-300
۱۱-۲-۳ انواع رک ۳۰۰
۱۱-۲-۴ نصب رک
۱۱-۲-۵ پیکربندی رک در محیط HW Config
۱۱-۳ Power Supply
۱۱-۳-۱ عملکرد
۱۱-۳-۲ انواع منبع تغذیه S7-300
۱۱-۳-۳ ویژگی‌های منبع تغذیه
۱۱-۳-۴ پیکربندی PS در محیط HWConfig
۱۱-۴ CPU
۱۱-۴-۱ عملکرد
۱۱-۴-۲ انواع CPUهای 300
۱۱-۴-۳ ویژگی‌های CPU
۱۱-۴-۴ نصب و سیم‌کشی CPU
۱۱-۴-۵ پیکربندی CPU در محیط HW Config
۱۱-۵ کارت‌های ورودی دیجیتال
۱۱-۵-۱ عملکرد کارت دیجیتال ورودی
۱۱-۵-۲ انواع کارت‌های DI
۱۱-۵-۳ مشخصات فنی کارت‌های DI
۱۱-۵-۴ روش نصب کارت DI در رک
۱۱-۵-۵ پیکربندی کارت DI در Hwconfig
۱۱-۵-۶ قابلیت‌های ویژه در کارت‌های DI
۱۱-۵-۷ کارت‌های DI خاص در S7-300
۱۱-۶ کارت‌های خروجی دیجیتال DO
۱۱-۶-۱ عملکرد
۱۱-۶-۲ انواع کارت DO
۱۱-۶-۳ مشخصات فنی کارت‌های DO
۱۱-۶-۴ روش نصب و سیم‌کشی کارت DO
۱۱-۶-۵ پیکربندی کارت DO در Hwconfig
۱۱-۶-۶ قابلیت‌های ویژه در کارت‌های DO
۱۱-۷ کارت‌های ورودی و خروجی دیجیتال (ترکیبی DI/DO)
۱۱-۸ کارت‌های آنالوگ ورودی
۱۱-۸-۱ عملکرد
۱۱-۸-۲ انواع کارت‌های آنالوگ ورودی (AI)
۱۱-۸-۳ مشخصات فنی کارت‌های آنالوگ ورودی
۱۱-۸-۴ نصب کارت‌های آنالوگ ورودی
۱۱-۸-۵ پیکربندی کارت AI در Hwconfig
۱۱-۹ کارت‌های آنالوگ خروجی
۱۱-۹-۱ عملکرد
۱۱-۹-۲ انواع کارت‌های AO
۱۱-۹-۳ مشخصات فنی کارت‌های آنالوگ خروجی
۱۱-۹-۴ نصب کارت آنالوگ خروجی
۱۱-۹-۵ پیکربندی کارت آنالوگ خروجی در Hwconfig
۱۱-۱۰ کارت‌های ورودی و خروجی آنالوگ (ترکیبی AI/AO)
۱۱-۱۱ کارت‌های خاص خانواده SM
۱۱-۱۱-۱ Dummy Module (DM)
۱۱-۱۱-۲ Simulator module (SM 374)
۱۱-۱۱-۳ Position Decoder Module SM 338
۱۱-۱۲ کارت‌های شبکه Communication Processor (CP)
۱۱-۱۳ Function Module (FM)
۱۱-۱۴ Interface Module (IM)
۱۱-۱۴-۱ عملکرد
۱۱-۱۴-۲ انواع IM در S7-300
۱۱-۱۴-۳ روش‌های برقراری اتصال از طریق IM:
۱۱-۱۴-۴ روش پیکربندی IM در hwconfig
۱۱-۱۵ پیکربندی سیستم C7
۱۱-۱۶ پرسش و تحقیق
۱۱-۱۷ تست‌های خودآزمایی

در این فصل نحوه پیکربندی S7-300 تشریح می‌شود. بحث S7-400 در کتاب سطح پیشرفته آمده است. مطالعه دقیق مباحث این فصل برای کار با PLC ضروری است.

چکیده مطالب

- رک در S7-300 به‌صورت ریل ساده است که فقط نقش نگهداری ماژول‌ها را دارد.
- اگر تعداد کارت‌ها زیاد باشد می‌توان از رک توسعه استفاده کرد و آنرا با رک اصلی از طریق ماژول IM مرتبط نمود.
- پیکربندی منبع تغذیه در محیط Hwconfig الزامی نیست و می‌توان اسلات ۱ را خالی گذاشت.
- CPUهای S7-300 بسیار متنوع هستند. از جنبه‌هایی مانند تعداد ورودی و خروجی و امکانات شبکه و یکپارچه یا ماژولار بودن با یکدیگر تفاوت دارند.
- ارتباط دیتا بین CPU و سایر ماژول‌ها از طریق کانکتوری که در پشت ماژول‌ها نصب می‌شود و به Backplane Connector موسوم است برقرار می‌شود.
- در CPUهای S-300 جدید باتری پشتیبان وجود ندارد.
- در CPUهای S-300 جدید وجود کارت حافظه فلش الزامی است.
- کارت‌های ورودی و خروجی بسیار متنوعند و از نظر تعداد ورودی یا خروجی ، نوع سیگنال و قابلیت‌های خاص با یکدیگر تفاوت دارند. این کارت‌ها در زیر مجموعه SM قرار می‌گیرند.
- کارت های ورودی و خروجی می‌توانند در اسلات ۴ تا ۸ قرار گیرند.
- آدرس دهی کارت‌ها به‌طور خودکار توسط برنامه انجام می‌شود با این وجود کاربر می‌تواند آدرس را به‌دلخواه تغییر دهد.
- قابلیت وقفه کارت‌ها و نحوه کار با سیگنال‌های آنالوگ در کتاب سطح پیشرفته تشریح می‌شود.
- وظیفه ماژول DM ذخیره‌ی اسلات برای یک ماژول پیکربندی نشده می‌باشد.
- SM 374 ماژول سیمولاتور است که برای شبیه‌سازی عملکرد ورودی و خروجی‌ها کاربرد دارد.
- SM 338 برای اتصال به انکودر و دریافت پالس‌های سریع است.
- کارت‌های CP برای اتصال PLC به شبکه‌های صنعتی پروفی‌باس، اترنت، مدباس و ASI عرضه شده‌اند.
- FM ماژول‌هایی هستند که فانکشن خاصی را مستقل از CPU پردازش و اجرا می‌کنند.

اصطلاحات و تعاریف

Slot

به محل قرارگیری ماژول روی رک اسلات گفته می‌شود.

Interrupt

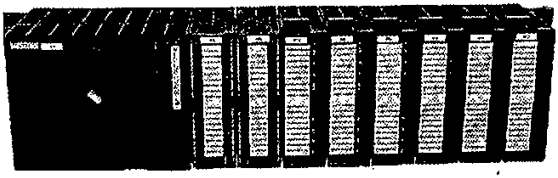
به معنی وقفه است و مفهوم آن عاملی است که برای لحظاتی کار نرمال CPU را متوقف کند. وقفه‌ها اولویت بالاتری نسبت به کار نرمال دارند.

Process image

ناحیه‌ای از حافظه System Memory که CPU از آن برای ذخیره‌سازی ورودی‌ها و ارسال نتایج به خروجی‌ها استفاده می‌کند.

۱۱-۱ مقدمه

PLC های سری S7-300 از قسمت‌های مختلفی تشکیل شده‌اند که به هرکدام از این قسمت‌ها ماژول گفته می‌شود. به‌منظور عملکرد مناسب PLC، باید ماژول‌ها را به نحو صحیح به یکدیگر متصل نموده (پیکربندی سخت‌افزاری به‌صورت عملی) و توسط نرم‌افزار Step7 نیز پیکربندی سخت‌افزار را به‌صورت مجازی در محیط برنامه‌ی HW Config انجام داد و به PLC دانلود نمود.



شکل ۱۱-۱ سخت‌افزار PLC S7-300

به‌طور کلی می‌توان گفت که یک PLC S7-300 از قسمت‌های زیر تشکیل شده است:

Rack

در سری S7-300 وظیفه رک نگهداری ماژول‌ها می‌باشد و قسمت‌های مختلف تشکیل‌دهنده‌ی PLC روی آن نصب می‌گردند.

PS

معرف Power Supply یعنی منبع تغذیه است و وظیفه‌ی آن تأمین ولتاژ و جریان مورد نیاز CPU و سایر قسمت‌های سیستم می‌باشد.

CPU

واحد پردازش مرکزی بوده و وظیفه آن پردازش برنامه کاربر می‌باشد.

SM

معرف Signal Module است و منظور از آن کارت رابط I/Oها (ورودی، خروجی) می‌باشد.

IM

معرف Interface Module به معنی ماژول رابط است و وظیفه آن برقرای ارتباط بین رک اصلی و رک‌های توسعه می‌باشد.

DM

معرف Dummy Module به معنی ماژول کاذب است و وظیفه‌ی آن رزرو اسلات‌های خالی جهت قرار دادن ماژول در آینده می‌باشد.

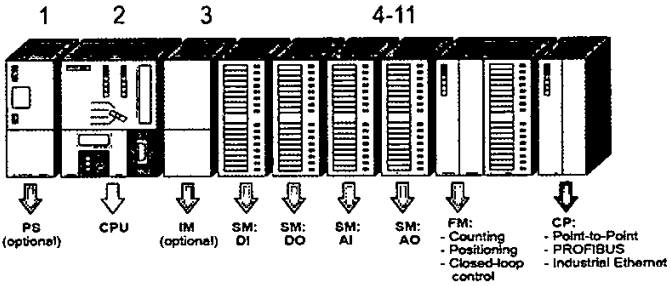
FM

معرف Function Module است و به‌منظور انجام مقاصد خاص مانند انجام PID کنترل، کنترل موقعیت، شمارش سریع و . . . طراحی شده است.

CP

معرف Communication Processor به معنی کارت شبکه است و به‌منظور ارتباط PLC با شبکه‌های صنعتی طراحی شده است.

شکل ۱۱-۲ ترتیب قرارگیری ماژول‌ها را روی رک نمایش می‌دهد. همانطور که بعداً خواهیم دید قرار دادن PS (منبع تغذیه) و IM (ماژول واسط) روی رک، در سری S7-300 به‌صورت اختیاری می‌باشد. در صورت دلخواه می‌توان به جای PS زیمنس، از هر منبع تغذیه دیگری با خروجی 24 V DC استفاده نمود.



شکل ۱۱-۲ نمونه ترتیب قرارگیری ماژول‌ها روی رک

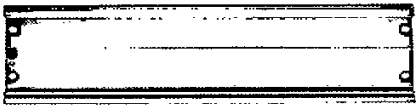
با این مقدمه به شرح و بررسی ماژول‌های مختلف PLC S7-300 می‌پردازیم.

Rack 300 ۲-۱۱

۱-۲-۱۱ عملکرد

همانطور که قبلاً ذکر شد، ماژول‌های مختلف PLC روی رک قرار می‌گیرند. در S7-300 رک از جنس آلومینیوم است و کارت‌ها روی آن از چپ به راست نصب می‌شوند. رک به‌جز اتصال به زمین هیچ ارتباط الکتریکی بین ماژول‌ها ایجاد نمی‌کند چون باس ارتباطی ندارد.

به‌طور کلی، Rack 300 فقط دو نقش ایفا می‌کند یکی نگهداری ماژول‌ها و دیگری اتصال زمین ماژول‌ها.



شکل ۱۱-۳ رک S7- 300

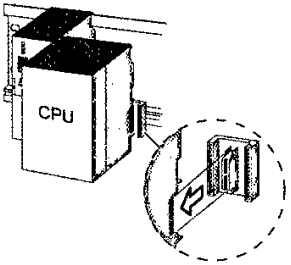
وقتی ماژول‌ها روی رک نصب می‌شوند قسمت فلزی پشت کارت که برای اتصال زمین است با بدنه ریل تماس پیدا می‌کند. با زمین کردن ریل تمام ماژول‌ها به زمین متصل خواهند شد.

هر ماژول روی رک فضایی را اشغال می‌کند که به آن *اسلات* گفته می‌شود. در عمل روی رک هیچ محلی به‌عنوان اسلات شماره‌گذاری نشده است و می‌توان اولین کارت را در هر نقطه‌ای نصب نمود ولی بقیه کارت‌ها به‌ترتیب پس از آن نصب خواهند شد. در S7-400 اسلات‌ها شماره‌گذاری شده و بایستی در هنگام نصب مراعات شود.

در عمل برخی کارت‌ها باریک‌تر و برخی دارای عرض بیشتر هستند ولی هردو در S7-300 فقط یک اسلات را اشغال می‌کنند، در حالی‌که در S7-400 کارت‌های عریض‌تر در عمل و نیز در پیکربندی HW Config دو یا حتی سه اسلات را اشغال می‌نمایند.

در Rack 300 به منظور برقراری ارتباط بین ماژول‌های مختلفی که روی آن نصب می‌شوند از یک کانکتور موسوم به Bus Connector که در برخی موارد Backplane Connector نیز گفته می‌شود استفاده می‌گردد. این کانکتور قبل از نصب هر ماژول در پشت آن قرار می‌گیرد تا کارت بعدی پس از نصب روی باس قرار گیرد.

بدیهی است اگر ماژولی را از وسط باس خارج کنیم ارتباط بین کارت‌های باقیمانده در دو طرف قطع خواهد شد؛ در حالی‌که در S7-400 که رک به‌صورت ریل نیست و مجهز به باس است این مشکل وجود ندارد.



شکل ۱۱-۴ استفاده از BUS CONNECTOR

۱۱-۲-۲ ویژگی‌های رک S7-300

رک در سری S7-300 دارای خصوصیاتی به شرح زیر می باشد:

- ۱۱ اسلات دارد؛ یعنی حداکثر ۱۱ ماژول می‌توان روی آن قرار داد.
- در صورتی‌که نیاز به استفاده از ماژول‌های بیشتری باشد، باید از رک توسعه استفاده نمود.
- به لحاظ ساختار، دارای یک مدل است که هم به‌عنوان رک اصلی و هم رک توسعه[1] استفاده می‌شود.
- حداکثر سه رک توسعه می‌توان داشت.
- ارتباط بین رک اصلی و رک‌های توسعه توسط ماژولی به نام IM انجام می‌گیرد.

1. Expansion Rack

- رک، صرفاً نقش نگهداری ماژول‌ها را بر عهده دارد و ارتباط بین ماژول‌ها با هم، از طریق رابطی به نام BUS Connector صورت می‌پذیرد.
- ترتیب چیدمان ماژول‌ها روی رک در S7-300 مهم بوده و باید رعایت گردد.
- فضای خالی بین اسلات‌ها روی رک مجاز نمی‌باشد.

۱۱-۲-۳ انواع رک ۳۰۰

رک‌های موجود در S7-300، از نظر طول به انواع مختلف مندرج در جدول ۱۱-۱ تقسیم‌بندی می‌گردند.

جدول ۱۱-۱ طول‌های مختلف رک

شماره سفارش	طول مفید قابل استفاده	طول ظاهری ریل
ES7 390-1AB60-0AA0	120 mm	160 mm
ES7 390-1AE80-0AA0	450 mm	482.6 mm
ES7 390-1AF30-0AA0	480 mm	530 mm
ES7 390-1AJ30-0AA0	780 mm	830 mm
ES7 390-1BC00-0AA0	cut to length if required	2,000 mm

اگر تعداد ماژول‌ها کم باشد از طول‌های کوچکتر استفاده می‌شود ولی آنچه که بدیهی است، این است که رک دارای هر ابعادی هم که باشد، ماکزیمم ۱۱ ماژول می‌توان روی آن قرار داد.

عرض ماژول‌های I/O در بیشتر موارد ثابت و 40 mm است ولی منبع تغذیه بسته به میزان جریان خروجی عرض متفاوت دارد که بین 50 mm تا 2000 mm است. CPU نیز بسته به نوع آن دارای عرض متغیر است. بنابراین با احتساب عرض ماژول‌ها و در نظر گرفتن دو اسلات به‌عنوان Spare[1] برای آینده رک مورد نظر انتخاب می‌گردد.

رک‌های بلند مانند طول ۲ متر را می‌توان به اندازه‌های دلخواه برش داد و آنها را به چند رک کوچکتر تقسیم نمود. ممکن است در مواردی یک رک ۲ متری هم به‌عنوان رک اصلی و هم به‌عنوان رک توسعه به‌کار رود. به این صورت که در یک سمت ماژول‌های رک اصلی و در سمت دیگر ماژول‌های رک توسعه نصب گردد.

۱۱-۲-۴ نصب رک

رک S7-300 را می‌توان به‌صورت افقی یا عمودی نصب نمود. نصب عمودی متداول نیست و ممکن است در برخی شرایط خاص که عرض پنل کم است استفاده شود. لازم است توجه شود که در انتخاب نصب رک به‌صورت عمودی دمای داخل پنل بایستی خنک‌تر باشد زیرا به دلیل چیدمان نصب، هوای خنک‌کنندگی به‌خوبی مانند حالت افقی از ماژول‌ها عبور نخواهد کرد. شرایط دمایی زیر برای نصب افقی و عمودی بایستی رعایت شود:

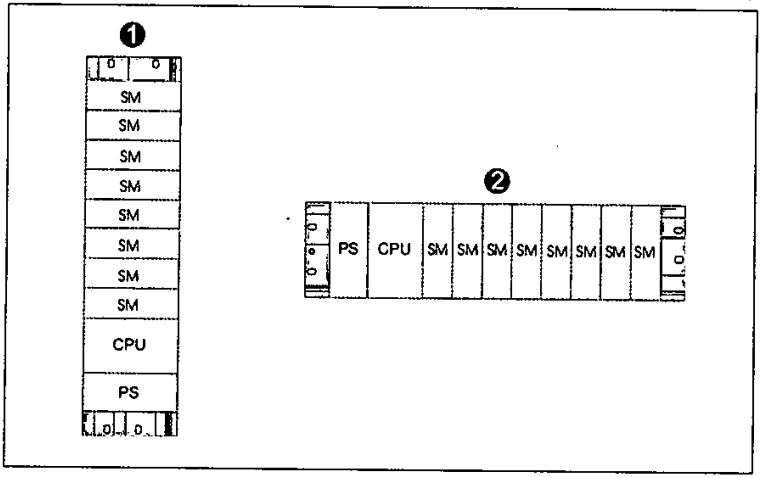
- در نصب عمودی دمای ۰ الی ۴۰ درجه سانتی‌گراد

---

۱. رزرو

- در نصب افقی دمای ۰ الی ۶۰ درجه سانتی‌گراد

شکل ۱۱-۵ طریقه نصب رک و ترتیب ماژول‌ها را در حالت افقی و عمودی نشان می‌دهد.

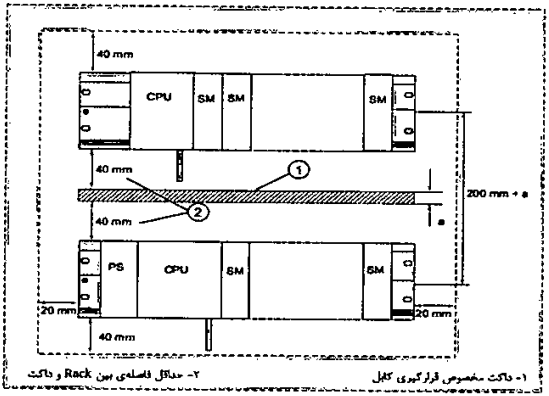


شکل ۱۱-۵ ۱. نصب رک به صورت عمودی، ۲. نصب رک به صورت افقی

طریقه نصب رک در پنل

قبل از نصب کارت ها ابتدا بایستی رک در محل مناسب از پنل نصب شود. رعایت فاصله‌های مجاز و نکاتی که در فصل ۴ ذکر شد ضروری است.

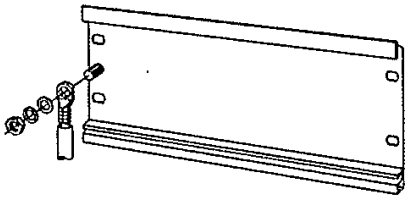
شکل ۱۱-۶ طریقه‌ی نصب صحیح رک در پنل را نشان می‌دهد.



شکل ۱۱-۶ طریقه‌ی نصب صحیح رک در کابین

اتصال زمین رک

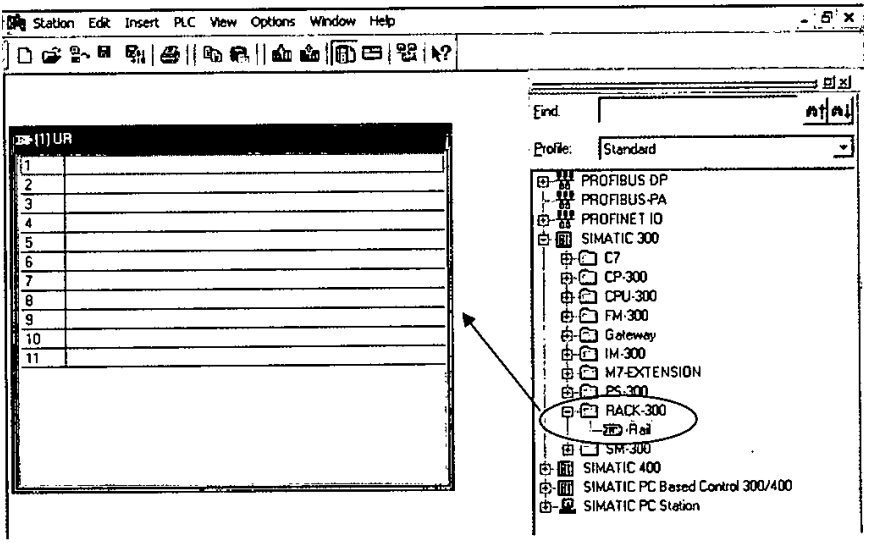
در رک S7-300 محلی به منظور اتصال زمین تعبیه شده است که می‌توان کابل زمین را به آن متصل نمود. شکل ۱۱-۷ طریقه‌ی اتصال کابل زمین به رک S7-300 را نشان می‌دهد.



شکل ۱۱-۷ طریقه‌ی اتصال کابل زمین به رک S7-300

۱۱-۲-۵ پیکربندی رک در محیط HW Config

لازم است که رک در محیط HW Config نیز به‌عنوان اولین قسمت وارد شود. برای این منظور می‌توان از منوی Insert یا از کاتالوگ به شرحی که در فصل قبل ذکر شد استفاده کرد. در استفاده از کاتالوگ می‌توان Station300 را انتخاب نموده و روی علامت + کنار آن کلیک کرد. سپس از لیست باز شده گزینهRACK-300 را باز نموده و Rail را انتخاب نمود. با دوبارکلیک نمودن بر روی Rail یا کشیدن آن با موس، رک در صفحه اصلی قرار می‌گیرد.



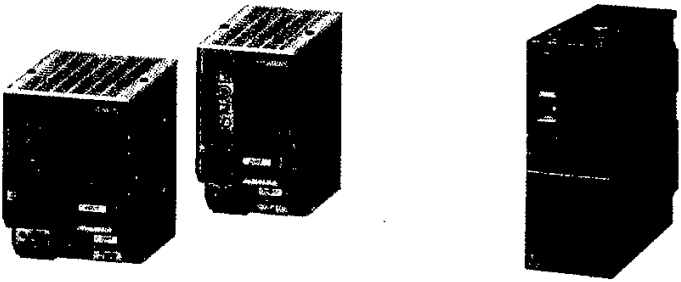
شکل ۱۱-۸ وارد نمودن رک در محیط HW Config

همانطور که در شکل ۱۱-۸ مشاهده می‌گردد، رک S7-300 دارای ۱۱ ردیف می‌باشد که به هر کدام از این ردیف‌ها یک اسلات گفته می‌شود. در ادامه بیان خواهد شد که چگونه سایر ماژول‌ها را می‌توان روی رک قرار داد.

Power Supply ۳-۱۱

۱-۳-۱۱ عملکرد

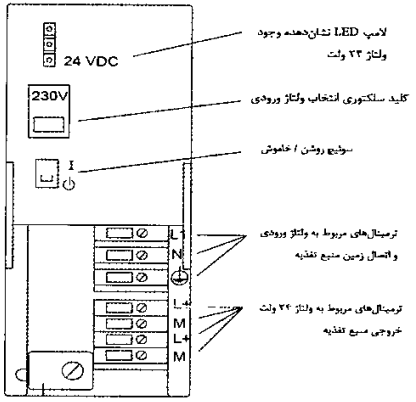
نقش منبع تغذیه، تأمین ولتاژ و جریان مورد نیاز CPU می‌باشد. در عین حال می‌توان ولتاژ مورد نیاز برای سایر قسمت‌ها (مانند تغذیه کارت‌ها) را نیز از آن دریافت نمود. در سری S7-300 ولتاژ مورد نیاز برای اتصال به CPU یا سایر کارت‌ها 24VDC است. این ولتاژ را می‌توان از هر منبع تغذیه دلخواهی دریافت کرد. یعنی استفاده از PS 300 الزامی نیست و برنامه HW Config نیز به آن حساسیتی نشان نمی‌دهد. همانطور که در ادامه خواهیم دید، در HW Config می‌توان اسلات اول که مربوط به تغذیه است را خالی گذاشت. در S7-400 استفاده از منبع تغذیه و معرفی آن در HW Config الزامیست.



الف) منبع تغذیه S7-300 — ب) منبع تغذیه عمومی با خروجی 24VDC

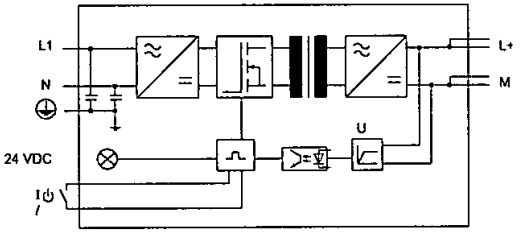
شکل ۱۱- ۹ انواع منبع تغذیه مورد استفاده در S7

شکل ۱۱-۱۰ منبع تغذیه‌ی 2A را همراه با جزئیات آن نمایش می‌دهد. ترمینال‌های اتصال و کلید و لامپ‌های منبع تغذیه در این شکل مشخص است. سایر منابع تغذیه سری S7-300 نیز دارای شکل مشابهی هستند.



شکل ۱۱-۱۰ PS 307 2A و جزئیات آن

ازنظر الکتریکی مدار داخلی منبع تغذیه به‌صورت زیر است. همانطور که دیده می‌شود بخش AC از بخش DC کاملاً ایزوله می‌باشد.



شکل ۱۱-۱۱ مدار داخلی منبع تغذیه

منابع تغذیه S7-300 دارای قابلیت تشخیص حالت‌های کاری غیر نرمال مانند اضافه بار، اتصال‌کوتاه و . . . می‌باشند. جدول ۱۱-۲ حالت‌های پیش‌آمده و وضعیت منبع تغذیه را نشان می‌دهد.

جدول ۱۱-۲ حالت‌های پیش‌آمده و وضعیت منبع تغذیه

خطای پیش آمده	وضعیت منبع تغذیه	وضعیت 24V DC LED
اضافه جریان در خروجی منبع	افت در ولتاژ خروجی، کاهش عمر مفید منبع	چشمک‌زن
اتصال کوتاه در خروجی منبع	ولتاژ خروجی صفر می‌گردد، بعد از برطرف شدن خطا، ولتاژ به‌صورت اتوماتیک به حالت نرمال باز می‌گردد.	نور لامپ تیره می‌شود.
اضافه ولتاژ در ورودی منبع	می‌تواند باعث خرابی منبع تغذیه گردد.	-
کاهش ولتاژ در ورودی منبع	به‌طور اتوماتیک ولتاژ خروجی قطع می‌گردد، پس از برطرف شدن خطا، ولتاژ به‌طور اتوماتیک به حالت نرمال باز می‌گردد.	نور لامپ تیره می‌شود.

۱۱-۳-۲ انواع منبع تغذیه S7-300

در S7-300 ولتاژ خروجی همه انواع منبع تغذیه 24 V DC می‌باشد. تقسیم‌بندی آنها بر اساس جریان خروجی، به‌صورت زیر است:

- PS 307 2A
- PS 307 5A
- PS 307 10A

۱۱-۳-۳ ویژگی های منبع تغذیه

برای هر منبع تغذیه مشخصات فنی خاصی عرضه شده است که موارد مهم در آن عبارتند از:

ولتاژ ورودی: که می‌تواند 120 /230 VAC باشد.

فرکانس ورودی: که می‌تواند 50 HZ یا 60 HZ باشد.

جریان نامی ورودی منبع تغذیه: دو مقدار مجاز یکی برای 120 AC و دیگری برای 230 AC داده می‌شود. به‌عنوان مثال برای PS 307 2A داریم:

Rated Input Current	
• at 120 VAC	0.8 A
• at 230 VAC	0.5 A

ولتاژ خروجی منبع تغذیه: 24VDC +/- 5%

جریان خروجی منبع تغذیه: که بسته به نوع آن 2A ، 5A یا 10A است.

بازدهی (راندمان): که براساس تلفات محاسبه می‌شود. برای برخی از PSها این مقدار 83% است.

توان ورودی: که متناسب با جریان منبع تغذیه متفاوت است. مثلاً برای نوع 2A این عدد 58W می‌باشد.

توان تلف شده: به‌صورت Typical بیان می‌شود و متناسب با جریان منبع تغذیه متفاوت است. مثلاً برای نوع 2A این عدد 10w می‌باشد.

شکل زیر مشخصات فنی منبع تغذیه PS307 2A را به‌عنوان نمونه نشان می‌دهد.

Other Parameters	
Protection class according to IEC 536 (DIN VDE 0106, Part 1)	I, with protective grounding conductor
Insulation	
• Rated insulation level (24 V to L1)	250 VAC
• Tested with	2800 VDC
Reliable isolation	SELV circuit
Bridging of power failures (at 93 and/or 187 V)	min. 20 ms
• Repeat rate	min 1 s
Efficiency	83 %
Power input	58 W
Power loss	typ. 10 W
Diagnostics	
LED for output voltage available	Yes, green LED
Dimensions and Weight	
Dimensions W × H × D (in millimeters)	50 × 125 × 120
Weight	Approx. 420 g

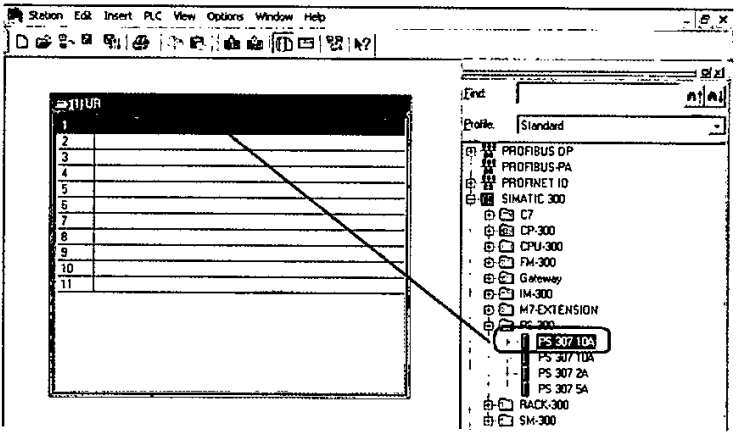
Input Rating	
Input voltage	
• Rated value	120 / 230 VAC
System frequency	
• Rated value	50 Hz or 60 Hz
• Permitted range	47 Hz to 63 Hz
Rated input current	
• At 230 V	0.5 A
• At 120 V	0.8 A
Inrush current (at 25°C)	20 A
I^2t (at inrush current)	1 A^2s
Output Rating	
Output voltage	
• Rated value	24 VDC
• Permitted range	24 V ± 5 %, stable at no load
• Ramp-up time	max. 2.5 s
Output current	
• Rated value	2 A, Cannot be connected in parallel
Short-circuit protection	Electronic, nonlatching, 1.1 to 1.3 × I_N
Residual ripple	max. 150 mV_{ss}

شکل ۱۱-۱۲ مشخصات فنی PS 307 2A

۱۱-۳-۴ پیکربندی PS در محیط HWConfig

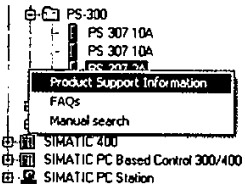
به‌منظور وارد نمودن منبع تغذیه در رک، از کاتالوگ گزینه PS-300 را انتخاب می‌نماییم. همانطور که در شکل ۱۱-۱۳ مشاهده می‌گردد، می‌توان هر کدام از منابع تغذیه‌ای که در این قسمت وجود دارند را درون رک و در اسلات شماره یک قرار

داد. برای این منظور می‌توان ابتدا اسلات یک را با موس انتخاب نمود که در این صورت رنگ این اسلات به صورت رنگ سبز در می‌آید، سپس روی PS مورد نظر در کاتالوگ دوبارکلیک نمود. روش دیگر این است که با موس PS مورد نظر را انتخاب نموده و آنرا به محیط رک کشیده و در اسلات شماره یک رها نمود (Drag And Drop).

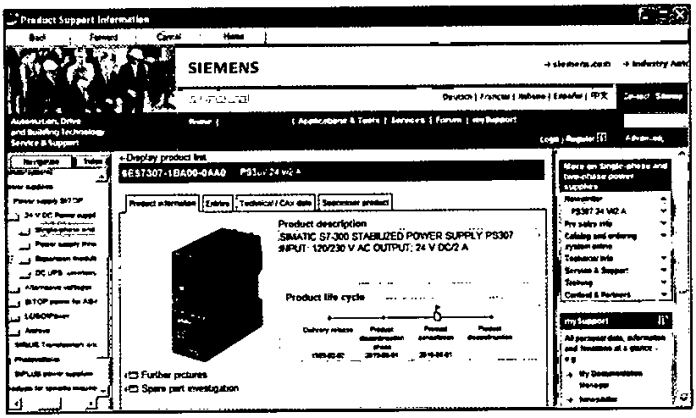


شکل ۱۱-۱۳ وارد نمودن PS در محیط HW Config

مشخصات کلی منبع تغذیه S7-300 را می‌توان در قسمت پایین پنجره کاتالوگ مشاهده کرد و اگر نیاز به جزئیات بیشتر باشد بایستی به کاتالوگ مربوط به آن مراجعه نمود. با اتصال کامپیوتر به اینترنت، می‌توان اطلاعات فنی را از سایت زیمنس به‌صورت Online مشاهده کرد. برای این منظور می‌توان مانند شکل ۱۱-۱۴ روی ماژول مورد نظر کلیک راست نموده و گزینه Product Support Information را انتخاب نمود. در این‌صورت به‌طور اتوماتیک وارد سایت اینترنتی زیمنس شده و مشخصات فنی ماژول مانند شکل ۱۱-۱۵ نمایش داده می‌شود.



شکل ۱۱-۱۴ نحوه دریافت اطلاعات فنی ماژول به‌صورت Online از سایت زیمنس

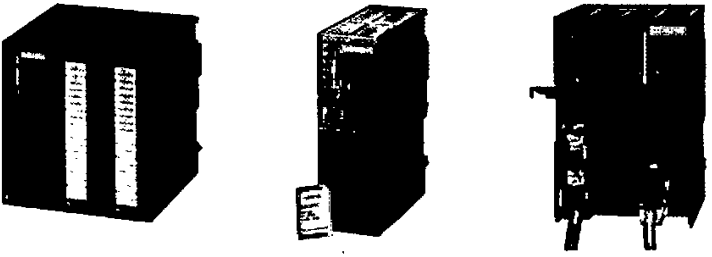


شکل ۱۱-۱۵ دسترسی Online به مشخصات فنی ماژول‌ها

CPU ۴-۱۱

۱۱-۴-۱ عملکرد

CPU مهم‌ترین واحد یک PLC محسوب می‌گردد که وظیفه آن پردازش برنامه‌ی کاربر است و بدون آن سایر اجزای PLC قادر به کارکردن نمی‌باشند. در S7-300، CPUها دارای تنوع زیادی هستند. اگر به کاتالوگ موجود در برنامه‌ی HW Config مراجعه کنیم، مشاهده می‌شود که این CPUها از شماره 312 شروع شده و تا 319 ادامه می‌یابند. هرچه شماره CPU بیشتر باشد، توانایی آن CPU بیشتر خواهد بود. در ادامه به بررسی انواع CPUهای S7-300 خواهیم پرداخت. در شکل ۱۱-۱۶ چند نمونه از CPU های S7-300 مشاهده می‌شود.



الف) CPU 317-2 PN/DP ب) CPU315-2DP ج) CPU 313C

شکل۱۱-۱۶ چند مدل از PLCهای S7-300

۱۱-۴-۲ انواع CPUهای 300

CPU های خانواده 300 را بر اساس مشخصه‌های مختلفی می‌توان تقسیم‌بندی کرد:

- ازنظر Compact یا Modular بودن
- از نظر توانایی ارتباط مستقیم با شبکه‌های صنعتی
- از نظر امکانات Fail Safe
- از نظر تعداد I/O ، سرعت پردازش و حجم حافظه

قبل از معرفی تقسیم‌بندی‌های فوق لازم است حروف و علائمی که در کد CPUها به‌کار می‌رود معرفی شود. هر CPU دارای کد خاصی است که از چپ به راست طبق ساختار جدول ۱۱-۳ نوشته می‌شود.

جدول ۱۱-۳ کدهای مندرج بر روی CPUهای مختلف S7-300

CPU	31x			
		C IFM F	-2 -3	PtP DP PN/DP

راهنما

- به‌جای x یک عدد وجود دارد که نوع CPU را نشان می‌دهد. این عدد از ۲ تا ۹ می‌تواند باشد مانند CPU312 یا CPU 315 .
- اگر پس از عدد سه رقمی فوق هیچ حرف یا عدد دیگری معرفی نشود نشانگر این است که CPU ماژولار است و به‌جز MPI هیچ پورت شبکه دیگری ندارد.
- حرف C به مفهوم Compact است.
- حروف IFM به معنای Integrated Function Module است.
- حرف F به معنای Fail safe است.
- علامت 2- یعنی CPU دارای دو پورت شبکه است
- علامت 3- یعنی CPU دارای سه پورت شبکه است.
- حروف PtP به معنای پورت سریال Point to Point است.
- حروف DP به معنای پورت شبکه Profibus DP است.
- حروف PN به معنای پورت شبکه Profinet است.

توضیح کامل موارد فوق در ادامه خواهد آمد. اکنون به بررسی انواع CPUهای S7-300 طبق تقسیم‌بندی ذکر شده می‌پردازیم.

۱۱-۴-۲-۱ CPU ها از نظر Compact یا Modular

CPUهای ماژولار آنهایی هستند که به تنهایی عرضه شده‌اند و در کنار آنها هیچگونه ماژول I/O که متصل به CPU و به آن یکپارچه باشد وجود ندارد؛ در حالی‌که در نوع Compact کارت‌های خاصی به CPU متصل هستند.

نمونه Compact را در شکل ۱۱-۱۶ با عنوان 313C مشاهده می‌کنید. بیشتر CPUهای خانواده 300 به‌صورت ماژولار و تعدادی به‌صورت Compact هستند.

CPUهای Compact را به‌سادگی از روی کد آنها می‌توان شناخت. حرف C یا حروف IFM در کد CPU معرف یکپارچه بودن آنست. به‌عنوان مثال CPU313C یا CPU314 IFM هردو از نوع یکپارچه هستند.

سوالی که در اینجا مطرح می‌شود این است که این CPUها چه ویژگی‌هایی نسبت به CPUهای ماژولار دارند؟ آیا هدف این بوده که به‌جای خرید CPU و کارت I/O به‌صورت مجزا، کاربر فقط یک ماژول که هر دو مورد را دارد خریداری کند که از نظر اقتصادی هم به صرفه باشد؟ جواب مثبت است ولی این تنها ویژگی CPUهای Comapct نیست. در مدل‌های Compact علاوه بر ویژگی فوق، CPU مجهز به یک فانکشن ماژول (FM) داخلی است که می‌تواند سیگنال‌های سریع را بگیرد و برای مواردی مانند کانتر سریع یا کنترل موقعیت نیز به‌کار رود. کد IFM که مخفف Integrated Function Module است به همین توانایی اشاره می‌کند.

لازم به ذکر است که نوع IFM نسبت به نوع C قدیمی‌تر هستند و فرکانس پالس‌هایی که می‌توانند دریافت کنند کمتر است، از اینرو به‌تدریج منسوخ شده و امروزه از نوع C به‌جای آنها استفاده می‌گردد. بنابراین برای کاربردهایی که پالس‌های سریع تا فرکانس 60kHZ بایستی شمارش شوند به‌جای خرید یک CPU ماژولار و کارت FM می‌توان از یک CPU مدل Compact مانند CPU314C استفاده کرد که اقتصادی‌تر خواهد بود.

برخی CPUهای Compact (مانند CPU314C-2DP) علاوه بر توانایی دریافت پالس‌های سریع تعداد محدودی آنالوگ ورودی و خروجی نیز در کنار خود دارند.

CPUهای Compact شبیه CPUهای ماژولار قابل توسعه بوده و می‌توان در کنار آنها سایر کارت‌های دیگر نظیر SM، CPو FM را نیز نصب نمود.

در جدول ۱۱-۴ انواع CPUهای Compact و Modular خانواده S7-300 که تا زمان تألیف کتاب عرضه شده‌اند لیست گردیده است.

جدول ۱۱-۴ لیست CPUهای Compact و Modular در S7-300

CPU 300 Modular Type	CPU 300 Compact Type
• CPU312	• CPU312 IFM
• CPU313	• CPU312C
• CPU314	• CPU313C
• CPU315	• CPU313C-2 PtP
• CPU315-2DP	• CPU313C-2 DP
• CPU315-2 PN/DP	• CPU314C
• CPU316	• CPU314C-2 PtP
• CPU316-2DP	• CPU314C-2 DP
• CPU317	• CPU314 IFM
• CPU318-2	
• CPU319-3 PN/DP	

۱۱-۴-۲-۲ انواع CPU از نظر پورت شبکه

تمام CPUهای S7-300 (و نیز S7-400) دارای پورت شبکه MPI می‌باشند ولی سایر پورت‌های شبکه فقط روی برخی از انواع CPUها وجود دارند. همانطور که قبلاً ذکر شد، حروف PtP ، DP و PN معرف پورت شبکه‌ای هستند که روی CPU وجود دارد.

پورت DP

در این CPUها، بعد از کلمه CPU و شماره آن، علامت 2DP به‌کار رفته است مانند CPU 313 C-2DP یا CPU 315 2DP. در این حالت CPU علاوه بر داشتن پورت MPI، دارای یک پورت دیگر به نام DP نیز می‌باشد که این پورت به‌منظور ارتباط PLC با شبکه Profibus DP استفاده می‌گردد. همانطور که مشخص است هم CPUهای ماژولار و هم CPUهای Compact دارای مدل‌های 2DP می‌باشند.

CPU های دارای پورت PtP

در این CPU ها، بعد از کلمه CPU و شماره آن، علامت PtP 2 به‌کار رفته است مانند CPU 313 C-2 PtP.

در این حالت CPU علاوه بر داشتن پورت MPI، دارای یک پورت دیگر به نام PtP نیز می‌باشد که این پورت به منظور ارتباط PLC به‌صورت Point to Point استفاده می‌گردد. این پورت ارتباطات سریال نظیر RS232 و RS422 و RS485 که می‌تواند برای شبکه Modbus استفاده شود را ساپورت می‌کند.

CPU های دارای پورت PN

در این CPUها پورت Profinet که از نظر ظاهری مشابه اترنت صنعتی است وجود دارد. فقط CPUهای جدید مانند CPU317 و CPU319 مجهز به این پورت شده‌اند. وسایل Remote I/O جدید می‌توانند از طریق سوئیچ‌های اترنت به این پورت متصل شوند.

CPU های دارای پورت قابل تنظیم MPI/DP

برخی از CPUهای S7-300 دارای پورت‌های قابل تنظیم می‌باشند. یعنی کاربر می‌تواند از طریق نرم‌افزار Step7 و در زیربرنامه HW Config، مشخص نماید که کدام پورت CPU به منظور چه ارتباطی در نظر گرفته شود. بر اساس نوع و تعداد پورت‌های قابل تنظیم، CPUهای S7-300 به انواع زیر تقسیم می‌شوند:

- برخی از CPUهای S7-300 دارای دو پورت می‌باشند، به‌گونه‌ای که یکی از پورت‌ها به‌عنوان پورت DP منظور شده و دیگری را می‌توان به‌صورت MPI یا DP تنظیم نمود. مثلاً در CPU 317 2DP، پورت اول را می‌توان به‌عنوان MPI یا DP تنظیم نمود، و پورت دوم به‌عنوان پورت DP در نظر گرفته شده است.

  در برخی از این نوع CPUها، بعد از شماره CPU عدد 2 نوشته شده ولی علامت DP درج نشده است، مانند CPU 317-2 یا CPU 318-2 ؛ در این CPUها نیز دو پورت وجود دارد. پورت اول می‌تواند به‌عنوان MPI/DP تنظیم گردد و پورت دوم به‌عنوان پورت DP در نظر گرفته شده است.

- برخی دیگر از CPUهای S7-300 مانند CPU 315-2 PN/DP نیز دارای دو پورت می‌باشند، با این تفاوت که پورت دوم آنها به‌منظور ارتباط با شبکه صنعتی پروفی‌نت طراحی شده است. پورت اول در این نوع CPUها، می‌تواند به‌عنوان MPI/DP مورد استفاده قرار گیرد.
- برخی از CPUهای S7-300 مانند CPU 319-3 PN/DP دارای سه پورت می‌باشند. پورت اول قابل تنظیم به‌عنوان MPI/DP است، پورت دوم به‌عنوان DP و پورت سوم به‌عنوان PN (پروفی‌نت) می‌باشد.

۱۱-۴-۲-۳ CPU های Fail Safe

این CPUها از مدل‌های **Fail Safe** (ایمن در برابر خطا) بوده و مشخصه‌ی آنها علامت F بعد از شماره CPU می‌باشد. این CPUها از نظر سخت‌افزاری با انواع معمولی متفاوت هستند و تنظیمات و پیکربندی و برنامه‌نویسی آنها نیز به روش‌های خاصی انجام می‌شود که در کتاب جداگانه‌ای در مورد آنها بحث خواهد شد.

۱۱-۴-۳ ویژگی های CPU

CPU های خانواده S7-300 را از جنبه‌های دیگری نیز می‌توان مورد بررسی و مقایسه قرار داد. برخی از ویژگی‌های مهم عبارتند از:

۱۱-۴-۳-۱ تعداد ورودی و خروجی

هر CPU تعداد ورودی و خروجی محدودی را ساپورت می‌کند. علت این محدودیت آن است که بخشی که به‌عنوان حافظه ورودی و خروجی‌ها در CPU وجود دارد دارای اندازه مشخصی است. برای اطلاع از ماکزیمم تعداد ورودی و خروجی یک CPU دو راه وجود دارد:

۱. مراجعه به کاتالوگ و مشخصات فنی CPU
۲. ارتباط Online با CPU و مشاهده بخش Module Information

روش دوم در فصل‌های بعدی توضیح داده شده است. در اینجا بحث را با استفاده از روش اول تشریح می‌کنیم. در مشخصات فنی CPU بخشی با عنوان Address Area(I/O) وجود دارد که تعداد ورودی و خروجی در آن مشخص است. در جدول ۱۱-۵ این مشخصه برای سه نوع CPU مقایسه شده است. CPUهای انتخاب شده از نوع ضعیف(CPU312) ، متوسط(CPU315-2 DP) و قوی (CPU319-3) هستند.

جدول ۱۱-۵ مقایسه تعداد ورودی، خروجی‌های چند مدل CPU

	تعداد دیجیتال ورودی	تعداد دیجیتال خروجی	تعداد آنالوگ ورودی	تعداد آنالوگ خروجی
CPU 312	1024	1024	256	128
CPU315-2	2048	2048	1024	1024
CPU319-3	65535	65535	4096	4096

۱۱-۴-۳-۲ تعداد تایمر و کانتر و متغیرهای حافظه

تعداد تایمر و کانتر و متغیرهای حافظه که در برنامه‌نویسی استفاده می‌شوند، در CPUهای مختلف متفاوت هستند. این موارد برای سه CPU فوق در جدول ۱۱-۶ مقایسه شده‌اند.

جدول ۱۱-۶ مقایسه تعداد کانترها، تایمرها و متغیرهای حافظه چند مدل CPU

	تعداد تایمر Timers	تعداد کانتر Counters	تعداد متغیر حافظه Bit Memory
CPU 312	128	128	1024
CPU315-2	256	256	2048
CPU319-3	2048	2048	8192

۱۱-۴-۳-۳ حجم حافظه

یکی از پارامترهای دیگر که برای CPUها مورد مقایسه قرار می‌گیرد حجم حافظه CPU است. ساختار حافظه و بخش‌های آن در فصل ۱۲ تشریح شده است، در اینجا فقط بخشی از حافظه که برنامه به آن دانلود می‌شود و در اصطلاح به آن Load Memory می‌گویند را بین سه CPU فوق مورد مقایسه قرار می‌دهیم.

جدول ۱۱-۷ مقایسه Load Memory چند مدل CPU

	حافظه داخلی Load Memory	قابل افزایش با کارت حافظهRAM	قابل افزایش با کارت حافظهFlash
CPU 312	ندارد	امکان‌پذیر نیست	4 MB
CPU315-2	ندارد	امکان‌پذیر نیست	8 MB
CPU319-3	ندارد	2 MB	8 MB

همانطور که دیده می‌شود در برخی CPU ها فقط می توان کارت حافظه Flash اضافه کرد.

۱۱-۴-۳-۴ سرعت پردازش

سرعت پردازش CPU بر حسب تعداد دستور در میلی‌ثانیه معرفی می‌شود. جدول۱۱-۸ این پارامتر را برای سه CPU فوق مورد مقایسه قرار داده است.

جدول ۱۱-۸ مقایسه سرعت پردازش سه مدل CPU

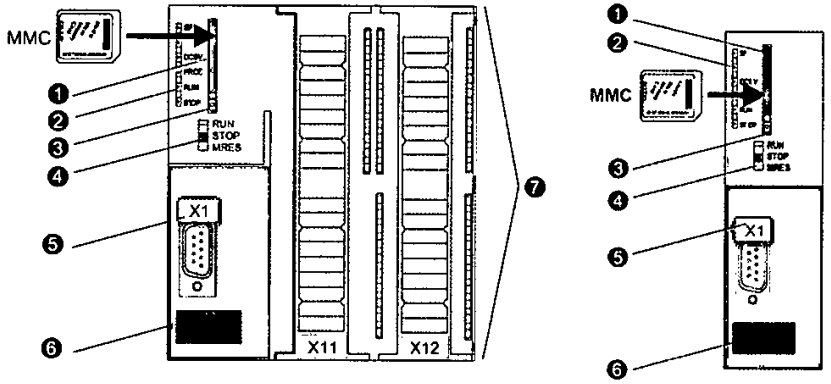
	سرعت پردازش
CPU 314	0.3 ms/1000 instructions
CPU316-2	0.3 ms/1000 instructions;
CPU318-2	0.1ms/1000 instructions

۱۱-۴-۳-۵ مقایسه CPUها از نظر وضعیت ظاهری

CPU های خانواده S7-300 از نظر ظاهری با هم تفاوت دارند. منظور از شرایط ظاهری موارد زیر است:

- عرض CPU
- لامپ‌های روی CPU
- کلید سلکتوری روی CPU
- پورت های روی CPU
- اسلات کارت حافظه روی CPU

در شکل ۱۱-۱۷ وضعیت ظاهری دو CPU ماژولار و یکپارچه مقایسه شده است.

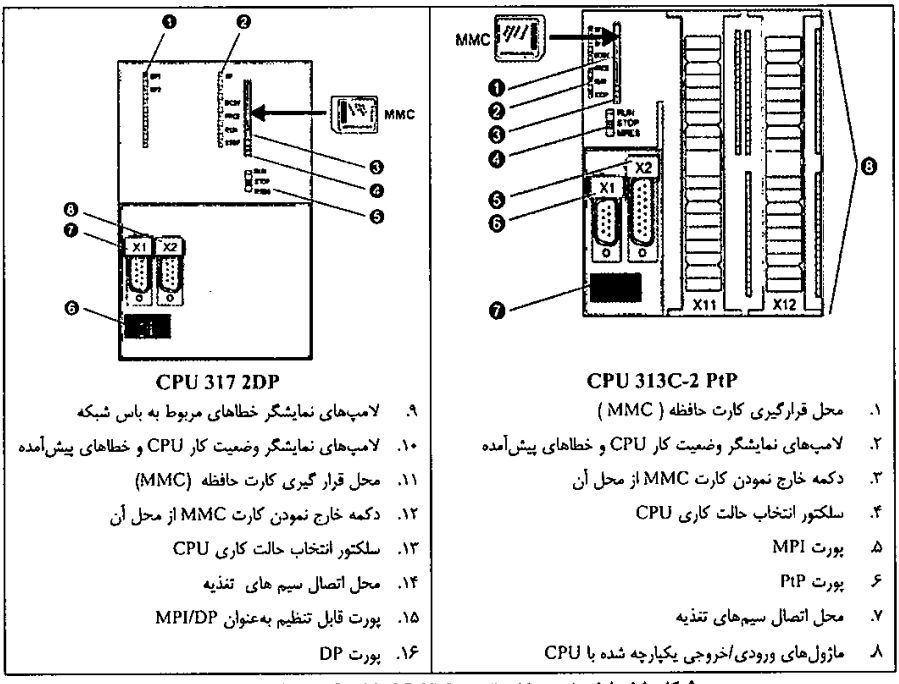


شکل ۱۱-۱۷ مقایسه شکل ظاهری CPU 313 با CPU 313c

قسمت‌های شماره‌گذاری شده عبارتند از:

شماره	توضیح
۱	محل قرارگیری کارت حافظه (MMC)
۲	لامپ‌های نمایشگر وضعیت کار CPU و خطاهای پیش‌آمده
۳	دکمه خارج نمودن کارت MMC از محل آن
۴	سلکتور انتخاب حالت کاری CPU (STOP/RUN/MRES)
۵	پورت MPI به منظور ارتباط با شبکه MPI و وسایل برنامه‌ریزی PLC مانند PC و لپ‌تاپ
۶	محل اتصال سیم‌های تغذیه
۷	ماژول‌های ورودی/خروجی یکپارچه شده با CPU

در شکل ۱۱-۱۸ وضعیت ظاهری دو CPU ماژولار و یکپارچه که علاوه بر پورت MPI دارای پورت DP نیز می‌باشند مقایسه شده است.



CPU 313C-2 PtP

۱. محل قرارگیری کارت حافظه (MMC)
۲. لامپ‌های نمایشگر وضعیت کار CPU و خطاهای پیش‌آمده
۳. دکمه خارج نمودن کارت MMC از محل آن
۴. سلکتور انتخاب حالت کاری CPU
۵. پورت MPI
۶. پورت PtP
۷. محل اتصال سیم‌های تغذیه
۸. ماژول‌های ورودی/خروجی یکپارچه شده با CPU

CPU 317 2DP

۹. لامپ‌های نمایشگر خطاهای مربوط به باس شبکه
۱۰. لامپ‌های نمایشگر وضعیت کار CPU و خطاهای پیش‌آمده
۱۱. محل قرار گیری کارت حافظه (MMC)
۱۲. دکمه خارج نمودن کارت MMC از محل آن
۱۳. سلکتور انتخاب حالت کاری CPU
۱۴. محل اتصال سیم های تغذیه
۱۵. پورت قابل تنظیم به‌عنوان MPI/DP
۱۶. پورت DP

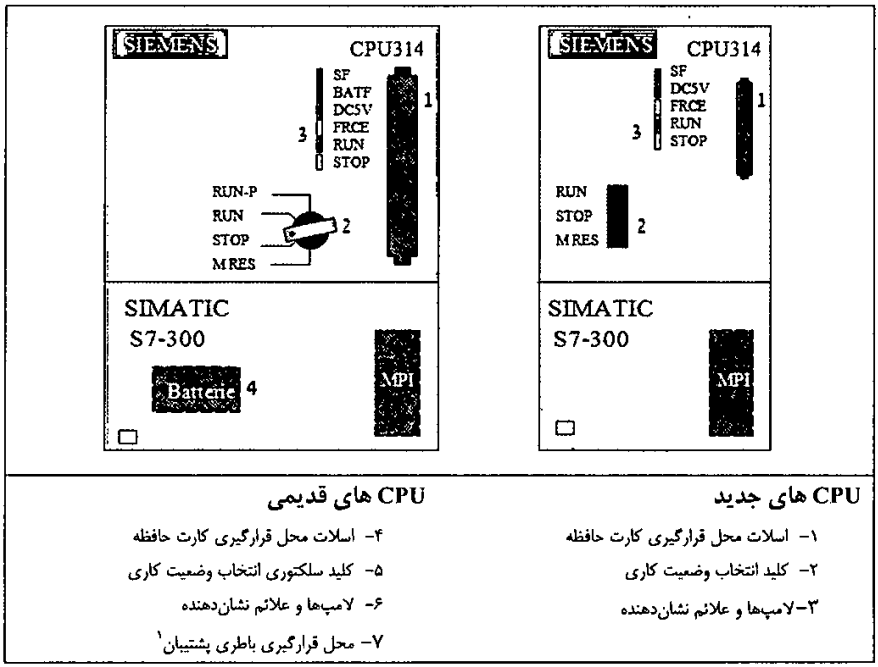
شکل ۱۱-۱۸ مقایسه شکل ظاهری CPU 313C-2PtP با CPU 317 2DP

۱۱-۴-۳-۶ مقایسه CPUهای قدیمی و جدید S7-300

CPU های جدید خانواده S7-300 ظاهری متفاوت با نوع قدیمی دارند. عمده تفاوت CPUهای جدید با CPUهای قدیمی عبارتست از:

- **عرض کوچکتر:** به‌دلیل تغییر در سخت‌افزار CPUهای جدید، این CPUها عرض کمتری نسبت به نوع قدیمی دارند.
- **حذف باطری پشتیبان:** در نوع قدیمی یک باطری کوچک روی CPU که در شکل ۱۱-۱۹ با شماره ۴ مشخص شده نصب می‌شد، ولی در CPUهای جدید این باتری حذف شده است. علت آن این است که در CPUهای قدیمی حافظه Load Memory از نوع RAM بود که به‌منظور حفظ اطلاعات نیاز به حفاظت در شرایط قطع تغذیه داشت، ولی در نوع جدید Load Memory به‌صورت کارت حافظه از جنس Flash EPROM است که با قطع تغذیه اطلاعات آن پاک نمی‌شود.

- **اسلات قرارگیری کارت حافظه:** در CPUهای قدیمی از کارت‌های حافظه که از نظر ظاهر بزرگتر هستند و به MC موسوم می‌باشند استفاده می‌شد، ولی در CPUهای جدید کارت‌های حافظه MMC که کوچک هستند به‌کار می‌رود. در CPUهای قدیمی استفاده از کارت حافظه الزامی نیست ولی CPUهای جدید بدون کارت حافظه (MMC) کار نمی‌کنند و MMC در واقع Load Memory محسوب می‌شود.
- **تغییر شکل کلید انتخاب وضعیت کاری:** سلکتور سوئیچ روی CPUهای قدیمی به‌صورت چرخشی است که می‌توان آنرا در چهار وضعیت Stop ، RUN ، RUN-P و MRES قرار داد. این کلید در CPUهای جدید به‌صورت عمودی جابه‌جا می‌شود و سه وضعیت Stop ، RUN و MRES دارد. در واقع RUN در CPUهای جدید همان RUN-P است. توضیحات کامل در مورد این وضعیت‌ها در فصل بعد آمده است.



CPU های جدید

۱- اسلات محل قرارگیری کارت حافظه
۲- کلید انتخاب وضعیت کاری
۳-لامپ‌ها و علائم نشان‌دهنده

CPU های قدیمی

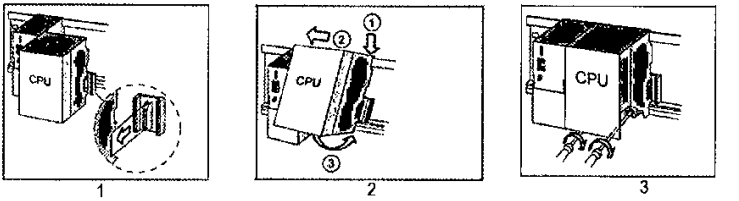
۴- اسلات محل قرارگیری کارت حافظه
۵- کلید سلکتوری انتخاب وضعیت کاری
۶- لامپ‌ها و علائم نشان‌دهنده
۷- محل قرارگیری باطری پشتیبان[1]

شکل ۱۱-۱۹ مقایسه شکل ظاهری CPUهای قدیمی و جدید S7-300

۱۱-۴-۴ نصب و سیم کشی CPU

به منظور نصب CPU در رک، می‌توان آنرا مانند شکل ۱۱-۲۰ روی رک نصب نمود. لازم است ابتدا کانکتور پشتی را نصب نموده، سپس CPU را در کنار منبع تغذیه قرار داده و پیچ‌های زیر آنرا در شیار پایین رک محکم کنیم.

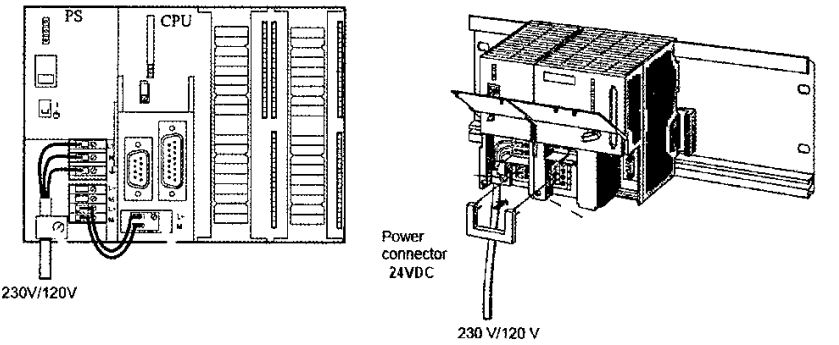
1. Back up



شکل ۱۱-۲۰ مراحل نصب CPU در رک

اتصال تغذیه CPU

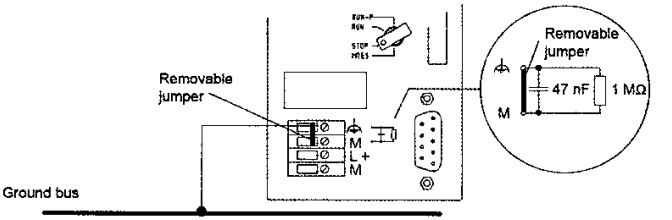
همه‌ی CPU ها دارای ترمینال‌هایی به منظور اتصال ولتاژ تغذیه می‌باشند. ولتاژ تغذیه را می‌توان از منبع تغذیه توسط کانکتور مخصوص زیمنس که به‌صورت U شکل است یا هر نوع کانکتور دیگری (حتی سیم مسی معمولی) به CPU منتقل نمود. شکل ۱۱-۲۱ طریقه اتصال ولتاژ تغذیه را نشان می‌دهد.



اتصال تغذیه با کانکتور U شکل **اتصال تغذیه به‌صورت معمولی**

شکل ۱۱-۲۱ اتصال ولتاژ تغذیه از PS به CPU

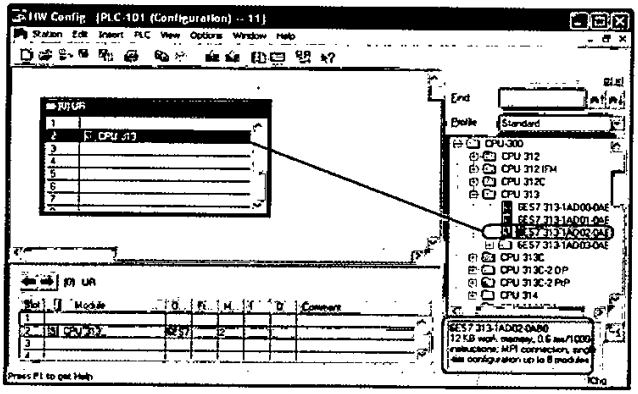
در برخی از CPUها در محل اتصال تغذیه به CPU جامپری وجود دارد که پایه M و زمین را به هم متصل کرده است. اگر این جامپر برداشته شود بین M و زمین یک مدار RC قرار می‌گیرد. در برخی کاربردها مانند نیروگاهها که نیاز به آشکار سازی خطای اتصال زمین است یا امکان عبور جریان زیاد به زمین وجود دارد می توان نقطه M را از زمین توسط مدار RC فوق جدا نمود.



شکل ۱۱-۲۲ مدار RC در محل اتصال زمین

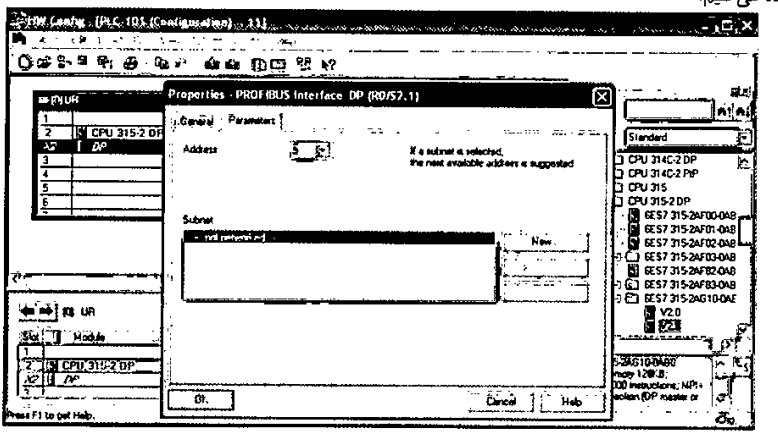
۱۱-۴-۵ پیکربندی CPU در محیط HW Config

جهت وارد نمودن CPU در رک، در قسمت کاتالوگ‌های برنامه‌ی HW Config و زیرشاخه SIMATIC300 به پوشه‌ی CPU-300 رفته و در آنجا می‌توان CPU مورد نظر را انتخاب کرده و آنرا مطابق شکل ۱۱-۲۳ در رک قرار داد.



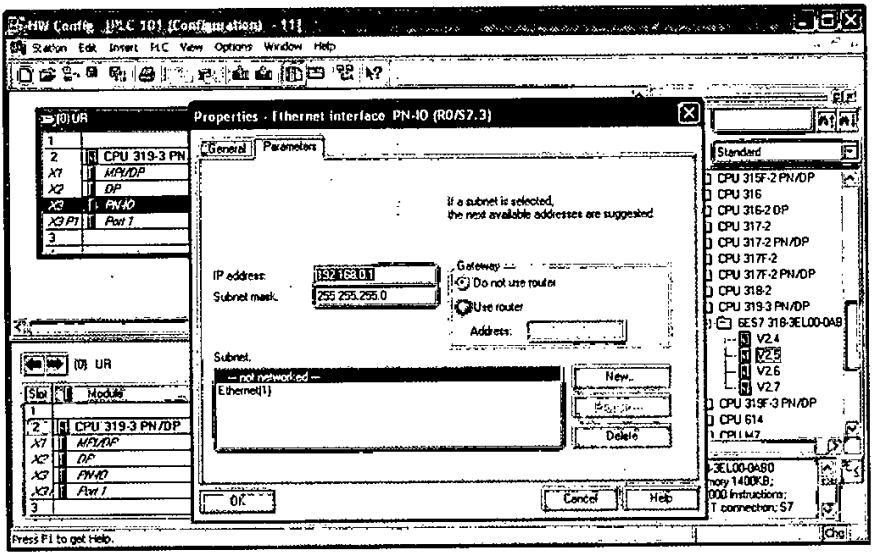
شکل ۱۱-۲۳ انتخاب CPU و وارد نمودن در رک

اگر CPU از نوع معمولی باشد که فقط پورت MPI دارد، شکل پیکربندی شبیه فوق خواهد بود ولی اگر CPU دارای پورت DP باشد پس از وارد کردن CPU پنجره شکل ۱۱-۲۴ ظاهر می‌شود که در مورد تنظیمات پروفی‌باس سوال می‌کند. اگر این پنجره را بدون انجام تنظیم خاصی Ok یا Cancel کنیم یک زیر اسلات در زیر CPU با عنوان DP مشاهده می‌کنیم.



شکل ۱۱-۲۴ انتخاب CPU که دارای پورت پروفی‌باس است.

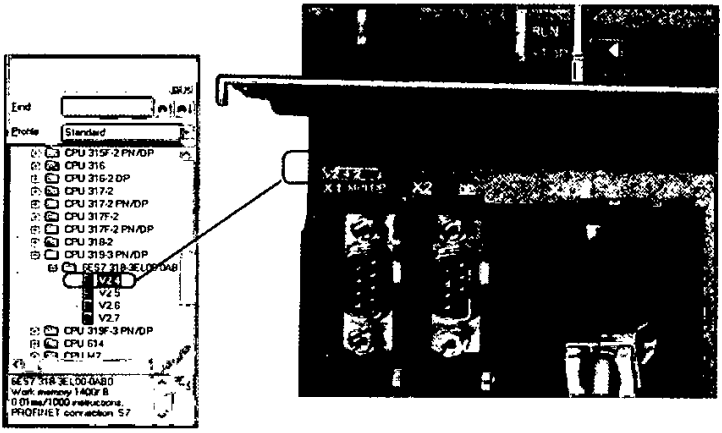
اگر CPU دارای پورت PN باشد، پس از وارد کردن CPU پنجره شکل ۱۱-۲۵ ظاهر می‌شود که در مورد تنظیمات پروفی‌نت (آدرس اترنت) سوال می‌کند. در این حالت نیز اگر روی OK یا Cancel کلیک کنیم در زیر CPU گزینه‌های دیگری که مربوط به PN است را مشاهده می‌کنیم.



شکل ۱۱-۲۵ انتخاب CPU که دارای پورت پروفی نت است

تذکر ۱: در لیست کاتالوگ برای یک CPU انواع مختلفی با شماره سفارش‌های متفاوت دیده می‌شود. معمولاً در هنگام سفارش فقط آخرین مدل را می‌توان از سازنده خریداری کرد، ولی مدل‌های قدیمی نیز در لیست وجود دارند تا بتوان سیستم‌های قدیمی را با آن پیکربندی نمود.

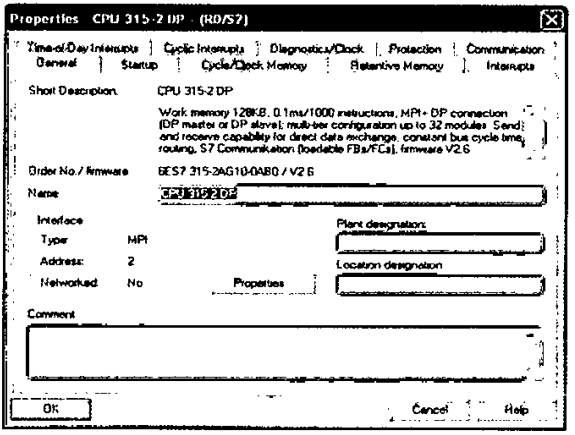
تذکر ۲: برای یک مدل CPU با یک Order Number خاص در لیست ممکن است Versionهای مختلفی دیده شود که ورژن سیستم عامل CPU یا در اصطلاح Firmware را نشان می‌دهد. بهتر است ورژنی انتخاب شود که با ورژن نوشته شده روی در CPU مطابقت دارد. شکل ۱۱-۲۶ این موضوع را نشان می‌دهد. در صورت عدم تطبیق ورژن، مشکلی پیش نمی‌آید و در عین حال می‌توان به روشی که در آینده تشریح می گردد ورژن را ارتقا داد.



شکل ۱۱-۲۶ انتخاب ورژن CPU با توجه به سخت‌افزار

تنظیم پارامترهای CPU

اگر در محیط hwconfig روی CPU دوبار کلیک کنیم پنجره‌ای مانند شکل ۱۱-۲۷ باز می‌شود که سربرگ‌های مختلفی در آن وجود دارد. در هر یک از قسمت‌ها گزینه‌های مختلفی وجود دارد که در برخی حالات لازم است تنظیمات آنها را تغییر داد. در اینجا که هنوز خواننده با برخی مفاهیم کلی آشنا نشده است، امکان تشریح این پارامترها وجود ندارد. در فصل‌های بعد بسته به کاربرد وارد جزئیات این سربرگ‌ها شده و نکات مربوط به آنها را تشریح می‌کنیم. سربرگ‌هایی که در آنها کلمه Interrupt یا وقفه آمده است در کتاب سطح پیشرفته مورد بحث قرار می‌گیرند.



شکل ۱۱-۲۷ پنجره Properties مربوط به CPU 315-2DP

۱۱-۵ کارت‌های ورودی دیجیتال

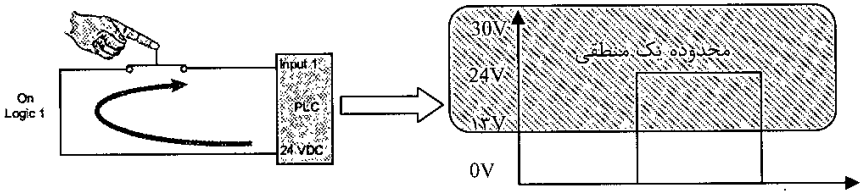
۱۱-۵-۱ عملکرد کارت دیجیتال ورودی

کارت دیجیتال ورودی سیگنال الکتریکی دریافتی را به دیتای منطقی صفر یا یک تبدیل می‌کند. یکی از مشخصه‌های مهم در کارت‌های دیجیتال ورودی، محدوده‌ی صفر و یک منطقی می‌باشد؛ روش کار به این صورت است که برای هر کارت با توجه به ولتاژ ورودی کارت، محدوده‌ای جهت صفر منطقی و محدوده‌ی دیگری جهت یک منطقی در نظر گرفته می‌شود. اگر سیگنال ارسالی به کارت، در محدوده‌ی تعیین شده جهت صفر منطقی باشد، کارت آنرا به‌عنوان دیتای صفر منطقی (حالت قطع) در نظر می‌گیرد. اگر سیگنال ارسالی به کارت، در محدوده‌ی تعیین شده جهت یک منطقی باشد، کارت آنرا به‌عنوان دیتای یک منطقی (حالت وصل) در نظر می‌گیرد. محدوده‌ی صفر و یک منطقی هر کارت را می‌توان در مشخصات فنی آن کارت مشاهده نمود. به‌عنوان مثال این مشخصات جهت کارت DI 32X DC 24V به‌صورت شکل ۱۱-۲۸ می‌باشد.

Data for Selecting a Sensor	
Input voltage	
• Rated value	24 VDC
• For signal "1"	13 to 30 V
• For signal "0"	–30 to + 5 V
Input current	
• At signal "1"	typ. 7 mA
Input delay	
• At "0" to "1"	1.2 to 4.8 ms
• At "1" to "0"	1.2 to 4.8 ms
Input characteristic curve	According to IEC 61131, type 1
Connection of Two-Wire BEROs	Possible
• Permitted bias current	max. 1.5 mA

شکل ۱۱-۲۸ مشخصات فنی کارت DI 32XDC 24V

همانطور که مشخص است، که ولتاژ محدوده 13V الی 30V به‌عنوان یک منطقی در نظر گرفته می‌شود. این موضوع در شکل ۱۱-۲۹ نیز نمایش داده شده است.



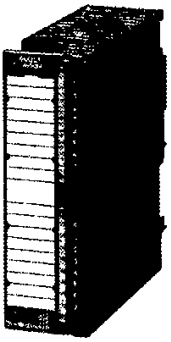
شکل ۱۱-۲۹

۱۱-۵-۲ انواع کارت‌های DI

هرکارت دیجیتال به‌صورتی کدگذاری شده که از روی آن می‌توان نوع آن را تشخیص داد. روش کد گذاری طبق فرمت زیر است.

معرف دیجیتال خروجی	معرف تعداد خروجی		معرف ولتاژ خروجی		معرف جریان خروجی	معرف نوع خروجی	معرف ویژگی خاص (در برخی کارت ها)
SM 322 DO	4	x	DC 24 V	/	0.5 A	Relay	-
	8		DC 15V		1 A	Switch	EX
	16		UC 24 / 48 V		1.5A		
	32		48-125 V DC		2 A		
	64		AC 120 V		8 A		
			AC 230V		0.3 A		
			AC 120/230 V		20 mA		
			AC Iso		10 mA		

به‌عنوان مثال کارت DI 16xDC24V را در نظر بگیرید. این کارت دارای ۱۶ ترمینال ورودی بوده و ولتاژ ورودی آن 24V DC می‌باشد. به شکل ۱۱-۳۰ توجه کنید.



شکل ۱۱-۳۰ کارت DI 16xDC24V

به‌طور کلی با توجه به سه مشخصه این کارت‌ها را می‌توان تقسیم‌بندی نمود.

- **تعداد ترمینال‌های ورودی:** ترمینال‌های ورودی می‌تواند ۴، ۸، ۱۶، ۳۲، یا ۶۴ عدد باشد.
- **ولتاژ ورودی:** که به‌صورت DC یا AC یا UC که معرف Universal است یعنی هم برای DC و هم برای AC قابل استفاده می‌باشد.
- **قابلیت‌های خاص:** جزئیات این تقسیم‌بندی‌ها در ادامه تشریح شده است.

۱۱-۵-۳ مشخصات فنی کارت‌های DI

برخی از مهم‌ترین پارامترهای مشخصه کارت دیجیتال ورودی در جدول ۱۱-۹ لیست شده و مقادیر آنها برای سه نوع کارت مختلف آورده شده است.

جدول ۱۱-۹ مقایسه پارامترهای مختلف کارت‌های DI

	DI 32 x 24 VDC	DI 16 x 120 VAC	SM 321; DI 8 x 120/230 VAC
تعداد کانال	32	16	8
طول کابل قابل اتصال	بدون شیلد : 600 m شیلد دار : 1000 m	بدون شیلد: 600 m شیلد دار: 1000 m	بدون شیلد : 600 m شیلد دار : 1000 m
رنج ولتاژ مجاز برای سیگنال 1	13 to 30 V	79 to 132 V	79 to 264 V
رنج ولتاژ مجاز برای سیگنال 0	– 3 to 5 V	0 to 20 V	0 to 40 V
رنج فرکانس ورودی AC	-	47 to 63 Hz	47 to 63 Hz
جریان مصرفی کارت (جذب شده از باس)	max 15 mA	max. 16 mA	max. 29 mA
جریان هر ورودی	typical 7 mA	typ. 6 mA	120 V, 60 Hz : typ. 6.5 mA 230 V, 50 Hz : typ. 11 mA
تاخیر ورودی	1.2 to 4.8 ms	max. 25 ms	max. 25 ms
توان تلف شده در کارت	6.5 W	typ. 4.1 W	typ. 4.9 W
نوع کانکتور جلویی	40 pin	20 pin	20 pin

منظور از کانکتور جلویی[1] کانکتوری است که سیم‌کشی به آن انجام شده سپس در جلوی کارت نصب می‌گردد. این کانکتور دارای دو نوع ۴۰ پین و ۲۰ پین است که برای تمام انواع کارت‌های دیجیتال و آنالوگ یکی از این دو استفاده می‌شود.

ب) 20 Pin

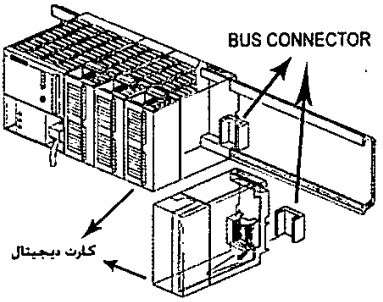
الف) 40 Pin

شکل ۱۱-۳۱ فرانت کانکتور

1. Front Connector

۱۱-۵-۴ روش نصب کارت DI در رک

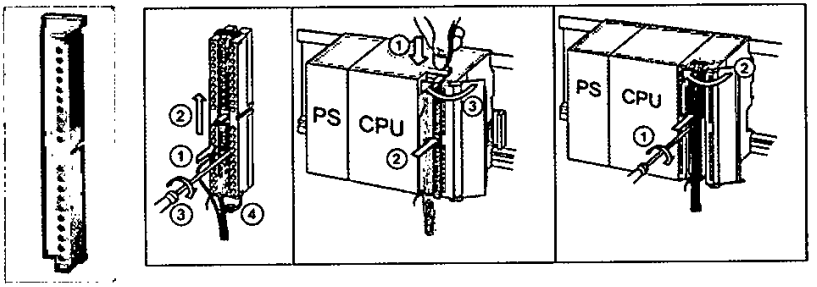
روش نصب کارت‌های DI در رک، همانند نصب CPU در رک می‌باشد و اتصال بین آن و سایر ماژول‌ها مطابق شکل ۱۱-۳۲ توسط BUS CONNECTOR برقرار می‌گردد.



شکل ۱۱-۳۲ نصب کارت DI در رک

نصب Front Connector

به‌منظور اتصال سیم‌های تغذیه و سیم‌های ورودی/خروجی در کارت‌های SM، از قطعه‌ای در روی کارت به نام **Front Connector** استفاده می‌گردد. شکل ۱۱-۳۳ نحوه‌ی قرارگیری **Front Connector** را در روی کارت نشان می‌دهد.

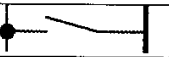


شکل ۱۱-۳۳ Front Connector و نصب آن بر روی کارت

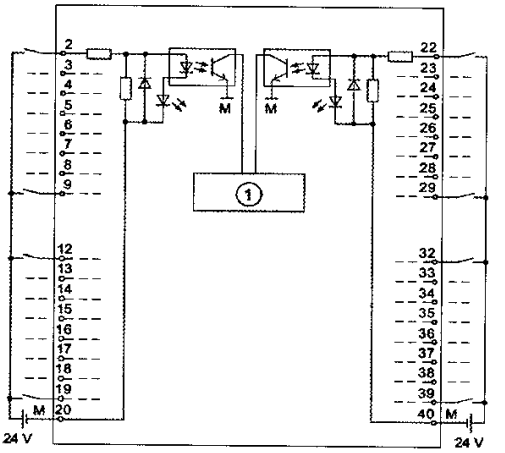
ترمینال‌های ورودی و اتصال تغذیه

در هر کارت دیجیتال ورودی، علاوه بر ترمینال‌هایی که به‌منظور اتصال سیگنال‌های ورودی به کارت در نظر گرفته شده است، تعدادی ترمینال دیگر نیز به منظور اتصال ولتاژ تغذیه‌ی کارت وجود دارد. معمولاً پشت درب کارت، نقشه‌ی اتصالات صحیح ورودی‌ها و تغذیه درج شده است. علائم به‌کار رفته پشت درب ماژول‌های DI و معانی آنها در جدول ۱۱-۱۰ نشان داده شده است.

جدول ۱۱-۱۰ علائم به‌کار رفته در ماژول DI

علامت به‌کار رفته روی درب ماژول DI	مفهوم
	ترمینال مربوط به سیگنال ورودی دیجیتال
L+	ولتاژ مثبت تغذیه‌ی کارت
M	ولتاژ منفی تغذیه‌ی کارت

البته ممکن است در بعضی کارت‌ها نیاز به اتصال تغذیه‌ی مثبت یا منفی وجود نداشته باشد، در اینصورت در پشت درب ماژول از این علائم استفاده نمی‌گردد. شکل ۱۱-۳۴ نقشه‌ی اتصالات و مدار داخلی کارت DI 32 x DC 24V را نشان می‌دهد.



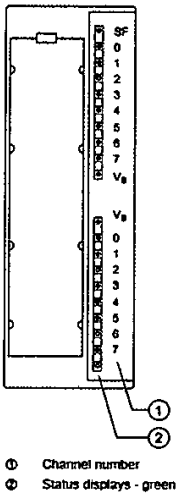
① Backplane bus interface

شکل ۱۱-۳۴ نقشه‌ی اتصالات و مدار داخلی DI 16X DC 24V

لامپ‌های نشان‌دهنده وضعیت و خطا

همانطور که در شکل ۱۱-۳۵ مشخص است، در روی درب ماژول شماره کانال‌ها و لامپ‌های نشان‌دهنده وضعیت هر کانال قرار گرفته است. لامپ وضعیت هر کانال در زمان فعال بودن آن کانال روشن شده و در زمان غیرفعال بودن کانال خاموش می‌شود. در کارت‌هایی که قابلیت تشخیص خطا را دارا هستند، لامپ دیگری که نشان‌دهنده وجود خطا می‌باشد در کنار لامپ وضعیت قرار گرفته است. در این کارت‌ها در زمان وجود خطا برای هر کانال، رنگ لامپ وضعیت آن کانال به رنگ قرمز درآمده و لامپ مربوط به خطای ماژول نیز با رنگ قرمز روشن می‌شود. در صورت برطرف‌شدن خطا، هر دولامپ

ذکر شده خاموش می‌شوند. از آنجایی که شکل ۱۱-۳۵ مربوط به یک کارت DI که دارای منبع تغذیه‌ی داخلی (Vs) می‌باشد، در روی درب آن دو لامپ دیگر به نام‌های Vs قرار گرفته‌اند. در صورتی‌که این منابع تغذیه در شرایط نرمال باشند، این لامپ‌ها با رنگ سبز روشن می‌شوند.

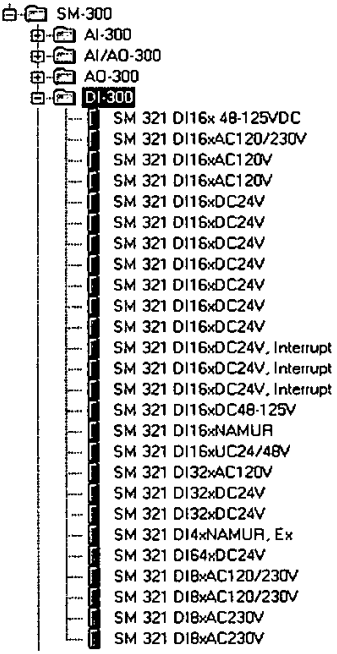


شکل ۱۱-۳۵ لامپ‌ها و علائم روی درب کارت‌های دیجیتال

اگر درب کارت باز شود، پشت آن نقشه‌ای سیم‌کشی رسم شده که راهنمای کاربر برای انجام اتصالات تغذیه و ورودی‌ها می‌باشد.

۱۱-۵-۵ پیکربندی کارت DI در Hwconfig

به‌منظورقرار دادن کارت DI در رک، می‌توان مطابق شکل ۱۱-۳۶ به کاتالوگ ماژول‌های موجود در برنامه HW Config مراجعه نموده و در قسمت Station 300 و در زیرشاخه SM-300 گزینه DI-300 را انتخاب نمود. در اینصورت انواع کارت‌های DI 300 نمایش داده شده و می‌توان کارت مورد نظر را انتخاب و آنرا در رک قرار داد.



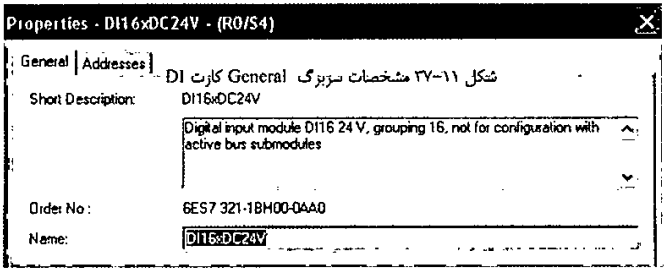
شکل ۱۱-۳۶ انواع کارت‌های DI-300

Properties کارت DI

هنگامی‌که کارت DI در رک قرار گرفت، می‌توان با دوبار کلیک نمودن بر روی آن به قسمت تنظیم مشخصات کارت دسترسی پیدا نمود. در کارت‌های معمولی در این قسمت دو سربرگ وجود دارد.

سربرگ General

در این سربرگ مشخصات مختصری در مورد ماژول به همراه Order Number آن ذکر شده است. محتویات این سربرگ در شکل ۱۱-۳۷ نشان داده شده است.

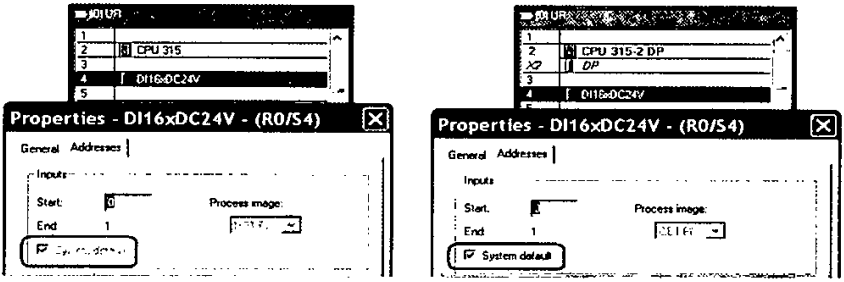


شکل ۱۱-۳۷ مشخصات سربرگ General کارت DI

شکل ۱۱-۳۷

سربرگ Addresses

وقتی کارت به اسلات رک در HW Config وارد شود، برنامه به‌طور خودکار به آن آدرس اختصاص می‌دهد. این آدرس در برنامه‌نویسی مورد نیاز خواهد بود. در این سربرگ می‌توان آدرس اختصاص یافته به کارت را مشاهده نمود. در برخی شرایط گزینه System default را می‌توان غیرفعال کرده و آدرس سیستم را به آدرس دلخواه تغییر داد. ولی این ویژگی همیشه وجود ندارد، در واقع این ویژگی مربوط به کارت نیست بلکه مربوط به CPU می‌باشد. به‌عنوان مثال وقتی کارت DI 16 x DC24V در کنار CPU 315 قدیمی با کد 315-1AF00-0AB0 قرار می‌گیرد گزینه تغییر آدرس مانند شکل زیر غیرفعال و وقتی همین کارت در کنار CPU 315-2DP قرار می‌گیرد، گزینه مزبور فعال است.



کارت DI در کنار CPU315-2DP　　کارت DI در کنار CPU315

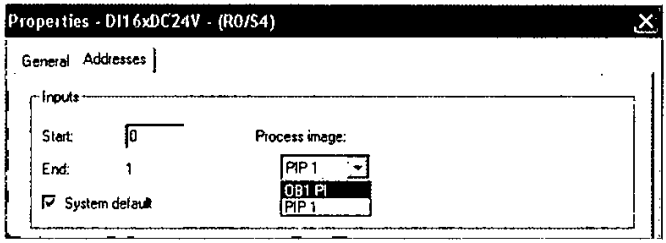
شکل ۱۱-۳۸ مشخصات سربرگ Addresses کارت DI

توصیه می‌شود آدرس‌های پیش‌فرض سیستم را تغییر ندهید، مگر آنکه ضرورت ایجاب کند که حافظه به‌صورت بهینه آدرس‌دهی گردد.

جزئیات نحوه آدرس دهی کانال‌های کارت DI در فصل‌های بعد آمده است.

همانطور که در شکل ۱۱-۳۸ دیده می‌شود، در سربرگ آدرس گزینه دیگری با عنوان Process Image وجود دارد که غیر فعال است. اطلاعات ورودی در حافظه CPU در بخشی با عنوان Process image قرار می‌گیرد. در این قسمت معمولاً گزینه‌ی OB1 PI انتخاب شده است. در اکثر CPUهای S7-300، ناحیه‌ی Process image به‌صورت یکپارچه می‌باشد و فقط یک بخش با عنوان OB1 PI وجود دارد، از اینرو این گزینه در این CPUها غیرقابل تغییر است.

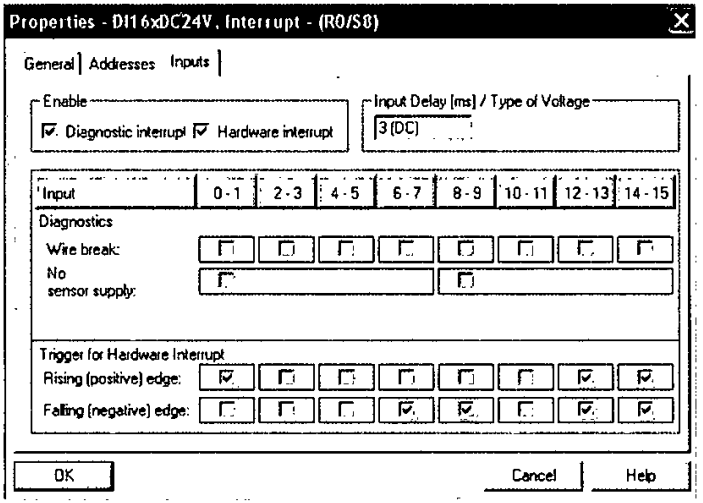
در برخی از CPUهای S7-300 مانند CPU 319-3 PN/DP، ناحیه‌ی Process image به‌صورت پارتیشن‌بندی شده وجود دارد. یعنی اینکه این ناحیه به قسمت‌های متعددی تقسیم‌بندی شده است که هر قسمت را می‌توان به‌صورت جداگانه Update نمود. در صورتی‌که CPU از این قابلیت برخوردار باشد، مطابق شکل ۱۱-۳۹ در تنظیمات کارت و در قسمت Process image می‌توان یکی از قسمت‌های Process image را برای کارت انتخاب نمود؛ ولی معمولاً لزومی ندارد که این گزینه از حالت پیش‌فرض خارج شود.



شکل ۱۱-۳۹ تنظیم ناحیه‌ی Process image مربوط به کارت DI

۱۱-۵-۶ قابلیت‌های ویژه در کارت‌های DI

در بین کارت‌های DI موجود در S7-300، برخی دارای قابلیت‌های ویژه‌ای می‌باشند. این کارت‌ها در کاتالوگ موجود در برنامه‌ی HW Config با مشخصه Interrupt در انتهای اسم آنها مشخص می‌گردند. این کارت‌ها همانطور که در شکل ۱۱-۴۰ مشاهده می‌شود، در پیکربندی دارای یک سربرگ Input هستند که در این بخش کلمه Interrupt یا وقفه را می‌بینیم. بحث وقفه[1] در کتاب سطح پیشرفته به‌طور مفصل بحث شده است. در اینجا فقط همین اندازه اشاره می‌شود که با استفاده از وقفه Hardware Interrupt می‌توان CPU را قادر ساخت تا به تغییرات ورودی به‌سرعت عکس‌العمل نشان دهد و با استفاده از وقفه Diagnostics Interrupt می‌توان CPU را قادر ساخت تا نسبت به اشکالاتی مانند قطع تغذیه کارت یا قطعی سیم یا اتصال کوتاه ورودی عکس العمل نشان داده و برنامه خاصی را اجرا کند.



شکل ۱۱-۴۰ سربرگ Input درکارت DI16xDC24V, Interrupt

1. Interrupt

Input Delay (ms) / Type of Voltage

در این قسمت می‌توان تعیین نمود که سیگنال‌های ورودی با چه میزان تأخیر دریافت شوند. تأخیر اعمالی بر حسب میلی‌ثانیه است. در موارد زیر توصیه می گردد مقدار زیاد را برای تأخیر انتخاب کنید:

- سوئیچ‌های ساده و بدون حفاظت که ممکن است در شرایط وصل، کنتاکت آنها قطع و وصل شود.
- اگر طول کابل تا سنسور زیاد بوده و کابل از نوع بدون شیلد باشد.

۱۱-۵-۷ کارت‌های DI خاص در S7-300

برخی از کارت‌های DI در S7-300 به‌گونه‌ای طراحی شده‌اند که می‌توانند سیگنال ارسالی از سنسورهای NAMUR را دریافت نمایند. مشخصه‌ی این کارت‌ها کلمه‌ی NAMUR بعد از اسم کارت می‌باشد.

سنسورهای NAMUR به‌گونه‌ای هستند که با اتصال آنها به PLC سه وضعیت زیر قابل شناسایی است:

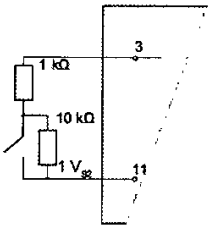
- On بودن سنسور
- Off بودن سنسور
- قطعی سیم سنسور

وضعیت سوم در سنسورهای عادی قابل شناسایی نیست. در نوع NAMUR برای ایجاد امکان فوق از مقاومت‌های سری و موازی همراه با سنسور استفاده می‌گردد و کارت دیجیتال مقدار مقاومت را بررسی می‌کند. به‌عنوان مثال شکل ۱۱-۴۱ را در نظر بگیرید، با توجه به این شکل داریم:

On بودن سنسور: مقاومت دیده شده $1\ K\ \Omega$ است.

Off بودن سنسور: مقاومت دیده شده $11\ K\Omega$ است.

قطعی سیم سنسور: مقاومت دیده شده بی‌نهایت است.



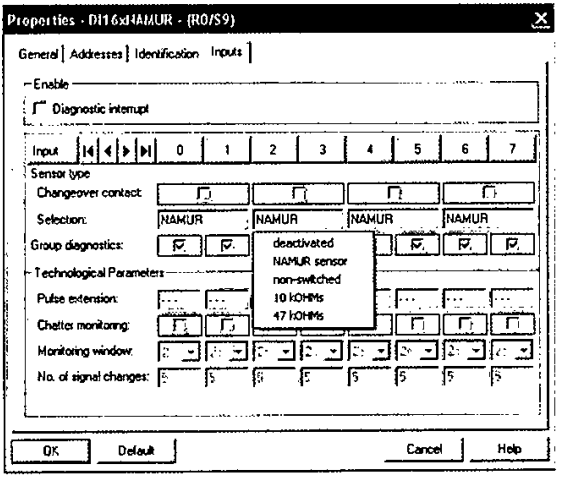
شکل ۱۱-۴۱ سنسور معمولی با اتصال طبق استاندارد NAMUR

تا زمان نگارش این کتاب، دو کارت NAMUR به شرح زیر در کاتالوگ برنامه‌ی HW Config موجود می‌باشد:

الف) DI 16xNAMUR

در این کارت قابلیت اتصال سنسورهای NAMUR، و همچنین سنسورهای با مقاومت مشخص وجود دارد. در شکل۱۱-۴۲ پنجره مشخصات این کارت نشان داده شده است. همانطور که مشاهده می‌شود، علاوه بر سربرگ‌های General و Addresses دو سربرگ دیگر نیز وجود دارد که عبارتند از:

- سربرگ Inputs
- سربرگ Identification



شکل ۱۱-۴۲ Properties کارت DI 16xNAMUR

بخش‌های اصلی در سربرگ Input از مشخصات کارت فوق عبارتند از:

Enable

در این بخش می‌توان قابلیت وقفه Diagnostic interrupt را فعال یا غیرفعال نمود. در صورت فعال نمودن این قابلیت، خطاهای زیر قابل شناسایی می‌باشند:

- اتصال کوتاه
- قطعی سیم
- خطای نوسان در سیگنال ورودی
- قطعی ولتاژ تغذیه‌ی کارت یا سنسور
- در حالت استفاده از گزینه‌ی Changeover contact، تشخیص خطاهای داخلی سنسور نظیر اتصال کوتاه در سنسور و . . .

تشریح وقفه فوق و تنظیمات دیگر این پنجره در کتاب سطح پیشرفته آمده است.

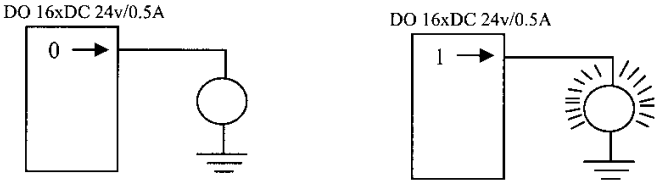
ب) DI 16xNAMUR EX

این کارت نیز قابلیت اتصال سنسورهای NAMUR را پشتیبانی می‌نماید. این کارت و سایر کارت‌هایی که دارای علامت EX هستند، کارت‌های قابل استفاده در محیط‌های خطرناک و انفجاری می‌باشند. مشخصات و تنظیمات این کارت، شبیه کارت DI 16xNAMUR بوده که در کتاب سطح پیشرفته در مورد آن بحث شده است.

۱۱-۶ کارت‌های خروجی دیجیتال DO

۱۱-۶-۱ عملکرد

در کارت دیجیتال خروجی، دیتای ارسالی از CPU به سیگنال الکتریکی تبدیل و به تجهیزات تحت کنترل ارسال می‌گردد. از آنجا که کارت‌های دیجیتال کارت‌های دو حالته محسوب می‌شوند، لذا تنها دارای دو حالت قطع و وصل می‌باشند. اگر دیتای یک منطقی به کانالی در کارت ارسال گردد، سیگنال خروجی کانال وصل می‌شود و اگر دیتای صفر منطقی به کارت ارسال گردد سیگنال خروجی کانال قطع می‌گردد. در شکل ۱۱-۴۳ نحوه اتصال مصرف‌کننده به کارت دیجیتال خروجی نشان داده شده است.



شکل ۱۱-۴۳ اتصال مصرف‌کننده به کارت دیجیتال

۱۱-۶-۲ انواع کارت DO

هر کارت دیجیتال خروجی به‌صورتی کدگذاری شده که از روی آن می‌توان نوع آن را تشخیص داد. روش کدگذاری طبق فرمت زیر است.

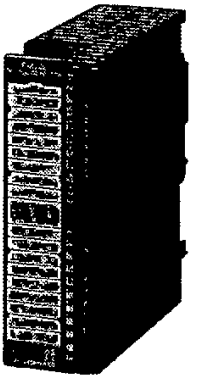
معرف دیجیتال خروجی	معرف تعداد خروجی		معرف ولتاژ خروجی	/	معرف جریان خروجی	معرف نوع خروجی	معرف ویژگی خاص (در پرخی کارت ها)
SM 322 DO	4	x	DC 24 V		0.5 A	Relay	-
	8		DC 15V		1 A	Switch	EX
	16		UC 24 / 48 V		1.5A		
	32		48-125 V DC		2 A		
	64		AC 120 V		8 A		
			AC 230V		0.3 A		
			AC 120/230 V		20 mA		
			AC Iso		10 mA		

همانطور که دیده می‌شود فاکتورهایی که براساس آنها کارت‌ها تقسیم‌بندی می‌شوند عبارتند از:

- تعداد کانال خروجی
- نوع ولتاژ خروجی که می‌تواند AC، DC و UC با سطح ولتاژهای مختلف باشد.
- میزان جریان خروجی هر کانال
- نوع خروجی که می‌تواند رله یا سوئیچینگ باشد.
- قابلیت‌های خاص نظیر استفاده در محیط‌های خطرناک

به‌عنوان مثال در کارت DO16xDC24V/0.5A مشخص است که:

- این کارت ۱۶ ترمینال خروجی دارد.
- ولتاژ خروجی کارت 24 v DC است.
- جریان مجاز هر کانال خروجی 0.5 A است.



شکل ۱۱-۴۴ کارت DO16xDC24V/0.5A

کارت‌های رله‌ای

برخی از کارت‌های DO دارای قابلیت جریان‌دهی بالایی هستند. مشخصه‌ی این کارت‌ها علامت Rel بعد از نام کارت می‌باشد. ولتاژ خروجی این کارت‌ها معمولاً 230V AC می‌باشد.

کارت‌های EX

برخی از کارت‌های دیجیتال خروجی به‌گونه‌ای طراحی شده‌اند که می‌توانند در محیط‌های خطرناک که احتمال انفجار وجود دارد کار نمایند. مشخصه‌ی این کارت‌ها علامت EX در انتهای نام کارت می‌باشد.

کارت DO16xUC24/48V

این کارت، یکی از کارت‌های DO بوده که می‌تواند هم خروجی AC و هم خروجی DC با ولتاژ 24 V یا 48 V به خروجی تحویل دهد.

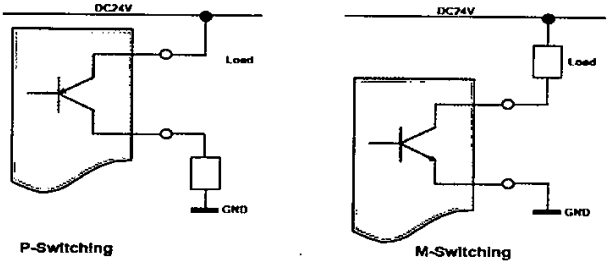
کارت DO64xDC24V/0.3A, switching

این کارت دارای ۶۴ ترمینال و ولتاژ خروجی 24V DC با جریان مجاز هر کانال 0.3 A می‌باشد. این کارت دارای دو نمونه زیر می‌باشد:

الف) DO64xDC24V/0.3A, switching M

ب) DO64xDC24V/0.3A, switching P

تفاوت نوع M و P در نحوه‌ی اتصال مصرف‌کننده به کارت می‌باشد. در شکل ۱۱-۴۵ نحوه اتصال مصرف کننده به کارت در نوع M و P نشان داده شده است (sink / source).



شکل ۱۱-۴۵ نحوه‌ی اتصالات در M و P

در اکثر کارت‌های DO نیاز به اتصالات فوق نمی‌باشد و ولتاژ مثبت از کانال کارت دریافت شده و ولتاژ منفی به‌طور مستقیم به مصرف‌کننده متصل می‌گردد.

۱۱-۶-۳ مشخصات فنی کارت‌های DO

برخی از مهمترین پارامترهای مشخصه کارت دیجیتال خروجی در جدول زیر لیست شده و مقادیر آنها برای سه نوع کارت مختلف آورده شده است.

جدول ۱۱-۱۱ برخی از پارامترهای کارت‌های دیجیتال خروجی

	DO 32 x 24 VDC / 0.5 A	DO 16 x 120 VAC /1A	SM 321; DO 8 x 120/230 VAC/2A
تعداد کانال	32	16	8
طول کابل قابل اتصال	بدون شیلد : 600 m شیلد دار : 1000 m	بدون شیلد : 600 m شیلد دار : 1000 m	بدون شیلد : 600 m شیلد دار : 1000 m
رنج جریان خروجی مجاز برای سیگنال 1	5 mA to 0.6 A	10 mA to 1 A	10 mA to 2 A
رنج جریان خروجی مجاز برای سیگنال 0	max. 0.5 mA	max. 1 mA	max. 2 mA
جریان مصرفی کارت (جذب شده از باس)	max 90 mA	max. 184 mA	max. 100 mA
جریان جذبی کارت از تغذیه ۲۴ ولت بیرونی در حالت بی بار	max. 200 mA	max. 3 mA	max. 2 mA
توان تلف شده در کارت	6.6 W	max. 9 W	typ. 8.6 W
نوع کانکتور جلویی	40 pin	20 pin	20 pin

۱۱-۶-۴ روش نصب و سیم‌کشی کارت DO

روش نصب کارت‌های DO در رک، همانند نصب کارت‌های DI می‌باشد. در این حالت نیز به منظور برقراری ارتباط این ماژول و سایر ماژول‌ها با CPU، از Backplane Bus استفاده می‌شود.

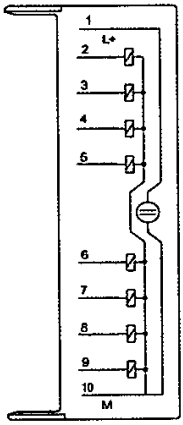
ترمینال‌های خروجی و اتصال تغذیه

در هر کارت دیجیتال خروجی، علاوه بر ترمینال‌هایی که به منظور اتصال مصرف‌کننده به کارت در نظر گرفته شده است، تعدادی ترمینال دیگر نیز به‌منظور اتصال ولتاژ تغذیه‌ی کارت وجود دارد. معمولاً پشت درب کارت، نقشه‌ی اتصالات صحیح خروجی‌ها و تغذیه درج شده است. علائم به‌کار رفته روی درب ماژول‌های DO و معانی آنها در جدول ۱۱-۱۲ نشان داده شده است.

جدول ۱۱-۱۲ علائم به‌کار رفته در ماژول DI

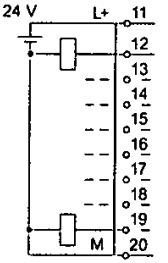
علامت به‌کار رفته روی درب ماژول DI	مفهوم
	ترمینال مربوط به سیگنال خروجی دیجیتال
L +	ولتاژ مثبت تغذیه‌ی کارت
M	ولتاژ منفی تغذیه‌ی کارت

شکل ۱۱-۴۶ نقشه‌ی سیم‌کشی و علائم پشت درب یک نمونه کارت DO را نشان می‌دهد.



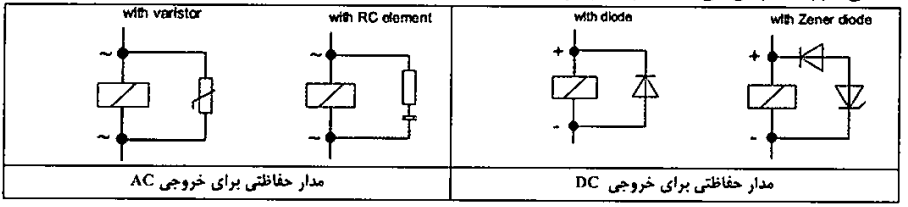
شکل ۱۱-۴۶ نقشه‌ی سیم‌کشی و علائم پشت درب یک نمونه کارت DO

در شکل ۱۱-۴۷ نقشه‌ی ترمینال‌های یک کارت دیجیتال خروجی معمولی نشان داده شده است. همانطور که در این شکل مشاهده می‌شود، ترمینال‌های 11و 12 مربوط به تغذیه‌ی کارت بوده و ولتاژ مثبت (+L) و منفی (M) به آنها متصل می‌شود. ترمینال‌های 12 تا 19 مربوط به خروجی بوده و مصرف‌کننده به آنها وصل می‌شود. مثلاً در این شکل، به ترمینال‌های شماره ۱۲ و ۱۹ مصرف‌کننده وصل شده است.



شکل ۱۱-۴۷ نقشه ترمینال‌های یک کارت دیجیتال خروجی معمولی

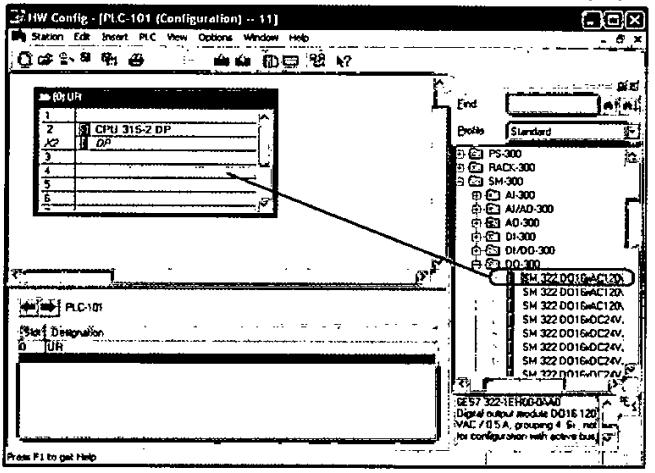
همانطور که در فصل ۴ توضیح داده شد، برای حفاظت کارت DO در مقابل ولتاژهای گذرای برگشتی از سمت بارهای القایی لازم است پیش‌بینی‌های لازم انجام شود. شکل زیر مدارهای حفاظتی مورد نیاز را یادآوری می‌کند.



شکل ۱۱-۴۸ مدارهای حفاظتی خروجی های دیجیتال

۱۱-۶-۵ پیکربندی کارت DO در Hwconfig

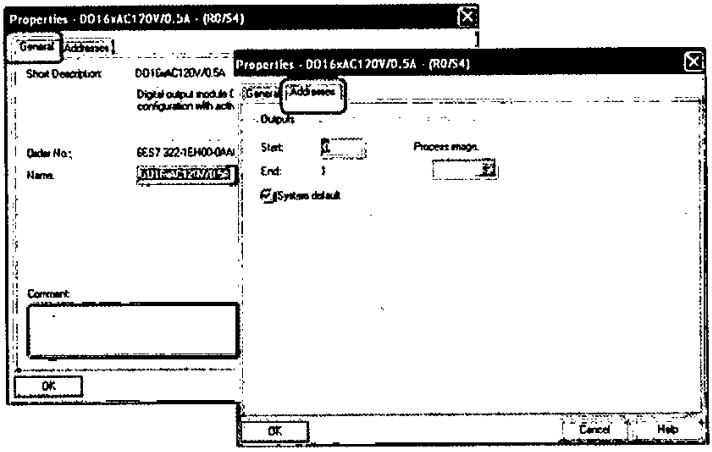
به‌منظور قراردادن کارت DO در رک، می‌توان مطابق شکل ۱۱-۴۹ به کاتالوگ ماژول‌های موجود در برنامه HW Config مراجعه نموده و در قسمت Station 300 و در زیرشاخه SM-300 گزینه DO-300 را انتخاب نمود.



شکل ۱۱-۴۹ وارد نمودن کارت DO در رک

Properties کارت DO

هنگامی‌که کارت DO در رک قرار گرفت، می‌توان با دوبار کلیک کردن بر روی آن به قسمت تنظیم مشخصات کارت دسترسی پیدا نمود. در کارت‌های‌معمولی‌دراین‌قسمت‌دوسربرگ General و Addresses مشابه کارت‌های DI وجود دارد.



شکل ۱۱-۵۰ سربرگ General و Addresses برای کارت DO 16xAC120V/0.5A

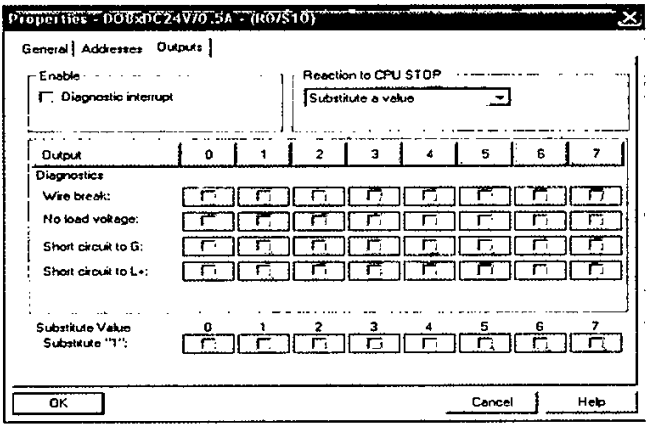
آنچه برای کارت DI در مورد این سربرگ‌ها ذکر شد، در اینجا نیز صدق می‌نماید لذا از ارائه توضیحات آن صرفنظر می‌گردد.

۱۱-۶-۶ قابلیت‌های ویژه در کارت‌های DO

برخی از کارت‌های دیجیتال خروجی دارای قابلیت‌های ویژه‌ای می‌باشند. این قابلیت‌ها تحت عنوان وقفه Diagnostic شناخته می‌شوند که در کتاب سطح پیشرفته تشریح شده است. برخی از این موارد عبارتند از:

- **No Load Voltage L+:** تشخیص قطعی ولتاژ مثبت مربوط به تغذیه‌ی کارت
- **Wire Break:** تشخیص قطعی سیم مصرف‌کننده‌ی متصل به کارت
- **Short Circuit to Ground:** تشخیص اتصال کوتاه سیگنال خروجی با زمین (ولتاژ منفی)
- **Short Circuit to L+:** تشخیص اتصال کوتاه سیگنال خروجی با ولتاژ مثبت

در کارت‌هایی که یکی یا چندتا از قابلیت‌های فوق وجود داشته باشد، علاوه بر دو سربرگ General و Addresses، سربرگ دیگری به نام Outputs نیز وجود دارد. در شکل۱۱-۵۱ محتویات این سربرگ برای کارت DO8xDC24V/0,5A تحت شماره فنی6ES7 322-8BF80-0AB0 نشان داده شده است.



شکل ۱۱-۵۱ سربرگ Outputs از Properties کارت DO

همانطور که در شکل۱۱-۵۱ مشخص است، مشابه کارت DI می‌توان هر کدام از قابلیت‌های ذکر شده را فعال نمود.

Reaction to CPU STOP

علاوه بر موارد ذکر شده در مورد Diagnostic، برخی از کارت‌ها دارای قابلیتی به نام Reaction to CPU STOP می‌باشند که منظور از آن، عکس‌العمل کارت نسبت به توقف CPU می‌باشد. در این قسمت دو انتخاب وجود دارد:

Substitute a value

با انتخاب این گزینه، می‌توان حالت هر یک از خروجی‌ها را در شرایط توقف CPU به صورت قطع یا وصل تنظیم نمود. در صورتی‌که گزینه‌ی Substitute 1 مربوط به هر کانال خروجی در قسمت Substitute Value علامت زده شود، در صورت توقف CPU خروجی مورد نظر وصل می‌شود. اگر این گزینه علامت زده نشود خروجی مورد نظر قطع می‌گردد.

Keep last valid value

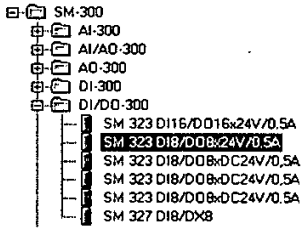
با انتخاب این گزینه، در صورت توقف CPU آخرین مقدار معتبر مربوط به هر خروجی حفظ می‌گردد.

سربرگ Identification

برخی از کارت‌های DO در پنجره‌ی Properties خود، دارای سربرگ دیگری به نام Identification می‌باشند. مشخصات قابل تنظیم در این بخش مشابه مشخصات ذکر شده در سربرگ Identification کارت‌های DI می‌باشد لذا از شرح آن خودداری می‌گردد.

۱۱-۷ کارت‌های ورودی و خروجی دیجیتال (ترکیبی DI/DO)

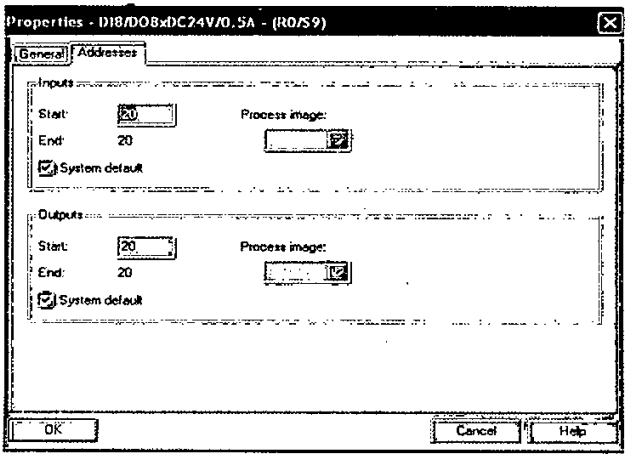
در S7-300 کارت‌هایی وجود دارند که هم دارای کانال‌های ورودی و هم دارای کانال‌های خروجی می‌باشند. این کارت‌ها در کاتالوگ ماژول‌ها در برنامه‌ی HW Config در قسمت SM-300 و در زیرشاخه‌ی DI/DO-300 قرار دارند. این کارت‌ها از جمله کارت‌های معمولی محسوب می‌گردند که هیچ قابلیت ویژه‌ای ندارند و فقط از نظر اقتصادی نسبت به دو کارت ورودی و خروجی مجزا به‌صرفه‌تر هستند. در شکل ۱۱-۵۲ مسیر دسترسی به این کارت‌ها نشان داده شده است.



شکل ۱۱-۵۲ مسیر دسترسی به کارت‌های DI/DO

کارت‌های SM323 DI/DO

این کارت‌ها از نظر فیزیکی دارای دو بخش مجزا برای ورودی و خروجی هستند. با قرار دادن این کارت‌ها در HW Config و مشاهده Properties کارت می‌بینیم که در سربرگ Address دو بخش مجزا برای ورودی و خروجی وجود دارد.



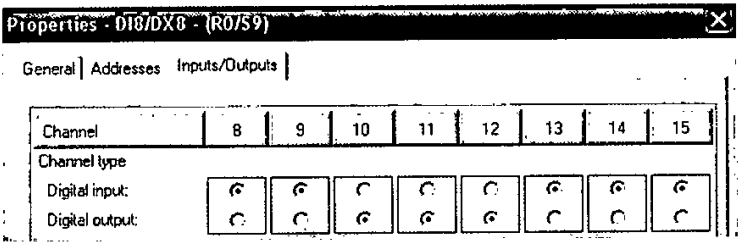
شکل ۱۱-۵۳ سربرگ‌های موجود در کارت‌های DI/DO

کارت SM 327 DI8/DX8

این کارت دارای ۱۶ کانال می‌باشد که:

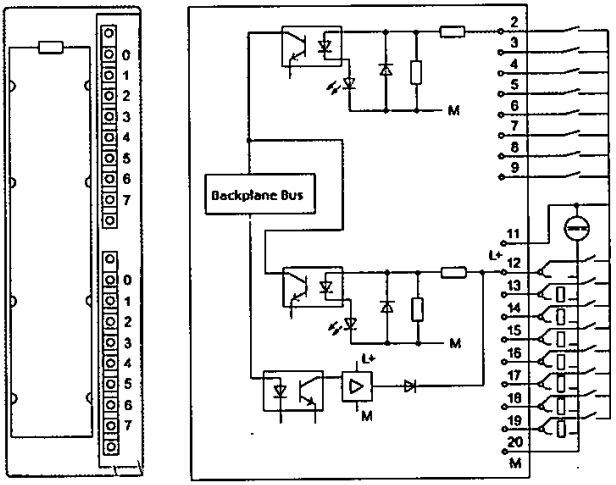
- ۸ کانال اول به‌عنوان ورودی محسوب می‌گردد.
- ۸ کانال دوم را می‌توان در نرم‌افزار به‌صورت ورودی یا خروجی تنظیم نمود.

شکل ۱۱-۵۴ چگونگی تنظیم ۸ کانال دوم این کارت را به‌عنوان ورودی یا خروجی نشان می‌دهد.



شکل ۱۱-۵۴ تنظیم ۸ کانال دوم SM 327 DI8/DX8

شکل ۱۱-۵۵ شماتیک و اتصالات کارت SM327 را نشان می‌دهد.



شکل ۱۱-۵۵ شماتیک و اتصالات کارتSM327

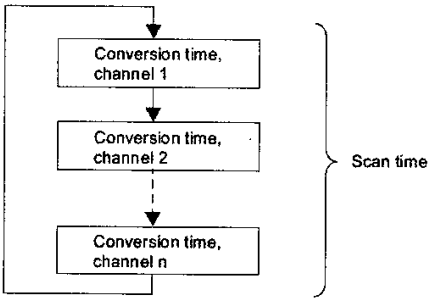
۱۱-۸ کارت‌های آنالوگ ورودی

در کتاب سطح پیشرفته، برنامه‌نویسی و نحوه کار با سیگنال‌های آنالوگ به‌طور کامل تشریح شده است. در این فصل کارت‌های آنالوگ ورودی به اختصار و در حدی که بتواند یک دیدگاه کلی به خواننده ارائه کند شرح داده می‌شود.

۱۱-۸-۱ عملکرد

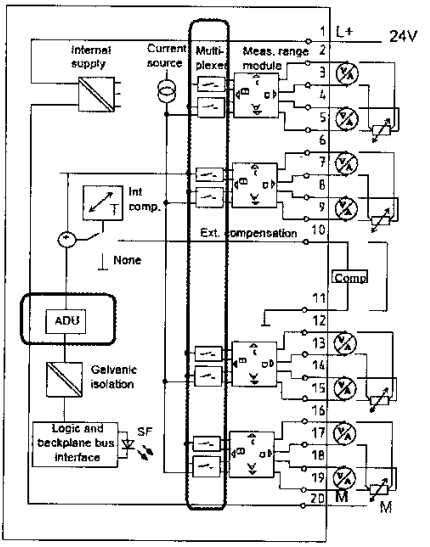
کارت‌های آنالوگ ورودی، سیگنال‌های الکتریکی آنالوگ را از سنسورها و ترانسمیترهای آنالوگ دریافت نموده و آنها را به دیتا تبدیل می‌نمایند. در این کارت‌ها به‌منظور تبدیل سیگنال الکرتیکی به دیتا، از واحدی به نام A/D استفاده می‌گردد. سیگنال ورودی به هر کانال کارت آنالوگ به A/D وارد شده و در آنجا به دیتا که یک عدد دیجیتال ۱۶ بیتی است تبدیل می‌گردد. لازم است کاربر در برنامه‌ی خود این عدد را به‌طور صحیح مقیاس نموده و سپس از آن استفاده نماید.

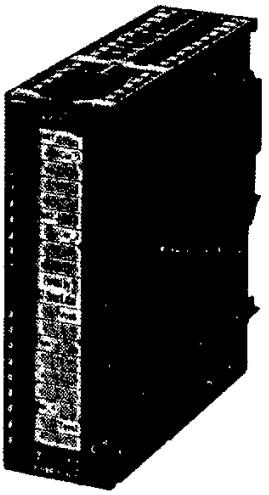
کارت‌های آنالوگ ورودی، دارای یک واحد تبدیل سیگنال الکتریکی آنالوگ به دیتا می‌باشند که به آن A/D (ADC) گفته می‌شود. نحوه‌ی کار بدین صورت است‌که سیگنال الکتریکی آنالوگ از سنسورها و ترانسمیترهای آنالوگ وارد کارت AI می‌شود. این سیگنال الکتریکی توسط واحد تبدیل آنالوگ به دیجیتال در کارت (A/D)، به یک عدد باینری ۱۶ بیتی تبدیل می‌گردد. این عمل برای هر کانال در کارت آنالوگ ورودی به‌طور جداگانه و به‌طور سیکلی انجام می‌پذیرد. شکل ۱۱-۵۶ نحوه اسکن کانال‌ها به‌منظور تبدیل سیگنال به دیتا را نشان می‌دهد.



شکل ۱۱-۵۶ نحوه اسکن هر کانال به‌منظور تبدیل سیگنال به دیتا در کارت AI

از آنجا که تعداد ورودی کارت‌ها معمولاً چند ورودی است، این ورودی‌ها ابتدا به یک Multiplexer وارد می‌شوند و در هر لحظه مالتی‌پلکسر یکی از ورودی‌ها را به A/D ارسال می‌کند. شکل ۱۱-۵۷ شماتیک یک کارت ۸ ورودی را نشان می‌دهد که در آن مالتی‌پلکسر و بلاک A/D مشخص هستند.

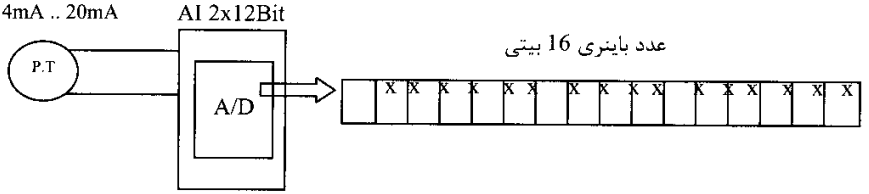



AI 8 x 13 bit

شکل ۱۱-۵۷ شماتیک کارت AI 8 x 13 bit

بدیهی است که به‌منظور کاهش زمان تبدیل سیگنال به دیتا بهتر است در نرم‌افزار کانال‌های استفاده نشده را به‌صورتی که در بخش پیکربندی HW Config شرح داده شده است غیرفعال[1] نمود تا مالتی‌پلکسر آنرا انتخاب نکند.

در شکل ۱۱-۵۸ نحوه تبدیل سیگنال الکتریکی به دیتا در یک کارت AI را مشاهده می‌نمایید.



شکل ۱۱-۵۸ تبدیل سیگنال الکتریکی به دیتا

نکته: عدد باینری حاصل را می‌توان به‌صورت Hex یا دسیمال یا عدد صحیح[2] در برنامه استفاده نمود. اما اینکه چه رابطه‌ای بین عدد حاصل با کمیت اندازه‌گیری‌شده وجود دارد سؤالی است که در کتاب پیشرفته به آن پاسخ داده خواهد شد.

1. Deactivate
2. Integer

دقت و حد تفکیک کارت آنالوگ ورودی

منظور از حد تفکیک قابلیت جداسازی می‌باشد و بیانگر تشخیص کوچکترین تغییر در سیگنال الکتریکی دریافتی توسط کارت می‌باشد.

در برخی کارت‌ها تغییرات سیگنال توسط مبدل A/D با استفاده از ۱۶ بیت تبدیل می‌شود؛ این نوع کارت‌ها دقت زیادی دارند. در برخی دیگر تغییرات سیگنال فقط با ۱۲ بیت تبدیل می‌شود؛ این نوع کارت‌ها دقت کمتری دارند. دقت کارت می‌تواند از ۸ بیت تا ۱۶ بیت باشد.

جزئیات حد تفکیک و میزان دقت کارت در کتاب سطح پیشرفته در بحث برنامه‌نویسی و کار با آنالوگ آمده است.

۱۱-۸-۲ انواع کارت های آنالوگ ورودی (AI)

کارت‌های آنالوگ ورودی با توجه به نوع و تعداد سیگنال‌های دریافتی و حد تفکیک کارت به انواع مختلفی تقسیم‌بندی می‌شوند.

جدول ۱۱-۱۳ انواع کارت‌های آنالوگ ورودی را به‌طور کلی نشان می‌دهد.

جدول ۱۱-۱۳ انواع کارت‌های آنالوگ ورودی

از نظر تعداد ورودی	از نظر نوع سیگنال	ارنظر قابلیت های خاص
• ۲ ورودی • ٤ ورودی • ۶ ورودی • ۸ ورودی	• ولتاژ • جریان • مقاومت • TC (ترموکوپل) • RTD • ترکیبی از موارد فوق	• بدون ویژگی خاص • ایجاد وقفه • تشخیص قطعی

به‌عنوان مثال کارت SM 331 AI2x12Bit را در نظر بگیرید. این کارت از نوع آنالوگ ورودی بوده (AI)، دارای دو کانال است و حد تفکیک A/D آن ۱۲ بیت است (2x12Bit). کارت‌هایی که دارای حروف RTD بعد از شماره کارت هستند، مخصوص اتصال RTD و مقاومت طراحی شده‌اند. کارت‌های TC نیز مخصوص اتصال ترموکوپل طراحی شده‌اند. البته در برخی موارد این کارت‌ها می‌توانند سیگنال‌های دیگر را نیز دریافت نمایند که تنظیم آن در توسط HW Config امکان‌پذیر می‌باشد.

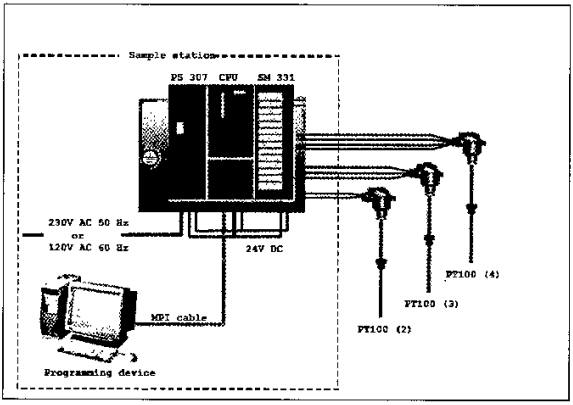
سیگنال‌های قابل دریافت توسط کارت AI

نوع سیگنال‌های الکتریکی دریافتی توسط کارت AI به چهار گروه تقسیم می‌شود.

۱. سیگنال دریافتی از RTD یا مقاومت

این سیگنال‌ها ماهیت اهمی دارند. RTDها مقاومت‌های متغیری هستند که برای اندازه‌گیری دما به‌کار می‌روند. تغییرات دما منجر به تغییرات مقاومت RTD شده و توسط کارت ورودی حس می‌گردد.

RTDها و مقاومت‌ها را می‌توان به‌صورت ۲ سیمه، ۳ سیمه یا ۴ سیمه به کارت آنالوگ متصل کرد.

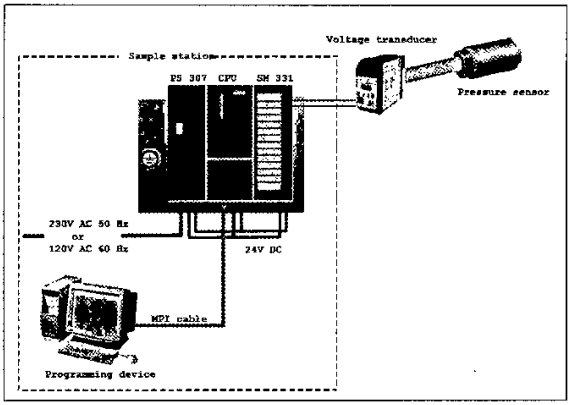


شکل ۱۱- ۵۹ اتصال RTD به کارت AI

۲. سیگنال ولتاژی

این‌سیگنال از ترانسدیوسرهاوترانسمیترها دریافت می‌گردد و دارای بازه‌های استاندارد است. برخی از انواع آن عبارت است از:

- +/- 25 mV
- +/- 50 mV
- +/-80mV
- +/-250mV
- +/-500mV
- +/-1V
- +/-2.5V
- +/-5V
- 1..5V
- +/-10V

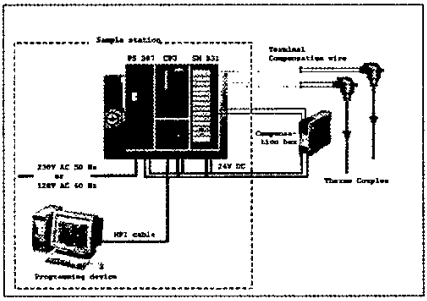


شکل ۱۱-۶۰ اتصال سیگنال‌های ولتاژی به کارت AI

۳. سیگنال دریافتی از ترموکوپل

این سیگنال به‌صورت میلی‌ولت از ترموکوپل دریافت می‌گردد. برخی از انواع آن عبارت است از:

- نوع E
- نوع J
- نوع K
- نوع T
- نوع S
- نوع R

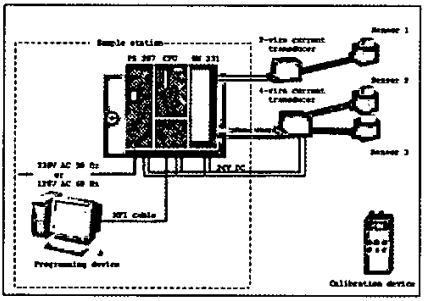


شکل ۱۱-۶۱ اتصال ترموکوپل به کارت AI

۴. سیگنال جریانی

این سیگنال از ترانسدیوسرها و ترانسمیترها دریافت می‌گردد. برخی از انواع آن عبارت است از:

- -5mA to +5mA
- -10mA to +10mA
- -20mA to +20mA
- 0mA to 20mA
- 4mA to 20mA



شکل ۱۱-۶۲ اتصال سیگنال جریانی به کارت AI

نکته: هر یک از کارت‌های آنالوگ ورودی می‌توانند یک یا چند نوع از سیگنال‌های فوق را دریافت نمایند. جهت اطمینان از نوع سیگنال‌های دریافتی یک کارت آنالوگ می‌توان به کاتالوگ فنی آن کارت مراجعه نمود، و یا از طریق برنامه‌ی HW Config در قسمت مشخصات کارت آنرا بررسی کرد.

۱۱-۸-۳ مشخصات فنی کارت‌های آنالوگ ورودی

مشابه موارد ذکر شده در مورد سایر ماژول‌ها می‌توان به مشخصات فنی کارت‌ها AI نیز دسترسی پیدا نمود. جدول ۱۱-۱۴ نمونه‌ای از مشخصات فنی چند نمونه کارت‌های AI را نشان می‌دهد. جزئیات بیشتر در کتاب سطح پیشرفته آمده است.

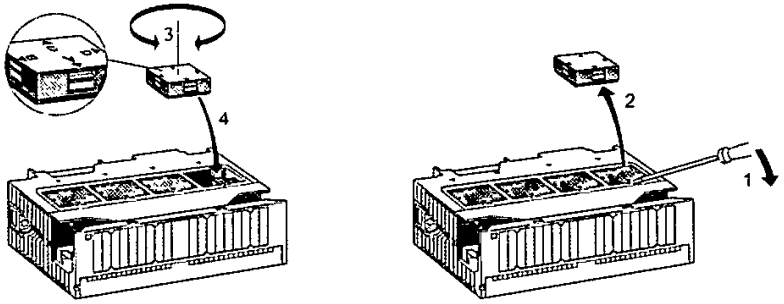
جدول ۱۱-۱۴ مشخصات فنی چند نمونه کارت AI-300

Properties	SM 331; AI 8 x TC (-7PF11-)	SM 331; AI 8 x thermoresistor (-7PF01-)	SM 331; AI 8 x 0/4...20 mA HART (-7TF00-)*
Number of inputs	8 inputs in 4 channel groups	8 inputs in 4 channel groups	8 inputs in 1 channel group
Resolution	Programmable for each channel	Programmable for each channel	programmable for each channel
	group: 15 bits+sign	group: 15 bits+sign	group: 15 bits+sign

Measurement type	programmable for each channel	programmable for each channel	Programmable for each channel
	group: Temperature	group: Resistance Temperature	group: Voltage Current Resistance
			Temperature
Measuring range selection	any, per channel group	any, per channel group	any, per channel group
Supports isochronous mode	Yes	Yes	Yes
Programmable diagnostics	No	No	no
Diagnostic interrupt	Programmable	Programmable	Programmable
Limit value monitoring	programmable for 8 channels	programmable for 8 channels	programmable for 8 channels
Hardware interrupt when limit	Programmable	Programmable	programmable
exceeded			
Hardware interrupt at end of cycle	Programmable	Programmable	No
maximum potential difference between inputs	60 VAC/75 VDC	60 VAC/75 VDC	60 VAC/75 VDC

۱۱-۸-۴ نصب کارت‌های آنالوگ ورودی

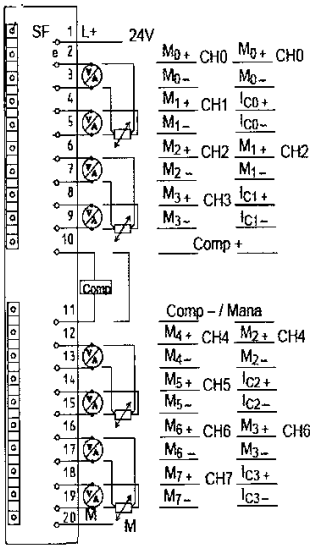
نصب کارت‌های آنالوگ روی Rack300 مشابه آنچه در مورد کارت‌های دیجیتال ذکر شد می‌باشد. تنها نکته‌ای قابل توجه این است که برخی از انواع کارت‌های آنالوگ قادر هستند تمام چهار نوع سیگنال را بپذیرند. این کارت‌ها دارای یک تنظیم سخت‌افزاری می‌باشند که این تنظیم ابتدا در کنار کارت بایستی مانند شکل ۱۱-۶۳ انجام شده، سپس کارت روی رک نصب گردد.



شکل ۱۱-۶۳ مراحل چهارگانه تنظیم ماژول انتخاب رنج

سیم‌کشی و اتصالات کارت آنالوگ بر اساس نقشه در کارت یا منوال مربوط به کارت انجام می‌شود. شکل ۱۱-۶۴ نقشه سیم‌کشی کارت شکل ۱۱-۶۳ را نشان می‌دهد. نقشه یا پشت در کارت حروف و علاماتی به‌شرح زیر دیده می‌شود:

- L+: محل اتصال تغذیه 24VDC
- M: محل اتصال زمین تغذیه
- M+: محل اتصال سیم مثبت سیگنال آنالوگ
- M-: محل اتصال سیم منفی سیگنال آنالوگ
- Comp+: اتصال سیم مثبت جبران‌ساز ترموکوپل
- Comp-: اتصال سیم منفی جبران‌ساز ترموکوپل
- IC+: اتصال منبع جریان ثابت به سیم RTD یا مقاومت
- IC-: اتصال منفی منبع جریان ثابت به سیم RTD یا مقاومت
- MANA: پایه رفرنس برای مدار سیگنال آنالوگ



شکل ۱۱-۶۴ نقشه سیم‌کشی یک نمونه کارت آنالوگ ۸ ورودی

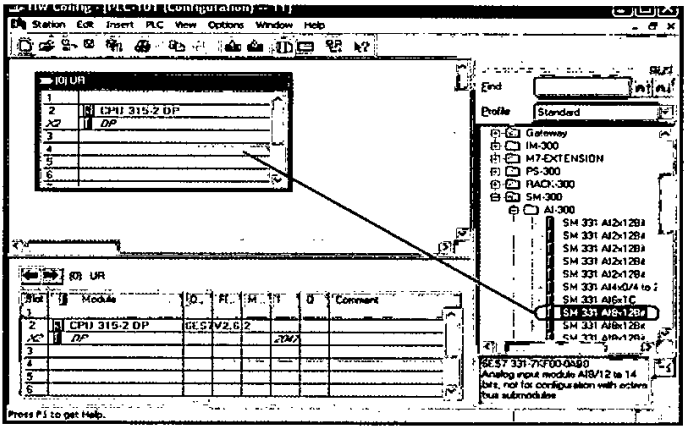
نکات زیر به‌طور خلاصه در مورد پایه‌های فوق ذکر می‌شود. تشریح آنها در کتاب سطح پیشرفته آمده است:

- سیگنال‌های ولتاژی و جریانی و ترموکوپل به +M و -M وصل می‌شوند.
- اگر از جبران‌ساز بیرونی برای ترموکوپل استفاده شود، به +Comp و Comp متصل می‌گردد.
- RTD و مقاومت‌ها می‌توانند به‌صورت ۴ سیمه بسته شوند. در این حالت یک زوج سیم بالا و پایین آنها به منبع جریانی که با IC مشخص شده بسته می‌شود و زوج سیم دیگر به پایه‌های +M و -M اتصال می‌یابد.

- MANA پایه رفرنس مدار آنالوگ است. اگر از M تغذیه جدا باشد ، گفته می‌شود که کارت ، ایزوله شده است.
- اگر MANA به M تغذیه وصل باشد، گفته می‌شود که *کارت ایزوله نشده است.*
- اگر -M به MANA وصل نباشد، گفته می‌شود که *سنسور ایزوله شده است.*
- اگر -M به MANA وصل باشد، گفته می‌شود که *سنسور ایزوله نشده است.*
- سنسورهای دو سیمه، جریانی و مقاومتی بایستی ایزوله شده بسته شوند؛ یعنی -M به MANA وصل نشود.

۱۱-۸-۵ پیکربندی کارت AI در Hwconfig

این کارت‌ها در کاتالوگ موجود در برنامه HW Config در قسمت SM-300 و در زیرشاخه AI-300 قرار گرفته‌اند که می‌توان مطابق شکل ۱۱-۶۵ کارت دلخواه را از این مجموعه انتخاب نموده و در رک قرار داد.



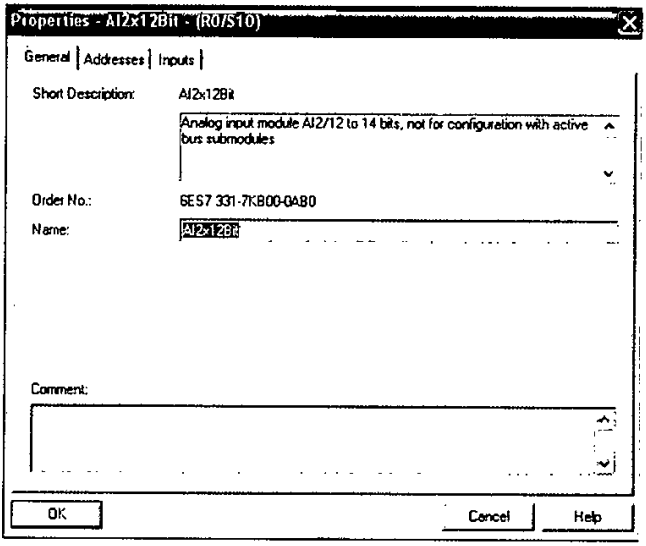
شکل ۱۱-۶۵ انتخاب کارت AI و وارد نمودن در رک

تنظیم پارامترهای کارت AI

هنگامی که یک کارت AI در رک قرار گرفت باید برخی از مشخصات آن تنظیم گردد. مهم‌ترین مشخصه، تعیین نوع و رنج سیگنال ورودی هر کانال می‌باشد. پس از اینکه کارت AI در رک قرار گرفت مشابه سایر کارت‌های ورودی/خروجی، می‌توان با دوبار کلیک نمودن بر روی آن به پنجره مشخصات کارت دسترسی پیدا نمود. سربرگ‌های موجود در این پنجره عبارتند از:

General

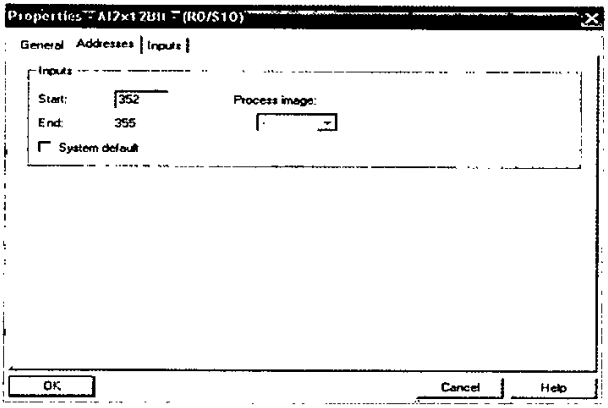
در این سربرگ، شماره فنی کارت به همراه توضیحات کوتاهی در مورد کارت نشان داده می‌شود.



شکل ۱۱-۶۶ سربرگ General از مشخصات کارت AI2x12Bit

Addresses

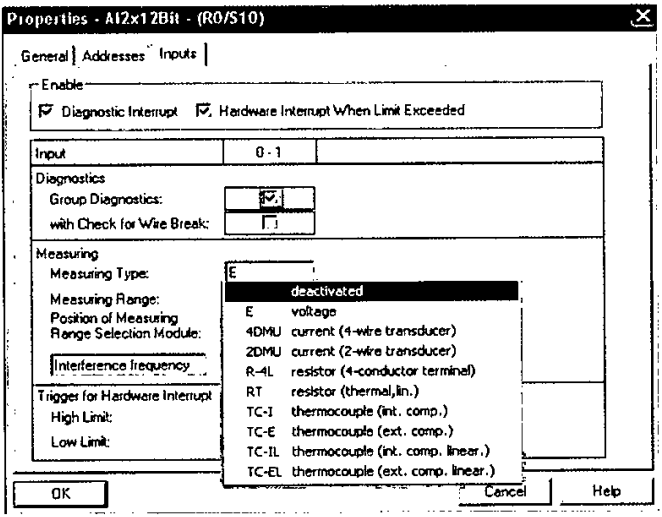
در این سربرگ می‌توان آدرس اختصاص‌یافته به کارت را مشاهده و ویرایش نمود. همانطور که بیان شد، عدد حاصل از A/D در16 بیت (یک word= دو بایت) قرار می‌گیرد. شکل زیر بخش آدرس یک کارت آنالوگ دو ورودی را نشان می‌دهد. چون دو ورودی در مجموع ۴ بایت را اشغال می‌کند، بنابراین آدرس از ۳۵۲ شروع و تا ۳۵۵ ادامه یافته است.



شکل ۱۱-۶۷ سربرگ Addresses از مشخصات کارت AI2x12Bit

Inputs

محتویات این سربرگ با توجه به نوع کارت AI متفاوت می‌باشد. به‌عنوان مثال محتویات این سربرگ جهت کارت AI 2x12Bit مانند شکل ۱۱-۶۸ است. یکی از بخش‌های مهم در این قسمت تعیین نوع سیگنال آنالوگ در قسمت Measuring Type است که به‌دنبال آن رنج سیگنال نیز در قسمت Measuring Range تعیین می‌شود.
تشریح جزئیات این تنظیم‌ها و سایر موارد مرتبط با وقفه در کتاب سطح پیشرفته آمده است.

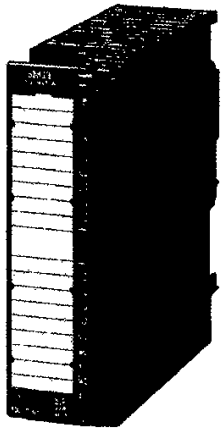


شکل ۱۱-۶۸ سربرگ Inputs در کارت AI 2x12Bit

۱۱-۹ کارت‌های آنالوگ خروجی

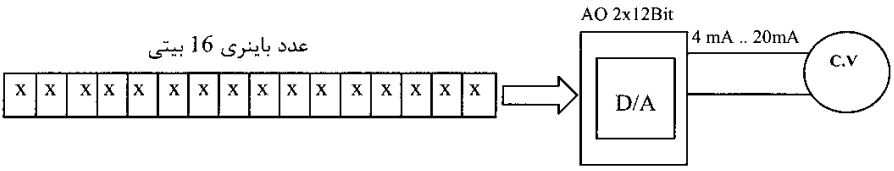
۱۱-۹-۱ عملکرد

دیتای ارسالی توسط CPU، در کارت‌های آنالوگ خروجی به سیگنال الکتریکی آنالوگ تبدیل شده و به تجهیزاتی آنالوگ مانند کنترل ولو، درایو و . . . ارسال می‌گردد. در این کارت‌ها به‌منظور تبدیل دیتا به سیگنال الکتریکی، از واحدی به نام D/A استفاده می‌گردد که عملکرد آن عکس A/D در آنالوگ ورودی است. در این واحد (D/A) دیتا به‌صورت عدد ۱۶ بیتی دریافت شده و به سیگنال الکتریکی تبدیل می‌شود. عمل تبدیل با توجه به حد تفکیک کارت صورت پذیرفته و هرچه حد تفکیک کارت بزرگتر باشد، دقت کارت در تبدیل دیتا به سیگنال الکتریکی بالاتر می‌باشد.



شکل ۱۱-۶۹ کارت SM322 AO 4x12Bit

نکته‌ی دیگری که باید به آن توجه نمود این است که در کارت AO تبدیل دیتا به سیگنال به‌طور سیکلی برای همه‌ی کانال‌ها تکرار می‌شود (مشابه سیکل تبدیل سیگنال به دیتا در کارت AI). به شکل ۱۱-۷۰ توجه کنید.



شکل ۱۱-۷۰ تبدیل دیتا به سیگنال آنالوگ

۱۱-۹-۲ انواع کارت‌های AO

کارت‌های آنالوگ خروجی با توجه‌به‌نوع‌وتعدادسیگنال‌های ارسالی و حد تفکیک کارت به انواع مختلفی تقسیم‌بندی می‌شوند. در جدول ۱۱-۱۵ انوع کارت‌های آنالوگ خروجی به‌طور کلی نشان داده شده است.

جدول ۱۱-۱۵ انواع کارت‌های آنالوگ خروجی

از نظر تعداد خروجی	از نظر نوع سیگنال	ازنظر قابلیت های خاص
• ۲ ورودی	ولتاژ	بدون ویژگی خاص
• ٤ ورودی	جریان	تشخیص قطعی
• ۸ ورودی	ترکیب دو مورد فوق	تشخیص اتصال کوتاه
		واکنش به توقف CPU

همانطور که در این جدول نشان داده شده است، تنها دو نوع سیگنال الکتریکی خروجی وجود دارد:

- سیگنال آنالوگ ولتاژی
- سیگنال آنالوگ جریانی

به‌عنوان مثال کارت SM 332 AO2x12Bit را در نظر بگیرید. این کارت از نوع آنالوگ خروجی بوده (AO) دارای دو کانال است که می‌توانند به‌صورت ولتاژی یا جریانی تنظیم شود. حد تفکیک D/A این کارت نیز ۱۲ بیت است.

۱۱-۹-۳ مشخصات فنی کارت های آنالوگ خروجی

جدول ۱۱-۱۶ نمونه‌ای از مشخصات فنی چند کارت AO را نشان می‌دهد.

جدول ۱۱-۱۶ مشخصات فنی چند کارت AO-300

Properties	SM 332; AO 4 x 16 Bit (-7ND02-)	SM 332; AO 2 x 12 Bit (-5HB01-)	SM 332; AO 8 x 0/4...20mA HART (-8TF00-) *
Number of outputs	4 outputs in 4 channel groups	2 output channels	8 output channels
Resolution	16 bits	12 bits	15 bits (0 mA to 20mA)
Output type	per channel: ☐ Voltage ☐ Current	per channel: ☐ Voltage ☐ Current	per channel: ☐ Voltage ☐ Current
Supports isochronous mode	yes	no	no
Programmable diagnostics	yes	yes	yes
Diagnostic interrupt	programmable	programmable	programmable
Substitute value	programmable	programmable	programmable

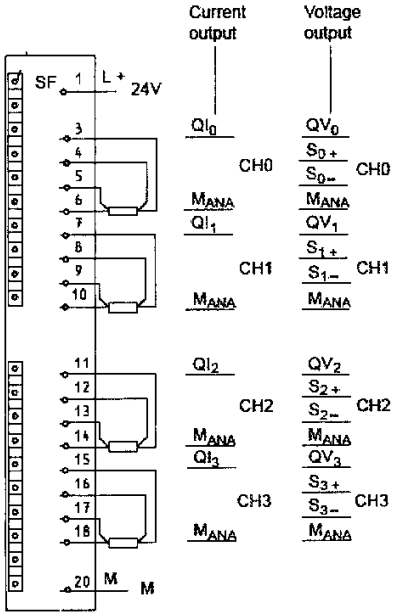
۱۱-۹-۴ نصب کارت آنالوگ خروجی

نصب کارت آنالوگ خروجی روی رک به‌سادگی و مشابه کارت‌های دیجیتال انجام می‌شود. هیچ تنظیم سخت‌افزاری نیز روی کارت لازم نیست.

مشابه کارت‌های دیگر، در این کارت‌ها نیز نقشه‌ی اتصالات و سیم‌کشی کارت در حالت خروجی ولتاژی و جریانی در پشت درب کارت درج شده است. شکل زیر نمونه‌ای از نقشه کارت آنالوگ ۴ خروجی را نشان می دهد. علائمی که در این نقشه به‌کار رفته عبارتند از:

- L+: تغذیه 24VDC
- M: زمین تغذیه

- QI: خروجی جریانی
- QV: خروجی ولتاژی
- +S: سنسور اندازه‌گیری ولتاژ (پایه مثبت)
- -S: سنسور اندازه‌گیری ولتاژ (پایه مثبت)
- MANA: رفرنس مدار آنالوگ



شکل ۱۱-۷۱ نقشه اتصال کارت نمونه آنالوگ ۴ خروجی

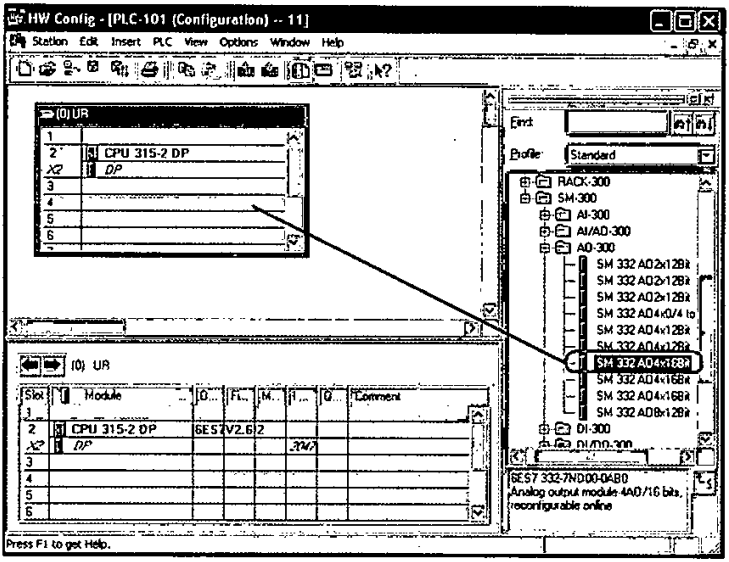
نکاتی که به اجمال در اینجا می توان اشاره کرد این است که:

- آنالوگ خروجی جریانی از پایه های QI و MANA گرفته می‌شود.
- آنالوگ خروجی ولتاژی از پایه های QV و MANA گرفته می‌شود.
- آنالوگ خروجی ولتاژی می‌تواند برای دقت بیشتر به‌صورت ۴ سیمه بسته شود. در این حالت +S و -S از دوسر بار فیدبک به کارت می‌دهد تا افت ولتاژ را جبران کند. در حالت ولتاژ دو سیمه +S و -S آزاد هستند و متصل نمی‌شوند.

جزئیات بحث آنالوگ خروجی در کتاب سطح پیشرفته تشریح شده است.

۱۱-۹-۵ پیکربندی کارت آنالوگ خروجی در Hwconfig

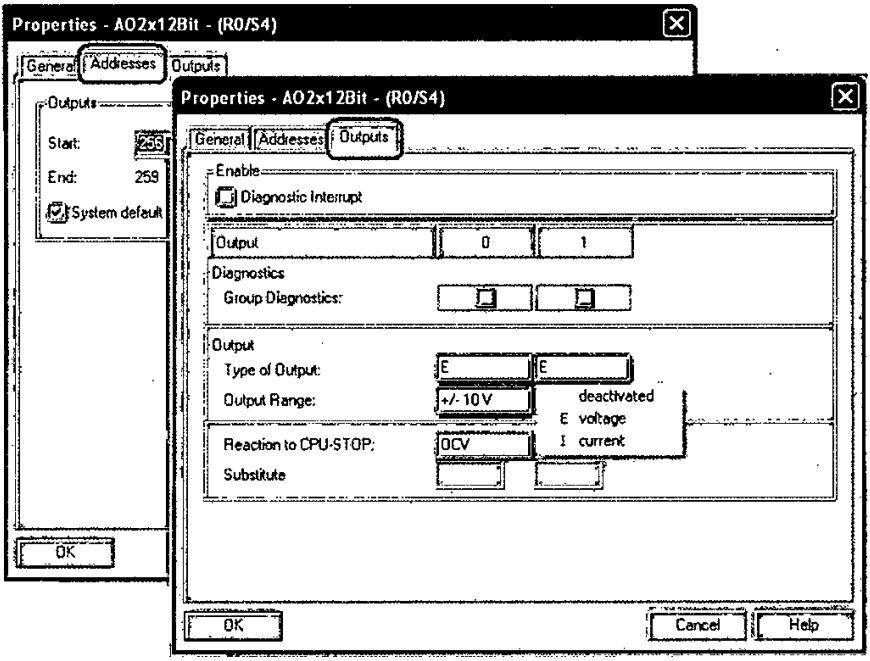
کارت‌های خروجی آنالوگ نسبت به ورودی‌های آنالوگ تنوع کمتری دارند. با انتخاب آنها از کاتالوگ و وارد کردن در اسلات‌های مجاز، پنجره‌ای مانند شکل ۱۱-۷۲ ظاهر خواهد شد.



شکل ۱۱-۷۲ انتخاب کارت AO و وارد نمودن در رک

به‌منظور تنظیم مشخصات کارت AO روی آن دوبار کلیک می‌کنیم؛ دراین‌صورت پنجره‌ای مانند شکل ۱۱-۷۳ که مربوط به کارت AO 2 x 12 Bit است باز می‌شود.

در سربرگ Address می‌بینیم که برای هر کانال خروجی نیز دو بایت (مشابه ورودی آنالوگ) در نظر گرفته شده است. در سربرگ Output آن نیز تنظیمات نوع سیگنال و برخی موارد دیگر مانند وقفه‌ها انجام می‌شود. تشریح جزئیات این بحث در کتاب سطح پیشرفته آمده است.



شکل ۱۱-۷۳ پنجره Properties مربوط به کارت AO2x12Bit

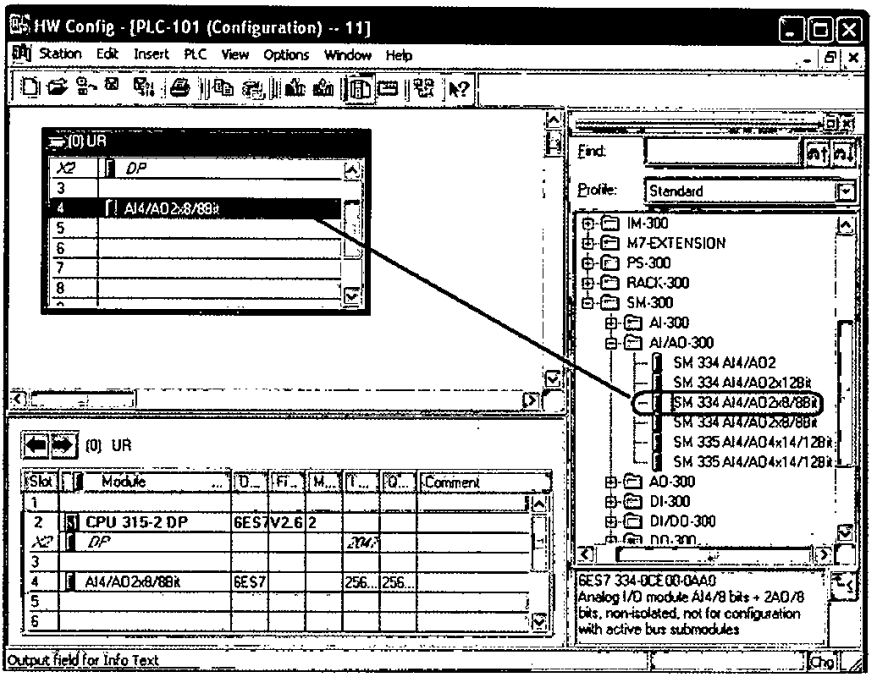
Reaction to CPU-STOP

در این قسمت می‌توان نحوه‌ی عکس‌العمل کارت به توقف CPU را تعیین نمود. به‌عبارت دیگر وقتی در حین کار CPU دچار توقف شود، می‌توان تعیین کرد که ولتاژ یا جریان عملگر در چه مقداری باقی بماند تا شرایط ایمنی یا سایر نیازهای فرآیند حفظ گردد.

۱۱-۱۰ کارت‌های ورودی و خروجی آنالوگ (ترکیبی AI/AO)

مشابه کارت‌های دیجیتال، برخی از کارت‌های آنالوگ هم دارای کانال‌های ورودی و هم کانال‌های خروجی می‌باشند. به این کارت‌ها می‌توان سیگنال الکتریکی آنالوگ متصل نمود و همچنین دریافت سیگنال الکتریکی آنالوگ از آنها امکان‌پذیر می‌باشد.

به‌طور کلی این‌گونه کارت‌ها قابلیت‌های کمتری نسبت به کارت‌های مجزا دارند ولی اقتصادی‌تر هستند. این کارت‌ها در کاتالوگ موجود در برنامه HW Config در قسمت SM-300 و در زیرشاخه AI/AO -300 قرار گرفته‌اند. شکل ۱۱-۷۴ انواع این کارت‌ها را نشان می‌دهد.



شکل ۱۱-۷۴ نحوه انتخاب و وارد نمودن کارت AI4/AO2x8/8Bit در رک

۱۱-۱۱ کارت‌های خاص خانواده SM

۱-۱۱-۱۱ Dummy Module (DM)

همانطور که اشاره شد، در S7-300 به‌دلیل اینکه ارتباط ماژول‌ها با CPU از طریق Bus Connector صورت می‌پذیرد، لذا وجود فاصله‌ی خالی در رک منجر به قطع ارتباط CPU و برخی از ماژول‌ها می‌گردد. از اینرو نمی‌توان فاصله‌ی خالی بین ماژول‌ها در رک ایجاد نمود، و در صورت نیاز به رزرو یک اسلات جهت ماژولی که قرار است در آینده به پیکربندی اضافه شود، می‌توان از DM استفاده نمود. وظیفه‌ی DM ذخیره‌ی اسلات برای یک ماژول پیکربندی نشده می‌باشد. DM در موارد زیر می‌تواند به‌کار گرفته شود:

- رزرو اسلات برای Signal Module های پیکربندی نشده مانند DI,DO,AI,AO, DI/DO, AI/AO (با ذخیره‌سازی آدرس)
- رزرو اسلات جهت کارت IM (بدون رزرو آدرس)

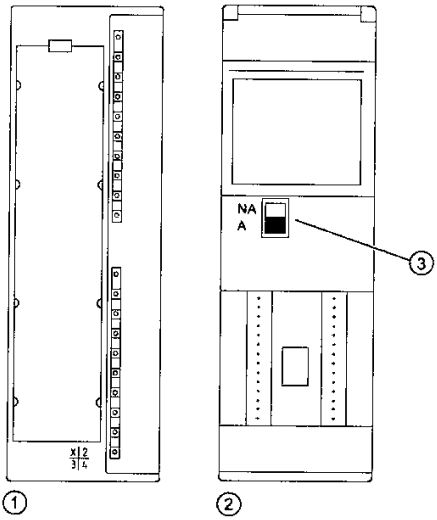
نکته: هنگامی‌که ماژول مورد نظر جایگزین DM شود، پیکربندی سخت‌افزار و آدرس اختصاص داده شده جهت سایر ماژول‌ها بدون تغییر باقی می‌ماند.

پیکربندی DM در STEP7

جهت انجام پیکربندی DM در نرم‌افزار STEP7 و زیربرنامه‌ی HW Config باید به نکات زیر توجه نمود:

- در صورتی‌که DM اسلات‌رابرای‌یک SM رزرونموده باشد، باید در نرم‌افزار HW Config آنرا پیکربندی نمود.
- در صورتی‌که DM اسلات‌رابرای‌یک IM رزرونموده‌باشد، درنرم‌افزار HWConfig نیاز به پیکربندی آن نمی‌باشد.

در شکل ۱۱-۷۵ شماتیک DM 307 را مشاهده می‌نمایید.



شکل ۱۱-۷۵ شماتیک DM 307

قسمت‌های شماره‌گذاری شده در شکل عبارتند از:

۱- تصویر DM 307 از نمای جلو

۲- تصویر DM 307 از نمای عقب

۳- کلید انتخاب وضعیت

با توجه به ماژولی که قرار است DM اسلات را برای آن ذخیره نماید، باید کلید انتخاب وضعیت را به‌طور صحیح تنظیم نمود. جدول ۱۱-۱۷ چگونگی انجام این تنظیم را نشان می‌دهد.

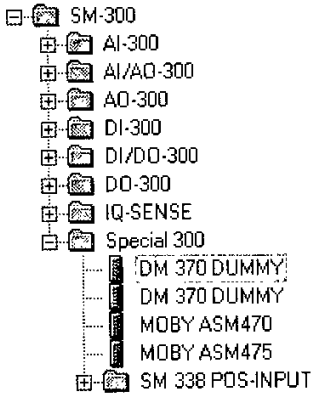
جدول ۱۱-۱۷ تنظیم کلید انتخاب وضعیت DM

وضعیت کلید	مفهوم
NA A	DM فقط یک اسلات را رزرو می‌نماید ولی در HW Config پیکربندی نمی‌شود و آدرسی را رزرو نمی‌کند.

DM یک اسلات را رزرو می‌نماید و در Hwconfig پیکربندی شده و آدرس را نیز رزرو می‌کند.	NA A

انتخاب DM و پیکربندی در رک

جهت وارد نمودن ماژول DM در رک، می‌توان از مسیر نشان داده شده در شکل ۱۱-۷۶ این ماژول را انتخاب نموده و آنرا با موس در اسلات مورد نظر قرار داد. پس از قرارگیری DM در رک، می‌توان با دوبارکلیک نمودن بر روی آن به پنجره‌ی مشخصات آن دسترسی پیدا نمود. تنها تنظیمی که در این قسمت می‌توان انجام داد، تغییر آدرس رزرو شده توسط ماژول می‌باشد.



شکل ۱۱-۷۶ مسیر دسترسی به DM در کاتالوگ برنامه HW Config

۱۱-۱۱-۲ Simulator Module (SM 374)

به‌منظور شبیه‌سازی نمودن عملکرد کارت‌های دیجیتال ورودی و دیجیتال خروجی یا تلفیقی از هر دو، از ماژولی به‌نام Simulator module استفاده می‌گردد. این ماژول دارای خصوصیات زیر است:

- می‌تواند عملکرد یک کارت دیجیتال با ۱۶ ورودی را شبیه‌سازی نماید.
- می‌تواند عملکرد یک کارت دیجیتال با ۱۶ خروجی را شبیه‌سازی نماید.
- می‌تواند عملکرد یک ماژول با ۸ ورودی و ۸ خروجی را شبیه‌سازی نماید.
- دارای چراغ وضعیت جهت بررسی ورودی یا خروجی‌هاست.
- دارای سوئیچ تنظیم جهت انتخاب وضعیت کاری ماژول است.

موارد کاربرد

در برخی موارد ممکن است نیاز شود که یکی از کارت‌های دیجیتال ورودی یا خروجی تست شوند. در این حالت می‌توان کارت مربوطه را از رک جدا نموده و ماژول شبیه‌ساز (SM 374) را جایگزین آن نمود. اگر کارت مورد نظر دیجیتال ورودی

باشد، می‌توان از طریق دکمه‌های روی SM 374 ورودی دیجیتال به PLC اعمال نموده و از نتیجه‌ی حاصل متوجه‌ی وجود اشکال در کارت یا سیم‌کشی‌ها یا سایر موارد شد.

اگر کارت مورد نظر جهت بررسی، کارت دیجیتال خروجی باشد می‌توان از طریق روشن و خاموش شدن لامپ‌های SM 374 متوجه صحت یا عدم صحت کار سیستم و وجود یا عدم وجود اشکال در کارت دیجیتال خروجی مورد نظر شد.

پیکربندی در STEP 7

این ماژول در کاتالوگ سخت‌افزارهای موجود در برنامه‌ی HW Config وجود ندارد. برای پیکربندی آن باید به نکات زیر توجه نمود:

- جهت استفاده از ماژول به‌عنوان کارت دیجیتال با ۱۶ ورودی، باید در نرم‌افزار یک کارت DI با ۱۶ ورودی را انتخاب نموده و در اسلات مورد نظر رک قرار داد.
- جهت استفاده از ماژول به‌عنوان کارت دیجیتال با ۱۶ خروجی، باید در نرم‌افزار یک کارت DO با ۱۶ خروجی را انتخاب نموده و در اسلات مورد نظر رک قرار داد.
- جهت استفاده از ماژول به‌عنوان کارت دیجیتال با ۸ ورودی و ۸ خروجی، باید در نرم‌افزار یک کارت DI8/DO8 را انتخاب نموده و در اسلات مورد نظر رک قرار داد.

کلیدها و علائم روی ماژول

به‌منظور شبیه‌سازی نمودن عملکرد ورودی‌های دیجیتال و نمایش خروجی‌های دیجیتال، در روی SM 374 از تعدادی کلید و لامپ نشان دهنده استفاده شده است. در شکل ۱۱-۷۷ قسمت‌های مختلف این ماژول به‌صورت شماره‌گذاری شده نشان داده شده‌اند.

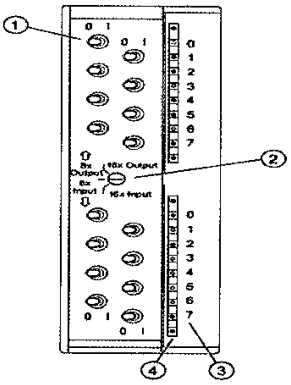
راهنما :

۱- کلید دو حالته‌ی قطع/ وصل، به‌منظور اعمال سیگنال دیجیتال ورودی

۲- پیچ انتخاب حالت کاری ماژول (قابل تنظیم در حالت‌های ۱۶ ورودی- ۱۶ خروجی – ۸ورودی، ۸ خروجی)

۳- شماره‌ی کانال‌های ماژول

۴- لامپ نشان‌دهنده وضعیت هر کانال



شکل ۱۱-۷۷ شماتیک Simulator Module

Position Decoder Module SM 338 ۳-۱۱-۱۱

این ماژول که در مجموعه‌ی SM-300 و در زیرشاخه‌ی Special 300 قرار گرفته است، به‌منظور دریافت پالس از انکودر طراحی شده و کاربرد اصلی آن در کنترل موقعیت می‌باشد. توضیحات بیشتر در کتاب سطح تکمیلی خواهد آمد.

۱۱-۱۲ کارت‌های شبکه Communication Processor (CP)

به‌منظور ارتباط PLC با شبکه‌های مختلف صنعتی نیاز به پورت شبکه روی CPU یا نصب کارت شبکه در کنار CPU می‌باشد. به‌طور کلی علت نیاز به کارت شبکه را می‌توان در موارد زیر خلاصه کرد:

- پورت‌های شبکه روی CPU محدود هستند و همه شبکه‌ها را نیز پشتیبانی نمی‌کنند.
- امکاناتی که کارت شبکه برای سیستم فراهم می‌کند وسیع‌تر از امکانات پورت همان شبکه روی CPU است.

برای جزئیات بیشتر لازم است به کتاب‌های مستقلی که در زمینه‌های شبکه از مؤلف به چاپ رسیده است مراجعه نمایید.

انواع کارت‌های شبکه

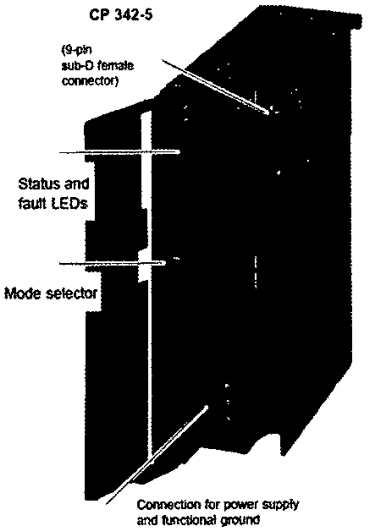
- کارت شبکه پروفی‌باس

کارت‌های شبکه پروفی‌باس که روی رک S7-300 قابل نصب هستند. به دو نوع اصلی زیر تقسیم می‌شوند:

۱. کارت شبکه Profibus –DP

۲. کارت شبکه Profibus –FMS

برای پروتکل Profibus –PA کارت شبکه‌ای که روی رک کنار PLC نصب شود عرضه نشده است و ارتباط آن از طریق یک مبدل به نام DP/PA انجام می‌گیرد.



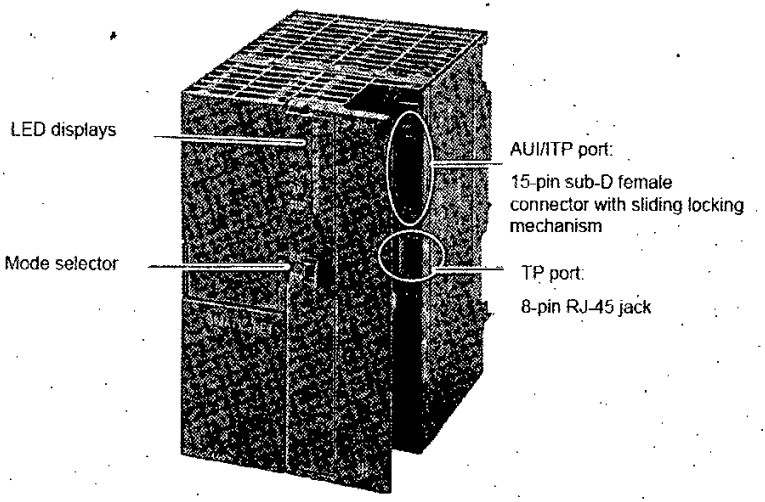
شکل ۱۱-۷۸ یک نمونه کارت شبکه پروفی‌باس

- کارت شبکه‌ی اترنت صنعتی

کارت‌های شبکه اترنت صنعتی که روی رک PLC قابل نصب هستند به چند دسته تقسیم می‌شوند:

۱. کارت‌هایی با امکانات کم ولی مقرون به‌صرفه مانند کارت CP343-1 Lean

۲. کارت هایی با امکانات متوسط و پرکاربرد مانند کارت CP343-1
۳. کارت هایی با امکانات پیشرفته و زیاد مانند کارت CP343-1 Advanced



شکل ۱۱-۷۹ کارت شبکه اترنت صنعتی

- کارت شبکه ASI

این کارت شبکه را می‌توان روی رک PLC و نیز روی رک Remote I/O نصب نمود. کارت ASI به دو نوع اصلی تقسیم می‌گردد:

۱. کارت ASI برای اتصال سیگنال‌های دیجیتال
۲. کارت ASI برای اتصال سیگنال‌های دیجیتال و آنالوگ

- کارت شبکه جهت اتصال PtP

این کارت شبکه برای اتصال به شبکه مدباس استفاده می‌شود و دارای دو نوع اصلی است:

۱. CP341 که پروتکل ASCII و RTU مدباس را ساپورت می‌کند.
۲. CP340 که فقط پروتکل ASCII مدباس را ساپورت می‌کند.

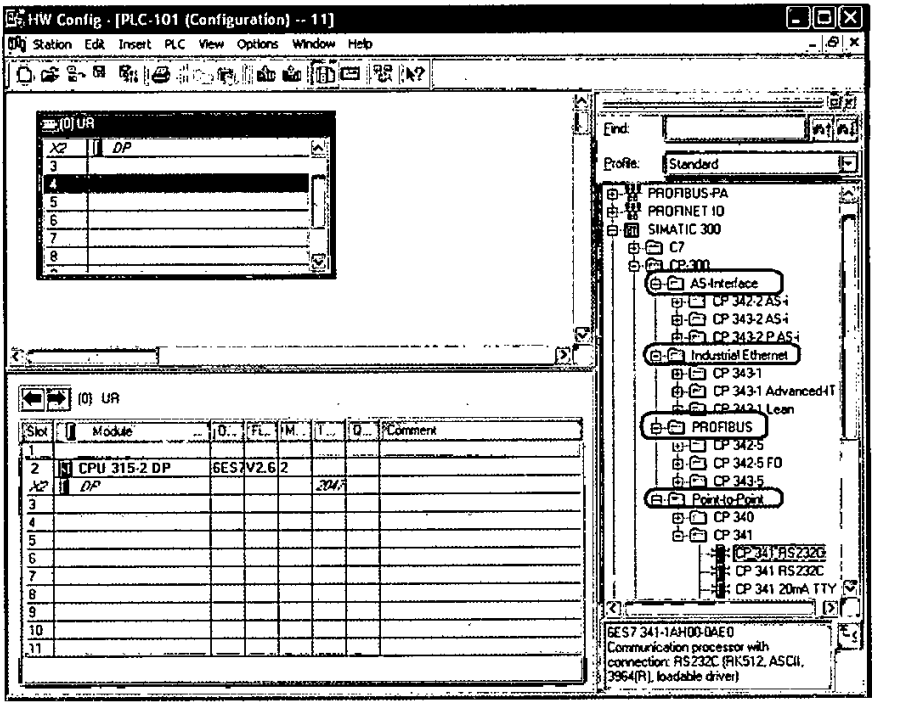
هر کدام از دو نوع فوق دارای انواع مختلفی است زیرا اتصال مدباس از نظر فیزیکی بسیار متنوع بوده و می‌تواند هر کدام از موارد زیر باشد:

- RS232
- RS422
- RS485
- 20mA TTY

بررسی تفاوت بین پروتکل‌ها و تفاوت بین ارتباط‌های سریال فوق خارج از چارچوب این کتاب است. علاقمندان به کتاب شبکه صنعتی مدباس تألیف محمدرضا ماهر – حمیدرضا رفیعیان مراجعه نمایند.

پیکر بندی CP در رک

وارد کردن کارت شبکه در رک (در محیط Hwconfig) در شکل ۱۱-۸۰ نشان داده شده است.

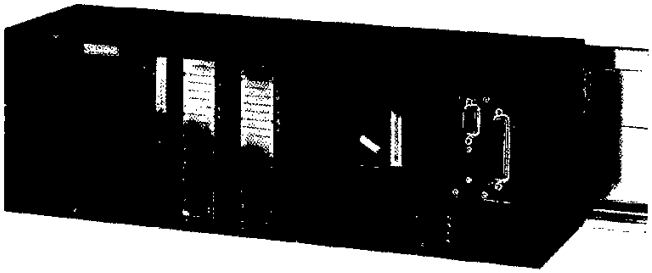


شکل ۱۱-۸۰ وارد نمودن کارت CP در رک

پس از وارد کردن کارت، هر کدام از کارت‌های شبکه تنظیمات خاص خود را نیاز دارند. به‌جز مدباس که برای تنظیماتش نرم‌افزار دیگری نیز موردنیاز است سایر شبکه‌ها از طریق Hwconfig و برنامه Netpro پیکربندی و تنظیم می‌شوند.

Function Module (FM) ۱۱-۱۳

همانطور که قبلاً اشاره شد Function Modules ماژول‌هایی هستند که فانکشن خاصی را مستقل از CPU پردازش و اجرا می‌کنند. ورودی‌ها مستقیماً به این ماژول‌ها داده می‌شوند و خروجی‌ها نیز مستقیماً از آنها دریافت می‌شوند، در عین حال FMها می‌توانند با CPU تبادل دیتا داشته باشند.



شکل ۱۱-۸۱ کارت FM در رک S7-300

برخی از موارد کاربرد FM عبارت است از:

- شمارش سریع
- کنترل موقعیت
- کنترل حلقه بسته به روش PID

در S7-300 فانکشن ماژول‌های مختلفی عرضه شده‌اند. برخی از آنها عبارتند از:

FM350-1 / FM350-2 (Counter Module)

کانتر برای شمارش‌های سریع هستند. پالس‌های فرکانس بالا که توسط کارت‌های DI قابل آشکارسازی نیستند به این ماژول‌ها داده می‌شوند.

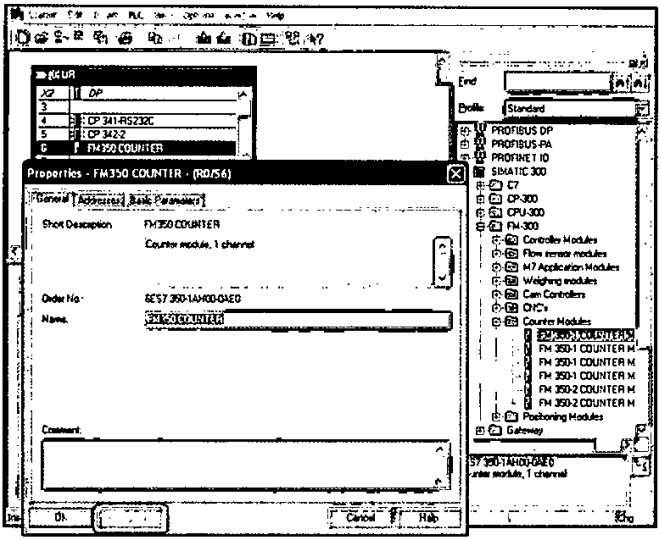
FM351 / FM353 / FM354 (Position Module)

برای کنترل موقعیت موتورهای پله‌ای و سرو موتورها و درایوهای لرزشی به‌کار می‌روند. دارای خروجی‌های دیجیتال برای کنترل موتورها هستند.

FM355 (Closed Loop Controller)

این ماژول یک لوپ کنترلر ۴ کاناله است که می‌تواند برای کنترل فشار و دما به‌کار رود. اگر به‌صورت Continues Control تنظیم شود از ۴ خروجی آنالوگ آن استفاده می‌شود و اگر به‌صورت Step Control تنظیم شود ۸ خروجی دیجیتال آن به‌کار برده می‌شود.

برای پیکربندی کارت‌های FM، آنها را در رک وارد می‌کنیم ولی برای تنظیمات مربوط به فانکشن داخلی FM علاوه بر Step7 پکیج پیکربندی جداگانه‌ای نیز لازم است. بدون نصب این پکیج گزینه Parameters در پایین پنجره پیکربندی مانند شکل ۱۱-۸۲ فعال نخواهد شد.



شکل ۱۱-۸۲ تنظیمات FM در محیط HW Config

۱۱-۱۴ Interface Module (IM)

۱۱-۱۴-۱ عملکرد

در برخی کاربردها ممکن است تعداد ماژول‌ها زیاد باشد، همان‌طور که بیان شد رک ۳۰۰ دارای یازده اسلات می‌باشد. اسلات اول برای PS و اسلات دوم برای CPU استفاده می‌شود، ماژول‌های I/O و ماژول های FM و CP و ... را فقط می‌توان در اسلات ۴ تا ۱۱ قرار داد. اگر در شرایطی همه اسلات‌های رک اصلی که CPU در آن قرار دارد پر شود و باز ماژول اضافی مورد نیاز باشد می‌توان از Expansion Rack که آنرا رک توسعه می‌نامیم استفاده کرد.

با نصب ریل جدید سایر ماژول‌ها روی آن قرار می‌گیرند. ارتباط بین رک اصلی و رک(های) توسعه توسط ماژولی به نام IM که مخفف Interface Module و به معنای ماژول واسط است صورت می‌پذیرد.

IM ها به‌صورت جفتی به‌کار می‌روند. روش کار به‌این صورت است که یک IM در رک اصلی و IM دیگر در رک توسعه قرار گرفته و ارتباط بین آنها توسط یک کابل مخصوص برقرار می‌شود.

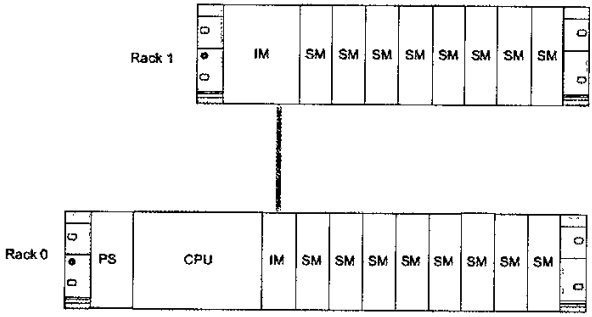
شکل ۱۱-۸۳ کابل اتصال IM

در یک رک توسعه نیز می‌توان هشت ماژول قرار داد، اگر باز هم ماژول اضافی مورد نیاز باشد می‌توان رک توسعه دیگری نیز به‌کار برد. رک اصلی معمولاً Rack0 و رک‌های دیگر از شماره یک یعنی Rack 1 شروع می‌شوند.

در مورد استفاده از IM باید به نکات زیر توجه نمود:

- وظیفه‌ی IM برقراری ارتباط رک اصلی و رک توسعه می‌باشد.
- در S7-300 حداکثر سه رک توسعه می‌توان استفاده نمود.
- برخی از CPUها مانند CPU312 رک توسعه را پشتیبانی نمی‌کنند که اصطلاحاً به آنها Single-Tire گفته می‌شود. CPUهایی که رک توسعه را پشتیبانی می‌کنند Multi- tire می‌نامند. این اصطلاح در ویژگی‌های CPU ذکر می‌گردد.
- ارتباط IMها با یکدیگر توسط کابل مخصوصی صورت می‌پذیرد.
- IM در اسلات ۳ قرار می‌گیرد.
- در صورت استفاده از چند IM، ارتباط بین آنها به‌صورت اتصال زنجیروار برقرار می‌گردد.

در شکل ۱۱-۸۴ استفاده از IM به‌منظور ایجاد رک توسعه را مشاهده می‌نمایید.



شکل ۱۱-۸۴ ارتباط بین رک اصلی و توسعه توسط IM

۱۱-۱۴-۲ انواع IM در S7-300

در S7-300 سه نوع IM وجود دارد که عبارتند از:

- IM 360 S: جهت استفاده در رک اصلی
- IM 361 R: جهت استفاده در رک توسعه
- IM 365 S-R: جهت استفاده در رک اصلی و توسعه

جدول ۱۱-۱۸ خصوصیات هر IM را به همراه برخی از جزئیات آن نشان می‌دهد.

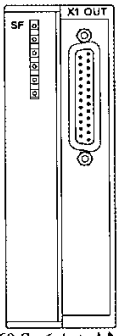
جدول ۱۱-۱۸ خصوصیات IMهای S7-300

	Interface module IM 360	Interface module Im 361	Interface module IM 365
قابل نصب در رک	0	1 to 3	0 and 1
قابل استفاده با	IM 361	IM 360	IM 365
طول کابل اتصال IM	max.10 m	Max. 10 m	1 m
خصوصیات	فقط در رک اصلی قرار می‌گیرد. فقط با IM 361 به‌کار می‌رود.	فقط در رک توسعه به‌کار می‌رود. در رک اصلی باید از IM 360 استفاده شده باشد.	استفاده از این IM به صورت جفتی با IM 365 می‌باشد. در رک توسعه برخی از نمونه‌هایCP و FM را نمی‌توان قرار داد.

IM 360 S

خصوصیات این IM عبارت است از:

- مخصوص قرارگیری در رک اصلی (رک شماره 0)
- فقط قابل اتصال به IM 361R می‌باشد.
- حداکثر فاصله بین این IM و IM361 R ده متر می‌باشد.
- دارای یک لامپ مربوط به خطا می‌باشد.
- در صورتی که کابل ارتباطی IM قطع شود یا IM361 R خاموش باشد، لامپ SF روشن می‌شود.
- دارای یک پورت اتصال می‌باشد که با علامت X1 نشان داده شده است.



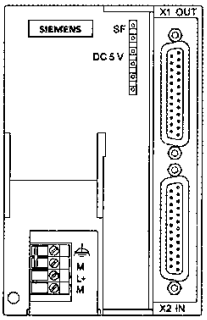
شکل ۱۱-۸۵ شماتیک IM 360 S

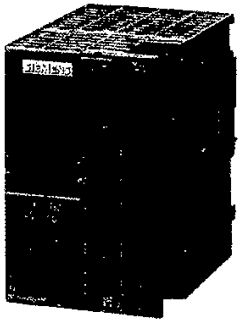
IM 361 R

خصوصیات این IM عبارت است از:

- مخصوص قرارگیری در رک توسعه (رک‌های شماره ۱ الی ۳)
- قابل اتصال به IM 360 S یا IM 361 R
- حداکثر طول کابل اتصال ده متر می‌باشد.

- نیاز به ولتاژ تغذیه‌ی 24 V DC
- دارای یک لامپ مربوط به خطا و یک لامپ نشان‌دهنده برقراری ولتاژ 5 V مربوط به باس می‌باشد.
- در صورتی که کابل ارتباطی IM قطع شود یا IM361 R که با این IM به‌طور سری وصل شده است خاموش باشد، لامپ SF روشن می‌شود.
- دارای‌دوپورت‌اتصال‌می‌باشد که پورت ورودی با علامت X2 و پورت خروجی با علامت X1 نشان داده شده است.



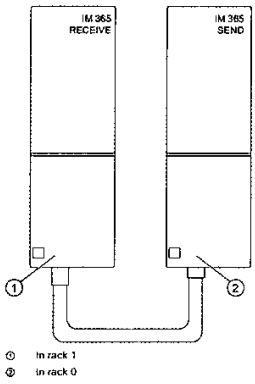


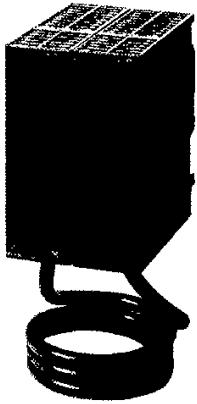
شکل ۱۱-۸۶ نمای واقعی و شماتیک IM 361 R

IM 365 S-R

خصوصیات این IM عبارت است از:

- به‌صورت جفتی با IM 365 S-R به‌کار می‌رود.
- یکی از IMها در رک اصلی و دیگری در رک توسعه قرار می‌گیرد.
- طول کابل اتصال فقط یک متر می‌باشد.





شکل ۱۱-۸۷ نمای واقعی و شماتیک IM 365

۱۱-۱۴-۳ روش‌های برقراری اتصال از طریق IM

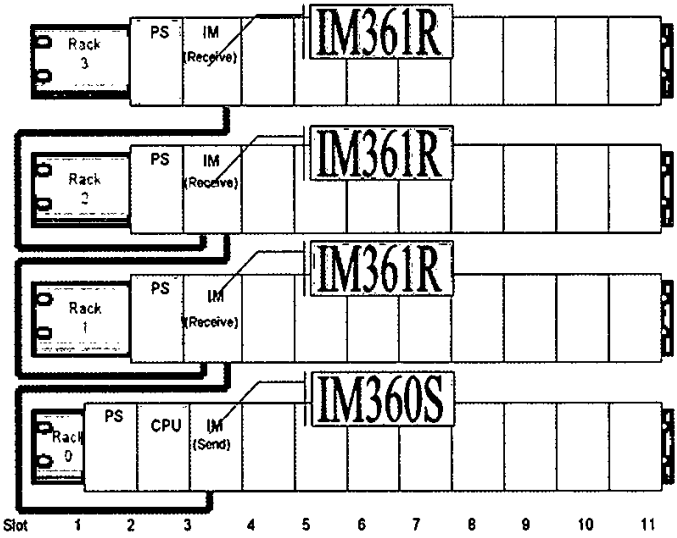
۱۱-۱۴-۳-۱ با استفاده از IM360S / IM361R

در این روش، IM 360S در رک اصلی و IM 361R در رک‌های توسعه نصب می‌شود. خصوصیات این روش عبارت است از:

- در این روش می‌توان حداکثر سه رک توسعه ایجاد نمود.
- رک‌های توسعه نیاز به منبع تغذیه دارند.
- طول کابل IM در این حالت می‌تواند یکی از چهارگزینه زیر باشد:

۱ متر ، ۲/۵ متر ، ۵ متر ، ۱۰ متر

- در رک‌های توسعه، علاوه بر SM می‌توان CP و FM نیز قرار داد.
- در صورت استفاده از چند رک توسعه، ارتباط IMها به‌صورت زنجیروار صورت می‌پذیرد.



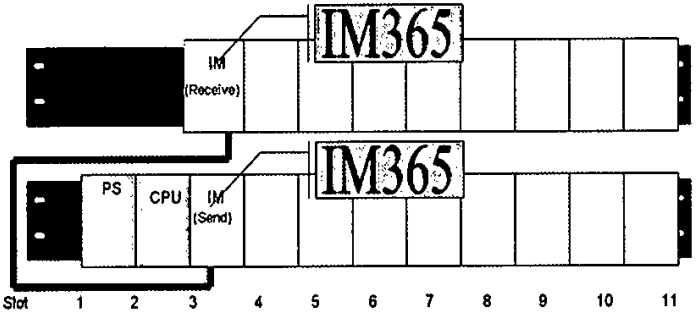
شکل ۱۱-۸۸ ایجاد سه رک توسعه از طریق IM 360 S و IM 361 R

۱۱-۱۴-۳-۲ با استفاده از دو IM365

در این روش یک IM 365 در رک اصلی و IM 365 دیگر در رک توسعه نصب می‌شود. این دو ماژول با کابل به‌هم متصل هستند و به‌صورت جفتی تهیه می‌شوند. خصوصیات این روش عبارت است از:

- می‌توان فقط یک رک توسعه ایجاد نمود.
- رک توسعه نیاز به منبع تغذیه ندارند.
- طول کابل IM در این حالت ۱ متر است.

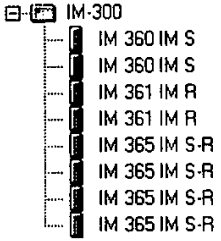
- در رک توسعه، CPهای مربوط به شبکه‌های پروفی‌باس و اترنت و برخی از FMها را نمی‌توان قرار داد.

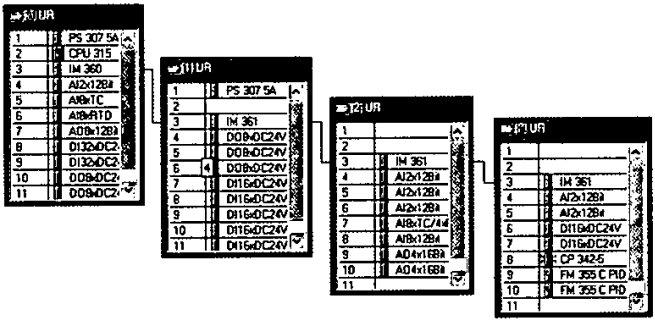


شکل ۱۱-۸۹ ایجاد یک رک توسعه با استفاده از IM365

۱۱-۱۴-۴ روش پیکربندی IM در hwconfig

مشابه سایر کارت‌ها، می‌توان IM را از کاتالوگ موجود در برنامه‌ی HW Config انتخاب و در اسلات شماره ۳ رک قرار داد. در شکل ۱۱-۹۰ سه رک توسعه که توسط IM361 با رک اصلی ارتباط دارند نشان داده شده است. در رک اصلی از IM360 استفاده شده است.

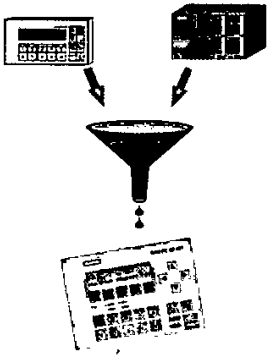




شکل ۱۱-۹۰ ارتباط سه رک توسعه و رک اصلی با استفاده از IM در محیط HW Config

۱۱-۱۵ پیکربندی سیستم‌های C7

همانطور که در فصل‌های قبل اشاره شد، سیستم‌های C7 در واقع ترکیبی از S7-300 همراه با یک پنل اپراتوری هستند که به‌صورت یکپارچه عرضه شده‌اند. برای پیکربندی سخت افزار C7 نیز از HW Config استفاده می‌شود ولی برای صفحات نمایشگر، نرم‌افزار Protool یا WinCC flexible مورد نیاز است.



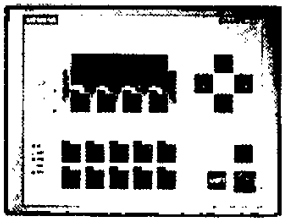
شکل ۱۱-۹۱ ساختار ترکیبی C7

C7 دارای انواع مختلف است و تفاوت آنها از نظر نوع PLC و نوع نمایشگر اپراتوری است. در جدول ۱۱-۱۹ مشخصه‌های اصلی برخی از انواع C7 مقایسه شده است.

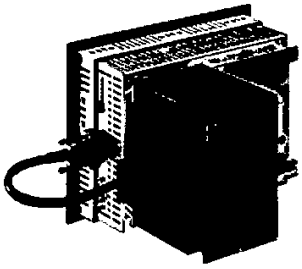
جدول ۱۱-۱۹ بررسی برخی از انواع C7

C7 Type	CPU Type	I/O	Display
C7 - 613	CPU 313	24DI , 16 DO , 4AI , 2 AO	Text Displat with 4x20 Characters
C7 - 635	CPU 314C-2DP	24DI , 16 DO , 4AI , 2 AO	Key or Touch Panel 6 "
C7 - 636	CPU315-2DP	24DI , 16 DO , 4AI , 2 AO	Key or Touch Panel 10 "

دربرخی از مدل‌های C7 می‌توان تعداد I/O را با اضافه‌کردن کارت I/O گسترش داد. به شکل ۱۱-۹۲ توجه کنید.



ب) C7- 613 از نمای روبرو

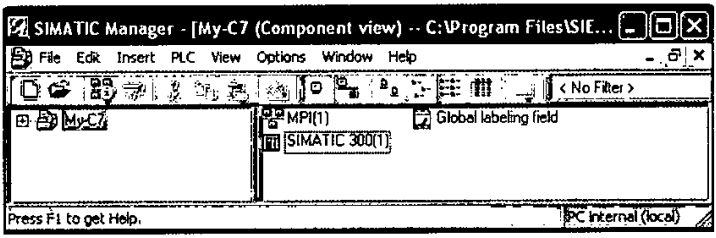


الف) C7 -613 با کارت اضافی

شکل ۱۱-۹۲ چند مدل مدل C7

پیکربندی سیستم C7 به‌سادگی انجام می‌شود. این کار در سه مرحله زیر خلاصه شده است:

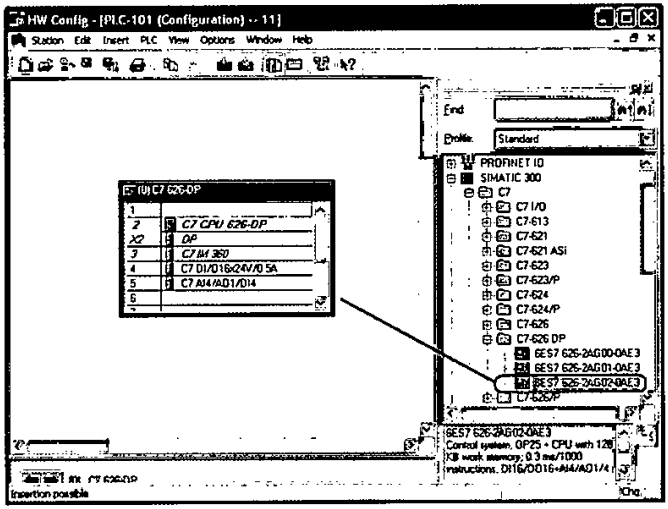
۱- وارد کردن یک Station 300 در Simatic Manager (مطابق شکل ۱۱-۹۳)



شکل ۱۱-۹۳ قرار دادن Station

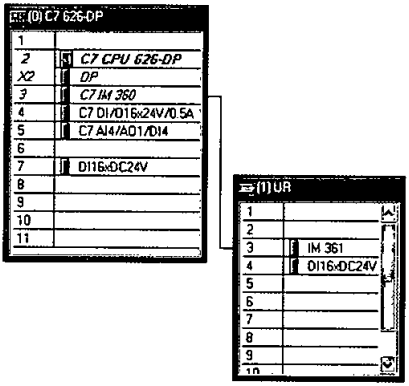
۲- باز کردن HW Config و وارد کردن C7 مورد نظر از کاتالوگ در آن

در اینصورت رک، CPU و ماژول‌های I/O همگی به‌طور خودکار وارد می‌شوند. در شکل ۱۱-۹۴، C7 مدل 626 وارد شده است. همانطور که دیده می‌شود، این سیستم مجهز به کارت‌های ورودی و خروجی دیجیتال و آنالوگ و IM360 برای توسعه است.



شکل ۱۱-۹۴ وارد نمودن C7 در محیط HW Config

۳- اضافه‌کردن کارت‌های I/O از زیر مجموعه SM (شبیه سیستم‌های S7-300) یا رک توسعه در صورت لزوم. (مطابق شکل ۱۱-۹۵)



شکل ۱۱-۹۵ ایجاد رک توسعه برای C7 با استفاده از IM 361

تنظیمات و سایر نکات ذکر شده برای ماژول‌های S7-300 در اینجا نیز صادق است.

۱۱-۱۶ پرسش و تحقیق

تنظیمات سخت افزاری و نصب سیستم M7 را بررسی کنید.

۱۱-۱۷ تست‌های خودآزمایی

۱- وظیفه رک در S7-300 کدام است؟

الف) ارتباط ماژول‌ها با یکدیگر
ب) نگهداری ماژول‌ها
ج) ارتباط و نگهداری ماژول‌ها
د) هیچ‌کدام

۲- در پیکربندی PLC، منظور از اسلات چیست؟

الف) فضای اشغالی توسط یک ماژول در رک
ب) فضای اشغالی توسط دو ماژول در رک
ج) فضای اشغالی توسط کارت حافظه
د) فضای اشغالی CPU

۳- وظیفه Bus Connector در S7-300 چیست؟

الف) نگهداری ماژول‌ها
ب) برقراری ارتباط بین ماژول‌ها
ج) برقراری ارتباط زمین ماژول‌ها از طریق رک
د) هر سه مورد

۴- در مورد نصب رک S7-300کدام مورد صحیح است؟

الف) رک به‌صورت عمودی و دمای ۰ الی ۶۰ درجه سانتی‌گراد نصب شود.
ب) رک به‌صورت عمودی و دمای ۰ الی ۴۰ درجه سانتی‌گراد نصب شود.
ج) رک به‌صورت افقی و دمای ۰ الی ۴۰ درجه سانتی‌گراد نصب شود.
د) رک به‌صورت افقی و دمای ۰ الی ۶۰ درجه سانتی‌گراد نصب شود.

۵- در پیکربندی S7-300 منبع تغذیه در اسلات . . . قرار گرفته و وارد نمودن آن در HW Config . . . می‌باشد.
الف) دوم – اختیاری ب) اول – اجباری
ج) دوم – اجباری د) اول- اختیاری

۶- کدامیک از منابع تغذیه زیر در S7-300 وجود ندارد؟
الف) 2A ب) 5A ج) 10A د) 20A

۷- مهم‌ترین واحد موجود در یک PLC است که بدون آن امکان عملکرد PLC وجود ندارد.
الف) FM ب) CPU ج) IM د) CP

۸- تفاوت CPUهای Compact نسبت به CPUهای Modular در چیست؟
الف) وجود ماژول‌های I/O به‌صورت یکپارچه با CPU
ب) وجود پورت DP به‌منظور ارتباط با شبکه پروفی‌باس
ج) قابلیت انجام برخی از وظایف FM، مانند شمارش سریع و کنترل موقعیت
د) موارد الف و ج

۹- حروف DP در کد نام CPU بیانگر کدام موضوع زیر است؟
الف) وجود پورت MPI ب) وجود پورت برای ارتباط با شبکه صنعتی پروفی‌باس
ج) وجود پورت برای ارتباط با شبکه صنعتی پروفی‌نت د) وجود پورت برای ارتباط به‌صورت Point to Point

۱۰- در سیگنال الکتریکی در کارت DI به کدام مورد زیر تبدیل می‌شود؟
الف) دیتای ۱۶ بیتی ب) دیتای صفر منطقی
ج) دیتای یک منطقی د) دیتای صفر و یک منطقی

۱۱- برای ساخت سنسور Namur کدام روش درست است؟
الف) سری کردن مقاومت با یک سوئیچ معمولی ب) موازی کردن مقاومت با یک سوئیچ معمولی
ج) سری- موازی کردن یک مقاومت با یک سوئیچ معمولی د) هر سه مورد

۱۲- مشخصه کارت‌های DO از نوع رله‌ای چیست؟
الف) سرعت سوئیچینگ بالا ب) سرعت سوئیچینگ پایین
ج) قابلیت جریان دهی بالا د) موارد ب و ج

۱۳- وظیفه A/D در کارت AI چیست؟
الف) تبدیل دیتا به سیگنال الکتریکی ب) تبدیل سیگنال الکتریکی به سیگنال الکتریکی
ج) تبدیل دیتا به دیتا د) تبدیل سیگنال الکتریکی به دیتا

۱۴- برای اندازه‌گیری دماهای پایین استفاده از کدام نوع سنسور بهتر است؟
الف) RTD ب) ترموکوپل ج) ترانسمیتر دما د) موارد الف وب

۱۵- هر آدرس آنالوگ در PLC چند بایت اشغال می‌کند؟
الف) یک بایت ب) دو بایت ج) سه بایت د) چهار بایت

۱۶- به‌منظور رزرو یک اسلات در Rack-300، جهت ماژولی که قرار است در آینده اضافه شود،از چه ماژولی استفاده می‌شود؟
الف) SM ب) DM ج) IM د) FM

۱۷- ماژول ..... برای انجام عملی خاص، به‌طور مستقل از CPU به‌کار می‌رود.

الف) SM ب) DM ج) IM د) FM

۱۸- به‌منظور ارتباط PLC با شبکه‌های صنعتی از چه ماژولی استفاده می‌شود؟

الف) CPU ب) DM ج) CP د) FM

۱۹- ارتباط بین رک اصلی و توسعه توسط چه ماژولی صورت می‌پذیرد؟

الف) IM ب) DM ج) CP د) FM

فصل ۱۲

مفاهیم پایه در عملکرد PLC

۱۲-۱ مقدمه

۱۲-۲ مدهای کاری PLC

۱۲-۲-۱ مد Stop

۱۲-۲-۲ مد Startup

۱۲-۲-۳ مد RUN

۱۲-۲-۴ جایگزینی RUN-P به‌جای RUN

۱۲-۲-۵ مد HOLD

۱۲-۲-۶ اولویت مدهای کاری CPU

۱۲-۳ بخش‌های حافظه

۱۲-۳-۱ Load Memory

۱۲-۳-۲ Work Memory

۱۲-۳-۳ System Memory

۱۲-۴ راه‌اندازی مجدد CPU

۱۲-۴-۱ Cold Restart

۱۲-۴-۲ Warm Restart

۱۲-۴-۳ Hot Restart

۱۲-۵ ری‌ست کردن CPU

۱۲-۵-۱ عملکرد Reset

۱۲-۵-۲ Reset با سوئیچ MRES

۱۲-۵-۳ Reset از طریق نرم‌افزار

۱۲-۶ تاثیر ری‌ست و راه اندازی روی بخش‌های مختلف حافظه

۱۲-۷ تاثیر عملکرد CPU روی LEDهای آن

۱۲-۸ پرسش و تحقیق

۱۲-۹ تست‌های خودآزمایی

در این فصل مفاهیم پایه عملکرد CPU که آشنایی با آنها برای درک مطالب فصل‌های بعد لازم است، تشریح گردیده است.

فصل ۱۲

چکیده مطالب

- CPU در مد Stop برنامه را اجرا نمی‌کند.
- وقتی CPU از Stop به RUN می‌رود ابتدا به مد Startup رفته و دستورات راه‌اندازی را اجرا می‌کند، سپس نتیجه را به خروجی می‌فرستد.
- CPU پس از راه اندازی به سیکل اسکن وارد می‌شود. ورودی‌ها را می‌خواند و در حافظه قرار می‌دهد. برنامه را اجرا می‌کند و سپس نتایج را به خروجی‌ها می‌فرستد.
- حافظه CPU به سه بخش Load ، Work و System تقسیم می‌شود.
- برنامه و اطلاعات سخت‌افزاری به Load Memory دانلود می‌شوند و از همین حافظه قابل آپلود هستند.
- برنامه اجرایی از Load Memory به Work Memory منتقل می‌شود.
- حافظه ورودی‌ها و خروجی‌ها و متغیرهای حافظه و تایمر و کانتر در System Memory قرار دارد.
- بخشی از System Memory را می‌توان به‌صورت ماندگار تعریف کرد.
- در اکثر CPUها حافظه Work و System قابل افزایش نیست.
- حافظه Load Memory درCPUهای جدید S7-300 فقط همان کارت حافظه MMC است .
- حافظه Load Memory را تا حد مشخصی می‌توان افزایش داد.
- ری‌ست کردن حافظه با کلید روی CPU دارای ترتیب خاصی از عملیات است.
- ری‌ست کردن می‌تواند از طریق برنامه انجام شود.
- راه اندازی CPU پس از وصل تغذیه می‌تواند به‌صورت Cold یا Warm یا Hot باشد.
- در حالت Cold محتوای System Memory پاک می‌شود و برنامه از ابتدا شروع می‌گردد.
- در حالت Warm فقط بخشی از System Memory که ماندگار تعریف نشده پاک می‌شود و برنامه از ابتدا شروع می‌شود.
- در حالت Hot هیچ حافظه‌ای پاک نمی‌شود و برنامه از محل قطع شده ادامه می‌یابد. این نوع راه اندازی خاص S7-400 است و نیاز به باتری پشتیبان دارد.

اصطلاحات و تعاریف

Download
به انتقال اطلاعات از کامپیوتر به CPU دانلود گفته می‌شود.

Upload
به انتقال اطلاعات از PLC به کامپیوتر آپلود گفته می‌شود.

Buffer
حافظه‌ای برای ذخیره‌سازی موقت اطلاعات که با رسیدن اطلاعات جدید اطلاعات قبلی آن پاک می‌شود.

فصل ۱۲

۱-۱۲ مقدمه

در فصل‌های قبل عملکرد PLC به‌طور کلی مورد بحث قرار گرفت. اگرچه نکات ذکر شده در مورد مفاهیمی مانند سیکل اسکن و حافظه برای PLCهای ساخت زیمنس نیز مصداق دارند ولی در این فصل به‌صورت تخصصی به تشریح عملکرد PLCهای S7 و مفاهیم مربوط به آن می‌پردازیم.

این مفاهیم پایه و اساس بحث‌های بعدی که در ارتباط با برنامه‌نویسی و عیب‌یابی و . . . هستند، به‌شمار می‌آیند. در این فصل مفاهیمی مانند سیکل اسکن، مدهای کاری، حافظه و . . . مورد بحث قرار می‌گیرد.

۲-۱۲ مدهای کاری PLC

وقتی ولتاژ تغذیه PLC وصل و پیکربندی سخت‌افزار و برنامه به آن دانلود شده باشد، مدهای کاری مختلفی برای CPU پیش می‌آید. این مدهای کاری توسط سوئیچی که روی CPU تعبیه شده یا از طریق ابزارهای نرم‌افزار قابل تغییر هستند. تغییر در مدهای کاری روی وضعیت LEDهای CPU تاثیر می‌گذارد.

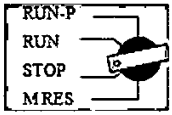
۱-۲-۱۲ مد Stop

در این مد پردازش برنامه متوقف می‌شود، دسترسی به I/Oها وجود ندارد، به‌عبارت دیگر تغییرات ورودی ها توسط CPU دیده نمی‌شود و هیچ فرمانی نیز توسط CPU به خروجی‌ها ارسال نمی‌شود. در مد Stop می‌توان به PLC دانلود کرد، همچنین می‌توان از PLC اطلاعات را گرفت یعنی upload نمود. به‌عبارت دیگر CPU به‌صورت Read/Write در دسترس است. کاربر به دو روش می‌تواند CPU را به مد Stop ببرد:

- از طریق کلید روی CPU
- از طریق نرم‌افزار

شکل ۱۲-۱ وضعیت کلید روی CPU را در حالت Stop نشان می‌دهد. در CPUهای قدیمی که کلید به‌صورت چرخشی است می‌توان پس از قرار دادن CPU در وضعیت Stop، آنرا از محل خود خارج نمود تا امکان تغییر وضعیت توسط افراد ناوارد وجود نداشته باشد. در CPUهای جدید امکان خارج کردن سوئیچ وجود ندارد.

کلید ۳ وضعیتی روی CPUهای جدید

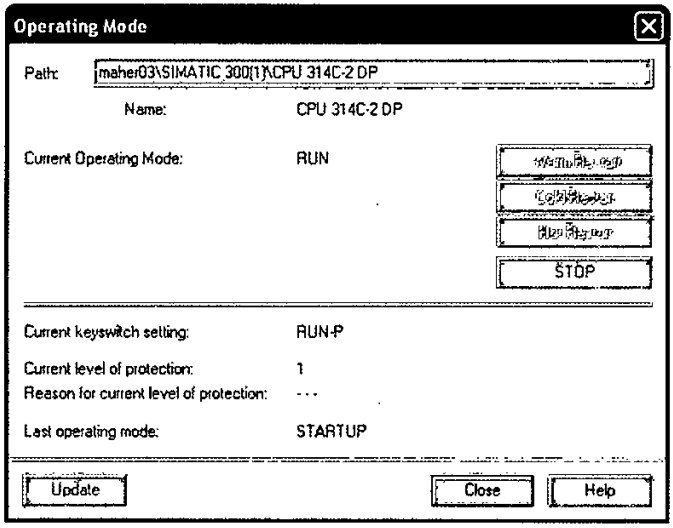


کلید ۴ وضعیتی روی CPUهای قدیمی

شکل ۱۲-۱ سوئیچ CPUهای قدیمی و جدید

از طریق نرم‌افزار نیز می‌توان CPU را به مد Stop برد. اگر CPU در حال کار باشد می‌توان با اتصال کامپیوتر به PLC، از هر نقطه‌ای توسط پنجره‌های مختلف نرم‌افزار Step7 آنرا به مد Stop برد. این کار در محیط Simatic Manager و محیط HW Config و محیط برنامه‌نویسی و حتی محیط پیکربندی شبکه امکان پذیر است. در تمام این

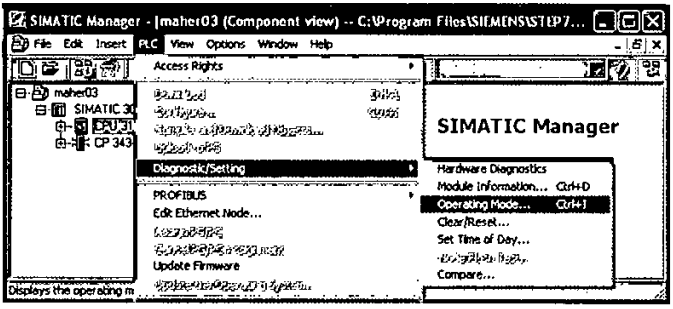
محیط‌ها کافیست از منوی PLC گزینه Operating Mode را انتخاب کرده و روی گزینه Stop کلیک کنیم یا از کلید میانبر Ctrl+I استفاده کنیم.

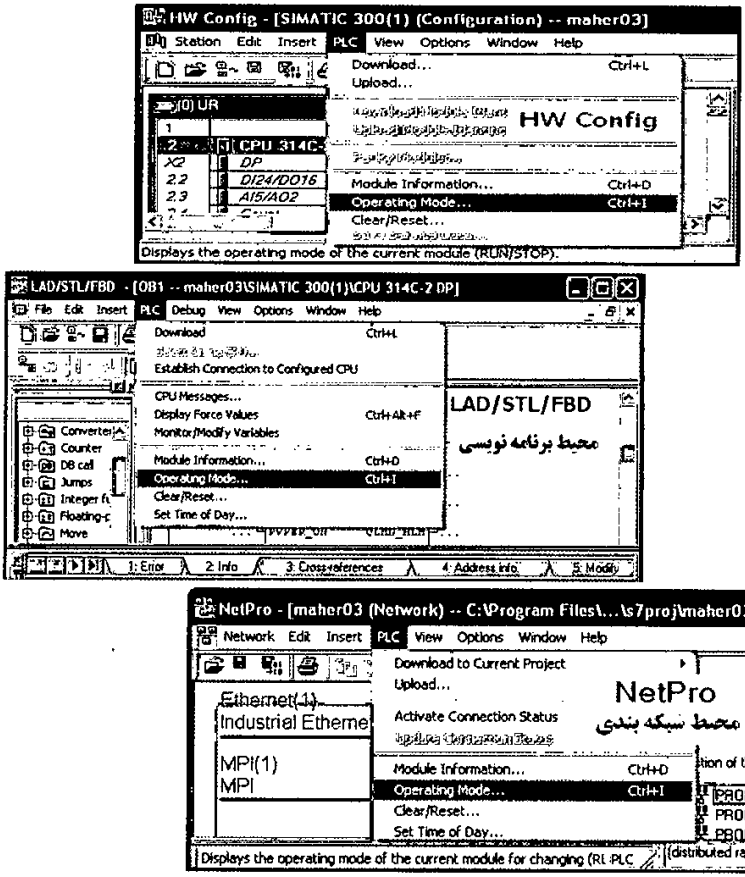


شکل ۱۲-۲ تغییر مدکاری CPU از طریق نرم‌افزار

توجه شود که اگر ارتباط بین PC و PLC برقرار نباشد یا تغذیه PLC وصل نباشد، این پنجره ظاهر نمی‌شود و به‌جای آن پیغام خطا را خواهیم داشت.

شکل ۱۲-۳ محل دسترسی به این گزینه را در برنامه‌های مختلف نشان می‌دهد. شایان ذکر است که به‌جز محیط برنامه‌نویسی در سایر موارد لازم است ابتدا روی CPU کلیک کنیم تا گزینه Operating Mode فعال شود.





شکل ۱۲-۳ امکان تغییر مدکاری CPU از محیط نرم‌افزارهای مختلف

در شرایط Stop بایستی توجه داشت که:

- وقتی CPU در مد Stop قرار می‌گیرد، چراغ Stop روی CPU روشن می‌شود. هرگاه با یک PLC مواجه می‌شویم بایستی به وضعیت چراغ‌ها توجه کنیم نه به وضعیت کلیدها، زیرا ممکن است کلید روی CPU در وضعیت Run باشد ولی CPU توسط نرم‌افزار Stop شده باشد.
- ممکن است Stop شدن CPU ناشی از فرمان توقف در برنامه باشد و کاربر دخالتی در تغییر مد نداشته باشد.
- در برخی حالات دیگر ممکن است بروز فالت منجر به Stop شدن CPU شده باشد. در این حالت علاوه بر چراغ Stop، چراغ فالت نیز روشن است.

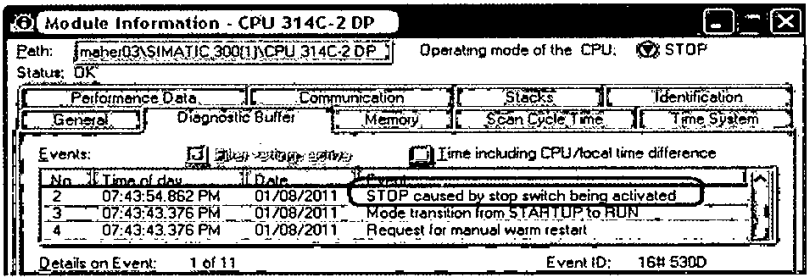
تذکر : هر اتفاقی که برای CPU رخ می‌دهد در بافر خاصی که برای عیب‌یابی تعبیه شده ذخیره می‌گردد. شکل ۱۲-۴ پیام بافر را برای Stop شدن CPU در چهار حالت مختلف نشان می‌دهد.

جزئیات این پنجره که به Module Information موسوم و با کلید میانبر Ctrl+D قابل مشاهده است، در فصل عیب‌یابی تشریح شده است.

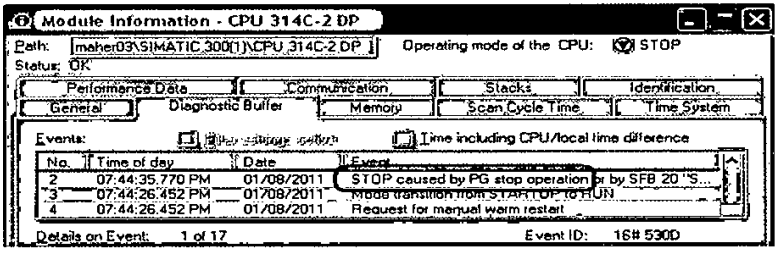
۱۲-۲-۲ مد Startup

وقتی CPU توسط کلید روی آن یا از طریق نرم‌افزار راه‌اندازی می‌شود (یعنی در حال وارد شدن به وضعیت Run است) در اصطلاح گفته می‌شود CPU در مد Startup قرار گرفته است. مد Startup شرایط گذرایی است که در آن موارد زیر اتفاق می‌افتد:

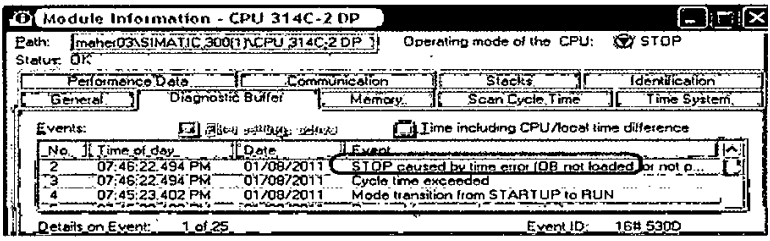
- وضعیت سخت‌افزار چک می‌شود.
- اگر برنامه‌ای برای راه‌اندازی نوشته شده باشد اجرا می‌گردد.
- بسته به نوع تنظیم راه‌اندازی ممکن است بخش‌هایی یا تمام حافظه سیستم پاک شود که این تنظیم در ادامه توضیح داده خواهد شد.



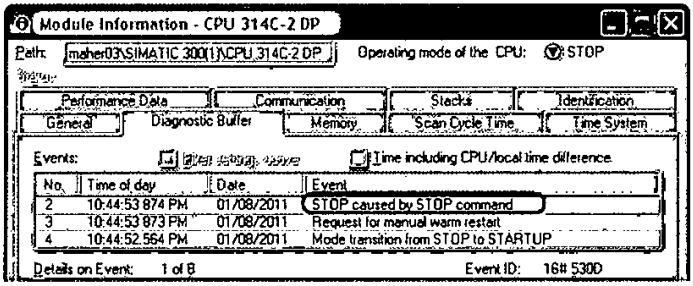
پیام ثبت شده در بافر وقتی CPU توسط سوئیچ Stop شود.



پیام ثبت شده در بافر وقتی CPU توسط نرم افزار Stop شود.



نمونه پیام ثبت شده در بافر وقتی CPU به علت بروز فالت Stop شود.

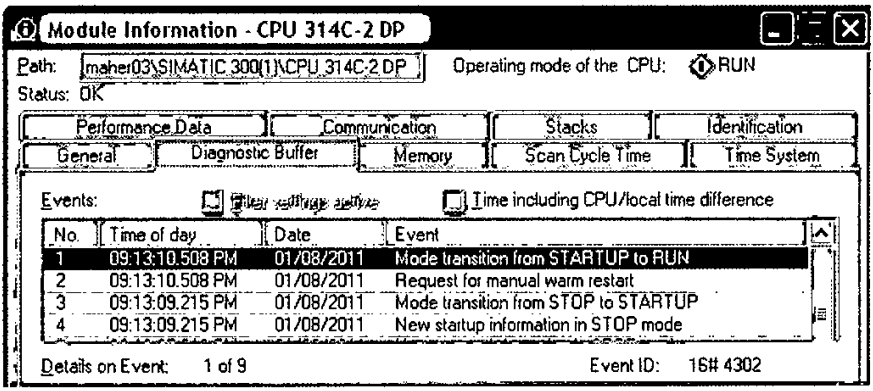


پیام ثبت شده در بافر در صورت فرمان توقف در برنامه نویسی

شکل ۱۲ – ۴ پیام‌های ثبت شده در بافر برای چهار حالت مختلف Stop شدن CPU

نکات دیگری که در مد راه‌اندازی بایستی به آن توجه داشت عبارتند از:

- برنامه راه‌اندازی در یکی از بلاک‌های OB100 یا OB101 یا OB102 نوشته می‌شود. تفاوت بین این سه بلاک در ادامه ذکر می‌شود ولی تشریح برنامه‌نویسی آنها در کتاب سطح پیشرفته آمده است.
- در مد راه‌اندازی، CPU هنوز به I/Oها دسترسی پیدا نکرده است. اگر در برنامه راه‌اندازی دستور خواندن ورودی‌ها نوشته شود، عملاً مقداری در برنامه وارد نمی‌شود. اگر در برنامه راه‌اندازی فرمان به خروجی‌ها نوشته شود، پس از اینکه CPU از مد راه‌اندازی خارج و به مد RUN وارد شد، فرمان به خروجی اعمال خواهد شد.
- در مد راه‌اندازی چراغ RUN روی CPU چشمک‌زن می‌شود.
- مدت زمان راه‌اندازی بسته به شرایط مختلف، متفاوت است. اگر برنامه راه‌اندازی طولانی باشد یا اگر اولین بار است که CPU نسبت به شناسایی سخت‌افزار یا تجهیزات پروفی‌باس اقدام می‌کند، زمان بیشتری طول خواهد کشید.
- در حین راه‌اندازی، CPU قابل نوشتن است یعنی می‌توان به آن دانلود کرد.
- به‌طور نرمال CPU پس از مد Startup وارد مد RUN می‌شود ولی اگر مشکلی وجود داشته باشد یا اگر دستور توقف در برنامه راه‌اندازی نوشته شده باشد، CPU از مد Startup به مد Stop برمی‌گردد.
- مراحل راه‌اندازی در بافر CPU ثبت می‌شود. به شکل ۱۲–۵ توجه کنید.

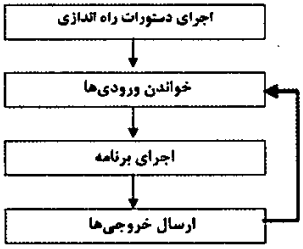


شکل ۱۲ – ۵ پیام‌های ثبت شده در بافر برای راه اندازی CPU

۱۲-۲-۳ مد RUN

وقتی CPU از مد راه‌اندازی عبور کرد وارد مد RUN می‌شود. در این مد، CPU به اجرای سیکل‌اسکن که یک کار مشخص و تکراری است می‌پردازد و تا زمانی که به مد Stop نرفته اجرای سیکل را ادامه می‌دهد. در مد RUN فقط امکان خواندن از CPU وجود دارد و CPU به‌صورت Read Only است. یعنی امکان آپلود داریم ولی امکان دانلود خیر، در این مد چراغ RUN روی CPU به‌صورت ثابت روشن است.

به‌طور کلی در این سیکل کاری مراحل زیر به‌ترتیب اجرا می‌شوند:



شکل ۱۲-۶ سیکل کاری CPU

مراحلی که در شکل ۱۲-۶ مشاهده می‌شود، عبارتند از:

۱. در ابتدا دستورات راه‌اندازی اجرا می‌شوند. این دستورات فقط در زمان راه‌اندازی CPU اجرا شده (سیکل کاری اول) و در سایر سیکل‌ها اجرا نمی‌شوند.

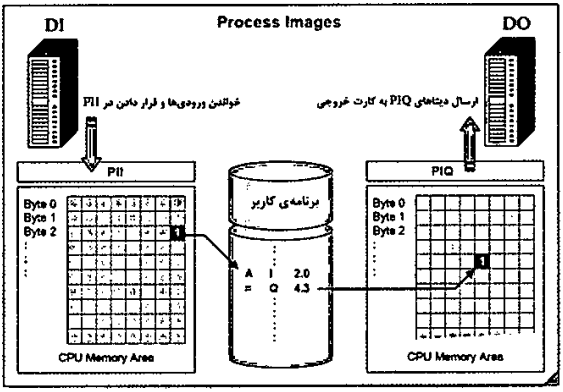
۲. مقدار ورودی‌ها از کارت‌های ورودی خوانده شده و در PII قرار می‌گیرد. PII مخفف Process Image Input است و معرف بخشی از حافظه CPU است که ورودی‌ها در آن ذخیره می‌شوند.

۳. پس از قرارگیری مقدار لحظه‌ای متغیرها در PII، برنامه‌ی نوشته شده در CPU اجرا می‌شود. در ضمن اجرای برنامه‌ی CPU، هر جا اطلاعات ورودی مورد نیاز بود، CPU به جدول PII مراجعه نموده و مقدار مورد نظر را از آن می‌خواند.

۴. در ضمن اجرای برنامه اگر نیاز به ایجاد فرمانی جهت خروجی‌ها باشد، این فرمان در ناحیه‌ی PIQ قرار داده می‌شود. PIQ معرف Process Image Output است که فرمان‌های ارسالی به خروجی در آن ذخیره می‌شود.

۵. پس از اینکه اجرای برنامه تمام شد، CPU مقادیر موجود در PIQ را به کارت‌های خروجی ارسال می‌کند.

در شکل۱۲-۷، ترتیب عملیات سیکل اسکن CPU نشان داده شده است.

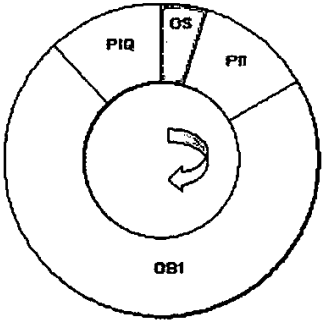


شکل ۱۲-۷ ترتیب عملیات سیکل اسکن CPU

باید توجه نمود که برخی از ورودی‌های PLC در ناحیه PII ذخیره نمی‌شوند بلکه CPU در صورت نیاز آنها را به‌طور مستقیم از ماژول مربوطه می‌خواند. به این نوع ورودی‌ها اصطلاحاً Peripheral Input گفته می‌شود.

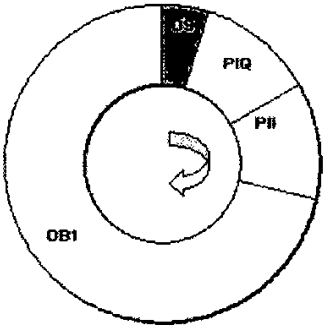
به‌طور مشابه برخی از خروجی‌ها بدون اینکه از PIQ فرمان بگیرند به‌طور مستقیم در صورت لزوم از CPU مقدار می‌گیرند. به این نوع خروجی‌ها اصطلاحاً Peripheral Outputs گفته می‌شود.

به‌طور کلی سیکل کاری CPU را که تا سال ۱۹۹۸، زیمنس از آن پیروی می‌کرد به‌صورت نشان داده شده در شکل ۱۲-۸ می‌توان ارائه نمود که در آن OS معرف Operating Systam یعنی سیستم عامل CPU می‌باشد.



شکل ۱۲-۸ سیکل اسکن تا سال ۱۹۹۸

در این روش اگر در حین راه‌اندازی فرمانی برای خروجی خاصی در برنامه راه‌اندازی صادر شود، بلافاصله به آن اعمال نمی‌شود و لازم است CPU به انتهای سیکل برسد تا فرمان را اعمال کند. در عین حال ممکن است در طول سیکل، فرمان توسط برنامه تغییر کند. برای رفع این مشکل از سال ۱۹۹۸ به بعد روال سیکل اسکن همانطور که در شکل ۱۲-۹ دیده می‌شود، تغییر کرد. این تغییر عبارت است از جابه‌جایی مرحله‌ی ارسال خروجی‌ها با خواندن ورودی‌ها، یعنی در ابتدا و پس از اجرای دستورات راه‌اندازی، مقدار موجود در PIQ به خروجی‌ها ارسال شده، سپس مقدار ورودی‌ها خوانده شده و در PII قرار می‌گیرد و سپس برنامه اجرا می‌شود. این ترتیب در حین کار نرمال هیچ تفاوتی با حالت قبلی ندارد، تفاوت فقط در لحظه ورود از مد Startup به مد RUN است که فرامین تولید شده به خروجی‌ها منتقل شده سپس سیکل شروع می‌گردد.



شکل ۱۲-۹ سیکل اسکن از سال ۱۹۹۸ به بعد

زمان سیکل‌اسکن

به مدت زمانی که طول می‌کشد تا یک سیکل کاری توسط CPU اجرا شود، زمان سیکل‌اسکن گفته می‌شود. این زمان از جمع زمان‌های زیر به‌دست می‌آید:

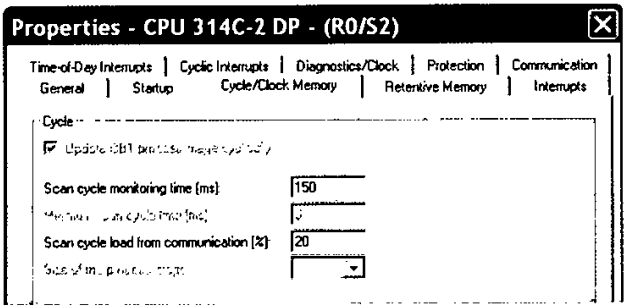
- زمان مربوط به خواندن ورودی‌ها و قرار دادن در PII (Update کردن PII)
- زمان مربوط به ارسال خروجی‌ها از PIQ به کارت‌های خروجی
- زمان مربوط به اجرای برنامه (پردازش برنامه)
- زمان مربوط به OS (سیستم عامل)
- زمان مربوط به تبادل دیتا با شبکه
- وقفه‌های پیش‌آمده در زمان کار CPU مثل فراخوانی OBهای وقفه

زمان مجاز سیکل‌اسکن

از آنجا که در طول زمان اجرای برنامه‌ی اصلی توسط CPU، مقادیر موجود در PII به‌روز رسانی نمی‌شوند، باید زمان سیکل‌اسکن زمان کمی باشد تا امکان به‌روز رسانی سریع PII و PIQ وجود داشته باشد. این زمان با توجه به پروسه‌های

مختلف متفاوت است. در هر صورت کاربر می‌تواند زمانی را به عنوان حداکثر زمان مجاز سیکل‌اسکن تعریف نماید که در صورت افزایش زمان سیکل‌اسکن CPU از این زمان، یک وقفه به CPU اعمال شود.

زمان مجاز سیکل‌اسکن درCPUهای S7-300 و در بسیاری از CPUهای S7-400 به‌طور پیش‌فرض 150 میلی‌ثانیه می‌باشد که امکان تغییر آن توسط کاربر نیز وجود دارد. برای تنظیم این زمان کافیست درHWconfig روی CPU دوبار کلیک کرده تا سربرگ Cycle/Clock مانند شکل زیر نمایش داده شود.



شکل ۱۲-۱۰ تنظیمات زمان سیکل اسکن در CPU314C-2DP

حداقل زمان سیکل اسکن

در CPUهای S7-400 امکان اختصاص زمانی وجود دارد که به‌عنوان حداقل زمان سیکل‌اسکن در نظر گرفته می‌شود. در صورتی‌که مدت زمان سیکل‌اسکن از مقدار تعریف شده کمتر باشد، OS (سیستم‌عامل) سیکل‌اسکن بعدی را پس از سپری‌شدن زمان مورد نظر آغاز می‌کند. مثلاً اگر در یک PLC حداقل زمان سیکل‌اسکن 30 میلی‌ثانیه تعریف شود و سیکل‌اسکن جاری 20 میلی‌ثانیه طول بکشد، OS (سیستم عامل CPU) به اندازه‌ی 10 ثانیه صبر نموده و سپس سیکل‌اسکن بعد را اجرا می‌کند. این تنظیم برای 300 همانطور که در شکل ۱۲-۱۰ دیده می شود، غیر فعال است.

۱۲-۲-۴ جایگزینی RUN-P به‌جای RUN

در CPUهای قدیمی علاوه بر مد RUN وضعیت دیگری با عنوان RUN-P وجود داشت که با کلید سلکتوری روی CPU قابل انتخاب بود. در مد RUN-P همانند RUN، سیکل اسکن در حال اجراست ولی تفاوت آن با RUN در این است که CPU هم قابل خواندن است و هم قابل نوشتن. یعنی در مد RUN-P می‌توان علاوه بر آپلود، به CPU دانلود نیز کرد.

در CPUهای جدید اگر چه به‌ظاهر کلمه RUN-P حذف شده و RUN باقی مانده است ولی عملاً RUN همان مفهوم RUN-P را دارد، یعنی با قرار دادن کلید در این وضعیت هم امکان دانلود و هم امکان آپلود وجود دارد.

توجه شود که در مد RUN-P:

- آپلود از CPU مشکلی برای عملکرد CPU ایجاد نمی‌کند.

- دانلود برنامه‌ای که فاقد خطا باشد مشکلی ایجاد نمی‌کند. در واقع CPU سیکل فعلی را با برنامه قبلی به اتمام می‌رساند و سیکل جدید را با برنامه جدید شروع می‌کند.
- دانلود اطلاعات سخت‌افزاری منجر به توقف CPU می‌شود. معمولاً در هنگام دانلود سخت‌افزار همانطور که در فصل ۲۵ تشریح خواهد شد، کاربر با هشداری مبنی براینکه CPU متوقف خواهد شد روبرو می‌گردد. در صورت تایید آن، CPU به مد Stop رفته و پس از کامل شدن دانلود، نرم‌افزار از کاربر سوال می‌کند که آیا دوباره سیستم راه‌اندازی شود یا خیر؟ فقط در سیستم‌های افزونه S7-400H است که می‌توان در حین کار بدون توقف فرآیند سخت‌افزار را نیز دانلود کرد که البته تنظیمات و نکات خاصی نیاز دارد.

۱۲-۲-۵ مد HOLD

در وضعیت Hold، پردازش برنامه کاربر متوقف شده و می‌توان برنامه را به‌صورت مرحله به مرحله تست نمود. یعنی می‌توان CPU را مجبور کرد که تا کاربر اجازه نداده سطر بعدی برنامه را اجرا نکند. از این وضعیت به‌منظور تست و عیب‌یابی برنامه استفاده می‌گردد و استفاده از آن در شرایط نرمال فرآیند ممکن است خطرناک باشد. نحوه استفاده از این وضعیت کاری، در کتاب سطح تکمیلی به‌طور کامل تشریح خواهد شد.

۱۲-۲-۶ اولویت مدهای کاری CPU

اگر چند وضعیت کاری همزمان به CPU اعمال شوند، وضعیتی انتخاب می‌گردد که دارای اولویت بیشتری باشد. مثلاً اگر CPU در حالت RUN قرار داشته باشد و توسط کامپیوتری که به PLC متصل است وضعیت کاری Stop انتخاب گردد، CPU متوقف می‌شود. جدول ۱۲-۱ اولویت‌های کاری CPU را نشان می‌دهد.

جدول۱۲-۱ اولویت وضعیت‌های کاری مختلف

وضعیت کاری	اولویت
STOP	بیشترین اولویت
HOLD	
Startup	
RUN	کمترین اولویت

۱۲-۳ بخش‌های حافظه

به‌طور کلی نواحی حافظه در CPU به سه قسمت زیر تقسیم‌بندی می‌شود:

۱- Load Memory
۲- Work Memory
۳- System memory

در شکل ۱۲-۱۱ نواحی مختلف حافظه به همراه جزئیات آن مشاهده می‌گردد.

Load Memory	Work Memory	SystemMemory
شامل	شامل	شامل
کل برنامه کاربر	قسمت‌های اجرایی برنامه کاربر	PII PIQ Bit memory Timers Counters L-Stack

Retentive Memory	شامل
	Bit memory Timers Counters

شکل ۱۲-۱۱ نواحی مختلف حافظه در CPUهای S7

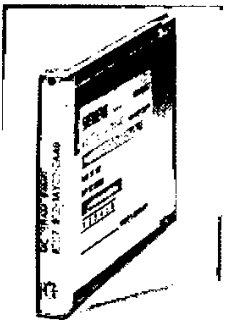
Load Memory ۱-۳-۱۲

زمانی که برنامه‌ی کاربر شامل پیکربندی سخت‌افزار، تنظیمات شبکه و دستورات برنامه‌نویسی و سایر جزئیات به PLC منتقل (Download) می‌گردد، در قسمت Load Memory قرار می‌گیرد.

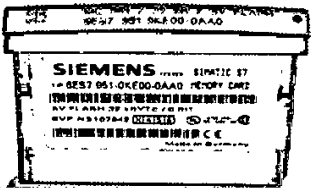
به‌عبارت دیگر Load Memory دروازه ارتباط PLC با PC است که از آن برای دانلود و آپلود و حتی ارتباط Online برای مقاصد عیب یابی و ... استفاده می‌گردد.

بخش Load Memory در CPUهای 300 و 400 طراحی متفاوتی دارند:

در CPUهای 400 و در CPUهای 300 قدیمی: حافظه Load Memory داخلی وجود دارد که از جنس RAM است. این حافظه را می‌توان با کارت حافظه که به MC موسوم است افزایش داد، ولی بدون کارت حافظه نیز می‌توان برنامه را به CPU دانلود کرد و با آن ارتباط برقرار نمود. کارت حافظه برای افزایش حافظه است و در نوع فلش نقش نگهداری اطلاعات در صورت قطع تغذیه را نیز دارد.



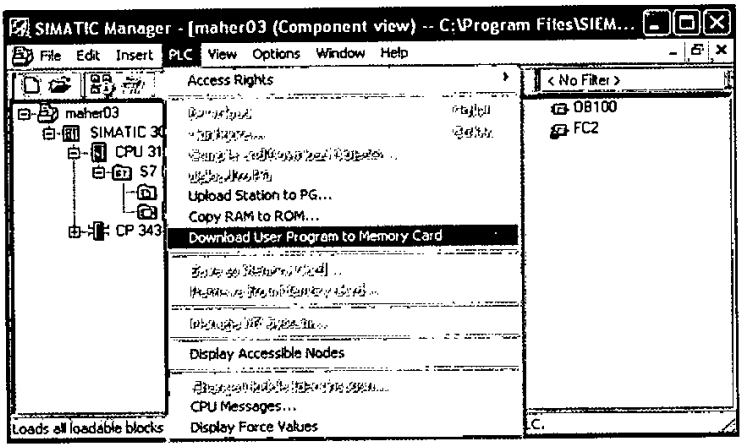
کارت حافظه فلش برای CPU 400



کارت حافظه فلش برای CPU 3 00 قدیمی

شکل ۱۲ – ۱۲ کارت‌های Falsh Eprom موسوم به MC

در این CPUها وقتی کارت حافظه از جنس فلش باشد پس از دانلود کردن، برنامه به‌طور اتوماتیک به RAM داخلی منتقل می‌شود. برای انتقال برنامه به کارت فلش بایستی از گزینه Download User Programm to Memory Card استفاده شود.



شکل ۱۲ – ۱۳ نحوه انتقال برنامه به کارت فلش MC

هر گونه تغییر در برنامه بایستی یک بار دیگر از طریق گزینه فوق ذخیره شود.

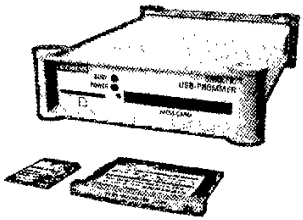
در CPUهای 300 جدید: حافظه Load Memory داخلی وجود ندارد و به‌صورت الزامی بایستی از کارت حافظه از نوع فلش استفاده کرد. این کارت‌ها به MMC معروف هستند و بدون کارت حافظه، CPU در دسترس نیست یعنی امکان دانلود و آپلود و عیب‌یابی وجود ندارد.

در این نوع با هر دانلود اطلاعات به‌طور خودکار به کارت فلش منتقل می‌شود و نیاز به استفاده از گزینه ذکر شده برای MC ندارد.

شکل ۱۲ – ۱۴ کارت حافظه Flash خاص S7-300 با نام MMC

فصل ۱۲

ظرفیت حافظه‌ی کارت‌های MMC از 64 KB شروع شده و تا زمان نگارش این کتاب بیشترین میزان حافظه‌ی کارت MMC، 8MB می‌باشد. توجه شود که این کارت دارای فرمت خاص است و اگر از آن Image تهیه شود سایز فایل Image به‌مراتب بیش از سایز ذکر شده روی کارت است. به همین دلیل این کارت را نباید با RAM Reader معمولی پاک یا فرمت کرد. برای این کار نیاز به PG یا USB Prommer می‌باشد که در شکل ۱۲-۱۵ نشان داده شده است.



شکل ۱۲ – ۱۵ USB Prommer خاص زیمنس

نکته ۱: وقتی کارت حافظه MC یا MMC را به CPU وارد یا از آن خارج می‌کنیم، چراغ Stop شروع به چشمک‌زدن می‌کند. در این‌حالت بایستی سوئیچ روی CPU را یک لحظه به وضعیت MRES برد و رها کرد، MRES در ادامه توضیح داده می‌شود.

نکته ۲: وضعیت حافظه را می‌توان با اتصال به PLC به‌صورت Online مشاهده کرد. جزئیات مربوط به این موضوع در فصل۲۵ آمده است.

نکته ۳: حافظه Load Memory هر CPU را تا حد مشخصی توسط کارت حافظه می‌توان افزایش داد. این میزان در حالت Online در قسمت Maximum Pluggable مشخص است و در مشخصات فنی CPU نیز می‌توان آنرا دید. جدول ۱۲-۲ بخشی از مشخصات فنی CPU 314 را نشان می‌دهد.

جدول ۱۲-۲ وضعیت حافظه در مشخصات فنی CPU314

Memory	
Work memory	
• Integrated	96 KB
• Expandable	No
Load memory	Pluggable by means of Micro Memory Card (max. 8 MB)

نکته ۴ : Load Memory فقط حاوی برنامه نیست، اطلاعات سخت‌افزار و شبکه نیز به‌صورت یک فایل سیستمی[1] به آن منتقل می‌شود. بعلاوه از Load Memory برای ذخیره‌سازی دیتاهای Data Block نیز استفاده می‌شود. Data

1. System Data

Block که اختصاراً DB خوانده می‌شود، بلاکی است که ساختار آن توسط کاربر تعیین می‌گردد. DB در کتاب سطح پیشرفته تشریح شده است.

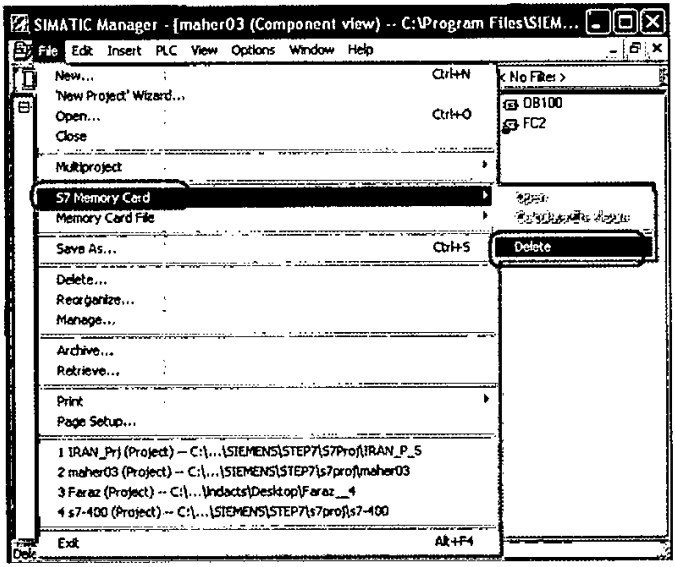
نکته ۵ : تعداد دفعات نوشتن روی کارت MMC یا پاک کردن آن محدود است. در دمای ۶۰ درجه سانتی‌گراد این تعداد 100,000 بار ذکر شده است.

نکته ۶: در برخی CPUهای قدیمی حافظه Load Memory داخلی دارای دو بخش RAM و ROM است، مانند CPU312IFM که اسلاتی برای کارت حافظه ندارد. در این حالت با گزینه Copy RAM to ROM می‌توان اطلاعات RAM را به ROM منتقل کرد تا در صورت قطع تغذیه پاک نشوند.

نکته ۷: کارت حافظه فلش را به دو طریق می‌توان پاک کرد:

۱. از طریق پنجره Online که در فصل ۲۵ آمده است.

۲. با استفاده از PG یا پروگرامر زیمنس که کارت حافظه در اسلات آن قرار می‌گیرد و از مسیر نشان داده شده در شکل ۱۲-۱۵ در Simatic Manager استفاده می‌شود. این گزینه بدون PG یا Programmer فعال نمی‌شود.



شکل ۱۲ – ۱۶ نحوه پاک کردن کارت حافظه فلش از طریق PG یا اتصال به Programmer

Work Memory ۲-۳-۱۲

حافظه‌ی کاری CPU بوده که برنامه اجرایی به آن منتقل می‌شود. از بین آنچه از برنامه‌ی کاربر که به Load Memory منتقل شده است، تنها بخش‌های اجرایی به Work Memory منتقل می‌گردند.

می‌توان حافظه Work Memory را به RAM کامپیوتر تشبیه کرد که برنامه اجرایی از Hard به آن Load می‌شود.

فصل ۱۲

نکته ۱: در برخی از CPUها از جمله S7-400 ناحیه Work Memory به دو بخش تقسیم می‌شود:

- Work Memory Code : که در آن برنامه اجرایی قرار می‌گیرد.
- Work Memory Data : که در آن Data Blockهایی که توسط فانکشن‌های برنامه‌نویسی تولید می‌شوند قرار می‌گیرد. توضیحات مرتبط با این موضوع در کتاب سطح تکمیلی آمده است.

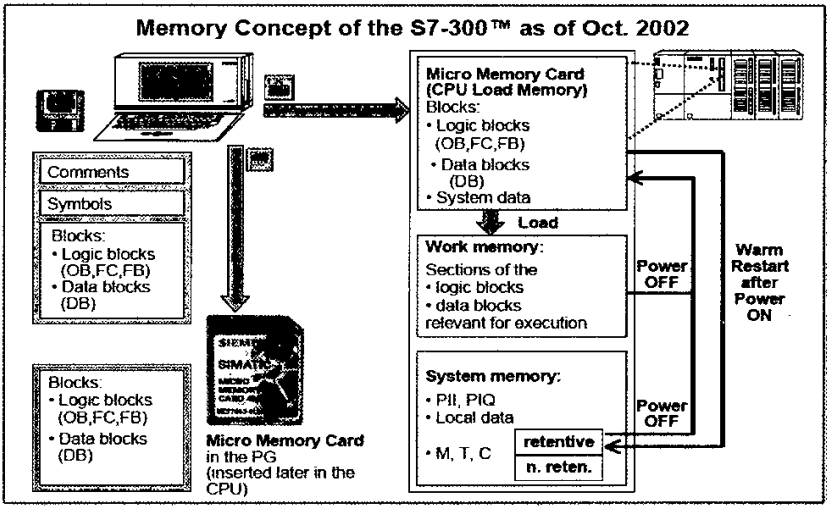
نکته ۲: به جز در CPU 417 این حافظه در سایر CPUهای S7-300 و 400 قابل توسعه نمی‌باشد.

۳-۳-۱۲ System Memory

در این بخش از حافظه، برنامه ذخیره نمی‌شود بلکه مخصوص ذخیره‌سازی دیتاهای مختلف است. این بخش قسمت‌های مختلفی دارد که در شکل ۱۲-۱۷ نشان داده شده است. حافظه ورودی‌ها (PII) و حافظه خروجی‌ها(PIQ)، حافظه تایمرها، حافظه کانترها از مهم‌ترین این قسمت‌ها هستند. این قسمت از حافظه‌ی CPU غیر قابل توسعه می‌باشد.

Retentive Memory

Retentive Memory که از آن به حافظه ماندگار نام می‌بریم، بخشی از حافظه است که در CPUهای S7-300 اگر دیتا در آن قرار گیرد در صورت قطع تغذیه حتی بدون وجود باتری پشتیبان نیز دیتا پاک نمی‌شود. در CPUهای S7-400 پاک شدن یا ماندگار بودن این حافظه بستگی به نوع راه‌اندازی دارد که در ادامه توضیح داده می‌شود.
در شکل ۱۲-۱۷ نواحی مختلف حافظه S7-300 و نحوه تبادل دیتا بین آنها نشان داده شده است.



شکل ۱۲-۱۷ نواحی مختلف حافظه S7-300 و نحوه تبادل دیتا بین آنها

۱۲-۴ راه‌اندازی مجدد CPU

وقتی CPU در مد Run و در حال اجرای سیکل اسکن است، اگر به هر دلیلی Stop شود و دوباره Run گردد به این حالت راه‌اندازی مجدد یا Restart گفته می‌شود. به‌عنوان مثال قطع و وصل تغذیه برق CPU می‌تواند این حالت را ایجاد نماید. سوالی که پیش می‌آید این است که:

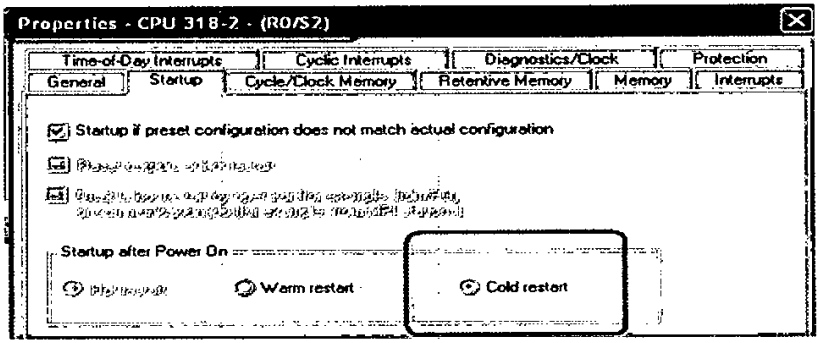
- آیا برنامه از جایی که قطع شده ادامه می‌یابد یا از ابتدا شروع می‌شود؟
- آیا مقادیری که تا آن نقطه از برنامه به‌عنوان نتیجه پردازش به‌دست آمده بود و در System Memory ذخیره شده بود، پس از راه‌اندازی مجدد باقی می‌مانند یا پاک می‌شوند؟ به‌عنوان مثال اگر یک شمارنده تا عددی شمارش کرده باشد و تغذیه قطع و وصل شود چه مقداری خواهد داشت؟

برای پاسخ به این سوالات لازم است بدانیم که برای CPUهای S7 سه نوع راه‌اندازی مجدد وجود دارد که در هر نوع شرایط متفاوتی ایجاد می‌شود. این سه نوع راه‌اندازی عبارتند از Cold، Warm و Hot.

۱-۴-۱۲ Cold Restart

در این نوع راه‌اندازی، اجرای برنامه کاربر از ابتدا آغاز می‌شود. کلیه‌ی محتویات System Memory، چه به صورت ماندگار[1] تعریف شده باشند یا به صورت غیر ماندگار، اطلاعات خود را از دست می‌دهند (مقادیر آنها پاک می‌شوند). این نوع راه‌اندازی در CPUهای S7-300 فقط برای CPU318-2 وجود دارد که در شکل ۱۲-۱۸ نشان داده شده است ولی همه CPUهای 400 آنرا پشتیبانی می‌کنند. در این نوع راه‌اندازی OB101 توسط CPU فراخوانی می‌شود.

اگر شرایط فرآیند ایجاب کند که راه‌اندازی مجدد از نوع Cold باشد بایستی در پارامترهای CPU آنرا تنظیم نمود. با دوبار کلیک نمودن روی CPU در HW Config گزینه‌های راه‌اندازی در بخش Startup نمایش داده می‌شود. شکل ۱۲-۱۷ این بخش را برای CPU318-2 نشان می‌دهد.

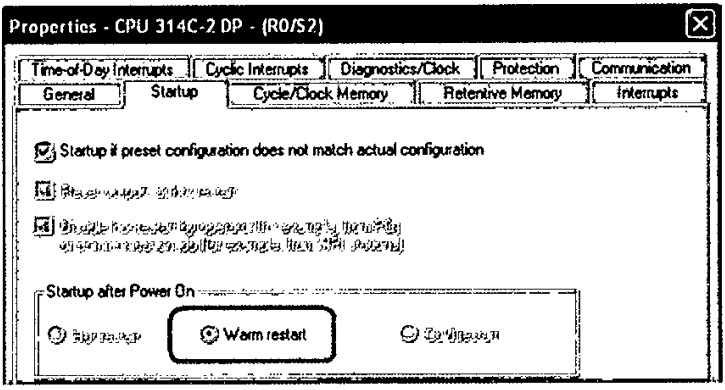


شکل ۱۲ – ۱۸ تنظیم راه اندازی Cold

1. Retentive

Warm Restart ۲-۴-۱۲

در این نوع راه‌اندازی شبیه نوع Cold، اجرای برنامه کاربر از ابتدا آغاز می‌گردد ولی تفاوت آن با Cold در این است که متغیرهایی که به‌صورت ماندگار Retentive تعریف شده باشند، اطلاعات خود را حفظ می‌کنند. به‌طور پیش‌فرض این نوع راه‌اندازی برای CPUها منظور شده است که می‌توان آنرا در پارامترهای CPU تغییر داد. در این نوع راه‌اندازی OB100 توسط CPU فراخوانی می‌شود.

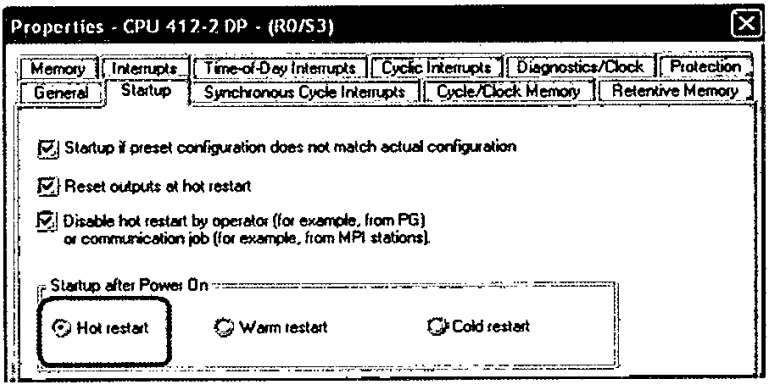


شکل ۱۲ – ۱۹ تنظیم راه‌اندازی Warm

Hot Restart ۳-۴-۱۲

در این نوع راه‌اندازی، برنامه‌ی کاربر از جایی که قطع شده بود ادامه پیدا می‌نماید. کلیه‌ی متغیرهای حافظه System، چه به صورت ماندگار یا غیرماندگار تعریف شده باشند، اطلاعات خود را حفظ می‌کنند.

به‌منظور انجام این نوع راه‌اندازی، CPU باید دارای باطری پشتیبان باشد. این نوع راه‌اندازی مخصوص CPUهای S7-400 است. در این نوع راه‌اندازی OB102 توسط CPU فراخوانی می‌شود.



شکل ۱۲-۲۰ تنظیم راه اندازی Hot

۱۲-۵ ری‌ست کردن CPU

۱۲-۵-۱ عملکرد Reset

Reset کردن CPU یعنی پاک کردن حافظه CPU، یعنی هم مقادیر متغیرهای حافظه و هم برنامه‌ای که توسط کاربر به حافظه ارسال شده پاک می‌گردد.

وقتی CPU ری‌ست می‌گردد، برنامه‌ای که درحافظه داخلی Load Memory و Work Memory وجود دارد پاک می‌شود و مقادیر متغیرهای System Memory نیز پاک می‌گردند. اگر کارت حافظه از نوع RAM وجود داشته باشد، ری‌ست محتوایات آن را نیز پاک می‌کند ولی اگر حافظه ROM یا Flash EPROM باشد با ری‌ست پاک نمی‌شود. باید توجه داشت که Reset و Cold Restart با یکدیگر متفاوتند، در Cold Restart حافظه‌های Work و Load پاک نمی‌شوند و فقط System Memory پاک می‌گردد.

عمل Reset کردن می‌تواند از طریق نرم‌افزار یا از طریق سوئیچ روی CPU که به MRES مخفف Memory Reset معروف است انجام شود. در برخی شرایط ممکن است نتوان از طریق نرم‌افزار به CPU متصل شد تا آنرا Reset نمود، در اینصورت الزاماً بایستی از سوئیچ روی CPU استفاده کرد.

۱۲-۵-۲ Reset با سوئیچ MRES

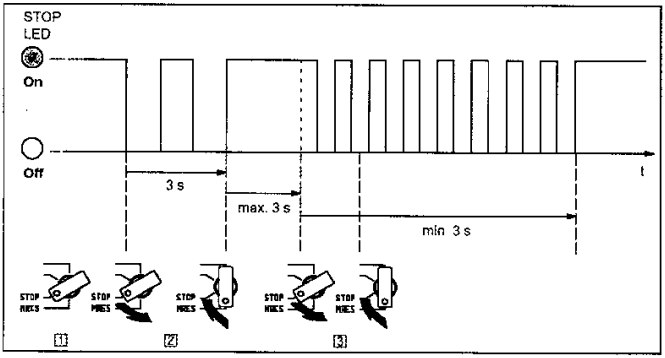
اگر سوئیچ روی CPU را به وضعیت MRES ببریم و رها کنیم سوئیچ به وضعیت Stop بر می‌گردد، بدین‌صورت عمل ری‌ست انجام نمی‌شود. برای‌ری‌ست‌کردن CPUبایدسوئیچ طبق سیکل زیر بین وضعیت MRES و STOP جابه‌جا شود:

۱- سوئیچ در وضعیت Stop است و LED مربوط به Stop روشن است.

۲- سوئیچ را از STOP به MRES می‌بریم و ۳ ثانیه نگه می‌داریم، در طول ۳ ثانیه LED مربوط به Stop چشمک زن است. با گذشت ۳ ثانیه LED مربوط به Stop ثابت می‌شود.

۳- وقتی LED مربوط به STOP ثابت شد، سوئیچ را از MRES به Stop برمی‌گردانیم و با مکث کوتاهی (حداکثر۳ ثانیه) سوئیچ را از Stop دوباره به MRES می‌بریم که در این‌حالت LED فوق به حالت چشمک‌زن سریع در می‌آید.

حالت چشمک‌زن فوق نشان‌دهنده این است که حافظه CPU ری‌ست شده است. اگر در این مرحله LED به‌صورت چشمک‌زن در نیامد باید مراحل فوق از اول تکرار شود.



شکل ۱۲-۲۱ مراحل ری‌ست کردن CPU از طریق سوئیچ

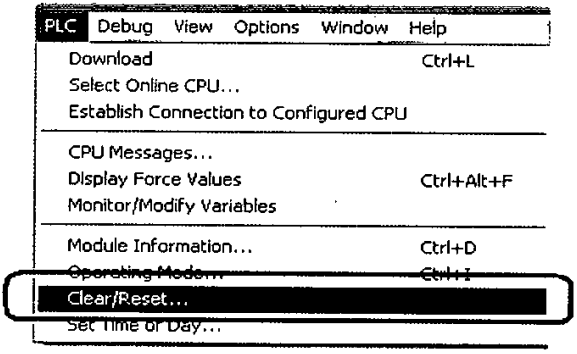
نکته: از این روش نمی‌توان برای CPUهای جدید S7-300 و S7-400 استفاده نمود.

فصل ۱۲

۱۲-۵-۳ Reset از طریق نرم‌افزار

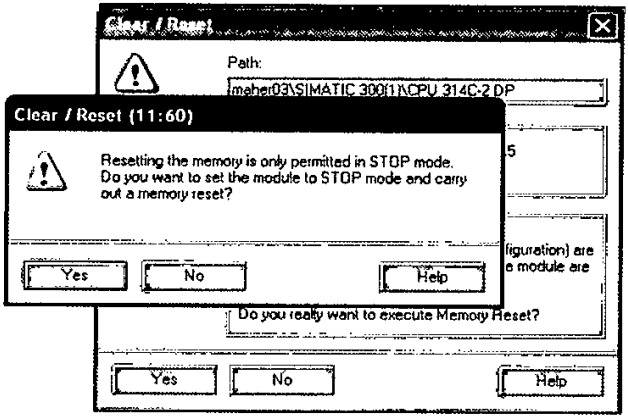
ری‌ست کردن می‌تواند از طریق نرم‌افزار Step7 یا زیر برنامه‌های آن انجام شود. در هر محیطی که باشیم این کار به‌سادگی با استفاده از منوی PLC > Clear / Reset که در برنامه‌های مختلف نرم‌افزار Step7 از جمله در HW Config وجود دارد، می‌توان عمل ری‌ست را بدون نیاز به سیکل عملیات سخت افزاری فوق انجام داد.

در Simatic Manager یا HW Config یا Netpro برای ریست کردن ابتدا CPU را انتخاب می‌کنیم تا گزینه Clear Reset در منوی PLC فعال گردد. شکل ۱۲-۲۲ این منو را نشان می‌دهد. شکل ۱۲-۳ در ابتدای این فصل به‌صورت کامل‌تر گزینه ری‌ست را در محیط‌های مختلف نشان می‌دهد.



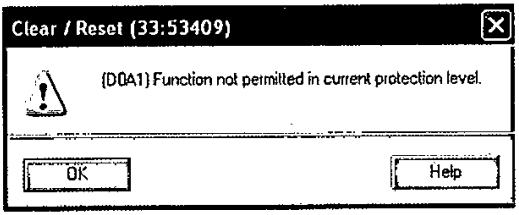
شکل ۱۲-۲۲ انجام عمل Reset به‌صورت نرم‌افزاری

اگر CPU در مد RUN-P باشد و از طریق نرم‌افزار اقدام به Reset کنیم با پیام زیر مواجه می‌شویم. این پیام اخطار می‌دهد که ابتدا CPU به مد Stop بر می‌گردد و سپس ری‌ست می‌شود.



شکل ۱۲-۲۳ ری‌ست کردن CPU وقتی در مد RUN-P است.

اگر CPU از نوع قدیمی باشد و کلید روی آن نیز در وضعیت RUN باشد، با ری‌ست نرم‌افزاری پیام فوق ظاهر شده و CPU به مد Stop می رود. دراین‌حالت با پیام خطای زیر مواجه می‌شویم و عمل ری‌ست انجام نمی‌شود.



شکل ۱۲-۲۴ پیام خطای ری‌ست وقتی کلید در وضعیت RUN باشد (CPUهای قدیمی).

۱۲-۶ تاثیر ری‌ست و راه اندازی روی بخش‌های مختلف حافظه

جدول ۱۲-۲ جمع‌بندی تأثیر Reset و انواع مختلف Restart را روی بخش‌های مختلف حافظه CPU نشان می‌دهد.

جدول ۱۲-۳ تأثیر Reset و Restart روی حافظه CPU

	Load Memory		Work Memory		System Memory	
	RAM	EPROM	Code	Data	Retentive	Non-Retentive
Reset	پاک می‌شود.	پاک نمی‌شود.	پاک می‌شود.	پاک می‌شود.	پاک می‌شود.	پاک می‌شود.
Cold Restart	پاک می‌شود.	پاک نمی‌شود.	پاک می‌شود.	پاک می‌شود.	پاک می‌شود.	پاک می‌شود.
Warm Restart	پاک می‌شود.	پاک نمی‌شود.	پاک می‌شود.	پاک می‌شود.	پاک نمی‌شود.	پاک می‌شود.
Hot Restart (با باتری)	پاک نمی‌شود.	پاک نمی‌شود.	پاک نمی‌شود.	پاک نمی‌شود.	پاک نمی‌شود.	پاک نمی‌شود.

۱۲-۷ تاثیر عملکرد CPU روی LEDهای آن

همه‌ی CPU های S7-300 اعم از قدیمی و جدید، دارای لامپ‌های نشانگر می‌باشند. لامپ‌های نشانگر به منظور نشان‌دادن وضعیت کاری CPU و خطاهای احتمالی پیش‌آمده استفاده می‌شوند.

CPU300

CPU400

شکل ۱۲-۲۵ LEDهای روی CPU

جدول ۱۲-۳ لامپ‌ها و نشانگرهای موجود روی CPU های S7-300 و مفاهیم آنها را شرح می‌دهد.

جدول۱۲-۳ لامپ‌ها و نشانگرهای موجود روی CPU های S7-300

عنوان LED	نوع CPU	وضعیت در شرایط عادی	وضعیت در زمان فعال شدن یا عملکرد	توضیح
STOP	400/ 300	خاموش	روشن- رنگ زرد	CPU در حالت STOP قرار دارد.
RUN	400/ 300	روشن	روشن- سبز	CPU در حالت RUN قرار دارد.
FRCE	400/ 300	خاموش	روشن- رنگ زرد	حالت FORCE فعال شده است. در فصل‌های بعد تشریح می‌شود.
DC 5V	300	روشن	روشن- سبز	تغذیه 5V DC مربوط به CPU و BUS برقرار است.
BATF	300قدیمی	خاموش	روشن- رنگ قرمز	خطا در باتری پشتیبان
SF	300	خاموش	روشن- رنگ قرمز	وجود خطای سیستمی
INTF	400	خاموش	روشن- رنگ قرمز	وجود خطای داخلی
EXTF	400	خاموش	روشن- رنگ قرمز	وجود خطای خارجی
BF-BUSF	300/400	خاموش	روشن یا چشمک‌زن – رنگ قرمز	اشکال در BUS شبکه

۱۲-۸ پرسش و تحقیق

در PLC سایر سازندگان ساختار حافظه چگونه است؟ آیا شبیه S7-300 استفاده از کارت حافظه فلش در آنها نیز الزامی است.

۱۲-۹ تست‌های خودآزمایی

۱- منظور از مد Startup در CPU چیست؟

الف) CPU در حالت توقف قرار دارد. ب) CPU در حالت RUN قرار دارد.

ج) CPU از حالت Stop به Run می‌رود. د) CPU از حالت Run به Stop می‌رود.

۲- از بین مدهای کاری زیر، کدامیک دارای بالاترین اولویت می‌باشد؟

الف) STOP ب) HOLD ج) RUN د) Startup

۳- مرحله اول در سیکل اسکن CPU کدام گزینه است؟

الف) خواندن ورودی‌ها ب) اجرای دستورات راه‌اندازی

ج) ارسال خروجی‌ها د) اجرای برنامه

۴- دیتاهای مربوط به ورودی در کدام ناحیه قرار می‌گیرد؟

الف) PII ب) PIQ ج) Bit Memory د) L-Stack

۵- در وضعیت . . . ، پردازش برنامه کاربر متوقف شده و می‌توان برنامه را به‌صورت مرحله به مرحله تست نمود.

الف) STOP ب) HOLD ج) RUN د) Startup

۶- به معنای حافظه بارگذاری بوده و برنامه پس از دانلود به CPU در آن قرار می‌گیرد.

الف) Load Memory ب) Work Memory

ج) System Memory د) Retentive Memory

۷- در این قسمت از حافظه CPU، عمل پردازش برنامه انجام می‌شود.

الف) Load Memory ب) Work Memory

ج) System Memory د) Retentive Memory

۸- نواحی PII، PIQ، Bit Memory در کدام ناحیه از حافظه CPU قرار دارند؟

الف) Load Memory ب) Work Memory

ج) System Memory د) Retentive Memory

۹- به‌معنای حافظه ماندگار بوده و دیتاهای تعریف شده در آن در صورت راه‌اندازی مجدد Warm پاک نمی‌شوند.

الف) Load Memory ب) Work Memory

ج) System Memory د) Retentive Memory

۱۰- محل قرارگیری System Data و Data Block در کدام ناحیه از حافظه CPU می‌باشد؟

الف) Load Memory ب) Work Memory

ج) System Memory د) Retentive Memory

۱۱- در این نوع راه‌اندازی، System Memory چه به‌صورت ماندگار تعریف شده باشد یا غیرماندگار، محتویات آن پاک می‌شود.

الف) Hot ب) Cold ج) Warm د) هیچ‌کدام

۱۲- در این نوع راه‌اندازی مجدد، پردازش برنامه از اول شروع شده و دیتاهایی که به‌صورت ماندگار تعریف شده‌اند، مقدار خود را حفظ می‌کنند.

الف) Hot ب) Cold ج) Warm د) هیچ‌کدام

۱۳- در این نوع راه‌اندازی وجود باتری پشتیبان الزامی است.

الف) Hot ب) Cold ج) Warm د) هیچ‌کدام

۱۴- کدام ناحیه حافظه را در اکثر CPUها می‌توان افزایش داد؟

الف) Load Memory ب) Work Memory

ج) System Memory د) Retentive Memory

۱۵- این ناحیه از حافظه CPU، فقط در CPU 417 قابل افزایش است.

الف) Load Memory ب) Work Memory

ج) System Memory د) Retentive Memory

فصل ۱۲

فصل ۱۳

مفاهیم پایه برنامه‌نویسی PLC

۱۳-۱ مقدمه

۱۳-۲ سیستم‌های عددی مورد استفاده در PLC

۱۳-۲-۱ سیستم عددی دسیمال (Decimal System)

۱۳-۲-۲ سیستم عددی باینری (Binary System)

۱۳-۲-۳ سیستم عددی BCD

۱۳-۲-۴ سیستم عددی Hex (Hexadecimal)

۱۳-۳ انواع Data Type (نوع داده)

۱۳-۳-۱ Elementary Data Types (داده‌های پایه)

۱۳-۳-۲ Complex Data Types

۱۳-۳-۳ Parameter Types

۱۳-۴ نحوه آدرس‌دهی متغیرهای حافظه

۱۳-۴-۱ نکات کلی آدرس‌دهی

۱۳-۴-۲ آدرس‌دهی کارت‌های دیجیتال

۱۳-۴-۳ آدرس‌دهی کارت‌های آنالوگ

۱۳-۴-۴ آدرس‌دهی Peripheral

۱۳-۴-۵ آدرس‌دهی متغیرهای حافظه

۱۳-۴-۶ آدرس‌دهی کانتر و تایمر

۱۳-۴-۷ آدرس دهی مطلق و سمبلیک

۱۳-۵ نکته پایانی

۱۳-۶ پرسش و تحقیق

۱۳-۷ تست‌های خودآزمایی

در این فصل مفاهیم پایه برنامه‌نویسی PLC که آشنایی با آنها برای شروع برنامه‌نویسی لازم است، تشریح شده است.

فصل ۱۳

چکیده مطالب

- در PLC می‌توان سیستم‌های عددی مختلفی مانند Binary, Hex, BCD, Decimal استفاده کرد.
- اعداد در سیستم‌های مختلف عددی به یکدیگر قابل تبدیل هستند.
- منظور از اعداد دسیمال اعداد مبنای ۱۰ می‌باشد.
- مبنای هگز برای سهولت در خواندن و نوشتن اعداد باینری به‌کار می‌رود.
- در BCD ارقام اگر چه شبیه دسیمال هستند ولی ارزش دهگان و صدگان و غیره ندارند.
- در PLC دیتاهای را می‌توان به انواع مختلف دسته‌بندی کرد.
- دسته اول دیتاها، دسته Elementary نام دارند که دیتاهای اصلی قابل استفاده در برنامه را شامل می‌شود.
- دسته دوم دیتاهای Complex هستند که در برخی از آنها همراه با دیتاهای Elementary به‌کار می‌روند.
- دسته سوم دیتاهای Parameter هستند که برای کاربردهای پیشرفته‌تر استفاده می‌شوند.
- متغیرهای حافظه را به شکل‌های مختلف Bool, Byte, Word,Dword می‌توان آدرس‌دهی کرد.
- ورودی و خروجی‌ها توسط برنامه HW Config به‌طور خودکار آدرس‌دهی می‌شوند.
- آدرس‌های داده شده توسط برنامه را در بسیاری از CPUها می‌توان تغییر داد.
- آدرس‌های آنالوگ ممکن است با حرف P شروع شوند که معرف Peripheral است.
- آدرس‌های Peripheral خارج از ناحیه PII و PIQ قرار می‌گیرند.
- امکان آدرس‌دهی بیتی برای Peripheral وجود ندارد.

اصطلاحات و تعاریف

Bit (بیت)

کوچکترین واحد حافظه است که می‌تواند صفر منطقی یا یک منطقی باشد.

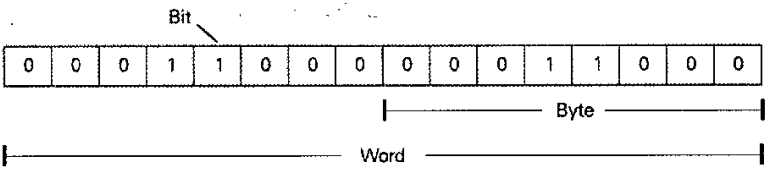
Byte (بایت)

واحد بزرگتر از بیت است. به هر هشت بیت یک بایت گفته می‌شود. واحدهای دیگری نیز بر اساس بایت بیان می‌شوند مانند کیلو بایت و مگابایت که عبارتند از:

1 Kilo Byte = 1024 bits
1 Mega Byte = 1024 Bytes

Word

به هر دو بایت یک Word گفته می‌شود. در واقع هر Word از شانزده بیت تشکیل شده است.



Double Word

به هر دو Word، یک Double Word گفته می‌شود، که از سی و دو بیت تشکیل شده است. پس:

1 DWord = 2 Words = 4 Bytes = 32 bits

۱-۱۳ مقدمه

در این فصل به بررسی سیستم‌های عددی مورد استفاده در PLC پرداخته و پس از آشنایی با انواع داده‌های موجود، به بررسی نحوه‌ی آدرس‌دهی متغیرهای حافظه می‌پردازیم.

تمام نکات بحث شده‌دراین‌فصل، پایه برنامه‌نویسی است و بدون شناخت آنها نمی‌توان وارد مبحث برنامه‌نویسی گردید.

۲-۱۳ سیستم‌های عددی مورد استفاده در PLC

PLC عضوی از خانواده کامپیوتر است و همانند آن، اطلاعات را به فرم دیتای صفر و یک منطقی می‌بیند. گاهی اوقات بیت‌ها به صورت تکی مورد استفاده قرار می‌گیرند، گاهی اوقات از آنها برای بیان یا ذخیره‌سازی مقادیر بزرگتر از یک بیت مانند مقادیر عددی مورد استفاده قرار می‌گیرند.

فهم اینکه چگونه این بیت‌ها می‌توانند برای مقادیر عددی استفاده شوند، به فهم سیستم عددی باینری و سایر سیستم‌های عددی وابسته است، لذا به بررسی سیستم‌های مختلف عددی مورد استفاده در PLC می‌پردازیم. انواع سیستم‌های عددی مورد استفاده در PLC عبارتند از:

- اعداد Decimal
- اعداد Binary
- اعداد BCD
- اعداد Hex

فصل ۱۳

۱-۲-۱۳ سیستم عددی دسیمال[1]

سیستم عددی دسیمال همان نمایش اعداد در مبنای ده می‌باشد که در زندگی روزمره‌ی خود با آن سروکار داریم. همه‌ی سیستم‌های عددی دارای سه مشخصه می‌باشند، این مشخصه‌ها عبارتند از: *ارقام، مبنا، وزن.*

در سیستم عددی دسیمال این مشخصه‌ها به شرح زیر می‌باشند:

- **ارقام:** ده رقم 0,1,2,3,4,5,6,7,8,9
- **مبنا:** ۱۰
- **وزن:** توان‌های مبنای ده (. . . ,1,10,100,1000)
- **مثال:** . . . ,12, 234,1362

۲-۲-۱۳ سیستم عددی باینری[2]

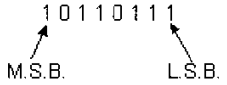
در این سیستم که مورد استفاده کامپیوترها می‌باشد، کلیه‌ی اعداد در مبنای دو و توسط دو عدد صفر و یک نمایش داده می‌شوند. هر بیت با توانی از ۲ با توجه به محل قرارگیری در عدد، در نظر گرفته می‌شود. مشخصات این سیستم عبارت است از:

- **ارقام:** دو رقم 0,1

1. Decimal System
2. Binary System

- **مبنا:** ۲
- **وزن:** توان‌های مبنای دو (. . . ,2,4,8,16)
- **مثال:** . . . ,00,01,10,11,100

در یک عدد باینری، آخرین بیت (رقم) از سمت چپ با ارزش‌ترین بیت در نظر گرفته شده و با MSB مخفف Most Significant Bit نمایش داده می‌شود. اولین بیت از سمت راست به عنوان کم‌ارزش‌ترین بیت (رقم) در نظر گرفته شده و با LSB مخفف Least Significant Bit نماش داده می‌شود.



شکل ۱۳-۱ ترتیب بیت‌های باینری

تبدیل دسیمال به باینری و برعکس

برای تبدیل یک عدد **دسیمال به باینری** کافی است تا عدد مورد نظر را مرتباً بر عدد دو تقسیم نموده و در هر مرحله، باقیمانده را یادداشت نمود. با مرتب نمودن باقیمانده‌ها از راست به چپ، عدد باینری به‌دست می‌آید. برای فهم چگونگی تبدیل عدد در مبنای باینری به یک عدد در مبنای دسیمال به مثال زیر توجه نمایید.

مثال ۱۳-۱ معادل باینری عدد دسیمال ۱۰ را محاسبه نمایید.

حل:

10/2= 5, 5/2=2, 2/2=1, 1/2=0

0	1	0	1

باقیمانده تقسیم

از راست به چپ: (1010) که معادل ۱۰ دسیمال است.

برای تبدیل عدد *باینری به عدد دسیمال* می‌توان مطابق روش زیر عمل نمود.

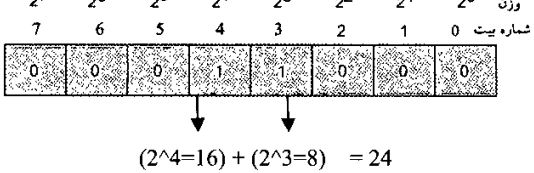
۱- ابتدا ارقام را از چپ به راست شماره‌گذاری کنید.

۲- عدد ۲ را به توان شماره‌ی تک‌تک رقم‌هایی که دارای مقدار یک هستند برسانید.

۳- اعداد به‌دست آمده را با هم جمع کنید.

مثال ۱۳-۲ عدد 00011000 در مبنای دو موجود است، نمایش این عدد در مبنای ده چگونه خواهد بود؟

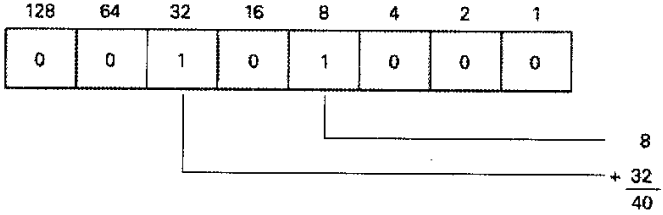
حل: ابتدا ارقام را از چپ به راست شماره‌گذاری می‌نماییم. سپس ارقامی که دارای مقدار یک هستند را مشخص نموده و عدد ۲ را به توان شماره‌ی رقم می‌رسانیم. در پایان اعداد به‌دست آمده را با هم جمع می‌کنیم. عدد حاصل عدد ۲۴ می‌باشد.



شکل ۱۳-۲ نحوه تبدیل عدد باینری به دسیمال مربوط به مثال ۱۳-۲

مثال ۱۳-۳ عدد 00101000 در مبنای دو موجود است. معادل این عدد در مبنای ده را به‌دست آورید.

حل: می‌توان مشابه روش بیان شده جهت مثال قبل، این مثال را نیز حل نمود. عدد حاصل در مبنای ده برابر است با 40.



شکل ۱۳-۳ نحوه تبدیل عدد باینری به دسیمال مربوط به مثال ۱۳-۳

فصل ۱۳

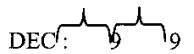
۱۳-۲-۳ سیستم عددی BCD

در این سیستم، هر چهار رقم باینری به‌صورت یک عدد دسیمال کددهی شده و نمایش داده می‌شود. در واقع هر چهار رقم باینری در کد BCD به یک عدد دسیمال تبدیل می‌شود. از قرارگیری اعداد تبدیل شده، یک عدد چند رقمی حاصل می‌شود. جدول ۱۳-۱ هر رقم BCD و معادل دسیمال آنرا نشان می‌دهد.

جدول ۱۳-۱ اعداد در کد BCD و معادل دسیمال آنها

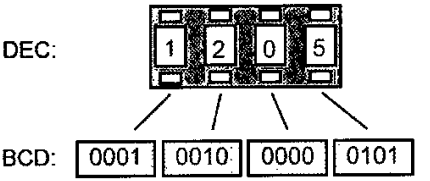
اعداد در سیستم	
دسیمال	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

مثال ۱۳-۴ در جدول ۱۳-۱ نشان داده شده است که معادل عدد 9 در کد دسیمال، عدد 1001 در کد BCD می‌باشد. برای ساخت معادل BCD عدد 99 دسیمال، کافیست که هر کدام رقم‌های عدد دسیمال (9) را به معادل BCD آن تبدیل کنیم، در نتیجه داریم: BCD : 1001 1001



تفاوت BCD با Decimal در این است که اعداد وزن دهگان یا صدگان و .. ندارند. به‌عنوان مثال وقتی عدد 1205 را روی صفحه نمایش یک شمارنده به‌صورت BCD می‌بینیم برخلاف مبنای ۱۰ ، عدد ۵ ارزش یکان دارد، ولی عدد صفر ارزش دهگان و عدد ۲ ارزش صدگان و عدد ۱ ارزش هزارگان نخواهد داشت. در این روش عدد باینری به قسمت‌های ۴

تایی شکسته می‌شود و معادل دهدهی هر ۴ بیت بدون توجه به وزن آن به‌صورت دسیمال به‌دست می‌آید. وقتی اعداد به‌دست آمده را در کنار هم قرار دهیم معادل BCD را خواهیم داشت.به شکل ۱۳-۴ توجه کنید.



شکل ۱۳-۴ مقایسه عدد دسیمال و BCD با یک مثال

همانطور که مشاهده می‌شود، تک تک ارقام دسیمال را به یک عدد باینری تبدیل نموده و از کنار هم قرار دادن آنها، عدد BCD مورد نظر حاصل می‌شود.

۱۳-۲-۴ سیستم عددی Hex (Hexadecimal)

مبنای ۱۶ ارتباط مستقیم با مبنای ۲ دارد. از آنجا که بیت‌های باینری در کنار هم به‌صورت طولانی ردیف می‌شوند فهم آنها مشکل است. با تبدیل باینری به Hex این بیتها در فضای کمتری جا می‌گیرند و کاربر راحت‌تر می‌تواند مقدار باینری را تشخیص دهد. اگر یک عدد باینری را که شامل تعداد زیادی صفر و یک است از سمت راست در دسته‌های ۴ تایی جدا کنیم و برای هر ۴ بیت یک عدد معادل قرار دهیم، معرف عدد هگز خواهد بود. از آنجا که یک عدد ۴ بیتی اگر همه بیت‌هایش یک باشند معادل ۱۵ خواهد بود، پس با یک رقم دسیمال قابل نمایش نیست و برای بیان آن از حروف استفاده می‌شود.

جدول ۱۳-۲ ارقام عدد Hex و معادل باینری آنرا نشان می‌دهد. جهت تبدیل یک عدد باینری به هگز، کافیست هر چهار رقم باینری را جداگانه به یک رقم Hex تبدیل نمود. از قرار دادن ارقام مختلف، یک عدد Hex به‌وجود می‌آید.

جدول ۱۳-۲ اعداد Hex و معادل باینری آن

عدد Hex	عدد باینری
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

همانطور که در جدول ۱۳-۲ دیده می‌شود، در سیستم عددی Hex، ده رقم اصلی موجود در دسیمال بعلاوه‌ی حروف انگلیسی A تا F ارقام مورد نظر را شکل می‌دهد.

- **ارقام:** شانزده رقم 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

A=10 , B= 11 , C=12 , D=13 , E=14 , F= 15

- **مبنا:** 16
- **وزن:** توان‌های مبنای شانزده (. . . ,1,16,256,4096)

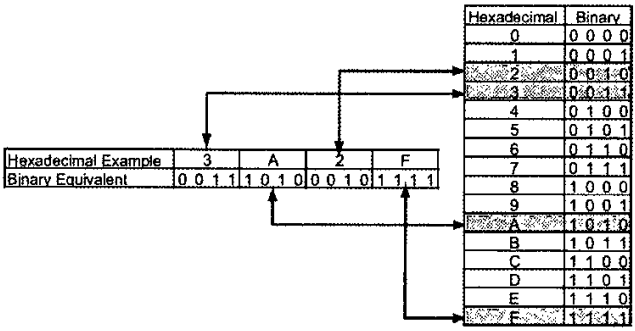
مثال ۱۳-۵ معادل Hex عدد باینری 00111111 را به‌دست آورید.

حل: ابتدا هر چهار رقم باینری را جدا نموده و سپس معادل Hex آن چهار رقم را محاسبه می‌نماییم. چهار رقم سمت راست عدد 1111 است که معادل Hex آن عدد F می‌باشد. چهار رقم سمت چپ عدد 0011 می‌باشد که معادل Hex آن عدد 3 ذکر شده است لذا معادل Hex عدد باینری 00111111 برابراست با 3F. به‌همین ترتیب، معادل هر عدد دیگری را نیز می‌توان محاسبه نمود.

فصل ۱۳

مثال ۱۳-۶ معادل Hex عدد باینری 0 0 1 1 1 0 1 0 0 0 1 0 1 1 1 1 را محاسبه کنید.

حل: همانطور که در شکل ۱۳-۵ مشخص است، می‌توان مشابه مثال قبل عمل نمود که عدد Hex حاصل عبارت است از 3A2F.



شکل ۱۳-۵ معادل هگز یک عدد باینری

تبدیل Hex به دسیمال

تبدیل یک عدد از فرم Hex به فرم دسیمال شبیه تبدیل عدد در فرم باینری به دسیمال است، با این تفاوت که عدد موجود در هر رقم Hex در عدد ۱۶ که به توان شماره‌ی رقم رسیده است، ضرب شده و نتایج با هم جمع می‌شوند. در شکل ۱۳-۶ چگونگی این عمل نشان داده شده است.

شکل ۱۳-۶ تبدیل عدد Hex به دسیمال

در جدول ۱۳-۳ ارتباط بین اعداد باینری و دسیمال و هگز و BCD نشان داده شده است.

جدول ۱۳-۳ رابطه بین اعداد باینری و دسیمال و هگز

Decimal	Binary	BCD	Hexadecimal
0	0	0000	0
1	1	0001	1
2	1	0010	2
3	11	0011	3
4	100	0100	4
5	101	0101	5
6	110	0110	6
7	111	0111	7
8	1000	1000	8
9	1001	1001	9
10	1010	0001 0000	A
11	1011	0001 0001	B
12	1100	0001 0010	C
13	1101	0001 0011	D
14	1110	0001 0100	E
15	1111	0001 0101	F
16	1 0000	0001 0110	10
17	1 0001	0001 0111	11
18	1 0010	0001 1000	12
19	1 0011	0001 1001	13
20	1 0100	0010 0000	14
.....	.....	.....	.....
126	111 1110	0001 0010 0110	7E
127	111 1111	0001 0010 0111	7F
128	1000 0000	0001 0010 1000	80
----	----	.—	——
510	1 1111 1110	0101 0001 000	1FE
511	1 1111 1111	0101 0001 0001	1FF
512	10 0000 0000	0101 0001 0010	200

۱۳-۳ انواع Data Type (نوع داده)

همانطور که بیان شد، داده‌ها به‌صورت باینری در اختیار PLC قرار می‌گیرند. جهت انجام برنامه‌های متفاوت، علاوه‌بر نوع داده باینری نیاز به انواع داده‌ی دیگر نیز می‌باشد. به‌عنوان مثال بیان یک ورودی آنالوگ که ۱۶ بیتی است به‌صورت باینری اگر چه امکان‌پذیر است ولی برای انجام کارهای محاسباتی روی آن لازم است که به فرمت داده‌های دیگر مانند عدد صحیح بیان گردد، از اینرو تعریف انواع داده‌ها در PLC ضرورت پیدا می‌کند. به‌طور کلی انواع داده‌های قابل تعریف در STEP7 را می‌توان به سه گروه تقسیم‌بندی نمود:

۱. Elementary Data Types یا داده‌های پایه
۲. Complex Data Types یا داده‌های مختلط
۳. Parameter Types یا داده‌های پارامتری

در هر کدام از گروه‌های فوق، چندین نوع داده وجود دارد که در ادامه به بررسی آنها خواهیم پرداخت.
در معرفی نوع داده‌ها نکات مهم عبارتند از:

فصل ۱۳

اندازه بر حسب بیت

نشان‌دهنده‌ی میزان فضای اشغالی توسط داده می‌باشد. برخی از داده‌های تک بیتی بوده و برخی 8، 16، 32 یا 64 بیتی می‌باشند.

فرمت (قالب)

بیانگر قالب داده و چگونگی عملکرد آن می‌باشد.

رنج عملکرد

بیانگر محدوده‌ی عملکرد یک نوع داده می‌باشد.

۱۳-۳-۱ Elementary Data Types (داده‌های پایه)

این گروه، شامل انواع داده‌های پایه و بسیار پرکاربرد در STEP7 می‌باشد. داده‌های موجود در این گروه در اکثر برنامه‌های نوشته شده مورد استفاده قرار می‌گیرند.

جدول ۱۳-۴ انواع داده‌های پایه را نشان می‌دهد. در این جدول علاوه بر اندازه و فرمت هر نوع داده، مثالی نیز ارائه شده است.

جدول ۱۳-۴ انواع داده‌های پایه

نوع داده	اندازه بر حسب بیت	مثال
BOOL	1	1 or 0
BYTE	8	B#16#A9
WORD	16	W#16#12AF
DWORD	32	DW#16#ADAC1EF5
CHAR	8	'w'
S5TIME	16	S5T#5s_200ms
INT	16	123
DINT	32	65539
REAL	32	1.2 or 34.5E-12
TIME	32	T#2D_1H_3M_45S_12MS
DATE	16	D#1993-01-20
TIME_OF_DAY	32	TOD#12:23:45.12

جزئیات داده‌های پایه در جدول ۱۳-۵ آمده است.

جدول ۱۳-۵ رابطه بین اعداد باینری و دسیمال و هگز

نوع داده	فرمت	سایز (bit)
Bool	-	1
Byte	عدد B#16# که در آن عدد بین 00 تا FF است.	8
Word	عدد W#16# که در آن عدد بین 0000 تا FFFF است.	16
DWord	عدد DW#16# که در آن عدد بین 0000_0000 تا FFFF_FFFF است.	32
Integer (INT)	عدد صحیح با علامت از ۳۲۷۶۸- تا ۳۲۷۶۸+	16
Double Integer (DINT)	عدد صحیح با علامت L# که در آن عدد بین 2147483648- تا 2147483647 است.	32
Real	عدد اعشاری (Floating Point)	32
S5Time	زمان S5T# در پله‌های ۱۰ میلی‌ثانیه‌ای از صفر تا	16

	۲ ساعت و ۴۶ دقیقه و ۳۰ ثانیه که به‌فرم 2H46M30S0MS نمایش داده می‌شود.	
Time	زمان#T در پله‌های ۱ میلی ثانیه‌ای از صفر تا ۲۴ روز و۲۰ ساعت و ۳۱ دقیقه و ۲۳ ثانیه و۶۴۷ میلی‌ثانیه که به‌فرم 24D20H31M23S647MS نمایش داده می‌شود.	32
Date	تاریخ به شکل yyy-mm-dd نوشته می‌شود.	16
Time_Of_Day	TOD#h:m:s.ms که در آن زمان بین 0:0:0.0 تا23:59:59.999 در پله‌های ۱ میلی‌ثانیه	32
CHAR	' یک کاراکتر ' کاراکتر می‌تواند حرف یا عدد باشد مانند 'a'.	8

فصل ۱۳

اکنون به بررسی کامل انواع داده‌های پایه در STEP7 می‌پردازیم.

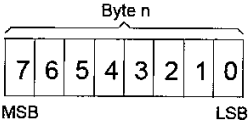
نوع داده‌ی BOOL

این نوع داده که اندازه‌ی آن یک بیت است، به‌منظور نمایش یا ذخیره‌سازی اطلاعات سیگنال‌های دیجیتال مورد استفاده قرار می‌گیرد. در یک بیت فقط می‌توان صفر یا یک منطقی را قرار داد.

مثلاً I0.0 بیانگر آدرس یک بیت از حافظه‌ی مربوط به ورودی‌های PLC می‌باشد که در آن می‌تواند مقدار صفر یا یک منطقی قرار گیرد.

نوع داده‌ی BYTE

اندازه‌ی این نوع داده، 8 بیت بوده و قالب آن به‌صورت نشان داده شده در شکل ۱۳-۷ می‌باشد. فرمت مقداردهی یک بایت به‌صورت باینری و Hex می‌باشد.

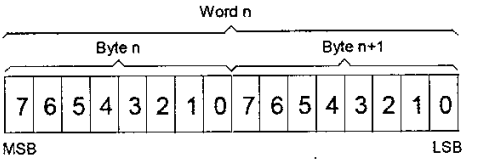


شکل ۱۳-۷ دیتا از نوع BYTE

مثلاً MB0 بیانگر یک بایت از حافظه‌ی PLC می‌باشد که در آن می‌توان دیتایی را به فرم باینری یا Hex قرار داد.

نوع داده‌ی WORD

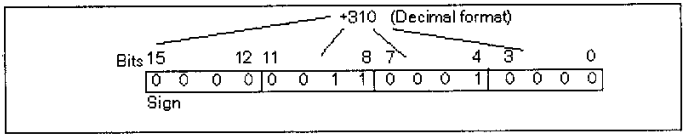
اندازه‌ی این نوع داده 16 بیت (معادل دو بایت) بوده و قالب آن به‌صورت نشان داده شده در شکل ۱۳-۸ می‌باشد. فرمت مقداردهی یک Word می‌تواند باینری، Hex، BCD یا دسیمال بدون علامت باشد.



شکل ۱۳-۸ ساختار داده‌ی Word

عددی که در نوع داده‌ی Word قرار می‌گیرد، یک عدد بدون علامت می‌باشد. البته اگر جهت مقداردهی از فرمت BCD استفاده شود، امکان ایجاد یک عدد علامت‌دار نیز وجود دارد. از آنجا که هر چهار رقم در کد BCD به یک رقم دسیمال تبدیل می‌شود، لذا دوازده بیت جهت عدد و چهار بیت باقیمانده جهت علامت در نظر گرفته می‌شود. از اینرو بیشترین و کمترین مقدار در فرم BCD عبارت است از 999+ و 999-.

به‌عنوان مثال در شکل ۱۳-۹ معادل دسیمال یک عدد BCD که در یک داده از نوع Word قرار گرفته است، نشان داده شده است.

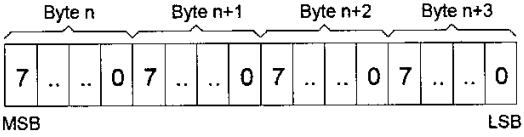


شکل ۱۳-۹ معادل دسیمال یک عدد BCD شانزده بیتی

اگر در چهار بیت علامت مقدار صفر قرار گرفته باشد، عدد مثبت بوده و اگر در آنها مقدار یک قرار گرفته باشد، عدد منفی می‌باشد.

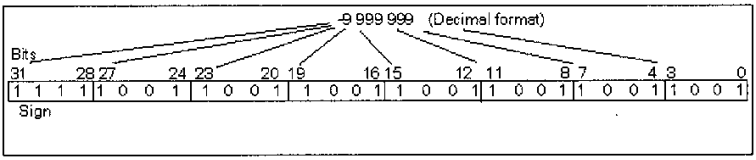
نوع داده‌ی Double Word

این نوع داده همانند نوع داده‌ی Word بوده با این تفاوت که اندازه‌ی آن دو برابر Word یعنی 32 بیت می‌باشد. همانطور که اشاره شده و در شکل ۱۳-۱۰ مشاهده می‌شود، یک D.Word از چهار بایت (32 بیت) تشکیل می‌شود.



شکل ۱۳-۱۰ نوع داده‌ی Double Word

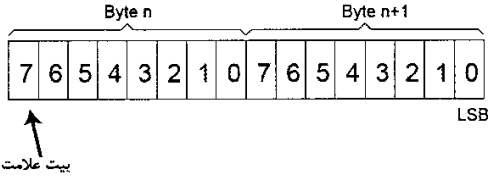
مشابه موارد بیان شده در مورد مقداردهی یک Word به فرم BDC، در اینجا نیز می‌توان یک دابل Word (D.Word) را آدرس‌دهی نمود. مثلاً در شکل ۱۳-۱۱ معادل دسیمال یک عدد BCD که در یک داده از نوع Double Word قرار گرفته است، نشان داده شده است.



شکل۱۳-۱۱ معادل دسیمال عدد BCD سی و دو بیتی

نوع داده‌ی Integer (INT)

همانطور که اشاره شد، در نوع داده‌ی Word و Double Word امکان قرارگیری اعداد صحیح علامت‌دار وجود نداشت، از اینرو جهت محاسبات از نوع داده‌ی دیگری به‌نام INT که بیانگر عدد صحیح 16 بیتی می‌باشد استفاده می‌شود. قالب نوع داده‌ی INT به‌صورت نشان داده شده در شکل ۱۳-۱۲ است.



شکل۱۳-۱۲ نوع داده‌ی INT

همانطور که مشخص است، بیت اول از سمت چپ، بیت علامت می‌باشد. بیت علامت می‌تواند مقادیر زیر را داشته باشد:

- 0: در این‌صورت عدد صحیح، مثبت در نظر گرفته می‌شود.
- 1: در این‌صورت عدد صحیح، منفی در نظر گرفته می‌شود.

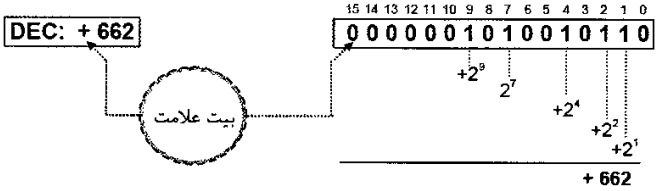
جهت محاسبه‌ی عدد صحیح به فرم INT کافیست مراحل زیر طی شود:

۱- مقدار دسیمال موجود در ۱۵ بیت (بدون احتساب بیت علامت) را مطابق روش بیان شده جهت تبدیل باینری به دسیمال، به یک عدد دسیمال تبدیل می‌نماییم.

۲- بیت علامت را با توجه به مقدار درون آن به یک عدد دسیمال تبدیل می‌کنیم. اگر مقدار بیت علامت، عدد صفر باشد که در محاسبات بی‌تأثیر بوده و حاصل همان عدد دسیمالی‌است که از تبدیل 15 بیت (بدون احتساب بیت علامت) محاسبه شده است.

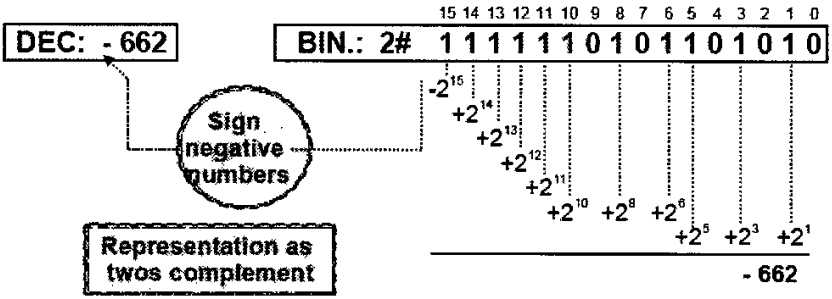
۳- اگر بیت علامت دارای مقدار یک باشد، به عدد دسیمال تبدیل شده و به‌صورت منفی در نظر گرفته می‌شود. این مقدار با مقدار به‌دست آمده از 15 بیت دیگر جمع می‌شود.

مثلاً در شکل ۱۳-۱۳ نحوه محاسبه‌ی عدد صحیح در یک نوع داده INT نشان داده شده است.



شکل۱۳-۱۳ تبدیل یک عدد صحیح با علامت به باینری

مثال ۱۰-۷ نوع داده‌ی INT مطابق شکل ۱۳-۱۴ را به فرم عدد صحیح محاسبه نمایید.



شکل ۱۳-۱۴ حل مثال ۱۰-۷

همانطور که در شکل ۱۳-۱۴ نشان داده شده است، چون بیت علامت دارای مقدار 1 است، بنابراین کل عدد یک عدد منفی در نظر گرفته می‌شود. در خانه‌های دیگری که عدد 1 قرار گرفته شده است، عدد 2 به توان شماره‌ی خانه رسیده و با سایر اعداد جمع می‌شود. حاصل این جمع با مقدار 15^2- (32768-) جمع شده که حاصل برابر 662- می‌شود. توجه شود که مقدار 15^2- مربوط به مقدار بیت علامت است.

در نرم‌افزار STEP7 یک عدد INT را می‌توان به‌صورت معمولی آدرس‌دهی نمود، مثلاً 12,456,7 و . . . در این-حالت نیاز به استفاده از مشخصه‌ی خاصی جهت مشخص نمودن نوع داده نمی‌باشد.

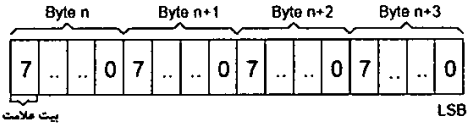
بیشترین و کمترین مقدار یک داده‌ی INT

جهت محاسبه‌ی بیشترین مقدار یک داده‌ی INT کافیست که به غیر از بیت علامت، سایر بیت‌های آن داده 1 شوند. در این‌صورت داریم: 32767 = 1- (15^2). همانطور که مشخص است بیشترین مقداری که می‌تواند در یک عدد صحیح شانزده بیتی قرار گیرد، عدد 32767 می‌باشد.

کمترین عدد INT وقتی است که بیت علامت ۱ یعنی عدد منفی و سایر بیت ها صفر باشند که این عدد برابر با - 32768 خواهد بود.

نوع داده‌ی Double Integer (D.INT)

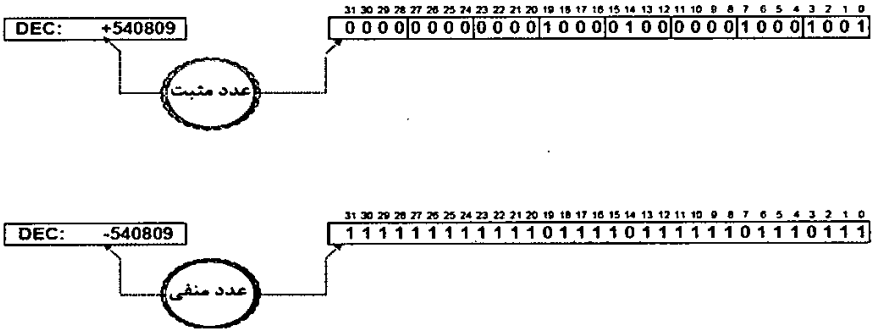
این نوع داده مانند نوع داده‌ی INT می‌باشد با این تفاوت که اندازه‌ی آن دو برابر یک داده INT است، یعنی 32 بیت، از اینرو مقادیر بزرگتری را می‌توان درآن قرار داد. در شکل۱۳-۱۵ فرمت داده‌ی دابل INT نشان داده شده است.



شکل ۱۳-۱۵ فرمت داده‌ی دابل INT

فصل ۱۳

در شکل ۱۳-۱۶ با یک مثال نحوه‌ی محاسبه‌ی مقدار درون یک داده‌ی دابل INT نشان داده شده است.



شکل ۱۳-۱۶ معادل باینری یک عدد صحیح ۳۲ بیتی علامت دار

تذکر: جهت مشخص نمودن یک عدد به فرم D.INT (دابل INT) در برنامه‌نویسی باید قبل از عدد، علامت L# قرار داد. مثلاً جهت مشخص نمودن عدد 23000 به فرم دابل INT، باید عدد را به‌صورت L# 23000 نوشت، در غیر این‌صورت عدد به‌عنوان یک داده‌ی INT در نظر گرفته می‌شود. بیشترین مقداری که در یک داده‌ی D.INT می‌توان قرار داد عبارت است از 2147483647+ = 1 -(2^31).

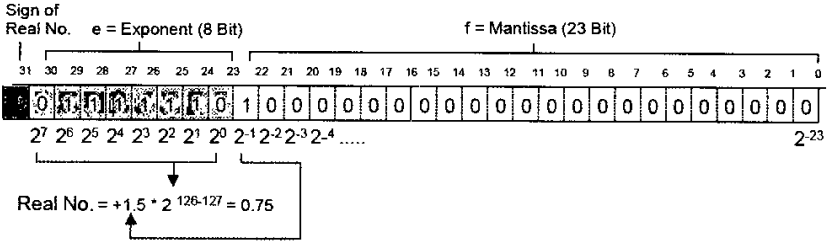
نوع داده‌ی Real (Floating-point Number, 32 Bit)

این نوع داده دارای اندازه‌ی 32 بیت بوده و مخصوص اعداد اعشاری می‌باشد. دو نوع داده‌ی INT و دابل INT مخصوص اعداد صحیح بودند، در حالی‌که بسیاری از کمیت‌هایی که در PLC پردازش می‌شوند به‌صورت اعداد اعشاری می‌باشند. از این نوع کمیت‌ها می‌توان به دما، فشار، ارتفاع سطح و . . . اشاره نمود. از اینرو باید نوع خاصی از داده وجود داشته باشد که بتواند اعداد اعشاری را پوشش دهد. برای این منظور از نوع داده‌ی Real استفاده می‌شود.

همانطور که بیان شد، داده‌ی Real از 32 بیت تشکیل می‌شود، اما ساختار آن با داده‌ی D.INT متفاوت است. یک عدد Real از فرمول زیر محاسبه می‌شود:

$$\text{عدد Real} = \text{علامت} * (1.f) * (2^{e-127})$$

همانطور که در شکل ۱۳-۱۷ مشاهده می‌شود، بیت شماره‌ی 31 جهت علامت اختصاص یافته شده است. بیت‌های شماره 23 الی 30 جهت ساختن مقدار e، و مابقی بیت‌ها جهت ساختن مقدار f به‌کار می‌روند. هنگامی‌که مقادیر f و e و علامت (مثبت یا منفی) محاسبه شدند، آنها را در فرمول بالا قرار داده و عدد Real محاسبه می‌شود.

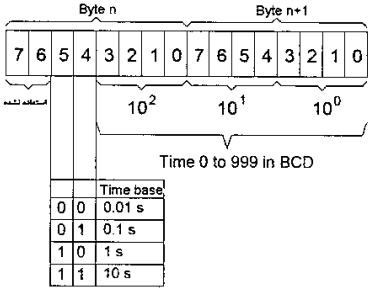


شکل ۱۳-۱۷ ساختار داده‌ی Real همراه با مثال

همانطور که در شکل مشخص است، توان خانه‌های مربوط به عدد e، از 2^0 الی 2^7 می‌باشد. توان مربوط به خانه‌های f، از 2^{-1} الی 2^{-23} می‌باشد. در شکل ۱۳-۱۷ عدد f مقدار 0.5 و مقدار e عدد 127 می‌باشد، از قراردادن مقادیر محاسبه شده در فرمول، مقدار 0.75 محاسبه می‌شود.

نوع داده‌ی S5 Time

این نوع داده که برای محاسبه‌ی زمان استفاده می‌شود، از 16 بیت تشکیل شده است. این فرمت خاص Step7 است و کاربرد آن به‌طور عمده در تایمرهای برنامه می‌باشد. ساختار این نوع داده به‌صورت نشان داده شده در شکل ۱۳-۱۸ می‌باشد.



شکل ۱۳-۱۸ قالب نوع داده‌ی S5 Time

همانطور که در شکل ۱۳-۱۸ مشاهده می‌شود، دو بیت آخر از سمت چپ به‌صورت استفاده نشده می‌باشند و در محاسبه‌ی زمان نقشی ندارند. از بایت شماره‌ی n، بیت‌های شماره‌ی 4 و 5 به‌عنوان پله زمانی در نظر گرفته شده‌اند و با توجه به عددی که در آنها قرار می‌گیرد، چهار پله زمانی به‌دست می‌آید. مابقی بیت‌ها جهت محاسبه‌ی زمان به فرم BCD استفاده می‌شوند. عدد محاسبه شده به عنوان زمان، حداکثر می‌تواند عدد 999 باشد که اگر پله زمانی (۱۰ثانیه) انتخاب شود، حداکثر زمان قابل محاسبه مقدار 9990 S معادل ۲ ساعت و ۴۶ دقیقه می‌باشد.

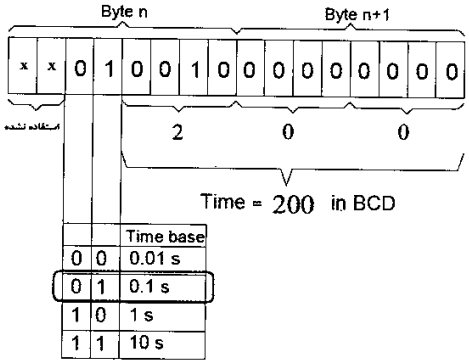
انتخاب صحیح پله زمانی یکی از نکات مهم کار با این نوع داده می‌باشد، که البته این عمل فقط در زبان برنامه‌نویسی STL امکان‌پذیر است. پله زمانی مشخص‌کننده‌ی حد تفکیک[1] زمان می‌باشد، بدین معنی‌که حداقل فواصل زمانی قابل محاسبه چقدر باشد. شکل ۱۳-۱۹ این موضوع را نشان می‌دهد.

Resolution	Range
0.01 s	10 ms to 9 s 990 ms
0.1 s	100 ms to 1 m 39 s 900 ms
1 s	1 s to 16 m 39 s
10 s	10 s to 2 h 46 m 30 s

شکل ۱۳-۱۹ پله های زمانی و حداکثر زمان قابل محاسبه برای هر کدام

فصل ۱۳

مثال ۱۳-۸ داده‌ی زیر به فرم S5 Time است. زمان و میزان دقت آنرا محاسبه کنید.



شکل ۱۳-۲۰ مربوط به مثال ۱۳-۸

همانطور که در شکل ۱۳-۲۰ مشخص است، پله زمانی 0.1 S در نظر گرفته شده است. زمان نیز عدد 200 در نظر گرفته شده و بنابراین داریم: 200 * 0.1 S = 20 S.

1. Resolution

از آنجایی که پله زمانی 0.1 S در نظر گرفته شده است، زمان در ضرایب 0.1S قابل تنظیم می‌باشد. همانطور که ذکر شد، این عمل در زبان برنامه‌نویسی STL امکان‌پذیر است.

جهت آدرس‌دهی این نوع داده باید به یکی از دو روش زیر عمل نمود:

الف. W#16#txyz که در آن

t: پله‌ی زمانی

Xyz: مقدار زمان به‌فرم BCD

ب. زمان S5T# که در آن زمان به فرم aH bM cS dMS می‌باشد، و در آن a و b و c و d ضریب می‌باشند, و حروف H,M,S,MS طبق جدول ۱۳-۶ به زمان اشاره می‌کنند.

مثلاً S5T#2H46M30S اشاره به زمان دو ساعت و چهل و شش دقیقه و سی‌ثانیه دارد. از آنجا که زمان قابل محاسبه توسط این نوع داده مقدار کمی‌است، لذا نوع داده‌ی دیگری به نام Time ایجاد شده است که توانایی محاسبه‌ی زمان‌های بالاتر را دارا می‌باشد.

جدول ۱۳-۶ حروف قابل استفاده در فرمت S5time

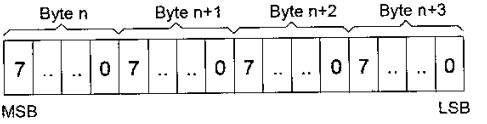
H	hour
M	minute
S	Second
Ms	millisecond

نوع داده‌ی Time

این نوع داده از 32 بیت تشکیل شده است. محاسبه‌ی زمان توسط آن به‌سادگی انجام می‌پذیرد. مقدار عدد موجود در این داده به فرم D.INT خوانده شده و در زمان 1 ms ضرب می‌شود.

به‌عنوان مثال اگر عدد درون این داده به فرم عدد صحیح خوانده شده و برابر 2000 باشد، زمان مورد نظر 2 S در نظر گرفته می‌شود.

در شکل ۱۳-۲۱ ساختار این نوع داده نشان داده شده است.



شکل ۱۳-۲۱ ساختار نوع داده‌ی Time

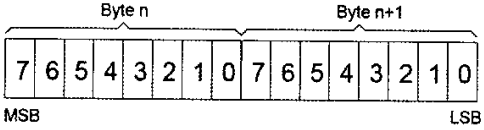
از آنجا که بزرگترین عدد در فرم D.INT عدد 1 - (2^31) می‌باشد، لذا بیشترین زمان قابل محاسبه توسط این نوع داده 24d20h31m23s647ms می‌باشد که به فرم T# 24d20h31m23s647ms نمایش داده می‌شود. در زمان

نشان داده شده علامت #T مبین نوع داده‌ی Time بوده و حروف d، h ، m، s و ms به‌ترتیب بیانگر روز، ساعت، دقیقه، ثانیه و میلی‌ثانیه می‌باشد. همانطور که دیده می‌شود، نسبت به فرمت S5TIME در اینجا استفاده از d اضافه گردیده است و زمان بسیار وسیع‌تر شده است.

یکی از کاربردهای این نوع داده در تایمرهای IEC می‌باشد که در جلدهای بعدی همین کتاب تشریح خواهد شد.

نوع داده‌ی DATE

این نوع داده به صورت شانزده بیتی بوده و ساختار آن به‌صورت نشان داده شده در شکل ۱۳-۲۲ می‌باشد.



شکل۱۳-۲۲ ساختار نوع داده‌ی Date

فصل ۱۳

در داده‌ی فوق می‌توان اعداد مختلفی به فرم دسیمال بدون علامت قرار داد که به‌عنوان روز با اضافه‌شدن به تاریخ 1990-01-01، تاریخ جدیدی را به‌وجود آورد. مثلاً اگر در این داده مقدار 20 قرار گرفته باشد، 20 روز به تاریخ 1990-01-01 افزوده شده و بیانگر تاریخ 1990-01-21 خواهد بود. حداکثر تاریخ قابل ذخیره‌سازی در این نوع داده، تاریخ 2168-12-31 می‌باشد.

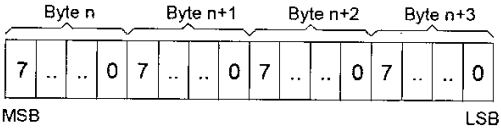
این نوع داده به‌صورت (تاریخ #D) آدرس‌دهی می‌گردد که علامت #D قبل از تاریخ، بیانگر نوع داده‌ی Date می‌باشد.

کاربرد اصلی این نوع داده در بلاک‌های سیستمی و سایر بلاک‌هایی است که این نوع داده را می‌پذیرند.

نوع داده‌ی TIME OF DAY

این نوع داده به‌منظور نمایش زمان مشخصی از روز به فرم TOD#H:M:S:Ms به‌کار می‌رود که در آن H، M، S و Ms به‌ترتیب بیانگر ساعت، دقیقه، ثانیه و میلی‌ثانیه می‌باشد و علامت #TOD نشان‌دهنده‌ی داده از نوع Time of day است. مثلاً مقدار TOD# 10:12:23:000 به ساعت ده و دوازده دقیقه و بیست و سه ثانیه اشاره دارد. این داده دارای اندازه‌ی 32 بیتی می‌باشد. اطلاعات در این داده به‌صورت دسیمال بدون علامت قرار گرفته و بیانگر زمان مورد نظر در روز بر حسب میلی‌ثانیه می‌باشد. مثلاً اگر در این داده مقدار 50000 قرار بگیرد، نشان‌دهنده زمان 00:00:50:00 یعنی پنجاه ثانیه‌ی بامداد می‌باشد.

ساختار این نوع داده در شکل ۱۳-۲۳ نشان داده شده است.



شکل ۱۳-۲۳ ساختار داده‌ی TOD

کاربرد اصلی این نوع داده در بلاک‌های سیستمی و سایر بلاک‌هایی است که این نوع داده را می‌پذیرند.

نوع داده‌ی CHAR (کاراکتر)

در این نوع داده، امکان ایجاد و ذخیره‌سازی کاراکتر وجود دارد. توسط تعدادی کاراکتر می‌توان پیامی را ایجاد و استفاده نمود.

هر کاراکتر 8 بیت فضا مصرف می‌کند و به‌صورت یک حرف یا عدد که بین دو علامت ' ' قرار می‌گیرد نشان‌داده می‌شود، که در آن کاراکتر می‌تواند یک حرف دلخواه باشد؛ مثلاً 'E' .

۱۳-۳-۲ Complex Data Types

در این گروه، داده‌هایی که از نظر اندازه بزرگتر از داده‌های پایه هستند قرار می‌گیرند. این داده‌ها را می‌توان از نوع داده مختلط نیز قلمداد نمود. جدول ۱۳-۷ انواع این داده‌ها را نشان می‌دهد. تشریح و مثال‌های کاربردی در مورد این متغیرها در جلدهای بعدی کتاب آمده است.

جدول ۱۳-۷ انواع داده‌ی Complex

Keyword	Length (in bits)	Example	
DATE_AND_TIME	64	DT#01-08-24-12:14:55:234-1	
STRING (character string with max. 254 characters)	8 * (number of characters +2)	'This is a string' 'SIEMENS'	
ARRAY (Group of elements of the same data type)	user-defined	Measured values: ARRAY[1..20] INT	
STRUCT (Group of elements of different data types)	user-defined	Motor: STRUCT Speed : INT Current: REAL END_STRUCT	
UDT (User Defined Data Type = "Template" consisting of elementary or complex data types	user-defined	UDT as block	UDT as array element
		STRUCT Speed : INT Current: REAL END_STRUCT	Drive: ARRAY[1..4] UDT1

جزئیات در جدول ۱۳-۸ آمده است.

جدول ۱۳-۸ جزئیات پارامترهای Complex

نوع	فرمت	سایز	مثال
DATE_AND_TIME	ms.DT#yy-mm-dd-h:m:s سال می‌تواند تا ۲۰۸۹ به‌کار رود. مقدار فوق در ۸ بایت به‌صورت BCD ذخیره می‌گردد.	64 Bit	DT#1993-12-25-8:01:1.23
STRING	STRING[n] n میتواند حداکثر ۲۵۴ باشد.	n+2 Byte	STRING[4]

	به‌عنوان مثال اگر بخواهیم کلمه 'Test' را داخل متغیر String بریزیم باید برای آن حداقل String[4] را به‌کار ببریم.	
ARRAY	فرمت آرایه یک بعدی به‌صورت زیر است ARRAY[x1..x2] که در آن x1 و x2 دو عدد صحیح هستند و می‌توانند بین ۳۲۷۶۸- تا ۳۲۷۶۸+ باشند؛ یعنی کلاً ۶۵۵۳۵عنصر میتوان تعریف کرد.	Test[1..3] که معادل با سه عنصر زیر است : Test[1], Test[2], Test[3]
	فرمت آرایه دو بعدی به‌صورت زیر است ARRAY[x1..x2] و [y1..y2] که در آن x1 و x2 و y1 و y2 عدد صحیح هستند و می‌توانند بین ۳۲۷۶۸- تا ۳۲۷۶۸+ باشند ولی کل عناصرنباید از ۶۵۵۳۵ بیشتر شود.	Test[1..3,1..2] که معادل با سه عنصر زیر است Test[1,1] Test[1,2] Test[2,1] Test[2,2] Test[3,1] Test[3,2]
	آرایه می‌تواند حداکثر ۶ بعدی باشد ولی کل عناصرنباید از ۶۵۵۳۵ بیشتر شود. تذکر: عناصر آرایه باید از یک جنس باشند مثلاً همه می‌توانند نوع Bool یا همگی به‌فرض از نوع INT باشند. نوع عناصر را در برنامه باید تعریف کرد.	یک عنصر از آرایه ۶ بعدی به‌صورت زیر است: Test[1,1,2,1,5,4]
STRUCT	نام Struct در این قسمت نام و نوع متغیرها تعریف می‌شود. End Struct	Test Struct a Bool b Int c Word End Struct

نوع داده‌ی DATE AND TIME

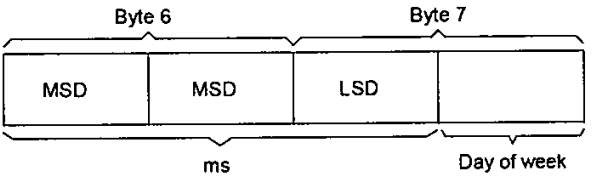
این نوع داده، 64 بیت مصرف می‌کند و در آن می‌توان زمان و تاریخ یک روز را مشخص نمود. طریقه نمایش به‌صورت Yr-Mo-Day-hr:min:s:ms می‌باشد که در آن Yr، Mo و Day به‌ترتیب بیانگر سال، ماه و روز مورد نظر می‌باشند. hr ، min ، s و ms بیانگر ساعت، دقیقه، ثانیه و میلی‌ثانیه می‌باشد. بدین‌ترتیب می‌توان تاریخ و زمان یک روز را مشخص نمود؛ مثلاً DT#1995-02-02-08:30:01.999.

همانطور که اشاره شد، داده‌ی Date And Time دارای اندازه‌ی 8 بایت (معادل 64 بیت) می‌باشد. جدول۱۳-۹ عملکرد هر یک از بایت‌ها را نشان می‌دهد.

جدول ۱۳-۹

Byte	Contents	Example
0	Year	B#16#93
1	Month	B#16#12
2	Day	B#16#25
3	Hour	B#16#08
4	Minute	B#16#12
5	Second	B#16#34
6	Two most significant digits of MSEC	B#16#56
7 (4MSB)	Two least significant digits of MSEC	B#16#7
7 (4LSB)	Day of week 1 = Sunday 2 = Monday .. 7 = Saturday	B#16#_5

همانطور که در جدول ۱۳-۹ نشان داده شده است، بایت شماره شش و چهار بیت از بایت شماره هفت به منظور ذخیره‌سازی زمان برحسب میلی‌ثانیه در نظر گرفته شده‌اند. چهار بیت باقیمانده از بایت شماره‌ی هفت، به روز هفته اختصاص دارد. این موضوع در شکل۱۳-۲۴ نیز نشان داده شده است.



شکل۱۳-۲۴ ساختار بایت ششم و هفتم از داده‌ی Date And Time

نکته‌ای که باید توجه نمود این است که توسط دستورات معمولی نمی‌توان این نوع داده را مقداردهی نمود، مگر اینکه تک‌تک بایت‌ها را جداگانه مقداردهی کرد.

برای مقداردهی این نوع داده، می‌توان از FC3: D_TOD_DT که در کتابخانه نرم‌افزار موجود است استفاده نمود. مباحث مرتبط با این موضوع در کتاب سطح تکمیلی همین کتاب ارائه خواهد شد.

نوع داده‌ی آرایه

شبیه بحث برنامه‌نویسی کامپیوتر، در PLC نیز می‌توان تعداد زیادی متغیر را با یک نام ولی با شماره متفاوت ایجاد نمود. به‌عنوان مثال اگر آرایه‌ای به‌صورت myData[1..10] تعریف شود به یک مجموعه ده عضوی اشاره می‌کند که اولین

عضو آن myData[1] و آخرین عضو آن myData[10] می‌باشد. مثال فوق یک آرایه یک بعدی را نشان می‌دهد. آرایه‌ها می‌توانند چند بعدی باشند. به‌عنوان مثال عنصر test[1,3] به عضوی از آرایه دو بعدی اشاره می‌کند. آرایه می‌تواند ماکزیمم ۶ بعدی باشد.

از آرایه به منظور ایجاد متغیرهایی با نوع داده‌ی پایه[1] که نوع دیتای مشابه دارند استفاده می‌شود.

نوع داده‌ی STRUCT

این نوع داده نیز به‌منظور ایجاد متغیرها با نوع داده‌ی ابتدایی استفاده می‌شود. در یک STRUCT می‌توان داده‌هایی با نوع مختلف را ایجاد نمود. به‌عنوان مثال می‌توان یک دیتا از نوع Struct برای یک مخزن در یک فرآیند تعریف کرد که در آن سیگنال آنالوگ فشار و سطح و دما و سوئیچ های سطح و ... همگی قابل تعریف باشند.

فصل ۱۳

نوع داده‌ی STRING

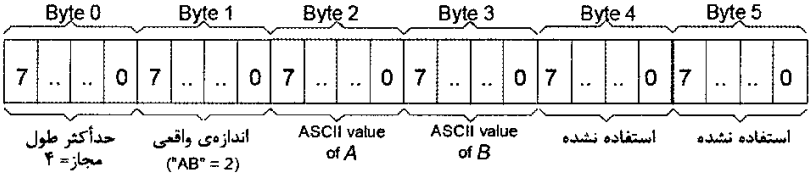
توسط یک داده از نوع STRING، می‌توان مجموعه‌ای از کاراکتر را (حداکثر تا 254 کاراکتر) ایجاد نمود. بدین‌ترتیب می‌توان یک پیام دلخواه ایجاد نمود. هنگامی‌که یک داده از نوع STRING تعریف می‌شود، تعداد کاراکترهای آن نیز مشخص می‌شود. به‌عنوان مثال برای ذخیره سازی دیتای استرینگ 'this is for test' به یک استرینگ ۱۶ کاراکتری نیاز است که به‌صورت STRING[18] تعریف می‌گردد.

هر کاراکتر نیاز به یک بایت فضا خواهد داشت، دو بایت نیز به منظور مشخص نمودن تعداد کاراکترهای مجاز و تعداد کاراکترهای استفاده شده اختصاص پیدا می‌کند. از اینرو تعداد بایت‌های مصرفی توسط یک داده از نوع STRING برابراست با تعداد کاراکترها + 2.

برای درک موضوع به ارائه‌ی مثالی می‌پردازیم.

مثال ۱۳-۹ متغیری به فرم STRING[4] با محتویات 'AB' ایجاد شده است. اندازه‌ی این متغیر و نحوه‌ی اختصاص بیت‌ها را توضیح دهید.

حل: عددی که در داخل علامت [ ] نوشته شده، بیانگر تعداد کاراکترهای استفاده شده از این نوع داده می‌باشد. در این مثال تعداد کاراکترهای قابل استفاده برابر 4 کاراکتر و کارکترهای استفاده شده برابر 2 کاراکتر می‌باشد. همانطور که اشاره شد، اندازه‌ی متغیر ایجاد شده برابر است با تعداد کاراکترهای مجاز + 2 که می‌شود 6 بایت. از آنجایی که فقط دو کاراکتر تعریف شده است، دو بایت این متغیر خالی و بدون استفاده مانده است. در شکل ۱۳-۲۵ ساختار این متغیر و نحوه‌ی اختصاص اطلاعات در آن نشان داده شده است.



شکل۱۳-۲۵ ساختار متغیر مثال ۱۳-۹

1. Elementray Data Types

نوع داده‌ی UDT

UDT که مخفف User define data type می‌باشد، می‌تواند به عنوان یک الگو (که خود از چندین متغیر با نوع داده‌ی متفاوت تشکیل شده است) برای تعریف نوع داده در یک Data Block یا در Local Data که در کتاب سطح پیشرفته تشریح می‌شوند استفاده شود. یک UDT می‌تواند از تعدادی داده‌ی پایه یا مختلط تشکیل شده باشد.

۱۳-۳-۳ Parameter Types

در این گروه از داده‌ها، انواع داده‌هایی مثل تایمر، کانتر (شمارنده) و . . . قرار می‌گیرند. همچنین داده‌هایی مثل داده‌ی ANY و POINTER نیز در این گروه از داده‌ها قرار می‌گیرند.

تشریح و ارائه مثال‌های کاربردی در این مورد را به جلدهای بعدی کتاب موکول می‌کنیم.

جدول ۱۳-۱۰ داده‌های پارامتری

نوع	فرمت	سایز	مثال
Timer	عدد صحیح T	2 bytes	T1
Counter	عدد صحیح C	2 bytes	C1
Pointer	آدرس P#	6 bytes	P#M50.0
Any	-	10 bytes	-

۱۳-۴ نحوه آدرس‌دهی متغیرهای حافظه

۱۳-۴-۱ نکات کلی آدرس‌دهی

متغیر حافظه که در برنامه نویسی از آنها استفاده می‌شود در جدول ۱۳-۱۱ لیست شده‌اند.

جدول ۱۳-۱۱ انواع متغیرهای حافظه

نوع متغیر حافظه	علامت اختصاری در برنامه	کاربرد
Input	I	ورودی‌های دیجیتال و آنالوگ
Output	Q	خروجی‌های دیجیتال و آنالوگ
Memory Bit	M	نتایج میان برنامه
Timer	T	زمان سنجی
Counter	C	شمارش
Local Data	L	مقادیر محلی داخل یک بلاک

I و Q و M و L می‌توانند به‌صورت بیت یا Byte یا Word یا Double Word آدرس‌دهی شوند. جدول ۱۳-۱۲ این موارد را نشان می‌دهد.

جدول ۱۳-۱۲ انواع روش‌های آدرس‌دهی متغیرهای حافظه

	نوع داده	نحوه نمایش	مثال
Input	Bit	I	I 0.1
	Byte	IB	IB 1
	Word	IW	IW 2
	Dword	ID	ID 8
Output	Bit	Q	Q0.1
	Byte	QB	QB1
	Word	QW	QW2
	Dword	QD	QD8
Memory Bit	Bit	M	M 0.1
	Byte	MB	MB 1
	Word	MW	MW 2
	Dword	MD	MD8
Local Data	Bit	L	L0.1
	Byte	LB	LB 1
	Word	LW	LW 2
	Dword	LD	LD8

فصل ۱۳

نکات قابل توجه

- برای آدرس‌دهی یک بیت باید ابتدا شماره بایت را نوشته و سپس با گذاشتن نقطه آدرس بیت را در آن بایت مشخص کنیم. جدول ۱۳-۱۳ مثالی در این زمینه ارائه کرده است.

جدول ۱۳-۱۳ مثالی از آدرس دهی بیت

آدرس بیت	توضیح
0.0	بیت صفر از بایت صفر
4.7	بیت ۷ از بایت ۴

بدیهی است عدد سمت راست که بیت را مشخص می‌کند نمی‌تواند از عدد ۷ بزرگتر باشد چون در یک بایت ۸ بیت داریم که از صفر تا ۷ شماره دهی می‌شوند، ازاینرو آدرسی مانند 0.8 نادرست خواهد بود.

- برای آدرس‌دهی بایت، عددی که در جلوی بایت نوشته می‌شود یک عدد صحیح است که می‌تواند تا حدی که حافظه اجازه می‌دهد بالا برود.
- برای آدرس‌دهی Word مرسوم این است که عددی که در روبروی آن نوشته می‌شود زوج باشد، علت این است که هر Word با Word بعدی به اندازه دو بایت فاصله دارد. اگر این نکته مراعات شود می‌توان آدرس فرد را

به‌کار برد. به‌عنوان مثال به‌کاربردن IW4,IW2,IW0 درست است، همینطور به‌کار بردن IW1 و IW3 و IW5 نیز درست است ولی به‌کار بردن IW0 و IW1 اشتباه است زیرا ایندو با یکدیگر در بایت شماره ۱ مشترک می‌باشند. به جدول ۱۳-۱۴ توجه کنید.

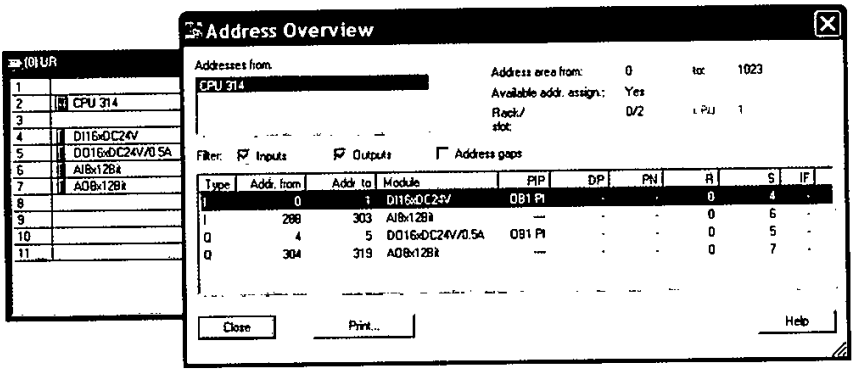
جدول ۱۳-۱۴ خطای Overlap

IW 0	=	BYTE0	+	BYTE1
IW 1	=	BYTE1	+	BYTE2

- نکته فوق را برای Double Word نیز باید رعایت کرد. یعنی هر آدرس با آدرس بعدی باید ۴ بایت فاصله داشته باشد. مثلاً MD0 و MD4.

- ورودی یا خروجی PLC می‌تواند از جنس Bit یا Byte یا Word یا Dword باشد. به‌عنوان مثال برای وضعیت یک سوئیچ که به کارت DI متصل است و فقط حالت صفر یا یک دارد یک Bit کافی است ولی وقتی لازم باشد ۸ ورودی DI به‌صورت یک‌جا خوانده شوند یک Byte لازم است. برای ورودی و خروجی‌های آنالوگ یک Word مورد نیاز خواهد بود، به همین ترتیب برای خواندن ۳۲ ورودی دیجیتال یا دو ورودی آنالوگ متوالی نیاز به یک Double Word خواهد بود.

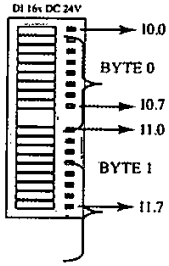
- در Hwconfig پس از اینکه کارت‌های دیجیتال و آنالوگ وارد گردید می‌توان از طریق منوی View با انتخاب گزینه Address Overview لیست آدرس کارت‌ها را مشاهده نمود. به شکل ۱۳-۲۶ توجه کنید.



شکل ۱۳-۲۶ مشاهده آدرس‌های اختصاص داده شده به کارت‌ها از طریق پنجره Address Overview

۱۳-۴-۲ آدرس‌دهی کارت‌های دیجیتال

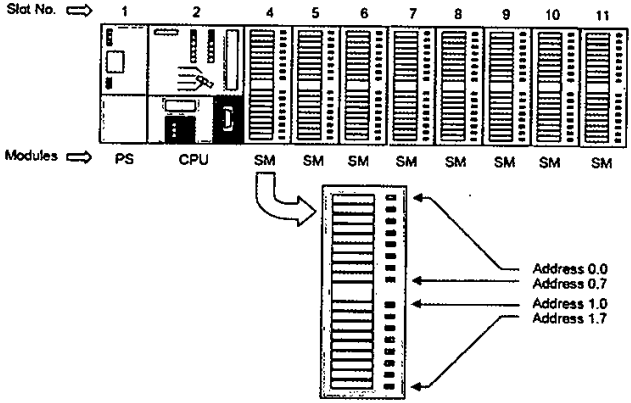
در کارت‌های دیجیتال به‌ازای هر کانال ورودی یا خروجی، یک بیت از ناحیه‌ی آدرس‌ها (PII) اختصاص داده می‌شود. در واقع می‌توان گفت به ازای هر هشت کانال ورودی یا خروجی دیجیتال، یک بایت از حافظه اختصاص داده می‌شود. برای آدرس‌دهی هر ورودی یا خروجی لازم است ابتدا شماره‌ی بایت اختصاص داده شده را مشخص نموده و سپس شماره‌ی بیت اختصاص داده شده را ذکر نمود. برای درک بهتر موضوع، شکل ۱۳-۲۷ را در نظر بگیرید. در این شکل یک کارت ورودی با 16 کانال ورودی نشان داده شده است.



شکل ۱۳-۲۷ آدرس‌های اختصاص داده شده به کارت DI 16x DC 24V

همانطور که بیان شد، به ازای هر هشت کانال یک بایت از ناحیه‌ی آدرس‌ها اختصاص داده می‌شود. بنابراین به هشت کانال اول کارت، آدرس Byte 0 و به هشت کانال دوم کارت، آدرس Byte 1 اختصاص داده شده است.

در شکل ۱۳-۲۷ مشخص است که به هشت کانال اول، آدرس Byte 0 اختصاص داده شده است. پس آدرس کانال اول I 0.0 و آدرس کانال آخر این کارت I1.7 خواهد بود.

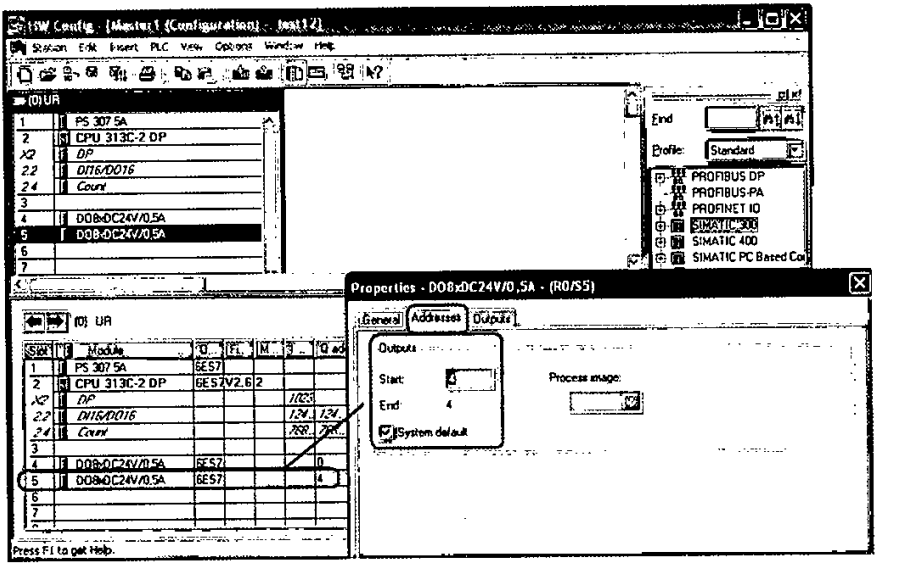


شکل ۱۳-۲۸ آدرس‌دهی کارت‌های دیجیتال

وقتی کارت دیجتال در محیط HW Config وارد می‌شود، برنامه به‌طور خودکار آدرسی را به آن اختصاص می دهد. این آدرس یک عدد شروع و یک عدد پایان دارد. طول این بازه به تعداد کانال کارت بستگی دارد.

آدرس دهی کارت دیجیتال خروجی

شکل ۱۳-۲۹ یک کارت دیجیتال خروجی ۸ کاناله در HW Config را نشان می‌دهد. با دوبار کلیک نمودن روی کارت در سربرگ Address می‌توان دید که شروع و پایان یک عدد است (در این شکل عدد ۴ است) زیرا این کارت فقط یک بایت را در حافظه اشغال می‌کند. اگر به پنجره Detail View رجوع کنیم نیز در روبروی کارت یک عدد در ستون Q Address نشان داده می‌شود.



شکل ۱۳-۲۹ آدرس کارت DO در HW Config

با توجه به شکل ۱۳-۲۹ کاربر متوجه خواهد شد که آدرس‌های این کارت از Q4.0 شروع و به Q4.7 ختم می شود.

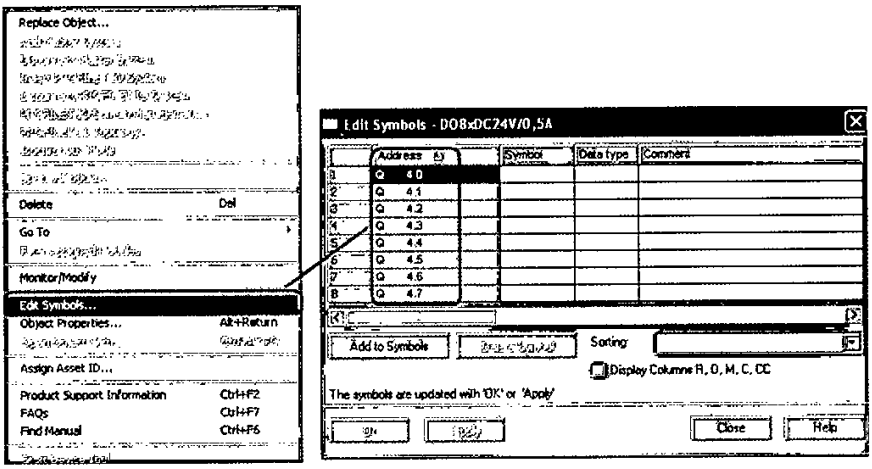
تذکر ۱: آدرس شروع و پایان به‌طور خودکار توسط برنامه داده می‌شود و همانطور که در شکل ۱۳-۲۹ دیده می شود، گزینه System Default فعال است که با غیرفعال کردن این گزینه کاربر می‌تواند آدرس دلخواه خود را وارد نماید.

تذکر ۲: آدرسی که به‌طور خودکار توسط برنامه داده می‌شود به اسلاتی که کارت در آن قرار می‌گیرد بستگی دارد. به‌عنوان مثال اگر کارت نشان داده شده در شکل ۱۳-۲۹ را به یک اسلات پایین‌تر منتقل کنیم، برنامه آدرس 8 را برای آن در نظر می‌گیرد.

تذکر ۳: در شکل ۱۳-۲۹ دو کارت DO هشت ورودی در اسلات‌های ۴ و ۵ دیده می‌شود به کارت اول آدرس 0 و به کارت دوم آدرس 4 داده شده است. بایت‌های 1 تا 3 استفاده نشده‌اند. برای استفاده بهینه از حافظه می‌توان آدرس‌ها را تغییر داد و به‌دنبال هم قرار داد تا بین آنها فضای خالی بدون استفاده نباشد.

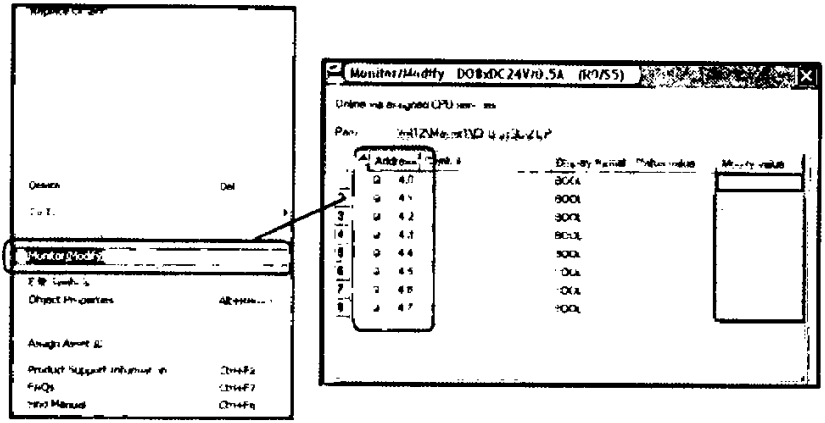
اگر کاربر بخواهد در هیمن پنجره آدرس کانال‌ها را به‌طور دقیق و تفکیکی ببیند می‌تواند به یکی از دو روش زیر عمل کند:

- روی کارت کلیک راست کرده و گزینه Edit Symbols که برای اختصاص نام به کانال‌هاست انتخاب شود، در این‌حالت پنجره‌ای مانند شکل ۱۳-۳۰ نشان داده می‌شود که آدرس واقعی همه کانال‌ها در آن لیست شده است. این پنجره در ادامه در بحث آدرس‌دهی مطلق و سمبلیک تشریح می‌گردد.



شکل ۱۳-۳۰ مشاهده آدرس‌های هر کانال خروجی دیجیتال توسط جدول سمبل ها

- روش دیگر برای مشاهده آدرس دقیق هر کانال این است که وقتی به PLC متصل هستیم و سخت‌افزار به آن دانلود شده است، روی کارت مورد نظر کلیک راست کرده و گزینه Monitor/Modify را انتخاب کنیم. در اینصورت پنجره‌ای مانند شکل ۱۳-۳۱ نمایش داده می‌شود. جزئیات این پنجره در فصل ۲۵ کار Online با PLC توضیح داده خواهد شد.

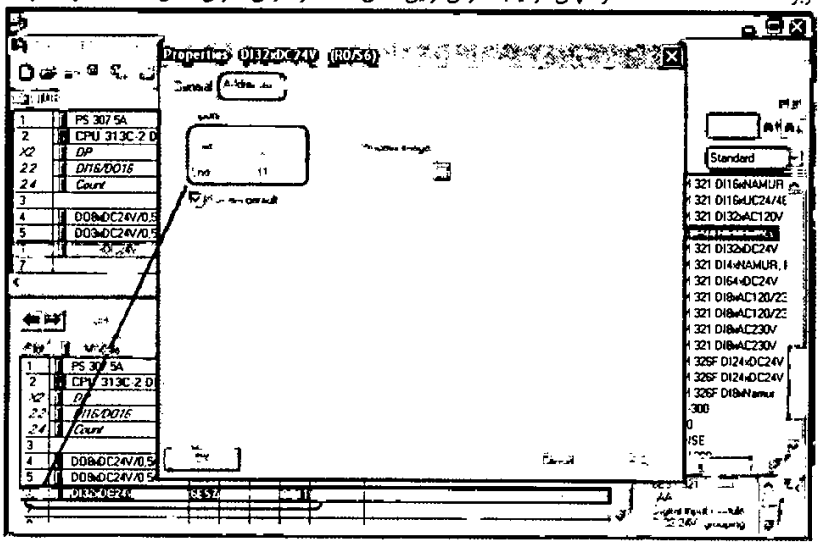


شکل ۱۳-۳۱ مشاهده آدرس‌های هر کانال خروجی دیجیتال توسط پنجره Monitor/Modify

آدرس‌دهی کارت دیجیتال ورودی

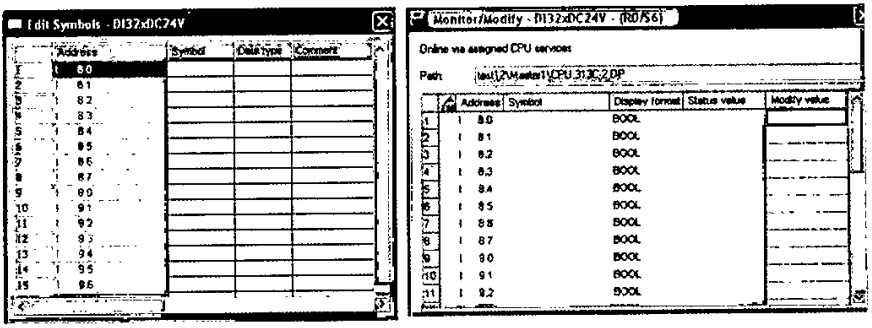
کارت DI نیز به همان صورت که در مورد کارت DO بحث شد آدرس‌دهی می‌شود. تمام نکات و تذکرهای داده شده برای کارت DO در اینجا نیز صادق است.

در شکل ۱۳-۳۲ یک کارت ورودی ۳۲ کاناله دیجیتال نشان داده شده است. همانطور که دیده می‌شود آدرس شروع تا پایان به اندازه ۴ بایت فاصله دارد. این فاصله بایتی را می‌توان در ستون I Address یا با دوبار کلیک نمودن روی کارت در سربرگ Address مشاهده نمود. پس در اینجا آدرس اولین کانال I 8.0 و آدرس آخرین کانال I 11.7 خواهد بود.



شکل ۱۳-۳۲ آدرس کارت DI در Hwconfig

این آدرس‌ها را می‌توان در پنجره Edit Symbols یا Montor /Modify به همان روش ذکر شده قبل مشاهده نمود. به شکل ۱۳-۳۳ توجه کنید.



شکل ۱۳-۳۳ مشاهده آدرس هر کانال دیجیتال ورودی در Hwconfig

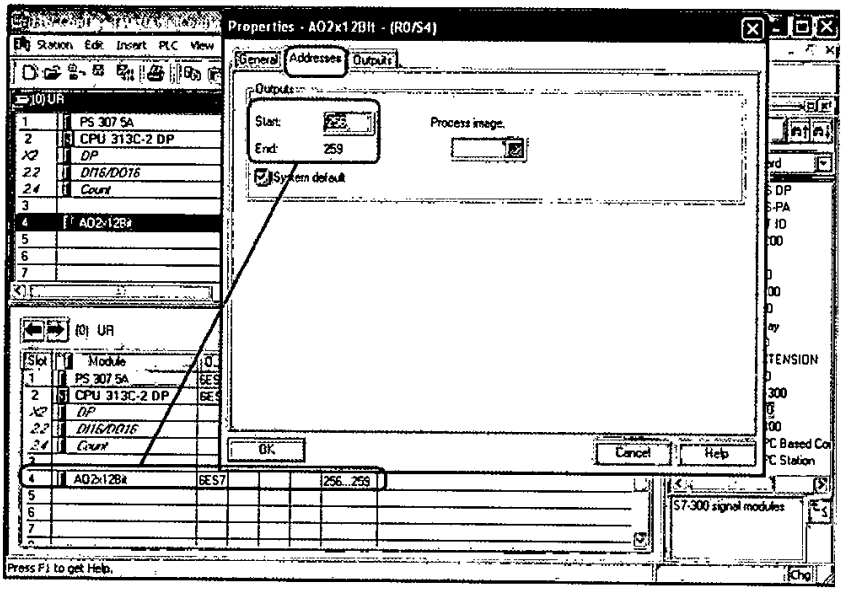
۱۳-۴-۳ آدرس‌دهی کارت‌های آنالوگ

همانطور که بیان شد، دیتاهایی که از کارت‌های آنالوگ دریافت یا به کارت‌های آنالوگ ارسال می‌شود به صورت 16 بیتی می‌باشند. در آدرس‌دهی 16 بیتی، آدرس‌ها به‌صورت word می‌باشند. مثلاً IW 10 به 16 بیت از ناحیه‌ی ورودی‌ها که با آدرس بایت 10 شروع و با بایت 11 خاتمه می‌پذیرد اشاره می‌کند.

در آدرس‌دهی به صورت Word، دو بایت از حافظه جهت قرارگیری داده مصرف می‌شود. بایت اول، بایت هم‌شماره‌ی آدرس و بایت دوم بایت بعدی می‌باشد. مثلاً IW 10 = IB 10 + IB 11 که در آن IB به معنی یک بایت از ورودی‌ها می‌باشد.

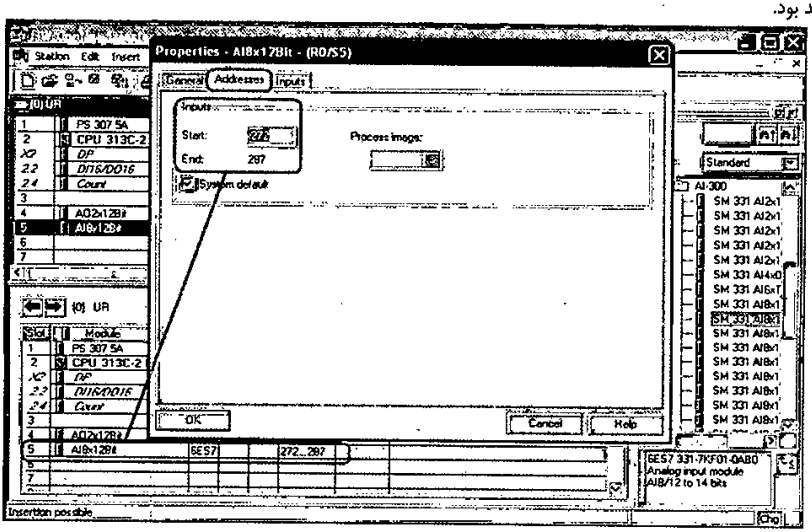
حداقل فاصله‌ی بین هر آدرس در کارت آنالوگ دو بایت می‌باشد. به‌همین روش می‌توان کانال‌های مربوط به کارت‌های آنالوگ خروجی را نیز آدرس‌دهی نمود.

شکل ۱۳-۳۴ یک کارت آنالوگ دو خروجی را در HW Config نشان می‌دهد. همانطور که دیده می‌شود بازه آدرس به اندازه ۴ بایت است (از بایت ۲۵۶ تا ۲۵۹) چون هر کانال ۲ بایت را اشغال می‌کند. بنابراین آدرس کانال اول PQW 256 و آدرس کانال دوم PQW 258 خواهد بود.



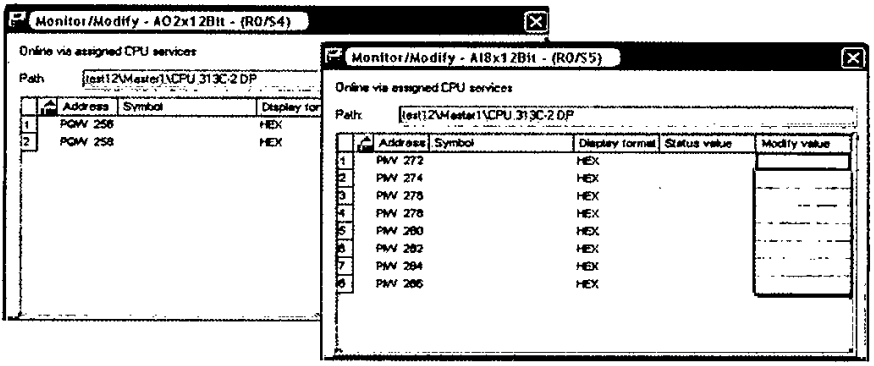
شکل ۱۳-۳۴ آدرس کارت AO در Hwconfig

شکل ۱۳-۳۵ یک کارت آنالوگ ۸ ورودی را نشان می‌دهد. فاصله آدرس ۱۶ بایت معادل word 8 است. آدرس از بایت 272 شروع و به بایت 287 ختم شده است، بنابراین آدرس اولین کانال IW272 و آدرس آخرین کانال IW286 خواهد بود.



شکل ۱۳-۳۵ آدرس کارت AI در Hwconfig

برای مشاهده آدرس هر کانال شبیه روشی که برای کارت‌های دیجیتال ذکر شد می‌توان عمل نمود. شکل ۱۳-۳۶ آدرس کانال‌های کارت‌های ورودی و خروجی آنالوگ فوق را در پنجره Monitor/Modify نشان می‌دهد.



شکل ۱۳-۳۶ مشاهده آدرس هر کانال آنالوگ توسط پنجره Monitor/Modify

فصل ۱۳

نکته‌ای که در شکل ۱۳-۳۶ دیده می‌شود این است که همه آدرس‌ها با حرف P شروع شده‌اند، PIW برای ورودی‌ها و PQW برای خروجی‌ها به‌کار می‌رود. حرف P معرف Peripheral است ولی در برخی حالات ممکن است آنرا در آدرس نبینیم و آدرس‌ها به‌صورت IW و QW ظاهر شوند. این تفاوت فقط به حافظه CPU و تنظیمات آن بستگی دارد و ربطی به خود کارت ندارد. در ادامه این موضوع تشریح خواهد شد.

۱۳-۴-۴ آدرس‌دهی Peripheral

همانطور که قبلاً اشاره شد، وظیفه‌ی کارت‌های ورودی در PLC، تبدیل سیگنال الکتریکی به دیتا (صفر و یک منطقی) می‌باشد. به‌منظور ذخیره‌سازی دیتاهای مربوط به کارت‌های ورودی از حافظه‌ای به نام PII به معنی جدول تصاویر ورودی‌ها استفاده می‌شود. همچنین دیتاهای ارسالی به کارت‌های خروجی در ناحیه‌ای از حافظه‌ی CPU به‌نام PIQ قرار داده می‌شود. محل قرارگیری PII و PIQ در System Memory می‌باشد.

هنگام پیکربندی سخت‌افزار، آدرسی بر حسب بایت به هر کارت اختصاص می‌یابد. تعداد بایت‌ها با توجه به تعداد کانال‌های کارت تعیین می‌گردد. آدرس اختصاص داده شده به یک کارت می‌تواند درون رنج PII و PIQ یا خارج از رنج آن باشد.

در صورتی‌که آدرس ورودی یا خروجی داخل رنج PII/PIQ باشد بدون حرف P ظاهر خواهد شد ولی اگر آدرسی خارج از رنج PII/ PIQ باشد با حرف P که معرف Peripheral است ظاهر می‌شود.

تذکر

- در S7-300 آدرس‌های ورودی و خروجی‌های دیجیتال به‌طور پیش‌فرض درون رنج PII و PIQ می‌باشند، ولی آدرس‌های آنالوگ خارج از رنج هستند.

- در S7-400 آدرس‌های دیجیتال به‌طور پیش‌فرض درون رنج بوده ولی آنالوگ‌ها متفاوت هستند. در اکثر CPUهای S7-400، آدرس آنالوگ‌ها نیز خارج از رنج می‌باشد و فقط در برخی CPUها در رنج می‌باشند.
- در اکثر CPUهای S7-400 می‌توان اندازه‌ی PII و PIQ را افزایش داد. در اینصورت آدرس‌های کارت‌های آنالوگ نیز درون رنج قرار می‌گیرند.
- در S7-300 و هم در S7-400 می‌توان با تغییر آدرس اختصاص یافته به کارت‌های دیجیتال و آنالوگ، آنها را از رنج خارج یا درون رنج وارد نمود تا با حرف P یا بدون حرف P ظاهر شوند.
- در روش آدرس‌دهی Peripheral در ابتدای آدرس مورد نظر حرف P قرار می‌گیرد که نشان‌دهنده‌ی این نوع آدرس‌دهی می‌باشد. در این حالت CPU به نواحی PII و PIQ مرجعه نکرده و جهت خواندان یا نوشتن، مستقیماً به خود کارت مراجعه می‌نماید. این روش آدرس‌دهی معمولاً برای آدرس‌دهی آنالوگ‌ها استفاده می‌شود. مثلاً PIW 760 ، PQW 256 و . . .
- اگر آدرس‌دهی Peripherals در برنامه‌نویسی استفاده شود و دسترسی به کارت توسط CPU موجود نباشد (کارت در محل خود نباشد یا آسیب دیده باشد) ممکن است CPU دچار توقف شود ولی وقتی آدرس‌دهی داخل رنج باشد در شرایط فوق مشکلی برای CPU پیش نمی‌آید.
- در آدرس‌دهی Peripheral امکان آدرس‌دهی بیتی وجود ندارد. آدرس‌ها به‌صورت Byte ، Word و Double Word هستند. انواع آدرس‌دهی Peripheral در جدول ۱۳-۱۵ نشان داده شده‌است.

جدول ۱۳-۱۵ انواع آدرس‌دهی Peripheral

نوع متغیر	نحوه نمایش	مثال
ورودی بایتی	**PIB**	PIB 120
ورودی وردی	**PIW**	PIW 760
ورودی دابل وردی	**PID**	PID 300
خروجی بایتی	**PQB**	PQB 200
خروجی وردی	**PQW**	PQW 752
خروجی دابل وردی	**PQD**	PQD 400

همانطور که دیده می‌شود در این حالت آدرس‌دهی برای Bit وجود ندارد.

۱۳-۴-۵ آدرس‌دهی متغیرهای حافظه

متغیرهای داخلی حافظه که Bit Memory خوانده می‌شوند برای ذخیره‌سازی و استفاده از نتایج میان برنامه که نه ورودی و نه خروجی هستند به‌کار می‌روند. علاوه بر آدرس‌دهی ورودی و خروجی‌ها، می‌توان Bit Memoryها را نیز آدرس‌دهی نمود. آدرس‌دهی Bit Memory مشابه ورودی و خروجی است، با این تفاوت‌که به‌جای I یا Q از حرف M استفاده می‌شود. مثلاً M2.3 به داده‌ای که در ناحیه‌ی Bit Memoryها و در بایت شماره 2، بیت شماره 3 قرار گرفته است اشاره می‌کند. حالات دیگر در جدول ۱۳-۱۶ آمده است.

جدول ۱۳-۱۶ نحوه آدرس دهی Bit Memory

نوع متغیر حافظه	نحوه نمایش	مثال
Bit	M	M 0.1
Byte	MB	MIB 1
Word	MW	MW 2
Dword	MD	MD 8

۱۳-۴-۶ آدرس‌دهی کانتر و تایمر

برای آدرس‌دهی کانتر از حرف C و برای تایمر از حرف T استفاده می‌شود و پس از آن، شماره‌ی کانتر و تایمر ذکر می‌گردد. مثلاً آدرس T3 به تایمر شماره‌ی 3 و آدرس C5 به کانتر شماره‌ی 5 اشاره می‌نماید. نکته‌ای که باید در این آدرس‌دهی به آن توجه نمود این است که شماره‌ی اختصاص یافته به کانتر یا تایمر در رنج کانترها و تایمرهای CPU باشد. مثلاً در CPU 314 C-2DP تعداد کانترها و تایمرها 256 عدد می‌باشد، بنابراین امکان آدرس‌دهی از رنج 0 الی 255 وجود دارد و آدرسی مثل T 300 غیر معتبر می‌باشد.

فصل ۱۳

۱۳-۴-۷ آدرس‌دهی مطلق و سمبلیک

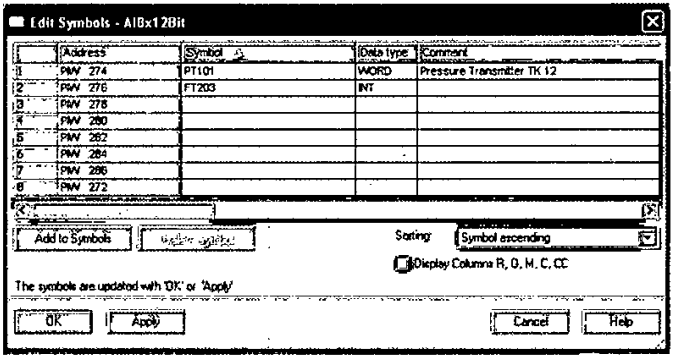
در STEP7 به‌منظور آدرس‌دهی متغیرها دو روش وجود دارد که عبارتند از:

۱- آدرس‌دهی مطلق یا Absolute: در این نوع آدرس‌دهی، آدرس متغیر با آدرس‌های واقعی که در قسمت قبل بیان شد ذکر می‌گردد. مثلاً I 0.0 ،Q 124.0 و ...

۲- آدرس‌دهی سمبلیک Symbolic: در این نوع آدرس‌دهی، به هر آدرس یک مشخصه (نام سمبلیک) اختصاص می‌یابد. مثلاً ممکن است جهت ورودی I 0.0 نام سمبلیک Start در نظر گرفته شود. در اینصورت در برنامه‌ی نوشته شده، در کنار آدرس I 0.0 کلمه‌ی Start نیز درج می‌شود. در پروژه‌های بزرگ که تعداد ورودی‌ها و خروجی‌ها زیاد است، به منظور جلوگیری از اشتباه در آدرس‌ها از آدرس‌دهی سمبلیک استفاده می‌شود. با این کار، برای هر آدرس یک سمبل و توضیح (Comment) در نظر گرفته می‌شود که برنامه‌نویس از روی آن متوجه می‌شود آدرس مورد نظر مربوط به چه قسمتی از فرآیند می‌باشد. در پروژه‌های کوچک نیز امکان این نوع آدرس‌دهی وجود دارد. به‌طور کلی توصیه می‌شود به‌منظور سهولت استفاده از آدرس‌ها این عمل (آدرس‌دهی سمبلیک) حتماً انجام پذیرد.

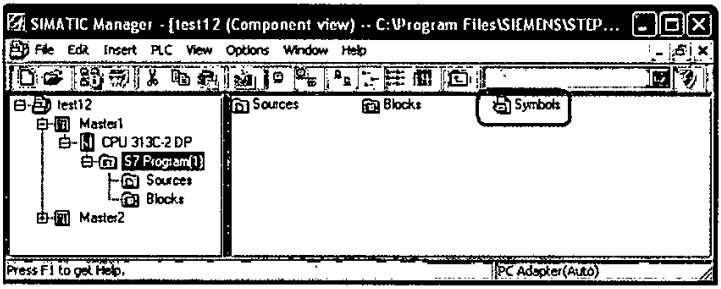
برای آدرس‌دهی سمبلیک از محیط های مختلفی می توان استفاده کرد مانند:

- در محیط HW Config با کلیک راست روی کارت ورودی و خروجی مورد نظر و انتخاب Edit Symbols می‌توان نام مورد نظر را به آن اختصاص داد. به شکل ۱۳-۳۷ توجه کنید.



شکل ۱۳-۳۷ اختصاص سمبل به آدرس‌ها در Hwconfig

- در HW Config به روش فوق فقط می‌توان به ورودی‌ها و خروجی‌ها سمبل اختصاص داد. روش اصلی که توسط آن می‌توان به تمام متغیرهای حافظه از جمله Memory و Timer و Counter نیز سمبل اختصاص داد. ایجاد سمبل‌ها در زیر برنامه‌ای به‌نام Symbol Editor است که در محیط اصلی Simatic Manager قابل دسترس است و در فصل بعد تشریح می‌شود.



شکل ۱۳-۳۸ آیکن اجرای برنامه Symbol Editor

- سمبل‌ها در محیط برنامه‌نویسی LAD/STL/FBD نیز قابل ایجاد و ویرایش هستند. توضیحات در فصل بعد خواهد آمد.

۱۳-۵ نکته پایانی

در بحث مفاهیم پایه برنامه‌نویسی معمولاً تشریح آکومولاتورها و رجیسترهای حافظه CPU S7 نیز صورت می‌گیرد. از آنجا که ما در کتاب مقدماتی به برنامه‌نویسی با زبان LAD/FBD می‌پردازیم، تشریح آکومولاتور و رجیسترها چندان ضرورتی ندارد. در جلدهای بعدی کتاب که به زبان برنامه‌نویسی STL پرداخته می‌شود، این مبحث به‌طور کامل شرح داده خواهد شد.

۱۳-۶ پرسش و تحقیق

در چه PLCهایی استفاده از اعداد بزرگتر از ۳۲ بیت امکان پذیر است؟

فصل ۱۳

۱۳-۷ تست‌های خودآزمایی

۱- کوچکترین واحد حافظه، که می‌توان در آن مقدار صفر یا یک منطقی قرار داد چه نام دارد؟

الف) بیت ب) بایت ج) Word د) Double Word

۲- از قرار گرفتن دو بایت در کنار هم یک . . . . . شکل می‌گیرد.

الف) بیت ب) بایت ج) Word د) Double Word

۳- اطلاعات در سیستم‌های کامپیوتری به چه فرمی مورد استفاده قرار می‌گیرند؟

الف) دسیمال ب) اینتیجر ج) باینری د) هیچ‌کدام

۴- این سیستم عددی دارای ۱۰ عضو می‌باشد و در کار روزمره از آن استفاده می‌شود.

الف) باینری ب) دسیمال ج) BCD د) Hex

۵- معادل دسیمال عدد باینری 1001 برابر است با:

الف) 11 ب) 10 ج) 9 د) 1

۶- معادل باینری عدد 12 دسیمال برابر است با:

الف) 1111 ب) 1101 ج) 1010 د) 1100

۷- معادل BCD عدد 27 دسیمال برابر است با:

الف) 10010111 ب) 01110010 ج) 00100111 د) 00100011

۸- معادل Hex عدد باینری 10011111 برابراست با:

الف) F9 ب) E9 ج) 9E د) 9F

۹- معادل دسیمال عدد هگز FD6A برابراست با:

الف) 16136 ب) 10613 ج) 64 874 د) 67847

۱۰- کدام نوع داده‌های زیر بیانگر داده‌های ابتدایی می‌باشد؟

الف) Elementary Data Types ب) Complex Data Types

ج) Parameter Types د) هر سه مورد

۱۱- کدام نوع داده‌های زیر بیانگر داده‌های مختلط می‌باشد؟

الف) Elementary Data Types ب) Complex Data Types

ج) Parameter Types د) هر سه مورد

۱۲- این نوع داده دارای اندازه‌ی یک بیت بوده و برای ذخیره‌سازی مقادیر مربوط به سیگنال‌های دیجیتال استفاده می‌شود.

الف) Byte ب) Bool ج) Word د) Integer

۱۳- فرمت مقداردهی یک داده‌ی Byte به‌کدام صورت زیر می‌تواند باشد؟

الف) باینری ب) دسیمال ج) Hex د) موارد الف و ج

۱۴- اگر در یک داده‌ی Integer، بیت علامت دارای مقدار صفر منطقی باشد، در مورد عدد قرار گرفته در این داده کدام مورد زیر درست است؟

الف) حتماً مثبت است. ب) صفر است. ج) حتماً منفی است. د) الف و ب

۱۵- در این نوع داده، امکان قرارگیری اعداد اعشاری وجود دارد.

الف) Integer ب) Double Integer ج) Real د) Word

۱۶- این نوع داده به منظور محاسبه‌ی زمان استفاده شده و از شانزده بیت تشکیل شده است.

الف) Time ب) S5Time ج) Date د) Date and Time

۱۷- کدام آدرس به یک خانه از ناحیه ورودی ها که در بیت شماره ۳ از بایت ۶ قرار گرفته اشاره می‌کند؟

الف) Q3.6 ب) I3.6 ج) I6.3 د) Q6.3

۱۸- منظور از آدرس‌دهی Pripheral چیست؟

الف) خارج از رنج PII ب) خارج از رنج PIQ

ج) داخل رنج PII و PIQ د) خارج از رنج PII وPIQ

۱۹- هر آدرس آنالوگ چه میزان فضا اشغال می‌کند؟

الف) یک بایت ب) دو Word ج) دو بایت د) سه بایت

۲۰- هر آدرس دیجیتال چه میزان فضا اشغال می‌کند؟

الف) یک بیت ب) یک بایت ج) یک Word د) دو بیت

فصل ۱۴

شروع برنامه‌نویسی با LAD/STL/FBD

۱۴-۱ مقدمه

۱۴-۲ انواع روش‌های برنامه‌نویسی

۱۴-۳ بلاک‌های برنامه‌نویسی

۱۴-۳-۱ بلاک‌های OB

۱۴-۳-۲ FC (فانکشن)

۱۴-۳-۳ FB (فانکشن‌بلاک)

۱۴-۳-۴ بلاک‌های سیستمی

۱۴-۳-۵ بلاک‌های دیتا Data Block

۱۴-۳-۶ UDT (User Defined Data Type)

۱۴-۴ زبان‌های برنامه‌نویسی

۱۴-۵ ایجاد بلاک‌ها در Simatic Manager

۱۴-۶ محیط برنامه‌نویسی LAD /STL/FBD

۱۴-۶-۱ آشنایی با بخش‌های مختلف LAD/STL/FBD

۱۴-۶-۲ منوها و نوارابزار

۱۴-۷ ایجاد سمبل‌ها

۱۴-۸ تست برنامه با سیمولاتور

۱۴-۹ گروه دستورات برنامه‌نویسی LAD و FBD

۱۴-۱۰ پرسش و تحقیق

۱۴-۱۱ تست‌های خودآزمایی

در این فصل انواع بلاک‌های برنامه و دیتا و نحوه ساخت آنها در محیط LAD/STL/FBD تشریح شده است. مطالعه این فصل پیش نیاز برنامه‌نویسی است.

چکیده مطالب

- بلاک‌های اصلی برنامه‌نویسی OB نام دارند. این بلاک‌ها توسط سیستم عامل فراخوان می‌شوند.
- فانکشن‌ها بایستی در OBها فراخوان شوند. سیستم عامل مستقیماً هیچ فانکشنی را اجرا نمی‌کند.
- در OB1 برنامه‌ای که در سیکل‌اسکن اجرا می‌گردد نوشته می‌شود.
- OB100، OB101 و OB102 بلاک‌هایی هستند که فقط در هنگام راه‌اندازی اجرا می‌شوند.
- OBهای مبتنی بر رخداد فقط در شرایط خاصی که خطایی رخ دهد یا سیگنال تعریف شده‌ای تغییر کنند اجرا می‌شوند.
- برنامه‌نویسی به‌صورت خطی متداول نیست، معمولاً از نوع Partioned یا Structrued استفاده می‌شود.
- FC و FB بلاک‌های برنامه‌نویسی هستند که در برنامه‌نویسی غیر خطی به‌کار می‌رود.
- FB برخلاف FC نیاز به حافظه خاصی دارد که از نوع DB است.
- SFC و SFB بلاک‌های سیستمی هستند که از کتابخانه نرم‌افزار فراخوان می‌شوند.
- DB برای خواندن و نوشتن دیتا به‌کار می‌رود و به دو نوع Shared و Instance تقسیم می‌شود.
- DB نوع Shared می‌تواند برای همه OB ، FB و FCها به‌کار رود.
- DB نوع Instance فقط برای FB و SFB به‌کار می‌رود.
- UDT ابزاری برای تعریف دیتاهای ساختار یافته است.
- زبان‌های برنامه‌نویسی می‌تواند LAD یا FBD یا STL باشد.

فصل ۱۴

اصطلاحات و تعاریف

OB

Organization Block بلاک برنامه‌نویسی است که توسط سیستم عامل فراخوان می‌شود.

FC

Function بلاک برنامه‌نویسی است که بایستی توسط کاربر در برنامه فراخوان شود.

FB

Function Block بلاک برنامه‌نویسی شبیه FC است با این تفاوت که دارای حافظه از جنس DB می‌باشد.

DB

Data Block بلاک دیتاست که در آن دیتاهای تعریف شده ذخیره یا از آن دیتاها خوانده می‌شود.

SFC/SFB

System FC / System FB بلاک‌های سیستمی است که در CPU وجود دارد و برای کاربردهای خاص طراحی شده است. کاربر می‌تواند از آنها در برنامه استفاده کند.

SDB

System DB حاوی دیتاهای سخت‌افزار و شبکه می باشد.

UDT

User Data Type بلاکی است که برای سهولت تعریف دیتا از آن استفاده می شود.

۱-۱۴ مقدمه

سیستم عامل CPU در سیکل‌اسکن به اجرای تکراری برنامه‌ای می‌پردازد که در بلاکی موسوم به OB1 نوشته شده است. OB مخفف Organization Block است. اگرچه در PLC ممکن است بلاک‌های برنامه‌نویسی دیگری نظیر فانکشن‌ها وجود داشته باشند ولی هیچگاه CPU فانکشنی را مستقلاً اجرا نمی‌کند مگر اینکه در بلاک OB صدا زده شده باشد، به این ترتیب سیستم عامل فقط با OB سرو کار دارد نه هیچ بلاک دیگری. OBها انواع مختلفی دارند و شماره‌های خاصی برای آنها مشخص شده است. OB با هر شماره‌ای را نمی‌توان به CPU دانلود کرد. به‌عنوان مثال برخی شماره‌ها در تمام CPUها غیر مجاز هستند مانند OB2 ولی برخی از OBها در یک CPU مجاز و در دیگری غیر مجاز هستند که این موضوع به قابلیت‌های CPU بستگی دارد مثلاً OB32 فقط در برخی CPUها قابل استفاده است.

به‌طور کلی بلاک‌های قابل استفاده برای S7-300 و S7-400 را می‌توان به سه گروه اصلی تقسیم نمود:

۱-بلاک‌های برنامه‌نویسی (OB, FC , FB)

۲-بلاک‌های سیستمی (SFC , SFB , SDB)

۳-بلاک‌های ذخیره‌ی اطلاعات (DB , UDT)

توضیح این بلاک‌ها و تفاوت بین آنها در ادامه خواهد آمد. در این کتاب ما فقط به برنامه‌نویسی با OB1 می‌پردازیم. سایر OBها و همچنین فانکشن‌های برنامه‌نویسی در کتاب‌های سطح پیشرفته و تکمیلی تشریح شده‌اند.

فصل ۱۴

۲-۱۴ انواع روش‌های برنامه‌نویسی

به‌طور کلی انواع برنامه‌نویسی در Step7 را می‌توان به سه نوع خطی، تقسیم شده و ساختار یافته تقسیم نمود:

الف. برنامه‌نویسی خطی[1]

در این نوع برنامه‌نویسی، همه‌ی برنامه در OB1 به‌صورت خطبه‌خط نوشته می‌شود. عیب‌یابی برنامه‌هایی که به روش خطی نوشته شوند مشکل و زمان‌بر است. این روش برنامه‌نویسی مخصوص پروسه‌های کوچک است که در آنها تعداد دستورات برنامه کم می‌باشد.

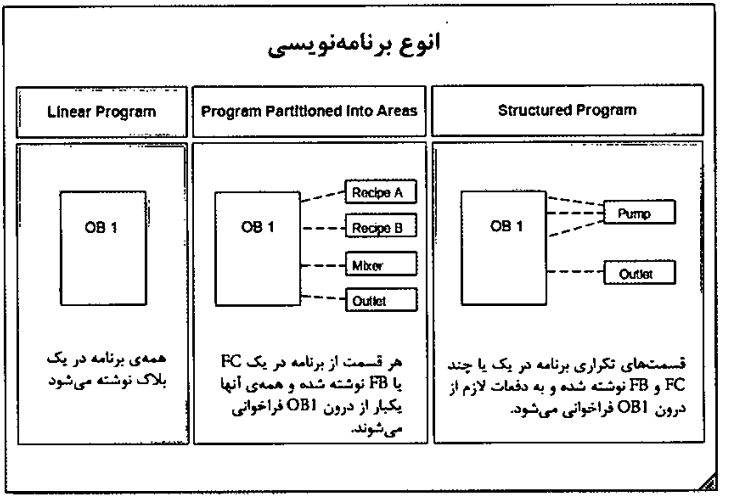
ب. برنامه‌نویسی تقسیم‌شده[2]

در این نوع برنامه‌نویسی، برنامه به قسمت‌های مجزایی تقسیم می‌شود. هر قسمت از برنامه در یک FC یا FB نوشته شده و در هر سیکل کاری، یک‌بار از درون OB1 فراخوانی می‌شود. در پروسه‌هایی دارای چند بخش مجزا هستند، می‌توان برنامه‌ی هر بخش را در یک FC یا FB نوشته و آنرا به روش گفته شده در OB1 فرخوانی نمود.

1. Linear Programming
2. Partitioned Program

ج. برنامه‌نویسی ساختاریافته[1]

در این نوع برنامه‌نویسی، قسمت‌هایی که باید چندین بار تکرار شوند را در یک FC یا FB نوشته و آن FC یا FB را در OB1 به تعداد دفعات لازم فراخوانی می‌نمایند. کاربرد این نوع برنامه‌نویسی زمانی است‌که نیاز باشد یک قسمت از برنامه چندین بار اجرا شود. در واقع با این عمل از نوشتن قسمت‌های تکراری برنامه جلوگیری می‌شود.

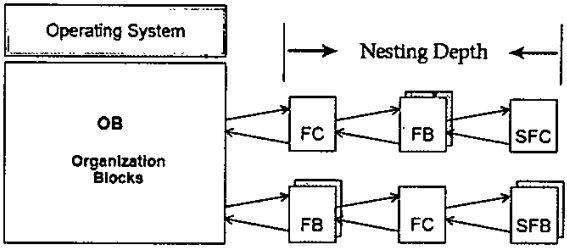


شکل ۱۴-۱ انواع روش‌های برنامه نویسی

در برنامه‌نویسی‌های غیر خطی که بلاک‌ها یکدیگر را فراخوان می‌کنند توجه به نکات زیر الزامی‌است.

- برای فراخوانی بلاک‌ها از درون یکدیگر حدی وجود دارد که به آن عمقِ تودرتویی گفته می‌شود که از مشخصات CPU است. هر CPU دارای عمقِ تودرتویی[2] مشخصی است‌که در کاتالوگ آن ذکر می‌گردد. مثلاً برای CPU 314C-2DP این مشخصه برابر ۸ می‌باشد، یعنی حداکثر ۸ بلاک را می‌توان از درون یکدیگر فراخوانی نمود. منظور این است که بلاک ۸ از درون بلاک ۷، بلاک ۷ از درون بلاک ۶، بلاک ۶ از درون بلاک ۵ و به همین‌ترتیب سایر بلاک‌ها فراخوانی شوند، در غیر این‌صورت از درون یک بلاک می‌توان بلاک‌های مختلفی را فراخوانی نمود.

1. Structure Programming
2. Nesting Depth



شکل ۱۴-۲ عمق تودرتویی

- در اکثر CPUهای S7-400 عمق تودرتویی برابر 24 است، یعنی 24 بلاک تودرتو می‌توان ایجاد نمود.
- هنگامی‌که از روش فوق استفاده شود، همه‌ی بلاک‌ها باید به CPU دانلود شوند، در غیر اینصورت CPU متوقف شده و چراغ SF روشن می‌شود.

فصل ۱۴

۱۴-۳ بلاک‌های برنامه‌نویسی

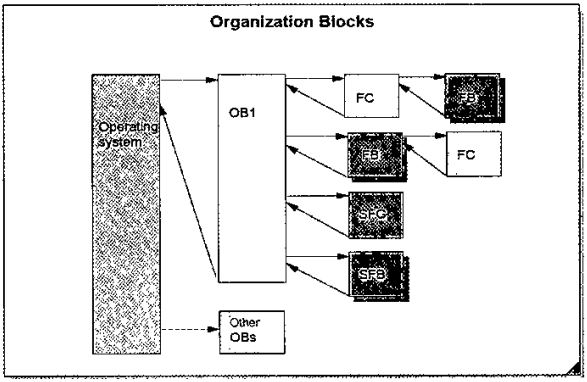
این بلاک‌ها که به Code Block معروف هستند به منظور برنامه‌نویسی استفاده می‌شوند. انواع این بلاک‌ها عبارتست از:

- OB (Organization Block)
- FC (Function)
- FB (Function Block)

۱۴-۳-۱ بلاک‌های OB

این بلاک‌ها رابط بین سیستم عامل[1] و برنامه‌ی کاربر[2] محسوب می‌شوند، OBها انواع مختلف دارند و هر کدام به‌منظور انجام مقاصد خاصی طراحی شده‌است. OB1 بلاکی است که برنامه‌ی اصلی کاربر در آن نوشته می‌شود و توسط سیستم عامل به‌طور سیکلی اجرا می‌شود. در واقع آنچه در بحث سیکل‌اسکن ذکر شد که CPU در سیکل اسکن به اجرای برنامه می‌پردازد، منظور از اجرای برنامه، اجرای OB1 می‌باشد. در صورتی‌که در برنامه از FC ، FB ، SFC یا SFB استفاده شود باید آنها را در OB1 فراخوانی نمود، در غیر اینصورت این بلاک‌ها اجرا نخواهند شد.

1. Operating System
2. User Program



شکل ۱۴-۳ فراخوانی بلاک‌ها در OB1

هر OB مستقلاً توسط سیستم عامل CPU فراخوان می‌شود و نمی‌توان از یک OB یک OB دیگر را شبیه فانکشن صدا زد. سیستم عامل OB1 را همیشه اجرا می‌کند ولی سایر OBها تا شرایط آنها محقق نشده اجرا نخواهند شد. به طور کلی می‌توان OBها را به چهار گروه زیر تقسیم نمود:

- OBهای راه‌اندازی
- OB سیکل آزاد
- OB های پریودیک
- OBهای مبتنی بر رخداد
-

الف. OB های راه‌اندازی

این OBها صرفاً در زمان راه‌اندازی CPU و با توجه به نوع راه‌اندازی که Warm ، Cold یا Hot باشد اجرا می‌شوند. بنابراین اگر در طول کار نرمال که CPU مشغول اجرای سیکل اسکن است این OBها به PLC دانلود شوند اجرا نخواهند شد بلکه وقتی CPU از مد Stop به RUN می‌رود و چراغ RUN در حال چشمک‌زدن است به اجرای آنها می‌پردازد. با خروج از مد Startup و ورود به مد RUN این OBها اجرا نخواهند شد. OBهای راه اندازی در جلد دوم کتاب تشریح شده‌اند.

ب. OB سیکل آزاد

منظور همان OB1 است که به Free Cycle نیز معروف می‌باشد.

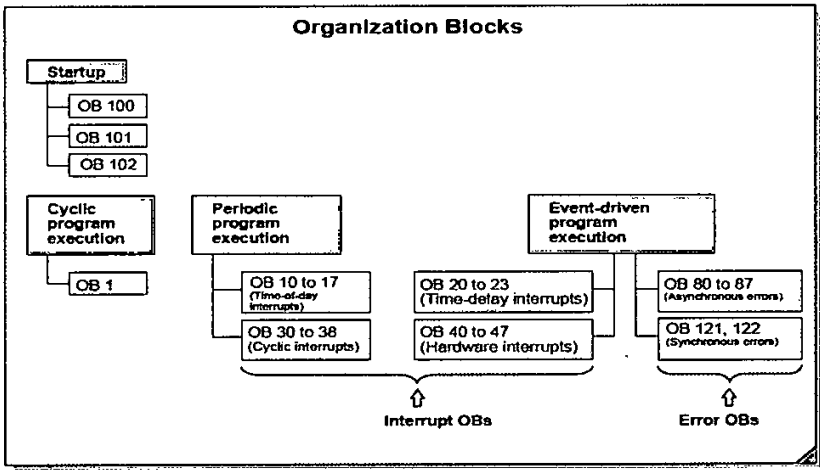
ج. OBهای پریودیک

این OBها به‌طور پریودیک و طبق زمان‌بندی مشخصی توسط CPU اجرا می‌شوند. این OBها از خانواده وقفه هستند و در کتاب سطح پیشرفته تشریح می‌شوند.

شروع اجرای آنها را می‌توان به‌صورت دستی یا با استفاده از SFCها تنظیم نمود.

د. OBهای مبتنی بر رخداد

این OBها با توجه به وقوع یک رخداد مشخص فراخوانی و اجرا می‌شوند. رخداد می‌تواند خطاهای پیش‌آمده در ضمن کار CPU یا یک رخداد دیگر باشد. این OBها نیز از خانواده وقفه هستند و در کتاب سطح پیشرفته تشریح می‌شوند.
شکل ۱۴-۴ انواع OBهای موجود و عملکرد آنها را نشان می‌دهد.



شکل ۱۴-۴ دسته‌بندی انواع OBها

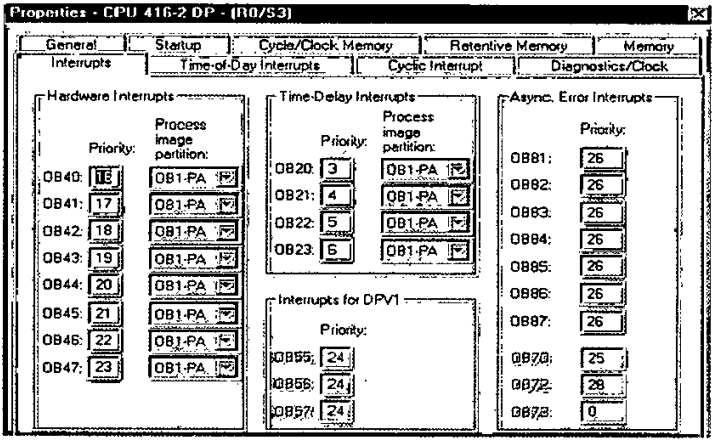
فصل ۱۴

هر OB دارای یک کلاس اولویت می‌باشد. منظور از کلاس اولویت، حق تقدم در اجراست. در زمانی که چند OB همزمان فراخوانی شوند، بلاکی که کلاس اولویت آن بیشتر باشد اجرا می‌شود. در بین OBها، OB1 دارای پایین‌ترین کلاس اولویت است و اجرای آن توسط سایر OBهای دیگر می‌تواند قطع شود. جدول ۱۴-۱ انواع OBها و کلاس اولویت آنها را نشان می‌دهد.

جدول ۱۴-۱ اولویت انواع OB ها

درجه اولویت	نوع	OB نام
1	Main program scan	OB1
2	Time-of-day interrupts	OB10 to OB17
3 to 6	Time-delay interrupts OB20	OB20 to OB23
7 to 15	Cyclic interrupts	OB30 to OB38
16 to 23	Hardware interrupts	OB40 to OB47
25	Multicomputing interrupt	OB60
25 28	Redundancy errors	OB70, OB72
26	Asynchronous errors	OB80, OB82 to OB87
29	Background cycle	OB90
27	Startup	OB100 to OB102
	Synchronous errors	OB121, OB122

در S7-300 (به‌غیر از CPU 318-2) کلاس اولویت OBها را نمی‌توان تغییر داد، اما در S7-400 امکان تغییر کلاس اولویت وجود دارد. تغییر اولویت در پارامترهای CPU امکان‌پذیر است که با دوبارکلیک نمودن روی آن در محیط HW Config ظاهر می‌شود. شکل۱۴-۵ تنظیم کلاس اولویت OBها را در S7-400 نشان می‌دهد.



شکل۱۴-۵ تغییر کلاس اولویت OB ها

۱۴-۳-۲ FC (فانکشن)

Function که به اختصار با FC در برنامه معرفی می‌شود، بلاکی است که حاوی دستورات برنامه‌نویسی است و از آن در برنامه‌نویسی تقسیم شده Prtioned یا برنامه‌نویسی ساختار یافته Structured استفاده می‌شود.

معمولاً FCای که در برنامه‌نویسی تقسیم شده به‌کار می‌رود فقط حاوی برنامه‌ی کاربر می‌باشد. این FC نیاز به ورودی خاصی ندارد و هیچ خروجی خاصی را نیز بر نمی‌گرداند و هدف از به‌کارگیری آن فقط تقسیم کردن برنامه به بخش‌های مجزا می‌باشد. اما FCای که در برنامه‌نویسی ساختار یافته استفاده می‌شود دارای پارامترهای ورودی و خروجی است که در هر فراخوانی آدرس اختصاص یافته به آن تغییر می‌کند. مثلاً ممکن است چهار سیستم وجود داشته باشد که منطق کنترل آنها یکی باشد، دراینصورت می‌توان برنامه‌ی کنترل را در یک FC نوشته و در هربار که این FC فراخوانی می‌شود، آدرس‌های آنرا عوض نمود.

برای اجرای یک FC باید آنرا در بلاک ماقبل فراخوانی نمود. منظور از بلاک ماقبل، OB1 یا بلاکی دیگری مانند FC یا FB است که به صورت مستقیم یا با واسطه در OB1 فراخوانی شده است. بحث کامل FC در کتاب سطح پیشرفته آمده است.

۱۴-۳-۳ FB (فانکشن‌بلاک)

همه‌ی موارد ذکر شده در مورد FC در مورد FB نیز صادق است. تنها تفاوت FC و FB در این است‌که FB دارای یک حافظه به منظور ذخیره‌سازی اطلاعات می‌باشد. این حافظه یک دیتابلاک (DB) است. هنگامی‌که یک FB فراخوانی

می‌شود، باید دیتابلاک مربوط به آن را نیز فراخوانی نمود. FBها بیشتر در برنامه‌نویسی ساختار یافته به‌کار می‌روند و دارای ورودی و خروجی هستند. تمام ورودی و خروجی‌های آنها در دیتا بلاک ذخیره می‌شوند، بعلاوه می‌توان برای آنها مقدار اولیه پیش‌فرض تعریف نمود.

۱۴-۳-۴ بلاک‌های سیستمی

این بلاک‌ها به سه دسته SFC و SFB و SDB تقسیم می‌شوند:

SFC: (System Function) مانند FC فاقد حافظه می‌باشد.

SFB: (System Function Block) مانند FB دارای حافظه از نوع DB است.

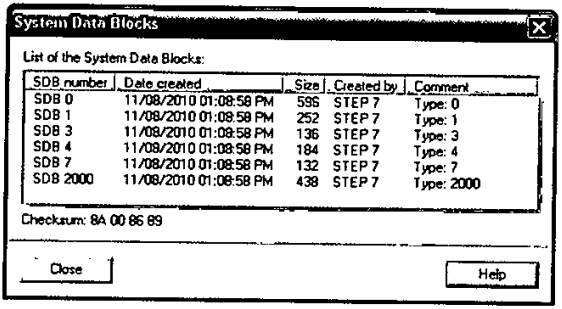
این بلاک‌ها حاوی برنامه‌های از قبل نوشته شده هستند و قفل می‌باشند و نمی‌توان برنامه داخل آنها را مشاهده کرد ولی می‌توان آنها را صدا زده و از قابلیتشان استفاده نمود. هر CPU دارای یک سری SFC/SFB است که در ROM داخلی آن ذخیره شده و قابل پاک شدن نیست. برای صدا زدن این بلاک‌ها بایستی از کتابخانه نرم افزار استفاده نمود.هر کدام از SFCها و SFBها وظیفه‌ی خاصی برعهده دارد، مثلاً SFC 46 منجر به توقف CPU می‌شود. نکات بیشتر مربوط به این بلاک‌ها در کتاب سطح تکمیلی ارائه می‌شود.

SDB: (System Data Block)

این بلاک‌ها به منظور ذخیره‌سازی تنظیمات سیستم و اطلاعات ماژول‌ها، مثل اطلاعات مربوط به پیکربندی سخت‌افزار، شبکه و . . . می‌باشند.

این بلاک‌ها پس از کامپایل HW Config به‌طور اتوماتیک توسط سیستم تولید شده و با دانلود به CPU منتقل می‌شوند و امکان ویرایش آنها، توسط کاربر وجود ندارد.

محل قرارگیری آنها در پوشه‌ی System Data در محیط نرم‌افزار Simatic Manager می‌باشد. در شکل ۱۴-۶ این بلاک‌ها را مشاهده می‌نمایید.



شکل۱۴-۶ SDBها

هر SDB دارای یک شماره‌ی مخصوص است و تنظیمات خاصی را در خود ذخیره می‌نماید. جدول ۱۴-۲ انواع SDB و دلیل ایجاد آنها را نشان می‌دهد.

فصل ۱۴

جدول ۱۴-۲ انواع SDB

شماره SDB	ایجاد شده توسط . . .
0	Configuring Hardware
1	Configuring Hardware or by the CPU (after a complete restart)
2	CPU (standard parameter assignment after a complete restart)
3, 4, and 7	Configuring Hardware
5	CPU (MPI parameters)
22 to 89	Configuring Hardware (DP configuration)
90 to 99	Configuring Hardware (H and F systems)
100 to 149	Configuring Hardware (parameters for central and distributed configurations)
150 to 152	Configuring Hardware (parameters for interface modules)
153 to 189	Configuring Hardware (DP configuration)
200	Configuring Connections (S7-300)
210	Configuring Global Data Communication
3xx	Configuring Symbol-Related Messages
7xx	Configuring Connections
999	Configuring Networks/Connections (S7-Routing)
≥ 1000	Configuring Hardware (DP configuration, parameters for CPs and FMs, S7-Routing, Connections)

۱۴-۳-۵ بلاک‌های دیتا Data Block

این بلاک‌ها همانطور که از نامشان پیداست، مربوط به دیتا هستند و در آنها کد برنامه‌نویسی به‌کار نمی‌رود. از این بلاک‌ها می‌توان به‌منظور ذخیره‌سازی اطلاعات مربوط به داده‌های برنامه‌نویسی یا ارتباط داده‌های بین PLC و HMI استفاده کرد. DB در کتاب سطح پیشرفته تشریح شده و در اینجا به اجمال معرفی می‌گردد. دیتا بلاک دارای دو نوع است:

- Shared DB
- Instance DB

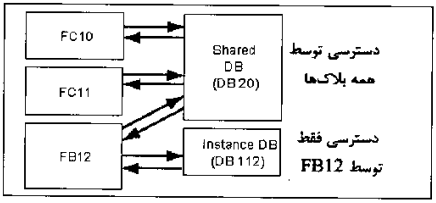
الف. نوع Shared (اشتراکی)

این نوع دیتا بلاک به‌صورت اشتراکی بوده و همه‌ی بلاک‌های برنامه‌نویسی می‌توانند از آن استفاده نمایند. ساختار این DB توسط کاربر می‌تواند تغییر داده شود.

ب. نوع Instance (اختصاصی)

این نوع دیتابلاک مخصوص FB یا SFB ها می‌باشد و به‌عنوان حافظه‌ای برای آنها در نظر گرفته می‌شود. در واقع هر FB یا SFB دارای یک دیتابلاک اختصاصی می‌باشد که برخی از متغیرها و پارامترهای آن در این دیتابلاک ذخیره می‌شود. ساختار این DB وابسته به پارامترهای FB/SFB است و به‌طور خودکار ایجاد می‌شود و کاربر نمی‌تواند این ساختار را تغییر دهد.

در شکل ۱۴-۷ نحوه دسترسی به دیتابلاک اشتراکی و اختصاصی نشان داده شده است.



شکل ۱۴-۷ دیتابلاک اشتراکی و اختصاصی

۱۴-۳-۶ UDT (User Defined Data Type)

UDT ابزاری است که اگر چه به‌صورت یک بلاک ظاهر می‌شود، ولی ابزاری است که توسط آن می‌توان دیتاها را ساختاربندی کرد. عملکرد UDT مشابه نوع داده Struct است که در فصل قبل به آن اشاره شد.

به‌عنوان مثال در یک DB می‌توان یک دیتا از نوع Struct با ساختار مشخص تعریف کرد ولی این فقط برای همان DB کاربرد دارد. در حالی‌که UDT در بیرون از بلاک DB ساختارش تعیین می‌شود و می‌توان این ساختار را در تمام DBها به‌کار برد و منحصر به DB خاصی نیست. استفاده از UDT در پروژه‌های بزرگ منجر به کاهش زمان ایجاد داده‌ها در دیتابلاک می‌شود.

پس UDT بلاکی است که برای تسهیل در ساختار دیتا از آن استفاده می‌شود. این بلاک به PLC دانلود نمی‌شود و مشابه سمبل‌ها در پروژه ذخیره شده روی هارد کامپیوتر باقی می‌ماند.

UDT در کتاب سطح پیشرفته تشریح شده است.

فصل ۱۴

۱۴-۴ زبان‌های برنامه‌نویسی

در STEP7 زبان‌های مختلفی به‌منظور انجام برنامه‌نویسی به‌کار می‌رود. اکثر این زبان‌ها با زبان‌های مندرج در استاندارد IEC 1131 مطابقت دارند. در این بین سه زبان LAD، FBD و STL به‌عنوان زبان‌های پایه محسوب شده و در همه‌ی بلاک‌های برنامه‌نویسی امکان برنامه‌نویسی به این سه‌زبان وجود دارد. علاوه بر این زبان‌ها، زبان‌های دیگری مانند SCL و S7-GRAPH نیز وجود دارد که کاربرد آنها خاص است.

زبان LAD

این زبان برنامه‌نویسی که به Ladder یا نردبانی مشهور است، یکی از زبان‌های اصلی و پرکاربرد در برنامه‌نویسی PLCهای S7 می‌باشد. این زبان شبیه طراحی نقشه‌های مدارفرمان در سیستم‌های رله‌کنتاکتوری بوده و با زبان LD موجود در استاندارد IEC 1131 تطابق دارد. در این زبان برنامه‌نویسی منطق کنترل توسط عناصری مانند تیغه‌های باز (Normally Open : NO) و تیغه‌ی بسته (Normally Close :NC) و بوبین (Coil) و ... پیاده‌سازی می‌شود. اکثر دستورات موجود در این زبان به‌صورت گرافیکی می‌باشند.

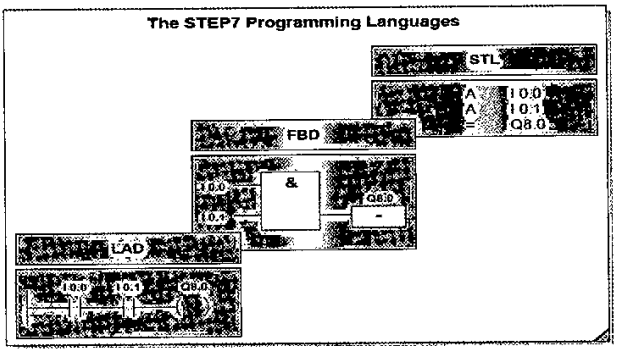
زبان FBD (Function Block Diagram)

این زبان نیز یکی از زبان‌های اصلی برنامه‌نویسی در Step7 محسوب شده و با زبان FBD در استاندارد IEC 1131 تطابق دارد. در این زبان برنامه‌نویسی، منطق کنترل توسط بلوک‌دیاگرام‌های آماده که حاوی گیت‌های منطقی مانند AND، OR، XOR، NOT و . . . می‌باشند پیاده‌سازی می‌شود. اکثر دستورات موجود در این زبان به‌صورت گرافیکی می‌باشند.

زبان STL (Statement list)

این زبان نیز از زبان‌های اصلی برنامه‌نویسی در Step7 محسوب می‌شود. دستورات موجود در این زبان به صورت غیر گرافیکی بوده و کاربر باید هنگام برنامه‌نویسی آنها را تایپ نماید. نمونه دستورات منطقی موجود در آن دستورات AND، OR، XOR و . . . می‌باشند.

دستورات زبان STL نسبت به LAD و FBD بیشتر است و زبان کاملتری نسبت به آنها محسوب می‌شود. این زبان سطح پایین است و برنامه‌نویسی و عیب‌یابی با آن دشوارتر از زبان‌های دیگر است. در شکل ۱۴-۸ برنامه‌ی نوشته شده به زبان‌های LAD ، FBD و STL نشان داده شده است.



شکل ۱۴-۸ انواع زبان‌های برنامه‌نویسی پایه (LAD/STL/FBD)

روش S7-GRAPH

علاوه بر زبان‌های LAD، FBD و STL که در همه‌ی بلاک‌های برنامه‌نویسی قابل استفاده می‌باشند، روش برنامه‌نویسی دیگری به‌نام GRAPH نیز وجود دارد که فقط در FB می‌توان از آن استفاده نمود. همانطور که قبلاً نیز اشاره شد، این زبان به منظور برنامه‌نویسی پروسه‌های ترتیبی استفاده می‌شود و کاملاً گرافیکی است. علاقه‌مندان می‌توانند برای توضیحات بیشتر به کتاب برنامه نویسیPLC به زبان S7-Graph تالیف احمد فرجی – علیرضا سینا مراجعه نمایند.

زبان SCL

این زبان سطح بالا و شبیه زبان پاسکال می‌باشد و کاربرد آن در نوشتن برنامه‌هایی است که منطق کنترلی پیچیده ریاضی دارند. در این روش نمی‌توان از برنامه LAD/STL/FBD استفاده کرد بلکه بایستی برنامه‌نویسی درفایل‌های Source

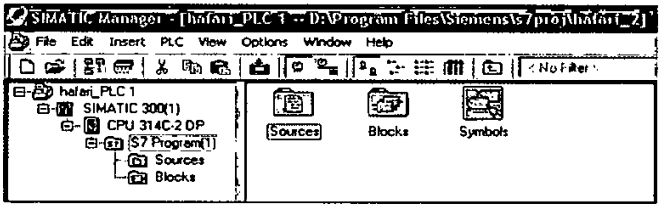
انجام شود، یک فایل Source را می‌توان با کامپایل به یک بلاک تبدیل نمود. این زبان در کتاب جداگانه‌ای تشریح می‌گردد.

۱۴-۵ ایجاد بلاک‌ها در Simatic Manager

همانطور که در فصل ۱۰ ذکر شد، پس از ایجاد ایستگاه‌کاری و انجام پیکربندی سخت‌افزار، توسط عمل Save and Compile فایل‌های سیستمی ایجاد می‌شوند. علاوه بر فایل‌های سیستمی، پوشه‌ی S7-Program که حاوی پوشه‌های Symbols ، Blocks و Sources می‌باشد در زیر CPU نیز ایجاد می‌شود.

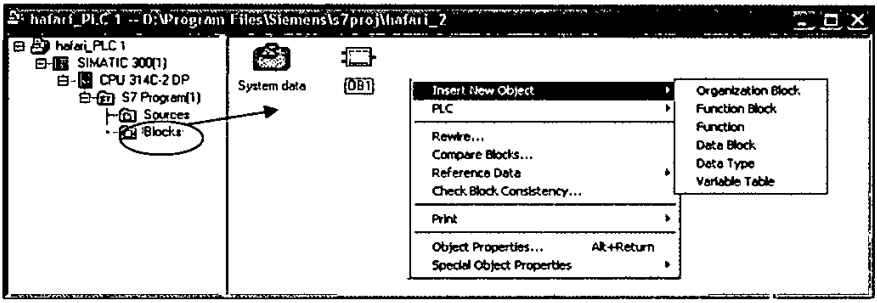
با دوبار کلیک نمودن روی گزینه‌ی Symbols، می‌توان وارد محیط برنامه‌ی Symbol Editor جهت ایجاد سمبل‌ها شد.

در قسمت Blocks، بلاک OB1 به‌طور پیش‌فرض موجود می‌باشد البته می‌توان سایر بلاک‌ها را نیز به صورت دستی ایجاد نمود. در قسمت Sources، فایل‌هایی که به‌روش Source ایجاد شده‌اند موجود می‌باشند. برای رسیدن به پوشه‌ی S7 Program می‌توان از ساختار درختی نشان داده شده در شکل ۱۴-۹ مسیر نشان داده شده را انتخاب نمود.



شکل ۱۴-۹ مسیر بلاک‌ها در Simatic Manager

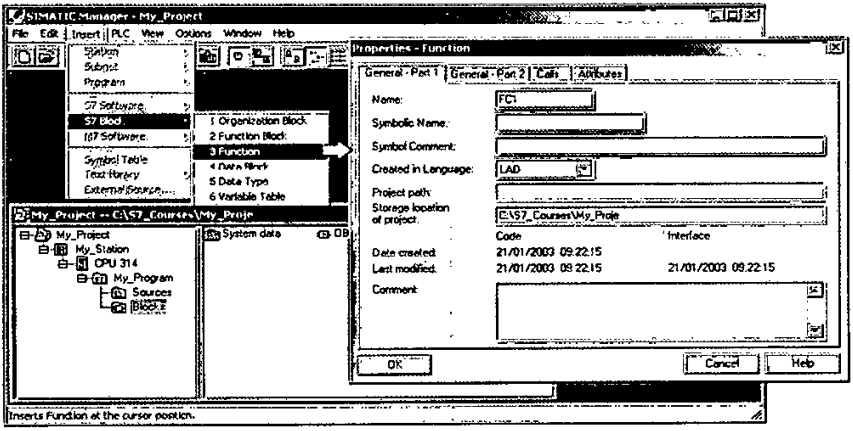
جهت ایجاد بلاک‌ها در نرم‌افزار Simatic Manager، راه‌های مختلفی وجود دارد. آسان‌ترین روش این‌است‌که در پوشه‌ی Blocks که در شکل ۱۴-۹ نشان داده شده است، در فضای خالی کلیک راست نموده و مطابق شکل ۱۴-۱۰ عمل نمود. از کادر باز شده می‌توان بلاک مورد نظر را انتخاب نمود.



شکل ۱۴-۱۰ ایجاد بلاک‌های جدید در Simatic Manager

علاوه بر روش فوق، می‌توان بر روی پوشه‌ی Blocks کلیک نموده و مطابق شکل ۱۴-۱۱ عمل نمود.

فصل ۱۴

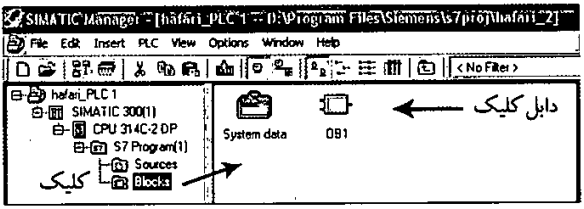


شکل ۱۴-۱۱ ایجاد بلاک‌های جدید

۱۴-۶ محیط برنامه‌نویسی LAD /STL/FBD

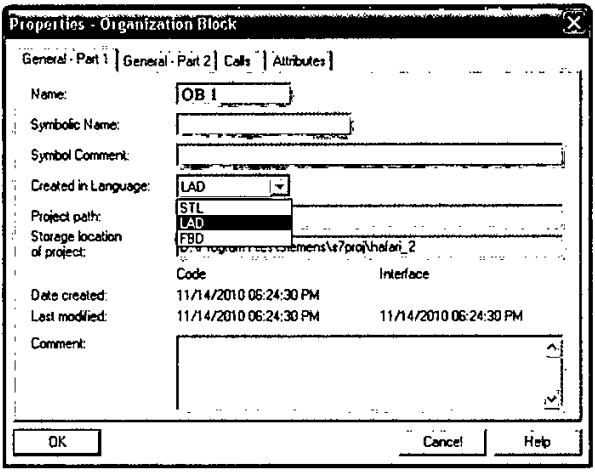
۱۴-۶-۱ آشنایی با بخش‌های مختلف LAD/STL/FBD

برنامه‌نویسی به زبان‌های LAD، FBD و STL در محیط نرم‌افزار LAD/STL/FBD انجام می‌پذیرد. یکی از ساده‌ترین روش‌های اجرای این محیط، رفتن به پوشه‌ی Blocks و دوبار کلیک نمودن برروی OB1، مطابق‌شکل ۱۴-۱۲ می‌باشد.



شکل ۱۴-۱۲ اجرای محیط برنامه‌نویسی از طریق Simatic Manager

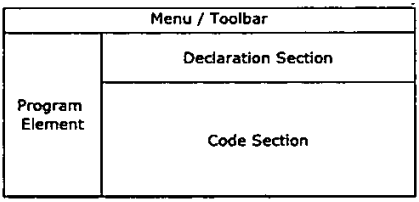
در اینصورت کادری مانند شکل ۱۴-۱۳ باز شده که در آن می‌توان زبان برنامه‌نویسی دلخواه را انتخاب نمود. همانطور که ملاحظه می‌شود در OB فقط امکان انتخاب یکی از سه زبان پایه وجود دارد.



شکل ۱۴-۱۳ انتخاب زبان برنامه‌نویسی بلاک

در پنجره‌ی باز شده علاوه بر انتخاب زبان برنامه‌نویسی، می‌توان یک نام سمبلیک نیز به بلاک اختصاص داد. پس از انجام تنظیمات لازم، می‌توان روی گزینه‌ی OK کلیک نمود. در اینصورت برنامه‌ی LAD/STL/FBD اجرا شده و محیطی شبیه شکل ۱۴-۱۴ نمایان می‌شود.

فصل ۱۴



شکل ۱۴-۱۴ بخش‌های مختلف محیط برنامه‌نویسی

همانطور که در شکل ۱۴-۱۴ مشاهده می‌شود، محیط نرم افزار LAD/STL/FBD از چهار قسمت اصلی تشکیل شده است:

- Menu / Toolbar
- Declaration Section
- Code Section
- Program Element

در ادامه به بررسی هر یک از این قسمت‌ها می‌پردازیم.

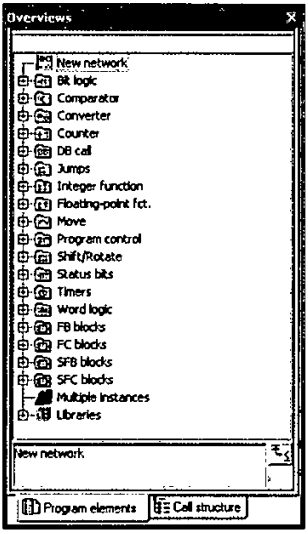
Program Element

در این قسمت المان‌های برنامه‌نویسی قرار گرفته‌اند. در زبان‌های LAD و FBD، برنامه‌نویسی توسط این المان‌ها انجام می‌شود ولی در زبان STL باید دستورات را تایپ نمود.

در صورت غیرفعال بودن این قسمت، می‌توان از مسیر Insert > Program Element آنرا فعال نمود.

همچنین در این بخش، Libraries (کتابخانه) نرم‌افزار نیز قرار گرفته است که در آن بلاک‌های سیستمی و برخی از بلاک‌های دیگر قرار دارند. کاربر می‌تواند بر حسب نیاز، بلاک‌های موجود در کتابخانه را در برنامه‌ی خویش استفاده نماید.

برخی از المان‌های پر کاربرد برنامه‌نویسی در نوارابزار موجود هستند و نیازی به مراجعه به Program Element نیست.



شکل۱۴-۱۵ پنجره‌ی Program Element

Declaration Section

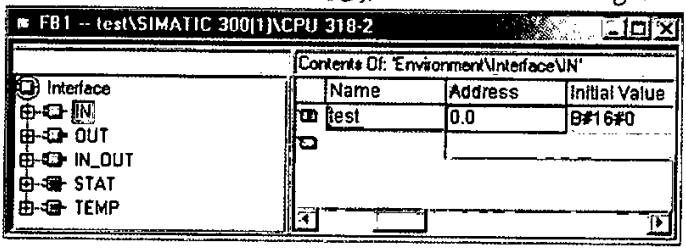
در این قسمت می‌توان متغیرهای محلی و پارامترهای صوری را تعریف نمود. البته در OBها امکان ایجاد پارامترهای صوری وجود ندارد ولی در FC و FB می‌توان آنها را ایجاد نمود.

جدول ۱۴-۳ نشان می‌دهد که کدامیک از متغیرها در چه نوع بلاکی وجود دارد. علامت X به‌معنی وجود داشتن و علامت – به معنی وجود نداشتن متغیر مورد نظر می‌باشد.

جدول ۱۴-۳

Declaration	FC	FB	OB
IN	X	X	--
OUT	X	X	--
IN_OUT	X	X	--
TEMP	X	X	X
STATIC	--	X	--

شکل ۱۴-۱۶نشان‌دهنده Declaration Section برای یک FB است.

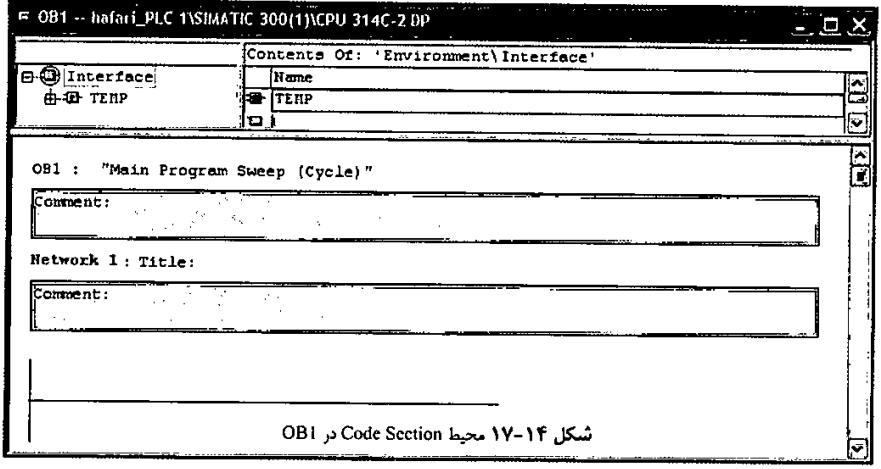


شکل ۱۴-۱۶ Declaration Section در یک FB

فصل ۱۴

Code Section

از این بخش برای برنامه‌نویسی بلاک‌های منطقی (OB,FC,FB) استفاده می‌شود. شکل ۱۴-۱۷ این محیط را برای OB1 در حالتی که View برنامه روی LAD قرار گرفته‌است، نشان می‌دهد.



شکل ۱۴-۱۷ محیط Code Section در OB1

برنامه‌ی کاربر در Network نوشته می‌شود. همچنین امکان ایجاد Network جدید نیز وجود دارد. انتخاب زبان برنامه‌نویسی از منوی View امکان‌پذیر می‌باشد.

همانطور که در شکل ۱۴-۱۷ مشاهده می‌شود، علاوه بر Network قسمت‌های دیگری نیز وجود دارد که عبارتند از:

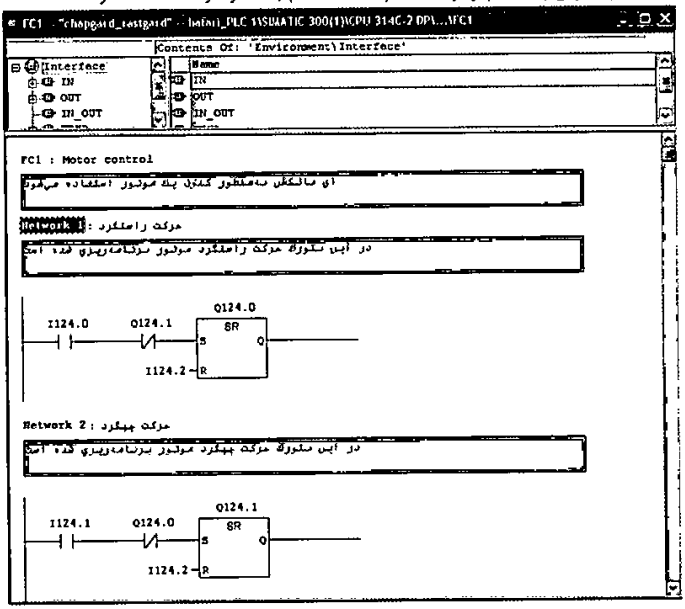
Block Title: عنوان یک بلاک را می‌توان در این قسمت وارد نمود. تعداد کارکتر مجاز ۶۴ کاراکتر می‌باشد.

Block Comment: توضیحات مربوط به بلاک را می‌توان در کادر خاکستری زیر Block Title، که در آن عبارت Comment درج شده است وارد نمود.

Network Title: در این قسمت می‌توان عنوان یک Network را مشخص نمود. حداکثر کارکتر مجاز 64 کارکتر می‌باشد.

Network Comment: در این قسمت که برای هر Network به‌طور مستقل وجود دارد، می‌توان توضیحاتی در مورد Network و عملکرد آن وارد نمود.

شکل ۱۴-۱۸ یک برنامه‌ی نوشته شده به‌زبان LAD را نشان می‌دهد که به‌منظور کنترل یک موتور به‌صورت چپگرد، راستگرد طراحی شده و عناوین (Titles) و توضیحات (Comment) بلاک و هر Network نوشته شده است.

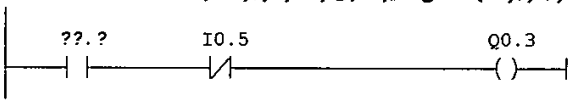


شکل ۱۴-۱۸ اختصاص Title و Comment برای بلاک و Network

جهت نوشتن برنامه در یک بلاک، توجه به نکات زیر توصیه می‌شود:

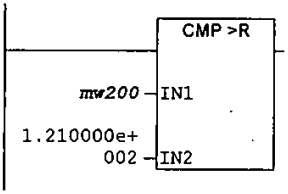
- بهتر است به ازای هر خروجی، از یک Network استفاده شود.
- در زبان‌های LAD و FBD می‌توان المان‌های مورد نظر را از قسمت Program Element انتخاب نموده و در Network قرار داد.

- در زبان‌های LAD و FBD نباید پایه‌ای از یک المان بدون آدرس باشد. اینگونه نواقص با علامت سوال قرمز رنگ ظاهر شده و با وجود آنها امکان ذخیره‌سازی و دانلود وجود ندارد.



شکل ۱۴-۱۹ نمونه ای از عدم اختصاص آدرس به یک المان برنامه‌نویسی

- آدرس داده شده باید با نوع داده‌ای که المان می‌پذیرد یکسان باشد. مثلاً به یک تابع که دو عدد INT (عدد صحیح 16 بیتی) را جمع می‌کند، نمی‌توان متغیر از نوع Bool اختصاص داد در غیر اینصورت رنگ آدرس مورد نظر به رنگ قرمز درآمده که بیانگر خطا در آدرس‌دهی می‌باشد. در این شرایط امکان ذخیره و دانلود وجود ندارد.



شکل ۱۴-۲۰ نمونه ای از اختصاص دیتای غلط به المان برنامه‌نویسی

- در زبان STL باید دستورات را در هر Network به‌صورت دستی تایپ نمود.
- در زبان STL نیز اگر آدرسی اشتباه وارد شود، رنگ آن قرمز می‌شود.

فصل ۱۴

۱۴-۶-۲ منوها و نوارابزار

همانطور که در شکل ۱۴-۲۱ مشاهده می‌شود، در محیط نرم‌افزار LAD/STL/FBD تعدادی منو و یک نوارابزار وجود دارند. در اینجا به این منوها و برخی از ابزارهای موجود در نوارابزار اشاره می‌شود. توضیحات تکمیلی در مورد منوها در بخش‌های آینده متناسب با نیاز ارائه می‌گردد.

شکل ۱۴-۲۱ منوها و Toolbar نرم‌افزار LAD/STL/FBD

برخی از منوهای مهم عبارتند از:

منوی File

در این منو می‌توان مدیریت روی فایل بلاک موجود، و ایجاد فایل جدید را انجام داد. همچنین در این منو امکان ایجاد Source مربوط به بلاک مورد نظر وجود دارد.

منوی Edit

در این منو دستوراتی جهت ویرایش بلاک و برخی از قسمت‌های مرتبط با بلاک قرار داده شده است.

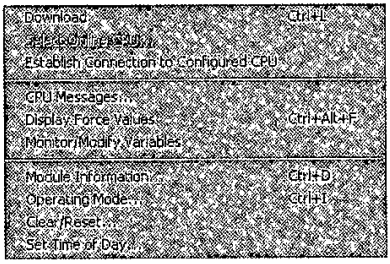
منوی Insert

در این منو امکان قرار دادن عناصر مختلف، مانند Network جدید، المان‌ها برنامه‌نویسی و . . . وجود دارد.

منوی PLC

در این منو امکانات مختلفی وجود دارد برخی از پرکاربردترین آنها عبارتند از:

- دانلود برنامه
- Monitor/Mofy و Force که در فصل‌های بعد تشریح می‌شود.
- مشاهده وضعیت CPU به‌صورت Online
- تغییر مدکاری
- ری‌ست کردن CPU
- تنظیم تاریخ و زمان CPU



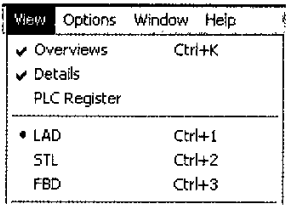
شکل ۱۴-۲۲ ابزارهای منوی PLC

منوی Debug

از ابزار موجود در این منو برای تست برنامه در مد Hold استفاده می‌شود.

منوی View

در این قسمت می‌توان تنظیماتی جهت نمایش بخش‌های مختلف موجود در محیط برنامه انجام داد. یکی از تنظیمات مهمی که در این منو قابل انجام می‌باشد، انتخاب زبان برنامه‌نویسی است. همانطور که در شکل ۱۴-۲۳ مشاهده می‌شود، می‌توان یکی از زبان‌های LAD، STL یا FBD را به‌عنوان زبان برنامه‌نویسی بلاک انتخاب نمود.

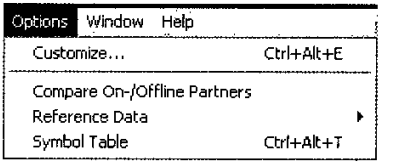


شکل ۱۴-۲۳ ابزارهای منوی View

منوی Options

در این منو برخی از کارهای اختیاری در نرم افزار قرار داده شده است. از جمله:

- Customize: می‌توان تنظیمات محیط برنامه‌نویسی را توسط آن تغییر داد. (توضیح در ادامه)
- Compare: می‌توان توسط آن برنامه Offline را با برنامه منتقل شده به CPU مقایسه نمود.

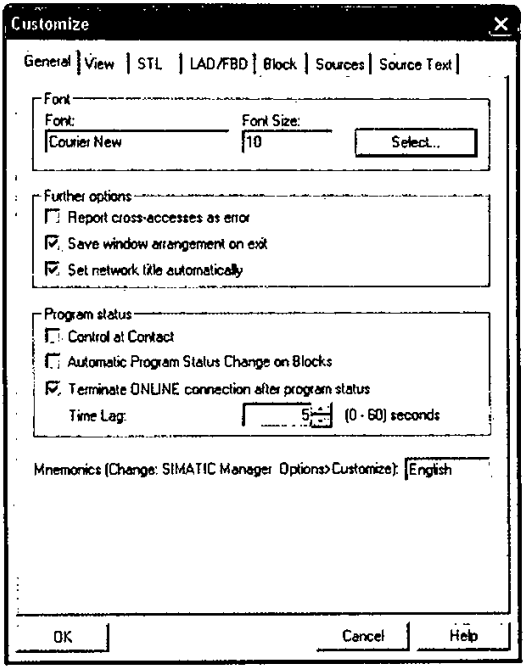


شکل ۱۴-۲۴ ابزارهای منوی Options

- Reference Data: می‌توان از آدرس‌های استفاده شده در برنامه اطلاع پیدا کرد (تشریح در فصل‌های بعد)
- Symbol Table: می‌توان جدول سمبل‌ها را باز کرد و تغییر داد. (توضیح در ادامه)

Customize

در صورتی‌که در منوی Options روی این گزینه کلیک شود، پنجره‌ای مانند شکل ۱۴-۲۵ اجرا می‌شود.



شکل ۱۴-۲۵ پنجره‌ی Customize

همانطور که در شکل ۱۴-۲۵ مشاهده می‌شود، در این پنجره سربرگ‌های مختلفی وجود دارد:

سربرگ General

در این سربرگ می‌توان تنظیمات عمومی مانند تنظیم فونت، ترتیب المان‌های موجود در محیط نرم‌افزار و . . . را انجام داد.

سربرگ View

در این سربرگ می‌توان تنظیمات نمایش مانند تنظیمات مربوط به نمایش یک بلاک پس از بازشدن، انتخاب رنگ حالت خطا و حالت نرمال، نمایش انتخاب زبان یک بلاک در هنگام باز شدن و . . . را انجام داد.

سربرگ STL

در این سربرگ برخی از تنظیمات مربوط به نمایش گزینه‌های مختلف، در زبان برنامه نویسی STL قابل انجام می‌باشد.

سربرگ LAD/FBD

دراین سربرگ برخی تنظیمات جهت نمایش المان‌ها و برخی از موارد دیگر در زبان‌های LAD/FBD قابل تنظیم می‌باشد.

سربرگ Block

در این سربرگ امکان تنظیم زبان یک بلاک به‌طور پیش‌فرض، همچنین ایجاد Reference Data با ایجاد بلاک و نیز تنظیم حالت یک FB در شرایطی که بتواند بیش از یک DB اختصاصی داشته باشد، قابل انجام می‌باشد.

سربرگ Sources

در این سربرگ تنظیمات مربوط به فایل‌های Source امکان‌پذیر است.

سربرگ Source Text

تنظیمات مربوط به نحوه نمایش فایل Source را می‌توان تغییر داد.

منوی Window

در این منو تنظیمات مربوط به چیدمان بخش‌های مختلف موجود در نرم‌افزار LAD/STL/FBD امکان‌پذیر می‌باشد.

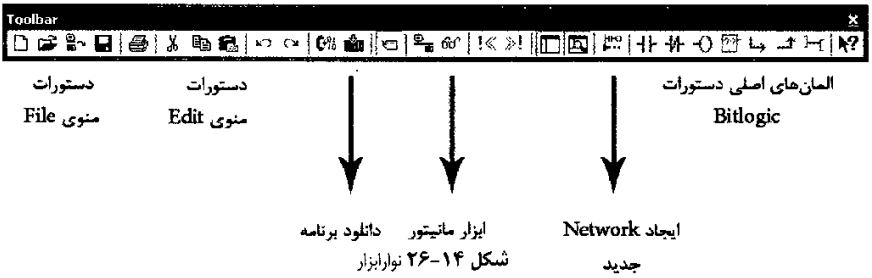
فصل ۱۴

منوی Help

در این منو یک Help نرم‌افزاری قدرتمند به زبان انگلیسی در اختیار کاربر قرار داده شده است که کاربر می‌تواند در مواقع نیاز اطلاعات مورد نظر خود را از آن استخراج نماید.

نوارابزار

در این نوار، برخی از المان‌ها و دستورات مهم موجود در منوها قرار داده شده است. فعال و غیر فعال نمودن نوار ابزار در منوی View با علامت زدن گزینه‌ی Toolbar صورت می‌پذیرد. در شکل ۱۴-۲۶ ابزارهای موجود در نوارابزار نشان داده شده و برای قسمت‌های مهم آن توضیحاتی ارائه شده است.

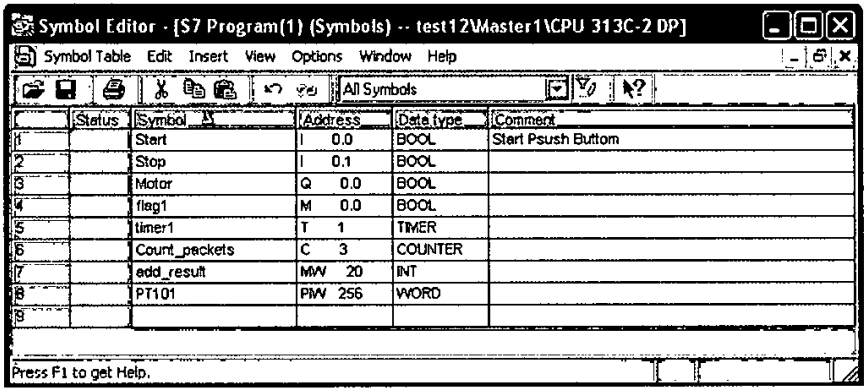


شکل ۱۴-۲۶ نوارابزار

۱۴-۷ ایجاد سمبل‌ها

برای ایجاد سمبل‌ها و مدیریت آنها کافیست تا در پوشه‌ی S7 Program روی گزینه‌ی Symbols دوبارکلیک نمود. البته از مسیرهای دیگری نیز می‌توان برنامه‌ی Symbol Editor را اجرا نمود که یکی از آنها کلیک بر روی گزینه‌ی Symbol Table در منوی Options در محیط نرم‌افزار LAD/STL/FBD می‌باشد.

در هر صورت از هر روشی که جهت اجرای نرم‌افزار Symbol Editor استفاده شود، محیطی مانند شکل ۱۴-۲۷ نشان داده می‌شود که در آن می‌توان سمبل‌های مورد نظر و توضیحات (Comment) را برای هر آدرسی مشخص نمود.



شکل ۱۴-۲۷ جدول سمبل‌ها در نرم‌افزار Symbol Editor

همانطور که در شکل ۱۴-۲۷ مشخص است، این جدول دارای ستون‌هایی به شرح زیر می‌باشد:

Symbol

در این ستون می‌توان نام سمبلیک را برای یک آدرس مشخص نمود. در انتخاب نام سمبلیک باید دقت شود که این نام مرتبط با نقشی باشد که آدرس مورد نظر برای آن درنظر گرفته شده است. مثلاً در سطر اول جدول فوق، نام سمبلیک Start استفاده شده است که نشان‌دهنده‌ی این است که آدرس I0.0 راه‌انداز موتور می‌باشد.

Address

در این ستون آدرسی که نام سمبلیک به‌آن اختصاص پیدا می‌کند، نوشته می‌شود. مثلاً در جدول فوق برای سطر اول آدرس I0.0 استفاده شده است. در اینجا می‌توان از هر آدرس مجاز دلخواهی استفاده کرد.

Data Type

در این ستون نوع داده مشخص می‌شود که مثلاً Bool یا Byte یا INT و ... باشد.

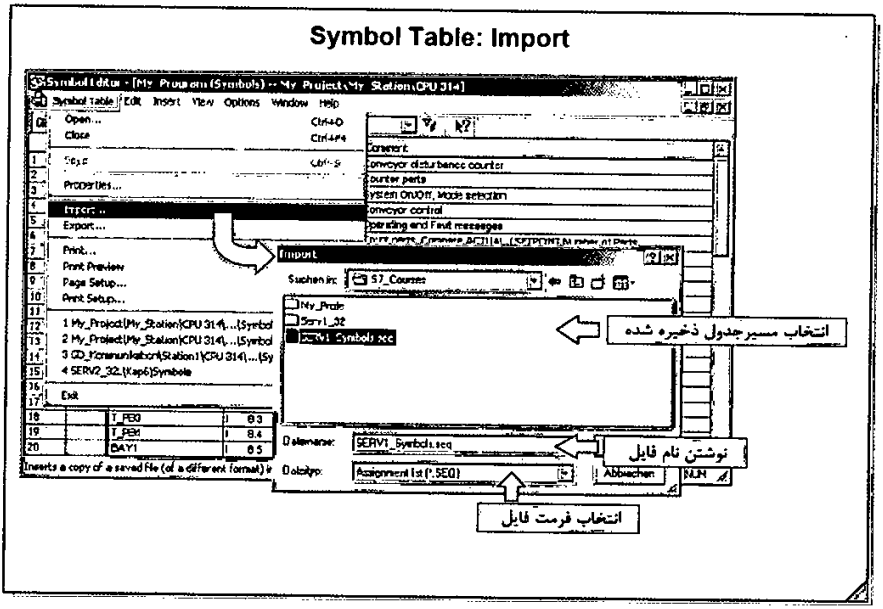
Comment

در این ستون می‌توان توضیحاتی را به‌زبان فارسی یا انگلیسی برای آدرس مورد نظر وارد نمود. مثلاً در جدول فوق مشخص است که آدرس 10.0 با نام سمبلیک Start به‌عنوان شستی استارت راه‌اندازی یک موتور می‌باشد.

در رابطه با تعریف سمبل‌ها، توجه به نکات زیر الزامی‌است:

- نام سمبل می‌تواند حداکثر 24 کاراکتر و تعدادسمبل‌ها حداکثر 16380 عدد باشد.
- هنگامی‌که برنامه به PLC دانلود می‌شود، جدول سمبل‌ها دانلود نمی‌شود.
- اگر برنامه‌ای را از PLC به کامپیوتر منتقل نماییم، در صورتی‌که سمبل‌های استفاده شده در برنامه‌ی آن، روی کامپیوتر از قبل موجود نباشد، نمایش داده نخواهد شد.

علاوه بر وارد نمودن دستی آدرس‌ها و سمبل‌ها، می‌توان آنها را از یک جدول سمبل‌های دیگر که قبلاً ایجاد و ذخیره شده است، مطابق شکل ۱۴-۲۸ با انتخاب گزینه‌ی Import از منوی Symbol Table وارد نمود.

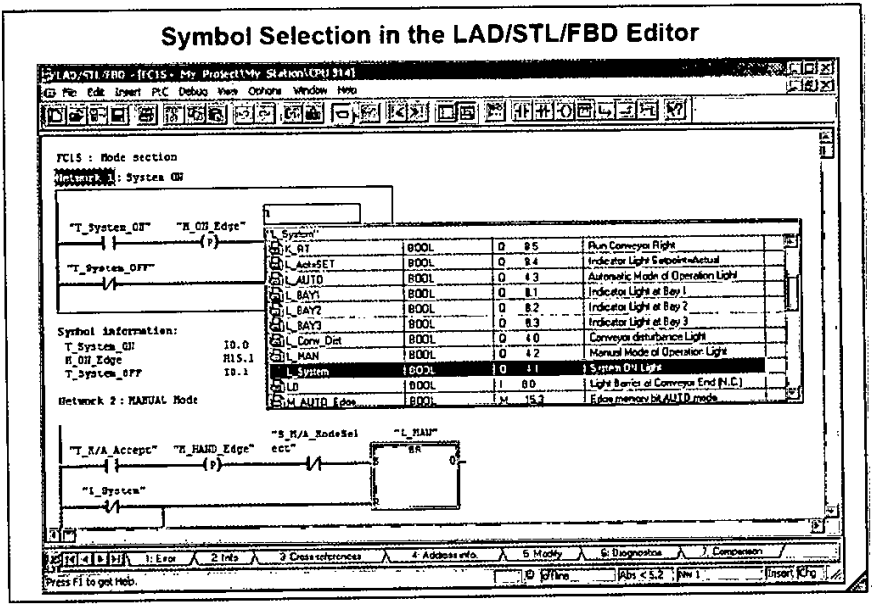


شکل ۱۴-۲۸ وارد نمودن جدول سمبل‌های ذخیره شده از قبل

همچنین از مسیر نشان داده شده در شکل ۱۴-۲۸ می‌توان با انتخاب گزینه‌ی Export، جدول سمبل‌های جاری را در مسیری دلخواه ذخیره نمود تا دز مواقع لزوم از آنها استفاده شود.

پس از اختصاص سمبل و Comment دلخواه به آدرس‌ها، می‌توان پس از انجام عمل Save از منوی Symbol Table، از محیط نرم‌افزار خارج شد.

حال اگر بخواهیم در برنامه به المانی آدرسی را اختصاص دهیم، یک کادر که حاوی سمبل‌های اختصاص داده شده است، باز شده و می‌توان از آنجا سمبل مورد نظر را انتخاب نمود. شکل ۱۴-۲۹ این موضوع را نشان می‌دهد.



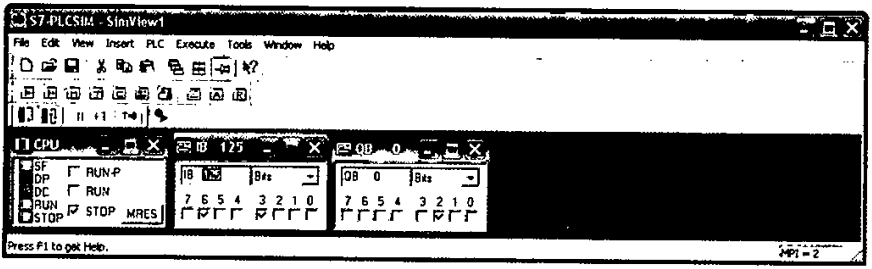
شکل ۱۴-۲۹ اختصاص آدرس سمبلیک به المان‌های برنامه

لازم به ذکر است‌که در محیط نرم‌افزار LAD/STL/FBD توسط منوی View زیر منوی Display With تنظیماتی جهت نمایش یا عدم نمایش سمبل‌ها در برنامه، انجام داد.

۱۴-۸ تست برنامه با سیمولاتور

به‌منظور تست برنامه، می‌توان از شبیه‌ساز نرم‌افزاری (PLC SIM) استفاده نمود. فراخوانی و نحوه‌ی ایجاد متغیرهای ورودی، خروجی در فصل بعد توضیح داده می شود. به‌منظور دانلود برنامه به PLC SIM، می‌توان روی گزینه دانلود از منوی PLC یا نوارابزار کلیک نمود؛ البته باید قبلاً PLC SIM اجرا شده باشد. همانطور که در شکل ۱۴-۳۰ مشخص است، علاوه بر کادر CPU یک کادر ورودی و یک کادر خروجی نیز قرار داده شده است. آدرس ورودی IB 125 و خروجی QB 0 وارد شده است که بیانگر آدرس یک بایت می‌باشد، از اینرو هر کدام از آنها دارای 8 خانه (8 بیت) می‌باشند. توسط موس می‌توان هر کدام از خانه‌های مورد نظر را علامت زد، در اینصورت مقدار آن متغیر در برنامه مقدار یک منطقی در نظر

گرفته می‌شود. توجه شود که به منظور اجرای برنامه توسط PLC SIM، باید وضعیت کاری CPU در حالت RUN یا RUN-P قرار داشته باشد.



شکل ۱۴-۳۰ نرم‌افزار PLC SIM

۱۴-۹ گروه دستورات برنامه‌نویسی LAD و FBD

در زبان LAD و FBD دستورات برنامه‌نویسی مختلفی وجود دارد، به‌طور کلی می‌توان این دستورات را به انواع زیر تقسیم نمود:

۱- **Bit logic:** این دستورات فقط برای خواندن و نوشتن روی یک بیت کار می‌کنند.

۲- **Comparator:** دستورات مقایسه‌گر که به منظور مقایسه دو متغیر با هم استفاده می‌شود.

۳- **Converter:** به‌منظور تبدیل انواع داده‌های مختلف به یکدیگر به‌کار می‌روند.

۴- **Counter:** به‌منظور انجام عمل شمارش از این دستورات استفاده می‌گردد.

۵- **DB call:** به‌منظور فراخوانی یک DB از این دستور استفاده می‌شود.

۶- **Jumps:** به‌منظور پرش از روی قسمتی از برنامه به‌کار می‌روند.

۷- **Integer Function:** به‌منظور انجام اعمال ریاضی روی اعداد صحیح به‌کار می‌روند.

۸- **Floating_Point Fuction:** به‌منظور انجام اعمال ریاضی روی اعداد اعشاری استفاده می‌شوند.

۹- **Move:** به‌منظور انتقال اطلاعات از یک ناحیه به ناحیه دیگر به‌کار می‌رود.

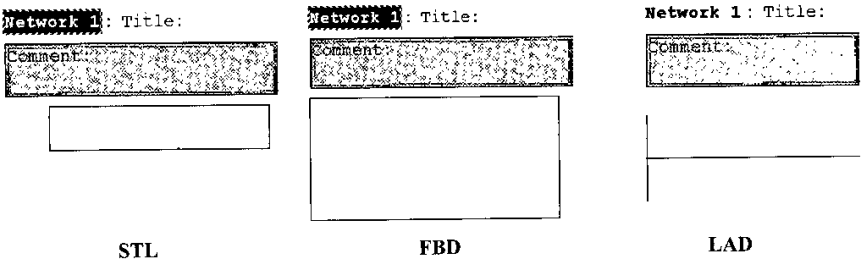
۱۰- **Program Control:** دستورات فراخوانی فانکشن‌ها و MCR هستند.

۱۱- **Shift/Rotate:** به‌منظور انجام جابه‌جایی یا چرخش در بیت‌های یک داده استفاده می‌گردد.

۱۲- **Timer:** به‌منظور محاسبه زمان، از دستورات تایمر استفاده می‌شود.

فصل ۱۴

برای شروع برنامه‌نویسی، لازم است ابتدا یک Network جدید در محیط برنامه ایجاد شود. اگر زبان STL باشد برنامه را به‌صورت متنی می‌نویسیم ولی در زبان LAD و FBD المان‌های برنامه‌نویسی از پنجره‌ی Programm Element وارد محیط می‌شوند که در فصل بعد تشریح خواهد شد.



شکل ۱۴-۳۱ وارد کردن Network در محیط برنامه‌نویسی

در فصل بعد دستورات براساس زبان LAD یا FBD تشریح می‌شوند. زبان STL در کتاب سطح تکمیلی آورده شده است.

۱۴-۱۰ پرسش و تحقیق

محیط برنامه‌نویسی Control Builder ABB را با محیط LAD/STL/FBD زیمنس مقایسه کنید.

۱۴-۱۱ تست‌های خودآزمایی

۱- برنامه اصلی PLC در کدام یک از بلاک‌های زیر نوشته می‌شود؟

الف) OB1 ب) FC ج) FB د) DB

۲- در این روش برنامه‌نویسی همه برنامه در OB1 نوشته می‌شود.

الف) ساختاریافته ب) خطی ج) تقسیم‌شده د) هر سه مورد

۳- در این نوع برنامه‌نویسی، برنامه به قسمت‌های مجزایی تقسیم می‌شود و در هر سیکل کاری، یک‌بار از درون OB1 فراخوانی می‌شود.

الف) ساختاریافته ب) خطی ج) تقسیم‌شده د) هر سه مورد

۴- در این روش برنامه‌نویسی از نوشتن قسمت‌های تکراری برنامه جلوگیری می‌شود.

الف) ساختاریافته ب) خطی ج) تقسیم‌شده د) هر سه مورد

۵- کدامیک از بلاک‌های زیر برای برنامه‌نویسی به‌کار نمی‌رود؟

الف) OB ب) FC ج) DB د) FC

۶- کدامیک از بلاک‌های زیر را نمی‌توان در OB1 فراخوانی نمود؟

الف) OB ب) FC ج) DB د) FC

۷- تفاوت FB با FC در چیست؟

الف) قابلیت فراخوانی از OB1 ب) اندازه بزرگتر

ج) وجود حافظه د) هر سه مورد

۸- این بلاک به‌منظور ذخیره‌سازی تنظیمات سیستم و اطلاعات ماژول‌ها به‌کار می‌رود.

الف) SFC ب) SFB ج) SDB د) هر سه مورد

۹- کدام تعریف برای DB از نوع Instance صحیح است؟

الف) مخصوص FC است. ب) برای همه بلاک‌ها قابل استفاده است.

ج) مخصوص FB است. د) به‌صورت اشتراکی است.

۱۰- در این زبان برنامه‌نویسی از عناصر گرافیکی مانند تیغه باز، تیغه بسته و بوبین استفاده می‌شود.

الف) LAD ب) FBD ج) GRAPH د) SCL

۱۱- در این زبان از دستورات گرافیکی گیت‌های منطقی استفاده می‌شود.

الف) LAD ب) FBD ج) GRAPH د) SCL

۱۲- از این زبان برنامه‌نویسی گرافیکی برای برنامه‌نویسی پروسه‌های ترتیبی استفاده می‌شود.

الف) LAD ب) FBD ج) GRAPH د) SCL

۱۳- استفاده از این زبان برنامه‌نویسی فقط در FB امکان‌پذیر است.

الف) LAD ب) FBD ج) GRAPH د) SCL

فصل ۱۴

فصل ۱۵

دستورات برنامه‌نویسی Bit Logic

۱۵-۱ مقدمه

۱۵-۲ دستورات Bit Logic در زبان LAD

۱۵-۳ دستورات Bit Logic در زبان FBD

۱۵-۴ تمرین

در این فصل برنامه‌نویسی با دستورات Bit Logic به زبان‌های LAD و FBD تشریح شده است.

چکیده مطالب

کاربرد	دستور LAD	دستور FBD
خواندن وضعیت یک بیت	---| |---	-
خواندن وضعیت معکوس یک بیت	---| / |---	-
عکس کردن نتیجه لاجیک	---| NOT |---	-o|
اختصاص نتیجه لاجیک به یک بیت	--( )--	=
ذخیره‌سازی نتایج میان برنامه	--(#)--	#
ری‌ست کردن یک بیت	--(R)	R
ست‌کردن یک بیت	--(S)	S
ست و ری‌ست کردن یک بیت با اولویت ست	RS S Q R	RS R S Q
ست و ری‌ست کردن یک بیت با اولویت ری‌ست	SR S Q R	SR S R Q
تشخیص لبه منفی نتیجه عملیات لاجیک	--(N)--	N
تشخیص لبه مثبت نتیجه عملیات لاجیک	--(P)--	P
ذخیره کردن RLO در یک بیت حافظه	-(SAVE)	SAVE
تشخیص لبه منفی یک بیت با آدرس مشخص	NEG Q M_BIT	NEG M_BIT Q
تشخیص لبه مثبت یک بیت با آدرس مشخص	POS Q M_BIT	POS M_BIT Q
OR دو یا چند بیت	-	>=1
AND دو یا چند بیت	-	&
Exclusive OR دو یا چند بیت	-	XOR

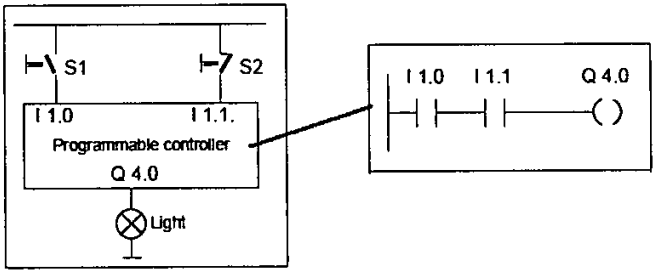
فصل ۱۵

۱۵-۱ مقدمه

در این بخش به تشریح دستوراتی که مرتبط با بیت هستند و به آنها Bit Logic اطلاق می‌شود می‌پردازیم.

روال بحث بر اساس دستورات LAD است. پس از تشریح آنها به معرفی دستورات FBD و شرح دستوراتی که در LAD معادل ندارند می‌پردازیم. دستورات برنامه نویسی STL در جلدهای بعدی کتاب خواهد آمد.

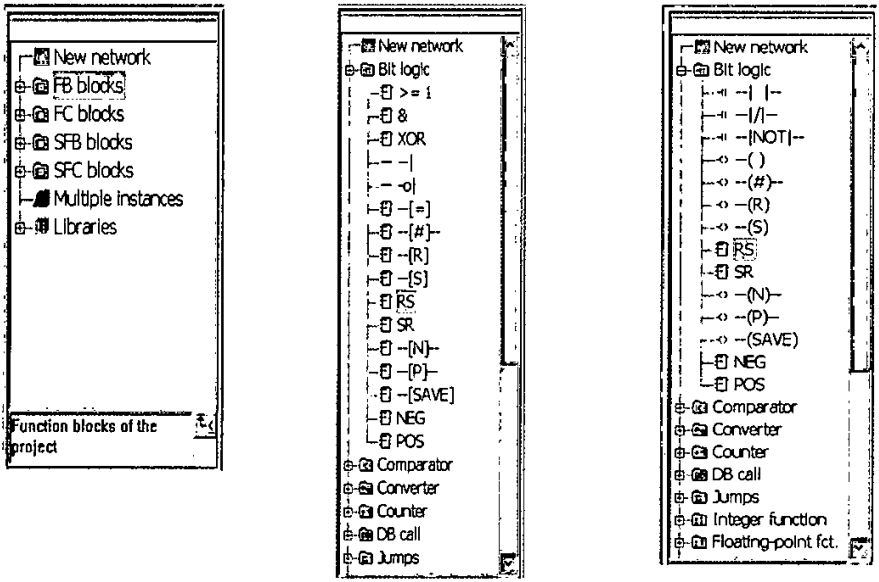
دستورات بیت بسیار پرکاربرد هستند. فرآیندهایی که در آنها سنسورو عملگر باینری زیادی به‌کار رفته باشد این دستورات حجم زیادی از برنامه نویسی را به‌خود اختصاص می‌دهند.



شکل ۱۵-۱ ارتباط بین ورودی و خروجی دیجیتال توسط برنامه

به‌طور کلی در ارتباط با دستورات Bit Logic به نکات زیر توجه شود:

- این دستورات برای خواندن یا نوشتن روی فقط یک بیت کاربرد دارند.
- بیت می‌تواند M و I و Q یا بیتی از دیتا بلاک یا کنتاکت تایمر باشد.
- سنسورهای دیجیتال، شستی‌ها و سوئیچ‌های اپراتوری که به کارت‌های دیجیتال ورودی متصل هستند توسط این دستورات خوانده می‌شوند.
- عملگرهای On/Off نظیر رله‌ها ، کنتاکتورها و سلونوئیدها توسط این دستورات فرمان می‌گیرند.
- این المان‌ها به‌طور کامل در Program Element موجود هستند. شکل ۱۵-۲ این بخش را برای زبان‌های LAD/FBD/STL نشان می‌دهد. باید توجه داشت که در زبان STL استفاده از دستورات Bit Logic به‌صورت متنی است و هیچ المانی برای آنها در Program Element موجود نیست.

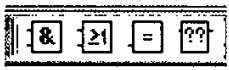


STL FBD LAD

شکل ۱۵-۲ دستورات Bit logic

- چند المان پر کاربرد این خانواده در نوارابزار بالای محیط LAD/STL/FBD در دسترس هستند.

المان‌های پر کاربرد LAD در نوارابزار



المان‌های پرکاربرد FBD در نوارابزار

شکل ۱۵-۳ بخشی از Toolbars محیط برنامه‌نویسی

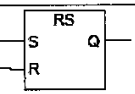
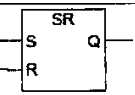
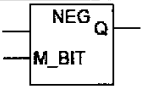
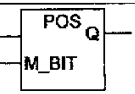
- نکته مهمی که در ارتباط با دستورات Bit Logic بایستی به آن توجه داشت این است که نتیجه عملیات منطقی در یک رجیستر یک بیتی از حافظه با عنوان RLO ذخیره می‌شود. این بیت در بسیاری از دستورات مبنای کار کنترل است. مفهوم RLO در خلال بحث توضیح داده می‌شود.

فصل ۱۵

۱۵-۲ دستورات Bit Logic در زبان LAD

این دستورات که به‌صورت المان‌های خاص در پنجره Program Element شکل ۱۵-۲ نمایش داده شده‌اند در جدول ۱۵-۱ لیست گردیده‌اند.

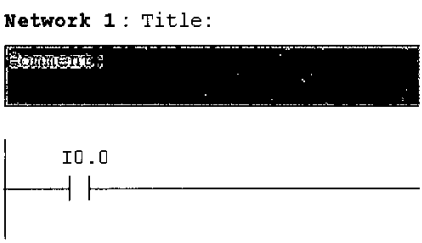
جدول ۱۵-۱

شکل	عنوان	کاربرد
---| |---	Normally Open Contact	خواندن وضعیت یک بیت
---| / |---	Normally Closed Contact	خواندن وضعیت معکوس یک بیت
--|NOT|--	Not	عکس کردن نتیجه لاجیک
---( )	Assign	اختصاص نتیجه لاجیک به یک بیت
---(#)---	Midline Output	ذخیره سازی نتایج میان برنامه
---(R)	Reset	ری‌ست کردن یک بیت
---(S)	Set	ست کردن یک بیت
	RS Flip Flop	ست و ری‌ست کردن یک بیت با اولویت ست
	SR Flip Flop	ست و ری‌ست کردن یک بیت با اولویت ری‌ست
---(N)---	Negative Edge Detection	تشخیص لبه منفی نتیجه عملیات لاجیک
---(P)---	Positive Edge Detection	تشخیص لبه مثبت نتیجه عملیات لاجیک
--(SAVE)	Save RLO	ذخیره کردن RLO در یک بیت حافظه
	Address Negative Edge Detection	تشخیص لبه منفی یک بیت با آدرس مشخص
	Address Positive Edge Detection	تشخیص لبه مثبت یک بیت با آدرس مشخص

کنتاکت NO

اولین المان موجود در لیست و جزو المان‌های پر کاربرد است که در نوارابزار نیز وجود دارد. توسط این کنتاکت وضعیت یک بیت خوانده می‌شود.

به‌عنوان مثال برای خواندن وضعیت سنسوری که به اولین کانال کارت دیجیتال با آدرس I0.0 متصل است داریم:



شکل ۱۵-۴ کنتاکت NO در برنامه نویسی LAD

در این شرایط:

- وقتی ورودی کارت یک می‌شود، کنتاکت NO فوق نیز یک می‌شود.
- وقتی ورودی کانال صفر می‌شود، کنتاکت NO فوق نیز صفر می‌گردد.

توجه شود که این کنتاکت را با کنتاکتور سنسور نباید اشتباه گرفت؛ چه کنتاکت سنسور به‌صورت نرمال باز باشد و چه کنتاکت سنسور به‌صورت نرمال بسته باشد می‌توان از کنتاکت NO در برنامه برای آدرس‌دهی آن استفاده کرد.

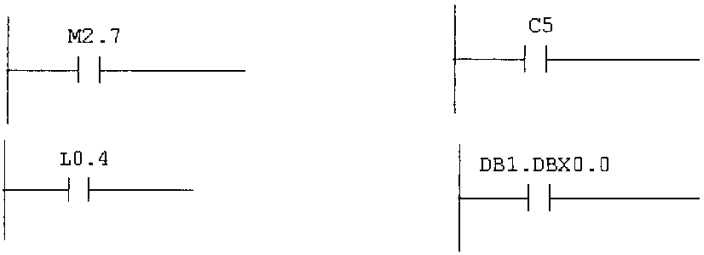
در این شرایط:

- وقتی کنتاکت سنسور نرمال باز باشد، با فعال شدن سنسور کنتاکت NO در برنامه یک می‌شود.
- وقتی کنتاکت سنسور نرمال بسته باشد، در شرایط عادی ورودی کارت یک است و کنتاکت NO در برنامه نیز یک است. با فعال شدن سنسور ورودی کارت صفر وکنتاکت NO نیز در برنامه صفر می‌شود.

علاوه بر خواندن ورودی‌ها، از کنتاکت NO می‌توان برای خواندن وضعیت یک خروجی نیز استفاده کرد. در شکل اول ۱۵-۵ وقتی خروجی Q0.1 یک شود این کنتاکت نیز یک می‌شود و وقتی خروجی فوق صفر شود این کنتاکت صفر خواهد بود. توجه شود که این کنتاکت فقط وضعیت را می‌خواند و نمی‌تواند فرمانی به خروجی بفرستد.

می‌توان از این کنتاکت برای خواندن یک بیت از حافظه یا یک بیت از دیتا بلاک یا وضعیت یک تایمر یا وضعیت کانتر نیز استفاده کرد.

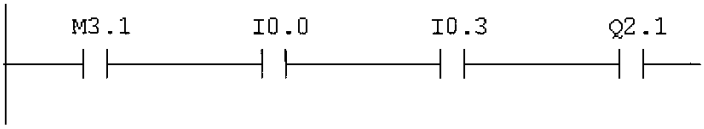
فصل ۱۵



شکل ۱۵-۵ بیت‌های مختلف قابل استفاده در Bit Logic

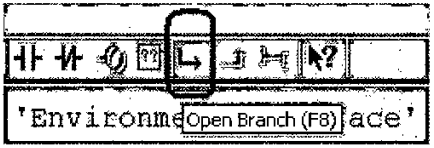
سری و موازی کردن کنتاکت‌ها

کنتاکت‌ها را می‌توان با هم سری کرد، برای این منظور آنها را به‌دنبال هم روی خط LAD قرار می‌دهیم.



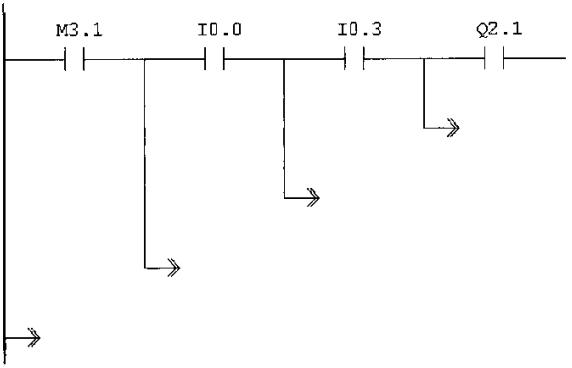
شکل ۱۵-۶ سری کردن کنتاکت‌ها در برنامه‌نویسی LAD

کنتاکت‌ها را می‌توان با هم موازی کرد. برای ایجاد شاخه موازی روی آیکن Open Branch از نوارابزار مانند شکل ۱۵-۷ کلیک می‌کنیم.



شکل ۱۵-۷ ایجاد شاخه موازی با آیکن نوارابزار

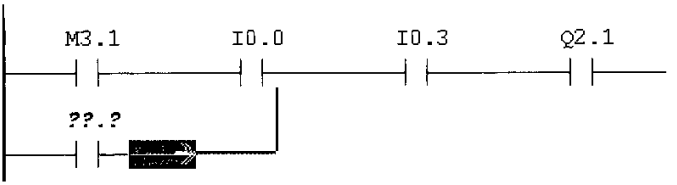
اگر هدف ایجاد یک مسیر موازی از ابتدا باشد، اول روی خط عمودی و سپس روی آیکن فوق کلیک می‌کنیم. اگر هدف ایجاد شاخه موازی از وسط کنتاکت‌ها باشد، در نقطه مورد نظر کلیک کرده سپس روی آیکن فوق کلیک می‌کنیم.



شکل ۱۵-۸ ایجاد انشعاب در برنامه‌نویسی LAD

شاخه موازی نمی‌تواند خالی باشد و نمی‌توان آنرا بلاتکلیف رها کرد؛ بلکه پس از چیدن کنتاکت‌ها روی آن لازم است به‌صورت مناسب به مسیر موازی دیگر متصل گردد. برای این‌کار می‌توان به یکی از دو روش زیر عمل کرد.

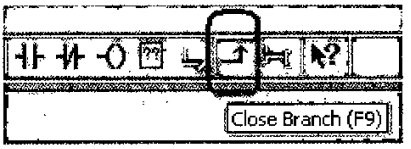
روش اول: با موس فلش انتهای مسیر موازی را گرفته و می‌کشیم. سپس در نزدیکی محل اتصال مورد نظر و با تغییر شکل موس آنرا رها می‌کنیم؛ در نتیجه اتصال موازی ایجاد می‌شود.



شکل ۱۵-۹ بستن شاخه موازی با Drag کردن توسط موس

فصل ۱۵

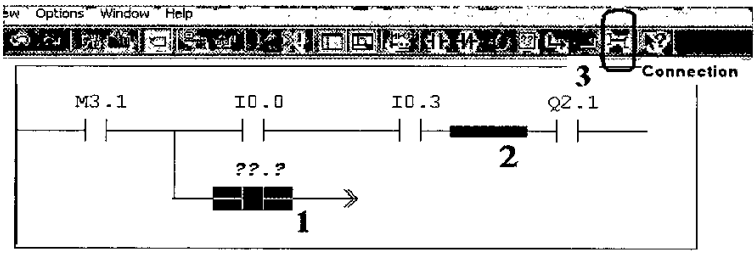
روش دوم: با موس روی فلش انتهای مسیر موازی کلیک می‌کنیم. اکنون آیکن Close Branch مانند شکل ۱۵-۱۰ در نوار ابزار فعال می‌شود؛ با کلیک روی آن مسیر موازی بسته می‌شود.



شکل ۱۵-۱۰ بستن شاخه موازی با آیکن نوار ابزار

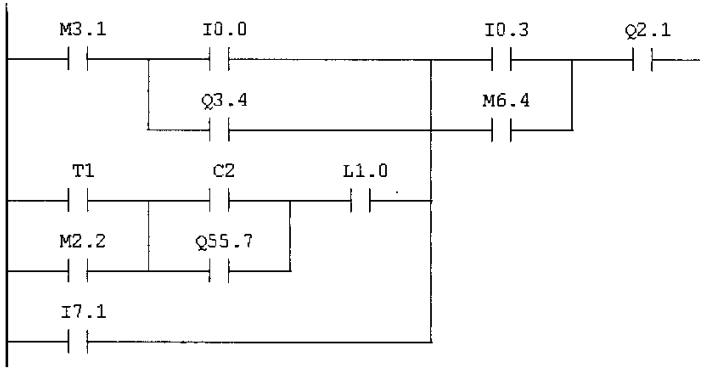
روش سوم : روی فلش انتهای مسیر موازی کلیک می‌کنیم. سپس کلید شیفت را پایین نگه داشته و در محل مقصد مورد نظر نیز کلیک می‌کنیم. این دو به‌صورت مستطیل‌های رنگی مانند شکل ۱۵-۱۱ نمایان می‌شوند. در این‌حالت با کلیک روی آیکن نشان داده شده در شکل ۱۵-۱۱ یا با استفاده از کلید میانبر F12 اتصال را ایجاد می‌کنیم.

اگر امکان اتصال بین مبدا و مقصد نباشد آیکن یا کلید فوق فعال نمی‌شوند.



شکل ۱۵-۱۱ بستن شاخه موازی با استفاده از آیکن Connection

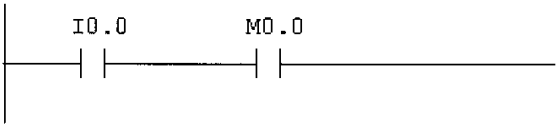
تمرین۱۵-۱: با توجه به توضیحات فوق برنامه زیر را پیاده‌سازی کنید.



شکل ۱۵-۱۲ برنامه تمرین ۱۵-۱

AND کنتاکت های NO

وقتی چند کنتاکت با یکدیگر سری شوند، منطق ایجاد شده مفهوم AND را دارد. به‌عنوان مثال در شکل۱۵- ۱۳ وقتی هر دو کنتاکت یک شوند *نتیجه عملیات منطقی* یک خواهد بود ولی اگر یکی از آنها صفر یا هر دو صفر باشند *نتیجه عملیات منطقی* صفر خواهد بود. ما در اینجا روی عبارت نتیجه عملیات منطقی تاکید کرده‌ایم، این عبارت ترجمه Result Of Logic Operation است که به‌اختصار RLO خوانده شده و معرف یک بیت در حافظه CPU است که نتیجه عملیات منطقی در آن قرار می‌گیرد. از RLO در بحث‌های آینده استفاده خواهیم نمود.

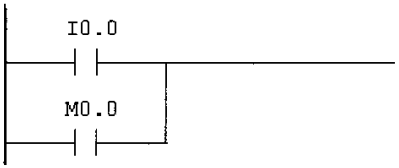


Result of Logic Operation = I0.0 AND M0.0

شکل ۱۵-۱۳ بیان سری کردن کنتاکت‌ها با بیان ریاضی AND

OR کردن کنتاکت‌های NO

وقتی چند کنتاکت با یکدیگر موازی شوند مفهوم OR خواهند داشت. در این شرایط اگر یکی از کنتاکت‌ها یا هر دو یک شوند نتیجه عملیات یعنی RLO یک خواهد بود.

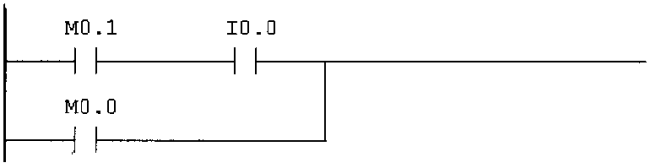


Result of Logic = I0.0 OR M0.0

شکل ۱۵-۱۴ بیان موازی کردن کنتاکت‌ها با بیان ریاضی OR

وقتی کنتاکت‌ها با هم سری و موازی می‌شوند با توجه به اینکه AND از نظر محاسباتی نسبت به OR دارای اولویت است، ممکن است نیاز به کاربرد پرانتز در بیان نتیجه به‌صورت منطقی باشد.

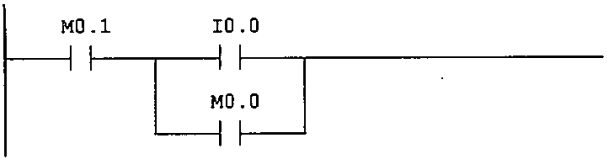
در شکل ۱۵-۱۵ ابتدا دو کنتاکت سری شده با هم AND می‌شوند و نتیجه با کنتاکت موازی شده OR می‌شود. در این‌حالت نیازی به استفاده از پرانتز نیست.



Result of Logic Operation = M0.1 AND I0.0 OR M0.0

شکل ۱۵-۱۵ کاربرد AND و OR بدون نیاز به پرانتز

در شکل زیر CPU ابتدا نتیجه کنتاکت‌های موازی را محاسبه کرده، سپس نتیجه را با کنتاکت سری AND می‌کند. بنابراین در بیان آن با AND و OR نیاز به استفاده از پرانتز وجود دارد.



Result of Logic Operation = M0.1 AND (I0.0 OR M0.0)

شکل ۱۵-۱۶ کاربرد AND و OR با نیاز به پرانتز

تمرین ۱۵-۲: معادل برنامه تمرین ۱۵-۱ را با استفاده از AND و OR بنویسید.

کنتاکت NC

NC نیز از المان‌های پر کاربرد خانواده Program Element است که آیکن آن در نوار ابزار نیز وجود دارد.

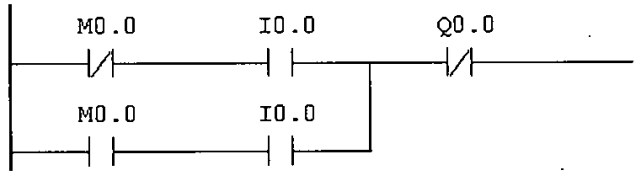
این کنتاکت را نباید با کنتاکت نرمال بسته سنسور اشتباه کرد. به‌عبارت دیگر نباید اینطور تصور شود که هرجا تیغه سنسوری نرمال بسته باشد در برنامه‌نویسی نیز باید از کنتاکت NC برای آن استفاده کرد. این تصور نادرست است.

NC در واقع معکوس NO است و یک عنصر لاجیکی برای آدرس دیجیتال ورودی است. برنامه نویس بسته به نیاز می‌تواند از کنتاکت NC یا NO در برنامه استفاده نماید. به توضیحات جدول ۱۵-۲ توجه کنید.

جدول ۱۵-۲

شرایط تیغه سنسور		وضعیت ورودی کارت		وضعیت کنتاکت NO در برنامه		وضعیت کنتاکت NC در برنامه	
عادی	فعال	عادی	فعال	عادی	فعال	عادی	فعال
باز	بسته	صفر	یک	صفر	یک	یک	صفر
بسته	باز	یک	صفر	یک	صفر	صفر	یک

مشابه آنچه برای کنتاکت‌های NO ذکر گردید، از کنتاکت NC می‌توان برای خواندن انواع بیت‌های حافظه استفاده کرد. سری و موازی کردن آنها نیز به همان روش است. شکل ۱۵-۱۷ برخی از این موارد را نشان می‌دهد. ترکیب کنتاکت‌های NC با کنتاکت‌های NO نیز در شکل نشان داده شده است.



$$RLO = ((\overline{M0.0}\ AND\ I0.0)\ OR\ (M0.0\ AND\ I0.0))\ AND\ \overline{Q0.0}$$

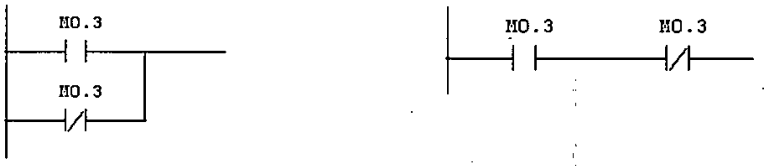
شکل ۱۵-۱۷ ترکیب کنتاکت های NC و NO

همانطور که در پایین شکل ۱۵-۱۷ دیده می‌شود، در بیان کنتاکت NC در عبارات AND و OR لازم است آنرا با علامت معکوس نمایش دهیم.

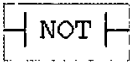
ساخت سیگنال‌های همیشه یک و همیشه صفر

گاهی اوقات در برنامه‌نویسی نیاز به استفاده از سیگنال Always one وجود دارد. یعنی منطقی که نتیجه عملیات آن همیشه یک باشد. برای این منظور می‌توان دو کنتاکت NO و NC یک آدرس دلخواه را با هم موازی کرد. بدیهی است نتیجه OR دو سیگنالی که نقیض هم باشند همواره یک است.

همینطور برای ساخت سیگنال همیشه صفر[1] می‌توان دو کنتاکت NO و NC یک آدرس دلخواه را با هم سری نمود زیرا نتیجه AND دو سیگنالی که نقیض هم باشند همواره صفر است.



شکل ۱۵-۱۸ نحوه ایجاد سیگنال های همیشه صفر و همیشه یک

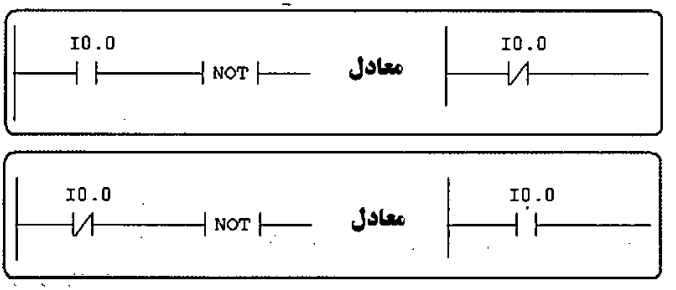


NOT

این المان همانطور که از نامش پیداست منجر به معکوس شدن نتیجه می‌گردد، بنابراین در هر جایی از برنامه نویسی LAD استفاده شود منجر به عکس شدن نتیجه عملیات منطقی یعنی معکوس شدن RLO می‌گردد.

بدیهی است که:

- اگر NOT با کنتاکت NO سری شود، معادل کنتاکت NC خواهد بود.
- اگر NOT با کنتاکت NC سری شود، معادل کنتاکت NO خواهد بود.

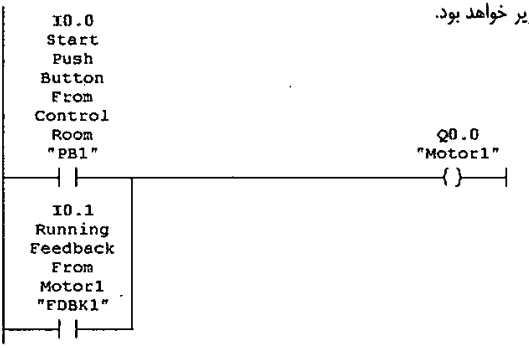


شکل ۱۵-۱۹ معادل‌سازی NOT با کنتاکت‌های NC و NO

1. Always Zero

در اینجا اگر یک لحظه شستی I0.0 وصل و سپس قطع شود خروجی یک شده و یک می‌ماند؛ زیرا در سیکل اسکن‌های بعد از فعال‌شدن شستی خروجی که یک شده با ورودی موازی و نتیجه عملیات منطقی یک می‌ماند.

راه حل دوم: در روش قبل لازم است دقت شود که کنتاکت موازی شده Q0.0 یک آدرس منطقی است نه یک آدرس واقعی که به‌عنوان فیدبک از کنتاکتور موتور گرفته شده باشد، به همین علت ممکن است در برخی شرایط کنترل صحیحی نداشته باشیم. فرض کنید بوبین کنتاکتور مشکل دارد و با ارسال فرمان از سمت PLC نتواند برق‌دار شود. در این شرایط از دیدگاه PLC خروجی Q0.0 یک است ولی عملاً موتور روشن نشده است. اگر فیدبکی از کنتاکت کمکی کنتاکتور گرفته شود مشکل فوق برطرف می‌شود. این فیدبک به یک ورودی دیجیتال دیگر متصل می‌شود، اگر آدرس این ورودی I0.1 باشد برنامه به‌صورت زیر خواهد بود.



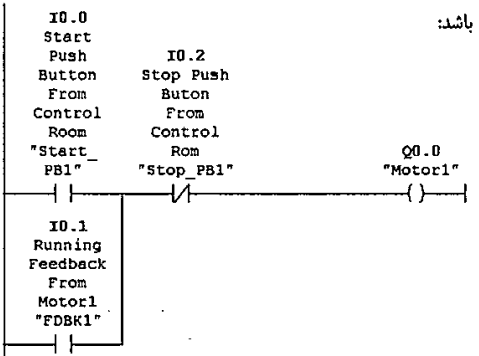
شکل ۱۵-۳۰ راه حل دوم برنامه مثال ۱۵- ۲

برنامه‌های فوق را با سیمولاتور تست کنید. برای دیدن عملکرد برنامه کافیست یک لحظه با موس روی I0.0 در سیمولاتور کلیک تا فعال شود؛ سپس با کلیک مجدد آن را غیرفعال کنیم. این حالت نقش شستی را خواهد داشت.

مثال ۱۵-۳: برنامه مثال ۱۵- ۲ پس از روشن شدن همچنان روشن می‌ماند برای قطع کردن آن نیاز به کنتاکتی است که با لاجیک قبلی سری شود.

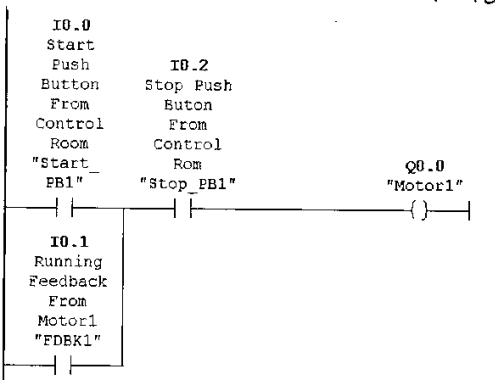
اگر فرض کنیم که اپراتور از شستی دیگری برای استپ استفاده می‌کندکه به I0.2 متصل است، ممکن است این شستی نرمال باز یا نرمال بسته باشد که لاجیک برای این دو متفاوت است.

اگر شستی استپ نرمال باز باشد:



شکل ۱۵-۳۰ برنامه مثال ۱۵-۳ وقتی از شستی نرمال باز استفاده شود.

اگر شستی استپ نرمال بسته باشد:

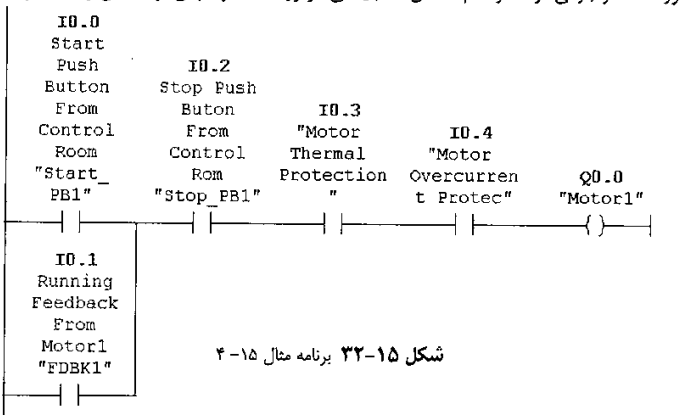


شکل ۱۵-۳۱ برنامه مثال ۱۵- ۳ وقتی از شستی نرمال بسته استفاده شود.

به‌طور معمول شستی استپ نرمال بسته است تا اگر سیم متصل به آن نیز قطع شد سیستم استپ شود و این از نظر ایمنی بهتر است.

برای تست برنامه با سیمولاتور لازم است روی آدرس‌هایی که کنتاکت‌های نرمال بسته دارند با موس کلیک کرده و آنها را یک کنیم. بدیهی است وقتی می‌خواهیم یک شستی نرمال بسته که توسط اپراتور فشار داده شده را تست کنیم، کافی است در سیمولاتور روی آدرس مورد نظر یک‌بار کلیک کرده تا به صفر برگردد و یک‌بار دیگر آنرا یک کنیم.

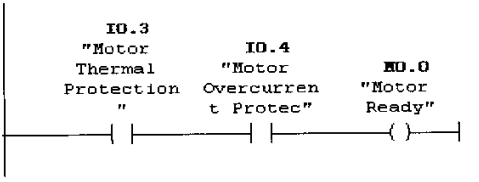
مثال ۱۵-۴: برنامه مثال ۱۵- ۳ را در حالتی که شستی استپ نرمال بسته است در نظر بگیرید. به‌طور معمول.این برنامه برای کنترل موتور کافی نیست و لازم است اینترلاک‌های ایمنی نیز در نظر گرفته شود. اینترلاک‌ها می‌توانند مربوط به اضافه جریان یا دمای بالای موتور یا دور بالای موتور و امثال آنها باشند. کنتاکت‌های حفاظتی فوق معمولاً نرمال بسته هستند ودرصورت‌عملکردبازمی‌شوند،هرکدام‌که‌عمل‌کنندبایستی موتور Off شود. پس برنامه می‌تواند به‌صورت زیر باشد:

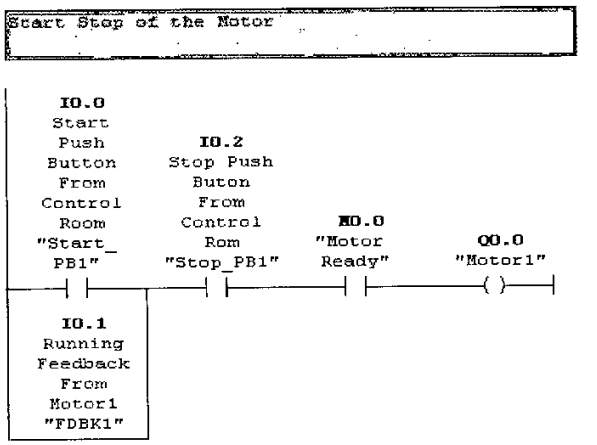


شکل ۱۵-۳۲ برنامه مثال ۱۵- ۴

فصل ۱۵

وقتی تعداد اینترلاک‌ها زیاد باشد به‌جای سری‌کردن آنها در برنامه می‌توان در Network دیگری مانند شکل ۱۵-۳۳ آنها را سری کرد و سیگنال آمادگی را ساخت و این سیگنال را در برنامه استارت و استپ موتور استفاده نمود. همانطور که در شکل ۱۵-۳۳ دیده می‌شود، این سیگنال در یک بیت حافظه Bit Memory ذخیره شده است.





شکل ۱۵-۳۳ راه حل دوم برنامه مثال ۱۵-۴

مثال ۱۵- ۵: استفاده از پالس برای تولید آلارم

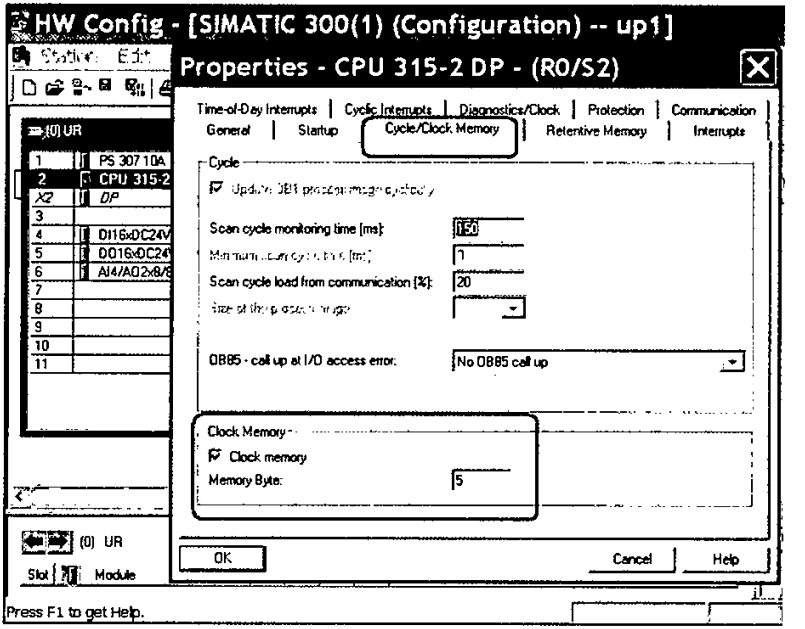
می‌خواهیم برنامه مثال قبل را به‌صورتی کامل کنیم که اگر به‌دلیل عمل کردن حفاظت، موتور قطع شد یا فرمان استارت نگرفت آلارم چشمک‌زن روی خروجی Q1.0 نمایش داده شود.

برای این کار لازم است ابتدا پالسی با فرکانس دلخواه ایجاد شده و از آن در برنامه استفاده گردد. تولید پالس به روش‌های مختلفی قابل انجام است. با استفاده از تایمر یا SFB یا با OB3x می‌توان پالس تولید کرد ولی ساده‌ترین روش استفاده از کلاک‌پالس CPU می باشد.

Clock Pulse در تمام CPUها وجود دارد و هشت پالس با فرکانس‌های مختلف به‌صورت سخت‌افزاری توسط CPU ایجاد می‌شود. کافیست این هشت پالس را به یک بایت حافظه[1] منتقل کنیم و از بیت دلخواه آن در برنامه استفاده نماییم.

[1]. Memory Byte

برای فعال‌سازی Clock Pulse در HW Config روی CPU دوبار کلیک نموده و در سربرگ Cycle/Clock Memory در پایین گزینه Clock Memory را فعال و آنرا به حافظه با شماره دلخواه نسبت می‌دهیم. شماره حافظه مورد نظر در اینجا عدد 5 انتخاب شده است که منظور از آن MB5 است.



شکل ۱۵-۳۴ فعال‌سازی پالس در پارامترهای CPU

پس از دانلود کردن این تنظیمات به CPU، هشت پالس در بیت‌های مختلف MB5 ریخته خواهند شد. شکل ۱۵-۳۵ فرکانس این پالس‌ها را نشان می‌دهد. بدین ترتیب مثلاً در بیت M5.3 پالسی با فرکانس 2Hz خواهیم داشت.

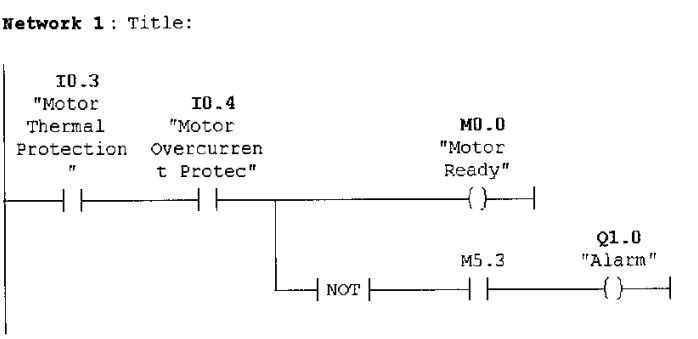
Bit	7	6	5	4	3	2	1	0
Period duration (s):	2	1.6	1	0.8	0.5	0.4	0.2	0.1
Frequency (Hz):	0.5	0.625	1	1.25	2	2.5	5	10

شکل ۱۵-۳۵ فرکانس پالس های CPU

فصل ۱۵

لازم است توجه شود که آدرس‌های MB5 در برنامه فقط برای پالس هستند و نباید برای کاربرد دیگری از آنها استفاده نمود.

پس از انجام مراحل فعال‌سازی پالس به برنامه باز می‌گردیم و Network1 را به‌صورت شکل ۱۵-۳۶ ویرایش می‌کنیم.



شکل ۱۵-۳۶ برنامه مثال ۱۵- ۵

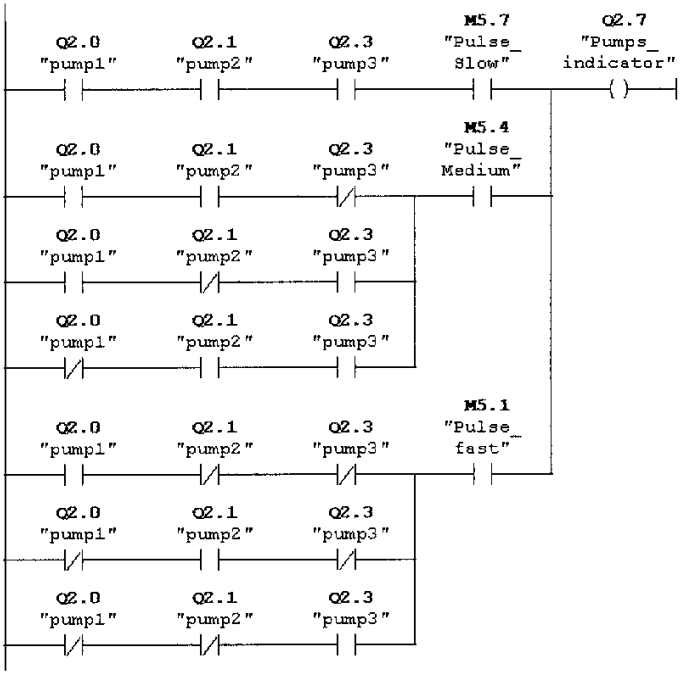
خواننده می‌تواند برنامه را به سیمولاتور دانلود و نتیجه را تست کند. در حالت Online بیت M5.3 خاموش روشن می‌شود. برای تست آدرس‌های دیگر مانند M5.1 و M5.7 را به‌جای M5.3 به‌کار ببرید و نتیجه را تست کنید.

مثال ۱۵- ۶ : نمایش عملکرد سه پمپ با چراغ چشمک‌زن

در فرآیندی سه پمپ که موتور آنها به‌ترتیب به آدرس‌های Q2.0 و Q2.1 و Q2.2 متصل است وجود دارند. نمایش وضعیت آنها با فرکانس چشمک‌زدن چراغ Q2.7 برای اپراتور نشان داده می‌شود. برنامه را به‌صورتی بنویسید که:

- اگر هر سه پمپ کار روشن باشند، چراغ Q2.7 با فرکانس کم چشمک بزند.
- اگر دو پمپ از سه پمپ روشن باشند، چراغ Q2.7 با فرکانس متوسط چشمک بزند.
- اگر یک پمپ از سه پمپ روشن باشند، چراغ Q2.7 با فرکانس زیاد چشمک بزند.
-

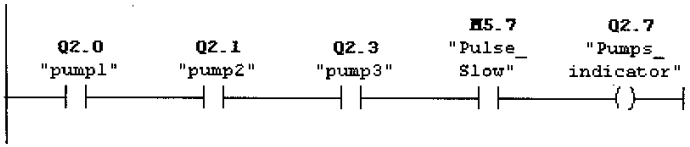
برنامه به‌صورت شکل ۱۵-۳۷ خواهد بود. تحلیل آن به عهده خواننده است.



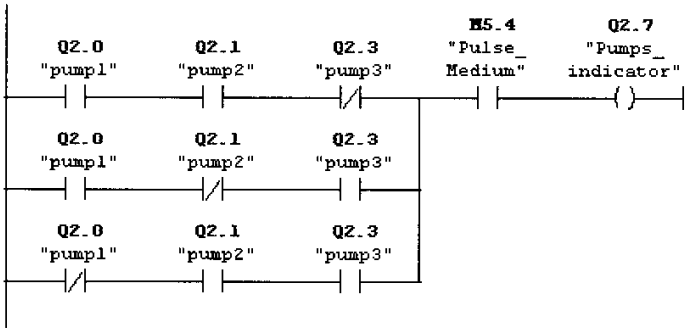
شکل ۱۵-۳۷ راه حل درست برنامه مثال ۱۵-۶

سوال: اگر برنامه قبل به اشتباه به‌صورت زیر نوشته شود، چه اتفاقی می‌افتد؟

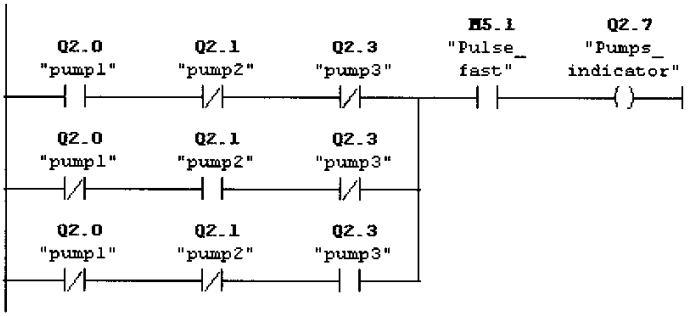
فصل
۱۵



Network 5 : Title:



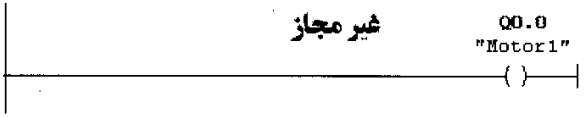
Network 6 : Title:



شکل ۱۵-۳۸ راه حل نادرست برنامه مثال ۱۵-۶

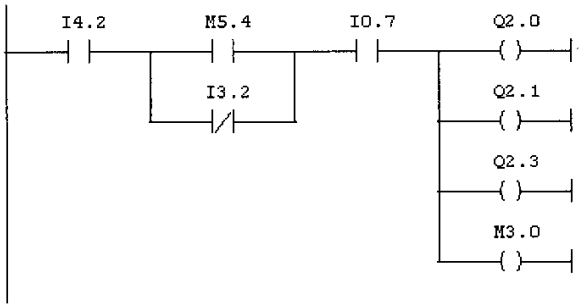
تذکر: با مثال‌هایی که تا اینجا ذکر گردید عملکرد Coil روشن شد. به نکات زیر در استفاده از Coil توجه داشته باشید:

- Coil به تنهایی نمی‌تواند در برنامه استفاده شود، بنابراین برنامه شکل ۱۵-۳۹ قابل ذخیره و دانلود کردن نیست.



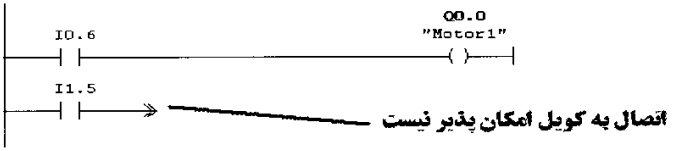
شکل ۱۵-۳۹ کاربرد غیر مجاز Coil

- مطابق شکل ۱۵-۴۰ نتیجه عملیات یک لاجیک را می‌توان به چندین Coil اختصاص داد.



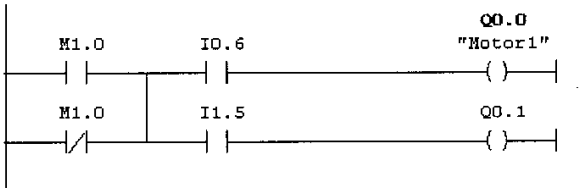
شکل ۱۵-۴۰ اختصاص RLO به چند Coil

- از مسیر موازی مستقل (یا در اصطلاح Rung مستقل) نمی‌توان به Coil جداگانه‌ای مقدار اختصاص داد. این کار فقط با زبان STL امکان‌پذیر است. به شکل ۱۵-۴۱ توجه کنید.



شکل ۱۵-۴۱ کاربرد غیر مجاز شاخه موازی

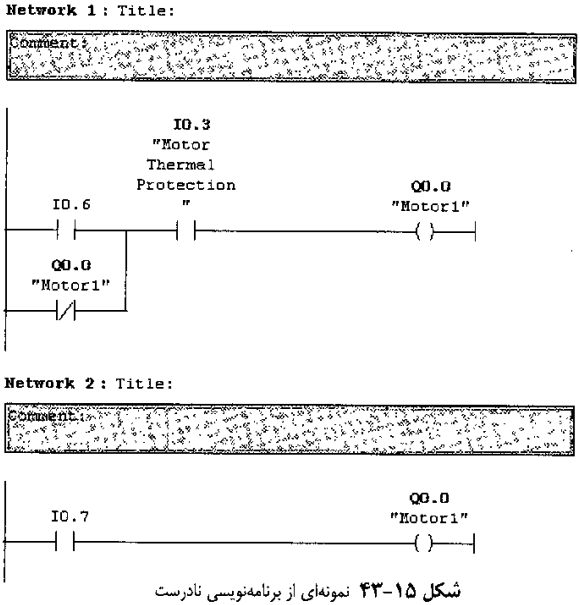
- برای رفع مشکل فوق می‌توان مانند شکل ۱۵-۴۲ یک لاجیک یا یک سیگنال همیشه یک در ابتدای خط قرار داد و از آن چندین انشعاب مستقل گرفت.



شکل ۱۵-۴۲ کاربرد درست شاخه موازی

- در برنامه نباید بیش از یک بار به Coil با یک آدرس مشخص فرمان داد. در صورت تکراری بودن نتیجه لاجیک آخر مبنا خواهد بود. در شکل ۱۵-۴۳ هر فرمانی که در Network1 برای Q0.0 ایجاد شده باشد، در Network2 توسط نتیجه عملیات منطقی جایگزین می‌شود. از آنجا که فرامین در انتهای سیکل اسکن به خروجی‌ها ارسال می‌شوند پس مبنا آخرین فرمانی است که توسط Network2 ایجاد شده است.

فصل ۱۵



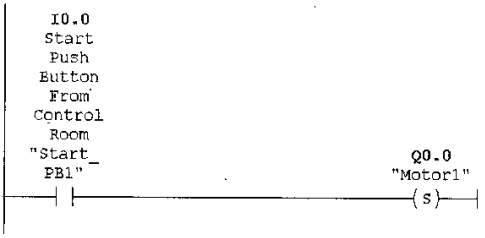
شکل ۱۵-۴۳ نمونه‌ای از برنامه‌نویسی نادرست

- می‌توان در Coil هر آدرس بیتی را به‌کار برد. می‌توان علاوه بر M و Q و بیت DB حتی از I نیز در Coil استفاده کرد؛ اگرچه این روش مرسوم نیست. اگر I در Coil استفاده شود و ورودی فیزیکی با این آدرس وجود نداشته باشد مشکلی نیست ولی اگر ورودی با این آدرس وجود داشته باشد که در ابتدا سیکل اسکن مقدار آن به PII وارد شده است، این برنامه باعث می‌شود که مقدار خوانده شده فوق از بین رفته و با نتیجه لاجیک جایگزین شود. پس به‌طور کلی نباید از I در آدرس Coil استفاده نمود.

Set

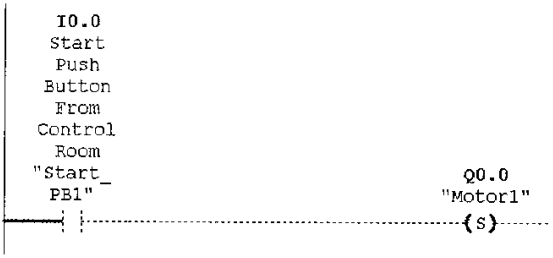
دستور Set باعث می‌شود که بیت آدرس داده شده یک شده و یک باقی بماند. در واقع یک حالت خودنگه‌دار برای آدرس مورد نظر ایجاد می‌کند.

مثال ۱۵-۷: برنامه مثال ۱۵-۲ با استفاده از دستور Set به‌صورت زیر خواهد بود. وقتی شستی استارت یک لحظه فشار داده شود، Q0.0 یک شده و یک باقی می‌ماند.



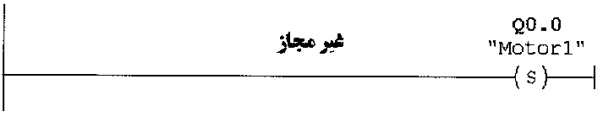
شکل ۱۵-۴۴ برنامه مثال ۱۵-۷

اگر برنامه فوق را به PLC دانلود و روی آیکن عینک شکل Monitor کلیک کنیم، می‌بینیم وقتی یک لحظه شستی فشار داده می‌شود خط سبز رنگ شده یعنی RLO یک شده و Q0.0 روشن می‌شود. پس از آن اگر شستی غیرفعال شود، خط به‌صورت خط‌چین در می‌آید ولی پرانتز دو طرف دستور Set سبز رنگ است که نشان دهنده یک بودن خروجی است.



شکل ۱۵-۴۵ برنامه مثال ۱۵-۷ در حالت Online

بدیهی است این دستور در صورتی عمل می‌کندکه نتیجه عملیات منطقی (RLO) قبل از آن یک باشد. از این رو نمی‌توان آنرا به تنهایی به‌کار برد.



شکل ۱۵-۴۶ کاربرد غیر مجاز Set

فصل ۱۵

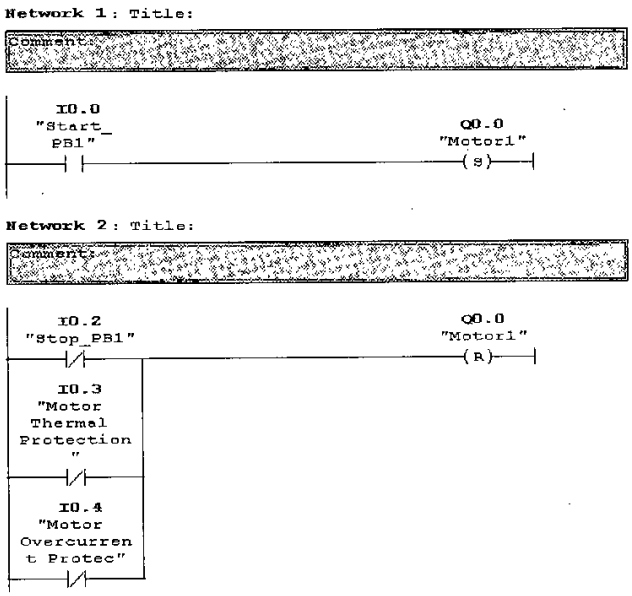
--(R)

Reset

این دستور برخلاف Set عمل می‌کند، یعنی منجر به صفر شدن بیت آدرس داده شده می‌گردد. برای فعال شدن آن کافیست یک لحظه نتیجه عملیات منطقی RLO قبل از آن یک شود.

Set و Reset معمولاً با یکدیگر برای کار کنترل به‌کار می‌روند. در بسیاری از کاربردها، برنامه‌نویسی با این دو دستور نسبت به استفاده از دستور Coil ارجحیت دارد.

مثال ۱۵-۸: برنامه مثال ۱۵-۴ را با استفاده از دستورات Set/Reset بازنویسی کنید.



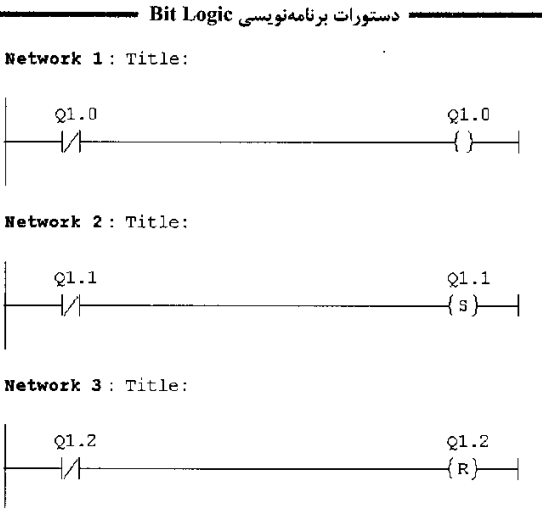
شکل ۱۵-۴۷ برنامه مثال ۱۵-۸

تذکر ۱: دقت شود که با توجه به اینکه شستی استپ و کنتاکت‌های حفاظتی همگی نرمال بسته هستند، بایستی برای دستور Reset از NC آنها استفاده کرد. اگر به اشتباه از NO استفاده شود خروجی همواره ری‌ست می‌گردد و روشن نخواهد شد.

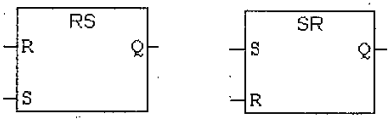
تذکر۲: در استفاده از دو دستور فوق لازم است توجه شود که هر کدام در Network پایین‌تر به‌کار روند اولویت خواهد داشت. به‌عنوان مثال اگر در برنامه ۱۵-۸ اپراتور به‌طور همزمان هر دو شستی استارت و استپ را فشار دهد موتور روشن نخواهد شد چون اولویت با Reset است که در Network2 قرار دارد. اگر دو Network فوق را جابه‌جا کنیم اولویت با Set خواهد بود و در صورت فعال‌شدن هر دو شستی موتور روشن خواهد گردید.

تمرین ۱۵-۳: نکته ذکر شده در تذکر ۲ را با سیمولاتور تست و نتایج را مقایسه کنید.

تمرین ۱۵-۴: در برنامه زیر خروجی‌های Q1.0 و Q1.1 و Q1.2 چه وضعیتی خواهند داشت.



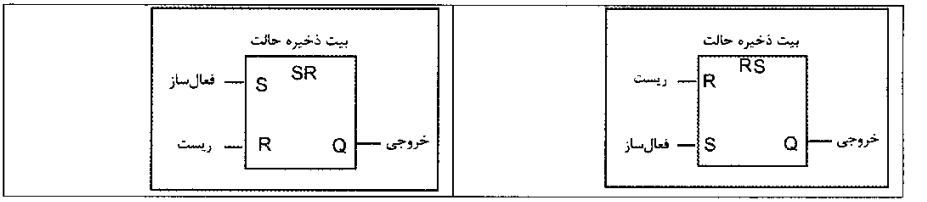
شکل ۱۵-۴۸ برنامه تمرین ۱۵-۴



Flip Flop

فیلیپ فلاپ المانی است که هر دو دستور Set و Reset را در خود دارد و به دو نوع RS و SR تقسیم می‌شود. برای استفاده از آن به نکات زیر دقت شود:

- هر فیلیپ‌فلاپ نیاز به یک حافظه یک بیتی دارد. آدرس این حافظه در بالای فیلیپ فلاپ نوشته می‌شود و معمولاً از آدرس Bit Memory برای آن استفاده می‌گردد.



شکل ۱۵-۴۹ فیلیپ‌فلاپ‌های SR و RS

- فیلیپ‌فلاپ المان خاص برنامه‌نویسی LAD/FBD است و در STL وجود ندارد.
- عملکرد دو فیلیپ‌فلاپ RS و SR مشابه و فقط در یک حالت با یکدیگر متفاوتند. اگر هر دو ورودی آنها همزمان یک شود، در نوع RS اولویت با Set و در نوع SR اولویت با Reset است یعنی همیشه پایه پایین اولویت دارد. جدول ۱۵-۳ این موضوع را بهتر نشان می‌دهد.

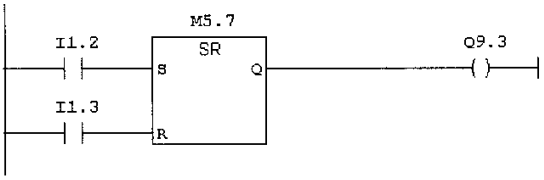
فصل ۱۵

جدول ۱۵-۳

خروجی Q در RS	خروجی Q در SR	ورودی R	ورودی S
1	1	0	1
0	0	1	0
وضعیت قبلی	وضعیت قبلی	0	0
1	**0**	1	1

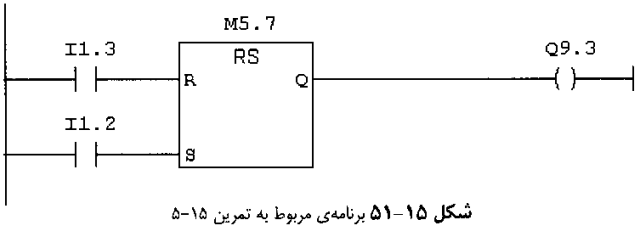
مثال۱۵-۹: برنامه نشان‌داده شده در شکل ۱۵-۵۰ را در نظر بگیرید.

در برنامه شکل ۱۵-۵۰ اگر ورودی I 1.2 فعال گردد، M5.7 و Q9.3 فعال می‌شوند. اگر I 1.2 صفر (غیرفعال) شود، M5.7 و Q9.3 همچنان روشن می‌مانند. حال اگر ورودی I1.3 فعال گردد، فلیپ- فلاپ ریست شده و M5.7 و Q9.3 خاموش می‌شوند.



شکل ۱۵-۵۰ برنامه‌ی نوشته شده جهت مثال ۱۵-۹

تمرین۱۵-۵: برنامه‌ی نوشته شده در شکل ۱۵-۵۱ را در نظر گرفته و تفاوت آنرا با برنامه‌ی مربوط به مثال ۱۵-۹ بیان نمایید.



شکل ۱۵-۵۱ برنامه‌ی مربوط به تمرین ۱۵-۵

مثال ۱۵ -۱۰ : کنترل موتور از دو نقطه

برنامه مثال را به‌صورتی کامل کنید که امکان استارت و استپ موتور از دو نقطه وجود داشته باشد؛ با توضیحات زیر:

- اگر سلکتور سوئیچ I0.7 در وضعیت On باشد، استارت و استپ موتور فقط از اتاق کنترل[1] امکان‌پذیر باشد.
- اگر سلکتور سوئیچ I0.7 در وضعیت Off باشد، استارت و استپ موتور فقط از محل[2] امکان‌پذیر باشد.
- کلید سلکتوری در هر وضعیتی باشد، با فعال‌شدن کلید قطع اضطراری چه از اتاق کنترل و چه از محل موتور خاموش شود.

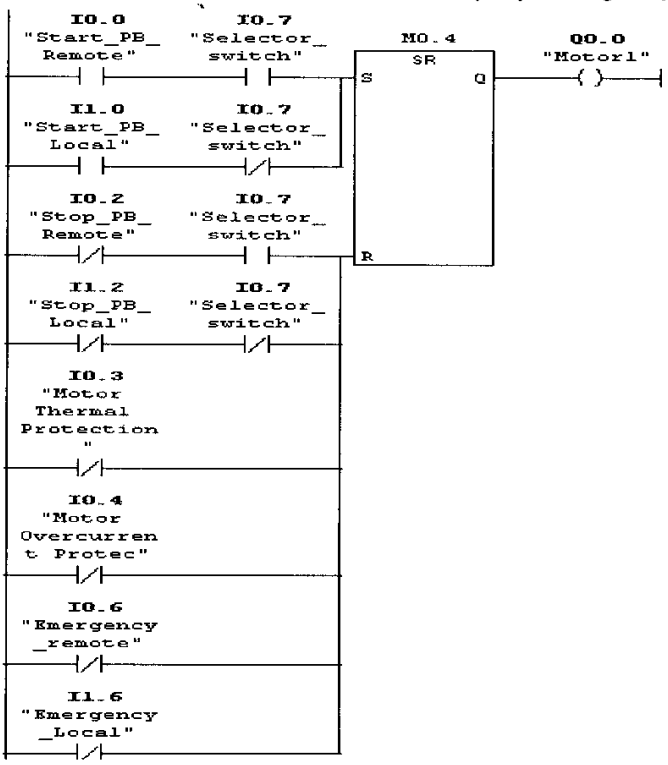
1. Remote
2. Local

جدول ۱۵-۴ لیست سیگنال‌ها را نشان می‌دهد.

جدول ۱۵-۴

آدرس	سمبل	حالت عادی
I0.0	Start_PB_Remote	باز
I0.2	Stop_PB_Remote	بسته
I0.7	Selector_Switch	باز
I0.6	Emergency_Remote	بسته
I1.0	Start_PB_Local	باز
I1.2	Stop_PB_Local	بسته
I1.6	Emergency_Local	بسته
I0.3	Thermal_Protection	بسته
I0.4	OverCurrent_protection	بسته

برنامه به‌صورت شکل ۱۵-۵۲ خواهد بود.



شکل ۱۵-۵۲ برنامه‌ی مربوط به مثال ۱۵-۱۰

مثال ۱۵- ۱۱: برنامه کنترل چپگرد راستگرد موتور

در یک پروسه صنعتی لازم است یک موتور را به‌صورت چپگرد/ راستگرد راه‌اندازی نمود. برای این کار نیاز به دو خروجی داریم، یکی خروجی چپگرد و دیگری خروجی راستگرد. همچنین به سه ورودی جهت راه‌اندازی به‌صورت راستگرد، جهت استپ نمودن موتور و جهت راه‌اندازی موتور به‌صورت چپگرد نیازمندیم. جدول ۱۵-۵ آدرس‌ها و سمبل‌های به‌کار رفته در این مثال را نشان می‌دهد.

جدول ۱۵-۵ آدرس‌ها و سمبل‌های به‌کار رفته در مثال ۱۵-۱۱

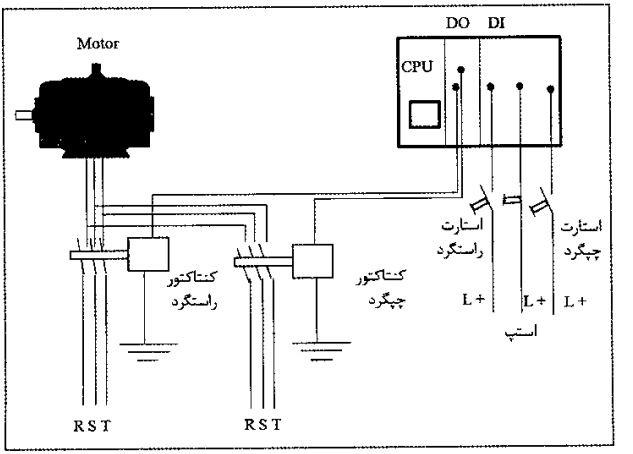
آدرس	سمبل	حالت عادی
I124.0	Start_R	باز
I124.1	Start_L	باز
I124.2	Stop	بسته
Q124.0	Motor_R	
Q124.1	Motor_L	

منطق کنترل

با فشرده‌شدن شستی استارت راستگرد، موتور به‌صورت راستگرد روشن شود (خروجی راستگرد فعال شود)، با فشرده‌شدن شستی استارت چپگرد، موتور به صورت چپگرد روشن شود (خروجی چپگرد روشن شود).

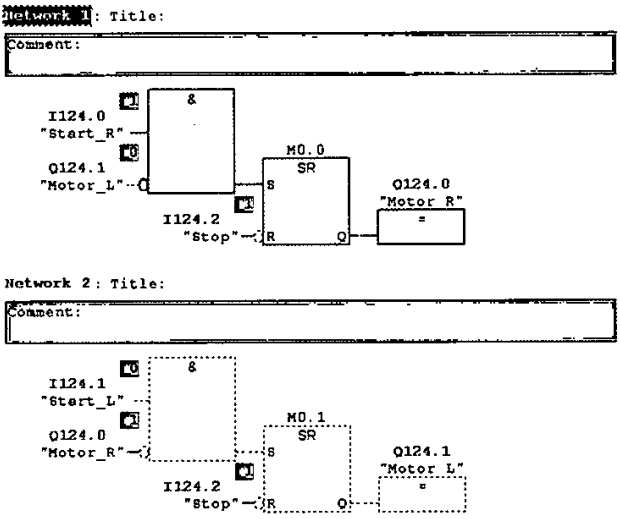
نکات ایمنی

هیچ‌گاه دو خروجی چپگرد و راستگرد با هم روشن نشوند. هر کدام از خروجی‌های راستگرد یا چپگرد که روشن بود، تا شستی استپ فشرده نشده است خروجی دیگر فعال نشود.



شکل ۱۵-۵۳ شماتیک اجزای مثال چپگرد راستگرد موتور

حل برنامه به زبان FBD



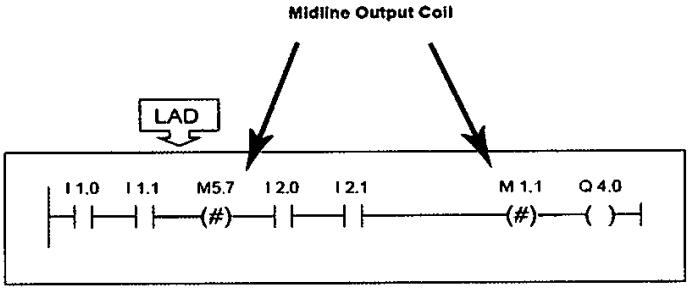
شکل ۱۵-۵۴ برنامه‌ی مثال چپگرد راستگرد موتور

تحلیل برنامه بر عهده خواننده می‌باشد.

--(#)--

Midline

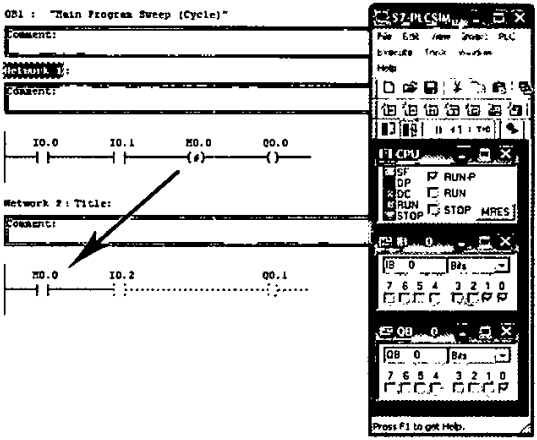
توسط این المان، می‌توان نتیجه عملیات منطقی قسمتی از برنامه را در یک متغیر حافظه[1] ذخیره کرد و از آن در سایر جاهای برنامه استفاده نمود.



شکل ۱۵-۵۵ استفاده از دستور Midline

1. Bit Memory

مثال ۱۵-۱۲: برنامه‌ای بنویسید که هرگاه دو ورودی I0.0 و I0.1 همزمان فعال شوند، خروجی Q0.0 روشن شود. همچنین اگر دو ورودی فوق و ورودی I0.2 همزمان فعال شوند، خروجی Q0.1 فعال شود.



شکل ۱۵-۵۶ برنامه‌ی مورد نظر جهت مثال ۱۵-۱۲

همانطور که در مثال مشاهده می‌شود، در Network اول، مقدار RLO بعد از ورودی‌های I0.0 و I0.1 در M0.0 ذخیره می‌شود. در Network2 از یک تیغه‌ی باز با آدرس M0.0 استفاده شده است که بعد از آن ورودی I0.2 قرار دارد. پس اگر I0.0 و I0.1 فعال شوند، M0.0 دارای مقدار یک منطقی شده و تیغه‌ی باز M0.0 در Network2 اجازه‌ی عبور RLO را می‌دهد. در این شرایط اگر I0.2 نیز فعال شود، خروجی Q0.1 فعال می‌شود. در واقع با استفاده از Midline، از نوشتن دوباره‌ی قسمتی از برنامه (I0.0 و I0.1) صرفنظر شده است.

تشخیص لبه RLO

|--(N)-- |--(P)--

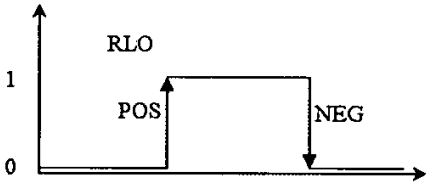
توسط این دو دستور در وسط برنامه می‌توان تغییر وضعیت نتیجه عملیات منطقی را تشخیص داد. یعنی می‌توان فهمید که آیا RLO از صفر به یک جابه‌جا شده (لبه مثبت) یا RLO از یک به صفر بازگشته (لبه منفی).

لبه‌ی بالارونده‌ی RLO: اگر مقدار RLO برابر صفر منطقی باشد و به مقدار یک منطقی تغییر وضعیت دهد، یک لبه‌ی بالارونده[1] به‌وجود می‌آید.

لبه‌ی پایین رونده‌ی RLO: اگر مقدار RLO برابر یک منطقی باشد و به مقدار صفر منطقی تغییر وضعیت دهد، یک لبه‌ی پایین‌رونده[2] به‌وجود می‌آید.

شکل ۱۵-۵۷ لبه‌ی بالارونده و پایین‌رونده‌ی RLO را نشان می‌دهد.

1. Positive Edge
2. Negative Edge



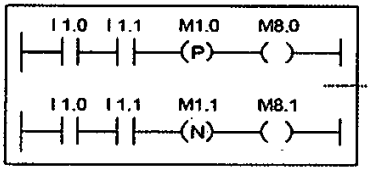
شکل۱۵-۵۷ لبه‌ی بالارونده (POS) و پایین‌رونده (NEG) مربوط به RLO

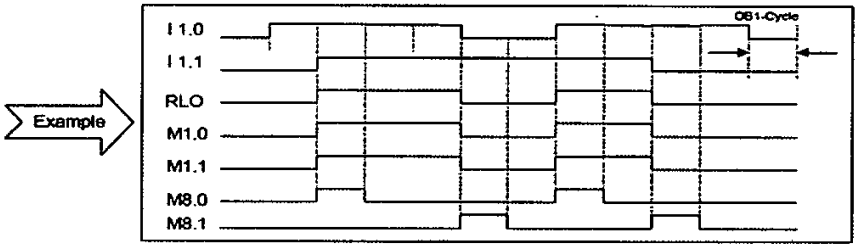
برای تشخیص لبه RLO دو دستور معرفی شده است:

دستور (P)– یا Positive RLO Edge detection: توسط این دستور می‌توان لبه‌ی بالارونده‌ی RLO را تشخیص داد. در اینصورت خروجی این دستور به‌مدت یک سیکل‌اسکن فعال شده و سپس غیرفعال می‌گردد.

دستور (N)– یا Negative RLO Edge Detection: توسط این دستور می‌توان لبه‌ی پایین‌رونده‌ی RLO را تشخیص داد. در اینصورت خروجی این دستور به‌مدت یک سیکل‌اسکن فعال شده و سپس غیرفعال می‌گردد.

شکل ۱۵-۵۸ عملکرد این دو دستور را نشان می‌دهد.

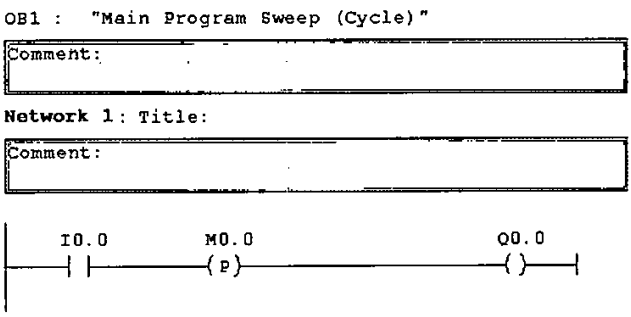




شکل ۱۵-۵۸ دستورات تشخیص لبه‌ی RLO (P و N) و چگونگی عملکرد آنها

در برخی کاربردها نیاز به استفاده دستور لبه وجود دارد. به‌عنوان مثال اگر سیستمی توسط شستی فرمان استارت و استپ بگیرد و برنامه آن با فیلیپ فلاپ RS نوشته شده باشد اگر شستی استارت گیر کند و به صفر بر نگردد مشکل پیش می‌آید و سیستم استپ نمی‌شود زیرا اولویت با Set است. برای رفع این مشکل از دستور لبه استفاده می‌کنیم یعنی Set فقط در لحظه تغییر شستی فعال می‌شود و پس از آن حتی اگر شستی یک بماند ورودی Set یک نخواهد بود

مثال ۱۵-۱۳: برای فهم بهتر موضوع، برنامه‌ی نشان‌داده شده در شکل ۱۵-۵۹ را در نظر بگیرید.



شکل ۱۵-۵۹ برنامه مثال ۱۵-۱۳

در این برنامه از یک تیغه‌ی باز با آدرس I0.0 و یک دستور تشخیص لبه‌ی بالارونده‌ی RLO (دستور P) استفاده شده است. بعد از این دو المان، خروجی Q0.0 قرار دارد. جدول ۱۵-۶ به بررسی حالت‌های مختلف پیش‌آمده برای این متغیرها در سیکل‌های مختلف کاری CPU پرداخته است.

جدول ۱۵-۶

متغیر	سیکل کاری n	سیکل کاری n+1	سیکل کاری n+2
I0.0	0	1	1
M0.0 (RLO)	0	1	1
لبه‌ی بالارونده	خیر	بله	خیر
Q0.0	0	1	0

همانطور که در جدول ۱۵-۶ مشخص است:

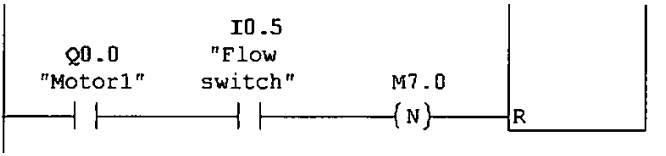
- در سیکل کاری n (مثلاً سیکل کاری دهم) مقدار I0.0 برابر صفر منطقی می‌باشد، بنابراین مقدار RLO که در M0.0 ذخیره می‌شود، برابر صفر منطقی می‌باشد. در این حالت لبه‌ی بالارونده‌ای به‌وجود نیامده و خروجی Q0.0 نیز قطع می‌باشد.
- در سیکل کاری n+1 (مثلاً سیکل کاری یازدهم) مقدار I0.0 برابر یک منطقی می‌شود، از اینرو مقدار RLO که در M0.0 ذخیره می‌شود نیز برابر یک منطقی می‌شود. چون مقدار RLO که در سیکل قبل صفر بوده، در این سیکل به مقدار یک تغییر وضعیت داده است، یک لبه‌ی بالارونده ایجاد می‌شود. در نتیجه خروجی Q0.0 وصل (فعال) می‌شود.
- در سیکل کاری n+2 (مثلاً سیکل کاری دوازدهم) مقدار I0.0 و M0.0 همچنان برابر یک منطقی است. در این حالت چون مقدار قبلی RLO (که در M0.0 ذخیره می‌شود) برابر مقدار یک منطقی بوده و مقدار جدید آن نیز یک است، لذا لبه‌ای به‌وجود نیامده و خروجی Q0.0 قطع می‌گردد.

مثال ۱۵-۱۴: در مثال ۱۵-۴ اگر موتور مربوط به یک پمپ باشد که سیالی را به داخل یک مخزن می‌فرستد و در مسیر سیال فلوسوئیچی تعبیه شده که کنتاکت آن در حالت عادی باز و با برقراری فلو بسته می‌شود، برنامه‌ای بنویسید که اگر در حین کار فلو صفر شد موتور به‌طور خودکار Off شود.

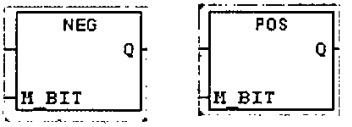
بدیهی است بایستی از کنتاکت فلوسوئیچ به پایه Reset فیلیپ‌فلاپ فرمان داد ولی نمی‌توان کنتاکت فلوسوئیچ را مستقیماً به این پایه داد، زیرا:

اگر کنتاکت NO را به ری‌ست وصل کنیم وقتی فلو برقرار شد پمپ از کار می‌افتد که درست نیست. اگر کنتاکت NC را به ری‌ست وصل کنیم در لحظه اول که فلو صفر است پایه ری‌ست را یک می‌کند و اجازه استارت نمی‌دهد.

پس برای رفع مشکل فوق از لبه پایین رونده RLO مانند شکل ۱۵-۶۰ استفاده می‌کنیم. مفهوم آن این است که وقتی پمپ روشن و فلو برقرار است، اگر فلو صفر شد پمپ را خاموش کن. این لاجیک با سایر لاجیک‌های متصل به R در مثال ۱۵-۴ موازی می‌شود.



شکل۱۵-۶۰ برنامه مثال ۱۵- ۱۴



تشخیص لبه سیگنال

دستورات تشخیص لبه‌ی سیگنال

علاوه بر دستوراتی که وجود لبه در RLO را تشخیص می‌دهند، دستوراتی وجود دارد که وجود لبه در سیگنال (مقدار) یک متغیر را تشخیص می‌دهند. می‌دانیم که هرگاه وضعیت یک متغیر از نوع Bool از صفر منطقی به یک منطقی تغییر نماید، یک لبه‌ی بالارونده به‌وجود می‌آید و هرگاه مقدار آن از یک منطقی به صفر منطقی تغییر نماید، لبه‌ی پایین رونده بوجود می‌آید.

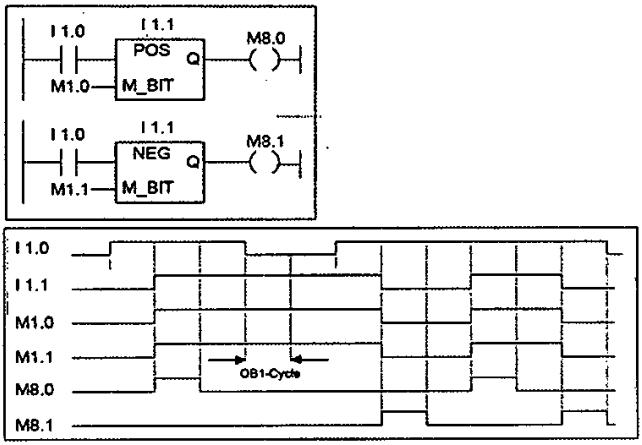
Address Positive Edge Detection (POS): این دستور وجود لبه‌ی بالارونده‌ی سیگنال یک آدرس از نوع Bool را تشخیص داده و به‌مدت یک سیکل‌اسکن خروجی آن فعال می‌شود.

Address Negative Edge Detection (NEG): این دستور وجود لبه‌ی پایین‌رونده‌ی سیگنال یک آدرس از نوع Bool را تشخیص داده و به‌مدت یک سیکل‌اسکن خروجی آن فعال می‌شود.

نکته‌ای که باید به آن توجه نمود این است که در دستورات تشخیص لبه‌ی RLO، وجود تغییر در RLO حس می‌شود. تغییر در RLO ممکن است نتیجه‌ی چند تیغه‌ی باز و بسته یا چند دستور منطقی باشد. اما در دستورات تشخیص لبه‌ی سیگنال، صرفاً تغییر در وضعیت یک متغیر بررسی می‌شود.

جهت درک بهتر موضوع به شکل ۱۵-۶۱ توجه نمایید.

فصل ۱۵



شکل۱۵-۶۱ دستورات تشخیص لبه‌ی سیگنال مربوط به یک آدرس

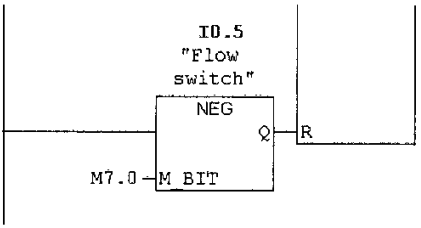
همانطور که در شکل ۱۵-۶۱ مشخص است، دستور POS و NEG دارای یک پایه به‌نام M_BIT است که مقدار وضعیت قبلی آدرسی که در بالای دستور نوشته می‌شود (در این مثال آدرس I1.1)، در آن ذخیره می‌شود. این تابع (دستور POS یا NEG) در هر سیکل مقدار وضعیت جاری آدرس را با وضعیت ذخیره شده در M_BIT (که در واقع وضعیت یا مقدار قبلی آدرس می‌باشد) مطابقت داده و اگر تغییری حاصل شود، وجود لبه تشخیص داده می‌شود.

در این دستورات (POS و NEG) در صورت تشخیص وجود لبه، به مدت یک سیکل کاری، خروجی تابع فعال می‌گردد.

تفاوت بین دستورات تشخیص لبه RLO و دستورات تشخیص سیگنال:

- دستورات (N)– و (P)– را نمی‌توان در ابتدای خط LAD که هنوز لاجیکی قبل از آن وجود ندارد به‌کار برد، درحالی که از POS و NEG می‌توان از ابتدا استفاده کرد.
- برای تشخیص لبه در وسط برنامه نمی‌توان از POS و NEG استفاده کرد و بایستی (N)– و (P)– را به‌کار برد.

مثال ۱۵- ۱۵: تشخیص لبه فلو سوئیچ مثال ۱۵-۱۴را می‌توان با دستور زیر نیز انجام داد:



شکل۱۵-۶۲ برنامه مثال ۱۵- ۱۵

--(SAVE)

(SAVE) Save RLO into BR Memory

توسط این المان می‌توان مقدار فعلی RLO را در بیت دیگری از رجیسترهای حافظه که BR نام دارد ذخیره کرد بیت‌های Status Word در کتاب سطح تکمیلی تشریح می‌شود. این دستور در برنامه‌نویسی پیشرفته استفاده می‌شود. یکی از کاربردهای آن برای برگرداندن کد خطا در اجرای فانکشن‌هاست که در کتاب سطح پیشرفته بحث می‌گردد.

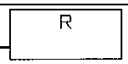
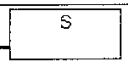
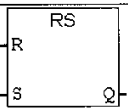
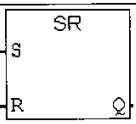
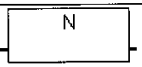
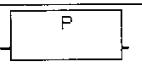
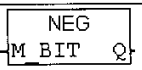
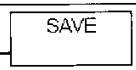
۱۵-۳ دستورات Bit Logic در زبان FBD

این المان‌ها که در پنجره Program Element شکل ۱۵-۲ نمایش داده شده‌اند، در جدول ۱۵-۷ معرفی می‌شوند.

جدول۱۵-۷

شکل	عنوان	کاربرد
>=1	OR Logic Operation	OR دو یا چند بیت
&	AND Logic Operation	AND دو یا چند بیت
XOR	XOR Logic Operation	Exclusive OR دو یا چند بیت
--|	Insert Binary Input	اضافه کردن ورودی در گیت‌های AND,OR,XOR
-O|	Negate Binary Input	عکس کردن وضعیت بیت ورودی به المان‌ها
=	Assign	اختصاص نتیجه لاجیک به یک بیت
#	Midline Output	ذخیره سازی نتایج میان برنامه

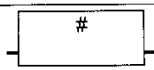
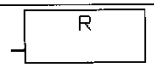
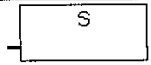
فصل ۱۵

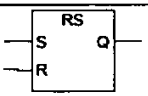
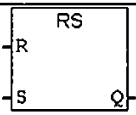
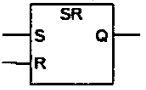
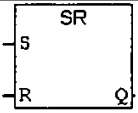
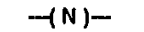
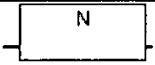
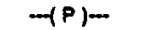
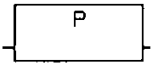
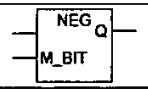
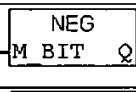
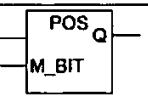
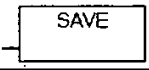
ری‌ست کردن یک بیت	Reset	
ست کردن یک بیت	Set	
ست و ری‌ست کردن یک بیت با اولویت ست	RS Flip Flop	
ست و ری‌ست کردن یک بیت با اولویت ری‌ست	SR Flip Flop	
تشخیص لبه منفی نتیجه عملیات لاجیک	Negative Edge Detection	
تشخیص لبه مثبت نتیجه عملیات لاجیک	Positive Edge Detection	
تشخیص لبه منفی سیگنال	Negative Edge Detection	
تشخیص لبه مثبت سیگنال	Positive Edge Detection	
ذخیره کردن RLO در یک بیت حافظه	Save RLO	

در بین المان‌های فوق مواردی هستند که شکل و عملکرد آنها با المان‌های LAD مشابه است ولی برخی دیگر به‌عنوان المان جدید مطرح می‌شوند.

المان‌های FBD که با LAD مشابه هستند

جدول۱۵-۸

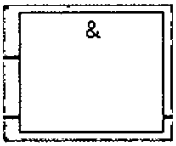
LAD	FBD
---( )	
---(#)---	
---(R)	
---(S)	

عملکرد این المان‌ها نیاز به توضیح اضافی ندارد و آنچه در بحث دستورات LAD ذکر شده برای این دستورات نیز کفایت می‌کند.

المان‌های FBD که با LAD متفاوت هستند.

در زبان FBD برای استفاده از آدرس‌ها و پیاده‌سازی منطق کنترل، کنتاکت‌های NO و NC به شکل مستقل شبیه LAD وجود ندارد بلکه آدرس‌ها به‌طور مستقیم به پایه گیت‌هایی مانند AND و OR و .... داده می‌شود و منطق کنترل توسط این گیت‌ها پیاده‌سازی می‌شود. به‌عنوان مثال سری کردن دو کنتاکت در زبان LAD توسط المان AND در زبان FBD پیاده‌سازی می‌شود.



AND

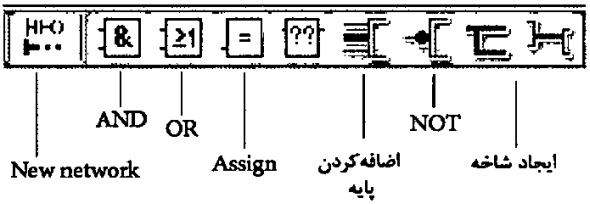
توسط این دستور می‌توان چند متغیر را با یکدیگر ترکیب AND منطقی نمود. در ترکیب AND منطقی، در صورتی خروجی فعال می‌گردد که همه‌ی آدرس‌هایی که با یکدیگر AND شده‌اند، دارای مقدار یک منطقی شوند. جدول ۱۵-۹، جدول صحت دستور AND را نشان می‌دهد.

جدول ۱۵-۹ جدول صحت دستور AND

ورودی اول	ورودی دوم	خروجی
0	0	0
0	1	0
1	0	0
1	1	1

همانطور که در جدول ۱۵-۹ مشخص است، زمانی که هر دو متغیر ورودی تابع AND دارای مقدار یک منطقی گردند، خروجی دارای مقدار یک منطقی می‌گردد. این دستور در زبان LAD، معادل مستقیم ندارد ولی با استفاده از دو تیغه‌ی باز که با یکدیگر سری شده‌اند، می‌توان آنرا معادل‌سازی نمود.

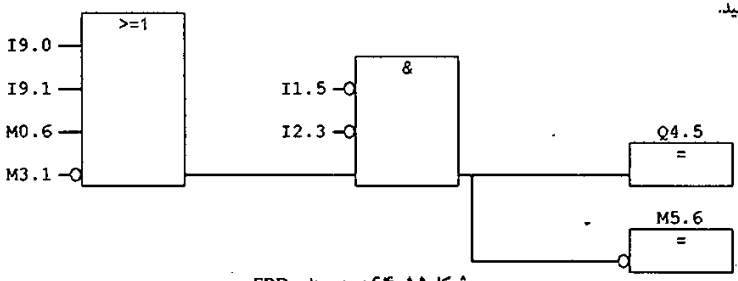
اگر View (نمایش) برنامه روی حالت FBD قرار داشته باشد، در Toolbar آیکن‌های نشان‌داده شده در شکل ۱۵-۶۰ نشان داده می‌شود. توسط این آیکن‌ها می‌توان دستورات AND، OR و Assign ایجاد نمود.



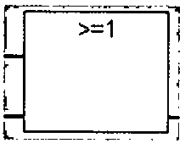
شکل۱۵-۶۳ دستورات FBD موجود در Toolbar

دستور AND به‌طور پیش‌فرض دارای دو ورودی است که امکان افزایش یا کاهش آن نیز وجود دارد. برای افزایش آن با کلیک روی آیکن اضافه کردن پایه از Toolbar یا آیکن |-- از پنجره المان‌ها می‌توان ورودی‌ها را افزایش داد. این کار برای سایر گیت‌ها مانند OR و XOR نیز امکان‌پذیر است.

برای معکوس کردن یک سیگنال یا معکوس کردن نتیجه عملیات منطقی از المان دایره شکل که در نوار ابزار و در پنجره المان‌ها موجود است استفاده می‌شود. معکوس کردن در ورودی گیت‌ها و المان‌ها صورت می‌گیرد. به شکل ۱۵-۶۴ توجه کنید.



شکل۱۵-۶۴ نمونه برنامه FBD



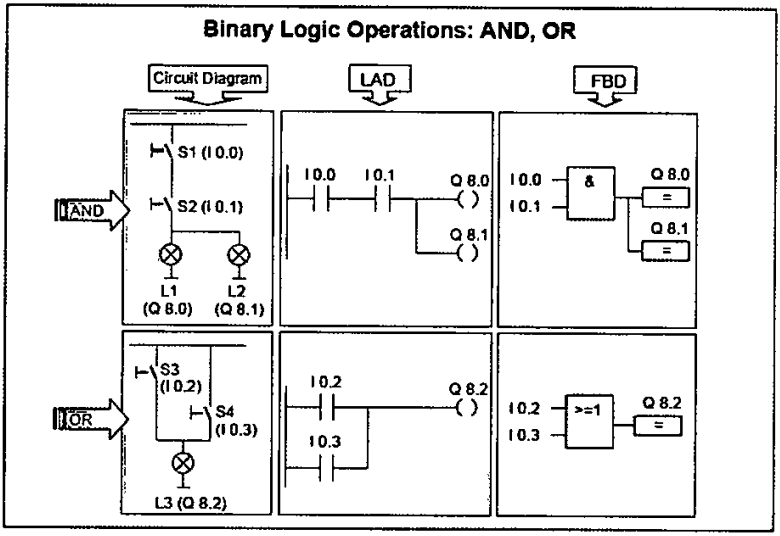
OR

توسط این دستور می‌توان چند متغیر را با یکدیگر ترکیب OR منطقی نمود. در حالت پیش‌فرض، این دستور دارای دو پایه‌ی ورودی است که با استفاده از المان نشان‌داده شده در شکل ۱۵-۶۳ می‌توان پایه‌های جدیدی به‌آن اضافه نمود. در ترکیب OR، اگر فقط یکی از متغیرهای ORشده دارای مقدار یک منطقی گردد، خروجی این دستور فعال می‌شود. جدول ۱۵-۱۰، جدول صحت این دستور را نشان می‌دهد.

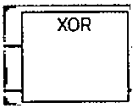
جدول ۱۵-۱۰ جدول صحت دستور OR

ورودی اول	ورودی دوم	خروجی
0	0	0
0	1	1
1	0	1
1	1	1

این دستور نیز در زبان LAD، معادل مستقیم ندارد ولی با استفاده از دو تیغه‌ی موازی می‌توان آنرا معادل‌سازی نمود.



شکل ۱۵-۶۵ دستورات AND، OR، Assign و معادل آنها در زبان LAD



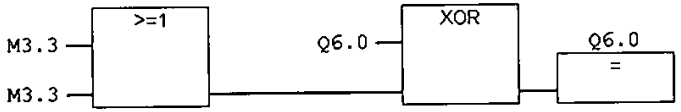
XOR

این دستور، ترکیب منطقی Exclusive OR را بین دو یا چند متغیر انجام می‌دهد. در ترکیب منطقی XOR در حالتی که دو ورودی حالت غیر مشابه داشته باشند، خروجی فعال می‌گردد. جدول ۱۵-۱۱، جدول صحت این دستور را نشان می‌دهد.

جدول ۱۵-۱۱ جدول صحت دستور XOR

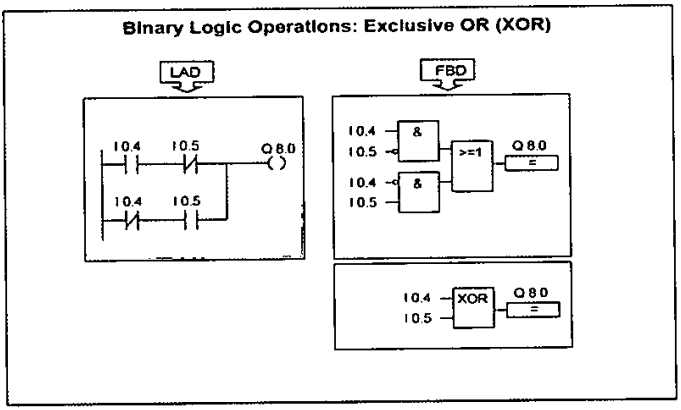
ورودی اول	ورودی دوم	خروجی
0	0	0
0	1	1
1	0	1
1	1	0

مثال ۱۵-۱۶: در برنامه زیر خروجی در یک سیکل روشن و در سیکل بعدی خاموش است. تحلیل به عهده خواننده است.



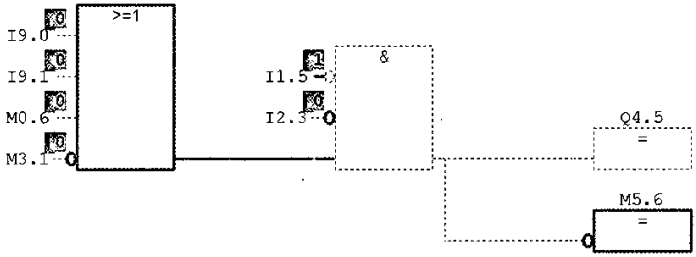
شکل۱۵-۶۶ برنامه مثال ۱۵- ۱۶

دستور XOR معادل LAD ندارد ولی با چند تیغه‌ی باز و بسته می‌توان آنرا مانند شکل ۱۵-۶۷ معادل‌سازی نمود. همچنین امکان معادل‌سازی این دستور در زبان FBD، توسط دستورات AND و OR نیز وجود دارد.



شکل۱۵-۶۷ دستور XOR و معادل آن در زبان LAD

وقتی برنامه نوشته شده به زبان FBD ذخیره و دانلود شود، می‌توان با آیکن عینک شکل نوارابزار آنرا Monitor کرد. در این حالت پنجره نمایش به‌صورت شکل ۱۵-۶۸ است. هرجا که نتیجه عملیات منطقی (RLO) یک است با خط به رنگ سبز و هر جا که نتیجه صفر است به‌صورت خط چین ظاهر می‌شود.

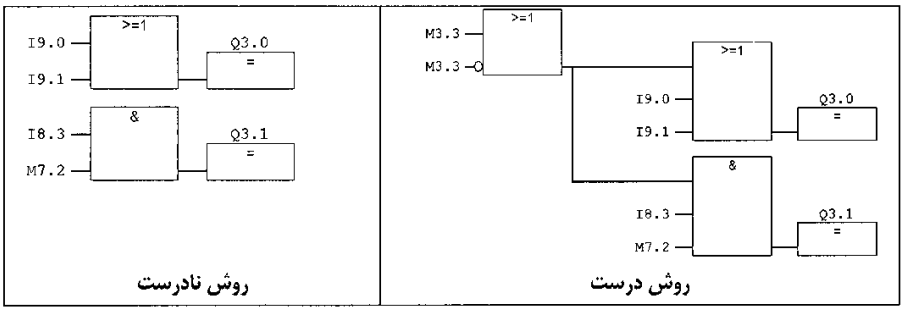


شکل۱۵-۶۸ برنامه FBD در حالت Online

تذکر ۱: در زبان FBD شکل دستور NOT به‌صورت یک دایره‌ی توخالی است که می‌توان آنرا به هر پایه‌ی دلخواه از دستورات منطقی اضافه نمود.

تذکر۲: اکثر دستورات LAD و FBD به یکدیگر قابل تبدیل هستند و هر دوی آنها به زبان STL قابل تبدیل و نمایش هستند. برای این منظور کافیست از منوی View زبان مورد نظر را انتخاب کنید.

تذکر ۳: شبیه نکته‌ای که در مورد LAD بیان شد در FBD نمی‌توان لاجیک‌های مستقل از هم را در یک Network نوشت (شکل ۱۵-۶۹ سمت چپ) در اینحالت امکان ذخیره و دانلود وجود ندارد. ولی شبیه شکل سمت راست می‌توان با ساخت سیگنال همیشه یک و اتصال به لاجیک‌های مستقل آنها را در یک Network جای داد.



شکل۱۵-۶۹ مقایسه دو روش برنامه‌نویسی FBD در یک Network 1

فصل ۱۵

پیاده‌سازی منطق فرآیند بر اساس شرح نوشتاری اینترلاکینگ

در برخی از کاربردها پیاده‌سازی منطق کنترل از زبان نوشتاری به زبان FBD ساده‌تر و ملموس‌تر از زبان LAD است.

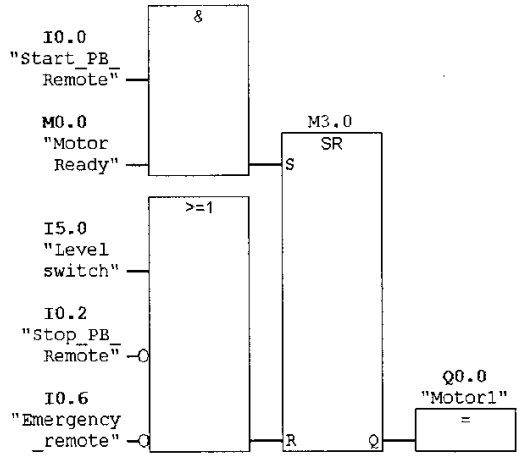
کافیست لاجیک کنترلی را به‌صورت نوشتاری بیان کرده سپس هر جا از حرف " و" استفاده شده آنرا AND و هر جا از حرف یا استفاده شده آنرا OR نمود.

مثال ۱۵-۱۷

در فرآیندی لازم است:

- پمپ استارت شود اگر، سیگنال آمادگی برسد **و** اپراتور شستی استارت را فشار دهد.
- پمپ استپ شود اگر، اپراتور شستی استپ را فشار دهد **یا** سوئیچ سطح مخزن فعال شود **یا** کلید اضطراری عمل کند.

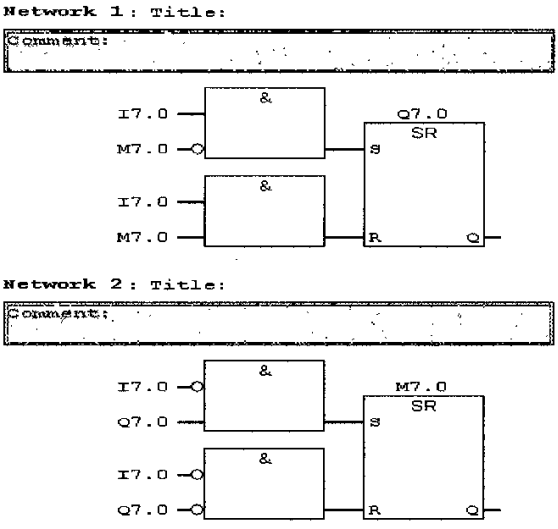
پیاده‌سازی برنامه به‌صورت زیر است. بلاک OR معادل حرف **یا** و بلاک AND معادل حرف **و** هستند.



شکل۱۵-۷۰ برنامه مثال ۱۵- ۱۷

تمرین ۱۵-۵: مثال ۱۵-۱۰ که مربوط به کنترل موتور از دو نقطه بود را به زبان نوشتاری بیان کرده و با FBD باز نویسی کنید.

مثال۱۵- ۱۸: به زبان FBD برنامه‌ای بنویسید که با فعال‌شدن شستی I7.0 خروجی Q7.0 روشن شود و با فشار دادن مجدد همان شستی خروجی فوق خاموش شود. تحلیل به عهده خواننده است.



شکل۱۵-۷۱ برنامه مثال ۱۵- ۱۸ خاموش و روشن کردن یک سیستم فقط با یک شستی

۱۵-۴ تمرین

تمرین۱: برنامه مربوط به راه‌اندازی موتور سه‌فاز به‌صورت چپگرد، راستگرد مثال ۱۵-۱۱ را به‌صورت تعویض دور سریع طراحی نمایید.

در اینصورت اگر موتور در هرکدام از جهات چپ یا راست در حال چرخش باشد و کلید دور معکوس زده شود، موتور متوقف شده و سریعاً با دور عکس شروع به‌کار نماید.

فصل ۱۵

فصل ۱۶

برنامه‌نویسی با دستورات تایمر

۱۶-۱ مقدمه
۱۶-۲ ورودی و خروجی‌های مشترک تایمرها
۱۶-۳ انواع تایمرها و عملکرد آنها
۱۶-۴ دستورات بیتی تایمر
۱۶-۵ بررسی ماندگاری تایمرها
۱۶-۶ بررسی خروجی های BCD و BI تایمر
۱۶-۷ تمرین

در این فصل با نحوه برنامه‌نویسی و نکات مربوط به تایمرها در زبان‌های LAD و FBD آشنا می‌شوید.

چکیده مطالب

- تایمرها با حرف T آدرس‌دهی شده واز T0 شروع می‌گردند.
- می توان تایمر را به‌صورت ماندگار تعریف کرد تا در صورت قطع تغذیه پاک نشوند.
- زمان تایمر حداقل ۱۰ میلی‌ثانیه و حداکثر ۲ ساعت و ۴۶ دقیقه و ۳۰ ثانیه است.
- زمان به فرمت S5Time به تایمر داده می‌شود.
- تایمرها به پنج نوع تقسیم می‌شوند:
- *تایمر تاخیر در وصل* که وقتی ورودی آن فعال شد ، خروجی را با تاخیر یک می‌کند.
- *تایمر تاخیر در وصل ماندگار* مشابه تاخیر در وصل است با این تفاوت که خروجی آن یک می ماند تا ری‌ست شود.
- *تایمر تاخیر در قطع* پس از غیر فعال شدن ورودی ، خروجی را با تأخیر صفر می‌کند.
- *تایمر پالس* پس از فعال شدن ورودی ، خروجی را به مدت مشخصی یک نگه می‌دارد سپس آن را صفر می‌کند.
- *تایمر پالس ادامه دهنده* رفتاری شبیه تایمر پالس دارد با این تفاوت که اگر ورودی آن قبل از اتمام کار از یک به صفر برگردد به کار خود ادامه می دهد.

۱۶-۱ مقدمه

تایمرها یکی از دستورات پرکاربرد در برنامه‌نویسی PLC هستند. یکی از مزایای PLC حذف تایمرهای سخت‌افزاری و جایگزینی آنها با تایمرهای نرم‌افزاری است.

در برنامه‌نویسی Step7 تایمرهای متنوعی وجود دارد که آنها را می‌توان به سه دسته زیر تقسیم‌بندی نمود:

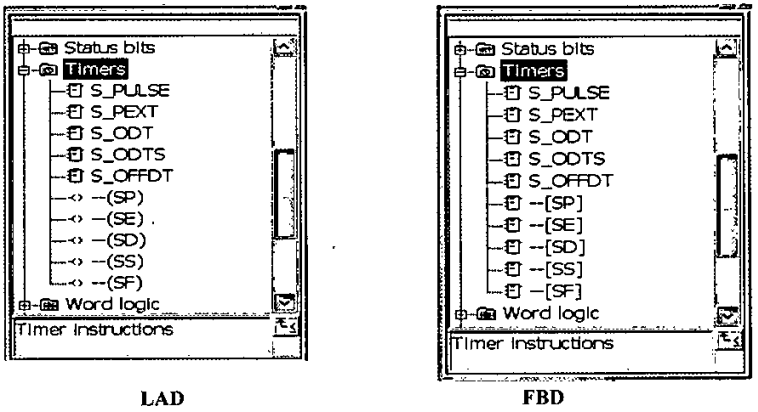
تایمرهای پایه: این تایمرها در پنجره Program Element به‌صورت یک المان پایه قابل استفاده هستند.

تایمرهای IEC: این تایمرها به‌صورت SFB هستند و در کتابخانه نرم‌افزار موجود هستند.

تایمرهای کاربر: کاربر می‌تواند از طریق برنامه‌نویسی تایمر دلخواهی ایجاد نماید.

در این فصل به تایمرهای نوع اول یعنی تایمرهای پایه می‌پردازیم و نحوه برنامه‌نویسی آنها را به زبان‌های LAD و FBD تشریح می‌کنیم. زبان STL و دو نوع دیگر تایمر در کتاب‌های سطح پیشرفته و تکمیلی تشریح شده‌اند.

اگر در پنجره Programm Element بخش تایمرها را باز کنیم زیر مجموعه آنها را به‌صورت شکل ۱۶-۱ خواهیم دید.



LAD **FBD**

شکل ۱۶-۱ محل قرارگیری دستورات Timers در پنجره Program Element

همانطور که دیده می‌شود، المان‌های برنامه‌نویسی تایمر برای هر دو زبان LAD و FBD یکسان هستند و نیازی نیست که به‌صورت مجزا مورد بحث قرار گیرند. در این فصل ما به بررسی این المان‌ها پرداخته و برخی مثال‌ها را به زبان LAD و برخی دیگر را به زبان FBD تشریح می‌کنیم.

قبل از وارد شدن به بحث لازم است برخی نکات که در فصل‌های قبلی نیز به آنها اشاره شده را مرور کنیم:

- تایمرها در حافظه System Memory قرار می‌گیرند.
- هر تایمر در حافظه سیستم دارای یک شمارهٔ منحصر بفرد است.
- شماره تایمرها از T0 شروع می‌شود.
- حد ماکزیمم برای شماره تایمر به ویژگی‌های CPU بستگی دارد. در بسیاری از CPUهای 300 تعداد تایمرها ۲۵۶ عدد است یعنی از T0 تا T255.

- شماره خارج از رنج در S7-400 و S7-300 قدیمی می‌تواند منجر به توقف CPU و روشن شدن چراغ فالت آن شود. در S7-300 جدید این خطا فقط چراغ فالت را روشن می‌کند.
- تایمرها را می‌توان به‌صورت ماندگار تعریف کرد. به‌طور پیش‌فرض هیچ تایمری در ناحیه Retentive Memory قرار ندارد. می‌توان در ویژگی‌های CPU تایمرهای مورد نظر را به حافظه ماندگار انتقال داد. این تایمرها در صورت قطع تغذیه پاک نمی‌شوند و با وصل تغذیه از جایی که زمان‌سنجی آنها متوقف شده ادامه می‌دهند.
- تایمرهای ماندگار حتماً باید از T0 شروع شوند سپس تعداد آنها مشخص گردد. امکان اینکه فقط یک تایمر با شماره غیر از T0 ماندگار شود وجود ندارد.
- هر تایمر نیاز به یک ورودی فعال‌کننده دارد و تا این ورودی یک نشود تایمر به‌کار نمی‌افتد.
- زمان تایمر بایستی به فرمت S5Time داده شود.
- هر تایمر در حافظه سیستم یک حافظه ۱۶ بیتی اشغال می‌کند که با به‌کار افتادن تایمر محتویات این حافظه مدام تغییر می‌کند .
- در حافظه ۱۶ بیتی فوق ۱۲ بیت برای زمان سنجی تایمر در نظر گرفته شده است که به‌صورت BCD است و می‌تواند از صفر شروع شده و به ۹۹۹ ختم شود.
- در حافظه ۱۶ بیتی فوق، دو بیت برای پله‌های زمان در نظر گرفته شده است. پله زمانی می‌تواند معرف 0.1 s یا 1 s یا 10 S باشد.
- با وارد کردن زمان به فرمت S5Time انتخاب پله زمانی به‌طور خودکار صورت می‌گیرد. جدول ۱۶-۱ این موضوع را نشان می‌دهد.

جدول ۱۶-۱

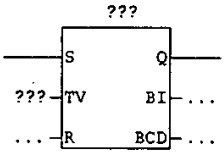
دو بیت معرف پله زمانی	Resolution	S5Time# Time
00	0.01 second	10MS to 9S_990MS
01	0.1 second	100MS to 1M_39S_900MS
10	1 second	1S to 16M_39S
11	10 seconds	10S to 2H_46M_30S

ماکزیمم زمان قابل تنظیم برای تایمر وقتی است که پله زمان ۱۰ ثانیه و عدد تایمر ۹۹۹ باشد که معادل ۹۹۹۰ ثانیه یعنی دو ساعت و چهل و شش دقیقه است.

۱۶-۲ ورودی و خروجی‌های مشترک تایمرها

همه‌ی انواع تایمرها دارای پایه‌های یکسانی می‌باشند، این پایه‌ها عبارتند از:

S: فعال‌ساز تایمر می‌باشد و آدرس بیتی قبول می‌کند. هرگاه فعال شود، تایمر به‌کار می‌افتد.



شکل ۱۶-۲ ورودی و خروجی های تایمرها

R: غیرفعال‌ساز (ری‌ست) تایمر بوده و آدرس بیتی قبول می‌کند. هرگاه فعال شود، تایمر از کار می‌افتد.

TV: زمان تایمر به فرم S5_TIME در این قسمت وارد می‌شود. در واقع تایمر بر اساس این زمان به ایفای نقش می‌پردازد. این ورودی می‌تواند به‌صورت متغیر حافظه باشد یا به‌صورت زمان مشخص در فرمت S5Time وارد شود. اختصاص متغیر به این ورودی را در فصل‌های بعد توضیح خواهیم داد.

Q: خروجی تایمر که قرار است قطع یا وصل شود، در این قسمت وارد می‌شود.

BI: زمان باقیمانده‌ی کار تایمر را به فرم Integer نشان می‌دهد.

BCD: زمان باقیمانده‌ی کار تایمر را به فرم BCD نشان می‌دهد.

تذکر۱: در استفاده از تایمر اختصاص آدرس به خروجی‌های BI وBCD و ورودی R الزامی نیست ولی سایر موارد بایستی مقداردهی شوند در غیر اینصورت امکان ذخیره و دانلود وجود ندارد.

تذکر۲: در اختصاص زمان بایستی قبل از زمان کلمه S5TIME# یا S5T# را به‌کار برد، مانند S5TIME#10s یا S5T#10s.

۱۶-۳ انواع تایمرها و عملکرد آنها

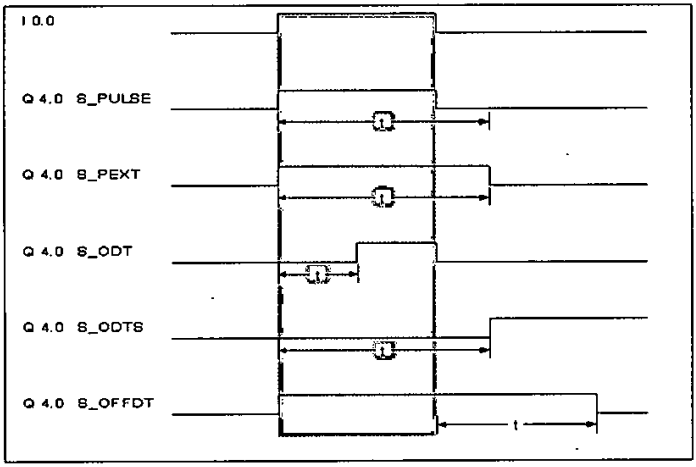
به‌طور کلی انواع تایمرهای موجود در STEP7 را می‌توان مطابق جدول ۱۶-۲ تقسیم‌بندی نمود.

جدول ۱۶-۲

نوع تایمر	عنوان تایمر	شروع زمان‌سنجی	نیاز به یک بودن ورودی فعال کننده در طول زمان سنجی	وضعیت خروجی تایمر
تاخیر در وصل	S_ODT	با یک شدن ورودی فعال کننده	دارد	پس از سپری شدن زمان خروجی یک می‌شود. اگر ورودی فعال کننده صفر شود خروجی نیز صفر می‌شود.
تاخیر در وصل ماندگار	S_ODTS	با یک شدن ورودی فعال کننده	دارد	پس از سپری شدن زمان خروجی یک می شود. اگر ورودی فعال کننده صفر شود خروجی یک می‌ماند.

تاخیر در قطع	S_OFFDT	وقتی ورودی فعال کننده از یک به صفر برگردد	ندارد	پس از فعال شدن خروجی یک است. با سپری شدن زمان خروجی صفر می شود
پالس	S_PULSE	با یک شدن ورودی فعال کننده	دارد	پس از فعال شدن خروجی یک است. با سپری شدن زمان خروجی صفر می‌شود.
پالس ادامه دهنده	S_PEXT	با یک شدن ورودی فعال کننده	ندارد	پس از فعال شدن خروجی یک است. با سپری شدن زمان خروجی صفر می شود.

شکل ۱۶-۳ چگونگی عملکرد انوع تایمر را نشان می‌دهد.در این شکل:
I0.0 = فعال‌ساز تایمر، t = زمان اختصاص داده شده با تایمر، Q4.0 = خروجی تایمر در انواع مختلف



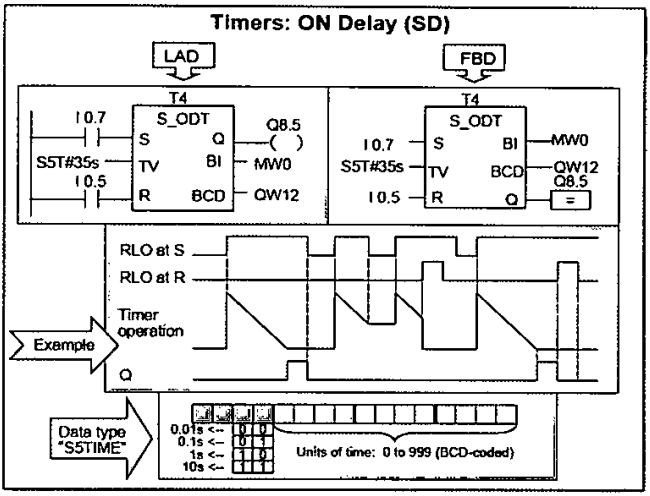
شکل ۱۶-۳ چگونگی عملکرد تایمرهای مختلف

در ادامه به بررسی انواع تایمرها، همراه با ذکر مثال می‌پردازیم.

تایمر تاخیر در وصل[1]

المان این تایمر با عنوان S_ODT در پنجره Program Element موجود است. در شکل ۱۶-۴ این تایمر در زبان LAD و FBD و چگونگی عملکرد آن نشان داده شده است.

1. On – Delay Timer

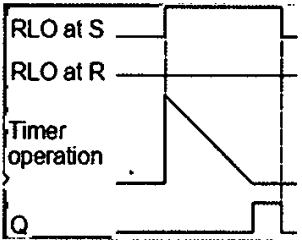


شکل ۱۶-۴ تایمر S_ODT و نحوه‌ی عملکرد آن

برای درک دقیق عملکرد این تایمر نمودار فوق را به سه بخش تقسیم می‌کنیم.

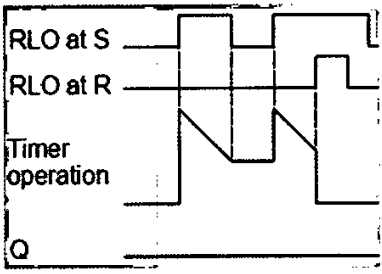
حالت ۱

- ورودی فعال کننده S یک شده، تایمر شروع به کار کرده و پس از پایان زمان‌سنجی خروجی آن یعنی Q یک شده است.
- پس از یک شدن Q، ورودی S به صفر برگشته و منجر به صفر شدن خروجی Q شده است.



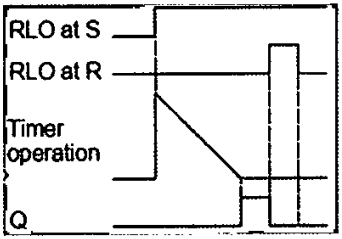
حالت ۲

- ورودی S یک شده ولی قبل از پایان زمان‌سنجی به صفر برگشته و همانطور که دیده می‌شود خروجی Q صفر است.
- در این شرایط مجدداً ورودی S فعال شده و تایمر زمان‌سنجی را از نو شروع کرده است.
- قبل از اتمام زمان‌سنجی با وجود اینکه ورودی S یک است، ورودی R یک شده و موجب شده که زمان‌سنجی قطع شود و خروجی Q همچنان صفر بماند.

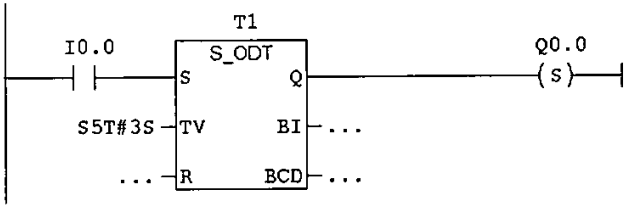


حالت ۳

- ورودی R پس از اتمام کار موفقیت‌آمیز تایمر فعال شده، در این شرایط با وجود آنکه سیگنال ورودی S فعال است خروجی تایمر صفر شده است.



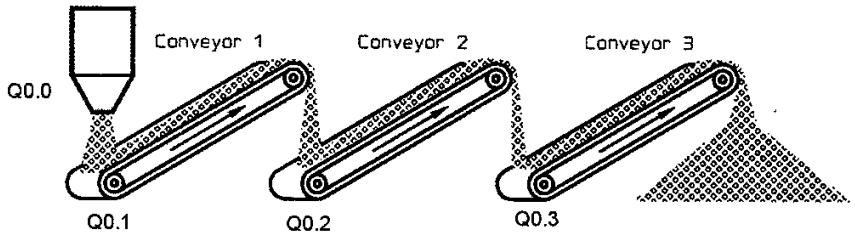
مثال ۱۶-۱: در سیستمی لازم است اگر اپراتور شستی استارت (متصل به I0.0) را به مدت ۳ ثانیه نگه داشت خروجی متصل به Q0.0 فرمان بگیرد. این تأخیر باعث می‌شود که اگر به اشتباه یک لحظه شستی وصل شد سیستم فرمان نگیرد. برنامه با استفاده از تایمر تاخیر در وصل به‌صورت شکل ۱۶-۱ خواهد بود.



شکل ۱۶-۵ برنامه مثال ۱۶-۱

مثال ۱۶-۲: در سیستم حمل مواد سه نوار نقاله مانند شکل ۱۶-۶ وجود دارد. وقتی اپراتور شستی استارت I0.0 را فعال کرد لازم است سیستم به‌صورت زیر به‌ترتیب استارت شود :

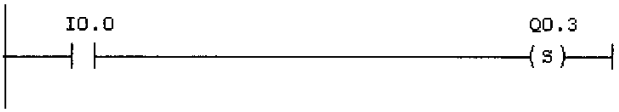
- ابتدا نوار ۳ روشن شود.
- پس از روشن شدن نوار ۳، با ۱۰ ثانیه تأخیر نوار ۲ روشن شود.
- پس از روشن شدن نوار ۲، با ۱۰ ثانیه تأخیر نوار ۱ روشن شود.
- پس از روشن شدن نوار، با ۵ ثانیه تأخیر دریچه بونکر باز شود.



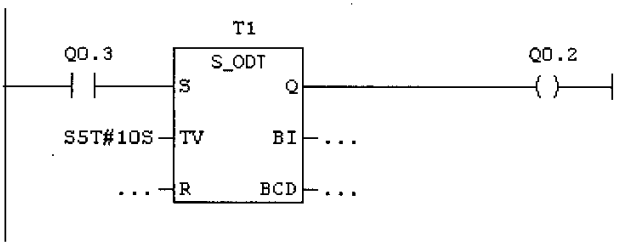
شکل ۱۶-۶ کنترل ترتیبی نوار نقاله‌های حمل مواد

روش اول: برنامه به‌صورت زیر خواهد بود:

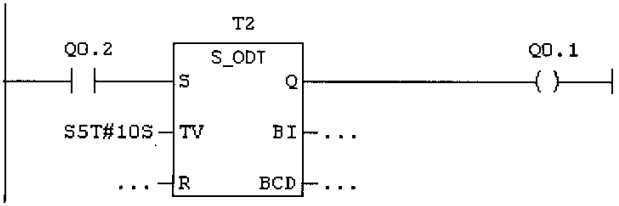
```
OB1 :  "Main Program Sweep (Cycle)"
Network 1 : Title:
```



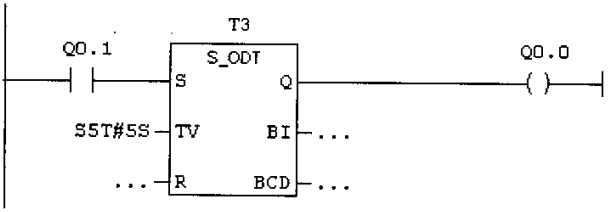

```
Network 2 : Title:
```




```
Network 3 : Title:
```



Network 4 : Title:



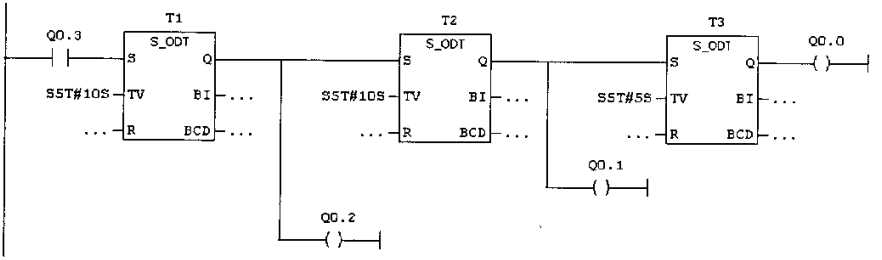
شکل ۱۶-۷ برنامه مثال ۱۶-۲ روش اول

توصیه می شود خواننده برنامه فوق را با سیمولاتور تست کند و در حالت Online وضعیت عملکرد تایمرها را مشاهده نماید.

روش دوم: در مثال فوق الزامی وجود ندارد که تایمرها در Networkهای مختلف باشند. می‌توان برنامه را به‌صورت زیر نوشت:

```
OB1 :  "Main Program Sweep (Cycle)"
Network 1 : Title:
```

Network 2 : Title:



شکل ۱۶-۸ برنامه مثال ۱۶-۲ روش دوم

تذکر: یکی از نکات مهم در برنامه‌نویسی تایمرها دقت در تکراری نبودن شماره آنهاست. اگر در برنامه فوق دو تایمر با شماره یکسان وجود داشته باشند کار کنترل به‌درستی انجام نخواهد شد. این اشکال را با سیمولاتور تست کنید.

در برنامه‌نویسی می‌توان از کنتاکت تایمر نیز استفاده کرد، به‌عنوان مثال در برنامه زیر اگر هر سه تایمر مثال ۱۶-۲ با موفقیت کار کنند و خروجی بدهند در این‌صورت M0.0 یک خواهد شد.

```
Network 3 : Title:
```

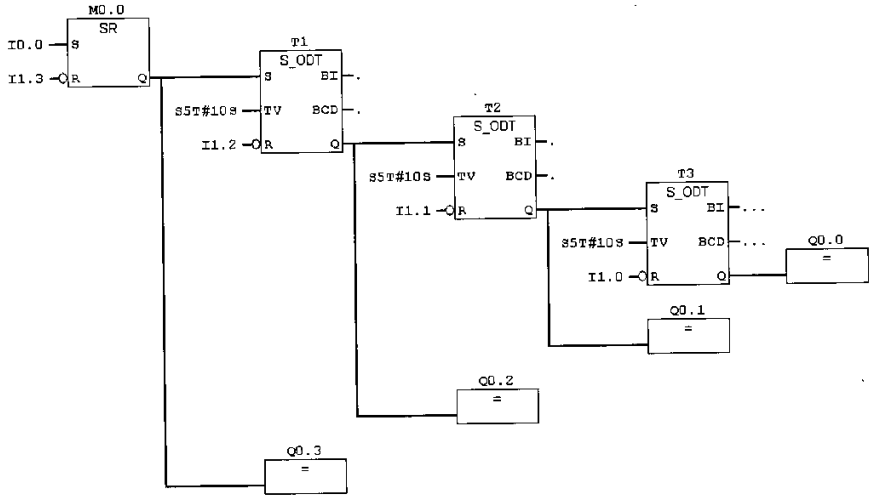
شکل ۱۶-۹ استفاده از کنتاکت تایمرها در برنامه‌نویسی

مثال ۱۶-۳: در مثال ۱۶-۲ اگر برای هر نوار نقاله سوئیچ قطع اضطراری وجود داشته باشد که به‌صورت نرمال بسته است می‌توان آدرس آن را در پایه ری‌ست تایمر به‌کار برد.

اگر هر کدام از سوئیچ‌های اضطراری عمل کنند، همان سیستم و سیستم‌های ماقبل از کار می‌افتند ولی سیستم‌های بعدی به‌کار ادامه می‌دهند. علت از کار افتادن سیستم‌های ماقبل این است که ورودی فعال کننده S آن تایمر صفر می‌شود.

در برنامه مثال ۱۶-۳ که به زبان LAD نوشته شده است، در روش اول می‌توان به ورودی ری‌ست تایمرها کنتاکت بسته کلید اضطراری را متصل کرد ولی در روش دوم این کار امکان‌پذیر نیست.

در FBD این محدودیت وجود ندارد. شکل ۱۶-۱۰ برنامه مثال ۱۶-۳ را به زبان FBD نشان می‌دهد که کل برنامه در یک Network نوشته شده است.



شکل ۱۶- ۱۰ برنامه مثال ۱۶-۳

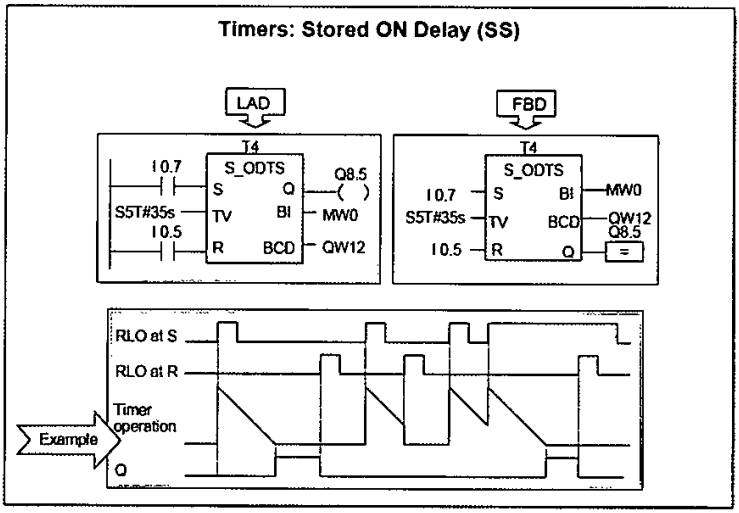
با سیمولاتور این مثال را چک کنید.

تذکر : اگر برنامه فوق را از طریق منوی View بخواهیم به زبان LAD ببینیم امکان‌پذیر نمی‌باشد و معادل STL آن نمایش داده می‌شود.

تمرین ۱۶-۱: برنامه مثال ۱۶-۳ را به‌صورتی اصلاح کنید که اگر هر کدام از کلیدهای قطع اضطراری عمل کرد کل سیستم از کار بیفتد.

تایمر تاخیر در وصل ماندگار[1]

این تایمر با المان S_OTDS معرفی شده است و رفتار آن مشابه تایمر S_ODT می‌باشد، فقط در یک مورد این دو تایمر با یکدیگر متفاوتند. در این تایمر اگر زمان‌سنجی به پایان رسید و خروجی یک شد همچنان یک می‌ماند و حتی اگر ورودی S به صفر برگردد خروجی یک باقی می‌ماند. برای صفر کردن خروجی این تایمر لازم است ورودی Reset آن فعال گردد. نمودار شکل ۱۶-۱۱ این موضوع را نشان می‌دهد.

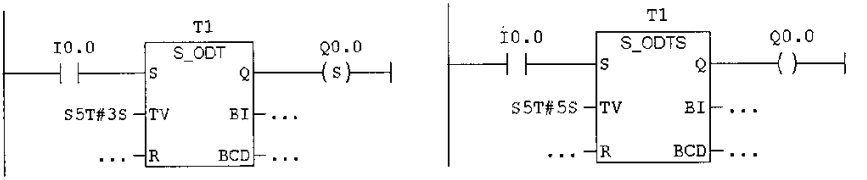


شکل ۱۶-۱۱ تایمر S_ODTS و نحوه‌ی عملکرد آن

مثال ۱۶-۴ برنامه مثال ۱۶-۱ را با استفاده از تایمر S_ODTS بنویسید.

در شکل ۱۶-۱۲ برنامه با دو نوع تایمر مقایسه شده است. همانطور که دیده می‌شود در خروجی S_ODTS نیازی به استفاده از Set وجود ندارد.

1. Retentive On – Delay Timer



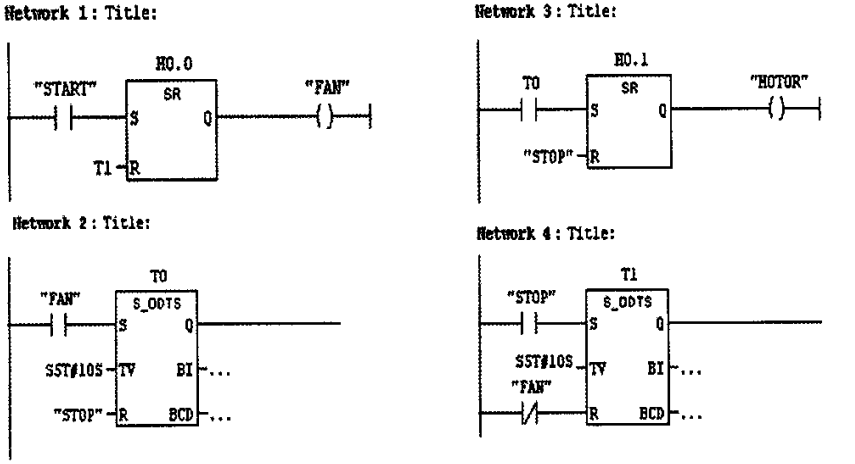
شکل ۱۶-۱۲ مقایسه برنامه مثال ۱۶-۱ و مثال ۱۶-۴

مثال ۱۶-۵: کنترل ترتیب روشن و خاموش شدن موتور و فن خنک کننده آن

در یک موتور الکتریکی بزرگ برای خنک نمودن از یک فن الکتریکی استفاده شده است. برنامه ای بنویسید که :

- با فشرده‌شدن شستی استارت، ابتدا موتور فن روشن شده و پس از ۱۰ ثانیه موتور اصلی روشن شود.
- با فشرده شدن شستی استپ، ابتدا موتور اصلی خاموش شده و پس از ۱۰ ثانیه موتور فن خاموش شود.

روش برنامه‌نویسی می‌تواند با استفاده از تایمرهای متفاوتی باشد. در این مثال ما از دو تایمر S_ODTS استفاده کرده‌ایم ولی راه ساده‌تر که بعداً خواهیم دید، استفاده از یک تایمر تاخیر در وصل و یک تایمر تاخیر در قطع است.

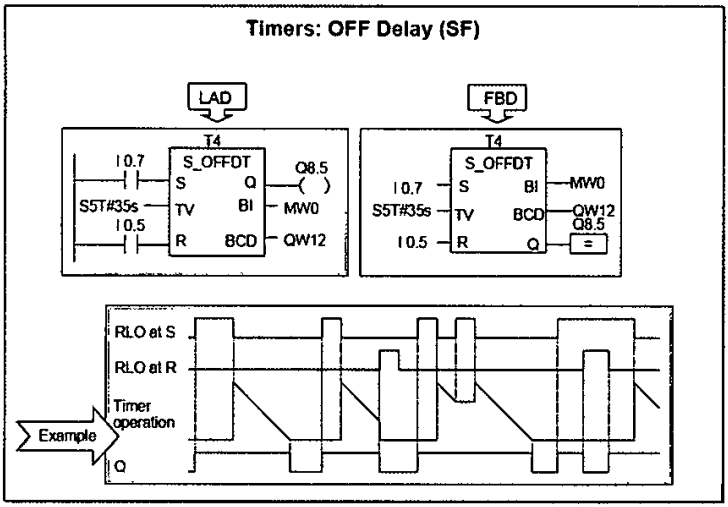


شکل ۱۶-۱۳ برنامه مثال ۱۶-۵

تایمر تاخیر در قطع[1]

این تایمر با المان موسوم به S-OFFDT معرفی شده است و همانطور که از نام آن مشخص است پس از قطع شدن ورودی‌اش، خروجی را با تأخیر زمانی قطع می‌کند. شکل ۱۶-۱۴ این تایمر را به‌صورت LAD و FBD نشان می‌دهد و عملکرد آن را با نمودار ترسیم کرده است.

1. Off – Delay Timer

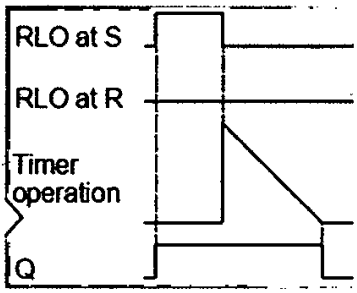


شکل ۱۶-۱۴ تایمر S_OFFDT و نحوه‌ی عملکرد آن

برای درک دقیق عملکرد این تایمر نمودار فوق را به چهار بخش تقسیم می‌کنیم.

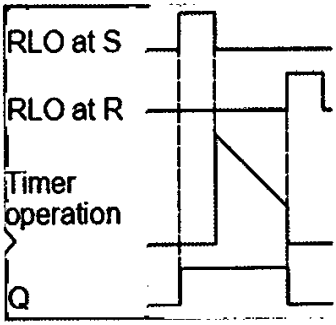
حالت ۱

با یک شدن ورودی S خروجی تایمر بلافاصله یک می‌شود، در این شرایط زمان‌سنجی هنوز انجام نمی‌شود. پس از اینکه ورودی S از یک به صفر برگشت زمان‌سنجی شروع شده و پس از سپری شدن زمان خروجی تایمر صفر می‌گردد.



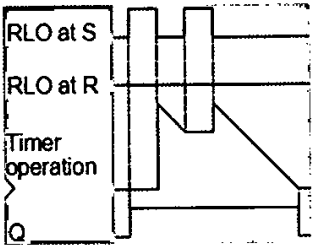
حالت ۲

با یک شدن ورودی S خروجی تایمر یک و با صفر شدن ورودی S زمان‌سنجی شروع می‌شود. قبل از اینکه زمان‌سنجی به پایان برسد ورودی R یک می‌شود، در این حالت خروجی تایمر فوراً صفر می‌گردد.



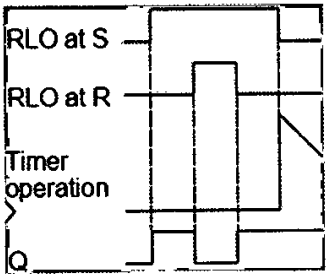
حالت ۳

با یک شدن ورودی S و صفر شدن آن زمان‌سنجی شروع می‌گردد ولی قبل از اینکه زمان‌سنجی به پایان برسد ورودی S مجدداً یک و صفر می‌شود. این کار باعث می‌شود تا زمان‌سنجی از ابتدا شروع شود. در طول این شرایط خروجی Q یک است.



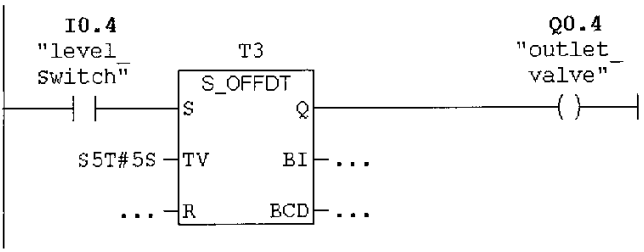
حالت ۴

با یک شدن ورودی S خروجی تایمر یک شده ولی قبل از اینکه ورودی S صفر شود ورودی Reset فعال می‌گردد که منجر به صفر شدن خروجی می‌شود. وقتی سیگنال R به صفر برگشت چون ورودی S هنوز فعال است خروجی مجدداً یک می‌شود.



مثال ۱۶-۶: در یک مخزن از یک سوئیچ سطح برای آشکارسازی سطح Low (پایین) استفاده می‌شود. وقتی سطح سیال بالاتر از Low باشد، این سوئیچ وصل و به PLC سیگنال یک می‌فرستد. وقتی سطح از Low پایین‌تر بیاید این سوئیچ به PLC صفر می‌فرستد. در این‌حالت لازم است ولو خروجی مخزن بسته شود.

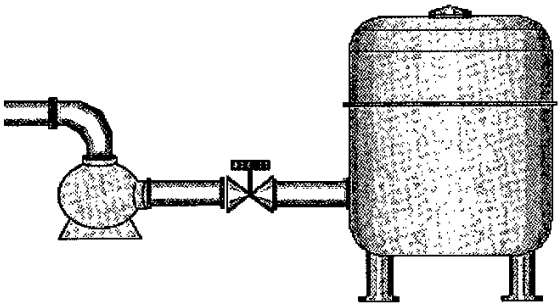
اگر از کنتاکت این سوئیچ به‌طور مستقیم به ولو فرمان بدهیم ممکن است نوسانات سطح مخزن منجر به قطع و وصل سوئیچ و در نتیجه قطع و وصل ولو شود. برای جلوگیری از این پدیده نیاز به هیسترزیس است. هیسترزیس را می‌توان با تایمر ایجاد کرد. در برنامه زیر در صورتی‌که صفر شدن سوئیچ به مدت ۵ ثانیه باقی بماند فرمان Off شدن به ولو صادر می‌گردد.



شکل ۱۶-۱۵ برنامه مثال ۱۶-۶

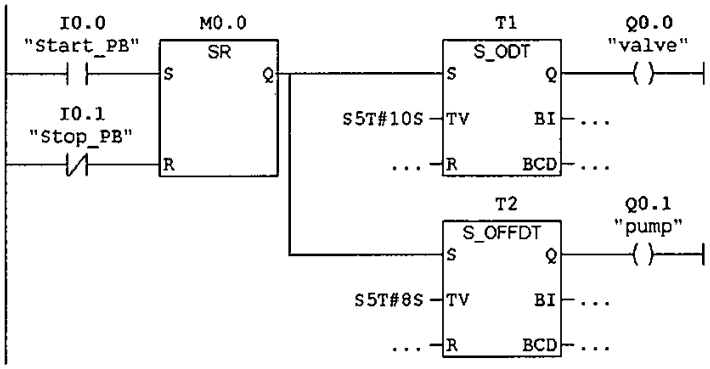
مثال ۱۶-۷: در فرآیندی از پمپ برای انتقال سیال به مخزن استفاده شده است و در مسیر انتقال از پمپ تا مخزن یک ولو On/Off به‌کار رفته است. برای استارت و استپ شرایط زیر بایستی برآورده شود:

- با استارت اپراتور ابتدا پمپ بکار بیفتد و با ۱۰ ثانیه تأخیر ولو باز شود (تا موتور زیر بار راه‌اندازی نشود)
- با استپ اپراتور ابتدا ولو بسته شود، سپس با ۸ ثانیه تأخیر پمپ از کار بیفتد.



شکل ۱۶-۱۶ فرآیند مثال ۱۶-۷

این برنامه را به روش‌های مختلف می‌توان نوشت. ساده‌ترین راه استفاده از دو تایمر تاخیر در وصل و تأخیر در قطع است. برنامه در شکل ۱۶-۱۷ آمده است و تحلیل آن به عهده خواننده می‌باشد.

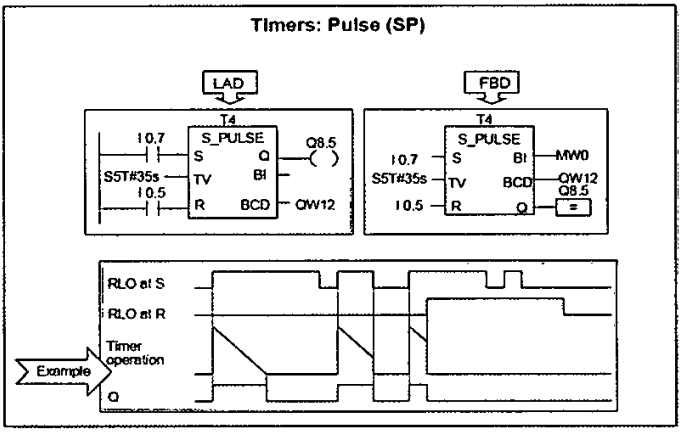


شکل ۱۶-۱۷ برنامه مثال ۱۶-۷

تمرین ۱۶-۲: برنامه نوار نقاله مثال ۱۶-۲ را به‌صورتی کامل کنید که با شستی اپراتور ابتدا دریچه بونکر بسته شود، سپس با ۱۵ ثانیه تأخیر نوار ۱ و پس از آن با ۱۵ ثانیه تأخیر نوار ۲ و پس از آن با ۲۰ ثانیه تأخیر نوار ۳ از کار بیفتد.

تایمر پالس[1]

این تایمر با المان موسوم به S-Pulse معرفی شده است. با فعال شدن ورودی تایمر به‌کار می‌افتد و خروجی‌اش یک می‌شود. پس از گذشت زمان تعیین شده بشرط اینکه ورودی‌اش همچنان یک بماند خروجی آن صفر می‌شود. شکل و عمکرد این تایمر در شکل ۱۶-۱۸ نشان داده شده است.



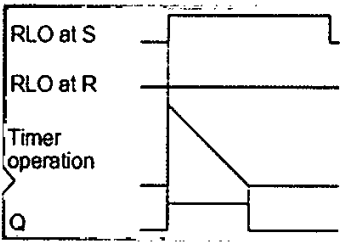
شکل ۱۶-۱۸ تایمر S_PULSE و نحوه‌ی عملکرد آن

1. Pulse Timer

برای درک دقیق عملکرد این تایمر نمودار فوق را به سه بخش تقسیم می‌کنیم:

حالت ۱

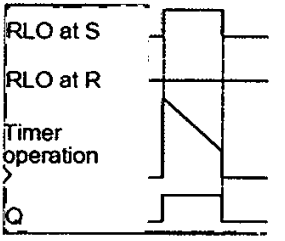
ورودی S یک می‌شود و تایمر به‌کار می‌افتد. خروجی تایمر یک است و با تمام شدن زمان‌سنجی خروجی صفر می‌شود.



دراین شرایط ورودی S همچنان یک بوده است.

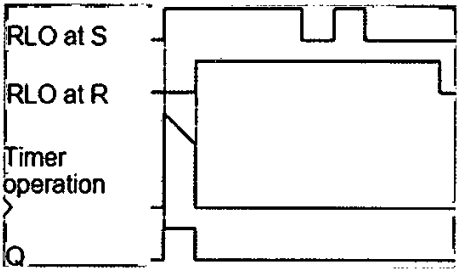
حالت ۲

ورودی S یک می‌شود و تایمر به‌کار می‌افتد ولی قبل از اتمام زمان‌سنجی ورودی S صفر می‌شود. خروجی تایمر در همین لحظه صفر می‌گردد.



حالت ۳

ورودی S فعال می‌شود و تایمر به‌کار می‌افتد ولی قبل از اتمام زمان‌سنجی ورودی Reset یک می‌شود و منجر به صفر شدن خروجی تایمر می‌گردد. در این شرایط تا زمانی که R یک است قطع و وصل ورودی S تاثیری ندارد.

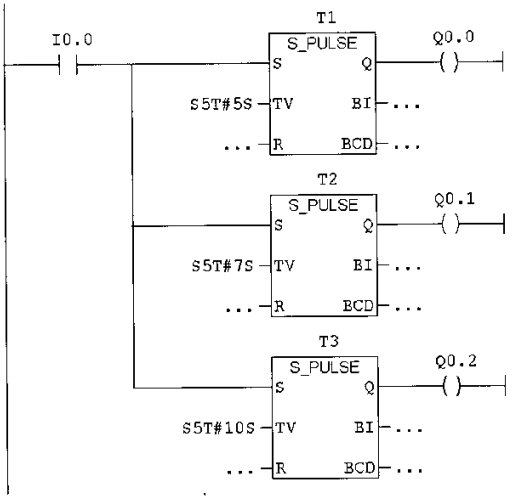


مثال ۱۶-۸: برنامه‌ای بنویسید که با فعال شدن ورودی I0.0، سه لامپ (خروجی) فعال شده و به‌ترتیب زیر خاموش شوند:

- **لامپ اول (Q0.0):** ۵ ثانیه
- **لامپ دوم (Q0.1):** ۷ ثانیه
- **لامپ سوم (Q0.2):** ۱۰ ثانیه

اگر در بین کار ورودی قطع شد هر سه لامپ خاموش شوند.

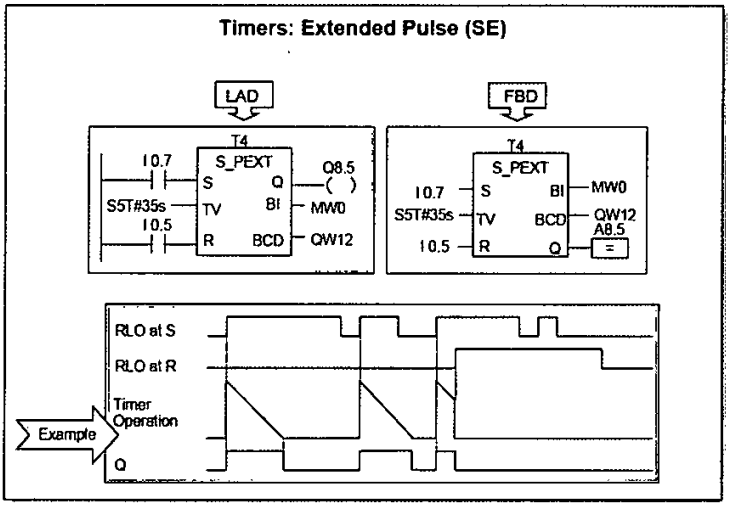
برنامه به‌سادگی توسط تایمر پالس قابل نوشتن است. به شکل ۱۶- ۱۹ توجه کنید.



شکل ۱۶-۱۹ برنامه مثال ۱۶-۸

تایمر پالس ادامه دهنده

این تایمر با المان موسوم به S-PEXT معرفی شده است. با فعال شدن ورودی تایمر به‌کار می‌افتد و خروجی‌اش یک می‌شود. پس از گذشت زمان تعیین شده خروجی آن صفر می‌شود. عملکرد آن مشابه S_Pulse است با این تفاوت که برای ادامه کار نیازی به یک بودن ورودی S ندارد. در واقع این تایمر با یک پالس نیز به‌کار می‌افتد. شکل و عمکرد این تایمر در شکل ۱۶-۲۰ نشان داده شده است.

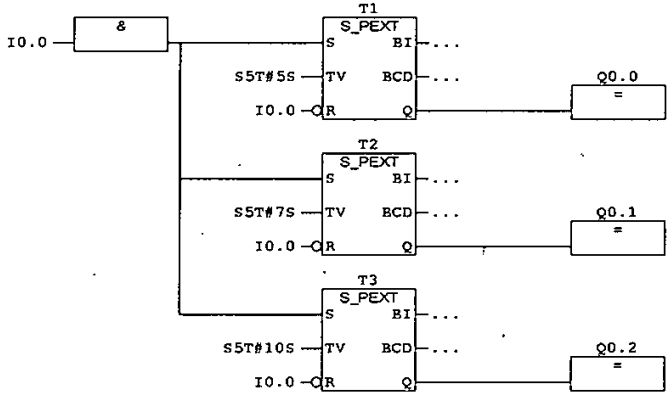


شکل ۱۶-۲۰ تایمر S_PEXT و نحوه‌ی عملکرد آن

با توضیحاتی که در مورد عملکرد S_Pulse داده شد این نمودار نیاز به توضیح ندارد. فقط لازم است به حالت وسط نمودار توجه شود که با وجود صفر شدن ورودی S تایمر به کار خود ادامه داده است.

مثال ۱۶-۹: برنامه مثال ۱۶-۸ را با تایمر PEXT بنویسید.

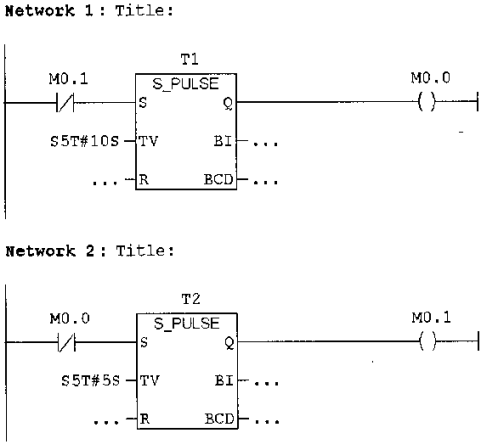
این تایمر به محض فعال شدن کلید شروع به‌کار می‌کند و شبیه S_PULSE به‌ترتیب لامپ‌ها را روشن می‌کند ولی مشکل این است که اگر سوئیچ قطع شود تایمر از کار نمی‌افتد و لامپ‌ها خاموش نمی‌شوند. برای این منظور از Not کلید I0.0 در ورودی‌های Reset تایمرها استفاده می‌کنیم. شکل ۱۶-۲۱ برنامه را به زبان FBD نشان می‌دهد.



شکل ۱۶-۲۱ برنامه مثال ۱۶-۹

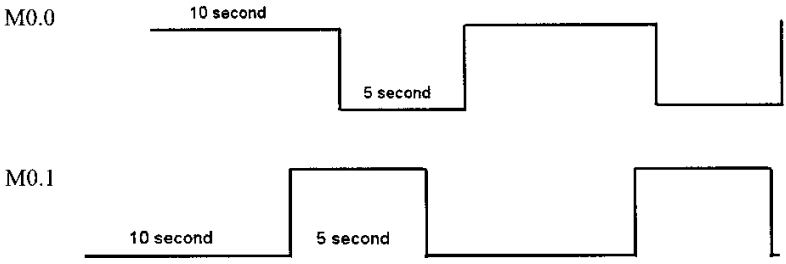
مثال ۱۶-۱۰: تولید موج مربعی با تایمر

با استفاده از هر کدام از دو تایمر پالس S_PULSE یا S_PEXT می‌توان موج مربعی تولید کرد که زمان صفر و یک شدن آن به دلخواه کنترل شود. به‌عنوان مثال برای تولید موج مربعی که ۱۰ ثانیه یک و ۵ ثانیه صفر باشد می‌توان از برنامه زیر استفاده کرد.



شکل ۱۶-۲۲ برنامه مثال ۱۶-۱۰

موج مربعی در M0.0 و M0.1 به‌صورت زیرخواهد بود، همانطور که دیده می‌شود این دو موج عکس یکدیگر هستند.

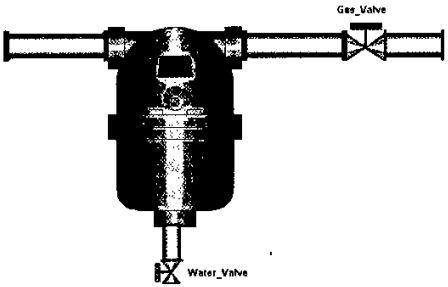


شکل ۱۶-۲۳ موج مربعی تولید شده با برنامه مثال ۱۶-۱۰

برنامه فوق را با سیمولاتور تست کنید.

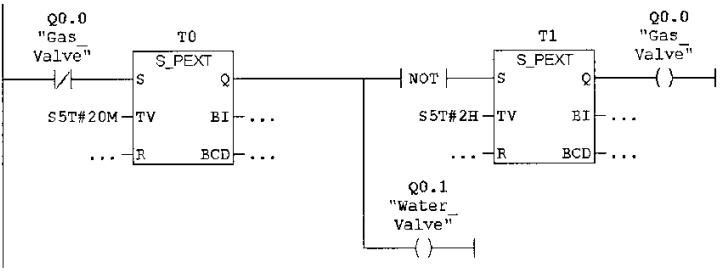
مثال ۱۶-۱۱: کنترل سیکل کاری Dryer

در فرآیندی گاز مرطوب وارد خشک‌کن شده و گاز خشک خارج می‌شود. پس از یک‌ساعت لازم است مسیر گاز بسته شده و مسیر تخلیه آب به مدت ۲۰ دقیقه باز و سپس بسته شود. با بسته‌شدن مسیر آب مجدداً مسیر گاز باز شده و این سیکل مرتباً تکرار شود.



شکل ۱۶-۲۴ فرآیند مثال ۱۶-۱۱

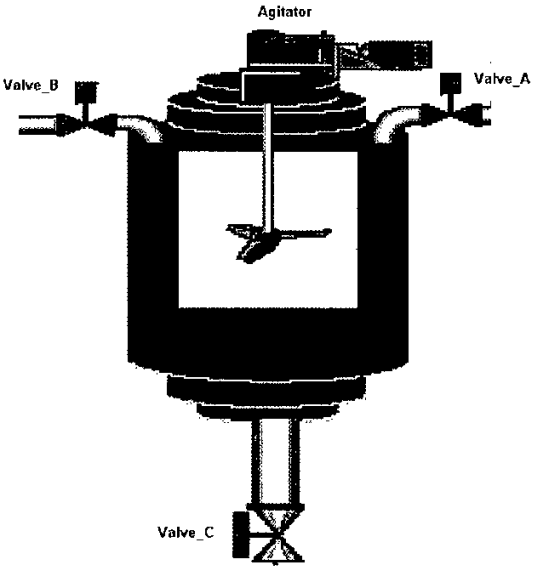
برنامه مشابه مثال ۱۶-۱۰ است که در اینجا آنرا با S_PEXT نوشته‌ایم.



شکل ۱۶-۲۵ برنامه مثال ۱۶-۱۱

مثال ۱۶- ۱۲: کنترل عملیات میکسر

در یک میکسر مانند شکل زیر ترتیب عملیات به‌صورت زیر است.



شکل ۱۶-۲۶ فرآیند میکسر مثال ۱۶-۱۲

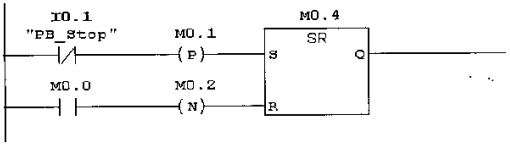
وقتی اپراتور شستی استارت را فشار دهد مراحل زیر به‌ترتیب انجام گیرد:

۱. ولو مسیر A به‌مدت ۲۰ ثانیه باز سپس بسته شود.
۲. ولو مسیر B به‌مدت ۱۵ ثانیه باز سپس بسته شود.
۳. موتور همزن به‌مدت ۱۲ ثانیه کار کند سپس از کار بیفتد.
۴. ولو خروجی C به‌مدت ۲۵ ثانیه باز سپس بسته شود.
۵. تا زمانی که اپراتور استپ را فعال نکرده سیکل از مرحله ۱ تکرار شود.

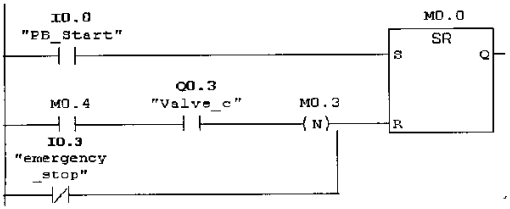
تذکر

- وقتی اپراتور شستی استپ را فشار دهد سیکل فعلی کامل شده ولی سیکل جدید شروع نشود.
- اگر اپراتور کلید قطع اضطراری را فعال کند سیکل در همان مرحله متوقف گردد.
-

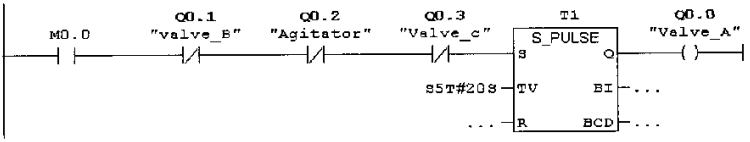
برنامه به‌صورت زیر خواهد بود. آنالیز به عهده خواننده است.



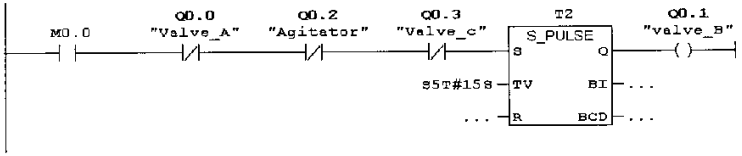
Network 2 : Title:



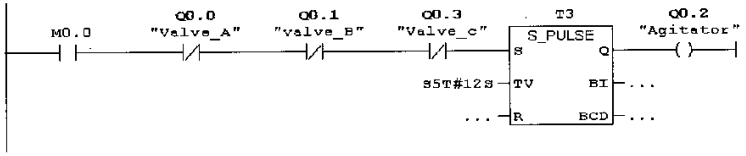
Network 3 : Title:



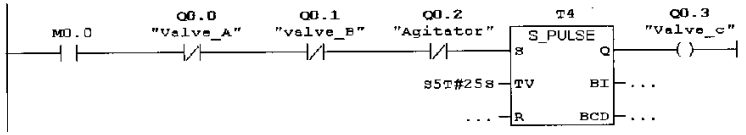
Network 4 : Title:



Network 5 : Title:



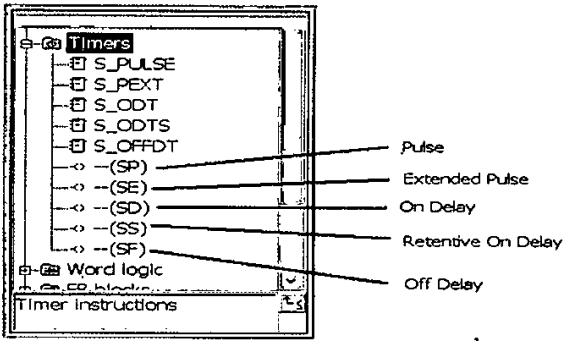
Network 6 : Title:



شکل ۱۶-۲۷ برنامه مثال ۱۶-۱۲

۴-۱۶ دستورات بیتی تایمر

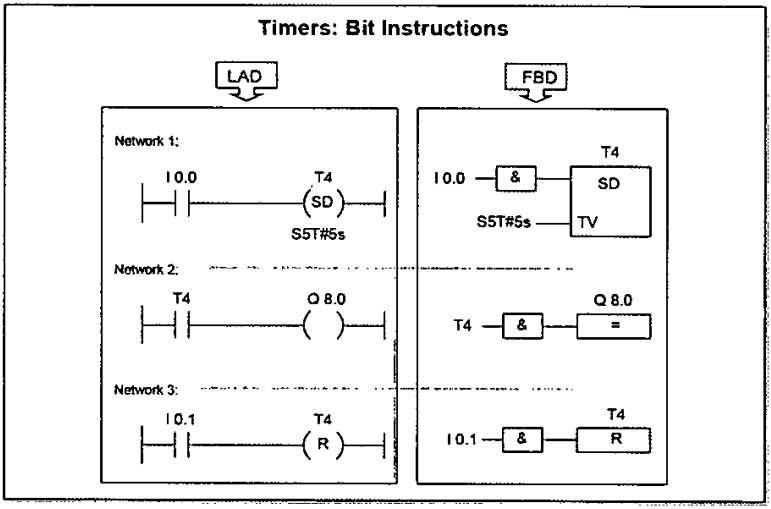
علاوه بر المان‌هایی که برای تایمرها معرفی شد می‌توان آنرا به‌صورت بیتی نیز به‌کار برد. دستورات در شکل ۱۶-۲۸ نشان داده شده‌اند.



شکل ۱۶-۲۸ دستورات بیتی تایمرها

در این حالت پس از به‌کار بردن المان تایمر، فقط شماره تایمر و زمان آن مشخص می‌شود. برای استفاده از خروجی آن یا ری‌ست کردن آن بایستی دستورات جداگانه‌ای به‌کار برد.

شکل ۱۶-۲۹ نمونه‌ای از کاربرد این دستور را برای فعال‌سازی تایمری از نوع S_ODT نشان می‌دهد.



شکل ۱۶-۲۹ دستور بیتی تایمر به‌منظور فعال و غیرفعال‌سازی تایمر از نوع S_ODT

۱۶-۵ بررسی ماندگاری تایمرها

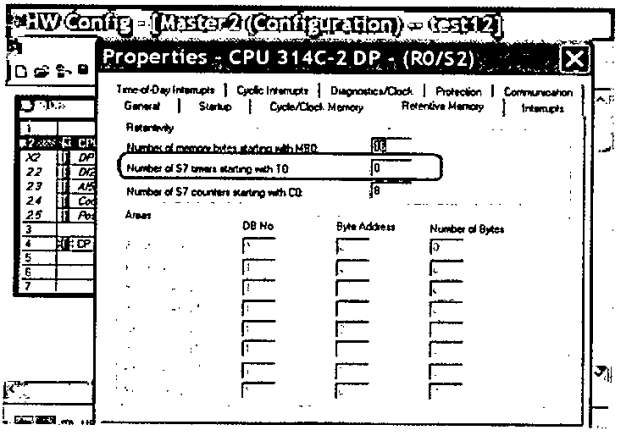
همانطور که قبلاً اشاره شد تایمرها به‌طور پیش فرض ماندگار نیستند. یعنی اگر زمان‌سنجی تایمر هنوز به‌پایان نرسیده و شرایط زیر رخ دهد، تایمر کار زمان‌سنجی را از اول شروع می‌کند:

- Stop شدن CPU و راه‌اندازی مجدد آن
- قطع و وصل تغذیه CPU

ممکن است در برخی فرآیندها لازم باشد که تایمر از اول شروع نکند بلکه از جایی که قطع شده بود ادامه دهد. یعنی حالت ماندگار داشته باشد. ماندگاری تایمرها را به یکی از دو طریق زیر می‌توان انجام داد:

- استفاده از Hot Restart که خاص S-400 است.
- استفاده از Warm Restart و تعیین کردن شماره تایمرهای ماندگار.

در S7-300 فقط روش دوم قابل استفاده است. برای این منظور در HW Config روی CPU دوبار کلیک کرده یا Properties آنرا باز می‌کنیم. در سربرگ Retentivity می‌توان شماره تایمرها را وارد کرد.



شکل ۱۶- ۳۰ تنظیم ماندگار کردن تایمرها

همانطور که دیده می‌شود به‌طور پیش‌فرض تعداد تایمرهای ماندگار صفر است. اگر تعداد را یک وارد کنیم فقط T0 ماندگار خواهد شد و اگر عدد را دو وارد کنیم فقط T0 و T1 ماندگار خواهند بود. پس همیشه شماره از T0 شروع می‌شود و امکان اینکه فقط به‌عنوان مثال T1 را ماندگار کنیم وجود ندارد. به همین جهت توصیه می‌شود در برنامه‌نویسی شماره‌های پایین را برای تایمرهایی که نیاز به ماندگاری دارند به‌کار ببرید.

پس از ذخیره و دانلود از HW Config تایمرهای تعریف شده به‌صورت ماندگار خواهند بود.

تذکر: با Cold Restart و با Clear /Reset تایمر از صفر شروع به‌کار می‌کند، حتی اگر ماندگار باشد.

تمرین ۱۶-۳: در مثال ۱۶-۸ تایمرها را به‌صورت ماندگار تعریف کرده و عملکرد آنها را با PLC آزمایش کنید.

مثال۱۶- ۱۳: در مثال ۱۶-۱۱ مربوط به سیستم خشک‌کن گاز، فرض کنید که تایمر مسیر آب فعال است و ۱۰ دقیقه زمان سپری شده باشد. اگر در این شرایط تغذیه قطع و وصل شود نباید تایمر از ابتدا شروع کند بلکه باید از همان ۱۰ دقیقه ادامه دهد. برای این منظور تعداد تایمرهای Retentive را دو تعریف می‌کنیم چون از T0 و T1 در برنامه استفاده کرده‌ایم.

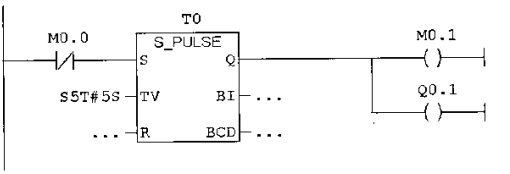
اگر برنامه را با PLC تست کنیم می‌بینیم مشکلی وجود دارد.

- وقتی ولو مسیر آب باز است و تغذیه را قطع و وصل می‌کنیم برنامه درست کار می‌کند و از همانجا که قطع شده ادامه می‌دهد.
- وقتی ولو مسیر گاز باز است و تغذیه را قطع و وصل می‌کنیم برنامه درست کار نمی‌کند و با وصل تغذیه تایمر ولو آب از اول شروع بکار می‌کند و ولو آب باز می‌شود. تایمر ولو گاز کار نمی‌کند.

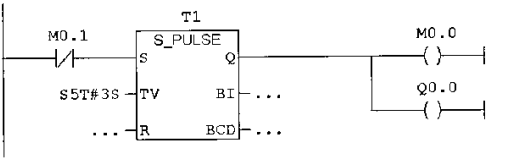
علت این است که با هر قطع و وصل تغذیه مقادیر Q0.0 و Q0.1 هر دو پاک می‌شوند، بنابراین چون در ابتدای برنامه از کنتاکت NC مربوط به ولو گاز برای تایمر آب استفاده شده این تایمر به‌کار می‌افتد.

برای رفع مشکل فوق می‌توان در برنامه به‌جای Q از M استفاده کرد و در انتها مقادیر M را به Q انتقال داد. از آنجا که تعداد ۱۶ بایت از Memory Bitها به‌صورت پیش‌فرض ماندگار هستند (رجوع شود به شکل۱۶-۳۰) بنابراین با قطع و وصل‌تغذیه‌مقادیر M0.0 و M0.1 نیز پاک نشده و کار کنترل به‌درستی انجام می‌شود. برنامه به‌صورت شکل ۱۶-۳۱ است.

```
Network 1: Title:
```




```
Network 2: Title:
```

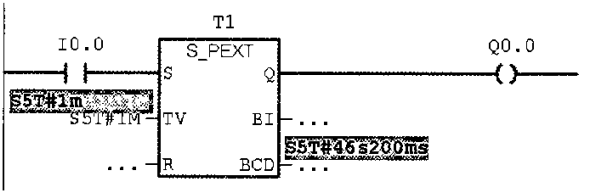


شکل ۱۶-۳۱ برنامه مثال ۱۶-۱۳

تمرین۱۶- ۴: با توجه به نکته فوق برنامه مثال ۱۶-۱۲ مربوط به میکسر را به‌صورتی ویرایش کنید که با قطع و وصل تغذیه مشکلی پیش نیاید و عملیات از همان نقطه ادامه یابد.

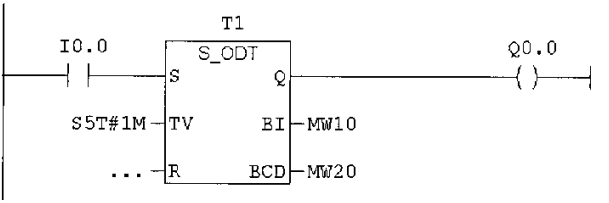
۱۶-۶ بررسی خروجی‌های BCD و BI تایمر

بلاک‌های تایمر دو خروجی BI و BCD دارند که مقدار زمان سپری شده را نشان می‌دهند. در حالت Online این زمان در خروجی تایمر قابل مشاهده است.



شکل ۱۶-۳۲ مشاهده خروجی BCD در حالت Online

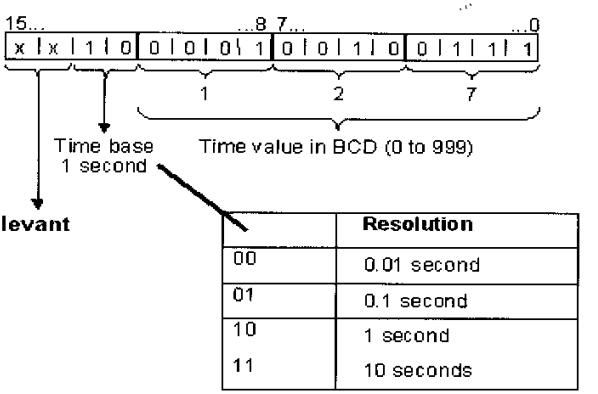
این دو خروجی از جنس Word هستند بنابراین برای اختصاص آدرس حافظه به آنها بایستی از فرمت MW استفاده کرد.



شکل ۱۶-۳۳ اختصاص آدرس به خروجی های BCD و BI

اگر برنامه فوق را به PLC دانلود کنیم و محتویات MW20 را به‌صورت بیتی ببینیم، حالت‌های مختلفی خواهیم داشت. در تمام این حالات دیده می‌شود که:

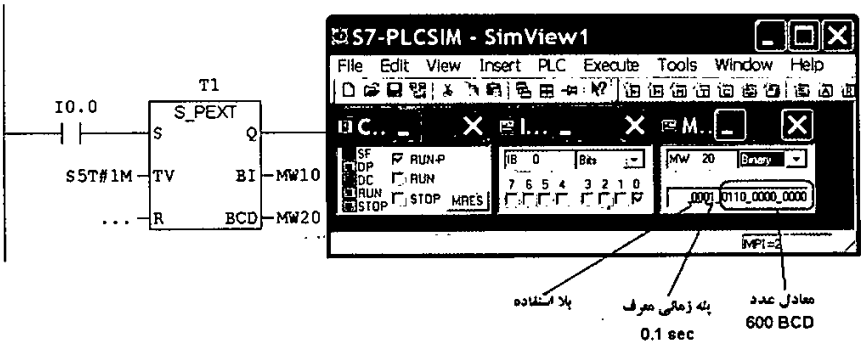
- دو بیت با ارزش بالاتر یعنی M20.7 و M20.6 صفر هستند و استفاده نشده‌اند.
- دو بیت بعدی یعنی M20.4 و M20.5 معرف پله‌های زمان هستند.
- ۱۲ بیت باقیمانده معرف زمان است.



	Resolution
00	0.01 second
01	0.1 second
10	1 second
11	10 seconds

شکل ۱۶-۳۴ ساختار ۱۶ بیت مربوط به حافظه تایمر

مثال۱۶-۱۴: در شکل ۱۶-۳۵ تایمر با زمان یک دقیقه تنظیم شده است. به محض فعال شدن سوئیچ I0.0 اگر مقداری که در MW20 قرار می‌گیرد را به‌صورت بیتی مشاهده کنیم همانطور که با سیمولاتور نشان داده شده است، مشاهده می‌کنیم که ۱۲ بیت سمت راست معرف عدد 600 BCD می‌باشند و دو بیت بعدی که مقدار 01 دارند معرف پله زمانی 0.1 ثانیه است. اگر دو عدد 600 و 0.1 را در هم ضرب کنیم عدد ۶۰ ثانیه یعنی یک دقیقه به‌دست می‌آید.
وقتی تایمر در حال کار است پله زمانی ثابت و مقدار BCD کاهش می‌یابد تا به صفر برسد.



شکل ۱۶-۳۵ برنامه مثال ۱۶-۱۴

در شرایط فوق اگر مقدار MW10 را که به خروجی BI متصل است مشاهده کنیم، عدد هگز 258 معادل 600 دسیمال است را نشان می‌دهد که معرف تعداد پله زمانی باقیمانده به‌صورت عدد صحیح است.

تمرین۱۶-۵: مشابه روش ذکر شده در مثال ۱۶-۱۴، معادل BCD و BI چند حالت دیگر را با اختصاص زمان‌های مختلف به تایمر بررسی کنید.

۱۶-۷ تمرین

۱- برنامه‌ای بنویسید که بتوان توسط آن یک موتور سه‌فاز را به‌صورت ستاره- مثلث به شرح زیر راه‌اندازی نمود:

- با فشردن شستی (NO) I0.0 خروجی‌های Q0.0 (خروجی اصلی) و Q0.1 (خروجی ستاره) همزمان روشن شده و پس از ۶ ثانیه ابتدا خروجی ستاره قطع و سپس خروجی Q0.2 (خروجی مثلث) وصل شود.
- هیچ‌گاه خروجی‌های ستاره و مثلث همزمان روشن نشوند.
- با فشرده شدن شستی (NC) I0.1 همه خروجی‌ها خاموش شوند.

۲- برنامه فوق را به‌صورتی طراحی نمایید که امکان کنترل دور موتور به‌صورت چپگرد-راستگرد نیز وجود داشته باشد. در اینصورت:

- I0.0 استارت راستگرد (تیغه باز)
- I0.1 استپ (تیغه بسته)
- I0.2 استارت چپگرد (تیغه باز)

- Q0.0 خروجی اصلی در حالت راستگرد
- Q0.1 خروجی اصلی در حالت چپگرد
- Q0.2 خروجی ستاره
- Q0.3 خروجی مثلث

نکات مهم

- هیچ‌گاه خروجی‌های ستاره و مثلث یا چپگرد و راستگرد همزمان روشن نشوند.
- در زمانی که موتور در یک جهت در حال چرخش است، فعال شدن شستی راه‌اندازی دور عکس، تاثیری در اجرای برنامه نداشته باشد.
- تا زمانی که موتور استپ نشده است، امکان تغییر جهت آن وجود نداشته باشد.

۳- برنامه راه‌اندازی یک موتور سه‌فاز به‌صورت چپگرد، راستگرد را طراحی نمایید که هرگاه موتور در حال کار استپ شود، تا ۱۰ ثانیه امکان راه‌اندازی آن در دور عکس زمانِ کار وجود نداشته باشد.

۴- برنامه راه‌اندازی یک موتور سه‌فاز به‌صورت چپگرد، راستگرد را طراحی نمایید که هرگاه دور موتور در حال کار تغییر داده شود، به‌طور اتوماتیک موتور خاموش شده و پس از ۱۰ ثانیه با دور جدید شروع به‌کار نماید.

فصل ۱۷

دستورات برنامه‌نویسی کانترها

۱۷-۱ مقدمه
۱۷-۲ انواع کانترها و ورودی و خروجی‌های آن‌ها
۱۷-۳ اصول عملکرد کانتر
۱۷-۴ بررسی خروجی‌های CV و CV_BCD کانتر
۱۷-۵ دستورات بیتی کانتر
۱۷-۶ تمرین

در این فصل با نحوه برنامه‌نویسی و نکات مربوط به شمارنده‌ها در زبان‌های LAD و FBD آشنا می‌شوید.

فصل
۱۷

چکیده مطالب

- شمارنده‌ها با حرف C آدرس‌دهی شده و از C0 شروع می‌گردند.
- به‌صورت پیش‌فرض تعدادی از کانترها ماندگار هستند و در صورت قطع تغذیه پاک نمی شوند.
- شمارش کانترها به‌صورت BCD است بنابراین به ۹۹۹ محدود می‌شود.
- می‌توان به کانترها مقدار اولیه اختصاص داد. عدد به فرمت C# داده می‌شود.
- کانترها در زبان LAD/FBD سه نوع هستند: افزایشی، کاهشی، دو کاره (افزایشی-کاهشی).

۱۷-۱ مقدمه

Counter یا شمارنده در برنامه‌نویسی برای کاربردهای مختلف به‌کار می‌رود. برخی از کاربردها عبارتند از:

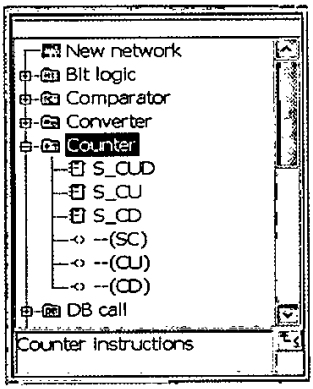
- شمارش تعداد بسته محصول
- شمارش تعداد دفعات انجام سیکل فرآیند
- شمارش تعداد دفعات قطع و وصل یک وسیله (برای مقاصد تعمیراتی)
- زمان‌سنجی بر اساس شمارش پالس

در برنامه‌نویسی Step7 سه دسته کانتر وجود دارد :

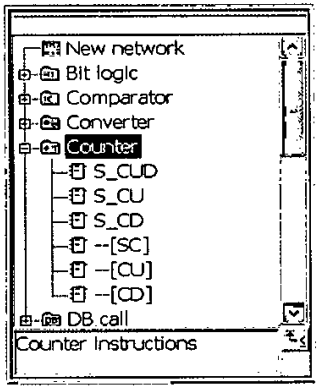
- کانترهای پایه: این تایمرها در پنجره Program Element به‌صورت یک المان پایه قابل استفاده هستند.
- کانترهای IEC: این کانترها به‌صورت SFB هستند و در کتابخانه نرم‌افزار موجود هستند.
- کانترهای کاربر: کاربر می‌تواند از طریق برنامه‌نویسی کانتر دلخواهی ایجاد نماید.

در این فصل به کانترهای نوع اول یعنی کانترهای پایه می‌پردازیم و نحوه برنامه‌نویسی آنها را به زبان‌های LAD و FBD تشریح می‌کنیم. زبان STL و کانترهای IEC در کتاب‌های پیشرفته و تکمیلی، و کانترهای کاربر در فصل‌های بعدی همین کتاب تشریح شده‌اند.

اگر در پنجره Program Element بخش کانترها را باز کنیم زیر مجموعه آنها را به‌صورت شکل ۱۷-۱۱ خواهیم دید.



LAD



FBD

شکل ۱۷-۱ محل قرارگیری دستورات Counters در پنجره Program Element

المان‌های مورد استفاده برای کانتر در هر دو زبان FBD و LAD مشابه هستند و نیازی نیست که به‌صورت مستقل تشریح گردند.

قبل از ورود به بحث کانترها نکاتی را در مورد عملکرد آنها یادآور می‌شویم:

- کانترها در حافظه System Memory قرار می‌گیرند.

- هر کانتر در حافظه سیستم دارای یک شماره منحصر بفرد است.
- شماره کانترها از C0 شروع می‌شود.
- حد ماکزیمم برای شماره کانتر به ویژگی‌های CPU بستگی دارد. در بسیاری از CPUهای 300 تعداد کانترها و تایمرها مشابه و ۲۵۶ عدد است یعنی از C0 تا C255.
- شماره خارج از رنج برای کانتر در S7-400 یا S7-300 قدیمی می‌تواند منجر به توقف CPU و روشن شدن چراغ فالت آن شود. این خطا در S7-300 جدید فقط چراغ فالت را روشن می‌کند.
- تعدادی از کانترها (۸ کانتر اول) به‌طور پیش‌فرض ماندگار هستند. این کانترها در صورت قطع تغذیه پاک نمی‌شوند و با وصل تغذیه از جایی‌که شمارش آنها متوقف شده ادامه می‌دهند. می‌توان در ویژگی‌های CPU کانترهای بیشتری را ماندگار کرد یا آنها را از حالت ماندگار خارج کرد.
- هر کانتر برای شمارش نیاز به لبه بالا رونده در ورودی خود دارد.
- شمارش کانترها به‌صورت BCD انجام می‌شود بنابراین به ۹۹۹ محدود می‌گردد.
- هر کانتر در حافظه سیستم یک حافظه ۱۶ بیتی اشغال می‌کند.

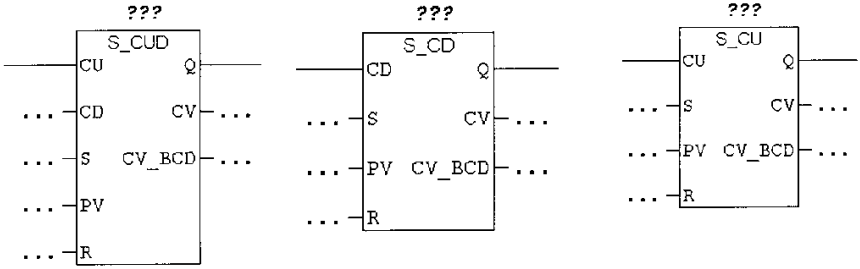
۱۷-۲ انواع کانترها و ورودی و خروجی‌های آنها

به‌طور کلی کانترهای موجود در STEP7 را می‌توان به انواع زیر تقسیم نمود:

۱- **کانتر افزایشی (S_CU):** مقدار عدد شمارش‌شده توسط این کانتر به‌صورت صعودی افزایش پیدا می‌کند. یعنی هر باری که کانتر فعال شود، یک عدد به مقدار درون آن افزوده می‌گردد.

۲- **کانتر کاهشی (S_CD):** با فعال شدن این نوع کانتر، مقدار شمارش شده توسط آن کاهش پیدا می‌نماید.

۳- کانتر افزایش/کاهشی (S_CUD): در این کانتر امکان شمارش افزایشی و کاهشی وجود دارد.



شکل ۱۷-۲ انواع کانترها در LAD و FBD

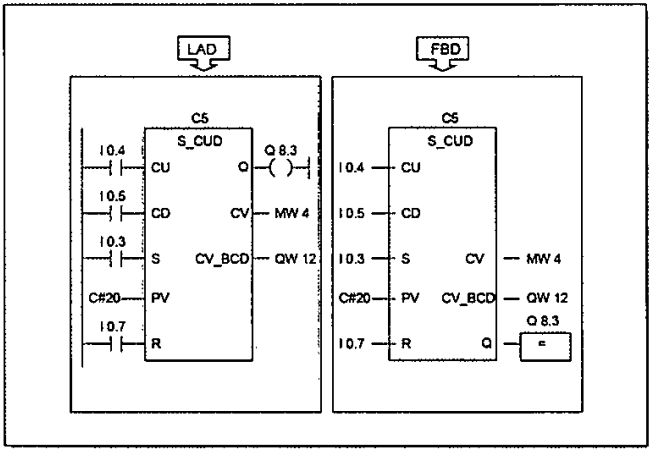
ورودی و خروجی‌های کانترها

همانطور که در شکل۱۷-۲ مشاهده می‌شود، کانتر دارای پایه‌های زیر می‌باشد:

- **CU:** فعال‌ساز افزایشی کانتر است، هرگاه فعال شود خروجی کانتر یک عدد افزایش پیدا می‌کند. این ورودی در کانتر کاهشی وجود ندارد.

- **CD:** فعال‌ساز کاهشی کانتر است، هرگاه فعال شود خروجی کانتر یک عدد کاهش پیدا می‌نماید. این ورودی در کانتر افزایشی وجود ندارد.
- **S:** هرگاه فعال شود، کانتر با عدد اختصاص داده شده در پایه‌ی PV ست می‌شود.
- **PV:** می‌توان در آن عددی به فرم BCD نوشت که با فعال‌شدن پایه‌ی S، کانتر این عدد را به‌عنوان مقدار اولیه می‌گیرد.
- **R:** به‌منظور ری‌ست نمودن کانتر از این پایه استفاده می‌شود. با فعال شدن آن مقدار شمارش شده توسط کانتر صفر می‌شود.
- **Q:** در شرایطی که مقدار شمارش شده توسط کانتر عددی غیر از صفر باشد، این پایه فعال می‌شود.
- **CV_BCD:** خروجی که مقدار شمارش شده توسط کانتر را به فرم BCD، نشان می‌دهد.
- **CV:** خروجی که مقدار شمارش شده را به فرم Hex نشان می‌دهد.

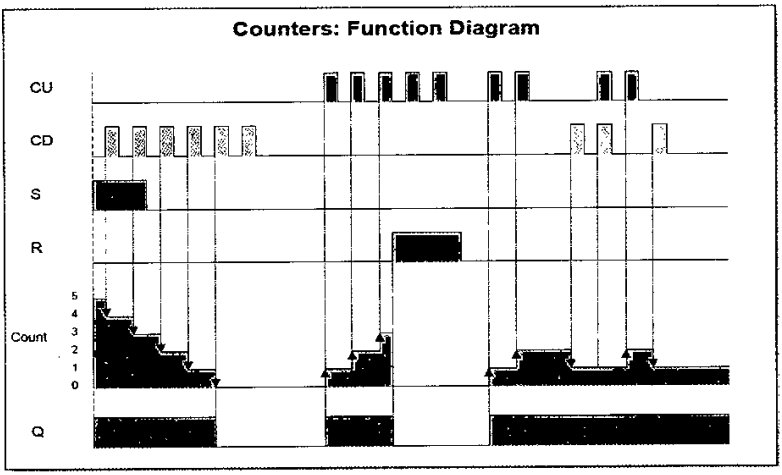
شکل ۱۷-۳ کانتر CUD را در زبان LAD و FBD نشان می‌دهد.



شکل ۱۷-۳ کانتر افزایشی/کاهشی (S_CUD)

۱۷-۳ اصول عملکرد کانتر

شکل ۱۷-۴ به‌طور کلی اصول کارکرد کانترها را نشان می‌دهد. همانطور که دیده می‌شود با رسیدن لبه مثبت به پایه CU مقدار شمارش افزایش و با رسیدن لبه مثبت پالس به پایه CD مقدار شمارش کاهش می‌یابد. یک بودن سیگنال در پایه‌های فوق هیچ تأثیری در شمارش ندارد و فقط لبه مثبت است که تأثیرگذار است.



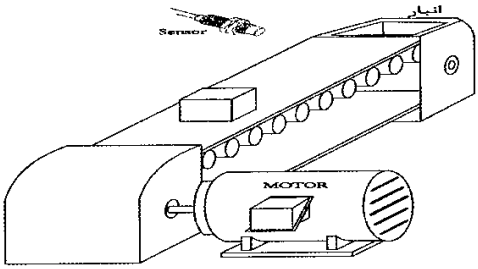
شکل ۱۷-۴ اصول کارکرد کانتر

اگر سیگنال Reset فعال شود، مقدار کانتر به صفر برمی‌گردد. تا زمانی‌که ورودیR یک باشد، پالس‌های رسیده به ورودی‌های CU یا CD شمارش نمی‌شوند.

ورودی‌های S و PV با یکدیگر به‌کار می‌روند، به‌صورت پیش‌فرض غیر فعال هستند ولی اگر به یکی از این دو آدرس یا مقدار اختصاص داده شود دیگری نیز به رنگ قرمز در می‌آید و باید آدرس‌دهی شود.

مثال ۱۷-۱: شمارش موجودی انبار

در یک پروسه‌ی صنعتی مطابق شکل ۱۷-۵، از یک نوارنقاله جهت انتقال قطعات مختلف به انبار استفاده می‌شود. نوار نقاله توسط یک موتور الکتریکی به حرکت در می‌آید. به‌منظور تشخیص ورود قطعات به انبار، از یک عدد سنسور فتوالکتریک (نوری) استفاده شده است. این سنسور در شرایط عادی قطع بوده و هنگامی‌که قطعه‌ای از مقابل آن عبور نماید وصل شده و یک سیگنال الکتریکی به PLC می‌دهد.



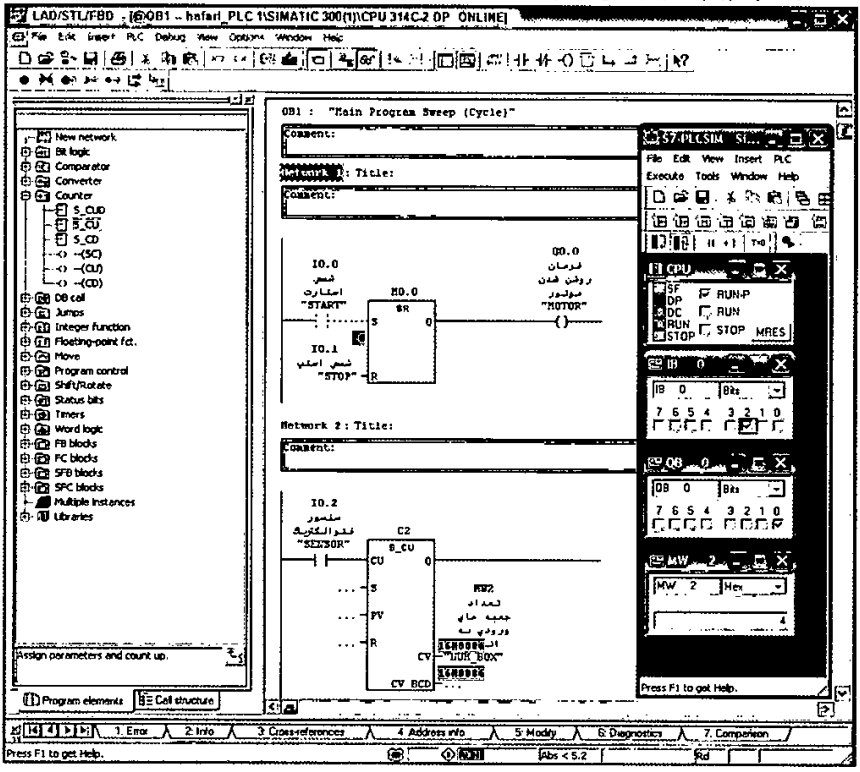
شکل ۱۷-۵ ادوات به‌کار رفته در مثال ۱۷-۱

منطق کنترل

۱- با فشرده‌شدن شستی استارت، موتور روشن شده و با فشرده‌شدن شستی استپ، موتور خاموش شود.

۲- تعداد جعبه‌های ورودی به انبار با توجه به سیگنال ارسالی توسط سنسور، شمارش شده و در حافظه MW2 ذخیره شود.

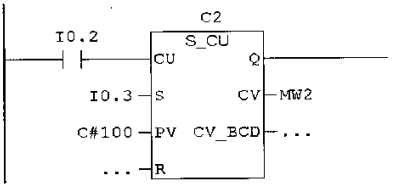
با توجه به منطق فوق نیاز به کانتر از نوع افزایشی داریم. برنامه در شکل ۱۷-۶ نشان داده شده است، می‌توان با استفاده از سیمولاتور برنامه را آزمایش کرد و مقدار ثبت شده در MW2 را به فرمت دلخواه Hex یا Dec مشاهده نمود.



شکل۱۷-۶ برنامه‌ی کنترلی مثال ۱۷-۱

مثال ۱۷-۲: شمارش موجودی انبار با لحاظ کردن مقدار اولیه

اگر در مثال ۱۷-۱ قبلاً در انبار تعداد ۱۰۰ قطعه موجود بوده و بخواهیم شمارش از ۱۰۰ به بعد انجام شود، می‌توان از برنامه زیر استفاده کرد. در این برنامه به محض اینکه اپراتور شستی I0.3 را فشار دهد کانتر با مقدار اولیه ۱۰۰ ست می‌شود که این مقدار را می‌توان در MW2 نیز مشاهده نمود. پس از آن با رسیدن پالس از سنسور شمارش از ۱۰۰ ادامه می‌یابد.



شکل ۱۷-۷ برنامه مثال ۱۷-۲

تذکر ۱: اگر I0.2 پس از فعال شدن در وضعیت یک بماند تأثیری بر شمارش ندارد ولی اگر به صفر برگردد و سپس یک شود یعنی لبه مثبت در آن اتفاق بیفتد، شمارنده مجدداً به 100 بر می‌گردد.

تذکر ۲: اگر در لحظه ست‌کردن مقدار اولیه با I0.3 سنسور نیز فعال باشد شمارنده به عدد 101 ست می‌شود.

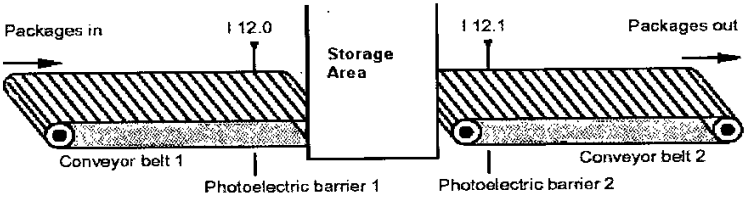
تذکر ۳: ممکن است اپراتور بخواهد مقدار اولیه را به هر عدد دلخواهی تنظیم کند. این کار چگونه امکان‌پذیر است؟ بدیهی است در این حالت بایستی ورودی PV یک عدد ثابت نباشد بلکه آدرس حافظه‌ای از جنس word به آن اختصاص داده شود مانند MW4. در این حالت MW4 از ابزارهای مانیتورینگ اپراتوری مانند صفحات WinCC مقدار می‌گیرد.

تذکر ۴: شمارش کانتر فقط تا ۹۹۹ صورت می‌گیرد. اگر تعداد قطعات بیش از این حد باشد می‌توان از کانتر دیگری نیز استفاده کرد و هر بار که کانتر اول به ۹۹۹ رسید کانتر دوم یکی افزایش یابد و کانتر اول ری ست شود. برای نوشتن این برنامه به دستورات مقایسه و محاسبه‌ای نیاز داریم. این مثال در فصل‌های بعد خواهد آمد.

مثال ۱۷-۳ : شمارش موجودی انبار با شمارش ورودی و خروجی

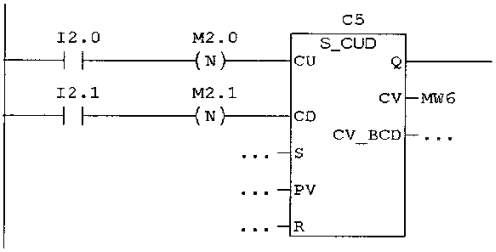
در شکل ۱۷-۸ سیستمی با دو نوار نقاله و یک انبار میانی نشان داده شده است. نوار۱ بسته‌های محصول را به این انبار تحویل می‌دهد. بعد از عبور بسته از روبروی سنسور فتوالکتریک که در انتهای نوار ۱ قرار دارد، شمارنده انبار بایستی یکی افزایش یابد.

نوار ۲ بسته‌ها را از انبار موقت به بیرون انتقال می‌دهد. با عبور جسم از روبروی سنسور فتوالکتریک که در ابتدای نوار ۲ قرار گرفته شمارنده یکی کاهش می‌یابد.



شکل ۱۷-۸ فرآیند مثال ۱۷-۳

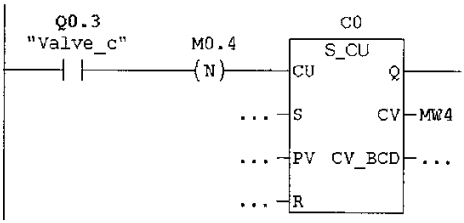
برنامه در شکل ۱۷-۹ با استفاده از کانتر S_CUD آمده است. دقت شود که شمارش بایستی پس از عبور بسته یعنی روی لبه پایین‌رونده سنسور انجام شود به همین علت از دستور <N> استفاده شده است. موجودی انبار در MW6 در دسترس خواهد بود.



شکل ۱۷-۹ برنامه مثال ۱۷-۳

تمرین ۱۷-۱: در برنامه فوق خروجی CV را به MW2 منتقل کنید و با سیمولاتور تست کنید که شمارش درست انجام نمی‌شود. علت چیست؟

مثال ۱۷-۴: در برنامه کنترل میکسر فصل قبل (مثال ۱۶-۱۲) اگر نیاز باشد که تعداد سیکل‌ها شمارش شود، لازم است پس از بسته‌شدن Valve_C یعنی با لبه پایین‌رونده، کانتر یک شماره افزایش یابد بنابراین می‌توان برنامه زیر را به برنامه مثال ۱۶-۱۲ جهت شمارش سیکل‌ها و ذخیره در MW4 اضافه نمود.



شکل ۱۷-۱۰ برنامه مثال ۱۷-۴

تمرین ۱۷-۲: در مثال ۱۷-۴ ماندگاری کانتر C0 را با قطع و وصل تغذیه چک کنید. توجه شود که شماره کانتر C0 انتخاب شده و به‌صورت پیش‌فرض ماندگار است.

مثال ۱۷-۵ : زمان‌سنجی با ترکیب تایمر و کانتر

در بحث تایمرها ذکر شد که ماکزیمم زمان تایمر دو ساعت و چهل و شش دقیقه است. اگر نیاز به زمان‌های بالاتر باشد یک راه ترکیب تایمر و کانتر است. ترکیب تایمر و کانتر به دو طریق قابل انجام است:

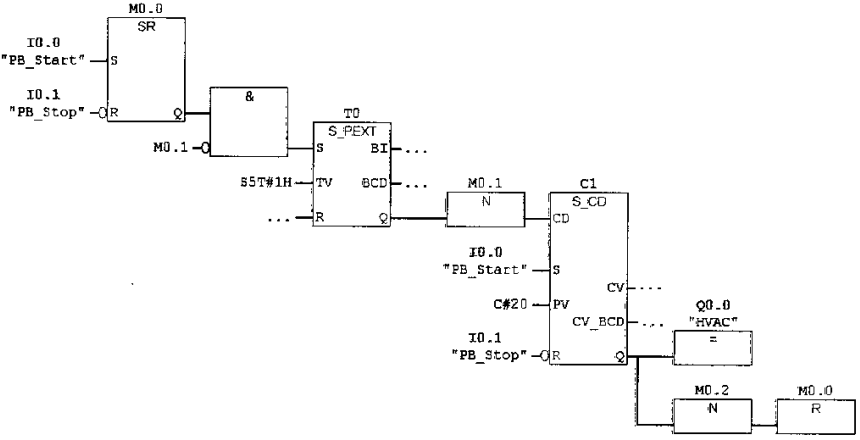
۱- ترکیب تایمر و کانتر افزایشی به‌صورتی که هر بار تایمر زمان‌سنجی را به‌پایان می‌رساند کانتر یک شماره افزایش یابد و تایمر از نو شروع کند.

۲- ترکیب تایمر و کانتر کاهشی به‌صورتی که کانتر با مقدار اولیه ست شود و با هر بار عملکرد تایمر یک شماره از کانتر کم شود تا به صفر برسد.

روش اول را در اینجا نمی‌توانیم شرح دهیم چون نیاز به دستورات مقایسه داریم که هنوز در مورد آنها بحث نکرده‌ایم ولی روش دوم را می‌توان توضیح داد.

در مثال زیر با شستی استارت اپراتور سیستم تهویه مطبوع (HVAC) به مدت ۲۰ ساعت کار می‌کند سپس خاموش می‌شود. برای این منظور از تایمر PEXT و کانتر CD استفاده شده است. روش کار بدین‌صورت است که:

- با فعال‌شدن شستی استارت، کانتر با مقدار اولیه ۲۰ ست می‌شود و تایمر نیز با زمان ۱ ساعت به‌کار می‌افتد.
- وقتی تایمر از کار افتاد (با لبه پایین رونده) کانتر یک شماره کاهش می‌یابد و مجدداً تایمر به‌کار می‌افتد.
- این سیکل آنقدر تکرار می‌شود که کانتر صفر شود. وقتی کانتر صفر شد خروجی Q آن از یک به صفر تغییر حالت می‌دهد، از لبه منفی این خروجی برای برداشتن فرمان استفاده شده است.



شکل ۱۷-۱۱ برنامه مثال ۱۷-۵

به‌سادگی با تغییر مقدار اولیه کانتر و زمان تایمر می‌توان زمان‌های دیگری را نیز به‌دست آورد. به‌عنوان نمونه با ست کردن مقدار 999 به کانتر و زمان 2h به تایمر، سیستم به مدت 1998 ساعت یعنی حدود ۸۳ روز کار خواهد کرد.

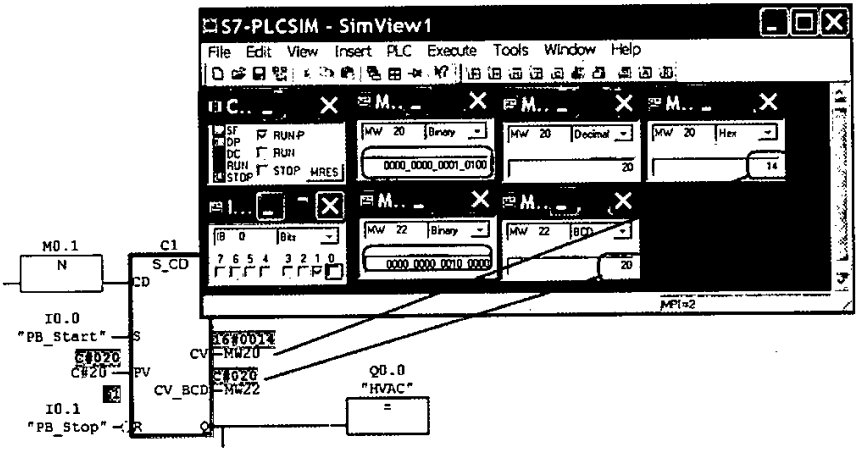
برای تست برنامه فوق با سیمولاتور زمان تایمر را به یک ثانیه تغییر دهید و خروجی را چک کنید.

۱۷-۴ بررسی خروجی‌های CV و CV_BCD کانتر

در هر دو خروجی CV و CV_BCD مقدار شمارش شده توسط کانتر وجود دارند که می‌توان آنها را به حافظه ۱۶ بیتی دلخواهی متصل نمود.

تفاوت بین این دو خروجی در این است که CV مقدار شمارش شده را به‌صورت هگز یا دسیمال نشان می‌دهد. در حالی که CV_BCD مقدار را به‌صورت عدد BCD نمایش می‌دهد ولی عملاً این دو مقدار معادل هم هستند و به یک شماره اشاره می‌کنند.

مثال ۱۷- ۶: شکل زیر کانتر C1 برنامه مثال ۱۷-۵ را نشان می‌دهد که خروجی‌های کانتر به MW20 و MW22 منتقل شده‌اند. اگر در لحظه‌ای که مقدار کانتر ۲۰ است مقادیر MW20 و MW22 را با سیمولاتور به‌صورت بیتی ببینیم، شکل ۱۷-۱۲ را خواهیم داشت.



شکل ۱۷-۱۲ برنامه مثال ۱۷-۶

همانطور که دیده می‌شود، در MW20 عدد زیر وجود دارد که با تبدیل آن به مبنای ۱۰ معادل ۲۰ خواهد بود.

Binary	**0000**	**0000**	**0001**	**0100**
Hex	**0**	**0**	**1**	**4**
Decimal	**20**			

در MW22 عدد زیر وجود دارد که مفهوم BCD 20 را دارد.

Binary	**0000**	**0000**	**0010**	**0000**
BCD	**0**	**0**	**2**	**0**

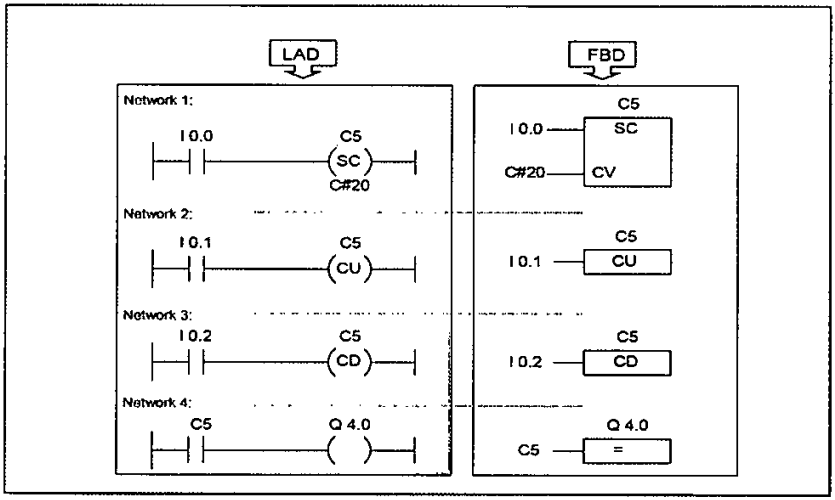
۱۷-۵ دستورات بیتی کانتر[1]

این دستورات به‌صورت بیتی بوده و توسط آنها می‌توان برخی از اعمالی که در توابع کانتر قابل انجام است را پیاده‌سازی نمود. این دستورات عبارتند از:

1. Counters: Bit Instrutions

- **SC:** عملکرد آن مانند پایه‌های S و PV یک کانتر می‌باشد.
- **CU:** عملکرد آن مانند پایه‌ی CU در کانتر می‌باشد.
- **CD:** عملکرد آن مانند پایه‌ی CD در کانتر می‌باشد.

در شکل ۱۷-۷ چگونگی استفاده از این دستورات همراه با یک مثال ارائه شده است.



شکل ۱۷-۷ دستورات بیتی کانتر

در برنامه‌ی نشان‌داده شده در شکل ۱۷-۷:

- با فعال‌شدن I0.0، مقدار 20 به خروجی کانتر C5 انتقال پیدا می‌نماید.
- با فعال‌شدن I0.1، یک‌عدد به مقدار شمارش‌شده توسط کانتر C5 اضافه می‌شود.
- با فعال شدن I0.2، یک عدد از مقدار شمارش‌شده توسط کانتر C5 کم می‌شود.
- در صورتی‌که کانتر C5 فعال باشد (مقداری غیر از صفر داشته باشد)، Q4.0 فعال می‌شود.

نکته: در این روش، نمی‌توان مستقیماً به عدد شمارش شده توسط کانتر دسترسی پیدا نمود. البته از روش‌های غیرمستقیم امکان دسترسی به عدد شمارش شده توسط کانتر وجود دارد. یکی از این روش‌ها استفاده از دستور Move است که در فصل ۲۱ توضیح داده می‌شود.

۱۷-۶ تمرین

۱- در یک پروسه صنعتی ابتدا ۱۰۰ جعبه در داخل انبار قرار گرفته و سپس جعبه‌ها از انبار خارج می‌شوند. خروج جعبه‌ها از انبار توسط سنسور I124.0 مشخص می‌شود (تینه باز)، برنامه‌ای بنویسید که موجودی انبار در MW2 نشان داده شده و هرگاه انبار خالی شد لامپ Q124.0 به‌صورت چشمک‌زن روشن شود.

فصل ۱۸

دستورات برنامه‌نویسی مقایسه‌گرها

۱۸-۱ مقدمه
۱۸-۲ انواع مقایسه‌گر
۱۸-۳ عملکرد مقایسه‌گرها
۱۸-۴ تمرین

در این فصل با نحوه برنامه‌نویسی و نکات مربوط به مقایسه‌گرها در زبان‌های LAD و FBD آشنا می‌شوید.

چکیده مطالب

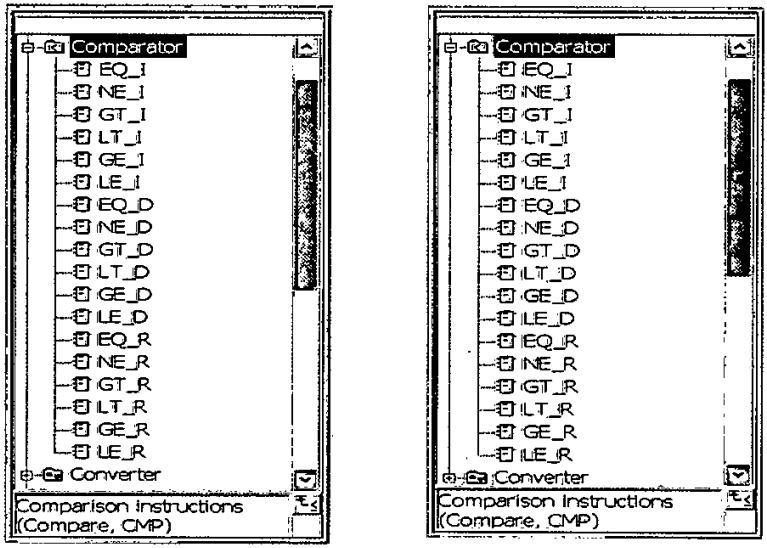
- مقایسه‌گر می‌تواند دو مقدار، یا دو متغیر یا یک متغیر با یک مقدار را مقایسه کند.
- مقایسه‌گر می‌تواند مقایسه بین دو مقدار یا متغیر صحیح ۱۶ بیتی انجام دهد. این مقایسه‌گرها با حرف I مشخص می شوند.
- مقایسه‌گر می‌تواند مقایسه بین دو مقدار یا متغیر صحیح ۳۲ بیتی انجام دهد. این مقایسه‌گرها با حرف DI مشخص می شوند.
- مقایسه‌گر می‌تواند مقایسه بین دو مقدار یا متغیر اعشاری انجام دهد. این مقایسه‌گرها با حرف R مشخص می‌شوند.
- برای تمام حالات مقایسه دستور مستقلی وجود دارد.

فصل ۱۸

۱۸-۱ مقدمه

مقایسه‌گرها از دستورات بسیار پرکاربرد در برنامه‌نویسی هستند. در فصل‌های قبل که با بیت سروکار داشتیم نیازی به مقایسه‌گر نبود. بیت دو حالت صفر و یک بیشتر ندارد و بر اساس این دو حالت می‌توان فرمان لازم را تولید کرد. در عمل با متغیرهایی سروکار داریم که از بیت بزرگتر هستند و در آنها مقادیری به‌صورت عدد صحیح یا عدد اعشاری ذخیره می‌گردد. بدیهی است این متغیرها دو وضعیتی نیستند و برای تولید فرمان بر اساس شرایط آن‌ها نیاز به مقایسه‌گر داریم.

اگر در پنجره Program Element بخش Comparator را باز کنیم زیر مجموعه آنها را به‌صورت شکل ۱۸-۱ خواهیم دید. همانطور که دیده می‌شود المان‌های مقایسه برای LAD و FBD مشابه هستند.



شکل ۱۸-۱ محل قرارگیری دستورات مقایسه‌گر

۱۸-۲ انواع مقایسه‌گر

اگر به لیست دستورات مقایسه‌گر توجه کنیم می‌بینیم که در انتهای دستورات یکی از حروف I و D و R وجود دارد که نشان دهنده نوع متغیر یا عددی است که می‌توان به ورودی این مقایسه‌گرها اختصاص داد. مفهوم این حروف به‌شرح زیر است:

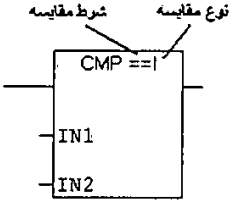
- I: متغیر از نوع Integer، منظور از آن عدد صحیح ۱۶ بیتی می‌باشد.
- D: متغیر از نوع Double Integer، منظور از آن عدد صحیح ۳۲ بیتی می‌باشد.
- R: متغیر از نوع Real ، منظور از آن اعداد اعشاری (ممیز دار) ۳۲ بیتی می‌باشد.

به‌جز این سه نوع، مقایسه‌گر دیگری وجود ندارد. به‌عنوان مثال برای اعداد صحیح ۸ بیتی یا اعداد BCD مقایسه‌گر وجود ندارد و بایستی آنها را به یکی از این سه مورد تبدیل نمود تا بتوانند مورد مقایسه قرار گیرند.

تمام المان‌های مقایسه دارای دو ورودی مانند شکل هستند:

IN1: ورودی اول مقایسه‌گر

IN2: ورودی دوم مقایسه گر



شکل ۱۸-۲ نمونه مقایسه‌گر

فصل ۱۸

حالت‌های مختلف مقایسه

حالت‌های مختلف مقایسه مانند بزرگتر، کوچکتر، مساوی از نام المان به‌شرح جدول ۱۹-۱ مشخص می‌گردد.

جدول ۱۸-۱

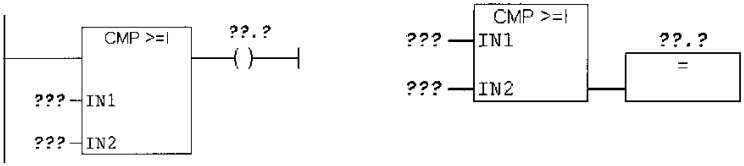
عنوان	مفهوم	علامت	شرح
EQ	Equal	= =	مساوی بودن دو وروی بررسی می‌شود
NE	Not Equal	< >	مخالف بودن دو ورودی بررسی می‌شود
GT	Greater Than	>	بزرگتر بودن IN1 نسبت به IN2 بررسی می‌شود
LT	Less Than	<	کوچکتر بودن IN1 نسبت به IN2 بررسی می‌شود
GE	Greater than or Equal	> =	بزرگتر یا مساوی بودن IN1 نسبت به IN2 بررسی می‌شود
LE	Less than or Equal	< =	کوچکتر یا مساوی بودن IN1 نسبت به IN2 بررسی می‌شود

۱۸-۳ عملکرد مقایسه‌گرها

هر مقایسه‌گر ورودی اول را با ورودی دوم طبق شرط داده شده مقایسه می‌کند. اگر شرط برقرار باشد خروجی آن یک می شود در غیر اینصورت خروجی صفر خواهد بود.

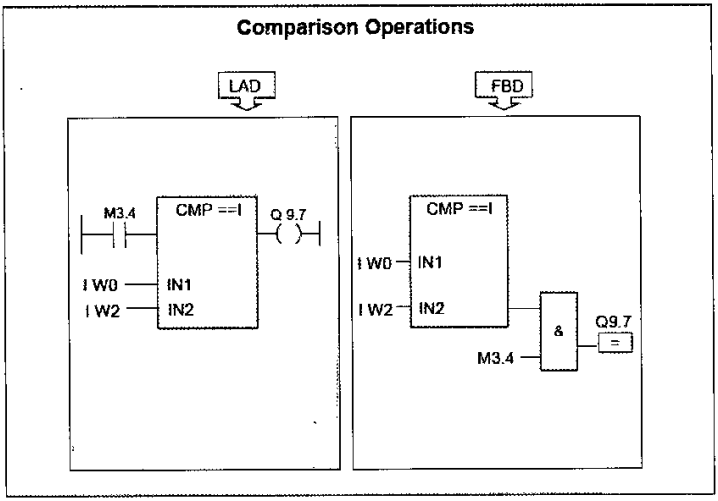
پس خروجی مقایسه‌گرها از جنس Bool است و یک بیت است که از RLO مقدار می‌گیرد، برقرار بودن شرط RLO را یک می‌کند.

خروجی مقایسه‌گر را نمی‌توان بلاتکلیف رها کرد؛ لازم است آنرا به ادامه لاجیک یا به خروجی مورد نظر اتصال داد.



شکل ۱۸-۳ مقایسه‌گر در LAD و FBD

شکل ۱۸-۴ نمونه دیگری از مقایسه‌گرها را نشان می‌دهد.



شکل۱۸-۴ مقایسه‌گر در زبان LAD و FBD

همانطور که در شکل ۱۸-۴ مشاهده می‌شود، مقایسه‌گر از نوع Integer است و شرط مساوی بودن مقدار IW0 را با مقدار IW2 بررسی می‌کند. بنابراین در صورتی‌که مقدار IW0 و IW2 با هم برابر باشد خروجی مقایسه‌گر فعال می‌شود. در زبان LAD مشاهده می‌کنیم که ورودی مقایسه‌گر به کنتاکت M3.4 وصل شده بنابراین خروجی Q9.7 در صورتی یک می‌شود که M3.4 یک باشد و شرط مقایسه نیز برقرار باشد. این مفهوم در زبان FBD با AND خروجی مقایسه‌گر و کنتاکت M3.4 بیان شده است.

نکاتی که در ارتباط با مقایسه‌گرها حایز اهمیت است:

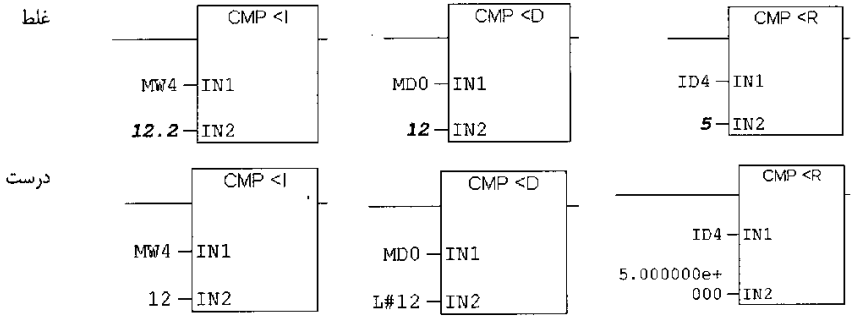
- در مقایسه‌گرها همواره ورودی اول نسبت به ورودی دوم سنجیده می‌شود. در دستورات مساوی و مخالف، جابه‌جا شدن جای دو متغیر تأثیری در نتیجه ندارد ولی در دستورات بزرگتر و کوچکتر اگر جای متغیر ورودی اول و دوم تغییر کند مفهوم مقایسه برعکس خواهد شد. به مثال شکل ۱۸-۵ دقت کنید.

شکل ۱۸-۵ اهمیت ترتیب ورودی‌ها

- در مقایسه‌گرها می‌توان دو متغیر را با یکدیگر مقایسه کرد یا یک متغیر را با یک عدد دلخواه مقایسه نمود. به شکل ۱۸-۶ توجه کنید.

شکل ۱۸-۶ انواع حالت‌های دو ورودی

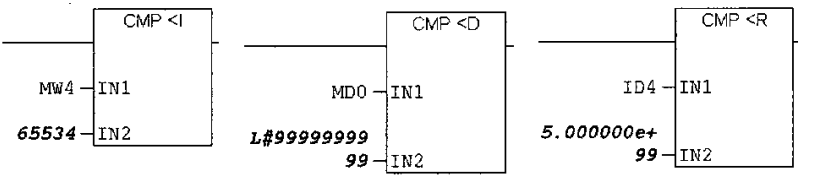
- اگر فرمت عدد با نوع مقایسه‌گر تطبیق نداشته باشد به رنگ قرمز ظاهر می‌شود که قابل استفاده نیست. به نمونه‌های شکل ۱۸-۷ دقت کنید.



شکل ۱۸-۷ فرمت اعداد در ورودی مقایسه‌گرها

- در استفاده از مقایسه‌گرهای Integer اعداد بدون ممیز و بدون علامت خاصی اختصاص می‌یابند.
- در استفاده از مقایسه‌گرهای Double Integer بایستی قبل از عدد صحیح علامت L# را قرار داد. L مخفف Long Integer است.

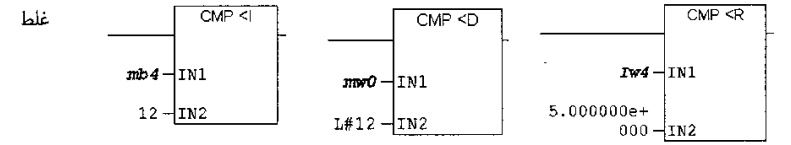
- در استفاده از مقایسه‌گرهای Real بایستی اعداد با ممیز به‌کار روند. پس از اختصاص آنها به پایه مقایسه‌گر برنامه آنها را به فرمت Exponential تبدیل می‌کند.
- در برخی حالات با وجود رعایت نکات فوق باز می‌بینیم که عدد به رنگ قرمز در می‌آید مانند شکل ۱۸-۸. به‌نظر شما مشکل کجاست؟



شکل ۱۸-۸ اعداد خارج از رنج در ورودی مقایسه‌گرها

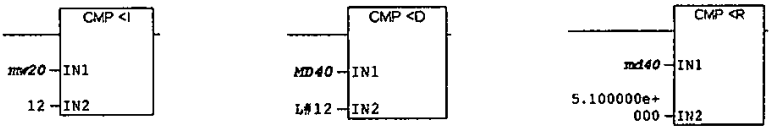
پاسخ این است که هر عدد دارای حد مینیمم و ماکزیمم است، اگر از این حدود تجاوز شود و عددی بزرگتر یا کوچکتر داده شود مقایسه‌گر قبول نمی‌کند. به‌عنوان مثال ماکزیمم عدد صحیح ۱۶ بیتی با علامت که برای مقایسه‌گر Integer قابل قبول است عدد 32767 می‌باشد و عددی که در شکل ۱۸-۸ دیده می‌شود از این حد بزرگتر است.

- اگر نوع متغیر با نوع مقایسه‌گر همخوانی نداشته باشد به رنگ قرمز ظاهر می‌شود و قابل ذخیره‌سازی و دانلود نیست نمونه‌های زیر. حالت درست در شکل ۱۸-۸ نشان داده شده است.



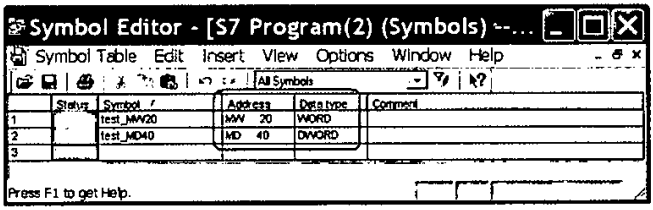
شکل ۱۸-۹ غلط بودن فرمت متغیر در ورودی مقایسه‌گر

- در مقایسه‌گر Integer متغیر بایستی ۱۶ بیتی یعنی به‌صورت Word باشد بنابراین آدرس‌های مجاز عبارتند از IW و PIW و QW و MW و LW و DBW.
- در مقایسه‌گرهای Double Integer و Real متغیر بایستی ۳۲ بیتی یعنی Double Word باشد بنابراین آدرس‌های مجاز عبارتند ازID و PID و QD و MD و LD و DBD.
- آدرس‌های PQW و PQD در مقایسه‌گرها قابل استفاده نیستند.
- در برخی اوقات با وجود رعایت نکات فوق دیده می‌شود که مقایسه‌گر آدرس را قبول نمی‌کند و به رنگ قرمز نمایش می‌دهد. مانند نمونه‌های شکل ۱۸-۱۰. به‌نظر شما مشکل کجاست؟



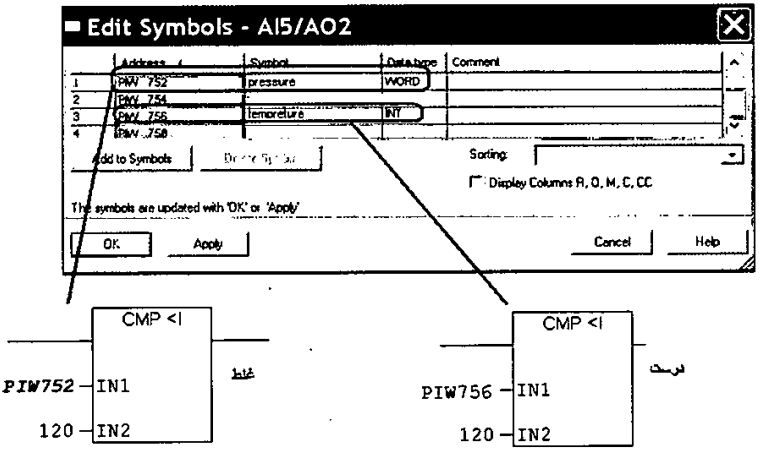
شکل ۱۸-۱۰ تعریف سمبولیک متغیر به‌صورت نادرست

پاسخ این است که متغیرها به‌طور معمول به نوع خاصی اشاره نمی‌کنند، مثلاً یک متغیر MW می‌تواند از نوع Integer یا از نوع Word باشد. تا زمانی‌که سمبل برای این متغیر تعریف نشده، هم در دستورات Integer و هم در دستورات Word قابل استفاده است ولی اگر در برنامه یا در جدول سمبل‌ها نوع مشخصی برای آن تعریف شود فقط در همان نوع دستورات قابل استفاده است. شکل ۱۸-۱۱ جدول سمبل‌ها را نشان می‌دهد. توجه کنید که برای سه متغیر شکل قبل چه نوع متغیری تعریف شده است.



شکل ۱۸-۱۱ تعریف متغیرها در جدول سمبل‌ها

این موضوع در مورد ورودی‌های آنالوگ نیز صادق است. اگر در پنجره HW Config روی کارت AI کلیک راست و گزینه Edit Symbols را انتخاب کرده باشیم و مانند شکل ۱۸-۱۲ به کانال‌ها سمبل اختصاص دهیم، بایستی به نوع دیتا Data Type دقت کنیم. شکل ۱۸-۱۲ دو نوع تعریف دیتا را برای آنالوگ نشان می‌دهد و همانطور که دیده می‌شود نوع Word را نمی‌توان در مقایسه‌گر استفاده کرد.

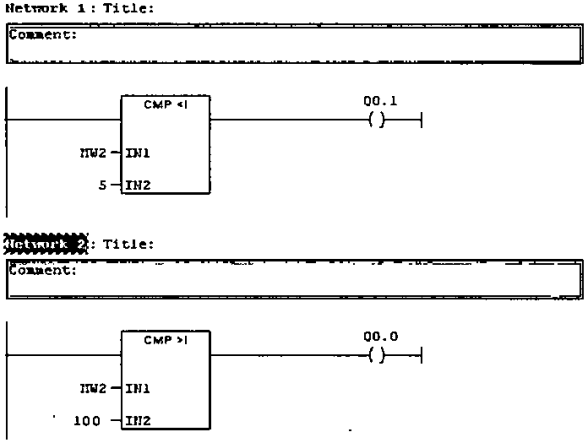


شکل ۱۸-۱۲ تعریف آنالوگ ورودی در HW Config

شایان ذکر است که در برخی نرم‌افزارها مانند PCS7 آدرس‌های آنالوگ فوق بهتر است به‌صورت word باشند زیرا برخی فانکشن‌هایی که برای آنالوگ تعبیه شده‌اند این فرمت را می‌پذیرند.

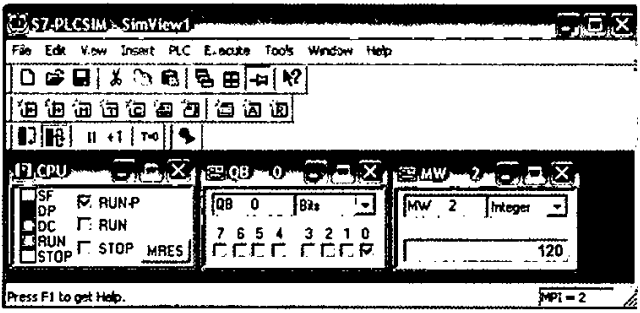
مثال ۱۸-۱: در یک پروسه‌ی صنعتی تعداد جعبه‌های ورودی به انبار، توسط یک کانتر شمرده شده و به‌صورت یک عدد INT در حافظه MW2 ذخیره می‌شود. برنامه‌ای بنویسید که اگر تعداد جعبه‌های موجود در انبار بیشتر از ۱۰۰ جعبه شد، خروجی Q0.0 روشن شود. اگر تعداد جعبه‌های درون انبار کمتر از ۵ جعبه شد، خروجی Q0.1 روشن شود.

حل: برنامه‌ی مورد نظر در OB1 نوشته می‌شود. به‌منظور پیاده‌سازی برنامه، نیاز به استفاده از دو مقایسه‌گر می‌باشد. مقیاسه‌گر اول کوچکتر بودن MW2 (تعداد جعبه‌های درون انبار) را از عدد ۵ بررسی نموده و مقایسه‌گر دوم بزرگتر بودن عدد درون MW2 را از عدد ۱۰۰ بررسی می‌نماید. شکل ۱۸-۱۳ این برنامه را نشان می‌دهد.



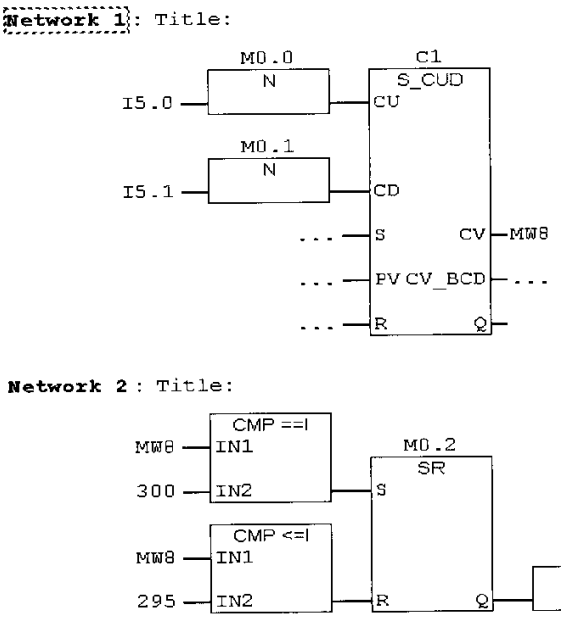
شکل۱۸-۱۳ برنامه‌ی مورد نظر جهت مثال ۱۸-۱

پس از نوشتن برنامه‌ی فوق، برنامه را توسط سیمولاتور تست نمایید. می‌توانید به MW2 به‌صورت دستی مقدار بدهید و نتیجه را مشاهده نمایید. همانطور که در شکل ۱۸-۱۴ مشاهده می‌شود، مقدار MW2 برابر عدد ۱۲۰ وارد شده است، بنابراین Q0.0 فعال شده است.



شکل۱۸-۱۴ محیط نرم‌افزار PLCSIM و متغیرهای ایجاد شده در آن به‌منظور تست مثال ۱۸-۱

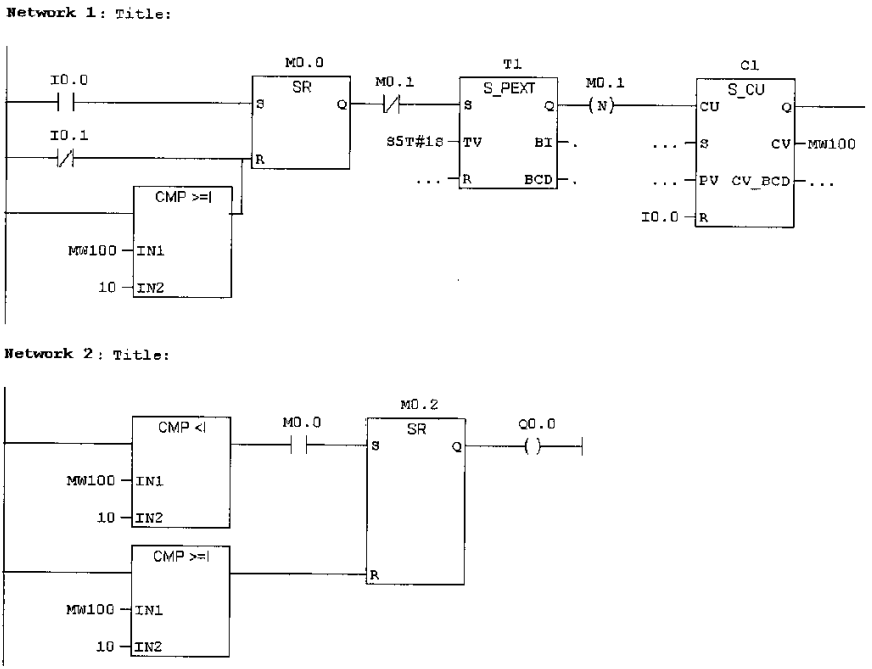
مثال ۱۸-۲: یک پارکینگ گنجایش ۳۰۰ دستگاه ماشین را دارد. ورود و خروج ماشین‌ها با دو سنسور جداگانه چک می‌شود. برنامه‌ای بنویسید که اگر تعداد ماشین‌های موجود به ۳۰۰ رسید، گیت ورودی بسته شود و وقتی از ۲۹۵ کمتر شد گیت باز شود.



شکل ۱۸-۱۵ برنامه‌ی مثال ۱۸-۲

مثال ۱۸-۳: با ترکیب تایمر و کانتر برنامه‌ای بنویسید که پس از فعال شدن شستی استارت سیستم HVAC به مدت ۱۰ ساعت کار کند.

این برنامه در فصل قبل با استفاده از کانتر کاهشی و تایمر نوشته شد. در اینجا با استفاده از کانتر افزایشی و تایمر ارائه می شود که اگرچه مفصل‌تر از راه حل قبلی است ولی برای آشنایی با مقایسه‌گرها مفید است.



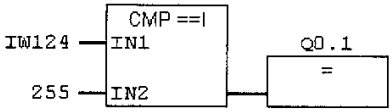
شکل ۱۸-۱۶ برنامه‌ی مثال ۱۸-۳

مثال ۱۸-۴: برنامه‌ای بنویسید که اگر ۸ ورودی آخر یک کارت دیجیتال ۱۶ ورودی که آدرس شروع آن در سخت افزار ۱۲۴ است همه یک باشند خروجی Q0.1 روشن شود.

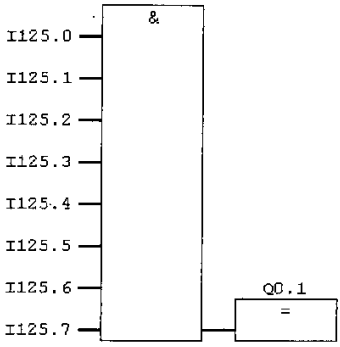
می‌توان به‌سادگی ۸ ورودی را با هم AND کرد و به خروجی اختصاص داد. در عین حال می‌توان با مقایسه‌گر برنامه را نوشت. وقتی ۸ ورودی آخر یک و ۸ ورودی اول صفر باشند داریم:

IW124	
IB124	IB125
00000000	11111111

پس IW124 معادل عدد ۱۶ بیتی 0000000011111111 خواهد بود که اگر آنرا به دسیمال تبدیل کنیم معادل 255 است بنابراین برنامه به‌صورت زیر خواهد بود.



با استفاده از مقایسه‌گر



با استفاده از منطق بیتی

شکل۱۸-۱۷ برنامه‌ی مثال ۱۸-۴

فصل ۱۸

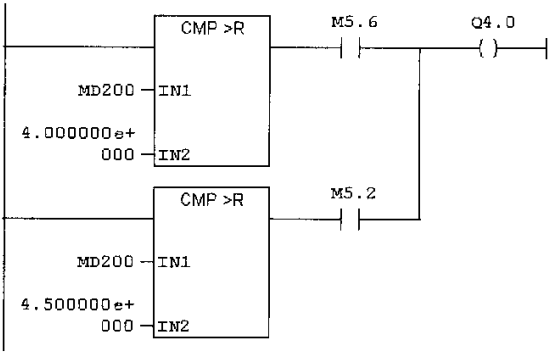
تمرین ۱۸-۱: برنامه فوق را با استفاده از مقایسه‌گر به‌صورتی بنویسید که اگر تمام ۱۶ ورودی یک شد خروجی Q0.1 روشن شود.

مثال ۱۸-۵: در فرآیندی سطح یک مخزن توسط ترانسمیتر سطح اندازه‌گیری می‌شود. سیگنال این ترانسمیتر به روشی که در کتاب سطح پیشرفته تشریح خواهد شد Scale شده و در بازه 0 تا 5 متر قرار می‌گیرد و به‌صورت یک عدد اعشاری در MD200 ذخیره می‌شود. برنامه‌ای بنویسید که:

وقتی سطح از ۴ متر بالاتر رفت آلارم چشمک‌زن با فرکانس کم روی Q4.0 ظاهر شود.

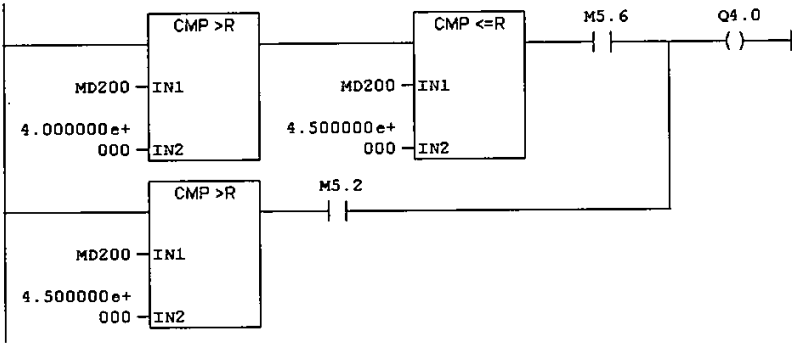
وقتی سطح از ۴/۵ متر بالاتر رفت Q4.0 با فرکانس سریع چشمک‌زن شود.

به روشی که قبلاً ذکر شد پالس CPU را فعال و به MB5 اختصاص می‌دهیم. اگر برنامه را به‌صورت شکل ۱۸-۱۸ بنویسیم مشکلی وجود دارد.



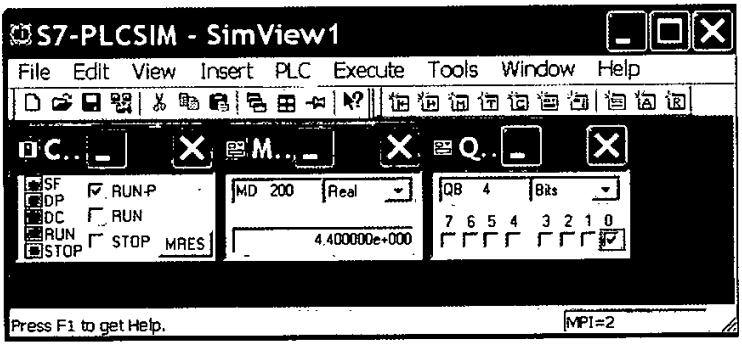
شکل۱۸-۱۸ برنامه‌ی نادرست برای مثال ۱۸-۵

مشکل این است که وقتی سطح از ۴ متر بیشتر و از ۴/۵ متر کمتر است فقط خروجی مقایسه‌گر بالایی فعال است ولی وقتی سطح از ۴/۵ متر بیشتر می‌شود خروجی هر دو مقایسه‌گر فعال است که پالس‌ها با یکدیگر تلاقی کرده و خروجی به‌صورت منظم خاموش و روش نمی‌شود. برای رفع مشکل فوق بایستی برنامه را به‌صورتی بنویسیم که در این شرایط شرط بالایی فعال نشود. روش درست برنامه مطابق شکل ۱۸-۱۹ است. بررسی کنید.



شکل۱۸-۱۹ برنامه‌ی درست برای مثال ۱۸-۵

می‌توانید برنامه را با سیمولاتور چک کنید. کافیست MD200 را در آن وارد کرده و نوع دیتا را Real انتخاب کنید و به آن مقدار بدهید. (شکل ۱۸-۲۰)



شکل۱۸-۲۰ تست برنامه‌ی مثال ۱۸-۵ با سیمولاتور

۱۸-۴ تمرین

۱- برنامه‌ای طراحی کنید که اگر یک موتور در طول شیفت کاری خود بیشتر از ۱۰ بار خاموش و روشن شد، تا یک ساعت امکان روشن نمودن مجدد آن وجود نداشته باشد.

۲- در یک پروسه صنعتی موتور ۱ جعبه‌های ساخته شده را وارد انبار موقت نموده و موتور ۲ آنها را از انبار خارج می‌کند. ورود جعبه‌ها به انبار با سنسور I124.0 و خروج جعبه‌ها توسط سنسور I124.1 تشخیص داده می‌شود. برنامه‌ای بنویسید که اگر موجودی انبار بیشتر از ۱۰ جعبه شد، موتور ۱ خاموش و اگر کمتر از ۵ جعبه شد، موتور ۱ روشن شود.

۳- برای کنترل سطح یک مخزن از سه پمپ استفاده شده که سیال را به داخل آن ارسال می‌کنند. سطح توسط ترانسمیتر اندازه‌گیری شده و درنهایت به‌صورت عدد اعشاری در MD100 بین صفر تا ۵ متر ذخیره می‌گردد. برنامه‌ای بنویسید که:

- اگر سطح کمتر از 1m باشد هر سه پمپ کار کنند.
- اگر سطح بین 1m تا 2m باشد فقط دو پمپ کار کنند.
- اگر سطح بین 2m تا 3m باشد فقط یک پمپ کار کند.
- اگر سطح بیش از 3m باشد هر سه پمپ خاموش شوند.

۴- برنامه قبل را به‌صورتی کامل کنید که تعداد ساعت کارکرد پمپ‌ها ثبت شود و وقتی سطح بیش از ۲ متر است پمپی که کمتر کار کرده روشن باشد.

فصل ۱۸

فصل ۱۹

دستورات محاسباتی

۱۹-۱ مقدمه
۱۹-۲ دستورات محاسباتی Integer
۱۹-۲-۱ دستورات محاسباتی اعداد صحیح ۱۶ بیتی
۱۹-۲-۲ دستورات محاسباتی اعداد صحیح ۳۲ بیتی
۱۹-۳ دستورات محاسباتی Real
۱۹-۴ تمرین

در این فصل با نحوه برنامه‌نویسی و نکات مربوط به دستورات محاسباتی و توابع مربوط به اعداد و متغیرهای صحیح و اعشاری در زبان‌های LAD و FBD آشنا می‌شوید.

چکیده مطالب

- دستورات محاسباتی برای انجام عملیات ریاضی در برنامه به‌کار می‌روند.
- این دستورات به سه دسته تقسیم می‌شوند.
- دستورات محاسباتی عدد صحیح ۱۶ بیتی که با علامت I مشخص می‌شوند شامل ۴ عمل اصلی جمع و تفریق و ضرب و تقسیم هستند.
- دستورات محاسباتی عدد صحیح ۳۲ بیتی که با علامت DI مشخص می‌شوند علاوه بر ۴ عمل اصلی، باقیمانده تقسیم دو عدد را نیز محاسبه می‌کنند.
- دستورات محاسباتی اعداد اعشاری علاوه بر ۴ عمل اصلی، فانکشن ریاضی مانند توابع مثلثاتی و جذر و قدر مطلق را نیز در بر می‌گیرند. فقط در دستورات ۴ عمل اصلی علامت R وجود دارد.

فصل
۱۹

چکیده مطالب

- دستورات محاسباتی برای انجام عملیات ریاضی در برنامه به‌کار می‌روند.
- این دستورات به سه دسته تقسیم می‌شوند.
- دستورات محاسباتی عدد صحیح ۱۶ بیتی که با علامت I مشخص می‌شوند شامل ۴ عمل اصلی جمع و تفریق و ضرب و تقسیم هستند.
- دستورات محاسباتی عدد صحیح ۳۲ بیتی که با علامت DI مشخص می‌شوند علاوه بر ۴ عمل اصلی، باقیمانده تقسیم دو عدد را نیز محاسبه می‌کنند.
- دستورات محاسباتی اعداد اعشاری علاوه بر ۴ عمل اصلی، فانکشن ریاضی مانند توابع مثلثاتی و جذر و قدر مطلق را نیز در بر می‌گیرند. فقط در دستورات ۴ عمل اصلی علامت R وجود دارد.

فصل
۱۹

۱۹-۱ مقدمه

در برنامه‌نویسی گاهی نیاز به استفاده از دستورات محاسبات ریاضی وجود دارد. جمع، تفریق، ضرب، تقسیم، قدر مطلق، توابع مثلثاتی و ... از جمله دستورات محاسباتی هستند.

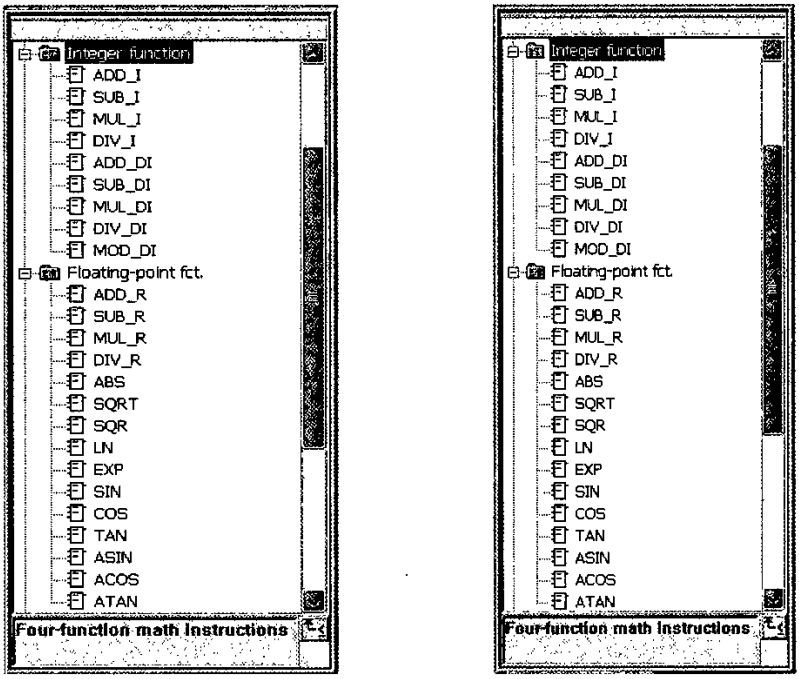
در محیط نرم‌افزار Step7 دستورات محاسباتی به دو دسته اصلی تقسیم شده‌اند:

- Integer Function
- Floating Point Function

دسته اول دستوراتی هستند که فقط روی اعداد صحیح ۱۶ بیتی یا ۳۲ بیتی به‌کار می‌روند.

دسته دوم دستوراتی هستند که روی اعداد اعشاری کاربرد دارند.

اگر پنجره Program Element بخش مربوط به این دستورات را باز کنیم زیر مجموعه آنها را به‌صورت شکل ۱۹-۱ خواهیم دید. همانطور که دیده می‌شود هیچ تفاوتی بین المان‌های محاسباتی LAD و FBD وجود ندارد.



شکل ۱۹-۱ دستورات محاسباتی در Program Element

۱۹-۲ دستورات محاسباتی Integer

این دستورات به دو گروه تقسیم می‌شوند:

Integer: دستورات ریاضی که جهت اعداد صحیح 16 بیتی به‌کار می‌روند فقط شامل چهار عمل اصلی ریاضی می‌باشند. در انتهای کد این دستورات حرف I دیده می‌شود.

Double Integer: دستورات ریاضی که جهت اعداد صحیح 32 بیتی به‌کار می‌روند شامل چهار عمل اصلی ریاضی و دستور باقیمانده تقسیم می‌باشند. در انتهای کد این دستورات حرف DI دیده می‌شود.

۱۹-۲-۱ دستورات محاسباتی اعداد صحیح ۱۶ بیتی

همانطور که در شکل ۱۹-۱ مشاهده می‌شود، این دستورات در جدول ۱۹-۱ آورده شده‌اند:

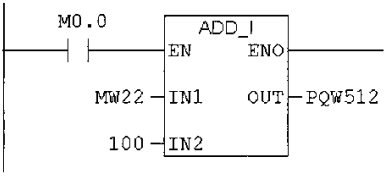
جدول ۱۹-۱

عنوان	مفهوم	شرح
ADD_I	Add Integer	جمع کننده دو عدد INT
SUB_I	Subtract Integer	تفریق کننده دو عدد INT
MUL_I	Multiply Integer	ضرب کننده دو عدد INT
DIV_I	Divide Integer	تقسیم کننده دو عدد INT

همه دستورات فوق دارای دو ورودی(IN1 , IN2) و یک خروجی (OUT) برای محاسبات هستند. علاوه بر اینها یک ورودی EN و یک خروجی ENO نیز دارند که از جنس Bool می‌باشند. تشریح ورودی‌ها و خروجی‌ها در زیر آمده است:

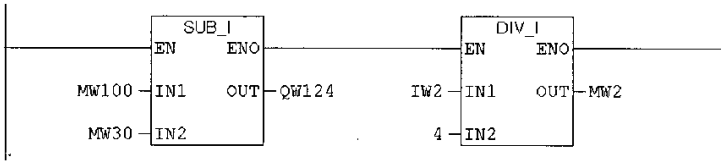
- **EN:** معرف Enable یعنی فعال‌کننده‌ی دستور است از جنس بیت می‌باشد. به‌طور پیش‌فرض وقتی EN به بیت یا لاجیکی متصل نیست دستور فعال است. می‌توان در صورت لزوم آنرا به کنتاکت یا لاجیک مورد نظر متصل کرد. در این‌حالت شرط اجرای دستور محاسباتی یک شدن RLO در پایه EN می‌باشد.

به‌عنوان مثال در شکل ۱۹-۲ تا M0.0 یک نشود دستور جمع اجرا نمی‌گردد.



شکل ۱۹-۲ اختصاص EN جمع کننده به یک کنتاکت

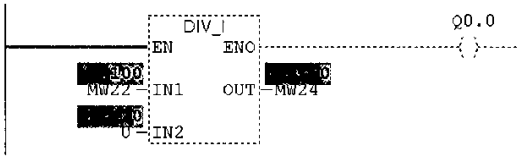
در شکل ۱۹-۳ دستور تفریق بدون هیچ شرطی فعال می‌شود ولی دستور تقسیم در صورتی اجرا می‌شود که قبل از آن تفریق بدون مشکل انجام شده باشد.



شکل ۱۹-۳ اجرای عمل تقسیم بعد از عمل جمع

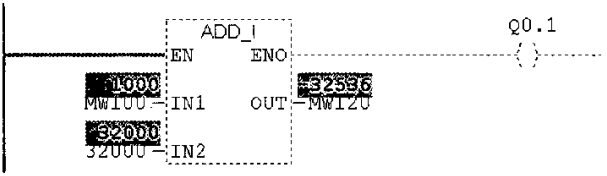
- **ENO:** در صورتی‌که پایه‌ی EN فعال شده باشد و در اجرای دستور مشکلی به‌وجود نیامده باشد، ENO دارای مقدار یک منطقی می‌شود.

اگر نتیجه عمل محاسباتی خارج از رنج باشد خروجی ENO صفر می‌شود. در شکل ۱۹-۴ چون تقسیم بر صفر اتفاق افتاده است ENO صفر می‌باشد.



شکل ۱۹-۴ اشکال در تقسیم بر صفر

در شکل ۱۹-۵ نتیجه جمع از ماکزیمم مقدار Integer یعنی عدد 32767 بیشتر شده در نتیجه ENO صفر گردیده است.



شکل ۱۹-۵ اشکال در سرریزی

- **IN1:** ورودی اول دستور محاسباتی است.
- **IN2:** ورودی دوم دستور محاسباتی است.

توجه شود که:

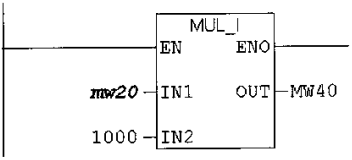
- در محاسباتی مانند تفریق و تقسیم جابه‌جایی در ورودی‌های IN1 و IN2 نتیجه را تغییر می‌دهد.
- به ورودی آنها می‌توان اعداد صحیح ۱۶ بیتی (مثبت یا منفی) اختصاص داد.

- به ورودی آنها می‌توان متغیرهای ۱۶ بیتی مانند IW ، PIW ، QW ، MW ، LW و DBW اختصاص داد؛ ولی آدرس PQW در ورودی مجاز نیست.

- **OUT:** حاصل عملیات محاسباتی انجام شده را بر می‌گرداند.

- به خروجی این دستورات می‌توان متغیرهای ۱۶ بیتی مانند IW، QW، PQW، MW، LW و DBW اختصاص داد؛ ولی آدرس PIW در خروجی مجاز نیست.

طبق نکته‌ای که در بحث مقایسه‌گرها مطرح شد، اگر قبلاً به متغیر ۱۶ بیتی آدرس سمبولیک از نوع غیر Integer داده شده باشد، دستورات محاسباتی Integer آنرا قبول نمی‌کند. شکل ۱۹-۶ MW20 که از جنس Word تعریف شده را نشان می‌دهد.



شکل ۱۹-۶ اشکال در نوع متغیر که ازجنس Word تعریف شده

فصل ۱۹

۱۹-۲-۲ دستورات محاسباتی اعداد صحیح ۳۲ بیتی

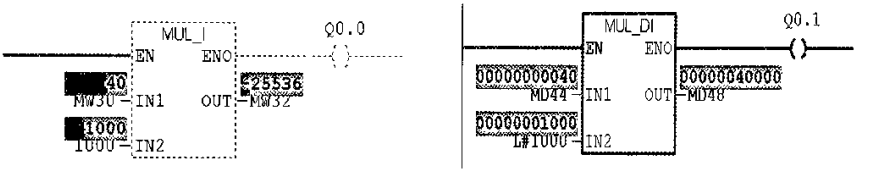
این دستورات در جدول ۱۹-۲ آورده شده‌اند.

جدول ۱۹-۲

عنوان	مفهوم	شرح
ADD_DI	Add Double Integer	جمع کننده دو عدد DINT
SUB_DI	Subtract Double Integer	تفریق کننده دو عدد DINT
MUL_DI	Multiply Double Integer	ضرب کننده دو عدد DINT
DIV_DI	Divide Double Integer	تقسیم کننده دو عدد DINT
MOD_DI	Fraction Double Integer	باقیمانده تقسیم دو عدد DINT

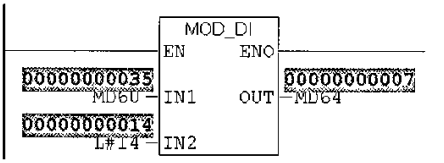
ورودی و خروجی این دستورات نیز مشابه نوع INT است که قبلاً توضیح داده شد. نکته‌ای که بایستی دقت کرد این است که در اختصاص اعداد به ورودی‌ها بایستی علامت L# را که معرف Long Integer است قبل از آنها به‌کار برد.

از آنجا که این دستورات برای ۳۲ بیت استفاده می‌شوند، اعداد بزرگتری را قبول می‌کنند. مثال زیر ضرب دو عدد را با MUL_I و MUL_DI مقایسه کرده است. در تابع اولی نتیجه خارج از رنج و خروجی صفر شده در حالی که در دومی مشکلی وجود ندارد و ENO یک است.



شکل ۱۹-۷ مقایسه ضرب کننده‌های ۱۶ بیتی با ۳۲ بیتی

بلاک MOD_DI باقیمانده تقسیم ورودی اول بر ورودی دوم را بر می‌گرداند. این دستور در نوع Integer وجود نداشت. مثال شکل ۱۹-۸ باقیمانده تقسیم ۳۵ بر ۱۴ را نشان می‌دهد.



شکل ۱۹-۸ دستور MOD_DI

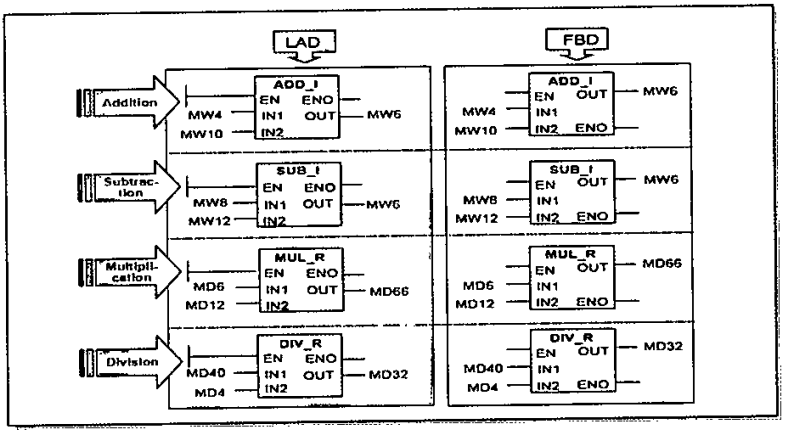
۱۹-۳ دستورات محاسباتی Real

این دستورات در جدول ۱۹-۳ آورده شده‌اند.

جدول ۱۹-۳

عنوان	مفهوم	شرح
ADD_R	Add Real	جمع‌کننده دو عدد Real
SUB_R	Subtract Real	تفریق‌کننده دو عدد Real
MUL_R	Multiply Real	ضرب‌کننده دو عدد Real
DIV_R	Divide Real	تقسیم‌کننده دو عدد Real
ABS	Absolute	قدر مطلق یک عدد Real
SQRT	Square Root	ریشه دوم (جذر) یک Real
SQR	Square	توان دوم یک عدد Real
LN	Natural Logarithm	لگاریتم طبیعی یک عدد Real
EXP	Exponential Value	عدد e (۲،۷۱۸۲۸) به‌توان دلخواه
SIN	Sine Value	سینوس یک عدد Real بر حسب رادیان
COS	Cosine Value	کسینوس یک عدد Real بر حسب رادیان
TAN	Tangent Value	تانژانت یک عدد Real بر حسب رادیان
ASIN	Arc Sine Value	آرک سینوس یک عدد Real
ACOS	Arc Cosine Value	آرک کسینوس یک عدد Real
ATAN	Arc Tangent Value	آرک تانژانت یک عدد Real

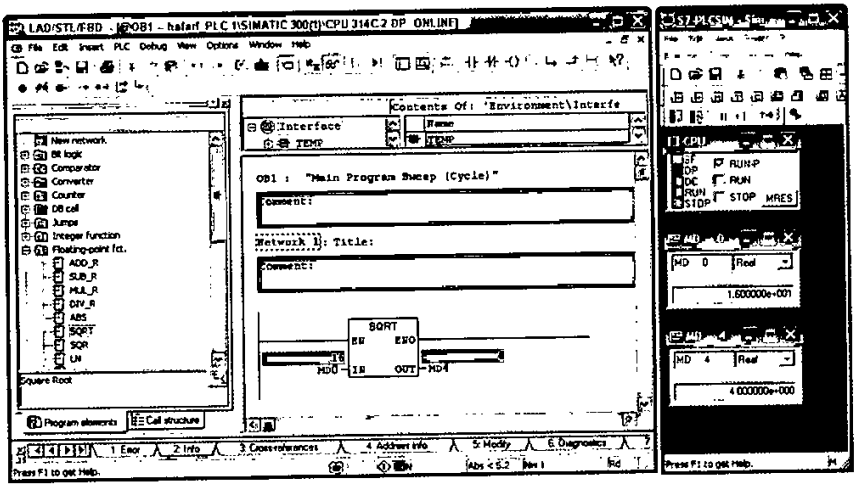
در بین دستورات فوق دستورات مربوط به چهار عمل اصلی جمع و ضرب و تفریق و تقسیم دارای دو ورودی و دو خروجی شبیه نوع INT و DINT هستند. شکل ۱۹-۹ دستورات ریاضی پایه، مخصوص اعداد Integer و Real را مقایسه کرده است.



شکل ۱۹-۹ دستورات پایه اعداد Integer و Real

فصل ۱۹

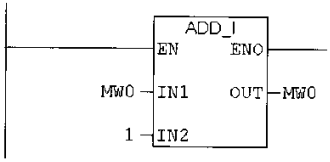
به‌جز چهار عمل اصلی سایر دستورات خانواده Real توابعی هستند که یک ورودی و یک خروجی دارند. شکل ۱۹-۱۰ تابع جذر را نشان می‌دهد که مقدار ۱۶ را از طریق سیمولاتور گرفته و جذر آن را محاسبه نموده است.



شکل ۱۹-۱۰ تست فانکشن جذر

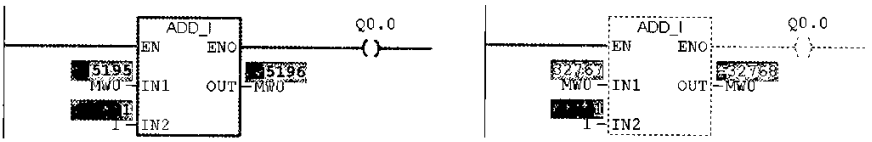
در استفاده از دستورات Real بایستی دقت کرد که عدد به فرمت ممیزدار وارد شود؛ در غیر اینصورت با رنگ قرمز یعنی غیر مجاز ظاهر می‌شود.

مثال۱۹-۱: عملکرد برنامه ساده زیر چگونه است؟



شکل ۱۹-۱۱ برنامه مثال ۱۹-۱

ابتدا مقدار MW0 صفر است، پس از آن در هر سیکل اسکن به MW0 یک عدد اضافه شده و به خود MW0 منتقل می‌شود. اگر MW0 را در حالت Online مشاهده کنیم می‌بینیم که به‌سرعت مقدار آن رو به افزایش است. سرعت تغییرات بستگی به زمان سیکل اسکن CPU دارد. هرچه زمان فوق کمتر باشد سرعت تغییرات MW0 بیشتر است.
تا زمانی که نتیجه جمع از ۳۲۷۶۷ بیشتر نشده خروجی ENO یک است وقتی عدد از ۳۲۷۶۷ یک شماره بالاتر رود منفی شده و به 32768- باز می‌گردد در این لحظه ENO صفر می‌شود. در سیکل بعد که به 32768- یک شماره اضافه می‌شود ENO مجدداً یک شده و روال کار عادی است پس فقط در نقطه سرریزی Overflow مقدار ENO صفر می‌شود.



شکل ۱۹-۱۲ برنامه مثال ۱۹-۱ در حالت Online

برای تحلیل این وضعیت لازم است MW0 را به‌صورت بیتی مورد بررسی قرار دهیم. ماکزیمم عدد ۱۶ بیتی با علامت وقتی است که بیت با ارزش بیشتر صفر و بقیه بیت‌ها یک باشند.

0111111111111111 = 32767

وقتی به عدد فوق یک اضافه کنیم بیت شانزدهم یک و سایر بیت‌ها صفر می‌شوند که معادل ۳۲۷۶۸ است ولی چون بیت شانزدهم یک است از دیدگاه CPU عدد منفی است.

1000000000000000 = -32768

پس از منفی شدن عدد با هر بار افزایش عدد به سمت صفر نزدیک می‌شود تا اینکه مثبت شود پس همواره مقدار MW0 بین 32768- تا 32767 در حال تغییر خواهد بود.

مثال ۱۹-۲: تولید پالس با استفاده از جمع کننده

با استفاده از برنامه مثال ۱۹-۱ می‌توان از بیت‌های MW0 برای پالس کنترل نشده استفاده کرد. بیت M1.0 بالاترین فرکانس و بیت M0.7 کمترین فرکانس را دارا خواهد بود. شکل ۱۹-۱۳ ساختار بیت‌های MW0 را نشان می‌دهد.

MW0															
MB0								MB1							
M0.7							M0.0	M1.7							M1.0

شکل ۱۹-۱۳ ترتیب بیت‌ها در MW0

اکنون ۸ بیت سمت راست یعنی بیت‌های MB1 را در نظر بگیرید. در هر سیکل که عدد یک به MW0 اضافه می‌شود، بیت‌ها صفر و یک می‌شوند. جدول ۱۹-۴ این بیت‌ها را برای چند سیکل‌اسکن نشان می‌دهد.

جدول ۱۹-۴

	MB1							
Scan Cycle	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0	1
6	0	0	0	0	0	1	1	0
7	0	0	0	0	0	1	1	1
8	0	0	0	0	1	0	0	0
9	0	0	0	0	1	0	0	1
10	0	0	0	0	1	0	1	0
11	0	0	0	0	1	0	1	1
12	0	0	0	0	1	1	0	0

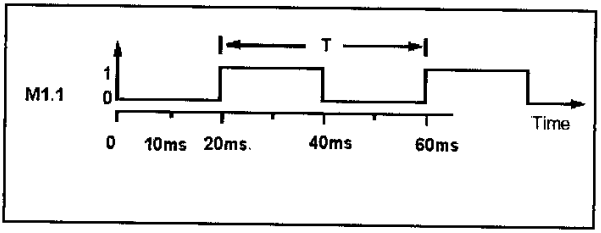
اکنون به ستون اول یعنی Bit 0 نگاه کنید. در هر سیکل وضعیت بیت عکس وضعیت قبلی خواهد بود. اگر به‌عنوان مثال سیکل اسکن ۱۰ میلی ثانیه باشد این بیت ۱۰ میلی ثانیه یک و ۱۰ میلی ثانیه صفر است.

برای تعیین فرکانس پالس کافیست با توجه به رابطه F=1/T زمان صفر و یک شدن را جمع زده و معکوس نماییم. بیت M1.0 به اندازه 10 میلی‌ثانیه (یا یک‌صدم ثانیه) صفر و باندازه 10 میلی‌ثانیه صفر است پس فرکانس آن 1/(0.01+0.01) که معادل 50 HZ خواهد بود.

اگر به ستون دوم یعنی Bit 1 نگاه کنید می‌بینید که هر دو سیکل این بیت تغییر وضعیت می‌دهد یعنی با فرض فوق این بیت ۲۰ میلی‌ثانیه روشن و ۲۰ میلی‌ثانیه صفر است. به‌عبارت دیگر فرکانس پالس آن نصف فرکانس Bit 0 و معادل 25 HZ خواهد بود.

Signal state of Bit 1 of MB 1 (M1.1)

Frequency = 1/T = 1/40ms = 25 HZ



شکل ۱۹-۱۴ فرکانس پالس در برنامه مثال ۱۹-۲

همینطور اگر به سایر ستونها توجه کنیم می‌بینیم که فرکانس صفر و یک شدن بیت‌ها نسبت به بیت ماقبل مرتباً نصف می‌شود.

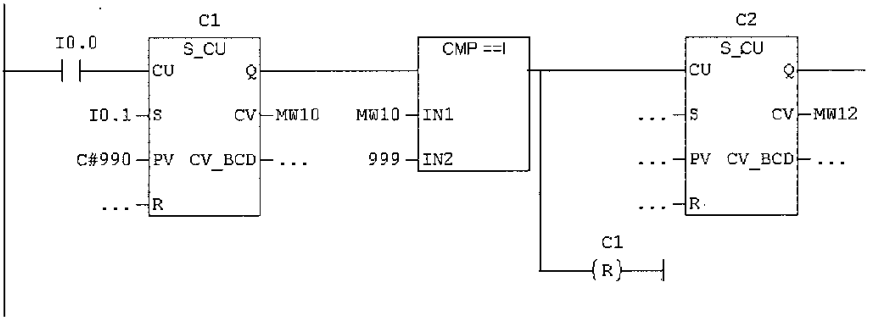
مسلم است که این پالس به‌صورت کنترل نشده است یعنی اگر حجم برنامه بیشتری به CPU دانلود شود زمان سیکل اسکن افزایش می‌یابد و فرکانس پالس‌ها کمتر می‌شود.

مثال ۱۹-۳: ترکیب دو کانتر برای شمارش بیش از ۹۹۹

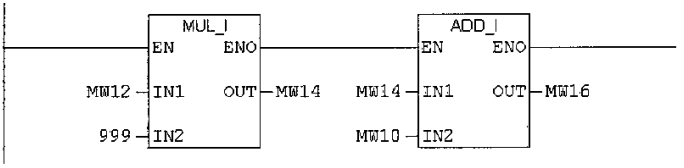
در فصل کانترها دیدیم که این دستورات فقط تا ۹۹۹ شمارش می‌کنند. اگر نیاز به شمارش بالاتر باشد می‌توان از ترکیب دو کانتر استفاده کرد. به این صورت که وقتی شمارش کانتر اول به ۹۹۹ رسید کانتر دوم یک شماره افزایش یابد و کانتر اول ری‌ست شود.

درنهایت عددی که در کانتر دوم به‌دست آمده در ۹۹۹ ضرب شده و با عدد کانتر اول جمع شود تا نتیجه شمارش به‌دست آید. نتیجه نهایی شمارش در MW16 در دسترس است. برنامه به‌صورت شکل ۱۹-۱۵ است.

```
OB1 :  "Main Program Sweep (Cycle)"
Network 1: Title:
```




```
Network 2: Title:
```



شکل ۱۹-۱۵ برنامه مثال ۱۹-۳

فصل ۱۹

مشکلی که در برنامه فوق وجود دارد این است که اگر نتیجه شمارش از ۳۲۷۶۷ بیشتر شود، دستورات ضرب و جمع با مشکل مواجه می‌شوند، ENO صفر شده و نتیجه غلط است. چه راهی برای رفع این مشکل وجود دارد؟ راهی که به ذهن می‌رسد این است که به‌جای انجام محاسبات به‌صورت INT آنها را به‌صورت DINT جمع کنیم ولی با مشکل دیگری مواجه می‌شویم و آن اینکه نتیجه شمارش کانترها به‌صورت ۱۶ بیتی است و نمی‌توان متغیرهای ۱۶ بیتی را به جمع و ضرب کننده ۳۲ بیتی داد. در اینجا نیاز به تبدیل از ۱۶ بیت به ۳۲ بیت وجود دارد که در فصل بعد با آن آشنا می‌شویم و این برنامه را کامل خواهیم کرد.

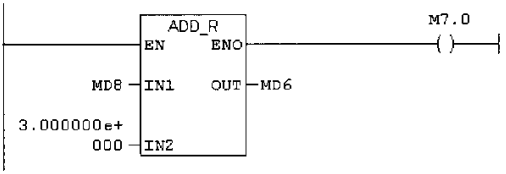
مثال ۱۹-۴: استفاده از جمع کننده برای شمارش به‌جای کانتر

به‌سادگی می‌توان به‌جای استفاده از کانتر که با مشکلاتی در شمارش‌های بالا همراه است از جمع‌کننده استفاده نمود. برنامه زیر را در نظر بگیرید. هر بار که لبه مثبت در ورودی I0.0 آشکار می‌شود جمع‌کننده فعال می‌شود و یک شماره افزایش می‌یابد.

با توجه به اینکه جمع‌کننده از نوع DINT است مقدار MD100 می‌تواند تا 147483647 که ماکزیمم عدد صحیح ۳۲ بیتی است افزایش یابد. آیا این روش ساده‌تر از به‌کار بردن کانترها نیست؟ ممکن است گفته شود که کانتر دارای ورودی ری‌ست است و در ضمن می‌تواند مقدار اولیه بگیرد. این کار نیز به‌سادگی امکان‌پذیر است در فصل‌های بعد با معرفی دستور MOVE این مثال را کامل می‌کنیم.

تمرین ۱۹-۱: اگر در برنامه مثال قبل دستور تشخیص لبه را حذف کنیم و I0.0 مستقیماً به جمع‌کننده متصل شود چه اتفاقی می‌افتد؟ چرا؟

تمرین ۱۹-۲: در برنامه زیر چه مشکلی وجود دارد؟ چرا نتیجه جمع درست نیست؟



شکل ۱۹-۱۶ برنامه تمرین ۱۹-۲

مثال ۱۹-۵: کنترل دما براساس انحراف از میانگین

در فرآیندی برای کنترل دمای اطراف یک قالب ریخته‌گری مذاب از ۴ عدد ترموکوپل استفاده شده است. نتیجه سیگنال‌های این چند ترموکوپل پس از Scale شدن در متغیرهای حافظه به‌شرح زیر ریخته شده است.

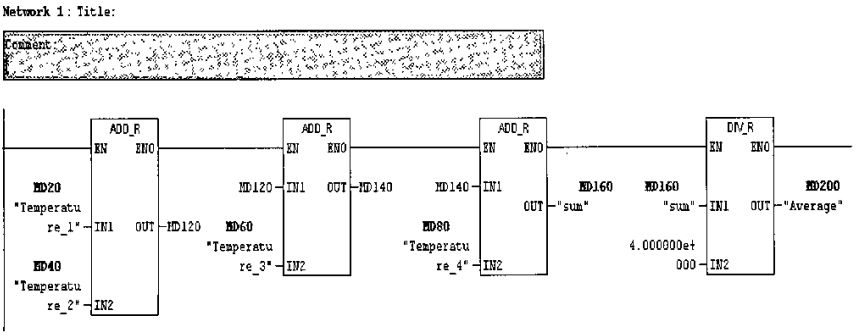
جدول ۱۹-۵

Temperature_1	MD20
Temperature_2	MD40
Temperature_3	MD60
Temperature_4	MD80

برنامه‌ای بنویسید که اگر دمای یک ترموکوپل بیش از ۵۰ درجه با دمای میانگین ترموکوپل‌ها اختلاف داشت (۵۰ درجه کمتر یا ۵۰ درجه بیشتر بود) خروجی آلارم Q0.0 روشن شود.

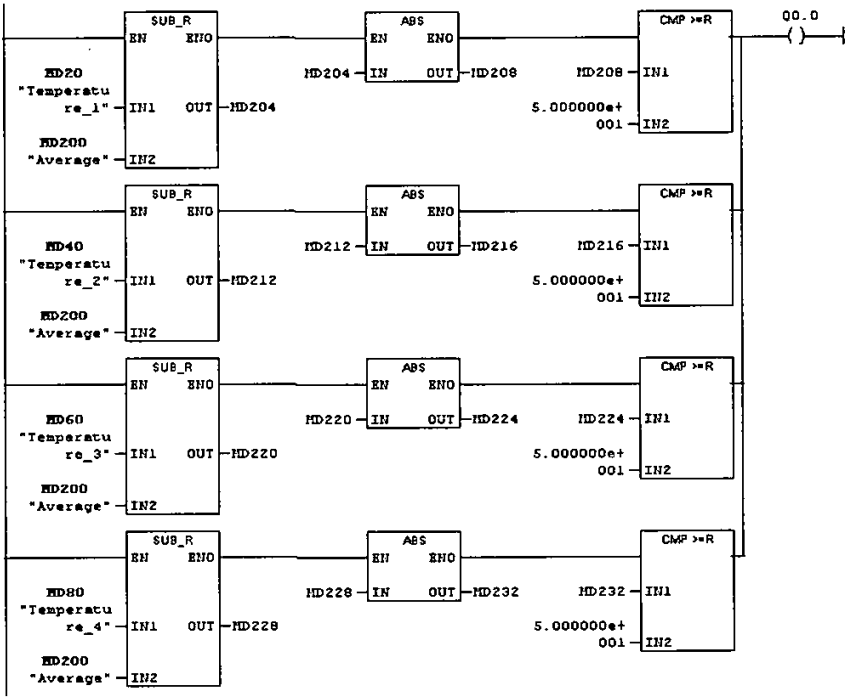
راه حل در برنامه زیر آمده است. در این برنامه برای محاسبه میانگین ابتدا بایستی ۴ سیگنال را با هم جمع و بر ۴ تقسیم کنیم. از آنجا که جمع‌کننده فقط دو ورودی دارد لازم است که چند بار از آن استفاده نماییم. پس از محاسبه میانگین، اختلاف بین هر آنالوگ با میانگین را به‌دست می‌آوریم و از آن قدر مطلق می‌گیریم تا اگر منفی بود مقایسه‌کننده بعد از آن درست عمل کند.

می‌توانید برنامه را با سیمولاتور تست کنید. MDها را وارد کرده و تغییر دهید و نتیجه را روی Q0.0 ببینید.

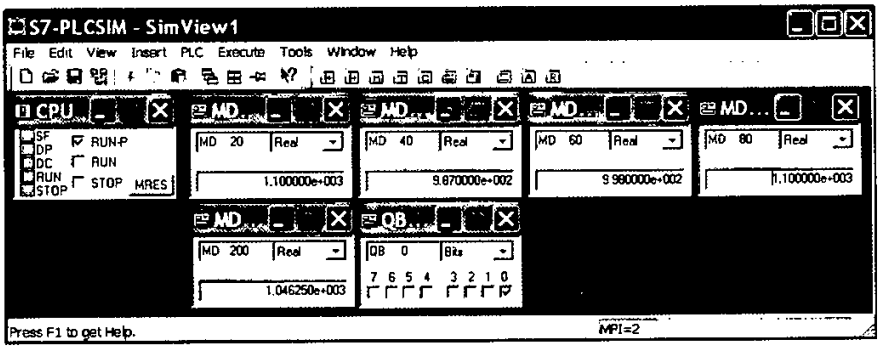


Network 2 : Title:

Comment:



شکل ۱۹-۱۷ برنامه مثال ۱۹-۵



شکل ۱۹-۱۸ تست برنامه مثال ۱۹-۵ با سیمولاتور

۴-۱۹ تمرین

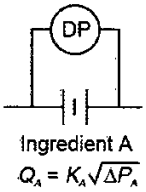
۱- با استفاده از دستورات جمع‌کننده، کانتری طراحی کنید که تا عدد 2^31 -1 را بشمارد.

۲- برنامه ای بنویسید که :

- مقدار MW0 هر ثانیه یکی افزایش یابد (ثانیه شمار)
- وقتی مقدار MW0 به ۶۰ رسید MW2 یکی افزایش یابد (دقیقه شمار)
- وقتی مقدار MW2 به ۶۰ رسید MW4 یکی افزایش یابد (ساعت شمار)
- وقتی مقدار MW4 به ۲۴ رسید MW6 یکی افزایش یابد (روز شمار)

۳- با استفاده از تمرین ۲ برنامه‌ای بنویسید که سیستمی را به مدت ۴ روز و ۱۰ ساعت و ۴۵ دقیقه روشن نگه دارد سپس آن را خاموش کند.

۴- در اندازه گیری وقتی از Orifice در مسیر سیال استفاده شود، می‌توان با ترانسمیتر اختلاف فشار دو طرف اریفیس را اندازه‌گیری کرد. مقدار فلو برابر است با جذر اختلاف فشار ضرب در ضریب Kبرنامه مربوطه را بنویسید.



فصل ۲۰

دستورات تبدیل

۲۰-۱ مقدمه

۲۰-۲ مبدل‌های فرمت اعداد به یکدیگر

۲۰-۲-۱ تبدیل BCD به Integer و بالعکس

۲۰-۲-۲ تبدیل BCD به Double Integer و بالعکس

۲۰-۲-۳ تبدیل Integer به Double Integer

۲۰-۲-۴ تبدیل Integer به Real

۲۰-۳ مبدل‌های متمم یک و دو (قرینه و معکوس‌سازی)

۲۰-۳-۱ دستور متمم یک (معکوس بیت‌های عدد)

۲۰-۳-۲ دستور متمم‌دو (قرینه‌ی عدد)

۲۰-۴ مبدل‌های Roundکننده اعداد اعشاری

۲۰-۵ تمرین

در این فصل با نحوه برنامه‌نویسی و نکات مربوط به دستورات تبدیل کننده فرمت‌های مختلف اعداد در زبان‌های LAD و FBD آشنا می‌شوید.

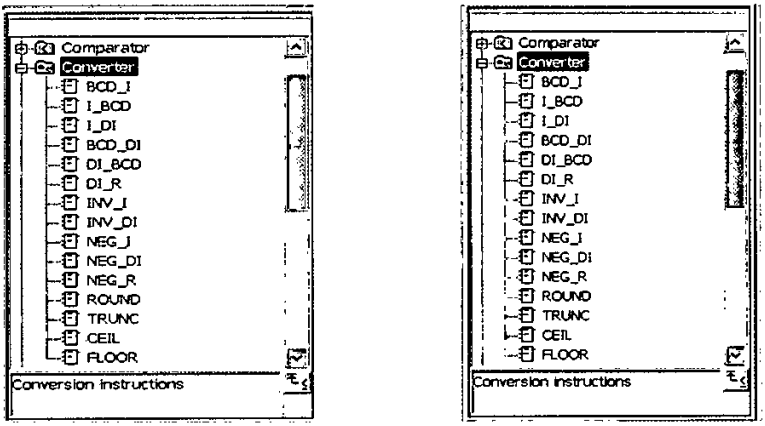
چکیده مطالب

	دستور تبدیل	توضیحات
مبدل‌های فرمت اعداد	BCD_I	تبدیل دیتا تایپ BCD به Integer
	I_BCD	تبدیل دیتا تایپ Integer به BCD
	I_DI	تبدیل دیتا تایپ Integer به Double Integer
	BCD_DI	تبدیل دیتا تایپ BCD به Double Integer
	DI_BCD	تبدیل دیتا تایپ Double Integer به BCD
	DI_R	تبدیل دیتا تایپ Double Integer به Real
متمم‌کننده‌ها	INV_I	متمم یک عدد صحیح ۱۶ بیتی (معکوس‌کردن بیت‌ها)
	INV_DI	متمم یک عدد صحیح ۳۲ بیتی (معکوس‌کردن بیت‌ها)
	NEG_I	متمم دوی عدد صحیح ۱۶ بیتی (قرینه عدد)
	NEG_DI	متمم دوی عدد صحیح ۳۲ بیتی (قرینه عدد)
	NEG_R	قرینه عدد اعشاری
رند کننده‌ها	ROUND	رند کردن عدد اعشاری به نزدیک‌ترین عدد صحیح
	TRUNC	رند کردن عدد اعشاری به عدد صحیح
	CEIL	رند کردن عدد اعشاری به عدد صحیح بالاتر
	FLOOR	رند کردن عدد اعشاری به عدد صحیح پایین‌تر

فصل ۲۰

۲۰-۱ مقدمه

دستورات تبدیل دستورات پرکاربردی نیستند ولی در برخی حالات، برنامه‌نویس به آنها نیاز پیدا می‌کند. نمونه‌ای از این نیاز در فصل قبل در مثال شمارش با دو کانتر توضیح داده شد. این دستورات در پنجره Program Element در زیر مجموعه Converter مانند شکل ۲۰-۱ قرار دارند و المان‌های آنها در LAD و FBD مشابه است.



شکل ۲۰-۱ دستورات تبدیل در Program Element

لیست این دستورات در جدول ۲۰-۱ آمده است.

جدول ۲۰-۱

گروه	عنوان	مفهوم	توضیحات
گروه اول مبدل‌های فرمت اعداد	BCD_I	BCD to Integer	تبدیل دیتا تایپ BCD به Integer
	I_BCD	Integer to BCD	تبدیل دیتا تایپ Integer به BCD
	I_DI	Integer to Double Integer	تبدیل دیتا تایپ Integer به Double Integer
	BCD_DI	BCD to Double Integer	تبدیل دیتا تایپ BCD به Double Integer
	DI_BCD	Double Integer to BCD	تبدیل دیتا تایپ Double Integer به BCD
	DI_R	Double Integer to Real	تبدیل دیتا تایپ Double Integer به Real
گروه دوم متمم‌کننده‌ها	INV_I	Ones Complement Integer	متمم یک عدد صحیح ۱۶ بیتی (معکوس کردن بیت‌ها)
	INV_DI	Ones Complement Double Integer	متمم یک عدد صحیح ۳۲ بیتی (معکوس کردن بیت‌ها)
	NEG_I	Twos Complement Integer	متمم دوی عدد صحیح ۱۶ بیتی (قرینه عدد)
	NEG_DI	Twos Complement Double Integer	متمم دوی عدد صحیح ۳۲ بیتی (قرینه عدد)

	NEG_R	Negate Real Number	قرینه عدد اعشاری
گروه سوم	ROUND	Round to Double Integer	رند کردن عدد اعشاری به نزدیک‌ترین عدد صحیح
	TRUNC	Truncate Double Integer Part	رند کردن عدد اعشاری به عدد صحیح
رند کننده‌ها	CEIL	Ceiling	رند کردن عدد اعشاری به عدد صحیح بالاتر
	FLOOR	Floor	رند کردن عدد اعشاری به عدد صحیح پایین‌تر

همه دستورات تبدیل فوق دارای یک ورودی و یک خروجی تبدیل هستند.

شکل ۲۰-۲ ورودی و خروجی دستورات تبدیل

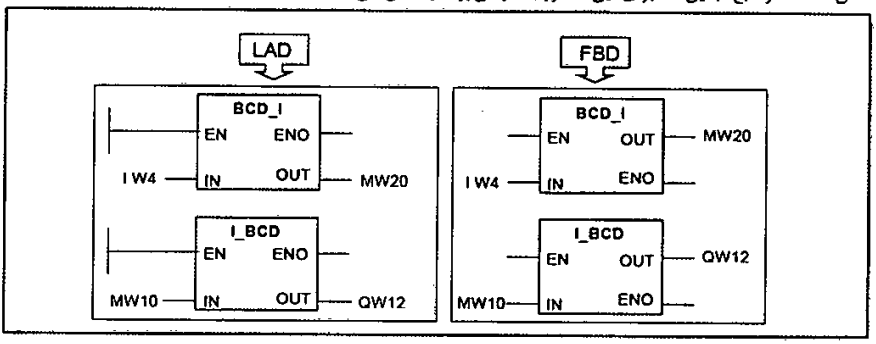
- ورودی IN عدد یا متغیر را می‌گیرد و خروجی OUT تبدیل شده آنرا تحویل می‌دهد.
- ورودی EN شبیه المان‌های قبل برای فعال‌سازی است.
- خروجی ENO وقتی تبدیل بدون مشکل انجام می‌شود یک، در غیر اینصورت صفر است.

۲۰-۲ مبدل‌های فرمت اعداد به یکدیگر

با استفاده از دستورات این گروه می‌تواند اعداد Integer را به BCD یا بالعکس تبدیل کرد؛ همینطور امکان تبدیل Integer به Real وجود دارد. این دستورات در ادامه توضیح داده می‌شوند.

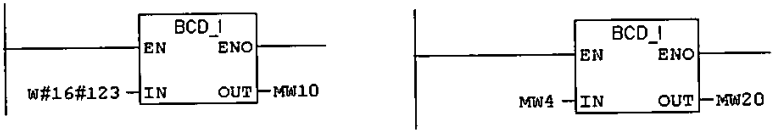
۲۰-۲-۱ تبدیل BCD به Integer و بالعکس

شکل ۲۰-۳ دو تابع تبدیل که برای این منظور به‌کار می‌روند را نشان می‌دهد.



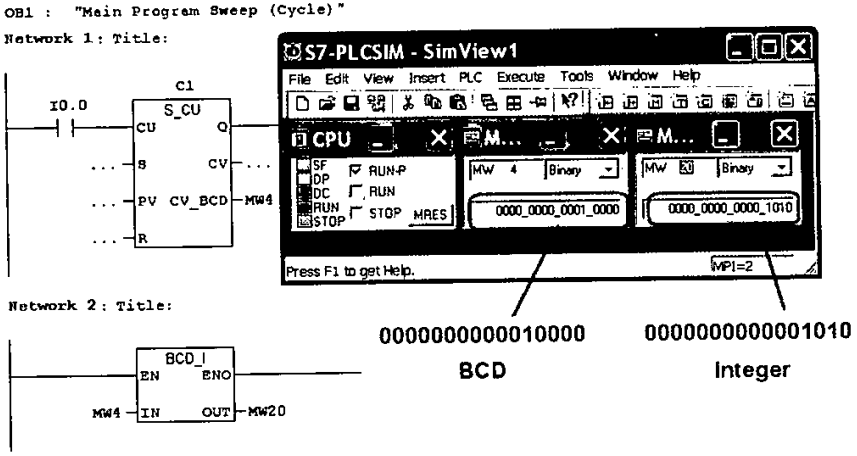
شکل ۲۰-۳ دستورات BCD_I و I_BCD

در مبدل I_BCD ورودی می‌تواند یک عدد BCD یا یک متغیر Word باشد. خروجی که یک عدد صحیح ۱۶ بیتی است را بایستی به متغیری ۱۶ بیتی اختصاص داد.



شکل ۲۰-۴ حالت‌های استفاده از مبدل I_BCD

مثال ۲۰-۱: برای درک بهتر تبدیل کننده فوق به مثال زیر دقت کنید. خروجی کانتر که به‌صورت BCD است را به MW4 انتقال داده‌ایم سپس آنرا با استفاده از تبدیل فوق به MW20 منتقل کرده‌ایم. اگر در حالتی که تعداد شمارش شده ۱۰ است بیت‌های MW4 و MW20 را ببینیم نحوه تبدیل را درک خواهیم کرد.

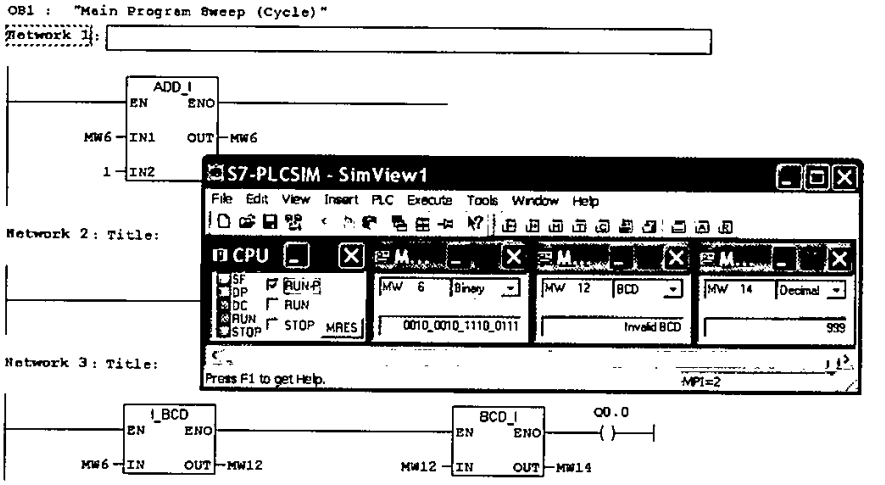


شکل ۲۰-۵ برنامه مثال ۲۰-۱

تبدیل I_BCD عکس دستور I_BCD عمل می‌نماید، یعنی یک داده‌ی Integer را به یک داده‌ی BCD تبدیل می‌نماید. از آنجا که ماکزیمم عدد BCD می‌تواند ۹۹۹ باشد که معادل 1001,1001,1001 باینری و ۲۴۵۷ دسیمال است بنابراین اگر عدد ورودی این تبدیل کننده از ۲۴۵۷ بیشتر شود خروجی از دیدگاه BCD نامعتبر خواهد بود. برای فهم بهتر موضوع به مثال زیر توجه کنید.

مثال ۲۰-۲: در برنامه زیر با استفاده از یک جمع‌کننده مقدار MW6 به‌طور دائم افزایش می‌یابد. MW6 را به مبدل I_BCD و مجدداً به مبدل BCD_I داده‌ایم. تا زمانی‌که MW6 کوچکتر از ۲۴۵۷ است عدد BCD درست و ENO یک است ولی وقتی MW6 بیش از مقدار فوق شود ENO صفر شده و مقدار تبدیل Invalid (نامعتبر) خواهد

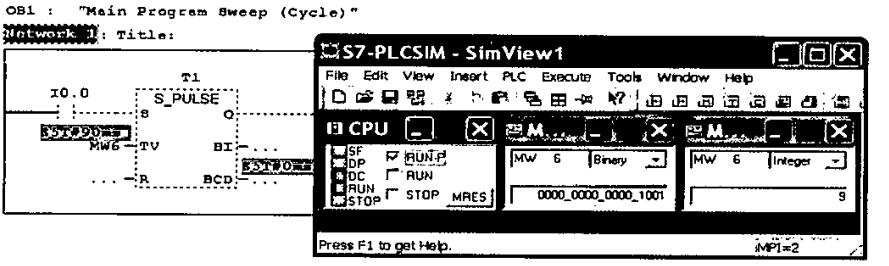
بود. برای فهم بهتر موضوع خروجی I_BCD را به ورودی BCD_I داده‌ایم. نتیجه‌ای که به‌دست می‌آید این است که اگر چه ورودی I_BCD یعنی MW6 درحال‌افزایش‌است ولی خروجی BCD_I یعنی MW14 روی عدد ۹۹۹ ثابت می‌ماند.



شکل ۲۰-۶ برنامه مثال ۲۰-۲

مثال ۲۰-۳: اختصاص زمان متغیر به تایمر

در برنامه زیر به ورودی تایمر پالس زمان را به‌صورت متغیر MW6 اختصاص داده‌ایم. مقدار MW6 را توسط سیمولاتور از صفر تا ۹ تغییر دهید. در حالت Online می‌بینیم که زمان بین صفر تا 90ms به ورودی تایمر داده شده است و مشکلی وجود ندارد.



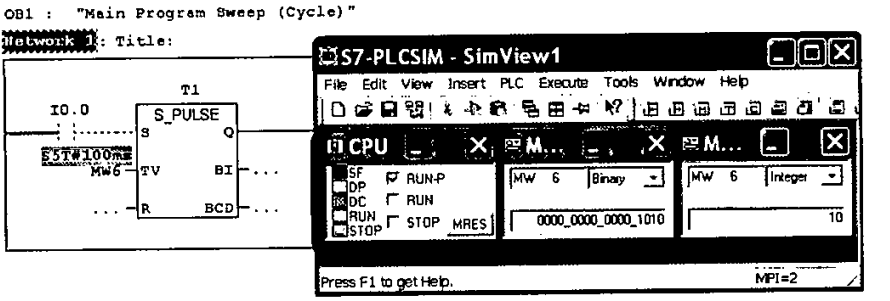
شکل ۲۰-۷ برنامه مثال ۲۰-۳

برای تحلیل برنامه وقتی MW6 معادل 9 است آنرا به‌صورت باینری مشاهده کنید. سپس ۱۲ رقم سمت راست را به‌صورت BCD تبدیل کنید، زیرا زمان تایمر به‌صورت BCD است. دو بیت بعدی معرف پله زمانی است که 00 معادل ۱۰ میلی‌ثانیه می‌باشد بنابراین زمان تایمر 90 ms است. به شکل ۲۰-۸ توجه کنید.

0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
				0				0				9			
		پله زمانی 10ms		9 BCD											

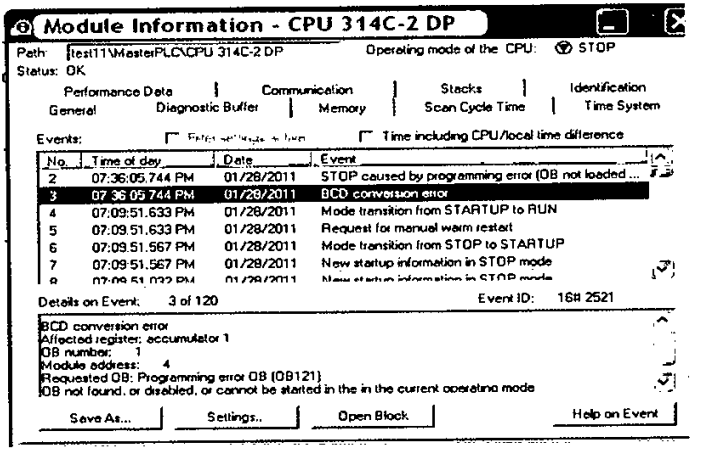
شکل ۲۰-۸ تحلیل بیت‌ها در برنامه مثال ۲۰-۳

اکنون با استفاده از سیمولاتور عدد 10 را به MW6 اختصاص دهید. با این کار CPU دچار توقف می‌شود و لامپ فالت آن (SF) روشن می‌گردد.



شکل ۲۰-۹ فالت و توقف CPU در برنامه مثال ۲۰-۳

در این شرایط در بافر پیغام BCD Conversion Error ثبت می‌گردد.



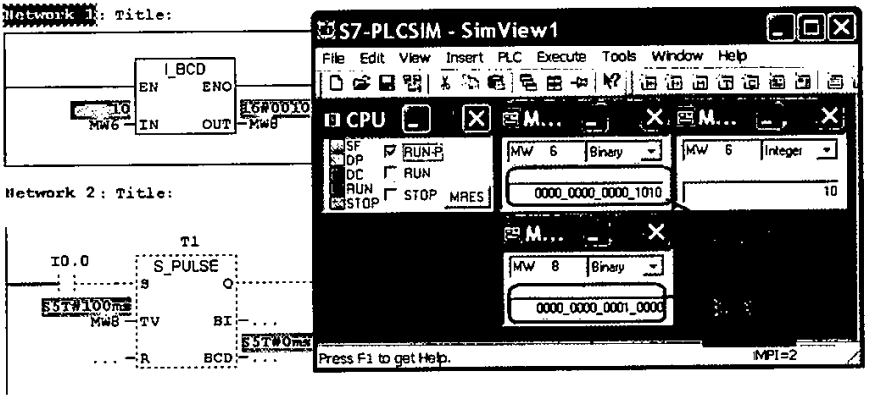
شکل ۲۰-۱۰ بافر CPU در رنامه مثال ۲۰-۳

اشکال در کجاست؟ با کمی دقت متوجه اشکال می‌شویم. به شکل ۲۰-۱۱ توجه کنید که عدد 10 را به‌صورت باینری نشان می‌دهد. همانطور که دیده می‌شود این عدد معادل BCD ندارد زیرا در BCD عددی بزرگتر از ۹ نداریم و چهار بیت سمت راست از ۹ بزرگتر است.

0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
				0				0				معادل BCD ندارد			

شکل ۲۰-۱۱ تحلیل بیت‌ها در شرایط بروز اشکال در برنامه مثال ۲۰-۳

چه باید کرد؟ پاسخ روشن است. با استفاده از مبدل I_BCD مشکل فوق برطرف می‌گردد. در برنامه زیر MW6 را به ورودی‌مبدل و خروجی مبدل را به MW8 اختصاص‌داده‌ایم تایمر نیز زمان‌را از MW8 گرفته است‌وبدون‌مشکل‌کار می‌کند.



شکل ۲۰-۱۲ استفاده از مبدل برای رفع مشکل مثال ۲۰-۳

در این‌حالت وقتی MW6 معادل 10 است اگر بیت‌های MW6 و MW8 را با هم مقایسه کنیم می‌بینیم که عدد صحیح 10 که 0000,1010 است توسط مبدل به 0001,0000 که معادل BCD 10 است تبدیل شده و تایمر آنرا می‌پذیرد.

فصل ۲۰

تمرین ۲۰-۱: در مثال فوق بررسی کنید اگر عدد ۱۶۴۲۶ را به MW6 اختصاص دهیم تایمر با چه زمانی تنظیم می‌شود؟ چرا؟

تمرین ۲۰-۲: شبیه مثال فوق کانتر را با ورودی متغیر برای پایه PV که عدد BCD می‌گیرد تست کنید. بدون مبدل چه حالتی پیش می‌آید؟

۲۰-۲-۲ تبدیل BCD به Double Integer و بالعکس

اعداد BCD می‌توانند ۳۲ بیتی باشند. در این‌حالت ماکزیمم مقدار آنها 9999999 خواهد بود. برای تبدیل این عدد BCD به عدد صحیح بایستی از عدد Double Integer که ۳۲ بیتی است استفاده کرد.

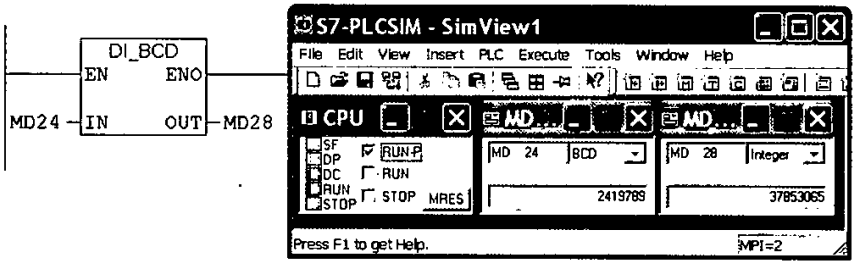
مثال ۲۰-۴: عدد ۲۴۱۹۷۸۹ به‌صورت **BCD** است. معادل صحیح آنرا به‌دست آورید.

ابتدا بدون استفاده از برنامه‌نویسی مطابق شکل ۲۰-۱۳ بیت‌های عدد BCD را مشخص می‌کنیم.

0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	1	0	1	1	1	1	0	0	0	1	0	0	1
				2				4				1				9				7				8				9		

شکل ۲۰-۱۳ بیت های عدد BCD برنامه مثال ۲۰-۴

اگر از روی بیت‌های صفر و یک به‌دست آمده معادل عدد صحیح را محاسبه کنیم عدد حاصل ۳۷۸۵۳۰۶۵ است که یک عدد صحیح ۳۲ بیتی است. اکنون با دستور تبدیل و سیمولاتور این تبدیل را چک می‌کنیم.



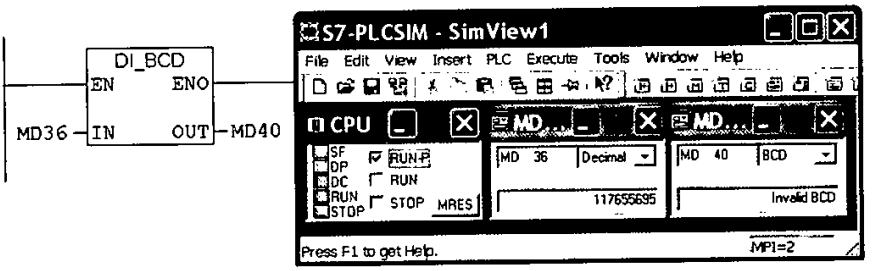
شکل ۲۰-۱۴ برنامه مثال ۲۰-۴

تبدیل DI_BCD

این تبدیل عکس BCD_DI عمل می‌کند یعنی عدد صحیح ۳۲ بیتی صحیح را در ورودی می‌گیرد و آنرا به BCD برمی‌گرداند.

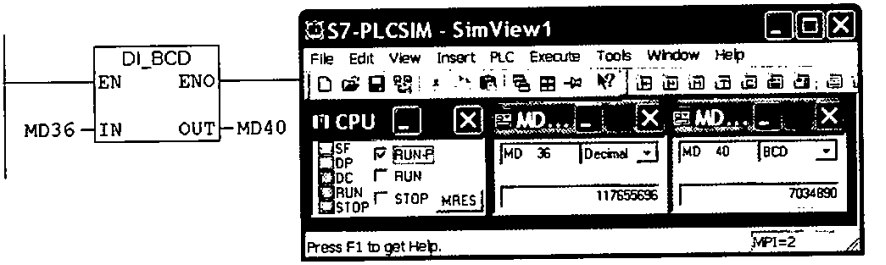
همانطور که قبلاً ذکر شد یک عدد BCD را همیشه می‌توان به عدد صحیح تبدیل کرد ولی عدد صحیح را همواره نمی‌توان به BCD تبدیل نمود. وقتی عدد صحیح را به‌صورت باینری می‌بینیم و از سمت راست به قسمت‌های ۴ بیتی تفکیک کنیم در صورتی که معادل یک ۴ بیت از عدد ۹ بیشتر شود معادل BCD نخواهد داشت. به‌عنوان مثال اگر معادل باینری ۱۰۱۰۰۰۰۱۱۱۱ باشد حاصل ۴ بیت سمت راست ۱۵ است. پس این عدد معادل BCD ندارد.

مثال ۲۰-۵: عدد صحیح ۱۱۷۶۵۵۶۹۵ که ۳۲ بیتی است معادل BCD ندارد. به بیت‌ها دقت کنید معادل چهار بیت سمت راست از ۹ بزرگتر است.



شکل ۲۰-۱۵ برنامه مثال ۲۰-۵ وقتی عدد BCD معتبر نیست

اکنون کافیست به عدد صحیح فوق یک شماره اضافه کنیم، آنگاه معادل BCD معتبر خواهد بود. کافیست خواننده اعداد صحیح فوق را به‌صورت بیتی بررسی کند تا این مثال را درک کند.



شکل ۲۰-۱۶ برنامه مثال ۲۰-۵ وقتی عدد BCD معتبر است

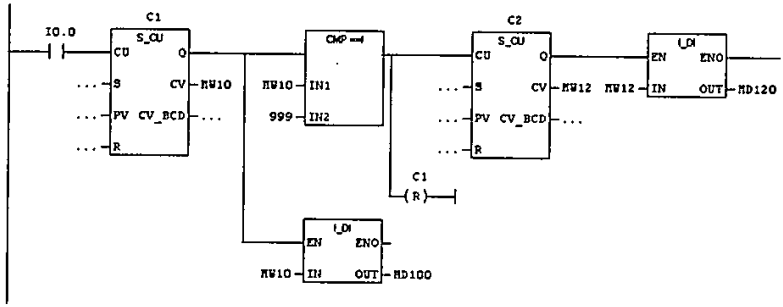
۲۰-۲-۳ تبدیل Integer به Double Integer

مبدل DI_I عدد صحیح ۱۶ بیتی را به‌صورت عدد صحیح ۳۲ بیتی تبدیل می‌کند. این تبدیل در برخی برنامه‌ها مورد نیاز است.

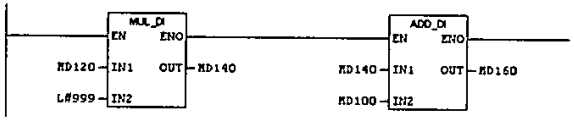
مثال ۲۰-۶: برای فهم بهتر موضوع به مثال ۱۹-۳ فصل قبل که مربوط به شمارش توسط دو کانتر بود بر می‌گردیم. در آنجا ذکر کردیم که برنامه مشکل دارد اگر نتیجه جمع از ۳۲۷۶۷ بیشتر شود برنامه جواب نادرست خواهد داشت. برنامه صحیح به‌صورت زیر است. همانطور که دیده می‌شود توسط مبدل DI_I نتیجه خروجی کانترها را ابتدا از Int به D.int تبدیل کرده سپس از دستورات محاسباتی DI استفاده کرده‌ایم.

فصل ۲۰

```
OB1 :  "Main Program Sweep (Cycle)"
Network 1 : Title:
```



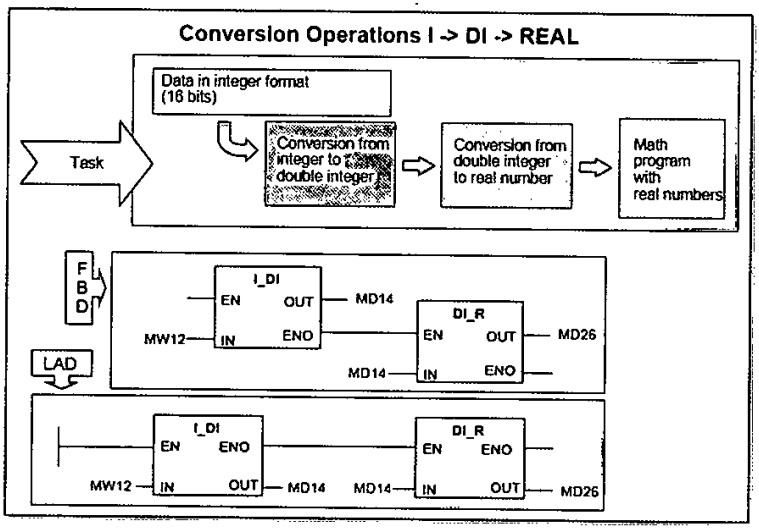

```
Network 2 : Title:
```



شکل ۲۰-۱۷ برنامه مثال ۲۰-۶

۲۰-۲-۴ تبدیل Integer به Real

جهت تبدیل یک داده‌ی Integer به یک داده‌ی Real دستور مستقیمی وجود ندارد. برای این کار باید ابتدا داده را از فرم INT به DINT تبدیل و سپس به Real تبدیل نمود. شکل ۲۰-۱۸ این موضوع را نشان می‌دهد.



شکل ۲۰-۱۸ چگونگی تبدیل یک داده‌ی Integer به یک داده‌ی Real

همانطور که در شکل مشخص است MW12 به‌صورت Integer خوانده شده و پس از تبدیل به‌صورت Double Integer در MD14 قرار می‌گیرد. در ادامه، MD14 به‌صورت Double Integer خوانده شده و پس از تبدیل به‌صورت Real در MD26 قرار می‌گیرد.

سوال: برای تبدیل عکس یعنی Real به Integer چه باید کرد؟ پاسخ این سوال در انتهای همین فصل روشن می‌شود.

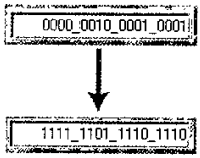
۲۰-۳ مبدل‌های متمم‌یک و دو (قرینه و معکوس‌سازی)

۲۰-۳-۱ دستور متمم یک (معکوس بیت‌های عدد)

دستور متمم ۱ در واقع معکوس‌کننده بیت‌های متغیر مورد نظر است. یعنی تمام یک‌ها را به صفر و تمام صفرها را به یک تبدیل می‌کند. به این تبدیل اصطلاحاً متمم یک می‌گویند.

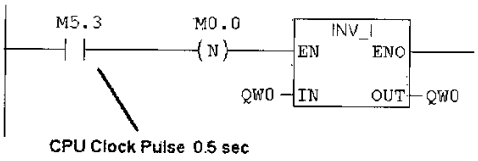
این تبدیل می‌تواند روی عدد صحیح ۱۶ بیتی یا عدد صحیح ۳۲ بیتی انجام شود. از اینرو به دو نوع زیر تقسیم می‌شود:

- **INV_I:** دستور متمم ۱ گرفتن از یک داده‌ی Integer
- **INV_DI:** دستور متمم ۱ گرفتن از یک عدد D.INT



شکل ۲۰-۱۹ متمم‌یک گرفتن از یک داده

مثال ۲۰ -۷: در برنامه زیر هر یک ثانیه یک‌بار تمام ۱۶ خروجی روشن و خاموش می‌شوند؛ چون از لبه منفی پالس 0.5 ثانیه‌ای CPU استفاده شده است و فاصله زمانی بین دو لبه یک ثانیه است.



شکل ۲۰-۲۰ برنامه مثال ۲۰-۷

این برنامه را با سیمولاتور تست کنید.

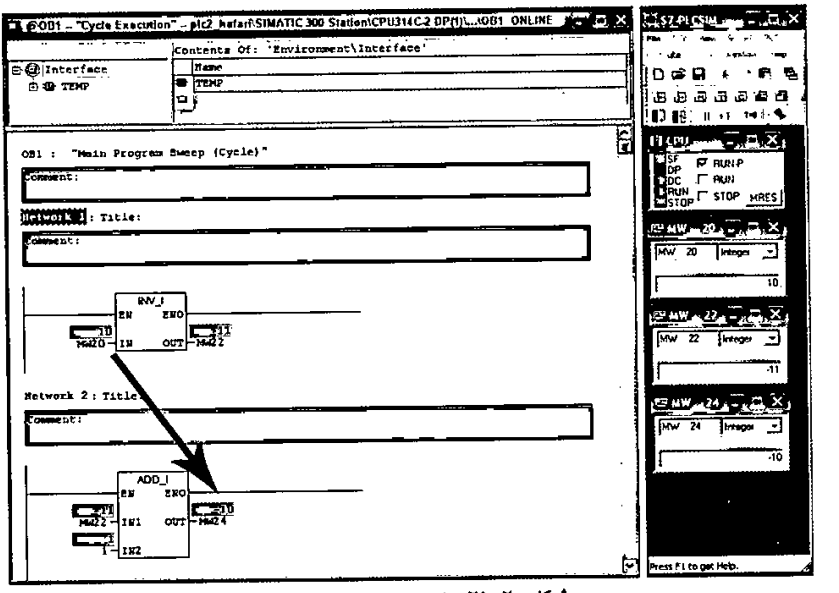
تذکر: اگر متمم یک، یک عدد با عدد 1 جمع شود قرینه‌ی آن عدد به‌دست می‌آید.

مثال ۲۰-۸: عددی به فرم Integer در MW20 قرار گرفته است. اگر این عدد برابر 14 باشد:

الف) متمم‌یک آنرا به‌دست آورید.

ب) قرینه‌ی آنرا به‌دست آورید.

حل: برای محاسبه متمم‌یک می‌توان از دستور INV_I استفاده نمود. از طرفی می‌دانیم که قرینه‌ی یک عدد برابر است با متمم‌یک آن عدد + عدد یک، بنابراین می‌توان با استفاده از دستور جمع‌کننده مقدار یک را با مقدار متمم‌یک جمع نمود تا قرینه‌ی عدد محاسبه شود.



شکل ۲۰-۲۱ برنامه مورد نظر جهت مثال ۲۰-۸

۲۰-۳-۲ دستور محاسبه‌ی متمم دو (قرینه‌ی عدد)

دستور متمم‌دو در واقع جمع متمم‌یک با عدد یک است که همانطور که در مثال قبل دیدیم، معادل قرینه عدد است ولی برای این منظور دستورات مستقیمی وجود دارند که به NEG موسوم هستند.

این دستورات عبارتند از :

- **NEG_I:** دستور متمم‌دو (قرینه) یک داده‌ی INT
- **NEG_DI:** دستور متمم‌دو (قرینه) یک داده‌ی D.INT
- **NEG_R:** دستور متمم‌دو (قرینه) یک داده‌ی Real

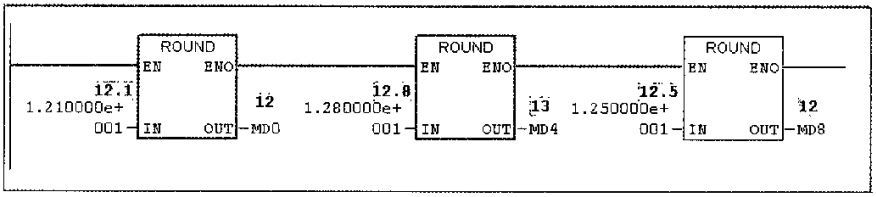
۲۰-۴ مبدل‌های Round کننده اعداد اعشاری

دستورات این خانواده عبارتند از:

- ROUND
- CEIL
- FLOOR
- TRUNC

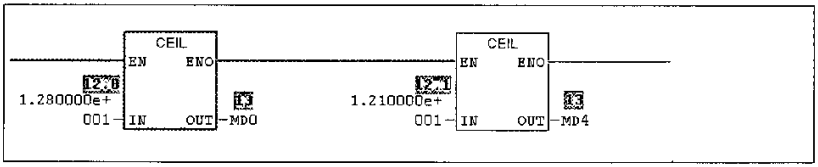
سه دستور اول برای رند کردن مقادیر اعشاری به‌کار می‌روند. نتیجه آنها عدد اعشاری است. به‌دستور آخر مقدار اعشاری را به عدد صحیح ۳۲ بیتی تبدیل می‌کند.

ROUND: توسط این دستور می‌توان یک عدد Real را به نزدیک‌ترین عدد صحیح به آن رند نمود. شکل ۲۰-۲۲ چگونگی عملکرد این تابع را نشان می‌دهد.



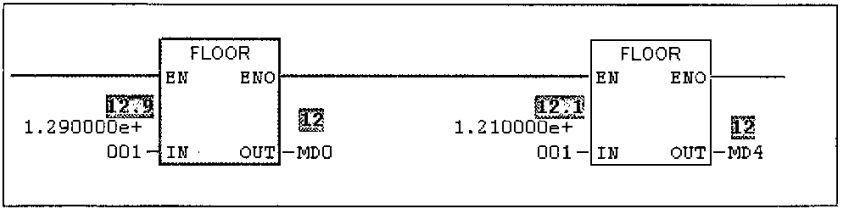
شکل ۲۰-۲۲ چگونگی عملکرد دستور ROUND

CEIL: توسط این دستور می‌توان یک عدد Real را به عدد صحیح بالاتر تبدیل نمود.



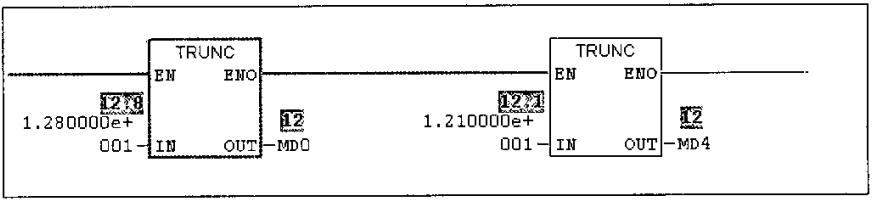
شکل ۲۰-۲۳ چگونگی عملکرد دستور CEIL

FLOOR: توسط این دستور می‌توان یک عدد Real را به عدد صحیح پایین‌تر تبدیل نمود.



شکل ۲۰-۲۴ چگونگی عملکرد دستور FLOOR

TRUNC: توسط این دستور می توان یک عدد Real را با حذف قسمت اعشاری به یک عدد صحیح تبدیل نمود. شکل ۲۰-۲۵ چگونگی عملکرد این تابع را نشان می‌دهد.



شکل ۲۰-۲۵ چگونگی عملکرد دستور TRUNC

تبدیل مقدار Real به Integer

تبدیل مستقیمی برای تبدیل یک متغیر Real به عدد صحیح ۱۶ بیتی وجود ندارد ولی می‌توان آنرا از خروجی Trunc گرفت. کافیست ۱۶ بیت با ارزش کمتر را جدا کنیم. به‌عنوان مثال در شکل ۲۰-۲۵ که مقدار به MD4 منتقل شده می‌توان از MW2 به‌عنوان مقدار صحیح ۱۶ بیتی استفاده کرد. زیرا:

MD4	
MW4	MW6

شکل ۲۰-۲۶ تبدیل Real به Integer در مثال شکل ۲۰-۲۵

۲۰-۵ تمرین

چگونه می توان عدد BCD شانزده بیتی را روی 7-Segment نمایش داد.

فصل ۲۱

دستور انتقال MOVE

۲۱-۱ مقدمه

۲۱-۲ عملکرد دستور MOVE

۲۱-۳ نکات کار با دستور MOVE

۲۱-۴ تمرین

در این فصل با نحوه برنامه‌نویسی و نکات مربوط به دستور پرکاربرد MOVE در زبان‌های LAD و FBD آشنا می‌شوید.

چکیده مطالب

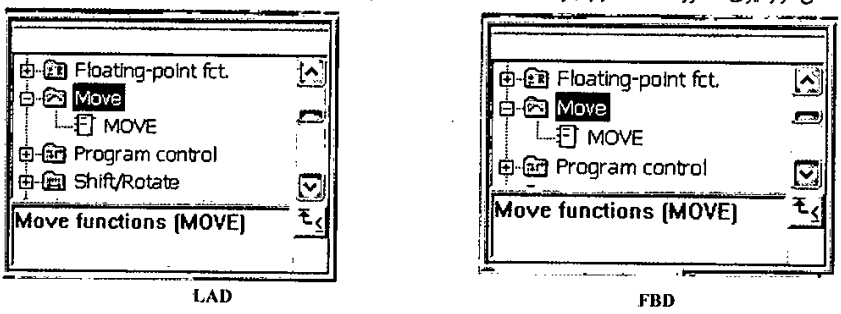
- دستور Move برای کپی کردن یک مقدار یا یک متغیر به متغیر دیگر به‌کار می‌رود.
- یکسان بودن تعداد بیت متغیر ورودی و خروجی در دستور Move الزامی نیست. بدیهی است وقتی تعداد بیت ورودی بیشتر باشد فقط بخشی از دیتا به خروجی منتقل می‌شود.
- ورودی دستور Move ماکزیمم ۳۲ بیتی است. برای انتقال مقادیر بزرگتر از فانکشن Block_Move که در کتاب سطح تکمیلی توضیح داده شده استفاده می‌گردد.

۲۱-۱ مقدمه

دستور انتقال که با المان MOVE به‌کار می‌رود یکی از دستورات پرکاربرد در برنامه‌نویسی‌های LAD و FBD است. توسط این دستور می‌توان:

- مقادیر دلخواه را به متغیرها نسبت داد.
- محتوای یک متغیر را به متغیر دیگر کپی نمود.

محل قرارگیری دستور Move در پنجره Program Element و در پوشه‌ی Move می‌باشد.

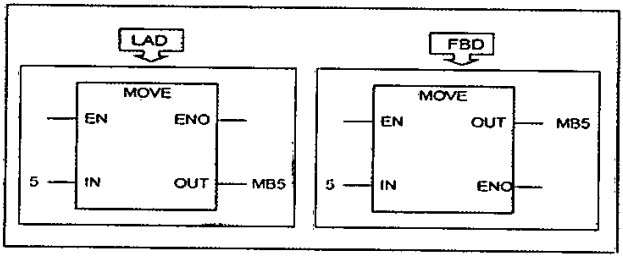


شکل ۲۱-۱ دستورات تبدیل در Program Element

۲۱-۲ عملکرد دستور MOVE

همانطور که در شکل ۲۱-۱ مشاهده می‌شود، دستور Move به‌صورت یک المان گرافیکی است که دارای پایه‌های زیر است:

- **EN:** فعال‌ساز تابع (متغیر از نوع Bool)
- **ENO:** نشان‌دهنده صحت کار تابع (متغیر از نوع Bool)
- **IN:** ورودی تابع
- **OUT:** خروجی تابع

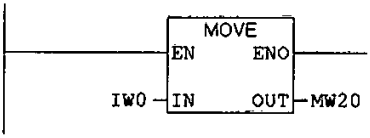


شکل ۲۱-۲ دستور Move در زبان LAD و FBD

عملکرد

هنگامی که پایه‌ی EN فعال شود، از داده‌ای که در پایه‌ی IN قرار گرفته شده است کپی برداری شده و در آدرسی که در پایه‌ی OUT وارد شده است قرار داده می‌شود. اگر پایه‌ی ENO مقدار صفرِمنطقی را برگرداند، یعنی در کار این تابع اشکالی به‌وجود آمده است. اگر پایه‌ی ENO دارای مقدار یکِ منطقی باشد، یعنی این تابع کار خود را به‌درستی انجام داده و خطایی به‌وجود نیامده است. اگر در پایه‌ی EN آدرسی وارد نشود تابع همواره فعال بوده و عمل انتقال انجام می‌شود. به‌عنوان مثال در برنامه‌ی نشان داده شده در شکل ۲۱-۲ ، عدد ۵ که در پایه‌ی EN وارد شده است به آدرس MB5 منتقل می‌شود. البته همانطور که ذکر شد، این عمل با کپی برداری صورت می‌پذیرد.

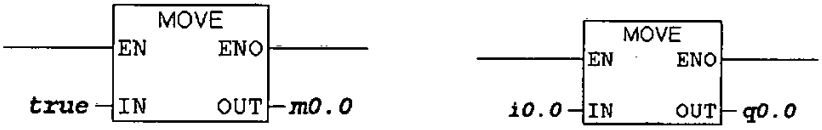
در شکل زیر مقدار یک متغیر به متغیر دیگری کپی می‌شود، بدیهی است محتوای متغیر اول یعنی (IW0) پاک نمی‌شود.



شکل ۲۱-۳ دستور Move برای کپی کردن یک متغیر به متغیر دیگر

۲۱-۳ نکات کار با دستور MOVE

- دستور Move را برای مقدار یا متغیر Bool نمی‌توان استفاده کرد. شکل ۲۱-۴ کاربرد نادرست این دستور را برای مقادیر بیتی نشان می‌دهد.



شکل ۲۱-۴ غیر مجاز بودن استفاده از مقدار یا آدرس بیتی در دستور MOVE

- دستور MOVE قادر است متغیرها و اعدادی ۸ بیتی، ۱۶ بیتی و ماکزیمم ۳۲ بیتی را کپی کند بنابراین متغیرهای ورودی و خروجی این دستور می‌تواند مطابق جدول ۲۱-۱ باشد.

جدول ۲۱-۱

نوع متغیر	کاربرد در ورودی	کاربرد در خروجی
IB , IW , ID	بله	بله
PIB , PIW , PID	بله	خیر
QB , QW , QD	بله	بله

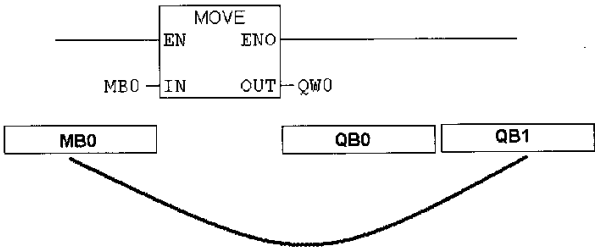
PQB , PQW , PQD	خیر	بله
MB , MW , MD	بله	بله
DBB , DBW , DBD	بله	بله
C	بله	خیر
T	بله	خیر

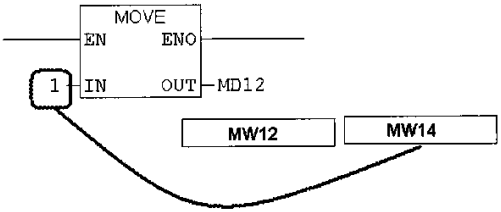
- دستور Move تمام فرمت‌های اعداد و یا سایر پارامترها که از ۳۲ بیت بیشتر نباشند را در ورودی می‌پذیرد (به‌جز Bool). جدول ۲۱-۲ این موضوع را نشان می‌دهد.

جدول ۲۱-۲

فرمت عدد	مثال
مبنای ۲	2#11101100
عدد صحیح ۱۶ بیتی	23456
عدد صحیح ۳۲ بیتی	L#23456789
عدد اعشاری	987.23
عدد BCD	C#945
عدد هگز ۸ بیتی	B#16#3A
عدد هگز ۱۶ بیتی	W#16#3AEF
عدد هگز ۳۲ بیتی	DW#16#2B3EFAE0
زمان S5Time	S5T#20S
زمان Time	T#20S
زمان Time of Day	TOD#7:20:30.5
تاریخ	D#2011-1-1
کاراکتر	'a'
استرینگ (ماکزیمم ۴ کاراکتری)	'TeSt'

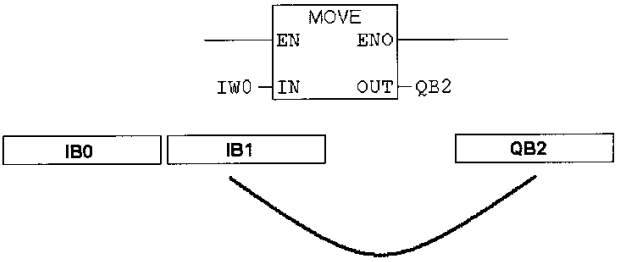
- در دستور Move ورودی می‌تواند از نظر تعداد بیت کمتر از خروجی باشد. بدیهی است در چنین حالتی ورودی به بیت‌های با ارزش کمتر منتقل می‌شوند. به‌عنوان مثال به شکل ۲۱-۵ توجه کنید.





شکل ۲۱-۵ عملکرد دستور Move وقتی سایز ورودی کوچکتر از سایز خروجی است.

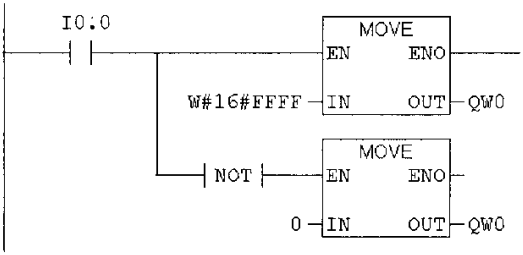
- در دستور Move ورودی می‌تواند از نظر تعداد بیت بیشتر از خروجی باشد. بدیهی است در چنین حالتی فقط بیت‌های با ارزش کمتر کپی می‌شوند. به شکل ۲۱-۶ توجه کنید.



شکل ۲۱-۶ عملکرد دستور Move وقتی سایز ورودی بزرگتر از سایز خروجی است.

مثال ۲۱-۱: روشن و خاموش کردن ۱۶ خروجی با یک سوئیچ

در برنامه زیر وقتی سوئیچ I0.0 یک است، با استفاده از دستور Move مقدار هگز FFFF که معادل ۱۶ یک است به تمام ۱۶ کانال خروجی منتقل می‌شود و وقتی سوئیچ فوق صفر است عدد صفر که معادل ۱۶ صفر باینری است به خروجی‌ها منتقل می‌گردد.

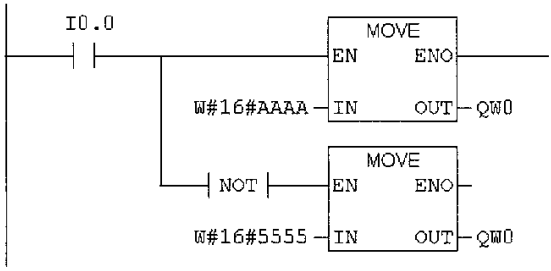


شکل ۲۱-۷ برنامه مثال ۲۱-۱

مثال ۲۱-۲: روشن و خاموش‌کردن خروجی به‌صورت یک در میان

برنامه مثال ۲۰-۱ را به‌گونه‌ای تغییر دهید که با فعال شدن I0.0 بیت‌های فرد خروجی روشن شوند و با غیرفعال شدن I0.0 بیت‌های زوج خروجی روشن شوند.

برنامه به‌صورت نشان داده شده در شکل ۲۱-۸ است. برای تحلیل آن توجه کنید که عدد هگز A معادل 1010 و عدد هگز 5 معادل 0101 می‌باشد.



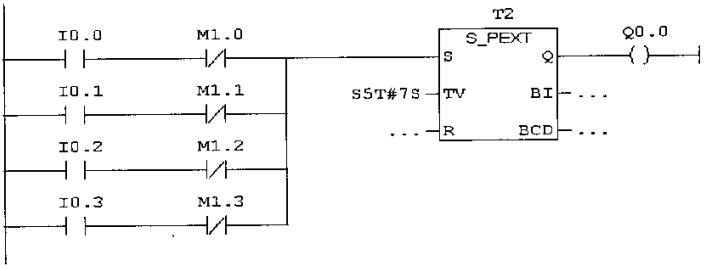
شکل ۲۱-۸ برنامه مثال ۲۱-۲

مثال ۲۱-۳: روشنایی راه پله

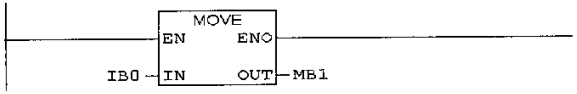
لامپ‌های روشنایی راه‌پله یک ساختمان چهار طبقه توسط شستی‌هایی که در هر طبقه وجود دارند و به ورودی‌های I0.0 و I0.1 و I0.2 و I0.3 متصل هستند روشن می‌شوند و به‌طور خودکار پس از یک دقیقه خاموش می‌گردند. اگر قبل از سپری شدن یک دقیقه شستی دیگری یا همان شستی فشار داده شد زمان‌سنجی از ابتدا شروع شود یعنی یک دقیقه از اول محاسبه گردد. در ضمن اگر شستی بیش از یک دقیق فعال بماند لامپ پس از همان یک دقیقه خاموش شود.

برنامه به‌صورت زیر است. تحلیل به عهده خواننده واگذار می‌شود.

```
OB1 :  "Main Program Sweep (Cycle)"
Network 1 : Title:
```



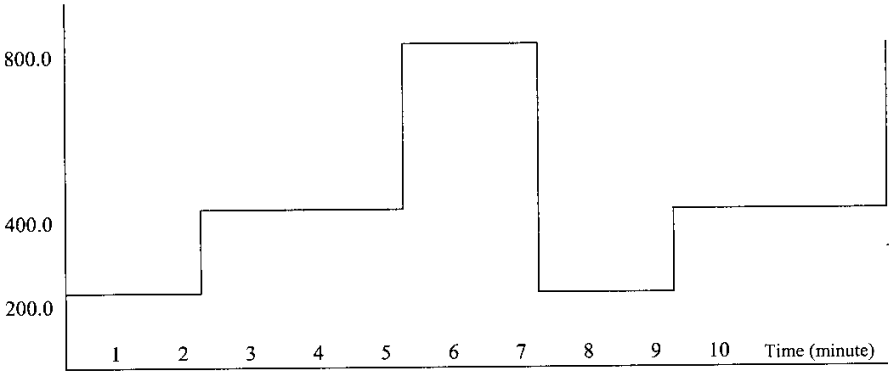

```
Network 2 : Title:
```



شکل ۲۱-۹ برنامه مثال ۲۱-۳

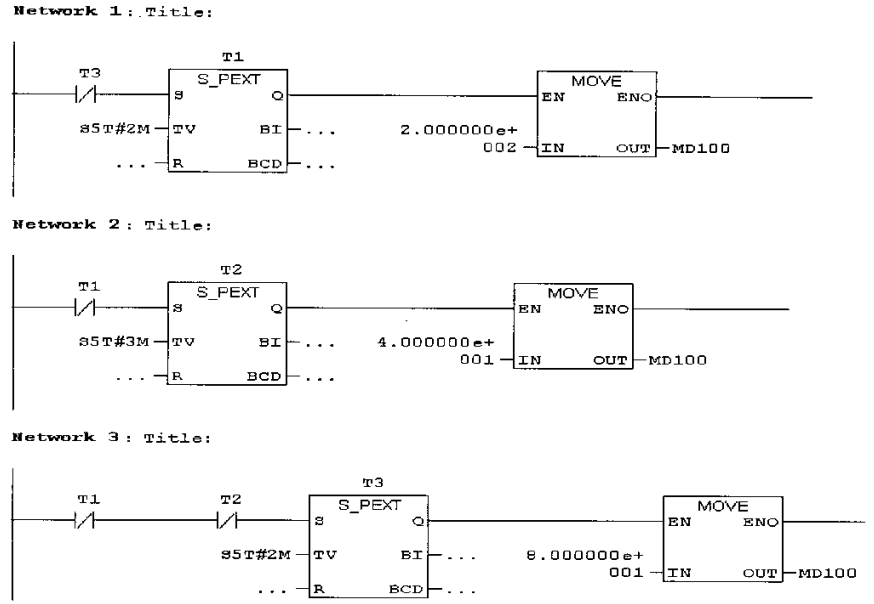
مثال ۲۱-۴: ایجاد Setpoint متغیر

در فرآیندی برای سیکل عملیات حرارتی که ویژگی‌های خاصی را به محصول می‌دهد لازم است مقدار مبنای کنترلر دما طبق شکل ۲۱-۱۰ متغیر باشد و به‌صورت سیکلی تکرار شود.



شکل ۲۱-۱۰ تغییرات مقدار مبنا در مثال ۲۱-۴

برنامه زیر طبق روال خواسته شده مقدار مبنا را به متغیر MD100 منتقل می‌کند.



شکل ۲۱-۱۱ برنامه مثال ۲۱-۴

مثال ۲۱-۵ : تایمر با زمان متغیر

با استفاده از دستور Move و یک تایمر از نوع S_PEXT، برنامه‌ای بنویسد که با فشرده‌شدن هر یک از شستی‌های زیر، خروجی Q0.0 به مدت ذکر شده روشن و سپس خاموش شود.

الف) شستی I0.0، خروجی به‌مدت ۵ ثانیه روشن باشد.

ب) شستی I0.1، خروجی به‌مدت ۷ ثانیه روشن باشد.

ج) شستی I0.2، خروجی به مدت ۱۰ ثانیه روشن باشد.

حل: لازم است ورودی TV تایمر S_PULSE به‌صورت متغیر باشد و توسط سه دستور Move، زمان‌های مورد نظر را به این متغیر داده شود.

فصل ۲۲

دستورات شیفت و چرخش

۲۲-۱ مقدمه
۲۲-۲ دستورات شیفت
۲۲- ۳ دستورات چرخش
۲۲-۴ تمرین

در این فصل با نحوه برنامه‌نویسی و نکات مربوط به دستور کم‌کاربرد شیفت و چرخش در زبان‌های LAD و FBD آشنا می‌شوید.

فصل
۲۲

چکیده مطالب

عنوان دستور	عملکرد
SHR_I	جابه‌جایی بیت‌های عدد صحیح ۱۶ بیتی به راست
SHR_DI	جابه‌جایی بیت‌های عدد صحیح ۳۲ بیتی به راست
SHL_W	جابه‌جایی بیت‌های یک Wordبه چپ
SHR_W	جابه‌جایی بیت‌های یک Wordبه راست
SHL_DW	جابه‌جایی بیت‌های یک DWord به چپ
SHR_DW	جابه‌جایی بیت‌های یک DWord به راست
ROL_DW	چرخش بیت‌های یک DWord به چپ
ROR_DW	چرخش بیت‌های یک DWord به راست

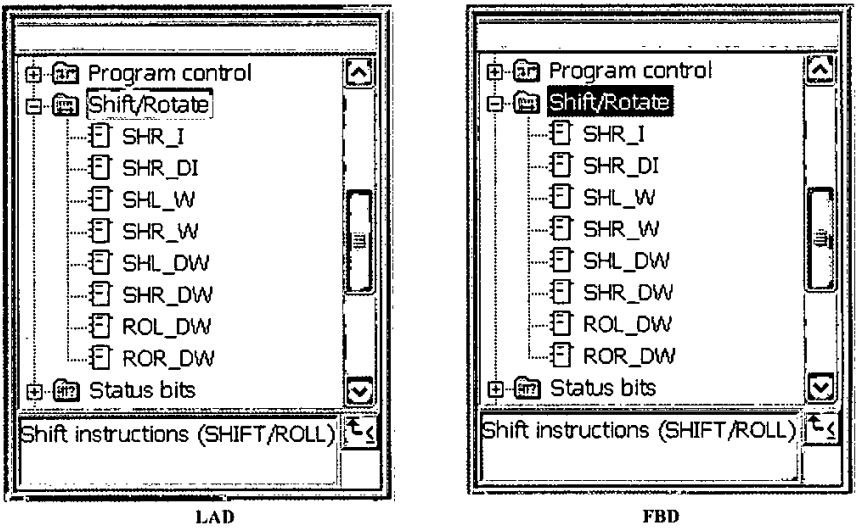
۲۲-۱ مقدمه

دستورات شیفت و چرخش پرکاربرد نیستند ولی در برخی موارد برای پیاده‌سازی منطق کنترل به آنها نیاز داریم.

الف) دستورات شیفت (Shift): توسط این دستورات می‌توان در بیت‌های یک داده از نوع INT، D.INT، Word و D.Word جابه‌جایی انجام داد.

ب) دستورات چرخش (Rotate): توسط این دستورات امکان چرخش در بیت‌های یک داده از نوع Word و Double Word وجود دارد.

محل قرارگیری دستورات شیفت و چرخش در پنجره Program Element در شکل ۲۲-۱ آمده است.



شکل ۲۲-۱ دستورات شیفت و چرخش در Program Element

جدول ۲۲-۱ این دستورات را نشان می‌دهد.

جدول ۲۲-۱

SHR_I	Shif Right Integer	جابه‌جایی بیت‌های عدد صحیح ۱۶ بیتی به راست
SHR_DI	Shif Right Double Integer	جابه‌جایی بیت‌های عدد صحیح ۳۲ بیتی به راست
SHL_W	Shif Left Word	جابه‌جایی بیت‌های یک Wordبه چپ
SHR_W	Shif Right Word	جابه‌جایی بیت‌های یک Wordبه راست
SHL_DW	Shif Left Double Word	جابه‌جایی بیت‌های یک DWord به چپ
SHR_DW	Shif Right Double Word	جابه‌جایی بیت‌های یک DWord به راست
ROL_DW	Rotate Left Double Word	چرخش بیت‌های یک DWord به چپ
ROR_DW	Rotate Right Double Word	چرخش بیت‌های یک DWord به راست

برای شناخت عملکرد این دستورات لازم است توجه شود که CPU مقدار متغیر را به یک انباره یا به‌اصطلاح آکومولاتور ۳۲ بیتی انتقال می‌دهد. دستورات شیفت و چرخش روی بیت‌های این انباره تاثیر می‌گذارد. تشریح کامل انباره-های CPU در کتاب سطح تکمیلی آورده شده است.

ورودی و خروجی‌های دستورات شیفت و چرخش

همه دستورات شیفت و چرخش دارای ورودی و خروجی یکسان هستند. پایه‌های آنها عبارتند از:

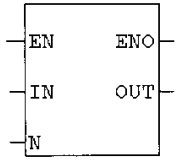
EN: فعال‌ساز تابع از نوع Bool

ENO: نشان‌دهنده صحت کار تابع از نوع Bool

IN: ورودی تابع (داده‌ای که قرار است عمل جابه‌جایی یا چرخش روی آن انجام شود)

N: تعداد جابه‌جایی یا تعداد چرخش (به فرم Word)

OUT: خروجی تابع بعد از انجام عمل جابه‌جایی



شکل ۲۲-۲ ورودی و خروجی‌های شیفت و چرخش

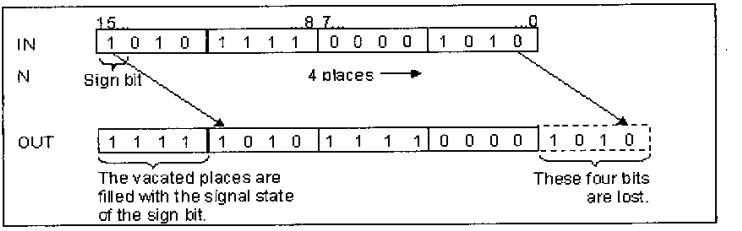
۲۲-۲ دستورات شیفت

(Shift Right Integer) SHR_I

توسط این دستور عمل جابه‌جایی به راست برای بیت‌های یک داده از نوع Integer انجام می‌پذیرد. عمل جابه‌جایی به اینصورت انجام می‌شود که:

- کلیه‌ی بیت‌های موجود در متغیر، با توجه به تعداد جابه‌جایی، به سمت راست منتقل می‌شوند. به‌عنوان مثال با یک جابه‌جایی تمام ۱۶ بیت به اندازه یک بیت به سمت راست منتقل می‌شوند.
- به تعداد عمل جابه‌جایی، بیت‌های موجود در سمت راست مقدار خود را از دست داده و مقادیر شیفت داده شده در آنها قرار می‌گیرند. پس با یک بیت جابه‌جایی اولین بیت سمت راست از بین می‌رود و بیت دوم به‌جای آن می‌نشیند و همینطور سایر بیت‌ها جابه‌جا می‌گردند.
- در هر بار شیفت راست بسته به مقدار بیت آخر (یعنی بیت پانزدهم که علامت عدد را نشان می‌دهد) 0 یا 1 از سمت چپ وارد آخرین بیت می‌شود. به‌عبارت دیگر اگر عدد منفی باشد 1 و اگر عدد مثبت باشد 0 وارد می‌شود. بنابراین علامت عدد عوض نمی‌شود.

برای درک بهتر عملکرد این تابع به مثال نشان‌داده شده در شکل ۲۲-۳ توجه نمایید.



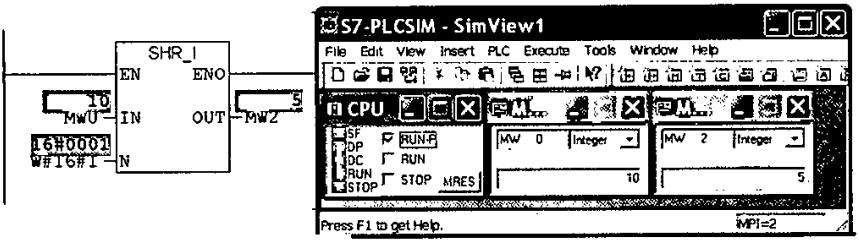
شکل ۲۲-۳ انجام جابه‌جایی به راست با تعداد چهار جابه‌جایی برای یک داده از نوع Integer

همانطور که در شکل ۲۲-۳ مشخص است، چهار عمل جابجایی روی یک متغیر Integer اعمال شده است، در نتیجه چهار بیت از سمت راست متغیر خارج شده و به‌جای آنها چهار بیت از سمت چپ وارد شده است. چون بیتِ علامت، که قبل از اجرای دستور دارای مقدار یک‌منطقی بوده، چهار بیت وارد شده همگی یک هستند و علامت عدد همچنان منفی است.

تذکر: شیفت به راست معادل عمل تقسیم است. شیفت به اندازه n بیت به راست معادل تقسیم بر عدد 2^n است، به‌عنوان مثال با یک شیفت راست عدد نصف می‌شود. با دو شیفت راست عدد بر چهار تقسیم می‌گردد و با سه شیفت راست عدد بر ۸ تقسیم می‌گردد.

مثال ۲۲-۱: در مثال شکل زیر مقدار MW0 به اندازه یک شیفت به راست منتقل شده و در MW2 قرار می‌گیرد. مقدار MW0 توسط سیمولاتور عدد ۱۰ داده شده است که نتیجه یک شیفت راست عدد را نصف می‌کند و مقدار ۵ را در خروجی می‌بینیم.

فصل ۲۲



بیت	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
قبل از شیفت	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
بعد از شیفت	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

شکل ۲۲-۴ برنامه مثال ۲۲-۱

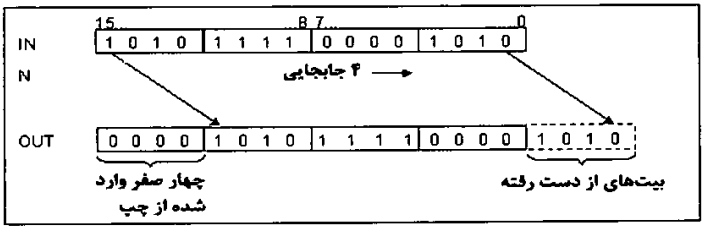
(Shift Right Double Integer) SHR_DI

توسط این دستور عمل جابه‌جایی به راست برای بیت‌های یک داده از نوع Double Integer صورت می‌پذیرد. عملکرد آن مانند دستور SHR_I می‌باشد، با این تفاوت که روی اعداد صحیح ۳۲ بیتی عمل می‌کند.

(Shift Right Word) SHR_W

توسط این دستور عمل جابه‌جایی به راست برای بیت‌های یک داده از نوع Word صورت می‌پذیرد. عملکرد آن مانند دستور SHR_I است با این تفاوت که وقتی بیت‌ها به سمت راست جابه‌جا می‌شوند، از سمت چپ همواره مقدار صفر وارد متغیر می‌شود در حالی که در SHR_I بسته به علامت عدد صفر یا یک وارد می‌شد.

برای درک بهتر موضوع، به شکل ۲۲-۵ توجه نمایید.

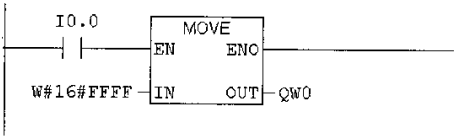


شکل ۲۲-۵ انجام جابه‌جایی به راست با تعداد چهار جابه‌جایی برای یک داده از نوع Word

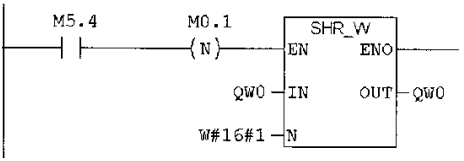
همانطور که در شکل فوق مشخص است، چهار جابجایی به راست برای یک داده از نوع Word صورت پذیرفته است. همه‌ی بیت‌ها چهار بار به سمت راست جابجا شده‌اند، بنابراین چهار بیت از سمت راست از بین رفته و چهار صفر از سمت چپ وارد متغیر می‌شود.

مثال ۲۲-۲: در برنامه زیر با فعال شدن شستی I0.0 تمام ۱۶ خروجی دیجیتال مربوط به QW0 یک می‌شوند سپس با هر لبه منفی پالس که از بیت چهارم CPU Clock گرفته شده خروجی‌ها به‌ترتیب خاموش می‌شوند.

```
OB1 : Title:
Network 1 : Title:
```



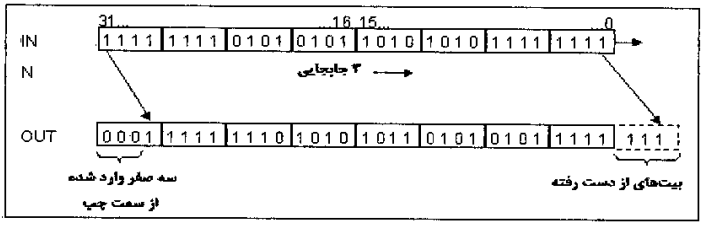

```
Network 2 : Title:
```



شکل ۲۲-۶ برنامه مثال ۲۲-۲

(Shift Right Double Word) SHR_DW

توسط این دستور عمل جابه‌جایی به راست برای بیت‌های یک داده از نوع Double Word که ۳۲ بیتی است صورت پذیرفته و عملکرد آن مانند دستور SHR_W است. برای درک بهتر موضوع به شکل ۲۲-۷ توجه نمایید.



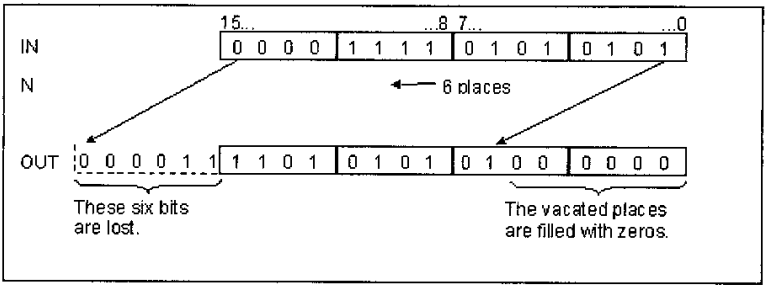
شکل ۲۲-۷ انجام جابه‌جایی به راست با تعداد چهار جابه‌جایی برای یک داده از نوع Double Word

(Shift Left Word) SHL_W

توسط این دستور عمل جابه‌جایی به چپ برای بیت‌های یک داده از نوع Word صورت می‌پذیرد. عملکرد آن مانند دستور SHR_W است، با این تفاوت که جابه‌جایی به چپ انجام می‌شود نه به راست. در این دستور به‌ازای بیت‌هایی که به چپ منتقل شده است، از سمت راست مقدار صفر وارد می‌شود.

شکل ۲۲-۸ چگونگی عملکرد این دستور را برای شش جابه‌جایی نشان می‌دهد.

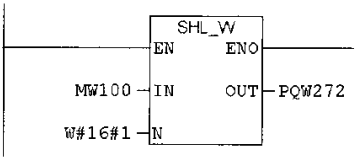
فصل ۲۲



شکل ۲۲-۸ انجام جابه‌جایی به چپ با تعداد شش جابه‌جایی برای یک داده از نوع Word

تذکر: شیفت به چپ معادل عمل ضرب است. شیفت به اندازه n بیت به چپ معادل ضرب محتوای آکومولاتور در عدد 2^n است، به‌عنوان مثال با یک شیفت چپ عدد ۲ برابر می‌شود و با سه شیفت چپ عدد ۸ برابر می‌شود.

مثال ۲۲-۳: با استفاده از دستور شیفت چپ برنامه‌ای بنویسید که مقدار موجود در MW100 دو برابر شده و به آنالوگ خروجی PQW272 منقل شود.



شکل ۲۲-۹ برنامه مثال ۲۲-۳

تمرین ۲۲-۱: مثال ۲۲-۲ را با استفاده از شیفت چپ آزمایش کنید. ترتیب خاموش‌شدن خروجی‌ها برعکس خواهد بود.

(Shift Left Double Word) SHL_DW

توسط این دستور عمل جابه‌جایی به چپ برای بیت‌های یک داده از نوع Double Word صورت پذیرفته و عملکرد آن مانند دستور SHL_W است.

۲۲- ۳ دستورات چرخش

(Rotate Left Double Word) ROL_DW

توسط این دستور عمل چرخش به چپ بیت‌های یک داده از نوع Double Word انجام می‌پذیرد. در عمل چرخش بیتی حذف نشده و بیتی از خارج متغیر وارد آن نمی‌شود، بلکه بین بیت‌های درون متغیر چرخش صورت می‌پذیرد. شکل ۲۲-۱۰ چگونگی عملکرد این دستور را برای سه چرخش به چپ نشان می‌دهد.

فصل ۲۳

سایر دستورات برنامه‌نویسی LAD/FBD

۲۳-۱ مقدمه

۲۳-۲ دستورات Word Logic

۲۳-۳ دستورات Jump

۲۳-۴ دستورات Program Control

۲۳-۵ دستور باز کردن DB

۲۳-۶ دستورات Status Bits

۲۳-۷ تمرین

در این فصل با برخی دیگر از دستورات برنامه‌نویسی در زبان‌های LAD و FBD آشنا می‌شوید. تعدادی از دستورات LAD/FBD در اینجا قابل تشریح نیستند و در کتاب سطح پیشرفته بحث می‌شوند.

چکیده مطالب

- دستورات Word Logic می‌توانند مقادیر و متغیرها را به‌صورت بیت به بیت با یکدیگر AND یا OR یا XOR کنند.
- دستورات Jump می‌توانند براساس وقوع شرط به نقطه دیگری از برنامه پرش انجام دهند.
- دستور MCR شبیه قطع اضطراری عمل می‌کند و می‌تواند یک ناحیه از برنامه را تحت تاثیر قرار دهد.
- دستورات CALL و RET و DB_CALL و Status_Bit در کتاب‌های سطح پیشرفته و تکمیلی تشریح می‌شوند.

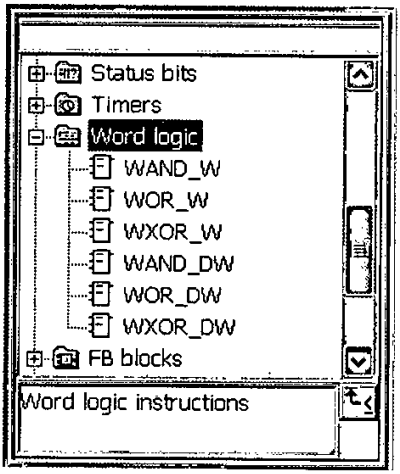
فصل
۲۳

۲۳-۱ مقدمه

در این فصل به سایر دستورات باقیمانده در پنجره Program Element اشاره می‌شود. برخی از این دستورات را در این کتاب نمی‌توان توضیح داد، تشریح آنها در کتاب‌های سطح پیشرفته و تکمیلی آمده است.

۲۳-۲ دستورات Word Logic

توسط این دستورات می‌توان اعمال منطقی را روی داده‌های از نوع Word و Double Word به‌کار برد. منظور از اعمال منطقی، اعمال AND، OR و XOR می‌باشد. در واقع توسط این دستورات، تک‌تک بیت‌های متناظر (هم‌شماره) با یکدیگر به‌صورت منطقی ترکیب می‌شوند.



شکل ۲۳-۱ محل قرارگیری دستورات Word logic

پایه‌های ورودی و خروجی این دستورات

همه‌ی دستورات Word Logic دارای پایه‌های مشترکی به شرح زیر می‌باشند:

EN: فعال‌ساز (از نوع Bool)

ENO: نشان‌دهنده صحت کار تابع (از نوع Bool)

IN1: ورودی اول

IN2: ورودی دوم

OUT: خروجی تابع

عملکرد: با فعال‌شدن پایه‌ی EN، مقدار ورودی‌های IN1 و IN2 طبق دستور تابع با یکدیگر ترکیب منطقی شده و نتیجه در پایه‌ی OUT قرار می‌گیرد.

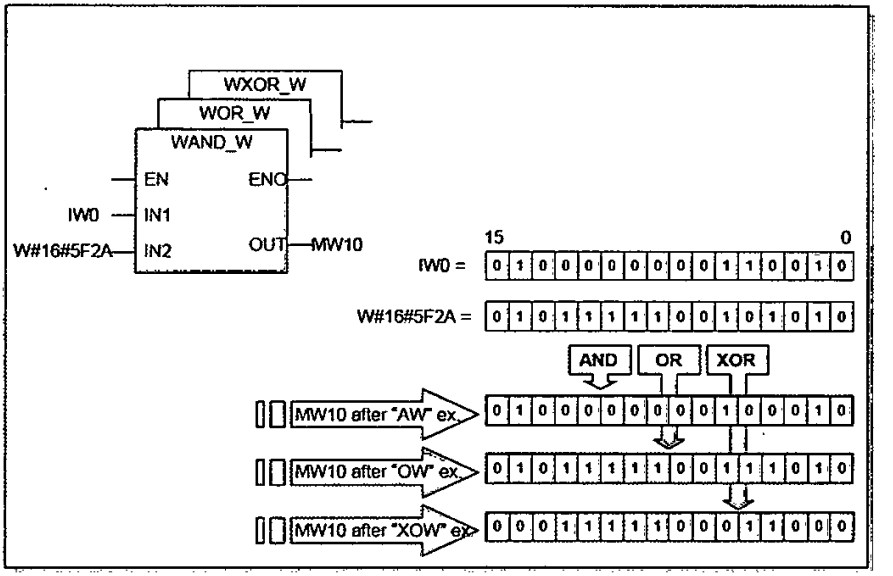
انواع دستورات Word Logic

WAND_W (AND Word): توسط این دستور تک‌تک بیت‌های دو متغیر از نوع Word با یکدیگر ترکیب AND منطقی می‌شوند.

WOR_W (OR Word): توسط این دستور تک‌تک بیت‌های دو متغیر از نوع Word با یکدیگر ترکیب OR منطقی می‌شوند.

WXOR_W (Exclusive OR Word): توسط این دستور تک‌تک بیت‌های دو متغیر از نوع Word با یکدیگر ترکیب XOR می‌شوند.

شکل۲۳-۲ چگونگی عملکرد این دستورات را نشان می‌دهد.



شکل ۲۳-۲ چگونگی عملکرد دستورات WAND_W، WOR_W و WXOR_W

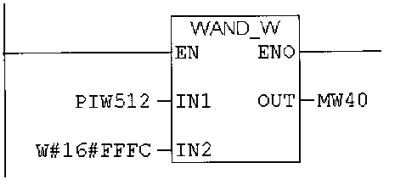
WAND_DW (AND Double Word): توسط این دستور تک‌تک بیت‌های دو متغیر از نوع Double Word با یکدیگر ترکیب AND منطقی می‌شوند.

WOR_DW (OR Double Word): توسط این دستور تک‌تک بیت‌های دو متغیر از نوع Double Word با یکدیگر ترکیب OR منطقی می‌شوند.

WXOR_W (Exclusive OR Word): توسط این دستور تک‌تک بیت‌های دو متغیر از نوع Double Word با یکدیگر ترکیب XOR می‌شوند.

مثال ۲۳-۱: ماسک کردن دو بیت اول آنالوگ ورودی

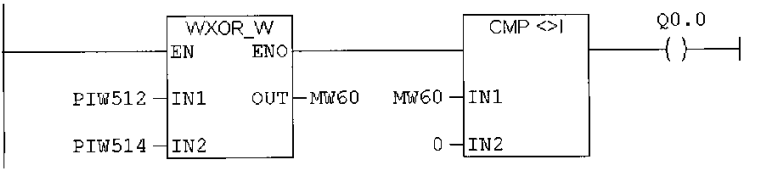
در فرآیندی سیگنال ترانسمیتر فشار به کارت آنالوگ با دقت 16 bit داده شده است. برنامه زیر با استفاده از دستور AW تغییرات دو بیت سمت راست را حذف و نتیجه را به خانه‌ای از حافظه منتقل می‌کند. در ورودی IN2 به عدد هگز داده شده یعنی FFFC دقت کنید. عدد F معادل 1111 است و تغییری در سیگنال ایجاد نمی‌کند ولی عدد C معادل 1100 است و دو بیت سمت راست را صفر می‌کند (Mask).



شکل ۲۳-۳ برنامه مثال ۲۳-۱

مثال ۲۳-۲: مقایسه مساوی بودن دو سیگنال آنالوگ

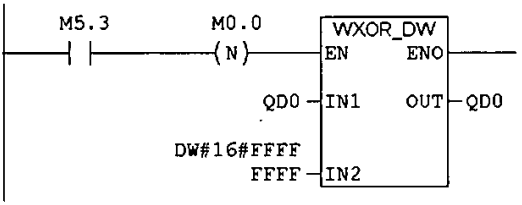
برنامه زیر با دستورات XOR دو سیگنال آنالوگ را بیت به بیت ترکیب می‌کند. اگر دقیقاً مساوی باشند نتیجه XOR صفر است در صورتی که مساوی نباشند Q0.0 روشن می‌شود.



شکل ۲۳-۴ برنامه مثال ۲۳-۲

مثال ۲۳-۳: خاموش و روشن شدن همه ۳۲ خروجی

برنامه زیر با لبه مثبت پالس دریافتی از CPU Clock همه ۳۲ خروجی را بیت به بیت با FFFFFFFF (که معادل ۳۲ تا یک باینری است) XOR می‌کند. وقتی خروجی‌ها صفر هستند نتیجه XOR یک است و همه روشن می‌شوند و وقتی همه یک هستند نتیجه XOR صفر است و همه خاموش می‌شوند. به این ترتیب همه ورودی ها با هم حالت چشمک‌زن پیدا می کنند.



شکل ۲۳-۵ برنامه مثال ۲۳-۳

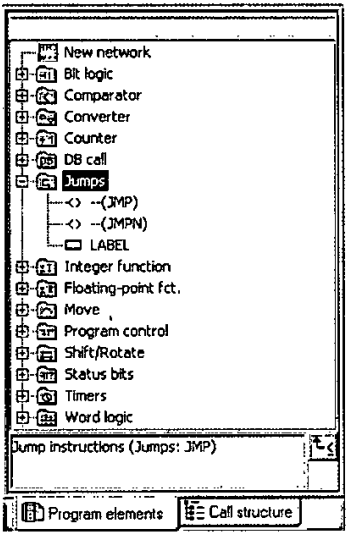
فصل ۲۳

۲۳-۳ دستورات Jump

توسط این دستورات می‌توان از روی بخشی از برنامه پرش نمود، به‌گونه‌ای که Networkهای موجود در آن بخش اجرا نشوند. به‌طور کلی به دو صورت می‌توان از دستورات Jump استفاده نمود:

الف) Jump شرطی: در این‌حالت برای پرش ازیک‌یاچندشرط‌استفاده‌می‌شودکه‌در صورت انجام شرط، پرش انجام پذیرد.

ب) Jump غیر شرطی: در این‌حالت برای پرش از شرط خاصی استفاده نشده و پرش دائماً صورت می‌پذیرد.

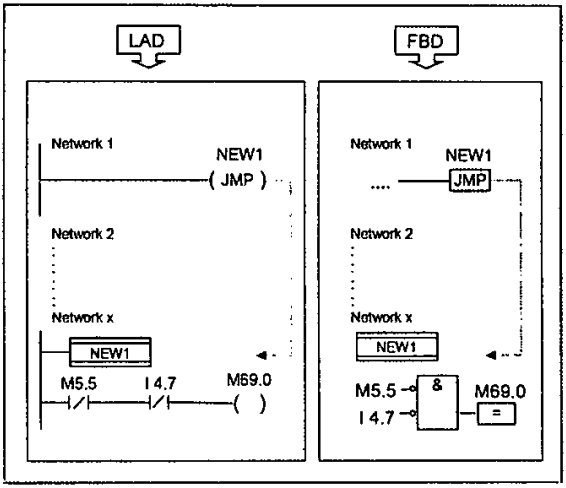


شکل۲۳-۶ محل قرارگیری دستورات JUMP

دستور پرش از دو بخش تشکیل می‌شود:

۱- **دستور پرش:** شامل دستور JMP (پرش در صورتی‌که RLO =1) و JMPN (پرش در صورتی‌که RLO = 0)

۲- **محل پرش (LABEL):** در Networkای که باید عمل پرش به آنجا انجام شود قرار داده می‌شود.

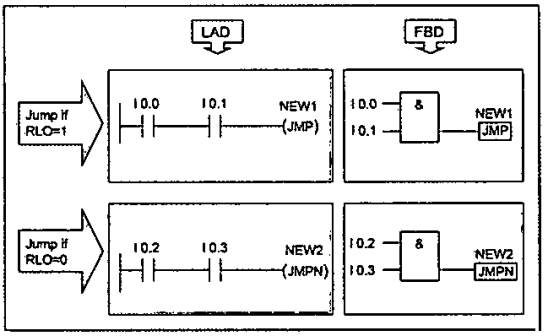


شکل ۲۳-۷ دستور پرش غیر شرطی با استفاده از دستور JMP

همانطور که در شکل ۲۳-۷ مشخص است، در Network1 از دستور JMP (دستور پرش) استفاده شده است. مقصد پرش جایی است که LABEL با عنوان NEW1 قرار دارد و از آنجایی‌که برای عمل پرش شرط خاصی لحاظ نشده است، عمل پرش همواره انجام می‌پذیرد.

در واقع در هر سیکل‌اسکن، برنامه از روی Network 2 تا Network x پرش نموده و تا Network x را اجرا نمی‌نماید. البته خود Network x اجرا می‌شود.

در شکل ۲۳-۸، در برنامه‌ی اول، با فعال‌شدن I0.0 و I0.1 مقدار RLO برابریکِ منطقی‌شده‌و پرش صورت می‌پذیرد.



شکل۲۳-۸ پرش شرطی با استفاده از دستور JMP و JMPN

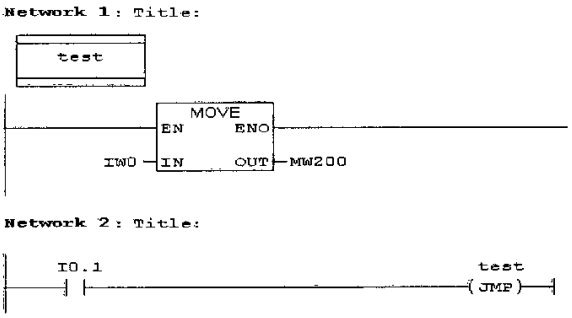
در برنامه‌ی دوم از دستور Jump Not (JMPN) استفاده شده است، بنابراین هرگاه I0.2 یا I0.3 غیرفعال باشند عمل پرش اتفاق می‌افتد.

از آنجایی که در این دو برنامه برای انجام عمل پرش شروطی نیز قرار داده شده است، به آنها پرش شرطی گفته می‌شود.

نکاتی که در استفاده از دستور پرش بایستی به آنها دقت کرد:

- Label نبایستی از چهار کاراکتر بیشتر باشد. کاراکترها می‌توانند حرف یا عدد باشند ولی کاراکتر شروع بایستی حرف باشد.
- پرش می‌تواند به Networkهای ماقبل نیز انجام شود.
- در استفاده از دستور پرش بایستی دقت کرد که برنامه در لوپ نیفتد. اگرCPU نتواند سیکل‌اسکن را در زمان تعیین شده تمام کند Stop می‌گردد. در برنامه زیر در صورتی‌که I0.1 فعال شود برنامه در لوپ بی نهایت می‌افتد و منجر به Stop شدن CPU می‌گردد.

فصل ۲۳



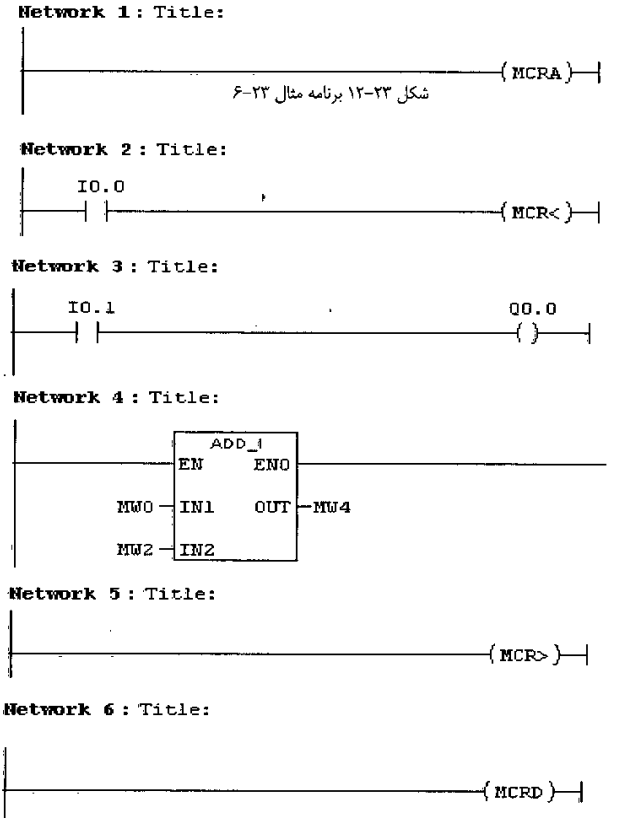
شکل ۲۳-۹ اشکال در برنامه به‌دلیل ایجاد لوپ

مثال ۲۳-۴: محاسبه فاکتوریل

در برنامه زیر فاکتوریل عدد صحیح داده شده به MW0 محاسبه شده و نتیجه به MW100 ریخته می‌شود. اصول کلی کار به اینصورت است که در یک حلقه هر بار یک شماره از عدد کم کرده و آنرا در نتیجه قبلی ضرب می‌کند، و وقتی مقدار کم شده به یک رسید از حلقه خارج می‌شود. تحلیل دقیق به عهده خواننده است.

- برنامه بین دو دستور MCR> و MCR < نوشته می‌شود که به آن ناحیه MCR می‌گویند.
- دستور MCR> ناحیه MCR را باز می کند.
- دستور MCR < ناحیه MCR را می بندد.
- در ناحیه MCR که بین دو دستور فوق قرار می‌گیرد برنامه مورد نظر نوشته می‌شود.
- اگر قبل از ناحیه فوق RLO=1 باشد در اینصورت MCR =1 یعنی ON است و پردازش برنامه عادی است یعنی MCR روی آن تاثیری نمی‌گذارد.
- اگر قبل از ناحیه فوق RLO = 0 شد در اینصورت MCR=0 یعنی OFF خواهد شد و خروجی‌ها تحت تاثیر قرار می‌گیرند. نحوه تاثیر طبق توضیحات صفحه قبل است.

مثال ۲۳-۶: برای فهم موضوع به برنامه‌ی نوشته شده در شکل ۲۳-۱۲ توجه نمایید.



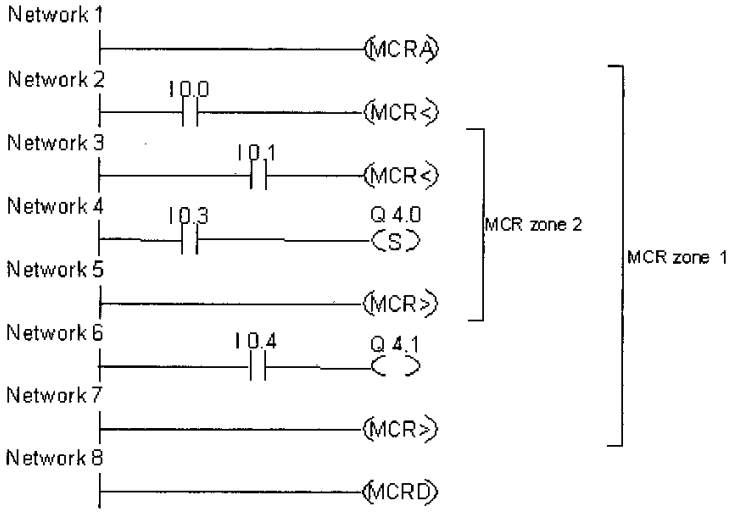
شکل ۲۳-۱۲ برنامه مثال ۲۳-۶

در برنامه فوق، در Network1 توسط دستور MCRA شروع برنامه‌نویسی MCR آغاز شده است. سپس در Network2 از تیغه‌ی باز I0.0 به‌منظور فعال‌نمودن قسمتی از برنامه که تحت تأثیر این دستور است، استفاده شده است. هرگاه I0.0 دارای مقدار یکِ منطقی شود، نتورک‌های ۳ و ۴ که بین دو دستور MCR> و <MCR قرار گرفته‌اند، اجرا شده و کار عادی صورت می‌گیرد ولی اگر I0.0 صفر شد خروجی Q0.0 صفر شده و MW4 نیز دارای مقدار صفر می‌شود.

استفاده از چند دستور MCR به‌صورت تودرتو

توسط دستورات MCR> و <MCR می‌توان حداکثر ۸ دستور تودرتو ایجاد نمود. برای درک بهتر موضوع به شکل۲۳-۱۳ توجه نمایید.

فصل ۲۳



شکل ۲۳-۱۳ استفاده از چند دستور MCR به‌صورت تودرتو

همانطور که در شکل ۲۳-۱۳ مشخص است، دو دستور MCR استفاده شده است. به هر قسمت از برنامه که بین دو دستور MCR> و <MCR قرار گرفته است یک MCR zone گفته می‌شود. در این برنامه دو MCR zone استفاده شده است که zone 2 در داخل zone 1 قرار گرفته است.

۲۳-۵ دستور باز کردن DB

دستور DB Call مربوط به دیتا بلاک است که در کتاب سطح پیشرفته بحث می‌شود.

۲۳-۶ دستورات Status Bits

این دستورات بیت‌های Status Word که یک رجیستر داخل حافظه است را در اختیار قرار می‌دهد. تشریح Status Word در کتاب سطح تکمیلی خواهد آمد.

۲۳-۷ تمرین

۱- در برنامه مثال ۲۳-۴ اگر نتیجه فاکتوریل از ۳۲۷۶۷ بیشتر شود، نتیجه غلط است.آنرا به‌صورتی اصلاح کنید که نتیجه برای اعداد بزرگ نیز مشکلی نداشته باشد.

۲- با استفاده از دستورات Word Logic و Jump برنامه‌ای بنویسید که بتواند در هر لحظه تعداد سوئیچ‌های فعال متصل به یک کارت دیجیتال ۱۶ ورودی را شمارش کند و نتیجه را در MW100 نمایش دهد.

فصل ۲۴

استفاده از Monitor/Modify و Force

۲۴-۱ مقدمه

۲۴-۲ Monitor / Modify

۲۴-۲-۱ ابزارهای Monitor/Modify

۲۴-۲-۲ Monitor/Modify با استفاده از جدول VAT

۲۴-۲-۳ نکات کار با Monitor/Modify

۲۴-۳ انجام Force

۲۴-۳-۱ نحوه دسترسی به پنجره Force

۲۴-۳-۲ تفاوت Force و Modify

۲۴-۳-۳ Force با استفاده از جدول VAT

۲۴-۴ پرسش و تحقیق

در این فصل با نحوه استفاده از ابزارهای Monitor/Modify و Force که در زمان تست و راه‌اندازی و نیز در زمان بهره‌برداری مورد نیاز هستند آشنا می‌شوید.

چکیده مطالب

- Monitor/Modify و Force یکی از ابزارهای مهم در فاز راه‌اندازی و فاز بهره‌برداری هستند.
- جدول VAT بهترین روش برای استفاده از ابزارهای فوق است.
- Monitor فقط برای نمایش وضعیت متغیرهاست.
- Modify برای تغییر وضعیت متغیرهاست. برای اعمال تغییر روی ورودی و خروجی‌ها بایستی تنظیمات تریگر به‌درستی انجام شده باشد.
- Force برای اعمال تغییر به یک متغیر است. در این حالت برخلاف Modify برنامه تاثیرگذار نیست.
- Force و Modify بایستی با رعایت جوانب ایمنی فعال گردند.

فصل
۲۴

۲۴-۱ مقدمه

Monitor، Modify و Force از ابزارهای متداول کاربران PLC است. توسط این ابزارها امکان مشاهده وضعیت سیگنال‌ها و متغیرها (Monitor یا تغییر در وضعیت برخی از آنها (Modify/Force) وجود دارد. از این ابزارها در مراحل مختلف کار استفاده می‌شود.

فاز تست و راه اندازی

از جمله کاربردهای Monitor/Modify/Force در فاز تست و راه‌اندازی می‌توان به موارد زیر اشاره نمود:

- تست سیم‌کشی‌ها و اتصالات ورودی، خروجی
- اعمال مقادیر دلخواه به خروجی‌های موجود در فرآیند با در نظر گرفتن منطق برنامه
- دیدن وضعیت سیگنال‌های ورودی و اطمینال از صحت سیم‌کشی و تجهیز مربوطه

فاز بهره برداری

از جمله کاربردهای Monitor/Modify/Force در فاز بهره‌برداری می‌توان به موارد زیر اشاره نمود:

- مشاهده وضعیت سیگنال‌ها به‌منظور بررسی روند اجرای برنامه
- بررسی صحت عملکرد تجهیزات و سنسورها
- رفع موقت مشکل ناشی از خرابی سنسور یا کابل

در این فصل با نحوه انجام اعمال Monitor و Modify و نیز عمل Force و خصوصیات آنها آشنا می‌شوید برای استفاده از سه ابزار فوق امکانات مختلفی در نرم افزار Step7 پیش‌بینی شده است. معمولاً برای Monitor/Modify پنجره مشابهی وجود دارد ولی Force دارای پنجره مستقلی است.

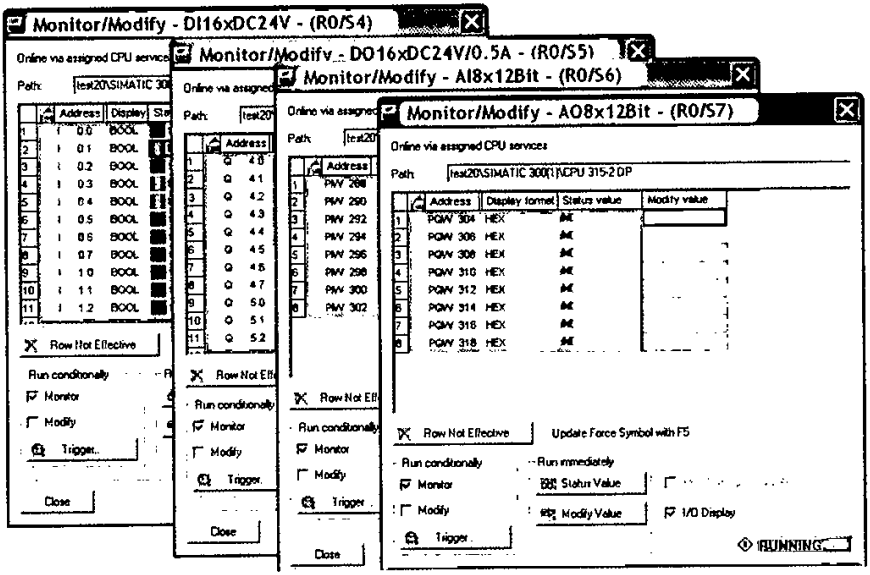
۲۴-۲ Monitor / Modify

۲۴-۲-۱ ابزارهای Monitor/Modify

ابزارهای Monitor/Modify در محیط‌های مختلف نرم افزار قابل دسترس است. به‌شرح زیر:

الف) در محیط پیکربندی سخت افزار HWConfig

در این محیط همانطور که در فصل‌های قبل اشاره شد، در حالی که ارتباط کامپیوتر با PLC برقرار است روی کارت ورودی یا خروجی کلیک راست کرده و گزینه Monitor/Modify را انتخاب می‌کنیم. شکل ۲۴-۱ نمونه‌هایی از این پنجره را برای کارت‌های مختلف نشان می‌دهد.



شکل ۲۴-۱ پنجره‌های Monitor/Modify در محیط Hwconfig

فصل ۲۴

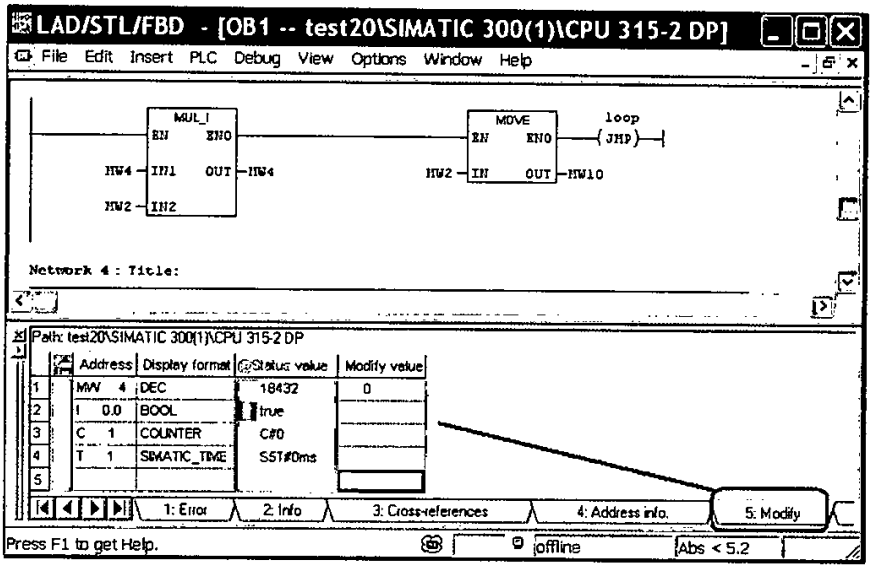
باید توجه داشت که:

- این پنجره فقط برای ورودی و خروجی‌هاست. سایر متغیرهای حافظه در این پنجره قابل دسترس نیستند.
- برای مشاهده سیگنال‌ها بایستی گزینه Monitor را در پایین پنجره فعال نمود.
- برای تغییر سیگنال‌ها بایستی ابتدا مقدار در ستون Modify وارد کرده سپس گزینه Modify را در پایین پنجره فعال شود.
- مقادیر آنالوگ ورودی که به‌صورت PIW هستند فقط قابل نمایش هستند و نمی‌توان آنها را Modify کرد.
- مقادیر آنالوگ‌های خروجی PQW قابل نمایش نیستند، ولی می‌توان آنها را Modify کرد.
- تنظیمات این پنجره با کلیک روی کلید Trigger فعال می‌شود. این تنظیمات در ادامه تشریح می‌شود.

ب) در محیط برنامه‌نویسی LAD/FBD/STL

در محیط برنامه‌نویسی ابتدا از منوی View گزینه Detail را فعال می‌کنیم تا نوارابزار پایین صفحه نمایش داده شود. سپس روی قسمت Modify کلیک کرده تا پنجره ای مانند شکل ۲۴-۲ باز شود.

در این پنجره آدرس‌های مورد نظر را وارد کرده، سپس با کلیک راست Monitor را فعال می‌کنیم. در صورت نیاز به تغییر می‌توان مقدار مورد نظر را در ستون Modify وارد کرده و با کلیک راست روی سطر یا جدول آن را تغییر داد.



شکل ۲۴-۲ پنجره Monitor/Modify در محیط LAD/STL/FBD

باید توجه داشت که:

- در این محیط علاوه بر ورودی و خروجی‌ها سایر آدرس‌های PLC نیز در دسترس هستند.
- متغیرها را به فرمت دلخواه می توان دید. کافیست در ستون Display Format نوع نمایش را با کلیک راست مشخص نماییم.
- مشاهده متن برنامه در حالت Online نتیجه را فقط برای همان Network نشان می‌دهد ولی جدول فوق نتیجه نهایی را مشخص می‌کند.

ج) در محیط Simatic Manager با استفاده از ابزار VAT

جدول VAT مناسب‌ترین روش برای نمایش و تغییر آدرس‌ها می‌باشد که در ادامه توضیح داده می‌شود.

۲۴-۲-۲ Monitor/Modify با استفاده از جدول VAT

مراحل مختلف انجام Monitor/Modify به شرح زیر می‌باشد:

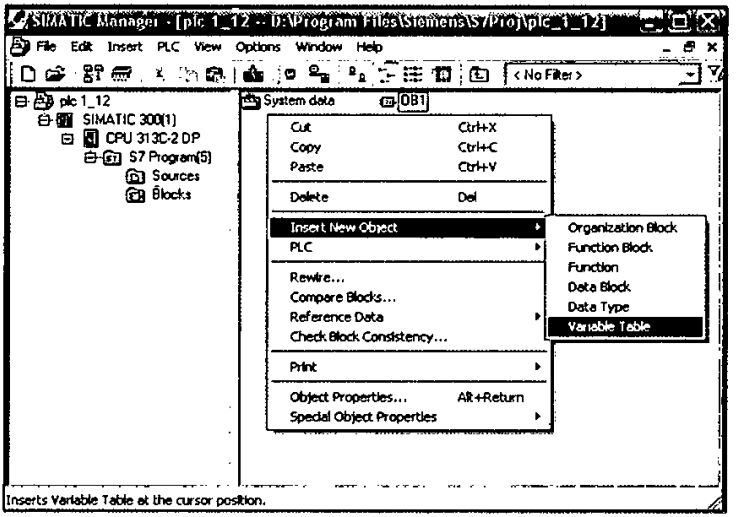
۱- ایجاد جدول VAT

۲- تعریف متغیرها در آن

۳- انجام عمل Monitor و Modify با استفاده از ابزارهای موجود در جدول VAT

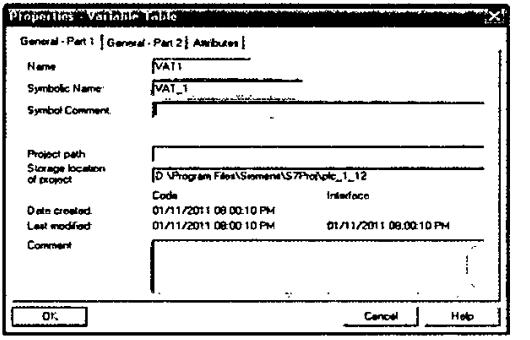
ایجاد جدول VAT

برای ایجاد جدول VAT می‌توان مطابق شکل ۲۴-۳ در محیط Simatic Manager کلیک راست نموده و از قسمت Insert New Object، گزینه Variable Table را انتخاب نمود.



شکل ۲۴-۳ ایجاد جدول VAT

در اینصورت کادری مانند شکل ۲۴-۴ باز می‌شود که می‌توان در آن نام سمبلیک بلاک و برخی مشخصات دیگر را تعیین نمود. با انتخاب گزینه OK بلاک ایجاد می‌شود.



شکل ۲۴-۴ تعریف جدول VAT

تعریف متغیرها در جدول VAT

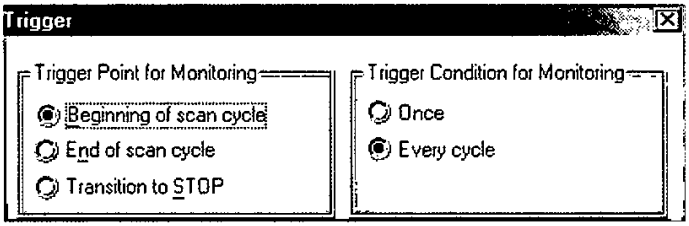
با دوبارکلیک نمودن بر روی بلاک ایجاد شده، محیطی مانند شکل ۲۴-۵ نشان داده می‌شود. برای تعریف یک متغیر جدید در جدول VAT می‌توان در سطر دلخواه کلیک نموده و در ستون Address، آدرس متغیر را وارد نمود.

در ارتباط با استفاده از آیکن‌های فوق به نکات زیر توجه شود:

- اگر از آیکن شماره ۵ استفاده شود، حالت Modify برای یک سیکل فعال می‌شود.
- در صورتی‌که از آیکن شماره ۴ استفاده شود، حالت Monitor برای سیکل‌اسکن جاری به‌روز می‌شود.
- در صورتی‌که از آیکن شماره ۲ استفاده شود، حالت Monitor فعال شده و مرتباً به‌روز می‌شود.
- در صورتی‌که از آیکن شماره ۳ استفاده شود، حالت Modify مطابق شرایط تعیین شده در قسمت تریگر فعال می‌شود.

تنظیمات تریگر برای Monitor

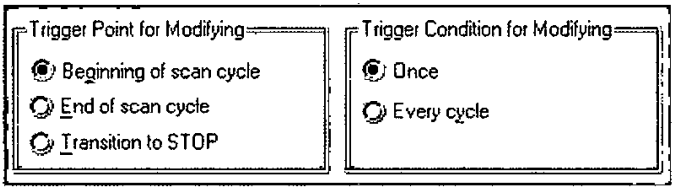
می‌توان با استفاده از منوی Varriable و انتخاب Triger در پنجره‌ای مانند شکل ۲۴-۸ مشخص کرد که آیا متغیر در یک سیکل Update شود یا به‌طور مداوم، و همینطور می‌توان انتخاب کرد که در ابتدای سیکل Update شود یا در انتهای آن. شکل ۲۴-۸ انتخاب‌های پیش‌فرض سیستم را نیز نشان می‌دهد.



شکل ۲۴-۸ تنظیم تریگر برای Monitoring

تنظیمات تریگر برای Modify

برای Modify کردن نیز Triger وجود دارد. می‌توان از منوی Varriable > Triger m در پنجره‌ای مانند شکل ۲۴-۹ انتخاب‌های پیش‌فرض را دید و آنها را در صورت لزوم تغییر داد.



شکل ۲۴-۹ تنظیم تریگر برای Modifying

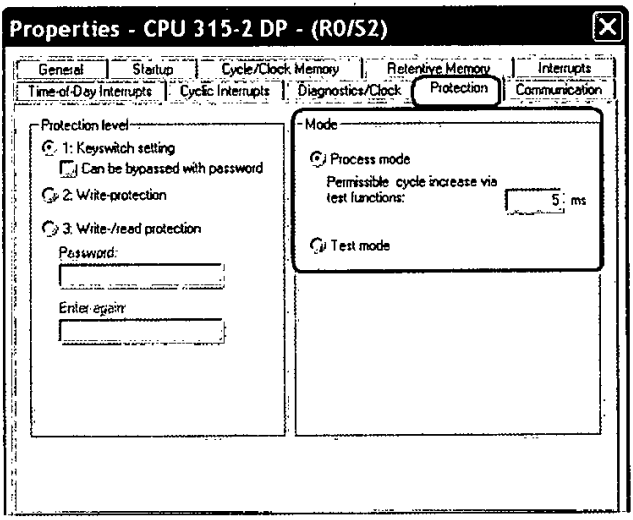
باید توجه داشت که:

- انتخاب Every Cycle از نظر ایمنی خطرناکتر از Once است زیرا مرتباً متغیر را به مقدار مورد نظر Update می‌کند.
- از آنجا که در سیکل اسکن ابتدا ورودی‌ها خوانده می‌شود و سپس برنامه اجرا می‌گردد، اگر بخواهیم یک ورودی را Modify حتماً باید Beginning of Scan Cycle انتخاب شده باشد. در غیر اینصورت یعنی اگر End of Scan Cycle انتخاب شود، نتیجه را نمی بینیم زیرا برنامه مقدار واقعی را از کارت ورودی خوانده و پردازش را انجام داده است.
- اگر بخواهیم یک خروجی را Modify کنیم حتماً باید End of Scan Cycle انتخاب شده باشد. در غیر اینصورت یعنی اگر Beginning of Scan Cycle انتخاب شود نتیجه را نمی‌بینیم زیرا برنامه قبلاً در انتهای سیکل اسکن مقدار واقعی را به کارت خروجی فرستاده است.

۲۴-۲-۳ نکات کار با Monitor/Modify

- **ورودی‌های دیجیتال:** می‌توان ورودی‌های دیجیتال را Monitor یا Modify نمود. البته در Modify چون دیتای ورودی از کارت ورودی دریافت می‌گردد، مقدار Modify شده سریعاً متناسب با مقدار ارسالی از کارت تغییر می‌کند مگر اینکه تنظیم تریگر روی Beginning Of Scan Cycle باشد.
- **خروجی‌های دیجیتال:** می‌توان خروجی‌های دیجیتال را Monitor و Modify نمود. در حالت Modify بایستی تریگر روی End of Scan Cycle باشد.
- **ورودی‌های آنالوگ درون رنج:** اگر آدرس آنالوگی درون رنج ناحیه PII باشد (یعنی به‌صورت IW آدرس دهی شده باشد)، می‌توان آنرا Monitor و Modify نمود.
- **ورودی‌های آنالوگ (خارج از رنج):** ورودی‌های آنالوگ از نوع Peripheral (که به‌صورت PIW آدرس دهی شده‌اند) را فقط می‌توان Monitor نمود و امکان Modify نمودن آنها وجود ندارد.
- **خروجی‌های آنالوگ درون رنج:** امکان Monitor و Modify نمودن خروجی‌های آنالوگ درون رنج PIQ (که به‌صورت QW آدرس دهی می‌شوند) وجود دارد.
- **خروجی‌های آنالوگ (خارج از رنج):** برای این نوع خروجی‌ها که با آدرس PQW نمایش داده می‌شوند فقط امکان Modify نمودن وجود داشته و انجام Monitor امکان‌پذیر نمی‌باشد.
- **Bit Memory:** می‌توان Memory Bitها را با هر نوع آدرس‌دهی که باشند Monitor و Modify نمود.
- **Timer، Counter:** تایمر و کانتر را می‌توان Monitor و Modify نمود.
- **Data Block:** آدرس‌های موجود در یک دیتابلاک را می‌توان Monitor و Modify نمود.
- در برنامه‌های بزرگ معمولاً چندین جدول VAT تعریف می‌کنند و فرآیندها یا متغیرها را بصورت دسته‌بندی شده در آنها قرار می دهند. به‌عنوان مثال یک جدول VAT فقط به آنالوگ‌های ورودی اختصاص می یابد.

نکته : اگر در تنظیمات CPU که در پنجره زیر نشان داده شده گزینه Process Mode فعال باشد Modify کردن امکان پذیر نیست.



شکل ۲۴-۱۰ تنظیم پارامترهای CPU که روی Modify تاثیرگذار است.

۲۴-۳ انجام Force

Force نیز برای تغییر دادن برخی ورودی و خروجی‌ها به‌کار می‌رود ولی عملکرد آن با Modify متفاوت است که در ادامه تشریح می‌گردد. توسط Force می‌توان بدون در نظر گرفتن منطق برنامه، مقدار متغیری را تغییر داد. تغییر اعمال شده تا زمانی‌که حالت Force فعال است، باقی می‌ماند.

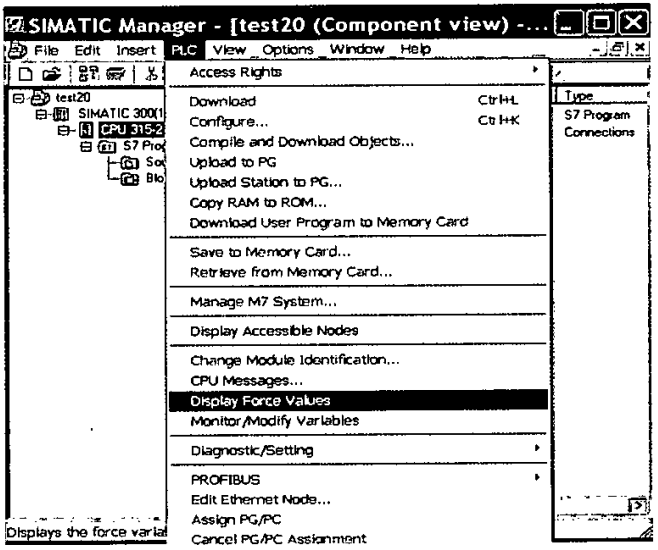
باید توجه داشت که:

- استفاده از Force بایستی با احتیاط صورت گیرد چون ممکن است منجر به خسارت جانی و مالی شود. Force به مراتب خطرناک‌تر از Modify است.
- تا حد امکان خروجی‌ها مستقیماً Force نشوند بلکه ورودی‌ها را Force کنید تا بقیه لاجیک کنترل تاثیرگذار باشد. به‌عنوان مثال اگر شرط روشن شدن پمپ آب وجود آب در مسیر و حس نمودن آن توسط یک فلوسوئیچ باشد، و سنسور دچار مشکل شود نباید آدرس پمپ را Force کرد بلکه باید به آدرس فلوسوئیچ Force اعمال نمود تا بقیه اینترلاک‌ها کار کنند.

۲۴-۳-۱ نحوه دسترسی به پنجره Force

الف) در محیط Step 7

در این محیط ابتدا CPU مورد نظر را انتخاب کرده سپس از منوی PLC روی گزینه Display Force Value کلیک می‌کنیم.



شکل ۲۴-۱۱ فعال‌سازی پنجره Force از طریق Simatic Manager

ب) از طریق جدول VAT

با ساختن جدول VAT به شرحی که در ادامه ذکر می‌شود، می‌توان به پنجره Force دسترسی پیدا کرد.

۲۴-۳-۲ تفاوت Force و Modify

تفاوت‌های اصلی به‌شرح زیر هستند:

۱- Force روی برخی از آدرس‌ها و متغیرهای حافظه مانند کانتر و تایمر و دیتا بلاک تاثیر گذار نیست ولی Modify می‌تواند آنها را نیز تغییر دهد. مقایسه در جدول ۲۴-۳ آمده است.

جدول ۲۴-۳ امکان Force و Modify در S7-300 و S7-400

	S7-400		S7-300	
نوع متغیر	Force	Modify	Force	Modify
Bit memory (M)	بله	بله	خیر	بله
Timers and counters (T, C)	خیر	بله	خیر	بله
Data blocks (DB)	خیر	بله	خیر	بله
Peripheral inputs (PIB, PIW, PID)	بله	خیر	خیر	خیر
Peripheral outputs (PQB, PQW, PQD)	بله	بله	خیر	بله
Inputs and outputs (I, Q)	بله	بله	بله	بله

۲- در Force برنامه تاثیرگذار نیست ولی در Modify برنامه تاثیرگذار است. از اینرو ممکن است اگر تنظیم تریگر برای Modify درست نباشد برنامه تاثیر Modify را از بین ببرد.

۳- Force چراغ FRCE روی CPU را روشن می‌کند ولی Modify منجر به روشن شدن چراغ خاصی نمی‌شود.

۴- Modify فقط تا زمانی که کامپیوتر به PLC وصل است و گزینه Modify فعال است اعمال می‌شود. اگر کامپیوتر خاموش شود یا ارتباط آن قطع شود Modify از کار می افتد ولی Force پس از اعمال نیازی به ارتباط کامپیوتر ندارد و فعال می‌ماند.

۴- Force در S7-300 جدید با قطع و وصل تغذیه از بین نمی‌رود. در S7-400 و S7-300 قدیمی که باتری پشتیبان دارند نیز Force پس از برگشت تغذیه فعال می‌ماند؛ ولی Modify با قطع تغذیه از بین می‌رود و با بازگشت تغذیه باید مجدداً به‌صورت دستی فعال شود.

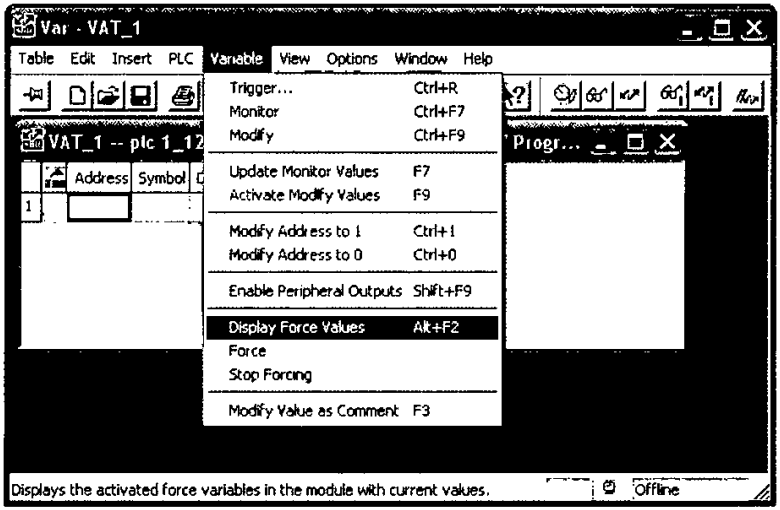
۵- جدول Force نیاز به ذخیره‌سازی ندارد و در CPU حفظ می‌شود. جدول Modify نیاز به ذخیره‌سازی روی هارد کامپیوتر دارد.

تذکر : Force و Modify هر دو با ری‌ست کردن CPU از بین می‌روند.

۲۴-۳-۳ Force با استفاده از جدول VAT

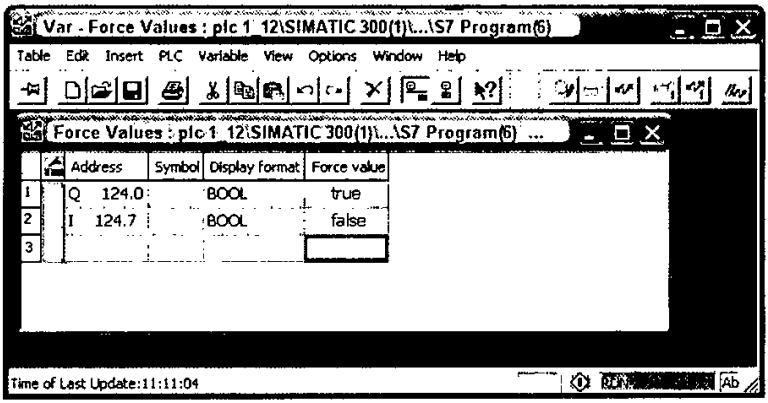
برای انجام عمل Force می‌توان مطابق مراحل زیر عمل نمود.

۱- از منوی Variable گزینه Display Force Values را انتخاب نمایید.



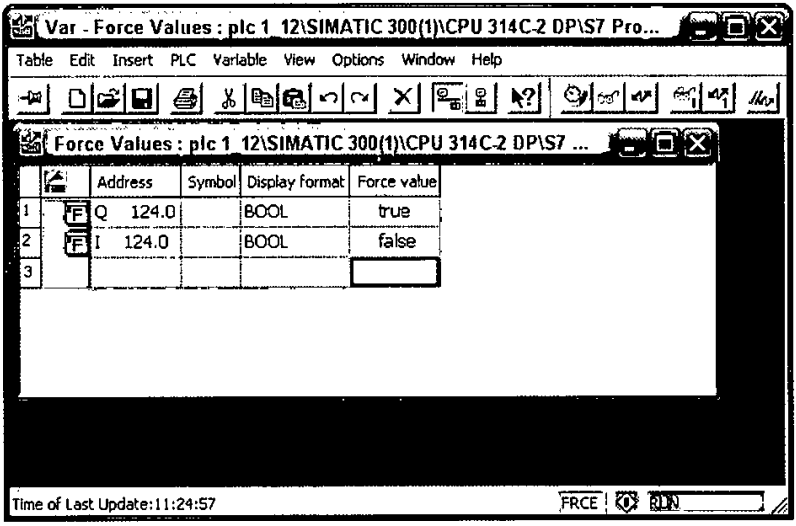
شکل ۲۴-۱۲ فعال‌سازی پنجره Force از طریق VAT

۲- در پنجره باز شده، آدرس متغیر یا متغیرهایی که قرار است Force شوند را وارد نموده و در ستون Force Value مقدار مورد نظر جهت Force را وارد نمایید.



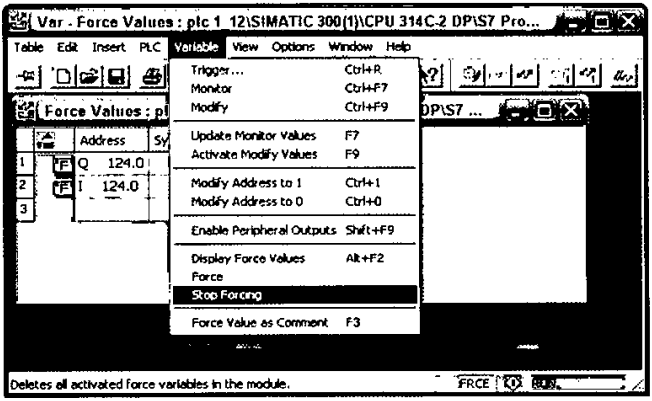
شکل ۲۴-۱۳ وارد نمودن متغیرهای مورد نظر جهت انجام Force

۳- از منوی Variable گزینه Force را انتخاب نمایید. در اینصورت در کنار آدرس متغیرها علامت F با رنگ قرمز نشان داده می‌شود که بیانگر فعال بودن Force برای این متغیرهاست. همچنین در قسمت پایین این محیط، علامت FRCE با رنگ زرد نشان داده می‌شود که بیانگر فعال بودن حالت Force است.



شکل ۲۴-۱۴ فعال نمودن حالت Force

۴- جهت غیرفعال نمودن Force از منوی Variable گزینه Stop Forcing را انتخاب نمایید، یا روی سطرهای جدول کلیک راست کرده و گزینه Delete Force را انتخاب کنید.

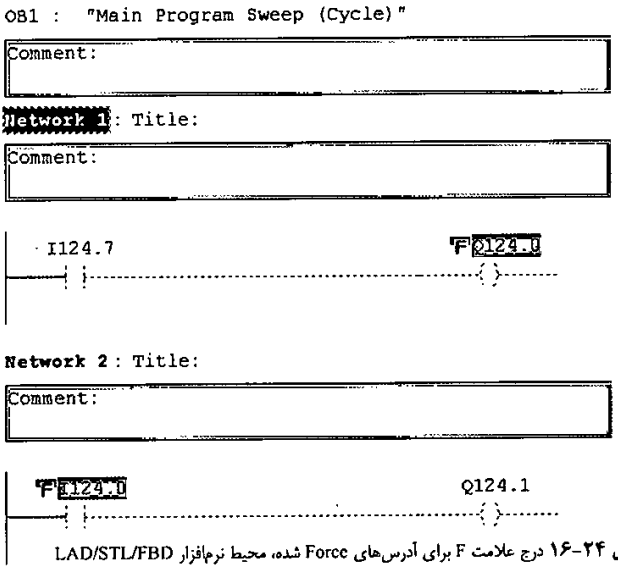


شکل ۲۴-۱۵ غیرفعال نمودن حالت Force

تذکر: با Stop Forcing یا Delete Force همه آدرس‌های جدول از Force خارج می‌شوند. اگر فقط آدرس خاصی باید از Force خارج شود نباید به‌طریق فوق عمل کرد، بلکه باید سطری که این آدرس در آن نوشته شده را پاک نمود و مجدداً جدول جدید را Force کرد.

نکات Force کردن

- هنگامی که قابلیت Force فعال شود، لامپ FRCE روی CPU روشن می‌شود. رنگ این لامپ در هنگام روشن شدن زرد می‌باشد و با غیرفعال شدن Force این لامپ نیز خاموش می‌شود.
- در هنگام فعال بودن Force چنانچه از محیط LAD/STL/FBD برنامه را Monitor نمایید، علامت F با رنگ قرمز روی آدرس‌های Force شده نشان‌داده می‌شود. در صورتی‌که علامت F نشان داده نشد، در محیط LAD/STL/FBD از منوی View > Display with > Address Identification این قابلیت را فعال نمایید.



شکل ۲۴-۱۶ درج علامت F برای آدرس‌های Force شده، محیط نرم‌افزار LAD/STL/FBD

- در صورت غیر فعال نمودن Force علامت F بالای متغیرها نشان داده نشده و لامپ FRCE روی CPU نیز خاموش می‌گردد.
- Force با سیمولاتور قابل شبیه‌سازی نیست.

۴-۲۴ پرسش و تحقیق

جدول Force در چه ناحیه‌ای از حافظه CPU ذخیره می‌شود؟

فصل ۲۵

ارتباط Online با PLC

۲۵-۱ مقدمه

۲۵-۲ Download / Upload

۲۵-۲-۱ Download و Upload در محیط Simatic Manager

۲۵-۲-۲ Download و Upload در محیط HW Config

۲۵-۲-۳ Download و Upload در محیط LAD/STL/FBD

۲۵-۲-۴ Download و Upload در محیط NetPro

۲۵-۳ Simatic Manager در نمایش Online

۲۵-۴ محیط پیکربندی سخت‌افزار در نمایش Online

۲۵-۵ محیط برنامه‌نویسی در نمایش Online

۲۵-۶ پرسش و تحقیق

در این فصل با پنجره‌های مختلفی که در حالت ارتباط Online با PLC نمایش داده می‌شود آشنا خواهید شد.

چکیده مطالب

- یکی از کارهای متداول در ارتباط Online انجام عملیات Download/Upload است.
- Download/Upload می‌تواند از طریق محیط اصلی Step7 یا زیر برنامه‌های آن انجام شود.
- در محیط اصلی Step7 همه اطلاعات Download یا Upload می‌شود ولی در زیر برنامه‌ها فقط اطلاعات همان محیط ارسال یا دریافت می‌شود.
- اگر پروژه را در Step7 به‌صورت Online ببینیم علاوه بر بلاک‌های کاربر، بلاک‌های سیستمی نیز مشاهده می‌شود که در پروژه Offline موجود نیستند.
- اگر Hwconfig را در حالت Online مشاهده کنیم وضعیت سخت‌افزار را نشان می‌دهد.
- مهمترین پنجره Online برای عیب‌یابی پنجره Module Information است که با کلید میانبر Ctrl+D فعال می شود.
- محیط برنامه‌نویسی را نیز می‌توان به‌صورت Online مشاهده نمود.

فصل
۲۵

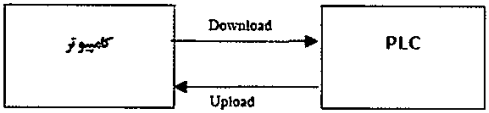
۲۵-۱ مقدمه

منظور از ارتباط Online با PLC عملیات مختلفی است که در حالت اتصال با PLC می‌توان انجام داد. این عملیات عبارتند از:

- Download / Upload
- مشاهده پروژه در حالت online
- مشاهده سخت‌افزار در حالت online
- مشاهده برنامه در حالت Online
- مشاهده پنجره Module Information

۲۵-۲ Download / Upload

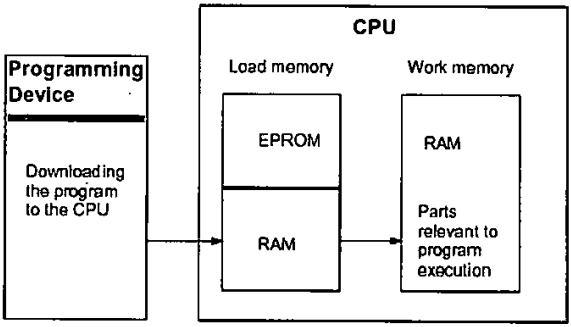
یکی از نکات مهم در هنگام کار با یک PLC، امکان Download و Upload می‌باشد. منظور از Download، انتقال برنامه از PC به PLC و منظور از Upload، انتقال برنامه از PLC به PC می‌باشد.



شکل۲۵-۱ Download و Upload

Download

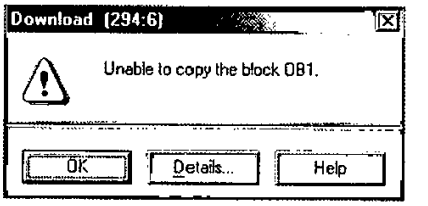
هنگامی‌که برنامه یا تنظیمات سخت‌افزار به PLC منتقل (Download) می‌شود، این اطلاعات در ابتدا به Load Memory منتقل شده و سپس به Work Memory منتقل می‌گردد.



شکل ۲۵-۲ Download برنامه از PC به PLC

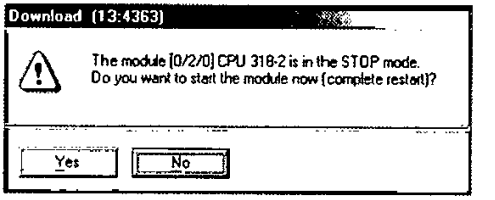
در ارتباط با Download به PLC نکات زیر قابل توجه می‌باشد:

- جدول سمبل‌ها، بلاک‌های UDT و VAT قابل Download نیستند و روی هارد کامپیوتر باقی می‌مانند.
- می‌توان قبل از انجام عمل Download، حافظه PLC را کامل پاک نمود. در غیر اینصورت فایل‌های Download شده روی فایل‌های قبلی Overwrite می‌شوند.
- در CPUهای قدیمی، Download برنامه فقط در حالت کاری RUN-P قابل انجام می‌باشد. در این CPUها در صورتی‌که عمل Download در حالت کاری RUN انجام شود، پیام اخطاری مانند شکل ۲۵-۳ ظاهر می‌گردد.



شکل ۲۵-۳ پیام خطا Download در حالت کاری RUN

- در حالت کاری RUN-P، اگر System Data (اطلاعات پیکربندی سخت‌افزار) به PLC دانلود شود؛ پیامی مبنی بر توقف PLC نشان‌داده می‌شود که با کلیک بر روی گزینه‌ی Yes عملاً CPU متوقف شده و عمل Download انجام می‌پذیرد. در صورتی‌که گزینه No انتخاب شود، عمل Download انجام نشده و CPU نیز متوقف نمی‌شود.



شکل ۲۵-۴ پیام انتخاب توقف CPU و انجام عمل Download

- در برنامه‌نویسی ساختاریافته که از FC و FB استفاده می‌شود، لازم است که کلیه‌ی بلاک‌ها به CPU دانلود شوند، در غیر اینصورت هنگامی که بلاک مورد نظر فراخوانی شود CPU با خطای عدم وجود بلاک مواجه شده و متوقف می‌شود. در این حالت لامپ SF (System Fault) نیز روشن می‌شود.
- اگر پیکربندی انجام شده در محیط HW Config با پیکربندی موجود در PLC متفاوت باشد، پیام خطایی نمایش داده می‌شود. در اینحالت با توجه به تنظیماتی که برای CPU انجام شده است، ممکن است CPU

متوقف شده یا شروع به‌کار نماید. در هر حال بهتر است که جهت جلوگیری از بروز اشکالات، پیکربندی انجام‌شده با پیکربندی واقعی یکسان باشد.

انجام عمل Download و Upload در محیط‌های مختلف برنامه‌ی Simatic Manager و زیربرنامه‌های آن امکان‌پذیر است که در هر برنامه محتویات مواردی که Download و Upload می‌شوند متفاوت می‌باشد. مهم‌ترین محیط‌های برنامه جهت Download و Upload عبارتند:

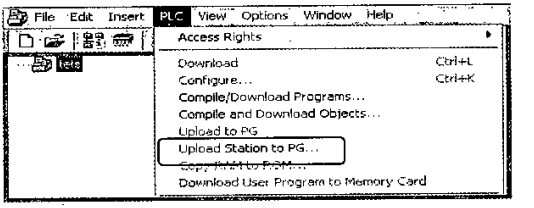
- محیط Simatic Manager
- محیط HW Config
- محیط LAD/STL/FBD
- محیط NetPro

۱-۲-۲۵ Download و Upload در محیط Simatic Manager

در محیط نرم‌افزار Simatic Manager می‌توان Download و Upload برنامه را انجام داد که در اینصورت می‌توان کل محتویات پروژه .شامل پیکربندی، تنظیمات شبکه و بلاک‌های برنامه‌نویسی را Download و Upload نمود. همچنین امکان اینکه فقط بلاک‌های برنامه‌نویسی نیز Download شود، نیز وجود دارد.

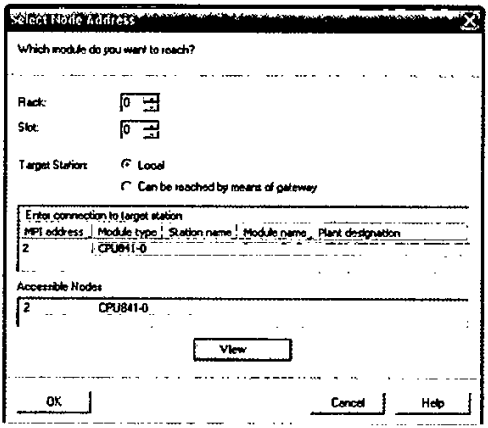
الف) Upload پروژه

برای Uploadگیری از یک PLC می‌توان ابتدا یک پروژه جدید ایجاد نموده و سپس مطابق شکل۲۵-۵ از منوی PLC گزینه Upload Station to PG را انتخاب نمود. این‌عمل‌زمانی که با یک سیستم جدید مواجه می‌شویم مفید خواهد بود.



شکل۲۵-۵ آپلود کل پروژه

در اینصورت کادری مانند شکل ۲۵-۶ باز می‌شود که می‌توان با کلیک بر روی گزینه‌ی View، Stationهای موجود را مشاهده و Station مورد نظر را انتخاب و با کلیک بر روی گزینه‌ی OK عمل Upload را انجام داد. در اینصورت کل محتویات Station مورد نظر شامل پیکربندی، تنظیمات شبکه و بلاک‌های برنامه‌نویسی وارد PC می‌شوند.



شکل ۲۵-۶ انتخاب Station و انجام عمل Upload

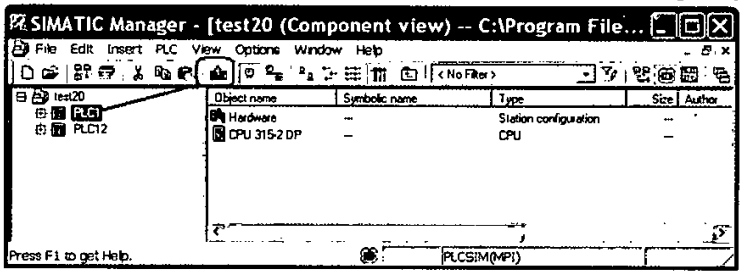
تذکر

- وقتی ارتباط با PLC از طریق پورت CPU نیست بلکه ارتباط از طریق کارت شبکه است بایستی در قسمت Slot شماره اسلاتی که CPU در آن قرار دارد به‌صورت دستی وارد شود.
- در سیستم‌های افزونه که CPU دوم در رک شماره ۱ قرار دارد برای آپلود از آن بایستی شماره Rack در این پنجره ۱ وارد شود.

ب) Download پروژه

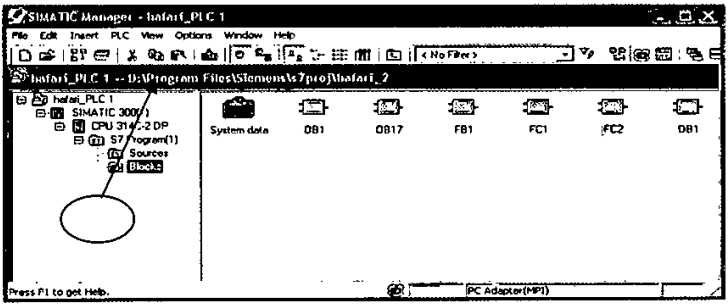
برای Download از محیط Siamtic Manager می‌توان مراحل زیر را انجام داد:

در صورتیکه می‌خواهید کل پروژه Download شود، می‌توانید در ساختار درختی موجود در سمت چپ محیط نرم‌افزار Simatic Manager روی Station مورد نظر کلیک نموده و سپس از منوی PLC گزینه Download را انتخاب نمود یا اینکه روی آیکن Download در Toolbars کلیک نمود.



شکل۲۵-۷ Download کل محتویات یک Station به PLC از محیط نرم‌افزار Simatic Manager

در صورتی‌که می‌خواهید فقط بلاک‌ها را Download نمایید، می‌توانید در ساختار درختی موجود در نرم‌افزار Simatic Manager گزینه Blocks را انتخاب نموده و مطابق شکل ۲۵-۸ روی گزینه Download کلیک نمود.



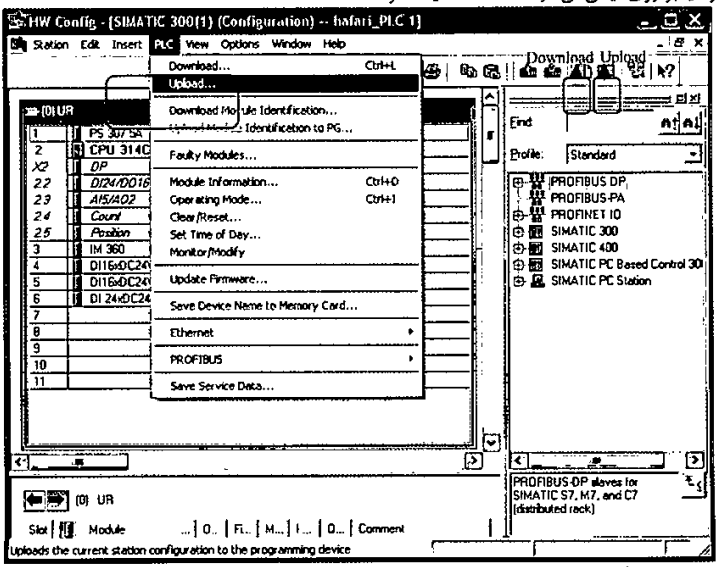
شکل ۲۵-۸ Download بلاک‌های مختلف به PLC از محیط نرم‌افزار Simatic Manager

در صورتی‌که از روش فوق استفاده شود، فقط بلاک‌های موجود در پوشه Blocks به PLC منتقل شده و سایر قسمت‌ها از جمله پیکربندی سخت‌افزار و تنظیمات شبکه منتقل نمی‌شود.

۲۵-۲-۲ Download و Upload در محیط HW Config

در محیط نرم‌افزار HW Config (پیکربندی سخت‌افزار) فقط امکان انتقال پیکربندی و تنظیمات شبکه از PC به PLC و بالعکس (از PLC به PC) وجود دارد.

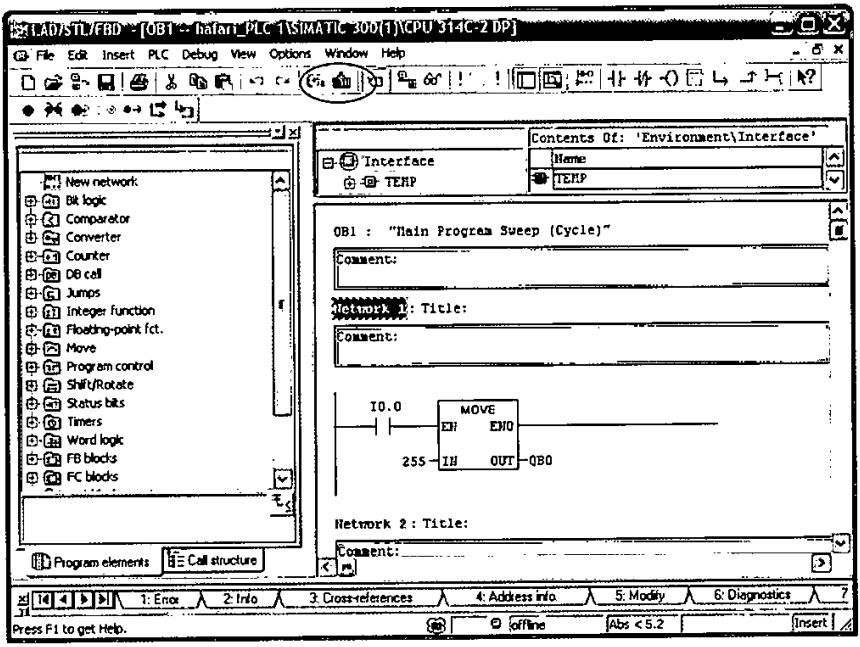
برای انجام عمل Download می‌توان مطابق شکل ۲۵-۹ از منوی PLC گزینه Download را انتخاب نمود یا بر روی آیکن آن در Toolbar کلیک نمود. همچنین برای انجام عمل Upload می‌توان از مسیر گفته شده گزینه Upload را انتخاب نمود یا بر روی آیکن آن در Toolbar کلیک کرد.



شکل ۲۵-۹ انجام عمل Download و Upload از محیط نرم‌افزار HW Config

۳-۲-۲۵ Download و Upload در محیط LAD/STL/FBD

از محیط LAD/STL/FBD فقط امکان انجام عمل Download وجود داشته و انجام Upload امکان‌پذیر نمی‌باشد. برای انجام عمل Download می‌توان از منوی PLC گزینه Download را انتخاب نموده یا بر روی آیکن آن در Toolbar کلیک نمود.



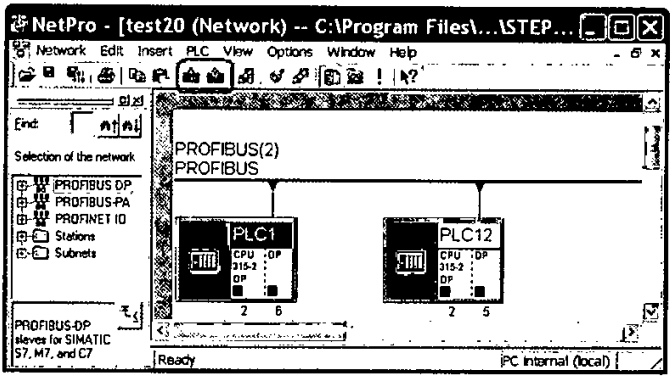
شکل ۲۵-۱۰ انجام عمل Download از محیط نرم‌افزار LAD/STL/FBD

در اینصورت محتویات بلاکی که در محیط نرم‌افزار LAD/STL/FBD باز است به PLC منتقل می‌شود.

۴-۲-۲۵ Download و Upload در محیط NetPro

در محیط Netpro که برای شبکه‌بندی استفاده می‌شود معمولاً چند PLC مانند شکل ۲۵-۱۱ وجود دارند. ابتدا PLC مورد نظر را انتخاب و روی آیکن دانلود از نوارابزار کلیک می‌کنیم . این کار باعث می‌شود که تنظیمات سخت افزار و شبکه به PLC منتقل گردد.

فصل ۲۵



شکل۲۵-۱۱ Download و Upload از محیط NetPro

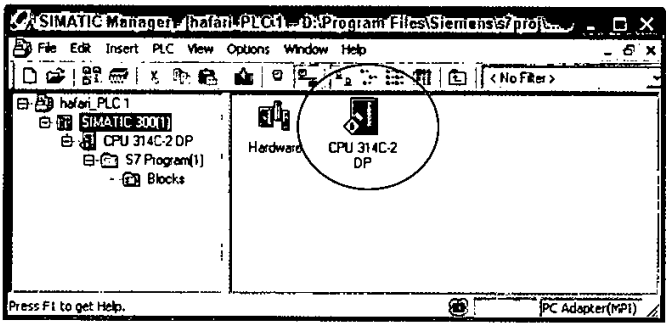
با استفاده از آیکن آپلود می‌توان اطلاعات سخت افزار و شبکه یک PLC را به محیط Netpro وارد نمود.

۳-۲۵ Simatic Manager در نمایش Online

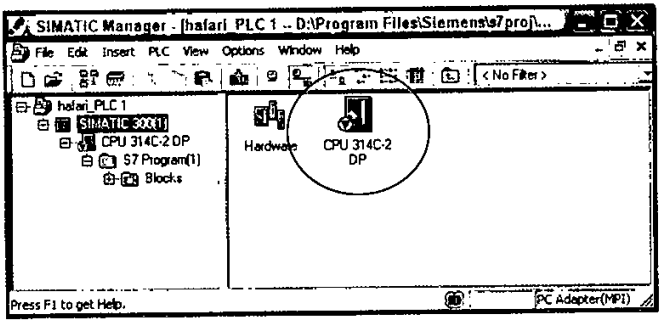
همانطور که اشاره شد، در حالت Online می‌توان به یک ایستگاه در حال کار متصل شد و برخی از محتویات ایستگاه را مشاهده نمود. در صورتیکه از محیط نرم‌افزار Simatic Manager به یک ایستگاه به صورت Online متصل شویم موارد زیر قابل مشاهده می‌باشد:

- وضعیت کار CPU، با کلیک بر روی Station مورد نظر
- بلاک‌های موجود در CPU، با کلیک برو روی پوشه Blocks

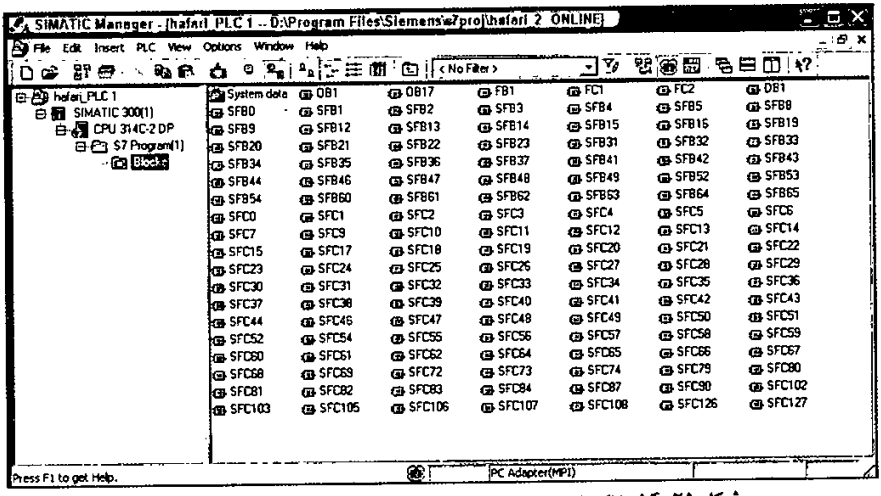
برای اجرای نمایش Online می‌توان از منوی View گزینه Open Online را علامت زد.



شکل ۲۵-۱۲ وضعیت CPU در حالت Run در نمایش Online در محیط Simatic Manager



شکل ۲۵-۱۳ وضعیت CPU در حالت Stop در نمایش Online در محیط Simatic Manager



شکل ۲۵-۱۴ بلاک‌های موجود در CPU در نمایش Online در محیط Simatic Manager

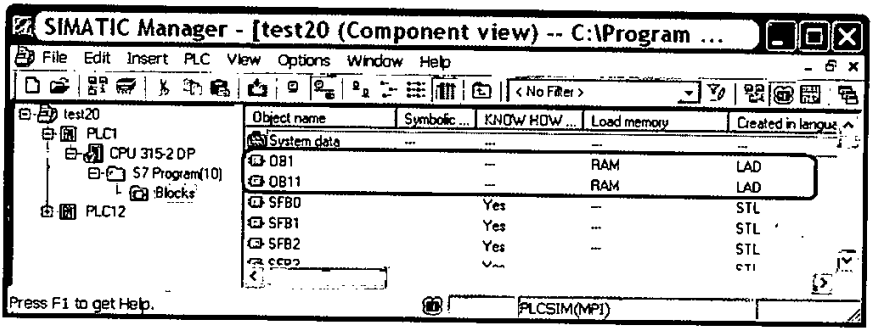
فصل ۲۵

نکته۱: در حالت Offline آنچه در پوشه بلاک دیده می‌شود، بلاک‌های موجود در کامپیوتر است ولی در حالت Online بلاک‌های موجود در PLC دیده می‌شود. بنابراین به منظور اطلاع از وجود یا عدم وجود یک بلاک در PLC، بایستی از نمایش Online استفاده شود.

نکته۲: در حالت Online می‌توان بلاکی را از محیط Offline وارد این محیط نمود. این کار مشابه دانلود است. می‌توان برعکس، بلاکی را از محیط Online کپی کرد و به محیط Offline برد. این کار مشابه آپلود است.

نکته۳: اگر در حالت Online بلاکی حذف شود، ممکن است منجر به توقف CPU گردد.

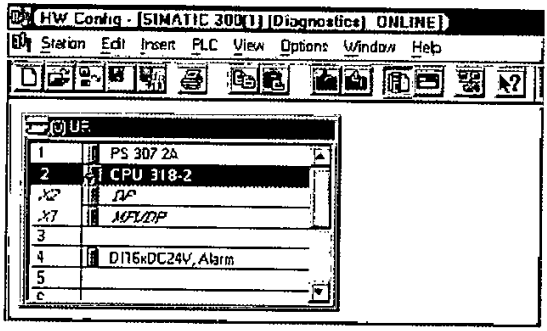
نکته۴: در حالت Online اگر از منوی View گزینه Detail را انتخاب کنیم جزئیات بیشتری را در مورد بلاک‌ها خواهیم دید، ازجمله اینکه محل ذخیره سازی آنها در RAM است یا EPROM.



شکل ۲۵-۱۵ نمایش محل ذخیره‌سازی بلاک‌ها در حالت online

۲۵-۴ محیط پیکربندی سخت‌افزار در نمایش Online

در صورتی‌که در محیط نرم‌افزار HW Config از منوی Station گزینه Open Online را انتخاب نمایید، وضعیت اجزای موجود در رک به‌صورت Online نمایش داده می‌شود. در این حالت وضعیت CPU، کارت‌های مختلف و کارت‌های موجود در شبکه با درج علامتی بر روی آن نشان داده می‌شود.



شکل ۲۵-۱۶ محیط HW Config در حالت Online

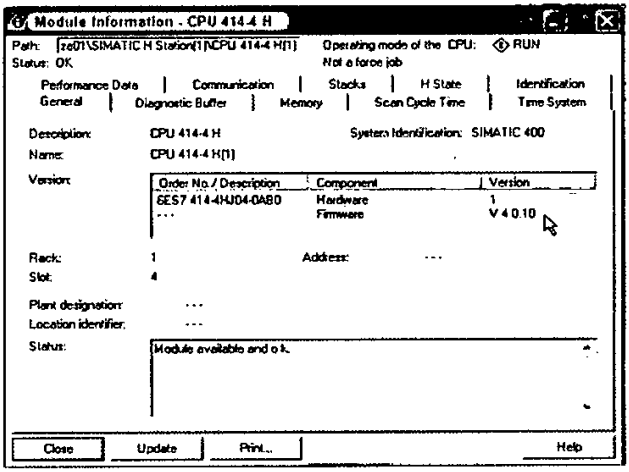
این حالت در عیب‌یابی سخت‌افزار بسیار کاربرد داشته و توسط آن برخی از خطاهای پیش‌آمده برای سخت‌افزار قابل شناسایی خواهد بود. شکل ۲۵-۱۷ این علائم را برای CPU به‌طور مختصر نشان می‌دهد، البته باید توجه داشت که علائم موجود بیش از موارد نشان‌داده شده در شکل می‌باشند که در کتاب پیشرفته در فصل عیب یابی تشریح می‌گردند.

Meaning	Symbol
STARTUP	
STOP	
RUN	
HOLD	

شکل ۲۵-۱۷ علائم نشان‌دهنده وضعیت کار CPU در حالت Online

CPU Module Information

نمایش اطلاعات CPU در حالت Online یکی از ابزارهای بسیار مهم جهت عیب‌یابی می‌باشد. برای نمایش این اطلاعات می‌توان در حالت Online روی CPU موجود در رک دوبارکلیک نمود. در این حالت پنجره‌ای مانند پنجره نشان داده شده در شکل ۲۵-۱۸ باز می‌شود.



شکل ۲۵-۱۸ سربرگ General در پنجره Module Information

جدول ۲۵-۱ بخش‌های مختلف پنجره فوق را به اختصار توضیح می‌دهد.

جدول ۲۵-۱ توضیحات بخش‌های مختلف پنجره CPU Module Information

بخش	کاربرد
General	مشاهده مشخصات کلی CPU
Diagnostic Buffer	برای شناسایی علت توقف CPU و ارزیابی ماژول های مرتبط با این فالت
Memory	برای فشرده سازی حافظه
Scan Cycle Time	مشاهده زمان سیکل اسکن واقعی CPU
Time System	برای نمایش و تنظیم کردن زمان و تاریخ مدول و چک سنکرون بودن زمان
Performance Data	معرفی مشخصات عملکردی CPU مانند سایز Load Memory یا PII و...
Communication	اینکه چه تعداد ارتباط با CPU امکان‌پذیر است و چه تعداد هم اکنون اشغال شده است.
Stacks	برای تعیین علت توقف CPU و عیب‌یابی با استفاده از اطلاعات Istack و Bstack

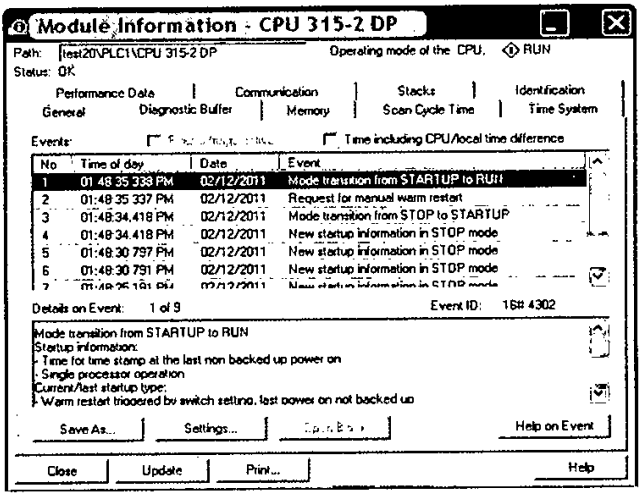
اطلاعات سربرگ‌های این پنجره برای شناسایی رفتار CPU و علت بروز فالت روی آن بسیار مهم است.
این پنجره در محیط‌های مختلف قابل دسترسی است و کلید میانبر آن Ctrl+D می باشد. در کتاب سطح پیشرفته در بحث عیب‌یابی PLC امکانات این پنجره به تفصیل شرح داده خواهد شد. در اینجا فقط به چند سربرگ آن به اختصار اشاره می‌کنیم.

سربرگ General

در این قسمت همانطور که در شکل ۲۵-۱۸ دیده می‌شود، مشخصات کلی CPU از جمله ورژن Firmware نمایش داده می‌شود. همچنین شماره رک و اسلاتی که CPU در آن قرار گرفته را نیز مشخص می‌کند.

سربرگ Diagnostics Buffer

در این قسمت تمام Eventهایی که برای CPU رخ می دهد با ذکر تاریخ و زمان دقیق نمایش داده می‌شود. همیشه اتفاقات جدید در بالا ظاهر می‌شوند و اتفاقات قبلی به پایین شیفت داده می‌شوند. در این قسمت زمان وصل یا قطع تغذیه، ریست شدن CPU و فالت‌ها مشخص است. این پنجره یکی از ابزارهای مهم برای عیب یابی است.



شکل۲۵-۱۹ سربرگ Diagnostic Buffer

سربرگ Memory

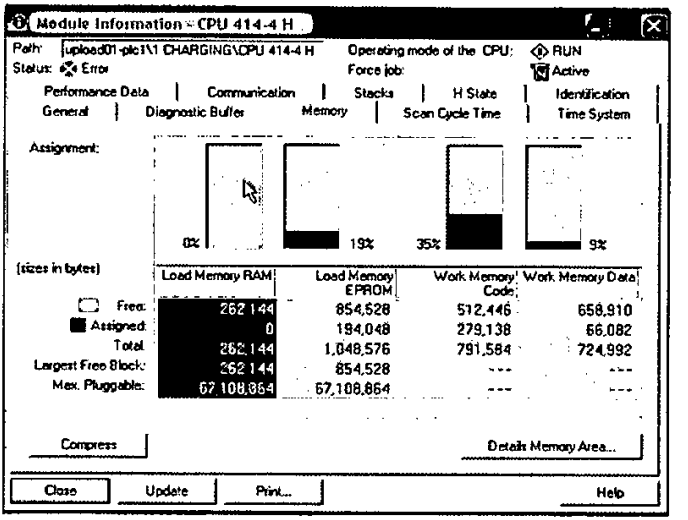
در این قسمت حافظه Load Memory و Work Memory نشان داده می‌شود و مشخص می‌کند که حافظه تا چه حدی پر شده و نیز مشخص می کند که حافظه آیا قابل افزایش است یا خیر و حد ماکزیمم برای افزایش آن چقدر است. درشکل ۲۵-۲۰ اولین باکس سمت چپ نشان می‌دهد که حافظه RAM مربوط به Load Memory به اندازه 256 KB فضا دارد که فعلاً خالی است و با کارت حافظه RAM می‌تواند تا 64 MB افزایش یابد.

باکس دوم نشان می‌دهد که CPU دارای یک کارت حافظه Flash EPROM با ظرفیت 1 MB است که فعلاً تا 19% پر شده است. این کارت می‌تواند تا 64MB افزایش یابد.

وقتی در Simatic Manager با استفاده از گزینه Download user program to memory card برنامه را به کارت فلش می‌فرستیم، مشاهده می‌کنیم که مانند شکل ۲۵-۲۰ برنامه از RAM حذف شده و در EPROM نشسته است.

درصورتی‌که حافظه تا نزدیک ماکزیمم پر شود، با کلیک بر روی Compress می‌توان وضعیت آنرا کمی بهبود بخشید.

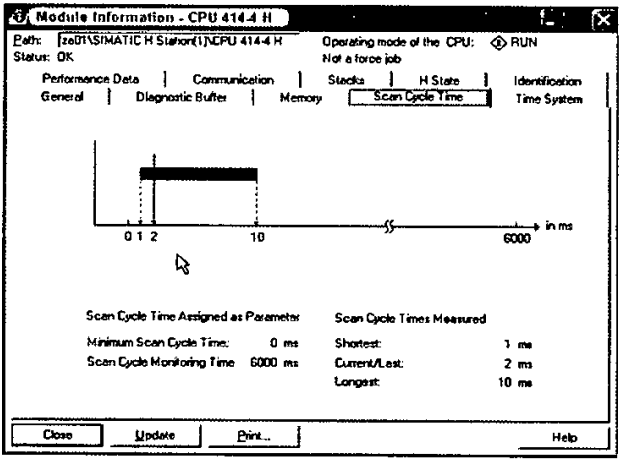
دو باکس سمت راست وضعیت Work Memory را نشان می‌دهد که بخشی از آن برای دیتا و بخشی برای برنامه استفاده شده است. همانطور که دیده می شود این دو بخش قابل افزایش نیستند.



شکل۲۵-۲۰ سربرگ Memory

سربرگ Scan Cycle

در این قسمت زمان سیکل اسکن نشان داده می شود. در شکل ۲۵-۲۱ این زمان برای یک CPU414-4H نشان داده شده است. همانطور که فلش عمودی نشان می‌دهد در این لحظه سیکل اسکن 2 ms است و تاکنون ماکزیمم سیکل 10 ms بوده است. در ضمن محور افقی نشان می‌دهد که ماکزیمم زمان مجاز برای سیکل اسکن 6000 ms است.

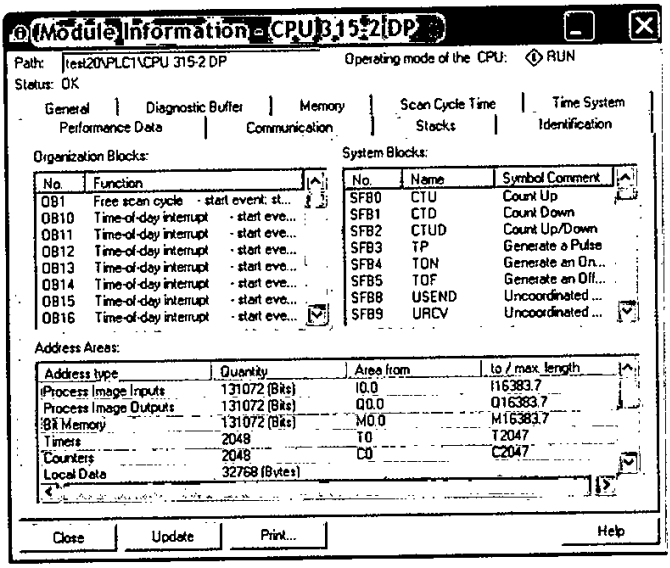


شکل۲۵-۲۱ سربرگ Scan Cycle

سربرگ Performance Data

در این قسمت می‌توان دید که CPU چه تعداد ورودی و خروجی و تایمر و کانتر و متغیر حافظه را ساپورت می‌کند. این اطلاعات در قسمت پایین پنجره نشان داده شده است.

در قسمت بالا می‌توان OBها و SFC/SFBهای مجازی که قابل استفاده هستند را مشاهده نمود. این بخش در کتاب سطح پیشرفته می‌گردد.

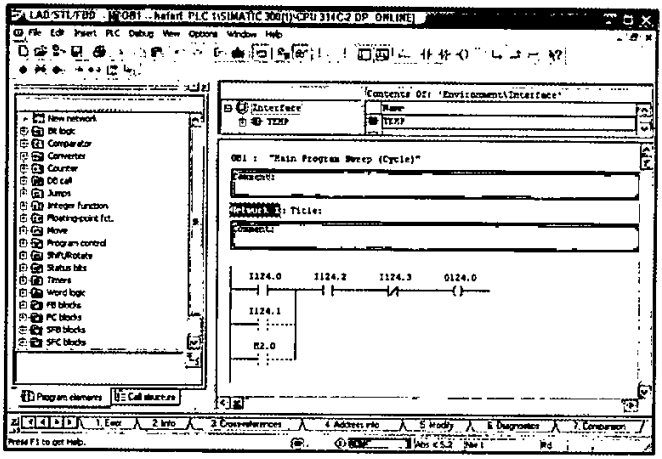


شکل۲۵-۲۲ سربرگ Performance Data

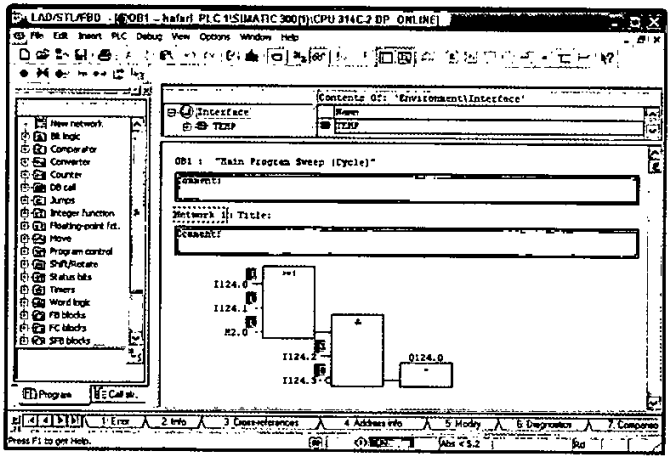
۲۵-۵ محیط برنامه‌نویسی در نمایش Online

در محیط LAD/STL/FBD نیز می‌توان به‌صورت Online برنامه را دید. برای این کار می‌توان روی ابزار مانیتور که در Toolbars موجود است کلیک نمود یا از منوی File گزینه Open Online را انتخاب نمود. در این حالت اگر زبان برنامه روی حالت LAD یا FBD تنظیم شده باشد سیگنال منطقی RLO با رنگ سبز نمایش داده شده و می‌توان وضعیت آدرس‌های موجود در برنامه را مشاهده نمود.

فصل ۲۵

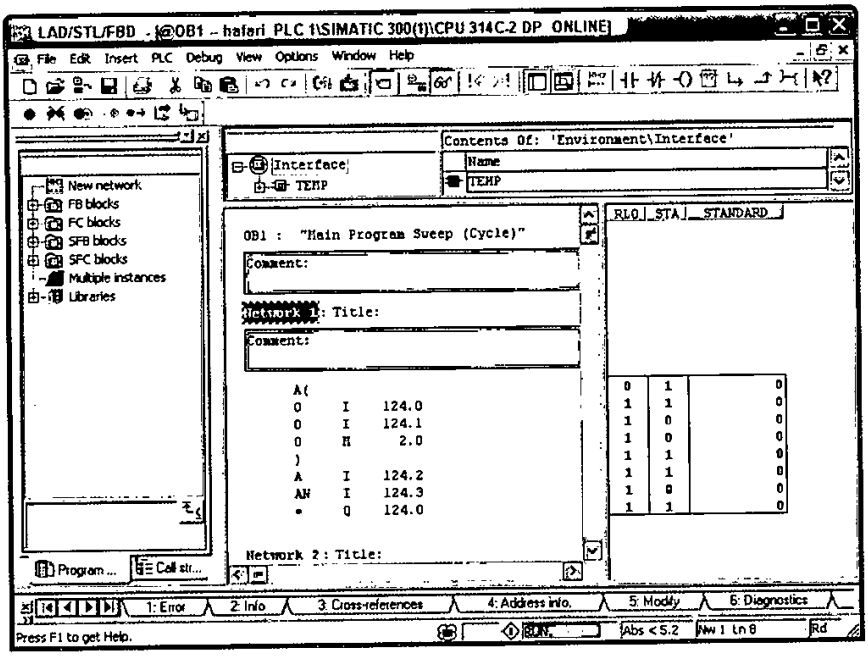


شکل۲۵-۲۳ نمایش محیط LAD/STL/FBD با انتخاب زبان LAD در نمایش Online



شکل ۲۵-۲۴ نمایش محیط LAD/STL/FBD با انتخاب زبان FBD در نمایش Online

باید توجه نمود که اگر زبان STL انتخاب شده باشد محیط LAD/STL/FBD به‌صورت نشان داده شده در شکل ۲۵-۲۵ می‌باشد. در این محیط می‌توان وضعیت سیگنال‌های مختلف را چک نمود. سایر موارد مرتبط با این موضوع در کتاب سطح تکمیلی ارائه می‌گردد.



شکل ۲۵-۲۵ نمایش محیط LAD/STL/FBD با انتخاب زبان STL در نمایش Online

۲۵-۶ پرسش و تحقیق

در محیط LAD/STL/FBD تفاوت مانیتور با منوی File>open Online چیست و آیا اگر در حالت Online برنامه را تغییر داده ذخیره کنیم، در CPU ذخیره می‌شود یا روی هارد کامپیوتر؟

فصل ۲۵

پیوست

آشنایی با استاندارد IEC61131

این بخش با نگاهی گذرا به استاندارد IEC1131 که استاندارد خاص PLCهاست، بخش‌های مختلف آنرا به اجمال معرفی کرده است.

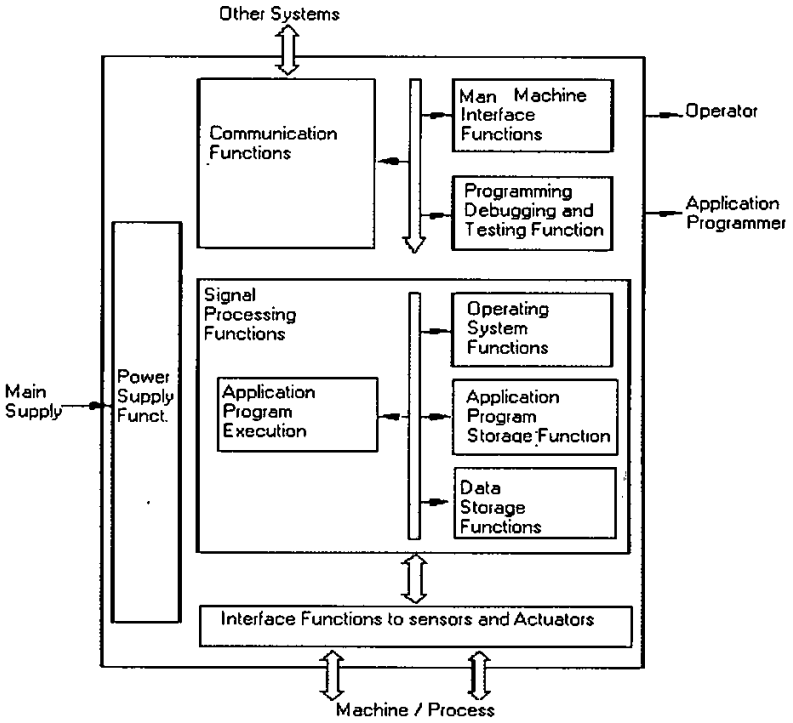
استاندارد IEC61131 که گاهی IEC1131 نیز خوانده می‌شود اگر چه اولین استاندارد مدونی است که برای PLCها عرضه شده است ولی در واقع تکمیل کننده بخش‌های استانداردهای مختلفی است که قبل از آن ارائه شده بود. استانداردهایی مانند:

- IEC559
- IEC 617-12 , 13
- IEC 848
- ISO/IEC 646
- ISO 8601
- ISO7185
- ISO7498

در سال ۱۹۷۹ یک گروه متخصص در IEC (International Elctro-technical Commission) کار بررسی جامع PLCها را به‌عهده گرفت. هدف این گروه تدوین روش‌های استانداردی بود که بتواند شامل سخت‌افزار، برنامه‌نویسی و ارتباطات PLC را پوشش دهد و توسط سازندگان PLC به‌کار گرفته شود. این کار حدود ۱۲ سال به‌طول انجامید و سرانجام پس از بحث‌های موافق و مخالفی که انجام شد استاندارد IEC1131 شکل گرفت. در سال ۱۹۹۰ بخش های ۱ و۲ و در سال ۱۹۹۳ بخش ۳ و در سال ۱۹۹۵ بخش ۴ این استاندارد تدوین گردید. در سال ۲۰۰۱ بخش ۷ با عنوان کنترل فازی به آن اضافه شد.

بخش اول IEC 61131-1 با عنوان General Information

این بخش به تعاریف کلی که PLC را از سیستم‌های کنترل متمایز می‌سازد می‌پردازد. رفتار کلی آن که سیکل اسکن و ذخیره‌سازی Image ورودی‌ها و خروجی‌ها را بیان می‌کند، ویژگی‌ها و عملکرد هر قسمت از PLC مانند CPU، منبع تغذیه، ورودی‌ها و خروجی‌ها و ارتباطات شبکه را تشریح کرده و یک ساختار کلی را به‌عنوان الگو مطابق شکل پ ۱-۱ ارائه می‌نماید. وسایل جانبی مانند وسایل برنامه‌ریزی، تجهیزات HMI و ... نیز در این الگو مورد توجه قرار گرفته‌اند.



شکل پ۱ -۱ مدل PLC

بخش دوم IEC 61131-2 با عنوان Equipment Requirements and Tests

این بخش ضمن بیان مشخصات الکتریکی و مکانیکی و حدود عملکردی PLC، شرایط محیطی (مانند دما، رطوبت)، حداقل ملزومات برای ساخت، سرویس، انبار کردن، حمل و نقل، عملکرد و ایمنیPLCها و وسایل جنبی آنها را تشریح کرده و تست‌های کاربردی مربوطه را توضیح می‌دهد.

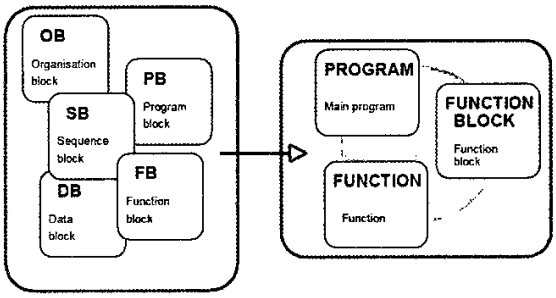
به‌طور کلی مباحث اصلی این بخش به سه دسته زیر تقسیم می‌شود:

- تشریح ملزومات منبع تغذیه، باتری، CPU، ورودی و خروجی‌های دیجیتال و آنالوگ، ارتباطات شبکه، Remote I/Oها وسایل جنبی متصل به PLC، نصب و زمین.
- تشریح تست‌های محیطی، مکانیکی و الکتریکی،تست تاثیر انواع نویزها، تست‌های ایمنی
- ملزومات ایمنی در ارتباط با شوک‌های الکتریکی، آتش‌سوزی، ماکزیمم و مینیمم ولتاژ، ماکزیمم و مینیمم دما و سایر موارد مربوط به Safety

بخش ۳ - زبان‌های برنامه‌نویسی

این بخش یکی از مهمترین بخش‌های استاندارد محسوب می‌شود زیرا به نحوه برنامه‌نویسی، انواع دیتاها و بلاک‌های کنترلی پرداخته و با ذکر مثال آنها را تشریح کرده است.

انواع دیتاهایی که می‌توانند در برنامه‌نویسی استفاده شوند مانندBool ، Byte ، Date , Word ,Real, Integer Time تعریف شده‌اند. همچنین POUها یعنی (Program Organization Units) مانند فانکشن (FC) و فانکشن بلاک (FB) مشخص گردیده‌اند. وجه تمایز FB از FC اینگونه تعیین شده که FB علاوه بر الگوریتم برنامه، دیتاها را نیز شامل می‌شود.

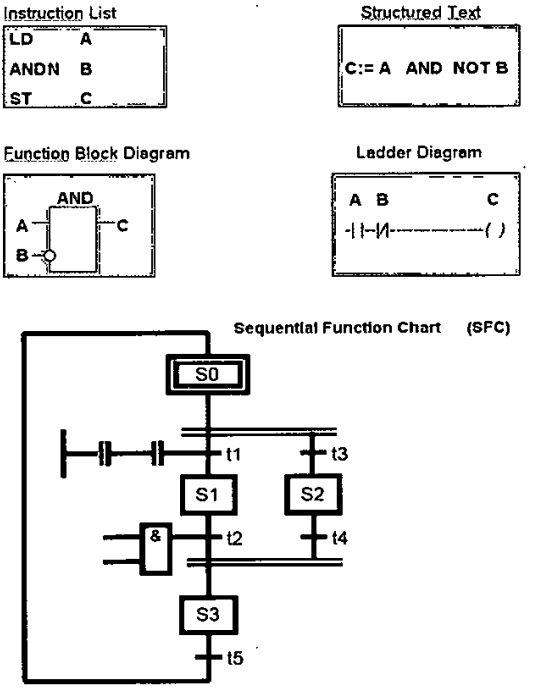


شکل پ۱ -۲ بلاک‌های برنامه‌نویسی قبل و بعد از IEC1131

IEC در این بخش چهار زبان برنامه‌نویسی که قبلاً نیز به‌کار می‌رفت را انتخاب کرده و یک زبان جدید نیز بر آن افزوده و در مجموع پنج زبان برنامه نویسی PLCها را به‌عنوان استاندارد ارائه نموده است. این زبان‌ها عبارتند از:

- IL یا Instruction List یک زبان سطح پایین و از زبان های قبلی PLC است که بصورت متنی می‌باشد. این زبان بیشتر شبیه زبان اسمبلی میکروپروسسور است.
- FBD یا Function Block Diagram زبان گرافیکی است که قبلاً نیز مورد استفاده قرار می‌گرفت. در FBD برنامه‌نویسی توسط یک سری بلوک‌های پایه که در کنار هم قرارمی‌گیرند انجام می‌شود.
- LD یا Ladder Diagram روش گرافیکی است که قبلاً نیز استفاده می‌شد ولی به‌صورت پیشرفته‌تر عرضه شده است. در روش جدید LD و FBD می‌توانند به‌صورت توام در برنامه به‌کار روند.
- ST یا Structured Text زبان جدیدی است که IEC به ۴ زبان قبلی افزوده است. ST یک زبان سطح بالا شبیه C و پاسکال است و کاربردی عالی به‌ویژه در الگوریتم‌های پیچیده ریاضی را داراست.
- SFC یا Sequential Function Control نیز روش گرافیکی جدیدی است. در این روش برنامه به مراحلی که ترتیب الگوریتم های کنترلی را نشان میدهد تقسیم می‌گردد و شامل Stepهای مختلف برنامه است. هرگاه شرایطی که در بخش Transition مشخص شده برآورده گردید Step قبلی غیر فعال و Step بعدی فعال می‌گردد.

در شکل، چهار زبان به‌صورت ساده مورد مقایسه قرار گرفته‌اند. اینکه چه زبانی انتخاب شود بستگی به ساختار سیستم کنترل و نیز تاحدودی بستگی به سلیقه استفاده کننده دارد.



شکل پ۱ -۳ روش‌های برنامه‌نویسی استاندارد

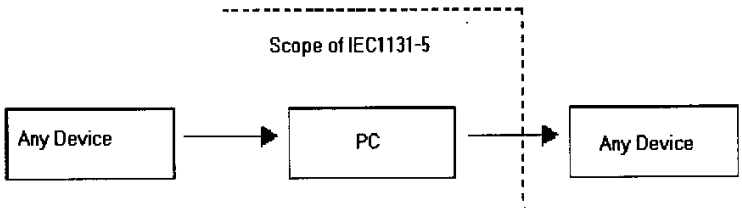
بخش ۴ - راهنمای کاربران[1]

بخش چهارم راهنمای کاربر نهایی (End User) است. در این بخش اطلاعاتی برای نحوه انتخاب تجهیزات مناسب، آنالیز سیستم و نگهداری آن به کاربر ارائه می شود.

بخش ۵ - ارتباطات[2]

در این بخش مدل ارتباطی PC برای ارتباط شبکه، فانکشن بلاک‌های ارتباطی و نکات مربوط به Data Acquizition بیان شده است و مدل ارتباطی زیر ارائه شده است:

1. User Guidelines
2. Communications



شکل پ۱ -۴ مدل ارتباطی PLC

بخش ۶- این بخش برای استفاده در آینده رزرو شده است.

بخش ۷- برنامه نویسی کنترل فازی[1]

این بخش برنامه‌نویسی کنترل فازی را معرفی می‌نماید. چگونگی فازی‌سازی، استنتاج فازی با استفاده از قواعد فازی و نحوه نافازی سازی در این بخش مورد بحث قرار گرفته و با ذکر مثال تشریح شده است.

بخش ۸ – راهنمای کاربرد زبانهای برنامه نویسی[2]

در بخش ۸ نکات کاربردی مربوط به زبان‌های برنامه‌نویسی بیان شده است. نحوه به‌کارگیری فانکشن بلاک، تفاوت فانکشن و فانکشن بلاک، شبیه‌سازی و عیب‌یابی برنامه و ... تشریح شده‌اند.

1. Fuzzy Control Programming
2. Guidelines for the applications of programming languages

منابع و مراجع

- Automating Manufacturing Systems with PLCs — Hugh Jack
- PLC Programming Methods and Applications — John R. Hackworth and Frederick D. Hackworth, Jr
- Programmable Controllers Theory and Implementation — L.A. Bryan, E.A. Bryan
- Introduction to PLC controllers — Nebojsa Matic
- Programmable Logic Controllers — W. Bolton
- Step7 in 7 Steps — C.T. Jones
- Siemens Manuals — SIEMENS
- Sitrain Programming Course 1 — SIEMENS
- Automating with SIMATIC — Hans Berger
- Practical Data Acquisition for Instrumentation and Control Systems — John Park , Steve Mackay
- Measurement and Instrumentation Principles — To Jane, Nicola and Julia
- INSTRUMENT ENGINEERS' HANDBOOK — Bela G. Liptak
- Sensors and Transducers — Ian R. Sinclair

- راهنمای جامع Step7 جلد اول — محمدرضا ماهر
- تکنولوژی فیلدباس — محمدرضا ماهر – احمد حیدریان
- پروژه های کاربردی با PLC S7 — احمد فرجی
- سیستم های کنترل خطی — علی خاکی صدیق
- مقدمه ای بر اتوماسیون صنعتی — علیرضا فاتحی
- کنترل صنعتی — محمود تارخ
- اصول ابزار دقیق صنعتی به زبان ساده — سیف الله پوربختیاری

مجموعه پروژه‌های کاربردی با

PLC SIEMENS

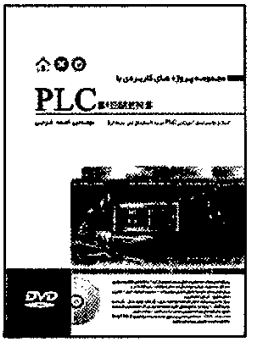
(چاپ دوم)

تألیف مهندس احمد فرجی

این کتاب مجموعه‌ای از مثال‌ها و پروژه‌های صنعتی می‌باشد که به‌منظور اجرای آنها از PLC های S7-300 و S7-400 زیمنس استفاده شده و برنامه‌نویسی آن توسط نرم‌افزار Simatic Manager انجام گرفته است.

این کتاب در پنج فصل تدوین شده است. فصل اول مجموعه پروژه‌های مقدماتی می‌باشد که در آن مثال‌های نسبتاً ساده به‌منظور آشنایی کاربر با محیط نرم‌افزار Simatic Manager و دستورات Bit Logic و تایمر بیان شده است. این فصل علاوه بر آنکه برای دانشجویان مفید است، با سرفصل آموزشی سازمان فنی و حرفه‌ای نیز مطابقت داشته و می‌تواند برای شرکت‌کنندگان در دوره‌ها و آمزمون‌های این مراکز نیز مفید باشد. در فصل دوم برنامه‌هایی در سطح متوسط بیان شده است که در آن از دستورات کانتر و محاسبات ریاضی نیز استفاده شده است. در فصل سوم تعدادی پروژه کامل صنعتی همراه با توضیحات و شرح عملکرد ارائه شده است. این فصل نیز قابل استفاده برای دانشجویان در دوره‌های مختلف آموزشی می‌باشد. فصل چهارم شامل یک پروژه عملی و وسیع در یک کارخانه تولید قطعات فلزی است که دانشجویان در این مثال، علاوه بر اینکه با برنامه‌نویسی پیشرفته آشنا می‌شوند، با ادوات مختلف صنعتی که در کارخانجات مورد استفاده قرار می‌گیرد نیز آشنا خواهند شد. فصل پنجم مثال‌هایی از نحوه‌ی به‌کارگیری ترفندها و نکات برنامه‌نویسی می‌باشد که خواننده را با نکات و تکنیک‌های برنامه‌نویسی آشنا می‌کند.

نحوه‌ی نگارش این کتاب به‌گونه‌ای است که خواننده بتواند ضمن انجام مثال‌های مختلف با نحوه‌ی تعریف پروژه و ادواتْ به‌کار رفته در صنعت نیز آشنا شده و مهارت کافی جهت انجام پروژه‌های صنعتی را به‌دست آورد



محتویات DVD شامل: نسخه دائمی و بدون محدودیت نرم‌افزار Step7 V5.4 ، با قابلیت تست کلیه‌ی پروژه‌ای کتاب

۳۱۲ صفحه، ۷۰۰۰ تومان

NEGARANDE
DANESH

ناشر کتاب‌های دانشگاهی وکامپیوتر

کاملترین مجموعه کتابهای کاربردی به همراه DVD های آموزشی

انتشارات نگارنده دانش

تلفن : ۶۶۴۲۵۵۹۵-۶۶۹۴۰۴۲۹

تهران - صندوق پستی : ۱۳۱۴۵-۹۱۶

weblog: negarandedanesh.blogfa.com

شابک : ۹۷۸-۶۰۰-۶۱۹۰-۰۱-۳