کتاب خود آموز سریع

PLC دلتا

رسول مقصودی

استفاده از فایل بصورت

صلواتی

مقدمه

پس از اختراع ترانزیستور و با پیشرفت سریع علوم الکترونیک وکامپیوتر، مدارات الکترونیکی قابلیت و انعطاف زیادی پیدا کرده اند. ترکیب نرم افزار و سخت افزار و استفاده آنها در کنترل پروسه های صنعتی به تدریج تکامل پیدا کرده و بصورت یک استاندارد در آمد و نام کنترل کننده‌های منطقی برنامه پذیر(PLC) را به خود اختصاص داد. به عبارتی دیگر PLC مغز سیستم کنترل در صنعت امروز می‌باشد. بزرگترین مزیت PLC ، برنامه‌پذیر بودن آن است و همین امر سبب انعطاف پذیری برای PLC می‌گردد؛ بنابراین در صنایع مختلف از PLC استفاده می‌گردد و در هر صنعتی متناسب با نیاز ، برنامه‌ای برای PLC نوشته و روی آن دانلود می گردد .

امروزه شرکتهای بزرگی در زمینه ی طراحی و ساخت این کنترلرها فعالیت دارند که به عنوان نمونه می‌توان به شرکتهای زیمنس ، امرن ، آلن بردلی ، میتسوبیشی و دلتا اشاره کرد.

در کشور ایران نیز پس از ورود این کنترلرها به بازار ، استفاده از محصولات کمپانی آلمانی زیمنس و کمپانی تایوانی دلتا مورد استقبال مهندسین قرار گرفت. در این میان استفاده از محصولات دلتا بدلیل کیفیت ، قیمت ، تنوع محصول و خدمات گسترده‌ی پشتیبانی، بصورت چشمگیری افزایش یافت. استاندارد آموزشی این برند در سازمان فنی و حرفه ای کشور تدوین گردیده است . در این کتاب مباحث پایه و پیشرفته به زبانی ساده بیان شده است تا عزیزانی که این کتاب را مطالعه می‌کنند درک درست و صحیحی از برنامه نویسی PLC دلتا بدست آورند.

رسول مقصودی شار کنترل

آبانماه ۱۳۹۵

فهرست

فصل اول ۱

مفاهیم دسیمال ، باینری،، هگزادسیمال، اکتال و BCD : ۲

مبنای دسیمال (decimal) : ۲

مبنای باینری(binary) : ۲

مبنای هگزادسیمال(hexadecimal) : ۲

مبنای اکتال (Octal) : ۳

مبنای BCD : ۳

فصل دوم ۵

آشنایی با محیط نرم افزار ۵

آشنایی با نرم افزار WPLSOFT ۶

فصل سوم ۱۳

معرفی انواع کارت‌های ورودی- خروجی دیجیتال ۱۳

نحوه سیم بندی کارت‌های دیجیتال ۱۳

معرفی ورودی ها و خروجی های دیجیتال PLC : ۱۴

انواع کارت‌های دیجیتال : ۱۴

مشخصات PLC های دلتا سری Slim : ۱۵

مشخصات PLC های سری EC3 : ۱۶

مشخصات PLC های سری ES2 : ۱۷

مشخصات plc های سری EX2 : ۱۸

نحوه سیم بندی PLC : ۱۹

فصل چهارم ۲۱

معرفی دستورات برنامه نویسی (دستورات دیجیتال) ۲۱
شروع برنامه نویسی : ۲۲
دستورات پرکاربرد در PLC های دلتا : ۲۷
دستور SET : ۲۸
دستور RST : ۲۹
دستور ZRST : ۲۹
دستور ALT : ۳۰
دستور المانهای حساس به لبه پالس : ۳۱
دستور المان حساس به لبه بالا رونده : ۳۱
دستور المان حساس به لبه پایین رونده : ۳۲
دستور NOT : ۳۲
تایمرها : ۳۳
دستور TMR : ۳۴
دستور ATMR : ۳۷
دستور TTMR : ۳۹
دستور HOUR : ۴۰
دستور STMR : ۴۰
دستور خواندن زمان داخلی PLC با دستور TRD : ۴۱
دستور تنظیم زمان داخلی TWR : ۴۲
دستور شمارنده CNT : ۴۲
دستور DCNT : ۴۴
کانترهای سرعت بالا (HSC) : ۴۵

نحوه اتصال انکودر به PLC مدل DVP14SS211T : ۴۶
دستور DHSCS : ۴۸
دستور DHSCR : ۴۹
دستور DHSZ : ۴۹
دستورات ریاضی : ۵۰
دستور جمع اعداد صحیح ADD : ۵۰
دستور تقسیم اعداد صحیح DIV : ۵۱
دستور ضرب اعداد صحیح MUL : ۵۲
دستور تفریق SUB : ۵۲
تبدیل فرمت عدد صحیح (INT) به اعشاری (FLOAT) : ۵۳
دستور FLT & DFLT : ۵۳
دستور جمع اعداد اعشاری DADDR : ۵۴
دستور جمع اعداد اعشاری DEADD : ۵۴
دستور تقسیم اعداد اعشاری DDIVR : ۵۵
دستور تقسیم اعداد اعشاری DEDIV : ۵۵
دستور تفریق اعداد اعشاری DSUBR : ۵۵
دستور تفریق اعداد اعشاری DESUB : ۵۵
دستور ضرب اعداد اعشاری DMULR : ۵۶
دستور انتقال MOV : ۵۶
دستور انتقال اعداد اعشاری DMOVR : ۵۶
دستور جذر SQR : ۵۷
دستور جذر اعداد صحیح ۳۲ بیتی DSQR : ۵۷

دستور قدر مطلق **ABS** : ۵۷
دستور **MEAN** : ۵۸
استفاده از ترمینال های ورودی **X** به صورت گروهی در فضای حافظه **...KnXn** : ۵۸
دستور فراخوانی **CALL** : ۵۹
دستور **CJ** : ۶۰
دستور **INC** : ۶۰
دستور **DEC** : ۶۱
دستورات مقایسه کننده : ۶۱
دستور **CMP** : ۶۲
دستور مقایسه اعداد اعشاری **DECMP** : ۶۳
دستور مقایسه ناحیه‌ای **ZCP** : ۶۴
دستور DEZCP : ۶۵
دستور مقایسه زمانی **TCMP** : ۶۶
دستور مقایسه زمانی ناحیه‌ای **TZCP** : ۶۷
دستور **SPD** : ۶۷
دستور **PLSY** : ۶۸
دستور **PLSR** : ۶۹
دستور **PLSV** : ۷۰
دستور **DRVA** : ۷۱
دستور **PWM** : ۷۲
دستورات وقفه (**interrupt**) : ۷۳
انواع وقفه در **PLC** دلتا : ۷۳

وقفه خارجی (سخت افزاری) : ۷۴
کاربرد وقفه در شمارنده های سرعت بالا DHSCS : ۷۵
وقفه زمانی : ۷۶
پسورد گذاری در PLC های دلتا : ۷۶
فصل پنجم ۷۹
آنالوگ ۷۹
کارت ورودی آنالوگ : ۸۰
کارت خروجی آنالوگ : ۸۰
کارت ورودی و خروجی : ۸۰
انواع کارت دما : ۸۰
مشخصات فنی کارتهای آنالوگ سری S : ۸۱
مشخصات فنی کارتهای آنالوگ که از سمت چپ به PLC متصل می‌شوند : ۸۲
سیم بندی کارتهای آنالوگ : ۸۳
دستور خواندن از کارت آنالوگ و کارت دما توسط دستور FROM : ۸۷
کنترل رجیسترهای کارت 04PT-S : ۸۸
کنترل رجیسترهای کارت 04TC-S : ۹۴
روش تنظیم کنترل رجیسترهای کارت آنالوگ: ۹۸
دستور TO : ۹۸
پیکربندی کارتهای دما : ۹۹
محاسبه‌ی کد هگز مقادیر تنظیمات در کارت 04TC-S به صورت دستی : ۱۰۱
تنظیمات سنسور کارت 04PT-S با استفاده از wizard : ۱۰۳
تنظیمات کارت دمای 04PT-S به روش دستی : ۱۰۳

محاسبه‌ی کد هگز تنظیمات سنسورهای ورودی کارت 04PT-S به صورت دستی : ۱۰۳
پیکربندی کارتهای آنالوگ : ۱۰۵
دستور اسکیل SCLP : ۱۰۶
دستور SCAL : ۱۰۷
PLC های دارای ورودی خروجی آنالوگ : ۱۱۱
معرفی کارت لودسل DVP-01LC-SL : ۱۱۷
کالیبره‌کردن کارت لودسل : ۱۱۹
کنترل کننده PID : ۱۲۰
فصل ششم ۱۲۳
دانلود و آپلود برنامه PLC دلتا ۱۲۳
فصل هفتم ۱۲۷
آموزش زبان SFC ۱۲۷
انواع Sequence در زبان SFC : ۱۲۸
برخی از ابزارهای کاربردی در زبان SFC : ۱۲۹
شروع برنامه نویسی به زبان SFC : ۱۲۹
دستورات Set و Out : ۱۴۶
شاخه های موازی همزمان ۱۵۰
شاخه ها ی موازی غیر همزمان Alternative Divergence ۱۵۱
Multiple Sequence : ۱۵۶
فلگ‌های مهم در زبان SFC : ۱۵۸
تشریح عملکرد دستور IST جهت کنترل فرایند برنامه در زبان SFC : ۱۵۸
تشریح پارامترهای دستور IST : ۱۵۹

معرفی عملکرد پارامترهای دستور IST : ۱۵۹
فلگ های مهم در استفاده از دستور IST : ۱۶۰
استفاده از دستور IST در مد MANUAL : ۱۶۰
مد اتوماتیک با استفاده از دستور IST : ۱۶۷
مد ONE STEP در وضعیت Automatic : ۱۶۸
مد ONE CYCLE در وضعیت Automatic : ۱۶۹
مد CONTINUOUS OPERATION در وضعیت اتوماتیک : ۱۶۹
تشریح فلگ های مربوط به دستور IST : ۱۷۰
فصل هشتم ۱۷۳
ساختمان انکودر و انواع ۱۷۳
انکودر چست؟ ۱۷۴
کاربرد انکودرها ۱۷۴
انواع انکودر از نظر ساختمان : ۱۷۵
۱ – انکودرهای خطی (LINEAR) : ۱۷۵
۲ – انکودرهای دورانی (ROTARY) ۱۷۵
انواع انکودرها از نظر ساختار ۱۷۶
۱ – نوری (Optical) ۱۷۶
۲ – خازنی (Capasitive) ۱۷۶
۳ – مغناطیسی (Magnetic) ۱۷۶
انکودرها از نظر نوع عملکرد ۱۷۶
۱ – مطلق (ABSOLUTE ENCODER) ۱۷۶
۲ – افزایشی (INCREMENTAL) ۱۷۷

انکودرهای چرخشی مغناطیسی ۱۷۸
انکودر های چرخشی سینوسی و کسیونوسیresolvers ۱۷۹
انکودر های افزایشی خطی ۱۷۹
انکودر های مطلق خطی ۱۸۰
سیم بندی انکودر ۱۸۰
پیوست ۱۸۳
جدول حافظه های خاص (فلگها) M ۱۸۳
جدول رجیسترهای خاص D ۱۸۳
منابع : ۱۸۹

فصل اول

بررسی مبنای اعداد و کاربرد آنها در دیجیتال

مفاهیم دسیمال ، باینری،، هگزادسیمال، اکتال و BCD :

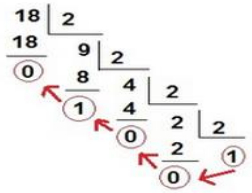
مبنای دسیمال (decimal) :

مبنای دسیمال که به آن مبنای ده دهی یا مبنای ده نیز گفته می‌شود، در کاربردهای روزمره در زندگی مورد استفاده قرار می‌گیرد به عنوان مثال زمانیکه می‌گوییم قیمت یک وسیله برابر ۲۳۵ تومان می‌باشد این عدد در مبنای دسیمال بوده چرا که هر کدام از ارقام این عدد مضربی از عدد ۱۰ می‌باشد و بصورت زیر بیان می‌شود :

$$2 \times 10^2 + 3 \times 10^1 + 5 \times 10^0 = 235$$

مبنای باینری(binary) :

از آنجاکه اساس کار رایانه ها و دستگاههای میکروپروسسوری مانند PLC ، بصورت صفر و یک می‌باشد مبنایی برای آنها در نظر گرفته شده ، که به آن مبنای باینری یا دودویی می‌گویند در این مبنا، کلیه اعداد بصورت صفر و یک بیان می‌شوند مطابق استانداردهای بین المللی، هر عدد دسیمال در مبنای باینری، از تعدادی صفر و یک تشکیل شده است یکی از روشهای محاسبه مبنای باینری اعداد دسیمال، استفاده از تقسیم‌های متوالی می‌باشد به عنوان مثال می‌خواهیم عدد ۱۸ دسیمال را به مبنای باینری تبدیل کنیم : $(18)_{10} = (?)_2$



$$(18)_{10} = (10010)_2$$

چنانچه بخواهیم مبنای باینری را به دسیمال تبدیل کنیم از ضرب‌های متوالی استفاده می‌کنیم مثال زیر را در نظر بگیرید: $(10010)_2 = (?)_{10}$

$$1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 18$$

مبنای هگزادسیمال(hexadecimal) :

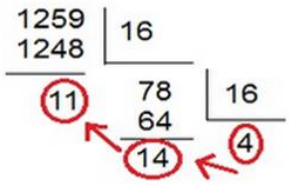
همانطور که در مثال قبل دیدیم هر عدد دسیمال در یک سیستم باینری از تعدادی صفر و یک تشکیل شده است حال اگر عدد دسیمال ما بزرگ باشد، عدد باینری معادل آن بسیار طولانی خواهد بود و کار کردن با این تعداد صفر و یک مشکل خواهد شد یکی از روشهای ساده سازی اعداد باینری استفاده از مبنای هگزادسیمال می‌باشد به این مبنا ، گاهی مبنای هگز و یا مبنای ۱۶ می‌گویند . در این مبنا، بیت‌های صفر و یک را به صورت دسته های ۴ تایی مرتب می‌کنند از آنجایی که هر ۴ بیت می‌تواند حداکثر ۱۶ حالت مختلف داشته باشد $(2^4) = 16$ لذا به آن مبنای ۱۶ می‌گویند . در مبنای هگز یا

۱۶ ، هر ۴ بیت می‌تواند برابر حداقل صفر و حداکثر ۱۵ باشد که ۱۶ حالت مختلف خواهد بود . این مبنا در سیستم های کامپیوتری و PLC ها، بسیار پرکاربرد می‌باشد.

برای نمایش اعداد در مبنای هگزادسیمال، از ۱۶ سمبل(کاراکتر) استفاده می‌شود. اعداد ۰ تا ۹ به صورت عادی (یعنی همان اعداد ۰ تا ۹ دسیمال) و اعداد ۱۰ تا ۱۵ را با حروف A (ده) و B (یازده) و C(دوازده) و D (سیزده) و E (چهارده) و F (پانزده) نشان می‌دهند.

برای تبدیل اعداد دسیمال به هگز ، می‌توان از تقسیمات متوالی استفاده نمود مثال زیر را در نظر بگیرید:

$$(1259)_{10} = (\ ?\ )_{16}$$



مطابق مطالب فوق، عدد ۱۴ در مبنای هگز با حرف E ، و عدد ۱۱ در مبنای هگز با حرف B نشان داده می‌شود .

$$(1259)_{10} = (\ 4EB)_{16}$$

مبنای اکتال (Octal) :

این مبنا که به آن مبنای ۸ نیز گفته می‌شود بیتهای باینری بصورت دسته های ۳ بیتی مرتب می‌شوند از آنجا که ۳ بیت میتواند حداکثر ۸ حالت مختلف داشته باشد به آن مبنای ۸ می‌گویند . در این صورت اعداد حداقل صفر تا حداکثر هفت می‌تواند در ۳ بیت وجود داشته باشد برای اعداد بزرگتر نیاز به بیتهای اضافی خواهیم داشت. کامپیوترهای جدید مانند سیستم های ۱۶ ، ۳۲ و ۶۴ بیتی ،اطلاعات را بصورت بایت (8-bit) دسته بندی می‌کنند و از طرفی عدم سازگاری سیستم اکتال با سیستم های جدید ، کاربرد سیستم های اکتال را محدود کرده اند .

نکته: در PLC های دلتا سری DVP ، روش آدرس‌دهی ورودیها (X) و خروجیها (Y) بصورت اکتال می‌باشد. اما در PLC های جدید دلتا سری AH و یا AS روش آدرس‌دهی ورودیها و خروجیها بصورت ۱۶ تایی می‌باشد. برای تبدیل اعداد دسیمال به اکتال و یا به العکس می‌توان از تقسیم های متوالی و یا از ضربهای متوالی استفاده نمود .

مبنای BCD :

این مبنا در واقع مبنای دسیمال تبدیل شده بصورت باینری می‌باشد همانطور که قبلا گفته شد ، در سیستم دسیمال اعداد بین 0...9 می‌باشند برای نمایش این اعداد در سیستم باینری نیاز به ۴ بیت داریم همانطورکه می دانیم در ۴ بیت اعداد 0... 15 قابل نمایش می‌باشد اما در سیستم BCD فقط تا عدد ۹ قابل نمایش خواهد بود. مبنای BCD به عنوان مثال در نمایشگرهای 7-segment مورد استفاده قرار می گیرد .

به عنوان مثال عدد ۱۰ در مبنای دسیمال با عدد ۱۰ در مبنای هگز متفاوت می‌باشد لذا برای نشان دادن یک عدد در یک مبنای خاص بصورت زیر عمل می‌کنند . به عنوان مثال عدد ۱۰ در مبنای هگز را بصورت های روبرو نشان می دهند : 10H یا H10 یا 10_{16}

معمولا برای نشان دادن اعداد در مبنای دسیمال ، عدد مورد نظر را بدون پیشوند و یا پسوند می نویسند.

245 یا $(245)_{10}$

جدول زیر مبنای اعداد را نشان می‌دهد . با مشاهده مقادیر موجود در جدول، تفاوت اعداد در مبناهای مختلف معلوم می‌شود .

دسیمال (مبنای ۱۰)	باینری (مبنای ۲)	هگزادسیمال (مبنای ۱۶)
0	0000 0000	0 0
1	0000 0001	0 1
2	0000 0010	0 2
3	0000 0011	0 3
4	0000 0100	0 4
5	0000 0101	0 5
6	0000 0110	0 6
7	0000 0111	0 7
8	0000 1000	0 8
9	0000 1001	0 9
10	0000 1010	0 A
11	0000 1011	0 B
12	0000 1100	0 C
13	0000 1101	0 D
14	0000 1110	0 E
15	0000 1111	0 F
16	0001 0000	1 0
17	0001 0001	1 1
18	0001 0010	1 2
19	0000 0011	1 3
20	0001 0100	1 4
21	0001 0101	1 5
22	0001 0110	1 6
23	0001 0111	1 7
.	.	.
.	.	.
.	.	.

باینری (مبنای ۲)	BCD
0000 0000	0 0
0000 0001	0 1
0000 0010	0 2
0000 0011	0 3
0000 0100	0 4
0000 0101	0 5
0000 0110	0 6
0000 0111	0 7
0000 1000	0 8
0000 1001	0 9
0001 0000	1 0
0001 0001	1 1
0001 0010	1 2
0001 0011	1 3
0001 0100	1 4
0001 0101	1 5
0001 0110	1 6
0001 0111	1 7
0001 1000	1 8
0000 1001	1 9
0010 0000	2 0
0010 0001	2 1
0010 0010	2 2
0010 0011	2 3
.	.
.	.
.	.

باینری (مبنای ۲)	اکتال (مبنای ۸)
000 000	0 0
000 001	0 1
000 010	0 2
000 011	0 3
000 100	0 4
000 101	0 5
000 110	0 6
000 111	0 7
001 000	10
001 001	11
001 010	1 2
001 011	1 3
001 100	1 4
001 101	1 5
001 110	1 6
001 111	1 7
010 000	20
010 001	21
010 010	22
010 011	23
010 100	24
010 101	2 5
010 110	26
010 111	27
.	.
.	.
.	.

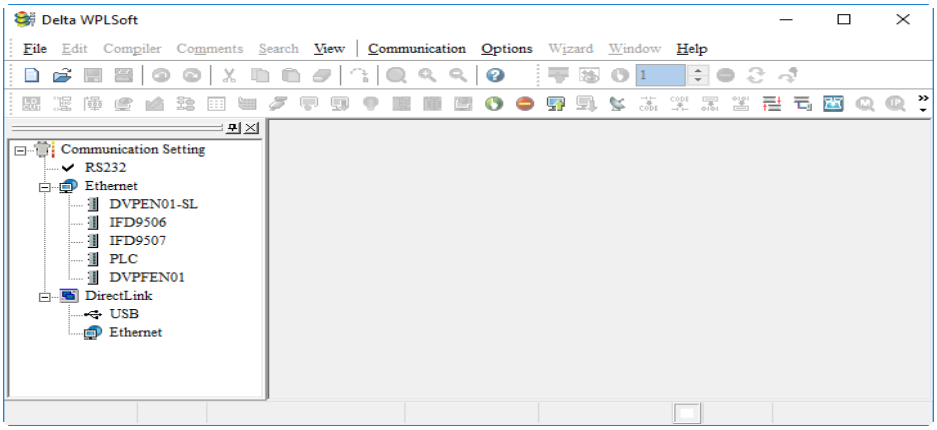
در جدول فوق به مقادیر باینری در مبناهای مختلف توجه شود و اینکه مقادیر باینری اعداد مبناهای هگز و BCD و اکتال چه فرق هایی با یکدیگر دارند .

فصل دوم

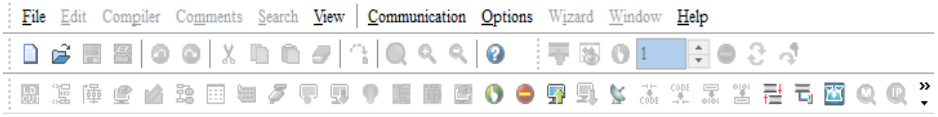
آشنایی با محیط نرم افزار

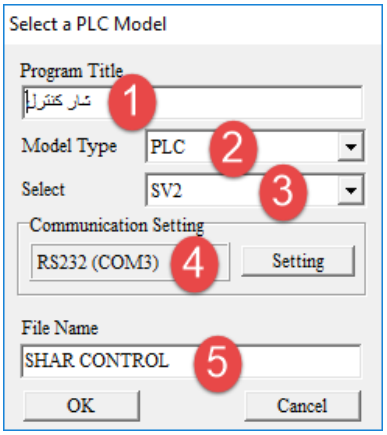
آشنایی با نرم افزار WPLSOFT

ابتدا نرم افزار را از سایت رسمی شرکت دلتا (Delta Automation)، دانلود و آن را نصب کنید. با کلیک بر روی آیکن ایجاد شده، برنامه WPLSoft مطابق شکل زیر اجرا می‌شود.



حال پروژه‌ای جدید ایجاد می‌کنیم. برای این کار کافیست همانند شکل زیر به قسمت File رفته و گزینه New را انتخاب کنید و یا در قسمت پایین گزینه فایل، آیکون مربوط به ایجاد پروژه‌ی جدید را انتخاب و یا از کلیدهای میانبر Ctrl + N استفاده کنید.





پس از کلیک بر روی گزینه New، پنجره‌ای به شکل روبرو باز می‌شود:

۱ – در این قسمت می‌توانید عنوان پروژه را وارد کنید.

۲ – در این قسمت نوع کنترلر مورد نظر را مشخص کنید.

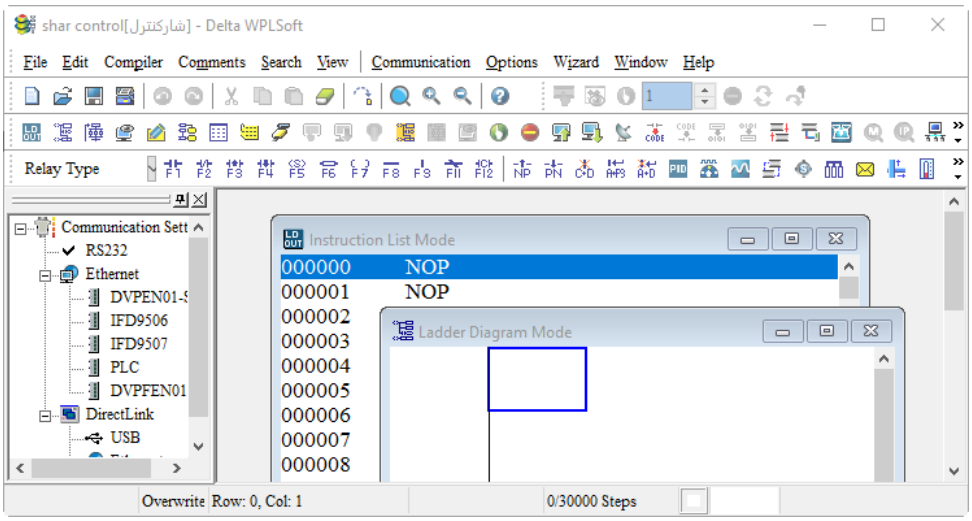
۳ – در این قسمت مدل CPU را مشخص کنید .

۴- در این قسمت تنظیمات ارتباطی را تنظیم کنید .

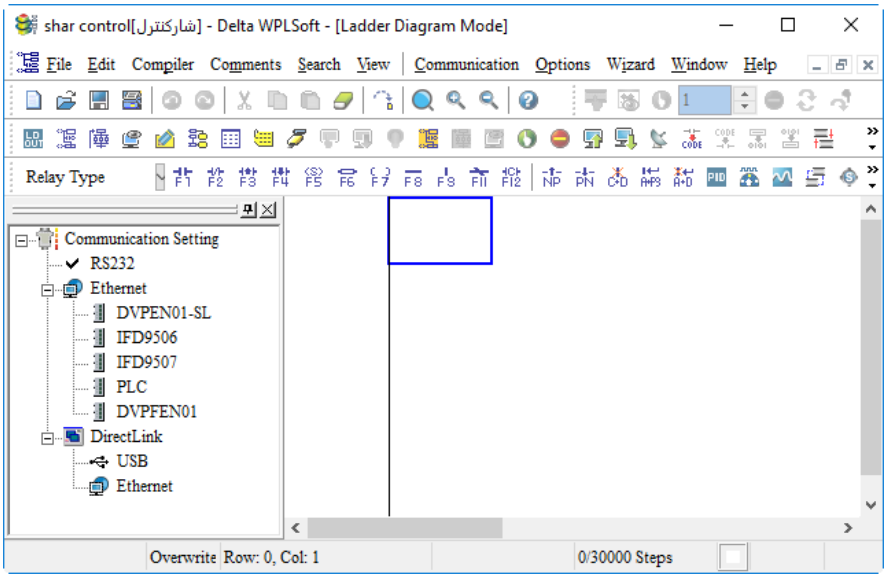
۵ – در این قسمت نام پروژه را وارد کنید.

در پایان با کلیک بر روی گزینه OK اطلاعات ثبت شده و وارد مرحله بعد می‌شود.

در این مرحله مطابق شکل زیر دو پنجره کوچک در صفحه اصلی برنامه WPLsoft ظاهر شده که هر کدام مربوط به یک زبان برنامه نویسی خاص می‌باشد.

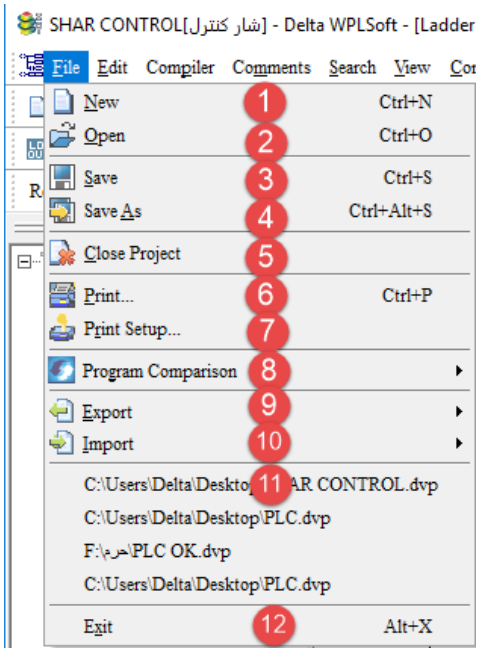


از آنجا که زبان برنامه نویسی LADDER (LAD) از محبوبیت بیشتری برخوردار است لذا پنجره مربوط به زبان Instruction List Mode(IL) را بسته و پنجره Ladder Diagram Mode را باز می‌کنیم.



حال نرم افزار آماده‌ی برنامه نویسی می‌باشد. در ادامه ابتدا به بررسی برخی از منوها و امکانات نرم افزار WPLsoft می پردازیم.

منوی File :



۱ – ایجاد پروژه‌ی جدید

۲ – باز کردن فایل

۳ – ذخیره کردن برنامه

۴ – ذخیره کردن با نام دیگر

۵ – بستن پروژه

۶ – پرینت گرفتن از برنامه

۷ – تنظیمات پرینت

۸ – مقایسه برنامه

۹ – Export کردن برنامه

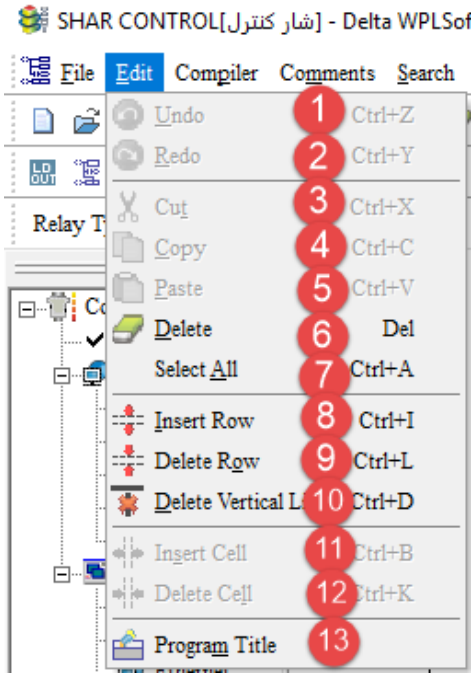
۱۰ – Import کردن برنامه

۱۱ – آخرین فایل ذخیره شده

۱۲ – خروج از نرم افزار

منوی Edit :

۲



۱ – بازگشت به عقب

– رفتن به جلو

۳ – بریدن برنامه

۴ – کپی کردن برنامه

۵ – چسباندن

۶ – حذف کردن

۷ – انتخاب همه

۸ – اضافه کردن یک سطر

۹ – حذف یک سطر

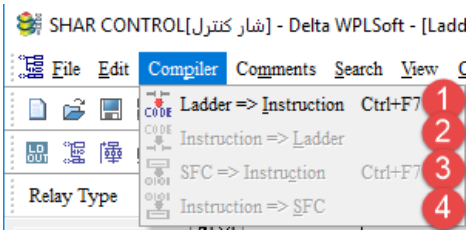
۱۰ – حذف خط عمودی

۱۱ – وارد کردن یک فاصله

۱۲ – حذف یک فاصله

۱۳ – عنوان برنامه

منوی Compile :



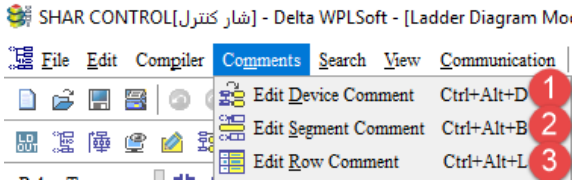
۱ – تبدیل زبان LAD به IL

۲ – تبدیل زبان IL به LAD

۳ – تبدیل زبان SFC به IL

۴ – تبدیل زبان IL به SFC

منوی Comments :

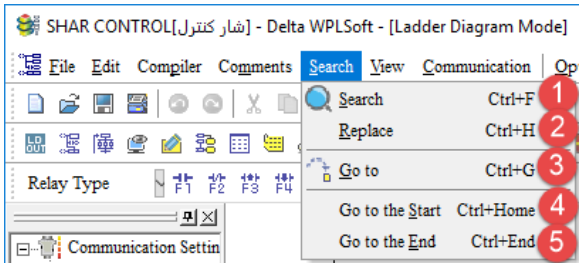


۱ – ویرایش توضیحات ابزارها

۲ – ویرایش توضیحات جمله ها

۳ – ویرایش توضیحات سطرها

منوی Search :



۱ – جستجو

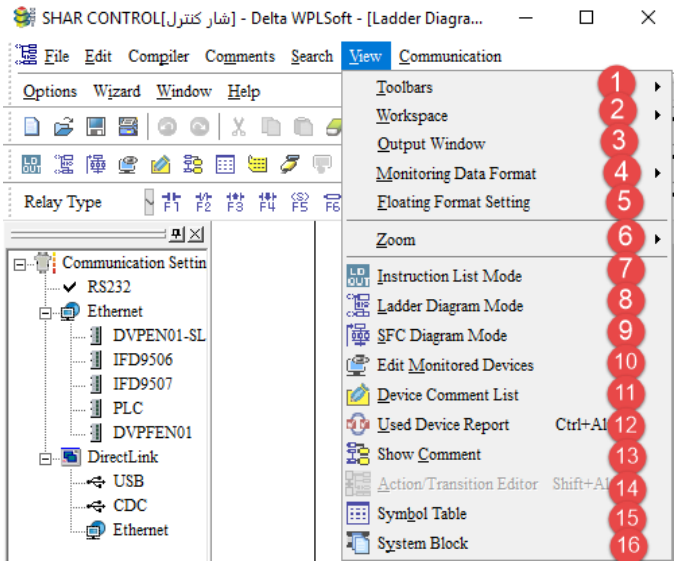
۲ – جابجا کردن

۳ – رفتن به

۴ – رفتن به ابتدا

۵ – رفتن به انتها

منوی View :



۱ – نوار ابزار

۲ – فضای کار

۳ – پنجره خروجی

۴ – نمایش نوع فرمت اعداد

۵ – تنظیمات نوع اعشاری

۶ – بزرگنمایی

۷ – زبان نوشتاری

۸ – زبان LAD

۹ – زبان SFC (فلوچارت)

۱۰ – ویرایش ابزارهای مانیتور شده

۱۱ – لیست توضیحات ابزارها

۱۲ – گزارش ابزار استفاده شده

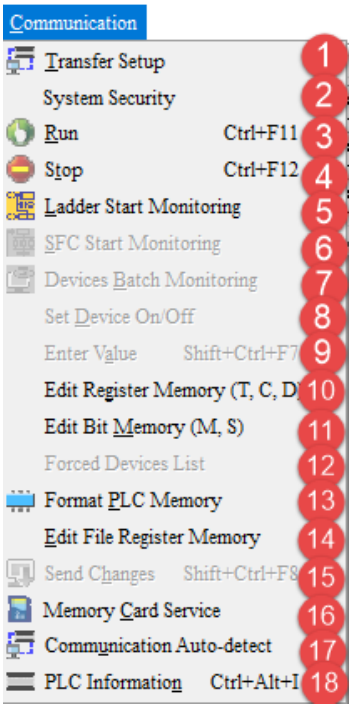
۱۳ – نمایش توضیحات

۱۴ – عملیات و شرط در SFC

۱۵ - جدول نمادها

۱۶ – بلوک اطلاعات سیستمی PLC

منوی **Communication** :



۱ – تنظیم تبادل ارتباطات

۲ – تنظیمات امنیتی

۳ – اجرا کردن برنامه

۴ – توقف برنامه

۵ – شروع مانیتورینگ LAD

۶ – شروع مانیتورینگ SFC

۷ - مانیتور کردن همزمان آدرسها

۸ – روشن و خاموش کردن آدرسها

۹ – وارد کردن مقدار

۱۰ – ویرایش حافظه‌های T,D,C

۱۱ – ویرایش حافظه‌های بیتی M,S

۱۲ – لیست ابزارهای اجباری

۱۳ – فرمت کردن حافظه PLC ، (Reset Factory)

۱۴ - ویرایش فایل رجیسترهای حافظه

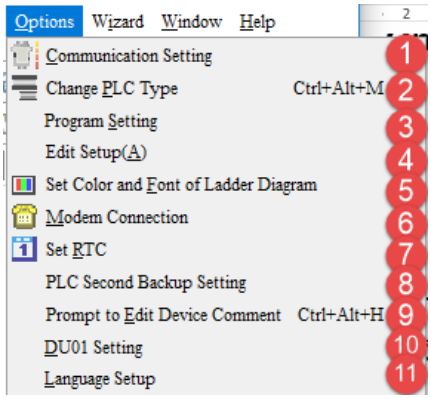
۱۵ – ارسال تغییرات

۱۶ – سرویس کارت حافظه

۱۷ - جستجوی خودکار برای ارتباط

۱۸ – اطلاعات PLC

منوی Options :



۱ – تنظیمات ارتباط

۲ – تغییر نوع PLC

۳ – تنظیمات برنامه

۴ – ویرایش تنظیمات

۵ – تنظیم رنگ و فونت برای زبان LAD

۶ – ارتباط مودم

۷ – تنظیم زمان PLC

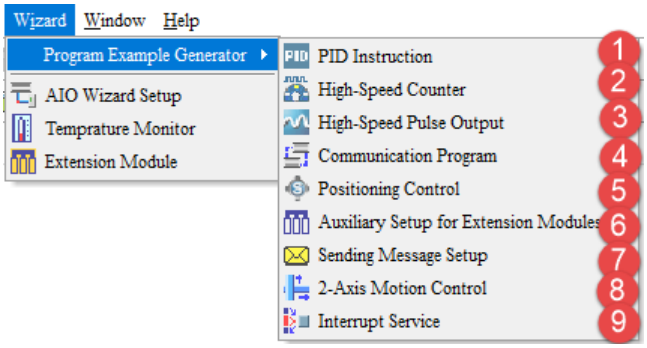
۸ – تنظیمات پشتبیانی ثابت PLC

۹ –ویرایش توضیحات المانها

۱۰ – عملیات پسورد

۱۱ – تنظیمات زبان

منوی Wizard :



– دستورات PID

۲ – شمارنده های سرعت بالا

۳ – خروجی پالس سرعت بالا

۴ – برنامه ارتباط شبکه

۵ – کنترل موقعیت

۶ – پیکربندی کارتهای افزایشی

۷ – تنظیمات ارسال پیام

۸ – کنترل موقعیت دو محوره

۹ – سرویس وقفه

فصل سوم

معرفی انواع کارت‌های ورودی- خروجی دیجیتال
نحوه سیم بندی کارت‌های دیجیتال

معرفی ورودی ها و خروجی های دیجیتال PLC :

در PLC های دلتا خروجی ها با Y و ورودی ها با X نشان داده می‌شوند. سیستم شماره گذاری این ورودی و خروجی ها به صورت Octal (هشت تایی) می‌باشد.

نکته : در PLC های دلتا درکنار مدل PLC یک عدد درج شده است که نشانگر تعداد ورودی و خروجی دیجیتال یا آنالوگ بر روی MPU(CPU) می‌باشد. برای مثال در PLC مدل 14SS2 عدد 14 نشان دهنده‌ی 8 ورودی دیجیتال و 6 خروجی دیجیتال می‌باشد.

انواع کارت‌های دیجیتال :

کارتهای دیجیتال با شماره و پسوندی که در نام خود دارند شناخته می‌شوند.

پسوند SM : کارتهای توسعه ای که دارای پسوند SM هستند، کارتهای ورودی دیجیتال می‌باشند مانند:

DVP 08SM که دارای 8 ورودی دیجیتال می‌باشد.

پسوند SN : کارتهایی که دارای پسوند SN هستند، کارتهای خروجی دیجیتال می‌باشند مانند :

DVP 08SN که دارای 8 خروجی دیجیتال می‌باشد.

پسوند SP : کارتهایی که دارای پسوند SP هستند، کارتهای ورودی و خروجی دیجیتال می‌باشند مانند :

DVP 16SP که دارای 8 ورودی و 8 خروجی دیجیتال می‌باشد.

نکته : تعداد کارتهای دیجیتال که می توان به CPU متصل کرد به مشخصات CPU بستگی دارد اولین ورودی کارت دیجیتال که به CPU مدل 14SS2 ، 12SA2 ، 12SE ، 28SV2 متصل می شوند X20 می باشد.

نکته : تعداد 8 کارت آنالوگ می توان از سمت راست به CPU متصل کرد هم چنین در بعضی از مدلها مانند 28SV2 می توان 8 کارت آنالوگ از سمت چپ به CPU متصل کرد البته آدرس کارتهای سمت چپ CPU از K100 الی K107 می باشد .

مشخصات PLC های دلتا سری Slim :

مشخصات	DVP14SS211R/T	DVP12SA211R/T	DVP10SX11R/T	DVP20SX211R/T	DVP12SE11R/T	DVP28SV11(R/T)2
تغذیه	24V DC					
تعداد ورودی های دیجیتال 24V	8	8	4	8	8	16
تعداد خروجی های دیجیتال	6	4	2	6	4	12
تعداد ورودی های آنالوگ	---	---	2(12Bit)	4(12Bit)	---	---
تعداد خروجی های آنالوگ	---	---	2(12Bit)	4(12Bit)	---	---
حافظه ورودی/خروجی دیجیتال	480					512
تعداد شمارنده های قابل اتصال AB MODE	2*10KHz 2*5KHz	1*50KHz 3*5KHz	1*10KHz	1*50KHz 3*5KHz	1*50KHz 3*5KHz	2*100KHz 2*10KHz
RTC	دارد					
باتری داخلی	دارد					
ظرفیت حافظه برنامه	8K step	16K step				30K step
حافظه های 16 بیتی عمومی	D0...D407 D600...D999 D3920...D4999 D5000...D9799	D0...D199		D0...D407 D600...D999 D3920...D4999		D0...D199
حافظه 16 بیتی پایدار	D408...D599 D2000...D3919	D200...D999 D2000...D4999		D408...D599 D2000...D3919		D200...D999 D2000...D9999
حافظه 16 بیتی خاص	D1000...D1999					
حافظه های بیتی عمومی	M0...M511 M768...M999 M2000...M2047	M0...M511		M0...M511 M768...M999 M2000...M2047	M0...M511 M768...M999 M2000...M2047	M0...M499
حافظه های بیتی پایدار	M512...M767 M2048...M4095	M512...M999 M2000...M4095		M512...M767 M2048...M4095	M512...M767 M2048...M4095	M500...M999 M2000...M4095
حافظه های بیتی خاص	M1000...M1999					
پورت سمت راست	دارد					
پورت سمت چپ (LEFT SIDE)	تعداد	دارد	تعداد	دارد	تعداد	دارد
پورت های ارتباطی	1 * RS232 1 * RS485	1 * RS232 2 * RS485	1 * RS232 1 * RS485	1 * RS232 1 * Mini USB	1 * RS232 1 * Ethernet 1 * Mini USB	1 * RS232 1 * RS485
دمای کاری و رطوبت مجاز	5~95%		0~55℃			

مشخصات PLC های سری EC3 :

مشخصات	RDVP10EC00R3	DVP14EC00R3	DVP16EC00R3	DVP20EC00R3	DVP24EC00R3	DVP30EC00R3	DVP32EC00R3	DVP40EC00R3	DVP48EC00R3	DVP60EC00R3
تغذیه	100 ~ 240 V AC									
تعداد ورودی های دیجیتال 24V DC	6	8	8	12	12	18	16	24	28	36
تعداد خروجی های دیجیتال	4	6	8	8	12	12	16	16	20	24
حداکثر ورودی و خروجی دیجیتال قابل افزایش	غیر قابل افزایش									
تعداد کانتر های قابل اتصال A/B MODE	1*4KHz									
تعداد سروو موتور های قابل اتصال (برای خروجی های ترانزیستوری)	1*10KHz									
RTC	ندارد									
حافظه	4k step									
پورت RS232 برای افزایش ماژولها	ندارد									
پورت هیب برای افزایش ماژولها	ندارد									
پورت های ارتباطی	1 * rs232 1 * rs485									

مشخصات PLC های سری ES2 :

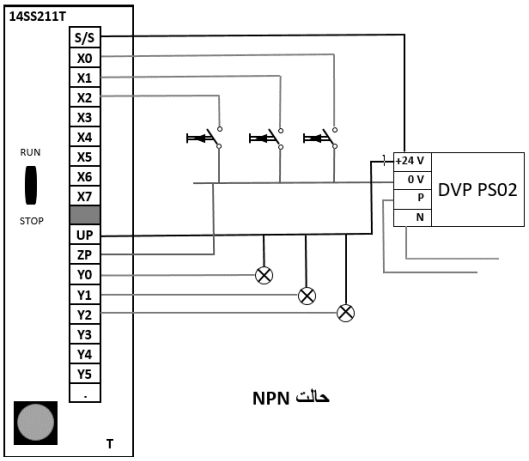
مشخصات	DVP16ES200[R/T]	DVP16ES200[R/T]	DVP16ES200[R/T]	DVP16ES200[R/T]	DVP16ES200[R/T]	DVP16ES200[R/T]	DVP16ES200[R/T]
تغذیه	100-240V DC		24V DC	100-240V DC			
تعداد ورودی های دیجیتال 24V DC	36	24	16	16	16	16	8
تعداد خروجی های دیجیتال	24	16	16	16	16	8	8
پورت های ارتباطی	1 * RS232 2 * RS485		1 * RS232 1 * RS485 1 * CANOpen	1 * RS232 2 * RS485			
حداکثر تعداد ورودی و خروجی قابل افزایش	256						
تعداد کانتر های دقت بالا A/B MODE	2 * 50KHz						
تعداد سرورد موتور های قابل اتصال (برای خروجی های ترانزیستوری)	2 * 100KHz						
RTC	دارد						
حافظه	16K Step						
پورت راست برای افزایش ماژول ها	دارد						
پورت چپ برای افزایش ماژول ها	ندارد						

مشخصات **plc** های سری **EX2** :

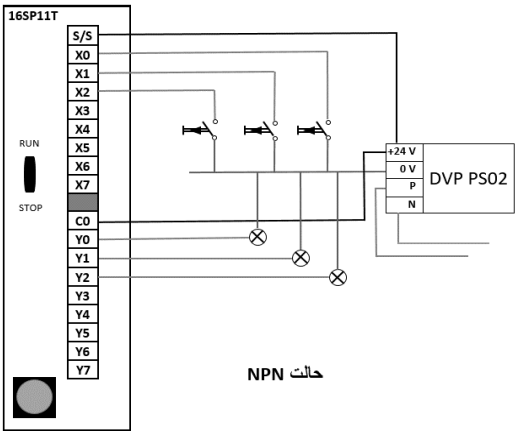
مشخصات	DVP20EX200(R/T)	0EX200(R/T)۳DVP
تغذیه	- 240V DC	DVP30EX200(R/T)
تعداد ورودی های دیجیتال 24V DC	8	16
تعداد خروجی های دیجیتال	6	10
تعداد ورودی های آنالوگ	4 (12 Bit)	3 (12 Bit)
تعداد خروجی های آنالوگ	2 (12 Bit)	1 (12 Bit)
حد اکثر تعداد ورودی و خروجی دیجیتال قابل افزایش	256	
تعداد انکدرهای قابل اتصال (A/B MODE)	1*15KHz	
تعداد سروو موتورهای قابل اتصال برای خروجی های ترانزیستوری	1*100KHz	
RTC	دارد	
حافظه	16K Step	
پورت راست برای افزایش ماژولها	دارد	
پورت چپ برای افزایش ماژولها	ندارد	
پورت های ارتباطی	2*RS485	1*RS232

نحوه سیم بندی PLC :

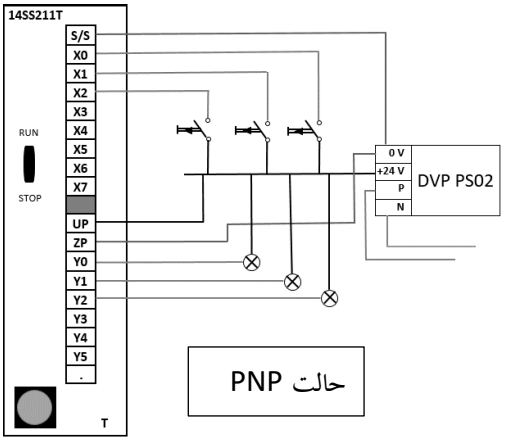
در شکل زیر ورودیها بصورت Sink بوده و خروجیها بصورت NPN می‌باشد.



در شکل زیر ورودیها بصورت Source بوده و خروجیها بصورت NPN می باشد



در شکل زیر ورودیها بصورت Sink بوده و خروجیها بصورت PNP می‌باشد.



۲۰.....اتوماسیون صنعتی DELTA

فصل چهارم

معرفی دستورات برنامه نویسی (دستورات دیجیتال)

شروع برنامه نویسی :

در PLC دلتا به المانهای X ، Y ، M ، D ، S ، T ، C یک Device گفته می‌شود . قبل از شروع برنامه نویسی به تعریف آنها می پردازیم :

X : ورودیهای PLC که بر مبنای اکتال می‌باشد . X0 ... X7 & X10 ... X17 & X20 ... X27

Y : خروجیهای PLC که بر مبنای اکتال می‌باشد . Y0 ... Y7 & Y10 ... Y17 & Y20 ... Y27

M : حافظه‌های داخلی که گاهی به آنها رله های کمکی یا فلگ نیز می‌گویند. این حافظه‌ها بصورت بیتی بوده و به سه دسته کلی تقسیم می شوند :

حافظه‌های عمومی که ناپایدار هستند . (با قطع برق ریست می شوند)

حافظه‌های عمومی که پایدار هستند .(با قطع برق وضعیت شان حفظ می شود)

حافظه‌های مخصوص که وظایف خاصی دارند و بعضی از آنها پایدار می‌باشد .

برای اطلاع از تعداد و مشخصات حافظه‌ها M ، به راهنمای PLC مربوطه مراجعه کنید . شکل زیر مشخصات حافظه داخلی PLC دلتا مدل SS2 14 را نشان می دهد

M	Auxiliary relay	General	M0~M511, 512 points, (*1) M768~M999, 232 points, (*1) M2000~M2047, 48 points, (*1)	Total 4096 points
		Latched	M512~M767, 256 points, (*2) M2048~M4095, 2048 points, (*2)	
		Special	M1000~M1999, 1000 points, some are latched	

همانطور که در جدول بالا دیده می‌شود حافظه‌های M1000 تا M1999 جزء حافظه‌های خاص می باشند . به عنوان مثال M1000 به محض RUN شدن PLC فعال (ON) می‌شود . M1013 بطور دائم پالس با فرکانس 1HZ تولید می کند و یا M1002 در اولین سیکل برنامه بعد از RUN شدن PLC فعال و سپس غیر فعال می‌شود . برای اطلاع از عملکرد کلیه حافظه ها به راهنمای مربوطه، مراجعه شود .

D : رجیسترهای 16 بیتی بوده و به چند دسته، مطابق زیر تقسیم می شوند :

رجیسترهای عمومی که ناپایدار می باشند .

رجیسترهای عمومی که پایدار می باشند .(Latched)

رجیسترهای خاص که هرکدام دارای عملکرد خاصی می باشند .

رجیسترهای خاص که برای ماژولهای خاصی تعیین شده اند .

برای اطلاع از تعداد و مشخصات رجیسترهای هر PLC باید به راهنمای آن مراجعه شود .

جدول زیر مشخصات رجیسترهای PLC دلتا مدل 14SS2 را نشان می دهد .

D	Data register	General	D0~D407, 408 words, (*1) D600~D999, 400 words, (*1) D3920~D4999, 1080 words, (*1)
		Latched	D408~D599, 192 words, (*2) D2000~D3919, 1920 words, (*2)
		Special	D1000~D1999, 1000 words, some are latched

T : تایمرها که به عنوان ابزارهای زمان سنج عمل کرده و در بخش تایمرها به آن پرداخته شده است

C : کانترها وظیفه شمارش را به عهده دارند و در بخش کانترها توضیح داده شده است .

S : رله های مرحله‌ای (Step Relay) بوده و در برنامه نویسی فرآیندهای ترتیبی مورد استفاده قرار می گیرد .

E ، F : ایندکس که می‌تواند به عنوان رجیسترهای کمکی در برخی دستورات مورد استفاده قرار بگیرد

نکته : اعداد صحیح دسیمال باید با پیشوند K استفاده شود مانند : K65

نکته : اعداد در مبنای هگز با پیشوند H استفاده می‌شود : H5A

نکته : اعداد ممیز دار با پیشوند F نمایش داده می شوند مانند : F34.62

برای شروع برنامه نویسی ابتدا به توضیح در مورد برخی المان‌های پرکاربرد در برنامه WPLSoft می‌پردازیم. این المان‌ها در قسمت نوار ابزار برنامه WPLSoft موجود هستند.

۱ – تیغه باز (Normally Open Contact) ۲ – تیغه بسته (Normally Close Contact)

۳ – لبه بالارونده (Rising Edge) ۴ – لبه پایین رونده (Falling Edge)

۵ – رله های مرحله ای (Step Relay)

۶ – دستورات برنامه (Application Instruction) ۷ – خروجی (Output Coil)

۸ – اتصال خط افقی (Horizontal Line) ۹ – اتصال خط عمودی (Vertical Line)

۱۰ – عملگر NOT (Inverse Logic) ۱۱ – توابع مقایسه‌ای (Comparision)

۱۲ – لبه بالارونده (مورد استفاده برای چندین تیغه)

۱۳ - لبه پایین رونده (مورد استفاده برای چندین تیغه) ۱۴ – پاک کردن خط عمودی

۱۵ – اضافه کردن خط عمودی (مکان نما به خط پایین منتقل می‌شود)

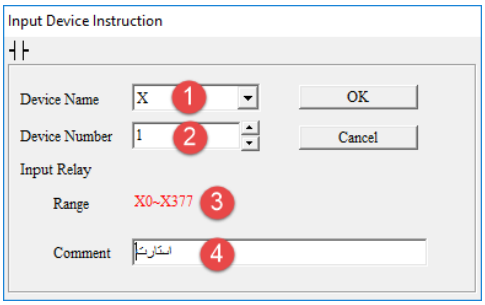
۱۶ – پاک کردن خط عمودی (مکان نما به خط پایین منتقل می‌شود)

درنظر داشته باشید که در PLC های دلتا ، ورودی ها با X و خروجیها با Y نشان داده می‌شوند.

در این قسمت به بررسی چند مثال ساده می پردازیم :

مثال ۳ :

می‌خواهیم با تحریک یک شستی استارت (X1) ، موتور (Y0) روشن شود و با زدن شستی استپ (X2) ، موتور (Y0) خاموش شود. با کلیک بر روی تیغه باز موجود در نوار ابزار، یک پنجره به شکل زیر باز می‌شود .

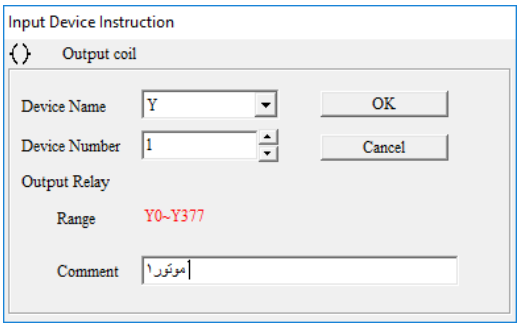


۱ – انتخاب ورودی X به عنوان ورودی . در ادامه به بررسی سایر گزینه ها خواهیم پرداخت.

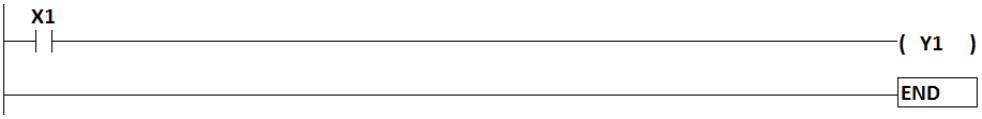
۲ – تعیین شماره ورودی (این ورودی ها به صورت Octal یا هشت تایی می‌باشند)

۳ – محدوده مجاز ورودیهای X

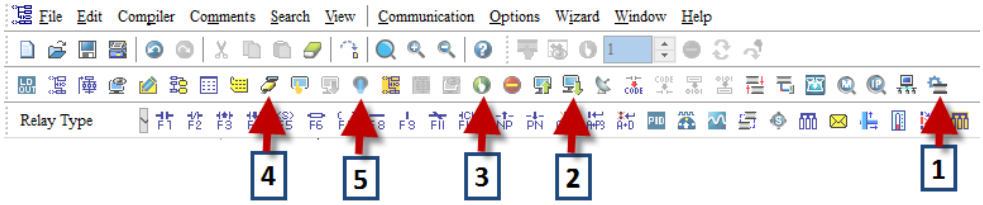
۴ – در این قسمت می‌توانید برای المان مورد نظر یک نام انتخاب کنید که در اینجا از "استارت" استفاده شده است. در انتها با زدن گزینه OK یک تیغه باز در برنامه ظاهر می‌شود. برای خروجی نیز با کلیک روی Coil در نوار ابزار، پنجره‌ای مشابه صفحه بعد باز می‌شود در این پنجره نیز باید نام و شماره خروجی مشخص شود.



برنامه بصورت زیر می‌باشد .



حال نیاز است برنامه را تست کنیم. برای اینکار می‌توانیم از بخش شبیه ساز نرم افزار WPLsoft استفاده کنیم. برای این منظور به ترتیب زیر عمل می‌کنیم.



۱ – ابتدا گزینه Simulator را فعال می‌کنیم.

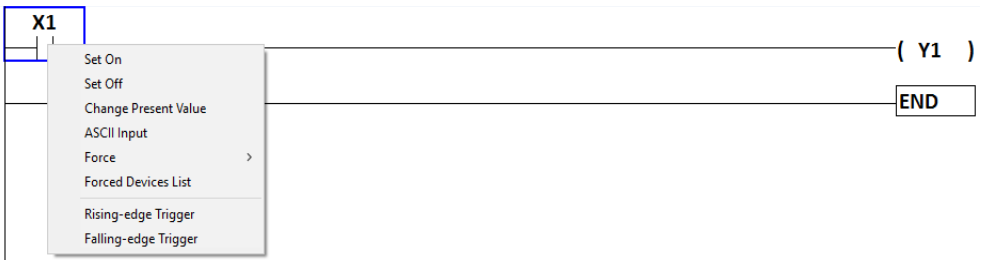
۲ – سپس گزینه Write to PLC را برای دانلود برنامه انتخاب می کنیم .

۳ – سپس برنامه را به حالت Online Mode قرار میدهیم.

۴ – سپس با زدن گزینه Run برنامه PLC را اجرا می‌کنیم.

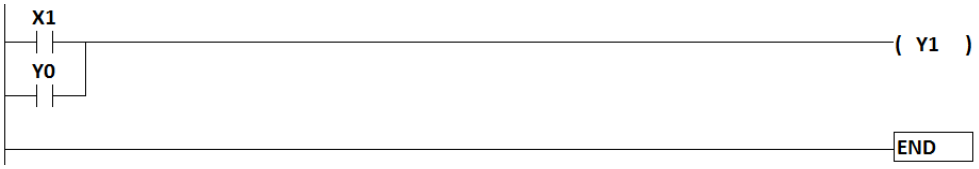
۵ - چنانچه بخواهیم ورودی‌های نوع X را از طریق سیمولاتور فعال کنیم باید این گزینه را فعال کنیم اما اگر ورودی از نوع M باشد نیازی به فعال کردن این ورودی نمی‌باشد .

حال با قرار دادن مکان نمای مستطیل آبی رنگ بر روی ورودی X1 (در حالت انتخاب شده قرار بگیرد) می‌توانیم این ورودی را تحریک کنیم. برای این کار کافیست روی X1 کلیک راست کرده و Set On را بزنیم و برای خارج شدن از حالت تحریک، کافیست روی X1 مجددا کلیک راست کرده و گزینه Set Off را بزنیم.



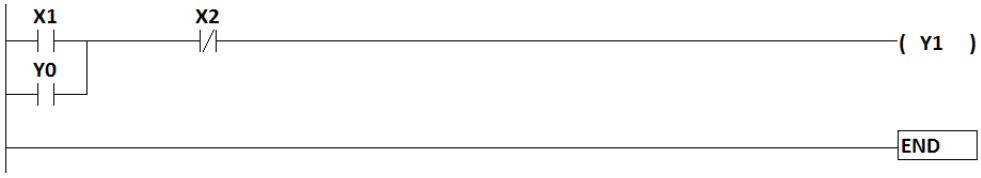
همانطور که ملاحظه می‌کنید با تحریک ورودی استارت (X1)، خروجی (Y1) روشن می‌شود. در نظر داشته باشید که خروجی Y1 تا زمانی روشن خواهد بود که دست ما روی شستی استارت باشد و با برداشتن دست از روی شستی استارت، خروجی Y0 غیرفعال خواهد شد. برای رفع این مشکل در برق صنعتی ، تیغه باز خروجی را با شستی استارت، موازی می‌کنند. برای ایجاد خطوط موازی در دو طرف یک کنتاکت، ابتدا کنتاکت مورد نظر انتخاب شده و سپس از نوار ابزار روی F9 کلیک کرده و یا از کلید میانبر F9 استفاده کنید. با این عمل یک خط عمودی در سمت راست کنتاکت مورد نظر به سمت پایین ایجاد می شود . برای ایجاد یک خط عمودی در سمت چپ کنتاکت، ابتدا سمت چپ کنتاکت، روی خط برنامه، کلیک کرده و مجددا کلید میانبر F9 را انتخاب کنید. برنامه صفحه بعد را ببینید .

نکته : برای ایجاد فاصله بین دو خط برنامه، از CTRL + I و برای حذف فاصله CTRL + Y استفاده کنید.



در تصویر بالا با غیر فعال کردن ورودی X1 ، خروجی Y1 همچنان روشن خواهد ماند.

برنامه ای که در شکل بالا ملاحظه می کنید ، ناقص بوده و برای این مدار نیاز به یک شستی استپ داریم تا بتوانیم در مواقع لزوم خروجی را غیر فعال کنیم لذا می‌توانیم از یک تیغه بسته (X2) برای استپ استفاده کنیم.



با فعال کردن ورودی X2 (استپ) ، خروجی خاموش می‌شود.

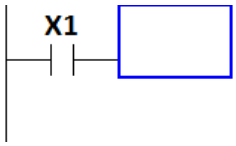
در جدول زیر دستورات المانها را مشاهده می‌کنید که باعث سرعت بخشیدن به برنامه نویسی می‌شود.

میانبر	دستور نوشتاری	ماهیت	شماتیک	نوع المان
F1	LD	NORMALLY OPEN	X0 ⊣ ⊢	تیغه باز
F2	LDI	NORMALLY CLOSE	X0 ⊣/⊢	تیغه بسته
F3	LDP	RISING EDGE	X0 ⊣↑⊢	لبه بالارونده
F4	LDF	FALLING EDGE	X0 ⊣↓⊢	لبه پایین رونده
F6		APPLICATION INSTRUCTIONS		دستورات
F7	OUT	OUTPUT COIL	−(Y0)	خروجی
F8		HORIZONTAL LINE	———	خط واصل افقی
F9		VERTICAL LINE	|	خط واصل عمودی

به عنوان مثال برای نوشتن دستور مثال قبل به روش نوشتاری مطابق شکل زیر عمل می‌کنیم:

نشانگر را ابتدای خط قرار داده و شروع به تایپ دستور می‌کنیم. پس از نوشتن عبارت LD X1، دکمه‌ی Enter را زده یا روی گزینه Ok کلیک می‌کنیم.

همانطور که ملاحظه می‌کنید المان تیغه ورودی در برنامه ثبت شده و نشانگر به یک خانه جلوتر انتقال یافت .



حال باید دستور نوشتاری خروجی را مطابق شکل زیر تایپ کنیم.

حال با زدن دکمه Enter بر روی کیبورد این المان در برنامه اعمال می‌شود.

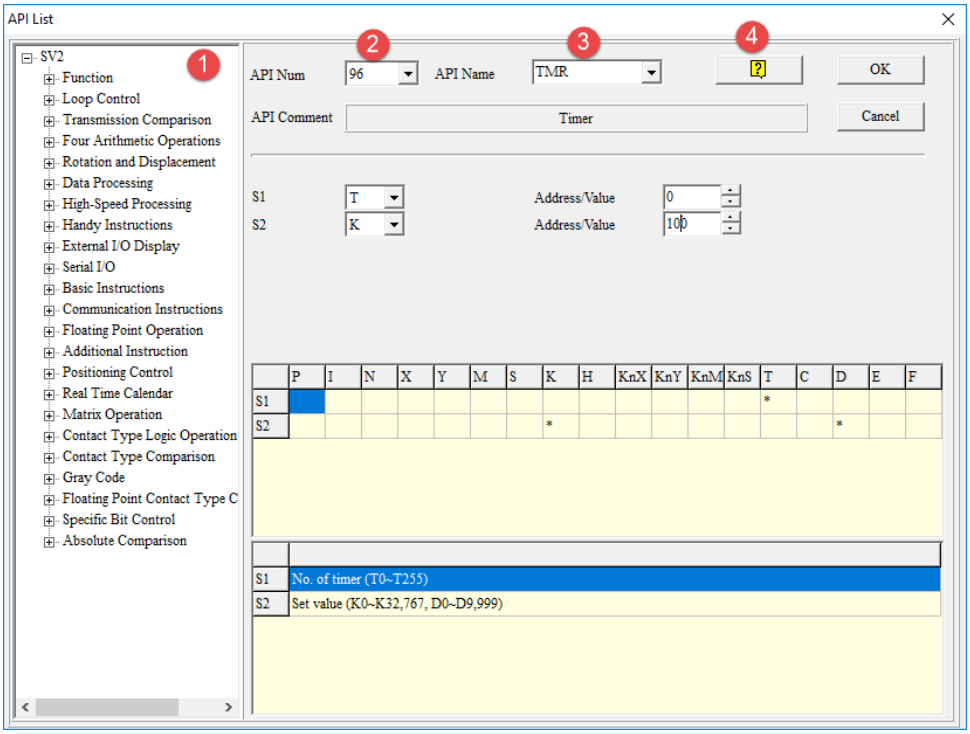
به همین ترتیب می‌توانیم ادامه برنامه را نیز به صورت نوشتاری تایپ کرده و سرعت برنامه نویسی خود را با تمرین زیاد افزایش دهیم .

دستورات پرکاربرد در PLC های دلتا :

همانند سایر برندها، در PLC های دلتا دستوراتی وجود دارد که کاربرد فراوانی در برنامه نویسی دارد لذا در این بخش به معرفی این دستورات برنامه نویسی می‌پردازیم.

برای دسترسی به دستورات موجود در برنامه WPLSoft می‌توانید از کلید میانبر F6 استفاده کرده و یا در قسمت نوار ابزار برنامه، روی آیکون نشان داده شده در شکل زیر کلیک کنید.

پس از کلیک بر روی آیکون نشان داده شده پنجره ای مطابق صفحه بعد باز خواهد شد.



۱ – در این قسمت می‌توانید به تمامی دستورات PLC دسترسی داشته باشید.

۲ – در این قسمت می‌توانید در صورتیکه شماره دستور مورد نظرتان را می‌دانید آنرا جستجو کنید.

۳ – در این قسمت می‌توانید در صورتیکه نام دستور مورد نظرتان را می‌دانید آنرا جستجو کنید.

۴ – در این قسمت می‌توانید اطلاعاتی را در مورد دستور مورد نظر خود بدست بیاورید.

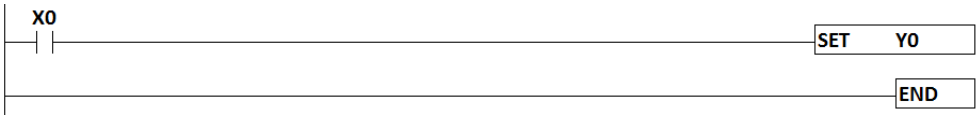
پس از یافتن دستور مورد نظر باید اطلاعات تکمیلی آنرا در فیلد های دیگر جدول وارد کنید. در پایان می‌توانید با زدن گزینه Ok به دستور مورد نظر خود دسترسی پیدا کنید.

دستور SET :

در برنامه زیر بجای استفاده از خروجی (Out Y0) از دستور (Set Y0) استفاده می کنیم.

مثال ۴ :

در این مثال با فعال کردن X0 خروجی Y0 روشن شده و با غیر فعال کردن ورودی X0 ، خروجی Y0 همچنان فعال باقی می‌ماند و دیگر نیازی به اضافه کردن تیغه خود نگهدار در برنامه نداریم.

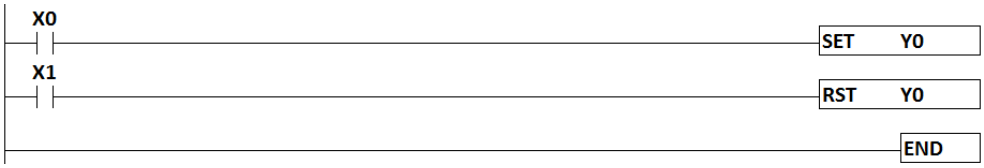


دستور RST :

نکته: در مثال قبل که از دستور Set استفاده شده ، برای غیر فعال کردن خروجی Y0 باید از دستور ریست RST استفاده شود و این تنها راه غیر فعال کردن خروجی Y0 است.

مثال ۵ :

در برنامه زیر با فعال شدن لحظه ای ورودی X0 ، خروجی Y0 بطور دائم روشن می ماند. سپس با فعال کردن لحظه‌ای X1 ، خروجی خاموش می شود .



مثال ۶ :

برنامه ای بنویسید که با فعال کردن یک ورودی (مثلا X1) چهار خروجی (مثلا Y1 و Y2 و Y3 و Y4) روشن شده و روشن بمانند و با زدن یک ورودی دیگر(مثلا X2) تمامی خروجی ها خاموش شوند.

نکته : در این مثال در صورتیکه ورودیهای X1 و X2 همزمان فعال شوند اولویت با X2 خواهد بود .

نکته : هر دستوری که به انتهای برنامه نزدیکتر باشد اولویت بالاتری دارد .

دستور ZRST :

این دستور برای ریست کردن ناحیه‌ای از خروجی های ست شده، کاربرد دارد و باعث کم شدن حجم برنامه می‌شود.

مثال ۷ :

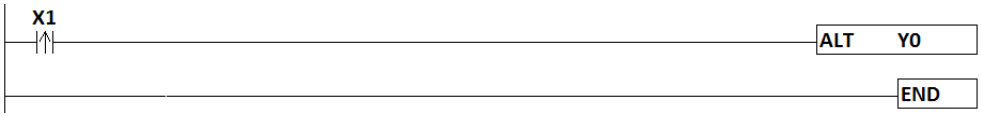
مثال صفحه قبل را در نظر بگیرید. با استفاده از این دستور می‌توان چندین خروجی با شماره های پشت سر هم را خاموش کرد. به این صورت که ابتدا دستور ZRST را نوشته و سپس اولین المان از بازه مورد نظر و سپس آخرین المان از بازه مورد نظر را می نویسیم. در برنامه زیر، با فعال شدن X1 ، تمام خروجی های مابین Y1 و Y4 خاموش خواهند شد.

دستور ALT :

این دستور می‌تواند وضعیت خروجی را معکوس کند. یعنی در صورتیکه خروجی روشن باشد (وضعیت 1) آنرا خاموش و اگر خاموش بود (وضعیت 0) آنرا روشن کند.

نکته : برای استفاده از این دستور باید ورودی را از نوع لبه بالارونده (Rising Edge) یا پایین رونده (Falling Edge) انتخاب کنید و یا بجای دستور ALT از دستور ALTP استفاده کنید به این دلیل که در صورت تحریک ورودی X تعداد زیادی پالس به دستور ALT رسیده و باعث ناپایداری این دستور می‌شود لذا از تیغه‌ی حساس به لبه و یا دستور ALTP استفاده می‌شود تا تنها یک پالس در هنگام تحریک ورودی به دستور برسد.

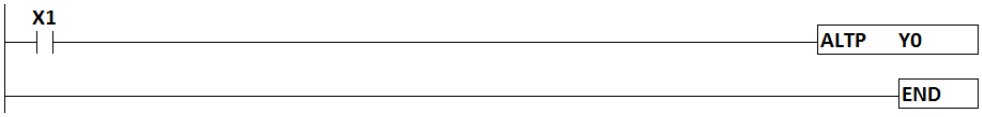
مثال ۸ :



با هر بار تحریک ورودی X1 ، وضعیت خروجی Y0 تغییر می کند .

در مثال بالا بجای استفاده از ورودی لبه دار X1 می‌توانید از دستور لبه دار ALTP استفاده کنید.

مثال ۹ :



در مثال صفحه قبل با تحریک X1 ، (منظور از تحریک ، SET ON و SET OFF ورودی است) خروجی Y0 روشن می‌شود. و با تحریک بعدی X1 ، خروجی Y0 خاموش می‌شود. در حقیقت اگر خروجی روشن باشد ، خاموش شده و اگر خروجی خاموش باشد ، روشن می‌شود.

دستور المانهای حساس به لبه پالس :

در بسیاری از موارد نیاز است در لحظه وصل یک کنتاکت، و یا لحظه قطع یک کنتاکت، یک پالس موقت تولید شود . این قابلیت کاربردهای فراوانی در صنعت دارد. مثال زیر را در نظر بگیرید.

دستور المان حساس به لبه بالا رونده :

مثال ۱۰ :

در خط یک برنامه، به محض فعال شدن ورودی X0 (لبه بالا رونده)، پالسی به مدت یک سیکل، خروجی Y0 را روشن می کند . ورودی X0 را غیر فعال کرده و مجددا ورودی X0 را فعال می کنیم دوباره یک پالس به مدت یک سیکل تولید خواهد شد . کلید میانبر این دستور F3 می باشد .

نکته : منظور از یک سیکل، یک دور اجرای کامل برنامه می باشد . شکل نوشتاری دستور فوق بصورت LDP X0 می باشد . کلیه المانهای می توانند حساس به لبه بالارونده پالس باشند مانند LDP M0 .

نکته : چنانچه بخواهیم در صورت تحریک چند المان ورودی، یک پالس با لبه بالارونده تولید شود از دستور ⊼ استفاده می شود . مثال زیر را ببینید .

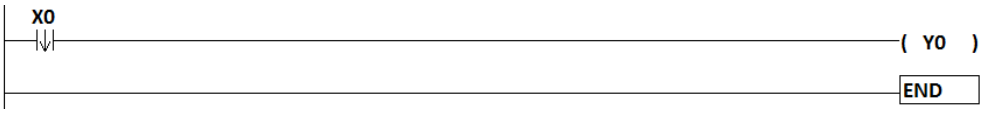
مثال ۱۱ :

در این مثال چنانچه ورودی X0 و ورودی X1 وصل شوند یک پالس با مدت زمان یک سیکل تولید خواهد شد . فرض کنید ابتدا ورودی X1 وصل شده باشد، به محض وصل ورودی X0 (لبه بالارونده) ، خروجی Y0 به مدت یک سیکل روشن و سپس خاموش خواهد شد . فرم نوشتاری این دستورNP می باشد .

چندین ورودی با المانهای مختلف می توانند بصورت سری و یا موازی قبل از دستور NP وجود داشته باشند .

دستور المان حساس به لبه پایین رونده :

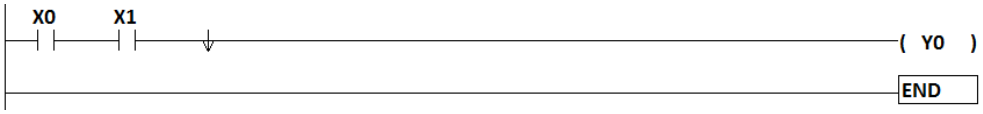
مثال ۱۲ :



در این برنامه، با فعال شدن ورودی X0 ، هیچ پالسی تولید نمی شود اما به محض غیر فعال کردن ورودی X0 ،(لبه پایین رونده) ، پالسی به مدت یک سیکل، تولید شده و خروجی Y0 را روشن می‌کند . شکل نوشتاری این دستور LDF X0 می باشد . بعد از پایان یک سیکل خروجی خاموش می شود . کلید میانبر این دستور F4 می‌باشد .

نکته : چنانچه بخواهیم در صورت تحریک چند المان ورودی، یک پالس با لبه پایین رونده تولید شود از دستور ⊣↓⊢ استفاده می شود . مثال زیر را ببینید .

مثال ۱۳ :



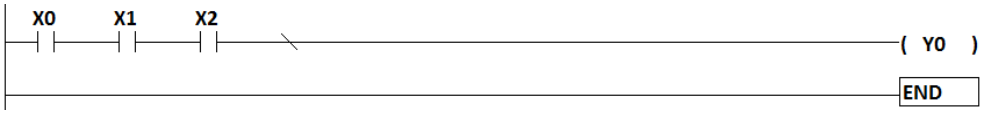
در این مثال چنانچه ورودی X0 و ورودی X1 وصل شوند هیچ پالسی تولید نخواهد شد . فرض کنید ورودیهای X0 و X1 وصل شده باشند، به محض قطع یکی از ورودیها مثلا X1 (لبه پایین رونده) ، خروجی Y0 به مدت یک سیکل روشن و سپس خاموش خواهد شد . فرم نوشتاری این دستورPN می باشد .

چندین ورودی با المانهای مختلف می توانند بصورت سری و یا موازی قبل از دستور PN وجود داشته باشند .

دستور NOT :

چنانچه بخواهیم منطق برنامه را معکوس کنیم از دستور ⊣/⊢ استفاده می کنیم . فرم نوشتاری این دستور INV می باشد . مثال زیر را ببینید .

مثال ۱۴ :



در این برنامه چنانچه یکی یا همه ورودیها غیر فعال باشند دستور INV منطق برنامه را معکوس کرده و خروجی Y0 روشن می شود . در صورتیکه هر سه ورودی وصل باشند خروجی Y0 خاموش خواهد بود .

تایمرها :

در PLC های دلتا دستورات متعددی در رابطه با زمان سنجی(Timer) وجود دارد که هریک کاربرد متفاوتی دارند. در این بخش می‌خواهیم در رابطه با برخی از این تایمرها توضیحات و مثالهایی را ارائه کنیم.

نکته : تایمرها برحسب شماره‌ی خود به دسته های مختلفی تقسیم می‌شوند که به شرح زیر است:

- ✓ تایمر با ضریب(رزولوشن) ۱۰۰ میلی ثانیه
- ✓ تایمر با ضریب(رزولوشن) ۱۰ میلی ثانیه
- ✓ تایمر با ضریب(رزولوشن) ۱ میلی ثانیه

تایمر های 100ms حداقل زمانی را که می‌توانند زمان سنجی کنند برابر 100ms می‌باشد . به عنوان نمونه، مطابق جدول مشخصات تایمرهای PLC مدل 14SS2 ، محدوده تایمرهای 100ms از T0 ... T63 می‌باشد . محدوده تایمرهای 10ms از T126 ... T64 می‌باشد . و تایمر T127 1ms می‌باشد برای اطلاع از مشخصات تایمر PLCهای به فایل راهنمای آن مراجعه کنید .

نکته : مطابق جدول زیر که مربوط به PLC مدل 14SS2 می‌باشد، در صورتیکه M1028 فعال باشد (OFF) ، تایمرهای T126 ... T64 تبدیل به تایمرهای 10ms می‌شود .

T	Timer	100ms (M1028=ON, T64~T126: 10ms)	T0~T126, 127 points, (*1) T128~T183, 56 points, (*1)	Total 256 points
			T184~T199 for Subroutines, 16 points, (*1)	
			T250~T255(accumulative), 6 points (*1)	
		10ms (M1038=ON, T200~T245: 1ms)	T200~T239, 40 points, (*1)	
			T240~T245(accumulative), 6 points, (*1)	
		1ms	T127, 1 points, (*1) T246~T249(accumulative), 4 points, (*1)	

مثال ۱۵ : دستور تایمر روبرو را در نظر بگیرید: **TMR T0 K50**

مطابق جدول بالا شماره تایمر T0 جزء تایمرهای 100ms می‌باشد لذا مدت زمانیکه دستور فوق زمان سنجی می کند برابر است با : k100 × 50 = 5000ms = 5sec

طبقه بندی این تایمرها بر حسب شماره، در هر CPU متفاوت است که می‌توانید در فایل مخصوص هر CPU ملاحظه کنید.

دستور TMR :

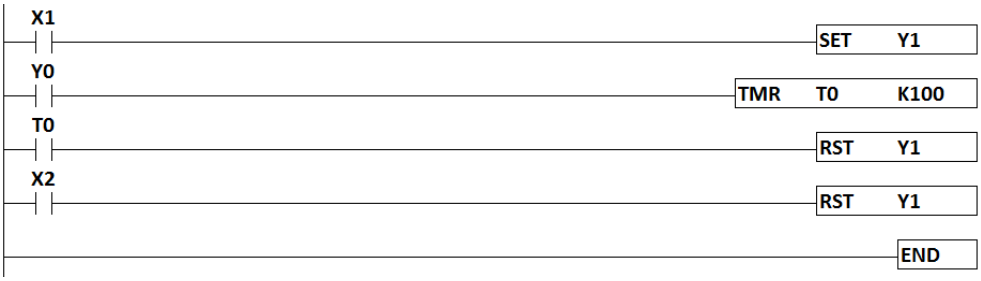
از دستورات پرکاربرد زمان سنجی، در بین سایر دستورات این خانواده می‌باشد. با استفاده از این دستور می‌توانیم تایمر های تاخیر در وصل (ON DELAY) و تاخیر در قطع (OFF DELAY) را بسازیم.

نکته : دستور TMR به ورودی خود وابسته می‌باشد و در صورتیکه ورودی این دستور غیر فعال شود تایمر متوقف می‌شود. به روشهای مختلف می‌توان از تایمر در برنامه استفاده کرد :

۱ – ابتدا یک خروجی مثلا Y0 را SET کرده و از تیغه‌ی باز آن برای تایمر استفاده کنیم که در مثال زیر خواهید دید.

مثال ۱۶ :

می‌خواهیم با زدن شستی استارت (مثلا X1) ، موتور (مثلا Y1) روشن شود و پس از ۱۰ ثانیه خاموش گردد.



همانطور که در شکل می بینید، با تحریک X1 خروجی Y1 فعال شده و تیغه‌ی Y1، تایمر T0 را فعال کرده و تایمر در حال شمارش بر حسب میلی ثانیه(100ms) می‌باشد. پس از اتمام زمان تعیین شده برای تایمر، تیغه‌ی داخلی تایمر(T0)، تغییر وضعیت داده و یک می‌شود (بسته می‌شود) لذا از تیغه‌ی تایمر برای خاموش کردن (RST) خروجی Y1 استفاده شده است. همچنین در مثال بالا X2 بعنوان استپ در نظر گرفته شده است .

نکته : همانطور که در شکل بالا ملاحظه می‌کنید مدت زمان تایمر با K100 نشان داده شده است و این مدت زمان چون بر حسب میلی ثانیه می‌باشد، برابر ۱۰ ثانیه است.

مثال ۱۷ :

می‌خواهیم برنامه ای بنویسیم که ۱۰ ثانیه بعد از تحریک شستی (مثلا X1) خروجی (مثلا Y1) روشن شود.

برنامه صفحه بعد را ببینید .

در این مثال برای خاموش کردن خروجی Y1 و رله کمکی M1 ، از ورودی X2 استفاده شده است.

مثال ۱۸ :

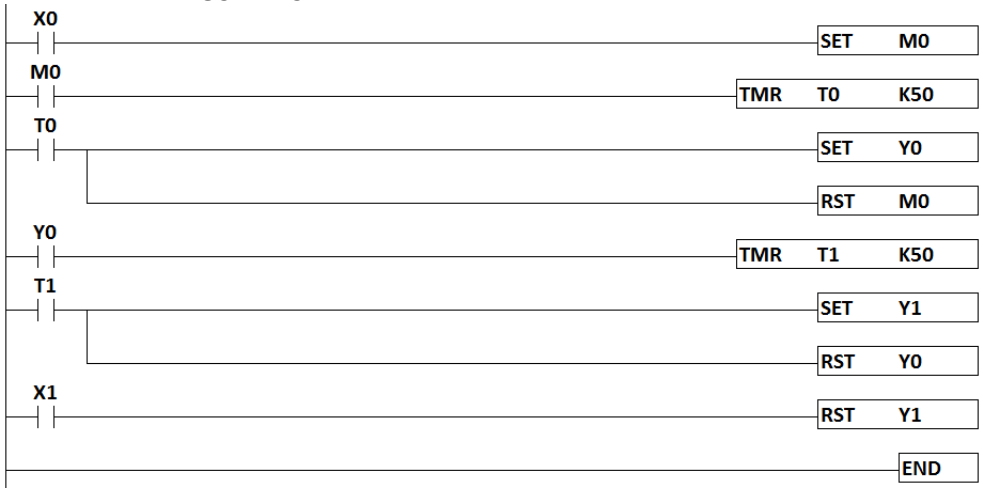
برنامه ای بنویسید که با تحریک ورودی X2 خروجی Y2 فعال شود و بعد از ۳ ثانیه خروجی Y3 فعال شود و پس از ۵ ثانیه هر دو خروجی خاموش شوند.

همانطور که ملاحظه می‌کنید با تحریک X2 خروجی Y2 فعال شده و از تیغه‌ی باز Y2 برای تایمر T0 استفاده شده است. تیغه‌ی این تایمر بعد از زمان تعیین شده که برابر ۳ ثانیه است، فعال شده و خروجی Y3 را فعال می‌کند. از تیغه‌ی باز Y3 برای فعال کردن تایمر T1 استفاده شده است بعد از ۵ ثانیه تیغه‌ی تایمر T1 فعال شده و خروجی های Y1 و Y2 را ریست می‌کند .

نکته : در مثال های قبلی بعد از تحریک X0 (SET ON) ، باید ورودی را (SET OFF) کنیم تا عملکردی شبیه شستی داشته باشد .

مثال ۱۹ :

برنامه‌ای بنویسید که ۵ ثانیه بعد از تحریک ورودی X0 ، خروجی Y0 فعال شود و ۵ ثانیه بعد، خروجی دوم Y1 فعال شود و همزمان خروجی اول Y0 غیر فعال شود. از X1 به عنوان Stop استفاده کنید.
برنامه در صفحه بعد آمده است .

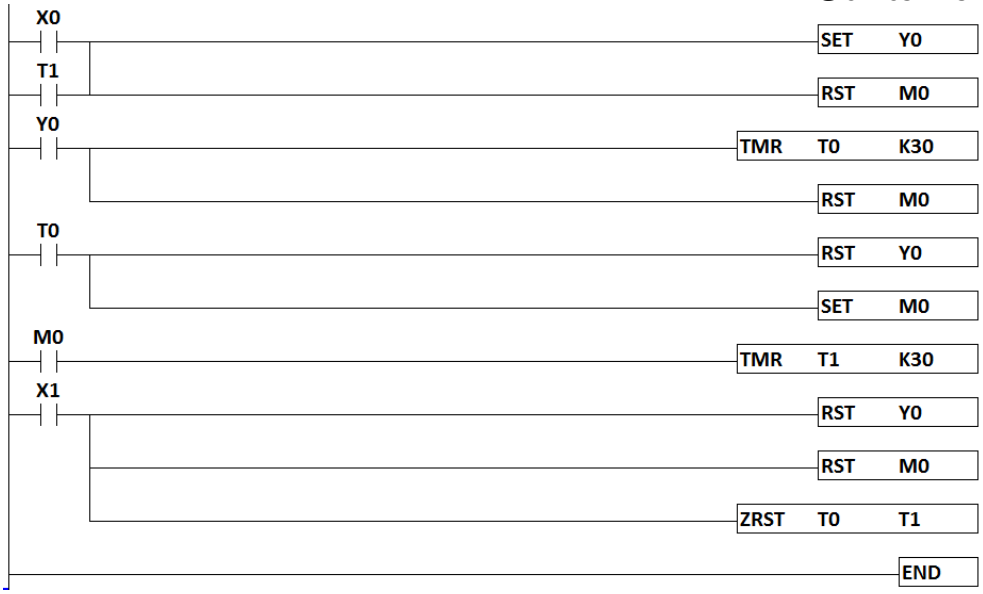


نکته : در این مثالها ورودیهای X0 و X1 به عنوان پوش باتن یا شستی در نظر گرفته شده اند لذا بعد از Set On کردن آنها ، فورا آنها را Set Off کنید .

در این مثال رله کمکی M0 به عنوان یک حافظه کمکی در نظر گرفته شده است . به این رله‌های کمکی، فلگ نیز گفته می‌شود . این رله های کمکی ماهیت سخت افزاری ندارند .

مثال ۲۰ :

دربرخی از صنایع و ماشین های صنعتی، یک پروسه به صورت تکراری انجام می‌شود. در این مثال می‌خواهیم برنامه‌ای بنویسیم تا با تحریک یک ورودی مثلا (X0) ، یک پروسه به صورت تکراری انجام شود. مثلا یک چراغ (Y0) شروع به چشمک زدن کند. فاصله‌ی بین روشن و خاموش بودن این چراغ ۳ ثانیه تعیین شود.

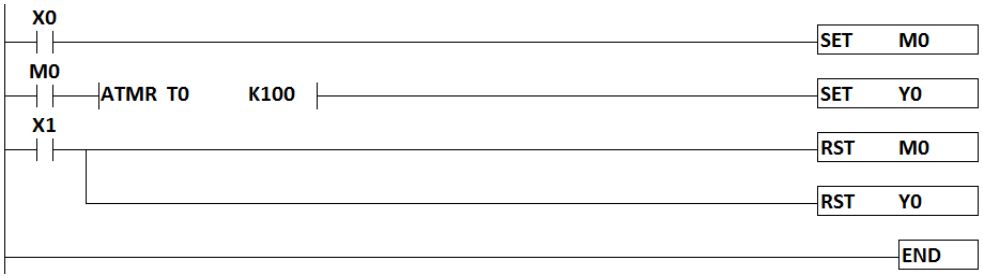


دستور ATMR :

عملکرد این تایمر شبیه تایمر TMR می‌باشد با این تفاوت که می‌تواند مستقیما یک خروجی را فعال کند مثال زیر را ببینید .

مثال ۲۱ :

برنامه ای بنویسید که بوسیله ی آن ۱۰ ثانیه پس از تحریک ورودی X0 ، خروجی Y0 فعال شود.



با تحریک لحظه‌ای X0 ، فلگ M0 روشن شده و تایمر T0 نیز فعال می‌شود . بعد از گذشت 10 ثانیه، خروجی Y0 روشن می‌شود با تحریک X1 خروجی خاموش می‌شود . در این برنامه می‌توانیم بجای Set Y0 از Out Y0 استفاده کنیم در این صورت برنامه نیازی به دستور RST Y0 نخواهد داشت.

استفاده از دیتا رجیسترها (D) بجای اعداد ثابت :

در برنامه نویسی می‌توان بجای استفاده از اعداد ثابت که با پیشوند k نمایش داده می‌شوند، از یک دیتا رجیستر استفاده کرد. دیتا رجیسترها نیز مانند حافظه‌ها دارای انواع مختلف هستند. مشخصات دیتا

رجیسترها در جدول حافظه‌ها در فایل راهنمای هر PLC موجود است. دیتا رجیستر ها با حرف D نمایش داده می‌شوند.

در صورت استفاده از دیتا رجیسترها بجای مقادیر ثابت که با پیشوند K یا H و یا F نمایش داده می‌شوند، می‌توانید توسط نمایشگر (HMI) اعداد را تغییر داده و یا فراخوانی کنید.

مثلا در مثال بالا در دستور کانتر می‌توانید بجای عدد ثابت K10 یا K40000 از یک دیتا رجیستر مانند D0 استفاده کنید. سپس مقدار این دیتا رجیستر را در HMI توسط یک تگ NUMERIC ENTERY تغییر دهید یا توسط یک تگ NUMERIC DISPLAY فراخوانی کنید.

نکته : در دستورات ۳۲ بیتی ، ۲ دیتا رجیستر بصورت متوالی اشغال می‌شود .

D	Data register	General	D0~D407, 408 words, (*1) D600~D999, 400 words, (*1) D3920~D4999, 1080 words, (*1)	Total 5000 points
		Latched	D408~D599, 192 words, (*2) D2000~D3919, 1920 words, (*2)	
		Special	D1000~D1999, 1000 words, some are latched	
		Index	E0~E7, F0~F7, 16 words, (*1)	

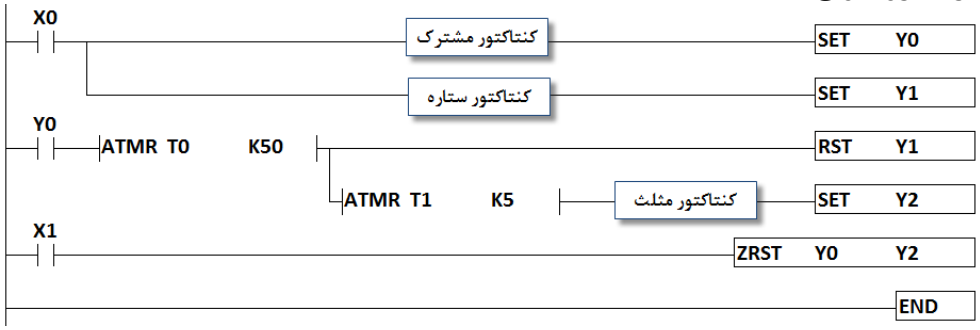
مثال ۲۲ :

در مثال قبل دیدیم که با تحریک X0 بعد از مدت زمان ثابت ۱۰ ثانیه خروجی Y0 فعال می‌شود. در این مثال می‌خواهیم برنامه را طوری بنویسیم که مدت زمان را بطور دلخواه تغییر دهیم .

در این برنامه بجای استفاده از k100 که یک مقدار ثابت است از D0 استفاده شده است. شما می‌توانید از طریق (HMI)، یک Numeric Entery به عنوان ورودی عدد و یا نمایش عدد تعریف کنید و مقدار D0 را از آن طریق مشخص کنید.

مثال ۲۳ :

برنامه‌ی راه اندازی یک موتور به روش ستاره - مثلث را با استفاده از PLC دلتا بنویسید به طوری که بعد از استارت X0 ، مدار بصورت ستاره کار کرده، و بعد از ۵ ثانیه به حالت مثلث برود .

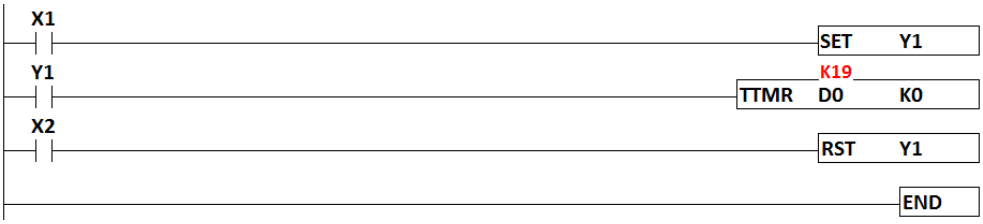


در این مثال با تحریک X0 (START) ، کنتاکتور مشترک Y0 و کنتاکتور ستاره Y1 ، همزمان وارد مدار می شوند. پس از گذشت مدت زمان تعیین شده برای تایمر T0 به مقدار ۵ ثانیه، کنتاکتور ستاره غیر فعال می‌شود و پس از 0.5sec ثانیه (تایمر T1) ، کنتاکتور مثلث Y2 وارد مدار می‌شود. این فاصله ی زمانی 0.5sec به این دلیل است که به علت سرعت بالای PLC امکان تداخل در کنتاکتور ستاره و مثلث وجود دارد و با قرار دادن این تاخیر زمانی از خطر اتصال همزمان کنتاکتورهای ستاره و مثلث جلوگیری می‌شود. البته این زمان نباید خیلی طولانی باشد. در هر مرحله با تحریک X1 (STOP) می‌توان مدار را خاموش کرد.

دستور TTMR :

با استفاده از این دستور می‌توان مدت زمان روشن بودن یک خروجی را برحسب ثانیه اندازه گیری نمود مثال زیر را ببینید .

مثال ۲۴ :



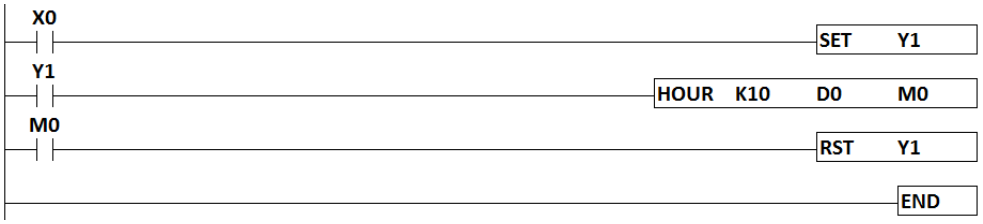
در این مثال اگر K0 بگذاریم ، تایمر برحسب ثانیه عمل می‌کند. اگر K1 بگذاریم ، تایمر برحسب ۱۰۰ میلی ثانیه عمل می‌کند. اگر K2 بگذاریم ، تایمر برحسب ۱۰ میلی ثانیه عمل می‌کند .

با تحریک لحظه‌ای ورودی X1 خروجی Y1 روشن شد و تایمر نیز فعال می‌شود و زمان روشن بودن خروجی را متناسب با واحد اندازه گیری زمان (K2 = 10ms , K1 = 100ms , K0= sec) در رجیستر D0 ثبت می کند . چنانچه بخواهیم مقدار اندازه گیری شده بعد از قطع برق ، از دست نرود باید از رجیسترهای پایدار استفاده کنیم . به عنوان مثال رجیستر D408 پایدار می‌باشد . در این برنامه از X2 به عنوان Stop استفاده می‌کنیم.

دستور HOUR :

این دستور به ساعت سنج معروف است. با استفاده از این دستور می‌توان مدت زمان روشن یا خاموش بودن یک خروجی را بر حسب ساعت تعیین نمود . به مثال زیر توجه کنید:

مثال ۲۵ :



در این مثال با فعال شدن X0 ، خروجی Y0 (مثلا واترپمپ) روشن می‌شود. دستور HOUR بعد از گذشت ۱۰ ساعت، M0 را فعال می‌کند و باعث خاموش شدن پمپ، جهت سرویس می‌شود. در این دستور مدت زمان کارکرد ماشین بر حسب ساعت در رجیستر D0 و بر حسب ثانیه در رجیستر D1 ذخیره می‌شود. این دستور دو رجیستر متوالی D0 و D1 را اشغال می‌کند .

دستور STMR :

این دستور چهار خروجی مختلف را اشغال کرده و هر یک از خروجیها عملکرد مستقلی دارند.

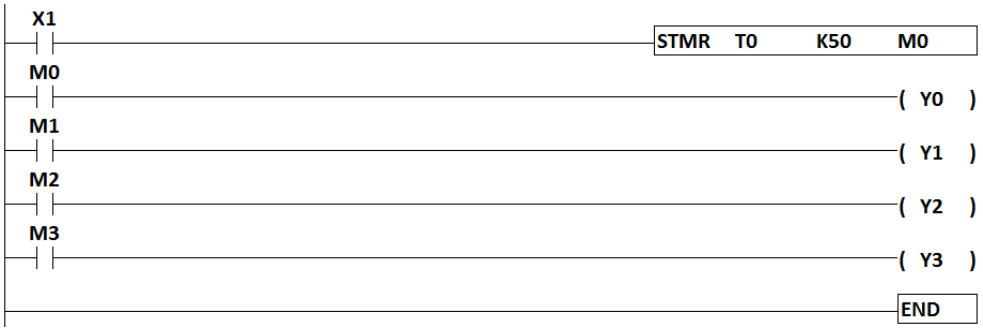
مثال ۲۶ :

خروجی اول : با فعال شدن X0 خروجی Y0 فعال می‌شود و با غیر فعال شدن X0 خروجی Y0 پس از ۵ ثانیه غیر فعال می‌شود.

خروجی دوم : با غیر فعال شدن X0 (لبه پایین رونده) خروجی Y1 به مدت ۵ ثانیه روشن شده و سپس خاموش می‌شود.

خروجی سوم : با فعال شدن X0 خروجی Y2 به مدت ۵ ثانیه روشن شده و سپس خاموش می‌شود.

خروجی چهارم : با فعال شدن X0 خروجی Y3 پس از ۵ ثانیه روشن می‌شود و با غیر فعال شدن X0 خروجی Y3 بعد از ۵ ثانیه غیر فعال می‌شود.



دستور خواندن زمان داخلی PLC با دستور TRD :

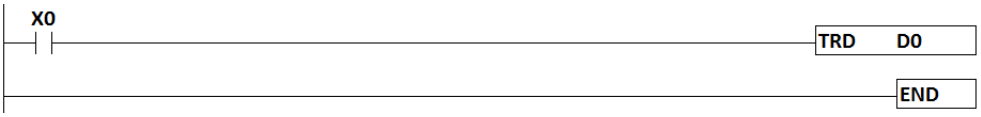
این دستور برای خواندن زمان داخلی PLC و یا به عبارتی (RTC) استفاده می‌شود. باید در نظر داشت که با استفاده از این دستور تعداد ۷ دیتا رجیستر اشغال می‌شود.

Special D	Item	Content		Normal D	Item
D1319	Year (A.D.)	00~99	→	D0	Year (A.D.)
D1318	Day (Mon.~Sun.)	1~7	→	D1	Day (Mon.~Sun.)
D1317	Month	1~12	→	D2	Month
D1316	Day	1~31	→	D3	Day
D1315	Hour	0~23	→	D4	Hour
D1314	Minute	0~59	→	D5	Minute
D1313	Second	0~59	→	D6	Second

مطابق جدول بالا، زمان واقعی PLC یا همان RTC ، شامل سال، هفته، ماه، روز، ساعت، دقیقه، ثانیه بوده و از طریق رجیسترهای مخصوص RTC یعنی D1319 الی D1313 قابل دسترسی می‌باشد . همچنین از طریق دستور TRD نیز می‌توان مقادیر رجیسترهای فوق را در رجیسترهای دیگری منتقل کرد به عنوان مثال مطابق جدول بالا، دستور TRD D0 ، رجیسترهای D1319 الی D1313 را به ترتیب در رجیسترهای D0 الی D6 منتقل میکند . البته با استفاده از دستور MOV هم می‌توان این انتقال را انجام داد اما تعداد دستورات بیشتر از زمانی خواهد که از دستور TRD اااستفاده شود .

مثال ۲۷ :

می‌خواهیم تاریخ و ساعت داخلی PLC را بخوانیم. برای این کار کافیست دستور TRD را به شکل زیر بنویسیم.



در این مثال با فعال شدن X0 تاریخ PLC خوانده شده و در رجیسترهای D0 تا D6 ریخته می‌شود. محتوای این رجیستر ها عبارتند از :

D0 : سال **D1 :** روز های هفته **D2 :** ماه

D3 : هفته **D4 :** ساعت **D5 :** دقیقه

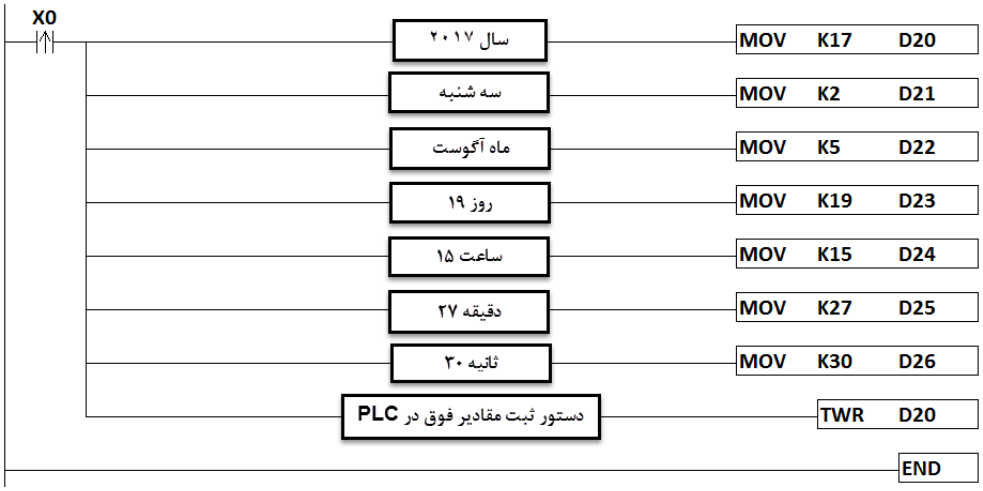
D6 : ثانیه

پس از نوشتن این برنامه، می‌توان رجیستر ها را در HMI فراخوانی کرد و تاریخ PLC را مشاهده نمود همچنین میتوان با استفاده از دستورات PLC متناسب با زمان مورد نظر فرامینی صادر نمود. زمان داخلی PLC یا همان RTC با قطع برق توسط باطری پشتیبان PLC تا مدت محدودی حفظ می‌شود .

دستور تنظیم زمان داخلی TWR :

برای تنظیم تاریخ و ساعت داخلیPLC یا همان RTC از این دستور استفاده می‌شود. برای استفاده از این دستور مطابق مثال زیر عمل می‌کنیم:

مثال ۲۸ :



با فعال شدن X0 تاریخ سه شنبه ۱۹ آگوست سال ۲۰۱۷ ساعت ۱۵:۲۷:۳۰ در حافظه‌ی PLC بخش RTC ذخیره می‌شود.

دستور شمارنده CNT :

این دستور به ازای هر پالس در ورودی خود یک واحد به مقدار قبلی اضافه می‌کند. کانترها به دسته های ۱۶ بیتی ، ۳۲ بیتی، کانتر های سرعت بالا(سخت افزاری و نرم افزاری) تقسیم می‌شوند. کانترها نیز مانند تایمرها بر اساس شماره کانتر، به این دسته‌ها تقسیم بندی می‌شوند که باید اطلاعات آنها را از فایل راهنمای هر PLC استخراج کرد. برای مثال کانترهای PLC مدل DVP14SS211R/T به شرح زیر می‌باشد:

16 بیتی:

از C0 – C99 ناپایدار (با قطع برق مقدارش شمارش شده پاک می‌شود)

از C100 – C199 پایدار(با قطع برق مقدار آن حفظ می‌شود)

32 بیتی :

C200 – C254 C200 – C215 ناپایدار C216 – C234 پایدار

C235 – C254 کانتر سرعت بالا(HSC)

C235 – C242 نرم افزاری C243 – C254 سخت افزاری

مشخصات کامل کانترهای PLC مدل 14SS2 مطابق جدول زیر می‌باشد .

Counter	16-bit count up		C0~C111, 112 points, (*1) C128~C199, 72 points, (*1)	Total 233 points
			C112~C127, 16 points, (*2)	
	32-bit count up/down		C200~C223, 24 points, (*1)	
			C224~C232, 9 points, (*2)	
	32bit high-speed count up/down	Soft-ware	C235~C242, 1 phase 1 input, 8 points, (*2)	Total 22 points
			C233~C234, 2 phase 2 input, 2 points, (*2)	
		Hard-ware	C243~C244, 1 phase 1 input, 2 points, (*2)	
			C245~C250, 1 phase 2 input, 6 points, (*2)	
			C251~C254 2 phase 2 input, 4 points, (*2)	

مطابق جدول ، کانترهای 16 بیتی بالا شمار هستند در حالیکه کانترهای 32 بیتی بالا و پایین شمار هستند . فلگ‌هایی جهت تنظیم بالا و یا پایین شمار بودن کانترها در نظر گرفته شده است همچنین حساس کردن کانترها به لبه پایین رونده و یا لبه بالا رونده توسط فلگ های خاصی قابل انجام می‌باشد.

C / X	1-phase input								2 phase 2 input		
	C235	C236	C237	C238	C239	C240	C241	C242	C232#2	C233	C234
X0	U/D								A		
X1		U/D									
X2			U/D						B		
X3				U/D							
X4					U/D					A	
X5						U/D				B	
X6							U/D				A
X7								U/D			B
R/F	M1270	M1271	M1272	M1273	M1274	M1275	M1276	M1277	-	-	-
U/D	M1235	M1236	M1237	M1238	M1239	M1240	M1241	M1242	-	-	-

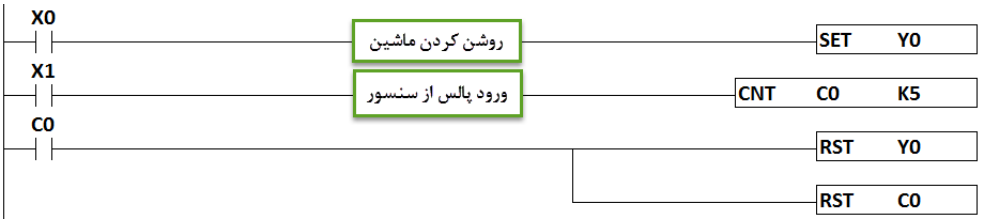
U: Count up D: Count down A: Phase A input B: Phase B input

مطابق جدول فوق ، هر کانتر سرعت بالا ، ورودیهای مخصوص به خود را دارد به عنوان مثال کانتر سرعت بالای c235 ، از ورودی X0 جهت دریافت پالس استفاده می کند. بالا شمار بودن (U) و یا پایین شمار بودن (D) توسط فلگ M1235 تعیین می‌شود در صورتیکه این فلگ Off باشد ، کانتر بالاشمار بوده و چنانچه ON باشد ، کانتر پایین‌شمار خواهد بود . همچنین در صورتیکه فلگ M1270 غیر فعال(Off) باشد ، کانتر حساس به لبه بالارونده و چنانچه این فلگ فعال(ON) باشد، کانتر حساس به لبه پایین رونده خواهد بود .

نکته : بعضی از کانترهای سرعت بالا ، دو ورودی را به خود اختصاص می دهند بطور مثال مطابق جدول، کانتر C233 دو ورودی X4 و X5 را برای خود در نظر می‌گیرد که به آنها کانترهای دوفاز-دو ورودی می‌گویند. برای شمارش پالسهای شفت انکودرها باید از این نوع کانترها استفاده نمود . کانترهایی که یک ورودی را به خود اختصاص می‌دهند به عنوان کانترهای یک فاز-یک ورودی شناخته می شوند .

مثال ۲۹:

می‌خواهیم با استفاده از یک سنسور دیجیتال تعداد محصول تولید شده را بشماریم و زمانیکه تعداد این محصولات به 5 رسید دستگاه خاموش شود.



مثال ۳۰:

می‌خواهیم با استفاده از یک کانتر ۱۶ بیتی، تعداد محصولات تولید شده را که توسط یک نوار نقاله حمل و توسط یک سنسور دیجیتال شمارش می‌شوند را کنترل کنیم و هنگامی که تعداد ۱۰ محصول شمارش شد، سیستم برای هشدار به اپراتور چراغ الارمی را به مدت ۵ ثانیه روشن کند و نوار نقاله نیز متوقف شود.

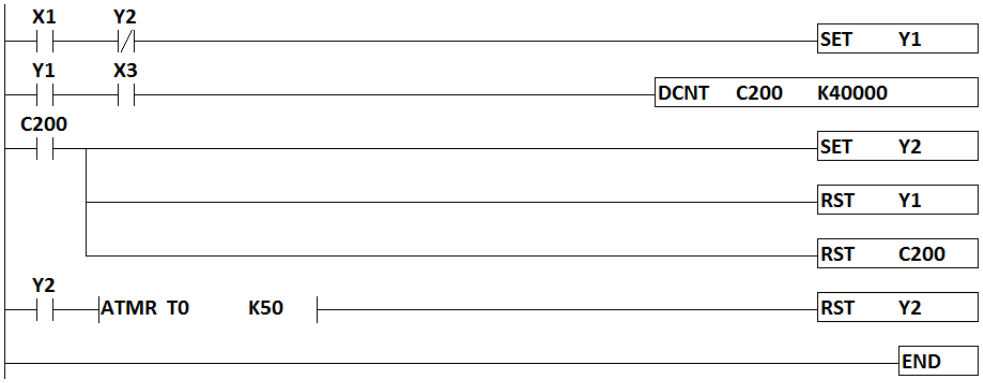
دستور DCNT :

در مواردی که تعداد محصولات و یا تعداد دفعات ارسال پالس به کانتر از محدوده‌ی یک عدد ۱۶ بیتی یعنی 32767 بیشتر شود، باید از یک کانتر ۳۲ بیتی برای ثبت و شمارش استفاده شود.

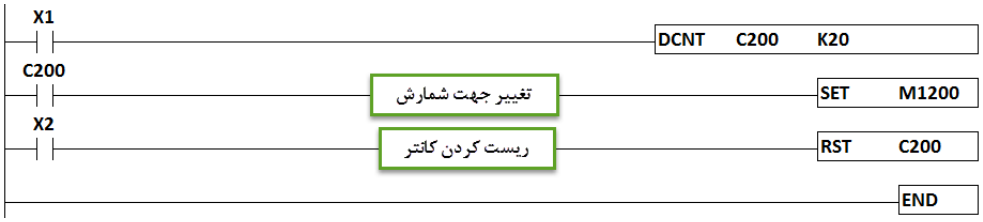
بدین منظور باید از دستور کانتر ۳۲ بیتی (DCNT) استفاده شود کانترهای 32 بیتی از C200 شروع می‌شوند. برای آگاهی از شماره کانتر ها در PLC های مختلف به فایل راهنمای آن PLC مراجعه کنید .

مثال ۳۱ :

در مثال قبل فرض کنید بجای ۱۰ محصول می‌خواهیم ۴۰۰۰۰ محصول تولید کنیم. این عدد یک عدد ۳۲ بیتی می‌باشد که برای شمارش آن نیاز به یک کانتر ۳۲ بیتی می‌باشد.



در نظر داشته باشید که کانترهای ۳۲ بیتی قابلیت پایین شمار نیز دارند. برای مثال در صورتیکه فلگ M1200 فعال (ON) شود ، کانتر C200 به صورت معکوس(پایین شمار) شمارش را انجام می‌دهد.



در این برنامه در صورتیکه مقدار شمارش به عدد ۲۰ برسد، M1200 فعال شده و باعث می‌شود کانتر به صورت معکوس شمارش را انجام دهد. برای صفر کردن (Reset) کانتر از ورودی X2 استفاده شده است.

کانترهای سرعت بالا (HSC) :

با استفاده از این کانترها می‌توان پالسهای ورودی با سرعت بالا که توسط یک سنسور و یا انکودر به PLC ارسال می‌شود را شمارش کرد. شماره این کانترها در PLC های مختلف متفاوت بوده و برای آگاهی از شماره این کانتر ها باید به فایل راهنمای هر PLC مراجعه کنید.

در نظر داشته باشید که هر PLC از نظر قدرت پردازش در دریافت پالس‌های ورودی با سایر مدلها متفاوت است و همچنین ورودی‌های پالس سرعت بالا در هر PLC مشخص است. به عبارت دیگر نمی‌توان از هر ورودی دلخواه برای شمارش پالسهای ورودی توسط کانترهای سرعت بالا استفاده کرد. تمامی این اطلاعات در فایلهای راهنمای هر PLC موجود است.

Applicable Hardware High Speed Counters:

C / X	1-phase input		1-phase 2-input						2-phase 2-input			
	C243	C244	C245	C246	C247	C248	C249#2	C250#2	C251	C252	C253	C254
X0	U		U/D	U/D	U	U			A	A		
X1	R		Dir	Dir	D	D			B	B		
X2		U					U/D	U/D			A	A
X3		R					Dir	Dir			B	B
X4				R		R				R		
X5								R				R

U: Count up D: Count down A: Phase A input B: Phase B input Dir: Directoin signal input R: Reset signal input

Note:

1. The max frequency of the 1-phase input counters X0 (C243) and X2 (C244) is 100kHz on ES2/EX2/SA2/SX2 model and 20kHz on SS2 model.
2. An SE model does not support the counters C249 and C250.
3. The max frequency of the 1-phase 2-input counters (X0, X1) (C245, C246) and (X2, X3) (C249, C250) is 100kHz on ES2/EX2/SA2/SX2 model and 20kHz on SS2 model.
4. The max frequency of the 1-phase 2-input counters (X0, X1) (C247, C248) is 10kHz on ES2/EX2/SS2/SX2 model and 100kHz on 32ES211T and SA2 model.
5. The max frequency of the 2-phase 2-input counter (X0, X1) (C251, C252) is 5kHz on ES2/EX2 model, 10kHz on SS2/SX2 model and 50kHz on 32ES211T and SA2 model.
6. The max frequency of the 2-phase 2-input counter (X2, X3) (C253, C254) is 5kHz on ES2/EX2/SA2 model, 10 kHz on SS2/SX2 model and 50kHz on 32ES211T.
7. 2-phase 2-input counting supports double and 4 times frequency, which is selected in D1022 as the table in next page. Please refer to the below table for detailed counting wave form.

برای مثال، طبق جدول فوق در حالت یک فاز و یک ورودی ، در PLC های مدل EX3 و ES2 و SX2 و SA2 ، در ورودی های X0 و X2 با شماره کانتر C243 و C244 حداکثر فرکانس ورودی 100KHz و در PLC مدل SS2 حداکثر فرکانس ورودی 20KHz را خواهیم داشت.

نحوه اتصال انکودر به PLC مدل DVP14SS211T :

برای اتصال انکودر به این نوع PLC ابتدا باید ببینیم که از کدام کانتر سرعت بالا می‌خواهیم استفاده کنیم برای این کار می‌توانیم از فایل راهنمای مخصوص این PLC بهره ببریم.

Applicable Software High Speed Counters:

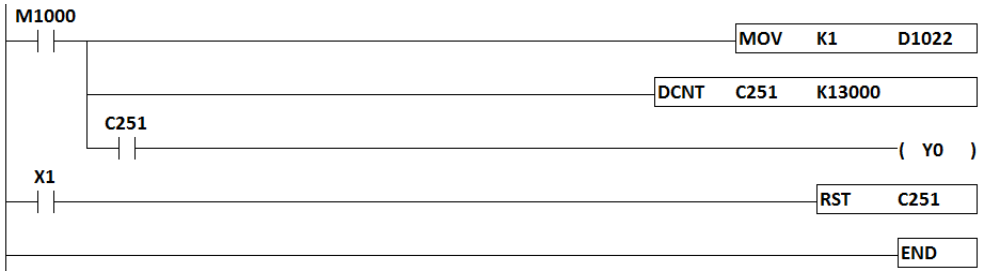
C / X	1-phase input								2 phase 2 input		
	C235	C236	C237	C238	C239	C240	C241	C242	C232#2	C233	C234
X0	U/D								A		
X1		U/D									
X2			U/D						B		
X3				U/D							
X4					U/D					A	
X5						U/D				B	
X6							U/D				A
X7								U/D			B
R/F	M1270	M1271	M1272	M1273	M1274	M1275	M1276	M1277	-	-	-
U/D	M1235	M1236	M1237	M1238	M1239	M1240	M1241	M1242	-	-	-

طبق جدول صفحه قبل در صورتیکه بخواهیم تنها از یک فاز انکودر استفاده کنیم کافیست طبق جدول، کانتر مورد نظر را در برنامه استفاده کنیم. مثلا در صورتیکه بخواهیم سیم مشکی انکودر که فاز A می‌باشد را به ورودی X0 این PLC وصل کنیم کافیست از کانتر C235 استفاده کنیم یا برای مثال در صورتیکه خواستیم از ورودی X5 استفاده کنیم کافیست که سیم مشکی که فاز A در انکودر می‌باشد را به این ورودی وصل کرده و از کانتر شماره C240 استفاده کنیم.

نکته : در صورتیکه مقدار درون دیتا رجیستر خاص D1022 برابر عدد 1 باشد به ازای هر پالسی که از طریق انکودر به شمارنده‌ی سرعت بالا میرسد به ازای هر پالس ،یک واحد محاسبه می‌شود و در صورتیکه مقدار درون این رجیستر برابر 2 باشد، به ازای هر پالس ورودی ، کانتر تعداد پالس را دو برابر شمارش می‌کند و در صورتیکه مقدار درون این رجیستر برابر 4 باشد، به ازای هر پالس ورودی ، کانتر تعداد پالس را ۴ برابر شمارش خواهد کرد.

مثال ۳۲ :

می‌خواهیم یک انکودر را به PLC مدل dvp14ss2 متصل کنیم و از طریق یک کانتر سرعت بالا به عنوان مثال C251 ، پالسهای ورودی را شمارش کنیم. بدین منظور ابتدا سیم های فاز A و فاز B انکودر را به پایه های X0 و X1 در PLC متصل می‌کنیم. سپس سیم های تغذیه‌ی انکودر (سیم قهوه‌ای به پایه ی +24V و سیم آبی به پایه ی 0v منبع تغذیه) وصل کرده و به ترمینال S/S در PLC نیز +24V را وصل می‌کنیم. سپس بطور مثال برنامه‌ای مطابق زیر می‌نویسیم .



۱ – در صورتیکه عدد K1 به دیتا رجیستر D0 منتقل شود ، مقدار کانتر به ازای هر پالس، یک شماره افزایش می یابد.

در صورتیکه عدد K2 به دیتارجیستر D0 منتقل شود ، مقدار کانتر به ازای هر پالس، دو شماره افزایش می یابد.

در صوتی که عدد K4 به دیتارجیستر D0 منتقل شود ، مقدار کانتر به ازای هر پالس، چهار شماره افزایش می یابد.

۲ – استفاده از دستور کانتر ۳۲ بیتی با شماره کانتر C251 طبق جدول کانترهای سرعت بالا.(صفحه ۴۴)

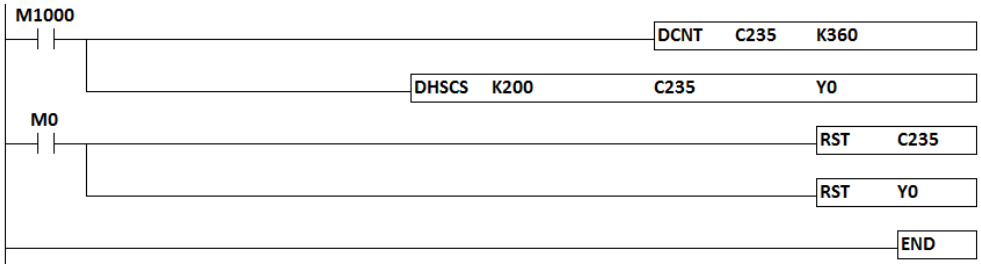
این کانتر برای شمارش پالسهای دوفاز دو ورودی از طریق پایه‌های X0 و X1 و فرکانس 10kHz می‌باشد.

۳ – پس از رسیدن شماره کانتر به عدد 13000 ، تیغه کانتر فعال شده و خروجی Y0 فعال می‌شود.

دستور DHSCS :

در برنامه‌های مربوط به شمارش پالس های با سرعت زیاد گاهی نیاز می شود با رسیدن به تعداد مشخصی از پالس، فورا یک خروجی فعال شود در این صورت روشن شدن خروجی مورد نظر، منتظر پایان سیکل جاری نمی شود و فورا وضعیت خروجی را در همان لحظه به ترمینالهای PLC ارسال می کند . مثال زیر عملکرد این دستور را نشان می دهد.

مثال ۳۳ :



در سطر اول از کانتر شماره C235 استفاده کرده‌ایم . این کانتر ورودی X0 را به عنوان ورودی پالس خود در نظر می‌گیرد. در سطر دوم برنامه، چنانچه تعداد پالسهای شمارش شده برابر K200 گردد، ترمینال خروجی Y0 فورا روشن می شود و منتظر پایان سیکل جاری برنامه نمی شود .در سطرهای بعدی از فلگ M0 برای ریست کردن کانتر و خروجی استفاده کرده‌ایم .

نکته : از فلگ M1235 برای پایین یا بالا شمار کردن کانتر C235 استفاده کنیم در صورتیکه فلگ M1235 خاموش باشد، کانتر C235 بالا شمار بوده و در صورت فعال بودن فلگ M1235 ، کانتر C235 پایین شمار خواهد بود . هر کانتری، یک فلگ اختصاصی برای پایین و یا بالا شمار بودن دارد .

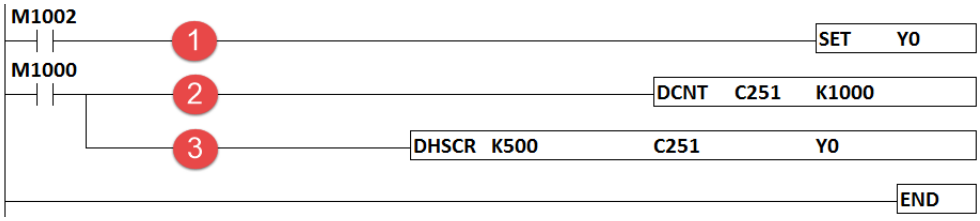
نکته : در این دستور فقط خروجیهای Y3 ... Y0 قابلیت روشن شدن آنی را دارند . چنانچه از خروجیهای دیگر مانند Y5، در این دستور استفاده کنیم بعد از پایان سیکل جاری ترمینالهای خروجی مورد نظر فعال می شود.

نکته : . در این دستور می توان از M و S بجای Y استفاده کرد . همچنین می توان مستقیما یک وقفه را فعال کرد در این صورت برنامه اصلی موقتا متوقف شده و زیربرنامه مربوط به وقفه فورا اجرا شده و سپس به برنامه اصلی باز می گردد . شماره وقفه های مجاز در این دستور از I080 ... I010 می باشد در بخش وقفه ها در این خصوص توضیح داده شده است .

دستور DHSCR :

در این دستور با رسیدن تعداد پالسهای شمارش شده به عدد معینی، یک خروجی و یا فلگ فورا ریست می شود و منتظر سیکل جاری برنامه نمی ماند . مثال زیر را ببینید.

مثال ۳۴ :



۱ – ست شدن خروجی Y0 توسط فلگ خاص M1002 در اولین سیکل برنامه.

۲ – استفاده از کانتر شماره C251 جهت شمارش تعداد K1000 پالس.

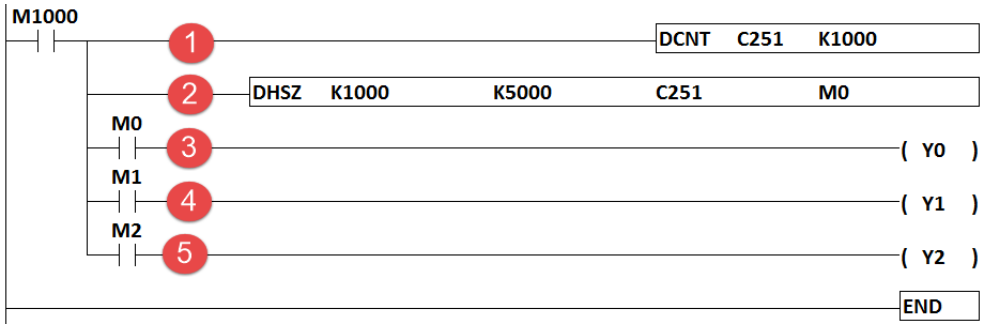
۳ – استفاده از دستور کانتر سرعت بالا جهت ریست کردن خروجی Y0 پس از رسیدن تعداد پالس به K500 .

نکته : در دستور فوق زمانیکه مقدار کانتر C251 از 501 به 500 تغییر کند خروجی Y0 فورا خاموش می شود .

دستور DHSZ :

از این دستور به منظور مقایسه تعداد پالس شمارش شده به صورت ناحیه‌ای (Zone) استفاده می‌شود. برای اطلاع از عملکرد این دستور به این مثال توجه کنید .

مثال ۳۵ :



۱ – شمارش پالسهای ورودی توسط شمارنده سرعت بالای C251

۲ – دستور مقایسه کننده پالس ناحیه‌ای

K1000 : حد پایین تعداد پالس K5000 : حد بالای تعداد پالس C251 : شماره کانتر سرعت بالا

M0 : اولین خروجی یا فلگ که برای عملیات مقایسه استفاده می‌شود.

۳ – در صورتیکه تعداد پالس شمارش شده توسط کانتر سرعت بالا از K1000 کمتر باشد این خروجی فورا روشن می‌شود.

۴ – در صورتیکه تعداد پالس شمارش شده توسط کانتر سرعت بالا بین K1000 و K5000 باشد این خروجی فورا روشن می‌شود.

۵ – در صورتیکه تعداد پالس شمارش شده توسط کانتر سرعت بالا از K5000 بیشتر باشد این خروجی فورا روشن می‌شود.

دستورات ریاضی :

دستور جمع ۱۶ بیتی : ADD

دستور جمع ۳۲ بیتی : DADD

دستور تفریق ۱۶ بیتی : SUB

دستور تفریق ۳۲ بیتی : DSUB

دستور ضرب ۱۶ بیتی : MUL

دستور ضرب ۳۲ بیتی : DMUL

دستور تقسیم ۱۶ بیتی : DIV

دستور تقسیم ۳۲ بیتی : DDIV

دستور جمع اعداد اعشاری : DEADD و DADDR

دستور تفریق اعداد اعشاری : DESUB و DSUBR

دستور ضرب اعداد اعشاری : DEMUL و DMULR

دستور تقسیم اعداد اعشاری : DEDIV و DDIVR

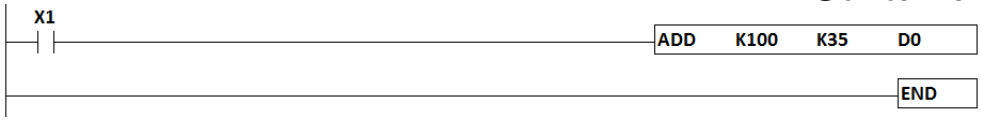
نکته : محدوده اعداد ۱۶ بیتی از 32768+ الی 32767- می‌باشد. اعداد بزرگتر از این محدوده به عنوان 32 بیتی در نظر گرفته می‌شوند .

دستور جمع اعداد صحیح ADD :

با استفاده از این دستور می‌توانید دو مقدار ثابت و یا محتویات درون دو رجیستر را با هم جمع کرده و حاصل را درون یک رجیستر دیگر قرار دهید.

مثال ۳۶ :

می‌خواهیم مقدار دو عدد ثابت K100 و K35 را با یکدیگر جمع کنیم.



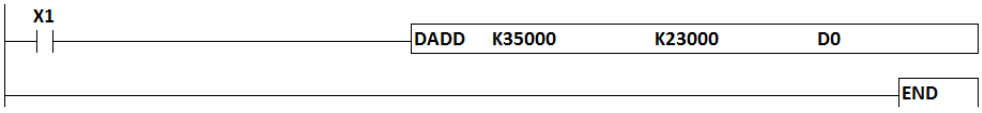
حال با فعال کردن X1 ، مقدار ثابت K100 با K35 جمع شده و درون رجیستر D0 ریخته می‌شود.

نکته : برای جمع اعداد ۳۲ بیتی بجای دستور ADD باید از دستور جمع اعداد ۳۲ بیتی یعنی DADD استفاده کرد که در این حالت ۲ رجیستر متوالی اشغال می‌شوند.

نکته : باید دقت کرد در دستورات 16 بیتی، مانند جمع و ضرب، حاصل از مقدار مجاز 32767+ بیشتر و یا از مقدار 32768- کمتر نشود .

مثال ۳۷:

می‌خواهیم دو عدد ۳۲ بیتی را با یکدیگر جمع کنیم.



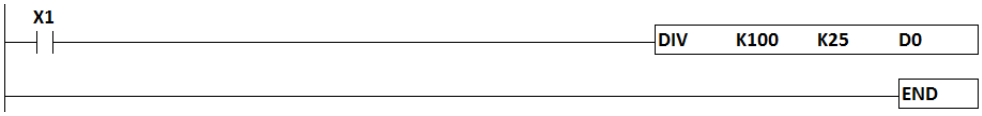
نکته : در این برنامه، دو رجیستر D0 و D1 مورد استفاده قرار می‌گیرند .

دستور تقسیم اعداد صحیح DIV :

با استفاده از این دستور می‌توانید دو مقدار ثابت و یا محتویات درون دو رجیستر را بر یکدیگر تقسیم کرده و حاصل را درون یک رجیستر دیگر قرار دهید.

مثال ۳۸ :

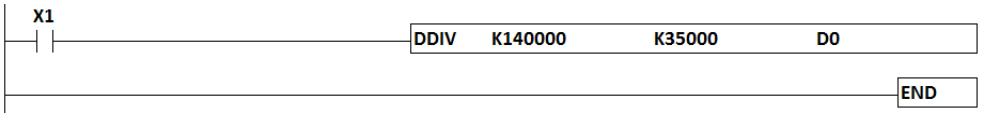
می‌خواهیم دو مقدار ثابت K100 و K25 را بر یکدیگر تقسیم کنیم.



مطابق برنامه با فعال شدن X1 ، عدد ثابت K100 بر عدد ثابت K25 تقسیم شده و حاصل درون دیتا رجیستر D0 ریخته شده است.

مثال ۳۹ :

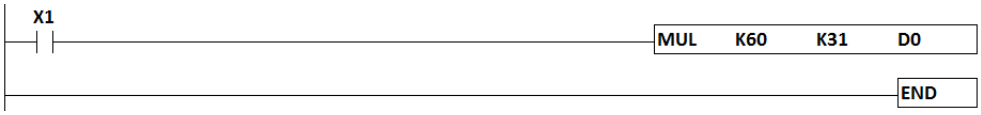
می‌خواهیم دو عدد ۳۲ بیتی K140000 و K35000 را بر یکدیگر تقسیم کنیم.



دستور ضرب اعداد صحیح MUL :

مثال ۴۰:

می‌خواهیم دو مقدار ثابت K60 و K31 را در یکدیگر ضرب کنیم. برنامه صفحه بعد را ببینید .

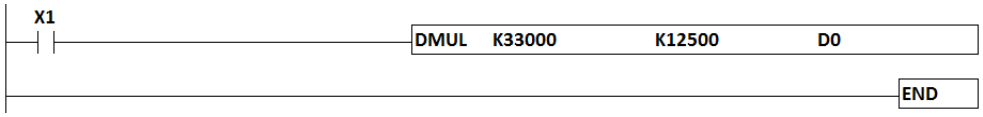


همانطور که در شکل بالا ملاحظه می‌کنید پس از فعال شدن X1 ، مقدار ثابت K60 در مقدار ثابت K31 ضرب شده و حاصل درون دیتا رجیستر D0 ریخته می‌شود. در دستور فوق می‌توان بجای استفاده از اعداد ثابت، از رجیسترها استفاده کرد .

نکته : دقت شود حاصل ضرب اعداد، از محدوده مجاز اعداد 16 بیتی بیشتر نشود .

مثال ۴۱ :

می‌خواهیم دو مقدار ۳۲ بیتی K33000 و K12500 را در یکدیگر ضرب کنیم.

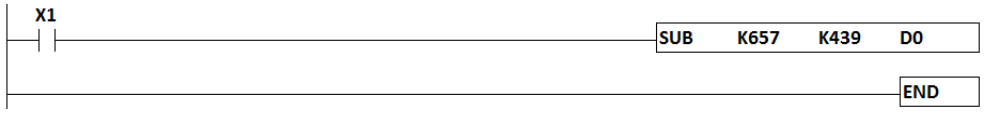


نکته : در این برنامه، دو رجیستر D0 و D1 مورد استفاده قرار می‌گیرند .

دستور تفریق SUB :

مثال ۴۲:

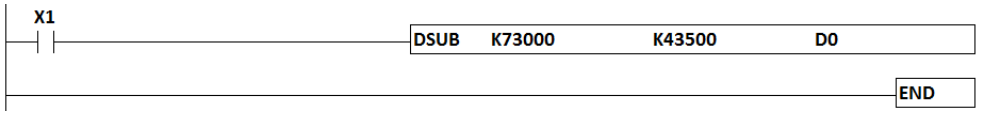
می‌خواهیم دو مقدار ثابت K657 و K439 را از یکدیگر کسر کنیم.



مطابق شکل بالا با فعال شدن X1 اعداد ثابت K657 و K439 بر یکدیگر تقسیم شده و حاصل درون دیتارجیستر D0 ریخته می‌شود.

مثال ۴۳ :

می‌خواهیم دو عدد ۳۲ بیتی K73000 و K43500 را از یکدیگر کسر کنیم.



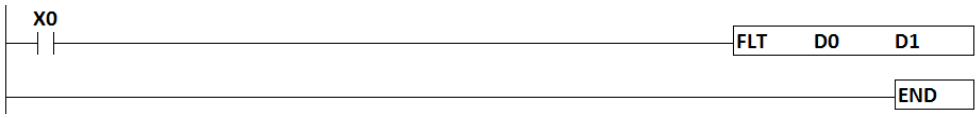
نکته : در این برنامه، دو رجیستر D0 و D1 مورد استفاده قرار می‌گیرند .

تبدیل فرمت عدد صحیح (INT) به اعشاری (FLOAT) :

دستور FLT & DFLT :

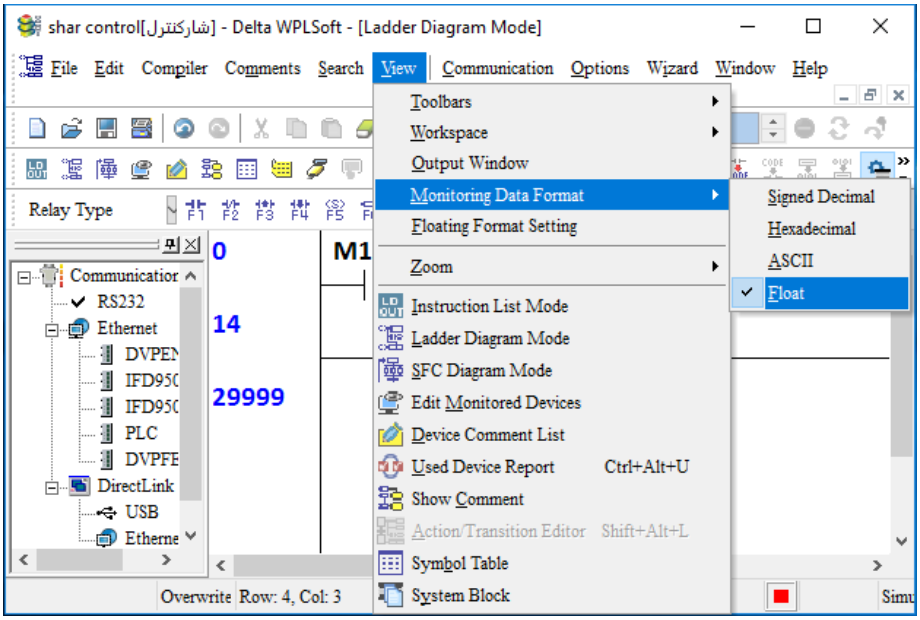
با استفاده از دستور FLT می‌توان فرمت عدد صحیح ۱۶ بیتی را به فرمت ۳۲ بیتی اعشاری تغییر داد و با استفاده از دستور DFLT می‌توان فرمت عدد صحیح ۳۲ بیتی را به فرمت ۳۲ بیتی اعشاری تغییر داد.

مثال ۴۴ :



نکته : در این برنامه عدد صحیح 16 بیتی درون رجیستر D0 به فرمت اعشاری32 بیتی تبدیل شده و درون رجیسترهای D1 و D2 منتقل می‌شود .

نکته : بعد از تبدیل فرمت عدد صحیح به عدد اعشاری باید نحوه نمایش عدد در برنامه را نیز به حالت اعشاری تغییر دهیم. برای این کار کافیست تا به ترتیب زیر عمل کنیم .

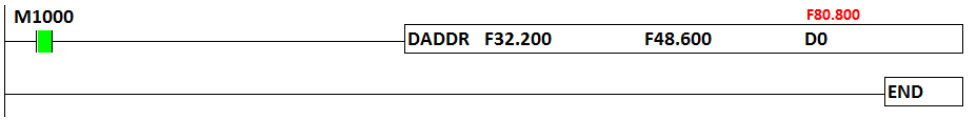


پس از انتخاب گزینه Float ، نحوه نمایش اعداد نیز به صورت اعشاری خواهد شد.

دستور جمع اعداد اعشاری DADDR :

مثال ۴۵ :

می‌خواهیم دو مقدار اعشاری 32.2 و 48.6 را با یکدیگر جمع کرده و حاصل را در یک رجیستر مانند D0 مشاهده کنیم.



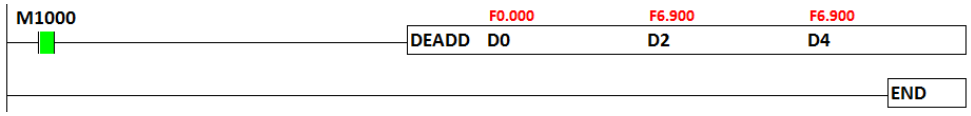
نکته : رجیسترهای D0 و D1 به دلیل ۳۲ بیتی بودن دستور در این مثال مورد استفاده قرار می‌گیرد.

نکته : برای وارد کردن اعداد ثابت اعشاری باید از پیشوند F استفاده کرد .

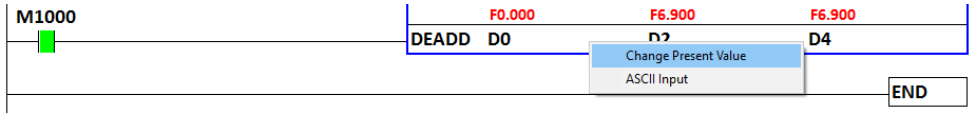
دستور جمع اعداد اعشاری DEADD :

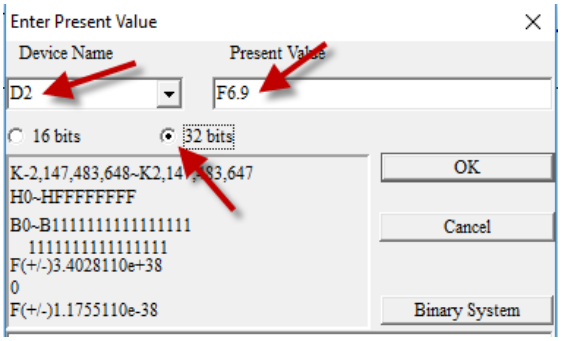
مثال ۴۶ :

می‌خواهیم دو مقدار اعشاری که را که درون رجیسترهای D0 و D2 هستند را با یکدیگر جمع کرده و حاصل را درون رجیستر D4 مشاهده کنیم.



نکته : برای مقدار دادن به رجیسترهای برنامه ، بعد از دانلود برنامه و در حالت ONLINE دستور فوق را انتخاب و سپس روی آن راست کلیک کرده گزینه Change present value را انتخاب می‌کنیم در پنجره باز شده از کادر سمت چپ رجیستر مورد نظر را انتخاب کرده و در کادر سمت راست، عدد مورد نظر را با پیشوند K برای اعداد صحیح و پیشوند F برای اعداد اعشاری و پیشوند H برای اعداد هگز وارد می کنیم. ضمنا گزینه ۱۶ یا ۳۲ بیتی بودن را نیز باید انتخاب کنیم. مانند شکل زیر:

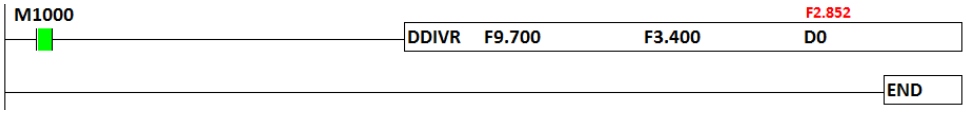




دستور تقسیم اعداد اعشاری DDIVR :

مثال ۴۷ :

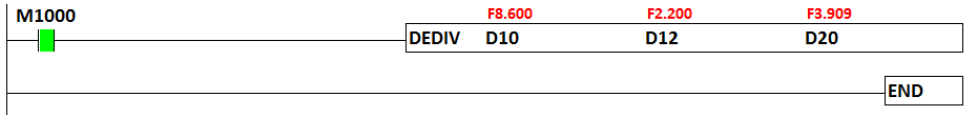
می‌خواهیم دو مقدار اعشاری 3.4 و 9.7 را بر یکدیگر تقسیم کرده و حاصل را درون رجیسترهای D0 و D1 ثبت کنیم.



دستور تقسیم اعداد اعشاری DEDIV :

مثال ۴۸ :

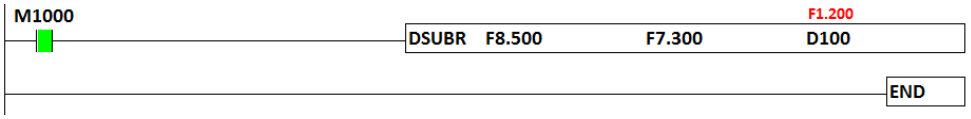
می‌خواهیم دو مقدار اعشاری که درون رجیسترهای D10 و D12 قرار دارند را بر یکدیگر تقسیم کرده و حاصل را درون رجیسترهای D20 و D21 مشاهده کنیم.



دستور تفریق اعداد اعشاری DSUBR :

مثال ۴۹ :

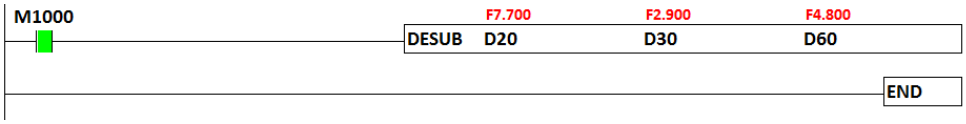
می‌خواهیم دو مقدار اعشاری 8.5 و 7.3 را از یکدیگر کم کرده و جواب را در رجیسترهای D100 و D101 مشاهده کنیم.



دستور تفریق اعداد اعشاری DESUB :

مثال ۵۰ :

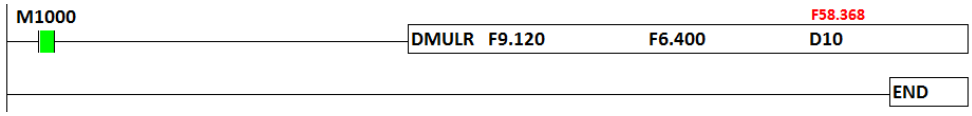
می‌خواهیم مقادیر اعشاری درون دو رجیستر D20 و D30 را از یکدیگر تفریق کرده و جواب را درون رجیسترهای D60 و D61 مشاهده کنیم.



دستور ضرب اعداد اعشاری DMULR :

مثال ۵۱ :

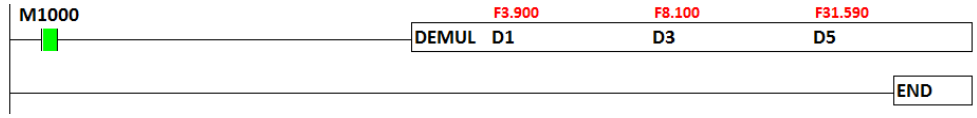
می‌خواهیم دو مقدار اعشاری 6.4 و 9.12 را در یکدیگر ضرب کرده و حاصل را در رجیسترهای D10 و D11 مشاهده کنیم.



دستور ضرب اعداد اعشاری DEMUL :

مثال ۵۲ :

می‌خواهیم مقادیر اعشاری درون دو رجیستر D1 و D3 را با یکدگیر ضرب کرده و حاصل را درون رجیسترهای D5 و D6 مشاهده کنیم.

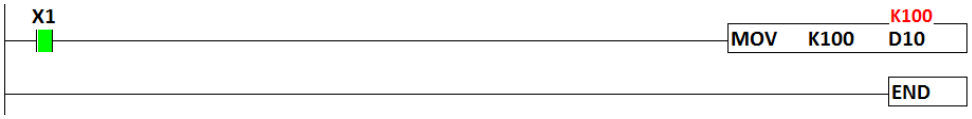


دستور انتقال MOV :

با استفاده از این دستور می‌توان مقدار درون یک رجیستر یا یک عدد ثابت را به یک رجیستر دیگر انتقال داد.

مثال ۵۳ :

می‌خواهیم مقدار ثابت K100 را به رجیستر D10 انتقال دهیم.

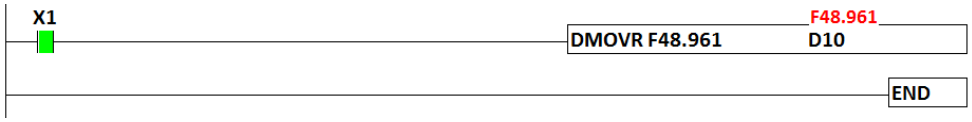


نکته : در نظر داشته باشید که برای انتقال اعداد ۳۲ بیتی باید از دستور انتقال اعداد ۳۲ بیتی (DMOV) و برای انتقال اعداد اعشاری باید از دستور انتقال اعداد اعشاری(DMOVR) استفاده کنید.

دستور انتقال اعداد اعشاری DMOVR :

مثال ۵۴ :

می‌خواهیم عدد اعشاری 48.961 را به رجیسترهای D10 و D11 منتقل کنیم.



نکته : در نظر داشته باشید که برای اینکه عدد درون D0 به صورت اعشاری نمایش داده شود باید همانگونه که در قسمت تبدیل اعداد اعشاری گفته شد، نحوه نمایش اعداد را از حالت Signe Decimal به حالت Float تغییر دهیم. این تغییر را می‌توان از مسیر زیر انجام داد:

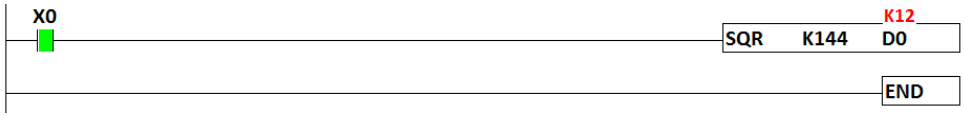
View >Monitoring Data Format >Float

دستور جذر SQR :

با این دستور می‌توان از یک مقدار ثابت یا مقدار درون یک رجیستر جذر گرفت.

مثال ۵۵ :

می‌خواهیم جذر عدد ثابت K144 را محاسبه کنیم.



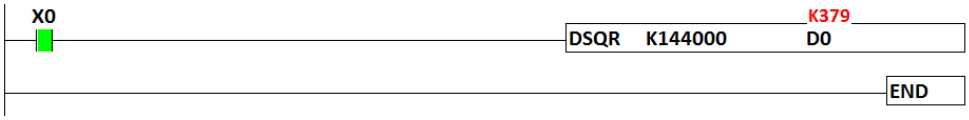
همانطور که دیده می‌شود با فعال شدن X0، جذر عدد ثابت 144 محاسبه شده و درون رجیستر D0 ریخته می‌شود.

نکته : برای گرفتن جذر یک عدد ۳۲ بیتی باید از دستور جذر اعداد ۳۲ بیتی یعنی DSQR استفاده کنید.

دستور جذر اعداد صحیح ۳۲ بیتی DSQR :

مثال ۵۶:

می‌خواهیم جذر عدد ثابت K144000 را محاسبه کنیم.



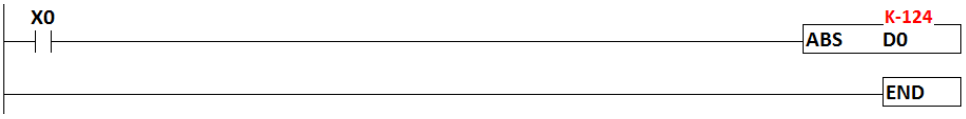
برای گرفتن جذر یک عدد اعشاری می‌توانید از دستور DESQR استفاده کنید.

دستور قدر مطلق ABS :

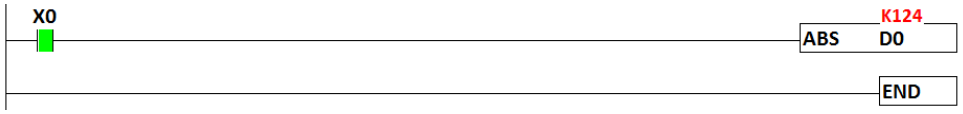
با این دستور می‌توان از مقدار درون یک رجیستر قدرمطلق گرفت.

مثال ۵۷ :

می‌خواهیم مقدار قدر مطلق عدد ثابت K-124 را حساب کنیم.



همانطور که ملاحظه می‌کنید مقدار عدد درون رجیستر D0 برابر با K-124 است. حال با فعال کردن ورودی X0 ، قدر مطلق محتوای این رجیستر محاسبه شده و جایگزین عدد اصلی می‌شود.



در صورتیکه بخواهیم از یک عدد ۳۲ بیتی قدر مطلق بگیریم می‌توانیم از دستور DABS استفاده کنیم.

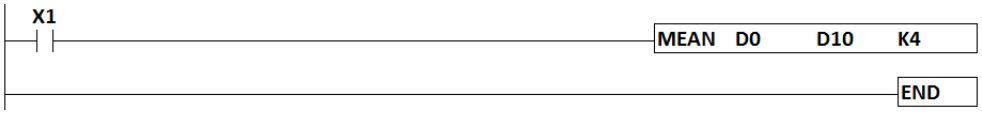
دستور MEAN :

با استفاده از این دستور می‌توانید مقدار میانگین چند رجیستر را محاسبه کرده و درون یک رجیستر دیگر قرار دهید .

مثال ۵۸ :

می‌خواهیم میانگین محتویات درون چهار رجیستر D0 – D1 – D2 – D3 را محاسبه کنیم.

در نظر داشته باشید که مقادیر این رجیسترها در این مثال به ترتیب برابر 10 – 20 – 30 – 40 می‌باشد.



حال با فعال کردن ورودی X1 ، از مقادیر 10 و20 و30 و 40 میانگین گرفته می‌شود و جواب درون رجیستر D10 ریخته می‌شود. (D0 + D1 + D2 + D3) ÷ 4 →→→ D10

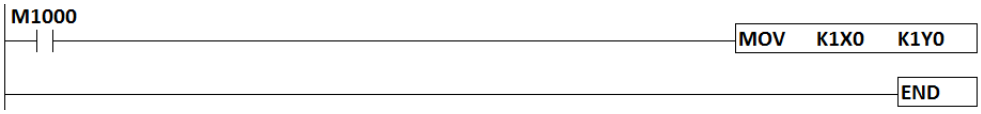
استفاده از ترمینال های ورودی X به صورت گروهی در فضای حافظه ...KnXn :

با استفاده از این فضا می‌توان ورودیها و خروجی‌ها را به صورت گروهی کنترل کرد.

حافظه ۳۲ بیتی برای kn		حافظه ۱۶ بیتی برای kn	
K1 (4 Bit)	0~15	K1 (4 Bit)	0~15
K2 (8 Bit)	0~255	K2 (8 Bit)	0~255
K3 (12 Bit)	0~4095	K3 (12 Bit)	0~4095
K4 (16 Bit)	0~65535	K4 (16 Bit)	−32468~32767
K5 (20 Bit)	0~1048575		
K6 (24 Bit)	0~167772165		
K7 (28 Bit)	0~268435455		
K8 (32 Bit)	−2147483648~2147483647		

برای برای درک بهتر به مثال ساده‌ی زیر توجه کنید

مثال ۵۹ :



در این مثال با تحریک X0 ، خروجی Y0 فعال می‌شود. با تحریک X1 ، خروجی Y1 فعال می‌شود. به همین ترتیب با تحریک ورودی های X2 و X3 به ترتیب خروجی های Y2 و Y3 فعال خواهند شد.

دستور فراخوانی CALL :

در نوشتن برنامه‌ها می‌توان از زیربرنامه‌ها استفاده کرد تا در مواقع لزوم بتوان برنامه‌های جانبی را فراخوانی کرد. استفاده از این دستور همچنین به تسهیل در توسعه برنامه و رفع ایرادات برنامه کمک می‌کند.

مثال ۶۰ :

همانطور که در این مثال ملاحظه می‌کنید، قسمت P1 با تحریک ورودی X1 فراخوانی شده لذا می‌توان توسط حافظه ی M1 ، قسمت P1 برنامه، که مربوط به خروجی Y0 می‌باشد را فعال کرد اما قسمت P2 فراخوانی نشده و لذا با تغییر حافظه ی M2 خروجی Y1 تغییری نمی‌کند.

دستور **FEND** به معنای انتهای قسمت برنامه اصلی می‌باشد چون بعد از این دستور، زیربرنامه ها ایجاد شده اند. دستور **SRET** به معنای پایان هر زیر برنامه(Sub Routin) می‌باشد. در این مثال دو زیر برنامه وجود دارد لذا در پایان هر زیر برنامه یک دستور SRET نوشته شده است .

P1 و **P2** برچسب (Lable) مربوط به زیربرنامه ها می‌باشد و دستور CALL در صورت اجرا شدن ،زیر برنامه هم نام با لیبل خود را فراخوانی می کند.

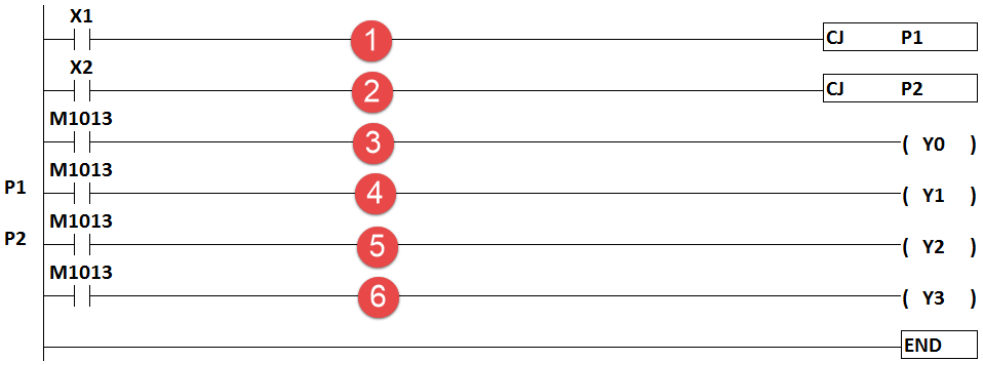
نکته: در قسمت برنامه اصلی می‌توان برنامه نویسی کرد و در هر سطر از برنامه اصلی که نیاز باشد با استفاده از دستور CALL زیر برنامه ها را فراخوانی کرد .

نکته : فقط تایمرهای T192 ... T199 می توانند در زیربرنامه ها مورد استفاده قرار گیرند .

دستور CJ :

دستور CJ یا پرش در مواقعی استفاده می‌شود که بخواهیم پردازش، از سطر خاصی از برنامه به سطر دیگری منتقل شود. برای درک بهتر این دستور، به مثال زیر توجه کنید.

مثال ۶۱ :



در برنامه‌ی نوشته شده، تا زمانی که تیغه های X1 و X2 فعال نشده باشند، خروجی های Y0 الی Y3 طبق روال عادی برنامه به صورت همزمان خاموش و روشن خواهند شد.

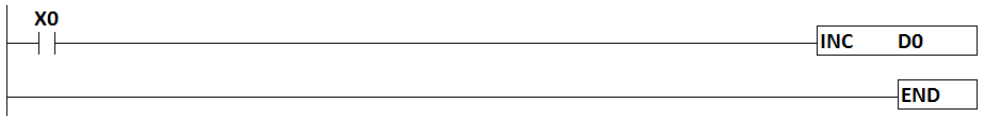
نکته : فلگ M1013 یک کلاک پالس (0.5s ON / 0.5s OFF) می‌باشد.

با فعال شدن X1، برنامه های مربوط به خطهای ۴ و ۵ و ۶ پردازش و اجرا می‌شوند و با فعال شدن تیغه‌ی X2، برنامه های مربوط به خطهای ۵ و ۶ پردازش و اجرا می‌شوند.

دستور INC :

در مواقعی نیاز داریم تا بدانیم که یک خروجی چند بار فعال شده است یا چند محصول توسط ماشین تولید شده است لذا این دستور به صورت یک شمارنده افزایشی عمل کرده و می‌تواند این اطلاعات را ثبت کند.

مثال ۶۲ :



نکته : با تحریک ورودی X0 ، دستور INC در هر سیکل اجرای برنامه یک واحد به مقدار رجیستر D0 اضافه میکند. از آنجا که سرعت اجرای هر سیکل بالا می‌باشد، با هر بار تحریک ورودی بیش از یک واحد به مقدار رجیستر D0 اضافه می‌شود . چنانچه بخواهیم با هر بار تحریک ورودی X0 ، فقط یک واحد به مقدار D0 اضافه شود دو روش وجود دارد :

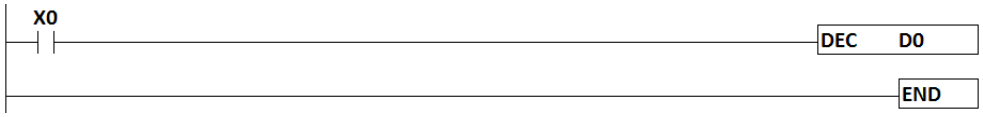
۱-از ورودیهای حساس به لبه بالارونده یا پایین رونده استفاده شود .

۲-از دستور INCP استفاده شود . این دستور بعد از هر بار اجرا، فقط یک سیکل فعال می‌شود و بعد غیر فعال می‌شود .

دستور DEC :

این دستور دقیقا عکس دستور INC عمل می‌کند و به صورت معکوس عمل شمارش را انجام میدهد.

مثال ۶۳ :



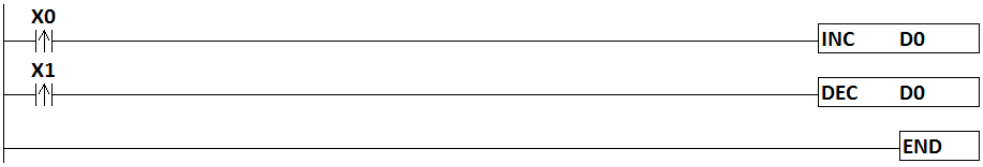
نکته : با تحریک ورودی X0 ، دستور DEC در هر سیکل اجرای برنامه یک واحد از مقدار رجیستر D0 کم می کند. از آنجا که سرعت اجرای هر سیکل بالا می‌باشد، با هر بار تحریک ورودی بیش از یک واحد از مقدار رجیستر D0 کم می‌شود . چنانچه بخواهیم با هر بار تحریک ورودی X0 ، فقط یک واحد از مقدار D0 کم شود دو روش وجود دارد :

۱-از ورودیهای حساس به لبه بالارونده یا پایین رونده استفاده شود .

۲-از دستور DECP استفاده شود . این دستور بعد از هر بار اجرا، فقط یک سیکل فعال می‌شود و بعد غیر فعال می‌شود .

نکته : در صورتیکه از دستور INC و DEC بجای INCP و DECP استفاده کنید باید از تیغه ورودی لبه دار استفاده کنید تا به ازای هر بار فعال و غیر فعال شدن تنها یکبار دستور اجرا شود.

مثال۶۴ :



به ازای هر پالسی که توسط X0 به دستور INC میرسد، این دستور یک واحد به مقدار قبلی D0 اضافه می کند و به ازای هر پالسی که توسط X2 به دستور DEC می‌رسد، این دستور یک واحد از مقدارD0 کم میکند .

دستورات مقایسه کننده :

در PLC های دلتا دستورات متعددی برای انجام عمل مقایسه وجود دارد که به بررسی برخی از آنها می‌پردازیم. اولین سری از مقایسه کننده ها ، مقایسه کننده هایی هستند که در یک سطر از برنامه بطور سری قرار می گیرد و به صورت زیر عمل میکنند:

LD= : حالت مساوی　　**LD<>** : حالت نامساوی

LD< : حالت کوچکتر　　**LD>** : حالت بزرگتر

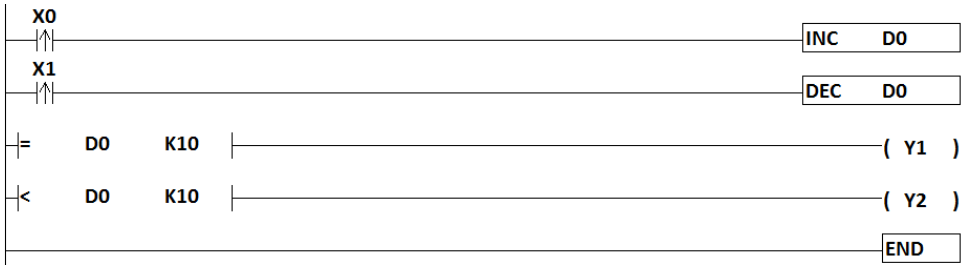
LD<= : حالت کوچکتر یا مساوی　　**LD>=** : حالت بزرگتر یا مساوی

نکته : برای مقایسه‌ی اعداد ۳۲ بیتی کافیست، یک حرف D به اول دستورات مقایسه ای بالا اضافه کنید. برای مثال برای مقایسه‌ی دو عدد ۳۲ بیتی در حالت مساوی، بجای LD= ، از DLD= استفاده می‌کنیم. برای درک بهتر این دستورات مقایسه‌ای به مثال زیر توجه کنید.

مثال ۶۵ :

پارکینگی با ظرفیت ۱۰ خودرو را در نظر بگیرید. سنسور X0 برای ماشین های ورودی به پارکینگ و سنسور X1 برای ماشینهای خروجی از پارکینگ می‌باشد .

می‌خواهیم در صورتیکه تعداد خودروها برابر ۱۰ بود خروجی Y1 ، و در صورتیکه تعداد خودرو ها کمتر از ۱۰ بود خروجی Y2 روشن شود.



در این برنامه از ورودیهای حساس به لبه استفاده شده است تا چنانچه یک خودرو از گیت ورودی وارد یا خارج شود فقط یک پالس به دستور INC یا DEC ارسال شود در نتیجه فقط یک واحد کم یا زیاد شود . در صورتیکه بخواهیم به روش نوشتاری دستورات مقایسه ای را بنویسیم به صورت زیر تایپ می کنیم : LD= d0 k10

این دستورات مقایسه ای بسیار در پروژها مورد استفاده قرار می گیرد .

دستور CMP :

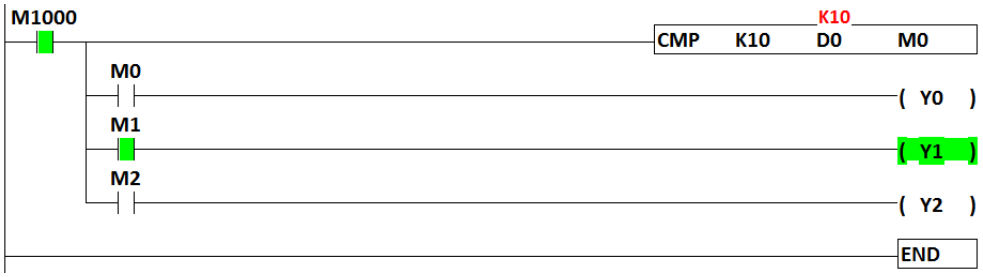
این دستور در غالب یک مثال توضیح داده خواهد شد.

مثال ۶۶ :

در صورتیکه عدد داخل رجیستر D0 کمتر از عدد ثابت تعیین شده (K10) باشد خروجی اول Y0 فعال می‌شود.

در صورتیکه عدد داخل رجیستر D0 برابر با عدد ثابت تعیین شده (K10) باشد خروجی دوم Y1 فعال می‌شود.

در صورتیکه عدد داخل رجیستر D0 بیشتر از عدد ثابت تعیین شده (K10) باشد خروجی سوم Y2 فعال می‌شود.



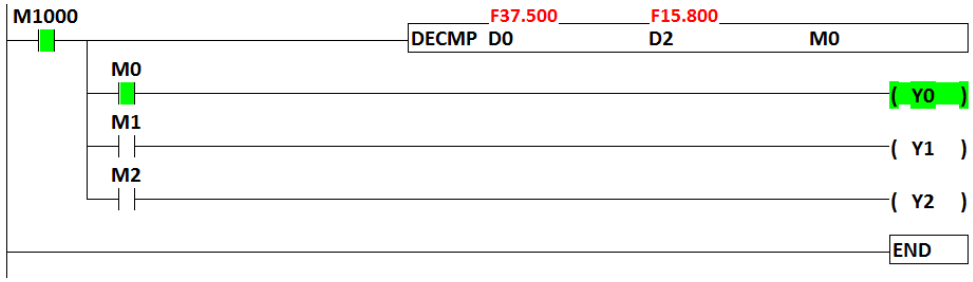
نکته: M1000 یکی از فلگ‌های خاص می‌باشد که با RUN شدن PLC روشن و با استپ شدن آن ، خاموش می‌شود. در این دستور بجای خروجی ها می‌توانید از M هم استفاده کنید البته در اینصورت سه فلگ متوالی را اشغال می کند .

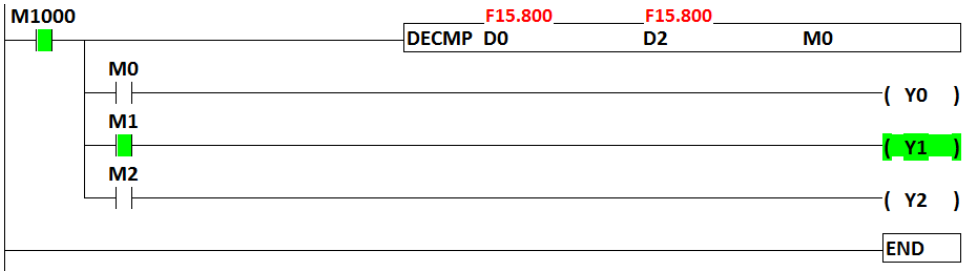
دستور مقایسه اعداد اعشاری DECMP :

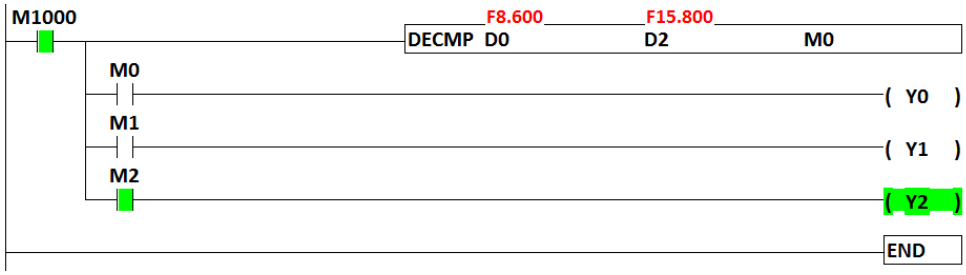
برای مقایسه اعداد اعشاری بجای دستور CMP باید از دستور DECMP استفاده کنیم.

مثال ۶۷ :

می‌خواهیم مقدار درون رجیستر D2 را با مقدار درون رجیستر D0 مقایسه کنیم.



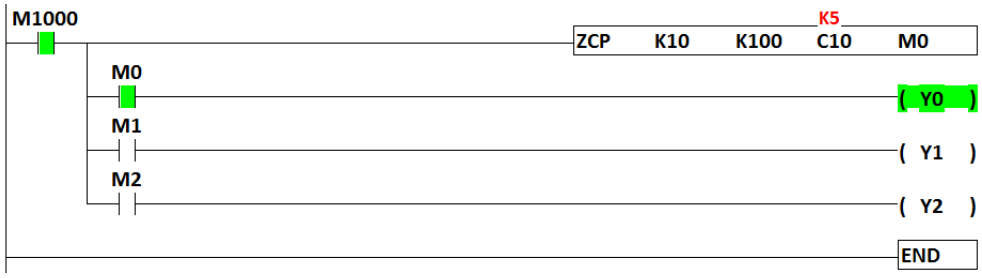


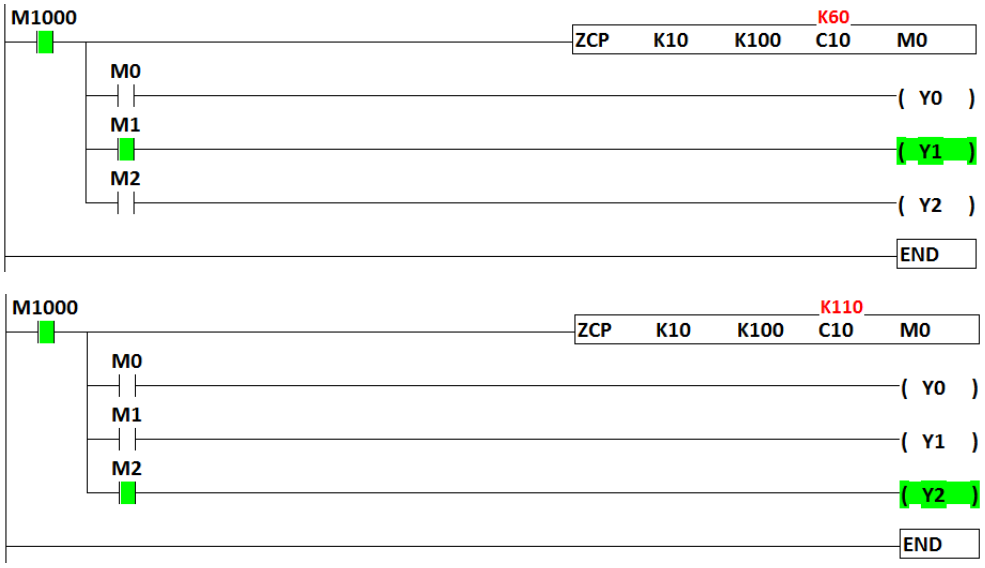


دستور مقایسه ناحیه‌ای ZCP :

در برخی برنامه‌ها نیاز می‌شود که مقایسه به صورت ناحیه‌ای (Zone) انجام شود. برای مثال در مرغداری ها دمای مطلوب بین ۲۶ تا ۳۲ درجه سانتی گراد می‌باشد لذا برای کنترل دمای این مجموعه‌ها می‌توان از این دستور استفاده کرد.

مثال۶۸ :





در این مثال یک شمارنده (C10) در حال شمارش تعداد قطعات تولید شده می‌باشد. حال با استفاده از این دستور مقایسه‌ای، می‌خواهیم در صورتیکه تعداد قطعات از عدد ثابت تعیین شده‌ی اول (K10) کمتر بود، خروجی Y0 فعال شود. در صورتیکه تعداد قطعات ما بین 10 تا 100 بود، خروجی Y1 روشن شود. در صورتیکه تعداد قطعات از عدد ثابت تعیین شده (K100) بیشتر بود، خروجی Y3 فعال شود.

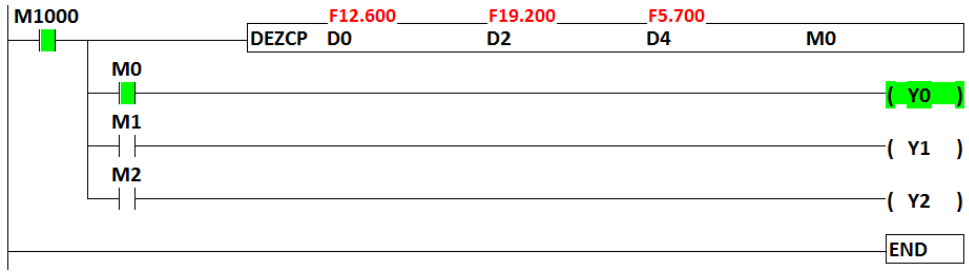
نکته : در این دستور می‌توانیم از حافظه‌ی M0 بجای Y0 استفاده کنیم.

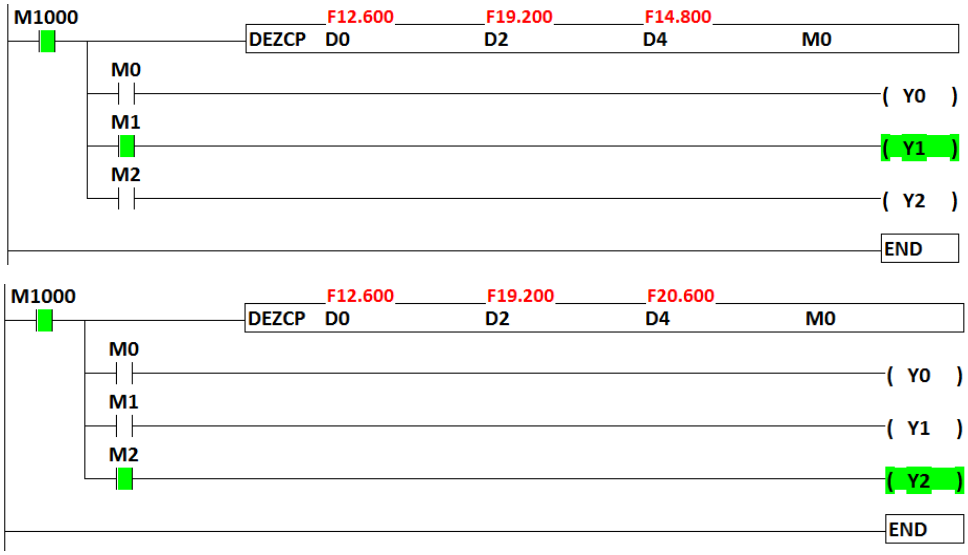
دستور DEZCP :

برای مقایسه ناحیه‌ای اعداد اعشاری بجای دستور ZCP از دستور DEZCP استفاده می‌کنیم.

مثال ۶۹ :

می‌خواهیم مقدار اعشاری درون رجیستر D4 را با دو مقدار اعشاری که درون رجیسترهای D0 و D2 قرار دارند مقایسه کنیم.





دستور مقایسه زمانی TCMP :

از این دستور برای مقایسه زمان داخلی (تاریخ و ساعت) PLC یا همان RTC با زمان مورد نظرمان استفاده می‌شود.

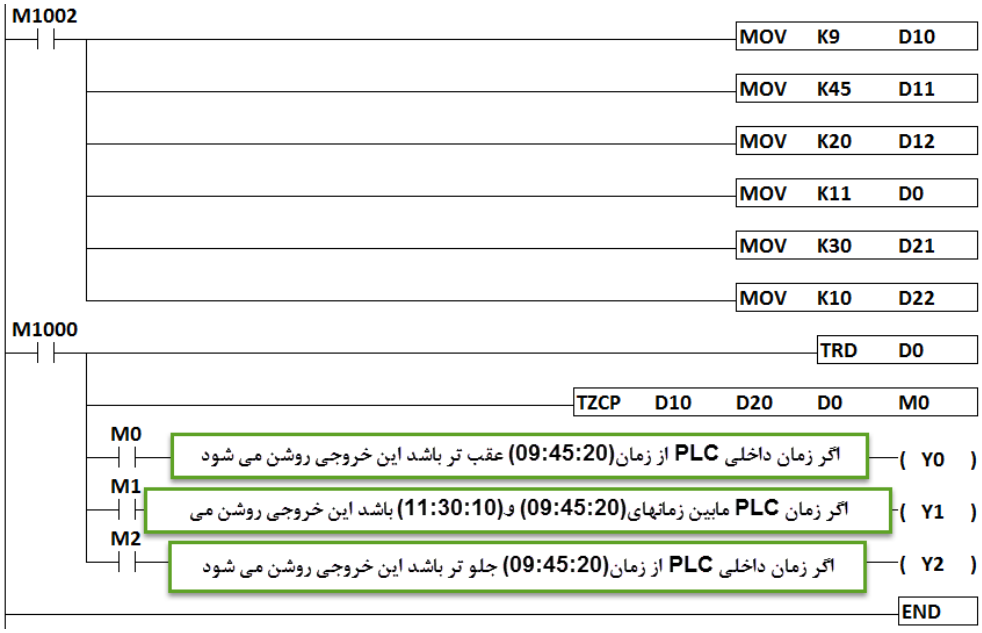
مثال ۷۰ :

توضیح : در خط اول این برنامه تاریخ داخلی PLC یا همانRTC توسط دستور TRD خوانده شده و درون رجیسترهای D0 تا D6 ریخته می‌شود در اینصورت رجیسترهای D4 مقدار ساعت ، D5 مقدار دقیقه و D6 مقدار ثانیه را در خود جای می دهد و ما در این مثال می‌خواهیم ساعت ، دقیقه و ثانیه‌ی PLC را با مقدار مورد نظرمان یعنی ساعت ۱۵ و دقیقه ۱۰ و ثانیه ۳۰ مقایسه کنیم. لذا در خط دوم از D4 استفاده می‌کنیم که شروع ساعت، دقیقه و ثانیه است.

دستور مقایسه زمانی ناحیه‌ای TZCP :

برای مثال می‌خواهیم زمان داخلی PLC را با دو زمان متفاوت که به برنامه می‌دهیم مقایسه کنیم.

مثال ۷۱ :



توضیح : تاریخ داخلی PLC یا همانRTC خوانده شده و درون رجیسترهای D0 تا D6 ریخته می‌شود در اینصورت رجیسترهای D4 مقدار ساعت ، D5 مقدار دقیقه و D6 مقدار ثانیه را در خود جای می دهد همچنین زمان مورد نظر اول، را توسط دستورهای MOV در رجیسترهای D10 و D11 و D12 و زمان مورد نظر دوم را توسط دستور MOV در رجیسترهای D20 و D21 و D22 قرار می دهیم . چنانچه ساعت جاری PLC از زمان اول عقب تر بود خروجی Y0 فعال می‌شود و اگر ساعت جاری PLC بین دو زمان مورد نظر ما باشد خروجی Y1 فعال می‌شود و در صورتیکه زمان جاری PLC از زمان دوم جلوتر باشد خروجی Y2 فعال می‌شود .

دستور SPD :

با استفاده از این دستور میتوان سرعت دوران یک محور را اندازه گیری کرد برای این کار می توان از یک سنسور استفاده کرد بطوریکه به ازای هر دور چرخش شفت ، یک پالس توسط سنسور تولید شود. این پالسهای تولید شده توسط سنسور، از طریق ورودیهای X و دستور SPD خوانده می شوند. برای محاسبه سرعت چرخش یک شفت بر حسب دور بر دقیقه (RPM) می توان از رابطه زیر استفاده کرد

$$\mathbf{N} = \frac{\mathbf{60}(\boldsymbol{D0})}{\boldsymbol{n}\times\boldsymbol{t}} \times \mathbf{10^3} \quad \text{(RPM)}$$

در این رابطه N سرعت چرخش شفت بر حسب دور بر دقیقه می باشد .

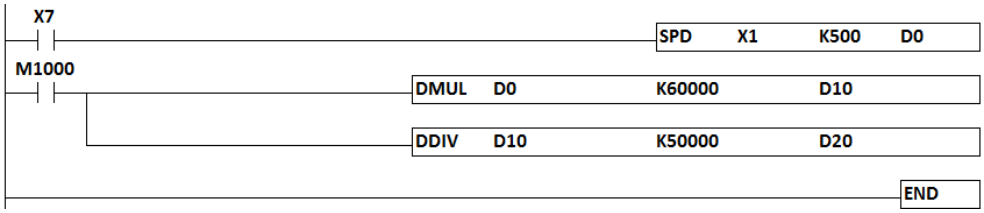
D0 تعداد پالسهای که در یک دوره زمانی مشخص دریافت شده است .

n تعداد پالس در یک دور کامل شفت می‌باشد .

t مدت زمانیکه که پالسهای ورودی دریافت شده‌اند .

مثال ۷۲ :

می‌خواهیم سرعت چرخش یک محور را با استفاده از یک سنسور نوری بر حسب RPM محاسبه کنیم.



سطر شماره یک برنامه دستور SPD برای خواندن پالسها از طریق ورودی X1 می باشد مدت زمان نمونه برداری پالسها، 500ms می باشد .

نکته مهم : دستور فوق 5 رجیستر متوالی را اشغال می کند در این مثال از رجیستر D0 تا D4 اشغال می‌شود .

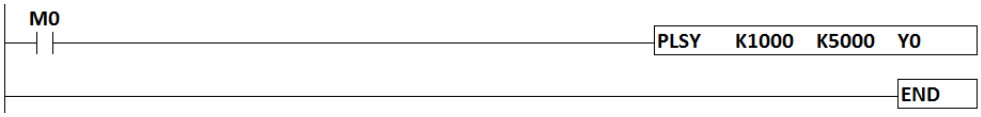
خطهای بعدی برنامه جهت محاسبه فرمول بیان شده در بالا برای بدست آوردن سرعت می‌باشد .

دستور PLSY :

با استفاده از این دستور می‌توان تعداد مشخصی پالس را با فرکانس مشخصی به یک خروجی ارسال کرد.

مثال ۷۳ :

می‌خواهیم با تحریک یک فلگ مانند M0، تعداد 5000 پالس را با فرکانس 1000 پالس در ثانیه (1KHz) در خروجی Y0 ارسال کنیم.



نکته : پس از پایان ارسال پالس به خروجی Y0 ، یک فلگ خاص روشن می‌شود که نشان دهنده‌ی پایان ارسال پالس هستند. این فلگهای خاص مطابق زیر می باشند:

M1029 : این فلگ پس از پایان ارسال پالس به خروجی Y0، فعال می‌شود .

M1030 : این فلگ پس از پایان ارسال پالس به خروجی Y1 ،فعال می‌شود.

M1102 : این فلگ پس از پایان ارسال پالس به خروجی Y2 ،فعال می‌شود.

M1103 : این فلگ پس از پایان ارسال پالس به خروجی Y3، فعال می‌شود.

رجیسترهای خاص D1220 و D1221 مرتبط با دستور PLSY بوده و مطابق جدول زیر تنظیم می شوند و در نتیجه عملکرد دستورPLSY مطابق آنها تغییر می‌کند .

Mode / Output	D1220						D1221					
	K0		K1	K2	K3		K0		K1	K2	K3#	
Y0	Pulse		Pulse	A	CW							
Y1		Pulse	Dir	B		Pulse						
Y2							Pulse		Pulse	A	CCW	
Y3								Pulse	Dir	B		Pulse

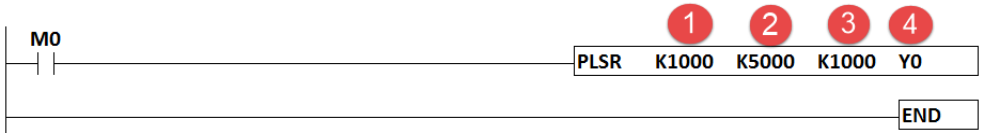
Pulse: Pulse A: A phase pulse CW: clockwise
Dir: Direction B: B phase pulse CCW: Counter-clockwise

مطابق جدول، چنانچه D1220 برابر 1 شودردر این صورت پالسهای خروجی به خروجیهای استفاده شده در دستور یعنی Y0 و یا Y1 ارسال می‌شود .چنانچه مقدار رجیستر D1220 برابر 1 انتخاب شود خروجی Y0 به عنوان پالس و همزمان خروجی Y1 به عنوان Dir یا جهت مورد استفاده قرار می گیرد این حالت به عنوان مثال در کنترل جهت چرخش در سرو موتور ها مورد استفاده قرار می گیرد. چنانچه 2 انتخاب شود خروجی Y0 و Y1 به صورت دوفاز پالس ارسال خواهند کرد بطوریکه اختلاف فاز انها ۹۰ درجه خواهد بود . انتخابهای دیگر نیز مطابق جدول خواهد بود .

دستور PLSR :

با استفاده از این دستور می‌توان به صورت صعودی و یا نزولی، پالسهایی را به خروجیهای تعیین شده، ارسال کرد. به مثال توجه کنید.

مثال ۷۴ :



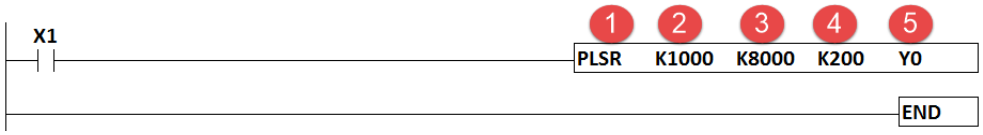
۱ – فرکانس خروجی ۲ – تعداد پالس

۳ – مدت زمان شتابگیری(Acc) و توقف(Dec) بر حسب میلی ثانیه

۴ – خروجی پالس (فقط Y0 و Y1)

با تحریک ورودی M0 تعداد 10000 پالس با فرکانس 1KHz و زمان شتابگیری(Acc.Time) 1000mS و زمان توقف(Dece.Time) 1000mS به خروجی Y0 ارسال می‌شود.

مثال ۷۵:



۱-دستور ارسال پالس با قابلیت Acc و Dec

۲-فرکانس ارسال پالس

۳-تعداد پالس ارسالی

۴-مدت زمان شیب افزایش(Acc) یا کاهش(Dec) سرعت

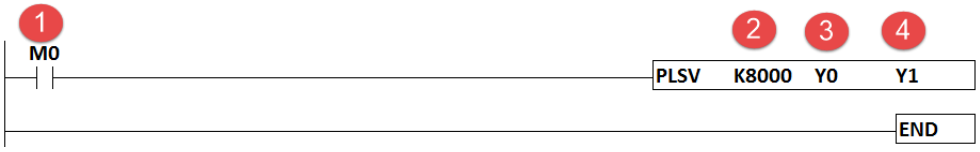
۵-خروجی ارسال پالس

این دستور بعد فعال شدن ورودی X1 ، طی مدت 200 میلی ثانیه سرعت را به تدریج تا حداکثر سرعت ارسال پالس یعنی 1000HZ افزایش می‌دهد همچنین 200 میلی ثانیه قبل از اتمام ارسال پالسها، یعنی 8000P ، به تدریج سرعت ارسال پالس را کاهش میدهد تا اینکه تعداد پالس به اتمام برسد . زمانیکه از این دستور برای کنترل سروو موتور ها استفاده می‌شود باعث حرکت نرم در لحظه راه‌اندازی و لحظه توقف می‌شود . این ویژگی باعث کاهش ضربات و استرس های مکانیکی در لحظه راه اندازی و لحظه توقف می‌شود همچنین دقت توقف را در موقعیت‌های مورد نظر افزایش می دهد .

دستور PLSV :

در این دستور تعداد پالس ذکر نمی‌شود و تنها فرکانس پالس مد نظر می‌باشد.

مثال ۷۶ :



۱ – با فعال شدن حافظه M0 ارسال پالس به خروجی های Y0 شروع می‌شود و با غیرفعال شدن این حافظه، ارسال پالس به این خروجی خاتمه می یابد.

۲ – فرکانس پالس ۳ – خروجی مورد نظر برای ارسال پالس

۴ – خروجی تعیین جهت چرخش موتور. در صورتیکه مقدار فرکانس منفی باشد وضعیت این خروجی یک شده و جهت چرخش (سرو موتور) عوض خواهد شد.

با استفاده از این دستور در حین ارسال پالس ، می‌توان تعداد پالس ارسالی به خروجی را تغییر داد.

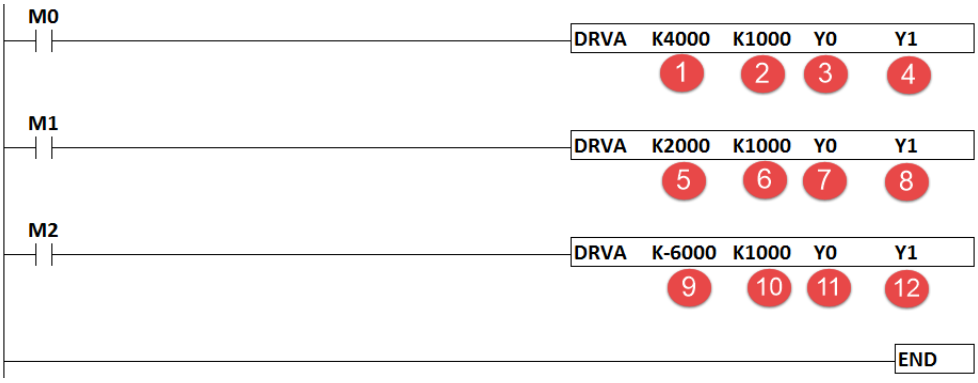
نکته : در این دستور چنانچه از خروجی Y0 برای ارسال پالس استفاده شود ، باید از خروجی Y1 به عنوان جهت (Dir) استفاده کرد . با استفاده از این دستور، می‌توان قابلیت Forward و یا Reverse داشت .

دستور DRVA :

این دستور برای تولید پالس مطلق می‌باشد با ارائه یک مثال به بررسی بیشتر این دستور برای فهم بهتر می‌پردازیم.

مثال ۷۷ :

فرض کنید می‌خواهیم تعداد 3000 پالس به سروو موتور ارسال کنیم تا از موقعیت 0 به موقعیت 1 برسد سپس تعداد 5000 پالس ارسال کنید تا از موقعیت 1 به موقعیت 2 برسد و سپس با ارسال -5000 پالس به موقعیت 0 بازگردد.



توضیحات برنامه به شرح زیر می‌باشد:

۱ –ارسال 4000 پالس به خروجی Y0 و رفتن از موقعیت 0 به موقعیت 1 (جهت چرخش راستگرد Y1=0)

۲ – ارسال پالسها با فرکانس 1000 پالس در ثانیه (1KHz)

۳ – خروجی پالسها

۴ – خروجی برای تعیین جهت(Forward/Reverse) . به عنوان مثال زمانیکه از سروو موتور استفاده کنیم در صورتیکه تعداد پالسها، عددی منفی باشد، مثلا k-3000 ، خروجی Y1 فعال شده و سروو در جهت مخالف میچرخد.

۵ – تعداد پالس مورد نظر (2000 پالس) و رفتن از موقعیت 1 به موقیت 2

۶ – ارسال پالس با فرکانس 1KHz

۷ – خروجی پالس

۸ - خروجی برای تعیین جهت چرخش سروو موتور (در صورتیکه تعداد پالس مورد نظر ما که درگزینه اول گفته شده ، عددی منفی باشد، مثلا k-6000 ، خروجی Y1 روشن شده و به صورت اتوماتیک سروو در جهت مخالف به تعداد K-6000 میچرخد).

۹ – تعداد پالس مورد نظر برای برگشتن به موقعیت 0 (6000- پالس)

۱۰ – ارسال پالس با فرکانس 1KHz

۱۱ – خروجی پالس

۱۲ – این خروجی در این قسمت فعال می‌شود و سروو موتور در جهت عکس میچرخد زیرا تعداد پالس مورد نظر ما یک عدد منفی می‌باشد (6000- پالس)

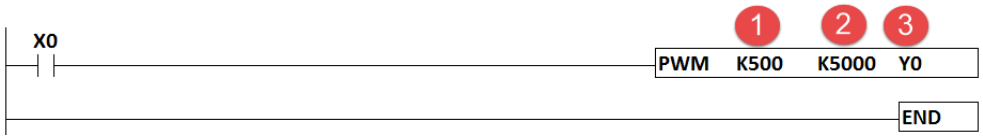
نکته : در شکل بالا با اعمال 4000 پالس، سروو به موقعیت 1 میرود . با نوشتن دستور بعدی و ارسال 2000 پالس، سروو تنها 2000 پالس دریافت می‌کند تا به موقعیت 2 برسد زیرا مبنای محاسبه‌ی موقعیت، ابتدای حرکت سروو موتور می‌باشد .

نکته : برای تعداد پالسهای بیشتر از 32767-/+ ، باید دستور را بصورت 32 بیتی مورد استفاده قرار دهیم برای این کار کافیست دستور را بصورت DDRVA بنویسیم.

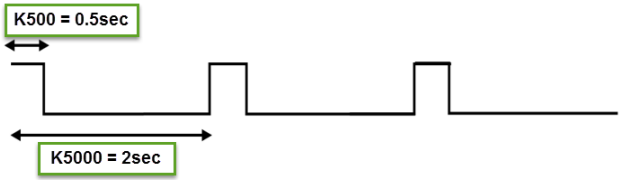
دستور PWM :

این دستور برای تولید پالس با دوره تناوب متغیر می‌باشد.

مثال ۷۸ :



۱ – طول پالس ۲ – دوره یا تناوب پالس ۳ – خروجی مورد نظر



پالسی که تولید کرده ایم 0.5s روشن بوده و 2s خاموش خواهد بود. ارسال پالس تا زمانی که ورودی فعال است، ادامه دارد.

دستورات وقفه (interrupt) :

سرعت پردازش plc ها در اکثر موارد به اندازه کافی بالا بوده تا مشکلی در روند برنامه پیش نیاید اما در برخی موارد ممکن است بدلیل حجم بالای برنامه ، مدت زمان پردازش زیاد شود. حال در نظر بگیرید میزان فشار یک مخزن توسط یک سنسور فشار کنترل می‌شود این مقدار فشار نباید از حد مجاز خود حتی در کسری از ثانیه فراتر رود و در صورت افزایش فشار، باعث ایجاد مشکل و خساراتی در سیستم خواهد شد. حال فرض کنید که در حین پردازش برنامه ، خطی از برنامه که مربوط به کنترل فشار می‌باشد، فشار بالاتر از حد مجاز را نشان دهد در این موارد باید پردازش برنامه اصلی موقتا متوقف شده و برنامه دیگری که باعث می‌شود فشار سیستم کاهش پیدا کند، اجرا شود (برنامه وقفه) . بعد از پایان پردازش برنامه وقفه، مجددا پردازش برنامه اصلی از همان خطی که متوقف شده بود، ادامه می یابد.

انواع وقفه در PLC دلتا :

۱ – وقفه های خارجی ۲ – وقفه های زمانی ۳ – وقفه های شمارنده های سرعت بالا

۴ – وقفه های ارتباطی ۵ – وقفه های پالس

Model / Interupt type	EH3/SV2	ES2/EX2	SA2/SX2	SS2	SE	MC	SX	ES/EX/EC3
External interupt	V	V	V	V	V	---	V	V
Time interrupt	V	V	V	V	V	V	V	V
HSC interrupt	V	V	V	V	V	---	V	---
Communication interupt	V	V	V	V	V	V	V	V
Pulse interupt	V	---	---	---	---	---	---	---

علامت V نشان دهنده ساپورت کردن و علامت --- نشان دهنده ی عدم ساپورت میباشد.

وقفه های خارجی	I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 points (01: rising-edge trigger ⌐, 00: falling-edge trigger ¬)
وقفه های زمانی	I602~I699, I702~I799, 2 points (Timer resolution = 1ms) I805~I899, 1 point (Timer resolution = 0.1ms) (Supported by V2.00 and above)
وقفه های HSC	I010, I020, I030, I040, I050, I060, I070, I080, 8 points
وقفه های شبکه	I140(COM1), I150(COM2), I160(COM3), 3 points, (*3)

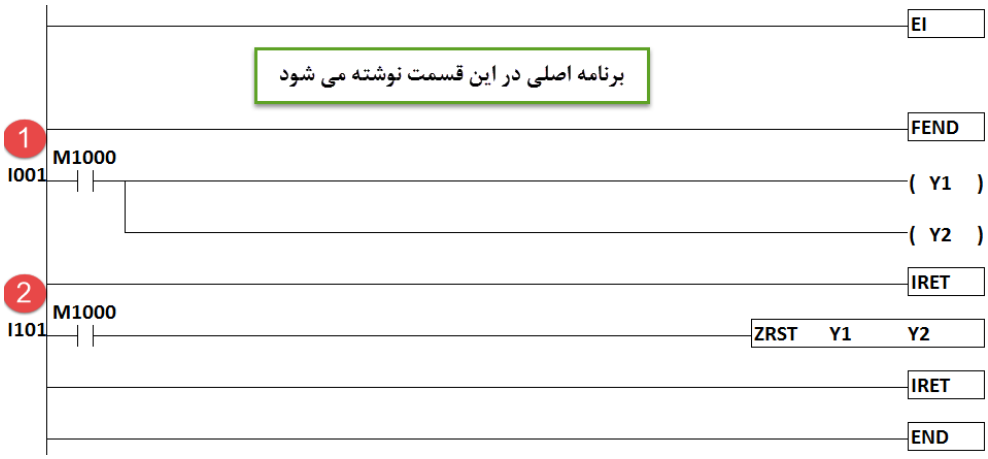
وقفه ها نیز مانند تایمرها و کانترها دارای شماره می‌باشند.

وقفه خارجی (سخت افزاری) :

در مثال زیر از وقفه خارجی شماره I001 استفاده شده است لذا زمانیکه که X0 فعال شود، این وقفه نیز فعال شده و زیربرنامه مربوط به آن نیز فورا اجرا شده و در پایان مجددا به برنامه اصلی باز می‌گردد.

مثال ۷۹ :

فرض کنید در کنار برنامه اصلی، یک برنامه بصورت وقفه(زیربرنامه وقفه) برای اتاقک کنترل نوشته‌اید که طبق آن در صورت ایجاد دود یا آتش ، سنسور X0 فعال شود(وقفه خارجی) در این صورت فورا برنامه اصلی موقتا متوقف شده و برنامه وقفه که با برچسب I001 مشخص شده اجرا شود و آژیر (Y1) و اسپری آب پاش (Y2) فعال شوند و با زدن کلید ریست (X1)، آژیر و آب پاش غیر فعال شوند.



توضیح : در این برنامه از وقفه‌های خارجی (External interrupt) که گاهی به آن وقفه‌ی سخت افزاری نیز گفته می‌شود، استفاده شده است یعنی با فعال شدن هر ورودی ، وقفه‌ی مربوط به همان ورودی فعال می‌شود.

در این مثال از **I001** و **I101** استفاده شده است که طبق جدول وقفه‌ها متعلق به X0 و X1 می‌باشند. طبق این برنامه با تشخیص دود و یا آتش، ورودی X0 تحریک شده در نتیجه یک وقفه سخت افزاری اتفاق می‌افتد سپس پردازش برنامه اصلی متوقف شده و زیربرنامه‌ی وقفه I001 که مخصوص وقفه سخت افزاری ورودی X0 می‌باشد ، اجرا می‌شود و خروجی های Y0 وY1 که مربوط به آبپاش و آژیر هستند فعال می‌شوند. در ادامه با زدن دکمه‌ی ریست که X1 می‌باشد ، زیر برنامه وقفه‌ی I101 فعال شده و آبپاش و آژیر را متوقف می‌کند.

دستور **EI** به معنای فعال کردن قابلیت وقفه در برنامه، مورد استفاده قرار می گیرد .

دستور **FEND** هم پایان برنامه اصلی را نشان می دهد .

نکته : فقط تایمرهای T192 ... T199 می توانند درون زیربرنامه‌ها، مورد استفاده قرار گیرند .

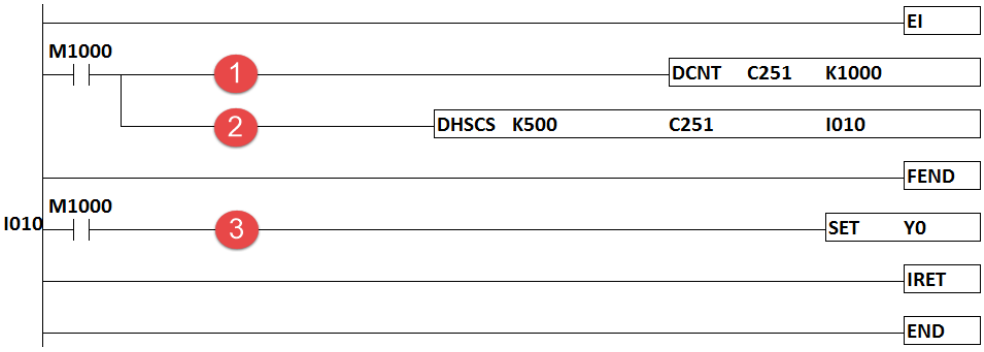
در این برنامه از دستور **IRET** به معنای برگشت از وقفه استفاده شده است. زمانیکه پردازش برنامه در زیربرنامه‌ی وقفه، به اتمام رسید، این دستور اجرای برنامه را به برنامه اصلی که قبلا متوقف شده بود منتقل می کند .

کاربرد وقفه در شمارنده های سرعت بالا DHSCS :

فرض کنید در برنامه‌ای پالسهای تولید شده توسط یک سنسور، از طریق شمارنده‌ی سرعت بالا در حال شمارش می‌باشد و می‌خواهیم در صورتیکه تعداد پالس شمارش شده به عدد مشخصی رسید ، یک زیربرنامه(وقفه) فعال شود. بدین منظور از دستور DHSCS استفاده می‌کنیم. مثال زیر را ببینید .

مثال ۸۰ :

این برنامه، تعداد پالس تولید شده توسط یک سنسور را به وسیله‌ی کانتر C251 شمارش می کند. سپس از طریق دستور DHSCS مقدار شمارش شده توسط کانتر C251 با مقدار ثابت K500 مقایسه شده و زمانی که مقدار کانتر C251 برابر عدد K500 شد وقفه کانتر سرعت بالا فعال شده و زیربرنامه وقفه اجرا می‌شود و در این زیربرنامه خروجی Y0 فورا روشن شده و منتظر سیکل جاری نمی ماند .



۱ – در این خط از برنامه کانتر شماره C251 در حال شمارش پالسهای ورودی می‌باشد در صورتیکه تعداد این پالس ها برابر K1000 برسد ، کنتاکت C251 بسته خواهد شد. البته از این کنتاکت در این مثال استفاده نشده است .

۲ – در این خط توسط دستور DHSCS به محض رسیدن تعداد پالسها به عدد K500 وقفه‌ی I010 فعال می‌شود. در این خط بجای استفاده از وقفه می‌توان از Y ، M و S استفاده کرد.

۳ – با اجرای زیربرنامه وقفه‌ی I010 درخط۳ ، خروجی Y0 فورا روشن می‌شود.

وقفه زمانی :

چنانچه بخواهیم عملی بطور منظم و در بازه‌های زمانی مشخص و البته دقیق اجرا شود از این نوع وقفه استفاده می شود .مثال زیر را ببینید .

مثال ۸۱ :

۱-دستور EI در خط اول برنامه جهت فعال کردن قابلیت وقفه می باشد .

۲-خط دوم برنامه که در واقع یک برنامه نمونه بوده و هر برنامه دیگری با هر تعداد خط میتواند باشد .

۳-خط سوم برنامه معرف پایان بخش اصلی برنامه می باشد .

۴-عبارت I799 در واقع بیانگر وقفه‌ی زمانی بوده و عدد 99 در آن به معنای 99ms میلی ثانیه می باشد به عبارتی هر 99 میلی ثانیه یک بار وقفه زمانی اجرا می شود . حداقل زمان 2ms میلی ثانیه می باشد

رزولوشن تایم وقفه زمانی 1ms می باشد .

نکته : در مجموع دو وقفه زمانی I699 و I799 داریم که در هر دو زمان وقفه از 2ms الی 99ms خواهد بود .

۵-هر بار که وقفه زمانی اجرا می شود یک واحد به مقدار قبلی D0 توسط دستور INC اضافه می شود .

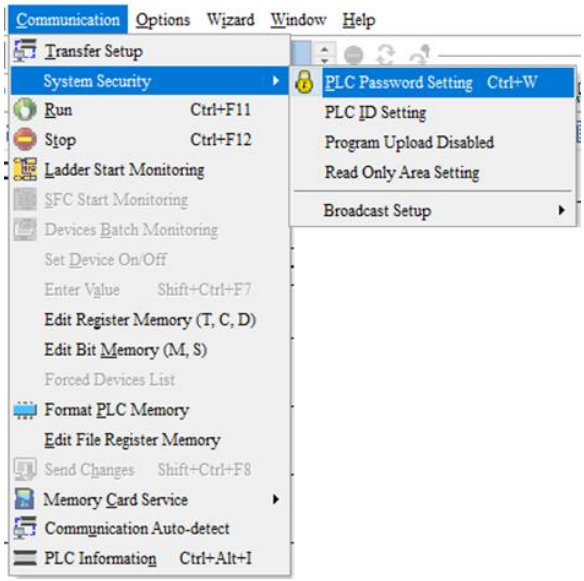
۶-زمانیکه برنامه به این دستور می رسد از زیربرنامه‌ی وقفه خارج و به برنامه اصلی باز می‌گردد . البته دقیقا به همان خطی از برنامه اصلی باز می‌گردد که وقفه ایجاد شده بود .

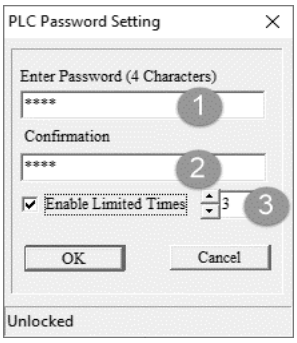
نکته : با فعال کردن فلگ M1057 می توانیم وقفه زمانی را غیر فعال کنیم .

پسورد گذاری در PLC های دلتا :

اصولا نوشتن برنامه برای PLC های به دو صورت باز و بسته (Open source / Close source) انجام می‌شود که برنامه نویس می‌تواند برای برنامه‌ای که می نویسد یک پسورد تنظیم کند تا از دسترسی افراد غیر مجاز به برنامه جلوگیری کند.

برای پسوردگذاری در PLCهای دلتا کافیست، مطابق شکل زیر از منوی برنامه وارد قسمت Communication شده و از منوی باز شده وارد قسمت System security شوید و از منوی جدید باز شده گزینه‌ی PLC password setting را انتخاب کنید. در ضمن می‌توانید از کلید میانبر Ctrl+w نیز استفاده کنید.





با انتخاب گزینه PLC Password Setting یک پنجره به شکل زیر باز خواهد شد اطلاعات خواسته شده را در این پنجره وارد کنید.

۱ – وارد کردن رمز مورد نظر (حداکثر ۴ کاراکتر)

۲ – وارد کردن مجدد همان رمز جهت تایید

۳ – در این قسمت در صورتیکه تیک گزینه‌ی Enable Limited Times فعال باشد به این معناست که در صورتیکه تعداد دفعات وارد کردن رمز بیش از عدد تعیین شده گردد، برنامه دیگر قابل بازیابی نخواهد بود و باید PLC را ریست کنید.

۷۸.....اتوماسیون صنعتی DELTA

فصل پنجم

آنالوگ

معرفی انواع کارتهای آنالوگ و دما
روش تنظیم رجیسترهای داخلی کارت های آنالوگ و دما
نحوه سیم بندی و اتصال کارت های آنالوک و دما به PLC دلتا

در PLC های دلتا ورودی و خروجی ها یا به صورت دیجیتال (ON / OFF) هستند یا به صورت آنالوگ (بازه پیوسته).

در طبیعت کلیه کمیت ها بصورت آنالوگ می‌باشند مانند فشار ، رطوبت ، حرارت .

اما از آنجایی که پردازشگر ها ماهیت دیجیتالی دارند ، نمی‌توانند بطور مستقیم سیگنالهای آنالوگ را پردازش کنند . برای این منظور از مبدل ها برای تبدیل کمیتهای آنالوگ به دیجیتال (ADC) و یا تبدیل کمیتهای دیجیتال به آنالوگ (DAC) استفاده می‌شود. کارتهای آنالوگ و کارتهای دما در واقع مبدلهای کمیتهای آنالوگ به دیجیتال هستند . کمپانی دلتا کارتهای دما و آنالوگ متنوعی تولید کرده که در این فصل با آنها آشنا می شویم .

کارت ورودی آنالوگ :

این کارتها در سه سری S و S2 و SL عرضه شده است که سری های S و S2 از سمت راست و سری SL از سمت چپ به PLC متصل می‌شود .

کارت خروجی آنالوگ :

این کارتها نیز مانند کارت آنالوگ ورودی ، در سه سری S و S2 و SL عرضه شده است که سری های S و S2 از سمت راست و سری SL از سمت چپ به PLC متصل می‌شود.

کارت ورودی و خروجی :

این کارتها تنها در دو سری S و S2 تولید شده‌اند که همزمان دارای تعدادی ورودی و خروجی آنالوگ می باشند .در ادامه به بررسی مشخصات فنی آن خواهیم پرداخت.

نکته : تفاوت کارتهای سری S و S2 و SL در دقت(رزولوشن) این کارت ها می‌باشد.

انواع کارت دما :

04PT و 04TC و DTC1000/2000 :

نکته : کارتهای DTC1000/2000 نیازی به PLC نداشته و خودشان کنترلر دمای مستقلی بوده و دارای خروجی با قابلیت PID و قابلیتهای دیگری می‌باشند که از طریق نرم افزار DTCOM برنامه ریزی می‌شود و یا می‌توان آنرا از طریق شبکه مدباس به PLC ویا HMI متصل نمود. (برای توضیحات بیشتر به کتاب مثالهای کاربردی مدباس گروه شارکنترل مراجعه شود) .

کارت های 04PT و 04TC کارتهای دمایی می باشند که بطور مستقیم قابلیت اتصال به PLC دلتا را دارند همچنین میتوانند بطور غیر مستقیم و از طریق شبکه مدباس با PLC و یا HMI دلتا ارتباط برقرار کنند .

برای اطلاع از مشخصات فنی این کارتها به جدولی که در صفحه بعد ارائه شده است مراجعه کنید.

مشخصات فنی کارتهای آنالوگ سری S :

مشخصات	DVP06XA-S	DVP06XA-S2	DVP04DA-S	DVP04DA-S2	DVP04AD-S	DVP04AD-S2	DVP06AD-S
تغذیه	24VDC						
تعداد کانال ورودی	4	4	---	4	4	4	6
تعداد کانال خروجی	2	2	4	---	---	---	---
نفت ورودی آنالوگ	دقتی: 12bit خروجی: 11bit	دقتی: 12/14 bit خروجی: 11/13 bit	---		دقتی: 14 bit خروجی: 13 bit	دقتی: 14/16 bit خروجی: 13/15 bit	دقتی: 14 bit خروجی: 13 bit
نفت خروجی آنالوگ	12 bit		12 bit	12/16 bit		---	---
مقادیر ورودی (مقدار عددی در PLC)	±10V ... ±2000 ±20mA ... ±1000	±10V ... (±2000_±8000) ±20mA ... (±1000_±4000)			±10V ... ±8000 ±20mA ... ±4000	±10V ... (±8000_±32000) ±20mA ... (±4000_±16000)	±10V ... ±8000 ±20mA ... ±4000
مقادیر خروجی (مقدار عددی در PLC)	0~10V ... 0~4000 0~20mA ... 0~4000	0~10V ... 0~4000 0~20mA ... 0~4000 ±10V ... ±2000	0~10V ... 0~4000 0~20mA ... 0~4000		±10V ... (±8000_±32000) ±20mA ... (±4000_±16000)		
زمان آپدیت کردن	سرعت کلیه ها: 3ms						
نحوه اتصال به PLC	اتصال از سمت راست CPU با آدرس دهی 0~7						
پورت RS485	دارای پورت RS485 و پروتکل های MODBUS RTU/ ASCII						
حداکثر ورودی آنالوگ	±15V ,±32mA						

مشخصات فنی کارتهای آنالوگ که از سمت چپ به PLC متصل می‌شوند :

مشخصات	DVP04DA-SL	DVP04AD-SL
تغذیه	24VDC	
تعداد کانال آنالوگ	4 خروجی	4 ورودی
دقت	16 Bit	16 Bit
مقدار عددی در plc	$\pm 10v \dots \pm 32000$ $0\sim 20mA, 4\sim 20mA \dots 0\sim 32000$	$\pm 10v, \pm 5v \dots . \pm 32000$ $\pm 20mA \dots . \pm 32000$ $0\sim 20mA, 4\sim 20mA \dots . 0\sim 32000$
زمان اندازه گیری	$250\mu s$	
نحوه اتصال به CPU	از سمت چپ CPU با آدرس دهی 100~107	
پورت RS485	دارای پورت RS485 و پروتکلهای MODBUS RTU/ASCII	
حداکثر ورودی آنالوگ	$\pm 15V$, $\pm 32mA$	

همانطور که در جداول بالا ملاحظه می‌کنید ، بخشی به دقت کارتها اختصاص داده شده است که نشان می‌دهد این کارتها از 11 Bit تا 16 Bit طراحی و ساخته شده‌اند. حال می‌خواهیم بدانیم مفهوم این دقت‌ها چیست.

برای مثال فرض کنید یک کارت در اختیار دارید که دارای دقت 2 Bit می‌باشد. همچنین ورودی کارت آنالوگ مورد نظر بصورت ولتاژی و در بازه 10V ... 0 ولت باشد این کارت 2^2 یا 4 عدد دیجیتال شده‌ی مختلف متناسب با ورودی 10V... 0 تولید می‌کند . یعنی کارت شما به ازای هر 2.5V ولت یک عدد دیجیتال شده، تولید می‌کند :

مقدار دیجیتال	00	01	10	11
0 ... 10V	0 ... 2.5V	2.6 ... 5V	5.1 ... 7.5V	7.6 ... 10V

حال فرض کنید یک کارت با دقت 16 Bit دارید در اینصورت مقدار عدد دیجیتال تولید شده توسط کارت به ازای مقادیر آنالوگ ورودی در PLC دلتا به صورت زیر خواهد بود :

$$\mathbf{2^{16} \approx 64000 \rightarrow \pm 10V = \pm 32000}$$

یعنی در صورتیکه مقدار ورودی آنالوگ1V باشد مقدار دیجیتال شده برابر 3200 خواهد بود به عبارتی دیگر به ازای ورودی آنالوگ 0. 1V مقدار دیجیتال شده برابر 320 خواهد بود .

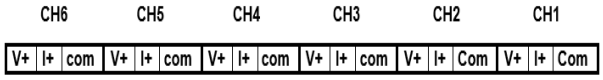
مثال ۸۲ :

فرض کنید توسط یک کارت آنالوگ DVP04AD-S (14 Bit) در حال خواندن مقدار یک سنسور فشار (Pressure Transmitter) هستید این سنسور به ازای 0 ~ 30 Bar ، ولتاژی برابر 0 ~ 10V ایجاد می‌کند. حال می‌خواهیم بدانیم به ازای فشار 15 Bar چه مقدار عدد دیجیتال در PLC خواهیم داشت .

طبق جدول صفحات قبل، این کارت آنالوگ، مقدار آنالوگ $\pm 10V$ ، را به مقدار دیجیتال شده ± 8000 تبدیل می کند لذا به ازای فشار 15 Bar ، مقدار ولتاژ آنالوگ ورودی کارت، برابر 5V+ خواهد بود که مقدار دیجیتال این ولتاژ با یک تناسب ساده برابر 4000 خواهد شد .

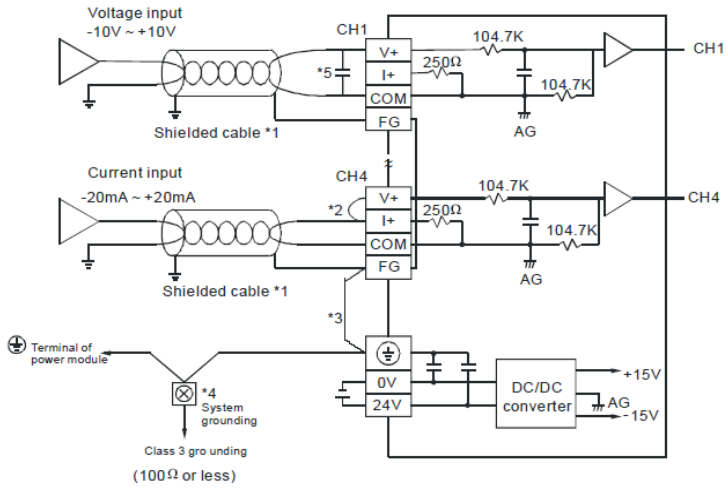
سیم بندی کارتهای آنالوگ :

DVP 06XA-S : این ماژول دارای چهار کانال ورودی برای سنسورهای آنالوگ (ولتاژی و جریانی) و همچنین دارای دو خروجی آنالوگ می‌باشد.



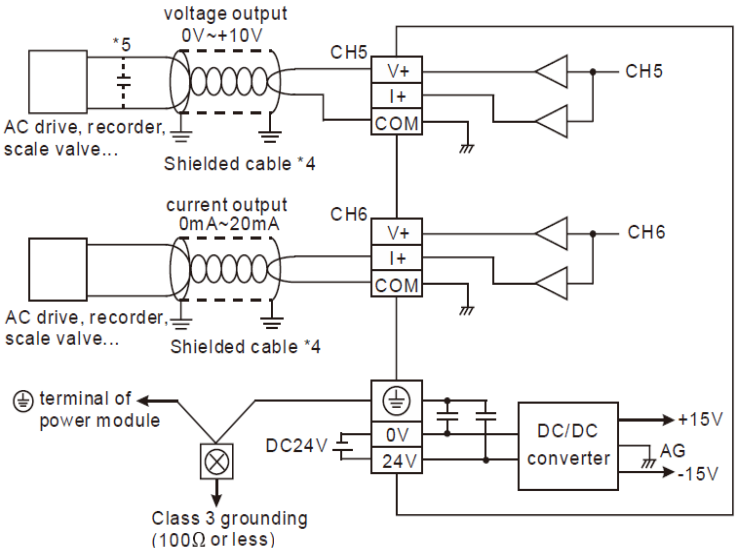
همانطور که در شکل می بینید این ماژول در هر کانال دارای سه ورودی می‌باشد.

در صورتیکه سنسور مورد استفاده ما از نوع جریانی باشد از ترمینالهای I+ و COM و در صورتیکه از نوع ولتاژی باشد از ترمینالهای V+ و COM استفاده می‌کنیم.



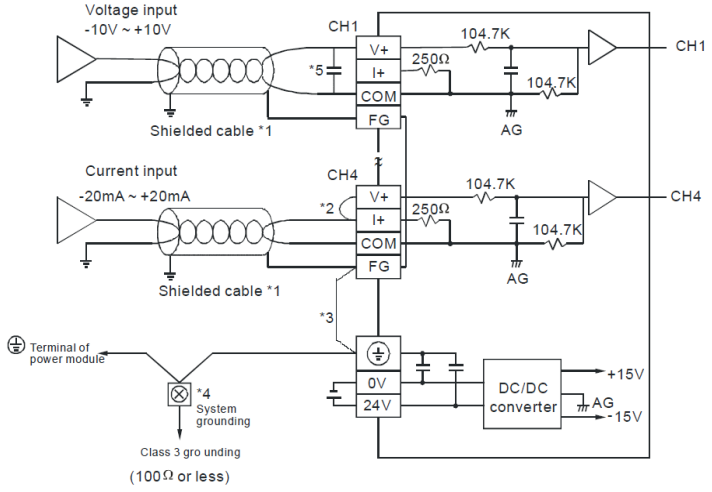
در تصویر بالا نحوه اتصال سنسور به ورودی های کارت 06XA توضیح داده شده است.

نکته : در حالت ورودی جریانی، طبق نقشه باید ترمینالهای V+ و I+ را به یکدیگر متصل کنید.



در تصویر بالا نیز نحوه سیم بندی در خروجی های کارت 06XA نشان داده شده است.

DVP04AD-S : این کارت آنالوگ دارای ۴ ورودی آنالوگ با رزولوشن 12-bit می‌باشد. این کارت مخصوص PLC های سری Slim دلتا بوده و از سمت راست به PLC متصل می‌شود.



در شکل بالا نحوه‌ی سیم بندی این کارت در حالت های ولتاژی و جریانی نشان داده شده است این نحوه ی سیم بندی دقیقا مشابه سیم بندی کارت آنالوگ 06XA می‌باشد.

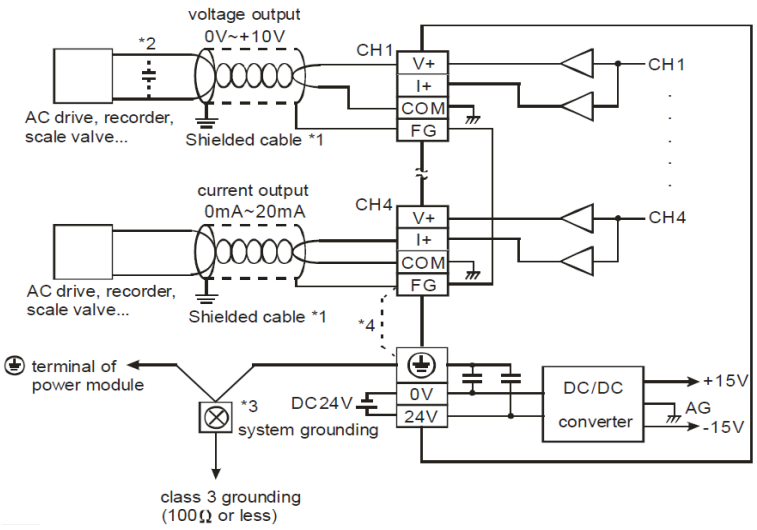
کانال چهارم				کانال سوم				کانال دوم				کانال اول			
FG	COM	I+	V+	FG	COM	I+	V+	FG	COM	I+	V+	FG	COM	I+	V+

نکته : در حالت جریانی باید ترمینالهای+V و +I را به یکدیگر متصل کنید.

نکته : در کارت‌های آنالوگ AD و DA و XA ، درحالت ولتاژی در صورتیکه محیط دارای نویز باشد عدد خوانده شده دارای پرش و نوسان خواهد بود در اینصورت می‌توانید با اضافه کردن یک خازن 0.1 تا 0.45 میکرو فاراد، نویز را تا حد زیادی کاهش دهید. همینطور می‌توانید با وصل کردن کانال FG به ارت یا بدنه‌ی تابلو اثر نویز را کاهش دهید.

DVP04DA-S : این کارت آنالوگ دارای 4 خروجی آنالوگ بوده و کانالها و نحوه‌ی سیم بندی آن به شکل زیر می‌باشد.

کانال چهارم				کانال سوم				کانال دوم				کانال اول			
FG	COM	I+	V+	FG	COM	I+	V+	FG	COM	I+	V+	FG	COM	I+	V+

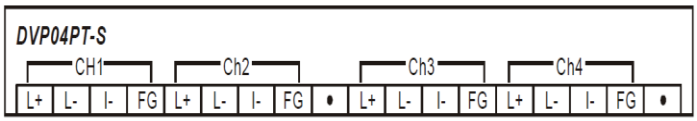


همانطور که در شکلهای بالا ملاحظه می‌کنید نحوه‌ی سیم بندی در کارت ورودی 04AD با کانالهای ورودی کارت 06XA مشابه بوده و نحوه ی سیم بندی در کارت خروجی 04DA و با کاناهای خروجی کارت 06XA مشابه می‌باشند.

DVP 04PT-S : این کارت دارای ۴ ورودی سنسور دما می‌باشد. سنسورهایی که این کارت پشتیبانی می‌کند عبارتند از:

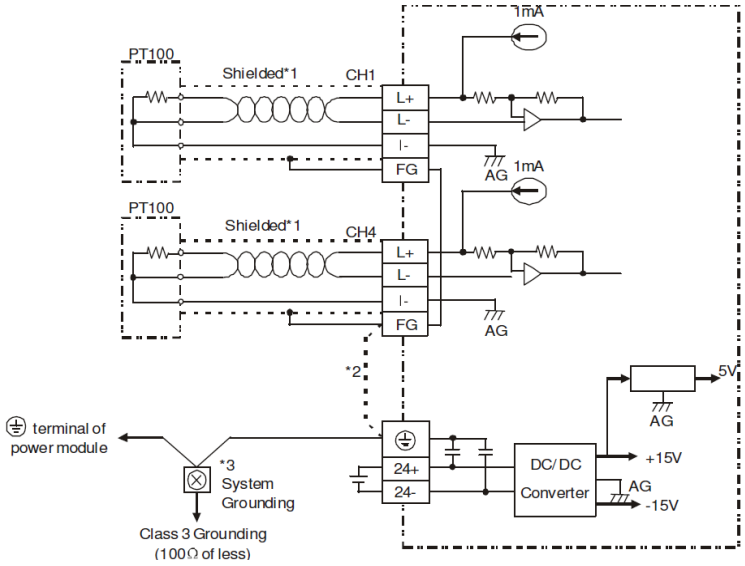
NI1000 NI100 PT1000 PT100

این کارت در هر یک از چهار کانال خود دارای ۴ ورودی برای سنسور می‌باشند. این کانال ها و ورودی ها بصورت زیر می‌باشند:



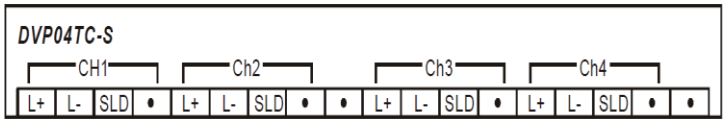
همانطور که در شکل بالا ملاحظه می‌کنید ، هر کانال دارای ۴ ورودی می‌باشد که شامل L+ ، L- ، I- و FG می‌باشد.

در سنسورهای PT سه سیمه ، معمولا یک رنگ را به L+ و دوسیم دیگر که معمولا هم‌رنگ هستند را به L- و I- متصل کنند. از آنجایی که دو سیم همرنگ سنسور PT نسبت به هم دارای مقاومت بسیار کمی می باشند، فرقی نمی‌کند کدام سیم به کدام یک از دو ترمینال L- و I- وصل شود.

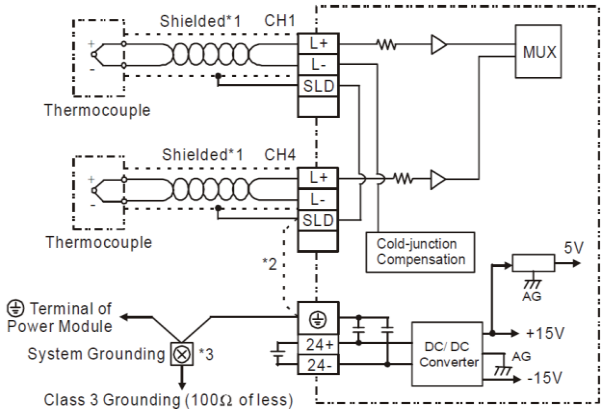


در تصویر بالا نحوه‌ی اتصال سنسور به کارت 04PT-S نشان داده شده است.

DVP 04TC-S : این ماژول از انواع سنسور دمای ترموکوپل مانند سری های K ، J ، S ، R ، T پشتیبانی می‌کند و دارای چهار ورودی سنسور دمای ترموکوپل می‌باشد.



همانطور که در شکل بالا می‌بینید هر کانال دارای چهار ورودی می‌باشد. برای اتصال سنسور به این کارت ، یک سیم (سرسیم قرمز) را به +L و سیم دیگر(معمولا سرسیم آبی) را به -L متصل کنید.

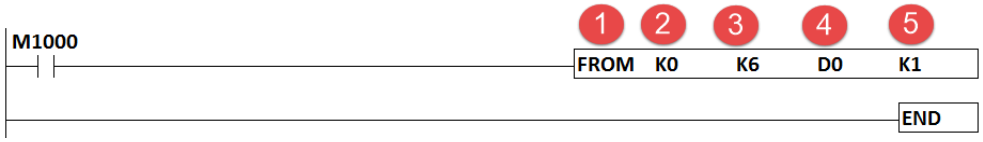


نحوه‌ی سیم کشی در کارت 04TC در شکل بالا نشان داده شده است.

مقادیر کارت‌های آنالوگ و دما توسط PLC با استفاده از دستور FROM خوانده می‌شود . کارت های آنالوگ و کارت دما قبل از شروع به کار، باید تنظیم شوند . جهت تنظیم پارامترهای کارت‌های آنالوگ از دستور TO استفاده می‌شود. حال به توضیح این دو دستور می‌پردازیم.

دستور خواندن از کارت آنالوگ و کارت دما توسط دستور FROM :

این دستور برای خواندن مقادیر از کارت های آنالوگ و دما استفاده می‌شود.



مطابق شکل بالا، به توضیح در مورد عملوندهای دستور FROM می‌پردازیم:

۱-دستور FROM جهت خواندن مقادیر کارت‌های آنالوگ و دما

۲- اولین کارت آنالوگی که در سمت راست PLC قرار می‌گیرد دارای آدرس پایه K0 می باشد. دومین کارت آنالوگ با K1 و نهایتا کارت هشتم با K7 نشان داده می‌شود. حتی اگر بین PLC و کارت آنالوگ تعدادی کارت دیجیتال قرار بگیرد ترتیب بالا تغییری نمی‌کند .

۳-دومین عملوند، عملوند مهمی در دستور FROM می‌باشد. به این عملوند، کنترل رجیستر(#CR) می‌گویند. کنترل رجیسترها که در فایل راهنمای هر کارت آنالوگ یا دما و یا کارت های دیگر، موجود هستند از K0 تا K34 می‌باشند. در ادامه توضیحاتی در خصوص کنترل رجیسترها ارائه خواهد شد.

۴-رجیستری که مقدار خوانده شده از کارت در آن قرار می‌گیرد .

۵- نشان دهنده تعداد کانالهایی از کارت است که اطلاعات آن باید خوانده شود.

کنترل رجیسترهای کارت 04PT-S :

■ Control Register

CR#	Address	Save		Register content	Description
#0	H'4064	O	R	Model name	Set up by the system: DVP04PT model code = H'8A.
#1	H'4065	O	R/W	Mode setting	CH1 mode: b0 ~ b3 CH2 mode: b4 ~ b7 CH3 mode: b8 ~ b11 CH4 mode: b12 ~ b15 Take CH1 mode (b3,b2,b1,b0) for example. The default value is H'0000. 1. (0,0,0,0): PT100 2. (0,0,0,1): NI100 3. (0,0,1,0): PT1000 4. (0,0,1,1): NI1000 5. (1,1,1,1): The channel is disabled.
#2	H'4066	O	R/W	CH1 average number	Number piece of readings used for the calculation of "average" temperature on channels CH1 ~ CH4. Setting range: For versions prior to V3.04: K1 ~ K4,095. For versions after V3.05: K1 ~ K20. Default setting is K10.
#3	H'4067	O	R/W	CH2 average number	
#4	H'4068	O	R/W	CH3 average number	
#5	H'4069	O	R/W	CH4 average number	
#6	H'406A	X	R	CH1 average degrees	Average degrees for channels CH1 ~ CH4. (Unit: 0.1°C).
#7	H'406B	X	R	CH2 average degrees	
#8	H'406C	X	R	CH3 average degrees	
#9	H'406D	X	R	CH4 average degrees	
#12	H'4070	X	R	CH1 average degrees	Average degrees for channels CH1 ~ CH4. (Unit: 0.1°F).
#13	H'4071	X	R	CH2 average degrees	
#14	H'4072	X	R	CH3 average degrees	
#15	H'4073	X	R	CH4 average degrees	
#18	H'4076	X	R	Present temp. of CH1	Present temperature of channels CH1 ~ CH4. (Unit: 0.1°C).
#19	H'4077	X	R	Present temp. of CH2	
#20	H'4078	X	R	Present temp. of CH3	
#21	H'4079	X	R	Present temp. of CH4	
#24	H'407C	X	R	Present temp. of CH1	Present temperature of channels CH1 ~ CH4. (Unit: 0.1°F).
#25	H'407D	X	R	Present temp. of CH2	
#26	H'407E	X	R	Present temp. of CH3	
#27	H'407F	X	R	Present temp. of CH4	
#29	H'4081	X	R/W	PID mode setting	Set H'5678 to enable PID mode, other set values are invalid. Default: H'0000.
#30	H'4082	X	R	Error status	Data register stores the error status. Refer to the error code chart for details.
#31	H'4083	O	R/W	Communication address setting	RS-485 communication address. Setting range is 01 ~ 254 and default setting is K1.
#32	H'4084	O	R/W	Communication baud rate setting	Communication baud rate. For ASCII mode, date format is 7 bits, even, 1 stop bit (7, E, 1), while RTU mode, date format is 8 bits, even, 1 stop bit (8, E, 1).
#32	H'4084	O	R/W	Communication baud rate setting	b0: 4,800 bps (bit/sec). b1: 9,600 bps (bit/sec). (default setting)
#32	H'4084	O	R/W	Communication baud rate setting	b2: 19,200 bps (bit/sec). b3: 38,400 bps (bit/sec). b4: 57600 bps (bit/sec). b5: 115,200 bps (bit/sec). b6 ~ b13: Reserved. b14: switch between low bit and high bit of CRC code (RTU mode only). b15: RTU mode.

#33	H'4085	O	R/W	Reset to default setting	b15~b12 | b11~ b9 | b8~b6 | b5~b3 | b2~b0 ERR LED | CH4 | CH3 | CH2 | CH1 Example: Setting of CH1 1. b0 ~ b1: Reserved. 2. b2: Set to 1 and PLC will be reset to default settings. Definition of ERR LED: b12~b15=1111 (default settings) 1. b12 corresponds to CH1: when b12=1, scale exceeds the range, ERR LED flashes. 2. b13 corresponds to CH2: when b13=1, scale exceeds the range, ERR LED flashes. 3. b14 corresponds to CH3: when b14=1, scale exceeds the range, ERR LED flashes. 4. b15 corresponds to CH4: when b15=1, scale exceeds the range, ERR LED flashes.
#34	H'4086	O	R	Software version	Display the software version in hexadecimal. ex: H'010A = version 1.0A
#35 ~ #48				System used	
Symbols: O: means latched. X: means not latched. (Support when using RS-485 communication, not support when connecting with MPU) R: able to read data by using FROM instruction or RS-485. W: able to write data by using TO instruction or RS-485.					

در جدول بالا کنترل رجیسترهای کارت 04PT-S را ملاحظه می‌کنید که از شماره K0 تا K32 می‌باشد.

K0 : این کنترل رجیستر، بیانگر نوع و مدل کارت می‌باشد. برای مثال در صورتیکه این کنترل رجیستر توسط دستور FROM خوانده شود، کد مدل کارت (H8A) که یک کد به صورت هگز می‌باشد نمایش داده می‌شود.

K1 : این کنترل رجیستر مهمترین کنترل رجیستر در بین ۳۴ کنترل رجیستر می‌باشد زیرا با استفاده از این کنترل رجیستر، می‌توان نوع سنسور ورودی کارتها را مشخص نمود.

K2 : خواندن دما و نمایش آن به صورت لحظه‌ای و یا میانگین می‌تواند باشد. زمانیکه از حالت میانگین دما استفاده کنیم، این کارت به صورت پیش فرض، ۱۰ بار از دمای محیط نمونه برداری کرده و میانگین آن به PLC ارسال می‌کند تعداد نمونه گیری توسط این کنترل رجیستر برای کانال اول قابل تغییر می‌باشد که از K1 تا K20 می‌توان تعداد نمونه گیری را تغییر داد.

K3 – K4 – K5 : این کنترل رجیسترها نیز مانند کنترل رجیستر قبلی عمل می‌کنند و برای کانالهای دیگرکارت می‌باشد .

K6 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال اول را برحسب درجه سانتیگراد و دقت 0.1 درجه، محاسبه کرد.

K7 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال دوم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K8 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال سوم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K9 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال چهارم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K12 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال اول را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K13 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال دوم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K14 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال سوم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K15 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال چهارم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K18 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال اول را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K19 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال دوم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K20 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال سوم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K21 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال چهارم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K24 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال اول را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K25 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال دوم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K26 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال سوم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K27 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال چهارم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K29 : این کنترل رجیستر برای تنظیمات PID می‌باشد.

K30 : این کنترل رجیستر وضعیت خطاها را بررسی می‌کند.

K31 : این کنترل رجیستر برای تنظیم آدرس ارتباطی در شبکه صنعتی MODBUS کاربرد دارد.

K32 : این کنترل رجیستر جهت تنظیم Baud rate در شبکه صنعتی MODBUS کاربرد دارد.

K33 : این کنترل رجیستر، تنظیمات کارت 04PT را ریست می‌کند.

K34 : این کنترل رجیستر ورژن نرم افزار کارت 04PT را مشخص می‌کند.

**در جدول کنترل رجیسترها علامت O نشان دهنده ی Latch (پایدار) بودن است.

**علامت X نشان دهنده ی Latch نبودن است.

**علامت R نشان دهنده ی خواندنی بودن کنترل رجیستر و علامت W نشان دهنده‌ی نوشتنی بودن کنترل رجیستر است.

مثال ۸۳ :

برای مثال اگر عدد مقابل این K عدد ۱ باشد ، در آنصورت فقط مقدار همان کانال خواسته شده خوانده خواهد شد و در رجیستر D0 ریخته می‌شود. در صورتیکه عدد مقابل K عدد ۲ باشد ، مقدار کانال خواسته شده (در اینجا از کنترل رجیستر K6 استفاده شده که طبق جدول کنترل رجیسترها به معنای مقدار میانگین دمای کانال اول می‌باشد) خوانده می‌شود و در رجیستر D0 ریخته می‌شود و همچنین مقدار کانال بعدی نیز خوانده شد و به صورت خودکار در رجیستر D1 ریخته می‌شود. به همین شکل اگر عدد مقابل این K عدد ۴ باشد، مقدار کانالهای اول تا چهارم خوانده شده و به صورت خودکار در رجیسترهای D0 تا D3 ریخته می‌شود.

برای مثال در صورتیکه بخواهیم هر چهار کانال کارت دمای 04PT را بخوانیم بجای اینکه چهار مرتبه از دستور FROM استفاده کنیم می‌توانیم ، به جای K1 در آخر این دستور ، از K4 استفاده کنیم تا مقادیر هر چهار کانال خوانده شد شود و به ترتیب در رجیسترها ریخته شود.

در ورودی های ولتاژی کارت آنالوگ به ازای 0 – 10V : 0 – 2000

در خروجی های ولتاژی کارت آنالوگ به ازای 0 – 4000 : 0 – 10V

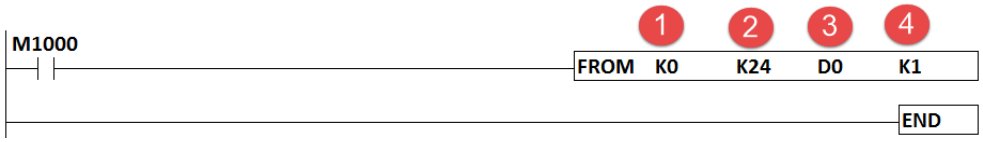
در ورودی های جریانی کارت آنالوگ به ازای -20mA – +20mA : -1000 – +1000

در خروجی های جریانی کارت آنالوگ به ازای 0 – 20mA : 0 – 4000

نکته : این مقادیر برای کارت آنالوگ 06XA-S می‌باشد و برای سایر کارتها مانند 04AD-SL یا 04DA-SL و ... به جدول معرفی کارتهای آنالوگ که در اول مبحث آنالوگ به آن اشاره شد مراجعه کنید.

مثال ۸۴ :

می‌خواهیم مقدار دمای لحظه‌ای کانال اول بر حسب درجه فارنهایت کارت 04PT-S را بخوانیم و این مقدار را درون رجیستر D0 مشاهده کنیم. فرض کنید تنظیمات کارت دما مطابق سنسورهای استفاده شده است.



مطابق برنامه فوق مقدار لحظه‌ای دمای کانال اول کارت 04PT-S برحسب درجه فارنهایت خوانده شده و درون رجیستر D0 ریخته می‌شود.

۱ – شماره کارت آنالوگ که در این مثال کارت 04PT اولین کارت آنالوگ بعد از CPU می‌باشد.

۲ – کنترل رجیستر شماره 24 که طبق جدول کنترل رجیسترها ، مربوط به مقدار دمای لحظه‌ای کانال اول برحسب درجه فارنهایت می‌باشد.

#24	H'407C	X	R	Present temp. of CH1	Present temperature of channels CH1 ~ CH4. (Unit: 0.1°F).
#25	H'407D	X	R	Present temp. of CH2	
#26	H'407E	X	R	Present temp. of CH3	
#27	H'407F	X	R	Present temp. of CH4	

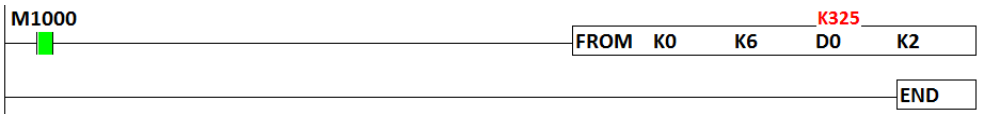
۳ – مقدار دمای خوانده شده توسط سنسور درون رجیستر D0 قرار می گیرد .

نکته : فرض کنید که مقدار دمای خوانده شده در رجیستر D0 برابر K785 باشد در اینصورت دمای واقعی برابر 78.5 درجه فارنهایت می‌باشد زیرا واحد محاسبه‌ی دما، 0.1 درجه می‌باشد.

۴ – این عملوند نیز بدین معناست که تنها دمای کانالی که کنترل رجیستر آن (K24) نوشته شده خوانده شود.

مثال ۸۵ :

می‌خواهیم مقدار میانگین دمای کانال اول و دوم کارت 04PT-S بر حسب درجه سانتیگراد را بخوانیم.



در برنامه بالا از کنترل رجیستر K6 استفاده شده است که طبق جدول کنترل رجیسترها به معنای خواندن مقدار میانگین دمای کانال اول برحسب درجه سانتیگراد می‌باشد.

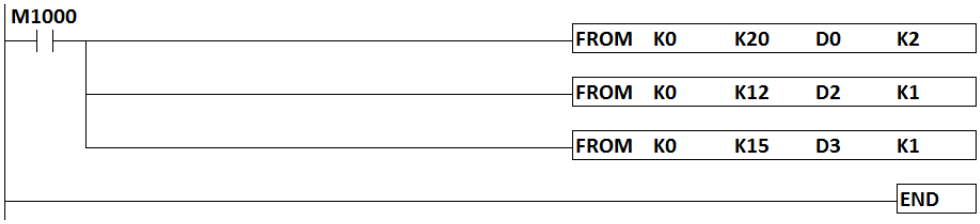
#6	H'406A	X	R	CH1 average degrees	Average degrees for channels CH1 ~ CH4. (Unit: 0.1°C).
#7	H'406B	X	R	CH2 average degrees	
#8	H'406C	X	R	CH3 average degrees	
#9	H'406D	X	R	CH4 average degrees	

مقدار دمای خوانده شده با دستور بالا عدد K325 می‌باشد در اینصورت مقدار واقعی دما برابر 32.5 درجه سانتیگراد می‌باشد .

آخرین عملوند نیز K2 بوده و بدین معناست که مقدار دمای کانال اول (کنترل رجیستر K6) و مقدار دمای کانال دوم(کنترل رجیستر K7) خوانده شده و به ترتیب درون رجیسترهای D0 و D1 ریخته شود برای دیدن مقدار دمای کانال دوم از طریق رجیستر D1 قابل دسترسی می‌باشد .

مثال ۸۶:

می‌خواهیم مقدار لحظه‌ای دمای کانال سوم و چهارم بر حسب درجه سانتیگراد و مقدار میانگین دمای کانال اول و چهار بر حسب درجه فارنهایت را بخوانیم.



در خط اول برنامه مقدار لحظه‌ای دمای کانال سوم بر حسب درجه سانتیگراد که طبق جدول کنترل رجیسترها برابر K20 می‌باشد، خوانده شده و درون رجیستر D0 ریخته می‌شود. همچنین به دلیل اینکه آخرین عملوند در خط اول برنامه برابر K2 قرار داده شده ، مقدار لحظه‌ای دمای کانال چهارم نیز بر حسب درجه سانتیگراد به صورت خودکار خوانده شده و درون رجیستر D1 ریخته می‌شود .

در خط دوم برنامه مقدار دمای میانگین کانال اول برحسب درجه فارنهایت خوانده شده و درون رجیستر D2 ریخته می‌شود.

در خط سوم نیز مقدار دمای میانگین کانال چهارم برحسب درجه فارنهایت خوانده شده و درون رجیستر D3 ریخته می‌شود.

نکته : برای خواندن مقادیر 32 بیتی از کارت‌های آنالوگ از دستورDFROM و برای نوشتن مقادیر 32 بیتی به کنترل رجیسترهای کارت آنالوگ از دستورDTO استفاده می‌کنیم .

نکته : چنانچه آخرین عملوند دستور FROM برابر K4 باشد می‌تواند همزمان مقادیر 4 کانال را بخواند و در رجیسترهای مورد نظر قرار دهد .

کنترل رجیسترهای کارت **04TC-S** :

CR#	Address	Save		Register content	Description
#0	H'4096	O	R	Model name	Set up by the system: DVP04TC-S model code=H'8B
#1	H'4097	O	R/W	Thermocouple type	b15~b12 | b11~b9 | b8~b6 | b5~b3 | b2~b0 Reserved | CH4 | CH3 | CH2 | CH1 Example: Setting of CH1 1. (b2, b1, b0) set to (0, 0, 0), use J-type. 2. (b2, b1, b0) set to (0, 0, 1), use K-type. 3. (b2, b1, b0) set to (0, 1, 0), use R-type. 4. (b2, b1, b0) set to (0, 1, 1), use S-type. 5. (b2, b1, b0) set to (1, 0, 0), use T-type.
CR#1: Used to set the working mode of four channels. There are 5 modes (J-type, K-type, R-type, S-type, and T-type) for each channel and can be set individually. For example, If you want to set CH1~CH4 as following: CH1: mode 0 (b2 ~ b0=000), CH2: mode 1 (b5 ~ b3=001), CH3: mode 0 (b8 ~ b6=000) and CH4: mode 1 (b11 ~ b9=001), you should set CR#1 to H'0208. The higher bits (b12 ~ b15) will be reserved and the default setting is H'0000.					
#2	H'4098	O	R/W	CH1 average number	Number piece of readings used for the calculation of "average" temperature on channels CH1 ~ CH4. Setting range: For versions prior to V3.04: K1 ~ K4,095. For versions after V3.05: K1 ~ K20. Default setting is K10.
#3	H'4099	O	R/W	CH2 average number	
#4	H'409A	O	R/W	CH3 average number	
#5	H'409B	O	R/W	CH4 average number	
CR#2 ~ CR#5: Please be noticed that when PLC sets average times via TO/DTO instructions, please use rising-edge/falling-edge detection instruction (such as LDP and LDF) to get correct average times.					
#6	H'409C	X	R	CH1 average degrees	Average degrees for channels CH1 ~ CH4. (Unit: 0.1°C).
#7	H'409D	X	R	CH2 average degrees	
#8	H'409E	X	R	CH3 average degrees	
#9	H'409F	X	R	CH4 average degrees	
#10	H'40A0	X	R	CH1 average degrees	Average degrees for channels CH1 ~ CH4. (Unit: 0.1°F).
#11	H'40A1	X	R	CH2 average degrees	
#12	H'40A2	X	R	CH3 average degrees	
#13	H'40A3	X	R	CH4 average degrees	
#14	H'40A4	X	R	Present temp. of CH1	Present temperature of channels CH1 ~ CH4. (Unit: 0.1°C).
#15	H'40A5	X	R	Present temp. of CH2	
#16	H'40A6	X	R	Present temp. of CH3	
#17	H'40A7	X	R	Present temp. of CH4	
#19	H'40A9	X	R	Present temp. of CH1	Present temperature of channels CH1 ~ CH2. (Unit: 0.1°F).
#20	H'40AA	X	R	Present temp. of CH2	
CR#	Address	Save		Register content	Description
#21	H'40AB	X	R	Present temp. of CH3	Present temperature of channels CH3 ~ CH4. (Unit: 0.1°F).
#22	H'40AC	X	R	Present temp. of CH4	
#24	H'40AE	O	R	CH1 OFFSET Value	Adjust offset value of channels CH1 ~ CH4. The range is -1,000 ~ +1,000 and default setting is K0. (Unit: 0.1°C).
#25	H'40AF	O	R	CH2 OFFSET Value	
#26	H'40B0	O	R	CH3 OFFSET Value	
#27	H'40B1	O	R	CH4 OFFSET Value	
#29	H'40B3	X	R/W	PID mode setting	Set H'5678 to enable PID mode, other set values are invalid. Default: H'0000.
#30	H'40B4	X	R	Error status	Data register stores the error status. Refer to the error code chart for details.
#31	H'40B5	O	R/W	Communication address setting	RS-485 communication address. Setting range is 1 ~ 254 and default setting is K1.

<table>
<tr><td>#32</td><td>H'40B6</td><td>O</td><td>R/W</td><td>Communication baud rate setting</td><td>Communication baud rate. For ASCII mode, date format is 7 bits, even, 1 stop bit (7, E, 1), while RTU mode, date format is 8 bits, even, 1 stop bit (8, E, 1).
b0: 4,800 bps (bit/sec).
b1: 9,600 bps (bit/sec). (default setting)
b2: 19,200 bps (bit/sec).
b3: 38,400 bps (bit/sec).
b4: 57600 bps (bit/sec).
b5: 115,200 bps (bit/sec).
b6 ~ b13: Reserved.
b14: switch between low bit and high bit of CRC code (RTU mode only).
b15: RTU mode.</td></tr>
<tr><td>#33</td><td>H'40B7</td><td>O</td><td>R/W</td><td>Reset to default setting</td><td><table><tr><td>b15~b12</td><td>b11~ b9</td><td>b8~b6</td><td>b5~b3</td><td>b2~b0</td></tr><tr><td>ERR LED</td><td>CH4</td><td>CH3</td><td>CH2</td><td>CH1</td></tr></table>Example: Setting of CH1
1. b0 ~ b1: Reserved.
2. b2: Set to 1 and PLC will be reset to default settings.
Definition of ERR LED: b12~b15=1111 (default settings)
1. b12 corresponds to CH1: when b12=1, scale exceeds the range, ERR LED flashes.
2. b13 corresponds to CH2: when b13=1, scale exceeds the range, ERR LED flashes.
3. b14 corresponds to CH3: when b14=1, scale exceeds the range, ERR LED flashes.
4. b15 corresponds to CH4: when b15=1, scale exceeds the range, ERR LED flashes.</td></tr>
<tr><td>#34</td><td>H'40B8</td><td>O</td><td>R</td><td>Software version</td><td>Display the software version in hexadecimal. Example: H'010A = version 1.0A</td></tr>
<tr><td colspan="4">#35 ~ #48</td><td>System used</td><td></td></tr>
</table>

تشریح کنترل رجیسترهای جدول بالا بصورت زیر می‌باشد:

K0 : این کنترل رجیستر بیانگر نوع و مدل کارت می‌باشد. برای مثال در صورتیکه این کنترل رجیستر توسط دستور FROM خوانده شود، کد این مدل (H8B) که یک کد به صورت هگز می‌باشد نمایش داده می‌شود.

K1 : این کنترل رجیستر، مهمترین کنترل رجیستر در بین ۳۴ کنترل رجیستر می‌باشد زیرا با استفاده ازاین کنترل رجیستر می‌توان نوع سنسور ورودی کارت را مشخص نمود.

K2 : خواندن دما به صورت لحظه‌ای و یا میانگین می‌باشد. در حالتی که از حالت میانگین استفاده کنیم، این کارت به صورت پیش فرض، ۱۰ بار از دمای محیط که توسط سنسور ارسال می‌شود نمونه گیری کرده و یک میانگین به PLC ارسال می‌کند تعداد نمونه گیری توسط این کنترل رجیستر برای کانال اول قابل تغییر می‌باشد که از K1 تا K20 (از یک بار نمونه گیری تا ۲۰ بار نمونه گیری) می‌توان تعداد نمونه گیری را تغییر داد.

K3 – K4 – K5 : این کنترل رجیسترها نیز مانند کنترل رجیستر قبلی عمل میکنند و مربوط به کانالها بعدی کارت می باشند.

K6 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال اول را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K7 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال دوم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K8 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال سوم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K9 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال چهارم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K10 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال اول را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K11 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال دوم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K12 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال سوم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K13 : با استفاده از این کنترل رجیستر می‌توان مقدار میانگین دمای کانال چهارم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K14 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال اول را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K15 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال دوم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K16 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال سوم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K17 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال چهارم را برحسب درجه سانتیگراد و دقت 0.1 محاسبه کرد.

K19 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال اول را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K20 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال دوم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K21 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال سوم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K22 : با استفاده از این کنترل رجیستر می‌توان مقدار لحظه‌ای دمای کانال چهارم را برحسب درجه فارنهایت و دقت 0.1 محاسبه کرد.

K24 : با استفاده از این کنترل رجیستر می‌توان مقدار OFFSET کانال اول را تنظیم کرد و سنسور را کالیبره نمود.

K25 – K26 – K27 : این کنترل رجیسترها نیز برای تنظیم مقدار OFFSET کانال‌های دوم و سوم و چهارم می‌باشند.

K29 : این کنترل رجیستر برای تنظیمات PID می‌باشد.

K30 : این کنترل رجیستر وضعیت خطاها را بررسی می‌کند.

K31 : این کنترل رجیستر برای تنظیم آدرس ارتباطی در شبکه MODBUS کاربرد دارد.

K32 : این کنترل رجیستر جهت تنظیم Baud rate در شبکه MODBUS کاربرد دارد.

K33 : این کنترل رجیستر، تنظیمات کارت 04TC را ریست می‌کند.

K34 : این کنترل رجیستر ورژن نرم افزار کارت 04TC را مشخص می‌کند.

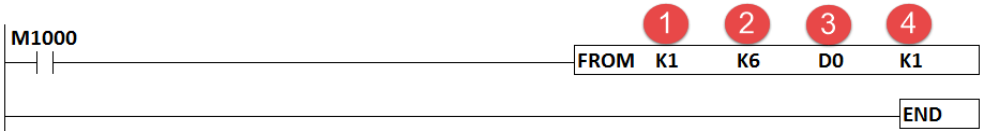
**در جدول کنترل رجیسترها علامت O نشان دهنده‌ی Latch بودن و علامت X نشان دهنده‌ی Latch نبودن مقادیر رجیسترهای داخلی کارت می‌باشد.

**علامت R نشان دهنده‌ی خواندنی بودن کنترل رجیستر و علامت W نشان دهنده‌ی نوشتنی بودن کنترل رجیستر است.

مثال ۸۷ :

می‌خواهیم مقدار دمای میانگین کانال اول کارت 04TC-S بر حسب درجه سانتیگراد را بخوانیم.

فرض کنید که کارت 04TC ، دومین کارت آنالوگ بعد از CPU می‌باشد.



۱ –فرض کنید کارت 04TC دومین کارت آنالوگ بعد از CPU باشد در این صورت آدرس آن K1 می‌شود.

۲ – کنترل رجیستر شماره K6 طبق جدول معادل مقدار میانگین دمای کانال اول برحسب درجه سانتیگراد است.

#6	H'409C	X	R	CH1 average degrees	Average degrees for channels CH1 ~ CH4. (Unit: 0.1°C).
#7	H'409D	X	R	CH2 average degrees	
#8	H'409E	X	R	CH3 average degrees	
#9	H'409F	X	R	CH4 average degrees	

۳ – مقدار دما در رجیستر D0 ریخته می‌شود. به عنوان مثال چنانچه مقدار این رجیستر K237 باشد دمای واقعی برابر 23.7 درجه سانتیگراد می‌باشد.

۴ – این عملوند نیز بدین معناست که تنها دمای کانالی که کنترل رجیستر آن (K6) نوشته شده خوانده شود.

روش تنظیم کنترل رجیسترهای کارت آنالوگ:

در بسیاری از پروژه ها مطابق نیاز باید تنظیمات کارت های آنالوگ و دما را تغییر دهیم به عنوان مثال برای اتصال سنسورهای مختلف دما به ورودی کارت دما و یا اتصال سنسورهای ولتاژی و یا جریانی به کارت ورودی آنالوگ، ابتدا باید کنترل رجیستر مربوطه را در کارت تنظیم کنیم . به دو روش می‌توان کنترل رجیسترهای کارت آنالوگ را تغییر داد یکی استفاده از دستور TO.

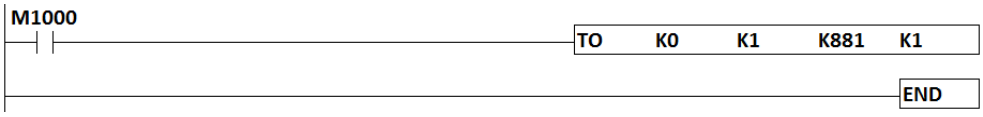
دستور TO :

این دستور به معنای نوشتن یا اعمال کردن مقداری به کنترل رجیستر کارت مورد نظر می‌باشد. زمانی از این دستور استفاده می‌کنیم که می‌خواهیم مقادیری را از CPU به کارت مورد نظر انتقال دهیم.

در مبحث آنالوگ نیاز است که تنظیمات کارت آنالوگ و سنسورها با یکدیگر از نظر نوع سنسور در (04PT – 04TC) و نوع ولتاژی یا جریانی بودن در (سنسور های فشار – فلو و...) تنظیم گردد.

برای مثال می‌خواهیم MODE SETTING در کارت آنالوگ (نوع سنسور از نظر جریانی یا ولتاژی) را مشخص کنیم.

مثال ۸۸ :



۱ - عملوند اول K0 همانند دستور FROM به معنای شماره کارت آنالوگ می‌باشد. (اولین کارت آنالوگ بعد از CPU)

۲ - عملوند دوم در این دستور مهمترین عملوند می‌باشد و با نام CR یا کنترل رجیستر شناخته می‌شود. در اینجا کنترل رجیستر شماره K1 قرار دارد که به معنای مشخص نمودن نوع سنسور از نظر مدل در کارت دما و یا ولتاژی یا جریانی بودن در کارت آنالوگ می‌باشد.

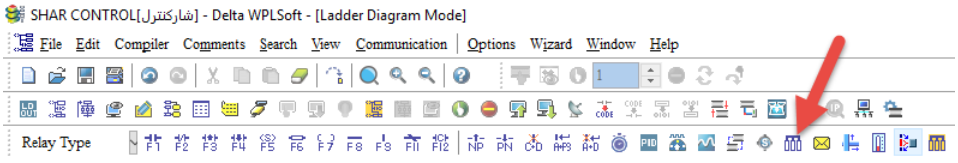
#1	H'4097	O	R/W	Thermocouple type	b15~b12	b11~b9	b8~b6	b5~b3	b2~b0
					Reserved	CH4	CH3	CH2	CH1
					Example: Setting of CH1 1. (b2, b1, b0) set to (0, 0, 0), use J-type. 2. (b2, b1, b0) set to (0, 0, 1), use K-type. 3. (b2, b1, b0) set to (0, 1, 0), use R-type. 4. (b2, b1, b0) set to (0, 1, 1), use S-type. 5. (b2, b1, b0) set to (1, 0, 0), use T-type.				

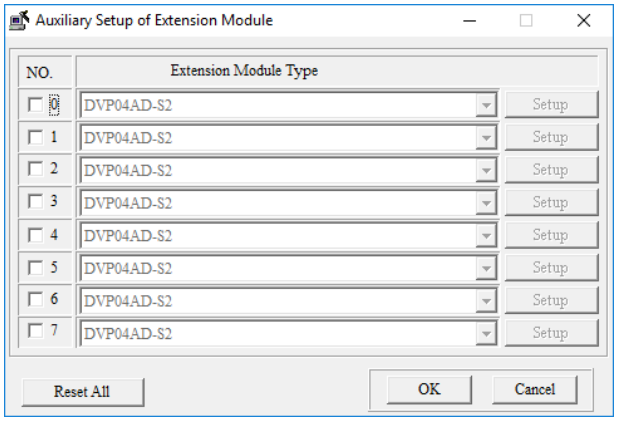
۳ - عملوند سوم، نوع ورودی یا خروجی را از نظر ولتاژی یا جریانی در آنالوگ و نوع سنسور در دما را مشخص می‌کند. در این قسمت مشاهده می‌کنید که از کد هگز H881 استفاده شده که در ادامه به توضیح آن می پردازیم.

۴ – عملوند چهارم در صورتیکه برابر K1 باشد، فقط یک کانال از کارت آنالوگ که در دستور TO به آن اشاره شده تنظیم می‌شود. حال فرض کنید برای مثال می خواهیم دو کانال کارت دمای 04TC-S را با یک دستور TO تنظیم کنیم کافیست عملوند چهارم را برابر K2 قرار دهیم. به همین ترتیب تا 4 کانال دمای 04TC-S را می‌توانیم با یک دستور TO تنظیم کنیم. در این صورت تنظیمات کانالها مشابه خواهند شد .

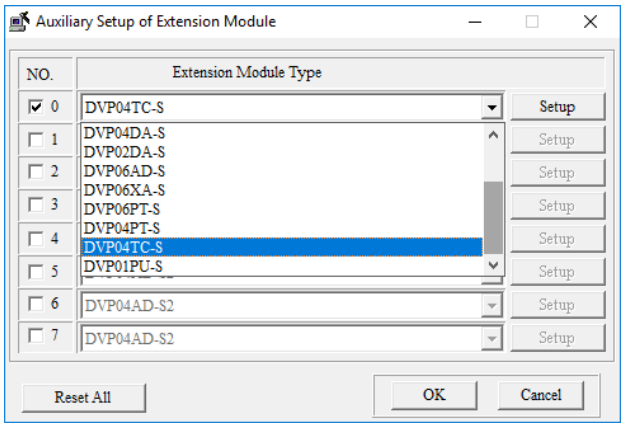
پیکربندی کارتهای دما :

در صورتیکه بخواهیم از انواع مختلف سنسورهای آنالوگ از نظر ولتاژی یا جریانی بودن استفاده کنیم، یا بخواهیم از انواع مختلف سنسورها از نظر نوع (دما) استفاده کنیم، نیاز داریم تا هرکدام از کانال‌های ورودی و خروجی را متناسب با نوع سنسور تنظیم کنیم. برای این کار کافیست تا مانند دستور زیر عمل کنیم: ابتدا نرم افزار WPLsoft را باز کرده مطابق شکل زیر روی **Auxiliary Setup Extention** کلیک می‌کنیم **Module** .



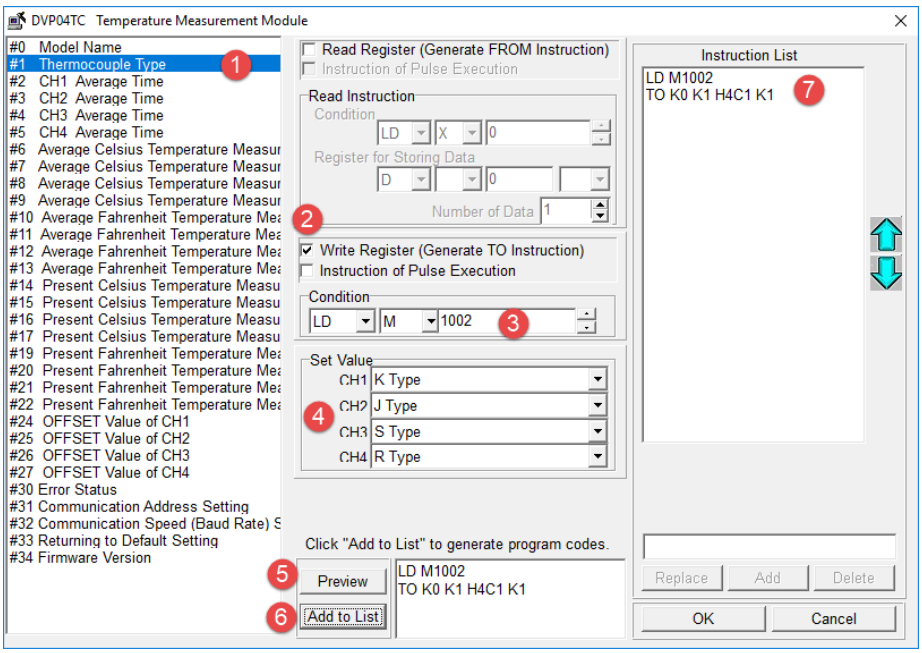


در منوی باز شده باید شماره کارت آنالوگی که می‌خواهیم تنظیماتش را انجام دهیم، انتخاب کرده و از منوی بازشوی مقابل شماره ، نوع کارت آنالوگ یا دما را مطابق شکل بعد انتخاب کنیم.



پس از انتخاب کارت دما، از منوی آبشاری گزینه Setup را انتخاب و وارد تنظیمات کارت می شویم.

در پنجره تنظیمات باز شده به ترتیب زیر عمل کنید:



توضیحات شکل صفحه قبل بصورت زیر می باشد :

۱ – ابتدا کنترل رجیستر شماره 1 را که مربوط به تنظیمات نوع سنسور ورودی کارت می‌باشد، انتخاب کنید.

۲ – سپس تیک گزینه‌ی Write Register (Generate TO instruction) را بزنید.

۳ – می‌توانیم برای برای فعال کردن دستور TO از فلگ خاص M1002 استفاده کنیم. این فلگ بعد از run شدن CPU به مدت یک سیکل فعال بوده و سپس غیر فعال می‌شود.

۴ – در این قسمت می‌توانید نوع سنسور ترموکوپل مورد نظر خود را برای هر کانال تنظیم کنید.

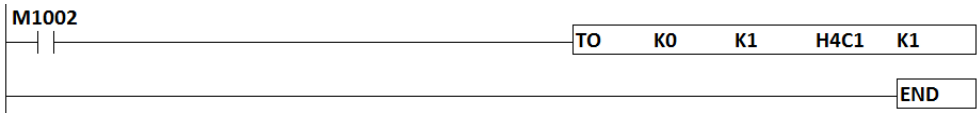
۵ – سپس با زدن دکمه Preview دستور مورد نظر شما به صورت پیش نمایش ظاهر می‌شود.

۶ – حال با زدن دکمه Add to list این برنامه در به کادر سمت راست منتقل می‌شود.

۷ – برنامه تولید شده در قسمت Instruction list نمایش داده می‌شود.

۸ – با زدن دکمه OK دستورات به محیط برنامه نویسی منتقل می‌شود.

همانطور که در تصویر زیر مشاهده می‌کنید برنامه تنظیم کارت مورد نظر شما به صفحه‌ی برنامه نویسی منتقل شده است.



طبق برنامه فوق با روشن شدن PLC در اولین سیکل برنامه، تنظیمات مورد نظر به کارت دما ی 04TC ارسال می‌شود.

نکته : در کارت دمای 04PT می‌توان مشابه روش کارت 04TC و از طریق WIZARD تنظیمات مربوط به سنسورها را تنظیم کرد .

محاسبه‌ی کد هگز مقادیر تنظیمات در کارت 04TC-S به صورت دستی :

کد هگزی که در برنامه صفحه قبل مشاهده می‌کنید (H4C1) معرف نوع سنسورهای تعریف شده برای کارت 04TC می‌باشد. حال می‌خواهیم بدون استفاده از قسمت Wizard و به صورت دستی این کد را محاسبه کنیم. برای این کار ابتدا باید فایل راهنمای کارت 04TC-S را باز کرده و به قسمت جدول کنترل رجیسترها مراجعه کنیم. در این جدول باید به کنترل رجیستر شماره 1 (K1) و توضیحات آن توجه کنیم.

#1	H'4097	O	R/W	Thermocouple type	b15~b12	b11~b9	b8~b6	b5~b3	b2~b0
					Reserved	CH4	CH3	CH2	CH1
					Example: Setting of CH1 1. (b2, b1, b0) set to (0, 0, 0), use J-type. 2. (b2, b1, b0) set to (0, 0, 1), use K-type. 3. (b2, b1, b0) set to (0, 1, 0), use R-type. 4. (b2, b1, b0) set to (0, 1, 1), use S-type. 5. (b2, b1, b0) set to (1, 0, 0), use T-type.				

طبق این فایل راهنما به هر یک از کانالهای کارت 04TC-S سه بیت اختصاص داده شده است.

کانال اول			کانال دوم			کانال سوم			کانال چهارم		
بیت۱	بیت۲	بیت۳	بیت۴	بیت۵	بیت۶	بیت۷	بیت۸	بیت۹	بیت۱۰	بیت۱۱	بیت۱۲

نکته : بیت‌های ۱۳ ، ۱۴ و ۱۵ رزرو می‌باشد. (این کنترل رجیستر ۱۶ بیتی می‌باشد)

جهت تنظیم کانالها برای سنسور نوع J : بیت‌های کانال باید به صورت 0 0 0 باشد.

جهت تنظیم کانالها برای سنسور نوع K : بیت‌های کانال باید به صورت 0 0 1 باشد.

جهت تنظیم کانالها برای سنسور نوع R : بیت‌های کانال باید به صورت 0 1 0 باشد.

جهت تنظیم کانالها برای سنسور نوع S : بیت‌های کانال باید به صورت 0 1 1 باشد.

جهت تنظیم کانالها برای سنسور نوع T : بیت‌های کانال باید به صورت 1 0 0 باشد.

حال مثال قبل را در نظر بگیرید که کانال ها را به صورت K – J – S – R تنظیم کردیم.

تنظیم بیتهای هر کانال بصورت مطابق شکل زیر می‌باشد :

نوع K			نوع J			نوع S			نوع R		
1	0	0	0	0	0	1	1	0	1	0	0

حال باید این بیت ها را به صورت چهارتایی از راست به چپ جدا کنیم که به صورت زیر خواهد شد:

0100 1100 0001

برای تبدیل مبنای این اعداد باینری به مبنای هگز، از جدول زیر استفاده کنید .

مبنای ۱۶	مبنای ۲	مبنای ۱۰
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

برای تبدیل عدد در مبنای 2 به عدد در مبنای 16 می‌توان از روش زیر استفاده نمود .

طبق جدول و فرمول فوق داریم :

8421	8421	8421
0100	1100	0001
4	12	1

4 = 0100

12 = 1100

1 = 0001

نکته : عدد 12 در مبنای هگز معادل **C** می‌شود .

بنابراین کد هگز مورد نظر برابر H4C1 می‌شود.(حرف H به معنای مبنای Hex یا 16 می‌باشد)

تنظیمات سنسور کارت 04PT-S با استفاده از wizard :

برای تنظیم کانالهای کارت 04PT-S توسط wizard مطابق مراحل گفته شده برای کارت 04TC عمل می کنیم .

تنظیمات کارت دمای 04PT-S به روش دستی :

تنظیم کانالهای این کارت را به صورت دستی مطابق مثال زیر انجام دهیم.

مثال ۸۹ :

از دستور TO جهت تنظیم کردن کارت 04PT-S استفاده می کنیم .

۱ – شماره کارت آنالوگ. (در اینجا کارت 04PT-S دومین کارت آنالوگ می‌باشد).

۲ – کنترل رجیستر شماره 1 مربوط به تنظیمات سنسور ورودی کارت

۳ – کد هگز مربوط به تنظیمات نوع سنسورها

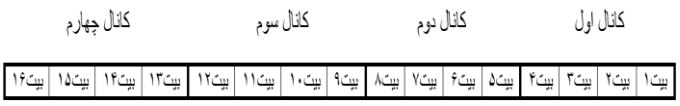
۴ – در صورتیکه این عملوند برابر عدد 1 باشد ، فقط یک کانال کارت تنظیم می‌شود.

محاسبه‌ی کد هگز تنظیمات سنسورهای ورودی کارت 04PT-S به صورت دستی :

برای این کار ابتدا باید فایل راهنمای کارت 04PT-S را باز کرده و به قسمت جدول کنترل رجیسترها مراجعه کنیم. در این قسمت باید به کنترل رجیستر شماره 1 که تعیین کننده نوع سنسورها می‌باشد رجوع کنید.

#1	H'4065	O	R/W	Mode setting	CH1 mode: b0 ~ b3 CH2 mode: b4 ~ b7 CH3 mode: b8 ~ b11 CH4 mode: b12 ~ b15 Take CH1 mode (b3,b2,b1,b0) for example. The default value is H'0000. 1. (0,0,0,0): PT100 2. (0,0,0,1): NI100 3. (0,0,1,0): PT1000 4. (0,0,1,1): NI1000 5. (1,1,1,1): The channel is disabled.

طبق این جدول به ازای هر کانال 4 بیت اختصاص داده شده است که مطابق صفحه بعد می‌باشد :



در صورتیکه بخواهیم کانال مورد نظر برای سنسور PT100 تنظیم شود بیتها باید به صورت 0 0 0 0 تنظیم شود.

در صورتیکه بخواهیم کانال مورد نظر برای سنسور NI100 تنظیم شود بیتها باید به صورت 1 0 0 0 تنظیم شود.

در صورتیکه بخواهیم کانال مورد نظر برای سنسور PT1000 تنظیم شود بیتها باید به صورت 0 1 0 0 تنظیم شود.

در صورتیکه بخواهیم کانال مورد نظر برای سنسور NI1000 تنظیم شود بیتها باید به صورت 1 1 0 0 تنظیم شود.

مثال ۹۰ :

می‌خواهیم کانال های کارت 04PT-S به صورت زیر تنظیم شود.

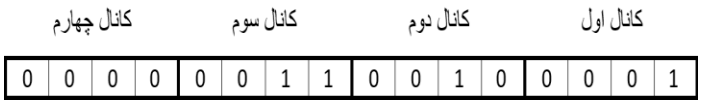
کانال اول : NI100

کانال دوم : PT1000

کانال سوم : NI1000

کانال چهارم : PT100

ابتدا جدولی مطابق زیر ترسیم کرده و بیت های مربوط به هر کانال را مطابق سنسورهای مربوطه تنظیم می کنیم .



8421	8421	8421	8421
0000	0011	0010	0001
0	3	2	1

حال باید طبق جدول یا فرمولی که برای کارت 04TC-S ارائه شد، اعداد مبنای ۲ را به مبنای ۱۶ تبدیل کرد. کد هگز مربوط به تنظیمات کارت 04PT-S به صورت H0321 می‌باشد که می‌توان آنرا بصورت H321 نوشت .

پیکربندی کارتهای آنالوگ :

مطابق مطالب گفته شده برای کارتهای دما می‌توان ازwizard برای تنظیمات کنترل رجیسترهای کارتهای آنالوگ مانند 04AD - 04DA و 06XA استفاده کرد .

جهت تنظیم کنترل رجیسترهای کارت 06XA به صورت دستی مطابق روش زیر عمل می‌کنیم.

برای این کار باید فایل راهنمای کارت مورد نظر را باز کرده و وارد بخش جدول کنترل رجیسترها شویم.

#1	H'40C9	○	R/W	Input mode setting	Input mode setting: (CH1 ~ CH4) Mode 0: input voltage mode (-10V ~ +10V). Mode 1: input voltage mode (-6V ~ +10V). Mode 2: input current mode (-12mA ~ +20mA). Mode 3: input current mode (-20mA ~ +20mA). Mode 4: none use. Output mode setting: (CH5 ~ CH6) Mode 0: output voltage mode (0V ~ 10V). Mode 1: output voltage mode (2V ~ 10V). Mode 2: output current mode (4mA ~ 20mA). Mode 3: output current mode (0mA ~ 20mA).

طبق فایل راهنما به هر کانال ورودی در این کارت ، 3 بیت و به هر کانال خروجی این کارت 2 بیت اختصاص داده شده است.

کانال اول		کانال دوم		کانال سوم		کانال چهارم		کانال اول		کانال دوم	
بیت۱	بیت۲	بیت۳	بیت۴	بیت۵	بیت۶	بیت۷	بیت۸	بیت۹	بیت۱۰	بیت۱۱	بیت۱۲	بیت۱۳	بیت۱۴	بیت۱۵	بیت۱۶
ورودی		ورودی		ورودی		ورودی		خروجی		خروجی	

چنانچه بیتهای کانال ورودی به صورت 0 0 0 باشد، کانال مطابق 10V+ تا 10V- خواهد بود.

چنانچه بیتهای کانال ورودی به صورت 1 0 0 باشد، کانال مطابق 10V+ تا 6V- خواهد بود.

چنانچه بیتهای کانال ورودی به صورت 0 1 0 باشد، کانال مطابق 20mA+ تا 12mA- خواهد بود.

چنانچه بیتهای کانال ورودی به صورت 1 1 0 باشد، کانال مطابق 20mA+ تا 20mA- خواهد بود.

چنانچه بیتهای کانال خروجی به صورت 0 0 باشد، کانال مطابق 10V+ تا 0V خواهد بود.

چنانچه بیتهای کانال خروجی به صورت 1 0 باشد، کانال مطابق 10V+ تا 2V ولت خواهد بود.

چنانچه بیتهای کانال خروجی به صورت 0 1 باشد، کانال مطابق 20mA+ تا 4mA خواهد بود.

چنانچه بیتهای کانال خروجی به صورت 1 1 باشد، کانال مطابق 20mA+ تا 0mA خواهد بود.

مثال ۹۱ :

می‌خواهیم کانال‌های کارت 06XA را به صورت زیر تنظیم کنیم :

کانال اول : 10V+ تا 10V-

کانال دوم : 20mA+ تا 12mA-

کانال سوم : 20mA+ تا 20mA-

کانال چهارم : 20mA+ تا 20mA-

کانال پنجم : 10V+ تا 0

کانال ششم : 20mA+ تا 0

کانال ششم		کانال پنجم		کانال چهارم			کانال سوم			کانال دوم			کانال اول		
1	1	0	0	0	1	1	0	1	1	0	1	0	0	0	0

حال باید بیتها را بصورت چهارتایی از راست به چپ جدا کنیم و سپس با استفاده از جدول تبدیل مبناها یا استفاده از فرمول گفته شده ، کد هگز مربوطه را بدست می آوریم.

8421	8421	8421	8421
1100	0110	1101	0000
12	6	13	0

طبق جدول تبدیل مبناها، عدد C = 12 و D = 13 می‌باشد لذا کد هگز بدست آمده برابر HC6D0 می‌باشد.(H به معنای مبنای هگز می‌باشد)

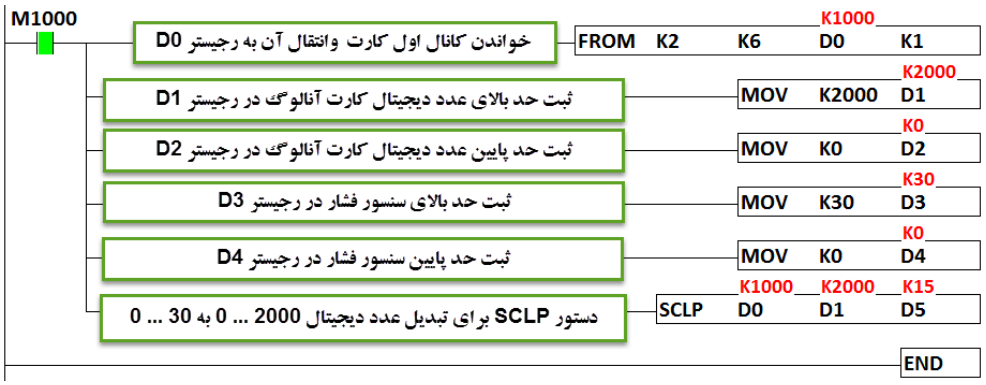
دستور اسکیل SCLP :

از انجایی که مقدار ورودی آنالوگ به عدد دیجیتالی تبدیل می‌شود ممکن است فهم اعداد دیجیتالی در برنامه مشکل باشد لذا می‌توانیم با استفاده از دستورات خاصی عدد دیجیتال شده را به اعداد واقعی کمیتهای فیزیکی مانند فشار ، رطوبت و ... تبدیل کنیم . برای درک این مطلب به مثال زیر توجه کنید

مثال ۹۲ :

فرض کنید توسط یک سنسور فشار نصب شده بر روی یک کلکتور آب، می‌خواهیم مقدار فشار آب درون لوله را اندازه گیری کنیم. سنسور فشاری که انتخاب کرده‌ایم از نوع ولتاژی و حداکثر فشار 30 Bar می‌باشد این سنسور به ازای 30Bar – 0 ، ولتاژی به اندازه‌ی 10V – 0 تولید می‌کند. کارت آنالوگ 06XA محدوده ولتاژ ورودی 10V.. 0 را به عدد دیجیتال شدخ 2000 ... 0 تبدیل می کند .

عدد دیجیتال 1000 در واقع معرف فشاری برابر15Bar در لوله می‌باشد. عدد دیجیتال 1000 از نگاه اپراتور نامفهوم است و باید کاری کنیم که اپراتور بجای عدد 2000 – 0 ، عدد واقعی 30Bar – 0 را مشاهده کند . همانطور که در برنامه زیر ملاحظه می‌کنید مقدار حقیقی k15 (فشار آب) درون رجیستر D5 ریخته شده است.لذا برنامه نویس می‌تواند از این رجیستر در برنامه استفاده کند .

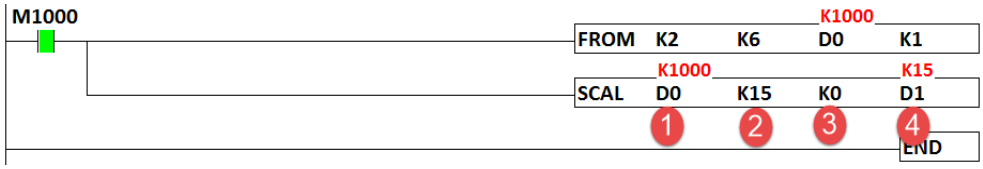


دستور SCAL :

این دستور نیز مانند دستور SCLP عمل می‌کند. مثال زیر را ببینید .

مثال ۹۳ :

در این مثال بجای استفاده از دستور SCLP از دستور SCAL استفاده کرده‌ایم تا مقدار دیجیتال شده کمیت فشار را به مقدار حقیقی آن تبدیل کنیم به عبارتی محدوده عدد دیجیتال 2000 ... 0 را به مقدار واقعی 30bar ... 0 تبدیل کنیم. برنامه زیر عملکرد این دستور را نشان می‌دهد .



۱ –مقدار دیجیتال شده‌ی فشار آب در این رجیستر ثبت می‌شود.

۲ – شیب نمودار

$$\frac{\left[\text{حداقل مقدار حساسیت سنسور (در اینجا 0Bar) - حداکثر مقدار حساسیت سنسور (در اینجا 30Bar)}\right] \times 1000}{\left[\text{حداقل مقدار عددی دیجیتال (در اینجا 0) - حداکثر مقدار عددی دیجیتال (در اینجا 2000)}\right]}$$

محاسبه شیب نمودار :

$$\frac{30-0}{2000-0} \times 1000 = 15$$

۳ – تعیین مقدار آفست (در این مثال آفست برابر صفر می‌باشد)

$$\frac{\text{شیب نمودار} \times \text{حداقل مقدار عددی دیجیتال (در اینجا 0)} - \text{حداقل مقدار حساسیت سنسور (در اینجا 0Bar)}}{1000}$$

$$\frac{0-0\times 0}{1000} = 0$$

۴ – مقدار اسکیل شده در این رجیستر قرار می‌گیرد .

مثال ۹۴ :

فرض کنید سنسور رطوبتی داریم که به ازای 100% – 0 رطوبت محیط ، جریانی معادل 4 – 20mA ارسال می‌کند. این سنسور را به ورودی کارت DVP06XA-S متصل کرده‌ایم. می‌خواهیم مقدار حقیقی رطوبت را در رجیستری مانند D5 نمایش دهیم.

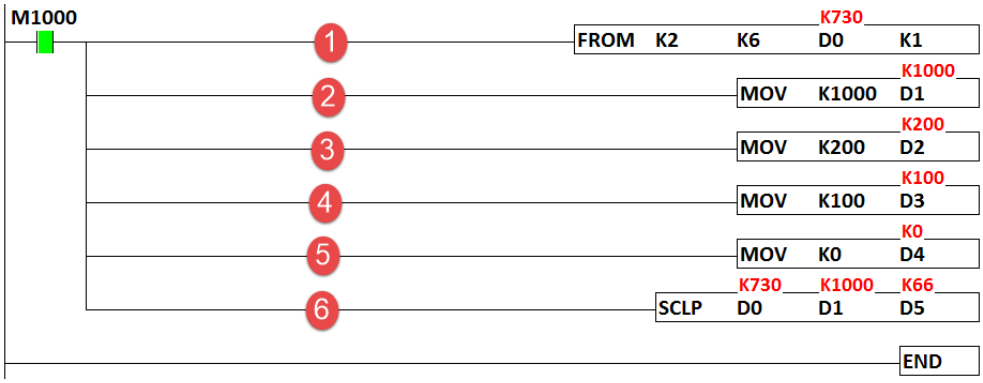
نکته مهم

در کارت DVP06XA-S ، در صورت استفاده از سنسور های جریانی 4 – 20mA ، برای انجام عملیات اسکیل کردن ، حد پایین عدد دیجیتال، برابر K200 می‌باشد زیرا طبق جدول مشخصات کارتهای آنالوگ، این کارت در حالت جریانی مقدار $-20mA \sim +20mA$ در ورودی آنالوگ را به عددی دیجیتال شده برابر $-1000 \sim +1000$ تبدیل می کند. با یک تناسب ساده می‌توان مقدار عدد دیجیتال را برای سنسور 4 – 20mA را بدست آورد.

40	2000
4	x

$$40 * x = 2000 * 4 \rightarrow x = \frac{8000}{40} = 200$$

نکته : چون حد پایین جریان سنسور برابر 4mA می‌باشد لذا حد پایین عدد دیجیتال برای استفاده در دستور SCLP و یا SCAL برابر K200 می‌باشد.



۱ – دستور خواندن مقدار کانال اول کارت 06XA و ثبت مقدار دیجیتال درون دیتا رجیستر D0

۲ – ثبت حد بالای دقت عدد دیجیتالی که توسط کارت خوانده می‌شود درون دیتا رجیستر D1

۳ - ثبت حد پایین دقت عدد دیجیتالی که توسط کارت خوانده می‌شود درون دیتا رجیستر D2

نکته : طبق محاسبات بالا حد پایین عدد دیجیتالی که توسط کارت آنالوگ تولید می‌شود برابر K200 می‌باشد .

۴ - ثبت حد بالای سنسور رطوبت (100%) درون دیتا رجیستر D3

۵ - انتقال حد پایین سنسور رطوبت (0%) درون دیتا رجیستر D4

۶ – استفاده از دستور SCLP برای انجام عملیات اسکیل. (جواب نهایی در دیتا رجیستر D5)

مثال ۹۵ :

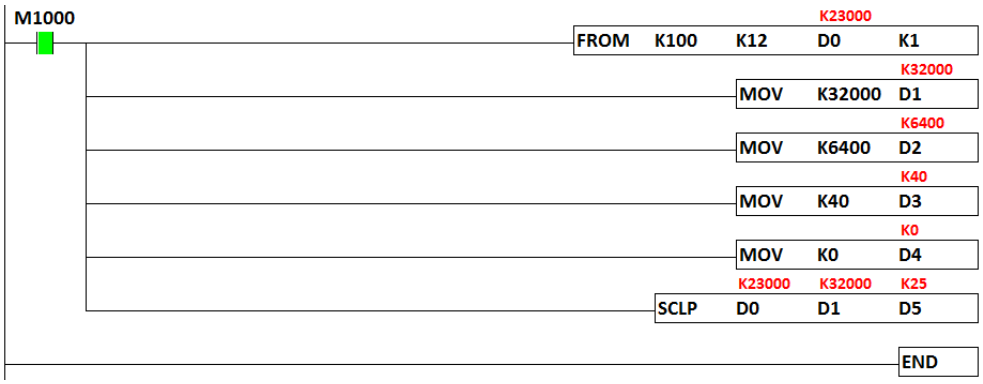
سنسور فشاری داریم که به ازای فشاری معادل 0 – 40bar ، جریانی معادل 4 – 20mA به کارت آنالوگ DVP04AD-SL ارسال می‌کند. برنامه مطلوب جهت نشان دادن فشار حقیقی را بنویسید.

طبق جدول مشخصات فنی کارتهای آنالوگ، کارت آنالوگ DVP04AD-SL دارای دقت بالایی(۱۶ بیت) بوده و ورودی آنالوگ $-20mA \sim +20mA$ را به عدد دیجیتالی برابر $-32000 \sim +32000$ تبدیل می کند. لذا می‌توانیم با یک تناسب ساده دقت کارت را به ازای $4mA \sim 20mA$ بدست بیاوریم.

نکته : کارتهای آنالوگ با دقت 16 بیت ، از سمت چپ به PLC دلتا متصل می شوند و آدرس پایه آنها از k100 الی k107 می‌باشد .

40	64000
4	X

$$40 \times X = 64000 * 4 \rightarrow X = \frac{256000}{40} = 6400$$



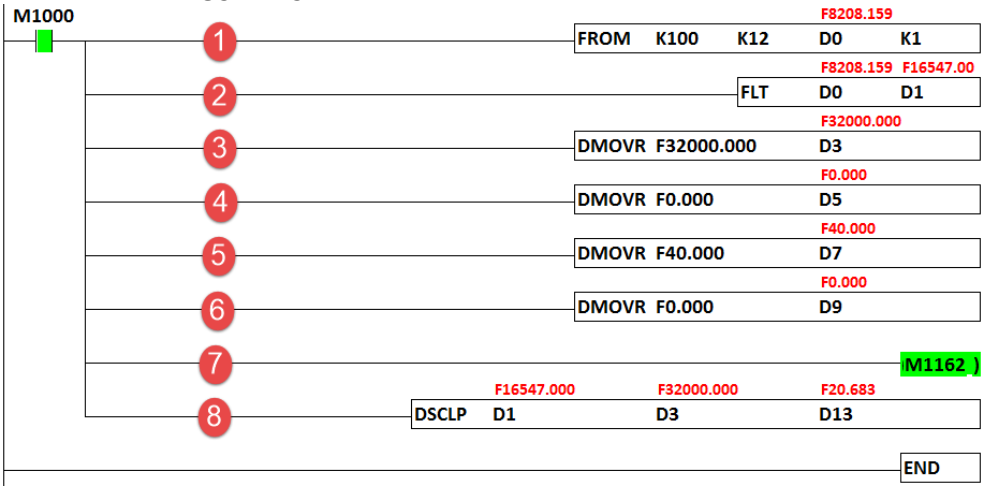
در این مثال مقدار فشار واقعی برابر 25bar بوده و در دیتا رجیستر D5 ثبت شده است.

توضیحات برنامه : در خط اول K100 به معنای اولین کارت آنالوگ وصل شده به PLC از سمت چپ بوده و K12 شماره کنترل رجیستر مربوط به کانال اول می‌باشد. (در این مثال فرض کنید سنسور به کانال اول متصل است)

در مثال‌های قبل مقدار حقیقی فشار که درون دیتا رجیستر D5 ریخته می‌شدند شامل اعداد صحیح بودند. به عبارتی فاقد رقم اعشاری بودند. حال فرض کنید مقدار دقیقتری از فشار یا رطوبت را نیاز دارید. در این صورت باید ابتدا دیتا رجیستری را که قرار است مقدار حقیقی فشار یا رطوبت در آن ریخته شود را از عدد صحیح به عدد اعشاری تبدیل کنید. بدین منظور می‌توانید از دستور FLT و یا DFLT که قبلا توضیح داده شده استفاده کنید.

مثال ۹۶ :

مثال قبل را در نظر بگیرید. می‌خواهیم مقدار دقیق فشار را بصورت عدد اعشاری را محاسبه کنیم.



۱ – دستور خواندن مقدار آنالوگ و ثبت آن در دیتا رجیستر D0

۲ – تبدیل فرمت دیتا رجیستر D0 از عدد صحیح به عدد اعشاری (D1 و D2) اشغال می‌شوند.

۳-ثبت حد بالای عدد دیجیتال کارت آنالوگ با دستور انتقال اعشاری به دیتا رجیستر D5

۴ - ثبت حد پایین عدد دیجیتال کارت آنالوگ با دستور انتقال اعشاری به دیتا رجیستر D7

۵ - ثبت حد بالای مقدار نامی سنسور با دستور انتقال اعشاری به دیتا رجیستر D9

۶ - ثبت حد پایین مقدار نامی سنسور با دستور انتقال اعشاری به دیتا رجیستر D11

۷ – جهت افزایش دقت تبدیل مقادیر دسیمال به اعشاری فلگ M1162 را فعال کنید.

۸- استفاده از دستور DSCLP جهت اسکیل کردن مقادیر و ثبت جواب نهایی در D13

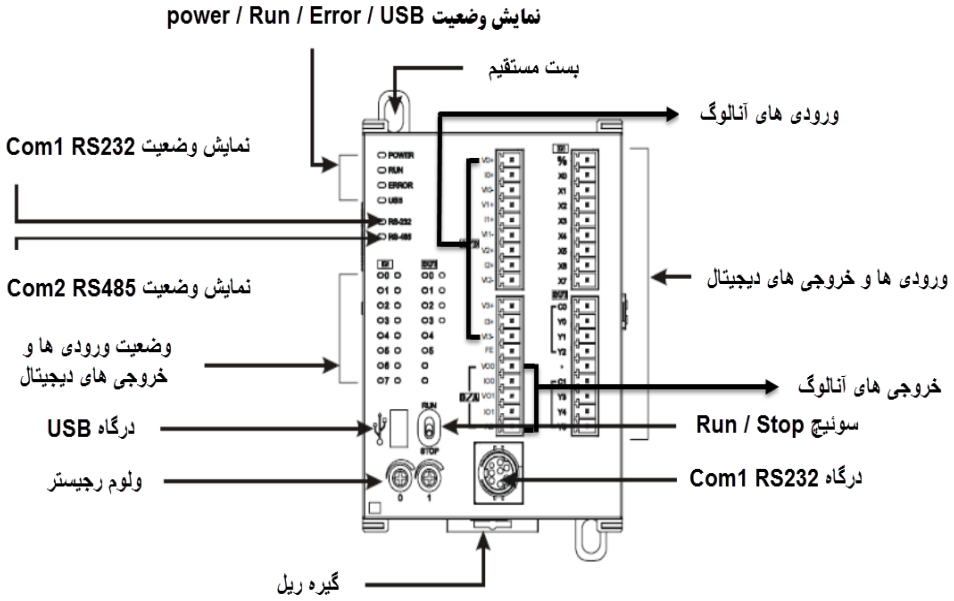
نکته : اولین کانال ورودی کارت آنالوگ سری SL که از سمت چپ به PLC متصل می‌شود برابر K100 می‌باشد.

نکته : جهت دیدن مقادیر بصورت اعشاری از منوی View و سپس Monitoring data format گزینه Float را انتخاب کنید .

PLC های دارای ورودی خروجی آنالوگ :

بعضی از مدلهای PLC دلتا، دارای ورودی خروجی دیجیتال می‌باشند نام این مدلها بصورت زیر می‌باشد

DVP-30EX2 & DVP-20EX2 & DVP-20SX2 & DVP-10SX



با استفاده از این PLC ها می‌توان به راحتی و با استفاده از رجیسترهای خاص ، مقدار ورودی آنالوگ در هر کانال را خواند به عبارت دیگر در این مدلها نیازی به استفاده از دستور FROM و یا TO نمی‌باشد .

در جدول زیر مقادیر دیجیتال مخصوص به هر کانال آنالوگ ورودی به همراه مدل‌های آن آورده شده است .

مشخصات سیگنالهای ورودی آنالوگ و مقادیر دیجیتال شده PLC های دارای ورودی

Mode \ Model		20EX2/SX2	30EX2
Voltage	-10 V~+10 V	-2000~+2000	-32000~+32000
	-5 V~+5 V	Not support	-32000~+32000
	+1 V~+5 V	Not support	+0~+32000
Current	-20 mA~+20 mA	-2000~+2000	-32000~+32000
	+4 mA~+20 mA	+0~+2000	+0~+32000
Temperature	PT100/PT1000 -180 ℃~+800 ℃	Not support	-1800~+8000
	NI100/NI1000 -80 ℃ ~ +170 ℃	Not support	-800~+1700

مشخصات سیگنالهای خروجی آنالوگ و مقادیر دیجیتال شده			
Mode / Model		20EX2/SX2	30EX2
Voltage	-10 V~+10 V	-2000~+2000	-32000~+32000
Current	+0 mA~+20 mA	+0~+4000	+0~+32000
	+4 mA~+20 mA	+0~+4000	+0~+32000

جدول زیر رجیسترهای مخصوص کانالهای ورودی خروجی آنالوگ را نشان می دهد .

رجیستر خاص	توضیحات	Sx2 ex2	Off ↓ On	Stop ↓ Run	Run ↓ stop	Attrib	Latched	Default
D1110	مقدار میانگین ورودی آنالوگ ازکانال0 (AD0) در PLCهای EX2/SX2 در این ذخیره میشود اگر تعداد مقادیری که از آنها میانگین گرفته میشود در رجیستر D1062 (average time) برابر ۱ تنظیم شود، d1110 مقدار لحظه ای را نشان میدهد	O	0	-	-	R	NO	0
D1111	مقدار میانگین ورودی آنالوگ ازکانال1 (AD1) در PLCهای EX2/SX2 در این ذخیره میشود اگر تعداد مقادیری که از آنها میانگین گرفته میشود در رجیستر D1062 (average time) برابر ۱ تنظیم شود، d1111 مقدار لحظه ای را نشان میدهد	O	0	-	-	R	NO	0
D1112	مقدار میانگین ورودی آنالوگ ازکانال2 (AD2) در PLCهای EX2/SX2 در این ذخیره میشود اگر تعداد مقادیری که از آنها میانگین گرفته میشود در رجیستر D1062 (average time) برابر ۱ تنظیم شود، d1112 مقدار لحظه ای را نشان میدهد	O	0	-	-	R	NO	0
D1113	مقدار میانگین ورودی آنالوگ ازکانال3 (AD3) در PLCهای EX2/SX2 در این ذخیره میشود اگر تعداد مقادیری که از آنها میانگین گرفته میشود در رجیستر D1062 (average time) برابر ۱ تنظیم شود، d1113 مقدار لحظه ای را نشان میدهد نمایش وضعیت کانالهای آنالوگ ورودی 30SX2	O	0	-	-	R	NO	0

ادامه کنترل رجیسترهای مخصوص ورودی خروجی های آنالوگ در PLC های 10SX , 20SX ,EX

D1114	فعال یا غیرفعال شدن کانالهای AD 20EX2/SX2. NOT 30EX2 (SUPPORTED) 0 : فعال (پیشفرض) 1 : غیرفعال بیت های 0 تا 3 مربوط به AD0 تا AD3	O	0	-	-	R/W	YES	0
D1115	تنظیمات ورودی و خروجیهای آنالوگ 20EX2/SX2	O	0	0	0	R/W	YES	0
	تنظیمات ورودی و خروجی های آنالوگ 30EX2	O	-	-	-	R/W	YES	H'FFFF
D1062	تعیین تعداد سیگنال آنالوگی که باید از آن میانگین گرفته شود. در کانال های CH0 – CH3 در PLCهای مدل 20EX2/SX2 به صورت پیشفرض بر روی K20 تنظیم است که میتواند از 20 – 1 باشد در کانالهای CH0 – CH2 در PLCهای مدل 30EX2 به صورت پیشفرض روی K20 تنظیم است که میتواند از 15 – 1 باشد.	O	2	-	-	R/W	YES	2
D1116	مقدار خروجی در کانال آنالوگ خروجی 0 (AD0) در PLCهای مدل EX2/SX2 در این رجیستر نشان داده میشود.	O	0	0	0	R/W	NO	0
D1117	مقدار خروجی در کانال آنالوگ خروجی 1 (AD1) در PLCهای مدل 20EX2/SX2 در این رجیستر نشان داده میشود. (30EX2 NOT SUPPORTED)	O	0	0	0	R/W	NO	0
D1118	تعیین زمان نمونه برداری در کانالهای A/D پیشفرض بر روی 2 و واحد میلی ثانیه تنظیم است	O	2	-	-	R/W	YES	2

برای مثال در PLC سری DVP-20SX2 تنظیمات کانالها به شکل زیر می‌باشد. برای سایر مدلها باید به فایل راهنمای همان مدل مراجعه کنید.

برای تنظیم ولتاژی و یا جریانی بودن کانالهای ورودی و خروجی مطابق جدول زیر عمل می کنیم .

دو کانال خروجی DA		چهار کانال ورودی AD				
Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	0	0	0	0	0	ولتاژی
1	1	1	1	1	1	جریانی

برای تنظیم نوع جریانی بودن کانالهای ورودی و خروجی مطابق جدول زیر عمل می کنیم .

چهار کانال ورودی AD				
Bit 11	Bit 10	Bit 9	Bit 8	
0	0	0	0	$-20mA \sim +20mA$
1	1	1	1	$4mA \sim +20mA$

دو کانال خروجی DA		
Bit 13	Bit 12	
0	0	$0mA \sim +20mA$
1	1	$4mA \sim +20mA$

نوع و محدوده ورودی خروجی‌های آنالوگ و مقدار دیجیتال شده آنها مطابق جدول زیر می‌باشد .

خروجی	ورودی	محدوده / بازه آنالوگ
حد تفکیک	حد تفکیک	
$-2000 \sim +2000$	$-2000 \sim +2000$	$-10v \sim +10v$
	$-2000 \sim +2000$	$-20mA \sim +20mA$
	$0 \sim +2000$	$4mA \sim +20mA$
$0 \sim +4000$		$0mA \sim +20mA$
$0 \sim +4000$		$4mA \sim +20mA$

مثال ۹۷ :

می‌خواهیم کانال اول و دوم ورودی PLC مدل DVP-20SX2 را به صورت ولتاژی و کانالهای سوم و چهارم ورودی را به صورت جریانی (4 – 20mA) و کانالهای خروجی را به صورت جریانی (4 – 20mA) تنظیم کنیم .

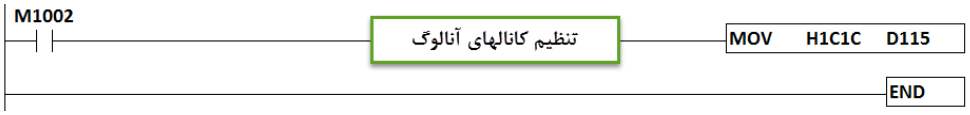
کانالهای ورودی				رزرو		کانالهای ورودی جریانی از نوع -20...+20 mA		جریانی بودن کانال اول خروجی		ولتاژی بودن کانال دوم خروجی		رزرو	
ورودی ولتاژی	ورودی جریانی			کانالهای خروجی									
					ولتاژی	جریانی							
Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13	Bit14	Bit15
0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0

حال باید این بیت‌ها را به صورت چهارتایی جدا کنیم و تبدیل مبنای اعداد را انجام دهیم:

8421	8421	8421	8421
0001	1100	0001	1100
1	12	1	12

طبق شکل بالا عدد تبدیل شده در مبنای هگز برابر H1C1C می‌شود. (عدد دسیمال ۱۲ طبق جدول تبدیل مبنا معادل کد هگز **C** می‌باشد).

برای ارسال این تنظیمات به PLC مدل DVP 20SX2 باید کد هگز بدست آمده را از طریق دستور MOV به رجیستر D1115 که مربوط به تنظیمات کانالها می‌باشد، ارسال کنیم.



مثال ۹۸ :

می‌خواهیم نمونه برداری کانالهای ورودی به صورت میانگین‌گیری و یا لحظه‌ای باشد. طبق جدول رجیسترها، برای تنظیم این حالت باید مقدار رجیستر D1062 بین اعداد 1 تا 20 باشد. در صورتیکه این مقدار برابر 1 باشد، نمونه برداری کانال به صورت لحظه‌ای و در صورتیکه این مقدار برابر 2 تا 20 باشد نمونه برداری بصورت میانگین گیری خواهد بود.

مثال ۹۹ :

می‌خواهیم کانالهای PLC مدل DVP10SX را طبق سنسورهای مورد نیازمان تنظیم کنیم.

کانال اول ورودی : ولتاژی

کانال دوم ورودی: جریانی 4 – 20mA

کانال اول خروجی: ولتاژی

کانال دوم خروجی: جریانی

ابتدا باید طبق فایل User Manual نوع ورودی و خروجی در کانالها را مشخص کنیم.

نوع ورودی	ورودی اول		ورودی دوم	
	B0	B1	B2	B3
حالت ولتاژی	0	0	0	0
حالت جریانی 0 – 20mA	1	0	1	0
حالت جریانی 4 – 20mA	1	1	1	1

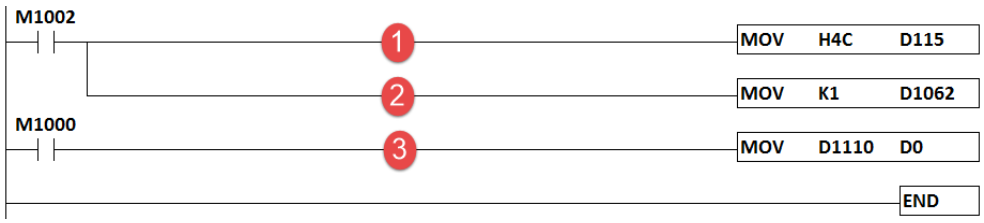
نوع خروجی	خروجی اول		خروجی دوم	
	B4	B5	B6	B7
حالت ولتاژی	0	0	0	0
حالت جریانی	1	0	1	0

ورودی	ورودی اول		ورودی دوم		خروجی اول		خروجی دوم		رزرو		رزرو	
نوع	ولتاژی		4 – 20mA		ولتاژی		جریانی					
	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11
بیت	0	0	1	1	0	0	1	0	0	0	0	0

حال کافیست عدد هگز مورد نظر را محاسبه و دستورات مورد نظر را در PLC بنویسیم.

0	0	0	0	0	1	0	0	1	1	0	0
0				4				12=C			

پس از بدست آوردن کد هگز مربوط به تنظیمات ، برنامه‌ای مطابق زیر می‌نویسیم.



۱ – انتقال تنظیمات مربوط به کانالها به دیتا رجیستر خاص D1115 (این دیتا رجیستر مربوط به نوع تنظیمات می‌باشد)

۲ – تنظیم دیتا رجیستر D1062 : این دیتا رجیستر برای مشخص کردن میانگین یا لحظه‌ای بودن نمونه برداری می‌باشد. اگر K1 به این دیتا رجیستر منتقل شود نوع نمونه برداری بصورت لحظه‌ای بوده و اگر مثلا عددی برابر K10 منتقل شود ، بعد از۱۰ بار نمونه گیری، میانگین آن را محاسبه می کند. این دیتا رجیستر به صورت پیشفرض بر روی K20 تنظیم است.

۳ - مقدار خوانده شده از کانال اول ورودی آنالوگ درون دیتا رجیستر خاص D1110 ریخته می‌شود و می‌توان با انتقال مقدار درون این دیتا رجیستر درون یک دیتا رجیستر دیگر مانند D0 ، مقدار آن را مشاهده کرد. همچنین می‌توانید به صورت مستقیم دیتا رجیستر D1110 را درون HMI فراخوانی کنید.

معرفی کارت لودسل DVP-01LC-SL :

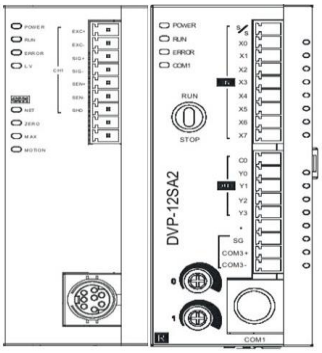
در صنعت ، مبحث توزین در خطوط تولید و بسته بندی و ... بسیار با اهمیت و پر کاربرد است لذا در این موارد می‌توان از سنسور لودسل و کارت لودسل دلتا استفاده کرد.

کارت لودسل DVP-01LC-SL دارای یک کانال برای خواندن مقدار وزن توسط سنسور لودسل می‌باشد. البته کارت DVP-02LC-SL نیز موجود بوده که قابلیت خواندن دو سنسور وزن را دارد.

این کارتها می‌توانند از طریق شبکه مدباس با PLC ارتباط برقرار کند همچنین می‌توانند مستقیم به PLC متصل می‌شود .

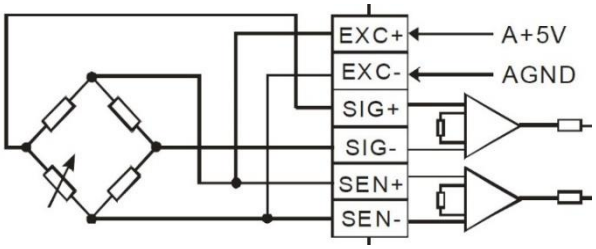
نکته : کارتهای لودسل از سمت چپ به PLC متصل می‌شوند (Left Side).

در شکل زیر همانطور که ملاحظه می‌کنید کارت لودسل (سمت چپ) به PLC مدل 12SA2 متصل شده است.



اگر می خواهید توسط دستورات آنالوگ (از طریق PLC) عملیات کالیبره ، صفر کردن و خواندن وزن کارت لودسل فوق را انجام دهید باید به این نکته توجه داشته باشید که کارت های لودسل دلتا فقط می‌توانند به CPU های با قابلیت Left Side متصل شوند .

روش سیم بندی ۴ و ۶ سیمه کارت لودسل تک کاناله دلتا بصورت زیر می‌باشد :



در شکل قبل از سنسور لودسل ۴ سیمه برای اتصال به کارت لودسل دلتا استفاده نموده‌ایم .

ترمینال‌های EXC + و EXC - ولتاژ 5VDC برای تغذیه سنسور و ترمینال های SIG + و SIG - برای اتصال سیگنال خروجی سنسور به کارت لودسل بکار می روند.

نکته: کارت های لودسل دلتا قابلیت اتصال به سنسورهای ۶ سیمه را دارا می‌باشند و از آنجایی که در صنعت بیشتر سنسورهای ۴ سیمه مورد استفاده می گیرند لذا باید ترمینال های SEN + را به EXC + و SEN – را به EXC - متصل نمائیم.

کارت های لودسل تک کاناله و دو کاناله دلتا دارای درگاه های ارتباطی RS232 و RS485 می باشند.

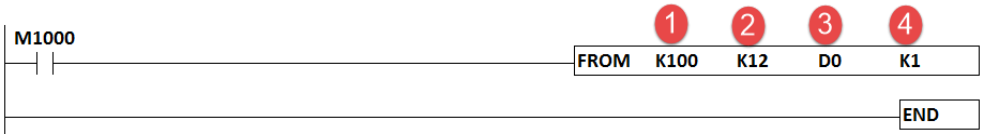
حال می‌خواهیم مقدار وزنی که که از سنسور لودسل به کارت لودسل ارسال می‌شود را توسط PLC بخوانیم.

نکته : مقدار عددی که کارت لودسل به plc ارسال می‌کند ، یک عدد دیجیتال نامفهوم بوده و لذا نیاز به کالیبره کردن دارد.

از طریق درگاه RS232 می‌توان توسط نرم افزار LC-SOFT عملیات کالیبراسیون ، تنظیمات ارتباطی ، تعیین زمان نمونه برداری و ...کارت لودسل را انجام دهیم .

مثال ۱۰۰ :

می‌خواهیم مقدار وزن یک گونی برنج را که توسط یک سنسور لودسل اندازه‌گیری می‌شود را در PLC مدل 12SA2 بخوانیم. برای این کار ابتدا کارت لودسل را به PLC مورد نظر متصل می‌کنیم. سپس ورودی های تغذیه‌ی کارت لودسل و PLC را به منبع تغذیه وصل می‌کنیم. حال باید سیم های سنسور لودسل را به کانالهای کارت لودسل مطابق شکل بالا وصل کنیم. پس از وصل‌کردن سنسور به کارت لودسل، برنامه‌ای مطابق زیر می نویسیم.



۱ – آدرس کارت لودسل

نکته : در نظر داشته باشید کارتهایی که از سمت راست به PLC متصل می شوند (Left Side)، از آدرس K100 شروع می‌شوند. در اینجا کارت لودسل، اولین کارت آنالوگ بعد از PLC می‌باشد.

۲ – کنترل رجیستر شماره 12 برای خواندن مقدار کانال اول کارت لودسل می‌باشد. (به User Manual کارت لودسل رجوع شود).

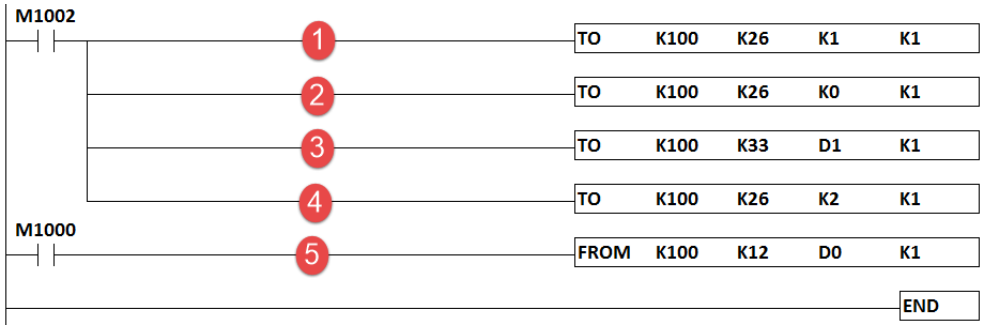
۳- مقدار خوانده شده، درون این رجیستر ریخته می‌شود.

۴ – فقط مقدار کانال اول خوانده می‌شود. اگر در این دستور بجای K1 از K2 استفاده کنیم مقدار کانال اول خوانده شده و درون رجیستر D0 ریخته می‌شود و مقدار کانال دوم خوانده شده و درون رجیستر D1 ریخته می‌شود.

کالیبره‌کردن کارت لودسل :

مثال ۱۰۱ :

برای کالیبره‌کردن کارت لودسل، برنامه‌ای بصورت زیر می‌نویسیم .



۱ – با تحریک M10، دستور به عنوان پاره سنگ‌گیر عمل کرده و مقدار وزن را صفر می‌کند.

۲ – با استفاده از این دستور مقدار صفر را برای PLC مشخص می‌کنیم.

۳– با استفاده از این دستور مقدار وزنه را مشخص می‌کنیم. مثلا یک وزنه یک کیلوگرمی روی سنسور قرار داده و مقدار D1 در دستور بالا را برابر 1000 گرم قرار می‌دهیم.

۴ – با تحریک M13 عملیات کالیراسیون ثبت می‌گردد.

۵- دستور خواندن مقدار وزن توسط کارت لودسل

مراحل نوشتن برنامه :

ابتدا دستور خط شماره 1 را نوشته تا بتوانیم مقدار دیجیتال وزن را مشاهده کنیم سپس با استفاده از خط 2 برنامه مقدار سنسور را صفر می‌کنیم (پارسنگ‌گیری) . سپس با استفاده از خط 3 برنامه مقدار صفر را به عنوان مبنای پایه، ثبت می‌کنیم. سپس یک وزنه مثلا 1 کیلوگرمی روی سنسور قرار داده و مقدار درون D0 را برابر 1000 قرار می دهیم. با استفاده از برنامه خط 5 این تنظیمات ثبت خواهد شد.

حال در صورتیکه یک وزنه‌ی مثلا ۳ کیلوگرمی روی سنسور قرار دهیم مقدار خوانده شده توسط خط اول برابر 3000gr خواهد بود.

کنترل کننده PID :

کنترل PID از ترکیب کنترل کننده تناسبی (P)، انتگرال‌گیر (I) و مشتق‌گیر (D)، تشکیل شده است . به عنوان مثال می‌خواهیم دمای یک اتاق را توسط هیتر، روی ۵۰ درجه سانتیگراد تثبیت کنیم لذا مقدار مطلوب ما یا همان Set Point برابر با SP=50 می‌باشد. چنانچه هیتر را روشن کنیم و دمای اتاق ۵۰ درجه شود یعنی اینکه دمای جاری اتاق (PV) و دمای مطلوب ما (SP) برابر شده‌اند به عبارت دیگر خطا صفر شده پس فقط کنترل‌کننده تناسبی (P) وارد عمل می‌شود و پروسه را کنترل می‌کند و قسمت انتگرالیر (I) و مشتق گیر (D) برابر صفر خواهند بود .

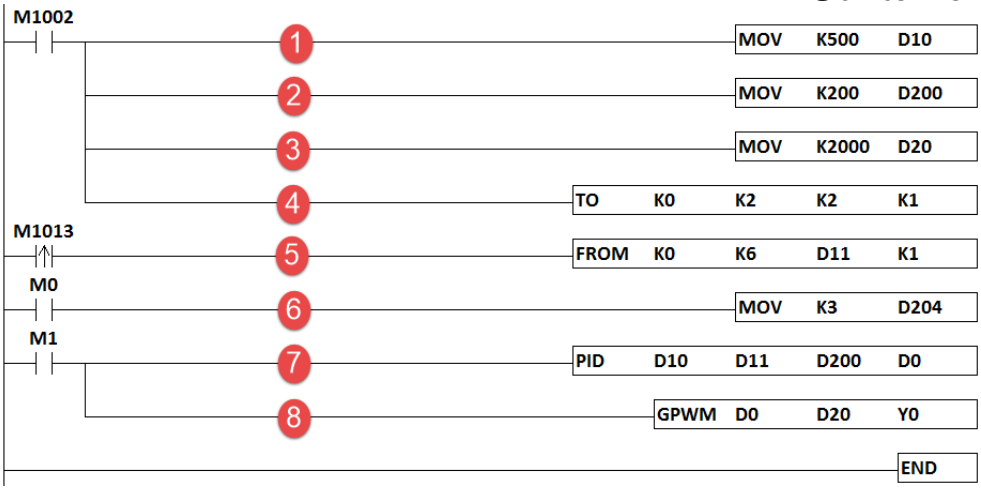
اگر دمای اندازه‌گیری شده بیشتر یا کمتر از مقدار مطلوب (SP) شود آنگاه کنترل کننده انتگرالی و مشتق گیر وارد عمل خواهند شد تا پروسه را به مقدار مطلوب برسانند و یا به عبارتی خطا را صفر کنند زمانی این دو کنترل کننده وارد عمل خواهند شد که ما سیگنال خطا داشته باشیم آنگاه قسمت انتگرال‌گیر از سیگنال خطا انتگرال می‌گیرد و قسمت مشتق‌گیر هم از سیگنال خطا مشتق می‌گیرد به یک چنین سیستم کنترلی ، اصطلاحا کنترل کننده PID می‌گویند .

کنترل کننده PID کاربرد مهمی در پروسه‌ها مخصوصا پروسه‌های غیر قابل پیش بینی دارد.

PID با محاسبه خطاهای قبلی و خطاهای حال، احتمالات آینده را پیش‌بینی کرده و تدابیری برای کنترل هر چه بهتر سیستم اتخاذ می‌کند.

مثال ۱۰۲ :

می‌خواهیم دمای یک کوره را به صورت PID کنترل کنیم، به صورتیکه تغییرات دما نسبت به دمای تنظیمی ما (Set Point)، که مقدار ۵۰ درجه سانتیگراد می‌باشد کمترین خطا را داشته باشد. این مثال برای ثابت نگهداشتن دما در فرآیند های صنعتی بسیار مفید می‌باشد . یکی از دستوراتی که در کنترل هیترها بسیار کاربرد دارد PWM می‌باشد این دستور با تغییر Duty Cycle پالسهای ارسالی به SSR و بالطبع قطع و وصل‌های مکرر، بطور دقیق میتواند حرارت تولید شده توسط هیتر را کنترل کند . مثال صفحه بعد نمونه ای کاربردی از کنترل هیترها با استفاده از دستور PID و دستور GPWM می‌باشد.



۱ – انتقال مقدار تنظیمی مورد نظر(SP) که معادل K500 می‌باشد به دیتا رجیستر D10

نکته : مقدار تنظیمی برابر ۵۰ درجه سانتیگراد می‌باشد و از آنجا که کارت دمای DVP 04TC دارای ضریب دقت 0.1 درجه می‌باشد، لذا از K500 استفاده کرده‌ایم.

۲ – زمان نمونه برداری که برابر 2s می‌باشد. 200 * 10ms = 2s

۳ – زمان یک سیکل یا دوره در دستور PWM که برابر 2s می‌باشد. 2000ms = 2s

۴ – تعداد نمونه گیری (مشخص کردن لحظه‌ای یا میانگین بودن مقدار)

۵ – خواندن مقدار دمای کانال اول کارت DVP04TC

۶ – تنظیم ضرایب PID به صورت اتو تیون (K3)

نکته : برای اتوتیون کردن، کافیست یک لحظه M0 فعال شود.

۷ – دستور PID

۸ – دستور GPWM برای فرمان دادن به رله حالت جامد یا اصطلاحا SSR

نکته : خروجی PLC باید از نوع ترانزیستوری باشد.

نکته : زمانیکه اتوتیون بعد از چند دقیقه کامل شد ، بطور خودکار دیتا رجیستر D204 برابر K4 می‌شود.

در این مثال از SSR با ورودی DC و خروجی AC استفاده شده است. همچنین از یک المنت برقی و یک کارت DVP04TC دلتا و PLC مدل DVP28SV2 از نوع ترانزیستوری استفاده کرده‌ایم.

فصل ششم

دانلود و آپلود برنامه PLC دلتا

برنامه نوشته شده توسط کاربر معمولا ابتدا در سیمولاتور نرم افزار تست شده و پس از اطمینان از صحت عملکرد آن، باید به داخل PLC منتقل شود .

PLC های دلتا دارای پورت های مختلفی جهت دانلود و یا آپلود می باشند انواع این پورت ها بصورت زیر می باشند :

پورت سریال RS232 : PLCهای مدل , 14SS2 , 12SA2 , 28SV2 , 10SX , 20SX2 , ES2 EX2 و .. مجهز به این پورت می باشند .

پورت سریال RS485 : PLC های مدل , 14SS2 , 12SA2 , 28SV2 , 10SX , 20SX2 , 12SE , ES2 , EX2 و .. مجهز به این پورت می باشند .

پورت USB : PLCهای مدل 12SE , 20SX2 مجهز به این پورت می باشند .

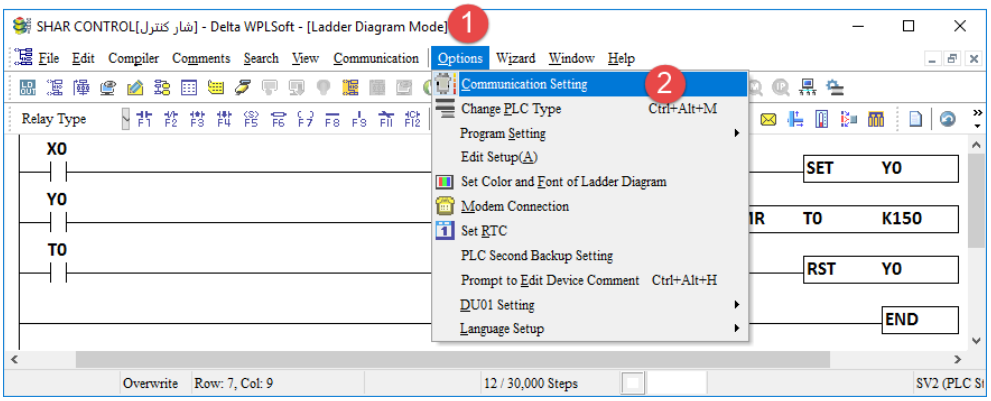
پورت LAN : PLCهای مدل 12SE مجهز به این پورت می باشند .

نکته : انتقال برنامه از کامپیوتر به PLC دانلود نامیده می شود . همچنین انتقال برنامه از PLC به کامپیوتر، آپلود نامیده می شود . قبل از اقدام به دانلود و یا آپلود برنامه، تنظیماتی باید صورت بگیرد :

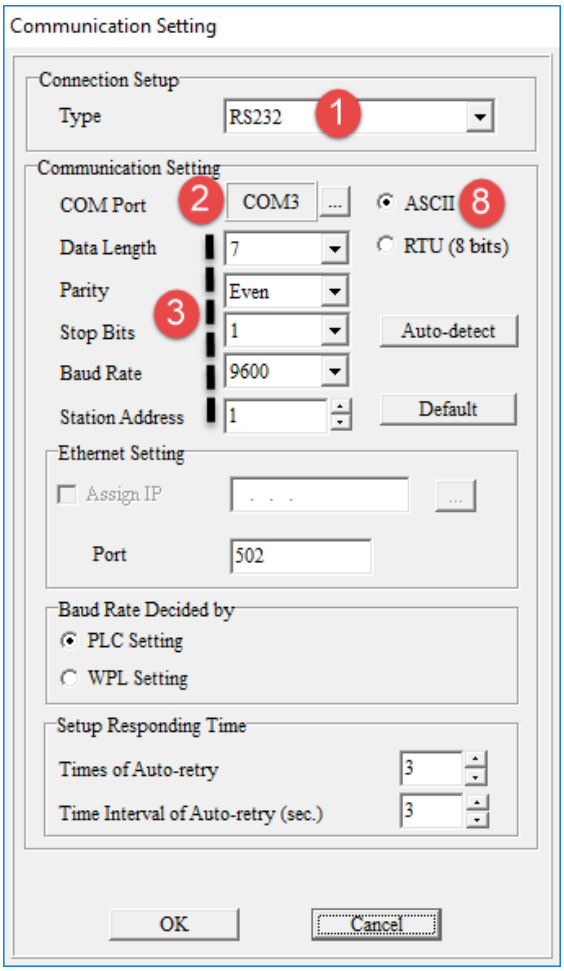
تنظیمات پورت RS232 :

چنانچه PLC دارای پورت سریال باشد، نیاز به کابل سریال ساخته شده توسط کمپانی دلتا داریم . یک سر این کابل(سوکت گرد) را با رعایت جهت قراگیری این سوکت، به PLC متصل می‌کنیم . سر دیگر این کابل را به پورت سریال کامپیوتر متصل می کنیم . سپس تغذیه PLC را وصل می کنیم . در این صورت چراغهای روی PLC روشن خواهند شد .

در نرم افزار WPLSOFT ، مطابق شکل زیر عمل می کنیم .



از منوی Options گزینه Communication Setting را انتخاب می کنیم . در اینصورت شکل صفحه بعد ظاهر می باشد توضیحات اعداد مشخص شده روی شکل در صفحه بعد می باشد .



۱-انتخاب نوع ارتباط که در این مثال، سریال و استاندارد RS232 می باشد .

۲-در صورتیکه کامپیوتر شما دارای پورت سریال باشد این قسمت COM1 و یا COM2 خواهد بود.

۳-این قسمت تنظیمات پروتکل ارتباطی می باشد مقادیر مشخص شده در شکل پیشفرض می باشد .

در PLC دلتا مقادیر پروتکل ارتباطی بصورت پیشفرض بصورت زیر می باشد:

9600 , 1 , E , 7 , ASCII و آدرس PLC نیز 1 می‌باشد.

(Station Address)

در صورتیکه پروتکل ارتباطی PLC قبلا تغییر کرده باشد، در اینجا هم باید تنظیماتی مشابه PLC انجام شود .

نکته: چنانچه آدرس PLC را نمی‌دانید میتوانید قسمت Station Address را برابر 0 قرار دهید در اینصورت آدرس PLC هر چه باشد، ارتباط برقرار خواهد شد .

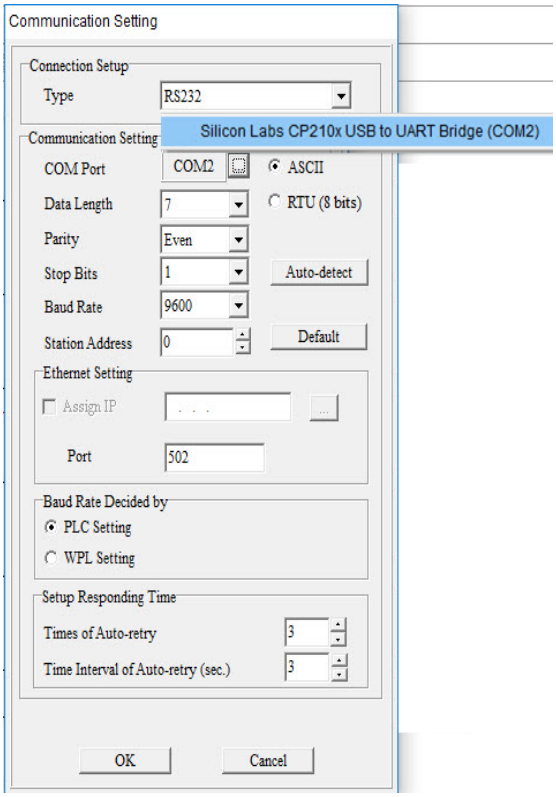
۴-تنظیمات RTU و یا ASCII نیز باید مطابق تنظیمات PLC باشد پیشفرض ASCII می باشد.

با تایید تنظیمات فوق توسط OK ، آماده دانلود و یا آپلود برنامه می شویم .

نکته : قبل از دانلود برنامه به داخل PLC حتما باید سیمولاتور برنامه بسته شود در غیر اینصورت برنامه به داخل سیمولاتور ارسال می شود .

نکته : در صورتیکه برنامه را قبلا کامپایل نکرده باشید، در هنگام دانلود پیامی در این خصوص ظاهر خواهد شد با تایید این پیام، دانلود و یا آپلود انجام خواهد شد .مطابق شکل زیر گزینه دانلود و یا آپلود را انتخاب نمایید. ۱-دانلود ۲- آپلود

نکته : بسیاری از کامپیوترهای شخصی (Laptop) ، فاقد پورت سریال می باشند در اینصورت جهت دانلود و یا آپلود برنامه به PLC باید از مبدل USB>RS232 استفاده کرد . این مبدلها در بازار موجود می باشند . پیشنهاد می شود از مبدل ساخت شرکت دلتا (IFD6601) که شبیه یک فلش مموری بوده استفاده نمایید .



زمانیکه از این مبدلها استفاده می‌کنید، درایور آن را نیز باید نصب کنید . سپس در نرم افزار PLC از منوی Options > Communication Setting مطابق شکل، درایور مبدل نصب شده به Laptop را انتخاب کنید . در این شکل با کلیک روی دکمه مقابل COM PORT ، درایورهای نصب شده روی کامپیوتر شما ظاهر می شود همانطور که از شکل دیده می شود، پورت USB، که مبدل به آن متصل شده به عنوان COM2 شناسایی شده است آنرا انتخاب کرده و تنظیمات دیگر را مطابق مطالب قبلی تکمیل نمایید .

نکته : آدرس PLC های دلتا بصورت پیش‌فرض 1 می باشد اما چنانچه آن را نمی دانید در قسمت Station Address عدد 0 را قرار دهید .

فصل هفتم

آموزش زبان SFC

در صنعت اتوماسیون برای فرآیندهایی که به صورت ترتیبی و تکراری انجام میگیرد می‌توان از زبان SFC استفاده کرد. این زبان یکی از پنج زبان استاندارد IEC61131 می‌باشد که بسیار شبیه برنامه نویسی فلوچارتی می‌باشد و برنامه نویسی را بسیار ساده می‌کند.

در برخی از فرآیند ها یک برنامه باید به صورت مرحله به مرحله انجام شود به طوری که تا یک مرحله به اتمام نرسیده است، مرحله بعدی آغاز نشود و هر مرحله مشروط به یک ورودی خواهد بود. به عنوان مثال فرآیند عملیات یک ماشین لباسشویی و یا یک کارواش تمام اتوماتیک بصورت مرحله به مرحله انجام می‌شود و در واقع تا یک مرحله به طور کامل انجام نگیرد مرحله بعدی اجرا نخواهد شد.

در PLC دلتا نیز این زبان ایجاد شده است، لذا مفاهیمی در خصوص زبان SFC وجود دارد که در ادامه به شرح آنها می پردازیم.

۱ – **Sequence :** به تعدادی مرحله یا Step که به صورت متوالی جهت انجام یک فرایند انجام می‌شوند گفته می شود و در هر برنامه می‌توان چند Sequence داشت.

۲ – **Step :** گام یا مرحله که در آن می‌توان برنامه‌ای اجرا شود مثلا خروجیهایی روشن یا خاموش شوند.

تعداد Step ها در برنامه WPLSoft برابر با S0 الی S1024 می‌باشد. این Step ها به ۲ دسته تقسیم می‌شوند:

الف) Step های اولیه یا آغازین (Initial) که از S0 الی S9 می‌باشد.

ب) Step های عمومی که از S10 الی S1024 می‌باشد.

۳ – **Action :** به عملیاتی که در هر Step انجام می‌شود اصطلاحا Action گفته می‌شود. در واقع هر Action می‌تواند شامل دستورات برنامه نویسی باشد مانند LD ، OUT ، SET ، RST ، CMP ، ZRST ، TMR ، CNT و ... باشد.

۴ – **Transition :** شرط عبور از یک Step یا مرحله به مرحله دیگر را گویند. مثلا اگر ورودی X0 فعال شود برنامه از مرحله فعلی به مرحله بعدی انتقال میابد.

نکته : وقتی اجرای برنامه در یک مرحله (Step) باشد هیچکدام از Step های بعدی یا قبلی اجرا نمی‌شوند لذا در اینصورت مدیریت کنترل برنامه راحتتر انجام می‌شود.

انواع Sequence در زبان SFC :

۱ – یک توالی ساده بدون شاخه های موازی همزمان یا غیر همزمان

۲ – یک توالی پیچیده که شامل شاخه‌های موازی همزمان و یا غیر همزمان است

۳ – توالی چندتایی یعنی بیش از یک توالی در برنامه وجود داشته و مستقل از یکدیگر اجرا می شوند.

برخی از ابزارهای کاربردی در زبان SFC :

۱ – F1 : LAD دیاگرام ۲ – F2 : مرحله ابتدایی ۳ – F3 : مرحله عمومی

۴ – F4 : پرش از یک مرحله به مرحله دیگر ۵ – F5 : شرط عبور از یک مرحله به مرحله دیگر

۶ – F6 : شرط انتخاب و جایگزینی مرحله (شرط واگرایی یا منشعب شدن مرحله) و اجرای یکی از آنها

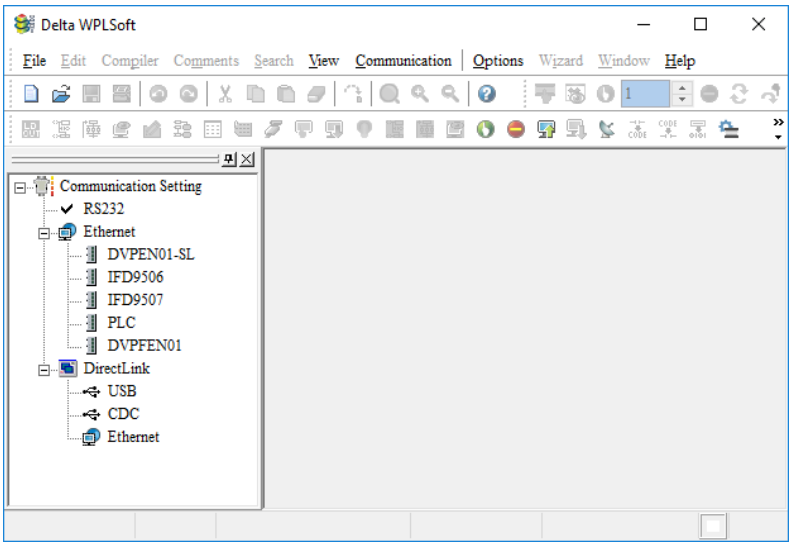
۷ – F7 : شرط همگرایی مرحله (شرط همگرا شدن یا یکی شدن انشعابات مرحله)

۸ – F8 : شرط واگرایی یا منشعب شدن مرحله و اجرای موازی آنها

۹ – F9 : شرط همگرایی یا یکی شدن مراحلی که قبلا موازی اجرا می شدند.

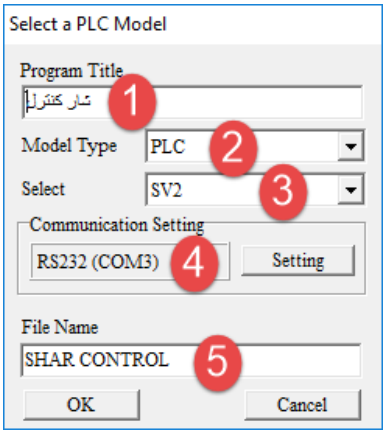
شروع برنامه نویسی به زبان SFC :

با کلیک بر روی این آیکن برنامه WPLSOFT مطابق شکل زیر اجرا می‌شود.



حال نیاز است که مانند تمام برنامه‌های مشابه در این زمینه، پروژه‌ای جدید ایجاد کنیم. برای این کار کافیست همانند شکل زیر به قسمت FILE رفته و گزینه NEW را انتخاب کنید و یا در قسمت پایین گزینه فایل، آیکون مربوط به ایجاد پروژه‌ی جدید را انتخاب کنید و یا از کلید های میانبر Ctrl + N استفاده کنید.

پس از کلیک بر روی گزینه NEW پنجره ای به شکل زیر اجرا می‌شود.



۱ – در این قسمت عنوان پروژه را وارد کنید.

۲ –نوع کنترلر مورد نظر خود را مشخص کنید.

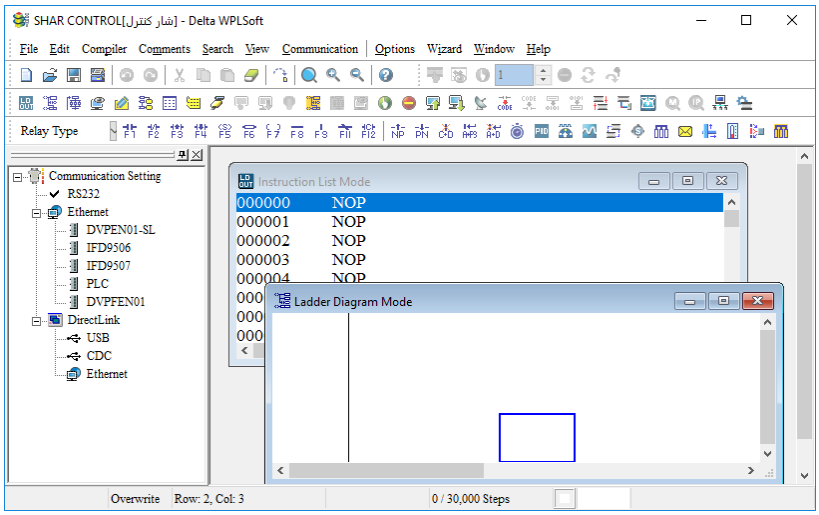
۳- مدل PLC را انتخاب کنید.

۴ –نوع ارتباط PLC با کامپیوتر مشخص شود.

۵ – در این قسمت نیز می‌توانید نام پروژه را برای ذخیره کردن وارد کنید.

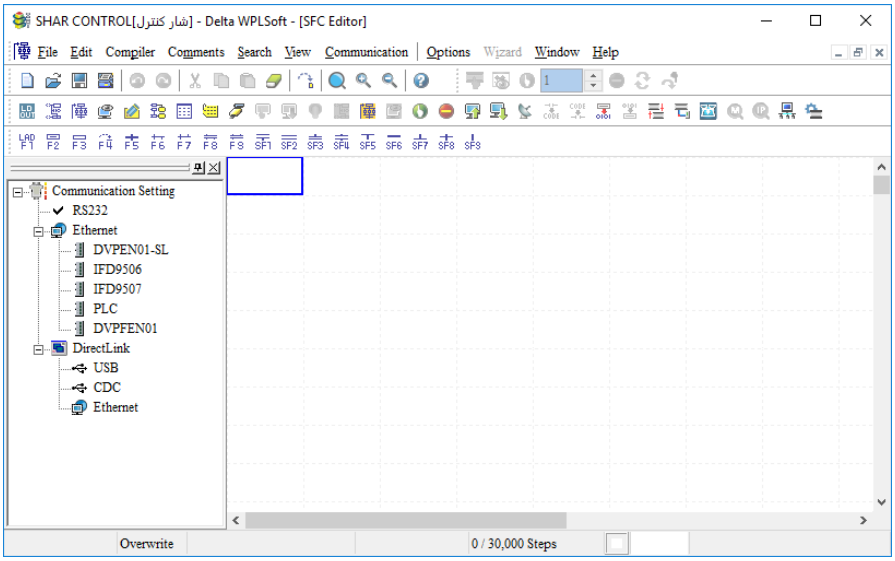
در پایان با کلیک بر روی گزینه OK اطلاعات ثبت شده و وارد مرحله بعد می‌شود.

در این مرحله مطابق شکل زیر دو پنجره کوچک در صفحه اصلی برنامه WPLSOFT ظاهر شده که یک پنجره مربوط به زبان LAD و پنجره بعدی مربوط به زبان IL می باشد .

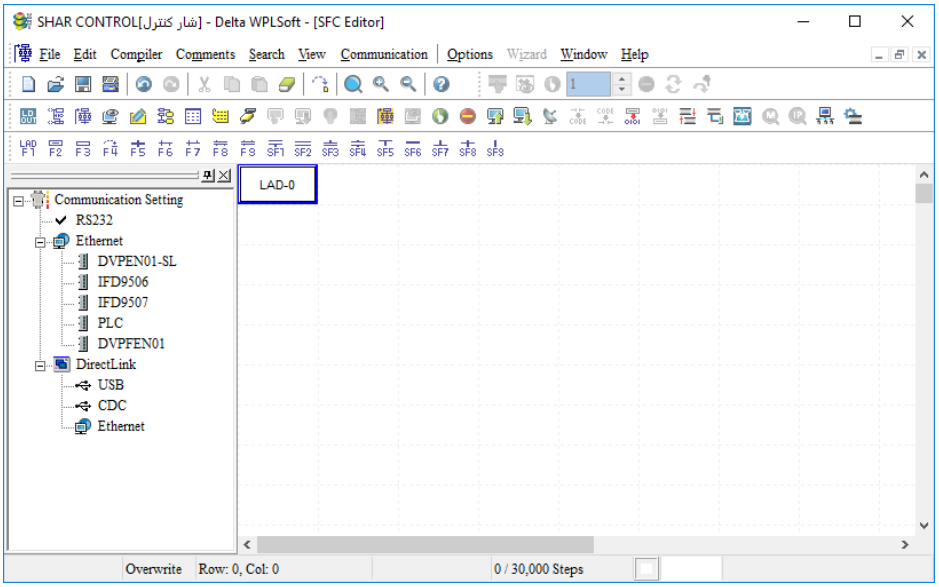


حال باید در این مرحله، زبان برنامه نویسی را به SFC تغییر دهیم لذا مانند شکل زیر باید نوع زبان را انتخاب کنیم.

حال صفحه برنامه نویسی در برنامه WPLSoft به شکل زیر تغییر خواهد کرد.

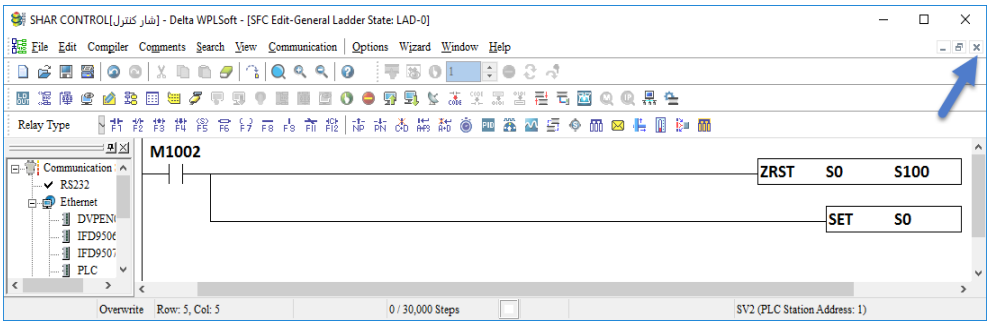


در نوار ابزار گزینه LAD F1 را انتخاب کرده و یا کلید F1 را می‌زنیم. در اینصورت عبارت LAD-0 در ابتدای برنامه، مانند شکل زیر ظاهر می‌شود.

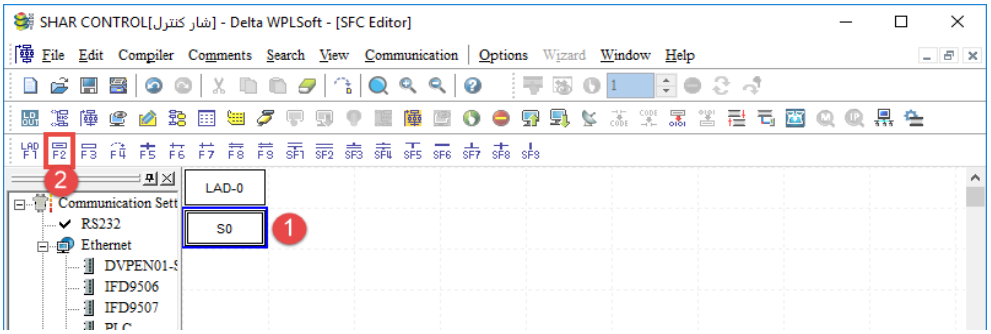


اگر بخواهیم در ابتدای شروع برنامه، تنظیمات خاصی انجام گیرد، می‌توان در این قسمت این کار را انجام داد مثلا می‌خواهیم از S0 الی S100 را که قبلا در برنامه روشن بوده‌اند را ریست کنیم. برای این کار بر روی LAD-0، راست کلیک کرده و گزینه‌ی Action/transition editor را انتخاب می‌کنیم و در پنجره باز شده برنامه را طبق شکل صفحه بعد می‌نویسم.

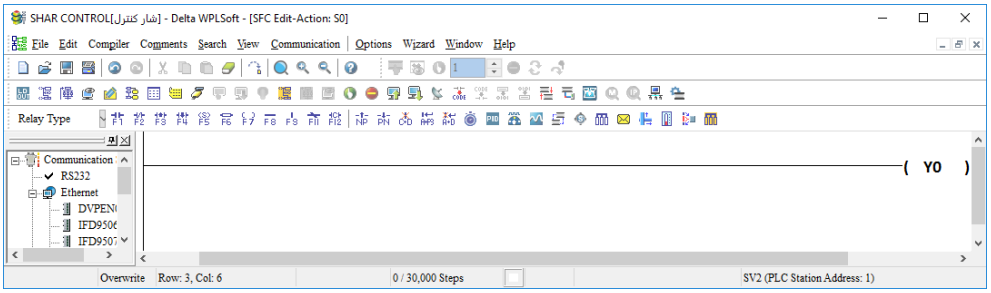
این برنامه باعث می‌شود که در اولین سیکل اجرای برنامه، مراحل S0 تا S100 پاک شده و اجرای برنامه از مرحله S0 شروع شود. سپس طبق شکل، روی گزینه close کلیک کرده و در ادامه در صورت باز شدن پنجره جدید، گزینه yes را می‌زنیم و بصورت خودکار به صفحه قبل باز می‌گردیم .



حال مطابق شکل زیر ابتدا مربع زیر LAD-0 را انتخاب کرده و از نوار ابزار، روی گزینه‌ی مشخص شده در شکل زیر کلیک کرده و یا از کلید میانبر F2 استفاده می‌کنیم. در اینصورت Step شماره S0 ظاهر می‌شود.

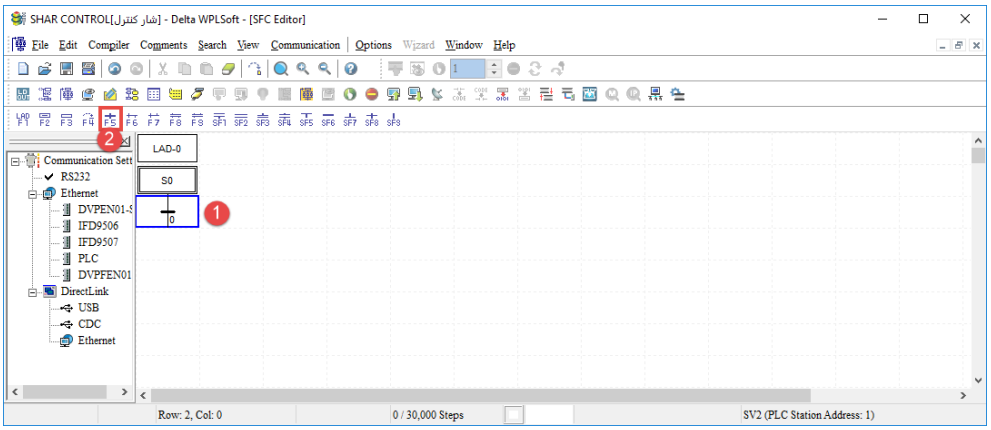


فرض کنید در این مرحله می‌خواهیم یک خروجی مانند Y0 فورا روشن شود لذا روی S0 راست کلیک کرده و گزینه‌ی Action/transition editor را انتخاب و در صفحه باز شده، برنامه مورد نظر را مطابق شکل زیر می نویسیم.

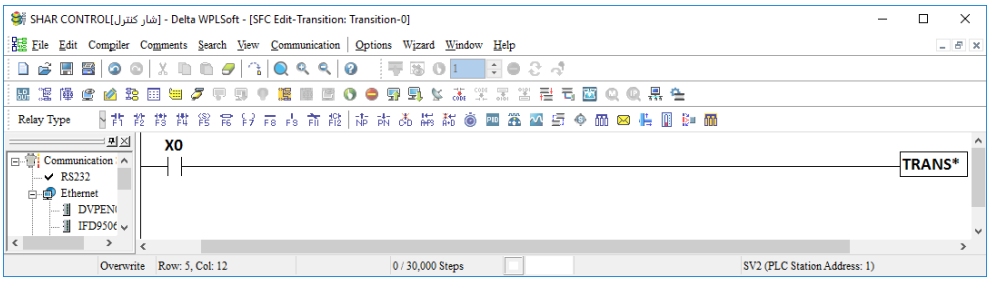


پس از نوشتن برنامه مطابق شکل صفحه قبل، ابتدا روی گزینه close کلیک کرده و در صورت نیاز گزینه yes را می‌زنیم تا برنامه به صورت اتوماتیک به صفحه SFC Editor باز گردد.

حال برای عبور از مرحله S0 به مرحله بعدی، باید یک شرط بگذاریم. برای اینکار ابتدا مربع زیر S0 را انتخاب و مطابق شکل زیر در قسمت نوار ابزار برنامه، گزینه Transition Condition Between Steps را زده و یا از کلید میانبر F5 استفاده می‌کنیم.



در این قسمت نیز مطابق مراحل قبلی عمل می‌کنیم بر روی شرطی جدید، راست کلیک کرده و گزینه‌ی Action/transition editor را انتخاب می‌کنیم. فرض کنید می‌خواهیم با تحریک X0 از مرحله فعلی یعنی S0 به مرحله بعدی یعنی S10 برویم. در این صورت کافیست در صفحه‌ی برنامه باز شده شرط X0 را اعمال کنیم.



سپس همانند مراحل قبل ابتدا روی گزینه‌ی close کلیک کرده و سپس در صورت نیاز گزینه yes را انتخاب می‌کنیم تا به صورت اتوماتیک به صفحه برنامه نویسی SFC Editor وارد شویم.

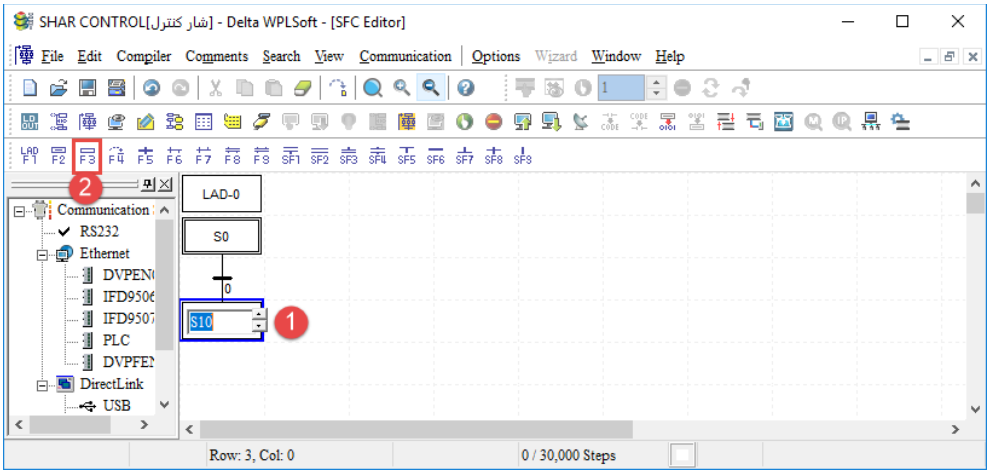
نکته : از آنجاییکه از S0 الی S9 جزء Step های اولیه می‌باشند ، لذا در ادامه این Sequence باید از Step های عمومی که از S10 شروع می‌شود استفاده کرد.

نکته : در قسمت Transition که مربوط به اعمال شرط در برنامه می‌باشد، اگر چیزی ننویسیم مشکلی پیش نمی آید.

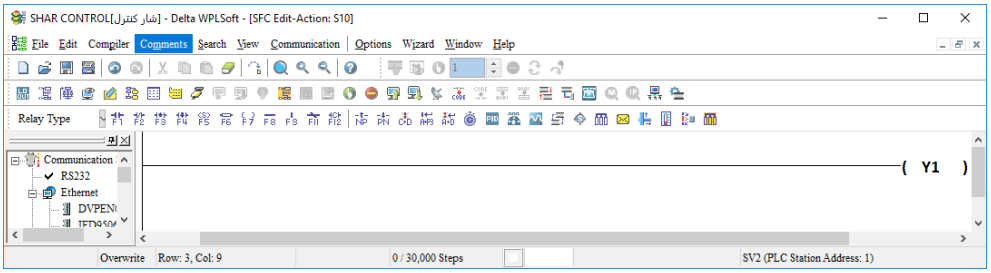
نکته : همیشه در اولین قسمت برنامه SFC یعنی LAD-0 برنامه‌ی ست کردن S0 برای رفتن به اولین مرحله نوشته شود.

مطابق برنامه‌ی نوشته شده به محض تحریک X0 برنامه از مرحله فعلی (S0) به مرحله بعدی(S10) خواهد رفت و به محض خروج از S0 ، خروجی Y0 خاموش خواهد شد. در صورتیکه بخواهیم خروجی Y0 هنگام رفتن به مرحله بعدی خاموش نشود باید این خروجی را با استفاده از دستور SET در برنامه بنویسیم.

حال در این مرحله باید مجدداً یک Step از نوع عمومی (S10 الی S1024) انتخاب کنیم لذا ابتدا مربع زیر شرط 0 را انتخاب کرده سپس در قسمت نوار ابزار از گزینه‌ی general steps و یا کلید میانبر F3 مطابق شکل زیر استفاده کنید.

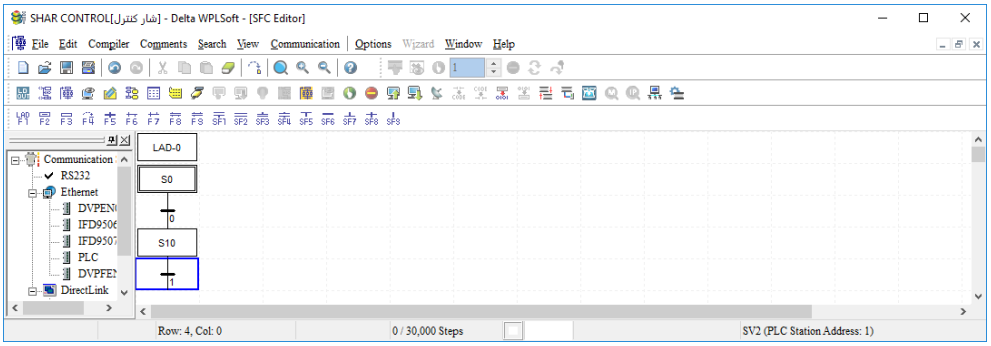


همانطور که ملاحظه می‌کنید یک Step جدید با شماره S10 به برنامه اضافه شده است. برای اینکه نوع عملکرد برنامه را در این Step مشخص کنیم، باید همانند مراحل قبل، روی این Step راست کلیک کرده و گزینه‌ی Action/transition editor را انتخاب و برنامه‌ی مورد نظر را در صفحه باز شده بنویسیم. فرض کنید در این مرحله می‌خواهیم خروجی (Y1) فعال شود .



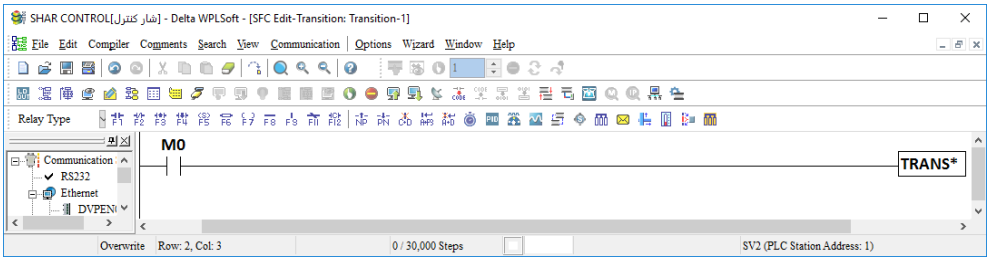
با انجام همان مراحل مشابه قبل ، این Step نیز نوشته شده و می‌توانیم به مرحله‌ی بعدی برویم. برای رفتن به مرحله بعدی باید دوباره یک شرط عبور برای خارج شدن از این Step قرار دهیم. برای قرار

دادن این شرط عبور می‌توانیم از قسمت نوار ابزار گزینه‌ی Transition Condition Between Steps را زده و یا از کلید میانبر F5 استفاده کنیم. این شرط می‌تواند فعال شدن یک فلگ مانند M0 باشد لذا مطابق شکل زیر می‌توانیم این شرط را قرار دهیم.

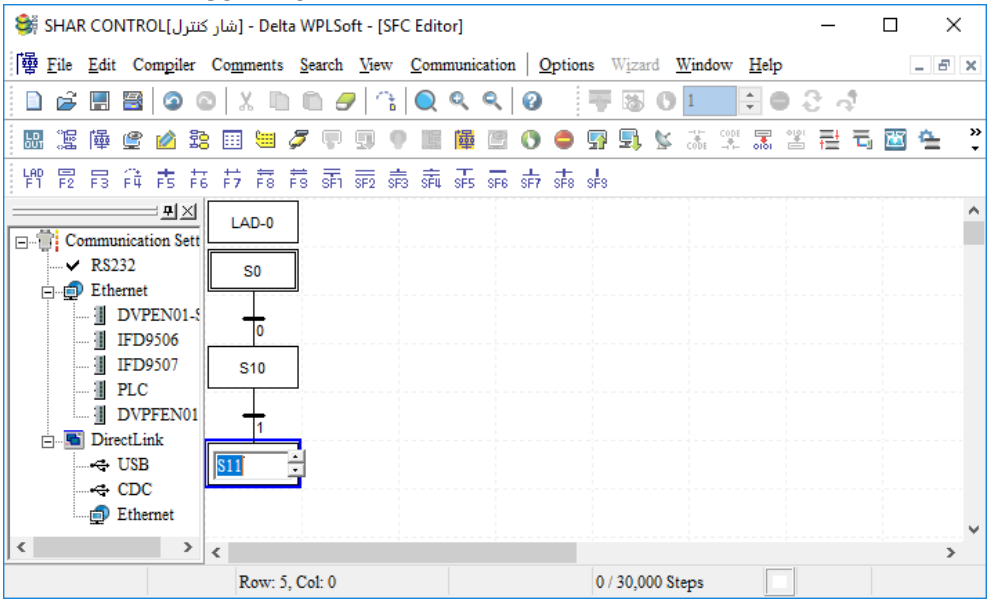


با کلیک بر روی گزینه‌ی Transition Condition Between Steps مطابق شکل بالا شرط دوم نیز به برنامه اضافه می‌شود.

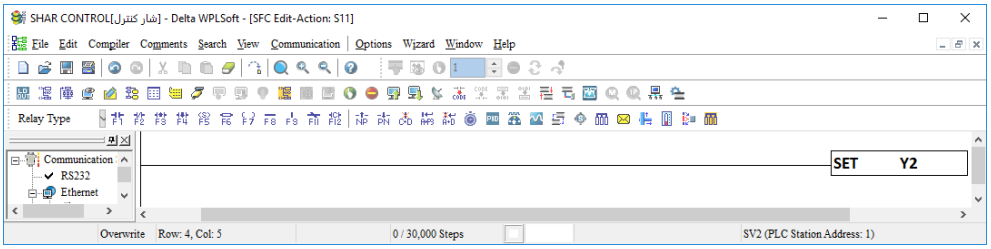
برای تعریف شرط نیز مانند مراحل قبلی باید بر روی این شرط راست کلیک کرده و گزینه‌ی Action/transition editor را انتخاب و شرط فعال شدن فلگ M0 را اعمال کنیم. سپس مطابق شکل صفحه بعد گزینه‌ی close را زده و بعد گزینه yes را می‌زنیم.



به محض تحریک M0 برنامه به Step بعدی خواهد رفت. حال می‌خواهیم ادامه برنامه را با یک دستور دیگر ادامه دهیم. برای این منظور باید در نوار ابزار بالای برنامه از گزینه General Steps و یا از کلید میانبر F3 استفاده کنیم.

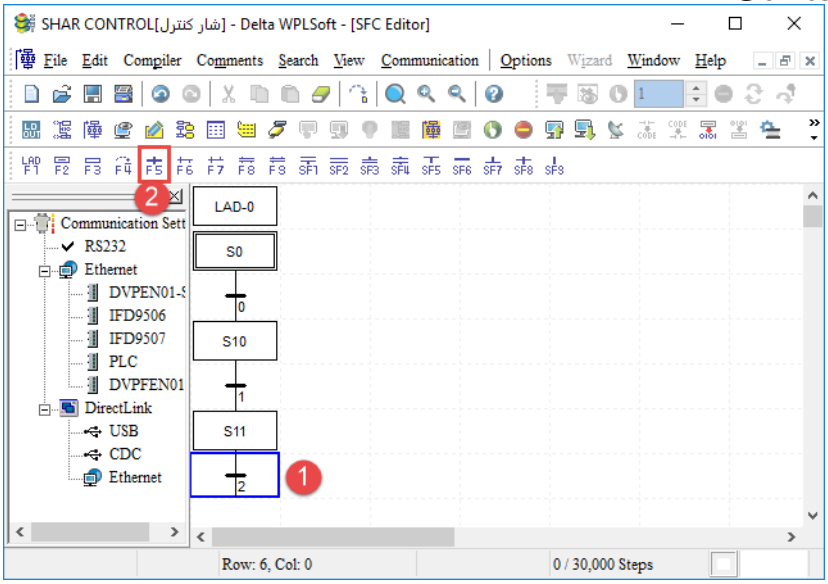


مطابق شکل بالا با انتخاب گزینه General Steps یک مرحله دیگر به برنامه اضافه شده است و ما باید به این مرحله، برنامه مورد نظر خود را اضافه نماییم. برای این کار همانند مراحل قبلی می‌توان با راست کلیک کردن بر روی این Step و انتخاب گزینه‌ی Action/transition editor برنامه‌ی مورد نظر را مطابق شکل زیر نوشت.

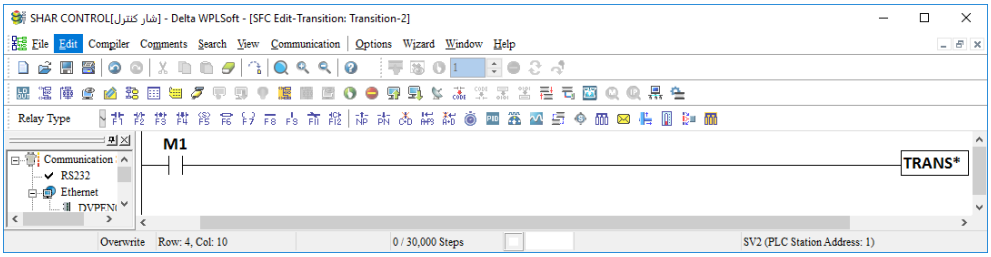


مطابق شکل بالا با رسیدن برنامه به این مرحله ، خروجی Y2 ست خواهد شد و در صورتیکه برنامه از این مرحله بگذرد، این خروجی خاموش نخواهد شد و برای خاموش کردن این خروجی باید در یکی از مراحل بعدی از دستور RST استفاده کنیم.

در این قسمت می‌خواهیم دوباره شرط عبور از این مرحله را تعیین کنیم. برای این کار باید همانند مراحل قبلی از نوار ابزار بالای برنامه از گزینه‌ی Transition Condition Between Steps را انتخاب و یا از کلید میانبر F5 استفاده کنیم.

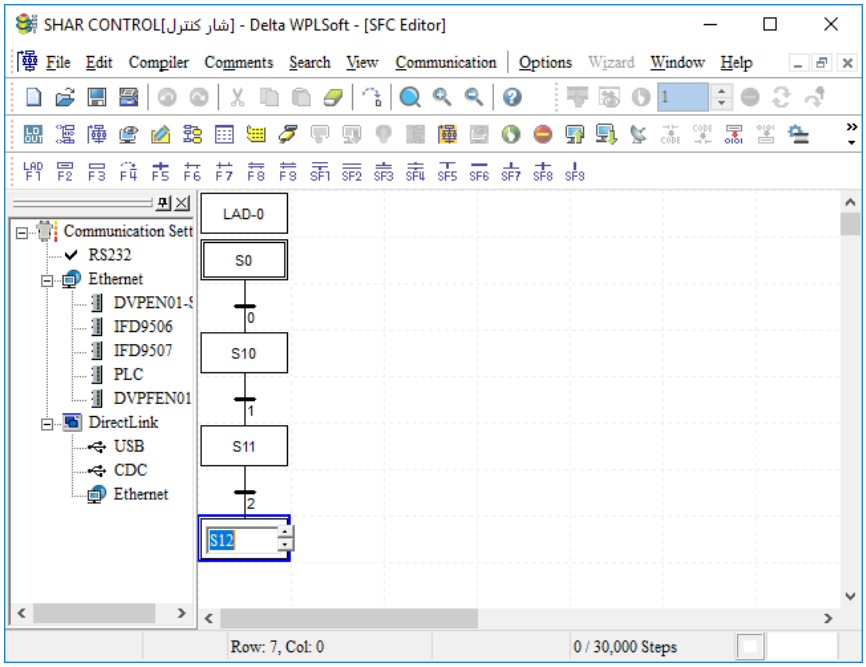


در این مرحله برای اعمال شرط عبور، باید بر روی مرحله شرط که با شماره ۲ نشان داده شده است راست کلیک کرده و با انتخاب گزینه‌ی Action/transition editor در پنجره ی باز شده شرط عبور را اعمال کنیم . مطابق شکل زیر در این قسمت می‌خواهیم از M1 به عنوان شرط سوم استفاده کنیم. یعنی در صورتیکه فلگ M1 فعال شد ، برنامه به مرحله بعد خواهد رفت.

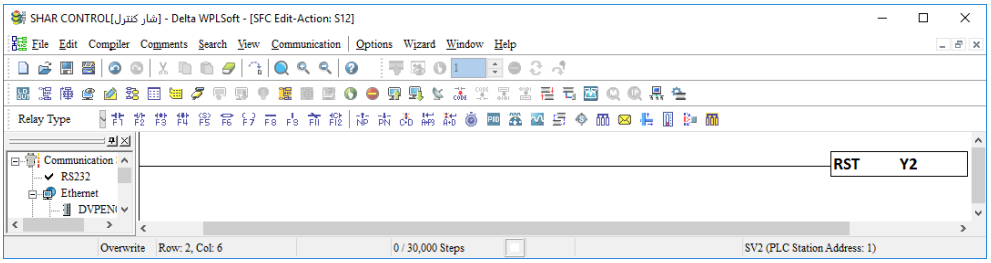


مطابق شکل فوق بعد از اعمال شرط عبور که فعال شدن فلگ M1 می‌باشد ، با زدن close و سپس انتخاب گزینه‌ی Yes به صورت اتوماتیک وارد صفحه برنامه‌ی SFC Editor خواهید شد.

در Step بعدی می‌خواهیم خروجی Y2 که در مرحله قبلی فعال شده بود را غیرفعال کنیم. بدین منظور باید مانند مراحل قبلی و مطابق شکل زیر از نوار ابزار بالای برنامه گزینه‌ی General Steps را انتخاب کرده و یا از کلید میانبر F3 استفاده کنیم.



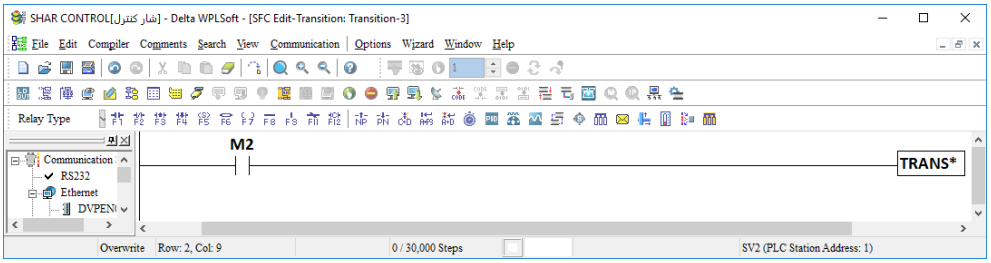
حال در این Step باید برای اعمال دستور ریست Y2 که در مرحله قبلی فعال شده بود، ابتدا روی این Step راست کلیک کرده و گزینه‌ی Action/transition editor را انتخاب کنیم و مطابق شکل زیر در صفحه‌ی باز شده برنامه‌ی مورد نظر را که ریست کردن خروجی Y2 می‌باشد را می‌نویسیم.



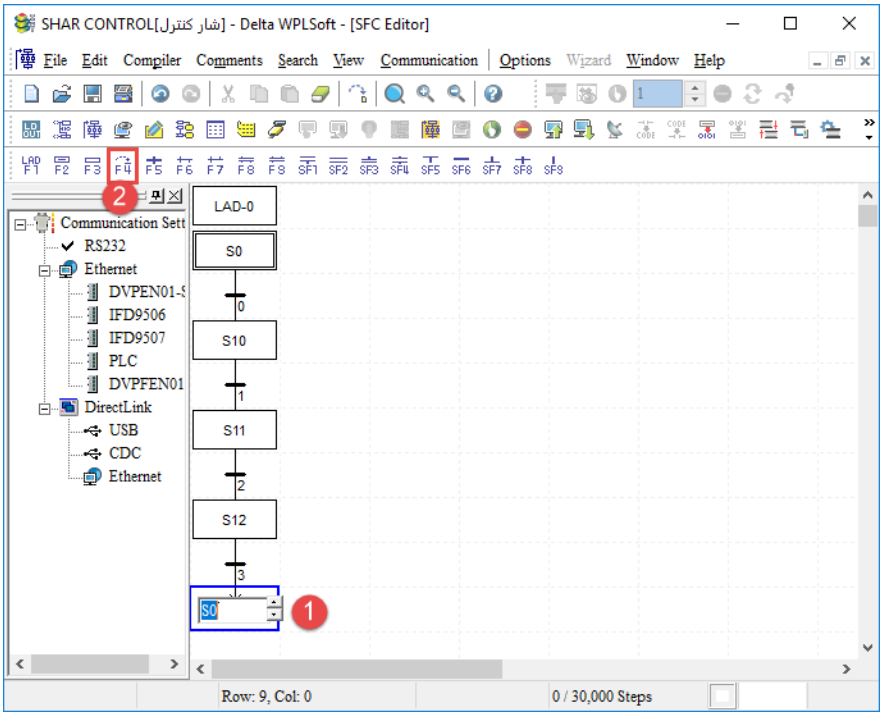
مطابق شکل بالا بعد از نوشتن برنامه جهت ریست کردن خروجی Y2 ، باید گزینه‌ی close را زده و سپس گزینه‌ی Yes را انتخاب کنیم تا برنامه به صورت اتوماتیک وارد صفحه‌ی SFC Editor شود.

در پایان می‌خواهیم یک Step به برنامه اضافه کنیم تا برنامه به مرحله اول رفته و دوباره سیکل برنامه تکرار شود. برای اینکار ابتدا باید یک شرط عبور تعیین کنیم تا بعد از برقرار بودن شرط ، برنامه به مرحله اول باز گردد. لذا بعد از S12 باید از منوی بالای برنامه از گزینه‌ی Transition Condition Between Steps را انتخاب کنیم و یا از طریق کلید میانبر F5 یک شرط را اعمال کنیم. همانند مراحل قبلی بر روی مرحله‌ی شرطی چهارم راست کلیک کرده و گزینه‌ی Action/transition editor را انتخاب می‌کنیم. سپس در صفحه‌ی باز شده مطابق شکل صفحه بعد یک شرط مثلا فعال

شدن M2 را قرار می‌دهیم. بعد از نوشتن دستور شرطی گزینه‌ی close را زده و سپس گزینه‌ی Yes را انتخاب می‌کنیم.



بعد از نوشتن شرط انتقال باید از منوی ابزار بالای نرم افزار گزینه Step Jump را انتخاب کنیم و در ادامه مطابق شکل زیر عمل کنیم.



همانطور که در این تصویر ملاحظه می‌کنید با اضافه کردن یک Step Jump ،شماره‌ی مرحله ای که میخواهید به آن منتقل شوید را از شما درخواست می‌کند در این برنامه می‌خواهیم به مرحله اول(S0) منتقل شویم.

حال باید برنامه را compile کرده و تست کنیم. برای این کار باید گزینه‌ی compile را از منوی ابزار انتخاب کرده و از منوی باز شده، گزینه‌ی SFC==> instruction را انتخاب و یا از کلید میانبر CTRL+F7 استفاده کنیم.

حال برای تست برنامه از طریق شبیه ساز نرم افزار WPLSoft به ترتیبی که در شکل زیر نشان داده شده است عمل می‌کنیم.

۱ – ابتدا گزینه Simulator را فعال می‌کنیم.

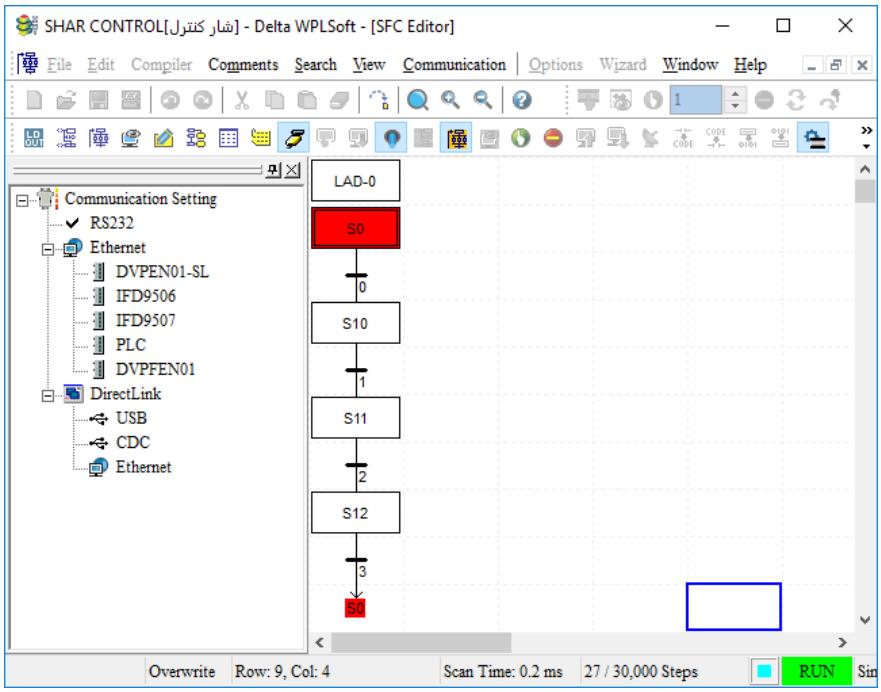
۲ – سپس گزینه Write to PLC را انتخاب می‌کنیم.

۳- با زدن گزینه Run برنامه را اجرا می‌کنیم.

۴ – سپس برنامه را در SFC Monitoring قرار می‌دهیم.

۵ – سپس Online Mode را فعال می کنیم .

۶-درصورت نیاز برای فعال کردن ورودیهای نوع X این گزینه را فعال کنید .



با انجام مراحل بالا مرحله S0 به رنگ قرمز در می‌آید. لذا برنامه در این مرحله(Step) بوده و خروجی Y0 نیز روشن می شود . و تا زمانیکه شرط عبور از Step بر قرار نشود برنامه در این مرحله می ماند.

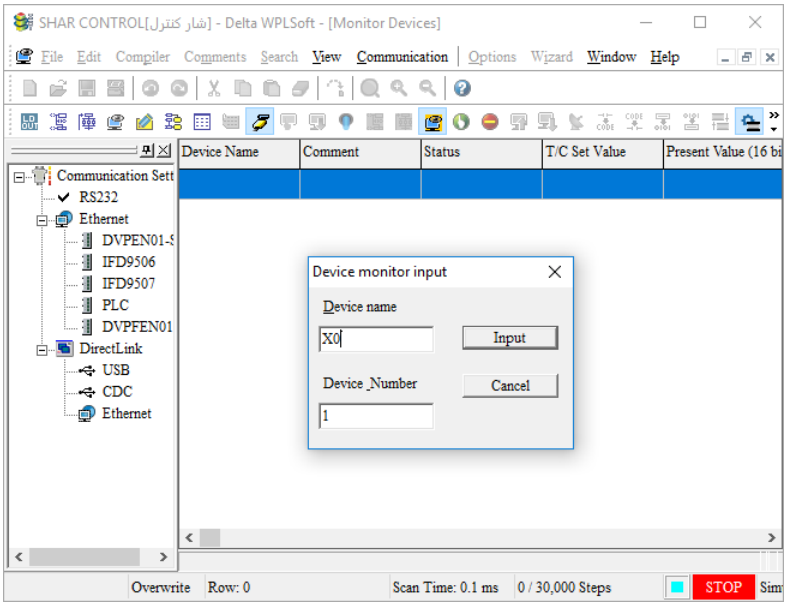
حال با قرار دادن مکان نمای آبی رنگ بر روی مرحله اول (S0) کلیک راست کرده و گزینه Set On را بزنید تا رنگ سلول مربوط به مرحله اول قرمز رنگ شود.

با انجام این مراحل، برنامه در حال اجرا می باشد .برای رفتن برنامه به مرحله بعد (S10) باید شرط اول برقرار شود لذا می‌توانید با قرار دادن مکان‌نمای آبی رنگ بر روی اولین شرط نوشته شده، راست کلیک کرده و گزینه‌ی Action/transition editor را بزنید تا وارد قسمتی شوید که شرط برنامه را مشخص کرده اید. با قرار دادن مکان نمای آبی بر روی شرط اول که X0 می‌باشد ، باید این شرط را یک بار Set On و یک بار Set Off کنید.(با کلیک راست روی المان مورد نظر)

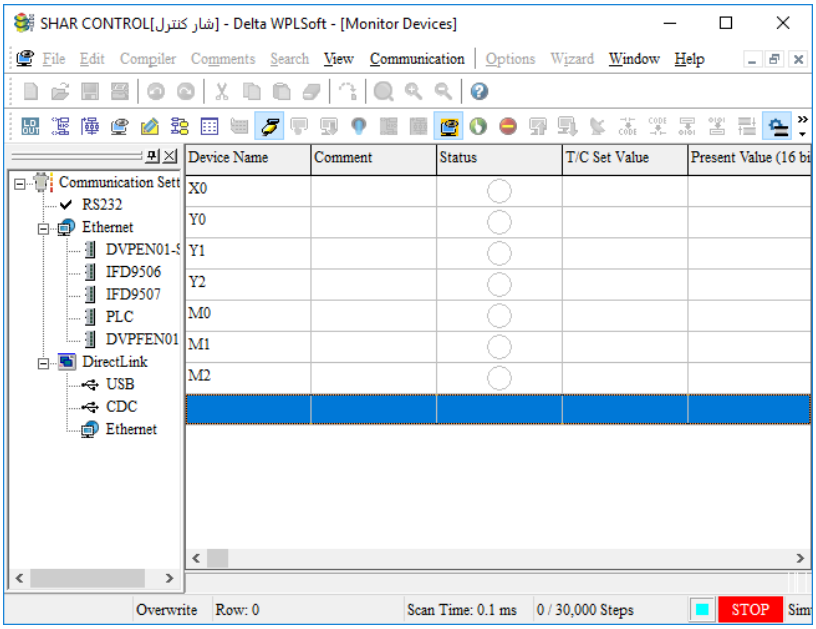
سپس با زدن گزینه‌ی close به صورت اتوماتیک به صفحه‌ی SFC Editor منتقل می‌شوید و مشاهده می‌کنید که مرحله دوم (S10) به رنگ قرمز در آمده و در حال اجراست. به همین ترتیب می‌توانید با تحریک سایر شروط به ترتیب سایر مراحل را اجرا نمایید. در پایان با اجرای شرط آخر ، برنامه به اولین مرحله‌یعنی (S0) باز می‌گردد.

برای مشاهده ی وضعیت خروجی ها در حالت شبیه ساز (Simulator) می‌توان مطابق شکل زیر ابتدا از نوار ابزار گزینه‌ی Simulator را انتخاب کرده و سپس گزینه‌ی Edit Monitored Devices را انتخاب کنیم و در پایان ورودی ها و خروجی ها و شروط را برای برنامه تعریف کنیم.

پس از انتخاب این گزینه مطابق شکل زیر یک صفحه جدید باز خواهد شد که شما باید المان‌های ورودی و خروجی و شرطی را ، طبق برنامه‌ای که نوشته‌اید وارد کنید.



بعد از باز شدن پنجره بالا با زدن دکمه Y یا X یا M و ... یک پنجره‌ی کوچک باز خواهد شد که شما باید شماره‌ی المان‌های بکار رفته و تعداد آنها را در آنجا وارد کرده و روی Input کلیک کرده تا در جدول قرار گیرند بعد از واردکردن کلیه المانهای مورد نیاز برنامه، شکل صفحه قبل بصورت زیر خواهد شد .



در پایان برای تست برنامه باید برنامه را به حالت Online Mode تغییر دهید. برای این کار کافیست از نوار ابزار در بالای صفحه، گزینه‌ی Online Mode را زده و سپس برنامه را Run کنید. سپس در پنجره قبل روی هرکدام از المانهای مورد نظر راست کلیک کرده آن را Set On و یا Set Off نمایید .

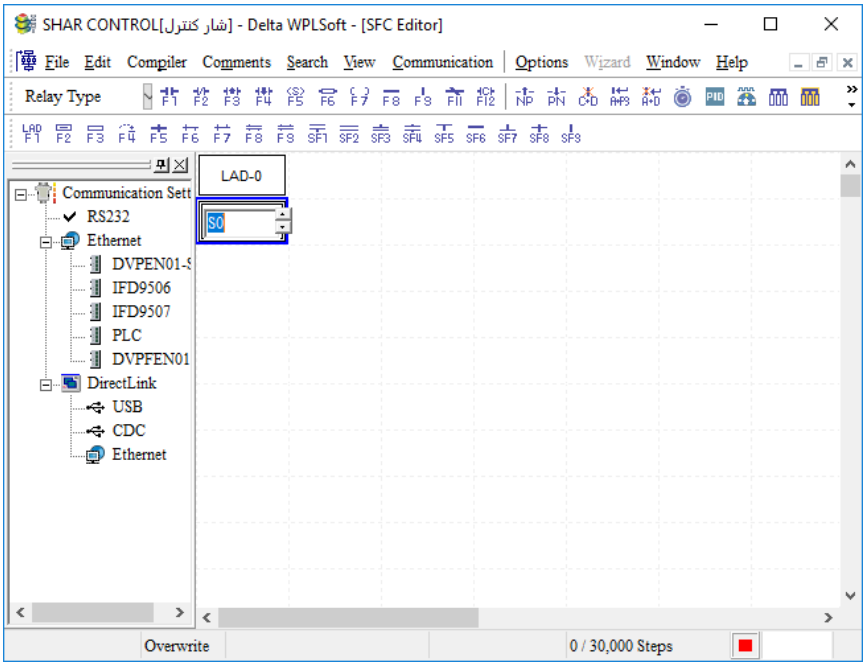
مطابق شکل فوق به محض Run شدن برنامه ، خروجی اول (Y0) فعال می‌شود.

با راست کلیک کردن بر روی شرط اول (X0) و Set On/Set Off کردن، برنامه از مرحله S0 به S10 منتقل و خروجی Y0 غیرفعال و خروجی Y1 فعال خواهد شد. به همین ترتیب می‌توان با تحریک سایر شروط، اجرای مرحله‌ای برنامه را مشاهده کرد.

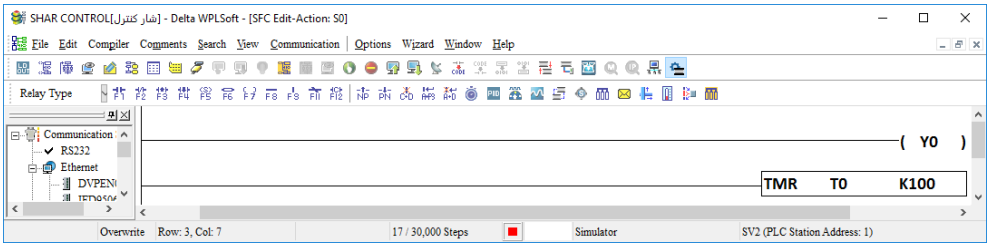
نکته : اگر تمام شرطهای برنامه را در حالت On نگه داریم واضح است که برنامه بصورت سریع از اولین Step تا آخرین Step اجرا و مرتب تکرار خواهد شد.

در ادامه می‌خواهیم در قالب یک مثال با استفاده از تایمرها یک فرایند اتوماتیک و تکراری را برنامه ریزی کنیم.

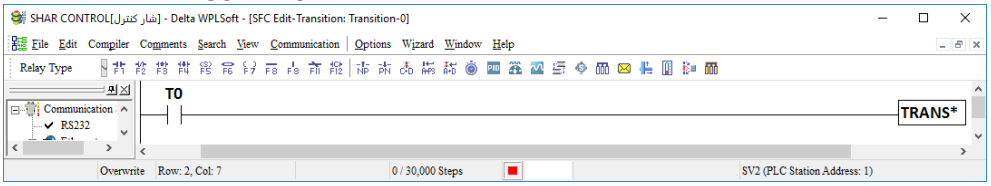
مانند مثال قبل وارد محیط برنامه نویسی به زبان SFC می‌شویم و در مرحله اول از قسمت نوار ابزار برنامه بر روی Initial Step کلیک کرده و یا از طریق کلید میانبر F3 یک Step اولیه به برنامه اضافه می‌کنیم.



با راست کلیک بر روی این Step و انتخاب گزینه‌ی Action/transition editor می‌توان برنامه‌ی مورد نظر را مطابق شکل زیر نوشت. در این Step می‌خواهیم یک تایمر به مدت ۱۰ ثانیه زمان بگیرد .

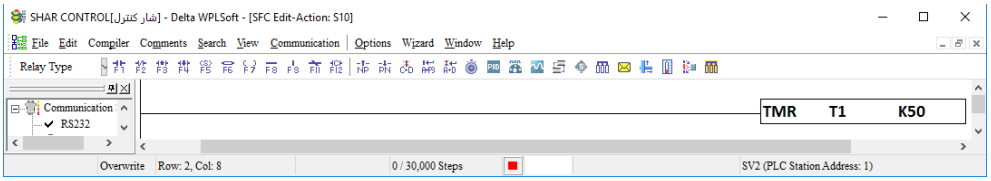


مطابق شکل بالا بعد از نوشتن برنامه‌ی مورد نظر همانند مثال قبل گزینه Close را زده و سپس گزینه Yes را میزنیم. بعد از این مرحله باید یک شرط عبور برای رفتن به Step بعدی تعیین کنیم لذا از نوار ابزار گزینه‌ی Transition Condition Between Steps را انتخاب می‌کنیم و یا می‌توانیم از طریق کلید میانبر F5 یک شرط اعمال کنیم. با زدن این گزینه یک شرط به برنامه اضافه خواهد شد می‌توانیم با راست کلیک کردن بر روی این Step و زدن گزینه‌ی Action/transition editor ، شرط مورد نظر را اعمال کنیم. در این قسمت شرط مورد نظر، تیغه‌ی تایمر در مرحله‌ی S0 می‌باشد.



پس از اعمال شرط همانند شکل بالا ابتدا گزینه‌ی Close و سپس گزینه‌ی Yes را میزنیم تا شرط در برنامه اعمال شده و برنامه به صورت اتوماتیک به صفحه‌ی SFC Editor منتقل شود.

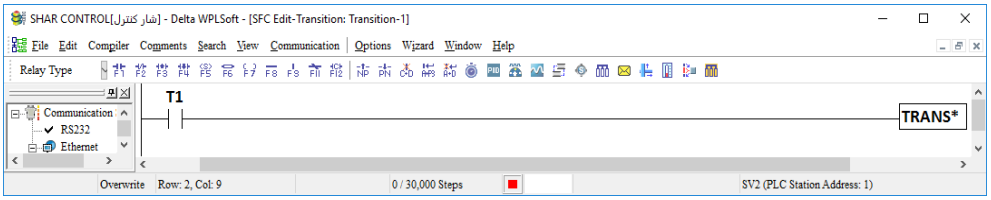
در این قسمت می‌خواهیم پس از برقراری شرط که فعال شدن تیغه تایمر T0 بوده ، برنامه وارد مرحله بعدی شود. بنابراین باید از قسمت نوار ابزار یک مرحله عمومی (General Step) به برنامه اضافه کنیم و مانند مراحل قبل با راست کلیک کردن بر روی Step جدید و زدن گزینه‌ی Action/transition editor برنامه‌ی مورد نظر را بنویسیم. در این مرحله می‌خواهیم از یک تایمر استفاده کنیم تا به مدت 5 ثانیه زمان بگیرد .



در این مرحله نیز مانند مراحل قبلی پس از نوشتن برنامه مورد نظر ابتدا گزینه‌ی Close و سپس گزینه‌ی Yes را میزنیم تا برنامه ثبت شده و به صورت اتوماتیک به صفحه‌ی SFC Editor منتقل شویم.

پس از نوشتن این مرحله می‌خواهیم یک شرط را برای رفتن به مرحله بعدی قرار دهیم. این شرط همان فعال شدن تیغه‌ی تایمر T1 خواهد بود. از نوار ابزار برنامه گزینه‌ی Transition Condition Between Steps را انتخاب و یا می‌توانیم از طریق کلید میانبر F5 یک شرط را اعمال کنیم.

حال با راست کلیک کردن بر روی این شرط و انتخاب گزینه‌ی Action/transition editor می‌توانیم شرط فعال شدن تیغه تایمر T1 را اعمال کنیم.



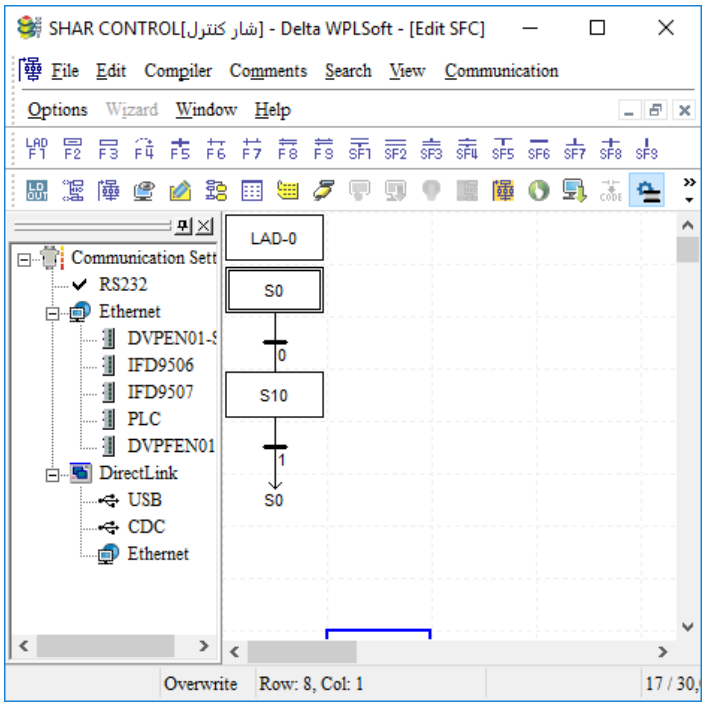
در این قسمت نیز همانند مراحل قبلی ابتدا گزینه‌یClose و سپس گزینه‌ی Yes را انتخاب کرده تا برنامه‌ی شرطی، اعمال گردد و به صورت اتوماتیک به محیط برنامه نویسی SFC باز گردیم.

پس از انجام این مراحل می‌خواهیم برنامه به اولین Step بازگردد. در این صورت با Run شدن برنامه، روی S0 کلیک راست کرده و Set ON را بزنید خروجی Y0 به مدت ۱۰ ثانیه روشن و به مدت ۵ ثانیه

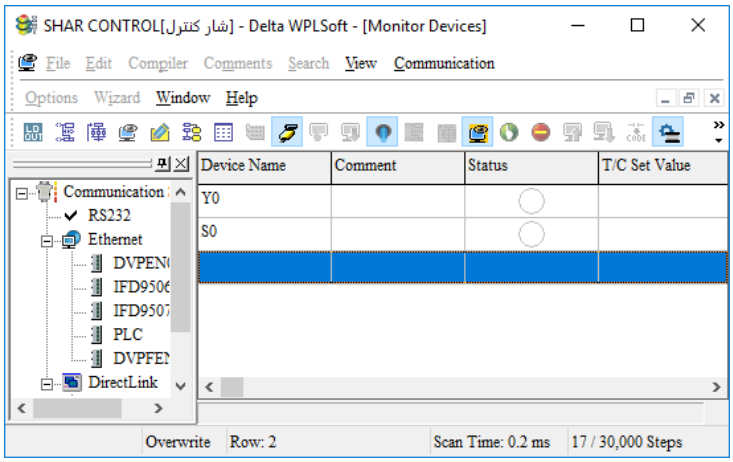
خاموش خواهد بود و این سیکل مرتب تکرار خواهد شد . برای توقف، مجددا روی S0 کلیک راست کرده Set Off را بزنید . چنانچه بخواهید بدون نیاز به این کار و با RUN شدن، برنامه اجرا شود باید مانند مثال قبل در قسمت LAD-0 برنامه‌ای مانند مثال قبل بنویسیم .

برای بازگشت به اولین Step می‌توانیم از قسمت نوار ابزار گزینه‌ی Step Jump را انتخاب کنیم و یا از طریق کلید میانبر F4 به این گزینه دسترسی داشته باشیم. با انتخاب این گزینه ، مرحله ای که می خواهید به آن بازگردید از شما پرسیده می‌شود که در این برنامه ما می‌خواهیم به اولین Step یا همان (S0) بازگردیم.

نکته : از یک تایمر نمی‌توان در دو Step متوالی استفاده کرد و حتما باید یک Step مابین دو مرحله ای که از یک تایمر مشابه استفاده شده است وجود داشته باشد.



حال برنامه، برای تست با استفاده از شبیه ساز (Simulator) آماده می‌باشد. برای این منظور همانند مثال اول عمل می‌کنیم.



مطابق شکل بالا با Set On کردن S0 خروجی Y0 روشن شده و شروع به چشمک زدن می‌کند.

دستورات Set و Out :

این دو دستور برای انتقال اجرای برنامه از یک مرحله به مرحله‌ی بعدی و یا فعال کردن یک المان مانند Y0 مورد استفاده قرار می‌گیرند. همچنین عمل انتقال در یک Sequence یا توالی و یا از یک Sequence به Sequence دیگر انجام گیرد. این دو دستور فرقهایی با یکدیگر دارند که در ادامه توضیح داده خواهد شد.

دستور Set Sn :

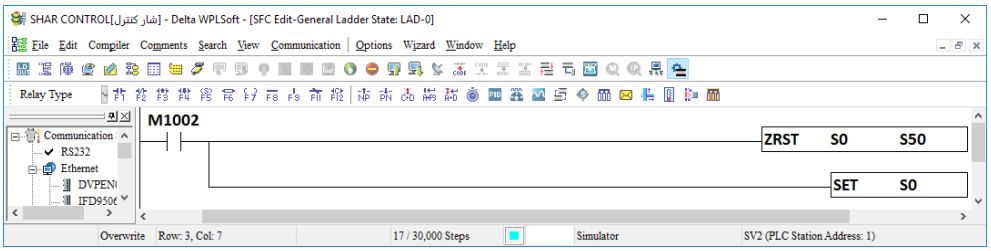
این دستورمطابق شکل برای فعال کردن یک Step جدید در Sequence جاری مورد استفاده قرار میگیرد و بعد از انتقال به مرحله جدید، کلیه‌ی خروجیهای مرحله قبلی غیر فعال می‌شوند. در واقع این دستور برای انتقال برنامه از یک Step به Step بعدی مورد استفاده می‌شود. برای درک بهتر این دو دستور می‌خواهیم ابتدا یک برنامه‌ی ساده به زبان SFC نوشته و سپس این برنامه را به زبان Ladder Step تبدیل کرده و بررسی کنیم.

مثال ۱۰۳ :

می‌خواهیم به محض Run شدن PLC ، سه خروجی Y0 و Y1 و Y2 فعال شده و یک تایمر نیز وارد مدار شود و مدت ۳ ثانیه را زمان بگیرد. پس از ۳ ثانیه برنامه به مرحله‌ی بعدی رفته و سه خروجی دیگر Y3 و Y4 و Y5 فعال شوند و یک تایمر دیگر مدت ۳ ثانیه زمان بگیرد. پس از سپری شدن مدت زمان ۳ ثانیه ای، برنامه دوباره به مرحله اول باز گردد و این حلقه ادامه داشته باشد.

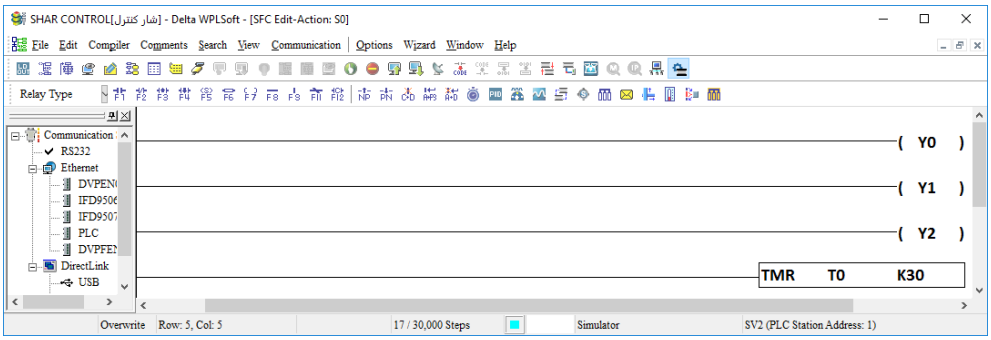
ابتدا برنامه‌ی WPLSoft را باز کرده و پس از ایجاد پروژه‌ی جدید وارد محیط برنامه نویسی به زبان SFC می‌شویم.

در اولین مرحله از قسمت نوار ابزار گزینه‌ی LAD-0 را انتخاب می‌کنیم. سپس با راست کلیک کردن بر روی آن و انتخاب گزینه‌ی Action/transition editor برنامه مورد نظر را مطابق شکل زیر نوشته و سپس گزینه‌ی Close را زده و پس از آن گزینه‌ی Yes را انتخاب می‌کنیم.



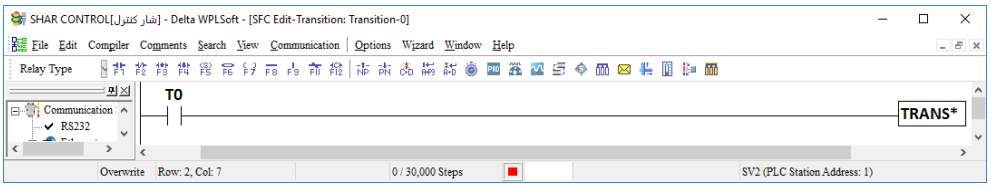
نکته : در صورتیکه اطمینان دارید در برنامه هیچ Step فعالی ندارید می‌توانید خط مربوط به ریست کردن Step ها را ننویسید. پس از انجام این مرحله از نوار ابزار بالای صفحه، گزینه‌ی Initial Step را انتخاب کرده و یا از کلید میانبر F2 استفاده می‌کنیم.

سپس می توانیم با راست کلیک کردن بر روی آن و انتخاب گزینه‌ی Action/transition editor برنامه‌ی مورد نظر را که فعال شدن سه خروجی و یک تایمر می‌باشد را مطابق شکل زیر بنویسیم.

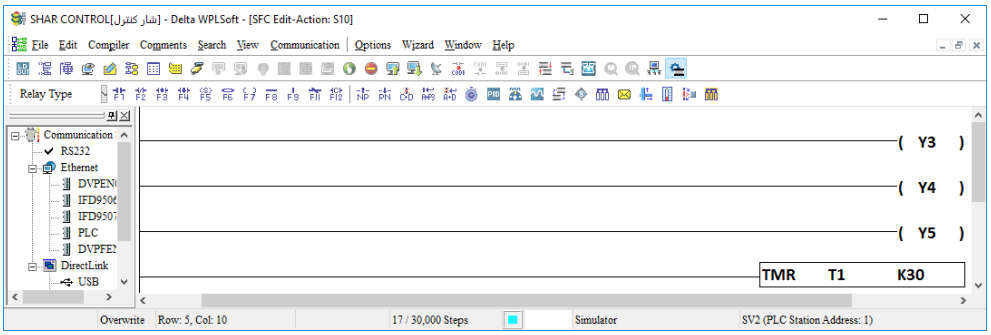


پس از نوشتن برنامه، گزینه‌ی Close را زده و سپس گزینه‌ی Yes را می‌زنیم تا برنامه ثبت شده و به صورت اتوماتیک وارد محیط برنامه نویسی SFC شویم.

در مرحله بعد باید یک شرط عبور برای انتقال به مرحله‌ی بعد تعیین کنیم لذا از نوار ابزار، گزینه‌ی Transition Condition Between Steps را انتخاب کرده و با راست کلیک کردن بر روی شرط و انتخاب گزینه‌ی Action/transition editor ، شرط را که فعال شدن تیغه‌ی تایمر T0 است را مطابق شکل در برنامه می‌نویسیم.

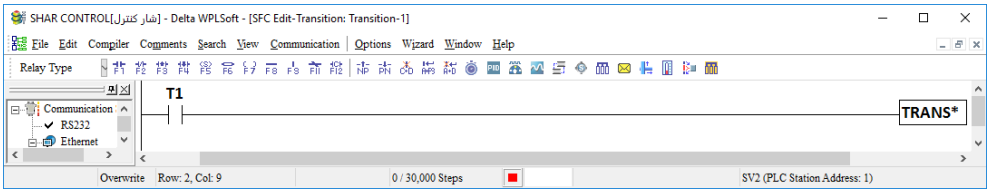


در ادامه باید برنامه مورد نظر در Step بعدی را بنویسیم لذا از نوار ابزار گزینه‌ی General Step را انتخاب کرده و یا از کلید میانبر F3 را میزنیم تا یک مرحله عمومی یا همان General Step به برنامه اضافه شود. سپس با راست کلیک کردن بر روی این Step و انتخاب گزینه‌ی Action/transition editor می‌توانیم برنامه‌ی مورد نظر را که روشن شدن Y3 و Y4 و Y5 و فعال شدن تایمر T1 می‌باشد را مطابق شکل صفحه بعد بنویسیم.



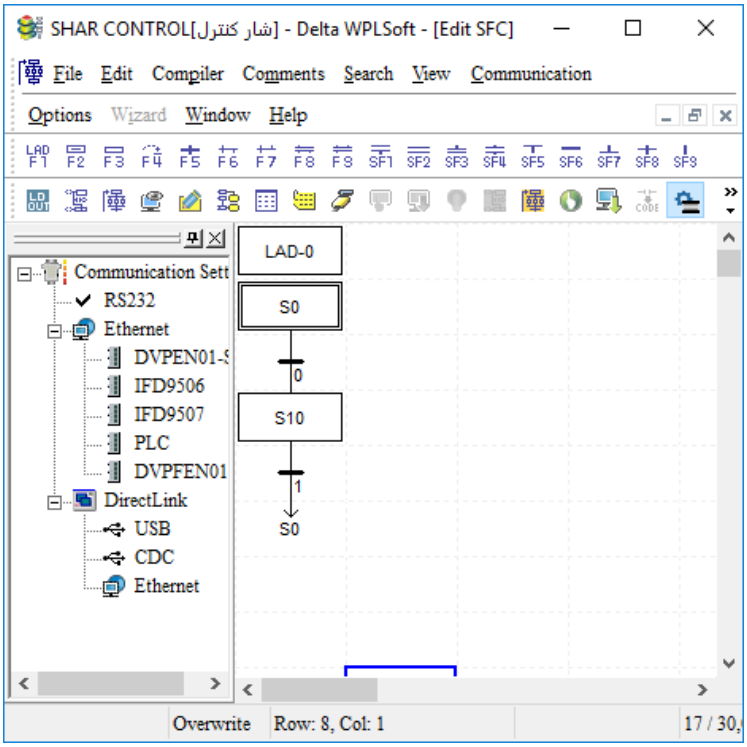
پس از نوشتن برنامه، ابتدا گزینه‌ی Close را زده و سپس گزینه‌ی Yes را می‌زنیم تا برنامه ثبت شده و به صورت اتوماتیک به محیط برنامه نویسی SFC بازگردیم.

حال باید شرط عبور از این مرحله به مرحله بعد را به برنامه‌ی خود اضافه کنیم. لذا باید از نوار ابزار گزینه‌ی Transition Condition Between Steps را انتخاب کرده و با راست کلیک کردن بر روی شرط و انتخاب گزینه‌ی Action/transition editor ، شرط را که فعال شدن تیغه‌ی تایمر T1 است را مطابق شکل زیر در برنامه بنویسیم.

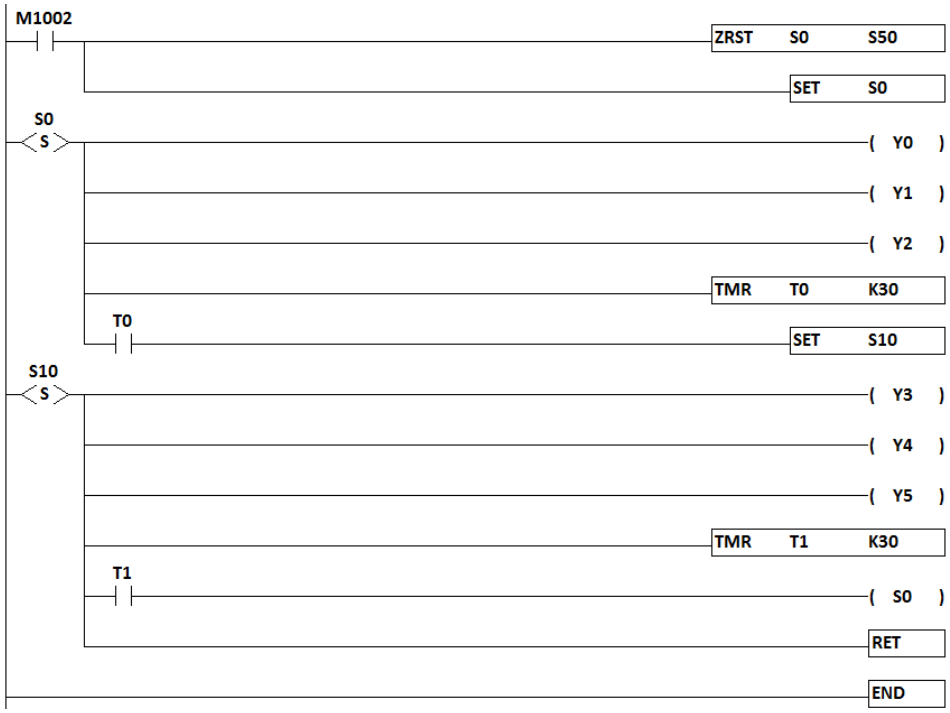


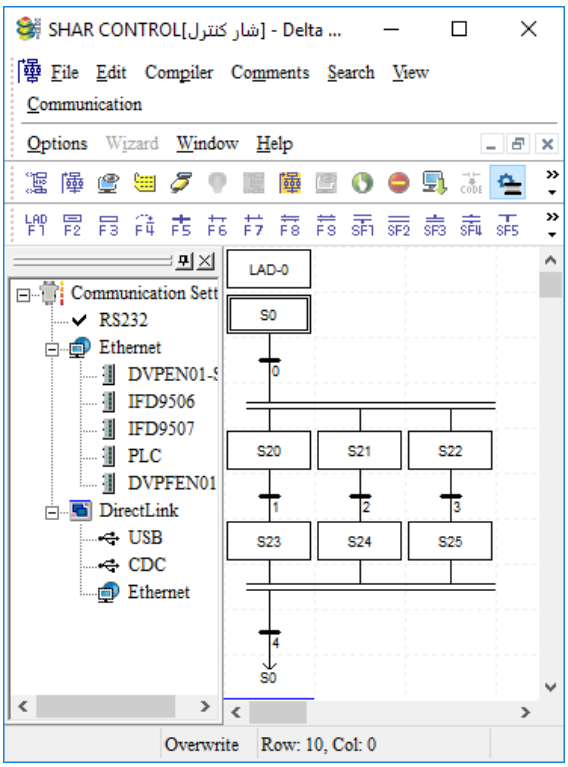
پس از نوشتن برنامه و اعمال شرط مورد نظر، گزینه‌ی Close را زده و سپس گزینه‌ی Yes را می‌زنیم تا برنامه ثبت شده و به صورت اتوماتیک وارد محیط برنامه نویسی SFC شویم.

در این مرحله پس از برقراری شرط برنامه، باید به اولین Step که همان S0 است بازگردد لذا از نوار ابزار، گزینه‌ی Step Jump را زده و یا از کلید میانبر F4 استفاده کنیم. در این مرحله کافیست شماره‌ی Step مورد نظر را مطابق شکل صفحه بعد S0 انتخاب کنیم.



برنامه‌ی مورد نظر به پایان رسیده و می‌توانید همانطور که در مثال قبلی گفته شد برنامه را بصورت شبیه سازی شده تست کنید. حال می‌خواهیم به بررسی برنامه‌ی نوشته شده در زبان Step Ladder بپردازیم. با Compile کردن برنامه و انتخاب زبان Ladder Diagram ، مطابق شکل صفحه بعد برنامه‌ی نوشته شده به زبان Step Ladder را می‌توانید مشاهده کنید.



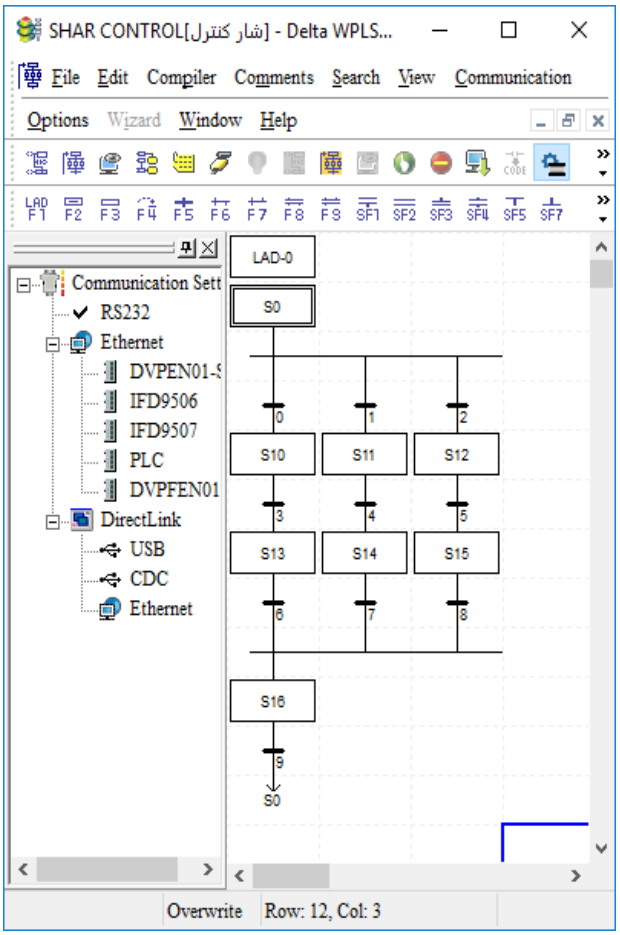


شاخه های موازی همزمان

Simultaneous Divergence

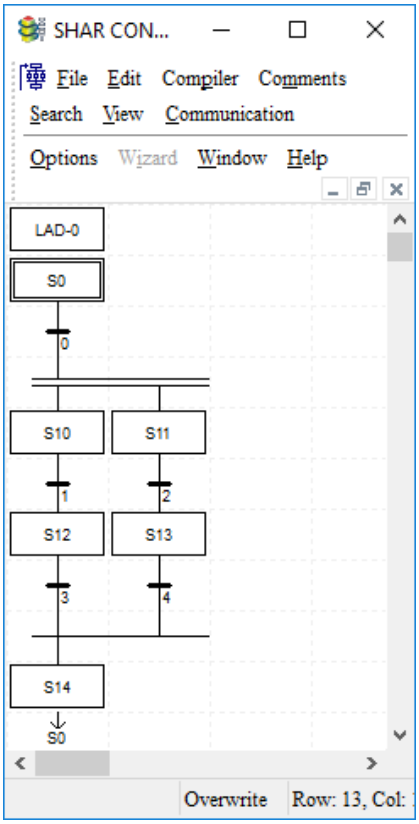
مطابق شکل زیر در صورتیکه شرط انتقال اول برقرار شود ، برنامه از S0 به S20 و S21 و S22 انتقال پیدا می‌کند.

نکته : برنامه در صورتی از به انتهای برنامه می رسد که مراحل S23 و S24 و S25 انجام شده و شرط شماره 4 برقرار شده باشد.



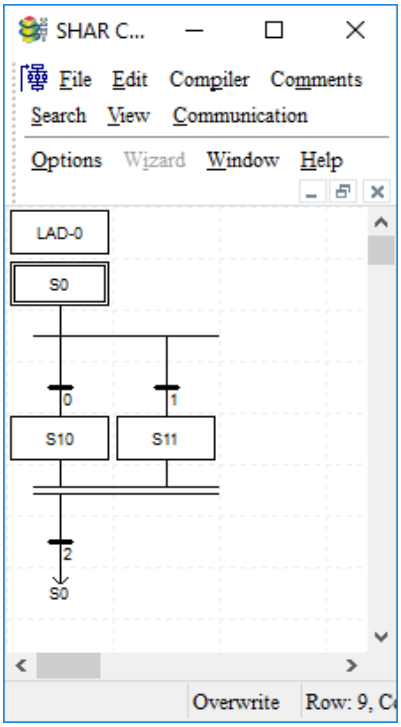
شاخه ها ی موازی غیر همزمان Alternative Divergence

مطابق شکل هر شرط که زودتر محقق شود، برنامه وارد همان Step و یا شاخه می‌شود. به عنوان مثال اگر شرط شماره‌ی 1 زودتر فعال شود اجرای برنامه وارد همان شاخه می‌شود. حال اگر شرط های دیگر 0 و 2 فعال شوند ، شاخه هایشان اجرا نخواهند شد. با اجرا شدن برنامه در شاخه‌ی مربوط به شرط شماره 1 پس از اینکه شرط شماره 7 برقرار شود، ادامه برنامه به Step16 منتقل خواهد شد.



نکته : شاخه های موازی ترکیبی نیز می‌توان نوشت. مطابق شکل که ابتدا از شاخه موازی همزمان استفاده شده و با تحریک شدن شرط 0 برنامه بطور همزمان وارد شاخه های موازی و Step های 10 و 11 می‌شود. در ادامه اگر مثلا شرط شماره 1 زودتر محقق شود اجرای برنامه وارد Step شماره 12 خواهد شد.در حالیکه در شاخه ی دیگر اجرای برنامه در همان Step شماره‌ی 11 متوقف شده است لذا با فعال شدن شرط شماره 3 برنامه وارد Step شماره‌ی 14 خواهد شد. در نظر داشته باشید اگر در دور بعدی اجرای برنامه، شرط شماره‌ی 2 زودتر محقق شود، اجرای برنامه وارد Step شماره 13 خواهد شد و اجرای برنامه در شاخه ی مجاور در همان Step شماره 10 متوقف شده و با برقرار شدن شرط شماره‌ی 4 ، برنامه به Step شماره 14 انتقال خواهد یافت.

در شکل صفحه بعد برای اینکه عملکرد برنامه صحیح باشد باید هر دو شرط شماره‌ی 0 و 1 هم برقرار باشند تا برنامه وارد Step های 10 و 11 شود. حال اگر شرط شماره‌ی 2 نیز فعال شود اجرای برنامه وارد Step شماره 12 خواهد شد.



نکته : در یک چنین ساختاری اگر فقط یک ورودی مثلا شرط شماره‌ی 0 برقرار شود ، برنامه به Step شماره‌ی 10 وارد می‌شود. حال چون خروجی Step شماره 10 از شاخه ی موازی همزمان استفاده شده و از طرفی شرط شماره‌ی 1 برقرار نشده و برنامه‌ی Step شماره‌ی 11 اجرا نشده، لذا برنامه در همین جا متوقف خواهد شد چون شرط اجرای هر دو Step شماره های 10 و 11 انجام نشده است.

نکته : با فعال شدن فلگ M1034 تمامی خروجیها غیر فعال می‌شوند.

نکته : در صورتیکه از Step Point های Latch در برنامه استفاده شود با قطع برق ، Step های فعال در حافظه ذخیره می شوند و با وصل مجدد برق ، Step هائیکه قبلا فعال بوده اند مجددا فعال می‌شوند.

این Step های پایدار (Latch) در هر مدل از CPU های دلتا دارای شماره‌ی معینی می‌باشند. برای مثال شماره Step Point های PLC مدل 14SS2 طبق فایل راهنمای این PLC به شکل زیر است.

S	Step point	Initial step point	S0~S9, 10 points, (*2)	Total 1024 points
		Zero point return	S10~S19, 10 points (use with IST instruction), (*2)	
		Latched	S20~S127, 108 points, (*2)	
		General	S128~S911, 784 points, (*1)	
		Alarm	S912~S1023, 112 points, (*2)	

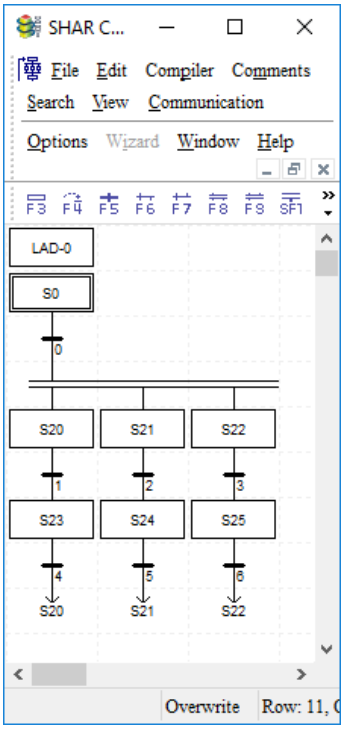
نکته : برای استفاده از Step Point های آلارم می‌توانید از دستور ANS استفاده کنید.

مثال ۱۰۴ :

می‌خواهیم چندین موتور را بطور کاملا مستقل روشن و خاموش کنیم به طوری که هر موتور برای خودش شستی استارت و استپ داشته باشد و بتواند مستقل از موتورهای دیگر کنترل شود.

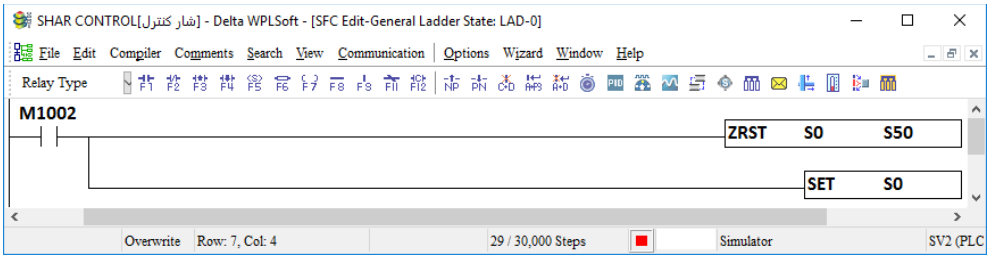
شرح برنامه : بعد از Run شدن PLC ، و بعد از گذشت ۳ ثانیه، برنامه از مرحله S0 وارد سه مرحله مجزای S20 و S21 و S22 بصورت موازی و همزمان می‌شود. در این سه Step هیچ برنامه ای وجود ندارد و بنابراین خروجی های Y0 و Y1 و Y2 نیز غیر فعال می‌باشند. در ادامه و در زیر هر Step یک شرط قرار داده شده که تحریک ورودی های X1 (استارت موتور اول) و X2 (استارت موتور دوم) و X3 (استارت موتور سوم) می‌باشند . با تحریک هر کدام از این شروط، برنامه به Step بعدی در همان

شاخه می‌رود این Step ها به ترتیب S23 (فعال شدن موتور اول) و S24 (فعال شدن موتور دوم) و S25 (فعال شدن موتور سوم) می‌باشند. در ادامه‌ی هر شاخه یک شرط دیگر وجود داشته که عبارتند از: X3 (خاموش شدن موتور اول) و X4 (خاموش شدن موتور دوم) و X5 (خاموش شدن موتور سوم). با برقراری این شروط، برنامه از Step های S23 و S24 و S25 به Step های S20 . S21 و S22 منتقل می‌شوند و از آنجایی‌که با عبور از یک Step به Step بعدی ، خروجیهایی که به صورت Out نوشته شده‌اند غیر فعال می‌شوند ، موتورهای اول ، دوم و سوم ، خاموش خواهند شد.

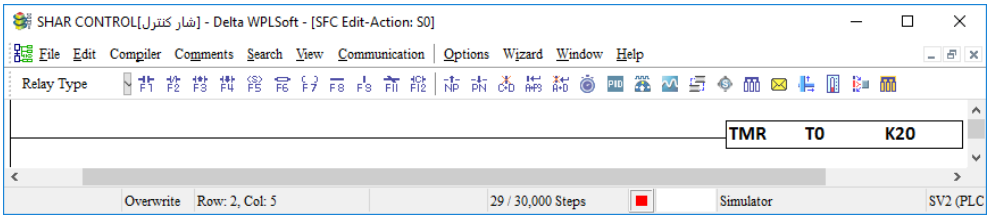


حال به بررسی برنامه‌هایی که باید در هر Step نوشته شود می‌پردازیم

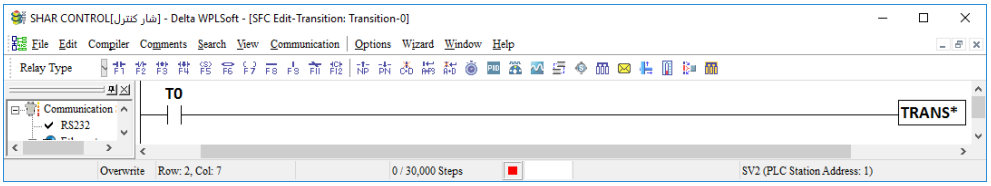
۱ – LAD-0 :



۲ – Initial Step :

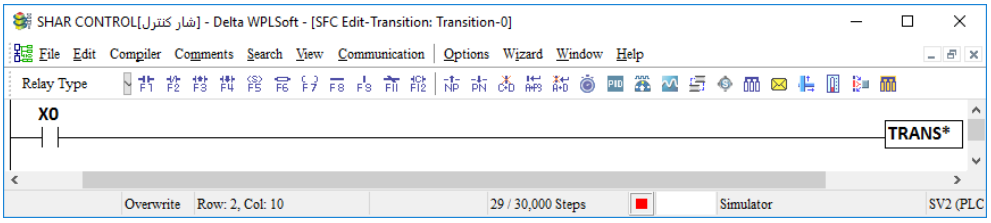


۳ – Transition Condition Between Steps (شرط شماره 0) :

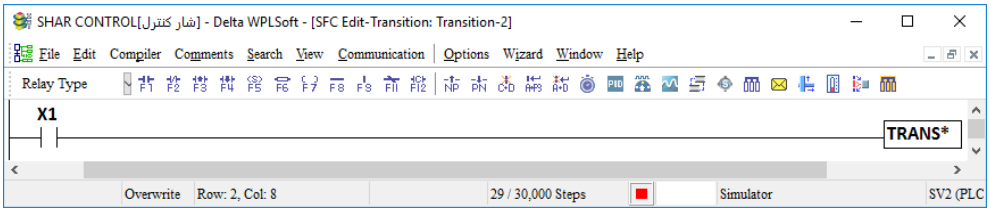


۴ – برای Step های شماره 20 و 21 و 22 نیازی به نوشتن برنامه نیست.

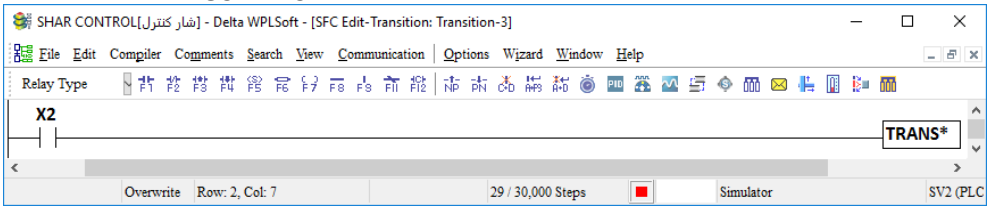
۵ – Transition Condition Between Steps (شرط شماره 1) :



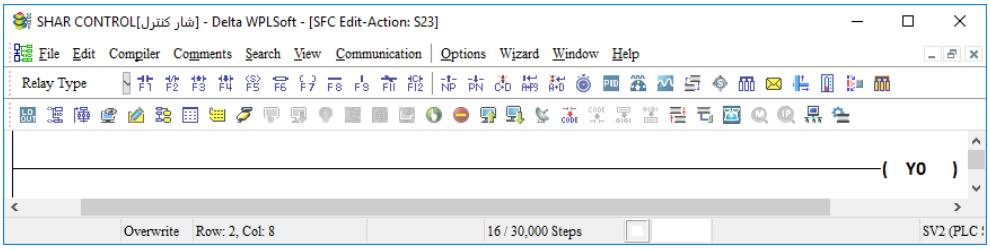
۶ – Transition Condition Between Steps (شرط شماره 2) :



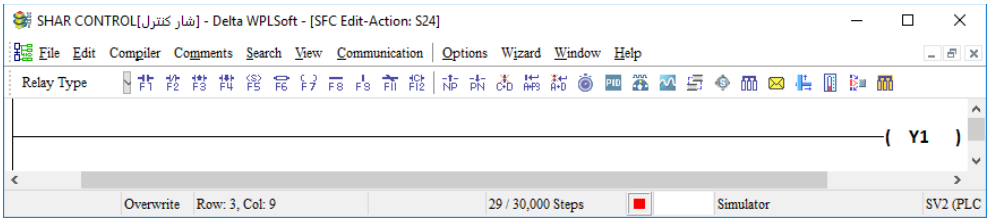
۷ – Transition Condition Between Steps (شرط شماره 3) :



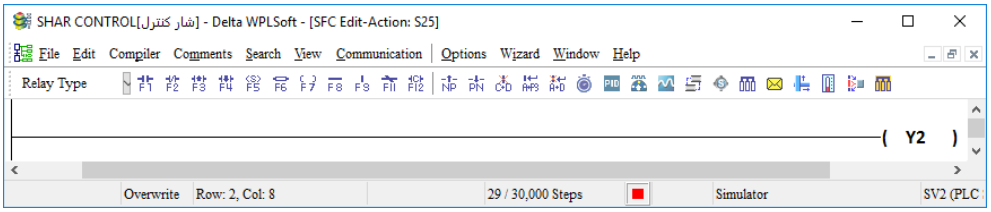
۸ – General Step (مرحله شماره 23) :



۹ – General Step (مرحله شماره 24) :



۱۰ – General Step (مرحله شماره 25) :



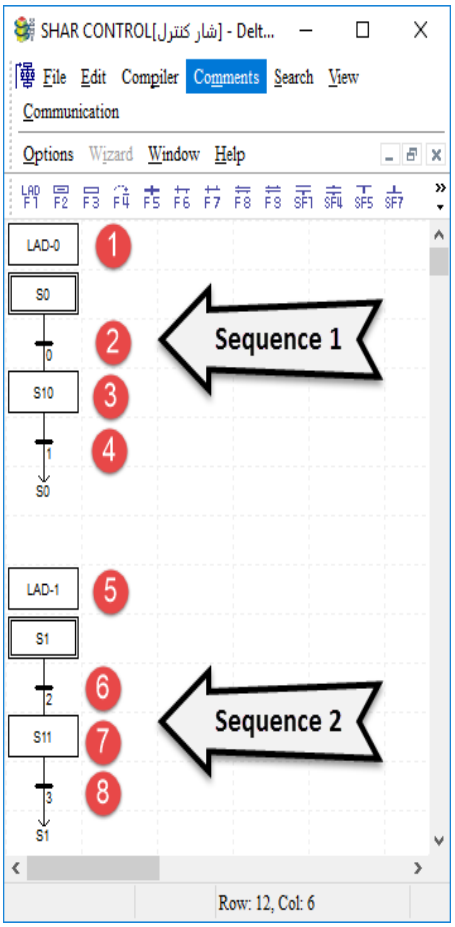
نکته : ورودی های حفاظتی مانند بی متال و یا کنترل فاز و ... باید توسط یک Sequence مجزا برنامه ریزی شوند تا هرگاه اتفاقی رخ داد فورا در برنامه واکنش نشان دهد.

Multiple Sequence :

فرض کنید می‌خواهیم دو فرایند به طور کاملا مستقل تحت کنترل باشند. در این صورت نمی‌توان از یک Sequence استفاده کرد بلکه باید برای هر فرایند یک Sequence مستقل ایجاد و برنامه ریزی کرد.

در مثال زیر دو موتور داریم که می‌خواهیم به طورکاملا مستقل تحت کنترل قرار گیرند لذا مطابق شکل صفحه بعد در دو Sequence برنامه را می‌نویسیم.

حال مطابق مثالهای قبل برنامه هر Step و شرط های عبور را مطابق توضیحات صفحه بعد می‌نویسیم.



LAD-0 :

۱– در این قسمت تمامی Step point ها از شماره‌ی S0 تا S50 بصورت گروهی ریست شده تا در صورت لزوم در برنامه استفاده شوند. همچنین S0 برای استفاده به عنوان Step اصلی در ابتدای برنامه Set شده است.

۲ – در این قسمت از شرط ورودی X0 استفاده شده تا با فعال شدن این شرط برنامه به مرحله بعد انتقال پیدا کند.

نکته : شرط در زبان SFC می‌تواند یک ورودی ، یک خروجی ، یک تایمر ، یک کانتر و یا یک حافظه باشد.

۳ – در این قسمت یک ورودی فعال شده است.

نکته : در صورتیکه در یک مرحله یک خروجی به صورت Out اجرا شود ، با رفتن برنامه به مرحله بعدی ، این خروجی غیر فعال می‌شود و در صورتیکه یک خروجی در یک مرحله به صورت Set برنامه نویسی شود ، با رفتن برنامه به مرحله بعد این خروجی همچنان فعال می‌ماند و غیر فعال شدن آن مستلزم استفاده از دستور RST می‌باشد.

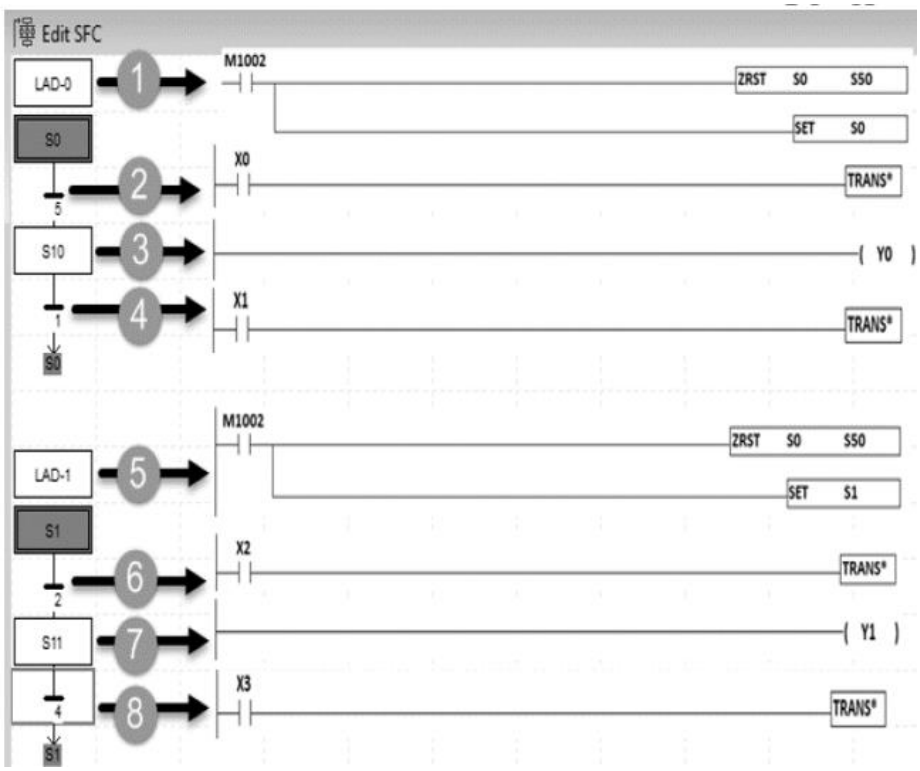
۴ – استفاده از ورودی X1 به عنوان شرط عبور

۵ - در این قسمت تمامی Step point ها از شماره‌ی S0 تا S50 بصورت گروهی ریست شده تا در صورت لزوم در برنامه استفاده شوند. همچنین S1 برای استفاده به عنوان Step اصلی در ابتدای قسمت دوم برنامه Set شده است.

۶ – استفاده از ورودی X2 به عنوان شرط عبور

۷ – فعال کردن خروجی Y1 ۸ – استفاده از ورودی X3 به عنوان شرط عبور از این مرحله

نمای کلی برنامه صفحه قبل بصورت زیر می باشد .



فلگ‌های مهم در زبان SFC :

M1031 : در صورت فعال شدن این حافظه کلیه‌ی ناحیه‌ی غیر Latch پاک می‌شود.

(D-C-M-S-Y-T)

M1032 : در صورت فعال شدن این حافظه کلیه‌ی ناحیه‌ی Latch پاک می‌شود.(D-C-M-S-Y-T)

M1033 : زمانی که PLC از حالت Run به حالت Stop برود خروجیها Latch خواهند شد.

M1034 : کلیه‌ی خروجیها غیر فعال می‌شوند.

تشریح عملکرد دستور IST جهت کنترل فرایند برنامه در زبان SFC :

این دستور باعث افزایش قابلیت عملکرد برنامه در زبان SFC می‌شود

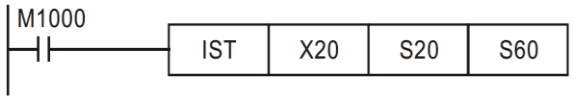
API	Mnemonic	Operands	Function
60	IST	(S) (D1) (D2)	Initial State

OP \ Type	Bit Devices				Word devices										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S	*	*	*												
D_1				*											
D_2				*											

این دستور برای کنترل سیکل کاری در زبان SFC و برای کنترل STEP ها بکار می‌رود و بسیار حائز اهمیت می‌باشد . با استفاده از این دستور می‌توان کنترل STEP را یا بصورت MANUAL و یا بصورت AUTOMATIC انجام داد به عنوان مثال می‌توانیم در هر STEP سیکل‌کاری را متوقف کرد و یا از ابتدا سیکل را آغاز کرد و یا کاری کرد که سیکل‌کاری فقط برای یکبار و یا بطور مداوم اجرا شود .

تشریح پارامترهای دستور IST :

بطور مثال فرض کنید دستور فوق را بصورت زیر نوشته ایم



در این مثال از X20 الی X27 توسط دستور فوق اشغال می‌شود اگراز X0 استفاده میکردیم در این صورت از X0 الی X7 توسط دستور اشغال می شد .

معرفی عملکرد پارامترهای دستور IST :

X20 : مد دستی (MANUAL OPERATION)

X21 : برگشت به نقطه صفر (ZERO RETURN)

X22 : حرکت مرحله به مرحله (STEP OPERATION)

X23 : اجرای فقط یک سیکل (ONE CYCLE OPERATION)

X24 : حرکت مداوم و پیوسته (CONTINIUOUS OPERATION)

X25 : استارت برگشت به نقطه صفر (ZERO RETURN START)

X26 : استارت (START SWITCH)

X27 : استپ (STOP SWITCH)

S20 شروع STEP های مورد استفاده در برنامه می‌باشد.

S60 در این مثال پایان STEP های مورد استفاده می‌باشد.

نکته : نهایتا 1024 تا STEP می‌تواند مورد استفاده قرار بگیرد .

نکته مهم :

زمانیکه از دستور IST استفاده می‌کنیم STEP های S0 مخصوص عملیات دستی Manual می‌باشد و S1 مخصوص عملیات برگشت به حالت صفر (ZERO POINT RETURN OPERATION) بوده و S2 مخصوص عملیات اتوماتیک (AUTOMATIC OPERATION) می‌باشد .

نکته مهم :

زمانیکه از دستور IST استفاده می‌شود STEP های S10 الی S19 توسط سیستم رزرو می‌شود لذا نباید از آنها استفاده کرد .

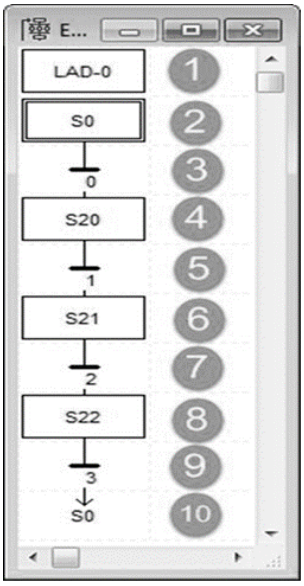
فلگ های مهم در استفاده از دستور IST :

M1040 : ممانعت از حرکت بین STEP ها (Movment Inhibited)

در صورتیکه این فلگ فعال باشد عملیات برنامه در هر step ای که باشد متوقف می‌شود حتی اگر شرط عبور بعد از step برقرار باشد .

M1041 : ادامه اجرای عملیات بین step ها بعد از اینکه متوقف شده باشد .

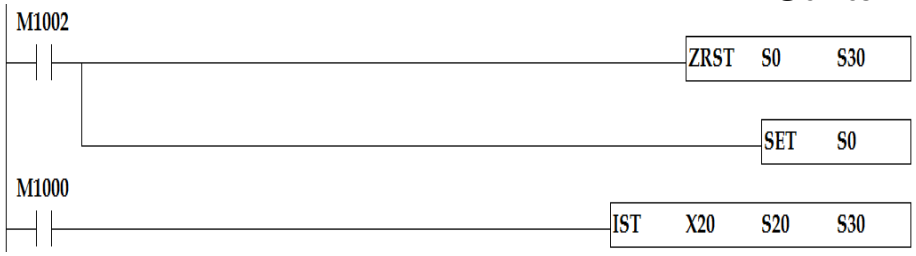
مثال ۱۰۵ :



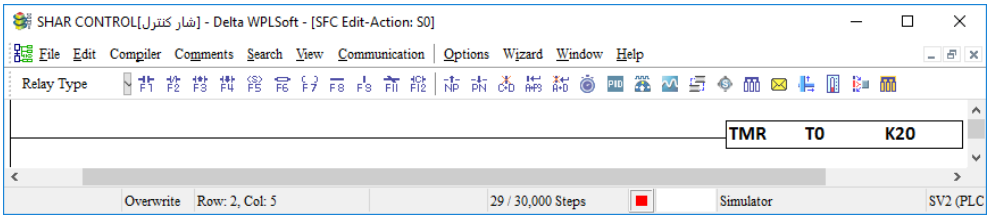
استفاده از دستور IST در مد MANUAL :

ابتدا در محیط SFC فلوچارتی مطابق شکل روبرو ایجاد کرده، سپس مطابق اعداد راهنما و توضیحات زیر عمل می‌کنیم

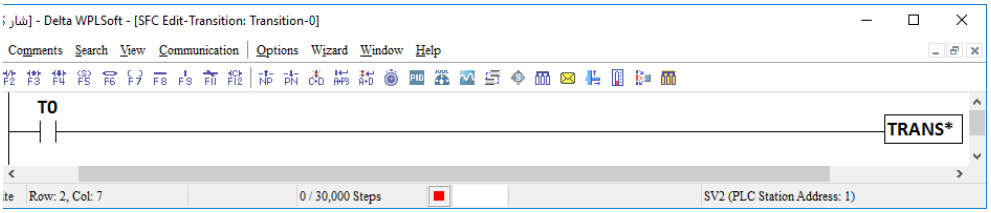
۱ – در قسمت 1 روی LAD-0 کلیک راست کرده گزینه Action / Transition را انتخاب و سپس برنامه صفحه بعد را وارد می‌کنیم و بعد از وارد کردن برنامه ، پنجره مربوطه را بسته و در پیغام ظاهر شده، گزینه‌ی yes را میزنیم.



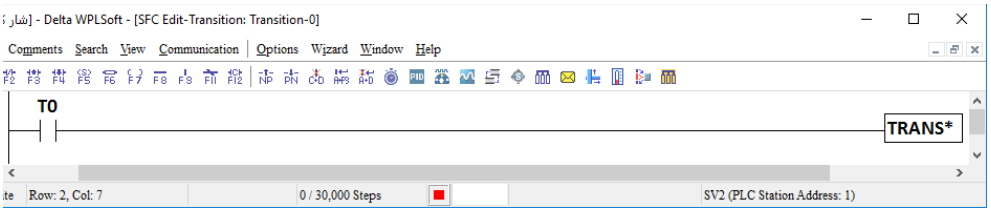
۲- در قسمت 2 روی S0 کلیک راست کرده گزینه Action /Transition را انتخاب می‌کنیم. به عنوان مثال برنامه زیر را وارد می‌کنیم در پایان پنجره مربوطه را بسته و پیغام ظاهر شده را yes میزنیم.



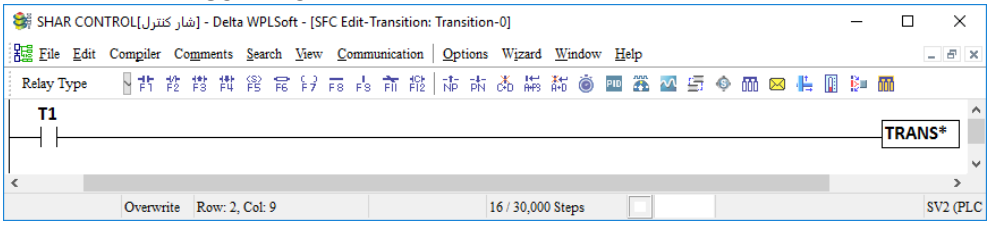
۳- در قسمت 3 مطابق بالا عمل کرده و برنامه زیر را وارد می‌کنیم در پایان پنجره مربوطه را بسته و پیغام ظاهر شده را yes میزنیم .



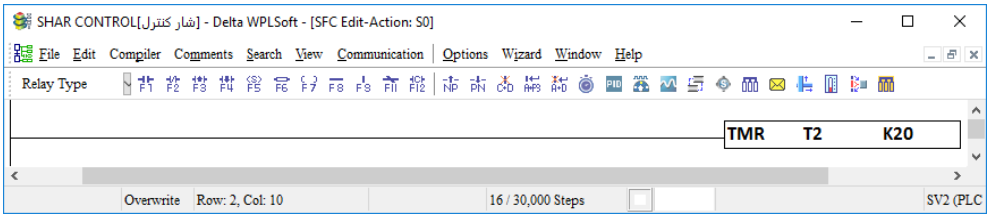
۴ - در قسمت 4 روی S20 کلیک راست کرده گزینه Action /Transition را انتخاب می‌کنیم به عنوان مثال برنامه زیر را وارد می‌کنیم در پایان پنجره مربوطه را بسته و پیغام ظاهر شده را yes میزنیم



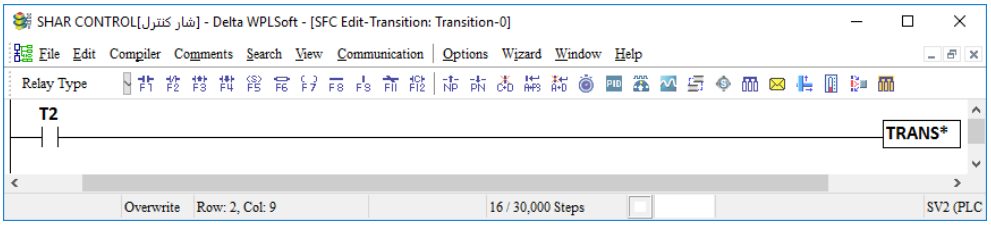
۵ - در قسمت ۵ مطابق بالا عمل کرده و برنامه زیر را وارد می‌کنیم در پایان پنجره مربوطه را بسته و پیغام ظاهر شده را yes میزنیم .



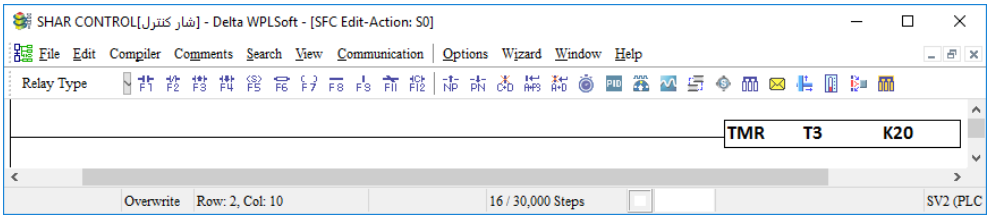
۶- در قسمت ۶ مطابق بالا عمل کرده و برنامه زیر را وارد می‌کنیم در پایان پنجره مربوطه را بسته و پیغام ظاهر شده را yes میزنیم .



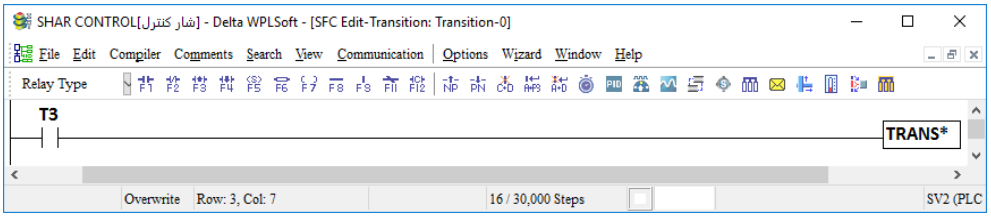
۷ - در قسمت ۷ مطابق بالا عمل کرده و برنامه زیر را وارد می‌کنیم در پایان پنجره مربوطه را بسته و پیغام ظاهر شده را yes میزنیم .



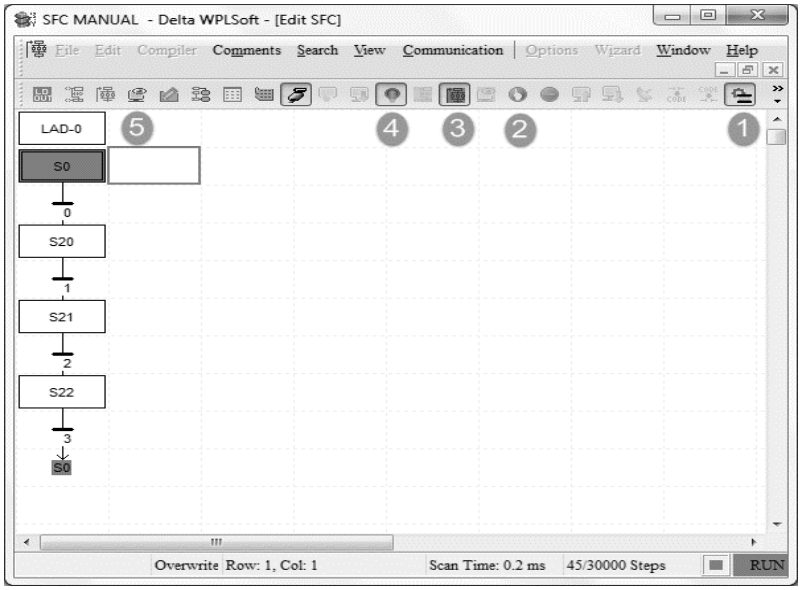
۸ - در قسمت ۸ مطابق بالا عمل کرده و برنامه زیر را وارد می‌کنیم در پایان پنجره مربوطه را بسته و پیغام ظاهر شده را yes میزنیم .



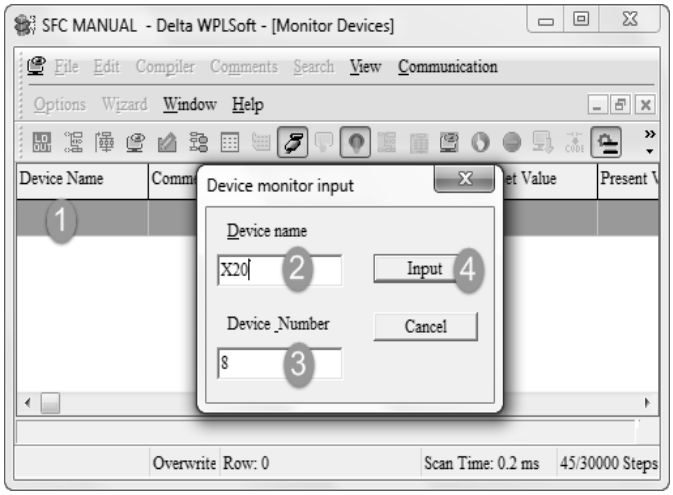
۹ - در قسمت ۹ مطابق بالا عمل کرده و برنامه زیر را وارد می‌کنیم در پایان پنجره مربوطه را بسته و پیغام ظاهر شده را yes میزنیم .

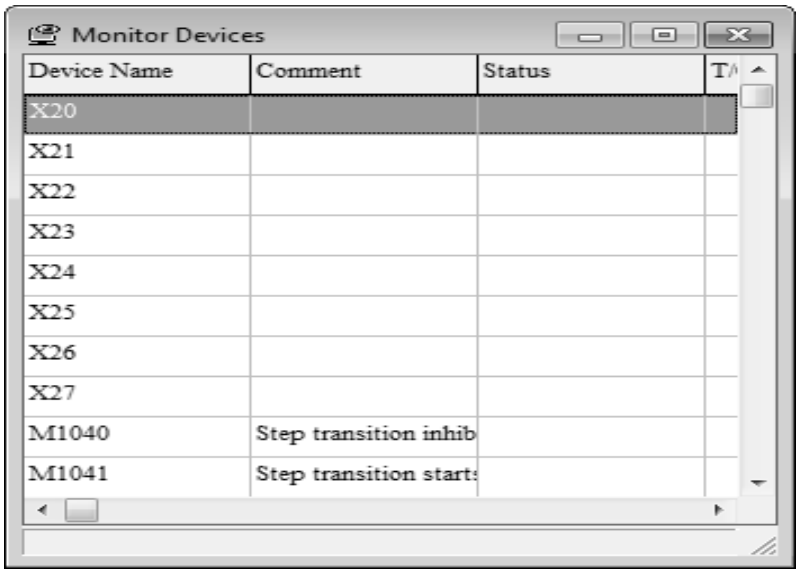


۱۰ – در قسمت ۱۰ شکل صفحه قبل S0 را وارد می‌کنیم و سپس برنامه را کامپایل می‌کنیم. مطابق شکل صفحه بعد، سیمولاتور برنامه را فعال کرده و برنامه را درون آن دانلود و سپس مانیتورینگ را فعال می‌کنیم. برای شبیه سازی و مانیتورینگ برنامه، مطابق اعداد مشخص شده در تصویر زیر عمل می‌کنیم

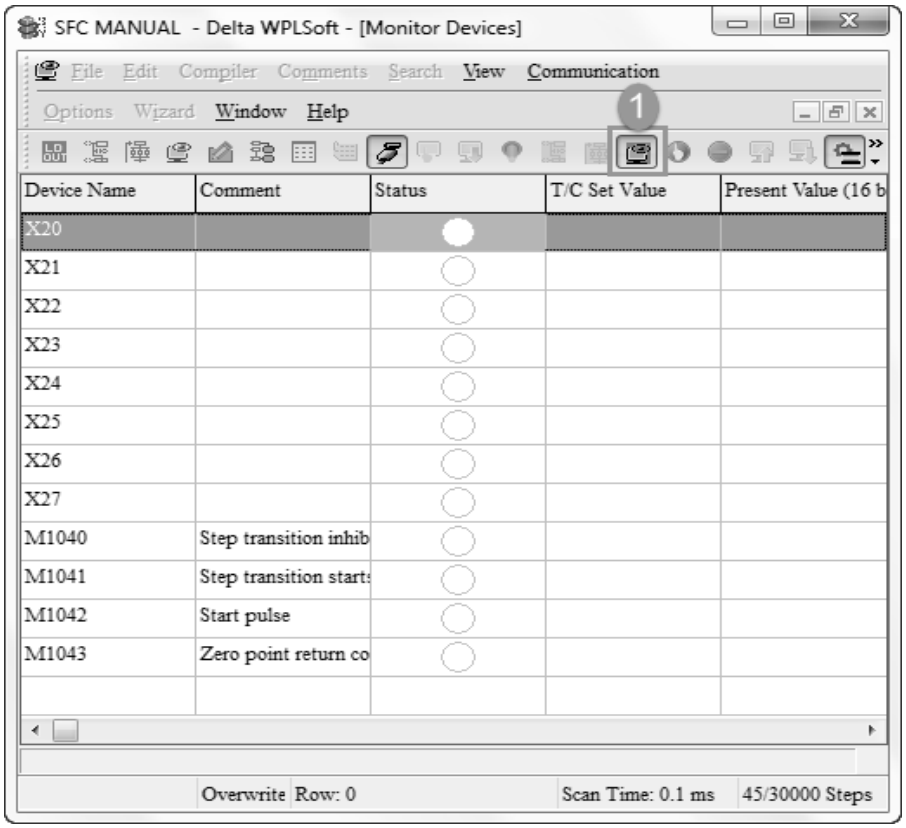


مطابق اعداد راهنمای شکل زیر در قسمت ۱ دبل کلیک کرده و در پنجره باز شده در قسمت ۲ عبارت X20 را وارد کرده و در قسمت ۳ عدد ۸ را وارد می‌کنیم و در نهایت INPUT را کلیک می‌کنیم . مجددا در قسمت پایین جدول دبل کلیک کرده و فلگ‌های M1040 و M1041 و M1042 و M1043 را وارد می‌کنیم شکل کلی بصورت صفحه بعد ظاهر می‌شود.

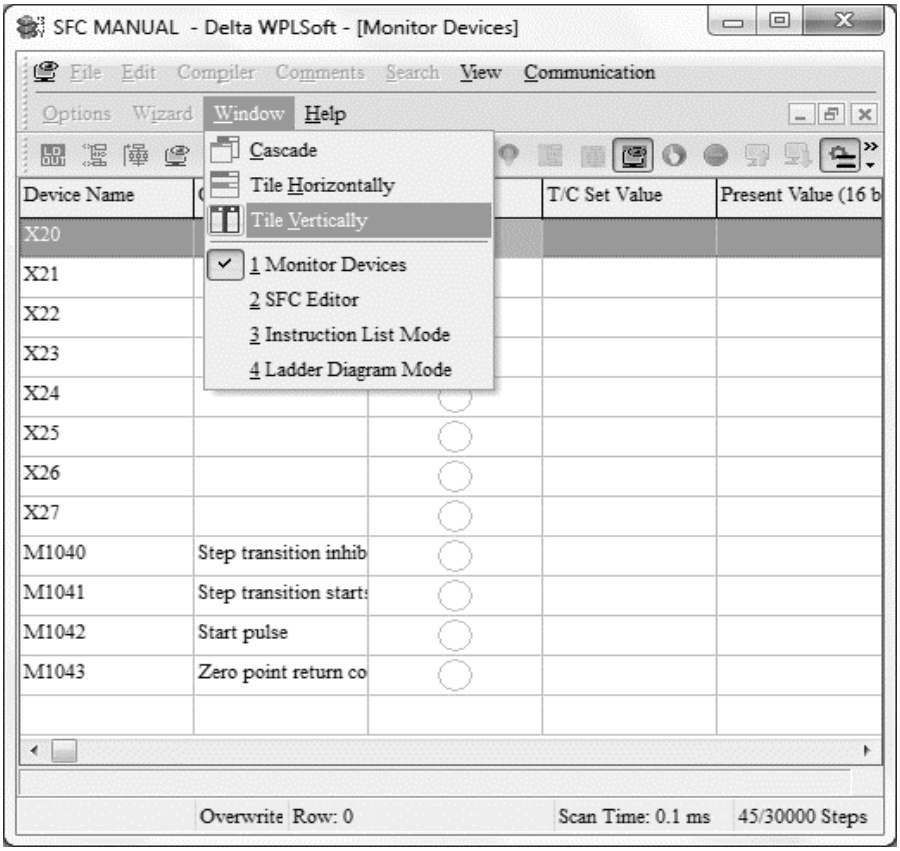




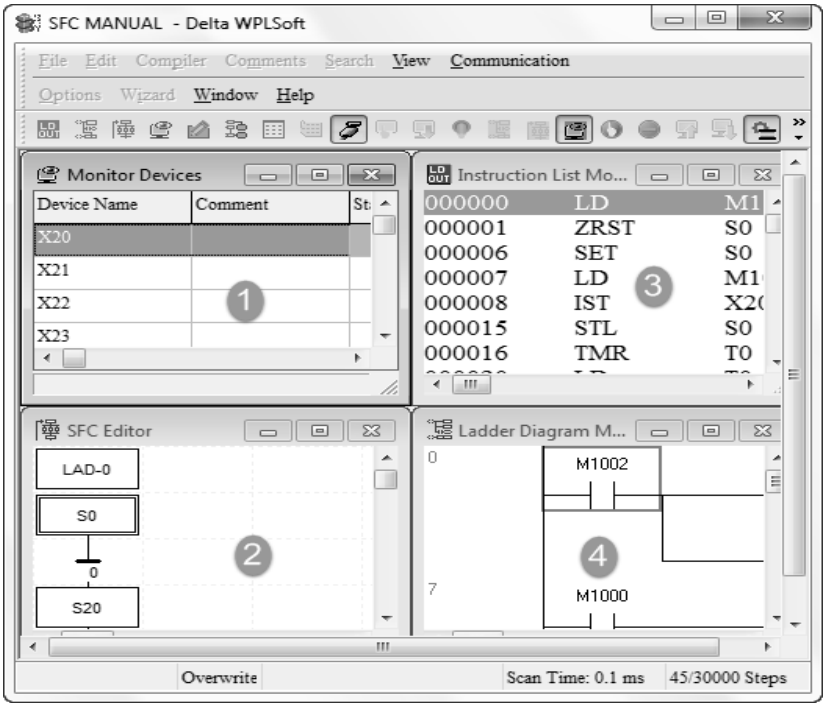
سپس مطابق شکل زیر و انتخاب قسمت مشخص شده با عدد ۱ مانیتورینگ فعال می‌شود



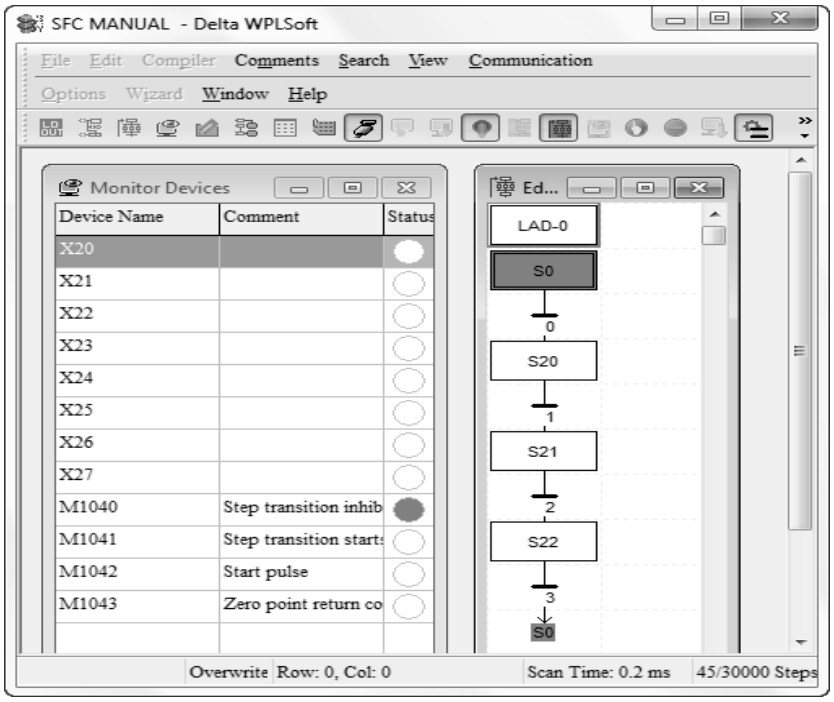
حال برای اینکه پنجره محیط SFC و محیط مانیتورینگ را بطور همزمان داشته باشیم مطابق شکل زیر عمل می‌کنیم. از منوی Window ، گزینه Title Vertically را انتخاب می‌کنیم.



با انجام مراحل بالا پنجره صفحه بعد باز می‌شود . در پنجره باز شده، در صفحه بعد قسمت مشخص شده با عدد ۱ محیط مانیتورینگ و قسمت ۲ محیط SFC و قسمت ۳ محیط زبان IL و قسمت ۴ محیط زبان STEP LADDER را نشان می‌دهد .



پنجره IL و پنجره STEP LADDER را ببندید و اندازه‌ی پنجره‌ها را به دلخواه تنظیم کنید.



همانطور که از شکل صفحه قبل دیده می‌شود با فعال کردن مانیتورینگ، اولین STEP یعنی S0 فعال می‌باشد همچنین فلگ M1040 نیز فعال می‌باشد اما حرکت بین STEP ها برقرار نمی‌باشد .

حال در پنجره مانیتورینگ، X26 را انتخاب و کلیک راست کرده و گزینه SET ON را فعال می‌کنیم.

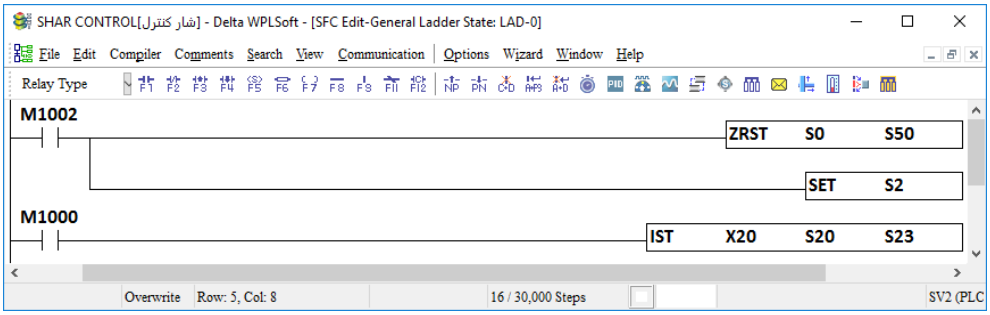
با این عمل ، فلگ M1040 غیر فعال شده و حرکت بین STEP ها آغاز می‌شود حتی در صورت SET OFF کردن X26 ، حرکت بین STEP ها همچنان برقرار خواهد بود .

در پنجره مانیتورینگ ، فلگ M1040 را انتخاب و کلیک راست کرده و گزینه SET ON را انتخاب می‌کنیم. فورا برنامه در همان STEP متوقف می‌شود و در صورتیکه این فلگ را SET OFF کنیم برنامه از همان STEP ای که متوقف شده بود ادامه پیدا می‌کند .

فرض کنید در حین اجرا ، برنامه در یکی از STEP ها باشد و ما به هردلیل می‌خواهیم برنامه به اولین STEP یعنی S0 برگردد برای این منظور کافیست X20 را انتخاب و کلیک راست کرده و SET ON کنیم با این عمل فورا فرایند اجرای برنامه به اولین STEP بر میگردد و البته در همان جا می ماند . برای شروع به کار مجدد ابتدا X20 را SET OFF می‌کنیم و سپس X26 را SET ON می‌کنیم با این عمل فرایند حرکت بین STEP ها آغاز می‌شود .

مد اتوماتیک با استفاده از دستور IST :

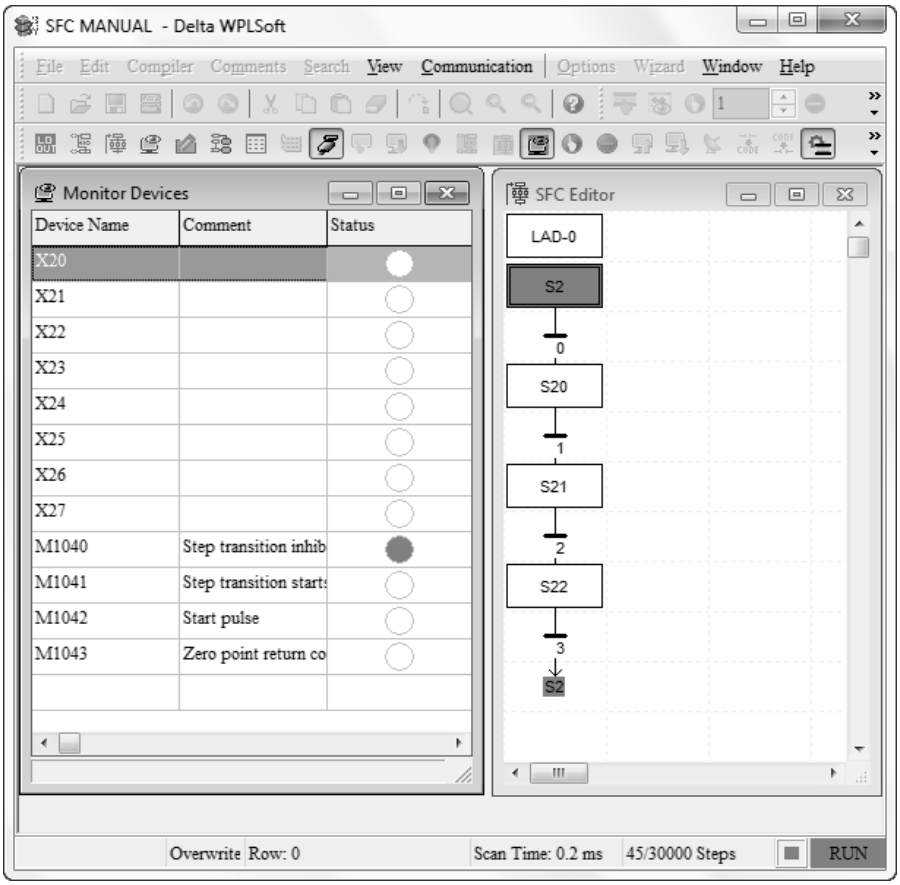
در محیط SFC مطابق روش گفته شده در مد دستی فلوچارت مورد نظر را ترسیم می‌کنیم همانطور که قبلا اشاره شد STEP شماره S2 مربوط به مد اتوماتیک می‌باشد . لذا باید از STEP شماره S2 بجای S0 استفاده کرد . همچنین در انتهای فلوچارت نیز بجای S0 از S2 استفاده می‌کنیم. برنامه ای که باید در قسمت LAD-0 نوشت بصورت زیر می‌باشد :



برنامه مربوط به STEP های دیگر مشابه مد دستی می‌باشد بعد از پایان برنامه نویسی STEP ها برنامه را Compile کرده و مطابق مراحل گفته شده در مد دستی وضعیت مانیتورینگ را فعال می‌کنیم .

مد ONE STEP در وضعیت Automatic :

اگر بخواهیم که اجرای برنامه بصورت تک STEP انجام شود یعنی به ازای هربار تحریک فقط یک STEP اجرا شود به صورت صفحه بعد عمل می‌کنیم.



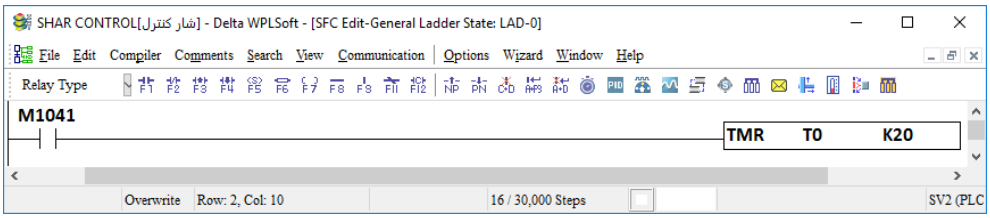
ابتدا در جدول مانیتورینگ X22 را انتخاب و روی آن راست کلیک کرده و SET ON می‌کنیم با این عمل وضعیت به اصطلاح ONE STEP انتخاب می‌شود حالا با انتخاب X26 و کلیک راست و SET ON کردن آن اجرای برنامه فقط به اندازه یک STEP اجرا می‌شود یعنی برنامه از S2 به S20 میرود مجددا X26 را SET OFF کرده و دوباره SET ON می‌کنیم برنامه به STEP بعدی میرود همین طور الی آخر. زمانیکه به انتهای STEP ها رسیدیم برای برگشت به STEP اول در حالیکه X26 غیر فعال است فلگ M1043 را SET ON کنیم فورا برنامه به اولین STEP برمیگردد .

نکته : برای اینکه برنامه به صورت خودکار از آخرین سطر برنامه به اولین سطر برنامه بیاید کافیست فلگ M1043 را در آخرین برنامه وارد کرده و با دستور Out M1043 آنرا فعال کنیم.

نکته : اگر در حین اجرای برنامه اشکالاتی بوجود امد و برنامه عملکرد مناسبی نداشت سیمولاتور را ببندید و مجددا باز کنید . برنامه را دانلود کرده برنامه را تست کنید .

مد ONE CYCLE در وضعیت Automatic :

برنامه نوشته شده در قسمت S2 بصورت زیر باشد در واقع فلگ M1041 این قابلیت را به مدار اضافه می‌کند .



برای این وضعیت X22 را SET OFF و ورودی X23 را SET ON می‌کنیم حال به محض اینکه ورودی X26 را SET ON کنیم بصورت خودکار یک بار تمام STEP ها در صورت برقراری شرط عبور مربرط به هر STEP اجرا میشود.

اگر X26 را بعد از اینکه SET ON کردیم ، آنرا SET OFF نکنیم ، برنامه بصورت خودکار بعد از اخرین STEP به اولین STEP برگشته و دوباره اجرا می‌شود .

مد CONTINUOUS OPERATION در وضعیت اتوماتیک :

در صورتیکه بخواهیم سیکل کار بصورت مداوم و پیوسته انجام شود ابتدا ورودی X24 فعال میگردد و سپس ورودی X26 فعال می‌شود در اینصورت سیکل آغاز می‌شود حتی با غیر فعال کردن X26 سیکل ادامه پیدا می‌کند .حال اگر در صورتیکه X26 غیر فعال است ورودی X27 را فعال کنیم ابتدا اجازه میدهد سیکل به اتمام برسد و سپس به اولین STEP برگشته و متوقف می‌شود

جهت توقف حرکت بین STEP ها می‌توان از فلگM1040 استفاده کرده با فعال کردن آن سیکل در همان SETP می ایستد و با غیر فعال M1040 مجددا از همان نقطه توقف به کار خود ادامه میدهد .

نکته : جهت برگشت سیکل کاری در هر زمان به اولین STEP از M1043 می‌توان استفاده کرد .

تشریح فلگ های مربوط به دستور IST :

M1040 : زمانیکه این فلگ ON باشد عبور از STEP ها غیر ممکن می‌شود وضعیت این فلگ در مد های مختلف بصورت زیر می‌باشد :

: MANUAL MODE

در این مد این فلگ در حالت ON باقی میماند .

: ZERO RETURN & ONE CYCLE OPERETION

فلگ در این وضعیتها در هنگام STOP و قبل از START در حالت ON می‌باشد .

: STEP OPERATION

بعد از تحریک X26 این فلگ فعال خواهد ماند.

: CONTINUOUS OPERATION

زمانیکه PLC از مد STOP به مد RUN برود ، فلگ M1040 برابر ON خواهد شد . زمانیکه X26 فعال شود آنگاه فلگ M1040 برابر OFF می‌شود .

: M1041

در مد MANUAL و ZERO RETURN این فلگ OFF می‌باشد.

در وضعیت STEP OPERATION و ONE-CYCLE OPERATION زمانیکه X26 برابر ON شود این فلگ نیز ON می‌شود

در وضعیت CONTINUOUS OPERATION زمانیکه X26 برابر ON شود این فلگ نیز ON می‌شود و زمانیکه X27 تحریک شود این فلگ خاموش می‌شود .

: M1042

زمانیکه استارت اتوماتیک تحریک شود PLC یک پالس برای شروع عملیات تولید می‌کند.

: M1043

این فلگ برای کامل کردن عملیات برگشت به نقطه صفر ZERO POINT RETURN می‌باشد.

: M1044

در مد CONTINUOUS OPERATION این فلگ می‌تواند به عنوان یک شرط برای فعال کردن شرایط عبور از S2 به STEP های بعدی باشد

: M1045

این فلگ برای ریست کردن تمامی خروجی ها مورد استفاده قرار میگیرد در شرایط زیر نیز عمل ریست کردن اتفاق می افتد

۱- تغییر وضعیت در مد دستی از S0 به S1

۲- تغییر وضعیت درمد اتوماتیک از S2 به S0

۳- تغییر وضعیت در مد اتوماتیک از S2 به S1

نکته : با توجه به عملکرد این فلگ ها به راحتی می‌توان از آنها در برنامه استفاده کرد .

۱۷۲.....اتوماسیون صنعتی DELTA

فصل هشتم

ساختمان انکودر و انواع

انکودر چست؟

به طورکلی انکودر به تجهیزی گفته می‌شود که یک حرکت مکانیکی (خطی و دورانی) را به سیگنال الکتریکی تبدیل کرده تا بتوان مشخصه های متفاوتی مانند سرعت چرخش و جهت چرخش را از آن سیگنال مورد استفاده قرار داد، یا به طور ساده تر انکودرها وسیله ای هستند که حرکت دورانی یا خطی را به سیگنال دیجیتالی 0 یا 1 تبدیل می‌کنند البته ناگفته نماند که بعضی از انکودر ها خروجی آنالوگ (0 تا 10 ولت و یا 4 تا 20 میلی آمپر) دارند.

برای مثال کاربرد انکودرها در سرو موتور ها مثال ساده و فراوان این کاربرد می‌باشد که خروجی انکودر به درایو مخصوص سرو منتقل می‌شود.در اتوماسیون صنعتی از انکودر استفاده های بسیاری می‌شود مانند صنایع بسته بندی و دستگاههای صنعتی که نیاز به اندازه گیری سرعت و موقعیت است از انکودر های مختلف استفاده می‌کنند.

کاربرد انکودرها

انکودر ها معمولا در سیستم های اتوماسیون و کنترلی در قسمتی نصب می‌شوند که بتوانند از خروجی فیدبک گرفته و برای کنترلر ارسال کنند و پارامترهای زیر را مشخص کنند:

- اندازه گیری سرعت
- اندازه گیری و کنترل موقعیت
- جهت چرخش شفت

شما تصور کنید که قصد کنترل سرعت یک موتور الکتریکی را دارید، اگر شما فرمان های موتور را از یک اینورتر به آن اعمال کنید و سرعت را کم و یا زیاد نمایید توانسته اید سرعت موتور را کنترل کنید ولی مشکل اصلی زمانی پیش می آید که شما قصد داشته باشید تا سرعت موتور را روی سرعت خاص به صورت دقیق و ثابت نگه دارید در این حالت شما نیاز به گرفتن فیدبک از موتور هستید اما این سوال پیش می آید که آیا سرعت دقیق موتور همان عددی که شما به آن اعمال کرده اید هست یا نه؟پس یکی دیگر از کاربرد های انکودر اندازه گیری سرعت موتور ها می‌باشد.

مواردی پیش می آید که شما می خواهید نوار نقاله مورد نظرتان به اندازه ای مشخص حرکت کند و در آن نقطه توقف کند که این موضوع یکی از پرکاربردترین نوع استفاده از انکودر در صنعت، مخصوصا در ماشین آلات بسته بندی و پرکن های صنعتی به کار گرفته می‌شود. برای این کار شما خروجی انکودر را اندازه گیری می‌کنید و با در نظر گرفتن ضرایب گیربکس و کوپلینگ ها مسافت طی شده را به دست می آورید، این کار به اصطلاح کنترل موقعیت نامیده می‌شود.

مواردی از کاربرد انکودرها

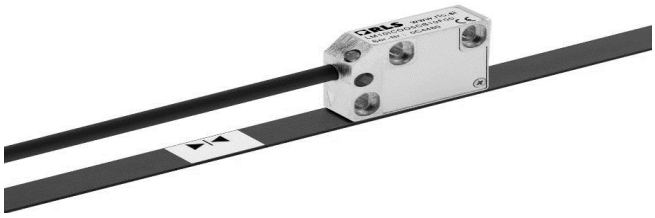
اندازه گیری سرعت اندازه گیری موقعیت

اندازه گیری زاویه	اندازه گیری جهت حرکت شفت
ماشین های cnc	ماشین های بسته بندی
آسانسور	روباتهای صنعتی

انواع انکودر از نظر ساختمان :

۱ – انکودرهای خطی (LINEAR) :

این نوع انکودر به منظور اندازه گیری حرکت خطی مورد استفاده قرار می گیرد و از لحاظ ساختمانی در انواع نوری، مغناطیسی و مقاومتی تولید می‌شوند .انکودر خطی مدل نوری، دارای خروجی دیجیتال و انکودر خطی مدل مغناطیسی و مقاومتی دارای خروجی آنالوگ است. این انکودرها در دستگاههایی مثل دستگاه پرس برای اندازه گیری میزان حرکت طولی پیستون جک ودر غلتکها برای اندازه گیری میزان فاصله طولی بین توپی های غلتک استفاده می‌شوند.



۲ – انکودرهای دورانی (ROTARY)

انکودر دوار یا شافت انکودر(Shaft encoder) ، در واقع یک تجهیز الکترومکانیکی است که موقعیت یا حرکت زاویه ای یک شافت یا یک محور را به یک کد آنالوگ یا دیجیتال تبدیل می‌کند. این نوع انکودر به طور معمول از یک یا دو جفت LED فرستنده و گیرنده و یک دیسک مدرج (سیاه و سفید یا جای خالی و جای پر) استفاده می کند. دیسک مدرج مابین سنسورهای فرستنده و گیرنده قرار دارد و این مجموعه بر روی محور چرخان دستگاه قراردارد. هنگام چرخش شافت دیسک درجه بندی شده به همراه شافت می چرخد و سنسورهای گیرنده وفرستنده ثابت می مانند. درنتیجه سیگنالی که از سمت فرستنده به گیرنده ارسال می‌شود توسط چرخش دیسک به طور متوالی قطع و وصل می‌شود به این صورت ما در خروجی گیرنده یک قطار از پالسهای متوالی صفر و یک منطقی داریم. این پالسها همان کدهایی هستند که انکودر تولید می‌کند و برای اینکه تبدیل به موقعیت و یا تعداد دوران شافت شود به ورودی یک دیکودر داده می‌شود. هر چه تعداد نقاط سیاه و سفید روی دیسک بیشتر باشد تعداد این صفر و یک های منطقی در یک دوران کامل (۳۶۰ درجه) بیشترمی‌شودو درنتیجه دقت اندازه گیری آن افزایش می یابد

.

انواع انکودرها از نظر ساختار

۱ – نوری (Optical)

انکودر های نوری، دقیق ترین نوع انکودر ها هستند. وقتی یک انکودر نوری را انتخاب می‌کنیم، خیلی مهم است که انکودر مذکور حفاظت های اضافی برای جلوگیری از آلودگی ناشی از گرد و غبار، لرزش (vibration)و سایر شرایط رایج در محیط های صنعتی را داشته باشد.

۲ – خازنی (Capasitive)

انکودرهای خازنی نسبت به دو مدل قبلی در دنیای صنعت تازه وارد محسوب می‌شوند. انکودرهای خازنی از نظر نیرومندی مانند مدل های مغناطیسی هستند ولی قدرت تفکیک پذیری انکودرهای نوری از انکودرهای خازنی بالاتر است. البته انکودرهای خازنی بدون توجه به نوع تکنولوژی مورد استفاده قابلیت تشخیص حرکت و تبدیل آن به سیگنال های استاندارد رایج در صنعت را دارا هستند.

۳ – مغناطیسی (Magnetic)

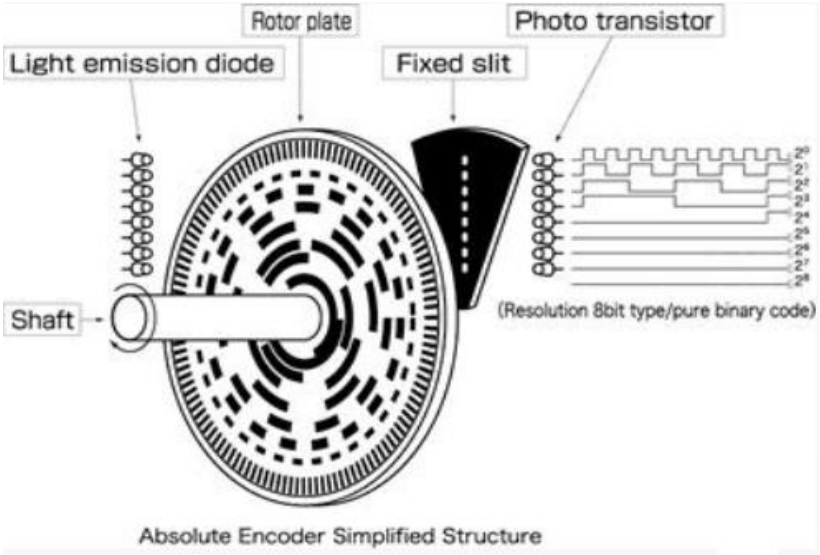
انکودرهای مغناطیسی با وجود اینکه از انکودرهای نوری مقاوم ترند و اغلب در محیط های در معرض آلودگی، بخار، لرزش و سایر انواع تداخل های محیطی استفاده می‌شوند ولی قدرت تفکیک (resolution)یا دقت انکودرهای نوری بالاتر می‌باشد.

انکودرها از نظر نوع عملکرد

۱ – مطلق (ABSOLUTE ENCODER)

یکی از ایرادات انکودرهای افزایشی این است که شمارش پالسهای آن در یک میانگیر یا اکسترنال کانتر (Counter External)ذخیره می‌شود. اگر برق قطع شود و یا به هر حال اختلال پیدا کند مقدار شمارش شده از بین خواهد رفت زیرا نقطه صفر برای دستگاه به هنگام راه اندازی مجدد نا معلوم یا تعریف نشده است این بدان معناست که چنانچه بار را از یک ماشین الکتریکی با انکودری که دارای چرخاننده الکتریکی است بگیریم دیگر انکودر موقعیت صحیح را نخواهد دانست. برای حل این مشکل از انکودرهای مطلق استفاده می‌شود و در این صورت ماشین همیشه موقعیت خود را می داند. در انکودر مطلق دیسکی که دارای چندین گروه از قطعات به فرم دایره های متحدالمرکز نقطه شروع دایره های

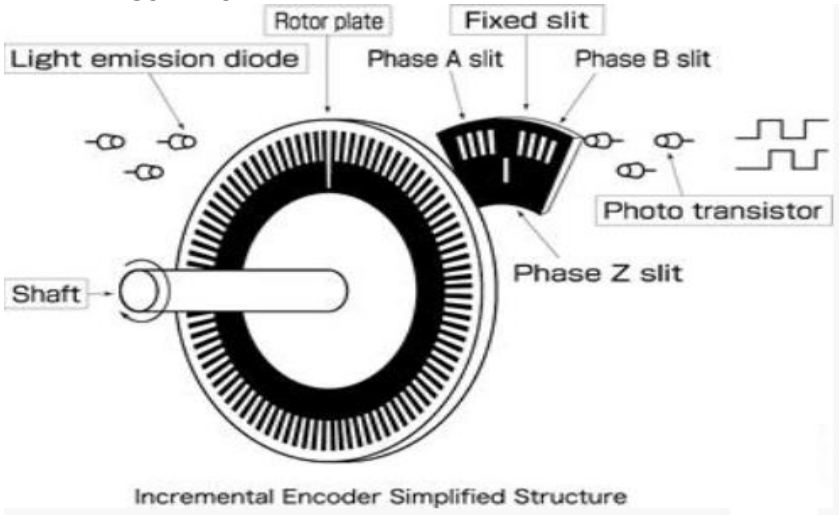
متحدالمرکز در مرکز دیسک انکودر چرخشی است استفاده می‌شود. چنانچه حلقه به سمت محیط دایره برود هر یک از آنها دارای دو برابر قطعه نسبت به حلقه قبلی و داخلی تر خود خواهند داشت اولین حلقه که داخلی ترین حلقه هاست یک قسمت شفاف و یک قسمت تیره خواهد داشت و حلقه سوم نیز از هر یک از قطعات ۴ قسمت خواهد داشت اگر انکودر دارای ۱۰ حلقه باشد آخرین حلقه یا دورترین آن از مرکز دارای ۵۱۲ قطعه می‌باشد چنانچه هر حلقه از انکودر مطلق دارای دو برابر قطعه از حلقه قبلی خود باشد بین شماره ها یک سیستم باینری برقرار می‌شود. در این نوع انکودر به ازای هر حلقه روی دیسک یک منبع نور و دریافت کننده آن وجود دارد.



انکودرهای چرخشی مطلق

۲ – افزایشی (INCREMENTAL)

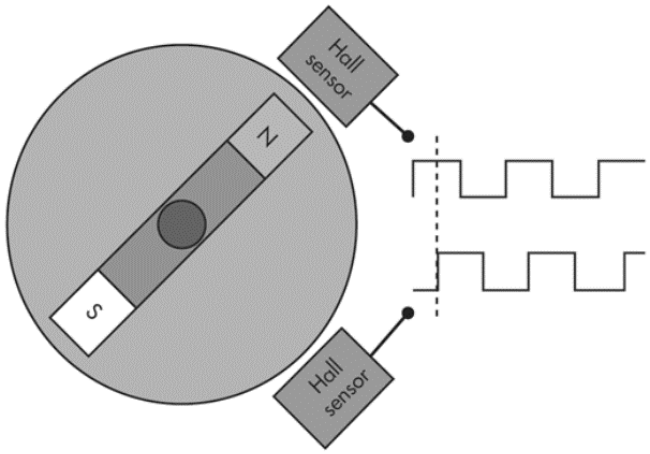
اگر انکودر فقط دارای یک ردیف پالس باشد نمی‌تواند به درستی جهت چرخش را نشان دهد و باید دارای پالس دومی نیز باشد که به این منظور باید یک فرستنده و گیرنده نوری دیگر به مجموعه اضافه شود. بنابر این در لحظاتی که نمی‌توان جهت چرخش شفت را تشخیص داد(در لحظاتی هر دو پالس یک منطقی و در لحظاتی صفر منطقی هستند) نیاز به پالس سومی داریم که این پالس بر اثر یک دور چرخش کامل به وجود می آید که به آن پالس فرمان می‌گویند و از آن برای شمارش تعداد دوران نیز استفاده می‌شود.



انکودرهای چرخشی افزایشی

انکودرهای چرخشی مغناطیسی

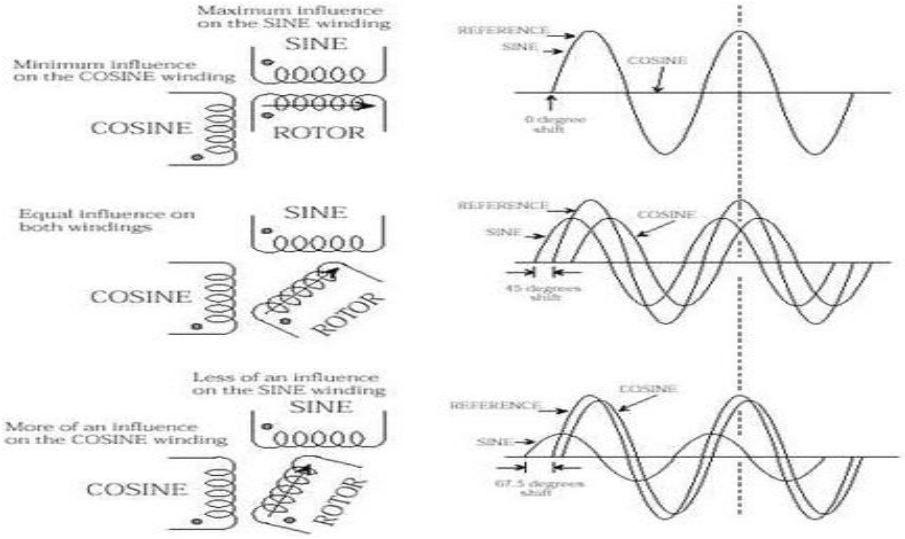
این انکودرها هم همانند انکدرهای افزایشی بوده ولی در تولید پالس از سنسورهای اثر هال استفاده شده است.



انکدرهای چرخشی مغناطیسی

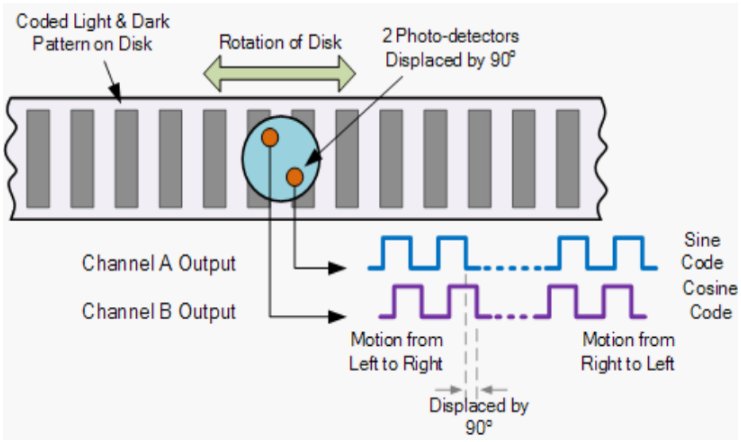
انکودر های چرخشی سینوسی و کسیونوسی resolvers

همانطور که در شکل زیر مشاهده می‌کنید،این انکودر ها تفاوت عمده ای با انکدر های افزایشی و مطلق دارد، با این تفاوت که در این انکودر ها پالس ها به صورت دیجیتال نبوده و به صورت سیگنال آنالوگ می‌باشد که با اندازه گیری فرکانس و اختلاف فاز و شمارش نقاط صفر سیگنالها می‌توانیم سرعت، موقعیت و جهت حرکت شفت را به دست آوریم.



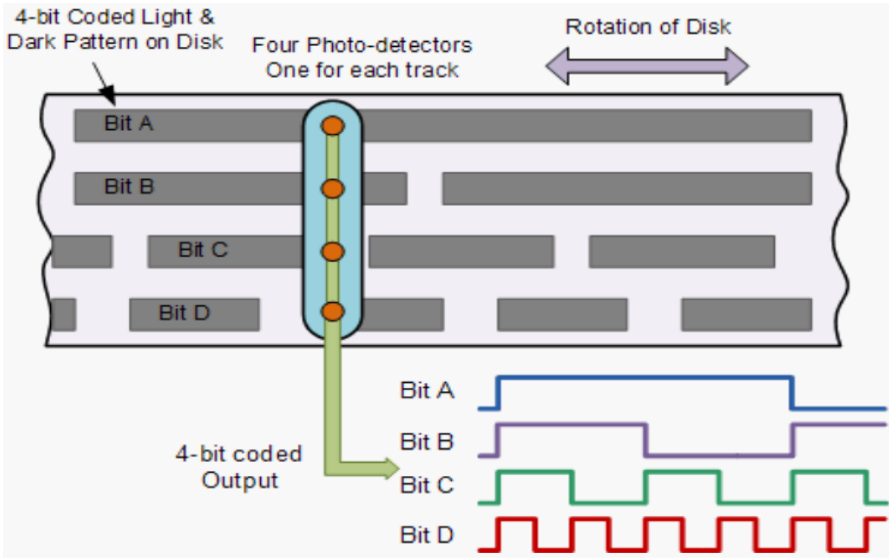
انکودر های افزایشی خطی

این انکودرها از نظر پالس های خروجی همانند انکودر های چرخشی بوده ولی این انکودرها در ظاهر بسیار ساده بوده و حرکت آنها به صورت خطی می‌باشد.



انکودر های مطلق خطی

یکی از مدل های دیگر انکودرها به صورت مطلق خطی بوده که کاربرد بسیاری در اندازه گیری های طولی دارد که به خط کش های دیجیتال معروف هستند.



سیم بندی انکودر

در تمام انکودرها دوسیم برای تغذیه ی انکودر موجود است. غیر از این دوسیم تعداد سیمهای خارج شده از انکدر با توجه به نوع انکودر متفاوت می‌باشد که در ادامه به بررسی آنها می‌پردازیم :

۱ – **سیم فاز A رنگ مشکی** : در صورتیکه اولین پالس از فاز A ارسال شود و دومین پالس از فاز B ارسال شود جهت چرخش راستگر است.

۲ – **سیم فاز B رنگ سفید** : در صورتیکه اولین پالس از فاز B ارسال شود و دومین پالس از فاز A ارسال شود جهت چرخش چپگر است.

۳ – **سیم فاز Z رنگ نارنجی** : به ازای هر یک دور کامل شفت انکودر این فاز یک پالس ارسال می‌کند.

۴ – **سیم فاز A'** : این سیم یک پالس 24- از نوع پالس فاز A تولید می‌کند. در برخی از محلهای نصب انکدر بدلیل وجود نویز امکان ارسال پالس اشتباه وجود دارد لذا از فاز A و A' با یکدیگر استفاده می‌شود تا هرگاه یک پالس مثلا A به سیستم ارسال شد و پالس منفی آن یعنی A' ارسال نشد، سیستم این پالس را به عنوان نویز شناسایی کند.

۵ – **سیم فاز B'** : این سیم یک پالس 24- از نوع پالس فاز B تولید می‌کند. در برخی از محلهای نصب انکدر بدلیل وجود نویز امکان ارسال پالس اشتباه وجود دارد لذا از فاز B و B' با یکدیگر استفاده می‌شود

تا هرگاه یک پالس مثلا B به سیستم ارسال شد و پالس منفی آن یعنی B' ارسال نشد، سیستم این پالس را به عنوان نویز شناسایی کند.

۶ – **سیم فاز Z'** : این سیم یک پالس 24- از نوع پالس فاز Z تولید می‌کند. در برخی از محلهای نصب انکودر بدلیل وجود نویز امکان ارسال پالس اشتباه وجود دارد لذا از فاز Z و Z' با یکدیگر استفاده می‌شود تا هرگاه یک پالس مثلا Z به سیستم ارسال شد و پالس منفی آن یعنی Z' ارسال نشد، سیستم این پالس را به عنوان نویز شناسایی کند.

۱۸۲.....اتوماسیون صنعتی DELTA

پیوست

جدول حافظه های خاص (فلگها) M

جدول رجیسترهای خاص D

Special M	Function	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Attrib.	Latch-ed	Default
M1000*	Monitor normally open contact	OFF	ON	OFF	R	NO	OFF
M1001*	Monitor normally closed contact	ON	OFF	ON	R	NO	ON
M1002*	Enable single positive pulse at the moment when RUN is activate (Normally OFF)	OFF	ON	OFF	R	NO	OFF
M1003*	Enable single negative pulse at the moment when RUN is activate (Normally ON)	ON	OFF	ON	R	NO	ON
M1004*	ON when syntax errors occur	OFF	OFF	-	R	NO	OFF
M1008*	Watchdog timer (ON: PLC WDT time out)	OFF	OFF	-	R	NO	OFF
M1009	Indicate LV signal due to 24VDC insufficiency	OFF	-	-	R	NO	OFF
M1011*	10ms clock pulse, 5ms ON/5ms OFF	OFF	-	-	R	NO	OFF
M1012*	100ms clock pulse, 50ms ON / 50ms OFF	OFF	-	-	R	NO	OFF
M1013*	1s clock pulse, 0.5s ON / 0.5s OFF	OFF	-	-	R	NO	OFF
M1014*	1 min clock pulse, 30s ON / 30s OFF	OFF	-	-	R	NO	OFF
M1015*	Enable high-speed timer	OFF	-	-	R/W	NO	OFF
M1016*	Indicate Year display mode of RTC.	OFF	-	-	R/W	NO	OFF
M1017*	±30 seconds correction on real time clock	OFF	-	-	R/W	NO	OFF
M1018	Flag for Radian/Degree, ON for degree	OFF	-	-	R/W	NO	OFF
M1025*	Indicate incorrect request for communication	OFF	-	-	R	NO	OFF
M1026	RAMP mode selection	OFF	-	-	R/W	NO	OFF
M1027	PR output mode selection (8/16 bytes)	OFF	-	-	R/W	NO	OFF
M1028	Switch T64~T126 timer resulotion (10ms/100ms). ON =10ms	OFF	-	-	R/W	NO	OFF
M1031*	Clear all non-latched memory	OFF	-	-	R/W	NO	OFF
M1032*	Clear all latched memory	OFF	-	-	R/W	NO	OFF
M1033*	Output state latched at STOP	OFF	-	-	R/W	NO	OFF
M1034*	Disable all Y outputs	OFF	-	-	R/W	NO	OFF
M1035*	Enable X7 input point as RUN/STOP switch	-	-	-	R/W	YES	OFF
M1037*	Enable 8-sets SPD function (Has to be used with D1037)	OFF	OFF	OFF	R/W	NO	OFF
M1038	Switch T200~T255 timer resulotion (10ms/1ms). ON = 1ms	OFF	-	-	R/W	NO	OFF
M1039*	Fix scan time	OFF	-	-	R/W	NO	OFF

M1040	Disable step transition	OFF	-	-	R/W	NO	OFF
M1041	Step transition start	OFF	-	OFF	R/W	NO	OFF
M1042	Enable pulse operation	OFF	-	-	R/W	NO	OFF
M1043	Zero return completed	OFF	-	OFF	R/W	NO	OFF
M1044	Zero point condition	OFF	-	OFF	R/W	NO	OFF
M1045	Disable "all output reset" function	OFF	-	-	R/W	NO	OFF
M1046	Indicate STL status	OFF	-	-	R	NO	OFF
M1047	Enable STL monitoring	OFF	-	-	R/W	NO	OFF
M1048	Indicate alarm status	OFF	-	-	R	NO	OFF
M1049	Enable alarm monitoring	OFF	-	-	R/W	NO	OFF
M1050	Disable interruption I000 / I001	OFF	-	-	R/W	NO	OFF
M1051	Disable interruption I100 / I101	OFF	-	-	R/W	NO	OFF
M1058	COM3 monitor request	OFF	-	-	R/W	NO	OFF
M1059	Disable high-speed counter interruptions I010~I080	OFF	-	-	R/W	NO	OFF
M1060	System error message 1	OFF	-	-	R	NO	OFF
M1061	System error message 2	OFF	-	-	R	NO	OFF
M1200	C200 counting mode (ON: count down)	OFF	-	-	R/W	NO	OFF
M1201	C201 counting mode (ON: count down)	OFF	-	-	R/W	NO	OFF
M1202	C202 counting mode ON: count down)	OFF	-	-	R/W	NO	OFF
M1203	C203 counting mode (ON: count down)	OFF	-	-	R/W	NO	OFF
M1204	C204 counting mode (ON: count down)	OFF	-	-	R/W	NO	OFF
M1205	C205 counting mode (ON :count down)	OFF	-	-	R/W	NO	OFF
M1270	C235 counting mode (ON: falling-edge count)	OFF	-	-	R/W	NO	OFF
M1271	C236 counting mode ON: falling-edge count)	OFF	-	-	R/W	NO	OFF
M1272	C237 counting mode (ON: falling-edge count)	OFF	-	-	R/W	NO	OFF
M1273	C238 counting mode (ON: falling-edge count)	OFF	-	-	R/W	NO	OFF
M1274	C239 counting mode (ON: falling-edge count)	OFF	-	-	R/W	NO	OFF
M1275	C240 counting mode (ON: falling-edge count)	OFF	-	-	R/W	NO	OFF
M1276	C241 counting mode (ON: falling-edge count)	OFF	-	-	R/W	NO	OFF
M1277	C242 counting mode (ON: falling-edge count)	OFF	-	-	R/W	NO	OFF

Special D	Content	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Attrib.	Latch-ed	Default
D1000*	Setting value of the watchdog timer (WDT) (Unit: 1ms)	200	-	-	R/W	NO	200
D1001	Displaying the firmware version of TP (For example, the firmware version is 1.0 if the value in D1001 is HXX10.)	-	-	-	R	NO	#
D1002*	Program capacity	7920	-	-	R	NO	7920
D1003	Sum of the PLC internal program memory	-	-	-	R	YES	7920
D1004*	Syntax check error code	0	0	-	R	NO	0
D1008*	Step address when WDT is ON	0	-	-	R	NO	0
D1009	Number of LV (Low voltage) signal occurrence	-	-	-	R	YES	0
D1010*	Current scan time (Unit: 0.1ms)	#	#	#	R	NO	0
D1011*	Minimum scan time (Unit: 0.1ms)	#	#	#	R	NO	0
D1012*	Maximum scan time (Unit: 0.1ms)	#	#	#	R	NO	0
D1015*	Value of accumulative high-speed timer (0~32,767, unit: 0.1ms)	0	-	-	R/W	NO	0
D1018*	π PI (Low word)	H'0FDB	H'0FDB	H'0FDB	R/W	NO	H'0FDB
D1019*	π PI(High word)	H'4049	H'4049	H'4049	R/W	NO	H'4049
D1022	Counting mode selection (Double frequency/ 4 times frequency) for AB phase counter (From X0, X1 input)	4	-	-	R/W	NO	4
D1025*	Code for communication request error	0	-	-	R	NO	0
D1028	Index register E0	0	-	-	R/W	NO	0
D1029	Index register F0	0	-	-	R/W	NO	0
D1036*	COM1 (RS-232) communication protocol	H'86	-	-	R/W	NO	H'86
D1038*	1. Delay time setting for data response when PLC is SLAVE in COM2 / COM3 RS-485 communication. Range: 0 ~ 10,000 (unit: 0.1ms). 2. By using PLC LINK in COM2 (RS-485), D1038 can be set to send next communication data with delay. Range: 0 ~ 10,000 (Unit: one scan cycle)	-	-	-	R/W	NO	0
D1039*	Fixed scan time (ms)	0	-	-	R/W	NO	0
D1040	No. of the 1st step point which is ON.	0	-	-	R	NO	0
D1041	No. of the 2nd step point which is ON	0	-	-	R	NO	0
D1042	No. of the 3rd step point which is ON.	0	-	-	R	NO	0
D1043	No. of the 4th step point which is ON	0	-	-	R	NO	0
D1044	No. of the 5th step point which is ON.	0	-	-	R	NO	0
D1045	No. of the 6th step point which is ON	0	-	-	R	NO	0
D1046	No. of the 7th step point which is ON.	0	-	-	R	NO	0
D1047	No. of the 8th step point which is ON	0	-	-	R	NO	0
D1049	No. of alarm which is ON	0	-	-	R	NO	0
D1050 ↓ D1055	Converted data for Modbus communication data processing. PLC automatically converts the ASCII data in D1070~D1085 into Hex data and stores the 16-bit Hex data into D1050~D1055	0	-	-	R	NO	0

<table>
<tr><td>D1110*</td><td>Average value of analog input channel 0 (AD 0) When average times in D1062 is set to 1, D1110 indicates present value.</td><td>0</td><td>-</td><td>-</td><td>R</td><td>NO</td><td>0</td></tr>
<tr><td>D1111*</td><td>Average value of analog input channel 1 (AD 1) When average times in D1062 is set to 1, D1111 indicates present value</td><td>0</td><td>-</td><td>-</td><td>R</td><td>NO</td><td>0</td></tr>
<tr><td>D1112*</td><td>Average value of analog input channel 2 (AD 2) Whenaverage times in D1062 is set to 1, D1112 indicates present value</td><td>0</td><td>-</td><td>-</td><td>R</td><td>NO</td><td>0</td></tr>
<tr><td>D1113*</td><td>Average value of analog input channel 3 (AD 3) Whenaverage times in D1062 is set to 1, D1113 indicates present value</td><td>0</td><td>-</td><td>-</td><td>R</td><td>NO</td><td>0</td></tr>
<tr><td>D1114*</td><td>Setting the mode of analog input/output
<table><tr><td>Bit</td><td>11-10</td><td>9-8</td><td>7-6</td><td>5-4</td><td>3-2</td><td>1-0</td></tr><tr><td>Channel</td><td>CH5</td><td>CH4</td><td>CH3</td><td>CH2</td><td>CH1</td><td>CH0</td></tr></table>Setting the mode of input:
00: Voltage mode
01: Current mode (0~20mA)
11: Current mode (4~20mA)
Setting the mode of output:
00: Voltage mode
01: Current mode</td><td>-</td><td>-</td><td>-</td><td>R/W</td><td>YES</td><td>0</td></tr>
<tr><td>D1115*</td><td>Analog input/output mode setting</td><td>-</td><td>-</td><td>-</td><td>R/W</td><td>YES</td><td>0</td></tr>
<tr><td>D1116*</td><td>Output value of analog output channel 0 (DA 0)</td><td>0</td><td>0</td><td>0</td><td>R/W</td><td>NO</td><td>0</td></tr>
<tr><td>D1117*</td><td>Output value of analog output channel 1 (DA 1)</td><td>0</td><td>0</td><td>0</td><td>R/W</td><td>NO</td><td>0</td></tr>
</table>

۱۸۸..... اتوماسیون صنعتی DELTA

منابع :

Delta Industrial Automation Product User manual

DVP 04PT manual

DVP 04TC Manual

DVP 06XA manual

DVP 14SS2 Manual

www.deltaww.com