ورودی و خروجی ها در PLC به دو قسمت دیجیتال و آنالوگ تقسیم می شوند.

الف- دیجیتال:

در دنیای دیجیتال ما با صفر و یک و یا روشن و خاموش بودن سرو کار داریم و در کل دو حالت بیشتر نداریم، یا روشن و یا خاموش.

ورودی: در PLCهای Thinget ورودی های دیجیتال با X مشخص می شوند. نحوه عدد دهی ورودی های دیجیتال به صورت اکتال(هشت تایی) می باشد. بدین شکل که ورودی اول X0،ورودی دوم X1، ورودی سوم X2 ... ورودی هفتم X6، ورودی هشتم X7 و ورودی نهم X10 می باشد. دقت کنید که X8,X9 و همچنین X18,X19 و...نداریم.

اگر برای ورودی بیشتر از ماژول های اضافی استفاده می کنید نحوه آدرس دهی بدین شکل است که آدرس ها در ماژول اول از ۱۰۰، ماژول دوم از ۲۰۰، ماژول سوم از ۳۰۰و...ماژ.ل هفتم از ۷۰۰ شروع می شوند و همانند PLC اصلی به صورت هشتایی می باشند.به طور مثال اگر ماژول ورودی اضافی در جایگاه دوم باشد آدرس ها بدین شکل می باشند:

X200 X201 X202, X203, X204, X205, X206, X207

خروجی: در PLCهای Thinget خروجی های دیجیتال با Y مشخص می شوند. نحوه عدد دهی خروجی های دیجیتال به صورت اکتال(هشت تایی) می باشد. بدین شکل که خروجی اول Y0،خروجی دوم Y1، خروجی سوم Y2 ... خروجی هفتم Y6، خروجی هشتم Y7 و خروجی نهم Y10 می باشد. دقت کنید که Y8,Y9 و همچنین Y18,Y19 و...نداریم.

اگر برای خروجی بیشتر از ماژول های اضافی استفاده می کنید نحوه آدرس دهی بدین شکل است که آدرس ها در ماژول اول از ۱۰۰، ماژول دوم از ۲۰۰، ماژول سوم از ۳۰۰و...ماژ.ل هفتم از ۷۰۰ شروع می شوند و همانند PLC اصلی به صورت هشتایی می باشند.به طور مثال اگر ماژول خروجی اضافی در جایگاه دوم باشد آدرس ها بدین شکل می باشند:

Y200, Y201, Y202, Y203, Y204, Y205, Y206, Y207

ب- آنالوگ:

در دنیای آنالوگ بر خلاف دیجیتال ما با مقادیر مختلف در یک محدوده مشخص سر و کار داریم. به طور مثال در خواندن دما بوسیله ترموکوپل ما با مقدار دما مثلا ۲۱۰درجه یا ۲۲۰ درجه مواجه هستیم و مقدار عددی به دست آمده برایمان مهم است و هر عددی ارزش خود را دارد و در محاسبات مهم می باشد.

ورودی: در PLCهای Thinget ورودی های آنالوگ با ID مشخص می شوند. نحوه عدد دهی ورودی های آنالوگ به صورت اکتال(هشت تایی) می باشد. بدین شکل که در مدل هایی مانند XC3-19-RE که ورودی آنالوگ در PLC اصلی وجود دارد ورودی اول ID0،ورودی دوم ID1، ورودی سوم ID2 ... ورودی هفتم ID6، ورودی هشتم ID7 و ورودی نهم ID10 می باشد. دقت کنید که ID8,ID9 و همچنین ID18,ID19 و...نداریم.

اگر برای ورودی بیشتر از ماژول های اضافی استفاده می کنید نحوه آدرس دهی بدین شکل است که آدرس ها در ماژول اول از ۱۰۰، ماژول دوم از ۲۰۰، ماژول سوم از ۳۰۰و...ماژ.ل هفتم از ۷۰۰ شروع می شوند و همانند PLC اصلی به صورت هشتایی می باشند.به طور مثال اگر ماژول ورودی اضافی در جایگاه دوم باشد آدرس ها بدین شکل می باشند:

ID200, ID201 ,ID202, ID203 ,ID204, ID205, ID206, ID207

خروجی: در PLCهای Thinget خروجی های آنالوگ با QD مشخص می شوند. نحوه عدد دهی خروجی های آنالوگ به صورت اکتال(هشت تایی) می باشد. بدین شکل که در مدل هایی مانند XC3-19-RE که خروجی آنالوگ در PLC اصلی وجود دار خروجی اول QD0،خروجی دوم QD1، خروجی سوم QD2 ... خروجی هفتم QD6، خروجی هشتم QD7 و خروجی نهم QD10 می باشد. دقت کنید که QD8,QD9 و همچنین QD18,QD19 و...نداریم.

اگر برای خروجی بیشتر از ماژول های اضافی استفاده می کنید نحوه آدرس دهی بدین شکل است که آدرس ها در ماژول اول از ۱۰۰، ماژول دوم از ۲۰۰، ماژول سوم از ۳۰۰و...ماژ.ل هفتم از ۷۰۰ شروع می شوند و همانند PLC اصلی به صورت هشتایی می باشند.به طور مثال اگر ماژول خروجی اضافی در جایگاه دوم باشد آدرس ها بدین شکل می باشند:

QD200, QD201, QD202, QD203, QD204, QD205, QD206, QD207

حافظه ی داخلی:

حافظه داخلی پی ال سی به دو قسمت ماندگار و غیر ماندگار تقسیم می شود. قسمت ماندگار دارای این ویژگی است که با قطع شدن برق پی ال سی مقدار و یا حالت خود را حفظ کرده و آن را از دست نمی دهد. ولی حافظه غیر ماندگار با قطع برق مقدار و یا حالت خود را از دست می دهد.

۱- رله های کمکی یا حافظه های بیتی

الف- بیت ها یا Flagها

این قسمت از حافظه داخلی با M شناخته می شوند. این M ها رله های کمکی هستند که برای برنامه نویسی مورد استفاده قرار می گیرند. این حافظه ها از شماره ۰ تا ۷۹۹۹ می باشند که از شماره ۳۰۰۰ به بعد آن جزو گروه ماندگار می باشند.

ب- تایمرها

این قسمت از حافظه داخلی با T شناخته می شوند.

پ- کانترها

این قسمت از حافظه داخلی باC شناخته می شوند.

ت- استیت رله ها

این قسمت از حافظه داخلی باS شناخته می شوند.

۲- حافظه های عددی یا رجیسترها

این قسمت از حافظه داخلی با D شناخته می شوند. این D ها رجیسترهستند که برای برنامه نویسی مورد استفاده قرار می گیرند.
این حافظه ها از ۰ تا ۷۹۹۹ می باشند که از آدرس ۴۰۰۰ به بعد آن ماندگار می باشند.

برخی از حافظه های ویژه:

M8000: این حافظه همیشه روشن است. اگر می خواهید دستوری را همیشه انجام دهید،مانند خواندن مقادیر آنالوگ، می توانید از کنتاکت باز این بیت استفاده کنید.

M8002: این حافظه در اولین اسکن تایم ست بوده و پس از آن ریست می شود. اگر می خواهید دستوری را با روشن شدن پی ال سی فقط یکبار انجام دهید می توانید از کنتاکت باز این بیت استفاده کنید.

M8003: بر خلاف M8002 این بیت در اسکن تایم اول ریست بوده و پس از آن ست می شود، اگر می خواهید دستوری در موقع روشن شدن پی ال سی انجام نشود و پس از آن همیشه انجام شود می توانید از کنتاکت باز این بیت استفاده کنید.

M8013: این بیت به فاصله زمانی یک ثانیه روشن و خاموش می شود.

D8013: در این رجیستر مقدار ثانیه مربوط به ساعت داخلی پی ال سی وجود دارد.

D8014: در این رجیستر مقدار دقیقه مربوط به ساعت داخلی پی ال سی وجود دارد.

D8015: در این رجیستر مقدار ساعت مربوط به ساعت داخلی پی ال سی وجود دارد.

D8016: در این رجیستر عدد روز مربوط به تاریخ داخلی پی ال سی وجود دارد.

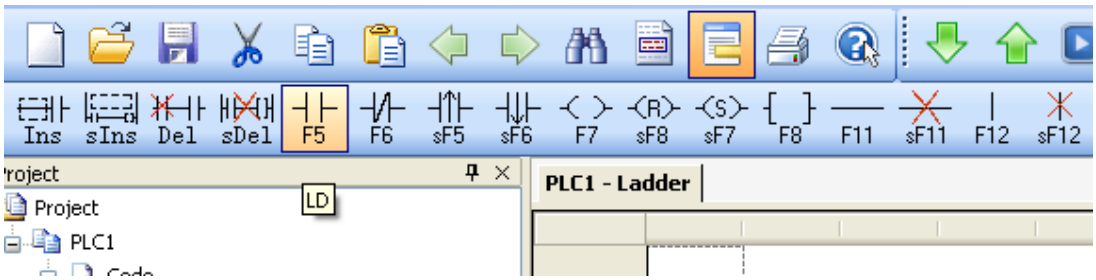
D8017: در این رجیستر عدد ماه مربوط به تاریخ داخلی پی ال سی وجود دارد.

D8018: در این رجیستر عدد سال مربوط به تاریخ داخلی پی ال سی وجود دارد.

زبان برنامه نویسی در PLC های Thinget هم به صورت Ladder و هم به صورت کد نویسی می باشد که به دلیل راحت تر بودن زبان Ladder آن را توضیح می دهیم. ساختار کلی زبان Ladder بدین صورت می باشد که در سمت چپ شرط ها و در سمت راست دستورات نوشته می شوند. اگر شرط برقرار باشد دستوری که به آن وصل است انجام می شود و اگر شرط برقرار نباشد دستور انجام نمی شود. برای ملموس تر شدن این موضوع فرض کنید که خط های دستور مانند مسیرهای جریانی می باشند که می بایست مسیر بسته شود تا جریان به انتهای خط رسیده و باعث به کار افتادن دستور شود.شرط ها را می توان با هم سری و یا موازی کرد و همچنین می توان با یک شرط چند دستور را به طور موازی با هم انجام داد.

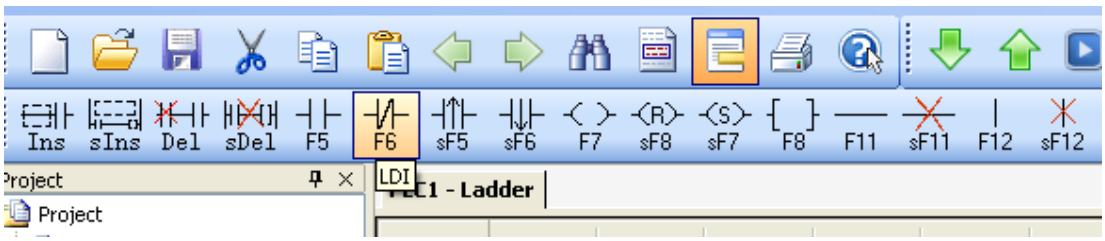
شرط ها:

:



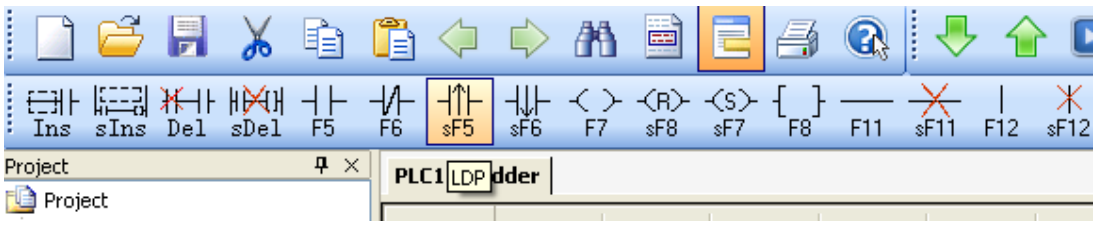
کنتاکت باز بدین معنی می باشد که با روشن بودن عملگر(ورودی،خروجی و یا رله های کمکی از حافظه ی داخلی،تایمر، کانتر) دستور انجام شود. این گزینه همانند کلید نرمالی اپن در کنتاکتورها می باشد که با روشن شدن عملگر تیغه آن بسته شده و باعث فعال شدن دستور می شوند. با این شرط تا زمانی که عملگر روشن است دستور انجام می شود.

:



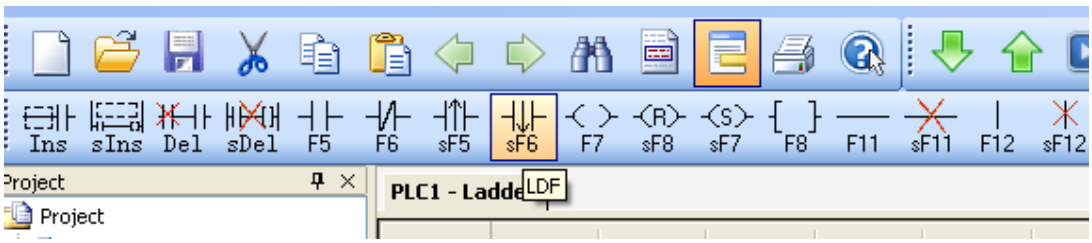
کنتاکت بسته بدین معنی می باشد که با خاموش بودن عملگر(ورودی،خروجی و یا رله های کمکی از حافظه ی داخلی،تایمر، کانتر) دستور انجام شود. این گزینه همانند کلید نرمالی کلوز در کنتاکتورها می باشد که تا زمانی که عملگر خاموش است تیغه آن بسته می باشد و باعث فعال شدن دستور می شود و با روشن شدن عملگر تیغه آن باز شده و دستور دیگر انجام نخواهد شد .با این شرط تا زمانی که عملگر خاموش است دستور انجام می شود.

:



لبه بالا رونده بدین معنی است که با روشن شدن عملگر فقط یکبار شرط برقرار شده و دستور انجام می شود و مادامی که عمگر روشن می ماند دیگر شرط برقرار نمی شود،تا زمانی که عملگر خاموش شده و دوباره روشن شود.

:

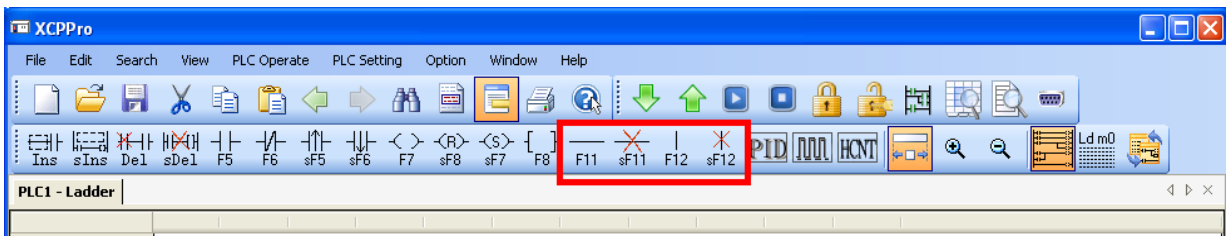


لبه پایین رونده بدین معنی است که با خاموش شدن عملگر فقط یکبار شرط برقرار شده و دستور انجام می شود و مادامی که عمگر خاموش می ماند دیگر شرط برقرار نمی شود،تا زمانی که عملگر روشن شده و دوباره خاموش شود.

<= :

این کنتاکت حاصل مقایسه دو عملگر به صورت رجیستری می باشد. با این شرط دو رجیستر با هم مقایسه شده و اگر نتیجه مقایسه درست بود شرط برقرار می شود. با این شرط بزرگتر بودن، کوچکتر بودن ، مساوی و یا نامساوی بودن دو رجیستر و یا یک رجیستر با یک مقدار ثابت بررسی می شود. برای استفاده از این شرط باید ابتدا LD را تایپ کنید ، سپس بدون فاصله علامت < = > را قرار داده ،سپس یک فاصله و بعد عملگر اول ،مجددا یک فاصله و در نهایت عملگر دوم نوشته شود.

** برای سری و موازی کردن شرطها و موازی کردن دستورات می توانید از گزینه های زیر استفاده کنید:



دستورات بیتی:

SET:

از این دستور برای Set (و یا روشن کردن و یا یک کردن) یک بیت (خروجی و یا رله های کمکی حافظه داخلی) استفاده می شود. اگر از این دستور استفاده کنید زمانی که شرط آن برقرار شود عملگر آن ست شده و در همان وضعیت باقی می ماند تا به وسیله دستور دیگری ریست شود.

RST:

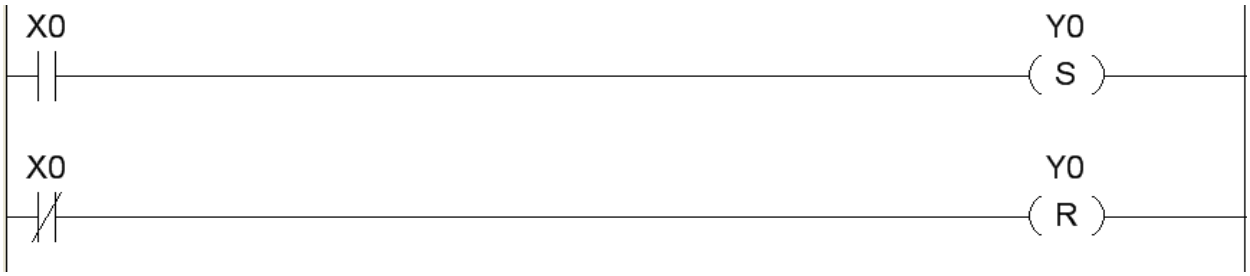
از این دستور برای Reset (و یا خاموش کردن و یا صفر کردن) یک بیت (خروجی و یا رله های کمکی حافظه داخلی) استفاده می شود. اگر از این دستور استفاده کنید زمانی که شرط آن برقرار شود عملگر آن ریست شده و در همان وضعیت باقی می ماند تا به وسیله دستور دیگری ست شود.

OUT:

این دستور به نوعی ترکیب دو دستور بالا می باشد. بدین معنی که اگر شرط کنترلی برقرار باشد عملگر ست شده و اگر شرط کنترلی خاموش شود عملگر نیز ریست می شود.

مثال:

فرض کنید می خوهیم برنامه ای بنویسیم که با روشن شدن ورودی X0 خروجی Y0 روشن شده و با خاموش شدن X0 ، Y0نیز خاموش شود.



با خط اول برنامه بالا زمانی که X0 روشن است دستور SET Y0 انجام می شود و با خط دوم زمانی که X0 خاموش است دستور RST Y0 انجام می شود. از آنجایی که در مثال بالا عملگر شرط ها یکی است می توان برنامه فوق را به شکل زیر نوشت:

در این برنامه از دستور OUT استفاده شده است و همانطور که گفته شد اگر شرط برقرار باشد دستور SET و اگر شرط برقرار نباشد دستور RST انجام می شود. بنابراین با روشن شدن X0،Y0 روشن شده و با خاموش شدن X0 ، Y0خاموش می شود.

در این برنامه X0وM0 با هم موازی شده و درنهایت با کنتاکت بسته X1 سری شده اند.این بدان معناست که اگر هرکدام از X0و یا M0 و یا هردوی آنها روشن شوند و X1 خاموش باشد دستور انجام می شود.اگر X1 روشن باشد دستور انجام نخواهد شد حتی اگر هردوی X0 و یا M0 روشن باشند.

دستورات ریاضی:

ADD:

این دستور عمل جمع بین دو عملگر را انجام می دهد. در این دستور برای عملگرهای اول و دوم هم می توان از رجیسترها حافظه و هم از مقادیر ثابت استفاده کرد ولی عملگر سوم باید حتما از رجیسترهای حافظه باشد.در این دستور عملگر اول با عملگر دوم جمع شده و درحاصل عملگر سوم ریخته می شود.

ADD	D0	D1	D2

داریم:

ADD: دستور جمع

D0: عملگر اول

D1:عملگر دوم

D2: عملگر سوم

SUB:

این دستور عمل تفریق بین دو عملگر را انجام می دهد. در این دستور برای عملگرهای اول و دوم هم می توان از رجیسترها حافظه و هم از مقادیر ثابت استفاده کرد ولی عملگر سوم باید حتما از رجیسترهای حافظه باشد.در این دستور عملگر دوم از عملگر اول کم شده و حاصل در عملگر سوم ریخته می شود.

SUB	D0	D1	D2

داریم:

SUB: دستور تفریق

D0: عملگر اول

D1:عملگر دوم

D2: عملگرسوم

MUL:

این دستور عمل ضرب بین دو عملگر را انجام می دهد. در این دستور برای عملگرهای اول و دوم هم می توان از رجیسترها حافظه و هم از مقادیر ثابت استفاده کرد ولی عملگر سوم باید حتما از رجیسترهای حافظه باشد.در این دستور عملگر اول در عملگر دوم جمع شده وحاصل در عملگر سوم ریخته می شود.

MUL	D0	D1	D2

داریم:

MUL: دستور ضرب

D0: عملگر اول

D1:عملگر دوم

D2: عملگرسوم

:DIV

این دستور عمل تقسیم بین دو عملگر را انجام می دهد. در این دستور برای عملگرهای اول و دوم هم می توان از رجیسترها حافظه و هم از مقادیر ثابت استفاده کرد ولی عملگر سوم باید حتما از رجیسترهای حافظه باشد.در این دستور عماگر اول بر عملگر دوم تقسیم شده وحاصل در عملگر سوم ریخته می شود بدین صورت که خارج قسمت در عملگر سوم و باقیمانده در آدرس بعدی عملگر سوم ریخته می شود. برای مثال اگر دستور را مطابق زیر بنویسیم D0 بر D1 تقسیم شده و خارج قسمت در D2 و باقیمانده در D3 ریخته می شود.

داریم:

DIV D0 D1 D2

DIV: دستور تقسیم

D0: عملگر اول

D1:عملگر دوم

D2: عملگر سوم

مثال:

در این مثال با روشن شدن M0 محتوای رجیستر D0 با عدد ۱۰ جمع شده و حاصل در رجیستر D20 ریخته می شود.

در این مثال با روشن شدن M0 ،محتوای رجیستر D12 از محتوای رجیستر D0کم شده و حاصل در D20ریخته می شود.

در این مثال با روشن شدن M0 محتوای رجیستر D10 بر محتوای رجیستر D12 تقسیم شده و خارج قسمت تقسیم در D20 و باقیمانده آن در D21 ریخته می شود.

دستور کپی کردن:

۱-MOV: کپی کردن یک رجیستر

این دستور به نوعی عمل کپی کردن را انجام می دهد. در این دستور برای عملگر اول هم می توان از رجیسترها حافظه و هم از مقادیر ثابت استفاده کرد ولی عملگر دوم باید حتما رجیستر باشد.در این دستور عماگر اول کپی شده و درعملگر دوم ریخته می شود.

داریم:

MOV D0 D1

MOV: دستور کپی کردن

D0: عملگر اول

D1:عملگر دوم

مثال:

در این مثال زمانی که M0 روشن شود عدد ۲۰۰ در داخل رجیستر D315 ریخته می شود.

در این مثال با روشن شدن M0 محتوای رجیستر D10 ،داخل رجیستر D315 ریخته می شود. لازم به تذکر است که محتوای D10 تغییر نمی کند و فقط به D315 منتقل می شود.

۲-BMOV: کپی کردن چند رجیستر همزمان

از دستور BMOV برای کپی کردن چند رجیستر با یک دستور استفاده می شود.در واقع می توان به جای چند دستور MOV ،برای گرفتن فضای کمتر از حافظه و قشنگ تر شدن برنامه از BMOV استفاده کرد. در این دستور شما ۳ عملگر را باید تعریف کنید. عملگر اول آدرس اولین رجیستریست که می خواهید مقدار داخل آن را کپی کنید. عملگر دوم آدرس اولین رجیستریست که می خواهید مقادیر کپی شده را به آن ها منتقل کنید و عملگر سوم تعداد رجیسترهاییست که می خواهید مقادیر آنها را کپی کنید.

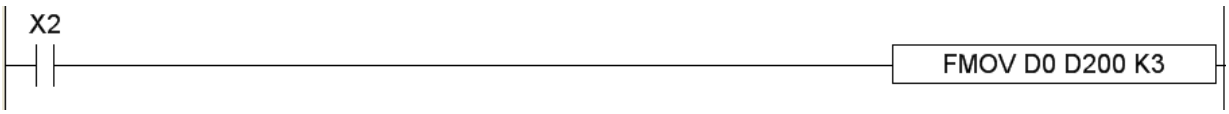
برای نوشتن دستور ابتدا BMOV را تایپ کرده،یک جای خالی گذاشته سپس عملگر اول،یک جای خالی، عملگر دوم،یک جای خالی و در نهایت عملگر سوم را تایپ کنید. دقت کنید که عملگر اول و دوم باید حتما حافظه رجیستری باشند ولی عملگر سوم می تواندهم رجیستر و هم مقدار ثابت باشد.

در مثال فوق مقدار رجیستر D0 در D200،D1 در D201و D2در D202 کپی می شود.

۳-FMOV: کپی کردن یک مقدار ثابت در چندین رجیستر

از دستور FMOV برای کپی کردن مقدار یک رجیستر یا یک عدد ثابت در چند رجیستر با یک دستور استفاده می شود. در واقع می توان به جای چند دستور MOV ،برای گرفتن فضای کمتر از حافظه و قشنگ تر شدن برنامه از FMOV استفاده کرد. در این دستور شما ۳ عملگر را باید تعریف کنید. عملگر اول آدرس رجیستر و یا مقدار ثابتیست که می خواهید آن را کپی کنید. عملگر دوم آدرس اولین رجیستریست که می خواهید مقادیر کپی شده را به آن ها منتقل کنید و عملگر سوم تعداد رجیسترهاییست که می خواهید عملگر اول را در آنها کپی کنید.

برای نوشتن دستور ابتدا FMOV را تایپ کرده،یک جای خالی گذاشته سپس عملگر اول،یک جای خالی، عملگر دوم،یک جای خالی و در نهایت عملگر سوم را تایپ کنید. دقت کنید که عملگر دوم باید حتما حافظه رجیستری باشد ولی عملگر اول و سوم می توانندهم رجیستر و هم مقدار ثابت باشند.



در مثال فوق محتوای رجیستر D0 در رجیسترهای D200,D201,D202 کپی میشود.

نحوه استفاده از تایمرها:

تایمر ها به دو شکل تقسیم بندی می شوند، یا بوسیله واحد زمانی و یا نوع ماندگاری.

تایمرها دارای ۳ واحد متفاوت می باشند، ۱۰۰، ۱۰ و ۱ میلی ثانیه .

از نظر ماندگاری نیز تایمر ها به دو دسته ماندگار و غیر ماندگار تقسیم می شوند. دسته غیر ماندگار با غیر فعال شدن شرط مقدارشان صفر می شود ولی دسته ماندگار با غیر فعال شدن شرطشان مقدار را نگه می دارند و اگر مجددا شرط برقرار شود از مقدار قبلی شروع به شمارش کرده و آن را ادامه می دهد. توجه داشته باشید که این ماندگاری برای زمانیست که شرط غیر فعال شود و اگر برق PLC قطع شود تمامی تایمرها صفر می شوند.

غیرماندگار	100mS	T0~T99
ماندگار	100mS	T100~T199
غیرماندگار	10mS	T200~T299
ماندگار	10mS	T300~T399
غیرماندگار	1mS	T400~T499
ماندگار	1mS	T500~T599
تایمرهای اینتراپت	1mS	T600~T618

تایمر ها به این شکل کار می کنند که با فعال شدن شرط، شروع به کار کرده و پس از رسیدن به زمان مورد نظر، بیت مربوط به تایمر ست می شود. البته اگر شرط غیر فعال نشود تایمر به شمارش ادامه می دهد. همچنین تایمرها دارای رجیستر می باشند که مقدار شمارش شده توسط تایمر در آن موجود است، بنابراین می توان برای شرط ها هم از بیت تایمر و هم از رجیستر آن استفاده کرد.

برای فعال کردن تایمرها باید از دستور OUT استفاده کرد. بدین شکل که ابتدا OUT را تایپ کرده ، یک جای خالی گذاشته سپس شماره تایمر (مثلا T1) ، مجددا یک جای خالی و درنهایت مقدار آن را می نویسیم. نحوه محاسبه زمان تایمرها بدین شکل است که عددی که درنهایت به عنوان مقدار نوشته می شود در واحد تایمر ضرب شده و مقدار زمان را مشخص می کند. به عنوان مثال فرض کنید دستور را بدین شکل نوشته ایم:

OUT T20 k300

در این مثال عدد ۳۰۰ در واحد تایمر شماره ۲۰ که ۱۰۰میلی ثانیه است ضرب شده و در نهایت مقدار ۳۰ ثانیه را مشخص می کند.

توجه داشته باشید که یک تایمر تا زمانی کار می کند که شرطش برقرار باشد، مثلا برای فعال کردن یک تایمر نمی توانید از لبه بالا رونده استفاده کنید زیرا این شرط فقط برای یک اسکن تایم فعال است و پس از آن غیر فعال می شود. بنا بر این باید از کنتاکت باز یا بسته استفاده کرد. در زمان هایی هم که به طور مثال می خواهید با فعال شدن یک ورودی به طور لحظه ای (کلید استارت) یک تایمر را فعال کنید باید از رله های کمکی استفاده کنید، بدین شکل که با فعال شدن ورودی مثلا M0 را ست کنید و با شرط کنتاکت باز M0 تایمر مورد نظر خود را فعال کنید.

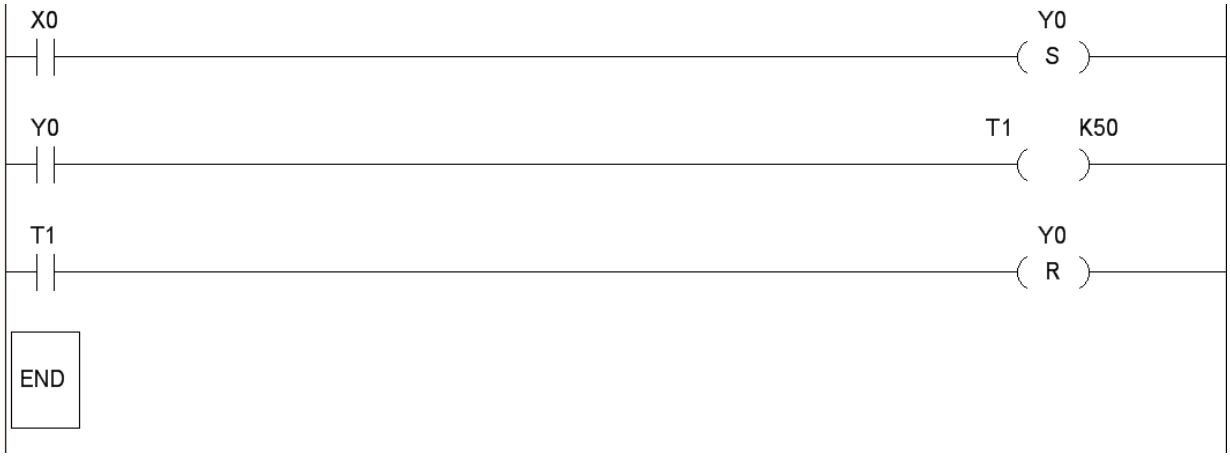
مثال:

در این مثال با روشن شدن M0 تایمر T1 فعال شده و پس از یک ثانیه بیت تایمر ست می شود.

مثال:

برنامه ای بنویسید که با زدن کلید استارت، خروجی اول روشن شده و پس از ۵ ثانیه خاموش شود.

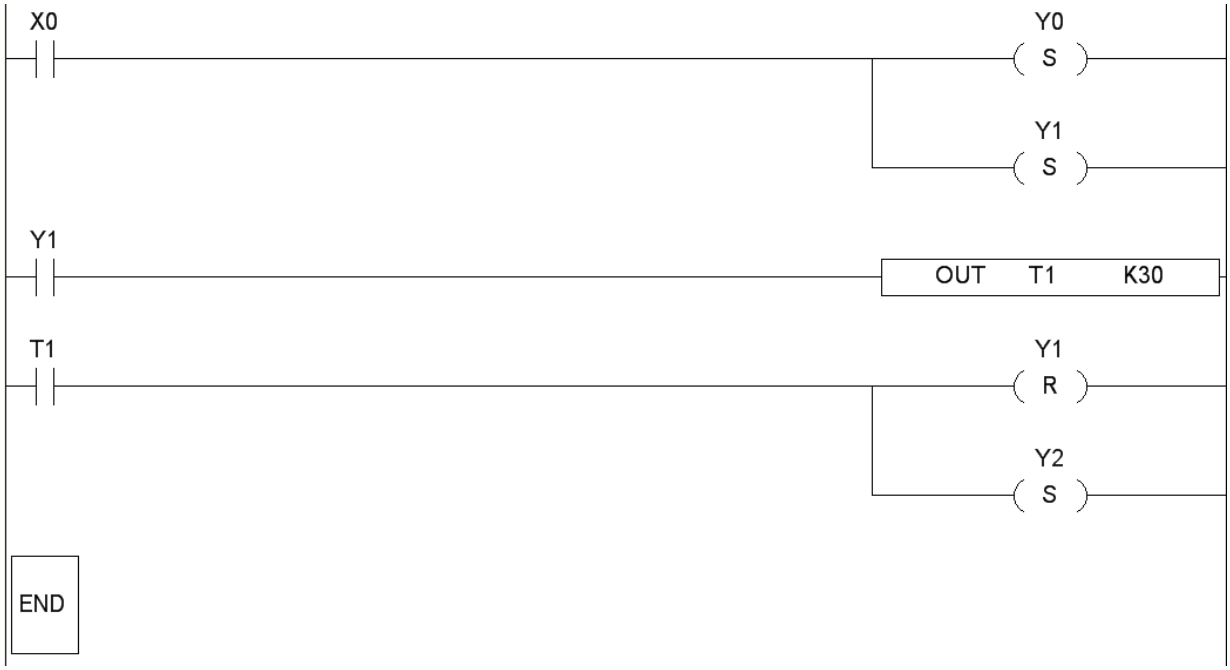
جواب: فرض می کنیم شاسی استارت به X0 متصل شده است.



مثال:

برنامه یک مدار ستاره مثلث اتوماتیک ،استارت به کنتاکتور شبکه را بنویسید که پس از ۳ ثانیه کنتاکتور ستاره از مدار خارج شود و کنتاکتور مثلث شروع به کار کند.

جواب:فرض می کنیم شاسی استارت به X0 متصل باشد.همچنین Y0به بوبیین کنتاکتور شبکه، Y1 کنتاکتور ستاره و Y2 کنتاکتور مثلث وصل شده باشند.


X0
Y0
S
Y1
S
Y1
OUT T1 K30
T1
Y1
R
Y2
S
END

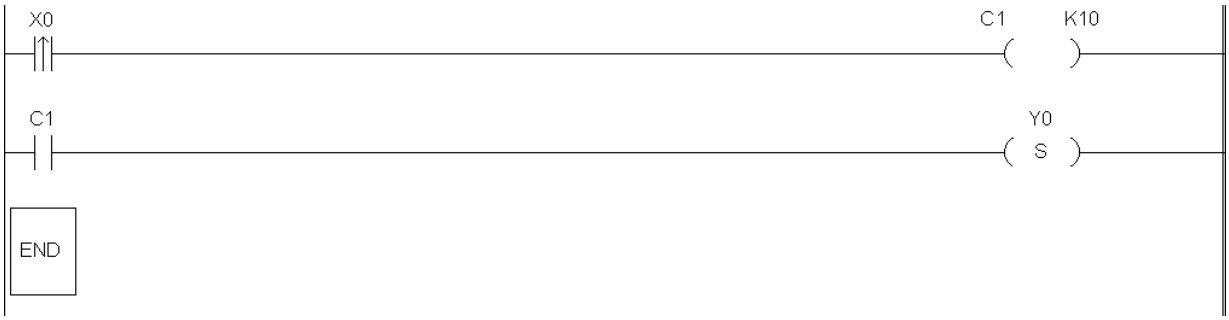
نحوه استفاده از کانترها:

کانترها یا شمارنده ها برای شمارش تعداد دفعات به کار می روند.کانترها باC شناخته می شوند و طریقه کارکرد آنها مشابه تایمرها می باشد.کانترها مشابه تایمرها هم بیت دارند و هم رجیستر. برای استفاده از کانتر باید از دستور OUT استفاده کنید، بدین منظور OUT را تایپ کرده ، یک جای خالی گذاشته، شماره کانتر(مثلا C1) را نوشته ، مجددا یک جای خالی و در نهایت تعداد دفعات مورد نظر خود را بنویسید.

۳۰۰کانتر معمولی وجود دارد که از C0 شروع می شوند و تا C299 ادامه پیدا می کنند.

مثال:

فرض کنید ورودی X0 به یک سنسور سوییچینگ وصل است که با هر بار دیدن محصول، ورودی را روشن می کند. می خواهیم برنامه ای بنویسیم که بعد از تولید ۱۰ محصول خروجی Y0 را روشن کند.



نحوه خواندن دما با ترموکوپل:

ترموکوپل ها جزو ورودی های آنالوگ حساب می شوند. ترموکوپل ها برای خواندن دما از ۰ تا ۱۰۰۰ درجه سانتیگراد مورد استفاده قرار می گیرند. پی ال سی های تینگت دارای ما ژول مختلفی برای خواندن دما توسط ترموکوپل می باشد.

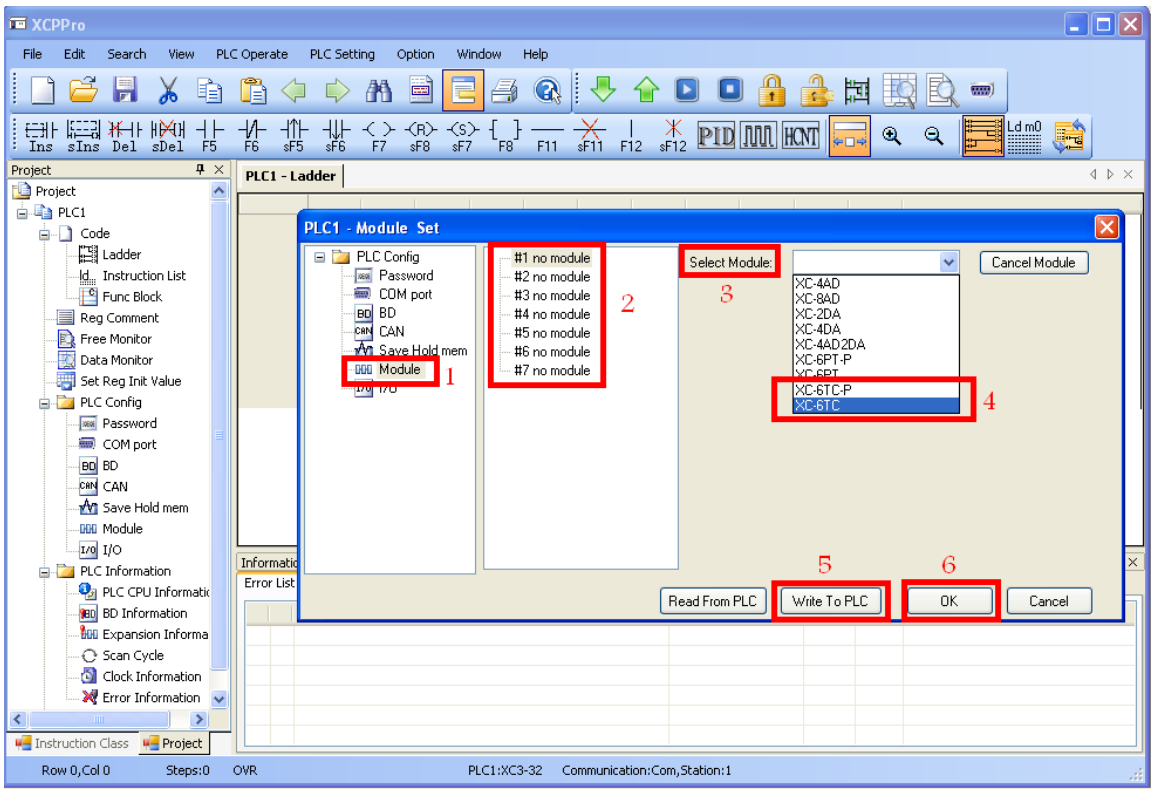
XC-6TC : این ماژول ۶ ترموکوپل تیپ K را می خواند.

XC-6TC-P: این ماژول ۶ ترموکوپل تیپ K را می خواند و برر روی خودش PID کنترل دارد و شش خروجی دارد که می تواند از آنها برای سیستم های کنترلی استفاده کرد.

XC-6TC-AP: این ماژول ۶ ترموکوپل (تمامی تیپ ها) را می خواند و برر روی خودش PID کنترل دارد و شش خروجی دارد که می تواند از آنها برای سیستم های کنترلی استفاده کرد.

برای خواندن ترموکوپل در ابتدا باید ماژول را به پی ال سی معرفی کنیم. . برای این کار مراحل زیر را انجام دهید:

۱- بر روی گزینه ماژول کلیک کنید .
۲- محل قرار گرفتن ماژول را مشخص کنید.
۳- منوی Select Model را باز کنید.
۴- مدل ماژول خود را انتخاب کنید.
۵- گزینه Write to PLC را انتخاب کنید.
۶- گزینه OK را انتخاب کنید.
۷- یک بار برق پی ال سی را قطع کرده و پس از چند لحظه مجددا وصل کنید.



اکنون کافیست رجیستر مربوط به ورودی مورد نظرتان را بخوانید. این رجیسترها با ID شناخته می شوند و شماره آن با توجه به محل قرار گرفتن ماژول مشخص می شود. فرض کنید ماژولتان در محل شماره ۱ قرار گرفته است. آدرس ها از ۱۰۰ شروع می شوند.یعنی اگر ترموکوپلتان را به پایه های +TC0-,TC0 (اولین کانال ورودی ماژول) وصل کرده باشید می بایست خانه ID100 را بخوانید. به همین ترتیب اگر به +TC1-,TC1 وصل کرده باشید ID101، +TC2-,TC2 خانه ID102 ... .اگر ماژول در جایگاه دوم بود آدرس ها از ۲۰۰ شروع می شوند، اگر ماژول در جایگاه سوم بود از ۳۰۰ ... .

بعد از مشخص شدن خانه مربوط به ورودی ترموکوپل، برای مقایسه و یا اعمال هرگونه محاسباتی آن را در یکی از رجیسترهای حافظه پی ال سی کپی کنید. برای این کار از دستور MOV استفاده کنید.

****برای دستور بالا از شرط کنتاکت باز M8000 استفاده کرده ایم. دلیل آن این است که این بیت از حافظه همیشه روشن است و در نتیجه دستوری که در مقابل آن نوشته شود همیشه اجرا خواهد شد.**

اگر از چندین ترموکوپل استفاده می کنید می توانید برای خواندن همزمان آنها از دستور BMOV استفاده کنید.

نحوه خواندن دما با PT100:

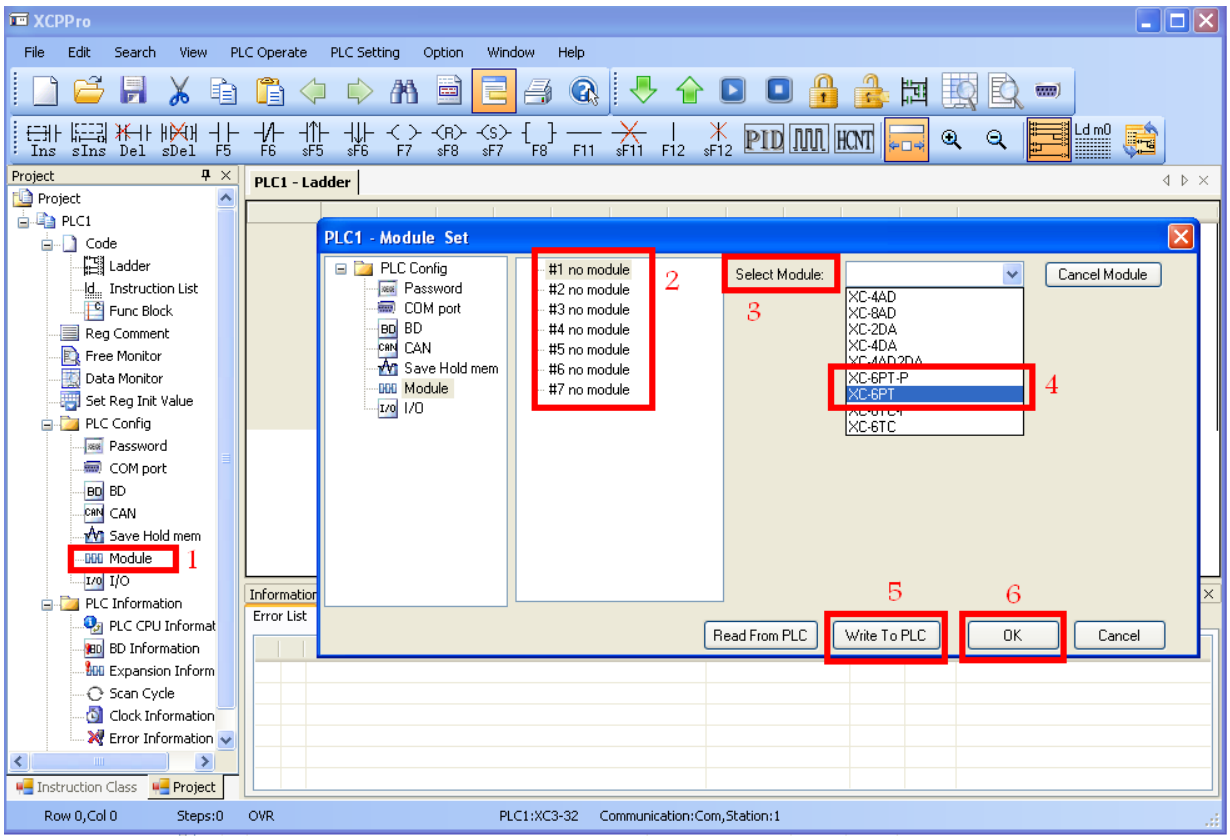
PT100 ها جزو ورودی های آنالوگ حساب می شوند. PT100 ها برای خواندن دما از ۱۰۰- تا ۳۵۰ درجه سانتیگراد مورد استفاده قرار می گیرند. پی ال سی های تینگت دارای ماژول های مختلفی برای خواندن دما توسط PT100 می باشد.

XC-6PT : این ماژول ۶ عدد PT100 را می خواند.

XC-6PT-P: این ماژول ۶ عدد PT100 را می خواند و برر روی خودش PID کنترل دارد و شش خروجی دارد که می تواند از آنها برای سیستم های کنترلی استفاده کرد.

برای خواندن PT100 در ابتدا باید ماژول را به پی ال سی معرفی کنیم. . برای این کار مراحل زیر را انجام دهید:

۱- بر روی گزینه ماژول کلیک کنید .
۲- محل قرار گرفتن ماژول را مشخص کنید.
۳- منوی Select Model را باز کنید.
۴- مدل ماژول خود را انتخاب کنید.
۵- گزینه Write to PLC را انتخاب کنید.
۶- گزینه OK را انتخاب کنید.
۷- یک بار برق پی ال سی را قطع کرده و پس از چند لحظه مجددا وصل کنید.



اکنون کافیست رجیستر مربوط به ورودی مورد نظرتان را بخوانید. این رجیسترها با ID شناخته می شوند و شماره آن با توجه به محل قرار گرفتن ماژول مشخص می شود. فرض کنید ماژولتان در محل شماره ۱ قرار گرفته است. آدرس ها از ۱۰۰ شروع می شوند.یعنی اگر PT100 را به پایه های A0,C0 (اولین کانال ورودی ماژول) وصل کرده باشید می بایست خانه ID100 را بخوانید. به همین ترتیب اگر به A1,C1 وصل کرده باشید ID101، A2,C2 خانه ID102 ... .اگر ماژول در جایگاه دوم بود آدرس ها از ۲۰۰ شروع می شوند، اگر ماژول در جایگاه سوم بود از ۳۰۰ ... .

بعد از مشخص شدن خانه مربوط به ورودی ترموکوپل، برای مقایسه و یا اعمال هرگونه محاسباتی آن را در یکی از رجیسترهای حافظه پی ال سی کپی کنید. برای این کار از دستور MOV استفاده کنید.

** **برای دستور بالا از شرط کنتاکت باز M8000 استفاده کرده ایم. دلیل آن اینست که این بیت از حافظه همیشه روشن است و در نتیجه دستوری که در مقابل آن نوشته شود همیشه اجرا خواهد شد.**

اگر از چندین PT100 استفاده می کنید می توانید برای خواندن همزمان آنها از دستور BMOV استفاده کنید.

نحوه خواندن مقادیر آنالوگ(ولتاژ یا جریان):

بعضی از وسایل اندازه گیری مانند خط کش ها ولتاژ یا جریان می دهند که با خواندن این ولتاژ یا جریان می توان از وضعیت دستگاه با خبر شد و با توجه به آن تصمیمات مناسب گرفت. بعضی از وسایل کنترلی فقط ولتاژ، بعضی فقط جریان و بعضی هم ولتاژ و هم جریان می دهند. مقادیر استاندارد ولتاژ، 5~0 و 10~0 ولت و مقادیر استاندارد جریان، 20~0 و 20~4 میلی آمپر می باشند.همان طور که می دانید ماژول های مختلف دارای دقت های متفاوت می باشند و بنابراین عددی که برای مقادیر ولتاژ و یا جریان می دهند متفاوت است. در پی ال سی های تینگت، ماژول های ورودی آنالوگ، ولتاژ و جریان را تبدیل به عددی بین ۰ تا ۱۶۳۸۳ می کنند. مثلا فرض کنید که آنالوگ ما ولتاژ10~0 ولت باشد، در این صورت اگر به ماژول ۰ ولت بدهید عدد ۰، اگر ۵ ولت بدهید حدودا عدد ۸۰۰۰، و اگر ۱۰ ولت بدهید عدد ۱۶۳۸۳ را می دهد. پی ال سی های تینگت دارای ما ژول های مختلفی برای خواندن مقادیر آنالوگ می باشد.

XC-8AD: این ماژول ۸ عدد ورودی آنالوگ را می خواند که چهار تای اول آن برای جریان و چهارتای دوم آن برای ولتاژ می باشد.

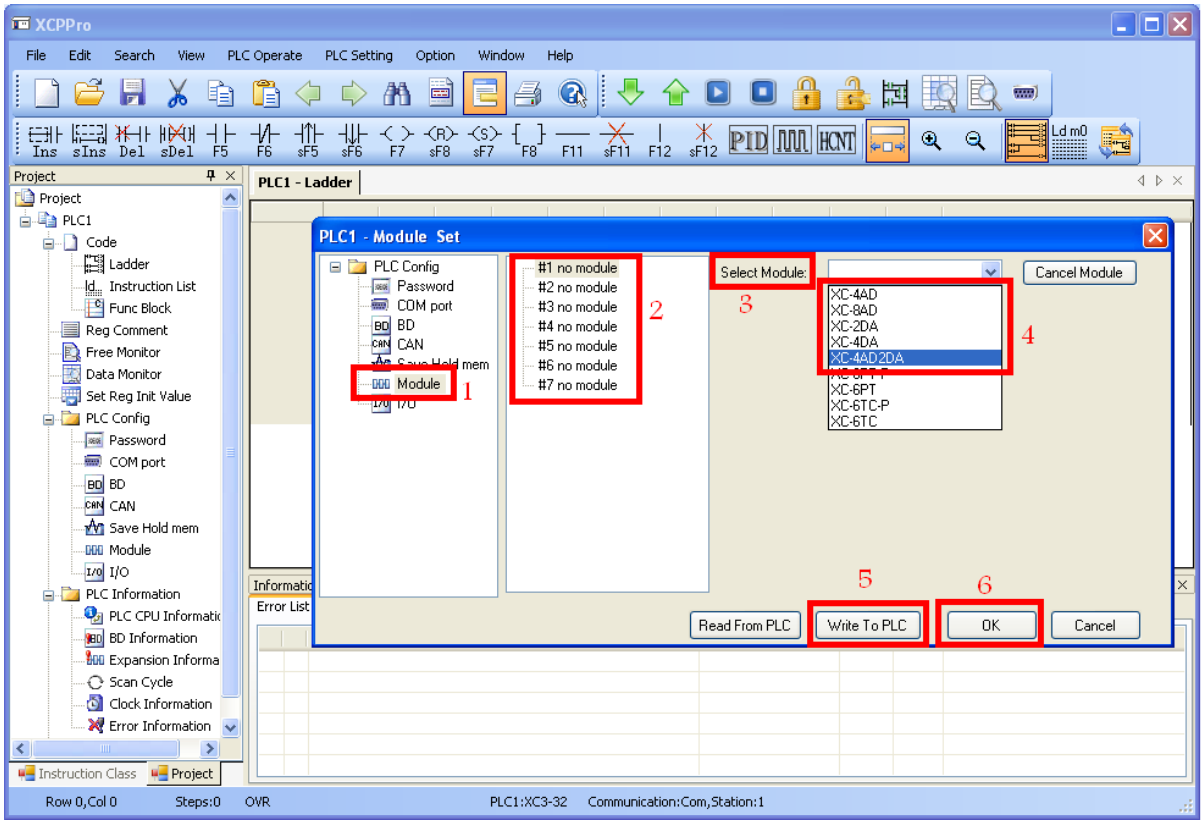
XC-4AD: این ماژول 4 عدد ورودی آنالوگ، هم ولتاژ و هم جریان را می خواند.

XC-4AD2DA : این ماژول 4 عدد ورودی آنالوگ، هم ولتاژ و هم جریان را می خواند. علاوه بر راین چهار ورودی آنالوگ، این ماژول دو خروجی آنالوگ هم دارد.

XC-3AD4PT2DA: این ماژول 3 عدد ورودی آنالوگ، هم ولتاژ و هم جریان را می خواند. علاوه بر این سه ورودی آنالوگ، این ماژول ۴ عدد PT100 را نیز می خواند و دو خروجی آنالوگ هم دارد.

برای خواندن مقادیر آنالوگ در ابتدا باید ماژول را به پی ال سی معرفی کنیم. . برای این کار مراحل زیر را انجام دهید:

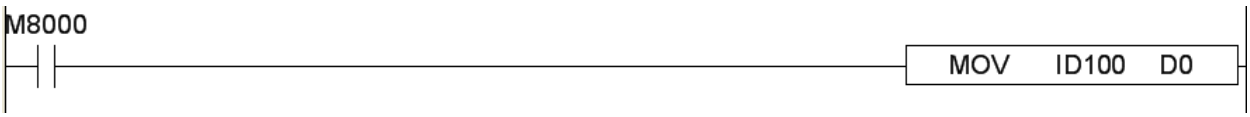
۱- بر روی گزینه ماژول کلیک کنید .
۲- محل قرار گرفتن ماژول را مشخص کنید.
۳- منوی Select Model را باز کنید.
۴- مدل ماژول خود را انتخاب کنید. پس از انتخاب مدل، ولتاژ ز یا جریان بودن و همچنین رنج آن را تعیین کنید.
۵- گزینه Write to PLC را انتخاب کنید.
۶- گزینه OK را انتخاب کنید.
۷- یک بار برق پی ال سی را قطع کرده و پس از چند لحظه مجددا وصل کنید.



اگر از ورودی ولتاژ استفاده می کنید سیم های آن را به پایه های VI,CI و اگر از ورودی جریان استفاده می کنید، سیم های آن را به پایه های AI,CI وصل کنید.

اکنون کافیست رجیستر مربوط به ورودی مورد نظرتان را بخوانید. این رجیسترها با ID شناخته می شوند و شماره آن با توجه به محل قرار گرفتن ماژول مشخص می شود. فرض کنید ماژولتان در محل شماره ۱ قرار گرفته است. آدرس ها از ۱۰۰ شروع می شوند.یعنی اگرورودی آنالوگ را به پایه های AI0(VI0),CI0 (اولین کانال ورودی ماژول) وصل کرده باشید می بایست خانه ID100 را بخوانید. به همین ترتیب اگر به AI1(VI1),CI1 وصل کرده باشید ID101، AI2(VI2),CI2خانه ID102 ... . اگر ماژول در جایگاه دوم بود آدرس ها از ۲۰۰ شروع می شوند، اگر ماژول در جایگاه سوم بود از ۳۰۰ ... .

بعد از مشخص شدن خانه مربوط به ورودی آنالوگ، برای مقایسه و یا اعمال هرگونه محاسباتی آن را در یکی از رجیسترهای حافظه پی ال سی کپی کنید. برای این کار از دستور MOV استفاده کنید.



** برای دستور بالا از شرط کنتاکت باز M8000 استفاده کرده ایم. دلیل آن اینست که این بیت از حافظه همیشه روشن است و در نتیجه دستوری که در مقابل آن نوشته شود همیشه اجرا خواهد شد.

اگر از چندین ورودی آنالوگ استفاده می کنید می توانید برای خواندن همزمان آنها از دستور BMOV استفاده کنید.

نحوه تولید خروجی آنالوگ:

بعضی از وسایل کنترلی مانند بعضی از شیرها و بعضی از اینورترها هستند که برای کنترل آنها باید یک ولتاژ و یا جریان به آنها داد و با تغییر این ولتاژ یا جریان خروجی مطلوب را از آنها گرفت. مقادیر استاندارد ولتاژ، 0~5 و 0~10 ولت و مقادیر استاندارد جریان، 0~20 و 4~20 میلی آمپر می باشند در پی ال سی های تینگت، ماژول های خروجی آنالوگ، عدد ۰ تا ۴۰۹۵ را به ولتاژ و جریان تبدیل به می کنند.مثلا فرض کنید می خواهیم ولتاژ خروجی 0~10 ولت داشته باشیم.اگر عدد ۰ را در خروجی بریزیم ماژول ۰ ولت خروجی می دهد، اگر حدود ۲۰۰۰را در خروجی بریزیم، ۵ ولت و اگر ۴۰۹۵ را در خروجی بریزیم ۱۰ ولت خروجی به ما می دهد.به همین ترتیب می توانید برای مقدار خروجی مورد نظر خود،عدد مربوطه را پیدا کرده و آن را در خروجی بریزید. پی ال سی های تینگت دارای ما ژول های مختلفی برای تولید مقادیر آنالوگ می باشد.

XC-2DA: این ماژول دارای ۲ خروجی آنالوگ ولتاژ و جریان می باشد.

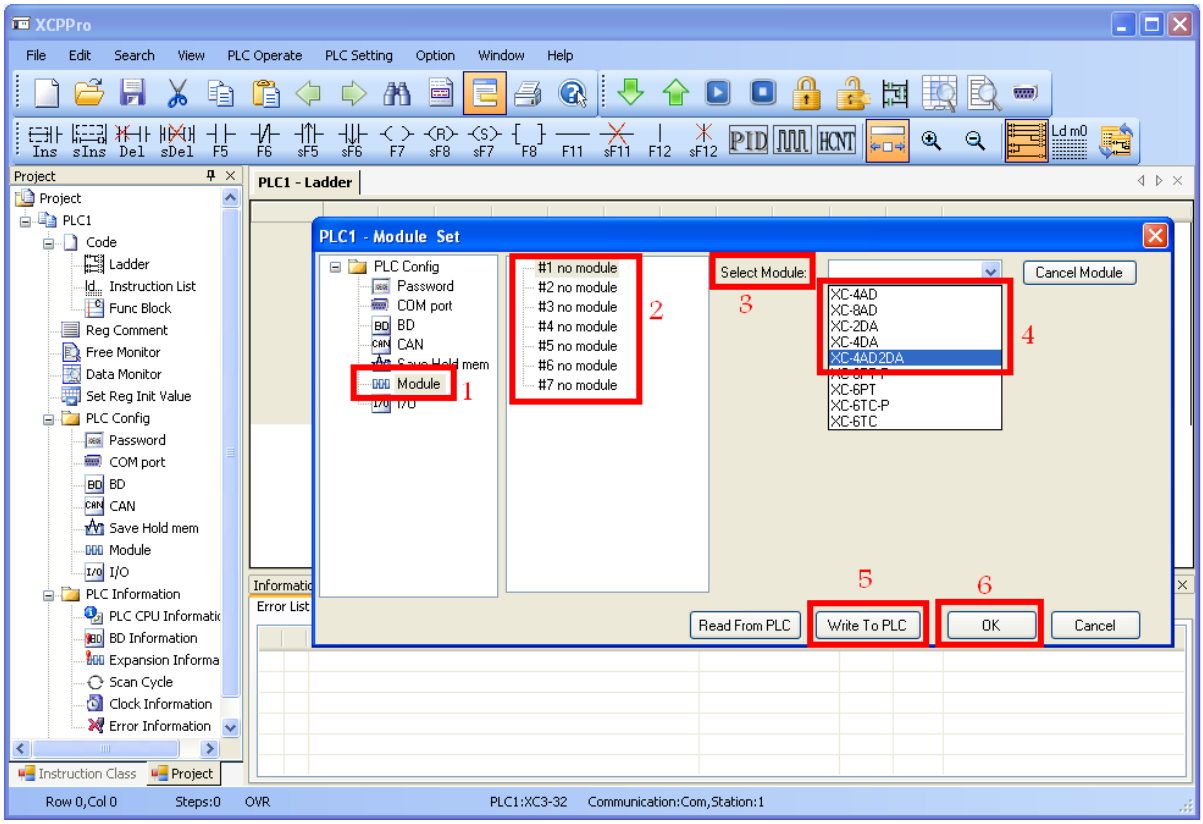
XC-4DA: این ماژول دارای ۴ خروجی آنالوگ ولتاژ و جریان می باشد.

XC-4AD2DA: این ماژول دارای ۲ خروجی آنالوگ ولتاژ و جریان به همراه ۴ ورودی آنالوگ می باشد.

XC-3AD4PT2DA: این ماژول. علاوه بر دو خروجی آنالوگ ،هم ۴ عدد PT100 را نیز می خواند و هم 3 عدد ورودی آنالوگ، هم ولتاژ و هم جریان را می خواند.

برای تولید مقادیر آنالوگ در ابتدا باید ماژول را به پی ال سی معرفی کنیم. . برای این کار مراحل زیر را انجام دهید:

۱- بر روی گزینه ماژول کلیک کنید .
۲- محل قرار گرفتن ماژول را مشخص کنید.
۳- منوی Select Model را باز کنید.
۴- مدل ماژول خود را انتخاب کنید. پس از انتخاب مدل، ولتاژ یا جریان بودن و همچنین رنج آن را تعیین کنید.
۵- گزینه Write to PLC را انتخاب کنید.
۶- گزینه OK را انتخاب کنید.
۷- یک بار برق پی ال سی را قطع کرده و پس از چند لحظه مجددا وصل کنید.



اگر از خروجی ولتاژ استفاده می کنید سیم های آن را به پایه های VO,CO و اگر از ورودی جریان استفاده می کنید، سیم های آن را به پایه های AO,CO وصل کنید.

اکنون کافیست عدد متناظر با مقدار خروجی مورد نظر خود را در رجیستر مربوط به خروجی مورد نظرتان بریزید. این رجیسترها با QD شناخته می شوند و شماره آن با توجه به محل قرار گرفتن ماژول مشخص می شود. فرض کنید ماژولتان در محل شماره ۱ قرار گرفته است. آدرس ها از ۱۰۰ شروع می شوند.یعنی اگرخروجی آنالوگ را به پایه های AO0(VO0),CO0 (اولین کانال ورودی ماژول) وصل کرده باشید می بایست خانه QD100 را بخوانید. به همین ترتیب اگر به AO1(VO1),CO1 وصل کرده باشید QD101، AO2(VO2),CO2 خانه QD102 ... . اگر ماژول در جایگاه دوم بود آدرس ها از ۲۰۰ شروع می شوند، اگر ماژول در جایگاه سوم بود از ۳۰۰ ... .

در مثال فوق محتوای رجیستر D0 در QD100 که مربوط به اولین خروجی آنالوگ می باشد ریخته می شود و ماژول با توجه به عددی که در داخل QD100 ریخته شده است ولتاژ و یا جریان تولید می کند.

نحوه خواندن انکدر با پی ال سی های تینگت

انکدر هایی که دارای خروجی پالس هستند ۳ نوع می باشند، پالس، پالس و جهت و حالت A,B فاز. پی ال سی های تینگت این قابلیت را دارند که هر سه نوع انکدرها را بخوانند. برای خواندن انکدر ها باید اول با توجه به نوع انکدر پایه ورودی مربوطه و شماره کانتر آن را پیدا کنید و با شرط کنتاکت باز M8000 کانتر را فعال کنیم.

در مدل های XC3-48,XC-60

XC3-48、XC3-60																		
	Increment mode										Pulse+direction mode					AB phase mode		
	C600	C602	C604	C606	C608	C610	C612	C614	C616	C618	C620	C622	C624	C626	C628	C630	C632	C634
Highest frequency	80K	80K	10K	10K							80K	80K				80K	80K	
4-times frequency																	√	
interruption	√	√	√	√								√					√	
X000	U										U					A		
X001											Dir					B		
X002		U										U					A	
X003												Dir					B	
X004			U															
X005				U														

در مدل XC3-32

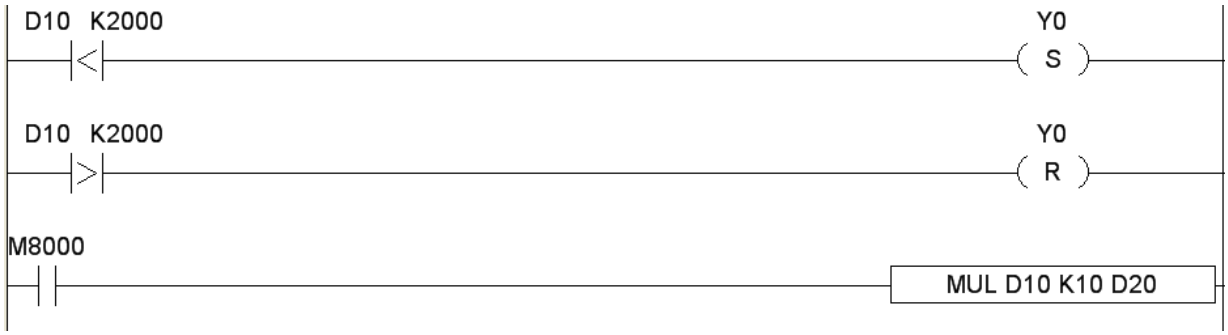
XC3-24、XC3-32, XC5-48、XC5-60																		
	Increment mode										Pulse+direction mode					AB phase mode		
	C600	C602	C604	C606	C608	C610	C612	C614	C616	C618	C620	C622	C624	C626	C628	C630	C632	C634
Highest frequency	80K	10K	10K	10K	10K	10K					80K	10K	10K			80K	5K	5K
4-times frequency																√		√
interruption	√	√	√	√	√	√					√					√		
X000	U										U					A		
X001		U									Dir					B		
X002																		
X003			U									U					A	
X004												Dir					B	
X005																		
X006				U									U					A
X007													Dir					B
X010																		
X011					U													
X012						U												

برای مثال فرض کنید می خواهید یک انکدر A,B فاز را به پی ال سی ۳۲ ورودی خروجی وصل کنید.بدین منظور باید به جدول دوم به قسمت مربوط به A,B فاز مراجعه کنید. همان طور که در جدول فوق می بینید این مدل دارای ۳ کانتر A,B فاز است، که برای مثال برای استفاده از اولین کانتر باید انکدر را به پایه های X0,X1 وصل کنید و از کانتر شماره C630 استفاده نمایید.

برای فعال کردن کانتر از شرط کنتاکت باز M8000 استفاده کنید. زیرا این بیت همیشه روشن است و باعث می شود تا انکدر همیشه کار کند. برای فعال کردن کانتر، OUT را تایپ کنید، یک جای خالی بگذارید، شماره کانتر مورد نظر خود (مثلاC630) را بنویسید مجددا یک جای خالی بگذارید و در نهایت عدد k2147483648 راتایپ کنید. دلیل استفاده از این عدد اینست که کانترهای انکدر، ۳۲ بیتی می باشند و ماکزیمم تا این عدد می شمارند و درنتیجه برای راحتی کار از این عدد استفاده می کنیم.

در این مثال عدد شمارش شده در رجیستر C630 ریخته می شود، برای مقایسه و یا هرگونه عمل ریاضی عدد داخل آن را در یک رجیستر از حافظه پی ال سی کپی کنید.

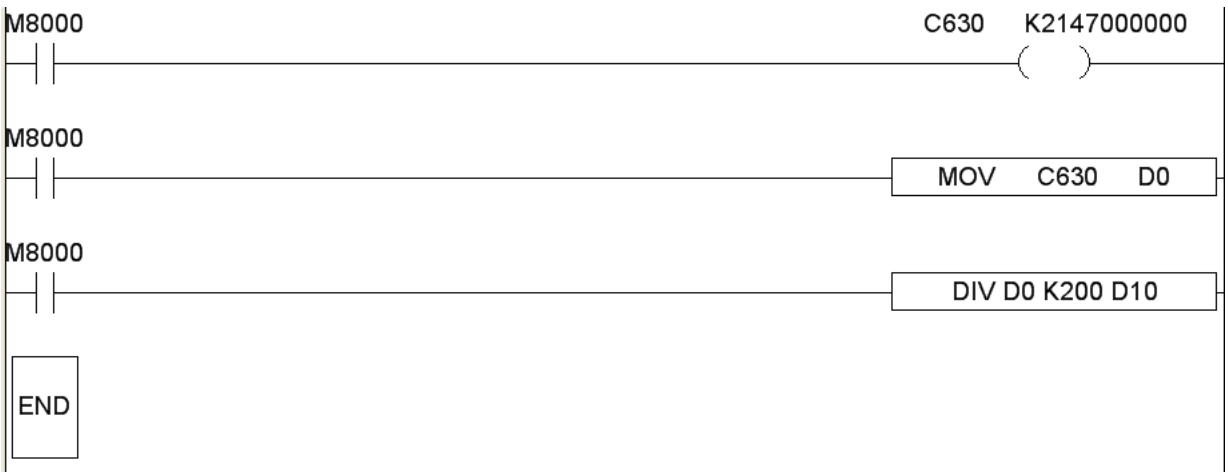
حال می توانید مقدار خوانده شده را مقایسه کرده و یا اعمال ریاضی را بر روی آن انجام دهید.



در مثال فوق در خط اول برنامه اگر تعداد پالس هایی که خوانده شده است از ۲۰۰۰ تا کمتر باشد خروجی Y0 روشن می شود ودر خط دوم اگر تعداد پالس ها از ۲۰۰۰ تا بیشتر شود خروجی Y0 خاموش می شود. در خط سوم برنامه تعداد پالس های شمارش شده در ۱۰ ضرب می شود و حاصل آن در D20 ریخته می شود.

مثال:فرض کنید انکدری داریم A,B فاز که به ازای هر ۱سانتیمتر ۲۰۰ پالس می دهد. مقدار جابه جایی را حساب کنید.

فرض می کنیم انکدر را به X0,X1 وصل کرده ایم. از C630 برای خواندن انکدر استفاده می کنیم.



در خط اول برنامه کانتر را فعال کرده ایم. در خط دوم مقدار شمارش شده را به D0 منتقل کرده ایم تا بتوانیم محاسباتمان را بر روی آن انجام دهیم. در خط سوم تعداد پالس های شمارش شده را بر ۲۰۰ تقسیم کرده ایم تا جابه جایی را بر حسب سانتیمتر به دست آوریم. مقدار جابه جایی بر حسب سانتیمتر در D10 ریخته می شود.

نحوه تولید پالس

برای تولید پالس باید از پی ال سی هایی استفاده کنید که دارای خروجی ترانزیستوری باشند. بدین منظور هم می توانید از مدل هایی که تمامی خروجی آنها ترانزیستور است استفاده کنید و هم از مدل هایی که دو خروجی اول آنها ترانزیستور و ما بقی رله ایست.

برای تولید پالس دستورات مختلفی وجود دارد که در اینجا دستور PLSR را توضیح می دهیم. در این دستور پی ال سی پالس را به طریقه "پالس و جهت" تولید می کند. این دستور این قابلیت را دارد که ۲۴ پالس را پشت سر هم تولید کند. برای تولید پالس شما باید تعداد، فرکانس و acc/dec را تعیین کنید. Acc/dec مدت زمانی است که تولید پالس به ماکزیمم فرکانس خود می رسد.

در این دستور شما باید یک رجیستر را به عنوان رجیستر کنترلی انتخاب کنید که در داخل این رجیستر تعداد پالس مورد نظر و به طور اتوماتیک در رجیستر بعدی فرکانس را باید بریزید. همچنین دو رجیستر بعدی باید حتما دارای مقدار صفر باشند. برای تولید پالس به طریقه پالس و جهت شما نیاز به دو خروجی دارید. خروجی اول مربوط به تولید پالس می باشد که حتما یا باید Y0 باشد و یا Y1، و خروجی دوم برای کنترل جهت می باشد.

برای نوشتن دستور PLSR را تایپ کنید، یک جای خالی گذاشته، آدرس رجیستر کنترلی، یک جای خالی، acc/dec ، یک جای خالی خروجی تولید پالس، مجددا یک جای خالی و در نهایت خروجی جهت را تایپ کنید. توجه داشته باشید که برای عملگر اول(رجیستر کنترلی) حتما باید از رجیستر استفاده کنید ولی برای acc/dec هم می توانید از رجیستر و هم از عدد ثابت استفاده کنید.

```
PLSR    D0    K100    Y0    Y2
```

در این دستور داریم:

PLSR : دستور تولید پالس

D0 : آدرس رجیستر کنترلی

acc/dec : K100

Y0 : خروجی تولید پالس

Y1 : خروجی کنترل جهت

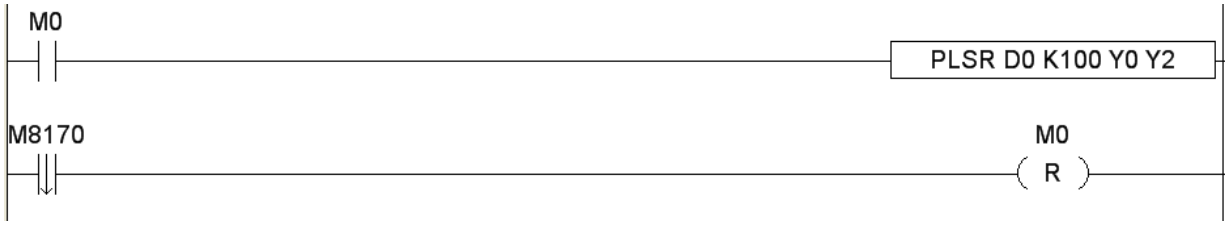
در مثال فوق در رجیستر D0 تعداد پالس مورد نظر و در D1 فرکانس را بریزید و در D2,D3 حتما صفر بریزید.

برای فعال کردن دستور باید از یک بیت به عنوان شرط استفاده کنید که با فعال شدن شرط دستور شروع به تولید پالس کند و اگر شرط غیر فعال شود تولید پالس نیز متوقف خواهد شد.

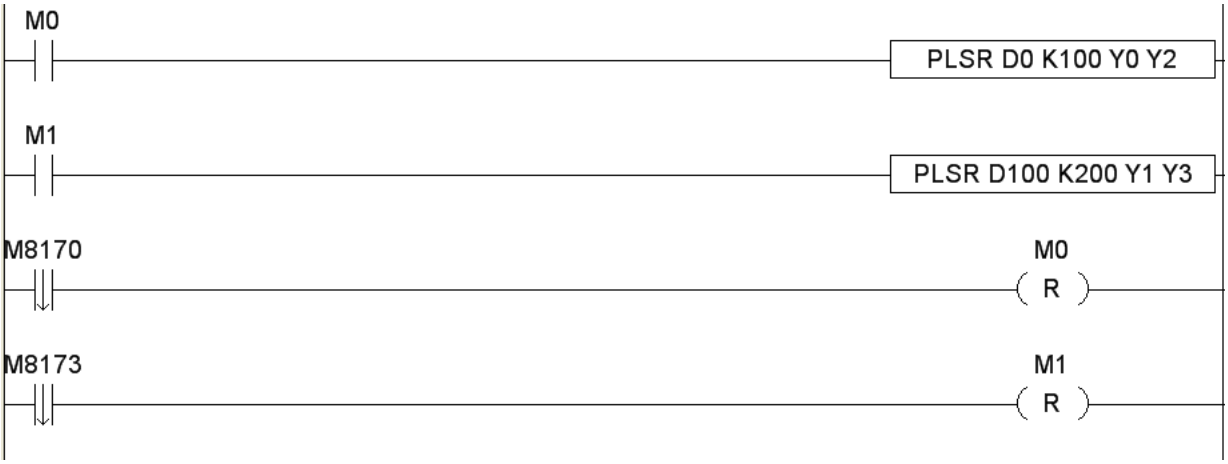
هرکدام از خروجی هایی که برای تولید پالس استفاده می کنید دارای یک بیت ویژه برای کنترل هستند. این بیت تا زمانی که پی ال سی در حال تولید پالس می باشد ست بوده و با تمام شدن تولید پالس ریست می شوند. در واقع شما با مشاهده این بیت می توانید از زمان پایان یافتن تولید پالس مطلع شوید. برای Y0 این بیت M8170 برای Y1 این بیت M8173 می باشد.

از طرفی این دو خروجی دارای رجیسترهای ویژه ای نیز هستند که تعداد پالس تولید شده را به شما می دهند. هرکدام از این خروجی ها دارای ۲ رجیستر برای نشان دادن این مقادیر هستند. یک رجیستر کل پالس های تولید شده را به شما می دهد و یک رجیستر فقط تعداد تولید شده در همان اجرا را به شما می دهد. برای مثال فرض کنید شما در هر بار تولید پالس، ۲۰۰۰۰ پالس تولید می کنید. به طور مثال اگر ۳ بار پالس تولید کرده باشید در رجیستری که کل پالس ها را نشان می دهد عدد ۶۰۰۰۰ و در رجیستر دیگر عدد ۲۰۰۰۰ را خواهید داشت. حال اگر مجددا تولید پالس را شروع کنید رجیستری که کل را نشان می دهد از ۶۰۰۰۰ شروع به شمارش کرده و تا ۸۰۰۰۰ می رود ولی رجیستر دیگر از ۰ شروع کرده و تا ۲۰۰۰۰ می شمارد. آدرس رجیستری که کل پالس ها را نشان می دهد برای خروجی اول D8170 و برای خروجی دوم D8173 و آدرس رجیستری که تعداد پالس همان اجرا را می دهد برای خروجی اول D8190 و برای خروجی دوم D8193 می شود.

مثال:

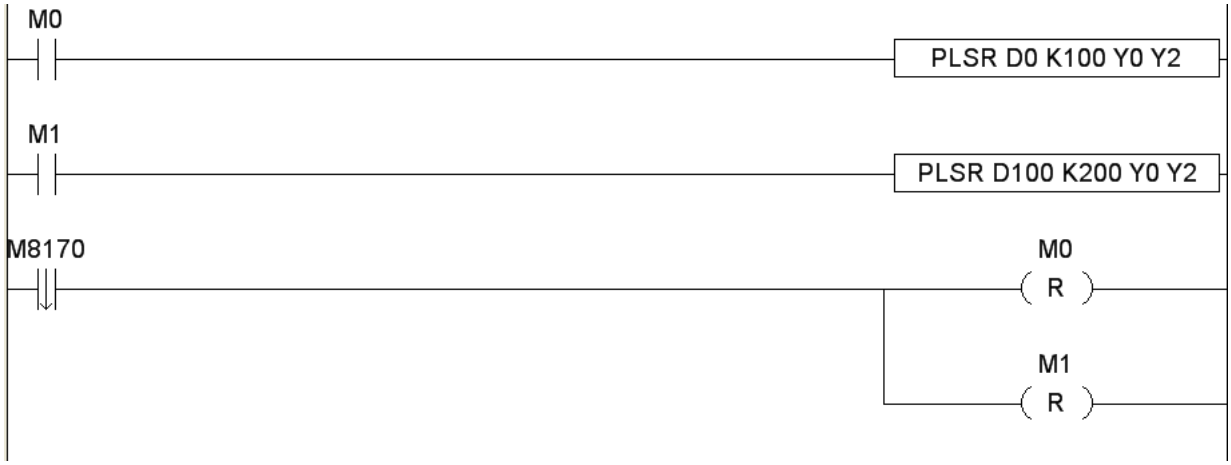


در مثال فوق با ست شدن M0 پی ال سی بر روی خروجی Y0 شروع به تولید پالس کرده و زمانی که تولید پالس تمام شود M8170 ریست شده و با ریست شدن آن ،M0 نیز ریست می شود که برنامه مجددا برای تولید پالس آماده می شود. پس از اتمام تولید پالس می توانید مجددا آن را با همان مقادیر قبلی فعال کرده و یا تعداد و فرکانس پالس را عوض کرده وسپس تولید پالس را شروع کنید.



در مثال فوق اگر M0,M1 همزمان با هم ست شوند پی ال سی دو پالس خروجی را با هم تولید می کند.

در هنگام برنامه نویسی دقت فرمایید که برای یک خروجی فقط یک دستور می توانید استفاده کنید و اگر ۲ دستور تولید پالس را با یک خروجی بنویسید حتی اگر شرط یکی از آنها فعال نباشد پی ال سی پالس تولید نخواهد کرد.



در این مثال پی ال سی نه با ست شدن M0 و نه با روشن شدن M1 پالس تولید نخواهد کرد.

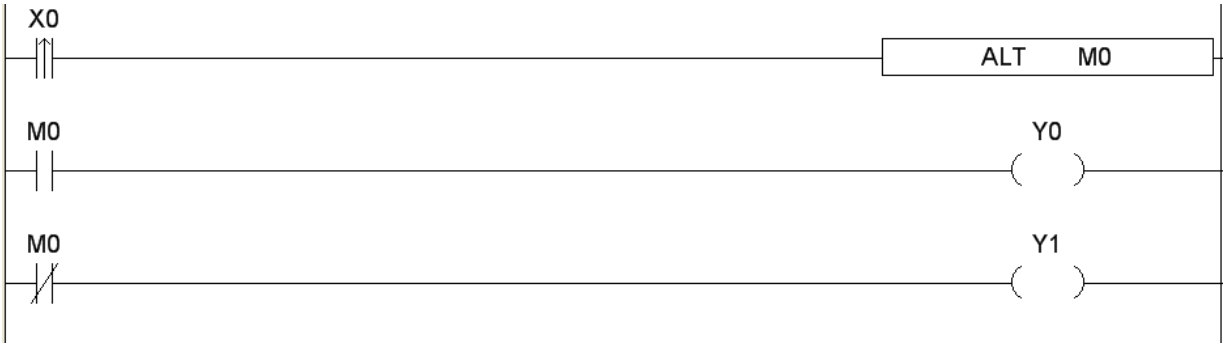
توضیح برخی از دستورات:

ALT:

این دستور عمل معکوس کردن را بر روی بیت ها انجام می دهد. با هر باری که این دستور انجام شود بیتی که به عنوان عملگر انتخاب کرده اید وضعیتش برعکس می شود، یعنی اگر ست بوده ،ریست شده و اگر قبلا ریست بوده با اعمال این دستور ست می شود.

برای نوشتن دستور ALT را تایپ کرده، یک جای خالی گذاشته و عملگر را تایپ کنید.

مثال:

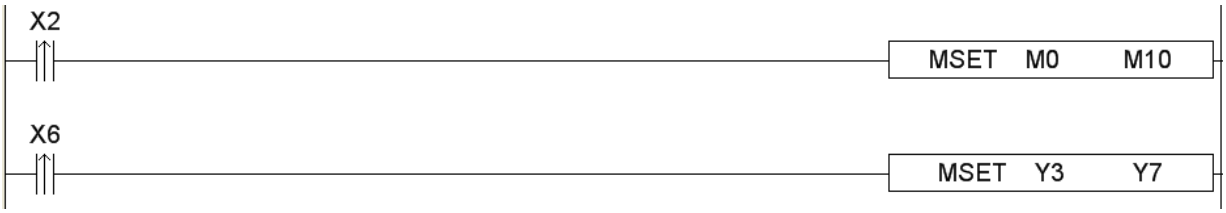


در مثال فوق با هر بار روشن شدن X0 ،دستور ALT انجام شده و باعث معکوس شدن وضعیت M0 می شود. فرض کنید M0 خاموش باشد که در نتیجه باعث خاموش شدن Y0 و روشن شدن Y1 می شود. حال X0 راروشن می کنیم، با این کار دستور ALT انجام شده و M0 روشن شده و در نتیجه Y0 روشن و Y1 خاموش می شود.حال اگر X0 خاموش شده و مجددا روشن شود، دستور ALT دوباره اجرا شده و این بار M0 را خاموش می کند که در نتیجه Y0 خاموش و Y1 روشن می شود.

MSET:

از این دستور برای ست کردن چند بیت پشت سر هم با یک دستور استفاده می شود. از این دستور هم برای خروجی ها و هم برای حافظه های داخلی پی ال سی (M) می توانید استفاده کنید.

برای نوشتن دستوز MSET را تایپ کنید،یک جای خالی گذاشته، آدرس اولین عملگر را تایپ کنید، مجددا یک جای خالی و در نهایت آدرس آخرین عملگر را بنویسید.

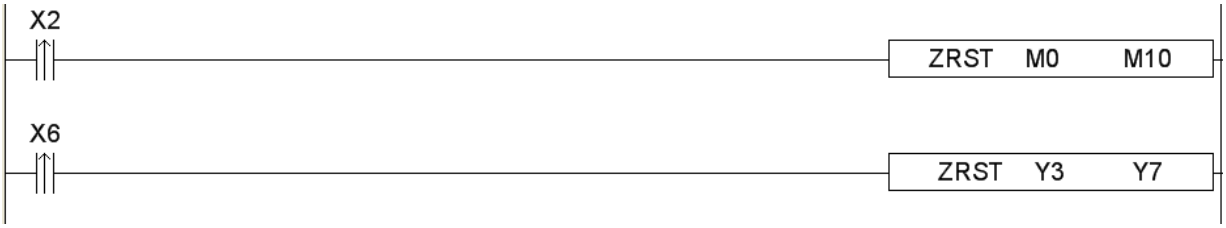


در مثال فوق در خط اول با ست شدن X2 دستور MSET اجرا شده و M0تا M10 همگی ست خواهند شد. در خط دوم نیز با ست شدن X6، از Y3 تا Y7 همگی ست خواهند شد.

:ZRST

از این دستور برای ریست کردن چند بیت پشت سر هم با یک دستور استفاده می شود. از این دستور هم برای خروجی ها و هم برای حافظه های داخلی پی ال سی (M) می توانید استفاده کنید.

برای نوشتن دستور ZRST را تایپ کنید، یک جای خالی گذاشته، آدرس اولین عملگر را تایپ کنید، مجددا یک جای خالی و در نهایت آدرس آخرین عملگر را بنویسید.

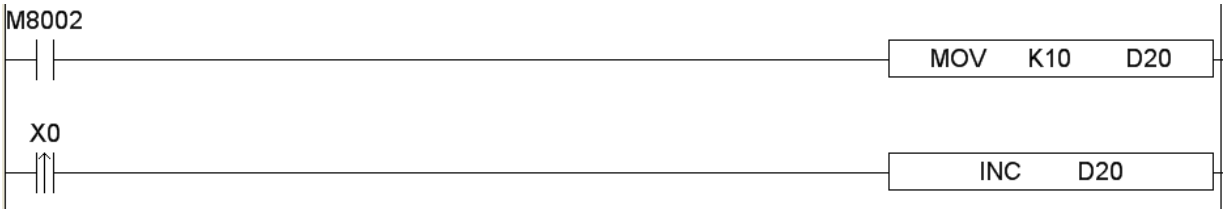


در مثال فوق در خط اول با ست شدن X2 دستور ZRST اجرا شده و M0تا M10 همگی ریست خواهند شد. در خط دوم نیز با ست شدن X6، از Y3 تا Y7 همگی ریست خواهند شد.

:INC

از این دستور برای اضافه کردن یک واحدی محتوای یک رجیستر استفاده می شود. با هر بار اجرا شدن این دستور محتوای رجیستری که به عنوان عملگر انتخاب شده است با یک جمع می شود.دقت فرمایید در این دستور برای عملگر حتما باید از رجیستر استفاده کنید.

برای نوشتن دستور INC را تایپ کنید، یک جای خالی گذاشته و سپس عملگر مورد نظر خود را تایپ کنید.

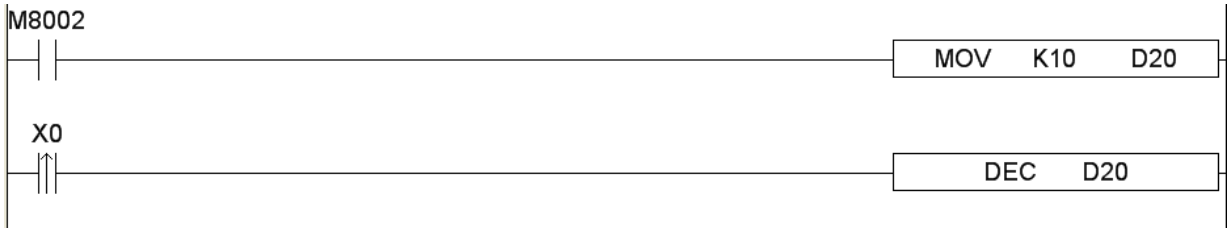


در مثال فوق با روشن شدن پی ال سی در خط اول برنامه، عدد ۱۰ داخل رجیستر D20 ریخته می شود. حال اگر X0 روشن شود دستور INC اجرا شده و عدد داخل D20 که عدد ۱۰ می باشد با ۱ جمع شده و درنتیجه عدد ۱۱ داخل درون D20 ریخته خواهد شد.

:DEC

از این دستور برای کم کردن یک واحدی محتوای یک رجیستر استفاده می شود. با هر بار اجرا شدن این دستور عدد ۱ از محتوای رجیستری که به عنوان عملگر انتخاب شده است کم می شود.دقت فرمایید در این دستور برای عملگر حتما باید از رجیستر استفاده کنید.

برای نوشتن دستور DEC را تایپ کنید، یک جای خالی گذاشته و سپس عملگر مورد نظر خود را تایپ کنید.



در مثال فوق با روشن شدن پی ال سی در خط اول برنامه، عدد ۱۰ داخل رجیستر D20 ریخته می شود. حال اگر X0 روشن شود دستور DEC اجرا شده و عدد ۱ از عدد داخل D20 که عدد ۱۰ می باشد کم شده و درنتیجه عدد ۹ داخل درون D20 ریخته خواهد شد.

:MEAN

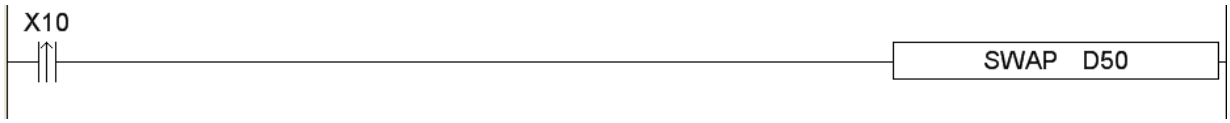
دستور عمل میانگین گرفتن را برروی چند رجیستر پشت سر هم انجام می دهد و حاصل را در رجیستر دیگری می ریزد.

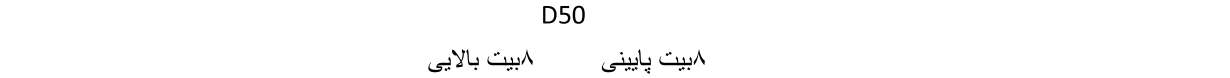
برای نوشتن این دستور MEAN را تایپ کنید، یک جای خال گذاشته، آدرس اولین رجیستر را نوشته، یک جای خالی، عملگر دوم را که می خواهید حاصل میانگیری در آن ریخته شود ، یک جای خالی و در نهایت تعداد رجیسترهای مورد نظر خود را تایپ کنید.

:SWAP

همان طور که می دانید در پی ال سی رجیستر ها ۱۶ بیتی می باشند. با استفاده از این دستور می توانید ۸ بیت پایینی یک رجیستر را با ۸ بیت بالایی آن عوض کنید.

برای نوشتن این دستور SWAP را تایپ کرده، یک جای خالی گذاشته و سپس عملگر را بنویسید. دقت کنید که عملگر باید حتما رجیستر باشد.





:XCH

این دستور محتوای دو رجیستر را با هم عوض می کند.

برای نوشتن این دستور XCH را تایپ کرده، یک جای خالی گذاشته و سپس عملگر اول، مجددا یک جای خالی و عملگر دوم را بنویسید. دقت کنید که عملگرها باید حتما رجیستر باشد.

در مثال فوق با ست شدن X10، دستور XCH اجرا شده و در نتیجه محتوای رجیسترهای D40 و D50 با هم عوض می شوند. برای مثال فرض کنید که قبل از اجرای دستور داخل D40 عدد ۱۰۰ و داخل D50 عدد ۲۰۰ موجود باشد. با انجام شدن دستور عدد ۱۰۰ درون D50 و عدد ۲۰۰ درون D40 ریخته می شوند.

NEG:

این دستور محتوای یک رجیستر را قرینه می کند و یا به عبارت دیگر در ۱- ضرب می کند.

برای نوشتن این دستور NEG را تایپ کرده، یک جای خالی گذاشته و سپس عملگر را بنویسید. دقت کنید که عملگر باید حتما رجیستر باشد.

در مثال فوفق با ست شدن M0، دستور NEG اجرا شده و در نتیجه محتوای رجیستر D20 ، قرینه می شود. فرض کنید قبل از اجرای دستور داخل D20 عدد ۵۰وجود داشته باشد، با اجرا شدن دستور عدد ۵۰در ۱- ضرب شده و درنتیجه عدد ۵۰- درون D20 ریخته می شود.

دستورات شیفت:

SHL: **شیفت ریاضی به چپ**

SHR: **شیفت ریاضی به راست**

LSL: **شیفت منطقی به چپ**

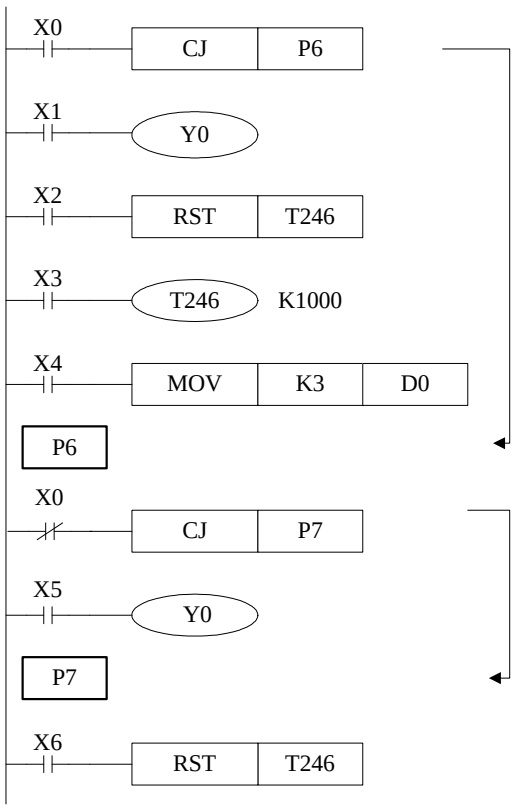
LSR: **شیفت منطقی به راست**

برای نوشتن دستور شیفتی، دستور مورد نظر خود را نوشته، یک جای خالی گذاشته؛ عملگر را تایپ کنید، مجددا یک جای خالی و در نهایت تعداد شیفت را تایپ کنید.

پرش شرطی یا CJ:

این دستور، دستور پرش شرطی می باشد. زمانی که شرط این دستور فعال شود و این دستور اجرا شود، برنامه به خطی که در این دستور مشخص می کنید می پرد. توجه داشته باشید که این دستور برای پرش در در برنامه اصلی می باشد و برای زیربرنامه ها و یا استیت نویسی استفاده نمی شود. برای مشخص کردن محل پرش در این دستور باید از برچسب های P استفاده کنید. بدین شکل که در جلوی دستور شماره برچسب مورد نظر را نوشته، سپس برچسب را در هر نقطه از برنامه که می خواهید بگذارید.

برای نوشتن دستور CJ را تایپ کرده، یک جای خالی گذاشته و سپس برچسب مورد نظر خود را بنویسید.

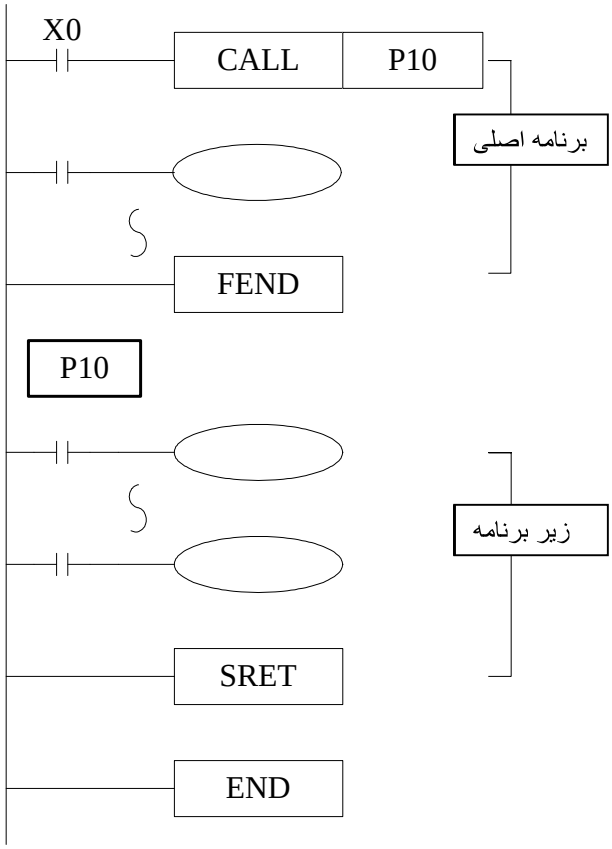


در مثال فوق زمانی که X0 ست شود، دستور پرش انجام شده و به برچسب P6 می پرد.دقت داشته باشید که اگر دستور پرش اجرا شود دیگر دستوراتی که بین این دستور پرش و برچسب می باشند اجرا نخواهند شد.

فراخوانی یا CALL:

این دستور برای اجرا شدن زیر برنامه استفاده می شود. زمانی که شرط این دستور فعال شود این دستور زیربرنامه مربوطه را فراخوانی کرده و در نتیجه زیر برنامه اجرا می شود و پس از اجرای زیر برنامه ، برنامه به خط زیری دستور CALL بر می گردد.

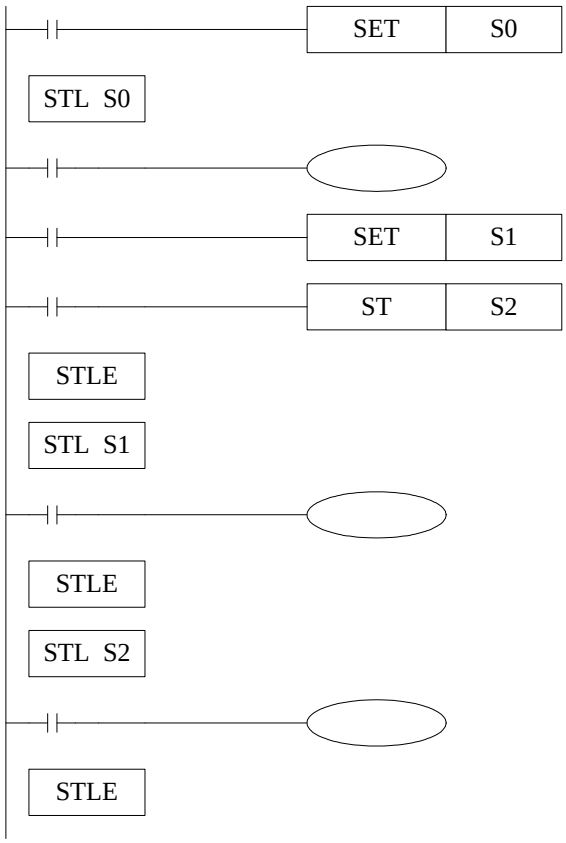
برای نوشتن برنامه باید از برچسب ها استفاده کنید. همچنین برای این شکل برنامه نویسی باید از دو دستور END استفاده کنید. یک بار در پایان برنامه اصلی باید از FEND و یک بار نیز در انتهای کل برنامه باید از END استفاده کنید. برای نوشتن زیر برنامه در پایان برنامه اصلی FEND را تایپ کنید و سپس در زیر آن برچسب مورد نظر خود را تایپ کنید. در زیر برچسب برنامه زیربرنامه را بنویسید و در انتهای زیربرنامه SRET را تایپ کنید.در پایان نیز END را فراموش نکنید.



طریقه استیت نویسی:

در استیت نویسی شما برنامه را می توانید به چند جز تقسیم کنید، که هر قسمت فقط زمانی که شما می خواهید اجرا می شود و اگر نخواستید می توانید آن قسمت از برنامه را به طور کلی غیر فعال کنید. در زیر برنامه نویسی، زیربرنامه جزو برنامه اصلی نمی باشد وفقط زمانی که آن را فراخوانی کنید یک بار اجرا شده و سپس به برنامه اصلی باز می گردد.ولی استیت ها جزو برنامه اصلی می باشند که اگر فعال باشند همواره اجرا خواهند شد.برای نمونه فرض کنید شما برنامه ای می خواهید بنویسید که هم حالت دستی داشته باشد و هم حالت اتوماتیک،زمانی که حالت دستی فعال است حالت اتوماتیک نباید کار کند و بالعکس. بدین منظور می توانید هرکدام از این دو حالت زپرا یک استیت تعریف کنید که در هر لحظه فقط یکی از آنها فعال باشد.

برای استیت نویسی در شروع استیت باید برچسب آن را تایپ کنید. برای این کار در اول خط و بدون شرط STL را تایپ کرده ، یک جای خالی گذاشته و سپس برچسب آن را که باS شناخته می شود را بنویسید. در زیر برچسب برنامه استیت را نوشته و در انتهای استیت STLE را تایپ کنید. در استیت نویسی برخلاف زیر برنامه نباید استیت را بعد از END بنویسید، بلکه باید در متن اصلی برنامه باشد.

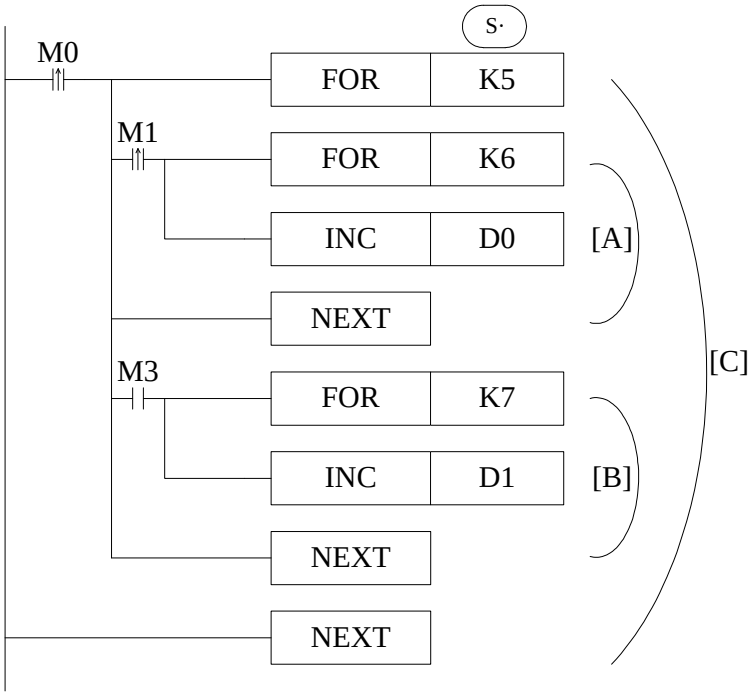


حال کافیست برچسب مربوط به هرکدام از استیت ها را که می خواهید ست کنید تا آن استیت فعال شده و برنامه آن اجرا شود.

**اگر برای فعال کردن یک استیت ، برچسب آن را در داخل یکی از استیت ها ست کنید، اتوماتیک بقیه برچسب ها ریست می شود ولی اگر یکی از برچسب ها را در خارج از استیت ها ست کنید ، وضعیت مابقی برچسب ها عوض نمی شود، یعنی اگر ست بودند ست مانده و اگر ریست بودند ، ریست می مانند.

طریقه نوشتن حلقه در برنامه:

برای حلقه تکرار نوشتن باید از دستورات FOR-NEXT استفاده کنید. بدین شکل که در ابتدای حلقه For را تایپ کرده، یک جای خالی گذاشته و سپس تعداد تکرار را بنویسید. در انتهای حلقه نیز NEXT را تایپ کنید.



در مثال بالا سه حلقه وجود دارد که دو تای آنها درون حلقه اول می باشند.

***برای هر حلقه باید حتما FOR-NEXT را بنویسید .**

***اگر درون یک حلقه، حلقه دیگری بنویسید و برای شرط آن از لبه استفاده کنید،آن حلقه فقط یک بار اجرا خواهد شد. مثلا در شکل بالا اگر M0,M1 ست شوند حلقه A اجرا می شود ولی فقط ۶ بار، در صورتیکه اگر M0,M3 ست شوند حلقه B به تعداد ۷×۵ بار، یعنی ۳۵ بار اجرا خواهد شد.**

***اگر در برنامه از تعداد زیادی حلقه استفاده کنید، زمان اسکن تایم طولانی خواهد شد و ممکن است باعث ایجاد مشکل در مانیتورینگ شود.**

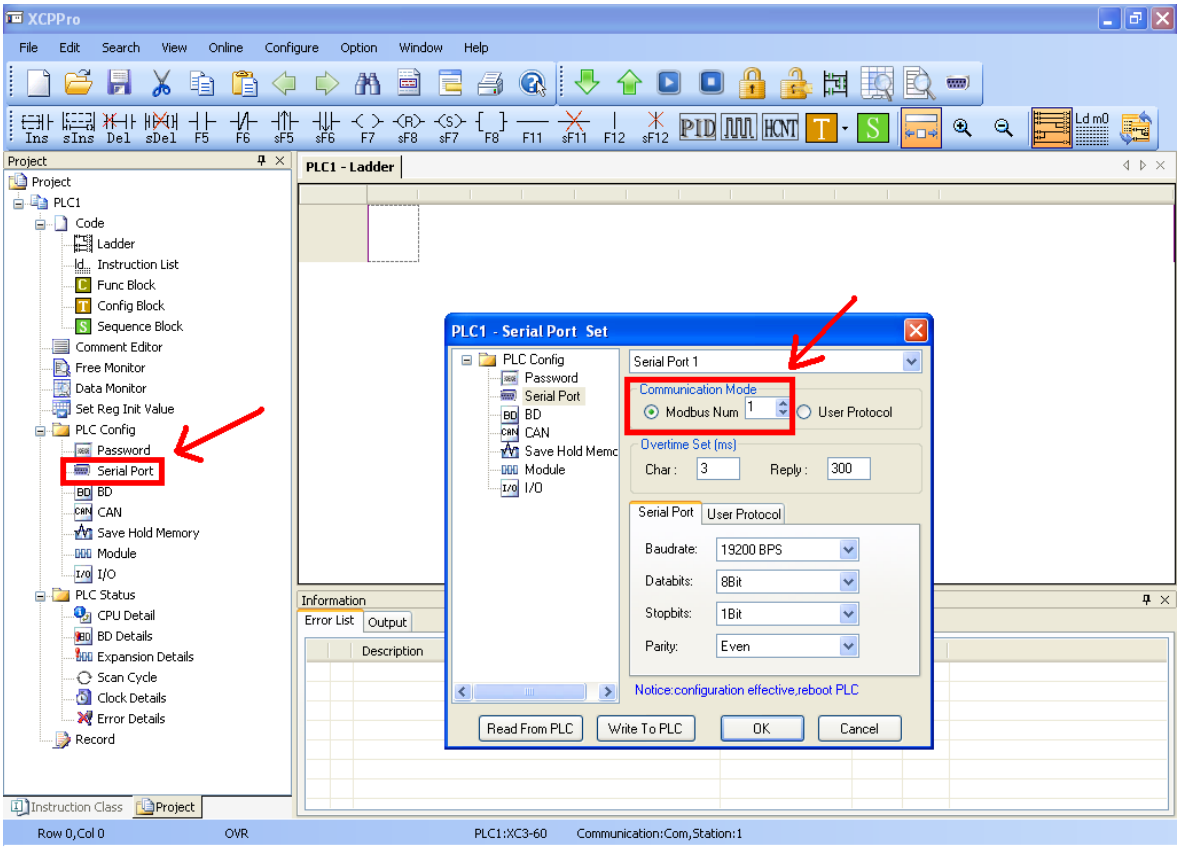
***در داخل یک حلقه نباید از دستور پرش شرطی استفاده شود.**

طریقه اتصال PLCهای THINGET از طریق Modbus

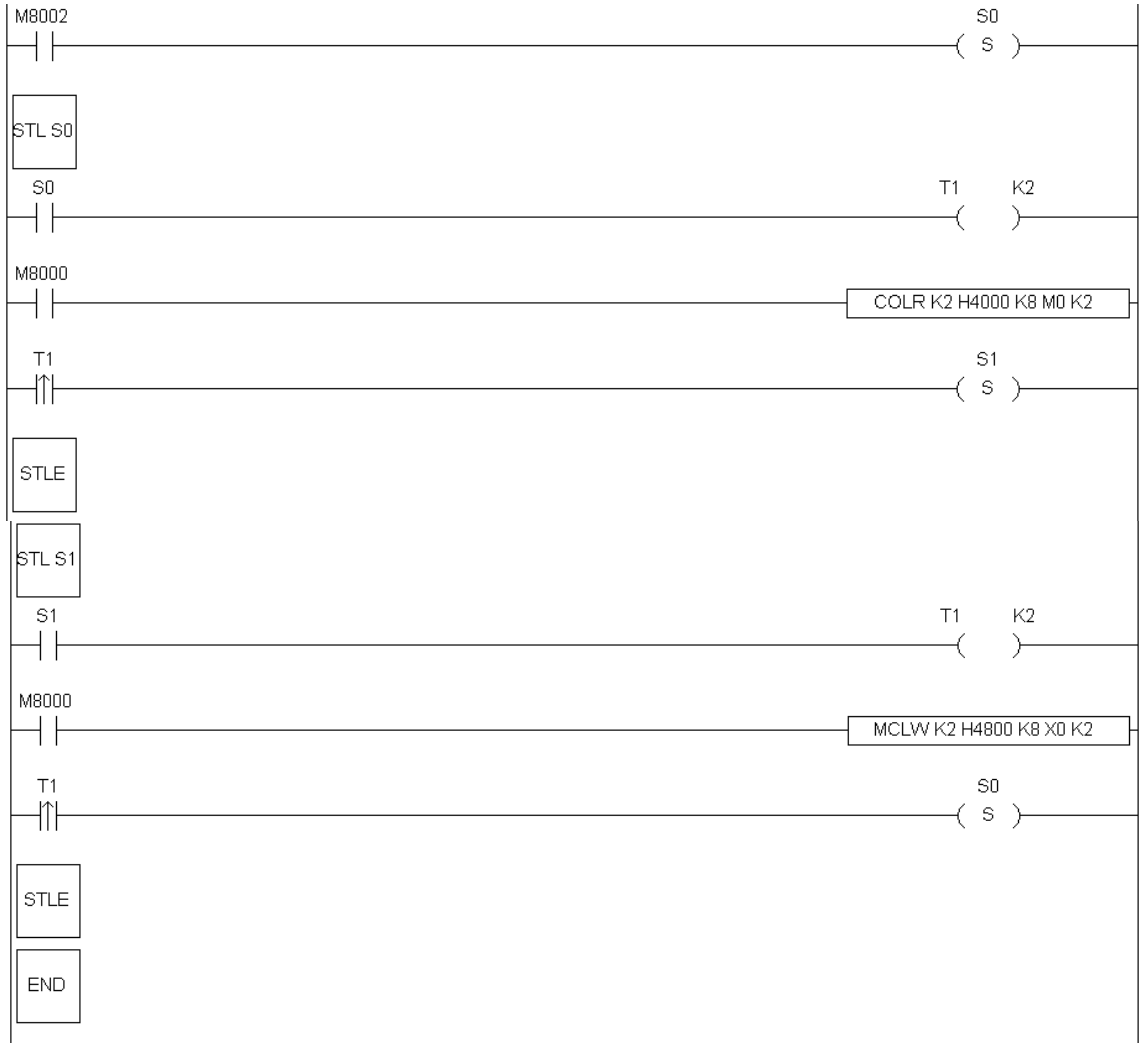
برای این منظور فرض کنید دو PLC را می خواهیم به یکدیگر وصل کنیم. همانگونه که می دانید در ارتباط Modbus یک PLC به عنوان Master و دیگری به عنوان Slave می باشد. برقراری ارتباط بدین شکل است که Master دستور را به Slave می فرستد و Slave جواب Master را می دهد.به دلیل آنکه PLC های Thinget پروتکل Modbus را ساپورت می کند نیازی به ریختن برنامه در Slave وجود ندارد و فقط باید دستورات خواندن و نوشتن را در Master دانلود کنید. برای ارتباط دو PLC فقط کافی است پایه های A و B مربوط به Master ، واقع بر ترمینال های سمت خروجی ها را به Slave وصل کنید.

برای برقراری ارتباط مراحل زیر را انجام دهید:

۱- ابتدا می بایست پورت های PLC را تنظیم کنید. برای این منظور در برنامه XCPPRO در قسمت Serial Port، Modbud NO. مربوط به Master را ۱ و Modbus No. مربوط به Slave را برابر ۲ قرار دهید.



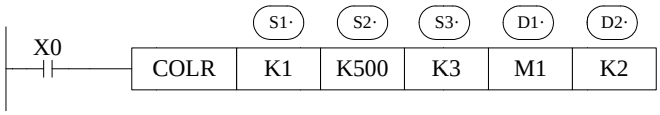
۲- سپس برنامه زیر را درون PLC Master دانلود کنید.



در این برنامه نمونه، در State S0 ورودی های X0~X7 ، Slave خوانده شده و به M0~M7 مربوط به Master منتقل می شوند.همچنین در S1 ورودی های X0~X7 مربوط به Master به خروجی های Y0~Y7 مربوط به Slave منتقل می شوند. دلیل برنامه نویسی بدین شکل این است که نمی توان همزمان دستور خواندن و نوشتن را ارسال کرد.بنابراین این برنامه در دو Stateنوشته می شود که یک State دستور خواندن و دیگری دستور نوشتن را ارسال می کند.

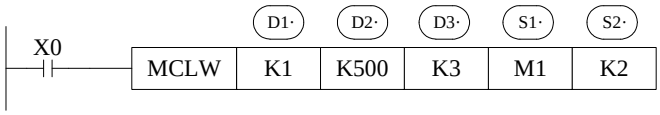
توضیح دستورات:

1、Coil Read [COLR]



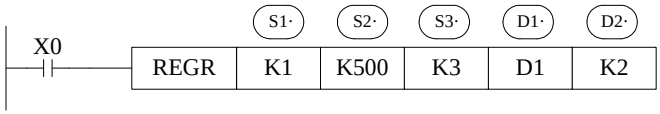
این دستور برای خواندن بیت ها از Slave می باشد.در این دستور داریم:
S1- Modbus num مربوط به Slave
S2- آدرسی از Slave که می خواهیم آن را بخوانیم
S3- تعدات بیت هایی که می خواهیم بخوانیم
D1- آدرسی از Master که می خواهیم بیت ها ی خوانده شده را به آنجا منتقل کنیم
D2- پورت ارتباطی، که در ارتباط بین دو PLC باید همیشه K2 باشد.

2、Multi-coil write [MCLW]



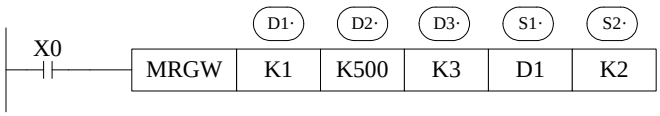
این دستور برای نوشتن بیت ها در Slave می باشد. در این دستور داریم:
D1- Modbus num مربوط به Slave
D2- آدرسی از Slave که می خواهیم در آنجا Write کنیم
D3- تعداد بیت هایی که می خواهیم Writeکنیم
S1- آدرسی از Master که می خواهیم بیت ها را از آنجا بخوانیم و به Slave منتقل کنیم
S2- پورت ارتیاطی، که در ارتباط بین دو PLC باید همیشه K2 باشد.

3、Register Read [REGR]



این دستور برای خواندن مقدار رجیسترها از Slave می باشد.در این دستور داریم:
S1- Modbus num مربوط به Slave
S2- آدرسی از Slave که می خواهیم آن را بخوانیم
S3- تعداد رجیسترهایی که می خواهیم بخوانیم
D1- آدرسی از Master که می خواهیم رجیستر ها ی خوانده شده را به آنجا منتقل کنیم
D2- پورت ارتیاطی، که در ارتباط بین دو PLC باید همیشه K2 باشد.

4、Multi-register Write [MRGW]



این دستور برای نوشتن رجیستر ها در Slave می باشد. در این دستور داریم:
D1- Modbus num مربوط به Slave
D2- آدرسی از Slave که می خواهیم در آنجا Write کنیم
D3- تعداد رجسیتر هایی که می خواهیم Writeکنیم
S1- آدرسی از Master که می خواهیم رجیستر ها را از آنجا بخوانیم و به Slave منتقل کنیم
S2- پورت ارتیاطی، که در ارتباط بین دو PLC باید همیشه K2 باشد

آدرس دهی حافظه داخلی پی ال سی مطابف زیر می باشد:

Coil’s start ID (**Dec.**)	M0	X0	Y0	S0	M8000	T0	C0
Corresponded Modbus ID (**Hex.**)	0	4000	4800	5000	6000	6400	6C00

Register's start ID (**Dec.**)	D0	TD0	CD0	D8000	FD0	FD8000
Corresponded Modbus ID (**Hex.**)	0	3000	3800	4000	4800	6800

طریقه کار با 6TC-AP:

این مدل از ورودی های ترموکوپل، قدری متفاوت از مدل های قبلی می باشند. در این سری علاوه بر این که شما تمامی مدل های ترموکوپل را می توانید به آن متصل کنید، این قابلیت را دارند که برای هر کانال پارامترهای متفاوتی برای PID تنظیم کنید. اگر فقط قصد خواندن دما را دارید، همانند مدل های قدیمی کافیست با توجه به محل قرار گرفتن ماژول ، آدرس ID متناظر با آن را پیدا کرده و آن را بخوانید. ولی برای تنظیم کردن پارامترهای آن باید از دستورهای مخصوص این مازول استفاده کنید. برای استفاده از 6TC-AP باید از Firmware 3.1i و ورژن XCPPRO3.1h به بعد استفاده کنید.

الف: نوشتن در ماژول:

برای نوشتن پارامترها باید از دستور To استفاده کنید. بدین منظور از جدول زیر شماره حافظه مربوط به حافظه ماژول را پیدا کرده و به فرمت زیر دستور را بنویسید:

Address	**Description**	
K0	Self-study PID trigger signal	
K1	Ch0	Setting temperature
K2	Ch1	Setting temperature
:	:	:
K6	Ch5	Setting temperature
K7	Ch0	PID parameter P
K8		PID parameter I
K9		PID parameter D
K10		PID parameter Diff

K11	Ch1	PID parameter P
K12		PID parameter I
K13		PID parameter D
K14		PID parameter Diff
:	:	:
K27	Ch5	PID parameter P
K28		PID parameter I
K29		PID parameter D
K30		PID parameter Diff
K31	Ch0	Temperature control period
K32	Ch1	Temperature control period
:	:	:
K36	Ch5	Temperature control period
K37	Ch0	Adjusting environment temperature
K38	Ch1	Adjusting environment temperature
:	:	:
K42	Ch5	Adjusting environment temperature
K43	Ch0	Self-study output range
K44	Ch1	Self-study output range
:	:	:
K48	Ch5	Self-study output range

ابتدا TO **را تایپ کرده، یک جای خالی گذاشته، محل قرار گرفتن ماژول،مجددا جای خالی، آدرس حافظه ماژول،مجددا جای خالی، تعداد،جای خالی و در نهایت عملگری را که می خواهید در ماژول کپی کنید را بنویسید.**

TO	K0	K1	K1	D10

در این مثال ماژول در جایگاه اول قرار گرفته است و می خواهیم دمای مطلوب آن را تعیین کنیم. عدد مورد نظر ما در رجیستر D10 می باشد که با این دستور داخل ماژول کپی می شود.

TO	K1	K7	K3	D20

در این مثال ماژول در جایگاه دوم می باشد. K7 مربوط به آدرس پارامتر P کانال اول می باشد، K3 یعنی ۳ رجیستر که در اینجا علاوه بر پارامتر P،پارامترهای I,D را نیز شامل می شود، D20 آدرس رجیستری است که پارامتر P را در آن ریخته ایم. این دستور به طور اتوماتیک رجیسترهای D21 را برای پارامتر I و D22 را برای پارامتر D می خواند.

ب- خواندن از ماژول

برای خواندن پارامترها باید از دستور FROM استفاده کنید. بدین منظور از جدول زیر شماره حافظه مربوط به حافظه ماژول را پیدا کرده و به فرمت زیر دستور را بنویسید:

Address	Description	
K0	Self-study PID control state signal	
K1	Ch0	PID output
K2	Ch1	PID output
:	:	:
K6	Ch5	PID output
K7	Ch0	PID parameter P
K8		PID parameter I
K9		PID parameter D
K10		PID parameter Diff
K11	Ch1	PID parameter P
K12		PID parameter I
K13		PID parameter D
K14		PID parameter Diff
:	:	:
K27	Ch5	PID parameter P
K28		PID parameter I
K29		PID parameter D
K30		PID parameter Diff
K31	Ch0	Temperature difference value
K32	Ch1	Temperature difference value
:	:	:
K36	Ch5	Temperature difference value

ابتدا **FROM** را تایپ کرده، یک جای خالی گذاشته، محل قرار گرفتن ماژول،مجددا جای خالی، آدرس حافظه ماژول،مجددا جای خالی، تعداد،جای خالی و در نهایت عملگری را که می خواهید اطلاعات را در آن کپی کنید را بنویسید.

FROM	K0	K1	K1	D10

در این مثال ماژول در جایگاه اول قرار گرفته است و می خواهیم خروجی **PID** کانال اول را بخوانیم و آن را در **D10** کپی می کنیم.

FROM	K1	K7	K3	D20

در این مثال پارامترهای **PID** کانال اول از ماژول دوم خوانده شده و در **D20** تا **D22** ذخیره می شوند.

نحوه فعال کردن PID:

برای فعال کردن PID ، پس از ست کردن پارامترها می بایست بیت مربوط به آن را ست کنید. این بیت را می توانید از جدول زیر =یدا کنید:

Enable signal	Module 1	Y100	Y101	Y10×	Y105
	Module 2	Y200	Y201	Y20×	Y205
	.....	Y×00	Y×01	Y×0×	Y×05
	Module 7	Y700	Y701	Y70×	Y705

کافیست با توجه به محل قرار گرفتن ماژول، آدرس مربوط به کانال مورد نظرتان را بیابید و با ست کردن ،PID کنترل را فعال کنید.

طریقه فعال کردن Self-study :

Self-study قابلیت جدیدی در ماژول های خواندن ترموکوپل پی ال سی های تینگت می باشد که می تواند پارامترهای PID را محاسبه کند. طریقه کارکرد بدین شکل است که پی ال سی پارامترهای مختلفی را برای PID ست کرده و نتیجه را بررسی می کند و بهترین پارامتها را به عنوان پارامترهای PID انتخاب می کند.

برای فعال کردن Self-study باید یک M را از حافظه داخلی پی ال سی برای کنترل آن اختصاص دهید و آن را به وسیله دستورTO درون ماژول کپی کنید، اگر آن M ست بود، Self-study فعال می شود و اگر ریست باشد Self-study غیر فعال می ماند. البته پس از آنکه ماژول پارامترها را محاسبه کرد حافظه داخلی ماژول که مربوط به Self-study می باشد را ریست می کند، بنا براین برای کپی کردن حافظه داخلی پی ال سی داخل ماژول نباید از شرطهای ماندگار استفاده کنید و فقط زمانی که می خواهید Self-study شروع شود با شرط های لبه بالا رونده و یا لبه پایین رونده باید حافظه پی ال سی را درون ماژول کپی کنید.

TO K0 K1 K1 M30

در مثال فوق M30 به عنوان بیت کنترلی Self-study انتخاب شده است.

در مثال فوق بیت های M30 تا M35 به عنوان بیت های کنترلی برای Self-study ماژول اول قرار داده شده اند که با ست شدن M100 این بیت ها درون ماژول کپی می شوند.

تعیین وضعیت اتصال ترموکوپل:

در ماژول های 6TC-AP این قابلیت وجود دارد که شما از اتصال ترموکوپل با خبر شوید. برای این کار کافیست تا با توجه به جایگاه ماژول، ورودی های متناظر با آن را چک کنید. برای این کار از جدول زیر استفاده کنید:

Connection state of thermocouple(0 is connection, 1 is cut connection)	Module 1	X110	X111	X11×	X115
	Module 2	X210	X211	X21×	X215
	·····	X×10	X×11	X×1×	X×15
	Module 7	X710	X711	X71×	X715

همان طور که در جدول بالا مشاهده می فرمایید مثلا برای ماژول اول باید آدرس X110 تا X115 را که مربوط به کانال های ۰ تا ۵ ورودی ترموکوپل می باشند را پک کنید. اگر این Xها خاموش باشند یعنی ترموکوپل وصل است و اگر روشن باشند یعنی ترموکوپل قطع می باشد.