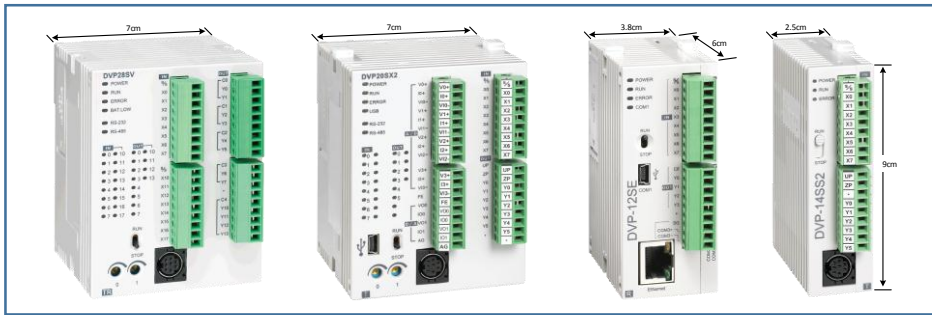


معرفی سخت افزار PLC :

در انتخاب مدل PLC عواملی باعث گرانتر شدن قیمت آن می شوند.



- 1- شمارنده های سرعت بالای سخت افزاری
- 2- مقدار فرکانس پالس تولیدی
- 3- ساعت و تاریخ داخلی
- 4- وجود یا عدم وجود ورودی/خروجی آنالوگ
- 5- ظرفیت حافظه برنامه
- 6- تعداد و نوع پورت های ارتباطی
- 7- قابلیت اتصال به ماژول های اضافه شونده از سمت چپ



DVP-28SV11(T/R)2	DVP20SX211(T/R)	DVP12SE11(T/R)	DVP12SA211(T/R)	DVP14SS211(T/R)	مشخصات	
24VDC (-15% ~ 20%)					تغذیه	
16	8	8	8	8	تعداد ورودی های دیجیتال (24VDC)	
12	6	4	4	6	تعداد خروجی های دیجیتال	
-	4 (12bit)	-	-	-	تعداد ورودی های آنالوگ	
-	2 (12bit)	-	-	-	تعداد خروجی های آنالوگ	
512	480				حداکثر ورودی + خروجی دیجیتال قابل افزایش	
2*100KHz 2*10KHz		1*50KHz 3*5KHz		2*10KHz 2*5KHz	تعداد آنکدرهای قابل اتصال (A/B mode)	
4*200KHz		2*100KHz		2*10KHz	تعداد سروموتورهای قابل اتصال (برای خروجی های ترانزیستوری) (Pulse/Direction mode)	
دارد				ندارد	RTC	
دارد	ندارد (بعد از 15 روز خاموش بودن، تاریخ و ساعت صفر می شوند)		دارد	ندارد	باتری داخلی	
30K STEP	16K STEP			8K STEP	ظرفیت حافظه برنامه	
D0 ~ D199, (200)	D0~D407, (408) D600~D999, (400) D3920~D4999, (5880)		D0~D407, (408) D600~D999, (400) D3920~D4999, (1080)		حافظه 16 بیتی عمومی برای استفاده در برنامه	
-	D5000~D9799, (4800)		D5000~D9799, (4800)	-		
-	D10000~D11999, (2000)		-	-		
D200 ~ D999, (800) D2000 ~ D9999, (8000)	D408~D599, (192) D2000~D3919, (1920)		D408~D599, (192) D2000~D3919, (1920)		حافظه 16 بیتی حافظه دار برای استفاده در برنامه	
D1000~D1999, (1000)						حافظه 16 بیتی خاص
M0 ~ M499, (500)	M0~M511, (512) M768~M999, (232) M2000~M2047, (48)		M0~M511, (512) M768~M999, (232) M2000~M2047, (48)		حافظه 16 بیتی عمومی برای استفاده در برنامه	
M500 ~ M999, (500) M2000 ~ M4095, (2096)	M512~M767, (256) M2048~M4095, (2048)		M512~M767, (256) M2048~M4095, (2048)		حافظه 16 بیتی حافظه دار برای استفاده در برنامه	
M1000~M1999, (1000)						حافظه 16 بیتی خاص
دارد						پورت راست برای اتصال به ماژولها
دارد			دارد	ندارد	پورت چپ برای اتصال به ماژولها	
1 * RS-232 1 * RS-485	1 * RS-232 1 * RS-485 1 * Mini USB	2 * RS-485 1 * Ethernet 1 * Mini USB	1 * RS-232 2 * RS-485	1 * RS-232 1 * RS-485	پورت های ارتباطی	
LD: 0.24µs MOV: 3.4µs	LD: 0.35µs MOV: 3.4µs	LD: 0.64µs MOV: 3.4µs	LD: 0.35µs MOV: 3.4µs	LD: 0.35µs MOV: 3.4µs	سرعت اسکن برنامه	
5~95% , 0~55°C						دمای کاری و رطوبت مجاز

PLC های سری ES2 :



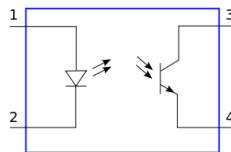
DVP60ES200(T/R)	DVP40ES200(T/R)	DVP32ES200TC	DVP32ES211T	DVP32ES200(T/R)	DVP24ES200(T/R)	DVP16ES200(T/R)	مشخصات	
100 ~ 240VAC			24VDC	100 ~ 240VAC			تغذیه	
36	24	16	16	16	16	8	تعداد ورودی های دیجیتال (24VDC)	
24	16	16	16	16	8	8	تعداد خروجی های دیجیتال	
1 * RS-232 2 * RS-485		1 * RS-2322 1 * RS-485 1 * CANOpen	1 * RS-232 2 * RS-485					پورتهای ارتباطی
256							حداکثر ورودی + خروجی دیجیتال قابل افزایش	
2*100KHz							تعداد آنکدهای قابل اتصال (A/B mode)	
2*100KHz							تعداد سروموتورهای قابل اتصال ( برای خروجی های ترانزیستوری ) (Pulse/Direction mode)	
دارد							RTC	
16K STEP							حافظه	
دارد							پورت راست برای اتصال به مازولها	
ندارد							پورت چپ برای اتصال به مازولها	
LD: 0.35µs MOV: 3.4µs							سرعت اسکین برنامه	

ورودی های دیجیتال :

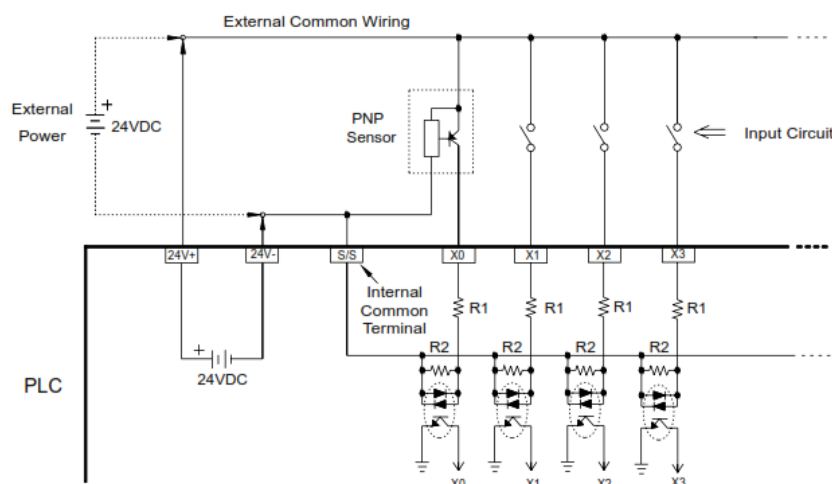
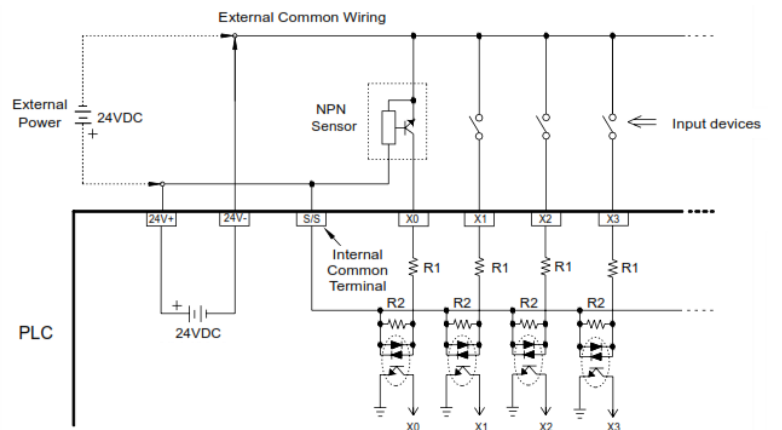


نحوه اتصال ورودی های دیجیتال :

ورودی ها با ولتاژ 24VDC فعال می شوند، برای ایزوله و ایمن کردن CPU از ورودی ها، از اپتوکوپلر استفاده می شود. با استفاده از اپتوکوپلر نویزها و ولتاژهای احتمالی به CPU راه پیدا نمی کنند.



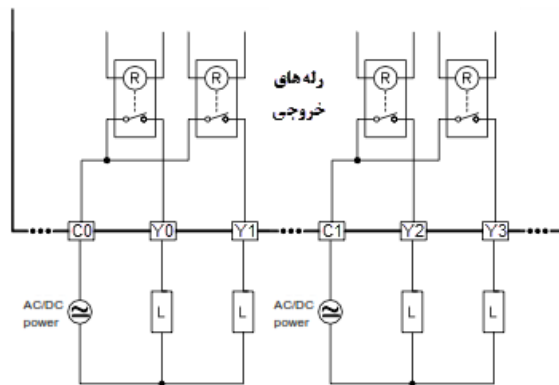
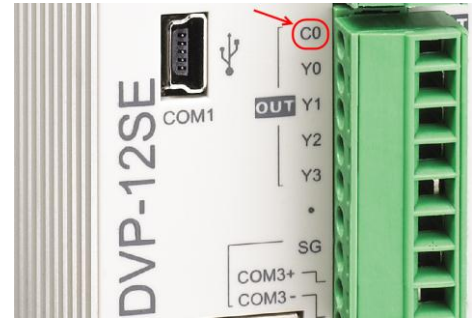
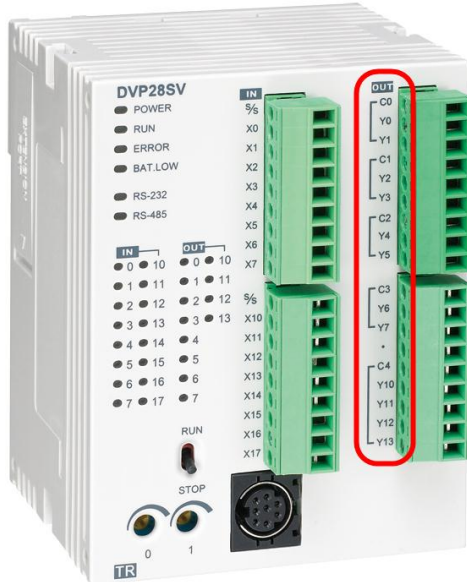
پایه S/S مشترک اپتوکوپلرهاست، چنانچه پایه به 24VDC- وصل شود، ورودی ها با 24VDC+ فعال می شوند و بالعکس.



سیم کشی و نحوه اتصال خروجی های PLC :

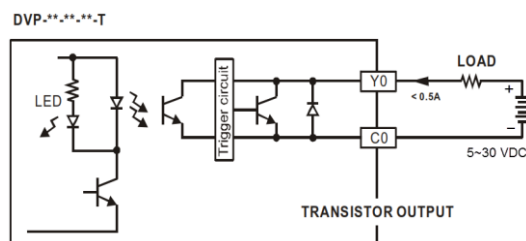
خروجی های دیجیتال دو نوع هستند : رله ایی ، ترانزیستوری

نحوه اتصال خروجی های دیجیتال رله ایی :

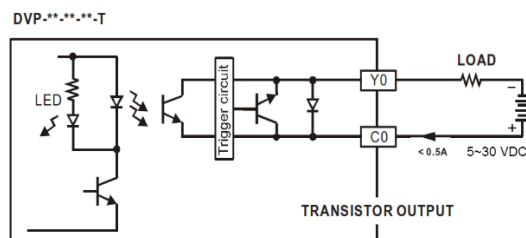


خروجی های ترانزیستوری دو نوع هستند : ترانزیستورهای SINK و ترانزیستورهای SOURCE

در مدل های ترانزیستوری نوع SINK پایه مشترک C... باید به پلاریته - وصل شود.



در مدل های ترانزیستوری نوع SOURCE پایه مشترک C... باید به پلاریته + وصل شود.



## LED های نمایشگر وضعیت

**LED Power**: این LED سبز رنگ هنگام اتصال تغذیه PLC، به صورت ممتد روشن می ماند.

**RUN**: هرگاه PLC در حالت RUN باشد، این چراغ سبز رنگ به صورت ممتد روشن می ماند.

**ERR**: در حالت های عملکرد PLC چه در حالت RUN و چه STOP این نمایشگر خاموش است، اگر این LED روشن شود، یعنی در سیستم خطایی رخ داده است (مثلا خطای برنامه، خطای تایمر WDT و غیره ...).

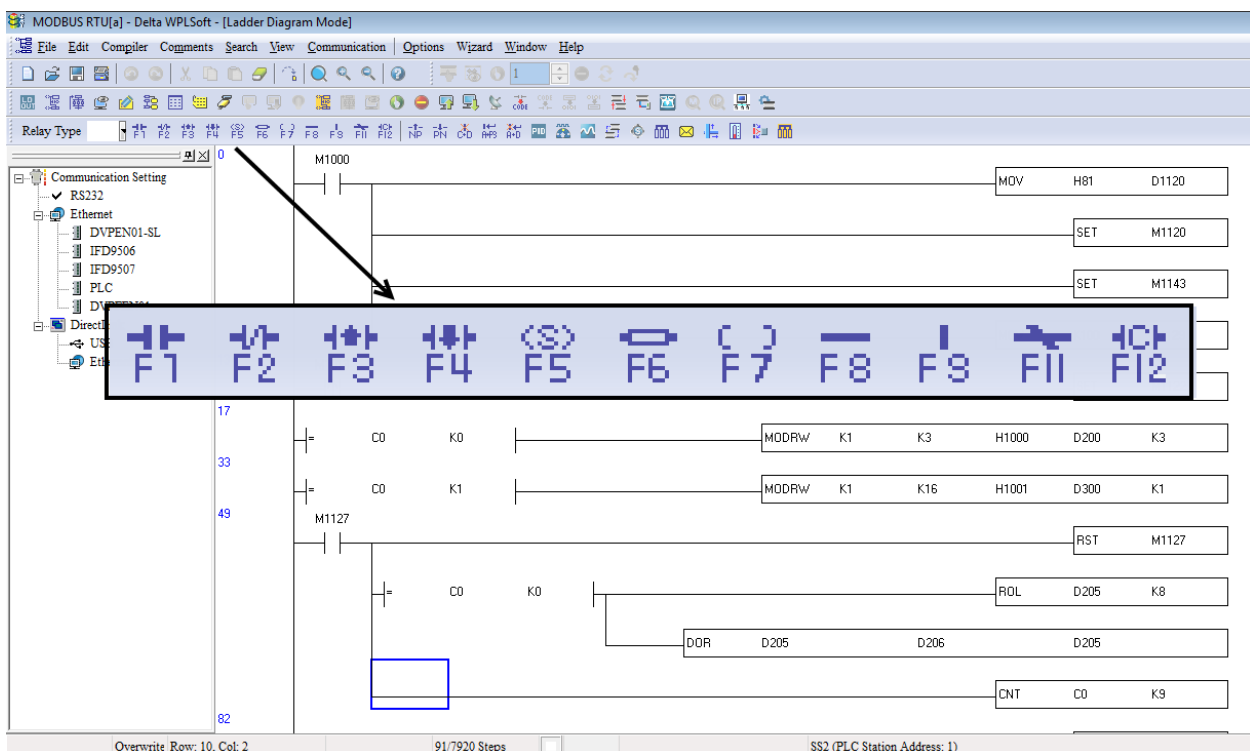
**LED** های وضعیت ورودی ( $X_n$ ): چنانچه هر یک از ورودی های دیجیتال فعال شوند نشانگر مربوط به آن نیز روشن می شود.

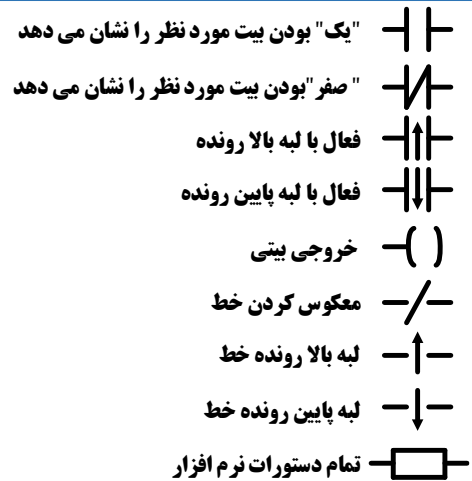
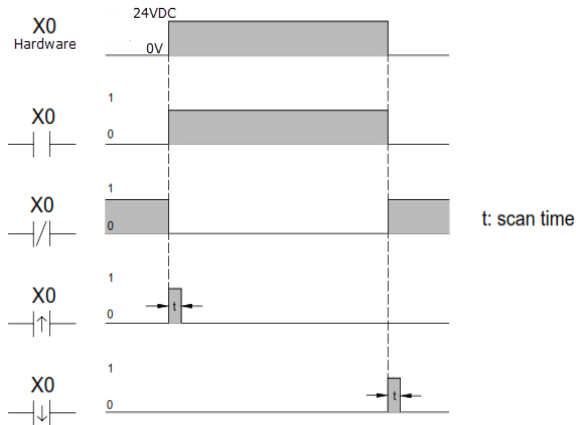
**LED** های وضعیت خروجی ( $Y_n$ ): چنانچه از طریق برنامه هر کدام از خروجی های دیجیتال فعال شود، نشانگرهای مربوط به آن خروجی روشن می شود.

نرم افزارهای برنامه نویسی PLC های دلتا ISPSOFT, WPLSOFT می باشند. این نرم افزارها را می توان از سایت شرکت دلتا ([www.deltaww.com](http://www.deltaww.com)) بطور رایگان دانلود کرد.

از جمله امکانات این نرم افزارها می توان به قابلیت های زیر اشاره کرد:

- ❖ اصلاح برنامه در حالت online
- ❖ تست و مانیتور کردن برنامه در حالت شبیه ساز بدون نیاز به PLC
- ❖ زبان برنامه نویسی LADDER بسیار ساده

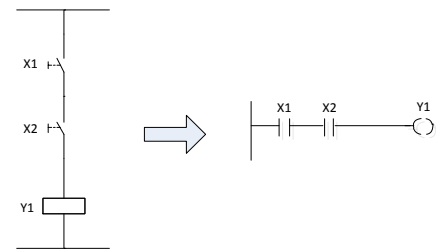




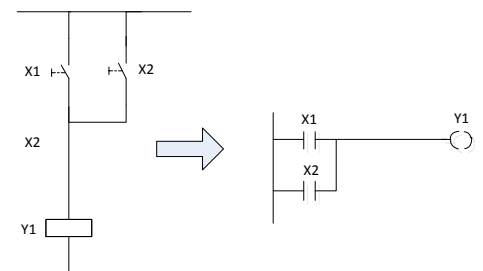
در PLC برنامه از سمت چپ به راست و از بالا به پایین اجرا می گردد و بعد از اجرای کامل برنامه ، اجرای آن دوباره از سر گرفته می شود توجه فرمایید که اگرچه شکل

ظاهری دیگرام نردبانی در مدارهای فرمان و برنامه های PLC یکسان است اما نحوه ی پردازش آن ها متفاوت می باشد.

❖ مثال 1) کاربری AND: در اینجا خروجی Y0 تنها وقتی فعال می شود که هر دو ورودی X0 و X1 وصل شده باشند.



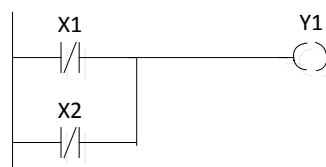
❖ مثال 2) کاربری OR: در این جا خروجی در صورتی فعال می شود که هر کدام از ورودی های A یا B یا هر دو وصل شوند.



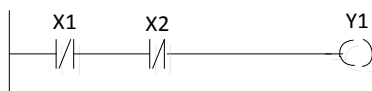
❖ مثال 3) کاربری NOT: در این وضعیت بوبین خروجی Y0 و ورودی X0 عکس یکدیگر می باشند.



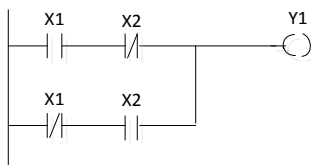
❖ مثال 4) کاربری NAND



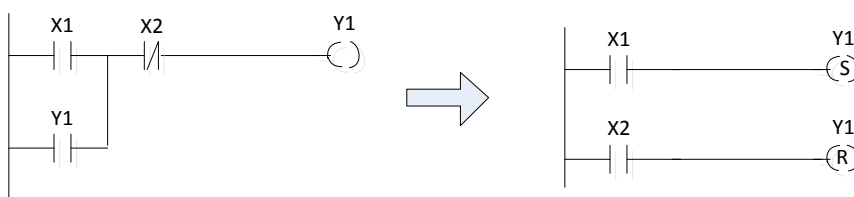
❖ مثال 5 ( کاربری NOR



❖ مثال 6 ( کاربری XOR



مثال 7 ( مدار خود نگهدار



برنامه نویسی PLC به زبان LADDER :

نحوه ی انجام عملیات در سیستم PLC به صورت زیر است :

PLC تمام ورودی ها را چک می کند (Scan Inputs)، ورودی هایی که وصل هستند از نظر PLC معادل "یک" و ورودی هایی که قطع هستند معادل "صفر" قرار داده می شوند .

CPU برنامه موجود در حافظه را خط به خط خوانده و اجرا می کند و پس از پایان اجرای برنامه ، وضعیت خروجی ها را به واحد خروجی می فرستد و این سیکل مجدداً از ابتدا آغاز می شود.

کل زمان انجام این مراحل را Scan Time می نامند ، (Scan Inputs + Scan Program+ Scan Outputs)

چنانچه این زمان بیشتر از 0.2 ثانیه گردد، نشان دهنده ی این مطلب می باشد، که یکی از قسمت های PLC دچار اشکال شده بنابراین تایمر سگ نگهدار (Watch Dog Timer) عمل نموده و تمامی خروجی ها را غیرفعال می کند تا عملکرد اشتباه PLC منجر به حادثه نگردد. مقدار این زمان را می توان با استفاده از رجیستر D1000 تغییر داد .

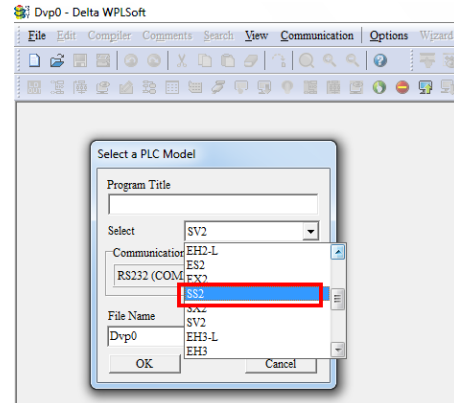
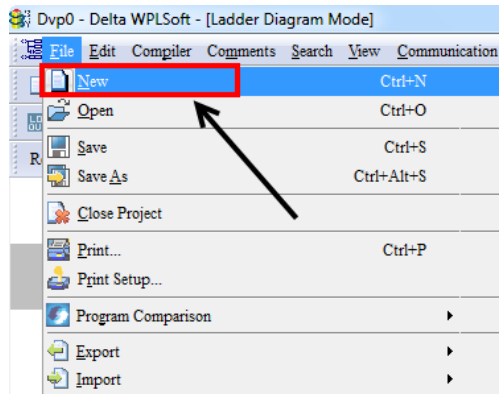
فرض کنید که در یک برنامه باید با وصل یک ورودی ، یک خروجی فعال گردد. حال اگر تصادفاً ورودی در لحظه ای وصل شود که PLC ، مرحله خواندن ورودی ها را به انجام رسانده باشد، در این صورت باید به اندازه ی یک اسکن کامل صبر کند تا وضعیت این ورودی به CPU انتقال یابد ، این تاخیر را تاخیر نرم افزاری PLC می نامند.

از طرف دیگر به دلیل نویزهای موجود در محیط های صنعتی ، ورودی ها عموماً دارای فیلتری می باشند که این نیز به نوبه ی خود تاخیری را در دریافت ورودی ایجاد می نماید (حدود 10ms)، همچنین اگر خروجی از نوع رله ای باشد مدت زمانی حدود 10ms نیز برای وصل رله ی خروجی خواهیم داشت ، مجموع این دو زمان را تاخیر سخت افزاری PLC می نامند.

بنابراین پاسخ زمانی PLC حاصل جمع تاخیر نرم افزاری و سخت افزاری موجود در آن می باشد.

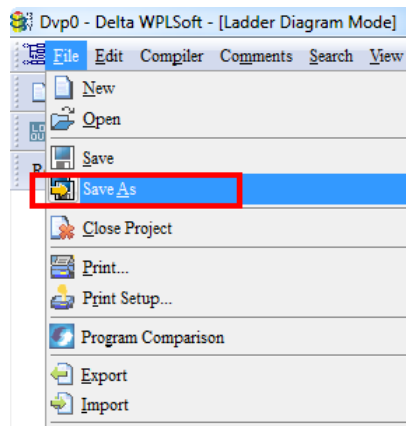
ایجاد پروژه جدید :

از منوی File و انتخاب گزینه New و انتخاب نوع PLC می توان پروژه جدیدی را ایجاد کرد و در آن برنامه مورد نظر را نوشت .

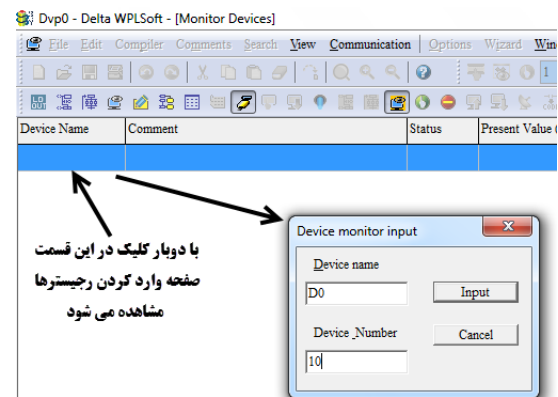
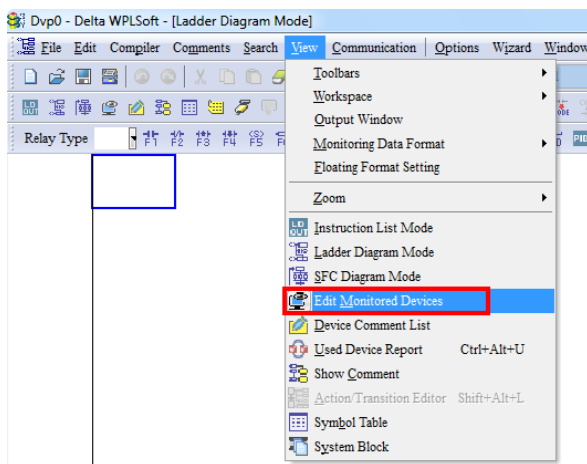


ذخیره برنامه :

در منوی File با استفاده از گزینه Save As می توان برنامه را بعنوان یک فایل با فرمت .dvp در کامپیوتر ذخیره کرد .



مانیتور کردن رجیسترها و ورودی ها و خروجی های PLC : با انتخاب گزینه Edit Monitored Devices از منوی View می توان مقادیر رجیسترها را مشاهده یا تغییر داد.



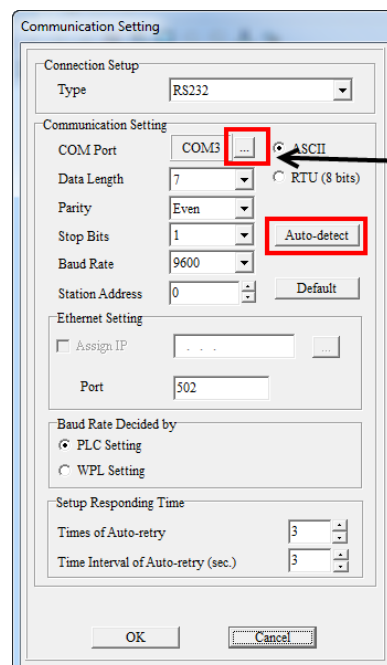
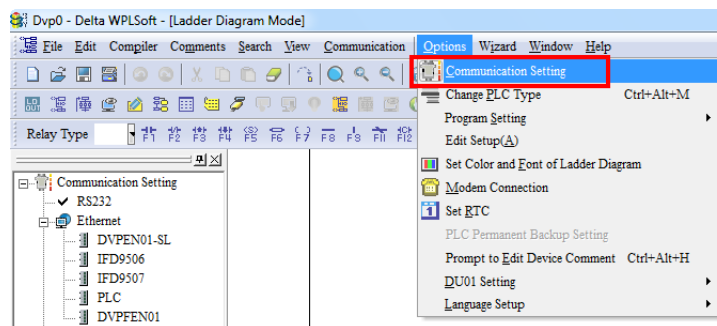
با وارد کردن نام اولین رجیستر و تعداد رجیستر بعد از آن می توانیم رجیسترهای مورد نیاز را در صفحه وارد کنیم



Device Name	Comment	Status	Present Value (16 bits)	Present Value (32 bits)	Floating Point	Format
D0			K0	K0	F0.000	Signed Decimal
D1			K0	K0	F0.000	Signed Decimal
D2			K0	K0	F0.000	Signed Decimal
D3			K0	K0	F0.000	Signed Decimal
D4			K0	K0	F0.000	Signed Decimal
D5			K0	K0	F0.000	Signed Decimal
D6			K0	K0	F0.000	Signed Decimal
D7			K0	K0	F0.000	Signed Decimal
D8			K0	K0	F0.000	Signed Decimal
D9			K0	K0	F0.000	Signed Decimal

انتقال برنامه به PLC

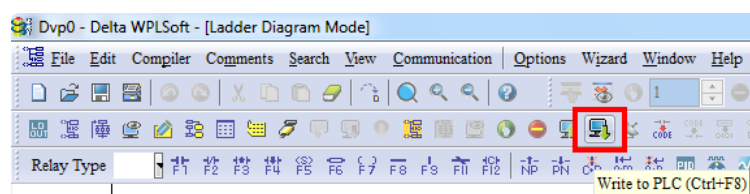
ابتدا باید بین PLC و کامپیوتر ارتباط برقرار کنیم :

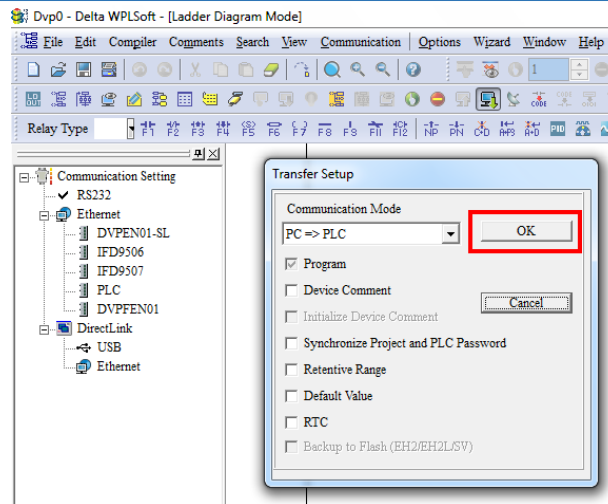


تعیین شماره پورت کامپیوتری که به PLC متصل است

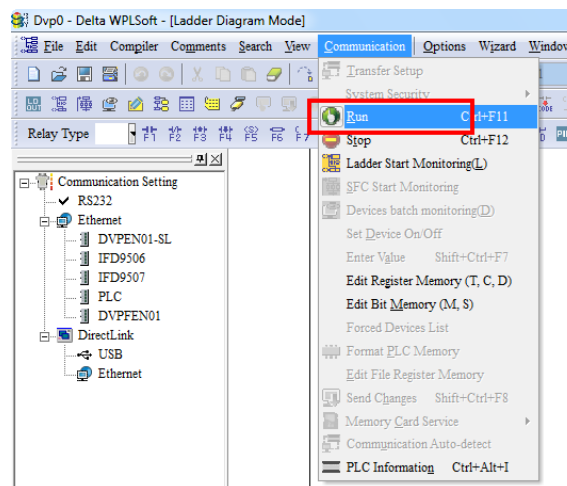
با انتخاب این گزینه می توان از ارتباط بین PLC و کامپیوتر اطمینان حاصل کرد

بعد از اطمینان از ارتباط مناسب بین PLC و کامپیوتر، با انتخاب گزینه Write to PLC می توان برنامه را به PLC انتقال داد.



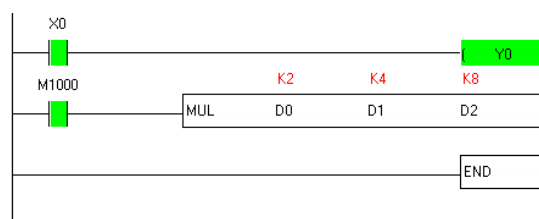


با انتخاب گزینه RUN از منوی ، Communication می توان برنامه را اجرا کرد.



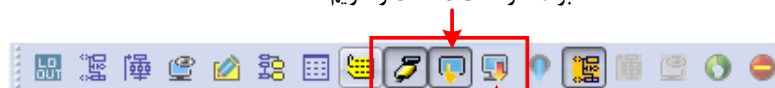
حالت On Line:

با انتخاب این گزینه، می توان برنامه نوشته شده را در حالت Online به PLC وصل کرد و مقادیر رجیسترها و حالت های ورودی و خروجی را مشاهده کرد



تغییر دادن برنامه در حالت Online:

در حالت Online با انتخاب این گزینه به CPU اعلام می کنیم که قصد تغییر برنامه در حالت Online را داریم



پس از تغییر برنامه، با انتخاب این گزینه، برنامه به PLC منتقل می شود.

### رجیسترهای داخلی PLC

رجیسترها بصورت تک بیتی یا 16بیتی قابل دسترسی می باشند (رجیسترهای 32بیتی از استفاده دو رجیستر 16بیتی بوجود می آیند)

رجیسترها دارای دو دسته هستند :

- ❖ رجیسترهای معمولی : این رجیسترها توسط برنامه نویس در برنامه برای نگهداری اطلاعات و اعداد قابل استفاده هستند و بر دو دسته می باشند
- 1 **General** : مقدار موجود در این رجیسترها با قطع و وصل برق صفر نمی شوند و ماندگار می مانند.
- 2 **Latched** : مقدار موجود در این رجیسترها با قطع و وصل برق پاک ( صفر ) می شوند .
- ❖ رجیسترهای خاص : (Special Registers) این رجیسترها رابط بین اطلاعات CPU و برنامه کاربر هستند برای مثال ساعت و تاریخ داخلی PLC در رجیسترهای مشخصی در دسترسی می باشند.
- ❖ (تمام رجیسترهای خاص از مسیر **HELP >> PLC Instruction and Special Registers** قابل دسترسی می باشند)

رجیسترهای بیتی :

X	ورودی	X0 ~ X377
Y	خروجی	Y0 ~ Y377

DVP-28SV11(T/R)2	DVP12SE11(T/R)	DVP20SX211(T/R)	DVP12SA211(T/R)	DVP14SS211(T/R)	مشخصات
M0 ~ M499, (500)	M0~M511, (512) M768~M999, (232) M2000~M2047, (48)		M0~M511, (512) M768~M999, (232) M2000~M2047, (48)		حافظه بیتی عمومی برای استفاده در برنامه
M500 ~ M999, (500) M2000 ~ M4095, (2096)	M512~M767, (256) M2048~M4095, (2048)		M512~M767, (256) M2048~M4095, (2048)		حافظه بیتی حافظه دار برای استفاده در برنامه
	M1000~M1999, (1000)				حافظه بیتی خاص

تعدادی از رجیسترهای بیتی خاص :

شماره رجیستر بیتی خاص	توضیحات
M1000	در صورت RUN بودن PLC ، این بیت 1 خواهد شد
M1002	وقتی CPU از حالت STOP به RUN یا از حالت خاموش به روشن می رود ، فقط در اسکن اول برنامه این بیت 1 می شود و تا آخر صفر می ماند.
M1031	با یک شدن این بیت، تمام رجیسترهای non-latched صفر می شوند
M1032	با یک شدن این بیت، تمام رجیسترهای latched صفر می شوند
M1034	با یک شدن این بیت خروجی های سخت افزاری خاموش می شوند ولی وضعیت خروجی ها در برنامه در همان حالت باقی می مانند.
M1011	بوسیله این بیت ، CPU هر 0.01 ثانیه یک پالس تولید می کند.
M1012	بوسیله این بیت ، CPU هر 0.1 ثانیه یک پالس تولید می کند.
M1013	بوسیله این بیت ، CPU هر 1 ثانیه یک پالس تولید می کند.
M1014	بوسیله این بیت ، CPU هر 60 ثانیه یک پالس تولید می کند.

رجیسترهای 16 بیتی :

DVP-28SV11(T/R)2	DVP12SE11(T/R)	DVP20SX211(T/R)	DVP12SA211(T/R)	DVP14SS211(T/R)	مشخصات
D0 ~ D199, (200)	D0~D407, (408) D600~D999, (400) D3920~D4999, (5880)		D0~D407, (408) D600~D999, (400) D3920~D4999, (1080)		حافظه 16 بیتی عمومی برای استفاده در برنامه
-	D5000~D9799, (4800)		D5000~D9799, (4800)	-	
-	D10000~D11999, (2000)	-	-	-	
D200 ~ D999, (800) D2000 ~ D9999, (8000)	D408~D599, (192) D2000~D3919, (1920)		D408~D599, (192) D2000~D3919, (1920)		حافظه 16 بیتی حافظه دار برای استفاده در برنامه
	D1000~D1999, (1000)				حافظه 16 بیتی خاص

مقدار عددی که می توان به رجیسترهای 16 بیتی اختصاص داد، در بین بازه های : [0~65535] یا [-32768~32767] می باشد.

با استفاده از دو رجیستر 16 بیتی در کنار یکدیگر، می توان رجیستری 32 بیتی داشت.

برای اینکار باید، از توابع 32 بیتی استفاده کرد. اگر در کنار توابع 16 بیتی یک حرف D اضافه کنیم، تابع بصورت 32 بیت استفاده می شود. برای مثال دستور ADD بصورت 16 بیت بوده ولی دستور DADD بصورت 32 بیت می باشد.

تعدادی از رجیسترهای 16 بیتی خاص :

❖ (تمام رجیسترهای خاص از مسیر PLC Instruction and Special Registers >> HELP قابل دسترسی می باشند)

Register No.	توضیحات
D1000	مقدار زمان تایمر watchdog (پیش فرض 200ms)
D1056~D1059	رجیسترهای ورودی آنالوگ در سری EX , SX
D1116 , D1117	رجیسترهای خروجی آنالوگ در سری EX , SX
D1313	"پارامتر ثانیه" مربوط به ساعت داخلی CPU
D1314	"پارامتر دقیقه" مربوط به ساعت داخلی CPU
D1315	"پارامتر ساعت" مربوط به ساعت داخلی CPU
D1316	"پارامتر روز" مربوط به تاریخ داخلی CPU
D1317	"پارامتر ماه" مربوط به تاریخ داخلی CPU
D1318	چندمین روز هفته
D1319	"پارامتر سال" مربوط به تاریخ داخلی CPU
D1010	زمان اسکن برنامه

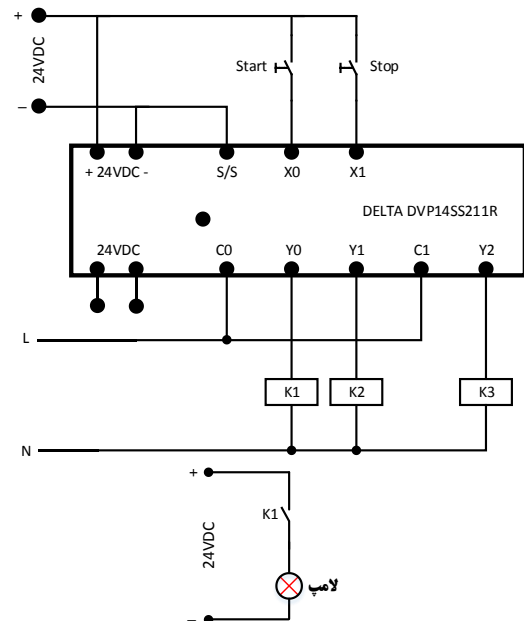
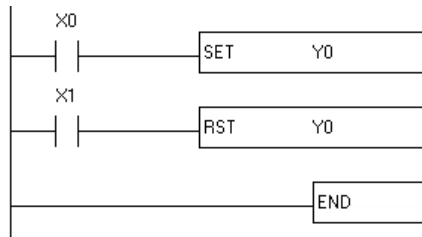
توابع :

ردیف	دستورات 16 بیتی	دستورات 32 بیتی	شرح
۰	CJ	-	پرش به لیبل خاص
۱	CALL	-	فراخوانی زیربرنامه
۲	SRET	-	پایان زیربرنامه
۳	IRET	-	پایان وقفه
۴	EI	-	فعال کردن وقفه ها
۵	DI	-	غیرفعال کردن وقفه ها
۶	FEND	-	پایان برنامه اصلی (تفکیک برنامه اصلی و زیربرنامه)
۷	WDT	-	ریست کردن تایمر Watchdog
۸	FOR	-	شروع حلقه FOR-NEXT
۹	NEXT	-	پایان حلقه FOR-NEXT
۱۰	CMP	DCMP	مقایسه محتوای یک رجیستر با رجیستر دیگر
۱۱	ZCP	DZCP	مقایسه محتوای یک رجیستر با دو رجیستر
۱۲	MOV	DMOV	کپی محتوای یک رجیستر به رجیستر دیگر
۱۳	SMOV	-	پس از شبفت دادن بیتهای رجیستر، به رجیستر دیگر انتقال می دهد
۱۴	CML	DCML	یک ها به صفر و صفر هارا به یک تبدیل می کند
۱۵	BMOV	-	کپی محتوای چند رجیستر به چند رجیستر دیگر
۱۶	FMOV	DFMOV	کپی محتوای یک رجیستر به چند رجیستر دیگر
۱۷	XCH	DXCH	مقدار محتوای دو رجیستر را با هم عوض می کند
۱۸	BCD	DBCD	تبدیل عدد باینری به BCD
۱۹	BIN	DBIN	تبدیل عدد BCD به باینری
۲۰	ADD	DADD	جمع کردن محتوای دو رجیستر با یکدیگر
۲۱	SUB	DSUB	تفریق کردن محتوای دو رجیستر با یکدیگر
۲۲	MUL	DMUL	ضرب کردن محتوای دو رجیستر با یکدیگر
۲۳	DIV	DDIV	تقسیم کردن محتوای دو رجیستر با یکدیگر
۲۴	INC	DINC	افزایش محتوای رجیستر
۲۵	DEC	DDEC	کاهش محتوای رجیستر
۲۶	WAND	DAND	اجرای دستور AND بین بیتهای متناظر دو رجیستر
۲۷	WOR	DOR	اجرای دستور OR بین بیتهای متناظر دو رجیستر
۲۸	WXOR	DXOR	اجرای دستور XOR بین بیتهای متناظر دو رجیستر
۲۹	NEG	DNEG	تولید عدد مکمل 2 محتوای یک رجیستر
۱۱۴	MUL16	MUL32	ضرب دو رجیستر 32/16 بیتی
۱۱۵	DIV16	DIV32	تقسیم دو رجیستر 32/16 بیتی
۳۰	ROR	DROR	چرخش بیتهای رجیستر به راست
۳۱	ROL	DROL	چرخش بیتهای رجیستر به چپ
۳۲	RCR	DRCR	چرخش به راست با بیت Carry
۳۳	RCL	DRCL	چرخش به چپ با بیت Carry
۳۴	SFTR	-	شبفت دادن بیتهای رجیستر به راست
۳۵	SFTL	-	شبفت دادن بیتهای رجیستر به چپ
۳۶	WSFR	-	انتقال محتوای رجیسترها به یکدیگر
۳۷	WSFL	-	انتقال محتوای رجیسترها به یکدیگر
۴۰	ZRST	-	صفر کردن یک بازه از رجیسترها

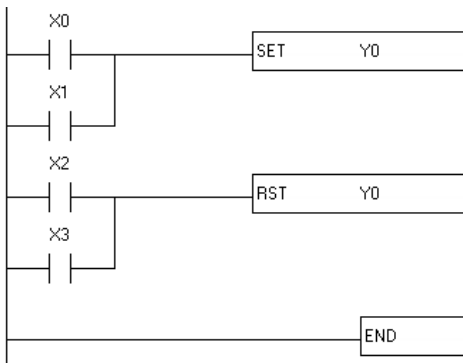
دیگد کردن	-	DECO	۴۱
انکد کردن	-	ENCO	۴۲
شمارش تعداد بیت‌های 1	DSUM	SUM	۴۳
مشاهده وضعیت یک یا صفر بودن یک بیت خاص از یک رجیستر	DBON	BON	۴۴
میانگین گرفتن از چند رجیستر	DMEAN	MEAN	۴۵
جذر گرفتن	DSQR	SQR	۴۸
تبدیل محتوای یک رجیستر از فرمت عدد صحیح به عدد اعشاری	DFLT	FLT	۴۹
رفرش کردن ورودی‌ها و خروجی‌های دیجیتال	-	REF	۵۰
رفرش کردن ورودی‌ها و خروجی‌های دیجیتال و تنظیم فیلترهای مربوطه	-	REFF	۵۱
درايو کردن صفحه کلید 16 تایی ماتریکسی	-	MTR	۵۲
مقایسه کننده‌ها در شمارنده‌های سرعت بالا و روشن کردن یک بیت	DHSCS	-	۵۳
مقایسه کننده‌ها در شمارنده‌های سرعت بالا و خاموش کردن یک بیت	DHSCR	-	۵۴
مقایسه کننده‌ها در یک بازه در شمارنده‌های سرعت بالا	DHSZ	-	۵۵
تعیین سرعت در شمارنده‌های سرعت بالا	-	SPD	۵۶
تولید پالس برای سرو	DPLSY	PLSY	۵۷
تولید پالس PWM	-	PWM	۵۸
تولید پالس برای سرو	DPLSR	PLSR	۵۹
جستجو در مقادیر چند رجیستر	DSEER	SER	۶۱
ثبت زمان روشن بودن یک ورودی	-	TTMR	۶۴
تایمر 16 بیت Off/On Delay	-	STMR	۶۵
با یکبار فعال شدن، خروجی را معکوس می‌کند	-	ALT	۶۶
مقدار آنالوگ با سرعتی زیاد شده و سپس کم می‌شود	DRAMP	RAMP	۶۷
انتقال بایتهای کم ارزش دو رجیستر به یک رجیستر 16 بیتی	-	DTM	۶۸
مرتب کردن رجیسترها براساس مقدار محتوای هر کدام	DSORT	SORT	۶۹
استفاده از 10 ورودی دیجیتال و تبدیل هر کدام به یک عدد	DTKY	TKY	۷۰
استفاده از 4 ورودی و 4 خروجی و اتصال صفحه کلید 16 تایی	DHKY	HKY	۷۱
استفاده از 4 ورودی و 4 خروجی و اتصال به 16 دیپ سویچ	-	DSW	۷۲
دیگدر برای یک سون سگمنت	-	SEGD	۷۳
دیگدر برای اتصال به چندین سون سگمنت	-	SEGL	۷۴
تبدیل 8 کاراکتر به کد کاراکتری	-	ASC	۷۶
ارسال کدهای کاراکتری به خروجی‌های دیجیتال	-	PR	۷۷
خواندن مقادیر رجیسترهای کارتهای اکسپنشن	DFROM	FROM	۷۸
نوشتن بر رجیسترهای کارتهای اکسپنشن	DTO	TO	۷۹
ارسال/دریافت اطلاعات از طریق COM	-	RS	۸۰
دستور حلقه PID	DPID	PID	۸۸
خروجی لبه بالا رونده	-	PLS	۸۹
لبه بالا رونده	-	LDP	۹۰
لبه پایین رونده	-	LDF	۹۱
تایمر 16 بیت	-	TMR	۹۶
کانتر	DCNT	CNT	۹۷
معکوس کردن بیت	-	INV	۹۸
خروجی لبه پایین رونده	-	PLF	۹۹
خواندن اطلاعات از شبکه مودباس	-	MODRD	۱۰۰
نوشتن اطلاعات بر رجیسترها در شبکه مودباس	-	MODWR	۱۰۱
دستور تولید LRC	-	LRC	۱۰۷

دستور تولید CRC	-	CRC	۱۰۸
خواندن / نوشتن توسط پورت اترنت	-	ETHRW	۱۱۳
خواندن / نوشتن اطلاعات در شبکه مودباس	-	MODRW	۱۵۰
مقایسه دو عدد اعشاری	DECMP	-	۱۱۰
مقایسه یک عدد اعشاری با دو عدد اعشاری دیگر	DEZCP	-	۱۱۱
کپی یک رجیستر با فرمت عدد اعشاری به رجیستر دیگر	DMOV	-	۱۱۲
تبدیل زاویه به رادیان	DRAD	-	۱۱۶
تبدیل رادیان به زاویه	DDEG	-	۱۱۷
جمع کردن دو عدد اعشاری	DEADD	-	۱۲۰
تفریق دو عدد اعشاری	DESUB	-	۱۲۱
ضرب کردن دو عدد اعشاری	DEMUL	-	۱۲۲
تقسیم کردن دو عدد اعشاری بر یکدیگر	DEDIV	-	۱۲۳
دستور LN برای اعداد اعشاری	DLN	-	۱۲۵
دستور LOG برای اعداد اعشاری	DLOG	-	۱۲۶
دستور جذر گرفتن اعداد اعشاری	DESQR	-	۱۲۷
به توان رساندن اعداد اعشاری	DPOW	-	۱۲۸
تبدیل اعداد اعشاری به فرمت اعداد صحیح	DINT	INT	۱۲۹
سینوس اعداد اعشاری	DSIN	-	۱۳۰
کسینوس اعداد اعشاری	DCOS	-	۱۳۱
تانژانت اعداد اعشاری	DTAN	-	۱۳۲
آرکسینوس اعداد اعشاری	DASIN	-	۱۳۳
آرک کوسینوس اعداد اعشاری	DACOS	-	۱۳۴
آرک تانژانت اعداد اعشاری	DATAN	-	۱۳۵
سینوس هیپربولیک اعداد اعشاری	DSINH	-	۱۳۶
کسینوس هیپربولیک اعداد اعشاری	DCOSH	-	۱۳۷
تانژاند هیپربولیک اعداد اعشاری	DTANH	-	۱۳۸
دستور مقایسه ساعت داخلی PLC	-	TCMP	۱۶۰
دستور مقایسه در یک بازه برای ساعت داخلی PLC	-	TZCP	۱۶۱
خواندن ساعت و تاریخ داخلی PLC	-	TRD	۱۶۶
تنظیم ساعت و تاریخ داخلی PLC	-	TWR	۱۶۷
دستور بررسی مساوی بودن دو رجیستر	DLD=	LD=	۲۲۴
دستور بررسی بزرگتر بودن دو رجیستر	DLD>	LD>	۲۲۵
دستور بررسی کوچکتر بودن دو رجیستر	DLD<	LD<	۲۲۶
دستور بررسی نامساوی بودن دو رجیستر	DLD<>	LD<>	۲۲۸
دستور بررسی کوچکتر یا مساوی بودن دو رجیستر	DLD<=	LD<=	۲۲۹
دستور بررسی بزرگتر یا مساوی بودن دو رجیستر	DLD>=	LD>=	۲۳۰
خواندن یک بیت از یک رجیستر 16 بیتی	DBOUT	BOUT	۲۶۶
یک کردن یک بیت از رجیستر 16 بیتی	DBSET	BSET	۲۶۷
صفر کردن یک بیت از رجیستر 16 بیتی	DDBRST	BRST	۲۶۸
دستور بررسی مساوی بودن دو عدد اعشاری	FLD=	-	۲۷۵
دستور بررسی بزرگتر بودن دو عدد اعشاری	FLD>	-	۲۷۶
دستور بررسی کوچکتر بودن دو عدد اعشاری	FLD<	-	۲۷۷
دستور بررسی نامساوی بودن دو عدد اعشاری	FLD<>	-	۲۷۸
دستور بررسی کوچکتر مساوی بودن دو عدد اعشاری	FLD<=	-	۲۷۹
دستور بررسی بزرگتر مساوی بودن دو عدد اعشاری	FLD>=	-	۲۸۰

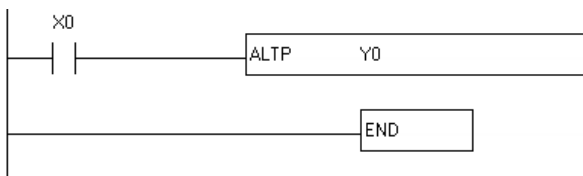
مثال 1: برنامه ایی بنویسید که با فشار یک شستی لامپ روشن شده و با فشار شستی دیگر لامپ خاموش شود.



مثال 2) برنامه ایی بنویسید که با فشار هر یک از شستی های X0 یا X1 موتور Y0 روشن شده و با فشار هر یک از شستی های X2 یا X3 موتور خاموش شود.

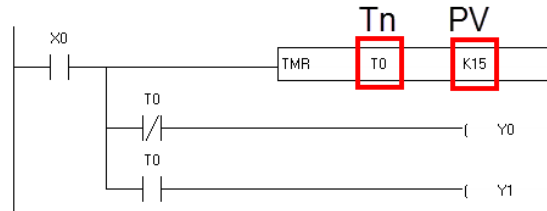


مثال 3: برنامه ایی بنویسید که با فشار یک شستی خروجی Y0 روشن شده و با فشار دوباره همان شستی لامپ خاموش شود





TIMER



Tn : شماره تایمر

PV : بارگذاری مقدار شمارش تایمر

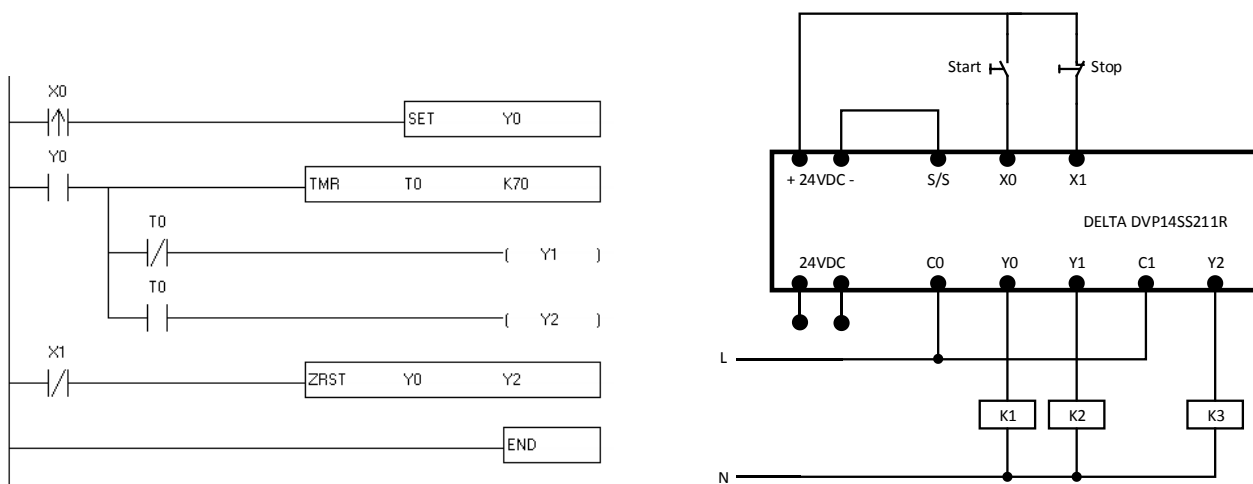
تعداد تایمرها مجموعاً 256 عدد است. (T0~T255). سه زمان پایه (TB) وجود دارد با مقادیر 0.01s , 0.1s , 1s  
T0~T183 با دقت 0.1 ثانیه می باشند.

محاسبه زمان تایمر بدین صورت می باشد: زمان تایمر = TB×PV

هرگاه پایه "EN" تایمر فعال شود (1 شود) تایمر فعال می شود، از مقدار 0 شروع به افزایش می کند تا زمانی که CV بعد از رسیدن به مقدار PV، دیگر اضافه نمی شود و بر روی همان مقدار ثابت می ماند و بیت تایمر (Timer Flag) فعال می شود.

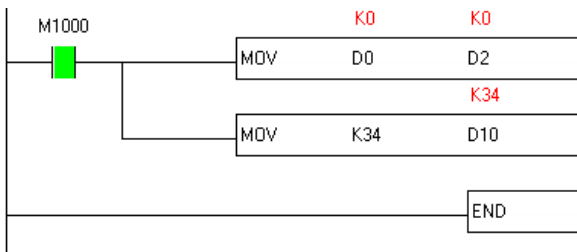
مثال ( روشن کردن موتور بصورت ستاره مثلث :

بعد از فشردن شستی استارت کنتاکتورهای Y0 و Y1 روشن می شوند، بعد از گذشت 7 ثانیه کنتاکتور Y1 خاموش شده و کنتاکتور Y2 روشن شود. با فشردن شستی استپ تمام خروجی ها خاموش شوند.

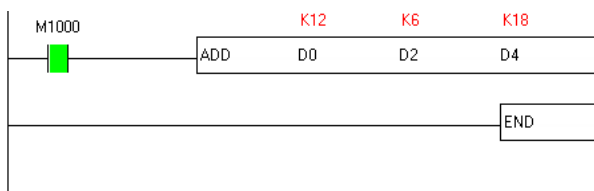


برخی توابع پر کاربرد

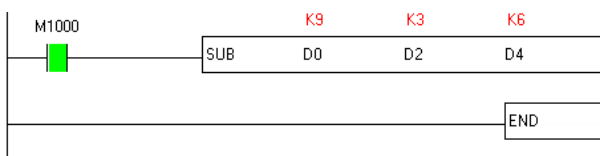
✓ **MOV**: کپی کردن مقدار یک رجیستر در رجیستر دیگر یا قرار دادن یک مقدار در یک رجیستر :



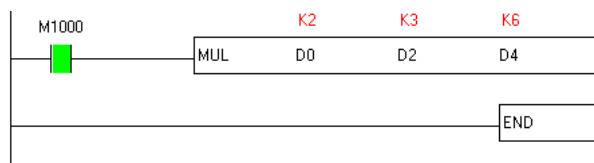
✓ تابع **ADD**: در مثال زیر مقدار دو رجیستر D0 و D2 با یکدیگر جمع شده و در رجیستر D4 ذخیره می شود:  $D4 = D0 + D2$



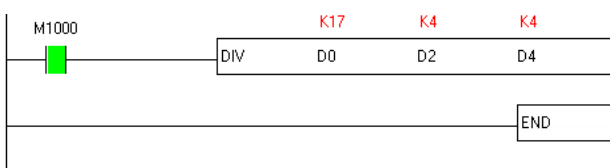
✓ تابع **SUB**: در مثال زیر مقدار D2 از D0 کم می شود و در رجیستر D4 ذخیره می شود:  $D4 = D0 - D2$



✓ تابع **MUL**: در مثال زیر، رجیسترهای D0, D2 در یکدیگر ضرب شده و نتیجه در رجیستر D4, D5 ذخیره می شود

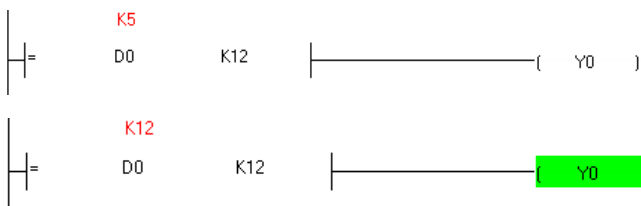


✓ تابع **DIV**: در مثال زیر، رجیستر D0 بر D2 تقسیم شده و نتیجه در رجیستر D4 و باقیمانده در رجیستر D5 ذخیره می شود.

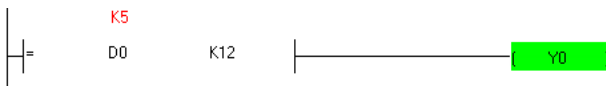


دستور مقایسه :

✓ **LD=** : شرط مساوی بودن : در مثال زیر چنانچه محتوای رجیستر **D0** عدد **12** باشد، خروجی روشن شده و در غیر این صورت خروجی خاموش می شود.



✓ **LD<>** : شرط نامساوی بودن : در مثال زیر چنانچه محتوای رجیستر **D0** عدد **12** باشد، خروجی خاموش شده و در غیر این صورت خروجی روشن می شود.



**LD<** : شرط کوچکتر بودن : در مثال زیر چنانچه محتوای رجیستر **D0** کوچکتر از عدد **12** باشد، خروجی روشن شده و در غیر این صورت خروجی خاموش می شود.

