

به نام خدا  
آموزش PLC دلتا  
قسمت سوم

✓ کانترها:

به جرات میتوان گفت کانترها جزء پرکاربردترین دستورات در برنامه نویسی PLCها هستند.

در PLC های دلتا نیز کانترهای مختلفی برای کاربردهای متفاوت وجود دارد.

در کل کانترها به بخش های زیر تقسیم میشوند:

\* کانترهای ۱۶ بیتی

\* کانترهای ۳۲ بیتی

\* کانترهای سرعت بالا

\* کانترهای نرم افزاری

\* کانترهای سخت افزاری

نکته: اگر مقدار صحیح عددی در یک رجیستر یا حافظه بین بازه ی ۳۲۷۶۷ - ۳۲۷۶۸ بود، این عدد یک عدد ۱۶ بیتی است.

کانتر نیز مانند تایمر دارای تیغه داخلی میباشد با این فرق که این تیغه در کانتر با C نمایش داده میشود.

کانترها نیز مانند تایمرها، رجیسترها، حافظه های داخلی و... دارای جدول مشخصات در فایل منوال مختص هر PLC میباشد. در زیر نمونه جدول کانترها به تفکیک نوع کانتر (R۱۱R۱۴) آورده شده است.

			16-bit count up	C0~C111, 112 points, (*1)		Total 233 points		
				C128~C199, 72 points, (*1)				
				C112~C127, 16 points, (*2)				
				C200~C223, 24 points, (*1)				
			32-bit count up/down	C224~C232, 9 points, (*2)		Total 22 points		
				32bit high-speed count up/down	Soft-ware		C235~C242, 1 phase 1 input, 8 points, (*2)	
							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)						

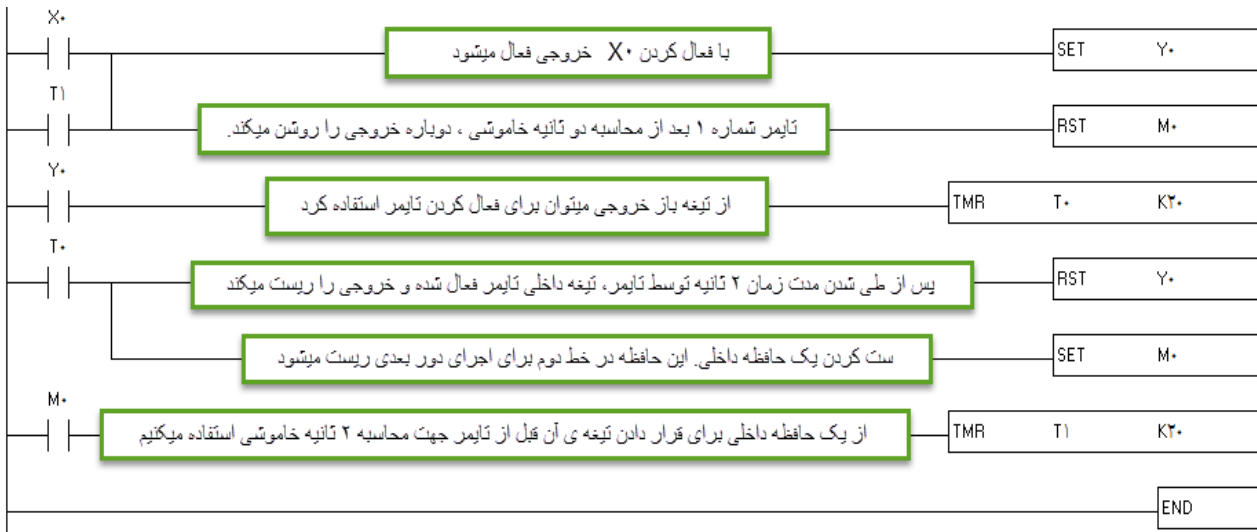
مثال: میخواهیم به ازای شمردن تعداد ۱۰ محصول توسط سنسور، خروجی دستگاه فعال شود

در این برنامه در صورتی که X۰، ۱۰ بار روشن و خاموش شود (۱۰ پالس به کانتر ارسال شود) تیغه داخلی کانتر (C۰) فعال شده و خروجی ۷۰ را روشن میکند.

نکته: مقدار تعیین شده برای کانتر در مثال بالا (K۱۰) است که این مقدار، یک مقدار حقیقی میباشد و ضریب ندارد.

مثال ترکیبی :

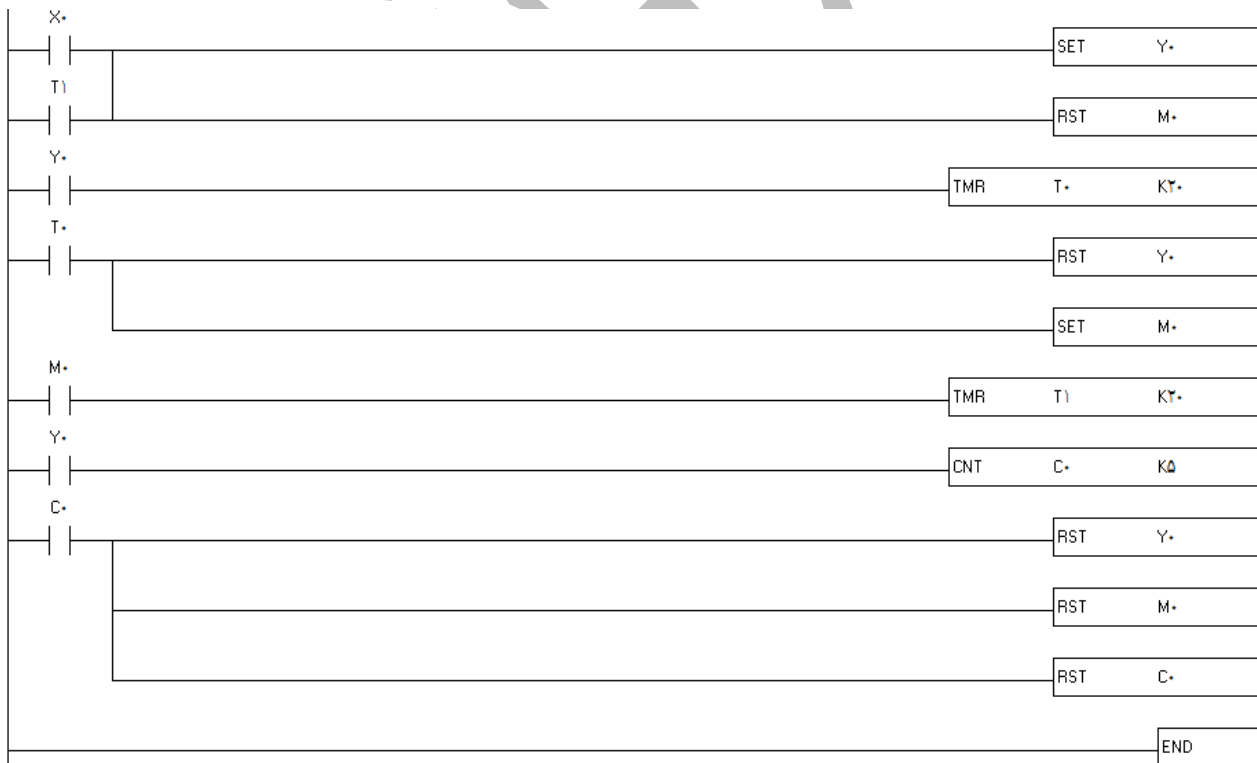
در اکثر صنایع یک فعالیت خاص به صورت تکراری انجام میشود. میخواهیم با فعال کردن یک ورودی، یک خروجی به صورت دائم روشن و خاموش شود. (چراغ چشمک زن)



نکته: در این برنامه خروجی  $Y_0$ ، ۲ ثانیه روشن و ۲ ثانیه خاموش است. برای ۲ ثانیه روشن بودن میتوان از تیغه  $Y_0$  استفاده کرد و تایمر را فعال نگه داشت (مقدار تایمر با قطع جریان ورودی صفر میشود) اما برای محاسبه ی ۲ ثانیه خاموشی میبایست از یک حافظه ی داخلی ( $M$ ) استفاده کنیم و با ست کردن آن، از تیغه ی آن برای فعال نگه داشتن تایمر برای محاسبه ۲ ثانیه خاموشی استفاده کنیم.

این حافظه داخلی بعد از انجام کار باید ریست شده تا برنامه برای سیکل اسکن بعدی آماده باشد لذا این حافظه در خط دوم (جایی که  $M$  فعال نیست) ریست میشود.

حال میخواهیم برنامه ی بالا را تعمیم داده و از یک کانتر استفاده کنیم تا در صورت ۵ بار خاموش و روشن شدن، کل سیستم خاموش شود.



در این برنامه خروجی  $Y_0$  در حال روشن و خاموش شدن است. بنابراین به ازاء هر بار روشن و خاموش شدن، یک پالس به کانتر  $C_0$  ارسال میکند. با رسیدن مقدار کانتر به عدد تنظیم شده ( $K_5$ )، تیغه کانتر فعال شده و  $Y_0$  و  $M_0$  را ریست میکند. همچنین خود کانتر را هم ریست کرده تا برای استارت بعدی مقدار درون آن صفر باش

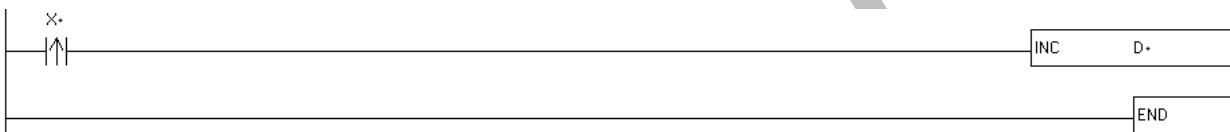
## ✓ دستور INC و DEC :

گاهی اوقات نمیخواهیم تعداد مشخصی محصول تولید کنیم و فقط میخواهیم بدانیم در کل چند محصول تولید کرده ایم. در این موارد میتوان از دستور INC برای شمارش صعودی و از DEC برای شمارش نزولی استفاده کرد.

در نظر داشته باشید هنگام استفاده از این دستور باید از ورودی با لبه بالارونده یا پایین رونده استفاده شود تا فقط یک پالس در هر تحریک به دستور برسد. شمار میتواند این برنامه را بدون استفاده از تیغه با لبه بالارونده یا پایین رونده استفاده کنید تا موضوع و دلیل لزوم استفاده از لبه در تیغه را متوجه شوید.

لبه بالارونده LDP : این تیغه ، تنها یک پالس از خود عبور میدهد. تصور کنید که ورودی شما به صورت یک تیغه با لبه بالارونده است و به یک شاستی استارت متصل است. وقتی این شاستی را فشار میدهید همزمان با فشرده شدن یک پالس ارسال میشود. حال اگر از لبه پایین رونده (LDF) استفاده کنید، با فشار دادن شاستی ، اتفاقی رخ نمیدهد ولی هنگام رها کردن شاستی یک پالس ارسال میشود.

به مثال زیر توجه کنید:



به ازای هر بار فعال شدن X0 یک پالس به دستور INC ارسال میشود و مقدار رجیستر D0 یک عدد اضافه میشود.

## ✓ مقایسه کننده ها :

در برنامه نویسی گاهی نیاز است که مقداری را با مقداری دیگر مقایسه کنیم. برای مثال میخواهیم اگر تعداد محصولی که تولید کردیم (از جلوی سنسور عبور کرده) بیشتر از یک مقدار مشخص شد، دستگاه خاموش شده و آلارم مخصوص سرویس و نگهداری به صدا در آید. یا مثلاً میخواهیم اگر مقدار دمای محیط از دمایی که ما تعیین کردیم بیشتر یا کمتر یا مساوی بود، خروجی متناسب با آن وضعیت فعال شود. در این مواقع میتوان با استفاده از دستورات مقایسه کننده این کار را انجام داد.

برای برنامه نویسی در PLC های دلتا انواع مقایسه کننده ها موجود است که میتوان برای راحتی کار از آنها استفاده کرد.

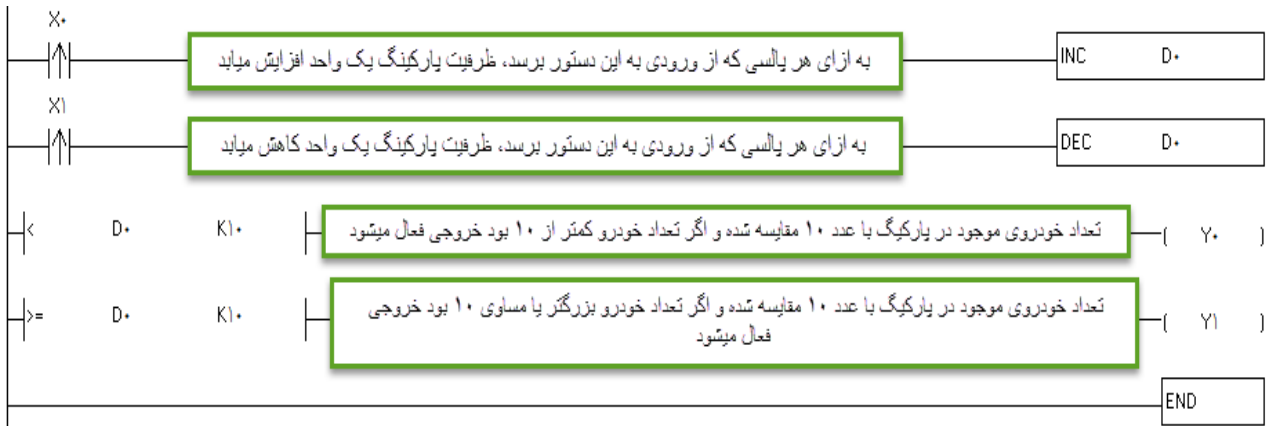
در این قسمت قصد داریم تا به تعدادی از این دستورات مقایسه کننده بپردازیم.

## ۱ - دستورات مقایسه ای بر پایه ی LD :

این دستور دارای انواع مختلف بوده و به شکل های زیر مورد استفاده قرار میگیرد.

- LD= •
- LD< •
- LD> •
- LD<> •
- LD<= •
- LD>= •

برای درک بهتر این دستور به مثال ترکیبی زیر توجه کنید:  
 پارکینگ را در نظر بگیرید. ظرفیت این پارکینگ ۱۰ خودرو میباشد  
 در ورودی پارکینگ سنسور X۰ تعبیه شده تا تعداد خودرو های ورودی را بشمارد.  
 در خروجی پارکینگ نیز سنسور X۱ تعبیه شده تا تعداد خودروهای خروجی را بشمارد.  
 تعداد خودروی باقی مانده در پارکینگ توسط دستورات مقایسه کننده مقایسه شده و خروجی متناسب فعال میشود.  
 در صورتی که تعداد خودرو در پارکینگ، از ۱۰ خودرو کمتر باشد خروجی Y۰ که به تابلوی "وارد شوید" متصل است فعال میشود.  
 در صورتی که تعداد خودرو در پارکینگ، بیشتر یا مساوی ۱۰ خودرو باشد خروجی Y۱ که به تابلوی "ظرفیت تکمیل است" متصل است فعال میشود.

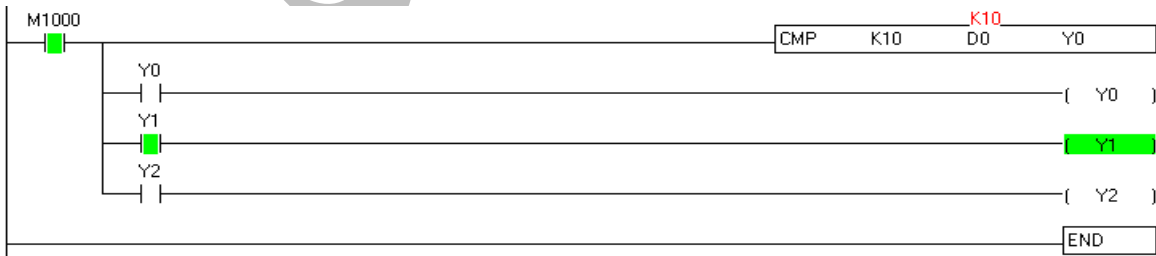
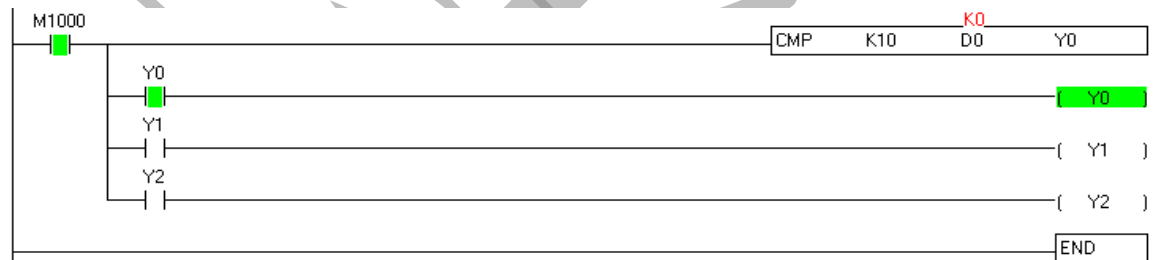


در این مثال D۰ ظرفیت پارکینگ است.

## ۲ - دستور CMP

این دستور نیز یکی دیگر از دستورات مقایسه ای میباشد.

در این مثال در صورتی که عدد داخل رجیستر D۰ کمتر از عدد ثابت تعیین شده (K۱۰) باشد خروجی اول Y۰ فعال میشود.  
 در صورتی که عدد داخل رجیستر D۰ برابر با عدد ثابت تعیین شده (K۱۰) باشد خروجی دوم Y۱ فعال میشود.  
 در صورتی که عدد داخل رجیستر D۰ بیشتر از عدد ثابت تعیین شده (K۱۰) باشد خروجی سوم Y۲ فعال میشود.



در آخرین قسمت این دستور از Y0 استفاده شده است. باید این را در نظر بگیرید که این دستور خروجی های Y0 و Y1 و Y2 را اشغال میکند.

همچنین در ابتدای برنامه میبینید که از M1000 استفاده شده است که در بحث های مربوط به حافظه های خاص به آن خواهیم پرداخت.

### ۳ - دستور ZCP :

این دستور هم یکی از دستورات مقایسه میباشد.

این دستور، دستور مقایسه ی ناحیه ای میباشد.

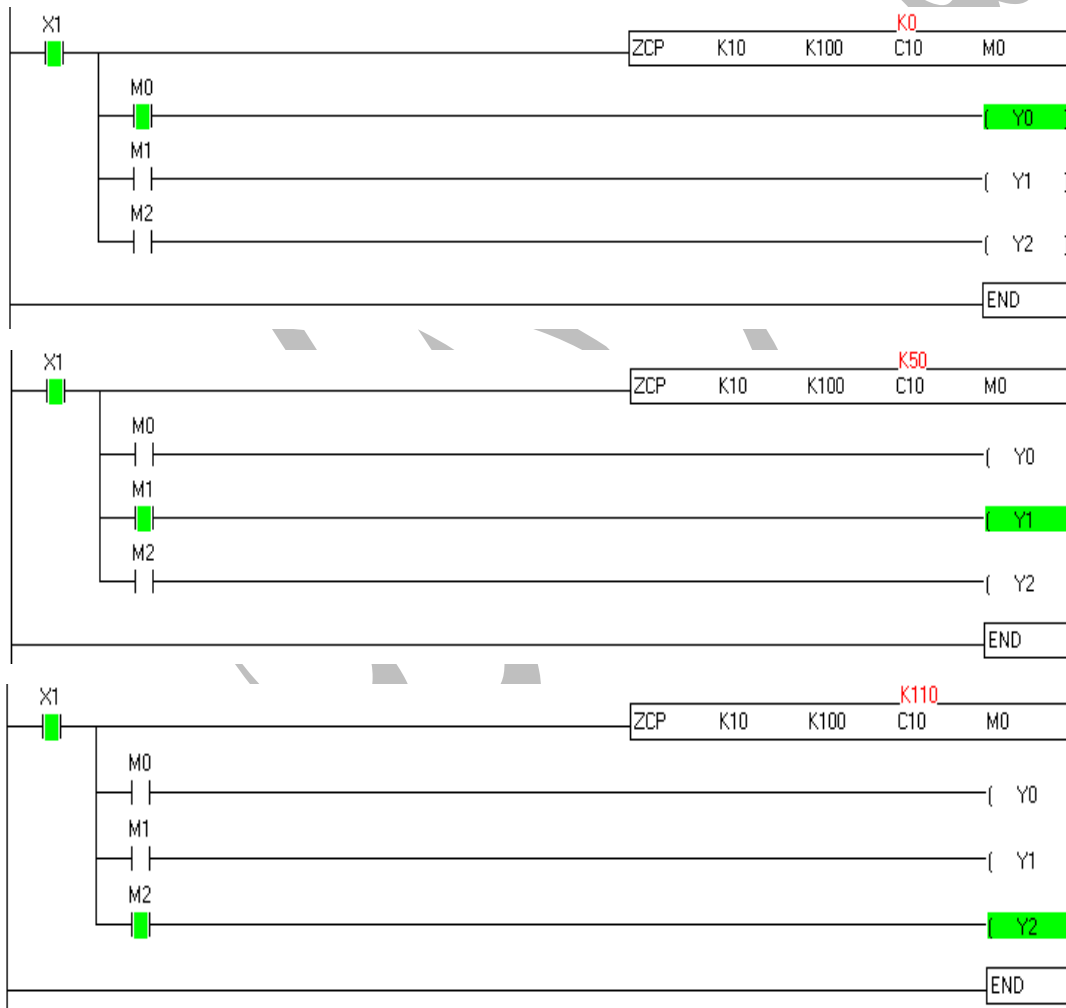
به مثال زیر توجه کنید:

در این مثال میخواهیم مقدار خوانده شده توسط کانتر C10 را با دو عدد K10 و K100 مقایسه کنیم.

در صورتی که مقدار شمرده شده توسط کانتر از عدد ثابت ۱۰ کمتر بود Y0 روشن میشود

در صورتی که مقدار شمرده شده توسط کانتر مساوی یا بین اعداد ۱۰ تا ۱۰۰ بود، Y1 روشن میشود

در صورتی که مقدار شمرده شده توسط کانتر از عدد ثابت ۱۰۰ بزرگتر بود، Y2 روشن میشود.



در صورت داشتن هرگونه سوال در رابطه با مباحث گفته شده میتوانید با شماره ۰۹۱۲۷۶۶۸۵۶۱ تماس بگیرید.

پایان قسمت سوم

مقصودی

شارکنترل