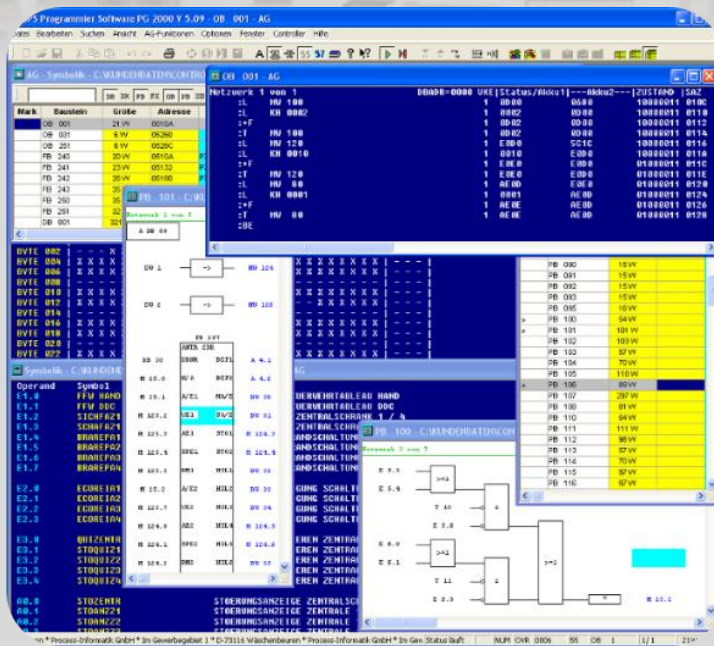


مرجع کاربردی PLC

SIMATIC STEP 5



مؤلف: مهندس اکبر اویسی فر

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

راهنمای جامع PLC

SIEMENS

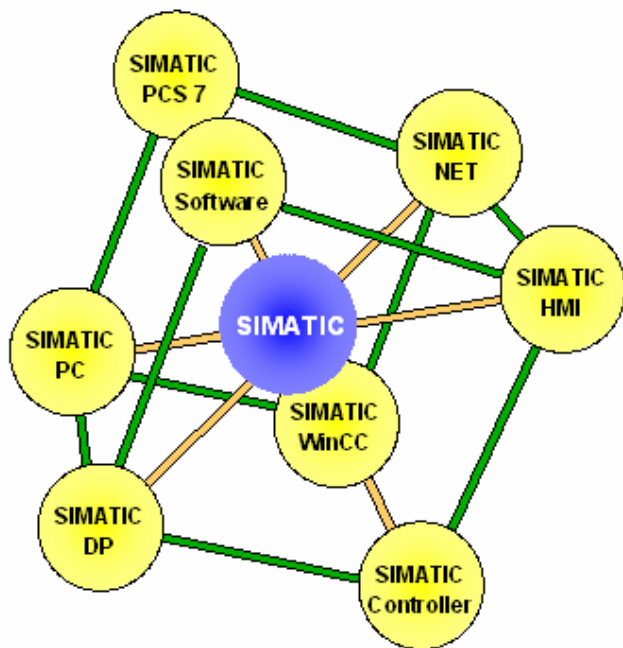
SIMATIC STEP 5

CD همراه شامل:

* نرم افزار STEP 5

* آموزش نرم افزار S5

* کتابچه های لاتین شرکت زیمنس



اکبر اویسی فر

کلیه حقوق قانونی برای ناشر محفوظ است. هرگونه چاپ و یا تکثیر از مطالب این کتاب بدون اجازه ناشر ممنوع می باشد. متخلفان به موجب قانون حقوق مولفان تحت پیگرد قانونی قرار می گیرند.

خدای مهربان را سپاس که به اینجانب توانایی بخشید تا بار دیگر بتوانم به جامعه صنعتی کشور به ویژه مهندسين و دانشجويان خدمت ناچيزی را ارائه دهم.

همانطور که می دانيد امروزه تقريباً در تمامی کارخانجات صنعتی به نوعی از سيستمهای PLC استفاده شده است که دليل اين امر پيچيدگی و گستردگی صنايع امروزی می باشد. کنترل کننده های برنامه پذير با توجه به نیاز صنايع پيشرفته ابداع گردیده اند. در اين کنترل کننده ها بيشتر عملکرد سيستم به صورت نرم افزاری تعيين می شود. در اين میان شرکتهای سازنده بسياری در جهان وجود دارند که PLC های مورد نیاز کاربران را طراحی و در اختيار آنها قرار می دهند. شرکت زیمنس آلمان از جمله اين شرکتهاست که محصولات PLC آنها از تنوع خاصی برخوردار بوده و در کشور ما نیز بسيار مورد استقبال قرار گرفته است. البته شرکتهای ديگری نیز همچون Festo , ABB , Omron , Mitsubishi در جهت ساخت PLC به رقابت پرداخته اند.

مجموعه ای که پيش رو داريد آموزش برنامه نویسی و آشنایی با سخت افزار PLC های سری S5 شرکت زیمنس می باشد که هنوز بعد از گذشت چندین سال از توليد آنها در مراکز صنعتی و پروسه های کنترلی مورد استفاده قرار می گيرد.

مطالب اين کتاب به گونه ای عنوان شده است که برای کاربران مبتدی تا پيشرفته قابل استفاده باشد. به همین دليل اين کتاب می تواند به عنوان راهنمایی مناسب برای مهندسين و تکنسین ها در محیطهای صنعتی باشد. همچون گذشته در انتظار دريافت نقطه نظرات همه خوانندگان محترم در مورد اين مجموعه و کمی و کاستی های آن هستم.

تمامی مهندسين در سراسر کشور می توانند مطالب و نرم افزارهای مورد نیاز خود را به آدرس پست الکترونیکی Akb_Oveisifar@yahoo.Com یا شماره 09361135347 ارسال تا در اسرع وقت مطالب مورد نیاز خدمتتان ارسال گردد. امیدوارم که اين کتاب بتواند برای تمامی مهندسين ، تکنسین ها و دانشجويان کشور مفيد واقع گردد.

اکبر اویسی فر

تابستان ۸۳

فصل اول - سخت افزار PLC

5

مشمول بر

* تاریخچه PLC

* آشنایی با کاربردهای PLC در صنعت

* آشنایی با سخت افزار PLC S5



مقدمه

PLC ها تاریخچه کوتاهی دارند و از تولد اولین آنها عمر چندانی نمی گذرد. اولین PLC در دهه ۷۰ برای استفاده در صنعت اتومبیل سازی طراحی شد. نخستین بار کنترلرهای MODICON در سال ۱۹۶۹ به صنعت معرفی شدند که با هدف جایگزینی رله های الکتریکی استفاده می شد. اما استفاده از PLC با توجه به مزایایی که داشت به تدریج رایج شد و سازندگان متعددی نیز در این رشته پدیدار شدند و در صنایع مختلف به کار گرفته شدند.

PLC به صورت کاملاً صنعتی و قابل اطمینان جهت استفاده در محیطهای صنعتی با شرایط محیطی دارای ارتعاش، نویز، حرارت و رطوبت ساخته شده است. در کشور ما نیز قبل از روی کار آمدن سیستمهای اتوماسیون، وظیفه کنترل پروسه های صنعتی بر عهده مدارات فرمان بود که استفاده از این مدارات مشکلات زیادی را به همراه داشت. با پیدایش تکنولوژی اتوماسیون صنعتی، در کشور ما نیز اغلب صنایع بزرگ پتروشیمی، گاز، نفت و نیروگاه ها مجهز به این تکنولوژی گردیده است. در حال حاضر شرکت های زیادی در اکثر کشورهای توسعه یافته تولید کننده PLC و قطعات مربوطه هستند. در ادامه به برخی از کمپانی های مطرح اشاره شده است:

SIEMENS
Modicon
Allen Bradley
ABB
GE Fanuc
Omron
Mitsubishi
LS
Contronic

در شکل زیر چند نمونه از PLC برندهای مختلف را ملاحظه می کنید.

SIEMENS



OMRON



7

LS



Mitsubishi



Allen Bradley



8

Modicon



GE Fanuc



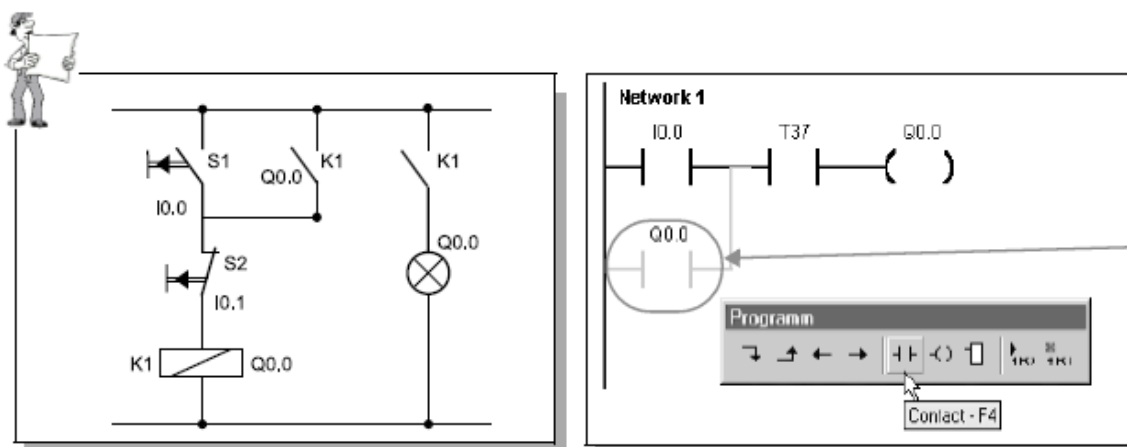
مفهوم کنترل کننده قابل برنامه ریزی PLC

در سیستمهای اتوماسیون وظیفه اصلی کنترل بر عهده PLC می باشد که با گرفتن اطلاعات از ورودی، وضعیت ماشین را حس نموده و نسبت به آن عکس العمل مناسب را نشان می دهد. امکان تعریف مدهای مختلف برای ترمینالهای ورودی و خروجی PLC، این امکان را فراهم کرده تا بتوان PLC را مستقیماً به المانهای دیگر وصل کرد.

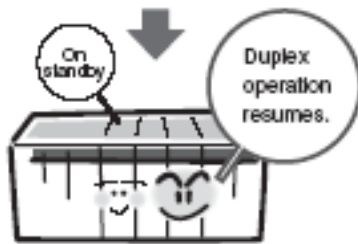
وظیفه PLC قبلاً بر عهده مدارات فرمان رله ای بود که استفاده از آنها در صنعت امروز تقریباً منسوخ و قدیمی شده است. اولین اشکالی که در این مدارها ظاهر می شود آن است که با افزایش تعداد رله ها حجم و وزن مدار فرمان بسیار بزرگ شده، همچنین موجب افزایش قیمت آن می گردد. برای رفع این اشکال مدارهای فرمان الکترونیکی ساخته شد، ولی با وجود این هنگامی که تغییری در روند یا عملکرد ماشین صورت می گیرد لازم است تغییرات بسیاری در سخت افزار سیستم کنترل داده شود.

با استفاده از PLC تغییر در روند یا عملکرد ماشین به آسانی صورت می پذیرد، زیرا دیگر لازم نیست سیم کشی ها و سخت افزار سیستم کنترل تغییر کند و تنها کافی است چند سطر برنامه نوشت و به PLC ارسال کرد تا کنترل مورد نظر تحقق یابد.

کسانی که با مدارات فرمان آشنایی دارند به خوبی می دانند که در طراحی تابلوهای فرمان در صورت نیاز به تغییر در منطق کارکرد، باز و بسته کردن سیم ها و جابه جایی قطعات الکتریکی امکان ناپذیر است. در صورتی که این عمل در PLC تنها با تغییر برنامه در نرم افزار صورت می گیرد.



معایب مدارت فرمان رله ای



- افزایش حجم تابلو
- هزینه بالا جهت راه اندازی
- استهلاک مکانیکی
- نویز پذیری زیاد
- عیب یابی دشوار
- و....

در قسمت زیر، برخی از مزایای استفاده از PLC بیان شده است:

- ۱- کاهش هزینه‌ها: به دلیل حذف بسیاری از وسایل و قطعات صنعتی همچون رله‌ها، تایمرها، شمارنده‌ها و....
- ۲- عیب یابی آسان توسط ابزارهای مناسب
- ۳- کاربرد بصورت منفرد یا مجتمع در شبکه
- ۴- تغییر منطق کارکرد PLC در سریع ترین زمان
- ۵- سرعت و دقت بالا در کنترل پروسه‌های صنعتی
- ۶- اجرای آزمایشی قبل از ارسال برنامه به کنترلر
- ۷- دریافت و آشکار نمودن پیغام‌های خطا
- ۸- قابلیت اتصال به پنل‌های صنعتی و تجهیزات مانیتورینگ (HMI)
- ۹- تحت شبکه در آوردن فرآیندهای مختلف (کنترل از راه دور توسط Modem)
- ۱۰- برنامه نویسی بسیار ساده برای کاربران
- ۱۱- رنج وسیع مدل‌ها و ماژول‌های افزایشی
- ۱۲- قابلیت اتصال به رنج وسیعی از سیگنال‌های استاندارد دیجیتال و آنالوگ
- ۱۳- قابلیت اتصال به انواع درایوهای صنعتی و بردهای کنترلی

امروزه، استفاده از PLC محدود به صنایع خاصی نمی‌باشد و در هر صنایع و ماشین صنعتی امکان استفاده از آنها وجود دارد.

در حال حاضر ارتقاء سطح کیفی محصولات تولیدی در صنایع مختلف و در کنار آن افزایش کمی تولید، هدف اصلی هر واحد صنعتی می‌باشد و مدیران صنایع نیز به این مهم واقف بوده و تمام سعی خود را در جهت نیل به این هدف متمرکز نموده اند .

نمونه ای از کاربردهای PLC



- ✓ صنایع نفت و گاز
- ✓ صنایع خودرو سازی
- ✓ سیستم های حمل و نقل
- ✓ سیستمهای نگهداری از راه دور
- ✓ پالایشگاه ها

11

- ✓ کنترل ایستگاه های فرعی غیر متمرکز
- ✓ کوره های صنعتی
- ✓ دستگاه های بسته بندی
- ✓ ماشین آلات تزریق و پرس
- ✓ ماشین آلات چاپ و کاغذ
- ✓ ماشین آلات ریخته گری



اجزای سخت افزاری PLC

هر PLC همانند سایر سیستم های کامپیوتری از بخشهای اصلی واحد پردازش مرکزی (CPU)، ورودی (Input)، خروجی (Output) و حافظه (Memory) تشکیل شده است. PLC ها در یک تقسیم بندی کلی در گروه های زیر قرار می گیرند:

Compact

این PLC ها دارای ظاهری یکپارچه می باشند. منظور از کلمه یکپارچه این است که تمامی واحدها و یا قسمتهای بیان شده مانند ورودی، خروجی، CPU، در یک مجموعه قرار گرفته اند. این مدل از PLC ها، قابلیت توسعه ندارند و در کاربردهای کوچک مورد استفاده قرار می گیرند.



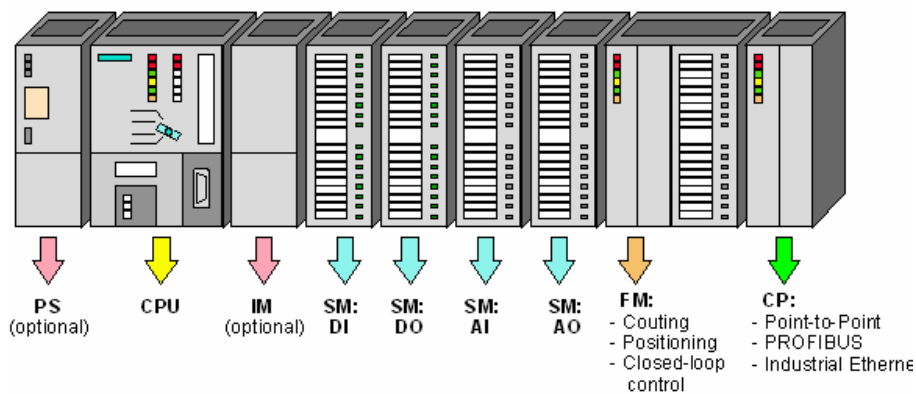
Compact/Modular

این گروه نیز دارای پایه کامپکت می باشد. تفاوت این گروه با گروه قبلی در این است که در این سری امکان افزایش ماژول های توسعه وجود دارد. در واقع در این گروه امکان ساپورت کردن تعداد I/O بیشتر وجود دارد که می تواند در کاربردهای کوچک و متوسط به کار برده شوند.



Modular

این سری از PLC ها در صنعت بسیار مورد استفاده قرار می گیرند. سرعت بالا، حجم زیاد حافظه و ساپورت تعداد بالای ورودی /خروجی از جمله ویژگی های این سری می باشد. لازم به ذکر است که تمامی واحدهای این سری بصورت کارت هایی مجزا می باشند که عمل ارتباط این کارت ها توسط کانکتور ارتباطی تعبیه شده در کنار کارت صورت می پذیرد. به عبارت دیگر در این گروه، ماژول ها توسط کانکتور به یکدیگر و با CPU ارتباط می گیرند. این گروه با توجه به ساپورت کردن تعداد زیاد I/O، می تواند در کاربردهای بزرگ استفاده شوند.



Rack

این سری از PLC های به عنوان پرقدرتترین و امن ترین کنترل کننده در صنعت، جهت کنترل پروسه های پیچیده که تعداد ورودی / خروجی و لوپ های کنترلی بسیار زیاد می باشد مورد استفاده قرار می گیرند. ماژول ها در این گروه در یک فضای زیرین همانند مادربرد کامپیوتر قرار می گیرند. این فضا را در اصطلاح رک می نامند. وجود فضای خالی در این گروه مشکلی در بحث ارتباط CPU با سایر ماژول ها ایجاد نمی کند.



14



در این کتاب PLC های سری S5 زیمنس آلمان مورد بررسی قرار خواهند گرفت. PLC های S5 را از نظر ساختار و کارایی می توان به سه گروه تقسیم نمود:

S5-90U ,S5-95U	Compact/Modular-۱
S5-100U , S5-115U	Modular-۲
S5-115U , S5-135U , S5-155U	Rack Base-۳

اجزای اصلی سخت افزاری PLC های سری S5 عبارتند از:

15

واحد پردازنده مرکزی CPU

CPU مخفف کلمات Central Processing Unit بوده و یکی از مهم ترین اجزای PLC محسوب می شود، زیرا مرکز محاسبات و کنترل PLC بوده و دستورالعمل ها به وسیله این واحد پردازش و اجرا می شود. به عبارت دیگر، این واحد را می توان مغز PLC دانست. به طور کلی، وظیفه واحد پردازنده مرکزی دریافت اطلاعات از ورودی، پردازش بر روی این اطلاعات مطابق برنامه موجود در حافظه و سپس، ارسال فرا مین به خروجی می باشد. قدرت و سرعت پردازش در یک کنترل کننده بستگی به سرعت پردازش CPU دارد. برای استفاده از CPU باید دستورالعمل های مناسبی به آن داده شود که این کار نیز با توجه به ساختار ماشین و پروسه-کنترلی، توسط کاربر وارد می شود. از سوی دیگر CPU باید اجزای سخت افزاری دیگر را در سیستم مانند حافظه، I/Oها و... را هدایت کند. برای انجام این کار، میان CPU و سایر اجزاء، گذرگاه هایی به نام Bus وجود دارد.

در ضمن، باید این نکته را نیز یادآور شد که CPU هوشمند نیست و خودش تصمیم نمی گیرد و فقط از مجموعه دستورالعمل هایی که توسط کاربر نوشته و در حافظه قرار می گیرد، تبعیت می کند.



ورودی دیجیتال DI

از این قسمت سیگنال‌های دیجیتال به PLC ارسال می‌شوند. این سیگنال‌ها می‌توانند توسط کلیدها، میکروسوییچ‌ها و سنسورها ارسال شوند. در اکثر PLC‌های بزرگ سطح ولتاژ این سیگنال‌ها باید 24 ولت DC باشد. CPU از اتفاقات محیط خارج از PLC توسط این سیگنال‌ها مطلع می‌شود. این اطلاعات بصورت دیجیتال و با سطوح استاندارد 0-24 VDC می‌باشد. وضعیت سیگنال‌های ورودی توسط LEDهایی که در جلوی این ماژول‌ها نصب شده‌اند، قابل رویت است. ورودی‌ها می‌توانند بر روی برد اصلی CPU و یا در قالب یک ماژول به PLC متصل شوند.



خروجی دیجیتال DO

از این قسمت سیگنال‌ها به محرک‌ها و مصرف کننده‌های پروژه ارسال می‌شوند. این فرمان‌ها اغلب به صورت سیگنال‌های استاندارد 24 VDC یا 220 VAC می‌باشد. این فرامین می‌تواند به وسایلی نظیر رله‌ها، کنتاکتورها، شیرها و لامپ‌های سیگنال منتقل شود.

درایورهای خروجی این ماژول‌ها می‌تواند به صورت رله‌ای یا ترانزیستوری باشد. در واقع، ماژول‌های DO معمولاً به دو صورت رله‌ای و ترانزیستوری در دسترس می‌باشند. در خروجی‌های رله‌ای امکان سوئیچ نمودن ولتاژ و جریان بالا وجود دارد، ولی این نوع خروجی برای کاربردهایی که نیاز به تعداد دفعات قطع و وصل زیاد می‌باشد، مناسب نمی‌باشد. در واقع، این نوع خروجی‌ها دارای استهلاک مکانیکی بوده و توانایی انجام سوئیچ در فرکانس‌های بالا را ندارد. خروجی‌های ترانزیستوری نیز نوع دیگری از خروجی‌های دیجیتال می‌باشند که قابلیت سوئیچینگ با فرکانس بالا در آنها وجود دارد.

نکته‌ای که در این نوع خروجی باید به آن توجه نمود، این است که در این نوع خروجی امکان سوئیچ نمودن ولتاژ و جریان بالا وجود ندارد. ماکزیمم جریان مجاز در خروجی‌های ترانزیستوری معمولاً در حد میلی آمپر (۵۰۰ میلی آمپر) و در خروجی‌های رله‌ای در حد چند آمپر می‌باشد. در ضمن، سطح ولتاژ نیز در خروجی‌های ترانزیستوری معمولاً 24 VDC و در خروجی‌های رله‌ای می‌تواند هم به صورت DC و هم به صورت (AC) 220VAC باشد.

ماژول های I/O



17

ورودی آنالوگ AI

در پروسه های صنعتی کمیت هایی مانند دما، وزن، فشار به عنوان کمیت های آنالوگ به حساب می آیند. برای اندازه گیری این کمیت ها از وسایلی مانند ترموکوپل، لودسل، ترمومتر و... استفاده می کنیم. کمیت های آنالوگ به صورت پیوسته می باشند و نمی توان توسط ورودی های دیجیتال آنها را اندازه گیری نمود. برای اندازه گیری این کمیت ها در صنعت از ورودی های آنالوگ استفاده می شود. این نوع ورودی ها در واقع، سطح سیگنال آنالوگ را به دیجیتال تبدیل مینمایند. سطوح استاندارد برای ورودی های آنالوگ عبارتند از:

0-20mA / 4-20mA / -2.5 _ +2.5 / 0-5V / 0-10V

کارت های ورودی آنالوگ دارای سخت افزار لازم برای تبدیل سیگنال آنالوگ ولتاژ و جریان به مقدار عددی می باشند. در این کارت ها مبدل A/D بکار رفته است. ماژول های آنالوگ سیگنال های پیوسته دریافتی از فرآیند را به منظور پردازش داخلی در CPU به سیگنال های دیجیتال تبدیل مینمایند. اکثر ماژول های آنالوگ مجهز به یک DIP Switch برای تعیین نوع و بازه سیگنال می باشند. همچنین، کارت های آنالوگ در PLC های مختلف دارای دقت های متفاوتی می باشند. به عنوان مثال، یک ماژول AI دارای ۴ کانال با دقت ۱۲ بیت و کارت دیگر با دقت ۱۵ بیت می باشد. در واقع، منظور از دقت یک کارت قدرت تفکیک کارت می باشد.

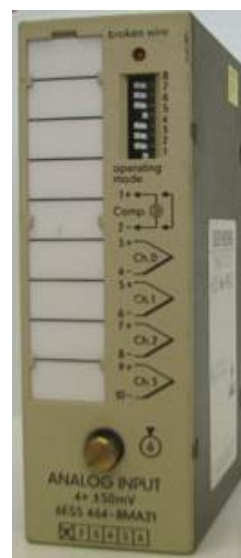
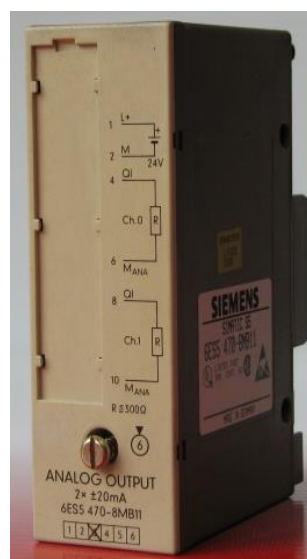


خروجی آنالوگ AO

برای ارسال سیگنال آنالوگ از PLC به سطح پروسه از خروجی‌های آنالوگ استفاده می‌شود. این خروجی‌ها در حقیقت سطح سیگنال داخلی PLC که یک سیگنال دیجیتال می‌باشد را به سیگنال آنالوگ تبدیل می‌کنند. سطوح سیگنال خروجی عبارتند از:

4-20mA , 0-10VDC , 0-5VDC

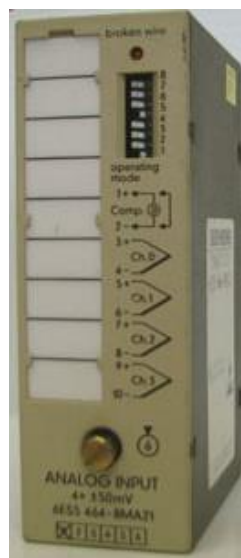
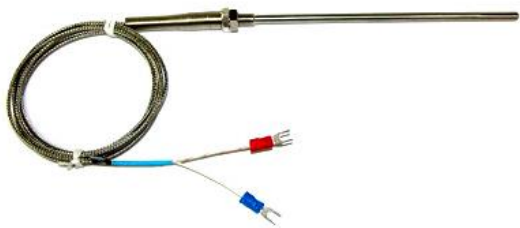
یکسری از شیرها در صنعت وجود دارند که می‌توانند به خروجی آنالوگ PLC متصل شوند. با استفاده از شیر کنترل قادر خواهیم بود تا مسیر عبور مواد را از 0 تا 100٪ در جهت نیاز باز و یا بسته نماییم. برخی از این شیرها دارای آمپلی فایر بوده و سیگنال تحریک به برد الکترونیکی آمپلی فایر ارسال می‌شود. همچنین، در بسیاری از پروژه‌های صنعتی از سیگنال خروجی کارت آنالوگ به عنوان منبع فرمان در ورودی اینورترها استفاده می‌شود. در این صورت PLC با اعمال سیگنال آنالوگ به ورودی درایو می‌تواند عملیات کنترل سرعت را بر عهده گیرد. در کارت‌های خروجی آنالوگ از مبدل‌های D/A استفاده شده است که یک عدد دیجیتال را به ولتاژ یا جریان تبدیل می‌کند.



آشنایی با ماژول‌های خاص

ماژول ترموکوپل

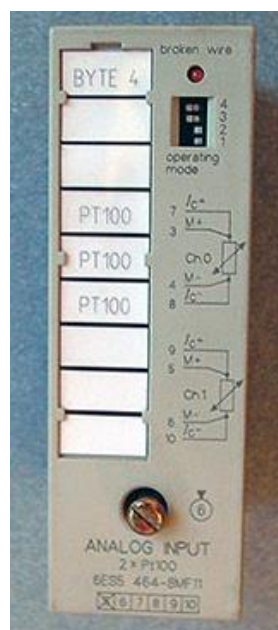
در PLC های سری S5 کارت‌هایی جهت اتصال مستقیم ترموکوپل‌ها جهت اندازه‌گیری دما طراحی شده است. این کارت تحت عنوان کارت مخصوص ترموکوپل شناسایی می‌شود. توسط این کارت، اتصال مستقیم انواع ترمومترها به S5 امکان پذیر می‌باشد. در واقع، در این صورت نیازی به قرار دادن ترانسمیتر بین ترمومتر و کارت نمی‌باشد.



19

ماژول RTD

این ماژول نیز برای اندازه‌گیری حرارت توسط ترمومترها استفاده می‌شود. در واقع، انواع ترمومترها به طور مستقیم می‌توانند به این ماژول متصل شوند.



ماژول تغذیه PS

برای تامین تغذیه 24VDC در PLC S5 با توجه با ساختار، می توان از منابع تغذیه تیپ S5 و یا هر منبع تغذیه دیگر استفاده نمود. تغذیه 24VDC می تواند به عنوان تغذیه پاور PLC، خروجی های دیجیتال و تغذیه ماژول های افزایشی مورد استفاده قرار گیرد. اگر به منبع تغذیه با آمپر بالا نیاز می باشد، می توان از منابع تغذیه SITOP زیمنس استفاده نمود.



کارت های واسط (IM)

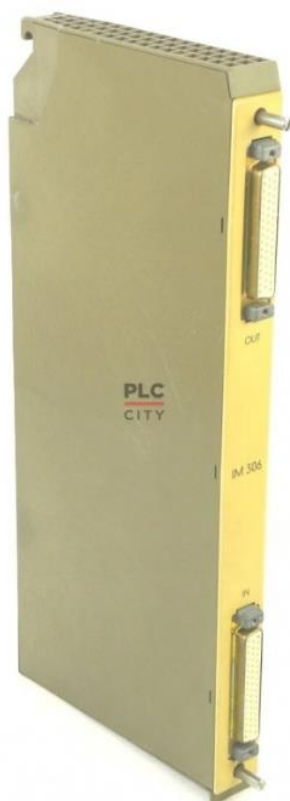
اگر در فرآیندهای صنعتی تعداد I/O های سیستم گسترش یابند، نصب کلیه ماژول ها در یک رک امکان پذیر نیست. در چنین مواردی لازم است تعداد رک ها گسترش یابند. مبادله اطلاعات بین رک اصلی و رک های اضافی از طریق ماژولهای IM صورت می گیرد. رک اصلی را با CR یا CU و رک های توسعه یافته را نیز با ER و یا EU نشان می دهند.



در شکل زیر یک PLC S5-115U را با یک رک توسعه مشاهده می کنید. کارت IM در آخرین اسلات قرار گرفته است.



21



کارت‌های تابعی Function Module

کارت‌های تابعی می‌توانند عملیاتی را مستقل از CPU انجام دهند. این عملیات می‌تواند یکی از کاربردهای زیر باشد:

- ✚ عملیات شمارش
- ✚ عملیات مقایسه
- ✚ عملیات زمان سنجی

آشنایی با ماژول High Speed Counter

از این ماژول‌ها در شمارش پالس‌های فرکانس بالا استفاده می‌شود. در پروسه‌های صنعتی در موارد متعدد مانند دستگاه‌های بسته‌بندی، دسته‌بندی و یا سیستم‌های توزیع می‌توان جهت شمارش تعداد پالس‌های ارسالی از سنسورها و انکودرها از این ماژول‌ها استفاده نمود.



ماژول مقایسه کننده (Comparator Module)

این ماژول دارای دو کانال مستقل برای مقایسه ولتاژ و جریان با یک مقدار دلخواه می باشد. زمانی که مقدار ولتاژ یا جریان به مقدار Preset Value برسد، LED تعبیه شده بر روی ماژول روشن و سیگنال 1 به CPU ارسال می شود.



23

ماژول تایمر (Timer Module)

این ماژول دارای دو تایمر پالسی می باشد. وقتی که تایمرها در حالت Run می باشند، LED تعبیه شده بر روی ماژول روشن و سیگنال 1 برای CPU ارسال می شود. توسط سوئیچ و پتانسیومتر تعبیه شده بر روی ماژول امکان تنظیم زمان برای هر کانال وجود دارد.



کارت‌های هوشمند IP (Intelligent processors)

کارت‌های هوشمند نیز می‌توانند عملیاتی را مستقل از CPU انجام دهند. این عملیات یک عملیات پیچیده که نیاز به پردازش و کنترل بیشتر نسبت به ماژول‌های FM می‌باشد:

کنترل‌های حلقه بسته PID

کنترل موقعیت

Cam‌های الکتریکی

24

ماژول PID (Close Loop Module-IP262)

این ماژول یک ماژول PID می‌باشد که در لوپ‌های کنترلی حلقه بسته استفاده می‌شود. این ماژول برای کنترل لوپ‌های پیوسته و پله‌ای می‌تواند به کار گرفته شود. یک مدل از این ماژول دارای ۳ خروجی آنالوگ و مدل دیگر دارای ۸ خروجی باینری می‌باشد. همچنین این ماژول ۴ ورودی آنالوگ و ۴ ورودی باینری را نیز فراهم می‌کند.

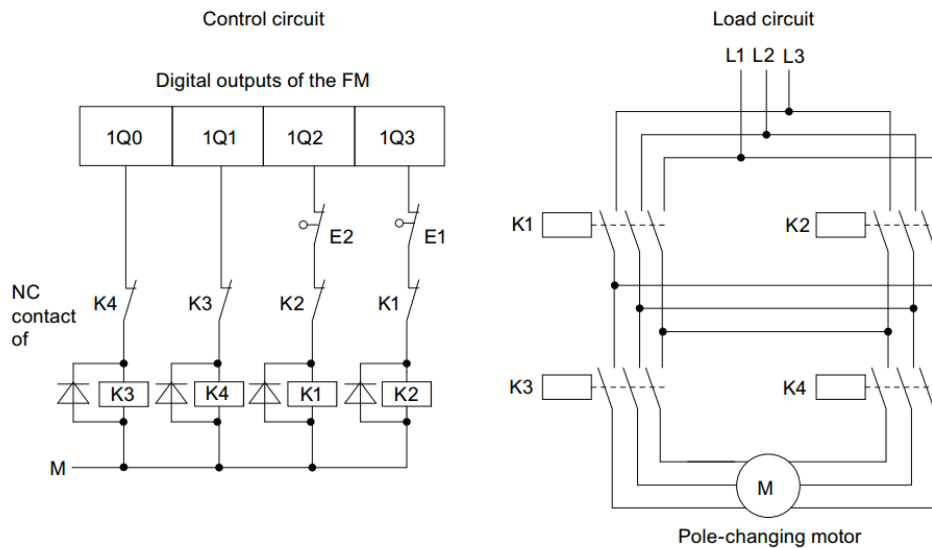


ماژول کنترل موقعیت (Positioning Module-IP263)

این ماژول یک ماژول با ۲ کانال مستقل برای کنترل موقعیت تراورس های سریع و درایوهای لرزشی می باشد. در واقع این ماژول یک ماژول با دو کانال می باشد. برای هر کانال ۴ خروجی دیجیتال در نظر گرفته شده است. هر کانال این ماژول می تواند انکودر افزایشی (Incremental) یا یک انکودر مطلق (SSI) باشد. خروجی های دیجیتال این ماژول برای کنترل ۲ محور مستقل استفاده می شود. برای هر محور از ۴ خروجی دیجیتال استفاده می شود. بخش قدرت توسط خروجی های دیجیتال کنترل می شود.

- Rapid traverse
- Creep speed
- Anti-clockwise rotation
- Clockwise rotation

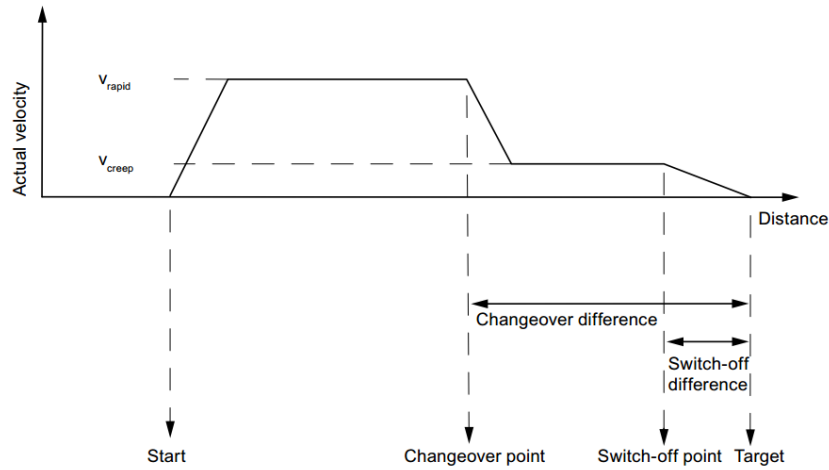
مدار کنتاکتوری



در شکل فوق، از کنتاکتورهای K1 و K2 برای کنترل جهت و از کنتاکتورهای K3 و K4 برای کنترل سرعت سریع و کند استفاده می شود.

حالتی که محرک با بیشترین سرعت شروع به حرکت می کند، همان Rapid Traverse می باشد. در نقطه ای نزدیک به مقصد، محرک به دور کند سوئیچ می شود که این حالت همان Creep Speed می باشد.

کمی مانده به نقطه مقصد، محرک خاموش می شود تا نهایتاً محور در موقعیت Target متوقف شود.

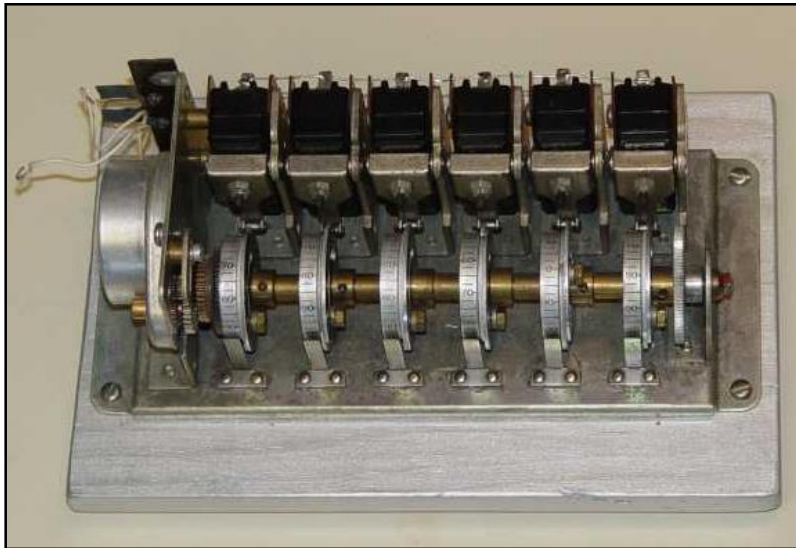


ماژول IP264 را در شکل زیر مشاهده می کنید.



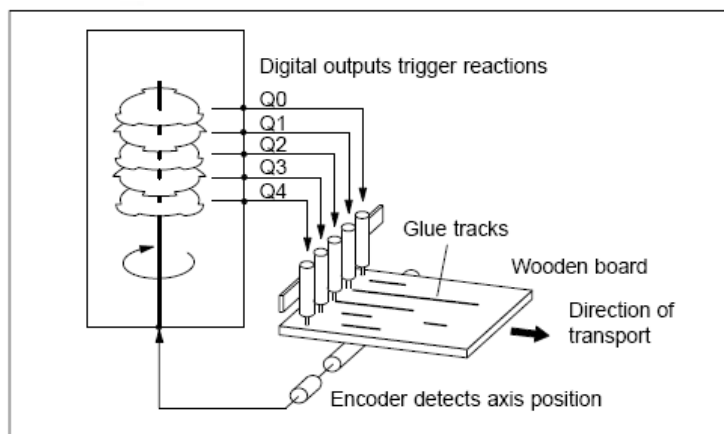
ماژول IP 264 Electronic Cam

این ماژول یک Cam Controller الکتریکی بسیار سریع می باشد که می تواند موقعیت را توسط انکودر تشخیص و فرمانهای مناسب را به پروسه ارسال کند. در بسیاری از ماشین آلات قدیمی سیستم کنترلی در قالب CAM های مکانیکی طراحی می شد.



27

در پروسه شکل فوق چسب روی چوب توسط لوله های حاوی چسب ریخته می شود. IP264 توسط پالس های ارسالی از انکودر به نسبت تعیین شده اقدام به فعال یا غیر فعال کردن خروجی های دیجیتال می کند.



توسط این ماژول می توان یک سیستم CAM الکترونیکی طراحی کرد. این ماژول دارای یک ورودی انکودر (افزایشی و مطلق) جهت دریافت پالس های فیدبک می باشد. همچنین، این ماژول دارای خروجی دیجیتال به عنوان مسیرهای Track می باشد.



ماژول کنترل موقعیت (Positioning Module-IP266)

این ماژول، یک ماژول کنترل موقعیت برای یک درایو با Setpoint آنالوگ می باشد. IP266 یک ماژول کنترل موقعیت با عملکرد بالا برای درایو سرو می باشد.

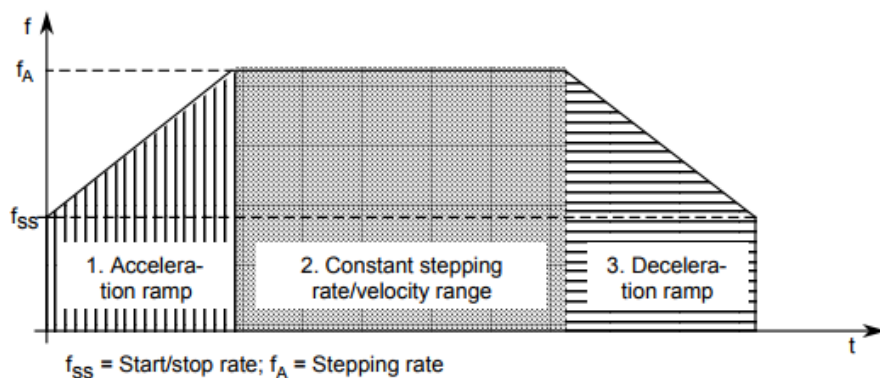


همانطور که در شکل فوق مشاهده می کنید، یک پورت از ماژول با واحد درایو و پورت دیگر نیز با انکودر جهت دریافت فیدبک ارتباط دارد. سیگنال Setpoint که ماژول برای درایو فراهم می کند، یک سیگنال +/- 10V می باشد.

ماژول کنترل موتور پله ای (Positioning Module-IP267)

این ماژول، یک ماژول کنترل موقعیت برای موتورهای پله ای می باشد. در واقع، این ماژول پالس های کنترلی را با درایو یک موتور پله ای یا سرو موتور مخابره می کند.

29



بین های این پورت مربوط به سیگنال های I/O می باشند که می باید برای درایو ارسال و یا از آن سیگنال های فیدبک دریافت شود. سیگنال ها می توانند سیگنال Enable, Direction, Pulse و یا Ready باشند.



ماژول های CP (Communication Module)

توسط این ماژول ها، CPU می تواند به شبکه های مختلف متصل شود. در واقع این ماژول ها، کارت های شبکه برای CPU محسوب می شوند.

Communications Module CP 521SI

این ماژول یک ماژول ارتباطی سریال با دو استاندارد زیر می باشد:

- A printer with TTY interface (active) or
- A printer with V.24 (RS 232C) interface.

از این ماژول جهت اتصال ایستگاه S5 با سایر ایستگاه ها استفاده می شود.

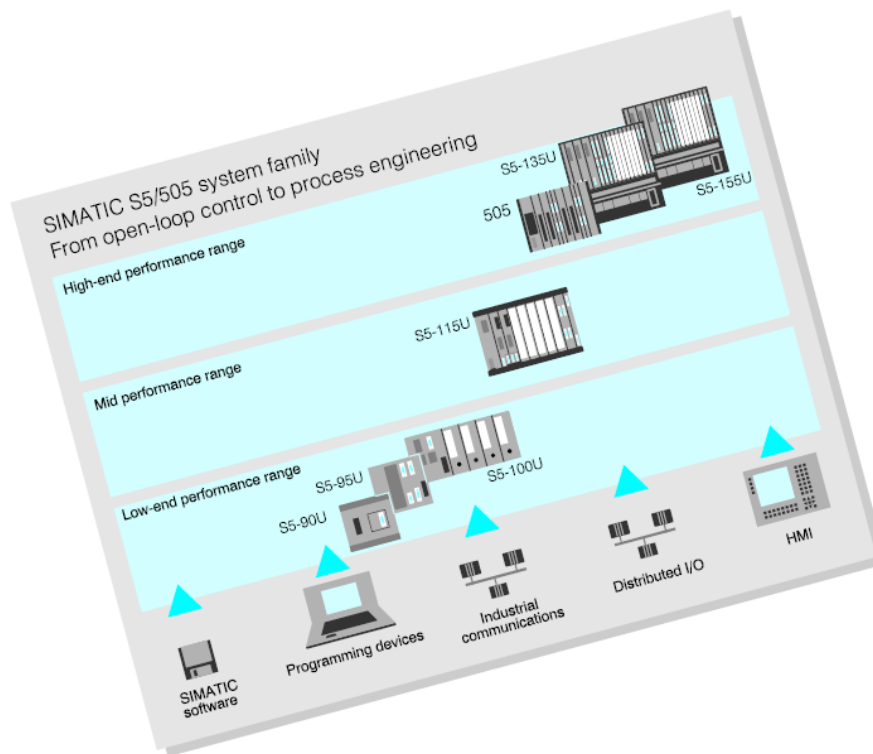
- Keyboard
- Terminal
- Another CP 521 SI
- CP 523
- S5-95U with second serial interface
- CP 524/CP 525-2 (in connection with special driver 6ES5 897-2AB11)
- CPU 944 (with ASCII driver, 3964(R) driver)
- Other I/O devices with serial interface, e.g. bar code readers

ماژول CP521 SI



لازم به ذکر است که PLC های S5 دارای کارت های بسیار متنوع دیگری نیز می باشند که در این مجموعه به آنها اشاره نشده است. در این بخش هدف آشنایی کلی با برخی از ماژول های پر کاربرد در PLC های سری S5 شرکت زیمنس می باشد. علاقمندان جهت آشنایی هر چه بیشتر با سایر ماژول های S5 می توانند به کاتالوگ های زبان اصلی مراجعه کنند.

آشنایی با مدل های مختلف PLC S5



آشنایی با S5-90U

S5-90U می تواند در پروژه های کوچک ، جهت صرفه جویی در حجم تابلو و هزینه به کار رود. S5-90U جایگزین مناسبی به جای رله ها و کنتاکتورها در تابلوهای فرمان می باشد. تعداد ورودی و خروجی های Onboard این PLC کم و قابلیت افزایش I/O ها توسط IM 90 وجود دارد. این S5 می تواند ۶ ماژول افزایشی را توسط IM90 ساپورت کند. حداکثر تعداد ورودی/خروجی دیجیتال برای این مدل 208 بیت و حداکثر تعداد ورودی و خروجی آنالوگ نیز ۱۶ کانال می باشد.

Inputs/Outputs	S5-90U	
	Onboard	Maximum
Digital inputs/outputs	16	208
Analog inputs/outputs	0	16
Counter inputs	1	13
Interrupt inputs	1	1

PLC S5-90U



33

برخی از مشخصات:

- ۱- ولتاژ پاور ورودی: 220/115 VAC
- ۲- دارای منبع تغذیه داخلی 24VDC با جریان 100mA
- ۳- دارای 10 ورودی دیجیتال به صورت Onboard
- ۴- دارای 6 خروجی دیجیتال رله ای
- ۵- دارای باتری Backup
- ۶- دارای رابط IM90 جهت افزایش
- ۷- دارای اسلات کارت حافظه

مشخصات فضای حافظه در S5-90U

فضای حافظه فلگ در این مدل 128 بایت بوده که از بایت 0 تا 63 بصورت پایدار و مابقی بصورت ناپایدار می باشد. همچنین این مدل دارای 32 شمارنده بوده که از C0 تا C7 بصورت پایدار و مابقی بصورت ناپایدار می باشد. منظور از پایدار بودن یک حافظه این است که در صورت قطع شدن برق PLC در صورت وجود باتری پشتیبان اطلاعات موجود در آن بخش از حافظه حفظ می شود.

S5-90U دارای ۳۲ عدد تایمر می باشد که تمامی آنها یعنی از شماره T0 تا T31 بصورت ناپایدار می باشند.

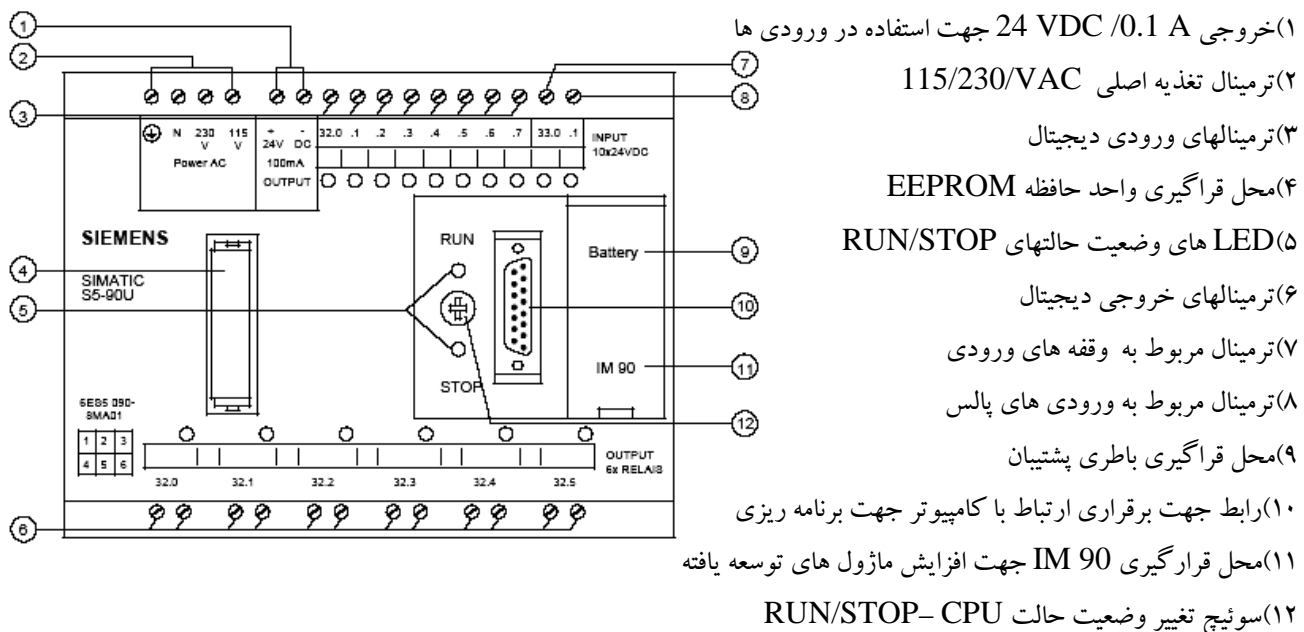
Operand	S5-90U	
	Retentive	Non-Retentive
Flags	0.0 to 63.7	64.0 to 127.7
Counters	0 to 7	8 to 31
Timers	_____	0 to 31

مشخصات کلی S5-90U

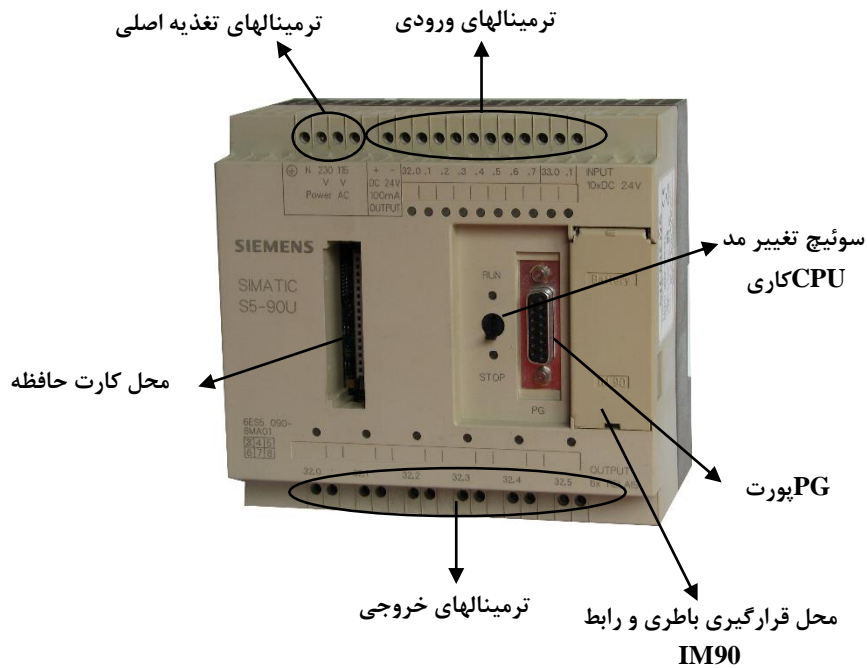
Software Overview	S5-90U
User memory (1 statement=2 bytes)	4 KB
Execution time for 1024 binary statements	2 ms
Flags (512 retentive)	1024
Counters (8 retentive)	32
Timers	32
Program organization:	
• Structured programming	Yes
• Interrupt-driven program execution	Yes
• Time-controlled program execution	No
• PID controller	No
• Standard function blocks	No

آشنایی با سخت افزار S5-90U

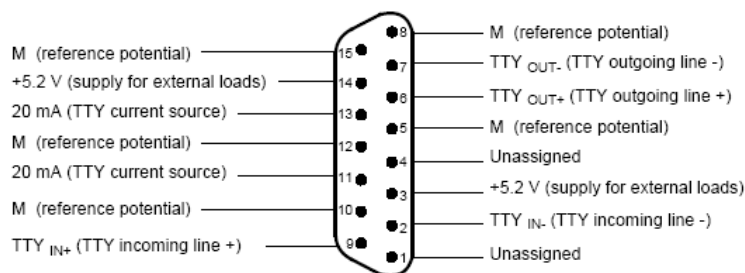
برد اصلی این S5 بصورت کامپکت بوده که دارای یکسری ورودی و خروجی دیجیتال نیز می باشد.



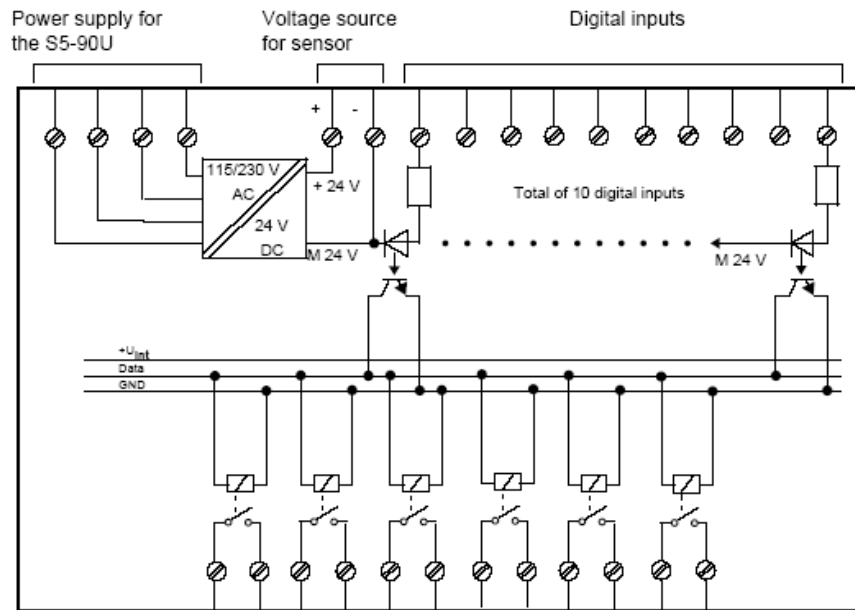
در شکل زیر یک S5-90U را ملاحظه می کنید.



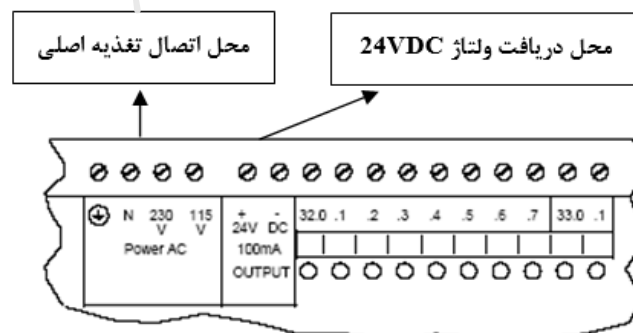
بر روی تمامی مدل های S5 یک پورت ۱۵ پین وجود دارد که توسط آن می توان به PG ها و یا کامپیوتر متصل شد. شکل زیر پین های این پورت را نشان می دهد.



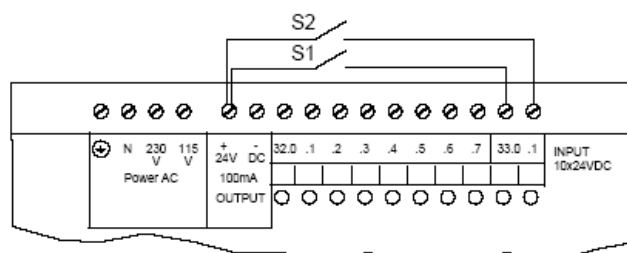
مدار داخلی PLC S5-90U



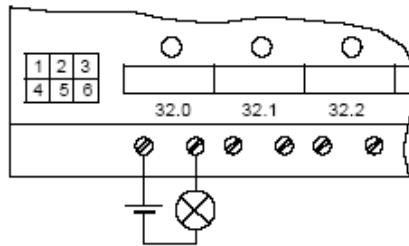
بر روی این مدل یک محل جهت اتصال تغذیه AC تعبیه شده است. این ترمینال ها با نام Power AC Input مشخص شده است. ولتاژ مورد نیاز ورودی های دیجیتال در این مدل 24VDC بوده و خروجی های آن نیز از نوع رله ای می باشند. این S5 دارای ۱۰ ورودی دیجیتال و ۶ خروجی دیجیتال رله ای می باشد. در ضمن بر روی برد اصلی این مدل محلی نیز جهت دریافت ولتاژ 24VDC در نظر گرفته شده است. جریان این منبع 100mA بوده که جهت تامین ولتاژ سنسورها و ورودی های دیجیتال مناسب می باشد.



همانطور که در شکل زیر ملاحظه می کنید از ولتاژ 24VDC می توان جهت تامین ولتاژ مورد نیاز ورودی های دیجیتال استفاده نمود.



شکل زیر نیز نحوه اتصال یک لامپ سیگنال را به خروجی دیجیتال نشان می دهد.



لازم به ذکر است که توسط خروجی های دیجیتال از نوع رله ای به راحتی می توان یک کنتاکتور بوین AC را تحریک نمود.

37

نصب S5-90U به همراه ورودی-خروجی های توسعه یافته

جهت افزایش تعداد ماژولها در S5-90U به قسمتهای زیر نیاز می باشد:

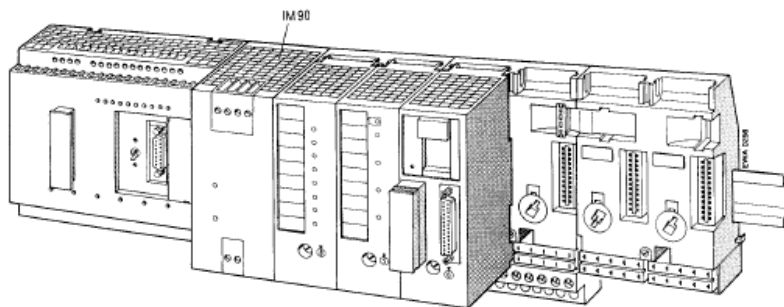
-برد اصلی S5-90U

-ماژول رابط IM 90

-واحد ارتباط

-ریل استاندارد

در شکل زیر نحوه ارتباط بین ماژولهای گسترش یافته و برد اصلی را ملاحظه می نمایید.



S5-90U از طریق IM90 می تواند با ماژول های افزایشی و یا ماژول های S5-100U ارتباط برقرار کند. بر روی برد اصلی این مدل یک سوکت تعبیه شده است که از این طریق می توان به ماژول های افزایشی متصل شد.

سوئیچ دو حالتی موجود بر روی S5-90U جهت RUN و STOP نمودن CPU به کار می رود. در حالت STOP، CPU هیچ برنامه ای را اجرا نمی کند و یا به عبارت دیگر به ورودی ها و خروجی ها دسترسی ندارد. برعکس در حالت RUN که LED سبز روشن می گردد، PLC در حال اجرای برنامه می باشد.

آشنایی با S5-95U

از این مدل در کاربردهای کوچک و بعضاً متوسط استفاده می شود. این مدل دارای ورودی و خروجی های دیجیتال و آنالوگ Onboard بوده و جوابگوی مناسبی در صنعت جهت استفاده در کنترل ماشین آلات صنعتی کوچک و متوسط می باشد.

از مهمترین مزایای این سری نسبت به S5-90U افزایش چشمگیر تعداد ورودی و خروجی ها می باشد. همچنین این مدل دارای ورودی و خروجی آنالوگ Onboard و ورودی های پالس نیز می باشد. رابط های موجود بر روی S5-95U به شرح زیر می باشند:

- ۱- رابط ورودی و خروجی های آنالوگ
- ۲- رابط OP، PC، PG و یا SINEC L1
- ۳- رابط ورودی های وقفه و شمارنده



قابلیت ها و امکانات S5 95U

- * ورودی و خروجی دیجیتال و آنالوگ Onboard
- * قابلیت استفاده در PID کنترلرها (کنترل های حلقه بسته)
- * قابلیت ایجاد شبکه با ایستگاه های مختلف از طریق SINEC L2
- * قابلیت اتصال ورودی های فرکانس بالا
- * قابلیت اتصال وقفه های خارجی

ورودی و خروجی ها:

S5-95U قابلیت افزایش ورودی و خروجی ها را از طریق ماژولهای ارتباط دهنده دارا می باشد. در S5-95U ماژول ها می توانند بصورت مستقیم به برد اصلی متصل شوند، اما در S5-90U جهت افزایش سایر کارت ها از IM90 استفاده می گردد.

در این سری حداکثر می توان تعداد I/O های دیجیتال را تا 480 بیت افزایش داد. در ضمن این مدل نیز به راحتی می تواند با ماژول های افزایشی که همان ماژول های S5-100U می باشند ارتباط برقرار کند. تعداد ماژولی که می توان به برد اصلی در این مدل اضافه نمود ۳۲ کارت می باشد. همچنین در این سری تعداد کانال های ورودی و خروجی آنالوگ را می توان تا ۴۱ کانال افزایش داد. بر روی برد اصلی S5-95U تعداد ۹ کانال بصورت Onboard تعبیه شده است.

تعداد ورودی های شمارنده و وقفه های خارجی

یکی از قابلیت های این مدل دارا بودن تعدادی ورودی، جهت اعمال پالس های خارجی جهت شمارش و یا اعمال وقفه به CPU می باشد. بر روی برد اصلی ۲ کانال جهت اعمال پالس با فرکانس بالا و ۴ ورودی جهت اعمال وقفه در نظر گرفته شده است. همچنین تعداد کانالهای شمارنده را می توان توسط ماژول های HSC تا ۶۶ کانال افزایش داد.

Inputs/Outputs	S5-95U	
	Onboard	Maximum
Digital inputs/outputs	32	480
Analog inputs/outputs	9	41
Counter inputs	2	66
Interrupt inputs	4	4

انواع ماژولهای ورودی و خروجی قابل افزایش :

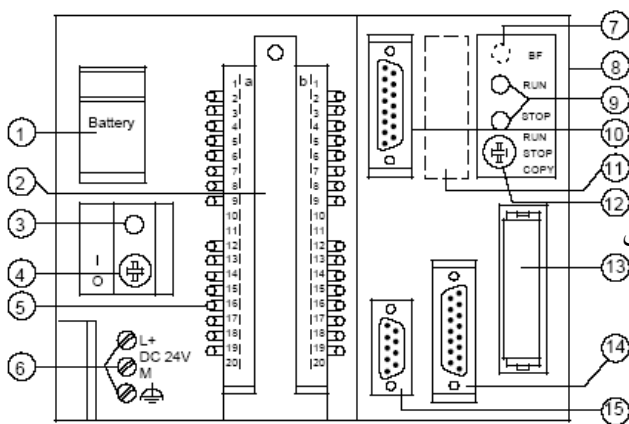
- ❖ ورودی و خروجی دیجیتال
- ❖ ورودی و خروجی آنالوگ
- ❖ ماژولهای EX جهت کاربرد در محیط های حساس و خطرناک
- ❖ I/O های هوشمند برای کاربردهای خاص
- ❖ ماژول Simulator

فضای حافظه فلگ در S5-95U دو برابر فضای حافظه S5-90U می باشد. تعداد تایمرها و شمارنده ها نیز در این مدل بسیار بیشتر از مدل قبلی بوده که تمامی موارد بیان شده نشان از قدرت و قابلیت های بیشتر این مدل نسبت به مدل قبلی می باشد.

Operand	S5-95U	
	Retentive	Non-Retentive
Flags	0.0 to 63.7	64.0 to 255.7
Counters	0 to 7	8 to 127
Timers	_____	0 to 127

آشنایی با سخت افزار S5-95U

بررسی قسمتهای مختلف S5-95U



(۱) محل قرارگیری باتری پشتیبان

(۲) ترمینالهای I/O

(۳) LED مربوط به وضعیت باتری

(۴) کلید قطع و وصل تغذیه

(۵) LED های نمایشگر وضعیت، مربوط به بیت های ورودی و خروجی

(۶) ترمینال مربوط به تغذیه

(۷) LED خطای شبکه SINEC L2

(۸) رابط ارتباط دهنده با سایر ماژول های افزایشی

(۹) LED های وضعیت حالت های RUN/STOP

(۱۰) رابط - جهت استفاده به عنوان ورودی و خروجی آنالوگ

(۱۱) رابط SINEC L2

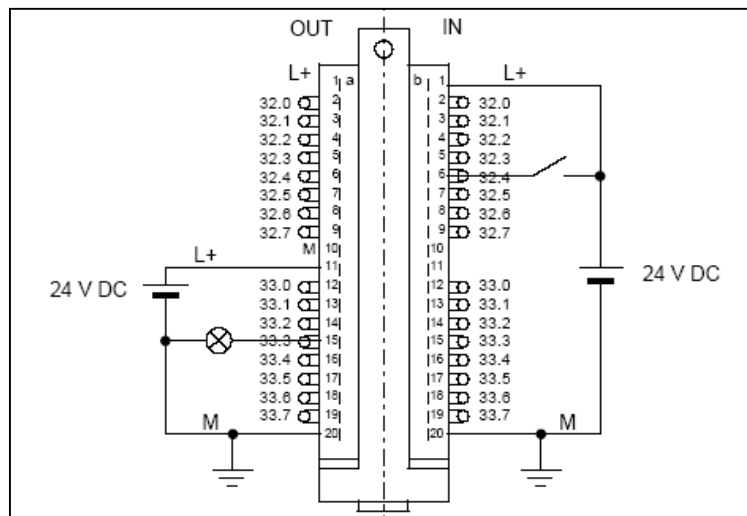
(۱۲) سوئیچ تغییر وضعیت حالت CPU - RUN/STOP/COPY

(۱۳) محل قرارگیری واحد حافظه

(۱۴) رابط پروگرامر، PC، OP یا باس SINEC L1

(۱۵) رابط برای ورودی های وقفه و شمارنده

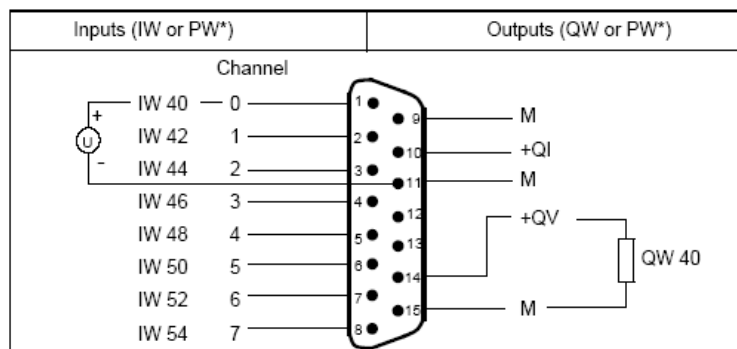
نحوه سیم کشی در I/O های دیجیتال



خروجی های دیجیتال در این مدل ترانزیستوری می باشند. پس اتصال ولتاژ 220VAC به سر مشترک خروجی ها امکان پذیر نبوده و تنها مطابق شکل صفحه قبل کاربر مجاز به اتصال مصرف کننده DC می باشد.

پورت مربوط به ورودی و خروجی های آنالوگ

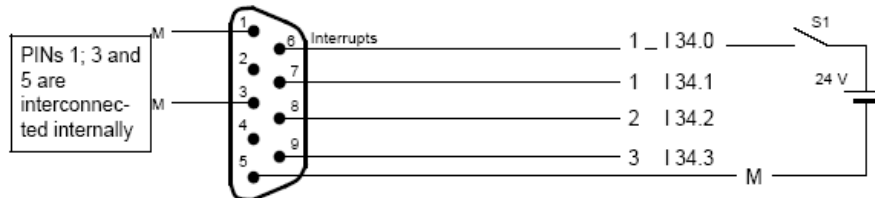
S5-95U دارای ۸ ورودی آنالوگ و ۱ خروجی آنالوگ بصورت Onboard می باشد. توسط رابط موجود بر روی این مدل، سیگنالهای استاندارد که از سنسورها، ترانسمیترها و ... ارسال می شوند را می توان جهت پردازش به PLC ارسال نمود.



سیگنال های آنالوگ ورودی و خروجی می توانند بصورت ولتاژ و یا جریان باشند.

پورت مربوط به ورودی های وقفه

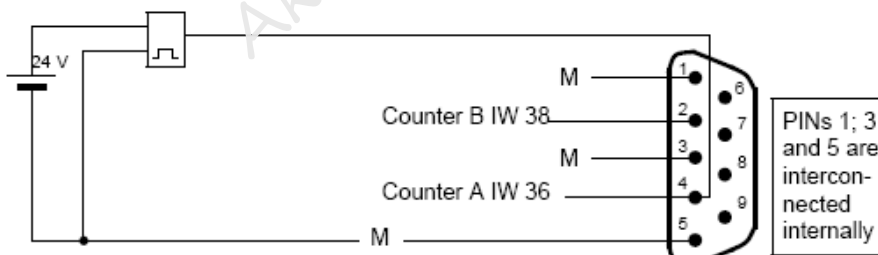
در این سری این امکان نیز وجود دارد که بتوان ورودی هایی را بصورت وقفه از خارج به PLC اعمال نمود. تعداد ورودی های وقفه در این مدل ۴ عدد می باشد.



42

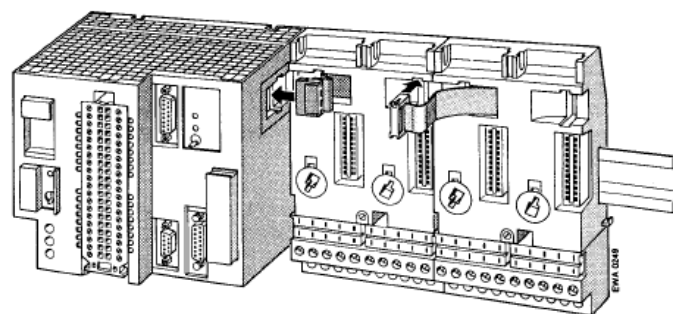
پورت مربوط به ورودی های پالس

در بسیاری از کاربردها مانند تعیین موقعیت یک محور و یا حرکت آن و همچنین شمارش قطعات در ماشین آلات بسته بندی از سنسورهای فرکانس بالا استفاده می شود. در این حالت با توجه به بالا بودن فرکانس پالس ارسالی از سنسور، می توان جهت شمارش پالس ها از ورودی های پالس تعبیه شده بر روی PLC ها استفاده نمود.

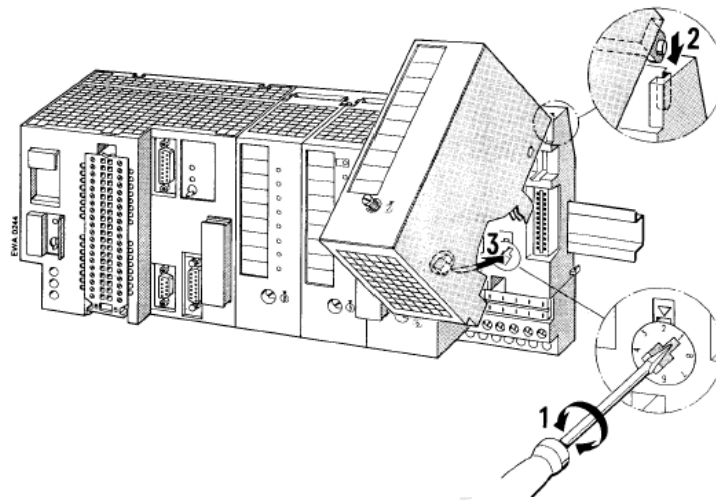


S5-95U دارای ۳۲ ورودی / خروجی دیجیتال بصورت Onboard می باشد. علاوه بر آن پورت هایی نیز جهت اعمال سیگنالهای آنالوگ و پالس به PLC وجود دارد. تغذیه اصلی در این مدل 24VDC می باشد که این حالت در مدل قبلی 220/115 VAC بود.

ماژول های افزایشی S5-100U به راحتی می توانند از طریق سوکت تعبیه شده در سمت راست برد اصلی S5-95U به CPU متصل شوند.

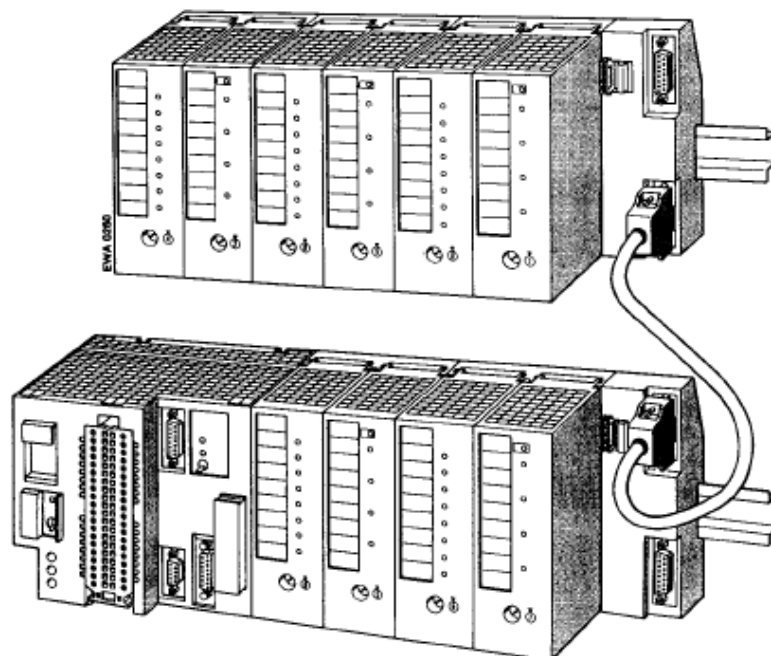


طریقه قرار گیری کارت ها بر روی رک



43

همچنین در بسیاری از مواقع می توان عملیات افزایش کارت ها را در چندین طبقه پیاده سازی نمود. جهت انجام این کار می بایست از کارت واحد IM 315 در هر طبقه استفاده کرد. فاصله بین طبقات می تواند حداکثر 0.5 متر باشد. توسط IM 315 می توان تنها یک رک اضافی به CPU متصل نمود.



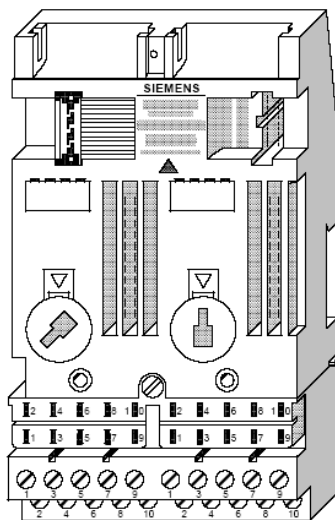
در شکل زیر یک IM 315 را ملاحظه می کنید.



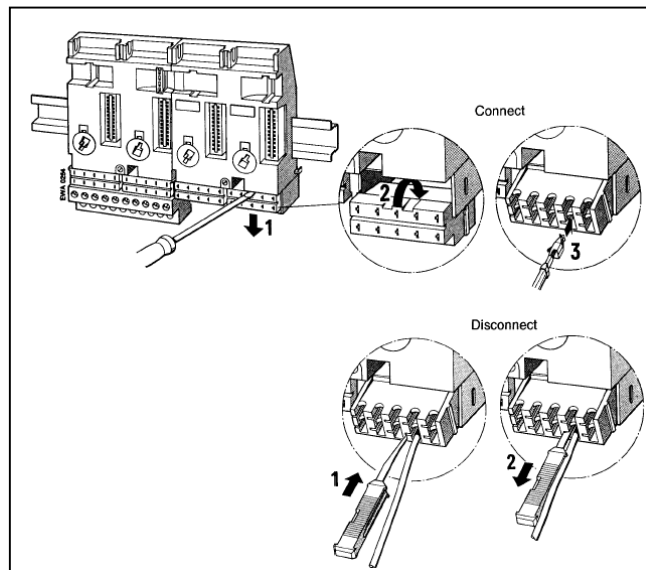
44

آشنایی با BUS UNIT

زمانی که کارت های افزایشی می خواهند با برد اصلی ارتباط برقرار کنند، ابتدا می بایست بر روی یک رک قرار گیرند. این واحد وظیفه ارتباط کارت ها را با CPU از طریق کابل و رابط مخصوص بر عهده دارد.



واحد رک نیز به راحتی می تواند بر روی ریلهای استاندارد نصب گردد. بر روی این واحدها اسلایدرهای نگهدارنده ریل همانند کنتاکتورها و سایر قطعات تعبیه شده است.



کاربرد پورت PG در S5

تمامی سری های S5 مجهز به یک پورت ۱۵ پین با نام PG می باشند که از این پورت جهت اتصال به کامپیوتر و یا PG های صنعتی جهت دانلود و آپلود کردن برنامه استفاده می شود. البته در بسیاری از کاربردها از این پورت جهت اتصال به صفحه نمایشگر نیز استفاده می گردد.



نکات مهم و کاربردی

۱- بر روی S5-95U یک LED با نام Battery Failure Indication که به رنگ زرد می باشد وجود دارد. زمانی که سطح ولتاژ باتری پشتیبان از 2.8 V کمتر شود و یا باطری خالی و یا در برخی موارد به درستی جازده نشود این LED روشن می شود. در این صورت تحت هیچ شرایطی نباید برق دستگاه را قطع نمود.

۲- زمانی که قرار است یک برنامه جدیدی به PLC دانلود شود حتما نیاز به یک Overall Reset می باشد. با انجام یک ری ست کلی موارد زیر پاک می شوند:

- ۱- برنامه موجود در حافظه (شامل بلوک های برنامه نویسی و اطلاعاتی)
- ۲- مقادیر حافظه (شامل فلگ ها ، تایمرها و شمارنده ها)
- ۳- مقادیر و اطلاعات سیستمی به حالت پیش فرض باز می گردند.

طریقه انجام Overall Reset در S5-90U

- ۱- قرار دادن سوئیچ در حالت STOP
- ۲- خارج نمودن باطری Backup
- ۳- قطع برق اصلی PLC به مدت ۱۵ ثانیه
- ۴- وصل مجدد تغذیه و جازدن باطری

طریقه انجام Overall Reset در S5-95U

- ۱- قرار دادن سوئیچ در حالت STOP
- ۲- خارج نمودن باطری Backup
- ۳- قرار دادن سوئیچ در وضعیت OFF
- ۴- قرار دادن سوئیچ در وضعیت ON
- ۵- جازدن باطری

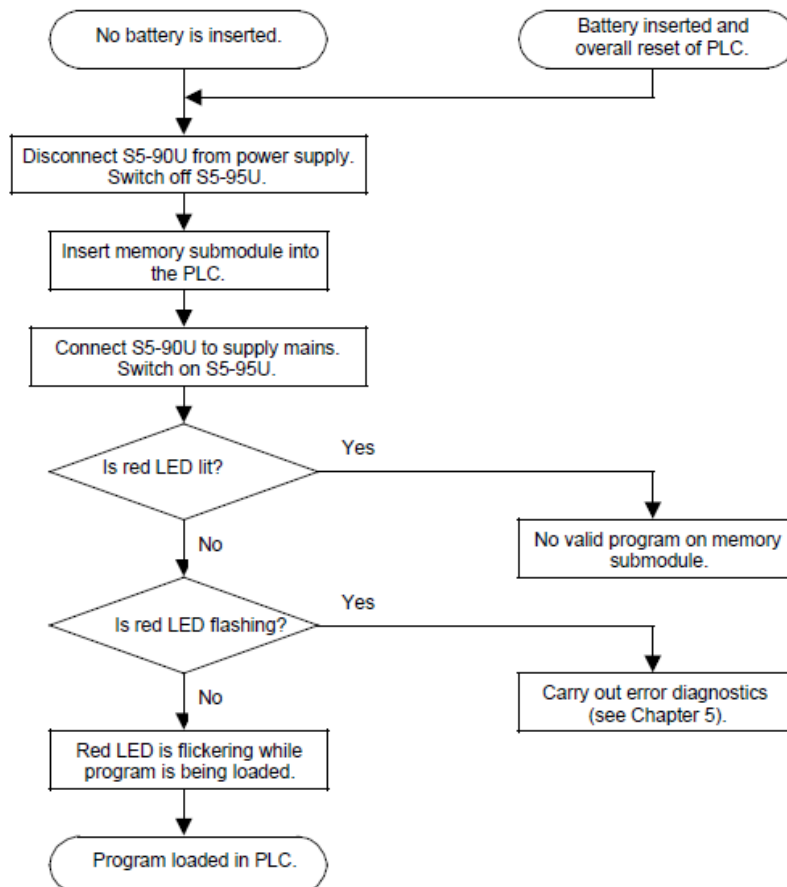
واحد کارت حافظه

همانطور که در قسمتهای قبلی مشاهده نمودید بر روی هر دو مدل S5 یک محل جهت قرار گیری کارت حافظه وجود دارد. در صورت وجود کارت حافظه و ذخیره برنامه کنترلی بر روی آن، در زمانی که برق دستگاه بدون وجود باطری قطع می شود و یا یک Overall Reset صورت می گیرد، PLC بعد از وصل تغذیه بصورت اتوماتیک اطلاعات را از روی کارت حافظه خوانده و داخل RAM خود کپی می کند. همچنین یکی دیگر از کاربردهای کارت حافظه، انتقال برنامه یک PLC به PLC دیگر بدون نیاز به کامپیوتر و یا دستگاه برنامه ریز می باشد.

47



مراحل خواندن برنامه بصورت اتوماتیک از روی کارت



۱- در صورت وجود و یا عدم وجود باطری

۲- قطع نمودن برق اصلی PLC

۳- قرار دادن کارت حافظه

۴- وصل برق اصلی PLC

۵- آیا LED قرمز روشن است؟

بله:

برنامه بر روی کارت حافظه موجود نمی باشد.

خیر:

مرحله بعد

۶- آیا LED قرمز به حالت چشمک زن می باشد؟

بله:

تشخیص خطا

خیر:

مرحله بعد

۷- زمانی که عملیات بارگذاری آغاز می شود، LED قرمز نیز همانند LED موجود بر روی CDROM یک کامپیوتر که در زمان خواندن CD چشمک می زند، شروع به چشمک زدن می کند.

۸- برنامه به داخل PLC بارگذاری گردید.

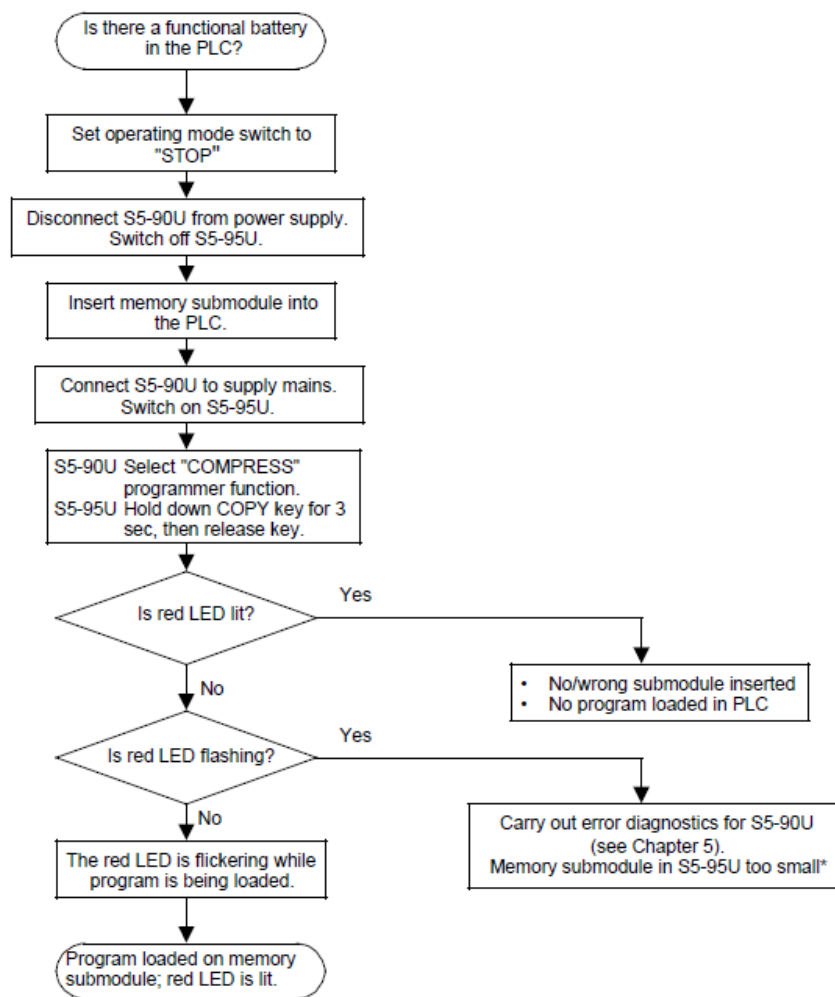
نحوه ذخیره نمودن برنامه بر روی کارت حافظه

مراحل ذخیره نمودن برنامه بر روی کارت حافظه در S5-90U تنها در صورتی گزینه Compress در حالت آنلاین در پروگرامر فعال شود امکان پذیر می باشد. در حالت عادی زمانی که برنامه ای به PLC دانلود می شود این برنامه بر روی حافظه RAM آن PLC ذخیره می گردد.

در بسیاری از کاربردها جهت داشتن نسخه ای مطمئن از برنامه کنترلی و یا انتقال یک برنامه از یک PLC به PLC دیگر بدون نیاز به داشتن کامپیوتر و یا دستگاه برنامه ریز از کارت های حافظه استفاده می شود.



مراحل ذخیره برنامه بر روی کارت حافظه



۱- آیا باتری بر روی PLC وجود دارد؟

۲- قرار دادن CPU در وضعیت STOP

۳- قطع نمودن برق اصلی PLC

۴- قرار دادن کارت حافظه

۵- وصل برق اصلی PLC

۶- در S5-90U: انتخاب گزینه Compress موجود در پروگرامر در وضعیت آنلاین

در S5-95U: نگه داشتن کلید در وضعیت Copy به مدت ۳ ثانیه و سپس رها کردن آن

۷- آیا LED قرمز روشن است

بله:

برنامه ای به داخل PLC لود نشده است - کارت حافظه به درستی جا زده نشده است

خیر:

مرحله بعد

۸- آیا LED قرمز به حالت چشمک زن می باشد؟

بله:

تشخیص خطا - کم بودن فضای حافظه کارت

خیر:

مرحله بعد

۹- زمانی که عملیات ذخیره اطلاعات بر روی کارت آغاز می شود LED قرمز نیز همانند LED موجود بر روی CDROM یک کامپیوتر که در زمان خواندن CD چشمک می زند، شروع به چشمک زدن می کند.

۱۰- برنامه به داخل کارت منتقل و سپس LED قرمز روشن می شود.

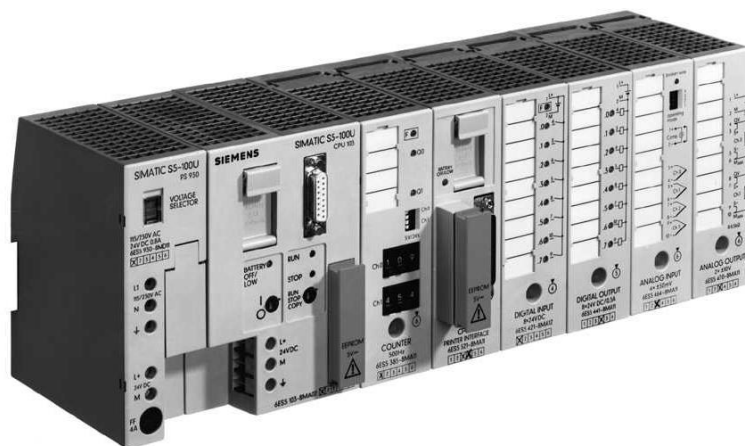
*هیچ وقت در زمانی که برق PLC قطع می باشد اقدام به تعویض و یا خروج باتری Backup نکنید. حتما عملیات تعویض باتری را در زمانی که برق PLC وصل می باشد انجام دهید.

*در S5-95U کلید موجود بر روی PLC دارای سه حالت RUN، STOP و COPY می باشد.

*هیچ گاه زمانی که تغذیه PLC وصل می باشد اقدام به وارد و یا خارج نمودن کارت حافظه نکنید.

آشنایی با S5-100U

S5-100U دارای ساختاری کاملا مدولار و از خانواده SIMATIC S5 می باشد. این مدل از PLC سری S5، جهت کنترل ماشین آلات صنعتی، خطوط تولید و استفاده در سیستمهای مانیتورینگ، بسیار مناسب می باشد. این مدل نیز همانند دو مدل قبلی قابلیت افزایش ماژولهای مختلف را در سطوح بالاتر دارا می باشد. S5-100U دارای ورودی / خروجی Onboard نمی باشد و تمامی واحدها بصورت ماژول های جداگانه به CPU متصل می شوند. همچنین این مدل دارای CPU های مختلفی می باشد.



انواع CPU در S5-100U

CPU100
CPU102
CPU103

فضای حافظه، تعداد تایمرها و شمارنده ها

Operand	Retentive CPU 100 to 103	Non-Retentive		
		CPU 100	CPU 102	CPU 103
Flags	0.0 to 63.7	64.0 to 127.7	64.0 to 127.7	64.0 to 255.7
Counters	0 to 7	8 to 15	8 to 31	8 to 127
Timers	_____	0 to 15	0 to 31	0 to 127

51

S5-100U شامل قسمت‌های زیر می باشد:

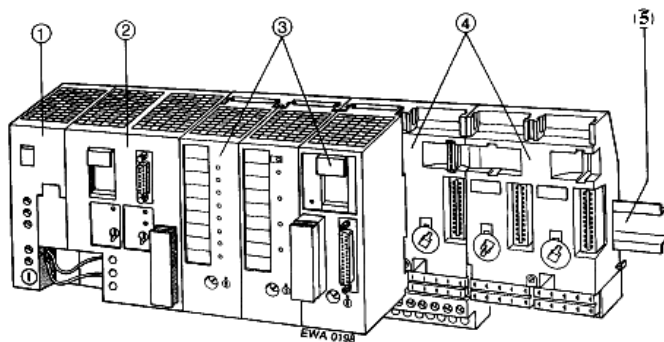
۱- منبع تغذیه تیپ S5-100U

۲- ماژول CPU

۳- رک و سایر کارت ها

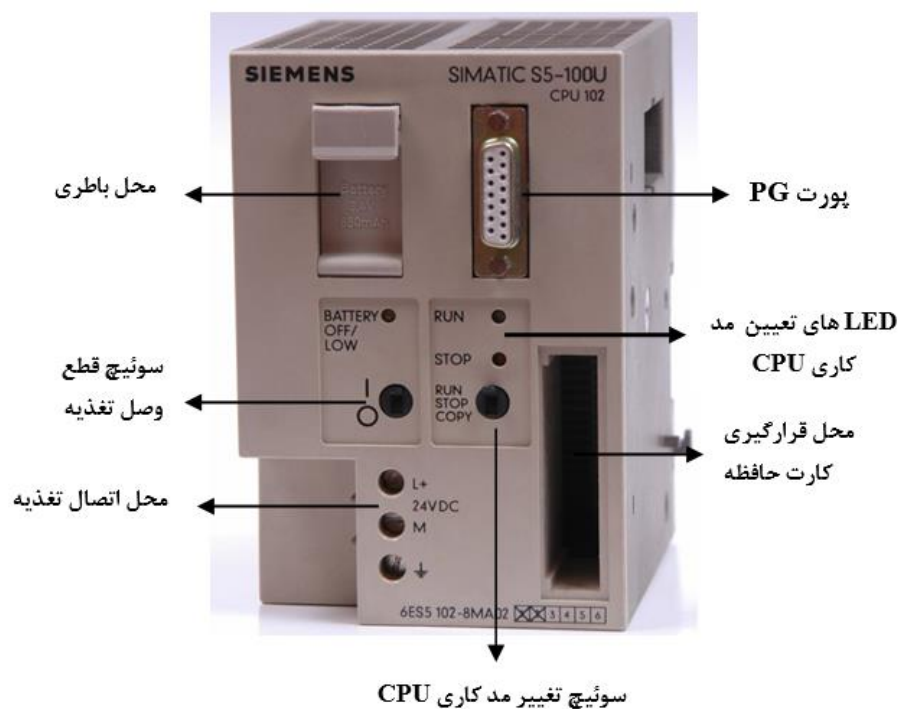


در ادامه قسمتهای مربوط به S5-100U نشان داده شده است.



- ۱- PS 930 (منبع تغذیه)
- ۲- واحد پردازنده مرکزی CPU
- ۳- ماژولهای ورودی، خروجی و IM
- ۴- واحد ارتباط (رک)
- ۵- ریل استاندارد

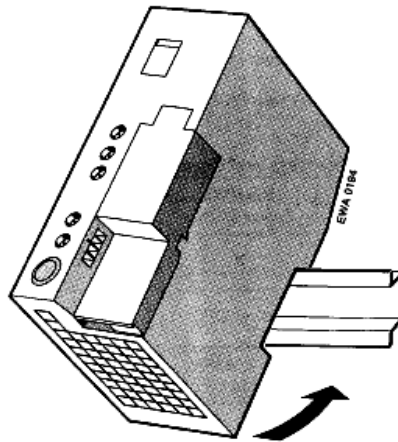
آشنایی با کارت CPU در S5-100U



با دقت در شکل فوق ملاحظه می کنید که بر روی CPU محلی جهت اتصال تغذیه ۲۴ ولت تعبیه شده است. این تغذیه می تواند از منابع تغذیه تیپ S5-100U و یا هر منبع تغذیه استاندارد دیگری تامین شود.

نصب کارتهای S5-100U بر روی ریل استاندارد، همانند نصب ماژولهای گفته شده در بحثهای قبلی می باشد.

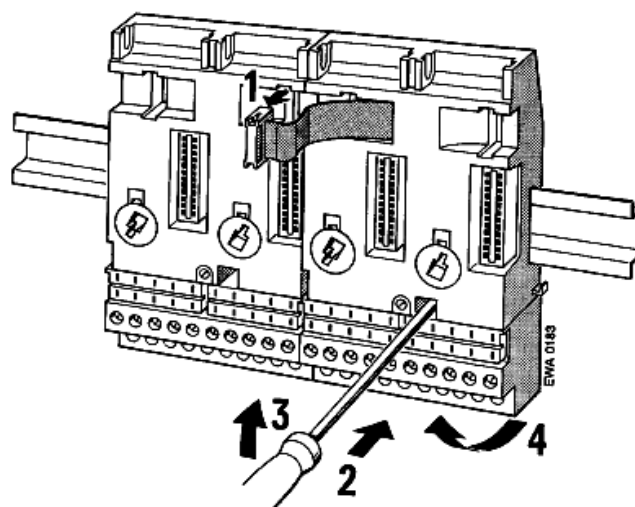
جهت نصب ماژولها، قلاب موجود در پشت کارت را در بالای ریل قرار داده و سپس اسلایدر را به طرف پایین کشیده و پس از قرار دادن کارت در روی ریل، اسلایدر را رها نموده تا کارت مورد نظر بر روی ریل قرار گیرد.



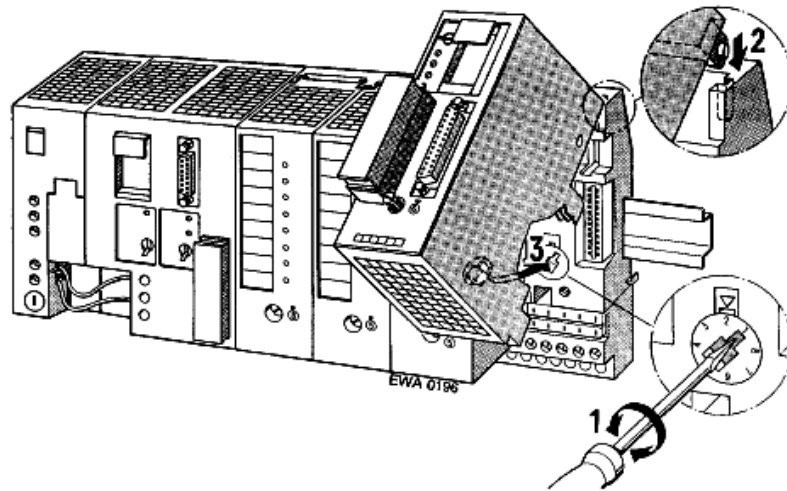
نصب رک بر روی ریل استاندارد:

جهت نصب Bus Unit مراحل زیر را دنبال می نمایم:

- ۱- کابل ارتباطی بین دو واحد را به یکدیگر متصل می کنیم.
- ۲- اسلایدر موجود در پشت Bus Unit را با پیچ گوشتی به طرف پایین می کشیم.
- ۳- Bus Unit را به طرف ریل نزدیک می نمایم تا با ریل درگیر شود.
- ۴- اسلایدر را رها کرده تا Bus Unit به ریل قفل گردد.



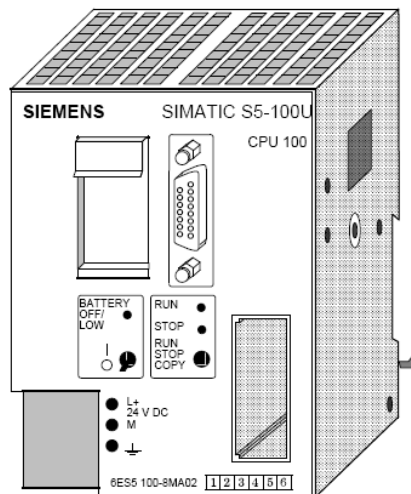
در شکل زیر نحوه قرارگیری ماژول ها را بر روی Bus Unit ملاحظه کنید.



54

آشنایی با انواع CPU در S5-100U

CPU 100



مشخصات فنی

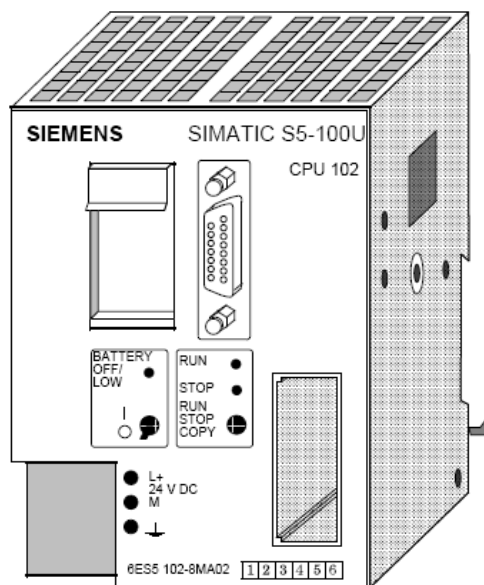
Memory configuration	
- internal memory	RAM 1024 statements
- memory submodule	EPROM/EEPROM

Timers: Number/range	approx. 16; 0.01 to 9990 s
Counters: Number/range	16; 8 retentive 0 to 999 (up/down)

Organization blocks	OB1, 21, 22, and 34
Program blocks	0 to 63
Function blocks	
- programmable	0 to 63
- integrated	none
Sequence blocks	none
Data blocks	2 to 63

Digital inputs, Digital outputs	together max. 256
Analog inputs, Analog outputs	together max. 8

CPU102

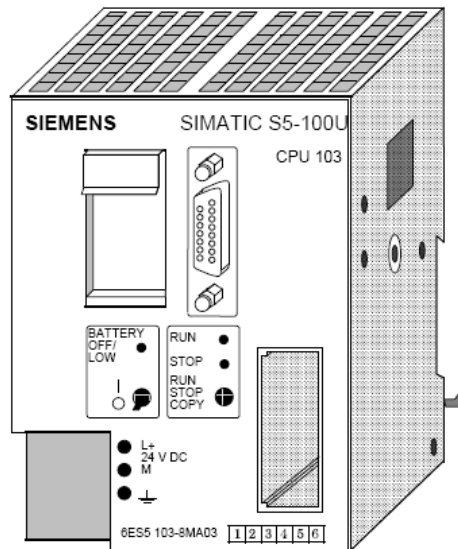


55

مشخصات فنی

Memory configuration	
- internal memory	RAM 2048 statements
- memory submodule	EPROM/EEPROM
Flags	1024; 512 retentive
Timers: Number/range	approx. 32; 0.01 to 9990 s
Counters: Number/range	32; 8 retentive 0 to 999 (up/down)
Organization blocks	OB1, 21, 22, and 34
Program blocks	0 to 63
Function blocks	
- programmable	0 to 63
- integrated	240 to 243, 250, and 251
Sequence blocks	none
Data blocks	2 to 63
Digital inputs, Digital outputs	together max. 256
Analog inputs, Analog outputs	together max. 16

CPU 103



56

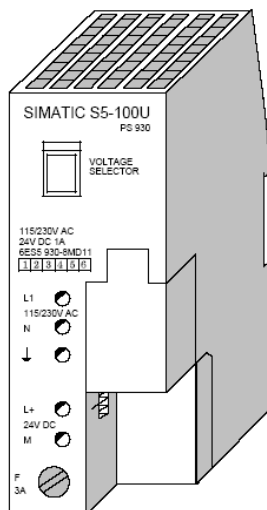
مشخصات فنی

Memory configuration	
- internal memory	RAM 10240 statements
- memory submodule	EPROM/EEPROM
Flags	
2048; 512 retentive	
Timers: Number/range	approx. 128; 0.01 to 9990 s
Counters: Number/range	128; 8 retentive
0 to 999 (up/down)	
Organization blocks	
OB1, 2, 13, 21, 22, 31, 34, and 251	
0 to 255	
Program blocks	
0 to 255	
Function blocks	
- programmable	0 to 255
- integrated	240 to 243, 250 and 251
Sequence blocks	
0 to 255	
Data blocks	
2 to 255	

آشنایی با ماژول های S5-100U

ماژول منبع تغذیه P.S

S5-100U دارای منابع تغذیه های مختلف از لحاظ جریان دهی می باشند. وظیفه این کارت دریافت برق شهر و تامین ولتاژ 24VDC می باشد.



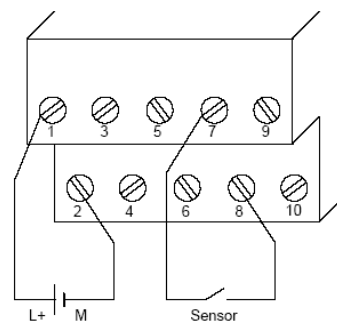
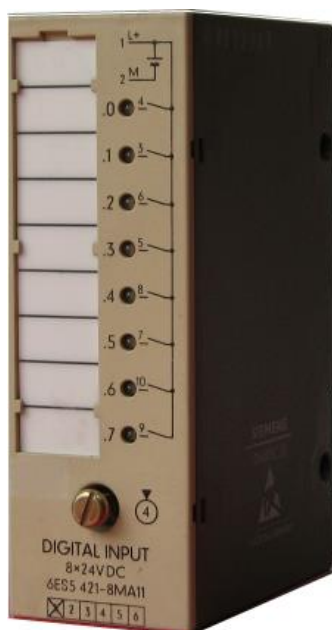
- PS 24V-1A
- PS 24V-2A
- PS 24V-2.5A

بر روی ماژول PS دو ترمینال با نام های L و N وجود دارد که محل اتصال تغذیه ورودی کارت (برق شهر) می باشد. دو ترمینال دیگر نیز جهت دریافت ولتاژ 24VDC وجود دارد که با نام های L و M مشخص شده است. همچنین از یک کلید دو حالت نیز جهت قطع و وصل تغذیه استفاده می شود.

ماژول ورودی دیجیتال DI

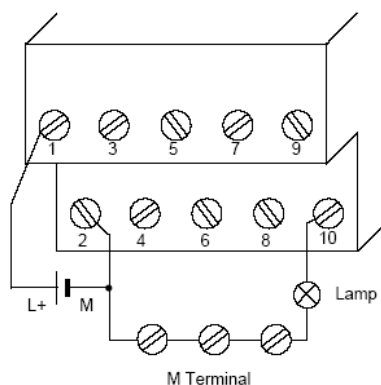
این ماژول سیگنالهای دیجیتال را از سطح پروسه دریافت می کند. این سیگنال ها می تواند از وسایلی مانند سنسورها ، شاستی ها و میکروسوییچ ها ارسال گردد. سطح سیگنال ولتاژ این ماژول ها می تواند 24VDC و 220VAC باشد که این ولتاژ را می توان از منبع تغذیه تامین نمود.

57

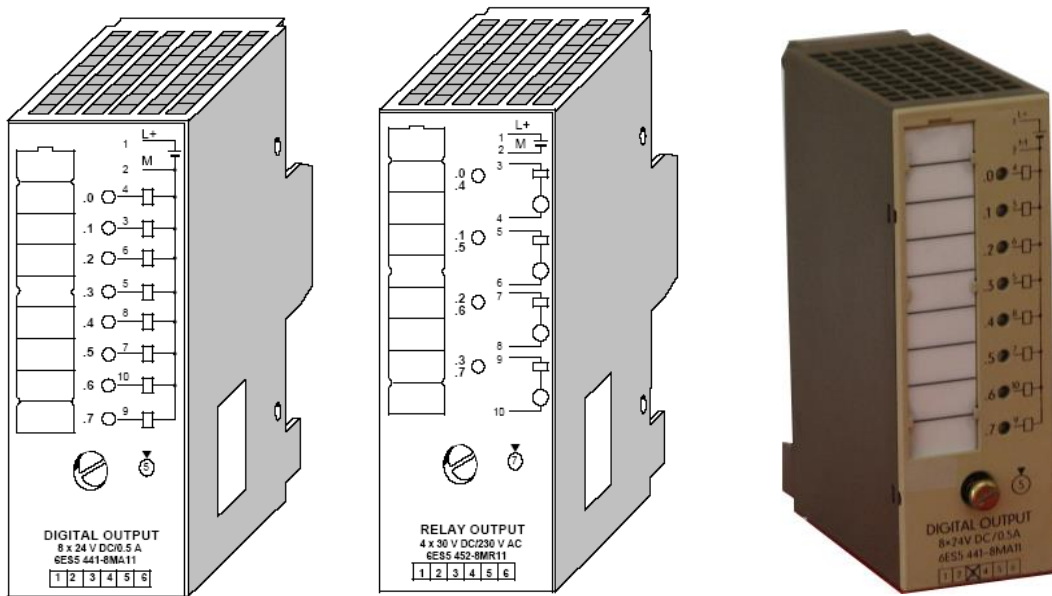


ماژول خروجی دیجیتال DO

فرمانهای PLC توسط این ماژول به سطح پروسه ارسال می شود. این فرمانها می تواند به وسایلی مانند شیرها ، کنتاکتورها ، رله ها و لامپ ها منتقل شود. خروجی این ماژول ها در دو نوع ترانزیستوری و رله ای ساخته می شوند.

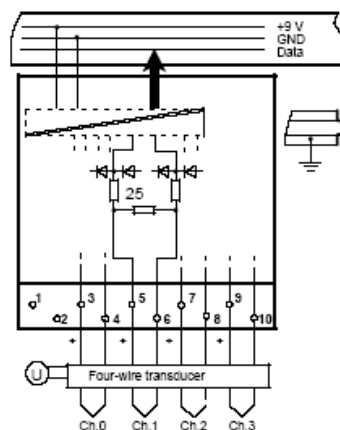


در شکل زیر ماژول های خروجی دیجیتال را مشاهده می کنید.



ماژول ورودی آنالوگ AI

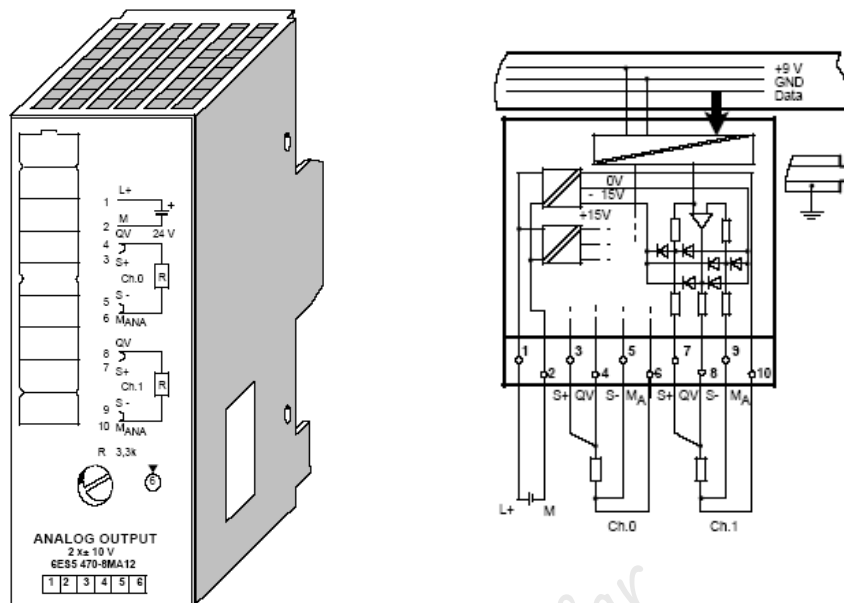
در پروسه های صنعتی کمیت هایی مانند دما ، وزن ، فشار و... به عنوان کمیت های آنالوگ به حساب می آیند. جهت اندازه گیری این کمیت ها از کارتهای ورودی آنالوگ استفاده می شود. سیگنالهای ارسالی توسط سنسورها به این کارتها می توانند به صورت ولتاژ و یا جریان باشند.



ماژول خروجی آنالوگ AO

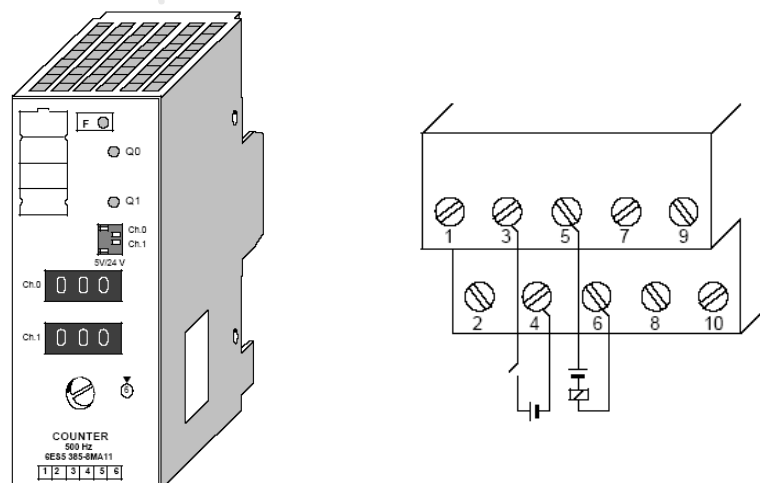
جهت ارسال سیگنال آنالوگ از PLC به سطح پروسه، از خروجی های آنالوگ استفاده می شود. این خروجی ها در حقیقت سطح سیگنال داخلی PLC که یک سیگنال دیجیتال می باشد را به سیگنال آنالوگ تبدیل می کنند. سطوح سیگنال خروجی عبارتند از:

4-20mA , 0-10VDC , 1-5VDC

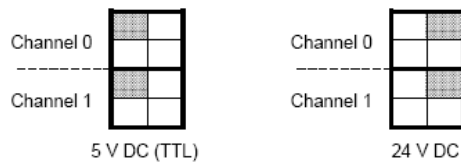


ماژول شمارنده (CM)

این کارت یک ماژول شمارنده 0 تا 500 هرتز 2 کاناله می باشد. این کارت قادر به دریافت پالسهای سرعت بالا از وسایلی مانند سنسورها و انکودرها می باشد. در ضمن سطح سیگنال پالس می تواند 5VDC و یا 24VDC باشد. جهت تعیین سطح سیگنال از یک سوئیچ دو حالتی بر روی این ماژول استفاده می شود.

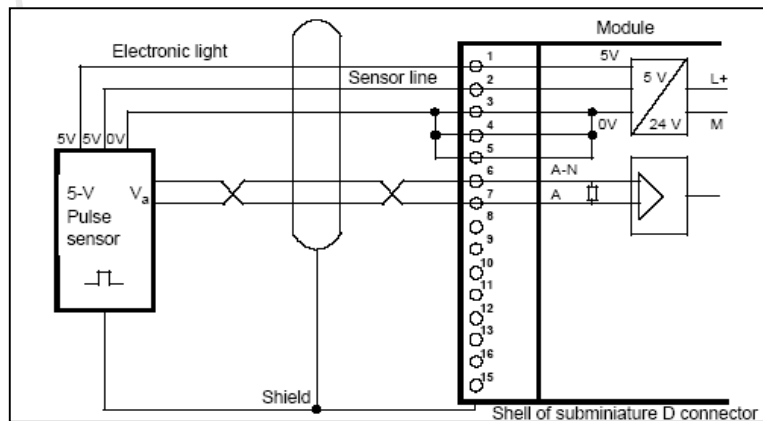
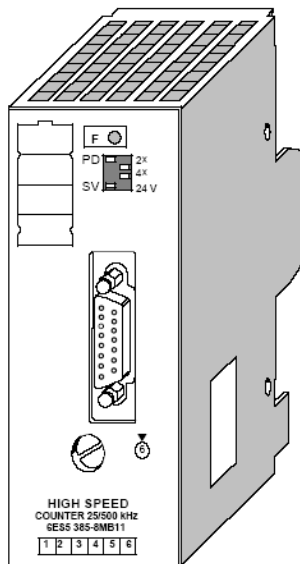


در شکل زیر سوئیچ دو حالتی نشان داده شده است.

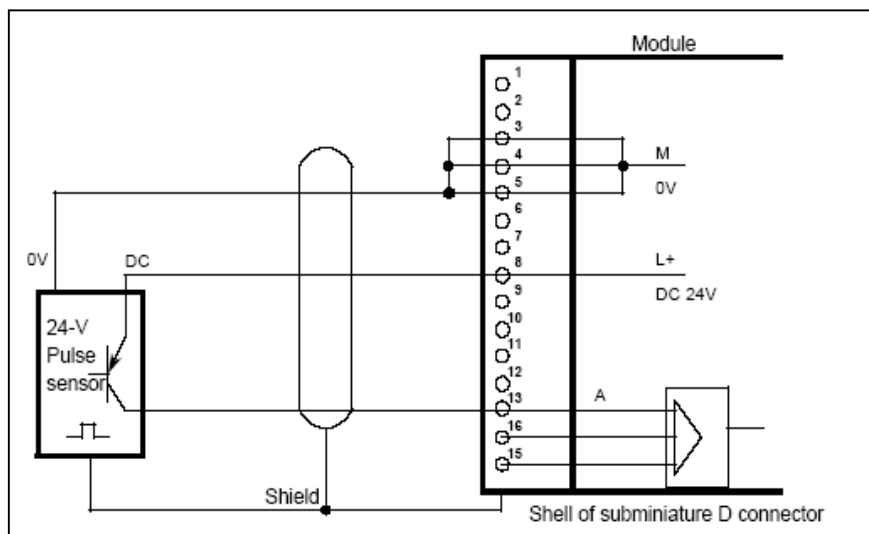


ماژول شمارنده (HSC)

این ماژول قادر به دریافت پالس هایی تا فرکانس 500KHZ می باشد. این کارت به کارت شمارنده های سرعت بالا معروف بوده و در اکثر PLC های زیمنس وجود دارد. یکی از کاربردهای مهم این ماژول جهت دریافت و شمارش پالس های ارسالی از یک انکودر می باشد.



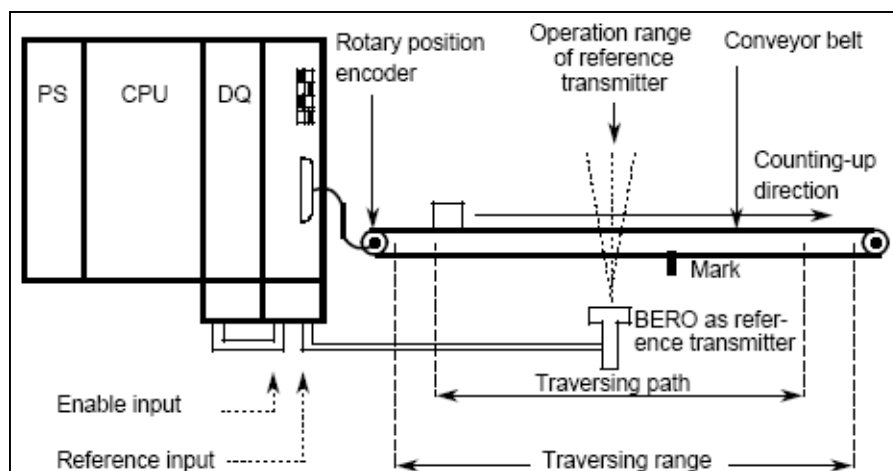
پالس 24V



یک نمونه از کاربرد کارت HSC

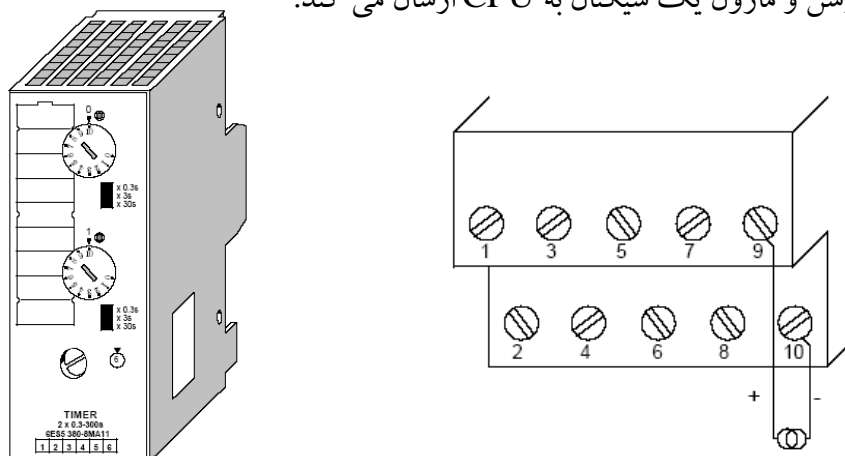
شکل زیر یک نوار نقاله را که قرار است قطعه ای را از نقطه A به B حمل کند نشان می دهد. در این ساختار از یک انکودر Rotary به همراه یک سنسور مجاورتی Proximity به عنوان نقطه مرجع استفاده شده است. این سنسور به ورودی دیجیتال این ماژول متصل شده است. از ورودی های دیجیتال ماژول HSC می توان به عنوان ورودی فعال ساز و یا تعیین نقطه مرجع استفاده نمود. همچنین این ماژول مجهز به خروجی دیجیتال نیز می باشد که توسط این خروجی ها می توان فرامین مورد نیاز را به مصرف کننده ها ارسال نمود. در این مثال جهت فرمان توقف و حرکت موتور می توان از خروجی های دیجیتال ماژول HSC استفاده نمود.

61



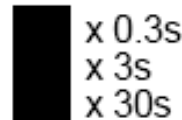
ماژول مقایسه کننده

این ماژول یک کارت مقایسه کننده بوده که دارای دو ورودی مقایسه می باشد. سیگنال مقایسه می تواند بصورت ولتاژ و یا جریان به این کارت ارسال شود. تعیین جنس سیگنال توسط سوئیچ دو حالت موجود بر روی کارت انجام می شود. همچنین بر روی این کارت دو سلکتور تنظیمی وجود دارد که توسط آنها می توان مقدار SP را برای انجام عملیات مقایسه تعیین نمود. زمانی که مقدار جاری به مقدار SP می رسد، نمایشگر موجود برای کانال مربوطه روشن و ماژول یک سیگنال به CPU ارسال می کند.



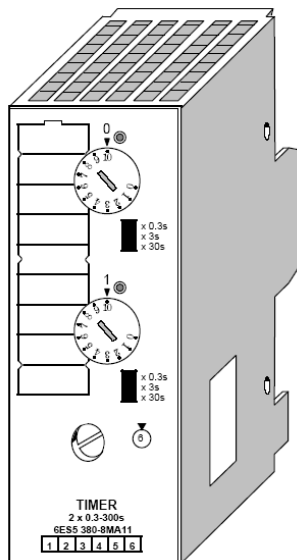
ماژول تایمر TM

این کارت دارای دو تایمر بوده که می تواند مستقل از CPU عمل زمان سنجی را انجام دهد. ماژول تایمر پس از سپری شدن زمان تنظیمی، سیگنالی را به CPU جهت اعلام اتمام زمان ارسال می کند. این واحد دارای یک سلکتور سه حالت با نام Time Rang جهت تعیین ضریب زمان می باشد. در ضمن از ولوم هایی با نام Scale Value نیز جهت تنظیم مقدار ضریب استفاده می شود.



62

به عنوان مثال فرض نمایید سلکتور را بر روی حالت 3s و ولوم را نیز بر روی عدد 7 قرار داده ایم. در این حالت زمان تایمر 21s تنظیم شده است.



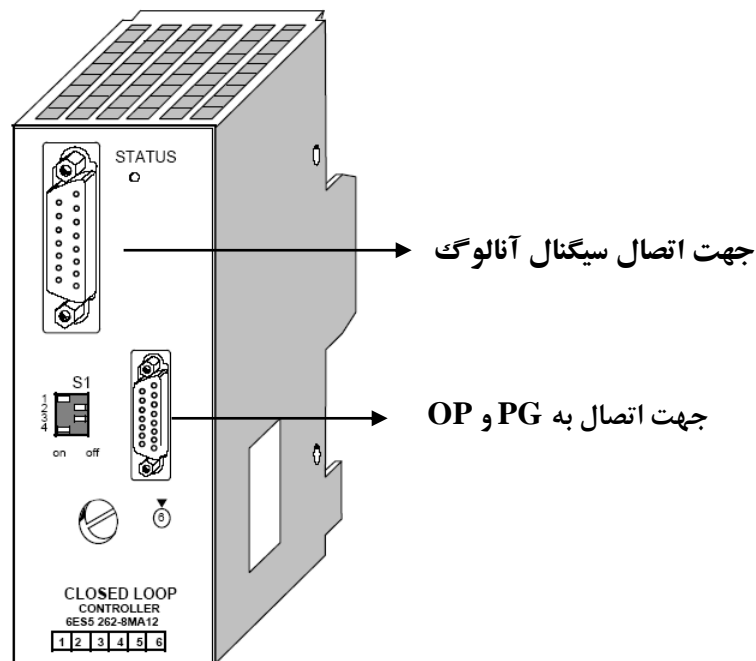
(time value=time range x scale value)

Example:	Time range:	x 3 s
	Scale value:	7
	Set time:	7 x 3 s=21 s

حداکثر زمان قابل تنظیم برای هر کانال ۳۰۰ ثانیه می باشد.

ماژول کنترل حلقه بسته IP 262

IP ها می توانند عملیاتی را مستقل از CPU انجام دهند. IP 262 در کنترل های حلقه بسته یا همان PID کاربرد دارد. این کنترل می تواند برای کمیت هایی مانند دما، فشار، رطوبت به کار گرفته شود.

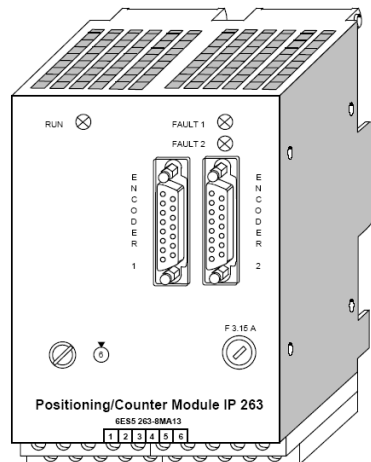


وسط سوئیچ های موجود بر روی این کارت نیز می توان جنس سیگنال ورودی را تعیین نمود. این سیگنال می تواند بصورت ولتاژ و یا جریان و همچنین ترموکوپل یا PT100 باشد.

در ضمن لازم است این نکته را نیز یادآور شویم که تمامی ماژول های بررسی شده و ماژول های IP می توانند به PLC های سری 90U و 95U نیز متصل شوند.

ماژول کنترل موقعیت IP 263

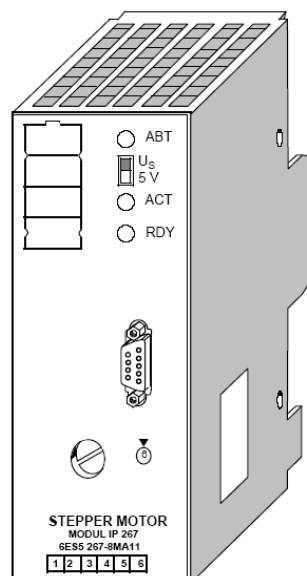
این ماژول یک ماژول کنترل موقعیت برای دو محور بوده که دارای پورت هایی جهت اتصال انکودر می باشد. در این ماژول ۴ خروجی دیجیتال به هر کانال جهت کنترل درایو اختصاص داده شده است.



64

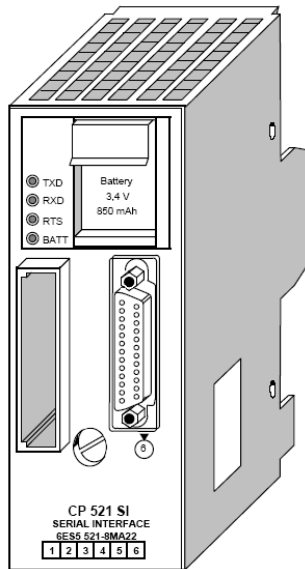
ماژول کنترل موتور پله ای IP 267

این کارت جهت کنترل موتور پله ای مورد استفاده قرار می گیرد. در واقع این ماژول پالس های کنترلی مورد نیاز را جهت گردش موتور پله ای برای درایو موتور پله ای فراهم می کند.



۱-۲۸ ماژول CP 521 SI

این ماژول یک واحد مجهز به پورت سریال می باشد که از آن می توان در ارتباط یکطرفه و یا دوطرفه استفاده نمود. توسط این پورت می توان با تجهیزات زیر ارتباط برقرار کرد:



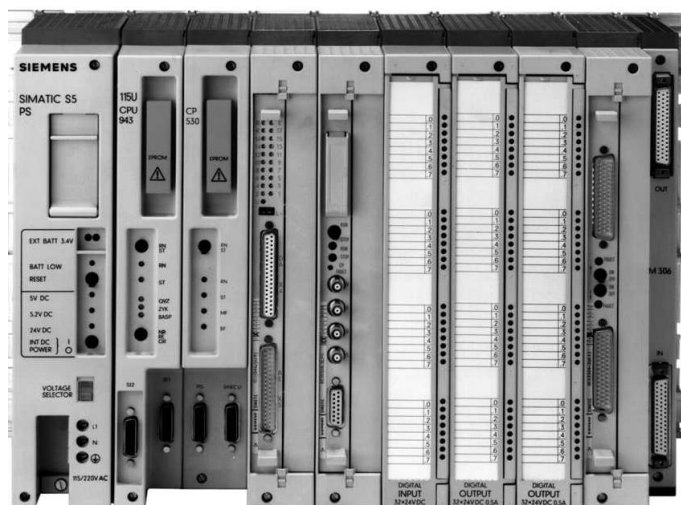
- ۱- پرینتر
- ۲- بارکد خوان
- ۳- صفحه کلید
- ۴- S5-95U
- ۵- CP 534
- ۶- CPU 944

65

آشنایی با S5-115U

این مدل به عنوان یکی از پرکاربردترین خانواده S5 می باشد. این سری با قابلیت های فراوان می تواند در پروسه های متوسط و بعضا بزرگ مورد استفاده قرار گیرد. در این مدل تمامی واحدها اعم از CPU و PS بر روی رک مرکزی نصب شده و رک مرکزی نیز به راحتی می تواند توسط کارت IM با رک های توسعه یافته ارتباط برقرار کند. تمامی کارتهایی که در قسمتهای قبل بررسی گردید در این سری نیز وجود دارد.

چند مورد از ویژگیهای این CPU نسبت به مدل های قبلی:

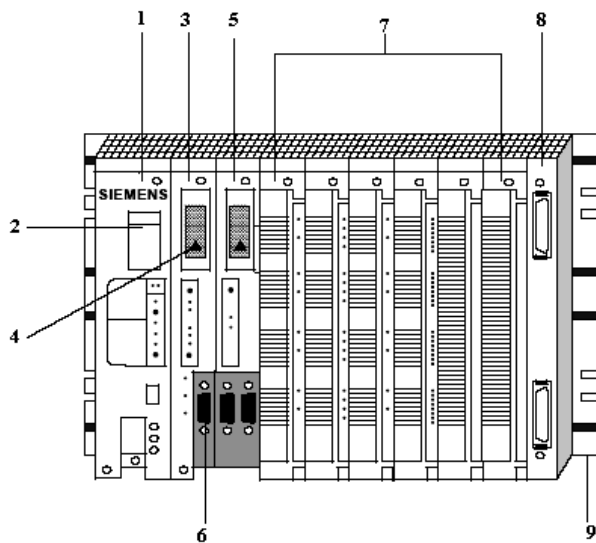


- *سرعت بالای پردازش
- *حجم زیاد فضای حافظه
- *دارا بودن ساختاری کاملا رک

انواع CPU در S5-155U

- 941*
- 942*
- 943*
- 944*
- 945*

در شکل زیر قسمت‌های یک S5-115U را مشاهده می‌نمایید.



- ۱- واحد منبع تغذیه PS
- ۲- محل قرارگیری باطری پشتیبان
- ۳- واحد پردازنده مرکزی CPU
- ۴- محل قرارگیری EEPROM
- ۵- ماژول CP
- ۶- پورت‌های چند منظوره
- ۷- ماژول‌های ورودی و خروجی
- ۸- ماژول IM
- ۹- رک Rack

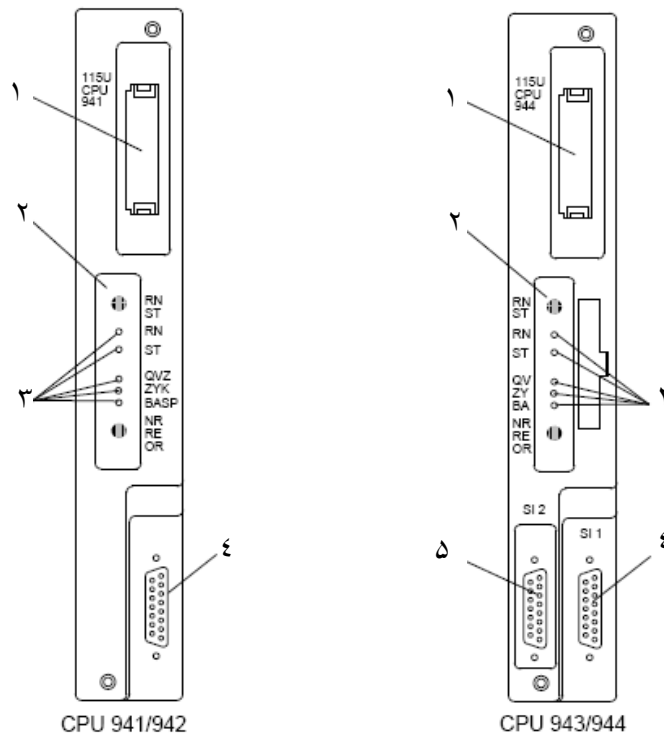
واحد پردازنده مرکزی CPU

همانطور که در بحث قبلی بیان شد S5-115U دارای چندین مدل CPU می‌باشد که هر کدام از آنها دارای قابلیت‌هایی می‌باشند.

جدول شکل زیر مشخصات هر CPU را به طور کامل نشان می‌دهد.

	CPU 941	CPU 942	CPU 943	CPU 944
Execution time per - 1000 statements (see Appendix A for specific information)	Approx. 10 msec.	Approx. 10 msec.	Approx. 5 msec.	Approx. 1.5 msec.
Internal program memory (RAM)	2 Kbytes	10 Kbytes	48 Kbytes	96 Kbytes
Total program memory, maximum	18 Kbytes*	42 Kbytes*	48 Kbytes	96 Kbytes
Cycle monitoring time	Default approx. 500 ms, programmable			
Program scanning	Cyclic, interrupt-driven, time-controlled			
Address range, maximum (digital inputs)	1024 I 0.0 to I 127.7			
Address range, maximum (digital outputs)	1024 Q 0.0 to Q 127.7			
Address range, maximum (analog inputs)	64 PW 128 to PW 254			
Address range, maximum (analog outputs)	64 PW 128 to PW 254			
Flags	1024, optionally	<ul style="list-style-type: none"> • all retentive • half retentive • all non-retentive 		
Timers	128, optionally	<ul style="list-style-type: none"> • all retentive • half retentive • all non-retentive 		
Counters	128, optionally	<ul style="list-style-type: none"> • all retentive • half retentive • all non-retentive 		
Time range	0.01 to 9990 s			
Counting range	0 to 999			
Operation set	Approx. 170 operations			

آشنایی با انواع CPU در S5-115U

**CPU 941/942**

- ۱- محل قرارگیری کارت حافظه
- ۲- سوئیچ تغییر مد کاری CPU
- ۳- LED های وضعیت CPU
- ۴- پورت OP، PG، و یا شبکه SINEC L1

CPU 943/944

- ۱- محل قرارگیری کارت حافظه
- ۲- سوئیچ تغییر مد کاری CPU
- ۳- LED های وضعیت CPU
- ۴- پورت S1
- ۵- پورت S2

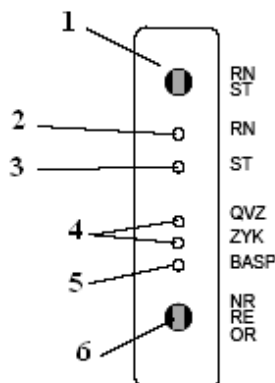
رابط های S1 و S2

به پورت سریال S5-115U وسایلی مانند پروگرامر، OP و مانیتور متصل می شوند. شبکه SINEC L1 نیز می تواند به این درگاه متصل شود. CPU های 943 و 944 دارای دو رابط سریال می باشند.

موارد استفاده Serial Interface :

- * اتصال پروگرامر
- * پانل اپراتوری
- * صفحه کلیدها و پرینترهای صنعتی
- * ارتباط به صورت PTP

بر روی CPU دو کلید چند حالت به همراه چندین LED وجود دارد که عملکرد هر یک در قسمت زیر بیان شده است.



- ۱- سوئیچ دو حالت RUN/STOP
 - ۲- LED نشان دهنده وضعیت RUN
 - ۳- LED نشان دهنده وضعیت STOP
 - ۴- LED های نشان دهنده خطا
 - ۵- LED نشان دهنده غیرفعال بودن خروجیها
 - ۶- کلید سه حالت RESTART
- نمایشگر QVZ (Time-out) :

اگر CPU با یک تاخیر مشخص سیگنال Ready (آماده به کار بودن) را از ماژول ها دریافت نکند، این LED به نشانه عدم دسترسی به ماژول های I/O و یا آدرس دهی نامناسب، روشن می شود. در جدول شکل زیر زمان تاخیر این سیگنال برای ماژول های مختلف مشخص می باشد.

نمایشگر ZYK

اگر زمان مرور برنامه از حد مجاز تجاوز کند، این LED به نشانه خطا روشن می گردد. به عنوان مثال اگر در برنامه از دستورات پرش و پایان جهت ایجاد حلقه استفاده و این حلقه به یک حلقه بی نهایت تبدیل شود، در این صورت زمان اجرای برنامه از حد مجاز تجاوز کرده و برنامه هیچ گاه به پایان نمی رسد.

نمایشگر BASP

زمانی که CPU در مد STOP می باشد، تمامی خروجی ها غیرفعال می باشند. این نمایشگر به نشانه غیرفعال بودن خروجی ها روشن می گردد. سیگنال BASP تنها در صورتی که بلوک های OB21 و OB22 اجرا شوند، غیرفعال می شود.

سوئیچ چند حالت جهت ری استارت CPU

وضعیت NR و RE: این دو حالت بعد از انجام یک Overall Reset وضعیت و محتوای فضاهای پایدار و ناپایدار حافظه را مشخص می کند. اگر بعد از انجام یک ریست سرد کلید در وضعیت RE قرار گیرد، تایمرها، کانترها و فلگ های پایدار مقادیرشان حفظ می گردد.

Switch Position	Flags	Timers	Counters
RE (retentive)	F 0.0 to F 127.7 retentive	T0 to T63 retentive	C0 to C63 retentive
	F128.0 to 255.7 nonretentive	T64 to T127 nonretentive	C64 to C127 nonretentive

در وضعیت RE نیمی از فضاهای نشان داده شده در جدول فوق پایدار و نیمی دیگر ناپایدار می باشند. فضاهای پایدار بعد از انجام یک Overall Reset به مقادیر پیش فرض باز می گردند یا در واقع اطلاعاتشان از بین نمی رود.

تمامی تایمرها، کانترها و فلگ ها در حالتی که سوئیچ در وضعیت NR قرار گیرد، ناپایدار محسوب می شوند.

Switch Position	Flags	Timers	Counters
NR (nonretentive)	No retentive flags	No retentive timers	No retentive counters

بعد از یک Overall Reset موارد زیر پاک می شوند:

۱- برنامه موجود در حافظه PLC

۲- تمامی دیتاهای موجود در فلگ ها، تایمرها و کانترها

۳- تمامی ID های مربوط به خطاها

نحوه انجام یک Overall Reset :

- ۱- تغذیه PLC را وصل نمایید.
- ۲- CPU را در حالت STOP قرار دهید.
- ۳- سوئیچ را به وضعیت OR برده و زمانی که نمایشگر STOP به صورت لحظه ای خاموش شد سوئیچ را رها کنید. در ادامه سوئیچ به صورت اتوماتیک به وضعیت RE برمی گردد. در این صورت برای حافظه داخلی PLC و کارت حافظه یک Overall Reset رخ داده است. بعد از انجام این مراحل CPU مجدداً سخت افزار و فضای حافظه خود را تست می کند.

دانلود نمودن برنامه به CPU به دو روش امکان پذیر می باشد:

- ۱-دانلود برنامه به کارت حافظه و انتقال آن به حافظه داخلی CPU
- ۲-دانلود برنامه به صورت مستقیم به حافظه داخلی CPU

در حالت اول توسط یک پکیج پروگرامر S5-DOS می توان برنامه را به کارت حافظه منتقل نمود. بعد از ریختن برنامه به کارت حافظه:

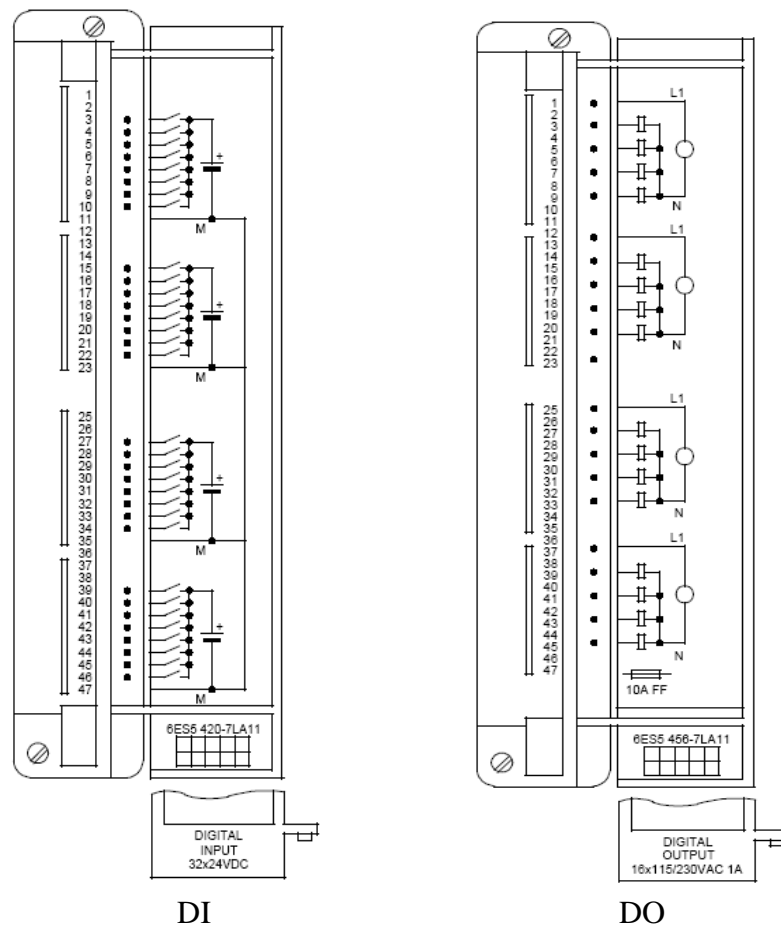
- ۱-کارت را در حالت خاموش بودن CPU در محل تعبیه شده وارد می کنیم.
- ۲-تغذیه را وصل می کنیم.
- ۳-توسط سوئیچ ری ست یک Overall Reset انجام می دهیم.

بعد از انجام Overall Reset سیستم به صورت اتوماتیک برنامه موجود در کارت را به حافظه داخلی CPU منتقل می کند. در روش دوم نیاز به نرم افزار ، کابل ارتباطی و دستگاه برنامه ریز می باشد.

ورودی و خروجی های دیجیتال

ماژول های I/O در این سری نیز جهت اتصال به سنسورها و محرک ها استفاده می شود. کارتهای ورودی و خروجی در این سری دارای ویژگیهای زیر می باشند:

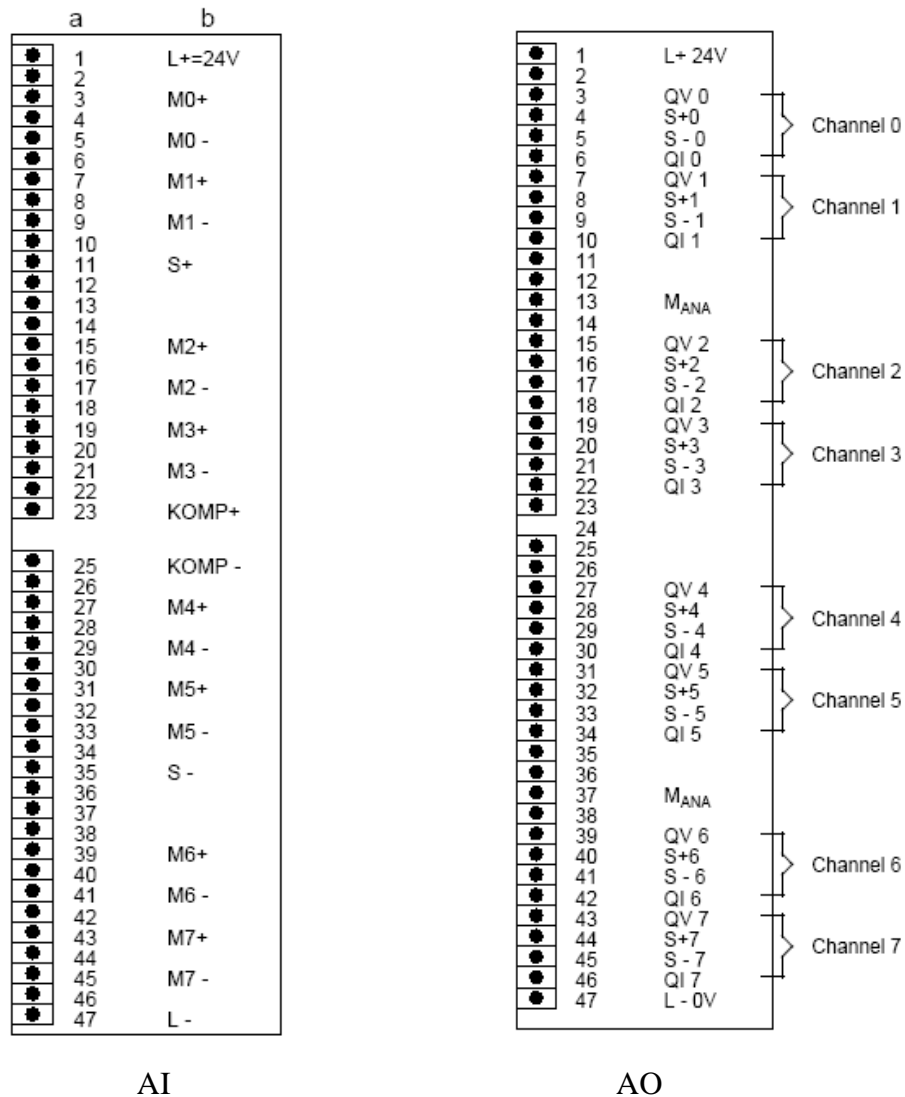
- ۱- نصب سریع
- ۲- دارا بودن کدسوئیچ مکانیکی
- ۳- دارا بودن برچسب های بزرگ در قسمت جلوی کارت



ورودی و خروجی های آنالوگ

کارت‌های مربوط به سیگنال‌های آنالوگ نیز همانند سایر کارت‌ها در انواع مختلف وجود دارند. سیگنال‌های استاندارد برای ورودی آنالوگ می‌تواند در بازه‌های زیر باشد:

میلی ولت ولت میلی آمپر ترمومترها ترموکوپل‌ها



I/O های هوشمند در S5 115U

همانطور که در بحث قبلی بیان شد، IP ها ماژول‌هایی هستند که وظایف خاصی را مستقل از CPU اجرا می‌کنند و با اصطلاح باری را از دوش CPU برمی‌دارند. این ماژول‌ها خود مستقیماً درگیر کار شده و می‌توانند با CPU تبادل اطلاعات داشته باشند. IP ها می‌توانند به عنوان یکی از ماژول‌های S5-115U انجام وظیفه کنند.

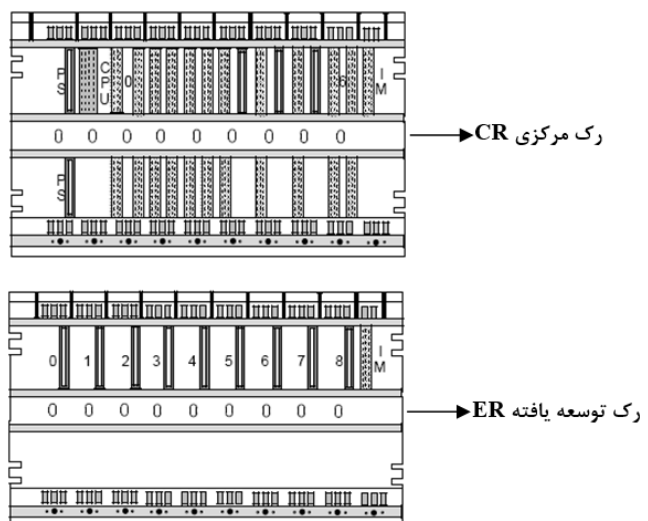
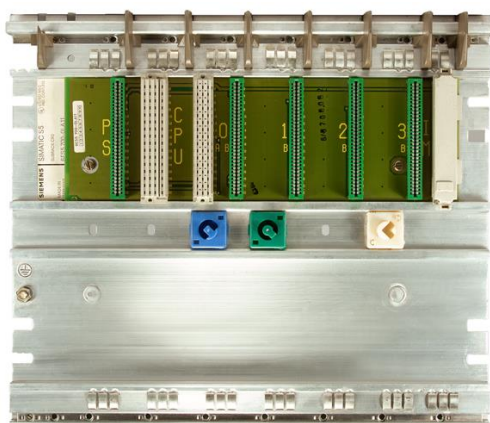
IP 240 counter and position decoder
IP 241 digital position decoder
IP 242 counter module
IP 243 analog module
IP 244 temperature control module
IP 245 valve control module
IP 246 positioning module
IP 247-4UA11 IP 247-4UA21 positioning module
IP 252 closed-loop control module
WF 625 positioning module
IP260 closed-loop control module
IP 261 proportioning module

در قسمت زیر به برخی از وظایف IP ها اشاره شده است:

- شمارش سریع پالس ها
- مکان یابی و کنترل مکان
- اندازه گیری و کنترل سرعت و زمان
- کنترل و اندازه گیری دما و ...

رک در S5-115U

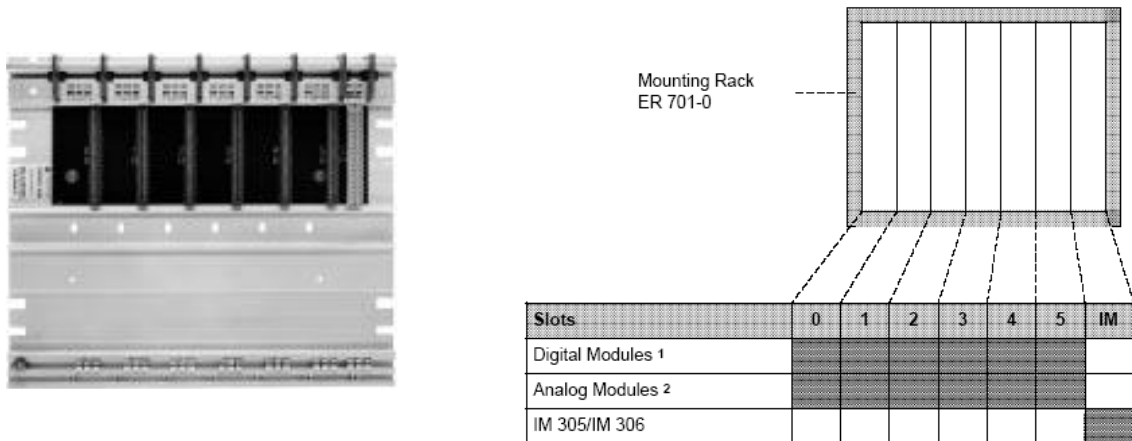
تمامی واحدهای S5-115U بر روی رک قرار می گیرند. در شکل زیر یک نمونه رک مرکزی را ملاحظه می کنید.



در قسمت زیر به بررسی چند نمونه از رک های ER و CR در S5-115U می پردازیم.

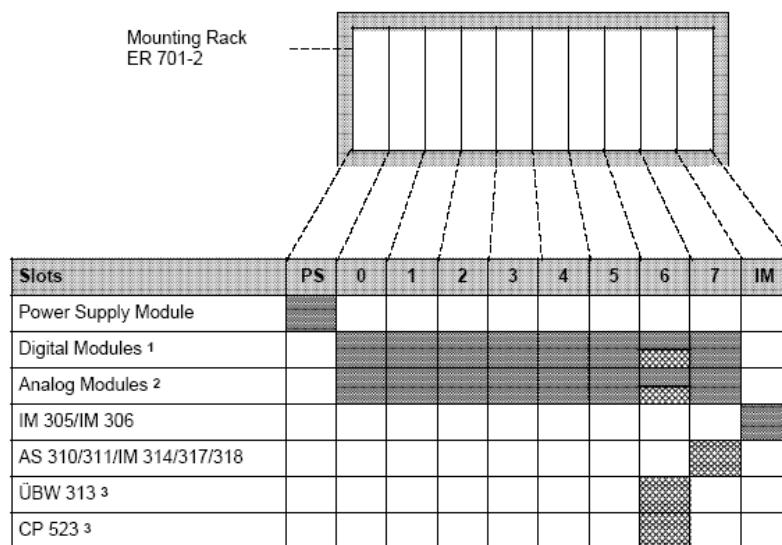
ER 701-0 , ER 701-1, ER 701-2 , ER 701-3

در شکل زیر ER 701-0 را ملاحظه می نمایید.

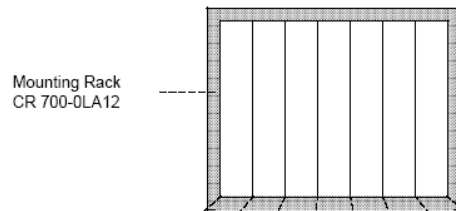
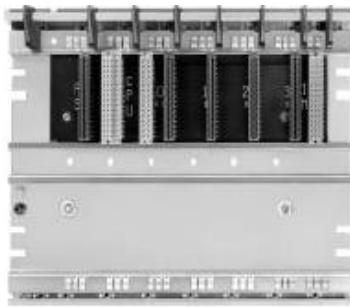


همانطور که در شکل فوق ملاحظه می کنید، در این واحد فقط کارتهای I/O قرار می گیرند. Slot آخر این EU می تواند یک IM 305/IM306 باشد.

در شکل زیر نمونه ای دیگر از این رک ها را ملاحظه می کنید.(ER701-2)



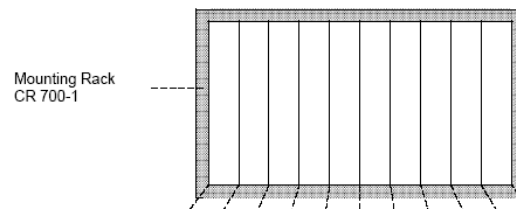
رک مرکزی CR 700-0



Slots	PS	CPU	0	1	2	3	IM
Power Supply Module	■						
Central Processing Unit		■					
Digital Module 1			■	■	■	■	
Analog Module 1			■	■	■	■	
IM 305/IM 306 2							■
CP 530-7			■				
CP 523			■				
IP 240/241/242 /243 /244/260/261			■				
DIMOS Interface Module			■				

75

رک مرکزی CR 700-1



Slots	PS	CPU	0	1	2	3	4	5	6	IM
Power Supply Module	■									
Central Processing Unit		■								
Digital Modules 1			■	■	■	■	■	■	■	
Analog Modules 2			■	■	■	■	■	■	■	
IM 305 / IM 306 3										■
CP 530-7			■							
CP 523			■							
IP 240/241 ... 245/260/261			■							
WF 625/725			■							
DIMOS Interface Module			■							

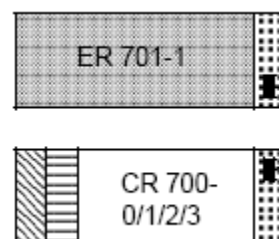
ماژول واسط (IM)

IM ها برای ارتباط بین رک مرکزی با رکهای توسعه و یا ارتباط بین رکهای توسعه به کار می روند. در صورت نیاز به اضافه نمودن واحدهای دیگر ورودی و خروجی به PLC از این رابط استفاده می شود. اگر در فرآیندهای صنعتی تعداد I/O های سیستم گسترش یابند، نصب کلیه ماژول ها در یک رک امکان پذیر نمی باشد. در چنین مواردی لازم است تعداد رک ها گسترش یابند. مبادله اطلاعات بین رک اصلی و رک توسعه یافته از طریق ماژول های واسط صورت می گیرد. در واقع جهت متصل نمودن رک ها به یکدیگر از این رابط یعنی IM می بایست استفاده نمود. کارتهای CPU، منبع تغذیه و I/O ها در رک اصلی با نام Central Controller که به اختصار CC می نامند، قرار می گیرند. رکهای توسعه یافته (Expansion Unit) را به اختصار با حرف EU نمایش می دهند. به عنوان مثال در PLC S5-115U از IM 306 می توان جهت ارتباط CC با EU استفاده نمود. کارتهای IM دارای دو پورت ارتباطی IN و OUT می باشند. عملیات ارتباط بین رک ها از طریق این پورتها، و توسط کابل های مخصوص صورت می گیرد.

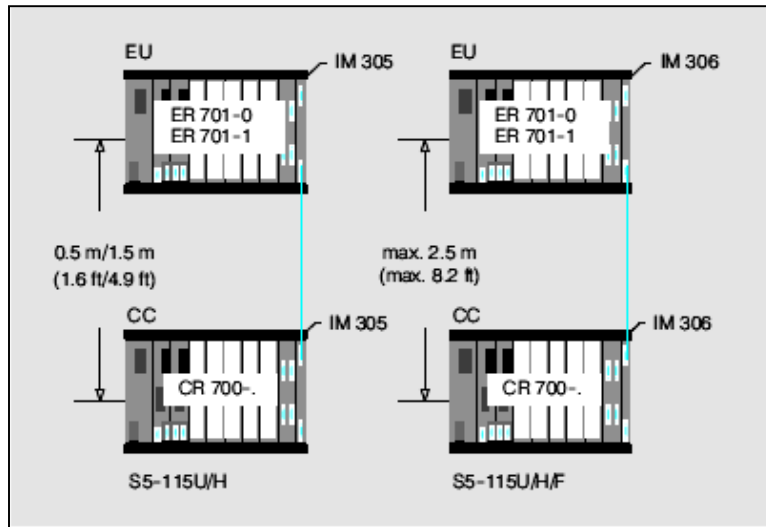
76



IM305: تنها با یک ER می توان توسط این IM ارتباط برقرار نمود. طول مجاز کابل در این IM می تواند بین 0.5 تا 1.5 متر و محل آن در رک CR و ER ثابت می باشد.

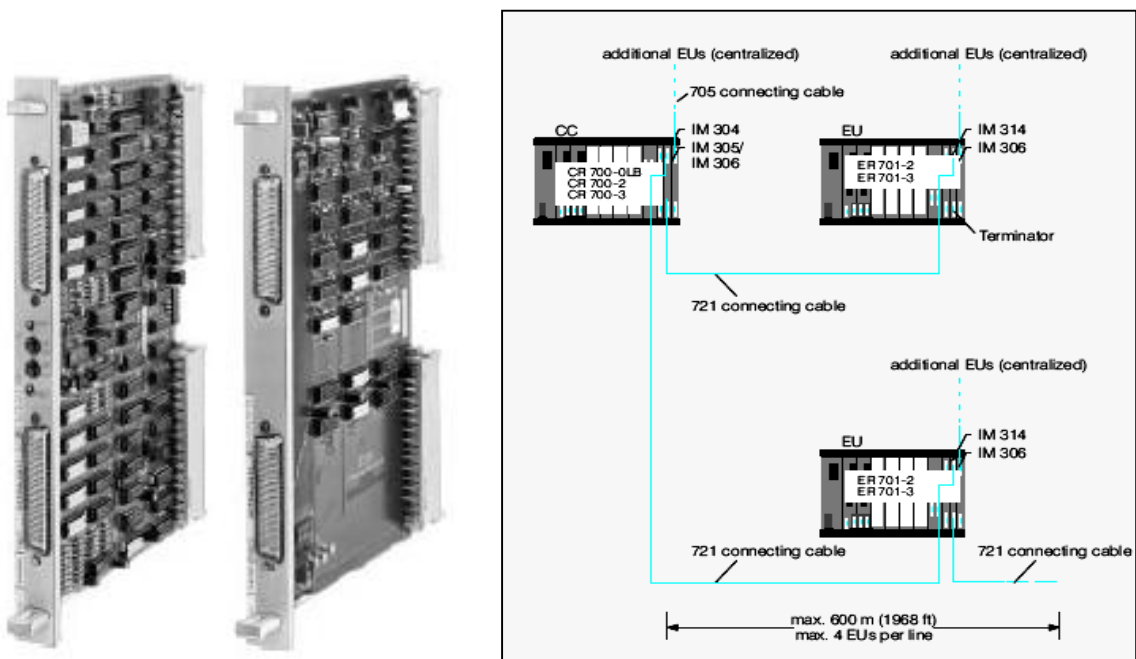


IM306 : توسط این IM با ۳ رک افزایشی (3ER) می توان ارتباط برقرار نمود و حداکثر طول کابل می تواند 2.5 متر باشد.



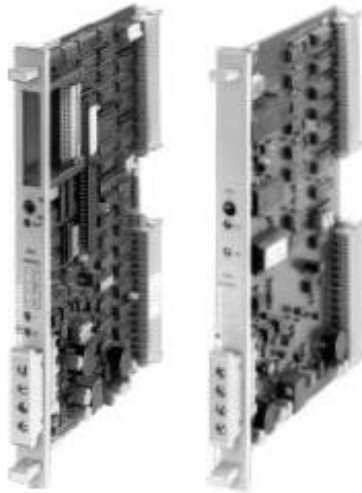
IM314 و IM304

با استفاده از این دو IM طول کابل می تواند حداکثر تا ۶۰۰ متر باشد. در IM314 در ER و در IM304 در CR قرار می گیرد.



IM318 و IM308

با استفاده از این دو IM طول کابل می تواند حداکثر ۳۰۰۰ متر باشد. در IM308 در CR و در IM318 در ER قرار می گیرد. همچنین توسط این رابط ها با توجه به ساپورت نمودن مسافت طولانی، امکان اتصال به ایستگاه I/O های توسعه یافته یا همان ET100U نیز وجود دارد.

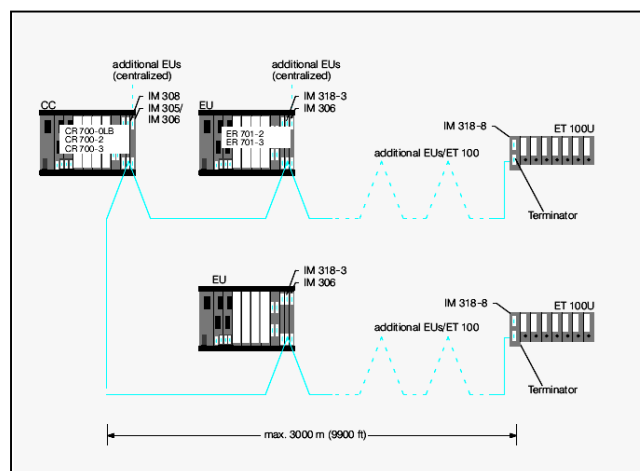


78

در صورت استفاده از ایستگاه های ET100U می بایست از IM318-8 در ایستگاه ET100U استفاده نمود. توسط این IM ها حداکثر می توان با ۶۳ ایستگاه ارتباط برقرار نمود. در واقع از هر رابط ارتباطی به ۳۲ ایستگاه می توان وصل شد.

Technical specifications			
Transmission speed (selectable)		31, 62, 187 or 375 kbit/s	
Number of units that can be connected			
• Per connection	max.	32	
• Total	max.	63	
Addressing range for I/O modules	max.	1 Kbyte	
Current consumption (at 5 V)			
• IM 308	max.	0.5 A	
• IM 318-3	max.	0.3 A	
Power loss			
• IM 308	max.	2.5 W	
• IM 318-3	max.	1.5 W	
Space requirements			
			1 slot or 1/2 slot (see subracks, starting page 3/58)
Weight	approx.		0.4 kg (0.9 lb)

در شکل زیر نحوه ارتباط رک مرکزی با سایر رک های توسعه یافته و همچنین ایستگاه ET100U را ملاحظه می کنید.



ماژول منبع تغذیه در S5 115U

PS در S5-115U نیز همانند سایر مدل ها با دریافت برق شهر، ولتاژ مورد نیاز PLC را تامین می کند. در شکل زیر ماژول PS 951 را ملاحظه می نمایید.

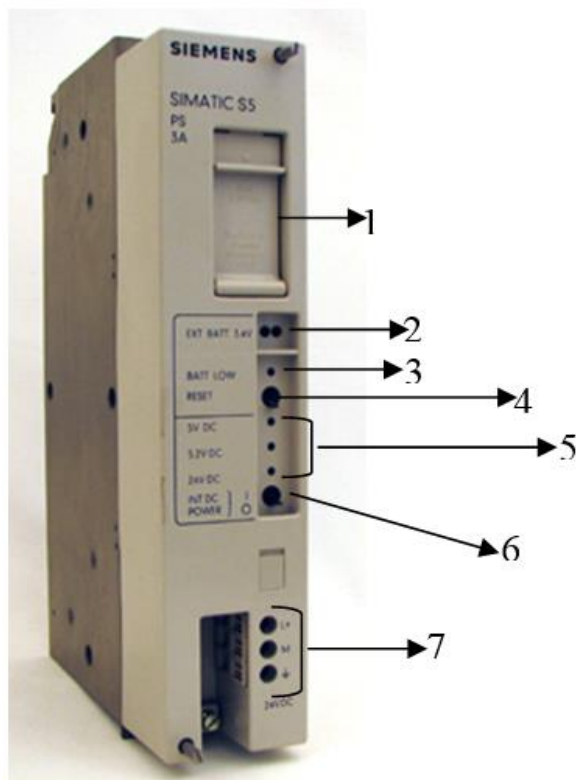


79

جریان دهی کارتهای PS نیز با یکدیگر متفاوت می باشد.

Power supply module	6ES5 951-7LB21	6ES5 951-7LD21
Supply voltage		
• Rated value	230/120 V AC	230/120 V AC
• Ripple V_{pp} max.	—	—
• Permissible range (including ripple)	187 ... 264 V 94 ... 132 V	187 ... 264 V 94 ... 132 V
• Frequency range	47 ... 63 Hz	47 ... 63 Hz
Output current¹⁾		
• Rated value for operation		
Without fan	3 A	7 A
With fan	3 A	15 A
• Permissible range	0.3 ... 3 A	0.3 ... 15 A
Backup battery	1 x lithium battery Size C, (3.6 V/5 Ah)	2 x lithium battery Size AA, (3.6 V/2 x 1.75 Ah)
• Backup time min.	1 year (at 0.3 mA, 25 °C and uninterrupted)	
• External backup voltage	+ 3.4 ... + 9 V	

در ادامه، قسمت‌های مختلف کارت PS مورد بررسی قرار گرفته است.



۱- محل قرارگیری باطری BACK UP

۲- ترمینال جهت اعمال ولتاژ خارجی -۳، ۴ تا ۹ ولت DC - زمانی که باطری در حال تعویض و یا منبع تغذیه قطع است.

۳- LED مربوط به وضعیت باطری

دلایل روشن شدن LED مربوط به باطری:

- باطری وجود ندارد.

- نصب باطری به درستی انجام نشده است.

- ولتاژ باطری از حد مجاز (2.8 V) کمتر است.

۴- کلید RESET

یکی از کاربردهای این سوئیچ بعد از تعویض باطری می باشد.

۵- LED های نشان دهنده سلامت تغذیه برای ولتاژهای زیر:

5 V

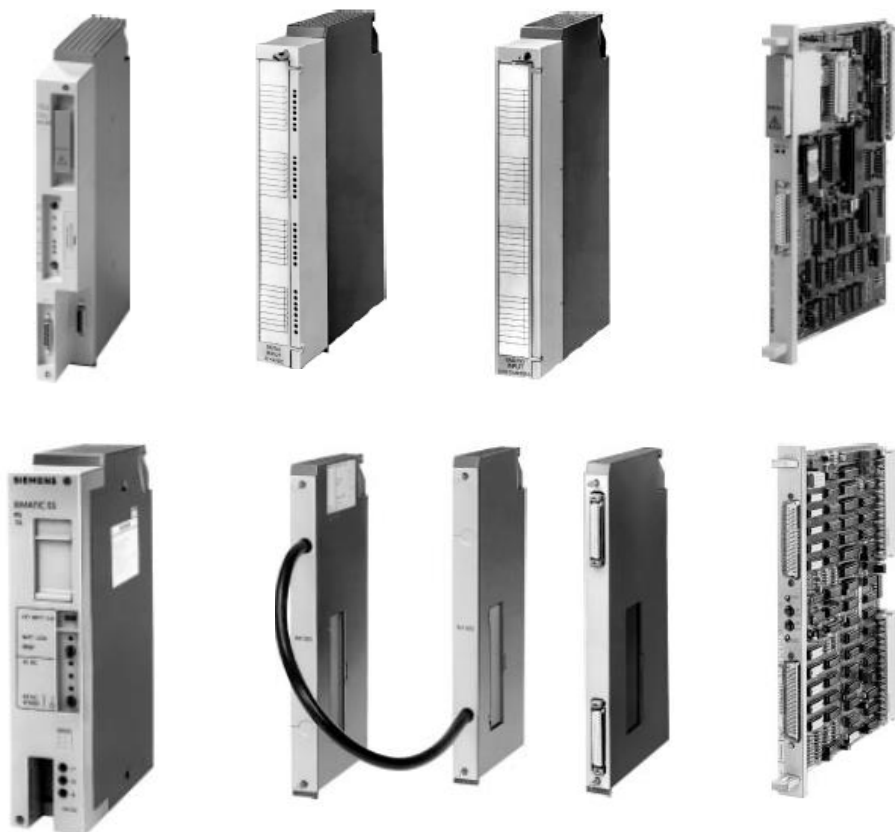
5.2 V

24 V

۶- سوئیچ مربوط به قطع و وصل تغذیه

۷- ترمینالهای مربوط به اتصال تغذیه

در شکل زیر کارتهای PLC S5-115U نشان داده شده است.

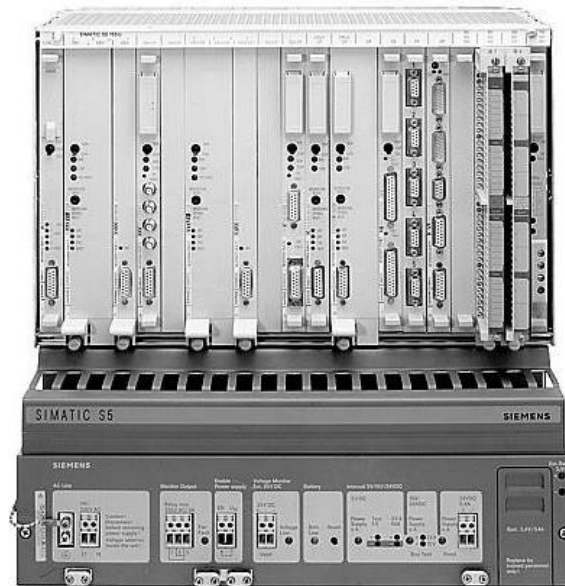


81

آشنایی با S5-135U و S5-155U

S5-155U, 135U آخرین و پر قدرترین مدل از خانواده STEP 5 شرکت زیمنس می باشند. به دلیل اینکه اکثر مباحثی که در مورد سری 155U بررسی می شود با سری 135U شباهت بسیاری دارد، لذا در ادامه در مورد ماژول ها و سخت افزار 155U که به عنوان آخرین مدل S5 می باشد بحث می گردد. S5-155U با قابلیت های فراوان و کارایی بالا بسیار، مناسب برای کنترل پروسه های صنعتی بزرگ و پیچیده می باشد.

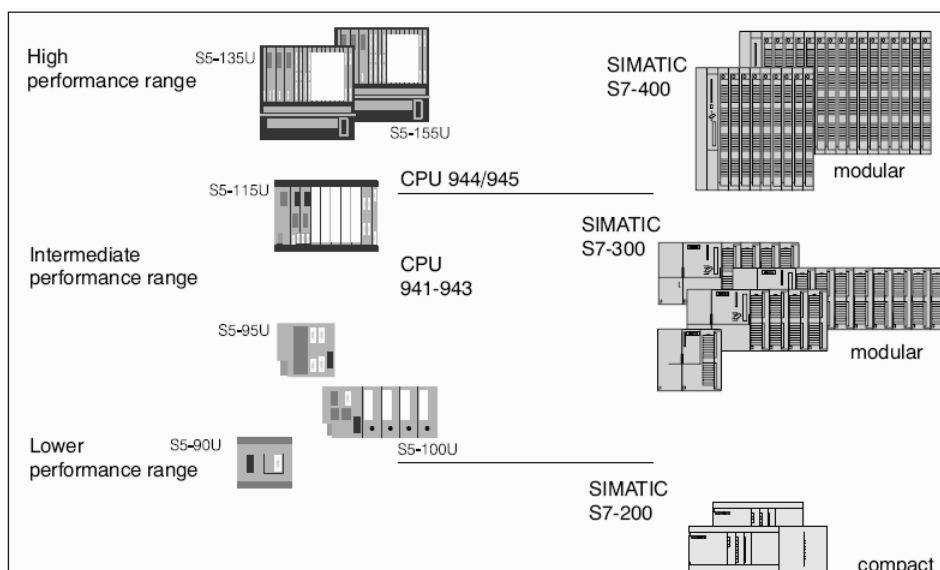
این مدل نیز دارای CPU های مختلف بوده که CPU 948 به عنوان پر قدرت ترین CPU در این خانواده می باشد.



انواع CPU در S5-155U

- CPU 928A
- CPU 928B
- CPU 946
- CPU 947
- CPU 948

S5-155U با کارایی بالا و دارا بودن پروسسور بسیار قدرتمند می تواند در پروسه های بزرگ به کار گرفته شود. شرکت زیمنس این دو مدل از S5 را در رده S7-400 قرار داده است.



کاربردها و ویژگیهای S5-155U

* در کاربردهای که نیاز به سرعت بسیار بالا جهت پردازش و کنترل سیستم های حلقه بسته می باشد . به عنوان مثال در صنایع خودرو سازی

* در کاربردهایی که نیاز به برقراری ارتباط با سرعت بالا بین چندین ایستگاه می باشد. به عنوان مثال برقراری ارتباط با کامپیوترهای اتاق های کنترل جهت کنترل ، نظارت و مانیتورینگ پروسه

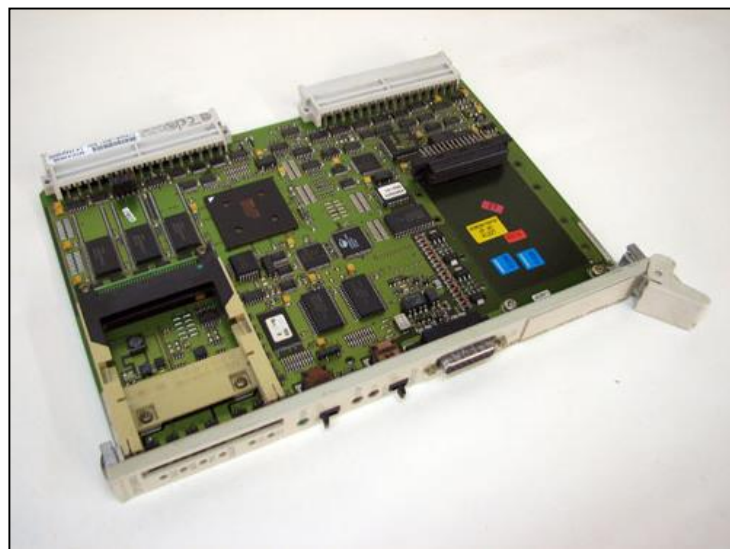
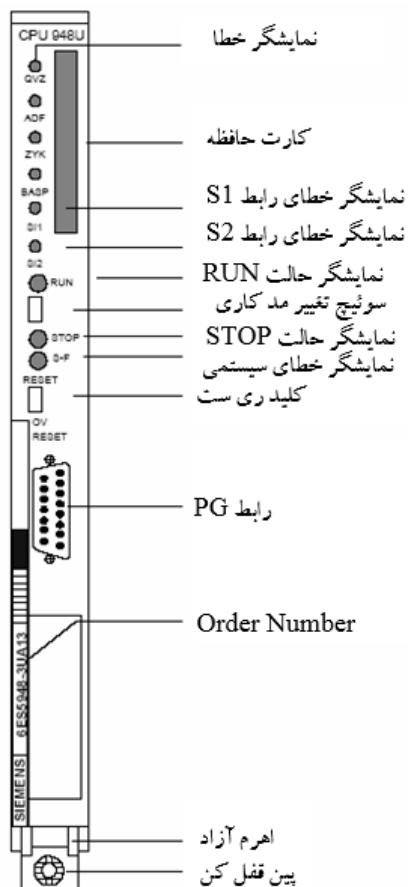
* در مواردی که نیاز به استفاده به زبانهای برنامه نویسی سطح بالا می باشد . مانند C و SCL

* ساپورت نمودن تعداد بالایی کارت در یک رک

* قابل استفاده در سیستم های Redundant

* در کاربردهایی که نیاز به استفاده از چند پروسسور می باشد.

ماژول CPU



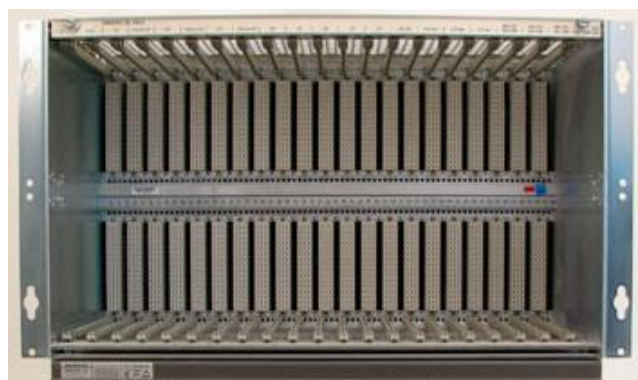
آشنایی با قسمتهای مختلف واحد رک

تمامی کارت ها در S5-155U در یک محفظه ای به نام رک قرار می گیرند که این مدل نیز دارای رک های مختلفی می باشد. در ادامه چندین رک را ملاحظه می کنید.



84

۱- محل قرارگیری کارت ها



این قسمت در واقع قسمت اصلی یک رک بوده که محل قرارگیری کارتها می باشد. در این قسمت اسلاتهایی جهت قرارگیری کارتها تعبیه شده است.

۲- داکت سیم ها

در قسمت پایین محل قرارگیری کارتها یک داکت جهت قراردادن سیم ها در آن وجود دارد.

در شکل سیم داکت مربوط به سیم ها مشخص شده است.



داکت سیم ها

۳- واحد منبع تغذیه

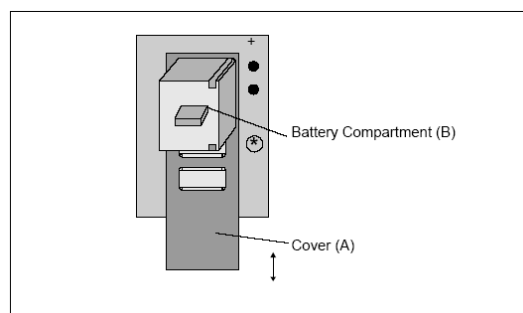
در S5-155U واحد تغذیه بصورت یک واحد مجزا از سایر کارتها بوده و در قسمت پایین واحد رک بصورت افقی قرار می گیرد.



رک های مرکزی شامل ۲ یا ۳ فن و یک باتری Backup می باشند. در ضمن رک های ER فاقد باتری و فن بوده و تنها محل قرارگیری I/O ها و سایر کارتهای توسعه یافته می باشد. بر روی منبع تغذیه دو ترمینال جهت اتصال ولتاژ 220/120 VAC وجود دارد که این قسمت پس از دریافت این ولتاژ، ولتاژهای مورد نیاز را تولید می کند. (5V-15V-24V)

بر روی واحد منبع تغذیه دو ترمینال جهت اعمال ولتاژ خارجی در هنگام تعویض باتری تعبیه شده است. این ولتاژ یک ولتاژ 3.4v می باشد.

این باتری زمانی که برق دستگاه قطع می شود جهت حفظ اطلاعات موجود در حافظه RAM وارد مدار می شود.



انواع منابع تغذیه در S5135,155U

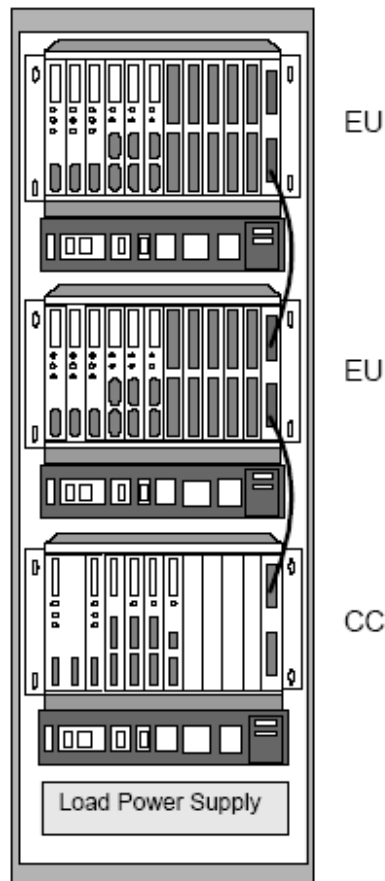
Type of PSU (Designation)	Input Voltage	Output Voltage	Application
6ES5 955-3LC42	120 V AC 230 V AC (selectable)	5V/18A DC 15V/0.5A DC 24V/1A DC	Central controller 6ES5 188-3UA12 Expansion units 6ES5 183-3UA13 6ES5 185-3UA13
6ES5 955-3LF42	120 V AC 230 V AC (selectable)	5V/40A DC 15V/2A DC 24V/2.8A DC	Central controller 6ES5 188-3UA22 Expansion units 6ES5 185-3UA33
6ES5 955-3NC42	24 V DC	5V/18A DC 15V/0.5A DC 24V/1A DC	Central controller 6ES5 188-3UA32 Expansion units 6ES5 183-3UA22 6ES5 185-3UA23
6ES5 955-3NF42	24 V DC	5V/40A DC 15V/2A DC 24V/2.8A DC	Central controller 6ES5 188-3UA52 Expansion units 6ES5 185-3UA43

رک در S5-155U/135U:

در S5-135,155U نیز رک ها به دو گروه زیر تقسیم بندی می شوند:

رک CC: رک مرکزی که CPU به همراه تعداد مشخصی کارت در آن قرار می گیرند.

رک EU: محل فراگیری کارت های افزایشی می باشند.

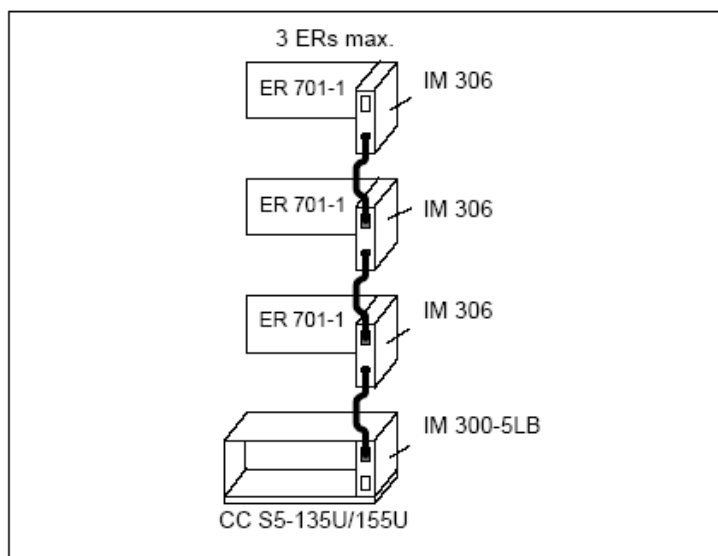
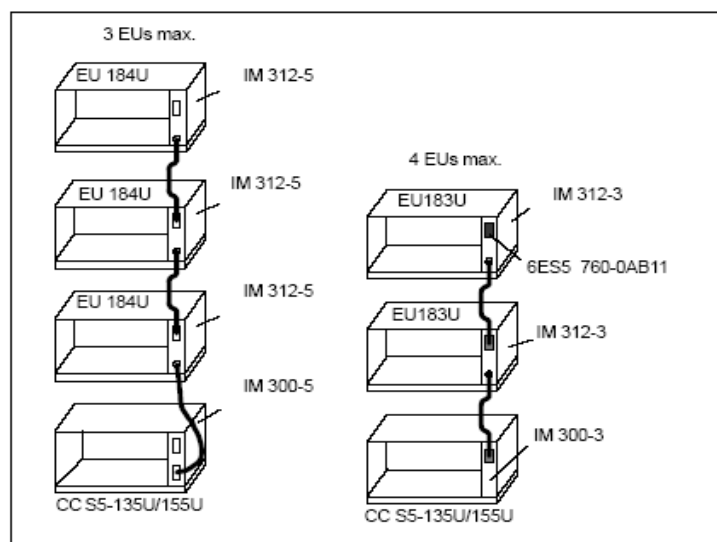


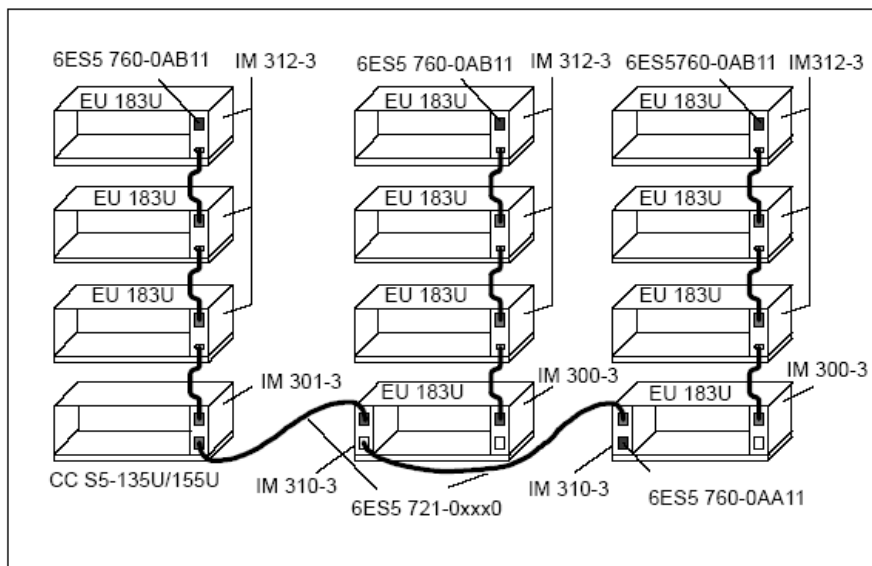
جهت ارتباط بین CC و EU ها نیز از کارتهای IM استفاده می شود. IM ها در این سری از خانواده S5 دارای مشخصه های متفاوتی می باشند. ارتباط بین رک ها می تواند بصورت سریال و پارالل باشد. ارتباط پارالل دارای سرعت تبادل دیتا بالا در فاصله حدود ۶۰۰ متر می باشد، در صورتی که در ارتباط سریال این سرعت کمتر بوده ولی فاصله می تواند تا حدود ۳۰۰۰ متر باشد.

جدول مشخصات کارتهای IM

Interface Module in the CC	Expansion Unit	Interface Module in the EU	Connecting Cable Max. Distance
IM 300-3 6ES5 300-3AB11	EU 183U EU 185U (I/O mod. only)	IM 312-3 ²⁾ 6ES5 312-3AB11	fixed to the IM 312 module 0.5 m; 0.95 m
IM 301-3 ¹⁾ 6ES5 301-3AB13	EU 183U EU 185U (I/O mod. only)	IM 312-3 ²⁾ 6ES5 312-3AB31	fixed to the IM 312 module 0.5 m; 0.95 m
IM 300-5 6ES5 300-5CA11	EU 184U EU 187U	IM 312-5 6ES5 312-5CA11	fixed to the IM 312 module 0.5 m; 1.5 m
IM 301-5 ¹⁾ 6ES5 301-5CA12	EU 184U EU 187U	IM 312-5 6ES5 312-5CA21	fixed to the IM 312 module 0.5 m; 1.5 m
IM 300-5 6ES5 300-5LB11	ER 701-1	IM 306 6ES5 306-7LA11	6ES5 705-0xxxx 0.5 m to 2,5 m

در شکل زیر نحوه ارتباط بین رک های مختلف را ملاحظه می کنید.





محل قرارگیری کارت ها در رک CC

Slot No.	3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163	
Module Type																						
923 coordinator	█																					
CPU 922/ CPU 928-3UA21/ CPU 928B-3UB21		█		█		█		█														
CPU 928, CPU 928B CPU 948		█	█	█	█	█	█	█	█													
CP 5XX, CP 143, CP 5430, CP 5431 1)			█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
IM 300-5 IM 301-5 2)																					█	█
IM 300-3, IM 301-3 IM 304, IM 308, IM 308B/IM 308 C IM 307 1) 3)																					█	█
DI, DQ, AI, AQ 1)		█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
IP 241USW, IP 244, IP 246, IP 247, IP 252 1)			█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
IP 240, IP 241, IP 242, IP 242A, IP 242B, IP 243, IP 281 1) 4) 5)			█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
IP 245 IP 257 6)			█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
IP 260, IP 261			█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Load power supply -951 1)			█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█

محل قرارگیری کارت‌ها در رک های EU

EU183U

Slot No.	3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163	
Module type																						
IM 300																						■
IM 310, IM 314, IM 318	■																					
IM 317	■																				■	
IM 312-3																						■
DI, DQ, AI, AQ	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Signal pre-processing modules (IPs)	See current Catalog ST 54.1 for slot numbers																					
Monitoring module 313	■																					■

EU184U

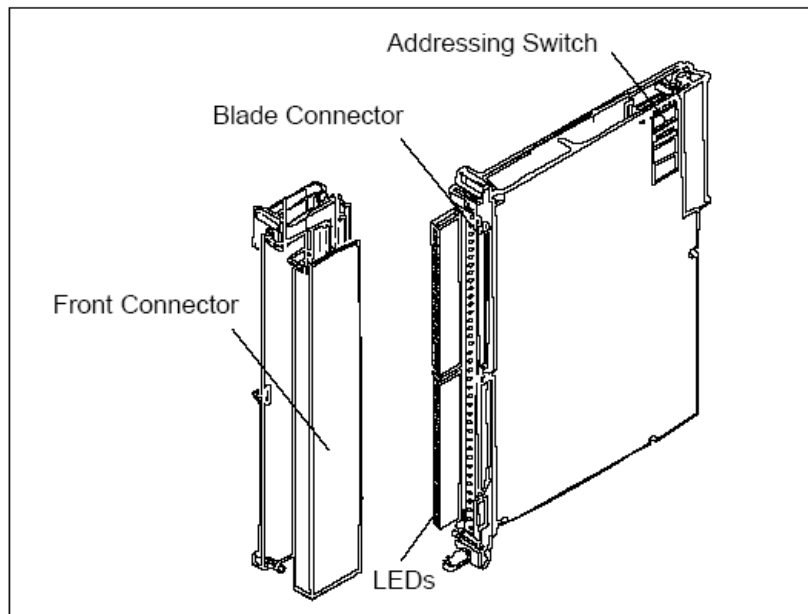
Slot No.	3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163	
Module type																						
IM 312-5																						■
DI, DQ, AI, AQ	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Signal pre-processing modules (IPs)	See current catalog ST 54.1 for slot numbers																					
Monitoring module 313	■																					

EU185U

Slot No.	3	11	19	27	35	43	51	59	67	75	83	91	99	107	115	123	131	139	147	155	163	
Module type																						
Coordinator 923 C		■																				
Communication processors (CPs)			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
IM 314 R																				■	■	
IM 300																						■
IM 310, IM 134, IM 318	■																					
IM 317	■																				■	
IM 308																				■	■	■
DI, DQ, AI, AQ	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Signal pre-processing modules (IPs)	See current catalog ST 54.1 for slot numbers																					
Monitoring module 313																						■

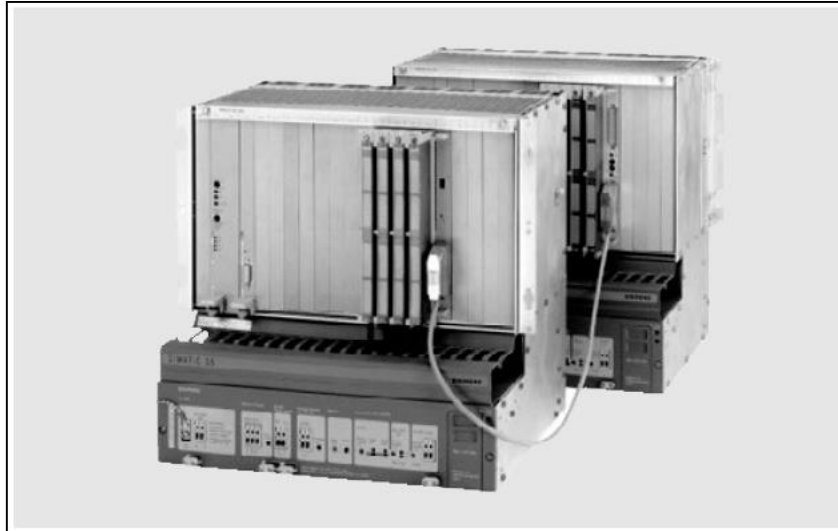
Slot No.	3	19	35	51	67	83	99	115	131	147	163
Module type											
IM 312-5											
DI, DQ, AI, AQ											
Monitoring module 313											

همچنین کارتهای ورودی و خروجی دارای یک کاور پلاستیکی جهت حفاظت اجزاء و قرارگیری برچسب می باشند. به ازای هر بیت نیز یک LED در سمت چپ پین ها قرار گرفته است. در پشت کارت نیز یک کد سوئیچ جهت تعیین آدرس و پارامترهای کارت قرار گرفته است. در اکثر کارتهای ورودی و خروجی در PLC های زیمنس نیاز به یک Front Connectors می باشد که عملیات سیم کشی و ارتباط با کارت را مقدور می سازد.

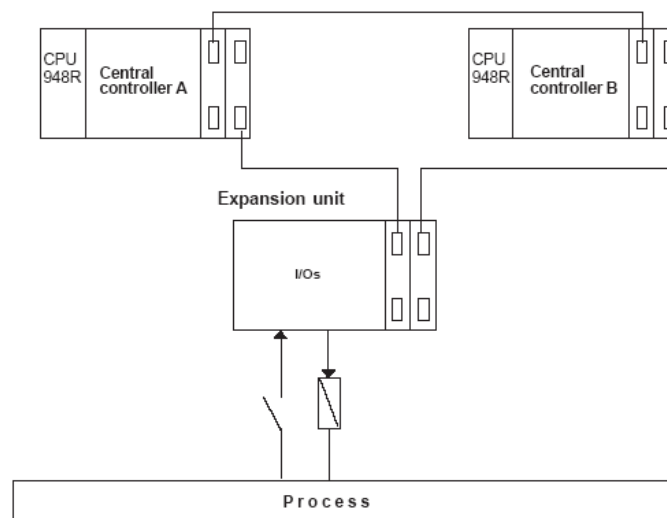


یکی دیگر از قابلیت های S5-155U استفاده در سیستم های Redundant می باشد. در این سیستم از دو CPU جهت کنترل پروسه های صنعتی استفاده می شود. در واقع این سیستم ها بصورت جفت یا دوقلو به کار می روند. از این سیستم در پروسه های حساس یا پروسه هایی که به هیچ وجه نباید عمل توقف صورت گیرد استفاده می شود. نحوه عملکرد این سیستم بدین صورت می باشد که در حالت عادی اجرای امور بدست یکی از CPU ها می باشد. زمانی که مشکلی در سیستم رخ دهد، در زمان بسیار کوتاهی CPU بعدی وارد مدار شده و کار کنترل را ادامه می دهد. وقتی یک سیستم در حال کار و دیگری در حال آماده بکار است مد کاری مجموعه را Redundant می گویند و وقتی یک سیستم بدلیل خطا متوقف می شود و دیگری در وارد مدار می شود، مد کاری مجموعه را single می نامند.

ایجاد سیستم Redundant در I/O ها نیز قابل اجرا می باشد. در مواردی که اهمیت I/O ها بالاست می توان با ایجاد این سیستم اطمینان سیستم را بالا برد. CPU هایی که دارای حرف H می باشند (S5-155H) قابلیت Redundant را ساپورت می کنند.



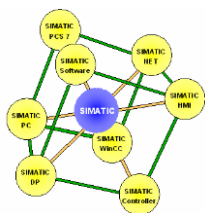
92



همانطور که در شکل فوق ملاحظه می کنید CPU 948R در S5-155U به عنوان یک CPU با قابلیت Redundant می باشد.

فصل دوم- برنامه نویسی PLC

93



مشمول بر

- * آشنایی با دستورات S5
- * آشنایی با انواع بلوکهای برنامه نویسی
- * ارائه مثالهای کاربردی
- * پردازش سیگنال های آنالوگ



مقدمه ای بر زبان STEP 5

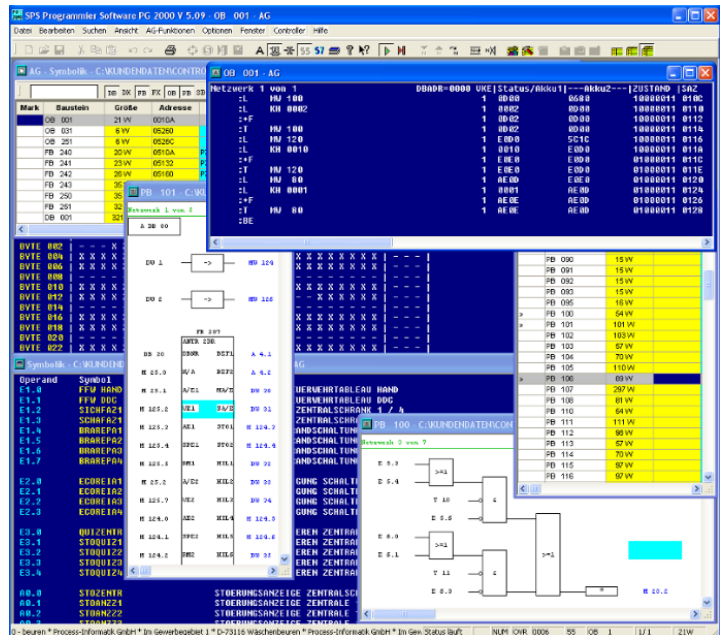
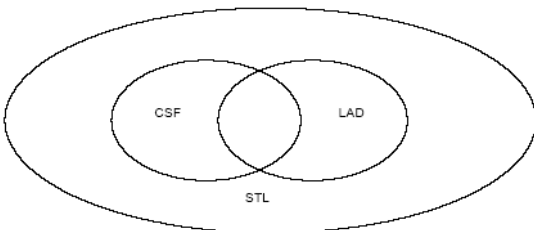
یک برنامه کنترلی شامل مجموعه ای از دستورات است که به کنترل کننده فرمانهایی جهت کنترل پروسه صادر می کند. با توجه به این مطلب هر شرکت سازنده برنامه و دستورالعملهای خاصی برای درک PLC همان شرکت ارائه می دهد که کاربران می بایست با این دستورات آشنایی لازم داشته باشند.

روشهای نمایش برنامه در S5

در PLC های شرکت زیمنس سه روش نمایش برنامه وجود دارد. که این سه عبارتند از:

- ۱- نردبانی (Ladder) LAD
- ۲- فلوچارتی (Control System Flochart) CSF
- ۳- نوشتاری (Statement List) STL

همانطور که ملاحظه کردید، S5 دارای سه روش نمایش برنامه یعنی فلوچارتی، نوشتاری و نردبانی می باشد. روش فلوچارتی و نردبانی زیرمجموعه روش نوشتاری می باشند. این بدان معنی است که اگر برنامه ای به یکی از دو روش CSF یا LAD نوشته شود، کامپایلر حتما روش STL این برنامه را نشان می دهد. ولی اگر برنامه ابتدا به روش STL نوشته شود، کامپایلر اکثر اوقات توانایی تبدیلی این برنامه را به دو روش دیگر ندارد. دلیل این مطلب این است که برخی از دستورات در روش STL وجود دارند که معادل آنها در دو روش دیگر تعریف نشده است. شکل زیر این مطلب را بصورت واضح تر نشان می دهد.



آشنایی با روشهای برنامه نویسی

روش STL

یک خط برنامه به روش STL از قسمتهای زیر تشکیل شده است.

1	2	3
Operation	Operand	Address
عملکرد	عملوند	آدرس

عملکرد:

قسمت عملکرد همان منطقی است که می خواهیم بین ورودی ها و سایر عملوندها در برنامه ایجاد نماییم. مهمترین عملکردها عبارتند از AND، OR، = که در جدول شکل زیر نشان داده شده اند.

در برنامه نویسی STL	در زبان محاوره
A	AND سری
O	OR موازی
AN	AND NOT نقیض AND
ON	OR NOT نقیض OR
=	ASSIGN مساوی

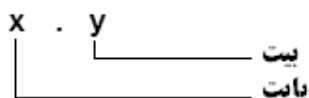
عملوند:

به قسمتی از عبارت گفته می شود که قرار است عمل منطقی در مورد آن اجرا شود. مانند ورودی ها، خروجی ها، فلگ ها در PLC های زیمنس ورودی را با حرف I، خروجی را با حرف Q و فلگ را با حرف F نمایش می دهند.

آدرس:

نحوه آدرس دهی در S5

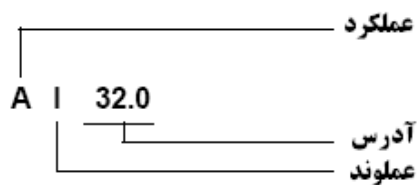
یکی از مهمترین مواردی که در برنامه، کاربر باید مد نظر داشته باشد، آدرس دهی صحیح ورودی، خروجی و فلگ ها می باشد. ترمینالهای ورودی، خروجی و فضای حافظه در دسته های هشت تایی می باشند. پس جهت آدرس دهی یک بیت از حافظه یا I/O باید، ابتدا شماره بیت و سپس شماره بیت را تعیین نمود. شماره بیت ورودی و خروجی، معمولاً توسط شرکت سازنده، بر روی ترمینالهای I/O، مشخص می گردد.



نحوه آدرس دهی را با چند مثال بررسی می کنیم.

I 8.7	بیت هفتم از بایت هشتم ورودی
Q 5.2	بیت دوم از بایت پنجم خروجی
F 1.3	بیت سوم از بایت اول حافظه فلگ

در شکل زیر یک خط برنامه به روش STL را به طور کامل ملاحظه می کنید.



روش LAD

روش نردبانی بسیار شبیه به مدار فرمان در صنعت می باشد. به همین دلیل کسانی که با مدارات فرمان و کنتاکت ها آشنایی دارند، از این روش بیشتر جهت برنامه نویسی استفاده می کنند. در روش LAD، از اتصالات و کنتاکت ها جهت ایجاد برنامه استفاده می شود و برنامه نوشته شده بسیار ساده می باشد. شکل زیر، نحوه نمایش ورودی و خروجی را در این روش نشان می دهد.



روش CSF

نمایش دستورات و توابع در روش CSF بسیار شبیه، علائم نشان دهنده در سیستمهای دیجیتالی می باشد. در این روش از بلوکهای مستطیل شکل جهت نمایش برنامه استفاده می شود. کسانی که با مدارات الکترونیکی و سیستمهای دیجیتالی آشنایی دارند، بیشتر از این روش جهت برنامه نویسی استفاده می کنند.

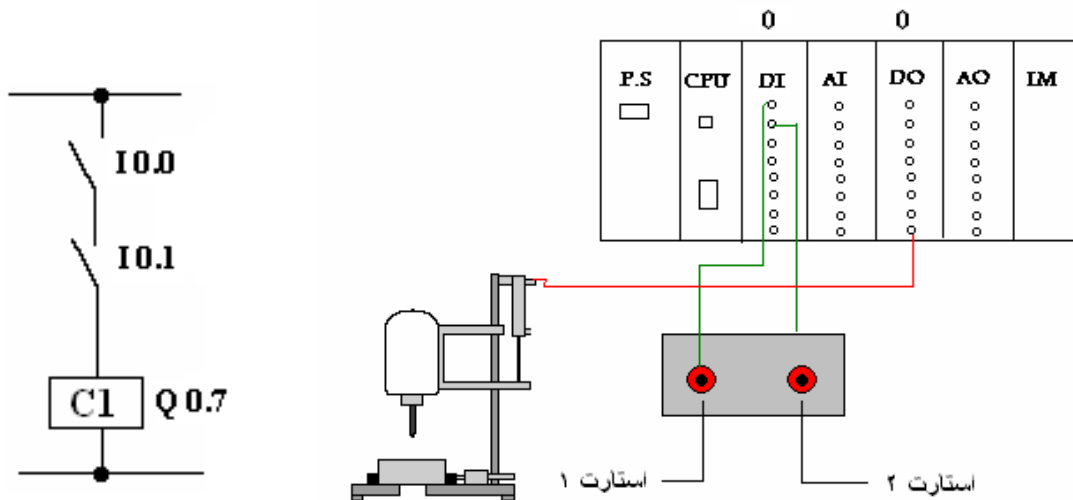


مثال ۱

بعد از آشنایی با روش نمایش برنامه، نوبت به ارائه یک مثال می باشد. در این مثال قصد داریم، موتور مربوط به یک مته سوراخکاری را کنترل نماییم. نحوه کار بدین صورت می باشد که جهت کنترل این موتور از دو کلید استفاده شده است. عمل سوراخکاری زمانی انجام می شود که اپراتور هر دو کلید را فعال نماید.

حل:

در این مثال، دو کلید به عنوان ورودی PLC و کنتاکتور مربوط به موتور به عنوان خروجی PLC می باشد. برنامه نوشته شده باید به گونه ای باشد که PLC دو کلید متصل شده به ورودیش را، با یکدیگر سری نماید. آدرس ورودی و خروجی ها را مطابق مدار شکل زیر تعریف می نماییم.



همانطور که در شکل فوق ملاحظه می کنید، بین ورودیهای PLC، ترکیب سری وجود ندارد. اعمال منطقی سری بین کلیدهای ورودی از طریق برنامه صورت می گیرد.

مدار فرمان	STL	CSF	LAD
	<pre> A I 0.0 A I 0.1 = Q 0.7 BE </pre>		

در روش برنامه STL مثال قبل :

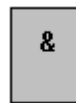
حرف A: مخفف کلمه AND و به معنی منطق سری می باشد.

حرف I: نشانه ورودی PLC می باشد.

علامت =: به معنای انتقال نتیجه برنامه به خروجی PLC می باشد.

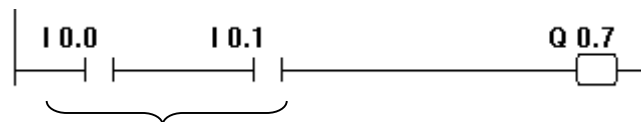
دستور BE: دستور پایان برنامه و مخفف عبارت Block End می باشد.

در روش CSF جهت ایجاد منطق سری یا AND بین ورودی ها از سمبل شکل زیر استفاده می شود.



بلوک مربوط به منطق AND

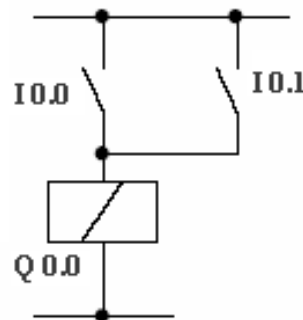
در روش LAD ایجاد منطق مورد نظر بین ورودی ها، بر عهده کاربر می باشد.



منطق سری

مثال ۲

نمایش مداري شکل زیر را در نظر بگیرید. می خواهیم برنامه کنترلي این مدار را توسط PLC طراحی نماییم.

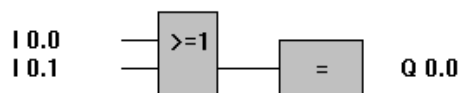


حل:

با دقت در این مدار ملاحظه می شود که عملکرد این مدار به صورت موازی می باشد. یعنی برای روشن شدن خروجی تنها کفایت یکی از دو ورودی فعال شود. نحوه اتصال این مدار به PLC نیز همانند مثال ۱ بوده با این تفاوت که برنامه کنترلي این مثال باید به گونه ای باشد که PLC دو ورودی متصل شده را با یکدیگر OR (موازی) نماید.

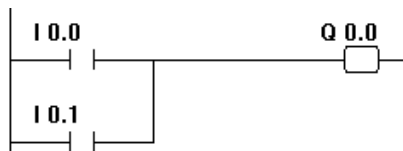
برنامه کنترلی :

O I 0.0
O I 0.1
= Q 0.0
BE



روش STL

روش CSF

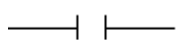


روش LAD

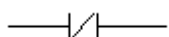
جدول شکل زیر برنامه مثال ۲، را با سه ورودی نشان می دهد.

Example		Circuit Diagram	
<p>با فعال شدن هر کدام از کلیدها، خروجی فعال می گردد</p>			
STL	CSF	LAD	
<p>O I 1.2 O I 1.7 O I 1.5 = Q 3.2</p>			

در PLC دو نوع کنتاکت وجود دارد:



۱- تیغه یا کنتاکت در حالت عادی باز (استارت)



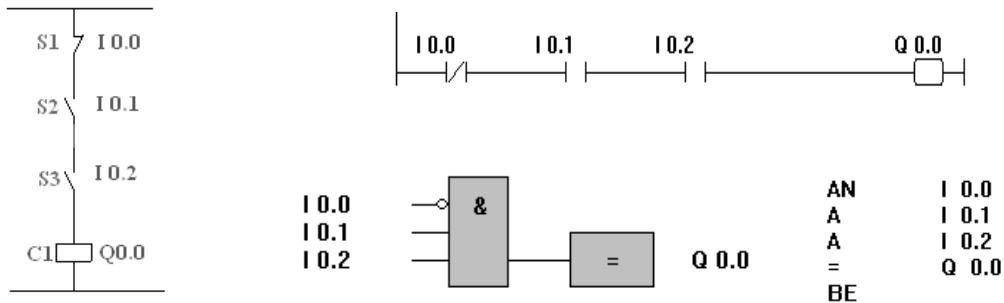
۲- تیغه یا کنتاکت در حالت عادی بسته (استپ)

در حالت ۱، کنتاکت زمانی فعال می شود که کلید مورد نظر فشرده شود. در این حالت در ورودی PLC مقدار ۱ ظاهر می شود.

در حالت ۲ این اتصال در حالت عادی فعال می باشد و با فشردن کلید، مقدار ۰ در ورودی PLC ظاهر می شود.

یکی از مزیت‌های استفاده از PLC این است که می‌توانیم تمامی ورودی‌های PLC را استارت انتخاب و هر جا که نیاز شد در برنامه از استارت به عنوان استپ استفاده کنیم.

مثال: می‌خواهیم برنامه مدار فرمان شکل زیر را به سه روش طراحی کنیم.



در برنامه نویسی STL برای نشان دادن تیغه در حالت عادی بسته از حرف N (NOT) استفاده می‌شود.

در روش فلوجارتی برای NOT کردن ورودی از سمبل روبرو استفاده می‌شود.



در روش نردبانی برای تعریف نمودن تیغه بسته از سمبل شکل زیر استفاده می‌شود.



کاربرد پرانتز در برنامه نویسی

برای روشن شدن این بحث، برنامه دو مدار فرمان را بررسی می‌کنیم.

عدم استفاده از پرانتز

در این مثال عدم استفاده از پرانتز در حالتی که دستور AND قبل از دستور OR قرار گرفته باشد مورد بررسی قرار می‌گیرد.

Example		Circuit Diagram	
Output Q32.5 is "1" when at least one AND condition has been satisfied. Output Q32.5 is "0" when neither of the two AND conditions has been satisfied.			
STL	CSF	LAD	
A I 32.0 A I 32.1 O Q 32.5 A I 32.2 A I 32.3 = Q 32.5			

اگر در مدار، منطق بین ورودی ها همانند برنامه شکل صفحه قبل باشد، نیازی به استفاده از پرانتز نمی باشد و فقط با قرار دادن یک حرف O (OR) می توان منطق موازی را بین توابع ایجاد نمود. این در حالتی است که منطق AND قبل از منطق OR در یک مدار فرمان وجود داشته باشد. اگر به مدار فرمان شکل صفحه قبل توجه نمایید، در می یابید که ابتدا ورودی ها با یکدیگر سری و در مجموع با یکدیگر موازی شده اند.

مثال ۲

در این مثال چگونگی استفاده از پرانتز تشریح شده است.

101

Example		Circuit Diagram	
در این مدار، خروجی زمانی فعال می شود که حاصل هر دو تابع OR برابر ۱ شود.			
STL	CSF	LAD	
<pre> A(O I 32.0 O I 32.1) A(O I 32.2 O I 32.3) = Q 32.5 </pre>			

با اندکی دقت در مدار شکل فوق در می یابید که زمانی از پرانتز در برنامه استفاده می شود که دستور OR قبل از دستور AND استفاده شده باشد. در این صورت هر کدام از توابع OR را داخل یک پرانتز قرار داده و با نوشتن حرف A (AND) دو پرانتز را با یکدیگر سری می کنیم.

$$(I\ 32.0 + I32.1) \cdot (I32.2 + I32.3) = Q32.5$$

مثال:

در این مثال کاربرد پراگماتر به همراه توابع AND و OR نشان داده شده است.

Example		Circuit Diagram	
در این مدار چندین راه جهت فعال شدن خروجی وجود دارد.			
STL	CSF	LAD	
<pre> O I 32.0 O A I 32.1 A(O I 32.2 O I 32.3) - Q 32.5 </pre>			

فلگ ها

فلگ ناحیه ای از حافظه می باشد که از آن جهت ذخیره اطلاعات در برنامه نویسی استفاده می شود. لذا در برنامه می توان حاصل و نتایج یکسری منطق ها و ترکیب ها را در فلگ های جداگانه قرار داد و در مواقع لازم این فلگ ها را فراخوانی نمود.

فضای حافظه و یا فلگ ها نیز همانند ورودی و خروجی ها در دسته های هشت تایی تقسیم بندی می شوند. تعداد فلگ ها و ظرفیت آن در هر PLC متفاوت می باشد. به عنوان مثال فرض نمایید در مشخصات یک PLC ظرفیت فلگ 127 بایت بیان شده است. این بدان معنی می باشد که فلگ های PLC مذکور از بایت 0 شروع و به بایت 126 ختم می شوند.

نحوه آدرس دهی فلگ ها

فلگها در محدوده های مختلف قابل دسترسی می باشند. این بدان معناست که فلگ ها می توانند در برنامه جهت ذخیره یک بیت ، یک بایت و یا یک کلمه به کار گرفته شوند.

$$F \ m.n$$

n : شماره بیت

m : شماره بایت

مثال

F 5.2 : بیت دوم از بایت پنجم فلگ

F 2.3 : بیت سوم از بایت دوم فلگ

در PLC S5 دو دسته فلگ وجود دارد:

۲- ناپایدار

۱- پایدار

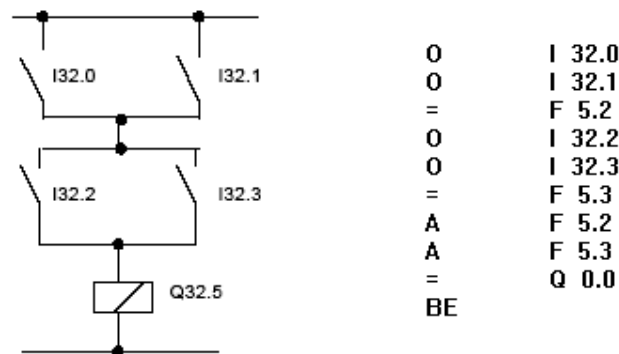
103

تعداد عناصر پایدار و ناپایدار با یکدیگر برابر است. فلگ های پایدار به آن دسته از فلگ ها اطلاق می شود که در صورت قطع برق PLC اطلاعات خود را از دست ندهند.

ناپایدار به آن دسته از فلگ ها اطلاق می شود که در صورت قطع برق اطلاعات خود را از دست می دهند. همیشه در PLC S5 نیمه اول این عناصر پایدار و نیمه دوم ناپایدار می باشند. مثلاً اگر PLC دارای 127 فلگ باشد نیمه اول آن یعنی از 0 تا 63 پایدار و از 64 تا 127 ناپایدار می باشند.

Operand	S5-90U	
	Retentive	Non-Retentive
Flags	0.0 to 63.7	64.0 to 127.7

حال قصد داریم که یک برنامه را با استفاده از فلگ بازنویسی نماییم.

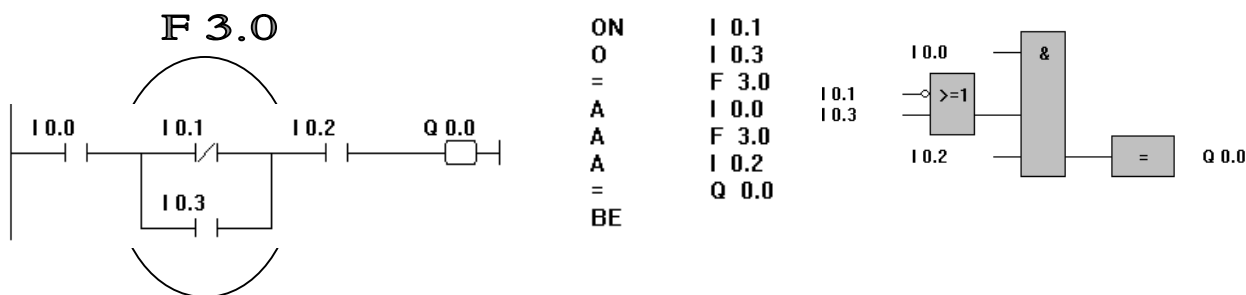


نکته:

اگر برنامه ای ابتدا به روش STL و با استفاده از فلگ نوشته شود، این برنامه به روشهای دیگر مانند CSF یا LAD تبدیل نمی شود.

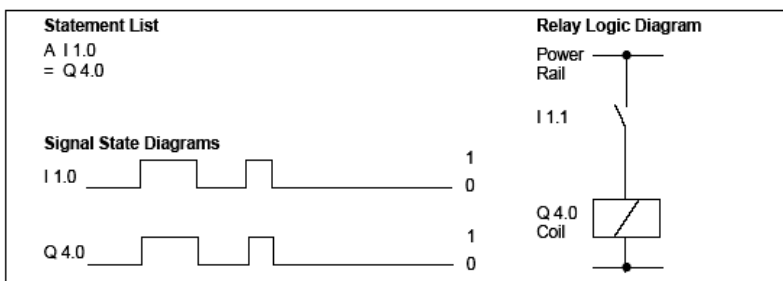
تمرین:

برنامه شکل زیر به روش LAD می باشد، آن را به دو روش دیگر بنویسید.



مثال: ویژگیهای دستور هم ارزی =

در این مثال قصد داریم برنامه ای به روش STL بنویسیم و وضعیت یک کلید را به یک رله منتقل نماییم.



اگر به دیاگرام مدار شکل فوق دقت کنید در می یابید که ورودی I 1.0، خروجی را بصورت لحظه ای وصل می کند. یعنی با غیر فعال شدن ورودی، خروجی نیز قطع می شود.

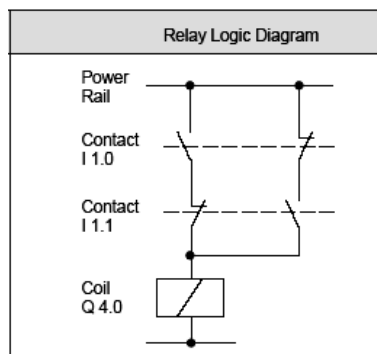
نتیجه: زمانی که از دستور هم ارزی (=) استفاده می شود، خروجی بصورت لحظه ای در مدار می باشد.

استفاده از تیغه های دوبل در برنامه

یکی دیگر از مزایای استفاده از PLC این است که نیازی به استفاده از استپ استارت دوبل بصورت سخت افزاری نمی باشد. در این حالت می توانیم از یک شاستی معمولی در ورودی PLC استفاده و با نوشتن یک برنامه، از آن شاستی بصورت دوبل استفاده کنیم. جهت ایجاد یک شاستی دوبل در برنامه، آدرسهای دو تیغه را مشترک فرض می کنیم.

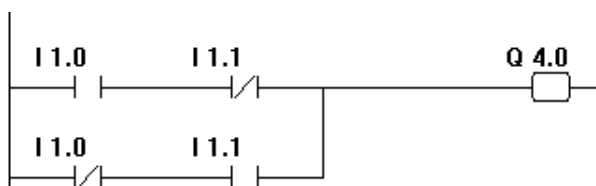
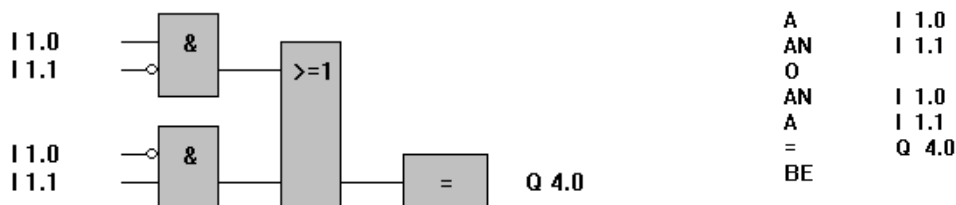
مثال:

می خواهیم برنامه مدار فرمان شکل زیر را طراحی نماییم.



105

برنامه کنترلی



یادآوری

در برنامه نویسی جهت تعریف نمودن تیغه دابل، آدرس بین دو تیغه را مشترک در نظر می گیریم.

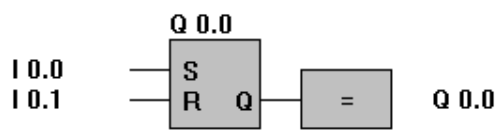
فلیپ فلاپها

در مدارات فرمان کنتاکتوری زمانی که بخواهیم یک مصرف کننده بصورت دائم در مدار قرار گیرد یا به عبارت دیگر در کنترل یک دستگاه لازم است فقط در یک لحظه کلیدی فشرده شود تا دستگاه شروع به کار نماید. در این حالت از یک تیغه باز کنتاکتور که به تیغه نگهدارنده معروف می باشد ، استفاده می گردد. یکی از دلایلی که موجب افزایش حجم سیم در تابلوهای فرمان می شود ، استفاده از تیغه های کمکی کنتاکتور می باشد. زمانی کنترل یک دستگاه برعهده PLC می باشد، دیگر نیازی به استفاده از تیغه نگهدارنده و یا هر تیغه کمکی نمی باشد. در PLC جهت کنترل دائم از دستورات فلیپ فلاپ استفاده می شود. در حقیقت فلیپ فلاپ را می توان معادل تیغه نگهدارنده در مدارات فرمان دانست.

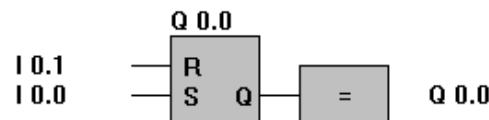
در PLC دو نوع فلیپ فلاپ وجود دارد:

۱- فلیپ فلاپ SR

۲- فلیپ فلاپ RS



فلیپ فلاپ SR



فلیپ فلاپ RS

106

تفاوت فلیپ فلاپها:

تنها تفاوت این دو فلیپ فلاپ در ارجعیت دو ورودی ست و ری ست می باشد. در فلیپ فلاپ SR ارجعیت با ورودی R و در فلیپ فلاپ RS ارجعیت با ورودی S می باشد. این بدان معنی است که در فلیپ فلاپ SR زمانی که هر دو ورودی ست و ری ست فعال شوند خروجی قطع، ولی در فلیپ فلاپ RS زمانی که هر دو ورودی فعال باشند، خروجی در وضعیت یک قرار می گیرد.

روش نمایش STL هر دو فلیپ فلاپ را ملاحظه می نمایید.

```
A I 0.0
S Q 0.0
A I 0.1
R Q 0.0
A Q 0.0
= Q 0.0
BE
```

فلیپ فلاپ SR

```
A I 0.1
R Q 0.0
A I 0.0
S Q 0.0
A Q 0.0
= Q 0.0
BE
```

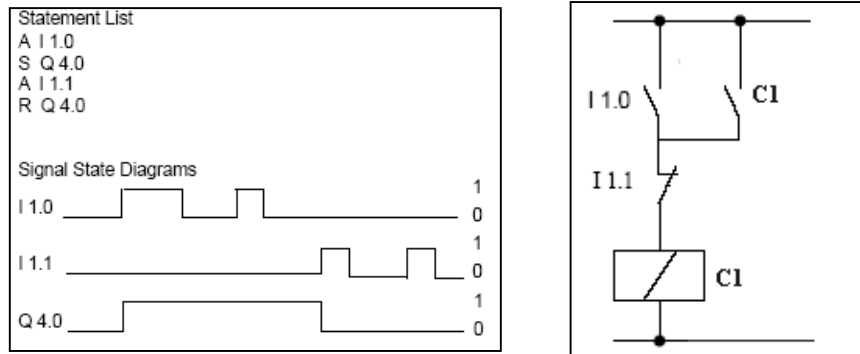
فلیپ فلاپ RS

طرز کار فلیپ فلاپ

زمانی که ورودی ری ست در وضعیت صفر باشد، کفایت در یک لحظه ورودی ست فعال شود تا خروجی بصورت دائم وارد مدار شود. حال با غیرفعال شدن ورودی ست، تغییری در خروجی ایجاد نمی شود و به عبارت دیگر خروجی بصورت پایدار در مدار قرار گرفته است. زمانی که ورودی R فعال می شود خروجی به وضعیت صفر تغییر حالت می دهد. ضمناً این نکته را نیز باید یادآور شد که هر دو ورودی فلیپ فلاپ به لبه بالا رونده ورودی حساس می باشند.

نکته:

در روش STL، آن دستوری که به دستور پایان برنامه یعنی BE، نزدیکتر می باشد، ارجعیت دارد. دیاگرام شکل زیر تاثیر یک ورودی ست و ری ست را بر یک خروجی نشان می دهد.

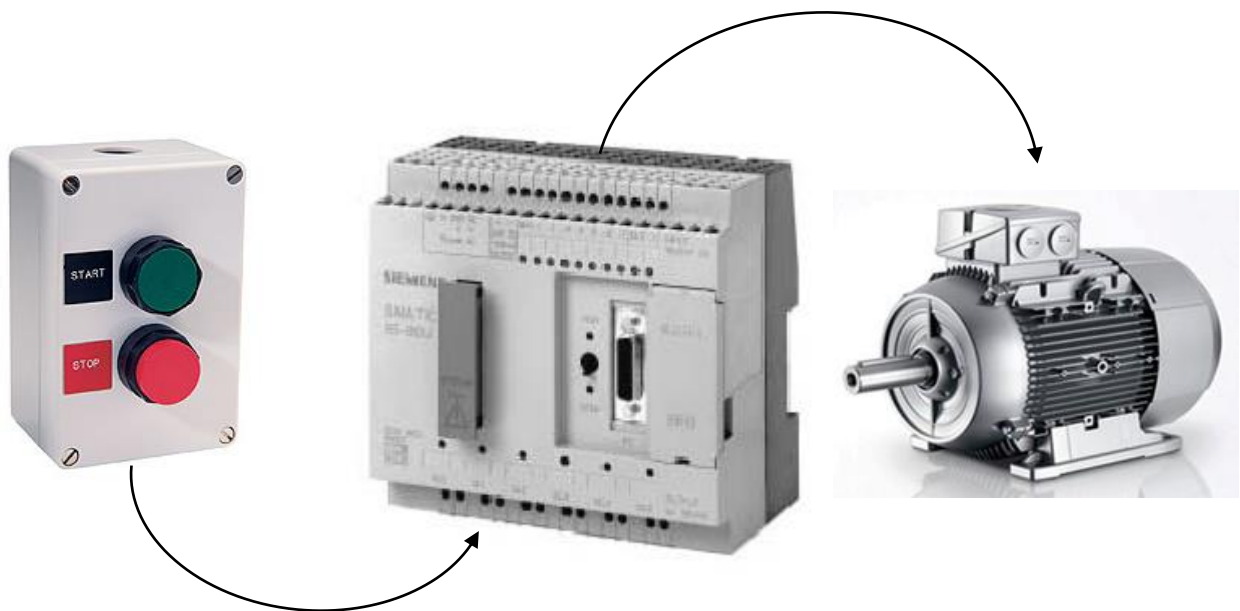


107

در کنترل اکثر فرآیندهای صنعتی از فیلیپ فلاپ SR استفاده می شود. این به این دلیل است که در مواقع خطرناک می خواهیم با فعال شدن ورودی ری ست در هر حالت، خروجی قطع شده و دستگاه از کار بیافتد.

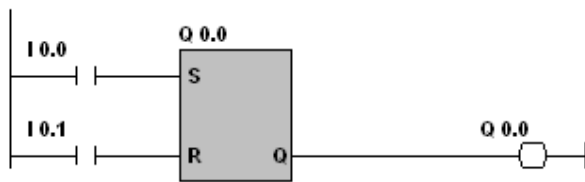
مثال

در این مثال، می خواهیم توسط یک شستی استارت، یک موتور به صورت دائم روشن شده و با فشردن شستی استپ نیز موتور خاموش شود.



فرمان خروجی PLC در ابتدا به یک کنتاکتور یا رله ارسال می شود.

برنامه



```

A      I 0.0
S      Q 0.0
A      I 0.1
R      Q 0.0
A      Q 0.0
=      Q 0.0
BE

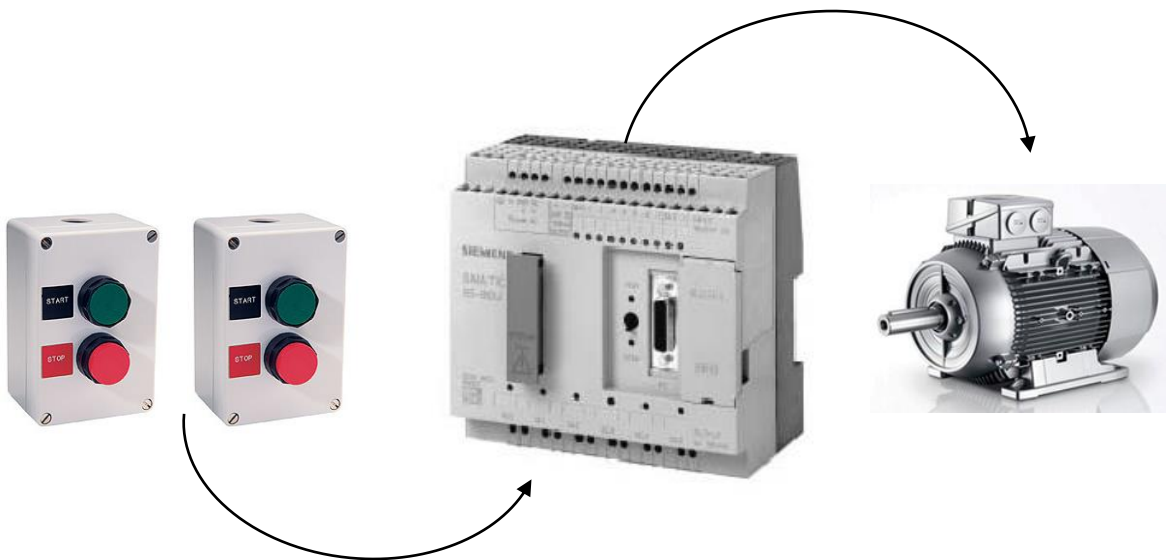
```

نکته: هر دو ورودی فیلپ فلاپ به لبه بالا رونده حساس می‌باشند. در این مثال فرض بر این است که هر دو ورودی متصل شده به PLC به صورت N.O یا همان استارت می‌باشند. اگر به ورودی I0.1 در عمل شستی N.C یا همان استپ متصل شود، برای ورودی I0.1 در برنامه باید از تیغه بسته استفاده شود.

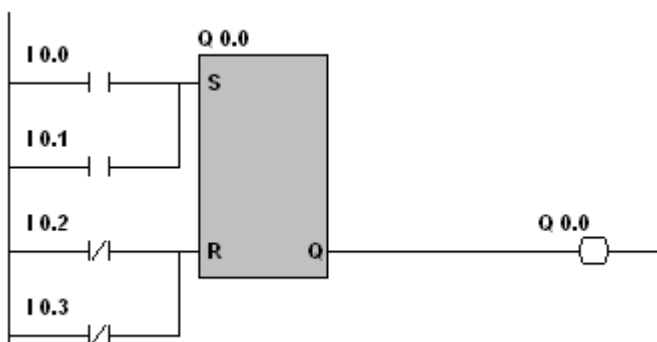
108

مثال - کنترل یک موتور از دو نقطه بصورت دائم

در این مثال، می‌خواهیم برنامه‌ای طراحی کنیم که یک موتور را بتوان از دو نقطه به صورت دائم استارت و استپ نمود. در این مثال، فرض بر این است که شستی‌های متصل شده به PLC برای استپ موتور، از جنس N.C یا همان استپ می‌باشند.



برنامه



```

O      I 0.0
O      I 0.1
S      Q 0.0
ON     I 0.2
ON     I 0.3
R      Q 0.0
A      Q 0.0
=      Q 0.0
BE

```

مثال - کنترل ۲ پمپ

فرض کنید در یک پروسه صنعتی برای راه اندازی دو پمپ از دو استارت مجزا (هر پمپ، یک استارت) استفاده می‌شود. استپ هر دو پمپ توسط یک شستی انجام می‌شود. در ضمن، هیچ شرطی یا اینترلاکی بین این دو پمپ وجود ندارد. می‌خواهیم برنامه کنترلی این مثال را طراحی کنیم.



IO.0 : استارت پمپ ۱

IO.1 : استارت پمپ ۲

IO.2 : استپ کل

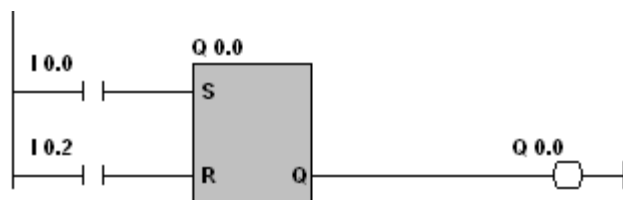
Q0.0 : فرمان پمپ ۱

Q0.1 : فرمان پمپ ۲

109

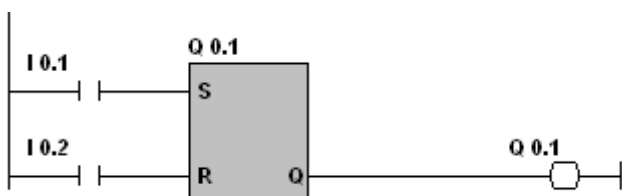
برنامه

Segment 1



A	I 0.0
S	Q 0.0
A	I 0.2
R	Q 0.0
A	Q 0.0
=	Q 0.0

Segment 2



A	I 0.1
S	Q 0.1
A	I 0.2
R	Q 0.1
A	Q 0.1
=	Q 0.1
BE	

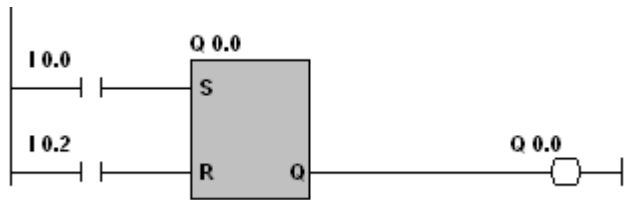
همانطور که ملاحظه می‌کنید، برنامه هر پمپ در یک Segment مجزا وارد شده و بین این دو پمپ نیز هیچ گونه اینترلاک یا شرطی وجود ندارد.

برای ساده‌تر شدن تست برنامه در محیط سیمولاتور که در فصل‌های بعدی مورد بررسی قرار می‌گیرد، در تمامی مثال‌های ارائه شده فرض بر این است که تمام شستی‌های متصل شده به PLC به صورت N.O (استارت) می‌باشند.

مثال

می‌خواهیم برنامه مثال قبل را به گونه ای تغییر دهیم که تا زمانی که پمپ ۱ روشن نشده است، پمپ ۲ روشن نشود. در واقع، روشن شدن پمپ دوم منوط به روشن بودن پمپ ۱ و زدن شستی استارت مربوطه باشد.

Segment 1

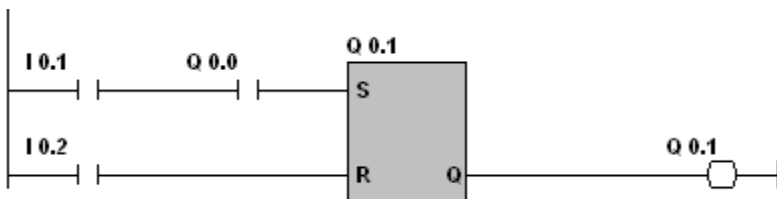


```

A      I 0.0
S      Q 0.0
A      I 0.2
R      Q 0.0
A      Q 0.0
=      Q 0.0
***
    
```

110

Segment 2



```

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

همانطور که در برنامه این مثال مشاهده می‌کنید، از تیغه Q0.0 که مربوط به پمپ ۱ می‌باشد در سر مسیر استارت پمپ ۲ استفاده شده است. این تیغه زمانی بسته می‌شود که Q0.0 فعال شده باشد.

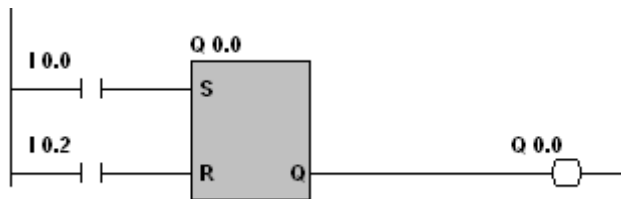
مثال

در مثال قبل، تنها پمپ ۱ به عنوان شرط روشن شدن برای پمپ ۲ می‌باشد. مثال قبل را به گونه ای تغییر دهید که اگر به هر دلیلی پمپ ۱ خاموش شد، پمپ ۲ نیز خاموش شود.

➤ نکته: اگر یک مصرف کننده توسط دستور فلیپ فلاپ ست شده باشد، تنها با تحریک پایه ریست فلیپ فلاپ خروجی مربوطه خاموش می‌شود.

➤ نتیجه: اگر در مثال قبل، تیغه Q0.0 بعد از روشن شدن پمپ ۲ به هر دلیلی قطع یا باز شود، پمپ ۲ خاموش نمی‌شود.

Segment 1

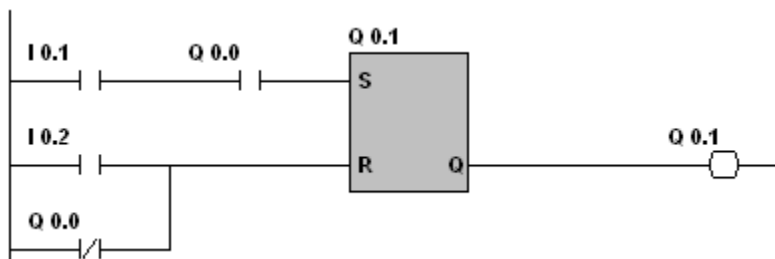


```

A      I 0.0
S      Q 0.0
A      I 0.2
R      Q 0.0
A      Q 0.0
=      Q 0.0
***

```

Segment 2



```

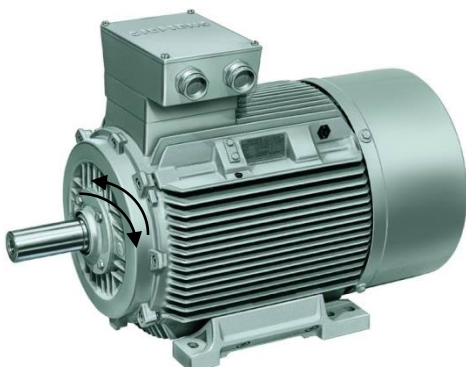
A      I 0.1
A      Q 0.0
S      Q 0.1
O      I 0.2
ON     Q 0.0
R      Q 0.1
A      Q 0.1
=      Q 0.1
BE

```

همان طور که در برنامه این مثال مشاهده می کنید، نبودن Q0.0 نیز منجر به ریست شدن Q0.1 می شود. البته لازم به ذکر است که این برنامه را به روش های دیگر نیز می توان طراحی نمود. در این مرحله از آموزش، هدف تحلیل برنامه ها توسط شما دوستان گرامی می باشد.

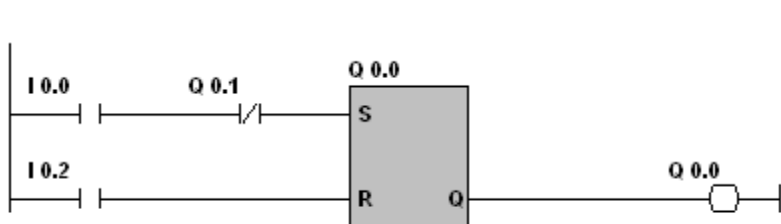
مثال - چپگرد و راستگرد

در یک پروسه صنعتی لازم است که یک موتور به صورت چپگرد و راستگرد راه اندازی شود. در این مثال می خواهیم برنامه ای برای کنترل این موتور طراحی کنیم. در ضمن، در این مثال برای تغییر دور موتور حتماً باید موتور ابتدا استپ شود.

لیست I/O :

- I0.0 : استارت چپگرد
- I0.1 : استارت راستگرد
- I0.2 : استپ
- Q0.0 : خروجی چپگرد
- Q0.1 : خروجی راستگرد

Segment 1

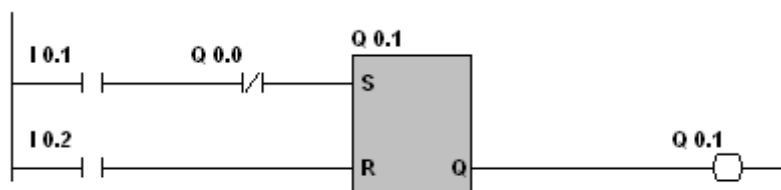


```

A      I 0.0
AN     Q 0.1
S      Q 0.0
A      I 0.2
R      Q 0.0
A      Q 0.0
=      Q 0.0
***

```

Segment 2



```

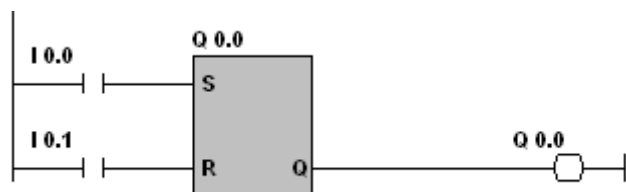
A      I 0.1
AN     Q 0.0
S      Q 0.1
A      I 0.2
R      Q 0.1
A      Q 0.1
=      Q 0.1
BE

```

در مدار فرمان چپگرد و راستگرد باید به این نکته توجه کرد که دو کنتاکتور نباید به طور همزمان در مدار قرار گیرند. به این دلیل، در مدارات فرمان از تیغه‌های بسته کنتاکتور ها استفاده می‌شود. مثلاً، زمانی که کنتاکتور C1 فعال شود، تیغه مربوطه اش در سر راه کنتاکتور C2 باز می‌شود و اجازه عبور جریان را به کنتاکتور C2 نمی‌دهد و برعکس.

مثال - راه اندازی دو موتور توسط یک شستی

در یک پروژه صنعتی، از یک پمپ روغن و یک موتور استفاده شده است. نحوه روشن شدن این مصرف کننده ها بدین صورت می‌باشد که با فشردن شستی استارت I0.0 پمپ روغن روشن شده و لازم است که دو موتور با اندکی تاخیر وارد مدار شوند. این تاخیر متغیر بوده و بسته به صلاحدید اپراتور می‌باشد. در این صورت، اپراتور برای ایجاد تاخیر لازم دست را بر روی شستی نگاه داشته و در زمان مناسب می‌تواند با برداشتن دست از روی شستی استارت، موتور را وارد مدار کند. اگر در حین کار به هر دلیلی خروجی Q0.0 خاموش شد، موتور نیز خاموش شود. در نتیجه، با فشردن شستی استارت، Q0.0 روشن شده و با برداشتن دست از روی شستی استارت Q0.1 نیز وارد مدار شود.

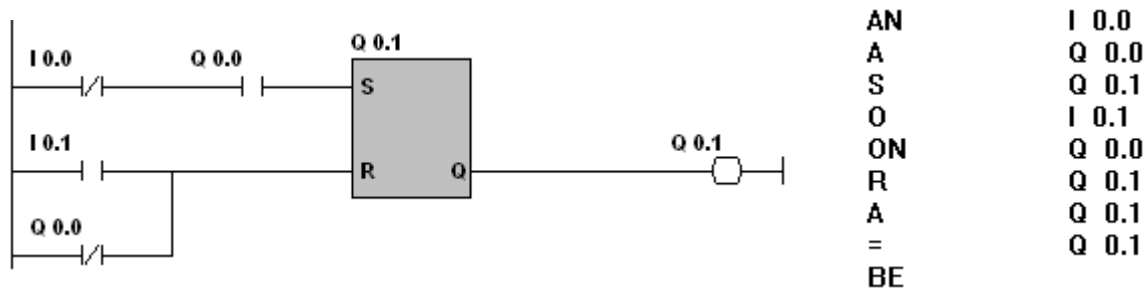


```

A      I 0.0
S      Q 0.0
A      I 0.1
R      Q 0.0
A      Q 0.0
=      Q 0.0
***

```


ادامه برنامه

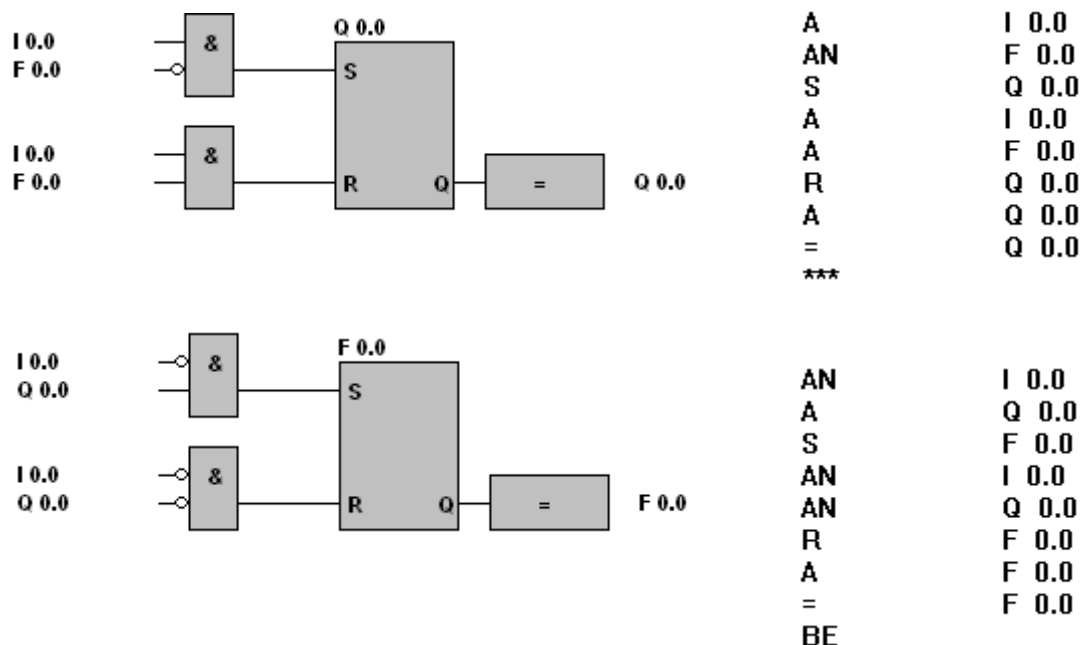


در برنامه این مثال از I0.0 بصورت دوبل استفاده شده است. اگر تیغه باز Q0.0 در مسیر استارت Q0.1 استفاده نشود، به محض RUN شدن PLC، خروجی Q0.1 روشن می شود.

مثال - پوش باتن

فرض کنید در یک پروسه صنعتی از یک کلید پوش باتن استفاده شده است. می خواهیم برنامه ای برای کنترل خروجی مرتبط با این پوش باتن بنویسیم. نحوه کار پوش باتن بدین صورت است که با فشردن شستی، خروجی فعال شده و با فشردن مجدد همان شستی، خروجی غیرفعال شود.

برنامه



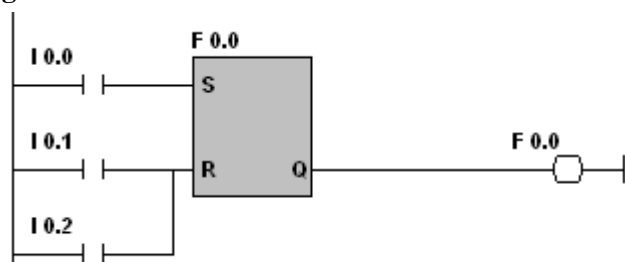
مثال- کنترل یک موتور در صنایع چوب بری

در یک ماشین چوب بری از یک موتور سه فاز استفاده شده است. برای اتصال دائمی این موتور به شبکه از یک شستی و برای قطع آن از شستی دیگری استفاده می شود. علاوه بر این دو شستی، توسط یک پدال نیز باید بتوان موتور را به طور موقت به شبکه متصل کرد. از این پدال در مواقعی که بخواهیم زمان کار موتور تحت کنترل باشد، استفاده می شود و باید تا زمانی که توسط پا بر روی آن فشار وارد می شود، موتور به چرخش خود ادامه داده و زمانی که پا از روی پدال برداشته شد، موتور نیز قطع شود.

لیست I/O :

- I0.0 : شستی استارت
- I0.1 : پدال
- I0.2 : شستی استپ
- Q0.0 : خروجی موتور

Segment 1



```

A      I 0.0
S      F 0.0
O      I 0.1
O      I 0.2
R      F 0.0
A      F 0.0
=      F 0.0
***
    
```

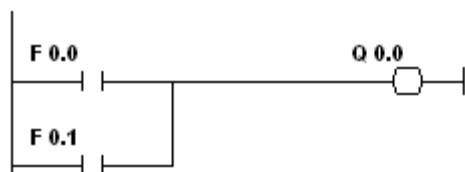
Segment 2



```

A      I 0.2
=      F 0.1
***
    
```

Segment 3



```

O      F 0.0
O      F 0.1
=      Q 0.0
BE
    
```

افرادی که با مدارات فرمان آشنایی دارند، در بسیاری از برنامه های خود مانند مدارات فرمان از تیغه باز یک خروجی در ورودی استفاده می کنند. با انجام این موضوع، یک مدار Latching ایجاد شده که عملکردی مشابه دستور SET یا تیغه نگهدارنده دارد.

مثال - کنترل ۳ ولو

در یک پروسه صنعتی از ۳ ولو برای شارژ مواد به یک مسیر استفاده شده است. فرمان باز شدن هر ولو از یک شستی استارت به صورت جداگانه و فرمان بسته شدن همه ولوها نیز از یک شستی استپ صادر می شود. نحوه باز شدن ولوها بدین صورت می باشد که تا زمانی که ولو V1 فعال یا باز نشده است، ولو V2 باز نشود. همچنین، تا زمانی که ولوهای V1 و V2 باز نشده باشند، امکان باز شدن ولو V3 نیز وجود نداشته باشد. اگر شرایط باز شدن ولو V3 فراهم شد، با فشردن شستی استارت مربوط، این ولو باز شده و دو ولو V1 و V2 بسته شوند.



I0.0 : فرمان باز شدن ولو V1

I0.1 : فرمان باز شدن ولو V2

I0.2 : فرمان باز شدن ولو V3

I0.3 : فرمان بسته شدن هر سه ولو

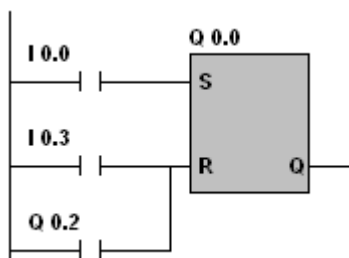
Q0.0 : خروجی ولو V1

Q0.1 : خروجی ولو V2

Q0.2 : خروجی ولو V3

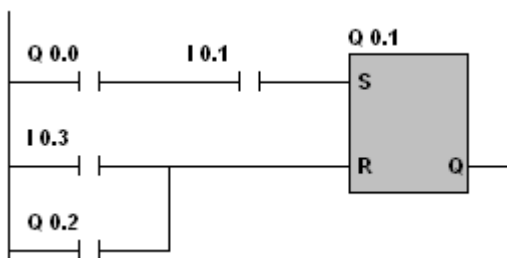
Segment 1

برنامه



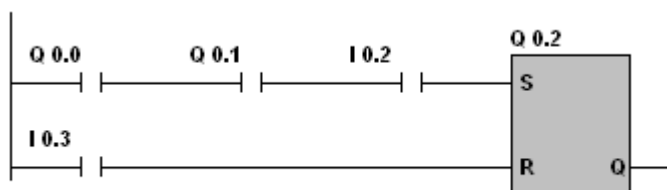
A	I 0.0
S	Q 0.0
O	I 0.3
O	Q 0.2
R	Q 0.0

Segment 2



A	Q 0.0
A	I 0.1
S	Q 0.1
O	I 0.3
O	Q 0.2
R	Q 0.1

Segment 3



A	Q 0.0
A	Q 0.1
A	I 0.2
S	Q 0.2
A	I 0.3
R	Q 0.2

همانطور که در برنامه این مثال مشاهده می کنید، فعال شدن هر ولو به عنوان شرط باز شدن ولو بعدی می باشد. همچنین، باز شدن ولو V3 موجب ریست شدن و یا به عبارت دیگر، بسته شدن دو ولو V1 و V2 می شود.

تایمرها TIMERS

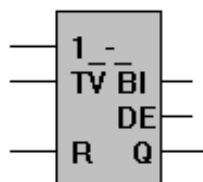
همانطور که می دانیم، یکی از تجهیزات مهم در مدارات فرمان کنتاکتوری، تایمرها می باشند که عمل زمان سنجی را انجام می دهند. استفاده از تایمر در مدارات فرمان هم موجب افزایش قیمت تابلو و هم باعث افزایش حجم سیم کشی می شود. یکی دیگر از مزایای استفاده از کنترل کننده های صنعتی، دارا بودن تایمرهای نرم افزاری می باشد.



در S5 سه نوع تایمر وجود دارد :

- | | |
|--------------------------------|-------------------------|
| 1- تایمر پله ای SP | (Pulse Timer) |
| 2- تایمر پله ای گسترده SE | (Extended Pulse Timer) |
| 3- تایمر با تاخیر روشن SD | (On-Delay Timer) |
| 4- تایمر تاخیر روشن ماندگار SS | (Stored On-Delay Timer) |
| 5- تایمر با تاخیر خاموش SF | (Off-Delay Timer) |

در ادامه مطالب نوبت به نحوه تعریف نمودن مدت زمان کارکرد تایمر یا به عبارت دیگر دادن زمان به تایمر می باشد. این مطلب بین تمامی تایمرها مشترک و مورد استفاده قرار می گیرد. در شکل زیر روش فلوجارتی تایمرها را ملاحظه می کنید.



ورودی های تایمر:

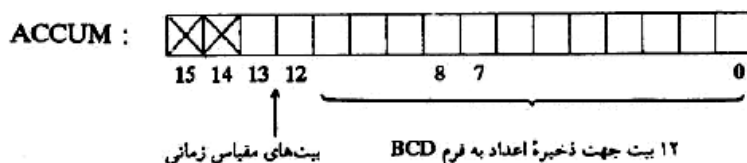
ورودی S: این ورودی در حقیقت فعال ساز یا راه انداز تایمر می باشد.

ورودی TV: عددی است که مدت زمان کارکرد تایمر را تعیین می کند.

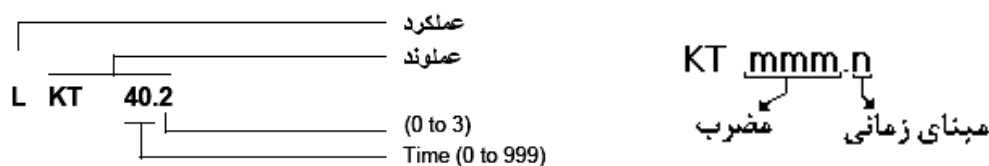
ورودی R: توسط این ورودی می توان تایمر را ریست می باشد.

ورودی TV که توسط آن می توان زمان مورد نیاز برای تایمر را تعیین نمود، بحث اصلی ما در این قسمت می باشد.

معمولا این ورودی با فرمت KT پر می شود. در واقع از فرمت KT جهت بارگذاری مدت زمان مورد نیاز برای تایمر استفاده می شود. پس از بارگذاری، مقدار KT در انبارک قرار می گیرد. KT در انباره ها شامل ۱۴ بیت بوده، به فرم BCD می باشد. دو بیت آخر در انبارک بدون استفاده می باشند. دو بیت باارزش بالاتر KT بیت های ضریب یا مقیاس زمانی نامیده می شوند. از ۱۲ بیت باقیمانده برای ذخیره اعداد BCD از 000 تا 999 استفاده می شود.



باید توجه داشته باشید که در دستورالعمل بارگذاری تایمر مقدار زمانی تایمر از دو بخش مضرب زمانی و مقیاس زمانی تشکیل شده است.



عدد n می تواند طبق جدول زیر تعریف شود.

عدد	0	1	2	3
ضریب	0.01 s	0.1 s	1 s	10 s

کمترین مقدار برای TV برابر KT 001.0 و بیشترین مقدار آن KT 999.3 می باشد. پس حداقل زمانی که تایمر می تواند داشته باشد 0.01 ثانیه و بیشترین مقدار ۹۹۹۰ ثانیه می باشد. مثال: فرض نمایید می خواهیم زمانی معادل ۵ ثانیه را بارگذاری نماییم.

- معادل ۵ ثانیه
- ۱- L KT 5.2
 - ۲- L KT 50.1
 - ۳- L KT 500.0

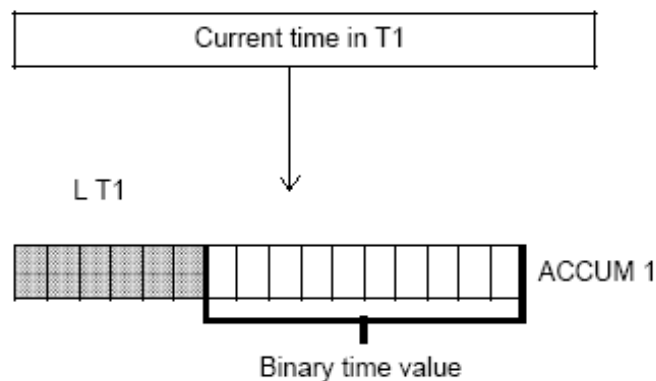
در جدول شکل زیر زمان ۴۰ ثانیه همراه با دقت تبدیل را ملاحظه می نمایید.

مثال	عملوند	زمان
تعریف کردن زمان 40 ثانیه	KT 400.1	400 x 0.1 sec. - 0.1 sec. 39.9 sec. to 40 sec.
	KT 40.2	40 x 1 sec. - 1 sec. 39 sec. to 40 sec.
	KT 4.3	4 x 10 sec. - 10 sec. 30 sec. to 40 sec.

نکته: در بارگذاری زمان، هر چه قدر مقدار مضرب کوچکتر باشد زمان دقیقتر می باشد. پس در مثال فوق زمان KT 400.1 دارای زمان دقیق تری نسبت به بقیه موارد می باشد.

یادآوری

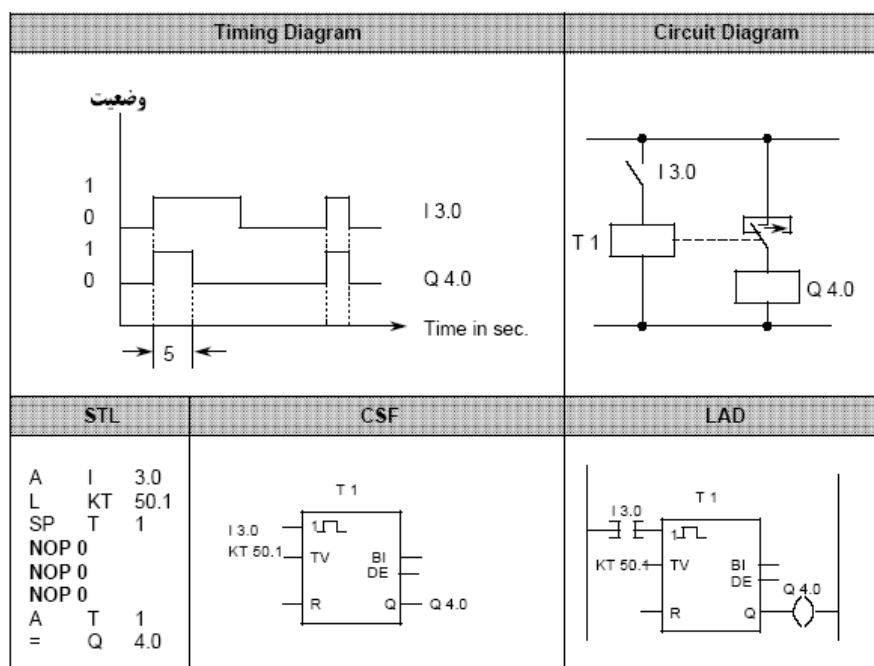
مقدار KT، ۱۴ بیت از انباره را تحت تاثیر قرار می دهد که ۲ بیت آن جهت ذخیره مقیاس و ۱۲ بیت آن نیز جهت ذخیره مضرب می باشد.



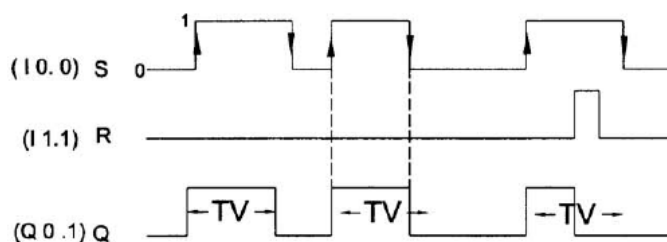
حال که با مفاهیم و اصطلاحات استفاده شده در تایمرها آشنا شدیم نوبت به ارائه توضیح در مورد انواع تایمرها می رسد.

تایمر SP

در این تایمر خروجی هم به لبه بالارونده و هم به لبه پایین رونده ورودی حساس است. یعنی با فعال کردن ورودی S خروجی فعال شده و پس از سپری شدن TV ثانیه خروجی غیر فعال می شود. حال اگر در حین سپری شدن زمان، ورودی S غیر فعال شود خروجی هم بلافاصله غیر فعال می شود. به عبارت دیگر خروجی تایمر به ورودی S بستگی دارد. به دیاگرام مداری این تایمر توجه نمایید.



در شکل زیر دیاگرام تایمر SP را به طور کامل ملاحظه می نمایید.

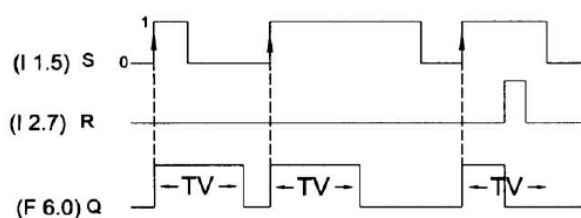
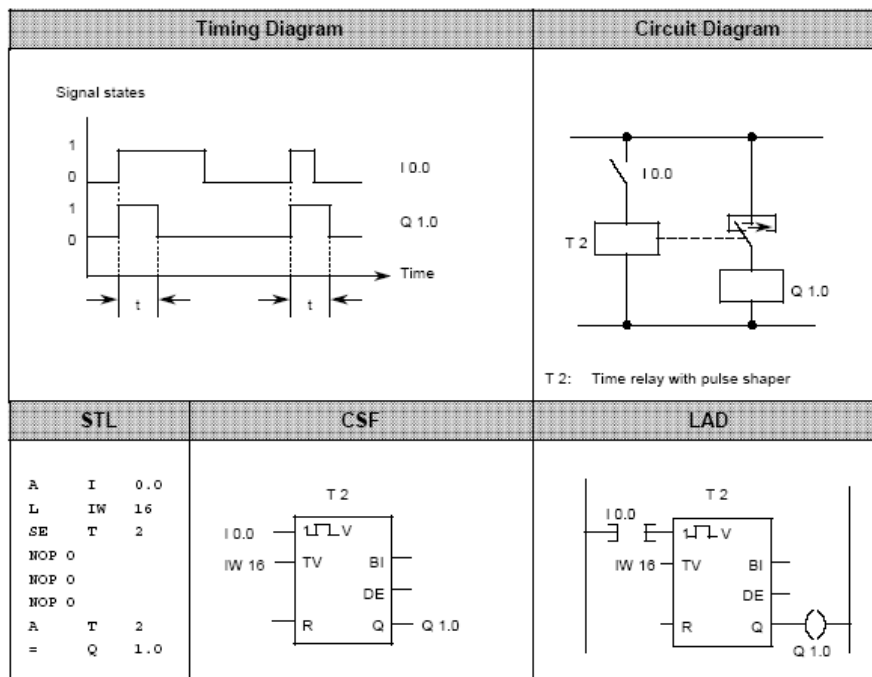


در این تایمر همانطور که ملاحظه کردید، زمانی که ورودی S غیر فعال می شود خروجی نیز غیر فعال، و با فعال شدن ورودی R خروجی تایمر به همراه زمان آن ری ست می شود.

تایمر SE

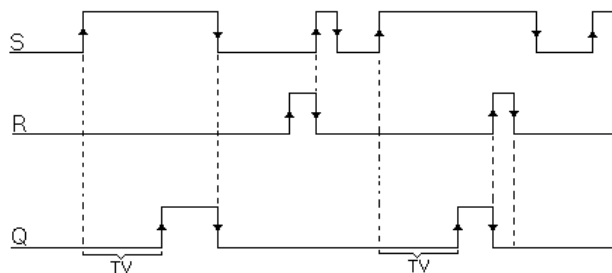
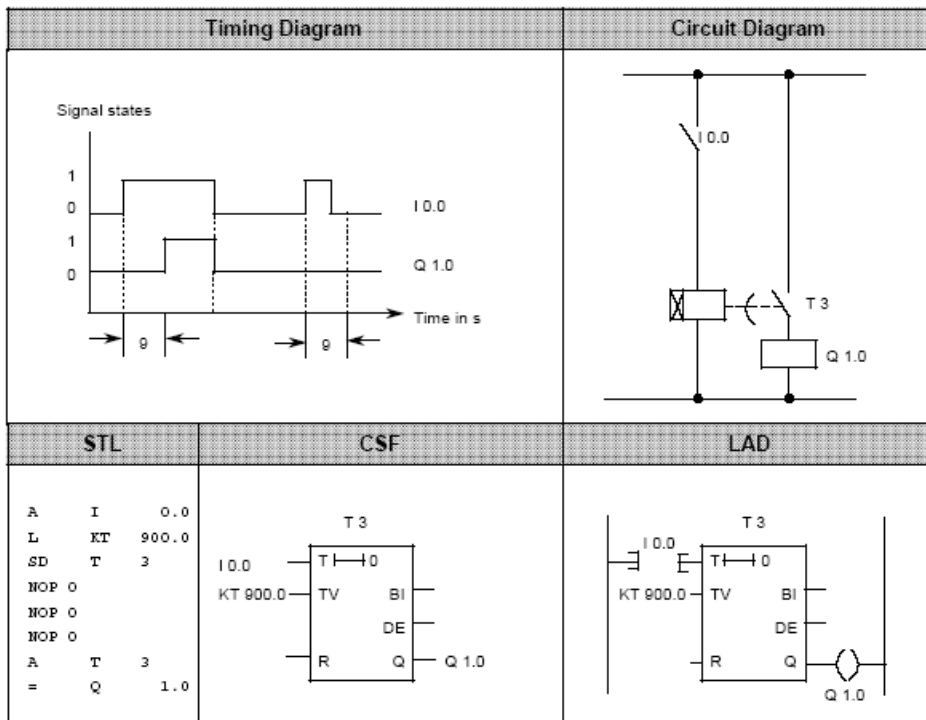
خروجی این تایمر فقط به لبه بالارونده ورودی حساس می باشد. یعنی کافیست که ورودی ست برای یک لحظه فعال شود تا تایمر شروع به کار نماید. پس از سپری شدن زمان TV خروجی غیر فعال می شود. این تایمر به تایمر راه پله معروف می باشد. به عبارت دیگر در صورتی که در مدت زمانی کمتر از TV ثانیه در ورودی S یک لبه پایین رونده داشته باشیم این لبه بر خروجی بی تاثیر بوده و پس از گذشت مدت زمان TV، خروجی غیر فعال می شود.

120



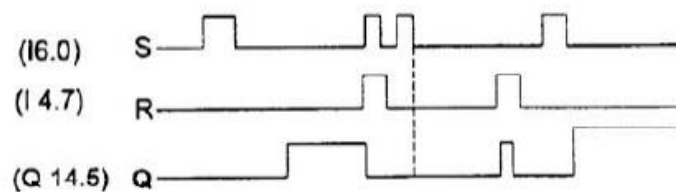
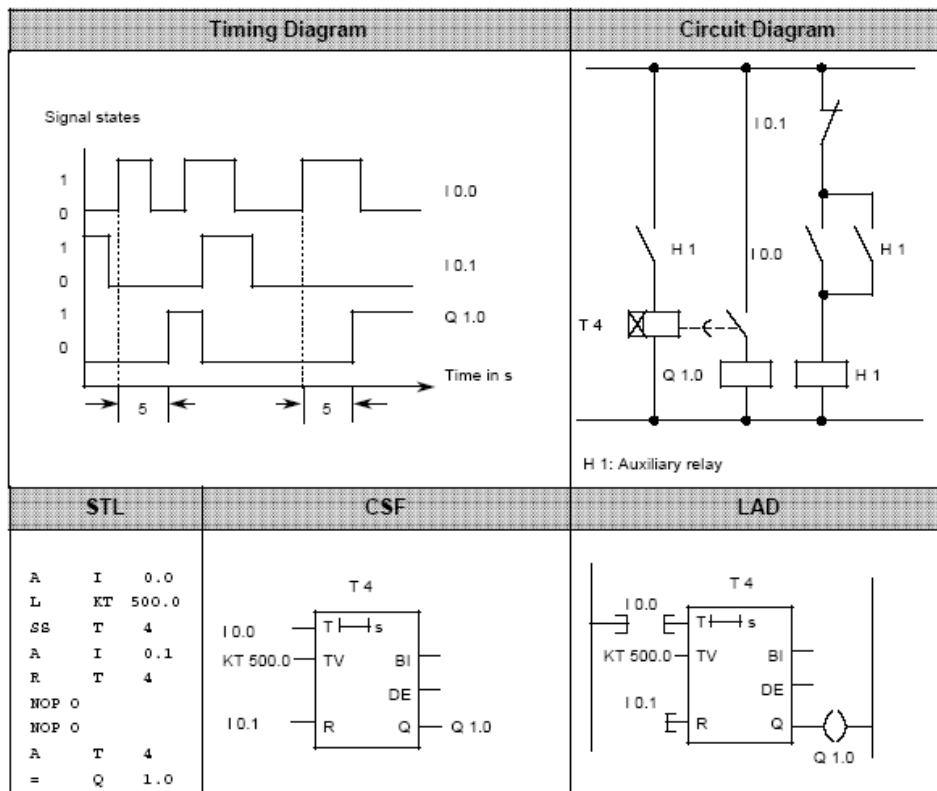
تایمر تاخیر در وصل ناپایدار (SD)

این تایمر یک تایمر تاخیر در وصل می باشد. خروجی این تایمر هم به لبه بالارونده ورودی و هم به لبه پایین رونده ورودی حساس است. با لبه بالارونده S، خروجی تایمر پس از مدت زمان TV ثانیه فعال و با لبه پایین رونده S، غیرفعال می شود. اگر کمی دقت نمایید درمی یابید در این تایمر مدت زمان TV تاخیر لازم جهت روشن شدن خروجی می باشد. یعنی با فعال شدن ورودی S، خروجی فعال نمی شود بلکه تایمر شروع به شمارش کرده و پس از سپری شدن زمان TV خروجی فعال می گردد.



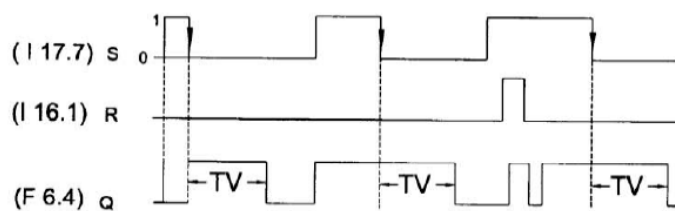
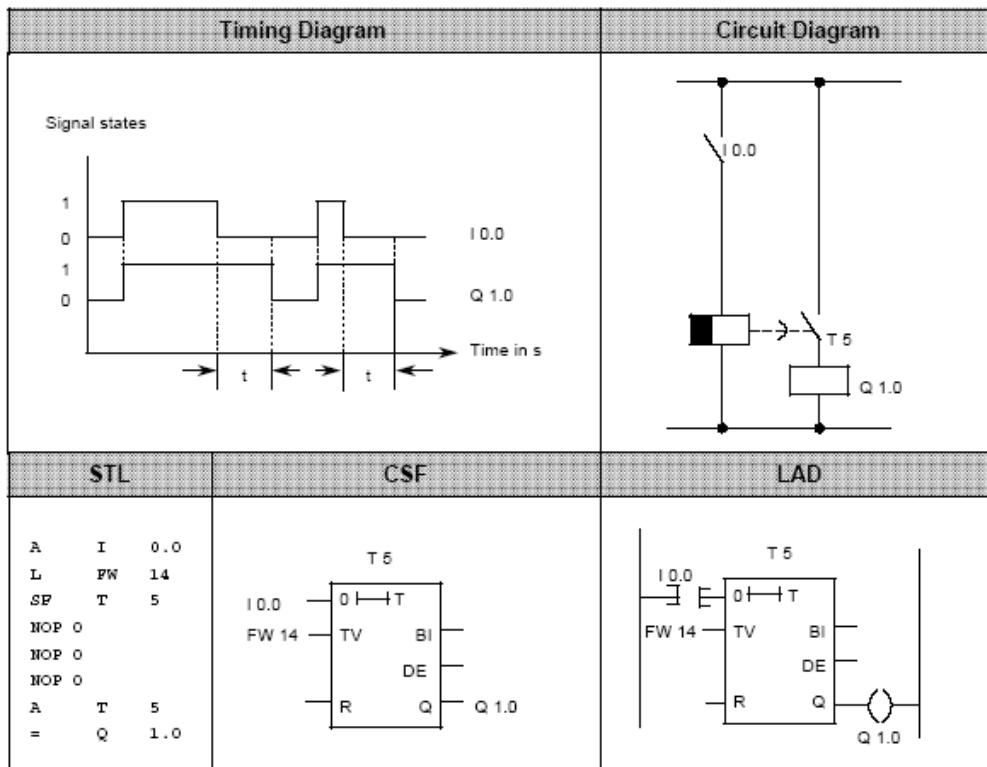
تایمر تاخیر در وصل ماندگار (SS)

SS نیز یک تایمر تاخیر در وصل بوده که عملکرد آن تقریباً شبیه تایمر SD می باشد. این تایمر تنها به لبه بالارونده ورودی S حساس است. این تایمر با لبه بالارونده ورودی S پس از TV ثانیه فعال شده، در همین وضعیت باقی می ماند و تنها با فعال شدن ورودی R غیرفعال می شود. عملکرد این تایمر دقیقاً برعکس تایمر SE می باشد.



تایمر تاخیر در قطع (SF)

خروجی این تایمر با لبه بالارونده ورودی S فعال و با لبه پایین رونده ورودی S پس از TV ثانیه غیرفعال می گردد. در واقع در این تایمر برای شروع زمان TV ثانیه جهت قطع خروجی نیاز به یک لبه پایین رونده در ورودی S می باشد.



در ادامه جهت آشنایی بیشتر با تایمرها به بررسی چندین مثال می پردازیم

مثال ۱

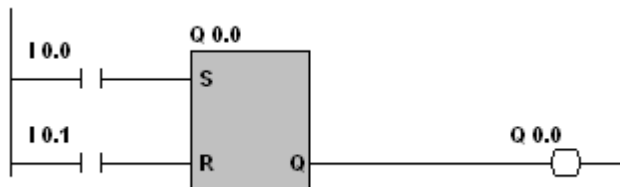
با فشردن شستی استارت IO.0 پمپ ۱ روشن و پس از سپری شدن ۳ دقیقه پمپ ۲ نیز روشن شود. با فشردن شستی استپ IO.1 هر دو پمپ خاموش شوند.



124

برنامه

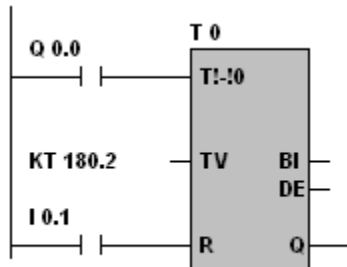
Segment 1



```

A      I 0.0
S      Q 0.0
A      I 0.1
R      Q 0.0
A      Q 0.0
=      Q 0.0
    
```

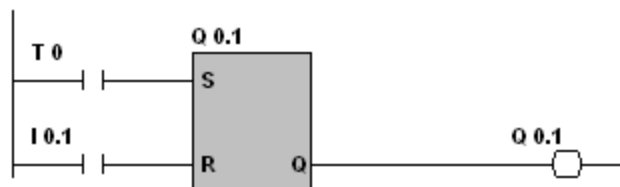
Segment 2



```

A      Q 0.0
L      KT 180.2
SD     T 0
A      I 0.1
R      T 0
NOP    0
NOP    0
NOP    0
***
    
```

Segment 3



```

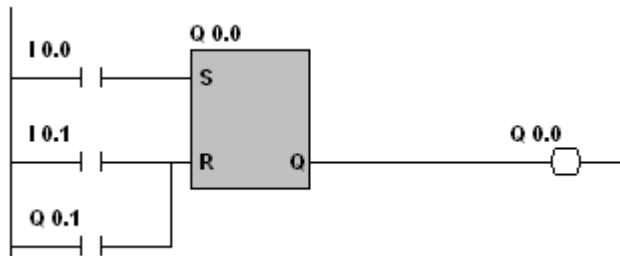
A      T 0
S      Q 0.1
A      I 0.1
R      Q 0.1
A      Q 0.1
=      Q 0.1
BE
    
```

- تایمرهای استفاده شده در این مثال نوع SR می باشد که در روش STL با نام SD نامگذاری می شود.
- جای خالی ورودی ها و خروجی های استفاده نشده در روش LAD یا CSF، در روش STL با عبارت NOP 0 نشان داده می شود.

مثال ۲

مثال قبل را به گونه ای تغییر دهید که روشن شدن پمپ ۲ باعث خاموش شدن پمپ ۱ شود.

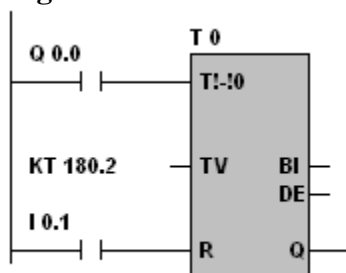
Segment 1



```

A      I 0.0
S      Q 0.0
O
O      I 0.1
R      Q 0.1
R      Q 0.0
A      Q 0.0
=
***
    
```

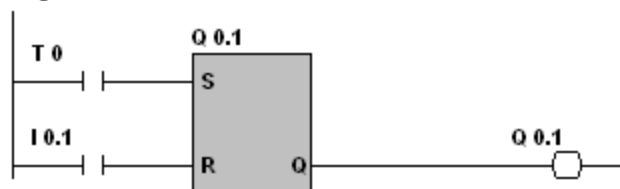
Segment 2



```

A      Q 0.0
L      KT 180.2
SD
A      I 0.1
R      T 0
NOP
NOP
NOP
***
    
```

Segment 3



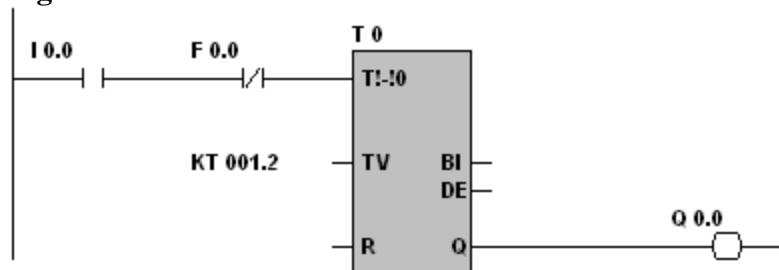
```

A      T 0
S      Q 0.1
A      I 0.1
R      Q 0.1
A      Q 0.1
=
BE
    
```

مثال ۳ - تولید پالس

می خواهیم برنامه‌ای بنویسیم که با فعال شدن ورودی I0.0، در Q0.0 قطار پالس (یک ثانیه روشن و یک ثانیه خاموش) تولید شود.

Segment 1

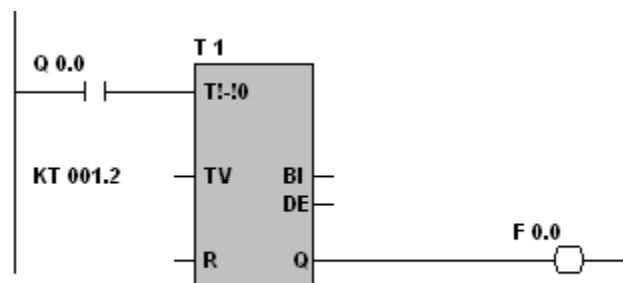


```

A      I 0.0
AN     F 0.0
L      KT 001.2
SD     T 0
NOP    0
NOP    0
NOP    0
A      T 0
=      Q 0.0
***

```

Segment 2



```

A      Q 0.0
L      KT 001.2
SD     T 1
NOP    0
NOP    0
NOP    0
A      T 1
=      F 0.0
***

```

توضیح :

با فعال شدن ورودی I0.0، تایمر T0 شروع به کار می کند و پس از ۱ ثانیه خروجی Q0.0 روشن می شود. با روشن شدن Q0.0 تایمر T1 وارد مدار شده و ۱ ثانیه بعد F0.0 را فعال می کند. فعال شدن F0.0 موجب از کار افتادن T0 و در نتیجه، قطع Q0.0 می شود. این روند در سیکل های بعدی نیز ادامه یافته و موجب تولید پالس در خروجی Q0.0 می شود. تایمرها مدل SR می باشند.

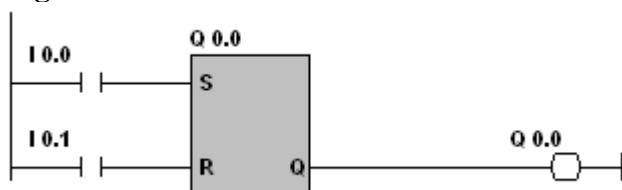
مثال ۴

در یک پروژه صنعتی از یک موتور با توان بالا (Q0.0) در یک محیط بسته استفاده شده و در کنار این موتور نیز یک فن خنک کن (Q0.1) قرار داده شده است. نحوه کار بدین صورت است که با فشردن استارت I0.0، هر دو مصرف کننده وارد مدار می‌شوند. اما زمانی که فرمان قطع موتور توسط I0.1 صادر گردید، موتور بلافاصله متوقف شده و فن خنک کن ۳ دقیقه بعد از خاموش شدن موتور، خاموش می‌شود. مطلوب است نوشتن برنامه این مثال توسط تایمر SR در این مثال، برای آشنایی از دستورات Set و Reset مجزا استفاده می‌کنیم.



برنامه

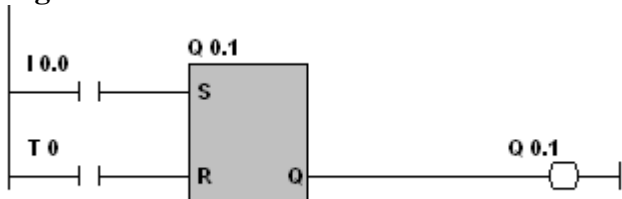
Segment 1



```

A      I 0.0
S      Q 0.0
A      I 0.1
R      Q 0.0
A      Q 0.0
=      Q 0.0
***
    
```

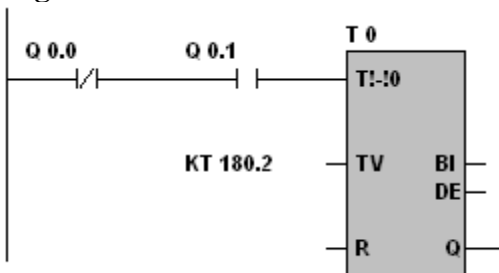
Segment 2



```

A      I 0.0
S      Q 0.1
A      T 0
R      Q 0.1
A      Q 0.1
=      Q 0.1
***
    
```

Segment 3



```

AN     Q 0.0
A      Q 0.1
L      KT 180.2
SD     T 0
NOP    0
NOP    0
NOP    0
NOP    0
BE
    
```

مثال ۵

فرض کنید در یک پروژه صنعتی از ۳ ولو در سه مسیر استفاده شده است. نحوه باز و بسته شدن ولوها بدین صورت است که با فشردن شستی استارت، ولو ۱ باز شده و ۱ دقیقه بعد ولو ۲ باز شده و ولو ۱ بسته می شود. در ادامه نیز ۱ دقیقه بعد، ولو ۳ باز شده و ولو ۲ بسته می شود. در مرحله پایانی نیز ۱ دقیقه بعد، ولو ۱ مجدداً باز شده و ولو ۳ بسته می شود. این روند تا زمانی که شستی استپ فشرده نشده است، ادامه می یابد. در ضمن، لازم به ذکر است که اگر در وسط سیکل، شستی استارت مجدداً فشرده شد، هیچ تاثیری در عملکرد مدار نداشته باشد.

لیست I/O

استارت: I0.0

استپ: I0.1

ولو ۱: Q0.0

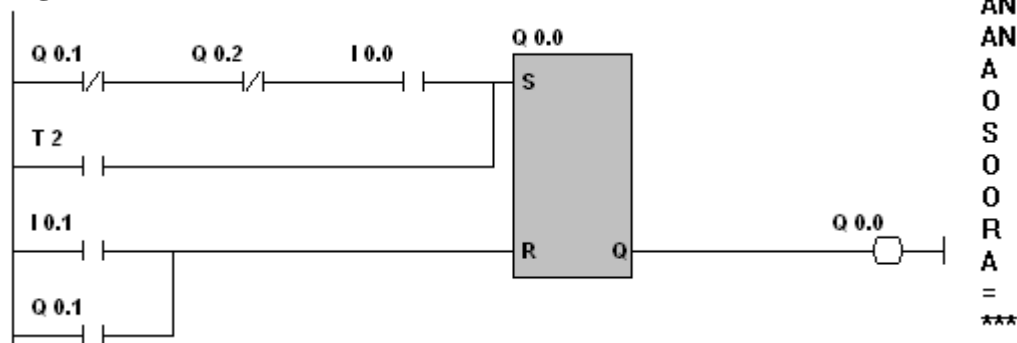
ولو ۲: Q0.1

ولو ۳: Q0.2

128

برنامه

Segment 1

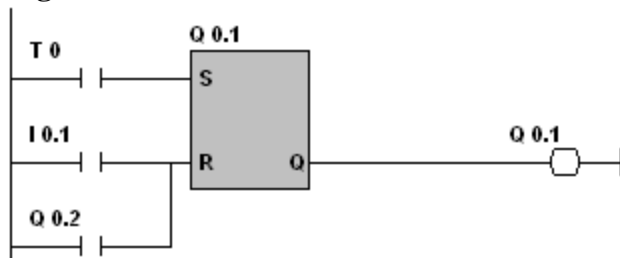


Segment 2



ادامه برنامه

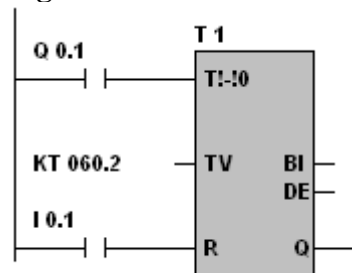
Segment 3



```

A      T 0
S      Q 0.1
O      I 0.1
O      Q 0.2
R      Q 0.1
A      Q 0.1
=      Q 0.1
***
    
```

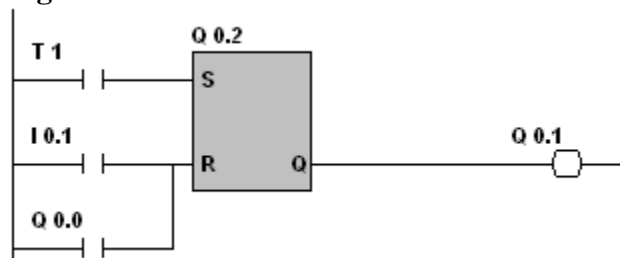
Segment 4



```

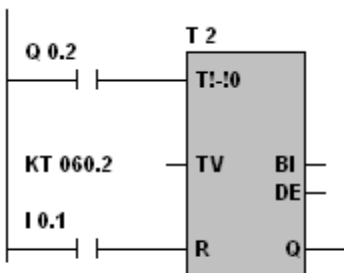
A      Q 0.1
L      T 1
SD     T 1
A      I 0.1
R      T 1
NOP    0
NOP    0
NOP    0
***
    
```

Segment 5



```

A      T 1
S      Q 0.2
O      I 0.1
O      Q 0.0
R      Q 0.2
A      Q 0.2
=      Q 0.1
***
    
```



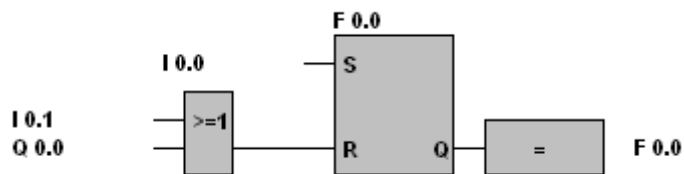
```

A      Q 0.2
L      T 2
SD     T 2
A      I 0.1
R      T 2
NOP    0
NOP    0
NOP    0
BE
    
```

مثال ۶

در این مثال می‌خواهیم توسط تایمر SR برنامه‌ای طراحی کنیم که با فعال کردن ورودی I0.0، یک دقیقه بعد خروجی Q0.0 فعال شود.

Segment 1



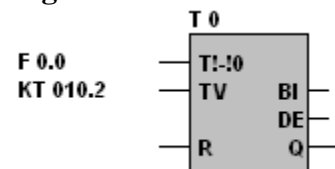
```

A      I 0.0
S      F 0.0
O      I 0.1
O      Q 0.0
R      F 0.0
A      F 0.0
=      F 0.0
***

```

130

Segment 2

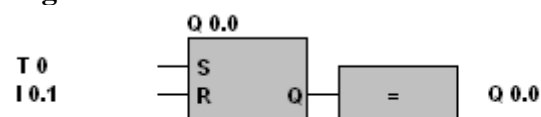


```

A      F 0.0
L      KT 010.2
SD     T 0
NOP    0
NOP    0
NOP    0
NOP    0
***

```

Segment 3



```

A      T 0
S      Q 0.0
A      I 0.1
R      Q 0.0
A      Q 0.0
=      Q 0.0
***

```

❖ در این مثال، برای فعال نگه داشتن تایمر از F0.0 استفاده شده است.

دستورات بارگذاری و انتقال

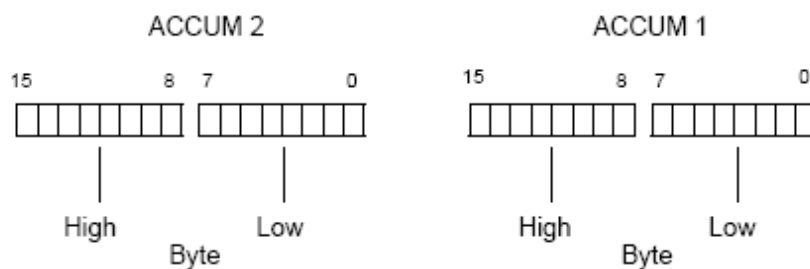
قبل از اینکه وارد این بحث شویم ابتدا لازم است که مطالبی در مورد آکومولاتور بیان شود.

آکومولاتور یک حافظه یا رجیستر ۱۶ بیتی می باشد. اکثر PLC ها دارای دو آکومولاتور هستند که با نام های ACCU1 و ACCU2 مشخص می شوند. البته ناگفته نماند که برخی از CPU ها دارای ۴ آکومولاتور می باشند.

PLC با تایمرها، شمارنده ها و اعداد سرکار دارد. جهت مبادله اعداد احتیاج به یک واسطه می باشد. بنابراین قسمتی از حافظه PLC که به آن انباره یا آکومولاتور می گوئیم به این کار اختصاص داده شده است. زمانی که از دستور Load استفاده می شود، داده ای را که توسط این دستور بار می شود، ابتدا وارد ACCU1 می گردد و زمانی که توسط دستور انتقال T مقدار باید انتقال داده شود از ACCU1 خوانده و ارسال می شود. در مورد دستورات L و T بطور مفصل بحث خواهد شد.

آکومولاتور ۲

همانطور که گفته شد، با اجرای اولین دستور Load، محتوای آن وارد ACCU1 می شود. زمانی که برای دومین بار دستور Load، برای بار کردن داده ای دیگر اجرا می شود، مقدار قبلی محتویات ACCU1 وارد ACCU2 شده و داده حاصل از Load دوم وارد ACCU1 می شود. آکومولاتور از دو بایت با ارزش و کم ارزش تشکیل شده است.



انواع دسترسی به فضاهای حافظه در S5

بایت: Byte - شامل ۸ بیت می باشد

0 FY: بایت شماره 0 از حافظه فلگ

5 IB: بایت شماره 5 ورودی

2 QB: بایت شماره 2 خروجی

Word - شامل دو بایت یعنی ۱۶ بیت می باشد.

FW 0: کلمه شماره 0 از حافظه (شامل FY 0 و FY 1)

IW 6: کلمه شماره 6 از ورودی (شامل IB 6 و IB 7)

QW 2: کلمه شماره 2 از خروجی (شامل QB 2 و QB 3)

Double word - شامل ۴ بایت یعنی ۳۲ بیت می باشد.

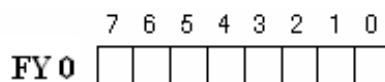
FD 30: دو کلمه‌ای شماره 30 از حافظه (شامل بایت های FY 30, FY 31, FY 32, FY 33)

ID 0: دو کلمه‌ای شماره 0 از ورودی (شامل IB 0, IB 1, IB 2, IB 3)

QD 2: دو کلمه‌ای شماره 2 از خروجی (شامل QB 2, QB 3, QB 4, QB 5)

یادآوری

در S5 یک بایت از حافظه فلگ را با FY نشان می دهند.

**دستور بارگذاری L**

توسط این دستور می توان داده ها و اطلاعاتی را که در یک بایت ، کلمه و یا دو کلمه وجود دارند را وارد ACCU 1 نمود.

مثال:

0 IB L: با اجرای این دستور اطلاعات بایت شماره صفر ورودی خوانده و وارد آکومولاتور ۱

می شود.

15 FW L: با اجرای این دستور اطلاعات و محتوای موجود در کلمه شماره ۱۵ از حافظه خوانده و وارد

آکومولاتور ۱ می شود.

32 QB L: با اجرای این دستور اطلاعات بایت شماره ۳۲ خروجی خوانده و وارد آکومولاتور ۱

می شود.

دستور انتقال T

توسط دستور T می توان محتویات ACCU 1 را به خروجی و یا یک فلگ منتقل نمود. دستور T را می توان در مورد تایمرها و شمارنده ها نیز به کار برد.

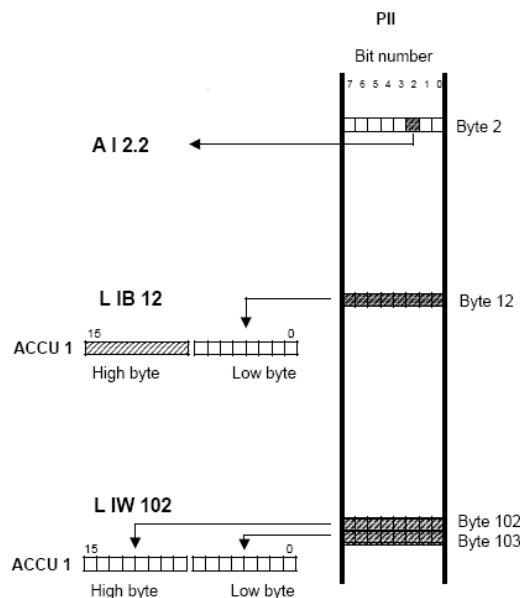
0 T FY: انتقال اطلاعات موجود در ACCU 1 به بایت شماره 0 از حافظه فلگ

0 T QB: انتقال اطلاعات موجود در ACCU 1 به بایت شماره 0 خروجی

12 T QW: انتقال محتویات موجود در ACCU 1 به کلمه شماره 12 خروجی

133

شکل زیر نحوه فراخوانی یک بیت، بایت و یک کلمه را نشان می دهد.

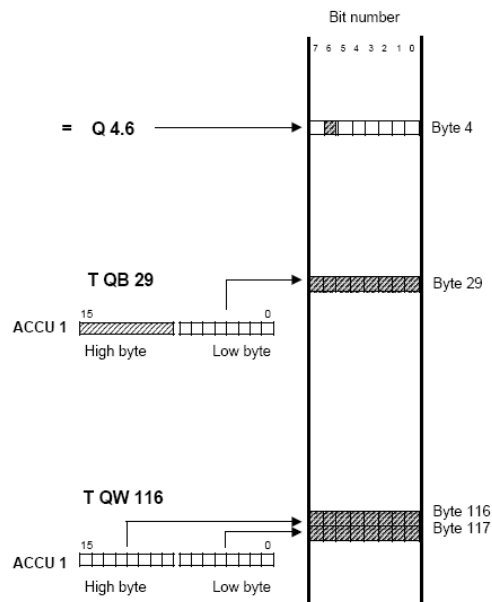


A I 2.2: تاثیر گذاشتن بر روی یک بیت از یک بایت ورودی

L IB 12: تاثیر گذاشتن بر روی یک بایت از ورودی

L IW 102: تاثیر گذاشتن بر روی یک کلمه از ورودی

شکل زیر نحوه انتقال اطلاعات بصورت بیتی، بایتی و کلمه ای را نشان می دهد.



= Q 4.6 : انتقال وضعیت بیت RLO به یک بیت از خروجی
 T QB 29 : انتقال محتوای آکومولاتور ۱ به یک بایت از خروجی
 T QW 116 : انتقال محتوای آکومولاتور ۱ به یک کلمه از خروجی

در PLC های سری S7 از دستور MOVE در روش FBD یا LAD به جای دستورات L و T استفاده می کنند.

دستورات ریاضی

در این قسمت با دستورات مربوط به عملیات جمع و تفریق آشنا می شویم.

Arithmetic Operations									
+F		N	N	N	55	26	1,6	0,8	Add two fixed-point numbers: ACCU 1+ACCU 2. CC 1/CC 0/OV are affected.
-F		N	N	N	58	23	1,6	0,8	Subtract one fixed-point number from another: ACCU 2 - ACCU 1. CC 1/CC 0/OV are affected.

توسط دستورات +F و -F می توان دو مقدار را با هم جمع و یا از هم کم نمود. به مثال زیر توجه نمایید.

```

L      IB 0
L      IB 1
+F
T      QB 0
BE
    
```

در مثال فوق PLC مقادیر ارسالی از دو بایت ورودی را خوانده و با هم جمع و نتیجه را به بایت صفر خروجی ارسال می نماید.

نکته:

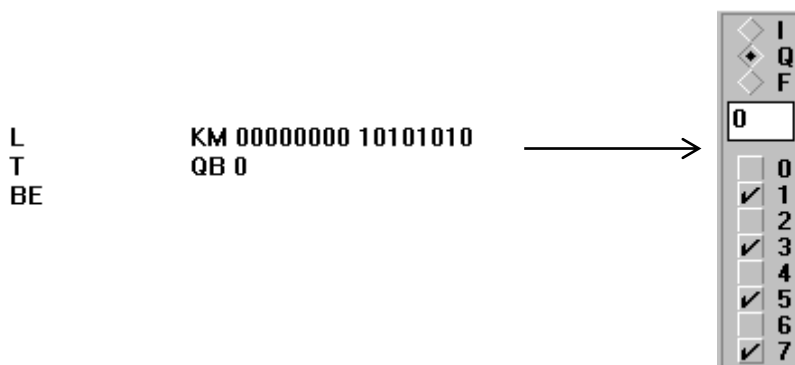
اعداد دریافتی از ورودی بصورت باینری بوده و جواب نیز بصورت باینری در خروجی ها ظاهر می شود.

فرمت اعداد در S5

در اکثر برنامه های صنعتی، برنامه نویس نیاز دارد در برنامه از اعداد ثابت در فرمت های مختلف استفاده کند. به عنوان مثال فرض کنید از یک سنسور جهت شمارش تعداد بسته های تولیدی در یک خط تولید استفاده شده است. می خواهیم برنامه را به گونه ای طرح کنیم که زمانی که تعداد بسته های موجود در کارتن به ۲۴ عدد رسید جک کاتون پر شده را به مسیر دیگر هدایت نماید. در برنامه این پروسه برنامه نویس نیاز به عدد ثابت ۲۴ دسیمال دارد. در واقع برنامه نویس می تواند اعداد مورد نیاز خود را از PLC تقاضا کند. جهت بار کردن اعداد ثابت در PLC های S5 از فرمت KF استفاده می شود. محدوده این فرمت از 32768- الی 32767+ می باشد. در جدول شکل زیر عبارات مورد نیاز، برای بار گذاری اعداد در فرمت های مختلف لیست شده است.

KB	Constant (1 byte)	0 to 255
KC	Constant (count)	0 to 999
KF	Constant (fixed-point number)	- 32768 to +32767
KH	Constant (hexadecimal code)	0 to FFFF
KM	Constant (2-byte bit pattern)	arbitrary bit pattern (16 bits)
KS	Constant (2 characters)	any two alphanumeric characters
KT	Constant (time)	0.0 to 999.3
KY	Constant (2 bytes)	0 to 255 (per byte)

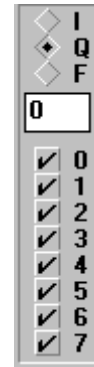
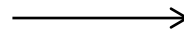
به عنوان مثال برای بار گذاری عدد در مبنای ۲ از عبارت KM استفاده می شود.



مثال

L
T
BE

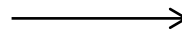
KH 00FF
QB 0



136

L
T
BE

KF +255
QB 0



مثال

مثال: جمع دو عدد دسیمال ۱۵۰ و ۳۷۵ و انتقال نتیجه به FW 0

L KF +150
L KF +375
+F
T FW 0
BE

مثال: افزایش ۲۰ واحد به مقدار دریافتی از ورودی IB 0

L IB 0
L KF +20
+F
T FW 0
BE

نکته ۱

تمامی دستوراتی که به دستور L و T وابسته می باشند، تنها به روش STL قابل نوشتن می باشند.

نکته ۲

با اجرای دستور T محتوای ACCUM 1 منتقل می شود.

نکته ۳

فرض کنید می خواهیم عدد ۱۰۰ را در یک کلمه از حافظه و عدد ۱۵۰ را در کلمه بعدی حافظه ذخیره نماییم.

برنامه ۱

```
L      KF +100
T      FW 0
L      KF +150
T      FW 1
BE
```

برنامه ۲

```
L      KF +100
T      FW 0
L      KF +150
T      FW 2
BE
```

توضیح

در برنامه ۱ عدد اول در FW0 و عدد دوم در FW 1 ذخیره ولی در برنامه ۲ عدد دوم در FW 2 ذخیره می شود. برنامه ۱ نادرست و برنامه ۲ درست می باشد.

دلیل

در برنامه ۱ از FW 0 جهت ذخیره عدد اول استفاده شده است. پس FW 0 شامل FY 0 و FY 1 می باشد. در ادامه ۲ از FW 1 استفاده شده است که شامل FY 1 و FY 2 می باشد. اگر کمی دقت نمایید درمی یابید که FY 1 بین دو عدد، مشترک در نظر گرفته می شود و عدد بدرستی ذخیره نمی شود. ولی در برنامه ۲ از FW 2 جهت ذخیره عدد دوم استفاده شده است که FW 2 شامل FY 2 و FY 3 می باشد و هیچ بایتی مشترک در نظر گرفته نمی شود. این قاعده نه تنها در مورد حافظه F، بلکه در مورد ورودی ها، خروجی ها نیز صادق می باشد.

قاعده جانشین سازی در آکومولاتورها

همانطور که در بحثهای قبلی بیان شد، هر PLC دارای تعداد محدودی آکومولاتور می باشد. زمانی که برنامه نویس قصد انجام عملیات ریاضی را بین چندین عدد دارد، حتما باید به قاعده جانشین سازی در مورد آکومولاتورها دقت لازم را داشته باشید. فرض کنید در PLC مورد بحث ما دو آکومولاتور وجود دارد. حال برنامه شکل زیر را در نظر بگیرید.

```
L      KF +25
L      KF +33
L      KF +48
+F
T      FY 0
BE
```

در خط اول این برنامه با اجرای دستور L KF 25 مقدار ۲۵ وارد ACCUM 1 می شود. اگر در همین حالت دستور بعدی یعنی L KF 33 اجرا گردد، ابتدا اطلاعات موجود در آکومولاتور ۱ یعنی ۲۵ وارد آکومولاتور ۲ شده و سپس مقدار ۳۳ به آکومولاتور ۱ منتقل می شود. چنانچه در این وضعیت دستور بارگذاری دیگری نظیر L KF 48 اجرا شود، محتویات فعلی آکومولاتور ۲ یعنی ۲۵ دور ریخته شده، عدد ۳۳ که در آکومولاتور ۱ بود، در آکومولاتور ۲ ذخیره می گردد. سپس L KF 48 در آکومولاتور ۱ قرار می گیرد. بنابراین ملاحظه می گردد که به دلیل وجود دو آکومولاتور و سه دستور بارگذاری، اولین عدد از بین می رود. پس در این گونه موارد برنامه نویس حتما باید ابتدا نتیجه جمع دو عدد را در یک حافظه ذخیره و در نهایت با عدد سوم عملیات جمع را انجام دهد.

برنامه صحیح عملیات جمع بین دو عدد را بار دیگر بازنویسی می کنیم. در این برنامه حاصل جمع دو عدد ۲۵ و ۳۳ در فلگ ۰ ذخیره می شود. پس محتوای فلگ ۰ عدد ۵۸ می باشد. در ادامه همان محتوای فلگ ۰ با عدد سوم یعنی ۴۸ جمع شده و جواب نهایی که عدد ۱۰۶ می باشد به فلگ ۱ منتقل می شود.

```
L      KF +25
L      KF +33
+F
T      FY 0
L      FY 0
L      KF +48
+F
T      FY 1
BE
```

مثال :

با استفاده از برنامه حاصل عبارت زیر را به دست آورید.

$$((125+200)-30)+15=FW18$$

```
L      KF +125
L      KF +200
+F
T      FW 0
L      FW 0
L      KF +30
-F
T      FW 2
L      FW 2
L      KF +15
+F
T      FW 18
BE
```

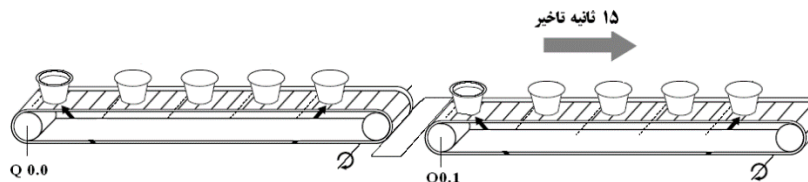
نکته: در این برنامه همانطور که مشاهده می کنید جهت انجام عملیات جدید بر روی یک FW مجدداً از دستور L در خط بعدی استفاده شده است که می توان از این خط در مراحل بعدی صرف نظر کنیم.

L	KF +125
L	KF +200
+F	
T	FW 0
L	KF +30
-F	
T	FW 2
L	KF +15
+F	
T	FW 18
BE	

یادآوری: در این برنامه به دلیل اینکه حاصل عملیات بیشتر از عدد ۲۵۵ می باشد، جهت ذخیره سازی از حافظه فلگ در محدوده Word استفاده شده است.

مثال ۱- کنترل نوار نقاله

شکل زیر را در نظر بگیرید. نحوه کار بدین صورت می باشد که با فشردن کلید استارت I0.1، موتور مربوط به کانوایر ۱ شروع به حرکت می کند. سپس بعد از ۱۵ ثانیه دیگر کانوایر ۲ بصورت اتوماتیک وارد مدار می شود. ضمناً با فشردن کلید استپ IO.0 کل مدار قطع می شود.



نکته:

جهت آشنایی بیشتر با روش مختلف برنامه نویسی، مثال های ارائه شده در این بخش را به روش STL بازنویسی می کنیم. نکته ای که می بایست به آن دقت شود، عدم تبدیل برنامه های STL نوشته شده به روش های دیگر می باشد.

حل:

تایمر SD یکی از پرکاربردترین تایمرها می باشد. به همین دلیل نیز در اکثر مثالهای ارائه شده از تایمر SD استفاده شده است.

```

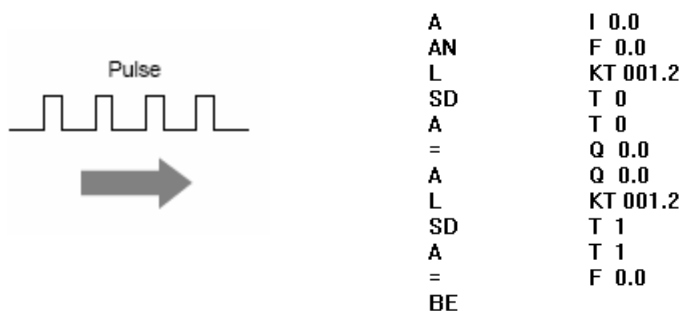
A      I 0.0
S      Q 0.0
A      Q 0.0
L      KT 015.2
SD     T 0
A      T 0
S      Q 0.1
A      I 0.1
R      Q 0.0
R      Q 0.1
BE

```

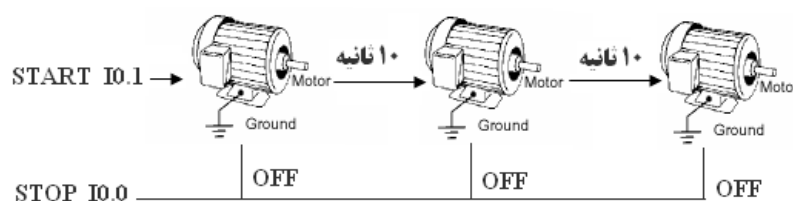
140

مثال ۲- تولید پالس در خروجی Q0.0 توسط تایمر SD

در این مثال می خواهیم برنامه ای بنویسیم، تا مادامی که ورودی IO.0 فعال می باشد، در خروجی Q0.0 یک پالس با زمان ۱ ثانیه روشن و ۱ ثانیه خاموش تولید شود.

**مثال ۳ - راه اندازی ترتیبی سه موتور**

فرض نمایید در یک پروسه صنعتی سه موتور وجود دارد. نحوه روشن شدن این موتورها بدین صورت می باشد که با فشردن کلید استارت موتور اول روشن شده، ۱۰ ثانیه بعد موتور دوم و به همین ترتیب ۱۰ ثانیه بعد موتور سوم وارد مدار می شود. می خواهیم برنامه کنترلی این مثال را به روش STL طراحی نماییم.



برنامه کنترلی

A	I 0.1
S	Q 0.0
A	Q 0.0
L	KT 010.2
SD	T 0
A	T 0
S	Q 0.1
A	Q 0.1
L	KT 010.2
SD	T 1
A	T 1
S	Q 0.2
A	I 0.0
R	Q 0.0
R	Q 0.1
R	Q 0.2
BE	

مثال ۴

در این مثال می خواهیم برنامه ای طراحی کنیم که میانگین مقادیر موجود در ۴ کلمه از حافظه محاسبه شود. مقادیر در ۴ حافظه زیر قرار می گیرند.

مقدار اول: FW0

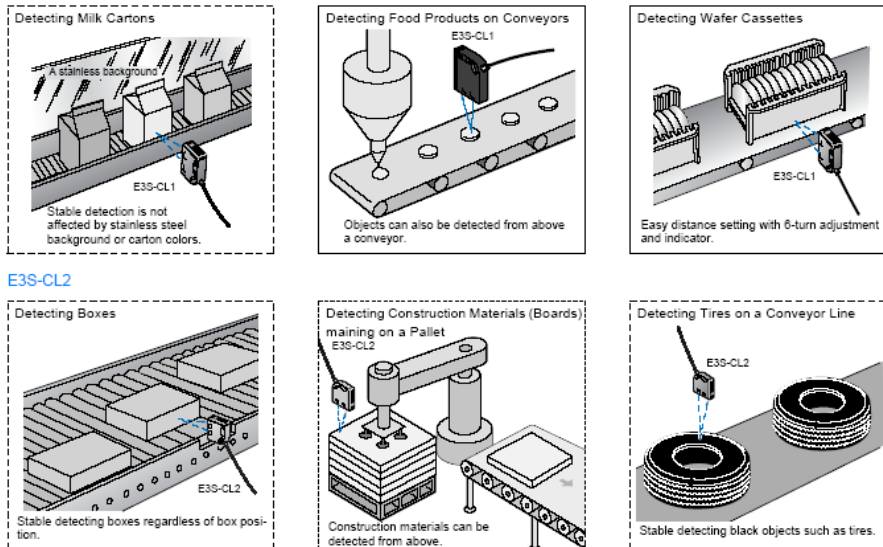
مقدار دوم: FW2

مقدار سوم: FW4

مقدار چهارم: FW6

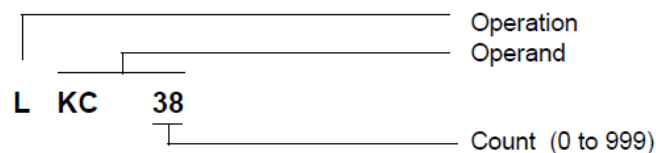
شمارنده ها

یکی دیگر از دستوراتی که در کنترل فرآیندهای صنعتی کاربرد فراوانی دارد، شمارنده‌ها می‌باشند. در بسیاری از پروسه‌ها و خطوط تولید نیاز به شمارش تعداد قطعات و یا سایر موارد بسیار زیاد دیده می‌شود.

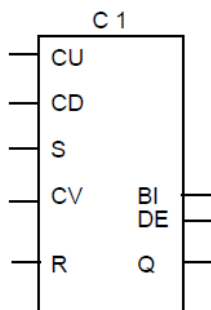


142

شمارنده‌ها دارای یک ناحیه معین در حافظه CPU می‌باشند که این ناحیه از حافظه برای هر شمارنده یک کلمه ۱۶ بیتی در نظر می‌گیرد. برای استفاده از هر شمارنده باید شماره آن را ذکر نمود. مثلاً C38 در برنامه نویسی یعنی شمارنده شماره ۳۸ فراخوانی شود.



در برنامه نویسی S5، دستور مربوط به شمارنده بصورت زیر نمایش داده می‌شود.



- ✓ ورودی CU: با اعمال پالس به این ورودی عدد شمارنده افزایش می‌یابد.
- ✓ ورودی CD: با اعمال پالس به این ورودی عدد شمارنده کاهش می‌یابد.
- ✓ ورودی S: با فعال نمودن این ورودی مقدار اولیه یا همان مقدار CV به شمارنده بار می‌گردد.
- ✓ ورودی CV: مقدار اولیه یا همان مقدار مورد نظر برای شروع شمارش می‌باشد.
- ✓ ورودی R: با فعال شدن این ورودی مقدار شمارنده صفر و خروجی بیتی شمارنده نیز غیرفعال می‌گردد.

خروجی های شمارنده :

- ✓ خروجی Q: این خروجی یک خروجی بیتی بوده و زمانی که مقدار شمارنده برابر یک شود فعال می شود. در واقع با آمدن اولین پالس به شمارنده این خروجی فعال می گردد.
- ✓ خروجی BI: توسط این خروجی می توان مقدار جاری شمارنده را به صورت باینری دریافت نمود. بازه این خروجی عددی یک Word می باشد.
- ✓ خروجی DE: توسط این خروجی می توان مقدار جاری شمارنده را با فرمت دسیمال دریافت نمود.

نحوه کار شمارنده :

در این شمارنده دو ورودی با نام های CU و CD وجود دارد که به ترتیب برای شمارش صعودی و نزولی استفاده می شوند. در این شمارنده با هر لبه بالارونده در ورودی CU یک واحد بر مقدار شمارنده افزوده، و با هر لبه بالارونده در ورودی CD از مقدار شمارنده یک واحد کاسته می شود. در هر لحظه با فعال شدن ورودی S نیز عدد شمارنده برابر مقدار اولیه یا همان مقدار CV می شود. با استفاده از ورودی R می توان شمارنده را بازنشانی نمود.

همانطور که در قسمت قبلی بیان شد، از ورودی CV جهت تنظیم مقدار اولیه برای شمارش استفاده می شود. جهت دادن مقدار اولیه به شمارنده در S5 از فرمت KC استفاده می شود. فرمت KC بصورت زیر تعریف می شود.

KC mmm

mmm عدد بار شده می باشد که می تواند از صفر تا ۹۹۹ را به خود بگیرد.

نکته ۱

لازمه اینکه شمارنده از عددی غیر از صفر شروع به شمارش کند این است که ورودی S شمارنده برای یک لبه فعال شود. به عبارت دیگر با فعال شدن ورودی S، عدد تنظیم شده در CV شمارنده به شمارنده منتقل می گردد. به عنوان مثال اگر ورودی CV شمارنده را عدد KC 50 قرار و ورودی S فعال نشود، مبنای شمارش همان عدد صفر می باشد.

نکته ۲

با فعال شدن ورودی R شمارنده، صرفنظر از هر وضعیتی، مقدار شمارنده برابر صفر می شود. در حقیقت از این ورودی جهت خاموش یا صفر نمودن شمارنده استفاده می شود. لازم است این نکته را نیز در همین قسمت یادآور شد که تمامی ورودیهای شمارنده ها به لبه بالارونده ورودی حساس می باشند.

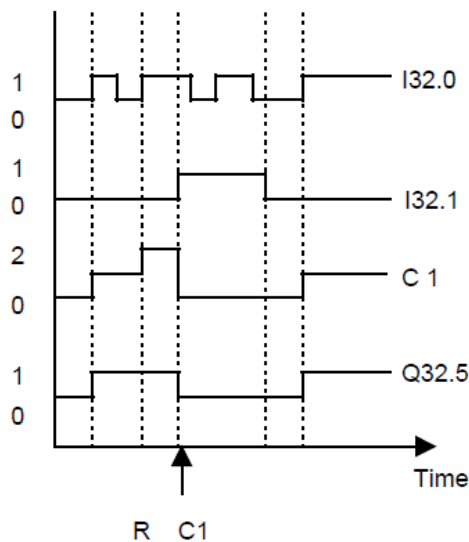
نکته ۳

خروجی بیتی شمارنده زمانی فعال می شود که مقدار شمارنده برابر عددی غیر از صفر باشد. (شمارنده بالا شمار)

اکنون با بررسی چند مثال با شمارنده ها بیشتر آشنا می شویم.

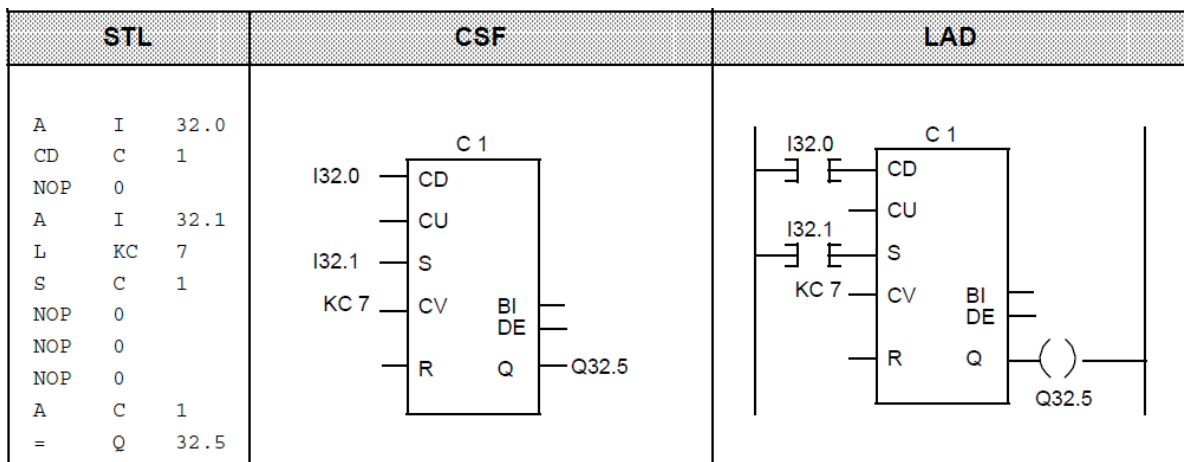
STL			CSF		LAD	
A	I	32.0				
CU	C	1				
NOP	0					
NOP	0					
NOP	0					
A	I	32.1				
R	C	1				
NOP	0					
NOP	0					
A	C	1				
=	Q	32.5				

در برنامه فوق، با هر پالس در ورودی I32.0، مقدار شمارنده یک واحد افزایش می یابد. با فعال شدن ورودی I32.1 نیز مقدار شمارنده ریست می شود. خروجی Q32.5 نیز به محض دریافت اولین پالس، یک می شود.

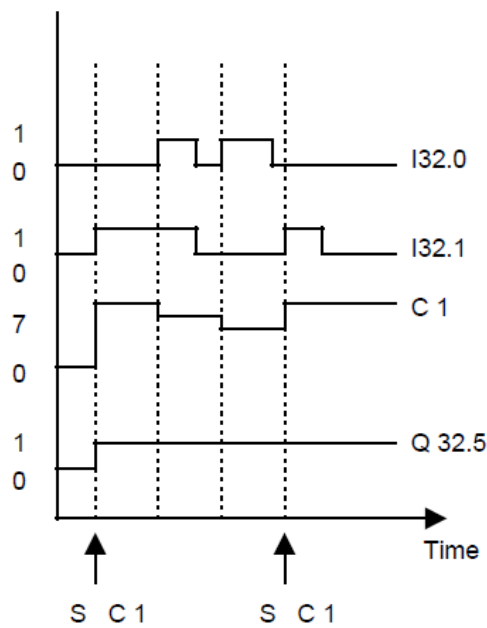


نکته

زمانی که در روش CSF یا LAD از ورودی یا خروجی استفاده نمی شود، جای خالی آن در روش STL عبارت NOP 0 می باشد.



در این مثال با فعال شدن ورودی I32.1، مقدار اولیه شمارنده برابر 7 می شود. در ادامه با هر پالس در ورودی I32.0، مقدار شمارنده یک واحد کاهش می یابد.

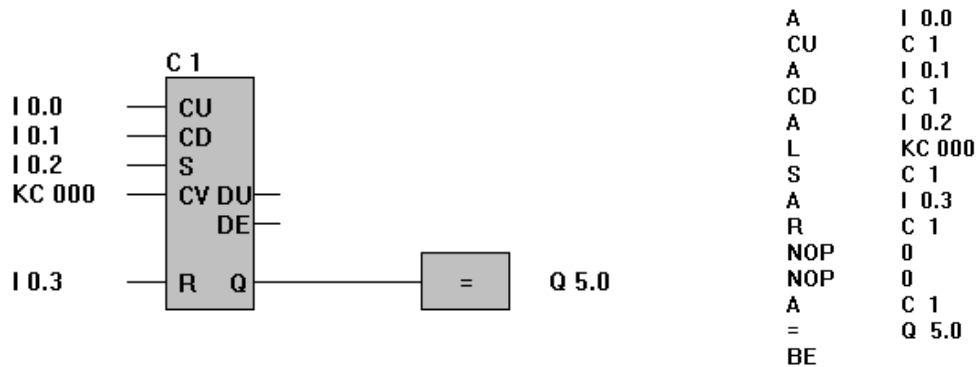


مثال

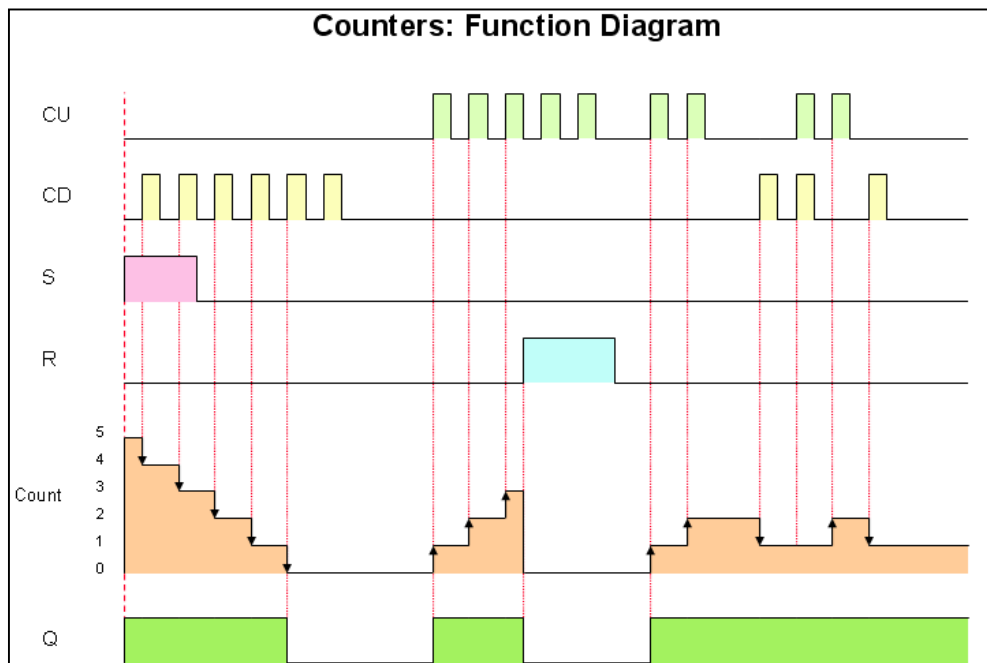
یک انبار موقت را در نظر بگیرید. یک سنسور در ورودی و یک در درب خروجی قرار دارد. می خواهیم تعداد بسته های داخل انبار هر لحظه بر روی صفحه نمایش نشان داده شود.

حل:

برای حل این مثال باید سنسور ورودی را به ورودی بالا شمار شمارنده و سنسور خروجی را به ورودی پایین شمار شمارنده متصل کنیم. مثلاً فرض نمایید که ۵۰ بسته وارد انبار شده اند، پس شمارنده عدد ۵۰ را نشان می دهد. با خروج بسته ها از انبار و ارسال پالس توسط سنسور خروجی، مقدار شمارنده شروع به کاهش می کند. در نتیجه صفحه نمایش در هر لحظه تعداد بسته های موجود در داخل انبار را نشان می دهد.



در شکل زیر نحوه عملکرد شمارنده ها به صورت کاملاً واضح نشان داده شده است.



مقایسه کننده ها (Comparators)

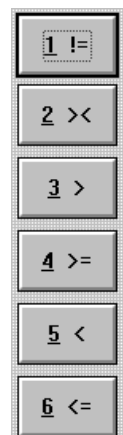
مقایسه کننده ها یکی از دستورات تکمیلی و پر کاربرد در برنامه نویسی می باشند. یک مقایسه کننده مقدار دو ورودی را دریافت و با توجه به نوع و نتیجه مقایسه، خروجی خود را فعال و یا غیر فعال می کند. اطلاعاتی که در ورودی مقایسه کننده قرار می گیرد می تواند یکی از مقادیر زیر باشد:

ورودی مقایسه کننده ها می تواند به یکی از حالت های زیر باشد. (در محدوده بیت نمی باشد)

- ۱-بایت IB0 QB0 FY0
- ۲-دو بیتی IW0 QW0 FW0
- ۳-چهار بیتی ID0 QD0 FD0
- ۴-اعداد ثابت KF
- ۵-مقدار شمارنده ها C1
- ۶-مقدار تایمرها T1

Table 8-6. Overview of Comparison Operations

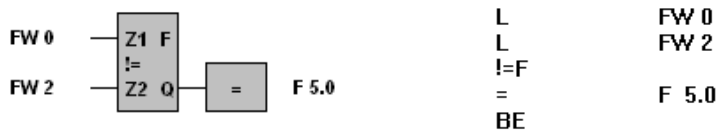
Operation	Operand	Meaning
! = F		Compare for "equal to" The contents of the two accumulators are interpreted as bit patterns and scanned to see if they are equal.
> < F		Compare for "not equal to" The contents of the two accumulators are interpreted as bit patterns and compared to see if they are not equal.
> F		Compare for "greater than" The contents of the two accumulators are interpreted as fixed-point numbers. They are compared to see if the operand in ACCU 2 is greater than the operand in ACCU 1.
> = F		Compare for "greater than or equal to" The contents of the two accumulators are interpreted as fixed-point numbers. They are compared to see if the operand in ACCU 2 is greater than or equal to the operand in ACCU 1.
< F		Compare for "less than" The contents of the two accumulators are interpreted as fixed-point numbers. They are compared to see if the operand in ACCU 2 is less than the operand in ACCU 1.
< = F		Compare for "less than or equal to" The contents of the two accumulators are interpreted as fixed-point numbers. They are compared to see if the operand in ACCU 2 is less than or equal to the operand in ACCU 1.



! = F	=	مساوی بودن دو عدد
> < F	≠	نامساوی بودن دو عدد
> F	>	عدد اول بزرگتر از عدد دوم
> = F	≥	عدد اول بزرگتر یا مساوی با عدد دوم
< F	<	عدد اول کوچکتر از عدد دوم
< = F	≤	عدد اول کوچکتر یا مساوی عدد دوم

مثال ۱

در این برنامه مقدار دو حافظه با یکدیگر مقایسه شده و در صورت برابری F5.0 فعال می شود.



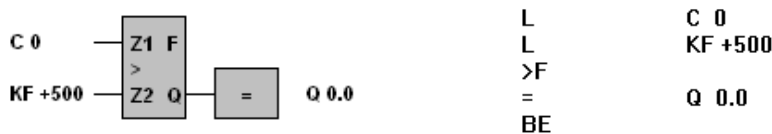
مثال ۲

در این برنامه مقدار موجود در FW0 با عدد ثابت ۱۵۰۰ مقایسه و در صورتی که مقدار FW0 از ۱۵۰۰ بیشتر باشد، Q0.0 فعال می شود.



مثال ۳

در این برنامه مقدار C0 با مقدار ثابت 500 مقایسه و در صورتی که مقدار شمارنده بیشتر از 500 شود، Q0.0 فعال می شود.



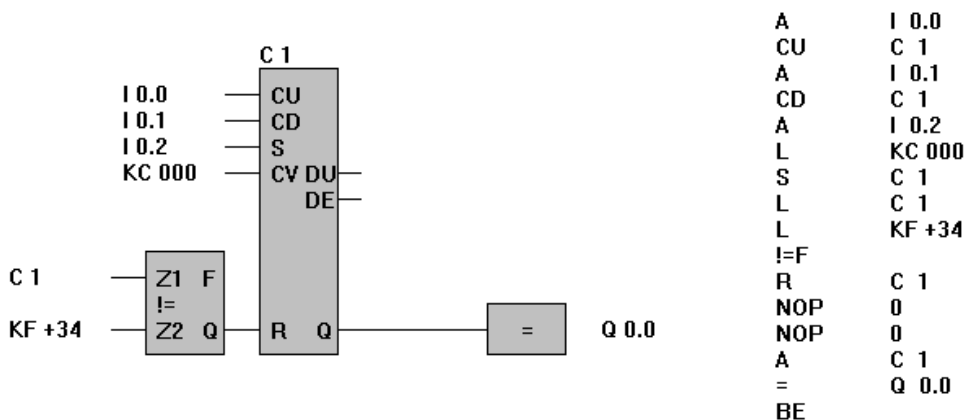
مثال ۴

در این مثال مقدار دو بابت ورودی با هم مقایسه می شود. در صورتی که مقدار این دو بابت برابر باشند، خروجی Q32.5 روشن می شود.

Circuit Diagram	STL	CSF/LAD
	<pre> L IB 19 L IB 20 ! = F = Q 32.5 </pre>	

مثال ۵

در این مثال می خواهیم برنامه شمارنده ای را طراحی نماییم که از ۰ تا ۳۴ شمرده، مجدداً صفر گردد.



149

مقدار شمارنده با عدد ۳۴ مقایسه و در صورت برابری، شمارنده ری ست می شود و مجدداً از صفر راه اندازی می گردد.

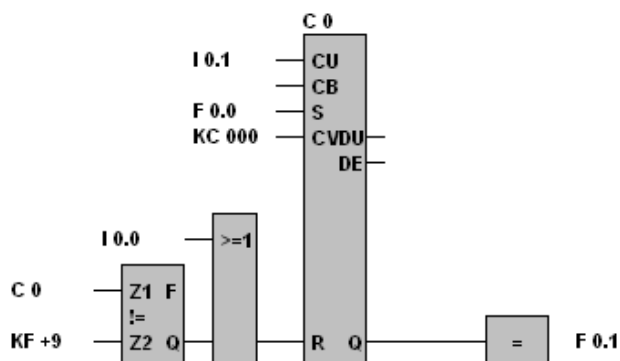
مثال ۶

در این مثال قصد داریم برنامه ای طراحی کنیم که توسط یک شاستی بتوان ۴ موتور را به صورت زیر کنترل نمود.

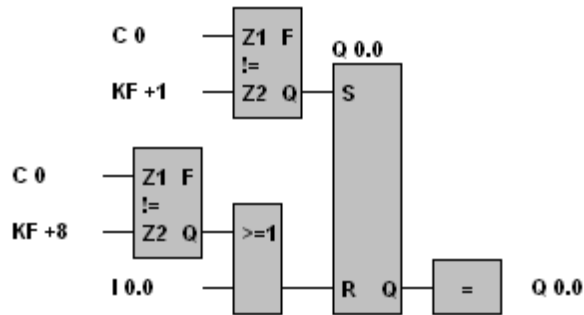
با هر بار فعال شدن شاستی یک موتور وارد مدار شود. در این صورت پس از ۴ بار فشردن شاستی IO.1 هر ۴ موتور روشن شوند. اگر شاستی برای بار پنجم فشار داده شد، به همان ترتیب که موتورهای روشن شده بودند خاموش شوند. به عبارت دیگر با هر دفعه فشردن شاستی از مرحله ۵، موتورهای خاموش گردند. یعنی در مرحله ۵ موتور ۴ خاموش و در مرحله ۶ موتور ۳ خاموش و این روند تا موتور ۱ ادامه پیدا کند. در ضمن برنامه باید به گونه ای طراحی شود که کل این سیکل مجدداً قابل تکرار باشد.

برنامه

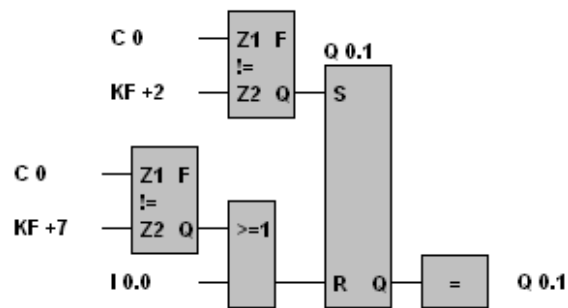
Segment 1



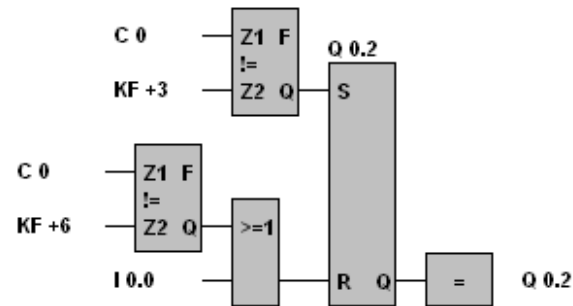
Segment 2



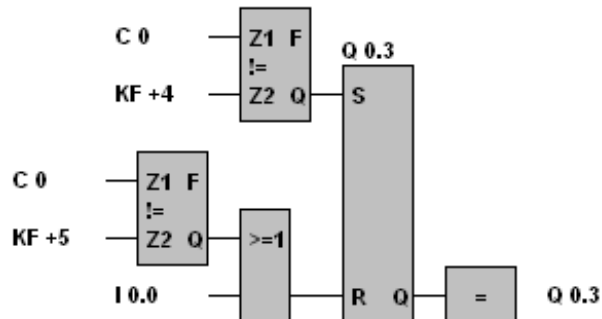
Segment 3



Segment 4



Segment 5



در این مثال نیز از ورودی IO.0 به عنوان استپ کل استفاده شده است.

بازنویسی مثال قبل به روش STL

```

A          I 0.1
CU         C 0
L          C 0
L          KF +1
! = F
S          Q 0.0
L          C 0
L          KF +2
! = F
S          Q 0.1
L          C 0
L          KF +3
! = F
S          Q 0.2
L          C 0
L          KF +4
! = F
S          Q 0.3
L          C 0
L          KF +5
! = F
R          Q 0.3
L          C 0
L          KF +6
! = F
R          Q 0.2
L          C 0
L          KF +7
! = F
R          Q 0.1
L          C 0
L          KF +8
! = F
R          Q 0.0
R          C 0
A          I 0.0
R          Q 0.0
R          Q 0.1
R          Q 0.2
R          Q 0.3
R          C 0

```

تمرین

در یک خط تولیدی فرض کنید که توسط یک شاستی استارت موتور اصلی مربوط به کل خط روشن می شود. در این پروسه چندین فیلتر وجود دارد که می بایست در زمان های مختلف تمیز و یا تعویض شوند. برای این منظور لازم است برنامه ای طراحی شود که پس از این که موتور اصلی راه اندازی شد، در فواصل زمانی مختلف آلامر هایی جهت چک نمودن فیلترها فعال شود.

آلامر ۱: پس از گذشت ۱۰۰۰ ساعت

آلامر ۲: پس از گذشت ۳۰۰۰ ساعت

152



روشن شدن لامپ سیگنال Q0.1 به عنوان آلامر ۱ و Q0.2 نیز به عنوان آلامر ۲ در نظر گرفته شده است. البته امروزه این آلامر ها در اتاق های کنترل بر روی PC های مرکزی به صورت یک عبارت نمایش داده می شود.

I0.0: استارت

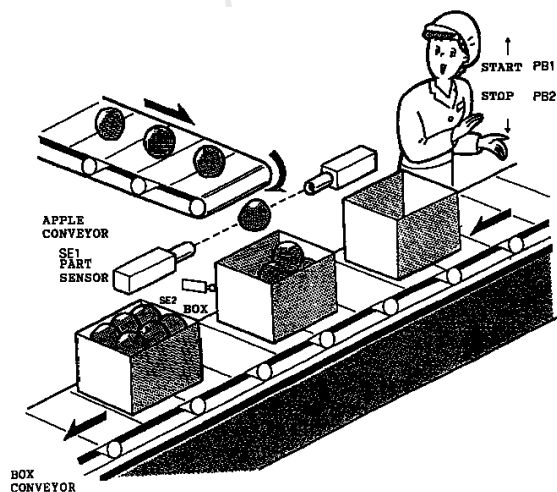
I0.1: استپ

Q0.0: موتور اصلی خط

در حل این تمرین فرض کنید که موتور اصلی مدام در حال کار می باشد و تا زمانی که فرمان استپ داده نشود متوقف نمی گردد.

تمرین ۱- کنترل دو کانوایر

در پروسه نشان داده شده در شکل زیر می خواهیم برنامه ای طراحی شود به طوریکه با فشردن شاستی استارت PB1، موتور کانوایر مربوط به جعبه ها به حرکت درآید. با رسیدن کارتون در موقعیت مناسب و تشخیص آن توسط سنسور SE2 موتور کانوایر در حال کار خاموش شده و موتور مربوط به کانوایر سیب ها روشن می شود. در ادامه تعداد ۱۰ سیب در کارتن قرار می گیرد. بعد از پر شدن کارتن مجدداً کانوایر مربوط به سیب ها توقف و کانوایر جعبه ها حرکت می کند. این روند برای سایر کارتن ها تا زمانی که فرمان استپ توسط شاستی PB2 صادر نشده باشد ادامه می یابد. در ضمن تعداد سیب ها نیز توسط سنسور دوطرفه SE1 شمرده می شود.



مشخصات:

- IO.0 : استارت
- IO.1 : استپ
- IO.2 : سنسور SE1
- IO.3 : سنسور SE2
- Q0.0 : کانوایر مربوط به جعبه ها
- Q0.1 : کانوایر مربوط به سیب ها

تمرین

در این مثال می خواهیم برنامه ای طراحی شود به گونه ای که در مثال قبل مدت زمان توقف و کار خط تولید ثبت و مشخص گردد. به عبارت دیگر زمانی که موتور اصلی خط (Q0.0) در حال کار می باشد زمان سنج ۱ همانند یک ساعت شروع به کار کند و زمانی که موتور متوقف می شود زمان سنج ۲ وارد مدار شود و زمان سنج ۱ آخرین مقدار خود را حفظ کند.

زمان سنج ۲

- C5 : ثانیه
- C6 : دقیقه
- C7 : ساعت
- C8 : روز

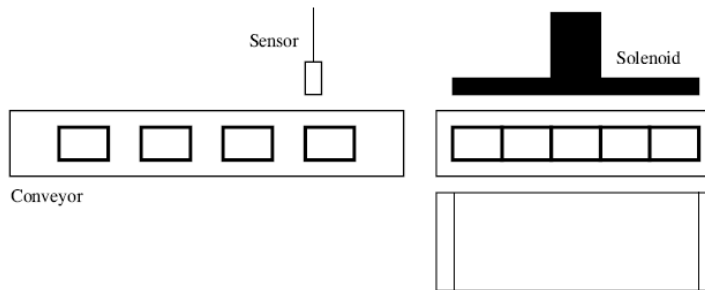
زمان سنج ۱

- C1 : ثانیه
- C2 : دقیقه
- C3 : ساعت
- C4 : روز

نکته: برنامه طراحی شده می بایست به گونه ای باشد که مقدار زمان سپری شده برای هر دو زمان سنج در هر لحظه قابل مشاهده باشد. در این سیستم ریست زمان سنج ها در پایان هر ماه توسط شاستی IO.1 به صورت دستی صورت می گیرد.

تمرین - کنترل ماشین بسته بندی

در یک پروسه صنعتی جهت بسته بندی محصولات تولید شده در انتهای خط از یک جک جهت بسته بندی محصولات مطابق شکل زیر استفاده شده است.



نحوه کار بدین صورت می باشد که با فرمان استارت اولیه، موتور مربوط به کانوایر جهت حمل قطعات روشن می شود. این کانوایر تا زمانی که فرمان استپ صادر نشود در مدار می باشد.

در ادامه بعد از اینکه ۵ قطعه از زیر سنسور عبور و بر روی میز تعبیه شده قرار گرفتند، فرمان تحریک شیر برقی صادر می شود. با تحریک شیر برقی جک به جلو آماده و قطعات را به مسیر بعدی منتقل می کند. فرمان وصل شیر ۲ ثانیه می باشد. در واقع ۲ ثانیه بعد که شیر قطع می شود جک نیز به موقعیت اولیه خود باز می گردد و برای سیکل بعدی آماده می شود.

مشخصات:

I0.0: استارت

I0.1: استپ

I0.2: سنسور

Q0.0: کانوایر

Q0.1: شیر

دستورات پرش (Jump)

دستورات روش STL بسیار زیاد بوده و در این کتاب از پرداختن به تمامی دستورات صرفنظر شده است. در این بخش تنها دستورات پر کاربرد در روش STL مورد بحث قرار گرفته است. این دستور در PLC S5 تنها به روش STL موجود می باشد، ولی در PLC S7، معادل این دستور در سایر روش های برنامه نویسی نیز موجود می باشد.

توسط دستورات پرش امکان رفتن از یک خط برنامه به خط مورد نظر و همچنین پرش به سایر بلوک های برنامه وجود دارد. مقصد پرش حتما می بایست توسط یک برچسب مشخص شده باشد. همین برچسب در مبدا نیز تکرار می شود.

155

انواع دستورات پرش

JC = پرش در صورتی که نتیجه بیت RLO ما قبل این دستور ۱ باشد. (پرش شرطی)
JU = پرش غیر شرطی (نتیجه بیت RLO مهم نمی باشد)

دستورات پایان برنامه

توسط این دستورات امکان به پایان رساندن برنامه در بین خطوط و یا در پایان برنامه وجود دارد. دستور BE تنها در پایان برنامه قرار می گیرد.

BE = پایان برنامه در انتهای برنامه

BEU = پایان برنامه غیر شرطی در بین خطوط

BEC = پایان برنامه شرطی در بین خطوط

مثال ۱

```

;
A      I 0.0
JC     =M000
O      I 0.1
O      I 0.2
=      Q 0.0
BEU
M000: A      I 0.1
      A      I 0.2
      =      Q 0.0
      BE

```

در این مثال با فعال بودن ورودی IO.0، ورودی های IO.1 و IO.2 با یکدیگر موازی و در غیر این صورت سری می شوند. زمانی که ورودی IO.0 غیرفعال باشد، پرش انجام نمی شود و برنامه منطق OR را اجرا می کند و با رسیدن به BEU، به پایان می رسد. با فعال بودن IO.0 نیز، روال برنامه به برجسب M000 پرش کرده و منطق AND را اجرا می کند.

مثال ۲

```

M000:      A      I 0.0
           JC      =M000
           BEU
           L      KM 00000000 11111111
           T      QB 0

           BE

```

در این مثال با فعال شدن IO.0، پرش به برجسب M000 صورت می گیرد و یک بایت از خروجی PLC روشن می شود. در واقع این برنامه را می توان معادل دستور MOVE در PLC S7 در نظر گرفت که به صورت شرطی، یک مقدار را انتقال می دهد.

مثال ۳

```

M000:      A      I 0.0
           JC      =M000
           L      KH 0000
           T      QB 0
           BEU
           L      KH 00FF
           T      QB 0

           BE

```

در این مثال با فعال شدن ورودی IO.0، یک بابت خروجی روشن و با صفر شدن ورودی، خروجی ها نیز خاموش می شوند. از این برنامه می توان به عنوان برنامه تست خروجی ها استفاده نمود. برای آشنایی هر چه بیشتر با انواع فرمت ها، در مثال های مختلف سعی شده است از فرمت های مختلف استفاده شود.

مثال ۴

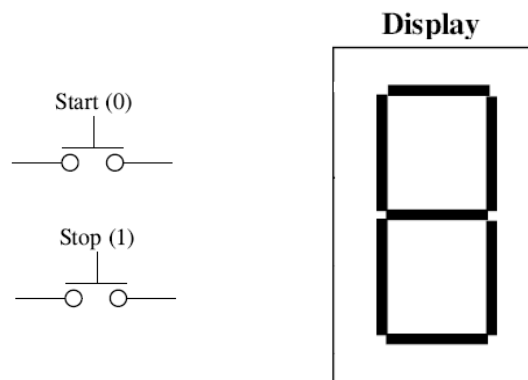
	A	I 0.0
	JC	=M000
	L	KS E
	T	FY 0
	BEU	
M000:	L	KS N
	T	FY 0
	BE	

در این مثال با فعال شدن ورودی I0.0، کاراکتر N در یک بایت حافظه فلگ منتقل می شود. زمانی که ورودی صفر شود، کاراکتر E جایگزین می شود.

157

تمرین

مطلوب است طراحی برنامه به روش STL مربوط به یک سون سگمنت که با فشردن شاستی استارت از عدد 0 شروع به شمارش کرده و هر یک ثانیه یک واحد به آن افزوده شود. با رسیدن به عدد 9 مجدداً صفر شده و این روند تا فشردن شاستی استپ تکرار شود. در ضمن جهت کنترل سون سگمنت از یک بایت خروجی PLC استفاده شده است.



مشخصات:

I0.0: استارت

I0.1: استپ

نکته: جهت نمایش عددهای 0 تا 9 بر روی سون سگمنت در ابتدا لازم است که کدهای باینری و یا HEX مربوط به هر عدد مشخص شود و سپس هر ۱ ثانیه این کدها توسط دستورات پرش، L و T، به QB0 منتقل شود. در ضمن بیت آخر QB0 که مربوط به نمایشگر h می باشد را همیشه 0 فرض کنید.

آشنایی با روش های برنامه نویسی

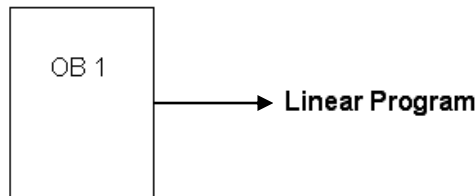
در اکثر نرم افزارهای برنامه نویسی معمولاً برنامه نویس می تواند برنامه خود را با توجه به حجم و نوع پروسه کنترلی به روش های مختلفی طراحی نماید. منظور از روش های مختلف برنامه نویسی روش های FBD، STL و LAD نمی باشد. منظور از روش های برنامه نویسی در این قسمت ساختاری است که برنامه نویس برای برنامه خود در نظر می گیرد. معمولاً در برنامه کنترلی می توان از ۳ ساختار استفاده نمود. البته لازم به ذکر است که انتخاب ساختار مناسب برای برنامه، بستگی به حجم و نوع برنامه دارد.

- Linear program
- Divided program
- Structured Program

Linear program

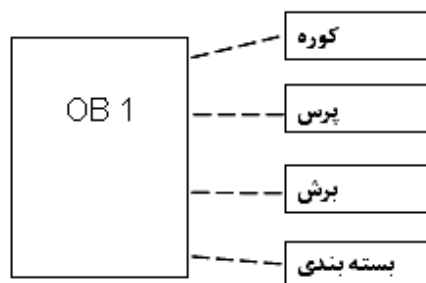
این روش برنامه نویسی تحت عنوان برنامه نویسی خطی شناخته می شود. در این روش تمامی برنامه مربوط به یک پروسه در یک بلوک و به صورت متوالی نوشته می شود. منظور از بلوک همان بلوک OB1 در نرم افزار می باشد. در ابتدا لازم است که ویژگی و نحوه اجرای بلوک OB1 مورد بررسی قرار گیرد. در اکثر PLC های شرکت زیمنس بلوکهایی با نام های مختلف جهت کاربردهای مختلف تعبیه شده است. اما وجه اشتراک بین تمامی PLC های زیمنس استفاده از بلوک OB1 به عنوان بلوک اصلی در برنامه می باشد. زمانی که CPU به حالت RUN سوئیچ می شود، CPU در ابتدا در صورت وجود OB های راه اندازی به آنها مراجعه و سپس نیاز به بلوک OB1 دارد و در واقع سیستم عامل بلوک OB1 را فراخوانی می کند. البته اجرای OB1 را نیز می توان توسط سایر OB ها در هر لحظه قطع نمود. پس در نتیجه بلوک OB1 به عنوان ساختار اصلی برنامه کاربر بوده که اکثر بلوک های برنامه جهت اجرا حتماً می بایست در این بلوک فراخوانی شوند. در برنامه نویسی خطی برنامه نویس کل برنامه خود را در بلوک OB1 در Network های متوالی وارد می کند. در پروژه های صنعتی که حجم برنامه کنترلی زیاد می باشد، نوشتن تمامی برنامه در بلوک OB1 کار صحیح و منطقی نمی باشد. به دلیل اینکه تمامی برنامه می بایست پشت سرهم و در یک بلوک طراحی شود. یکی از معایب برنامه نویسی خطی عیب یابی مشکل در برنامه نوشته شده می باشد. همچنین در بسیاری از مواقع نیاز می باشد که قسمتی از برنامه توسط CPU پردازش نشود. در برنامه نویسی خطی به دلیل اینکه تمامی برنامه در یک بلوک می باشد، برنامه نویس معمولاً کمتر می تواند بر روی بخش های مختلف برنامه کنترل لازم را داشته باشد. از طرفی برخی از ماشین آلات صنعتی دارای چند مد کاری می باشند. به عنوان مثال فرض کنید که یک دستگاه دارای ۲ مد دستی و اتوماتیک می باشد.

برنامه این دو مد کاری با یکدیگر متفاوت می باشند. از طرفی هر دو مد مربوط به یک ماشین بوده و اگر از برنامه نویسی خطی استفاده گردد امکان تداخل آدرس و المانهای موجود در برنامه با یکدیگر وجود دارد. پس در نتیجه استفاده از این روش برنامه نویسی در پروژه هایی که حجم برنامه زیاد می باشد توصیه نمی شود.



Divided program

در این روش برنامه نویسی از بلوک های دیگری جهت نوشتن برنامه استفاده می شود. در واقع در این روش قسمت های مختلف مربوط به یک پروژه در بلوک های مجزا از یکدیگر نوشته می شوند. این بلوک ها به عنوان زیربرنامه مورد استفاده قرار می گیرند. به عنوان مثال یک خط تولید را در نظر بگیرید که شامل بخش های مختلفی از جمله کوره، پرس، برش و بسته بندی می باشد. در این حالت با استفاده از بلوک های زیربرنامه می توان برنامه مربوط به هر بخش را از بخش دیگر جدا نمود.

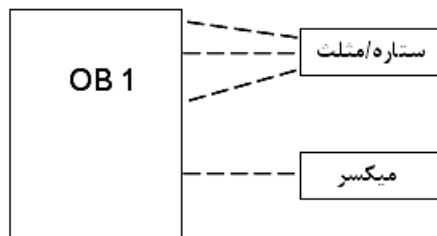


این نوع برنامه نویسی که کاربرد فراوانی در پروژه های بزرگ دارد، تحت عنوان برنامه نویسی تقسیم بندی شده شناخته می شود و یکی از مزایای این روش عیب یابی آسان و تحت کنترل درآوردن هر چه بیشتر اجزای مختلف برنامه می باشد. در PLC مختلف شرکت زیمنس نام های مختلفی برای این بلوک ها در نظر گرفته شده است که در ادامه در نرم افزار مورد بحث با این بلوک ها آشنا می شویم.

Structured Program

روش دیگری که در پروژه های بزرگ مورد استفاده قرار می گیرد روش برنامه نویسی ساختار یافته می باشد. از این روش در پروژه هایی که یک منطق بین چندین بخش از پروژه مشابه می باشد استفاده می شود. در واقع استفاده از این روش زمان برنامه نویسی را بسیار کاهش می دهد. به عنوان مثال فرض کنید که قرار است ۱۰ موتور توسط یک PLC کنترل شوند. راه اندازی تمامی موتورها به صورت ستاره/ مثلث دستی می باشد. تنها نقطه تفاوت در این پروژه آدرس های I/O مربوط به هر موتور می باشد. در این گونه موارد نیازی به تکرار برنامه برای ۱۰ موتور به صورت مجزا نمی باشد. در این صورت برنامه نویس می تواند یک تابع با منطق ستاره/ مثلث طراحی و این تابع را برای تمامی موتورها استفاده کند. در نتیجه استفاده از روش ساختار یافته در بخش هایی از برنامه که قرار است تکرار شود بسیار مفید می باشد.

160



همانطور که در شکل فوق ملاحظه می کنید تابع ستاره/ مثلث چندین بار در بلوک اصلی برنامه برای موتورهای مختلف فراخوانی شده است.

نکته: در یک تابع، آدرس های I/O در زمان فراخوانی اعمال می شود.

آشنایی با بلوک های PB

همانطوریکه بیان شد، بلوک OB1 به عنوان بلوک اصلی برنامه می باشد که بصورت سیکلی توسط سیستم عامل پردازش می شود. اما در اکثر PLC های زیمنس بلوک هایی نیز به عنوان زیربرنامه می توانند در برنامه مورد استفاده قرار گیرند. کسانی که با PLC های سری S7 آشنایی دارند به خوبی می دانند که در PLC های سری S7، علاوه بر بلوک OB1 امکان نوشتن برنامه در بلوک های FC و FB نیز وجود دارد. در PLC S5 نیز بلوک هایی با نام PB و FB جهت نوشتن برنامه کنترلی تعبیه شده است.

مثال ۱

از بلوک های PB می توان بصورت جداگانه به عنوان زیربرنامه های OB1 استفاده نمود. در این مثال هدف نوشتن دو برنامه در دو PB بصورت مجزا و فراخوانی آنها در OB1 می باشد. (برنامه نویسی تقسیم شده)

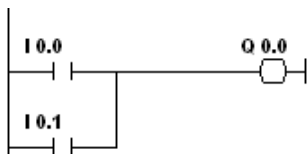
فرض کنید می خواهیم یک موتور را به دو صورت کنترل کنیم. جهت راه اندازی این موتور از دو کلید استارت استفاده می شود. به غیر از این دو کلید از یک کلید سلکتوری نیز برای تعیین منطق بین استارت ها استفاده شده است. در واقع این کلیدها می توانند هم بصورت سری و هم بصورت موازی فرمان وصل موتور را صادر کنند. سری یا موازی بودن کلیدها توسط کلید سلکتوری مشخص می گردد. این کلید دارای دو حالت A و B می باشد.

اگر کلید در وضعیت A قرار داشت، باید کلیدهای استارت سری و اگر کلید در وضعیت B بود همان دو کلید استارت بصورت موازی وارد مدار شوند. حالت A به ورودی I 2.0 و حالت B به ورودی I 2.1 متصل می باشند.

برنامه 1 PB (حالت سری)



برنامه 2 PB (حالت موازی)



فراخوانی در OB1

A	I 2.0
JC	PB 1
A	I 2.1
JC	PB 2
BE	

همانطور که در برنامه OB 1 ملاحظه می کنید، از دو کلید I 2.0 و I 2.1 جهت شروط اجرا شدن PB ها استفاده شده است. در این برنامه به عنوان مثال برای اجرا شدن PB1 و راه اندازی موتور به صورت سری ، حتما باید ورودی I 2.0 فعال باشد. با غیر فعال شدن I 2.0 مرور برنامه توسط CPU در PB1 قطع می گردد. با توجه به شرایط کاری این مثال هیچ گاه نباید شرایطی ایجاد شود تا هر دو PB خوانده شوند. برای این منظور از کلید سلکتوری جهت انتخاب شرط استفاده شده است.

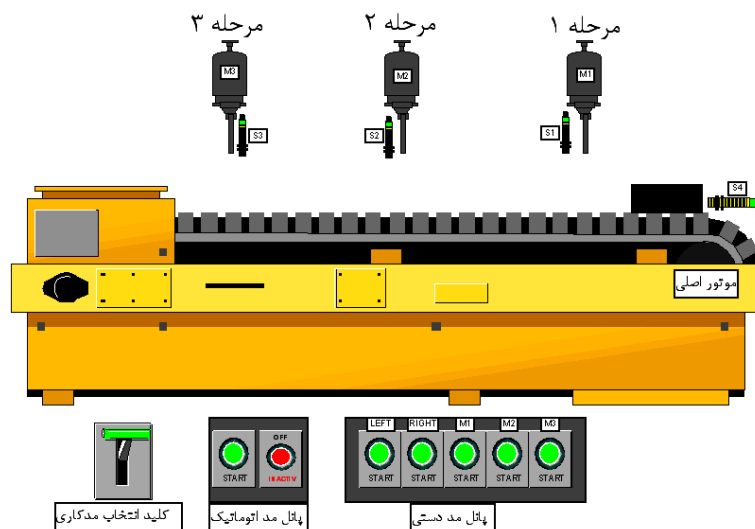
دانلود کلیه بلوک ها به PLC

زمانی که تعداد بلوک های برنامه نویسی در نرم افزار زیاد می گردد، دانلود تک تک بلوک ها به صورت جداگانه بر روی CPU کار غیرمنطقی می باشد. همچنین دانلود تک تک بلوک ها به CPU درصد خطا را بیشتر می کند. در این صورت می توان با انجام یکسری مراحل بسیار ساده تمامی بلوک ها را به طور یکجا به CPU دانلود نمود. این موضوع در رابطه با محیط شبیه سازی نیز صادق می باشد.

تمرین

162

در این پروژه می خواهیم برنامه مربوط به یک دستگاه که دارای دو مد دستی و اتوماتیک می باشد را طراحی کنیم. نحوه کار بدین صورت می باشد که در این ماشین صنعتی از یک موتور که به صورت چپگرد راستگرد کار می کند جهت انتقال قطعه استفاده می شود. این قطعه می بایست در مراحل مختلف جهت انجام عملیات ماشین کاری متوقف شود. زمان توقف نیز در هر مرحله متفاوت می باشد. در ضمن عملیات ماشین کاری تنها در مسیر رفت انجام می گردد.



مد اتوماتیک: در این مد اپراتور با فشردن شاستی استارت در پانل مربوط به مد اتوماتیک، موتور را در جهت چپگرد وارد مدار می کند. با روشن شدن موتور، قطعه تا سنسور S1 جهت انجام عملیات ماشین کاری در مرحله ۱ پیش می رود. با فعال شدن سنسور S1، موتور اصلی متوقف و موتور M1 وارد مدار می شود. زمان انجام عملیات ماشین کاری در این مرحله ۶۰ ثانیه می باشد. پس از سپری شدن این زمان موتور M1 خاموش و مجدداً موتور اصلی در جهت چپگرد شروع به حرکت نموده و این سیکل را در مراحل بعدی نیز تکرار می کند. پس از اتمام مرحله ۳، موتور در جهت راستگرد وارد مدار شده و قطعه را به نقطه اولیه که توسط سنسور S4 محدود می شود باز می گرداند. در ضمن در مسیر برگشت عملیات ماشین کاری انجام نمی شود. در هر لحظه با فشردن شاستی استپ نیز کل پروژه متوقف می گردد.

مد دستی: در این مد اپراتور می تواند هر کدام از تجهیزات دستگاه را به صورت لحظه ای وارد کند. در این مد زمان هیچ نقشی در روشن بودن موتورها ندارد. نکته قابل توجه در این مد حفاظت برای موتور اصلی و محدود شدن حرکت در هر دو جهت می باشد. فرمان استارت تجهیزات توسط شاستی های تعبیه شده در پانل مربوط به مد دستی انجام می شود.

تعیین مد کاری: جهت تعیین مد اتوماتیک و دستی از یک کلید دو حالت استفاده می شود. زمانی که کلید در وضعیت 0 می باشد دستگاه در مد اتوماتیک و زمانی که همین کلید در وضعیت یک قرار گرفت دستگاه به مد دستی سوئیچ می شود.

مشخصات

I0.0: شاستی استارت (مد اتوماتیک)

I0.1: شاستی استپ

I0.2: شاستی استارت چپگرد موتور (مد دستی)

I0.3: شاستی استارت راستگرد موتور (مد دستی)

I0.4: شاستی موتور M1 (مد دستی)

I0.5: شاستی موتور M2 (مد دستی)

I0.6: شاستی موتور M3 (مد دستی)

I0.7: سنسور مرحله ۱ (S1)

I1.0: سنسور مرحله ۲ (S2)

I1.1: سنسور مرحله ۳ (S3)

I1.2: سنسور نقطه ابتدایی (S4)

I1.3: کلید تعیین مد کاری

Q0.0: کنتاکتور مربوط به چپگرد

Q0.1: کنتاکتور مربوط به راستگرد

Q0.2: موتور M1

Q0.3: موتور M2

Q0.4: موتور M3

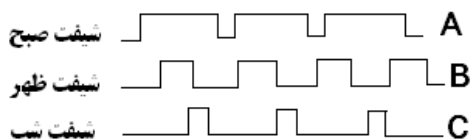
بلوک PB1: برنامه مربوط به مد اتوماتیک

بلوک PB2: برنامه مربوط به مد دستی

مثال

کنترل سه فن در ۳ مد کاری

در داخل یک پروسه صنعتی فرض نمایید از سه فن استفاده شده است. این فن ها با توجه به سه شیفت کاری دارای فرکانس متفاوتی می باشند. تعیین فرکانس کاری این فن ها توسط یک کلید سلکتوری تعیین می شود. در شکل زیر فرکانس هر شیفت کاری برای فن ها نشان داده شده است.



A : ۴۰ ثانیه ON ، ۸ ثانیه OFF

B : ۲۵ ثانیه ON ، ۲۰ ثانیه OFF

C : ۱۰ ثانیه ON ، ۳۰ ثانیه OFF

همانطور که بیان شد برای تعیین سرعت کاری فن ها از یک کلید سلکتوری استفاده شده است. سه حالت این کلید دارای آدرسهای مطابق قسمت زیر می باشند.

مد ۱: I 2.0

مد ۲: I 2.1

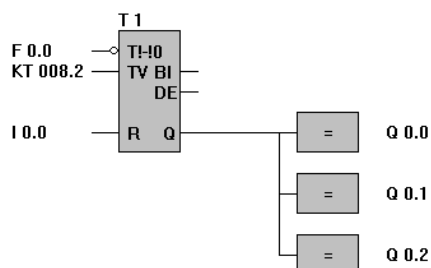
مد ۳: I 2.2

توضیح

منظور از فرکانس کاری همان سرعت قطع و وصل فن ها می باشد.

برنامه 1 (فرکانس A)

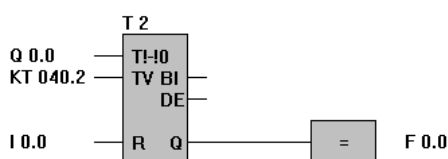
SEGMENT 1



```

AN      F 0.0
L       KT 008.2
SD      T 1
A       I 0.0
R       T 1
NOP     0
NOP     0
A       T 1
=       Q 0.0
=       Q 0.1
=       Q 0.2
    
```

SEGMENT 2

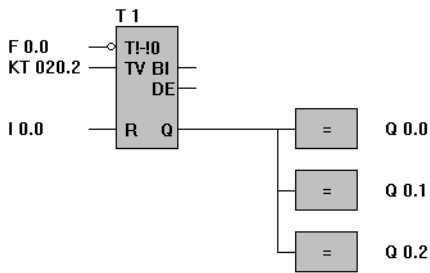


```

A       Q 0.0
L       KT 040.2
SD      T 2
A       I 0.0
R       T 2
NOP     0
NOP     0
A       T 2
=       F 0.0
BE
    
```

برنامه PB2 (فرکانس B)

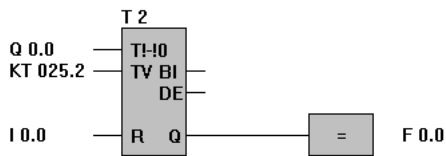
SEGMENT 1



```

AN      F 0.0
L       KT 020.2
SD      T 1
A       I 0.0
R       T 1
NOP     0
NOP     0
A       T 1
=       Q 0.0
=       Q 0.1
=       Q 0.2
    
```

SEGMENT 2

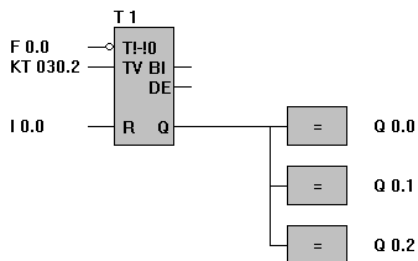


```

A       Q 0.0
L       KT 025.2
SD      T 2
A       I 0.0
R       T 2
NOP     0
NOP     0
A       T 2
=       F 0.0
BE
    
```

برنامه PB 3 (حالت C)

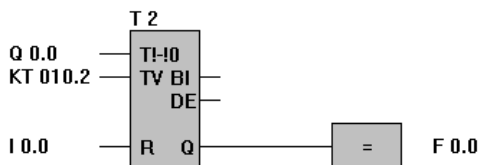
SEGMENT 1



```

AN      F 0.0
L       KT 030.2
SD      T 1
A       I 0.0
R       T 1
NOP     0
NOP     0
A       T 1
=       Q 0.0
=       Q 0.1
=       Q 0.2
    
```

SEGMENT 2



```

A       Q 0.0
L       KT 010.2
SD      T 2
A       I 0.0
R       T 2
NOP     0
NOP     0
A       T 2
=       F 0.0
BE
    
```

برنامه OB 1

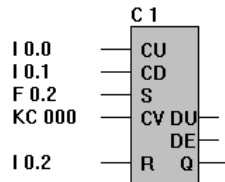
```

A       I 2.0
JC      PB 1
A       I 2.1
JC      PB 2
A       I 2.2
JC      PB 3
    
```

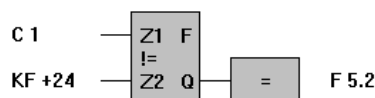
مثال

می خواهیم برنامه ای طراحی نماییم که زمانی مقدار شمارنده برابر عدد ۲۴ شد، PLC متوقف شود.

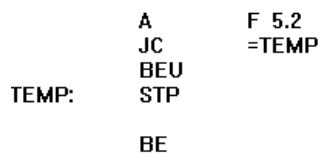
Segment 1



Segment 2



Segment 3



در سگمنت ۱ از یک شمارنده جهت شمارش تعداد پالسهای ارسالی توسط I 0.0 استفاده شده است. در سگمنت ۲ زمانی که مقدار شمارنده برابر ۲۴ شود فلگ F5.2 فعال که با فعال شدن این بیت در سگمنت ۳ عمل پرش به برجسب TEMP انجام و PLC با اجرای دستور STP متوقف می گردد.

کاربرد فرمت KY

در یک مثال ساده فرض کنید می خواهیم عدد ۱۰۰ را به FY0 و عدد ۱۵۰ را در FY1 منتقل کنیم.

این برنامه حالت عادی برنامه نویسی می باشد.

```

L    KY +100
T    FY 0
L    KY +150
T    FY 1
BE
  
```

برنامه نویس می تواند به جای نوشتن برنامه ای مطابق مثال صفحه قبل، برنامه ای خلاصه تر توسط دستور KY

```

L    KY 100,150
T    FW 0
BE
  
```

طراحی کند.

در این برنامه فرمت KY هر دو عدد را یکجا دریافت و عدد اول را در بایت ۰ حافظه فلگ و عدد دوم در بایت بایت بالایی قرار می دهد.

دستورات تکمیلی در S5

در برنامه نویسی S5 به روش STL دستورات بسیاری وجود دارد که در این قسمت به توضیح دستورات کاربردی می پردازیم. لازم به توضیح است که این دستورات در بلوکهای تابع ساز یا FB کاربرد دارد که در یحتهای بعدی در مورد برنامه نویسی در این بلوکها مطالبی ارائه شده است.

دستورات مکمل

دستور مکمل ۱ (CFW)

توسط این دستور می توان مکمل یک یک کلمه ورودی را در یک کلمه خروجی قرارداد. همانگونه که می دانید، مکمل 0 برابر 1 و مکمل 1 برابر 0 می باشد. پس برای بدست آوردن مکمل یک یک کلمه کافیسست ارزش بیتهای آن را عکس نمود.

L	IB 0
CFW	
T	QB 0
BE	

در این مثال زمانی که بایت صفر ورودی صفر باشد، خروجیهای بایت صفر روشن می باشند. زمانی که هر کدام از بیت های IB 0 فعال شوند، بیت متناظر آنها در QB 0 خاموش می گردد.

دستور مکمل ۲ (CSW)

توسط این دستور می توان مکمل ۲ یک کلمه ورودی را در خروجی قرار داد. همانطور که می دانید برای بدست آوردن مکمل ۲ یک عدد کافیسست به مکمل ۱ آن یک عدد اضافه نمایید.

در شکل زیر نحوه عملکرد دستور CFW نشان داده شده است.

Bit Patterns

0 1 0 1 1 1 0 1 0 0 1 1 1 0 0 0

↓ Ones Complement Integer

1 0 1 0 0 0 1 0 1 1 0 0 0 1 1 1

Bit Patterns

0 1 0 1 1 1 0 1 0 0 1 1 1 0 0 0

"23,864"
integer

↓ Twos Complement Integer

1 0 1 0 0 0 1 0 1 1 0 0 1 0 0 0

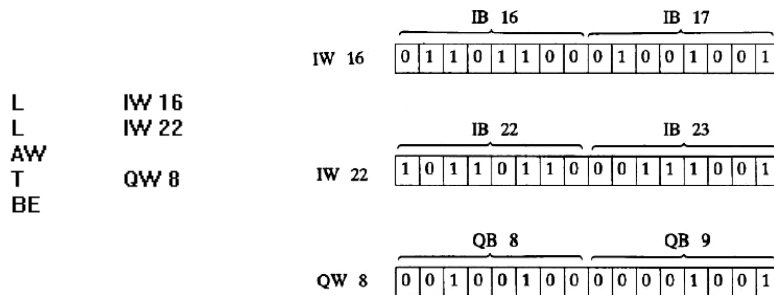
"-23,864"
integer

مکمل ۲

دستور عملهای منطقی

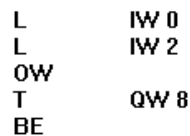
دستور AW

توسط این دستور می توان محتوای دو بایت یا دو کلمه را بیت به بیت با یکدیگر AND و نتیجه را در خروجی منتقل نمود.



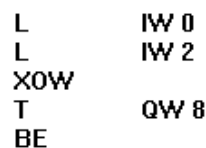
دستور OW

توسط این دستور می توان محتوای دو بایت یا دو کلمه را بیت به بیت با یکدیگر OR و نتیجه را در خروجی منتقل نمود.



دستور XOW

توسط این دستور می توان بیتهای دو کلمه ورودی را بیت به بیت با هم XOR نمود و به خروجی ارسال کرد.



دستورات افزایشی/کاهشی

I دستور افزایشی

توسط این دستور می توان حداکثر ۲۵۵ واحد معادل یک بایت به ورودی اضافه نمود.

L IW 22
I 15
T QW 14
BE

IW 22

0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 $111001101_{(2)} = 461_{(10)}$

QW 14

0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 $111011100_{(2)} = 476_{(10)}$

در این مثال به ورودی IW 22 که دارای عدد ۴۶۱ بوده ۱۵ واحد نیز اضافه و نتیجه به خروجی QW 14 منتقل شده است.

D دستور کاهشی

توسط این دستور می توان حداکثر ۲۵۵ واحد معادل با یک بایت از کلمه ورودی کم نمود.

L IW 22
D 100
T QW 8
BE

IW 22

0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 $111001101_{(2)} = 461_{(10)}$

QW 8

0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 $101101001_{(2)} = 361_{(10)}$

دستور افزایشی/کاهشی

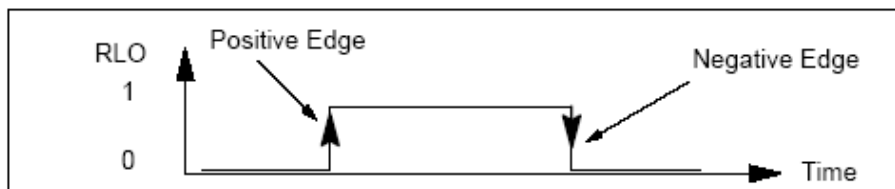
در مورد دستورات I و D بیان کردیم که حداکثر مقداری که می توان به کلمه ورودی اضافه یا از آن کسر نمود ۲۵۵ واحد می باشد. حال در صورتی که در اجرای برخی از برنامه ها به جمع و یا تفریق نمودن ورودی با مقادیری بیش از مقادیر ذکر نیاز داشته باشیم با مشکل مواجه خواهیم شد. برای رفع این مشکل از دستور ADD استفاده می کنیم. به کمک این دستور می توان اعدادی را در فاصله +۳۲۷۶۸ و -۳۲۷۶۸ قرار دارند با عدد مورد نظر جمع نمود. در هنگام استفاده از این دستور عدد اضافه شونده می تواند مثبت یا منفی باشد و با فرمت KF در برنامه استفاده می شود.

L FW 0
ADD KF +3000
T FW 10
BE

در برنامه مثال قبل، به محتوای FW 0 مقدار ۳۰۰۰ واحد اضافه و نتیجه به FW 10 منتقل می شود.

دستورات تشخیص لبه

این دستورات یکی از دستورات پر کاربرد در برنامه نویسی می باشند. توسط این دستورات می توان لبه های ایجاد شده (بالا یا پایین) در یک پالس را آشکار نمود. با استفاده از این دستورات تغییر ورودی از قطع به وصل (خاموش به روشن) یا از وصل به قطع (روشن به خاموش) آشکار می شود. مثلاً هر گاه آشکارسازی قطع به وصل (لبه بالا رونده) یک حسگر مورد نظر باشد، PLC به ازای هر قطع به وصل در یک مرور اجازه عبور برنامه از طریق اتصال مربوطه را می دهد. دستور تغییر از وصل به قطع (روشن به خاموش) نیز به همین ترتیب و در جهت عکس دستور قبل عمل می کند. مثلاً یک کلید را فرض کنیم. زمانی که کلید را فشار می دهیم، اگر از دستور تغییر مثبت استفاده کرده باشیم خروجی فعال می شود. ولی اگر از دستور تغییر منفی استفاده کرده باشیم، زمانی که کلید را رها کنیم خروجی فعال می شود.



در PLC های سری S5 بر خلاف S7، دستوراتی جهت تشخیص لبه وجود ندارد. برنامه نویس می بایست با یک ایده، اقدام به ایجاد یک برنامه نماید که بتواند کار دستورات تشخیص لبه را انجام دهد.

به مثال زیر دقت نمایید.

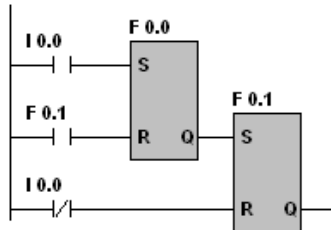
در این مثال می خواهیم یک شمارنده طراحی کنیم به گونه ای که بازه شمارش آن خیلی بیشتر از شمارنده هایی که در بحث های قبلی بررسی شد باشد.

	A	I 0.0
	JC	=M000
	BEU	
M000:	L	FW 2
	ADD	KF +1
	T	FW 2
	BE	

در این مثال انتظار می رود که با اعمال هر پالس به ورودی I0.0، به مقدار FW2 یک واحد افزوده شود. اگر این برنامه را در محیط نرم افزار تست کنیم، ملاحظه می شود که با هر بار فعال و غیر فعال کردن I0.0، به یکباره چند ده واحد به مقدار FW2 اضافه می شود. این مقدار افزایش هم بستگی به سرعت سیکل دارد.

برای حل این مشکل می بایست برنامه افزایش مقدار، حساس به لبه گردد. به عبارت دیگر برنامه باید به گونه باشد که با هر بار فعال کردن I0.0، تنها یک بار عملیات جمع انجام شود.

Segment 1

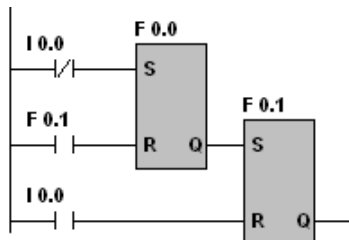


Segment 2

	A	F 0.0
	JC	=M000
	BEU	
M000:	L	FW 2
	ADD	KF +1
	T	FW 2
	BE	

در این برنامه، با هر بار فعال کردن ورودی I0.0، یک پالس بر روی F0.0 تولید می شود. عرض این پالس به اندازه یک سیکل پردازش برنامه می باشد. در نتیجه مقدار FW2 به اندازه یک واحد افزایش می یابد.

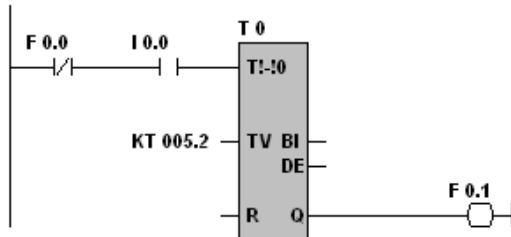
اگر نیاز به تشخیص لبه منفی باشد، می توان از برنامه زیر استفاده نمود.



مثال

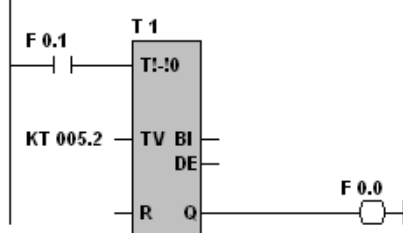
در این مثال می خواهیم برنامه ای طراحی کنیم که ۱۶ بیت از خروجی PLC به حالت چشمک زن تبدیل شوند.

Segment 1

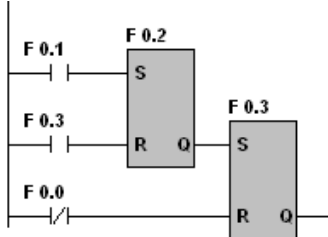


I	Q	I	Q	I	Q	I	Q	I	Q
0	1	2	3	4	5	6	7	0	1
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Segment 2



Segment 3



Segment 4

	A	F 0.2
	JC	=TEMP
	L	KH 0000
	T	QW 0
	BEU	
TEMP:	L	KH FFFF
	T	QW 0
	BE	

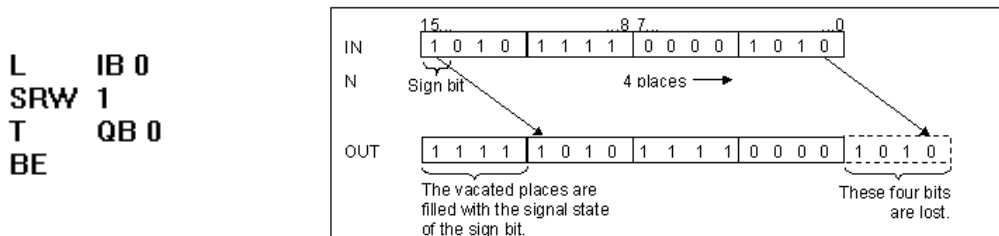
در این مثال تا مادامی که ورودی I0.0 فعال می باشد، ۱۶ خروجی PLC، ۵ ثانیه روشن و ۵ ثانیه خاموش می شوند.

View	Dec	Hex	Bin
QW0	-1	FFFF	11111111 11111111

دستورات جابه جایی SHIFT

دستور SRW (شیفت به سمت راست)

توسط این دستور می توان بیت های یک عدد صحیح را به تعداد N مرتبه به سمت راست شیفت داد.



173

دستور SLW (شیفت به سمت چپ)

توسط این دستور می توان بیت های یک عدد صحیح را به تعداد N مرتبه به سمت چپ شیفت داد.

L IB 0
SLW 2
T QB 0
BE

دستورات پرش ریاضی

در S5 علاوه بر دو دستور پرش JC و JU یکسری دستورات پرش نیز وجود دارند که به نتیجه عملیات ریاضی حساس می باشند.

دستور JZ:

در این دستور زمانی که نتیجه برخی اعمال ریاضی صفر باشد پرش انجام می گیرد و یا بعبارت دیگر دو عدد ورودی مساوی باشند اجرای برنامه از سطری که با برچسب مشخص شده است دنبال می شود.

دستور JN:

برخلاف دستور فوق در این دستور زمانی پرش صورت می گیرد که نتیجه اعمال ریاضی مخالف صفر باشد. پس در حقیقت این دستور نوعی دستور مقایسه برای نامساوی می باشد.

دستور JP:

در این دستور، پرش زمانی صورت می‌گیرد که حاصل عملیات مثبت باشد. یا بعبارت دیگر زمانی که عدد اول بزرگتر از عدد دوم باشد پرش صورت می‌گیرد.

دستور JM:

برخلاف دستور JP، در این حالت زمانی پرش انجام می‌شود که حاصل عملیات ریاضی عددی منفی باشد. یا به عبارت دیگر زمانی که عدد دوم بزرگتر از عدد اول باشد پرش صورت می‌گیرد.

دستور JO:

در این دستور زمانی که در انجام عملیات ریاضی سرریز پیش‌آید پرش صورت می‌گیرد. مثلاً نتیجه ضرب دو عدد در یک بایت جا نشود.

مثال

	L	FW 0
	L	FW 2
	-F	
	JM	=M000
	BEU	
M000:	L	KH 00FF
	T	FY 4
	BE	

در این مثال زمانی که نتیجه تفریق دو کلمه حافظه منفی شود، پرش به برجسب M000 صورت می‌گیرد و مقدار هگزی FF به بایت ۴ حافظه منتقل می‌شود. در غیر این صورت پرش انجام نمی‌گیرد و برنامه با اجرای دستور BEU تمام می‌شود. در واقع زمانی که مقدار FW0 از FW2 کمتر باشد، پرش صورت می‌گیرد.

مثال

	L	FW 0
	L	FW 2
	-F	
	JN	=M000
	L	KH 0000
	T	FY 4
	BEU	
M000:	L	KH 00FF
	T	FY 4
	BE	

برنامه فوق را تحلیل کنید.

دستور ENT

کاربرد این دستور در سری هایی از S5 می باشد که دارای ۴ آکومولاتور هستند. از این دستور در محاسبات ریاضی و برای جابه جا کردن محتوای آکومولاتورها استفاده می شود.

با اجرای این دستور تغییرات محتوای آکومولاتورها مطابق جدول شکل زیر می باشد.

Operation	Operand	Function
ENT	-	این دستور باعث می شود که پشته سمت آکومولاتورهای ۲ و ۳ پیش رود $\langle \text{ACCU 4} \rangle := \langle \text{ACCU 3} \rangle$ $\langle \text{ACCU 3} \rangle := \langle \text{ACCU 2} \rangle$ $\langle \text{ACCU 2} \rangle := \langle \text{ACCU 2} \rangle$ $\langle \text{ACCU 1} \rangle := \langle \text{ACCU 1} \rangle$ محتویات آکومولاتور ۱ و ۲ بدون تغییر می ماند

175

با اجرای این دستور محتوای آکومولاتور ۲ به آکومولاتور ۳ و محتوای آکومولاتور ۳ به سمت آکومولاتور ۴ پیش می رود.

مثال: برنامه ای بنویسید که عبارت $(30+3*4)/6$ را محاسبه کند.

	ACCU 1	ACCU 2	ACCU 3	ACCU 4
محتویات آکومولاتور قبل از محاسبه	a	b	c	d
L KF +30	30	a	c	d
L KF +3	3	30	c	d
ENT	3	30	30	c
L KF +4	4	3	30	c
x F	12	30	c	c
+ F	42	c	c	c
L KF +6	6	42	c	c
: F	7	c	c	c

دستورالعملهای تست بیت

توسط این دستور امکان تست بیت از یک کلمه وجود دارد. این ست می تواند برای تشخیص 0 با 1 باشد.

TB = تست بیت برای یک

TBN= تست بیت برای صفر

مثلا زمانی که بخواهیم بیت پنجم از IB 0 را برای صفر بودن تست کنیم، بصورت زیر عمل می کنیم.

مثال

```
L      KH 0000
T      FW 0
TBN    F 0.5
=      Q 0.0
BE
```

با اجرای این برنامه، یک بیت از یک کلمه برای صفر بودن تست می شود. اگر این بیت صفر باشد، خروجی Q0.0 فعال می باشد. با توجه به مقدار منتقل شده در این مثال، خروجی Q0.0 فعال می باشد.

176

مثال

```
L      KH AAFF
T      FW 0
TBN    F 0.5
=      Q 0.0
TB     F 1.7
=      Q 0.1
BE
```

برنامه فوق را تحلیل کنید

دستور TAK

با اجرای این دستور، محتوای آکومولاتور ۱ به آکومولاتور ۲ و محتوای آکومولاتور ۲ به آکومولاتور ۱ منتقل می شود.

مثال

```
L      KF +100
L      KF +200
T      FW 0
```

با اجرای این برنامه، عدد ۲۰۰ به FW 0 منتقل می شود.

دلیل:

با اجرای دستور L اول، عدد ۱۰۰ در آکومولاتور ۱ قرار می گیرد. زمانی که L دوم اجرا می شود، محتوای آکومولاتور ۱ به آکومولاتور ۲ منتقل و عدد ۲۰۰ در آکومولاتور ۱ قرار می گیرد. با اجرای دستور T عدد ۲۰۰ که در آکومولاتور ۱ قرار دارد به FW 0 منتقل می شود.

حال به برنامه زیر توجه کنید.

L	KF +100
L	KF +200
TAK	
T	FW 0

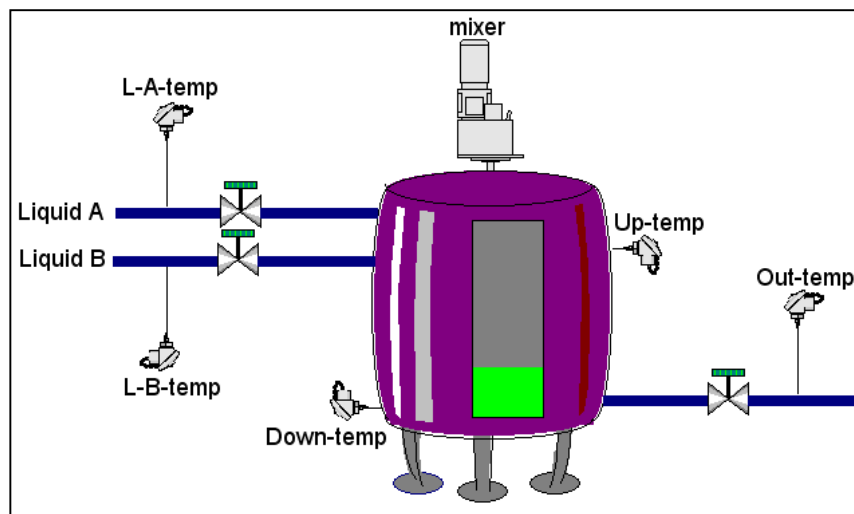
در این برنامه به دلیل اینکه محتوای آکومولاتورها جابه جا می شود، عدد ۱۰۰ به FW 0 منتقل می شود.

177

مثال

در این مثال جهت آشنایی هر چه بیشتر با دستورات روش STL و کاربرد آنها، یک برنامه بسیار جالب را مورد بررسی قرار می دهیم.

فرض کنید در یک پروسه صنعتی قرار است که دمای 5 نقطه از یک میکسر ثبت گردد. این دماها می بایست برای هر نقطه، دمای فعلی، حداقل دما و حداکثر دما باشد.



L-A-Temp: دمای ورودی مایع A

L-B-Temp: دمای ورودی مایع B

Down_Temp: دمای پایین مخزن

Up-Temp: دمای بالای مخزن

Out-Temp: دمای خروجی

اطلاعاتی که قرار است از هر نقطه ثبت گردد به صورت زیر می باشد.

نقطه	دمای فعلی	حداقل دما	حداکثر دما
L-A-Temp	FW10	FW20	FW30
L-B-Temp	FW12	FW22	FW32
Down_Temp	FW14	FW24	FW34
Up-Temp	FW16	FW26	FW36
Out-Temp	FW18	FW28	FW38

در ادامه فرض کنید که دماهای خوانده شده از هر PT100 در حافظه های زیر قرار می گیرد.

*دمای ورودی مایع A : FW0

*دمای ورودی مایع B : FW2

*دمای پایین مخزن : FW4

*دمای بالای مخزن : FW6

*دمای خروج : FW8

بحث نحوه پردازش سیگنال های آنالوگ در بخش های بعدی به طور کامل مورد بررسی قرار خواهد گرفت.

Segment 1

در این قسمت برنامه مربوط به ثبت دماهای فعلی نوشته می شود. جهت نوشتن برنامه مربوط به ثبت دماهای فعلی نیاز به عملیات و دستورات خاصی نبوده و تنها توسط دستورات بارگذاری و انتقال می توان دمای هر نقطه را در ستون اول جایگذاری نمود.

```
L      FW 0
T      FW 10
L      FW 2
T      FW 12
L      FW 4
T      FW 14
L      FW 6
T      FW 16
L      FW 8
T      FW 18
***
```

Segment 2

در این قسمت از برنامه، حداقل دمای مربوط به هر نقطه در ستون ۲ ثبت می شود.

	L	FW 10
	L	FW 20
	>F	
	JC	=M000
	TAK	
	T	FW 20
M000:	NOP	0
	L	FW 12
	L	FW 22
	>F	
	JC	=M001
	TAK	
	T	FW 22
M001:	NOP	0
	L	FW 14
	L	FW 24
	>F	
	JC	=M002
	TAK	
	T	FW 24
M002:	NOP	0
	L	FW 16
	L	FW 26
	>F	
	JC	=M003
	TAK	
	T	FW 26
M003:	NOP	0
	L	FW 18
	L	FW 28
	>F	
	JC	=M004
	TAK	
	T	FW 28
M004:	NOP	0

در برنامه این قسمت دمای فعلی با حداقل دمای قبلی در هر نقطه مقایسه می شود. در صورتی که دمای فعلی از دمای حداقل قبلی بیشتر باشد پرش به برچسب صورت گرفته و اتفاقی در سیستم با اجرای دستور NOP 0 رخ نمی دهد. اما اگر دمای فعلی از حداقل دمای قبلی بزرگتر نباشد، دمای فعلی که کوچکتر از دمای حداقل قبلی می باشد، توسط دستور TAK به آکومولاتور ۱ منتقل تا در ادامه توسط دستور T به نقطه حداقل جایگزین شود.

توسط دستور TAK محتوای آکومولاتور ۱ و آکومولاتور ۲ جابه جا می شوند.

Segment 3

در این قسمت از برنامه، حداکثر دمای مربوط به هر نقطه در ستون ۳ ثبت می شود.

	L	FW 10
	L	FW 30
	<F	
	JC	=M005
	TAK	
	T	FW 30
M005:	NOP	0
	L	FW 12
	L	FW 32
	<F	
	JC	=M006
	TAK	
	T	FW 32
M006:	NOP	0
	L	FW 14
	L	FW 34
	<F	
	JC	=M007
	TAK	
	T	FW 34
M007:	NOP	0
	L	FW 16
	L	FW 36
	<F	
	JC	=M008
	TAK	
	T	FW 36
M008:	NOP	0
	L	FW 18
	L	FW 38
	<F	
	JC	=M009
	TAK	
	T	FW 38
M009:	NOP	0

در برنامه این قسمت دمای فعلی با حداکثر دمای قبلی در هر نقطه مقایسه می شود. در صورتی که دمای فعلی از دمای حداکثر قبلی کوچکتر باشد پرش به برجسب صورت گرفته و اتفاقی در سیستم با اجرای دستور NOP 0 رخ نمی دهد. اما اگر دمای فعلی از حداکثر دمای قبلی کوچکتر نباشد دمای فعلی که بزرگتر از دمای حداکثر قبلی می باشد توسط دستور TAK به آکومولاتور ۱ منتقل تا در ادامه توسط دستور T به نقطه حداقل جایگزین شود.

دستورات تبدیل (Convert)

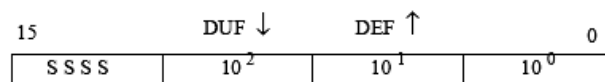
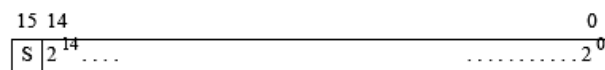
توسط این دستورات امکان تبدیل یک عدد از یک قالب به قالب دیگر امکان پذیر می باشد. به عنوان مثال فرض کنید در یک برنامه هدف جمع نمودن یک عدد حقیقی با یک عدد صحیح می باشد. در این صورت انجام این عملیات توسط دستورات بررسی شده در قسمتهای قبلی امکان پذیر نمی باشد. در این حالت می توان با تبدیل عدد صحیح به یک عدد حقیقی، فرمت هر دو عدد را یکسان و سپس عملیات جمع را انجام داد.

دستور DEF

توسط این دستور می توان یک عدد صحیح ۱۶ بیتی را از فرمت دودویی به BCD تبدیل نمود.

دستورالعمل DUF

توسط این دستور می توان یک عدد صحیح ۱۶ بیتی را از فرمت BCD به دودویی تبدیل نمود.



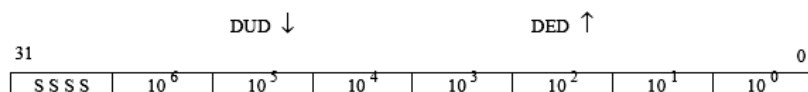
S (sign): • = مثبت
 ۱ = منفی

دستور DED

توسط این دستور می توان یک دو کلمه ای را از فرمت BCD به دودویی تبدیل نمود.

دستور DUD

توسط این دستور می توان یک دو کلمه ای را از فرمت دودویی به BCD تبدیل نمود.



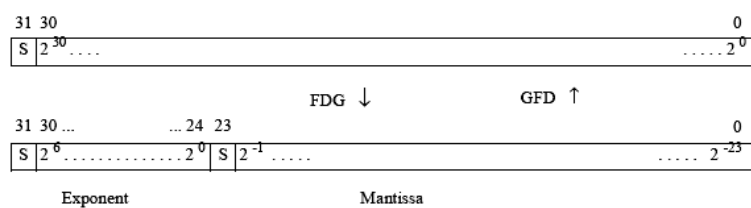
S (sign): 0 = positive
 1 = negative

دستورالعمل FDG

مقدار ACCU 1 بعنوان یک عدد ۳۲ بیتی صحیح تفسیر می شود. بعد از عمل تبدیل، ACCU 1 شامل یک عدد اعشاری می باشد.

دستورالعمل GFD

مقدار ACCU 1 بعنوان یک عدد اعشاری تعبیر می شود. بعد از عمل تبدیل، ACCU 1 شامل یک عدد ۳۲ بیتی صحیح می باشد.



کاربرد دستور GFD

در حقیقت با کمی دقت متوجه می شوید که توسط این دستور می توان یک عدد اعشاری را به نزدیکترین عدد صحیح کوچکتر گرد نمود.

عدد اعشاری	GFD	عدد صحیح (۳۲ بیت)
+5,7	→	5
-2,3	→	-3
-0,6	→	0
+0,9	→	0

پردازش سیگنال های آنالوگ

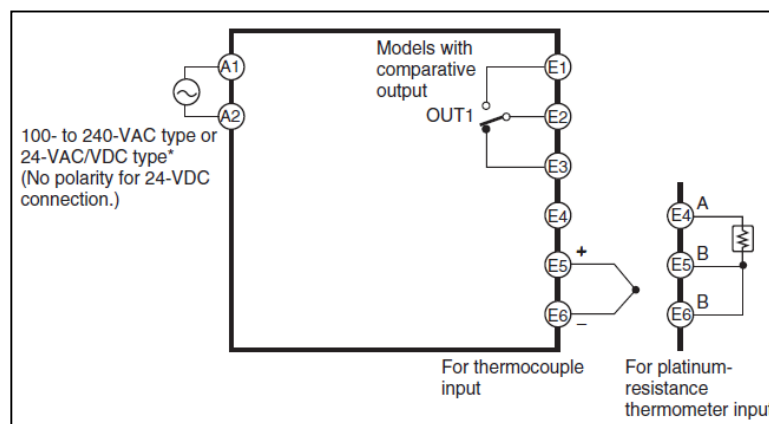
آشنایی با کمیت های آنالوگ

در صنعت امروز کمیت هایی مانند دما، فشار، وزن و سرعت به عنوان کمیت های آنالوگ به حساب می آیند. تجهیزاتی که قادر به اندازه گیری این کمیت ها می باشند نیز به عنوان تجهیزات آنالوگ محسوب می شوند. در پروسه های صنعتی معمولاً برای هر یک از موارد فوق یک کنترلر به عنوان اندازه گیر تعبیه شده است. این کنترلرها به صورت مستقیم قادر به اندازه گیری کمیت توسط سنسور مربوطه می باشند. به عنوان مثال کنترلرهای دما به عنوان یکی از تجهیزات پر کاربرد در مراکز صنعتی می باشند. در این کنترلرها این امکان وجود دارد که یک PT100 و یا یک ترموکوپل به صورت مستقیم جهت اندازه گیری حرارت متصل شود. در این صورت عددی که بر روی نمایشگر این کنترلر نشان داده می شود دمای اندازه گیری شده می باشد.

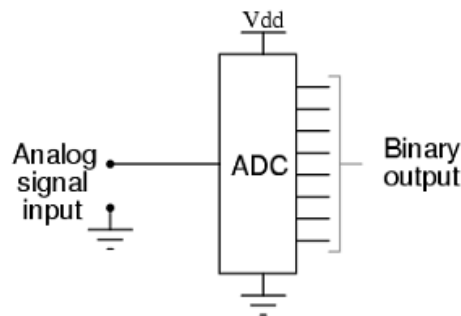
183



بسیاری از این کنترلرها دارای خروجی آنالوگ نیز می باشند که از این خروجی آنالوگ به راحتی می توان در کنترل کننده های دیگر به عنوان ورودی آنالوگ استفاده نمود. همچنین اکثر کنترلرها مجهز به تیغه های فرمان می باشند که این فرمان یک فرمان دیجیتال با توجه به پارامتر S.P می باشد. از این خروجی نیز می توان در ورودی دیجیتال PLC استفاده نمود.



اما زمانی که در یک پروسه هدف اندازه گیری کمیت های بیان شده توسط PLC می باشد، این عمل توسط کارتهای ورودی آنالوگ انجام می شود. در سیگنال های آنالوگ تغییرات جزئی در سیگنال نیز دارای مفهوم بوده که می بایست در سیستم در نظر گرفته شود. زمانی که جهت اندازه گیری یک سیگنال آنالوگ از PLC استفاده می شود قبل از اینکه سیگنال مورد نظر توسط CPU پردازش شود می بایست به فرم دیجیتال تبدیل گردد. این عملیات تبدیل توسط مبدل های ADC کارتهای ورودی آنالوگ انجام می شود. نتیجه مقادیر تبدیل شده را می توان در یک حافظه ذخیره نمود.



در قسمت زیر به برخی از تجهیزات ورودی آنالوگ اشاره شده است:

- ۱- ترموکوپل ها (TC)
- ۲- ترمومترها (RTD)
- ۳- سنسورهای وزن (Load Cell)
- ۴- سنسورهای فشار
- ۵- سنسورهای سطح
- ۶- سنسورهای فاصله
- ۷- سنسورهای فلومتر

سنسورهای دما: حرارت یکی از مهمترین کمیت های فیزیکی می باشد که جهت اندازه گیری آن از مدت ها پیش تجهیزات مختلفی وارد بازار شده است. به عنوان مثال یک دماسنج جیوه ای را در نظر بگیرید که به آسانی توسط آن می توان دمای یک اتاق را اندازه گیری نمود.

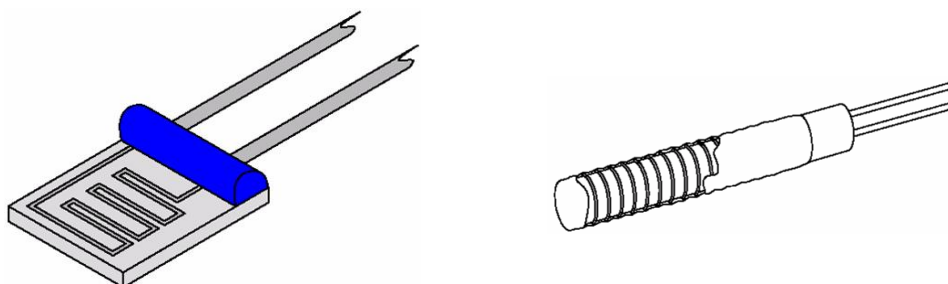
ترموتر RTD: یکی از سنسورهای معروف جهت اندازه گیری دقیق دما در محیط های صنعتی ترمومترها می باشند که به سنسور های مقاومتی خطی نیز معروف هستند. معروفترین نمونه از این سنسورها PT100 می باشد که دقت بسیار بالایی دارد و اخیرا برای اندازه گیری تا دماهای ۸۵۰ درجه نیز ساخته شده است.

در این سنسورها مقاومت فلز به ازای دماهای مختلف تغییر می کند. به همین دلیل در ساخت سنسورهای مقاومتی معمولاً از فلزهایی با دقت بالا استفاده می شود که دارای حساسیت بالایی می باشند. عموماً این فلزها از جنس نیکل، آلیاژ نیکل، آهن و مس می باشند.

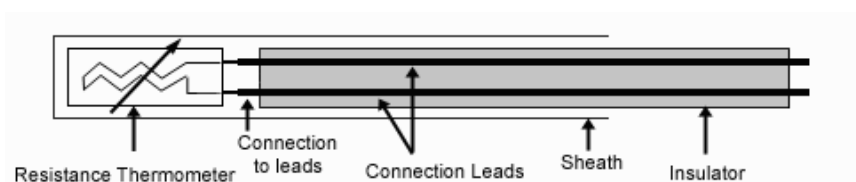


185

سنسورهای RTD معمولاً در دو نوع سیم پیچی شده و فیلم نازک ساخته می شوند. نوع اول از پیچیده شدن یک رشته بسیار ظریف از سیم پلاتینی به شکل یک پیچک به دور یک ماده غیر هادی مثل سرامیک ساخته می شود. RTD نوع دوم از یک فیلم نازک با نشاندن یک لایه پلاتین به شکل مقاومت روی یک زیر لایه سرامیکی ساخته می شود. از مزایای این نوع قیمت ارزان و پاسخ حرارتی بسیار بالا می باشد.



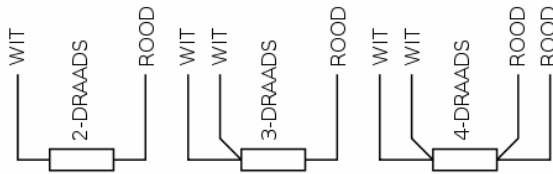
ساختمان RTD



این عناصر معمولاً نیاز به یک عایق دارند. در دماهای حدود زیر ۲۵۰ درجه از عایق PVC، سیلیکون و یا PTFE استفاده می شود. در دماهای بالاتر از این معمولاً از فیبر شیشه ای و یا سرامیکی استفاده می شود.

سنسور PT100 در دمای صفر درجه دارای مقاومت ۱۰۰ اهم می باشد و به ازای هر درجه افزایش دما ۰.۳۸۵ اهم به مقاومتش افزوده می شود.

مزایای استفاده از RTD :



- ۱- دقت بالا
- ۲- پایداری طولانی مدت
- ۳- محدوده دمایی مناسب
- ۴- توانایی تبادل خوب

در بین ۳ نوع RTD مدل ۴ سیمه دارای بیشترین دقت و استفاده از نوع ۳ سیمه مرسوم تر می باشد.

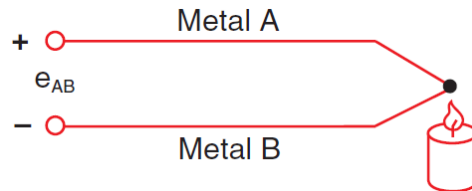
Values for various popular resistance thermometers

Temperature in °C	Pt100 in Ω	Pt1000 in Ω	PTC in Ω	NTC in Ω	NTC in Ω	NTC in Ω	NTC in Ω	NTC in Ω
	Typ: 404	Typ: 501	Typ: 201	Typ: 101	Typ: 102	Typ: 103	Typ: 104	Typ: 105
-50	80.31	803.1	1032					
-45	82.29	822.9	1084					
-40	84.27	842.7	1135			50475		
-35	86.25	862.5	1191			36405		
-30	88.22	882.2	1246			26550		
-25	90.19	901.9	1306		26083	19560		
-20	92.16	921.6	1366		19414	14560		
-15	94.12	941.2	1430		14596	10943		
-10	96.09	960.9	1493		11066	8299		
-5	98.04	980.4	1561	31389	8466			
0	100.00	1000.0	1628	23868	6536			
5	101.95	1019.5	1700	18299	5078			
10	103.90	1039.0	1771	14130	3986			
15	105.85	1058.5	1847	10998				
20	107.79	1077.9	1922	8618				
25	109.73	1097.3	2000	6800			15000	
30	111.67	1116.7	2080	5401			11933	
35	113.61	1136.1	2162	4317			9522	
40	115.54	1155.4	2244	3471			7657	
45	117.47	1174.7	2330				6194	
50	119.40	1194.0	2415				5039	

جدول مقادیر خروجی در ترمومترهای مختلف

ترموکوپل TC

ترموکوپل از اتصال دو فلز غیر هم جنس در یک نقطه بدست می آید. اصول کار این وسیله بر مبنای اثر سیبک است. توماس سیبک کشف نمود که وقتی دو فلز غیر هم جنس از یک سمت به هم متصل شوند و محل پیوند حرارت داده شود در سمت دیگر اختلاف پتانسیل کوچکی به وجود می آید.



187

سیگنال خروجی ترموکوپل یک ولتاژ ضعیفی می باشد که این سیگنال می تواند توسط یک مبدل به یک سیگنال استاندارد تبدیل شود. یکی از ویژگیهای ترموکوپل ها استفاده از آنها در دماهای بالا می باشد. از این رو ترموکوپل ها نیز در انواع مختلف ساخته می شوند.

B	0	1820
C	-120	2315
E	-270	1000
J	-210	1200
K	-270	1372
L	-200	900
N	-270	1300
R	-50	1769
S	-50	1769
T	-270	400
U	-270	600

°C -300 0 300 600 900 1200 1500 1800 2100 2400

ترموکوپل های نوع J و K از مدل های پر کاربرد در صنعت امروز می باشند. ترموکوپل ها معمولاً در دماهای پایین دارای دقت خوبی نمی باشند. از این رو از آنها معمولاً در دماهای بالا استفاده می شود. جنس آلیاژ استفاده شده در هر نوع از ترموکوپل منجر به تفاوت در بازه دمایی قابل اندازه گیری می شود. به عنوان مثال ترموکوپل نوع K دارای آلیاژ (نیکل کروم-نیکل آلومینیم) می باشد

Nickel-Chromium
vs.
Nickel-Aluminum



لودسل (Load Cell)

لودسل نیز یک سنسور جهت اندازه گیری وزن در سیستم های توزین می باشد. معمولاً شرکت های سازنده لودسل های خود را در انواع کششی، خمشی، فشاری و پیچشی تولید می کنند. اندازه گیری وزن به طریق دیجیتال در ترازوهای الکترونیکی نیز نیازمند لودسل می باشد. امروزه انواع مختلف لودسل با ظرفیت های متفاوت در ساخت ترازوها و باسکول های الکترونیکی کاربرد فراوان دارد.

لودسل شامل یک هسته فلزی (از آلیاژ خاص) و تعدادی Strain gauge (مجموعه ای از مقاومت های الکتریکی) می باشد که در اثر اعمال نیرو مانند تمام مواد تغییر شکل می دهد. اما پس از برداشتن نیرو به حالت اولیه خود برمی گردد. میزان برگشت پذیری این ماده تعیین کننده کیفیت و دقت و دیرپایی لودسل است. در اکثر سیستم های توزین کنترلرهای جهت اندازه گیری و نمایش وزن توسط شرکت های سازنده طراحی می شوند. این کنترلرها به صورت مستقیم به لودسل متصل می شوند و معمولاً مجهز به خروجی آنالوگ نیز می باشند.

سنسورهای لودسل تغییرات وزن را بر اساس تغییرات ولتاژ و وزن بار وارده حس می کند. دقت لودسل که به عنوان یکی از پارامترهای مهم در انتخاب لودسل می باشد معمولاً به صورت یک درصد از ظرفیت کل بیان می شود. به عنوان مثال اگر یک لودسل 100kg با دقت کلی $\pm 0.03\%$ موجود باشد این لودسل در بدترین حالت اندازه گیری دارای تolerانس $\pm 30\text{gr}$ می باشد. در کاتالوگ یک لودسل معمولاً پارامترها و مشخصات ذکر شده در قسمت زیر بیان می شود:

Capacity (ظرفیت): ظرفیت کل لودسل

Overload (اضافه بار): حداکثر درصد کامل بار قبل از آسیب دیدن لودسل. (این عدد به طور معمول 150% می باشد)

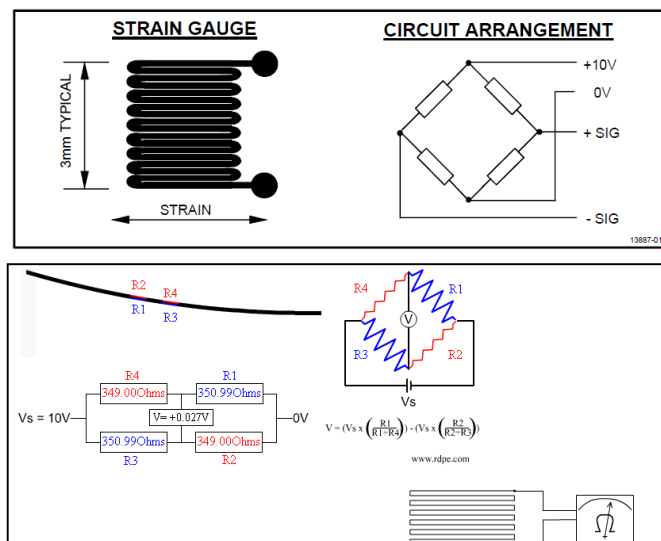
Accuracy (دقت): این پارامتر ممکن است به صورت اصطلاحاتی همچون خطی، پسماند، تکرارپذیری و خزش بیان شود. این ارقام به صورت یک درصد از ظرفیت کلی بیان می شود.

Sensitivity (حساسیت): حساسیت لودسل مقدار واقعی ولتاژ خروجی را زمانی که حداکثر بار بر روی لودسل قرار می گیرد تعیین می کند. این پارامتر به صورت mV/V بیان می شود. به عنوان مثال اگر لودسل دارای خروجی 3mV/V و ولتاژ تحریک 10VDC باشد پس خروجی لودسل در بار کامل 30mv می باشد.

Thermal Sensitivity (حساسیت حرارتی): این پارامتر مقدار تغییرات خروجی را به ازای تغییرات دما بیان می کند. این پارامتر نیز معمولاً به صورت درصد کل بار بیان می شود. $(\text{Load}/^\circ\text{C})$

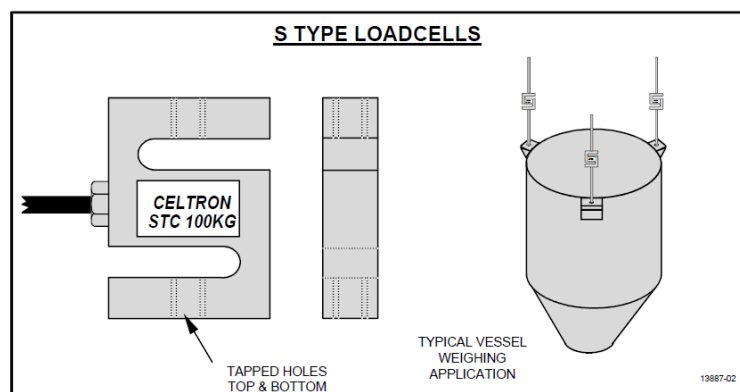
نحوه کار لودسل: زمانی که یک نیروی مکانیکی به لودسل وارد می شود، strain gauge ها کشیده می شوند. با کشیده شدن آنها مقاومتشان تغییر یافته که در نتیجه این تغییر مقاومت توسط مدارات مقاومتی و همچنین ولتاژ تحریک به صورت تغییرات ولتاژ ظاهر می شود. بدنه لودسل ها معمولاً از جنس فولاد ضدزنگ و یا آلومینیوم می باشد. زمانی که لودسل جهت اندازه گیری وزن در یک سیستم توزین نصب می شود بیشترین فشار به strain gauge می بایست وارد شود.

strain gauge در سال ۱۹۳۸ توسط دو بچه باهوش اختراع گردید. strain gauge خود از سیم نازک و یا فویل که به بدنه لودسل چسبیده شده اند تشکیل شده است. رابطه بین فشار و تغییر در مقاومت در لودسل ها خطی می باشد.

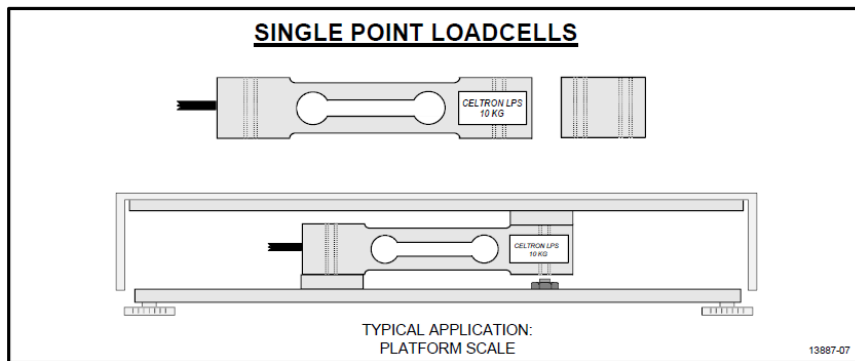


همانطور که در بحث قبلی بیان شد لودسل ها از لحاظ ساخت دارای تپ های مختلفی می باشند که در ادامه با برخی از مدل های پرکاربرد آشنا می شویم.

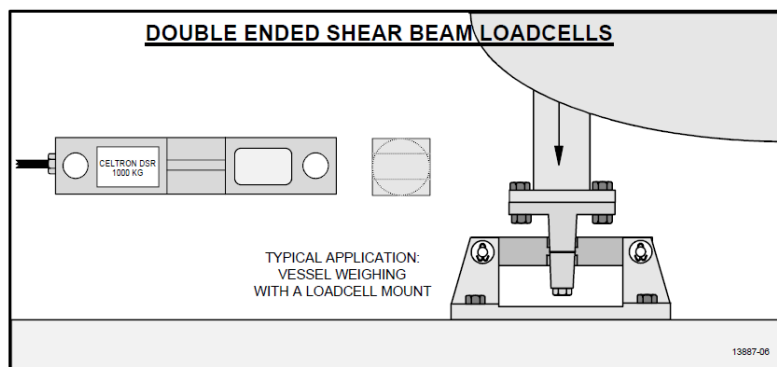
لودسل کششی تپ S: در این نوع لودسل فشار وارده بر لودسل منجر به تغییر در طول لودسل می گردد. البته لازم به ذکر است که این تغییر بسیار کوچک می باشد. در این نوع نیروی وارده می بایست به صورت عمود بر مرکز لودسل اعمال شود. این لودسل ها معمولاً در ظرفیت های 20Kg تا 10000kg در دسترس می باشند.



لودسل تک پایه (Single Point): این سری به عنوان یکی از پرکاربردترین نوع لودسل در سیستم های توزین می باشد. از این نوع معمولا در کفه های کوچک و باسکول ها استفاده می شود. یکی از مزایای این لودسل نسبت به سایر لودسل ها این است که در این نوع نیاز نیست که بار توسط یک نقطه به لودسل متصل شود. این لودسل ها معمولا برای ظرفیت های 200gr تا 2000kg طراحی می شوند.

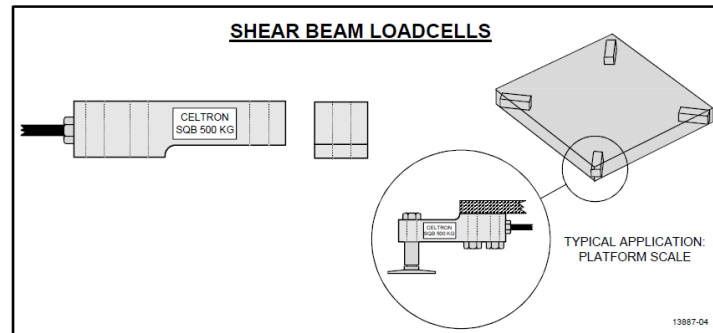


لودسل دو منتهی (Double ended): این لودسل ها جهت نصب در سیستم هایی که دارای بار ناهموار هستند مناسب می باشد. این لودسل ها معمولا برای ظرفیتهای 500kg تا 50000kg طراحی می شوند.



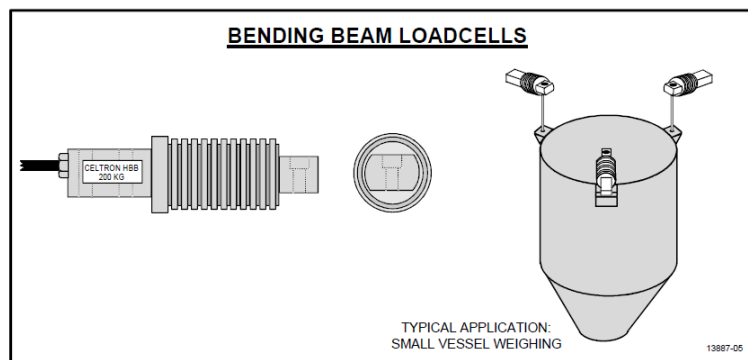
لودسل خمشی (Shear-Beam): این لودسل ها دارای یک پیچ ثابت با یک ساختار ثابت می باشند. در این نوع لودسل وزن در انتهای دیگر به لودسل وارد می شود. در این صورت لودسل دچار خمش شده و strain gauge ها تحت تاثیر قرار می گیرند. از این نوع اغلب در پایه ها و مفصل های گردان استفاده می گردد. این لودسل ها معمولا برای ظرفیتهای 100kg تا 10000kg طراحی می شوند.

در شکل زیر کاربرد و شکل این نوع لودسل را ملاحظه می کنید.



191

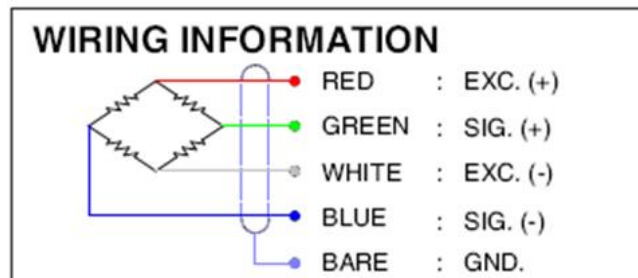
لودسل خمشی (Bending-Beam): این نوع لودسل با دارا بودن IP بالا و همچنین دقت بالا مناسب جهت استفاده در صنایع لبنیات و مواد غذایی می باشد. این لودسل دارای بدنه تمام فولاد ضدزنگ می باشد و در سیستم های توزین کوچک با دقت بالا بسیار کاربرد دارد. این لودسل ها معمولا برای ظرفیتهای 5kg تا 1000kg طراحی می شوند.



اتصالات:

لودسل ها معمولا دارای ۴ یا ۶ رشته سیم می باشند. در لودسل های ۴ سیمه، ۲ سیم برای تغذیه ثابت DC و ۲ سیم دیگر نیز خروجی سیگنال می باشد.

در لودسل های ۶ سیمه دو سیم اضافی جهت اتصال به کنترلر جهت اندازه گیری استفاده می شود. این دو سیم تحت عنوان سیم های sense شناسایی می شوند. از این سیم جهت اندازه گیری دقیق ولتاژ تحریک در لودسل استفاده می شود.



192

نکات:

- * همیشه در اتصالات لودسل از کابل های شیلد دار استفاده کنید.
- * هیچ گاه از کابل تکه شده جهت برقراری اتصالات استفاده نکنید.
- * حداقل امکان سعی کنید مسیر کابل کوتاه باشد.
- * مسیر عبور کابل مربوط به لودسل ها از مسیر کابل های قدرت مجزا باشد.
- * زمانی که از چند لودسل جهت اتصال به یک کنترلر استفاده می کنید حتما سعی شود از junction box استفاده گردد.



سنسورهای فشار

سنسورهای فشار نیز از سنسورهای پر کاربرد در صنعت می باشند. از این سنسورها جهت اندازه گیری فشار در یک مخزن و یا در یک لوله استفاده می شود. سنسورهای فشار نیز در انواع مختلف ساخته و طراحی می گردند. ساده ترین نوع که فقط جهت نمایش استفاده می شود در شکل زیر نشان داده شده است. در این نوع فشارسنج ها از یک لوله خمیده استفاده می شود که یک طرف آن مسدود و به عقربه ای متصل شده است. سمت دیگر لوله باز است و به مخزنی که قرار است فشار آن اندازه گیری شود متصل می گردد. در این حالت فشار وارد به فشارسنج منجر به منحرف شدن عقربه می شود. این فشارسنج ها دارای دقت کمی می باشند و هیچ کنترلی در خروجی آنها وجود ندارد. در این نوع هیچ سیگنال الکتریکی به ازای فشارهای مختلف تولید نمی شود و صرفاً جهت نمایش استفاده می گردد.

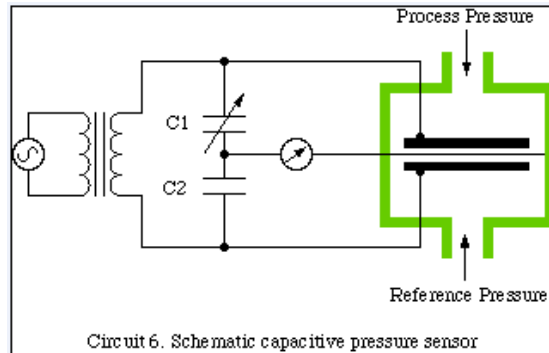


نوع دیگری از سنسورهای فشار، سیگنالی تابع فشار تولید می کنند. این سیگنال یک سیگنال الکتریکی می باشد که می توان آن را به یک کنترلر جهت پردازش ارسال نمود. سنسورهای فشار از لحاظ ساختار دارای مدل های متنوعی می باشند که در ادامه به برخی از آنها اشاره شده است:

نوع Capacitive (خازنی)

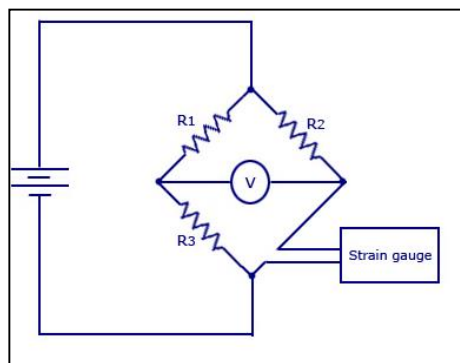
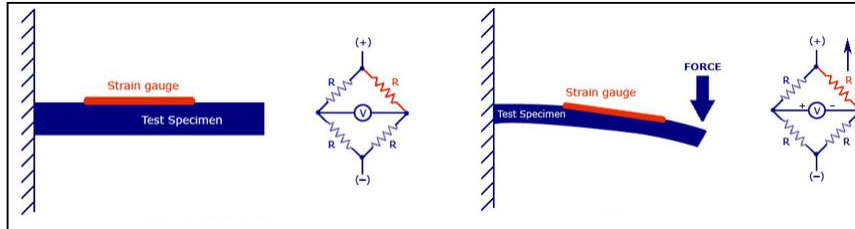
همانطور که می دانید ظرفیت C یک خازن صفحه ای بستگی به ثابت دی الکتریک نسبی، فاصله بین صفحات d و مساحت A دارد. از این بحث در بسیاری از سنسورها استفاده می شود. نوعی از سنسورهای فشار نیز با این روش کار می کنند. در این سنسورها تغییرات فشار موجب جابه جایی صفحات خازن می شود. در این صورت تغییر در ظرفیت خازن منجر به تغییر سیگنال الکتریکی می گردد.

در سنسور فشار نوع خازنی فشار تفاضلی به دیافراگم اعمال می شود که باعث می شود دیافراگم به یکی از صفحات خازن نزدیک شده و از دیگری دور شود. بنابراین ظرفیت خازن تغییر می کند که این تغییر متناسب با فشار اعمال شده به دیافراگم است.



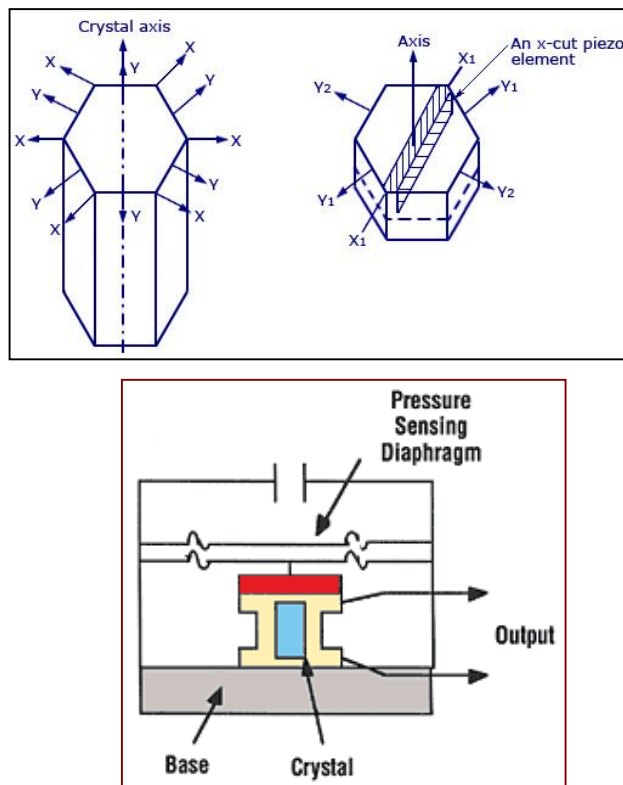
نوع strain gauge

در این نوع از یک مدار پل استفاده می شود. در هر حالت بدون بار مدار در حالت تعادل می باشد. زمانی که فشار یا نیرویی به این صفحه اعمال می شود طول صفحه تغییر می یابد. تغییر در طول صفحه موجب تغییر در مقاومت می شود. بنابراین خروجی پل وابسته به فشار وارده می باشد. این نوع از فشارسنج ها از سیم های نازکی که بر روی یک صفحه نازک قرار گرفته اند تشکیل شده است.



نوع Piezoelectric

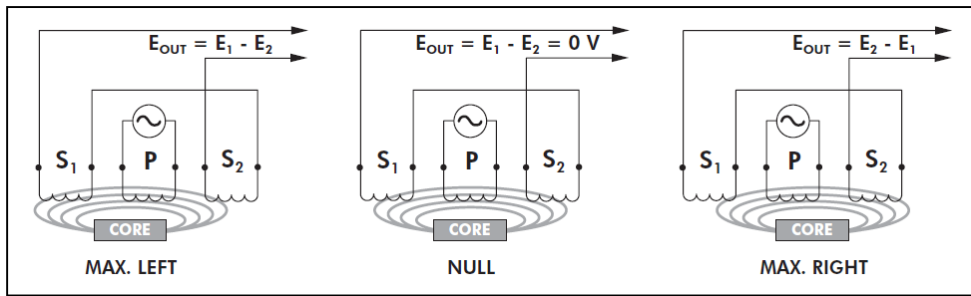
نوعی دیگر از سنسورهای فشار سنسورهایی هستند که تحت عنوان سنسورهای پیزوالکتریک شناخته می شوند. اصطلاح پیزو بیانگر تغییر در مقاومت الکتریکی ماده ای است که در معرض یک نیروی مکانیکی همچون کشش یا فشار قرار داده می شود. این پدیده در کریستال های خاص رخ می دهد و به خوبی در نیمه هادی ها نمایان می شود. در این سنسورها کریستال های خاص را تحت فشار قرار می دهند و این فشار منجر به تولید سیگنال توسط مدارهای مختلف می گردد.



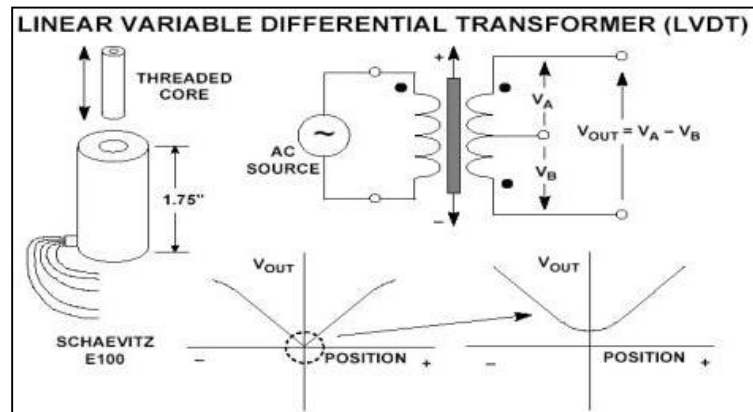
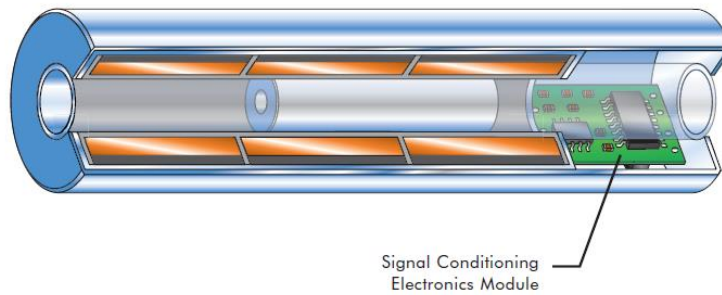
نوع LVDC

برخی دیگر از سنسورهای فشار می توانند توسط جابه جایی هسته در فشارهای مختلف تولید سیگنال کنند. LVDT یک تجهیز الکترومکانیکی می باشد که توسط جابه جایی هسته می تواند تولید سیگنال کند. LVDT از یک سیم پیچ اولیه و دو سیم پیچ ثانویه تشکیل شده است. در وسط این دو سیم پیچ یک میله نرم به عنوان هسته قرار گرفته شده است. تغییرات فشار محفظه هسته موجود در LVDT را جابه جا می کند و LVDT این جابه جایی را تبدیل به ولتاژ الکتریکی می کند.

همانطور که در شکل زیر ملاحظه می کنید جابه جایی هسته منجر به تغییر ولتاژ در خروجی LVDT می شود.



جابه جایی هسته در LVDT می تواند توسط مدارات الکترونیکی تبدیل به سیگنالهای استاندارد جهت پردازش کنترلرها شود.



در ادامه با مشخصات یک سنسور فشار آشنا می شویم.

مدل

Series RSP0180

این سنسور دارای مشخصات زیر می باشد.

۱- دقت بسیار بالا $\pm 0.25\%$ ۲- خروجی 3mv/v

۳- جنس بدنه از فولاد ضد زنگ

۴- resolution بالا

۵- دارا بودن خروجی کاملا خطی

۶- پایین ترین رنج اندازه گیری 50 PSI

۷- بالاترین رنج اندازه گیری 40000 PSI

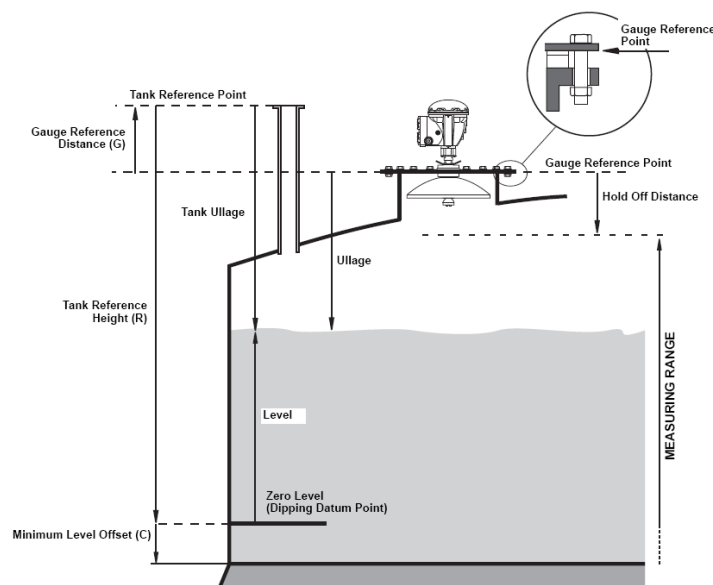


197

این سنسور دارای ولتاژ تحریک حداکثر 15Vdc می باشد. اما شرکت های سازنده معمولاً ولتاژ تحریک 10Vdc را توصیه می کنند.

سنسورهای Radar

در بسیاری از محیط های صنعتی جهت اندازه گیری سطح مخازن و یا به عبارت دیگر سطح مایع از سنسورهای راداری استفاده می شود. در این سنسورها از ارسال امواج به سطح مایع و بازتاب آن جهت اندازه گیری سطح مایعات استفاده می شود. این نوع سنسورها معمولاً مجهز به یک نمایشگر جهت نمایش سطح مخزن به صورت درصد و یا متر می باشند



همانطور که در قسمت فوق مشاهده می کنید یکی از قسمت های سنسورهای راداری ترانسمیتر می باشد که توسط این قسمت سطح مایع در قالب یک سیگنال استاندارد جریانی به یک کنترلر همانند PLC ارسال می شود. بازه این سیگنال ۴ الی ۲۰ میلی آمپر می باشد.



سنسورهای فلومتر

فلوسنسور یا سنسوردبی وسیله ای است که میزان جریان مایع را اندازه گیری می کند. به عبارت دیگر فلومتر یا دبی سنج وسیله ایست که حجم مواد عبوری را نسبت به زمان معرفی می کند. در بسیاری از کاربردها نیاز به اندازه گیری مقدار دقیق فلو عبوری می باشد که در این صورت می توان از این سنسورها استفاده نمود.



مفاهیم اولیه:

هدف: پیدا کردن حجم سیالی همانند آب که در واحد زمان از یک نقطه به نقطه دیگر منتقل می شود.

فرمول فلوجریان: $Q = AV$

Q: سرعت جریان (فلو)

A: سطح مقطع

V: سرعت

199

تخلیه یا میزان flow rate معمولاً به صورت Megalitres/day (ML/day) یا Litres/second (L/s) معرفی می شود.

برای محاسبه میزان flow rate نیاز به دو پارامتر می باشد:

۱- اندازه لوله یا ابعاد کانال

۲- شتاب یا سرعت آب که این پارامتر می تواند توسط فشار یا ارتفاع افزایش یابد.

برخی از روش های اندازه گیری فلو

* توربینی Turbine

* آلتراسونیک با پدیده دوپلر Ultrasonic Doppler

* آلتراسونیک با روش زمان انتقال Ultrasonic

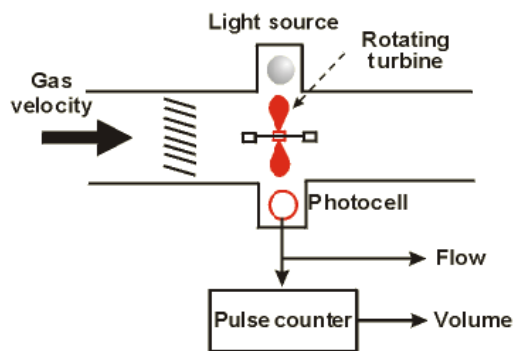
* مغناطیسی Magnetic

* گرمائی Thermal

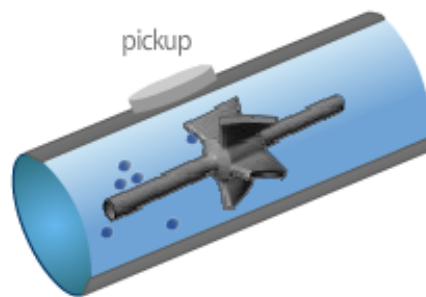
* گردابی Vortex

ساده ترین نوع فلومترها، فلومتر توربینی می باشد. در این ساختار از یک توربین در مسیر مایع استفاده می شود. مایع با برخورد به توربین، پره های آن را چرخانده که در این حالت می توان توسط یک سنسور پروکسمیتی پالس هایی را تولید نمود. فرکانس پالس های تولید شده به میزان فلوی عبوری در مسیر بستگی دارد.

با دریافت پالس های تولید شده توسط PLC و طراحی یک برنامه جهت شمارش پالس ها در واحد زمان، می توان مقدار فلو مایع را مشخص نمود.



از این فلومترها معمولاً برای جریان هایی با سرعت کم استفاده می شود. این فلومترها معمولاً در سرعت های بالا به دلیل اعمال انرژی مکانیکی سیال و حرکت توربین آسیب پذیر می باشند. این آسیب پذیری می تواند به دلیل سایش بلبرینگ ها باشد.



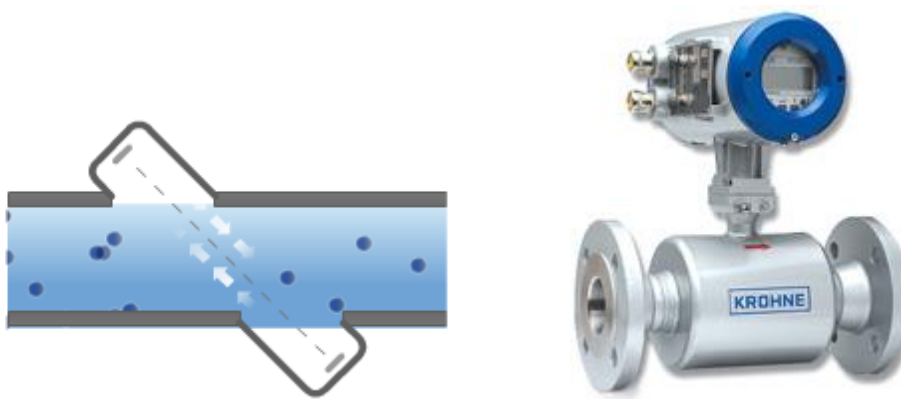
در شکل زیر نمونه هایی از فلومترهای توربینی را مشاهده می کنید.



فلومترهای آلتراسونیک با پدیده دوپلر

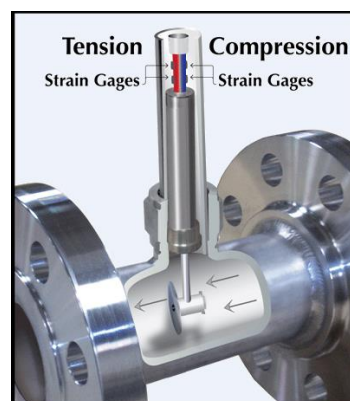
در این سنسورها با استفاده از امواج صوتی در یک لوله سرعت جریان یک سیال تعیین می شود. در این فلومترها زمانی که در داخل لوله جریان برقرار نباشد فرکانس امواج آلتراسونیک همان فرکانس ارسال شده به داخل لوله می باشد. با برقراری فلو یا جریان در مسیر فرکانس موج منعکس شده با توجه به پدیده دوپلر متفاوت می باشد. در این حالت با سریعتر حرکت کردن مایع، جابه جایی فرکانس به صورت خطی افزایش می یابد. پردازش بر روی سیگنال فرستاده شده با توجه به امواج منتقل شده و بازتابش آن می تواند فلو سیال را مشخص کند. زمان ارسال و دریافت امواج بین مبدل در هر دو سمت بالا و پایین لوله می باشد.

201



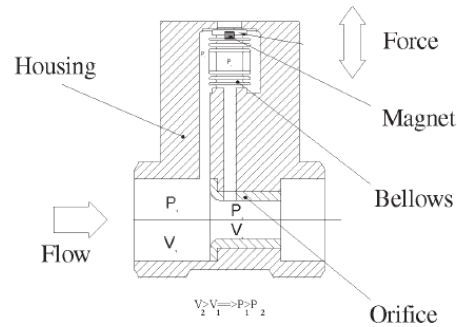
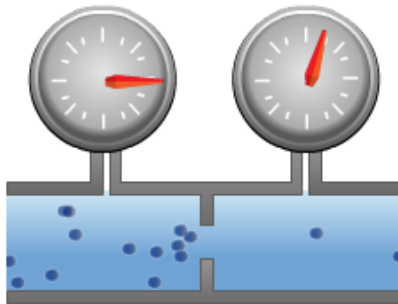
فلومترهای Target

در این فلومترها با استفاده از نیروی وارد شده توسط جریان سیال به یک هدف مشخص در فلومتر می توان فلو سیال را اندازه گیری نمود. نیروی اعمال شده به هدف توسط جریان، متناسب است با افت فشار. در قسمت فوقانی این هدف مجموعه ای از strain gauge ها وجود دارند که با توجه به نیروی وارده به هدف کشیده می شوند.



فلومترهای اختلاف فشار

این فلومترها از معادله برنولی جهت اندازه گیری فلو استفاده می کنند. در این فلومترها از انقباض لوله جهت ایجاد اختلاف فشار در مسیر لوله استفاده می شود. با افزایش جریان، افت فشار نیز بیشتر می گردد.



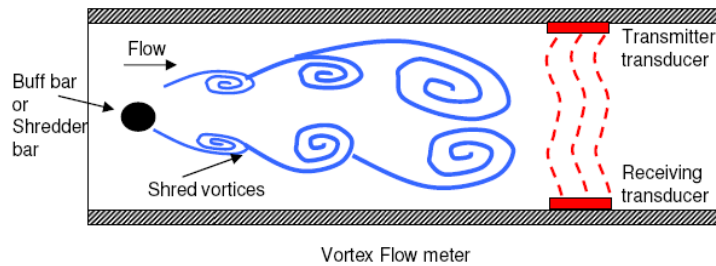
202



فلومترهای گردابی Vortex

وقتی که سیال در یک گرفتگی با یک مانعی برخورد کند نوسان رخ می دهد. نمونه هایی از این نوسانات در طبیعت عبارتند از صدای تولید شده توسط برگ درختان در هنگام وزش باد و یا اهتزاز پرچم در زمان وزش باد. نکته ای که در نوسانات تولید شده در این دو مثال وجود دارد کاهش نوسانات در زمان کاهش باد می باشد. این فلومترها نیز در اثر عبور سیال تولید نوسان می کنند. اساس کار این سنسورها بدین صورت است که با قرار دادن یک مانع بر سر راه سیال، جریان گردابی در پشت مانع تولید می گردد که باعث بوجود آمدن افت فشار می گردد. این افت فشار متناسب با سرعت سیال می باشد.

حال بوسیله یک سنسور پیزوالکتریک (سنسور فشار) می توان این نوسانات مکانیکی را به سیگنال الکتریکی تبدیل کرد.

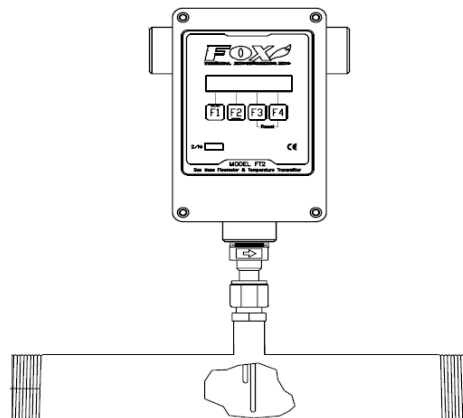
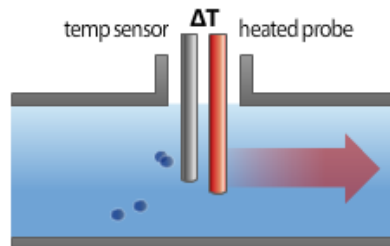


Vortex Flow meter



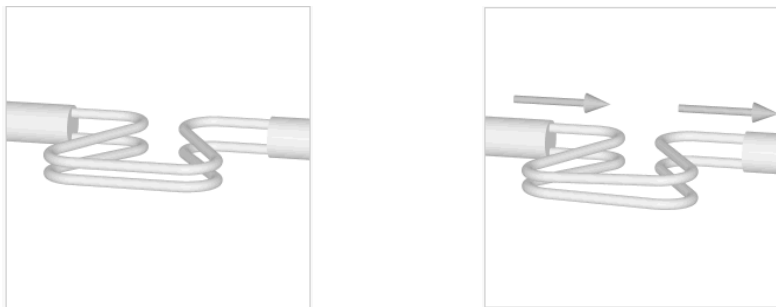
فلومترهای گرمائی Thermal

این فلومترها با توجه به خواص حرارتی سیالات در یک مجرا یا یک لوله اقدام به اندازه گیری فلو می کنند. در این فلومتر از یک منبع گرما یا هیتر جهت تولید حرارت در مسیر استفاده می شود. با افزایش جریان، گرمای بیشتری در مسیر از دست می رود. این گرما نیز توسط یک سنسور حرارتی اندازه گیری می شود. در واقع در این فلومترها مقدار مشخصی گرما به سیال اعمال می شود و تغییرات دما متناسب با دبی سیال می باشد. این روش برای اندازه گیری جریان انواع گازها ایده آل می باشد.



فلومترهای Coriolis Mass

در این روش دو لوله موازی توسط یک سیم پیچ تحریک با فرکانس از پیش تعیین شده به نوسان در می آید. با عبور سیال درون دو لوله نیرویی در جهت عمود بر لوله ها وارد می شود که بنام نیروی کوریولیس معروف است. این نیرو هنگامی تولید می شود که سیال درون لوله در حال جابه جایی بوده و به طور همزمان لوله در حال نوسان می باشد. این نیرو در ورودی و خروجی فلومتر به صورت متضاد عمل می کند که باعث تغییر شکل جزئی در دو لوله می شود. این انحراف توسط سنسورهای مغناطیسی که در ورودی و خروجی لوله هستند اندازه گیری شده و اختلاف فاز میان نوسان دو لوله ناشی از نیروی کوریولیس بدست می آید. این اختلاف فاز با دبی جرمی سیال درون لوله متناسب است. همچنین فرکانس رزونانس لوله ها مطابق با چگالی سیال تغییر خواهد کرده که از این کمیت جهت اندازه گیری چگالی سیال استفاده می شود. دامنه انحراف دو لوله نیز بستگی به دمای سیال دارد که در برخی دستگاه ها با این روش می توان دمای سیال را تخمین زد. این روش فلومتری جهت اندازه گیری فلوی جرمی مایعات و گازها صرفنظر از ضریب هدایت، چگالی، دما، فشار و چسبندگی سیال به کار می رود.



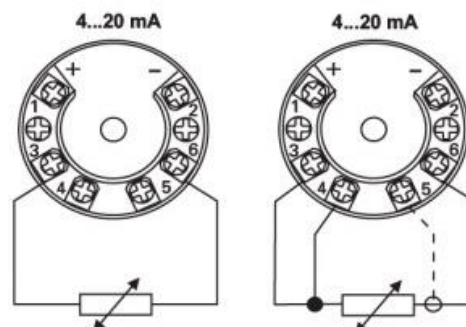
ترانسمیتر چیست ؟

ترانسمیتر وسیله ای است که یک سیگنال الکتریکی ضعیف را دریافت کرده و به سطوح قابل قبول برای کنترلرها و مدارهای الکترونیکی تبدیل می کند ، مثلاً یک حلقه فیدبک سیگنالی در سطح میکروولت یا میلی ولت یا میلی آمپر تولید می کند و این سیگنال ضعیف می تواند با عبور از ترانسمیتر به سیگنالی در سطوح صفر تا ده ولت و یا ۴ تا ۲۰ میلی آمپر تبدیل شود. ترانسمیترها عموماً از قطعاتی مثل op-amp برای تقویت و خطی کردن این سطوح ضعیف سیگنال استفاده می کند . سنسورها و ملحقات آنها مثل ترانسدیوسرها را در گروه های بزرگی تحت عنوان ابزار دقیق قرار داده و آنها را بر اساس نوع انرژی قابل استفاده و روشهای تبدیل ، دسته بندی می کنند.

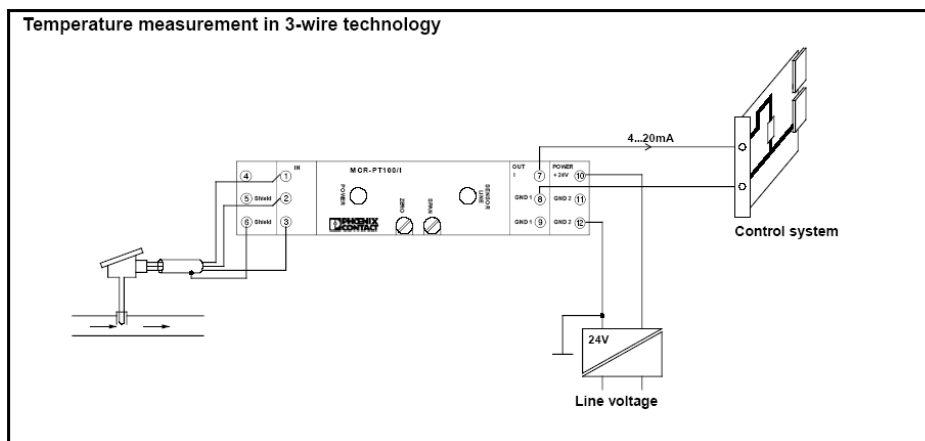
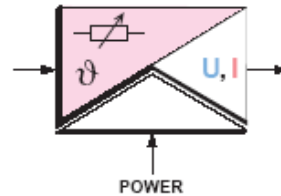


آشنایی با مشخصات چند مدل ترانسمیتر

در بسیاری از کاربردها جهت اتصال سنسور حرارتی PT100 به ورودی آنالوگ یک PLC از ترانسمیتر استفاده می شود. در واقع این ورودی آنالوگ مختص PT100 نبوده و قابلیت دریافت سیگنال های استاندارد را دارا می باشد. در این صورت می بایست از ترانسمیتر با خروجی استاندارد استفاده نمود. برای این منظور شرکتهای سازنده بسیاری اقدام به ساخت ترانسمیتر مختص PT100 می نمایند. با استفاده از این تجهیز خروجی PT100 تبدیل به یک سیگنال استاندارد از جنس جریان می شود. البته لازم به ذکر است که ترانسمیتر با یک ولتاژ ثابت DC می بایست تغذیه شود که این ولتاژ معمولاً بین 10 الی 30 ولت DC می باشد.



همچنین شرکت فونیکس نیز دارای مبدل های متنوع برای استفاده در سیگنال های آنالوگ می باشد. یک نوع از این مبدل ها، مبدل MCR-PT100 می باشد. توسط این MCR خروجی یک PT100 را می توان به یک سیگنال استاندارد تبدیل نمود.



همانطور که در شکل فوق ملاحظه می کنید این مبدل نیاز به یک تغذیه ثابت دارد. خروجی این مبدل نیز یک سیگنال استاندارد از نوع جریان بوده که جهت ارسال به بردهای کنترلی استفاده می شود. یکی از مزایای این مبدل ها اتصال انواع RTD به ورودی می باشد.

ماژول های ورودی آنالوگ در PLC

کارت های ورودی آنالوگ در PLC های سری S5 در انواع مختلف در دسترس می باشند. برخی از این کارت ها قادر به پردازش سیگنال های آنالوگ استاندارد و برخی دیگر نیز مختص سنسورهای حرارتی RTD و یا ترموکوپل می باشند. اما به طور کلی سیگنال های آنالوگ می توانند از جنس ولتاژ جریان و یا مقاومت باشند. در قسمت زیر برخی از سیگنال های استاندارد معرفی شده است.

به عنوان مثال در قسمت زیر مشخصات فنی ماژول های آنالوگ در سری های 135U و 155U را مشاهده می کنید.

207

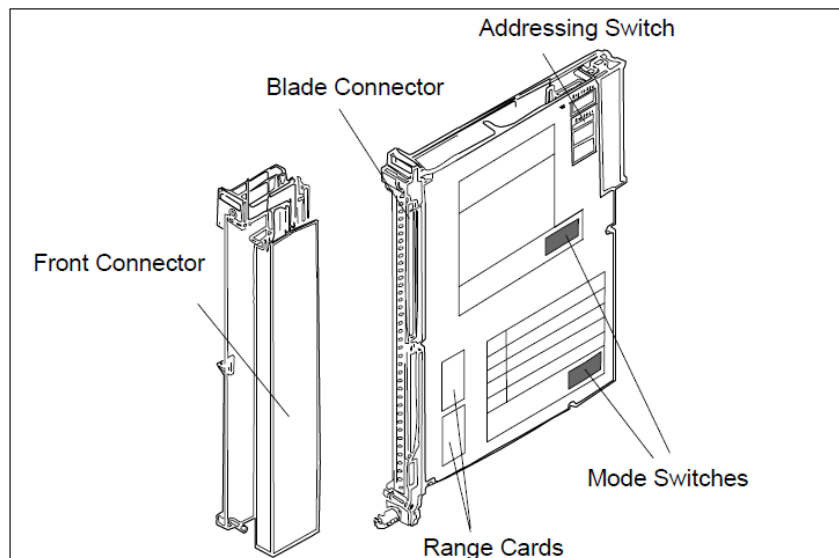
Analog Input Modules and Cards				
Order No. of the Module	No. of Inputs	Isolation/Groups	Range Card (4 Channels)	Order No. of the Range Card
6ES5 460-4UA13	8	Yes/8 inputs isolated from M, M _{ext} and L+/L-	± 12.5/50/500 mV/Pt 100	6ES5 498-1AA11
6ES5 465-4UA12 6ES5 465-4UA13	16/8	No/none	± 50/500 mV/Pt 100 ± 1 V ± 10 V ± 20 mA 4 to 20 mA/4-wire trans. ± 5 V 4 to 20 mA/4-wire trans.	6ES5 498-1AA11 6ES5 498-1AA21 6ES5 498-1AA31 6ES5 498-1AA41 6ES5 498-1AA51 6ES5 498-1AA61 6ES5 498-1AA71
Analog Input Modules				
Order No. of the Module	No. of Inputs	Isolation/Groups	Measuring Range	
6ES5 463-4UA12 6ES5 463-4UB12,	4	Yes/4 inputs isolated from M, M _{ext} and L+/L-	1 V, 10 V, 20 mA, 4 to 20 mA	
6ES5 466-3LA11 6ES5 466-4UA11	16/8	Yes	1.25 V, 2.5 V, 5 V, 10 V 20 mA, 4 to 20 mA	

سیگنال های مشخص شده در جدول فوق معمولاً برای اکثر ماژول های ورودی آنالوگ تعریف شده می باشند. اما برخی از کارت ها نیز قابلیت دریافت ولتاژ میلی ولت و همچنین سیگنال از جنس مقاومت را نیز دارا می باشند. به همین دلیل تنوع در کارت های AI بسیار زیاد می باشد. حتماً در موقع خرید ماژول AI به خروجی سنسور مورد استفاده و بازه سیگنال آن توجه نمایید. زیرا کارت های AI در انواع متنوعی تولید می گردند. همانطور که بیان شد برخی از آنها قابلیت دریافت ولتاژ میلی ولت و یا حتی قابلیت اتصال مستقیم ترموکوپل و یا ترمومترها را دارا می باشند و برخی دیگر نیز این قابلیت ها را ندارند.

در ادامه با برخی از ماژول های آنالوگ آشنا می شویم.

ماژول ورودی آنالوگ 460

بر روی ماژول های آنالوگ یک Dip Switch برای تعیین آدرس ماژول تعبیه شده است. در قسمت زیر قسمت های مختلف یک کارت آنالوگ را ملاحظه می کنید.



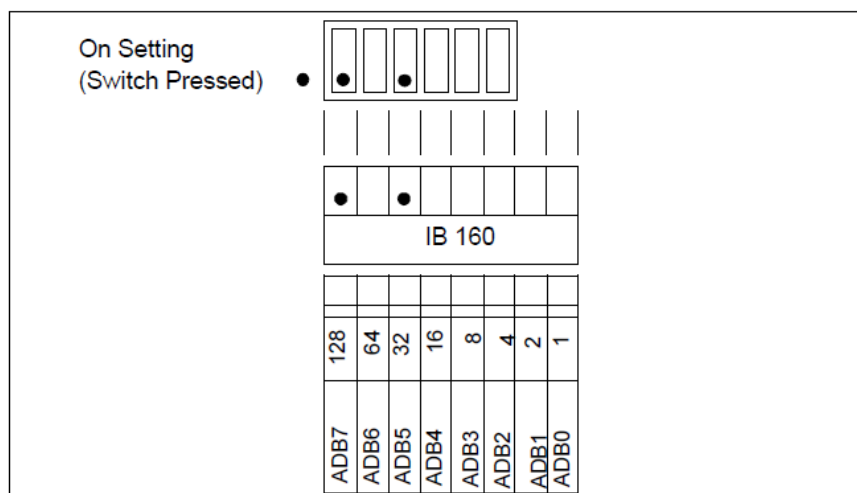
ماژول 460 نیاز به یک تغذیه 24v خارجی دارد. این تغذیه به ترمینال های +F و -F فرانت کانکتور می بایست متصل شود. در شکل زیر یک مثال جهت تعیین آدرس ماژول ورودی آنالوگ ارائه شده است.

Example:

Analog input module with 8 inputs

The address is the sum of the significances set with the individual coding switches.

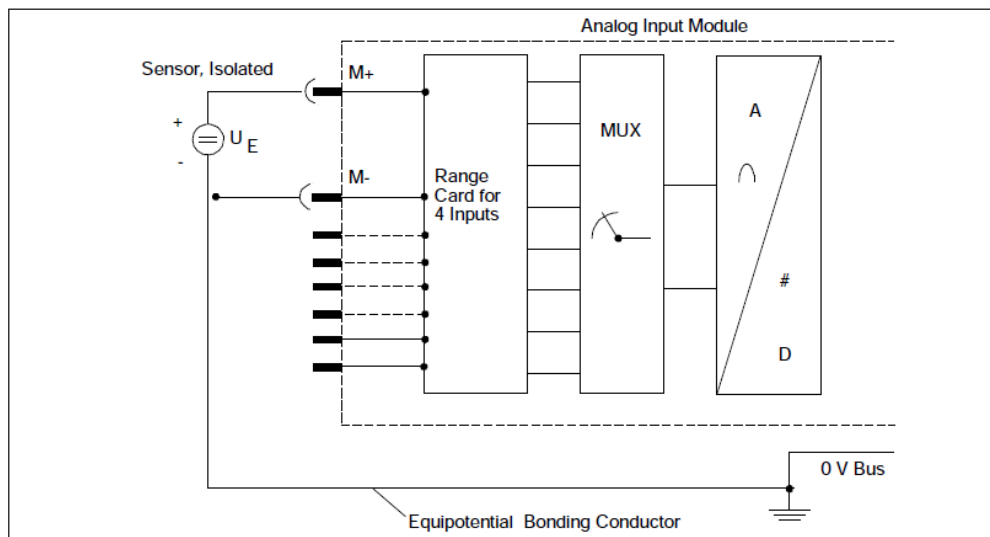
$$160 = 128 + 32 = 2^7 + 2^5$$



با تعیین Dip Switch در مثال قبل، آدرس اولین کانال کارت 160 تعیین می شود. آدرس کانال بعدی 162 می باشد و به همین ترتیب آدرس کانال ۸ برابر ۱۷۴ می باشد. در این حالت آدرس کارت بعدی می بایست از ۱۷۶ شروع شود. در واقع آدرس یک ماژول، از مجموع سوئیچ هایی که در وضعیت On قرار گرفته شده اند، محاسبه می شود.

$$160 \text{ to address } 160 + 7 \times 2 = 174$$

در شکل زیر نحوه اتصال یک سنسور با خروجی ولتاژ را به کارت 460 ملاحظه می کنید.



کار با ترموکوپل

ترموکوپل از دو فلز غیرهمجنس تشکیل شده است که در یک نقطه به هم متصل می شوند. وقتی این نقطه در محیط گرم قرار گیرد، در دو سر آزاد ترموکوپل ولتاژ میلی ولت ظاهر می شود.

* ترموکوپل برخلاف سایر ورودی های آنالوگ که تغذیه بیرونی دارند هیچ تغذیه ای لازم ندارند و در واقع خود ترموکوپل به عنوان یک منبع جریان محسوب می شود.

* سیگنال میلی ولت ترموکوپل به شدت نویز پذیر است و اگر انتخاب کابل و کابل کشی به طور صحیح انجام نگیرد ممکن است سیستم کنترل در شرایط بروز نویز با مشکل مواجه شود.

* در اتصال ترموکوپل به کارت ورودی آنالوگ، در اتصال سیم ترموکوپل به سیم مسی در محل اتصال ترموکوپل جدیدی ایجاد می شود که منجر به خطا در اندازه گیری می شود. از این رو به روش مختلف می توان این خطا را برطرف کرد.

جبران سازی خطا در ترموکوپل

در اتصال ترموکوپل به سیستم کنترل با دو موضوع مواجه هستیم.

Hot Junction: اتصال گرم که در داخل محیط گرم قرار می گیرد

Cold Junction: یا محل اتصال سرد که در بیرون محیط گرم قرار دارد و محلی است که سیم ترموکوپل به سیم یا فلز دیگر متصل می شود.

در ترموکوپل ها در محل اتصال سرد خطا ایجاد می شوند. منشاء این خطا ایجاد ترموکوپل جدید در محل اتصال فلز ترموکوپل به فلز دیگر است. اگر سیم ترموکوپل کوتاه باشد، محل اتصال سرد نقطه ای است که سیم ترموکوپل به سیم مسی متصل گردیده است. اگر سیم ترموکوپل بلند باشد و به طور مستقیم به ترمینال ورودی PLC متصل گردد، محل اتصال سرد، ترمینال ورودی است که جنس آن با جنس سیم ترموکوپل متفاوت است.

استفاده از سیم هم جنس ترموکوپل

اگر از ترموکوپل با سیم بلند استفاده کنیم یا اگر در محل اتصال از سیم هم جنس ترموکوپل استفاده کنیم خطای فوق برطرف خواهد شد. ولی در محل اتصال به PLC باز ترموکوپل ضعیفی خواهیم داشت. این روش در فواصل طولانی هزینه زیادی را در بر دارد.

استفاده از جبران ساز بیرونی

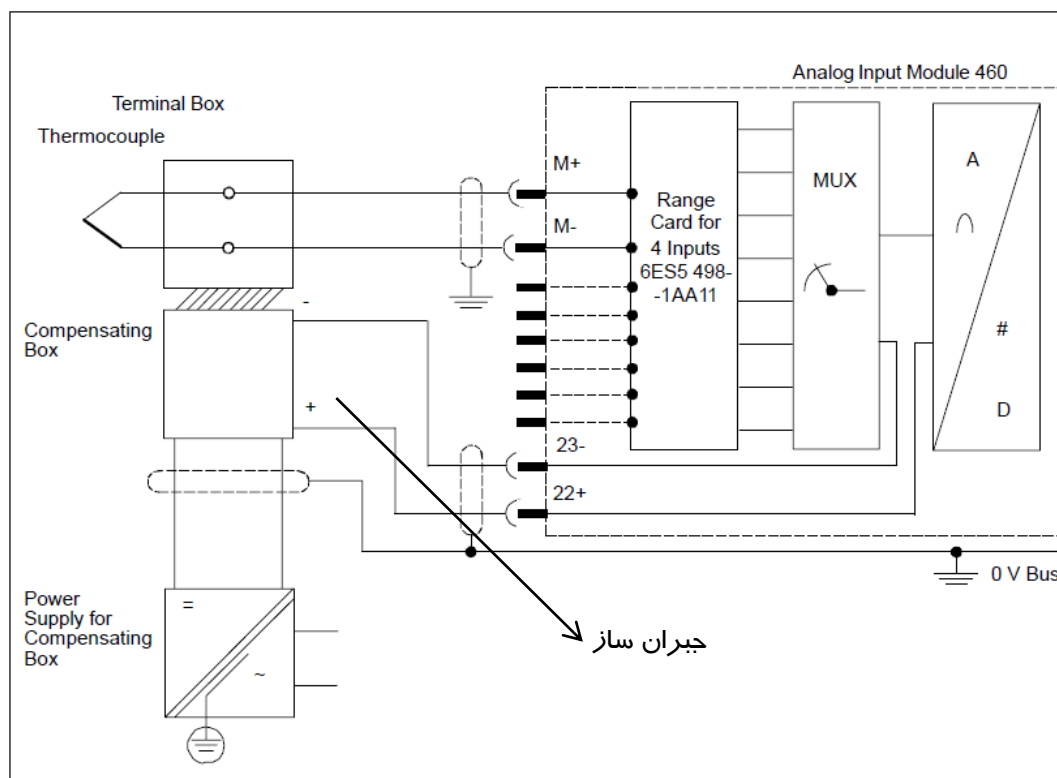
روش دیگری که نسبت به روش قبلی اقتصادی تر است این است که ترموکوپل را به همان سیم های مسی معمولی متصل کنیم. این روش برای مسافت های زیاد از لحاظ هزینه مقرون به صرفه تر است. برای جبران خطا ناشی از تشکیل ترموکوپل در محل **Cold Junction** با استفاده از سنسور جداگانه ای دمای محل اتصال سرد را اندازه گیری کرده و به کنترلر انتقال دهیم. کنترلر به صورت داخلی دمای اندازه گیری شده را با این دما اصلاح می کند.

برای این که این روش قابل استفاده باشد، معمولاً کارت های آنالوگ ورودی که سیگنال های ترموکوپل را دریافت می کنند دارای یک ورودی مجزا برای جبران ساز هستند.

شرکت زیمنس نیز یک جبران ساز خارجی تولید کرده است. در این تجهیز یک سنسور حرارتی تعبیه شده است که دمای محیط را اندازه گیری کرده و به قسمت جبران ساز کارت اعمال می کند.

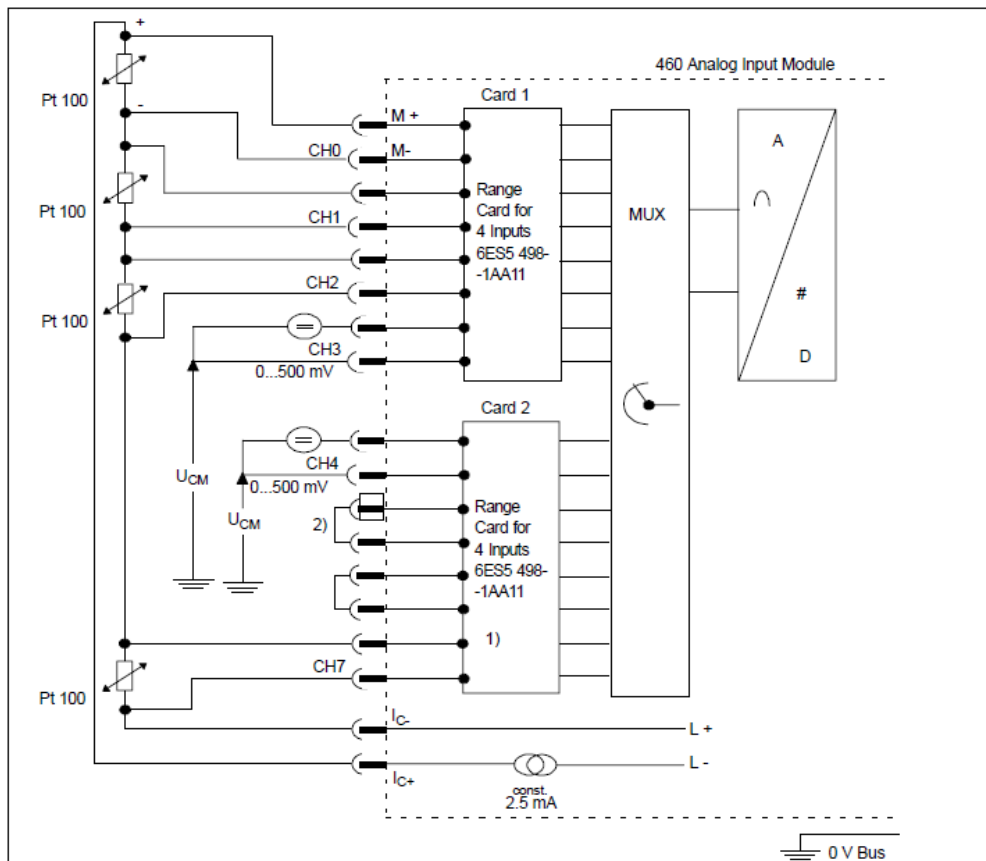


در شکل زیر نحوه اتصال یک ترموکوپل به کارت 460 به همراه نحوه اتصال جبران ساز بیرونی را مشاهده می کنید.



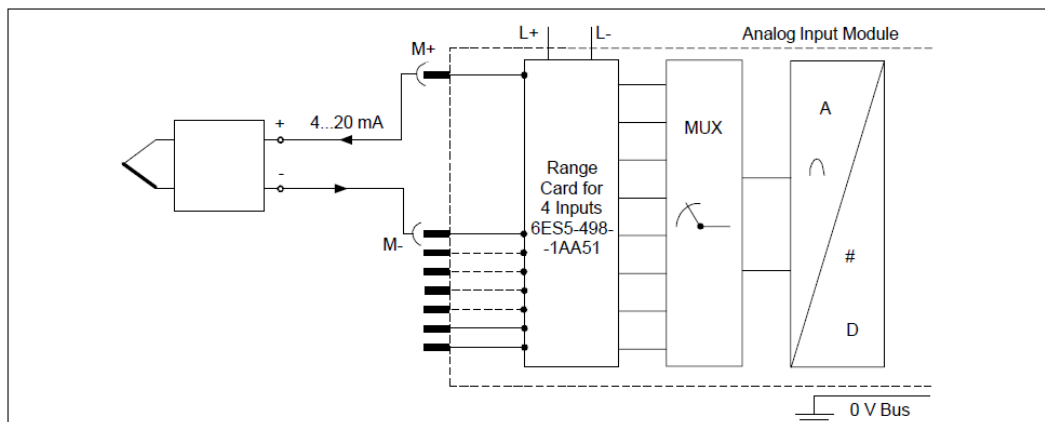
اتصال سنسورهای RTD به کارت 460

بر روی این کارت منبع جریان ثابت 2.5 میلی آمپری جهت عبور از سنسورهای PT100 تعبیه شده است. با عبور جریان از سنسور مقاومتی PT100، ولتاژ تولید شده توسط ترمینال های +M و -M اندازه گیری می شود. در این حالت بازه دمایی بین 200- تا 840 درجه سانتیگراد قابل اندازه گیری می باشد.



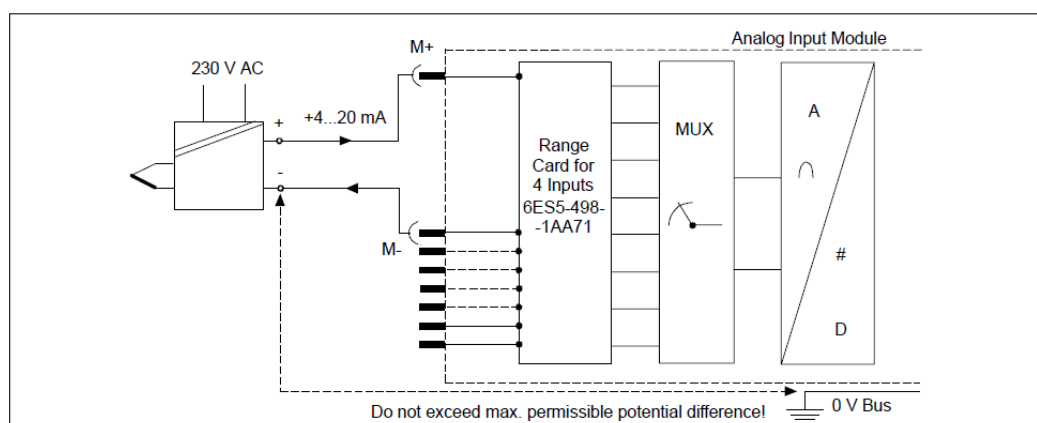
اتصال ترانسیمترهای جریانی دو سیمه

در شکل زیر نحوه اتصال ترانسیمتر جریانی دو سیمه را به کارت 460 ملاحظه می کنید.

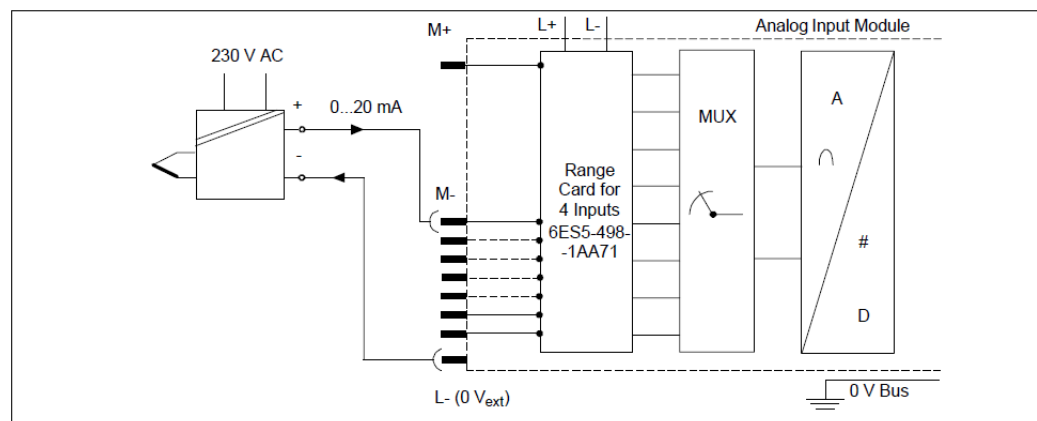


اتصال ترانسیمترهای جریانی چهار سیمه

در شکل زیر نحوه اتصال ترانسیمتر جریانی چهار سیمه را به کارت 460 ملاحظه می کنید.



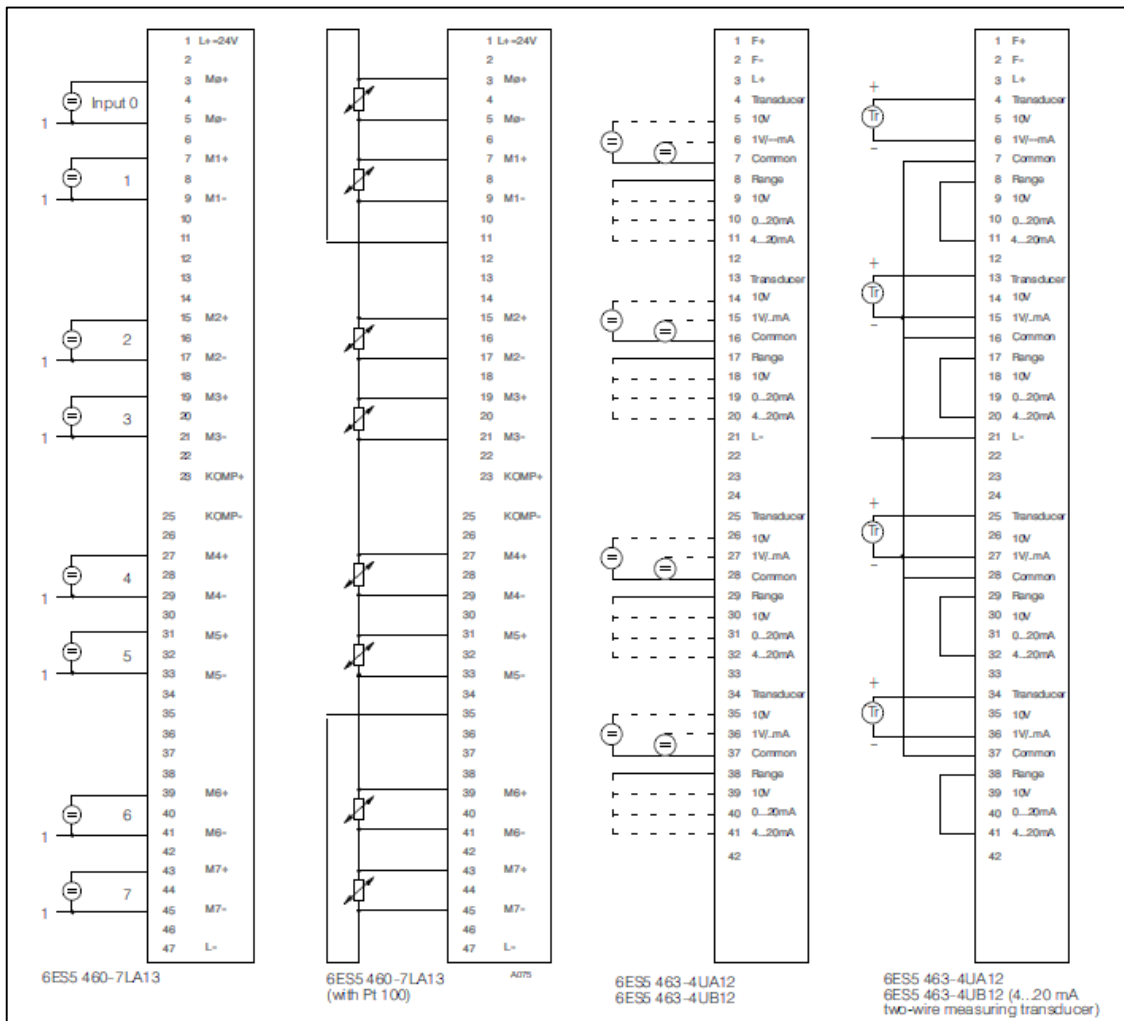
نحوه اتصال ترانسیمتر جریانی با خروجی 0 تا 20 میلی آمپر

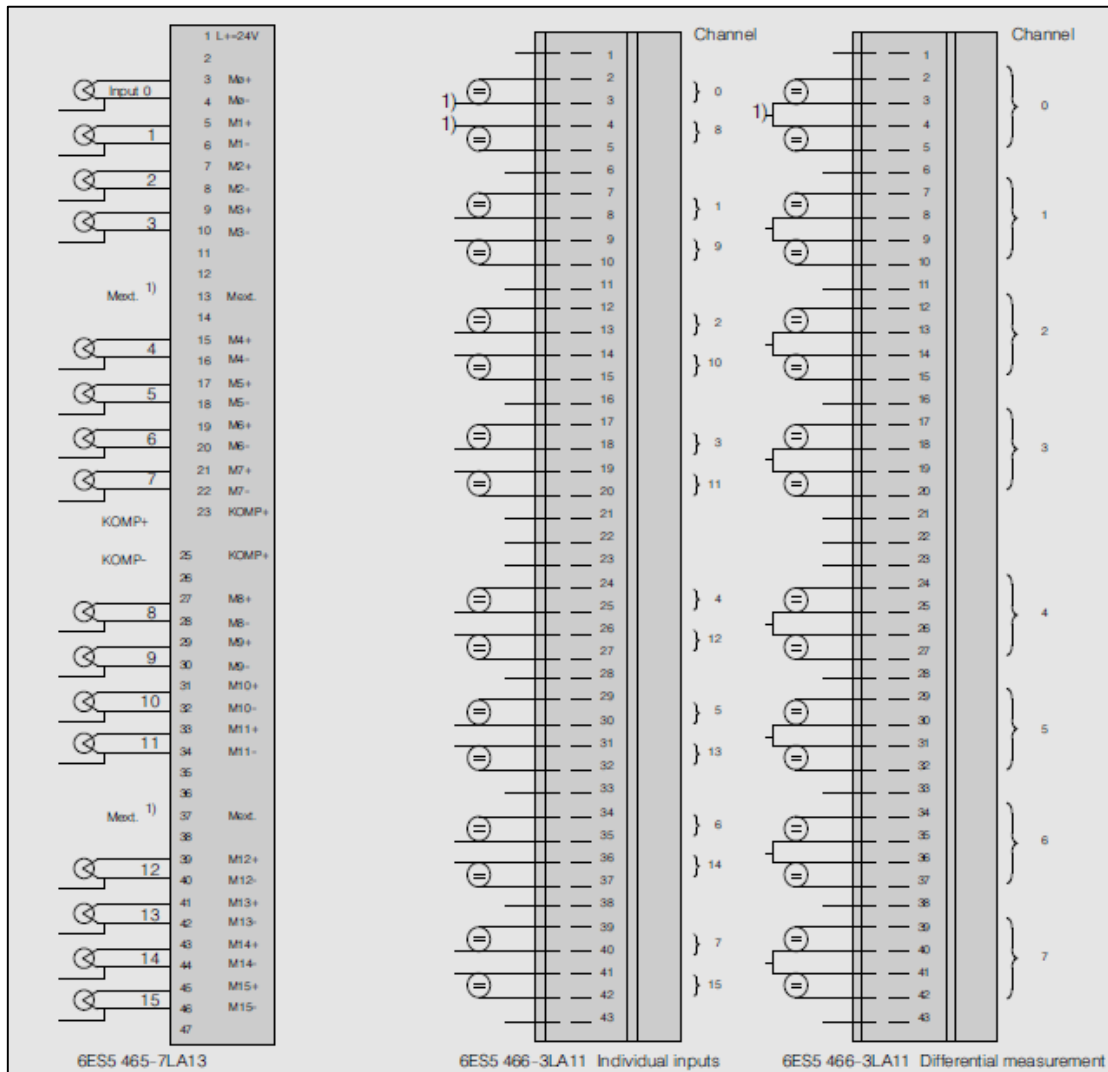


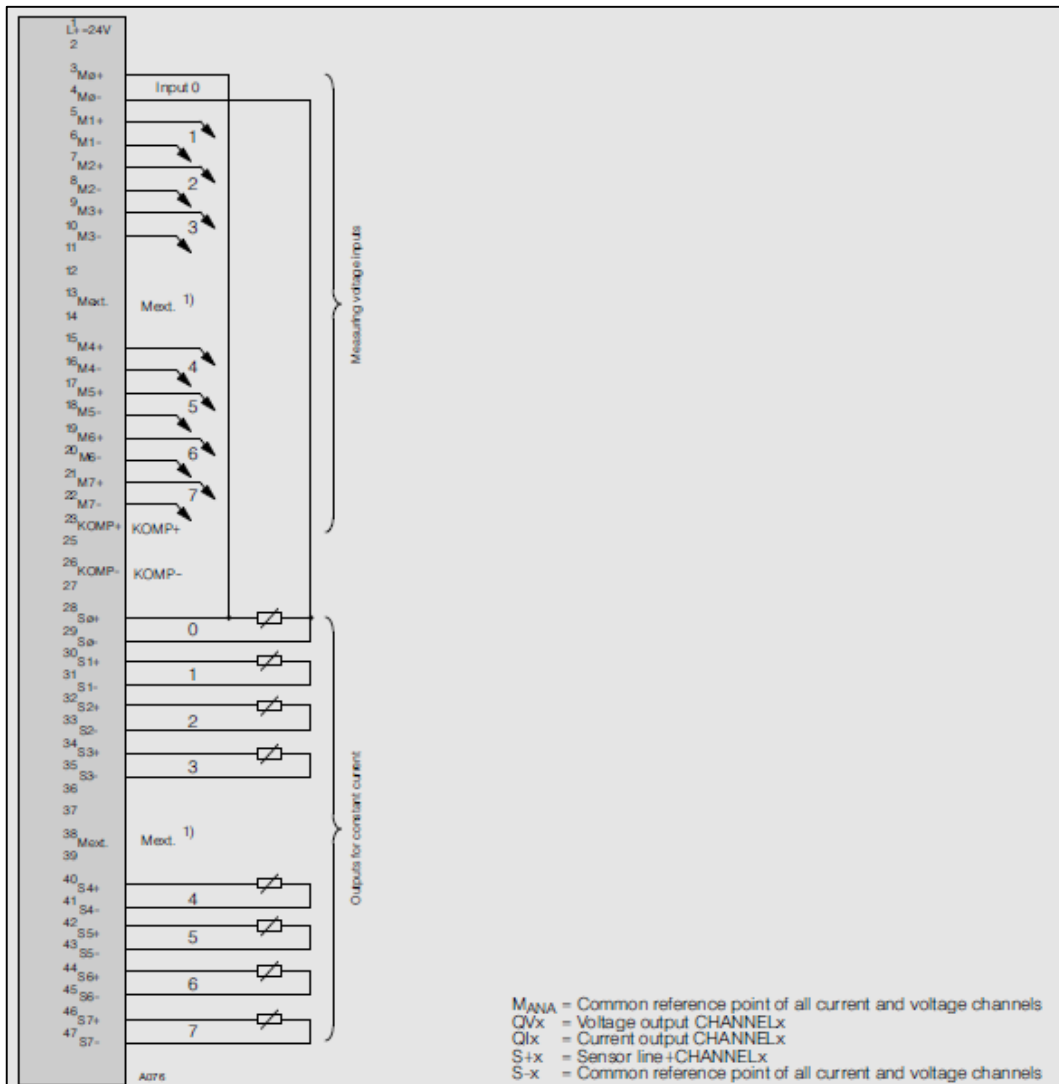
در ادامه به مشخصات کارت آنالوگ در سری 115U دقت نمایید.

در این سری ماژول ها می توانند دارای ۴، ۸ یا ۱۶ کانال با مشخصات زیر باشند:

Technical specifications			
Analog input module	6ES5 460-7LA13	6ES5 465-7LA13	6ES5 463-4U.12
Number of inputs	8 voltage/current inputs or 8 inputs for Pt 100 resistance thermometer	16 voltage/current inputs or 8 inputs for Pt 100 resistance thermometer	4 voltage/current inputs
Galvanic isolation	Yes	No	Yes
Input ranges (rated values)	± 50 mV; ± 500 mV; Pt 100; ± 1 V; ± 5 V; ± 10 V; ± 20 mA; + 4 ... 20 mA		0 ... 1 V, 0 ... 10 V, 0 ... 20 mA + 4 ... 20 mA for two-wire and four-wire measuring transducers
Input resistance in the individual ranges	50 mV: ≥ 10 MΩ 1 V: 90 kΩ; 2 % 500 mV: ≥ 10 MΩ 5 V: 50 kΩ; 2 % Pt 100 : ≥ 10 MΩ 10 V: 50 kΩ; 2 %		1 V: ≥ 10 MΩ; 10 V: 90 kΩ; 20 mA: 50 Ω 4 ... 20 mA: 62.5 Ω
Types of connection of sensors	Two-wire connection; four-wire connection for Pt 100		Two-wire connection
Digital representation of the input signal	12 bit + sign or 13 bit two's complement (2048 units = rated value)		11 bit two's complement (1024 units = rated value)







دقت در مبدل آنالوگ به دیجیتال

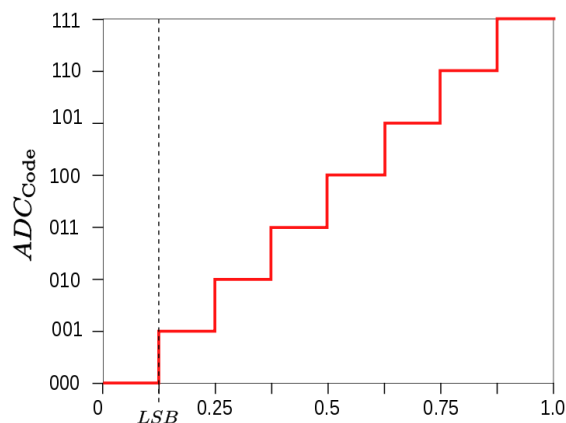
یکی از مشخصات مهم در انتخاب یک کارت ورودی آنالوگ چند بیتی بودن آن می باشد. معمولاً مبدل های ADC دارای قدرت تفکیک ۸ الی ۱۶ بیت می باشند. منظور از قدرت تفکیک یک کارت دقت آشکار سازی کوچکترین تغییر در سیگنال می باشد.

فرض کنیم به این کارت که دارای قدرت تفکیک ۱۱ بیت می باشد ولتاژ ۰ تا ۱۰ ولت اعمال می شود. در این صورت با تقسیم ولتاژ ۱۰ ولت بر ۲ به توان حد تفکیک کارت می توان حداقل تغییر قابل تشخیص توسط کارت را مشخص نمود.

همانطور که در شکل زیر مشاهده می کنید حداقل این سیگنال در حدود ۵ میلی ولت می باشد. این عدد بدین معنا می باشد که در این کارت تغییرات کمتر از ۵ میلی ولت ثبت نمی شوند.



اگر این موضوع را برای یک کارت با حد تفکیک ۱۶ بیت بررسی کنید ملاحظه می شود که عدد به دست آمده بسیار کوچکتر از ۵ میلی ولت می باشد. نتیجه این بحث این است که دقت در کارت هایی که دارای قدرت تفکیک بالاتر می باشند بیشتر است.



مقیاس گذاری یا SCALE چیست؟

زمانی که یک ورودی آنالوگ در PLC خوانده می شود، حاصل، یک عدد در بازه مشخصی می باشد. این عدد را نمی توان بصورت مستقیم در برنامه نویسی به کار برد. دلیل این مطلب این است که این عدد یک عدد خام بوده و هنوز معادل به واحد نشده است. در نتیجه قبل از شروع به برنامه نویسی می بایست عدد حاصل از دریافت کانال ورودی را معادل به واحد نمود که این عمل توسط برنامه نویس انجام می گیرد.

SCALE یک رابطه و یا یک معادله خط می باشد که توسط شرکت سازنده مشخص شده است. در اکثر مواقع این فرمول بصورت تابعی آماده در اختیار برنامه نویس قرار می گیرد. بازه نرمال در ورودی آنالوگ PLC مورد بحث به ازای سیگنال های مختلف، مطابق جدول شکل زیر می باشد. همچنین بازه هایی نیز به عنوان بازه های سرریز در نظر گرفته شده است.

Digital Measured-Value Representation as Value and Sign (rated input range ± 50 mV)

Units	Input Voltage in mV	Byte 0								Byte 1								
		7 S	6 2 ¹¹	5 2 ¹⁰	4 2 ⁹	3 2 ⁸	2 2 ⁷	1 2 ⁶	0 2 ⁵	7 2 ⁴	6 2 ³	5 2 ²	4 2 ¹	3 2 ⁰ A	2 E	1 OV	0	
≥ 4096	100.0	0	1	1	1	1	1	1	1	1	1	1	1	1	0/1	0/1	1	Overflow
4095	99.976	0	1	1	1	1	1	1	1	1	1	1	1	1	0/1	0/1	0	Overrange
2049	50.024	0	1	0	0	0	0	0	0	0	0	0	0	1	0/1	0/1	0	
2048	50.0	0	1	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0	Rated range
2047	49.976	0	0	1	1	1	1	1	1	1	1	1	1	1	0/1	0/1	0	
1024	25.0	0	0	1	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0	
1023	23.976	0	0	0	1	1	1	1	1	1	1	1	1	1	0/1	0/1	0	
1	0.024	0	0	0	0	0	0	0	0	0	0	0	0	1	0/1	0/1	0	
0	0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0	
-0	0.0	1	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0	
-1	-0.024	1	0	0	0	0	0	0	0	0	0	0	0	1	0/1	0/1	0	
-1023	-24.976	1	0	0	1	1	1	1	1	1	1	1	1	1	0/1	0/1	0	
-1024	-25.0	1	0	1	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0	
-2047	-49.976	1	0	1	1	1	1	1	1	1	1	1	1	1	0/1	0/1	0	
-2048	-50.0	1	1	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0	
-2049	-50.024	1	1	0	0	0	0	0	0	0	0	0	0	1	0/1	0/1	0	Overrange
-4095	-99.976	1	1	1	1	1	1	1	1	1	1	1	1	1	0/1	0/1	0	Overrange
-4096	-100.0	1	1	1	1	1	1	1	1	1	1	1	1	1	0/1	0/1	1	Overflow

Measured-Value Representation for Resistance Thermometers in the Standard Pt 100 Range The resolution with Pt 100 resistance thermometers is approximately 0.25 °C.
 1 ohm 10 units

Units	Resistance in ohm	Temp. in °C	Byte 0								Byte 1								
			7 S	6 2 ¹¹	5 2 ¹⁰	4 2 ⁹	3 2 ⁸	2 2 ⁷	1 2 ⁶	0 2 ⁵	7 2 ⁴	6 2 ³	5 2 ²	4 2 ¹	3 2 ⁰	2 A	1 E	0 OV	
4096	400.0	-	0	1	1	1	1	1	1	1	1	1	1	1	1	0/1	0/1	1	Overflow
4095	399.90	-	0	1	1	1	1	1	1	1	1	1	1	1	0/1	0/1	0	Overrange	
2049	200.98	269.1	0	1	0	0	0	0	0	0	0	0	0	1	0/1	0/1	0		
2048	200.0	266.5	0	1	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0		
2047	199.90	266.0	0	0	1	1	1	1	1	1	1	1	1	1	0/1	0/1	0		
1024	100.00	0	0	0	1	0	0	0	0	0	0	0	0	0	0/1	0/1	0	Rated range	
1023	99.90	-0.2	0	0	0	1	1	1	1	1	1	1	1	1	0/1	0/1	0		
1	0.098	-	0	0	0	0	0	0	0	0	0	0	0	1	0/1	0/1	0		
0	0.0	-	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0		

Measured-Value Representation in the Extended Pt 100 Measuring Range (Two's Complement)

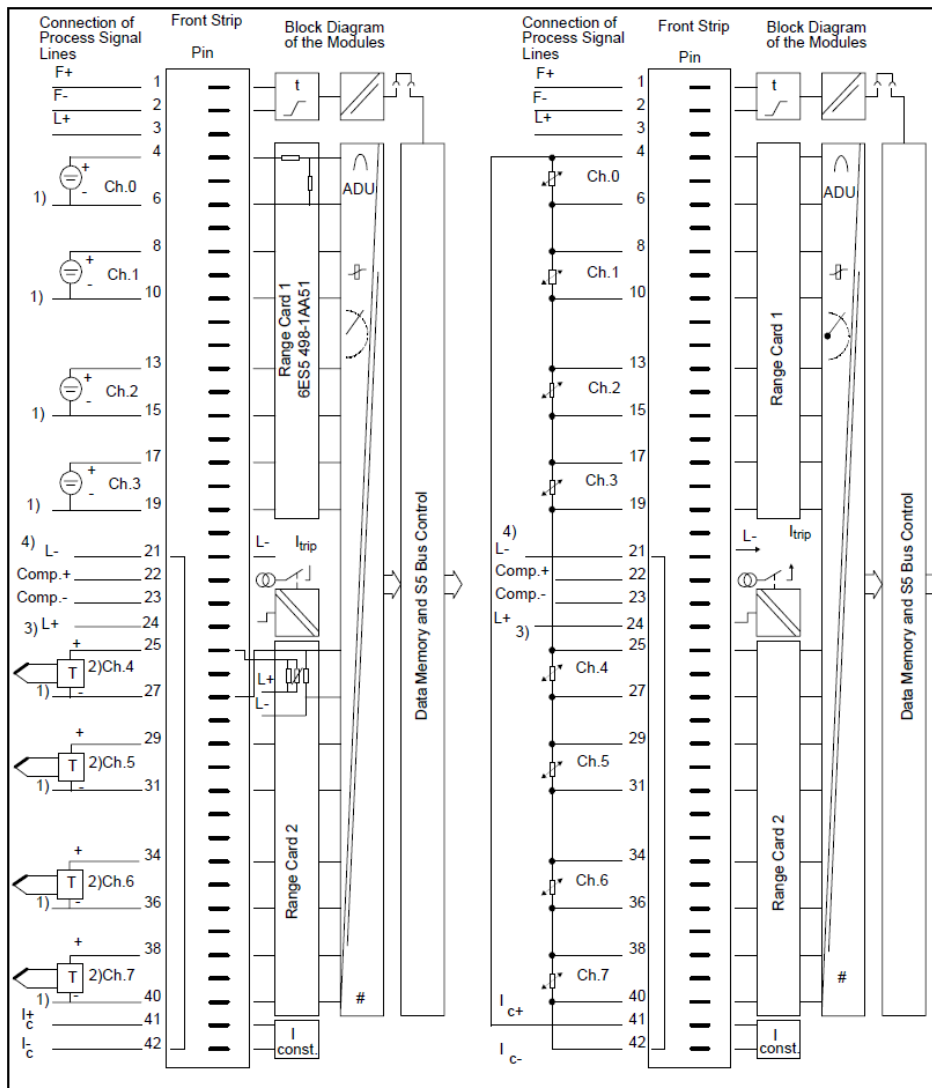
Units	Pt 100/ohm	Temp. in °C	Byte 0								Byte 1								
			7 S	6 2 ¹¹	5 2 ¹⁰	4 2 ⁹	3 2 ⁸	2 2 ⁷	1 2 ⁶	0 2 ⁵	7 2 ⁴	6 2 ³	5 2 ²	4 2 ¹	3 2 ⁰	2 A	1 E	0 OV	
>4095	≥ 140.0		0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	Overflow
4095	139.99	103.74	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange	
2049	120.01	51.61	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0		
2048	120.0	51.58	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0		
+1	100.01	0.026	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	Rated range	
0	100.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
-1	99.99	-0.026	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0		
-2048	80	-50.78	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		
-2049	79.99	-50.81	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	Overrange	
-4095	60.01	-100.60	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0		
<-4095	≤ 60		1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	Overflow	
Broken wire																			
-4095	Arbitrary	Arbitrary	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	Broken wire Ic+/Ic- ²⁾
-4095	Arbitrary	Arbitrary	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	Broken wire Sensor Measuring line ¹⁾	

ادامه

جدول تبدیلات سیگنال 4 تا 20 میلی آمپر

Units	Input Current in mA	Byte 0								Byte 1								
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
		S	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	A	E	OV	
≥ 4096	≥ 32.000	0	1	1	1	1	1	1	1	1	1	1	1	1	0/1	0	1	Overflow
4095	31.992	0	1	1	1	1	1	1	1	1	1	1	1	1	0/1	0	0	Overrange Short-circuit with two-wire transducer
3072	24.0	0	1	1	0	0	0	0	0	0	0	0	0	0	0/1	0	0	
3071	23.992	0	1	0	1	1	1	1	1	1	1	1	1	1	0/1	0	0	
2561	20.008	0	1	0	1	0	0	0	0	0	0	0	0	1	0/1	0	0	
2560	20.0	0	1	0	1	0	0	0	0	0	0	0	0	0	0/1	0	0	Rated range
2048	16.0	0	1	0	0	0	0	0	0	0	0	0	0	0	0/1	0	0	
512	4.0	0	0	0	1	0	0	0	0	0	0	0	0	0	0/1	0	0	
511	3.992	0	0	0	0	1	1	1	1	1	1	1	1	1	0/1	0	0	Range underflow
384	3.0	0	0	0	0	1	1	0	0	0	0	0	0	0	0/1	0	0	
383	2.992	0	0	0	0	1	0	1	1	1	1	1	1	1	0/1	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0	0	Broken wire

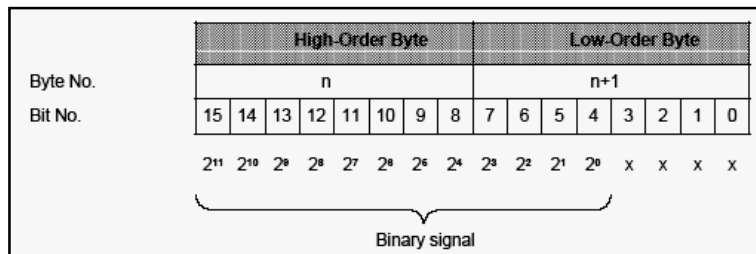
ترمینال های ماژول 460



تنوع کارت های آنالوگ در PLC های S5 همانند سایر PLC ها زیاد می باشد. در این قسمت جهت آشنایی، تنها یکی از ماژول ها مورد بررسی قرار گرفت. علاقمندان می توانند با مراجعه به کاتالوگ های زیمنس، در رابطه با مشخصات سایر ماژول ها، اطلاعات لازم را دریافت کنند.

نمایش مقدار دیجیتال یک سیگنال خروجی

جهت نمایش مقدار دیجیتال برای کارت های آنالوگ خروجی ، احتیاج به یک Word می باشد. ۴ بیت از این Word بلااستفاده می باشد. در شکل زیر بیت های مربوط به این Word نشان داده شده است.



خواندن مقادیر آنالوگ

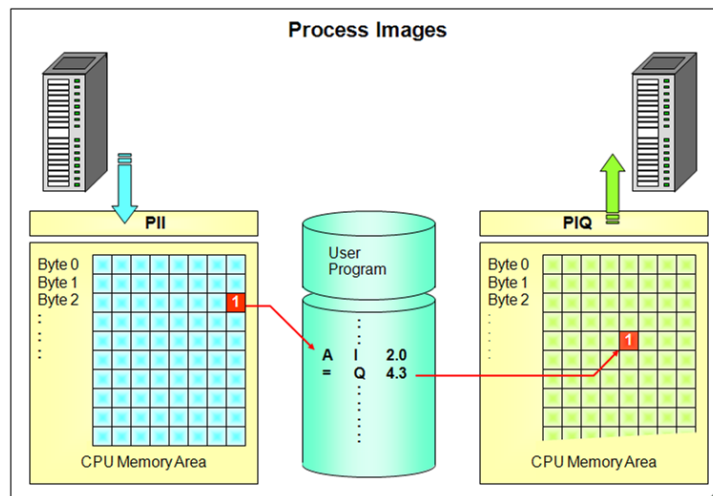
Possibilities for Reading in Analog Values	Onboard I/Os	External I/Os (Analog Input Modules)
Reading in analog values from the process image	After parameterizing DB1 accordingly: <ul style="list-style-type: none"> Using the L IW ... operation or Using FB250 	<ul style="list-style-type: none"> Using the L IW ... operation or Using FB250
Reading in analog values with direct I/O access	<ul style="list-style-type: none"> Using the L PW ... operation or Using FB250 	Not possible

همانطور که در جدول شکل فوق ملاحظه می کنید، برای خواندن کانال های ماژول ورودی آنالوگ، از عبارت PW یا IW استفاده می شود. برای خواندن مقدار آنالوگ به صورت مستقیم از کارت از PW و برای خواندن مقدار از تصویر PII ، از عبارت IW استفاده می کنیم.

آشنایی با تصاویر PII و PIO

بعد از Run کردن CPU در صورت وجود بلوک های Start Up ابتدا برنامه راه اندازی خوانده می شود. در ادامه مقدار ورودی ها از کارت خوانده می شود و در حافظه PII قرار می گیرد. PII مخفف Process Image Input می باشد و معرف بخشی از حافظه CPU است که ورودی ها در آن ذخیره می شوند. پس از قرارگیری مقدار لحظه ای متغیرها در PII، برنامه نوشته شده در CPU اجرا می شود. در ضمن اجرای برنامه، CPU هر جا که لازم باشد اطلاعات را از جدول PII می خواند. باید توجه نمود که برخی از ورودی های PLC در ناحیه PII ذخیره نمی شود. بلکه CPU در صورت نیاز آنها را به طور مستقیم از ماژول می خواند. به این ورودی ها به اصطلاح Peripheral Input می گویند.

به طور مشابه برخی از خروجی ها بدون اینکه از PIQ فرمان بگیرند به طور مستقیم در صورت لزوم از CPU مقدار می گیرند. به این خروجی ها اصطلاحاً Peripheral output می گویند.



تابع آماده در نرم افزار برای استفاده در سیگنالهای ورودی آنالوگ بلوک FB250 می باشد. در واقع توسط این بلوک می توان ورودی آنالوگ را به مقدار کمیت در بازه دلخواه تبدیل نمود.

Table 12-8. Calling FB250

STL	CSF
<pre> : JU FB 250 NAME : RLG.AE BG : KNKT : OGR : UGR : EINZ : XA : FB : BU : </pre>	<pre> FB 250 RLG.AE BG XA KNKT FB OGR BU UGR TBIT EINZ </pre>

پارامترهای FB 250

BG: آدرس مدول - فرمت از نوع KF

KNKT: شماره و نوع کانال - فرمت از نوع KY

OGR: حد بالای کمیت

UGR: حد پایین کمیت

EINZ: نمونه برداری انتخابی

XA: مقدار خروجی

FB: بیت مربوط به خطا

BU: بیت سرریز

TBIT: بیت فعالیت

Assignment	
External I/Os	Onboard I/Os
0 to 7	8
KY=x,y x=0 to 3 y=3 to 6 3: Absolute value repres. (4 to 20 mA) 4: Unipolar representation 5: Bipolar absolute value 6: Bipolar fixed-point no.	KY=x,y x=0 to 7 y=4 Unipolar representation

تنظیمات این ورودی از تابع برای ورودی/خروجی های که به صورت Onboard بر روی CPU قرار دارند، با ماژول های خارجی متفاوت می باشد.

*برای کانال تیپ ۳ (مقدار 4 تا 20 میلی آمپر)

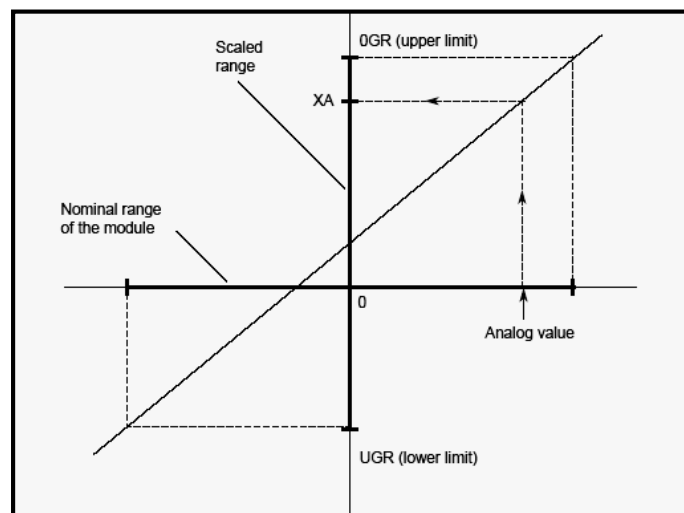
$$XA = \frac{UGR \cdot (2560 - xe) + OGR \cdot (xe - 512)}{2048}$$

*برای کانال تیپ ۴ (تک قطبی-مثبت)

$$XA = \frac{UGR \cdot (2048 - xe) + OGR \cdot xe}{2048}$$

*برای کانالهای تیپ ۵ و ۶ (دو قطبی-مثبت و منفی)

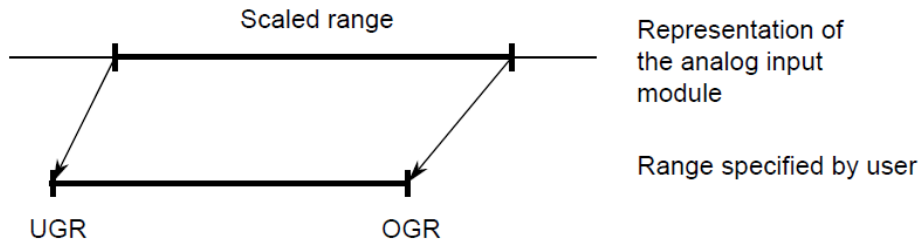
$$XA = \frac{UGR \cdot (2048 - xe) + OGR \cdot (xe + 2048)}{4096}$$



توسط ورودی های OGR و UGR نیز حدود کمیت مورد نظر که می بایست عملیات تبدیل مقدار خام ورودی به این بازه انجام شود نیز تعیین می شود.

OGR	Upper limit of the output value	D KF	- 32767 to +32767
UGR	Lower limit of the output value	D KF	- 32767 to +32767

Scaling Schematic



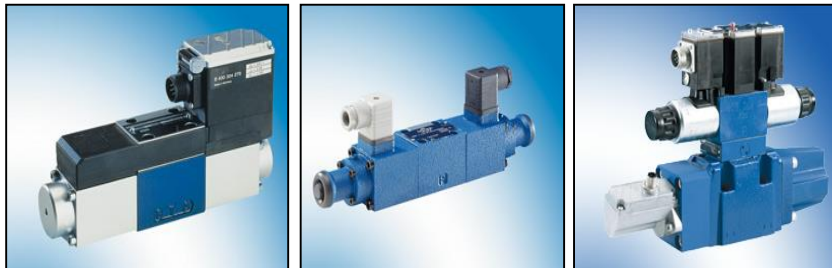
آشنایی با سیگنالهای خروجی آنالوگ

در صنعت امروز تجهیزاتی نیز تحت عنوان خروجی های آنالوگ شناخته می شوند. این مصرف کننده ها با توجه به مقدار سیگنال دریافتی می توانند تغییر وضعیت دهند. تغییر وضعیت در این تجهیزات به صورت دو حالت قطع و وصل نمی باشد. بلکه این تغییرات به صورت پیوسته و یا به عبارت دیگر آنالوگ می باشد. در ادامه جهت آشنایی هر چه بیشتر دوستان عزیز با چند نمونه پر کاربرد از این تجهیزات آشنا می شویم.

شیرهای Proportional

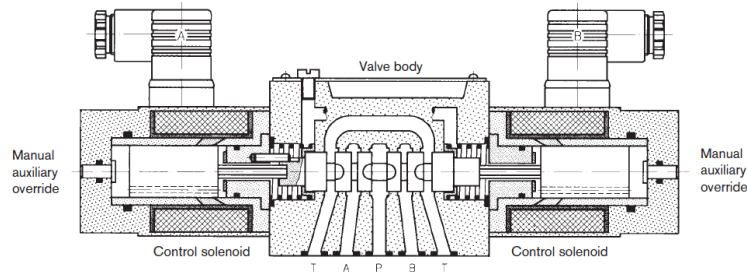
226

در بسیاری از پکیج های هیدرولیک شیرهای تحت عنوان شیرهای تدریجی وجود دارند. از این شیرها زمانی که بخواهیم مسیر عبور یک سیال همانند روغن را کنترل کنیم استفاده می گردد. به عنوان مثال فرض کنید در یک دستگاه پرس عملیات پرس کردن در فشارهای مختلف می بایست صورت گیرد. در این حالت می بایست حجم روغن تزریق شده پشت جک در فواصل زمانی افزایش و یا کاهش یابد. در واقع توسط این شیرها می توان مسیر عبور روغن را از 0 تا 100% کنترل نمود. برای کنترل این شیرها معمولا شرکت های سازنده بردهای الکترونیکی را وارد بازار می کنند که از این بردها به عنوان آمپلی فایر استفاده می شود. در این شیرها سیگنال الکتریکی آنالوگ به یک جابه جایی که متناسب با مقدار سیگنال ورودی است تبدیل می شود. بدین ترتیب می توان مقدار حرکت و یا گشودگی خروجی شیر را به طور دلخواه تنظیم نمود. این شیرها در انواع مختلف طراحی می شوند.



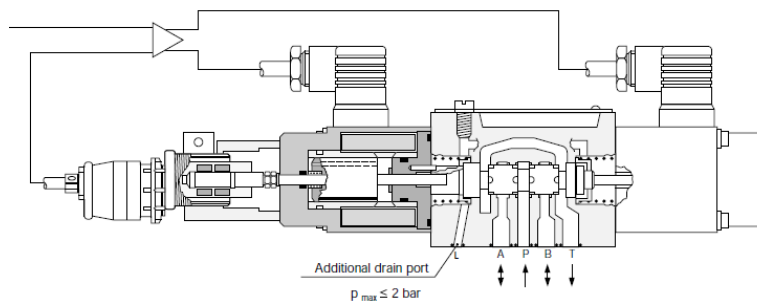
اما ساده ترین حالتی که می توان برای نحوه عملکرد آنها بیان نمود استفاده از یک اسپول داخلی می باشد که توسط آهنربا و مغناطیس ایجاد شده در بوبین جابه جا می شود. مقدار جابه جایی متناسب با سیگنال الکتریکی و همچنین شدت میدان مغناطیسی تولید شده می باشد. در حالت اولیه فرض کنید که هیچ جریانی از آهنربا نمی گذرد. در این حالت اسپول یا همان پیستون شیر توسط دو فنر در موقعیت وسط قرار داده می شود. با قرار گرفتن پیستون در وسط، دریچه کاملا بسته و در نتیجه مسیر عبور سیال بسته می باشد. با تزریق سیگنال الکتریکی به شیر، آهنربا تحریک شده و پیستون را در جهت مناسب حرکت می دهد. این جابه جایی پیستون متناسب با شدت میدان مغناطیسی تولید شده در بوبین می باشد. با جابه جایی پیستون در مسیر کاری، سیال با دبی حجمی Q از شیر عبور می کند.

در شکل زیر ساختار یک شیر پروپورشنال بدون کنترل موقعیت را مشاهده می کنید.



227

برخی از شیرهای پروپورشنال مجهز به سیگنال فیدبک توسط LVDT می باشند. در این شیرها موقعیت اسپول داخلی توسط LVDT تبدیل به یک سیگنال شده و به صورت یک سیگنال الکتریکی برای دریافت سیگنال فیدبک ظاهر می شود. در این حالت تنظیم موقعیت شیر به وسیله یک سیستم کنترل مدار بسته انجام می شود. این تقویت کننده بر روی شیر نصب می باشد. این شیرها نسل پیشرفته شیرهای پروپورشنال می باشند.

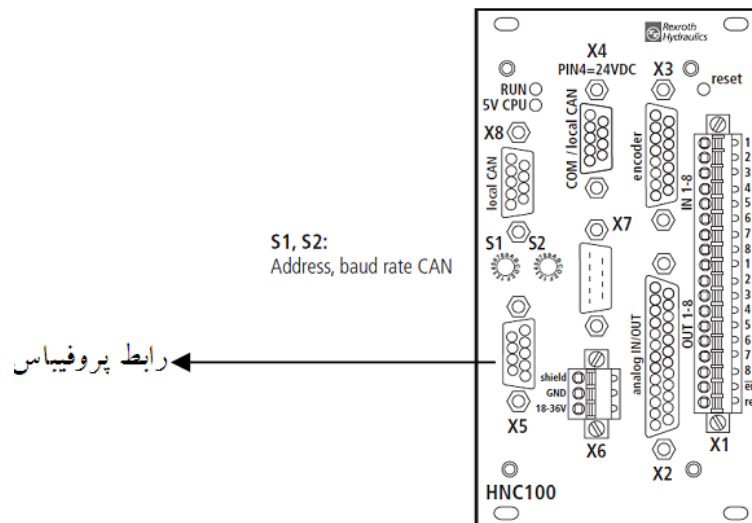


کارت کنترل الکترونیکی

همانطور که در قسمت قبلی بیان شد شیرهای پروپورشنال توسط بردهای الکترونیکی کنترل می شوند. این کارت ها در واقع به عنوان آمپلی فایر عمل می کنند. این بردها سیگنال کنترلی متغیر را برای کنترل شیر فراهم می کنند. جهت ساخت سیگنال های مورد نظر که به عنوان سیگنال های ورودی به برد تزریق می شوند معمولا از پتانسیومتر استفاده می شود. همچنین توسط این بردها می توان حساسیت و سایر تنظیمات از جمله تنظیم نقطه صفر را انجام داد. یکی از شرکتهای سازنده مطرح در امر تولید شیرهای پروپورشنال و همچنین بردهای کنترلی شرکت REXROTH می باشد. در شکل زیر چندین برد کنترلی برای کنترل شیر پروپورشنال را ملاحظه می کنید.



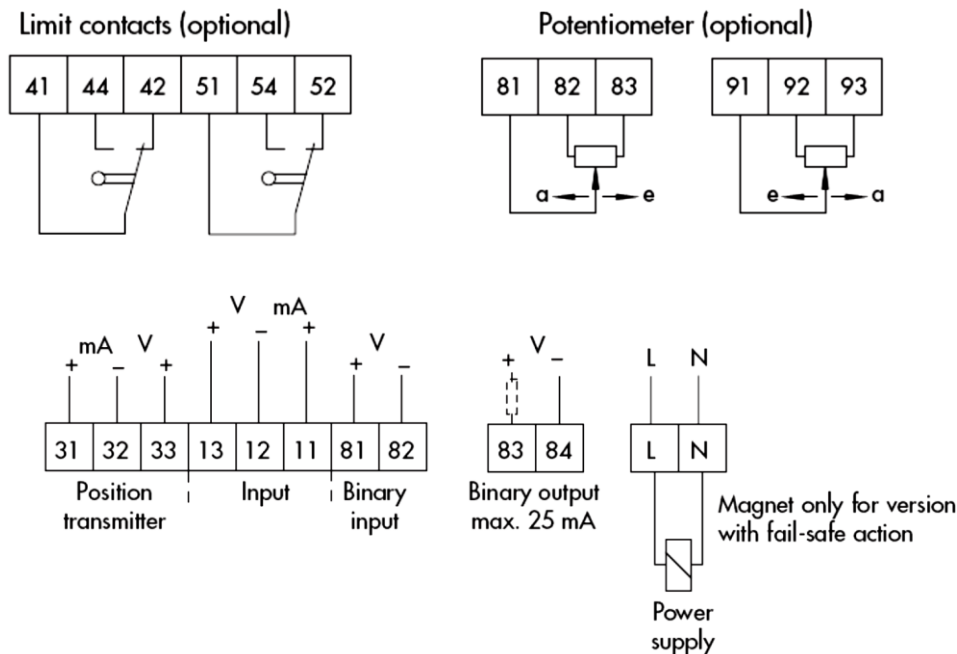
برخی از بردهای کنترلی پیشرفته مجهز به رابط PC می باشند. در واقع اختصاص پارامترها و تنظیمات این بردها می تواند توسط نرم افزار صورت گیرد. همچنین یکی دیگر از قابلیت‌های مهم این بردها قابلیت اتصال به شبکه پروفیباس می باشد.



در صنعت امروز تنوع ساخت شیرهای کنترلی بسیار زیاد می باشد. نوعی دیگر از شیرهای کنترلی شیرهایی می باشند که جهت کنترل فلو و فشار یک سیال همانند آب و یا بخار استفاده می شوند. به عنوان مثال فرض کنید که خروجی یک مخزن که محتوای آن یک سیال همانند اسید می باشد می بایست به صورت درصدی از سایر مایعات به یک مایع دیگر اضافه شود. در این صورت خروجی این مخزن، یک شیر کنترلی آنالوگ می باشد که قادر است مسیر سیال را بین 0 تا 100% کنترل کند. این شیرها معمولاً توسط سیگنال‌های آنالوگ جریانی (۴ الی ۲۰ میلی آمپر) کنترل می شوند.



این شیرها معمولا دارای دو لیمیت سوئیچ جهت تشخیص باز و یا بسته بودن شیر در هر جهت می باشند. در شکل زیر نحوه اتصالات را در یک نمونه از شیرهای شرکت SAMSON مشاهده می کنید.



آشنایی با درایوها

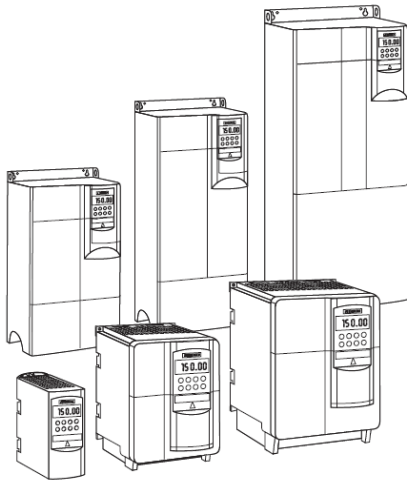
امروزه استفاده از اینورترها در مراکز صنعتی بیش از پیش ضرورت یافته است. از آنجایی که درایوها نیز می توانند به عنوان یک تجهیز در خروجی آنالوگ PLC به کار برده شوند، لازم است که اطلاعاتی کلی در رابطه با اجزای یک درایو و نحوه عملکرد آن در اختیار شما دوستان عزیز قرار گیرد. اینورترها همان کنترل دور موتورها می باشند که توسط آنها می توان بسیاری از پارامترهای موتور را تحت کنترل گرفت.

درایوهای میکرومستر زیمنس MICROMASTER

شرکت زیمنس طیف وسیعی از درایوهای AC را ارائه می دهد. در گذشته، جهت راه اندازی درایوها نیاز به کارشناسان متخصص بود. میکرومسترهای زیمنس درایوهایی را با قابلیت تنظیم خودکار، کنترل شار جریان، کنترل برداری و حلقه تنظیم PID ارائه می دهند. میکرومستر دارای یک پروسور برنامه ریزی شده می باشد. نصب و راه اندازی این درایوها بسیار ساده و سریع می باشد.

سری های میکرو مستر:

- MM 410
- MM 420
- MM 430
- MM 440



اجزای درایوهای میکرومستر

230

پانل های اپراتوری

میکرومسترها دارای ۲ پانل اپراتوری مبتدی و پیشرفته می باشند.



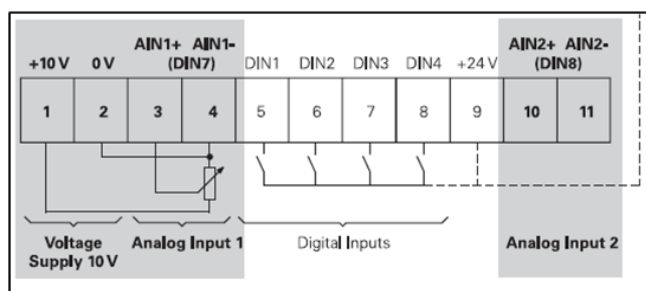
BOP

AOP

ورودی آنالوگ AI

درایوهای MM440 دارای دو ورودی آنالوگ (AIN1 و AIN2) PID می باشند. توسط این ورودی ها می توان توسط یک پتانسیومتر سرعت موتور را کنترل نمود. در کنترل به صورت PID کنترل سرعت بسیار مطلوب می باشد. به عنوان مثال در کنترل دما و فشار. سوئیچ های S1 و S2 جهت تعیین نوع سیگنال ورودی (جریان یا ولتاژ) استفاده می شوند. همچنین ورودی های آنالوگ می توانند به عنوان ورودی دیجیتال نیز استفاده شوند. البته تعداد ورودی ها و سایر قسمتهایی که در ادامه بحث می شود برای سایر مدل های میکرومستر متفاوت می باشد. این سیگنال می تواند توسط کارت خروجی آنالوگ PLC تامین شود.

در مثال زیر از ورودی آنالوگ AIN1 به عنوان ورودی آنالوگ مرجع جهت کنترل سرعت موتور بین 0 تا 100% استفاده می شود. ترمینال ۱ دارای ولتاژ داخلی 10VDC ثابت می باشد. ترمینال ۲ ترمینال برگشتی و یا زمین برای تغذیه ۱۰ ولت می باشد. ترمینال ۳ نیز به عنوان ترمینال مثبت ورودی آنالوگ می باشد. بین ترمینال ۲ و ۴ یک جامپر قرار داده شده است.

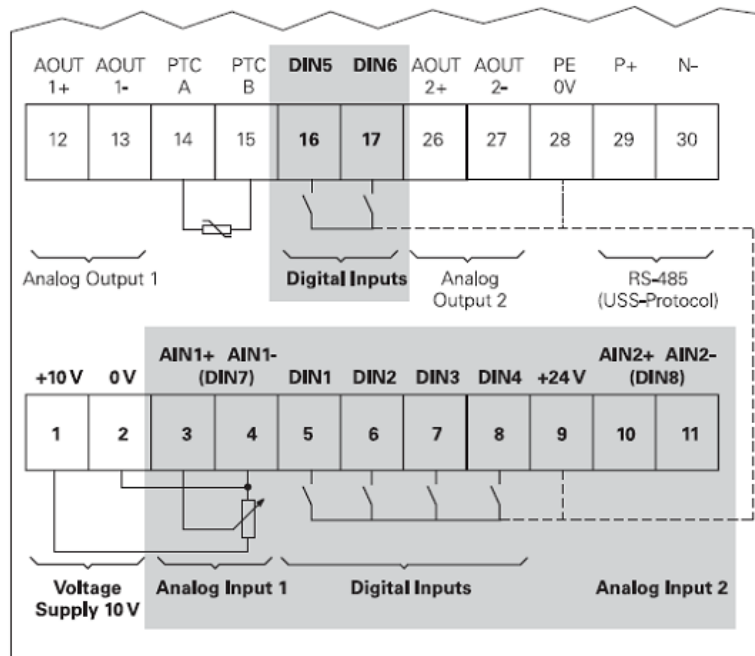


سیگنال ورودی هم می تواند در بازه 0 تا 10V و یا بین 0 تا 20 میلی آمپر باشد.

ورودی های دیجیتال DI

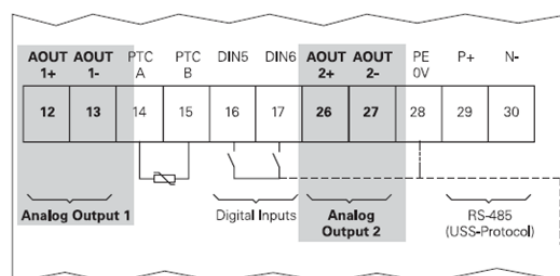
MM440 دارای ۶ ورودی دیجیتال می باشد. همچنین ورودی های آنالوگ نیز می توانند در حالت دیجیتال پیکربندی شوند. سوئیچ ها و کنتاکت ها می توانند بین ورودی 24VDC (ترمینال ۹) و ورودی های دیجیتال قرار گیرند.

استاندارد کارخانه ورودی DIN1 را به عنوان ورودی استارت/ استپ در نظر گرفته است. DIN2 به عنوان سوئیچ تغییر جهت و DIN3 به عنوان ورودی ری ست خطاها می باشد. سایر توابع مانند PRESET سرعت و JOG نیز می توانند برنامه ریزی شوند.



خروجی آنالوگ AO

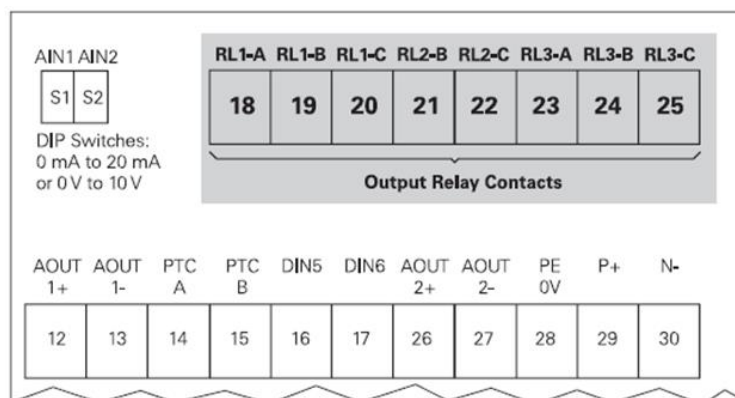
از خروجی آنالوگ جهت نظارت بر فرکانس، SetPoint فرکانس، ولتاژ خط DC، جریان موتور، گشتاور و دور موتور استفاده می شود. خروجی آنالوگ درایو می تواند به عنوان ورودی آنالوگ یک PLC باشد. در این صورت به راحتی می توان یک سیستم حلقه بسته طراحی نمود.



MM440 دارای دو خروجی AOUT1 و AOUT2 می باشد.

خروجی دیجیتال DO

در MM440 سه خروجی دیجیتال قابل برنامه ریزی وجود دارد. (RL1، RL2 و RL3). توسط خروجی های رله ای می توان شرایط مختلف کاری را از جمله فعال بودن درایو، خطاهای رخ داده برای درایو، فرکانس صفر یا نقطه مینیمم برای درایو را نشان داد.



نکته

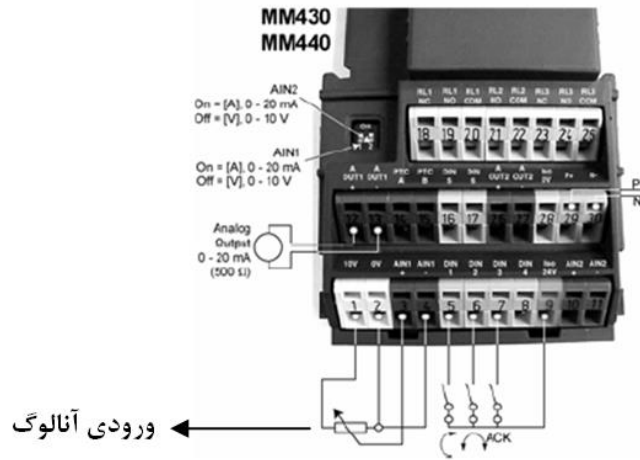
تعیین منبع فرمان و همچنین منبع تنظیم SetPoint برای درایو توسط پارامترهای درایو تعیین می شود. اگر منبع تنظیم SetPoint، ورودی های آنالوگ تعیین شود در این صورت توسط یک پتانسیومتر خارجی و یا کارت خروجی آنالوگ PLC می توان دور موتور را کنترل نمود.

یادآوری

ورودی آنالوگ در اینورترهای زیمنس

اکثر اینورترهای شرکت های مختلف دارای ورودی آنالوگ می باشند. توسط ورودی آنالوگ می توان یک ولتاژ 0 تا 10 ولت DC به اینورتر تزریق نمود که اینورتر نیز با دریافت این ولتاژ می تواند سرعت یک موتور را تغییر دهد.

این ولتاژ می تواند از یک پتانسیومتر و یا از کارت خروجی آنالوگ یک PLC ارسال شود



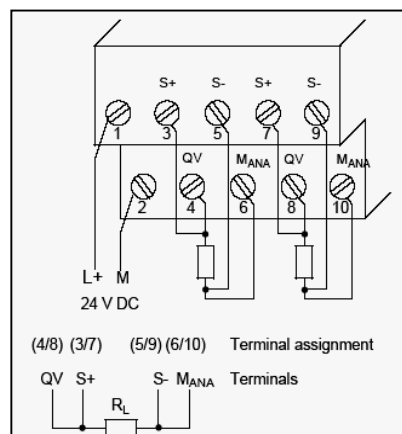
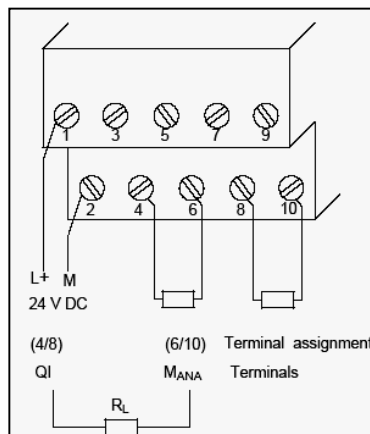
ماژول های خروجی آنالوگ در PLC

همانطور که می دانید یک PLC توسط ماژول های خروجی آنالوگ می تواند سیگنال های پیوسته ای را جهت کنترل تجهیزات آنالوگ تولید کند. این سیگنال می تواند به صورت جریان و یا ولتاژ باشد. در خروجی های آنالوگ، اعداد دیجیتال وارد مبدل DAC شده و سپس تبدیل به سیگنال آنالوگ می شوند. در قسمت زیر به برخی از سیگنال های پر کاربرد که توسط خروجی های آنالوگ می تواند تولید شود اشاره شده است.

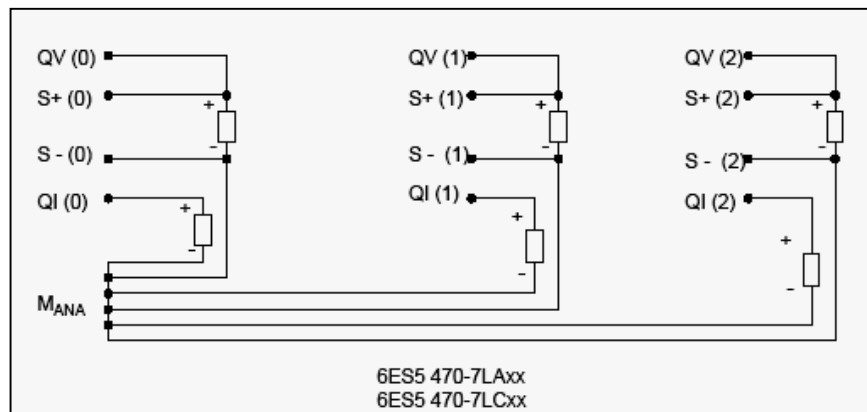
- 1..5V
- 0...10V
- 10V...+10V
- 4...20mA

در PLC های S5 شرکت زیمنس نیز در هر سری ماژول های متنوعی ارائه شده است. در این قسمت هدف آشنایی کلی با ماژول های خروجی آنالوگ می باشد.

اتصال مصرف کننده به کارت S5-100U



در شکل زیر نحوه اتصال بارها به خروجی های ولتاژ و جریان نشان داده شده است.



در شکل زیر یک ماژول AO را در PLC S5-115U مشاهده می کنید.

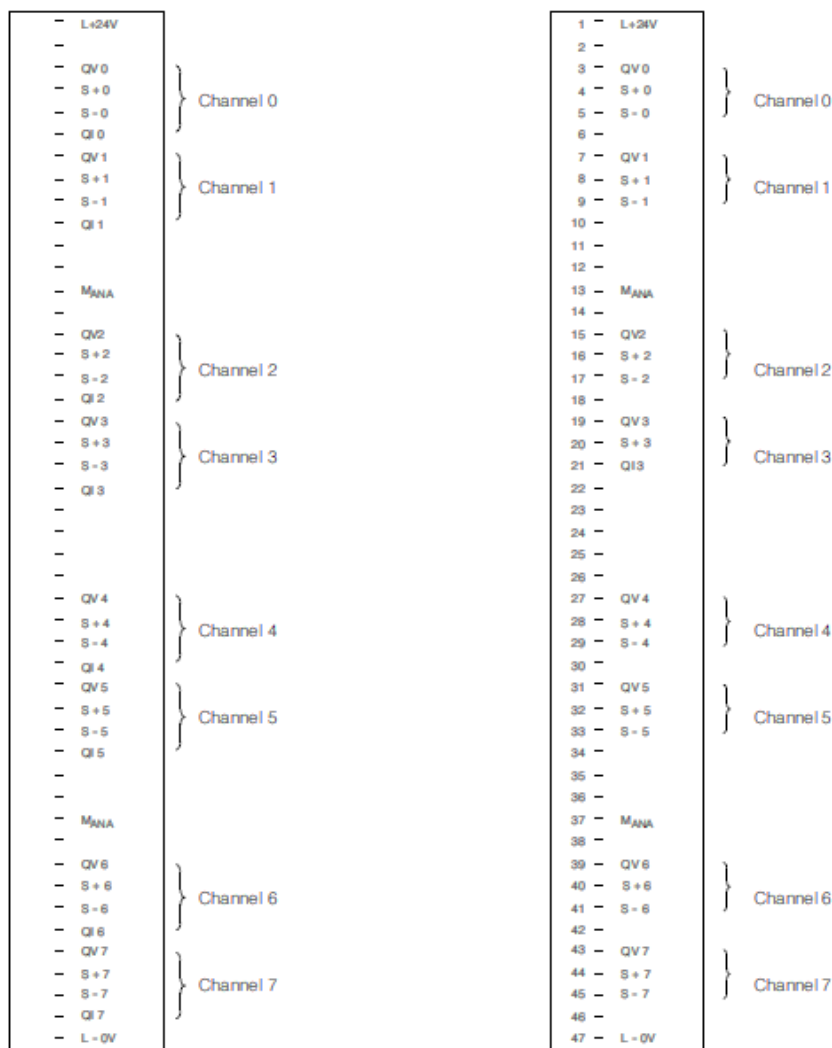
Analog output modules

Application



این کارت دارای ۸ کانال خروجی آنالوگ با دقت 12 Bit می باشد.

Technical specifications			
Analog output module		6ES5 470-7L.13	Analog output module
			6ES5 470-7L.13
Number of outputs (voltage and current outputs)		8	Open-circuit voltage (current output) approx. 18 V
Galvanic isolation		Yes (not between the outputs)	Voltage between the reference potential of the load (0 V connection) and the housing max. 60 V AC/75 V DC
Output ranges (rated values)		±10 V; 0...20 mA	Linearity in the rated range ± 2.5 %; ± 3 units
• 6ES5 470-7LA12		±10 V	Operational limits (0 to + 55 °C) ± 6 ‰
• 6ES5 470-7LB12		+1...5 V; +4...20 mA	Cable length (shielded) max. 200 m (650 ft)
• 6ES5 470-7LC12			Supply voltage
Load resistance		Only ohmic resistance	• Rated value + 24 V
• For voltage outputs min. 3.3 kΩ			• Ripple V_{pp} 3.6 V
• For current outputs max. 300 Ω			• Permissible range (including ripple) 20...30 V
Load connection		Load to the 0 V terminal	• Value at $t < 0.1$ s 36 V
Digital representation of the output signal		12 bit two's complement (1024 units = rated value)	Power consumption
Conversion time		1 ms	• Internal (at 5 V) typ. 0.25 A
Permiss. overload capability approx.		25 % (up to 1280 units)	• External (at 24 V) typ. 0.3 A
Short-circuit protection		Yes	Front connector 46-pin
Short-circuit current (voltage output) approx.		25 mA	Weight approx. 0.4 kg (0.9 lb)



در جدول شکل زیر مقادیر ولتاژ و جریان ، برای کارت آنالوگ خروجی 470 نشان داده شده است.

Table 11-15. Output Voltages and Currents for Analog Output Modules (Fixed-Point Number Bipolar)

Units	Output Values		High Byte	Low Byte	Range
	in V	in mA			
1280	12.5	25.0	0 1 0 1 0 0 0 0	0 0 0 0 x x x x	Overrange
1025	10.0098	20.0195	0 1 0 0 0 0 0 0	0 0 0 0 x x x x	
1024	10.0	20.0	0 1 0 0 0 0 0 0	0 0 0 0 x x x x	Nominal range
512	5.0	10.0	0 0 1 0 0 0 0 0	0 0 0 0 x x x x	
1	0.0098	0.0195	0 0 0 0 0 0 0 0	0 0 0 0 x x x x	
0	0.0	0.0	0 0 0 0 0 0 0 0	0 0 0 0 x x x x	
-1	-0.0098	-0.0195	1 1 1 1 1 1 1 1	1 1 1 1 x x x x	
-512	-5.0	-10.0	1 1 1 0 0 0 0 0	0 0 0 0 x x x x	
-1024	-10.0	-20.0	1 1 0 0 0 0 0 0	0 0 0 0 x x x x	
-1025	-10.0098	-20.0195	1 0 1 1 1 1 1 1	1 1 1 1 x x x x	Overrange
-1280	-12.5	-25.0	1 0 1 1 0 0 0 0	0 0 0 0 x x x x	

2x±10 V 6ES5 470-8MA12
2x±20 mA 6ES5 470-8MB12

Table 11-16. Output Voltages and Currents for Analog Output Modules (Unipolar)

Units	Output Values		High Byte	Low Byte	Range
	in V	in mA			
1280	6.0	24.0	0 1 0 1 0 0 0 0	0 0 0 0 x x x x	Overflow
1025	5.004	20.016	0 1 0 0 0 0 0 0	0 0 0 1 x x x x	
1024	5.0	20.0	0 1 0 0 0 0 0 0	0 0 0 0 x x x x	Nominal range
512	3.0	12.0	0 0 1 0 0 0 0 0	0 0 0 0 x x x x	
1	1.004	4.016	0 0 0 0 0 0 0 0	0 0 0 1 x x x x	
0	1.0	4.0	0 0 0 0 0 0 0 0	0 0 0 0 x x x x	
-1	0.996	3.984	1 1 1 1 1 1 1 1	1 1 1 1 x x x x	Overrange
-256	0.0	0.0	1 1 1 1 0 0 0 0	0 0 0 0 x x x x	
-512	-1.0	-4.0	1 1 1 0 0 0 0 0	0 0 0 0 x x x x	
-1024	-3.0	-12.0	1 1 0 0 0 0 0 0	0 0 0 0 x x x x	
-1280	-4.0	-16.0	1 0 1 1 0 0 0 0	0 0 0 0 x x x x	

2x 1 to 5 V 6ES5 470-8MD12
2x 4 to 20 mA 6ES5 470-8MC12

همانطور که در جداول فوق ملاحظه می کنید، در ماژول های خروجی آنالوگ بر خلاف ورودی آنالوگ، عدد فرستاده شده به کارت، تبدیل به یک سیگنال الکتریکی می شود. به عنوان مثال اگر عدد 1024 به کارت فرستاده شود، ولتاژ 10V و همچنین اگر عدد 0 فرستاده شود، کارت ولتاژ 0V را تولید می کند. البته لازم به ذکر است در صورتی که عدد منفی به کارت ارسال شود، کارت نیز سیگنال منفی تولید می کند. خروجی OUT همان آدرس کانال های کارت AO بوده که به صورت زیر آدرس دهی می شود.

T QWxxx

T PWxxx

مقدار خروجی آنالوگ در دو بایت نوشته می شود.

Table 12-12. General Representation of the Analog Output Value as Bit Pattern

Bit number	High-Order Byte								Low-Order Byte							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Analog value represent.	S	2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	X	X	X	X

Legend: X Irrelevant bits

تابع (FB251)Unscale

زمانی که از کارتهای خروجی آنالوگ جهت کنترل شیرهای تدریجی و یا کنترل سرعت موتور توسط اینورترها استفاده می شود، حتما نیاز می باشد که جهت برگرداندن ورودی Real به عدد صحیح جهت ارسال به کارت AO از تابع Unscale استفاده شود. عملکرد در کارتهای AO بدین صورت می باشد که کارت با دریافت عدد صحیح، آن را به توجه به حد بالا و حد پایین مشخص شده، تبدیل به سیگنال آنالوگ می کند. عدد ارسالی به کارت همانند آنچه که در مورد AI بیان شد می تواند بازه منفی را نیز شامل شود.

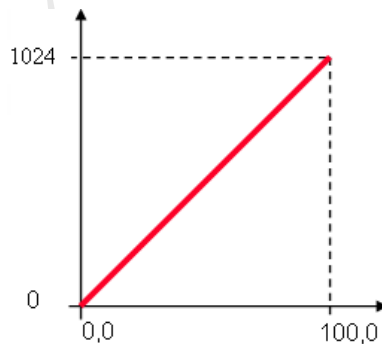


Table 12-14. Call of FB251

STL	CSF
<pre> : JU FB 251 NAME : RLG.AA XE : BG : KNKT : OGR : UGR : FEH : BU : </pre>	<pre> FB 251 RLG.AA XE FEH M BG BU M KNKT OGR UGR </pre>

آشنایی با FB251

Calling and Initializing -FB251-

Parameter	Meaning	Type	Data Type	Assignment	STL
XE	Analog value to be output	I	W	Input value (fixed-point) in the UGR to OGR range	: JU FB 251 NAME : RLG:AA XE : BG : KNKT : OGR : UGR : FEH : BU :
BG	Module address	D	KF	128 to 240	
KNKT	KN= channel number KT= channel type	D	KY	KY =x,y x =0 to 7 y =0;1 0: unipolar representation 1: bipolar fixed-point number	
OGR	Upper limit of the output value	D	KF	- 32768 to +32767	
UGR	Lower limit of the output value	D	KF	- 32768 to +32767	
FEH	Error when setting the limit value	Q	BI	"1" if UGR=OGR, for illegal channel or slot number, or illegal channel type	
BU	Analog value to be output exceeds UGR or OGR	Q	BI	At "1", "XE is outside the range (UGR;OGR). XE assumes the limit value	

238

XE: مقدار ورودی آنالوگ بر حسب واحد می باشد.

BG: آدرس اسلات

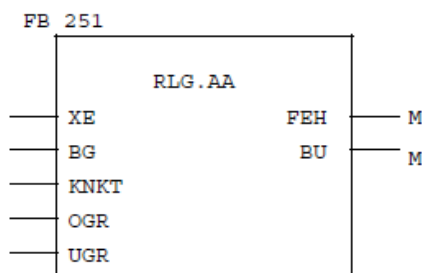
KNKT: آدرس کانال و تایپ آن

OGR: حد بالای واحد

UGR: حد پایین واحد

FEH: خطا در تنظیمات

BU: افزایش مقدار ورودی از حدود OGR و UGR



رابطه USCALE برای دو حالت تک قطبی و دو قطبی به صورت زیر می باشد.

For channel type 0 (unipolar representation):

$$xa = \frac{1024 \cdot (XE - UGR)}{OGR - UGR}$$

For channel type 1 (bipolar representation):

$$xa = \frac{1024 \cdot (2 \cdot XE - OGR - UGR)}{OGR - UGR}$$

XE: مقدار دیجیتال یا همان ورودی تابع می باشد.

XA: مقدار خام خروجی می باشد.

مثال ۱

در این مثال فرض کنید از یک سنسور جهت اندازه گیری دما استفاده شده است. این سنسور به یک ترانسمیتور متصل شده است که دارای خروجی 0 تا 10 ولت می باشد. بازه ای که ترانسمیتور بر روی آن کالیبره شده است، 0 تا 280 درجه می باشد. می خواهیم توسط PLC S5 مقدار این دما را بر روی یک پانل صنعتی نمایش دهیم.



239

برنامه

در این برنامه مقدار دما در حافظه FW130 ذخیره می شود

STL	
	: JU FB 250
NAME	: RLG:AE
BG	: KF=+8
KNKT	KY=0,4
OGR	: KF=+2800
UGR	: KF=+0
EINZ	: F 50.0
XA	: FW 130
FB	: F 120.0
BU	: F 121.0

مثال: پردازش سیگنال آنالوگ توسط S5-115U

در این مثال قصد داریم که سطح یک مخزن در بسته را که حاوی مایع می باشد، بر روی یک نمایشگر نمایش دهیم و زمانی که سطح تانک به مقدار مشخصی رسید، یک بیت از فلگ فعال شود.

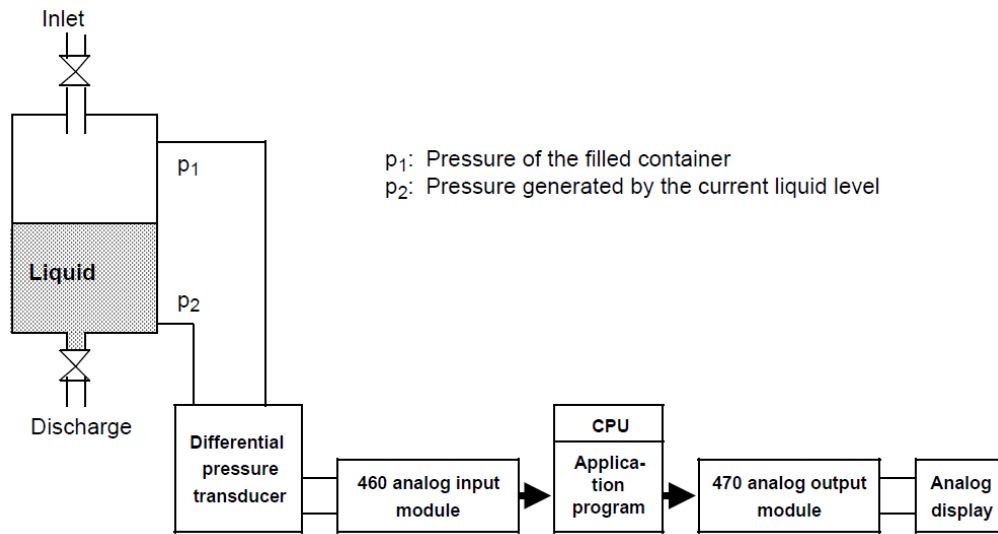
شرح وظایف:

ترانسمیتور: کنترل سطح تانک بین 0 تا 10 متر

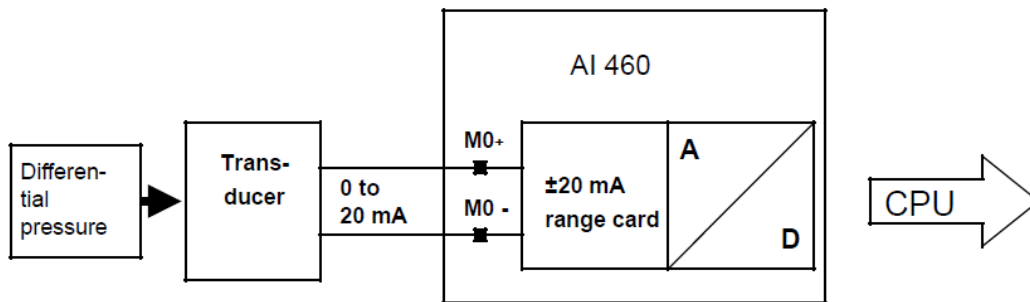
کارت AI: دریافت سیگنال از مبدل و تبدیل آن به مقدار دیجیتال جهت پردازش

کارت AO: تبدیل مقدار دیجیتال به آنالوگ و ارسال آن به کارت AO جهت نمایش بر روی ولتمتر

شکل کلی پروسه

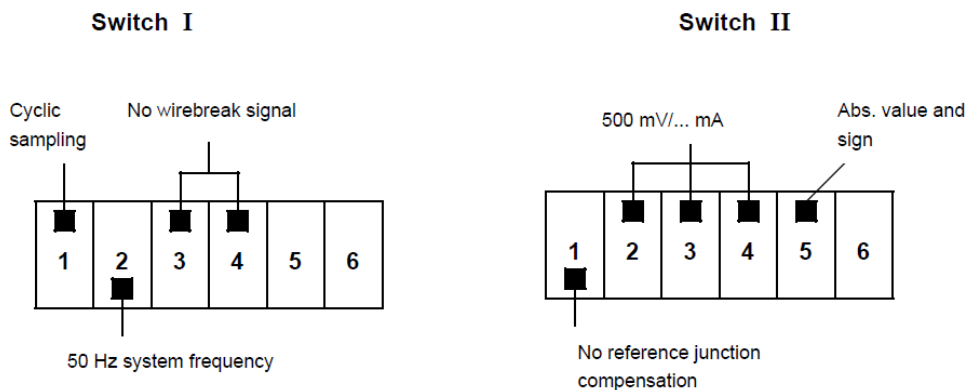


ترانسمیتور جریان را می توان بصورت مستقیم به کارت آنالوگ 460 متصل نمود. (ترمینالهای - MO, MO+)



جریان 0 میلی آمپر برابر ارتفاع 0 متر، و جریان 20 میلی آمپر برابر ارتفاع 10 متر می باشد. در خروجی کارت AI، سیگنال ورودی به یک عدد دیجیتال 0 تا 2048 تبدیل می شود که در برنامه کاربر مورد استفاده قرار می گیرد.

تنظیمات مربوط به سوئیچهای کارت آنالوگ



سوئیچ I

- نمونه برداری سیکلی
- فرکانس سیستم ۵۰ هرتز
- بدون سیگنال قطع سیم

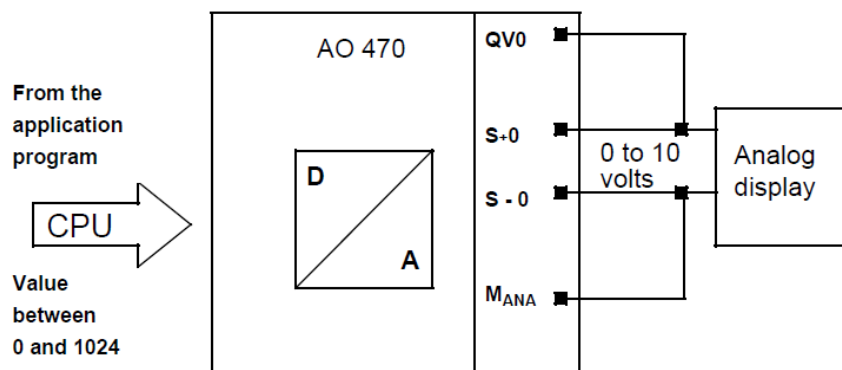
سوئیچ II

- بدون اتصال مرجع جبران ساز
- جریان - در محدوده میلی آمپر
- مقدار مطلق و با علامت

241

ماژول خروجی آنالوگ 470

نمایشگر ولتاژ نیز بصورت مستقیم به کارت AO متصل می شود. (پینهای M, S-0, S+0, QV0) کارت آنالوگ خروجی باید مقدار دیجیتال را به یک ولتاژ، بین 0 تا 10 ولت جهت نمایش روی نمایشگر تبدیل کند. در شکل زیر نحوه اتصال نمایشگر ولتاژ به کارت AO نشان داده شده است.



برنامه بلوک FB250

PB1 STL	Description
:JU FB 250 NAME :RLG:AE BG :KF +128 KNKT :KY 0,4 OGR :KF +1000 UGR :KF +0 EINZ :F 12.0 XA :FW 10 FB :F 12.1 BU :F 12.2 TBIT :F 12.3	MODULE STARTING ADDR: 128 (WHEN SLOT ADDRESSING IS FIXED: SLOT 0) CHANNEL NO.: 0; UNIPOLAR REPRESENTATION: 4 PHYSICAL MEASURING RANGE: 0<XA<1000CM RELEVANT ONLY FOR SELECTIVE SAMPLING (SET IN EXAMPLE FOR: CYCLIC SAMPLING) IN FW 10: XA VALUE 0<XA<1000CM RELEVANT ONLY WHEN SET FOR WIREBREAK SIGNAL IF LEVEL > 1000CM, BU = 1. RELEVANT ONLY FOR SELECTIVE SAMPLING.

242

با اجرای بلوک FB250، سیگنال دریافت شده از ترانسمیتر، به بازه 0 تا 1000 سانتیمتر تبدیل و در FW10 ذخیره می شود.

برنامه بلوک FB251

PB1 STL (cont.)	Description
:JU PB 9 :JU FB 251 NAME :RLG:AA XE :FW 10 BG :KF +160 KNKT :KY 0,0 OGR :KF +1000 UGR :KF +0 FEH :F 12.4 BU :F 12.5 :BE	GENERATE LIMITING VALUE OUTPUT ANALOG VALUE XA (FB 250) = XE (FB 251) MODULE STARTING ADDR.; 160 (FIXED SLOT ADDRESSING: SLOT 1) CHANNEL NO.: 0; UNIPOLAR REPRESENTATION: 0 PHYSICAL MEASURING RANGE: 0<XA<1000CM WHEN UGR = OGR, FEH = 1 WHEN XA<UGR OR XA>OGR, BU = 1.

با اجرای بلوک FB251، مقدار ورودی XE در بازه 0 تا 1000 سانتی متر، تبدیل به مقدار خام شده و مقدار خام تولید شده نیز روی کارت به صورت ولتاژ 0 تا 10 ولت ظاهر می شود.

PB9 STL	Description
:L KF +900 :L FW 10 :<=F :=F 12.6 :BE	MAX. VAL. FOR LIQUID LEVEL MEASURED VALUE MEASURED VALUE > 900 ? IF YES, F 12.6 = INITIATE REACTION IN SAME PROGRAM CYCLE.

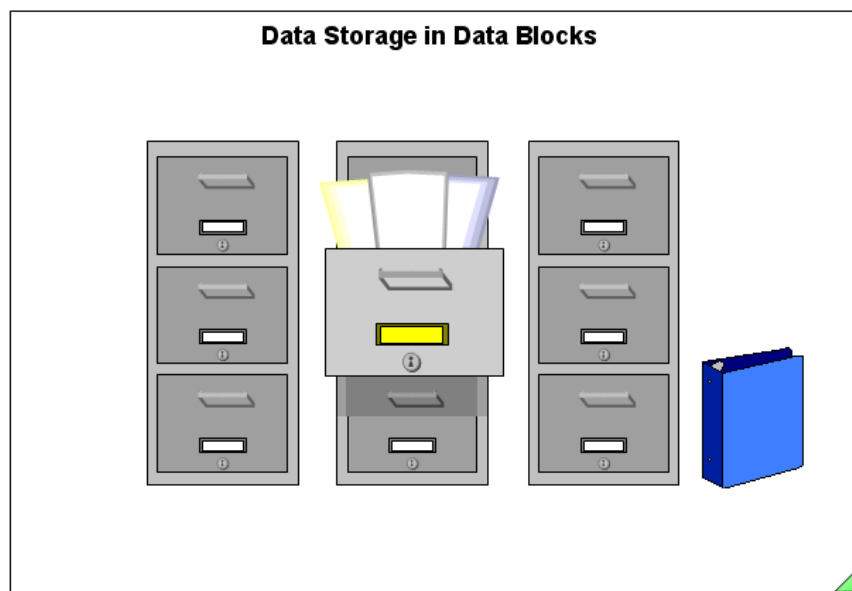
زمانی که مقدار سطح بیشتر از ۹ متر شود، فلگ F12.6 فعال می شود.

بلوکهای اطلاعاتی DB

یکی از موارد مشترک بین تمامی نرم افزارهای مربوط به PLC های زیمنس، وجود بلوکی با نام DB جهت استفاده در برنامه های کنترلی می باشد. DB ها محل نوشتن برنامه کنترلی نمی باشند، بلکه اطلاعات و مقادیر پروسه می تواند در این بلوک ها وارد و یا از آن خارج شود. در PLC های سری 400 , S7-300 بلوک های DB بسیار گسترده و پرکاربرد می باشند. یکی از کاربردهای مهم بلوک های اطلاعاتی در سیستم های مانیتورینگ می باشد. در بسیاری از برنامه های صنعتی جهت ذخیره اعداد مربوط به S.P ها و پارامترهای فرآیند از این بلوک ها استفاده می شود.

در PLC S5 بلوک DB1 به عنوان بلوک سیستمی در نظر گرفته شده است. این بلوک ها در حافظه داخلی برنامه استفاده نمی شوند.

243

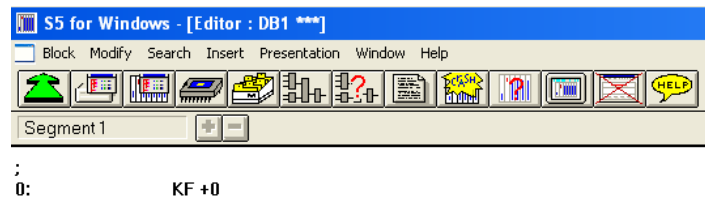


بلوک های DB در S5 می تواند دارای سطرهای متعددی باشد که هر سطر یک DW نامیده می شود. هر DW نیز می تواند دارای مقادیر با تایپ های مختلف باشد.

مثال ۱

فرض کنید از یک سنسور جهت شمارش تعداد بسته های تولید شده بر روی یک خط تولید استفاده می شود. می خواهیم تعداد بسته های تولید شده را در فضای حافظه DB2 ذخیره نماییم. همانطور که در قسمتهای قبلی بیان شد با قطع و وصل برق PLC محتوای DB ها از بین نمی رود.

ساخت DB و ایجاد یک سطر با شماره DW0



این سطر دارای مقدار اولیه 0 می باشد.

برنامه OB1

244

```

C      DB 2
A      I 0.0
CU     C 0
L      C 0
T      DW 0
BE

```

در بلوک OB1 ابتدا توسط DB 2، بلوک DB2 باز می شود. در ادامه برنامه شمارنده نوشته شده است. مقدار شمارنده به سطر اول DB2 منتقل شده است.

مثال ۱

در یک مثال فرض کنید در بلوک اطلاعاتی DB 2 دو عدد ۱۰۰۰ و ۱۵۰۰ قرار دارد. می خواهیم برنامه ای در OB 1 طراحی کنیم به گونه ای که حاصل جمع این دو عدد در FW 0 منتقل شود.

```

;
0:      KF +1000
1:      KF +1500

```

برنامه OB1

```

C      DB 2
L      DW 0
L      DW 1
+F
T      FW 0
BE

```

در ابتدا حتما باید کل بلوک DB 2 فراخوانی شود. عملیات فراخوانی توسط دستور C انجام می شود. بعد از اینکه بلوک DB 2 توسط دستور C باز شد، حال می توانیم به خطوط DB دسترسی پیدا کنیم که در این مثال هر دو خط DB 2 فراخوانی و پس از جمع کردن این دو خط نتیجه به FW 0 منتقل شده است.

خواندن اطلاعات از DB بدون شرط انجام می پذیرد. هنگامی که در اجرای یک برنامه ، DB خاصی فعال یا معتبر می گردد، برنامه با توجه به اطلاعات موجود در آن DB اجرا می شود و این عمل تا زمانی که DB دیگری معتبر نگردیده، انجام می گیرد. DB ها می توانند در تمامی بلوک ها فراخوانی شوند.

مثال ۳

می خواهیم توسط DB 2 یک شمارنده با سرعت Cycle Time در FW 0 ایجاد نماییم.

DB 2

```
;
0:      KF +0
1:      KF +1
```

برنامه OB1

```
C      DB 2
L      DW 0
L      DW 1
+F
T      DW 0
T      FW 0
BE
```

در برنامه OB 1 ، با سرعت سیکل برنامه مراحل زیر تکرار می شود:

در سیکل اول مقدار ۰ با ۱ جمع و نتیجه مجددا در خط اول ریخته می شود.
در سیکل دوم محتوای خط اول یعنی عدد ۱ با خط دوم (عدد ۱) مجددا جمع و این عمل در سیکل های بعدی با سرعت سیکل برنامه تکرار می شود.
پس اگر کمی دقت کنید در می یابید که در خط اول DB 2 یک شمارنده با سرعت سیکل اسکن ایجاد شده است.

مثال ۴

تولید پالس در خروجی Q0.0 توسط کاربرد DB

در این مثال می خواهیم از تاخیر ایجاد شده در DB یک پالس مربعی تولید نماییم. زمان این پالس ها تقریبی بوده و در کاربردهای عملی باید در ابتدا توسط برنامه نویس محاسبه شود.

DB 2

```
;
0:      KF +0
1:      KF +1
2:      KF +100
3:      KF +200
```

برنامه OB1

C	DB 2
L	DW 0
L	DW 1
+F	
T	DW 0
T	FW 0
L	DW 0
L	DW 2
!=F	
S	Q 0.0
L	DW 0
L	DW 3
!=F	
R	Q 0.0
JC	=TEXT
BEU	
TEXT:	
L	KF +0
T	DW 0
BE	

246

در این مثال زمانی که مقدار شمارنده موجود در DW 0 به عدد ۱۰۰ برسد، خروجی Q0.0 فعال و زمانی که مقدار DW 0 برابر ۲۰۰ شود، خروجی Q0.0 غیرفعال می شود. در پایان نیز برای اینکه پالس خروجی تداوم داشته باشد، عدد صفر به DW0 منتقل می شود تا مجدداً DW 0 از عدد صفر شروع به تغییر کند.

بلوکهای تابع ساز FB

همانطور که در بحث های قبلی بیان شد، یکی از روش های برنامه نویسی، روش ساختار یافته می باشد که دارای مزیت های زیادی می باشد:

- ۱- برنامه نویسی سمبلیک
- ۲- ایجاد توابع دلخواه
- ۳- جلوگیری از تکرار برنامه
- ۴- محیط بسیار جالب برای عیب یابی

نکته: برنامه نویسی در بلوکهای FB فقط به روش STL بوده و دستورات این بلوک نیز دستورات خاصی می باشد که در ادامه با آنها آشنا می شویم.

برای درک بهتر کاربرد تابع نویسی توسط بلوک های FB به مثال ارائه شده توجه فرمایید.

فرض کنید در یک فرآیند صنعتی از ۴ موتور استفاده شده است. عملکرد هر ۴ موتور به صورت چپگرد راستگرد می باشد. در این مثال هدف ساخت یک تابع چپگرد راستگرد برای ۴ موتور در یک بلوک FB می باشد. برای این منظور ابتدا یک بلوک FB ایجاد و وارد محیط آن می شویم.

یک بلوک FB از سه قسمت تشکیل شده است.

نام بلوک : این قسمت یک نام اختیاری برای بلوک مورد نظر می باشد.

محل ایجاد پارامترها : در این قسمت کاربر می بایست در ابتدا پارامترهایی که در برنامه کنترلی نیاز دارد را وارد کند. این پارامترها می تواند ورودی ، خروجی ، تایمر و یا شمارنده باشند.

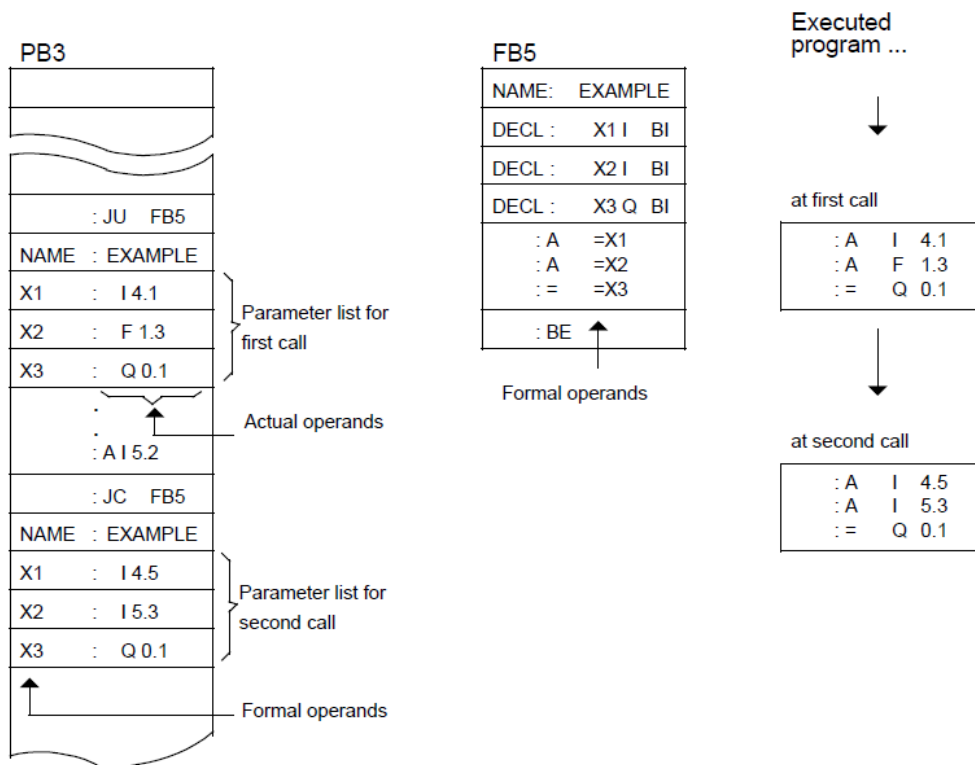
محل نوشتن برنامه : در این قسمت کاربر توسط پارامترهایی که ایجاد نموده است شروع به به برنامه نویسی به روش STL می کند.

به برنامه موجود در FB 1 این مثال توجه نمایید.

NAME:	L_R	
DECL:	STAL	IBI
DECL:	STAR	IBI
DECL:	STOP	IBI
DECL:	CONL	QBI
DECL:	CONR	QBI
	A	=STAL
	AN	=CONR
	S	=CONL
	A	=STAR
	AN	=CONL
	S	=CONR
	A	=STOP
	RB	=CONL
	RB	=CONR
	BE	

در این برنامه نام بلوک را به اختیار L_R قرار دادیم. سپس پارامترهای تابع را با اسامی دلخواه ایجاد کردیم. به عنوان مثال ورودی STAL یک ورودی (I) با فرمت BI می باشد. به همین منظور عبارت IBI جلوی نام پارامتر درج می شود. از عبارت DECL نیز برای تعریف کردن پارامتر استفاده می کنیم.

نحوه اختصاص پارامترهای فانکشن بلوک را مشاهده می کنید.



دستورات منطقی در FB ها به صورت زیر می باشند.

Boolean Logic Operations													
A=	Formal operand					N	Y	N	160 *	160 *	160 *	3.6 *	AND operation: scan formal operand for "1". (Data type: BI)
								
AN=	Formal operand					N	Y	N	163 *	163 *	163 *	3.6 *	AND operation: scan formal operand for "0". (Data type: BI)
								
O=	Formal operand					N	Y	N	164 *	164 *	164 *	3.6 *	OR operation: scan formal operand for "1". (Data type: BI)
								
ON=	Formal operand					N	Y	N	165 *	165 *	165 *	3.6 *	OR operation: scan formal operand for "0". (Data type: BI)
								

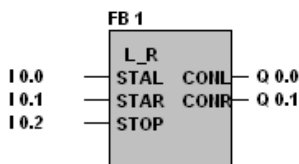
دستورات SET و RESET در بلوک های FB

Set/Reset Operations													
S=	Formal operand					Y	N	Y	150*	150*	150*	3.6*	Set a formal operand, (with RLO =1). (Data type: BI)
								
RB=	Formal operand					Y	N	Y	150*	150*	150*	3.6*	Reset a formal operand, (with RLO =1). (Data type: BI)
								

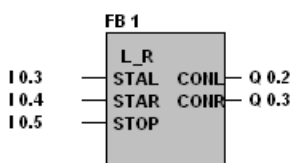
فراخوانی FB1 در OB 1

در ادامه بلوک FB1 را برای ۴ موتور در OB1 فراخوانی می کنیم.

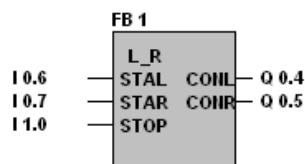
MOTOR1



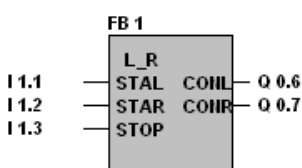
MOTOR2



MOTOR3



MOTOR4



مثال ۲

در این مثال می خواهیم یک تابع جهت گرفتن میانگین ۴ ورودی طراحی کنیم.

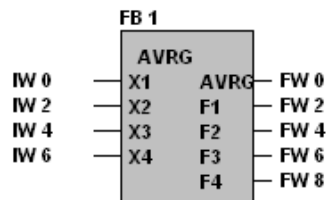
بلوک FB1

```

NAME:      AVRG
DECL:      X1      IW
DECL:      X2      IW
DECL:      X3      IW
DECL:      X4      IW
DECL:      AVRG     QW
DECL:      F1      QW
DECL:      F2      QW
DECL:      F3      QW
DECL:      F4      QW

L          =X1
L          =X2
+F
T          =F1
L          =X3
L          =X4
+F
T          =F2
L          =F1
L          =F2
+F
T          =F3
L          =F3
L          KF +4
/F
T          =AVRG
BE
    
```

فراخوانی بلوک FB1 در OB1



این تابع را می توان برای ورودی های دیگر نیز فراخوانی نمود.

دستورات Load و Transfer در بلوک های FB

Load and Transfer Operations									
L=	Formal operand • • • • •	N	N	N	147 *	147 *	147 *	3.6 *	Load the value of the formal operand into ACCUM 1 (Data type: BY, W; additional actual operands: DL, DR, DW).
L	RS	N	N	N	89	89	89	89	Load a word from the system data range into ACCUM 1.
LC=	Formal operand • • • • •	N	N	N	145 *	145 *	145 *	3.6 *	Load the value of the formal operand in BCD code into ACCUM 1.
LW=	Formal operand • • • • •	N	N	N	124 *	124 *	124 *	3.6 *	Load a formal operand bit pattern into ACCUM 1 (parameter type: D; Data type: KC, KF, KH, KM, KS, KT, KY).
T=	Formal operand • • • • •	N	N	N	148 *	148 *	148 *	3.6 *	Transfer the contents of ACCUM 1 to the formal operand (Data type: BY, W; additional actual operands: DL, DR, DW).

مثال ۳

در این مثال می خواهیم توسط بلوک FB1، یک تابع پالس با عرض های دلخواه بسازیم.

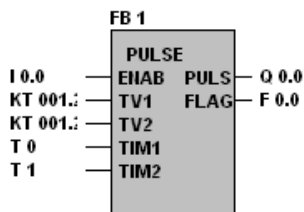
بلوک FB1

NAME:	PULSE	
DECL:	ENAB	IBI
DECL:	TV1	DKT
DECL:	TV2	DKT
DECL:	TIM1	T
DECL:	TIM2	T
DECL:	PULS	QBI
DECL:	FLAG	QBI
	A	=ENAB
	AN	=FLAG
	LW	=TV1
	SD	=TIM1
	A	=TIM1
	=	=PULS
	A	=PULS
	LW	=TV2
	SD	=TIM2
	A	=TIM2
	=	=FLAG
	BE	

251

فراخوانی FB1 در OB1 برای ۲ پالس با عرض های مختلف

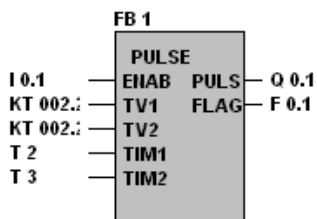
PULSE1



;PULSE1

NAME:	JU	FB 1
ENAB:	PULSE	
TV1 :	I 0.0	
TV2 :	KT 001.2	
TIM1:	KT 001.2	
TIM2:	T 0	
PULS:	T 1	
FLAG:	Q 0.0	
	F 0.0	

PULSE2



;PULSE2

NAME:	JU	FB 1
ENAB:	PULSE	
TV1 :	I 0.1	
TV2 :	KT 002.2	
TIM1:	KT 002.2	
TIM2:	T 2	
PULS:	T 3	
FLAG:	Q 0.1	
	F 0.1	
	BE	

دستورات تایمر و کانتر در بلوک های FB

Timer and Counter Operations														
FR					•	•	Y	N	Y	3.7	3.7	3.7	1.9	Enable a timer/counter for cold restart. If RLO="1", - "FR T" restarts the timer - "FR Z" sets, decrements, or increments the counter
FR=	Formal operand						Y	N	Y	144 *	144 *	144 *	3.6 *	Enable formal operand (timer/counter) for cold re-start (for detailed description, see "FR" operation).
SP=	Formal operand						Y	N	Y	144 *	144 *	144 *	3.6 *	Start a timer (formal operand) as pulse with the value stored in ACCUM 1.
SR=	Formal operand						Y	N	Y	144 *	144 *	144 *	3.6 *	Start an on-delay timer (formal operand) with the value stored in ACCUM 1.
SEC=	Formal operand						Y	N	Y	144 *	144 *	144 *	3.6 *	Start a timer (formal operand) as extended pulse with the value stored in ACCUM 1, or set a counter (formal operand) with the next indicated count value.
SSU=	Formal operand						Y	N	Y	144 *	144 *	144 *	3.6 *	Start a stored on-delay timer (formal operand) with the value stored in ACCUM 1, or increment a counter (formal operand).
SFD=	Formal operand						Y	N	Y	144 *	144 *	144 *	3.6 *	Start an off-delay timer () (formal operand) with the value stored in ACCUM 1, or decrement a counter () (formal operand).

سایر دستورات

Set/Reset Operations																		
S=	Formal operand				•	•	•	•	•	Y	N	Y	150*	150*	150*	3.6*	Set a formal operand, (with RLO =1). (Data type: BI)	
RB=	Formal operand				•	•	•	•	•	Y	N	Y	150*	150*	150*	3.6*	Reset a formal operand, (with RLO =1). (Data type: BI)	
RD=	Formal operand								•	•	Y	N	Y	146*	146*	146*	3.6*	Reset a formal operand (digital) (with RLO =1).
==	Formal operand				•	•	•	•	•	N	N	Y	150*	150*	150*	3.6*	The value of the RLO is assigned to the status of the formal operand (Data type: BI)	

Load and Transfer Operations																		
L=	Formal operand				•	•	•	•	•	N	N	N	147 *	147 *	147 *	3.6 *	Load the value of the formal operand into ACCUM 1 (Data type: BY, W; additional actual operands: DL, DR, DW).	
L	RS									N	N	N	89	89	89	89	Load a word from the system data range into ACCUM 1.	
LC=	Formal operand								•	•	N	N	N	145 *	145 *	145 *	3.6 *	Load the value of the formal operand in BCD code into ACCUM 1.
LW=	Formal operand				•	•	•	•	•	N	N	N	124 *	124 *	124 *	3.6 *	Load a formal operand bit pattern into ACCUM 1 (parameter type: D; Data type: KC, KF, KH, KM, KS, KT, KY).	
T=	Formal operand				•	•	•	•	•	N	N	N	148 *	148 *	148 *	3.6 *	Transfer the contents of ACCUM 1 to the formal operand (Data type: BY, W; additional actual operands: DL, DR, DW).	

Operation (STL)	Operands					1 RLO depend.? 2 RLO affected? 3 RLO reloaded?			Typical Execution Time in μ sec.				Function
	I	Q	F	T	C	1	2	3	CPU 941	CPU 942	CPU 943	CPU 944	
Conversion Operations													
CFW						N	N	N	1.6	1.6	0.8	0.8	Form the one's complement of ACCUM 1.
CSW						N	N	N	1.6	1.6	0.8	0.8	Form the two's complement of ACCUM 1. CC 1/CC 0 and OV are affected.
Shift Operations													
SLW	Parameter n=0 to 15					N	N	N	1.6	1.6	0.8	0.8	Shift the contents of ACCUM 1 to the left by the value specified in the parameter. Unassigned positions are padded with zeros. CC 1/CC 0 are affected.
SRW	Parameter n=0 to 15					N	N	N	1.6	1.6	0.8	0.8	Shift the contents of ACCUM 1 to the right by the value specified in the parameter. Unassigned positions are padded with zeros. CC 1/CC 0 are affected.

Jump Operations													
JU=	Symbolic address maximum 4 characters					N	N	N	1.6	1.6	0.8	0.8	Jump unconditionally to the symbolic address.
JC=	Symbolic address maximum 4 characters					Y	Y ¹	Y	1.6	1.6	0.8	0.8	Jump conditionally to the symbolic address. (If the RLO is "0", it is set to "1").
JZ=	Symbolic address maximum 4 characters					N	N	N	1.6	1.6	0.8	0.8	Jump if the result is zero. The jump is made only if CC 1=0 and CC 0=0. The RLO is not changed.
JN=	Symbolic address maximum 4 characters					N	N	N	1.6	1.6	0.8	0.8	Jump if the result is not zero. The jump is made only if CC 1 CC 0 . The RLO is not changed.
JP=	Symbolic address maximum 4 characters					N	N	N	1.6	1.6	0.8	0.8	Jump if the result >0. The jump is made only if CC 1=1 and CC 0=0. The RLO is not changed.
JM=	Symbolic address maximum 4 characters					N	N	N	1.6	1.6	0.8	0.8	Jump if the result <0. The jump is made only if CC 1=0 and CC 0=1. The RLO is not changed.
JO=	Symbolic address maximum 4 characters					N	N	N	1.6	1.6	0.8	0.8	Jump on overflow. The jump is made only if the OVERFLOW bit is set. The RLO is not changed.

Arithmetic Operations													
ADD	BF					N	N	N	57	57	57	57	Add byte constant (fixed point) to ACCUM 1.
ADD	KF					N	N	N	90	90	90	90	Add fixed-point constant (word) to ACCUM 1.

سایر دستورات نیز در منوال های S5 موجود می باشد.

بلوکهای تابع ساز خاص

در S5 یکسری بلوک های سیستمی FB وجود دارد که توسط شرکت زیمنس برنامه ریزی شده است. کاربرد این بلوکها بسیار متنوع بوده و برای کاربردهای مختلف طراحی شده اند. مثلا FB 250 یک فانکشن بلوک استاندارد بوده که توسط آن می توان یک سیگنال ورودی آنالوگ را مقیاس گذاری نمود. به عنوان مثال در S5-115U لیست برخی از این بلوک ها را ملاحظه می کنید. بلوک FB240 تا FB251 به عنوان بلوک های سیستمی می باشند که امکان نوشتن به آنها وجود ندارد.

Table 11-1. Overview of Integral Blocks

Type	Block		Call Length (in Words)	Execution time (in Millise- conds)	Function
	No.	Name			
FB	238	COMPR ¹	4	< 0.6	Compress PLC memory
FB	239	DELETE	5	< 0.8	Delete block
FB	240	COD : B4	5	< 0.8	4-tetrad BCD code converter
FB	241	COD : 16	6	< 1.0	16-bit fixed-point converter
FB	242	MUL : 16	7	< 1.0	16-bit binary multiplier
FB	243	DIV : 16	10	< 2.3	16-bit binary divider
FB	244	SEND ²			Send data
FB	245	RECEIVE ²			Receive data
FB	246	FETCH		< 4	Fetch data
FB	247	CONTROL		approx. 0.6	Monitor job processing
FB	248	RESET		< 4	Delete job
FB	249	SYNCHRON		6.5 ms to 10 s	Initialize interface
FB	250	RLG : AE	11	2.0	Read analog value (see Chapter 10)
FB	251	RLG : AA	9	5.2	Output analog value (see Chapter 10)
OB	31			< 0.14	Restart scan time
OB	160				Variable time loop
OB	251			2.0	PID control algorithm
OB	254 ³				Read digital inputs
OB	255 ³				Transfer PIQ to outputs
DB	1				Parameterize internal functions

بلوک FB240

توسط این بلوک می توان یک کد BCD را به باینری (۱۶ بیت) تبدیل کرد.

Call and Parameter Assignments

Parameter	Type	Data Type	Assignment	Meaning	STL
BCD	I	W	- 9999 to +9999	BCD number	: JU FB 240
SBCD	I	Bi	"1" for "-" "0" for "+"	Sign of the BCD number	Name : COD : B4 BCD : SBCD : BINARY:
BINARY	Q	W	16 bits "0" or "1"	Binary number	

بلوک FB241

توسط این بلوک می توان یک عدد باینری را به کد BCD تبدیل کرد.

Call and Parameter Assignments

Parameter	Type	Data Type	Assignment	Meaning	STL
BINARY	I	W	- 32768 to+32767	Binary number	: JU FB 241 Name : COD : 16 BINARY: SBCD : BCD2 : BCD1 :
SBCD	Q	Bi	"1" for "-" "0" for "+"	Sign of the BCD number	
BCD2	Q	By	2 tetrads	BCD number tetrads 4 and 5	
BCD1	Q	W	4 tetrads	BCD number tetrads 0 to 3	

255

بلوک FB242

این بلوک یک بلوک ضرب می باشد.

Call and Parameter Assignments

Parameter	Type	Data Type	Assignment	Meaning	STL
Z1	I	W	- 32768 to+32767	Multiplier	: JU FB 242 Name : MUL : 16 Z1 : Z2 : Z3=0 : Z32 : Z31 :
Z2	I	W	- 32768 to+32767	Multiplicand	
Z3=0	Q	Bi	"1", if the product is zero	Scan for zero	
Z32	Q	W	16 bits	Product, high-order word	
Z31	Q	W	16 bits	Product, low-order word	

بلوک FB243

این بلوک یک بلوک تقسیم می باشد.

Call and Parameter Assignment

Parameter	Type	Data Type	Contents	Meaning	STL
Z1	I	W	- 32768 to+32767	Dividend	: JU FB 243 Name : DIV : 16 Z1 : Z2 : OV : FEH : Z3=0 : Z4=0 : Z3 : Z4 :
Z2	I	W	- 32768 to+32767	Divisor	
OV	Q	Bi	"1", if overflow	Overflow indicator	
FEH	Q	Bi	"1" for division by zero		
Z3=0	Q	Bi	"1": quotient is zero	Scan for zero	
Z4=0	Q	Bi	"1": remainder is zero	Scan for zero	
Z3	Q	W	16 bits	Quotient	
Z4	Q	W	16 bits	Remainder	

مثال- تولید پالس مداوم (کنترل شده)

برنامه این مثال را توسط FB طراحی نمایید.

در این برنامه با استفاده از یک تایمر SE هر ۲۵۰ میلی ثانیه یکبار در خروجی Q0.0 پالس خواهیم داشت. در این مثال در طول زمانی که تایمر فعال است، مقدار FW100 یکی افزایش می یابد ولی وقتی تایمر غیرفعال است خاتمه می یابد. بلاک با دستور BEC خاتمه می یابد و FW 100 افزایش پیدا نمی کند.

برنامه FB1

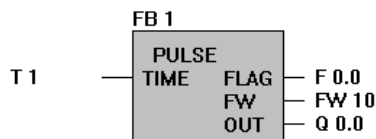
256

```

NAME:      PULSE
DECL:      FLAG      QBI
DECL:      TIME      T
DECL:      FW        QW
DECL:      OUT       QBI

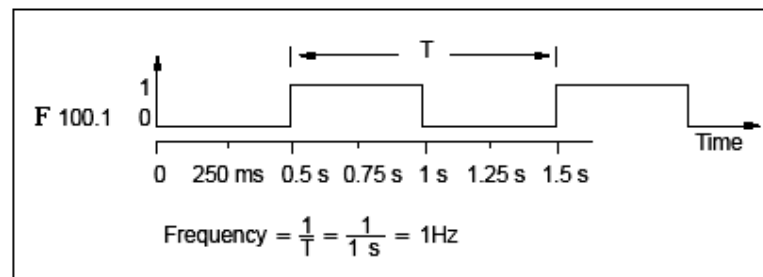
          AN          =FLAG
          L           KT 025.0
          SEC        =TIME
          A           =TIME
          BEC
          L           =FW
          I           1
          T           =FW
          A           F 11.2
          =           =OUT
          BE
    
```

برنامه OB 1



اگر بخواهیم فرکانس پالس را تعیین نماییم، کفایت با توجه به رابطه $F=1/T$ زمان صفر و یک روشن شدن را جمع نماییم و معکوس کنیم. جدول زمان های تناوب این مثال بصورت زیر می باشد.

Bits of FW 100	Frequency in Hertz	Duration
F 100.0	2.0	0.5 s (250 ms on/250 ms off)
F 100.1	1.0	1 s (0.5 s on/0.5 s off)
F 100.2	0.5	2 s (1 s on/1 s off)
F 100.3	0.25	4 s (2 s on/2 s off)
F 100.4	0.125	8 s (4 s on/4 s off)
F 100.5	0.0625	16 s (8 s on/8 s off)
F 100.6	0.03125	32 s (16 s on/16 s off)
F 100.7	0.015625	64 s (32 s on/32 s off)



بلوک های OB

در PLC های S5 و S7-300,400 علاوه بر بلوک OB1 بلوک هایی نیز تحت عنوان OB وجود دارند. این OB ها با شماره های خاص و کاربردهای متفاوت در دسترس می باشند. اکثر OB ها به غیر از OB1 در شرایط عادی توسط CPU پردازش نمی شوند، بلکه اجرای آنها منوط به برقراری شرایطی می باشد. یکی از خصوصیات بلوک های OB عدم فراخوانی آنها در سایر بلوک ها می باشد. همچنین هر یک از این بلوک ها دارای اولویت می باشند که فراخوانی و اجرای آنها با توجه به اولویت آنها صورت می گیرد. در ادامه با گروه های مختلف این بلوک ها آشنا می شویم.

OB های راه اندازی (Startup)

زمانی که PLC شروع به کار می کند، در صورت وجود بلوک های Startup، سیستم عامل یکبار این بلوک ها را اجرا کرده و سپس به بلوک OB1 مراجعه می کند. در بسیاری از کاربردها نیاز می باشد که در لحظه راه اندازی شرایطی را به عنوان شرایط اولیه به PLC و یا پروسه اعمال کرد که در این صورت می توان از این بلوک ها استفاده نمود. به عنوان مثال فرض کنید که می خواهیم زمانی که PLC به هر دلیلی به حالت STOP سوئیچ گردید، مجدداً با RUN شدن PLC اعدادی به عنوان مقادیر اولیه به فضاهایی از حافظه ارسال شوند. در این صورت می توان برنامه انتقال اعداد به حافظه ها را در بلوک های Startup برنامه ریزی نمود.

اجرای برنامه بعد از ری استارت

بعد از یک ری استارت:

- ✓ بعد از انجام یک STOP به RUN (Cold Restart دستی)
- ✓ بعد از یک Power Off به Power On (Cold Restart) اتوماتیک در صورتی که CPU در وضعیت قبلی، Run باشد)

بعد از رخ دادن هر یک از این دو حالت، CPU به صورت اتوماتیک، بلوک های STARTUP را برای یک سیکل اجرا می کند. منظور از بلوک های STARTUP همان بلوک های OB می باشند. از این بلوک ها می توان جهت ارسال مقادیر اولیه به حافظه ها و یا چک کردن شرایط اولیه در برنامه استفاده نمود.

زمانی که یک ری استارت Cold به صورت دستی انجام می شود، CPU بلوک OB21 را فراخوانی می کند و در صورت وجود، برنامه آن برای یک سیکل اجرا می شود.

زمانی که یک ری استارت Cold به صورت اتوماتیک (Power On) انجام می شود، CPU بلوک OB22 را فراخوانی می کند و در صورت وجود، برنامه آن برای یک سیکل اجرا می شود.

اگر این بلوک ها برنامه ریزی شده باشند، برنامه آنها قبل از اجرای بلوک OB1 اجرا می شود.

مثال: در این مثال می خواهیم زمانی که تغذیه CPU وصل شد، ابتدا CPU تمام مازول های ورودی / خروجی را از لحاظ سالم و موجود بودن در اسلات بررسی کند. اگر مشکلی تشخیص داده شد، CPU به مد استپ سوئیچ شود.

OB22 STL	Explanation	
:L KF +0	Output words 0, 2, and 4 are set to "0".	
:T PW 0		
:T PW 2		
:T PW 4		
:L PW 6		
:L PW 8		
:L PW 10		
:BE		
		The information in input words 6, 8, and 10 is loaded into ACCUM 1 consecutively.

در این مثال در صورتی که CPU نتواند به فضاهای آدرسی دهی شده در OB22 دسترسی پیدا کند، به مد STOP سوئیچ می شود.

مثال: برنامه ریزی OB21 و FB1

در این مثال می خواهیم بعد از انجام یک ری استارت سرد توسط سلکتور روی CPU، فلگ بایت 0 تا فلگ بایت 99 از حافظه، مقادیرشان 0 و فلگ بایت 100 تا فلگ بایت 127 که مربوط به دیتاهای مهم ماشین هستند، مقادیرشان حفظ شوند.

OB21 STL	Explanation
:JU FB 1	Call FB 1 unconditionally.
NAME :CLEAR F	
:BE	

برنامه FB1

FB1 STL	Explanation
NAME :CLEAR F	
:L KF +0	FW 200 is preset with "0".
:T FW 200	"0" is stored in ACCUM 1.
L10 :L KF +0	The contents of FW 200 indicate the address of the current flag word.
:DO FW 200	The current flag word is set to "0".
:T FW 0	
:L FW 200	
:I 2	The contents of FW 200 are incremented by 2.
:T FW 200	
:L KF +100	The comparison value "100" is loaded into ACCUM 1.
:<F	If the contents of FW 200 < 100, the program jumps to label 10.
:JC =L10	
:BE	Bytes FB 0...99 are set to "0".

در این مثال FW200 به عنوان کانتر حلقه می باشد که در هر سیکل ۲ واحد به مقدارش افزوده می شود.

OB های سیکلی (Cyclic Interrupt)

این بلوک ها بطور سیکلی و مداوم همانند OB1 اجرا نمی شوند. بلکه زمان مراجعه به این بلوک ها توسط کاربر مشخص می شود. یکی از کاربردهای این بلوک ها در کنترلرهای PID و یا نمونه برداری در زمان های مشخص از سیگنالهای ورودی آنالوگ می باشد.

بلوک های OB10 تا OB13 به عنوان بلوک های Time-controlled در اختیار برنامه نویس می باشند. زمان پیش فرض مراجعه به OB13، ۱۰۰ میلی ثانیه می باشد. جهت تغییر این زمان ها می بایست به بلوک DB1 که یک دیتا بلاک رزرو شده برای سیستم می باشد، مراجعه شود. در این بلوک، DW97 تا DW100 مربوط به زمان این بلوک ها می باشند. مقدار زمان اجرای این بلوک ها می تواند از 10 میلی ثانیه تا 10 دقیقه تعیین شوند.

در جدول شکل مقادیر پیش فرض برای هر بلوک را ملاحظه می کنید.

Setting an Interval Time of 1 sec. for OB13:		
OB 21	OB 22	FB 21
NAME : JU FB 21	NAME :JU FB 21	NAME :TIME ON
NAME : TIME ON	NAME :TIME ON	:L KF +100
.	.	:T BS 97
.	.	:BE

در مثال زیر، در زمان Startup به صورت دستی یا اتوماتیک، زمان OB13 به ۱ ثانیه تغییر می کند.

OB 21	OB 22	FB 21
: JU FB 21	:JU FB 21	NAME :TIME ON
NAME : TIME ON	NAME :TIME ON	:L KF +100
.	.	:T BS 97
.	.	:BE

همانطور که در برنامه FB21 ملاحظه می کنید، مقدار ۱۰۰ که معادل ۱ ثانیه می باشد به سطر ۹۷ بلوک سیستمی منتقل می شود.

260

بلوک OB19

وقتی که در برنامه یک بلوک به CPU لود نشده باشد و این بلوک فراخوانی شود، سیستم عامل، بلوک وقفه OB19 را فراخوانی می کند. در مثال زیر زمانی که این موضوع رخ دهد، CPU به OB19 مراجعه و CPU را با اجرای دستور STP، متوقف می کند.

OB19 STL	Explanation
:STP :BE	STOP statement

بلوک OB23

وقتی CPU در زمان RUN شدن نتواند به ماژول ها با آدرس دهی PW، PB و... در زمان مشخص دسترسی پیدا کند، این بلوک را تقاضا می کند. عدم دسترسی به این فضاها می تواند به دلیل معیوب شدن ماژول، نبود ماژول در رک و یا خطای برنامه باشد. در صورتی که این بلوک موجود نباشد، CPU به مد STOP سوئیچ می شود.

بلوک OB34

به صورت پیوسته وضعیت سطح ولتاژ باتری را چک می کند. اگر سطح ولتاژ باتری کمتر از حد مجاز شود و یا باتری خارج شود، LED خطای باتری روشن می شود و CPU این بلوک را تقاضا می کند.

چندین بلوک وقفه دیگر نیز در S5 وجود دارند که سیستم عامل به ازای رخ دادن شرایط مختلف آنها را فراخوانی می کند. علاقمندان جهت کسب اطلاعات بیشتر در رابطه با کاربرد سایر بلوک های وقفه، می توانند به منوال های زیرمنس مراجعه کنند.

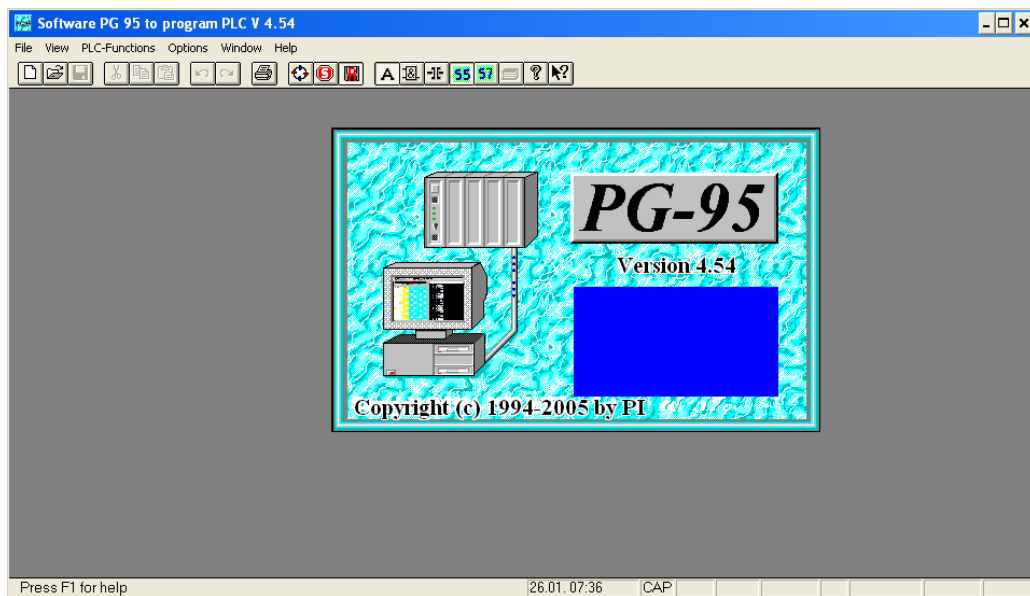
آشنایی با نرم افزارهای برنامه نویسی

برای پروگرام کردن PLC های سری نرم افزارهای متعددی عرضه شده است.

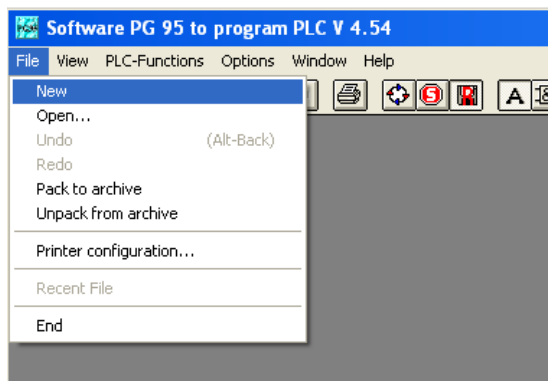
- SIMATIC STEP5 V7.23
- S5 For Windows
- PG95
- PG2000

در این بخش قصد داریم نحوه کار با نرم افزار PG95 را به طور مختصر بررسی کنیم.

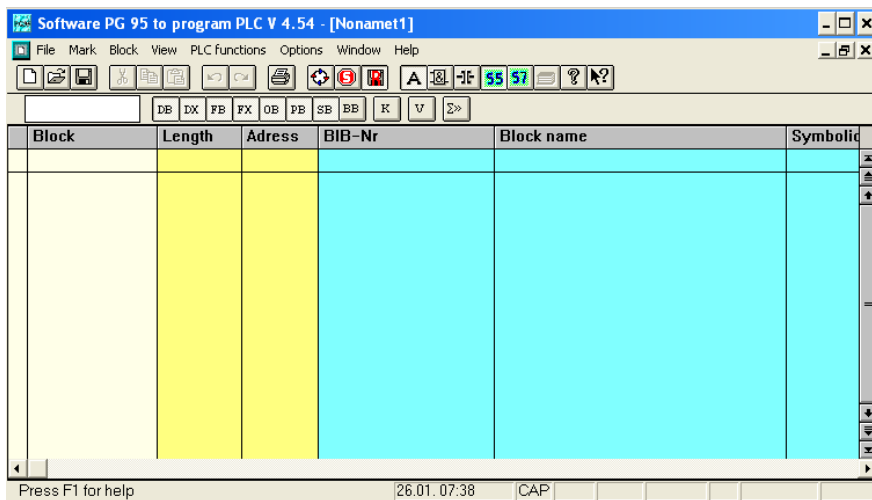
261



✓ ساخت پروژه

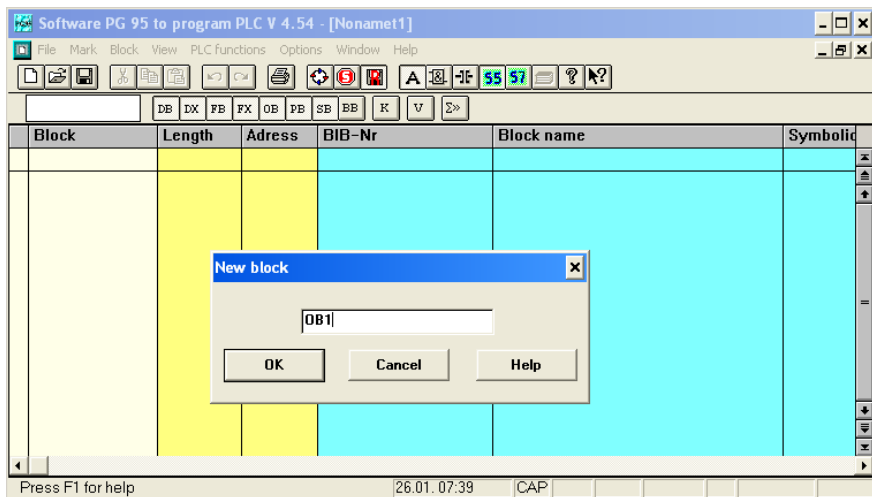
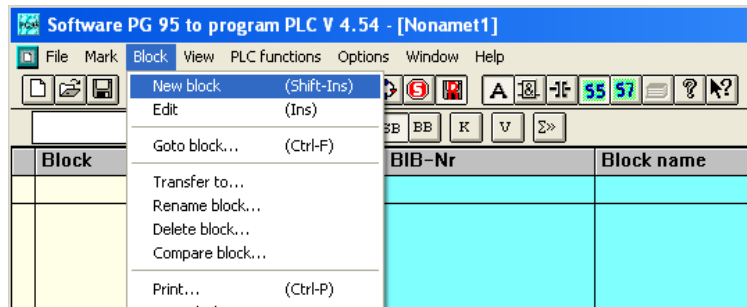


محیط نرم افزار را در شکل زیر ملاحظه می کنید.



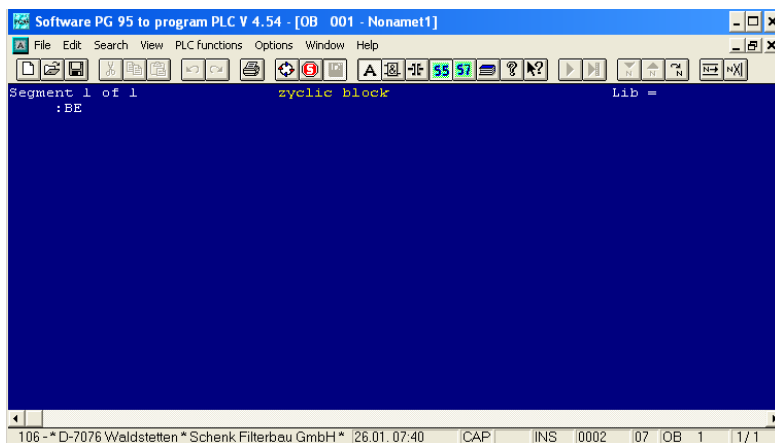
262

✓ ساخت بلوک OB1



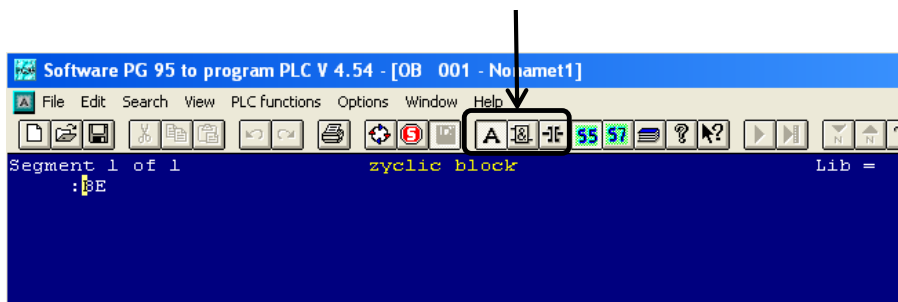
بعد از ساخت بلوک OB1، وارد محیط برنامه نویسی این بلوک می شویم.

نرم افزار به صورت پیش فرض به زبان STL بلوک OB1 را باز می کند.

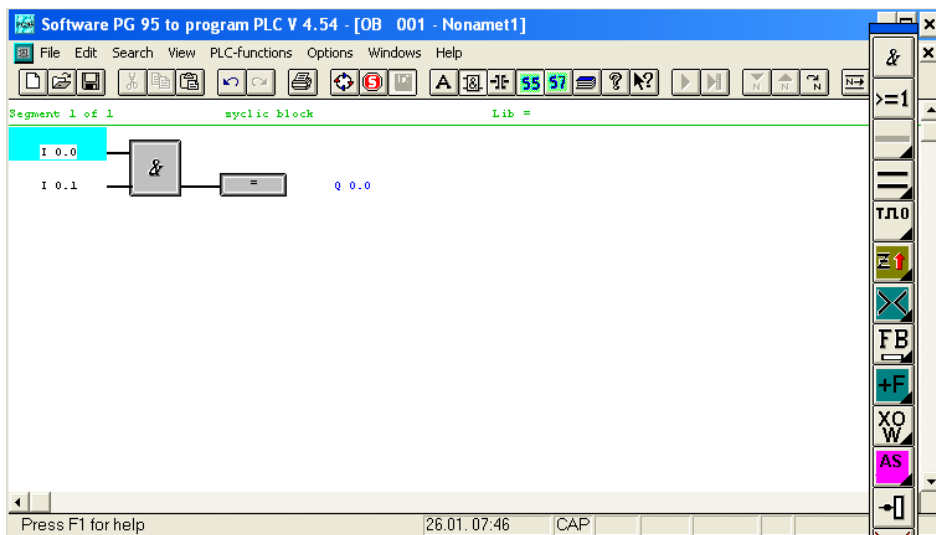


263

توسط آیکن های زیر می توان روش های برنامه نویسی را تغییر داد.

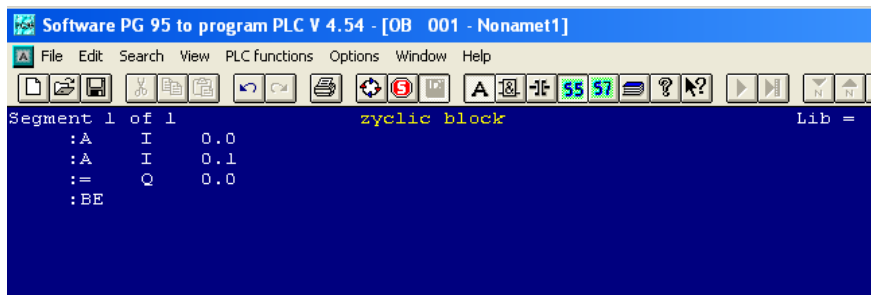


یک برنامه ساده به زبان FBD ایجاد کنید.



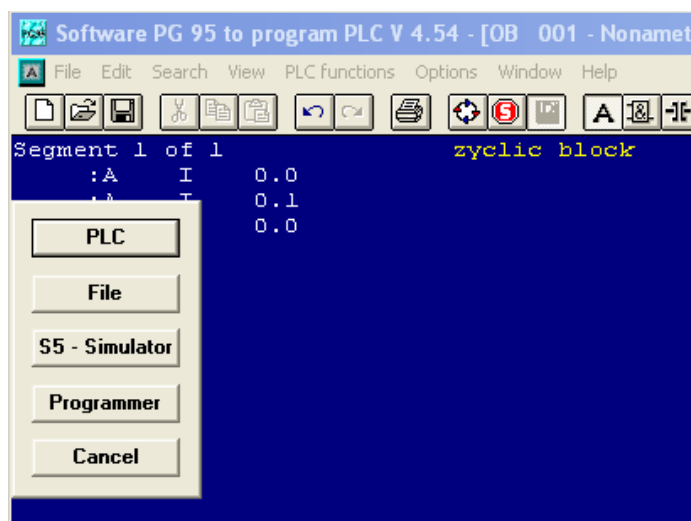
همانطور که در شکل فوق ملاحظه می کنید، ابزارهای مورد نیاز در روش LAD به صورت یک نوار در سمت راست نمایان می باشد.

برنامه را به روش های دیگر تبدیل کنید.

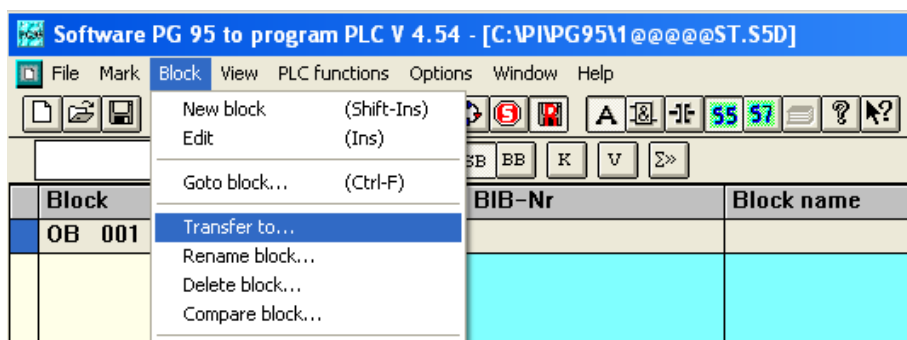


در ادامه بر روی آیکون Save کلیک می کنیم. با کلیک بر روی این آیکون پنجره ای جهت مشخص نمودن محل ذخیره فایل نمایان می شود.

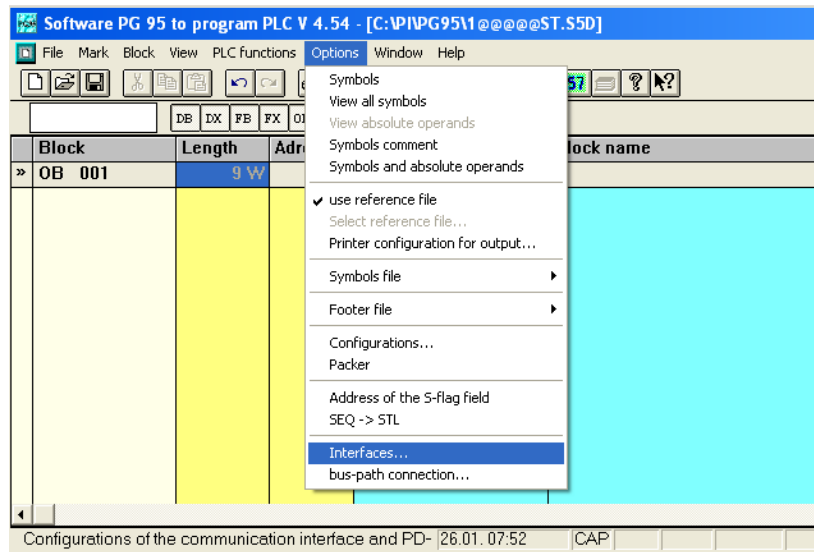
264



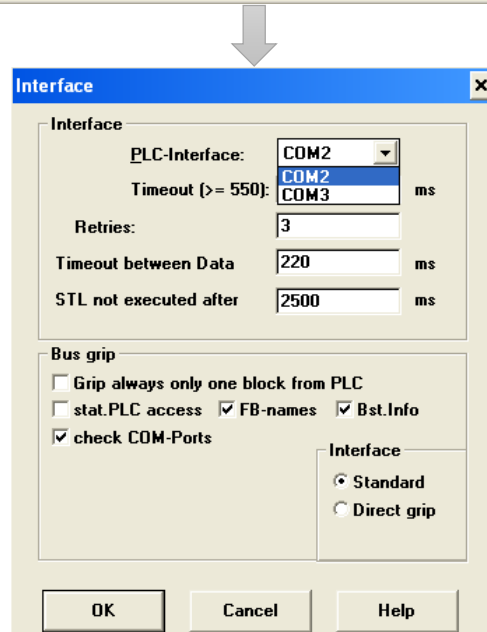
اگر کابل برنامه ریزی به CPU متصل باشد، عملیات ذخیره سازی می تواند به صورت مستقیم بر روی CPU انجام شود. همچنین کاربر می تواند پروژه خود را در قالب یک فایل بر روی PC ذخیره کند. بعد از بستن بلوک OB1، با بازگشت به محیط اصلی برنامه، امکان دانلود بلوک توسط مسیر زیر امکان پذیر می باشد.



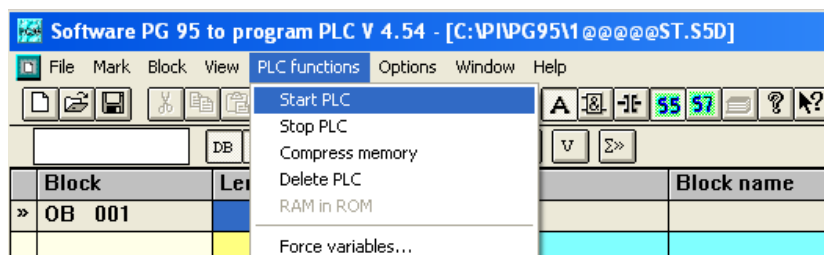
✓ تنظیم درگاه ارتباطی با CPU



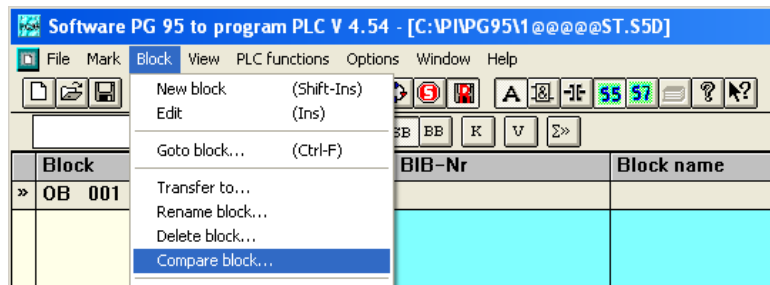
265



✓ انجام عملیات STOP یا RUN کردن CPU توسط نرم افزار

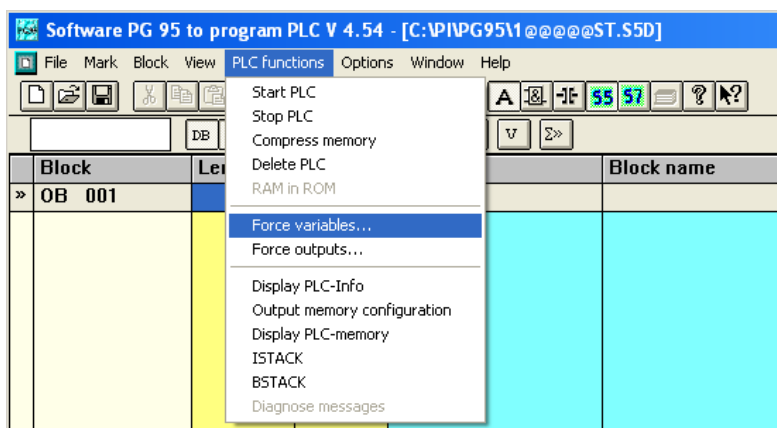


✓ مقایسه بلوک های دو مسیر

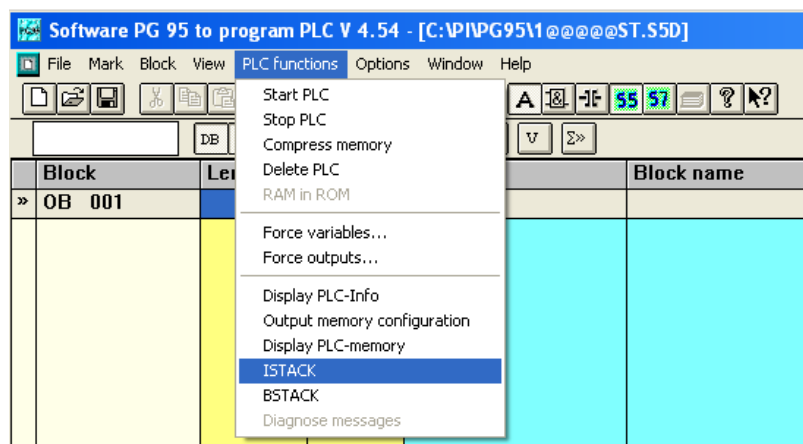


266

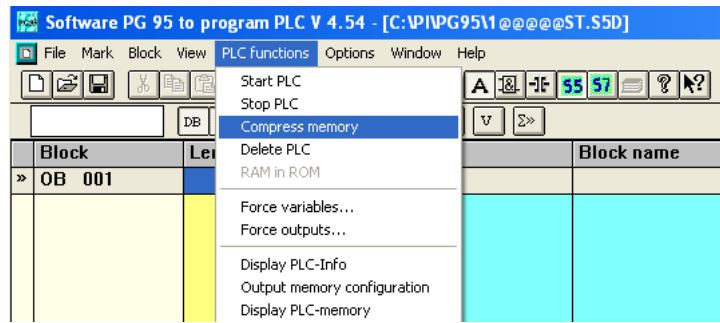
✓ انجام عملیات FORCE بر روی متغیرها و خروجی ها



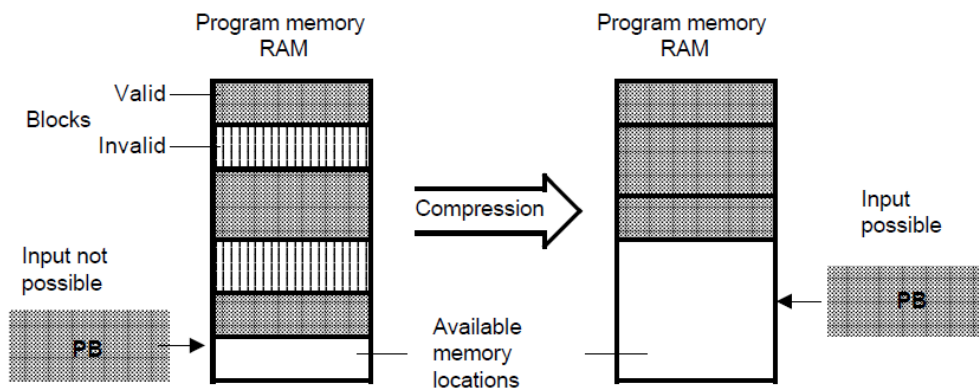
✓ مشاهده فضای ISTACK و BSTACK جهت انجام عملیات عیب یابی در زمان STOP شدن CPU



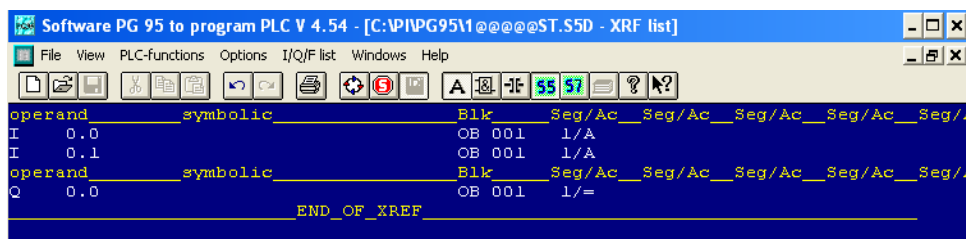
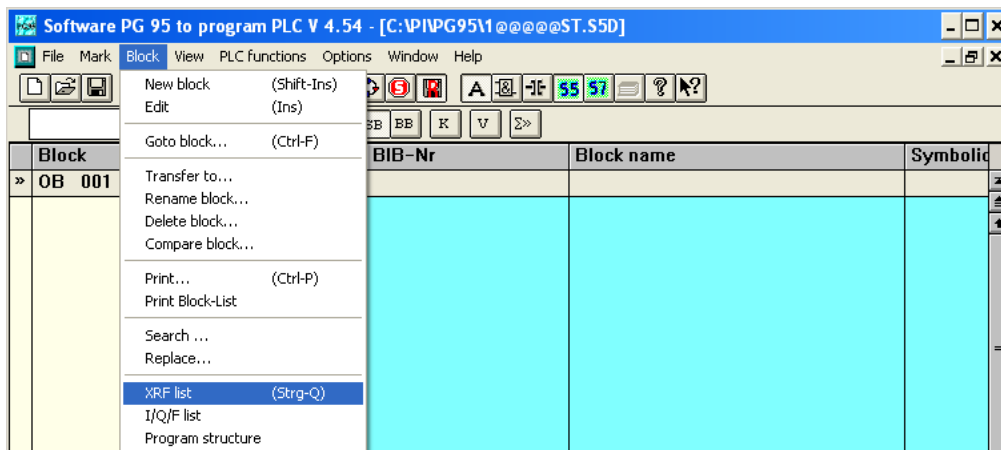
✓ یکپارچه سازی فضای حافظه CPU



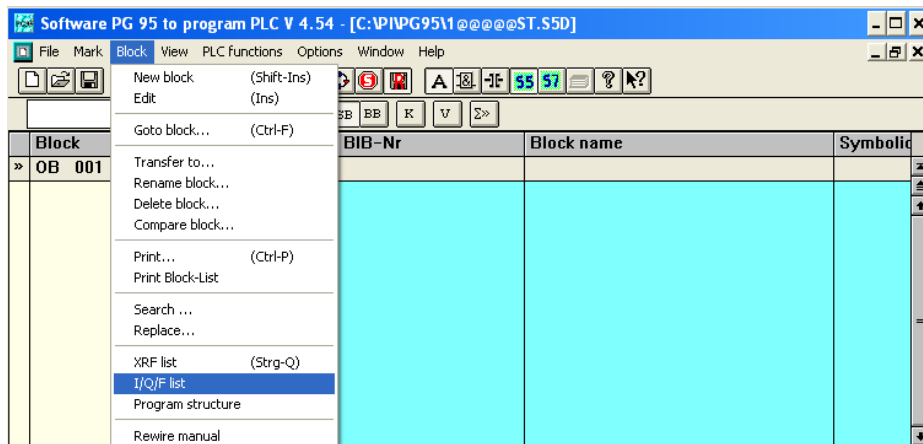
267



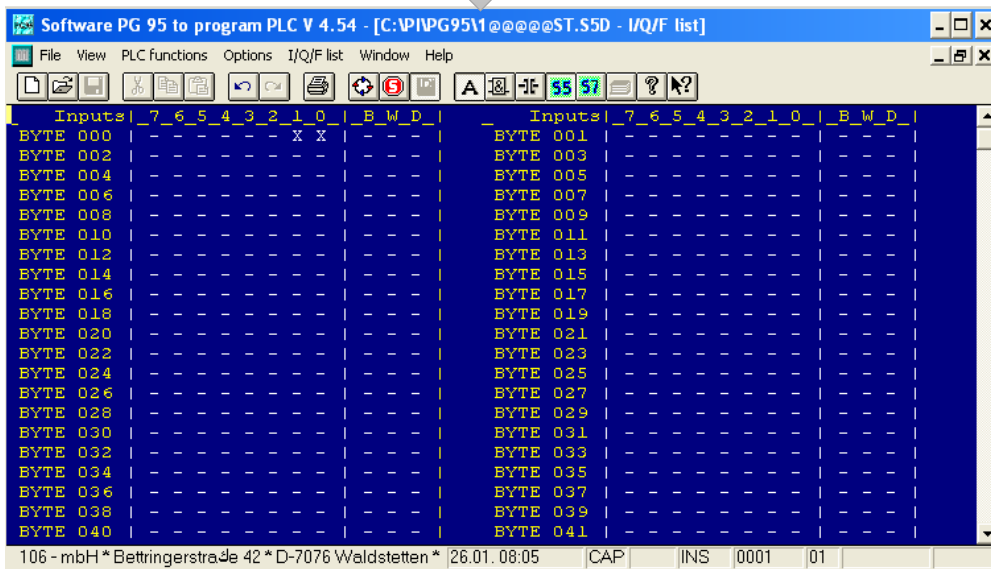
✓ مشاهده آدرس های استفاده شده در برنامه و محل آنها



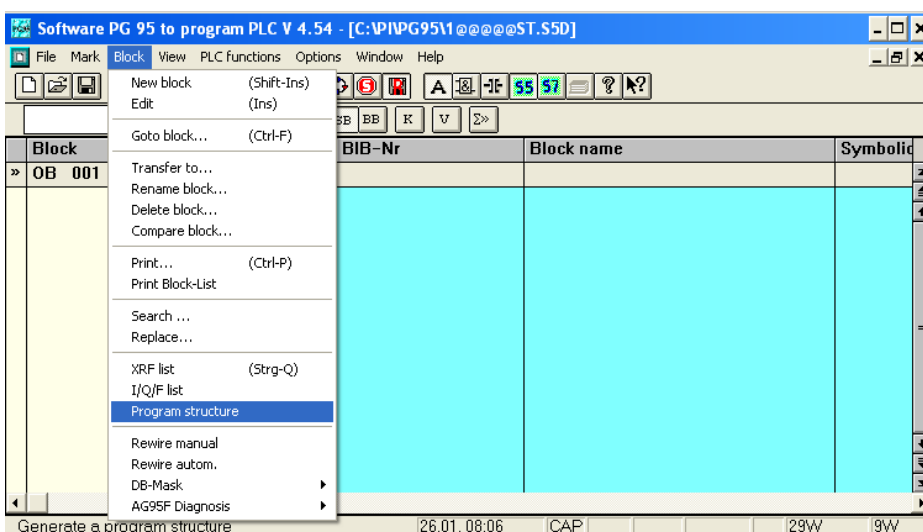
✓ مشاهده فضاهای استفاده شده در برنامه



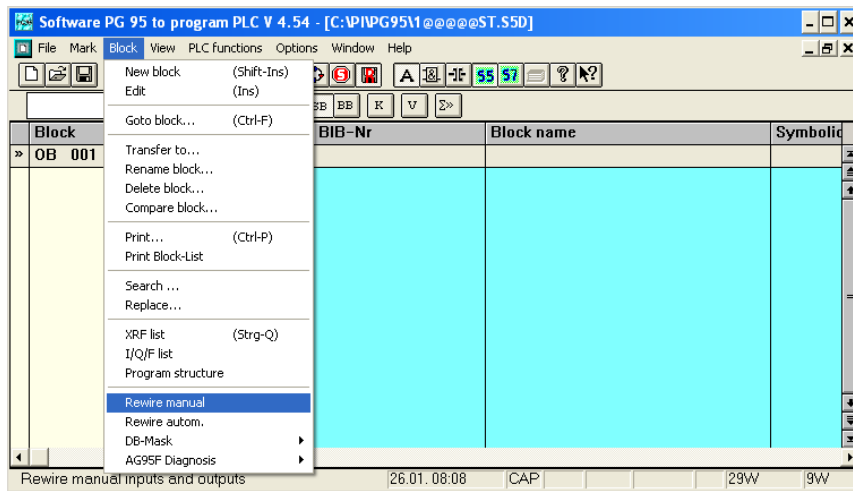
268



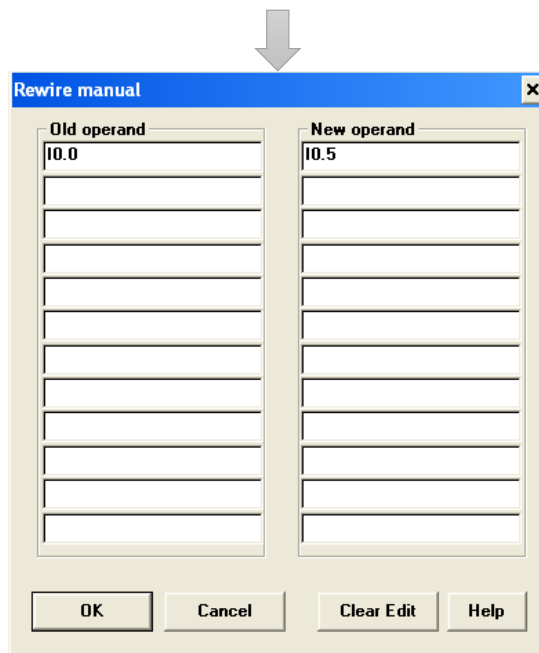
✓ مشاهده ساختار بلوک های پروژه



✓ جایگزین کردن آدرس جدید به جای آدرس قدیم به یکباره در کل پروژه



269



❖ آکبر اویسی فر
❖ کارشناس ارشد برق-الکترونیک
❖ متخصص سیستم های اتوماسیون صنعتی
❖ ایمیل: Akb_Oveisifar@yahoo.com

ردیف	عنوان
۱	تسلط بر PLC های S5 ، S7-200 ، S7-300 ، S7-400 ، S7-1200 ، S7-1500 (برنامه نویسی و عیب یابی)
۲	تسلط بر PLC های شرکت Allen Bradley (CompactLogix ، ControlLogix ، SLC 500) STUDIO 5000
۳	تسلط بر سیستم DCS شرکت زیمنس (PCS7) S7-400FH ، S7-400H
۴	تسلط بر سیستم های مانیتورینگ و طراحی و راه اندازی آنها توسط نرم افزارهای WinCC ، WinCC Flexible و Protocol
۵	تسلط بر شبکه های صنعتی Profibus ، Profinet و Ethernet (نصب ، برنامه نویسی و عیب یابی)
۶	تسلط بر نصب ، برنامه نویسی و راه اندازی ET های زیمنس در شبکه Profibus و Profinet
۷	تسلط بر نصب ، برنامه نویسی و راه اندازی درایوهای ABB ، SIEMENS ، SEW و LENZE در شبکه Profibus
۸	تسلط بر سرو درایوهای YASKAWA ، SIEMENS ، MITSUBISHI
۹	تسلط کامل بر نرم افزارهای طراحی تابلوهای فرمان ، قدرت و کنترل (EPLAN ، CAD)
۱۰	تسلط کامل بر تجهیزات ابزار دقیق (سنسورهای دما، فشار، لودسل، کنترلر سطح و...)، نصب و برنامه نویسی توسط PLC
۱۱	تسلط بر طراحی، اجراء و عیب یابی مدارات پنیوماتیک ، هیدرولیک
۱۲	توانایی راه اندازی انواع لوپ های کنترلی PID توسط PLC
۱۳	توانایی از بین بردن پسروردهای نرم افزاری و سخت افزاری PLC
۱۴	توانایی برنامه نویسی و طراحی بردهای صنعتی توسط میکروکنترلرهای AVR و ARM ، FPGA
۱۵	تسلط بر برنامه نویسی تراشه های FPGA و کار با نرم افزارهای ISE ، XILINX ، QUARTUS و ModelSim

برخی از کتاب های تالیف شده





نوین فن آوران آریا

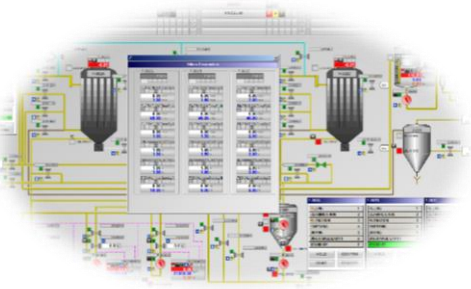
(مشاور، طراح و مجری پروژه های اتوماسیون صنعتی)

Consulting, designing and executing of Industrial Automation Projects

شرکت نوین فن آوران آریا مجری کلیه پروژه های اتوماسیون صنعتی، ابزار دقیق و اسکادا در سطح کشور می باشد. این شرکت با اتکاء به تجربه ۲۰ ساله موسسین در زمینه ساخت تابلوهای کنترل، تابلوهای محلی RIO، تابلوهای فشار ضعیف و متوسط و همچنین طراحی و اجرای پروژه های اتوماسیون صنعتی و ابزار دقیق، بنیان نهاده شده است. تیم اجرایی این شرکت با بهره گیری از کادر مهندسی توانمند و با تجربه، توانایی اجرای کلیه پروژه های EPC در صنایع نفت و گاز، خودروسازی، شیشه، صنایع غذایی و ... را به صورت کاملاً استاندارد دارا می باشد.



- طراحی و ساخت تابلوهای توزیع، قدرت و کنترل
- طراحی و برنامه نویسی PLC و DCS مطابق استانداردهای روز دنیا
- طراحی و برنامه نویسی سیستم های Fail Safe و Redundant
- طراحی سیستم های SCADA کاملاً حرفه ای
- طراحی و راه اندازی شبکه های صنعتی در سطوح مختلف
- نصب و راه اندازی انواع درایوهای صنعتی
- تعمیرات تخصصی بردهای صنعتی
- ارائه خدمات مشاوره، بهینه سازی خطوط تولید و نگهداری
- تامین کننده تجهیزات اتوماسیون صنعتی و ابزار دقیق با قیمت رقابتی
- آموزش دوره های تخصصی اتوماسیون صنعتی
- تالیف و چاپ کتاب های تخصصی اتوماسیون صنعتی





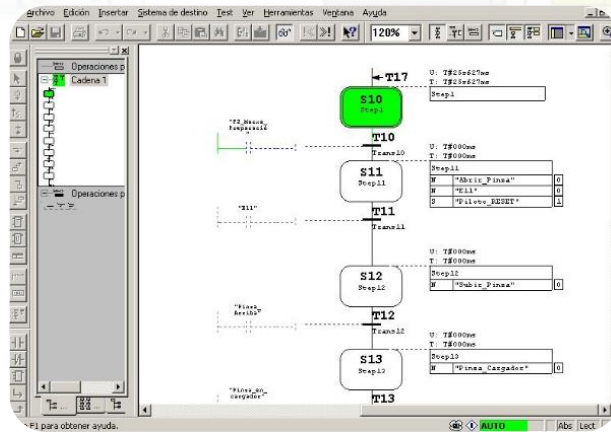
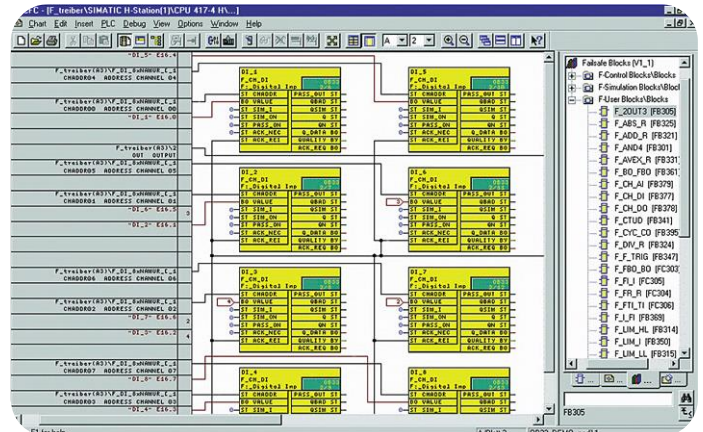
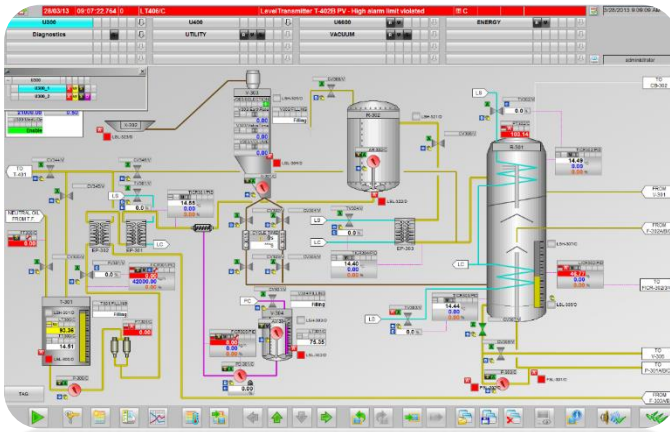
شرکت نوین فن آوران آریا
Novin Fanavaran Aria Co.

شرکت فنی و مهندسی نوین فن آوران آریا

(مشاور، طراح و مجری پروژه های برق، اتوماسیون صنعتی و ابزار دقیق)

❖ طراحی سیستم های مانیتورینگ و برنامه نویسی PLC

در جهت تسهیل استفاده و نظارت بهره برداران و اپراتورها در سایت ها و پلنت های تولید مراکز صنعتی، این شرکت با توجه به درخواست کارفرما، توانایی طراحی سیستم های اسکادا مبتنی بر PC در سایت و همچنین Panel در فیلد را دارا می باشد. سیستم های مانیتورینگ، مطابق استانداردهای روز دنیا و به صورت کامل کاربرپسند طراحی و ارائه می شود. تیم برنامه نویسی این شرکت نیز با استفاده از جدیدترین متدهای برنامه نویسی توسط زبان های سطح بالا، لاجیک های تحلیل شده را در قالب توابع استاندارد، جهت کنترل پروسه های صنعتی توسط PLC ارائه می کنند.





شرکت نوین فن آوران آریا
Novin Fanavaran Aria Co.

شرکت فنی و مهندسی نوین فن آوران آریا

(مشاور، طراح و مجری پروژه های برق، اتوماسیون صنعتی و ابزار دقیق)

❖ طراحی شبکه های صنعتی در سطوح مختلف

در عرصه سیستم های کنترل صنعتی سالهاست که بحث شبکه های صنعتی مطرح و روز به روز هم در حال توسعه می باشد. تیم اجرایی این شرکت در همین راستا قادر است، بحث شبکه را در سطوح مختلف یک پروژه طراحی و اجرا نماید.

طراحی و اجرای شبکه های ASI, CAN, Modbus, Profinet, Ethernet, Profibus

کنترل درایوها و تجهیزات ابزار دقیق توسط شبکه های صنعتی

راه اندازی سایت های کنترل و طراحی سیستم های Client/Server

طراحی و اجرای سیستم های Distributed I/O

طراحی و اجرای شبکه های صنعتی بی سیم

طراحی و اجرای شبکه های صنعتی توسط فیبر نوری



❖ تعمیرات تخصصی بردهای صنعتی

از دیگر فعالیت های این شرکت، تعمیر تخصصی کلیه بردهای صنعتی می باشد. واحد تعمیرات این شرکت متشکل از کارشناسان الکترونیک بوده که سال ها در زمینه طراحی و تعمیر بردهای الکترونیک فعالیت مستمر داشته اند.



تعمیر تخصصی کلیه ماژول های PLC

تعمیر تخصصی پانل های صنعتی HMI

تعمیر تخصصی کلیه درایوها و بردهای الکترونیکی

تعمیر تخصصی ماژول های CNC

❖ آموزش دوره های تخصصی اتوماسیون صنعتی

واحد آموزش این شرکت با در اختیار داشتن اساتید با تجربه در عرصه علم و صنعت، اقدام به برگزاری دوره های تخصصی اتوماسیون صنعتی SIEMENS و Allen Bradley نموده است. تمامی دوره های این شرکت از لحاظ محتوا و کار عملی مطابق با استاندارد بوده و در هر یک از دوره ها شرکت کننده ها به طور کاملا تخصصی با سیستم های کنترلی آشنا می شوند. موسسین این شرکت در سال های فعالیت خود، افتخار برگزاری دوره های آموزشی برای بسیاری از مهندسین و تکنسینهای شرکت ها و صنایع مختلف در سطح کشور را دارا می باشند. مدرسین این شرکت مورد تایید شرکت نفت، گاز و صنایع خودرو سازی می باشند.



Studio 5000-ControlLogix

Studio 5000_PanelView 5000

RSLogix500_SLC500

Factory Talk View Studio

PLC- S5

S7-400H/FH

PCS7

S7-300,400

S7-400H/FH

S7-1200(TIA)

S7-300,400(TIA)

WinCC

Protool

WinCC Flexible

Ethernet و Profibus

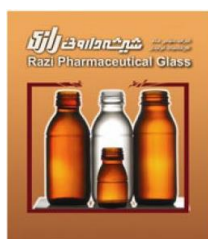
SINAMICS Servo Drive

Micromaster Drive

Simotion Scout

S7 PID

Eplan





شرکت نوین فن آوران آریا
Novin Fanavaran Aria Co.

شرکت فنی و مهندسی نوین فن آوران آریا

(مشاور، طراح و مجری پروژه های برق، اتوماسیون صنعتی و ابزار دقیق)

❖ تالیف و چاپ کتاب های تخصصی اتوماسیون صنعتی

این شرکت جهت پیشبرد اهداف آموزشی خود اقدام به چاپ کتاب های تخصصی اتوماسیون و همچنین توزیع سی دی های آموزشی جهت استفاده مهندسين و دانشجویان نموده است. برخی از کتاب های تالیف شده توسط دپارتمان آموزش این شرکت به شرح زیر می باشند.

جلد دوم
SIMATIC Controller

مرجع کاربردی PLC SIMATIC S7-300,400 (نرم افزار)

آموزش جامع:
- نرم افزار SIMATIC MANAGER
- دستورات کاربردی و بلوک های برنامه
- تکنیک ها و ابزارهای عیب یابی در نرم افزار
- پردازش و برنامه نویسی سینکرونال آنالوگ

نویسنده: اکبر اویسی فر

SABCO
Savant Automation Control
Systems & Electronic Industries

آموزش برنامه نویسی PLC به زبان S7-GRAPH

مهندس اکبر اویسی فر
مهندس نازنین زنجیریان

SIEMENS
Ingenuity for Life

جلد اول
SIMATIC Controller

مرجع کاربردی PLC SIMATIC S7-300,400 (سخت افزار)

آموزش جامع:
- سخت افزار کلیه PLC های زیمنس
- شناخت و کار با E:T های زیمنس
- شبکه های صنعتی Profibus
- شناخت کلیه محصولات HMI زیمنس

مؤلف: اکبر اویسی فر

SIEMENS

جلد اول
Controller

آموزش سریع مانیتورینگ با SIMATIC WinCC 6.0

آموزش سریع مانیتورینگ با SIMATIC WinCC 6.0

نویسنده: اکبر اویسی فر

آموزش درایوهای MICROMASTER

- آشنایی با درایوهای میکرومستر
- بررسی ساخت فیلتر و موزون های مختلف
- بررسی پارامترهای درایو
- بررسی نحوه کنترل درایو در شبکه پرمیسیو

مهندس اکبر اویسی فر
مهندس نازنین زنجیریان

PLC S7-300.400 LEVEL(I)

مهندس نازنین زنجیریان

SIEMENS