جزوه آموزشی برنامه نویسی PLC

با درود،

از اینکه PLC برند LS را انتخاب کردید سپاسگزاریم.

کتابچه راهنمای کاربر، نحوه استفاده صحیح از این محصول و مواردی که باید احتیاط کرد را شرح می‌دهد.

رعایت نکردن دستورالعمل‌های ذکر شده در این راهنما ممکن است باعث آسیب شخصی یا آسیب به محصول شود.

قبل از استفاده از این محصول حتما این دفترچه راهنما را به دقت بخوانید و تمام دستورالعمل‌های موجود در آن را دنبال کنید.

محتویات این دفترچه راهنما طبق نسخه‌ی فعلی PLC (سخت‌افزاری/نرم‌افزاری)آماده گردیده است.

این دفترچه به اهتمام پرسنل ایمن تابلو نماینده انحصاری محصولات LS کره جنوبی در ایران، تهیه شده است.

فهرست

نکات ایمنی ۸

مقدمه ۱۰

انواع PLC ۱۱

Mini PLC ۱۱

Compact ۱۱

modular ۱۲

PLCهای دارای رک(Rack) ۱۲

باس یا رک ۱۳

ماژول منبع تغذیه ۱۳

ماژول‌های ورودی/خروجی ۱۴

مزایای PLC نسبت به مدارات فرمان رله کنتاکتوری ۱۴

برنامه‌نویسیPLC ۱۵

نحوه نصب نرم‌افزار XG5000 ..... ۱۶

کدهای PLCها و ماژول‌های سری XGB ۱۹

کد شناسایی PLC ۱۹

کد شناسایی کارت‌های ورودی/خروجی ۱۹

کد شناسایی کارت‌های جانبی(آنالوگ/ دما/ موقعیت) ۲۰

کد شناسایی کارت‌های شبکه ۲۰

ساختمان داده ۲۱

متغیرها ۲۲

شیوه آدرس‌دهی مستقیم ۲۳

انواع متغیر ۲۴

متغیرهای عمومی (محلیLocal) ۲۴

متغیر همگانی(جهانیGlobal) ۲۴

متغیرهای Retain (ماندگار) ۲۴

متغیرهای ثابت ۲۴

نحوه اتصال ورودی/خروجی ۲۴

ورودی نوع PNP یا Source ۲۴

ورودی نوع NPN یا Sink ۲۵
خروجی NPN ۲۵
خروجی PNP ۲۵
خروجی رله‌ای ۲۶
برنامه۱: قطع و وصل یک لامپ با شستی(روشن شدن لحظه‌ای لامپ) ۲۶
ایجاد پروژه ۲۷
شبیه‌ساز(Simulator) ۳۳
استپ/استارت با یک شستی ۴۰
لبه بالارونده (Positive transition-sensing contact) ۴۰
انتخاب توابع ۴۱
تابع SR ۴۳
تابع RS ۴۳
شمارنده‌ها (Counters) ۴۵
شمارنده صعودیCTU ۴۶
ساخت شمارنده با استفاده از جمع‌کننده ۴۸
شمارنده نزولیCTD ۴۸
شمارنده صعودی/نزولیCTD ۴۹
شمارنده حلقوی ۵۰
مقایسه‌گرها ۵۱
تابع بزرگتر بودن ۵۱
تابع بزرگتر یا مساوی بودن ۵۱
تابع کوچکتر بودن ۵۲
تابع کوچکتر یا مساوی بودن ۵۲
تابع مساوی بودن ۵۳
تابع نامساوی بودن ۵۳
تابع MOVE ۵۳
عملگرهای ریاضی ۵۴
تابع جمع‌کننده (ADD) : ۵۴
تابع ضرب (MUL) : ۵۴
تابع تقسیم (DIV) ۵۵

تابع تفریق (SUB) :..... ۵۵
تابع قدر مطلق (ABS) :..... ۵۵
تابع باقیمانده خارج قسمت (MOD) :..... ۵۶
تابع جذر (SQRT) : ۵۶
تابع توان (EXPT) :..... ۵۶
عملگرهای منطقی..... ۵۷
عملگر منطقی AND :..... ۵۷
عملگر منطقی OR :..... ۵۷
عملگر منطقی NOT :..... ۵۸
عملگر منطقی XOR :..... ۵۸
توابع مربوط به String :..... ۵۹
تابع LEN..... ۵۹
تابع CONCAT..... ۵۹
تابع INSERT..... ۵۹
تابع DELETE..... ۶۰
تابع REPLACE..... ۶۰
تابع RIGHT..... ۶۰
تابع LEFT..... ۶۱
تابع MID..... ۶۱
مالتی‌پلکسر (MUX)..... ۶۱
تبدیل داده‌ها..... ۶۲
تبدیل زمان روز به داده نوع UDINT..... ۶۲
تابع تبدیل تعداد روزهای سپری شده به UINT..... ۶۴
تابع تبدیل String به Date..... ۶۴
تابع تبدیل BOOL-TO-INT..... ۶۵
تابع BSUM..... ۶۶
تابع Shift..... ۶۶
تابع Word-Byte..... ۶۷
تنظیم ساعت و تاریخ PLC از طریق HMI..... ۶۸
ساخت فانکشن بلاک..... ۶۹

تفاوت Function با Function Block ۷۴
تبدیل کد گری به دسیمال ۷۴
تایمرها ۸۳
تایمر تاخیر در وصل(TON) : ۸۴
تایمر تاخیر در قطع (TOF) ۸۵
تایمر ضربه (TP) ۸۵
ساخت تایمر پالس ۸۷
تبدیل تاریخ شمسی به میلادی ۸۸
تایمر پالس دارای پایه ریست ۹۱
تایمر پالس UINT ۹۱
تایمر TMR ۹۲
تایمر TMR_FLK ۹۲
تایمر TMR_UINT ۹۳
حافظه Stack ۹۳
پیوست۱ ..... ۹۵
سخت‌افزار PLCهای برند LS ۹۵
ماژول‌های PLCهای مختلف ۹۷
سری XGI/XGK ۹۸
پیوست۲ ..... ۱۰۷
روش اتصال ورودی/ خروجی‌های PLCهای سری Compact ۱۰۷
ورودیPLC سری XEC-D...32H ۱۰۸
خروجیPLC سری XEC-DR32H ۱۰۸
خروجیPLC سری XEC-DN32H ۱۰۹
خروجیPLC سری XEC-DP32H ۱۰۹
ورودیPLC سری XEC-D...32H ۱۱۰
خروجیPLC سری XEC-DR64H ۱۱۱
خروجیPLC سری XEC-DN64H ۱۱۲
خروجیPLC سری XEC-DP64H ۱۱۳
ورودیPLC سری XEC-D...10E ۱۱۴
خروجیPLC سری XEC-DR10E ۱۱۴

خروجیPLC سری XEC-DN10E ۱۱۵
خروجیPLC سری XEC-DP10E ۱۱۵
ورودیPLC سری XEC-D...14E ۱۱۶
خروجیPLC سری XEC-DR14E ۱۱۶
خروجیPLC سری XEC-DN14E ۱۱۷
خروجیPLC سری XEC-DP14E ۱۱۷
ورودیPLC سری XEC-D...20E ۱۱۸
خروجیPLC سری XEC-DR20E ۱۱۸
خروجیPLC سری XEC-DN20E ۱۱۹
خروجیPLC سری XEC-DP20E ۱۱۹
ورودیPLC سری XEC-D...30E ۱۲۰
خروجیPLC سری XEC-DR30E ۱۲۰
خروجیPLC سری XEC-DN30E ۱۲۱
خروجیPLC سری XEC-DP30E ۱۲۱
ورودیPLC سری XEC-DN20SU, XEC-DR20SU ۱۲۲
خروجیPLC سری XEC-DN20SU ۱۲۲
خروجیPLC سری XEC-DR20SU ۱۲۳
ورودیPLC سری XEC-DN30SU, XEC-DR30SU ۱۲۳
خروجیPLC سری XEC-DR30SU ۱۲۴
خروجیPLC سری XEC-DN30SU ۱۲۴
ورودیPLC سری XEC-DN40SU, XEC-DR40SU ۱۲۵
خروجیPLC سری XEC-DN40SU ۱۲۶
خروجیPLC سری XEC-DR40SU ۱۲۷
ورودیPLC سری XEC-DN60SU, XEC-DR60SU ۱۲۸
خروجی PLC سری XEC-DN60SU ۱۲۹
خروجی PLC سری XEC-DR60SU ۱۳۰
پیوست۳ ۱۳۱
جدول اخطارها ۱۳۱
خطاهای شمارنده سرعت بالا ۱۳۲
جدول خطاهای PID ۱۳۳

نکات ایمنی

در دفترچه راهنمای هر دستگاهی به دو کلیدواژه Warning و Caution دقت کنید. معنای هرکدام بصورت زیر است.

Warning: این نماد احتمال آسیب جدی یا مرگ را در صورت نقض برخی دستورالعمل‌های قابل اجرا نشان می‌دهد.

Caution: این نماد احتمال آسیب جزئی یا آسیب به محصولات را در صورت نقض برخی دستورالعمل‌های قابل اجرا نشان می‌دهد.

ممکن است شوک الکتریکی رخ دهد.

تجهیزات حفاظتی را قبل از PLC قرار دهید و در خروجیPLC تجهیز حفاظتی قرار ندهید تا کل PLC حفاظت شود.

هرگز بیش از حد مجاز به خروجی بار وصل نکنید و اجازه ندهید مدار خروجی دارای اتصال کوتاه شود که ممکن است باعث آتش سوزی شود.

هرگز اجازه ندهید برق خروجیPLC مدار طوری طراحی شود که زودتر از برق PLC روشن شود، که ممکن است باعث خروجی یا عملکرد غیرعادی شود.

سیگنال ورودی/خروجی یا خط ارتباطی باید حداقل ۱۰۰ میلیمتر از کابل ولتاژ بالا یا خط برق سیم‌کشی شود. اگر نه، ممکن است باعث خروجی یا عملکرد غیرعادی شود.

قبل از نصب ماژول، مطمئن شوید که برق PLC خاموش است. در غیر این صورت ممکن است برق گرفتگی یا آسیب به محصول وارد شود.

مطمئن شوید که اتصال ورودی/خروجی یا کارت توسعه به درستی ایمن شده باشد. در غیر این صورت ممکن است شوک الکتریکی، آتش سوزی یا عملکرد غیرعادی ایجاد شود.

اگر ارتعاش زیادی در محیط نصب انتظار می‌رود، اجازه ندهید PLC مستقیماً ارتعاش کند. ممکن است برق‌گرفتگی، آتش سوزی یا عملکرد غیرعادی ایجاد شود.

مواد خارجی فلزی را داخل محصول نگذارید، که ممکن است باعث برق گرفتگی، آتش سوزی یا عملکرد غیرعادی شود.

قبل از سیم کشی، مطمئن شوید که برق PLC و برق خروجی خاموش است. در غیر این صورت ممکن است برق گرفتگی یا آسیب به محصول وارد شود.

قبل از روشن شدن سیستم PLC مطمئن شوید که تمام درپوش های ترمینال به طور ایمن بسته شده‌اند. در غیر این صورت ممکن است شوک الکتریکی ایجاد شود.

هنگام سیم‌کشی، پیچ های ترمینال‌ها را با گشتاور مشخص محکم کنید. اگر پیچ ترمینال ها شل شوند، ممکن است اتصال کوتاه، آتش سوزی یا عملکرد غیرعادی ایجاد شود.

حتماً از سیم زمین کلاس ۳ برای ترمینال‌های FG استفاده کنید که منحصراً برای PLC استفاده می‌شود. اگر پایانه ها به درستی زمین نشوند، ممکن است عملکرد غیرعادی ایجاد شود.

هنگام روشن شدن برق، ترمینال را لمس نکنید. ممکن است برق گرفتگی یا عملکرد غیرعادی رخ دهد.

قبل از تمیز کردن یا سفت کردن پیچ های ترمینال، اجازه دهید برق PLC قطع شود. در غیر این صورت، برق گرفتگی یا عملکرد غیرعادی ممکن است رخ دهد.

اجازه ندهید باتری دوباره شارژ شود، جدا شود، گرم شود، اتصال کوتاه یا لحیم شود. گرما، انفجار یا اشتعال ممکن است باعث جراحت یا آتش سوزی شود.

PCB را از جعبه ماژول خارج نکنید و ماژول را تغییر ندهید.

بی‌سیم یا تلفن همراه را حداقل ۳۰ سانتی متر از PLC دور نگه دارید.

مقدمه

PLC (Programmable Logic Controller) یا کنترل‌کننده منطقی برنامه‌پذیر، یک رایانه دیجیتال صنعتی است که برای استفاده در کاربردهای صنعتی استحکام بخشیده شده‌است و در فرایندهای صنعتی مانند کنترل فرایندهای تولید، کنترل خطوط مونتاژ، کنترل دستگاه‌های رباتیک یا هر فرایندی که نیاز به کنترل دقیق و قابل اطمینان داشته باشد از آن استفاده می‌شود تا عیب‌یابی آن نیز ساده باشد. PLC ها را می‌توان از نظر اندازه، حافظه، تعداد ورودی/خروجی، وسعت عملیات (محلی یا وسیع) و ... دسته‌بندی نمود. باید توجه داشت که برای ارزیابی قابلیت یک PLC باید ویژگی‌های دیگری مانند پردازنده، زمان اجرای یک سیکل، سادگی زبان برنامه‌نویسی، قابلیت توسعه و ... را در نظر داشت. PLCهای ماژولار دارای یک اسکلت (یا رک) هستند که واحدهای مختلف PLC با توجه به نیاز بر روی آن قرار می‌گیرند. در این PLC ها پردازنده و ماژول‌های ورودی/خروجی می‌توانند با توجه به کاربرد مدنظر انتخاب شوند. همچنین این امکان وجود دارد که چندین رک (Rack)یا تعداد خیلی بسیاری از ورودی/خروجی‌ها به وسیله یک پردازنده کنترل شوند. هر سیستم PLC حداقل به سه ماژول زیر نیاز دارد: ماژول پردازنده، ماژول منبع تغذیه، یک یا چند ماژول ورودی/خروجی.

PLC ها مشخصه های فراوانی دارند ولی پرکاربردترین آنها عبارتند از:

۱. تعداد ورودی و خروجی دیجیتال و آنالوگ
۲. تعداد تایمر
۳. تعداد شمارنده
٤. تعداد رله های کمکی
٥. حجم حافظه قابل برنامه ریزی
٦. سرعت اجرای دستورات

معمولا PLCها دارای پورت های ارتباطی صنعتی هستند که به کمک آن می توانند با سایر دستگاه ها ارتباط برقرار کنند. این امر باعث تسریع در امر تبادل اطلاعات می گردد و امکان کنترل دقیق تر را به PLC می‌دهد.

انواع PLC

Mini PLC

در عملکرد مانند پی ال سی های بزرگ است اما تعداد وروردی و خروجی کمتری دارند (نهایتاً ۴۰ ورودی خروجی) و در پروژه های کوچک مورد استفاده قرار می‌گیرد. در اغلب این PLC ها یک صفحه کوچک با تعدادی دکمه وجود دارد که علاوه بر برنامه‌ریزی با کامپیوتر، توسط همین صفحه کوچک و دکمه هم می‌توان به PLC برنامه داد. شرکت های سازنده مختلف این PLCهای کوچک را با اسم های متفاوتی نام می‌برند مثلا در شرکت اُمرون به این مدل پی ال سی ها زِن می‌گویند. این mini PLC در برند زیمنس به LOGO معروف است.

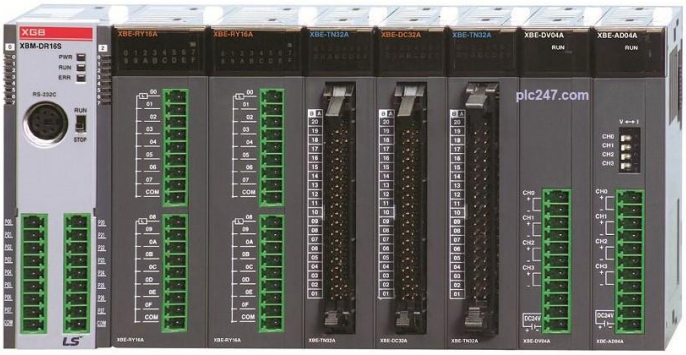
Compact

در این نوع PLC ها برخلاف مدل ماژولار و Rack type هر سه بخش تغذیه، CPU یا پردازنده و بخش ورودی خروجی ها داخل یک باکس قرار دارد. شرکت‌های سازنده PLC، در مدل‌های گوناگون این نوع PLC را ساخته‌اند که برخی از آنها دارای امکانات سخت افزاری و نرم افزاری بیشتری هستند.

modular

این نوع PLC ها به صورت ماژول ماژول می‌باشند، به این صورت که سه بخش اصلی آن یعنی منبع تغذیه، CPU، ورودی/خروجی از هم جدا بوده و می‌توانیم ماژول‌های تغذیه،CPU و ورودی/ خروجی‌ها را در کنار هم قرار داده تا تشکیل یک PLC بدهند. این نوع PLC ها نیز از کارت‌های سخت افزاری مختلف پشتیبانی می‌کنند. PLC های ماژولار مزیتی که نسبت به دو مدل دیگر دارند این است که می‌توانیم چندین ماژول ورودی خروجی را به آن اضافه کنیم. این که چه مقدار ورودی خروجی را می‌توانیم به این نوع PLC ها اضافه کنیم به قدرت پردازنده بستگی دارد ولی نهایتاً تا ۲۰۰۰ ورودی خروجی را می‌توانیم به آن اضافه کنیم. این نوع PLC نیز بر روی ریل قرار می‌گیرد.

نکته :در انتهای این نوع PLC ها یک قاب انتهایی می‌بایست بسته شود تا PLC تکمیل شود و در غیر این صورت PLC کار نخواهد کرد.

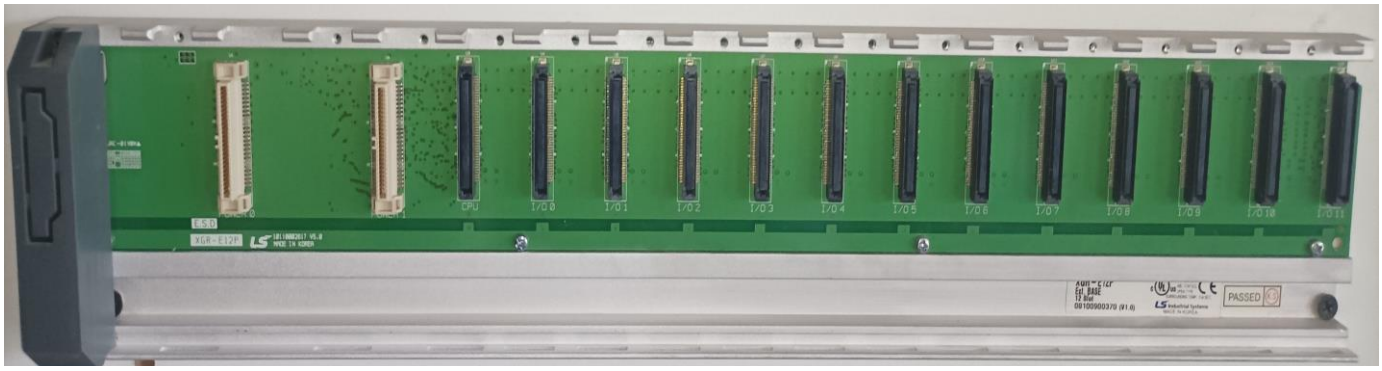


PLCهای دارای رک (Rack)

این نوع PLC ها مانند PLC های ماژولار است با این تفاوت که پشت آن یک ماژول مرکزی وجود دارد که رابط تمام ماژول‌ها و ورودی/خروجی‌ها به هم است ولی در PLC های ماژولار تمام تجهیزات به هم وصل می‌شود. این PLCها برای پروژهای بزرگ که ورودی خروجی زیادی لازم دارد مورد استفاده قرار می‌گیرند و می‌توانیم تا ۵۰۰۰ ورودی خروجی با این PLC ها داشته باشیم.

باس یا رک

در PLCهای ماژولار در پنل پشتی مدار رک‌هایی در نظر گرفته شده‌اند که تمام ماژول‌ها مانند پردازنده و دیگر ماژول‌های ورودی یا خروجی به اسلات‌های آن نصب می‌شود. این باس‌ها ارتباط بین پردازنده و ماژول‌های ورودی/خروجی را برای ارسال و دریافت داده‌ها فراهم می‌آورد. این ارتباط با آدرس‌دهی ماژول‌های ورودی/ خروجی مطابق با قرارگیری ماژول پردازنده در رک ایجاد می‌شود.



ماژول منبع تغذیه

این ماژول قدرت مورد نیاز کل سیستم را با تبدیل برق AC به برق DC که مورد نیاز پردازنده و ماژول‌های ورودی/خروجی است را فراهم می‌آورد. خروجی ۵ ولت DC به مدارات PLC، و برخی از PLCهای ۲۴ولت DC در رک، چند سنسور و محرک‌ها هدایت می‌شود.

ماژول‌های ورودی/خروجی

ماژول‌های ورودی/خروجی PLC امکان اتصال سنسورها و محرک‌ها به سیستم را فراهم می‌آورند تا به صورت بلادرنگ متغیرهایی مانند دما، فشار، جریان و ... نظارت و کنترل شوند. این ماژول‌های ورودی/ خروجی دارای نوع، دامنه و قابلیت‌های متفاوتی هستند و برخی از این موارد عبارتند از:

ماژول ورودی/خروجی دیجیتال: برای اتصال سنسورها و محرک‌هایی که به طور ذاتی دیجیتال هستند استفاده می‌شود، برای مثال فقط برای سوئیچ ON و OFF. این ماژول‌ها در هر دو ولتاژ AC و DC در دسترس هستند و دارای تعداد ورودی‌ها و خروجی‌های دیجیتال متغیری هستند.

ماژول ورودی/خروجی آنالوگ: برای اتصال سنسورها و محرک‌هایی که سیگنال‌های الکتریکی آنالوگ را فراهم می‌کنند، استفاده می‌شود. در داخل این ماژول، مبدل آنالوگ به دیجیتال و دیجیتال به آنالوگ برای تبدیل سیگنال‌های آنالوگ به اطلاعات قابل درک برای پردازنده استفاده می‌شود (به عنوان مثال داده‌های دیجیتال).

مزایای PLC نسبت به مدارات فرمان رله کنتاکتوری

۱. سیم بندی کمتر سیستم‌های جدید در مقایسه با سیستم‌های کنترل رله‌ای
۲. از آنجایی که پی ال سی توان بسیار کمی مصرف می‌کند، توان مصرفی به شدت کاهش پیدا خواهد کرد.
۳. توابع عیب یابی داخلی سیستمPLC ، تشخیص و عیب یابی سیستم را بسیار سریع و راحت می‌کند.
٤. برعکس سیستم‌های قدیمی در سیستم های کنترلی جدید اگر نیاز به تغییر در نحوه کنترل یا ترتیب مراحل آن داشته باشیم، بدون نیاز به تغییر سیم‌بندی و تنها با نوشتن چند خط برنامه این کار را انجام می‌دهیم. در نتیجه وقت و هزینه بسیار بسیار اندکی صرف انجام اینکار خواهد شد.
٥. در مقایسه با تابلوهای قدیمی در سیستم‌های مبتنی بر PLC نیاز به قطعات کمکی از قبیل رله، کانتر، تایمر، مبدل‌های A/D و D/A و ... بسیار کمتر شده است. همین امر نیز باعث شده در سیستم های جدید از سیم‌بندی، پیچیدگی و وزن تابلو به نحو چشمگیری کاسته شود.

٦. PLC ها استهلاک مکانیکی ندارند. بنابراین علاوه بر عمر بیشتر، نیازی به تعمیرات و سرویس‌های دوره‌ای نخواهد داشت.
٧. برخلاف مدارات رله کنتاکتوری، نویزهای الکترونیکی و صوتی ایجاد نمی‌کنند. از آنجایی که سرعت عملکرد و پاسخ دهی PLC در حدود میکروثانیه و نهایتا میلی‌ثانیه است. لذا زمان لازم برای انجام هر سیکل کاری ماشین بطور قابل ملاحظه‌ای کاهش یافته و این امر باعث افزایش میزان تولید و بالا رفتن بازدهی دستگاه می شود.
٨. ضریب اطمینان و درجه حفاظت این سیستم‌ها بسیار بالاتر از ماشین‌های رله‌ای است.
٩. وقتی توابع کنترل پیچیده‌تر و تعداد I/O ها خیلی زیاد باشد، جایگزین کردن PLC بسیار کم هزینه‌تر و راحت‌تر خواهد بود.

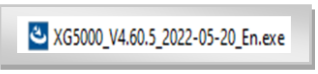
برنامه‌نویسی PLC

برای نمایش برنامه PLC از سه روش استفاده می‌شود:

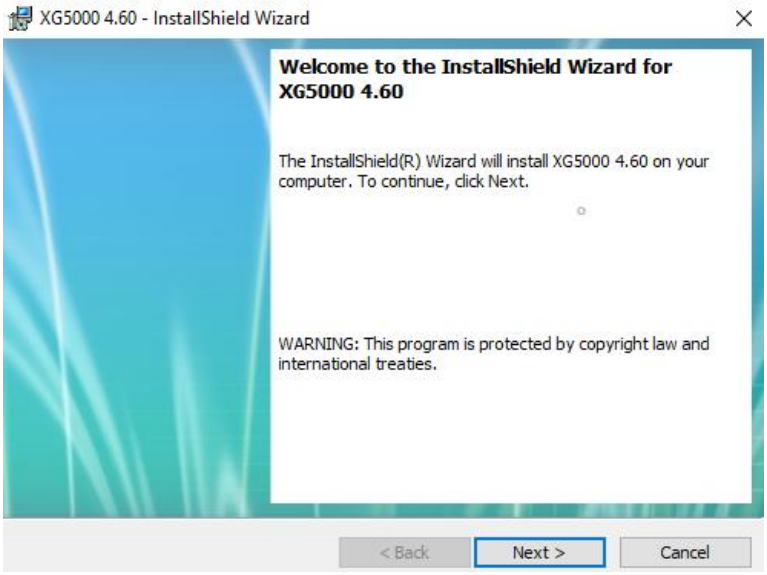
- در روش نردبانی ساختار برنامه شبیه مدارهای فرمان رله‌ای می‌باشد.
- در نمایش فلوچارتی از نمادهای مستطیلی استفاده می‌شود؛ و در هر مستطیل عمل منطقه‌ای نمایش داده می‌شود.
- در روش نمایش نوشتاری از دستورات و جملات کلیدی برای نوشتن برنامه استفاده می‌شود که در آن هر عبارت دارای دو بخش عملگرها و عملوندها می‌باشد.

PLC هر برندی نرم افزار مخصوص به خود را دارد و مثلا شما با نرم‌افزار زیمنس قادر نخواهید بود PLC برندی غیر از زیمنس را برنامه‌نویسی کنید. نرم‌افزار مربوط به PLC های مدلXGT برندLS ، XG5000 می‌باشد.

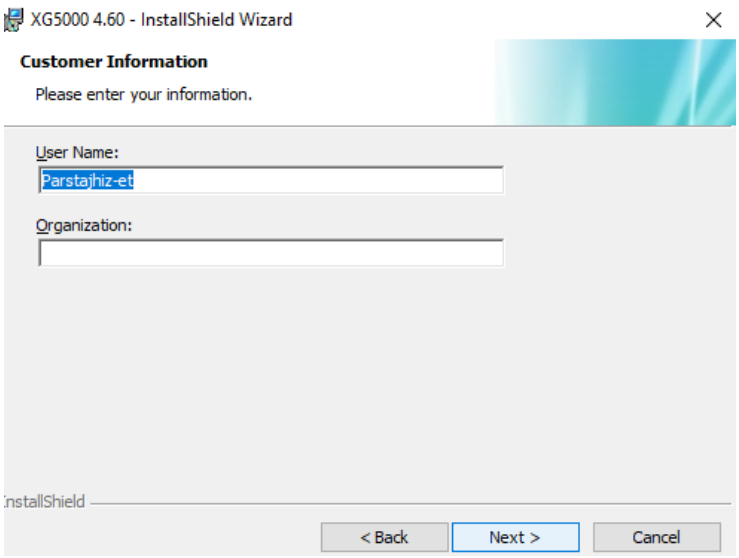
نحوه نصب نرم‌افزار XG5000



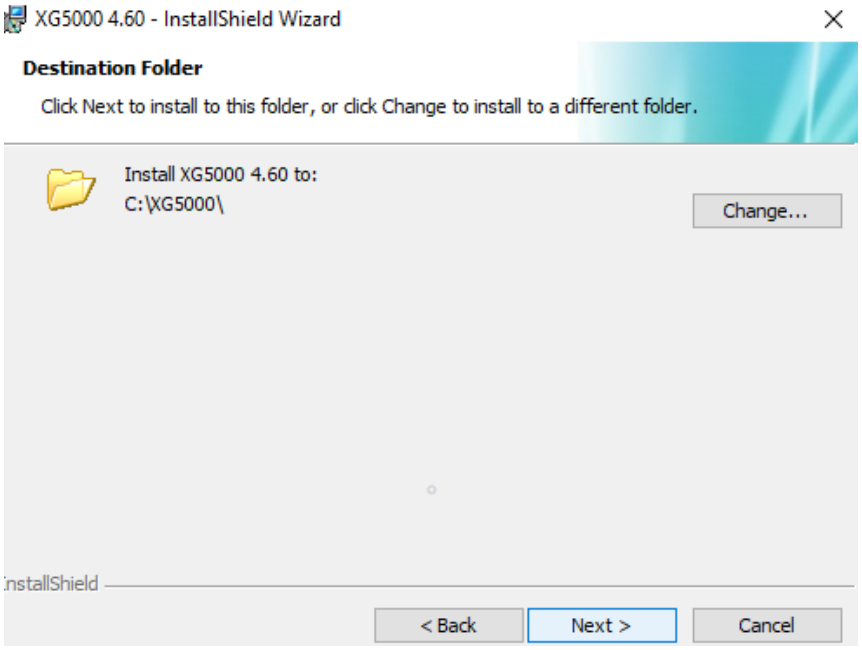
جهت نصب نرم‌افزار XG5000 ابتدا نرم افزار را از سایت دانلود نمایید سپس بر روی آیکون کلیک کنید در پنجره باز شده بر روی next کلیک کنید.



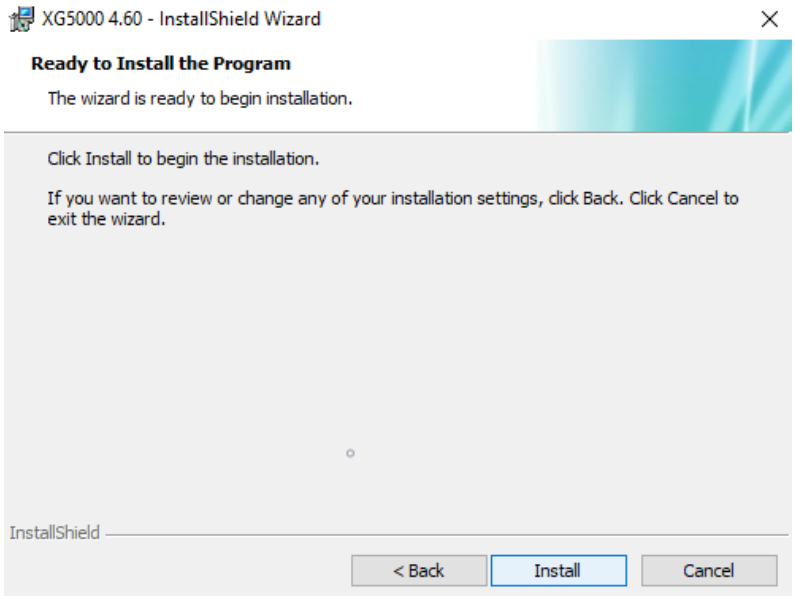
سپس پنجره زیر نمایش داده می‌شود سپس یک نام انتخاب نموده و بر روی next کلیک کنید.



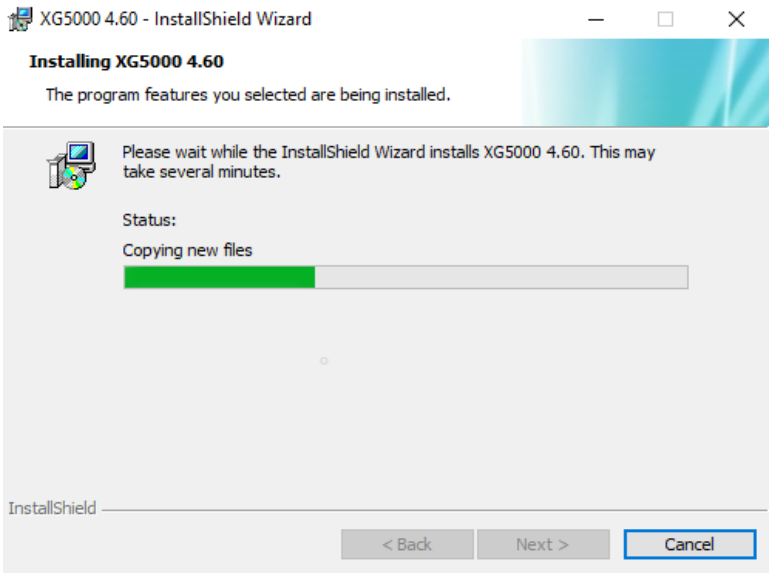
سپس پنجره زیر نمایش داده می‌شود با انتخاب change می‌توان محل نصب برنامه را تغییر داد بهتر است محل نصب را تغییر ندهید و بر روی next کلیک کنید.

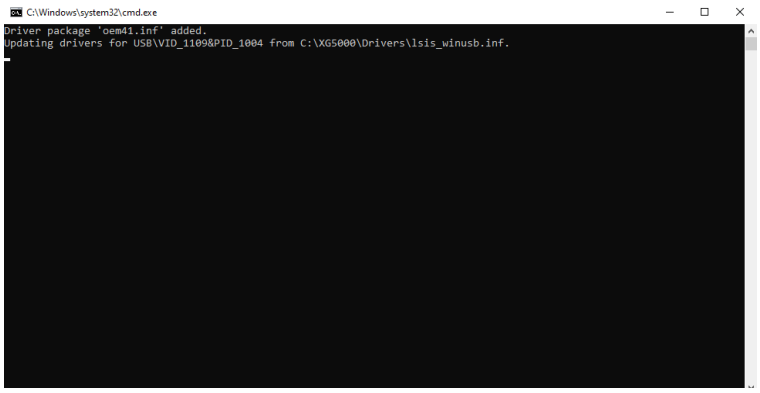


سپس پنجره زیر نمایش داده می‌شود بر روی install کلیک کنید.

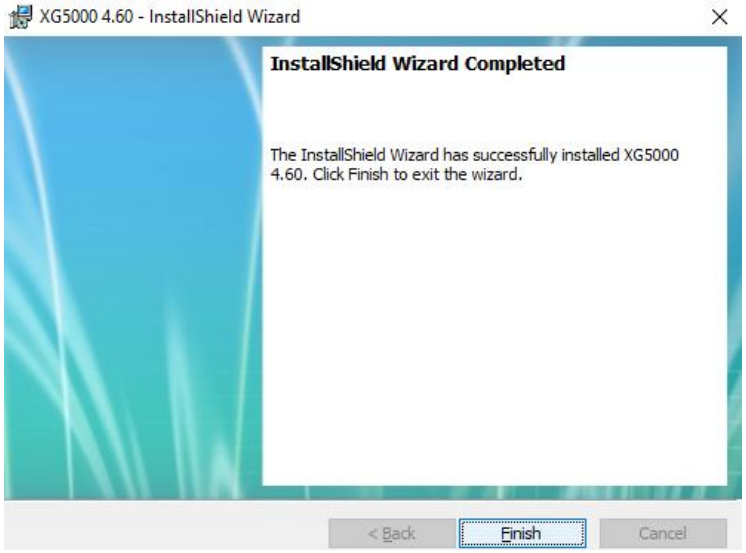


سپس پنجره زیر نمایش داده می‌شود. صبر کنید تا مراحل نصب پایان پذیرد.



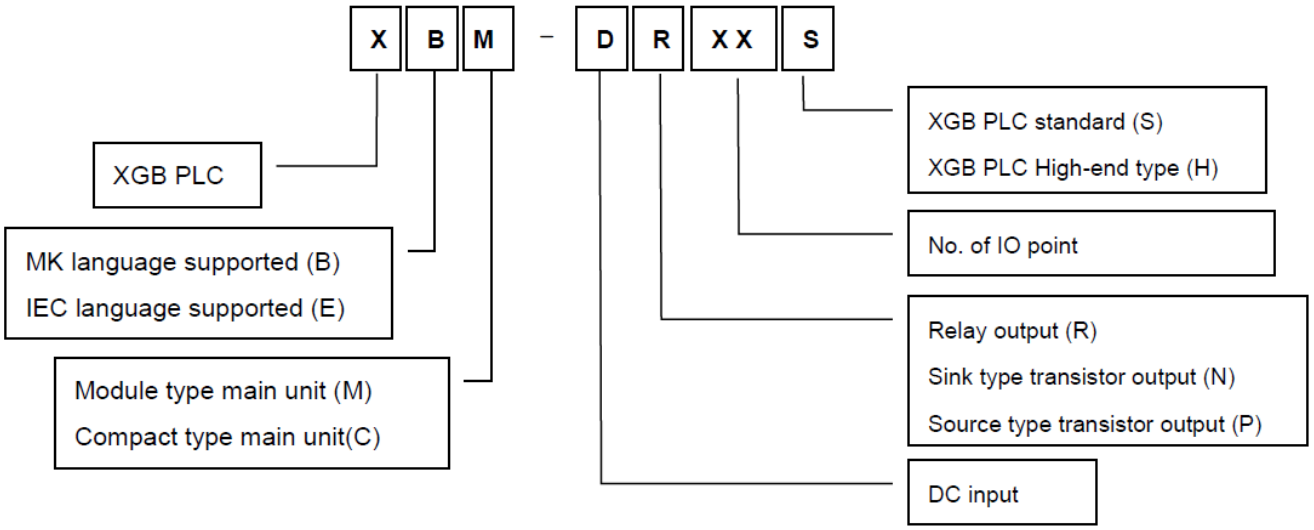


سپس در پنجره باز شده زیر بر روی finish کلیک کنید تا فرایند نصب پایان یابد.

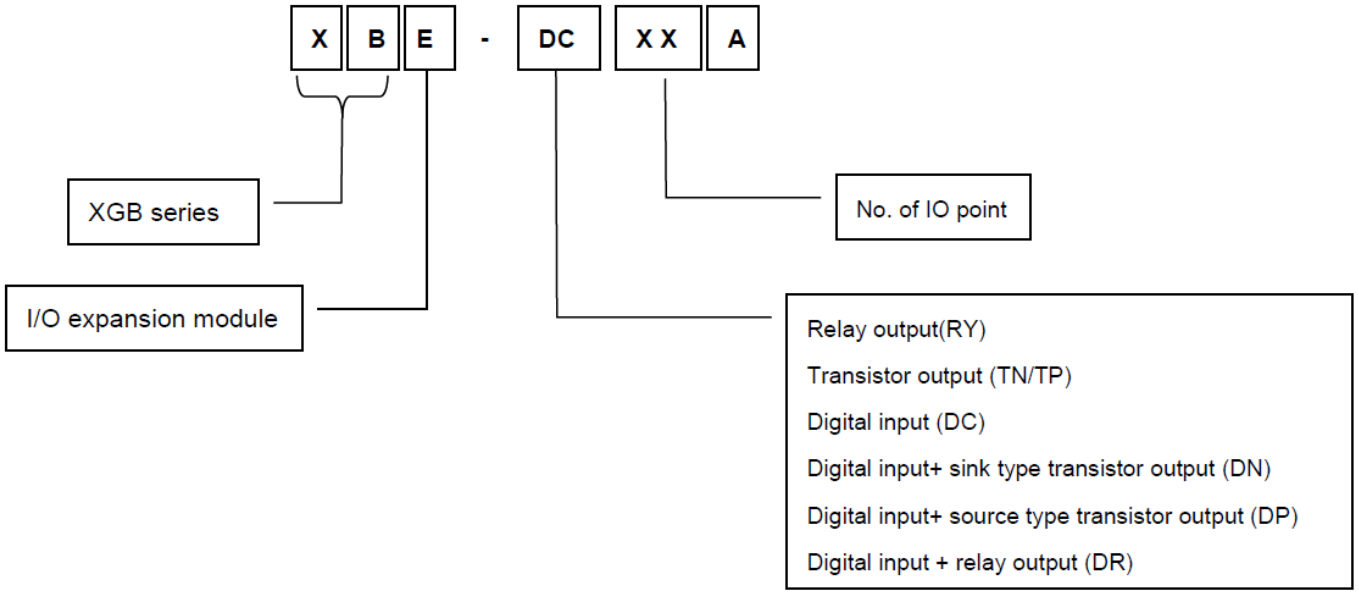


کدهای PLCها و ماژول‌های سری XGB

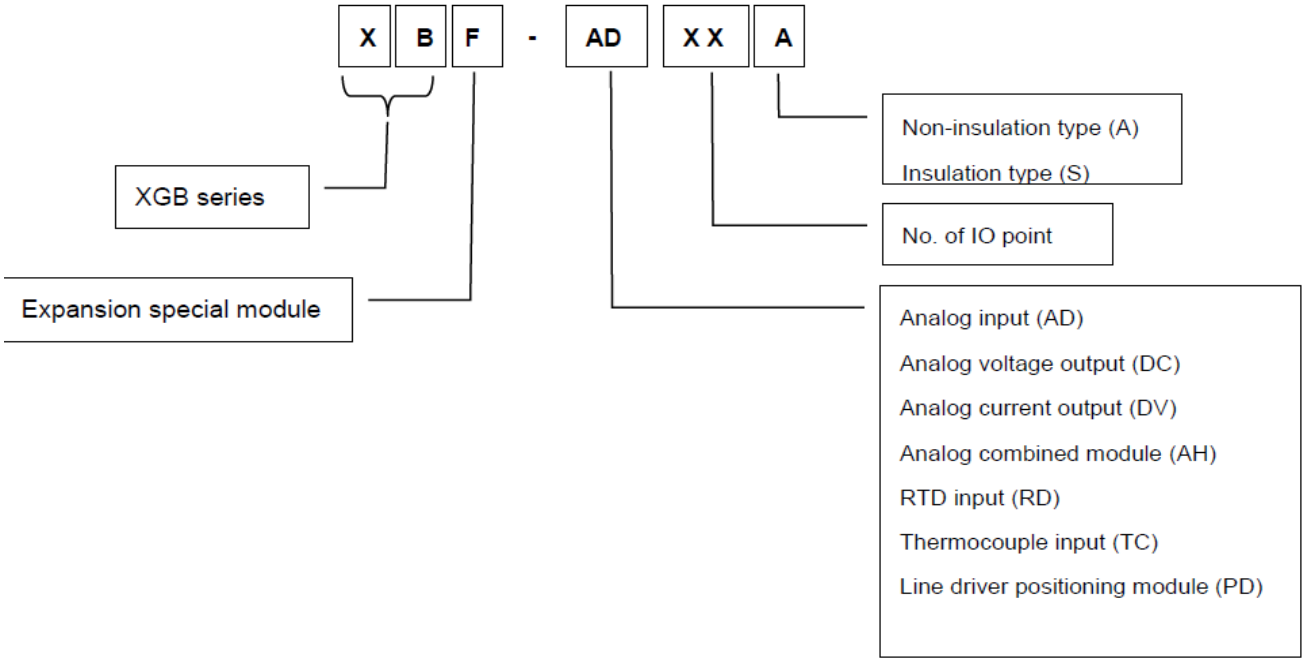
کد شناسایی PLC



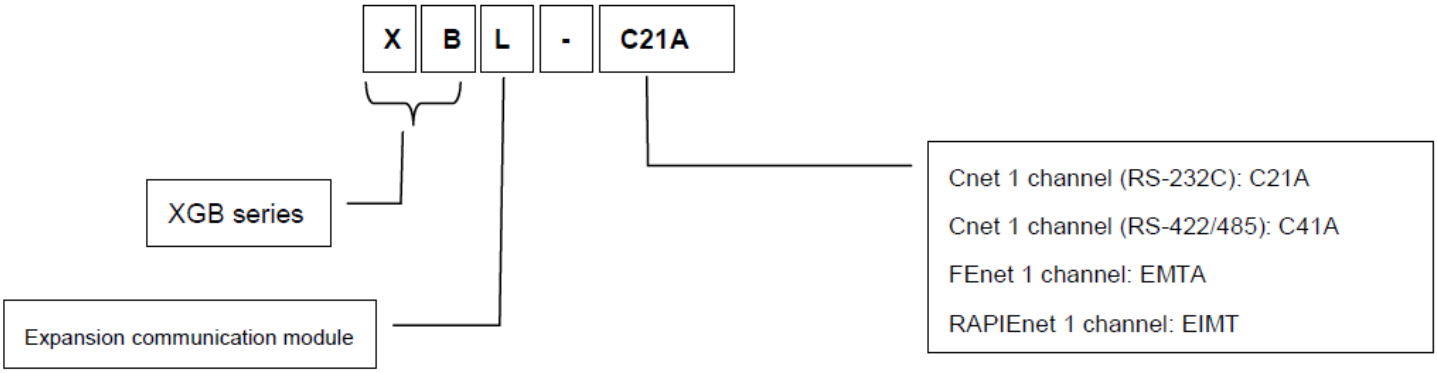
کد شناسایی کارت‌های ورودی/خروجی



کد شناسایی کارت‌های جانبی(آنالوگ/ دما/ موقعیت)



کد شناسایی کارت‌های شبکه



ساختمان داده

BOOL: این داده یک بیت از فضای حافظه را به خود اختصاص می‌دهد و دارای یک حالت صفر یا یک می‌باشد. برای ضبط یک عدد سمبل صفر یا یک مورد استفاده قرار می‌گیرد.

BYTE: این داده ۸ بیت از فضای حافظه را به خود اختصاص می‌دهد. بیت‌های درون این داده دارای ارزش مکانی نیستند. برای ضبط ۸عدد سمبل صفر و یک مورد استفاده قرار می‌گیرد.

WORD: این داده ۱۶ بیت از فضای حافظه را به خود اختاص می‌دهد. بیت‌های درون این داده دارای ارزش مکانی نیستند. برای ضبط ۱۶عدد سمبل صفر و یک مورد استفاده قرار می‌گیرد.

Double Word (DWORD): این داده ۳۲ بیت از فضای حافظه را به خود اختصاص می‌دهد. بیت‌های درون این داده دارای ارزش مکانی نیستند. برای ضبط ۳۲عدد سمبل صفر و یک مورد استفاده قرار می‌گیرد.

Long word(LWORD): این داده ۶۴ بیت از فضای حافظه را به خود اختصاص می‌دهد. بیت‌های درون این داده دارای ارزش مکانی نیستند. برای ضبط ۶۴عدد سمبل صفر و یک مورد استفاده قرار می‌گیرد.

SINT(short integer): این داده ۸بیت از فضای حافظه را به خود اختصاص می‌دهد. بازه این داده از ۱۲۸- تا ۱۲۷ می‌باشد. چون این داده دارای عدد منفی است پس یک بیت آن بیت علامت می‌باشد. به همین علت بازه این نوع داده نسبت به نوع USINT دارای اعداد کمتری می‌باشد.

USINT(unsigned short integer): این داده ۸بیت از فضای حافظه را به خود اختصاص می‌دهد. بازه این نوع داده از ۰تا ۲۵۵ می‌باشد.

INT(Integer) : این داده ۱۶ بیت از فضای حافظه را به خود اختصاص می‌دهد. بازه این داده از ۳۲۷۶۸- تا ۳۲۷۶۷ می‌باشد. چون این داده دارای عدد منفی است پس یک بیت آن بیت علامت می‌باشد. برای ضبط اعداد مثبت و منفی در این محدوده، از این نوع داده استفاده می‌شود.

UINT(Unsigned Integer): این داده ۱۶ بیت از فضای حافظه را به خود اختصاص می‌دهد. بازه این داده از ۰ تا ۶۵۵۳۵ می‌باشد. برای ضبط اعداد مثبت در این محدوده، از این نوع داده استفاده می‌شود.

DINT(Double integer): این داده ۳۲ بیت از فضای حافظه را به خود اختصاص می‌دهد. بازه این داده از ۲۱۴۷۴۸۳۶۴۸- تا ۲۱۴۷۴۸۳۶۴۷ می‌باشد. برای ضبط اعداد مثبت و منفی در این محدوده، از این نوع داده استفاده می‌شود. چون این داده دارای عدد منفی است پس یک بیت آن بیت علامت می‌باشد.

UDINT(unsigned double Integer): این داده ۳۲ بیت از فضای حافظه را به خود اختصاص می‌دهد. بازه این داده از ۰ تا ۴۲۹۴۹۶۷۲۹۵ می‌باشد. برای ضبط اعداد مثبت در این محدوده، از این نوع داده استفاده می‌شود.

LINT(Long Integer): این داده ۶۴ بیت از فضای حافظه را به خود اختصاص می‌دهد. چون این داده دارای عدد منفی است پس یک بیت آن بیت علامت می‌باشد. بازه این داده از $2^{63} - 1$ تا -2^{63} می‌باشد. برای ضبط اعداد مثبت و منفی در این محدوده، این نوع داده استفاده می‌شود.

ULINT(Unsigned Long Integer): این داده ۶۴ بیت از فضای حافظه را به خود اختصاص می‌دهد.

بازه این داده از صفر تا $2^{64} - 1$ می‌باشد. برای ضبط اعداد مثبت این محدوده، این نوع داده استفاده می‌شود.

REAL (اعداد اعشاری): این داده ۳۲ بیت از فضای حافظه را به خود اختصاص می‌دهد. بازه این داده از $\pm 3.402823 * 10^{38}$ و $\pm 1.401298 * 10^{-45}$ می‌باشد. برای ضبط اعداد مثبت و منفی در این محدوده، از این نوع داده استفاده می‌شود.

LREAL(اعداد اعشاری): این داده ۶۴ بیت از فضای حافظه را به خود اختصاص می‌دهد. بازه این داده از $\pm 1.797693 * 10^{308}$ و $\pm 4.9406564 * 10^{-324}$ می‌باشد. برای ضبط اعداد مثبت و منفی در این محدوده، از این نوع داده استفاده می‌شود.

TIME_OF_DAY: : این داده ۳۲ بیت از فضای حافظه را به خود اختصاص می‌دهد. برای ضبط ساعت روز استفاده می‌گردد. مثلا : TOD#12:15:54.

DATE : این داده ۱۶ بیت از فضای حافظه را به خود اختصاص می‌دهد. برای ضبط تاریخ استفاده می‌گردد. مثلا : D#2023:02:25.

DATE_AND_TIME : این داده ۸ بایت از فضای حافظه را به خود اختصاص می‌دهد. برای ضبط تاریخ و ساعت روز استفاده می‌گردد. مثلا: DT#2023-05-24-12:15:23.

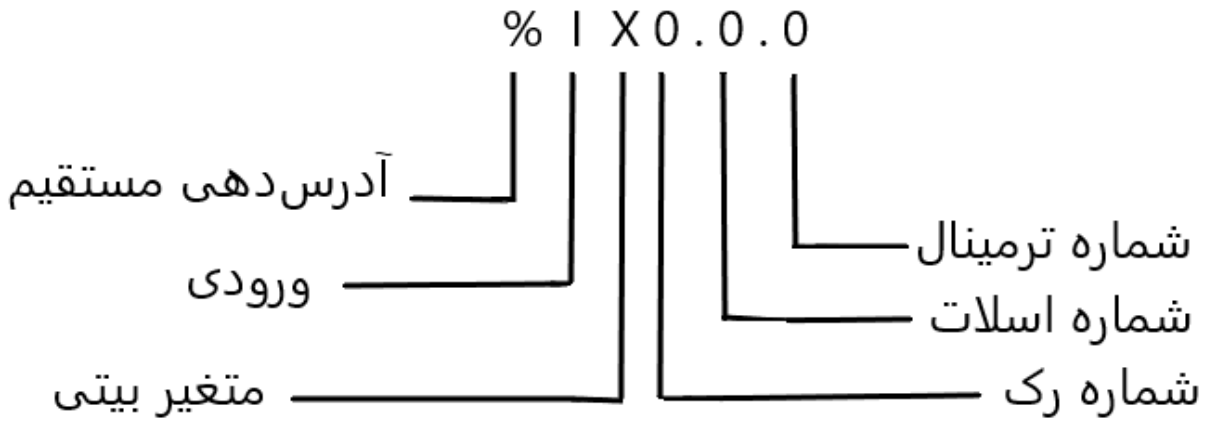
STRING : به زنجیره‌ای از کاراکترها یک STRING می‌گویند. مثلا ALIREZA . برای ضبط هرکاراکتر از یک بایت فضا استفاده می‌شود.

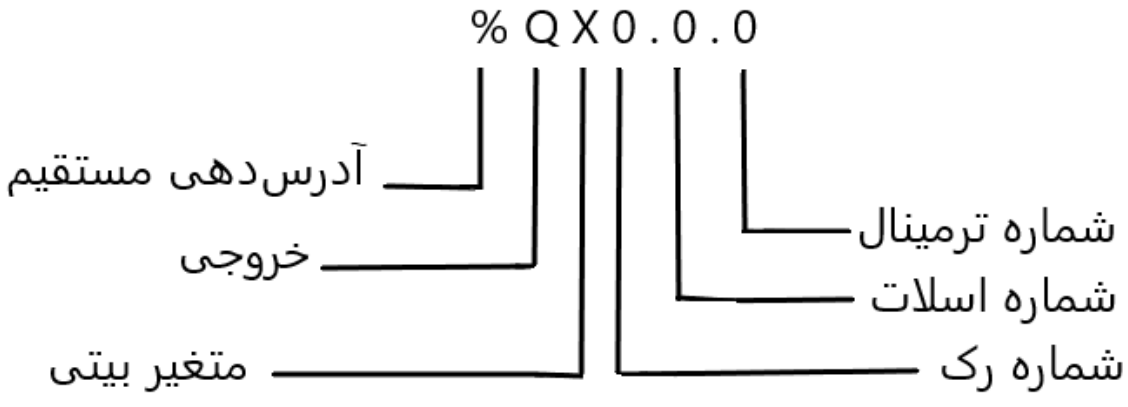
متغیرها

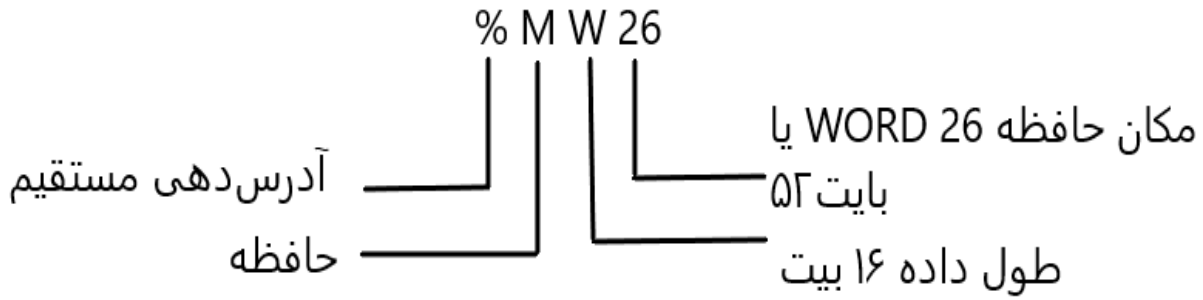
در برنامه‌نویسیPLC، متغیر عبارت است از نامی مستعار برای مکانی از حافظه برای ذخیره اطلاعات. هنگام تعریف متغیر حتما باید نوع داده آن را مشخص نمود. به دو شکل می‌توان متغیر را تعریف نمود در صورتی که به متغیر آدرس اختصاص داده نشود به آن آدرس‌دهی غیر مستقیم گویند و متغیر در حافظه ذخیره می‌شود. با اختصاص آدرس به متغیر، متغیر در آدرس خاصی از حافظه ذخیره می‌گردد که به این روش آدرس‌دهی مستقیم می‌گویند. برای این منظور قبل از آدرس علامت درصد(%) را قرار می‌دهند. معمولا نام متغیر را به نحوی انتخاب می‌کنیم که با عملکرد آن تطابق داشته باشد. ابتدا باید آن بخش حافظه را که می‌خواهیم به متغیر اختصاص دهیم را مشخص کنیم که ورودی(I)/خروجی(Q) و یا حافظه(M) است. سپس طول داده را باید تعیین نمایید. طول متغیر یک بیت باشد را با **x** نمایش می‌دهند مثلا ورودی یا خروجی‌ها، طول متغیر۸بیت باشد با **B** (بایت)

نمایش می‌دهند مثلا در حافظه‌ها. طول متغیر۱۶بیت باشد با **w** (WORD) نمایش می‌دهند مثلا در حافظه‌ها. طول متغیر۳۲بیت باشد با **D** (Double WORD) نمایش می‌دهند مثلا در حافظه‌ها. طول متغیر۶۴بیت باشد با **L** (Long Word) نمایش می‌دهند مثلا در حافظه‌ها.

شیوه آدرس‌دهی مستقیم







انواع متغیر

متغیرهای عمومی (محلیLocal)

این نوع متغیر فقط در برنامه‌ای که در حال اجرا است اعتبار دارد و درصورتی که چندین برنامه مختلف نوشته باشید در برنامه‌های دیگر اعتبار ندارد. این نوع متغیر را با VAR نشان می‌دهند.

متغیر همگانی(جهانیGlobal)

درصورتی که چندین برنامه مختلف نوشته باشید این نوع متغیر در تمامی برنامه‌ها اعتبار دارد و آن را با VAR_EXTERNAL نشان می‌دهند.

متغیرهای Retain (ماندگار)

این نوع متغیر بصورت VAR_RETAIN تعریف شده و با stop و سپس run نمودن برنامه مقدار متغیر پاک نمی‌شود و همچنین اگر restart سیستم بصورت warm باشد یا برق قطع شود مقدار متغیر پاک نمی‌شود.

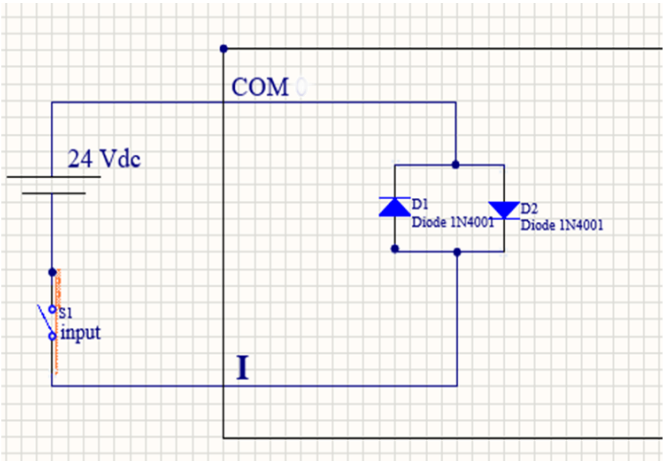
متغیرهای ثابت

این نوع متغیر بصورت VAR_CONSTANT تعریف شده و در هنگام تعریف این متغیر باید یک مقدار اولیه به آن واگذار کرد و در حین اجرای برنامه دیگر مقدار آن را نمی‌توان تغییر داد.

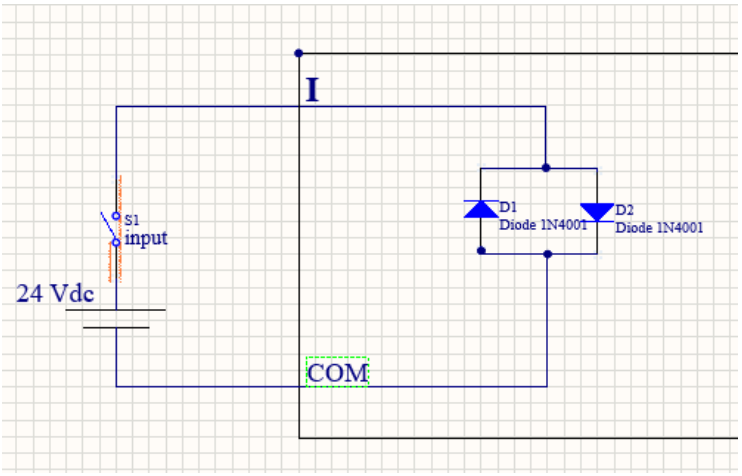
متغیرهای ورودی و خروجی(I,Q) را نمی‌توان بعنوان متغیر VAR_RETAIN تعریف کرد.

نحوه اتصال ورودی/خروجی

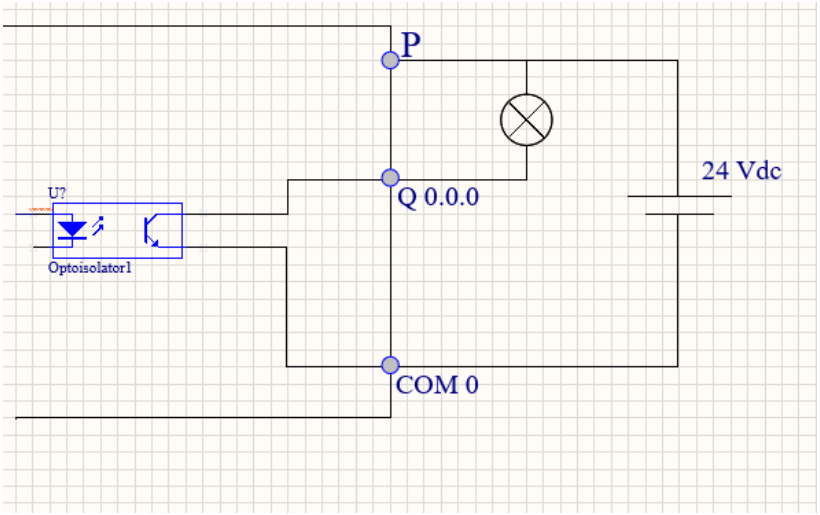
ورودی نوع PNP یا Source



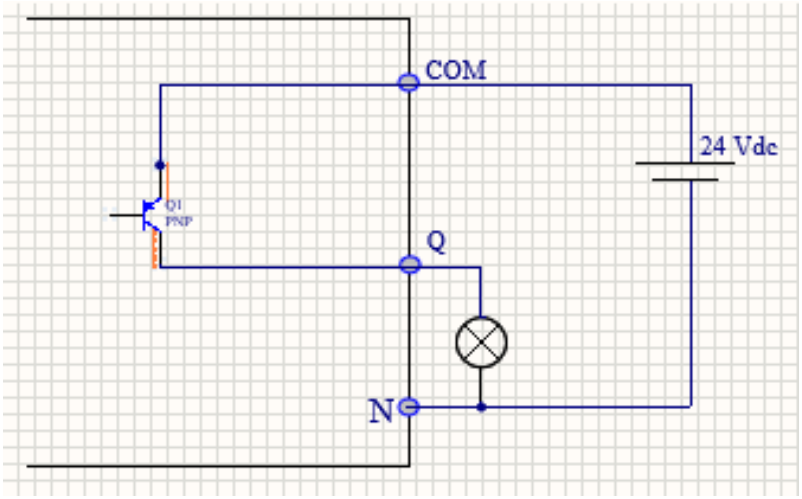
ورودی نوع NPN یا Sink



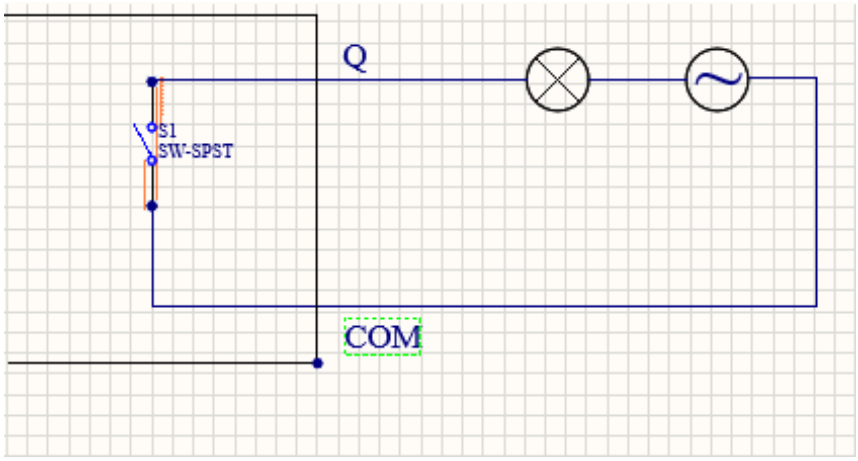
خروجی NPN



خروجی PNP



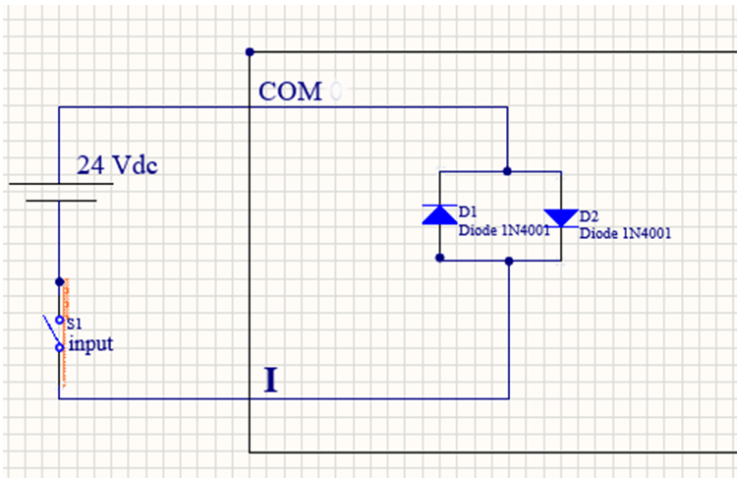
خروجی رله‌ای



برنامه۱: قطع و وصل یک لامپ با شستی(روشن شدن لحظه‌ای لامپ)

در این تست بعلت استفاده از شستی، پس لامپ بصورت دایم کار نمی‌کند و به محض برداشتن دست از روی شستی لامپ خاموش می‌گردد.

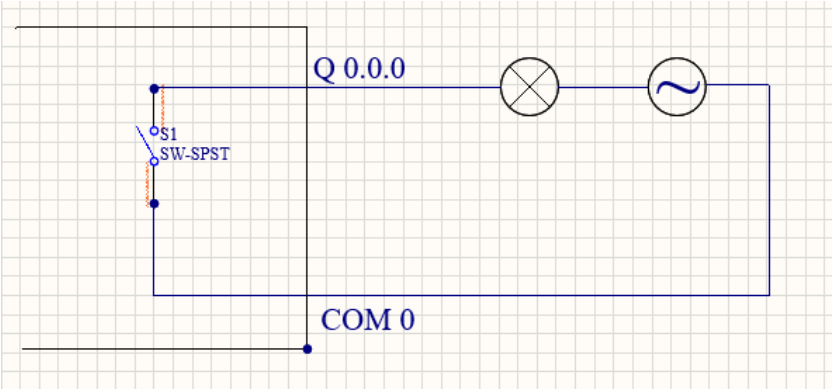
چون ورودی یک شستی ساده هست و سنسور با خروجی NPN یا PNP نیست ما می‌توانیم ۲۴+ و یا ۲۴- را به پایه COM وصل کنیم و مدار زیر را ببندیم.



پایه‌های ورودی PLC مدل XEC-(DR/DN(P)20SU بصورت زیر است.

DC IN 24 Vdc . 4mA (DR/DN(P)20)													
RX	TX	SG	I 01	I 03	I 05	I 07	I 09	I 11	_	_	_		
	+485	-485	I00	I 02	I 04	I 06	I 08	I 10	_	_	_	COM	

برای اتصال خروجی به PLC باید مدار زیر بسته شود.



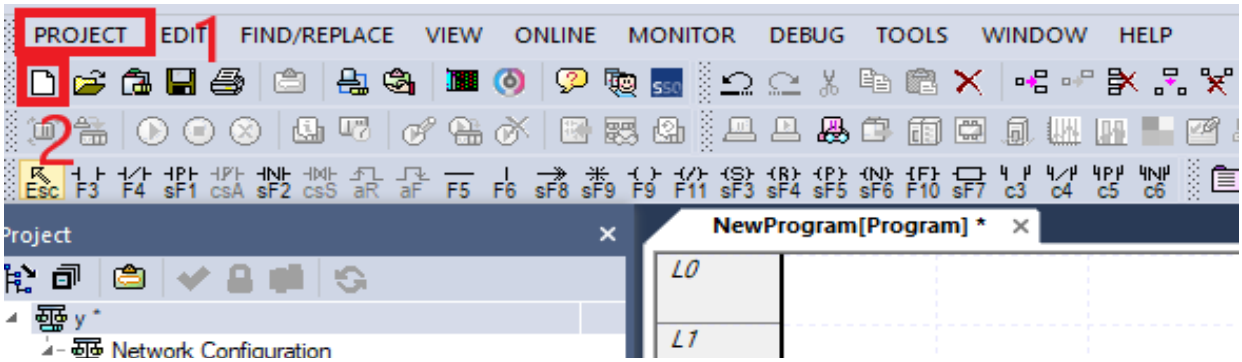
پایه‌های خروجی PLC مدلXEC-DR20SU بصورت زیر است. پایه ۴۰ به COM0 پایه۴۱ به COM1 پایه ۴۲و ۴۳ به COM2و پایه ۴۴تا ۴۷ به COM3 متصل می‌گردد.

0.5-0.3 A 50/60 Hz			RELAY OUT:24 Vdc.220Vac,2 A/P, 5 A/C (DR20SU)										
L	100-240 Vac	N	40	41	42	_	44	45	_	_	_	24 V	
PE		COM 0	COM 1	COM2	43	COM 3	46	47	_	_	_	24G	

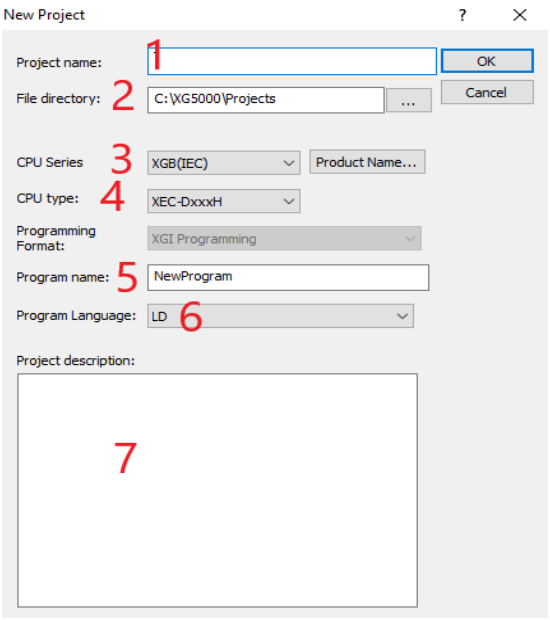
ابتدا به پایه‌های اول همانطور که درشکل پیداست ولتاژ ۲۲۰ ولت را متصل می‌نماییم. اگر خروجی رله‌ای باشد و یا NPN باشد ۲۴- را به پایه COM مربوط به خروجی فعال شده در برنامه متصل می‌کنیم، و پایه خروجی را به لامپ متصل کرده و از لامپ به ۲۴+متصل می‌کنیم. در نتیجه با قطع وصل شستی، لامپ روشن و خاموش می‌شود. برای قطع و وصل دایم و استفاده از شستی باید در برنامه از تیغه باز همنام خروجی بصورت موازی با استارت استفاده کرد. تا پس از برداشتن دست از روی شستی خروجی قطع نشود.

ایجاد پروژه

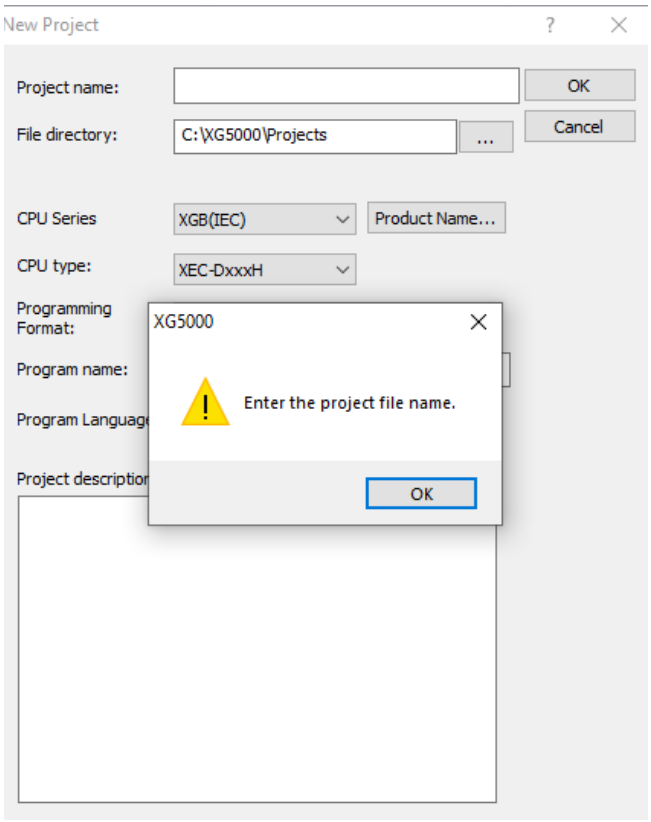
با انتخاب گزینه Project و انتخاب new Project پروژه جدید ایجاد می‌شود. یا با انتخاب قسمت ۲ که به شکل برگه کاغذ است پروژه ایجاد می‌شود.



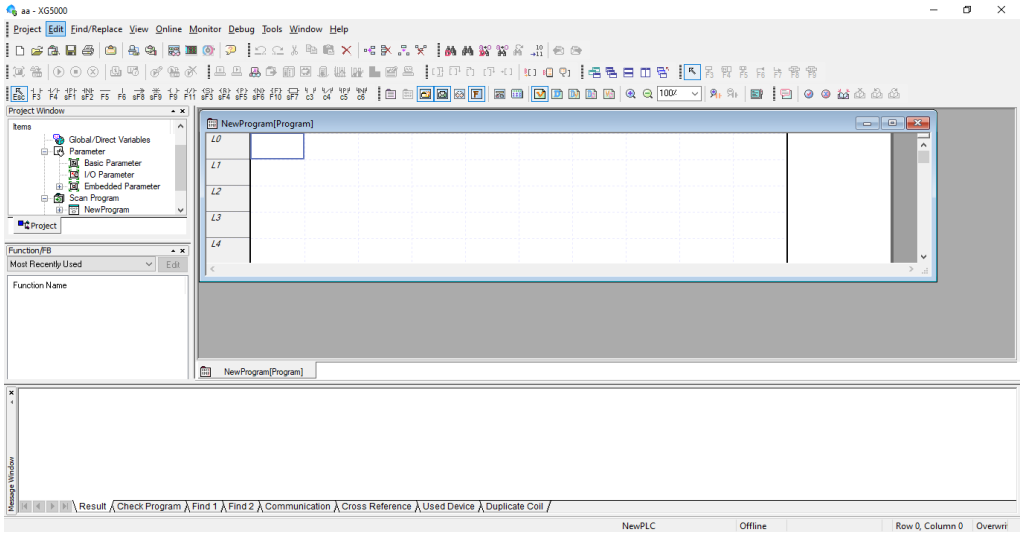
سپس پنجره زیر نمایش داده می‌شود. در قسمت۱ نام پروژه را وارد کنید. در قسمت۲ آدرس محل ذخیره برنامه را وارد نمایید. در قسمت ۳ خانواده CPU یا سری آن را وارد نمایید. درقسمت۴ مدل CPU را تنظیم کنید. در قسمت۵ اسم برنامه را وارد نمایید در قسمت۶ زبان برنامه نویسی را تنظیم نمایید. در قسمت۷ توضیحات لازم درمورد برنامه را بنویسید.



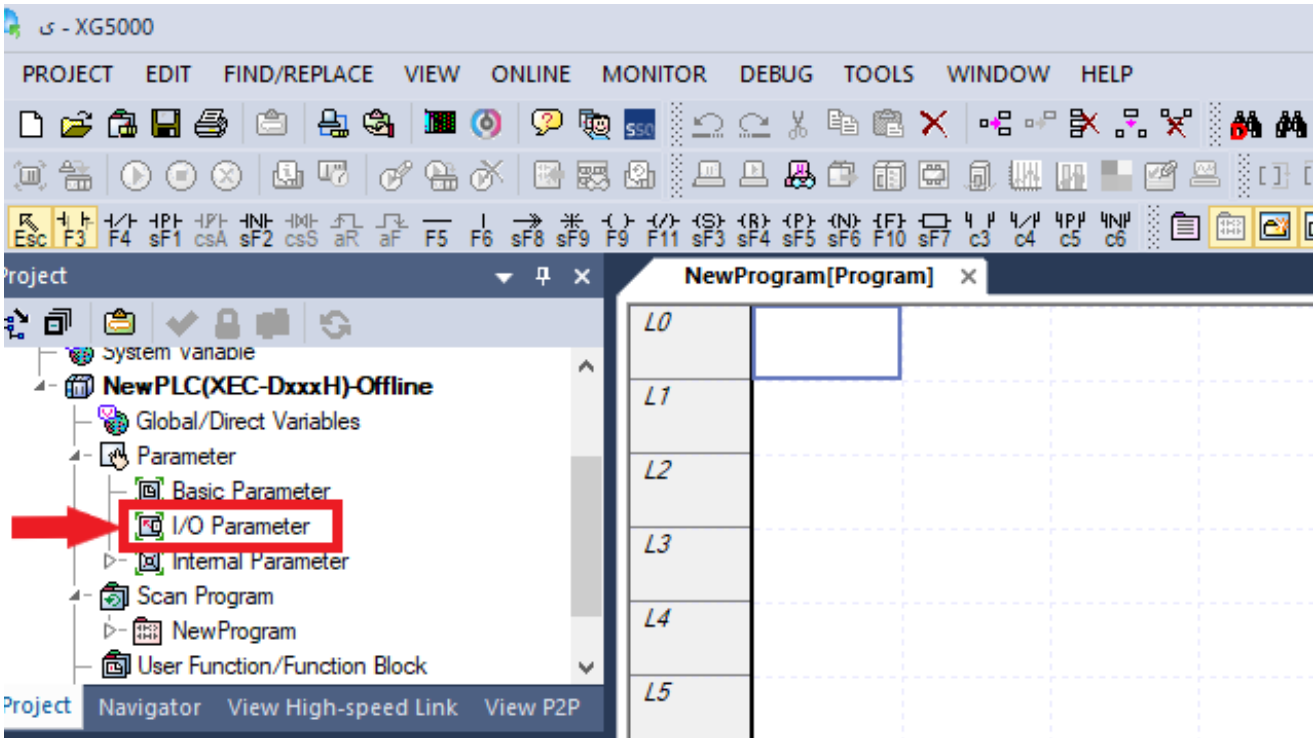
در این پروژه نوع PLC ما از نوع XGB-XEC-DR20S می‌باشد و همچنین زبان برنامه‌نویسی ما LD می‌باشد. درصورت ننوشتن نام پروژه اخطار زیر نمایان می‌شود.



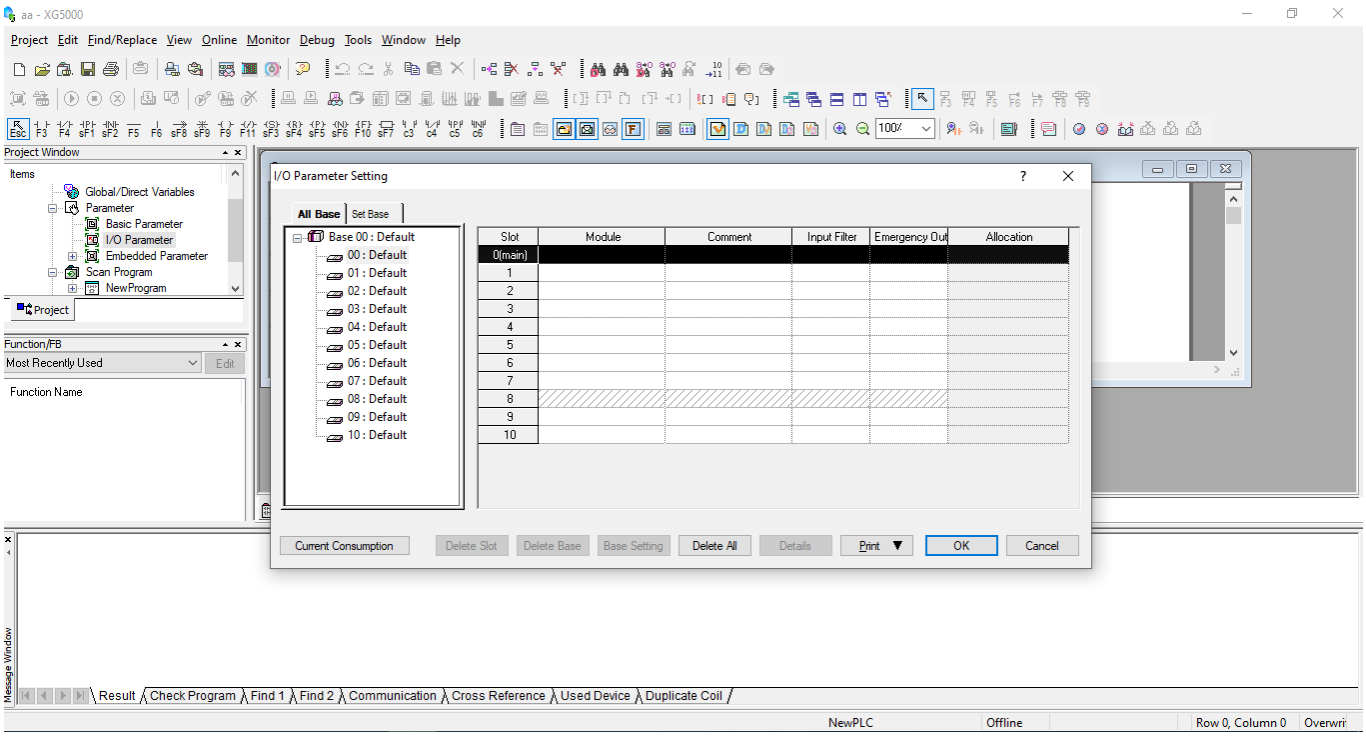
پس از تایید صفحه زیر نمایان می‌شود.



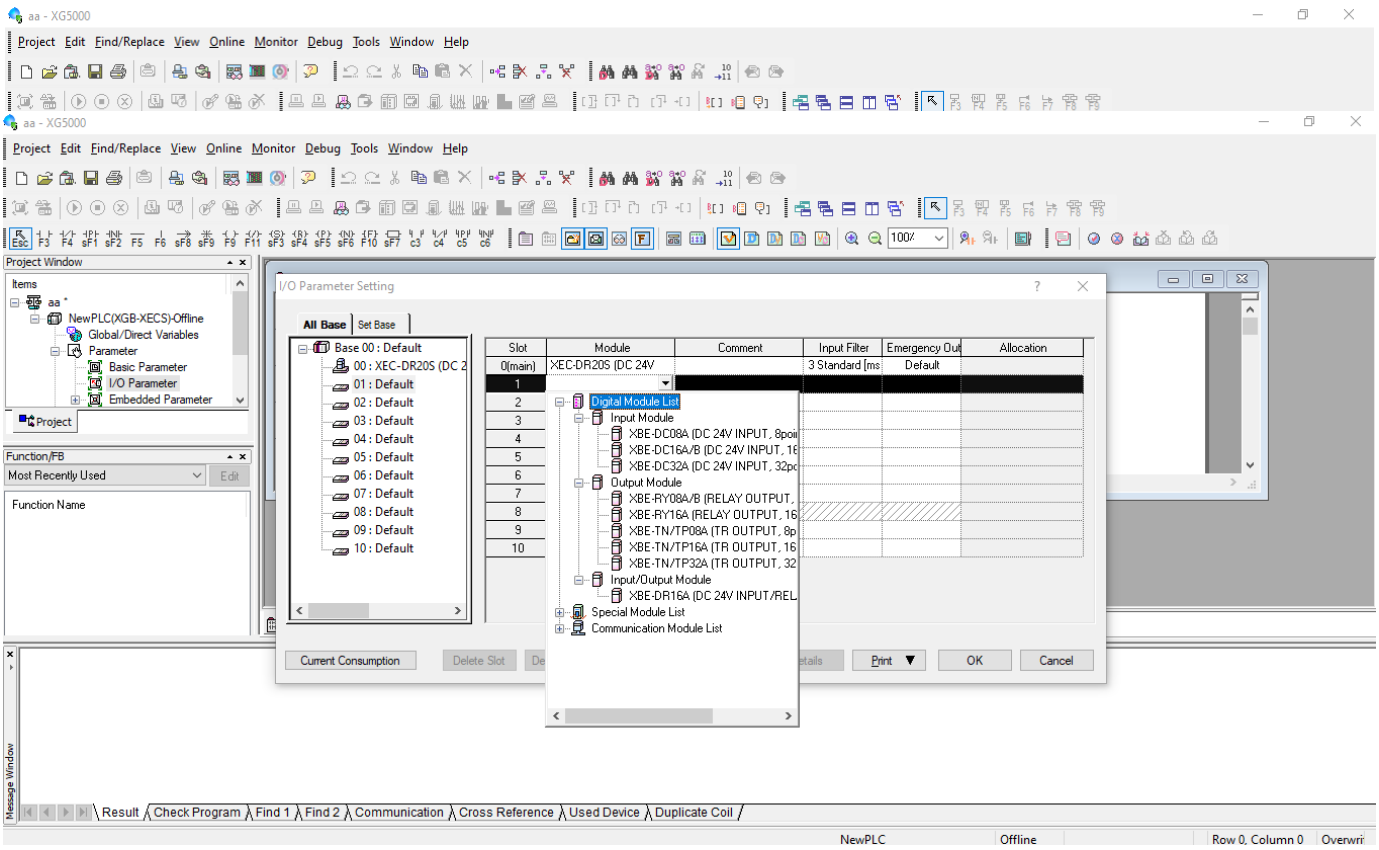
سپس پیکربندی یا چیدمان PLC و ماژول‌ها را تعیین می‌کنیم. با انتخاب گزینه I/o parameter چیدمان را انتخاب می‌کنیم.



پس از انتخاب I/O Parameter پنجره زیر باز می‌شود.

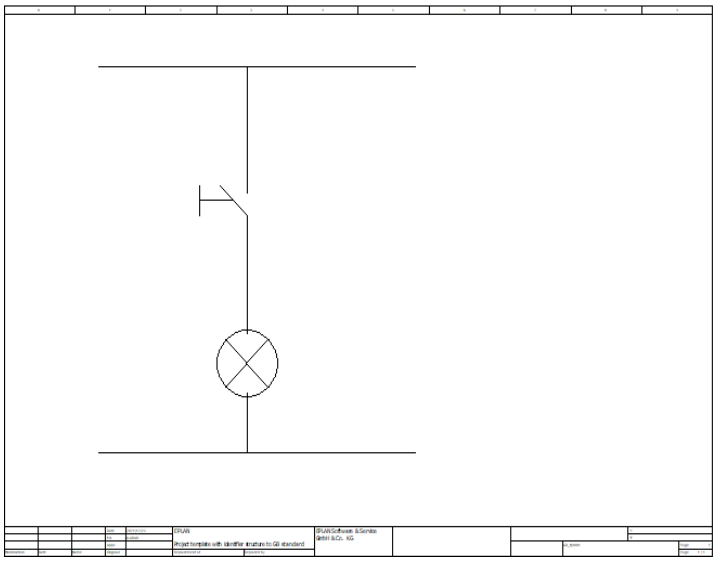


در قسمت ۰ نوع CPU را انتخاب می‌کنیم، که در PLC در دسترس ما XEC-DR20S می‌باشد.



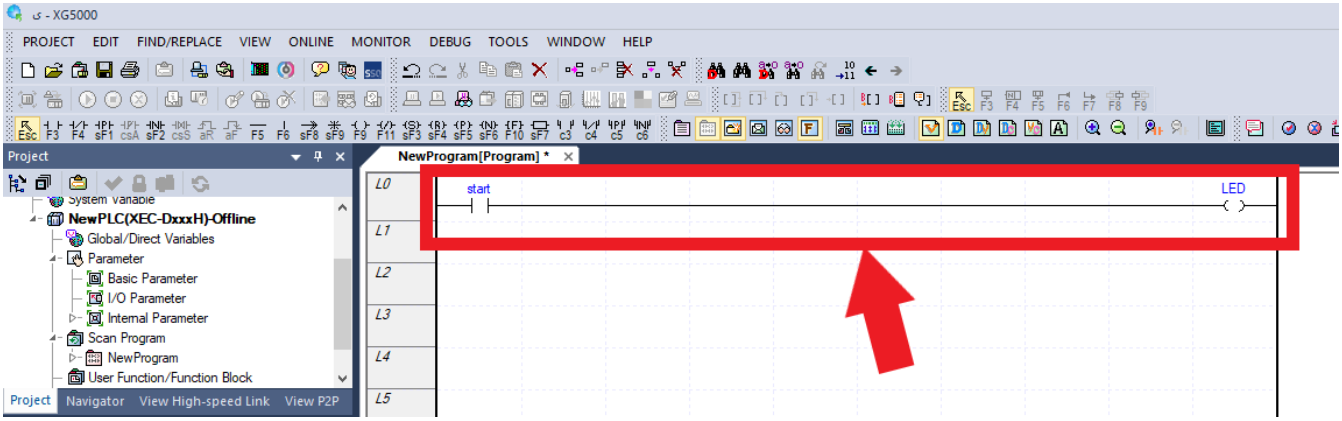
سپس در قسمت۱ نوع ماژول متصل به PLC را وارد می‌کنیم.

پس از پیکربندی به سراغ برنامه نویسی می‌رویم. ابتدا مدار کنترلی را رسم می‌کنیم.

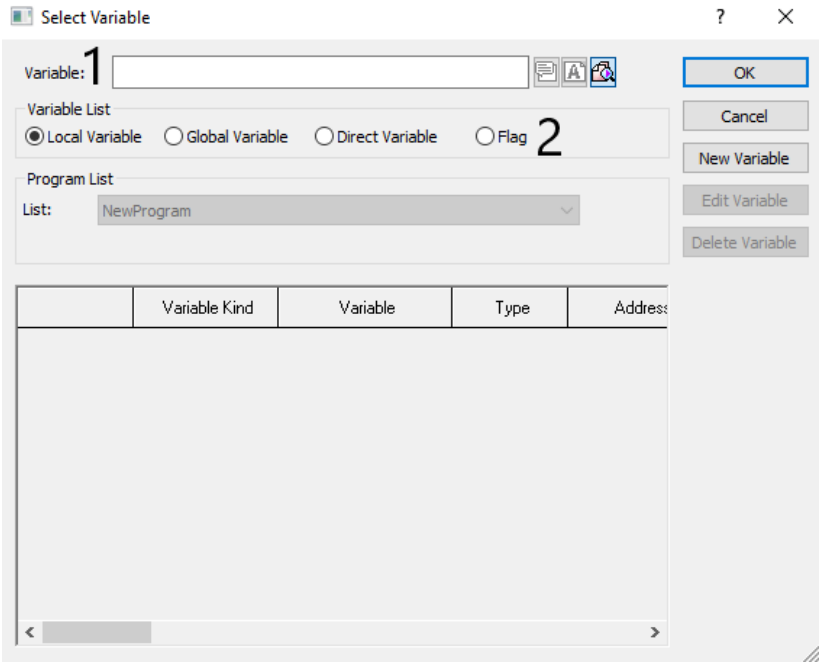


سپس در برنامه XG5000 برنامه را رسم می‌کنیم. برای روشن و خاموش کردن یک لامپ با کلید/شستی به یک تیغه باز (F3) بعنوان ورودی و یک کویل(F9) بعنوان خروجی نیاز داریم. می‌توان با کلید میانبر F3 تیغه باز و با کلید میانبر F9 کویل را انتخاب کرد.

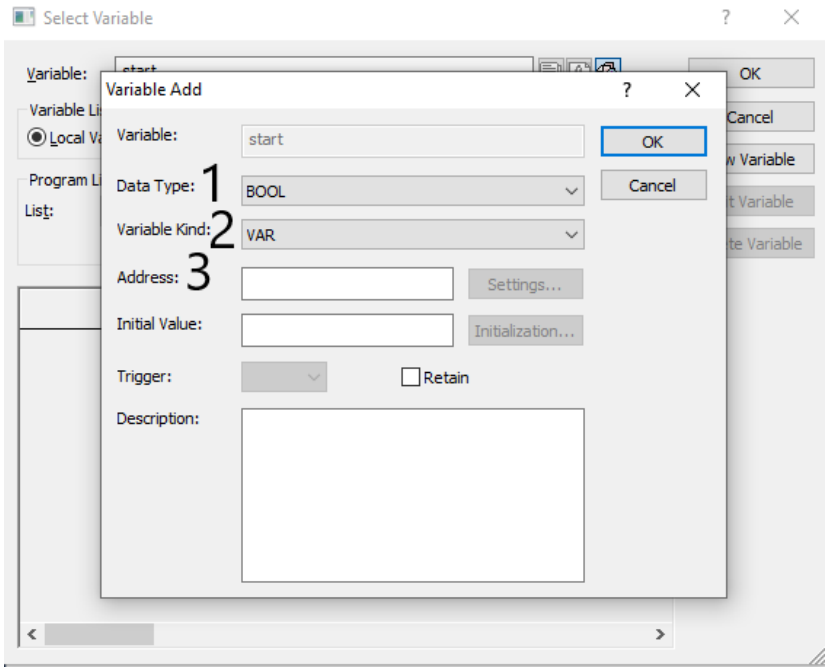
برنامه این تست به شکل زیر می‌باشد.



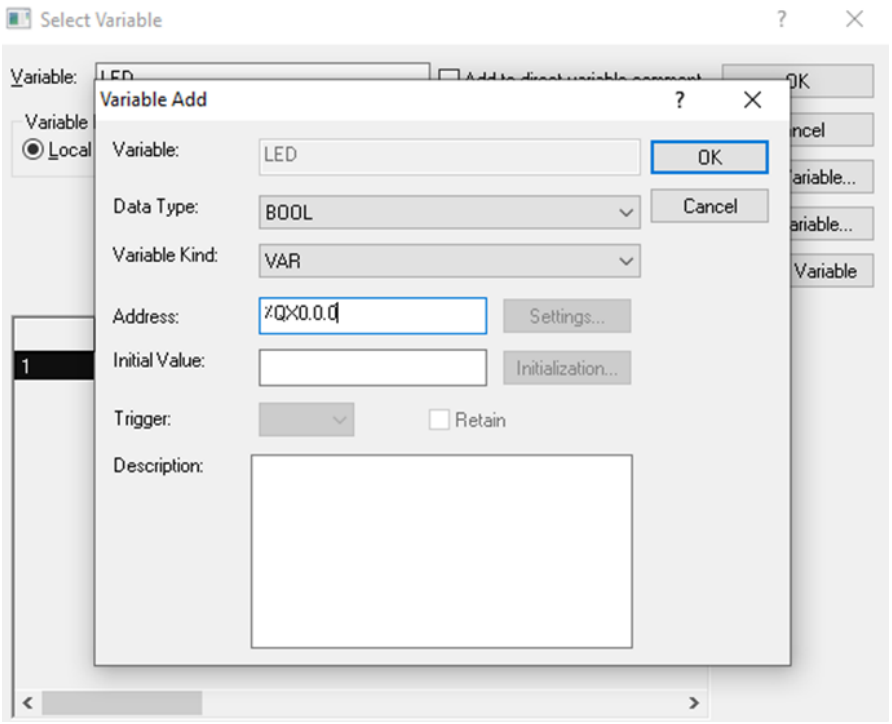
پس از انتخاب تیغه باز و یا کویل، صفحه زیر باز می‌شود. در قسمت ۱ نام متغیر ورودی را قرار می‌دهیم که بهتر است همنام وظیفه آن ورودی باشد. زیرا در برنامه‌های با تعداد سطر زیاد دلیل استفاده از آن المان در برنامه واضح باشد. در قسمت۲ نوع متغیر را انتخاب می‌کنیم.



پس از تایید صفحه زیر باز می‌شود. در قسمت۱ نوع داده را انتخاب می‌کنیم. در قسمت۲ نوع متغیر را انتخاب می‌کنیم. در قسمت۳ آدرس ورودی را وارد می‌کنیم. برای آدرس ورودی باید دستور %ix0.0.0و برای آدرس خروجی دستور %Qx0.0.0 را وارد کنیم.

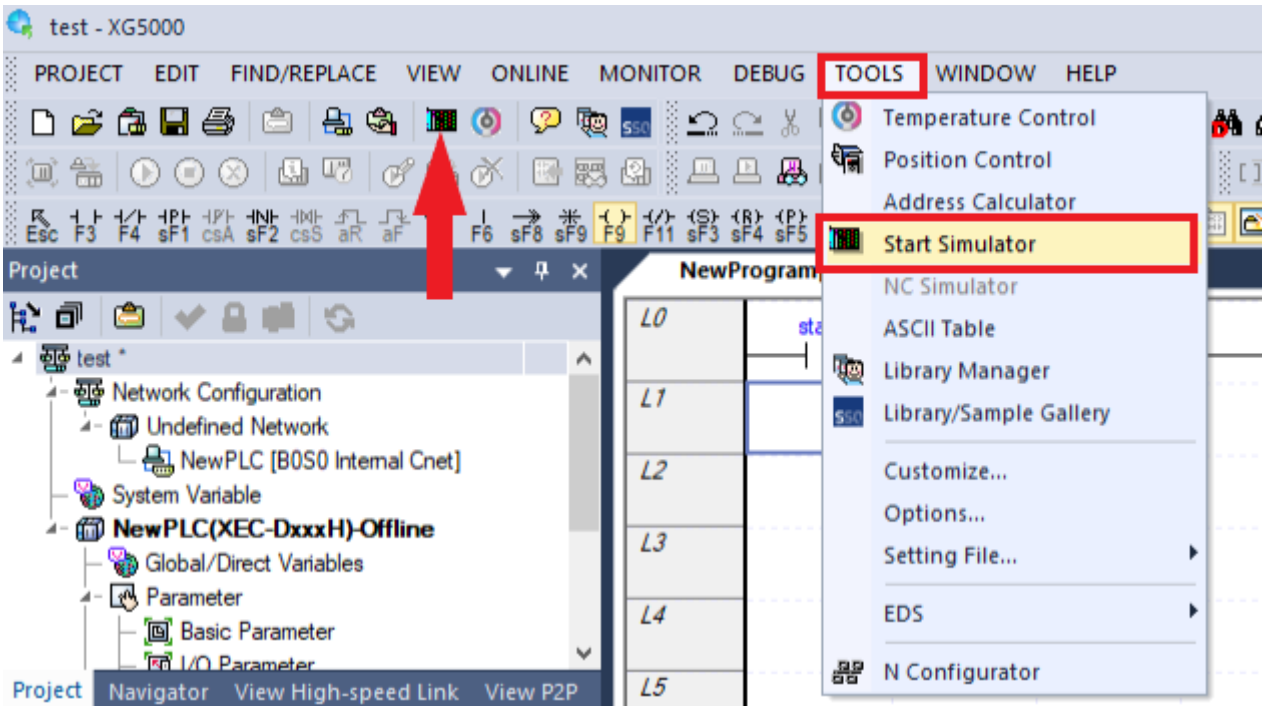


برای خروجی بصورت زیر می‌باشد.

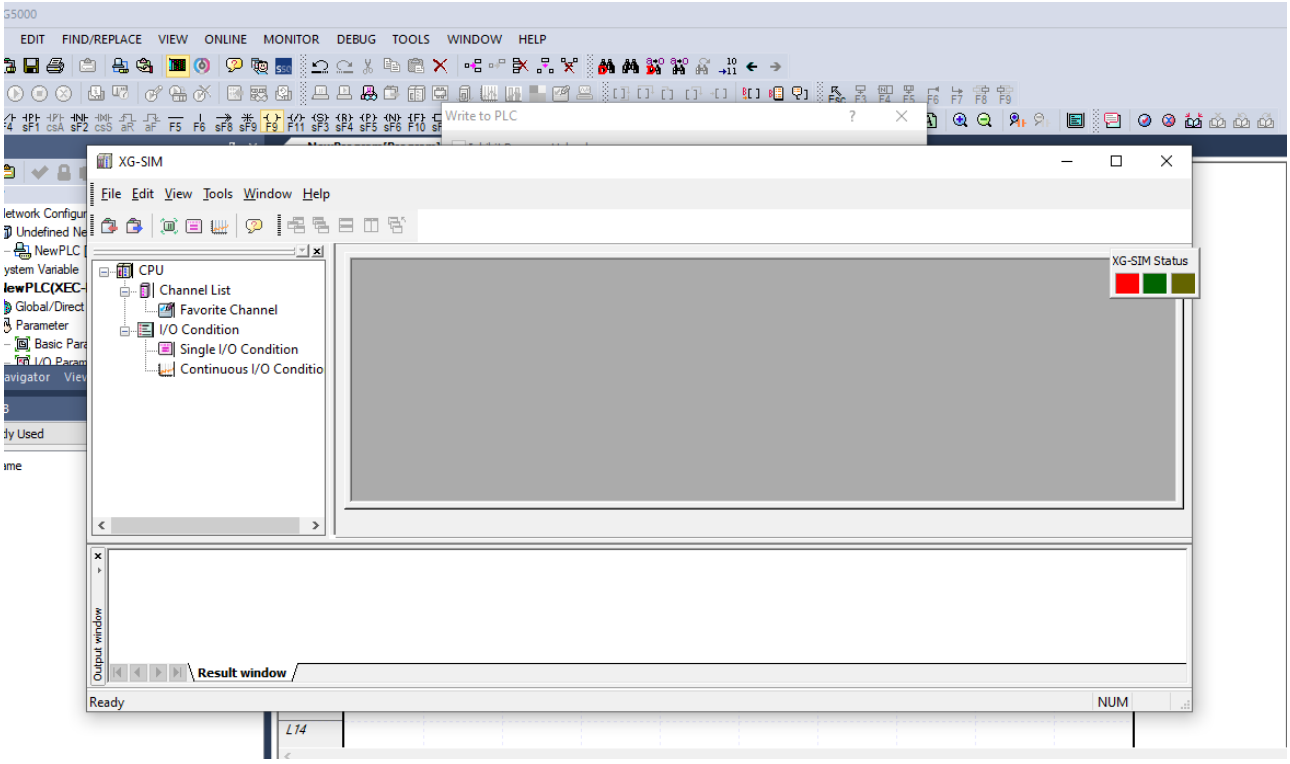


شبیه‌ساز (Simulator)

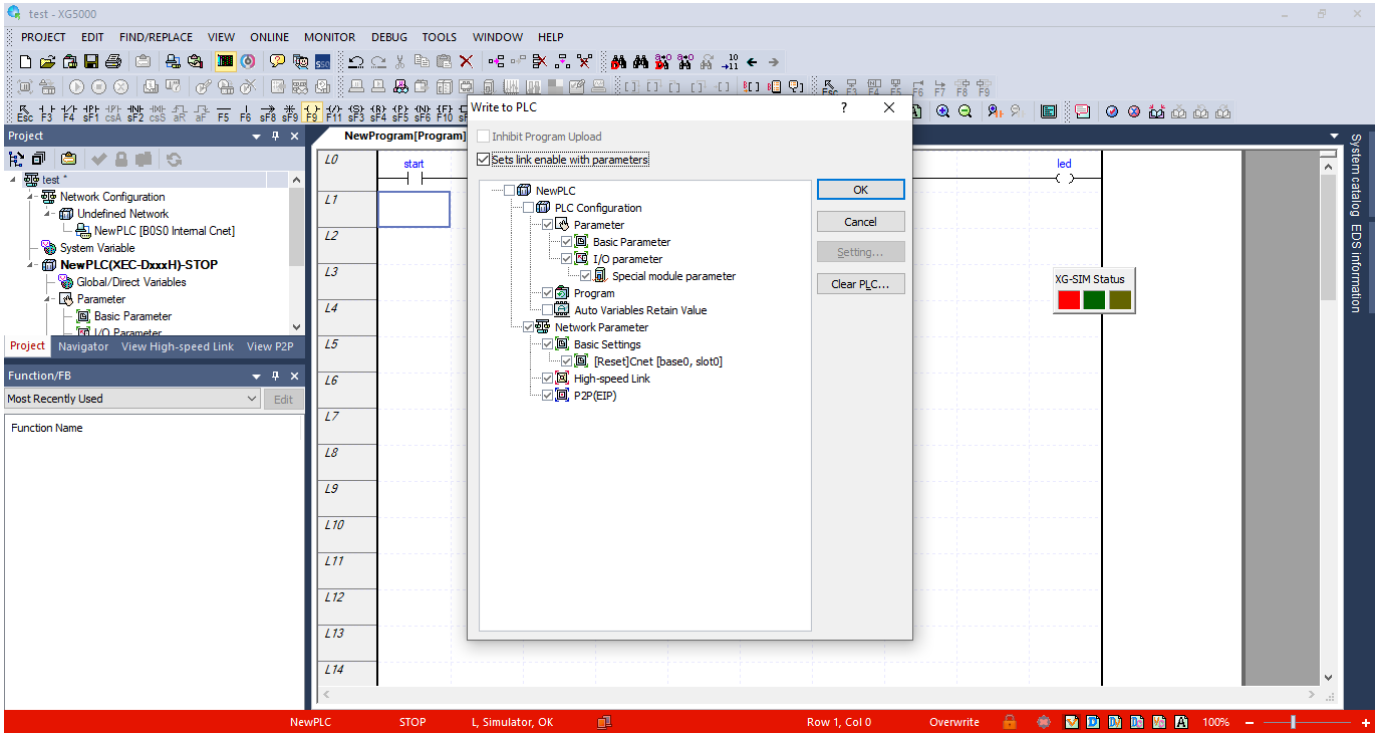
برای بررسی درستی برنامه از قسمت simulator یا شبیه‌ساز نرم‌افزار استفاده می‌کنیم، برای انتخاب شبیه‌ساز یا از منوی Tools گزینه start simulator را انتخاب نموده یا ( ) را در نوار ابزار انتخاب می‌کنیم.



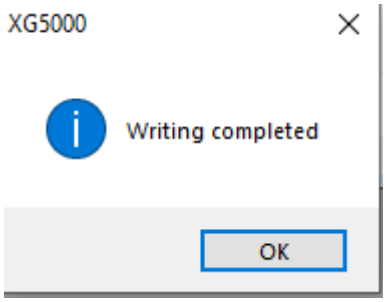
سپس پنجره زیر باز می‌شود.



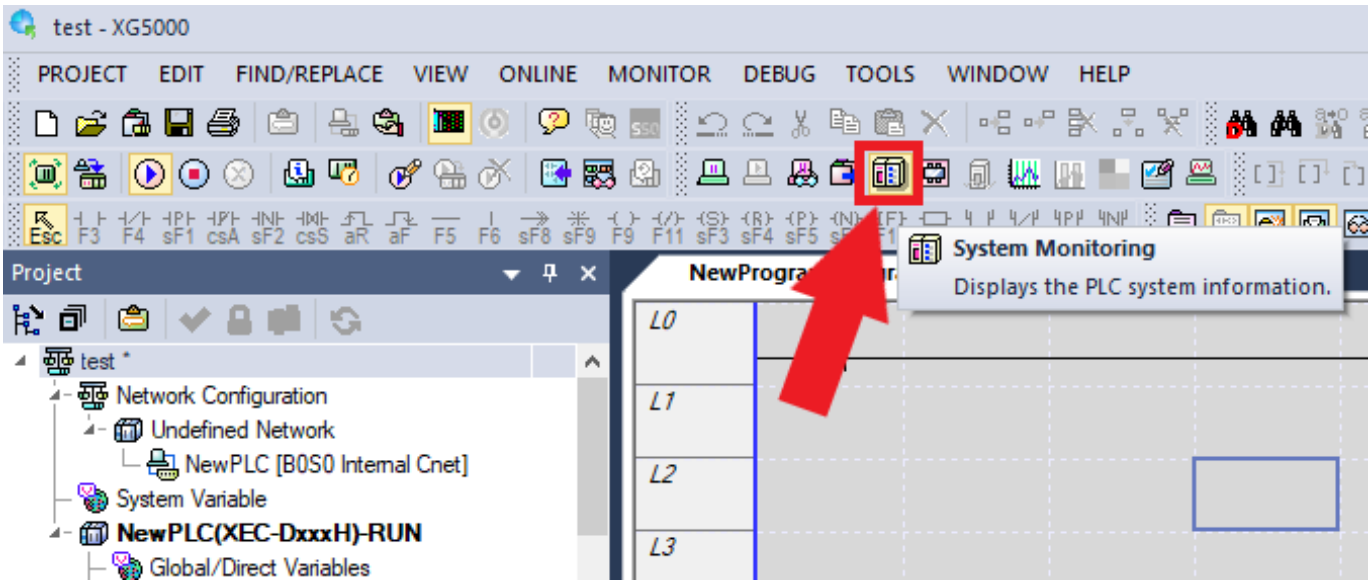
سپس پنجره زیر باز می‌شود که با تایید آن برنامه در PLC مجازی بارگذاری می‌شود.

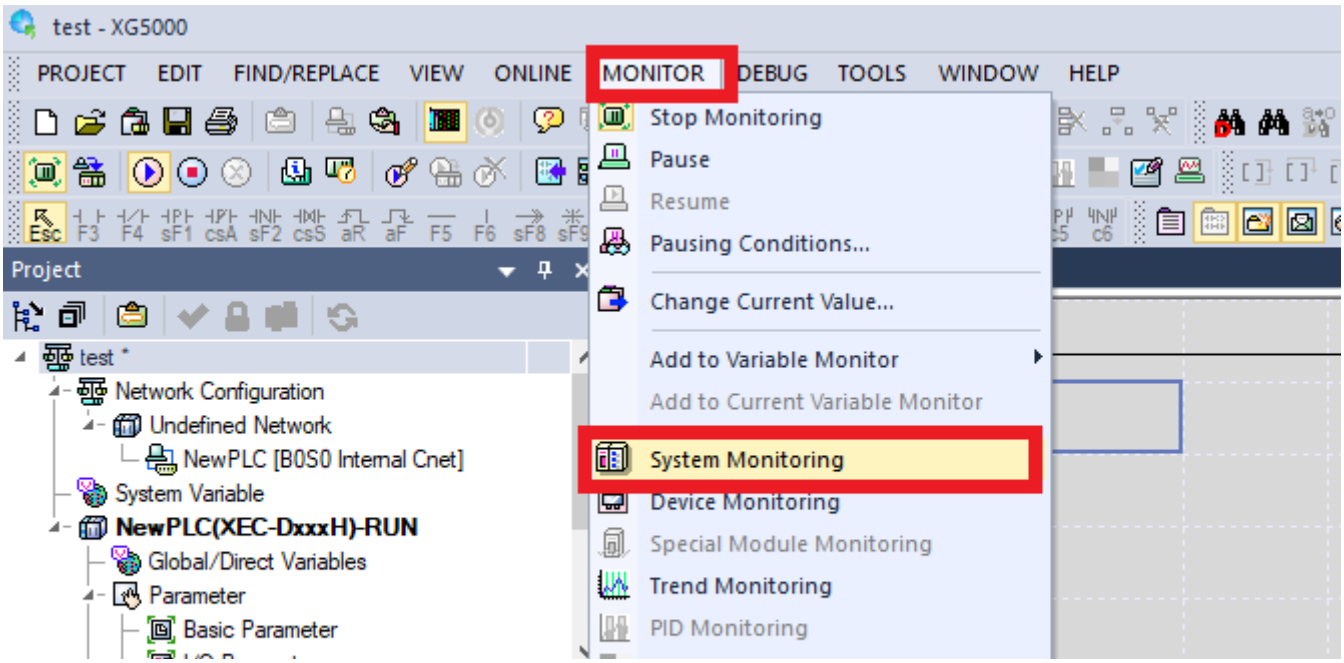


در صورتی که بارگذاری بدون مشکل باشد پنجره زیر ظاهر می‌شود.

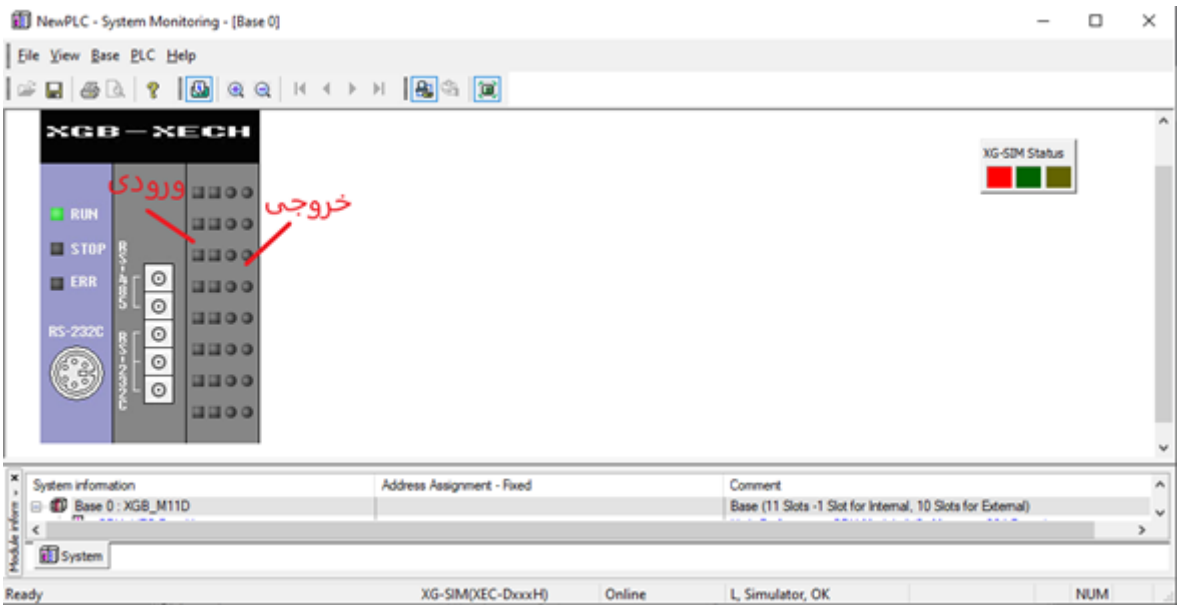


سپس گزینه system monitoring را در منوی monitor انتخاب می‌کنیم و یا بر روی شکل ( ) در نوار ابزار کلیک می‌کنیم.

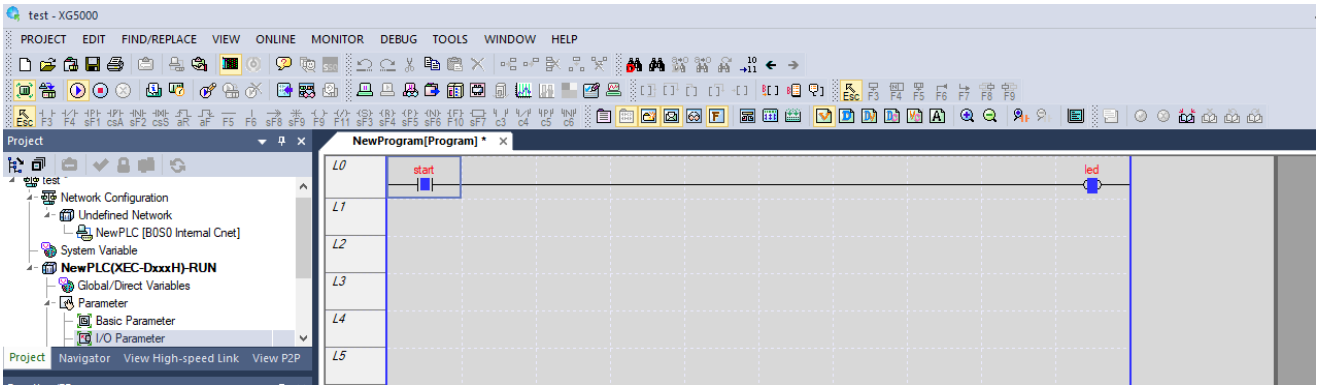


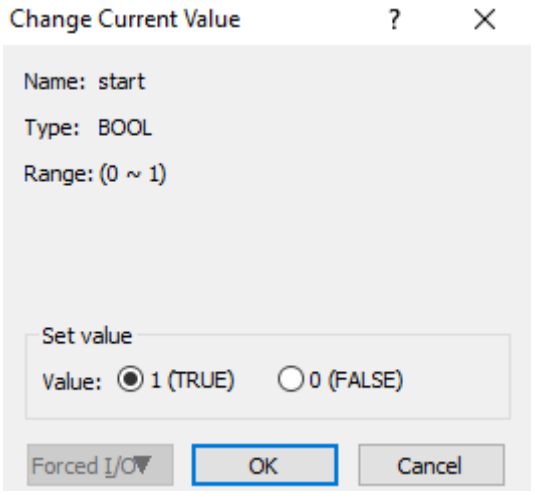


سپس در پنجره باز شده PLC نمایش داده می‌شود.

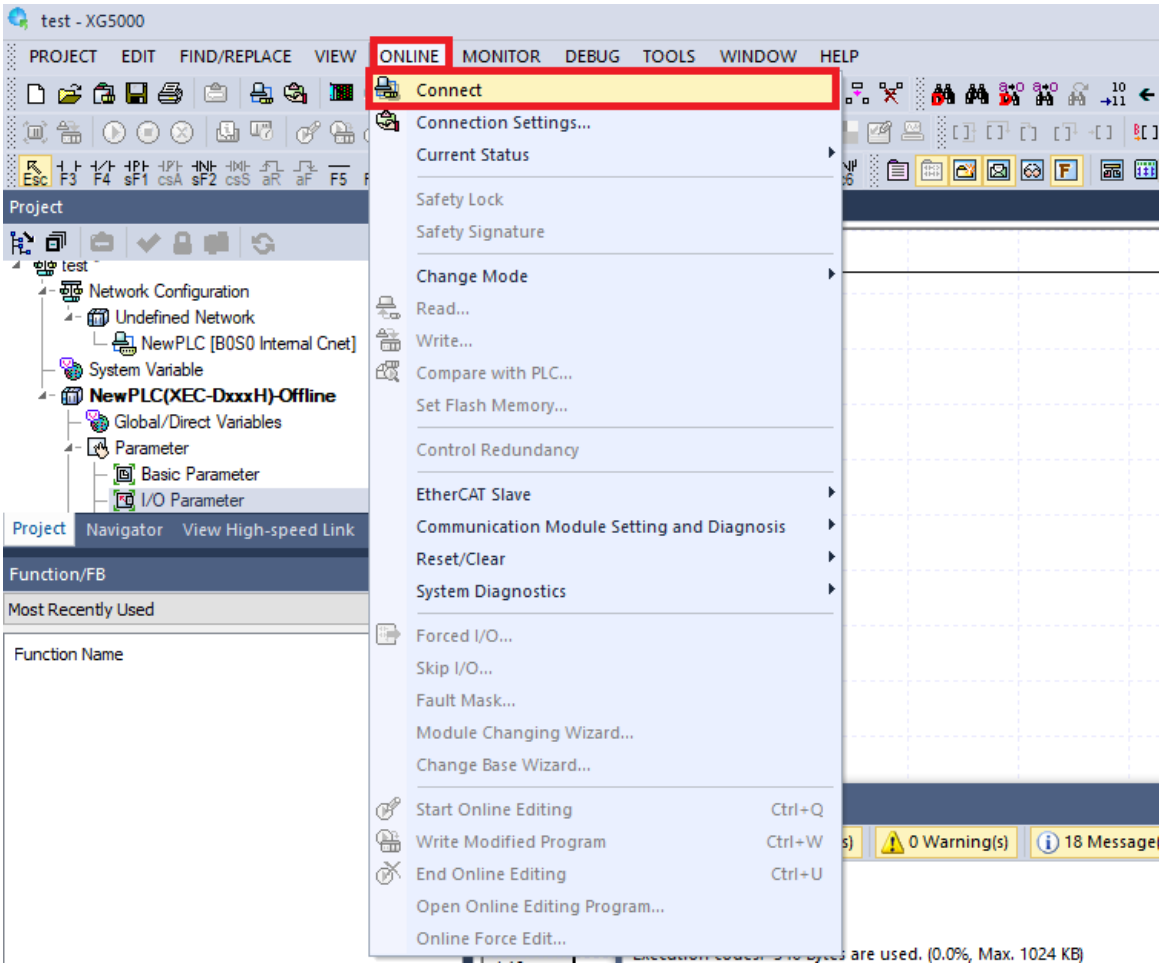


سپس در نرم‌افزار بر روی ورودی دبل کلیک کرده و در پنجره باز شده مقدار ۱ را انتخاب نمایید تا ورودی فعال شود و خروجی نیز فعال می‌شود.



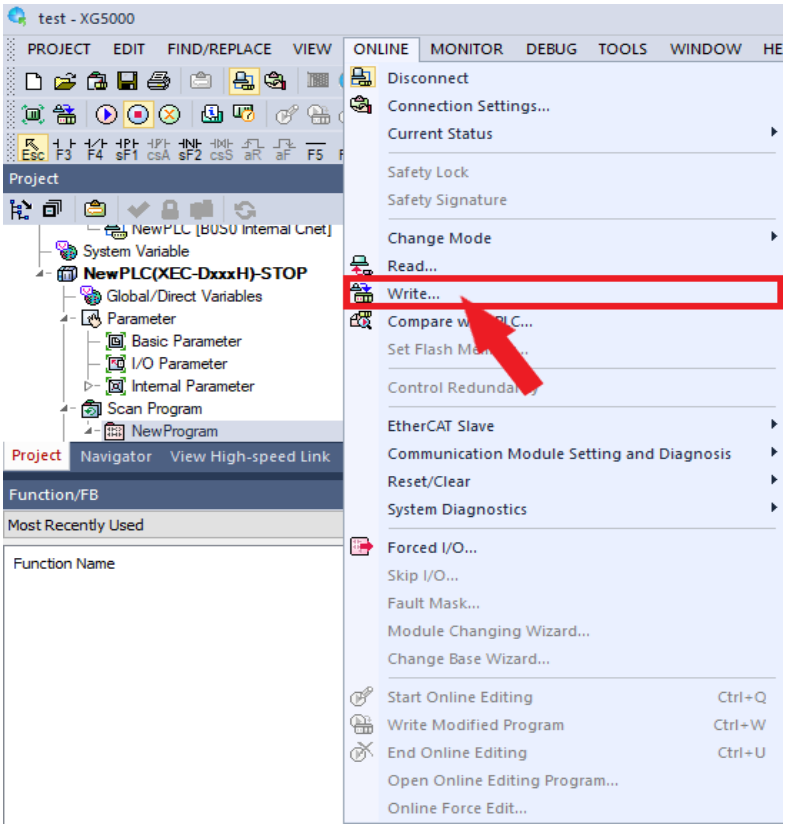


اکنون که از درستی برنامه نوشته شده مطمین شدیم PLC را به کامپیوتر متصل می‌کنیم. برای اتصالPLC به کامپیوتر از کابلmini-USB استفاده می‌کنیم. سپس گزینه Online را انتخاب کرده و در پنجره باز شده گزینه connect را انتخاب می‌کنیم.

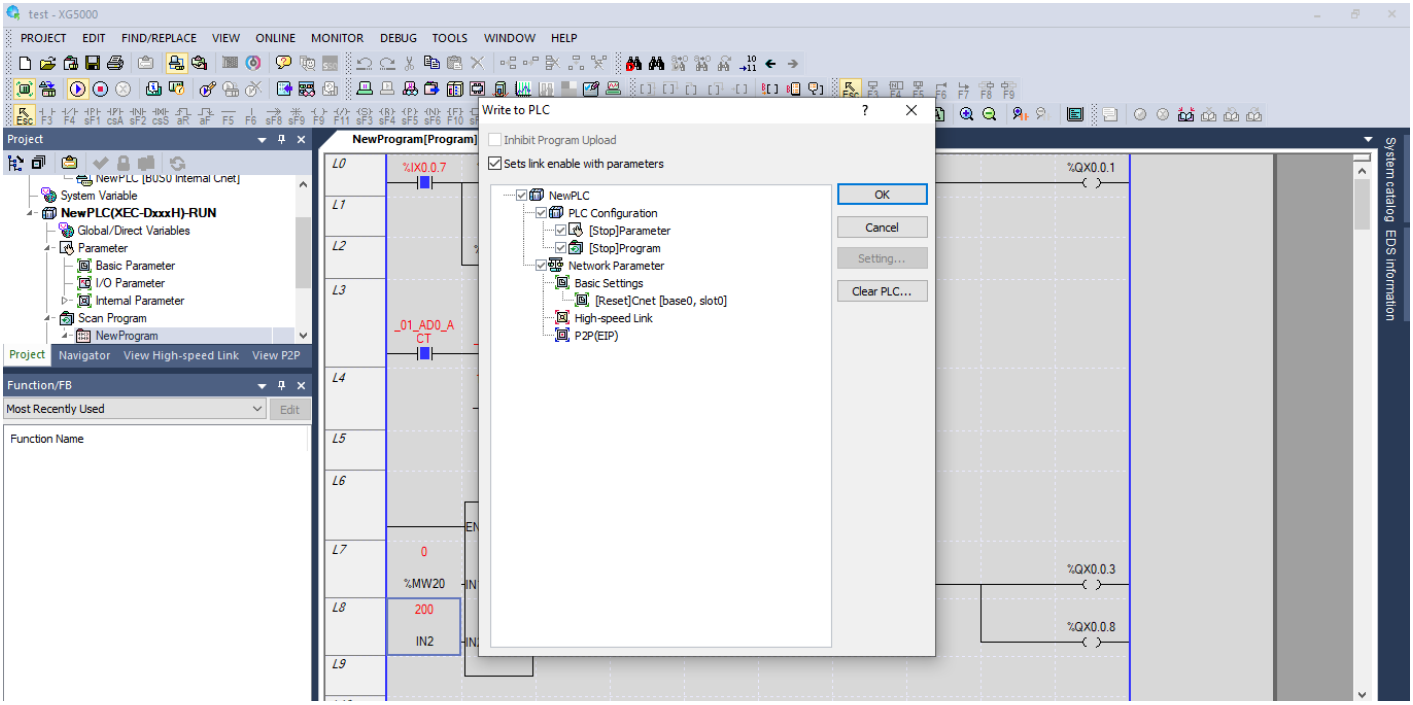


بعد از اتصال PLC به کامپیوتر نیز می‌توان مانند قبل simulator و monitor system را فعال نموده و بصورت نرم افزاری به PLC دستور داد و یا بصورت سخت افزاری با اتصال المان‌ها به PLC فرمان داد.

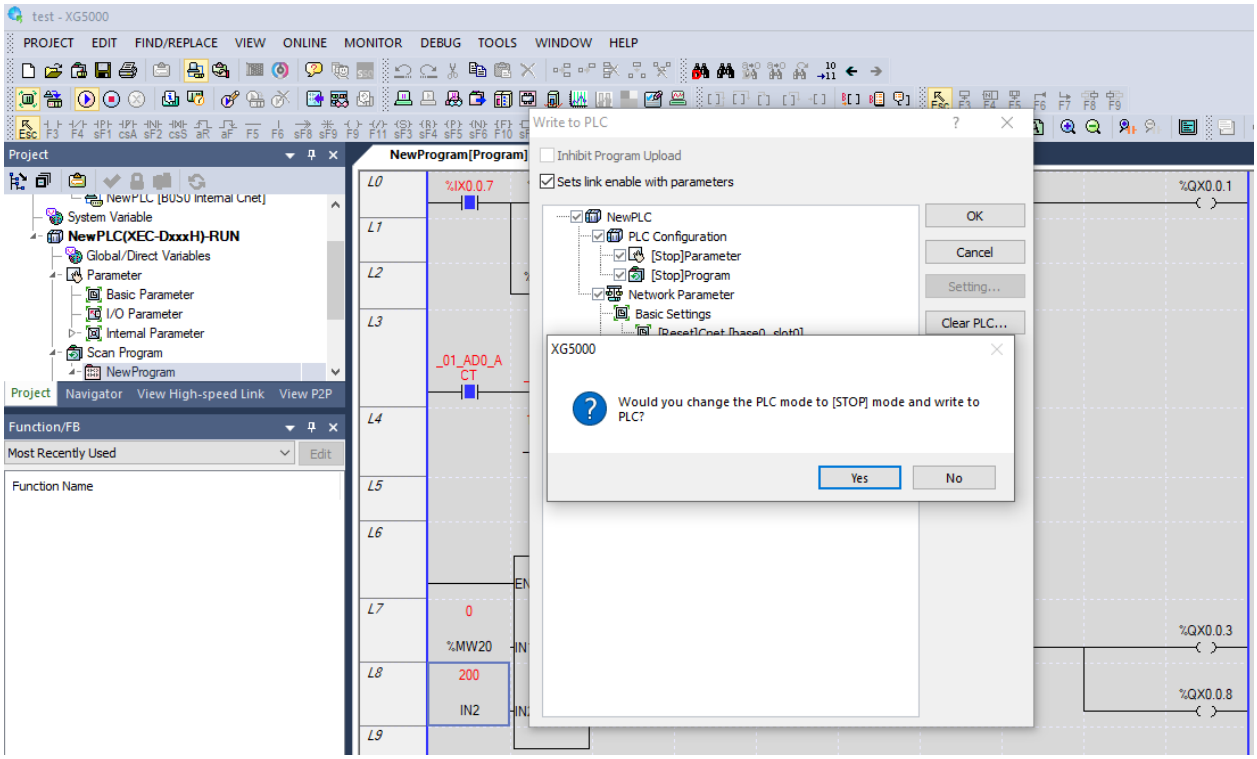
برای دانلود برنامه درون PLC باید در پنجره Online گزینه write را انتخاب کرد تا برنامه به PLC منتقل شود.



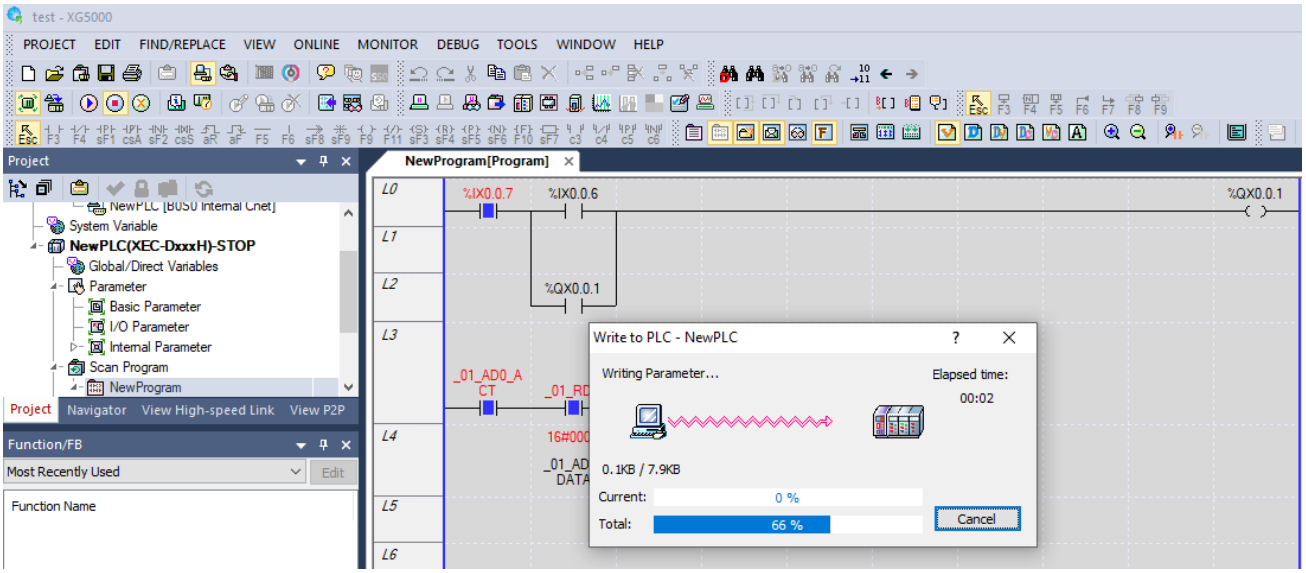
سپس پنجره زیر باز می‌شود و بر روی ok کلیک کنید.



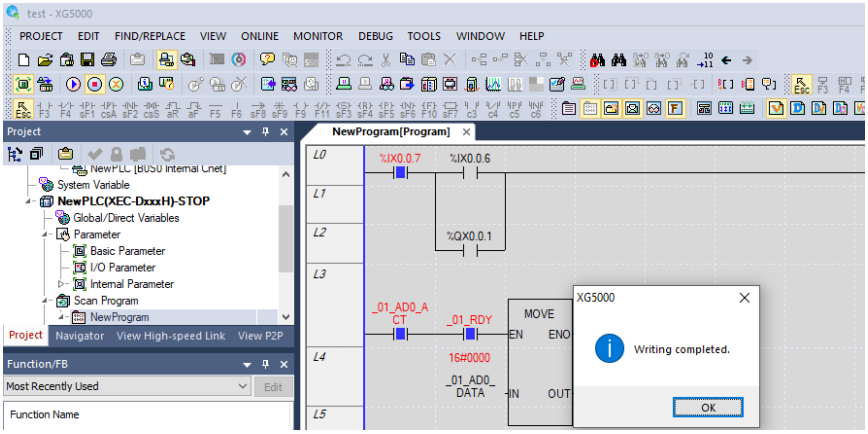
اگر PLC در مد run باشد پنجره زیر باز می‌شود و با انتخابyes ، PLC در مد stop قرار گرفته و برنامه درون PLC بارگذاری می‌شود.



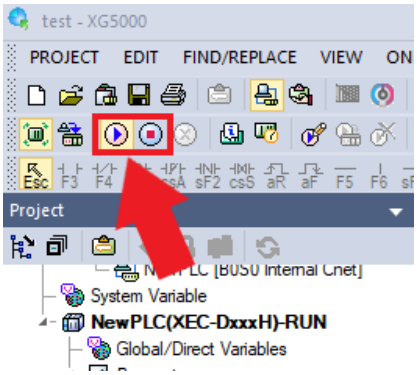
سپس پنجره زیر نمایش داده شده و برنامه درونPLC دانلود می‌شود.



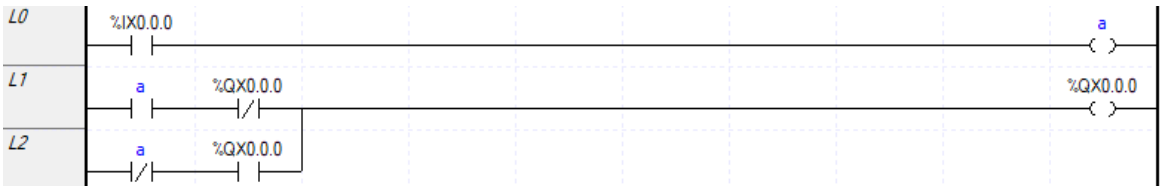
در انتها پنجره زیر نمایش داده می‌شود که نمایانگر دانلود کامل برنامه در PLC است.



سپس PLC را بصورت نرم‌افزاری در مد run قرار داده و برنامه را اجرا نمایید. برای run و stop نرم‌افزاری بر روی گزینه زیر کلیک نمایید.



استپ/استارت با یک شستی

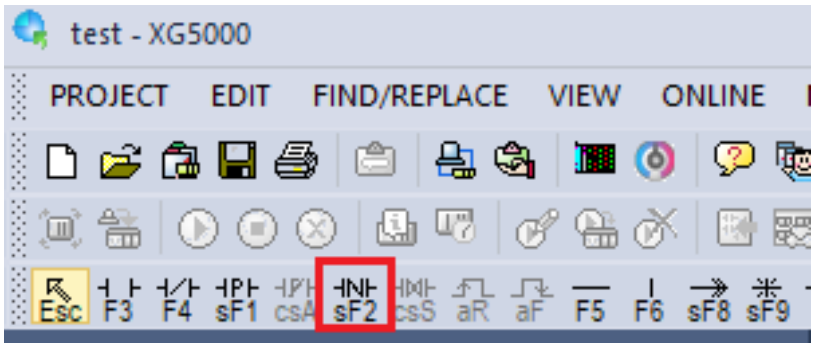


لبه بالارونده (Positive transition-sensing contact)

این کنتاکت که به لبه بالارونده نیز معروف است به شکل زیر است. این عملگر فقط به اندازه یک سیکل اسکن فعال شده و سپس غیرفعال می‌شود. درصورتیکه شستی بصورت دایم فعال باشد تا شستی غیرفعال نشده و دوباره فعال نشود این عملگر فعال نمی‌شود. بسته به نوع برنامه اکثر اوقات برای استارت بجای تیغه باز در برنامه از این عملگر(لبه بالارونده) استفاده می‌کنند. یا برای تشخیص روشن شدن مثلا موتور نیز از این تیغه استفاده می‌کنند.

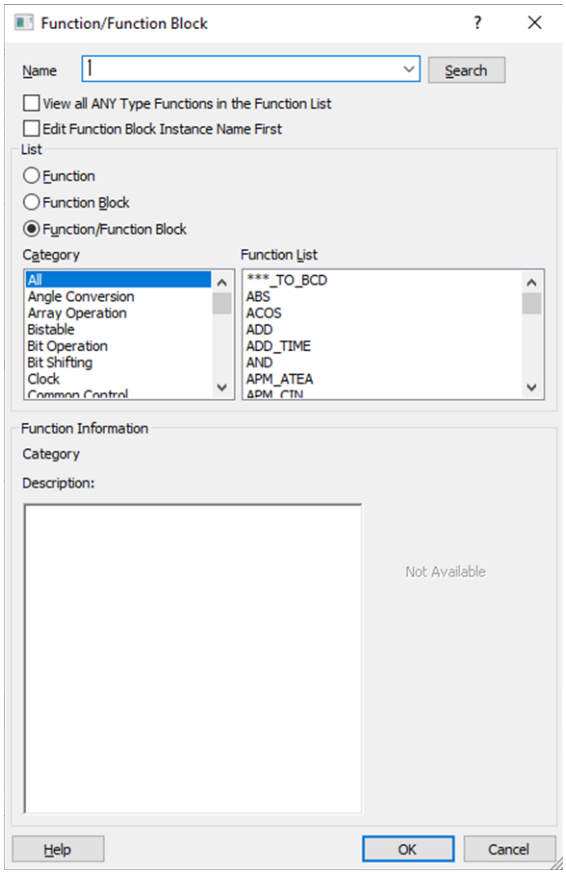
لبه پایین‌رونده (Negative transition-sensing contact)

این کنتاکت که به لبه پایین‌رونده معروف است به شکل زیر است. این عملگر فقط به اندازه یک سیکل اسکن فعال شده و سپس غیرفعال می‌شود. درصورتیکه شستی بصورت دایم فعال باشد تا شستی غیرفعال نشود این عملگر فعال نمی‌شود. بسته به نوع برنامه از این تیغه برای تشخیص خاموش شدن مثلا موتور استفاده می‌گردد.

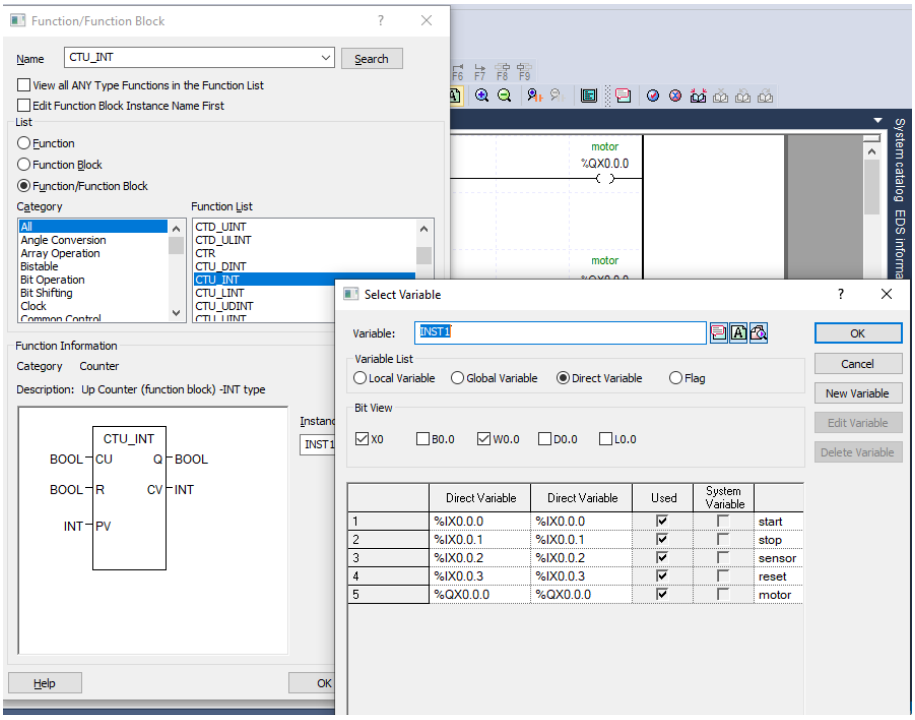


انتخاب توابع

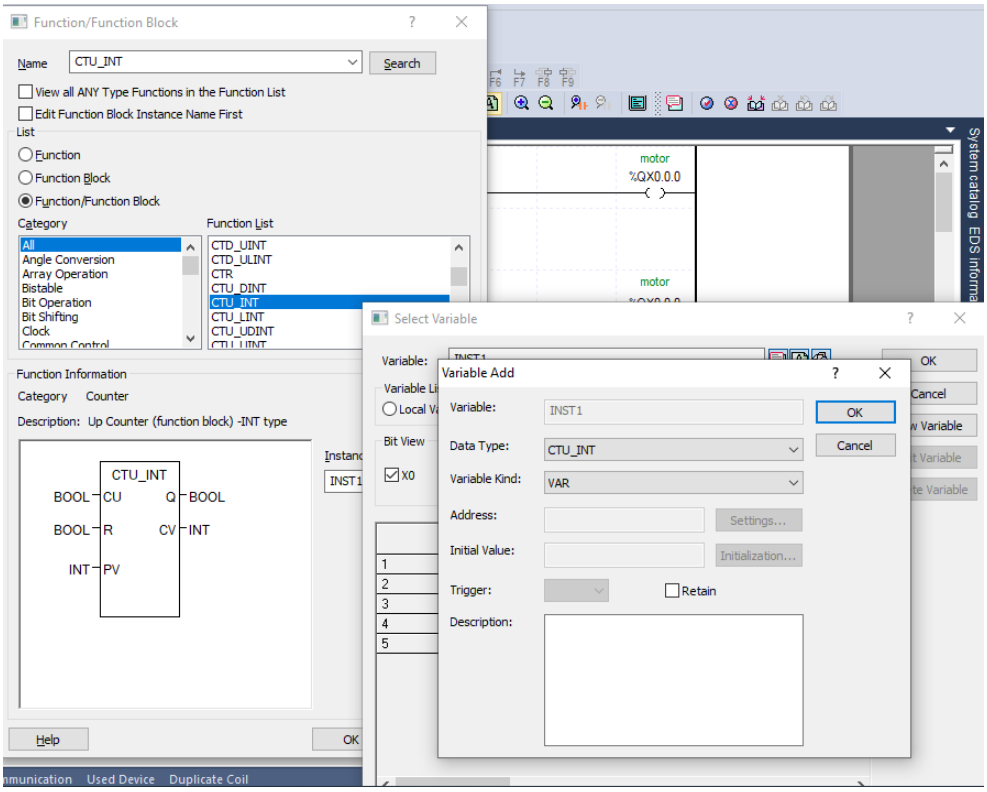
برای انتخاب شمارنده‌ها، تایمرها و سایر توابع مختلف، بر روی کلید F10 بر روی کیبورد کلیک کرده تا پنجره زیر باز شود. سپس در قسمت Name اسم تابع مدنظر را وارد نمایید.



بعد از انتخاب مثلا تابع شمارنده پنجره زیر باز می‌شود.

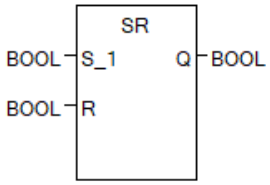


سپس برای خوانا شدن برنامه می‌توان در قسمت variable یک نام به شمارنده اختصاص داد. بعد از انتخاب OK پنجره زیر باز می‌شود. سپس بر روی OK کلیک کرده تا شمارنده در صفحه برنامه‌نویسی قرار بگیرد.



تابع SR

این تابع ترکیبی از set و reset می‌باشد. یعنی با اعمال ورودی به پایه **S** خروجی تا زمانی که پایه R فعال نشود، فعال باقی می‌ماند. زمانی که هر دو ورودی R و S در وضعیت یک منطقی قرار دارند اولویت با SET بوده و خروجی فعال است. با استفاده از این تابع دیگر نیازی به قرار دادن تیغه نگه دارنده موازی با استارت نیست. با انتخاب این تابع و دوبار تایید این تابع به برنامه اصلی اضافه می‌شود.

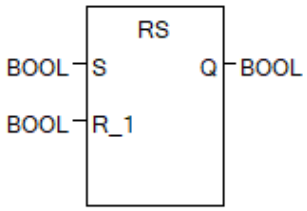


جدول عملکرد این عملگر بصورت زیر است:

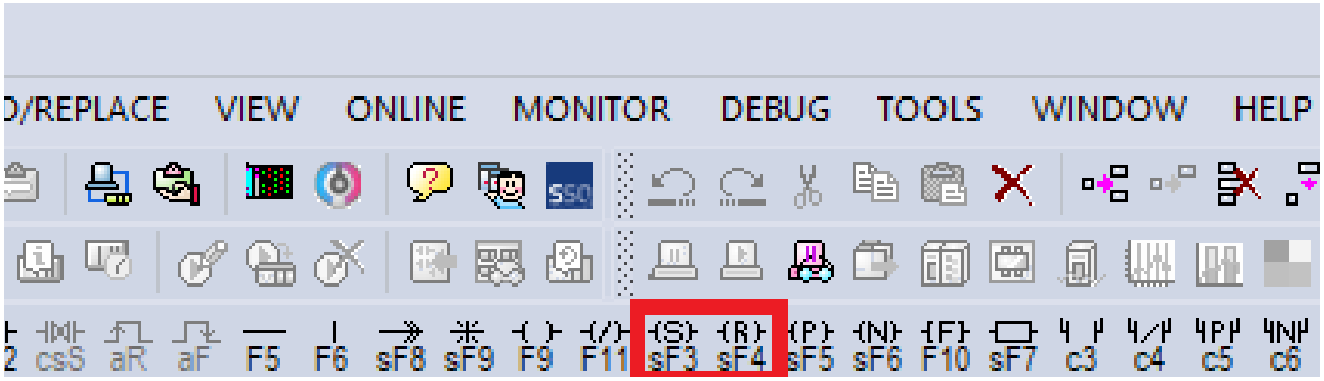
S-1	R	Q
0	0	حالت قبلی
0	1	0
1	0	1
1	1	1

تابع RS

این تابع ترکیبی از set و reset می‌باشد و تفاوت آن با تابع قبلی در اولویت پایه‌ها می‌باشد. یعنی با اعمال ورودی به پایه S خروجی تا زمانی که پایه R فعال نشود، فعال باقی می‌ماند. زمانی که هر دو ورودی SوR در وضعیت یک منطقی قرار دارند اولویت با R بوده و خروجی غیرفعال است.

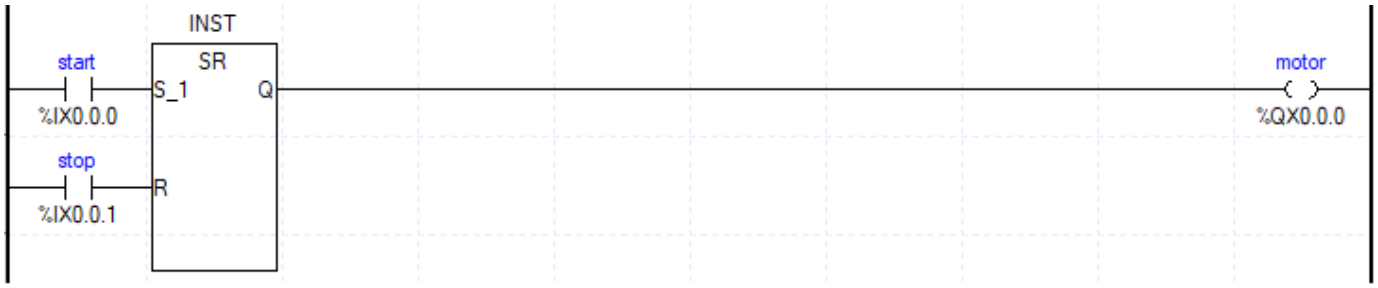


همچنین بجای استفاده از این توابع از کویل آنها نیز می‌توان استفاده نمود.

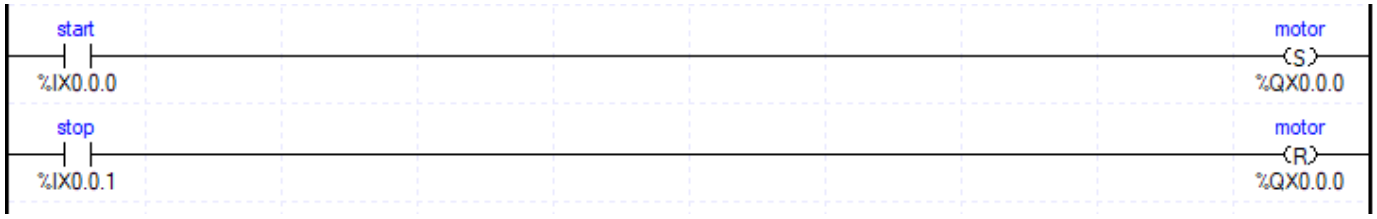


مثال: با زدن استارت موتور بصورت دایم روشن و با زدن استپ موتور خاموش گردد.

شیوه اتصال ورودی/خروجی مانند مثال قبل است اما برنامه آن به دوصورت زیر است:



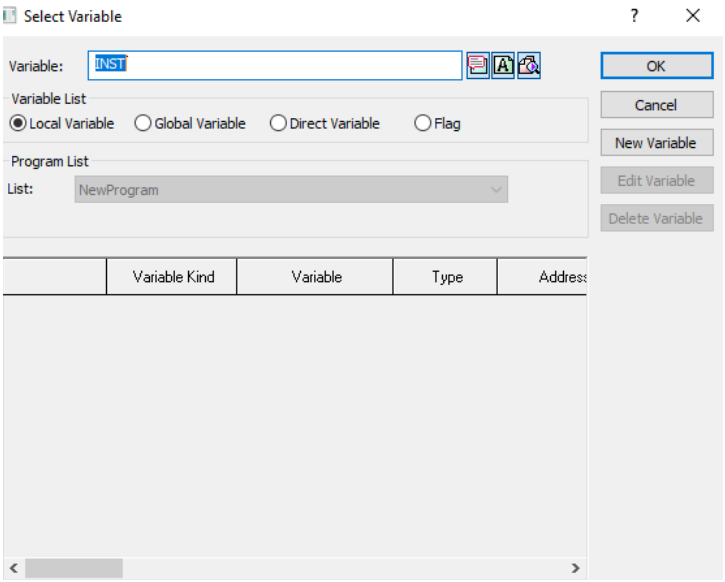
یا



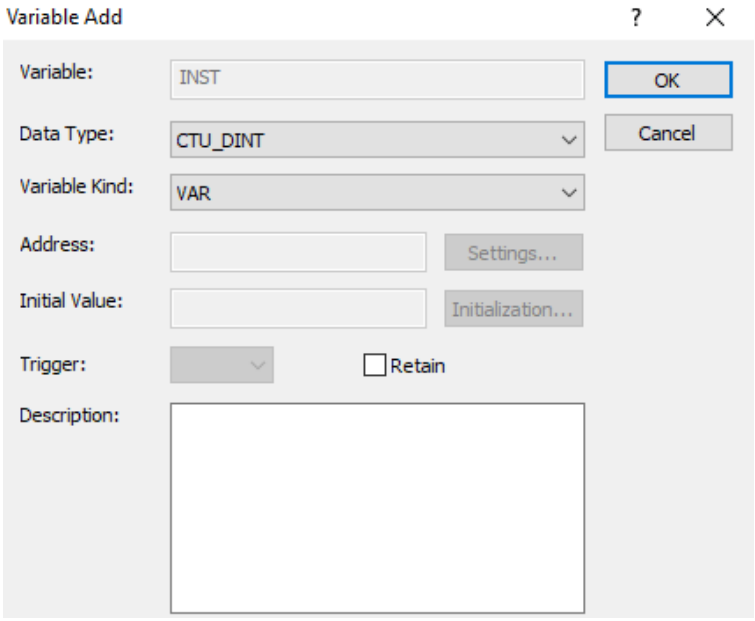
اما این برنامه دو ایراد دارد یکی اینکه اگر شستی استارت گیر کند و خراب شود دیگر نمی‌توان موتور را خاموش کرد و باید از کنتاکت لبه بالارونده استفاده نمود. و دیگری ما از تیغه باز مکانیکی استپ استفاده نموده‌ایم اما در عمل باید از تیغه بسته استپ استفاده نمود و در برنامه نیز از تیغه بسته استفاده نمود تا اگر شستی استپ خراب شد موتور خاموش و ایمنی برقرار باشد.

شمارنده‌ها (Counters)

از این تابع برای شمارش قطعات در فرایند تولید، شمارش تعداد ماشین یا نفرات عبوری و..... استفاده می‌گردد. برای اینکار یک سنسور دیجیتال به این تابع فرمان شمارش صعودی CTU (افزایشیUP Counter)، شمارش نزولی CTD (کاهشیDown Counter)، شمارش صعودی/ نزولی(UP/DOWN Counter) و شمارش بصورت حلقوی (Ring Counter) را می‌دهد. لازم به ذکر است که براساس بازه مقدار قابل شمارش (DINT,INT,UDINT,LINT,UINT,ULINT) می‌توان شمارنده مربوط به آن بازه را انتخاب کرد. پس از انتخاب نوع شمارنده (که بازه قابل شمارش در نام آن قرار دارد) پنجره زیر نمایش داده می‌شود. سپس یک نام برای شمانده در قسمت Variable انتخاب نمایید که بهتر است جهت خوانا بودن برنامه همنام وظیفه شمارنده باشد. در قسمت Variable list نوع متغیر را تنظیم نمایید که محلی، global، متغیر مستقیم، یا از Flagهای درون برنامه PLC باشد. در صورتی که بخواهید متغیر را ویرایش کنید در سمت راست تصویر بر روی Edit Variable کلیک نمایید. برای حذف متغیر بر روی Delete Variable کلیک نمایید. درصورتی که علاوه بر این متغیر بخواهیم متغیر دیگری ایجاد نماییم بر روی New Variable کلیک نمایید. سپس بر روی OK کلیک نمایید.

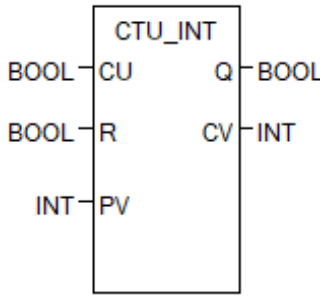


سپس پنجره زیر نمایش داده می‌شود. تنها می‌توان در این پنجره نوع متغیر را در Variable kind تغییر داد و یا این متغیر را Retain یا ماندگار نمود (که با قطع و وصل برق یا Stop و RUN شدن مقدار پاک نمی‌شود).



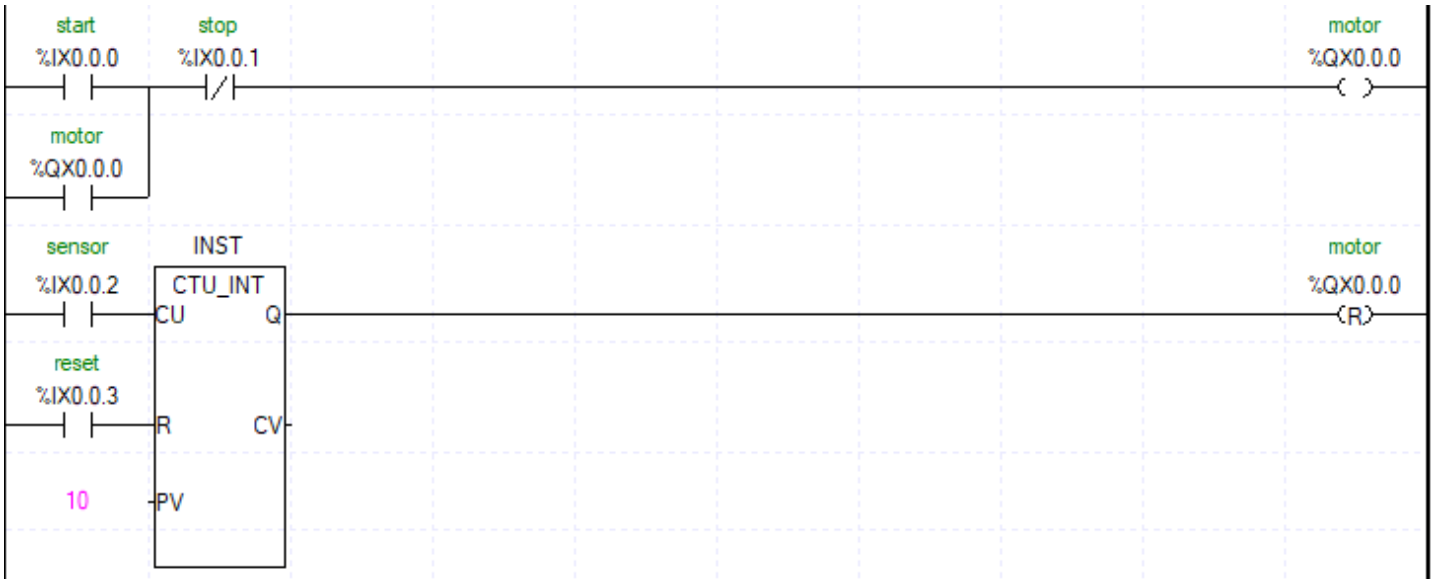
شمارنده صعودی CTU

این شمارنده برای شمارش بصورت افزایشی استفاده می‌گردد. یکی از کاربردهای آن شمارش تعداد قطعات تولید شده می‌باشد. با اتصال یک سنسور دیجیتال یا لیمیت سوییچ به پایه CU مقدار شمارنده افزایش می‌یابد به این صورت که با هر بار یک شدن این پایه مقدار شمارنده یک عدد افزایش می‌یابد. اگر پایه R برابر یک شود مقدار شمارش شده صفر می‌گردد. این پایه باعث ریست شمارنده می‌باشد. در پایهPV مقداری قرار دهید که وقتی مقدار شمارنده برابر این مقدار شد خروجی یا پایه Q فعال شود. اگر مقدار شمارش شده بیشتر یا مساوی با پایه PV شد خروجی فعال می‌شود. پایه CV مقدار شمارش شده را نمایش می‌دهد. مثلا می‌توان تنظیم کرد که اگر ۱۰ قطعه از جلوی سنسور عبور کرد خروجی فعال شده و بسته‌بندی محصول انجام شود. در پایه PV عدد بدون هیچ علامتی قرار می‌گیرد اما عدد باید در بازه شمارنده باشد.

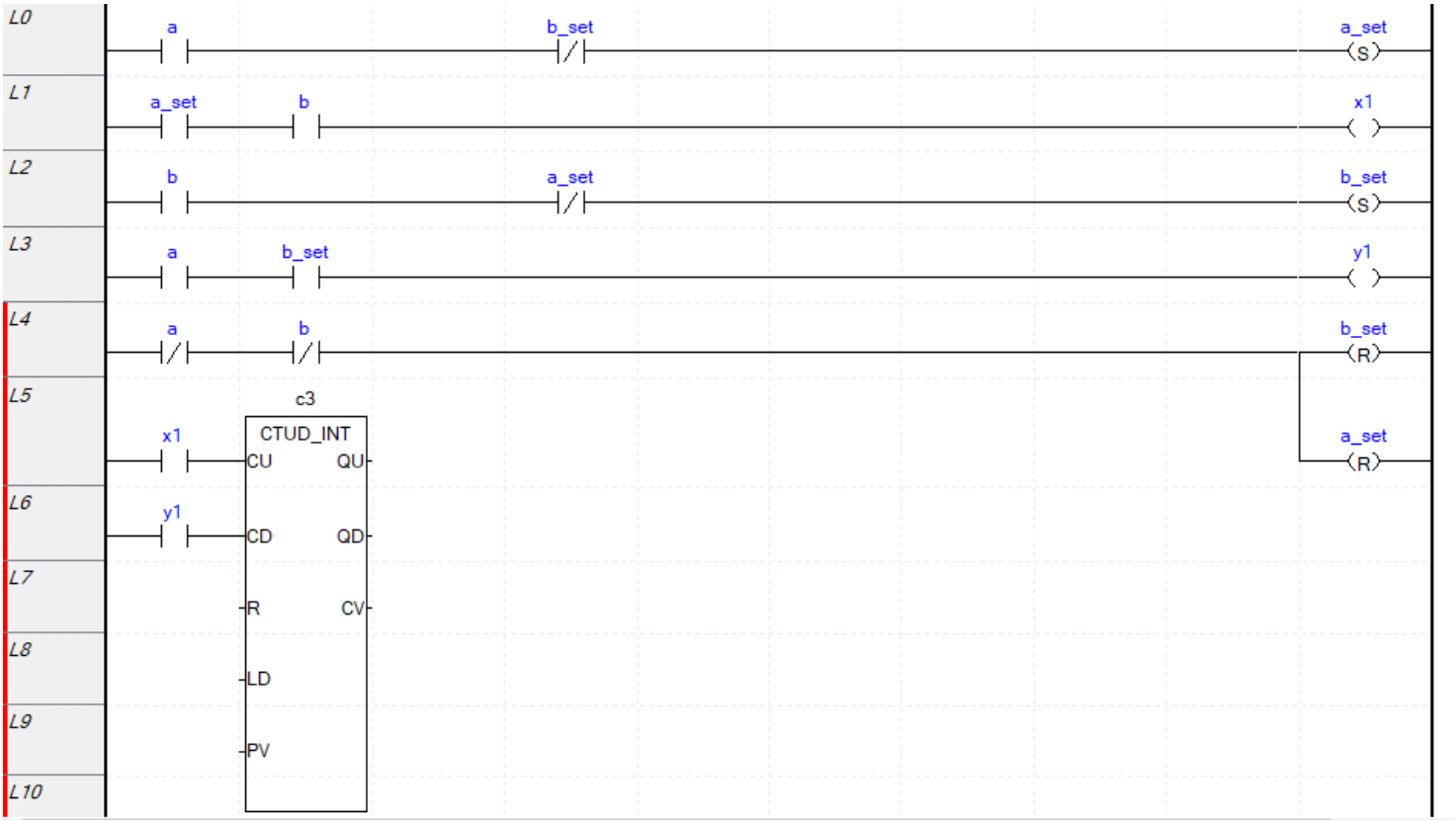


مثال: برنامه‌ای بنویسید که درصورت عبور ۱۰ محصول از جلوی سنسور، موتور خاموش گردد.

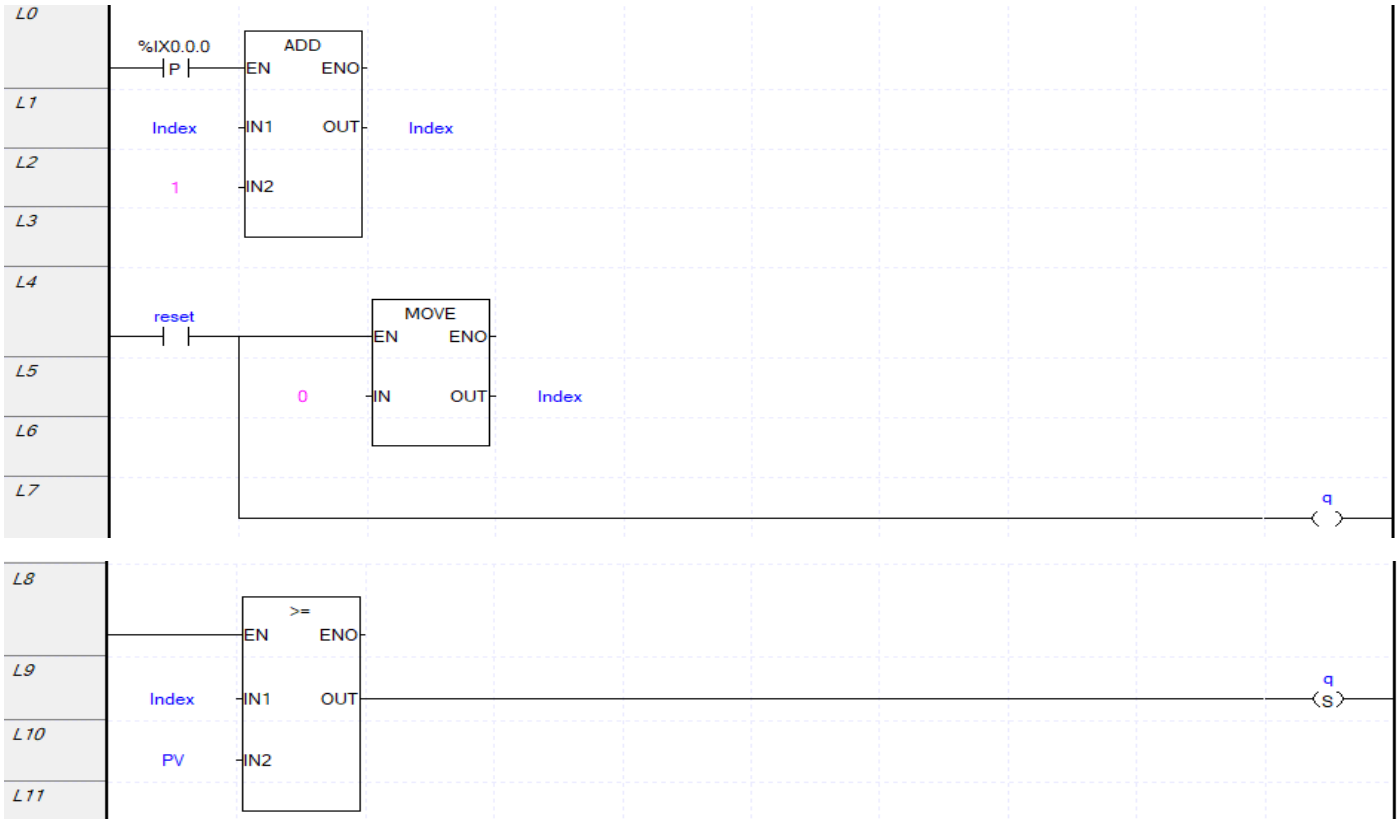
حتما یک شستی یا کلید به پایه ریست متصل نمایید.



مثال: تشخیص عبور از درب ورودی و شمارش تعداد ماشین‌های درون پارکینگ

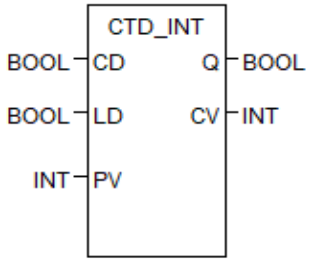


ساخت شمارنده با استفاده از جمع‌کننده



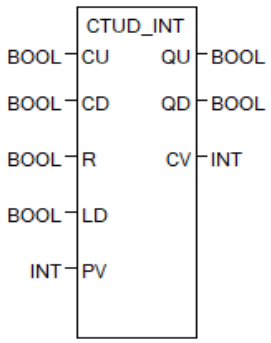
شمارنده نزولی CTD

این شمارنده برای شمارش بصورت کاهشی استفاده می‌گردد. یکی از کاربردهای آن شمارش تعداد قطعات تولید شده جهت بسته‌بندی می‌باشد. با اتصال یک سنسور دیجیتال یا لیمیت سوییچ به پایه CD مقدار شمارنده کاهش می‌یابد به این صورت که با هر بار یک شدن این پایه مقدار شمارنده یک عدد کاهش می‌یابد. در پایهPV مقدار شمارش را وارد کنید. پایه CV مقدار شمارش شده را نمایش می‌دهد. مقدار CVصفر شود خروجی فعال می‌شود. پایه LD برای بارگذاری مقدار PV در CV جهت شمارش معکوس می‌باشد. مثلا می‌توان تنظیم کرد که اگر ۱۰ قطعه از جلوی سنسور عبور کرد خروجی فعال شده و بسته‌بندی محصول انجام شود. در پایه PV عدد بدون هیچ علامتی قرار می‌گیرد اما عدد باید در بازه شمارنده باشد.



شمارنده صعودی/نزولی CTD

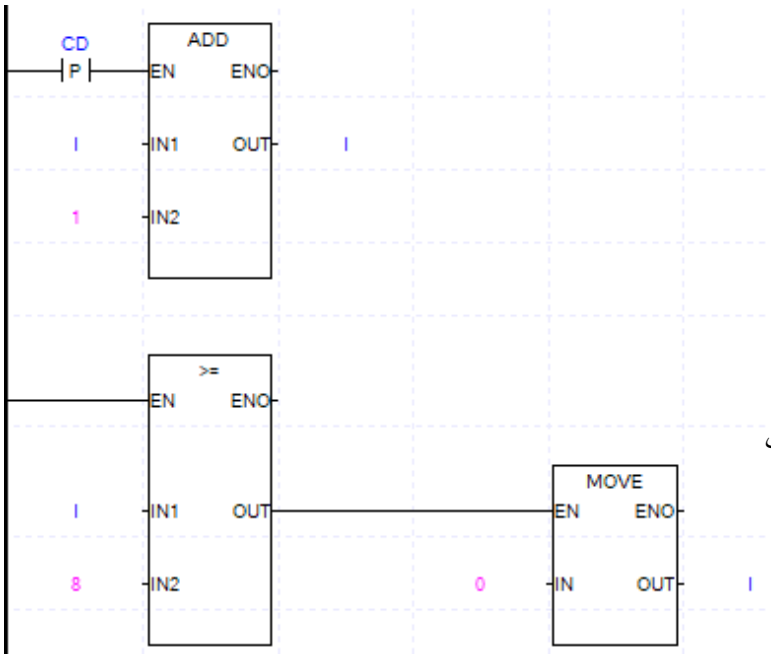
این شمارنده هم می‌تواند بصورت افزایشی شمارش انجام دهد هم بصورت کاهشی. با یک شدن پایه CU مقدار شمارنده افزایش می‌یابد و با یک شدن پایه CD مقدار شمارنده کاهش می‌یابد. اگر هر دو پایه CU وCD همزمان یک شوند شمارنده بصورت صعودی می‌شمارد. با فعال شدن پایه R مقدار شمارش شده صفر می‌گردد. اگر مقدار شمارش شده از مقدار PV بیشتر باشد خروجی Q_u فعال می‌شود و اگر مقدار شمارش شده صفر باشد خروجی Q_D فعال می‌گردد. دقت کنید که تنها به یکی از پایه‌های خروجی می‌توان کویل متصل کرد. با یک شدن پایه LD مقدار PV درون CV قرار گرفته و شمارش از آن عدد به بالا و یا به پایین انجام می‌شود. در صورتی که پایه LD یک بماند خروجی Qu فعال ولی شمارشی انجام نمی‌پذیرد. حتما در پایه CV نیز یک متغیر قراردهید در غیراینصورت برنامه خطا می‌دهد و ستون کنار سمت چپ به رنگ قرمز تبدیل می‌شود.



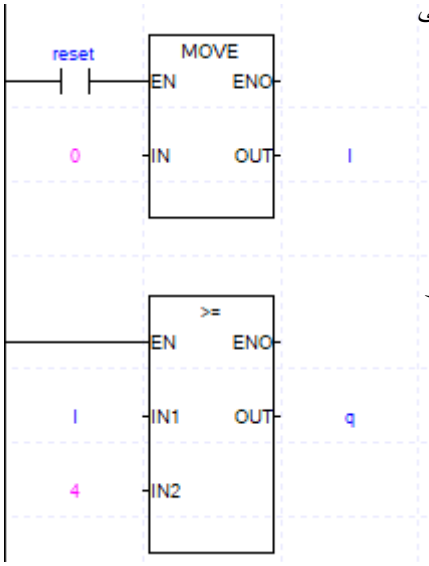
شمارنده حلقوی

این شمارنده بصورت بلاک وجود ندارد ولی با بلاک های دیگر می‌توان آنرا ایجاد کرد و به علت کاربرد زیاد در برنامه‌نویسی PLC بیان می‌گردد. یکی از کاربردهای این شمارنده در تنظیم بسته‌بندی محصول است

مثال: بعد از عبور ۸محصول بسته‌بندی شروع شده و مقدار شمارنده صفر شود و پس از آن دوباره از صفر تا ۸ شمارش شود و این فرایند بصورت دایم تکرار شود.



به ازای هشتمین پالس این عدد در خروجی شمارنده نمایش داده می‌شود.



درصورت یک شدن متغیر ریست خروجی شمارنده صفر می‌شود.

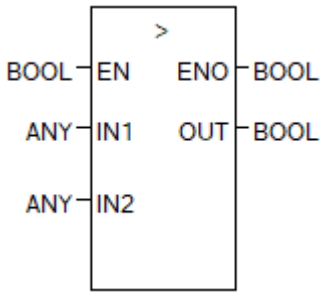
هرگاه مقدار خروجی شمارنده برابر یا بزرگتر از مثلا عدد۴ شد مقدار q برابر یک می‌شود.

مقایسه‌گرها

از این توابع برای مقایسه دو یا چند کمیت با یکدیگر استفاده می‌شود که شامل بزرگتر بودن، کوچکتر بودن، بزرگتر یا مساوی بودن، کوچکتر یا مساوی بودن، مساوی بودن، نامساوی بودن می‌باشد.

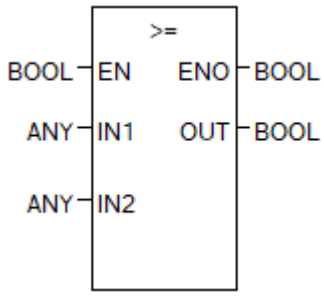
تابع بزرگتر بودن

این عملگر مخفف greater than یعنی بزرگتر بودن است که آنرا با GT و یا " > " نمایش می‌دهند. تعداد ورودی‌های این تابع می‌تواند حداقل ۲ عدد و حداکثر ۸ عدد باشد. نوع داده همه ورودی‌ها باید یکسان باشد مثلا همه INT. هر ۲۰نوع داده مجاز می‌توانند با یکدیگر مقایسه شوند. خروجی این تابع صفر و یک می‌باشد. ابتدا ورودی IN1 با IN2 مقایسه می‌شود اگر IN1 بزرگتر بود سپس IN2 با IN3 مقایسه می‌شود و اگر IN2 بزرگتر بود همینطور تا آخرین ورودی با یکدیگر مقایسه می‌شوند سپس خروجی برابر یک می‌گردد.



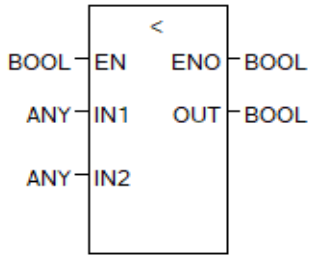
تابع بزرگتر یا مساوی بودن

این عملگر مخفف greater than or equal to یعنی بزرگتر یا مساوی بودن است که آنرا با GE و یا " =< " نمایش می‌دهند. تعداد ورودی‌های این تابع می‌تواند حداقل ۲ عدد و حداکثر ۸ عدد باشد. نوع داده همه ورودی‌ها باید یکسان باشد مثلا همه INT. هر ۲۰نوع داده مجاز می‌توانند با یکدیگر مقایسه شوند. خروجی این تابع صفر و یک می‌باشد. ابتدا ورودی IN1 با IN2 مقایسه می‌شود اگر IN1 بزرگتر یا مساویIN2 بود سپس IN2 با IN3 مقایسه می‌شود و اگر IN2 بزرگتر یا مساویIN3 بود همینطور تا آخرین ورودی با یکدیگر مقایسه می‌شوند سپس خروجی برابر یک می‌گردد.



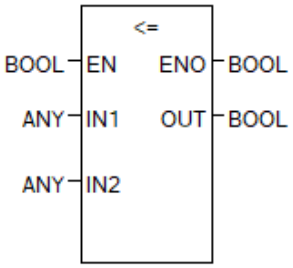
تابع کوچکتر بودن

این عملگر مخفف Less than یعنی کوچکتر بودن است که آنرا با LT و یا " < " نمایش می‌دهند. تعداد ورودی‌های این تابع می‌تواند حداقل ۲ عدد و حداکثر ۸ عدد باشد. نوع داده همه ورودی‌ها باید یکسان باشد مثلا همه INT. هر ۲۰نوع داده مجاز می‌توانند با یکدیگر مقایسه شوند. خروجی این تابع صفر و یک می‌باشد. ابتدا ورودی IN1 با IN2 مقایسه می‌شود اگر IN1 کوچکتر از IN2 بود سپس IN2 با IN3 مقایسه می‌شود و اگر IN2 کوچکتر از IN3 بود IN3 با IN4 مقایسه می‌شود و همینطور تا آخرین ورودی با یکدیگر مقایسه می‌شوند سپس خروجی برابر یک می‌گردد.



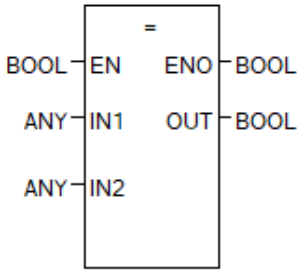
تابع کوچکتر یا مساوی بودن

این عملگر مخفف Less than or equal to یعنی بزرگتر یا مساوی بودن است که آنرا با LE و یا " <= " نمایش می‌دهند. تعداد ورودی‌های این تابع می‌تواند حداقل ۲ عدد و حداکثر ۸ عدد باشد. نوع داده همه ورودی‌ها باید یکسان باشد مثلا همه INT. هر ۲۰نوع داده مجاز می‌توانند با یکدیگر مقایسه شوند. خروجی این تابع صفر و یک می‌باشد. ابتدا ورودی IN1 با IN2 مقایسه می‌شود اگر IN1 کوچکتر یا مساویIN2 بود سپس IN2 با IN3 مقایسه می‌شود و اگر IN2 کوچکتر یا مساویIN3 بود همینطور تا آخرین ورودی با یکدیگر مقایسه می‌شوند سپس خروجی برابر یک می‌گردد.



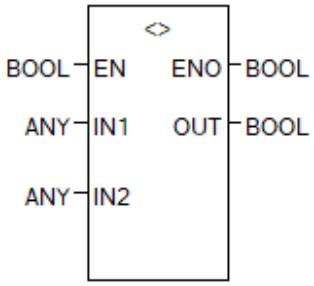
تابع مساوی بودن

این عملگر مخفف Equal to یعنی مساوی بودن است و آنرا با EQ و یا "==" نمایش می‌دهند. تعداد ورودی‌های این تابع می‌تواند حداقل ۲ عدد و حداکثر ۸ عدد باشد. نوع داده همه ورودی‌ها باید یکسان باشد مثلا همه INT. هر ۲۰نوع داده مجاز می‌توانند با یکدیگر مقایسه شوند. در این عملگر خروجی زمانی یک می‌شود که تمامی ورودی‌ها با یکدیگر برابر باشند.



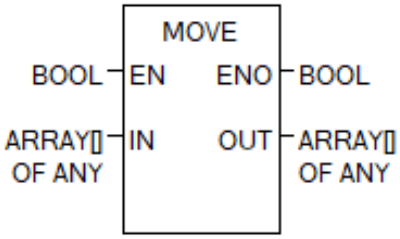
تابع نامساوی بودن

این عملگر مخفف Not Equal to یعنی نامساوی بودن است و آنرا با NE و یا "<>" نمایش می‌دهند. تعداد ورودی‌های این تابع ۲ورودی است. نوع داده باید یکسان باشد مثلا هردو INT. هر ۲۰نوع داده مجاز می‌توانند با یکدیگر مقایسه شوند. در این عملگر خروجی زمانی یک می‌شود که ۲ ورودی با یکدیگر برابر نباشند.



تابع MOVE

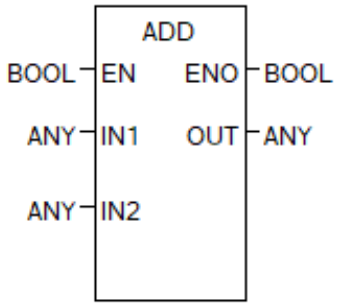
از این عملگر برای اختصاص دادن حافظه، مقدار، یا متغیر به یک حافظه یا متغیر استفاده می‌شود. نوع داده ورودی و خروجی باید با یکدیگر مشابه باشد. مقدار ورودی به خروجی اختصاص می‌یابد.



عملگرهای ریاضی

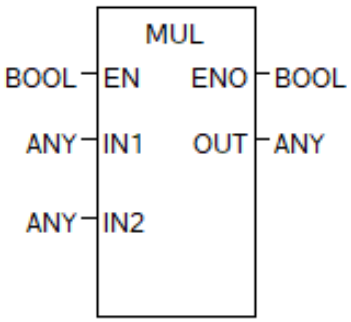
تابع جمع‌کننده (ADD) :

این تابع مقدار دو یا چند ورودی را با یکدیگر جمع می‌کند و نتیجه را در حافظه خروجی نمایش می‌دهد. نوع داده همه ورودی‌ها باید یکسان بوده و تمام ورودی‌ها باید از نوع داده‌های عددی باشند. تعداد ورودی‌های این تابع می‌تواند حداقل ۲ عدد و حداکثر ۸ عدد باشد. در انتخاب نوع داده به بازه آن داده توجه نمایید که حاصل‌جمع از بازه داده بیشتر نشود. درصورت اجرا نشدن تابع مقدار خروجی آن برابر مقدار قبلی است.



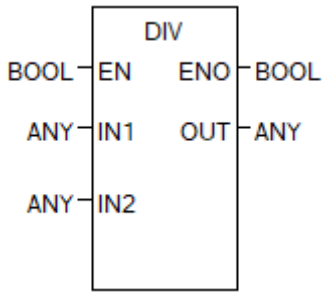
تابع ضرب (MUL) :

این تابع مقدار دو یا چند ورودی را در یکدیگر ضرب می‌کند و نتیجه را در حافظه خروجی نمایش می‌دهد. نوع داده همه ورودی‌ها باید یکسان بوده و تمام ورودی‌ها باید از نوع داده‌های عددی باشند. تعداد ورودی‌های این تابع می‌تواند حداقل ۲ عدد و حداکثر ۸ عدد باشد. در انتخاب نوع داده به بازه آن داده توجه نمایید که حاصل‌ضرب از بازه داده بیشتر نشود. درصورت اجرا نشدن تابع مقدار خروجی آن برابر مقدار قبلی است.



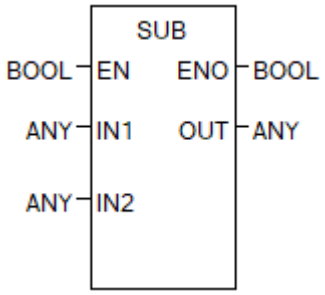
تابع تقسیم (DIV) :

این تابع مقدار ورودی IN1 را بر IN2 تقسیم می‌کند و نتیجه را در حافظه خروجی نمایش می‌دهد. نوع داده ورودی‌ها باید یکسان بوده و ورودی‌ها باید از نوع داده‌های عددی باشند. در انتخاب نوع داده به بازه آن داده توجه نمایید که حاصل‌تقسیم قسمت صحیح تقسیم می‌باشد غیر از اینکه ورودی و خروجی بصورت REAL تعریف شده باشند. درصورت اجرا نشدن تابع مقدار خروجی آن برابر مقدار قبلی است.



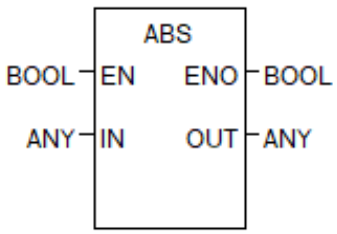
تابع تفریق (SUB) :

این عملگر دارای دو ورودی است که حاصل IN1-IN2 در خروجی نمایش داده می‌شود. نوع داده ورودی و خروجی باید یکسان باشد. درصورت اجرا نشدن تابع مقدار خروجی آن برابر مقدار قبلی است.



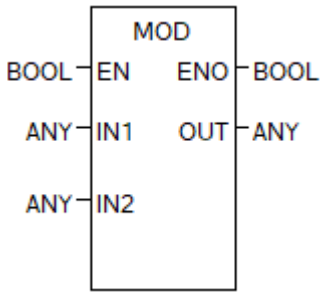
تابع قدر مطلق (ABS) :

این عملگر فقط دارای یک ورودی بوده و خروجی این تابع قدرمطلق ورودی می‌باشد. نوع داده ورودی و خروجی باید یکسان باشد.



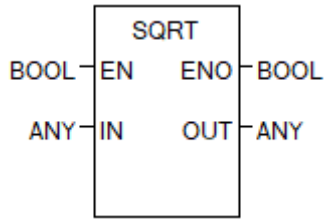
تابع باقیمانده خارج قسمت (MOD) :

این عملگر باقیمانده خارج قسمت دو عدد صحیح ورودی را محاسبه می‌کند. IN1 را بر IN2 تقسیم می‌کند و باقیمانده تقسیم را در خروجی نمایش می‌دهد.



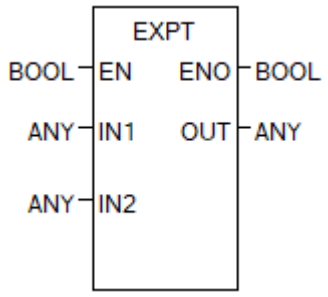
تابع جذر (SQRT) :

این عملگر دارای یک ورودی و یک خروجی بوده و جذر ورودی را در خروجی نمایش می‌دهد. نوع داده ورودی و خروجی باید یکی باشد و حتما از نوع any-real باشد.



تابع توان (EXPT) :

این تابع دو ورودی و یک خروجی دارد ورودی IN1 به توان ورودی IN2 می‌رسد و در خروجی نمایش داده می‌شود. نوع داده ورودی و خروجی باید یکی باشد.

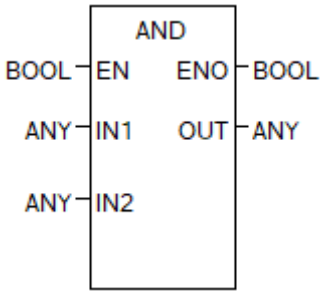


عملگرهای منطقی

عملگر منطقی AND :

هنگامی که تمام ورودی‌های این عملگر برابر یک باشد خروجی آن فعال می‌گردد. لازم به ذکر است که تمام ورودی‌های این تابع باید بصورت بیتی یعنی صفر یا یک باشد. برای مثال اگر استارت ۱ و استارت۲ همزمان فشرده شدند دستگاه عمل نماید (مثلا دستگاه پرس). جدول عملکرد این تابع به شکل زیر است.

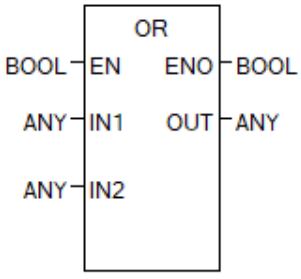
IN1	IN2	OUT
0	0	0
0	1	0
1	0	0
1	1	1



عملگر منطقی OR :

هنگامی که یکی از ورودی‌های این تابع برابر یک باشد خروجی این تابع فعال می‌شود. لازم به ذکر است که تمام ورودی‌های این تابع باید بصورت بیتی یعنی صفر یا یک باشد. برای مثال اگر استارت در نقطه۱ یا استارت در نقطه ۲ فشرده شود موتور روشن شود. جدول عملکرد این تابع به شکل زیر است.

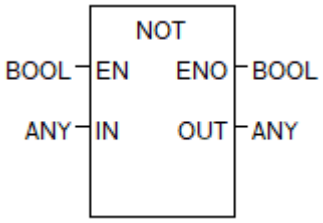
IN1	IN2	OUT
0	0	0
0	1	1
1	0	1
1	1	1



عملگر منطقی NOT :

این عملگر ورودی را NOT می‌کند یعنی اگر ورودی برابر یک باشد خروجی صفر و اگر ورودی برابر صفر باشد خروجی را یک می‌کند. مثلا اگر شستی فشرده نشود موتور روشن و با فشردن شستی موتور خاموش گردد.

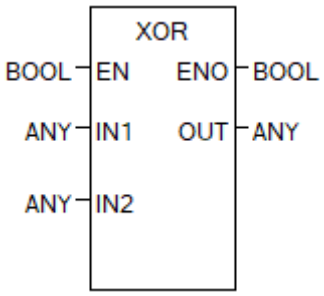
IN	OUT
0	1
1	0



عملگر منطقی XOR :

هنگامی خروجی این عملگر فعال می‌شود که تعداد یک‌های ورودی‌ها فرد باشد. لازم به ذکر است که تمام ورودی‌های این تابع باید بصورت بیتی یعنی صفر یا یک باشد. مثلا اگر سه شستی فعال شوند موتور روشن و اگر ۴ شستی فعال شوند موتور خاموش گردد.

IN1	IN2	OUT
0	0	0
0	1	1
1	0	1
1	1	0

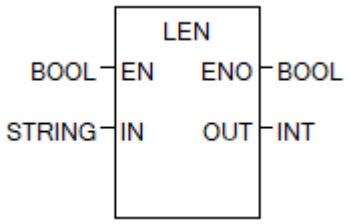


توابع مربوط به String :

توجه: یک string باید بین دو single quote(‘ ’) قرار بگیرد.

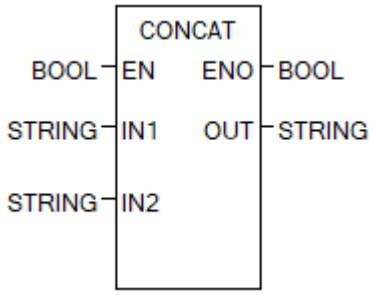
تابع LEN

تعداد کاراکترهای یک زنجیره یا string را شمرده (طول یک زنجیره بر حسب تعداد کاراکتر) و در متغیر خروجی ذخیره می‌گردد. نوع داده خروجی باید از نوع USINT باشد.



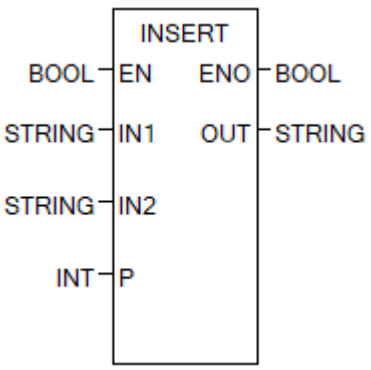
تابع CONCAT

این عملگر وظیفه بهم چسباندن چند string را به عهده دارد و دارای حداقل۲ و حداکثر ۸ ورودی است.
تمامی stringها به ترتیب از بالا به پایین چسبانده و از چپ به راست در خروجی نمایش می‌دهد.



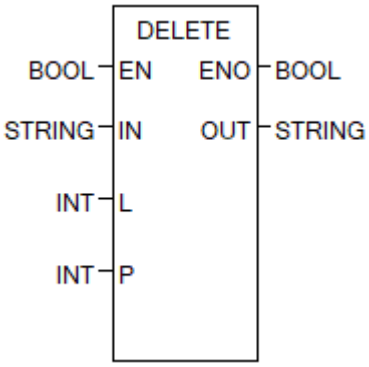
تابع INSERT

این عملگر یک string را درون یک string دیگر قرار می‌دهد. متغیر IN2 را درون IN1 قرار می‌دهد. در پایه P تعریف می‌کنیم که از چندمین کاراکتر string ورودی IN1 ، ورودیIN2 درون IN1 قرار گیرد.



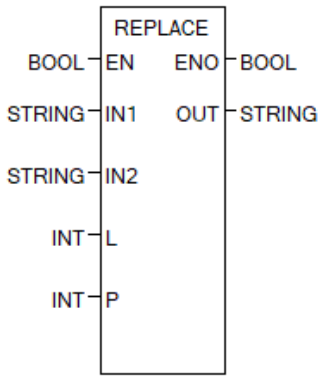
تابع DELETE

این عملگر قسمتی از string را حذف می‌کند. در پایه IN ، string را وارد نمایید سپس در پایه L تعداد متغیرهایی که باید حذف شوند را وارد نمایید و سپس در پایه P تعیین می‌کنیم از چندمین کاراکتر پاک شود.



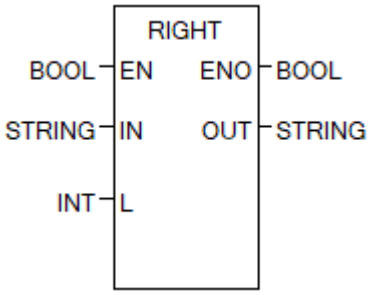
تابع REPLACE

این عملگر بخشی از string را درون string دیگر جایگزین می‌کند. در پایه IN1 ، string اول را وارد نمایید در پایه IN2، Stringدومی که قرار است وارد String اول شود را وارد نمایید. در پایه L تعداد متغیرهایی که باید جایگزین شوند را وارد نمایید و سپس در پایه P تعیین می‌کنیم از چندمین کاراکتر جایگزین شود.



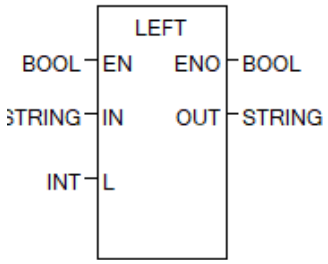
تابع RIGHT

این عملگر تعداد مشخصی از کاراکترها را که در پایه L تعیین می‌کنیم را از سمت راست یک string جدا می‌کند و در خروجی نمایش می‌دهد.



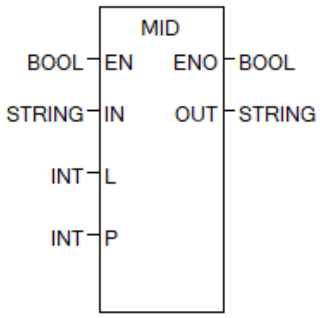
تابع LEFT

این عملگر تعداد مشخصی از کاراکترها را که در پایه L تعیین می‌کنیم از سمت چپ یک string جدا می‌کند و در خروجی نمایش می‌دهد.



تابع MID

این عملگر تعداد مشخصی از کاراکترها را که در پایه L تعیین می‌کنیم از وسط یک string جدا می‌کند و در خروجی نمایش می‌دهد. در پایه P تعیین می‌کنیم از چندمین کاراکتر جدا شود.



مالتی‌پلکسر (MUX)

این عملگر دارای یک پایه کنترل بنام K بوده و حداکثر ۷ورودی دارد. با توجه به مقداری که به K داده می‌شود یکی از متغیرهای ورودی‌ها به خروجی واگذار می‌شود. درصورتی که K>6 و یا K<0 باشد مقدار خروجی برابر IN0 می‌شود و عملگر خطا می‌دهد سپس ENO برابر صفر می‌گردد.

K=0→ out=IN0

K=1→ out=IN1

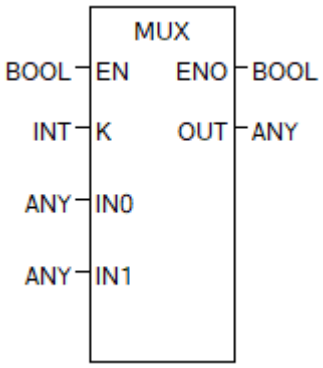
K=2→ out=IN2

K=3→ out=IN3

K=4→ out=IN4

K=5→ out=IN5

K=6→ out=IN6

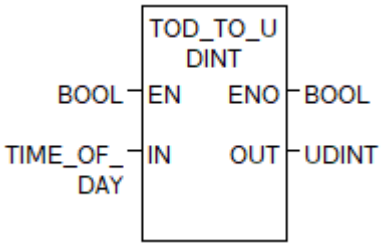


تبدیل داده‌ها

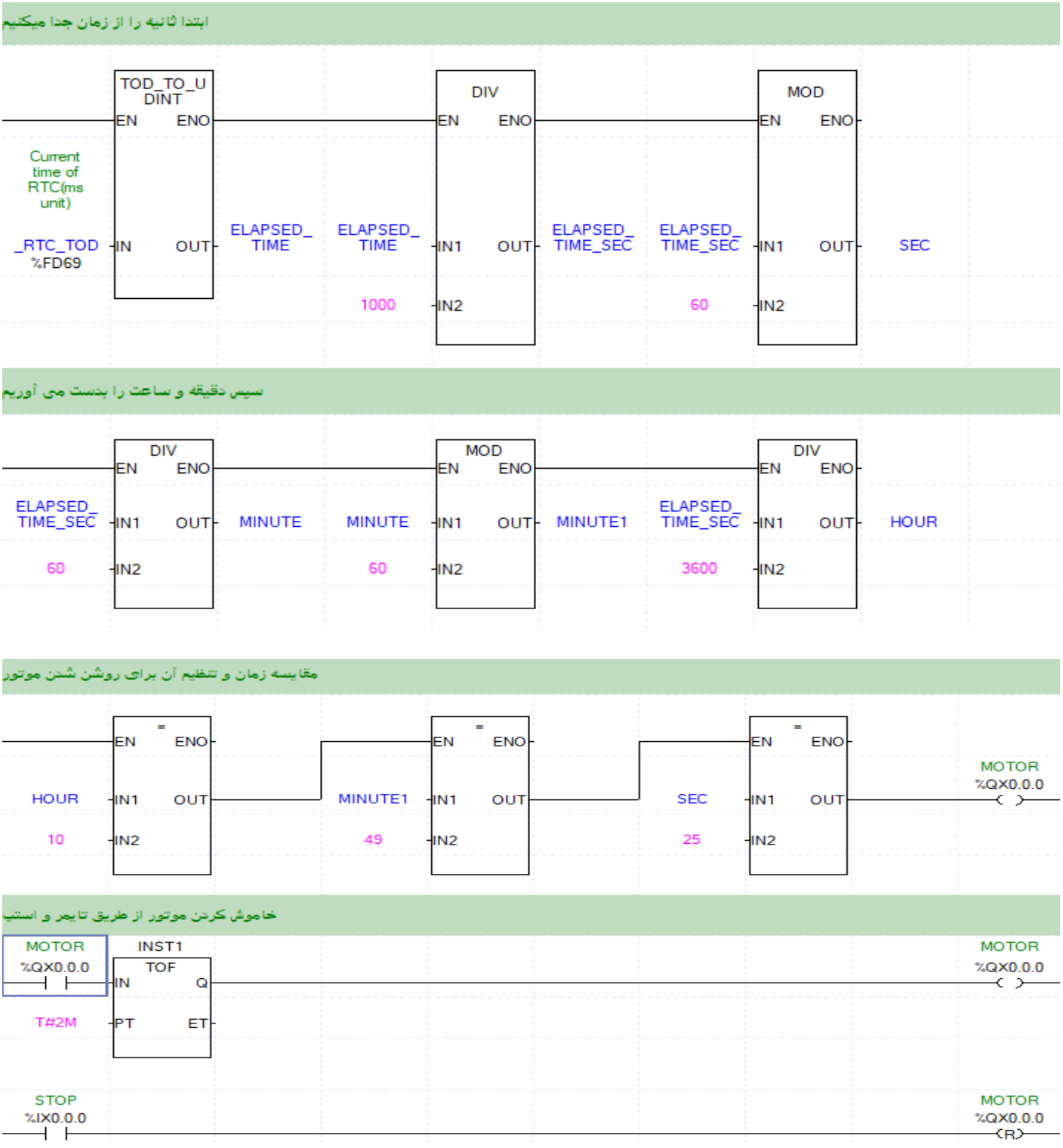
از این توابع برای تبدیل نوعی از داده به نوع دیگر داده استفاده می‌گردد

تبدیل زمان روز به داده نوع UDINT

با استفاده از فلگ RTC_TOD_ می‌توان زمان روز را به این تابع داده و خروجی تابع زمان سپری شده از صفر بامداد تا آن لحظه را برحسب میلی‌ثانیه نمایش می‌دهد. سپس این زمان را می‌توان به ثانیه،دقیقه و ساعت تبدیل کرده و برای زمانبندی انجام عملکرد خاصی مثلا روشن شدن موتور در ساعتی خاص استفاده شود.

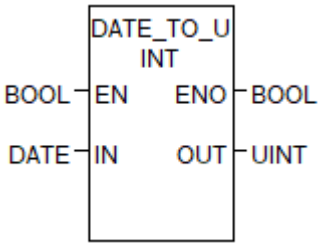


مثال: موتور در ساعت 10:49:25 روشن شده و پس از ۲دقیقه خاموش گردد.



تابع تبدیل تعداد روزهای سپری شده به UINT

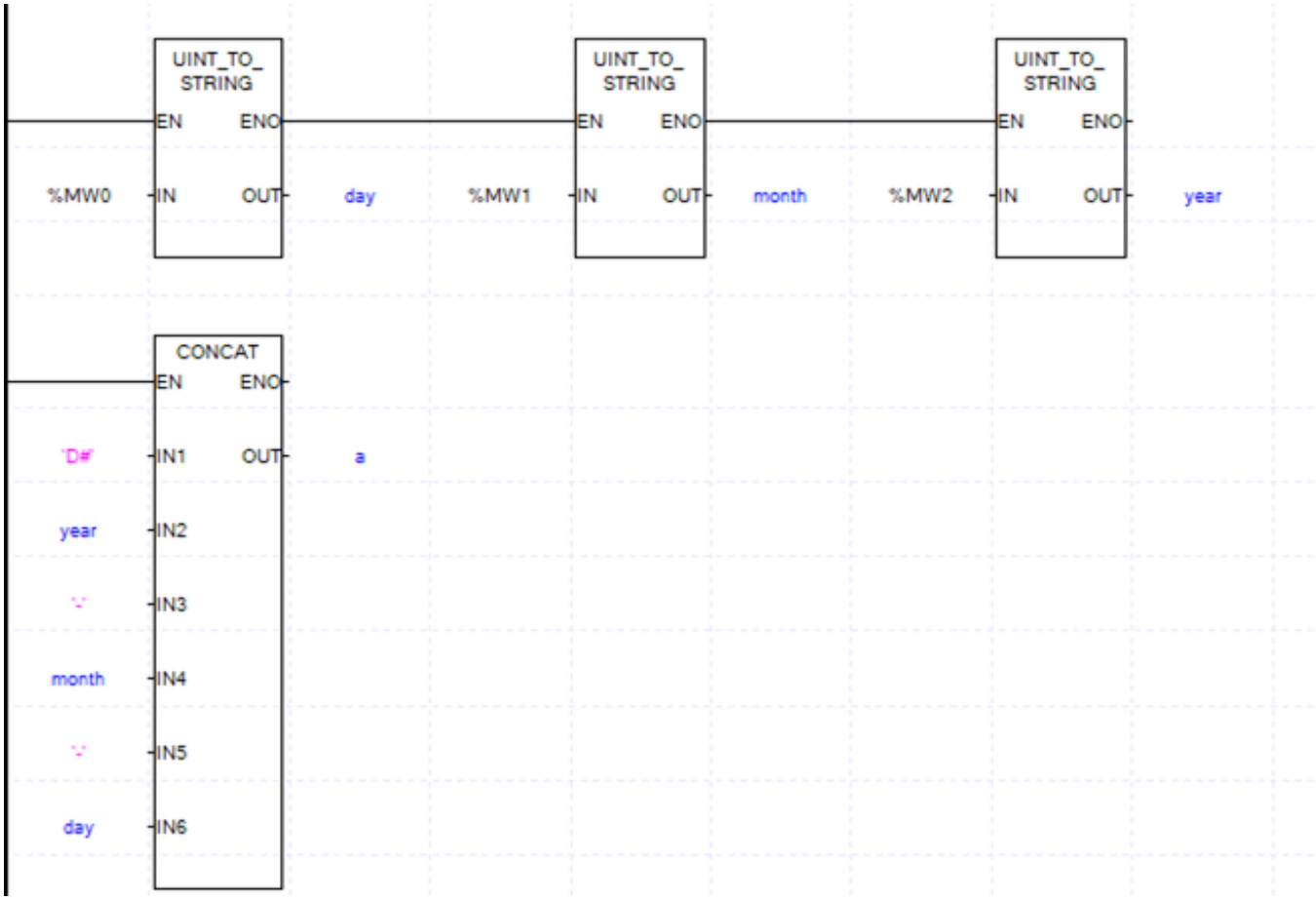
این تابع تعداد روزهای سپری شده از تاریخ ۱۹۸۴/۱/۱ تا زمان حال حاضر را نشان می‌دهد.

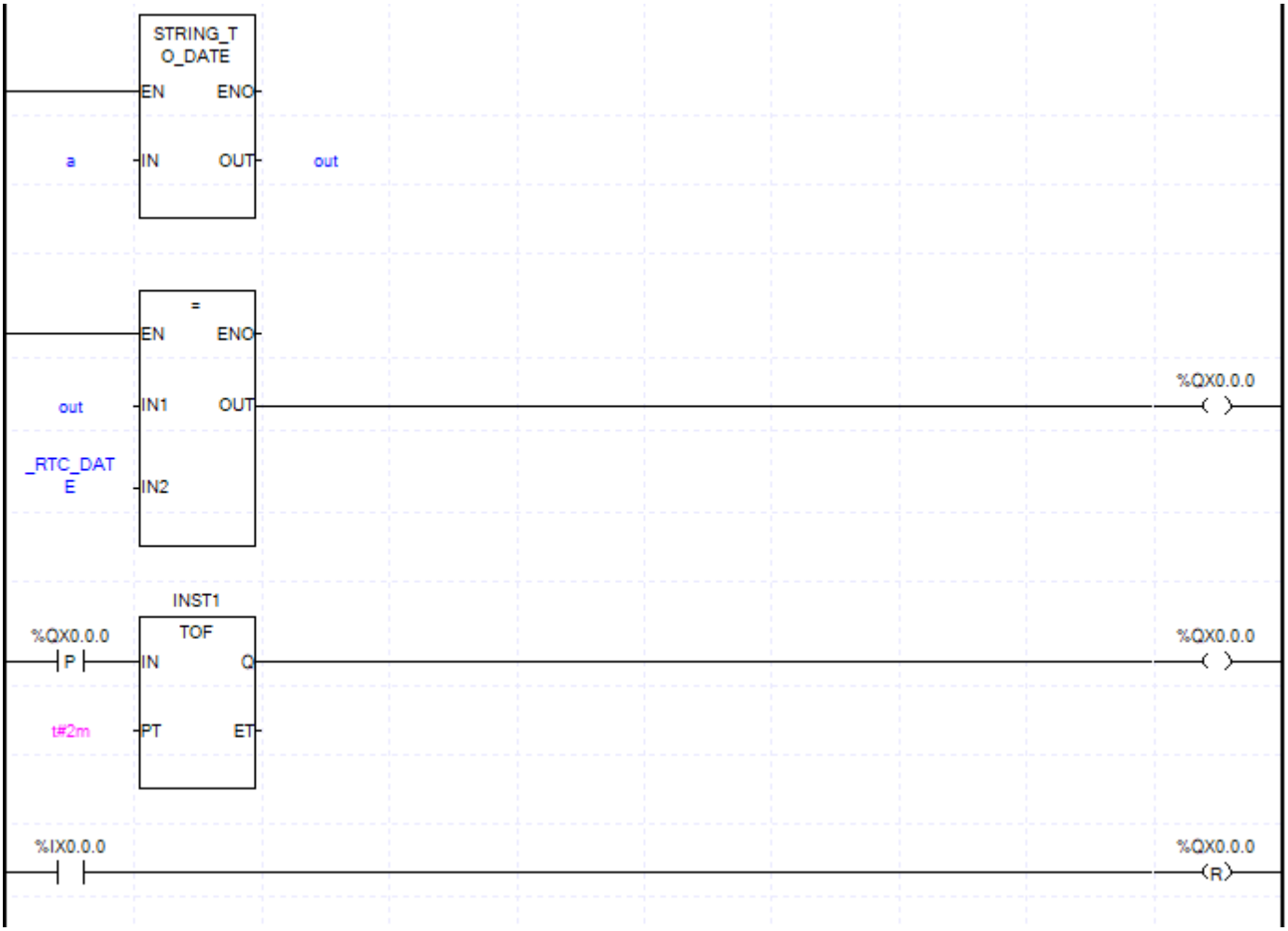


تابع تبدیل String به Date

مثال: موتور در تاریخی که توسط اپراتور تعیین می‌شود روشن شده و پس از ۳دقیقه خاموش شود.

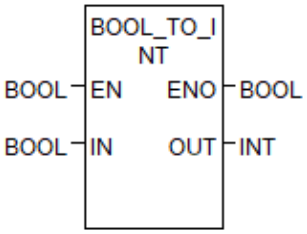
چون HMI با حافظه‌ها سروکار دارد و عددی که از HMI به PLC داده می‌شود Integer است پس باید ابتدا integer را به String تبدیل کرده و سپس String را به Date تبدیل می‌کنیم. اپراتور ۳عدد را در حافظه‌های mw0, mw1, mw2 قرار می‌دهد. دقت نمایید در تابع Concat حتما بعضی ورودی‌ها را با ’ ‘ وارد نمایید.





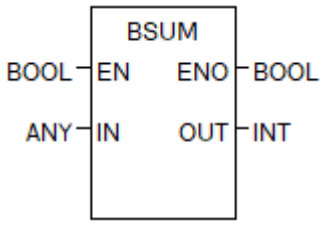
تابع تبدیل BOOL-TO-INT

یکی از کاربردهای این تابع تبدیل، جهت تشخیص فشرده‌شدن تعداد شستی‌های متصل به ورودی‌ها و یا سنسورهای متصل به ورودی‌ها است. مثلا در شمارش رای‌ها(تعداد کسانی که رای مثبت داده‌اند). که البته بعد از این تابع یک جمع‌کننده قرار می‌دهند که تعداد یک‌ها را شمارش کند.



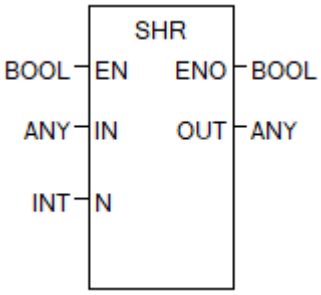
تابع BSUM

این تابع تعداد یک‌های ورودی را شمارش می‌کند.

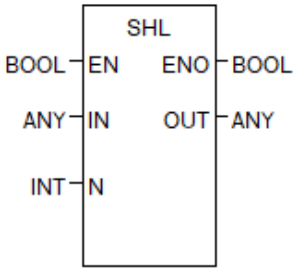


تابع Shift

در بعضی پروژه‌های صنعتی گاهی لازم است که اطلاعات درون یک بایت و یا یک Wordو... را به اندازه یک یا چند بیت از سمت چپ به راست و یا از سمت راست به چپ جابجا (شیفت) شود. تابع شیفت به سمت راست به شکل زیر است. داده ورودی باید از نوع ANY-BIT باشد در پایه N تعداد بیت‌هایی که باید شیفت پیدا کنند را وارد نمایید.



تابع شیفت به سمت چپ به شکل زیر است.

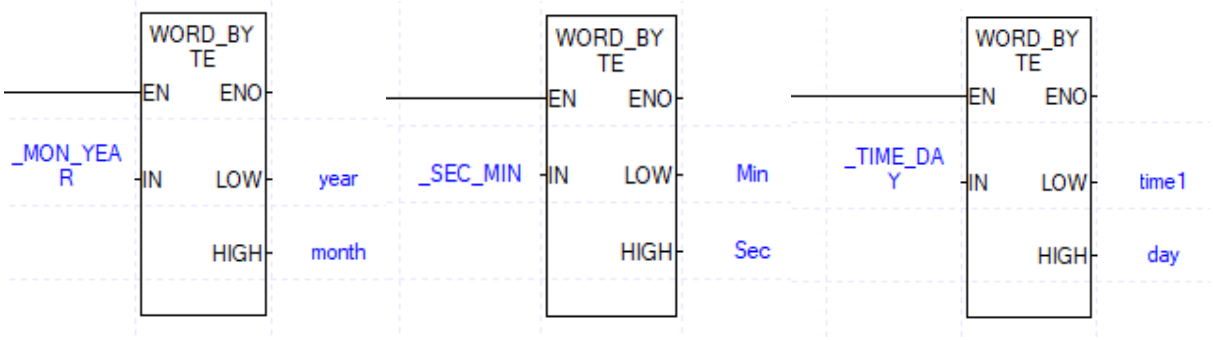


مثال: در یک مسابقه از تماشاگران سوالی پرسیده شده و جواب آنها از طریق یک شستی مشخص می‌شود. میخواهیم تعداد کسانی که رای مثبت داده‌اند را بشماریم.

از طریق IW0.0.0 وضعیت ۱۶ ورودی اول مشص می‌شود. تعداد یک‌ها یا رای‌های مثبت در B نمایش داده می‌شود.

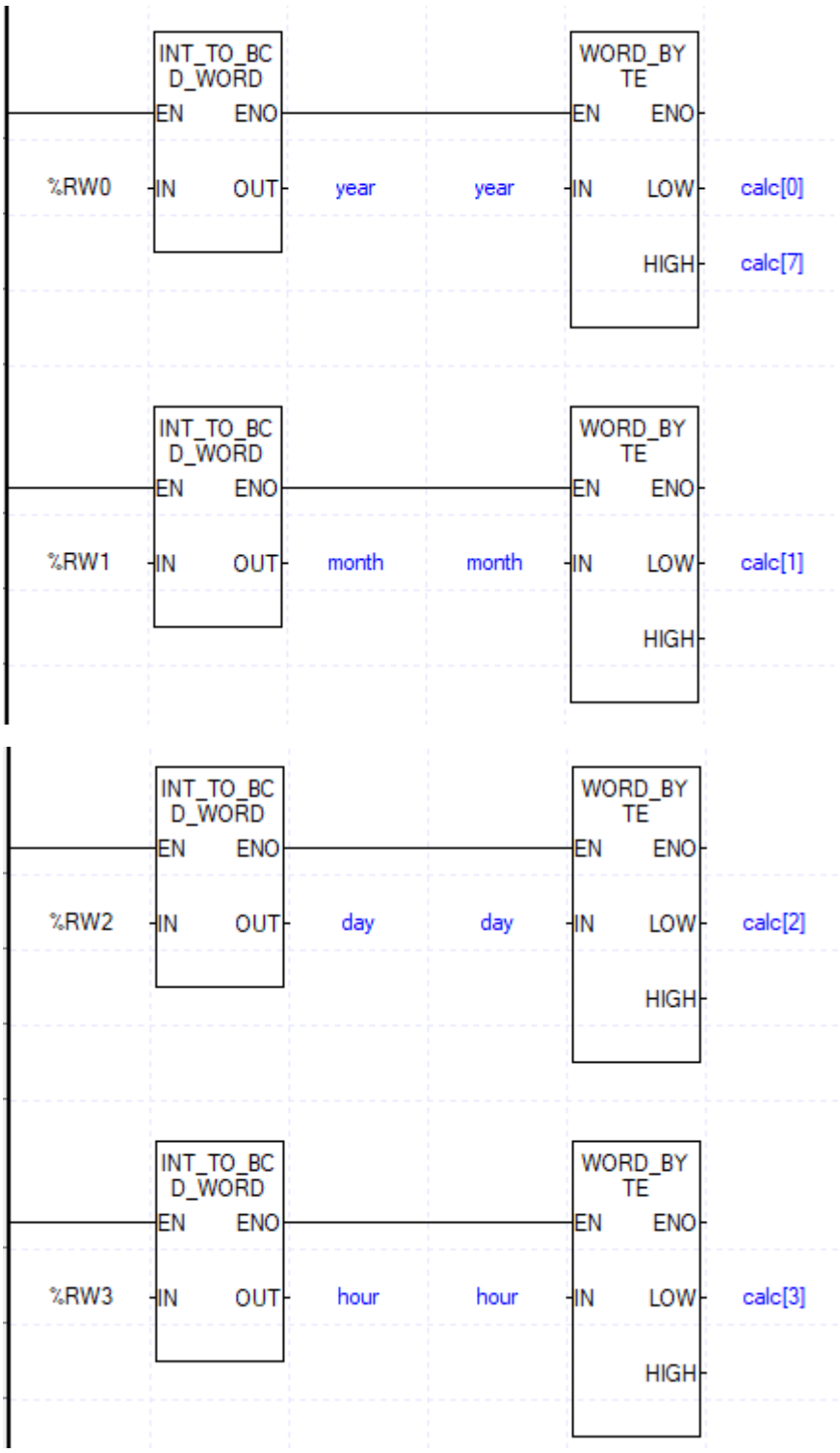
تابع Word-Byte

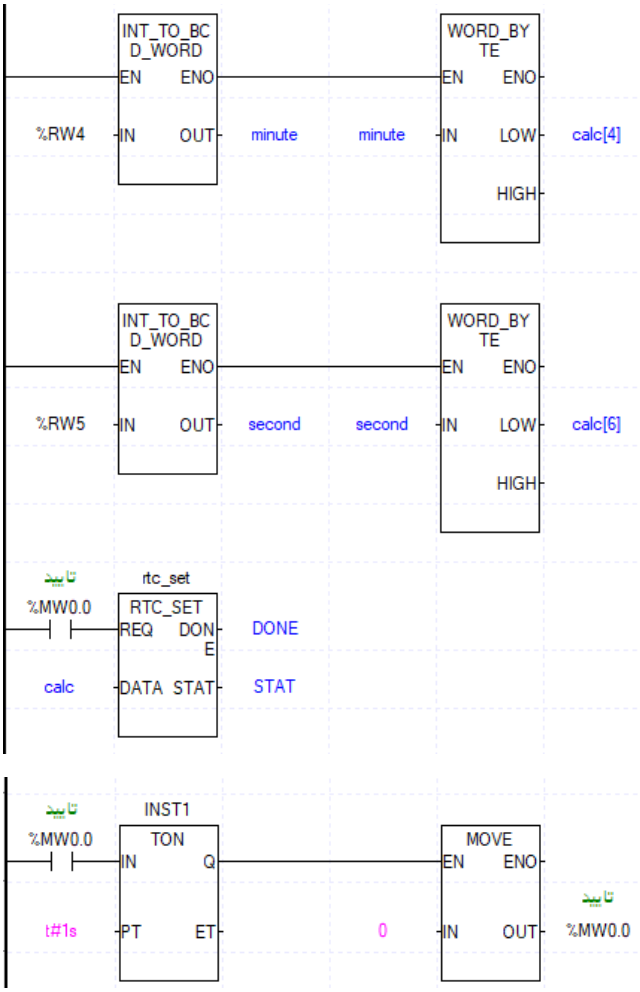
از طریق فلگ‌های زیر و تابع Word -Byte می‌توان بعضی متغیرها را جدا کرد مثلا



تنظیم ساعت و تاریخ PLC از طریق HMI

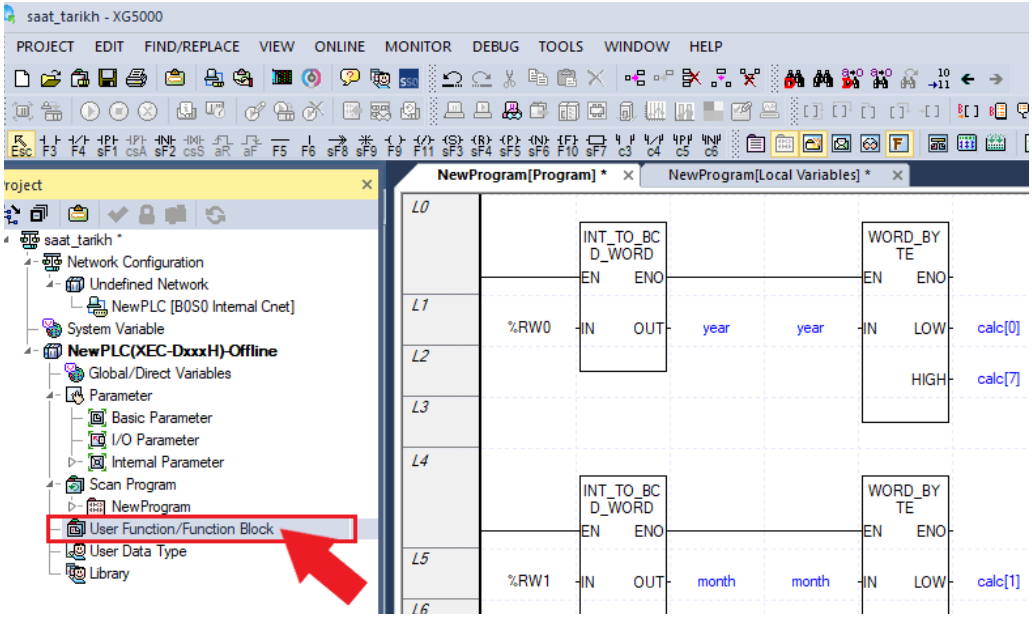
مثال: برنامه بنویسید که بتوان از طریق HMI ساعت و تاریخ plc را تنظیم کرده و با انتخاب تایید تاریخ و ساعت تنظیم شود؟ درRWها تاریخ و زمان از HMI به PLCها منتقل می‌شود.



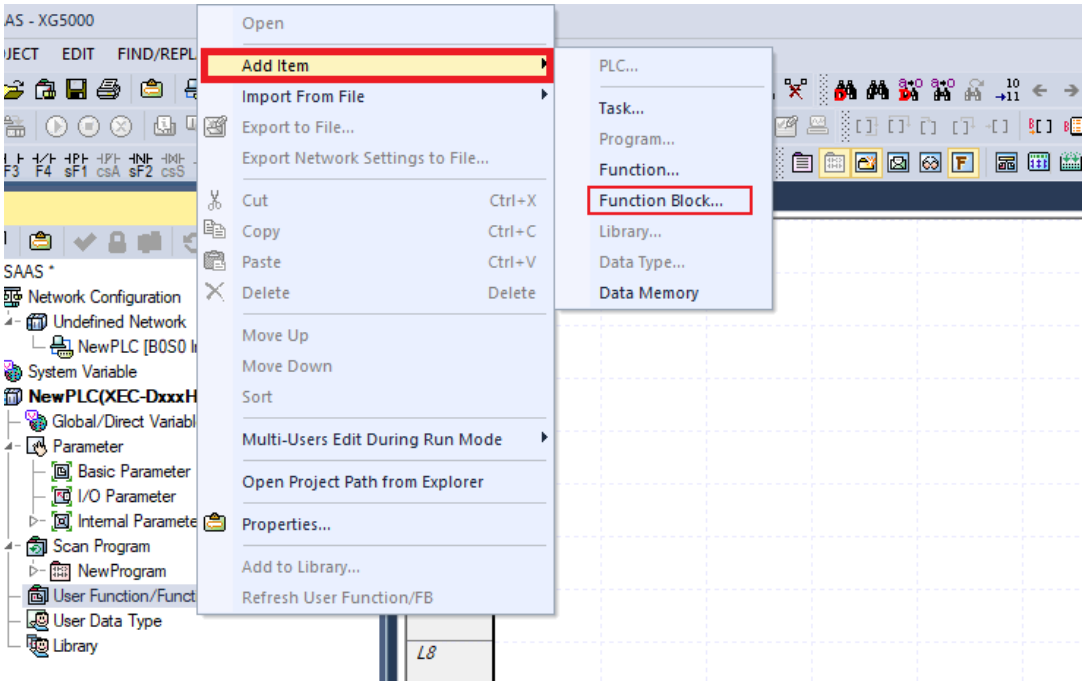


ساخت فانکشن بلاک

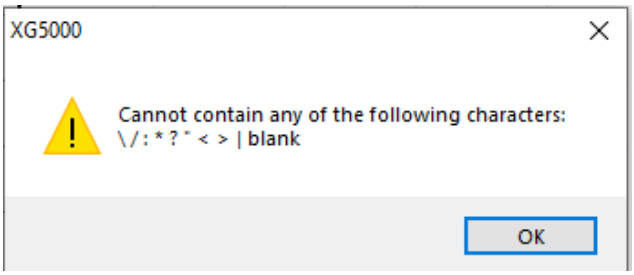
برای ساخت فانکشن بلاک ارتباط کامپیوتر با PLC نباید برقرار باشد. سپس در سمت چپ صفحه برروی User Function/Function block کلیک نمایید.

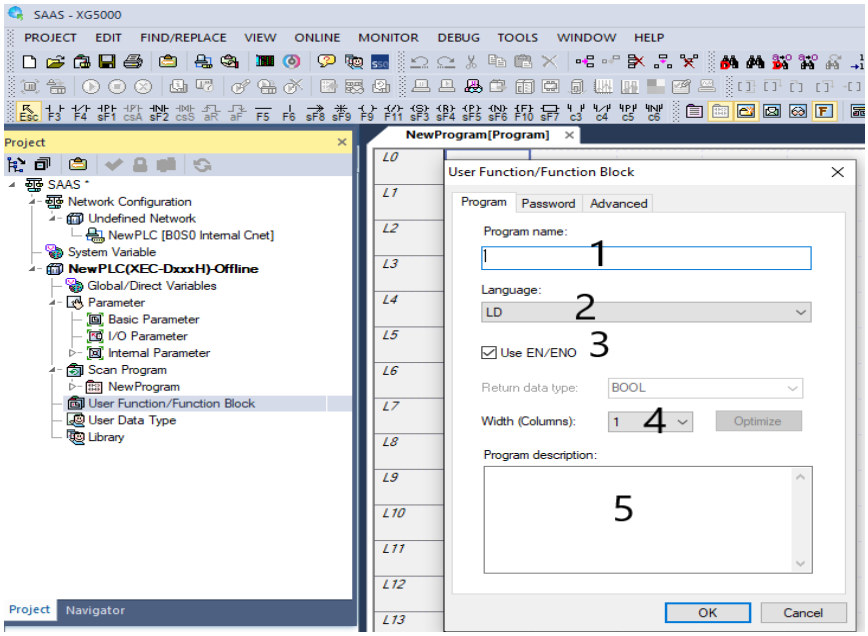


سپس پنجره زیر نمایش داده می‌شود. گزینه Add item و سپس Function Block را انتخاب نمایید.

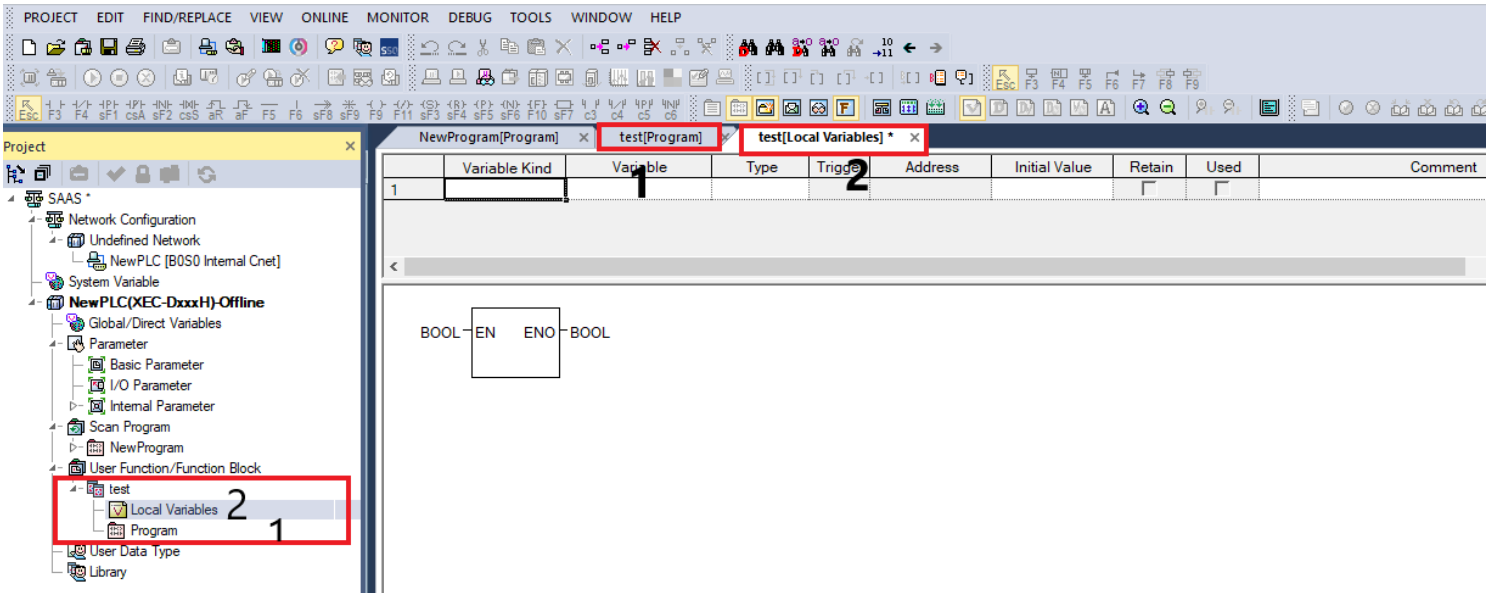


سپس پنجره زیر نمایش داده می‌شود. در قسمت یک، نام Function Block را انتخاب نمایید. برای نام دقت نمایید از خط تیره (-) و موارد زیر نمی‌توان استفاده نمود ولی از خط زیر (_) می‌توان استفاده نمود.

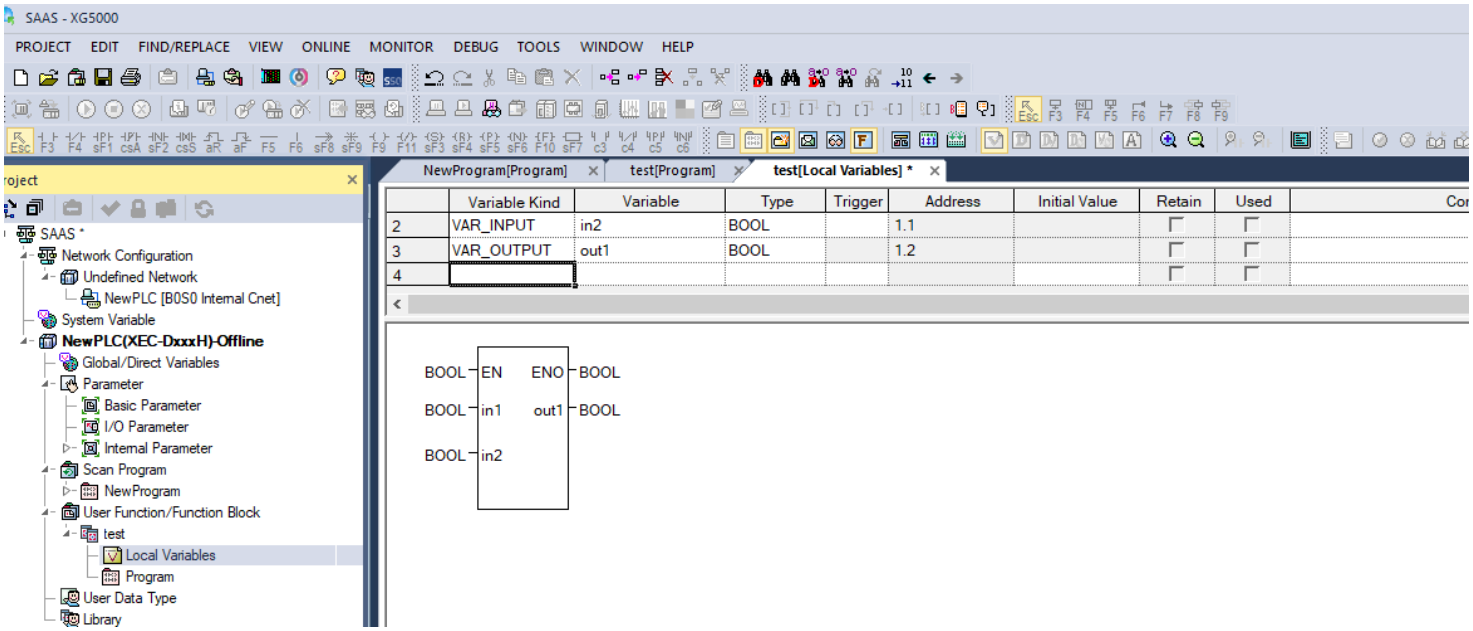




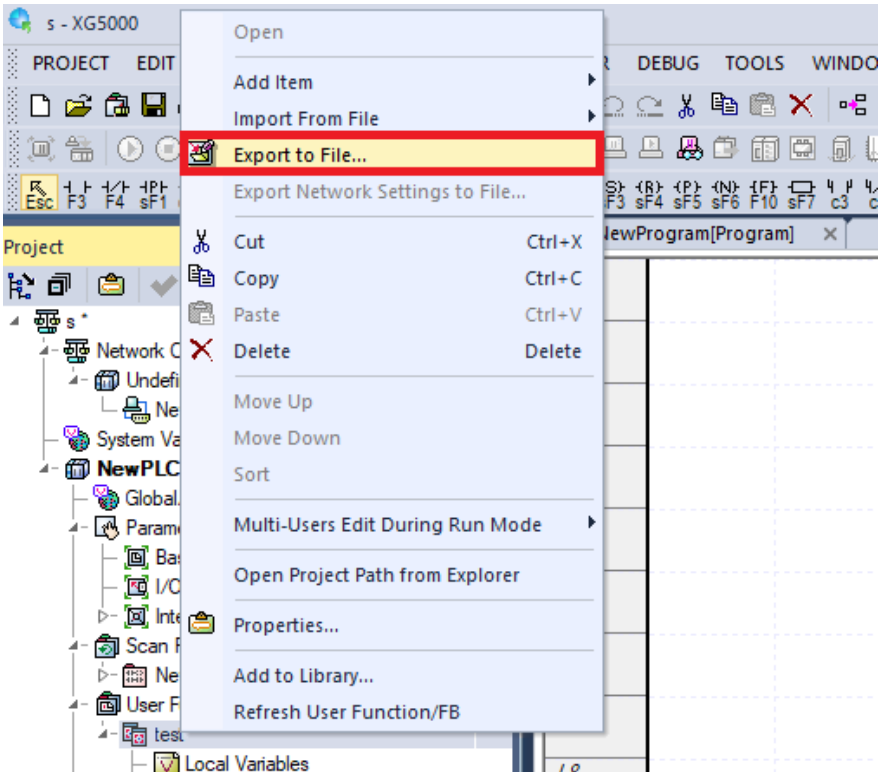
درقسمت۱ نام پروژه را انتخاب کنید. قسمت۲ زبان برنامه‌نویسی Function block را تعیین نمایید. قسمت ۳ قرار دادن پایه‌های EN/ENO تعیین می‌کند که در Function Block باشد یا خیر. در قسمت ۴ عرض بلوک را تعیین می‌کند و در قسمت ۵ اگر توضیحی برای بلوک نیاز باشد نوشته می‌شود. در سربرگ پسورد می‌توانید برای Function Block رمز اختصاص دهید تا دیگران به برنامه درون آن دسترسی نداشته باشند. با زدن کلید OK بلوک ساخته شده و در سمت چپ صفحه نمایش داده می‌شود.



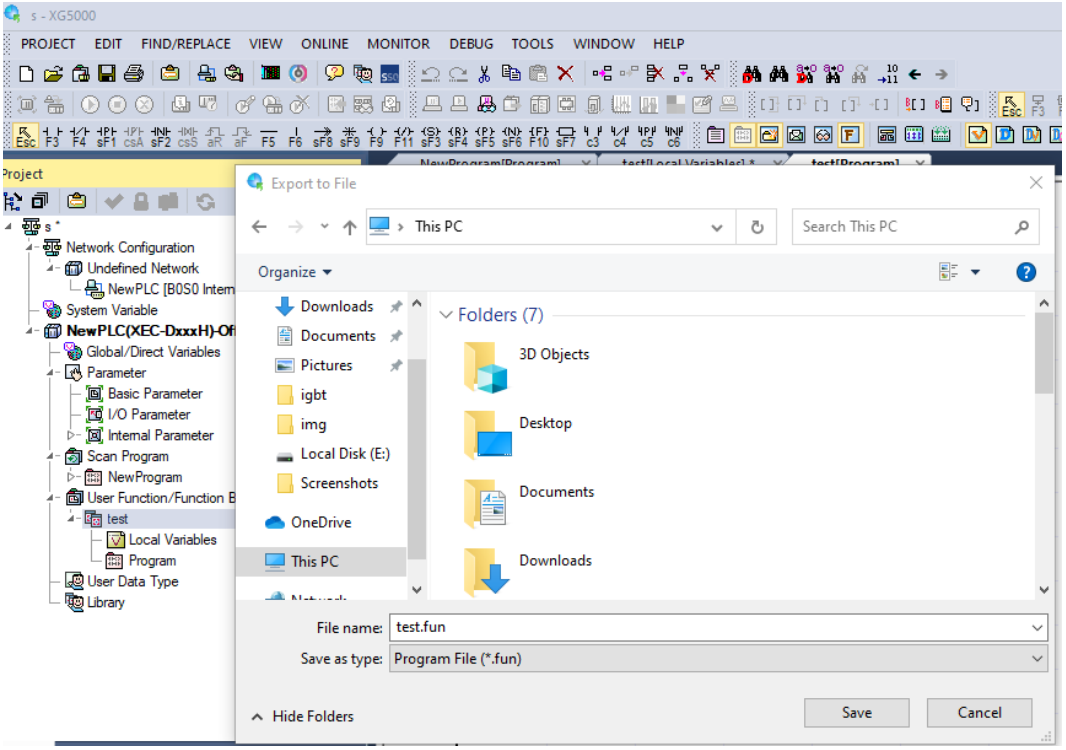
در قسمت Program برنامه مربوط به تابع را بنویسید. در قسمت Local Variable متغیرهای مورد نیاز برای Function Block را وارد نمایید. پس از وارد نمودن متغیرها شکل تابع بصورت زیر می‌شود.



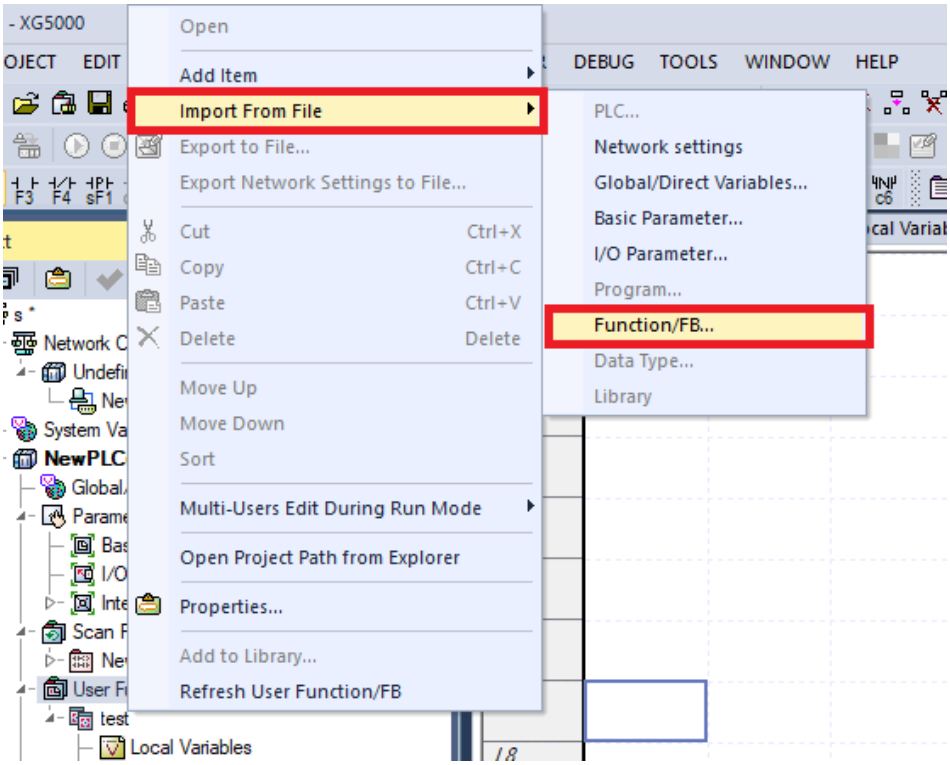
برای اینکه از این تابع در برنامه دیگر استفاده نمایید باید این تابع را Export نمایید و در برنامه دیگر آن را Import نمایید. با کلیک راست بر روی تابع ساخته شده Export را برای استفاده از این تابع در برنامه دیگر انتخاب نمایید



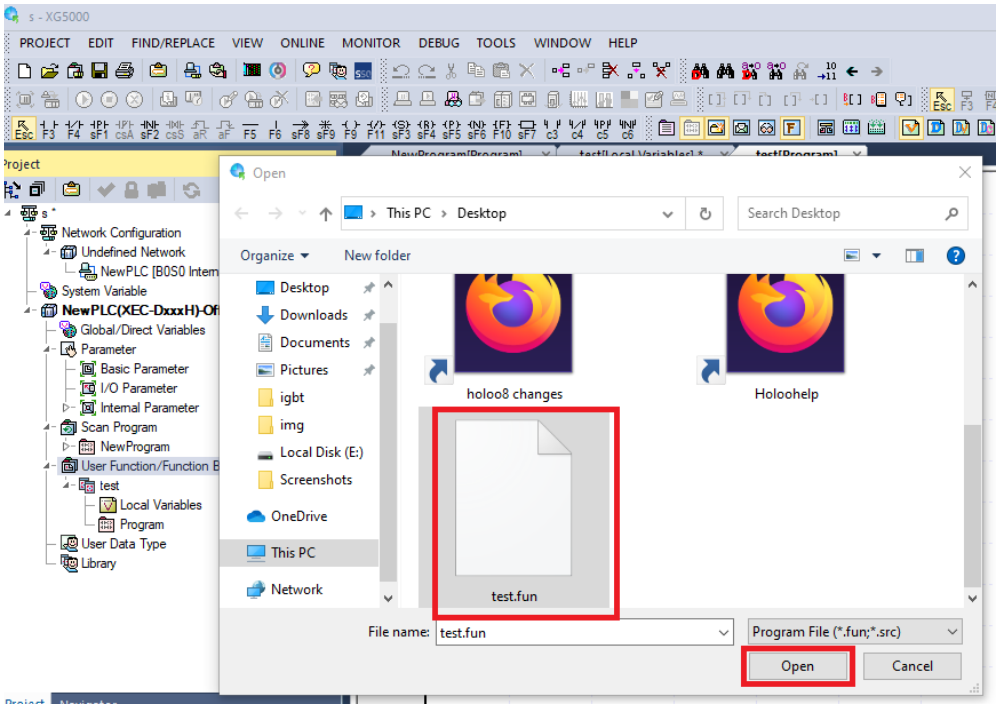
سپس در پنجره باز شده محل ذخیره تابع را تعیین نمایید.



برای استفاده از تابعی که قبلا ساخته شده است و در برنامه ما وجود ندارد می‌توان درون برنامه از گزینه Import استفاده نمود تا تابع را وارد برنامه نموده و از آن استفاده نمود. برای اینکار بر روی User Function/function Block کلیک راست نموده و سپس Import و سپس...Function/FB را انتخاب نمایید.



سپس از پنجره باز شده تابع را انتخاب نمایید.



تفاوت Function با Function Block

در Function بعد هربار اسکن تمام متغیرهای Function صفر می‌شود درصورتی که در Function Block اینگونه نیست. Function نیازی به نامگذاری ندارد ولی Function Block نیاز به نامگذاری دارد. Function Block فضای بیشتری را اشغال می‌کند و برای اجرا زمان بیشتری نیاز است. Function Block حافظه‌دار بوده درصورتی که Functionحافظه ندارد.

تبدیل کد گری به دسیمال

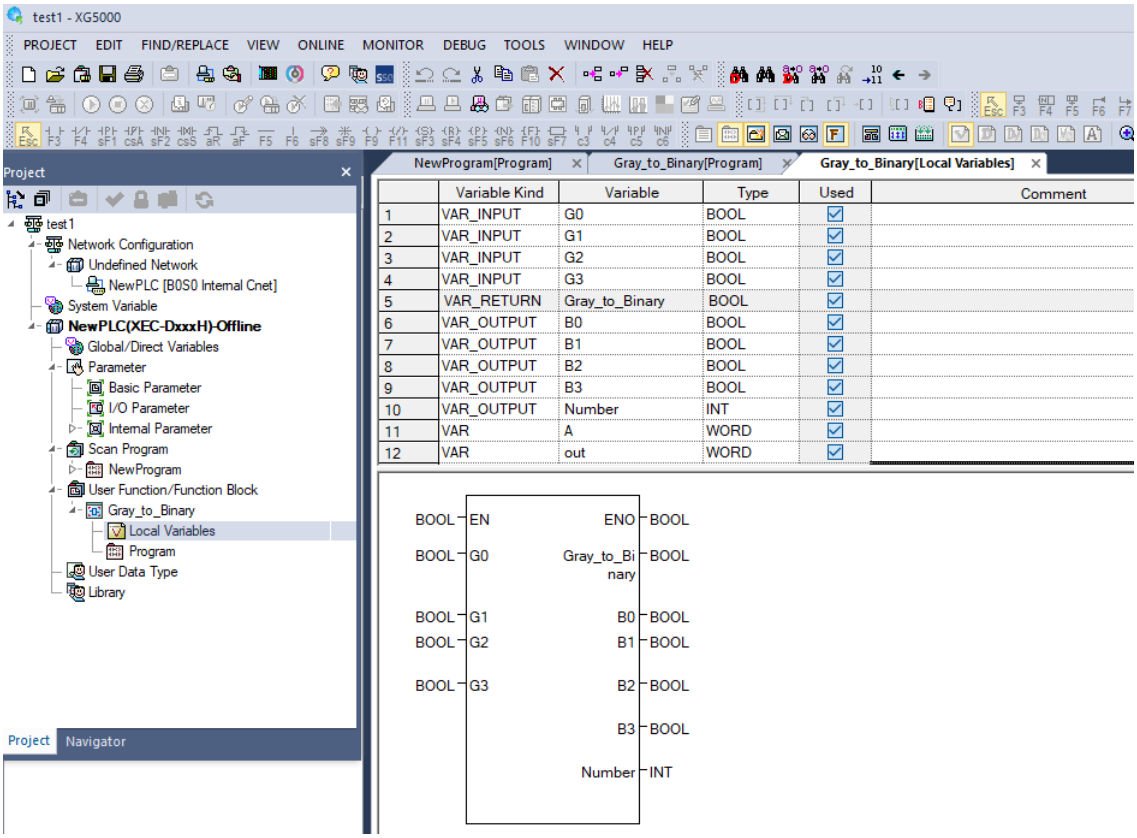
مثال: یک تابع بسازید که ورودی کد Gray گرفته و خروجی عدد بدهد؟ یکی از کاربردهای این مثال در انکودرهای مطلق که خروجی کد Gray دارند می‌باشد تا بتوانیم مقدار انکدر را بخوانیم. ابتدا کد Gray را به باینری تبدیل و سپس آن را به عدد تبدیل می‌کنیم. کد Gray از XOR بیت‌های کد باینری تولید می‌شود.

کد باینری				کد Gray			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

$G0=B0\oplus B1,\ G1=B1\oplus B2,\ G2=B2\oplus B3,\ G3=B3$

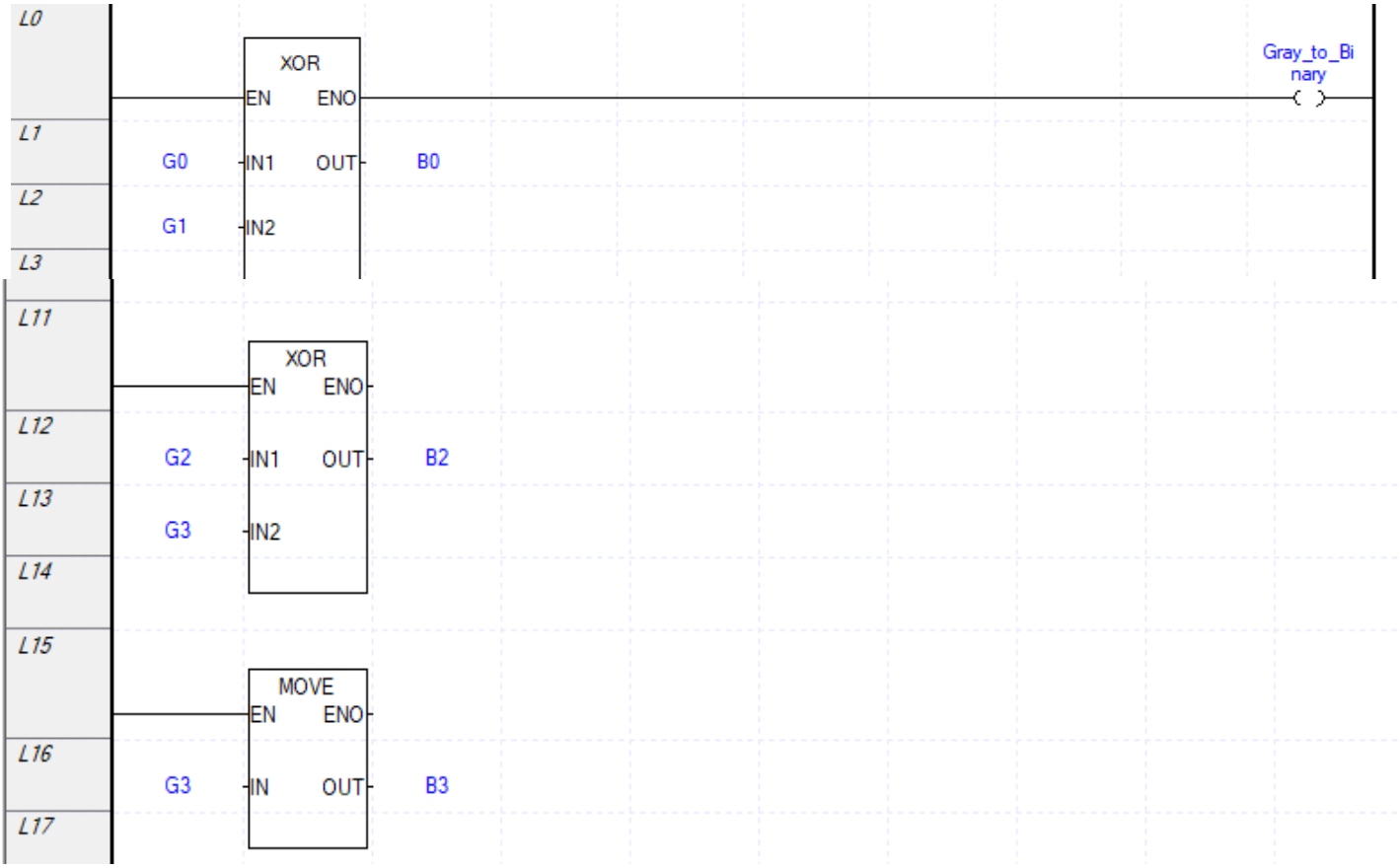
$B0=G3\oplus G2\oplus G1\oplus G0,\ B1=G3\oplus G2\oplus G1,\ B2=G3\oplus G2,\ B3=G3$

ابتدا یک Function می‌سازیم. ساخت Function مانند Function Block است با این تفاوت که بعد از کلیک راست و Add item گزینه Function را انتخاب می‌کنیم. برای تعریف متغیرها در تابع باید حتما یک متغیر برگشتی تعریف کنیم. چهار ورودی G0 تا G3 را بعنوان متغیر ورودی تعریف می‌کنیم. B0 تا B3 را نیز بعنوان متغیر خروجی تعریف می‌کنیم.

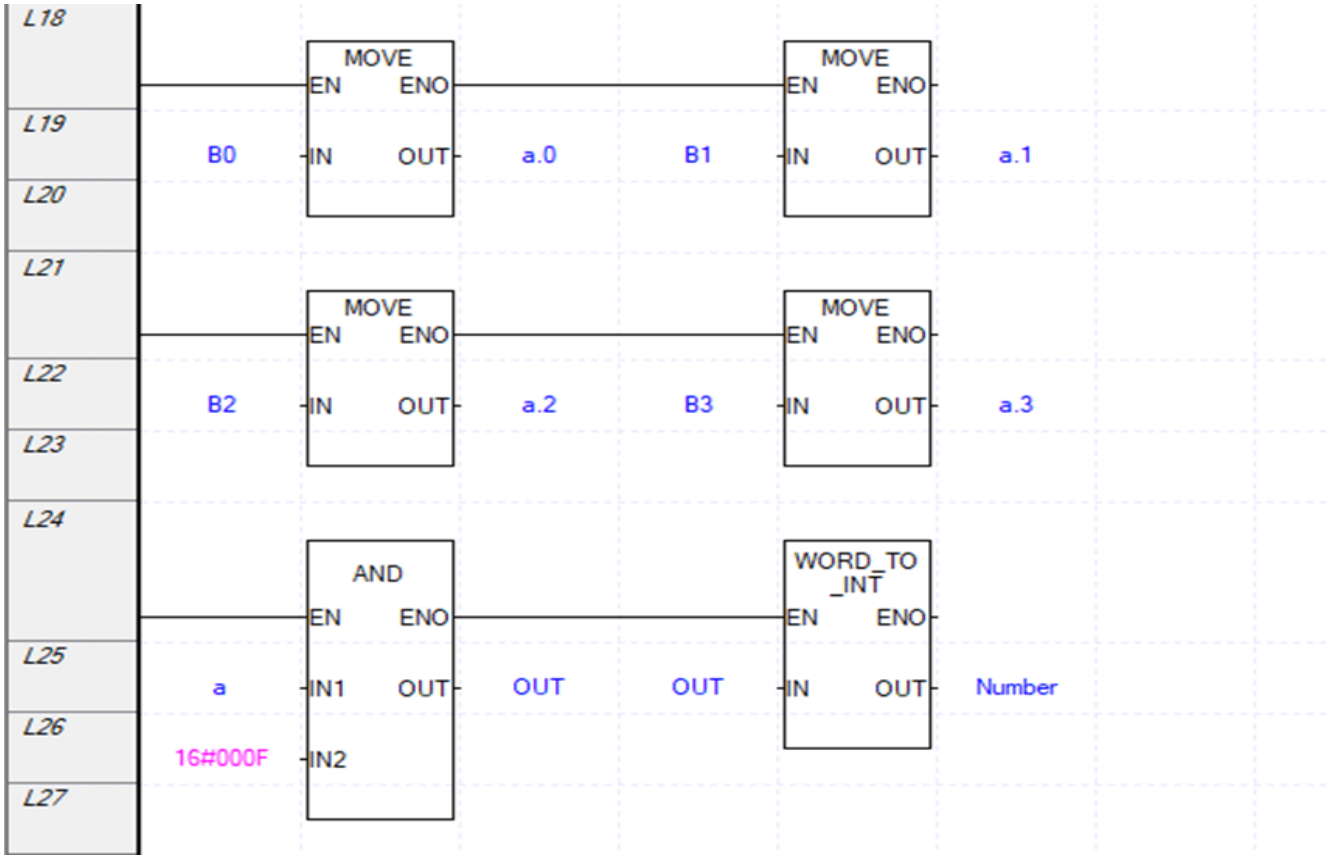


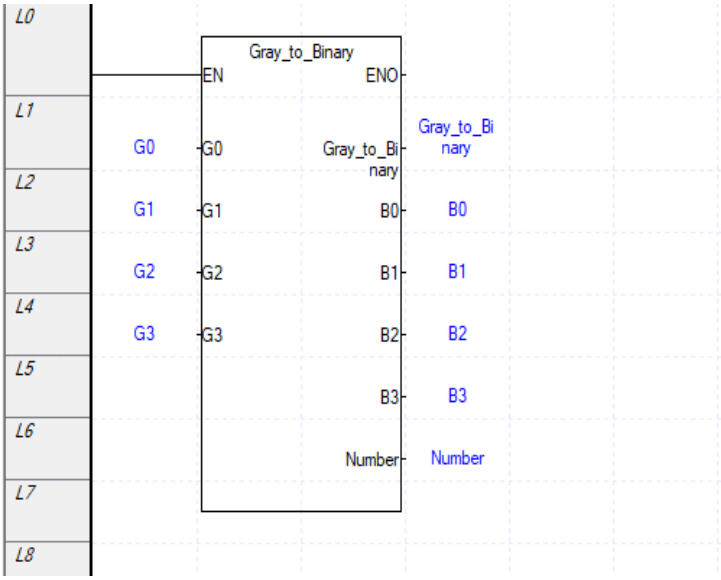
در Function بندرت از EN استفاده می‌شود.

سپس وارد محیط برنامه‌نویسی شده و برنامه را بنویسید.

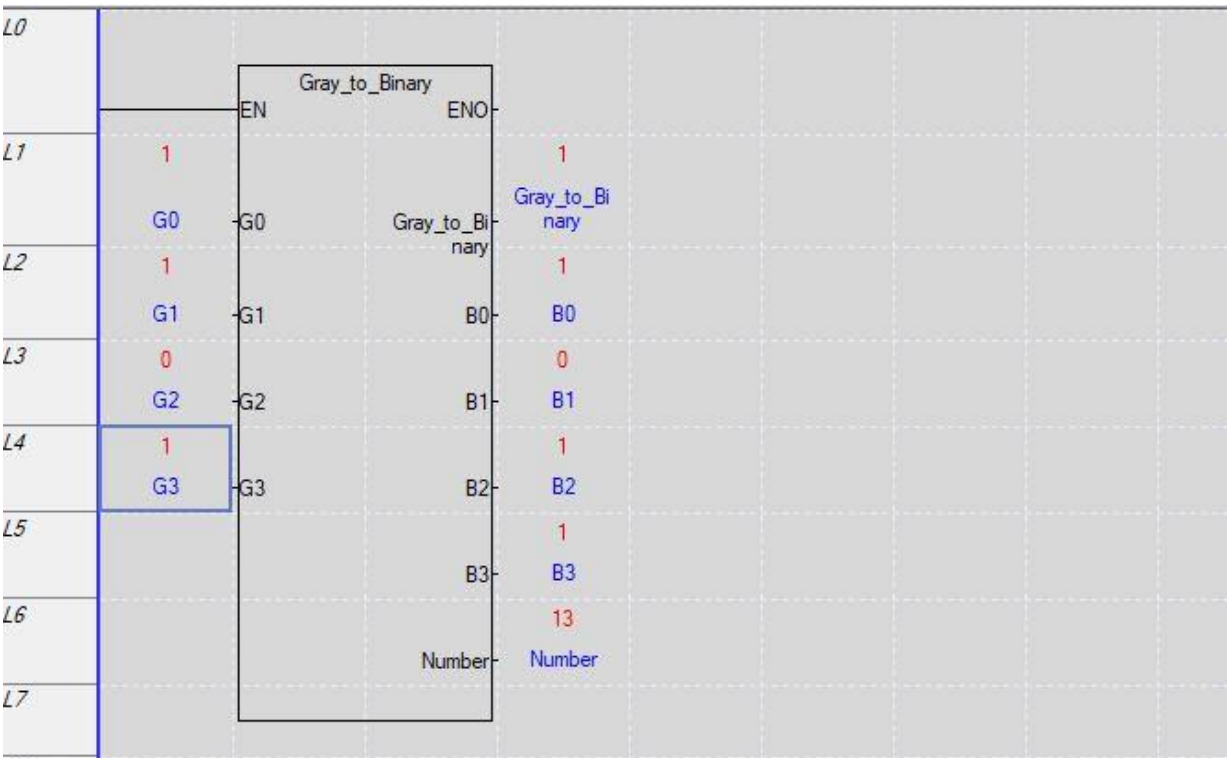


سپس اعداد باینری تولیدشده را در یک Word مرتب می‌کنیم و چون بجز ۴ بیت اول بقیه بیت‌ها خالی هستند باید آنهارا صفر نموده تا خطا بوجود نیاید. سپس Word را به INT تبدیل می‌کنیم. سپس Save را انتخاب و ذخیره کنید و برنامه را ببندید. سپس در برنامه اصلی تابع را فراخوانی کنید. اکنون به آن متغیر اختصاص داده و برنامه را Run کنید. در قسمت Number عدد دسیمال یا INT را به شما نمایش می‌دهد.

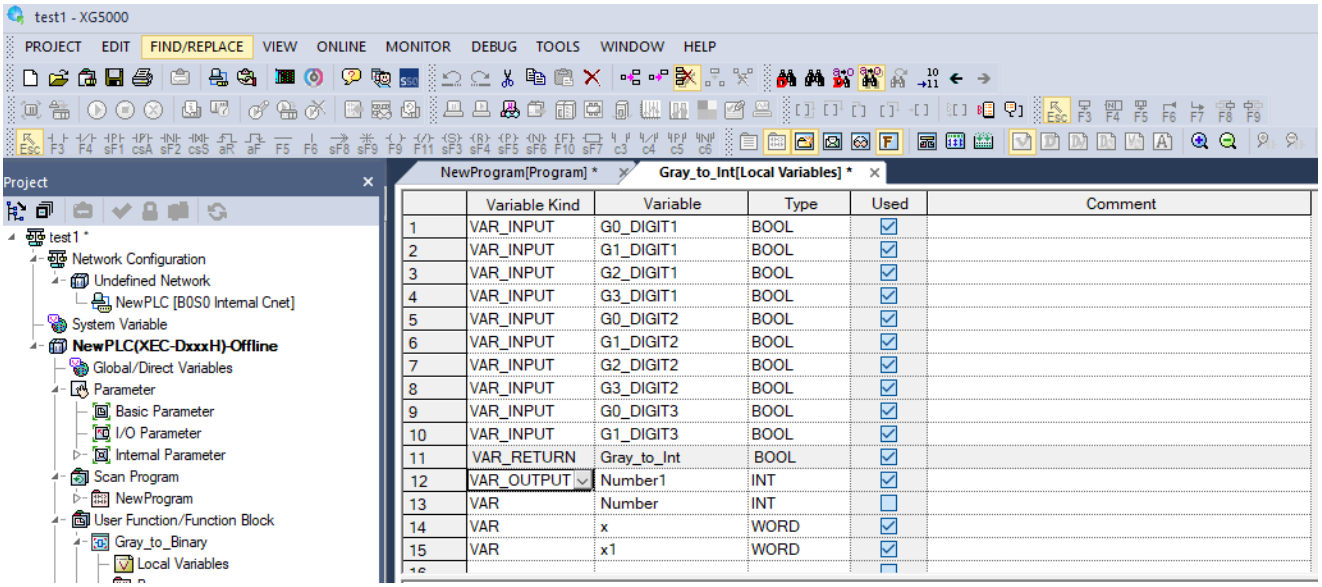




برنامه پس از Run شدن بصورت زیر است.

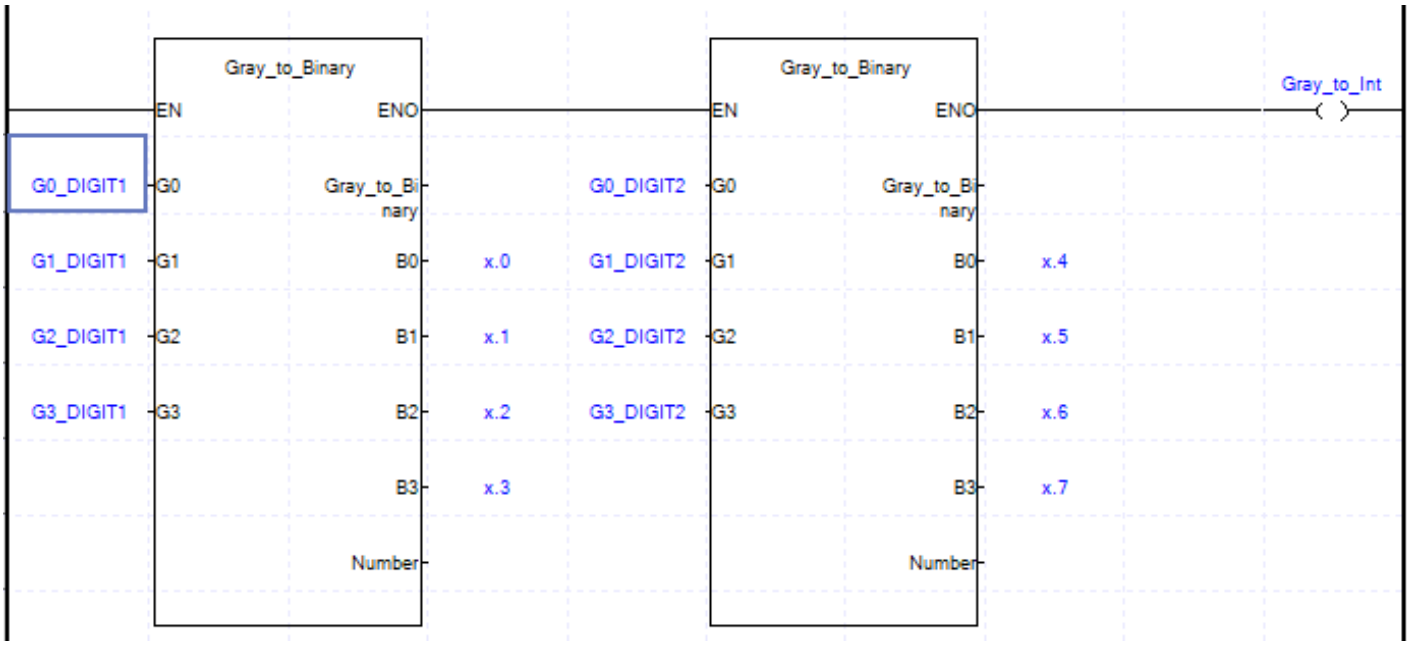


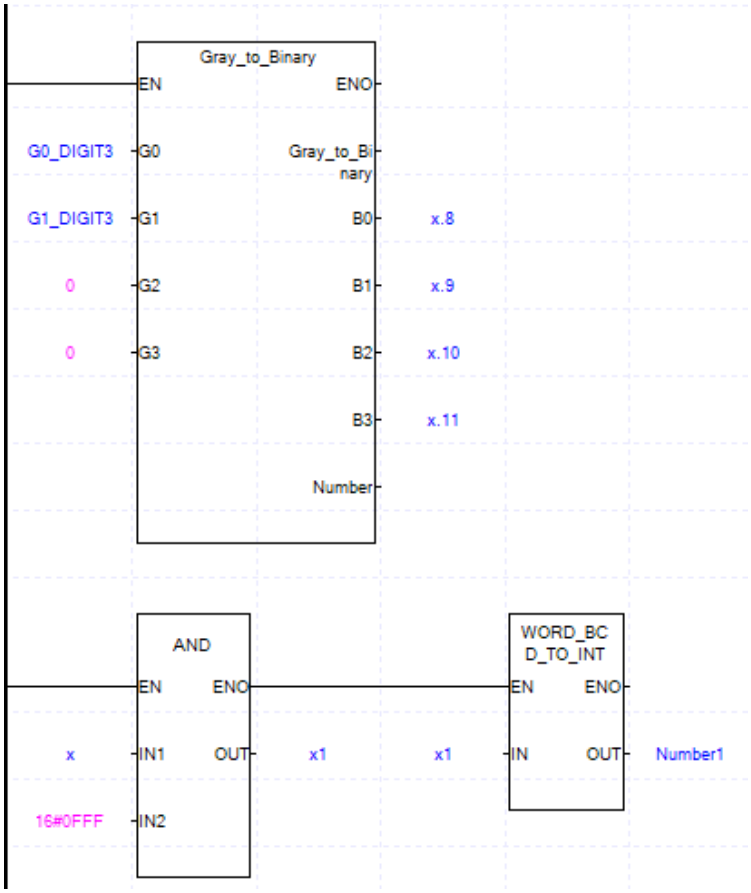
تا اینجا برای یک کد Gray برنامه نوشته شد اما برای انکدر برنامه کاملتر خواهد بود و به شکل زیر است.



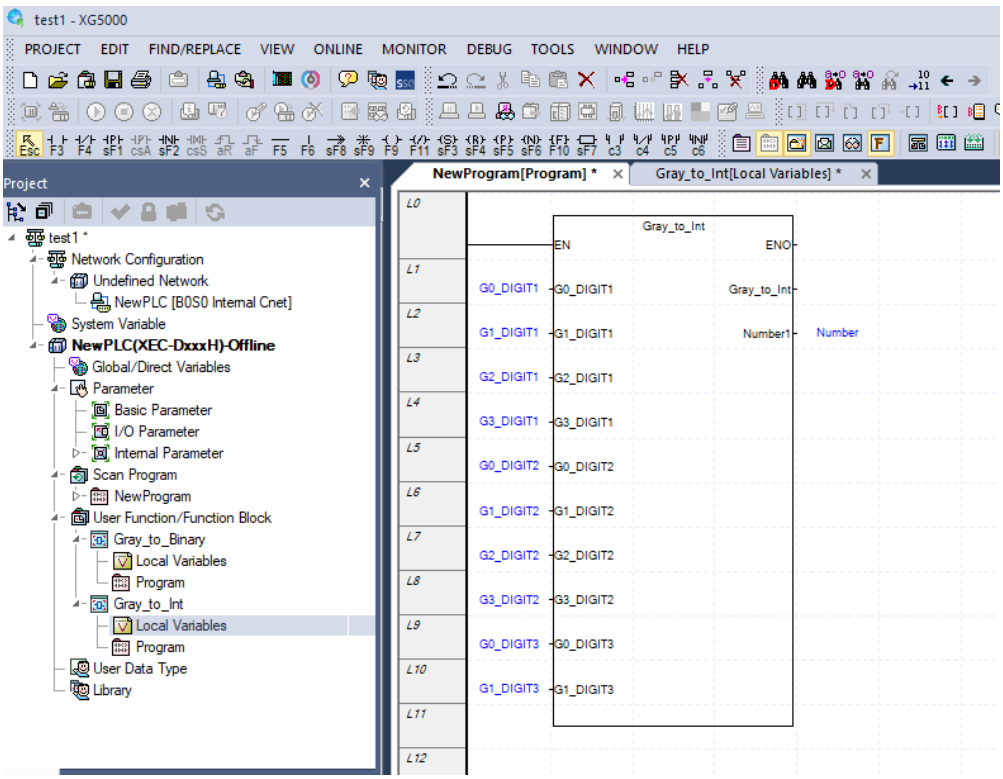
	Variable Kind	Variable	Type	Used	Comment
1	VAR_INPUT	G0_DIGIT1	BOOL	☑	
2	VAR_INPUT	G1_DIGIT1	BOOL	☑	
3	VAR_INPUT	G2_DIGIT1	BOOL	☑	
4	VAR_INPUT	G3_DIGIT1	BOOL	☑	
5	VAR_INPUT	G0_DIGIT2	BOOL	☑	
6	VAR_INPUT	G1_DIGIT2	BOOL	☑	
7	VAR_INPUT	G2_DIGIT2	BOOL	☑	
8	VAR_INPUT	G3_DIGIT2	BOOL	☑	
9	VAR_INPUT	G0_DIGIT3	BOOL	☑	
10	VAR_INPUT	G1_DIGIT3	BOOL	☑	
11	VAR_RETURN	Gray_to_Int	BOOL	☑	
12	VAR_OUTPUT	Number1	INT	☑	
13	VAR	Number	INT	☐	
14	VAR	x	WORD	☑	
15	VAR	x1	WORD	☑	

متغیر X1و X را بصورت Word تعریف نمایید. برنامه در محیط Gray_ to_ Int بصورت زیر است.

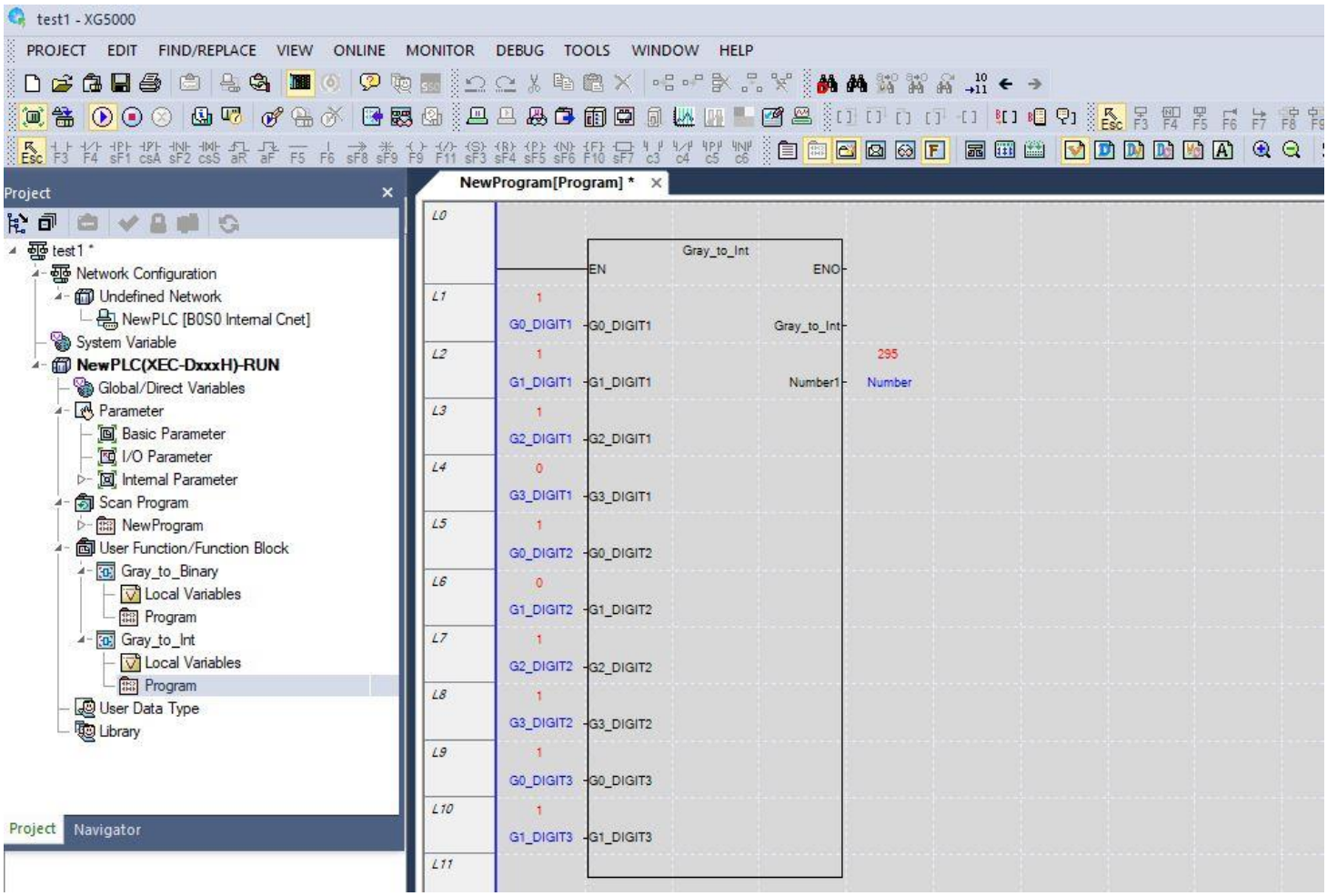




برنامه اصلی بصورت زیر می‌باشد.

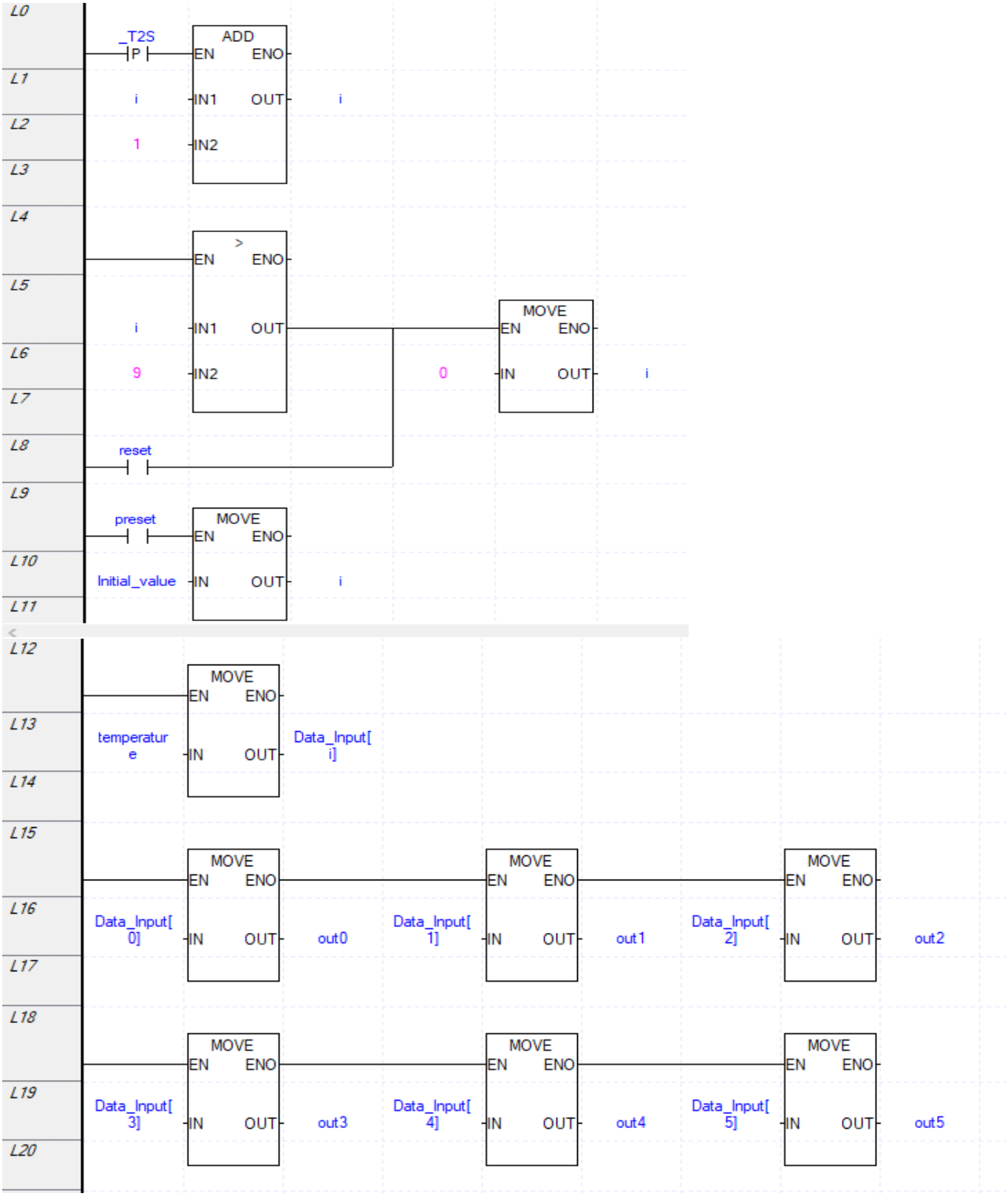


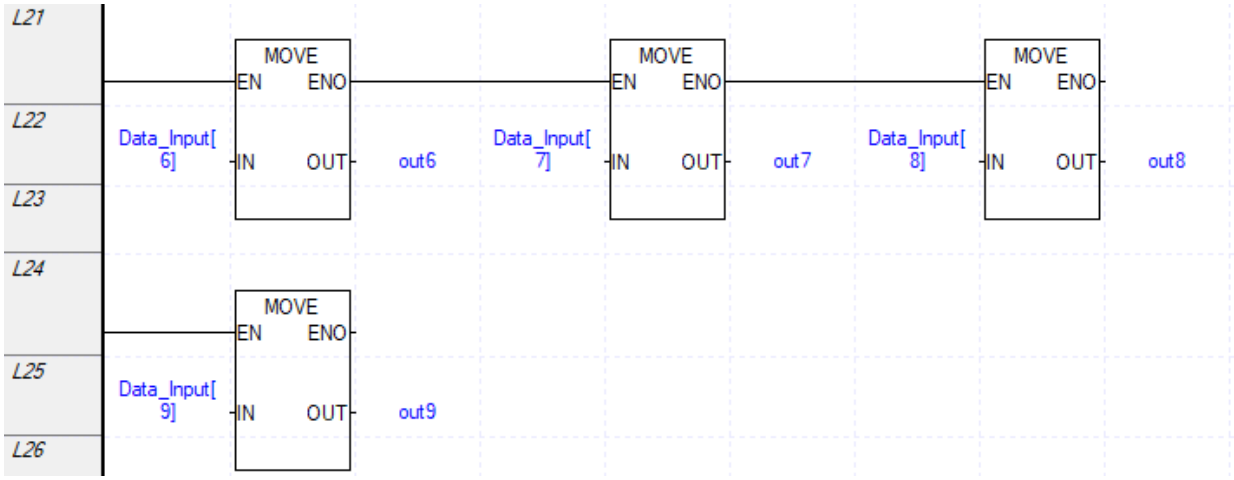
و پس از Run شدن برنامه بصورت زیر است. دقت کنید که کد Gray است و نمی‌تواند همه یک باشد.

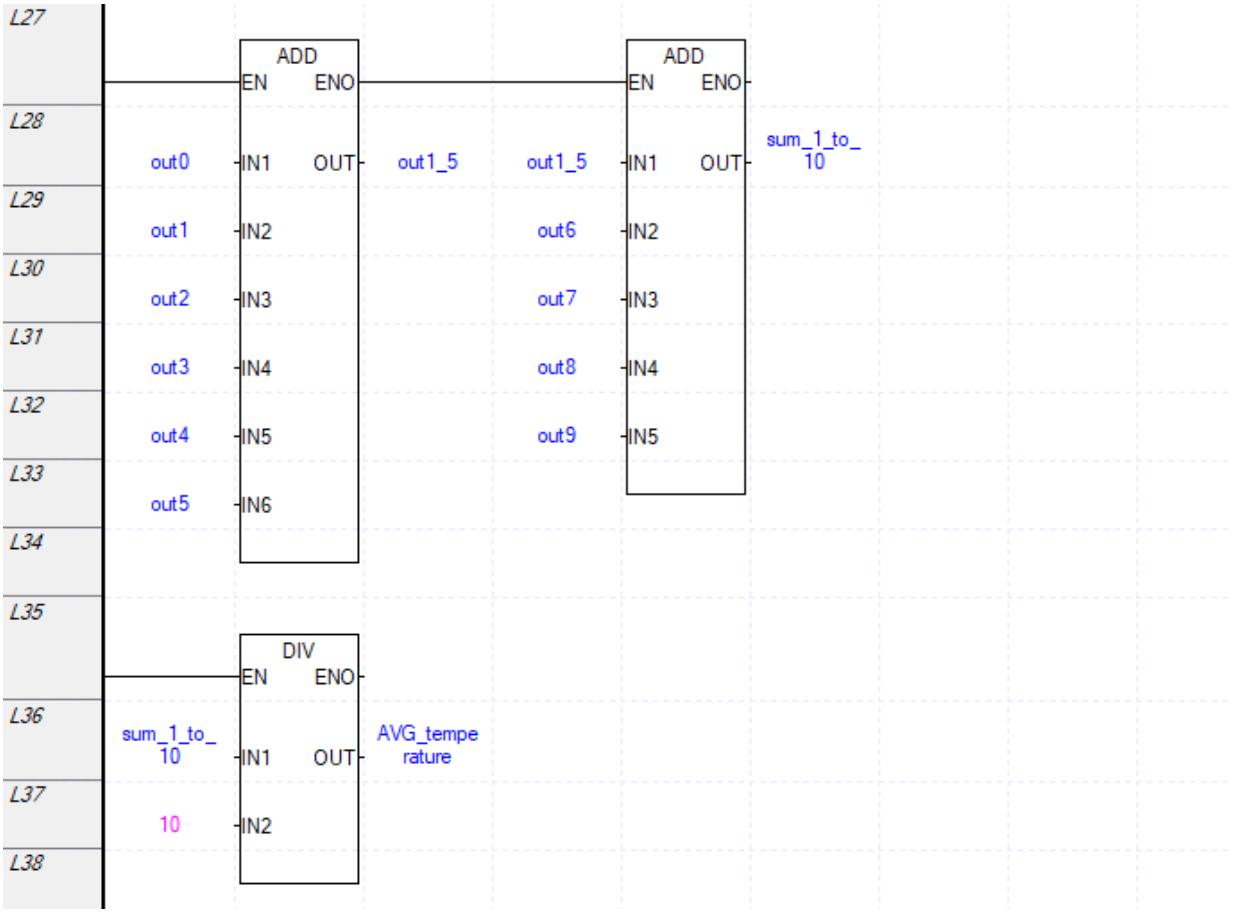


برای اتصال انکدر به PLC بجای G0-Digit1 مقدار Ix0.0.0 را قرار دهید و بقیه متغیرها را براساس پایه ورودی وصل شده تنظیم نمایید.

مثال: شمارنده‌ای طراحی کنید که از صفر تا ۹ را شمرده و سپس ریست شود یا بعبارتی میانگین ۱۰مقدار اخیر را محاسبه کند. برای اندازه‌گیری تعیین می‌کنیم که هر چندثانیه شمرده شود. Data_ Input[I] از نوع آرایه شامل ۰تا۹ است. برنامه آن بصورت زیر است.




L21
L22
L23
L24
L25
L26
MOVE
EN
ENO
IN
OUT
Data_Input[6]
out6
Data_Input[7]
out7
Data_Input[8]
out8
Data_Input[9]
out9


L27
L28
L29
L30
L31
L32
L33
L34
L35
L36
L37
L38
ADD
EN
ENO
IN1
IN2
IN3
IN4
IN5
IN6
OUT
out0
out1
out2
out3
out4
out5
out1_5
out1_5
out6
out7
out8
out9
sum_1_to_10
DIV
sum_1_to_10
10
AVG_temperature

برای میانگین‌گیری می‌توان از جمع‌کننده(ADD) و تقسیم(DIV) استفاده نکرد و مستقیما از تابع Array-AVE استفاده نمود که در قسمت index محل شروع مثلا صفر را وارد نمایید و در قسمت length طول کل آرایه را وارد نمایید برای مثال ما ، ۱۰ را وارد نمایید.

از تابع Array-fill می‌توان تمام آرایه را مقداردهی نمود. که قسمت index نقطه شروع خانه آرایه و length طول مد نظر خانه‌های آرایه است.

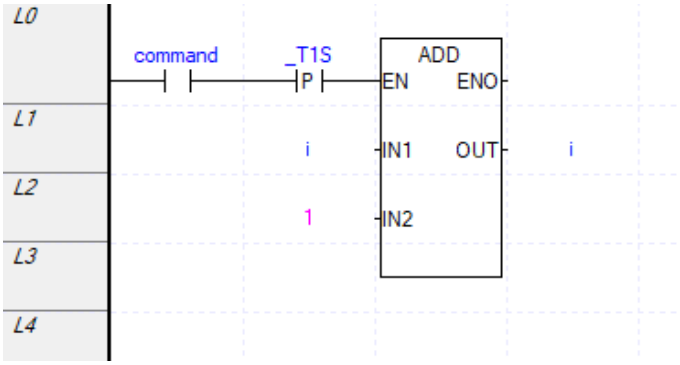
تایمرها

وظیفه تایمرها محاسبه زمان جهت انجام عملکردی خاص می‌باشد. مثلا پس از گذشت چند دقیقه موتور خاموش/ روشن شود. تایمرها عمدتا به سه دسته تقسیم می‌شوند: تایمر تاخیر در وصل(TON)، تایمر تاخیر در قطع(TOF)، تایمر ضربه یا پالس(TP).

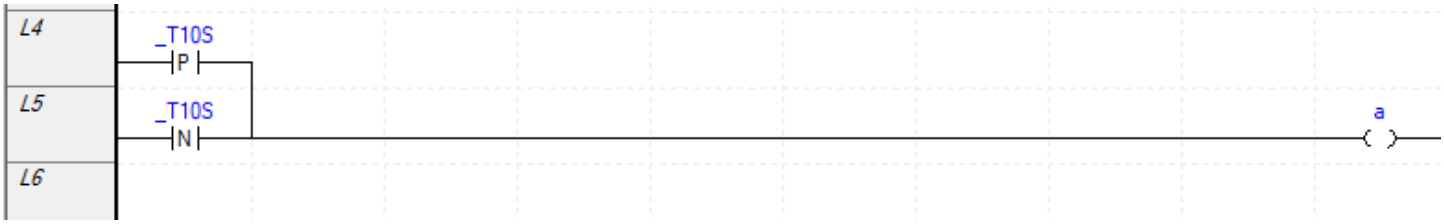
فلگ‌هایی(تیغه باز با اسم ...T_) وجود دارد که نقش تایمر را ایفا می‌کنند. این فلگ‌ها شامل:

_T20ms, _T100ms, _T200ms, _T1s, _T2s, _T10s, _T60s که زمان‌های آنها از ۲۰میلی‌ثانیه تا ۶۰ ثانیه می‌باشد.

مثال: یک تایمر با فلگ‌ها و بدون تایمرهای اصلی برنامه طراحی نمایید. می‌توان با استفاده از i و مقایسه‌گرها خروجی را در زمانی خاص خاموش یا روشن نمود مقدار i این تایمر برحسب ثانیه می‌باشد. تیغه فرمان یا command به ما امکان می‌دهد هر زمانی مایل بودیم تایمر شروع به کار نماید.

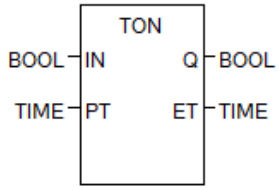


از طریق برنامه زیر می‌توان زمان این فلگ‌ها را نصف نمود برای مثال برای فلگ ۱۰ میلی‌ثانیه

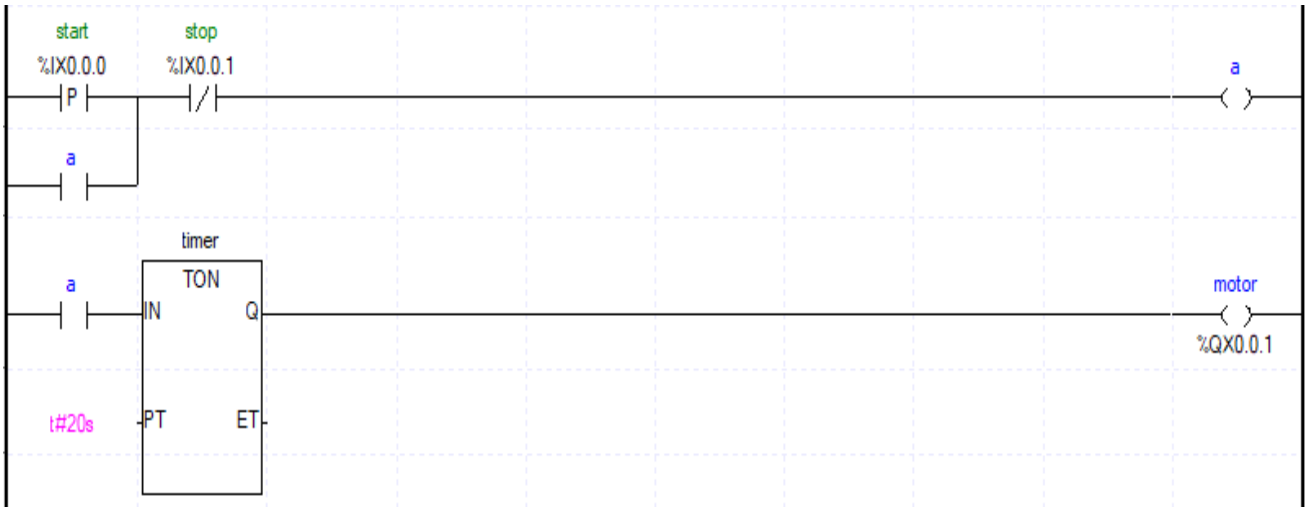


تایمر تاخیر در وصل(TON) :

TON مخفف on delay timer می‌باشد. برای اینکه زمان‌شماری به درستی انجام شود باید به پایه ورودی (IN) تایمر یک منطقی اعمال نمود و اگر در حین زمان‌شماری پایه ورودی غیرفعال شود زمان‌شماری متوقف و با فعال شدن مجدد پایه، زمان‌شماری از ابتدا آغاز می‌گردد. در صورتی که این پایه غیرفعال شود خروجی نیز حتی پس از سپری شدن زمان مدنظر غیرفعال می‌گردد. زمان مورد نظر در پایه PT یا preset time تنظیم می‌گردد. فرمت وارد کردن زمان بصورت T#number می‌باشد. حتما بعد از عدد زمان(ثانیه یا دقیقه یا ساعت) را وارد نمایید. مثلا T#25s, T#21m25s, T#2h10m24s . پایه ET یا Elapsed time زمان سپری شده را نمایش می‌دهد.

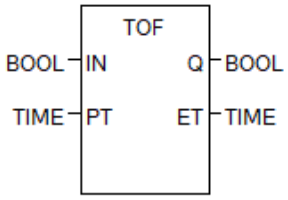


مثال: با زدن استارت موتور پس از ۲۰ثانیه روشن شود و با زدن استپ موتور متوقف گردد.

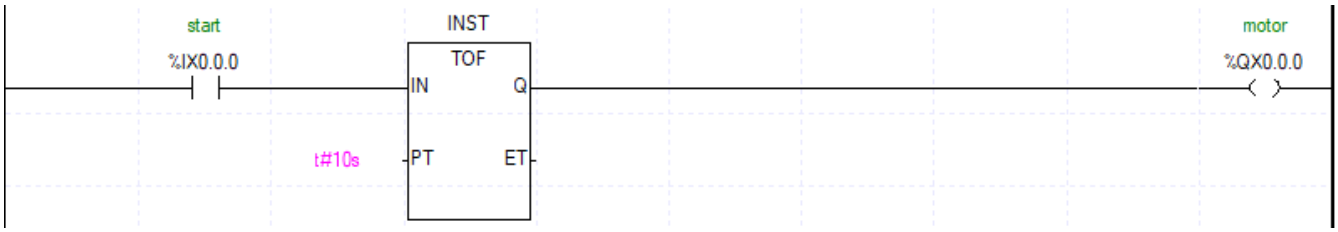


تایمر تاخیر در قطع (TOF)

TOF مخفف off delay timer می‌باشد. با یک شدن مقدار متغیری که به پایه IN واگذارشده است بلافاصله خروجی فعال می‌گردد ولی تایمر زمانگیری را شروع نمی‌کند تا پایه IN صفر شود سپس زمان‌شماری تایمر آغاز می‌شود و پس از سپری شدن زمان تعیین شده خروجی صفر می‌گردد. اگر در هنگام زمانگیری مجدد پایه ورودی فعال شود زمانگیری متوقف می‌گردد تا پایه ورودی غیرفعال شود و زمانگیری از ابتدا شروع شود. بعبارتی خروجی پس از طی زمانی خاص خاموش می‌گردد و بلافاصله خروجی قطع نمی‌شود. زمان مورد نظر در پایه PT تنظیم می‌گردد. فرمت وارد کردن زمان بصورت T#number می‌باشد. حتما بعد از عدد زمان(ثانیه یا دقیقه یا ساعت) را وارد نمایید. مثلا T#25s, T#21m25s, T#2h10m24s . پایه ET زمان سپری شده را نمایش می‌دهد.

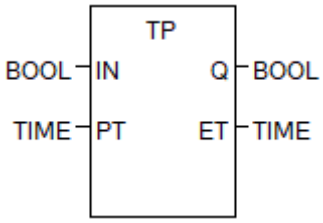


مثال: با زدن یک شستی موتور روشن شده و پس از ۱۰ثانیه موتور خاموش گردد.

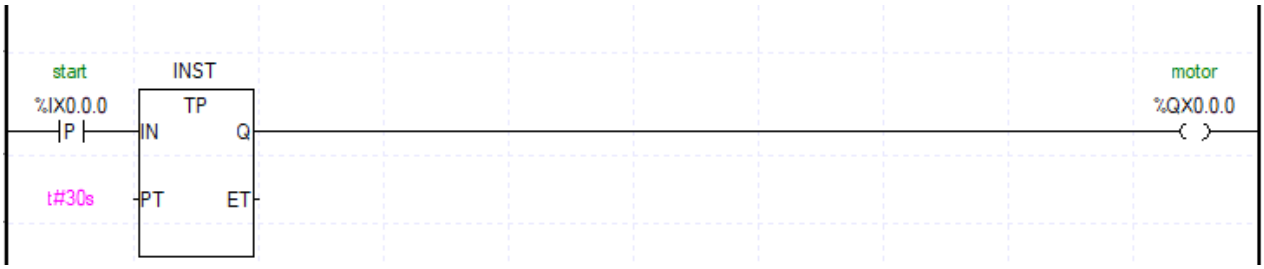


تایمر ضربه (TP)

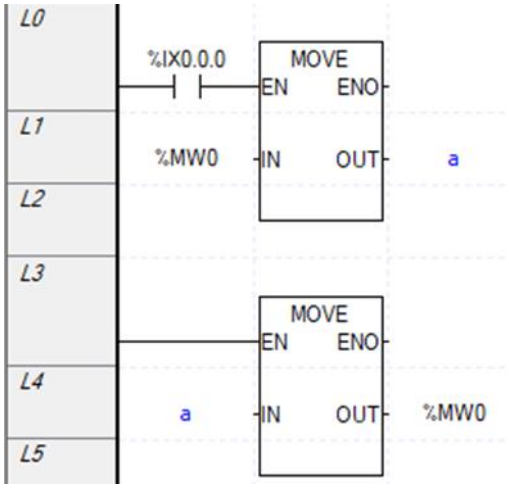
TP مخفف Pulse timer می‌باشد. برای اینکه زمان‌شماری به درستی انجام شود باید به پایه ورودی (IN) تایمر یک منطقی اعمال نمود. و نیازی به اعمال یک دائم به پایه ورودی نمی‌باشد. این تایمر حتی با یک لبه بالارونده نیز فعال شده و خروجی فعال می‌گردد و پس از پایان زمان مدنظر خروجی غیرفعال می‌گردد. زمان مورد نظر در پایه PT تنظیم می‌گردد. فرمت وارد کردن زمان بصورت T#number می‌باشد. حتما بعد از عدد زمان(ثانیه یا دقیقه یا ساعت) را وارد نمایید. مثلا T#25s, T#21m25s, T#2h10m24s . پایه ET زمان سپری شده را نمایش می‌دهد.



مثال: با فشردن یک شستی موتور ۳۰ ثانیه روشن شده و سپس خاموش گردد.

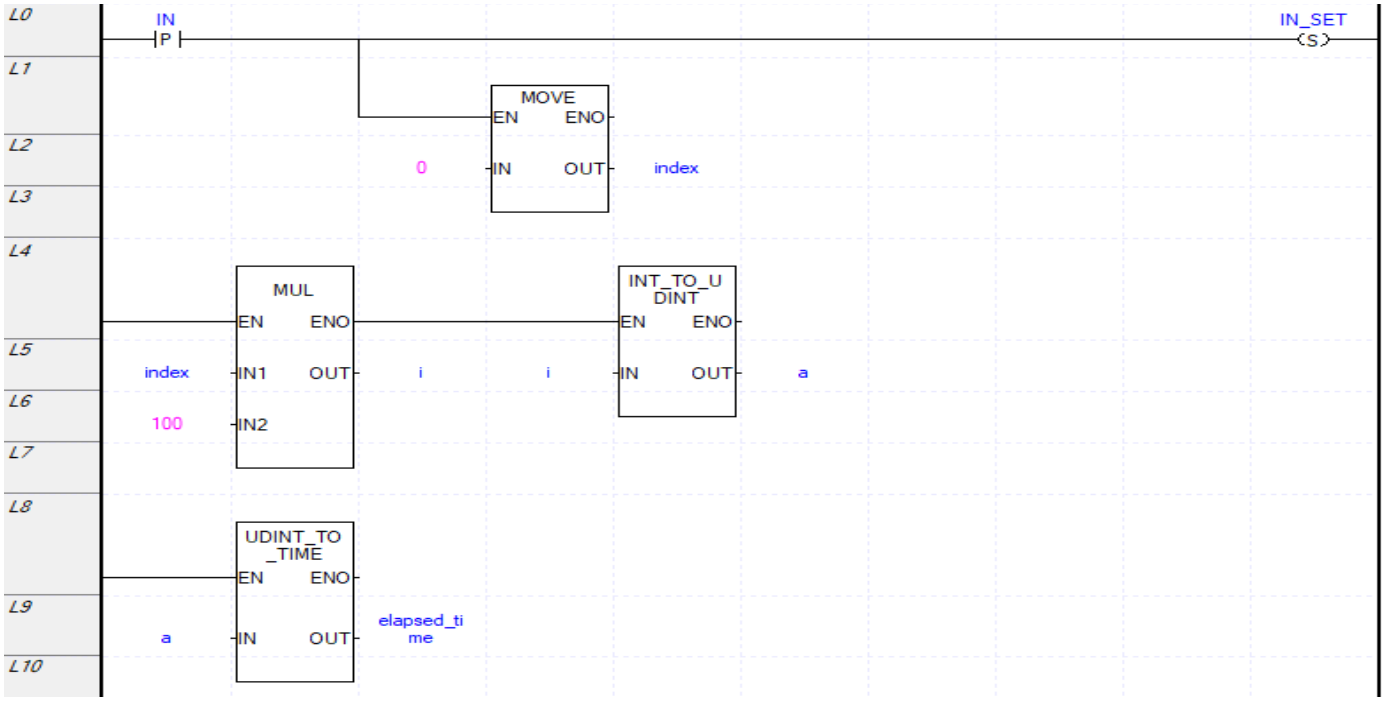


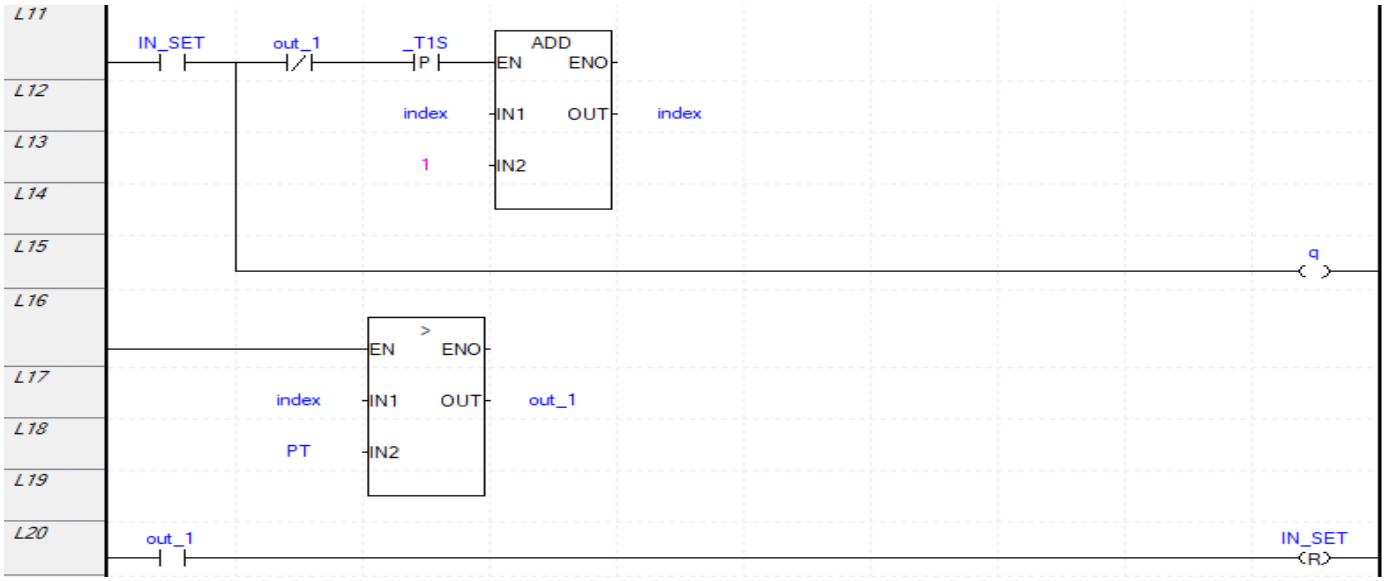
مثال: برنامه‌ای بنویسید که تا یکی از ورودی‌ها فعال نشده نتوان مقدار متغیر را تغییر داد.

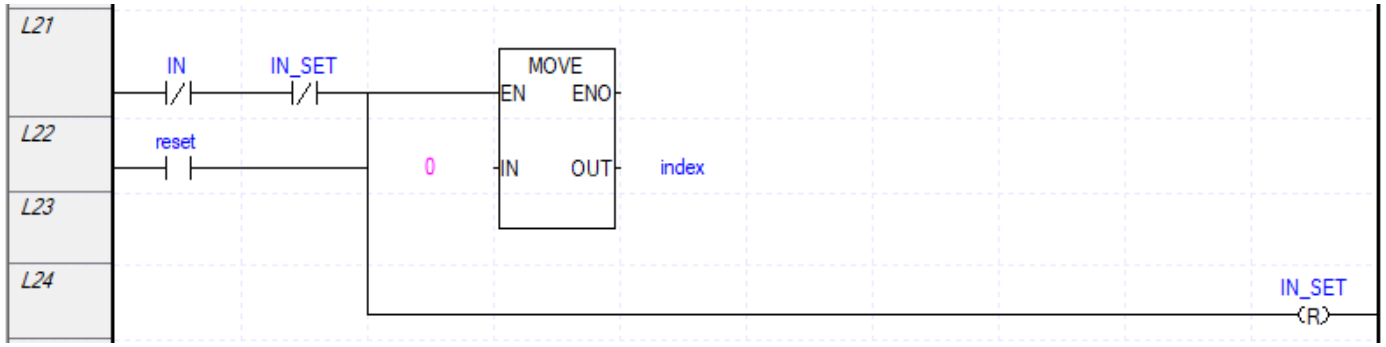


ساخت تایمر پالس

خط اول برای شناسایی و ذخیره لبه می‌باشد. این تایمر دارای پایه ریست می‌باشد.

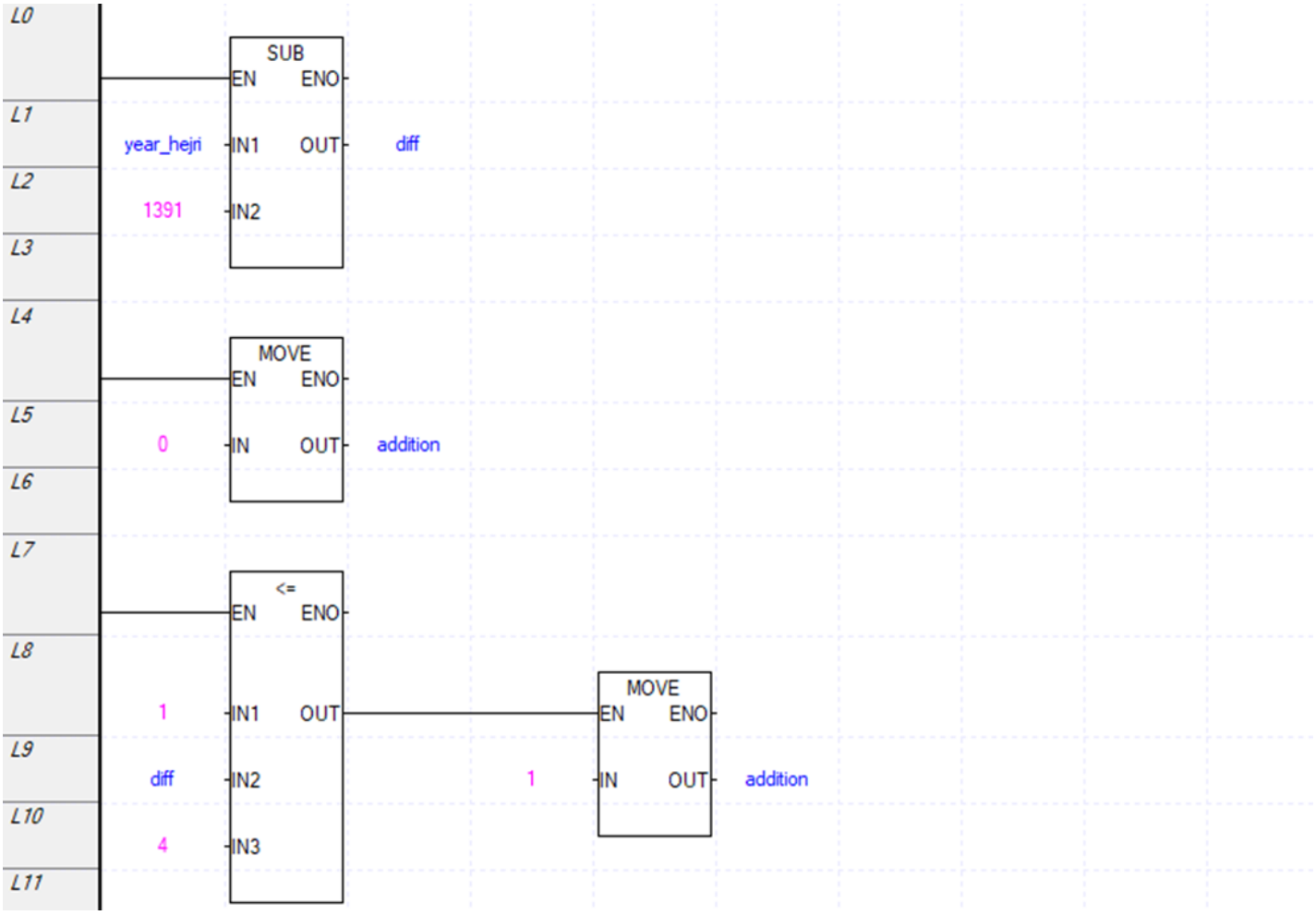


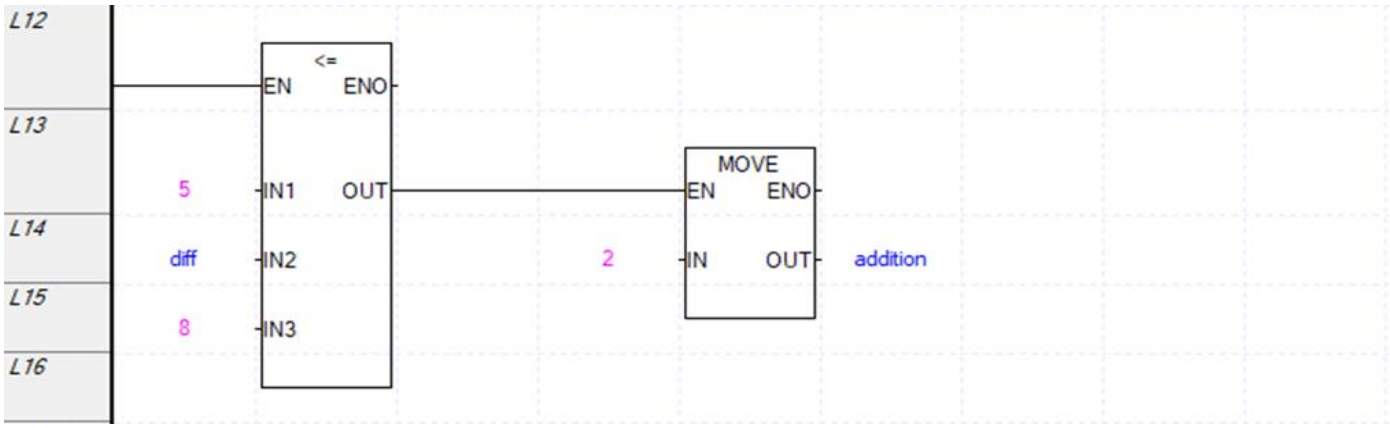


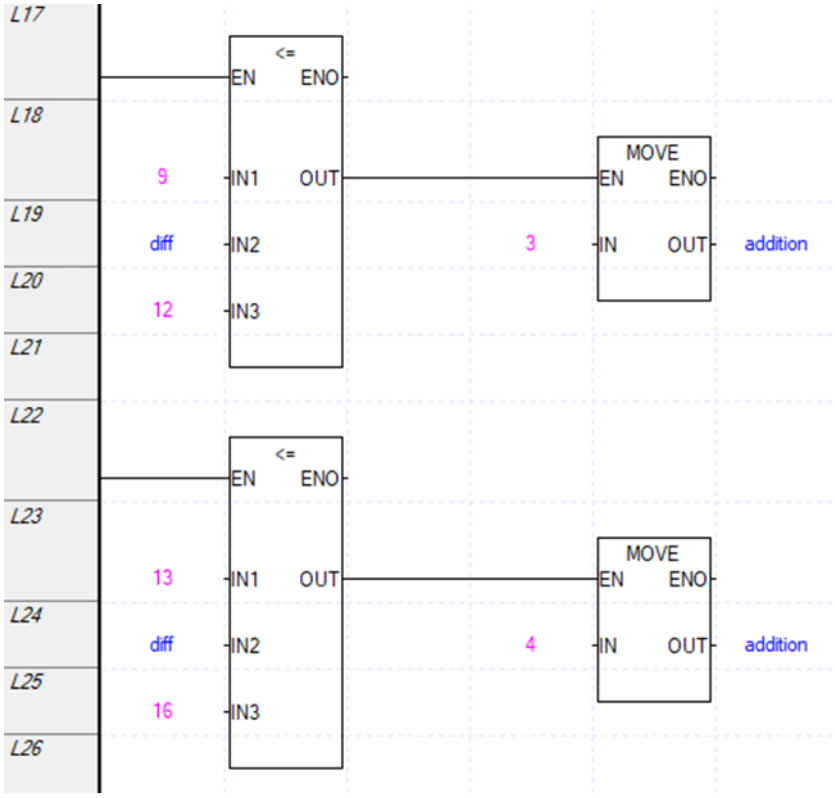


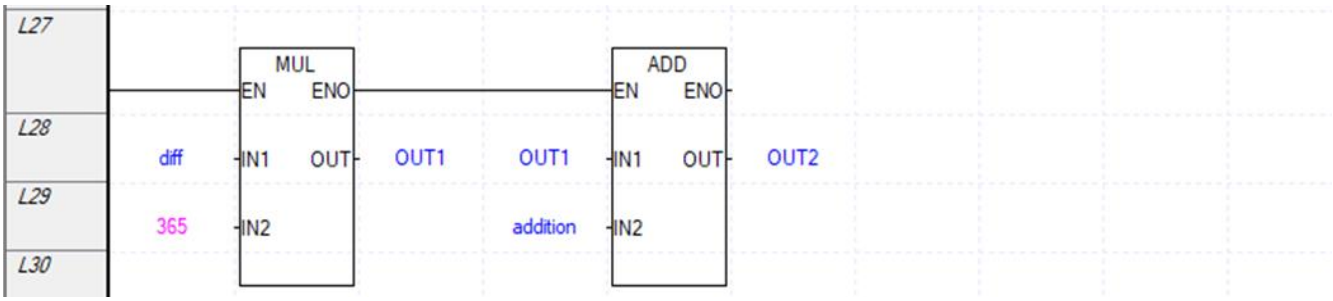
تبدیل تاریخ شمسی به میلادی

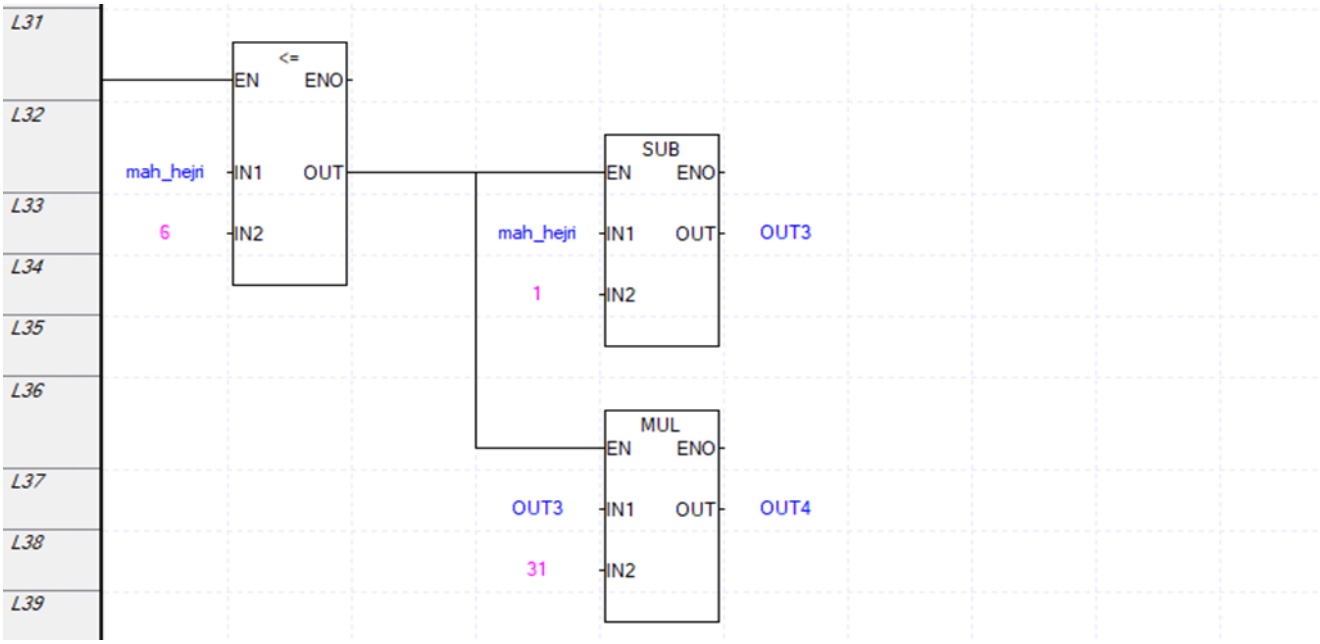
مبنا را سال ۱۳۹۱ درنظر می‌گیریم چون کبیسه است. و تا سال ۱۴۰۷ برنامه آن را می‌نویسیم.

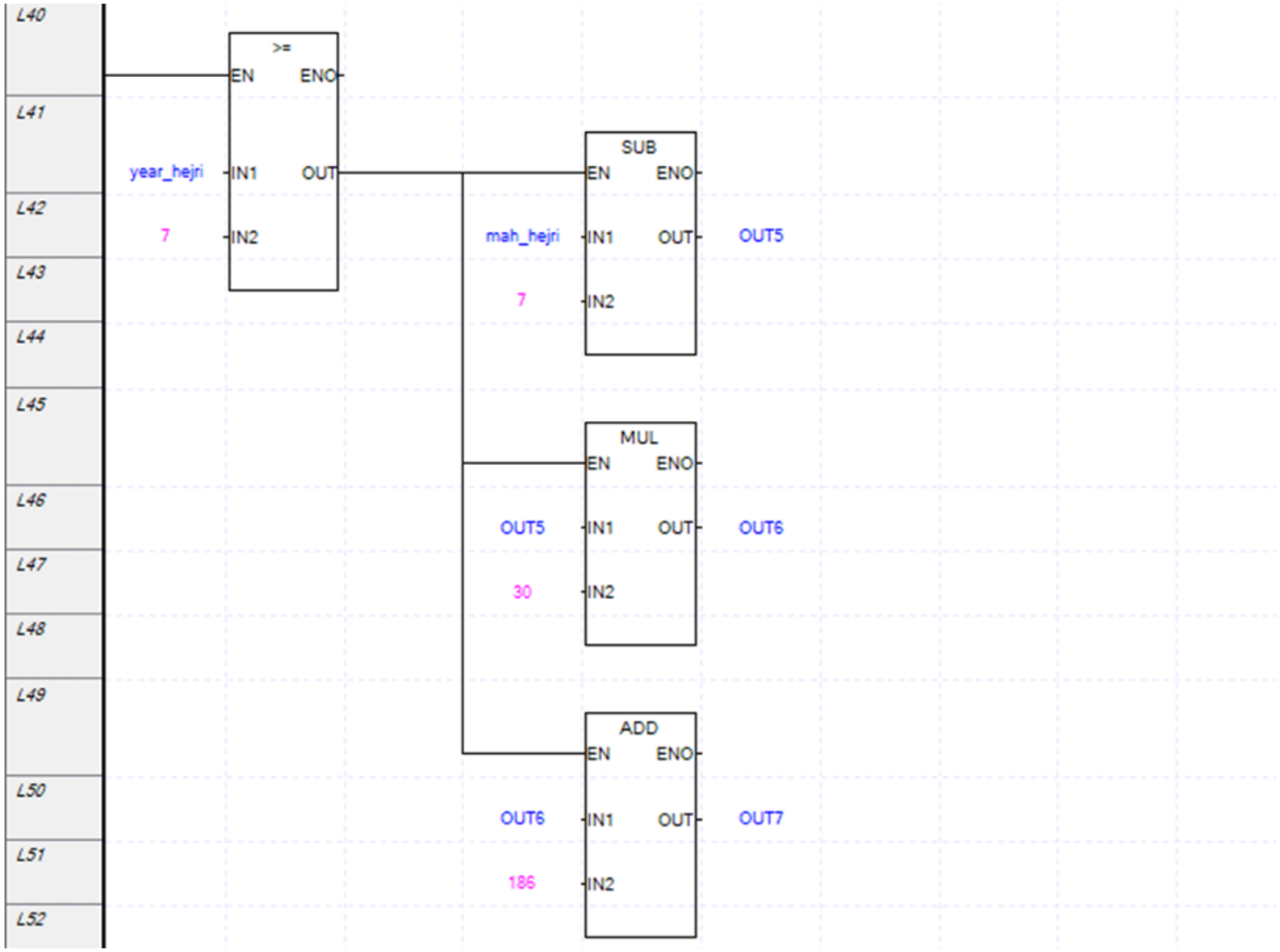


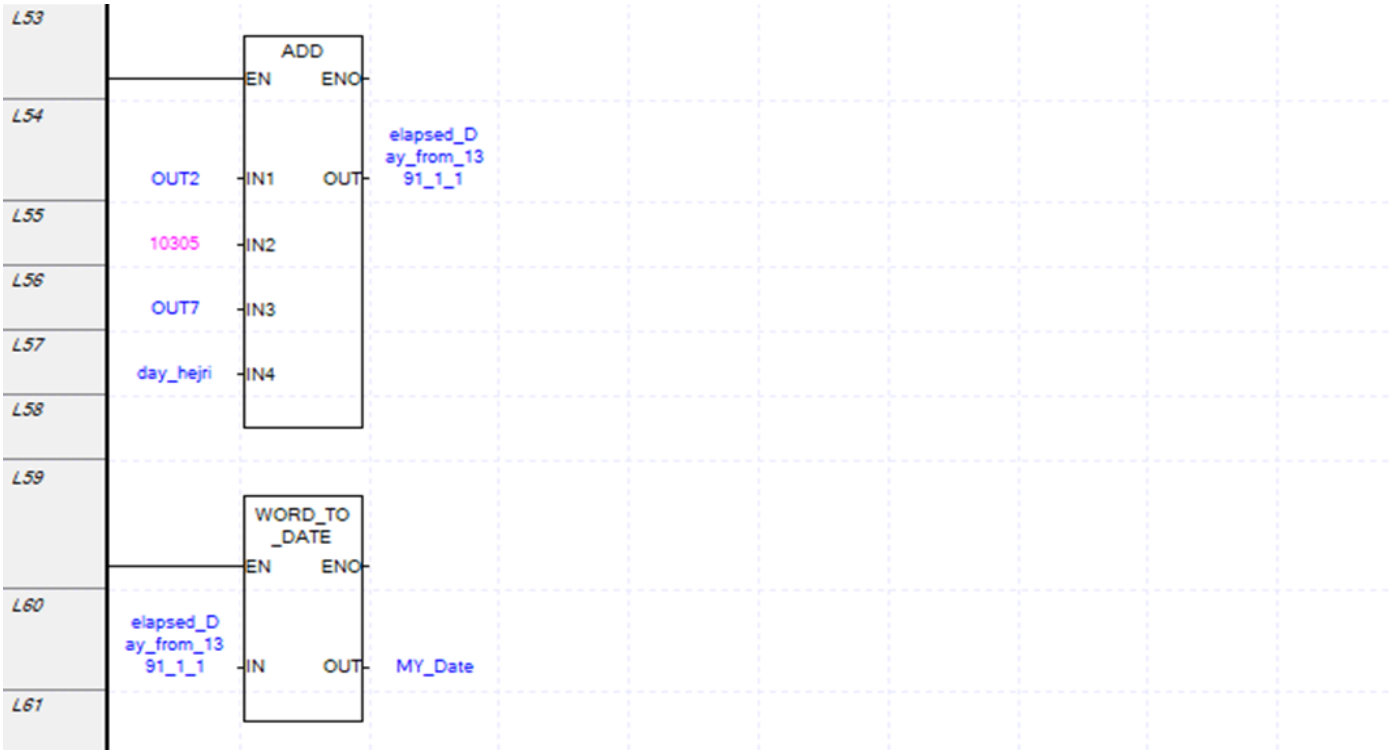



L17
L18
L19
L20
L21
L22
L23
L24
L25
L26
<=
EN
ENO
9
IN1
OUT
diff
IN2
12
IN3
MOVE
EN
ENO
3
IN
OUT
addition
<=
EN
ENO
13
IN1
OUT
diff
IN2
16
IN3
MOVE
EN
ENO
4
IN
OUT
addition


L27
L28
L29
L30
MUL
EN
ENO
diff
IN1
OUT
OUT1
365
IN2
ADD
EN
ENO
OUT1
IN1
OUT
OUT2
addition
IN2

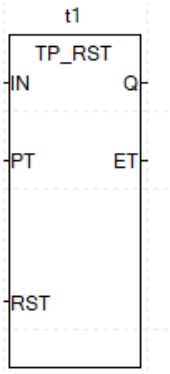

L31
L32
L33
L34
L35
L36
L37
L38
L39
<=
EN
ENO
mah_hejri
IN1
OUT
6
IN2
SUB
EN
ENO
mah_hejri
IN1
OUT
OUT3
1
IN2
MUL
EN
ENO
OUT3
IN1
OUT
OUT4
31
IN2


L40
L41
L42
L43
L44
L45
L46
L47
L48
L49
L50
L51
L52
>=
EN
ENO
year_hejri
IN1
OUT
7
IN2
SUB
EN
ENO
mah_hejri
IN1
OUT
OUT5
7
IN2
MUL
EN
ENO
OUT5
IN1
OUT
OUT6
30
IN2
ADD
EN
ENO
OUT6
IN1
OUT
OUT7
186
IN2


L53
L54
L55
L56
L57
L58
L59
L60
L61
ADD
EN
ENO
OUT2
IN1
OUT
elapsed_Day_from_1391_1_1
10305
IN2
OUT7
IN3
day_hejri
IN4
WORD_TO_DATE
EN
ENO
elapsed_Day_from_1391_1_1
IN
OUT
MY_Date

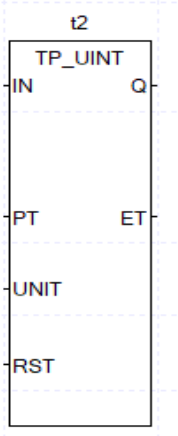
تایمر پالس دارای پایه ریست

تفاوت این تایمر با تایمر پالسی در پایه ریست آن می‌باشد و اصول عملکردی آن با تایمر پالسی مشابه است. به پایه IN ورودی تایمر را جهت شروع زمان‌شماری وصل می‌کنیم در پایه PT مقدار زمان مدنظر را با فرمت T#..... وارد نمایید. پایه RST جهت ریست کردن تایمر می‌باشد.



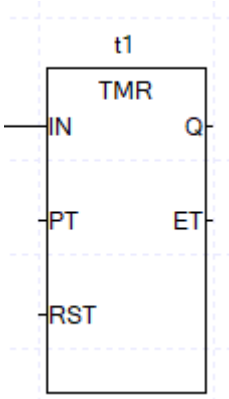
تایمر پالس UINT

این تایمر علاوه بر دارا بودن پایه ریست، مقدار زمان آن بصورت ضریبی از دو عدد می‌باشد که حاصلضرب این دوعدد برحسب میلی ثانیه می‌باشد. این ضرب بر روی مقدار دو پایه PT و UINT انجام شده و زمان برحسب میلی‌ثانیه را شمارش می‌کند. یک پایه می‌تواند واحد زمان را از میلی‌ثانیه به ثانیه، دقیقه و یا ساعت تبدیل کند.



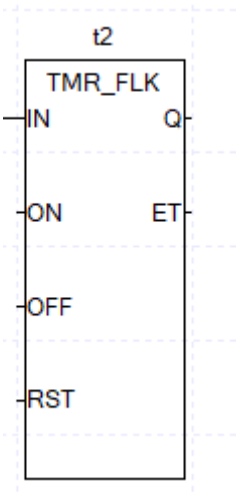
تایمر TMR

ورودی این تایمر باید یک لبه باشد. پس از پایان زمان‌گیری خروجی فعال می ماند تا تایمر ریست نشود تایمر مجدد فعال نمی‌شود. اگر ورودی تایمر غیرفعال و مجدد فعال شود زمان‌شماری از ادامه زمان سپری شده انجام می‌شود تا زمان پایان یابد.



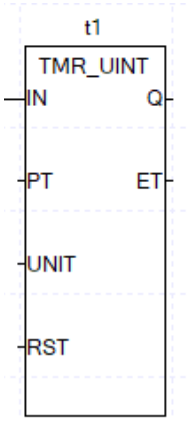
تایمر TMR_FLK

این تایمر بصورت چشمک‌زن عمل می‌کند و زمان روشن و خاموش بودن آن را تنظیم می‌کنیم و از طریق پایه RST تایمر را ریست نمایید. از این تایمر برای اعلام هشدار می‌توان استفاده نمود



تایمر TMR_UINT

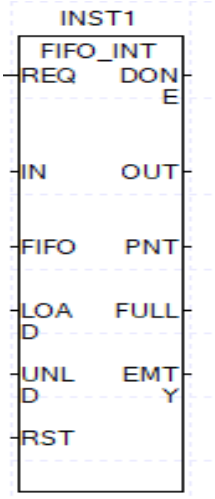
علاوه بر پایه ریست، مقدار زمان بصورت ضریبی از دو عدد می‌باشد که حاصلضرب این دوعدد برحسب میلی‌ثانیه می‌باشد. این ضرب بر روی مقدار دو پایه PT و UINT انجام شده و زمان برحسب میلی‌ثانیه را شمارش می‌کند. یک پایه می‌تواند واحد زمان را از میلی‌ثانیه به ثانیه، دقیقه و یا ساعت تبدیل کند.



حافظه Stack

قراردادن داده درون حافظه از طریق پالس. برای ترتیب چیدن و یا استفاده از داده از دو واژه FIFO و LIFO استفاده می‌شود. FIFO و LIFO یک آرایه هستند.

FIFO: First In First Out، LIFO: Last In First Out

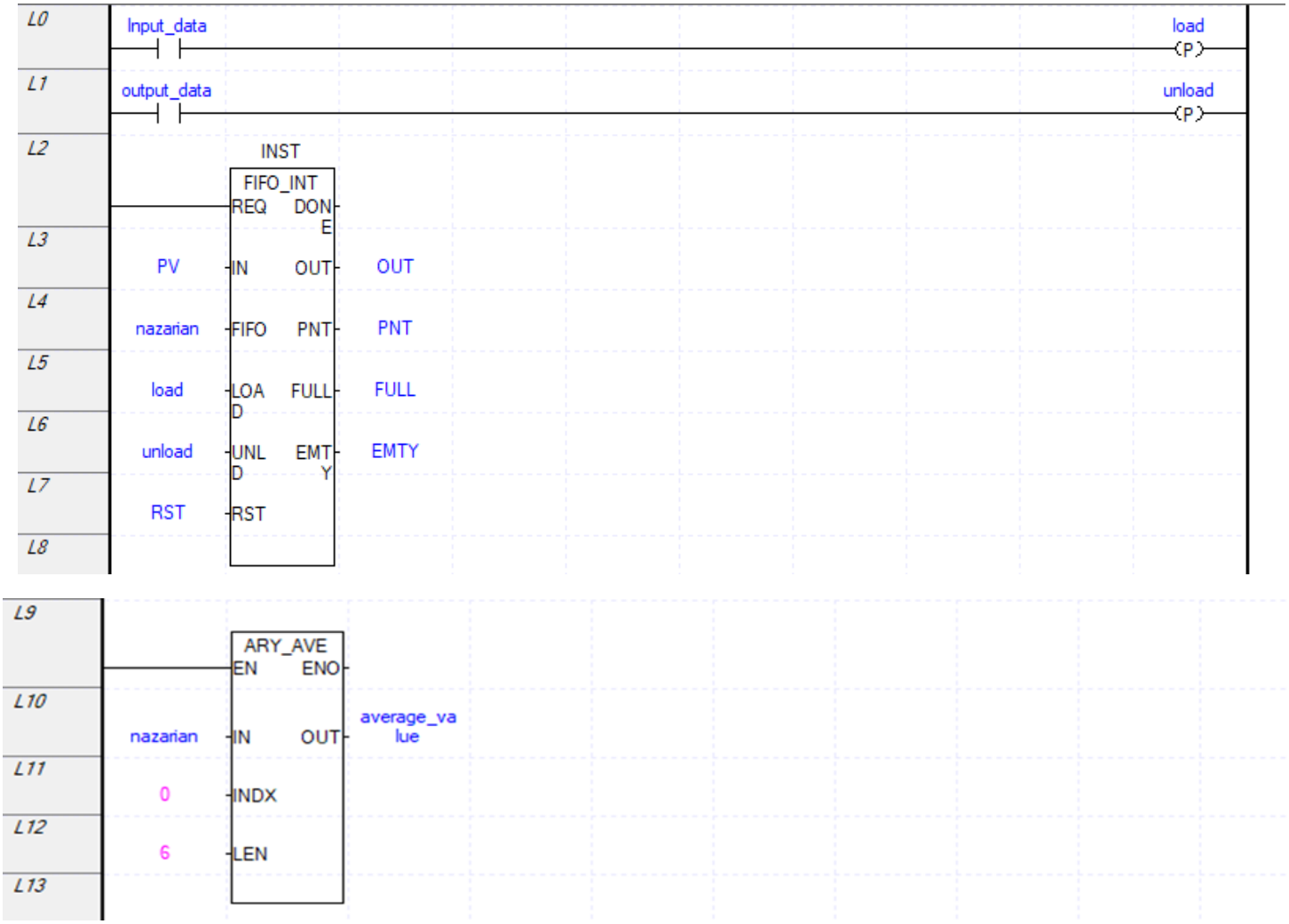


داده به پایه IN در تابع FIFO وصل می‌شود. به پایه FIFO یک اسم با فرمت آرایه ۶عنصری USINT نسبت دهید. پایه LOAD داده را درون آرایه قرار می‌دهد و پایه UNLOAD داده را از این آرایه استخراج می‌کند. با پایه RST تمام عناصر آرایه صفر می‌شود. پایه PNT یا Pointer هرلحظه مکان حافظه را نشان می‌دهد. پایه EMTY اگر یک باشد یعنی حافظه خالی است. پایه FULL زمانی که حافظه پر شود فعال می‌شود. با فعال کردن پایه Unload اولین

ورودی که داده شده بود در پایه Out نمایش داده شده و از حافظه حذف می‌شود و مقدار PNT یک عدد کاهش می‌یابد.

LIFO نیز مانند FIFO می‌باشد تنها تفاوت آن در حذف مقدار از حافظه است یعنی با فعال کردن پایه Unload آخرین مقدار در پایه Out نمایش داده و سپس از حافظه حذف می‌شود و مقدار PNT یک عدد کاهش می‌یابد.

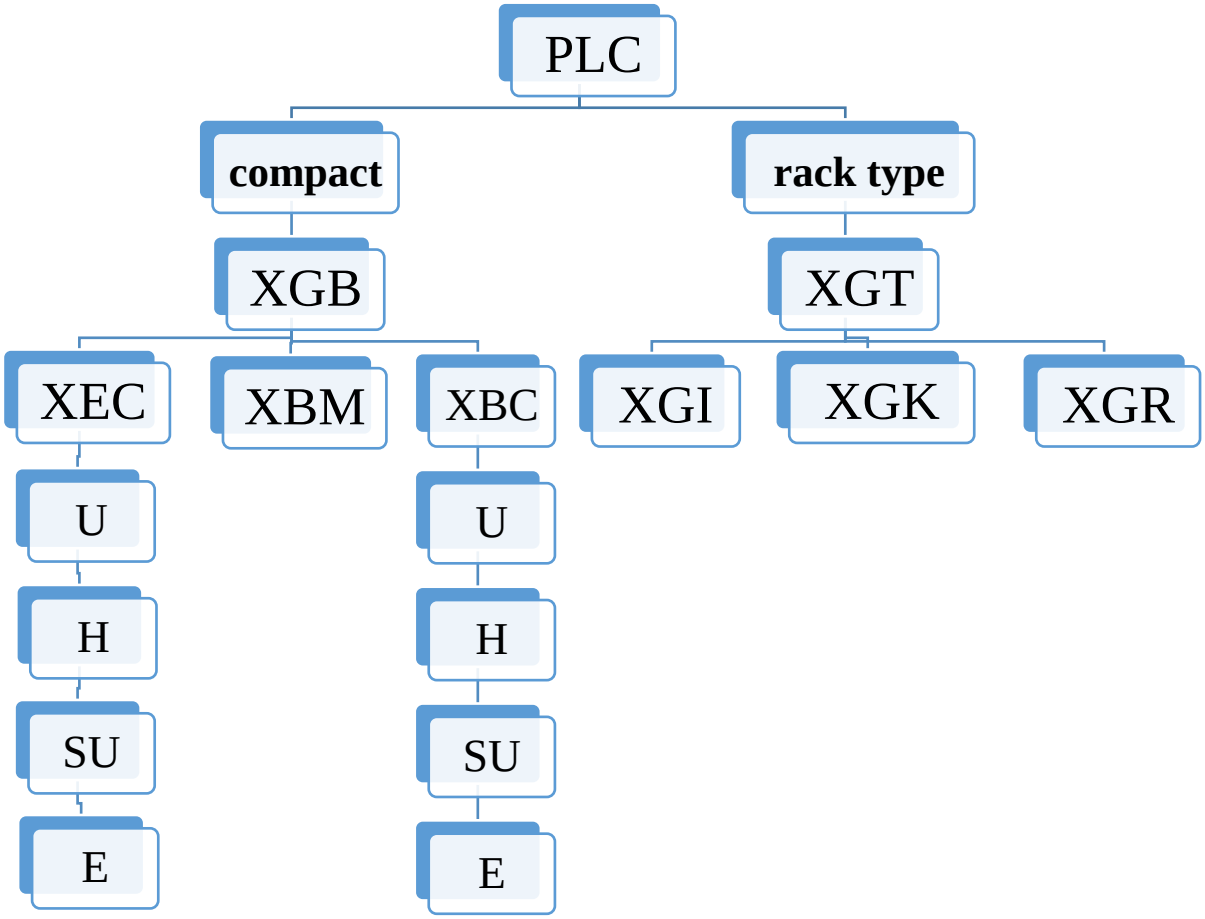
مثال: میانگین ۵۰ داده اخیر را محاسبه نمایید.



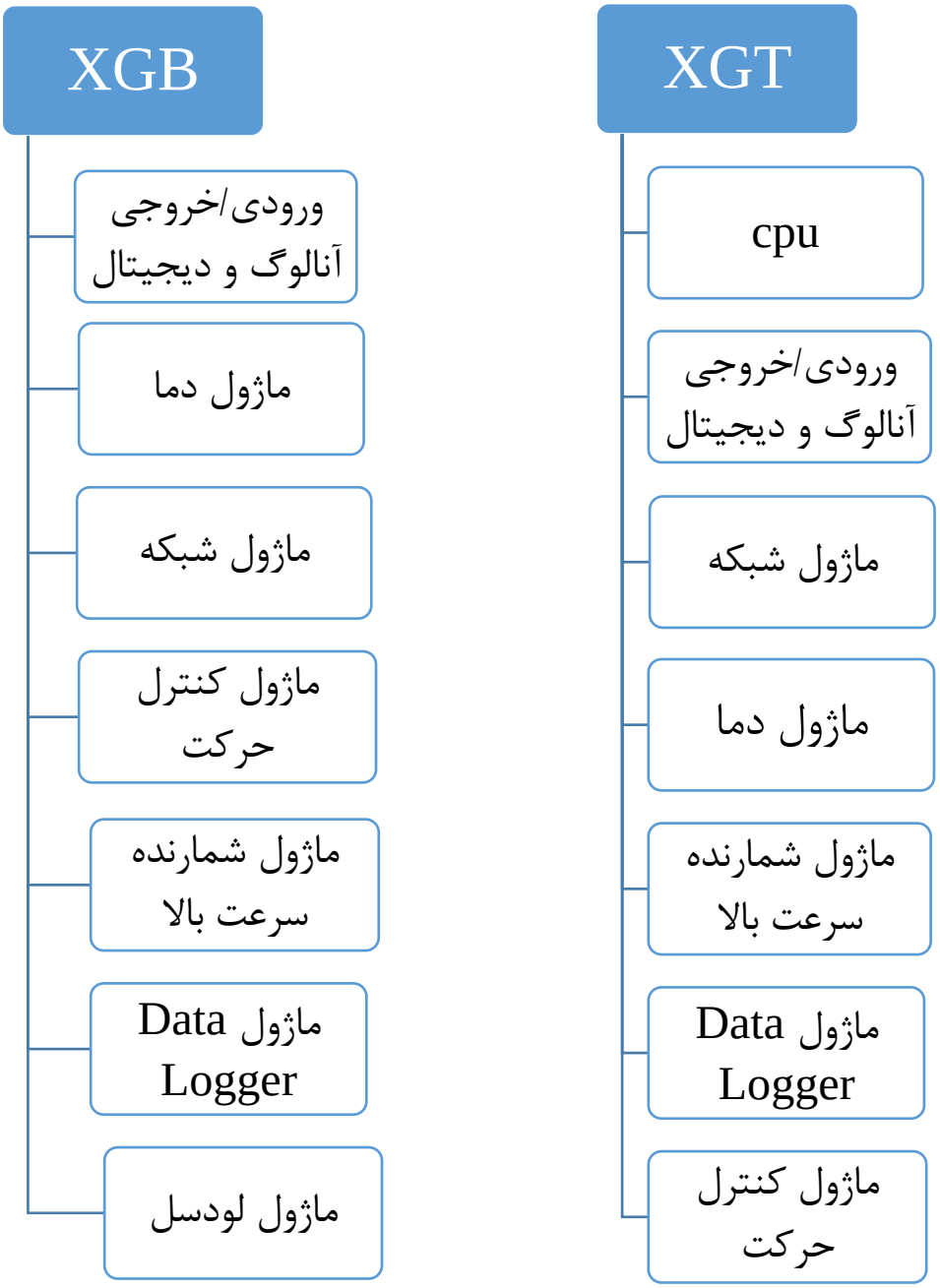
پیوست۱

سخت‌افزار PLCهای برند LS

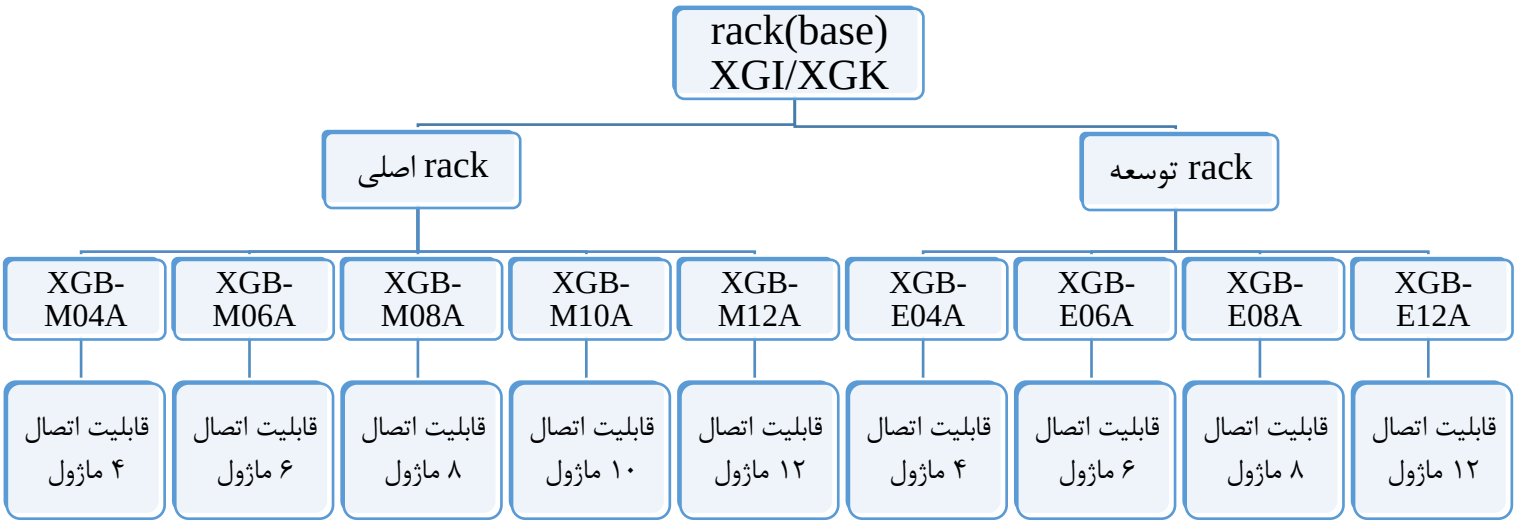
PLCهای سری جدید برند LS به دو خانواده XGT و XGB دسته بندی می‌شوند. که دارای CPUهای مختلفی می‌باشند.

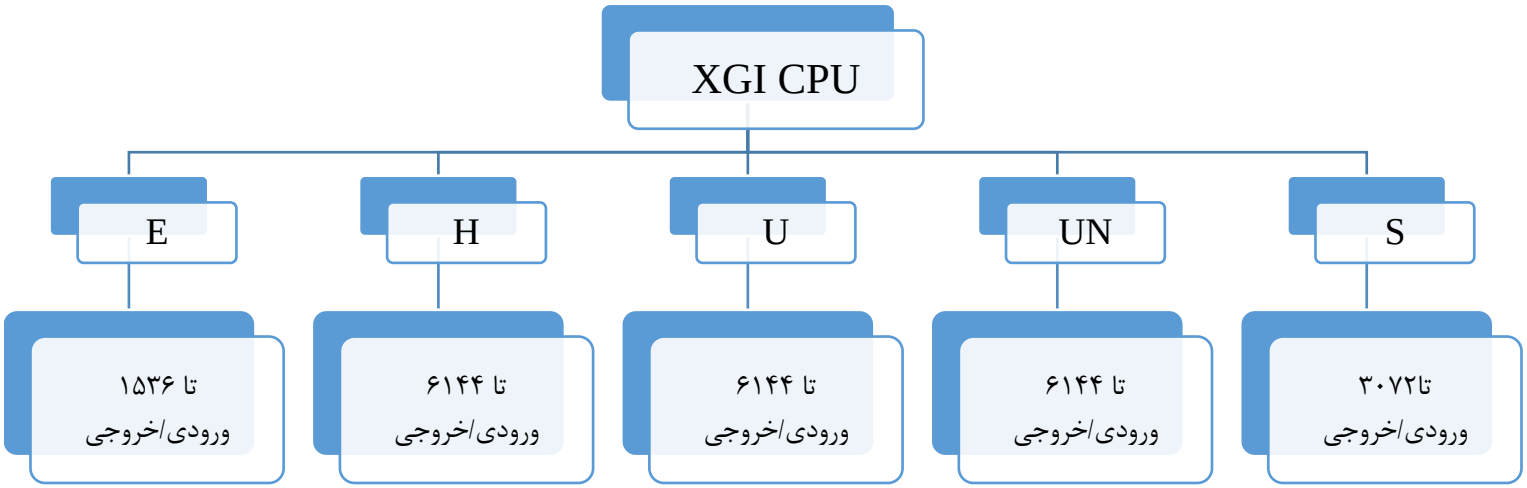


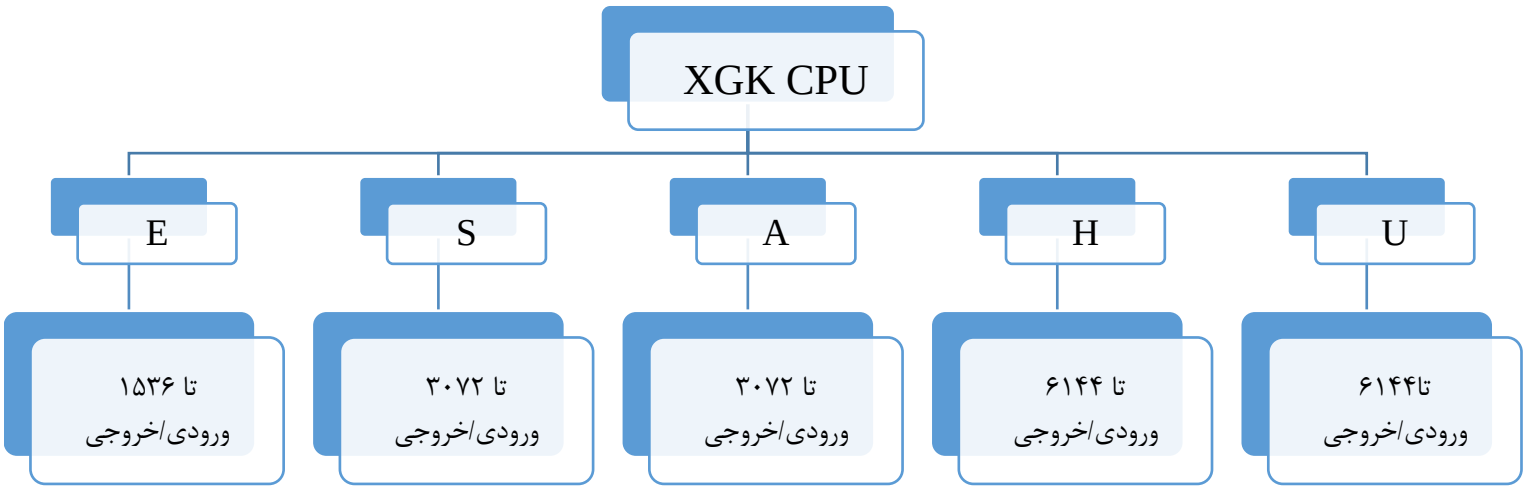
ماژول‌های PLCهای مختلف

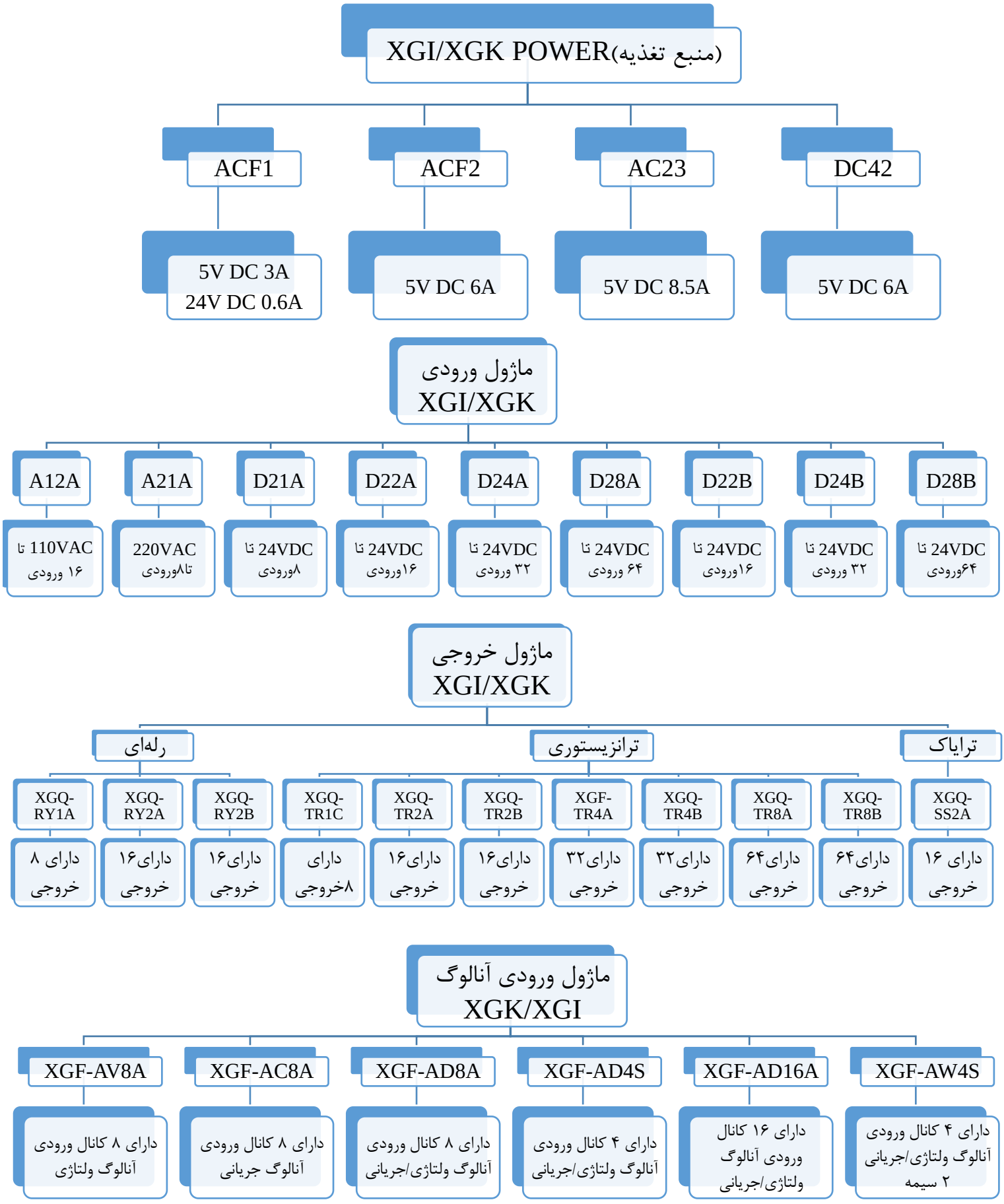


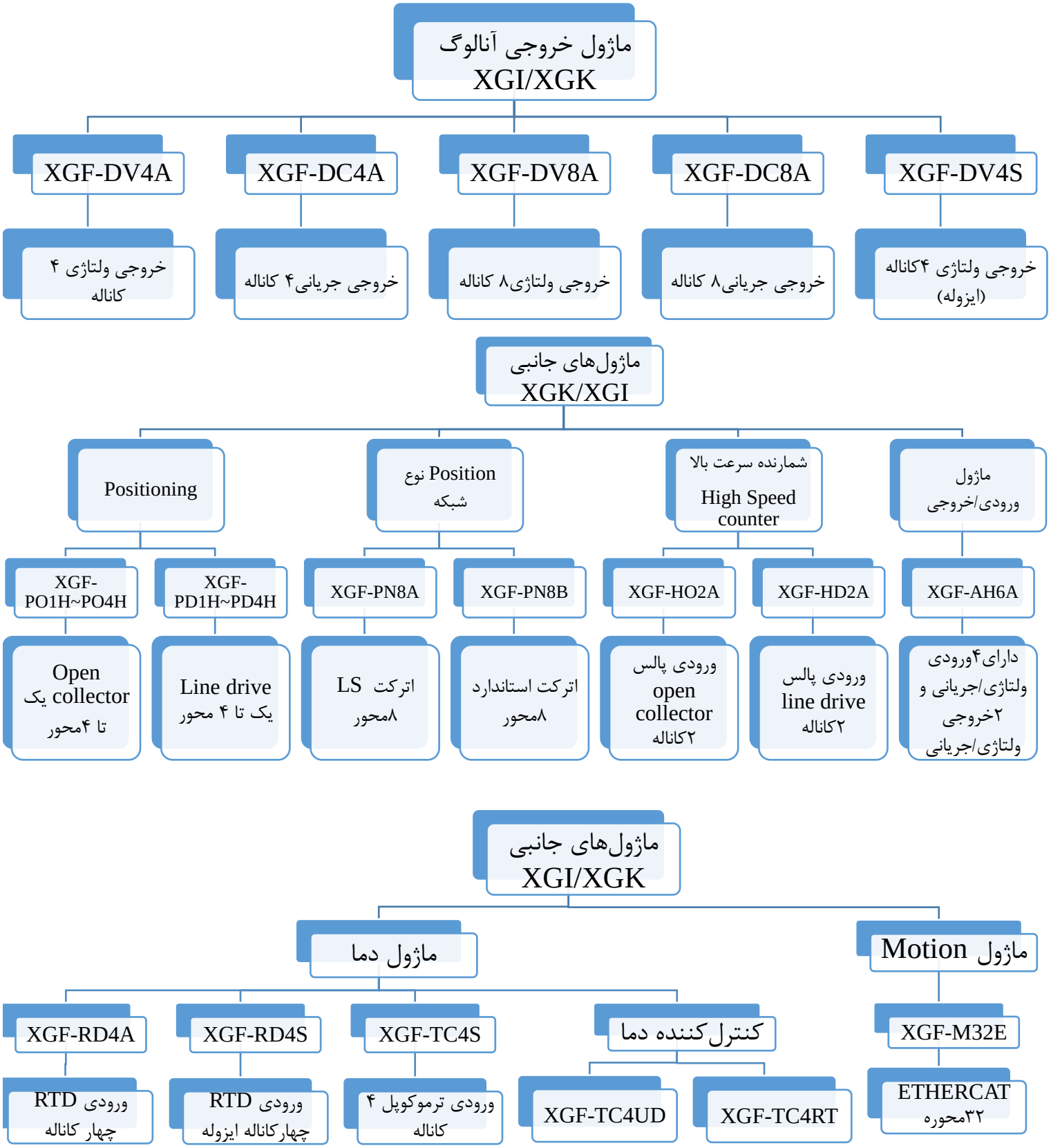
سری XGI/XGK

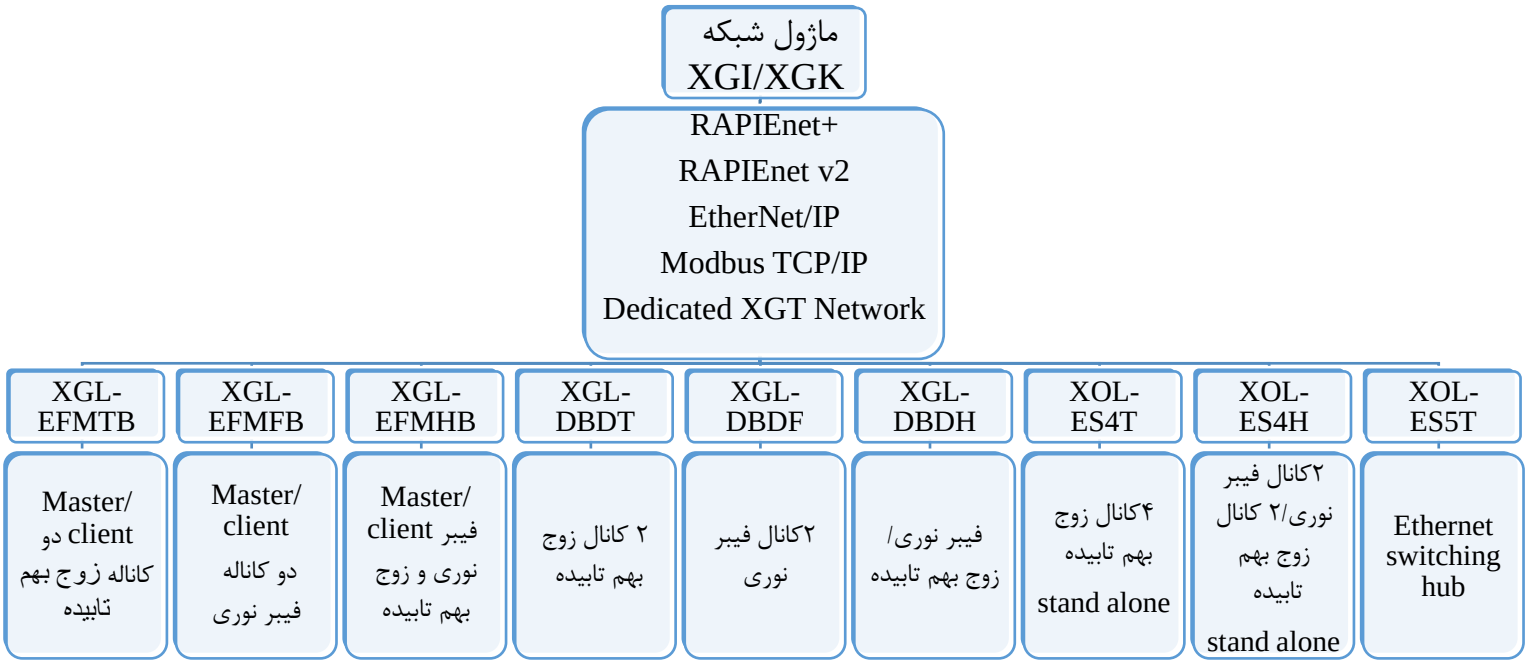

rack(base)
XGI/XGK
rack اصلی
rack توسعه
XGB-M04A
XGB-M06A
XGB-M08A
XGB-M10A
XGB-M12A
XGB-E04A
XGB-E06A
XGB-E08A
XGB-E12A
قابلیت اتصال ۴ ماژول
قابلیت اتصال ۶ ماژول
قابلیت اتصال ۸ ماژول
قابلیت اتصال ۱۰ ماژول
قابلیت اتصال ۱۲ ماژول
قابلیت اتصال ۴ ماژول
قابلیت اتصال ۶ ماژول
قابلیت اتصال ۸ ماژول
قابلیت اتصال ۱۲ ماژول

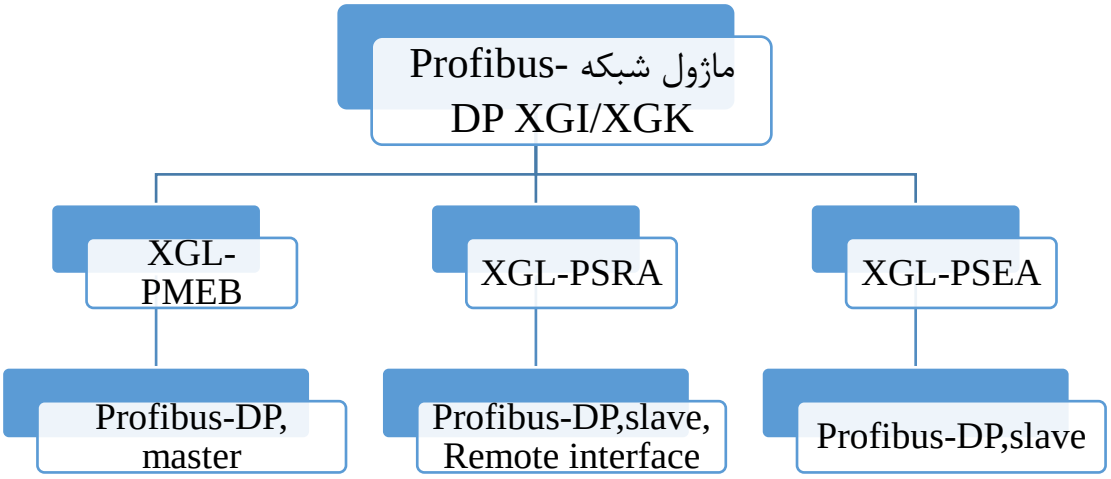

XGI CPU
E
H
U
UN
S
تا ۱۵۳۶ ورودی/خروجی
تا ۶۱۴۴ ورودی/خروجی
تا ۶۱۴۴ ورودی/خروجی
تا ۶۱۴۴ ورودی/خروجی
تا۳۰۷۲ ورودی/خروجی

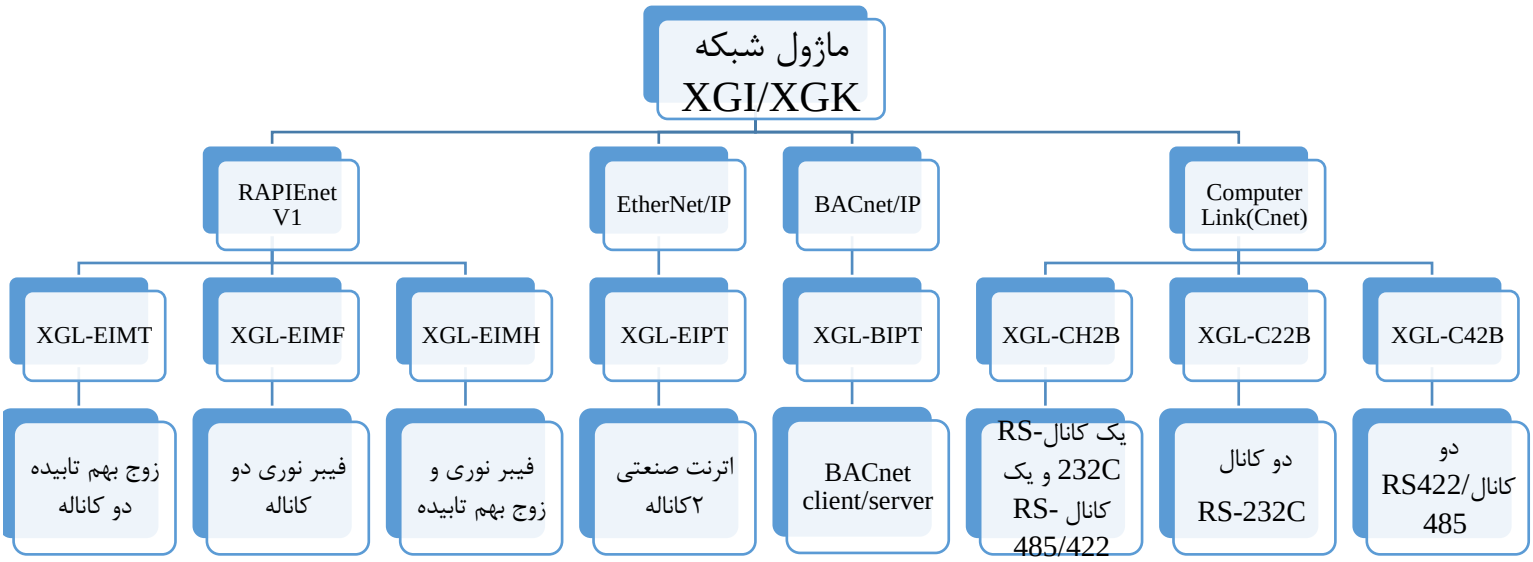

XGK CPU
E
S
A
H
U
تا ۱۵۳۶ ورودی/خروجی
تا ۳۰۷۲ ورودی/خروجی
تا ۳۰۷۲ ورودی/خروجی
تا ۶۱۴۴ ورودی/خروجی
تا۶۱۴۴ ورودی/خروجی

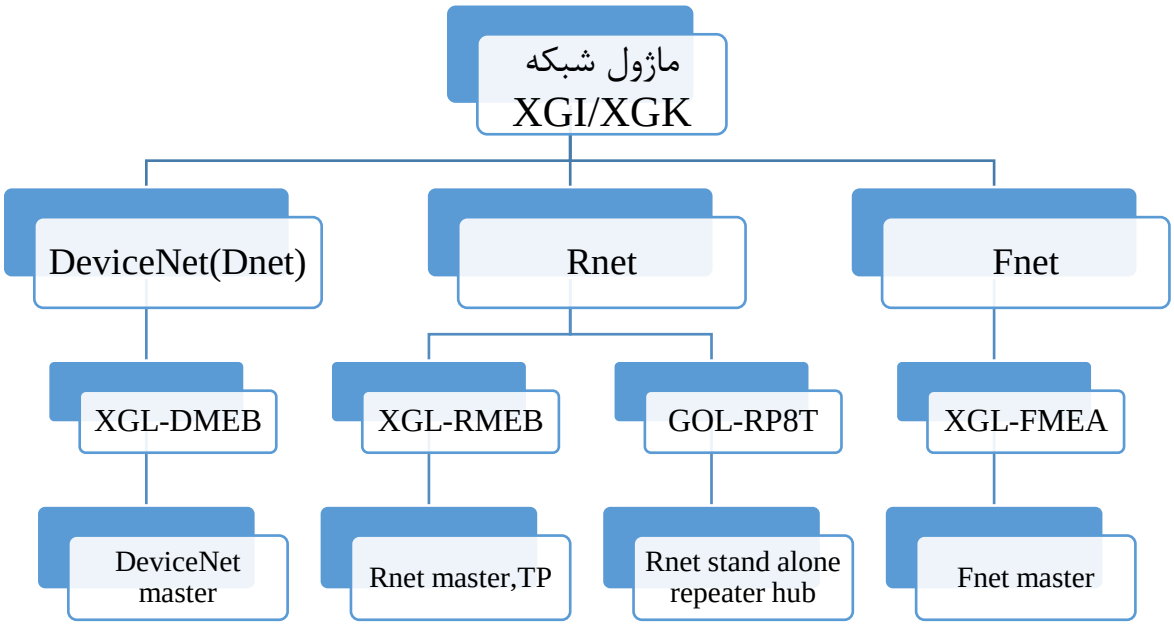

XGI/XGK POWER(منبع تغذیه)
ACF1
ACF2
AC23
DC42
5V DC 3A
24V DC 0.6A
5V DC 6A
5V DC 8.5A
5V DC 6A
ماژول ورودی
XGI/XGK
A12A
A21A
D21A
D22A
D24A
D28A
D22B
D24B
D28B
110VAC تا
۱۶ ورودی
220VAC
تا۸ورودی
24VDC تا
۸ورودی
24VDC تا
۱۶ورودی
24VDC تا
۳۲ ورودی
24VDC تا
۶۴ ورودی
24VDC تا
۱۶ورودی
24VDC تا
۳۲ ورودی
24VDC تا
۶۴ورودی
ماژول خروجی
XGI/XGK
رله‌ای
ترانزیستوری
ترایاک
XGQ-RY1A
XGQ-RY2A
XGQ-RY2B
XGQ-TR1C
XGQ-TR2A
XGQ-TR2B
XGF-TR4A
XGQ-TR4B
XGQ-TR8A
XGQ-TR8B
XGQ-SS2A
دارای ۸ خروجی
دارای۱۶ خروجی
دارای۱۶ خروجی
دارای ۸خروجی
دارای۱۶ خروجی
دارای۱۶ خروجی
دارای۳۲ خروجی
دارای۳۲ خروجی
دارای۶۴ خروجی
دارای۶۴ خروجی
دارای ۱۶ خروجی
ماژول ورودی آنالوگ
XGK/XGI
XGF-AV8A
XGF-AC8A
XGF-AD8A
XGF-AD4S
XGF-AD16A
XGF-AW4S
دارای ۸ کانال ورودی آنالوگ ولتاژی
دارای ۸ کانال ورودی آنالوگ جریانی
دارای ۸ کانال ورودی آنالوگ ولتاژی/جریانی
دارای ۴ کانال ورودی آنالوگ ولتاژی/جریانی
دارای ۱۶ کانال ورودی آنالوگ ولتاژی/جریانی
دارای ۴ کانال ورودی آنالوگ ولتاژی/جریانی ۲ سیمه


ماژول خروجی آنالوگ
XGI/XGK
XGF-DV4A
XGF-DC4A
XGF-DV8A
XGF-DC8A
XGF-DV4S
خروجی ولتاژی ۴ کاناله
خروجی جریانی۴ کاناله
خروجی ولتاژی۸ کاناله
خروجی جریانی۸ کاناله
خروجی ولتاژی ۴کاناله (ایزوله)
ماژول‌های جانبی
XGK/XGI
Positioning
نوع Position شبکه
شمارنده سرعت بالا High Speed counter
ماژول ورودی/خروجی
XGF-PO1H~PO4H
XGF-PD1H~PD4H
XGF-PN8A
XGF-PN8B
XGF-HO2A
XGF-HD2A
XGF-AH6A
Open collector یک تا ۴محور
Line drive یک تا ۴ محور
اترکت LS ۸محور
اترکت استاندارد ۸محور
ورودی پالس open collector ۲کاناله
ورودی پالس line drive ۲کاناله
دارای۴ورودی ولتاژی/جریانی و ۲خروجی ولتاژی/جریانی
ماژول‌های جانبی
XGI/XGK
ماژول دما
ماژول Motion
XGF-RD4A
XGF-RD4S
XGF-TC4S
کنترل‌کننده دما
XGF-M32E
ورودی RTD چهار کاناله
ورودی RTD چهارکاناله ایزوله
ورودی ترموکوپل ۴ کاناله
XGF-TC4UD
XGF-TC4RT
ETHERCAT ۳۲محوره

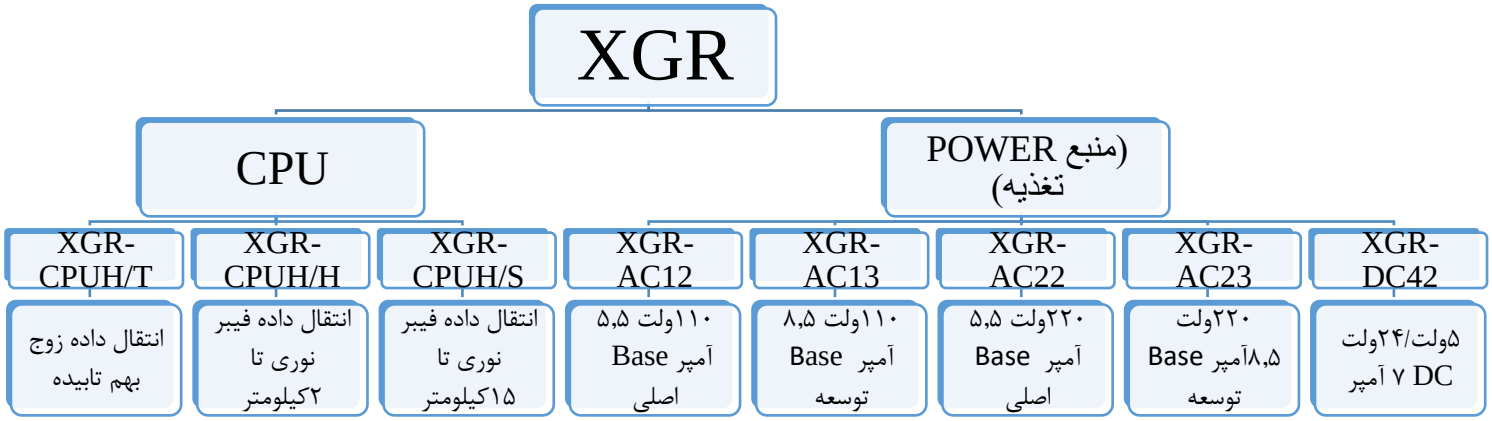

ماژول شبکه XGI/XGK
RAPIEnet+
RAPIEnet v2
EtherNet/IP
Modbus TCP/IP
Dedicated XGT Network
XGL-EFMTB
XGL-EFMFB
XGL-EFMHB
XGL-DBDT
XGL-DBDF
XGL-DBDH
XOL-ES4T
XOL-ES4H
XOL-ES5T
Master/ client دو کاناله زوج بهم تابیده
Master/ client دو کاناله فیبر نوری
Master/ client فیبر نوری و زوج بهم تابیده
۲ کانال زوج بهم تابیده
۲کانال فیبر نوری
فیبر نوری/ زوج بهم تابیده
۴کانال زوج بهم تابیده stand alone
۲کانال فیبر نوری/۲ کانال زوج بهم تابیده stand alone
Ethernet switching hub

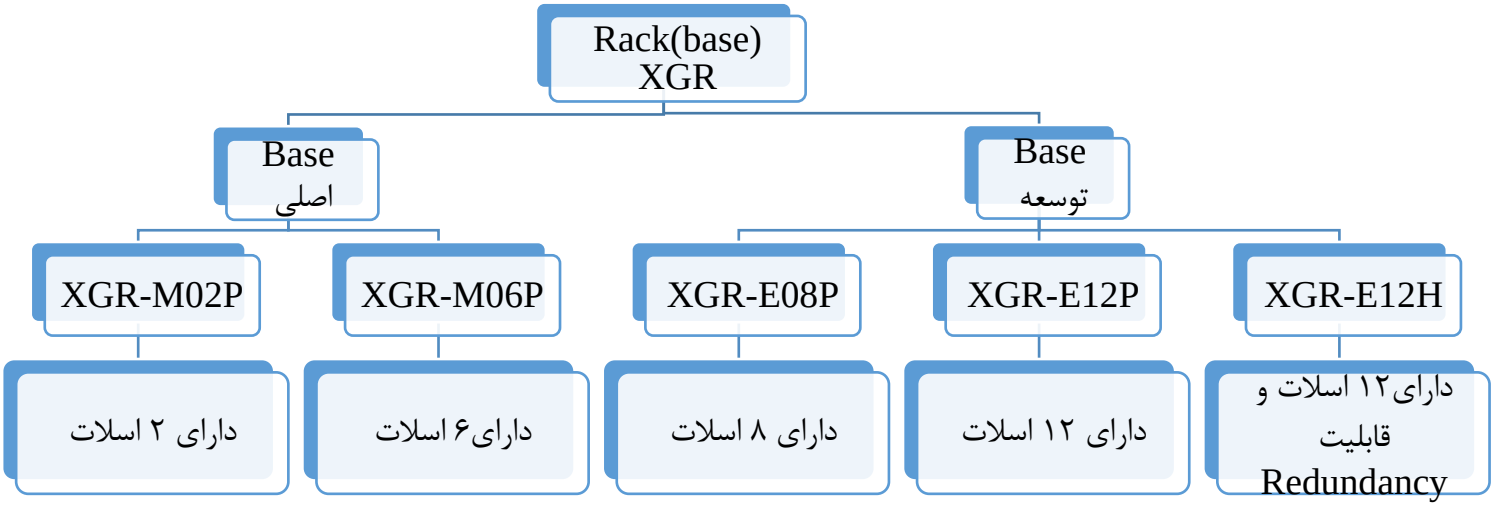

ماژول شبکه Profibus-DP XGI/XGK
XGL-PMEB
XGL-PSRA
XGL-PSEA
Profibus-DP, master
Profibus-DP,slave, Remote interface
Profibus-DP,slave

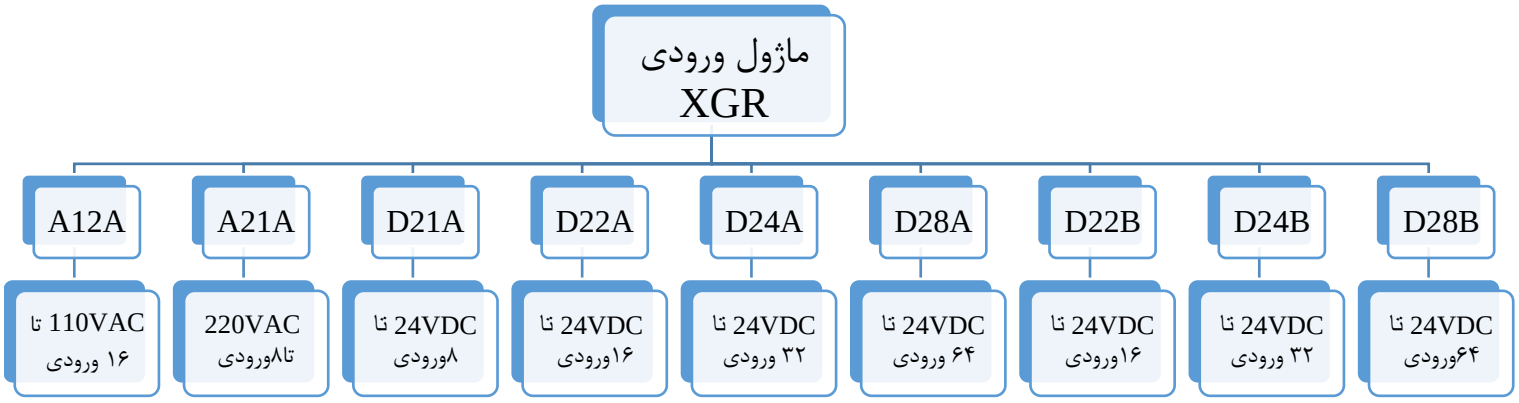

ماژول شبکه XGI/XGK
RAPIEnet V1
EtherNet/IP
BACnet/IP
Computer Link(Cnet)
XGL-EIMT
XGL-EIMF
XGL-EIMH
XGL-EIPT
XGL-BIPT
XGL-CH2B
XGL-C22B
XGL-C42B
زوج بهم تابیده دو کاناله
فیبر نوری دو کاناله
فیبر نوری و زوج بهم تابیده
اترنت صنعتی ۲کاناله
BACnet client/server
یک کانال RS-232C و یک کانال RS-485/422
دو کانال RS-232C
دو کانال RS422/485

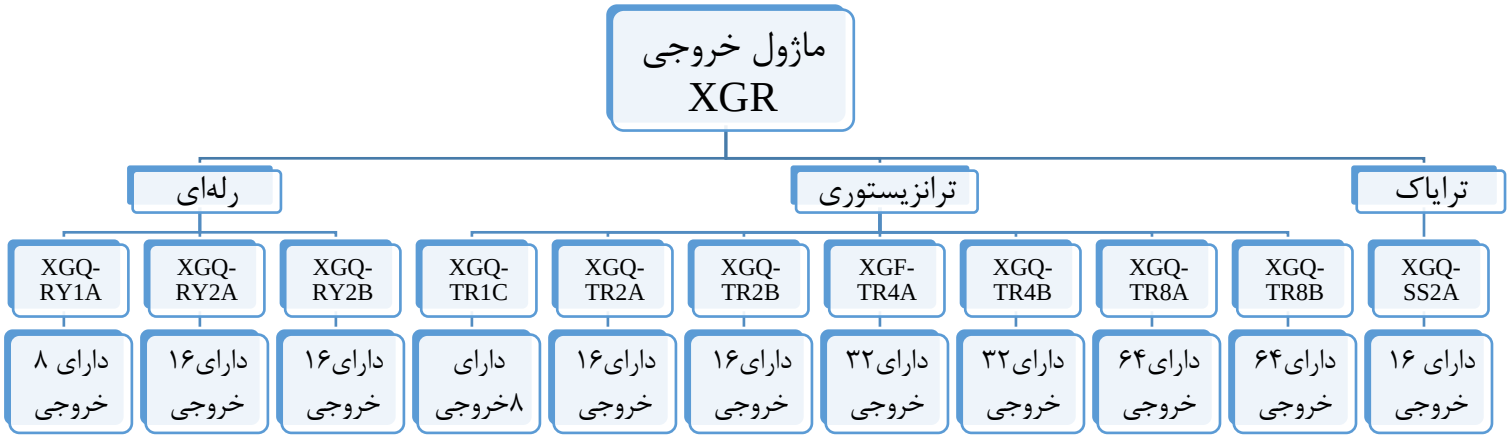

ماژول شبکه
XGI/XGK
DeviceNet(Dnet)
Rnet
Fnet
XGL-DMEB
XGL-RMEB
GOL-RP8T
XGL-FMEA
DeviceNet master
Rnet master,TP
Rnet stand alone repeater hub
Fnet master

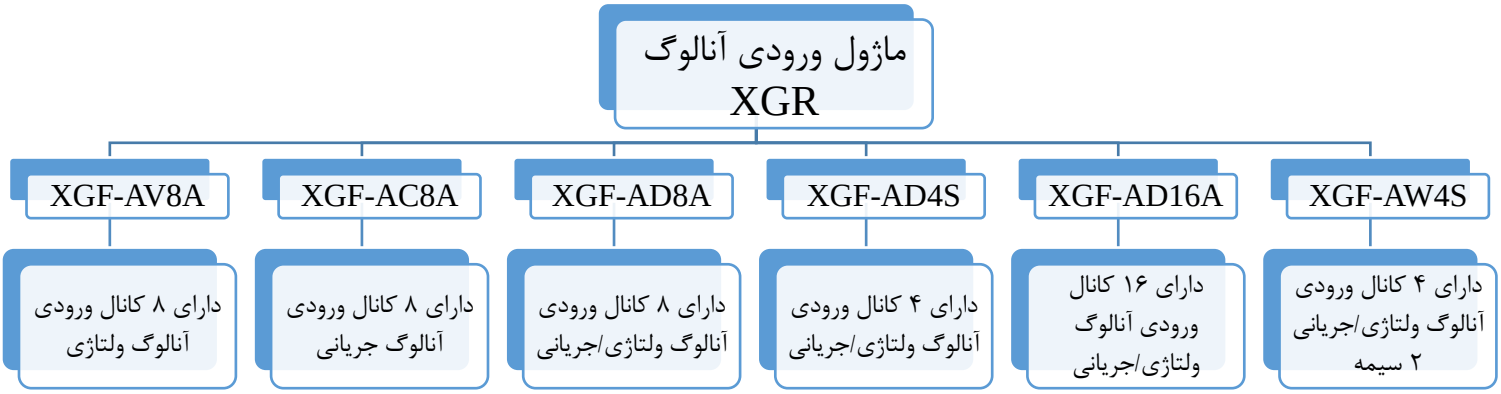
کابل توسعه بین دو RACK یا BASE در XGI/XGK
XGC-E041
XGC-E061
XGC-E121
XGC-E301
XGC-E501
XGC-E102
XGC-E152
کابل توسعه ۴۰ سانتیمتری
کابل توسعه ۶۰ سانتیمتری
کابل توسعه ۱۲۰ سانتیمتری
کابل توسعه ۳ متری
کابل توسعه ۵متری
کابل توسعه ۱۰ متری
کابل توسعه ۱۵ متری

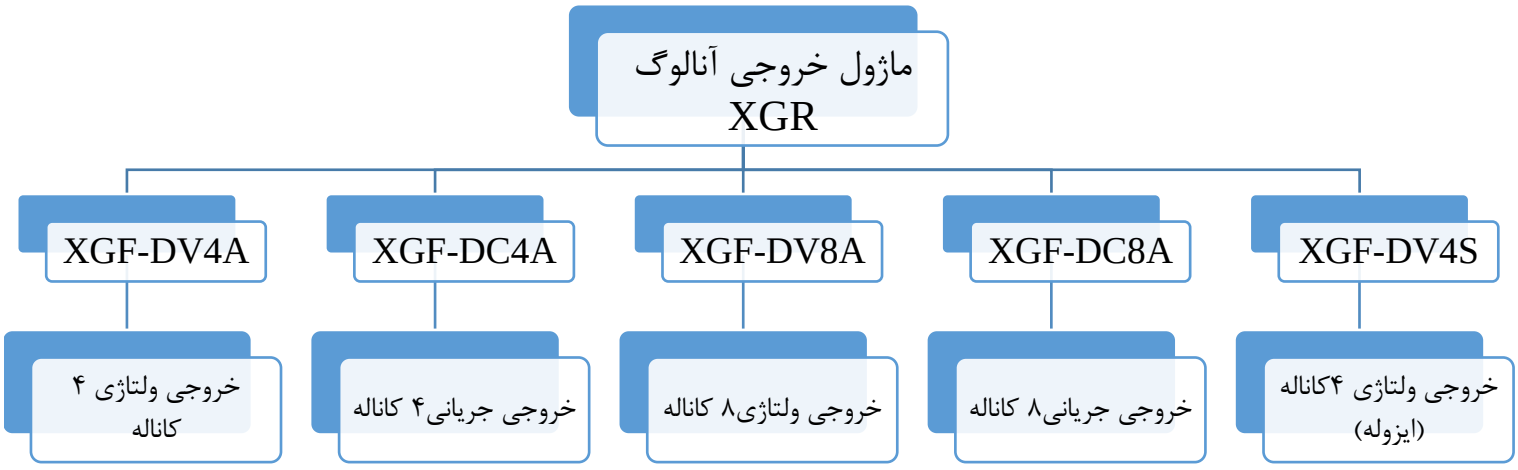

XGR
CPU
POWER (منبع تغذیه)
XGR-CPUH/T
XGR-CPUH/H
XGR-CPUH/S
XGR-AC12
XGR-AC13
XGR-AC22
XGR-AC23
XGR-DC42
انتقال داده زوج بهم تابیده
انتقال داده فیبر نوری تا ۲کیلومتر
انتقال داده فیبر نوری تا ۱۵کیلومتر
۱۱۰ولت ۵,۵ آمپر Base اصلی
۱۱۰ولت ۸,۵ آمپر Base توسعه
۲۲۰ولت ۵,۵ آمپر Base اصلی
۲۲۰ولت ۸,۵آمپر Base توسعه
۵ولت/۲۴ولت ۷ آمپر DC

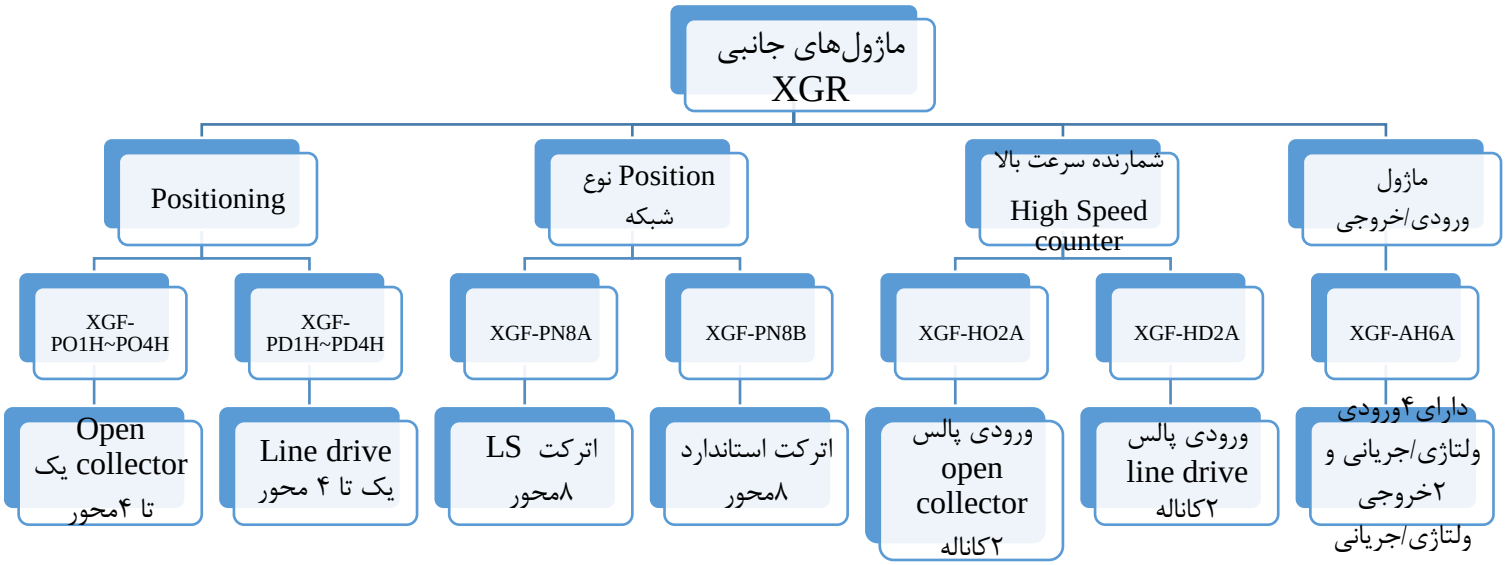

Rack(base) XGR
Base اصلی
Base توسعه
XGR-M02P
XGR-M06P
XGR-E08P
XGR-E12P
XGR-E12H
دارای ۲ اسلات
دارای۶ اسلات
دارای ۸ اسلات
دارای ۱۲ اسلات
دارای۱۲ اسلات و قابلیت Redundancy

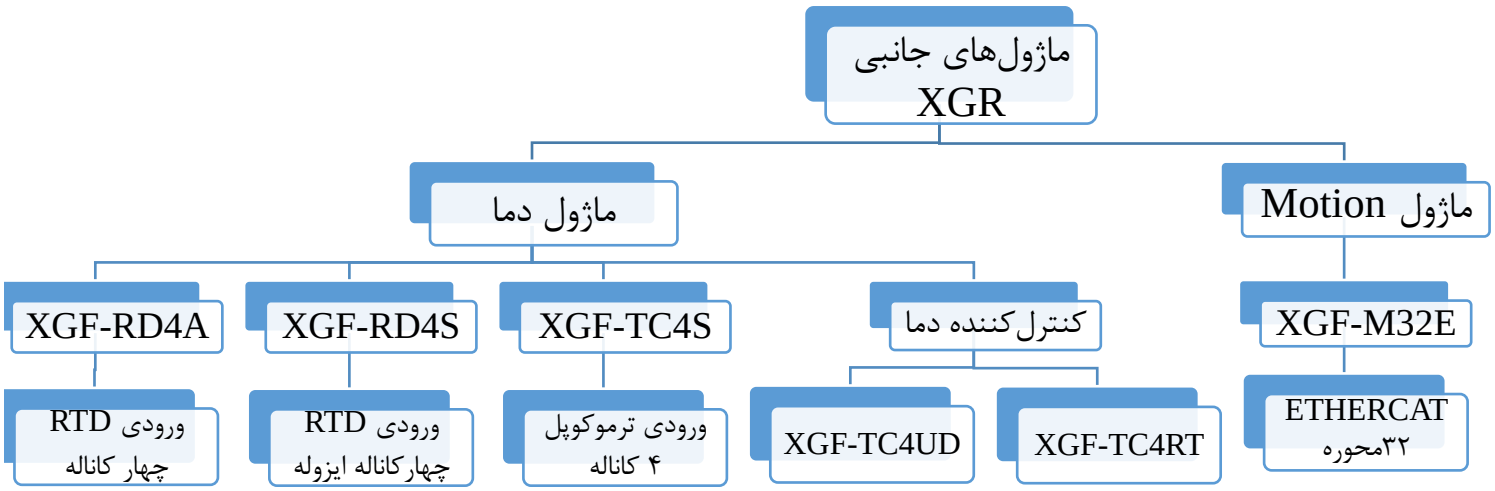

ماژول ورودی XGR
A12A
A21A
D21A
D22A
D24A
D28A
D22B
D24B
D28B
110VAC تا ۱۶ ورودی
220VAC تا۸ورودی
24VDC تا ۸ورودی
24VDC تا ۱۶ورودی
24VDC تا ۳۲ ورودی
24VDC تا ۶۴ ورودی
24VDC تا ۱۶ورودی
24VDC تا ۳۲ ورودی
24VDC تا ۶۴ورودی

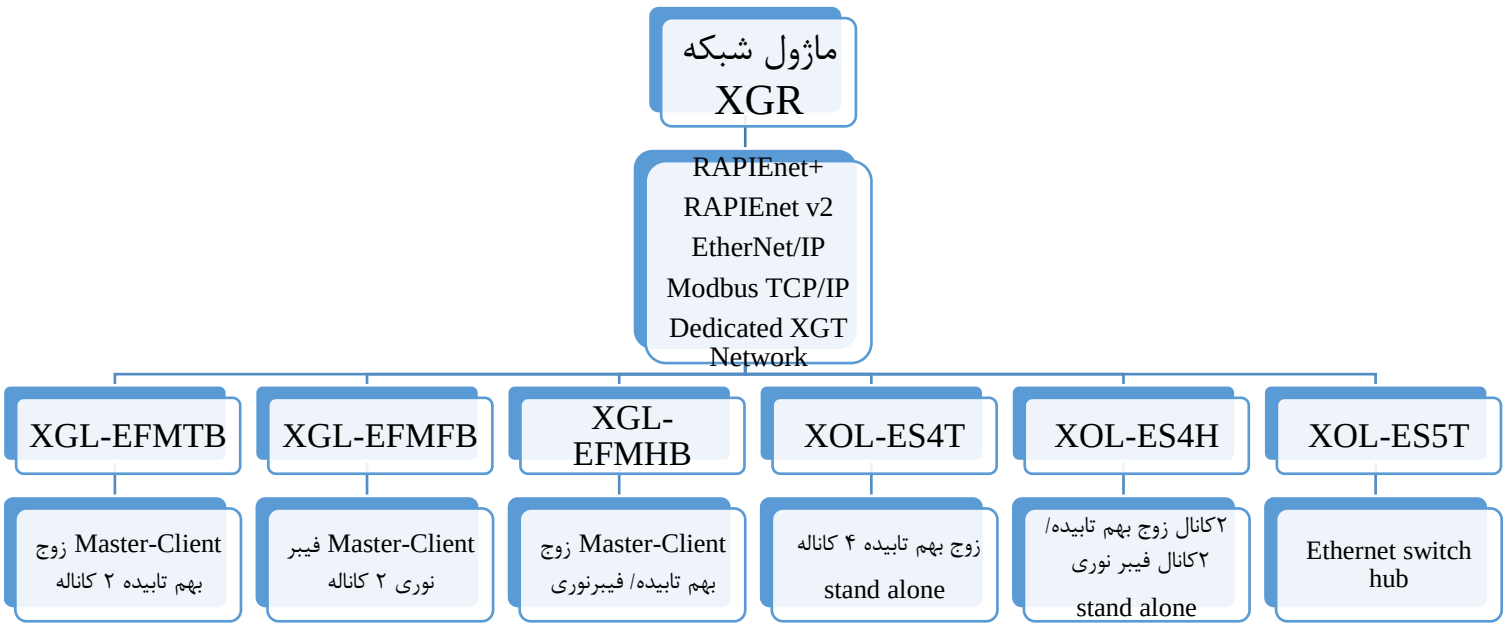

ماژول خروجی
XGR
رله‌ای
ترانزیستوری
ترایاک
XGQ-RY1A
XGQ-RY2A
XGQ-RY2B
XGQ-TR1C
XGQ-TR2A
XGQ-TR2B
XGF-TR4A
XGQ-TR4B
XGQ-TR8A
XGQ-TR8B
XGQ-SS2A
دارای ۸ خروجی
دارای۱۶ خروجی
دارای۱۶ خروجی
دارای ۸خروجی
دارای۱۶ خروجی
دارای۱۶ خروجی
دارای۳۲ خروجی
دارای۳۲ خروجی
دارای۶۴ خروجی
دارای۶۴ خروجی
دارای ۱۶ خروجی

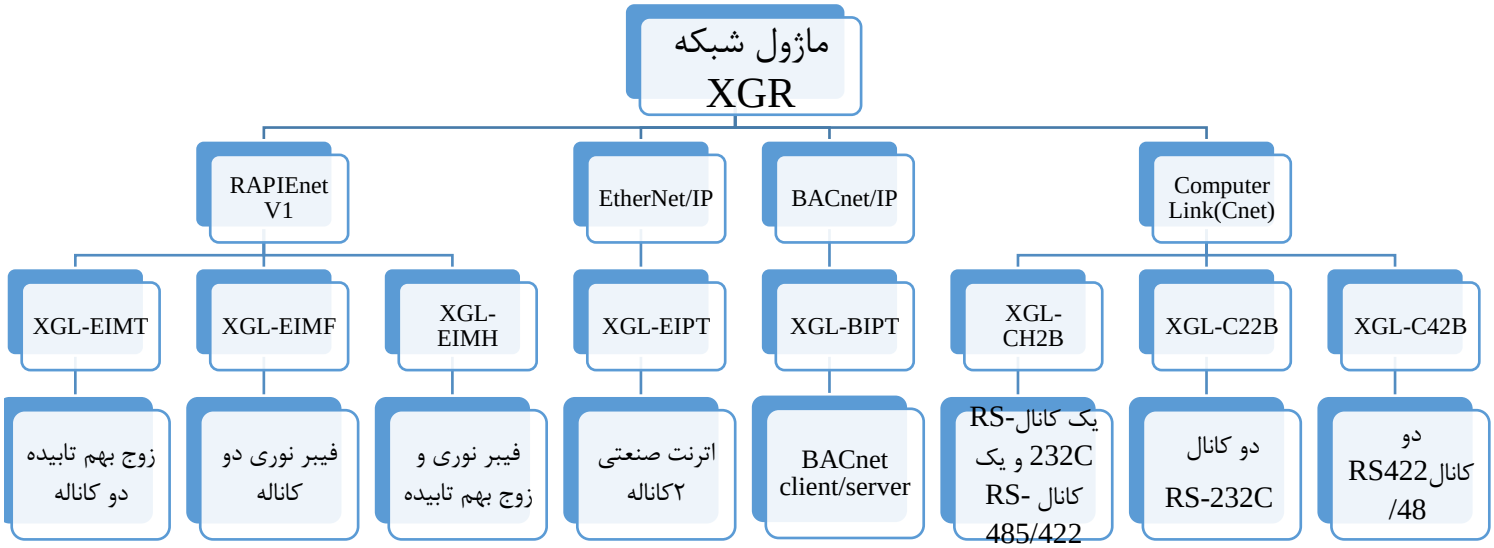

ماژول ورودی آنالوگ
XGR
XGF-AV8A
XGF-AC8A
XGF-AD8A
XGF-AD4S
XGF-AD16A
XGF-AW4S
دارای ۸ کانال ورودی آنالوگ ولتاژی
دارای ۸ کانال ورودی آنالوگ جریانی
دارای ۸ کانال ورودی آنالوگ ولتاژی/جریانی
دارای ۴ کانال ورودی آنالوگ ولتاژی/جریانی
دارای ۱۶ کانال ورودی آنالوگ ولتاژی/جریانی
دارای ۴ کانال ورودی آنالوگ ولتاژی/جریانی ۲ سیمه

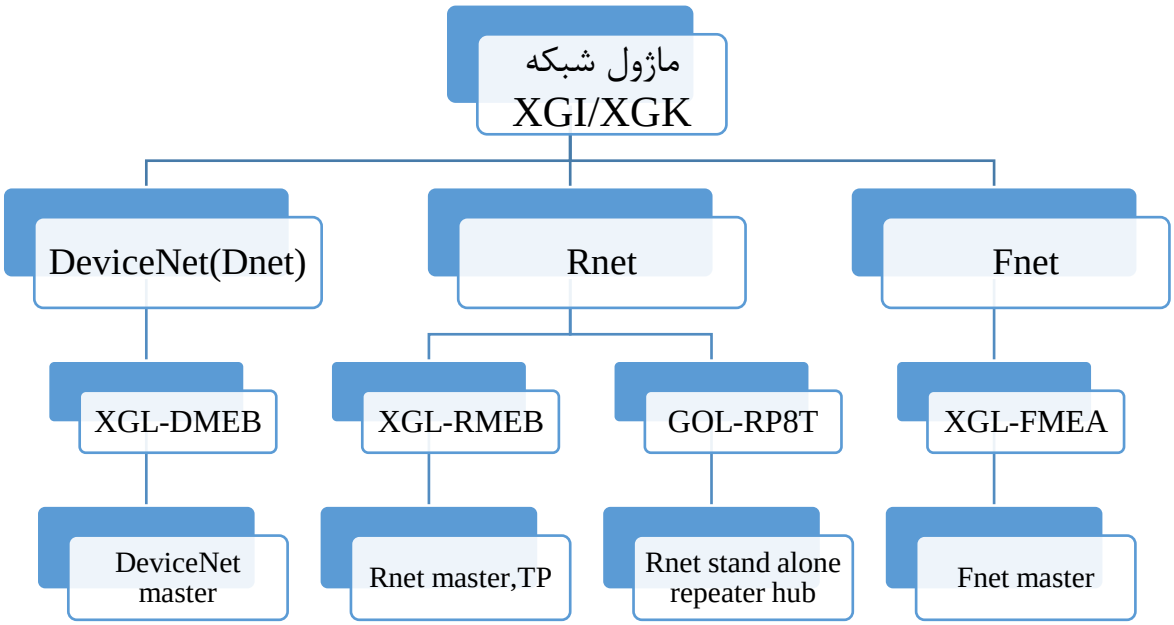

ماژول خروجی آنالوگ
XGR
XGF-DV4A
XGF-DC4A
XGF-DV8A
XGF-DC8A
XGF-DV4S
خروجی ولتاژی ۴ کاناله
خروجی جریانی۴ کاناله
خروجی ولتاژی۸ کاناله
خروجی جریانی۸ کاناله
خروجی ولتاژی ۴کاناله (ایزوله)

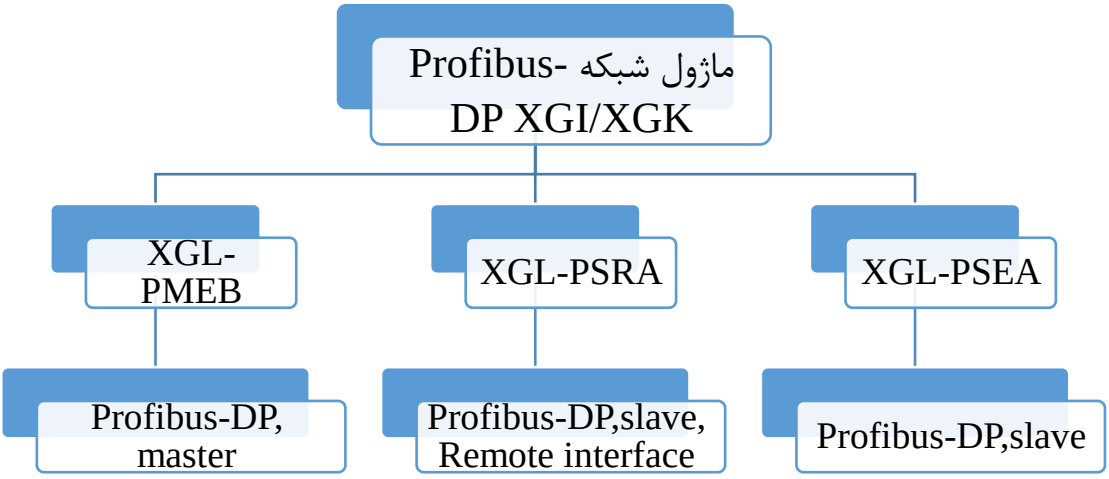

ماژول‌های جانبی XGR
Positioning
نوع Position شبکه
شمارنده سرعت بالا High Speed counter
ماژول ورودی/خروجی
XGF-PO1H~PO4H
XGF-PD1H~PD4H
XGF-PN8A
XGF-PN8B
XGF-HO2A
XGF-HD2A
XGF-AH6A
Open collector یک تا ۴محور
Line drive یک تا ۴ محور
LS اترکت ۸محور
اترکت استاندارد ۸محور
ورودی پالس open collector ۲کاناله
ورودی پالس line drive ۲کاناله
دارای۴ورودی ولتاژی/جریانی و ۲خروجی ولتاژی/جریانی


ماژول‌های جانبی XGR
ماژول دما
ماژول Motion
XGF-RD4A
XGF-RD4S
XGF-TC4S
کنترل‌کننده دما
XGF-M32E
RTD ورودی چهار کاناله
RTD ورودی چهارکاناله ایزوله
ورودی ترموکوپل ۴ کاناله
XGF-TC4UD
XGF-TC4RT
ETHERCAT ۳۲محوره


ماژول شبکه XGR
RAPIEnet+
RAPIEnet v2
EtherNet/IP
Modbus TCP/IP
Dedicated XGT Network
XGL-EFMTB
XGL-EFMFB
XGL-EFMHB
XOL-ES4T
XOL-ES4H
XOL-ES5T
Master-Client زوج بهم تابیده ۲ کاناله
Master-Client فیبر نوری ۲ کاناله
Master-Client زوج بهم تابیده/ فیبرنوری
زوج بهم تابیده ۴ کاناله stand alone
۲کانال زوج بهم تابیده/ ۲کانال فیبر نوری stand alone
Ethernet switch hub


ماژول شبکه
XGR
RAPIEnet V1
EtherNet/IP
BACnet/IP
Computer Link(Cnet)
XGL-EIMT
XGL-EIMF
XGL-EIMH
XGL-EIPT
XGL-BIPT
XGL-CH2B
XGL-C22B
XGL-C42B
زوج بهم تابیده دو کاناله
فیبر نوری دو کاناله
فیبر نوری و زوج بهم تابیده
اترنت صنعتی ۲کاناله
BACnet client/server
یک کانال RS-232C و یک کانال RS-485/422
دو کانال RS-232C
دو کانال RS422 /48

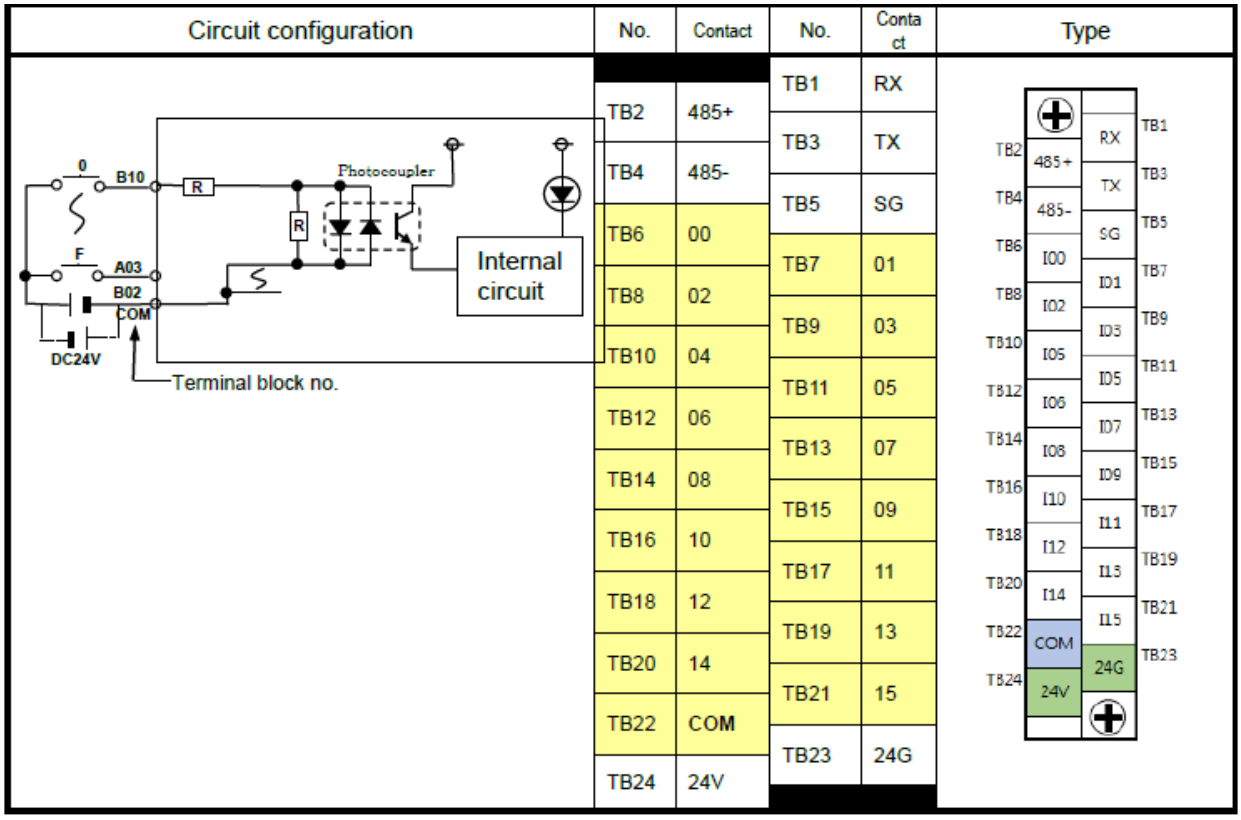

ماژول شبکه
XGI/XGK
DeviceNet(Dnet)
Rnet
Fnet
XGL-DMEB
XGL-RMEB
GOL-RP8T
XGL-FMEA
DeviceNet master
Rnet master,TP
Rnet stand alone repeater hub
Fnet master


ماژول شبکه Profibus-
DP XGI/XGK
XGL-PMEB
XGL-PSRA
XGL-PSEA
Profibus-DP, master
Profibus-DP,slave, Remote interface
Profibus-DP,slave

پیوست۲

روش اتصال ورودی/ خروجی‌های PLCهای سری Compact

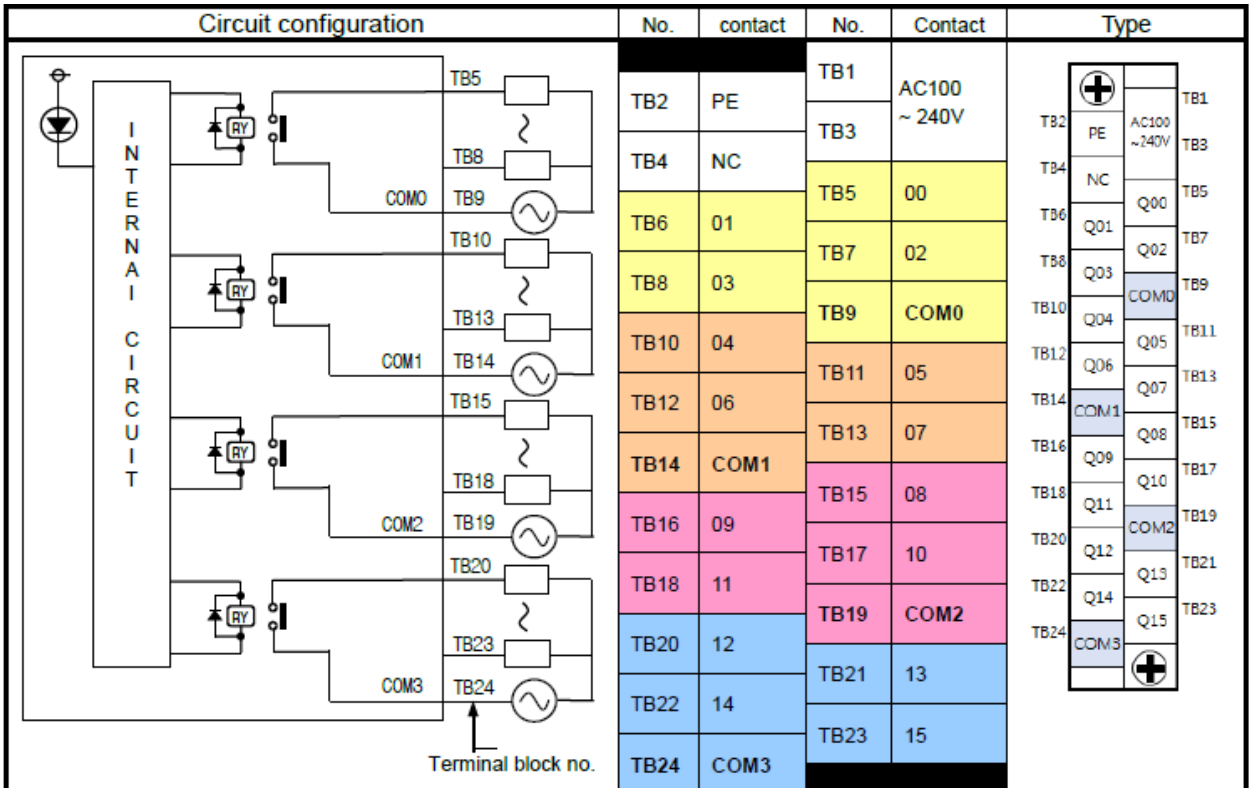
ورودی PLC سری XEC-D...32H



No.	Contact	No.	Contact
TB2	485+	TB1	RX
TB4	485-	TB3	TX
TB6	00	TB5	SG
TB8	02	TB7	01
TB10	04	TB9	03
TB12	06	TB11	05
TB14	08	TB13	07
TB16	10	TB15	09
TB18	12	TB17	11
TB20	14	TB19	13
TB22	COM	TB21	15
TB24	24V	TB23	24G

Type

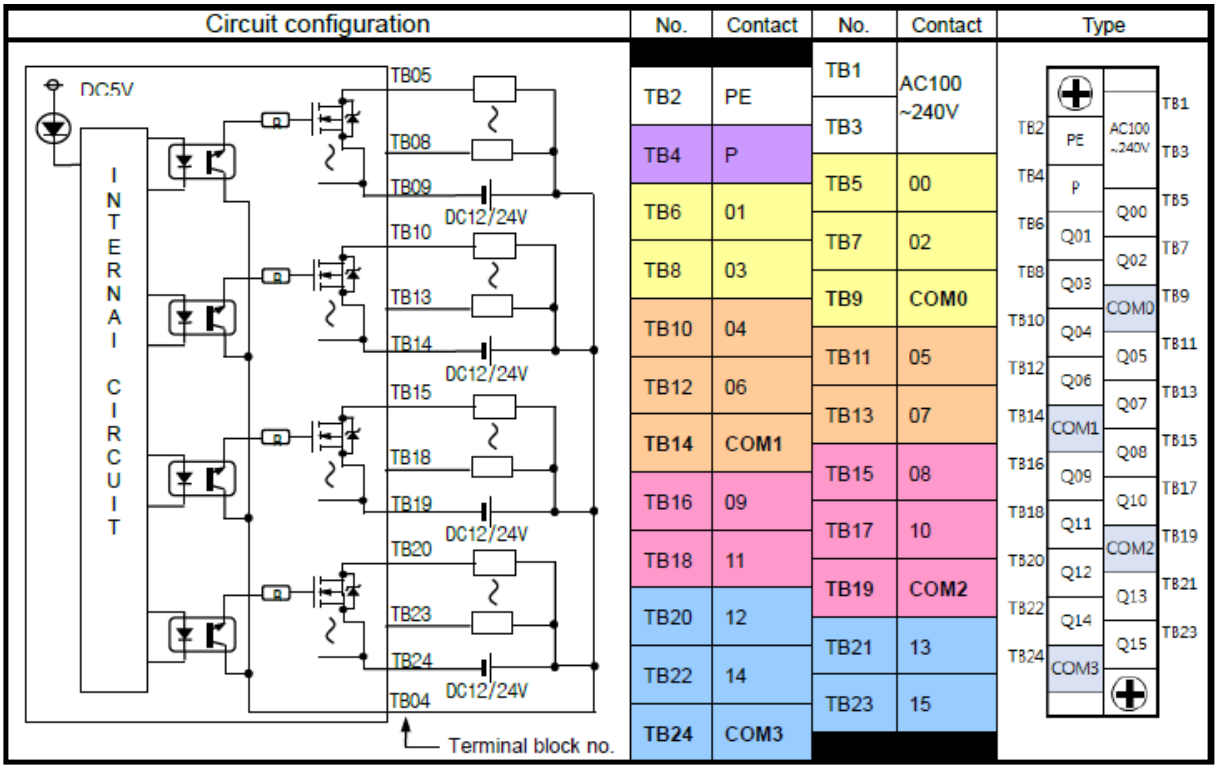
خروجی PLC سری XEC-DR32H



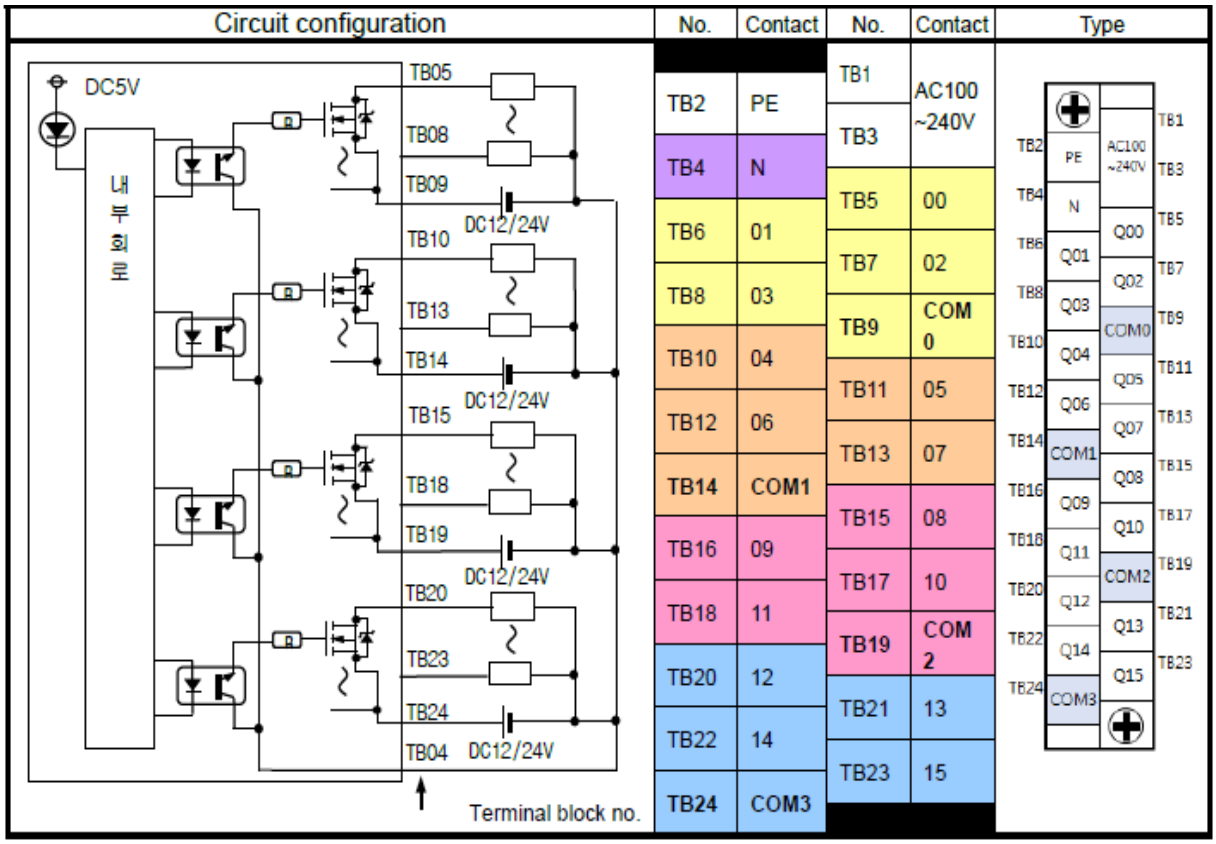
No.	contact	No.	Contact
TB2	PE	TB1	AC100 ~ 240V
TB4	NC	TB3	
TB6	01	TB5	00
TB8	03	TB7	02
TB10	04	TB9	COM0
TB12	06	TB11	05
TB14	COM1	TB13	07
TB16	09	TB15	08
TB18	11	TB17	10
TB20	12	TB19	COM2
TB22	14	TB21	13
TB24	COM3	TB23	15

Type

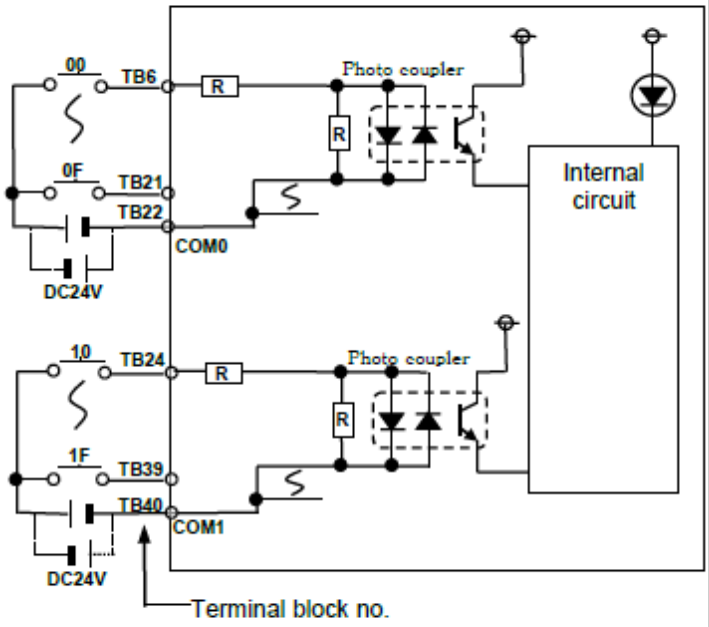
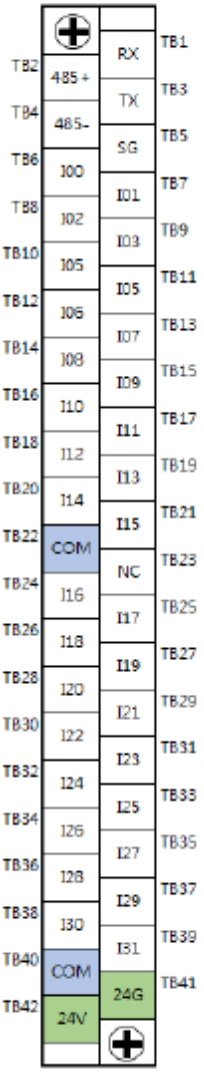
خروجیPLC سری XEC-DN32H



خروجیPLC سری XEC-DP32H

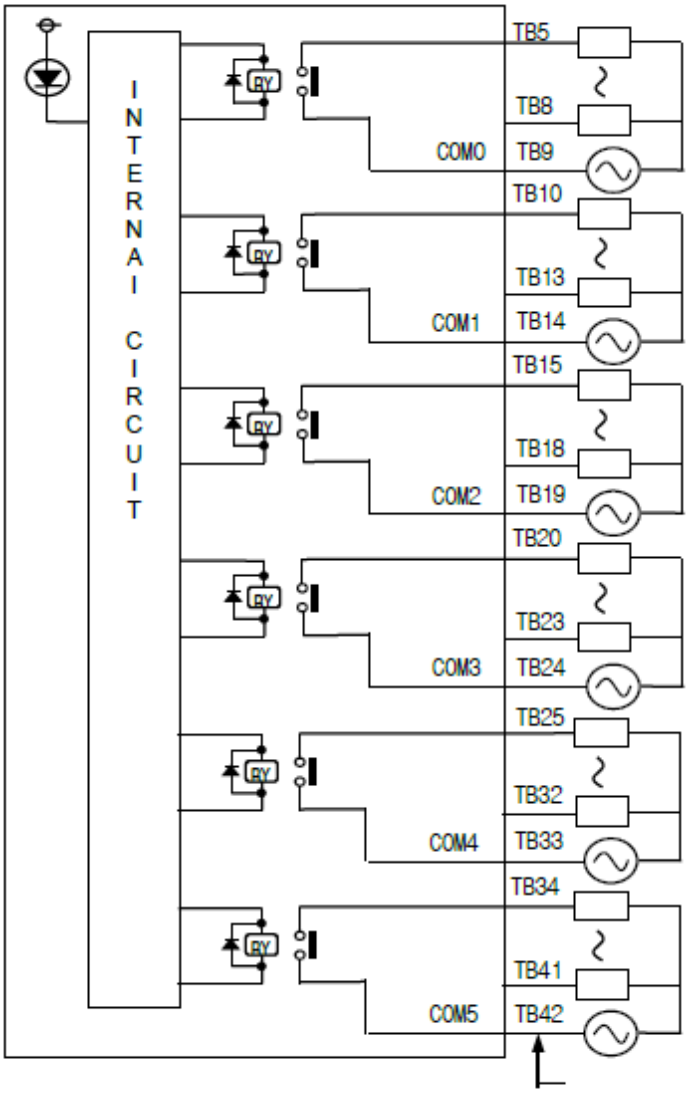


ورودیPLC سری XEC-D...32H

Circuit configuration	No.	contact	No.	contact	type
			TB1	RX	
	TB2	485+	TB3	TX	
	TB4	485-	TB5	SG	
	TB6	00	TB7	01	
	TB8	02	TB9	03	
	TB10	04	TB11	05	
	TB12	06	TB13	07	
	TB14	08	TB15	09	
	TB16	10	TB17	11	
	TB18	12	TB19	13	
	TB20	14	TB21	15	
	TB22	COM	TB23	NC	
	TB24	16	TB25	17	
	TB26	18	TB27	19	
	TB28	20	TB29	21	
	TB30	22	TB31	23	
	TB32	24	TB33	25	
	TB34	26	TB35	27	
	TB36	28	TB37	29	
	TB38	30	TB39	31	
	TB40	COM	TB41	24G	
	TB42	24V			

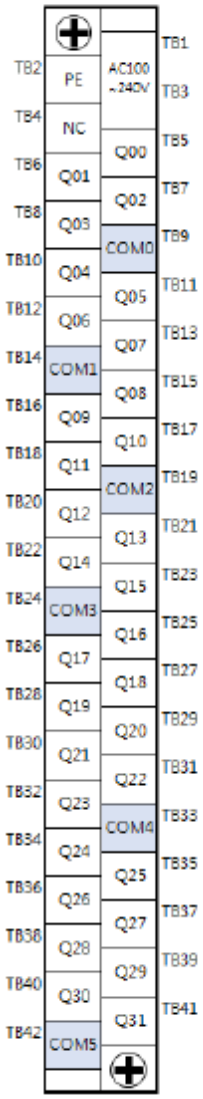
خروجی PLC سری XEC-DR64H

Circuit configuration



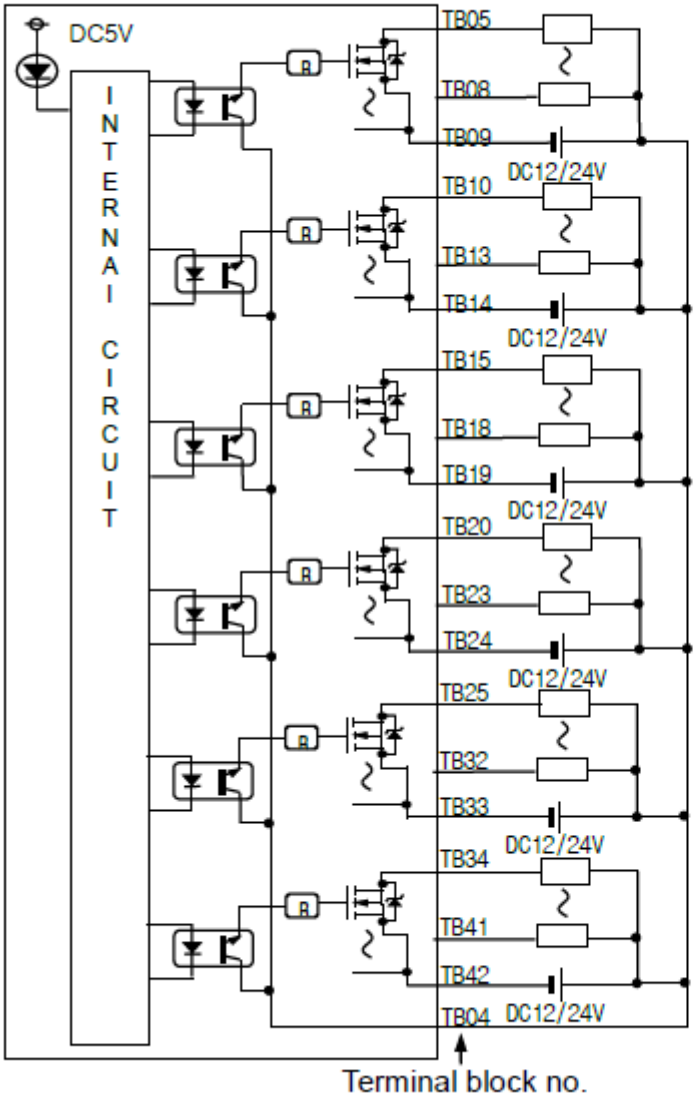
No.	Contact	No.	Contact
TB2	PE	TB1	AC100 ~240V
TB4	NC	TB3	
TB6	01	TB5	00
TB8	03	TB7	02
TB10	04	TB9	COM0
TB12	06	TB11	05
TB14	COM1	TB13	07
TB16	09	TB15	08
TB18	11	TB17	10
TB20	12	TB19	COM2
TB22	14	TB21	13
TB24	COM3	TB23	15
TB26	17	TB25	16
TB28	19	TB27	18
TB30	21	TB29	20
TB32	23	TB31	22
TB34	24	TB33	COM4
TB36	26	TB35	25
TB38	28	TB37	27
TB40	30	TB39	29
TB42	COM5	TB41	31

type



خروجی PLC سری XEC-DN64H

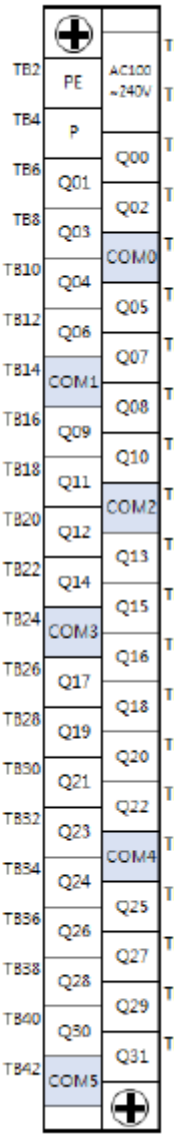
Circuit configuration



No.	contact
TB2	PE
TB4	P
TB6	01
TB8	03
TB10	04
TB12	06
TB14	**COM1**
TB16	09
TB18	11
TB20	12
TB22	14
TB24	**COM3**
TB26	17
TB28	19
TB30	21
TB32	23
TB34	24
TB36	26
TB38	28
TB40	30
TB42	**COM5**

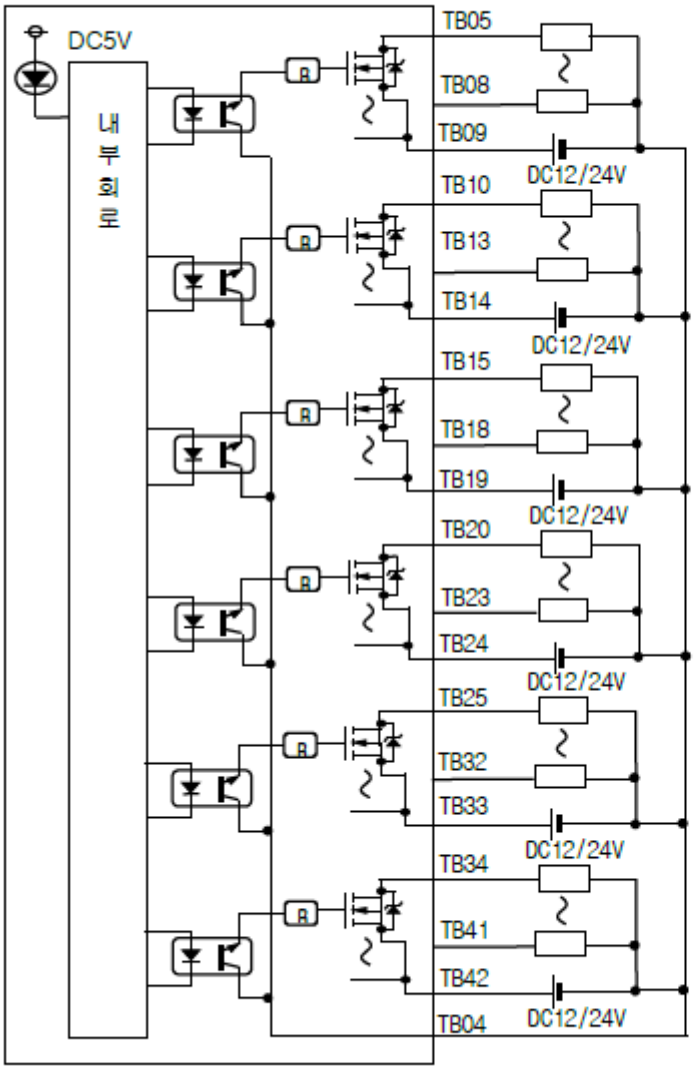
No.	contact
TB1	AC100 ~240V
TB3	
TB5	00
TB7	02
TB9	**COM0**
TB11	05
TB13	07
TB15	08
TB17	10
TB19	**COM2**
TB21	13
TB23	15
TB25	16
TB27	18
TB29	20
TB31	22
TB33	**COM4**
TB35	25
TB37	27
TB39	29
TB41	31

Type



خروجی PLC سری XEC-DP64H

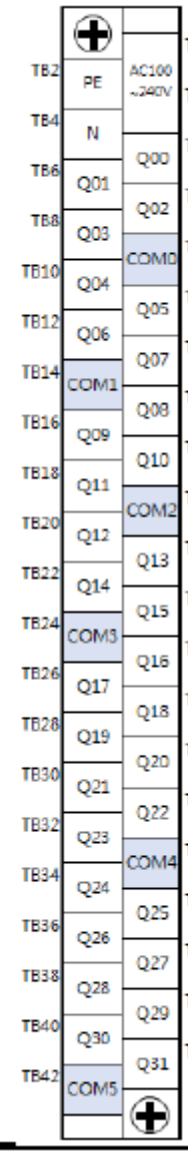
Circuit configuration



No.	contact
TB2	PE
TB4	N
TB6	01
TB8	03
TB10	04
TB12	06
TB14	COM1
TB16	09
TB18	11
TB20	12
TB22	14
TB24	COM3
TB26	17
TB28	19
TB30	21
TB32	23
TB34	24
TB36	26
TB38	28
TB40	30
TB42	COM5

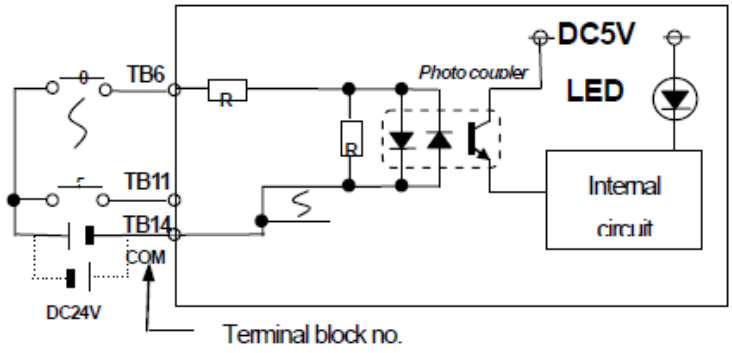
No.	contact
TB1	AC100 ~240V
TB3	
TB5	00
TB7	02
TB9	COM 0
TB11	05
TB13	07
TB15	08
TB17	10
TB19	COM 2
TB21	13
TB23	15
TB25	16
TB27	18
TB29	20
TB31	22
TB33	COM 4
TB35	25
TB37	27
TB39	29
TB41	31

Type



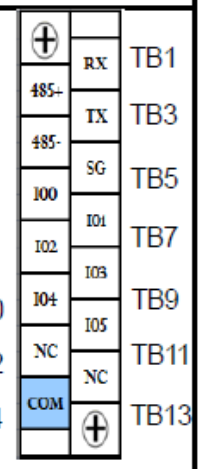
ورودی PLC سری XEC-D...10E

Circuit configuration



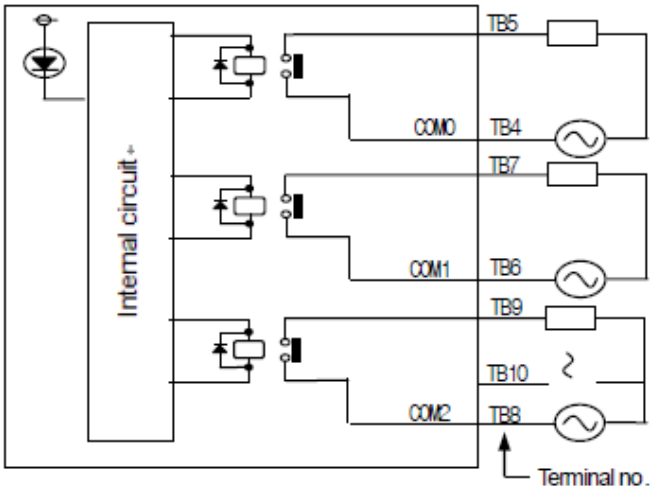
No.	Contact	No.	Contact
TB2	485+	TB1	RX
TB4	485-	TB3	TX
TB6	I00	TB5	SG
TB8	I02	TB7	I01
TB10	I04	TB9	I03
TB12	NC	TB11	I05
TB14	COM	TB13	NC

Type



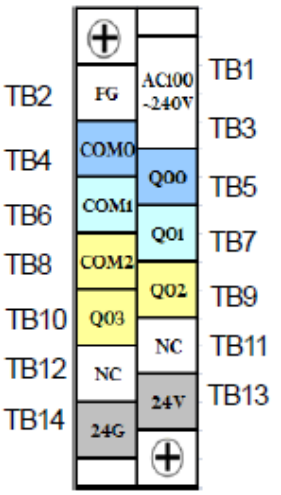
خروجی PLC سری XEC-DR10E

Circuit configuration



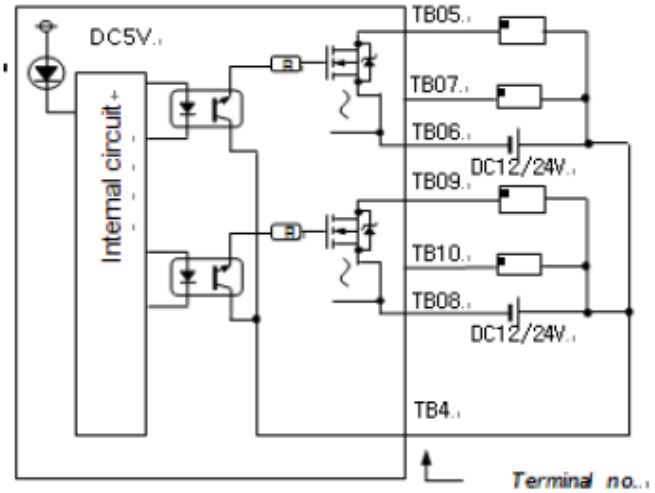
No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	COM0	TB3	
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	NC	TB11	NC
TB14	24G	TB13	24V

Type



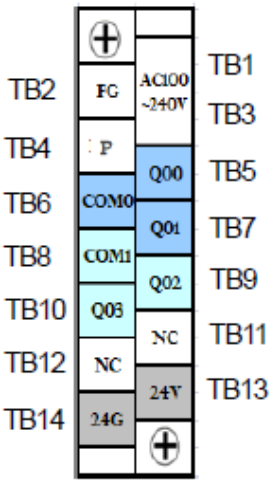
خروجی PLC سری XEC-DN10E

Circuit configuration



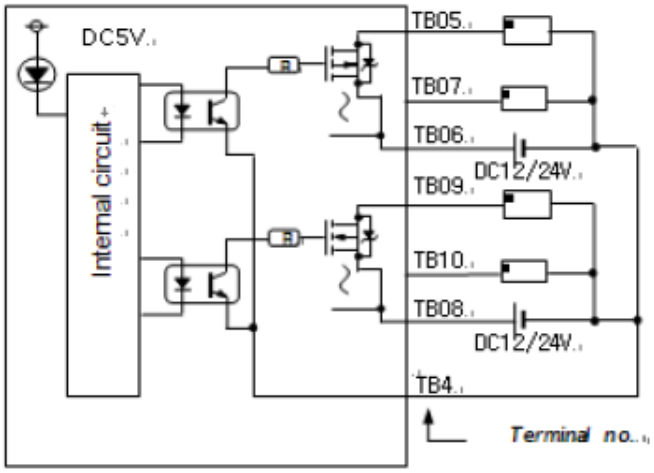
No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	P	TB3	
TB6	COM0	TB5	Q00
TB8	COM1	TB7	Q01
TB10	Q03	TB9	Q02
TB12	NC	TB11	NC
TB14	24G	TB13	24V

Type



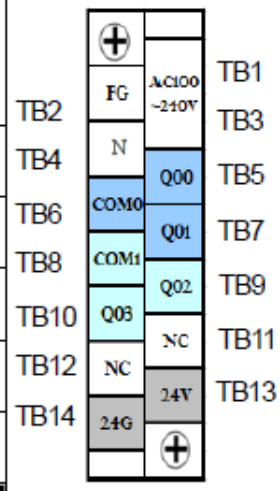
خروجی PLC سری XEC-DP10E

Circuit configuration



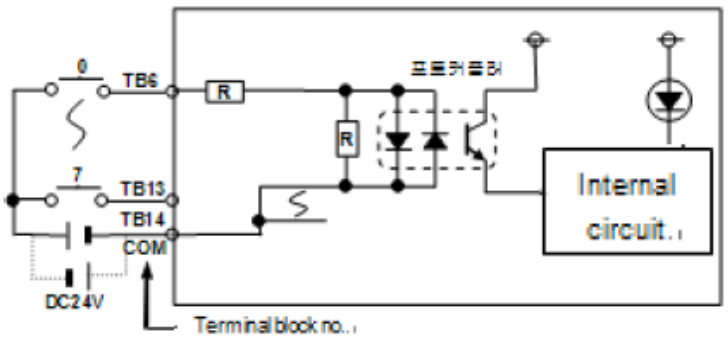
No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	N	TB3	
TB6	COM0	TB5	Q00
TB8	COM1	TB7	Q01
TB10	Q03	TB9	Q02
TB12	NC	TB11	NC
TB14	24G	TB13	24V

Type

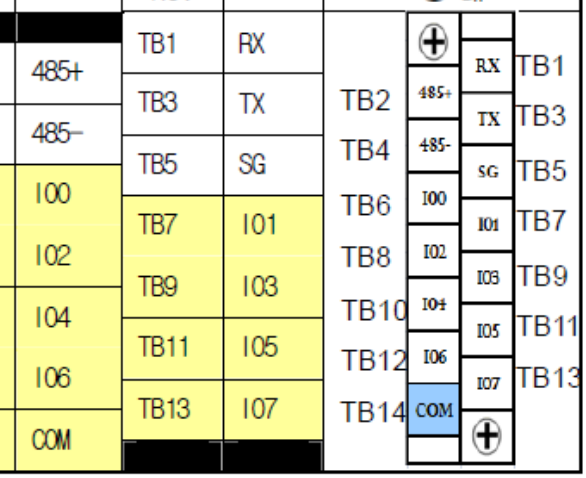


ورودی PLC سری XEC-D...14E

Circuit configuration

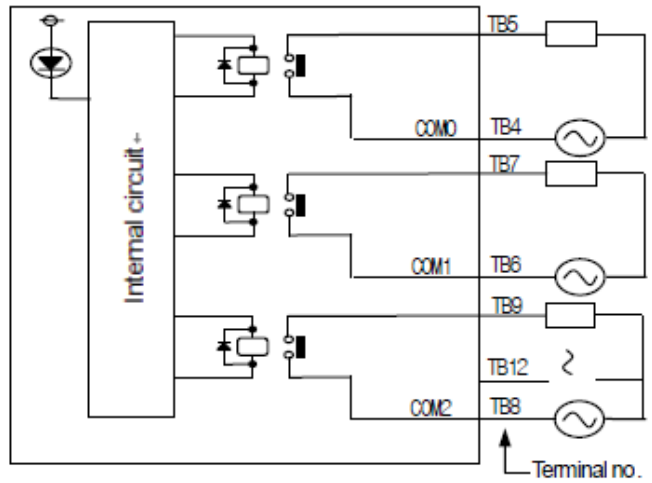


No.	Contact	No.	Contact
		TB1	RX
TB2	485+	TB3	TX
TB4	485-	TB5	SG
TB6	I00	TB7	I01
TB8	I02	TB9	I03
TB10	I04	TB11	I05
TB12	I06	TB13	I07
TB14	COM		

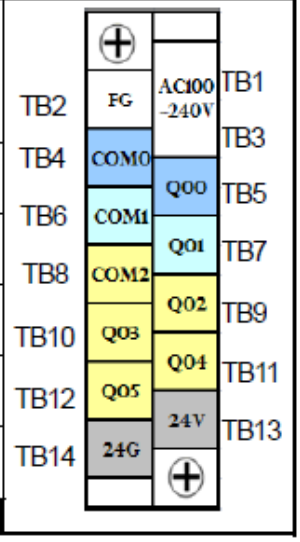


خروجی PLC سری XEC-DR14E

Circuit configuration

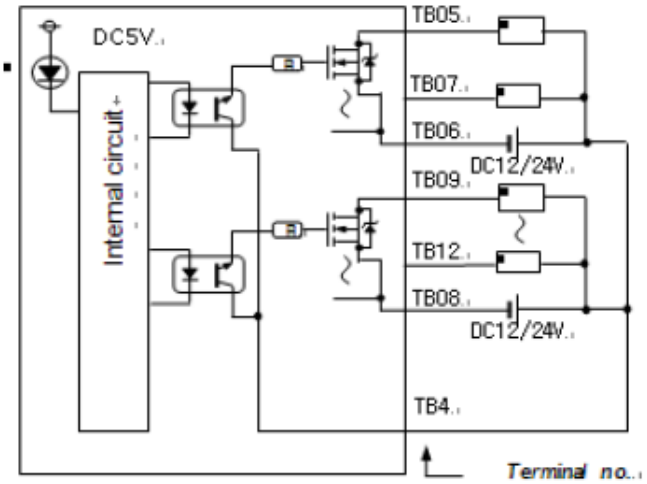


No.	Contact	No.	Contact
		TB1	AC100 ~240V
TB2	FG	TB3	
TB4	COM0	TB5	Q00
TB6	COM1	TB7	Q01
TB8	COM2	TB9	Q02
TB10	Q03	TB11	Q04
TB12	Q05	TB13	24V
TB14	24G		



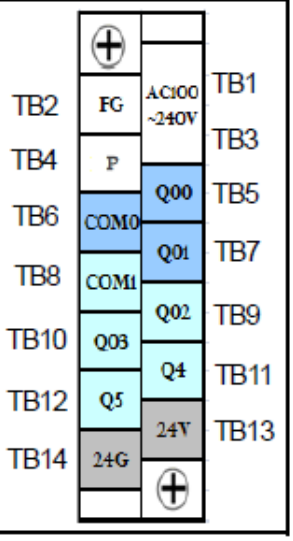
خروجیPLC سری XEC-DN14E

Circuit configuration



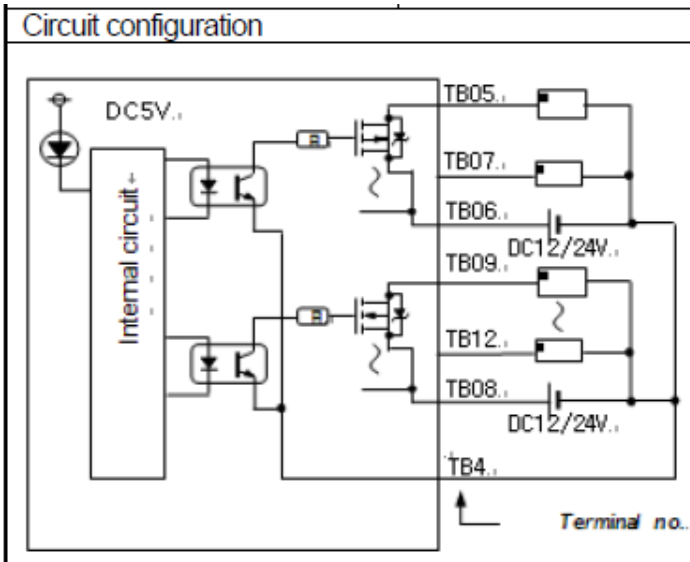
No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	P	TB3	
TB6	COM0	TB5	Q00
TB8	COM1	TB7	Q01
TB10	Q03	TB9	Q02
TB12	Q05	TB11	Q04
TB14	24G	TB13	24V

Type



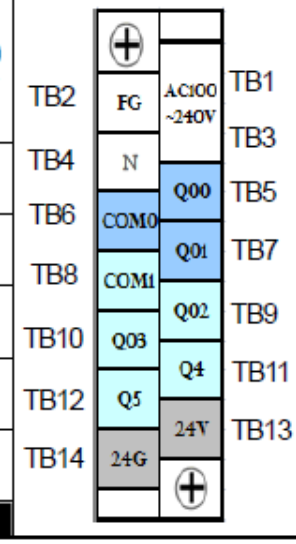
خروجیPLC سری XEC-DP14E

Circuit configuration



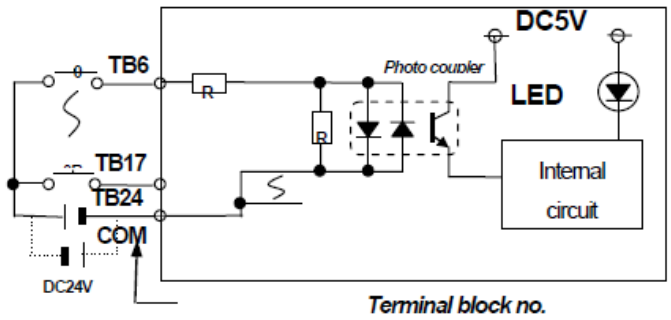
No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	N	TB3	
TB6	COM0	TB5	Q00
TB8	COM1	TB7	Q01
TB10	Q03	TB9	Q02
TB12	Q05	TB11	Q04
TB14	24G	TB13	24V

Type



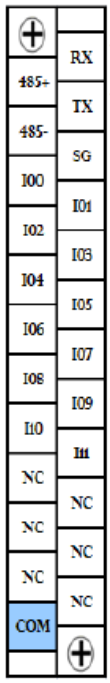
ورودی PLC سری XEC-D...20E

Circuit configuration



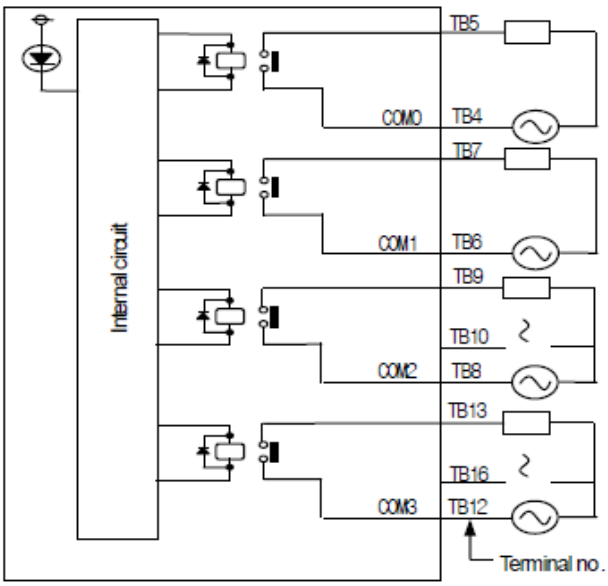
No.	Contact	No.	Contact
		TB1	RX
TB2	485+	TB3	TX
TB4	485-	TB5	SG
TB6	I00	TB7	I01
TB8	I02	TB9	I03
TB10	I04	TB11	I05
TB12	I06	TB13	I07
TB14	I08	TB15	I09
TB16	I10	TB17	I11
TB18	NC	TB19	NC
TB20	NC	TB21	NC
TB22	NC	TB23	NC
TB24	COM		

Type



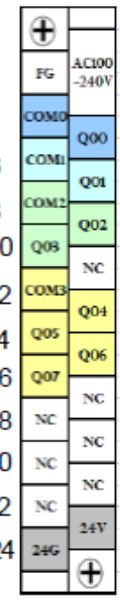
خروجی PLC سری XEC-DR20E

Circuit configuration



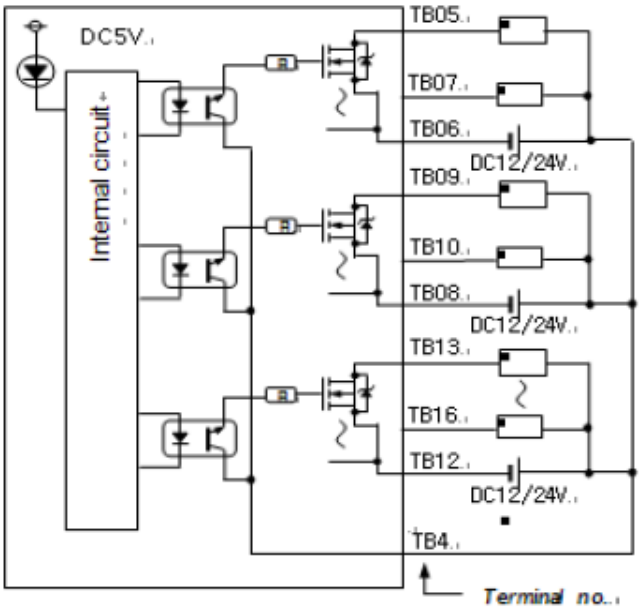
No.	Contact	No.	Contact
		TB1	AC100 ~240V
TB2	FG	TB3	
TB4	COM0	TB5	Q00
TB6	COM1	TB7	Q01
TB8	COM2	TB9	Q02
TB10	Q03	TB11	NC
TB12	COM3	TB13	Q04
TB14	Q05	TB15	Q06
TB16	Q07	TB17	NC
TB18	NC	TB19	NC
TB20	NC	TB21	NC
TB22	NC	TB23	24V
TB24	24G		

Type



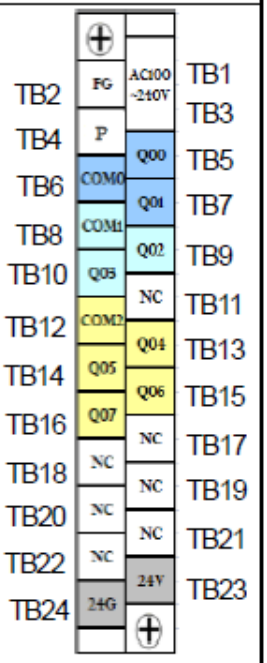
خروجیPLC سری XEC-DN20E

Circuit configuration



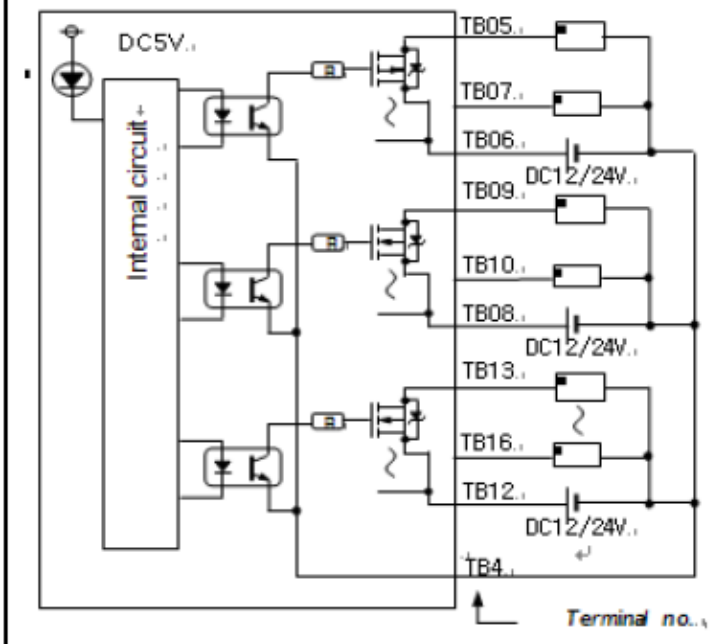
No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	P	TB3	
TB6	COM0	TB5	Q00
TB8	COM1	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM2	TB11	NC
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	NC	TB17	NC
TB20	NC	TB19	NC
TB22	NC	TB21	NC
TB24	24G	TB23	24V

Type



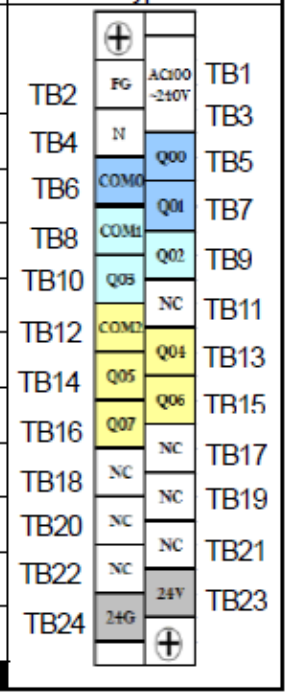
خروجیPLC سری XEC-DP20E

Circuit configuration

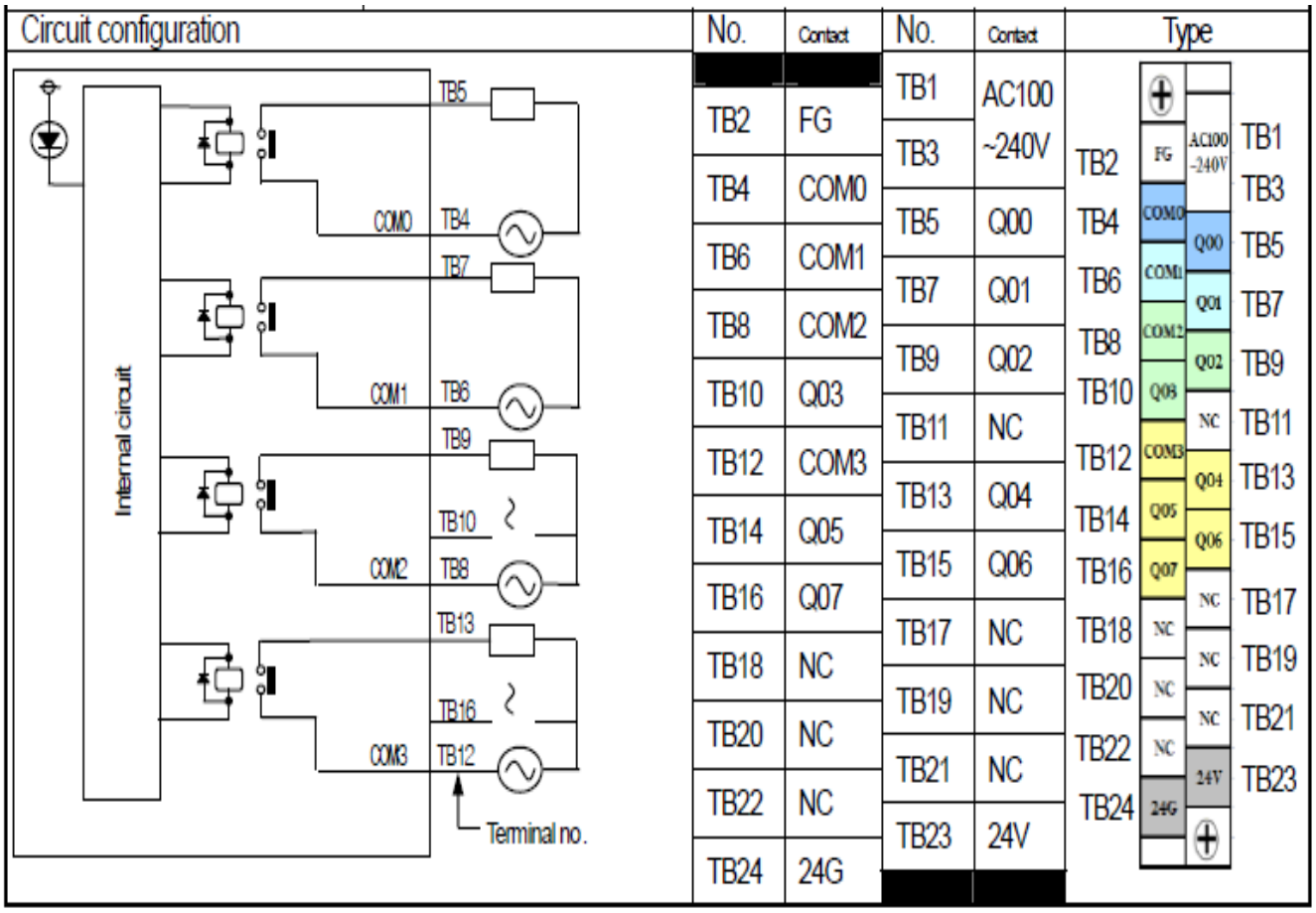


No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	N	TB3	
TB6	COM0	TB5	Q00
TB8	COM1	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM2	TB11	NC
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	NC	TB17	NC
TB20	NC	TB19	NC
TB22	NC	TB21	NC
TB24	24G	TB23	24V

Type

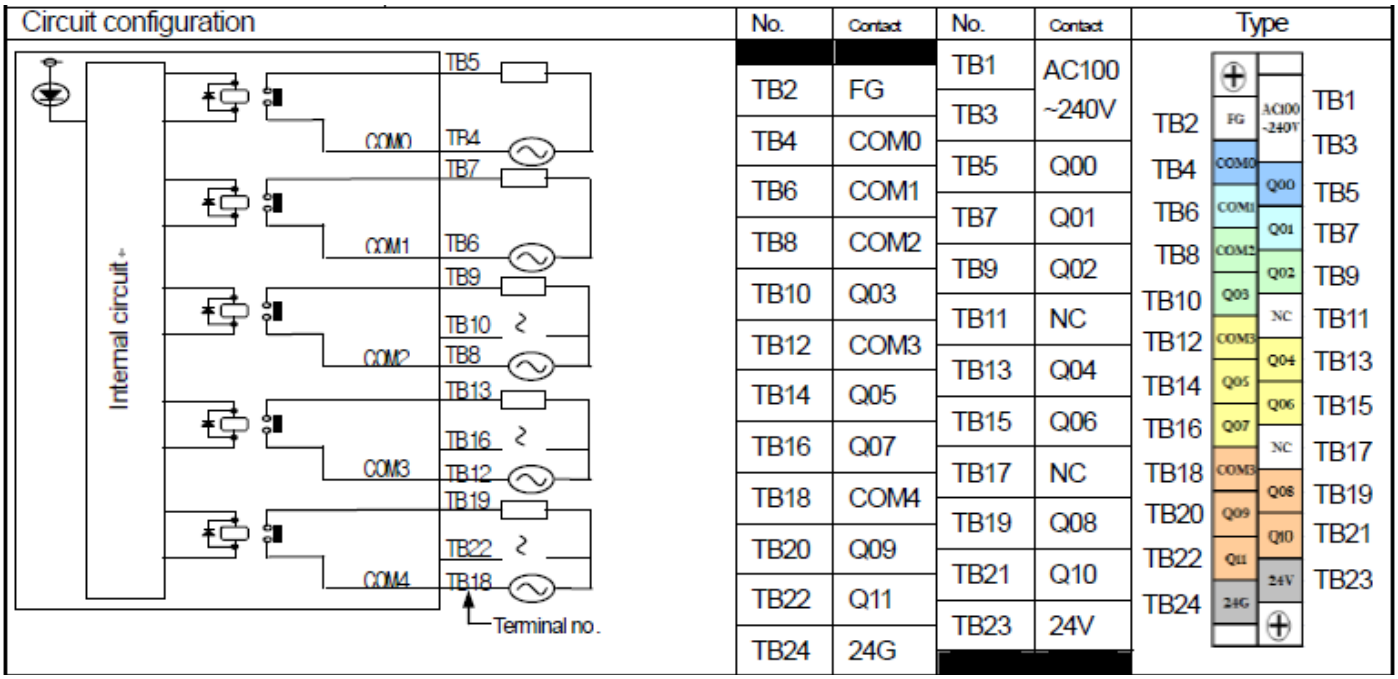


ورودی PLC سری XEC-D...30E



No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	COM0	TB3	
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM3	TB11	NC
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	NC	TB17	NC
TB20	NC	TB19	NC
TB22	NC	TB21	NC
TB24	24G	TB23	24V

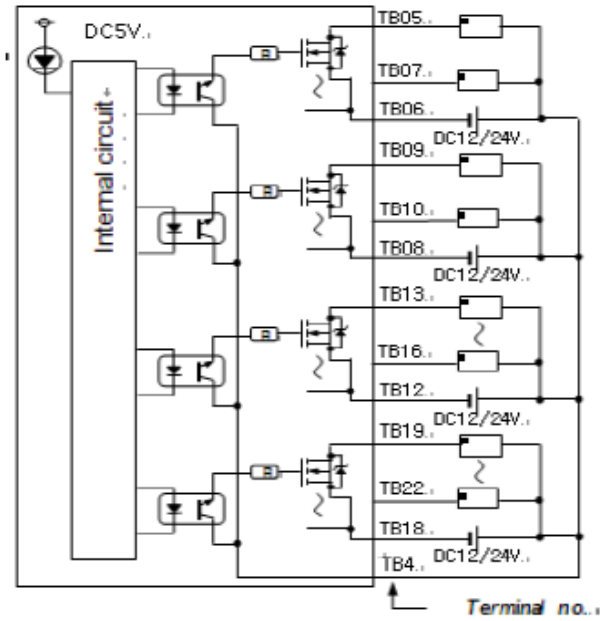
خروجی PLC سری XEC-DR30E



No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	COM0	TB3	
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM3	TB11	NC
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	COM4	TB17	NC
TB20	Q09	TB19	Q08
TB22	Q11	TB21	Q10
TB24	24G	TB23	24V

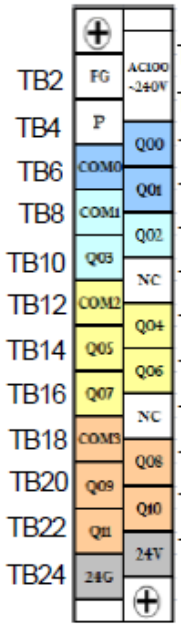
خروجی PLC سری XEC-DN30E

Circuit configuration



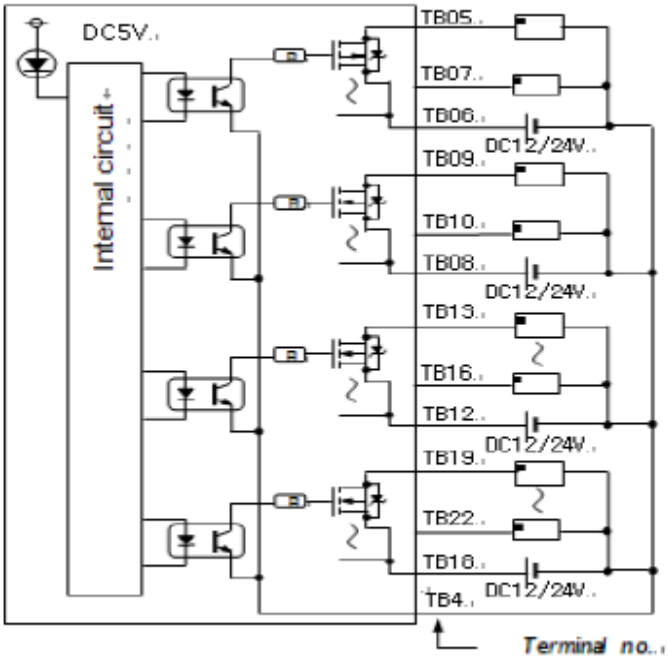
No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	P	TB3	
TB6	COM0	TB5	Q00
TB8	COM1	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM2	TB11	NC
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	COM3	TB17	NC
TB20	Q09	TB19	Q08
TB22	Q11	TB21	Q10
TB24	24G	TB23	24V

Type



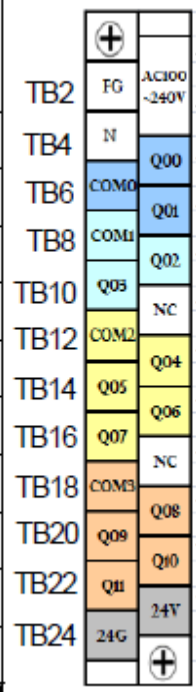
خروجی PLC سری XEC-DP30E

Circuit configuration

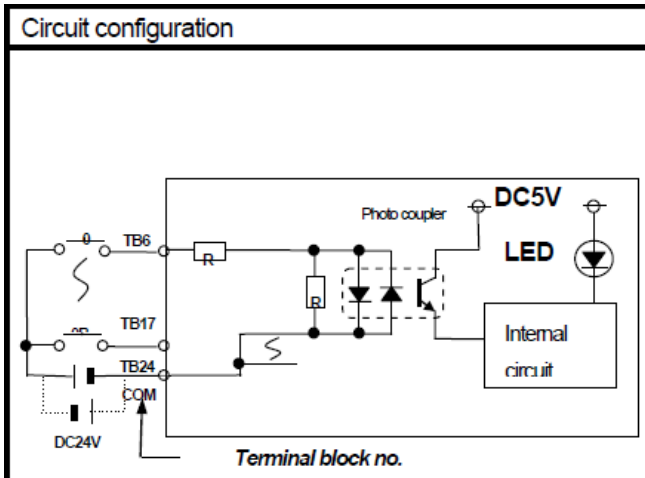


No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	N	TB3	
TB6	COM0	TB5	Q00
TB8	COM1	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM2	TB11	NC
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	COM3	TB17	NC
TB20	Q09	TB19	Q08
TB22	Q11	TB21	Q10
TB24	24G	TB23	24V

Type

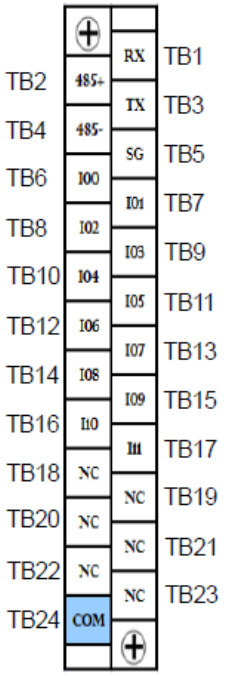


ورودی PLC سری XEC-DN20SU, XEC-DR20SU

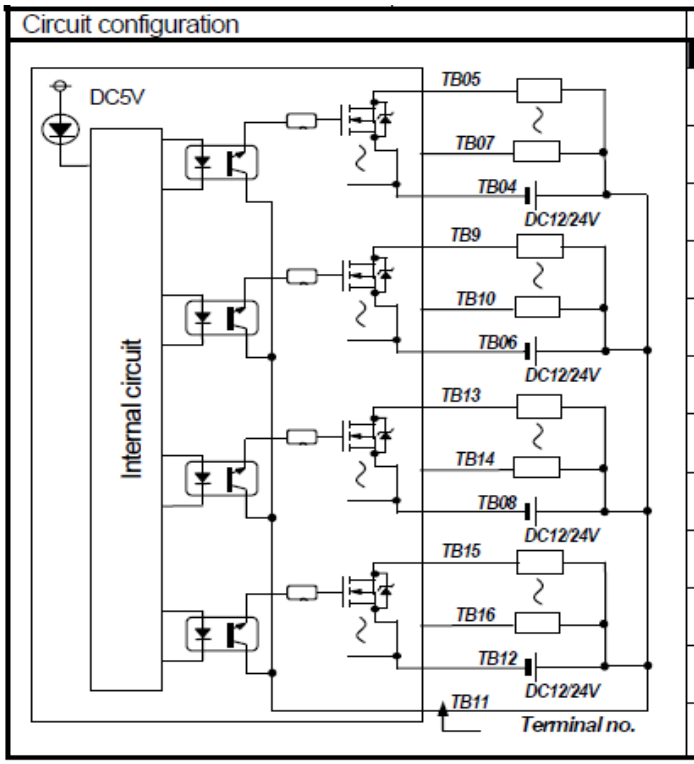


No.	Contact	No.	Contact
TB2	485+	TB1	RX
TB4	485-	TB3	TX
TB6	I00	TB5	SG
TB8	I02	TB7	I01
TB10	I04	TB9	I03
TB12	I06	TB11	I05
TB14	I08	TB13	I07
TB16	I10	TB15	I09
TB18	NC	TB17	I11
TB20	NC	TB19	NC
TB22	NC	TB21	NC
TB24	COM	TB23	NC

Type

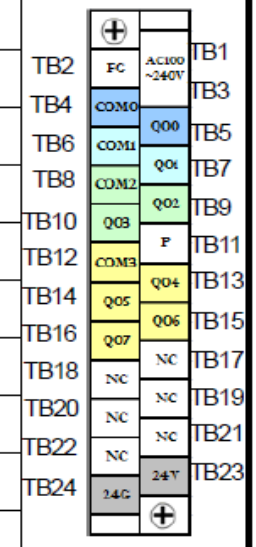


خروجی PLC سری XEC-DN20SU

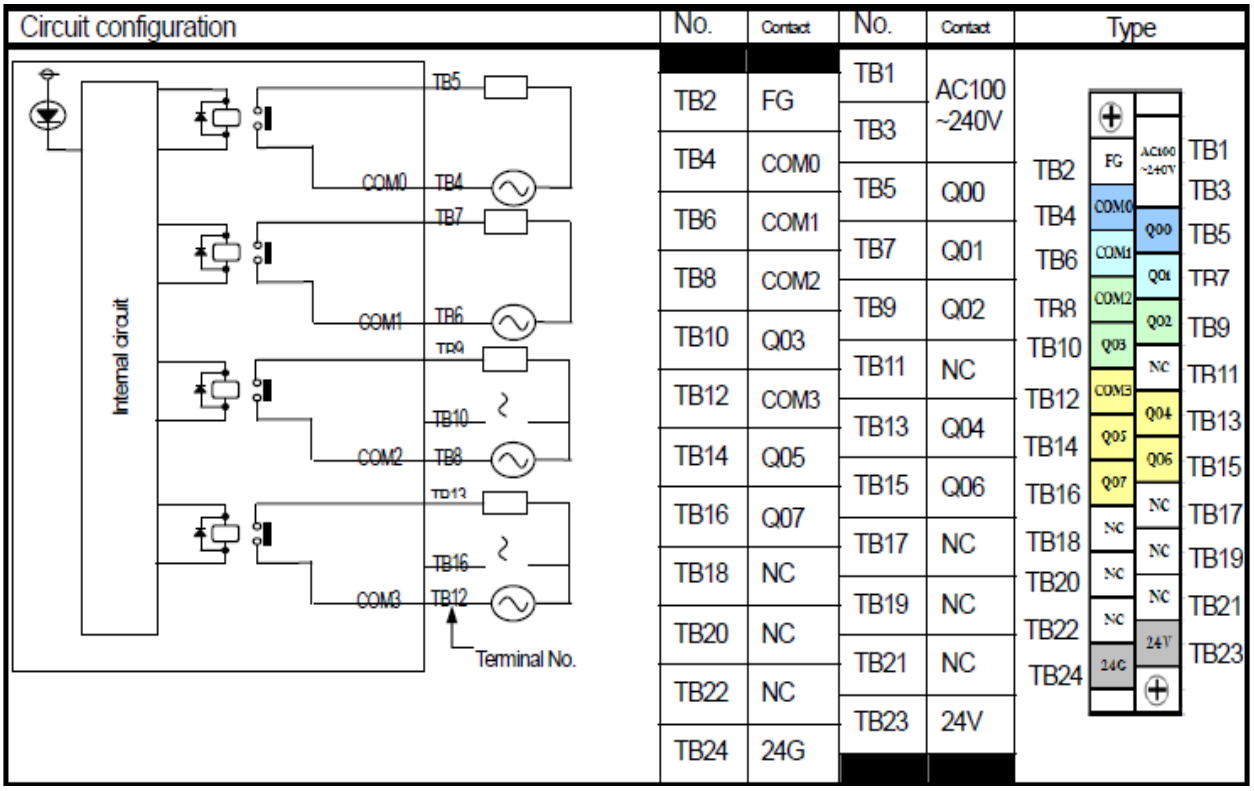


No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	COM0	TB3	
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM3	TB11	P
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	NC	TB17	NC
TB20	NC	TB19	NC
TB22	NC	TB21	NC
TB24	24G	TB23	24V

Type



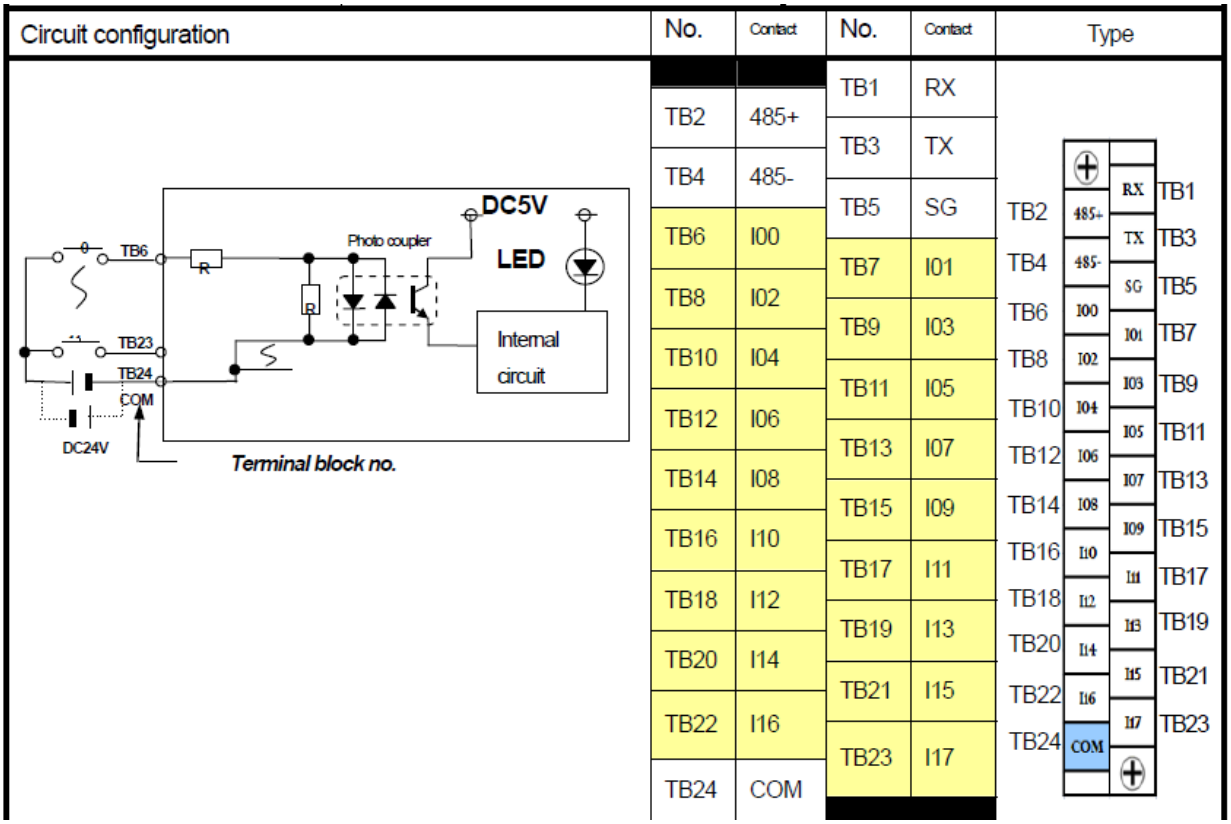
خروجی PLC سری XEC-DR20SU



No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	COM0	TB3	
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM3	TB11	NC
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	NC	TB17	NC
TB20	NC	TB19	NC
TB22	NC	TB21	NC
TB24	24G	TB23	24V

Type

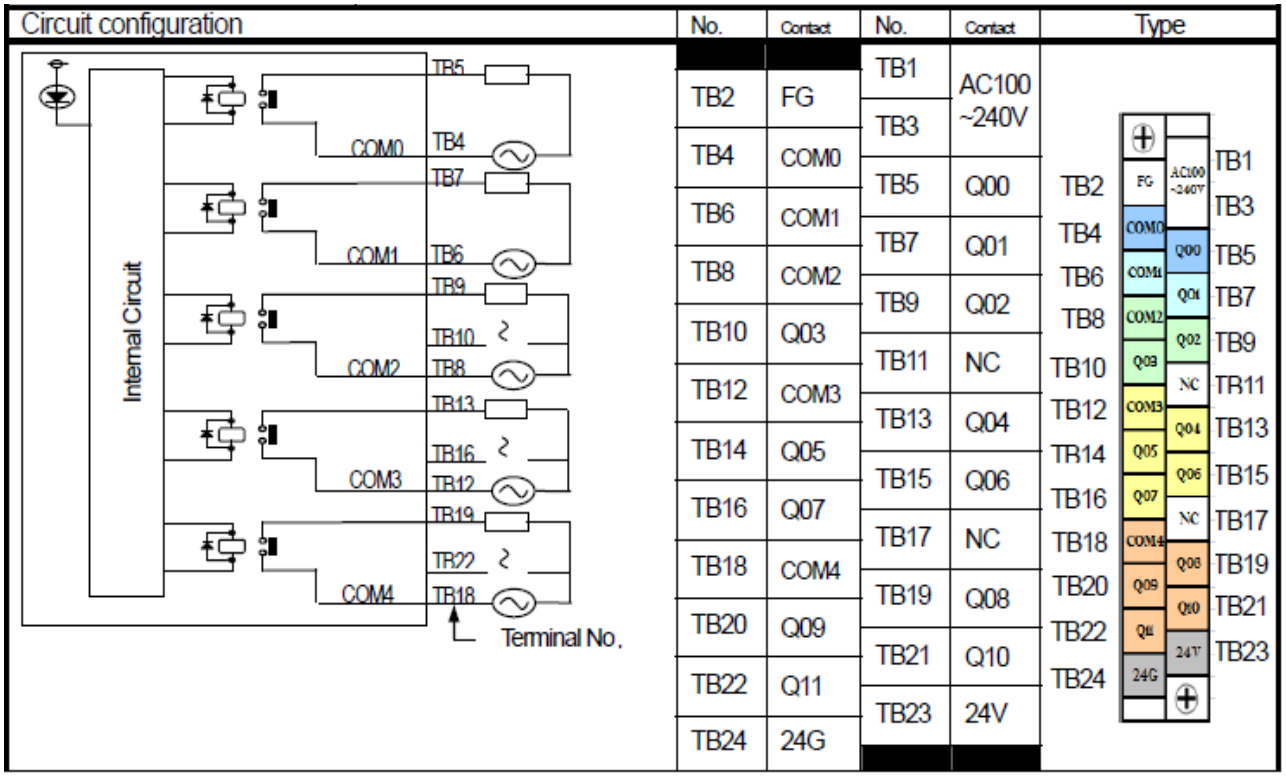
ورودی PLC سری XEC-DN30SU, XEC-DR30SU



No.	Contact	No.	Contact
TB2	485+	TB1	RX
TB4	485-	TB3	TX
TB6	I00	TB5	SG
TB8	I02	TB7	I01
TB10	I04	TB9	I03
TB12	I06	TB11	I05
TB14	I08	TB13	I07
TB16	I10	TB15	I09
TB18	I12	TB17	I11
TB20	I14	TB19	I13
TB22	I16	TB21	I15
TB24	COM	TB23	I17

Type

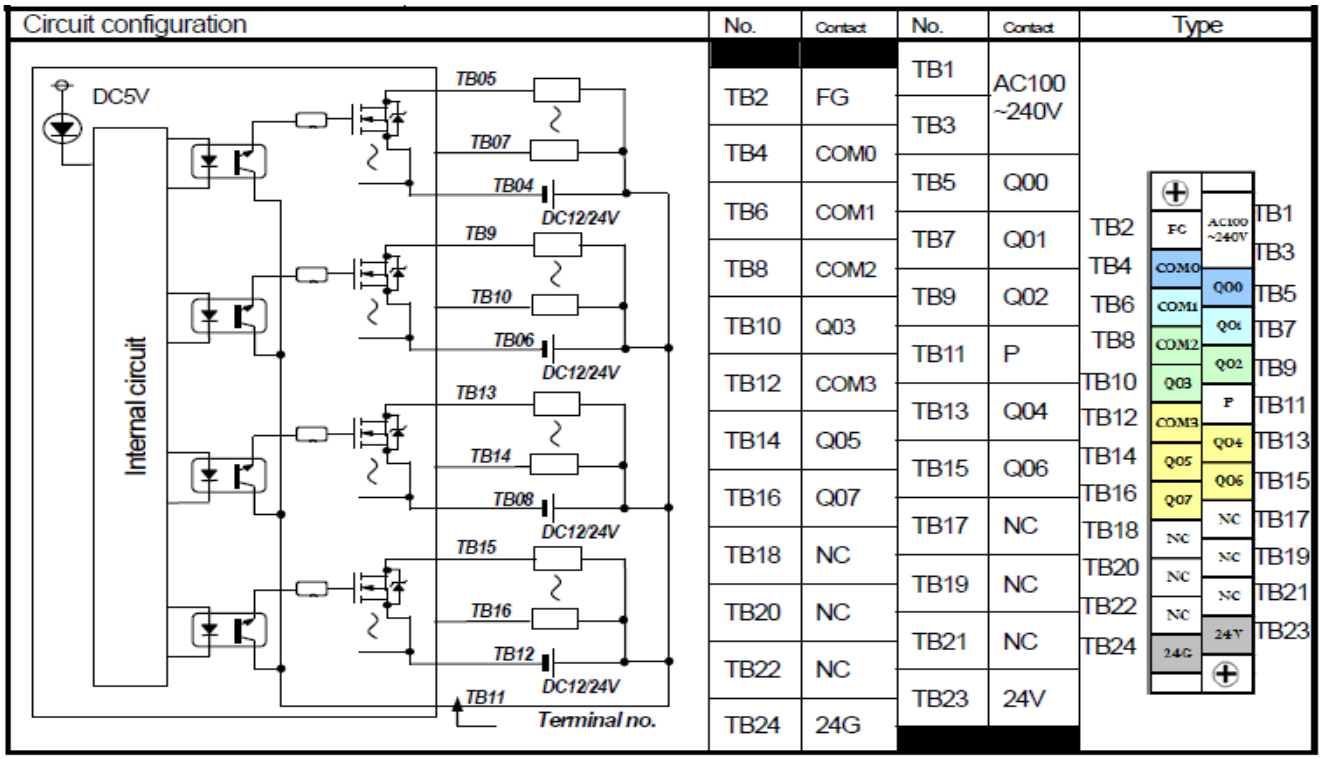
خروجی PLC سری XEC-DR30SU



No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	COM0	TB3	
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM3	TB11	NC
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	COM4	TB17	NC
TB20	Q09	TB19	Q08
TB22	Q11	TB21	Q10
TB24	24G	TB23	24V

Type

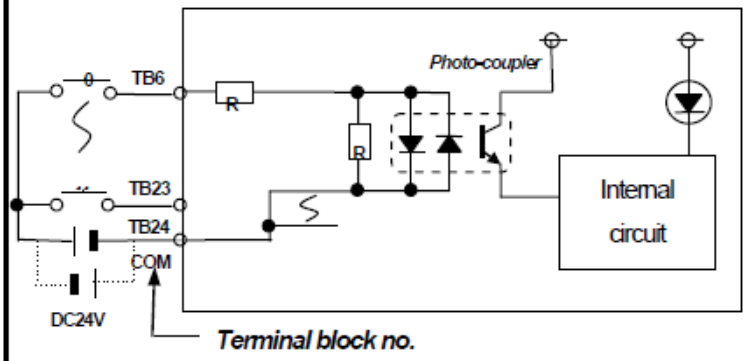
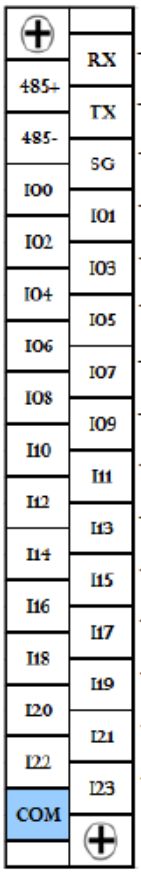
خروجی PLC سری XEC-DN30SU



No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	COM0	TB3	
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM3	TB11	P
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	NC	TB17	NC
TB20	NC	TB19	NC
TB22	NC	TB21	NC
TB24	24G	TB23	24V

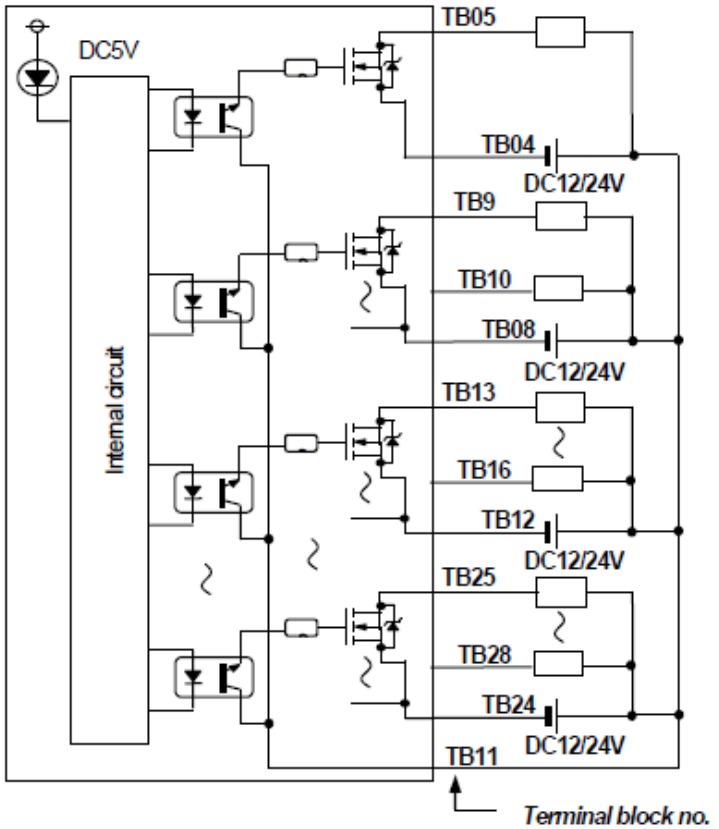
Type

ورودی PLC سری XEC-DN40SU, XEC-DR40SU

Circuit configuration	No.	Contact	No.	Contact	Type
			TB1	RX	
	TB2	485+	TB3	TX	
	TB4	485-	TB5	SG	
	TB6	I00	TB7	I01	
	TB8	I02	TB9	I03	
	TB10	I04	TB11	I05	
	TB12	I06	TB13	I07	
	TB14	I08	TB15	I09	
	TB16	I10	TB17	I11	
	TB18	I12	TB19	I13	
	TB20	I14	TB21	I15	
	TB22	I16	TB23	I17	
	TB24	I18	TB25	I19	
	TB26	I20	TB27	I21	
	TB28	I22	TB29	I23	
	TB30	COM			

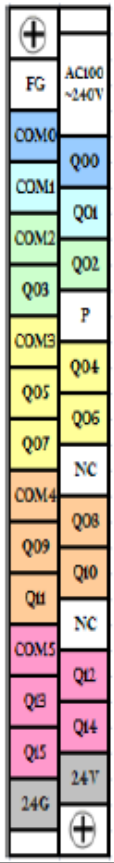
خروجی PLC سری XEC-DN40SU

Circuit configuration



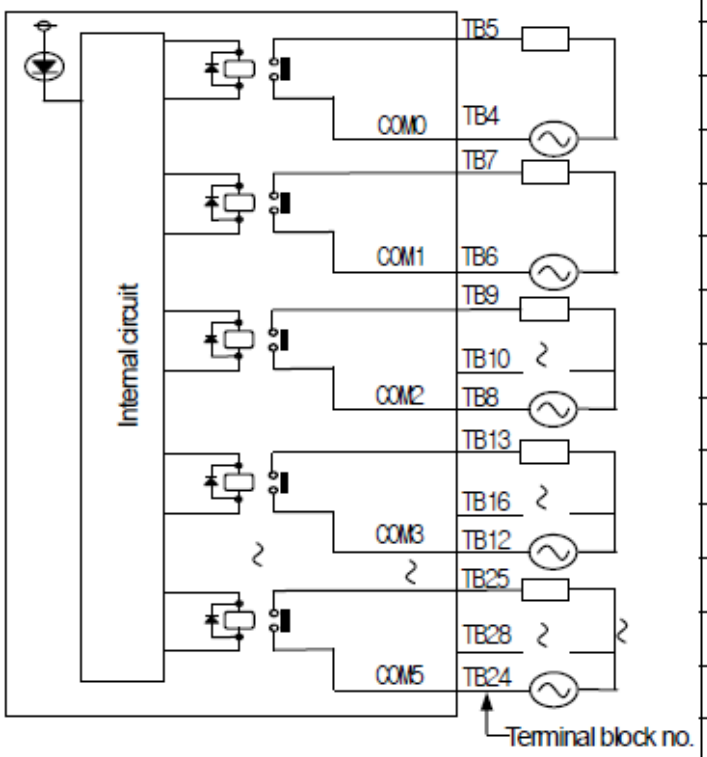
No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	COM0	TB3	
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM3	TB11	P
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	COM4	TB17	NC
TB20	Q09	TB19	Q08
TB22	Q11	TB21	Q10
TB24	COM5	TB23	NC
TB26	Q13	TB25	Q12
TB28	Q15	TB27	Q14
TB30	24G	TB29	24V

Type



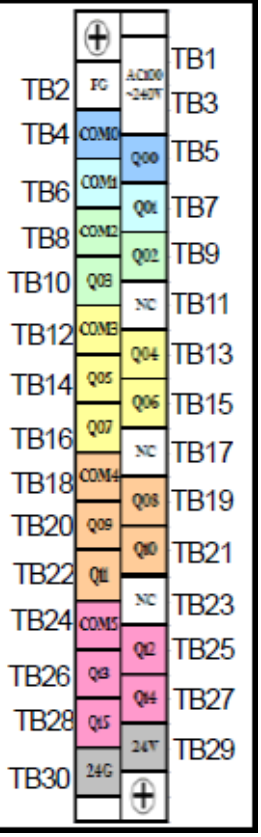
خروجی PLC سری XEC-DR40SU

Circuit configuration



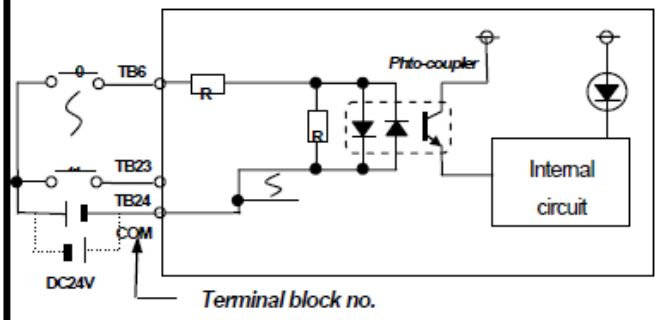
No.	Contact	No.	Contact
TB2	FG	TB1	AC100 ~240V
TB4	COM0	TB3	
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM3	TB11	NC
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	COM4	TB17	NC
TB20	Q09	TB19	Q08
TB22	Q11	TB21	Q10
TB24	COM5	TB23	NC
TB26	Q13	TB25	Q12
TB28	Q15	TB27	Q14
TB30	24G	TB29	24V

Type



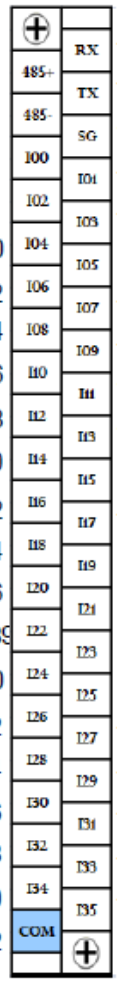
ورودی PLC سری XEC-DN60SU, XEC-DR60SU

Circuit configuration



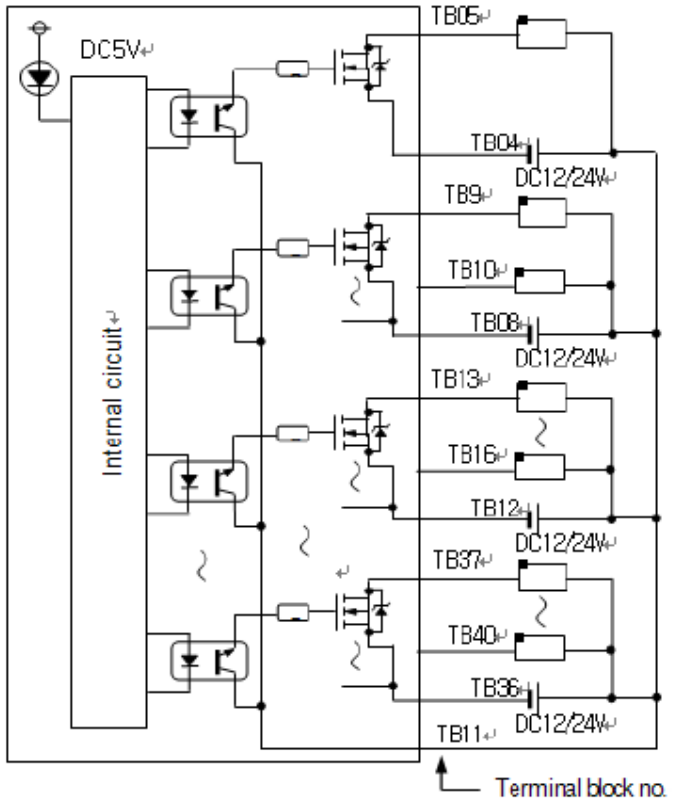
No.	Contact	No.	Contact
TB2	485+	TB1	RX
TB4	485-	TB3	TX
TB6	I00	TB5	SG
TB8	I02	TB7	I01
TB10	I04	TB9	I03
TB12	I06	TB11	I05
TB14	I08	TB13	I07
TB16	I10	TB15	I09
TB18	I12	TB17	I11
TB20	I14	TB19	I13
TB22	I16	TB21	I15
TB24	I18	TB23	I17
TB26	I20	TB25	I19
TB28	I22	TB27	I21
TB30	I24	TB29	I23
TB32	I26	TB31	I25
TB34	I28	TB33	I27
TB36	I30	TB35	I29
TB38	I32	TB37	I31
TB40	I34	TB39	I33
TB42	COM	TB41	I35

Type



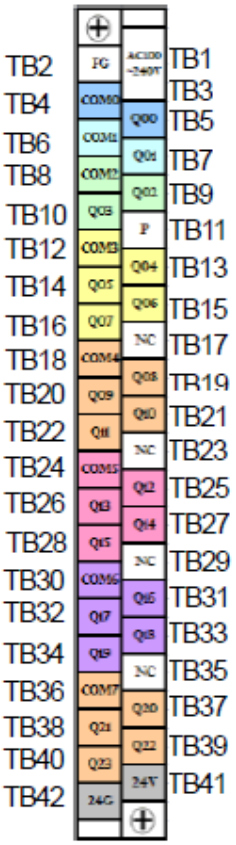
خروجی PLC سری XEC-DN60SU

Circuit configuration



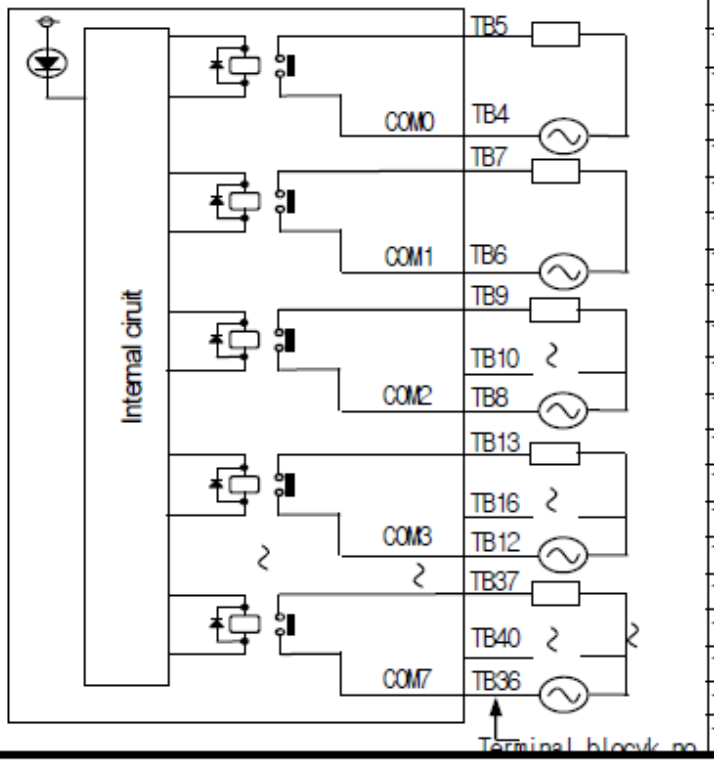
No.	Contact	No.	Contact
TB2	FG	TB1	AC100
TB4	COM0	TB3	~240V
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM3	TB11	P
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	COM4	TB17	NC
TB20	Q09	TB19	Q08
TB22	Q11	TB21	Q10
TB24	COM5	TB23	NC
TB26	Q13	TB25	Q12
TB28	Q15	TB27	Q14
TB30	COM6	TB29	NC
TB32	Q17	TB31	Q16
TB34	Q19	TB33	Q18
TB36	COM7	TB35	NC
TB38	Q21	TB37	Q20
TB40	Q23	TB39	Q22
TB42	24G	TB41	24V

Type



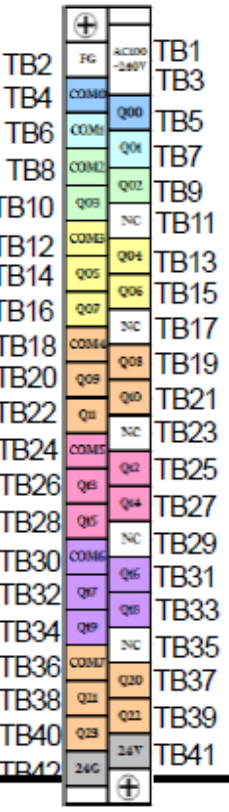
خروجی PLC سری XEC-DR60SU

Circuit configuration



No.	Contact	No.	Contact
TB2	FG	TB1	AC100
TB4	COM0	TB3	~240V
TB6	COM1	TB5	Q00
TB8	COM2	TB7	Q01
TB10	Q03	TB9	Q02
TB12	COM3	TB11	NC
TB14	Q05	TB13	Q04
TB16	Q07	TB15	Q06
TB18	COM4	TB17	NC
TB20	Q09	TB19	Q08
TB22	Q11	TB21	Q10
TB24	COM5	TB23	NC
TB26	Q13	TB25	Q12
TB28	Q15	TB27	Q14
TB30	COM6	TB29	NC
TB32	Q17	TB31	Q16
TB34	Q19	TB33	Q18
TB36	COM7	TB35	NC
TB38	Q21	TB37	Q20
TB40	Q23	TB39	Q22
TB42	24G	TB41	24V

Type



پیوست ۳

جدول اخطارها

کد خطا	علت خطا	اقدام مورد نیاز	وضعیت عملکردی	وضعیت LED	نقطه تشخیص خطا
23	برنامه برای اجرا با مشکل مواجه است	برنامه را مجدد در PLC دانلود کنید.	هشدار	نیم ثانیه بصورت چشمکزن	حالت RUN
24	پارامتر ورودی/خروجی	اگر باتری مشکل دارد باتری را تعویض کنید. وضعیت ورودی/خروجی و پارامتر تعریف ورودی/خروجی را بررسی کنید. PLC را برای تعمیر یا تعویض به نماینده شرکت تحویل دهید.	هشدار	نیم ثانیه بصورت چشمکزن	حالت RUN
25	پارامترهای پایه	اگر باتری مشکل دارد باتری را تعویض کنید. پارامترهای پایه را مجدد دانلود کنید. PLC را برای تعمیر یا تعویض به نماینده شرکت تحویل دهید.	هشدار	نیم ثانیه بصورت چشمکزن	تغییرحالت به RUN
30	ماژول به درستی نصب یا تنظیم نشده است.	ماژول را نصب و پارامتر آن را تنظیم کنید در غیر اینصورت ماژول را تغییر دهید و مجدد راه‌اندازی کنید.	هشدار	نیم ثانیه بصورت چشمکزن	تغییرحالت به RUN
31	جداشدن ماژول یا نصب اضافی ماژول	ماژول را بررسی کرده که جدا نشده باشد و یا ماژول اضافی متصل نشده باشد.	هشدار	نیم ثانیه بصورت چشمکزن	تغییرحالت به RUN
33	در حین کار دسترسی به داده‌های ماژول ورودی/خروجی امکانپذیر نیست	پس از بررسی خطای پیش آمده در خط برنامه XG5000 ماژول را تغییر دهید و مجدد برنامه را راه‌اندازی کنید.	هشدار	0.1 ثانیه بصورت چشمکزن	هر سیکل اسکن
34	در حین کار دسترسی به داده‌های ماژول ارتباطی/ماژول خاص امکانپذیر نیست	پس از بررسی خطای پیش آمده در خط برنامه XG5000 ماژول را تغییر دهید و مجدد برنامه را راه‌اندازی کنید.	خطا	0.1 ثانیه بصورت چشمکزن	پایان سیکل اسکن
39	توقف غیرعادی CPU یا نقض عملکرد CPU	خطا بعلت نویز یا خرابی در سخت‌افزار	خطا	0.1 ثانیه بصورت چشمکزن	زمان عادی
40	زمان اسکن برنامه در حین کار از زمان اسکن تعیین شده در نرم‌افزار بیشتر است.	پس از بررسی Watchdog time تعیین شده، پارامتر یا برنامه را تغییر دهید و سپس راه اندازی مجدد کنید.	هشدار	نیم ثانیه بصورت چشمکزن	در حین اجرای برنامه
41	خطای عملیات هنگام اجرای برنامه رخ می‌دهد.	خطا را حذف کرده و برنامه را مجدد دانلود کرده و راه‌اندازی کنید.	هشدار	نیم ثانیه بصورت چشمکزن	در حین اجرای برنامه
44	خطای شاخص	برنامه را اصلاح کرده، مجدد دانلود و راه‌اندازی کنید	هشدار	نیم ثانیه بصورت چشمکزن	در حین اجرای برنامه
50	خطای ناشی از دستگاه خارجی	به Flag تشخیص خطای سنگین و اصلاحات مراجعه کنید دستگاه و راه اندازی مجدد (پارامتر.Acc)	هشدار	نیم ثانیه بصورت چشمکزن	در حین اجرای برنامه

کد خطا	علت خطا	اقدام مورد نیاز	وضعیت عملکردی	وضعیت LED	نقطه تشخیص خطا
60	تابع E-Stop فعال شده است.	پس از حذف خطای E-Stop برنامه را مجدد راه‌اندازی کنید.	خطا	1 ثانیه بصورت چشمکزن	در حین اجرای برنامه
500	پشتیبان گیری از حافظه داده امکان پذیر نیست.	در صورت عدم وجود خطا در باتری، PLC به حالت STOP تغییر می‌کند.	هشدار	1 ثانیه بصورت چشمکزن	هنگام ریست
501	داده ساعت تنظیم نیست.	زمان را با XG5000 تنظیم کنید.	هشدار	0.1 ثانیه بصورت چشمکزن	زمان عادی
502	افت ولتاژ باتری	باتری را تعویض کنید.	هشدار	0.1 ثانیه بصورت چشمکزن	زمان عادی

خطاهای شمارنده سرعت بالا

کد خطا	توضیحات
20	نوع شمارنده خارج از محدوده است
21	نوع ورودی پالس خارج از محدوده تنظیم شده است.
22	درخواست کانال یک در حین اجرای کار کانال دو در حالت دریافت ۲فاز ورودی
23	تنظیم نوع خروجی مقایسه کننده شمارنده خارج از محدوده تنظیم شده است
25	مقدار از پیش تعیین شده داخلی، خارج از محدوده شمارنده تنظیم شده است
26	مقدار از پیش تعیین شده خارجی، خارج از محدوده شمارنده تنظیم شده است
27	تنظیمات شمارنده حلقه‌ای خارج از محدوده تنظیم شده است و باید حداقل عدد ۲یا بیشتر باشد.
28	مقدار خروجی مقایسه شده از مقدار حداکثر محدوده ورودی مجاز کمتر تنظیم شده است.
29	مقدار خروجی مقایسه شده از مقدار حداکثر محدوده ورودی مجاز بیشتر تنظیم شده است.
30	خطای حداقل مقدار خروجی مقایسه شده بزرگتر حداکثر خروجی مقایسه شده است.
31	خروجی مقایسه شده خارج از محدوده مقدار پیش‌فرض خروجی تنظیم شده است.
34	مقدار تنظیم شده واحد زمان خارج از محدوده است.
35	مقدار پالس در هر ۱ دور خارج از محدوده تنظیم شده است.

اگر دو یا چند خطا رخ دهد، ماژول کد خطای دوم را ذخیره می‌کند و اولی را حذف می‌کند.

جدول خطاهای PID

کد خطا	نام خطا	علت
H’0001	MV_MIN_MAX_ERR	زمانی اتفاق می افتد که حداکثر MV کمتر از حداقلMV تنظیم شده است. حتما حداکثرMV را بزرگتر از حداقلMV تنظیم کنید.
H’0002	PV_MIN_MAX_ERR	زمانی اتفاق می افتد که حداکثر PV کمتر از حداقلPV تنظیم شده است. حتما حداکثرPV را بزرگتر از حداقلPV تنظیم کنید.
H’0003	PWM_PERIOD_ERR	زمانی رخ می دهد که زمان Auto tuning یا حلقه عملیات PID زیر۱۰ میلی‌ثانیه تنظیم شود.
H’0004	SV_RANGE_ERR	هنگامی که SV بزرگتر از PV در زمان شروع Auto tuning باشد، اگر Auto-tuning درحالتForward باشد، یا زمانی که SV بزرگتر از PV در زمان شروع Auto-tuning باشد، اگر tuning درحالت reverse باشد.
H’0005	PWM_ADDRESS_ERR	زمانی اتفاق می‌افتد که محل اتصال به عنوان PWM تعیین شود اتصال خروجی خارج از P20 ~ P3F است.
H’0006	P_GAIN_SET_ERR	زمانی اتفاق می‌افتد که ضریب P کمتر از ۰ تنظیم شود.
H’0007	I_TIME_SET_ERR	زمانی اتفاق می‌افتد که ضریب I کمتر از ۰ تنظیم شود.
H’0008	D_TIME_SET_ERR	زمانی اتفاق می‌افتد که ضریب D کمتر از ۰ تنظیم شود.
H’0009	CONTROL_MODE_ERR	زمانی اتفاق می‌افتد که حالت کنترلP ، PI، PD یا PID نباشد.
H’000A	TUNE_DIR_CHG_ERR	زمانی اتفاق می‌افتد که جهت عملیات در حین Auto-tuning تغییر کند. هرگز سعی نکنید جهت عملکرد را در طول Auto-tuning تغییر دهید.
H000B	PID_PERIOD_ERR	زمانی رخ می‌دهد که زمان Auto tuning یا حلقه عملیات PID زیر۱۰ میلی‌ثانیه تنظیم شود. مطمئن شوید که زمان عملکرد PID را بزرگتر از ۱۰۰ تنظیم کنید.
H000C	HBD_WRONG_DIR	زمانی اتفاق می‌افتد که پارامتر جهت عملکرد Forward بر روی عملکرد Reverse یا پارامتر جهت عملکرد Reverse روی عملکرد Forward تنظیم شود. مطمئن شوید که جهت مناسب هر حلقه را تنظیم کنید.
H000D	HBD_SV_NOT_MATCH	در عملیات مختلط، زمانی اتفاق می‌افتد که مقدار Set هر حلقه i هم مدل نباشد. مطمئن شوید که مقدار را همجنس تنظیم کنید.

جدول هشدارهای PID

کد خطا	نام هشدار	علت
H’0001	PV_MIN_MAX_ALM	زمانی اتفاق می‌افتد که PV تنظیم شده خاج از مقدار حداقل/حداکثرPV باشد.
H’0002	PID_SCANTIME_ALM	زمانی اتفاق می‌افتد که مدت زمان عملکرد PID خیلی کوتاه باشد. زمان عملکرد PID را بیشتر از زمان اسکن PLC تنظیم کنید.
H’0003	PID_ dPV_ WARN	زمانی اتفاق می‌افتد که تغییر PV فرایند PID از حد تغییر PV بیشتر شود.
H’0004	PID_ dMV_ WARN	زمانی اتفاق می‌افتد که تغییر MV فرایند PID از حد تغییر MV بیشتر شود.
H’0005	PID_MV_MAX_WARN	زمانی اتفاق می‌افتد که MV محاسبه شده چرخه PID از حداکثرMV بیشتر شود.
H’0006	PID_MV_MIN_WARN	زمانی اتفاق می‌افتد که MV محاسبه شده چرخه PID از حداقلMV کمتر شود.