



SABCO

Specialized Automation Control
Systems & Electronic Industries

پروژه‌های کاربردی **PLC** های سری **DVP**

Delta



تأليف: مهندس مصطفی رحمتی

مهندس حسین حسینی



پروژه‌های کاربردی PLC های سری DVP دلتا

مؤلف: مصطفی رحمنی - حسین رحمانی



انتشارات قدیس

WWW.Qeddīs.com

سرشناسه	رحمنی، مصطفی، 1365 -
عنوان و نام پدیدآور	پروژه‌های کاربردی PLC های سری DVP دلتا/مولف مصطفی رحمنی، حسین رحمانی.
مشخصات نشر	تهران: قدیس، 1396.
مشخصات ظاهری	75ع.ص.
شابک	978-600-8050-66-7
وضعیت فهرست نویسی	فیپا
موضوع	کنترل‌کننده‌های برنامه‌پذیر
موضوع	Programmable controllers
شناسه افزوده	رحمانی، حسین، 1366 -
رده بندی کنگره	223TJ / 26,9,1396
رده بندی دیویی	629/896
شماره کتابشناسی ملی	4869969



انتشارات قدیس

پروژه‌های کاربردی PLC های سری DVP دلتا

مؤلفان: مصطفی رحمنی - حسین رحمانی

ناشر: قدیس

لیتوگرافی، چاپ و صحافی: سحر گرافیک، محدث

نوبت و سال چاپ: اول، 1396

شمارگان: 500 نسخه

شابک: 978-600-8050-66-7

حق چاپ محفوظ و منحصراً مخصوص ناشر است.

دفتر مرکزی و مرکز پخش:

نشر قدیس: تهران، میدان انقلاب، خیابان 12 فروردین، پایین تر از وحید نظری، نبش بن بست حقیقت، پلاک 4

واحد 5 تلفن: 66411381-66403548-66715255

آزمون تاپ: خیابان آزادی-زارع-پلاک ۱۹-طبقه ۲ تلفن: ۰۹۱۹۵۹۶۶۶۷۹-۰۲۱۶۶۰۱۵۲۳۴

WWW.AZMOONTOP.COM



تقدیم به

روح پر فتوح پدرم، که در دوران حیاتش مانند کوهی در تمام طول زندگی حامی من بود.
روحش شاد

تقدیم به

همرم، که در سایه همیاری و همدلی او به این مقصود نائل شدم.



فصل 1 / مثال‌های وابسته به دستورات پایه	15
1-1 نحوه‌ی خواندن برنامه‌ی دیاگرام نردبانی در PLC دلتا	16
2-1 مدار حفاظتی با حق تقدم استوپ	16
3-1 مدار حفاظتی با حق تقدم استارت	16
4-1 مدار حفاظتی با فرمان SET و RESET	17
5-1 کنترل شرطی	17
6-1 کنترل اینترلاک	18
7-1 کنترل ترتیبی	18
8-1 مدار تریگر کننده	19
9-1 مثالی از تشخیص بطری افتاده بر روی کانوایر	19
10-1 کنترل روشنایی راه‌پله در خانه هوشمند	20
11-1 بررسی لبه‌بالارونده پالس ورودی در یک سیکل اسکن برنامه	22
12-1 کنترل خروجی با لبه‌ی بالارونده پالس ورودی در یک سیکل اسکن برنامه	23
13-1 مدار کنترل خودنگهدار	24
14-1 مدار کنترل اینترلاک جهت کنترل ورود و خروج اتومبیل در پارکینگ	25
15-1 مدار قفل‌شونده (خودنگهدار)	27
16-1 مدار SET/RST_Latched و مدار Unlatched	28
17-1 مدار خروجی متناوب با تابع Latched	30
18-1 مدار کنترل شرطی یکی پس از دیگری	31
19-1 مدار حق تقدم (اولین و یا بالاترین اولویت)	33
20-1 کنترل ورود و خروج ماشین به پارکینگ	36
فصل 2 / مثال‌های وابسته به تایمرها و کانترها	39
1-2 بسته‌بندی محصول در حال حرکت بر روی کانوایر	40
2-2 شمارش تعداد محصول تولید شده در روز	41
3-2 محاسبه‌ی مقدار محصول تولید شده‌ی نهایی	43
4-2 طراحی ساعت با استفاده از شمارنده	45
5-2 کنترل جهت چرخش (چپگرد و راستگرد) یک موتور سه فاز آسنکرون	46

48.....	2-6 انتخاب یک برنامه از میان چند برنامه توسط سوئیچ سلکتوری
51.....	2-7 برنامه انتخاب مد کنترل دستی/اتوماتیک
55.....	2-8 برنامه‌ی کنترل دستی/اتوماتیک با استفاده از دستورالعمل STL
9-2	به روز کردن ورودی/خروجی دیجیتال و تنظیم زمان فیلتر ورودی دیجیتال توسط
57.....	REF/REFF
59.....	2-10 مدار نوسان ساز
60.....	2-11 مدار چشمک زن
61.....	2-12 مدار تأخیر زمانی
61.....	2-13 توسعه دادن مقدار زمان تایمر
63.....	2-14 تایمر تأخیر در قطع
65.....	2-15 تایمر تأخیر در وصل
66.....	2-16 تایمر تأخیر در قطع و تأخیر در وصل
67.....	2-17 راه اندازی 3 الکتروموتور به صورت ترتیبی با تایمر تأخیری
70.....	2-18 مدولاسیون پهنای پالس
71.....	2-19 مانیتور کردن سطح آب استخر پرورش ماهی
74.....	2-20 کنترل کوره پخت
76.....	2-21 مدار ستاره مثلث برای کاهش ولتاژ و جریان راه اندازی الکتروموتورها
79.....	2-22 کنترل درب اتوماتیک
81.....	2-23 کنترل همزن مایعات به صورت اتوماتیک
83.....	2-24 دستگاه قهوه ساز اتوماتیک
86.....	2-25 برنامه وقفه اتوماتیک تایمر
88.....	2-26 مقایسه زمان
91.....	فصل 3 / دستورات پیشرفته دیجیتال
92.....	3-1 اعلام آتش سوزی
94.....	3-2 مدار آخرین اولویت
97.....	3-3 برنامه کارواش
99.....	3-4 برنامه تابع آموزش به دستگاه
101.....	3-5 کنترل چراغ چشمک زن رنگی
102.....	3-6 کنترل نمایشگر 7-segment
104.....	3-7 برنامه جمع زدن پیوسته اعداد
106.....	3-8 سیستم اتوماتیک بستن درب های صندوق پول در سوپر مارکت

108.....	3-9 کنترل آب‌نمای فواره‌ای
111.....	3-10 کنترل چراغ راهنمایی و رانندگی
116.....	3-11 برنامه پخت مواد غذایی به همراه تنظیمات پارامتری پخت
119.....	3-12 کنترل سطح آب مخزن
122.....	3-13 مخلوط کن مواد (میکسر)
124.....	3-14 بکاپ‌گیری داده در چندین زمان مختلف
126.....	3-15 کنترل یک رشته لامپ نئون
128.....	3-16 آشکارسازی محصول معیوب بر روی خط تولید
130.....	3-17 مخلوط کردن اتوماتیک چند محصول
134.....	3-18 مکالمه‌ی تلفنی اتاق‌ها توسط سرویس‌دهنده‌ی تماس

فصل 4 / مثال‌های کاربردی وابسته به محیط‌های کنترل پیوسته آنالوگ . 137

138.....	4-1 دستورالعمل‌های وابسته به ماژول آنالوگ به همراه مثال
138.....	4-1-1 دستورالعمل شیب متغیر RAMP
140.....	4-1-2 دستورالعمل خواندن دیتای رجیسترهای کنترلی (CR#) از ماژول‌های توسعه‌ی خاص (FROM)
142.....	4-1-3 دستورالعمل نوشتن داده در رجیستر کنترلی (CR) ماژول‌های توسعه‌ی خاص (TO)
151.....	4-1-4 دستورالعمل محاسبه‌ی مقدار تناسبی SCAL
154.....	4-1-5 دستورالعمل محاسبه‌ی پارامتر مقدار تناسبی (SCLP)
160.....	4-2 یک برنامه‌ی کاربردی از ماژول DVP04AD
160.....	4-2-1 اندازه‌گیری جریان
161.....	4-2-2 تنظیمات سریع ماژول ورودی آنالوگ توسط نرم‌افزار WPLSoft
166.....	4-2-3 اندازه‌گیری سرعت الکتروموتور متصل شده به اینورتر
168.....	4-3 یک برنامه‌ی کاربردی از ماژول خروجی آنالوگ DVP04DA
168.....	4-3-1 خروجی آنالوگ از جنس جریان
169.....	4-3-2 تنظیمات سریع پارامترهای ماژول خروجی آنالوگ توسط نرم‌افزار WPLSoft
173.....	4-3-3 تنظیم فرکانس خروجی اینورتر توسط ماژول D/A
175.....	4-4 مثال کاربردی برای ماژول ورودی/خروجی آنالگ DVP06XA
175.....	4-4-1 خواندن سرعت از اینورتر
177.....	4-4-2 روش تنظیم پارامترهای ماژول DVP06XA توسط نرم‌افزار WPLSoft
182.....	4-4-3 تنظیمات ماژول DVP04PT در نرم‌افزار WPLSoft
186.....	4-5 کاربرد ماژول آنالوگ دما DVP04TC

188.....	1-5-4 پیکربندی ماژول DVP04TC در نرم افزار WPLSoft
193.....	4-6-4 توابع ماژول لودسل
193.....	4-6-1 اندازه گیری وزن
194.....	4-6-2 بررسی پایداری وزن (Standstill check)
196.....	4-2-3 تعیین کردن نقطه (وزن) صفر
196.....	4-6-4 فیلتر کردن وزن
197.....	4-6-5 تنظیمات ماژول DVP02LG-S در نرم افزار LCSOft توسط کامپیوتر
200.....	4-6-6 تنظیمات ارتباطات شبکه
203.....	4-6-7 تنظیمات پارامترهای ماژول لودسل
206.....	4-6-8 تنظیم تصحیح وزن (کالیبره کردن وزن)
207.....	4-6-9 بررسی وضعیت های ماژول
210.....	4-6-10 مراحل تصحیح (کالیبره کردن) وزن
211.....	4-6-11 تصحیح وزن توسط PLC
213.....	4-6-12 تصحیح وزن در نرم افزار LCSOft
215. 8 VRs (Variable Resistors)	4-7-2 کنترل ولتاژ خروجی 2 ماژول DVP-04DA توسط
218.....	4-8-2 کنترل سطح آب
220.....	4-9-2 اندازه گیری جریان آب داخل لوله
222.....	4-10-2 ذخیره کردن مقدار آنالوگ بدست آمده
224.....	4-11-2 مانیتور کردن دمای اتاق
226.....	4-12-2 کنترل شیب حرکت جرثقیل

فصل 5 / کار با کنترلر PID و کنترل فازی در PLC دلتا سری DVP..... 231

232.....	5-1-1 دستورالعمل خروجی PWM عمومی (GPWM)
233.....	5-2-1 دستورالعمل PID
256.....	5-3-1 مثال کاربردی کنترل کننده ی PID
266.....	5-4-1 خودتنظیم اتوماتیک پارامترهای PID کنترلر دما
271.....	5-5-1 کنترل کشش جسمی با استفاده از لودسل توسط دستورالعمل PID
276.....	5-6-1 مثال کاربردی PID توسط ماژول DVP04TC-S
281.....	5-7-1 مثال کاربردی PID توسط ماژول DVP04TC-H2
283.....	5-8-1 دستورالعمل کنترل فازی دما (FTC)
296.....	5-9-1 کنترل فازی دمای کوره

فصل 6 / کنترل موقعیت توسط PLC دلتا 303

6-1 کنترل موقعیت محور بصورت حلقه باز.....304

6-1-1 روش کنترل Rapid/Creep با خروجی دیجیتال برای کنترل سرعت موتور بصورت

کنتاکتوری.....306

6-1-2 موقعیت‌دهی با تغذیه سیستم از طریق مبدل فرکانس (اینورتر).....308

6-2 کنترل موقعیت محور بصورت حلقه بسته.....309

6-3 لیست PLC های دارای ورودی/خروجی سرعت بالا و آنالوگ و پورت شبکه.....310

6-4 مشخصات ورودی خروجی سرعت بالا و توابع کنترل حرکت.....316

6-5 جدول نام ترمینال‌های PLC برای سیم‌بندی ورودی/خروجی‌ها.....321

6-6 سیم‌بندی ترمینال‌های PLC دلتا.....322

6-7 کنترل ماشین برش توسط دستورالعمل DHSCS.....327

6-8 کنترل رنگ چندین قسمت توسط دستورالعمل DHSZ/DHSCR.....329

6-9 اندازه‌گیری سرعت چرخش چرخ گردان توسط دستورالعمل SPD.....330

6-10 برنامه کنترل خط تولید توسط دستورالعمل PLSY.....332

6-11 برنامه کنترل شیربرقی اسپری توسط دستورالعمل PWM.....333

6-12 کنترل زمان شتاب مثبت/شتاب منفی سروموتور توسط دستورالعمل PLSR.....335

6-13 سیستم حرکت دمونستراسیون (تجربی) ساده موقعیت سرودرایوهای AC سری

ASDA دلتا.....338

6-14 شمارنده سرعت بالای پالس از جنس AB-phase.....345

6-15 تنظیمات دستورالعمل کارکرد سرو توسط دستورالعمل CJ.....347

6-16 کنترل حرکت JOG با تنظیمات حرکتی خیلی کم توسط دستورالعمل INC/DEC.....349

6-17 کنترل تغییر مکان معکوس توسط دستورالعمل ENG.....351

فصل 7 / پروژه‌های کاربردی شبکه‌ی RS-485 با پروتکل Modbus استاندارد

و غیراستاندارد DVPSCM52-SL 353

7-1 روش پیشنهادی برقراری شبکه ارتباطات RS-485 با پروتکل MODBUS توسط دلتا

.....354

7-2 اصلاح فرمت ارتباطات پورت COM.....358

7-2-1 اصلاح فرمت ارتباطات پورت COM1.....358

7-2-2 اصلاح فرمت ارتباطات پورت COM2.....358

359.....	3-2-7 اصلاح فرمت ارتباطات پورت COM3
360.....	4-2-7 فعال کردن مد RTU برای پورت COM1 و COM2
361.....	5-2-7 تنظیم پورت COM توسط ابزار جادوی نرم افزار WPLSoft و ISPSOft
364.....	6-2-7 استفاده از وقفه ارتباطات پورت COM2
365.....	3-7 تابع اتصال مودم
367.....	1-3-7 مازول DVP-F232 (RS-232 card)
374.....	4-7 پروتکل PLC LINK
374.....	1-4-7 تبادل داده در شبکه‌ی PLC LINK توسط فلگ M1353
375.....	2-4-7 تبادل داده در شبکه‌ی PLC LINK توسط فلگ M1353 و وقفه‌ی ارتباطات I170
377.....	3-4-7 اتصال یک Master PLC و دو Slave PLC به هم توسط شبکه RS-485 و تبادل دیتا توسط پروتکل PLC LINK
379.....	4-4-7 تبادل داده در شبکه‌ی PLC LINK بین یک PLC و اینورتر VFD-M دلتا
380.....	5-7 تبادل دیتا در شبکه RS-485 با پروتکل Modbus غیراستاندارد
380.....	1-5-7 تبادل دیتا در شبکه‌ی RS-485 توسط دستورالعمل RS
382.....	2-5-7 سوئیچ بین مد 8 بیتی و مد 16 بیتی دستورالعمل RS
384.....	3-5-7 تبادل دیتا بین PLC و اینورتر VFD-B دلتا توسط دستورالعمل RS در مد ASCII
386.....	4-5-7 تبادل دیتا بین PLC و اینورتر VFD-B دلتا توسط دستورالعمل RS در مد RTU
388.....	5-5-7 ارتباط شبکه بین PLC و اینورتر VFD-S دلتا با مد ASCII
390.....	6-5-7 تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل RS در مد RTU و بررسی درستی دیتا توسط دستورالعمل CRC
393.....	6-7 تبادل دیتا در شبکه‌ی RS-485 با پروتکل Modbus استاندارد
393.....	1-6-7 تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل MODBUS در مد ASCII
394.....	2-6-7 تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل MODBUS در مد RTU
396.....	3-6-7 دومین مثال از تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل MODBUS در مد ASCII
398.....	4-6-7 تبادل دیتا بین PLC و اینورتر VFD-B دلتا توسط دستورالعمل MODBUS در مد ASCII
399.....	5-6-7 دومین مثال از تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل MODBUS در مد RTU
400.....	6-6-7 سومین مثال از تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل MODBUS در مد ASCII
402.....	7-6-7 چهارمین مثال از تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل MODBUS در مد ASCII

404.....	8-6-7	تبادل دیتا بین PLC و اینورتر VFD-A دلتا توسط دستورالعمل MODBUS
405.....	7-7	استفاده از کدهای تابعی استاندارد در شبکه Modbus
405.....	1-7-7	خواندن یک دسته دیتا توسط پورت COM2 با استفاده از کد تابع K3(H03)
410.....	2-7-7	نوشتن مقدار در یک دیتا رجیستر توسط پورت COM2 با استفاده از کد تابع K6(H06)
413.....	3-7-7	نوشتن مقدار در یک دسته دیتا رجیستر توسط پورت COM2 با استفاده از کد تابع K16(H10)
417.....	4-7-7	خواندن مقدار در یک دسته ابزار بیتی توسط پورت COM2 با استفاده از کد تابع K2(H02)
420.....	5-7-7	نوشتن مقدار در یک دسته ابزار بیتی توسط پورت COM2 با استفاده از کد تابع K15(H0F)
423.....	6-7-7	نوشتن مقدار در یک ابزار بیتی توسط پورت COM2 با استفاده از کد تابع K5(H05)
428.....	7-7-7	خواندن و نوشتن مقدار در یک دسته ابزار بیتی توسط پورت COM2 با استفاده از کد تابع K23(H17)
432.....	8-7	دستورات اختصاصی دلتا برای راهاندازی سروسیستم‌های دلتا در شبکه Modbus
432.....	1-8-7	راهاندازی سرودرایو دلتا از طریق شبکه RS-485 پورت COM2 (RS-485)
433.....	2-8-7	راهاندازی سرودرایو دلتا از طریق شبکه RS-485 پورت COM3 (RS-485)
435.....	9-7	کاربرد ماژول RTU-485
436.....	1-9-7	توسعه تعداد ورودی/خروجی توسط ماژول RTU-485
438.....	2-9-7	کنترل خروجی دیجیتال Y0 متصل شده به ماژول RTU-485
439.....	3-9-7	اتصال ماژول SCM به شبکه‌ی MODBUS
451.....	4-9-7	اتصال ماژول SCM به نرم‌افزار WPLSoft
452.....	5-9-7	اتصال ماژول SCM به شبکه‌ی ارتباطات RS-485 (مدباس غیر استاندارد)
477.....	10-7	ماژول DVPEN01
477.....	1-10-7	راهاندازی ماژول DVPEN01 با استفاده از نرم‌افزار WPLSoft
482.....	2-10-7	راهاندازی ماژول DVPEN01 با استفاده از نرم‌افزار ISPSoft
487.....	11-7	ماژول DVPEN01-SL
487.....	1-11-7	برنامه‌ی ارسال ایمیل توسط ماژول DVPEN01-SL
490.....	2-11-7	نخستین برنامه‌ی تبادل داده در شبکه توسط ماژول DVPEN01-SL
493.....	3-11-7	دومین برنامه‌ی تبادل داده در شبکه توسط ماژول DVPEN01-SL
495.....	4-11-7	سومین برنامه‌ی تبادل داده در شبکه توسط ماژول DVPEN01-SL
498.....	5-11-7	چهارمین برنامه‌ی تبادل داده در شبکه توسط ماژول DVPEN01-SL
499.....	6-11-7	پنجمین برنامه‌ی تبادل داده در شبکه توسط ماژول DVPEN01-SL
501.....	7-11-7	راهاندازی MODBUS TCP Master

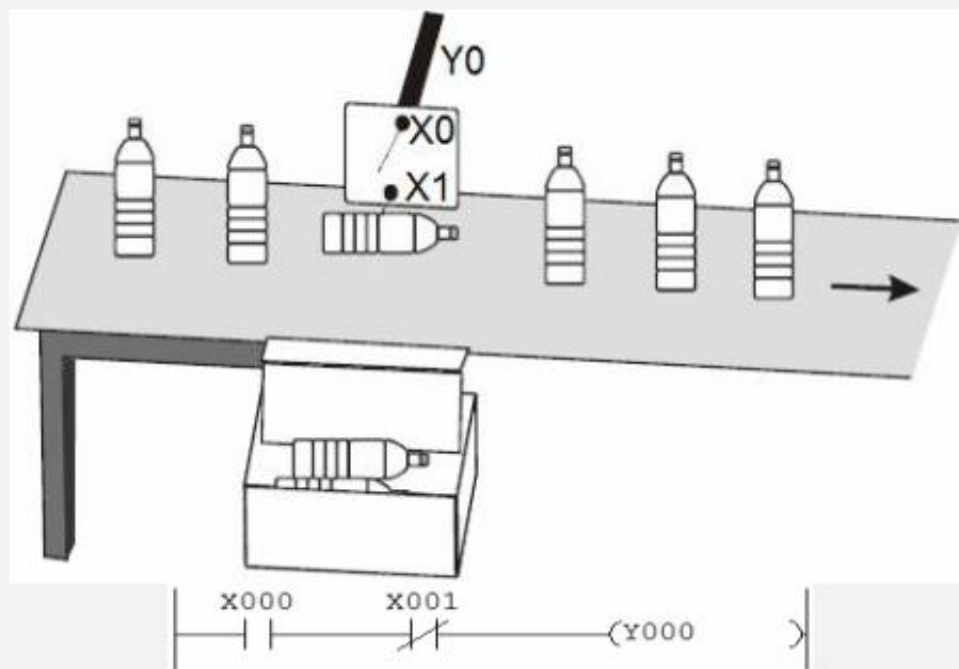
503.....	7- 11- 8 مثالی از RTU Mapping
506.....	7- 11- 9 مثالی از کاربرد پروتکل MELSEC
507.....	7- 12- 7 برقراری ارتباط بین 4 ماژول ترنس미터 با PLC دلتا از طریق پروتکل Modbus
509.....	7- 13- 7 برقراری ارتباط بین PLC و اینورتر سری VFD-M دلتا با دستورالعمل MODRD/MODWR
513.....	7- 14- 7 برقراری ارتباط بین PLC و اینورتر سری VFD-B دلتا با دستورالعمل MODRD/MODWR
517.....	7- 15- 7 برقراری ارتباط بین PLC و اینورتر سری VFD-V دلتا با دستورالعمل MODRD/MODWR
521.....	7- 16- 7 برقراری ارتباط بین PLC و سرودرایو ASD-A دلتا با دستورالعمل MODRD/MODRW در مد موقعیت
525.....	7- 17- 7 برقراری ارتباط بین PLC و سرودرایو ASD-A دلتا با دستورالعمل MODRD/MODRW در مد سرعت
531.....	7- 18- 7 برقراری ارتباط بین PLC دلتا و کنترلر دمای سری DTA دلتا با دستورالعمل MODRD/MODWR
535.....	7- 19- 7 برقراری ارتباط بین PLC دلتا و کنترلر دمای سری DTB دلتا با دستورالعمل MODRD/MODWR/MODRW
540.....	7- 20- 7 برقراری ارتباط بین PLC سری E دلتا به عنوان Master با PLC سری S دلتا به عنوان Slave
544.....	7- 21- 7 برقراری ارتباط بین PLC های سری E دلتا به عنوان Master/Slave
548.....	7- 22- 7 برقراری ارتباط بین PLC دلتا با اینورتر سری VFD-B و سرودرایو سری ASD-A دلتا توسط شبکه‌ی Modbus
553.....	7- 23- 7 برقراری ارتباط بین PLC دلتا با کنترلر فرآیند سری DTA و DTB دلتا توسط شبکه‌ی Modbus
557.....	7- 24- 7 برقراری ارتباط بین دو PLC دلتا با دستورالعمل RS
562.....	7- 25- 7 برقراری ارتباط بین PLC دلتا و اینورتر میکرومستر 420 زیمنس توسط دستورالعمل RS
568.....	7- 26- 7 برقراری ارتباط بین PLC دلتا و اینورتر دانفوس VLT6000 توسط دستورالعمل RS
575.....	7- 27- 7 اتصال لودسل به مبدل مدباس RS2LM و خواندن وزن توسط PLC دلتا در شبکه مدباس

28-7 دریافت اس ام اس از طریق ماژول GSM توسط شبکه Modbus در PLC های دلتا	580
29-7 ارتباط وایرلس چندین وسیله در شبکه‌ی RS-485 با پروتکل Modbus	592
30-7 ارتباط شبکه بین PLC و سرودرایو AC سری ASDA دلتا	593
31-7 ارتباطات سریال بین PLC و سرودرایو AC سری ASDA دلتا	598
فصل 8 / مثال‌های PLC داخلی اینورتر VFD-C2000	605
1-8 ملاحظات قبل از استفاده از PLC داخلی اینورتر	606
2-8 روشن کردن PLC	608
1-2-8 متصل شدن به کامپیوتر	608
2-2-8 ابزارهای ورودی خروجی اینورتر	611
3-8 مثال کاربردی برای ماژول CMC-MOD01	612
4-8 مثال کاربردی از ماژول CMC-EIP01	614
5-8 مثال کاربردی از ماژول CMC-PD01	616
6-8 مثال کاربردی برای ماژول CMC-DN01	640
7-8 مثالی از تقویم PLC داخلی اینورتر	651
8-8 مثال‌هایی از دستورات اختصاصی PLC داخلی اینورتر VFD-C2000	653
1-8-8 مثالی از نوشتن مقدار از PLC داخلی اینورتر به CPU اینورتر	653
2-8-8 مثالی از راه‌اندازی مد کنترل PID اینورتر توسط PLC داخلی اینورتر	654
3-8-8 مثالی از راه‌اندازی مد کنترل سرعت اینورتر توسط PLC داخلی اینورتر	655
4-8-8 مثالی از به کارگیری اینورتر در مد کنترل سرعت توسط PLC داخلی اینورتر	658
5-8-8 مثالی از راه‌اندازی مد کنترل گشتاور اینورتر توسط PLC داخلی اینورتر	661
6-8-8 مثالی از به کارگیری اینورتر در مد کنترل گشتاور توسط PLC داخلی اینورتر	663
7-8-8 مثالی از راه‌اندازی مد کنترل موقعیت نقطه به نقطه اینورتر توسط PLC داخلی اینورتر	665
8-8-8 مثالی از به کارگیری اینورتر در مد کنترل موقعیت و جایگاه‌یابی توسط PLC داخلی اینورتر	666
9-8 مثال‌هایی از دستورات اختصاصی شبکه در PLC داخلی اینورتر VFD-C2000	671
1-9-8 مثالی از راه‌اندازی شبکه‌ی CANopen اینورتر توسط PLC داخلی اینورتر	671
2-9-8 مثالی از ارتباط داخلی CPU اینورتر و PLC CPU داخلی اینورتر	673
3-9-8 به کارگیری CANopen Master Control در PLC داخلی اینورتر	675
4-9-8 کنترل گره (Node) اصلی در ارتباطات داخلی	697
5-9-8 شبکه‌ی Modbus و کنترل از راه دور ورودی/خروجی‌ها توسط دستورالعمل MODRW	702
10-8 تابع شمارش سرعت بالا با استفاده از ورودی دیجیتال M18	711

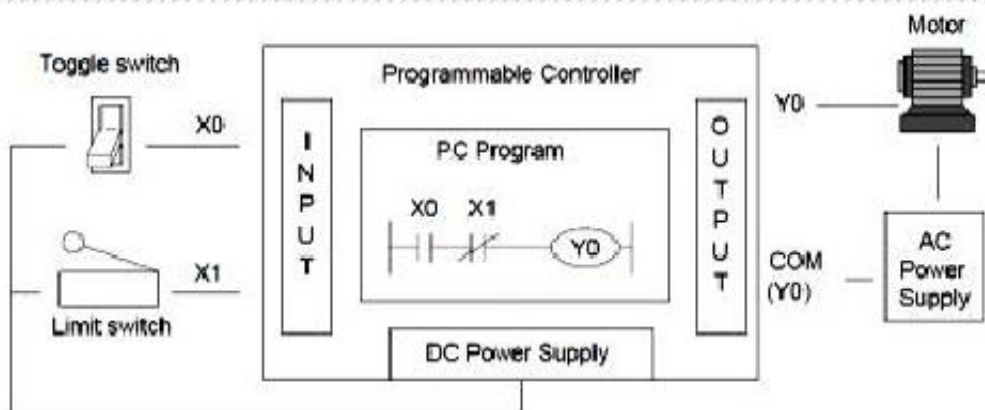
712.....	11-8 تابع محاسبه‌ی فرکانس
715.....	فصل 9 / مانیتورینگ و کنترل تجهیزات آنالوگ در HMI دلتا
716.....	9-1 مقدمه
716.....	9-2 برنامه‌نویسی HMI دلتا
717.....	9-3 ایجاد یک پروژه‌ی جدید
720.....	9-4 اجرای تنظیمات پورت RS-485 در نرم‌افزار WPLSoft برای PLC
722.....	9-5 سیم‌بندی پورت ارتباطات PLC و HMI
722.....	9-6 اصلاح تنظیمات HMI
724.....	9-7 دانلود/آپلود برنامه به/از HMI
727.....	9-8 برنامه‌ی مانیتورینگ و کنترل I/O آنالوگ PLC در HMI
727.....	9-8-1 سخت‌افزار مورد نیاز
728.....	9-8-2 آدرس‌دهی و برنامه‌نویسی در سمت PLC
729.....	9-8-3 برنامه‌نویسی برای PLC
732.....	9-8-4 برنامه‌نویسی برای HMI
741.....	9-9 نمونه برنامه‌ی مانیتورینگ یک کوره
744.....	9-10 یک برنامه‌ی کاربردی جهت مانیتور کردن دو مد موقعیت و سرعت ثابت سروموتور
744.....	9-10-1 طراحی صفحه‌ی Homing Operation
744.....	9-10-2 طراحی صفحه‌ی Position Control 1
745.....	9-10-3 طراحی صفحه‌ی Position Control 2
745.....	9-10-4 طراحی صفحه‌ی JOG Operation
746.....	9-10-5 طراحی صفحه‌ی Speed Control

فصل 1

مثال‌های وابسته به دستورات پایه



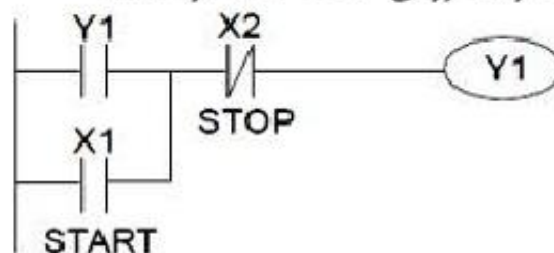
1- نحوه خواندن برنامه دیاگرام نردبانی در PLC دلتا



در این مثال، برای فعال شدن خروجی Y0، باید جریان از کنتاکت‌های سری شده‌ی X0 و X1 عبور کند. بنابراین، با فعال شدن ورودی X0 خروجی Y0 فعال خواهد شد. اگر میکروسوییچ متصل شده به ورودی X1 فعال شود، خروجی Y0 غیرفعال می‌شود. همانطور که متوجه شدید، با قطع شدن خط افقی بین دو خط عمودی، جریان جاری در آن خط افقی قطع شده و بوبین غیرفعال می‌شود.

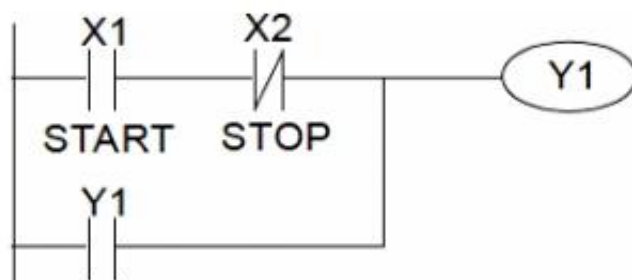
1-2 مدار حفاظتی با حق تقدم استوپ

در این مدار وقتی که ورودی X1=ON و ورودی X2=OFF باشد، خروجی Y1=ON خواهد شد. اگر کنتاکت بسته‌ی X2=ON (فعال) شود، خروجی Y1=OFF خواهد شد. اگر ورودی X1 و X2 به صورت همزمان ON شوند، خروجی Y1=OFF خواهد شد.



1-3 مدار حفاظتی با حق تقدم استارت

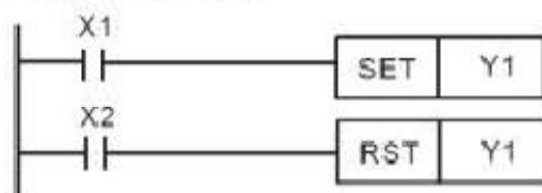
وقتی که کنتاکت X1=ON و کنتاکت X2=OFF باشد، خروجی Y1=ON می‌شود و ON باقی خواهد ماند. اگر ورودی X2=ON شود، خروجی Y1 همچنان ON باقی خواهد ماند.



1- 4- مدار حفاظتی با فرمان SET و RESET

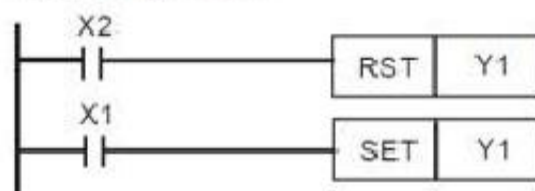
مطابق شکل زیر، از فرمان SET و RESET به صورت ترکیبی و در کنار هم استفاده می‌شود. وقتی که فرمان RST بعد از فرمان SET قرار گیرد، اولویت با استوپ خواهد بود. دلیل این امر آن است که PLC برنامه را از بالا به پایین اجرا خواهد کرد. در این صورت، اگر ورودی X1, X2 با هم فعال شوند، خروجی Y1 غیرفعال خواهد شد.

Top priority of stop



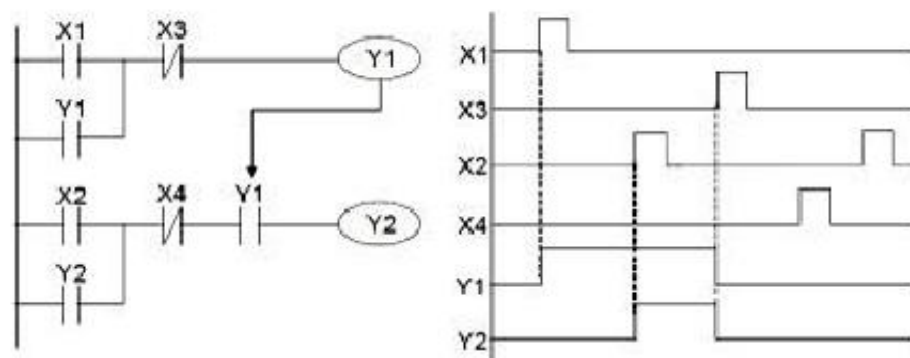
وقتی که فرمان SET بعد از فرمان RST قرار گیرد، اولویت با استارت خواهد بود. در این مثال، اگر ورودی X1 و X2 هر دو با هم فعال شوند، خروجی Y1 فعال خواهد شد.

Top priority of start



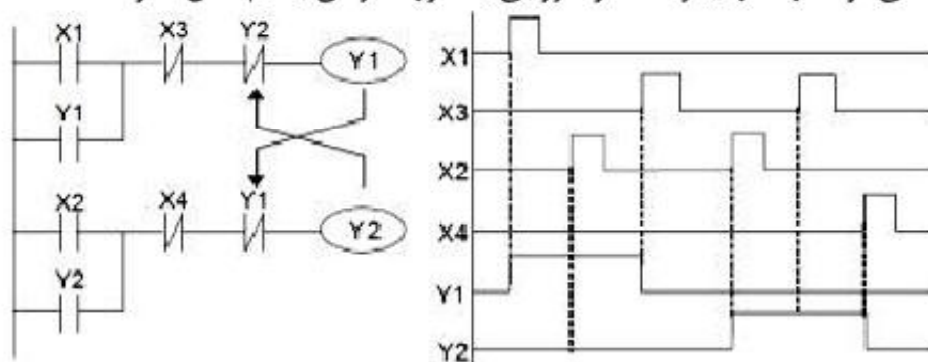
1- 5- کنترل شرطی

از ورودی‌های X1, X3 به ترتیب بعنوان استارت و استوپ خروجی Y1 و از ورودی‌های X2, X4 به ترتیب بعنوان استارت و استوپ خروجی Y2 استفاده می‌شود. کنتاکت باز رله‌ی خروجی Y1 در مسیر رله‌ی خروجی Y2 قرار گرفته است. در این مدار، تا زمانی که ورودی Y1 توسط شستی استارت که به ورودی X1 متصل شده فعال نشود، با فشردن شستی استارت که به ورودی X2 برای فعال کردن خروجی Y2 متصل شده است، فعال نخواهد شد.



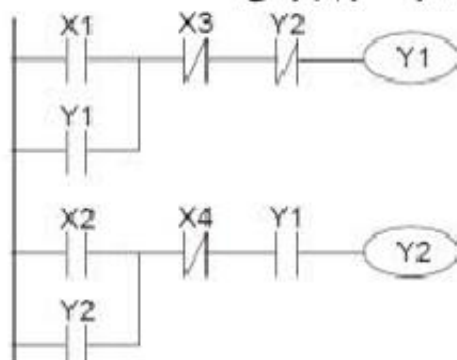
1- 6- کنترل اینترلاک

همانطور که در شکل زیر مشاهده می‌کنید، با فعال شدن یک خروجی، خروجی دیگر فعال نخواهد شد زیرا کنتاکت بسته یک خروجی در مسیر خروجی دیگر قرار دارد. به این عمل، اینترلاک کردن دو مدار با یکدیگر گفته می‌شود. در مدار اینترلاک دو خروجی به طور همزمان با هم فعال نخواهند شد.



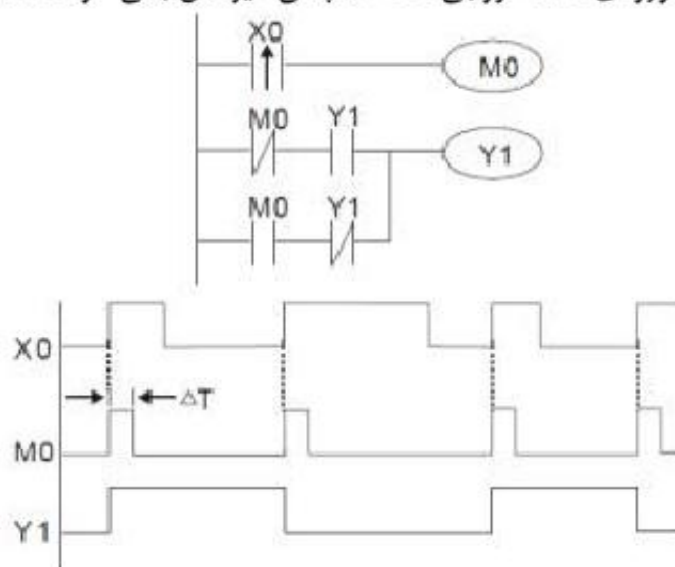
1- 7- کنترل ترتیبی

در این مدار، باید ابتدا خروجی Y1 فعال شود تا بتوان خروجی Y2 را فعال کرد. بعد از اینکه خروجی Y2 را فعال کردید خروجی Y1 غیرفعال خواهد شد. دلیل این امر، سری کردن کنتاکت‌های رله خروجی Y1 و Y2 در مسیر یکدیگر به فرم زیر می‌باشد.



1- 8- مدار تریگر کننده

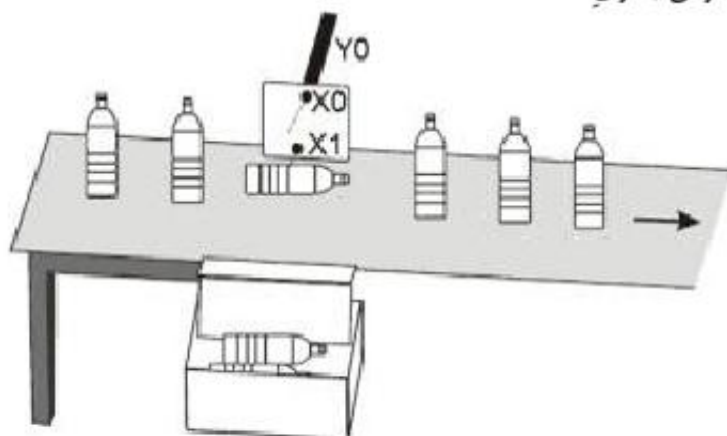
همانطور که در شکل زیر مشاهده می‌کنید، ما توانسته‌ایم با یک ورودی، قطع و وصل یک خروجی را کنترل کنیم. با فعال کردن ورودی X1 خروجی Y1 فعال خواهد شد. با غیرفعال کردن ورودی X1 خروجی Y1 همچنان فعال باقی خواهد ماند. مجدداً با فعال کردن ورودی X1 خروجی Y1 غیرفعال شده و با غیرفعال کردن ورودی X1 خروجی Y1 همچنان غیرفعال باقی خواهد ماند.



1- 9- مثالی از تشخیص بطری افتاده بر روی کانوایر

فرض کنید بر روی یک کانوایر، بطری‌ها در حال انتقال می‌باشند. اگر بر روی این کانوایر یک بطری افتاده باشد، باید به خارج از کانوایر هدایت شود.

در این مثال، برای تشخیص بطری افتاده نیاز به دو سنسور فتوالکتریک (نوری) بوده و از یک شیر برقی هوا برای شوت کردن بطری افتاده استفاده شده است.

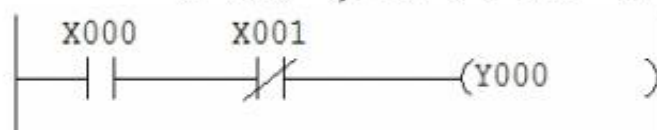


همانطور که متوجه شدید، هدف از کنترل، آشکارسازی بطری افتاده بر روی کانوایر و هدایت کردن (هول دادن) بطری افتاده به خارج از کانوایر می‌باشد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	سیگنال این ورودی زمانی آشکار می‌شود - $X0=ON$ - که سنسور ته بطری را سنس کند
X1	سیگنال این ورودی زمانی آشکار می‌شود - $X1=ON$ - که سنسور سر بطری را سنس کند
Y0	به این خروجی، شیر برقی پنوماتیکی برای باز کردن مسیر هوای شوت کن بطری افتاده متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



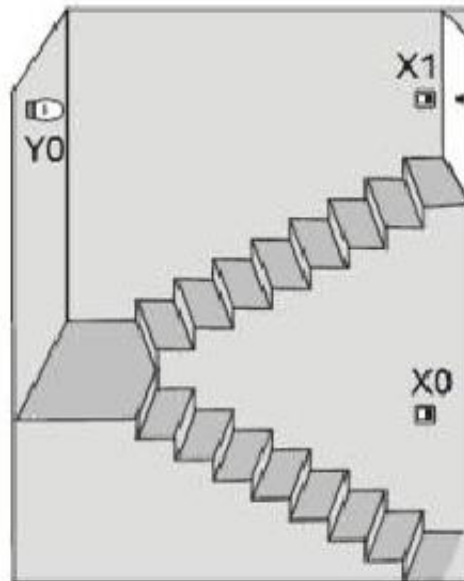
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

اگر بطری بر روی کانوایر ایستاده باشد، سیگنال ورودی هر دو سنسورهای فتوالکتریکی برای آشکارسازی سر بطری و ته بطری فعال خواهد شد، که در این حالت هر دو ورودی $X0=ON$ و $X1=ON$ می‌شوند. بنابراین، کنتاکت در حالت عادی باز (NO) ورودی $X0$ و کنتاکت در حالت عادی بسته (NC) ورودی $X1$ در برنامه فعال شده و تغییر وضعیت می‌دهند. در این حالت، خروجی $Y0$ که به شیر برقی متصل شده است، غیرفعال OFF باقی مانده و بطری به خارج کانوایر شوت نمی‌شود.

اگر بطری بر روی کانوایر افتاده باشد، فقط سیگنال ورودی سنسور فتوالکتریک متصل شده به ترمینال $X0$ که مربوط به ته بطری می‌باشد، فعال می‌شود، که در این حالت $X0=ON$ و $X1=OFF$ خواهد بود. بنابراین، خروجی $Y0$ فعال (ON) خواهد شد. در این صورت، بطری افتاده توسط شیر پنوماتیک به خارج از کانوایر شوت می‌شود.

10-1 کنترل روشنایی راه‌پله در خانه هوشمند

در این مثال، می‌خواهیم روشنایی یک راه‌پله را کنترل نماییم. به این گونه مدار اتصال بلوک پارالل می‌گویند.

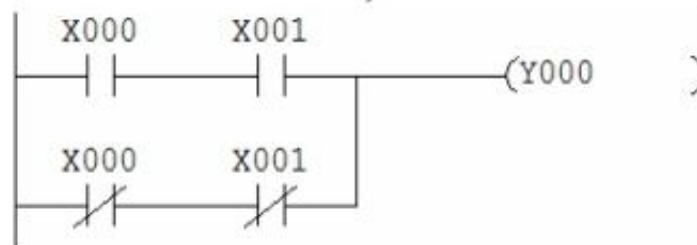


هدف از کنترل در این مثال، کنترل روشنایی راه پله توسط افراد در پایین و بالای راه پله با استفاده از سوئیچ‌های ON/OFF قفل شونده می‌باشد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ترمینال PLC شستی قفل شونده متصل می‌شود. این سوئیچ در پایین راه پله متصل شده است.
X1	به این ترمینال PLC شستی قفل شونده متصل می‌شود. این سوئیچ در بالای راه پله متصل شده است.
Y1	به این خروجی PLC لامپ روشنایی راه پله متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

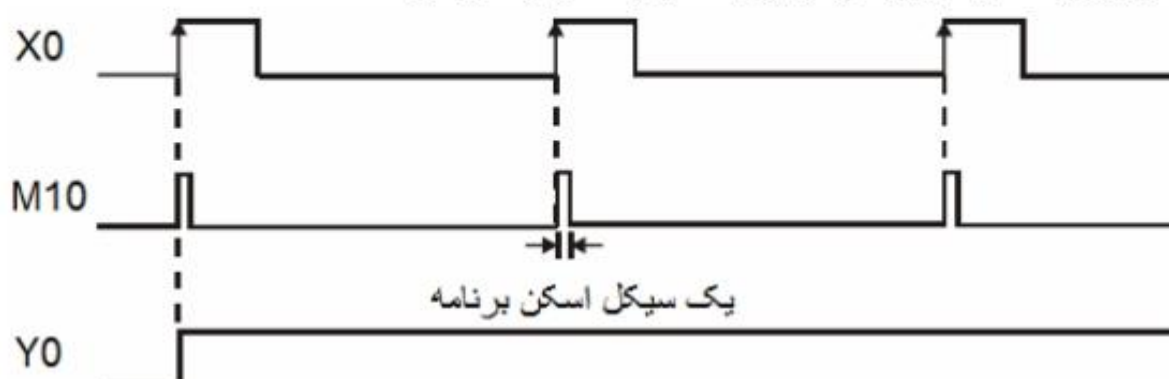
اگر وضعیت هر دو سوئیچ پایین و بالای راه پله یکسان باشد، هر دو ON و یا OFF باشند، لامپ روشن خواهد بود. در غیر این صورت، اگر یک سوئیچ ON و سوئیچ دیگر OFF شود، لامپ خاموش می‌شود.

وقتی که لامپ خاموش باشد، افراد می‌توانند لامپ راه پله را با تغییر وضعیت هر یک از سوئیچ‌های بالا یا پایین راه پله روشن کنند. همچنین، وقتی که لامپ روشن باشد، با تغییر وضعیت هر یک از سوئیچ‌ها می‌توان لامپ را خاموش کرد.

1-11 بررسی لبه‌بالارونده پالس ورودی در یک سیکل اسکن برنامه

در این مثال، می‌خواهیم عملکرد پالس خروجی لبه‌ی بالارونده‌ی سیگنال ورودی را برای یک سیکل اسکن برنامه بررسی کنیم.

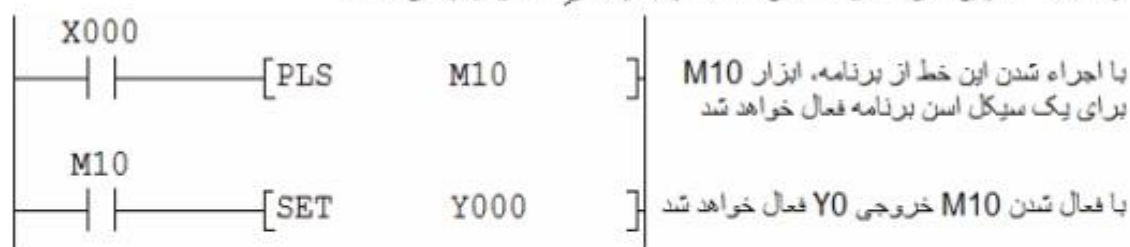
هدف از این کنترل در این مثال، ایجاد کردن یک پالس در یک سیکل اسکن برنامه برای راه‌اندازی (تریگر کردن) یک نمایشگر و یا یک ابزار در هنگام فعال شدن سوئیچ X0 می‌باشد. شکل زیر، یک نمونه از ایجاد پالس در یک سیکل اسکن را نشان می‌دهد.



ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی، یک سوئیچ فشاری (پوش پاتون) دو حالتی وصل می‌شود که از ON به OFF تغییر وضعیت خواهد داد
M10	این ابزار یک پالس راه‌انداز را برای یک سیکل اسکن برنامه ایجاد خواهد کرد
Y0	به این خروجی PLC یک نمایشگر (به عنوان مثال LED) متصل خواهد شد

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



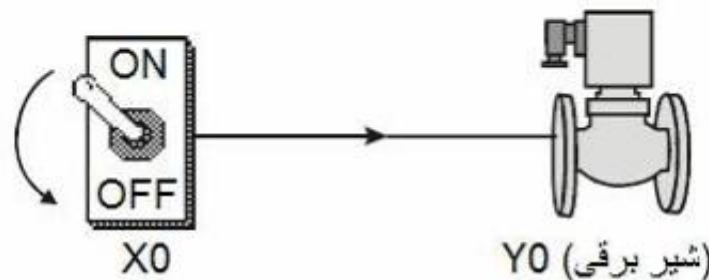
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که ورودی X0 فعال شود (لبه بالارونده راه‌اندازی شود)، دستورالعمل PLS اجرا خواهد شد و M10 یک پالس را برای یک سیکل اسکن برنامه ارسال خواهد کرد.

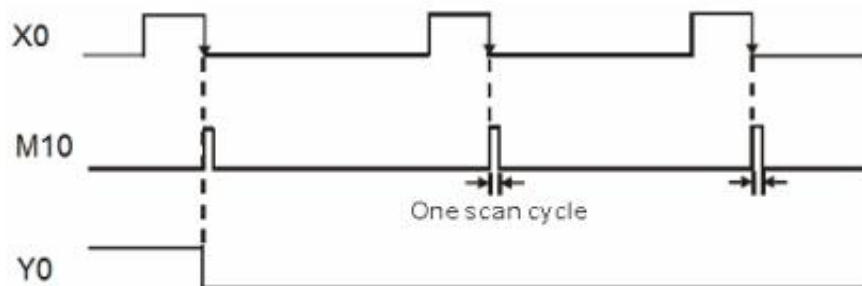
وقتی که M10 = ON شود، دستورالعمل [SET Y0] اجرا شده و Y0 فعال (ON) می‌شود. از این رو، لامپ LED روشن خواهد شد و یا دیگر ابزارهای متصل شده به خروجی Y0 فعال خواهند شد.

12-1 کنترل خروجی با لبه‌ی بالارونده پالس ورودی در یک سیکل اسکن برنامه

در این مثال، می‌خواهیم عملکرد خروجی پالس با لبه‌ی پایین‌رونده را برای یک سیکل اسکن بررسی کنیم.



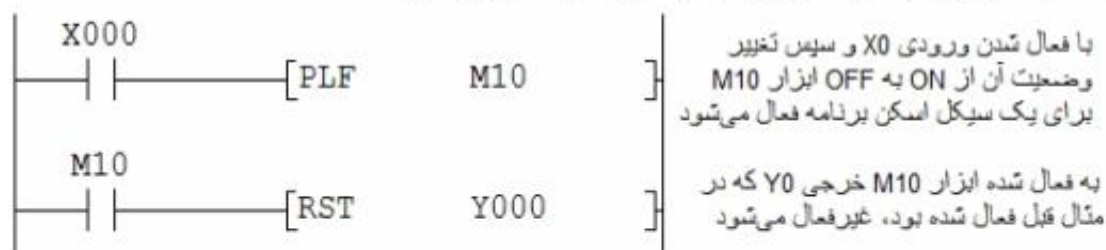
هدف از کنترل در این مثال، ایجاد یک پالس برای یک سیکل اسکن برنامه برای راه‌اندازی یک شیر الکترومغناطیسی و یا دیگر تجهیزات قابل اتصال به خروجی دیجیتال PLC در هنگامی است که سوئیچ یا پوش پاتون از ON به OFF تغییر وضعیت می‌دهد. دیاگرام زمان‌بندی عملکرد دستورالعمل لبه‌ی پایین‌رونده برای این مثال در شکل زیر نشان داده شده است:



ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، یک سوئیچ فشاری دوحالتی وصل می‌شود و تغییر وضعیت برای برنامه از ON به OFF قابل قبول است.
M10	از این ابزار برای ایجاد کردن یک پالس راه‌انداز برای یک سیکل اسکن برنامه استفاده خواهد شد.
Y0	به این خروجی PLC، یک شیر الکترومغناطیسی متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



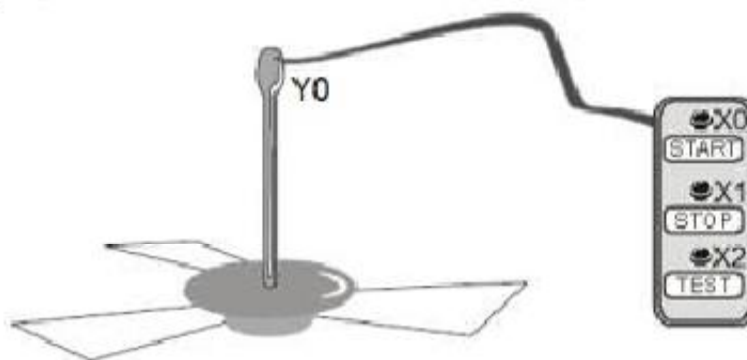
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که ورودی $X0$ فعال شده و سپس غیرفعال می‌شود و در واقع، از ON به OFF تغییر وضعیت می‌دهد (لبه‌ی پایین‌رونده راه‌اندازی می‌شود)، دستورالعمل PLF اجرا شده و ابزار $M10$ یک پالس را برای یک سیکل اسکن ارسال خواهد کرد.

وقتی که $M10 = ON$ شود، دستورالعمل $[RST Y0]$ اجرا شده و $Y0$ غیرفعال OFF می‌شود. از این رو، شیر الکترومغناطیسی بسته خواهد شد.

1- 13- مدار کنترل خودنگهدار

در این مثال، قصد داریم مدار کنترل خودنگهدار یا قفل کننده (Latching) را بررسی نماییم.

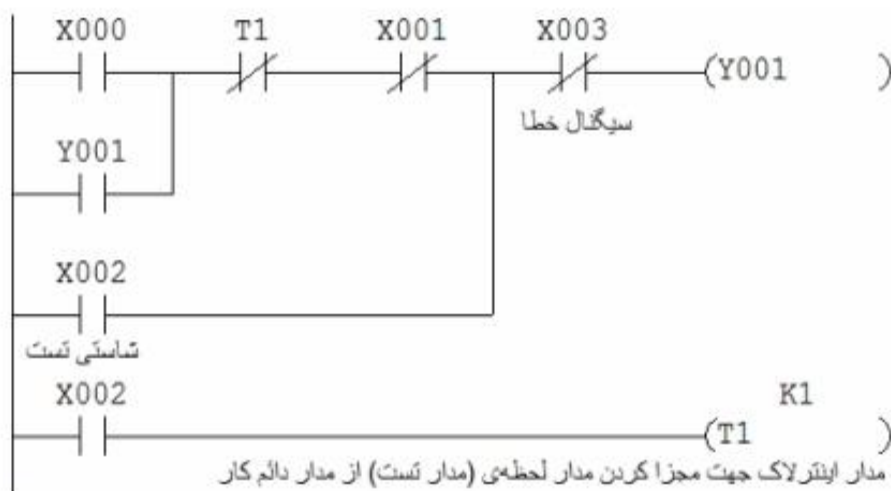


هدف از کنترل در این مثال، کنترل وضعیت Run بودن پنکه‌ی سقفی توسط فشار دادن شستی $START$ و $STOP$ می‌باشد. همچنین، می‌خواهیم با استفاده از فشردن دکمه $TEST$ وضعیت Run بودن نرمال فن سقفی را چک کنیم.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
$X0$	شستی $START$ به این ورودی از PLC متصل می‌شود و کنتاکت این شستی در حالت عادی باز (NO) است.
$X1$	شستی $STOP$ به این ورودی از PLC متصل می‌شود و کنتاکت این شستی در حالت عادی باز (NO) است.
$X2$	شستی $TEST$ به این ورودی از PLC متصل می‌شود و کنتاکت این شستی در حالت عادی باز (NO) می‌باشد.
$X3$	سیگنال خطا به این ورودی از PLC متصل شده است. سیگنال خطا می‌تواند مربوط به بی‌متال یا کلید حرارتی و ... باشد.
$Y0$	به این خروجی از PLC ، سیگنال کنترل موتور فن سقفی متصل شده است.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

با فشردن شستی START، ورودی $X0 = ON$ می‌شود. در این حالت، فن سقفی روشن شده ($Y0 = ON$) و تا زمانی که خطایی رخ دهد ($X3 = ON$) و یا شستی STOP (ورودی $X1 = ON$) فعال شود، فن روشن باقی خواهد ماند. در این غیر اینصورت، با فشردن شستی استوپ و یا بروز خطا، فن خاموش می‌شود ($Y0 = OFF$). بعد از روشن شدن فن، خروجی $Y1$ توسط مدار خودنگهدار (Latching) فعال باقی خواهد ماند. در این حالت، حتی اگر شستی START رها شود، فن به کار خود ادامه می‌دهد.

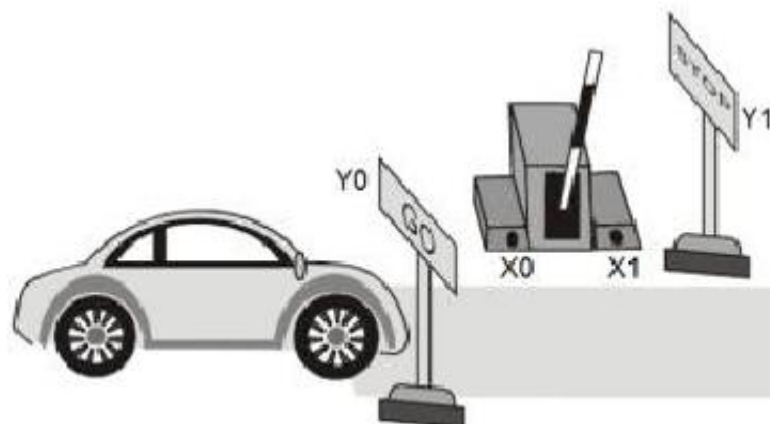
وقتی که شستی STOP فشرده شود، ورودی $X1 = ON$ شده و خروجی $Y1 = OFF$ می‌شود. در این حالت، فن از کار خواهد افتاد (متوقف می‌شود).

اگر خطایی رخ دهد (بی‌مثال عمل کند، کلید حرارتی عمل کند) ورودی $X3 = ON$ شده و $Y1 = OFF$ می‌شود که در این حالت نیز فن خاموش می‌شود.

وقتی که دکمه TEST فشرده شود ($X2 = ON$)، خروجی $Y1 = ON$ می‌شود. این در صورتی ممکن است که خطایی وجود نداشته باشد. با رها کردن این شستی، فن که روشن شده بود، خاموش می‌شود.

1- 14- مدار کنترل اینترلاک جهت کنترل ورود و خروج اتومبیل در پارکینگ

در این مثال، می‌خواهیم مدار کنترل اینترلاک (Interlock) را بررسی نماییم.

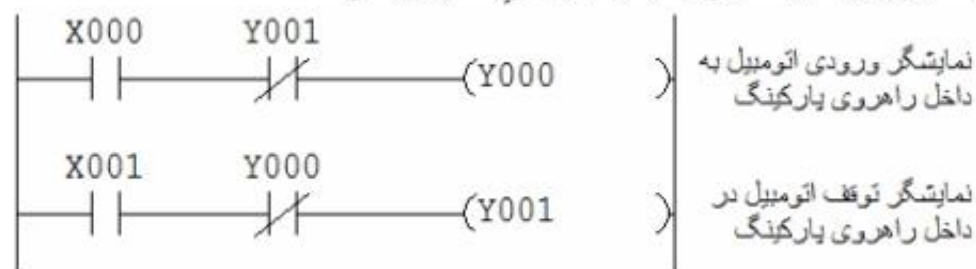


هدف از کنترل در این مثال، کنترل ورود و خروج ماشین به داخل پارکینگ از جلوی نگهبانی و از زیر راهبند می‌باشد. توسط این نوع کنترل می‌توان از بروز حادثه در هنگام ورود و خروج ماشین به پارکینگ جلوگیری کرد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، سنسور ورود ماشین به پارکینگ متصل می‌شود. وقتی که یک ماشین جلوی سنسور باشد، ورودی X0 = ON خواهد شد
X1	به این ورودی از PLC، سنسور بودن ماشین در زیر راهبند متصل می‌شود. وقتی که یک ماشین جلوی سنسور باشد، ورودی X1 = ON خواهد شد
Y0	به این خروجی PLC، نمایشگر (لامپ سیگنال) ورودی ماشین به پارکینگ متصل می‌شود. (ON یعنی "GO" و OFF یعنی "STOP")
Y1	به این خروجی PLC، نمایشگر (لامپ سیگنال) بودن ماشین در زیر راهبند متصل می‌شود. (ON یعنی "GO" و OFF یعنی "STOP")

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

در بسیاری از پارکینگ‌ها دو نمایشگر ورود ماشین به پارکینگ و بودن ماشین جلوی راهبند پارکینگ وجود دارد. توسط مدار کنترل اینتراک، یک نمایشگر سیگنال "GO" برای حرکت ماشین به داخل

راهروی پارکینگ نشان داده خواهد شد. با این کار، از بروز حادثه در هنگام ورود ماشین به پارکینگ و خروج اتومبیل از پارکینگ جلوگیری می‌شود.

وقتی که یک ماشین برای وارد شدن به پارکینگ نزدیک به راه‌بند می‌شود، ورودی $X0$ فعال (ON) خواهد شد، که در این صورت خروجی $Y0$ فعال می‌شود. حال، نمایشگر (لامپ سیگنال) ورود ماشین به داخل پارکینگ "GO" روشن خواهد شد. در همین زمان، نمایشگر بودن ماشین در زیر راه‌بند "STOP" نیز روشن می‌شود و تا زمانی که ماشین کاملاً وارد پارکینگ نشده است، سنسور $X1$ فعال بوده و اجازه‌ی ورود ماشین بعدی صادر نمی‌شود. از نمایشگر "STOP" برای جلوگیری از حادثه‌ی تصادف استفاده می‌شود.

1- 15- مدار قفل‌شونده (خودنگهدار)

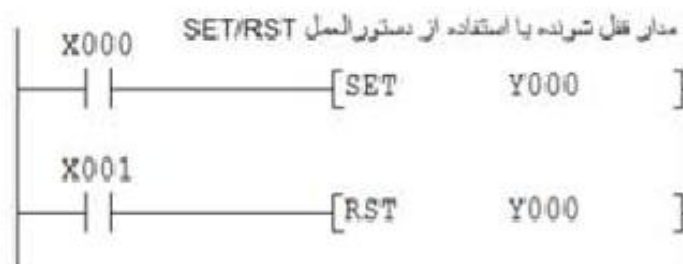
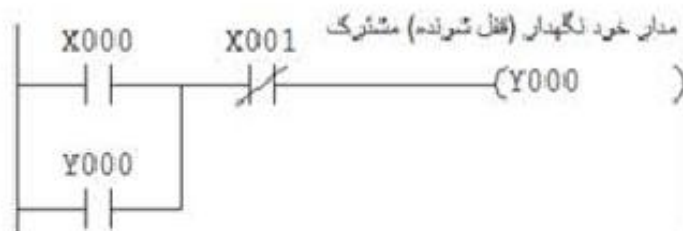
در این مثال، می‌خواهیم مدار قفل‌کننده (خودنگهدار) و معادل آن، یعنی دستورالعمل‌های SET/RST را با یکدیگر توضیح دهیم.

هدف از کنترل در این مثال، روشن و خاموش کردن یک نمایشگر LED با تغییر وضعیت سوئیچ مدار کنترلی می‌باشد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
$X0$	به این ورودی PLC، پوش پاتون با کنتاکت NO متصل می‌شود. با فشردن شستی متصل شده به ورودی $X0$ ، این ابزار در برنامه ON می‌شود.
$X1$	به این ورودی از PLC، پوش پاتون با کنتاکت NC متصل می‌شود. با فشردن شستی متصل شده به ورودی $X1$ ، این ابزار در برنامه ON می‌شود.
$Y0$	به این خروجی PLC، نمایشگر LED متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیگرام نردبانی شکل زیر می‌باشد:



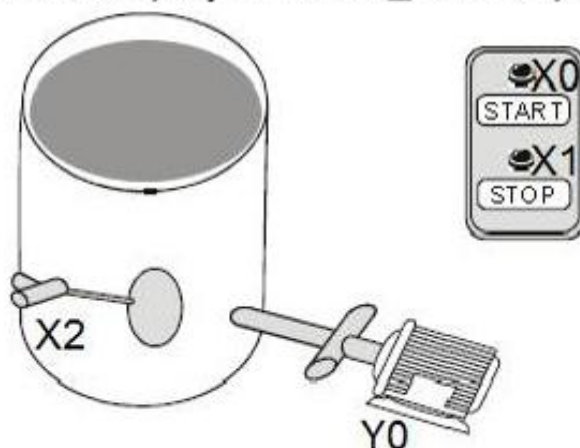
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

با فعال شدن ورودی X0، خروجی Y0 فعال خواهد شد. وقتی که ورودی X0 از ON به OFF تغییر وضعیت می‌دهد، خروجی Y0 همچنان در وضعیت فعال (ON) باقی خواهد ماند. وقتی که ورودی X1 از OFF به ON تغییر وضعیت می‌دهد، خروجی Y1 به وضعیت OFF خواهد رفت.

وقتی که X0 و X1 هر دو با هم در یک زمان فعال (ON) شوند، ابتدا مدار STOP خواهد شد، سپس خروجی Y0 غیرفعال شده و لامپ خاموش می‌شود. بنابراین، در این برنامه‌ی نوشته شده اولویت عملکرد با ریست می‌باشد.

1- 16 مدار SET/RST_Latched و مدار Unlatched

در این مثال، می‌خواهیم مدار SET/RST_Latched و مدار Unlatched را توضیح دهیم.

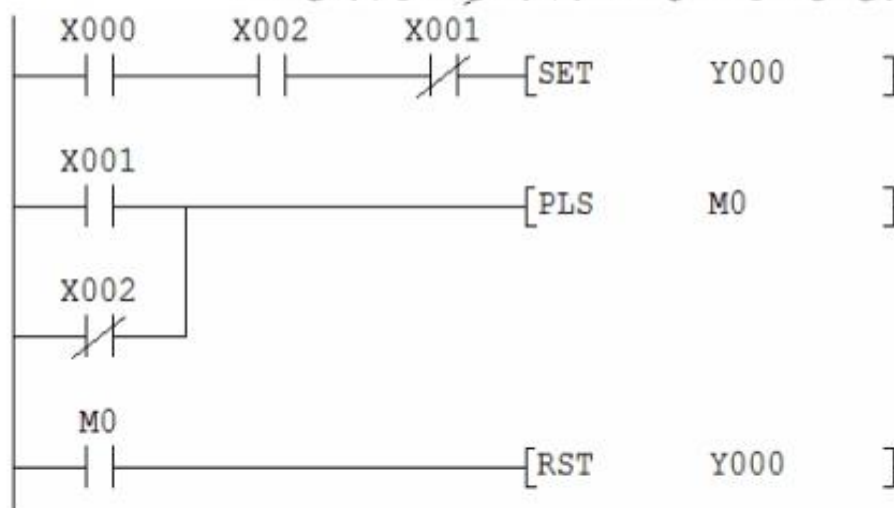


هدف از کنترل در این مثال، راه‌اندازی پمپ آب برای تخلیه کردن مخزن و توقف پمپ با شستی STOP و یا با سنسور خالی بودن آب می‌باشد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC، شستی START متصل می‌شود. وقتی که این شستی فشرده شود $X0 = ON$ خواهد شد.
X1	به این ورودی از PLC، شستی STOP متصل می‌شود. وقتی که این شستی فشرده شود $X0 = ON$ خواهد شد.
X2	به این ورودی از PLC، سنسور آشکارکننده‌ی سطح (لول سنج) متصل می‌شود. اگر آب در تانکر باشد، $X2 = ON$ خواهد بود.
M0	از این ابزار برای راه‌اندازی پالس برای یک سیکل اسکن استفاده می‌شود.
Y0	به این خروجی PLC، موتور پمپ متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

اگر آب در تانکر باشد، $X2 = ON$ خواهد بود. وقتی که شستی START فشرده شود، $X0 = ON$ شده و دستورالعمل SET اجرا می‌شود. خروجی $Y0$ فعال شده و موتور پمپ شروع به پمپاژ کردن آب می‌کند.

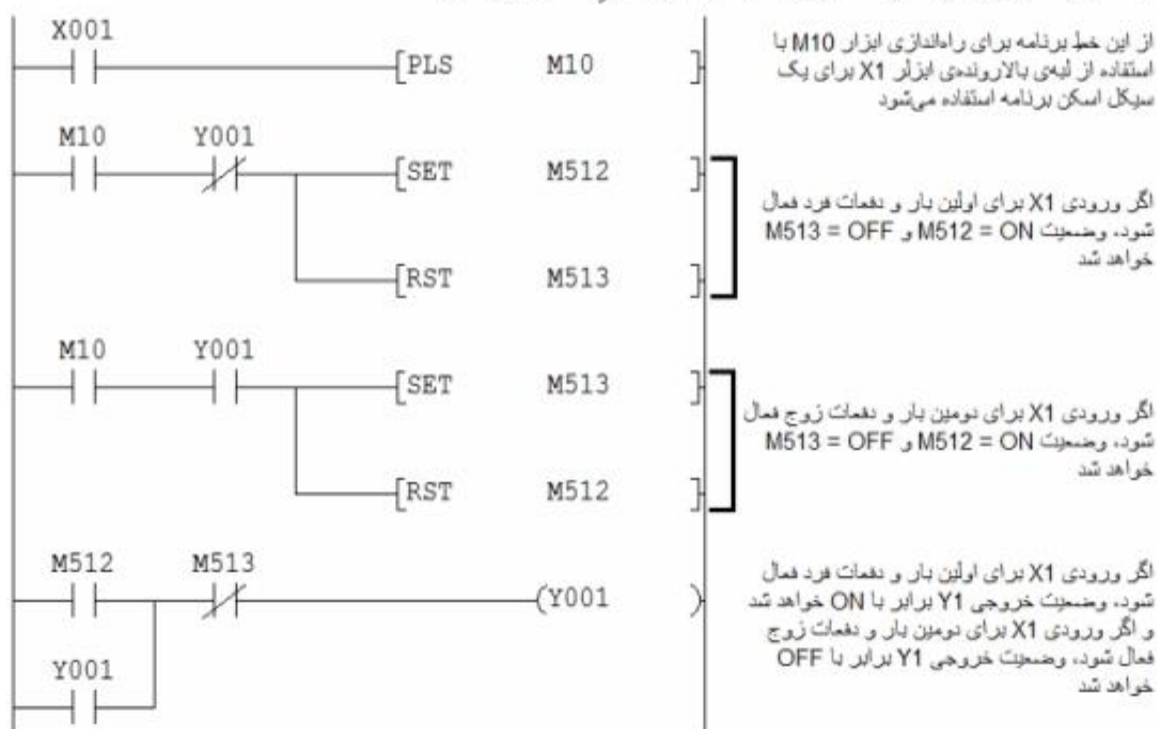
دو حالت برای متوقف کردن موتور پمپ وجود دارد. روش اول: وقتی که دکمه STOP فشرده شود، $X1 = ON$ خواهد شد که در این حالت دستورالعمل PLS اجرا شده و $M0$ برای یک سیکل اسکن برنامه فعال (ON) می‌شود. دستورالعمل RST نیز اجرا خواهد شد و خروجی $Y0$ ریست می‌شود که باعث توقف موتور خواهد شد. روش دوم: وقتی که آب داخل مخزن تخلیه شود، سنسور متصل شده به $X2$ غیرفعال (OFF) می‌شود و دستورالعمل PLS جهت راه‌اندازی $M0$ برای ریست کردن $Y0$ اجرا می‌شود. از این رو، موتور پمپ متوقف خواهد شد.

1- 17- مدار خروجی متناوب با تابع Latched

در این مثال، می‌خواهیم مدار خروجی متناوب با تابع Latched را بررسی نماییم. هدف از کنترل در این مثال، روشن کردن و خاموش کردن لامپ با یک شستی می‌باشد. به طوری که با یکبار فشردن شستی لامپ روشن شده و با فشردن مجدد همان شستی لامپ خاموش شود و وقتی که برق PLC قطع شده و مجدداً وصل شود، وضعیت نمایشگر LED که به خروجی Y1 متصل شده است، به وضعیت قبل از قطع برق برگردد. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X1	به این ورودی PLC، شستی فشاری برگشت با فنر متصل می‌شود. وضعیت کنتاکت این شستی در حالت عادی باز (NO) می‌باشد
M10	از این ابزار برای راه‌اندازی یک پالس برای یک سیکل اسکن استفاده می‌شود.
M512	اگر برای اولین بار و یک در میان ورودی X1 فعال (ON) شود، ابزارهای M512 = ON, M513 = OFF خواهند شد.
M513	اگر برای دومین بار و از آن به بعد یک در میان ورودی X1 فعال شود، ابزارهای M512 = OFF, M513 = ON خواهند شد.
Y1	به این خروجی از PLC، نمایشگر LED متصل شده است.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



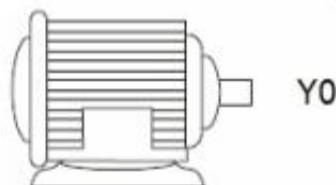
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

با فشردن شستی متصل شده به ورودی $X1$ برای اولین بار و برای دفعات فرد، ورودی $X1$ در برنامه فعال (ON) شده و دستورالعمل $[PLS M10]$ اجرا خواهد شد. در این حالت، ابزار $M10$ برای یک سیکل اسکن برنامه راه‌اندازی (ON) می‌شود. از این رو، وضعیت ابزارهای $M10 = ON$ و $Y1 = OFF$ می‌باشند و دستورالعمل SET و RST در دو خط بعدی برنامه اجرا خواهد شد. سپس خروجی $Y1 = ON$ می‌شود و کنتاکت بسته‌ی این خروجی برای دو خط 2 و 3 باز شده و کنتاکت باز خروجی $Y1$ برای دو خط 4 و 5 بسته می‌شود. در نتیجه، در خط 2، ابزار $M512$ فعال و در خط 3 ابزار $M513$ غیرفعال می‌شود. با رها کردن شستی به دلیل وجود کنتاکت خودنگهدار برای خروجی $Y1$ ، این خروجی فعال باقی خواهد ماند.

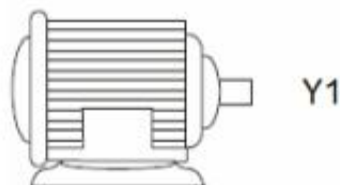
با فشردن مجدد شستی متصل شده به ورودی $X1$ برای دومین بار و برای دفعات زوج، ورودی $X1$ در برنامه فعال (ON) شده و ابزار $M10$ برای یک سیکل اسکن برنامه فعال می‌شود. همانطور که متوجه شدید، برای اولین بار که $X1$ فعال شد، خروجی $Y1$ نیز فعال شد. با فشردن شستی $X1$ برای دومین بار به دلیل فعال بودن خروجی $Y1$ در خط 4 و 5 برنامه، دستورالعمل‌های SET/RST اجرا خواهد شد و به دلیل باز شدن کنتاکت $Y1$ در خط 2 برنامه، خط 2 و 3 اجرا نخواهد شد. از این رو، وضعیت ابزارهای $M512 = OFF$, $M513 = ON$ شده و لامپ LED متصل شده به خروجی $Y1$ خاموش می‌شود. با فشردن مجدد $X1$ لامپ روشن شده و روند توضیح داده شده مجدداً اجرا می‌شود.

1- 18 مدار کنترل شرطی یکی پس از دیگری

در این مثال، می‌خواهیم مدار کنترل شرطی را بررسی نماییم:



موتور پمپ روغن



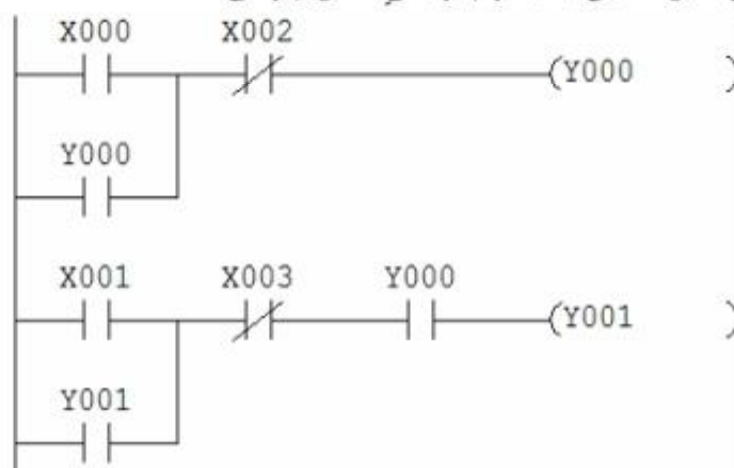
موتور اصلی



هدف از کنترل در این مثال، راه‌اندازی موتور پمپ دستگاه تراش قبل از راه‌اندازی موتور اسپیندل (موتور باریک و دراز) می‌باشد. به این مدار، "مدار کنترل یکی بعد از دیگری" نیز می‌گویند. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC، شستی Oil pump START متصل می‌شود. کنتاکت این شستی در حالت عادی باز (NO) می‌باشد.
X1	به این ورودی PLC، شستی Main motor START متصل می‌شود. کنتاکت این شستی در حالت عادی باز (NO) می‌باشد.
X2	به این ورودی PLC، شستی Oil pump STOP متصل می‌شود. کنتاکت این شستی در حالت عادی باز (NO) می‌باشد.
X3	به این ورودی PLC، شستی Main motor STOP متصل می‌شود. کنتاکت این شستی در حالت عادی باز (NO) می‌باشد.
Y0	به این خروجی از PLC، کنتاکت راه‌اندازی موتور پمپ روغن (Oil pump motor) متصل می‌شود.
Y1	به این خروجی PLC، کنتاکت راه‌اندازی موتور اصلی (Main motor) متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

این برنامه یک نمونه‌ی کاربردی برای مدار کنترل شرطی می‌باشد. وقتی که شستی استارت پمپ روغن فشرده شود، خروجی $Y0 = ON$ می‌شود. حال شرایط برای راه‌اندازی موتور اصلی (موتور اسپیندل) آماده شده است.

با برآورده شدن شرط قبلی و با فشردن شستی استارت موتور اصلی، خروجی $Y1$ فعال می‌شود. هنگامی که موتور اصلی در حال کار می‌باشد، موتور پمپ نیز باید کار کند. با خاموش کردن موتور پمپ، موتور اسپیندل نیز خاموش می‌شود و با خاموش کردن موتور اسپیندل (فشاردن شستی استوپ موتور اصلی)، موتور پمپ خاموش نخواهد شد.

1- 19 مدار حق تقدم (اولین و یا بالاترین اولویت)

در این مثال، می‌خواهیم مدار حق تقدم (اولین و یا بالاترین اولویت) را بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

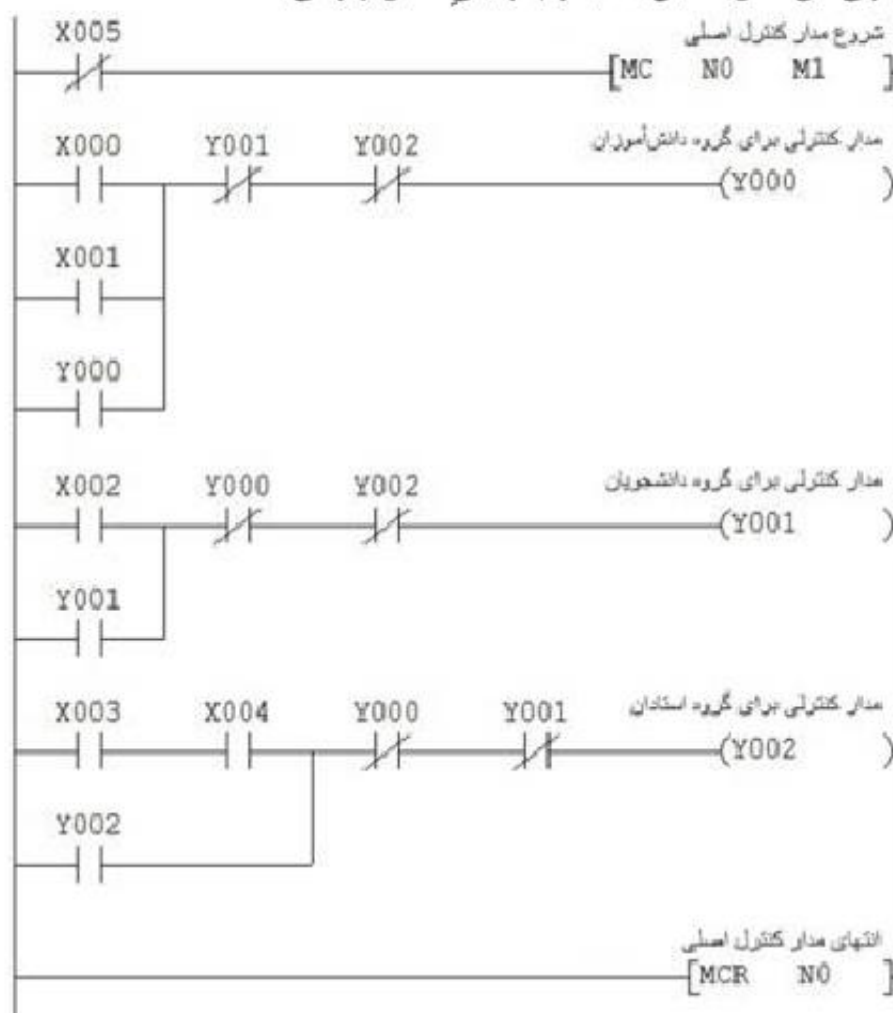
سه گروه شرکت کننده شامل دانش آموزان، دانشجویان و استادان برای یک بازی پرسش و پاسخ وجود دارند. اگر هر گروه بخواهند شانس پاسخ به سوال پرسشگر (مجری) را داشته باشند، باید دکمه‌ی پاسخ که روی میز قرار دارد را فشار دهند. با فشردن دکمه‌ی پاسخ توسط اولین گروه، شستی رومیزی دو گروه دیگر کار نخواهد کرد و آن دو گروه شانس برای پاسخ نخواهند داشت. دو شستی پاسخ برای گروه دانش آموزان و استادان و یک شستی برای گروه دانشجویان بر روی میز آنها وجود دارد که در کل 5 شستی پاسخ بر روی میزها وجود دارد. به منظور اجازه‌ی پاسخ دادن به گروه دانش آموزان، اگر یکی از شستی‌های X0 یا X1 فشرده شوند لامپی که به خروجی Y0 متصل شده روشن می‌شود و برای اجازه‌ی پاسخ دادن به گروه استادان، اگر هر دو شستی X3 و X4 با هم فشرده شوند، لامپی که به خروجی Y2 متصل است روشن می‌شود و برای اجازه‌ی پاسخ دادن به گروه دانشجویان با فشردن دکمه‌ی X2 لامپی که به خروجی Y1 متصل است، روشن می‌شود. اگر مجری دکمه‌ی ریست X5 را فشار دهد، خروجی‌های Y0 و Y1 و Y2 غیرفعال OFF خواهد شد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

عملکرد ابزار	ابزار
اولین شستی پاسخ برای گروه دانش آموزان به این ورودی از PLC متصل می‌شود	X0
دومین شستی پاسخ برای گروه دانش آموزان به این ورودی از PLC متصل می‌شود	X1
شستی پاسخ برای گروه دانشجویان به این ورودی از PLC متصل می‌شود	X2
اولین شستی پاسخ برای گروه استادان به این ورودی از PLC متصل می‌شود	X3

X4	دومین تستی پاسخ برای گروه استادان به این ورودی از PLC متصل می‌شود.
X5	به این ورودی از PLC، تستی ریستی که بر روی میز مجری می‌باشد، متصل می‌شود.
M1	از این ابزار برای قطع و وصل کردن تغذیه مدار کنترلی اصلی با دستور MC N0 M1 جهت ریست کردن مدار داخل دو دستورالعمل MC ~ MCR استفاده شده است.
Y0	به این خروجی از PLC، لامپ نمایشگر بر روی میز گروه دانش آموزان متصل می‌شود.
Y1	به این خروجی از PLC، لامپ نمایشگر بر روی میز گروه دانشجویان متصل می‌شود.
Y2	به این خروجی از PLC، لامپ نمایشگر بر روی میز گروه استادان متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

اگر مجری دکمه ریست X5 را فشار ندهد، دستورالعمل [MC N0 M1] اجرا خواهد شد و برنامه‌ی بین دو دستور MC و MCR به صورت نرمال اجرا می‌شود. دقت داشته باشید که در

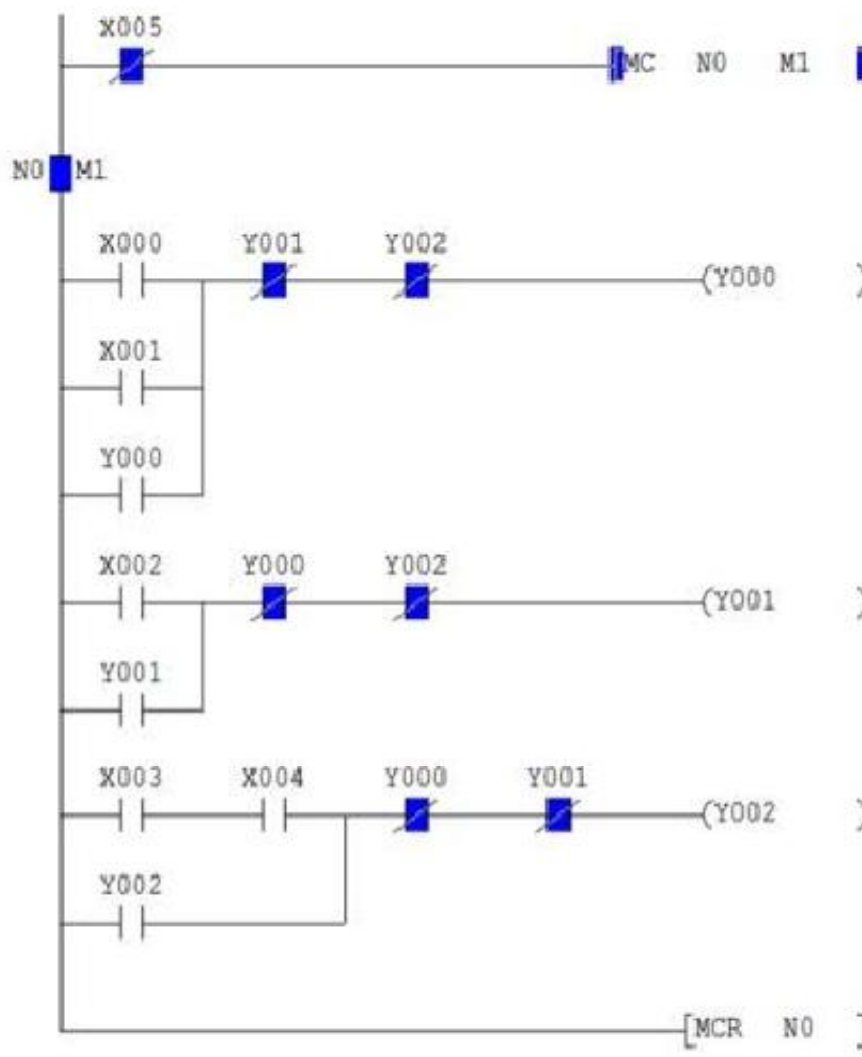
برنامه‌نویسی کنتاکت استفاده شده برای ورودی X5 باید از نوع کنتاکت در حالت عادی بسته (NC) باشد. در این امر اجازه وصل تغذیه به مدار بین دو دستورالعمل $MC \sim MCR$ می‌باشد.

شستی‌های پاسخ برای میز دانش آموزان با یکدیگر موازی هستند و شستی‌های گروه استادان با یکدیگر سری شده‌اند و برای گروه دانشجو فقط یک شستی وجود دارد. اگر شرط برقراری مدار با فشردن شستی‌ها برقرار شود، لامپ نمایشگر گروه مربوطه روشن شده و حتی اگر شستی رها شود لامپ نمایشگر توسط کنتاکت خودنگهدار (مدار Latching) روشن باقی خواهد ماند.

از آنجایی که از مدار اینترلاک استفاده شده است، با فشردن شستی توسط یک گروه، لامپ نمایشگر پاسخ روی میز آن دو گروه دیگر روشن نخواهد شد.

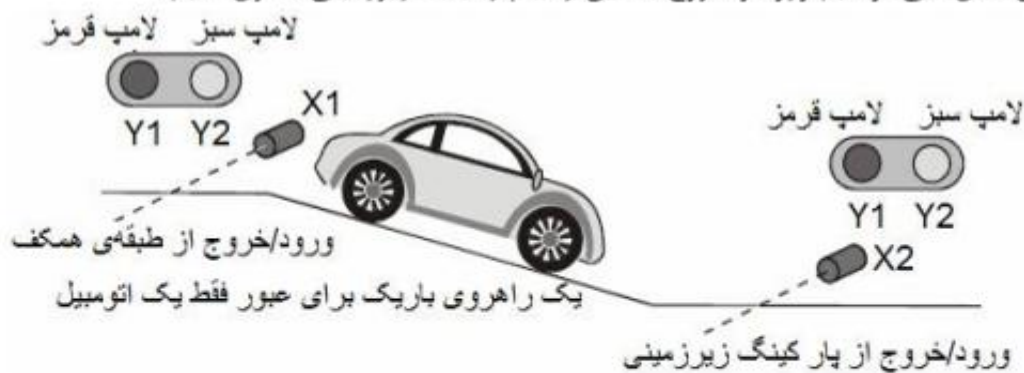
با فشردن دکمه ریست X5 توسط مجری، دستورالعمل [MC N0 M1] اجرا خواهد شد و برنامه‌ای که بین دو دستور MC و MCR قرار دارد، اجرا نخواهد شد. در این حالت تغذیه‌ی خروجی‌های Y0, Y1, Y2 قطع شده و تمام لامپ‌های نمایشگر برای هر سه گروه خاموش می‌شوند. وقتی که مجری دکمه ریست را رها کند، مجدداً تغذیه بین دو دستورالعمل MC و MCR وصل می‌شود و برنامه‌ی بین این دو دستور به صورت نرمال اجرا خواهد شد. در این زمان، مجری می‌تواند سوال بعدی را پرسیده و هر سه گروه آماده‌ی پاسخ می‌باشند.

مدار شکل بالا مربوط به زمان آفلاین بودن نرم‌افزار برنامه‌نویسی با PLC می‌باشد. اما مدار نشان داده شده در شکل پایین مربوط به زمان آنلاین بودن نرم‌افزار برنامه‌نویسی با PLC و همچنین، در مد مانیتور بودن نرم‌افزار می‌باشد. همانطور که مشاهده می‌کنید، تفاوت زمان آنلاین بودن و آفلاین بودن نرم‌افزار با PLC در این برنامه‌ی طراحی شده، وجود و عدم وجود کنتاکت باز N0 M1 می‌باشد. کار این کنتاکت در قسمت تغذیه‌ی اصلی مدار برقراری و یا قطع تغذیه‌ی بین دو دستورالعمل $MC \sim MCR$ می‌باشد.



1-20 کنترل ورود و خروج ماشین به پارکینگ

در این مثال، می‌خواهیم ورود و خروج ماشین را به پارکینگ زیرزمینی کنترل کنیم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

محل ورود و خروج اتومبیل به پارکینگ زیرزمینی از یک راهرو می‌باشد که برای کنترل ترافیک اتومبیل‌ها در این قسمت به یک لامپ نشانگر ترافیک نیاز است. همچنین، به یک چراغ قرمز رنگ برای ممنوع کردن ورود اتومبیل به داخل راهرو و یا توقف اتومبیل داخل راهرو، و یک چراغ سبز برای اجازه‌ی عبور اتومبیل از راهرو نیاز می‌باشد.

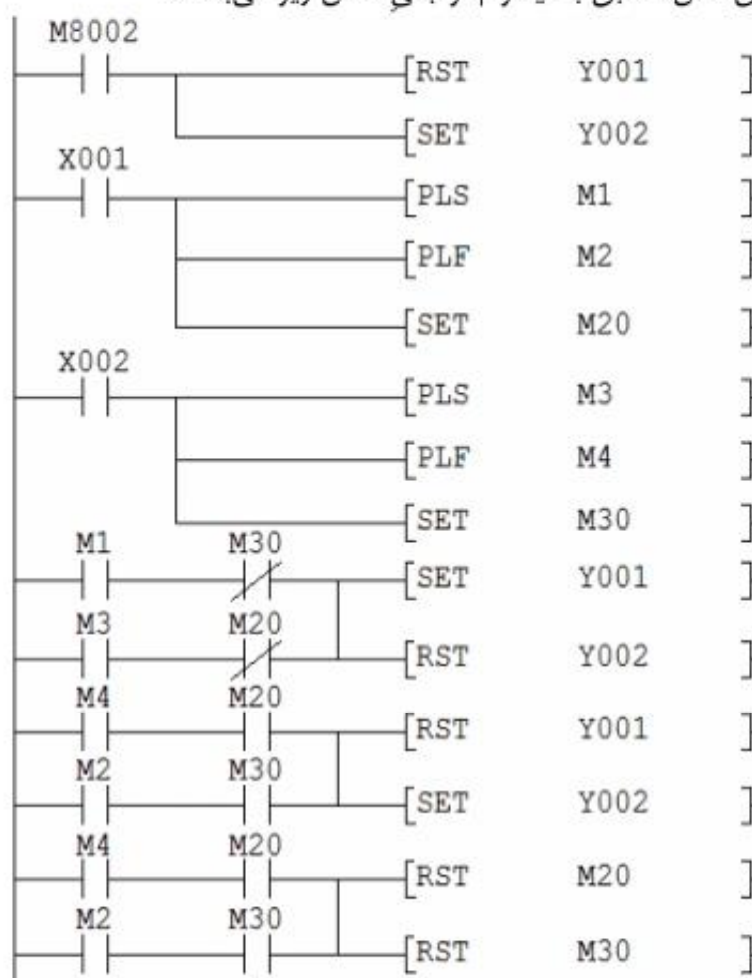
وقتی که اتومبیلی از طبقه همکف وارد راهروی پارکینگ می‌شود، هر دو چراغ قرمز ورودی پارکینگ در طبقه همکف و زیرزمین روشن شده و چراغ‌های سبز خاموش می‌شوند. هر اتومبیلی که وارد راهروی پارکینگ شود و یا داخل راهروی پارکینگ باشد، چراغ قرمز تا هنگامی که اتومبیل به صورت کامل از راهرو خارج شود، روشن می‌ماند. با خارج شدن کامل اتومبیل از راهروی پارکینگ، چراغ سبز روشن شده و اجازه‌ی وارد شدن اتومبیل به راهروی پارکینگ برای ورود به پارکینگ و یا خروج از پارکینگ صادر می‌شود.

وقتی که PLC روشن شود، تنظیمات اولیه چراغ‌های ترافیک به صورت چراغ قرمز خاموش و چراغ سبز روشن خواهد بود.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X1	به این ورودی PLC، سنسور فتوالکتریک در طبقه همکف برای کنترل ورود و خروجی اتومبیل متصل می‌شود. وقتی که یک ماشین از جلوی سنسور عبور کند، ورودی فعال می‌شود.
X2	به این ورودی PLC، سنسور فتوالکتریک در زیرزمین برای کنترل ورود و خروج اتومبیل متصل می‌شود. وقتی که یک ماشین از جلوی سنسور عبور کند، ورودی فعال می‌شود.
M1	وقتی که اتومبیلی از جلوی سنسور X1 از طبقه‌ی همکف به سمت زیرزمین عبور کند، ابزار M1 برای یک سیکل اسکن برنامه فعال (ON) خواهد شد.
M2	وقتی که اتومبیلی از جلوی سنسور X1 از زیرزمین به سمت طبقه‌ی همکف عبور کند، ابزار M2 برای یک سیکل اسکن برنامه فعال (ON) خواهد شد.
M3	وقتی که اتومبیلی از جلوی سنسور X2 از سمت زیرزمین به سمت طبقه‌ی همکف عبور کند، ابزار M3 برای یک سیکل اسکن برنامه فعال (ON) خواهد شد.
M4	وقتی که اتومبیلی از جلوی سنسور X2 از سمت طبقه‌ی همکف به سمت زیر زمین عبور کند، ابزار M4 برای یک سیکل اسکن برنامه فعال (ON) خواهد شد.
M20	وقتی که یک اتومبیل از سمت طبقه‌ی همکف وارد راهروی ورود و خروج پارکینگ شود، ابزار $M20 = ON$ خواهد شد.
M30	وقتی که یک اتومبیل از سمت زیرزمین وارد راهروی ورود و خروج پارکینگ شود، ابزار $M30 = ON$ خواهد شد.
Y1	به این خروجی PLC، چراغ قرمز کنترل ورودی/ خروجی پارکینگ در دو سمت طبقه‌ی همکف و زیرزمین متصل می‌شود.
Y2	به این خروجی PLC، چراغ سبز کنترل ورودی/ خروجی پارکینگ در دو سمت طبقه‌ی همکف و زیرزمین متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

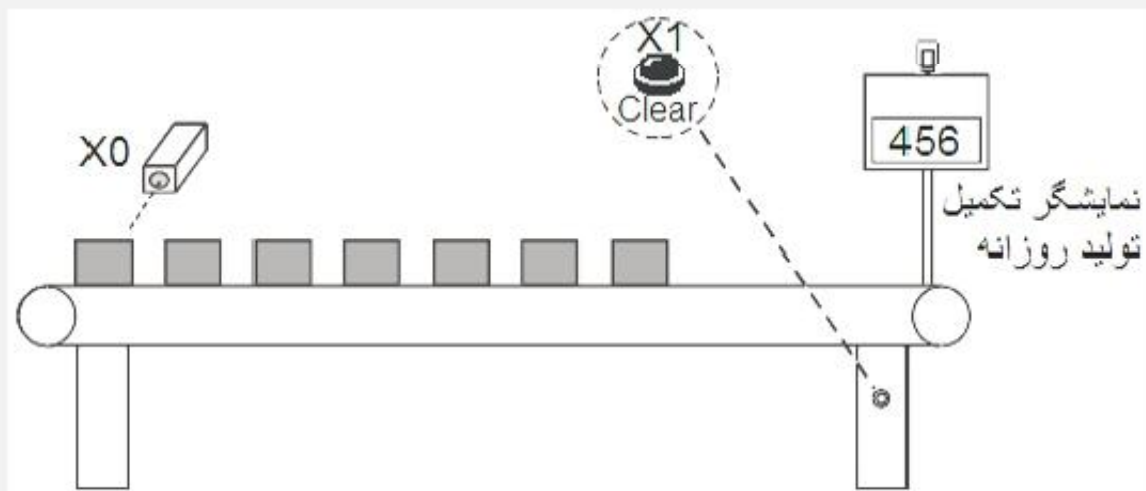
در هر دو سمت راهروی پارکینگ در طبقه‌ی همکف و زیرزمین به طور یکسان دو چراغ سیگنال قرمز (خروجی Y1) و چراغ سیگنال سبز (خروجی Y2) متصل شده است.

در هر دو شرایط ورود به راهروی پارکینگ از سمت طبقه‌ی همکف و توقف اتومبیل در راهروی پارکینگ، ابزار M1 برای فعال کردن خروجی Y1 با استفاده از دستورالعمل [PLS M1] فعال (ON) می‌شود. بنابراین، سیگنال تأیید M20 برای تأیید ورودی اتومبیل به راهروی پارکینگ از سمت طبقه‌ی همکف مورد نیاز می‌باشد.

همچنین وقتی که در هر دو شرایط خروج از پارکینگ از سمت زیرزمین و یا توقف اتومبیل در راهروی پارکینگ، ابزار M3 برای فعال کردن خروجی Y2 با استفاده از اجرای دستورالعمل [PLS M3] فعال (ON) می‌شود. بنابراین، سیگنال تأیید M30 برای تأیید خروج اتومبیل از راهروی پارکینگ از سمت زیرزمین مورد نیاز می‌باشد.

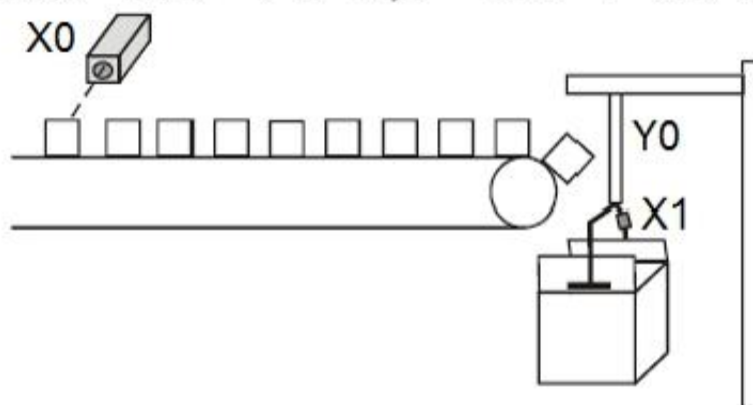
فصل 2

مثال‌های وابسته به تایمرها و کانترها



2- بسته‌بندی محصول در حال حرکت بر روی کانوایر

در این مثال، می‌خواهیم برنامه بسته‌بندی محصول در حال حرکت بر روی کانوایر را بررسی نماییم.

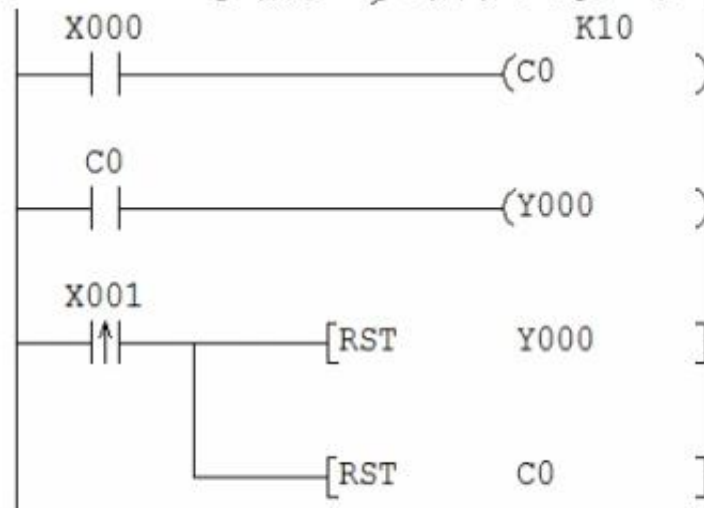


هدف از کنترل در این مثال به شرح زیر می‌باشد:

از یک سنسور فتوالکتریک برای شمارش 10 بسته‌ی در حال حرکت بر روی کانوایر استفاده شده است. بعد از تکمیل شمارش، ربات بازو شروع به بستن کارتون می‌کند. وقتی که عمل بسته‌بندی تکمیل شد، ربات بازو و شمارنده هر دو ریست شده و به حالت اولیه باز می‌گردند. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی، سنسور فتوالکتریک برای شمارش محصول متصل می‌شود. وقتی که محصول توسط سنسور آشکار شود، $X0=ON$ می‌شود.
X1	به این ورودی PLC، سنسور تکمیل عملکرد بازو متصل می‌شود. وقتی که عمل بسته‌بندی تکمیل شد، $X1=ON$ می‌شود.
C0	از این ابزار به عنوان شمارنده‌ی بالارونده‌ی 16 بیتی برای مصارف عمومی استفاده می‌شود.
Y0	به این خروجی، ربات بازو برای بسته‌بندی کردن متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



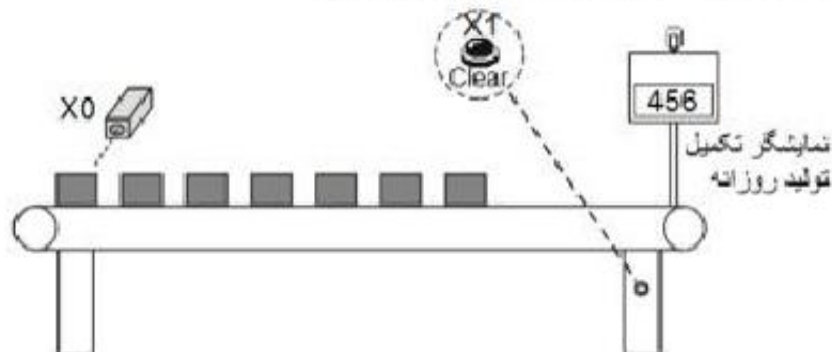
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

با هر بار آشکار شدن محصول توسط سنسور فتوالکتریک، ورودی X0 از OFF به ON تغییر وضعیت داده و شمارنده C0 برای یک بار شمارش خواهد کرد. وقتی که مقدار جاری شمارنده‌ی C0 به 10 رسید، کنتاکت در حالت عادی باز (NO) شمارنده‌ی C0 بسته می‌شود. در این حالت $Y0 = ON$ شده و ربات باز شروع به بستن در کارتون (شروع به بسته‌بندی) می‌کند.

وقتی که عمل بسته‌بندی کارتون تکمیل شد، سنسور تکمیل شدن عملکرد ربات بازو فعال می‌شود. در این حالت، ورودی X1 از OFF به ON تغییر وضعیت داده و دستورالعمل RST اجرا می‌شود که باعث ریست شدن Y0 و C0 برای عمل بسته‌بندی بعدی می‌شود.

2-2 شمارش تعداد محصول تولید شده در روز

در این مثال، می‌خواهیم برنامه شمارش تعداد محصول تولید شده در روز توسط دستورالعمل 16-bit Counting Up Latched Counter را بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

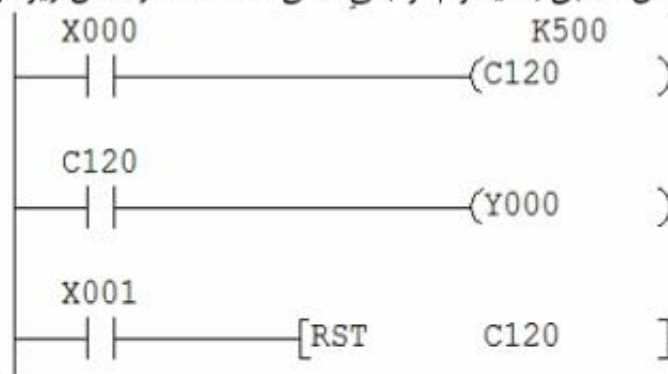
ممکن است خط تولید به خاطر خرابی دستگاه و یا در هنگام استراحت و وقت غذا خاموش شود. در این برنامه قصد داریم در هنگام خاموش بودن دستگاه عمل شمارش تعداد محصول را در حافظه ماندگار PLC نگهداریم و سپس با روشن شدن مجدد دستگاه عمل شمارش ادامه یابد. وقتی که تعداد شمارش محصول به 500 رسید، لامپ نمایشگر تکمیل تولید روزانه روشن شده و روشن باقی می‌ماند.

برای ریست کردن شمارنده باید دکمه‌ی Clear را فشار داده تا مقدار شمارنده ریست شود. در این زمان، دستگاه آماده‌ی شمارش محصول تولید شده در روز بعد می‌باشد. با ریست کردن شمارنده مقدار آن برابر با 0 می‌شود.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC، سنسور فتوالکتریک شمارش بطری متصل می‌شود. با هر بار آشکارسازی محصول توسط سنسور، ورودی X0=ON خواهد شد.
X1	به این ورودی از PLC، شستی Clear برای صفر کردن مقدار شمارنده متصل می‌شود.
C120	این ابزار یک شمارنده 16 بیتی بالارونده‌ی قفل شونده با حافظه‌ی ماندگار (latched) می‌باشد.
Y0	به این خروجی، لامپ نمایشگر تکمیل تولید روزانه متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر می‌باشد:



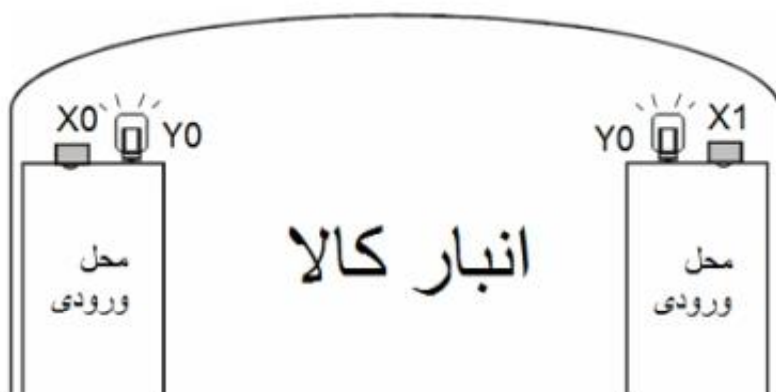
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

از شمارنده‌ی قفل شونده با حافظه‌ی ماندگار برای زمانی که می‌خواهیم مقدار شمرده شده در هنگام قطع و وصل مجدداً تغذیه PLC نگهداری شود استفاده می‌کنیم. وقتی که تولید یک محصول تکمیل شد، مقدار شمارنده‌ی C120 برای یک بار افزایش می‌یابد. وقتی که شمارش به 500 رسید، لامپ نمایشگر تکمیل تولید روزانه روشن خواهد شد.

لازم به ذکر است که برای اطلاع از تعداد و شماری شماری‌های قفل شونده با حافظه‌ی ماندگار باید به دفترچه‌ی راهنمای مدل PLC مورد نظر خود مراجعه نمایید. به دلیل جلوگیری از حجیم شدن کتاب از آوردن اطلاعات لازم برای تعداد و شماری شماری‌های قفل شونده برای سری PLC‌های مختلف تولید شده توسط شرکت میتسوبیشی صرفنظر شده است.

2- 3 محاسبه‌ی مقدار محصول تولید شده‌ی نهایی

در این مثال، می‌خواهیم مقدار تولید را با استفاده از دستورالعمل 32-bit Counting Up/Down Counter محاسبه می‌کنیم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

در این برنامه می‌خواهیم تعداد کالای تولید شده را که به انبار کالا وارد شده و از آن خارج می‌شود را مانیتور کنیم. این کار توسط سنسورهای فتوالکتریک انجام می‌شود. وقتی که مقدار کالای وارد شده و خارج شده از انبار به 40000 رسید، یک آلارم فعال خواهد شد.

نکته :

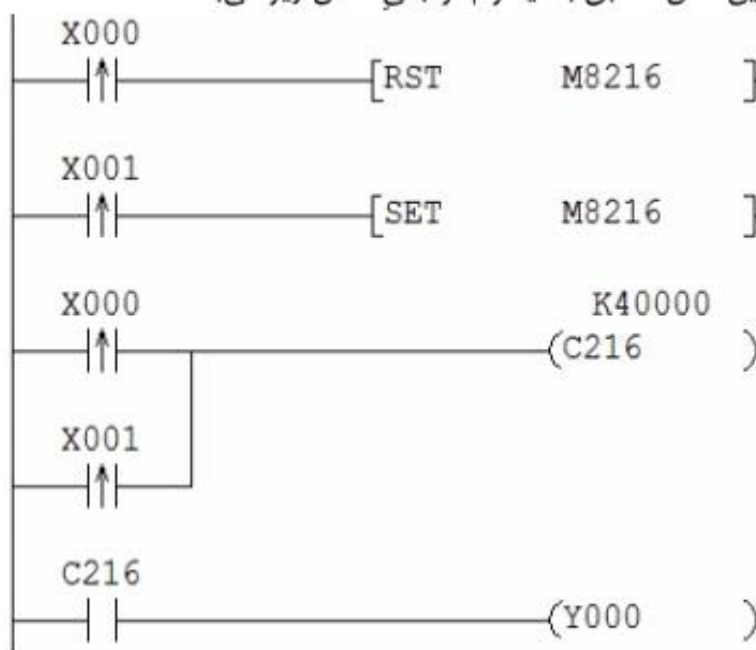
ابزار M8000: وقتی که PLC از وضعیت STOP به Run برود، این فلگ ON می‌شود، و بالعکس وقتی که PLC از وضعیت Run به STOP برود، این ابزار غیرفعال می‌شود. M8216 مربوط به تعیین مد کاری شمارنده‌ی C216 می‌باشد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، سنسور فتوالکتریک برای شمارش و مانیتور کردن کالاهای وارد شده به انبار متصل می‌شود. وقتی که کالایی جلوی سنسور قرار گیرد، ورودی $X0 = ON$ می‌شود.
X1	به این ورودی از PLC، سنسور فتوالکتریک برای شمارش و مانیتور کردن کالاهای خارج شده از انبار متصل می‌شود. وقتی که کالایی جلوی سنسور قرار گیرد، ورودی $X0 = ON$ می‌شود.

M1216	این ابزار مد کاری شمارنده‌ی C216 را تعیین می‌کند. به این صورت که اگر $M8216 = ON$ شود، مد شمارنده پایین‌رونده خواهد شد.
C216	این ابزار یک شمارنده‌ی 32 بیتی پایین‌رونده/ بالا رونده (Up/Down Counter) می‌باشد.
Y0	خروجی از PLC که آلارم به آن متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

نکته‌ی کلیدی این برنامه این است که فلگ M8216 مربوط به تعیین مد کاری شمارنده‌ی 32 بیتی می‌باشد. مد کاری شمارنده‌ی C216 برابر یا Up/Down می‌باشد. وقتی که X0 از OFF به ON تغییر وضعیت دهد، فلگ $M8216 = OFF$ شده و شمارنده‌ی C216 به مد بالا رونده (Count up) تغییر وضعیت داده و وقتی که ورودی X1 از OFF به ON تغییر وضعیت دهد فلگ M8216 = ON شده و مد کاری شمارنده‌ی C216 به مد پایین‌رونده (Count down) تغییر وضعیت خواهد داد.

وقتی که مقدار جاری شمارنده‌ی C216 به 40000 رسید، خروجی شمارنده $C216 = ON$ شده و آلارم متصل شده به خروجی Y0 را فعال خواهد کرد.

4-2 طراحی ساعت با استفاده از شمارنده

در این مثال، می‌خواهیم عملکرد ساعت 24 ساعته را توسط سه شمارنده بررسی نماییم.



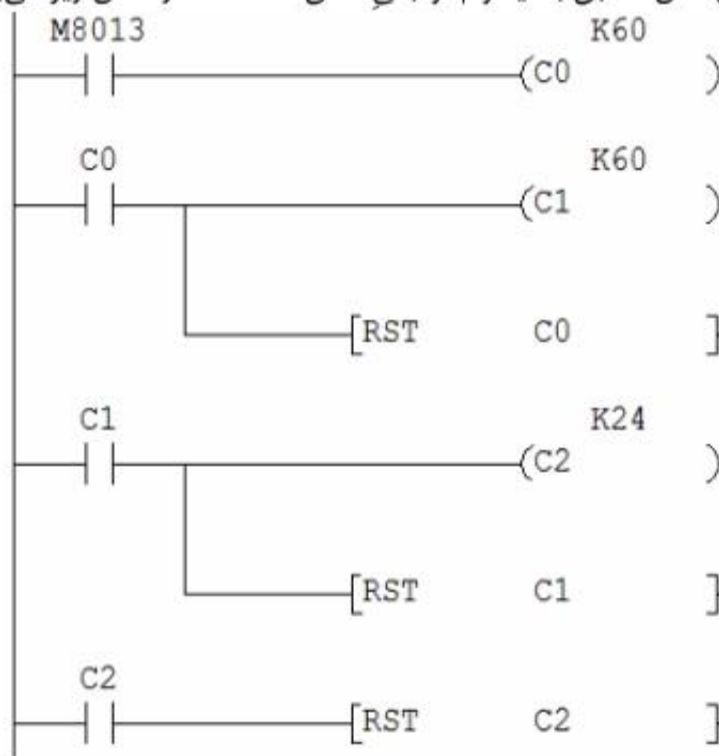
هدف از کنترل در این مثال به شرح زیر می‌باشد:

هر سه شمارنده با یکدیگر از فلگ M8013 (کلاک پالس 1 ثانیه) برای عملکرد ساعت 24 ساعته استفاده می‌کند.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

عملکرد ابزار	ابزار
از این شمارنده برای شمارش ثانیه استفاده می‌شود.	C0
از این شمارنده برای شمارش دقیقه استفاده می‌شود.	C1
از این شمارنده برای شمارش ساعت استفاده می‌شود.	C2
از این ابزار برای ایجاد کلاک پالس 1 ثانیه استفاده می‌شود.	M8013

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر می‌باشد:



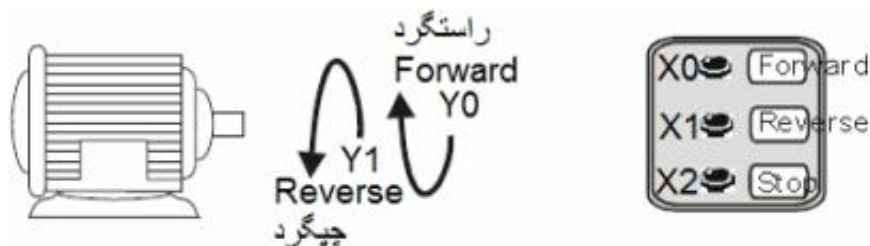
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

نکته قابل توجه برای عملکرد یک ساعت 24 ساعته، استفاده از فلگ M8013 (کلاک پالس 1 ثانیه) می‌باشد. وقتی که برنامه اجرا شود، شمارنده C0 هر یک ثانیه یک بار شمارش خواهد کرد. وقتی که عدد شمرده شده توسط شمارنده‌ی C1 برابر با 60 شد (1 دقیقه)، خروجی شمارنده $C0 = ON$ خواهد شد. سپس شمارنده C1 برای یک بار شمرده شده و شمارنده C0 در همین هنگام ریست می‌شود. همینطور، زمانی که مقدار شمرده شده توسط شمارنده C1 به 60 رسید (1 ساعت)، خروجی $C1 = ON$ خواهد شد. در این حالت، شمارنده‌ی C2 یک بار شمارش خواهد کرد و شمارنده C1 در همان هنگام ریست خواهد شد. بعلاوه، وقتی که مقدار جاری شمارنده C2 به 24 رسید، شمارنده C2 ریست شده و فرآیند شمارش 24 ساعته مجدداً شروع خواهد شد.

در این برنامه از شمارنده C0 برای شمارش ثانیه، شمارنده‌ی C1 برای شمارش دقیقه و از شمارنده‌ی C2 برای شمارش ساعت استفاده می‌شود. در این ساعت، مقدار ثانیه، دقیقه و ساعت توسط شمارنده‌ی C0 و C1 و C2 خوانده می‌شود. وقتی که مقدار تنظیمی شمارنده C2 برابر 12 تنظیم شود، مد ساعت به 12 ساعته تغییر خواهد کرد.

2-5 کنترل جهت چرخش (چپگرد و راستگرد) یک موتور سه فاز آسنکرون

در این مثال، می‌خواهیم کنترل جهت چرخش (چپگرد و راستگرد) یک موتور سه فاز آسنکرون را بررسی نماییم.



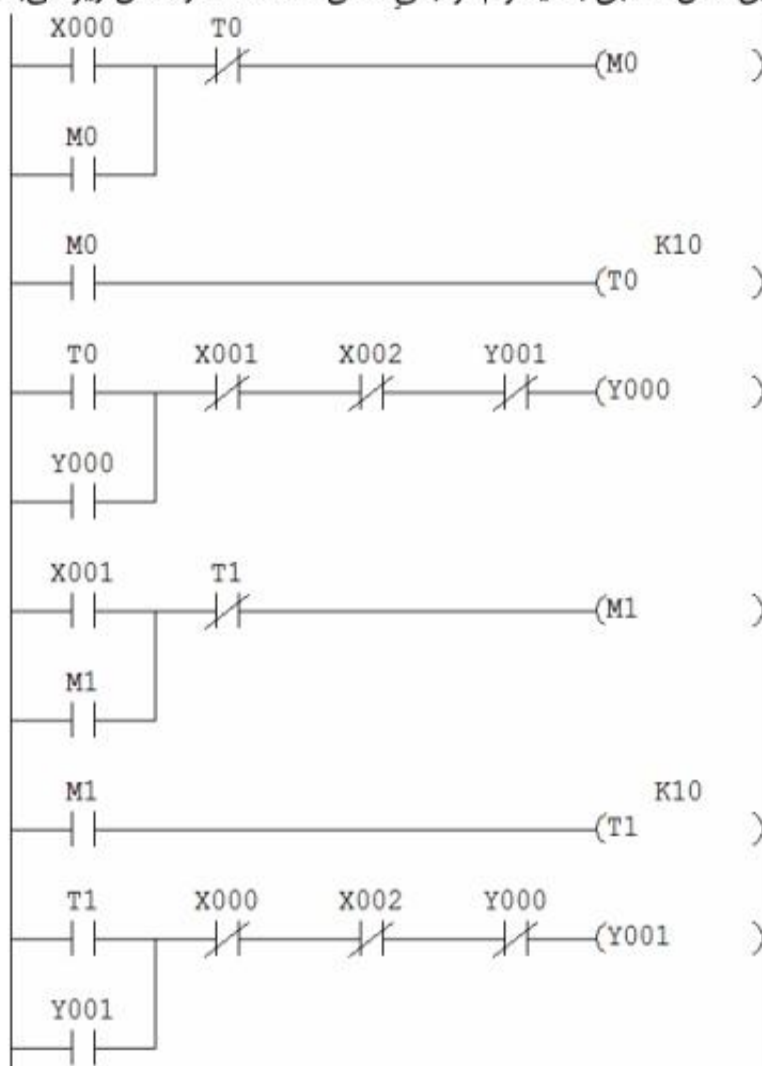
هدف از کنترل در این مثال به شرح زیر می‌باشد:

وقتی که دکمه Forward فشرده شود، موتور در سمت راستگرد شروع به چرخش می‌کند و وقتی که دکمه Reverse فشرده شود، موتور در سمت چپگرد شروع به چرخش نموده و در صورتی که دکمه‌ی Stop فشرده شود، موتور متوقف می‌شود.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند :

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، شستی Forward (راه‌اندازی راستگرد موتور) متصل می‌شود. وقتی که شستی فشرده شود، این ورودی ON می‌شود.
X1	به این ورودی از PLC، شستی Reverse (راه‌اندازی چپگرد موتور) متصل می‌شود. وقتی که شستی فشرده شود، این ورودی ON می‌شود.
X2	به این ورودی از PLC، شستی Stop (توقف موتور) متصل می‌شود. وقتی که شستی فشرده شود، این ورودی ON می‌شود.
T1	از این ابزار برای زمان‌دهی یک قسمت از برنامه استفاده می‌شود. زمان این تایمر 1 ثانیه تنظیم می‌شود.
T2	از این ابزار برای زمان‌دهی یک قسمت از برنامه استفاده می‌شود. زمان این تایمر 1 ثانیه تنظیم می‌شود.
Y0	به این خروجی PLC، کنتاکت Forward (راه‌اندازی الکتروموتور در جهت راستگرد) متصل می‌شود.
Y1	به این خروجی PLC، کنتاکت Reverse (راه‌اندازی الکتروموتور در جهت چپگرد) متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر می‌باشد:



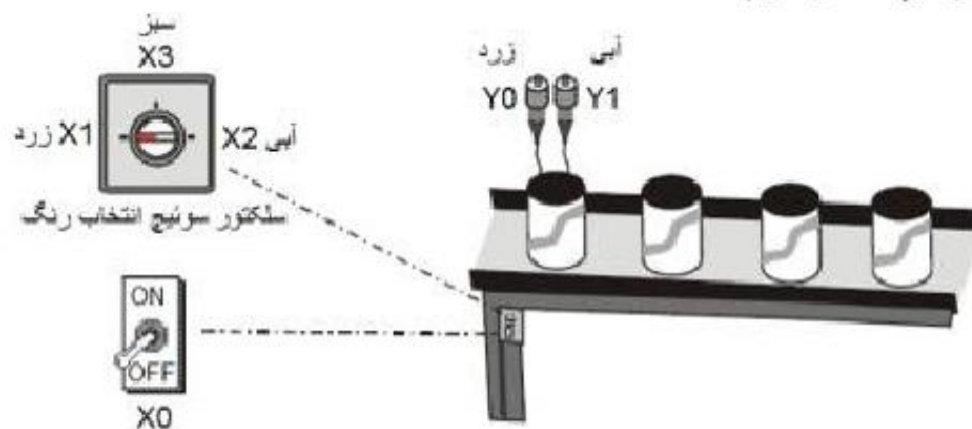
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

با فشردن شستی Forward، ورودی $X0 = ON$ می‌شود. بعد از گذشت یک ثانیه، کنتاکتور $Y0$ فعال شده و موتور در جهت راستگرد شروع به حرکت می‌کند. با فشردن شستی Reverse، ورودی $X1 = ON$ می‌شود. بعد از گذشت یک ثانیه کنتاکتور $Y1$ فعال شده و موتور در جهت چپگرد شروع به حرکت می‌کند. با فشردن دکمه‌ی STOP، ورودی $X2 = ON$ شده و باعث غیرفعال شدن کنتاکتور $Y0$ و $Y1$ شده و موتور متوقف می‌شود.

از دو تایمر در برنامه برای تأخیر انداختن در هنگام تغییر وضعیت موتور به مد Run (استارت) استفاده شده است. علت این وقفه، جلوگیری از اتصال کوتاه در هنگامی فعال بودن یک کنتاکتور می‌باشد. در این مدار برای جلوگیری از اتصال کوتاه از اینترلاک الکتریکی توسط کنتاکت بسته $Y0$ و $Y1$ در مسیر بوبین یکدیگر استفاده شده است.

2-6 انتخاب یک برنامه از میان چند برنامه توسط سوئیچ سلکتوری

در این مثال، قصد داریم برنامه‌ای بنویسیم که توسط یک سوئیچ سلکتوری، از میان چندین برنامه یک برنامه را برای اجرا انتخاب کنیم.



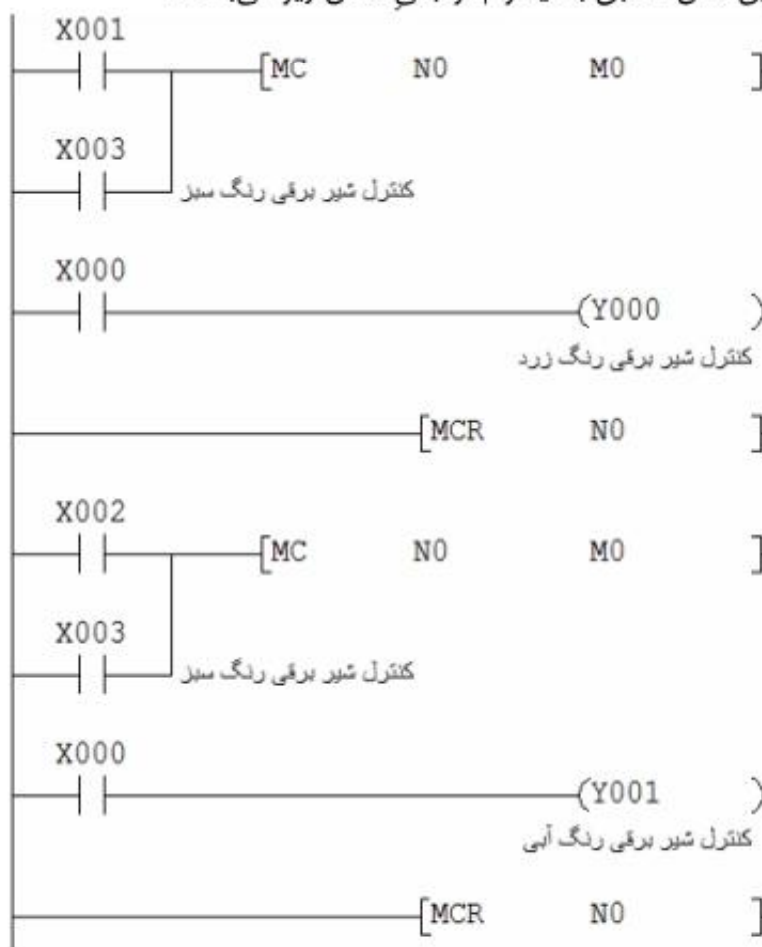
هدف از کنترل در این مثال به شرح زیر می‌باشد:

در این مثال سه نوع ماده‌ی رنگی وجود دارد. با استفاده از سوئیچ‌های مختلف کنترل، اپراتور می‌تواند بطوری را مطابق با ماده‌ی رنگی مورد نظر خود پر کند. در حقیقت، دو رنگ زرد و آبی وجود دارد و با ترکیب این دو رنگ با هم رنگ سبز ساخته می‌شود.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی، سوئیچ شروع پر کردن رنگ متصل می‌شود. وقتی که سوئیچ در وضعیت ON قرار گیرد، ورودی $X0 = ON$ می‌شود.
X1	به این ورودی از PLC، سوئیچ کنترل رنگ زرد متصل می‌شود. وقتی که سوئیچ در وضعیت ON قرار گیرد، ورودی $X1 = ON$ می‌شود.
X2	به این ورودی PLC، سوئیچ کنترل رنگ آبی متصل می‌شود. وقتی که سوئیچ در وضعیت ON قرار گیرد، ورودی $X2 = ON$ می‌شود.
X3	به این ورودی PLC، سوئیچ کنترل رنگ سبز متصل می‌شود. وقتی که سوئیچ در وضعیت ON قرار گیرد، ورودی $X3 = ON$ می‌شود.
M0	از این ابزار برای قطع و وصل کردن تغذیه مدار کنترلی اصلی با دستور $MC N0 M0$ جهت ریست کردن مدار داخل دو دستورالعمل $MC \sim MCR$ استفاده شده است.
Y0	به این خروجی از PLC، شیر برقی پمپاژ رنگ زرد به داخل بطری متصل می‌شود.
Y1	به این خروجی از PLC، شیر برقی پمپاژ رنگ آبی به داخل بطری متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



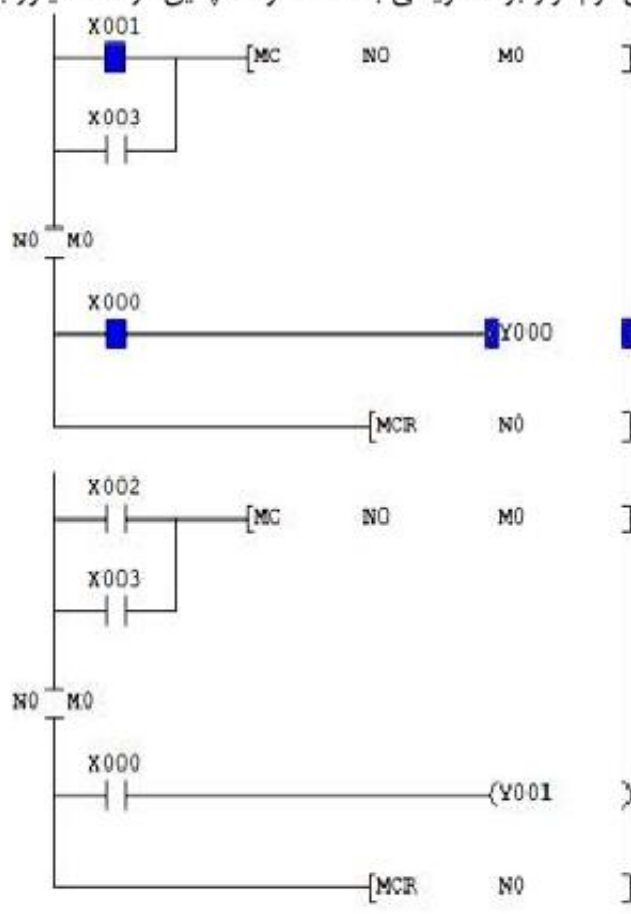
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

در ابتدا، قبل از کنترل پر کردن رنگ داخل بطری، باید سوئیچ اصلی $X0 = ON$ شود. وقتی که وضعیت سلکتور سوئیچ را برابر با $X1 = ON$ قرار دهید، رنگ زرد به داخل بطری ریخته می‌شود. اولین دستورالعمل $MC \sim MCR$ اجرا شده و سپس $Y0 = ON$ شده و سیستم شروع به پر کردن رنگ زرد می‌کند.

وقتی که وضعیت سلکتور سوئیچ را برابر با $X2 = ON$ قرار دهیم، برنامه رنگ آبی فعال شده و دومین دستورالعمل $MC \sim MCR$ اجرا می‌شود. خروجی $Y1 = ON$ شده و سیستم شروع به پر کردن رنگ آبی داخل بطری می‌کند.

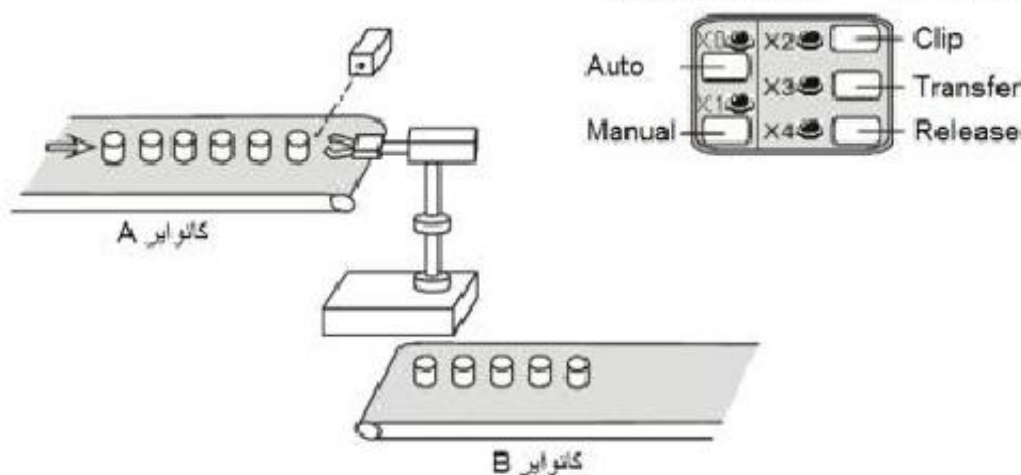
وقتی که وضعیت سلکتور سوئیچ را برابر با $X3 = ON$ قرار دهیم، برنامه رنگ سبز فعال شده و هر دو دستورالعمل $MC \sim MCR$ اجرا می‌شوند. در این حالت، هر دو خروجی $Y = ON$ و $Y1 = ON$ شده و سیستم با ترکیب دو رنگ زرد و آبی شروع به پر کردن رنگ سبز می‌نماید.

مدار شکل بالا مربوط به زمان آفلاین بودن نرم‌افزار برنامه‌نویسی با PLC است. اما مدار شکل پایین مربوط به زمان آنلاین بودن نرم‌افزار برنامه‌نویسی با PLC و همچنین در مد مانیتور بودن نرم‌افزار می‌باشد.



2-7 برنامه انتخاب مد کنترل دستی / اتوماتیک

در این مثال، می‌خواهیم برنامه انتخاب مد کنترل دستی / اتوماتیک (Man/Auto) را با استفاده از دستورالعمل MC ~ MCR بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

وقتی که شستی **Manual** فشرده شود، ربات باز شروع به کار کرده و به صورت دستی کنترل می‌شود. در حالت کنترل دستی با فشردن دکمه **Clip** ربات قوطی را از روی کانوایر **A** برداشته و با فشردن دکمه **Transfer** بطری را به سمت کانوایر **B** منتقل می‌کند و با فشردن دکمه **Release** بطری را بر روی کانوایر **B** قرار داده و آن را رها می‌کند.

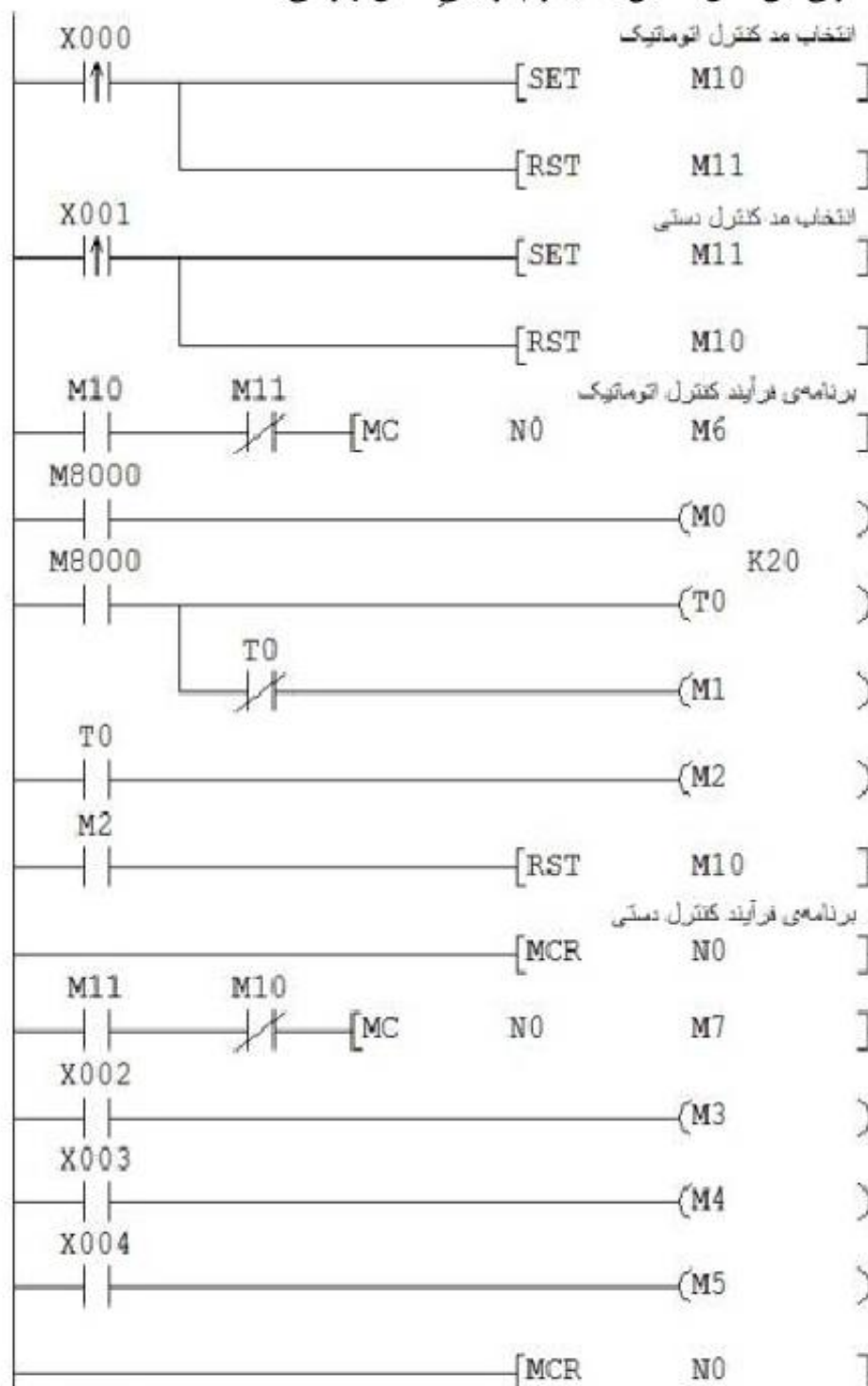
وقتی که شستی **Auto** فشرده شود، ربات باز شروع به کار کرده و برای یک سیکل به صورت اتوماتیک عمل می‌کند. عملکرد اتوماتیک ربات به این صورت است که ابتدا ربات بطری را از روی کانوایر **A** برداشته و آن را نگه می‌دارد و سپس بطری را به سمت کانوایر **B** حرکت می‌دهد. این عمل در مدت زمان 2 ثانیه انجام می‌شود و در نهایت بطری را بر روی کانوایر **B** قرار داده و رها می‌کند. اگر دکمه‌ی **Auto** مجدداً فشرده شود، فرآیند کنترل اتوماتیک ربات دوباره اجرا می‌شود.

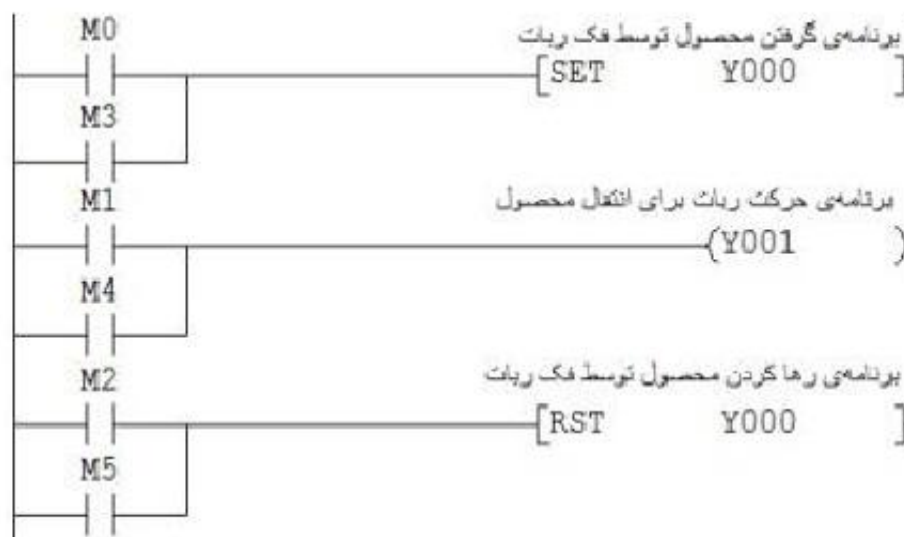
لازم است متذکر شویم که فرآیند کنترل دستی و فرآیند کنترل اتوماتیک ربات با یکدیگر اینترلاک شده‌اند.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند :

ابزار	عملکرد ابزار
X0	به این ورودی PLC، شستی Auto متصل می‌شود. با فشردن این شستی کنتاکت آن از OFF به ON تغییر وضعیت می‌دهد.
X1	به این ورودی، شستی Manual متصل می‌شود. با فشردن این شستی کنتاکت آن از OFF به ON تغییر وضعیت می‌دهد.
X2	به این ورودی از PLC، شستی Clip متصل می‌شود. با فشردن شدن شستی، ورودی ON می‌شود.
X3	به این ورودی PLC، شستی Transfer متصل می‌شود. با فشردن شدن شستی، ورودی ON می‌شود.
X4	به این ورودی، شستی Release متصل می‌شود. با فشردن شدن شستی، ورودی ON می‌شود.
M0~M2	از این ابزارها در فرآیند کنترل اتوماتیک استفاده شده است.
M3~M5	از این ابزارها در فرآیند کنترل دستی استفاده شده است.
M6	از این ابزار برای قطع و وصل کردن تغذیه مدار کنترلی اصلی با دستور MC N0 M6 جهت ریست کردن مدار داخل دو دستورالعمل MC ~ MCR استفاده شده است.
M7	از این ابزار برای قطع و وصل کردن تغذیه مدار کنترلی اصلی با دستور MC N0 M7 جهت ریست کردن مدار داخل دو دستورالعمل MC ~ MCR استفاده شده است.
M10	از این ابزار برای انتخاب کنترل اتوماتیک استفاده شده است.
M11	از این ابزار برای انتخاب کنترل دستی استفاده شده است.
T0	این تایمر برابر با 2 ثانیه تنظیم شده است.
Y0	به این خروجی PLC، گیره (فک) گرفتن و رها کردن بطری متصل شده است. برای گرفتن بطری $Y0 = ON$ شده و برای رها کردن بطری $Y0 = OFF$ می‌شود.
Y1	به این خروجی، مکانیسم حرکت و انتقال بطری (ریات بازو) متصل شده است.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:





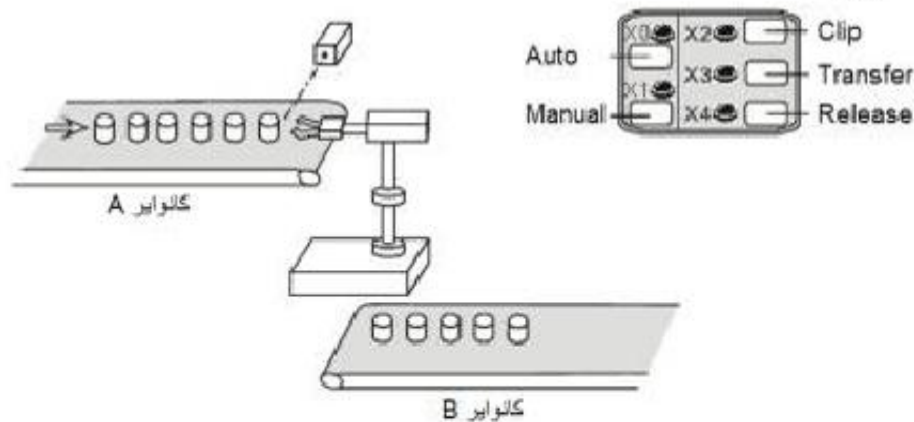
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که شستی متصل شده به ورودی X0 از OFF به ON تغییر وضعیت دهد، فرآیند کنترل اتوماتیک برای یک بار اجرا می‌شود و وقتی که شستی متصل شده به ورودی X1 از OFF به ON تغییر وضعیت دهد، فرآیند کنترل دستی اجرا می‌شود. در فرآیند کنترل دستی، گرفتن و رها کردن بطری توسط فک ربات، نیازمند فشردن شستی‌های مربوطه برای یک بار کار کرد ربات می‌باشد. با فشردن شستی Transfer ربات حرکت کرده و به مدت 2 ثانیه می‌چرخد که در این هنگام بطری به سمت کانوایر B منتقل می‌شود.

شستی‌های متصل شده به ورودی X0 و X1 در برنامه با یکدیگر اینترلاک شده‌اند. وقتی که فرآیند کنترل اتوماتیک اجرا شود، ربات باز شروع به کار می‌کند. ابتدا بطری توسط فک ربات گرفته می‌شود و سپس برای 2 ثانیه ربات چرخیده بطری را به سمت کانوایر B منتقل می‌کند و در نهایت، بطری بر روی کانوایر B قرار گرفته و رها می‌شود. وقتی که فرآیند کنترل اتوماتیک اجرا شود، عمل کنترل در سه مرحله و توسط سه شستی اجرا می‌شود که در مرحله‌ی اول ربات باز در سمت کانوایر A بوده و با فعال شدن خروجی Y0 فک ربات بسته شده و بطری گرفته می‌شود، با فعال شدن خروجی Y1 به مدت 2 ثانیه ربات به سمت کانوایر حرکت می‌کند و در نهایت، با غیرفعال شدن خروجی Y0 فک باز شده و بطری بر روی کانوایر B رها می‌شود.

2- برنامه‌ی کنترل دستی / اتوماتیک با استفاده از دستورالعمل STL

در این مثال، می‌خواهیم برنامه‌ی کنترل دستی / اتوماتیک را با استفاده از دستورالعمل STL برای ربات بررسی نماییم:



هدف از کنترل در این مثال به شرح زیر می‌باشد:

وقتی که شستی **Manual** فشرده شود، ربات باز شروع به کار کرده و به صورت دستی کنترل می‌شود. در حالت کنترل دستی با فشردن دکمه **Clip** ربات قوطی را از روی کانوایر **A** برداشته و با فشردن دکمه **Transfer** بطری را به سمت کانوایر **B** منتقل می‌کند و با فشردن دکمه **Release** بطری را بر روی کانوایر **B** قرار داده و آن را رها می‌کند.

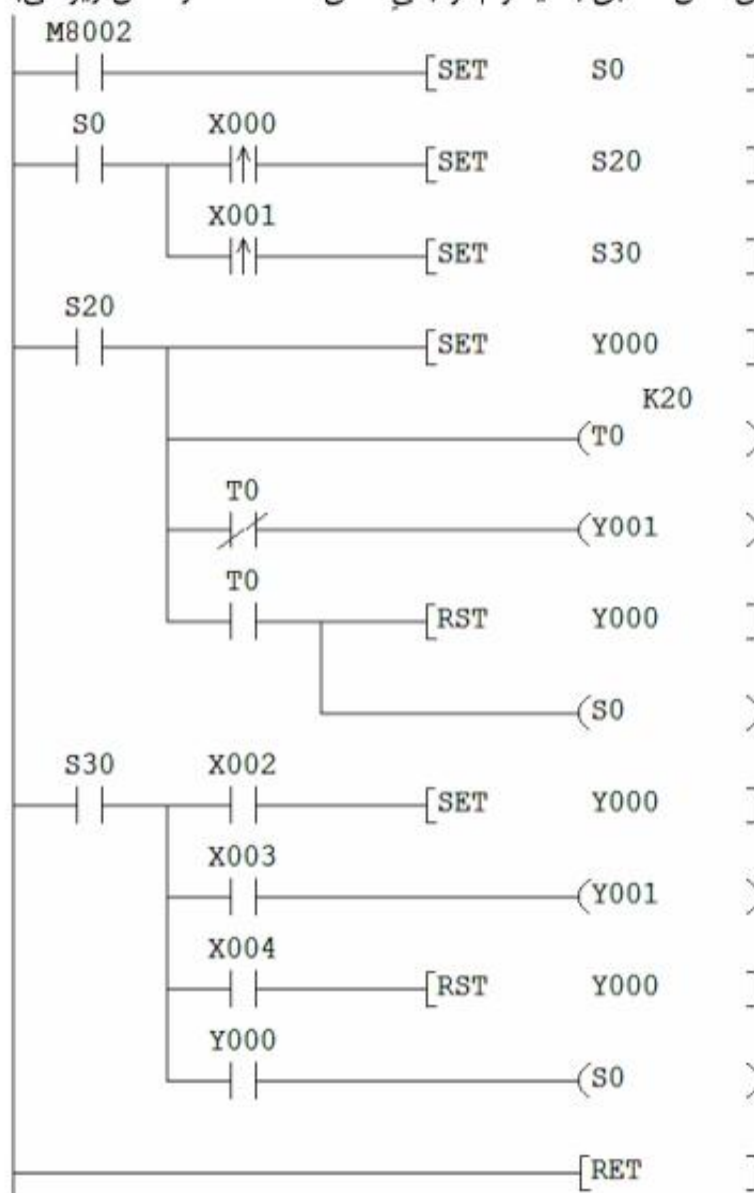
وقتی که شستی **Auto** فشرده شود، ربات باز شروع به کار کرده و برای یک سیکل به صورت اتوماتیک عمل می‌کند. عملکرد اتوماتیک ربات به این صورت است که ابتدا ربات بطری را از روی کانوایر **A** برداشته و آن را نگه می‌دارد، سپس بطری را به سمت کانوایر **B** حرکت می‌دهد که این عمل در مدت زمان 2 ثانیه انجام می‌شود و در نهایت بطری را بر روی کانوایر **B** قرار داده و رها می‌کند. اگر دکمه‌ی **Auto** مجدداً فشرده شود، فرآیند کنترل اتوماتیک ربات دوباره اجرا می‌شود. لازم است متذکر شویم که فرآیند کنترل دستی و فرآیند کنترل اتوماتیک ربات با یکدیگر اینترلاک شده‌اند.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، شستی Auto متصل می‌شود. با فشردن این شستی کنتاکت آن از OFF به ON تغییر وضعیت می‌دهد.
X1	به این ورودی از PLC، شستی Manual متصل می‌شود. با فشردن این شستی کنتاکت آن از OFF به ON تغییر وضعیت می‌دهد.
X2	به این ورودی از PLC، شستی Clip متصل می‌شود. با فشردن شدن شستی، ورودی ON می‌شود.

X3	به این ورودی از PLC، شستی Transfer متصل می‌شود. با فشردن شدن شستی، ورودی ON می‌شود.
X4	به این ورودی از PLC، شستی Release متصل می‌شود. با فشردن شدن شستی، ورودی ON می‌شود.
S0	این ابزار مربوط به مرحله اصلی (نخستین مرحله، اولین مرحله) می‌باشد.
S20	از این ابزار برای مرحله کنترل اتوماتیک استفاده شده است.
S21	این ابزار مربوط به مرحله کنترل دستی می‌باشد.
T0	زمان این تایمر برابر با 2 ثانیه تنظیم شده است.
Y0	به این خروجی از PLC، گیره (فک) گرفتن و رها کردن بطری متصل شده است. برای گرفتن بطری $Y0 = ON$ شده و برای رها کردن بطری $Y0 = OFF$ می‌شود.
Y1	به این خروجی از PLC، مکانیسم حرکت و انتقال بطری توسط ربات بازو متصل شده است.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که ورودی $X0$ از OFF به ON تغییر وضعیت دهد، مرحله‌ی Step S20 فعال شده و فرآیند کنترل اتوماتیک برای یک بار اجرا می‌شود. در همین زمان از اجرا فرآیند کنترل دستی جلوگیری می‌شود. اگر شستی Auto مجدداً فشرده شود، فرآیند کنترل اتوماتیک مجدداً اجرا خواهد شد.

عملکرد کنترل اتوماتیک ربات بازو به این صورت است که با فعال شدن $X0 = ON$ ، فک ربات بسته شده و بطری را می‌گیرد. سپس با فعال شدن $Y1 = ON$ عمل چرخش و انتقال ربات از کانوایر A به کانوایر B برای دو ثانیه اجرا می‌شود. درنهایت، با غیرفعال شدن $Y1 = OFF$ بر روی کانوایر B فک ربات باز شده و بطری بر روی کانوایر B رها می‌شود.

وقتی که ورودی $X1$ از OFF به ON تغییر وضعیت دهد، مرحله‌ی Step S21 فعال شده و فرآیند کنترل دستی برای یک بار اجرا خواهد شد که در این هنگام از اجرای فرآیند کنترل اتوماتیک جلوگیری می‌شود.

عملکرد فرآیند کنترل دستی به این صورت است که ابتدا با فعال شدن ورودی $X2 = ON$ توسط اپراتور، فک ربات بازو بر روی کانوایر A فعال شده و بطری را می‌گیرد، سپس با فعال شدن ورودی $X3 = ON$ توسط اپراتور، ربات بازو برای 2 ثانیه چرخیده و بطری را از روی کانوایر A به روی کانوایر B انتقال می‌دهد و درنهایت، با فعال شدن ورودی $X4 = ON$ توسط اپراتور، فک ربات بازو بر روی کانوایر B باز شده و بطری را بر روی کانوایر رها می‌کند.

2-9 به روز کردن ورودی / خروجی دیجیتال و تنظیم زمان فیلتر ورودی دیجیتال توسط REF/REFF

هدف از کنترل به‌روزرسانی وضعیت ورودی/خروجی‌های دیجیتال بدون واسطه و تنظیم/نمایش زمان فیلتر ورودی دیجیتال است.

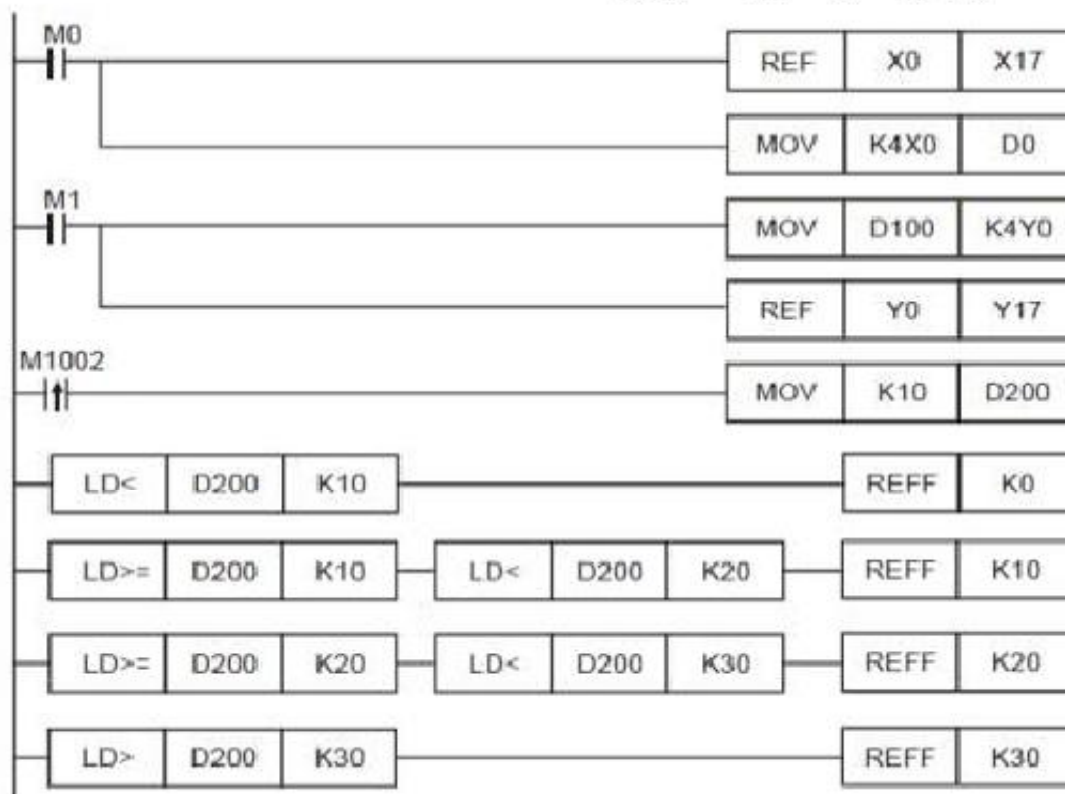
وقتی که $M0=ON$ شود، وضعیت ورودی‌های $X0 \sim X17$ به‌روزرسانی شده و وضعیت‌ها به $D0$ ارسال می‌شود. وقتی که $M1=ON$ شود، مقدار $D100$ به خروجی‌های $Y0 \sim Y17$ ارسال شده و وضعیت‌های خروجی فوراً و بدون واسطه و قبل از دستورالعمل END به ترمینال‌های خروجی ارسال می‌شود.

توسط کنترل مقدار $D200$ ، کاربر می‌تواند زمان فیلتر ورودی دیجیتال، به عنوان مثال $D10$ را (حداقل مقدار واقعی $50MS$)، $10MS$ ، $20MS$ یا $30MS$ تنظیم نماید.

ابزارهای مورد استفاده در این مثال به شرح زیر است:

ابزار	تابع
M0	شروع به به‌روزرسانی وضعیت‌های ورودی‌ها X0~X17
M1	شروع به به‌روزرسانی وضعیت‌های خروجی‌ها Y0~Y17
D200	انباره کردن زمان فیلتر در ورودی‌ها

برنامه کنترلی این مثال مطابق با شکل زیر است:



توضیحات برنامه‌ی فوق به شرح زیر است:

معمولاً وضعیت ورودی (X) در شروع برنامه و همراه سیکل اسکن برنامه به‌روزرسانی می‌شوند، وضعیت‌های خروجی (Y) در پایان برنامه و دستور END به‌روزرسانی می‌شوند. هرچند، وضعیت‌های ورودی/خروجی فوراً و بدون واسطه می‌توانند توسط دستورالعمل Ref در هنگام اجرای برنامه به‌روزرسانی شوند.

به دلیل سخت‌گیری عملکرد (نویز و تداخل امواج) در محیط اطراف، در سیگنال ورودی دیجیتال PLC مکرراً تداخل وجود دارد. بنابراین خطای عملکرد رخ می‌دهد. معمولاً، تداخل امواج برای یک زمان طولانی کم نمی‌باشد. پس می‌توان از یک فیلتر سیگنال ورودی دیجیتال برای کاهش تداخل امواج استفاده کرد.

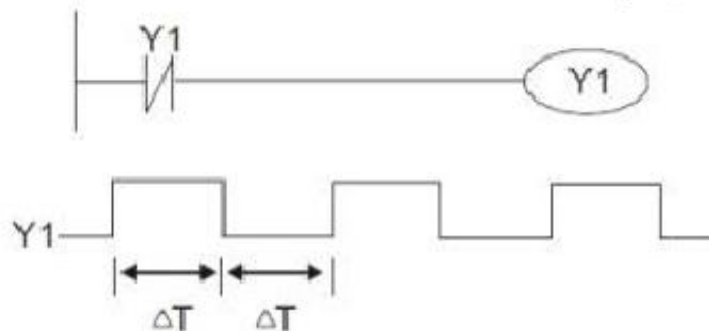
وقتی که $D200 < K10$ باشد، زمان فیلتر سیگنال ورودی دیجیتال برابر 0 است (مقدار واقعی 50MS). وقتی که $K10 \leq D200 < K20$ باشد، زمان فیلتر برابر 10ms است. وقتی که $K20 \leq D200 < K30$ باشد، زمان فیلتر برابر 20ms است. وقتی که $K30 < D200$ باشد، زمان فیلتر برابر 30ms است. تنظیمات اولیه این برنامه در D200 برابر با K10 است. بنابراین زمان فیلتر سیگنال ورودی دیجیتال در این قسمت برابر 0ms تنظیم می‌شود.

با اجرای دستور MOV زمان فیلتر سیگنال ورودی دیجیتال به D1020 (مطابق با X0~X7) و به D1021 (مطابق با X10~X17) منتقل می‌شود.

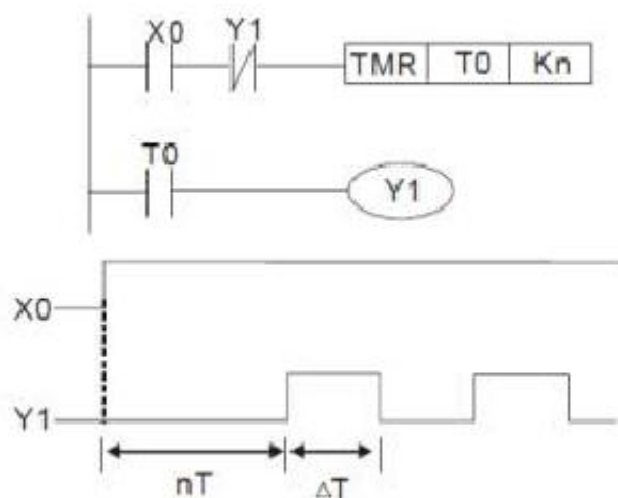
زمان فیلتر شارژ شده می‌تواند با اجرای دستورالعمل REF در هنگام اجرای برنامه پروسه در سیکل اسکن بعدی برنامه اصلاح شود.

2-10 مدار نوسان‌ساز

در این مثال، یک مدار نوسان‌ساز (اسیلاتور) با دوره‌ی زمانی $\Delta T + \Delta T$ را خواهیم ساخت. همانطور که در شکل زیر مشاهده می‌کنید، برنامه این مثال بسیار ساده است. وقتی که اسکن برنامه شروع می‌شود، کنتاکت خروجی Y1 بسته است زیرا بوبین آن فعال نمی‌باشد. در اسکن بعدی برنامه بوبین خروجی Y1 فعال خواهد شد. با فعال شدن بوبین خروجی Y1 کنتاکت بسته‌ی آن باز می‌شود. در اسکن بعدی برنامه به دلیل باز بودن مسیر بوبین خروجی Y1، این خروجی غیرفعال خواهد شد. این روند با هر سیکل اسکن برنامه تکرار شده و یک شکل موج نوسانی با دوره‌ی زمانی $\Delta T(\text{on}) + \Delta T(\text{off})$ را خواهد ساخت.

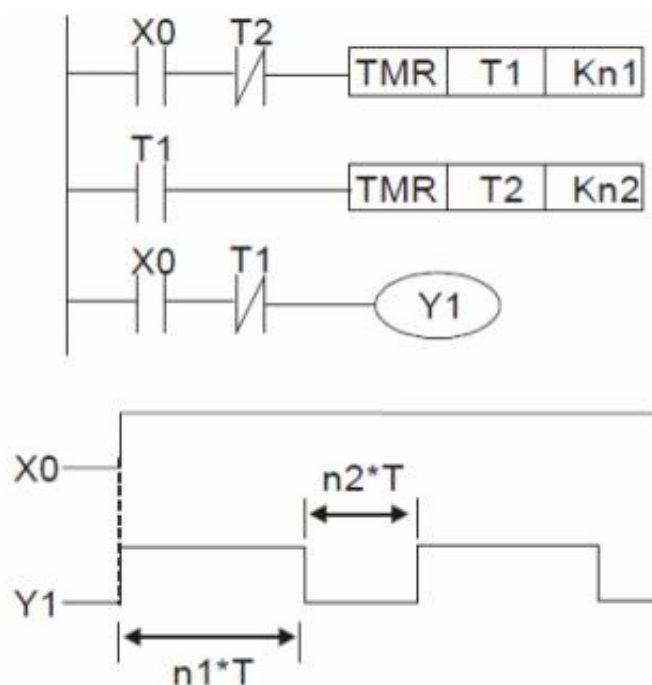


حال می‌خواهیم یک مدار نوسان‌ساز با دوره‌ی زمانی $nT + \Delta T$ را بسازیم. برای این کار لازم است که از یک تایمر استفاده کنیم. ΔT همان زمان یک سیکل برنامه بوده و nt زمان تنظیمی برای تایمر (Kn) می‌باشد. توسط تایمر می‌توان زمان غیرفعال بودن خروجی Y1 را کنترل کرد و زمان فعال بودن خروجی Y1 برابر با یک سیکل اسکن برنامه می‌باشد. در اینجا n برابر با مقدار تنظیمی تایمر برحسب دسیمال است.



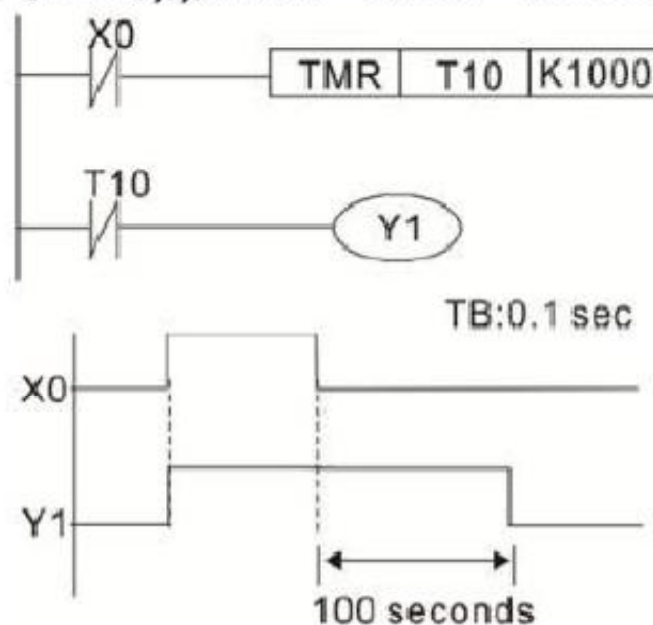
2- 11 مدار چشمک‌زن

همانطور که در شکل زیر مشاهده می‌کنید، برای ساختن یک مدار چشمک‌زن باید از یک مدار نوسان‌ساز استفاده کرد. از مدار چشمک‌زن برای روشن خاموش کردن چراغ سیگنال آلارم به صورت چشمک‌زن و یا به صدا در آوردن بیزر به صورت نوسانی استفاده می‌شود. برای این کار باید از دو تایمر استفاده کرد. از یک تایمر برای کنترل زمان On بودن و از تایمر دیگر برای کنترل زمان OFF بودن خروجی Y1 استفاده می‌شود. در اینجا از $n1$ و $n2$ برای تنظیم مقدار زمان تایمر T1 و T2 استفاده می‌شود.



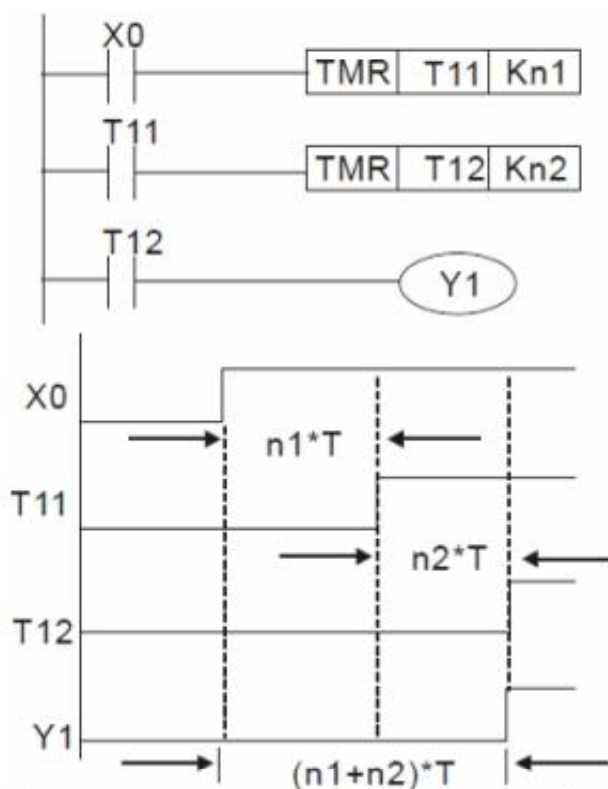
2- 12 مدار تأخیر زمانی

وقتی که ورودی $X1=ON$ شود، تایمر $T10$ به دلیل استفاده از کنتاکت بسته‌ی ورودی $X1$ غیرفعال شده و خروجی $Y1$ فعال خواهد شد. به محض اینکه ورودی $X1$ غیرفعال شود، تایمر شروع به شمارش کرده و بعد گذشت مدت زمان تنظیم شده خروجی $Y1$ غیرفعال می‌شود. در اینجا چون مقدار تنظیمی برابر با 1000 دسیمال بوده و واحد تایمر $T10$ برابر با 100 میلی ثانیه است، مقدار زمان تنظیمی مطابق با فرمول $1000 \times 0.1 \text{ sec} = 100 \text{ sec}$ برابر با 100 ثانیه می‌باشد.

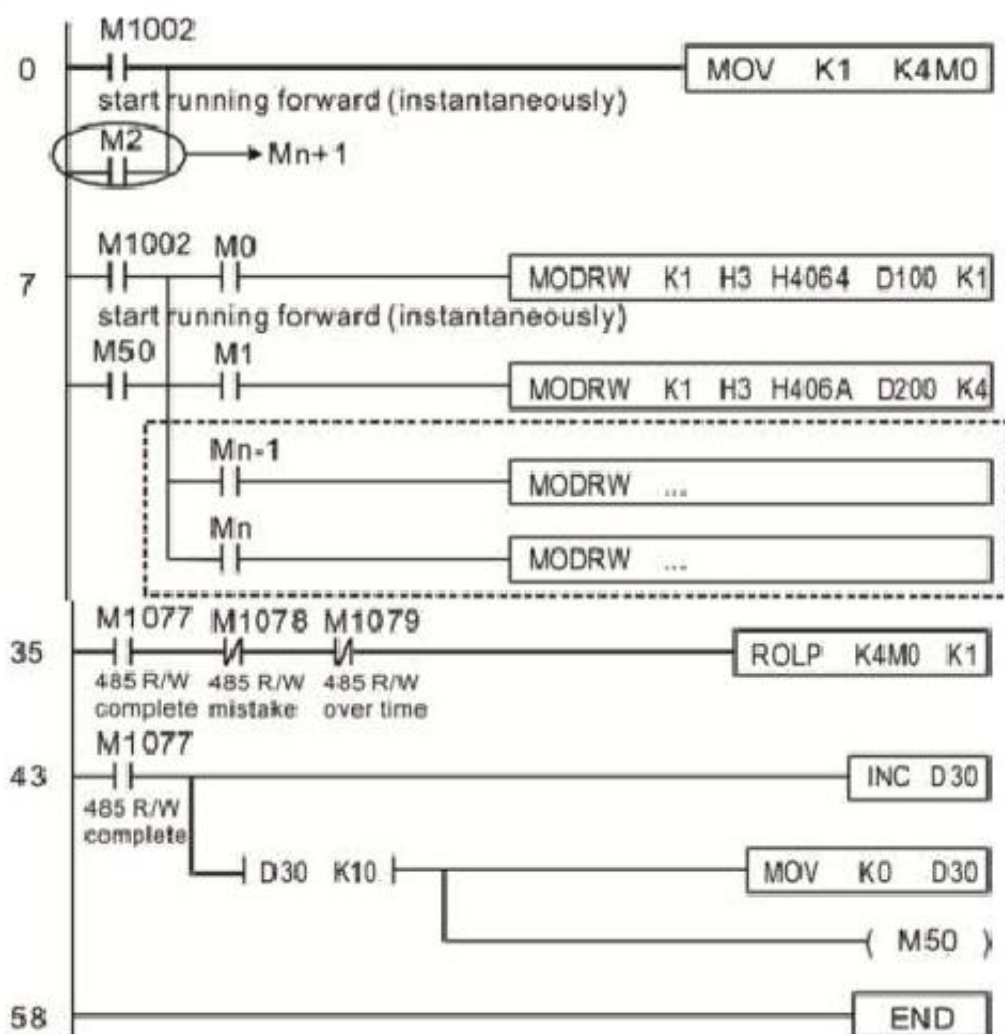


2- 13 توسعه دادن مقدار زمان تایمر

همانطور که در مدار شکل زیر مشاهده می‌کنید، برای توسعه دادن زمان شمارش از دو تایمر استفاده کرده‌ایم. با فعال شدن ورودی $X0$ ابتدا تایمر $T11$ شروع به شمارش کرده و بعد از سپری شدن زمان تنظیمی خروجی آن فعال می‌شود. با فعال شدن خروجی تایمر $T11$ تایمر $T12$ شروع به شمارش کرده و با سپری شدن مدت زمان تنظیم شده خروجی $Y1$ فعال خواهد شد. مدت زمان تنظیمی از فرمول $(n1 + n2) \times T$ بدست می‌آید. $n1$ و $n2$ مقدار دسیمال بوده و T واحد زمانی تایمر می‌باشد.



بعنوان مثال، وقتی که PLC شروع به کار کند و در این هنگام اگر فلگ M0 فعال (ON) باشد، دستورالعمل MODRW یکبار اجرا خواهد شد. بعد از اینکه دستگاه‌های Slave پاسخ را دریافت کردند، اگر دستورالعمل صحیح بود، فرمان ROL یک بار اجرا می‌شود تا M1 فعال (ON) شود. بعد از اینکه دستگاه Slave پاسخ را دریافت کرد، بعد از یک تأخیر برابر با 10 سیکل اسکن برنامه فلگ M50 فعال (1) شده و سپس دستورالعمل MODRW یکبار اجرا خواهد شد. بعد از اینکه مجدداً دستگاه Slave پاسخ را دریافت کرد، اگر فرمان صحیح بود، دستورالعمل ROL یکبار اجرا خواهد شد و فلگ M2 به وضعیت ON تغییر خواهد کرد، K4M0 به K1 تغییر کرده و فقط فلگ M0 فعال (1) باقی خواهد ماند. عمل انتقال دیتا می‌تواند در هر سیکل به طور پیوسته اجرا شود.

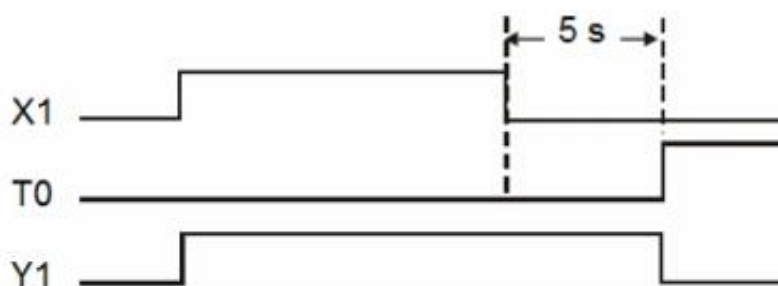


2- 14- تایمر تأخیر در قطع

در این مثال، می‌خواهیم برنامه تایمر تأخیر در قطع را بررسی نماییم.

هدف از کنترل در این مثال به شرح زیر می‌باشد:

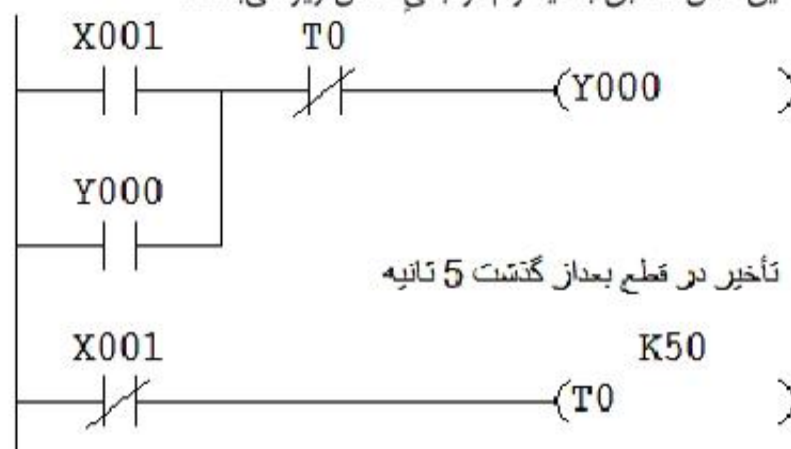
فعال کردن یک چراغ سیگنال بعد از وصل (ON) کردن سوئیچ و فعال ماندن همان چراغ سیگنال به مدت 5 ثانیه بعد از قطع (OFF) کردن سوئیچ می‌باشد. بعد از گذشت 5 ثانیه از قطع شدن سوئیچ، چراغ سیگنال باید خاموش شود.



ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X1	به این ورودی PLC، سوئیچ قفل شونده متصل می‌شود. اگر سوئیچ در وضعیت C قرار گیرد، کنتاکت آن OFF می‌باشد.
T1	تایمر 5 ثانیه. ضرب این تایمر 100 میلی‌ثانیه است. برای تنظیم E ثانیه باید مقدار تایمر را برابر با K50 تنظیم نماییم.
Y1	به این خروجی از PLC، چراغ سیگنال متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که سوئیچ فعال شود، ورودی $X1 = ON$ می‌شود. کنتاکت سخت‌افزاری $X1$ ، یک کنتاکت در حالت عادی باز می‌باشد و در برنامه برای این ورودی دو کنتاکت در حالت عادی باز (NO) و در حالت عادی بسته (NC) در نظر گرفته شده است. از کنتاکت بسته $X1$ برای فعال کردن تایمر استفاده می‌شود. با فعال شدن کنتاکت بسته $X1$ در نرم‌افزار، دستورالعمل TMR اجرا نخواهد شد که در این حالت بوبین تایمر T1 غیرفعال (OFF) می‌باشد، بنابراین کنتاکت بسته خواهد بود. با فعال شدن ورودی سخت‌افزاری $X1 = ON$ ، چراغ سیگنالی که به خروجی $Y1$ وصل شده است، روشن شده و روشن می‌ماند.

وقتی که سوئیچ غیرفعال شود، ورودی $X1 = OFF$ می‌شود، کنتاکت بسته $X1$ در نرم‌افزار فعال نخواهد شد (یعنی کنتاکت بسته می‌ماند) که در این حالت دستورالعمل TMR اجرا می‌شود. چراغ

سیگنالی که به خروجی Y1 متصل شده است توسط مدار قفل شونده (خود نگهدارنده) تا وقتی که زمان واقعی تایمر به زمان تنظیم شده برسد، روشن باقی خواهد ماند.

وقتی که زمان واقعی تایمر T1 به مقدار تنظیمی (5 ثانیه) رسید، بوبین تایمر T1 فعال (ON) خواهد شد. کنتاکت NC تایمر T1 فعال شده و چراغ سیگنالی که به خروجی Y1 متصل است، خاموش می‌شود.

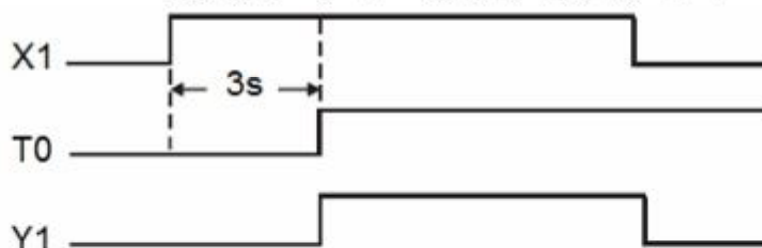
توسط دستورالعمل STMR نیز می‌توان فانکشن تایمر تأخیر در قطع (Delay OFF) را اجرا کرد.

2- 15- تایمر تأخیر در وصل

در این مثال، می‌خواهیم برنامه تایمر تأخیر در وصل (Delay ON) را بررسی نماییم.

هدف از کنترل در این برنامه به شرح زیر می‌باشد:

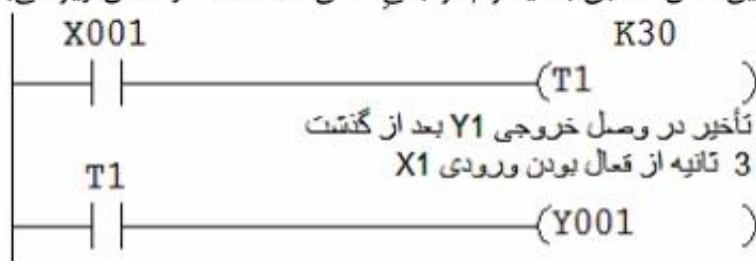
در این برنامه می‌خواهیم با فعال (ON) شدن یک سوئیچ بعد از گذشت 3 ثانیه یک چراغ سیگنال را روشن کنیم و تا هنگام غیرفعال کردن سوئیچ آن لامپ سیگنال روشن بماند.



ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X1	به این ورودی PLC، یک سوئیچ قفل شونده‌ی دو وضعیت O-I متصل می‌شود. وقتی که این سوئیچ در وضعیت I قرار گیرد، ورودی $X0 = ON$ می‌شود.
T1	تایمر 3 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 3 ثانیه باید مقدار تایمر را برابر با K30 تنظیم نماییم.
Y1	به این خروجی PLC یک چراغ سیگنال متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که ورودی $X1 = ON$ شود، دستورالعمل TMR اجرا خواهد شد. تایمر T1 فعال (ON) شده و برای 3 ثانیه شروع به شمارش می‌کند. وقتی که مقدار جاری تایمر T1 به مقدار تنظیمی تایمر رسید، کنتاکت باز تایمر فعال شده و بسته خواهد شد که در این حالت چراغ نمایشگر متصل شده به خروجی Y1 روشن می‌شود.

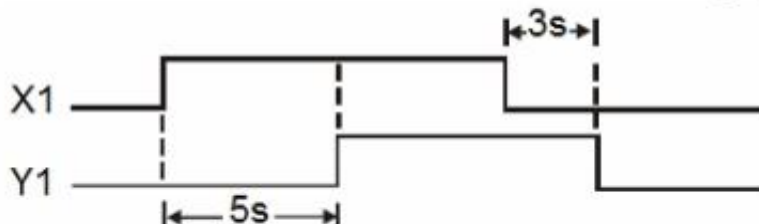
وقتی که ورودی $X1 = OFF$ شود، دستورالعمل TMR اجرا نخواهد شد. تایمر T1 غیرفعال OFF شده و بنابراین کنتاکت تایمر T1 باز خواهد شد. از این رو، چراغ سیگنال متصل شده به خروجی Y1 خاموش می‌شود.

2- 16- تایمر تاخیر در قطع و تاخیر در وصل

در این مثال، می‌خواهیم برنامه تایمر تاخیر در قطع و تاخیر در وصل Delay ON/OFF را بررسی نماییم.

هدف از کنترل در این مثال به شرح زیر می‌باشد:

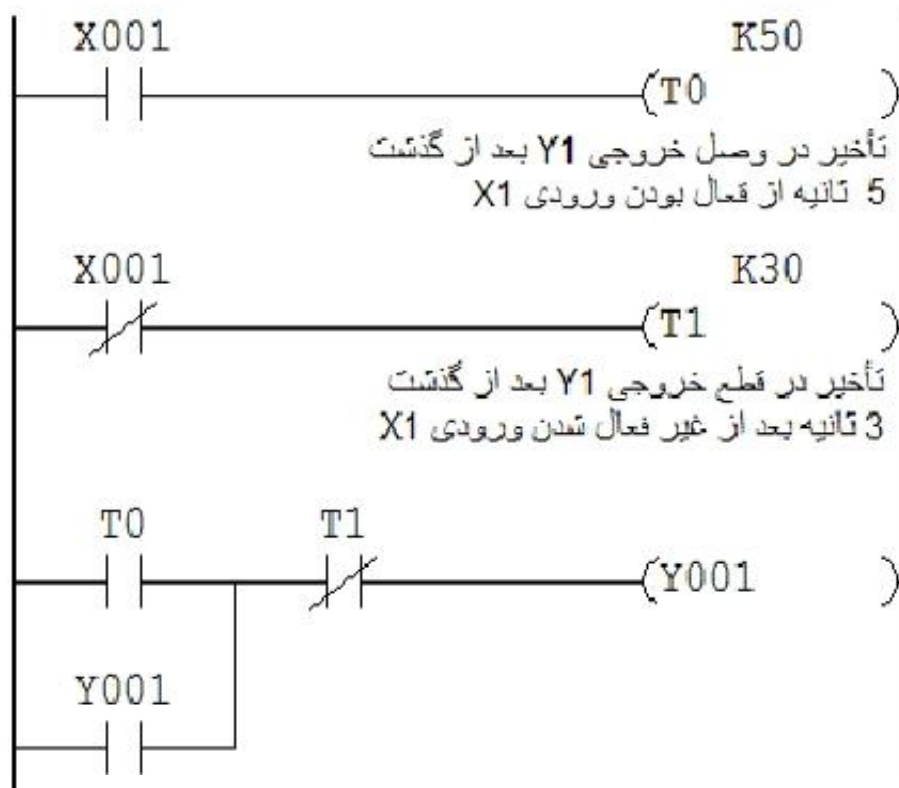
با فعال کردن سوئیچ دو وضعیتی، ورودی $X1 = ON$ شده و بعد از گذشت 5 ثانیه چراغ سیگنال روشن می‌شود و با غیرفعال کردن سوئیچ قفل شونده، چراغ سیگنال به مدت 3 ثانیه روشن باقی بماند و سپس چراغ خاموش شود.



ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X1	به این ورودی، سوئیچ دو وضعیتی O-I قفل شونده متصل می‌شود. اگر این سوئیچ در وضعیت I قرار گیرد، ورودی $X1 = ON$ می‌شود.
T0	تایمر 5 ثانیه ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 5 ثانیه باید مقدار تایمر برابر با K50 تنظیم شود.
T1	تایمر 3 ثانیه ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 3 ثانیه باید مقدار تایمر را برابر با K30 تنظیم شود.
Y1	به این خروجی PLC یک چراغ سیگنال متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



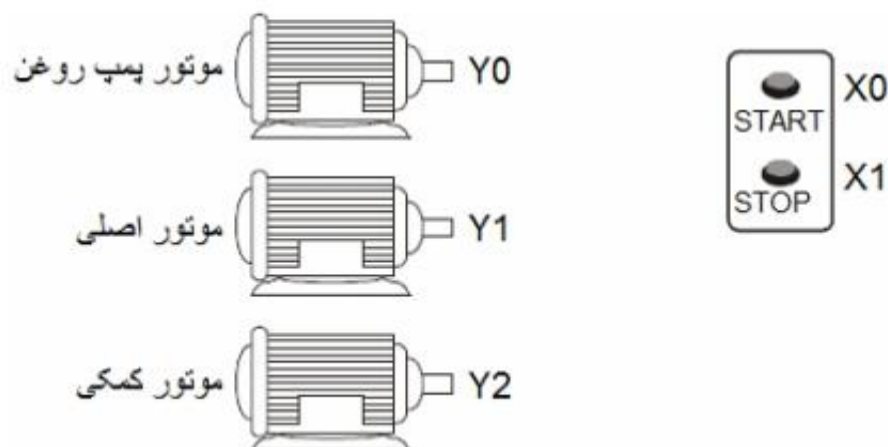
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که ورودی $X1 = ON$ شود، تایمر $T0$ برای 5 ثانیه شروع به شمارش می‌کند. وقتی که مقدار جاری تایمر به مقدار تنظیمی آن رسید، کنتاکت باز تایمر $T0$ فعال (بسته) می‌شود. همانطور که در شکل مشاهده می‌کنید، کنتاکت بسته تایمر $T1$ در این زمان بسته باقی خواهد ماند. بنابراین، چراغ سیگنال متصل شده به خروجی $Y1$ روشن شده و روشن باقی می‌ماند.

وقتی که ورودی $X1 = OFF$ شود، تایمر $T1$ برای مدت 3 ثانیه شروع به شمارش می‌کند. وقتی که مقدار جاری تایمر $T1$ به مقدار تنظیمی آن رسید، کنتاکت بسته تایمر $T1$ فعال (باز) می‌شود. از آنجایی که کنتاکت باز تایمر $T0$ غیرفعال (باز) می‌باشد، چراغ متصل شده به خروجی $Y1$ خاموش می‌شود.

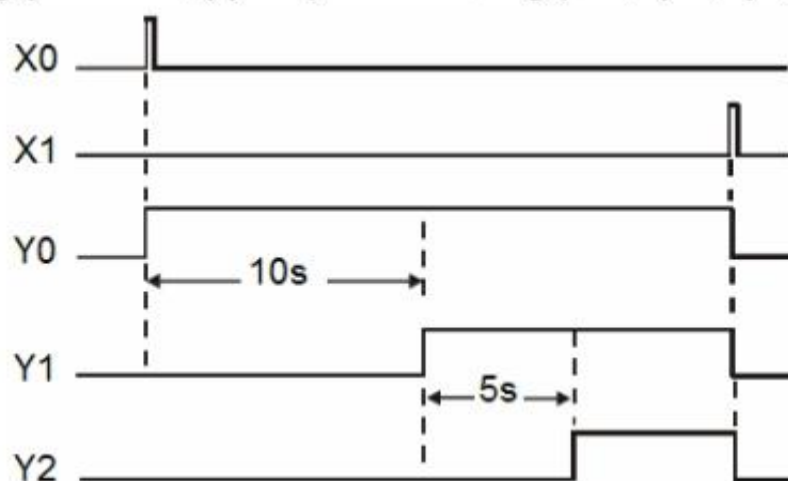
2- 17 راه‌اندازی 3 الکتروموتور به صورت ترتیبی با تایمر تأخیری

در این مثال، می‌خواهیم برنامه راه‌اندازی 3 الکتروموتور را به صورت ترتیبی با تایمر تأخیری بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

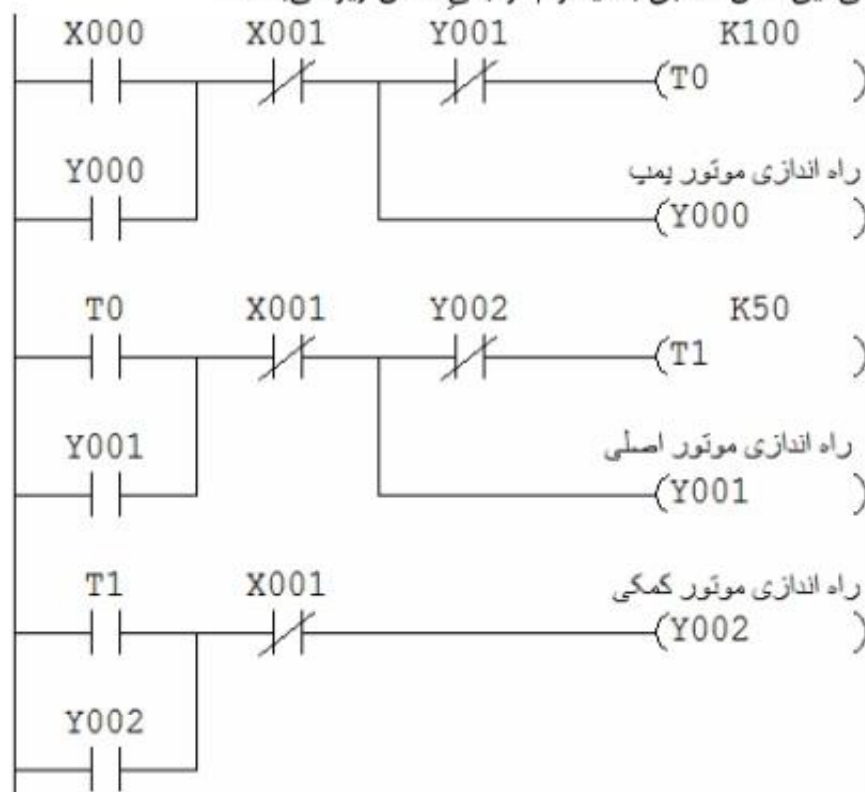
در این مثال، می‌خواهیم با فشردن دکمه START، موتور پمپ روغن روشن شود. سپس بعد از گذشت 10 ثانیه موتور اصلی روشن شده و درنهایت، با روشن شدن موتور اصلی بعد از گذشت 5 ثانیه موتور کمکی روشن شود. به‌علاوه، با فشردن دکمه STOP هر سه موتور بلافاصله خاموش شوند.



ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی، شستی START متصل می‌شود. وقتی شستی فشرده شود، ورودی فعال (ON) می‌شود.
X1	به این ورودی PLC، شستی STOP متصل می‌شود. وقتی شستی فشرده شود، ورودی فعال (ON) می‌شود.
T0	تایمر 10 ثانیه، ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 10 ثانیه باید مقدار تایمر را K100 تنظیم کرد.
T1	تایمر 5 ثانیه، ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 5 ثانیه باید مقدار تایمر را K50 تنظیم کرد.
Y0	به این خروجی از PLC، موتور پمپ روغن متصل می‌شود.
Y1	به این خروجی از PLC، موتور اصلی متصل می‌شود.
Y2	به این خروجی از PLC، موتور کمکی متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که شستی **START** فشرده شود، کنتاکت باز ورودی **X0** فعال (بسته) می‌شود، در این هنگام خروجی **Y0** فعال شده و فعال باقی می‌ماند. این عمل باعث روشن شدن موتور پمپ روغن می‌شود. در این هنگام دستورالعمل **[T0 K100]** اجرا خواهد شد. وقتی که مقدار جاری تایمر **T0** به مقدار تنظیمی **10** ثانیه رسید، کنتاکت باز تایمر **T0** فعال (ON) می‌شود.

وقتی که کنتاکت باز تایمر **T0** فعال (ON) شد، خروجی **Y1** فعال شده و فعال باقی خواهد ماند که در این هنگام موتور اصلی روشن شده و تایمر **T0** متوقف می‌شود. در همین زمان، دستورالعمل **[T1 K50]** اجرا می‌شود و هنگامی که مقدار جاری تایمر **T1** به مقدار تنظیمی **5** ثانیه برسد، کنتاکت باز این تایمر بسته شده و ON می‌شود.

وقتی که کنتاکت باز تایمر **T1** فعال (ON) شد، خروجی **Y2** فعال شده و فعال باقی خواهد ماند. در این هنگام موتور کمکی روشن شده و تایمر **T1** متوقف می‌شود.

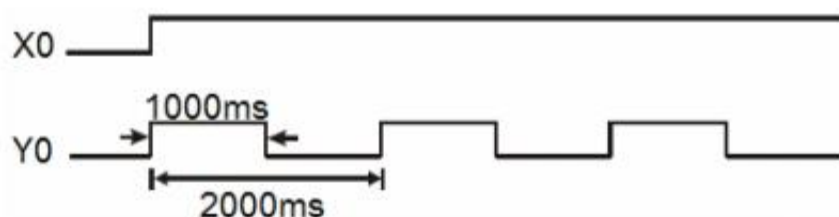
وقتی که دکمه **STOP** فشرده شود، کنتاکت بسته ورودی **X1** فعال (باز) خواهد شد. در این هنگام هر سه خروجی **Y0**، **Y1**، **Y2** غیرفعال (OFF) شده و موتور پمپ روغن، موتور اصلی و موتور کمکی متوقف می‌شوند.

2- 18- مدولاسیون پهنای پالس

در این مثال، می‌خواهیم مدولاسیون پهنای پالس را بررسی نماییم.

هدف از کنترل در این مثال به شرح زیر می‌باشد:

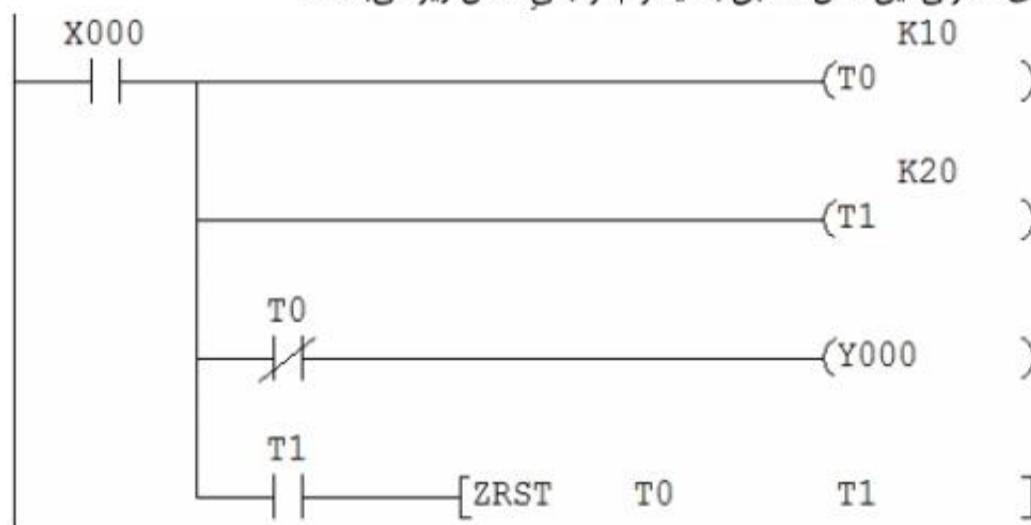
در این مثال، می‌خواهیم با تغییر مقدار تنظیمی تایمرها در برنامه، تابع مدولاسیون پهنای پالس (Pulse-Width Modulation) را اجرا نماییم. نوسان پالس در این مثال به صورت $Y0 = ON$ برای یک ثانیه و زمان یک سیکل برابر با 2 ثانیه می‌باشد. یعنی در یک سیکل 2 ثانیه‌ای، 1 ثانیه خروجی $Y0$ فعال بوده و برای یک ثانیه خروجی $Y0$ غیرفعال می‌باشد و این سیکل به صورت مداوم تکرار خواهد شد.



ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC، سوئیچ دو وضعیت O-I قفل شونده متصل می‌شود. وقتی که سوئیچ در وضعیت I قرار گیرد، ورودی $X1 = ON$ می‌شود.
T0	تایمر 1 ثانیه. ضرب این تایمر 100 میلی‌ثانیه است. برای تنظیم 1 ثانیه باید مقدار تایمر را برابر با K10 تنظیم نماییم.
T1	تایمر 2 ثانیه. ضرب این تایمر 100 میلی‌ثانیه است. برای تنظیم 2 ثانیه باید مقدار تایمر را برابر با K20 تنظیم نماییم.
Y0	با اجرا شدن برنامه‌ی طراحی شد، در این خروجی پالس نوسانی وجود خواهد داشت.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:

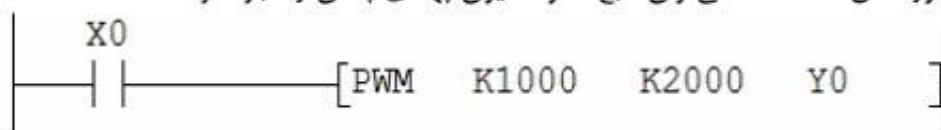


عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که ورودی $X0 = ON$ شود، تایمر $T0$ و $T1$ فعال خواهند شد. تا وقتی که مقدار جاری تایمر $T0$ به مقدار تنظیمی 1 ثانیه برسد، خروجی $Y0$ فعال (ON) خواهد بود. وقتی که مقدار جاری تایمر $T1$ به مقدار تنظیمی 2 ثانیه رسید، هر دو تایمر $T0$ و $T1$ ریست خواهند شد. بنابراین، با فعال بودن سوئیچ متصل شده به ورودی، خروجی $Y0$ به صورت نوسانی و پیوسته ON و OFF می‌شود. وقتی که ورودی $X0 = OFF$ شود، خروجی $Y0$ غیرفعال (OFF) خواهد شد.

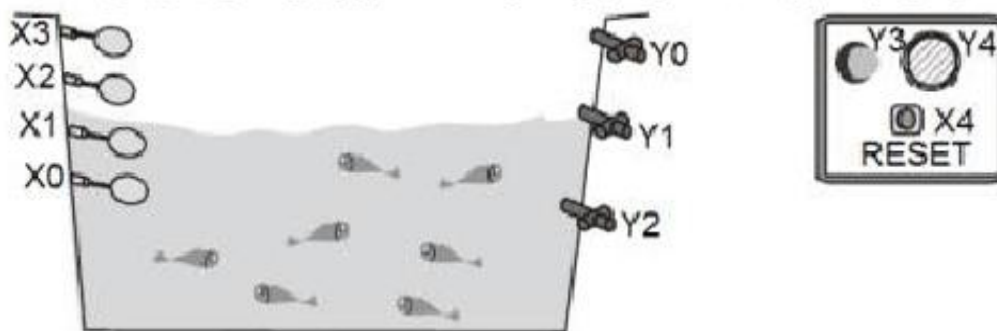
توسط تغییر زمان تنظیمی تایمرها در برنامه، می‌توانید پهنای پالس مربوط به تابع مدولاسیون پهنای پالس را اصلاح نمایید.

توسط دستورالعمل PWM می‌توان تابع مدولاسیون پهنای پالس را اجرا کرد.



2- 19 مانیتور کردن سطح آب استخر پرورش ماهی

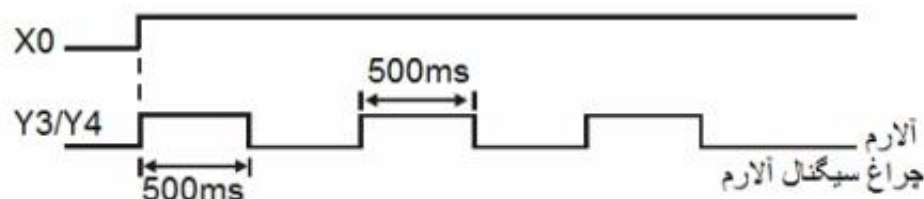
در این مثال می‌خواهیم برنامه‌ی مانیتور کردن سطح آب استخر پرورش ماهی را بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

وقتی که سطح آب استخر پرورش ماهی در ناحیه نرمال قرار ندارد، آب استخر به صورت اتوماتیک پر شود. بعلاوه، در هنگام پر شدن آب، اگر آب از سطح آلارم بالا بودن سطح آب، بالاتر رفت و یا در هنگام کم شدن آب، آب از سطح آلارم پایین بودن سطح آب، پایین‌تر رفت چراغ آلارم به صورت چشمک‌زن روشن شود.

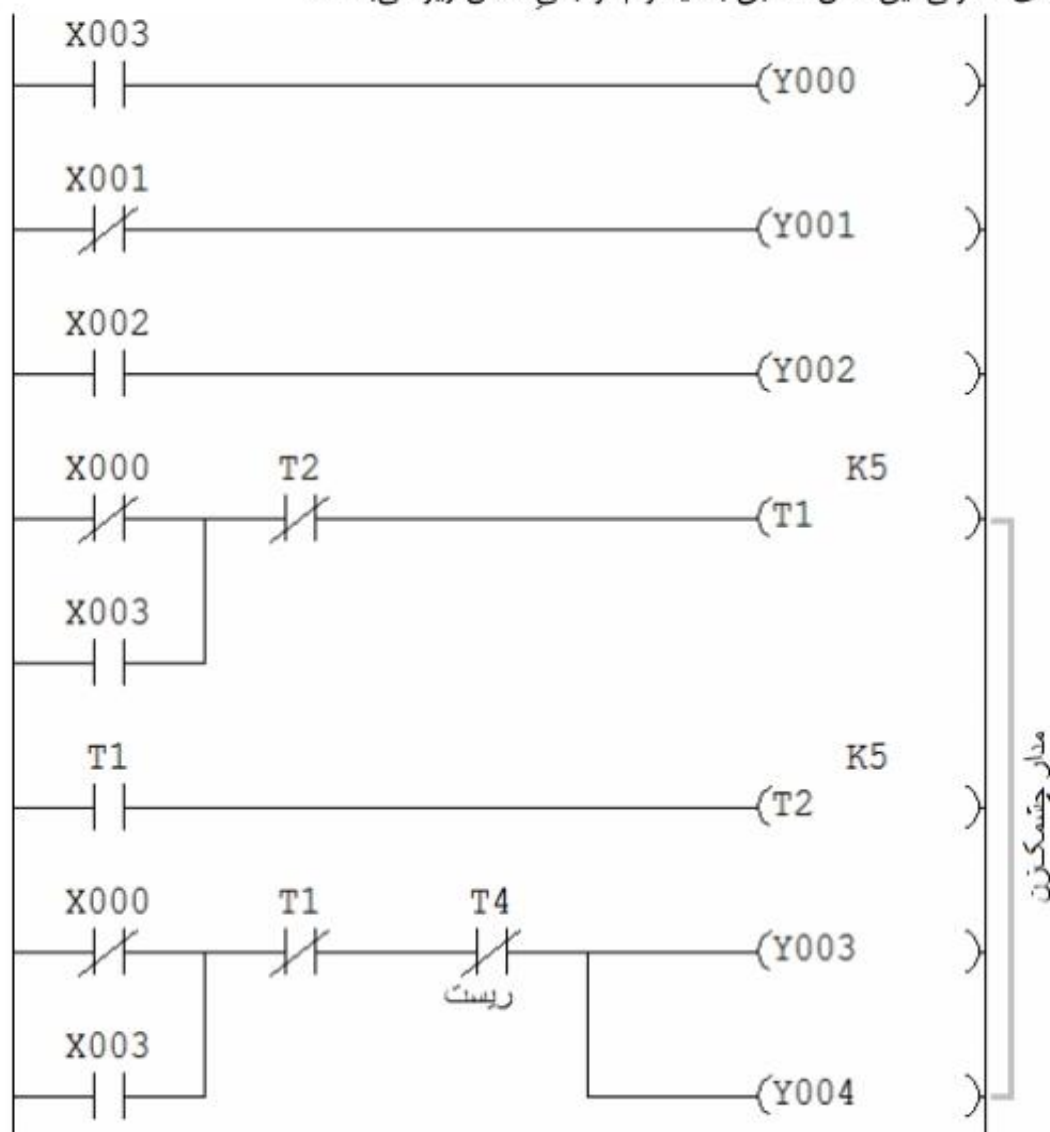
با فشردن دکمه RESET، آلارم متوقف شده و چراغ خاموش شود.



ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزارها
X0	به این ورودی از PLC، لول سنچ آلارم مربوط به پایین‌ترین سطح آب متصل می‌شود. اگر آب بالاتر از این سطح باشد، ورودی $X0 = ON$ می‌شود.
X1	به این ورودی از PLC، لول سنچ نرمال بودن مربوط به سطح پایین آب متصل می‌شود. اگر آب بالاتر از این سطح باشد، ورودی $X1 = ON$ می‌شود.
X2	به این ورودی از PLC، لول سنچ نرمال بودن مربوط به سطح بالای آب متصل می‌شود. اگر آب بالاتر از این سطح باشد، ورودی $X2 = ON$ می‌شود.
X3	به این ورودی از PLC، لول سنچ آلارم مربوط به بالاترین سطح آب متصل می‌شود. اگر آب بالاتر از این سطح باشد، ورودی $X3 = ON$ می‌شود.
X4	به این ورودی از PLC، شستی RESET متصل می‌شود. وقتی که شستی فشرده شود، ورودی $X4 = ON$ می‌شود.
T1	تایمر 500 میلی‌ثانیه. ضرب این تایمر 100 میلی‌ثانیه است. برای تنظیم 500 میلی‌ثانیه مقدار تایمر را برابر با K5 تنظیم کنید.
T2	تایمر 500 میلی‌ثانیه. ضرب این تایمر 100 میلی‌ثانیه است. برای تنظیم 500 میلی‌ثانیه مقدار تایمر را برابر با K5 تنظیم کنید.
Y0	به این خروجی PLC، نخستین پمپ Drainage متصل می‌شود.
Y1	به این خروجی PLC، پمپ تغذیه آب استخر متصل می‌شود.
Y2	به این خروجی PLC، دومین پمپ Drainage متصل می‌شود.
Y3	به این خروجی PLC، چراغ سیگنال آلارم متصل می‌شود.
Y4	به این خروجی PLC، آلارم بیزر متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که آب در سطح نرمال باشد، وضعیت ورودی‌های $X0 = ON$, $X1 = ON$, $X2 = OFF$ و $X3 = OFF$ می‌باشند. بنابراین، خروجی‌های $Y0$ و $Y2$ غیرفعال (OFF) خواهند بود. از این رو، هر دو پمپ Drainage و پمپ تغذیه کار نخواهند کرد.

وقتی که آب استخر به پایین‌تر از سطح نرمال رفت، ورودی‌های $X0 = ON$, $X1 = OFF$, $X2$ و $X3 = OFF$ می‌شوند. اگر $X1 = OFF$ شود، خروجی $Y1$ فعال (ON) می‌شود. در این حالت، پمپ تغذیه شروع به کار می‌کند.

وقتی که آب استخر به پایین‌تر از کمترین سطح آلام رفت، ورودی‌ها $X0 = OFF$ ، $X1 = OFF$ می‌شوند. اگر ورودی $X1 = OFF$ شود، خروجی $Y1$ فعال (ON) می‌شود و پمپ تغذیه شروع به کار می‌کند. بعلاوه، اگر ورودی $X0 = OFF$ شود، مدار آلام چشمک‌زن فعال شده و هر دو خروجی $Y3$ و $Y4$ با هم فعال می‌شوند که باعث راه‌اندازی آلام بیزر و آلام چشمک‌زن می‌شود.

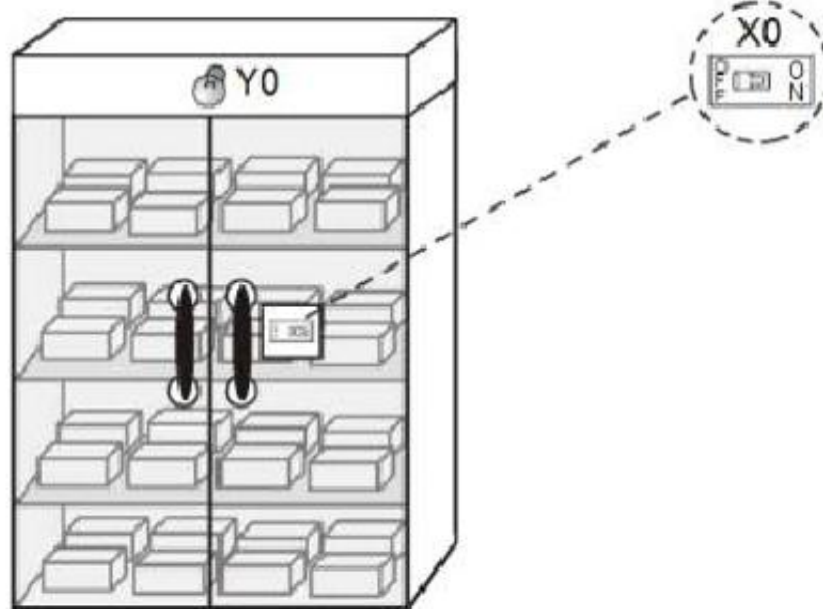
وقتی که آب استخر به بالاتر از سطح نرمال رفت، وضعیت ورودی‌های $X0 = ON$ ، $X1 = ON$ می‌شوند. اگر ورودی $X2 = ON$ شود، خروجی $Y2$ فعال شده که باعث راه‌اندازی دومین پمپ Drainage خواهد شد و آب استخر پر می‌شود.

وقتی که آب استخر از آلام سطح بالا بودن آب بالاتر رفت، وضعیت ورودی‌ها $X0 = ON$ ، $X1 = ON$ می‌شوند. اگر ورودی $X2 = ON$ شود، خروجی $Y2$ فعال خواهد شد که در این حالت، دومین پمپ Drainage شروع به کار می‌کند. گذشته از این، مدار آلام اجرا خواهد شد ($Y3 = Y4 = ON$).

وقتی که دکمه‌ی RESET فشرده شود، کنتاکت بسته‌ی ورودی $X4$ فعال شده و خروجی‌های $Y3$ و $Y4$ را غیرفعال (OFF) می‌کند که در این حالت، هر دو آلام بیزر و چشمک‌زن خاموش خواهند شد.

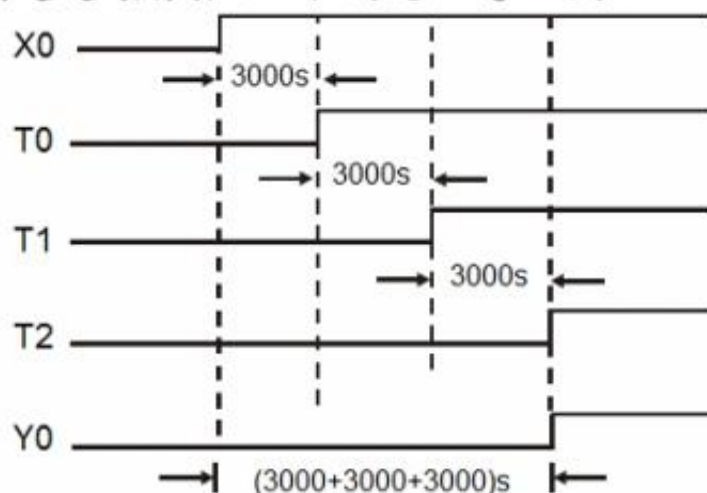
20-2 کنترل کوره پخت

در این مثال می‌خواهیم عملکرد یک کوره پخت را بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

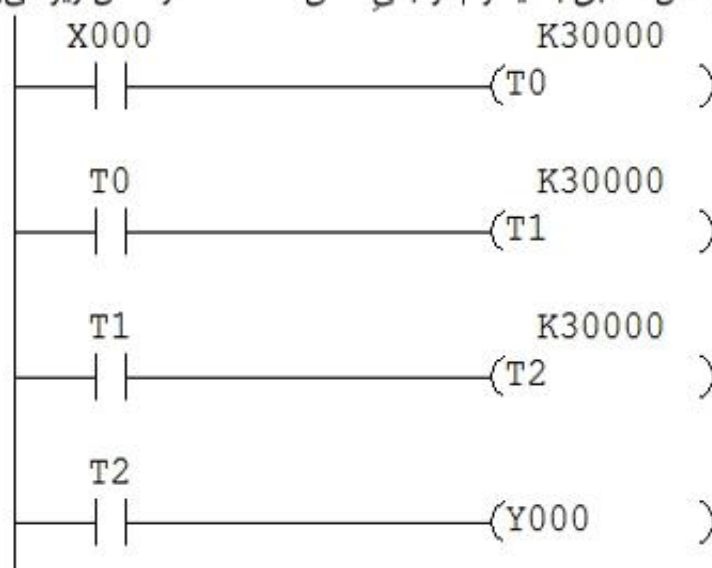
با روشن کردن کوره‌ی پخت، فرآیند پخت به مدت دو و نیم ساعت اجرا شود و بعد از تکمیل فرآیند پخت یک آلارم توسط PLC برای اطلاع تکمیل فرآیند پخت به اپراتور روشن می‌شود.



ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC، سوئیچ راه‌اندازی کوره‌ی پخت متصل می‌شود. برای فعال کردن این ورودی وضعیت سوئیچ را به ON تغییر وضعیت دهید.
T0	تایمر 3000 ثانیه، ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 3000 ثانیه باید مقدار تایمر را برابر با K 30000 تنظیم کرد.
T1	تایمر 3000 ثانیه، ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 3000 ثانیه باید مقدار تایمر را برابر با K 30000 تنظیم کرد.
T2	تایمر 3000 ثانیه، ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 3000 ثانیه باید مقدار تایمر را برابر با K 30000 تنظیم کرد.
Y0	به این خروجی از PLC، آلارم تکمیل فرآیند پخت متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

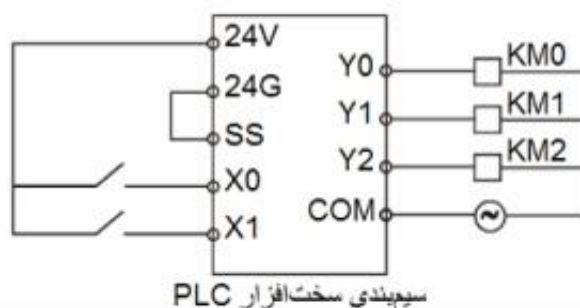
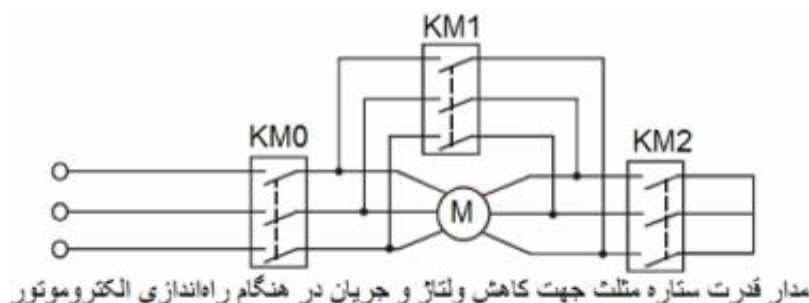
مقدار کران بالا (حد بالا) برای یک تایمر 16 بیتی برابر با $100\text{ms} * 32767 = 3276.7\text{s}$ می‌باشد. یک ساعت برابر با 3600 ثانیه می‌باشد. بنابراین، برای تنظیم یک ساعت به دو تایمر و برای تنظیم دو و نیم ساعت به سه تایمر نیاز می‌باشد.

وقتی که ورودی $X0 = \text{ON}$ شود، کوره‌ی پخت شروع به کار نموده و تایمر $T0$ برای $100\text{ms} * 30000 = 3000\text{s}$ شروع به کار می‌کند. وقتی که مقدار جاری تایمر $T0$ به مقدار تنظیمی 3000 ثانیه رسید، کنتاکت باز تایمر $T0$ فعال (ON) شده و تایمر $T1$ برای $100\text{ms} * 30000 = 3000\text{s}$ شروع به شمارش خواهد کرد. وقتی که مقدار جاری تایمر $T1$ به مقدار تنظیمی 3000 ثانیه رسید، تایمر $T2$ برای 3000 ثانیه شروع به شمارش می‌کند. بعد از اتمام شمارش تایمر، کنتاکت باز تایمر $T2$ بسته می‌شود. سرانجام، عملکرد فرآیند پخت تکمیل شده و چراغ آلام $Y0$ روشن می‌شود. زمان کل فرآیند پخت برابر با $3000\text{s} + 3000\text{s} + 3000\text{s} = 9000\text{s} = 150\text{min}$ می‌باشد. 2.5h

توسط دستورالعمل HOUR می‌توان تابع توسعه زمان را به راحتی اجرا کرد.

2- 21- مدار ستاره مثلث برای کاهش ولتاژ و جریان راه‌اندازی الکتروموتورها

در این مثال، می‌خواهیم مدار ستاره مثلث برای کاهش ولتاژ و جریان راه‌اندازی الکتروموتورهای بالای 11KW را بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

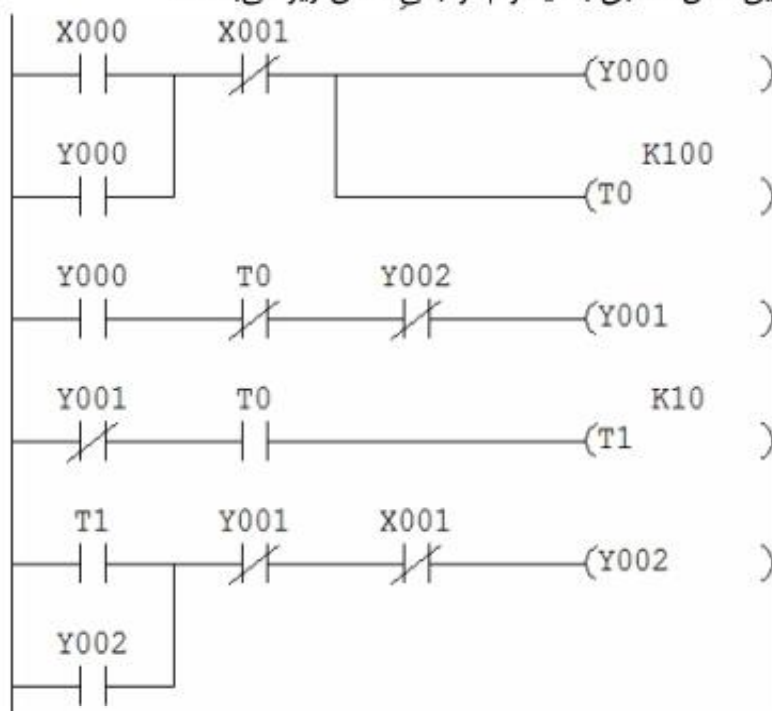
معمولاً جریان راه‌اندازی موتورهای آسنکرون سه فاز AC حدوداً 5 الی 7 برابر بزرگتر از جریان نامی می‌باشد. برای کاهش جریان راه‌اندازی و افت ولتاژ کابل‌ها، باید از مدار راه‌اندازی ستاره مثلث استفاده کرد.

فرآیند راه‌اندازی مدار ستاره مثلث به این صورت است که وقتی که شستی استارت فشرده شود، هر دو کنتاکتور اصلی و ستاره با هم فعال می‌شوند. بعد از گذشت 10 ثانیه کنتاکتور ستاره غیرفعال شده و پس از غیرفعال شدن کنتاکتور ستاره بعد از گذشت یک ثانیه کنتاکتور مثلث فعال شود. در این حالت، مدار اصلی موتور به طور نرمال برقرار شده و موتور در حالت مثلث کار خواهد کرد. هدف از کنترل در این فرآیند غیرفعال کردن کامل کنتاکتور ستاره و سپس فعال کردن کنتاکتور مثلث می‌باشد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، شستی START متصل می‌شود. با فشردن شدن شستی، این ورودی ON می‌شود.
X1	به این ورودی از PLC، شستی STOP متصل می‌شود. با فشردن شدن شستی، این ورودی ON می‌شود.
T1	تایمر 1C ثانیه، ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 1C ثانیه باید مقدار ثابت تایمر را برابر با K100 تنظیم نماییم.
T2	تایمر 1 ثانیه، ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 1 ثانیه باید مقدار ثابت تایمر را برابر با K10 تنظیم نماییم.
Y0	به این خروجی از PLC، کنتاکتور اصلی KM0 مدار راه‌انداز موتور آسنکرون به صورت ستاره مثلث متصل می‌شود.
Y1	به این خروجی از PLC، کنتاکتور ستاره KM1 مدار راه‌انداز موتور آسنکرون به صورت ستاره مثلث متصل می‌شود.
Y2	به این خروجی از PLC، کنتاکتور مثلث KM2 مدار راه‌انداز موتور آسنکرون به صورت ستاره مثلث متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

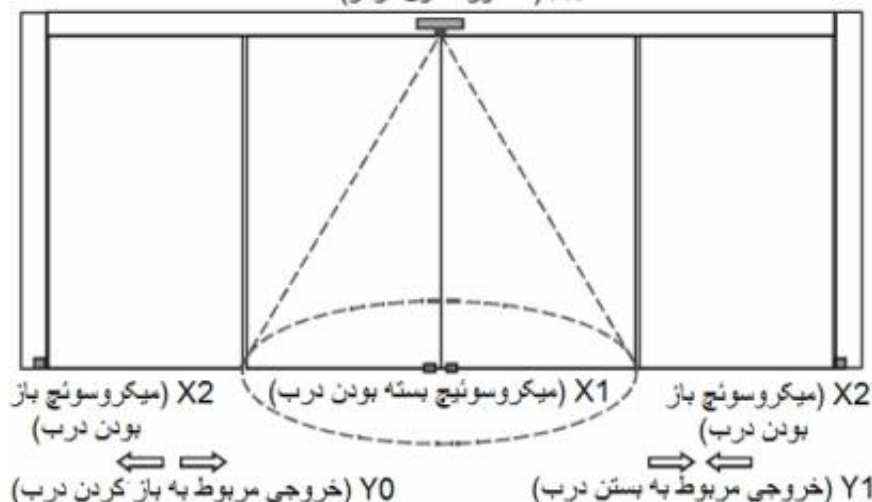
وقتی که شستی START فشرده شود، خروجی Y0 فعال شده و توسط مدار خودنگه‌دار فعال باقی خواهد ماند. در همین زمان، خروجی Y1 فعال شده و باعث فعال شدن کنتاکتور KM1 می‌شود. همزمان با فعال شدن خروجی Y0 تایمر T0 نیز برای 10 ثانیه شروع به شمارش می‌کند. بنابراین، $Y0 = ON$, $T0 = OFF$, $Y1 = ON$, $Y2 = OFF$ می‌باشند.

وقتی که مقدار جاری تایمر T0 به مقدار تنظیمی 10 ثانیه رسید، تایمر T0 فعال شده و کنتاکت‌های آن تغییر وضعیت می‌دهند که در این حالت خروجی Y1 فعال می‌شود. حال تایمر T1 برای 1 ثانیه شروع به شمارش می‌کند، بعد از گذشت 1 ثانیه $T1 = ON$ و $Y2 = ON$ می‌شوند. این به آن معنی است که کنتاکتور KM2 فعال شده است.

وقتی که شستی STOP فشرده شود، ورودی $X1 = ON$ می‌شود که در این حالت هر سه خروجی Y0، Y1 و Y2 غیرفعال شده و موتور متوقف می‌شود.

2- 22 کنترل درب اتوماتیک

در این برنامه می‌خواهیم عملکرد کنترل درب اتوماتیک را بررسی نماییم.
(سنسور مادون قرمز) X0



هدف از کنترل در این مثال به شرح زیر می‌باشد:

وقتی کسی وارد ناحیه قابل سنس توسط سنسور مادون قرمز (اینفرارد) شود، موتور باز کردن درب اتوماتیک شروع به کار کرده و درب شروع به باز شدن می‌کند. درب اتوماتیک تا نقطه‌ای باز می‌شود که میکروسوییچ باز بودن درب تحریک شود.

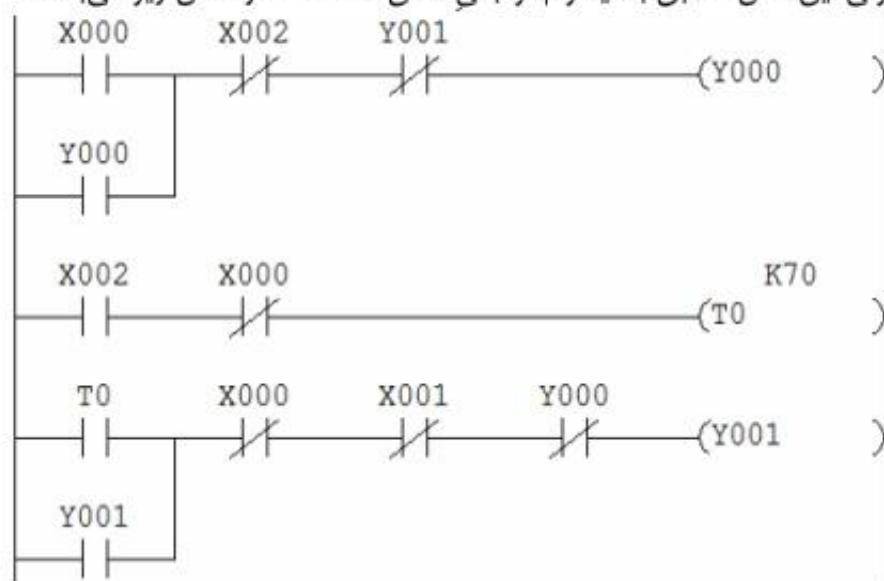
بعد از تحریک شدن میکروسوییچ باز بودن درب اتوماتیک، اگر کسی جلوی سنسور مادون قرمز نباشد، بعد از گذشت 7 ثانیه موتور بستن درب شروع به کار کرده و درب شروع به بسته شدن می‌کند. این عمل تا زمانی که میکروسوییچ بسته بودن درب اتوماتیک فعال شود، اجرا خواهد شد.

اگر کسی در هنگام بسته شدن درب جلوی سنسور مادون قرمز قرار گیرد، عمل بسته شدن درب فوراً متوقف شده و درب شروع به باز شدن می‌کند.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ایزار	عملکرد ایزار
X0	به این ورودی از PLC، سنسور مادون قرمز متصل می‌شود. اگر کسی در ناحیه قابل سنس این سنسور قرار گیرد، این ورودی فعال می‌شود.
X1	به این ورودی از PLC، میکروسوییچ بسته بودن درب متصل می‌شود. اگر هر دو میکروسوییچ بسته بودن درب با هم تحریک شوند، ورودی فعال می‌شود.
X2	به این ورودی از PLC، میکروسوییچ باز بودن درب متصل می‌شود. اگر این میکروسوییچ تحریک شود، ورودی فعال می‌شود.
T0	تایمر 7 ثانیه ضریب این تایمر 100 میلی‌ثانیه می‌باشد. برای تنظیم 7 ثانیه مقدار ثابت تایمر باید برابر با K70 تنظیم شود.
Y0	به این خروجی از PLC، موتور باز کردن درب متصل می‌شود.
Y1	به این خروجی از PLC، موتور بستن درب متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

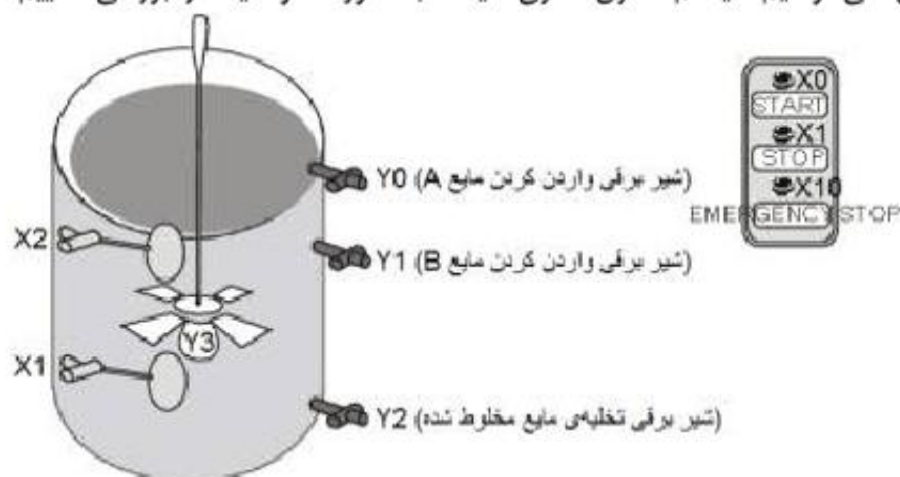
اگر کسی وارد ناحیه قابل سنس سنسور مادون قرمز شود، ورودی $X0 = ON$ می‌شود که در این حالت $Y0 = ON$ شده و فعال باقی می‌ماند. حال درب شروع به باز شدن کرده و تا زمانی که میکروسوییچ‌های متصل شده به ورودی $X0 = OFF$ شوند، در باز خواهد شد. غیرفعال شدن ورودی $X0$ باعث از کار افتادن موتور باز کردن درب می‌شود.

وقتی که میکروسوییچ باز بودن درب تحریک شد ($X2 = ON$)، تایمر $T0$ برای 7 ثانیه شروع به شمردن می‌کند و همچنین اگر بعد از گذشت 7 ثانیه کسی در ناحیه سنسور مادون قرمز نباشد ($X0 = OFF$)، خروجی $Y1 = ON$ شده و درب شروع به بستن می‌کند.

اگر کسی هنگام بسته شدن در وارد ناحیه قابل سنسور مادون قرمز شود ($X0 = ON$) کنتاکت بسته $X1$ فعال (باز) شده و از بسته شدن درب جلوگیری می‌شود ($Y1 = OFF$)، اگر $X0 = ON$ ، $X2 = OFF$ و $Y1 = OFF$ باشند، خروجی $Y0$ فعال (ON) شده و درب مجدداً شروع به باز شدن می‌کند.

2- 23 کنترل همزن مایعات به صورت اتوماتیک

در این مثال، می‌خواهیم سیستم کنترل همزن مایعات به صورت اتوماتیک را بررسی نماییم.



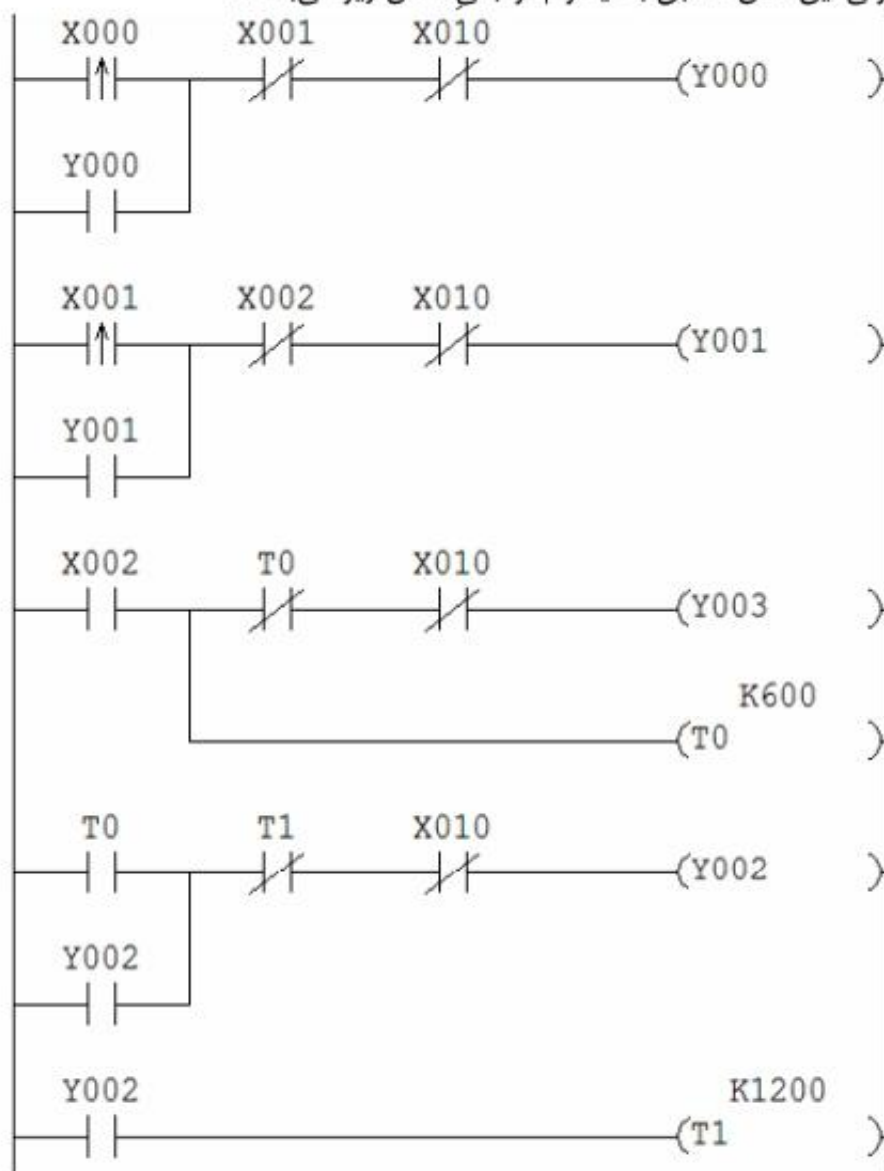
هدف از کنترل در این مثال به شرح زیر می‌باشد:

وقتی که دکمه **START** فشرده شود، دو مایع **A** و **B** به صورت اتوماتیک داخل مخزن همزن ریخته شوند. وقتی که مقدار مایعات ریخته شده در داخل مخزن به حد تنظیمی رسید، میکسر هر دو مایع را هم زده و سپس بعد از تکمیل فرآیند مخلوط کردن، شیر برقی تخلیه‌ی مخزن باز شده و مواد داخل تانکر خالی شود.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، شستی START متصل می‌شود. وقتی که این شستی فشرده شود ورودی فعال (ON) می‌شود.
X1	به این ورودی از PLC، سنسور لول سنج شناور سطح پایین متصل می‌شود. وقتی که سطح مایع داخل مخزن به سنسور رسید، ورودی $X1 = ON$ می‌شود.
X2	به این ورودی از PLC، سنسور لول سنج شناور سطح بالا متصل می‌شود. وقتی که سطح مایع داخل مخزن به سنسور رسید، ورودی $X2 = ON$ می‌شود.
X10	به این ورودی از PLC، دکمه امرجنسی استوپ متصل می‌شود. وقتی که این شستی فشرده شود ورودی فعال (ON) می‌شود.
T0	تایمر 60 ثانیه. ضریب این تایمر 100 میلی‌ثانیه می‌باشد. برای تنظیم 60 ثانیه مقدار ثابت تایمر باید برابر با K600 تنظیم شود.
T1	تایمر 120 ثانیه. ضریب این تایمر 100 میلی‌ثانیه می‌باشد. برای تنظیم 120 ثانیه مقدار ثابت تایمر باید برابر با K1200 تنظیم شود.
Y0	به این خروجی از PLC، شیر برقی ورود مایع A به داخل مخزن متصل می‌شود.
Y1	به این خروجی از PLC، شیر برقی ورود مایع B به داخل مخزن متصل می‌شود.
Y2	به این خروجی از PLC، شیر برقی تخلیه مایع داخل مخزن متصل می‌شود.
Y3	به این خروجی از PLC، الکتروموتور میکسر (همزن) متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که شستی START فشرده شود ورودی $X0 = ON$ می‌شود. خروجی $Y0$ فعال شده و فعال باقی خواهد ماند. در این حالت شیر برقی مایع A باز شده و تا زمانی که سطح مایع داخل مخزن به سنسور لول سنج شناور سطح پایین برسد، باز خواهد ماند. وقتی که سطح مایع داخل مخزن به سنسور لول سنج سطح پایین رسید، ورودی $X1 = ON$ می‌شود. خروجی $Y1$ فعال شده و فعال باقی خواهد ماند. در این حالت، شیر برقی مایع B باز شده و تا زمانی که سطح مایع داخل مخزن به سنسور لول سنج شناور سطح بالا برسد، شیر برقی باز خواهد ماند.

2- 24 دستگاه قهوه ساز اتوماتیک

The diagram shows a robotic arm with the following components and labels:

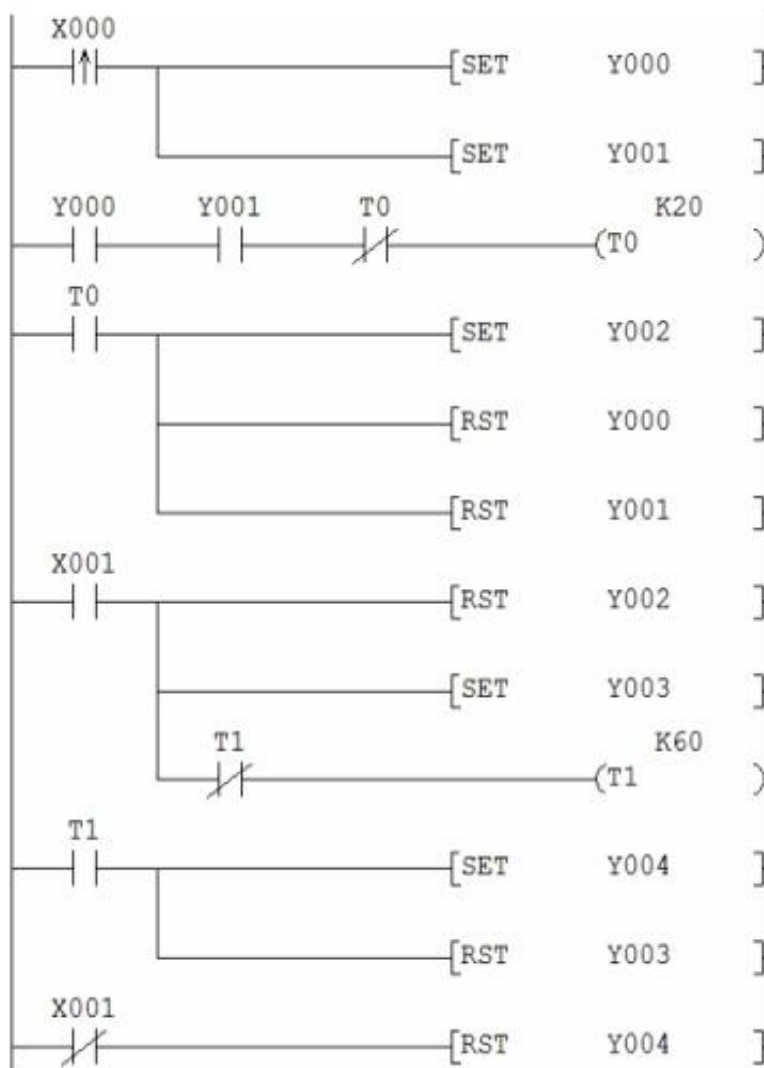
- قہوہ** (Coffee) - Y1
- آب جوش** (Boiling Water) - Y2
- محل انداختن سکه** (Coin dropping place) - X0
- سنسور آشکار ساز سکه** (Coin detector sensor)
- مخزن مخلوط کن** (Mixer container) - X1
- موتور میکسر** (Mixer motor) - Y3
- محل خروج لیوان کاغذی** (Paper cup exit place) - Y0
- +** Y4 (Output to the cup)

وقتی که یک سکه داخل دستگاه انداخته شود یک لیوان کاغذی از دستگاه بیرون می‌آید. در این هنگام مخزن مخلوط کن دستگاه شروع به کار می‌کند. بعد از گذشت 2 ثانیه، به مدت 60 ثانیه آب و قهوه وارد مخزن مخلوط کن شده و درنهایت قهوه‌ی آماده شده از شیر تخلیه به داخل لیوان کاغذی ریخته شود.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند :

ابزار	عملکرد ابزار
X0	این ورودی PLC به سنسور آشکارساز سکه متصل شده است. وقتی سکه داخل دستگاه انداخته شود، این ورودی ON می‌شود.
X1	این ورودی PLC به سنسور فشار متصل می‌شود. وقتی مایع داخل مخزن به یک فشار معین رسید، این ورودی ON می‌شود.
T0	تایمر 2 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 2 ثانیه مقدار ثابت تایمر را باید برابر با K20 تنظیم نماییم.
T1	تایمر 60 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 60 ثانیه مقدار ثابت تایمر باید برابر با K600 تنظیم شود.
Y0	خروجی انداختن لیوان کاغذی به بیرون از دستگاه
Y1	خروجی تخلیه‌ی قهوه‌ی خام به داخل مخزن مخلوط کن
Y2	خروجی تخلیه‌ی آب گرم به داخل مخزن مخلوط کن
Y3	موتور میکسر
Y4	به این خروجی از PLC، شیر برقی تخلیه قهوه‌ی آماده متصل می‌شود

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد :



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که یک سکه وارد دستگاه شود، ورودی $X1 = ON$ می‌شود. خروجی‌های $Y0$ و $Y1$ فعال شده و فعال باقی خواهند ماند. یک لیوان کاغذی از دستگاه بیرون می‌آید و در این زمان یک مقدار معینی قهوه خام داخل مخزن همزن ریخته می‌شود.

مطابق با مقدار تنظیمی تایمر $T0$ خروجی $Y0$ و $Y1$ برای 2 ثانیه فعال می‌ماند. وقتی که کنتاکت باز تایمر $T0$ فعال (بسته) شود، خروجی $Y2$ فعال شده و آب گرم وارد مخزن همزن می‌شود. در این زمان، هر دو خروجی لیوان کاغذی و قهوه خام بسته می‌شوند.

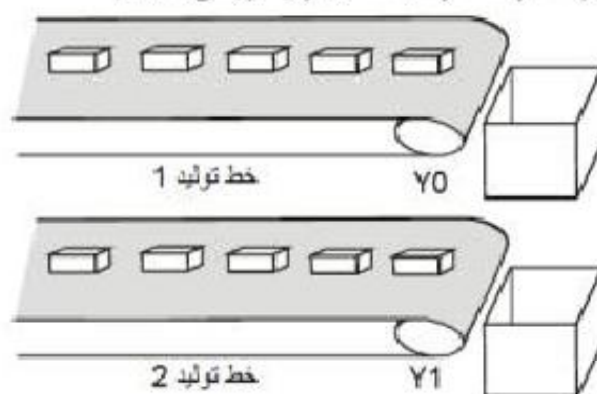
وقتی که مایع داخل مخزن به یک فشار معین شده رسید، ورودی $X1=ON$ می‌شود. از این رو، خروجی $Y0$ (آب داغ) ریست شده و خروجی $Y3$ (میکسر) به مدت 60 ثانیه فعال می‌شود. بعد از گذشت 60 ثانیه، کنتاکت باز تایمر $T1$ فعال (بسته) خواهد شد. خروجی $Y4$ فعال شده و فعال باقی

خواهد ماند و در همین زمان خروجی $Y3$ ریست می‌شود. همزن متوقف شده و قهوه‌ی آماده از طریق شیر تخلیه به داخل لیوان کاغذی ریخته می‌شود.

وقتی که عمل تخلیه قهوه آماده به داخل لیوان تکمیل شد، ورودی $X1 = OFF$ شده و خروجی $Y4$ ریست خواهد شد. در این زمان، شیر تخلیه قهوه‌ی آماده بسته می‌شود.

2- 25 برنامه وقفه اتوماتیک تایمر

در این مثال، می‌خواهیم برنامه وقفه اتوماتیک تایمر را بررسی نماییم.

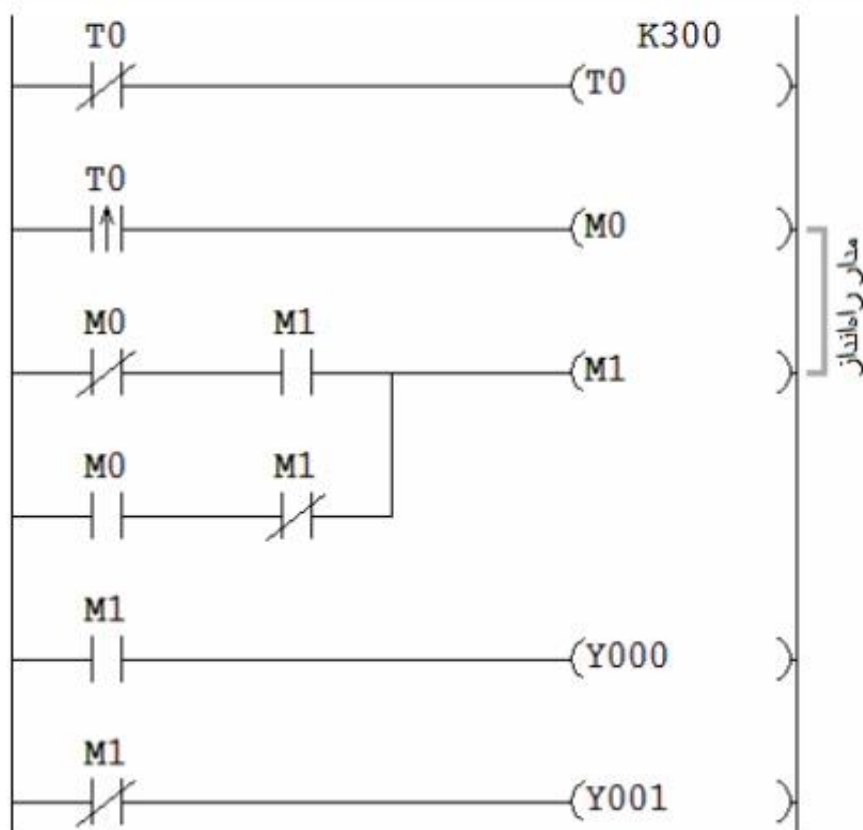


هدف از کنترل در این مثال به شرح زیر می‌باشد:

در خط‌های تولید مونتاژ شده توسط PLC، یک اپراتور باید محصول را بر روی دو کانوایر در داخل جعبه‌های روی کانوایر قرار دهد. برای اطمینان از بسته‌بندی درست، باید زمان بسته‌بندی به اندازه‌ی کافی باشد. برنامه‌ی کنترل دو کانوایر برای راه‌اندازی به ترتیب شامل: بعد از 30 ثانیه کار کردن کانوایر، کانوایر باید متوقف شود و سپس کانوایر دو به اندازه‌ی 30 ثانیه شروع به کار کند. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
T0	تایمر 30 ثانیه، ضرب این تایمر 100 میلی‌ثانیه می‌باشد. برای تنظیم زمان 30 ثانیه برای تایمر باید مقدار ثابت تایمر را برابر تنظیم کرد. K300
M0	از این ابزار برای کنترل کردن مدار راه‌انداز استفاده شده است.
M1	از این ابزار برای ترتیب روشن کردن کانوایرها استفاده شده است.
Y0	به این خروجی از PLC، کنتاکتور موتور کانوایر شماره 1 خط تولید متصل می‌شود.
Y1	به این خروجی از PLC، کنتاکتور موتور کانوایر شماره 2 خط تولید متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

در این برنامه از کنتاکت بسته‌ی تایمر T0 برای شرط اجرای خود تایمر T0 استفاده شده است. وقتی که مقدار جاری تایمر T0 به مقدار تنظیمی 30 ثانیه رسید، خروجی تایمر فعال خواهد شد. در این حالت، مدار راه‌انداز اجرا شده و وضعیت ابزار M1 تغییر می‌کند که باعث شروع به کار کردن کانوایر خط 1 می‌شود.

بعد از گذشت 30 ثانیه، تایمر T0 فعال می‌شود و کنتاکت بسته‌ی تایمر T0 فعال (باز) خواهد شد. در این هنگام، تایمر T0 غیرفعال خواهد شد و سپس مجدداً کنتاکت بسته‌ی تایمر T0 غیرفعال (بسته) خواهد شد. در دوره‌ی اسکن بعدی، کنتاکت تایمر T0 بسته است و تایمر مجدداً شروع به شمارش خواهد کرد. بعد از گذشت 30 ثانیه، تایمر T0 فعال شده، بنابراین مدار راه‌انداز اجرا خواهد شد. از این رو، ابزار M1 مجدداً تغییر وضعیت خواهد داد. در این هنگام کانوایر خط 1 متوقف شده و کانوایر خط 2 شروع به کار می‌کند.

با استفاده از مدار راه‌انداز می‌توان خروجی Y0 و Y1 را به ترتیب فعال و غیرفعال کرد که باعث کار کردن دو کانوایر به ترتیب با یک فاصله زمانی معین به صورت دوره‌ای خواهد شد.

2- 26- مقایسه زمان

از دستورالعمل TZCP برای مقایسه‌ی زمان با یک محدوده‌ی معین استفاده می‌شود. این دستورالعمل دارای 4 عملوند است. در عملوند S1 محدوده‌ی پایین قابل تنظیم را وارد می‌کنیم. در عملوند S2 محدوده‌ی بالای زمان قابل تنظیم را وارد می‌کنیم. و به عملوند S زمان جاری سیستم را اختصاص می‌دهیم. نتیجه‌ی بدست آمده از عمل مقایسه در عملوند D ذخیره می‌شود.

در عملوند S1 محدوده‌ی پایین ساعت، در عملوند S1+1 محدوده‌ی پایین دقیقه و در عملوند S1+2 محدوده‌ی پایین ثانیه برای مقایسه در یک محدوده‌ی خاص تنظیم می‌شود.

در عملوند S2 محدوده‌ی بالای ساعت، در عملوند S2+1 محدوده‌ی بالای دقیقه و در عملوند S2+2 محدوده‌ی بالای ثانیه در یک محدوده‌ی خاص تنظیم می‌شود.

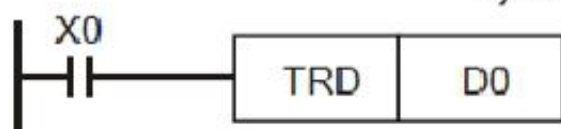
در عملوند S مقدار ساعت جاری سیستم، در عملوند S+1 دقیقه‌ی جاری سیستم و در عملوند S+2 ثانیه جاری سیستم قرار می‌گیرد.

زمان جاری سیستم را ابتدا باید با دستورالعمل TRD خوانده و سپس در دیتا رجیسترها ذخیره کرد. سپس دیتا رجیسترهای مربوطه را به عملوند S دستورالعمل TZCP اختصاص داد.

دقت داشته باشید که در عملوند S1 باید محدوده‌ی پایین تنظیم شود. اگر در عملوند S2 مقدار زمان کمتر از مقدار زمان تنظیم شده در عملوند S1 تنظیم شود، عمل مقایسه فقط با مقدار تنظیم شده در عملوند S2 صورت خواهد گرفت. این به آن معنی است که مقایسه‌ی محدوده‌ی غیرفعال خواهد شد. اگر مقدار تنظیم شده در عملوندهای S, S2, S1 از مقدار مجاز زمان تجاوز کند، یک خطا در سیستم رخ خواهد داد. در همین زمان فلگ خاص M1068=ON خواهد شد. وقتی که مقدار جاری سیستم که در عملوند S قرار دارد کمتر از مقدار اختصاص داده شده به عملوند S1 بود و همچنین کمتر از مقدار اختصاص داده شده به عملوند S2 بود، عملوند D فعال (ON) خواهد شد.

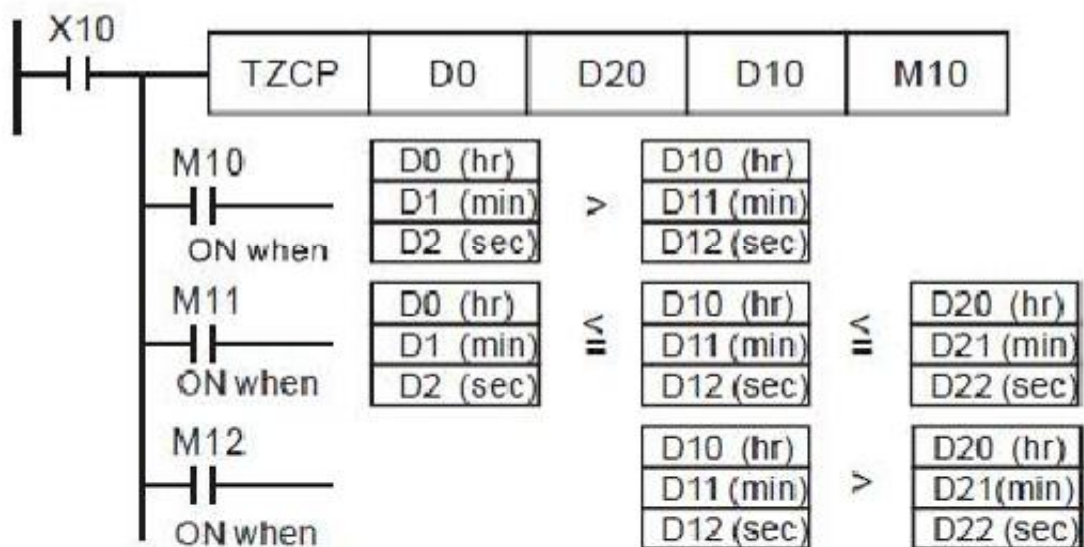
وقتی که مقدار جاری سیستم که در عملوند S قرار دارد بیشتر از مقدار اختصاص داده شده به عملوند S1 و S2 باشد، عملوند D+2 فعال خواهد شد. در دیگر شرایط عملوند D+1 فعال خواهد بود.

بعنوان مثال، وقتی که ورودی X0=ON شود، زمان تقویم جاری از رجیسترهای خاص D1063~D1069 خوانده شده و در دیتا رجیسترهای D0~D6 ذخیره می‌شود. برای محاسبات زمان توسط دستورالعمل‌های TCMP, TZCP, TADD, TSUB می‌توان از دیتا رجیسترهای D4~D6 در این مثال استفاده کرد.



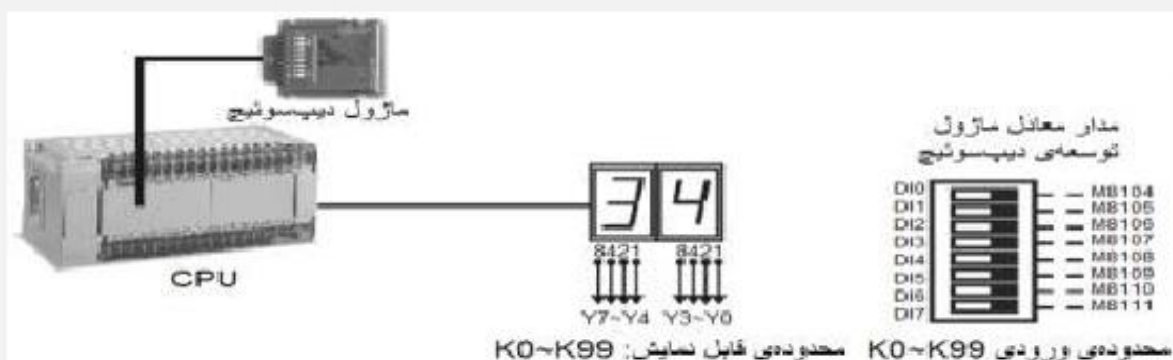
Special D	Item	Content		General D	Item
D1063	Year (Western)	00~99	→	D0	Year (Western)
D1064	Weeks	1~7	→	D1	Weeks
D1065	Month	1~12	→	D2	Month
D1066	Day	1~31	→	D3	Day
D1067	Hour	0~23	→	D4	Hour
D1068	Minute	0~59	→	D5	Minute
D1069	Second	0~59	→	D6	Second

وقتی که $X10=ON$ شود، دستورالعمل TZCP اجرا شده و یکی از فلگ‌های M10~M12 فعال خواهد شد. وقتی که $X10=OFF$ شود، دستورالعمل TZCP اجرا نخواهد شد و وضعیت فلگ‌های M10~M12 بدون تغییر باقی خواهد ماند.



فصل 3

دستورات پیشرفته دیجیتال



3- 1- اعلام آتش سوزی

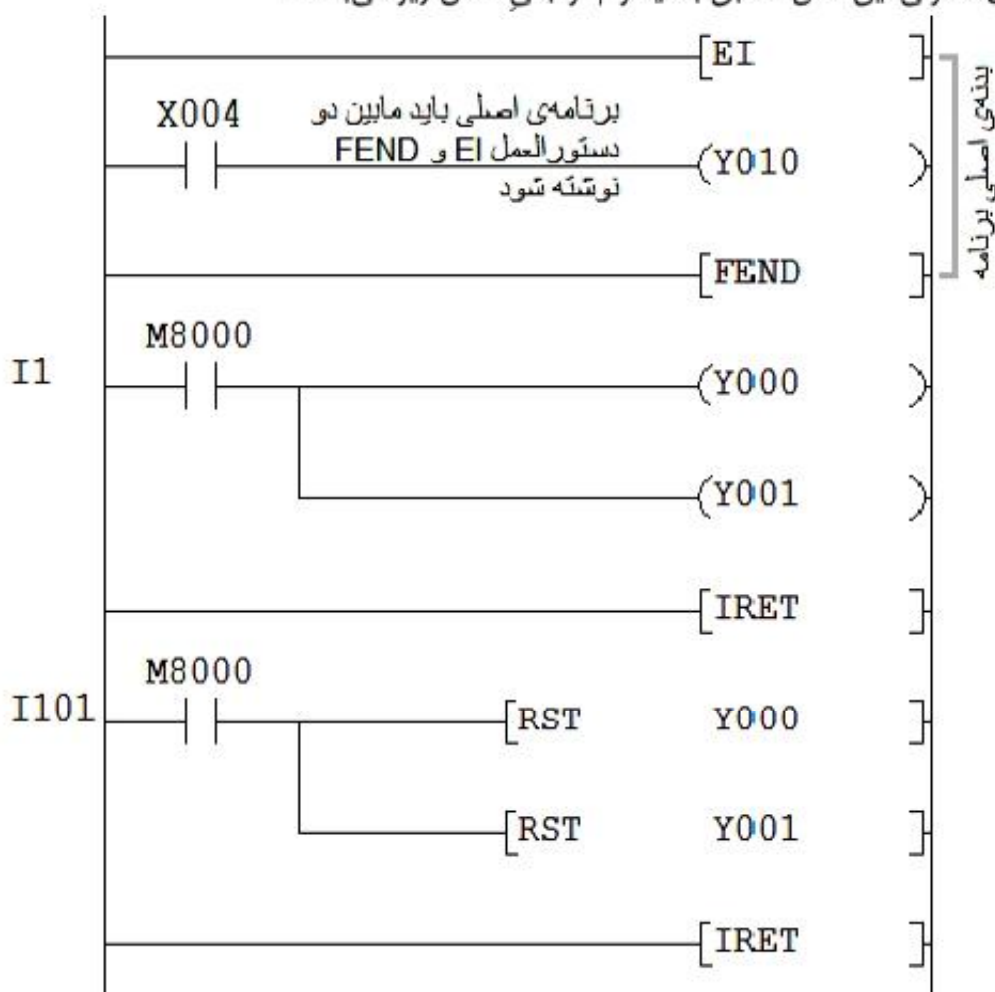
در این مثال، می‌خواهیم برنامه آلام آتش سوزی در دفتر کار (کاربرد وقفه در برنامه) را بررسی نماییم. هدف از کنترل در این مثال به شرح زیر می‌باشد:

وقتی که سنسور دما، دمایی بالاتر از دمای تنظیم شده را سنس کرد، یک آلام و شیر برقی اسپری آب فعال شود.

وقتی که شستی ریست فشرده شد هم آلام و هم شیر برقی اسپری آب خاموش شوند. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، سنسور دما (ترموستات) متصل می‌شود. وقتی که دما بالاتر از رنج تنظیم شده رفت، ورودی ON می‌شود. این ورودی یک وقفه سخت‌افزاری با آدرس I1 می‌باشد. با فعال شدن ورودی در صورت نوشتن برنامه برای وقفه، اجرای برنامه اصلی رها شده و برنامه‌ی وقفه اجرا می‌شود. در صورت نوشتن برنامه‌ی وقفه از ورودی X0 در بدنه‌ی برنامه‌ی اصلی استفاده نکنید.
X1	به این ورودی از PLC، شستی RESET متصل می‌شود. وقتی که شستی فشرده شود، ورودی ON می‌شود. این ورودی یک وقفه سخت‌افزاری با آدرس I101 می‌باشد. با فعال شدن ورودی در صورت نوشتن برنامه برای وقفه، اجرای برنامه اصلی رها شده و برنامه‌ی وقفه اجرا می‌شود. در صورت نوشتن برنامه‌ی وقفه از ورودی X0 در بدنه‌ی برنامه‌ی اصلی استفاده نکنید.
Y0	به این خروجی از PLC، شیر برقی اسپری آب متصل می‌شود.
Y1	به این خروجی از PLC، آلام آتش سوزی متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

در این برنامه اشاره‌گر وقفه‌انداز I001 و I101 مطابق با ترمینال ورودی سخت‌افزاری X0 و X1 می‌باشند. وقتی که ورودی X0 و X1 فعال (ON) شوند، زیرروال‌های متناظر I001 و I101 اجرا خواهند شد.

اگر دمای دفتر کار نرمال باشد، ورودی $X0 = OFF$ است و آلارم دما یا همان آتش سوزی اجرا نخواهد شد. سیگنال وقفه تولید نمی‌شود و همچنین زیرروال وقفه نیز اجرا نمی‌شود.

اگر دمای دفتر کار بالاتر از دمای تنظیم شده‌ی ترموستات باشد، ورودی $X0 = ON$ شده و آلارم آتش سوزی فعال می‌شود. در این هنگام PLC کار نرمال خود را رها کرده (اسکن ترتیبی برنامه اصلی توسط CPU متوقف می‌شود) و زیرروال I001 را مطابق با وقفه‌ی ایجاد شده اجرا خواهد کرد. از این رو، شیر برقی اسپری آب Y0 و آلارم آتش سوزی Y1 هر دو فعال می‌شوند. بعد از اجرای زیرروال

1001، برنامه‌ی اصلی مجدداً شروع به اجرا کردن و اسکن ترتیبی برنامه اصلی توسط CPU از نقطه‌ای که اسکن برنامه متوقف شده بود اجرا خواهد شد.

برای ریست کردن آلارم باید دکمه‌ی ریست آلارم را فشار دهید. با فشردن دکمه‌ی ریست ورودی $X1 = ON$ شده مجدداً PLC روند اجرای برنامه‌ی اصلی را متوقف کرده و زیرروال I101 وقفه‌ی ایجاد شده را اجرا خواهد کرد. از این رو، هر دو خروجی آلارم $Y1$ و شیر برقی اسپری آب $Y0$ خاموش می‌شوند. بعد از اجرای زیرروال I101، PLC به برنامه‌ی اصلی برگشته و برنامه از جایی که قبل از اجرای زیرروال متوقف شده بود، شروع به اجرا می‌کند.

3-2 مدار آخرین اولویت

در این مثال قصد داریم مدار آخرین اولویت را بررسی نماییم.

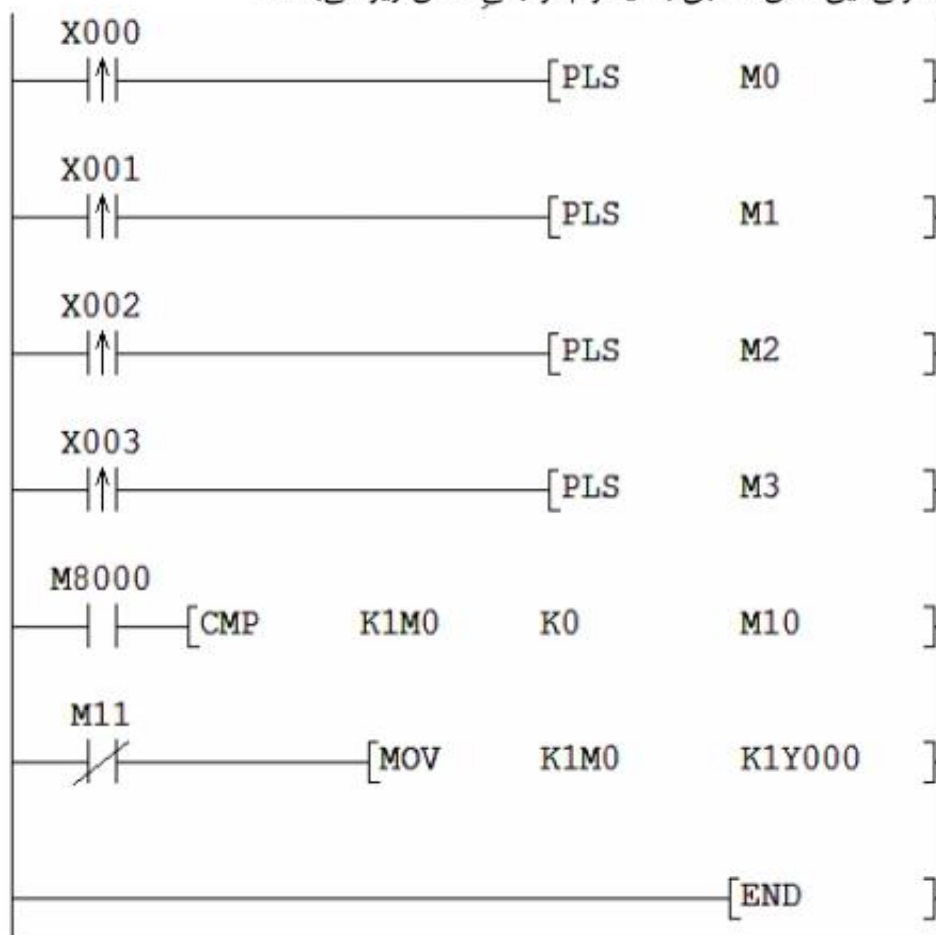
هدف از کنترل در این مدار به شرح زیر می‌باشد:

در این مدار چهار شستی و چهار لامپ نمایشگر وجود دارد. در این مدار با فشردن شستی، نمایشگر مربوطه روشن شده و با رها کردن شستی لامپ نمایشگر همچنان روشن باقی خواهد ماند.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC شستی شماره‌ی 1 متصل می‌شود. وقتی که شستی فشرده شود کنتاکت آن از OFF به ON تغییر وضعیت خواهد داد
X1	به این ورودی PLC دومین شستی متصل می‌شود. وقتی که شستی فشرده شود کنتاکت آن از OFF به ON تغییر وضعیت خواهد داد
X2	به این ورودی PLC سومین شستی متصل می‌شود. وقتی که شستی فشرده شود کنتاکت آن از OFF به ON تغییر وضعیت خواهد داد
X3	به این ورودی PLC چهارمین شستی متصل می‌شود. وقتی که شستی فشرده شود کنتاکت آن از OFF به ON تغییر وضعیت خواهد داد
Y0	به این خروجی PLC، اولین لامپ نمایشگر متصل می‌شود
Y1	به این خروجی PLC، دومین لامپ نمایشگر متصل می‌شود
Y2	به این خروجی PLC، سومین لامپ نمایشگر متصل می‌شود
Y3	به این خروجی PLC، چهارمین لامپ نمایشگر متصل می‌شود

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



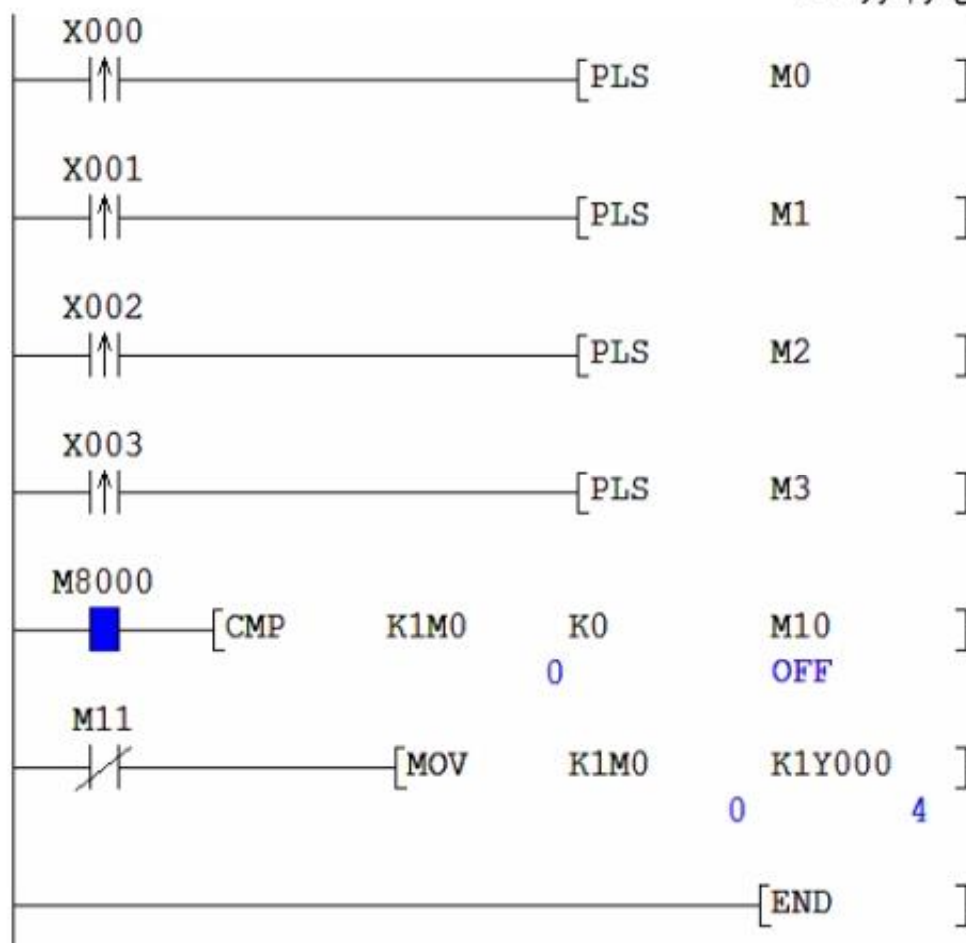
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که یکی از شستی‌ها فشرده شود، ابزار X مربوطه فعال (ON) می‌شود. در این سیکل اسکن برنامه، دستورالعمل PLS اجرا می‌شود و رله داخلی M مربوطه فعال خواهد شد. سپس دستورالعمل CMP اجرا شده و نتیجه با $K1M0 > 0$ مقایسه خواهد شد که در این حالت ابزار $M10 = ON$ می‌شود، اما $M11 = OFF$ خواهد شد. سپس دستورالعمل $[MOV K1M0 K1Y0]$ اجرا خواهد شد و وضعیت M مطابق با خروجی Y به خروجی ارسال می‌شود. در این هنگام، لامپ نمایشگر متصل شده به خروجی سخت‌افزاری (Y) که ON بوده خاموش می‌شود.

وقتی که دومین سیکل اسکن برنامه اجرا شود، دستورالعمل PLS اجرا نخواهد شد و مقدار M0 الی M3 برابر 0 خواهند شد. از این رو، دستورالعمل CMP اجرا شده و $M11$ فعال (ON) خواهد شد که در این حالت $K1M0 = 0$ می‌شود. دستورالعمل $[MOV K1M0 K1Y0]$ اجرا نخواهد شد و

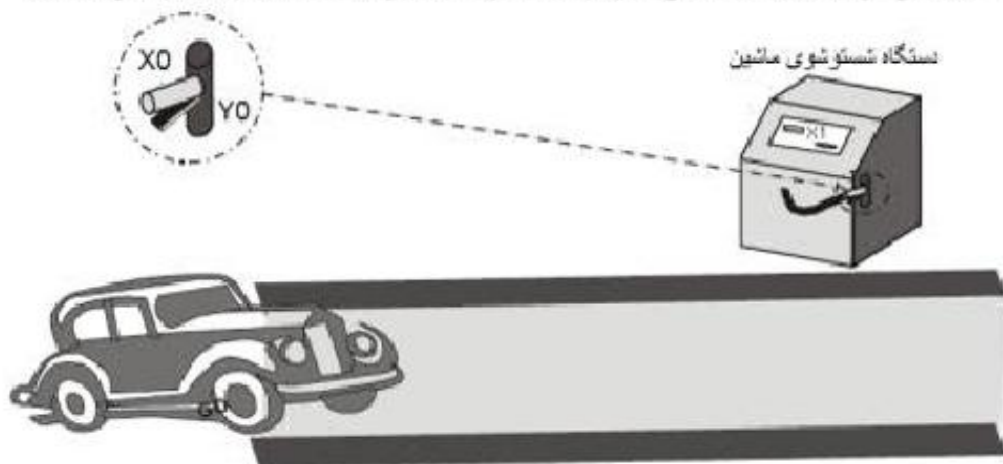
وضعیت 0 یا غیرفعال (OFF) ابزار M به خروجی ارسال نخواهد شد. از این رو، خروجی Y تا زمان فشردن یک شستی دیگر در وضعیت قبلی خود باقی خواهد ماند.

مدار شکل بالا مربوط به زمان آفلاین بودن نرم‌افزار برنامه‌نویسی با PLC می‌باشد. اما مدار نشان داده شده در شکل پایین مربوط به زمان آنلاین بودن نرم‌افزار برنامه‌نویسی با PLC و همچنین در مد مانیتور بودن نرم‌افزار است.



3- برنامه کارواش

در این مثال، می‌خواهیم برنامه‌ی تابع اجرای شستشوی ماشین را با تایمر نرمال بررسی نماییم.

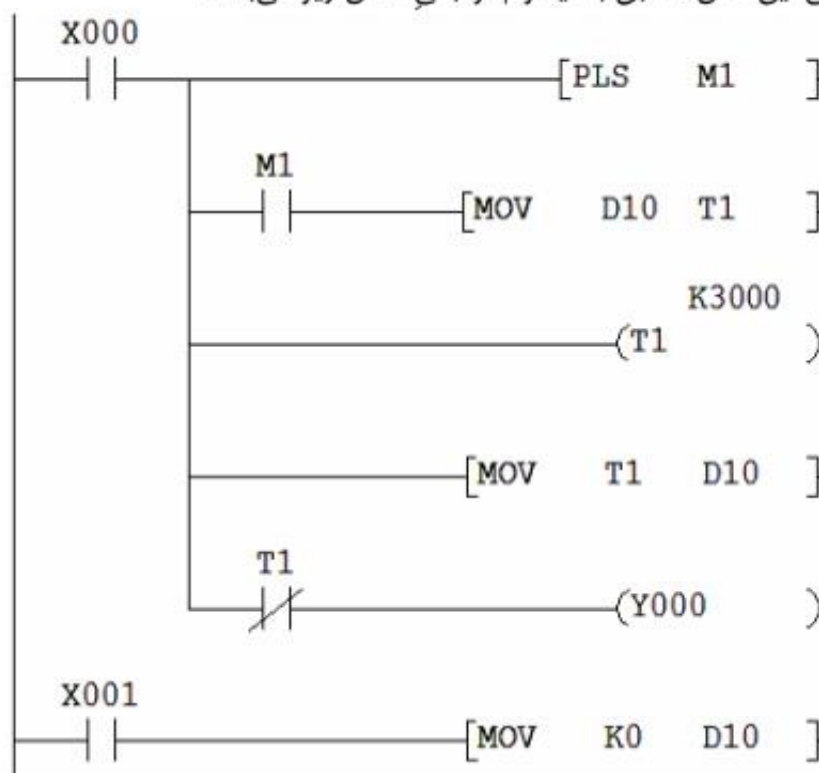


هدف از کنترل در این مثال به شرح زیر می‌باشد:

بعد از وارد شدن ماشین به کارواش، با فعال کردن ورودی $X0$ دستگاه کارواش به مدت 5 دقیقه شروع به شستن ماشین کرده و بعد از گذشت 5 دقیقه شیر برقی اسپری آب و صابون قطع می‌شود. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ایزار	عملکرد ایزار
$X0$	به این ورودی از PLC، سوئیچ فعال کردن شیر برقی اسپری متصل می‌شود. وقتی که دستگیره‌ی اسپری محکم نگهداشته شود، ورودی $X0$ فعال می‌شود.
$X1$	به این ورودی PLC، سنسور آشکارساز سکه متصل می‌شود. وقتی که سکه وارد دستگاه شود، این ورودی فعال می‌شود.
$M1$	این ایزار راه‌انداز یک پالس برای یک سیکل اسکن برنامه می‌باشد.
$T1$	تایمر. ضریب این تایمر 100 میلی‌ثانیه است.
$D10$	از این ایزار برای ذخیره سازی یک مقدار ثابت برای تنظیم زمان تایمر استفاده می‌شود. برای تنظیم 5 دقیقه باید مقدار ثابت K3000 را به این ایزار اختصاص دهیم.
$Y0$	به این خروجی از PLC، شیر برقی اسپری متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که مشتری سکه را وارد دستگاه کند، ورودی $X1 = ON$ می‌شود. مقدار زمان تایمر در ابزار $D10$ پاک خواهد شد.

وقتی که مشتری دستگاهی اسپری را فشار دهد، ورودی $X0 = ON$ خواهد شد. دستورالعمل PLS اجرا شده و ابزار $M1$ برای یک سیکل اسکن برنامه فعال (ON) می‌شود. در این هنگام تایمر از 0 تا 5 دقیقه ($T1 = K3000$) شروع به شمارش می‌کند. با فعال شدن تایمر خروجی $Y0 = ON$ شده و شیر برقی اسپری آب و صابون باز می‌شود.

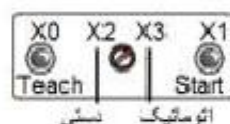
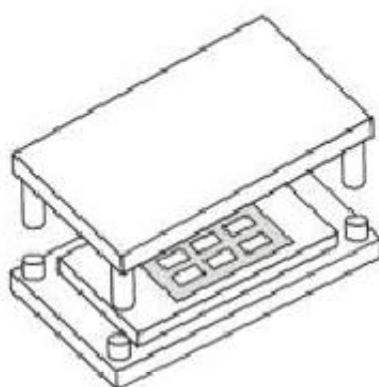
اگر دستگاهی شیر برقی اسپری رها شود، شمارش تایمر متوقف خواهد شد. در این حالت، مقدار جاری زمان سپری شده در تایمر ذخیره شده و در عملکرد شیر برقی اسپری وقفه خواهد افتاد.

وقتی که مشتری مجدداً دستگاهی شیر برقی را فشار دهد، تایمر از مقدار ذخیره شده در ابزار $D10$ شروع به شمارش ادامه‌ی زمان سپری شده می‌کند. وقتی که تایمر $T1$ در حال کار باشد، مقدار جاری تایمر $T1$ به ابزار $D10$ برای ذخیره سازی مقدار سپری شده ارسال می‌شود. وقتی که تایمر، قبل از اتمام زمان تنظیم شده غیرفعال شود و اگر مجدداً تایمر $T1$ فعال شود، مقدار ذخیره شده در ابزار $D10$ به تایمر $T1$ ارسال شده و تایمر از ادامه زمان سپری شده قبل از غیرفعال شدنش شروع به کار می‌کند.

با این عمل، حتی اگر گاهی اوقات وقفه‌ای در عملکرد شیر برقی برای فرآیند شستشوی ماشین به وجود آید، به مشتری این اطمینان داده می‌شود که سرویس شستشوی ماشین به مدت 5 دقیقه‌ی کامل خواهد بود.

3-4 برنامه تابع آموزش به دستگاه

در این مثال، می‌خواهیم تابع آموزش به دستگاه برای هوشمندسازی دستگاه را با استفاده از تایمر نرمال بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

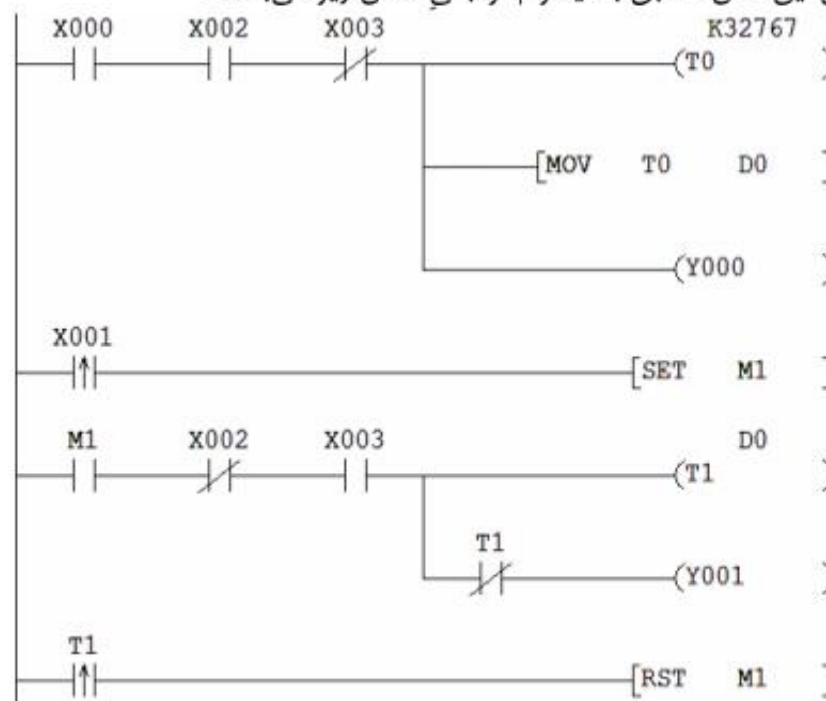
در مد دستی، مهندسین باید زمان چاپ را مطابق با تجربه خودشان تنظیم نمایند. زمان چاپ مطابق با زمان آموزش داده شده به دستگاه می‌باشد.

در مد اتوماتیک، اگر دکمه‌ی **START** فشرده شود، ماشین فرآیند چاپ را مطابق با مقدار زمان ذخیره شده توسط فرآیند آموزش داده شده به دستگاه اجرا خواهد کرد. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC، دکمه آموزش دادن به دستگاه (Teach) متصل می‌شود. وقتی که این شستی فشرده شود، این ورودی فعال (ON) می‌شود.
X1	به این ورودی از PLC، شستی START متصل می‌شود. وقتی که این شستی فشرده شود، این ورودی فعال (ON) می‌شود.
X2	این ورودی از PLC به کنتاکت مربوط به مد دستی Manual mode از سوئیچ‌گردان قفل‌شونده‌ی دو وضعیتی متصل می‌شود.
X3	به این ورودی کنتاکت مربوط به مد اتوماتیک Auto mode از سوئیچ‌گردان قفل‌شونده‌ی دو وضعیتی متصل می‌شود.
M1	از این ابزار برای راه‌اندازی START در مد اتوماتیک استفاده می‌شود.
T0	تایمر. ضریب این تایمر 100 میلی‌ثانیه می‌باشد.
T1	تایمر. ضریب این تایمر 100 میلی‌ثانیه می‌باشد.
D0	رجیستر داده. از این ابزار برای ذخیره‌سازی مقدار زمان چاپ استفاده می‌شود.
Y0	وقتی که دکمه Teach فشرده شود، دستگاه چاپ برای آموزش مقدار زمان تنظیمی شروع به کار می‌کند.

Y1 وقتی که دکمه START در مد اتوماتیک فشرده شود، دستگاه چاپ شروع به کار برای فرآیند چاپ می‌کند.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

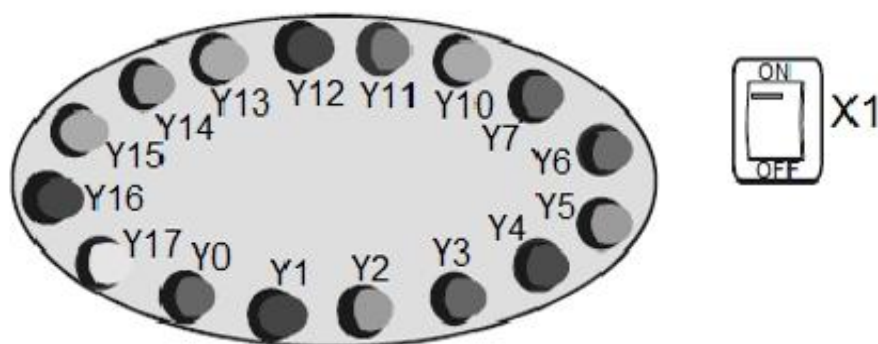
وقتی که سوئیچ دو وضعیت مد دستی قرار گیرد، ورودی $X2 = ON$ می‌شود. در این هنگام، با فشردن شستی Teach و نگهداشتن آن ورودی $X0 = ON$ می‌شود. از این رو، بوبین $Y0$ فعال شده و فرآیند چاپ شروع می‌شود، و همزمان با فرآیند چاپ دستورالعمل تایمر $T0$ اجرا شده و مقدار جاری تایمر به ابزار $D0$ ارسال می‌شود. وقتی که فرآیند چاپ تمام شد با برداشتن انگشت از روی شستی Teach خروجی $Y0$ غیرفعال شده و فرآیند چاپ متوقف می‌شود.

وقتی که سوئیچ دو وضعیت مد اتوماتیک قرار گیرد، ورودی $X3 = ON$ می‌شود. هر وقت که شستی START فشرده شود، ورودی $X1 = ON$ شده و خروجی $Y1 = ON$ خواهد شد که در این حالت فرآیند چاپ اجرا می‌شود. در این هنگام، تایمر $T1$ شروع به شمارش کرده و تا زمانی که مقدار جاری این تایمر به مقدار ذخیره شده در تایمر $T0$ برسد، فعال خواهد بود. وقتی که زمان چاپ تکمیل شد، کنتاکت بسته‌ی تایمر $T1$ و لبه‌ی بالارونده تایمر $T1$ فعال شده که در این حالت هر دو ابزار $M1$ و $Y1$ غیرفعال می‌شوند. همین امر باعث متوقف شدن فرآیند چاپ می‌شود. وقتی که شستی START مجدداً فشرده شود، ابزار $M1$ فعال شده و فرآیند چاپ مجدداً تکرار می‌شود.

تابع آموختن به تایمر را نیز می‌توان توسط دستورالعمل TTMR اجرا کرد.

3-5 کنترل چراغ چشمک‌زن رنگی

در این مثال، می‌خواهیم برنامه چراغ چشمک‌زن رنگی را با استفاده از دستورالعمل CML بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

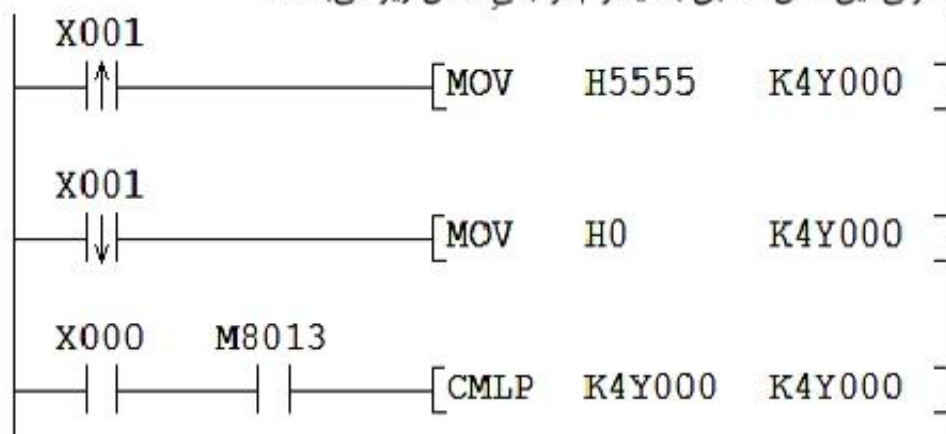
وقتی که سوئیچ کنترل چشمک‌زن فعال شد، چراغ‌های سیگنال فرد و چراغ‌های سیگنال زوج به صورت متناوب و هر 1 ثانیه یک بار روشن شوند.

وقتی که ورودی $X1 = OFF$ شد تمام چراغ‌های سیگنال خاموش می‌شوند.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

عملکرد ابزار	ابزار
به این ورودی از PLC، سوئیچ کنترل چراغ چشمک‌زن متصل می‌شود. وقتی که سوئیچ در وضعیت ON قرار گیرد، ورودی فعال (ON) می‌شود.	X1
ابزار کلاک پالس 1S، این کنتاکت 0.5 ثانیه فعال (ON) بوده و 0.5 ثانیه غیرفعال (OFF) می‌باشد.	M8013
به این خروجی‌های PLC، 16 عدد چراغ سیگنال رنگی متصل می‌شود.	Y0-Y17

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



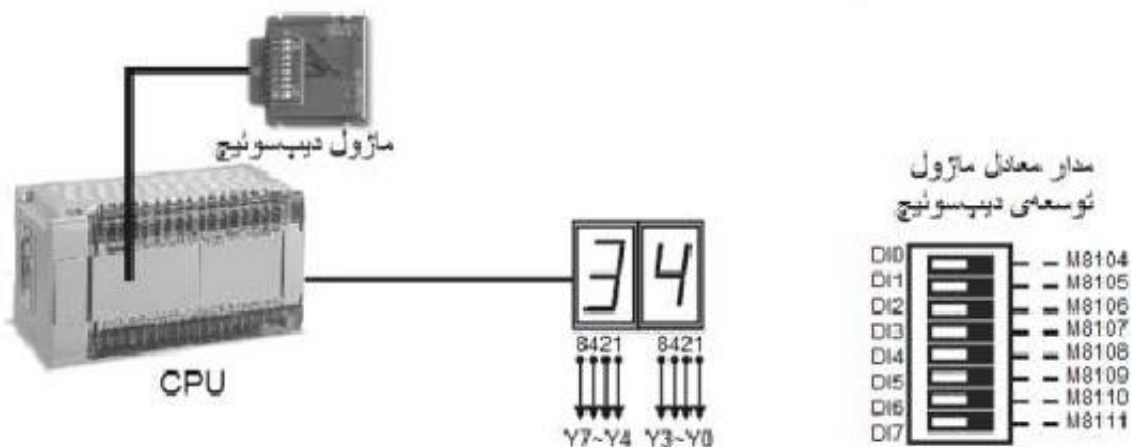
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که ورودی $X1 = ON$ شد، $K4Y0 = H5555$ شده و وضعیت خروجی‌های $Y0 \sim Y17$ برابر با "01010101 01010101" خواهد شد که این به معنی روشن شدن چراغ سیگنال‌های زوج می‌باشد. وقتی که $M8013 = ON$ شد، دستورالعمل CMLP اجرا شده و محتویات داخل $K4Y0$ معکوس خواهند شد. در این حالت، مقدار خروجی‌های $Y0 \sim Y17$ برابر با "10101010 10101010" شده که این به معنی روشن شدن چراغ سیگنال‌های فرد می‌باشد. این وضعیت برای 1 ثانیه طول خواهد کشید.

وقتی که ابزار $M8013$ مجدداً ON شود، دستورالعمل CMLP اجرا شده و وضعیت مقادیر داخل ابزار $K4Y0$ مجدداً معکوس خواهد شد. از این رو، چراغ سیگنال‌های زوج روشن خواهند شد. تا زمانی که ابزار $M8013$ خاموش (OFF) و سپس فعال (ON) شود، وضعیت داخل ابزار $Y0 \sim Y17$ دائماً معکوس خواهد شد و این روند هر 1 ثانیه یک بار تکرار خواهد شد. همانطور که متوجه شدید، چراغ‌های زوج و فرد متناوباً در حال خاموش و روشن شدن (چشمک زن) می‌باشند.

3- 6- کنترل نمایشگر 7-segment

در این مثال، می‌خواهیم با استفاده از دیپ‌سوئیچ‌های متصل شده به ورودی‌های نمایشگرهای (7-seg) متصل شده به خروجی‌ها را کنترل نماییم.



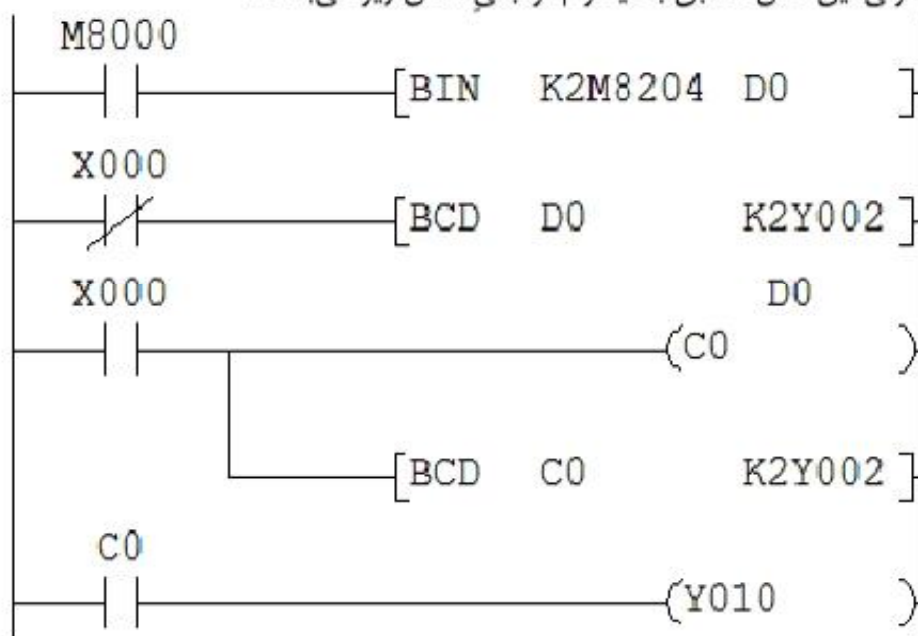
محدوده‌ی ورودی $K0 \sim K99$ محدوده‌ی قابل نمایش: $K0 \sim K99$ هدف از کنترل در این مثال به شرح زیر می‌باشد:

می‌خواهیم مقدار تنظیمی (SV یا SP) شمارنده‌ی $C0$ را در محدوده‌ی بین $K0 \sim K99$ تنظیم نموده و همچنین، مقدار جاری (PV) شمارنده‌ی $C0$ که بین $K0 \sim K99$ می‌باشد را بر روی نمایشگر 7-seg مشاهده کنیم.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

عملکرد ابزار	ابزار
به این ورودی از PLC، سوئیچ راه‌اندازی شمارنده C0 متصل می‌شود.	X0
از این ابزارها برای مسیریابی وضعیت ON/OFF بودن 8 سوئیچ خارجی موجود بر روی دیپ سوئیچ استفاده می‌شود.	M8104~M8111
از این ابزار برای تنظیم مقدار مطلوب (SV یا SP) شمارنده‌ی C0 استفاده می‌شود.	D0
به این خروجی‌های PLC، نمایشگرهای 7-seg برای نمایش مقدار جاری (PV) متصل می‌شود.	Y0~Y17
به این خروجی PLC، چراغ سیگنال رسیدن مقدار جاری شمارنده به مقدار تنظیمی شمارنده متصل می‌شود. وقتی که SP=PV شد، خروجی ON می‌شود.	Y10

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که PLC شروع به کار کرد، وضعیت ON/OFF بودن هر هشت عدد دیپ سوئیچ‌ها بر روی رله‌های کمکی داخلی مسیریابی می‌شوند. شماره‌ی این رله‌های کمکی داخلی شامل M8104~M8111 می‌باشند. دقت کنید که با هشت دیپ سوئیچ می‌توانیم دو رقم یکان و دهگان را توسط دستورالعمل‌ها ایجاد کنیم.

وقتی که برنامه اجرا شود و PLC از مد STOP به Run تغییر وضعیت دهد، فلگ M8000 = ON شده و مقدار تنظیمی شمارنده که توسط دیپ سوئیچ‌ها تنظیم شده است، در ابزار D0 ذخیره خواهد شد. وقتی که شمارنده OFF باشد، ورودی X0 = OFF می‌شود و با اجرای دستورالعمل BCD دو رقم تنظیم شده برای شمارنده C1 بر روی نمایشگر 7-seg نشان داده خواهد شد.

وقتی که شمارنده ON شود، ورودی $X0 = ON$ می‌شود. شمارنده شروع به شمارش کرده و دستورالعمل BCD اجرا خواهد شد که در این حالت بر روی نمایشگر 7-seg دو رقم جاری (PV) شمارنده‌ی C0 نمایش داده می‌شود.

اگر دو رقم نشان داده شده بر روی 7-seg از چپ به راست عدد 34 باشد، این به آن معنی است که وضعیت دیپ سوئیچ‌های متصل شده به ورودی‌های $X0 \sim X7$ برابر با "0011 0100" خواهد بود. وقتی که مقدار جاری (PV) شمارنده‌ی C0 به مقدار تنظیم شده (SP یا SV) در ابزار D0 رسید، کنتاکت باز شمارنده C0 فعال (بسته) شده و خروجی $Y10$ فعال می‌شود که در این حالت چراغ سیگنال تکمیل شمارش روشن می‌شود.

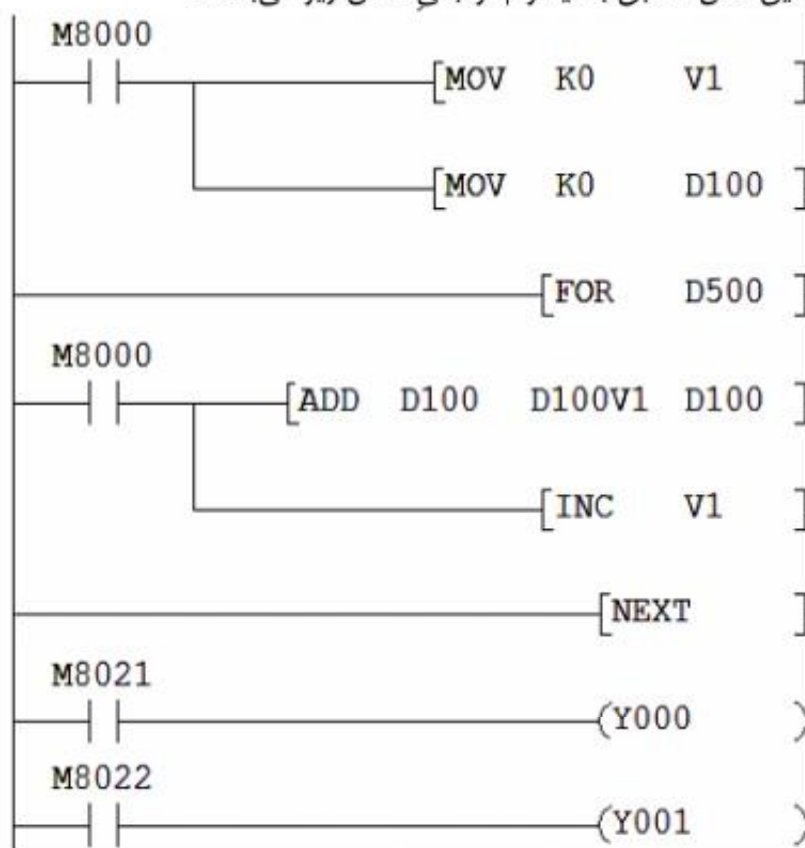
3- برنامه جمع زدن پیوسته اعداد

در این مثال می‌خواهیم برنامه‌ی جمع زدن پیوسته‌ی محتوای رجیسترهای D (مجموعه از رجیسترهای تعیین شده توسط رجیستر شاخص) را بررسی نماییم. هدف از کنترل در این مثال به شرح زیر می‌باشد:

باید محتویات (مقادیر) رجیسترهای D از D101 تا DN (عدد N را خودتان می‌توانید تعیین کنید) جمع زده شوند و نتیجه‌ی حاصل جمع در رجیستر D100 ذخیره شود. اگر حاصل جمع $K-32768 <$ (کوچکتر از عدد 32768-) شد فلگ رقم قرضی (Borrow flag) فعال (ON) شود، و اگر حاصل جمع $K+32767 >$ (بزرگتر از عدد 32767+) شد فلگ رقم نقلی (Carry flag) فعال (ON) شود. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
V1	ابزار رجیستر شاخص
D100	در این ابزار حاصل جمع تمام رجیسترهای D (از D101 الی DN) ذخیره می‌شود.
D500	در این ابزار تعداد دفعات اجزای حلقه‌ی FOR-NEXT ذخیره می‌شود.
Y0	به این خروجی از PLC، چراغ سیگنال فلگ رقم قرضی متصل می‌شود. وقتی که $D100 < K-32768$ شود، خروجی $Y0 = ON$ می‌شود.
Y1	به این خروجی از PLC، چراغ سیگنال رقم نقلی متصل می‌شود. وقتی که $D100 > K32767$ شود، خروجی $Y1 = ON$ می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

نکته‌ی کلیدی این برنامه استفاده از رجیستر شاخص $V1$ و حلقه‌ی $FOR \sim NEXT$ با افزودن $D100V1$ به برنامه می‌باشد. وقتی که $V1 = K1$ شود، $D100V1$ بیانگر رجیستر $D101$ می‌باشد، وقتی که $V1 = K2$ شود، $D100V1$ بیانگر رجیستر $D102$ می‌باشد. همچنین، وقتی که $V1 = K10$ شود، $D100V1$ بیانگر رجیستر $D110$ است.

تعداد دفعات اجرای جمع بستن پیوسته‌ی رجیسترهای D توسط تعداد اجرای حلقه‌ی $FOR \sim NEXT$ تعیین می‌شود. همچنین، تعداد اجرای حلقه‌ی $FOR \sim NEXT$ در ابزار $D500$ تنظیم می‌شود. اگر مقدار $D500$ کوچکتر مساوی با 1 باشد ($D500 \leq 1$)، حلقه $FOR \sim NEXT$ برای یک بار اجرا خواهد شد. اگر مقدار $D500 = K10$ باشد، حلقه FOR ابتدا برای 10 بار اجرا شده و سپس دستورالعمل عقب‌تر از حلقه‌ی FOR اجرا می‌شود.

ابتدا برای حلقه‌ی $FOR \sim NEXT$ مقدار ثابت $V1 = K1$ را تنظیم می‌نماییم، بنابراین ابزار $D100V1$ بیانگر $D101$ خواهد شد. وقتی که دستورالعمل جمع اجرا شود، حاصل جمع $D100 + D101$ در ابزار $D100$ ذخیره خواهد شد. از آنجایی که مقدار اولیه $D100$ صفر تنظیم

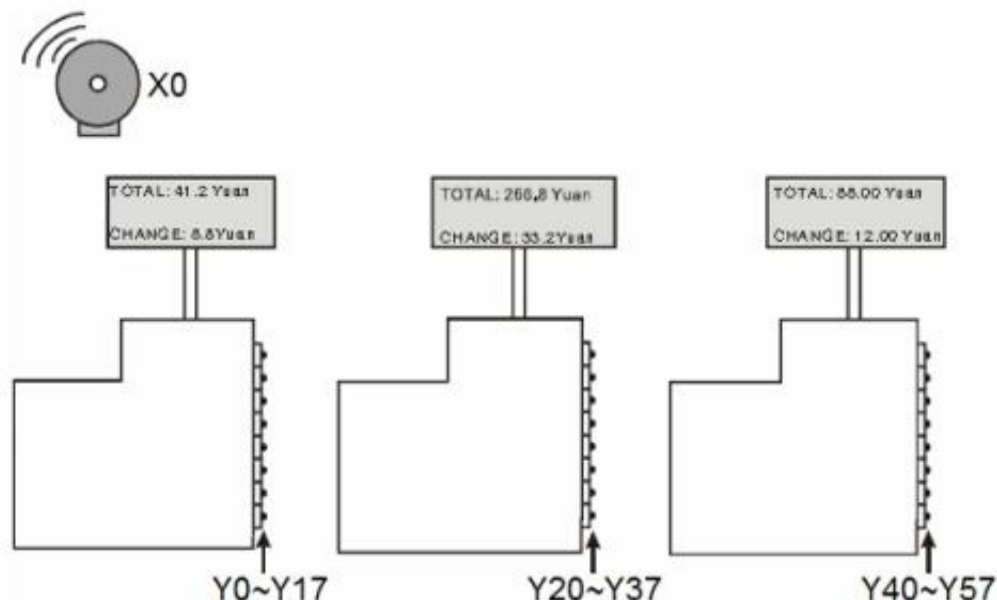
شده است ($D100 = K0$)، مقدار ذخیره شده در ابزار $D100$ برابر با مقدار جاری ابزار $D101$ می‌باشد. در همین هنگام، دستورالعمل NC اجرا شده و $V1 = K2$ تنظیم می‌شود. در دومین مرحله‌ی اجرای حلقه‌ی $FOR \sim NEXT$ مقدار ثابت $V1 = K2$ تنظیم خواهد شد، بنابراین ابزار $D100V1$ حاوی ابزار $D102$ خواهد بود. دستورالعمل جمع اجرا شده و حاصل جمع مقدار $D100 + D102$ در ابزار $D100$ ذخیره می‌شود. از آنجایی که در این مرحله مقدار جاری $D100$ برابر با مقدار $D101$ (حاصل جمع مرحله‌ی قبل) می‌باشد، مقدار ذخیره شده در ابزار $D100$ برابر با مجموع $D101 + D102$ می‌شود.

مطابق با فرآیند جمع، مقدار ابزار $D100$ در دهمین مرحله‌ی اجرای حلقه‌ی $FOR \sim NEXT$ حاصل جمع ابزارهای $D101, D102, D103, D104, D105, D106, D107, D108, D109, D110$ می‌باشد.

اگر در این عملیات حاصل جمع کوچکتر از عدد ثابت 32768 - شود، ابزار $M1021$ (فلگ رقم قرضی) فعال شده و خروجی $Y0 = ON$ می‌شود که در این حالت، چراغ نمایشگر رقم قرضی روشن خواهد شد. همینطور، اگر حاصل جمع بزرگتر از عدد ثابت 32767 شود، ابزار $M1022$ (فلگ رقم نقلی) فعال شده و خروجی $Y1 = ON$ می‌شود که در این حالت، چراغ نمایشگر رقم نقلی روشن خواهد شد.

3-8 سیستم اتوماتیک بستن درب‌های صندوق پول در سوپر مارکت

در این مثال می‌خواهیم با استفاده از دستورالعمل $FOR \sim NEXT$ سیستم اتوماتیک بستن درب‌های صندوق پول را در سوپر مارکت بررسی نماییم.



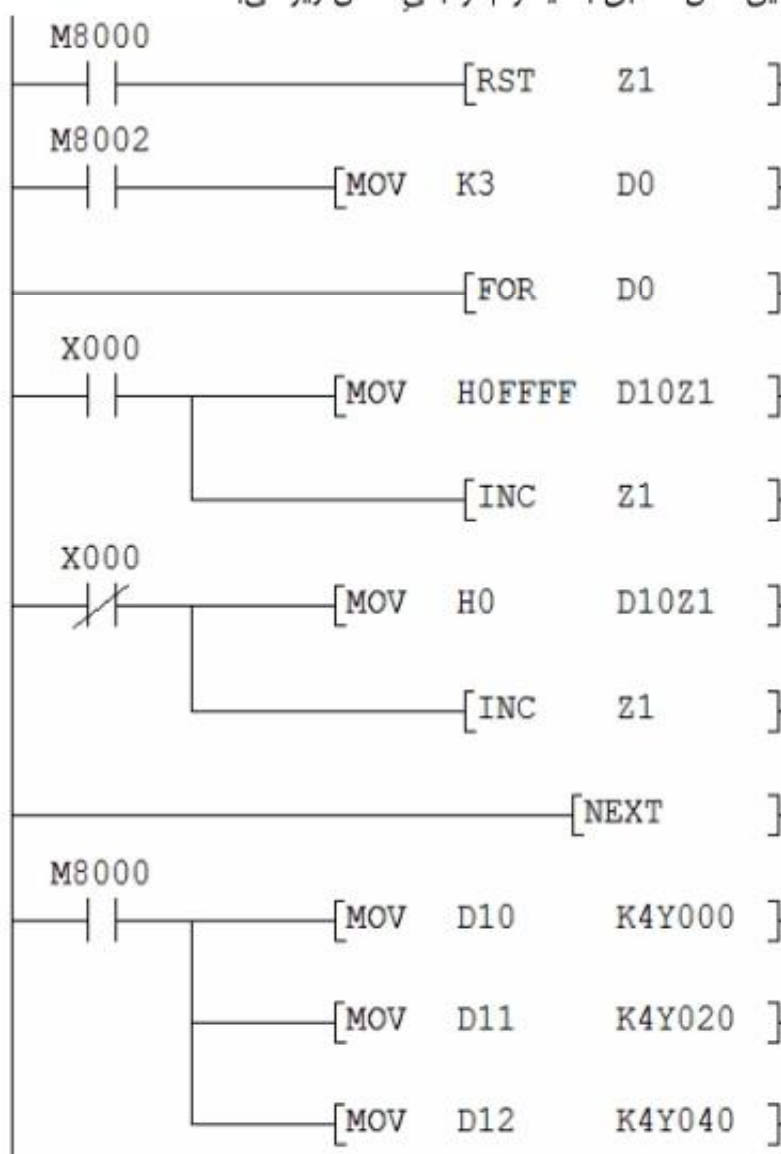
هدف از کنترل در این مثال به شرح زیر می‌باشد:

وقتی که در سوپر مارکتی دزدی و یا آتش سوزی رخ دهد، تمام صندوق‌های پول بسته شده و قفل شوند و تا هنگامی که وضعیت آلامر پاک نشود، صندوق‌ها باز نشوند.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC، سنسور آلامر متصل می‌شود. وقتی که آلامر فعال شد، این ورودی ON شود.
D0	در این ابزار تعداد صندوق‌های پول وجود دارد.
D10	آدرس شروع رجیسترهای مقصد در این ابزار تعیین می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

تعداد دفعات اجرای حلقه‌ی FOR~NEXT را می‌توان توسط تعداد باجه‌های پول که در ابزار D0 وجود دارد، کنترل کرد. هر باجه‌ی پول دارای 16 صندوق (کشو) می‌باشد. در این برنامه $D0 = K3$ می‌باشد که این به معنی 48 کشوی پول در سه باجه است. $(3 * 16 = 48)$.

$F1 = K0$ است، بنابراین D10F1 بیانگر ابزار D10 می‌باشد. اگر $F1 = K1$ شود، D10F1 بیانگر D11 خواهد بود و اگر $F1 = K2$ شود، D10F1 بیانگر D12 بوده و در نهایت، اگر $F1 = K3$ شود، D10F1 بیانگر D13 خواهد بود.

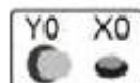
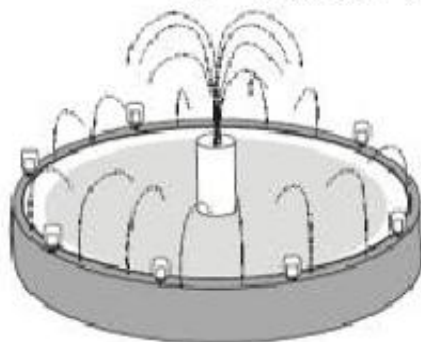
اگر ورودی $X0 = ON$ شود، آلام به صدا در می‌آید. حلقه‌ی FOR~NEXT برای سه بار اجرا شده و مقدار ثابت هگزادسیمال HFFFF به ابزارهای D10~D12 ارسال خواهد شد. بعد از اجرای حلقه، مقادیر موجود در ابزارهای D10~D12 به خروجی‌های خارجی (ماژول خروجی سخت‌افزاری) ارسال خواهند شد. از این رو، تمام خروجی‌های Y فعال (ON) می‌شوند. در نتیجه، سیستم تمام کشوهای پول را قفل خواهد کرد.

اگر ورودی $X0 = OFF$ شود، آلام حذف شده و وضعیت آلام در برنامه پاک می‌شود. حلقه‌ی FOR~NEXT برای 3 بار اجرا شده و عدد ثابت هگزادسیمال H0 به ابزارهای D10~D12 ارسال می‌شوند. بعد از اجرای حلقه، مقادیر موجود در ابزارهای D10~D12 به ترمینال‌های خروجی ماژول سخت‌افزاری ارسال می‌شوند. در این حالت، تمام خروجی‌های Y ریست (OFF) می‌شوند. در نتیجه، سیستم صندوق‌های پول را باز خواهد کرد.

در این برنامه، از رجیستر شاخص F1 برای ذخیره‌سازی مقدار سیگنال در یک پشته‌ی داده (منظور از پشته‌ی داده همان رجیسترهای D می‌باشد) استفاده شده است. مطابق با کاربردهای مختلف برنامه، برنامه‌نویس می‌تواند از پشته‌های داده برای کنترل کردن تایمرها و شمارنده‌ها استفاده کند.

3-9 کنترل آب‌نمای فواره‌ای

در این مثال، می‌خواهیم برنامه‌ی آب‌نمای فواره‌ای را بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

وقتی که شستی START فشرده شد، چراغ سیگنال روشن شده و روشن باقی بماند. بعد از این که چراغ‌های سیگنال برای 2 ثانیه فعال (ON) شود، دیگر ابزارها به ترتیب زیر فعال خواهند شد.

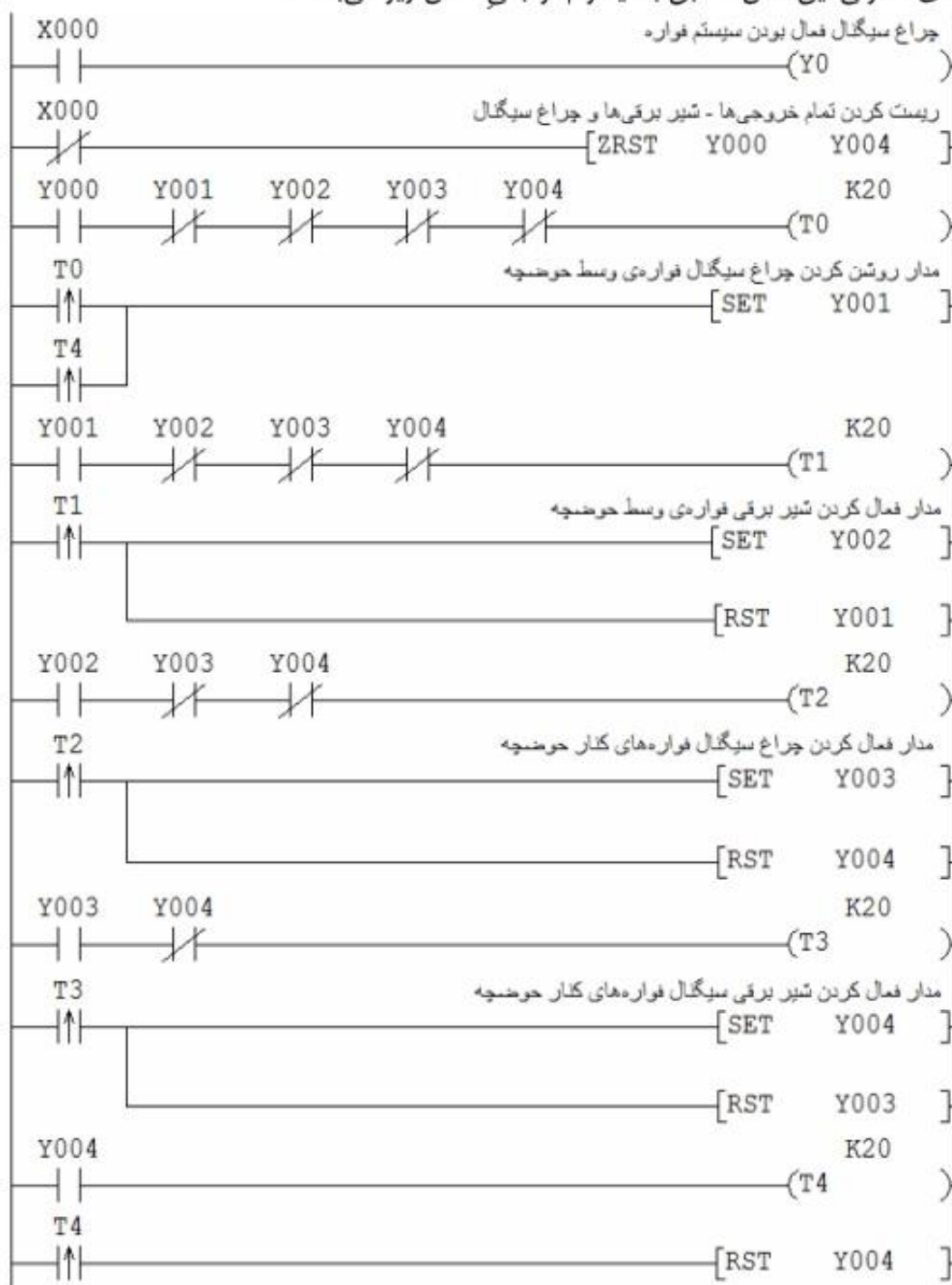
چراغ شیر برقی فواره‌ی آب وسط حوضچه ← شیر برقی فواره‌ی آب وسط حوض ← چراغ‌های دور حوضچه ← شیر برقی فواره‌ی آب دور حوضچه.

ترتیب روشن شدن ابزارهای گفته شده در بالا در فواصل زمانی 2 ثانیه خواهد بود.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	وقتی که شستی START فشرده شود، ورودی X0 فعال می‌شود.
T0	تایمر 2 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. زمان این تایمر را باید برابر با K20 تنظیم نمایید.
T1	تایمر 2 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. زمان این تایمر را باید برابر با K20 تنظیم نمایید.
T2	تایمر 2 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. زمان این تایمر را باید برابر با K20 تنظیم نمایید.
T3	تایمر 2 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. زمان این تایمر را باید برابر با K20 تنظیم نمایید.
T4	تایمر 2 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. زمان این تایمر را باید برابر با K20 تنظیم نمایید.
Y0	به این خروجی، چراغ سیگنال فعال بودن سیستم فواره متصل می‌شود.
Y1	به این خروجی PLC، چراغ سیگنال فواره‌ی وسط حوضچه متصل می‌شود.
Y2	به این خروجی PLC، شیر برقی فواره‌ی وسط حوضچه متصل می‌شود.
Y3	به این خروجی PLC، چراغ سیگنال فواره‌های کنار حوضچه متصل می‌شود.
Y4	به این خروجی از PLC، شیر برقی فواره‌های کنار حوضچه متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که شستی START فشرده شود، ورودی $X0 = ON$ می‌شود، بوبین متصل شده به $Y0 = ON$ شده و چراغ سیگنال فعال بودن سیستم فواره روشن خواهد شد. بعد از این که $Y0 = ON$ شد،

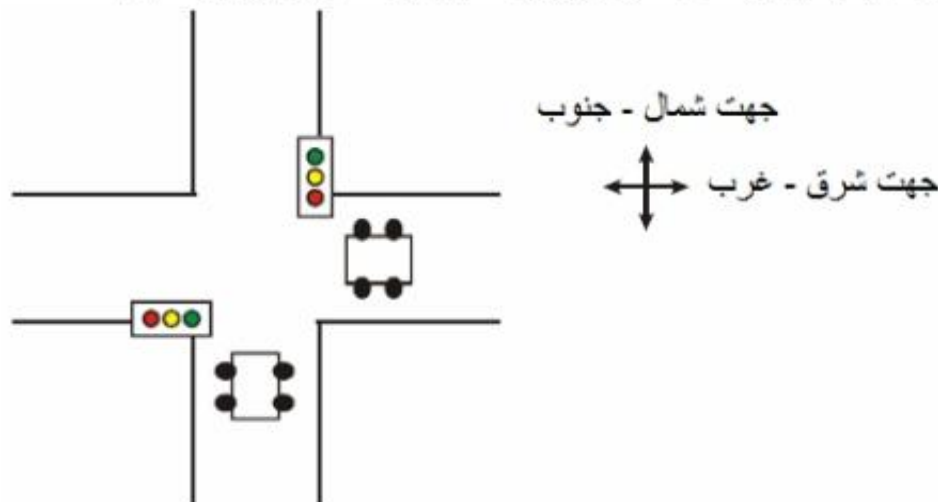
تایمر T0 فعال شده و برای 2 ثانیه شروع به شمارش می‌کند و بعد از گذشت 2 ثانیه، تایمر T0 از OFF به ON تغییر وضعیت داده و دستورالعمل [SET Y1] اجرا خواهد شد. با فعال شدن خروجی Y1 چراغ فواره‌ی وسط حوضچه روشن می‌شود. در تمام زمان کار فرآیند سیستم فواره چراغ سیگنال فعال بودن سیستم فواره روشن خواهد بود.

با فعال شدن خروجی Y0، شرایط اجرای تایمر T1 فراهم می‌شود. و همچنین شرط فعال شدن تایمر T2 فعال شدن خروجی Y2 = ON و شرط فعال شدن تایمر T3 فعال شدن خروجی Y3 می‌باشد. چراغ سیگنال وسط حوضچه، شیر برقی فواره‌ی وسط حوضچه، چراغ سیگنال دور حوضچه و شیر برقی فواره‌ی آب کنار حوضچه به یکدیگر برای راه‌اندازی نیاز دارند. وقتی که هر یک از تایمرهای T3, T1, T2 از OFF به ON تغییر وضعیت دهند، شرایط اجرای تایمر بعدی فراهم خواهد شد و همچنین هر یک اجرای تایمر قبلی را ریست می‌کنند. بعلاوه، از کنتاکت‌های بسته‌ی خروجی‌های Y4, Y3, Y1, Y2 برای غیرفعال کردن تایمرهای T3, T2, T1, T0 استفاده شده است.

بعد از این که اجرای آخرین مرحله‌ی برنامه‌ی فواره تکمیل شد، لبه‌ی بالارونده تایمر T4 خروجی Y4 را ریست کرده و خروجی Y1 را فعال می‌کند. حال مجدداً عملکرد فواره تکرار خواهد شد. وقتی که ورودی X0 = OFF شود، بوبین خروجی Y0 = OFF شده و چراغ سیگنال فعال بودن سیستم فواره خاموش می‌شود. بعلاوه دستورالعمل ZRST در همین زمان اجرا خواهد شد که در این زمان تمام خروجی‌های Y1, Y2, Y3, Y4 ریست شده و تمام شیر برقی‌ها و چراغ‌های سیگنال سیستم فواره‌ی حوضچه فوراً متوقف خواهند شد.

3-10 کنترل چراغ راهنمایی و رانندگی

در این مثال، می‌خواهیم برنامه‌ی کنترل چراغ راهنمایی و رانندگی را بررسی نماییم.

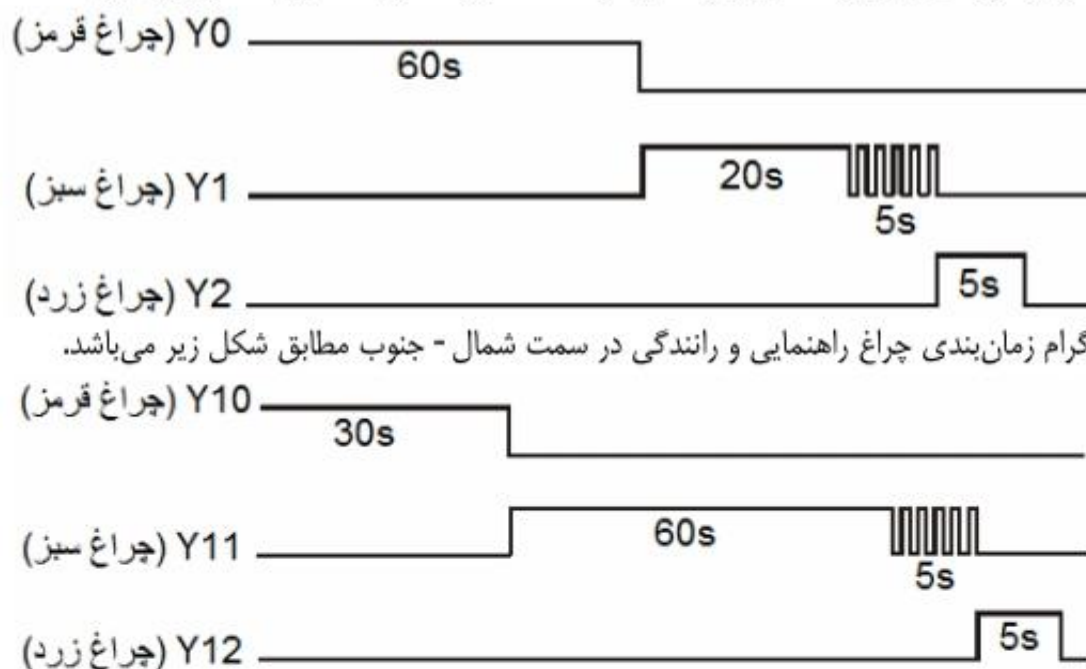


هدف از کنترل در این مثال به شرح زیر می‌باشد:

با فشردن شستی START، ورودی X0 فعال شده و سیستم چراغ راهنمایی و رانندگی شروع به کار می‌کند. همچنین، با فشردن شستی STOP، ورودی X1 فعال شده و سیستم چراغ راهنمایی و رانندگی از کار خواهد افتاد.

زمان تنظیم شده برای چراغ قرمز در سمت شرق - غرب (East-West Direction) برابر با 60 ثانیه بوده و در سمت شمال - جنوب (North-South Direction) برابر با 30 ثانیه می‌باشد. زمان روشن بودن چراغ قرمز در سمت شرق به غرب برابر با زمان - روشن بودن چراغ سبز + روشن بودن چراغ چشمک زن سبز + چراغ زرد - در سمت شمال به جنوب می‌باشد. وقتی که چراغ زرد روشن شد، ماشین‌ها و عابرین پیاده نباید از عرض جاده عبور کنند، در این حالت، چراغ زرد به مدت 5 ثانیه برای عبور ماشین‌ها از تقاطع چهارراه و حفاظت از افرادی که می‌خواهند از عرض خیابان عبور کنند، روشن خواهد شد.

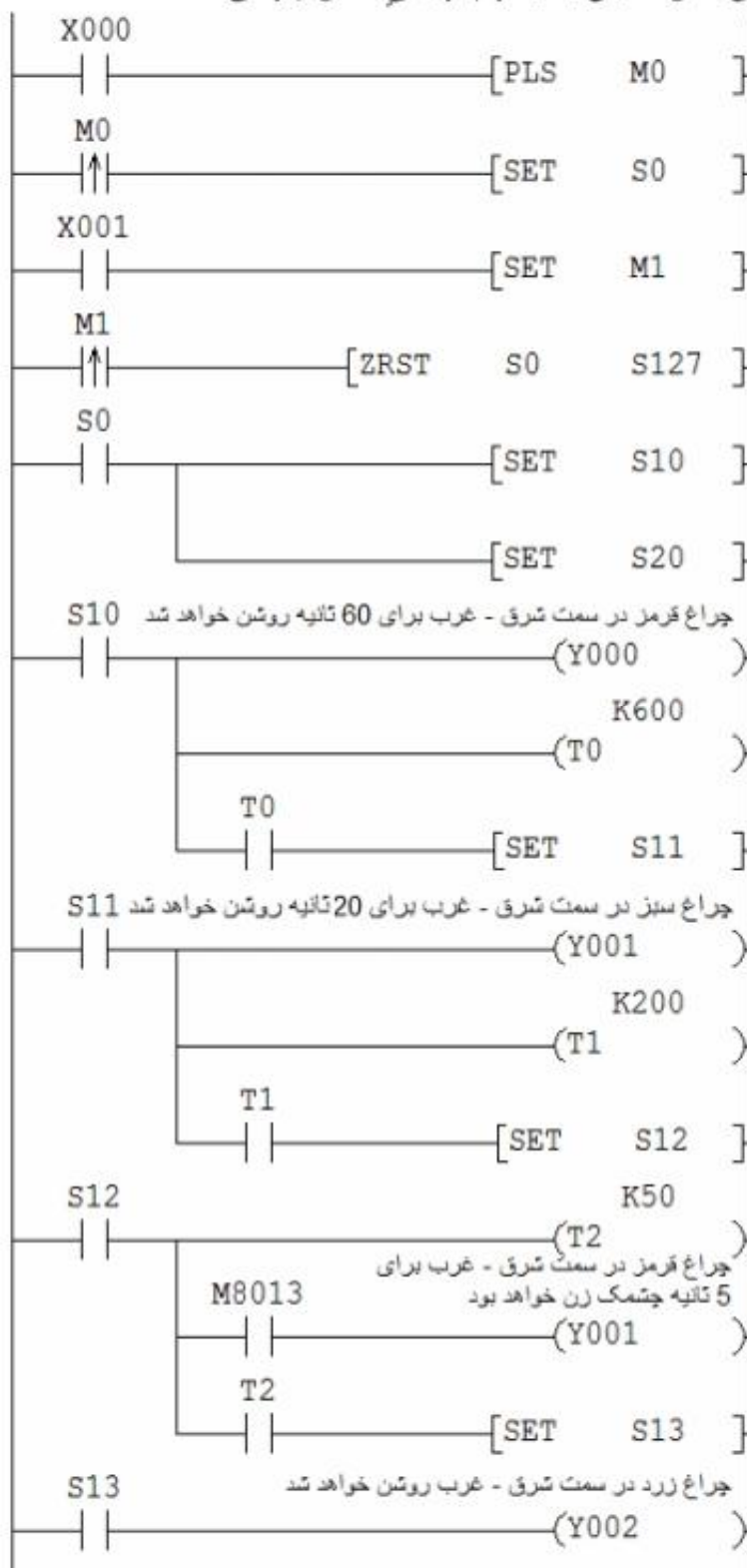
دیاگرام زمان‌بندی چراغ راهنمایی و رانندگی در سمت شرق - غرب مطابق با شکل زیر می‌باشد.

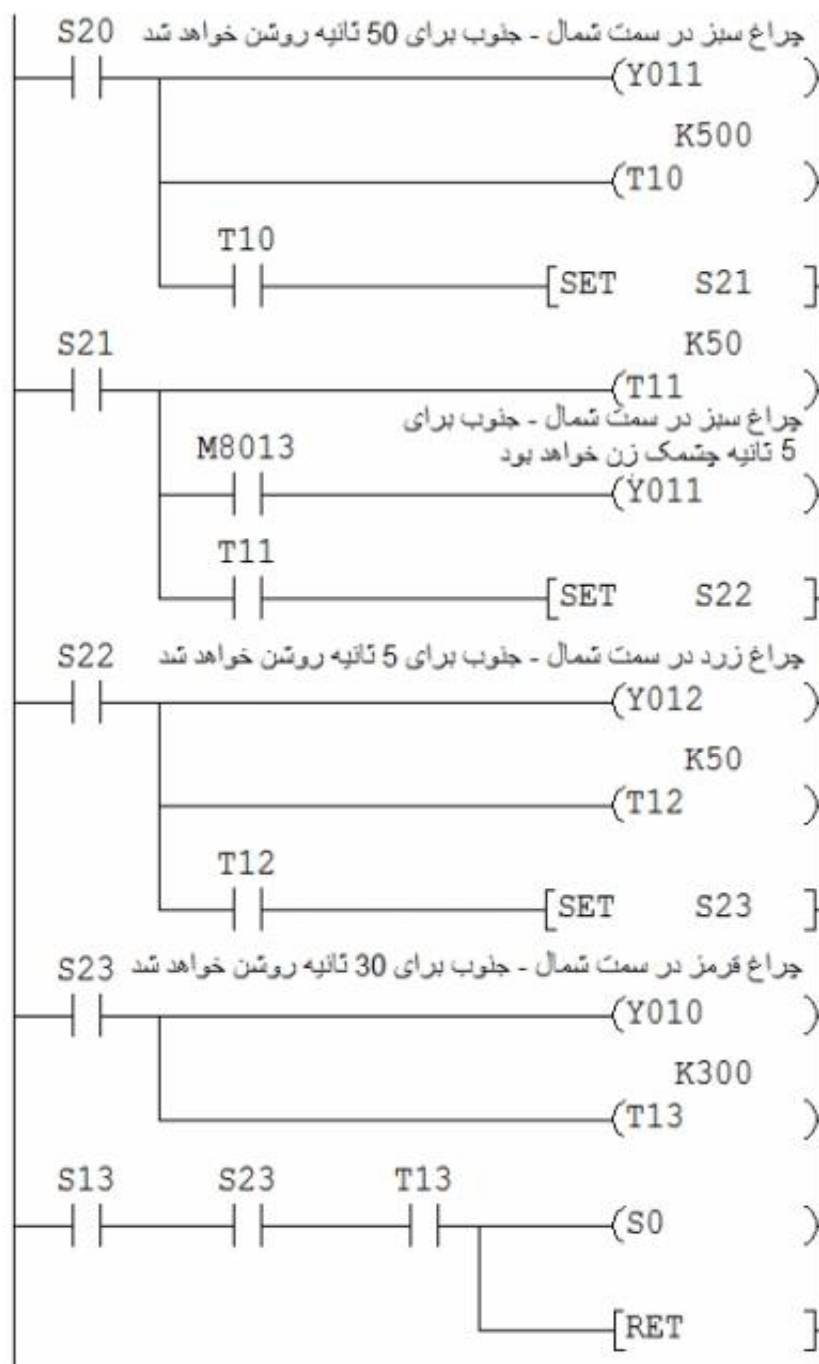


ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

عملکرد ابزار	ابزار
به این ورودی از PLC، شستی START متصل می‌شود.	X0
به این ورودی PLC، شستی STOP متصل می‌شود.	X1
تایمر 60 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 60 ثانیه باید مقدار ثابت تایمر برابر با K600 تنظیم شود.	T0
تایمر 20 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 20 ثانیه باید مقدار ثابت تایمر برابر با K200 تنظیم شود.	T1
تایمر 5 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 5 ثانیه باید مقدار ثابت تایمر برابر با K50 تنظیم شود.	T2
تایمر 50 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 50 ثانیه باید مقدار ثابت تایمر برابر با K500 تنظیم شود.	T10
تایمر 5 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 5 ثانیه باید مقدار ثابت تایمر برابر با K50 تنظیم شود.	T11
تایمر 5 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 5 ثانیه باید مقدار ثابت تایمر برابر با K50 تنظیم شود.	T12
تایمر 30 ثانیه. ضریب این تایمر 100 میلی‌ثانیه است. برای تنظیم 30 ثانیه باید مقدار ثابت تایمر برابر با K300 تنظیم شود.	T13
از این ابزار برای مقداردهی اولین مرحله، نخستین مرحله برای کنترل چراغ راهنمایی و رانندگی استفاده شده است.	S0
از این ابزارها برای کنترل ترتیبی چراغ راهنمایی در سمت شرق - غرب استفاده شده است.	S10~SB
از این ابزارها برای کنترل ترتیبی چراغ راهنمایی در سمت شمال - جنوب استفاده شده است.	S20~S23
به این خروجی PLC، چراغ قرمز در سمت شرق - غرب متصل می‌شود.	Y0
به این خروجی PLC، چراغ سبز در سمت شرق - غرب متصل می‌شود.	Y1
به این خروجی PLC، چراغ زرد در سمت شرق - غرب متصل می‌شود.	Y2
به این خروجی PLC، چراغ قرمز در سمت شمال - جنوب متصل می‌شود.	Y10
به این خروجی PLC، چراغ سبز در سمت شمال - جنوب متصل می‌شود.	Y11
به این خروجی PLC، چراغ زرد در سمت شمال - جنوب متصل می‌شود.	Y12

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:





عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که شستی **START** فشرده شود، ورودی $X0 = ON$ خواهد شد. دستورالعمل **PLS** اجرا شده و ابزار **M1** یک پالس با لبه‌ی بالارونده را برای فعال کردن تایمر **T0** ایجاد می‌کند. در این حالت برنامه وارد فرآیند نردبانی مرحله‌ی (**STEP**) می‌شود.

وقتی که شستی STOP فشرده شود، ورودی $X1 = ON$ شده و دستورالعمل PLS اجرا خواهد شد که در این حالت، ابزار M1 یک پالس لبه‌ی بالارونده را برای اجرا دستورالعمل [ZRST S0 S127] ایجاد می‌کند. با اجرای این دستور، تمام مراحل (ابزارهای S) ریست خواهند شد و تمام چراغ‌های راهنمایی و رانندگی در تقاطع چهارراه خاموش می‌شوند.

این مثال برنامه‌ی ترتیبی همزمان با چندین انشعاب را توضیح می‌دهد که دو برنامه‌ی همزمان ترتیبی در کنار هم و با هم کار می‌کنند و آن هم برنامه‌ی ترتیبی سمت شمال - جنوب و برنامه‌ی ترتیبی سمت شرق - غرب است.

وقتی که چراغ قرمز سمت شرق - غرب روشن شود، متناظر با این وضعیت در سمت شمال - جنوب به ترتیب چراغ سبز روشن شده و سپس چراغ سبز چشمک زن می‌شود و در نهایت، چراغ قرمز رنگ روشن خواهد شد.

وقتی که در سمت شرق - غرب ترتیب عملکرد چراغ راهنمایی تمام شد (چراغ زرد خاموش شد)، در سمت شمال - جنوب نیز ترتیب عملکرد تمام خواهد شد (چراغ قرمز خاموش می‌شود). در این حالت برنامه به نخستین مرحله S0 باز خواهد گشت.

وقتی که در یک مرحله، انتقال از یک ترتیب به ترتیب بعدی صورت گرفت، مرحله قبلی که شامل مرحله و نقطه خروجی Y می‌باشد، ریست می‌شود.

وقتی که چراغ قرمز در سمت شمال - جنوب خاموش باشد، زمان چراغ زرد در سمت شرق - غرب (Y2) توسط تایمر کنترل نخواهد شد، در همین زمان چراغ زرد در سمت شمال - جنوب ریست خواهد شد. از این رو، تایمر T13 به طور غیرمستقیم با اجرای برنامه مرحله‌ی اول (S0) فعال شده و خروجی‌های Y2 و Y10 مطابق با مراحل S13 و S23 ریست خواهند شد.

3- 11 برنامه پخت مواد غذایی به همراه تنظیمات پارامتری پخت

در این مثال، می‌خواهیم برنامه تنظیمات پارامتری برای پختن محصول غذایی توسط انتقال پارامترهای تعیین شده در رجیسترهای شاخص به برنامه را بررسی نماییم.

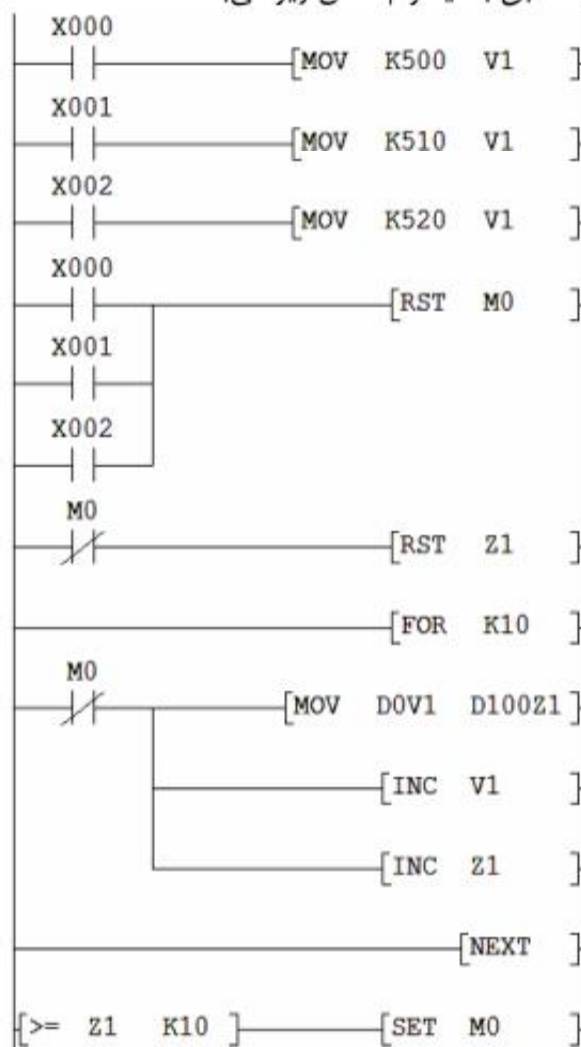
هدف از کنترل در این مثال به شرح زیر می‌باشد:

برای یک محصول، سه روش پخت وجود دارد. هر روش پخت شامل 10 پارامتر می‌باشد. با استفاده از یک سوئیچ سه وضعیتی می‌توان روش پخت را انتخاب کرده و پارامترهای مربوط به آن روش را به برنامه اعمال کرد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC، اولین وضعیت از سوئیچ سلکتوری سه وضعیتی متصل می‌شود. این وضعیت مربوط به اعمال کردن پارامترهای اولین روش پخت به برنامه می‌باشد.
X1	به این ورودی PLC، دومین وضعیت از سوئیچ سلکتوری سه وضعیتی متصل می‌شود. این وضعیت مربوط به اعمال کردن پارامترهای دومین روش پخت به برنامه می‌باشد.
X2	به این ورودی PLC، سومین وضعیت از سوئیچ سلکتوری سه وضعیتی متصل می‌شود. این وضعیت مربوط به اعمال کردن پارامترهای سومین روش پخت به برنامه می‌باشد.
D500~D509	از این ابزارها برای ذخیره‌سازی مقادیر پارامترهای اولین روش پخت استفاده می‌شود.
D510~D519	از این ابزارها برای ذخیره‌سازی مقادیر پارامترهای دومین روش پخت استفاده می‌شود.
D520~D529	از این ابزارها برای ذخیره‌سازی مقادیر پارامترهای سومین روش پخت استفاده می‌شود.
D100~D109	از این ابزارها به عنوان مقادیر جاری پارامترهای برنامه‌ی پخت استفاده می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

نکته‌ی کلیدی این برنامه استفاده از رجیسترهای شاخص V1 و Z1 در کنار حلقه‌ی FOR~NEXT برای کار بر روی تعدادی از رجیسترهای D می‌باشد. بعلاوه، مقدار پارامترهای انتخاب شده توسط سوئیچ سه وضعیتی به رجیسترهای کاری در برنامه منتقل می‌شوند.

وقتی که یکی از ورودی‌های X0، X1 و یا X2 توسط سوئیچ سه وضعیتی فعال شود، پارامترهای مورد نظر انتخاب می‌شوند. مطابق با مقدار انتخاب شده برای V1 توسط سوئیچ سه وضعیتی، عدد رجیستر D0V1 یکی از رجیسترهای D500، D510 و یا D520 خواهد شد. دستورالعمل [RST M0] اجرا شده و Z1 ریست می‌شود. سپس دستورالعمل FOR~NEXT اجرا می‌شود. از آنجایی که Z1 ریست شده مقدار آن K0 خواهد بود، بنابراین D100Z1 برابر با D100 خواهد شد.

در این برنامه حلقه FOR~NEXT برای 10 بار اجرا می‌شود. اگر توسط سوئیچ سه وضعیتی این روش پخت انتخاب شود، رجیستر شاخص D0V1 به رجیستر D500 تا D509 تغییر کرده و رجیستر شاخص D100Z1 به رجیسترهای D100 تا D109 تغییر می‌کند.

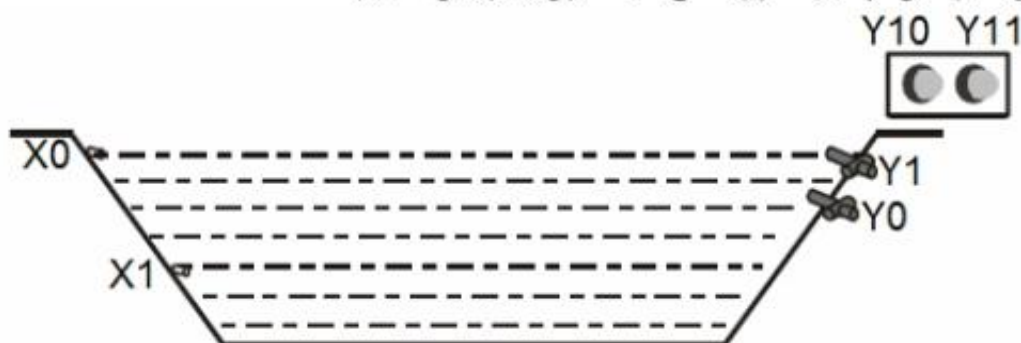
بعلاوه، در نخستین حلقه‌ی FOR~NEXT مقدار جاری در رجیستر D500 به رجیستر D100 ارسال خواهد شد. در دومین بار اجرای حلقه‌ی FOR~NEXT مقدار جاری در رجیستر D501 به رجیستر D101 ارسال خواهد شد. به همین ترتیب، در هر 10 باری که حلقه اجرا شود، این روند ادامه خواهد داشت. در دهمین بار اجرای حلقه‌ی FOR~NEXT مقدار جاری در رجیستر D509 به رجیستر D109 ارسال خواهد شد.

وقتی که دفعات اجرای حلقه به مقدار تنظیم شده رسید، یعنی $Z1 = K10$ شد دستورالعمل [SET M0] اجرا خواهد شد. کنتاکت بسته‌ی ابزار M0 فعال (باز) شده و عملکرد دستورالعمل حلقه‌ی FOR~NEXT متوقف می‌شود.

در هر وضعیت (انتخاب روش پخت) انتقال مقادیر هر 10 پارامتر در برنامه انجام خواهد شد. تعداد پارامترها را می‌توان به سادگی با استفاده از تنظیم اجرای تعداد دفعات اجرای حلقه‌ی FOR~NEXT تغییر داد. گذشته از این، اگر بخواهیم به تعداد روش‌های پخت اضافه کنیم، می‌توان با استفاده از اضافه کردن دستورالعمل MOV به عنوان مثال [MOV K530 V1] به برنامه تعداد روش‌های پخت (گروه پارامترهای اضافی) افزود.

3- 12- کنترل سطح آب مخزن

در این مثال، می‌خواهیم کنترل سطح آب مخزن را بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

وقتی که سطح آب داخل مخزن بیشتر از بالاترین محدوده‌ی تعیین شده رفت، یک آلارم وضعیت نرمال نبودن سطح آب فعال شده و آب داخل مخزن تخلیه شود.

وقتی که سطح آب داخل مخزن کمتر از پایین‌ترین محدوده‌ی تعیین شده رفت، یک آلارم وضعیت نرمال نبودن سطح آب فعال شده و آب داخل مخزن پر شود.

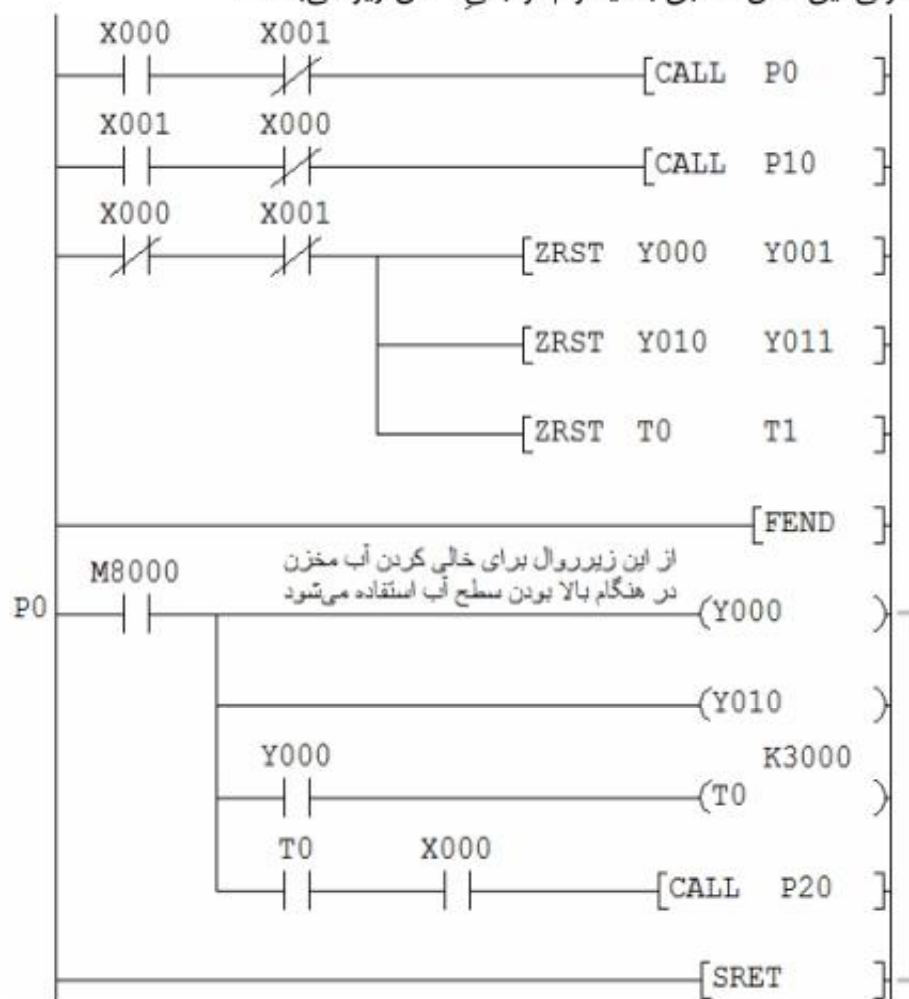
اگر سنسور بالاترین محدوده‌ی آب (سطح آب) X0 بعد از گذشت 10 دقیقه از فرآیند تخلیه‌ی آب مخزن فعال (ON) بود، آلارم خطای مکانیکی روشن شود.

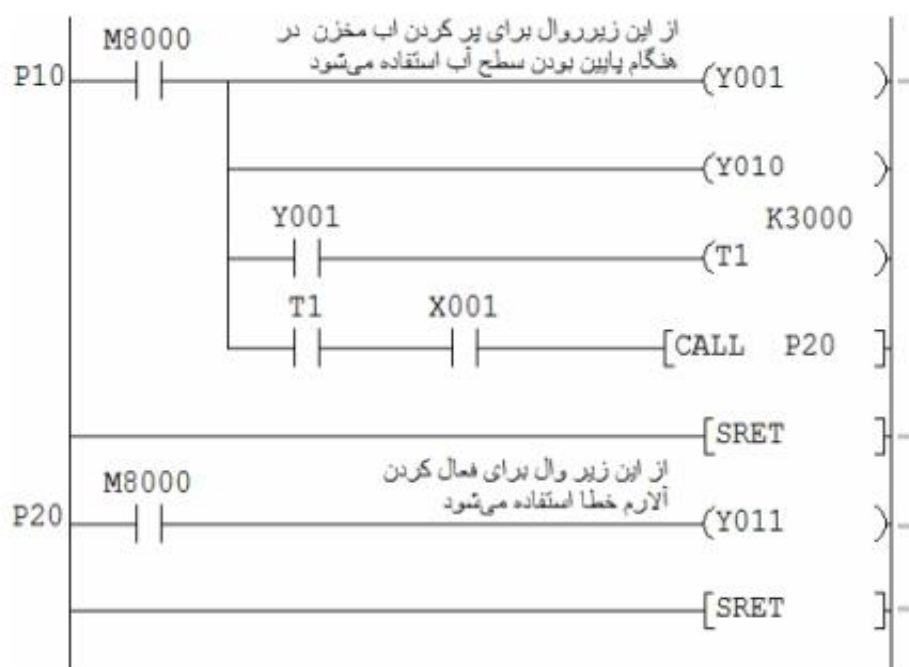
اگر سنسور پایین‌ترین محدوده‌ی آب (سطح آب) X1 بعد از گذشت 5 دقیقه از فرآیند پر کردن آب مخزن فعال (ON) بود، آلارم خطای مکانیکی روشن شود.

وقتی که سطح آب به موقعیت (محدوده‌ی) نرمال رسید، تمام آلارم‌ها و شیربرقی‌ها ریست شوند. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، سنسور بالاترین سطح آب (لول سنج شناور، فلوتر) متصل می‌شود. وقتی که آب به بالاترین محدوده‌ی مخزن رسید، ورودی ON می‌شود.
X1	به این ورودی از PLC، سنسور پایین‌ترین سطح آب (لول سنج شناور، فلوتر) متصل می‌شود. وقتی که آب به پایین‌ترین محدوده‌ی مخزن رسید، ورودی ON می‌شود.
Y0	به این خروجی از PLC، شیر برقی تخلیه آب مخزن متصل می‌شود.
Y1	به این خروجی از PLC، شیر برقی پر کردن آب مخزن متصل می‌شود.
Y10	به این خروجی از PLC، آلارم وضعیت نرمال نبودن سطح آب متصل می‌شود.
Y11	به این خروجی از PLC، آلارم خطای مکانیکی متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:





عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که سطح آب داخل مخزن به بالاترین محدوده رسید، ورودی $X0 = ON$ شده و دستورالعمل $[CALL P0]$ اجرا می‌شود. آلام وضعیت غیرعادی $Y10$ و شیر برقی تخلیه آب مخزن $Y0$ هر دو تا هنگامی که سطح آب مخزن به پایین محدوده‌ی بالا برسد، شروع به کار می‌کنند.

وقتی که سطح آب به پایین‌ترین محدوده رسید، ورودی $X1 = ON$ شده و دستورالعمل $[CALL P10]$ اجرا می‌شود. آلام وضعیت غیرعادی $Y10$ و شیر برقی پر کردن آب مخزن $Y1$ تا هنگامی که آب مخزن به بالای سنسور محدوده‌ی پایین آب برسد، شروع به کار می‌کنند.

زیرروال $CALL P20$ تودرتو بوده و حاوی دو زیرروال $P0$ و $P1$ می‌باشد. اگر سنسور بالاترین محدوده‌ی آب مخزن بعد از گذشت 10 دقیقه از روشن بودنش هنوز فعال بود، زیرروال $P20$ اجرا خواهد شد. خروجی $Y11$ فعال شده و آلام خطای مکانیکی فعال می‌شود.

همچنین، اگر سنسور فلوتر محدوده‌ی پایین آب مخزن فعال شود و بعد از گذشت 5 دقیقه از پر شدن آب هنوز شیر برقی پر کردن آب مخزن فعال بود، Subroutine زیرروال $P20$ اجرا می‌شود. خروجی $Y11$ فعال شده و آلام خطای مکانیکی فعال می‌شود.

اگر سطح آب در موقعیت نرمال باشد، ورودی‌های $X0 = OFF$, $X1 = OFF$ می‌شود که در این حالت، دستورالعمل $ZRST$ اجرا خواهد شد. خروجی‌های $Y0$, $Y1$, $Y10$, $Y11$, $T0$, $T1$ ریست شده و تمام شیربرقی‌ها و آلام‌ها غیرفعال می‌شوند.

3- 13 مخلوط کن مواد (میکسر)

در این مثال، می‌خواهیم دستگاه مخلوط کن مواد (میکسر) را با استفاده از دستورالعمل CMP بررسی نماییم.

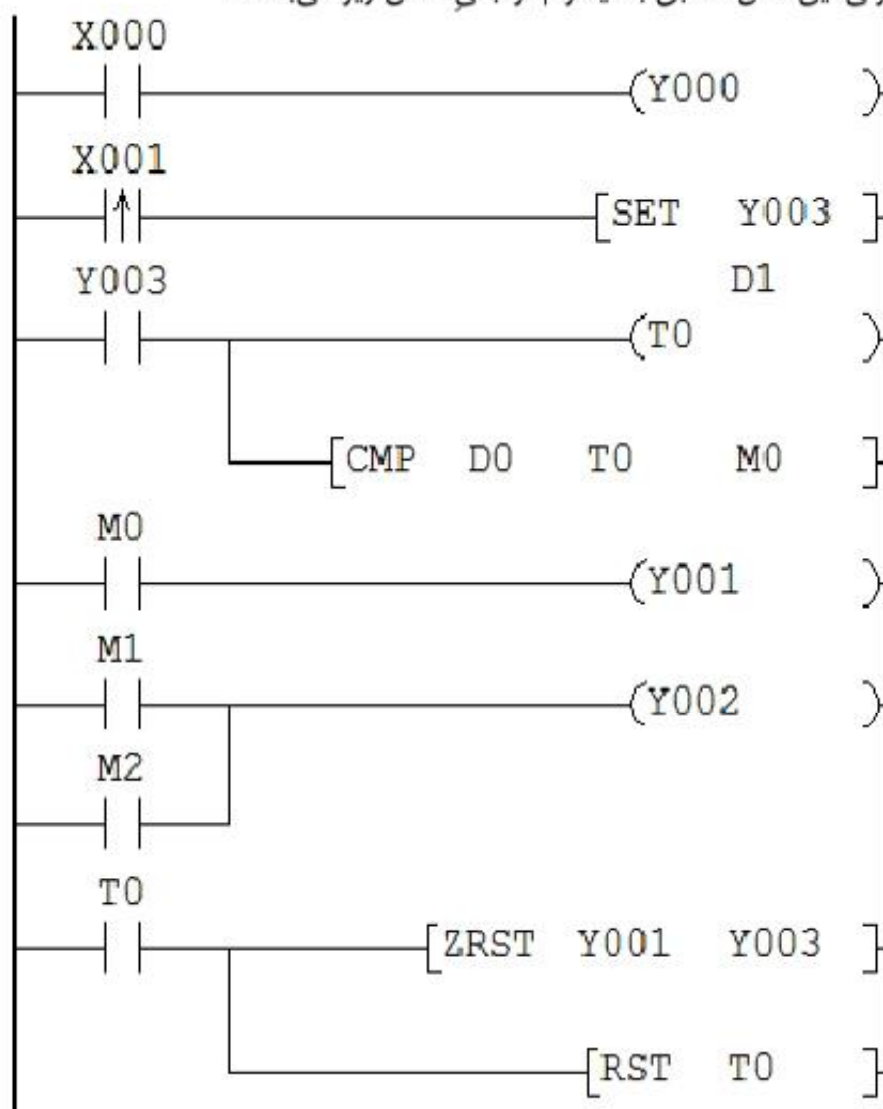
هدف از کنترل در این مثال به شرح زیر می‌باشد:

دو نوع مواد اولیه‌ی A و B برای مخلوط کردن توسط دستگاه میکسر وجود دارد. وقتی که شستی Power on فشرده شود، نمایشگر خروجی متصل شده به Y0 روشن می‌شود. با فشردن شستی Process ورودی X1 = ON شده و دریچه ورود مواد A (خروجی Y1) به میکسر باز شده و موتور هم زن Y3 شروع به کار می‌کند. وقتی که زمان فرآیند تغذیه مواد A به میکسر به زمان تنظیم شده در ابزار D0 رسید، دریچه ورود مواد B (خروجی Y2) به میکسر فعال شده و مواد B شروع به تخلیه در میکسر می‌کند. وقتی که زمان مخلوط کردن هر دو مواد با هم در میکسر به مقدار تنظیم شده در ابزار D1 رسید، تمام فرآیند میکسر استوپ می‌شود.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، شستی Power on متصل می‌شود. وقتی که شستی فشرده شود، ورودی ON می‌شود.
X1	به این ورودی از PLC، شستی Process متصل می‌شود. وقتی که شستی فشرده شود، ورودی ON می‌شود.
D0	در این ابزار زمان تغذیه مواد A به داخل میکسر وجود دارد.
D1	در این ابزار مجموع زمان تغذیه‌ی مواد A و B به داخل میکسر ذخیره شده است.
Y0	به این خروجی از PLC، چراغ سیگنال Power on متصل می‌شود.
Y1	به این خروجی از PLC، دریچه تخلیه مواد A به داخل میکسر متصل می‌شود.
Y2	به این خروجی از PLC، دریچه تخلیه مواد B به داخل میکسر متصل می‌شود.
Y3	به این خروجی از PLC، موتور همزن متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که شستی Power on فشرده شود، ورودی $X0 = ON$ می‌شود که باعث روشن شدن چراغ سیگنال Power on متصل شده به خروجی $Y0$ خواهد شد.

وقتی که شستی Process فشرده شود، ورودی $X1 = ON$ می‌شود. دستورالعمل $[SET Y3]$ اجرا شده و همچنین دستورالعمل TMR اجرا خواهد شد که باعث فعال شدن تایمر $T0$ می‌شود. در همین زمان، دستورالعمل CMP اجرا خواهد شد. وقتی که مقدار جاری (PV = Present Value) در تایمر $T0$ کوچکتر از مقدار تنظیمی (SV (Set Value) در ابزار $D0$ باشد، ابزار $M0$ $= ON$ می‌شود. بنابراین، با فعال شدن $M1$ خروجی $Y1$ فعال خواهد شد. مواد A شروع به تخلیه در

میکسر می‌کند. وقتی که مقدار PV در تایمر T0 بزرگتر مساوی با مقدار SV در ابزار D0 شود، ابزار M1 و M2 فعال شده و ابزار M0 غیرفعال (OFF) می‌شود. در این حالت خروجی Y2 فعال می‌شود و مواد B نیز شروع به تخلیه‌ی در میکسر می‌کند. در همین زمان، فرآیند تخلیه مواد A داخل میکسر متوقف می‌شود.

وقتی که مقدار PV در تایمر T0 به مقدار SV در D1 رسید، کنتاکت باز تایمر T0 فعال (ON) (بسته) شده و دستورالعمل‌های RST و ZRST اجرا می‌شوند. خروجی‌های T1, Y1, Y2, Y3 ریست خواهند شد و موتور همزن میکسر تا هنگامی که مجدداً دکمه Process فشرده شود، متوقف خواهد شد.

3- 14- بکاپ‌گیری داده در چندین زمان مختلف

در این مثال، می‌خواهیم برنامه‌ی بکاپ‌گیری داده در چندین زمان مختلف را توسط دستورالعمل BMOV بررسی نماییم.

هدف از کنترل در این مثال به شرح زیر می‌باشد:

می‌خواهیم داده‌های DUT (Device Under Test) (ابزار زیرتست) را در رجیسترهای D0~D99 بعد از کار کردن برای نخستین بار دستگاه ثبت کنیم. در اولین بار کارکرد دستگاه باید مقادیر دستگاه را به صورت تجربی تنظیم کنیم تا دستگاه به درستی کار کند. بعد از ثبت مقادیر باید در هر 30 دقیقه یک بار بکاپ گرفته شود. یک سیکل تست DUT به مدت 2 ساعت طول می‌کشد. لازم به ذکر است که مقادیر رجیسترهای D0~D99 را می‌توان مجدداً تغییر داده و کامپایل کرد. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC، شستی START متصل می‌شود. وقتی که این شستی فشرده شود، ورودی ON می‌شود.
X1	به این ورودی PLC، شستی RETEST متصل می‌شود. وقتی که این شستی فشرده شود، ورودی ON می‌شود.
X2	به این ورودی PLC، شستی STOP متصل می‌شود. وقتی که این شستی فشرده شود، ورودی ON می‌شود.
D0~D99	ابزارهای مربوط به کامپایل کردن داده‌ها
D100~D499	ابزارهای مربوط به بکاپ‌گیری از مقادیر داده‌ها

[illegible]

وقتی که ورودی $X0 = ON$ شود، تایمر $T0$ شروع به شمارش کرده و کنتاکت باز این تایمر هر 30 دقیقه یک بار فعال (بسته) می‌شود.

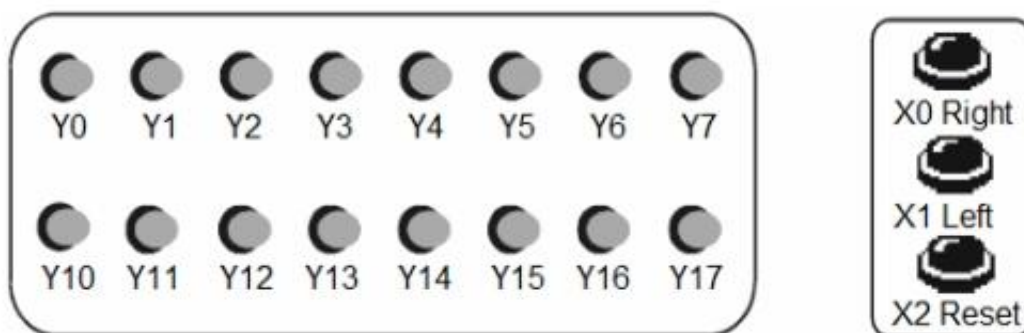
در این برنامه، از شمارنده C1 برای شمارش تعداد دفعات فعال (بسته) شدن کنتاکت باز تایمر T0 استفاده شده است. وقتی که $C0 = 1$ شد، داده‌های موجود در ابزارهای D0~D99 به ابزارهای D100~D199 ارسال خواهد شد. وقتی که $C0 = 2$ شد، داده‌های موجود در ابزارهای D0~D99 به ابزارهای D200~D299 ارسال خواهد شد. وقتی که $C0 = 3$ شد، داده‌های موجود در ابزارهای D0~D99 به ابزارهای D300~D399 ارسال خواهد شد. و وقتی که $C0 = 4$ شد، داده‌های موجود در ابزارهای D0~D99 به ابزارهای D400~D499 ارسال می‌شوند که در این زمان فرآیند تست پایان خواهد یافت.

اگر لازم باشد که مجدداً تایمر DUT تست شود، کافی است که مجدداً شستی START فشرده شود.

وقتی که ورودی $X2 = ON$ شود، فرآیند تست متوقف خواهد شد. از این رو، هیچ داده‌ای برای DUT توسط PLC کامپایل نخواهد شد، و مقدار شمارنده‌ی C0 پاک شده و ریست می‌شود.

3- 15- کنترل یک رشته لامپ نئون

در این مثال، می‌خواهیم برنامه یک لامپ نئون را با استفاده از دستورالعمل ROL/ROR بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

می‌خواهیم 16 لامپ نئون را که به خروجی‌های Y0~Y7 و Y10~Y17 متصل هستند را فعال کنیم. وقتی که شستی Rotation right فشرده شود، لامپ‌ها روشن خواهند شد. هر لامپ به مدت 200ms روشن خواهد بود.

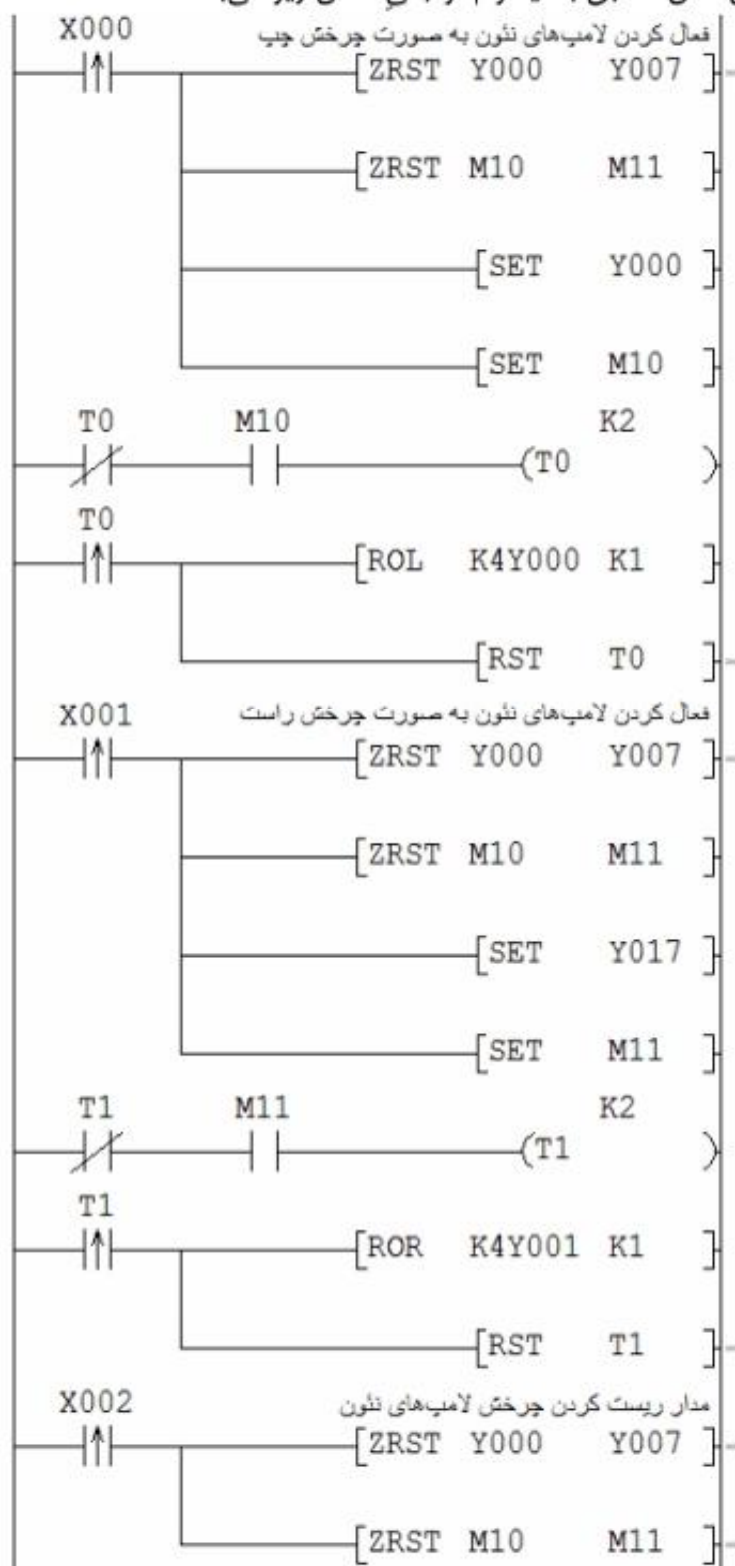
وقتی که شستی Rotation left فشرده شود لامپ‌های نئون به ترتیب Y7~Y0 و Y17~Y10 روشن خواهند شد. هر لامپ به مدت 200ms روشن خواهد بود.

وقتی که نیازی به روشن شدن لامپ از چپ به راست و یا از راست به چپ نباشد، باید دکمه‌ی Reset فشرده شود. در این صورت تمام لامپ‌های نئون خاموش شوند.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، شستی Rotation right متصل می‌شود. وقتی که شستی فشرده شود، ورودی ON می‌شود.
X1	به این ورودی از PLC، شستی Rotation left متصل می‌شود. وقتی که شستی فشرده شود، ورودی ON می‌شود.
X2	به این ورودی از PLC، شستی Reset متصل می‌شود. وقتی که شستی فشرده شود، ورودی ON می‌شود.
T0/T	تایمر 200 میلی ثانیه ضرب این تایمر 100 میلی ثانیه می‌باشد. برای تنظیم 200 میلی ثانیه عدد ثابت تایمر را برابر با K2 تنظیم نمایید.
Y0~Y17	به این خروجی‌های PLC، 16 لامپ نئون متصل می‌شود. یعنی به هر خروجی یک لامپ نئون متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



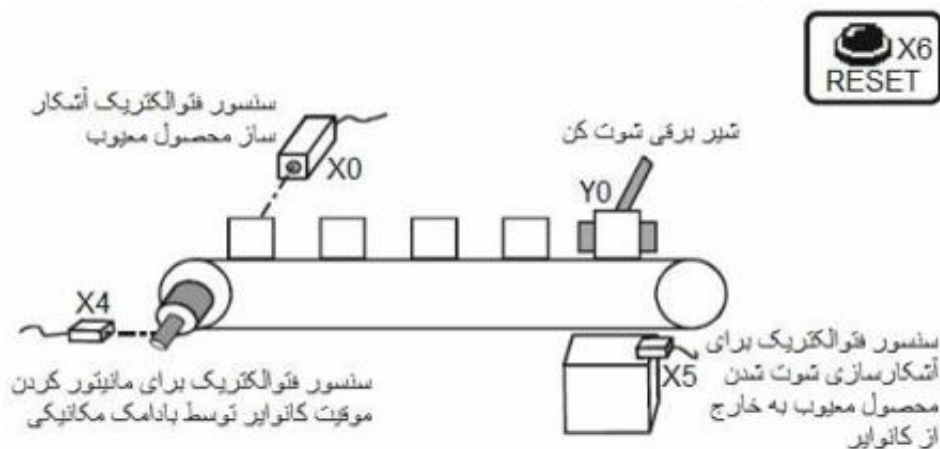
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که شستی Rotation right فشرده شود، ورودی $X0 = ON$ شده و دستورالعمل‌های ZRST و SET اجرا می‌شوند. ابتدا خروجی‌های $Y0 \sim Y7$ و ابزارهای $M10 \sim M11$ ریست می‌شوند. سپس خروجی $Y0$ و ابزار $M10$ فعال (ON) می‌شوند. دستورالعمل TMR اجرا خواهد شد. بعد از گذشت 200ms کنتاکت تایمر $T0$ فعال شده و دستورالعمل ROL یکبار اجرا می‌شود. سپس وضعیت ON از خروجی $Y0$ به خروجی $Y1$ شیفت یافته و تایمر $T0$ ریست خواهد شد. در سیکل اسکن بعدی، تایمر $T0$ مجدداً شروع به شمارش خواهد کرد. بعد از گذشت 200ms، دستورالعمل ROL اجرا شده که در این صورت وضعیت ON از $Y1$ به $Y2$ تغییر خواهد یافت. تا زمان فشرده شدن دکمه Reset این روند برای خروجی‌های $Y0 \sim Y17$ در هر 200 میلی‌ثانیه یکبار اجرا خواهد شد.

روند چرخش به چپ مانند روند چرخش به راست گفته شده در بالا خواهد بود، با این تفاوت که برای برنامه‌ی چرخش به راست روشن شدن لامپ‌ها باید از دستورالعمل ROR استفاده شود. با استفاده از این دستورالعمل ترتیب فعال شدن خروجی‌ها $Y0 \sim Y7$ و $Y10 \sim Y17$ می‌باشد. وقتی که دکمه‌ی RESET فشرده شود، ورودی $X2 = ON$ شده و تمام ابزارهای $Y0 \sim Y17$ و $M0 \sim M11$ ریست می‌شوند. تمام لامپ‌های نئون خاموش می‌شوند. توجه داشته باشید که در این برنامه هدف از قرار دادن دستورالعمل ZRST بعد از کنتاکت‌های $X0$ و $X1$ با لبه بالارونده این است که از شروع چشمک زدن لامپ‌های نئون از $Y0$ تا $Y17$ مطمئن شویم.

3- 16 آشکارسازی محصول معیوب بر روی خط تولید

در این مثال، می‌خواهیم برنامه‌ی آشکارسازی محصول معیوب بر روی خط تولید را با استفاده از دستورالعمل SFTL بررسی نماییم.

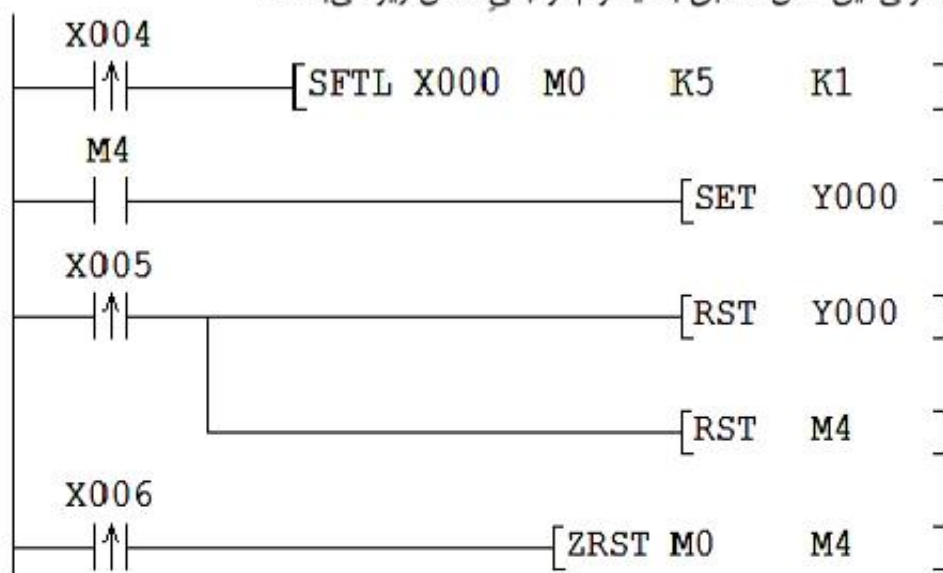


هدف از کنترل در این مثال عبارت است از :

آشکارسازی محصول معیوب در حال حرکت بر روی کانوایر توسط یک سنسور فتوالکتریک صورت می‌گیرد و بعد از آشکار شدن محصول معیوب در پنجمین موقعیت بعد از سنسور آشکارساز عیب، آن محصول معیوب در داخل یک کارتون برای کنترل مجدد محصول توسط یک جک شوت کن انداخته می‌شود. وقتی که عمل انداختن محصول معیوب در کارتون توسط یک سنسور آشکار شد، جک شوت کن ریست خواهد شد. وقتی که خطایی در روند عملکرد برنامه به وجود آمد، با فشردن دکمه RESET می‌توان بی‌نظمی عملکرد حافظه را پاک کرده و سیستم را ریستارت کرد. ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند :

ابزار	عملکرد ابزار
X0	به این ورودی PLC، سنسور فتوالکتریک برای آشکارسازی محصول معیوب متصل می‌شود. (detecting defective product)
X4	به این ورودی PLC، سنسور فتوالکتریک برای مانیتور کردن موقعیت کانوایر متصل می‌شود. (monitoring the cam)
X5	به این ورودی PLC، سنسور فتوالکتریک برای آشکارسازی افتادن محصول معیوب داخل کارتون متصل می‌شود. (detecting the falling of defective product)
X6	به این ورودی PLC، شستی RESET متصل می‌شود.
Y0	به این خروجی PLC، شیر برقی جک شوت کن محصول معیوب به داخل کارتون متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد :



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد :

با چرخیدن کانوایر، بادامکی که به غلطک متصل است، جلوی سنسور قرار می‌گیرد. با هر بار آشکارسازی بادامک توسط سنسور مانیتور کردن موقعیت کانوایر، محصول در حال انتقال بر روی کانوایر از یک موقعیت به موقعیت بعدی منتقل شده است. در این مرحله، ورودی X4 فعال شده و دستورالعمل

SFTL برای یک بار اجرا خواهد شد. محتوای (وضعیت‌های) M0~M4 برای یک بیت به سمت چپ شیفت شده و وضعیت X0 به M0 ارسال خواهد شد.

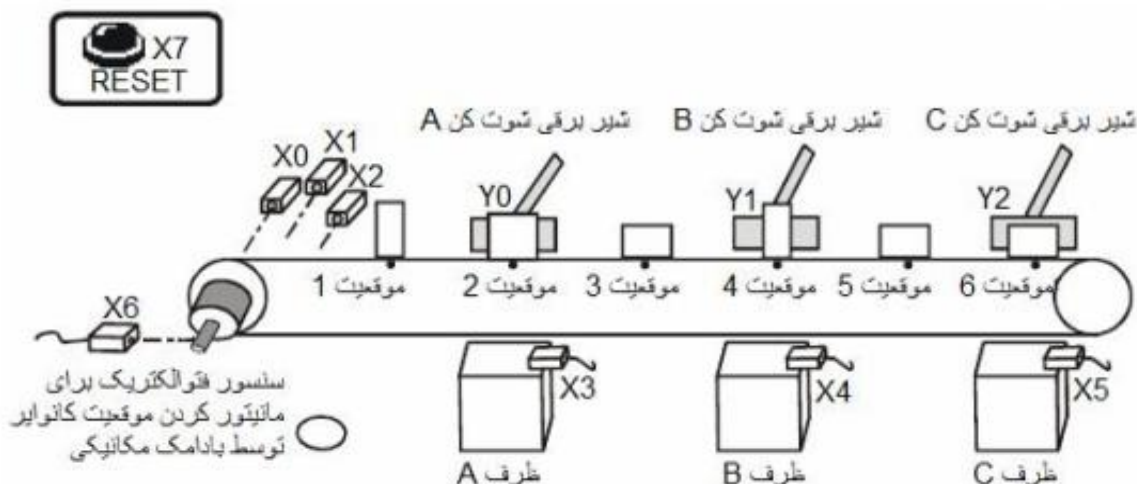
وقتی که ورودی $X0 = ON$ شد (محصول معیوب توسط سنسور آشکار سازی شد)، مقدار "1" به ابزار M0 ارسال خواهد شد و این وضعیت بعد از 4 بار شیفت به پنجمین موقعیت کانوایر خواهد رسید. از این رو، ابزار $M0 = ON$ شده و شیر برقی متصل شده به خروجی Y0 فعال خواهد شد که در اینصورت جک شوت کن برای هل دادن محصول معیوب داخل جعبه به جلو حرکت می‌کند.

وقتی که محصول معیوب داخل جعبه افتاد، ورودی X5 فعال شده و دستورالعمل‌های $[RST Y0]$ و $[RST M4]$ اجرا می‌شوند که باعث ریست شدن M0 و Y0 خواهد شد. شیر برقی غیرفعال شده و تا زمانی که محصول معیوب آشکار سازی شده به ابزار M4 منتقل شود، غیرفعال باقی می‌ماند.

وقتی که دکمه‌ی RESET فشرده شود، ورودی X6 فعال شده و ابزارهای M0~M4 ریست خواهند شد. در این حالت، وقتی که محصول معیوب آشکار شده در حافظه‌ی PLC ثبت شده باشد و همزمان عیبی در عملکرد سیستم آشکار شود، با فشردن دکمه ریست تمام حافظه ریست می‌شود. حال باید به صورت دستی تمام محصولات موجود در موقعیت 1 تا 5 را از روی کانوایر برداشت.

3- 17- مخلوط کردن اتوماتیک چند محصول

در این مثال، می‌خواهیم برنامه مخلوط کردن اتوماتیک چند محصول را با استفاده از دستورالعمل WSFL بررسی نماییم.



هدف از کنترل در این مثال عبارت است از:

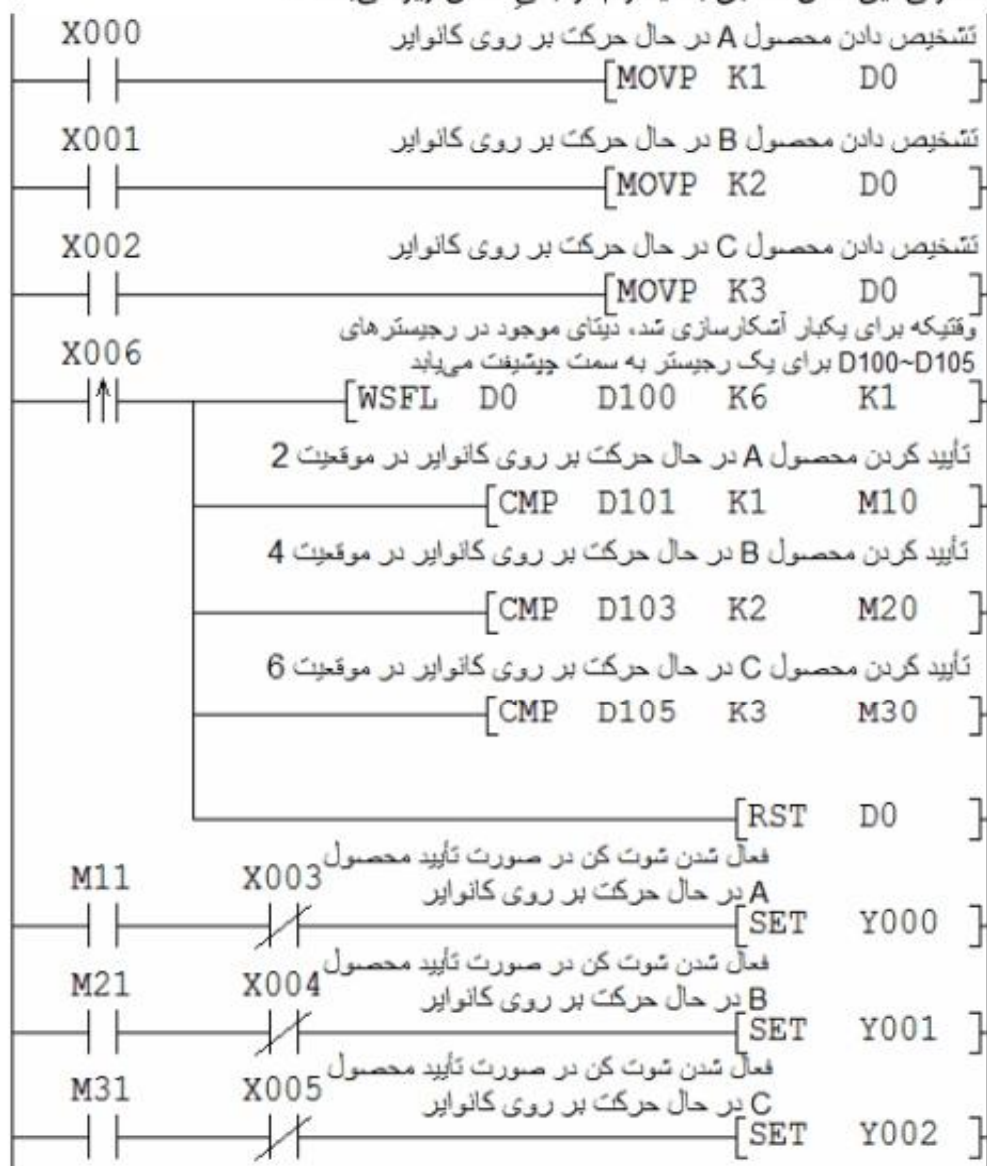
هدف ترکیب محصولات مختلف بر روی کانوایر و هل دادن هر محصول داخل ظرف مربوطه می‌باشد. سه نوع محصول A و B و C وجود دارد و برای هر محصول بر روی کانوایر 6 موقعیت وجود دارد.

وقتی که بادامک متصل به غلطک محرک کانوایر توسط سنسور مانیتور کردن موقعیت کانوایر آشکارسازی شود، به آن معنی است که محصولات روی کانوایر به اندازه‌ی یک موقعیت بر روی کانوایر به سمت جلو حرکت کرده‌اند. نوع هر محصول توسط سنسورهای ID (شناسایی و تعیین هویت) محصول شناسایی می‌شود. محصول A در حال حرکت بر روی کانوایر به ظرف A موجود در موقعیت 2 هل داده می‌شود، محصول B به ظرف B موجود در موقعیت 4 هل داده می‌شود و در نهایت، محصول C در حال حرکت به ظرف C موجود در موقعیت 6 هل داده می‌شود. وقتی که محصولات به درستی در ظرف خودشان افتادند یک سیگنال تأیید توسط سنسور موجود در بالای هر ظرف به PLC ارسال خواهد شد که باعث ریست شدن شیر برقی شوت کن مربوطه خواهد شد. وقتی که دکمه‌ی RESET فشرده شود، تمام حافظه پاک شده و سیستم تشخیص نوع محصول ریستارت خواهد شد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، سنسور ID (تعیین هویت) محصول A متصل می‌شود. وقتی که محصول A توسط این سنسور شناسایی شود، ورودی ON می‌شود.
X1	به این ورودی از PLC، سنسور ID (تعیین هویت) محصول B متصل می‌شود. وقتی که محصول B توسط این سنسور شناسایی شود، ورودی ON می‌شود.
X2	به این ورودی از PLC، سنسور ID (تعیین هویت) محصول C متصل می‌شود. وقتی که محصول C توسط این سنسور شناسایی شود، ورودی ON می‌شود.
X3	به این ورودی از PLC، سنسور افتادن محصول A در ظرف A متصل می‌شود. وقتی که محصول A در ظرف A به درستی بیفتد، ورودی ON می‌شود.
X4	به این ورودی از PLC، سنسور افتادن محصول B در ظرف B متصل می‌شود. وقتی که محصول B در ظرف B به درستی بیفتد، ورودی ON می‌شود.
X5	به این ورودی از PLC، سنسور افتادن محصول C در ظرف C متصل می‌شود. وقتی که محصول C در ظرف C به درستی بیفتد، ورودی ON می‌شود.
X6	به این ورودی از PLC، سنسور آشکارساز بادامک (Cam) متصل شده به غلطک کانوایر متصل می‌شود. وقتی که بادامک جلوی سنسور قرار گیرد، سنسور ورودی را یک بار فعال می‌کند.
X7	به این ورودی از PLC، شستی RESET متصل می‌شود. وقتی که شستی فشرده شود، ورودی ON می‌شود.
Y0	به این خروجی از PLC، شیر برقی جک شوت کن محصول A متصل می‌شود.
Y1	به این خروجی از PLC، شیر برقی جک شوت کن محصول B متصل می‌شود.
Y2	به این خروجی از PLC، شیر برقی جک شوت کن محصول C متصل می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:





عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

وقتی که محصول A در حال حرکت بر روی کانوایر شناسایی شود، ورودی X0 فعال شده و برای یک بار دستورالعمل [MOV P0 K1] اجرا می‌شود. مقدار ابزار D0 = K1 می‌شود. وقتی که محصول B و محصول C در حال حرکت بر روی کانوایر شناسایی شوند، مقدار ابزار D0 به ترتیب برابر با K2 و K3 خواهد شد.

وقتی که بادامک متصل شده به غلطک کانوایر برای یک دور بچرخد، به آن معنی است که محصولات بر روی کانوایر به اندازه‌ی یک موقعیت به سمت جلو حرکت کرده‌اند. با فعال شدن ورودی X6 دستورالعمل WSFL برای یکبار اجرا خواهد شد. داده‌های موجود در ابزارهای D100~D105 به اندازه‌ی یک رجیستر به سمت چپ شیفت خواهند یافت. در این هنگام، دستورالعمل CMP برای تأیید محصول A در موقعیت 2 (ابزار D101) اجرا شده و همچنین برای تأیید محصول B در موقعیت 4 (ابزار D103) و تأیید محصول C در موقعیت 6 (ابزار D105) دستورالعمل‌های CMP اجرا خواهد شد. بعد از هر بار اجرای دستورالعمل CMP، دستورالعمل RST برای پاک کردن محتوای داخل رجیستر D105 اجرا می‌شود.

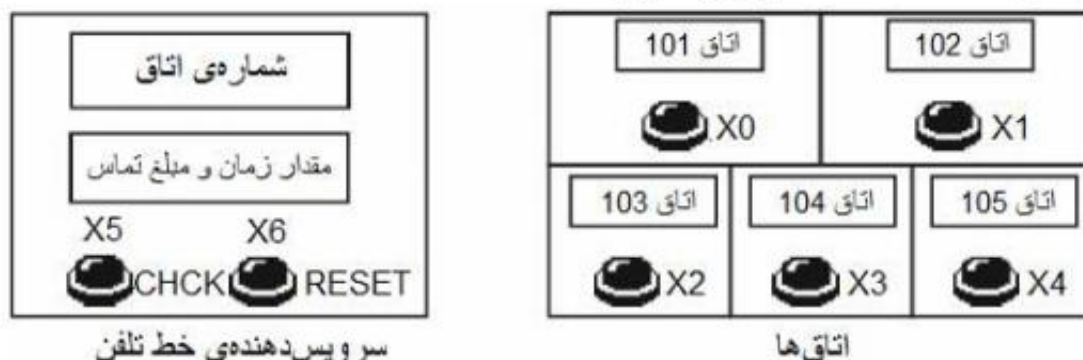
اگر محصول A و B و یا C در موقعیت‌های 2 و یا 4 و یا 6 تأیید شوند، مطابق با موقعیت جاری ابزار M11 و یا M21 و یا M31 فعال خواهد شد که این باعث فعال شدن شیر برقی A و یا B و یا C برای شوت کردن محصول داخل ظرف می‌شود.

وقتی که محصول به درستی به داخل ظرف افتاد، سنسور آشکارساز افتادن محصول داخل ظرف (ورودی X3 و یا X4 و یا X5) فعال شده و شیر برقی شوت کن A و یا B و یا C را ریست خواهد کرد.

وقتی که دکمه‌ی RESET فشرده شود ورودی $X7 = ON$ شده و دستورالعمل ZRST اجرا خواهد شد. مقادیر داخل ابزارهای D100~D105 ریست شده و برابر با 0 خواهند شد. این به آن معنی است که تمام اطلاعات داخل حافظه پاک می‌شود.

3- 18 مکالمه‌ی تلفنی اتاق‌ها توسط سرویس‌دهنده‌ی تماس

در این مثال می‌خواهیم برنامه‌ی مکالمه‌ی تلفنی اتاق‌ها را توسط سرویس‌دهنده‌ی تماس با استفاده از دستورالعمل SFWR/SFRD بررسی نماییم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

می‌خواهیم با استفاده از اصل ترتیب ورود تماس‌ها شماره تلفن اتاق را در هنگام تماس تلفنی ثبت کرده و میزان (زمان) مکالمه را چک کرده و ثبت نماییم. این بدان معنی است که اولین سرویس مکالمه به نخستین اتاق تماس گیرنده داده خواهد شد.

وقتی که دکمه RESET فشرده شود، تمام اطلاعات داخل حافظه‌ی PLC پاک خواهد شد. با هر بار فشردن دکمه‌های Call موجود در اتاق‌ها تعداد تماس‌های گرفته شده افزایش یافته و با هر بار فشردن دکمه‌ی CHECK موجود در اتاق سرویس‌دهنده‌ی تماس از این تعداد تماس‌های گرفته شده کاسته خواهد شد. اگر شماره‌ی تماس‌های تمام اتاق‌ها چک شود، تعداد تماس‌های گرفته شده برابر با 0 خواهد شد.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
X0	به این ورودی PLC، دکمه Call اتاق 101 متصل می‌شود. وقتی که این شستی فشرده شود، ورودی ON می‌شود
X1	به این ورودی PLC، دکمه Call اتاق 102 متصل می‌شود. وقتی که این شستی فشرده شود، ورودی ON می‌شود
X2	به این ورودی PLC، دکمه Call اتاق 103 متصل می‌شود. وقتی که این شستی فشرده شود، ورودی ON می‌شود
X3	به این ورودی PLC، دکمه Call اتاق 104 متصل می‌شود. وقتی که این شستی فشرده شود، ورودی ON می‌شود
X4	به این ورودی PLC، دکمه Call اتاق 105 متصل می‌شود. وقتی که این شستی فشرده شود، ورودی ON می‌شود
X5	به این ورودی PLC، دکمه‌ی Check متصل می‌شود. وقتی که این شستی فشرده شود، ورودی ON می‌شود.
X6	به این ورودی PLC، دکمه‌ی RESET متصل می‌شود. وقتی که این شستی فشرده شود، ورودی ON می‌شود.
D0	در این ابزار میزان زمان یا مبلغ تماس نمایش داده می‌شود.
D1~D5	از این ابزارها برای ذخیره کردن شماره‌ی اتاق‌های زیر چک استفاده می‌شود.
D10	از این ابزار برای ذخیره‌سازی موقت شماره‌ی اتاق ورودی استفاده می‌شود.
D11	نمایش شماره اتاق (به ترتیب ورود)

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر می‌باشد:



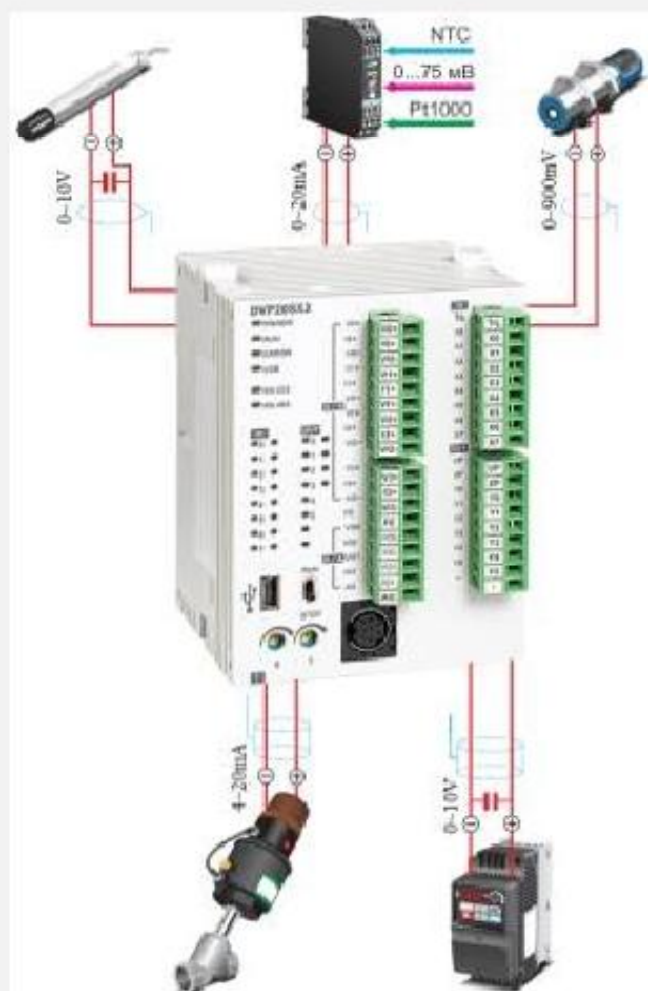
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

با استفاده از دستورالعمل‌های SFWR و SFRD، برنامه عمل کنترل نوشتن و خواندن داده‌های موجود در پشته را با استفاده از قاعده‌ی ترتیب ورودی FIFO (first in, first out) به ترتیب از اول تا آخر انجام می‌دهد. در این مثال، ابتدا شماره‌ی اولین اتاق تماس گرفته شده بررسی (چک) خواهد شد.

وقتی که شستی‌های Call فشرده شوند، ابتدا شماره‌ی اتاق در ابزار D10 ذخیره شده و سپس شماره‌ی اتاق توسط ابزار D0 به ابزارهای D1~D5 مطابق با شماره‌ی اتاق جاری ارسال خواهد شد. وقتی که دکمه‌ی CHECK فشرده شود، ابتدا نخستین شماره تماس گرفته شده در ابزار D11 خوانده شده و سپس میزان (تعداد، مقدار) تماس‌های گرفته شده مطابق با مقدار موجود در ابزار D0 کاهش خواهد یافت. با استفاده از یک واسط گرافیکی یا متنی (Text Panel یا HMI) می‌توان مقدار موجود در ابزار D0 (میزان یا تعداد تماس‌ها) و ابزار D11 (نمایش شماره‌ی اتاق) مانیتور کرد. برنامه با استفاده از دستورالعمل‌های ZRST و RST مقادیر موجود در ابزارهای D0~D5 را پاک خواهد کرد. این به آن معنی است که تعداد تماس‌ها و شماره‌ی اتاق‌ها در واسط بین انسان و ماشین (HMI یا Text Panel 7-seg) پاک شده و برابر با 0 خواهد شد.

فصل 4

مثال‌های کاربردی وابسته به محیط‌های کنترل پیوسته آنالوگ



4-1 دستورالعمل‌های وابسته به مازول آنالوگ به همراه مثال

4-1-1 دستورالعمل شیب متغیر RAMP

API	Mnemonic	Operands																Function																													
67	RAMP	$(S_1) (S_2) (D) (n)$																Ramp Variable Value																													
Type OP		Bit Devices								Word Devices																Program Steps RAMP: 9 steps																					
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F																															
S ₁																	*																														
S ₂																	*																														
D																	*																														
n							*	*																																							
PULSE																16-bit																32-bit															
ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2																		

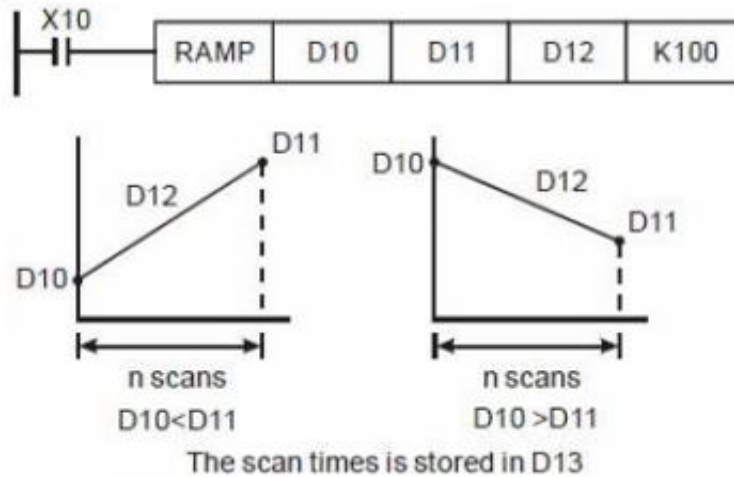
این دستورالعمل از چهار عملوند زیر تشکیل شده است:

- عملوند S_1 : در این عملوند نقطه‌ی شروع سیگنال شیب تنظیم می‌شود.
- عملوند S_2 : در این عملوند نقطه‌ی پایان سیگنال شیب تنظیم می‌شود.
- عملوند D: در این عملوند مدت زمان یا فاصله‌ی سیگنال شیب تنظیم می‌شود.
- عملوند n: در این عملوند باید زمان اسکن تنظیم شود. محدوده‌ی قابل تنظیم برای عملوند n بین 1~32767 می‌باشد.

عملوند D دو نقطه‌ی متوالی را اشغال می‌کند، منظور از دو نقطه دو ابزار (رجیستر) می‌باشد. از فلگ خاص M1062 برای فعال کردن تابع RAMP و از فلگ خاص M1029 برای اعلام تکمیل شدن (به پایان رسیدن) اجرا دستورالعمل RAMP استفاده می‌شود. از این دستورالعمل (RAMP) برای بدست آوردن یا ساختن شیب استفاده می‌شود. در این دستورالعمل یک وابستگی بین خطی بودن شیب و زمان اسکن وجود دارد. قبل از استفاده از این دستورالعمل، ابتدا باید شما زمان اسکن را تنظیم نمایید.

مثال:

می‌خواهیم از این دستورالعمل برای کنترل شیب سیگنال خروجی آنالوگ استفاده نماییم. وقتی که از این دستورالعمل برای سیگنال خروجی آنالوگ استفاده شود، این دستورالعمل وابسته به بافر START و STOP خواهد شد. منظور از بافر START همان عملوند S_1 و بافر STOP همان عملوند S_2 است.



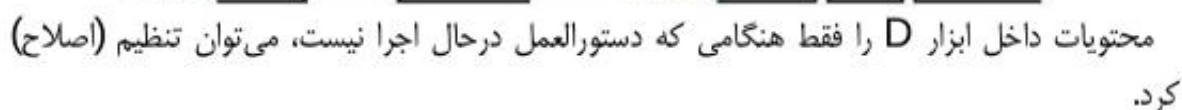
در این مثال باید مقدار شروع سیگنال شیب را در رجیستر D10 و مقدار پایین سیگنال شیب را در رجیستر D11 تنظیم نمایید. این دو رجیستر را می‌توان در یک HMI قرار داده و کاربر (اپراتور) خودش متناسب با نیاز این دو مقدار را تنظیم نماید.

وقتی که $X10=ON$ شود، مقداری که در رجیستر D10 ذخیره شده به سمت مقداری که در رجیستر D11 ذخیره شده در مدت زمان اسکن $n=100$ خواهد رفت و نتیجه‌ی این حرکت شیب‌دار در رجیستر D12 ذخیره می‌شود. در همین هنگام زمان اسکن در رجیستر D13 ذخیره می‌شود. همانطور که گفته شد، عملوند D دو رجیستر را اشغال می‌کند که رجیستر اولی میزان شیب بوده و رجیستر دوم مدت زمان اسکن سپری شده می‌باشد.

در این برنامه، برای ثابت کردن زمان از فلگ خاص M1039 استفاده شده است. وقتی که $M1039=ON$ شود، زمان ثابت نگهداشته می‌شود. از دستورالعمل MOV برای نوشتن یک مدت زمان ثابت در داخل رجیستر خاص D1039 استفاده شده است. فرض کنید که مدت زمان اسکن برابر با 30ms می‌باشد و $n=100$ است. از این رو، مدت زمان افزایش شیب از مقدار تنظیم شده در D10 به مقدار تنظیم شده در D11 در عرض 3 ثانیه ($30ms \times 100 = 3s$) می‌باشد.

وقتی که $X10=OFF$ شود، اجرای دستورالعمل RAMP متوقف خواهد شد. وقتی که مجدداً $X10=ON$ شود، مقدار ذخیره شده در رجیستر Dn (در این مثال D13 می‌باشد) برای محاسبه مجدد ریست شده و صفر می‌شود.

وقتی که $M1026=OFF$ شود، فلگ خاص M1029 فعال (ON) خواهد شد و محتویات داخل رجیستر D12 ریست خواهد شد و مقدار آن برابر با مقدار تنظیم شده در رجیستر D10 می‌شود. دقت داشته باشید که از رجیستر Dn (در این مثال D13 می‌باشد) برای فعال و غیرفعال کردن (ON/OFF) فلگ خاص M1026 استفاده شده است.



4-1-2 دستورالعمل خواندن دیتای رجیسترهای کنترلی (CR#) از ماژول‌های توسعه‌ی خاص (FROM)

این دستورالعمل از چهار عملوند زیر تشکیل شده است:

- عملوند m_1 : که در این عملوند شماره ID ماژول توسعه‌ی خاص متصل شده به CPU نوشته می‌شود.

- عملوند m_2 : که در این عملوند شماره رجیستر CR# مازول توسعه‌ی خاص برای خواندن مقدار آن تنظیم می‌شود.

- عملوند D: که در این عملوند شماره رجیستر D جهت ذخیره کردن مقدار خوانده شده از رجیستر کنترلی CR# ماژول توسعه‌ی خاص نوشته می‌شود.

- عملوند n: که در این عملوند تعداد داده برای خواندن (انتقال) در یک زمان تنظیم می‌شود. محدوده‌ی قابل تنظیم برای عملوند m_1 (16 بیتی و 32 بیتی) در CPUهای سری ES, SA برابر با 0~7 می‌باشد، برای CPUهای سری EH2, EH3, SV2, برابر با 0~255 بوده و برای

CPU سری SV برابر با 0~107 (منظور از 0~107 این است که ما می‌توانیم هشت ماژول توسعه‌ی خاص را از 0~7 به سمت راست CPU متصل نموده و هشت ماژول توسعه‌ی خاص را از شماره‌ی 100~107 به سمت چپ CPU متصل نماییم) است. دقت داشته باشید که CPU‌های نام برده شده فقط برای مثال بوده و تعداد CPU‌ها بیشتر می‌باشد. بنابراین برای تشخیص CPU‌ها باید به پورت توسعه‌ی آن دقت کنید. برخی از CPU‌ها تنها در سمت راست خود پورت توسعه دارند و برخی دیگر در هر دو سمت راست و چپ خود از پورت توسعه پشتیبانی می‌کنند و در نهایت CPU‌های سری E در دو سمت راست و چپ و همچنین بر روی خود از اتصال ماژول‌های توسعه‌ی خاص پشتیبانی می‌کنند.

محدوده‌ی قابل تنظیم برای عملوند m_2 (دستورالعمل 16 بیتی و 32 بیتی) در CPU‌های سری ES, SA برابر با 0~48 می‌باشد، برای CPU سری EH برابر با 0~254 و برای CPU‌های سری EH2, SV, EH3, SV2 برابر با 0~499 است. همانطور که متوجه شدید، هر یک از ماژول‌های توسعه‌ی خاص دارای تعداد رجیستر کنترلی می‌باشند که منظور از اعداد گفته شده در این قسمت برای عملوند m_2 همان تعداد رجیسترهای کنترلی ماژول می‌باشد.

محدوده‌ی قابل تنظیم برای عملوند n در دستورالعمل 16 بیتی برای CPU‌های سری ES, SA برابر با $1 \sim (49 - m_2)$ می‌باشد، برای CPU سری EH برابر با $1 \sim (255 - m_2)$ و برای CPU‌های سری EH2, SV, EH3, SV2 برابر با $1 \sim (500 - m_2)$ است.

همچنین، محدوده‌ی قابل تنظیم برای عملوند n در دستورالعمل 32 بیتی برای CPU‌های سری ES, SA برابر با $1 \sim (49 - m_2)/2$ می‌باشد، برای CPU‌های سری EH برابر با $1 \sim (255 - m_2)/2$ و برای CPU‌های سری EH2, SV, EH3, SV2 برابر با $1 \sim (500 - m_2)/2$ است.

برای اطلاعات بیشتر در مورد مقادیر ذکر شده در بالا به دفترچه‌ی راهنمای ماژول‌های توسعه‌ی خاص مورد نظر خود رجوع نمایید. در فصل‌های بعدی ما تمام ماژول‌های توسعه خاص آنالوگ را کاملاً توضیح داده‌ایم و در رابطه با تمام جزئیات آنها بحث کرده‌ایم.

عملوند m_1 , m_2 و n در CPU‌های سری EH از یک رجیستر (Word) یا 16 بیتی D پشتیبانی نمی‌کند.

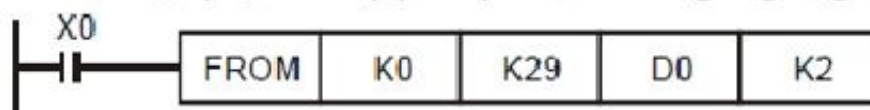
وقتی که وقفه‌ای در هنگام عملکرد دستورالعمل FROM/TO بوجود آید، فلگ خاص M1083 فعال (ON) می‌شود. از دستورالعمل FROM برای خواندن داده‌ی موجود در رجیسترهای کنترلی (CR) ماژول‌های توسعه‌ی خاص از قبیل ماژول‌های آنالوگ، دما، شمارنده‌ی سرعت بالا استفاده می‌شود. همانطور که می‌دانید، ماژول‌های توسعه‌ی خاص هر کدام دارای یک پردازشگر مستقل از

CPU می‌باشند که جهت تنظیمات دارای چندین رجیستر کنترلی هستند. برخی از این رجیسترهای کنترلی فقط خواندنی هستند. برای خواندن این رجیسترهای کنترلی از ماژول‌های توسعه‌ی خاص توسط CPU اصلی از دستورالعمل FROM استفاده می‌شود.

به دستورالعمل 16 بیتی FROM می‌توان $D=K1 \sim K4$ پو به دستورالعمل 32 بیتی FROM می‌توان $D=K1 \sim K8$ را اختصاص داد.

مثال

در این مثال می‌خواهیم مقدار رجیستر کنترلی CR#29 از ماژول توسعه‌ی خاص شماره 0 را خوانده و سپس مقدار خوانده شده را در رجیستر D0 ذخیره کنیم. همچنین، می‌خواهیم مقدار رجیستر کنترلی CR#30 را از همین ماژول خاص خوانده و سپس مقدار خوانده شده را در رجیستر D1 ذخیره نماییم. همانطور که متوجه شدید، در این مثال فقط دو گروه از داده‌ی ماژول توسعه خاص به صورت همزمان خوانده می‌شود. این به آن معنی است که باید عملوند n برابر با 2 تنظیم شود ($n=2$).



وقتی که $X0=ON$ شود، دستورالعمل FROM شروع به اجرا خواهد کرد. در این حالت تغییرات مقدار رجیستر کنترلی خوانده می‌شود. همچنین، وقتی که $X0=OFF$ شود، دستورالعمل FROM اجرا نشده و تغییرات داده‌ی رجیستر کنترلی تنظیم شده خوانده نخواهد شد.

4-1-3 دستورالعمل نوشتن داده در رجیستر کنترلی (CR) ماژول‌های توسعه‌ی خاص (TO)

API	Mnemonic			Operands				Function															
79	D	TO	P	(m1)	(m2)	S	n	Write CR Data into Special Modules															
OP	Type	Bit Devices				Word Devices												Program Steps					
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		TO, TOP: 9 steps					
	m1					*	*								*			DTOP: 17 steps					
	m2					*	*								*								
	S					*	*	*	*	*	*	*	*	*	*	*	*						
	n					*	*								*								

PULSE										16-bit										32-bit									
ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2

این دستورالعمل شامل 4 عملوند زیر می‌باشد:

- عملوند m_1 : در این عملوند شماره‌ی ID ماژول توسعه‌ی خاص از قبیل ماژول توسعه آنالوگ، دما، شمارنده سرعت و ... متصل شده به CPU را وارد می‌نماییم.

- عملوند m_2 : در این عملوند شماره‌ی رجیستر کنترلی CR# مازول توسعه‌ی خاص انتخاب شده جهت نوشتن داده در آن رجیستر کنترلی را می‌نویسیم.
- عملوند S: در این عملوند باید مقدار داده‌ای را که باید در رجیستر کنترلی CR نوشته شود را تنظیم نماییم.
- عملوند n: در این عملوند تعداد دیتایی را که می‌خواهیم به صورت همزمان با هم بنویسیم را تنظیم می‌نماییم.

محدوده‌ی قابل تنظیم برای عملوند m_1 در دستورالعمل 16 بیتی و 32 بیتی TO برای CPU‌های سری ES, SA برابر با 0~7، برای CPU‌های سری EH, EH2, EH3, SV2 برابر با 0~255 و برای CPU‌های سری SV برابر با 0~107 می‌باشد. منظور از 0~107 این است که ما می‌توانیم هشت مازول توسعه‌ی خاص را از 0~7 به سمت راست CPU متصل کرده و هشت مازول توسعه‌ی خاص را از شماره‌ی 100~107 به سمت چپ CPU متصل نماییم. دقت داشته باشید که CPU‌های نام برده شده فقط برای مثال بوده و تعداد CPU‌ها بیشتر می‌باشد. بنابراین برای تشخیص CPU‌ها باید به پورت توسعه‌ی آن دقت کنید. برخی از CPU‌ها تنها در سمت راست خود پورت توسعه دارند و برخی دیگر در هر دو سمت راست و چپ خود از پورت توسعه پشتیبانی می‌کنند و در نهایت، CPU‌های سری E در دو سمت راست و چپ و همچنین بر روی خود از اتصال مازول‌های توسعه‌ی خاص پشتیبانی می‌کنند.

محدوده‌ی قابل تنظیم برای عملوند m_2 در دستورالعمل 16 بیتی و 32 بیتی TO برای CPU‌های سری ES, SA برابر با 0~48، برای CPU‌های سری EH برابر با 0~254 و برای CPU‌های سری EH2, SV, EH3, SV2 برابر با 0~499 می‌باشد. همانطور که متوجه شدید، هر یک از مازول‌های توسعه‌ی خاص دارای تعداد رجیستر کنترلی می‌باشند که منظور از اعداد گفته شده در این قسمت برای عملوند m_2 همان تعداد رجیسترهای کنترلی مازول می‌باشد.

محدوده‌ی قابل تنظیم برای عملوند n در دستورالعمل 16 بیتی TO برای PU‌های سری ES, SA برابر با $1-(49-m_2)$ ، برای CPU‌های سری EH برابر با $1-(255-m_2)$ می‌باشد و برای CPU‌های سری EH2, SV, EH3, SV2 برابر با $1-(255-m_2)$ می‌باشد.

همچنین، محدوده‌ی قابل تنظیم برای عملوند n در دستورالعمل 32 بیتی برای CPU‌های سری ES, SA برابر با $1-(49-m_2)/2$ ، برای CPU‌های سری EH برابر با $1-(255-m_2)/2$ و برای CPU‌های سری EH2, SV, EH3, SV2 برابر با $1-(255-m_2)/2$ است.

عملوندهای m_1 و m_2 و n در CPU های سری EH ابزارهای از نوع رجیستر داده‌ی Word 16) بی‌تی D را پشتیبانی نمی‌کنند. چنانچه در هنگام عمل کردن دستورالعمل FROM/TO یک وقفه اجرا شود، فلگ خاص M1083 فعال (ON) خواهد شد.

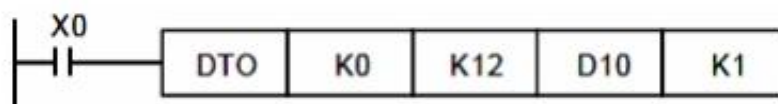
از دستورالعمل TO جهت نوشتن یک مقدار عددی در داخل رجیستر کنترلی CR تعیین شده در ماژول توسعه‌ی خاص از قبیل ماژول ورودی خروجی آنالوگ، دما، شمارنده‌ی سرعت بالا و ... استفاده می‌شود.

به دستورالعمل 16 بی‌تی TO می‌توان $S=K1\sim K4$ و به دستورالعمل 32 بی‌تی TO می‌توان $S=K1\sim K8$ را اختصاص داد.

مثال 1

در این مثال، می‌خواهیم با استفاده از دستورالعمل 32 بی‌تی DTO محتویات داخل رجیسترهای D10 و D11 را در داخل رجیسترهای کنترلی CR#12 و CR#13 نخستین ماژول توسعه خاص (نزدیک‌ترین ماژول توسعه به CPU) که شماره آن 0 است را بنویسیم. این ماژول یک ماژول شمارنده‌ی سرعت بالا می‌باشد.

همانطور که متوجه شدید، در این مثال از یک دستورالعمل 32 بی‌تی DTO استفاده شده و مقادیر دو رجیستر 16 بی‌تی D10 و D11 را می‌خواهیم در ماژول توسعه‌ی خاص شمارنده‌ی سرعت بالا بنویسیم. از این رو، ما فقط از یک گروه داده برای نوشتن به صورت همزمان در ماژول توسعه خاص استفاده می‌کنیم ($n=1$).



وقتی که ورودی $X0=ON$ شود، دستورالعمل 32 بی‌تی DTO اجرا خواهد شد. وقتی که ورودی $X0=OFF$ شود، دستورالعمل DTO اجرا نخواهد شد و عمل نوشتن داده برای تغییر مقادیر رجیسترهای کنترلی متوقف می‌شود.

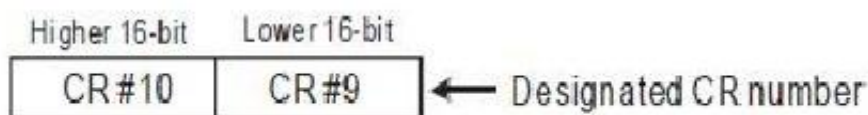
قاعده عملکرد این دستورالعمل به شرح زیر می‌باشد:

در عملوند m_1 شماره‌ی ماژول توسعه‌ی خاص که به CPU متصل شده است را می‌نویسیم. در این مثال، ماژول توسعه خاص، نخستین ماژول می‌باشد که به CPU متصل شده است. منظور از نخستین ماژول توسعه خاص متصل شده به CPU، نزدیک‌ترین ماژول توسعه‌ی متصل شده به CPU است. از این رو، نزدیک‌ترین ماژول توسعه خاص متصل شده به CPU شماره‌ی 0 را به خود اختصاص می‌دهد، همچنین، به هر دو سمت برخی از CPU های دلتا می‌توان تنها 8 ماژول توسعه‌ی خاص را متصل کرد.

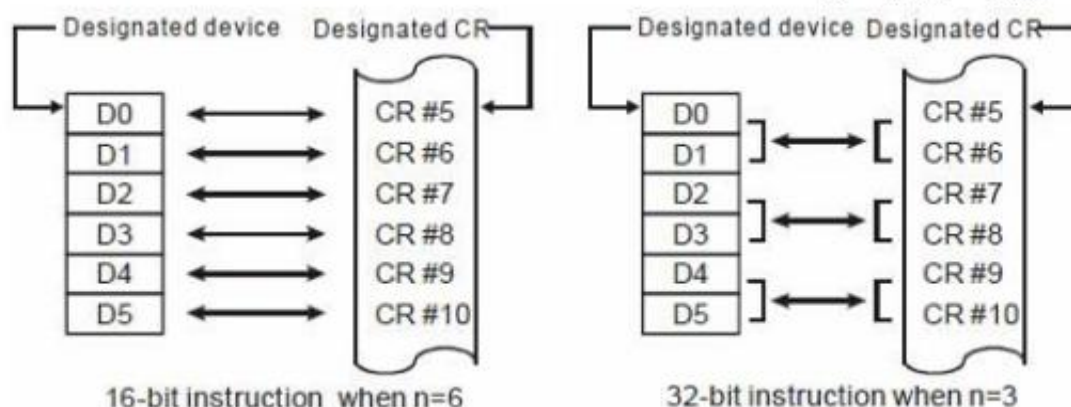
بنابراین اگر هر 16 ماژول توسعه‌ی خاص به CPU متصل شده باشند، دورترین ماژول توسعه خاص متصل شده به سمت راست CPU شماره‌ی 7 را به خود اختصاص می‌دهد. در این مثال، تنها یک ماژول توسعه‌ی خاص به CPU متصل شده است، بنابراین به این عملوند شماره‌ی 0 را باید اختصاص داد. دقت داشته باشید که این ماژول‌های توسعه‌ی خاص هیچ نقطه‌ی ورودی خروجی I/O دیجیتال را اشغال نخواهد کرد.

در عملوند m_2 باید شماره‌ی CR (رجیستر کنترلی CR#) را وارد نماییم. رجیسترهای کنترلی CR در برخی از ماژول‌ها 16 بیتی بوده و در برخی دیگر از ماژول‌ها 32 بیتی می‌باشند و هر کدام بیانگر یک صف خاص از ماژول هستند. از این رو، هر رجیستر کنترلی CR دارای یک شماره‌ی از 0 تا n می‌باشد. منظور از n شماره آخرین رجیستر کنترلی است که در ماژول‌های مختلف این شماره نیز متفاوت است. رجیسترهای کنترلی CR حافظه‌دار می‌باشند و با قطع تغذیه ماژول مقدار موجود در آنها حذف نخواهند شد. دقت داشته باشید که تمام تنظیمات وابسته به ماژول توسعه‌ی خاص و تمام وضعیت‌های عملکرد جاری ماژول توسعه‌ی خاص در رجیسترهای کنترلی CR قرار می‌گیرند.

از دستورالعمل‌های FROM/TO برای Reading/Writing (خواندن و نوشتن) یک رجیستر کنترلی CR در یک زمان استفاده می‌شود. از دستورالعمل‌های DFROM/DTO برای Reading/Writing (خواندن و نوشتن) یک رجیستر کنترلی 32 بیتی CR در یک زمان استفاده می‌شود.



تعداد گروه رجیسترهای کنترلی جهت مخابره‌ی دیتا (انتقال داده) در عملوند n تنظیم می‌شود. دقت داشته باشید که $n=2$ در دستورالعمل 16 بیتی و $n=1$ در دستورالعمل 32 بیتی هر دو به یک معنی می‌باشند و با یکدیگر یکسان هستند.



CPU های سری ES, EX, SS از فلگ خاص M1083 پشتیبانی نمی‌کنند. در مدتی که دستورالعمل FROM/TO در حال اجرا می‌باشد، از اجرای تمام زیرروال‌های وقفه‌ای داخلی و خارجی جلوگیری می‌شود. تنها زمانی که اجرای دستورالعمل‌های FROM/TO به پایان برسد، وقفه‌ها قابل اجرا می‌باشند. همچنین از دستورالعمل‌های FROM/TO می‌توان در زیرروال‌های وقفه نیز استفاده کرد.

از فلگ خاص M1083 برای سوئیچ کردن مدهای دستورالعمل در CPU های SA, SX, SC, SA2, SX2, SC2, EH, EH2, SV, EH3, SV2 استفاده می‌شود. وقتی که M1083=OFF باشد، در مدت زمان اجرای دستورالعمل FROM/TO، از اجرای تمام زیرروال‌های وقفه‌ای داخلی و خارجی جلوگیری می‌شود. بعد از این که اجرای دستورالعمل‌های FROM/TO به پایان رسید می‌توان از وقفه‌ها استفاده کرد. همچنین، از دستورالعمل‌های FROM/TO می‌توان در زیرروال‌های وقفه نیز استفاده کرد.

وقتی که فلگ خاص M1083=ON باشد و در مدت زمان اجرای دستورالعمل FROM/TO یک سیگنال وقفه رخ دهد، با یک تأخیر 10 میکروثانیه‌ای ابتدا وقفه پردازش می‌شود و اجرای دستورالعمل FROM/TO متوقف خواهد شد. بعد از اینکه اجرای زیرروال وقفه به پایان رسید، برنامه به دستورالعمل FROM/TO پرش خواهد کرد. در این حالت، اگر M1083=ON باشد، از دستورالعمل FROM/TO نمی‌توان در برنامه‌ی زیرروال وقفه استفاده کرد.

مثال 2

در این مثال، می‌خواهیم با استفاده از دستورالعمل FROM/TO منحنی تبدیل A/D (آنالوگ به دیجیتال) را برای ماژول ورودی آنالوگ DVP04AD تنظیم نماییم. مقدار آفست (OFFSET) کانال 1 برابر با $0V (=K0_{LSB})$ و مقدار ضریب تقویت (GAIN) برابر با $2.5V (=K2000_{LSB})$ تنظیم شده است.



مقدار H0 را داخل رجیستر کنترلی CR#1 نخستین ماژول ورودی آنالوگ با شماره کارت 0 نوشته و کانال 1 ماژول ورودی آنالوگ را برابر با مد 0 (ولتاژ ورودی بین $-10V \sim +10V$) تنظیم کرده‌ایم.

مقدار H0 را داخل رجیستر کنترلی CR#33 برای تنظیم OFFSET/GAIN کانال‌های 1 تا 4 می‌نویسیم.

وقتی که ورودی X0 از OFF به ON تغییر وضعیت دهد، مقدار $K0_{LSB}$ برای آفست (OFFSET) در رجیستر کنترلی CR#18 و مقدار $K2000_{LSB}$ برای ضریب تقویت (GAIN) در داخل رجیستر کنترلی CR#24 نوشته می‌شود.

مثال 3

در این مثال، می‌خواهیم با استفاده از دستورالعمل FROM/TO منحنی تبدیل آنالوگ به دیجیتال (A/D) برای مازول ورودی آنالوگ DVP-04AD را تنظیم نماییم. مقدار آفست (OFFSET) کانال 2 را برابر با $2mA (=K400_{LSB})$ و مقدار ضریب تقویت (GAIN) برابر با $18mA (=K3600_{LSB})$ تنظیم می‌نماییم.



مقدار H18 را داخل رجیستر کنترلی CR#1 نخستین مازول ورودی آنالوگ با شماره کارت توسعه خاص 0 نوشته و کانال 3 را برابر با مد 3 (جریان ورودی بین $-20mA \sim +20mA$) تنظیم می‌نماییم. مقدار H0 را در داخل رجیستر کنترلی CR#33 برای تنظیم OFFSET/GAIN tuning کانال‌های 1 تا 4 را می‌نویسیم.

وقتی که ورودی X0 از OFF به ON تغییر وضعیت داد، مقدار آفست برابر با $K400_{LSB}$ شده و در داخل رجیستر کنترلی CR#19 نوشته می‌شود و همچنین، مقدار ضریب تقویت برابر با $K3600_{LSB}$ شده و در داخل رجیستر کنترلی CR#25 نوشته می‌شود.

مثال 4

در این مثال، می‌خواهیم با استفاده از دستورالعمل FROM/TO منحنی تبدیل دیجیتال به آنالوگ (D/A) برای مازول خروجی آنالوگ DVP02DA را تنظیم نماییم. مقدار آفست (OFFSET) کانال 2 را برابر با $0mA (=K0_{LSB})$ و مقدار ضریب تقویت (GAIN) را برابر با $10mA (=K1000_{LSB})$ تنظیم می‌نماییم.



ابتدا مقدار H18 را در داخل رجیستر کنترلی CR#1 ماژول خروجی آنالوگ متصل شده به CPU با شماره 1 (دومین ماژول توسعه‌ی خاص متصل شده به CPU) نوشته و کانال 1 این ماژول را برابر با مد 3 (جریان خروجی 0mA~20 mA) تنظیم می‌نماییم.

در مرحله‌ی دوم، ضریب تقویت و آفست برای کانال 1 و 2 این ماژول را برابر با 0 قرار می‌دهیم. با نوشتن مقدار هگز H0 در داخل رجیستر کنترلی CR#33 این امر ممکن است.

وقتی که ورودی X0 از OFF به ON تغییر وضعیت دهد، در داخل رجیستر کنترلی CR#22 (آفست (OFFSET) مقدار K0_{LSB} نوشته شده و در داخل رجیستر کنترلی CR#28 (ضریب تقویت (GAIN) مقدار دسیمال K1000_{LSB} نوشته می‌شود.

مثال 5

در این مثال، می‌خواهیم با استفاده از دستورالعمل‌های FROM/TO منحنی تبدیل دیجیتال به آنالوگ (D/A) ماژول مقدار خروجی آنالوگ DVP02DA را تنظیم نماییم. در این مثال، باید مقدار آفست (OFFSET) کانال 2 برابر با 2mA (=K400_{LSB}) و مقدار ضریب تقویت (GAIN) همین کانال برابر با 18mA (=K3600_{LSB}) تنظیم شود.

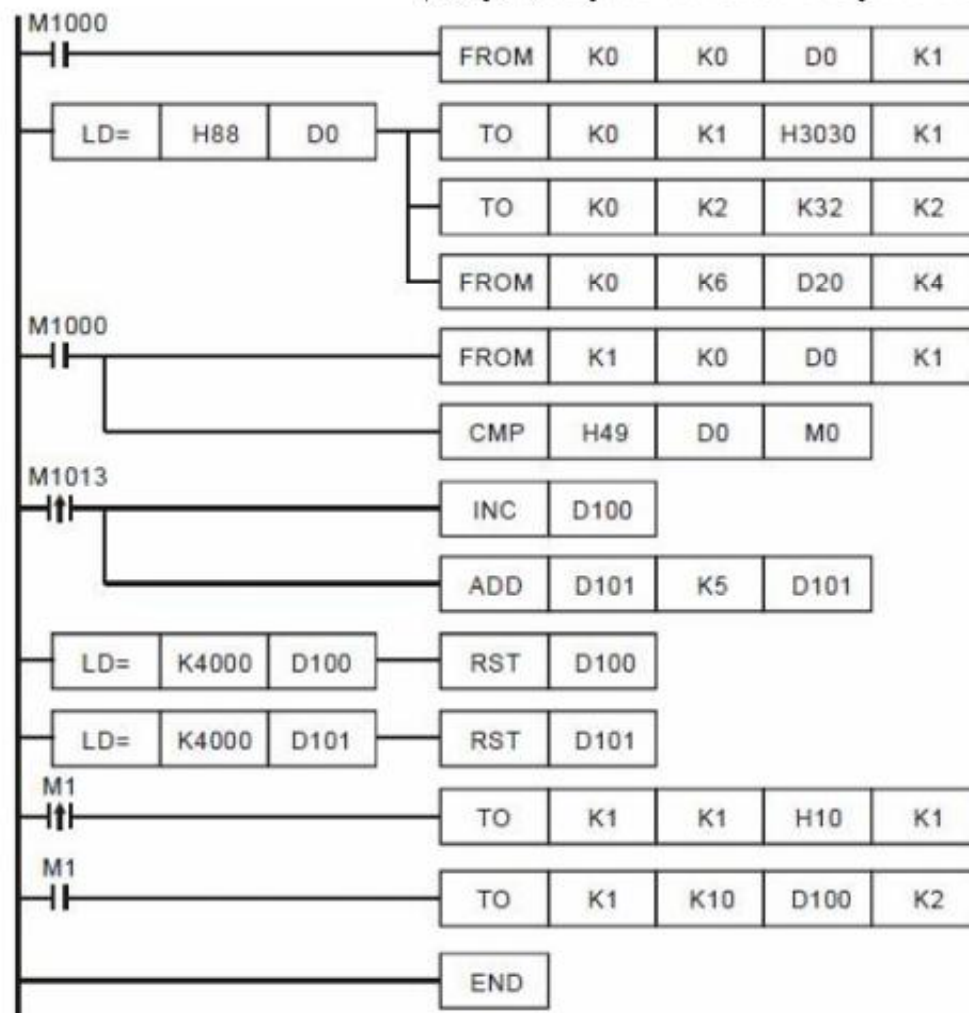


ابتدا باید مقدار H10 را در داخل رجیستر کنترلی CR#1 ماژول خروجی آنالوگ متصل شده به CPU با شماره‌ی 1 (دومین ماژول توسعه‌ی خاص متصل شده به CPU) نوشته و کانال 2 همین ماژول را برابر با مد 2 (جریان خروجی 4mA~20mA) تنظیم نماییم.

در مرحله دوم، ضریب تقویت و آفست برای کانال‌های 1 و 2 این ماژول را برابر با 0 قرار داده‌ایم. این امر با نوشتن مقدار هگز H0 در داخل رجیستر کنترلی CR#33 ممکن شده است. وقتی که ورودی X0 از OFF به ON تغییر وضعیت دهد، در داخل رجیستر کنترلی CR#33 (آفست (OFFDET مقدار $K400_{LSB}$ نوشته شده و در داخل رجیستر کنترلی CR#29 (ضریب تقویت (GAIN مقدار دسیمال $K3600_{LSB}$ نوشته می‌شود.

مثال 6

در این مثال، می‌خواهیم با استفاده از دستورالعمل‌های FROM/TO مقادیر را از/ به ماژول‌های DVP02DA-S و DVP04AD-S خوانده یا بنویسیم.



ابتدا می‌خواهیم بررسی کنیم که ماژول متصل شده به CPU ماژول مدل DVP04AD-S می‌باشد، یا خیر. هر ماژول توسعه برای شناسایی دارای یک شماره ID منحصر بفرد می‌باشد که این شماره در ماژول توسعه‌ی DVP04AD-S برابر با H'88 هگز است. یعنی بعد از اتصال ماژول

توسعه به CPU، این ماژول مقدار هگز H'86 را به CPU ارسال می‌کند. برای اطمینان از درست بودن ماژول، این مقدار (H'88) را با استفاده از یک مقایسه گر LD= بررسی می‌نماییم. در صورت درست بودن، اجازه‌ی دسترسی به برنامه‌های بعد به وجود می‌آید. بعد از درست بودن ماژول متصل شده به CPU می‌خواهیم مقدار موجود در رجیستر کنترلی CR#0 همین ماژول توسعه را بخوانیم. این ماژول نخستین ماژول توسعه‌ی خاص می‌باشد که به CPU متصل شده است. بنابراین شماره‌ی این ماژول برابر با 0 می‌باشد.

در مرحله دوم، اگر D0=H'88 شد، مدهای کانال‌های 1 و 3 ورودی آنالوگ برابر با 0 و مدهای کانال‌های 2 و 4 ورودی آنالوگ برابر با مد 3 تنظیم می‌شود.

تعداد دفعات میانگین‌گیری برای کانال‌های 1 و 2 را در رجیسترهای CR#2 و CR#3 برابر با K32 تنظیم می‌کنیم. مقدار سیگنال‌های ورودی میانگین گرفته شده را برای کانال‌های 1 الی 4 را از رجیسترهای کنترلی CR#6 الی CR#9 را می‌خوانیم و این مقادیر را در رجیسترهای داده D20~D23 ذخیره می‌نماییم.

در این مرحله می‌خواهیم بررسی کنیم که آیا ماژول متصل شده به CPU، ماژول مدل DVP02DA-S می‌باشد، یا خیر. شماره ID شناسایی ماژول توسعه‌ی DVP02DA-S برابر با H'49 هگز می‌باشد، یعنی بعد از اتصال ماژول توسعه به CPU، این ماژول مقدار هگز H'49 را به CPU ارسال می‌کند. برای اطمینان از درست بودن ماژول متصل شده به CPU، این مقدار (H'49) را با استفاده از یک مقایسه گر CMP بررسی می‌نماییم. در صورت درست بودن، اجازه‌ی دسترسی به برنامه‌های بعدی داده می‌شود. بعد از درست بودن ماژول متصل شده به CPU می‌خواهیم مقدار موجود در رجیستر کنترلی CR#0 همین ماژول توسعه را بخوانیم. این ماژول، دومین ماژول توسعه‌ی خاص می‌باشد که به CPU متصل شده است. بنابراین شماره‌ی این ماژول برابر با 1 می‌باشد.

در هر ثانیه یک عدد (K1) به رجیستر D100 اضافه شده و پنج عدد (K5) به رجیستر D101 اضافه می‌شود.

وقتی که مقدار موجود در رجیسترهای D100 و D101 برابر با K4000 (مقدار دسیمال 4000) شد، مقدار آنها ریست شده و برابر با 0 می‌شود.

وقتی که فلگ M1=ON شود، این به آن معنی است که دومین ماژول توسعه‌ی خاص متصل شده به CPU ماژول DVP02DA-S می‌باشد. از این رو، مد کانال 1 خروجی آنالوگ برابر با مد 0 تنظیم شده و مد کانال 2 خروجی آنالوگ برابر با مد 2 تنظیم می‌شود.

در این مرحله، مقادیر موجود در رجیسترهای D100 و D101 در داخل رجیسترهای کنترلی CR#10 و CR#11 نوشته می‌شود. همانطور که متوجه شدید، مقدار خروجی‌های آنالوگ با تغییر مقادیر موجود در رجیسترهای D100 و D101 تغییر خواهد کرد.

4-14 دستورالعمل محاسبه‌ی مقدار تناسبی SCAL

API	Mnemonic	Operands	Function
202	SCAL P	(S ₁) (S ₂) (S ₃) (D)	Proportional Value Calculation
Type	Bit Devices	Word Devices	Program Steps
OP	X Y M S K H KnX KnY KnM KnS T C D E F		SCAL, SCALP: 9 steps
S ₁			
S ₂			
S ₃			
D			
PULSE			
16-bit			
32-bit			
ES EX SS SA SX SC EH SV EH3 SV2	ES EX SS SA SX SC EH SV EH3 SV2	ES EX SS SA SX SC EH SV EH3 SV2	ES EX SS SA SX SC EH SV EH3 SV2

این دستورالعمل از 4 عملوند تشکیل شده است:

- عملوند S₁: در این عملوند مقدار داده ورودی (Source value) وارد می‌شود.
 - عملوند S₂: در این عملوند شیب خط (Slope) تنظیم می‌شود.
 - عملوند S₃: در این عملوند مقدار آفست (Offset) تنظیم می‌شود.
 - عملوند D: خروجی دستورالعمل (Destination value) در این عملوند قرار می‌گیرد.
- محدوده‌ی قابل تنظیم برای عملوندهای S₁, S₂, S₃ بین 32767~32768- می‌باشد.
- واحد عملوند S₂ برابر با 0.001 است.

معادله‌ی عملکرد این دستورالعمل $D = (S_1 \times S_2) / 1000 + S_3$ می‌باشد. از این رو، شما باید مقادیر عملوند S₂, S₃ را برای شیب خط و آفست مطابق با معادله زیر بدست آورید.

معادله‌ی شیب خط:

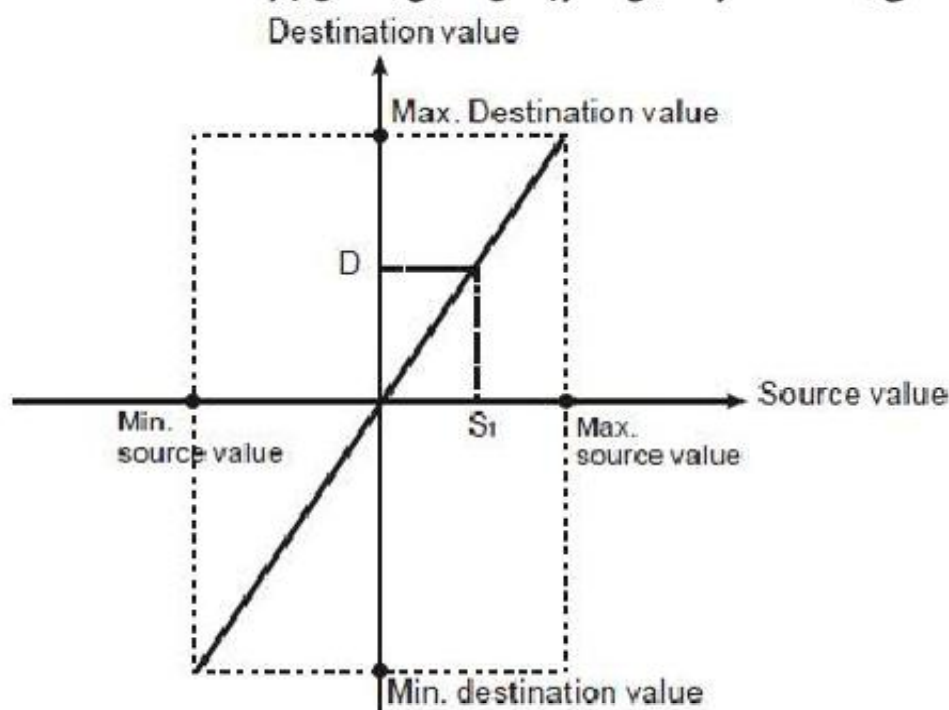
$$S_2 = [(max. destination value - min. destination value) / (max. source value - min. source value)] \times 100$$

معادله‌ی آفست:

$$S_3 = min. destination value - min. source value \times S_2 / 1000$$

مقادیر عملوندهای S₂ و S₃ یک عدد دسیمال بوده و بدون اعشار می‌باشد. عدد اعشاری بدست آمده به سمت عدد بالاتر گرد می‌شود. جنس اعداد این دو عملوند از نوع عدد صحیح 16-bit integer می‌باشد.

خروجی منحنی بدست آمده توسط این دستورالعمل مطابق با شکل زیر است :

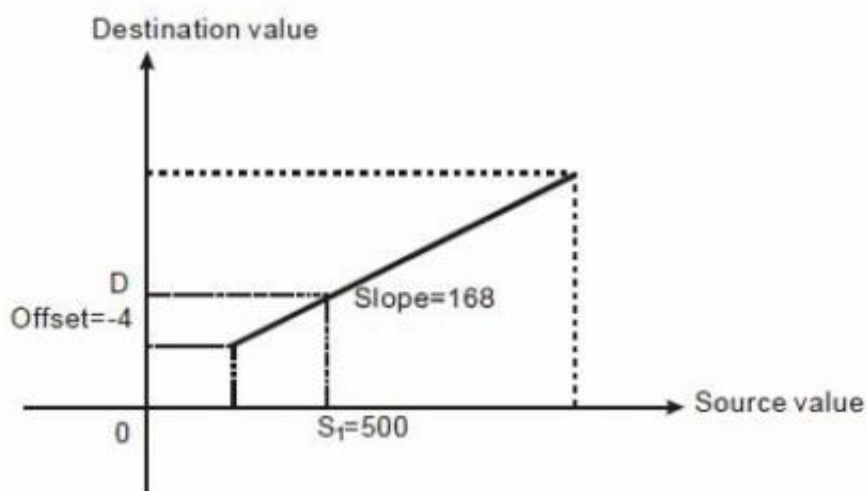


نکته:

از دستورالعمل SCAL می‌توان برای تعیین شیب خط و آفست استفاده کرد. اگر از مقدار شیب خط و آفست اطلاعاتی نداشته باشید، از دستورالعمل SCLP برای محاسبه استفاده نمایید. مقدار S_2 را باید در داخل محدوده‌ی -23768~32767 تنظیم نمایید. اگر مقدار عملوند S_2 خارج از این محدوده بود، باید از دستورالعمل SCLP برای محاسبه استفاده نمایید. وقتی که از معادله‌ی شیب خط استفاده می‌کنید، باید دقت داشته باشید که مقدار max source value بیشتر از مقدار min source value باشد. لذا لزومی ندارد که حتماً مقدار max destination value بیشتر از مقدار min destination value باشد.

مثال 1

در این مثال، می‌خواهیم طبق داده‌های ورودی $S_1=500$, $S_2=168$, $S_3=-4$ شیب خط خروجی را بدست آوریم.

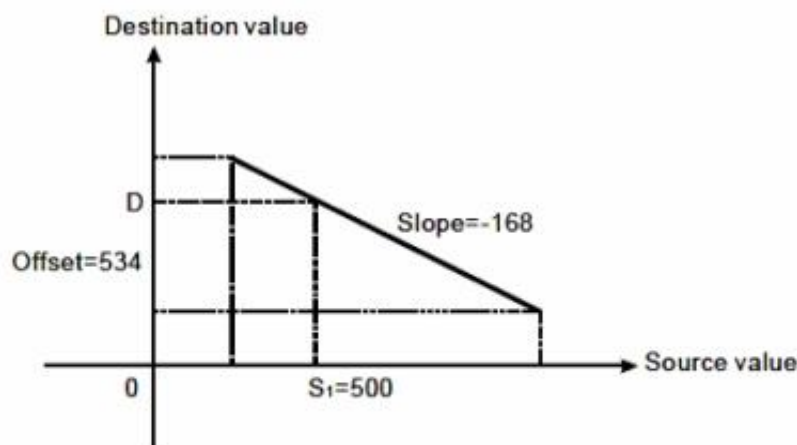
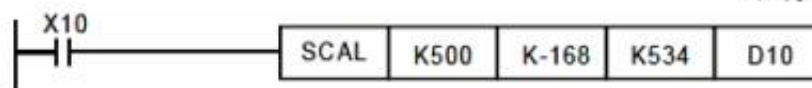


وقتی که ورودی $X0=ON$ شود، دستورالعمل SCAL شروع به اجرا نموده و مقدار تناسبی را در خروجی $D0$ بدست خواهد آورد. معادله این مثال به شرح زیر می‌باشد:

$$D0 = (500 \times 168) / 1000 + (-4) = 80$$

مثال 2

در این مثال، می‌خواهیم طبق داده‌های ورودی $S_1=500$, $S_2=-168$, $S_3=534$ شیب خط خروجی را بدست آوریم.



وقتی که ورودی $X10=ON$ شود، دستورالعمل SACL شروع به اجرا نموده و مقدار تناسبی را در خروجی $D10$ بدست خواهد آورد.

معادله این مثال به شرح زیر می‌باشد :

$$D0 = (500 \times -168) / 1000 + 534 = 450$$

4-1-5 دستورالعمل محاسبه‌ی پارامتر مقدار تناسبی (SCLP)

API	Minemonic			Operands			Function																						
203	D	SCLP	P	(S ₁)	(S ₂)	(D)	Parameter Proportional Value Calculation																						
Type	Bit Devices				Word Devices										Program Steps														
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F														
OP																	SCLP, SCLPP: 7 steps												
S ₁					*	*							*				DSCLP, DSCLPP: 13 steps												
S ₂													*																
D													*																
PULSE										16-bit								32-bit											
ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2

این دستورالعمل از سه عملوند تشکیل شده است :

- عملوند S_1 : در این عملوند مقدار ورودی (Source value) تنظیم می‌شود.

- عملوند S_2 : این عملوند مربوط به پارامترهای دستورالعمل می‌باشد.

- عملوند D : در این عملوند مقدار خروجی ذخیره می‌شود.

برای این دستورالعمل می‌توان از فلگ خاص $M1162$ (مقدار صحیح دسیمال یا اعداد اعشاری باینری) استفاده کرد. اگر $M1162=ON$ باشد، عدد اعشاری باینری فعال خواهد شد.

تنظیمات عملوند S_2 برای دستورالعمل 16 بیتی SCLP که 4 ابزار متوالی را اشغال می‌کند به شرح

زیر می‌باشد :

شماره ابزار	پارامترها	محدوده‌ی قابل تنظیم
S_2	max source value	-32768~32767
S_{2+1}	min source value	-32768~32767
S_{2+2}	max destination value	-32768~32767
S_{2+3}	min destination value	-32768~32767

تنظیمات عملوند S_2 برای دستورالعمل 32 بیتی SCLP که 8 ابزار (رجیستر) متوالی را اشغال می‌کند

به شرح زیر می‌باشد :

شماره ابزار	پارامترها	محدوده‌ی قابل تنظیم	
		عدد صحیح (integer)	عدد اعشاری (Floating point)
S_2, S_{2+1}	max source value	-32768~32767	محدوده‌ی قابل پوشش اعداد اعشاری 32 بیتی
$S_{2+2,3}$	min source value		
$S_{2+4,5}$	max destination value		
$S_{2+6,7}$	min destination value		

معادله عملکرد این دستورالعمل به شرح زیر می‌باشد:

$$D = [(S_1 - \text{min. source value}) \times (\text{max. destination value} - \text{min. destination value})] \div (\text{max. source value} - \text{min. source value}) + \text{min. destination value}$$

رابطه‌ی مؤثر بین source value و destination value به شرح زیر می‌باشد:

$$y = kx + b$$

$$y = \text{Destination value (D)}$$

$$k = \text{Slope} = (\text{max. destination value} - \text{min. destination value}) \div (\text{max. source value} - \text{min. source value})$$

$$x = \text{Source value (S}_1\text{)}$$

$$b = \text{Offset} = \text{Min. destination value} - \text{Min. source value} \times \text{slope}$$

آوردن تمام پارامترها در داخل معادله خطی $y = kx + b$ و بدست آوردن معادله این دستورالعمل به

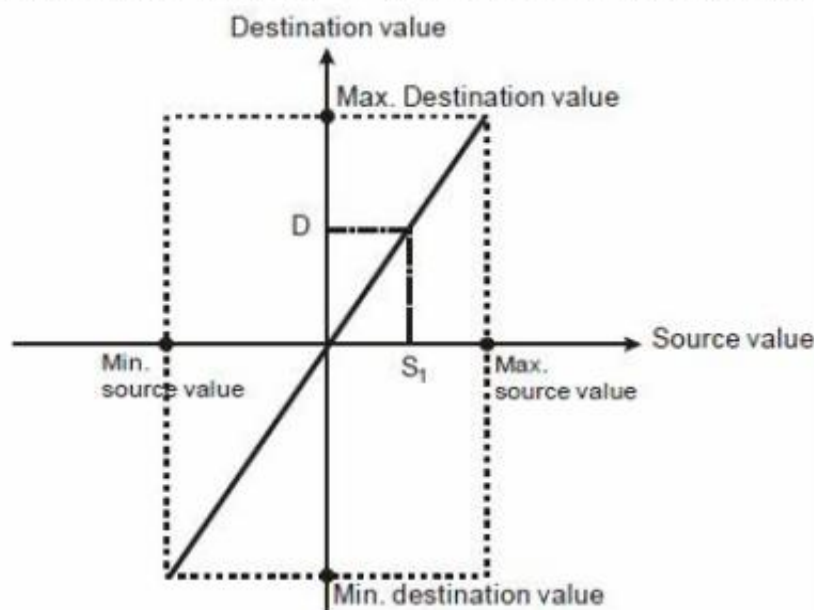
شرح زیر می‌باشد:

$$y = kx + b = D = k S_1 + b = \text{slope} \times S_1 + \text{offset} = \text{slope} \times S_1 + \text{min. destination value} - \text{min. source value} \times \text{slope} = \text{slope} \times (S_1 - \text{min. source value}) + \text{min. destination value} = (S_1 - \text{min. source value}) \times (\text{max. destination value} - \text{min. destination value}) \div (\text{max. source value} - \text{min. source value}) + \text{min. destination value}$$

اگر $S_1 > \text{max source value}$ باشد، آنگاه $S_1 = \text{max source value}$ می‌شود.

اگر $S_1 < \text{min source value}$ باشد، آنگاه $S_1 = \text{min source value}$ می‌شود.

وقتی که تمام مقادیر ورودی و پارامترها را تنظیم کردید، منحنی خروجی مطابق با شکل زیر خواهد شد:



نکته:

محدوده‌ی قابل تنظیم برای عملوند S_1 در دستورالعمل 16 بیتی برابر با $\max \text{ source value} \geq S_1 \geq \min \text{ source value}$ بوده و مقدار آن بین 32767~32768- می‌باشد. اگر مقدار S_1 خارج از این محدوده باشد، مقدار کران بالا و یا کران پایین برای محاسبه استفاده خواهد شد.

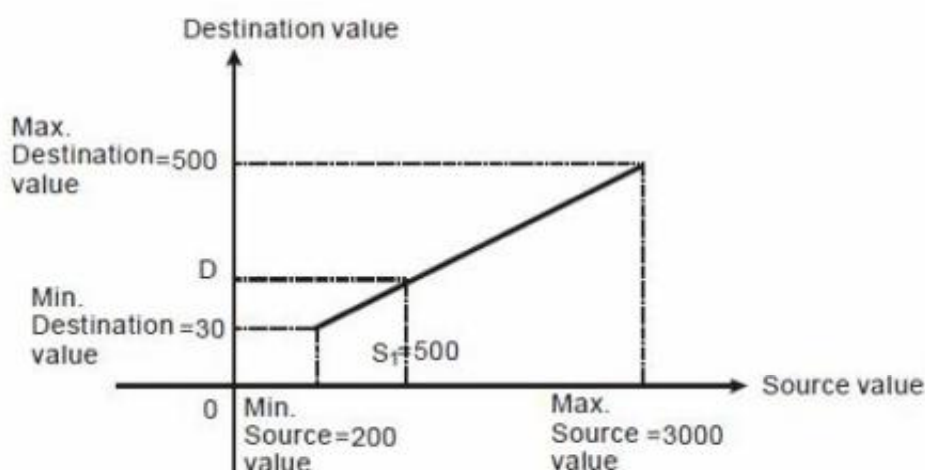
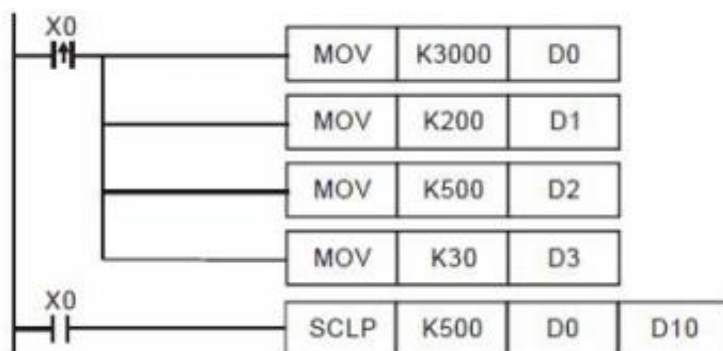
محدوده‌ی عدد صحیح عملوند S_1 برای دستورالعمل 32 بیتی برابر با $\max \text{ source value} \geq S_1 \geq \min \text{ source value}$ بوده و مقداری آن بین 2147483647~2147483648- می‌باشد. اگر مقدار S_1 خارج محدوده‌ی کران بالا و یا کران پایین باشد، مقدار کران بالا و یا کران پایین برای محاسبه استفاده می‌شود.

محدوده‌ی قابل تنظیم اعداد اعشاری برای عملوند S_1 دستورالعمل 32 بیتی برابر با $\max \text{ source value} \geq S_1 \geq \min \text{ source value}$ بوده و مقدار آن عدد اعشاری 32 بیتی می‌باشد. اگر مقدار S_1 خارج از محدوده‌ی کران بالا و یا کران پایین عدد اعشاری باشد، مقدار کران بالا و یا کران پایین برای محاسبه در نظر گرفته می‌شود.

دقت داشته باشید که باید مقدار $\max \text{ source value}$ بیشتر از مقدار $\min \text{ source value}$ باشد. اما لزومی ندارد که مقدار $\max \text{ destination value}$ بیشتر از $\min \text{ destination value}$ باشد.

مثال 1

در این مثال، می‌خواهیم با در دست داشتن مقادیر $S1=500$, $\max. \text{ source value } D0=3000$, $\min. \text{ source value } D1=200$, $\max. \text{ destination value } D2=500$, $\min. \text{ destination value } D3=30$ خروجی تناسبی را بدست آوریم.



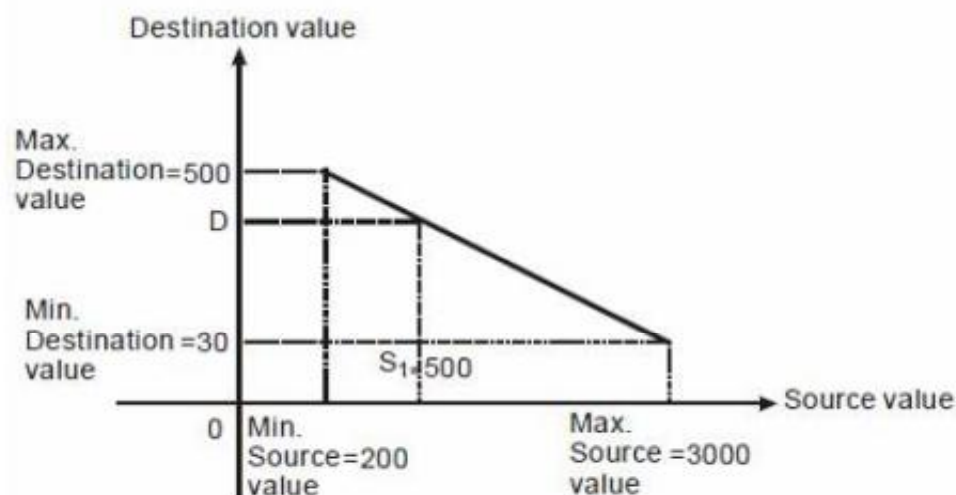
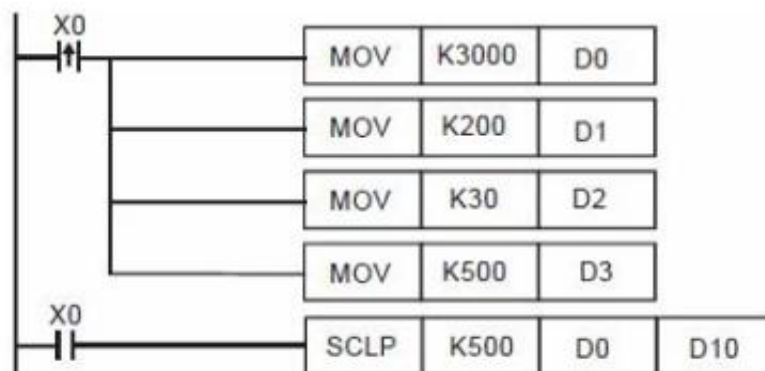
با فعال شدن ورودی $X0=ON$ ، دستورالعمل SCLP شده و مقدار تناسبی بدست آمده در رجیستر D10 ذخیره می‌شود. معادله این مثال به شرح زیر می‌باشد:

$$D10 = [(500 - 200) \times (500 - 30)] / (3000 - 200) + 30 = 80.55$$

از آنجایی که این عدد باید یک عدد صحیح باشد، عدد اعشاری گرد شده و به $D10=80$ تبدیل می‌شود.

مثال 2

در این مثال، می‌خواهیم با در دست داشتن مقادیر $S_1 = 500$, max source value $D0 = 3000$, min source value $D1=200$, max destination value $D2=30$, min destination value $D3=500$ مقدار تناسبی خروجی را به دست آوریم.



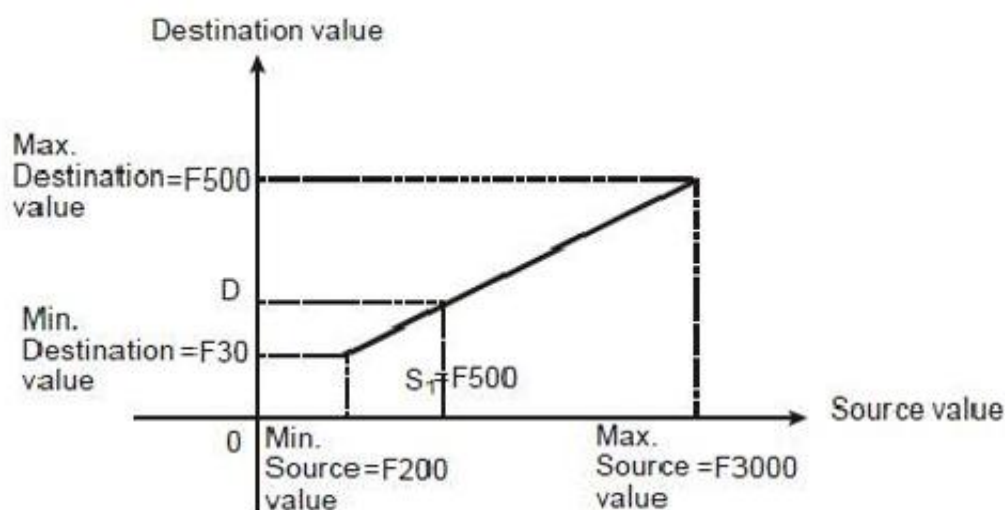
وقتی که ورودی $X0=ON$ شود، دستورالعمل SCLP اجرا شده و مقدار تناسبی بدست آمده در رجیستر D10 ذخیره خواهد شد. معادله‌ی این مثال به شرح زیر می‌باشد:

$$D10 = [(500-200) \times (30-500)] / (3000 - 200) + 500 = 449.64$$

از آنجایی که این عدد باید یک عدد صحیح باشد، عدد بدست آمده به مقدار $D10=450$ گرد خواهد شد.

مثال 3

در این مثال، می‌خواهیم با در دست داشتن مقادیر $S_1 D100=F500$, max source value $D0=F3000$, min source value $D2=F200$, max destination value $D4=F500$, min destination value $D6=F30$ مقدار تناسبی خروجی را بدست آوریم.



وقتی که ورودی $X0=ON$ شود، فلگ خاص M1162 فعال شده و فعال باقی خواهد ماند. در این حالت، سیستم اعداد اعشاری باینری فعال خواهد شد. همچنین، با فعال شدن ورودی $X0$ دستورالعمل DSCLP اجرا شده و مقدار تناسبی بدست آمده در رجیستر D10 ذخیره خواهد شد.

معادله‌ی این مثال به شرح زیر می‌باشد:

$$D10 = [(F500 - F200) \times (F500 - F30)] / (F3000 - F200) + F30 = F80.35$$

از آنجایی که این عدد باید یک عدد صحیح باشد، عدد بدست آمده به مقدار $D10=F80$ گرد خواهد

شد.

4-2 یک برنامه‌ی کاربردی از ماژول DVP04AD

4-2-1 اندازه‌گیری جریان

فرض کنید که یک سنسور اندازه‌گیری سطح آنالوگ از جنس جریان $0 \sim 20\text{mA}$ داریم که باید این مقدار پوسته‌ی آنالوگ به مقدار دیجیتال تبدیل شده و مقدار آن در رجیستر D0 جهت مانیتور کردن مقدار بدست آمده جهت نمایش به اپراتور ذخیره شود.

ازاینرو، باید سیگنال ورودی ماژول آنالوگ به دیجیتال A/D را برابر با مد 3 (مد ورودی جریانی - $20 \sim +20\text{mA}$) تنظیم کرد.

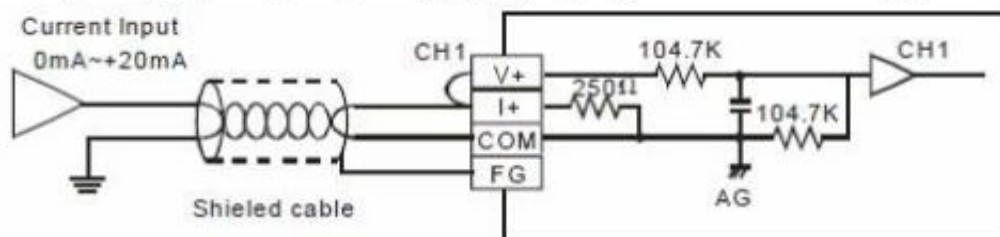
ابزارهای مورد استفاده در این برنامه به شرح زیر می‌باشد:

D40: در این رجیستر مقدار میانگین گرفته شده از سیگنال‌های ورودی آنالوگ ذخیره می‌شود.

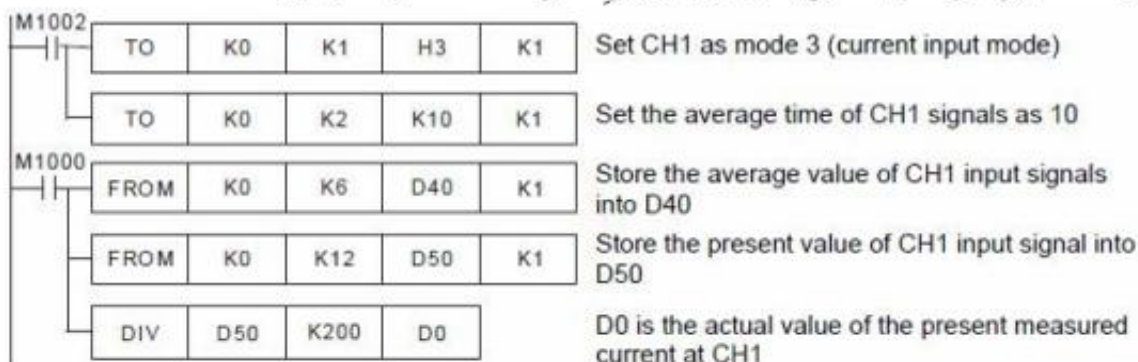
D50: در این رجیستر مقدار جاری (لحظه به لحظه) سیگنال ورودی آنالوگ ذخیره می‌شود.

D0: در این رجیستر مقدار واقعی اندازه‌گیری شده‌ی جریان جاری ذخیره می‌شود.

سیم‌بندی ماژول ورودی آنالوگ ← سیگنال جریان خروجی از سنسور اندازه‌گیری آنالوگ را باید به کانال CH1 ماژول DVP04AD متصل نموده و ترمینال $V+$ و $I+$ را اتصال کوتاه نمایید.



برنامه کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر است:



عملکرد این برنامه کنترلی به شرح زیر می‌باشد:

وقتی که PLC از وضعیت STOP به وضعیت RUN تغییر وضعیت داد، مد کاری کانال CH1 برابر با مد ورودی جریانی (مد 3) و همچنین، تعداد دفعات میانگین‌گیری سیگنال ورودی آنالوگ کانال CH1 برابر با 10 تنظیم خواهد شد.

مقدار میانگین گرفته شده‌ی سیگنال ورودی آنالوگ اندازه‌گیری شده در رجیستر D0 و مقدار جاری سیگنال ورودی آنالوگ اندازه‌گیری شده در رجیستر D50 ذخیره می‌شود.

مد جریان ماژول DVP04AD، مقدار آنالوگ 0~20mA و مقدار دیجیتال K0~K4000 را اندازه‌گیری می‌نماید. در رجیستر D50 حداکثر مقدار جاری آنالوگ برابر با 200 خواهد بود. برای این منظور 4000/20 شده و مقدار بدست آمده برابر با 200 می‌باشد. مقدار موجود در رجیستر D50 تقسیم بر 200 شده و مقدار بدست آمده در رجیستر D0 ذخیره خواهد شد. این مقدار همان مقدار مقیاس‌گذاری شده برای مقدار واقعی اندازه‌گیری شده از سیگنال ورودی آنالوگ جریانی می‌باشد.

2-2-4 تنظیمات سریع ماژول ورودی آنالوگ توسط نرم‌افزار WPLSoft

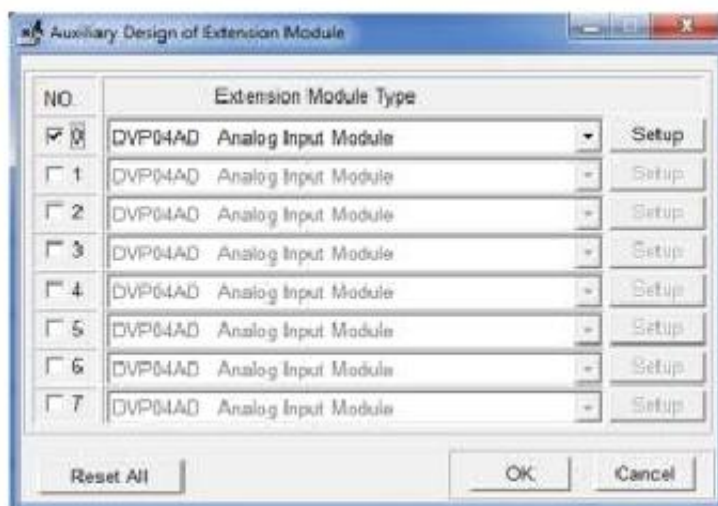
1. ابتدا نرم‌افزار WPLSoft را اجرا نموده و سپس یک برنامه جدید را ایجاد کنید. بر روی آیکن



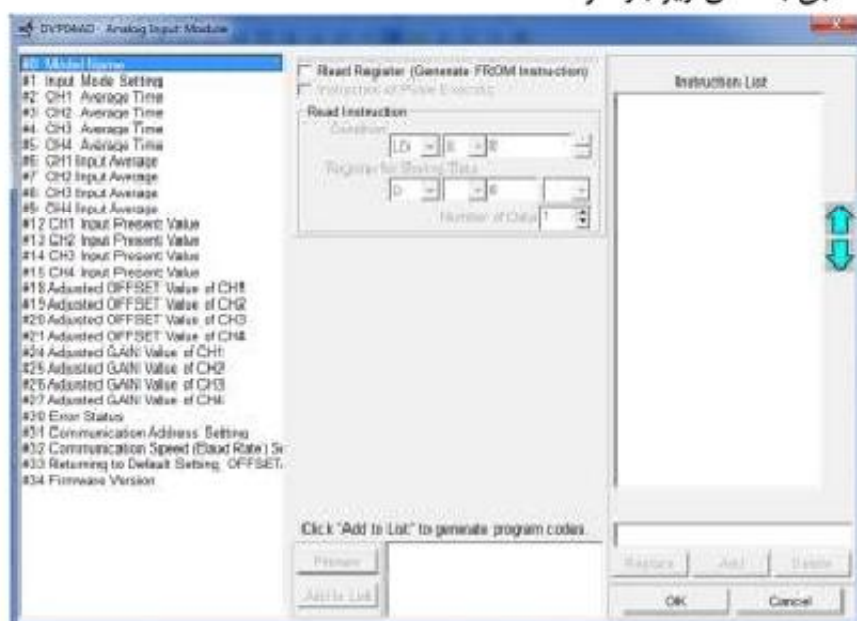
کلیک نمایید.



2. پنجره‌ی مطابق با شکل زیر باز خواهد شد. بر روی "0" NO. کلیک کرده و ماژول DVP04AD Analog Input Module را انتخاب نمایید. سپس بر روی گزینه Setup برای اجرای تنظیمات ماژول کلیک نمایید.



3. پنجره‌ای مطابق با شکل زیر باز خواهد شد.



4. حال باید تنظیمات ماژول را مطابق با توضیحات گفته شده در بالا برای این برنامه کاربردی اجرا نمایید.

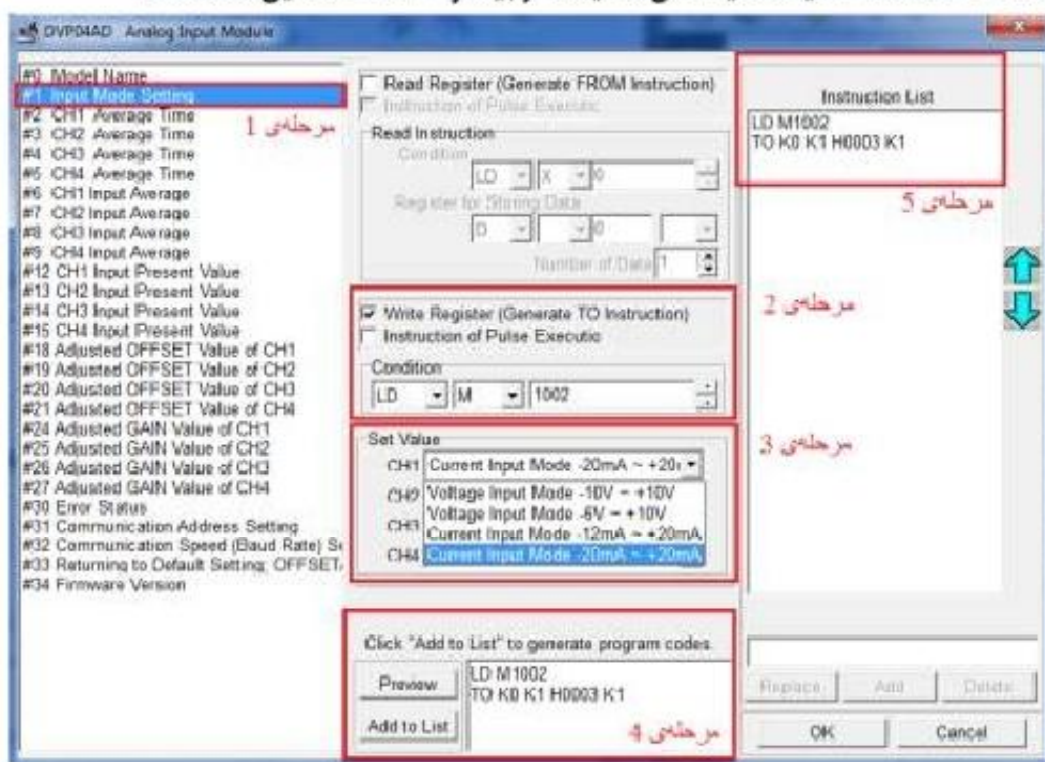
مرحله 1: گزینه **Set Up Input Mode #1** را انتخاب نمایید.

مرحله 2: جهت تولید دستورالعمل **TO** به صورت اتوماتیک، بر روی گزینه **Write Register** کلیک نموده و شرط اجرای دستورالعمل **TO** را برابر با **LD M1002** تنظیم کنید.

مرحله 3: مد کاری کانال **CH1** را برابر با **Current Input Module -** $20\text{mA} \sim +20\text{mA}$ تنظیم نمایید.

مرحله‌ی 4: جهت چک کردن درست بودن کدهای تولید شده توسط برنامه بر روی دکمه‌ی Preview کلیک نمایید.

مرحله‌ی 5: برای نمایش کدهای دستورالعمل به زبان برنامه‌نویسی Instruction List بر روی دکمه‌ی Add to List کلیک کنید. حال تنظیمات رجیستر CR#1 تکمیل شده است.



5. می‌خواهیم رجیستر کنترلی CR#2 را تنظیم نماییم. دقت داشته باشید که تنظیم این رجیستر کنترلی مانند تنظیمات رجیستر کنترلی CR#1 می‌باشد.

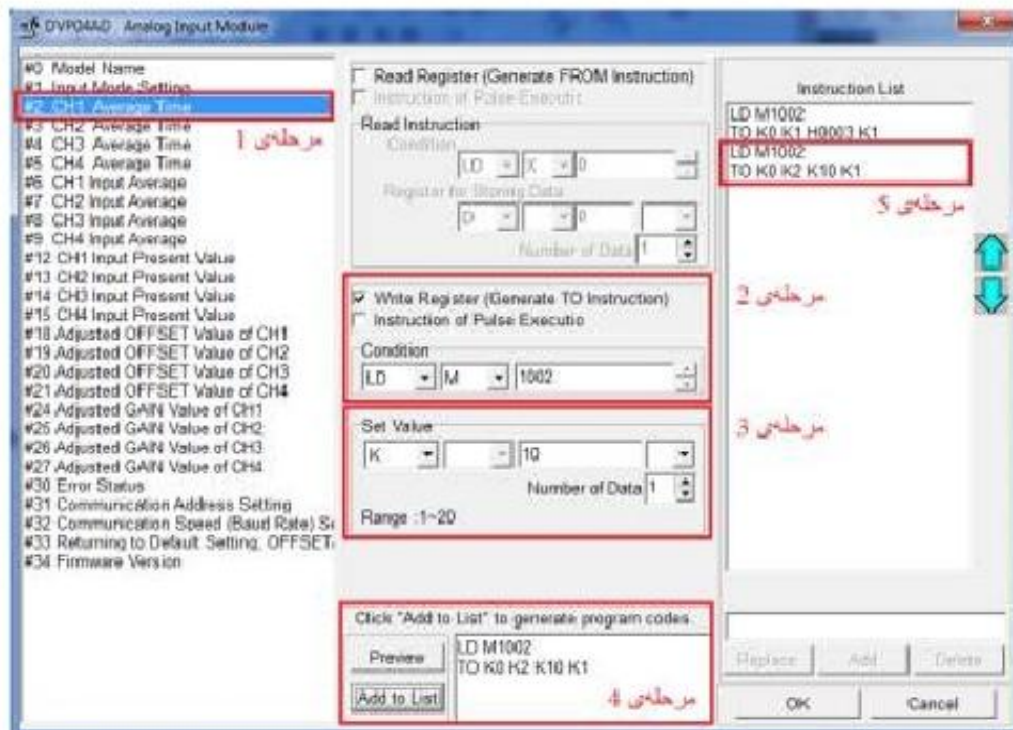
مرحله‌ی 1: گزینه‌ی CH1 Average Times #1 را انتخاب نمایید.

مرحله‌ی 2: جهت تولید دستورالعمل TO به صورت اتوماتیک، بر روی گزینه Write Register کلیک نمایید و شرط اجرا دستورالعمل TO را برابر با LD M1002 تنظیم کنید.

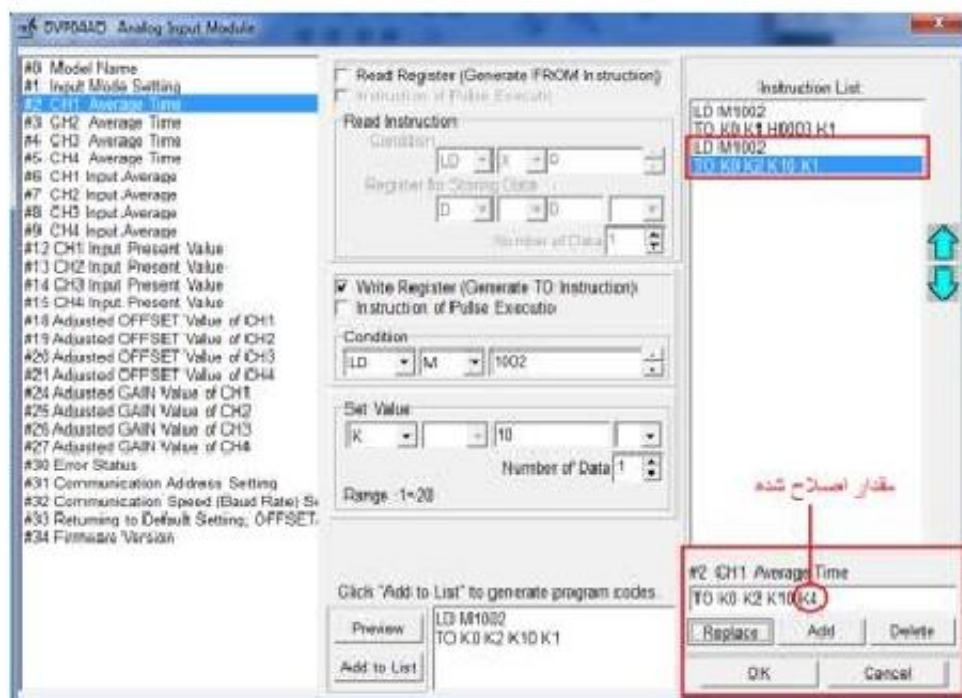
مرحله‌ی 3: در کادر Set value برای تنظیم مقدار مطلوب، مقدار دسیمال K10 را تنظیم کنید و سپس تعداد داده را برابر با 1 تنظیم نمایید.

مرحله‌ی 4: جهت چک کردن درست بودن کدهای تولید شده توسط برنامه، بر روی دکمه Preview کلیک نمایید.

مرحله‌ی 5: برای نمایش کدهای دستورالعمل به زبان برنامه‌نویسی Instruction List بر روی دکمه Add to List کلیک کنید. حال تنظیمات رجیستر CR#2 تکمیل شده است.

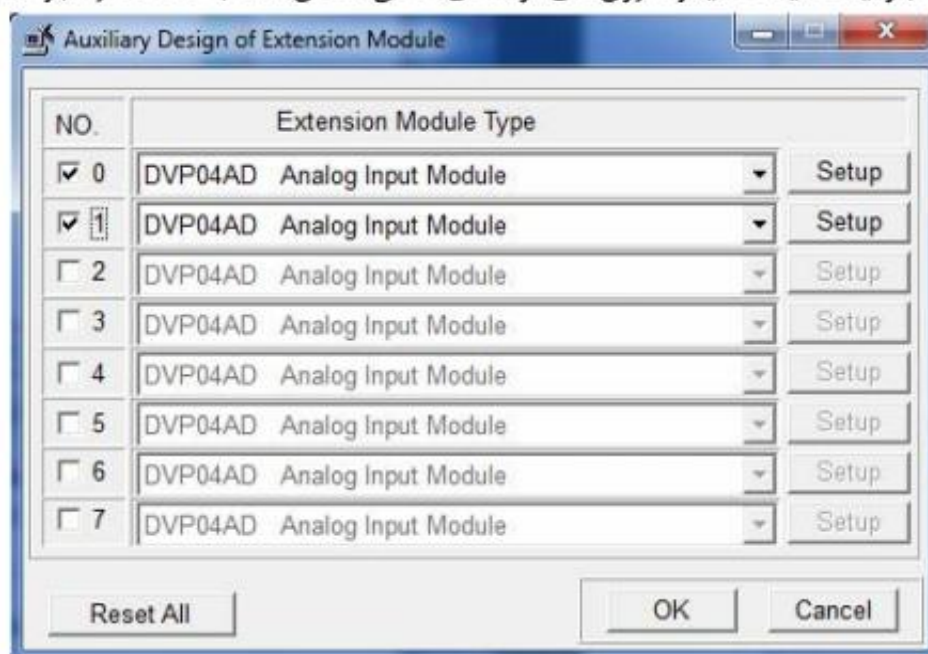


6. در صورت نیاز، می‌توانید مقادیر تنظیم شده را اصلاح کرده و یا یک دستورالعمل جدید را به زبان برنامه‌نویسی **Instruction List** اضافه کنید. برای مثال، اگر می‌خواهید تعداد دفعات میانگین‌گیری رجیستر کنترلی **CR#2** را اصلاح کنید، ابتدا باید بر روی آیتم مورد نظر در کادر **Instruction List** کلیک نموده و سپس مقدار **K1** را بعنوان مثال به **K4** تغییر دهید. حال برای ذخیره کردن تغییرات بوجود آمده باید بر روی دکمه **"Replace"** کلیک نمایید.



7. مطابق با توضیحات گفته شده در بالا، باید دیگر پارامترهای CR را تنظیم نمایید.

8. بعد از اینکه تنظیم تمام پارامترهای CR مربوط به ماژول ورودی آنالوگ تکمیل شد، برای بازگشت به پنجره‌ی Auxiliary Design of Extension Module باید بر روی دکمه OK کلیک کنید تا بتوانید تنظیمات دیگر ماژول‌های توسعه‌ی خاص متصل شده به CPU را اجرا نمایید.



9. بعد از اینکه تنظیم تمام پارامترهای CR مربوط به ماژول‌های توسعه خاص متصل شده به CPU تکمیل شد، برای تولید برنامه مربوط به ماژول‌ها باید بر روی دکمه OK کلیک نمایید.



10. اگر نیاز به اصلاح قسمتی از کدهای تولید شده داشتید و یا نیاز بود که قسمت جدیدی به برنامه‌ی کنترلی اضافه نمایید، می‌توانید به صورت مستقیم در پنجره‌ی برنامه‌نویسی به زبان نردبانی در نرم‌افزار WPLSoft اصلاحات و تغییرات مورد نظر خود را اعمال نمایید.

3-2-4 اندازه‌گیری سرعت الکتروموتور متصل شده به اینورتر

فرض کنید که یک الکتروموتور را به اینورتر سری VFD-B دلتا متصل کرده‌ایم و فرکانس خروجی این اینورتر 0~50.0 Hz می‌باشد. این اینورتر دارای یک ترمینال خروجی آنالوگ 0~10V برای مانیتور کردن سرعت الکتروموتور مطابق با فرکانس خروجی 0~50.0 Hz است. در این مثال، این سیگنال آنالوگ از اینورتر به ماژول DVP04AD ارسال می‌شود. در این برنامه از رجیستر D0 جهت نمایش مقدار ولتاژ آنالوگ استفاده شده است.

بعد از اینکه الکتروموتور شروع به چرخیدن کرد، مقدار ولتاژ آنالوگ مطابق با سرعت الکتروموتور و مقدار واقعی فرکانس خروجی از اینورتر به PLC ارسال شده و در رجیستر D0 ریخته می‌شود و مطابق با برنامه نوشته شده، فرکانس در رجیستر D4 ذخیره می‌شود.

مد کاری سیگنال ورودی آنالوگ ماژول A/D برابر با مد 0 تنظیم شده است. دقت داشته باشید که مد کاری 0 همان مد ولتاژ ورودی آنالوگ با سطح ولتاژ $-10V \sim +10V$ می‌باشد.

ابزارهای مورد استفاده در این مثال به شرح زیر است:

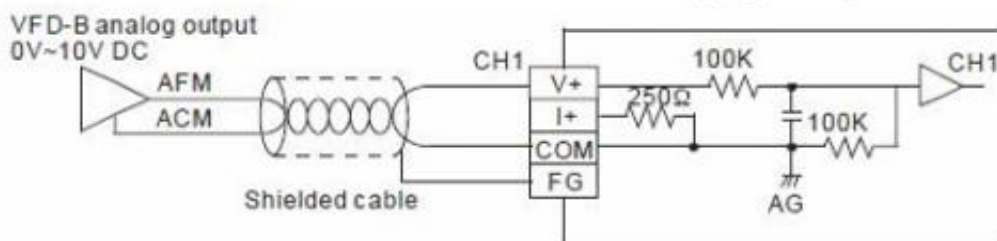
D40: در این رجیستر مقدار میانگین گرفته شده از سیگنال‌های ورودی ذخیره می‌شود.

D50: در این رجیستر مقدار جاری سیگنال ورودی ذخیره می‌شود.

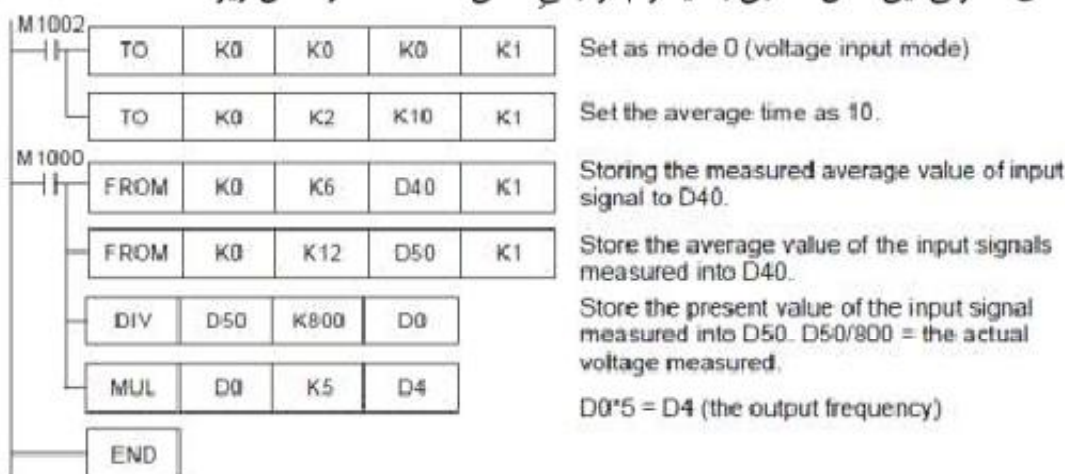
D0: در این رجیستر مقدار واقعی ولتاژ اندازه‌گیری شده‌ی جاری ذخیره می‌شود.

D4: در این رجیستر فرکانس واقعی اینورتر سری VFD-B ذخیره می‌شود.

سیم‌بندی و نحوه‌ی اتصال سیگنال آنالوگ 0~10VDC خروجی اینورتر و اتصال آن به ماژول DVP04AD مطابق با شکل زیر است :



برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر است :



عملکرد این برنامه‌ی کنترلی به شرح زیر می‌باشد :

وقتی که PLC از وضعیت STOP به RUN تغییر وضعیت داد، ولتاژ ورودی آنالوگ 0~10VDC از اینورتر VFD-B به PLC (ماژول ورودی آنالوگ DVP04AD) ارسال می‌شود. بنابراین، مد ورودی این ماژول را باید برابر با مد 0 (مد ولتاژ ورودی 0~10V) تنظیم کرد.

مقدار جاری (لحظه به لحظه) از سیگنال ورودی اندازه‌گیری شده در رجیستر D50 ذخیره می‌شود. در مد کاری ولتاژی ماژول DVP04AD، محدوده‌ی آنالوگ 0~10VDC برابر با مقدار دسیمال K0~K8000 می‌باشد. برای نمایش مقدار ولتاژ ورودی آنالوگ 0~10V از اینورتر، مقدار K8000 را باید تقسیم بر 800 نماییم ($8000/10V=800$) یعنی به ازای هر یک ولت عدد دسیمال داریم. بنابراین همانطور که متوجه شدید، مقدار موجود در رجیستر D50 برابر با 800 مقدار ولتاژ واقعی می‌باشد. مقدار بدست آمده از این تقسیم ($D50/800$) در رجیستر D0 جهت نمایش مقدار واقعی اندازه‌گیری شده‌ی ولتاژ ورودی برحسب V (ولت) ذخیره خواهد شد.

مقدار بدست آمده در رجیستر D0 را باید 5 برابر کنید تا فرکانس خروجی اینورتر بدست آید. دقت داشته باشید که فرکانس 0~50.0 Hz مطابق با 0~10V بوده و تغییر خواهد کرد. بنابراین، برای

تبدیل ولتاژ به فرکانس مقدار D0 را باید در 5 ضرب کرده و مقدار بدست آمده را در رجیستر D4 ذخیره کرد. از این رو، مقدار بدست آمده همان مقدار فرکانس خروجی واقعی از اینورتر سری VFD-B دلتا می‌باشد.

4-3 یک برنامه کاربردی از مازول خروجی آنالوگ DVP04DA

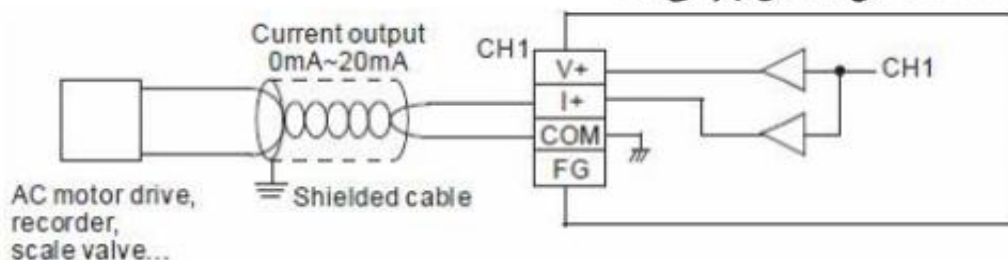
4-3-1 خروجی آنالوگ از جنس جریان

فرض کنید یک اینورتر داریم که باید جریان خروجی $0\sim 20\text{mA}$ را از مازول خروجی آنالوگ به ورودی آنالوگ اینورتر جهت کنترل سرعت الکتروموتور ارسال کنیم. مد کاری سیگنال خروجی D/A در مازول خروجی آنالوگ برابر با مد 3 تنظیم شده است. جریان خروجی این مد کاری برابر با $0\sim 20\text{mA}$ می‌باشد. ابزارهای مورد استفاده در این مثال به شرح زیر است:

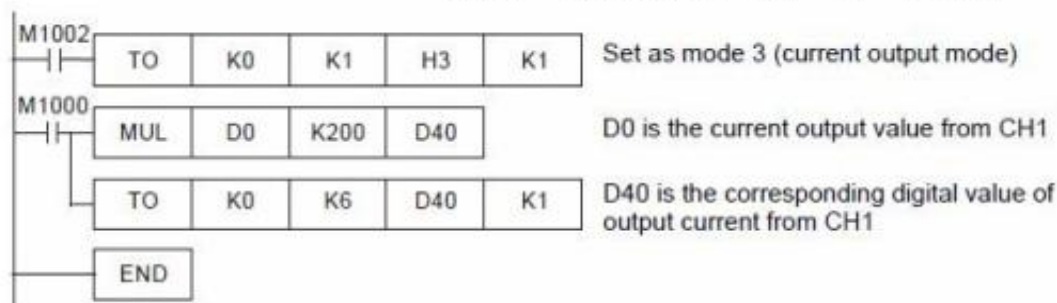
D0: در این رجیستر جریان خروجی کانال CH1 تنظیم می‌شود.

D40: در این رجیستر مقدار دیجیتال تبدیل شده مطابق با خروجی کانال CH1 قرار دارد.

اتصال و سیم‌بندی سیگنال خروجی جریانی برای کنترل خروجی آنالوگ کانال CH1 مازول DVP04DA مطابق با شکل زیر می‌باشد.



برنامه کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر است:



عملکرد این برنامه به شرح زیر می‌باشد:

وقتی که PLC از وضعیت STOP به وضعیت RUN تغییر وضعیت دهد، مد کاری کانال CH1 برابر با مد خروجی جریانی (مد 3) تنظیم خواهد شد.

مد خروجی جریانی ماژول DVP04DA، در محدوده‌ی آنالوگ 0~20mA و محدوده‌ی دیجیتال K0~K4000 تغییر خواهد کرد. مقدار خروجی جریانی توسط رجیستر D0 کنترل می‌شود. مقدار موجود در رجیستر D0 تقسیم بر 200 شده و مقدار بدست آمده در رجیستر D40 ذخیره می‌شود. مقدار موجود در رجیستر D40 به صورت مستقیم، مقدار جریان خروجی آنالوگ کانال CH1 را کنترل کرده و آن را تغییر می‌دهد. اگر مقدار D0 را برابر با 8000 تنظیم نماییم، آنگاه 8000/200 شده و مقدار 40 در رجیستر D40 ذخیره می‌شود. در غیر این صورت، مقدار دیجیتال خروجی کانال CH1 برابر با 40 بوده و مقدار آنالوگ آن $40 \times 0.5 \mu A = 20 \mu A$ خواهد شد.

2-3-4 تنظیمات سریع پارامترهای ماژول خروجی آنالوگ توسط نرم‌افزار WPLSoft

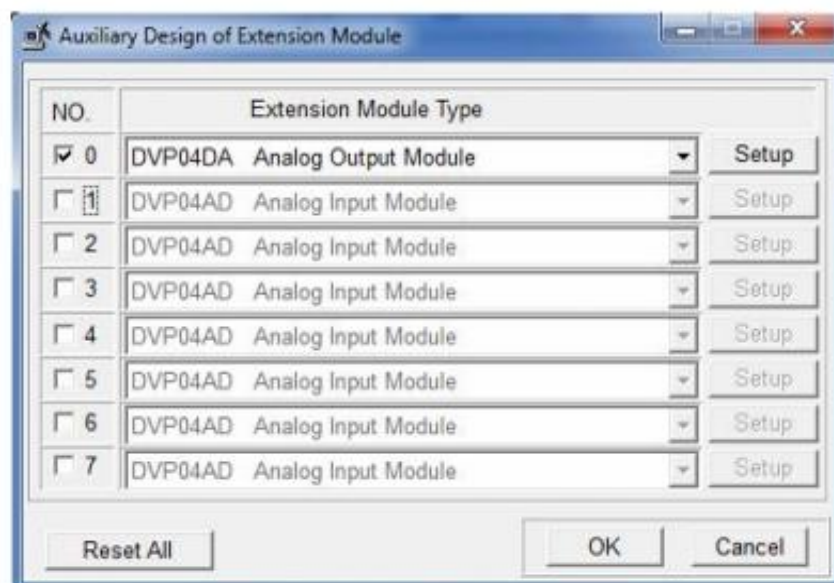
1. ابتدا نرم‌افزار WPLSoft را اجرا نموده و سپس یک برنامه جدید را ایجاد کنید. بر روی آیکن



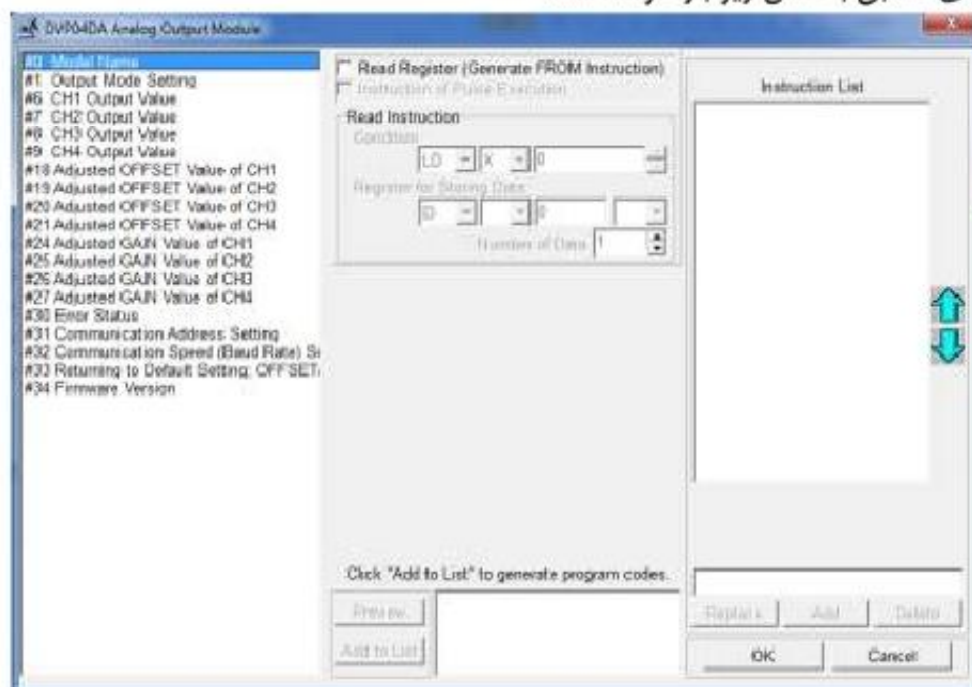
کلیک نمایید.



2. پنجره‌ای مطابق با شکل زیر باز خواهد شد. بر روی "0" NO. کلیک نموده و ماژول DVP04DA Analog Output Module را انتخاب نمایید. سپس بر روی گزینه Setup برای اجرای تنظیمات ماژول کلیک نمایید.



3. پنجره‌ای مطابق با شکل زیر باز خواهد شد.



4. حال باید تنظیمات ماژول را مطابق با توضیحات گفته شده در بالا برای این برنامه کاربردی اجرا نمایید.

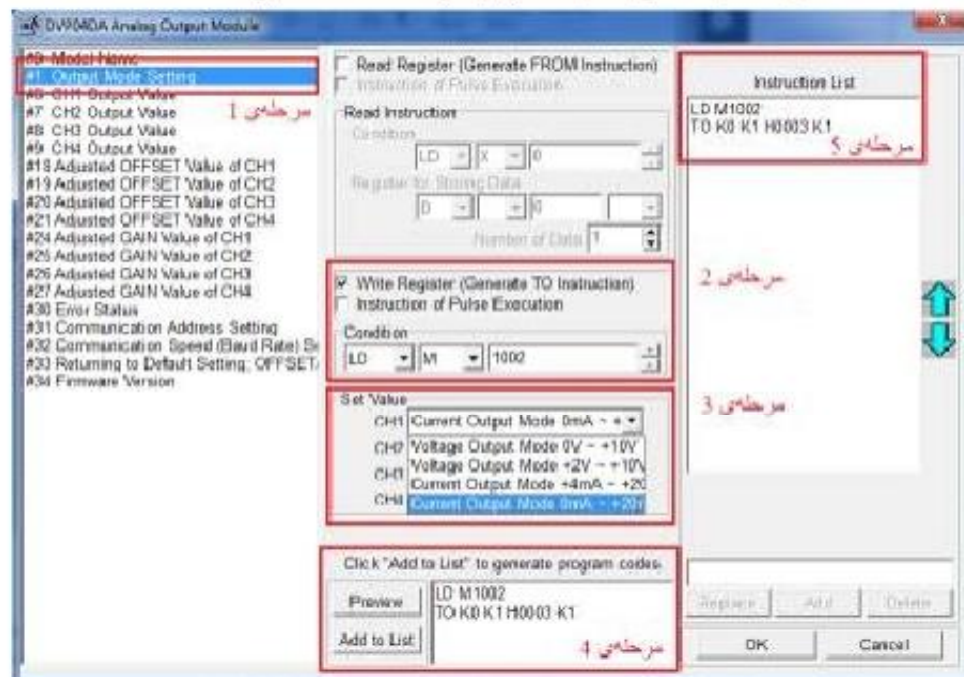
مرحله‌ی 1: گزینه‌ی **#1 Set Up Output Mode** را انتخاب نمایید.

مرحله‌ی 2: جهت تولید دستورالعمل **TO** به صورت اتوماتیک، بر روی گزینه **Write Register** کلیک نموده و شرط اجرای دستورالعمل **TO** را برابر با **LD M1002** تنظیم کنید.

مرحله‌ی 3: مد کاری کانال CH1 را برابر با Current Output Module 0mA~+20mA تنظیم نمایید.

مرحله‌ی 4: جهت چک کردن درست بودن کدهای تولید شده توسط برنامه بر روی دکمه Preview کلیک نمایید.

مرحله‌ی 5: برای نمایش کدهای دستورالعمل به زبان برنامه‌نویسی Instruction List بر روی دکمه Add to List کلیک کنید. حال تنظیمات رجیستر CR#1 تکمیل شده است.



5. می‌خواهیم رجیستر کنترلی CR#6 را تنظیم نماییم. دقت داشته باشید که تنظیم این رجیستر کنترلی مانند تنظیمات رجیستر کنترلی CR#1 می‌باشد.

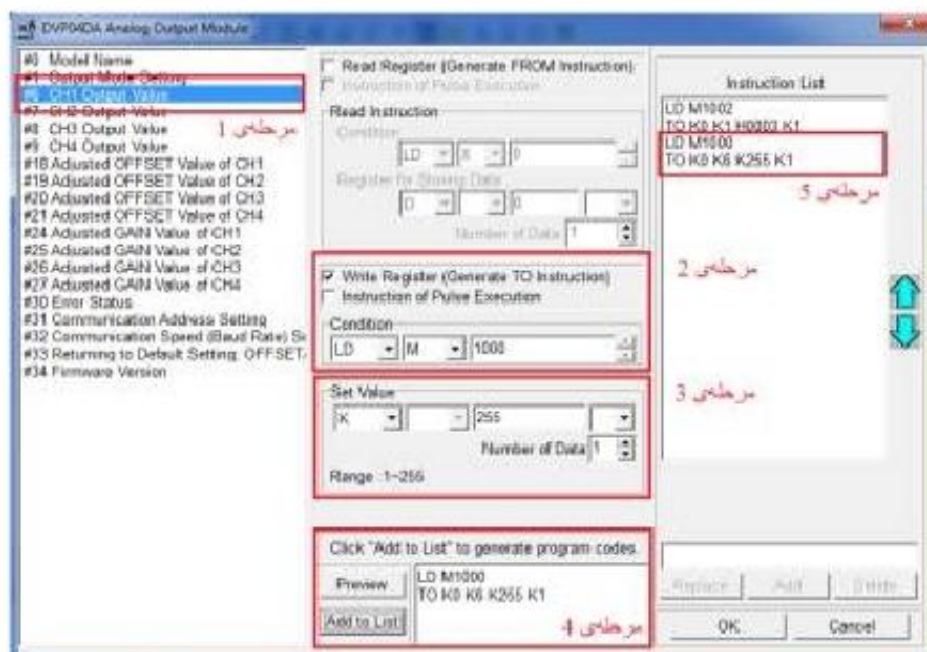
مرحله‌ی 1: گزینه‌ی CH1 Output Value #6 را انتخاب نمایید.

مرحله‌ی 2: جهت تولید دستورالعمل TO به صورت اتوماتیک، بر روی گزینه Write Register کلیک نموده و شرط اجرای دستورالعمل TO را برابر با LD M1000 تنظیم کنید.

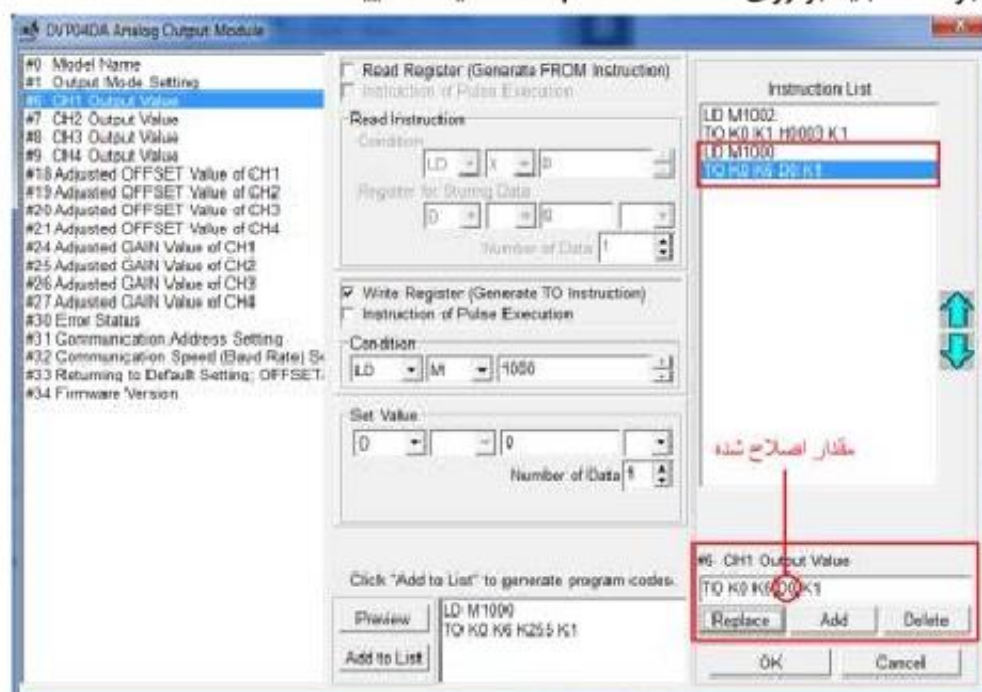
مرحله‌ی 3: در کادر Set value برای تنظیم مقدار مطلوب، مقدار دسیمال K255 را تنظیم کرده و سپس تعداد داده را برابر با 1 تنظیم نمایید.

مرحله‌ی 4: جهت اطمینان از درست بودن کدهای تولید شده توسط برنامه، بر روی دکمه Preview کلیک نمایید.

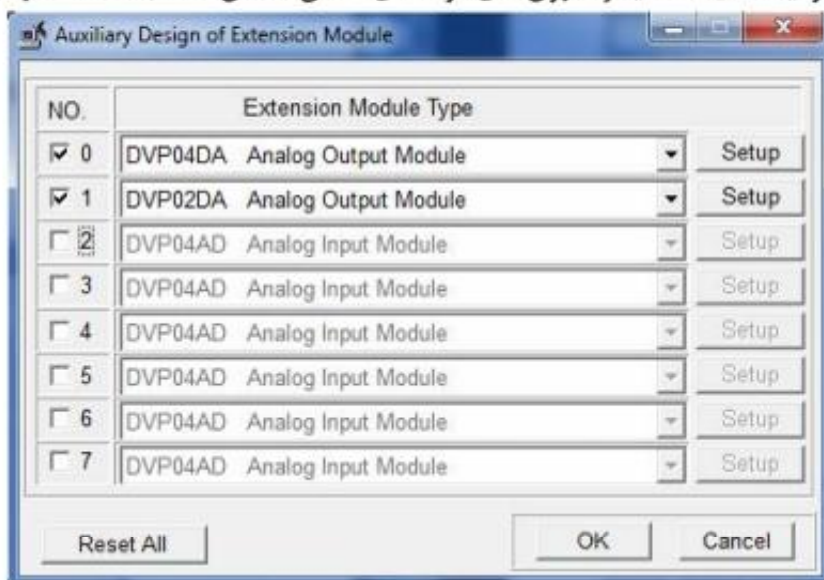
مرحله‌ی 5: برای نمایش کدهای دستورالعمل به زبان برنامه‌نویسی Instruction List بر روی دکمه Add to List کلیک کنید. حال تنظیمات رجیستر CR#6 تکمیل شده است.



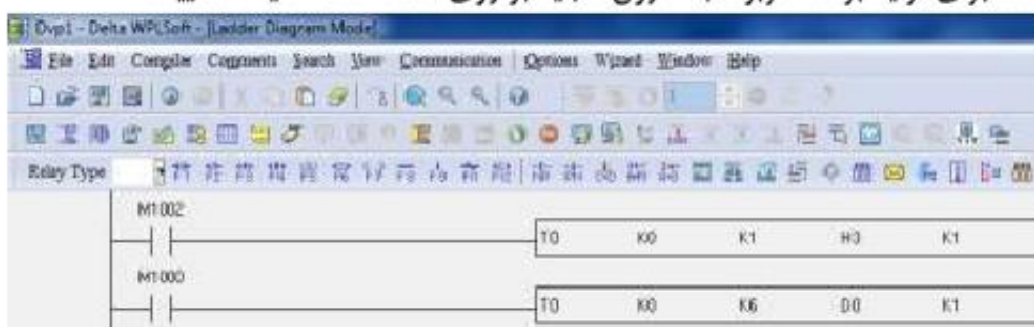
6. در صورت نیاز، می‌توانید مقادیر تنظیم شده را اصلاح کرده و یا یک دستورالعمل جدید را به زبان برنامه‌نویسی Instruction List اضافه کنید. برای مثال، اگر می‌خواهید مقدار خروجی آنالوگ موجود در رجیستر کنترلی CR#6 را اصلاح کنید، ابتدا باید بر روی آیتم مورد نظر در کادر Instruction List کلیک نموده و سپس مقدار K255 را بعنوان مثال به D0 تغییر دهید. حال برای ذخیره کردن تغییرات بوجود آمده باید بر روی دکمه "Replace" کلیک نمایید.



7. مطابق با توضیحات گفته شده در بالا، باید دیگر پارامترهای CR را تنظیم نمایید.
8. بعد از اینکه تنظیم تمام پارامترهای CR مربوط به ماژول خروجی آنالوگ تکمیل شد، برای بازگشت به پنجره‌ی Auxiliary Design of Extension Module باید بر روی دکمه OK کلیک نمایید تا بتوانید تنظیمات دیگر ماژول‌های توسعه‌ی خاص متصل شده به CPU را اجرا نمایید.



9. بعد از اینکه تنظیم تمام پارامترهای CR مربوط به ماژول‌های توسعه خاص متصل شده به CPU تکمیل شد، برای تولید برنامه مربوط به ماژول‌ها باید بر روی دکمه OK کلیک نمایید.



10. اگر نیاز به اصلاح قسمتی از کدهای تولید شده داشتید و یا نیاز بود که قسمت جدیدی به برنامه‌ی کنترلی اضافه نمایید، می‌توانید به صورت مستقیم در پنجره‌ی برنامه‌نویسی به زبان نردبانی در نرم‌افزار WPLSoft اصلاحات و تغییرات مورد نظر خود را اعمال نمایید.

3-3-4 تنظیم فرکانس خروجی اینورتر توسط ماژول D/A

اینورتر سری VFD-B دلتا دارای دو ترمینال ورودی آنالوگ به نام AVI/ACM می‌باشد که از سیگنال آنالوگ از جنس ولتاژ و مقدار 0~10V پشتیبانی می‌کند. با استفاده از این مقدار آنالوگ می‌توان

فرکانس 0~50Hz خروجی اینورتر را کنترل کرد. از این رو، شما می‌توانید با استفاده از یک ماژول خروجی آنالوگ سری DVP04DA سرعت الکتروموتور متصل شده به اینورتر را کنترل نمایید. مد کاری ماژول خروجی آنالوگ D/A را باید برابر با مد 0 تنظیم نمایید. این مد (مد 0) از سیگنال خروجی آنالوگ از جنس ولتاژ و با سطح ولتاژ 0~10V پشتیبانی می‌کند.

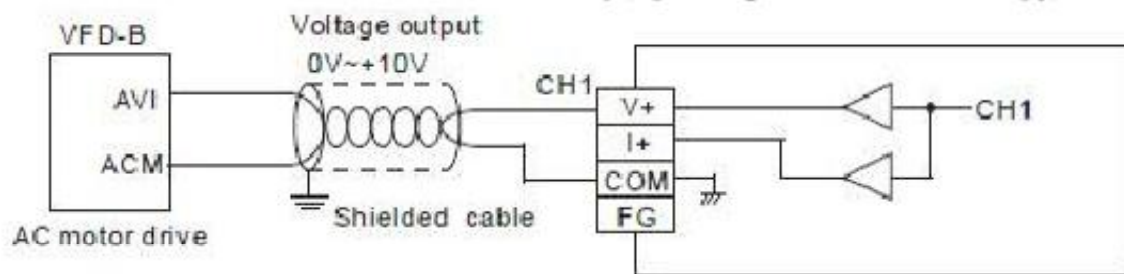
ابزارهای مورد استفاده در این برنامه به شرح زیر می‌باشند:

D0: از این رجیستر جهت کنترل فرکانس 0~50Hz خروجی اینورتر استفاده می‌شود.

D40: در این رجیستر مقدار دیجیتال ولتاژ خروجی آنالوگ کانال CH1 قرار دارد.

سیم‌بندی و اتصال ترمینال ورودی آنالوگ اینورتر (AVI/ACM) سری VFD-B به کانال

CH1 ماژول DVP04DA مطابق با شکل زیر است:



برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



Set as mode 0 (voltage output mode)

D0 = the frequency of AC motor drive (0 ~ 50Hz). Store the operational result into D40.

D40 is the corresponding digital value of output voltage from CH1

عملکرد این برنامه به شرح زیر است:

وقتی که PLC از وضعیت STOP به RUN تغییر وضعیت دهد، مد کاری کانال CH1 برابر با مد

0 (خروجی ولتاژی) تنظیم خواهد شد.

مقدار ولتاژ خروجی مد کاری 0 ماژول DVP04DA برابر با 0~10V آنالوگ و K0~K4000

دیجیتال می‌باشد. مقدار فرکانس اینورتر (0~50Hz) در رجیستر D0 تنظیم شده و این مقدار (مقدار

موجود در داخل رجیستر D0) در عدد دسیمال 80 ضرب می‌شود ($50\text{Hz} \times 80 = 4000$). مقدار

بدست آمده از این ضرب همان مقدار واقعی ولتاژ خروجی آنالوگ کانال CH1 ماژول DVP04DA

می‌باشد. مقدار بدست آمده از این ضرب در رجیستر D40 جهت کنترل مقدار ولتاژ خروجی کانال

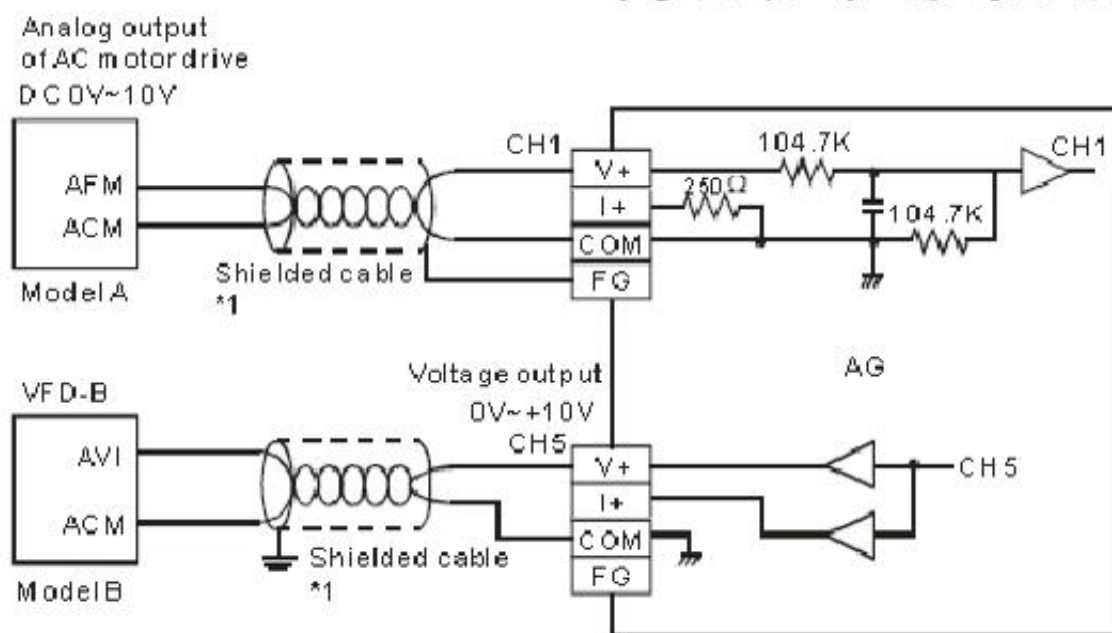
CH1 ماژول DVP04DA ریخته می‌شود. بنابراین با تغییر مقدار موجود در رجیستر D0، مقدار موجود در رجیستر D40 نیز تغییر کرده و مقدار ولتاژ خروجی نیز تغییر می‌کند.

4-4 مثال کاربردی برای ماژول ورودی/خروجی آنالگ DVP06XA

4-4-1 خواندن سرعت از اینورتر

اینورترهای سری VFD-B دلتا دارای ترمینال خروجی آنالوگ ولتاژی چند منظوره (نام ترمینال AFM می‌باشد) می‌باشند که می‌توان فرکانس خروجی اینورتر ($0\sim 50\text{Hz}$) را مطابق با ولتاژ $0\sim 10\text{V}$ به صورت خطی به این ترمینال اختصاص داد و با استفاده از سیم‌بندی این ولتاژ آنالوگ را به یکی از کانال‌های ورودی ولتاژی ماژول DVP06XA-S متصل کرد. همچنین، برای کنترل سرعت اینورتر سری VFD-B باید با استفاده از سیم‌بندی، خروجی آنالوگ ولتاژی ماژول DVP06XA را به ورودی آنالوگ ولتاژی اینورتر (ترمینال AVI) متصل نماییم. از این نوع سیم‌بندی برای ردیابی سرعت موتور به صورت اتوماتیک استفاده می‌شود.

سیگنال ورودی کانال CH1 را باید برابر با مد 0 (مد ورودی ولتاژی $0\sim 10\text{V}$) تنظیم نمایید. سیگنال خروجی کانال CH5 را باید برابر با مد 0 (مد خروجی ولتاژی $0\sim 10\text{V}$) تنظیم نمایید. سیم‌بندی این مثال به صورت زیر انجام می‌شود:



همانطور که متوجه شدید، ترمینال خروجی آنالوگ از جنس ولتاژ (ترمینال AFM/ACM) اینورتر سری VFD-B دلتا را به کانال CH1 ماژول DVP06XA و همچنین، ترمینال ورودی آنالوگ از

جنس ولتاژ (ترمینال AVI/ACM) اینورتر سری VFD-B دلتا را به کانال CH5 مازول DVP06XA متصل کردیم.

ابزارهای مورد استفاده در این برنامه به شرح زیر است:

از رجیستر D0 برای ذخیره‌سازی مقدار ولتاژ اندازه‌گیری شده استفاده می‌شود.

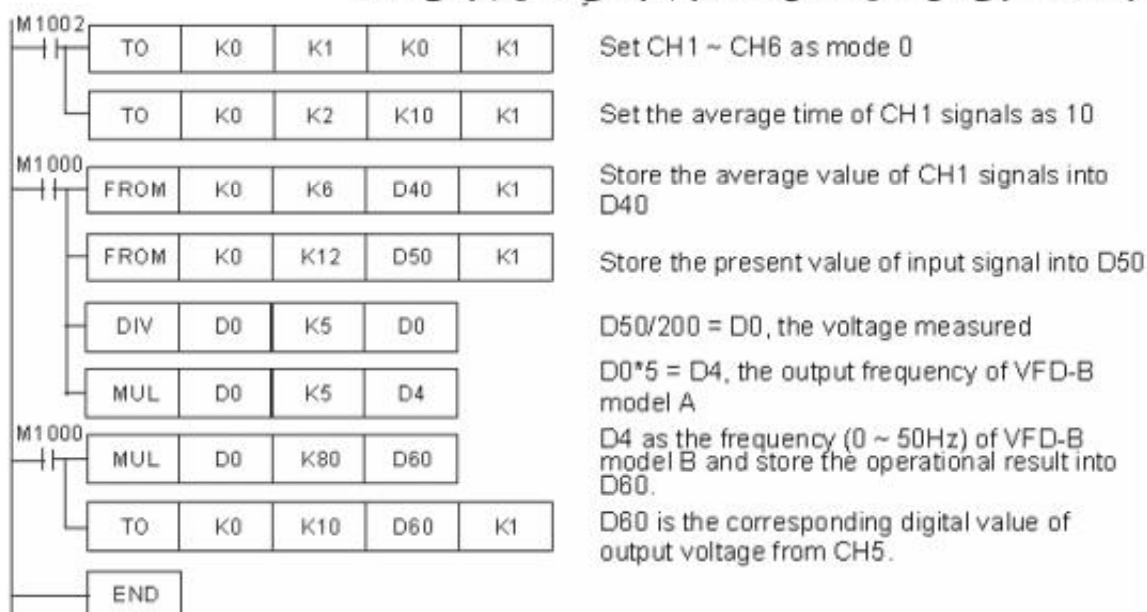
از رجیستر D4 برای نمایش مقدار فرکانس اینورتر سری VFD-B استفاده می‌شود.

از رجیستر D40 برای نمایش مقدار میانگین گرفته شده از سیگنال ورودی کانال CH1 استفاده می‌شود.

از رجیستر D50 برای نمایش مقدار لحظه به لحظه سیگنال ورودی کانال CH1 استفاده می‌شود.

از رجیستر D60 برای ارسال مقدار دیجیتال به خروجی آنالوگ ولتاژ کانال CH5 استفاده می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی بالا به شرح زیر است:

وقتی که PLC از وضعیت STOP به RUN برود، مد 0 برای کانال 1 و کانال 5 تنظیم خواهد شد. علت این مهم این است که خروجی آنالوگ ولتاژ اینورتر سری VFD-B بین 0~10V است، بنابراین شما باید مد 0 (مد ورودی آنالوگ از جنس ولتاژ) را برای کانال‌های CH1~CH4 و مد 0 (مد خروجی آنالوگ از جنس ولتاژ) را برای کانال‌های CH5~CH6 تنظیم نمایید. علاوه، تعداد دفعات میانگین‌گیری از سیگنال ورودی کانال CH1 را برابر با 10 تنظیم نمایید.

مقدار سرعت جاری (لحظه به لحظه) الکتروموتور که توسط سیگنال ورودی آنالوگ اندازه‌گیری می‌شود را در داخل D50 ذخیره نمایید.

محدوده‌ی مقدار دیجیتال ولتاژ ورودی 0~10VDC آنالوگ در مد ولتاژی ماژول DVP06XA برابر با K0~K2000 می‌باشد. از اینرو، مقدار موجود در رجیستر D50 دویست برابر مقدار واقعی ولتاژ می‌باشد ($K2000/10V=200$). برای اندازه‌گیری مقدار واقعی ولتاژ باید مقدار موجود در رجیستر D50 را تقسیم بر 200 نموده و سپس نتیجه‌ی بدست آمده را در رجیستر D0 ذخیره کنید.

مقدار فرکانس خروجی از اینورتر (0~50Hz) پنج برابر مقدار ولتاژ خروجی آنالوگ از اینورتر (0~10V) می‌باشد. از این رو، برای بدست آوردن مقدار فرکانس خروجی اینورتر باید مقدار موجود در رجیستر D0 را در 5 ضرب نماییم. نتیجه‌ی بدست آمده از عمل ضرب را در رجیستر D4 جهت نمایش مقدار فرکانس واقعی اعمالی به الکتروموتور ذخیره می‌کنیم.

محدوده‌ی مقدار دیجیتال ولتاژ خروجی آنالوگ 0~10VDC در مد ولتاژی ماژول DVP06XA برابر با K0~K4000 است. در رجیستر D50 مقدار فرکانس (0~50Hz) اینورتر ذخیره می‌شود. همانطور که متوجه شدید مقدار دیجیتال ولتاژ خروجی آنالوگ 80 برابر مقدار فرکانس خروجی اینورتر می‌باشد. ($50Hz/K4000=80$). از اینرو باید مقدار موجود در رجیستر D4 را در 80 ضرب کرده و مقدار بدست آمده را در رجیستر D60 ذخیره نماییم. حال برای تغییر ولتاژ خروجی آنالوگ باید مقدار موجود در رجیستر D60 را به کانال CH5 ماژول DVP06XA ارسال نماییم.

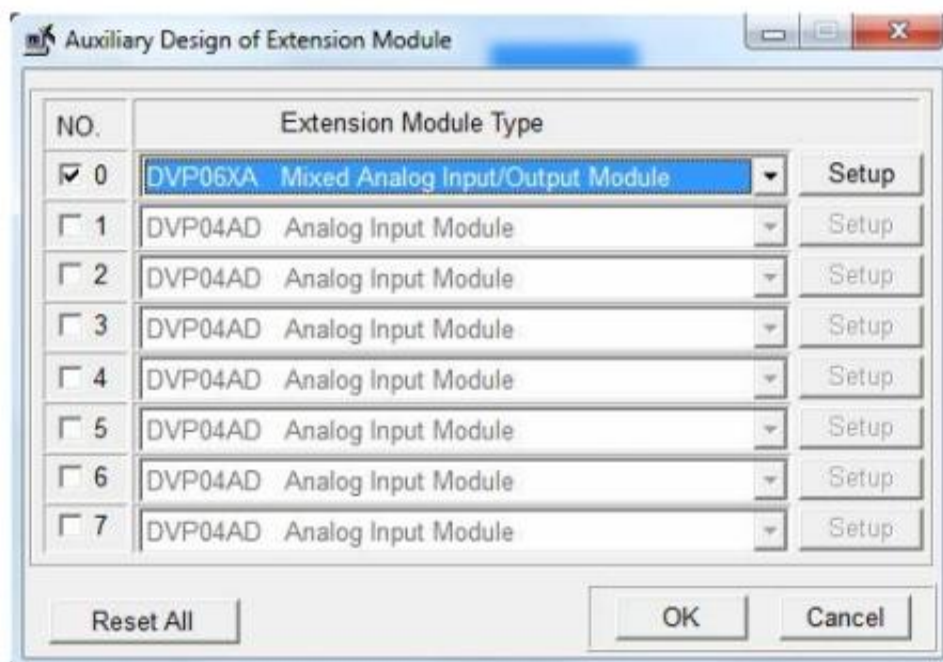
4-2 روش تنظیم پارامترهای ماژول DVP06XA توسط نرم‌افزار WPLSoft

1. ابتدا نرم‌افزار WPLSoft را اجرا نموده و سپس یک برنامه جدید را ایجاد کنید. بر روی آیکن

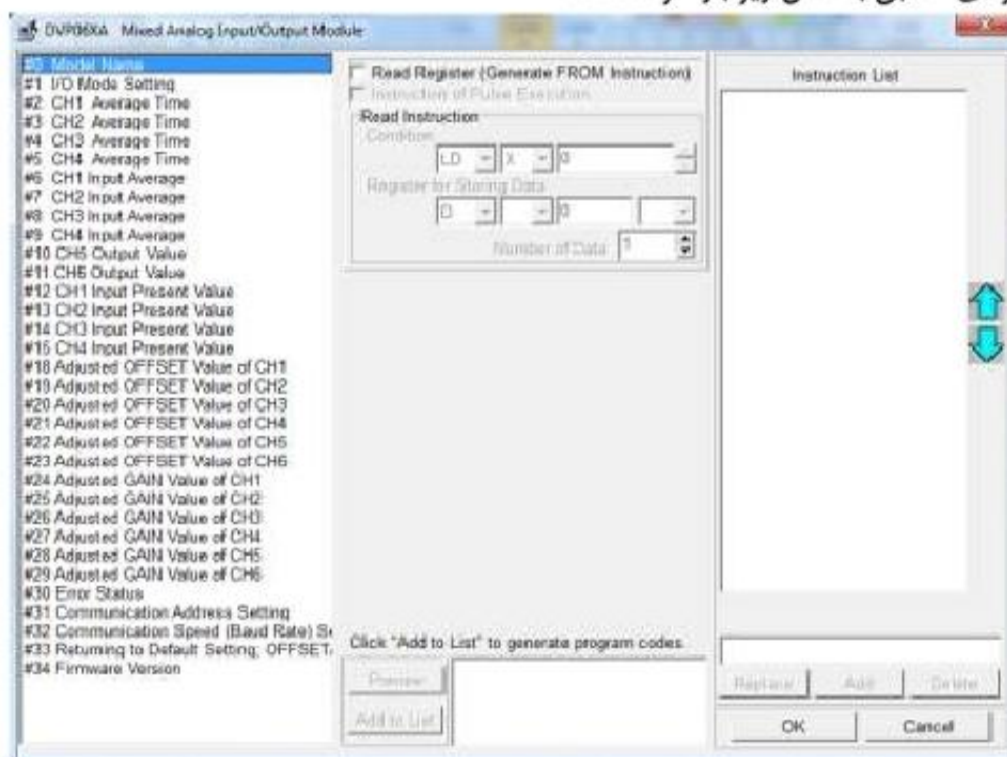
کلیک نمایید.



2. پنجره‌ای مطابق با شکل زیر باز خواهد شد. بر روی "0" NO. کلیک کرده و ماژول DVP06XA Mixed Analog Input/Output Module را انتخاب نمایید. سپس بر روی گزینه Setup برای اجرای تنظیمات ماژول کلیک نمایید.



3. پنجره‌ای مطابق با شکل زیر باز خواهد شد.



4. حال باید تنظیمات مازول را مطابق با توضیحات گفته شده در بالا برای این برنامه کاربردی اجرا

نمایید.

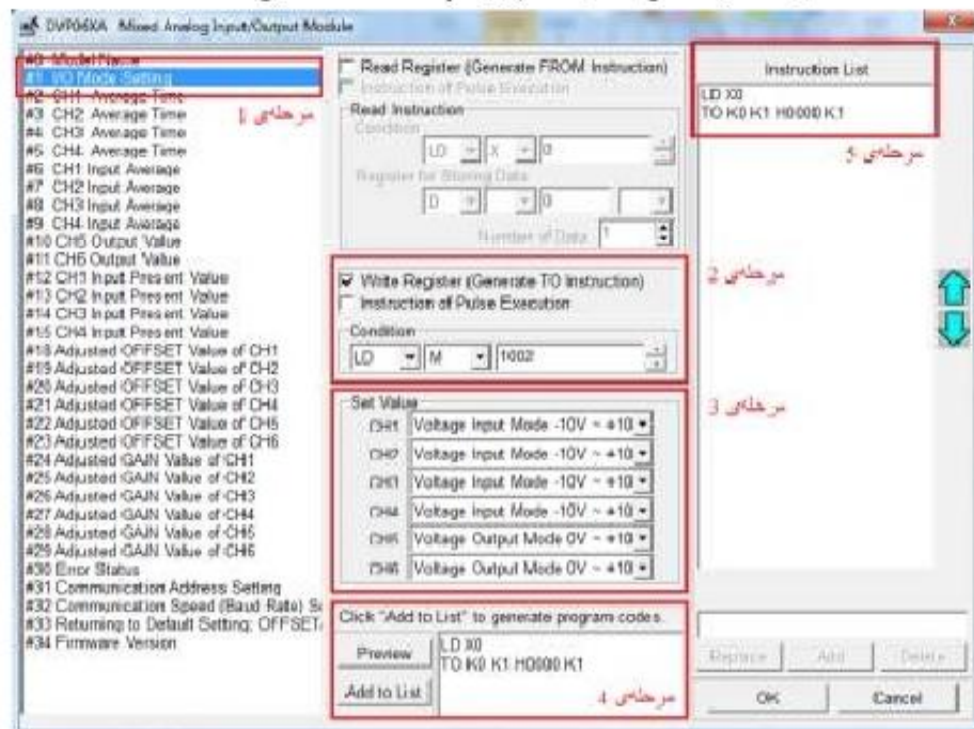
مرحله‌ی 1: گزینه‌ی #1 Set up Input/Output Mode را انتخاب کنید.

مرحله 2: جهت تولید دستورالعمل TO به صورت اتوماتیک، بر روی گزینه Write Register کلیک نموده و شرط اجرای دستورالعمل TO را برابر با LD M1002 تنظیم کنید.

مرحله 3: مد کاری کانال CH1~CH4 را برابر با Voltage Input Mode -10V ~ +10V و مد کاری کانال CH5~CH6 را برابر با Voltage Output Mode 0V ~ +10V تنظیم کنید.

مرحله 4: جهت اطمینان از درست بودن کدهای تولید شده توسط برنامه بر روی دکمه Preview کلیک نمایید.

مرحله 5: برای نمایش کدهای دستورالعمل به زبان برنامه‌نویسی Instruction List بر روی دکمه Add to List کلیک کنید. حال تنظیمات رجیستر CR#1 تکمیل شده است.



5. می‌خواهیم رجیستر کنترلی CR#2 را تنظیم نماییم. دقت داشته باشید که تنظیم این رجیستر کنترلی مانند تنظیمات رجیستر کنترلی CR#1 می‌باشد.

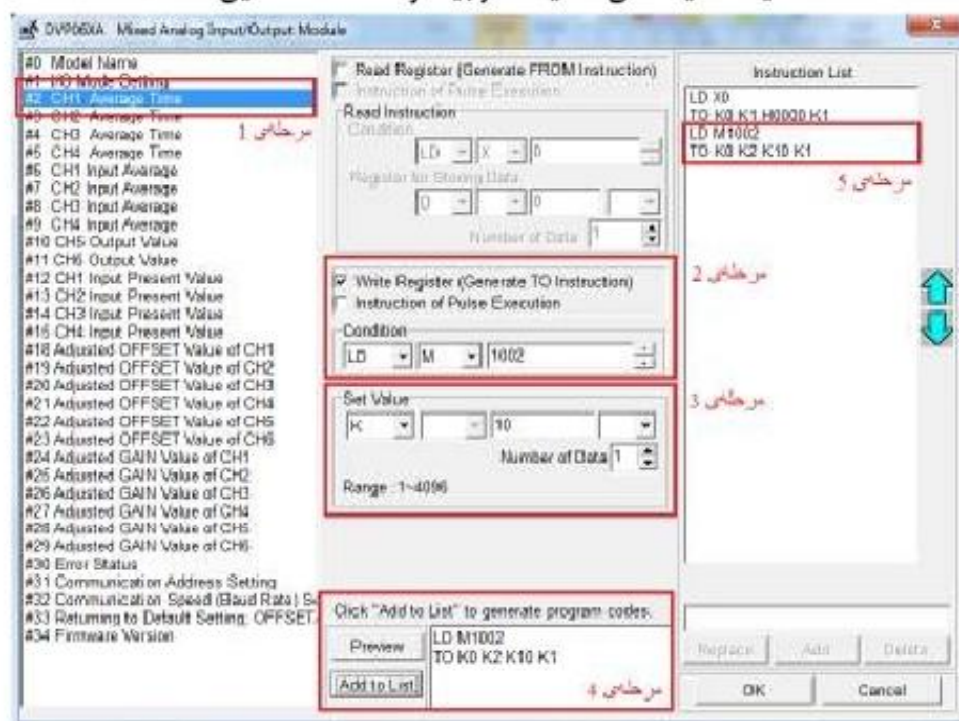
مرحله 1: گزینه CH1 Average Times #2 را انتخاب کنید.

مرحله 2: جهت تولید دستورالعمل TO به صورت اتوماتیک، بر روی گزینه Write Register کلیک نموده و شرط اجرای دستورالعمل TO را برابر با LD M1002 تنظیم کنید.

مرحله 3: در کادر Set value برای تنظیم مقدار مطلوب، مقدار دسیمال K10 را تنظیم کنید و سپس تعداد داده را برابر با 1 تنظیم نمایید.

مرحله‌ی 4: جهت اطمینان از درست بودن کدهای تولید شده توسط برنامه، بر روی دکمه Preview کلیک نمایید.

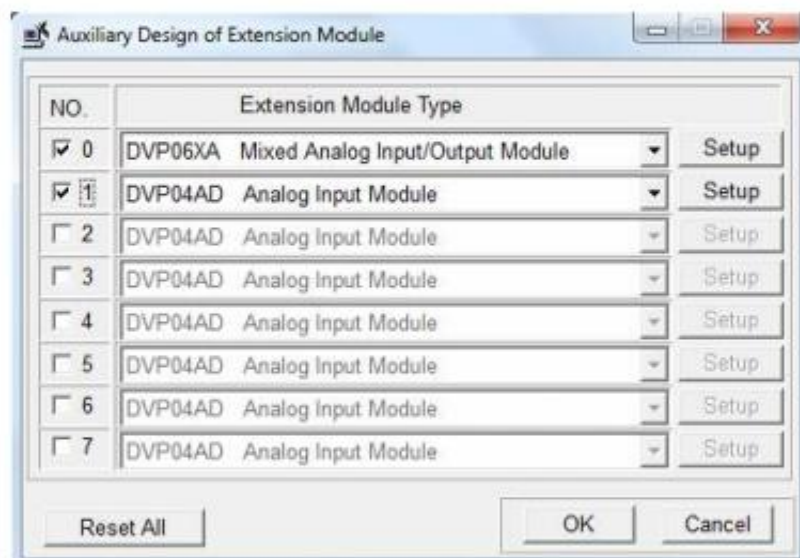
مرحله‌ی 5: برای نمایش کدهای دستورالعمل به زبان برنامه‌نویسی Instruction List بر روی دکمه‌ی Add to List کلیک کنید حال تنظیمات رجیستر CR#2 تکمیل شده است.



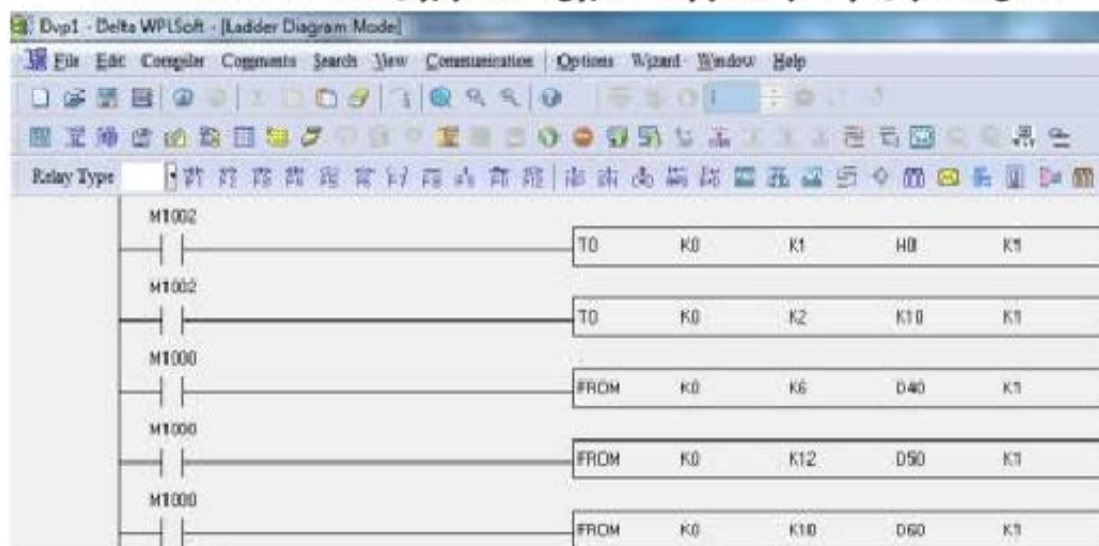
6. در صورت نیاز، می‌توانید مقادیر تنظیم شده را اصلاح کرده و یا یک دستورالعمل جدید را به زبان برنامه‌نویسی Instruction List اضافه کنید. برای مثال، اگر می‌خواهید مقدار خروجی آنالوگ موجود در رجیستر کنترلی CR#6 را اصلاح کنید، ابتدا باید بر روی آیتم مورد نظر در کادر Instruction List کلیک کرده و سپس مقدار K255 را بعنوان مثال به D0 تغییر دهید. در نهایت، برای ذخیره کردن تغییرات بوجود آمده باید بر روی دکمه "Replace" کلیک کنید.

7. مطابق با توضیحات گفته شده در بالا، باید دیگر پارامترهای CR را تنظیم نمایید.

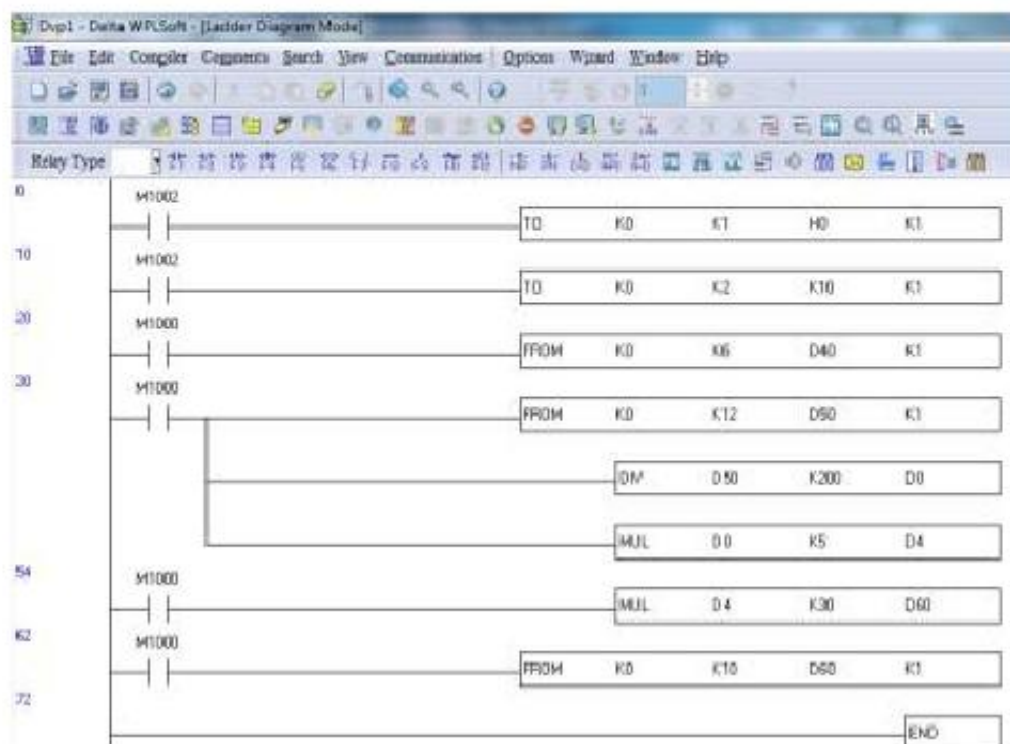
8. بعد از اینکه تنظیم تمام پارامترهای CR مربوط به مازول ترکیبی ورودی خروجی آنالوگ تکمیل شد، برای بازگشت به پنجره‌ی Auxiliary Design of Extension Module باید بر روی دکمه OK کلیک نمایید تا بتوانید تنظیمات دیگر مازول‌های توسعه‌ی خاص متصل شده به CPU را اجرا نمایید.



9. بعد از اینکه تنظیم تمام پارامترهای CR مربوط به ماژول‌های توسعه‌ی خاص متصل شده به CPU تکمیل شد، برای تولید برنامه مربوط به ماژول‌ها باید بر روی دکمه OK کلیک نمایید.




10. اگر نیاز به اصلاح قسمتی از کدهای تولید شده داشتید و یا نیاز بود که قسمت جدیدی به برنامه‌ی کنترلی اضافه نمایید، می‌توانید به صورت مستقیم در پنجره‌ی برنامه‌نویسی به زبان نردبانی در نرم‌افزار WPLSoft اصلاحات و تغییرات مورد نظر خود را اعمال نمایید.



4-3 تنظیمات ماژول DVP04PT در نرم افزار WPL Soft

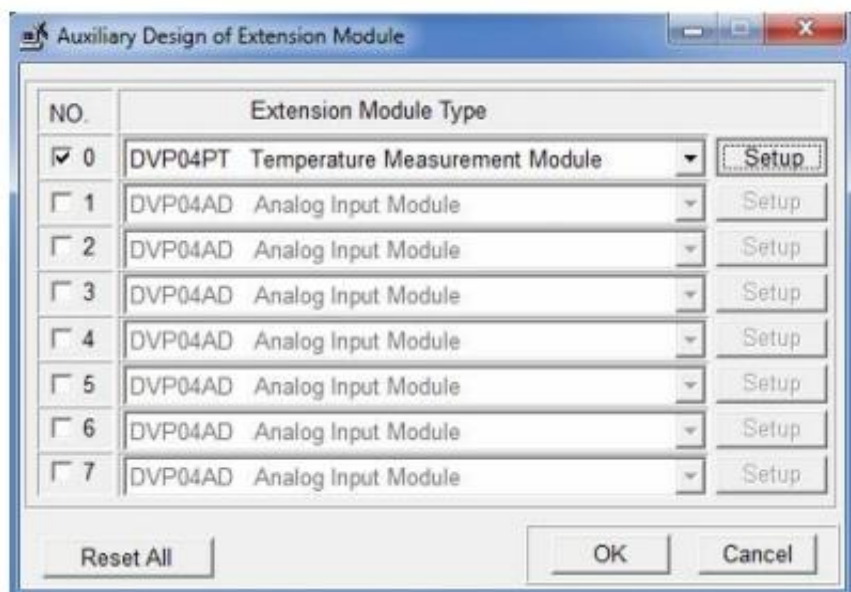
1. بعد از اینکه نرم افزار WPL Soft را باز کردید، باید یک پروژه‌ی جدید را ایجاد کنید. بر روی

آیکن  کلیک نمایید.

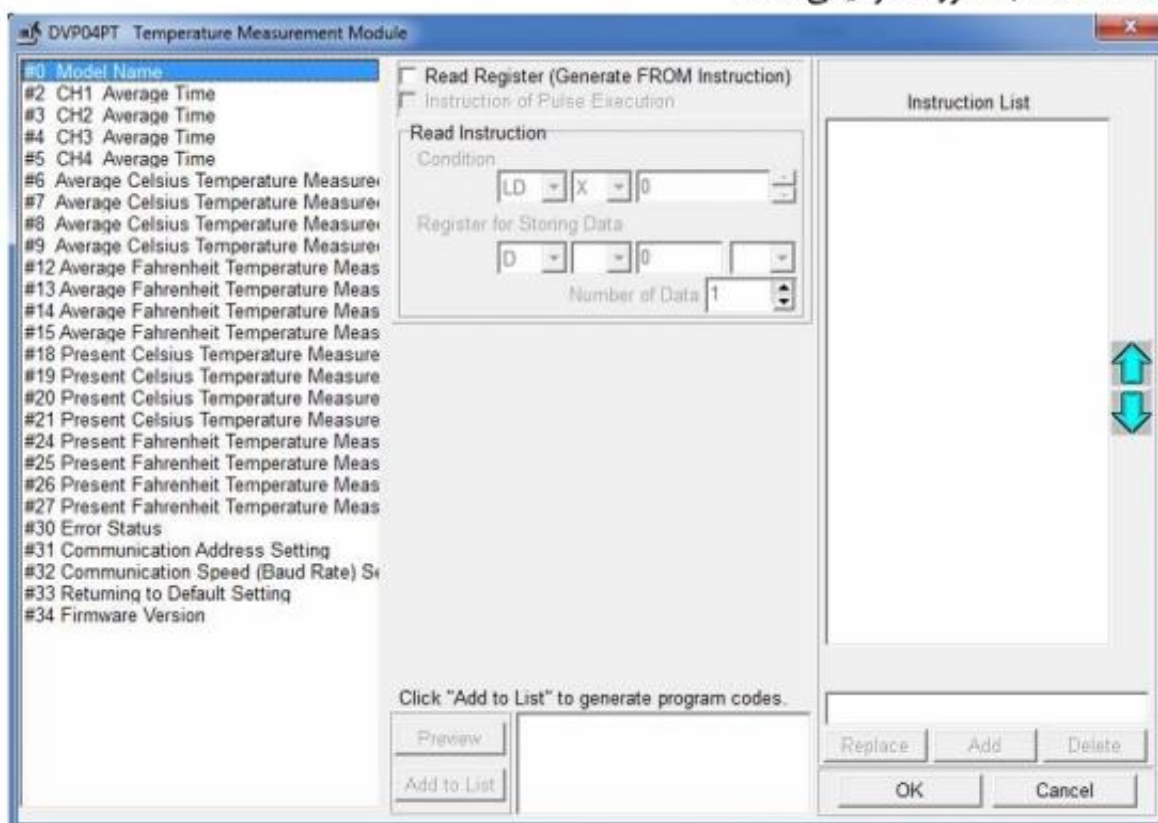


2. پنجره‌ای با نام Auxiliary Design of Extension Module مطابق با شکل زیر باز

خواهد شد. در این پنجره بر روی آیکن "0". NO. کلیک نموده و سپس گزینه‌ی DVP04PT Temperature Measurement Module انتخاب کنید.

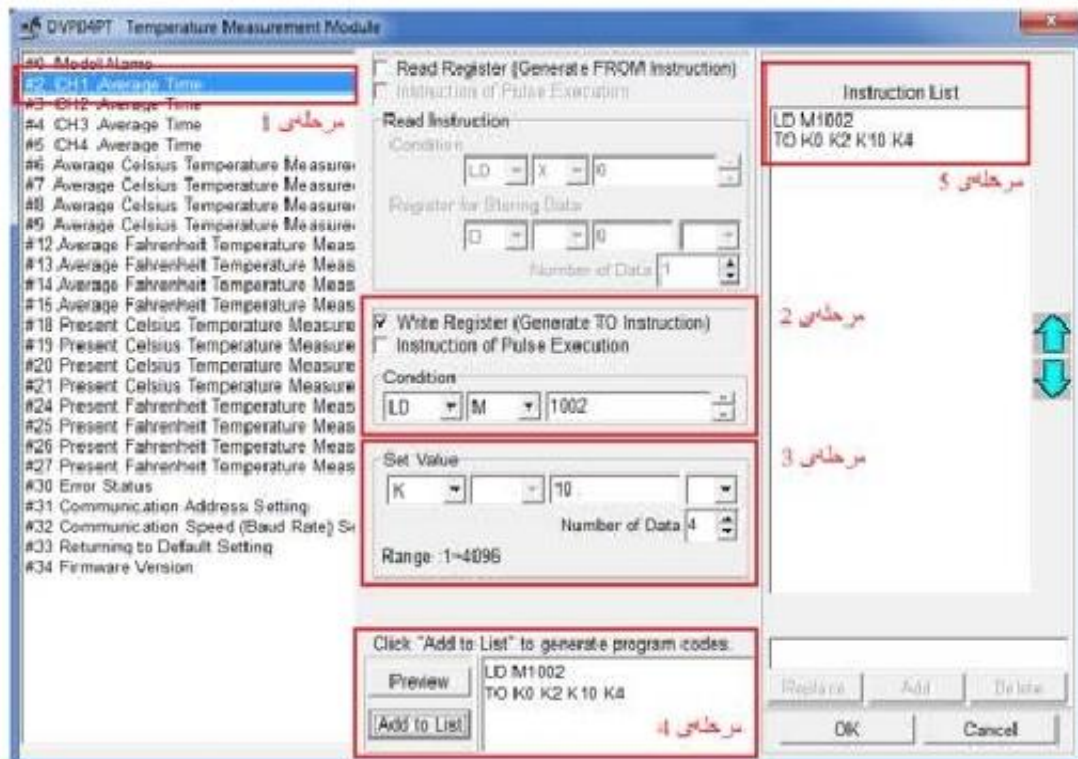


3. پنجره‌ای مطابق با شکل زیر باز خواهد شد که شامل تنظیمات رجیسترهای کنترلی ماژول DVP04PT به صورت گرافیکی است.



4. در این قسمت تنظیمات مربوط به برنامه‌ی کنترلی نوشته شده در بالا را توضیح خواهیم داد.
مرحله‌ی 1: در پنجره‌ی درختی گزینه‌ی #2 CH1 Average Time را انتخاب نمایید.

- مرحله‌ی 2: تیک گزینه‌ی **Write Register** را برای تولید دستورالعمل **TO** بزنید. شرط دستورالعمل **TO** را برابر با **LD M1002** تنظیم کنید.
- مرحله‌ی 3: مقدار **K10** را برای این رجیستر تنظیم نمایید. همچنین، تعداد داده را برابر با 4 تنظیم نمایید تا مقدار این رجیستر کنترلی و 3 رجیستر کنترلی بعدی برابر با **K10** تنظیم شود.
- مرحله‌ی 4: برای اطمینان از درست بودن کد تولید شده برای برنامه بر روی دکمه **Preview** کلیک نمایید.
- مرحله‌ی 5: برای تبدیل و نمایش کد تولید شده به زبان برنامه‌نویسی **Instruction List** بر روی دکمه **Add to List** کلیک نمایید. در این مرحله تنظیمات رجیسترهای کنترلی **CR#2~CR#5** تکمیل شده است.

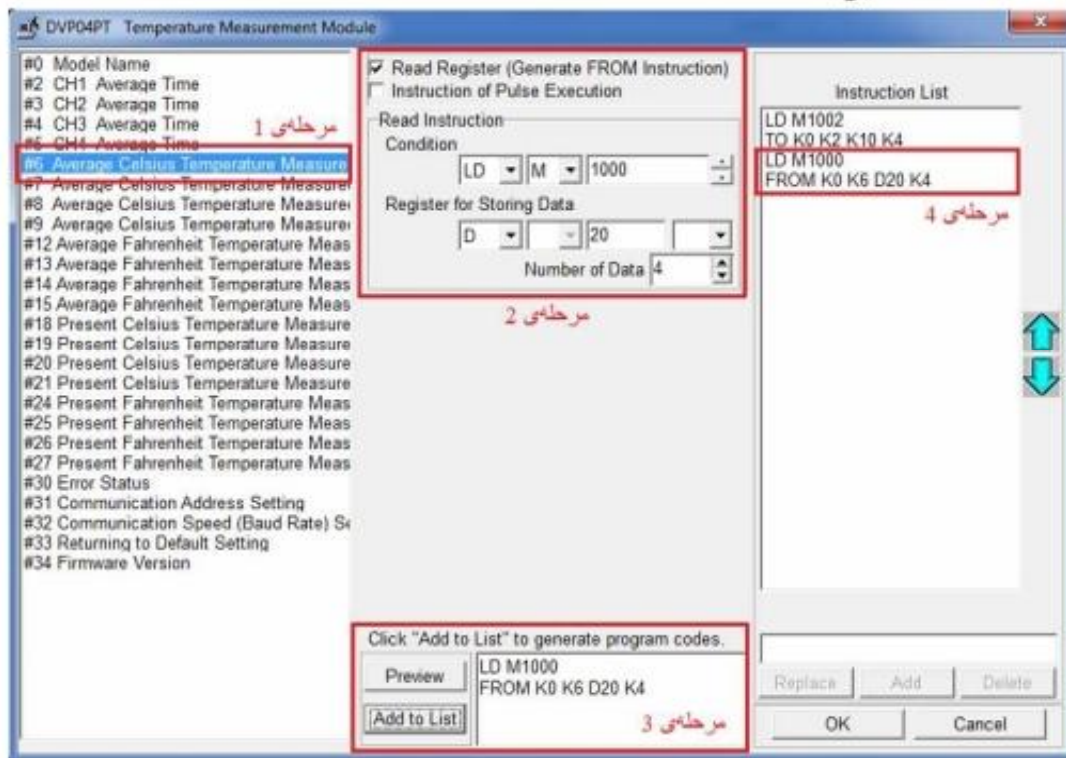


5. تنظیمات رجیسترهای کنترلی **CR#6~CR#9** مانند تنظیمات **CR#2~CR#5** می‌باشد.
- مرحله‌ی 1: از منوی درختی سمت چپ گزینه‌ی **#6 Average Celsius Temperature** **Measured at CH1** کلیک نمایید.
- مرحله‌ی 2: تیک گزینه‌ی **Read Register** را برای تولید دستورالعمل **FROM** بزنید. شرط اجرای دستورالعمل **FROM** را برابر با **LD M1000** تنظیم نمایید. رجیستر ذخیره‌ی مقدار دما در **CPU** را برابر با **D20** تنظیم نمایید. همچنین، تعداد داده را برابر با 4 تنظیم کنید تا رجیسترهای

D20~D23 به صورت اتوماتیک به رجیسترهای کنترلی CR#6~CR#9 برای ذخیره‌سازی مقدار دمای هر چهار کانال CH1~CH4 متصل شوند.

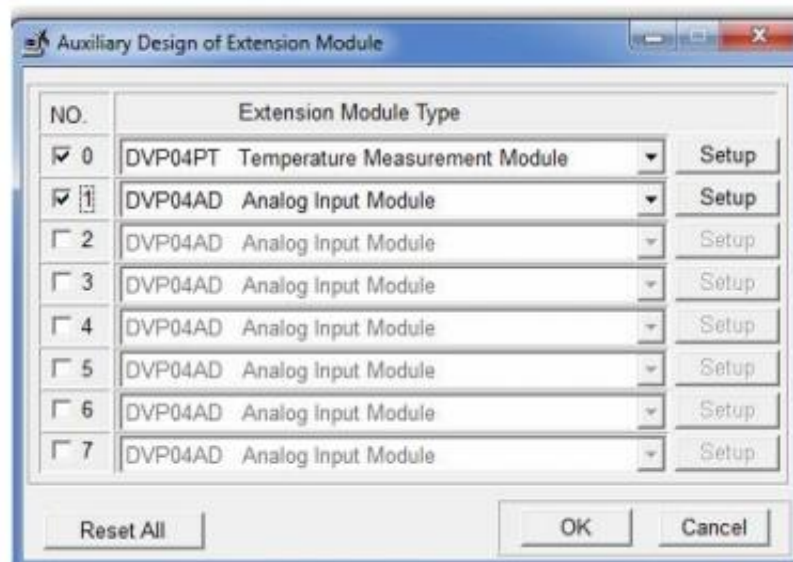
مرحله‌ی 3: برای اطمینان از درست بودن کد تولید شده برای برنامه‌ی PLC بر روی دکمه‌ی Preview کلیک نمایید.

مرحله‌ی 4: برای تبدیل و نمایش کد تولید شده به زبان برنامه‌نویسی متنی Instruction List روی دکمه Add to list کلیک نمایید. در این مرحله تنظیمات رجیسترهای کنترلی CR#6~CR#9 تکمیل شده است.



6. تنظیمات دیگر پارامترهای کنترلی CR مطابق با توضیحات گفته شده در بالا می‌باشد. حال دیگر پارامترهای مورد نیاز خود را تنظیم نمایید.

7. بعد از اینکه تنظیمات تمام پارامترها تکمیل شد، برای برگشت به صفحه‌ی Auxiliary Design of Extension Module بر روی دکمه OK کلیک نمایید. در این پنجره، می‌توانید تنظیمات دیگر ماژول‌های توسعه را انجام دهید.



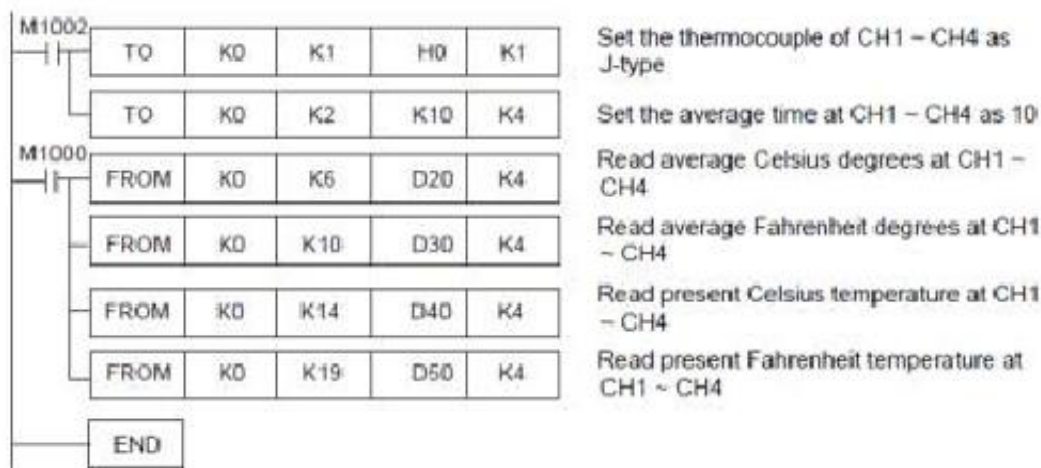
8. بعد از اینکه تمام تنظیمات ماژول‌های توسعه را انجام دادید، برای تولید برنامه به زبان Ladder بر روی دکمه OK کلیک نمایید.



9. اگر نیاز بود که برنامه‌ی کنترلی بیشتری را به برنامه اضافه کنید، می‌توانید مستقیماً برنامه‌ی کنترلی خود را در پنجره‌ی دیاگرام نردبانی در نرم‌افزار WPLSoft بنویسید.

4-5 کاربرد ماژول آنالوگ دما DVP04TC

یکی از کاربردهای این ماژول، اندازه‌گیری دما توسط ترموکوپل می‌باشد. در شکل زیر، یک نمونه برنامه به صورت دیاگرام نردبانی نشان داده شده است.



ابزارهای مورد استفاده در این مثال به شرح زیر است :

D20~D23: از این رجیسترها برای نمایش مقدار میانگین گرفته شده‌ی دمای کانال‌های CH1~CH4 برحسب سانتی‌گراد استفاده شده است.

D30~D33: از این رجیسترها برای نمایش مقدار میانگین گرفته شده‌ی دمای کانال‌های CH1~CH4 برحسب فارنهایت استفاده شده است.

D40~D43: از این رجیسترها برای نمایش دمای جاری (لحظه به لحظه) کانال‌های CH1~CH4 برحسب سانتی‌گراد استفاده شده است.

D50~D53: از این رجیسترها برای نمایش دمای جاری (لحظه به لحظه) کانال‌های CH1~CH4 برحسب فارنهایت استفاده شده است.

عملکرد برنامه به شرح زیر می‌باشد :

وقتی که PLC از مد STOP به RUN تغییر وضعیت دهد، تعداد دفعات میانگین‌گیری از سیگنال‌های ورودی CH1~CH4 برابر با 10 تنظیم می‌شود.

مقدار دمای میانگین گرفته شده از آخرین ده دمای خوانده شده از کانال‌های CH1~CH4 برحسب سانتی‌گراد در رجیسترها D20~D23 ذخیره می‌شود.


مقدار دمای میانگین گرفته شده از آخرین ده دمای خوانده شده از کانال‌های CH1~CH4 برحسب فارنهایت در رجیسترهای D30~D33 ذخیره می‌شود.

مقدار دمای جاری (لحظه به لحظه) خوانده شده از کانال‌های CH1~CH4 برحسب سانتی‌گراد در رجیسترهای D40~D43 ذخیره می‌شود.

مقدار دمای جاری (لحظه به لحظه) خوانده شده از کانال‌های CH1~CH4 برحسب فارنهایت در رجیسترهای D50~D53 ذخیره می‌شود.

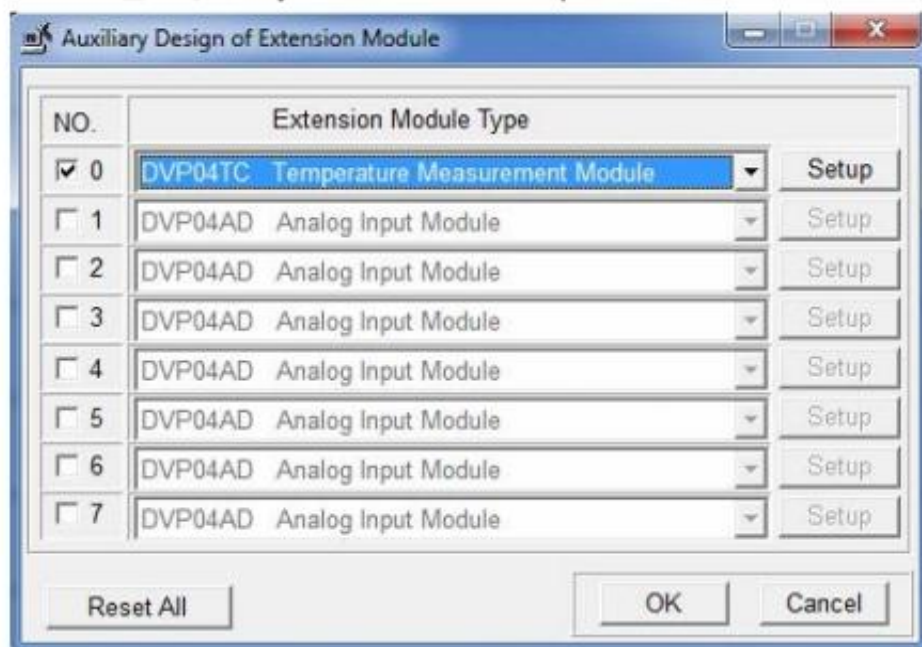
ماژول DVP04TC مقدار دمای بدست آمده (خوانده شده) را در رجیسترهای کنترلی CR ذخیره می‌کند. از این رو، فقط لازم است که برای خواندن و اندازه‌گیری دما، محتویات داخل رجیسترهای کنترلی CR را بخوانید. واحد اندازه‌گیری دما برحسب 0.1°C یا 0.1°F می‌باشد.

4-1-5 پیکربندی ماژول DVP04TC در نرم‌افزار WPL Soft

1. در نرم‌افزار WPL Soft یک صفحه برای تنظیم پارامترهای ماژول DVP04TC وجود دارد. برای باز کردن پنجره‌ی تنظیمات ماژول DVP04TC بر روی آیکن  کلیک نمایید.

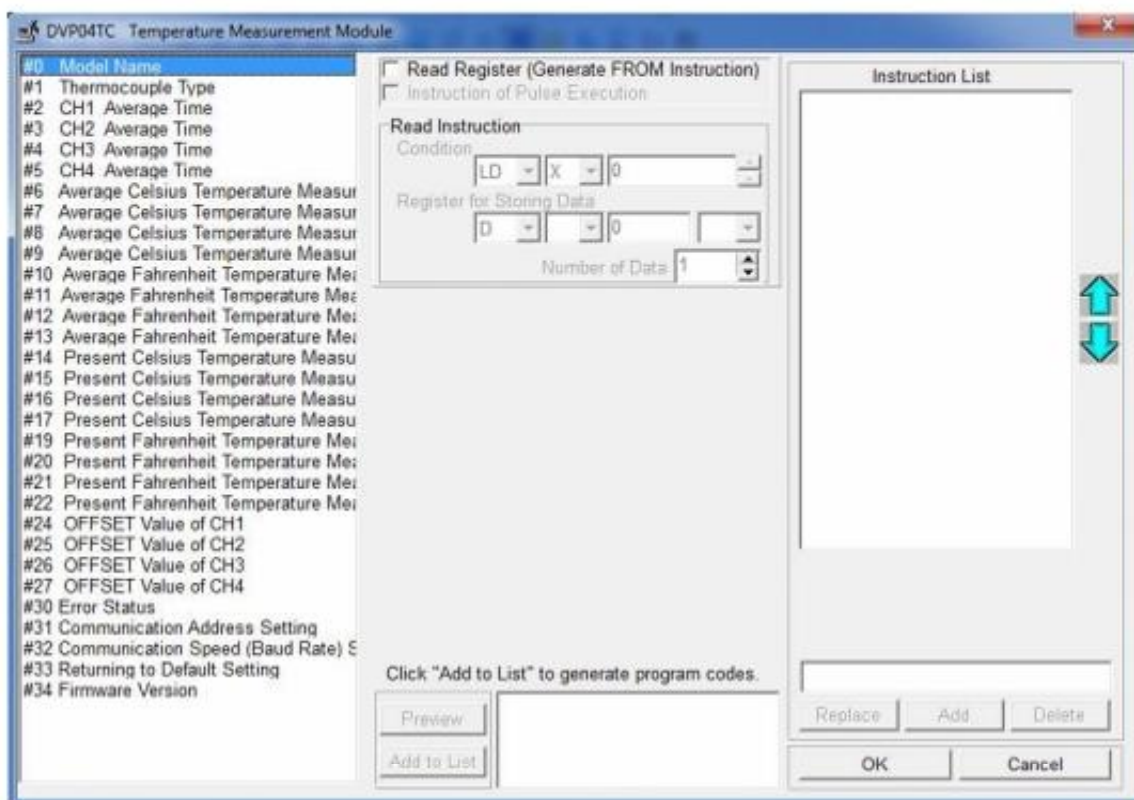


2. پنجره‌ی مطابق با شکل زیر به نام Auxiliary Design of Extension Module خواهد شد. در این پنجره بر روی "0". NO. کلیک نموده (در این مثال، فرض شده است که اولین ماژول توسعه‌ی خاص متصل شده به سمت راست CPU، ماژول DVP04TC می‌باشد) و گزینه DVP04TC Temperature Measurement Module را انتخاب نمایید.

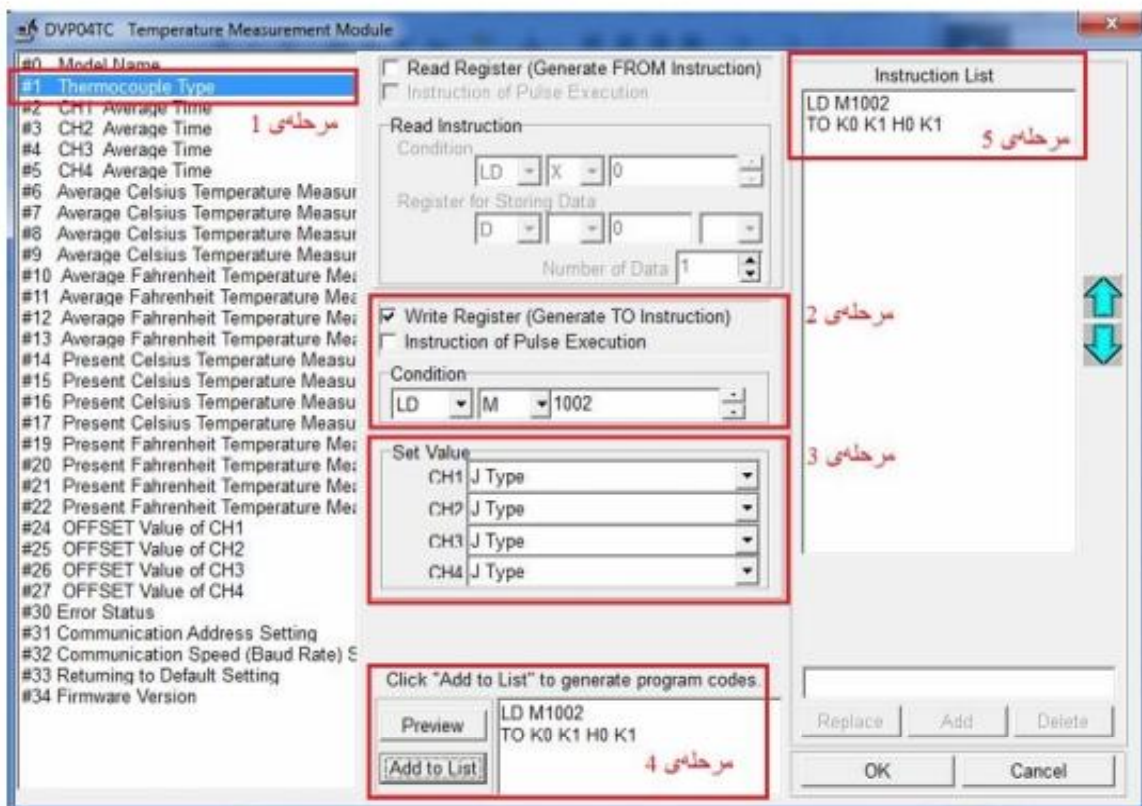


3. حال پنجره‌ی مطابق با شکل زیر، جهت تنظیم پارامترهای (رجیسترهای کنترلی) ماژول باز خواهد

شد.



4. دقت داشته باشید که توضیحات گفته شده برای مثال بالا می‌باشد.
- مرحله 1: گزینه **#1 Thermocouple Type** را انتخاب نمایید.
- مرحله 2: بر روی گزینه **Write Register** کلیک نمایید تا دستورالعمل **TO** تولید شود. شرط **LD M1002** را برای فعال سازی دستورالعمل **TO** وارد نمایید.
- مرحله 3: نوع ترموکوپل کانال‌های **CH1~CH4** را برابر با **J Type** تنظیم نمایید.
- مرحله 4: بر روی دکمه **Preview** کلیک نمایید تا کد برنامه برای دستورالعمل **TO** تولید شود.
- مرحله 5: برای نمایش کدهای دستورالعمل به زبان **Instruction List** بر روی گزینه **Add to List** کلیک نمایید. در این مرحله تنظیم رجیستر **CR#1** تکمیل شده است.



5. تنظیم رجیستر CR#2 نیز مانند تنظیم رجیستر CR#1 می‌باشد.

مرحله 1: گزینه CH1 Average Time #2 را انتخاب کنید.

مرحله 2: برای تولید دستورالعمل TO بر روی گزینه Write Register کلیک نمایید. شرط

LD M1000 را برای فعال کردن این دستورالعمل تنظیم کنید.

مرحله 3: مقدار تنظیمی (set value) را برابر با K10 تنظیم کرده و number of data را

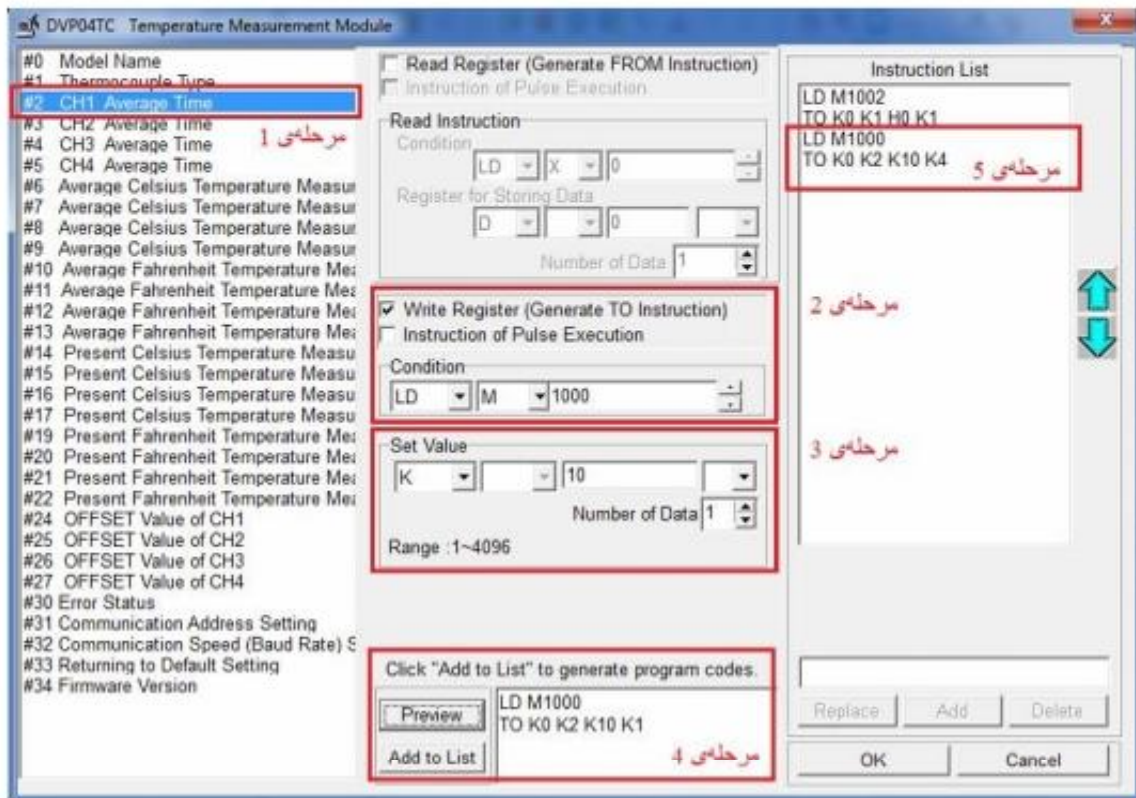
برابر با 1 تنظیم نمایید.

مرحله 4: اگر تنظیمات انجام شده درست بودند، برای تولید دستورالعمل TO بر روی گزینه

Preview کلیک نمایید.

مرحله 5: برای نمایش کدهای دستورالعمل به زبان Instruction List بر روی گزینه Add

to List کلیک نمایید. تنظیم رجیسترهای CR#2~CR#5 نیز به همین ترتیب می‌باشد.



6. تنظیم رجیسترهای کنترلی CR#6~CR#9 نیز مانند تنظیم رجیسترهای CR#2~CR#5 است.

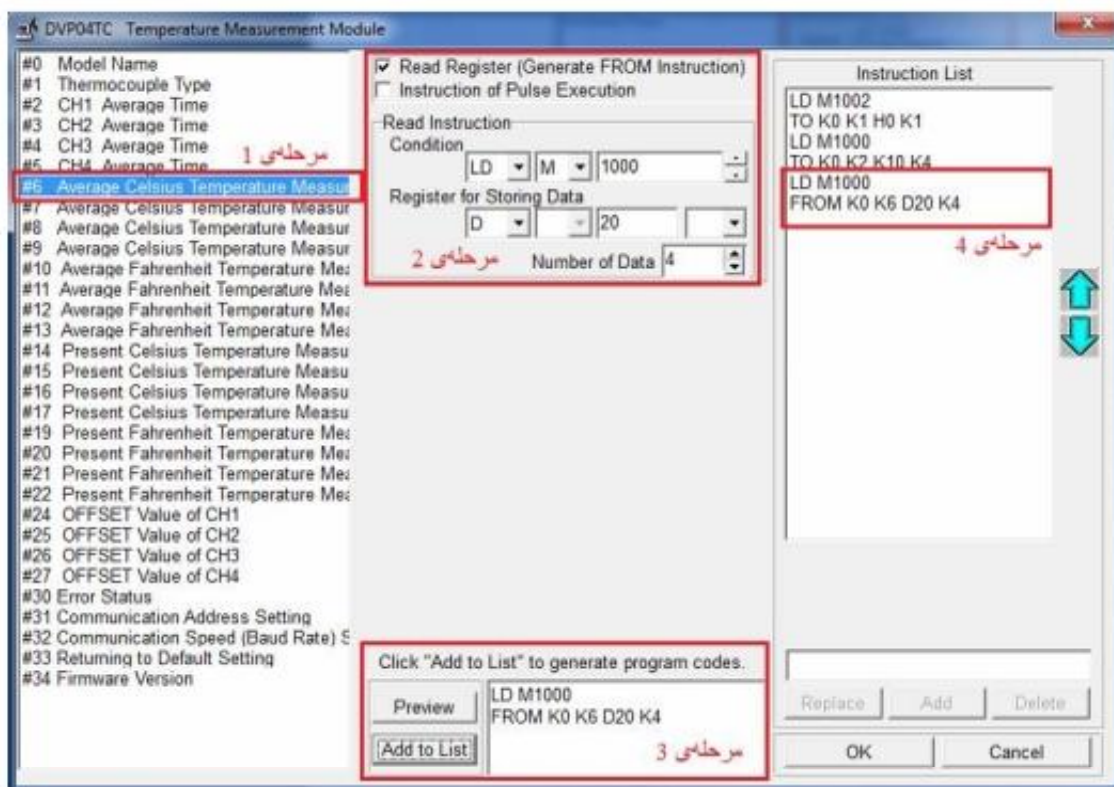
مرحله 1: گزینه‌ی CH1 Average Celsius Temperature Measured at را انتخاب کنید.

مرحله 2: برای تولید دستورالعمل FROM بر روی گزینه Read Register کلیک نمایید. شرط LD M1000 را برای فعال کردن این دستورالعمل تنظیم کنید.

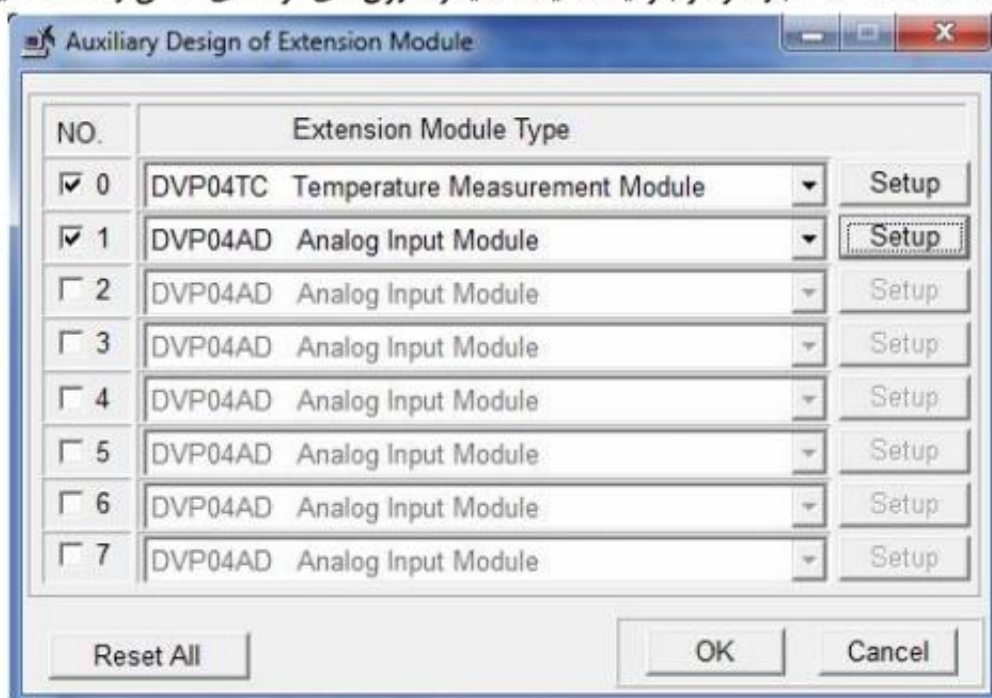
مرحله 3: رجیستر D20 را برای ذخیره‌سازی مقدار میانگین گرفته شده از دمای اندازه‌گیری شده برای کانال CH1 برحسب سانتی‌گراد تنظیم نموده و number of data را برابر با 4 تنظیم کنید.

مرحله 4: اگر تنظیمات انجام شده درست بودند، برای تولید دستورالعمل FROM بر روی گزینه Preview کلیک نمایید.

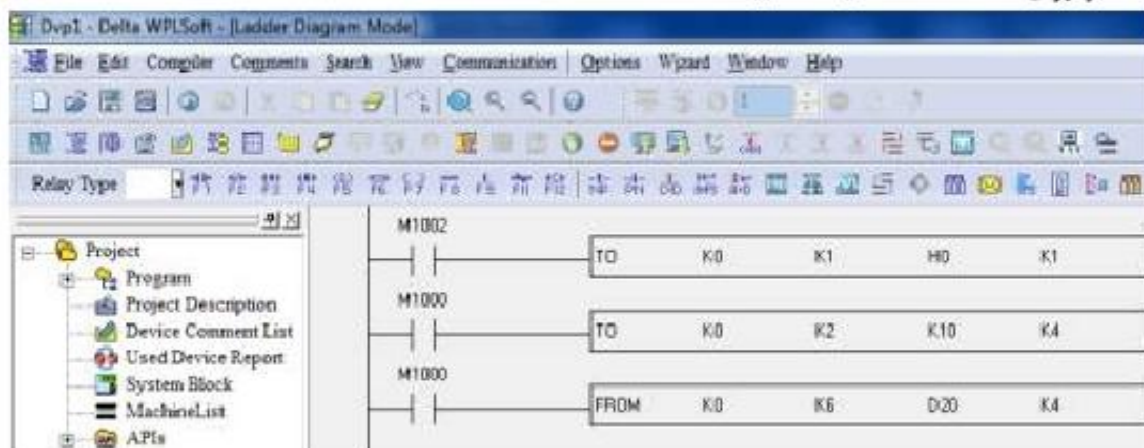
مرحله 5: برای نمایش کدهای دستورالعمل به زبان Instruction List بر روی گزینه Add to List کلیک نمایید. تنظیم رجیسترهای CR#6~CR#9 نیز به همین ترتیب می‌باشد.



7. تنظیم دیگر پارامترهای CR مطابق با مراحل گفته شده در بالا می‌باشد. بعد از اینکه تنظیم تمام پارامترها تکمیل شد، بر روی دکمه OK کلیک نمایید تا پنجره‌ی Auxiliary Design of Extension Module باز شود و بتوانید تنظیمات دیگر ماژول‌های توسعه‌ی خاص را ادامه دهید.



8. بعد از اینکه تنظیم پارامترهای کنترلی تمام ماژول‌های توسعه‌ی خاص تکمیل شد، برای تولید برنامه بر روی دکمه OK کلیک نمایید.



9. اگر لازم است که برنامه‌ی کنترلی دیگری را به این برنامه اضافه کنید، می‌توانید برنامه کنترلی را به صورت مستقیم در پنجره‌ی دیاگرام نردبانی در نرم‌افزار WPLSOFT وارد نمایید.

4- 6- توابع ماژول لودسل

4- 6- 1 اندازه‌گیری وزن

شما معمولاً می‌توانید برای اندازه‌گیری وزن یک شیء از اندازه‌گیری وزن خالص (net weight) و یا اندازه‌گیری وزن ناخالص (gross weight) استفاده نمایید. اندازه‌گیری وزن خالص (net weight) به معنی وزن خود محصول، یعنی وزن واقعی محصول بدون در نظر گرفتن بسته‌بندی خارجی می‌باشد. در اینجا وزن بسته‌بندی، وزن ظرف بسته‌بندی و یا صفحه‌ی ترازو tare weight نامیده می‌شود. اندازه‌گیری وزن ناخالص (gross weight) به معنی مجموع وزن net weight + tare weight می‌باشد.

Tare weight وزن بسته‌بندی (وزن هاپر، صفحه‌ی ترازو و یا هر ظرفی که محصول برای وزن کشی داخل آن ریخته می‌شود) می‌باشد.

Net weight وزن محصول بدون در نظر گرفتن (باستثنای) وزن بسته‌بندی می‌باشد.

Gross weight مجموع وزن محصول و وزن بسته‌بندی محصول است.

$$\text{Gross weight} = \text{Net weight} + \text{Tare weight}$$

برای مثال، وزن یک محصول برابر با 10kg بوده و وزن کارتونی که محصول در آن قرار می‌گیرد، برابر با 0.2kg است. در اینجا وزن خالص و ناخالص برابر با 10.2kg می‌باشد.

$$\text{Net weight} = 10\text{kg}, \text{tare weight} = 0.2\text{kg}, \text{gross weight} = 10.2\text{kg}$$

رجیسترهای کنترلی وابسته به این سه وزن عبارتند از:
 از رجیستر CR#6 برای خواندن وزن بسته‌بندی tare weight استفاده می‌شود.
 از رجیستر CR#7 برای انتخاب وزن مجموع (gross weight) و یا وزن محصول (net weight) استفاده می‌شود.
 از رجیسترهای CR#8, 9 برای نوشتن و خواندن وزن بسته‌بندی (tare weight) استفاده می‌شود.

مثال

در این مثال، از مقدار اندازه‌گیری شده در کانال CH1 بعنوان وزن محصول استفاده شده است و کانال CH2 غیرفعال می‌باشد. اگر وزن بسته‌بندی را از پیش می‌دانید، می‌توانید از مرحله خواندن وزن بسته‌بندی (tare weight) صرف‌نظر کنید.
 1. خواندن وزن بسته‌ی (tare weight) شامل سه مرحله می‌باشد که عبارتند از:
 مرحله‌ی 1: ابتدا باید مقدار H'0000 هگز را در داخل رجیستر کنترلی CR#7 برای کانال CH1 بنویسید.

مرحله‌ی 2: بسته‌بندی را بر روی لودسل متصل شده به کانال CH1 قرار دهید.
 مرحله‌ی 3: مقدار H'0001 را در داخل رجیستر کنترلی CR#6 جهت در نظر گرفتن وزن بسته‌بندی بعنوان tare weight وارد نمایید.
 2. مقدار CR#7 را برابر با H'00F1 تنظیم کنید. 1 به معنی اندازه‌گیری وزن net weight در کانال CH1 بوده و F به معنی غیرفعال بودن کانال CH2 می‌باشد.

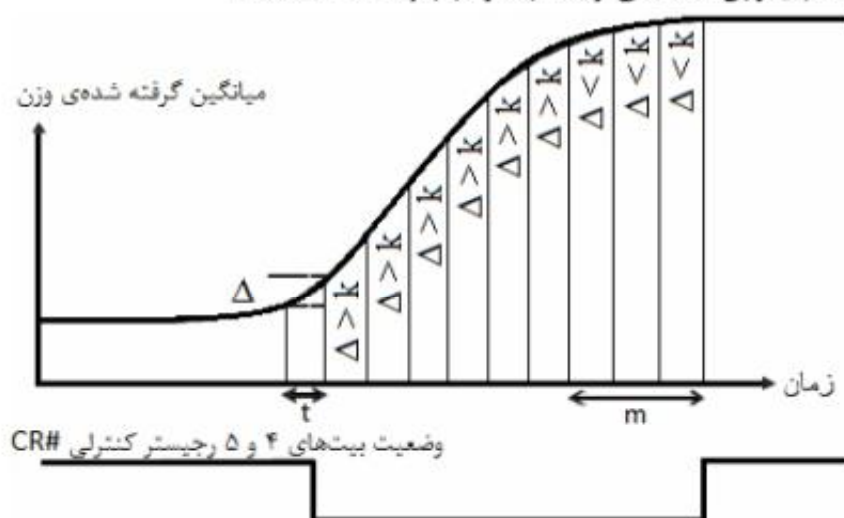
4-6-2 بررسی پایداری وزن (Standstill check)

وقتی که یک جسمی را برای اندازه‌گیری وزن آن جسم بر روی لودسل قرار داده‌اید، می‌توانید از تابع Standstill check برای اطلاع از پایداری و بدون پرش بودن وزن اندازه‌گیری شده‌ی آن شیء استفاده نمایید.

اگر مقدار اندازه‌گیری شده‌ی وزن در داخل محدوده‌ی تنظیم شده برای بررسی پایداری وزن (CR#18, 19) توسط کاربر (شما) بود، بیت‌های b4, b5 از رجیستر کنترلی CR#50 فعال (1) خواهند شد.

اگر مقدار اندازه‌گیری شده‌ی وزن، خارج از محدوده‌ی تنظیم شده برای بررسی پایداری وزن توسط کاربر بود، بیت‌های b4, b5 از رجیستر کنترلی CR#50 غیرفعال (0) خواهند شد. وقتی که وزن اندازه‌گیری شده به محدوده‌ی تنظیم شده بازگردد، وضعیت این بیت‌ها 1 خواهد شد.

بعنوان مثال، زمان اندازه‌گیری وزن 10ms ، تعداد دفعات بررسی پایداری وزن 10 بار و محدوده‌ی تنظیمی برای بررسی پایداری وزن 1000 است. وقتی که محدوده‌ی بررسی پایداری وزن از 1000 تجاوز کند، مقدار اندازه‌گیری شده ناپایدار خواهد بود و وضعیت بیت‌های $b4$ ، $b5$ رجیستر کنترلی CR#50 برابر با 0 خواهند شد. وقتی که زمان اندازه‌گیری وزن در داخل محدوده‌ی $(10 \times 10\text{ms})$ باشد و محدوده‌ی وزن به داخل مقدار بررسی پایداری (1000) وزن بازگشت، بیت‌های $b4$ ، $b5$ رجیستر کنترلی CR#50 برابر با 1 خواهند شد. پیشنهاد ما به شما این است که ابتدا قبل از کنترل وزن محصول، بررسی نمایید که آیا مقدار اندازه‌گیری شده‌ی وزن به اندازه‌ی کافی پایدار است، یا خیر. برای بررسی پایداری وزن، شما می‌توانید از نمودار زیر استفاده نمایید:



سمبل‌های مورد استفاده در این نمودار عبارتند از:

Δ به معنی تغییر میانگین گرفته شده از وزن می‌باشد.

t به معنی زمان تنظیم شده‌ی اندازه‌گیری وزن توسط کاربر می‌باشد.

k به معنی محدوده‌ی تنظیم شده‌ی بررسی پایداری وزن توسط کاربر می‌باشد.

m به معنی تعداد دفعات تنظیم شده برای بررسی پایداری وزن می‌باشد.

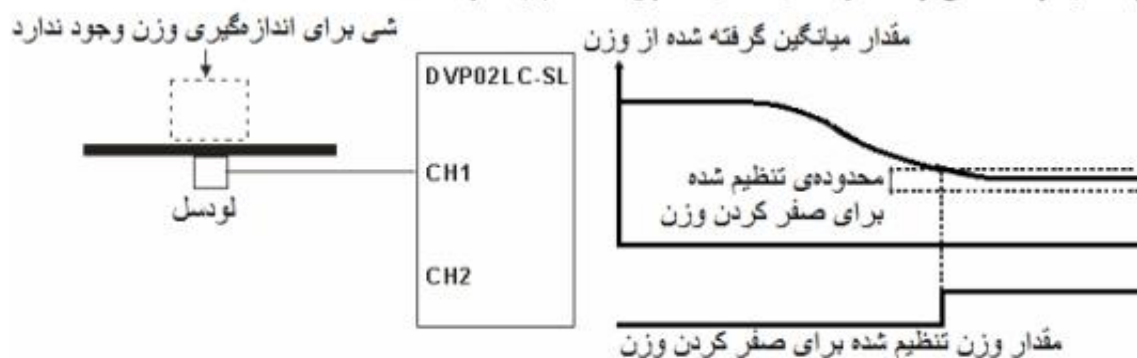
رجیسترهای کنترلی وابسته به بررسی پایداری وزن به شرح زیر است:

از رجیسترهای کنترلی CR#16, 17 برای تنظیم تعداد دفعات بررسی پایداری وزن استفاده می‌شود.

از رجیسترهای کنترلی CR#18, 19 برای تنظیم محدوده‌ی بررسی پایداری وزن استفاده می‌شود.

4-2-3 تعیین کردن نقطه (وزن) صفر

می‌توانید از تابع تعیین نقطه‌ی (وزن) صفر برای دانستن اینکه آیا شیء روی لودسل وجود دارد و یا اینکه شیء از روی لودسل برداشته شده است، استفاده نمایید. اگر بیت‌های b4, b5 رجیستر کنترلی CR#50 برابر با 1، و بیت‌های b0, b1 نیز برابر با 1 شوند، بیانگر این است که شیء بر روی لودسل وجود ندارد و شما می‌توانید مرحله‌ی بعدی کنترل سیستم را اجرا نمایید.

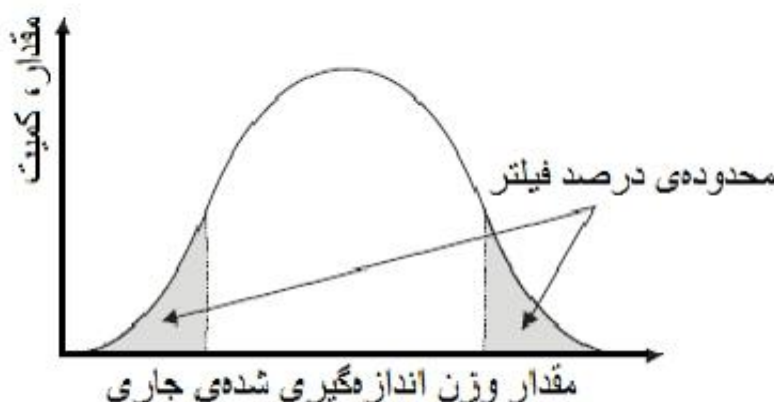


رجیسترهای کنترلی وابسته به تعیین (تنظیم) نقطه (وزن) صفر عبارتند از :

از رجیسترهای کنترلی CR#37, 38, 39, 40 جهت تنظیم محدوده‌ی بالا و محدوده‌ی پایین بررسی نقطه (وزن) صفر استفاده می‌شود.

4-6-4 فیلتر کردن وزن

مقدار میانگین گرفته شده، یک مقدار متوالی بدست آمده از مجموع مقادیر خوانده شده می‌باشد. به علت اینکه فاکتورهای خارجی بر روی وزن اندازه‌گیری تأثیر می‌گذارند و نمی‌توان از آنها اجتناب کرد، مقدار وزن خوانده شده ممکن است به صورت پالسی بوده و دارای پرش باشد و تغییرات میانگین گرفته شده خیلی شدید باشد. از تابع فیلترینگ برای جلوگیری از خواندن مقدار وزن به صورت پالسی استفاده می‌شود. از این رو مقدار میانگین وزن بدست آمده تحت تأثیر مقدار خوانده شده به صورت پالسی و مقطع (پرش وزن) نخواهد شد. شرط فعال کردن تابع فیلترینگ این است که تعداد دفعات میانگین‌گیری وزن بزرگتر و یا مساوی با 30 بار بوده و درصد فیلتر 10 تا 50٪ تنظیم شده باشد.



رجیسترهای کنترلی وابسته به تابع فیلترینگ عبارتند از:
 از رجیسترهای کنترلی 44, CR#43 برای تنظیم درصد فیلتر استفاده می‌شود.
 از رجیسترهای کنترلی 46, CR#45 برای تنظیم مقدار میانگین فیلتر استفاده می‌شود.

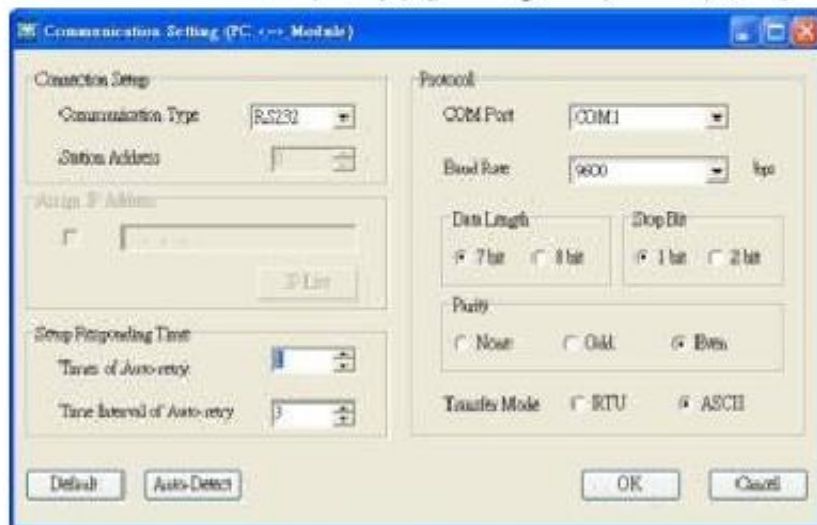
4-6-5 تنظیمات ماژول DVP02LC-SL در نرم‌افزار LCSOft توسط کامپیوتر

1. برای اجرای تنظیمات اولیه‌ی پارامترهای ماژول DVP02LC-SL ابتدا باید ماژول را توسط کابل واسط و پورت RS-232 به کامپیوتر متصل نمایید. نحوه‌ی اتصال ماژول به کامپیوتر در همین فصل توضیح داده شده است.

2. بعد از نصب نرم‌افزار LCSOft، آن را باز کنید. از منوی Option گزینه‌ی Communication Setting را انتخاب نمایید.



3. تنظیمات پارامترهای ارتباطات را مطابق با شکل زیر انجام دهید:



- تنظیم زمان پاسخ (Setup Responding Time)

تعداد دفعات سعی مجدد در برقراری ارتباط ماژول با کامپیوتر بین 0~50 بار می‌باشد. مقدار تنظیمی (پیش فرض) آن 1 است. مدت زمان سعی مجدد در برقراری ارتباطات بین ماژول و کامپیوتر بین 1~20 می‌باشد. مقدار پیش فرض آن 3 می‌باشد.

- تنظیم پورت ارتباط با کامپیوتر (Protocol)

از گزینه COM Port برای انتخاب پورت ارتباطی که قرار است کامپیوتر را به ماژول متصل کنید، استفاده می‌شود.

از گزینه Baud Rate برای تنظیم سرعت تبادل داده (9600, 19200, 38400, 57600 or 115200) استفاده می‌شود.

از گزینه Data Length برای تنظیم طول داده (7bit or 8bit) استفاده می‌شود. وقتی که مد ارسال و مخابره داده برابر با مد RTU تنظیم شده باشد، طول داده به صورت اتوماتیک برابر با 8bit تنظیم خواهد شد.

از گزینه Stop Bit برای تنظیم تعداد بیت توقف (1bit or 2bit) استفاده می‌شود.


از گزینه Parity برای تنظیم بیت توازن (None, Odd or Even) استفاده می‌شود.

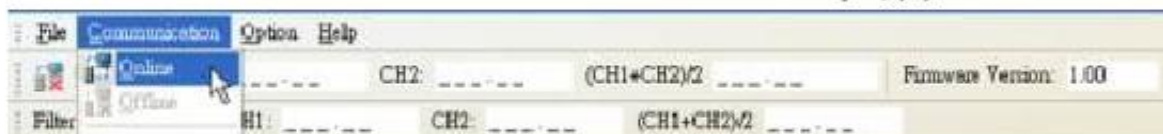
از گزینه Transfer Mode برای تنظیم مد ارسال داده (RTU or ASCII) استفاده می‌شود.

- آشکارسازی اتوماتیک (Auto-detect)

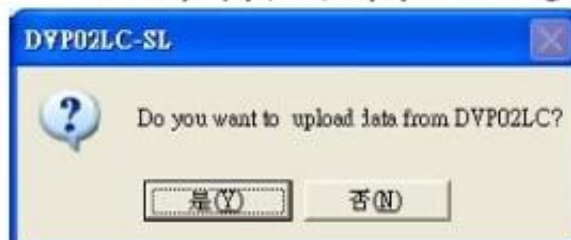
برای آشکارسازی اتوماتیک تنظیمات ماژول لودسل و اتصال آن به کامپیوتر باید بر روی دکمه‌ی Auto-Detect کلیک نمایید. در این حالت اگر ماژول به کامپیوتر متصل شده باشد، توسط کامپیوتر شناسایی شده و تنظیمات ارتباطات ماژول و ارتباطات کامپیوتر با یکدیگر هماهنگ خواهند شد.



4. بعد از اینکه تنظیمات ارتباطات تکمیل شد، بر روی آیکن  کلیک کرده و یا از منوی Communication گزینه‌ی Online را انتخاب نمایید. در این حالت، ارتباط بین نرم‌افزار و ماژول DVP02LC-SL برقرار خواهد شد.



5. وقتی که بر روی گزینه‌ی Online کلیک کردید، یک پنجره برای آپلود کردن اطلاعات ماژول ظاهر خواهد شد. و از شما خواهد پرسید که آیا شما می‌خواهید اطلاعات ماژول را بر روی کامپیوتر آپلود نمایید. برای آپلود کردن اطلاعات ماژول باید بر روی دکمه‌ی OK کلیک نمایید. در این حالت، اطلاعات داخل ماژول جایگزین اطلاعات موجود در کامپیوتر خواهد شد.



6. تنها نیاز است که فقط یک بار وارد وضعیت آنلاین شوید، صفحه‌ی اصلی نرم‌افزار بلافاصله اطلاعات ماژول DVP02LC-SL را نشان خواهد داد. این اطلاعات شامل ورژن نرم‌افزار نصب شده بر روی ماژول، مقادیر میانگین گرفته شده‌ی وزن از کانال‌های CH1 و CH2، مقدار میانگین گرفته شده‌ی وزن از کانال CH1+CH2، مقادیر میانگین گرفته شده با فیلتر از کانال‌های CH1 و CH2 و مقدار میانگین گرفته شده‌ی وزن با فعال بودن فیلتر در کانال CH1+CH2 می‌باشد. شما می‌توانید بر روی یک مقدار کلیک نمایید تا یک پنجره با کاراکترهای بزرگتر به نمایش درآید.

CH1:	32767 g	CH2:	327.67 g	(CH1+CH2)/2	16547.335 g
------	---------	------	----------	-------------	-------------

بر روی مقدار نمایش داده شده برای کانال CH1 کلیک نمایید تا یک پنجره با کاراکترهای بزرگتر جهت نمایش وزن مطابق با شکل زیر ظاهر شود.



- مقادیر میانگین گرفته شده‌ی وزن با فعال بودن فیلتر



Filter Average Value: CH1: 327.67 g CH2: 327.67 g (CH1+CH2)/2 327.67 g

بر روی مقدار نمایش داده شده برای کانال CH1 کلیک نمایید تا یک پنجره مطابق با شکل زیر برای نمایش مقدار وزن میانگین گرفته شده با لحاظ کردن فیلتر به نمایش درآید.



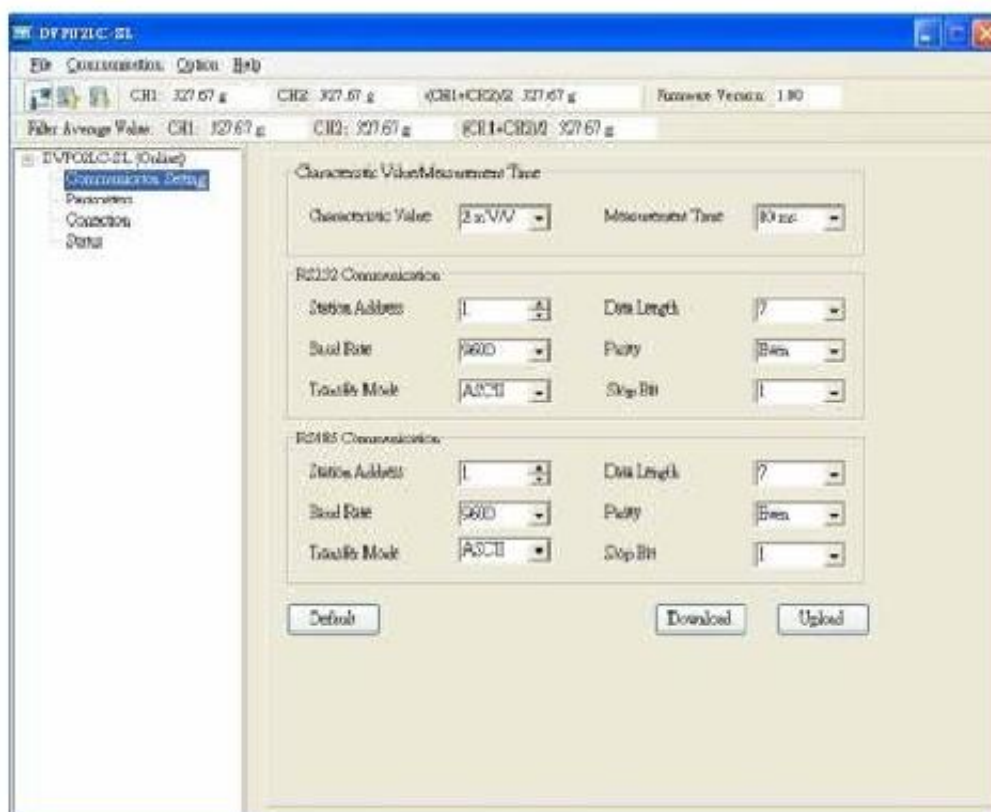
توجه:

وقتی که تابع فیلترینگ غیرفعال باشد، پنجره‌ای برای نمایش مقدار وزن میانگین گرفته شده با فیلتر ظاهر خواهد شد که مقدار 32767 در آن نوشته شده است. با فعال کردن تابع فیلترینگ و تنظیم تعداد دفعات میانگین‌گیری بیشتر از 30 بار و تنظیم درصد فیلتر، این پنجره مقدار وزن میانگین گرفته شده‌ی فیلتر شده را نمایش خواهد داد.

7. برای خواندن اطلاعات ماژول توسط نرم‌افزار بر روی دکمه  و برای نوشتن تنظیمات جدید در داخل پارامترهای ماژول DVP02LC-SL باید بر روی آیکن  کلیک نمایید.

4-6-6 تنظیمات ارتباطات شبکه

در پنجره‌ی Communication Setting می‌توانید فرمت ارتباطات پورت RS-232 و RS-485، مقدار مشخصه‌ی (eigenvalue) و زمان نمونه‌برداری را تنظیم نمایید. وقتی که تمام تنظیمات را انجام دادید، برای دانلود کردن تنظیمات پارامترها به ماژول DVP02LC-SL بر روی آیکن  کلیک نموده و یا برای نمایش مقادیر پارامترهای ماژول DVP02LC-SL در نرم‌افزار بر روی گزینه‌ی Upload کلیک نمایید. برای بازگرداندن مقادیر تمام پارامترها به مقدار پیش‌فرض کارخانه باید بر روی گزینه‌ی Default کلیک نمایید.



- مقدار مشخصه / زمان اندازه‌گیری

مقدار مشخصه: مقدار مشخصه را در ماژول لودسل با نام *eigenvalue* و یا *Characteristic Value* معرفی می‌کنند. مقدار مشخصه‌ی لودسل را باید در رجیستر کنترلی *CR#2* تنظیم نمایید. شما می‌توانید یکی از مشخصه‌های 1 یا 2 یا 4 و یا 6mV/V را انتخاب نمایید. مقدار پیش‌فرض این رجیستر 2mV/V می‌باشد.

زمان اندازه‌گیری: زمان اندازه‌گیری وزن را می‌توان توسط رجیستر کنترلی *CR#3* تنظیم کرد. مقدار این زمان قابل انتخاب بوده و می‌توان یکی از زمان‌های 80ms، 40ms، 20ms، 10ms، 2ms و یا 80ms را انتخاب کرد. مقدار پیش‌فرض این رجیستر 80ms می‌باشد.

- ارتباطات شبکه‌ی RS-232

آدرس ایستگاه (*Station address*): آدرس ایستگاه همان آدرس گره‌ی ماژول لودسل در شبکه‌ی RS-232 می‌باشد که می‌توان این آدرس را در رجیستر کنترلی *CR#52* تنظیم کرد. محدوده‌ی قابل تنظیم این رجیستر کنترلی بین 1~255 می‌باشد. مقدار پیش‌فرض این رجیستر 1 است.

سرعت تبادل داده (*Bad rate*): سرعت تبادل داده‌ی ارتباطات RS-232 را می‌توان توسط رجیستر کنترلی *CR#53* تنظیم کرد. یکی از سرعت‌های 38400، 19200 bps، 9600 bps را می‌توان

bps, 57600 bps و یا 115200 bps را می‌توان انتخاب کرد. مقدار پیش‌فرض این رجیستر 9600 bps می‌باشد.

مد ارسال داده (Transfer mode): توسط بیت 15 رجیستر کنترلی CR#53 مد ارسال داده را می‌توان تعیین کرد. دو مد ارسال داده‌ی RTU و یا ASCII قابل انتخاب می‌باشند که مد ارسال داده‌ی پیش‌فرض ASCII است.

طول داده (Data length): توسط بیت 3 رجیستر کنترلی CR#52 می‌توان طول داده را انتخاب کرد. طول داده‌ی قابل انتخاب 7bit یا 8bit می‌باشد. مقدار پیش‌فرض 7bit است. وقتی که مد ارسال داده برابر با RTU تنظیم شده باشد، طول داده به صورت اتوماتیک برابر با 8bit تنظیم خواهد شد. بیت توازن (Parity): توسط دو بیت b0 و b1 رجیستر کنترلی CR#53 می‌توان بیت توان شبکه RS-232 را انتخاب کرد. سه انتخاب odd, none یا even قابل دسترس می‌باشد. مقدار پیش‌فرض این رجیستر برابر با even تنظیم شده است.

بیت توقف (Stop bit): توسط بیت b2 رجیستر کنترلی CR#53 می‌توان تعداد بیت توقف را تنظیم کرد. دو انتخاب 0 بیت و یا 1 بیت قابل دسترس می‌باشد. تعداد بیت‌های توقف پیش‌فرض برابر با 1 بیت می‌باشد.

- ارتباطات شبکه‌ی RS-485

آدرس ایستگاه (Station address): آدرس ایستگاه همان آدرس گره‌ی ماژول لودسل در شبکه‌ی RS-485 می‌باشد که می‌توان این آدرس را در رجیستر کنترلی CR#54 تنظیم کرد. محدوده‌ی قابل تنظیم این رجیستر کنترلی بین 1~255 می‌باشد. مقدار پیش‌فرض این رجیستر 1 است. سرعت تبادل داده (Bad rate): سرعت تبادل داده‌ی ارتباطات RS-485 را می‌توان توسط رجیستر کنترلی CR#55 تنظیم کرد. یکی از سرعت‌های 9600 bps, 19200 bps, 38400 bps, 57600 bps و یا 115200 bps را می‌توان انتخاب کرد. مقدار پیش‌فرض این رجیستر 9600 bps می‌باشد.

مد ارسال داده (Transfer mode): توسط بیت 15 رجیستر کنترلی CR#55 مد ارسال داده را می‌توان تعیین کرد. دو مد ارسال داده‌ی RTU و یا ASCII قابل انتخاب می‌باشند که مد ارسال داده‌ی پیش‌فرض ASCII است.

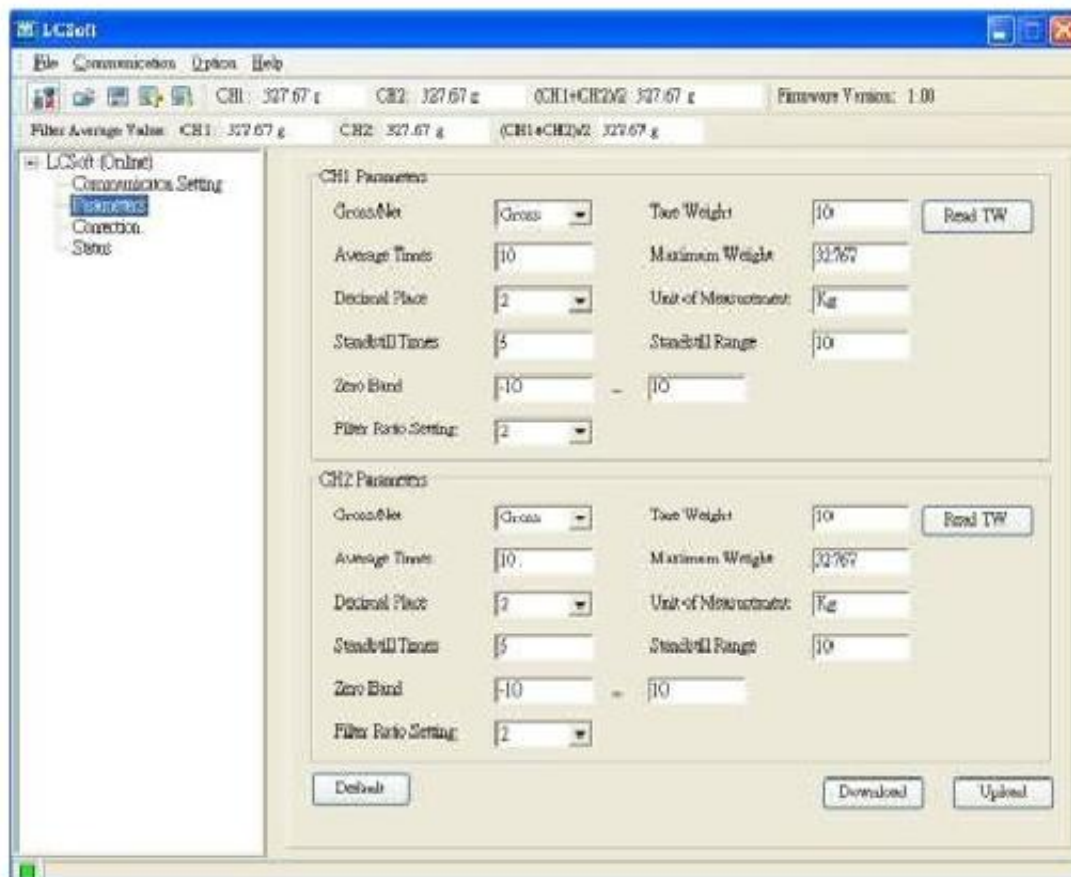
طول داده (Data length): توسط بیت 3 رجیستر کنترلی CR#54 می‌توان طول داده را انتخاب کرد. طول داده‌ی قابل انتخاب 7bit یا 8bit می‌باشد. مقدار پیش‌فرض 7bit است. وقتی که مد ارسال داده برابر با RTU تنظیم شده باشد، طول داده به صورت اتوماتیک برابر با 8bit تنظیم خواهد شد.

بیت توازن (Parity): توسط دو بیت b0 و b1 رجیستر کنترلی CR#55 می‌توان بیت توازن شبکه RS-485 را انتخاب کرد. سه انتخاب odd, none یا even قابل دسترس می‌باشد. مقدار پیش‌فرض این رجیستر برابر با even تنظیم شده است.

بیت توقف (Stop bit): توسط بیت b2 رجیستر کنترلی CR#55 می‌توان تعداد بیت توقف را تنظیم کرد. دو انتخاب 0 بیت و 1 بیت قابل دسترس می‌باشد. تعداد بیت‌های توقف پیش‌فرض برابر با 1 بیت می‌باشد.

4-6-7 تنظیمات پارامترهای ماژول لودسل

از طریق پنجره‌ی تنظیمات پارامترهای لودسل شما مجاز به تنظیم پارامترهای کانال CH1 و کانال CH2 ماژول می‌باشید. در این پنجره می‌توانید پارامترهای tare weight, net weight, gross weight تعداد دفعات میانگین‌گیری، تنظیم حداکثر وزن، تنظیم واحد اندازه‌گیری وزن، تنظیم تعداد اعشار (ممیز)، تنظیم محدوده و تعداد دفعات برای بررسی بدون حرکت بودن و بدون پرش بودن وزن، تنظیم بررسی وزن صفر و تنظیم درصد فیلتر را اجرا کرده و یا مشاهده نمایید. وقتی که تمام تنظیمات را انجام دادید، برای دانلود پارامترها به ماژول بر روی گزینه‌ی download کلیک نمایید. برای نمایش مقادیر پارامترهای ماژول DVP02LC-SL بر روی کامپیوتر باید بر روی گزینه‌ی Upload کلیک نمایید.



- وزن خالص و ناخالص (Gross/Net): توسط پارامتر CR#7 می‌توانید انتخاب کنید که وزن ناخالص (gross weight) و یا وزن خالص (net weight) نمایش داده شود.

- وزن ظرف (Tare weight): توسط رجیسترهای کنترلی CR#8, CR#9 می‌توانید وزن ظرفی که بر روی لودسل قرار دارد را برای کانال‌های CH1 و CH2 تنظیم نمایید. محدوده‌ی قابل تنظیم برای این پارامترها بین -32767 تا $+32767$ می‌باشد. مقدار پیش‌فرض این پارامتر 0 است.

- تعداد دفعات میانگین‌گیری (Average times): توسط رجیسترهای کنترلی CR#10 و CR#11 می‌توانید تعداد دفعات میانگین‌گیری برای خواندن وزن جسم را تنظیم نمایید. محدوده‌ی قابل تنظیم بین 1 تا 100 بار می‌باشد. مقدار پیش‌فرض میانگین‌گیری برابر با 10 تنظیم شده است. یعنی مازول 10 بار وزن را خوانده و از این 10 بار میانگین گرفته و مقدار میانگین را بعنوان وزن نمایش می‌دهد.

- حداکثر وزن (Maximum weight): توسط رجیسترهای کنترلی CR#35 و CR#35 می‌توان حداکثر وزن قابل اندازه‌گیری لودسل را برای کانال‌های CH1 و CH2 تنظیم کرد. وقتی که

وزن اعمال شده به لودسل از این وزن تجاوز کند، یک خطا بوجود خواهد آمد. محدوده‌ی قابل تنظیم برای این رجیسترها بین $32767+ \sim 32768-$ می‌باشد که مقدار پیش‌فرض آنها برابر با 32767 تنظیم شده است.

- واحد اندازه‌گیری وزن (Unit of measurement): توسط رجیسترهای کنترلی CR#22 و CR#23 می‌توان واحد اندازه‌گیری وزن کانال CH1 و توسط رجیسترهای کنترلی CR#24 و CR#25 می‌توان واحد اندازه‌گیری وزن کانال CH2 را تنظیم کرد. واحدهای اندازه‌گیری وزن را می‌توانید گرم g، کیلوگرم kg، تن T تنظیم نمایید. البته شما می‌توانید واحد را مطابق با نیازتان تنظیم کنید. بعنوان مثال، از لودسل می‌توانید برای اندازه‌گیری نیرو استفاده کرده و واحد آن را نیوتون متر N.m تنظیم نمایید. برای تنظیم واحد، می‌توانید حداکثر 4 کاراکتر را تنظیم نمایید. واحد پیش‌فرض تنظیم شده کیلوگرم kg می‌باشد.

- تعداد دفعات وقفه در اندازه‌گیری وزن (Standstill times): تعداد دفعات بررسی وقفه در وزن‌گیری را می‌توان توسط رجیسترهای کنترلی CR#16 و CR#17 تنظیم نمود. محدوده‌ی قابل تنظیم بین 1~500 و مقدار پیش‌فرض آن 5 است.

- محدوده‌ی بررسی وقفه در اندازه‌گیری وزن (Standstill range): محدوده‌ی بررسی وقفه در اندازه‌گیری وزن را می‌توان توسط CR#18 و CR#19 تنظیم کرد. محدوده‌ی قابل تنظیم بین 1~10000 بوده و مقدار پیش‌فرض آن برابر با 10 می‌باشد.

- محدوده‌ی وزن صفر (Zero band): برای بررسی وزن (نقطه‌ی) صفر ابتدا باید یک محدوده‌ی پایین و یک محدوده‌ی بالا را تعیین کرد. توسط رجیستر کنترلی CR#37 می‌توان محدوده‌ی بالا را برای کانال CH1، و توسط رجیستر کنترلی CR#39 می‌توان محدوده‌ی پایین را برای این کانال تنظیم کرد. توسط رجیستر کنترلی CR#38 می‌توان محدوده‌ی بالا را برای کانال CH2، و توسط رجیستر کنترلی CR#40 می‌توان محدوده‌ی پایین را برای همین کانال تنظیم کرد.

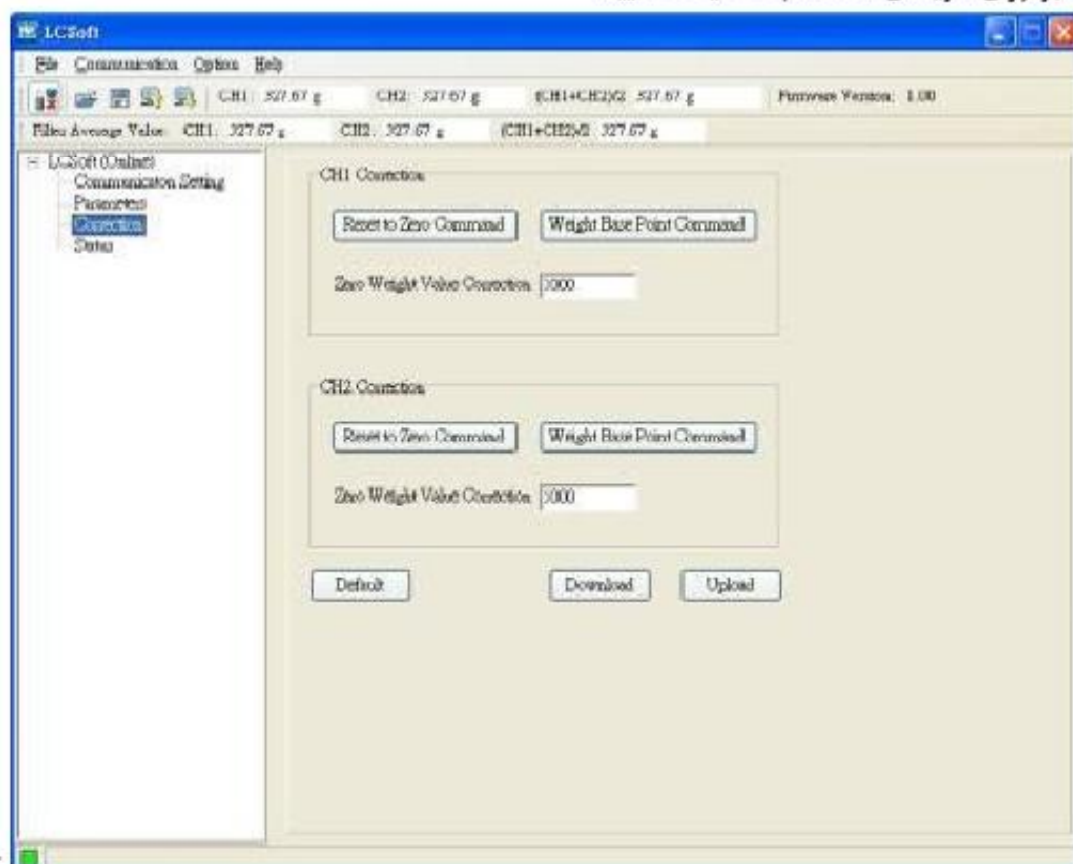
وقتی که وزن اندازه‌گیری شده داخل این محدوده باشد، کد وضعیت Zero bit مازول فعال (1) خواهد شد، که این کد بیانگر خالی بودن ترازو می‌باشد. محدوده‌ی قابل تنظیم برای این رجیسترها بین $32767+ \sim 32768-$ است و مقدار پیش‌فرض آنها برابر با $10+ \sim 10-$ تنظیم شده است.

- تنظیم نسبت فیلتر (Filter ratio setting): توسط رجیسترهای کنترلی CR#43 و CR#44 می‌توان درصد فیلتر را تنظیم کرد. محدوده‌ی قابل تنظیم آن بین 1~5 است. مقدار میانگین فیلتر برای کانال‌های CH1 و CH2 مانند مقدار وزن میانگین گرفته شده در نرم‌افزار بر روی نوار ابزار

نمایش داده خواهد شد. همچنین، برای مشاهده‌ی مقدار میانگین فیلتر شما می‌توانید از رجیسترهای کنترلی CR#45 و CR#46 استفاده کنید. مقدار پیش‌فرض این رجیسترها برابر با 2 می‌باشد.

4-6-8 تنظیم تصحیح وزن (کالیبره کردن وزن)

در پنجره‌ی Correction در نرم‌افزار LcSoft می‌توان مقادیر رجیسترهای کنترلی وابسته به کالیبره کردن وزن را تنظیم کرده و وزن نشان داده شده را کالیبره نماییم. پارامترهای مورد نیاز برای کالیبره کردن وزن شامل فرامین ریست کردن وزن به مقدار صفر، فرامین اعمال کردن و وارد کردن یک وزن مینا و فرامین اصلاح وزن برای کانال‌های CH1 و CH2 می‌باشند. وقتی که تمام تنظیمات را انجام دادید، برای دانلود تنظیمات اجرا شده در نرم‌افزار به ماژول DVP02LC-SL بر روی گزینه‌ی Download کلیک نمایید و یا برای خواندن مقادیر موجود در ماژول DVP02LC-SL در نرم‌افزار باید بر روی گزینه‌ی Upload کلیک نمایید.



فرمان ریست کردن وزن به صفر (Reset to zero Command): توسط رجیستر کنترلی CR#26 می‌توان این کار را انجام داد. ابتدا باید ترازو را خالی کرده و هیچ جسمی بر روی ترازو نباشد، و یا مقدار وزنی را که بر روی ترازو است را در کادر Zero weight Value Correction وارد

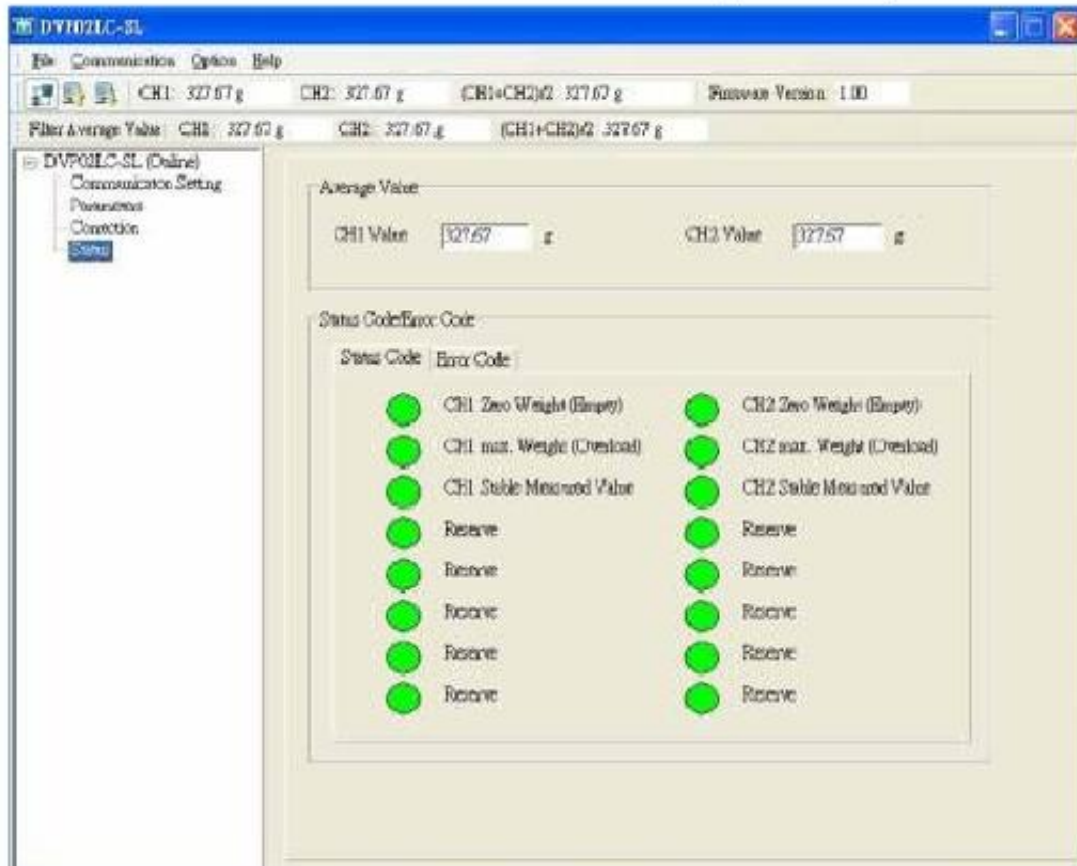
نمایید، سپس بر روی دکمه‌ی **Reset to zero Command** کلیک کنید. حال این وزن بعنوان وزن صفر در نظر گرفته می‌شود.

- فرمان وارد کردن یک وزن مبنا (**weight base point command**): توسط رجیستر کنترلی **CR#26** می‌توان این کار را انجام داد. یک جسم با یک وزن معین را بر روی ترازو قرار دهید و سپس بر روی دکمه‌ی **weight base point command** کلیک نمایید تا آن بعنوان وزن مبنا در نظر گرفته شود.




- تصحیح مقدار وزن صفر (**Zero weight Value correction**): در رجیسترهای کنترلی **CR#33** و **CR#34** می‌توان مقدار وزن صفر را تصحیح کنید. محدوده‌ی قابل تنظیم برای این رجیسترها بین $+32767 \sim -32768$ بوده و مقدار پیش‌فرض آن 1000 می‌باشد.

4-6-9 بررسی وضعیت‌های ماژول

در پنجره‌ی **Status** نرم‌افزار **LCSoft** شما می‌توانید مقدار میانگین گرفته شده‌ی وزن اندازه‌گیری شده، واحد اندازه‌گیری وزن برای کانال‌های **CH1** و **CH2**، کدهای وضعیت و کدهای خطای ماژول **DVP02LC-SL** را مشاهده نمایید.



- مقدار میانگین گرفته شده از وزن (Average value): توسط رجیسترهای کنترلی CR#12 و CR#13 می‌توان مقدار وزن میانگین گرفته شده برای کانال‌های CH1 و CH2 را مشاهده کرد. واحد اندازه‌گیری که در شکل نشان داده شده است (g) از پیش تنظیم شده است.
- کد وضعیت (Status Code): توسط مقدار موجود در رجیستر کنترلی CR#50 می‌توان از وضعیت مازول مطلع شد. این وضعیت‌ها شامل خالی بودن ترازوها، پر بودن ترازوها و پایدار بودن وزن‌های اندازه‌گیری شده در کانال‌های CH1 و CH2 می‌باشد.

Status Code	Error Code
 CH1 Zero Weight (Empty)	 CH2 Zero Weight (Empty)
 CH1 max. Weight (Overload)	 CH2 max. Weight (Overload)
 CH1 Stable Measured Value	 CH2 Stable Measured Value
 Reserve	 Reserve
 Reserve	 Reserve
 Reserve	 Reserve
 Reserve	 Reserve
 Reserve	 Reserve

















- وزن صفر (خالی بودن) کانال یک CH1 Zero weight (empty): از بیت b0 رجیستر کنترلی CR#50 برای نمایش وضعیت خالی بودن ترازو استفاده می‌شود. وقتی که مقدار اندازه‌گیری شده‌ی وزن در کانال CH1 برابر با 0 شود، این نمایشگر به رنگ قرمز روشن خواهد شد.
- حداکثر وزن (اورلود، وزن غیرمجاز) کانال یک CH1 max. weight (overload): از بیت b2 رجیستر کنترلی CR#50 برای نمایش وضعیت پر بودن ترازو استفاده می‌شود. وقتی که مقدار اندازه‌گیری شده توسط ترازوی متصل شده به کانال CH1 از حداکثر وزن تنظیم شده تجاوز کند، این نمایشگر به رنگ قرمز روشن می‌شود.
- پایدار بودن مقدار اندازه‌گیری شده در کانال یک (CH1 Stable measured value): از بیت b4 رجیستر کنترلی CR#50 برای نمایش وضعیت پایدار بودن مقدار وزن اندازه‌گیری شده استفاده می‌شود. وقتی که مقدار اندازه‌گیری شده توسط کانال CH1 پایدار و بدون پرش باشد، چراغ قرمز روشن می‌شود.

وزن صفر (خالی بودن) کانال دو (CH2 Zero weight (empty) : از بیت b1 رجیستر کنترلی CR#50 برای نمایش وضعیت خالی بودن ترازو استفاده می‌شود. وقتی که مقدار اندازه‌گیری شده‌ی وزن در کانال CH2 برابر با 0 شود، این نمایشگر به رنگ قرمز روشن خواهد شد.

حداکثر وزن (اورلود، وزن غیرمجاز) کانال دو (CH2 max. weight (overload) : از بیت b3 رجیستر کنترلی CR#50 برای نمایش وضعیت پر بودن ترازو استفاده می‌شود. وقتی که مقدار اندازه‌گیری شده توسط ترازوی متصل شده به کانال CH2 از حداکثر وزن تنظیم شده تجاوز کند، این نمایشگر به رنگ قرمز روشن می‌شود.

پایدار بودن مقدار اندازه‌گیری شده در کانال دو (CH1 Stable measured value) : از بیت b5 رجیستر کنترلی CR#50 برای نمایش وضعیت پایدار بودن مقدار وزن اندازه‌گیری شده استفاده می‌شود. وقتی که مقدار اندازه‌گیری شده توسط کانال CH2 پایدار و بدون پرش باشد، چراغ قرمز روشن می‌شود.

- کد خطا (Error Code) : کدهای خطای بوجود آمده در ماژول DVP02LC-SL در رجیستر کنترلی CR#51 ذخیره خواهند شد. کدهای خطا شامل خطای منبع تغذیه، خطای سخت‌افزار، خطاهای ولتاژ سیگنال و خطاهای تبدیل ولتاژ به وزن می‌باشد.

Status Code	Error Code
	Power Supply Abnormality
	CH1 SEN Voltage Error
	CH1 Conversion Error
	Reserve
	Reserve
	Reserve
	Reserve
	Reserve
	Hardware Abnormality
	CH2 SEN Voltage Error
	CH2 Conversion Error
	Reserve
	Reserve
	Reserve
	Reserve
	Reserve

نرمال نبودن ولتاژ منبع تغذیه (Power supply abnormality) : از بیت b0 رجیستر کنترلی CR#51 برای نمایش خطای نرمال نبودن ولتاژ منبع تغذیه استفاده می‌شود. اگر ولتاژ تغذیه‌ی ماژول نرمال نباشد، این نمایشگر قرمز خواهد شد.

خطای سخت‌افزار (Hardware abnormality): از بیت b1 رجیستر کنترلی CR#51 برای نمایش خطای سخت‌افزار استفاده می‌شود. اگر خطایی در سخت‌افزار رخ دهد، این نمایشگر به رنگ قرمز روشن می‌شود.

خطای ولتاژ سیگنال کانال یک (CH1 SEN voltage error): از بیت b3 رجیستر کنترلی CR#51 برای نمایش خطای ولتاژ سیگنال کانال CH1 استفاده می‌شود. اگر خطایی در mV سنس شده توسط ماژول به وجود آید به آن معنی است که لودسل بد عمل کرده و نرمال نمی‌باشد. در این حالت، این نمایشگر به رنگ قرمز روشن می‌شود.

خطای ولتاژ سیگنال کانال دو (CH2 SEN voltage error): از بیت b5 رجیستر کنترلی CR#51 برای نمایش خطای ولتاژ سیگنال کانال CH2 استفاده می‌شود. اگر خطایی در mV سنس شده توسط ماژول به وجود آید، به آن معنی است که لودسل بد عمل کرده و نرمال نمی‌باشد. در این حالت، این نمایشگر به رنگ قرمز می‌شود.

خطای تبدیل ولتاژ به وزن در کانال یک (CH1 Conversion error): از بیت b4 رجیستر کنترلی CR#51 برای نمایش خطای تبدیل mV به وزن در کانال CH1 استفاده شده است. اگر خطایی در هنگام تبدیل سیگنال اندازه‌گیری شده به وجود آید، این نمایشگر قرمز می‌شود.

خطای تبدیل ولتاژ به وزن در کانال دو (CH2 Conversion error): از بیت b6 رجیستر کنترلی CR#52 برای نمایش خطای تبدیل mV به وزن در کانال CH2 استفاده شده است. اگر خطایی در هنگام تبدیل سیگنال اندازه‌گیری شده به وجود آید، این نمایشگر قرمز می‌شود.

4-6-10 مراحل تصحیح (کالیبره کردن) وزن

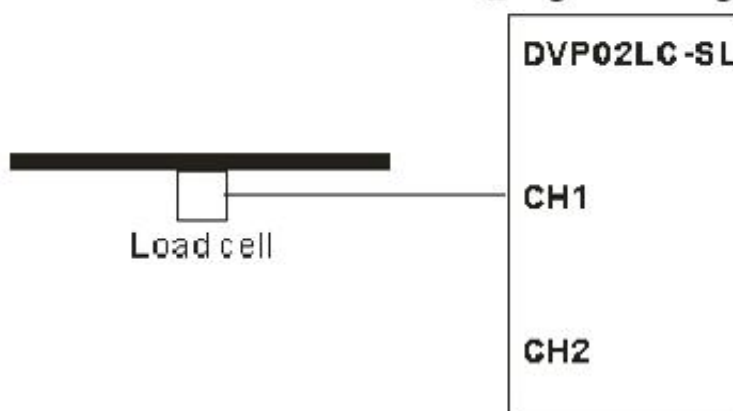
از گزینه‌ی Correction برای تطبیق دادن (همانگ کردن) ماژول DVP02LC-SL با وزن اعمال شده به لودسل استفاده می‌شود. عمل تصحیح وزن را می‌توان توسط کامپیوتر در نرم‌افزار LCSOFT و یا توسط CPU های دلتا انجام داد. برای تصحیح وزن توسط PLC های دلتا، ابتدا باید ماژول DVP02LC-SL را به MPU (CPU) سری DVP دلتا متصل نمایید و سپس توسط دستورالعمل‌های TO/FROM عمل تصحیح وزن را اجرا نمایید. تصحیح وزن توسط نرم‌افزار به صورت دستی صورت می‌گیرد. برای این کار ابتدا باید ماژول DVP02LC-SL را با استفاده از یک کابل ارتباطات RS-232 به کامپیوتر متصل نموده و سپس وارد پنجره‌ی Correction در نرم‌افزار LCSOFT شوید.



4-6-11 تصحیح وزن توسط PLC

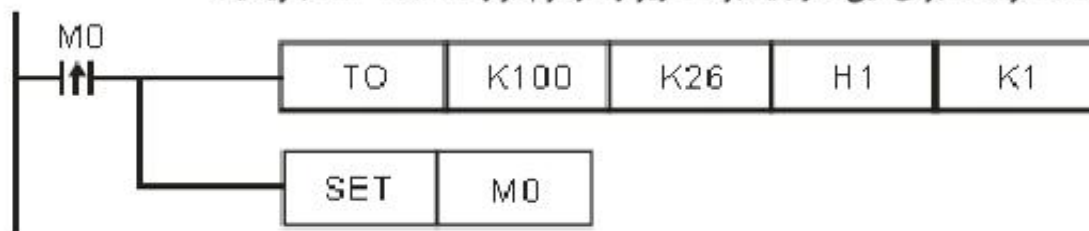
در این مثال، ما ماژول DVP02LC-SL را به PLC MPU سری DVP متصل کرده‌ایم و توسط دستورالعمل TO وزن کانال CH1 را تصحیح می‌کنیم.

1. ابتدا باید ماژول DVP02LC-SL را به سمت چپ CPU دلتا متصل نموده و سپس ماژول لودسل و تغذیه CPU را متصل کنید.
2. لودسل را به کانال CH1 متصل نمایید.

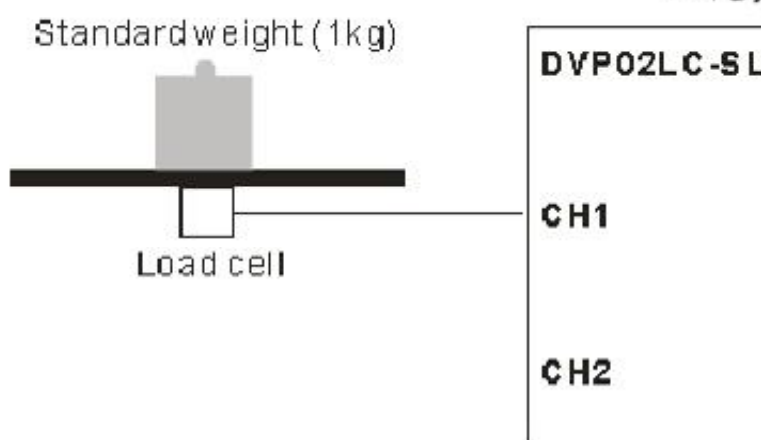


3. مقادیر هر یک از پارامترهای ماژول را باید تنظیم نمایید. مقدار مشخصه‌ی لودسل (eigenvalue) را باید مطابق با مشخصات لودسلی که به ماژول متصل شده است، تنظیم نمایید. در این مثال، ما از تنظیمات پیش‌فرض پارامترهای ماژول استفاده می‌کنیم.

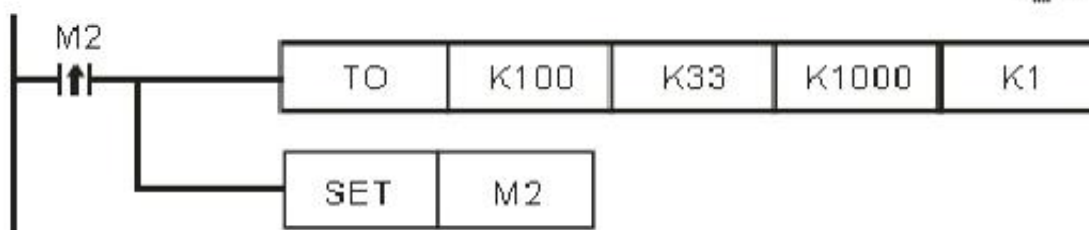
4. برای اجرای فرمان ریست وزن به صفر باید مقدار H'0001 هگز را در داخل رجیستر کنترلی CR#29 نوشت. برای این کار باید برنامه زیر را در نرم‌افزار WPL Soft بنویسید.



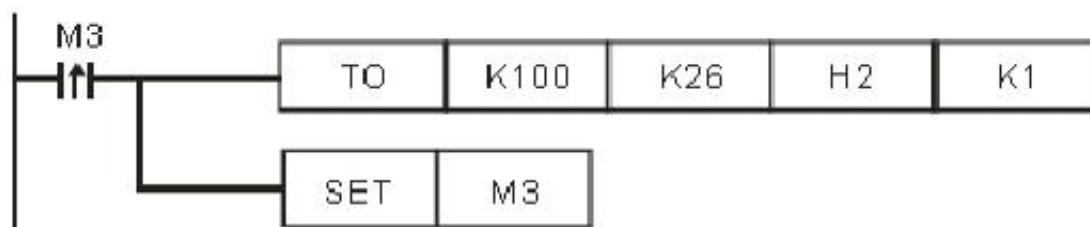
5. یک وزنه استاندارد 1 کیلوگرمی را بر روی ترازوی لودسل قرار دهید. شما باید از حداکثر وزن قابل اندازه‌گیری توسط لودسل متصل شده به ماژول مطلع باشید. این وزنه‌ی استاندارد 1 کیلوگرمی باید توسط لودسل قابل اندازه‌گیری باشد.



6. این وزن را باید بعنوان وزن مبنا در داخل رجیستر کنترلی CR#33 بنویسید. برای نوشتن این وزن بعنوان وزن مبنا در PLC باید برنامه زیر را در نرم‌افزار WPL Soft نوشته و به برنامه بند 4 اضافه نمایید.



7. برای اجرای فرمان وزن مبنا باید مقدار هگز H'0002 را در داخل رجیستر کنترلی CR#26 مانند شکل زیر به برنامه نوشته شده در نرم‌افزار WPL Soft اضافه نمایید.

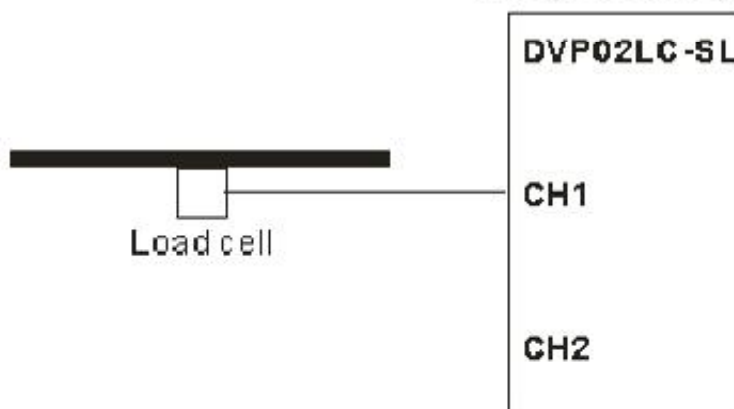


8. درنهایت، برنامه را به PLC دانلود نمایید تا CPU تنظیمات پارامترها را به ماژول DVP02LC-SL ارسال کرده و تنظیمات انجام شود.

12-6-4 تصحیح وزن در نرم‌افزار LCSOft

در این مثال، می‌خواهیم شرح دهیم که چگونه می‌توان وزن کانل CH1 ماژول DVP02LC-SL را توسط نرم‌افزار LCSOft به صورت دستی اصلاح کرد.

1. ابتدا باید کابل واسط RS-232 ماژول را به کامپیوتر متصل نمایید. سپس باید تغذیه PLC و ماژول DVP02LC-SL را متصل کنید.
2. لودسل را به کانال CH1 متصل کنید.



3. نرم‌افزار LCSOft را اجرا کنید. برای برقرار کردن ارتباط بین نرم‌افزار و ماژول DVP02LC-SL باید تنظیمات پورت ارتباطات RS-232 را مطابق با مطالب توضیح داده شده در بخش‌های قبل اجرا نمایید.

4. در پنجره‌ی درختی سمت چپ نرم‌افزار بر روی گزینه‌ی Parameters جهت شروع اجرا تنظیمات کلیک نمایید. تمام پارامترها و مشخصه‌ی ویژه (eigenvalue) لودسل را مطابق با مشخصات لودسل متصل شده به ماژول تنظیم نمایید. بعد از اینکه تنظیمات تمام پارامترها اجرا شد، جهت ارسال تنظیمات اجرا شده به ماژول DVP02LC-SL باید بر روی گزینه‌ی Download کلیک کنید.

CH1 Parameters

Gross/Net	Gross	Gross Weight	0	Read GW
Average Times	10	Maximum Weight	32767	
Decimal Place	2	Unit of Measurement	g	
Standstill Times	5	Standstill Range	10	
Zero Band	-10 ~ 10			
Filter Ratio Setting	2			

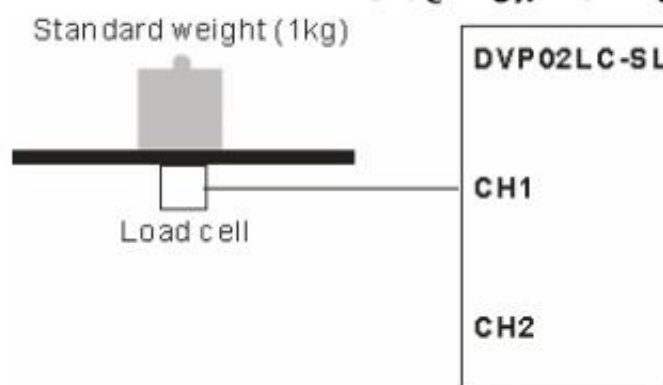
5. برای شروع تصحیح وزن باید در پنجره‌ی درختی سمت چپ نرم‌افزار بر روی گزینه‌ی **Correction** کلیک نمایید. وقتی که هیچ جسمی (باری) بر روی ترازوی لودسل وجود ندارد، باید بر روی گزینه‌ی **Reset to Zero Command** کلیک نمایید.

CH1 Correction

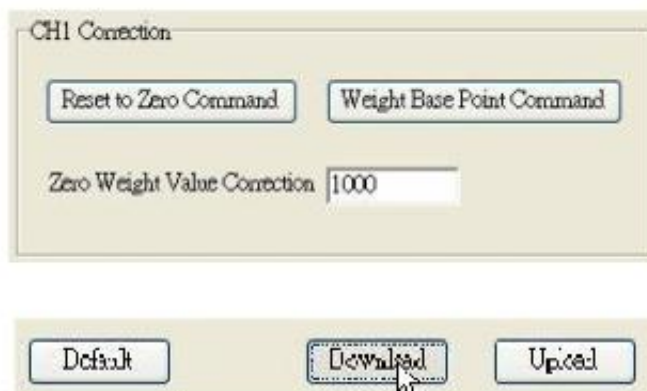
Reset to Zero Command Weight Base Point Command

Zero Weight Value Correction 0

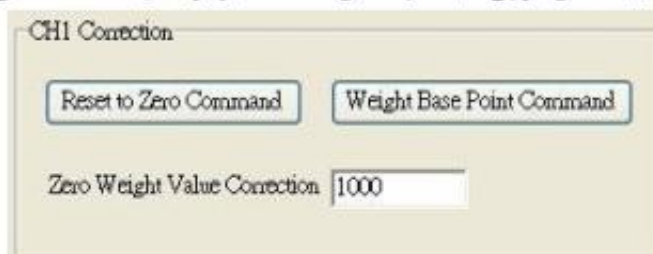
6. یک وزنه‌ی استاندارد 1kg را بر روی ترازوی لودسل قرار دهید. شما حتماً باید از حداکثر وزن قابل اندازه‌گیری لودسل متصل شده به ماژول مطلع باشید.



7. مقدار 1000 را در کادر **Zero Weight Value Correction** وارد نموده و برای دانلود تنظیمات به ماژول DVP02LC-SL بر روی دکمه‌ی **Download** کلیک نمایید.

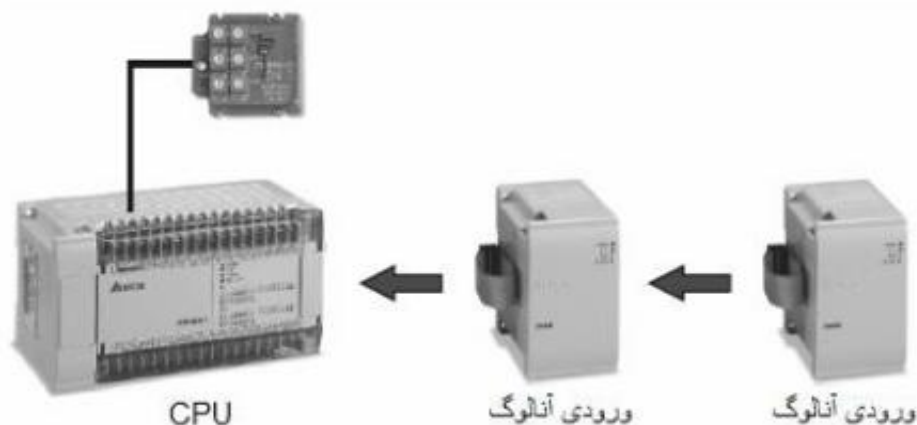


8. برای تنظیم این وزن بعنوان وزن مبنا باید بر روی دکمه‌ی **Weight Base Point Command** کلیک نمایید. حال، وزن مبنا در کانال CH1 برابر با 1000kg می‌باشد.



4- 7 کنترل ولتاژ خروجی 2 ماژول DVP-04DA توسط 8 VRs (Variable Resistors)

در این مثال می‌خواهیم برنامه کنترل ولتاژ خروجی را در دو ورودی آنالوگ از جنس ولتاژ توسط 8 رجیستر متغیر بررسی نماییم.

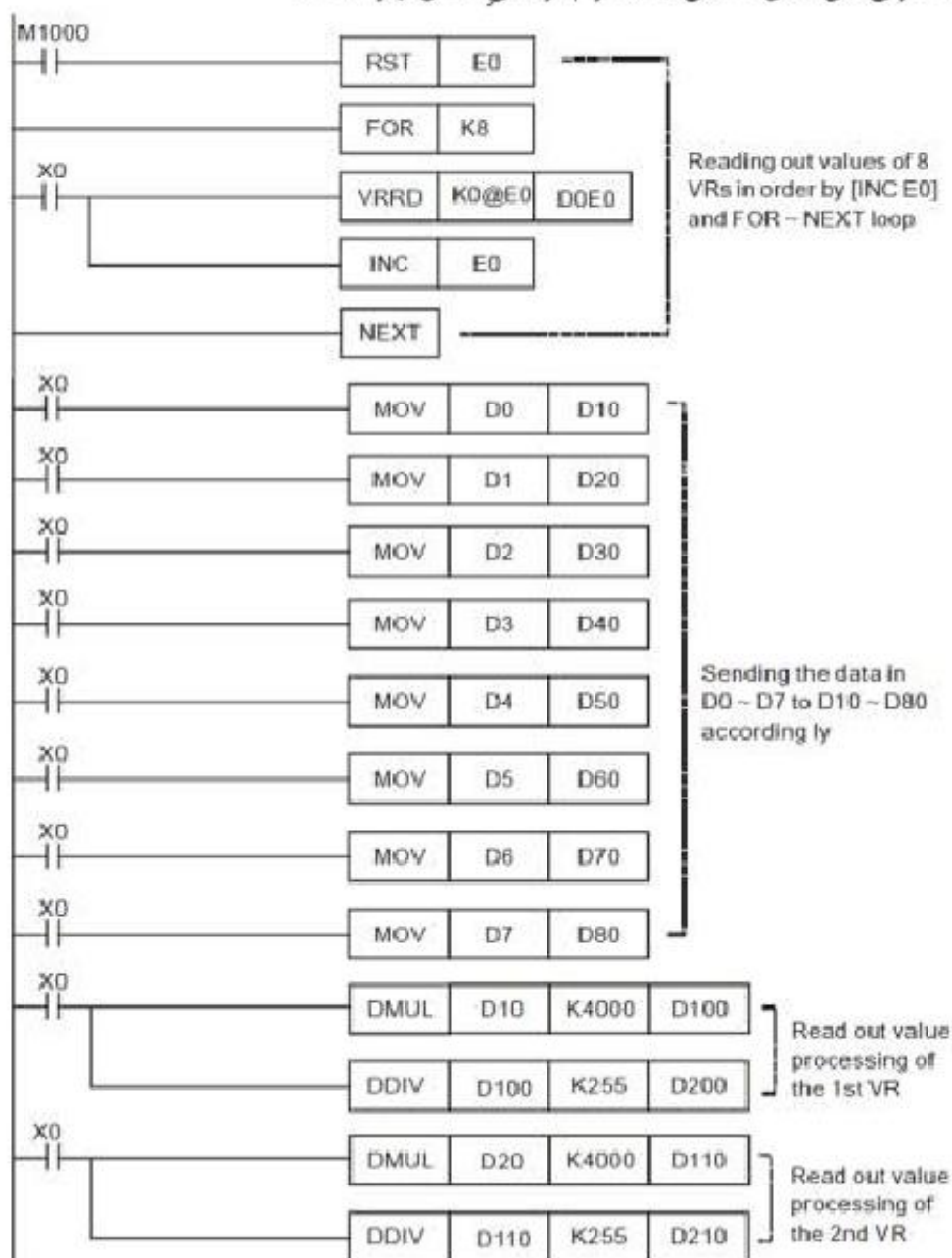


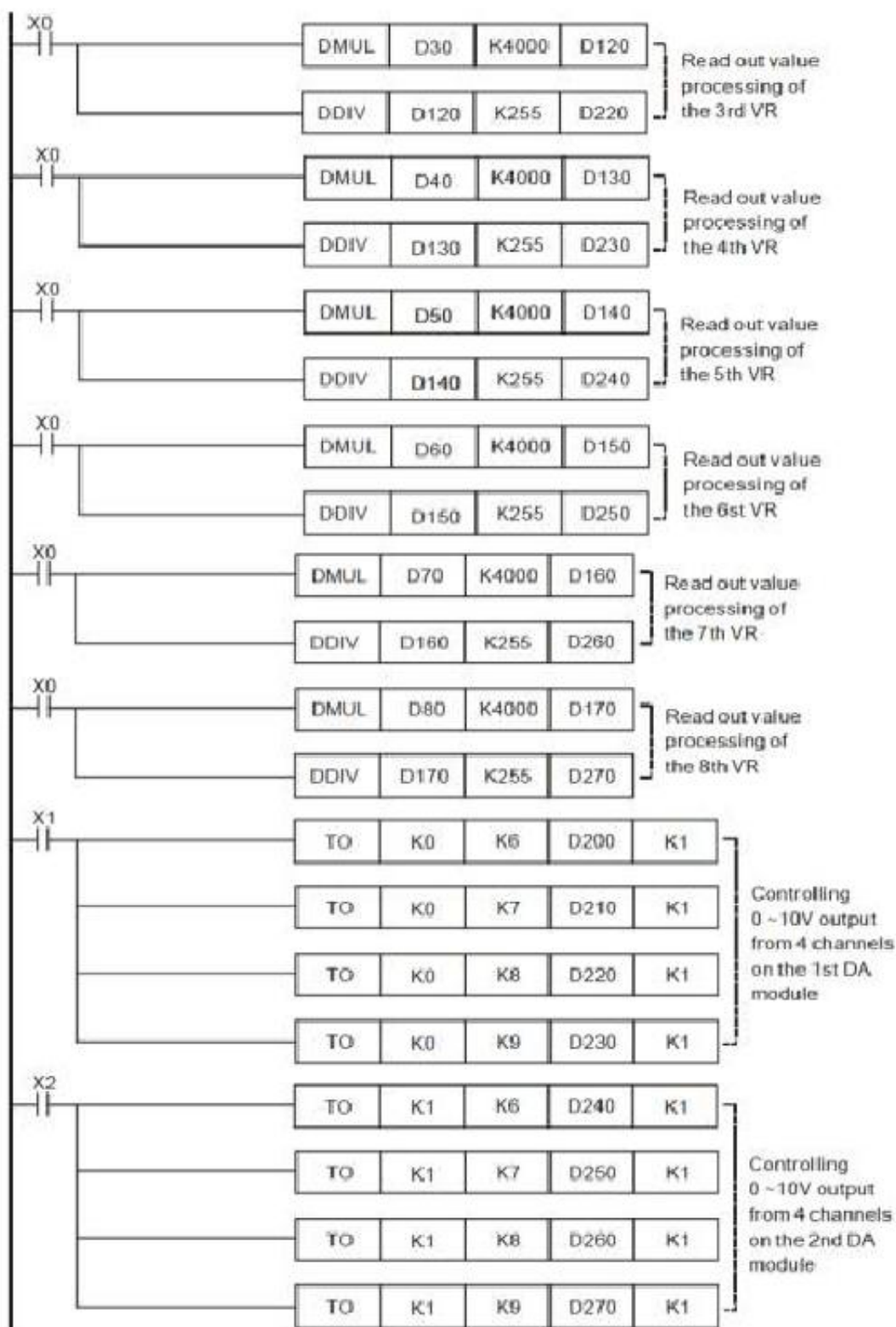
هدف از کنترل در این مثال به شرح زیر می‌باشد:
می‌خواهیم مقدار ولتاژ خروجی آنالوگ (0~10V) در 8 کانال خروجی آنالوگ را کنترل نموده و مقدار ولتاژ هر کانال را از 0 تا 10 ولت توسط 8 رجیستر متغیر بر روی PLC تغییر دهیم.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند :

ابزار	عملکرد ابزار
X0	به این ورودی از PLC، سوئیچ START برای خواندن مقادیر رجیسترهای متغیر متصل می‌شود.
X1	به این ورودی از PLC، شستی نوشتن مقدار عددی در نخستین مازول خروجی آنالوگ متصل می‌شود.
X2	به این ورودی از PLC، شستی نوشتن مقدار عددی در دومین مازول خروجی آنالوگ متصل می‌شود.
E0	رجیستر شاخص

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر است :





عملکرد برنامه‌ی کنترلی این مثال به شرح زیر می‌باشد:

در این برنامه از رجیسترهای شاخص E0 و حلقه‌ی FOR~NEXT برای تعیین کردن شماره رجیستر متغیر (VR) و شماره‌ی رجیسترهای D برای ذخیره‌سازی مقدار و خواندن مقدار از رجیستر متغیر (VR) استفاده شده است.

در حلقه‌ی FOR~NEXT، با اجرای دستورالعمل [INC E0] مقدار E0 از 0 تا 7 تغییر خواهد کرد. بنابراین مقدار K0@E0 از K0 تا K7 تغییر نموده و رجیستر D0E0 از D0 تا D7 تغییر می‌کند. از این رو، مقادیر 8 رجیستر متغیر (VRs) از VR0→D0, VR1→D1, ..., VR7→D7 خوانده خواهد شد.

محدوده‌ی مقدار رجیسترهای متغیر (VR) از K0 تا K255 و محدوده‌ی ولتاژ خروجی ماژول‌های آنالوگ از 0 تا 10 ولت مطابق با K0~K4000 است. بنابراین، در برنامه‌ی نوشته شده، مقدار رجیستر متغیر K0~K255 در برنامه به مقدار K0~K4000 در ماژول آنالوگ تبدیل خواهد شد. همانطور که متوجه شدید، در این فرآیند، هدف، کنترل کردن ولتاژ 0 تا 10 ولت خروجی توسط تنظیمات مقدار رجیسترهای متغیر (VR) می‌باشد که می‌توان از طریق رجیسترهای D به این تنظیمات دست یافت. مقدار تبدیل شده‌ی K0~K4000 به رجیسترهای D200, D210, D220, ..., D270 ارسال خواهد شد و توسط دستورالعمل TO مقدار ولتاژ خروجی آنالوگ به کانال خروجی آنالوگ انتخاب شده (تنظیم شده) منتقل می‌شود.

دقت داشته باشید که این برنامه را می‌توان توسط دستورالعمل VRRD و دستورالعمل TO به سادگی نوشت.

4-8 کنترل سطح آب

در این مثال، می‌خواهیم برنامه‌ی کنترل آلام سطح آب را با استفاده از دستورالعمل ZCP بررسی نماییم.

هدف از کنترل عبارتست از:

می‌خواهیم سطح آب داخل یک برج خنک کن را توسط یک سنسور اندازه‌گیری سطح آنالوگ (سنسور آلتراسونیک) اندازه‌گیری نماییم. وقتی که آب در سطح نرمال بود، نمایشگر سطح نرمال روشن می‌شود. وقتی که در برج خنک کن فقط 25% آب وجود داشت، شیر برقی پر کن آب داخل برج روشن شود. وقتی که سطح آب به بالاترین محدوده رسید، آلام فعال شده و شیر برقی پرکن آب برج خنک کن خاموش می‌شود.

ابزارهای مورد استفاده در این مثال در جدول زیر لیست شده‌اند:

ابزار	عملکرد ابزار
Y0	به این خروجی شیر برقی پر کن آب برج خنک کن متصل می‌شود. (مقدار پایین‌ترین محدوده برابر با K1000 می‌باشد).
Y1	به این خروجی چراغ سیگنال نرمال بودن سطح آب داخل برج خنک کن متصل می‌شود.
Y2	به این خروجی چراغ سیگنال آلارم بالاترین محدوده سطح آب داخل برج خنک کن وصل می‌شود. (مقدار بالاترین محدوده‌ی آب برابر با K4000 است).
D0	در این رجیستر مقدار سطح آب اندازه‌گیری شده در داخل مخزن توسط سنسور آنالوگ که به ورودی آنالوگ PLC (مقدار K0~K4000) متصل شده است، ذخیره می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:

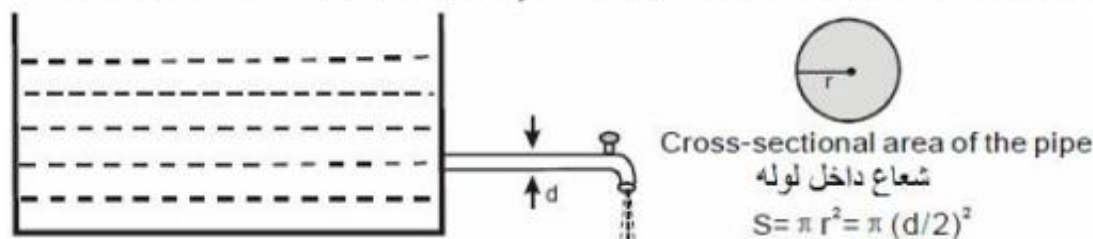


عملکرد برنامه‌ی کنترلی این مثال به شرح زیر است:

سطح آب توسط سنسور آنالوگ اندازه‌گیری سطح مایعات، اندازه‌گیری می‌شود. خروجی این سنسور از نوع ولتاژ و در رنج 0~10V می‌باشد. برای خواندن مقدار سطح آب توسط سنسور باید از مازول توسعه آنالوگ استفاده کرد. این مازول مقدار آنالوگ 0~10V را به مقدار دیجیتال K0~K4000 تبدیل کرده و می‌توان مقدار سطح آب را توسط مقدار ذخیره شده در ابزار D0 خوانده و تشخیص داد. وقتی که مقدار $D0 < K1000$ (مقدار سطح آب داخل برج 25%) شود، فلگ عمومی M1=ON شده و شیر برقی متصل شده به خروجی Y0 فعال می‌شود. تا زمانی که سطح آب به 100% برسد. وقتی که مقدار اندازه‌گیری شده‌ی آب داخل برج بیشتر از K4000 شود، ابزار M2=ON شده و آلارم بالاترین سطح محدوده‌ی آب (Y2) روشن می‌شود. در این هنگام، خروجی Y0 ریست شده و شیر برقی پرکردن آب داخل برج غیرفعال می‌شود. برای ساده‌تر کردن برنامه می‌توان از دستورالعمل FROM استفاده کرد.

4-9 اندازه‌گیری جریان آب داخل لوله

در این مثال، می‌خواهیم دبی (جریان) دقیق آب جاری شده در داخل لوله را اندازه‌گیری نماییم.



هدف از کنترل در این مثال به شرح زیر است:

هدف از کنترل، اندازه‌گیری دقیق جریان داخل لوله می‌باشد. در این مثال، قطر لوله برحسب mm (میلی‌متر) اندازه‌گیری شده است، میزان سرعت جریان جاری در لوله برحسب dm/s اندازه‌گیری شده است و جریان جاری در لوله برحسب cm³/s اندازه‌گیری می‌شود. سطح مقطع عرضی لوله با معادله

$$S = \pi r^2 = \pi \left(\frac{d}{2} \right)^2$$

محاسبه می‌شود و جریان جاری در لوله با معادله (سرعت جریان

جاری در لوله × سطح مقطع عرضی لوله = جریان جاری در لوله) محاسبه می‌شود.

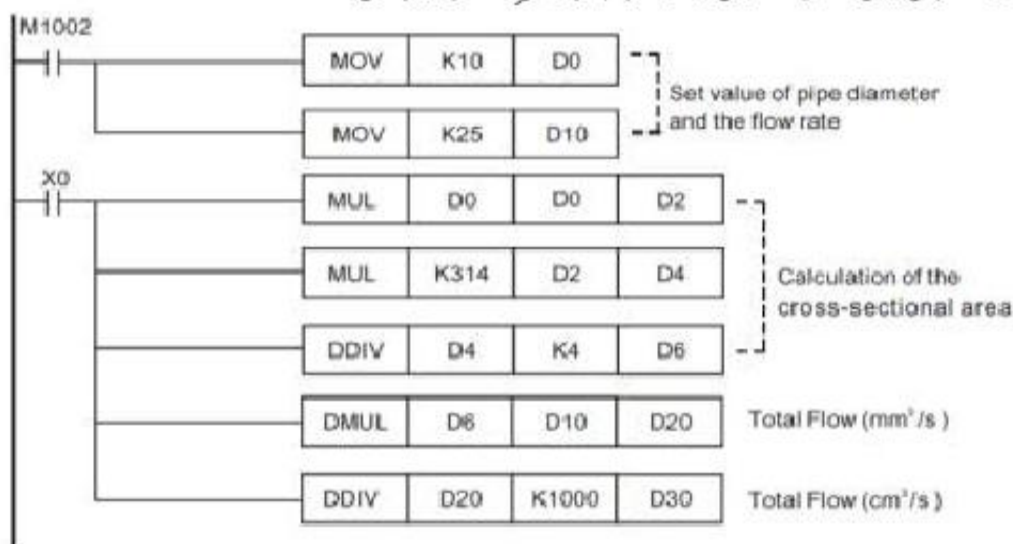
$$\text{Cross-sectional area of the pipe} = \pi r^2 = \pi (d/2)^2$$

$$\text{Flow} = \text{cross-sectional area} \times \text{flow rate}$$

ابزارهای مورد استفاده در این مثال به شرح زیر است:

ابزار	توضیحات
X0	وقتی که این ورودی فعال شود، برنامه شروع به اندازه‌گیری می‌نماید.
D0	در این رجیستر قطر لوله نوشته می‌شود. (واحد قطر لوله برحسب mm و مقدار تنظیمی در این برنامه برابر با 10mm است).
D6	در این رجیستر نتیجه‌ی محاسبه‌ی شده‌ی قطر عرضی لوله ذخیره می‌شود (واحد قطر عرضی لوله برحسب mm² است).
D10	سرعت جریان جاری در لوله در این رجیستر نوشته می‌شود. (واحد سرعت جریان جاری در لوله برحسب dm/s بوده و مقدار تنظیمی در این برنامه برابر با 25 dm/s می‌باشد).
D20	در این رجیستر نتیجه‌ی محاسبه‌ی جریان جاری در لوله ذخیره می‌شود (واحد جریان جاری در لوله برحسب mm³/s است).
D30	در این رجیستر نتیجه‌ی محاسبه‌ی جریان جاری در لوله ذخیره می‌شود (واحد جریان جاری در لوله برحسب cm³/s می‌باشد).

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر است:

از نقطه‌ی شناور **floating point** یا همان اعداد اعشاری معمولاً برای اجرای محاسبات اعداد دسیمال (اعشاری) استفاده می‌شود. برای ساده کردن محاسبات باید اعداد اعشاری را تبدیل کنیم تا از پیچیدگی آن کاسته شود. ما در این مثال از دستورالعمل‌های مقدماتی ریاضی جهت محاسبه‌ی اعداد دسیمال (صحیح) استفاده کرده‌ایم. واحدهای مورد استفاده در این مثال mm , cm , dm می‌باشد. برای محاسبه لازم است که در برنامه واحد را برحسب mm^2 تنظیم نموده و سپس آن را به cm^3 تبدیل کنیم.

مقدار π تقریباً $3/14$ می‌باشد. از این رو، ما مقدار π را برای محاسبه‌ی ناحیه قطر عرضی لوله نیاز داریم. به منظور محاسبه‌ی دقیق با دو اعشار (اعشار صدم) باید مقدار π را در 100 ضرب کرده و مقدار $K314$ را جایگزین مقدار π نماییم و واحد dm/s را نیز در 100 ضرب کرده و این واحد را به واحد mm/s تبدیل نماییم.

در پایان، برنامه را تقسیم بر مقدار موجود در رجیستر $d20$ که واحد آن mm^3/s است کرده‌ایم. سپس این مقدار را در 1000 ضرب می‌کنیم تا واحد به cm^3/s تبدیل شود.

$$(1 \text{ cm}^3 = 1 \text{ ml}, 1 \text{ l} = 1000 \text{ ml} = 1000 \text{ cm}^3 = 1 \text{ dm}^3)$$

فرض کنید که قطر لوله، که در رجیستر $D0$ ذخیره شده برابر با $10mm$ و سرعت جریان جاری در لوله که در رجیستر $D10$ ذخیره شده برابر با $25dm/s$ است. در این حالت، نتیجه‌ی محاسبه شده مجموع جریان جاری برابر با $196 \text{ cm}^3/s$ خواهد بود.

4-10 ذخیره کردن مقدار آنالوگ بدست آمده

در این مثال، می‌خواهیم با استفاده از دستورالعمل SORT مقادیر اندازه‌گیری شده توسط ماژول‌های آنالوگ DVP04AD-S و DVP04TC-S را بدست آوریم و مقادیر بدست آمده از ورودی‌های آنالوگ را از کم به زیاد به صورت صعودی طبقه‌بندی (دسته‌بندی) نماییم.

هدف از کنترل در این مثال به شرح زیر می‌باشد:

هدف از کنترل، گردآوری 4 داده ولتاژی (مطابق با فرکانس اینورتر) توسط ماژول ورودی آنالوگ DVP04AD-S و چهار داده‌ی دمای توسط ماژول ورودی ترموکوپل DVP04TC-S می‌باشد.

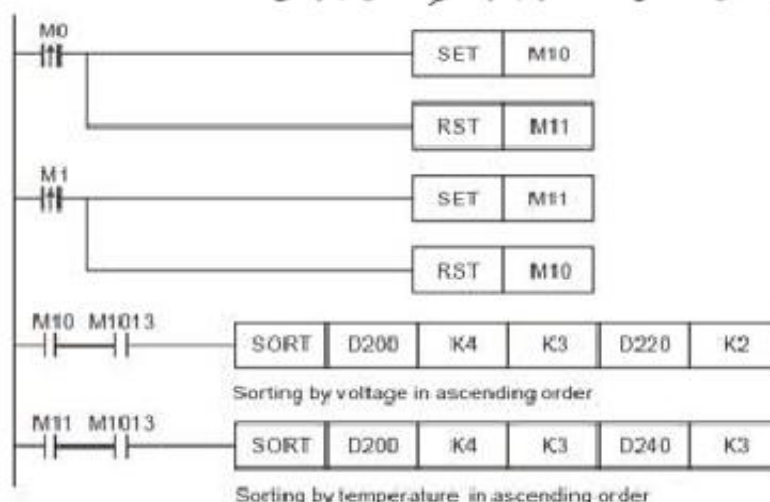
وقتی که ورودی $M0=ON$ شود، مقادیر ولتاژی 4 کانال ورودی آنالوگ به صورت صعودی از کم به زیاد دسته‌بندی (طبقه‌بندی) می‌شوند. همچنین، وقتی که $M1=ON$ شود، مقادیر دمای 4 کانال ورودی آنالوگ اندازه‌گیری دما به صورت صعودی از کم به زیاد دسته‌بندی (طبقه‌بندی) می‌شوند.

هدف نهایی طبقه‌بندی (دسته‌بندی) داده‌های ورودی از ماژول آنالوگ و نمایش نتیجه‌ی بدست آمده از این طبقه‌بندی می‌باشد.

ابزارهای مورد استفاده در این مثال به شرح زیر است:

ابزار	توضیحات
M0	طبقه‌بندی (دسته‌بندی) کردن داده‌های ولتاژ ورودی از ورودی آنالوگ از کم به زیاد
M1	طبقه‌بندی (دسته‌بندی) کردن داده‌های دمای ورودی از ورودی ترموکوپل از کم به زیاد
D200~D203	تعداد کانال‌ها برای طبقه‌بندی (دسته‌بندی) کردن
D204~D207	طبقه‌بندی کردن 4 داده‌ی ورودی از جنس ولتاژ
D208~D211	طبقه‌بندی کردن 4 داده‌ی ورودی از جنس دما
D220~D231	نمایش نتیجه‌ی ولتاژ طبقه‌بندی شده
D240~D251	نمایش نتیجه‌ی دمای طبقه‌بندی شده

برنامه کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر است:

داده‌های بدست آورده شده قبل از طبقه‌بندی (دسته‌بندی) مطابق با جدول زیر می‌باشد:

	1	2	3
	Channel (CH1~CH4)	Voltage (DVP04AD-S)	Temp. (DVP04TC-S)
1	(D200)1	(D204)57	(D208)47
2	(D201)2	(D205)59	(D209)42
3	(D202)3	(D206)55	(D210)46
4	(D203)4	(D207)53	(D211)43

وقتی که فلگ M0=ON شود، مقادیر ولتاژ ورودی از کم به زیاد دسته‌بندی می‌شوند.

	1	2	3
	Channel (CH1~CH4)	Voltage (DVP04AD-S)	Temp. (DVP04TC-S)
1	(D220)4	(D224)53	(D228)43
2	(D221)3	(D225)55	(D229)46
3	(D222)1	(D226)57	(D230)47
4	(D223)2	(D227)59	(D231)42

نتیجه ولتاژ طبقه‌بندی شده از کم به زیاد به صورت CH4, CH3, CH1, CH2 می‌باشد. حداقل

مقدار K53 بوده و حداکثر مقدار K59 می‌باشد.

وقتی که فلگ M1=ON شود، مقادیر دما از کم به زیاد طبقه‌بندی می‌شوند.

	1	2	3
	Channel (CH1~CH4)	Voltage (DVP04AD-S)	Temp. (DVP04TC-S)
1	(D240)4	(D244)59	(D248)42
2	(D241)1	(D245)53	(D249)43
3	(D242)2	(D246)55	(D250)46
4	(D243)3	(D247)57	(D251)47

نتیجه‌ی بدست آمده از طبقه‌بندی مقدار دما از کم به زیاد به صورت CH4, CH1, CH2, CH3 می‌باشد. حداقل مقدار دما برابر با K42 بوده و حداکثر مقدار دما برابر با K47 می‌باشد. هدف از استفاده از فلگ خاص M1013 (کلاک پالس 1 ثانیه) بعد از راه‌اندازی کنتاکت‌های M11, M10, کسب اطمینان از طبقه‌بندی مقادیر با استفاده از نتیجه‌ی بدست آمده در هر یک ثانیه می‌باشد. از آنجایی که برای به روزرسانی مقادیر موجود در دستورالعمل SORT، این دستورالعمل باید اجرای مکرر داشته باشد، برای بروز رسانی مقادیر یا باید از لبه‌ی بالارونده‌ی یک کنتاکت در ورودی دستورالعمل استفاده کرد و یا باید از یک کلاک پالس استفاده کرد. در اینجا، برای به روزرسانی مداوم مقادیر برای دستورالعمل SORT از کلاک پالس استفاده شده است. شما می‌توانید نتیجه بدست آمده از طبقه‌بندی (دسته‌بندی) مقادیر از کم به زیاد ولتاژ و دما را مانیتور نمایید.

4- 11- مانیتور کردن دمای اتاق

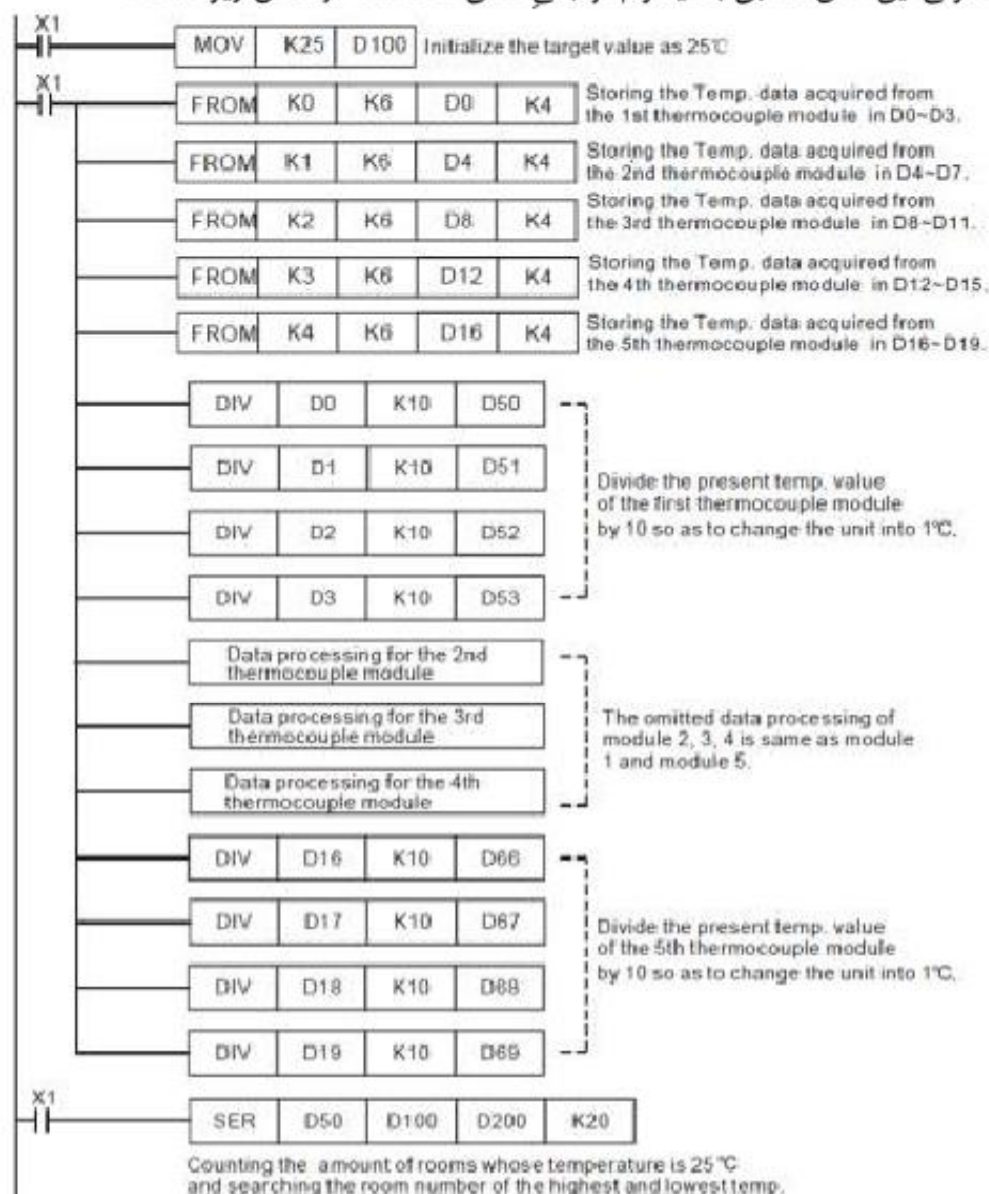
در این مثال، می‌خواهیم با استفاده از دستورالعمل SER دمای اتاق را مانیتور نماییم. هدف از کنترل در این مثال به شرح زیر است:

می‌خواهیم تمام شرایط دمایی را توسط داده‌های دمایی بدست آورده شده در شرایط هوایی مختلف در 20 اتاق مستقل از یکدیگر را مانیتور نماییم. سپس مقدار دمای واقعی را با دمای هدف مقایسه نماییم. همچنین، می‌خواهیم با تجهیزات کنترل دما شرایط هوای داخل اتاق را برای کمترین و بیشترین دما تنظیم نماییم.

ابزارهای مورد استفاده در این مثال به شرح زیر می‌باشد:

ابزار	توضیحات
X1	با فعال شدن این ورودی، دستورالعمل SER برای جستجوی دیتا اجرا خواهد شد.
D50~D53	مقدار دمای اندازه‌گیری شده در نخستین مازول ترموکوپل (این مازول شامل 4 کانال ورودی ترموکوپل می‌باشد) در این رجیسترها ذخیره می‌شوند. واحد دما برحسب 1°C می‌باشد.
D54~D57	مقدار دمای اندازه‌گیری شده در دومین مازول ترموکوپل (این مازول شامل 4 کانال ورودی ترموکوپل می‌باشد) در این رجیسترها ذخیره می‌شوند. واحد دما برحسب 1°C می‌باشد.
D58~D61	مقدار دمای اندازه‌گیری شده در سومین مازول ترموکوپل (این مازول شامل 4 کانال ورودی ترموکوپل می‌باشد) در این رجیسترها ذخیره می‌شوند. واحد دما برحسب 1°C می‌باشد.
D62~D65	مقدار دمای اندازه‌گیری شده در چهارمین مازول ترموکوپل (این مازول شامل 4 کانال ورودی ترموکوپل می‌باشد) در این رجیسترها ذخیره می‌شوند. واحد دما برحسب 1°C می‌باشد.
D66~D69	مقدار دمای اندازه‌گیری شده در پنجمین مازول ترموکوپل (این مازول شامل 4 کانال ورودی ترموکوپل می‌باشد) در این رجیسترها ذخیره می‌شوند. واحد دما برحسب 1°C می‌باشد.
D100	در این رجیستر مقدار دمای هدف (SP) تنظیم می‌شود.
D200~D204	در این رجیسترها نتیجه دمای جستجو شده تنظیم می‌شود.

برنامه کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر است :



عملکرد این برنامه به شرح زیر می‌باشد :

مقدار دمای بدست آورده شده و نتیجه‌ی جستجو شده در 20 اتاق در جدول زیر آورده شده است :

شماره‌ی اتاق	دمای مطلوب (هدف)	شماره	مقایسه‌ی نتیجه‌ی بدست آمده با دمای هدف
D50 = K24		0	-
D51 = K25		1	دمای واقعی برابر با دمای مطلوب می‌باشد
D52 = K25		2	دمای واقعی برابر با دمای مطلوب می‌باشد
D53 = K25		3	دمای واقعی برابر با دمای مطلوب می‌باشد
D54 = K25		4	دمای واقعی برابر با دمای مطلوب می‌باشد

دمای واقعی کمتر از دمای مطلوب می‌باشد	5	D100 = K25	D55 = K22
دمای واقعی برابر با دمای مطلوب می‌باشد	6		D56 = K25
دمای واقعی برابر با دمای مطلوب می‌باشد	7		D57 = K25
دمای واقعی برابر با دمای مطلوب می‌باشد	8		D58 = K25
دمای واقعی برابر با دمای مطلوب می‌باشد	9		D59 = K25
دمای واقعی برابر با دمای مطلوب می‌باشد	10		D60 = K25
دمای واقعی بیشتر از دمای مطلوب می‌باشد	11		D61 = K27
دمای واقعی برابر با دمای مطلوب می‌باشد	12		D62 = K25
دمای واقعی برابر با دمای مطلوب می‌باشد	13		D63 = K25
-	14		D64 = K26
دمای واقعی برابر با دمای مطلوب می‌باشد	15		D65 = K25
دمای واقعی برابر با دمای مطلوب می‌باشد	16		D66 = K25
دمای واقعی برابر با دمای مطلوب می‌باشد	17		D67 = K25
دمای واقعی برابر با دمای مطلوب می‌باشد	18		D68 = K25
دمای واقعی برابر با دمای مطلوب می‌باشد	19		D69 = K25

مقدار	نتیجه‌ی جستجو شده
مقدار دمای اتاق‌ها برابر با 25C می‌باشد	D200 = K16
شماره‌ی نخستین اتاقی برای اندازه‌گیری دمای آن	D201 = K1
شماره‌ی آخرین اتاقی برای اندازه‌گیری دمای آن	D202 = K19
شماره اتاقی که کمترین دما را دارد	D203 = K5
شماره اتاقی که بیشترین دما را دارد	D204 = K11

4- 12 کنترل شیب حرکت جرثقیل

در این مثال، می‌خواهیم با استفاده از دستورالعمل RAMP شیب حرکت جرثقیل را کنترل کنیم.



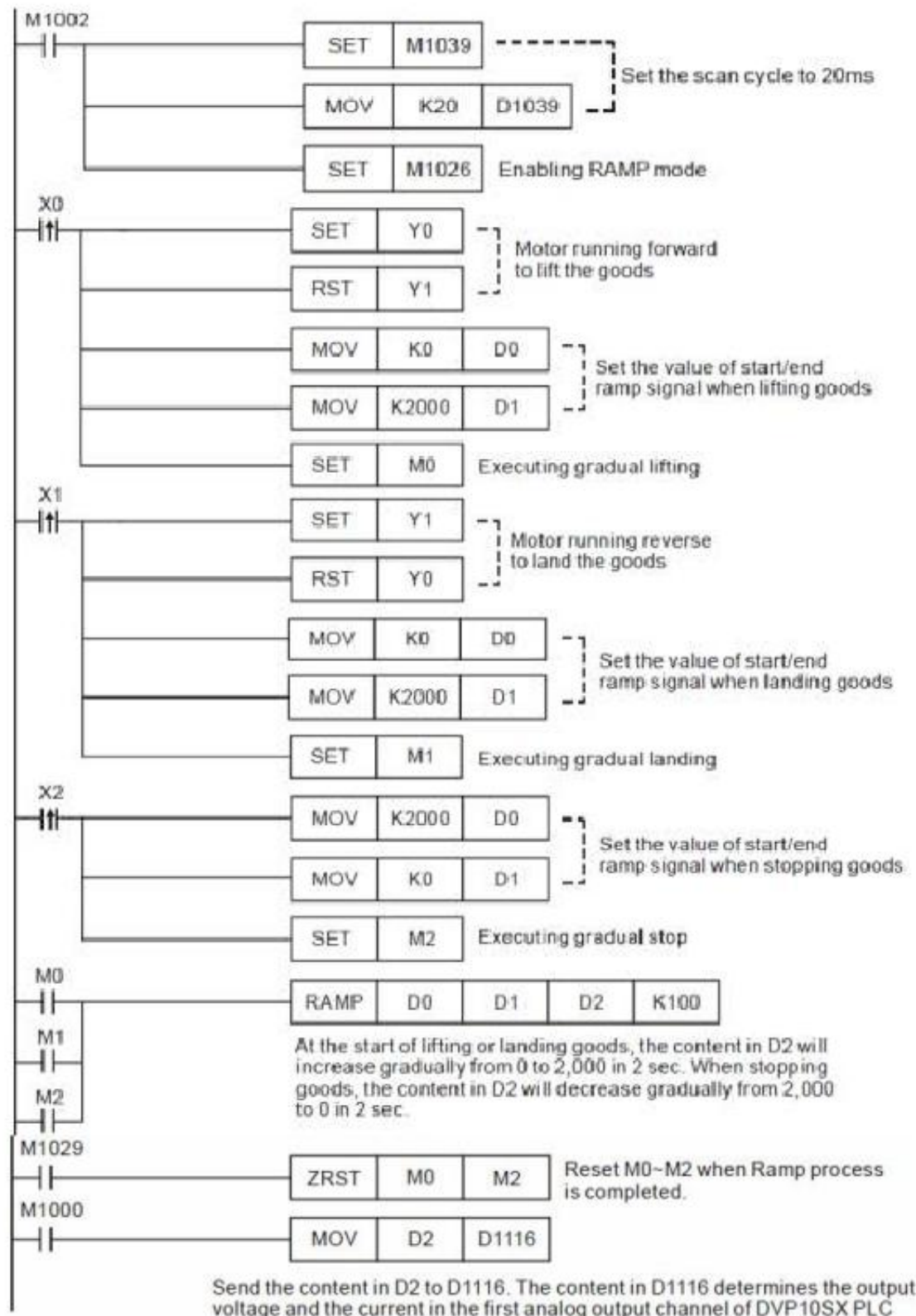
هدف از کنترل در این مثال به شرح زیر می‌باشد:

زمانی که بار جرثقیل خیلی زیاد باشد، الکتروموتور در هنگام شروع به حرکت به تدریج با یک شیب ملایم بالارونده راه‌اندازی می‌شود و هنگامی که فرایند توقف الکتروموتور اجرا شود، موتور با یک شیب ملایم پایین‌رونده متوقف خواهد شد.

در این مثال از یک ماژول خروجی آنالوگ عمومی DVP10SX استفاده شده که به CPU متصل می‌شود. این ماژول ولتاژ آنالوگ 0~10V تولید می‌کند. از این ولتاژ برای کنترل فرکانس اینورتر و راه‌اندازی خروجی فرکانس متغیر جهت کنترل سرعت چرخش الکتروموتور جرثقیل استفاده می‌شود. ابزارهای مورد استفاده در این مثال به شرح زیر است:

ابزار	توضیحات
X0	وقتی که X0=ON شود، فرایند شروع به حرکت الکتروموتور با یک شیب ملایم جهت بلند کردن بار توسط جرثقیل اجرا می‌شود. (Up button)
X1	وقتی که X1=ON شود، فرایند شروع به حرکت الکتروموتور با یک شیب ملایم جهت پایین آوردن بار توسط جرثقیل اجرا می‌شود. (Down button)
X2	وقتی که X2=ON شود، فرایند توقف الکتروموتور اجرا خواهد شد.
Y0	راه‌اندازی الکتروموتور در جهت راستگرد جهت بلند کردن بار متصل شده به جرثقیل
Y1	راه‌اندازی الکتروموتور در جهت چپگرد جهت پایین آوردن بار متصل شده به جرثقیل

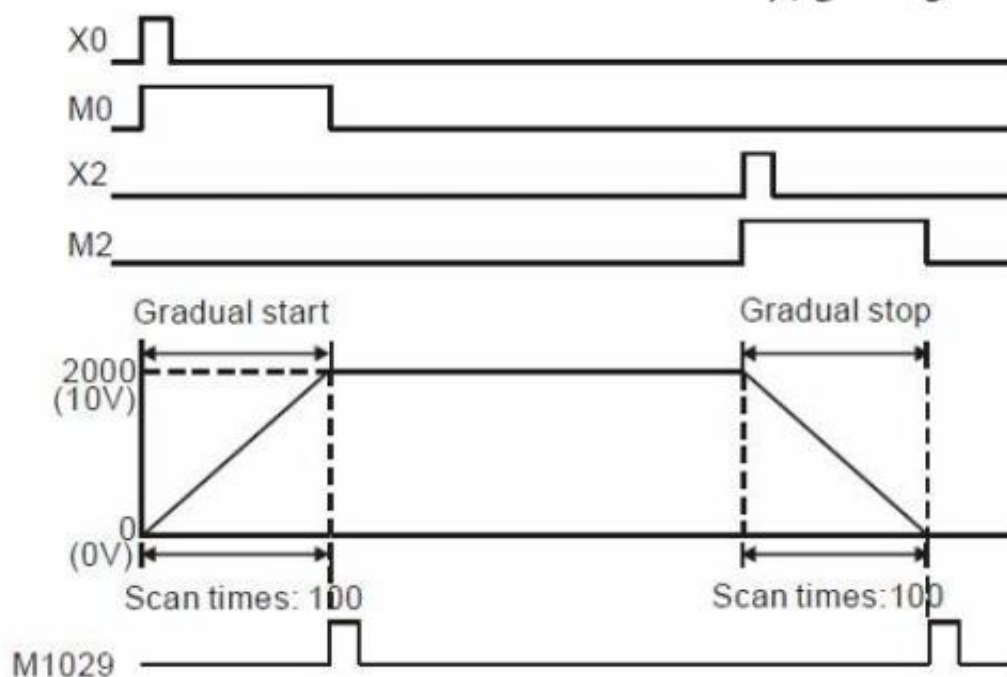
برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد این برنامه به شرح زیر است:

در این برنامه از PLC با تابع خروجی آنالوگ، از قبیل CPU سری DVP20EX و یا DVP10SX استفاده شده است. این به آن معنی است که این CPU ها به صورت کامپکت بوده و دارای خروجی آنالوگ بر روی خود می‌باشند. در CPU های سری DVP10SX، وقتی که مقدار داخل رجیستر خاص D1116 از K0 به K2000 تغییر کند، به آن معنی است که ولتاژ نخستین کانال خروجی آنالوگ از 0~10V تغییر کرده است.

پارامتر دستورالعمل RAMP مستقیماً وابسته به سیکل اسکن برنامه می‌باشد. بنابراین شما باید یک سیکل اسکن را در شروع نخستین برنامه تنظیم نمایید، سپس در این حالت مدت زمان سیگنال شیب می‌تواند ثابت باشد. در این برنامه، زمان ثابت یک سیکل اسکن برنامه برابر با 20ms بوده و تعداد دفعات اسکن دستورالعمل RAMP برابر با 100 بار می‌باشد. بنابراین مدت زمان شیب برابر با 2s است. وقتی که شستی Up button فشرده شود، M0=ON می‌شود، جرثقیل به تدریج و با یک شیب ملایم شروع به بلند کردن بار متصل شده به خود را کرده و ولتاژ خروجی آنالوگ PLC در مدت زمان 2 ثانیه از 0 به 10 ولت افزایش خواهد یافت. وقتی که جرثقیل به بالاترین حد خود رسید، اپراتور می‌تواند با فشردن دکمه STOP با فعال شدن M2=ON اجرای حرکت جرثقیل را متوقف نماید. در این حالت، ولتاژ خروجی، از کانال خروجی آنالوگ، در مدت 2 ثانیه از 10 ولت به 0 ولت کاهش خواهد یافت. فرآیند تغییر ولت مطابق با منحنی زیر است:

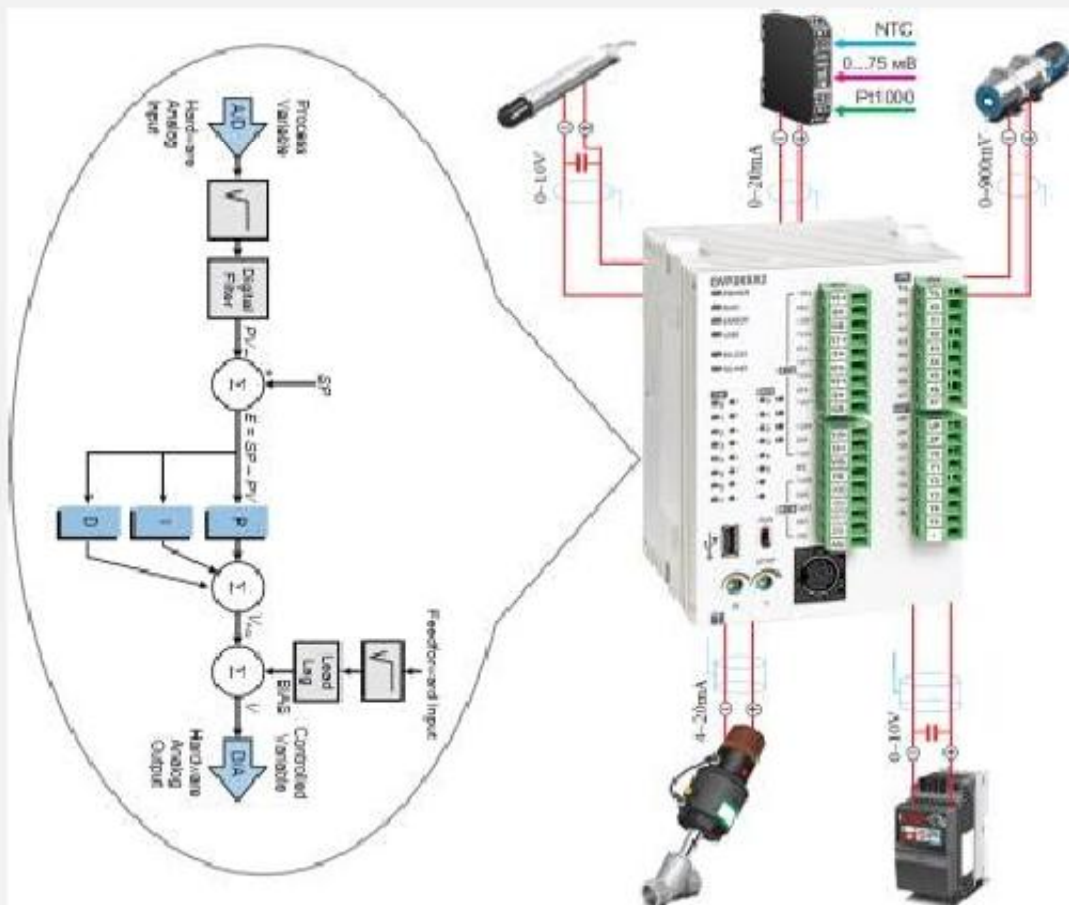


با فشردن دکمه Down button فرآیند حرکت به پایین با یک شیب ملایم و رسیدن سرعت الکتروموتور به سرعت تنظیمی در مدت 2 ثانیه صورت خواهد گرفت و توقف جرثقیل در نقطه مطلوب نیز با فشردن دکمه Stop button امکان پذیر است. فرآیند تغییر ولتاژ خروجی کانال 1 خروجی آنالوگ مطابق با توضیحات گفته شده در قسمت قبل می باشد.

فرکانس اینورتر متناسب با ولتاژ خروجی از PLC تغییر خواهد کرد. برای مثال، فرکانس اینورتر سری VFD-M دلتا از 0 تا 60 هرتز قابل تغییر می باشد. با استفاده از ولتاژ خروجی 0~10V آنالوگ از PLC سری DVP10SX می توان این فرکانس را کنترل کرد. بعلاوه، سرعت چرخش الکتروموتور متناسب با فرکانس خروجی از اینورتر می باشد. از این رو، مدت زمان شیب تدریجی شروع به کار و مدت زمان شیب تدریجی توقف جرثقیل را می توان توسط کنترل ولتاژ خروجی از PLC سری DVP10SX کنترل کرد. به این نکته دقت داشته باشید که در این مثال از CPU سری DVP10SX که دارای ورودی خروجی آنالوگ به صورت کامپکت می باشد، استفاده شده است. شما می توانید از CPU هایی که به صورت کامپکت ورودی خروجی آنالوگ ندارد نیز استفاده کنید و یک ماژول توسعه‌ی خاص خروجی آنالوگ را به CPU متصل کنید. این عمل همان کاری را انجام می دهد که CPU سری DVP10SX انجام می دهد.

فصل 5

کار با کنترل PID و کنترل فازی در PLC دلتا سری DVP



5-1 دستورالعمل خروجی PWM عمومی (GPWM)

API	Mnemonic	Operands	Function
144	GPWM	S_1 S_2 D	General PWM Output

Type	Bit Devices				Word Devices												Program Steps
OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	GPWM: 7 steps	
S_1													*				
S_2													*				
D		*	*	*													

PULSE										16-bit										32-bit									
ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2

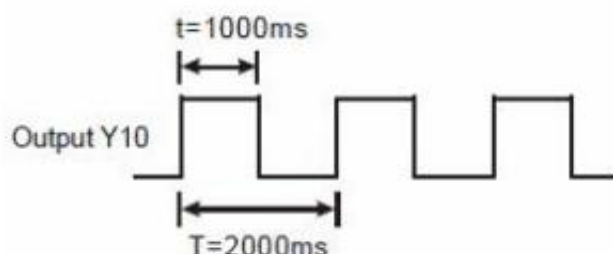
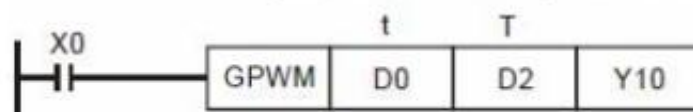
این دستورالعمل از 3 عملوند زیر تشکیل شده است:

- عملوند S_1 : در این عملوند پهنای پالس خروجی تنظیم می‌شود.
 - عملوند S_2 : در این عملوند سیکل پالس خروجی تنظیم می‌شود.
 - عملوند D : در این عملوند پالس خروجی تنظیم شده ظاهر خواهد شد.
- عملوند S_2 سه ابزار (رجیستر) متوالی را اشغال خواهد کرد. یعنی اگر رجیستر $D10$ را به این عملوند اختصاص دهید این عملوند رجیسترهای $D11$ و $D12$ را به صورت اتوماتیک اشغال خواهد کرد. اگر از دو رجیستر دیگر (یعنی $D11$ و $D12$ در این مثال) در برنامه‌نویسی استفاده نمایید، برنامه بد عمل کرده و خطا می‌دهد.

توجه داشته باشید که $S_1 \leq S_2$ باشد. در غیر این صورت، برنامه بد عمل کرده و خطا می‌دهد. محدوده‌ی قابل تنظیم برای عملوند S_1 می‌تواند بین $t = 0 \sim 32767ms$ باشد. محدوده‌ی قابل تنظیم برای عملوند S_2 می‌تواند بین $t = 1 \sim 32767ms$ باشد. دقت داشته باشید که از رجیسترهای S_2+1 و S_2+2 در برنامه‌نویسی نباید استفاده کنید. به عملوند خروجی پالس D می‌توانید ابزارهای بیتی Y و M یا S را اختصاص دهید. وقتی که دستورالعمل شروع به اجرا نماید، دستورالعمل GPWM مقادیر اختصاص داده شده به عملوندهای S_1 و S_2 را به پالس خروجی تبدیل کرده و به خروجی D ارسال خواهد کرد. وقتی که $S_1 \leq 0$ شود، پالسی در خروجی وجود نخواهد داشت. وقتی که $S_1 \geq S_2$ باشد، پالس خروجی فعال (ON) نگهداشته خواهد شد. وقتی که دستورالعمل GPWM در حال اجرا بود، شما می‌توانید مقادیر عملوند S_1 و S_2 را اصلاح نمایید.

مثال

در این مثال، می‌خواهیم یک پالس PWM با پهنای پالس 2000ms را ایجاد نماییم.



وقتی که ورودی $X0=ON$ شود، تنظیمات این دستورالعمل برابر با $D0=K1000$ و $D2=K2000$ باشد و خروجی $Y10$ به عملوند خروجی پالس اختصاص داده شده باشد، خروجی $Y1$ برای 1000ms فعال (on) بوده و برای 1000ms غیرفعال (off) می‌شود. این سیکل تا زمانی که ورودی $X0$ غیرفعال (OFF) شود، ادامه خواهد داشت.

توجه داشته باشید که از این دستورالعمل نباید در وقفه و یا زیرروال وقفه استفاده کرد. زمان این دستورالعمل توسط سیکل اسکن برنامه شمارش می‌شود. دقت داشته باشید که مقادیر S_1 و S_2 باید بزرگتر از زمان اسکن یک سیکل برنامه PLC باشد. در غیر این صورت، در هنگامی که دستورالعمل GPWM خروجی دارد، خطایی رخ خواهد داد.

5-2 دستورالعمل PID

API	Mnemonic			Operands				Function																					
88	D	PID			(S₁)	(S₂)	(S₃)	(D)	PID Control Loop																				
OP	Type	Bit Devices				Word Devices										Program Steps													
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PID : 9 steps	DPID: 17 steps											
	S ₁													*															
	S ₂													*															
	S ₃													*															
D														*															
PULSE										16-bit										32-bit									
ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2

این دستورالعمل از 4 عملوند تشکیل شده است که عبارتند از:

- عملوند S_1 : در این عملوند باید مقدار تنظیمی یا همان نقطه‌ی مطلوب (SV یا همان SP) را وارد نمایید.

- عملوند S_2 : در این عملوند مقدار جاری فرآیند (PV) قرار می‌گیرد.

- عملوند S_3 : در این عملوند باید نخستین ابزار یا همان رجیستر داده را جهت تنظیم کردن پارامترهای PID را وارد نمایید. این به آن معنی است که با اختصاص دادن یک رجیستر به این عملوند، 19 رجیستر متوالی بعدی توسط این عملوند جهت تنظیم پارامترهای PID اشغال می‌شود.

- عملوند D : در این عملوند مقدار خروجی (MV) دستورالعمل PID قرار می‌گیرد.

اگر از دستورالعمل 16 بیتی PID استفاده می‌کنید، عملوند S_3 تعداد 20 ابزار متوالی را اشغال می‌کند. و اگر از دستورالعمل 32 بیتی PID استفاده می‌کنید، دقت داشته باشید که عملوند S_3 تعداد 21 ابزار متوالی را اشغال خواهد کرد. بعنوان مثال، ما در برنامه‌نویسی از دستورالعمل 16 بیتی PID استفاده کرده‌ایم و اولین ابزاری که به عملوند S_3 اختصاص داده‌ایم، ابزار $D150$ می‌باشد. از این رو، عملوند S_3 ابزارهای $D150 \sim D169$ را اشغال خواهد کرد. بنابراین تعداد ابزارهای اشغال شده توسط عملوند S_3 برابر با 20 ابزار (رجیستر) متوالی خواهد بود. دقت کنید که اگر از یکی از این ابزارهای اختصاص داده شده به عملوند S_3 دستورالعمل PID در عملوند D دستورالعمل‌های دیگر استفاده کنید، برنامه بد عمل کرده و دچار خطا خواهد شد. ولی استفاده از این رجیسترها در دستورالعمل MOV و برخی از دستورالعمل‌های ریاضی جهت تنظیم مقادیر پارامترهای PID مانعی ندارد. از این دستورالعمل برای پیاده سازی کنترل PID در برنامه استفاده می‌شود. کنترل PID تنها در زمانی از اسکن برنامه اجرا خواهد شد که زمان نمونه‌برداری تکمیل شده باشد. بنابراین، ابتدا باید نمونه‌برداری از ورودی صورت بگیرد تا کنترل PID برای یک سیکل اسکن اجرا شود. این روند در هر اسکن تا زمانی که سیستم RUN است، اجرا می‌شود. از کنترل PID به طور وسیعی در ماشین‌ها پنوماتیک و تجهیزات الکترونیکی به کار برده می‌شود.

اصولاً کنترلر وظیفه‌ی مقایسه‌ی مقدار کنونی متغیر فرآیند (PV) با مقدار مطلوب یا همان نقطه‌ی تنظیم (SV یا SP) و صدور فرمان به عملگر فرآیند را برای کاهش خطای سیستم بر عهده دارد (خروجی - نقطه‌ی تنظیمی = خطا). از کنترلر PID در فرآیندهای کنترلی نظیر کنترل سرعت موتورهای الکتریکی، کنترل فشار، کنترلر دما و ... استفاده می‌شود.

یک کنترلر PID از سه مدل تناسبی، مشتق‌گیر، و انتگرال‌گیر بهره می‌برد که نام آن از P حرف اول Proportional یا تناسبی، I حرف اول Integral یا انتگرال‌گیر و D حرف اول Derivative یا مشتق‌گیر گرفته شده است.

پارامتر تناسبی (P) بیانگر بهره‌ی حلقه‌ی کنترل (Gain) یا نسبت تغییرات خروجی با خطا می‌باشد. از این پارامتر برای رفع مشکل قطع و وصل مکرر سیستم استفاده شده است.

پارامتر انتگرال‌گیر (I) بیانگر میزان پایداری (Stability) حلقه‌ی کنترل نسبت به تغییرات در ورودی یا میزان تغییرات خروجی نسبت به انتگرال خطای ماندگار می‌باشد. این کنترل‌کننده حافظه‌دار بوده و خروجی آن تحت تأثیر خطاهای سیستم در زمان گذشته می‌باشد. در این کنترلر تا زمانی که خطا مخالف با صفر باشد خروجی تغییر می‌کند و تغییرات خروجی جهت اصلاح خطا به کار گرفته می‌شود. این کنترلر کند بوده و در پاسخ تأخیر بوجود می‌آورد که احتمال ناپایداری را دارد.

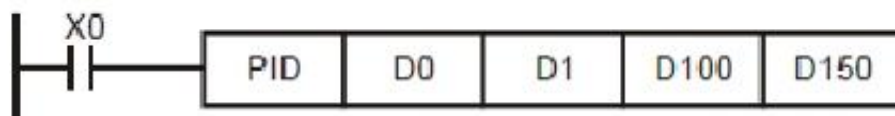
پارامتر مشتق‌گیر (D) بیانگر حساسیت خروجی نسبت به تغییرات ناگهانی ورودی یا نسبت به تغییرات خطا می‌باشد. این کنترلر تنها به تغییرات خطا و نه به مقدار آن حساس است. اگر خطا مقداری ثابت باشد و تغییر نکند، کنترلر هیچ عکس‌العملی نخواهد داشت. این کنترلر دیدی آینده‌نگر و پیش بین دارد. این کنترلر به پایداری سیستم می‌افزاید.

برای مقداردهی به پارامترهای تابع کنترلر PID می‌توان از همه‌ی رجیسترهای PLC استفاده نمود و به تعداد کانال‌های ورودی آنالوگ، تابع کنترلر PID را پیاده سازی کرد، اما هر دستورالعمل PID فقط یک حلقه را کنترل می‌کند.

تعداد پارامترهای دستورالعمل 16 بیتی PID بین $S_3 \sim S_3+19$ و تعداد پارامترهای دستورالعمل 32 بیتی PID بین $S_3 \sim S_3+20$ می‌باشد. بعد از اینکه تمام پارامترهای دستورالعمل PID تنظیم شدند، دستورالعمل PID شروع به اجرا می‌کند، و نتیجه‌ی بدست آمده از محاسبه‌ی فرمول PID در عملوند D ذخیره می‌شود. دقت داشته باشید که برای آدرس‌دهی عملوند D باید از ناحیه رجیستر داده بدون حافظه‌ی ماندگار استفاده نمایید. اگر شما یک ابزار از ناحیه رجیستر داده با حافظه ماندگار را به عملوند D اختصاص دهید، در هنگام شروع اجرای برنامه مقدار خروجی صفر نمی‌باشد بلکه مقدار داده‌ای که از پیش در حافظه‌ی رجیستر بوده است، به خروجی اعمال می‌شود. همین موضوع باعث می‌شود که سیستم کنترلر PID بد عمل کرده و خروجی مطلوب را به ما ندهد. از این رو، سیستم از ابتدای شروع عملکرد نوسانی است. بنابراین حتماً باید به عملوند D یک ابزار با حافظه‌ی فرار را اختصاص دهید.

مثال 1

در این مثال می‌خواهیم نحوه‌ی تنظیم پارامترهای دستورالعمل PID قبل از شروع اجرا دستورالعمل را توضیح دهیم.



وقتی که ورودی $X0=ON$ شود، دستورالعمل PID شروع به اجرا خواهد کرد و نتیجه‌ی بدست آمده از محاسبه‌ی فرمول PID در رجیستر D150 ذخیره می‌شود. وقتی که ورودی $X0=OFF$ شود، دستورالعمل PID اجرا نخواهد شد و دیتای قبلی در دستورالعمل بدون هیچ کم و کاستی باقی خواهد ماند. ولی دیتای موجود در رجیستر D150 برابر با 0 خواهد شد.

دقت داشته باشید که در PLC های سری ES, EX, SS با ورژن V5.7 و بالاتر از دستورالعمل PID پشتیبانی می‌کنند و همچنین تمام PLC های جدید تولید شده توسط شرکت دلتا از این دستورالعمل پشتیبانی می‌کنند.

در تعداد دفعات استفاده از کنترلر PID هیچ محدودیتی وجود ندارد. توجه داشته باشید که شماره رجیسترهای اختصاص داده شده به عملوند S_3 نباید تکراری باشد.

برای دستورالعمل 16 بیتی PID، عملوند S_3 تعداد 20 رجیستر متوالی را اشغال می‌کند. در این مثال، ناحیه‌ی اختصاص داده شده به عملوند S_3 برابر با D100~D119 می‌باشد. قبل از اجرای دستورالعمل PID، شما باید مقادیر تنظیم شده برای پارامترهای کنترلر PID را توسط دستورالعمل MOV به ناحیه رجیسترهای اختصاص داده شده به دستورالعمل PID منتقل نمایید. اگر رجیسترهای اختصاص داده شده به پارامترهای دستورالعمل PID حافظه‌دار بودند، برای انتقال تمام مقادیر تنظیم شده به پارامترهای PID در یک زمان باید از دستورالعمل MOVP استفاده نمایید.

تنظیمات عملوند S_3 برای دستورالعمل 16 بیتی PID در جدول زیر آورده شده است.

شماره ابزار	تابع	محدوده‌ی قابل تنظیم	توضیحات
S_3	زمان نمونه‌گیری (T_s) (واحد: 10ms)	1~2000 (واحد: 10ms)	اگر زمان تنظیمی برای T_s کمتر از زمان یک سیکل اسکن برنامه باشد، دستورالعمل PID فقط برای یک سیکل اسکن برنامه اجرا خواهد شد. اگر $T_s=0$ باشد، دستورالعمل PID فعال نخواهد شد. حداقل زمان تنظیمی T_s باید بیشتر از زمان یک سیکل اسکن برنامه باشد.
S_3+1	ضریب تناسبی (K_p)	0~30000%	توسط این پارامتر قادر به بزرگ کردن مقدار خطای تناسبی بین SV-PV می‌باشیم. تنظیم مقدار K_p بر روی عملیات مشتق‌گیر و انتگرال‌گیر تأثیر می‌گذارد. تأثیر افزایش ضریب K_p بر روی مشخصات پاسخ‌گذاری کنترل‌کننده می‌باشد با افزایش مقدار K_p خطای حالت ماندگار کاهش

می‌یابد ولی امکان نوسانی شدن سیستم وجود دارد.			
این پارامتر وابسته به مد کنترل K0~K8 می‌باشد. K_I زمان انتگرال‌گیر بوده و آن مدت زمانی است که طول می‌کشد تا خروجی انتگرال‌گیر، هنگامی که ورودی آن پله‌ی واحد است از مقدار صفر به مقدار واحد برسد. به عبارت ساده‌تر K_I زمان ریست می‌باشد این پارامتر به تنهایی سیستم را کند کرده و ناپایدار می‌کند. افزایش آن سیستم را کند کرده و کاهش K_I سیستم را نوسانی و ناپایدار می‌کند.	0~30000%	ضریب انتگرال‌گیر (K_I)	S_3+2
این پارامتر وابسته به مد کنترل K0~K8 می‌باشد این پارامتر باید بعد از K_P و K_I تنظیم شود و از آن در فرآیندهای کند و نوسانی استفاده می‌شود.	- 30000~300000%	ضریب مشتق‌گیر (K_D)	S_3+3
<p>0: کنترل اتوماتیک</p> <p>1: کنترل مستقیم ($E = SV - PV$)، خطا، SV مقدار تنظیم، PV مقدار جاری</p> <p>2: کنترل معکوس ($E = PV - SV$)</p> <p>3: خودتنظیم پارامترها برای کنترل دما. از این مد برای تنظیم اتوماتیک مقادیر پارامترهای K_D، K_I، K_P استفاده می‌شود. وقتی که عمل خودتنظیمی اتوماتیک تکمیل شد، مد کنترل به صورت اتوماتیک برابر با K_4 خواهد شد. از این مد به طور انحصاری برای کنترل دما استفاده می‌شود.</p> <p>4: از این مد به طور انحصاری برای تنظیم کردن کنترل دما استفاده می‌شود. این مد و مد 3 برای دستورالعمل 32 بیتی PID قابل تنظیم نمی‌باشد.</p> <p>5: کنترل اتوماتیک جهت: از این مد برای محدود کردن محدوده‌ی کران بالا و پایین عمل انتگرال‌گیری PID استفاده می‌شود. مد K_5 فقط برای CPUهای سری SV، $EH2$، $EH3$، $SV2$ با ورژن 1.2 و SA، SX با ورژن 1.8 و SC با ورژن 1.6 قابل دسترسی می‌باشد.</p> <p>6: نخستین مد کنترل دستی: در این مد شما باید یک MV را برای خروجی کنترل‌کننده‌ی PID تنظیم نمایید. در این مد مقدار انتگرال‌گیری شده روی هم انباشته شده مطابق با خطا افزایش می‌یابد. پیشنهاد می‌شود که از این مد کنترل برای کنترل محیط‌های که تغییرات آن بسیار کند است، استفاده شود. این مد در PLCهای سری $EH3$ با ورژن 1.62 و $SV2$ با ورژن 1.62 پشتیبانی می‌شود.</p> <p>7: دومین مد کنترل دستی: در این مد شما باید یک MV را برای خروجی کنترل‌کننده‌ی PID تنظیم نمایید. در این مد مقدار انتگرال‌گیری شده روی هم انباشته شده و افزایش مقدار خطا متوقف می‌شود. وقتی که مد کنترل به مد اتوماتیک تغییر کرد (مد کنترل برابر با K_5 شد)، خروجی‌های دستورالعمل PID یک مقدار انتگرال‌گیری شده‌ی مناسب را که روی هم انباشته شده‌اند را مطابق با آخرین مقدار خروجی MV انتخاب کرده و به دستورالعمل اعمال می‌کند. این مد در PLCهای سری $EH3$ با ورژن 1.62 و $SV2$ با ورژن 1.62 پشتیبانی می‌شود.</p> <p>توجه داشته باشید که PLCهای نام برده شده فقط برای ذکر مثال بوده و شما می‌توانید برای اطلاعات بیشتر به دفترچه‌ی راهنمای PLCهای دیگر مراجعه نمایید.</p>			
		مد کنترل	S_3+4

مقدار جاری - مقدار تنظیمی = خطا ($E = SV - PV$). وقتی که $S_3+5=0$ باشد، این تابع PID فعال نخواهد بود. بعنوان مثال، وقتی که S_3+5 برابر با مقدار 7 تنظیم شود، خروجی MV از خطا E بین 7- تا 7 نسبت به صفر خواهد بود.	0~32767	کار نکردن محدوده‌ی مقدار خطا (E)	S_3+5
Ex: اگر S_3+6 برابر با 1000 تنظیم شود، وقتی که خروجی MV بزرگتر از 1000 شود خروجی برابر با 1000 خواهد شد. S_3+6 باید بزرگتر و یا برابر با S_3+7 تنظیم شود.	-32768~32767	تنظیم محدوده‌ی کران بالای خروجی MV	S_3+6
Ex: اگر S_3+7 برابر با 1000 - تنظیم شود، وقتی که خروجی MV کمتر از 1000 - شود خروجی برابر با 1000 - خواهد شد.	-32768~32767	تنظیم محدوده‌ی کران پایین خروجی MV	S_3+7
Ex: اگر S_3+8 برابر با 1000 تنظیم شود، وقتی که مقدار انتگرال گیر بیشتر از 1000 شود، مقدار خروجی برابر با 1000 شده و عمل انتگرال گیری متوقف خواهد شد. پارامتر S_3+8 باید بزرگتر و یا برابر با S_3+9 تنظیم شود.	-32768~32767	محدود کردن کران بالای مقدار انتگرال گیر	S_3+8
Ex: اگر S_3+9 برابر با 1000 - تنظیم شود، وقتی که مقدار انتگرال گیر کمتر از 1000 - شود، مقدار خروجی برابر با 1000 - شده و عمل انتگرال گیری متوقف خواهد شد. اگر S_3+8 و S_3+9 برابر با صفر تنظیم شوند، محدوده‌ی کران بالا برای عمل انتگرال گیر وجود نخواهد داشت.	-32768~32767	محدود کردن کران پایین مقدار انتگرال گیر	S_3+9
از این پارامتر فقط برای رجوع کردن به مقدار انتگرال گرفته شده‌ی انباشته شده استفاده می‌شود. از این رو، شما می‌توانید این مقدار را پاک کنید و یا مطابق با نیازتان اصلاح نمایید.	مقدار اعشاری 32 بیتی	مقدار انتگرال گرفته شده‌ی انباشته شده	$S_3+10,11$
از این پارامتر فقط برای رجوع کردن به مقدار جاری (PV) قبلی استفاده می‌شود. از این رو، شما می‌توانید آن را مطابق با نیازتان اصلاح نمایید.	-32768~32767	مقدار جاری (PV) قبلی	S_3+12
از این پارامترها (رجیسترها) فقط برای سیستم (استفاده شده برای فرمول PID) استفاده می‌شود و در اختیار شما نمی‌باشد. از این رو، شما نباید هیچ گونه تغییری در این رجیسترها بوجود آورید.			$S_3+13\sim S_3+19$

کنترل کننده‌ی PID برای ثابت نگهداشتن خروجی سیستم در نقطه‌ی تنظیمی و نیز پاسخ سریع به اغتشاشات خارجی مورد استفاده قرار می‌گیرد. حال اگر پاسخ به اغتشاش‌های خارجی از اهمیت بیشتری برخوردار باشد (مقدار پارامترهای K_p ، K_i کوچک و مقدار پارامتر K_D بزرگ تنظیم می‌شوند)، پاسخ نهایی ما علاوه بر فراجهش (Overshoot) دارای نوسان نیز خواهد بود. بالعکس، اگر تثبیت خروجی در نقطه تنظیمی (بدون فراجهش) از اهمیت بالایی برخوردار باشد (پارامترهای K_p ، K_i بزرگ و مقدار پارامتر K_D کوچک تنظیم می‌شوند)، کنترل کننده قادر به پاسخ‌گویی سریع به اغتشاش‌ها نخواهد بود. در واقع، یک کنترل کننده‌ی PID قادر نخواهد بود هم پاسخ بدون فراجهش داشته باشد و هم به اغتشاش

ها پاسخ سریع دهد. از آنجایی که کنترل توسط دو PID، این عیب را از بین برده است، می‌توان هر دو نوع پاسخ را بهبود بخشید.

وقتی که مقدار تنظیمی پارامترها از محدوده‌ی قابل تنظیم‌شان تجاوز کنند، محدوده‌ی کران بالای تنظیم شده و محدوده‌ی کران پایین تنظیم شده جایگزین مقدار تنظیمی توسط کاربر خواهد شد. همچنین، اگر جهت حرکت (DIR) از محدوده‌ی امن تجاوز کند، مقدار 0 برای آن تنظیم خواهد شد. از دستورالعمل PID نمی‌توان در زیرروال‌های وقفه، برنامه مرحله (Step point) و دستورالعمل پرش CJ استفاده کرد.

فرمول حداکثر خطا از زمان نمونه‌برداری (T_S) به شرح زیر می‌باشد:

$$\text{Maximum error of sampling time } T_S = -(1 \text{ scan time} + 1\text{ms}) \sim + (1 \text{ scan time})$$

وقتی که خطا بر روی خروجی تأثیر گذاشت، شما باید ابتدا زمان اسکن برنامه را ثابت کنید و یا دستورالعمل PID را در یک زیرروال وقفه‌ی وابسته به زمان (timer) اجرا نمایید.

قبل از اجرا دستورالعمل PID مقدار دمای واقعی (PV) دستورالعمل PID باید ثابت باشد. اگر مقدار ورودی کنترل PID را از ماژول‌های DVP04AD/ DVP04DA/ DVP06XA/ DVP04PT/ DVP04TC دریافت می‌کنید، شما باید از زمان تبدیل آنالوگ به دیجیتال A/D ماژول‌ها مطلع باشید. در دستورالعمل 32 بیتی PID، اگر به پارامتر S_3 ابزار D100 را اختصاص دهید، ابزارهای D100~D120 به ناحیه تنظیم پارامتر اختصاص داده خواهد شد. از این رو، عملوند S_3 تعداد 21 رجیستر متوالی را اشغال می‌کند. قبل از اجرای دستورالعمل PID، باید ابتدا از دستورالعمل MOV برای ارسال مقدار تنظیمی پارامترها به پارامترهای دستورالعمل استفاده نمایید. اگر رجیسترهای اختصاص داده شده به پارامترهای دستورالعمل PID از نوع حافظه‌دار باشد، برای ارسال مقادیر تنظیمی پارامترها به تمام پارامترهای دستورالعمل به صورت همزمان باید از دستورالعمل MOVP استفاده نمایید. تنظیمات پارامترهای عملوند S_3 در دستورالعمل 32 بیتی PID مطابق با جدول زیر می‌باشد:

شماره ابزار	تابع	محدوده‌ی قابل تنظیم	توضیحات
S_3	زمان نمونه‌برداری (T_S) (واحد: 10ms)	1~2000 (واحد: 10ms)	اگر زمان تنظیمی برای T_S کمتر از زمان یک سیکل اسکن برنامه باشد، دستورالعمل PID فقط برای یک سیکل اسکن برنامه اجرا خواهد شد. اگر $T_S=0$ باشد، دستورالعمل PID فعال نخواهد شد. حداقل زمان تنظیمی T_S باید بیشتر از زمان یک سیکل اسکن برنامه باشد.
S_3+1	ضریب تناسبی (K_P)	0~300000%	مقدار خطای تناسبی بزرگ شده بین SV-PV می‌باشد.
S_3+2	ضریب انتگرال‌گیر (K_I)	0~300000%	وقتی که مد کنترل برابر با K0 یا K1 یا K2 یا K5 باشد، این پارامتر فعال خواهد شد.

S ₃ +3	ضریب مشتق‌گیر (K _D)	-30000~30000%	وقتی که مد کنترل برابر با K0 یا K1 یا K2 یا K5 باشد، این پارامتر فعال خواهد شد
S ₃ +4	جهت (مد) کنترل (DIR)	0: مد کنترل اتوماتیک 1: کنترل مستقیم (E = SV - PV) 2: کنترل معکوس (E = PV - SV) 3: خودتنظیم پارامترها برای کنترل دما: از این مد برای تنظیم اتوماتیک مقادیر پارامترهای K _P , K _I , K _D استفاده می‌شود. وقتی که عمل خودتنظیمی اتوماتیک تکمیل شد، مد کنترل به صورت اتوماتیک برابر با K4 خواهد شد. از این مد به طور انحصاری برای کنترل دما استفاده می‌شود. این مد و مد 4 برای دستورالعمل 32 بیتی PID قابل تنظیم نمی‌باشد. 4: از این مد به طور انحصاری برای تنظیم کنترل دما استفاده می‌شود. این مد و مد 3 برای دستورالعمل 32 بیتی PID قابل تنظیم نمی‌باشد. 5: مد اتوماتیک با کنترل محدوده‌ی کران بالا و پایین خروجی MV. وقتی که مقدار خروجی MV به محدوده‌ی کران بالا یا پایین رسید، انباشته کردن مقدار انتگرال گرفته شده متوقف می‌شود. 6: نخستین مد کنترل دستی: در این مد شما باید یک MV را برای خروجی کنترل‌کننده‌ی PID تنظیم نمایید. در این مد مقدار انتگرال‌گیری شده‌ی روی هم انباشته شده مطابق با خطا افزایش می‌یابد. پیشنهاد می‌شود که از این مد کنترل برای کنترل محیط‌های که تغییرات آن بسیار کند است، استفاده شود. این مد در PLC‌های سری EH3 با ورژن V1.62 و SV2 با ورژن V1.62 پشتیبانی می‌شود. 7: دومین مد کنترل دستی: در این مد شما باید یک MV را برای خروجی کنترل‌کننده‌ی PID تنظیم نمایید. در این مد مقدار انتگرال‌گیری شده‌ی روی هم انباشته شده و افزایش مقدار خطا متوقف می‌شود. وقتی که مد کنترل به مد اتوماتیک تغییر کرد (مد کنترل برابر با K5 شد)، خروجی‌های دستورالعمل PID یک مقدار انتگرال‌گیری شده‌ی مناسب را که روی هم انباشته شده‌اند را مطابق با آخرین مقدار خروجی MV انتخاب کرده و به دستورالعمل اعمال می‌کند. این مد در PLC‌های سری EH3 با ورژن V1.62 و SV2 با ورژن V1.62 پشتیبانی می‌شود. دقت داشته باشید که PLC‌های نام برده شده فقط برای ذکر مثال بوده و شما می‌توانید برای اطلاعات بیشتر به دفترچه‌ی راهنمای PLC‌های دیگر مراجعه نمایید.	0: مد کنترل اتوماتیک 1: کنترل مستقیم (E = SV - PV) 2: کنترل معکوس (E = PV - SV) 3: خودتنظیم پارامترها برای کنترل دما: از این مد برای تنظیم اتوماتیک مقادیر پارامترهای K _P , K _I , K _D استفاده می‌شود. وقتی که عمل خودتنظیمی اتوماتیک تکمیل شد، مد کنترل به صورت اتوماتیک برابر با K4 خواهد شد. از این مد به طور انحصاری برای کنترل دما استفاده می‌شود. این مد و مد 4 برای دستورالعمل 32 بیتی PID قابل تنظیم نمی‌باشد. 4: از این مد به طور انحصاری برای تنظیم کنترل دما استفاده می‌شود. این مد و مد 3 برای دستورالعمل 32 بیتی PID قابل تنظیم نمی‌باشد. 5: مد اتوماتیک با کنترل محدوده‌ی کران بالا و پایین خروجی MV. وقتی که مقدار خروجی MV به محدوده‌ی کران بالا یا پایین رسید، انباشته کردن مقدار انتگرال گرفته شده متوقف می‌شود. 6: نخستین مد کنترل دستی: در این مد شما باید یک MV را برای خروجی کنترل‌کننده‌ی PID تنظیم نمایید. در این مد مقدار انتگرال‌گیری شده‌ی روی هم انباشته شده مطابق با خطا افزایش می‌یابد. پیشنهاد می‌شود که از این مد کنترل برای کنترل محیط‌های که تغییرات آن بسیار کند است، استفاده شود. این مد در PLC‌های سری EH3 با ورژن V1.62 و SV2 با ورژن V1.62 پشتیبانی می‌شود. 7: دومین مد کنترل دستی: در این مد شما باید یک MV را برای خروجی کنترل‌کننده‌ی PID تنظیم نمایید. در این مد مقدار انتگرال‌گیری شده‌ی روی هم انباشته شده و افزایش مقدار خطا متوقف می‌شود. وقتی که مد کنترل به مد اتوماتیک تغییر کرد (مد کنترل برابر با K5 شد)، خروجی‌های دستورالعمل PID یک مقدار انتگرال‌گیری شده‌ی مناسب را که روی هم انباشته شده‌اند را مطابق با آخرین مقدار خروجی MV انتخاب کرده و به دستورالعمل اعمال می‌کند. این مد در PLC‌های سری EH3 با ورژن V1.62 و SV2 با ورژن V1.62 پشتیبانی می‌شود. دقت داشته باشید که PLC‌های نام برده شده فقط برای ذکر مثال بوده و شما می‌توانید برای اطلاعات بیشتر به دفترچه‌ی راهنمای PLC‌های دیگر مراجعه نمایید.
S ₃ +5,6	کار نکردن محدوده‌ی مقدار خطا (E)	0~2147483647	مقدار جاری - مقدار تنظیمی = خطا (E = SV - PV). وقتی که S ₃ +5,6 برابر با K0 تنظیم شود، این تابع از دستورالعمل فعال نخواهد بود. بعنوان مثال، وقتی که مقدار S ₃ +5,6 برابر با 4 تنظیم شود، خروجی MV در هنگام بروز خطا بین 4- و 4 نسبت به 0 خواهد شد.
S ₃ +7,8	تنظیم کران (محدوده‌ی) بالای مقدار خروجی (MV)	-2147483648 ~ 2147483647	Ex: اگر S ₃ +7,8 برابر با 1000 تنظیم شود، وقتی که خروجی MV بزرگتر از 1000 شود، خروجی برابر با 1000 خواهد شد. S ₃ +7,8 باید بزرگتر از 0 و یا برابر با S ₃ +7,8 تنظیم شود. در غیر این صورت، محدوده‌ی بالای خروجی با محدوده‌ی پایین خروجی سوئیچ خواهند شد.

تنظیم کران (محدوده‌ی پایین مقدار خروجی (MV)	-2147483648 ~ 2147483647	Ex : اگر $S_3+9,10$ برابر با 1000 - تنظیم شود، وقتی که خروجی MV کمتر از 1000 - شود خروجی برابر با 1000 - خواهد شد	$S_3+9,10$
محدود کردن کران بالای مقدار انتگرال گیر	-2147483648 ~ 2147483647	Ex : اگر $S_3+11,12$ برابر با 1000 تنظیم شود، وقتی که مقدار انتگرال گرفته شده بیشتر از 1000 شود، مقدار خروجی برابر با 1000 شده و عمل انتگرال‌گیری متوقف خواهد شد. پارامتر $S_3+11,12$ باید بزرگتر و یا برابر با $S_3+13,14$ تنظیم شود	$S_3+11,12$
محدود کردن کران پایین مقدار انتگرال گیر	-2147483648 ~ 2147483647	Ex : اگر $S_3+13,14$ برابر با 1000 - تنظیم شود، وقتی که مقدار انتگرال گرفته شده کمتر از 1000 - شود، مقدار خروجی برابر با 1000 - شده و عمل انتگرال‌گیری متوقف خواهد شد	$S_3+13,14$
مقدار انتگرال گرفته شده‌ی انباشته شده	عدد اعشاری 32 بیتی	از این پارامتر فقط برای رجوع کردن به مقدار انتگرال گرفته شده‌ی انباشته شده روی هم استفاده می‌شود. از این رو، شما قادر به پاک کردن و یا اصلاح کردن مطابق با نیازتان می‌باشید	$S_3+15,16$
مقدار جاری (PV) قبلی	از این پارامترها فقط برای رجوع کردن به مقدار جاری (PV) قبلی استفاده می‌شود. از این رو، شما می‌توانید آن را مطابق با نیازتان اصلاح نمایید		$S_3+17,18$
از این پارامترها (رجیسترها) فقط برای سیستم (استفاده شده برای فرمول PID) استفاده می‌شود و در اختیار شما نمی‌باشد. از این رو، شما نباید هیچ گونه تغییری در این رجیسترها بوجود آورید			$S_3+19 \sim S_3+20$

همانطور که متوجه شدید، توضیحات گفته شده برای عملوند S_3 دستورالعمل 32 بیتی مانند عملوند S_3 دستورالعمل 16 بیتی است. تنها اختلاف آنها در ظرفیت پارامترهای $S_3+5 \sim S_3+20$ می‌باشد. معادله‌ی PID استفاده شده در PLC‌های سری DVP دلتا به شرح زیر است.

فرمول پایه یا اصلی معادله‌ی PID:

$$MV = K_P \times E(t) + K_I \times E(t) \frac{1}{S} + K_D \times PV(t)S$$

$PV(t)S$ مقدار دیفرانسیل (مشتق) $PV(t)$ می‌باشد.

$E(t)/S$ مقدار انتگرال $E(t)$ ، وقتی که $E(t)$ بزرگتر از صفر است می‌باشد.

مد (جهت کنترل)	معادله‌ی PID
مستقیم، اتوماتیک	$E(t) = SV - PV$
معکوس	$E(t) = PV - PV$

عملکرد PID رفتاری مطابق با سرعت و دیفرانسیل مقدار واقعی فرایند (PV) دارد.

PID دارای سه جهت (مد) کنترل می‌باشد که عبارتند از: اتوماتیک، مستقیم و معکوس. جهت کنترل مستقیم یا معکوس را می‌توان توسط پارامتر S_3+4 تعیین کرد. دیگر تنظیمات وابسته به عملکرد PID توسط رجیسترهای $S_3 \sim S_3+5$ صورت می‌گیرد.

چنانچه جهت (مد) کنترل انتخاب شده مستقیم یا معکوس باشد، $E(t)$ وابسته به صفر خواهد بود. معادله‌ی توضیح داده شده در بالا برای این دستورالعمل با یک دستورالعمل PID عمومی که مقادیر دیفرانسیل آن توسط متغیرها استفاده می‌شوند، متفاوت می‌باشد. وقتی که یک دستورالعمل PID عمومی برای نخستین بار اجرا می‌شود، برای اجتناب از بزرگ شدن خطای مقدار دیفرانسیل گذرا، باید وضعیت مشتق‌گیری مقدار جاری فرآیند (PV) وابسته به دستورالعمل PID را مانیتور نماییم. وقتی که اختلاف مقدار جاری فرآیند (PV) نسبت به نقطه‌ی مطلوب (SV یا SP) خیلی زیاد شود، این دستورالعمل مقدار خروجی MV را کاهش خواهد داد تا این اختلاف را به حداقل یا صفر برساند. با مانیتور کردن این مقدار می‌توانید از سرعت عملکرد دستورالعمل مطلع شوید.

سمبل‌های استفاده شده در فرمول عمومی معادله‌ی PID به شرح زیر می‌باشد:

MV: مقدار خروجی

K_P : ضریب تناسبی

$E(t)$: مقدار خطا

PV: مقدار اندازه‌گیری شده‌ی جاری (فعلی) فرآیند تحت کنترل

SV: مقدار هدف، مقدار تنظیمی، نقطه‌ی مطلوب

K_D : ضریب مشتق‌گیر

$PV(t)S$: مقدار دیفرانسیل $PV(t)$

K_I : ضریب انتگرال‌گیر

$E(t) 1/S$: مقدار انتگرال گرفته شده‌ی $E(t)$

معادله‌ی کنترل دما به شرح زیر می‌باشد:

وقتی که پارامتر S_3+4 دستورالعمل PID برابر با K_3 و K_4 تنظیم شود، معادله‌ی PID به معادله‌ی زیر تغییر خواهد کرد.

$$MV = \frac{1}{K_P} \left[E(t) + \frac{1}{K_I} \left(E(t) \frac{1}{S} \right) + K_D \times PV(t)S \right]$$

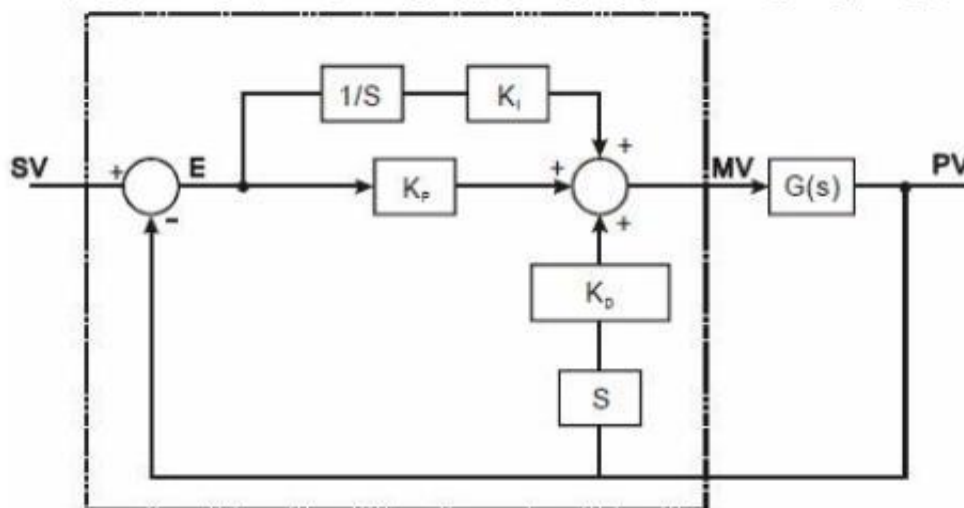
در این معادله مقدار خطا ثابت بوده و برابر با $E(t) = SV - PV$ می‌باشد

معادله گفته شده در بالا به صورت انحصاری به کنترل دما اختصاص داده شده است. از این رو، وقتی که زمان نمونه‌برداری (T_S) برابر با 4 ثانیه (K_400) تنظیم شود، محدوده‌ی مقدار خروجی MV بین K_0 تا K_4000 خواهد بود و همینطور زمان سیکل دستورالعمل GPWM استفاده شده نیز باید برابر با 4 ثانیه (K_4000) تنظیم شود تا هر دو دستورالعمل به درستی عمل کنند.

اگر تنظیمات شما برای پارامترهای دستورالعمل PID کنترل دما ایده‌آل نبود، شما می‌توانید مد خودتنظیم اتوماتیک (K3 Auto-tuning) را انتخاب نمایید و بعد از اینکه تمام پارامترها تنظیم شدند جهت (مد) کنترل به صورت اتوماتیک برابر با K4 تنظیم می‌شود. از این رو، می‌توانید پارامترهای دما را به بهترین نحو و یک بار مطابق با نتیجه کنترل اصلاح نمایید.

دیاگرام‌های کنترل PID مورد استفاده در PLC‌های دلتا به شرح زیر می‌باشند:

دیاگرام کنترل دستورالعمل PID عمومی در شکل زیر، در ناحیه‌ی خط چین شده قرار گرفته است:



برای فعال کردن دیاگرام کنترل دستورالعمل PID عمومی باید پارامتر S_3+4 را برابر با K0~K2 تنظیم کرد.

در دیاگرام نشان داده شده در بالا، S مقدار دیفرانسیل بوده و فرمول آن $S = PV - \text{previous PV} / \text{sampling time}$ است.

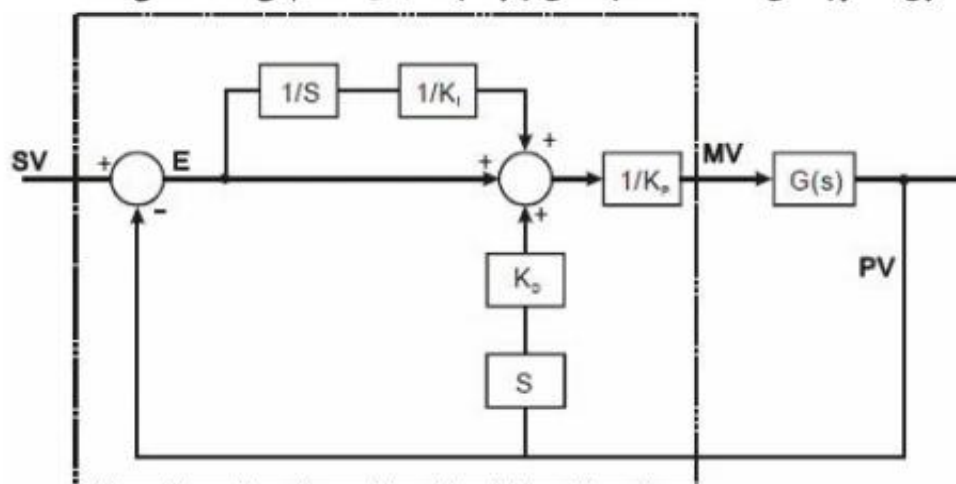
T_s زمان نمونه برداری / previous PV مقدار جاری قبلی فرایند - PV مقدار جاری فرایند $S =$ معادله دیفرانسیل

همچنین، در این دیاگرام $1/S$ معادله‌ی انتگرال گرفته شده بوده و فرمول آن $1/S = (\text{previous integral value} + \text{error value}) \times \text{sampling time}$ است.

زمان نمونه برداری \times (مقدار خطا + مقدار انتگرال گرفته شده‌ی قبلی) $1/S =$ معادله انتگرال

در این معادله $G(S)$ برای شروع کنترل ابزار خروجی در نظر گرفته شده است.

دیاگرام کنترل دستورالعمل PID دما در شکل زیر، در ناحیه‌ی خط چین شده نشان داده شده است:



برای فعال کردن دیاگرام کنترل دستورالعمل PID عمومی باید پارامتر S_3+4 را برابر با $K_3 \sim K_4$ تنظیم کرد.

در دیاگرام نشان داده شده در بالا، $1/K_P$ و $1/K_I$ برای کنترل دما رزرو شده‌اند. شما باید همراه با دستورالعمل PID از دستورالعمل GPWM استفاده کنید. برای اطلاع از جزئیات بیشتر به برنامه 3 در همین قسمت مراجعه نمایید.

برای کار با دستورالعمل PID باید به نکات زیر دقت نمایید:

- پارامترهای $S_3+6 \sim S_3+13$ فقط در CPU های سری SV, EH, EH2, SC, SX, SA و SS, EX, ES با ورژن 5.6 V و بالاتر قابل دسترس هستند.

- در CPU های سری SS, EX, ES با ورژن 5.6 V و بالاتر از دستورالعمل PID فقط یکبار می‌توان استفاده کرد. در CPU های سری SS, EX, ES با ورژن 5.7 V و بالاتر و SA, SX, SC, EH, EH2, SV, EH3, SV2 هیچ محدودیتی در تعداد دفعات استفاده از دستورالعمل PID وجود ندارد.

- مقدار پارامتر S_3+3 را در CPU های سری SS, EX, ES با ورژن 5.7 V و بالاتر، SA, SX, SC با ورژن 1.1 V و بالاتر، EH با ورژن 1.0 V و بالاتر فقط باید در محدوده‌ی 0~30000 تنظیم کرد.

- برای اطلاع از جزئیات CPU های دیگر به دفترچه‌ی راهنمای آنها مراجعه نمایید.

- مکان استفاده از دستورالعمل PID دارای اهمیت می‌باشد. از این رو، باید تابع کنترل مناسب را انتخاب نمایید. برای مثال، وقتی که پارامتر خودتنظیم اتوماتیک (Auto-tuning) را انتخاب می‌کنید

($S_3+4=K3$)، قادر به استفاده از آن در یک وسیله کنترل موتور (اینورتر) نیستید. دلیل آن نامناسب بودن تابع کنترل برای موتور می‌باشد که این امر موجب خسارت در سیستم خواهد شد.

- وقتی که شما سه پارامتر اصلی ($S_3+4=K0\sim K2$) K_P , K_I , K_D را تنظیم می‌کنید، باید ابتدا پارامتر K_P مطابق با تجربه‌تان (منظور از تجربه همان آزمون خطا می‌باشد) تنظیم نمایید و در این حالت باید پارامترهای K_I و K_D را برابر با صفر تنظیم نمایید در این هنگام می‌توانید کنترل دستی سریع داشته باشید که این نوع کنترل ناهنجار می‌باشد، در این صورت، باید مقدار K_I و K_D را تنظیم کنید. با افزایش مقدار این دو پارامتر می‌توانید یک کنترل دستی داشته باشید که با آزمون خطا می‌توانید مقادیر آنها را تنظیم نمایید. در مثال 4 چگونگی تنظیم این پارامترها و نحوه‌ی افزایش آنها توضیح داده شده است. دقت داشته باشید که $K_P=100$ به معنی 100% می‌باشد. این به آن معنی است که ضریب تقویت خطا برابر با 1 است. از این رو، اگر $K_P<100\%$ شود، خطا کاهش می‌یابد و اگر $K_P>100\%$ شود، خطا افزایش خواهد یافت.

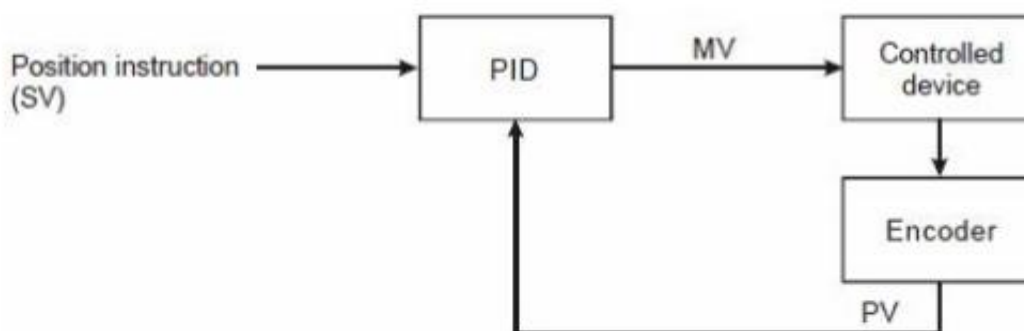
- وقتی که شما پارامترهای ویژه‌ی کنترل دما را انتخاب می‌کنید ($S_3+4=K3, K4$)، پیشنهاد می‌شود که مقدار پارامترها را در رجیستر D حافظه‌دار ذخیره کنید. دلیل این موضوع، تنظیم اتوماتیک پارامترها بعد از قطع تغذیه‌ی PLC و وصل مجدد تغذیه‌ی PLC می‌باشد. هیچ ضمانتی برای تنظیم مناسب پارامترهای کنترلر PID وجود ندارد. بنابراین، می‌توانید تنظیمات پارامترها را مطابق با نیاز واقعی خودتان تنظیم نمایید. اما پیشنهاد ما به شما این است که برای هر چه بهتر کردن کنترل فرآیند توسط دستورالعمل PID فقط پارامترهای K_I یا K_D را تنظیم نمایید.

- دستورالعمل PID توانایی کار با تمام پارامترها را دارد. از این رو، پیشنهاد می‌شود که پارامترها را به صورت تصادفی اصلاح نکنید. دلیل این موضوع آن است که کنترلر PID در این حالت به صورت نرمال اجرا نخواهد شد.

مثال 1

می‌خواهیم از دستورالعمل PID برای کنترل موقعیت استفاده کنیم.

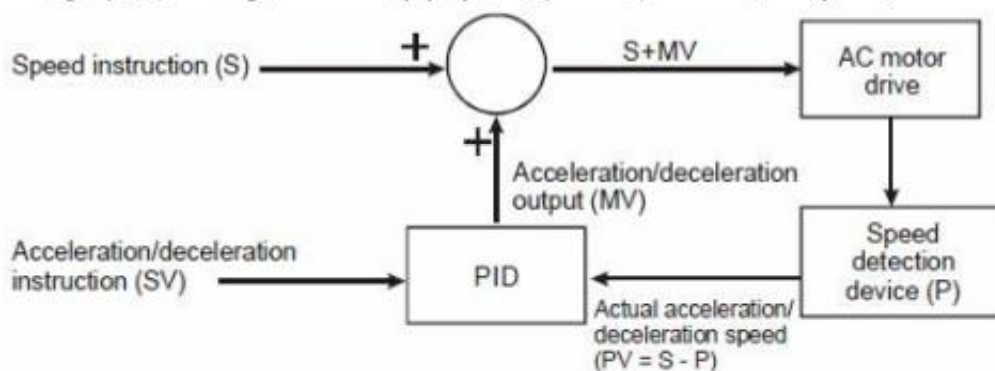
برای کنترل موقعیت توسط دستورالعمل PID باید پارامتر $S_3+4=0$ تنظیم شود. دیاگرام استفاده از دستورالعمل PID جهت کنترل موقعیت مطابق با شکل زیر می‌باشد:



مثال 2

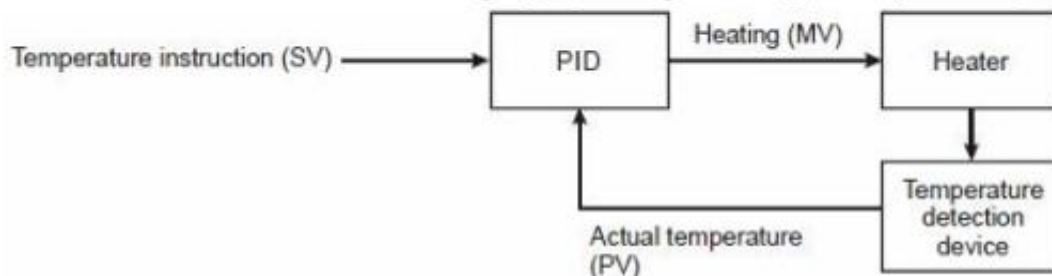
می‌خواهیم از دستورالعمل PID برای کنترل اینورتر استفاده نماییم. برای کنترل یک اینورتر (درایو راه‌انداز و کنترل موتور AC) توسط دستورالعمل PID باید پارامتر $S_3+4=0$ تنظیم شود.

دیاگرام استفاده از دستورالعمل PID جهت کنترل درایو موتور AC مطابق با شکل زیر می‌باشد:



مثال 3

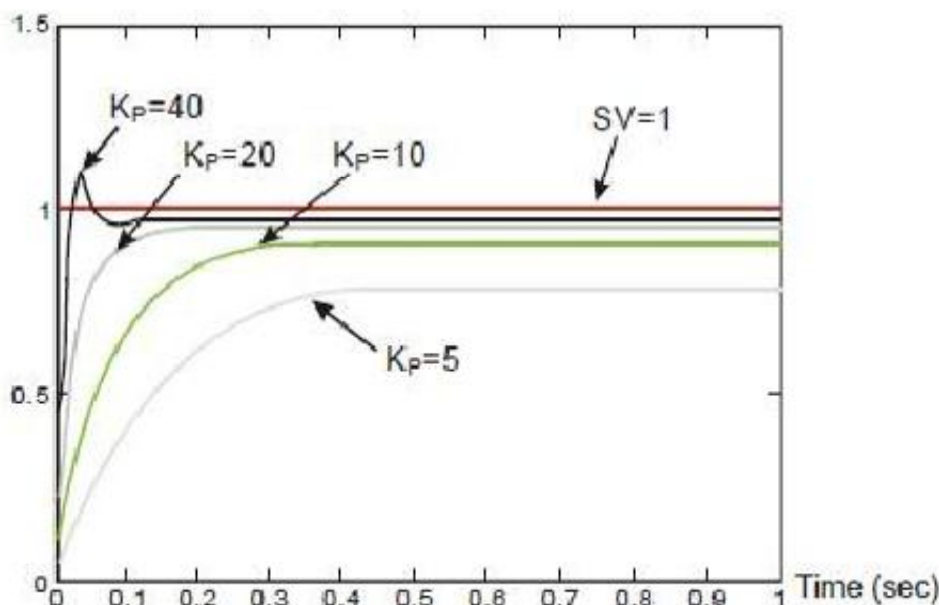
می‌خواهیم از دستورالعمل PID برای کنترل دما استفاده نماییم. برای کنترل دما توسط دستورالعمل PID باید پارامتر $S_3+4=1$ تنظیم شود. دیاگرام استفاده از دستورالعمل PID جهت کنترل دما مطابق با شکل زیر می‌باشد:



مثال 4

در این مثال می‌خواهیم چگونگی تنظیم پارامترهای دستورالعمل PID را بررسی نماییم. فرض کنید که تابع انتقال (تابع تبدیل) از ابزار کنترل شده $G(S)$ در یک سیستم کنترل یک تابع مرتبه‌ی اول $G(S) = b / S + a$ باشد. بیشتر مدل‌های کنترلی موتورها، این تابع مرتبه‌ی اول می‌باشد. همچنین، فرض کنید که مقدار هدف $SV=1$ و زمان نمونه‌برداری $T_s=10ms$ باشد. برای تنظیم پارامترهای این تابع پیشنهاد می‌شود که مراحل گفته شده در زیر را طی کرده و پارامترهای تابع را مطابق با این مراحل تنظیم نمایید.

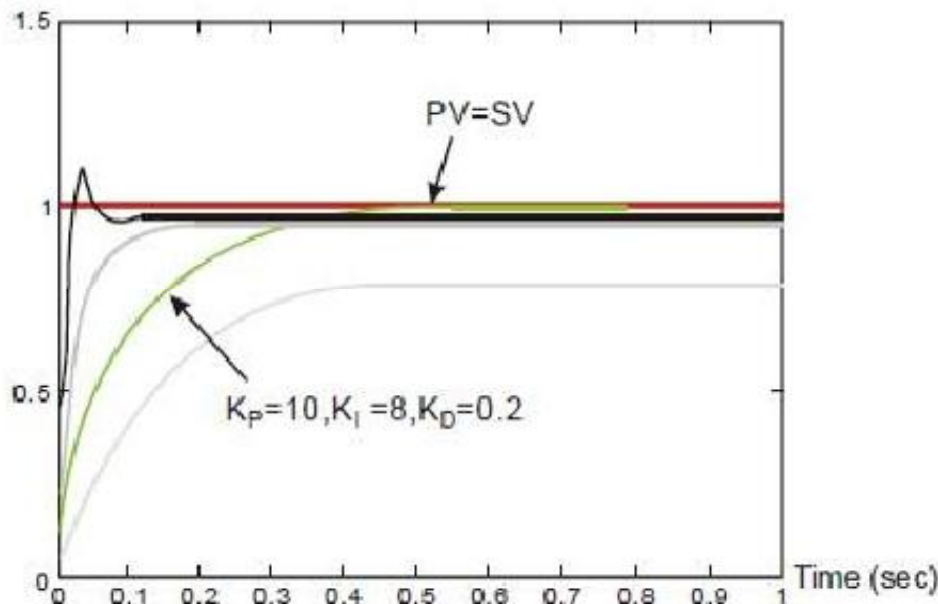
مرحله‌ی 1: مقدار پارامترهای K_D و K_I را برابر با صفر و مقدار پارامتر K_P را به ترتیب در چندین مرحله برابر با 5، 10، 20 و 40 تنظیم نمایید. مقدار SV و PV به ترتیب ثبت شده و نتیجه در شکل زیر نشان داده شده است.



مرحله‌ی 2: مطابق با شکل بالا می‌توان متوجه شد که وقتی که $K_P=40$ است، واکنش خروجی خیلی سریع بوده و یک فراجهش نیز ایجاد شده است. از این رو، بهتر است که مقدار K_P برابر با 40 تنظیم نشود. وقتی که $K_P=20$ است، واکنش منحنی PV نزدیک به مقدار SV بوده و واکنش سریع یا فراجهش (Over-reaction) وجود نخواهد داشت. اما به علت راه‌اندازی سریع و با ناپایداری بزرگ (زودگذر) خروجی MV ، ما باید به قرار داشتن موقعیت PV نسبت به SV دقت داشته باشیم. از این رو، بهتر است که مقدار K_P برابر با 20 تنظیم نشود. وقتی که $K_P=10$ است، واکنش منحنی PV نزدیک به مقدار SV خواهد بود، اما در این حالت سیستم کندتر شده و PV با شیب ملایم بیشتر به SV نزدیک می‌شود. بنابراین ما از این مقدار رضایت بیشتری خواهیم داشت. درنهایت، وقتی که $K_P=5$

باشد، به علت کند بودن بیش از حد سیستم و خیلی دیر رسیدن مقدار PV به مقدار SV این مقدار مطلوب نبوده و ما از آن استفاده نمی‌کنیم.

مرحله‌ی 3: در این مرحله ما $K_P=10$ تنظیم خواهیم کرد و مقدار K_I را از کم تا زیاد بعنوان مثال 1 و 2 و 4 و 8 تنظیم می‌کنیم. دقت داشته باشید که مقدار K_I نباید بزرگتر از K_P تنظیم شود. در نهایت باید مقدار K_D را از کم تا زیاد بعنوان مثال 0.01 و 0.05 و 0.1 و 0.2 تنظیم کنیم. دقت داشته باشید که مقدار K_D نباید از 10% نسبت به K_P تجاوز کند. در نهایت باید بهترین مقادیر را انتخاب کرده و در پارامترهای K_P و K_I و K_D قرار دهیم. در این مثال بهترین مقادیر بدست آمده و همچنین منحنی آن در شکل زیر نشان داده شده است.

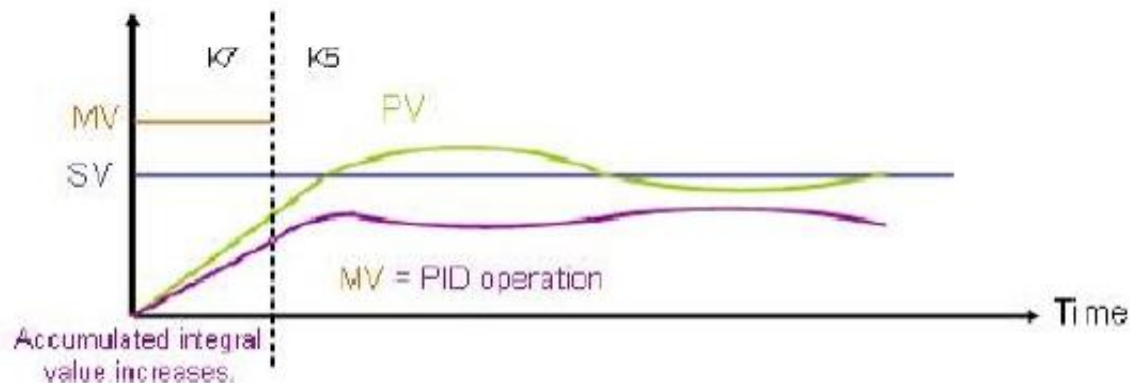


توجه داشته باشید که این مثال یک مرجع برای درک بهتر شما نسبت به عملکرد PID می‌باشد. بنابراین تنظیمات پارامترهای دستورالعمل PID برای فرآیندی که آن را پیاده سازی می‌کنید، باید مطابق با شرایط واقعی سیستم کنترلی خود باشد.

مثال 5

در این مثال، می‌خواهیم نحوه‌ی سوئیچ شدن بین نخستین مد دستی (K7) و مد اتوماتیک (K5) را بررسی نماییم.

ابتدا باید تنظیمات پارامترهای PID را انجام دهید، سپس بعد از اینکه تنظیمات پارامترهای K_P ، K_I و K_D تکمیل شد و اگر دستورالعمل PID در مد کنترل اولین مد کنترل دستی (K7) بود، منحنی کنترل مطابق با شکل زیر می‌باشد:



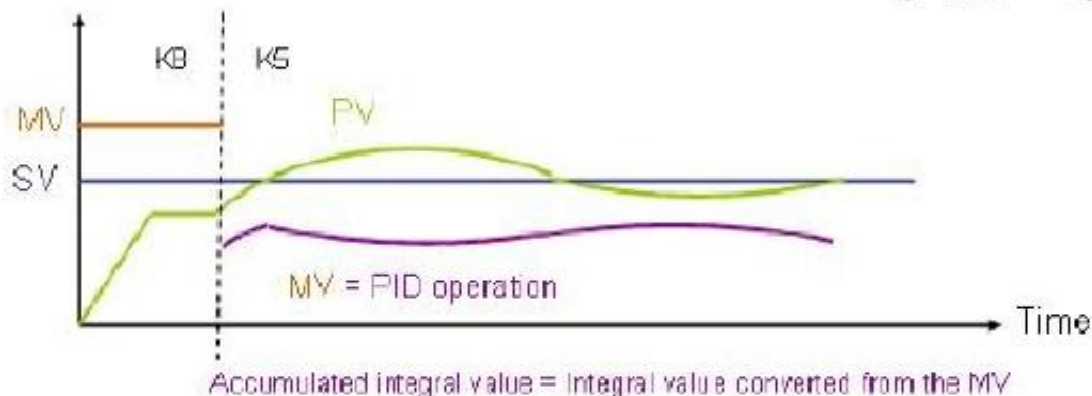
شکل بالا علاوه بر تغییر مد کنترل دستی به مد کنترل اتوماتیک، بیانگر مقدار انتگرال گرفته شدهی انباشته شده به صورت افزایشی در مد کنترل دستی می‌باشد.

اگر مد کنترل به مد اتوماتیک (K5) تغییر کرد، مقدار خروجی MV دستورالعمل PID از مقدار خروجی تنظیم شده توسط کاربر به مقدار خروجی تنظیم شده توسط دستورالعمل PID تغییر خواهد کرد.

مثال 6

در این مثال، می‌خواهیم سوئیچ شدن بین دومین مد دستی (K8) و مد اتوماتیک (K5) را بررسی نماییم.

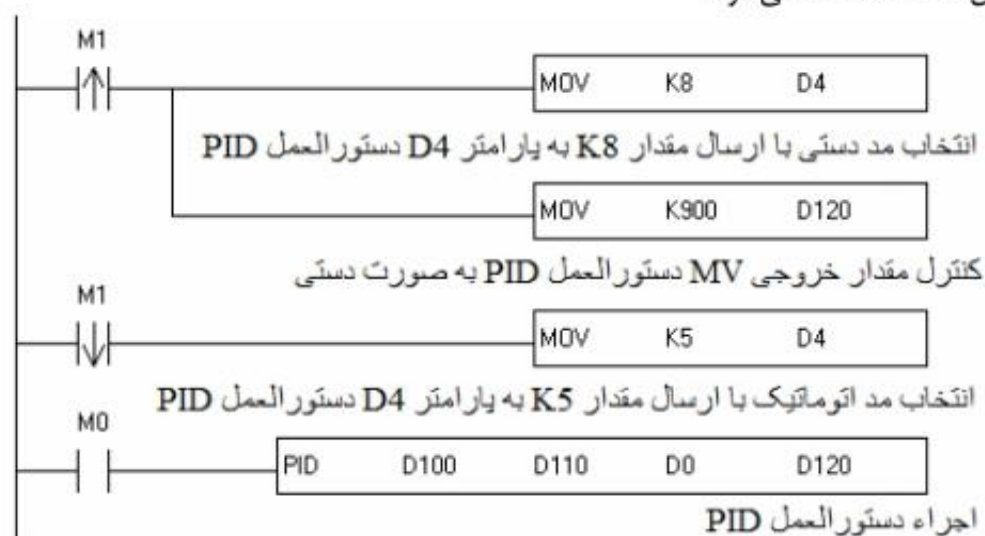
ابتدا باید تنظیمات پارامترهای PID را انجام دهید، بعد از اینکه تنظیمات پارامترهای K_P ، K_I و K_D تکمیل شد و اگر دستورالعمل PID در مد کنترل دومین مد کنترل دستی (K8) بود، منحنی کنترل مطابق با شکل زیر می‌باشد:



اگر دومین مد کنترل دستی به مد اتوماتیک (K5) تغییر کند، مقدار انتگرال گرفته شدهی انباشته شده به مقدار انتگرال گرفته شده از آخرین مقدار خروجی MV تبدیل شده و مقدار انتگرال گرفته شدهی انباشته شده به مقدار خروجی از عملکرد PID تبدیل می‌شود.

برنامه‌ای را برای مثال 5 و 6 نوشته‌ایم که در شکل زیر نشان داده شده است. در این برنامه، از فلگ M0 برای فعال کردن دستورالعمل PID استفاده شده است. وقتی که M1 فعال (ON) شود، از مد

دستی دستورالعمل PID استفاده می‌شود. وقتی که M1 غیرفعال OFF شود، از مد اتوماتیک دستورالعمل PID استفاده می‌شود.

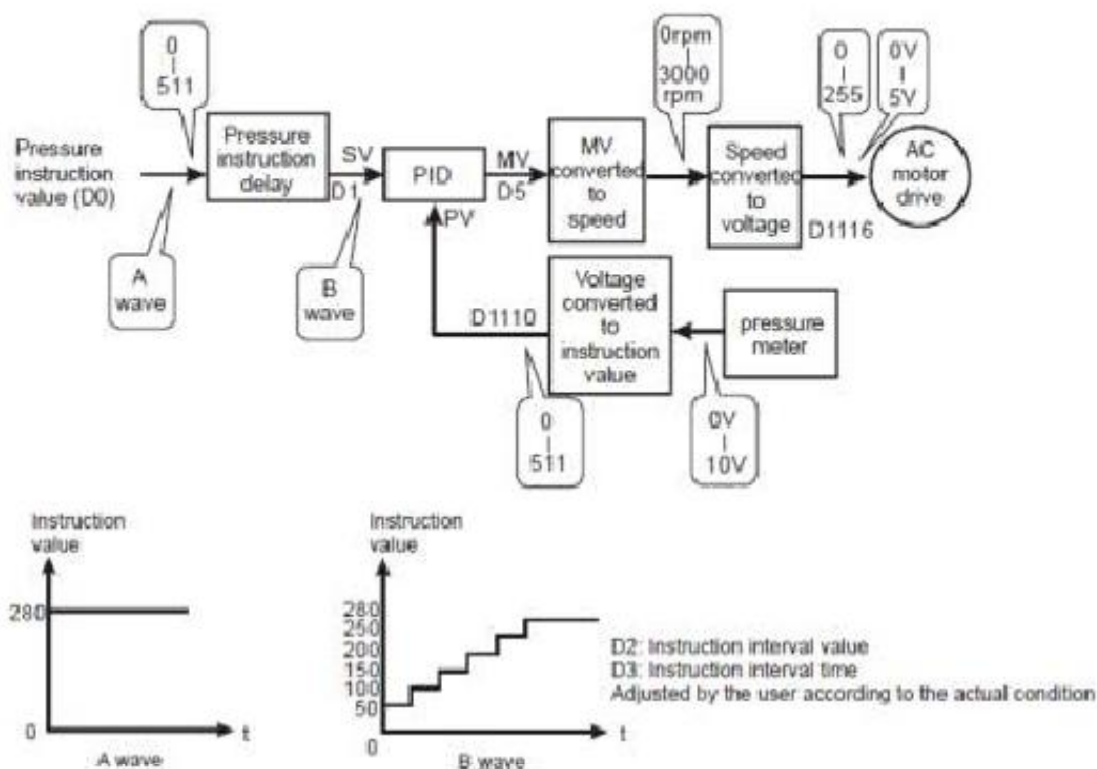


برنامه 1

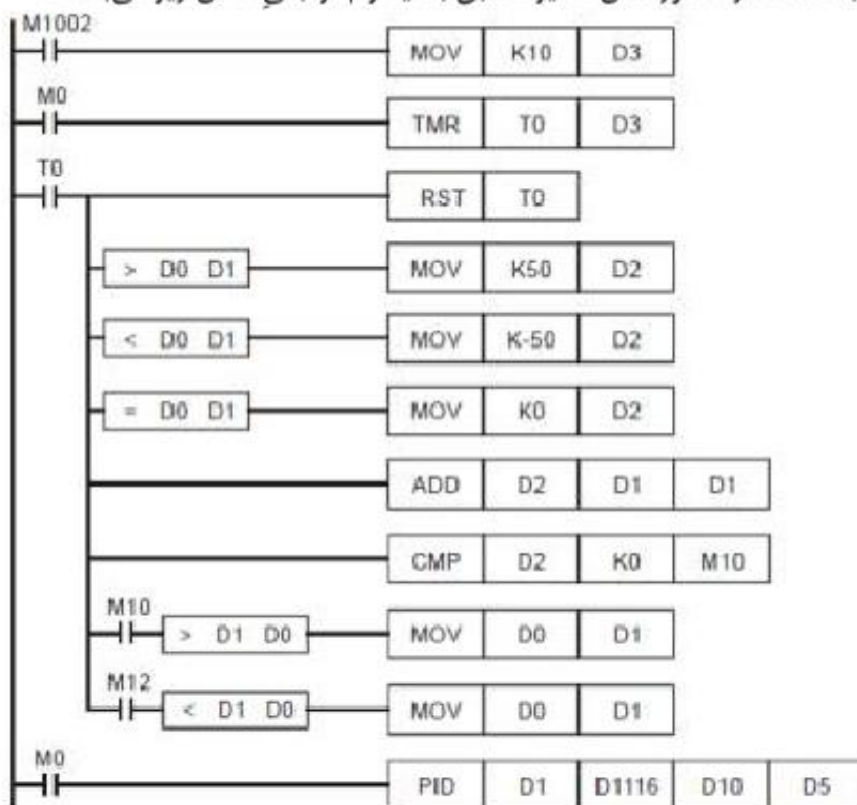
در این برنامه از دستورالعمل PID برای کنترل سیستم فشار استفاده شده است. در این برنامه از دیاگرام نشان داده شده در مثال 1 استفاده شده است.

هدف از کنترل در این برنامه فعال کردن سیستم کنترل برای رسیدن فشار واقعی به فشار تنظیم شده می‌باشد. در این سیستم به کنترلی با شیب تدریجی و آهسته نیاز است. بنابراین، اگر فشار فرایند تحت کنترل خیلی سریع (زیاد) باشد، در سیستم فراجاهش بوجود خواهد آمد و یا سیستم از کنترل خارج می‌شود.

راه حل پیشنهادی این است که یا زمان نمونه‌برداری (T_s) را طولانی‌تر کنید و یا از دستورالعمل تأخیر مطابق با دیاگرام زیر استفاده کنید.



برنامه این مثال با استفاده از دستورالعمل تأخیر مطابق با دیاگرام نردبانی شکل زیر می‌باشد:

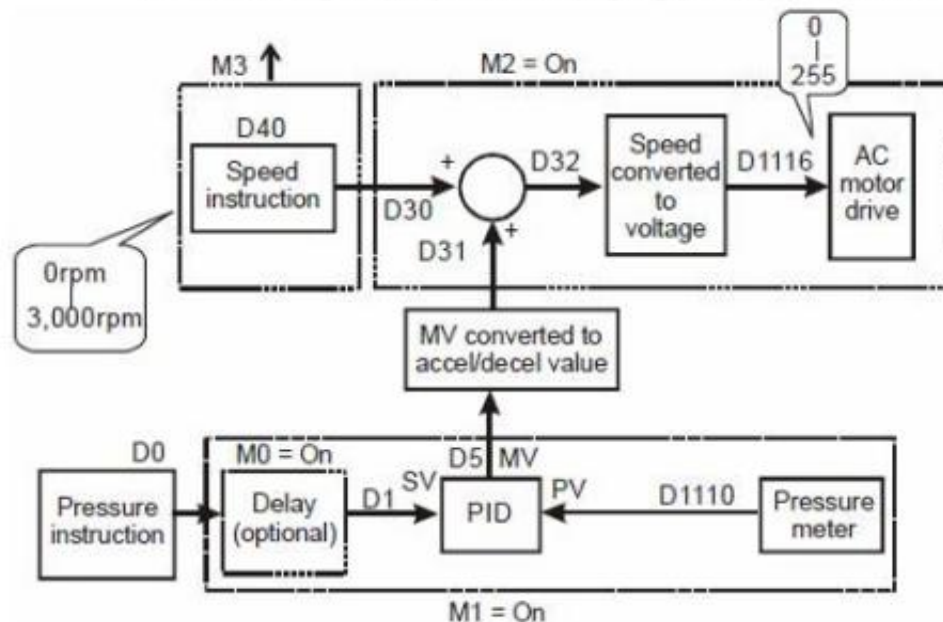


برنامه 2

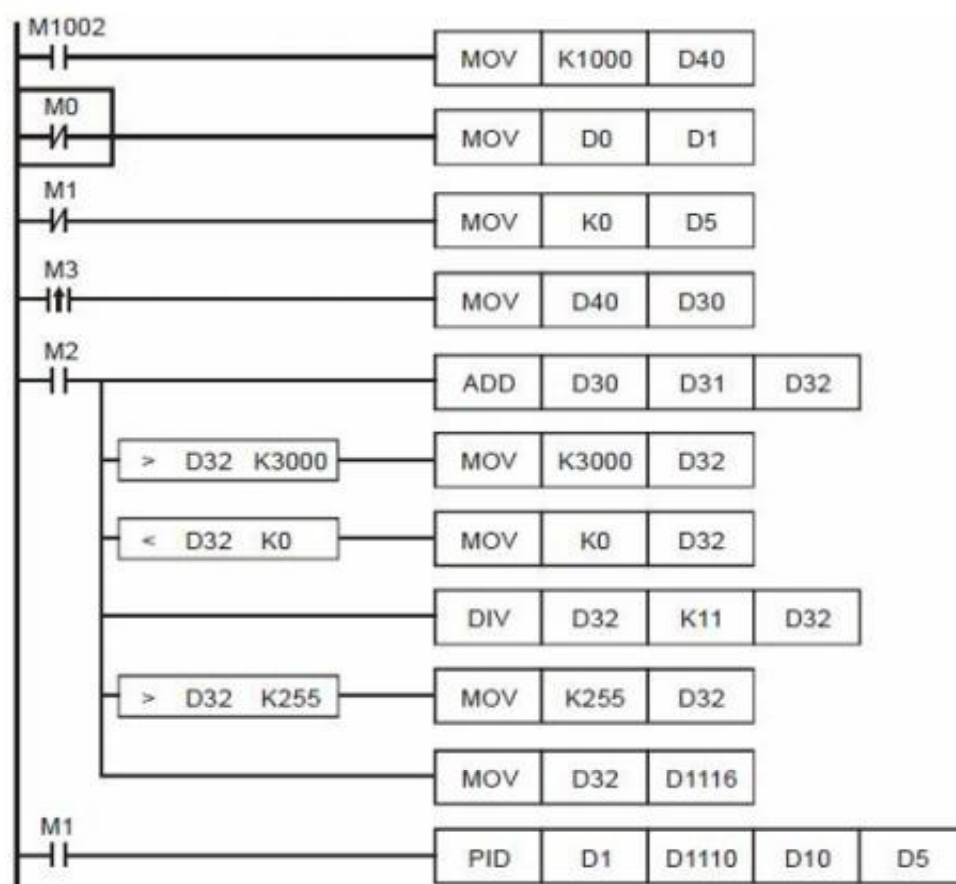
در این برنامه می‌خواهیم با سیستم کنترل سرعت و سیستم کنترل فشار به صورت منحصر بفرد و جدا از یکدیگر کار کنیم.

در این برنامه از دیاگرام نشان داده شده در مثال 2 استفاده شده است. هدف از کنترل در این برنامه، این است که بعد از به کار انداختن کنترل سرعت در یک حلقه‌ی باز برای یک دوره‌ی زمانی، آن را داخل سیستم کنترل فشار (دستورالعمل PID) برای کنترل حلقه‌ی بسته اضافه کنیم. از این رو، از یک سیستم کنترل حلقه باز برای حلقه بسته کردن یک سیستم کنترلی دیگر استفاده می‌شود.

از آنجایی که سیستم‌های کنترل سرعت و کنترل فشار به یکدیگر وابسته نیستند، ما باید ابتدا از ساختار یک حلقه‌ی باز برای کنترل سرعت پیرو یک حلقه‌ی بسته‌ی کنترل فشار استفاده نماییم. از این رو، با استفاده از این روش ما یک حلقه‌ی بسته را به یک حلقه‌ی باز نسبت داده‌ایم. برای درک بهتر این مطلب، به دیاگرام نشان داده شده در زیر مراجعه نمایید. اگر احتمال می‌دهید که در این سیستم کنترل فشار، تغییرات سریع وجود دارد، می‌توانید به این برنامه یک دستورالعمل تأخیر داخل برنامه کنترلی اضافه کنید. برای اضافه کردن دستورالعمل تأخیر داخل برنامه‌ی کنترلی توسط PID به توضیحات داده شده در برنامه 1 مراجعه نمایید. دیاگرام کنترلی این برنامه در شکل زیر نشان داده شده است.



در شکل زیر، قسمتی از برنامه‌ی توضیح داده شده در بالا مطابق با دیاگرام نردبانی نشان داده شده است.

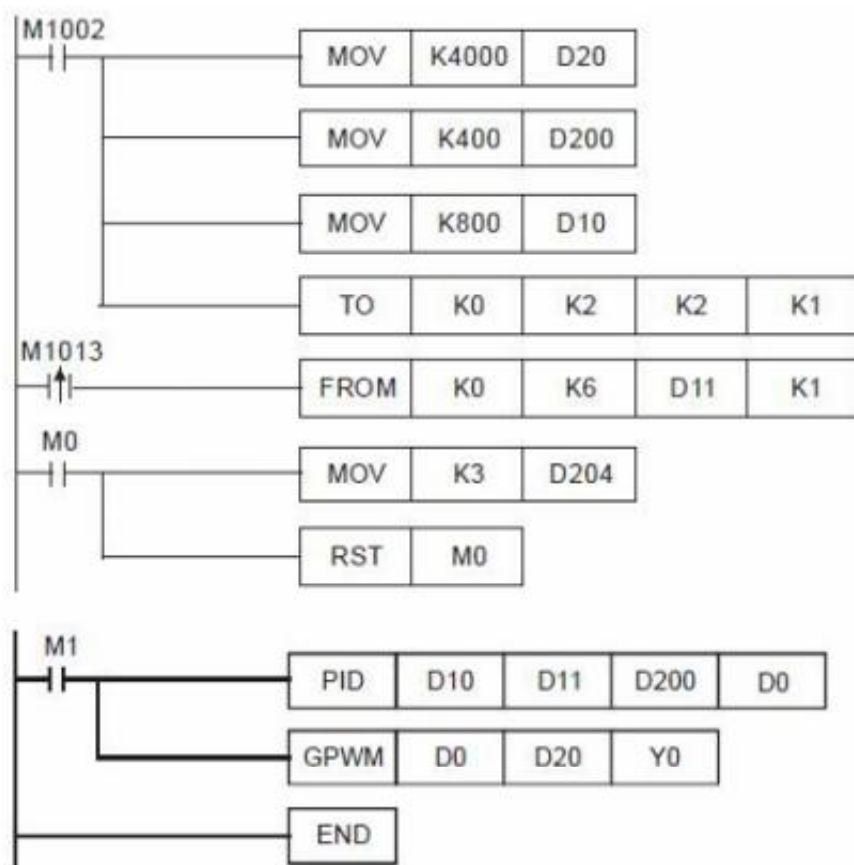


برنامه 3

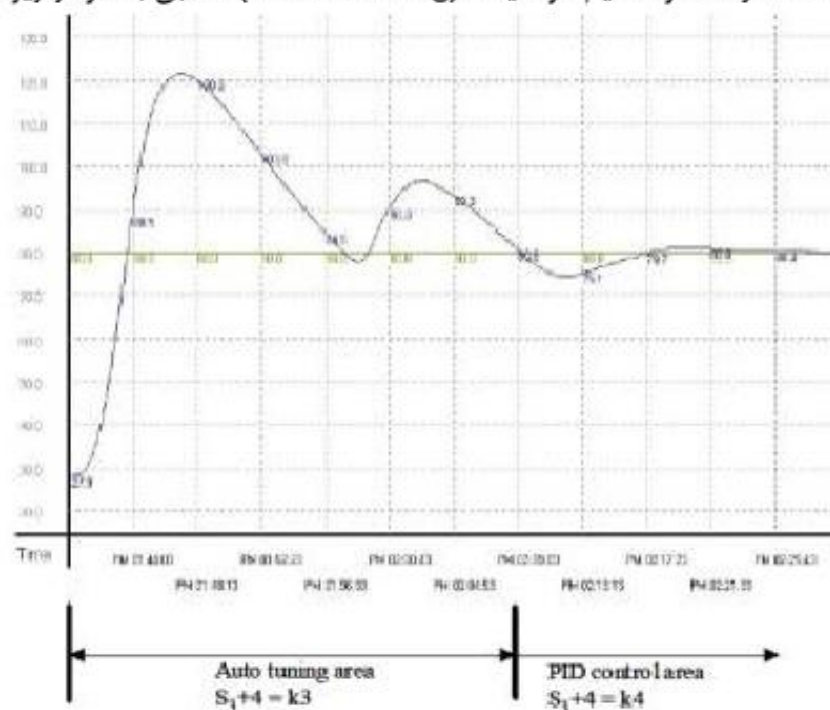
در این برنامه می‌خواهیم با استفاده از مد خودتنظیم اتوماتیک، پارامترهای کنترل دما را برای دستورالعمل PID تنظیم نماییم.

هدف از کنترل در این برنامه، استفاده از مد خودتنظیم اتوماتیک جهت محاسبه‌ی اتوماتیک مقادیر پارامترهای K_p ، K_i و K_d مناسب برای کنترل PID دما می‌باشد.

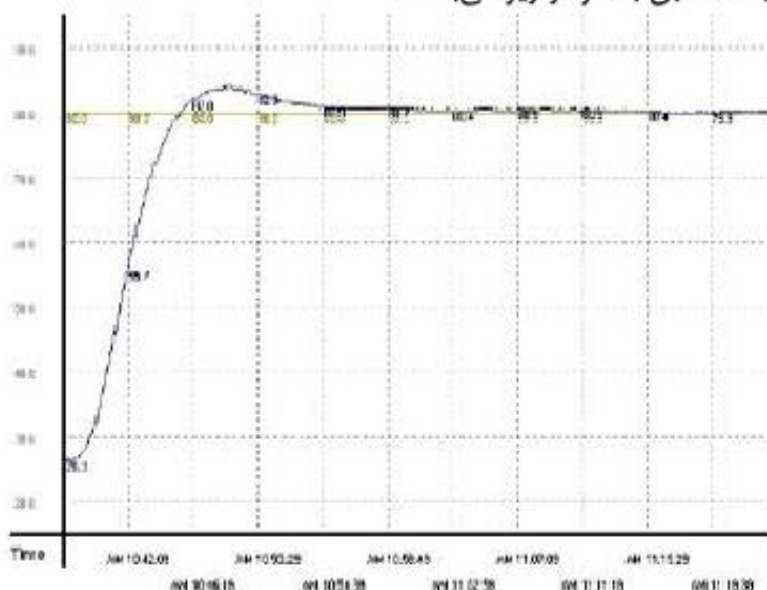
از آنجایی که صنعت با کنترل دما عجین شده است، باید شما با مباحث کنترل دمای یک فرآیند آشنا باشید. همچنین، از آنجایی که شما با دمای محیط برای نخستین بار آشنایی ندارید، می‌توانید از مد خودتنظیم اتوماتیک ($S_3+4=K3$) برای سازگاری پارامترهای PID با محیط استفاده کنید. این به آن معنی است که این مد مقدار پارامترهای K_p ، K_i و K_d را برای عملکرد بهتر کنترل PID دما محاسبه کرده و بدست می‌آورد و بهترین مقادیر را جایگزین می‌کند. بعد از این عمل، دستورالعمل PID به مد کنترل دما ($S_3+4=K4$) خواهد رفت. در این مثال، محیط کنترل دما یک کوره آجر پزی می‌باشد. در شکل زیر، دیاگرام نربانی این برنامه نشان داده شده است.



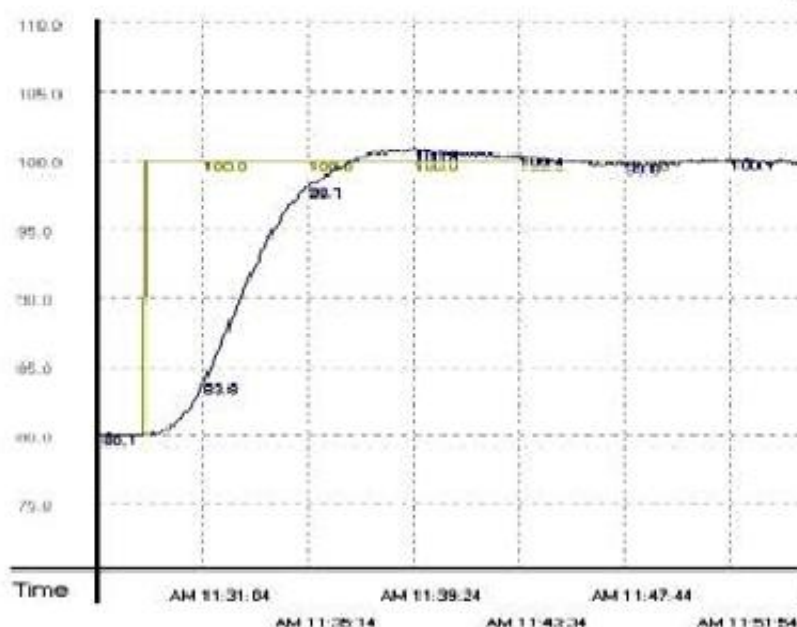
نتیجه‌ی بدست آمده در مد خودتنظیم اتوماتیک (Auto-tuning) مطابق با نمودار زیر می‌باشد:



نتیجه‌ی بدست آمده، با استفاده از تنظیم و تغییر پارامترهای کنترل PID دما برای کنترل دما بعد از مد خودتنظیم اتوماتیک مطابق با نمودار زیر می‌باشد:



در شکل بالا می‌توان مشاهده کرد که کنترل دما بعد از خودتنظیم اتوماتیک به خوبی کار خواهد کرد. از این رو، ما مجاز به استفاده از مد خودتنظیم اتوماتیک برای کنترل دما تا حدود 20 دقیقه می‌باشیم. در غیر این صورت، اگر زمان استفاده از مد خودتنظیم اتوماتیک بیشتر از 20 دقیقه شود، سیستم نوسانی شده و ناپایدار می‌شود. در مرحله بعد، ما مقدار تنظیمی دما را از 80 درجه سانتی‌گراد به 100 درجه سانتی‌گراد اصلاح می‌کنیم (تغییر می‌دهیم). نتیجه بدست آمده در این تغییر دما مطابق با نمودار نشان داده شده در شکل زیر می‌باشد:



همانطور که در شکل بالا مشاهده می‌کنید، وقتی که پارامتر SV (نقطه‌ی مطلوب) از 80 به 100 درجه سانتی‌گراد تغییر دادیم، ما یک کنترل نرم و بدون نوسان دما را بدون صرف کردن زمان طولانی را داشتیم. این موضوع باعث می‌شود که کوره شوک حرارتی نداشته باشد و دما به تدریج بالا رود.

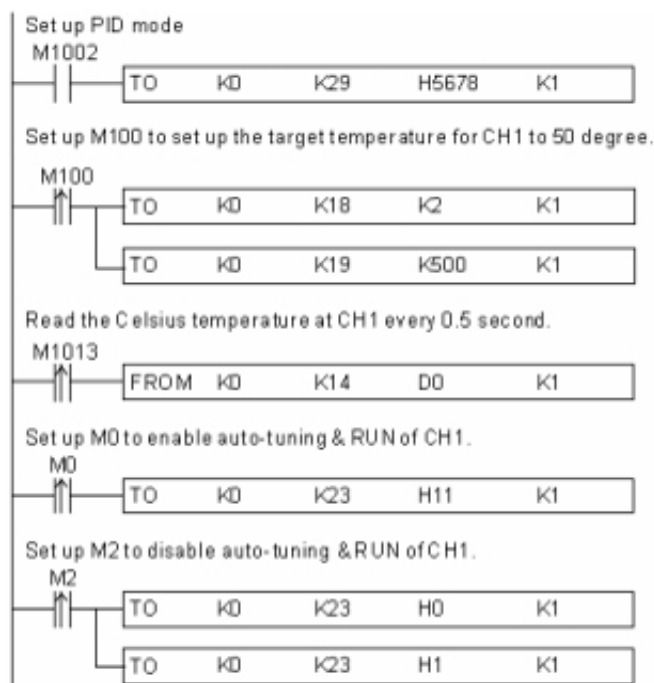
3-5 مثال کاربردی کنترل کننده‌ی PID

مثال 1

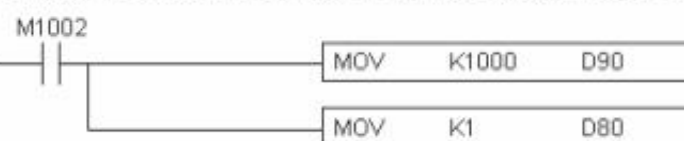
ابتدا می‌خواهیم برنامه کنترل یک فرآیند حرارتی را با استفاده از ماژول DVP04PT-S بررسی نماییم. هدف از کنترل در این مثال به شرح زیر است:

وقتی که فلگ عمومی M100=ON شد، مقدار دمای مطلوب برای کانال CH1 تنظیم شود. وقتی که فلگ عمومی M0=ON شد، تابع خودتنظیم اتوماتیک پارامترهای کنترل کننده‌ی PID برای کانال CH1 فعال شود. منتظر بمانید که عملکرد تابع خودتنظیم اتوماتیک کانال CH1 تکمیل شود (Y10=ON)، و بعد از تکمیل شدن عملکرد این تابع برنامه به صورت اتوماتیک مقادیر پارامترهای K_P , K_I , K_D را استخراج کرده و سپس برای توقف عملکرد تابع خودتنظیم باید فلگ عمومی M0=OFF نماییم.

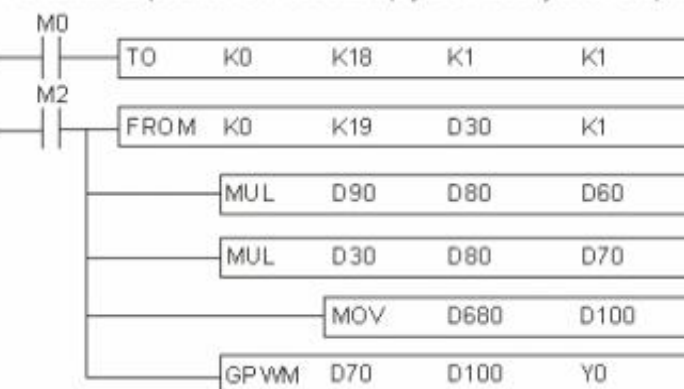
فقط لازم است که یک مرتبه تابع خودتنظیم اتوماتیک ضرایب PID را اجرا نمایید و سپس در دفعات بعد با فعال کردن فلگ M2=ON تابع PID دستی را فعال نمایید تا بتوانید ضرایب این تابع را به صورت دستی تنظیم کنید.



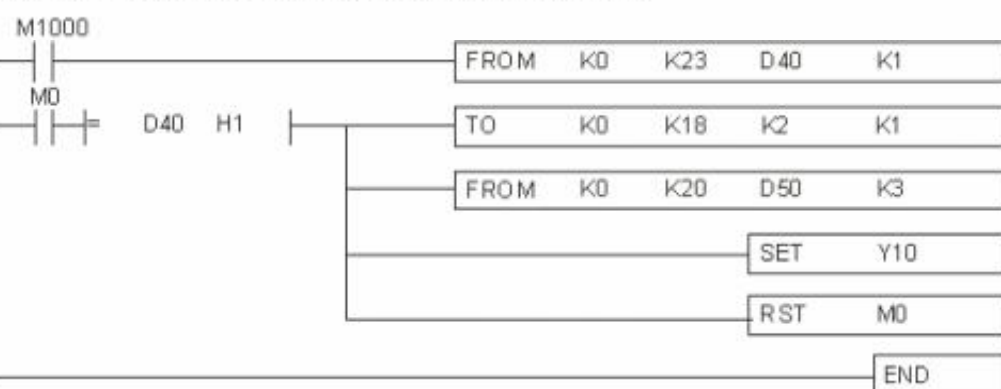
Change the content in D80 if you would like to modify the output cycle for GPWM. (Unit: 1s)



Read the Output % of CH1 and multiply the value by D80. Output the value to Y0.



Determine if the auto-tuning is completed.
Once the auto-tuning is completed, read K_p , K_i , and K_d of CH1.



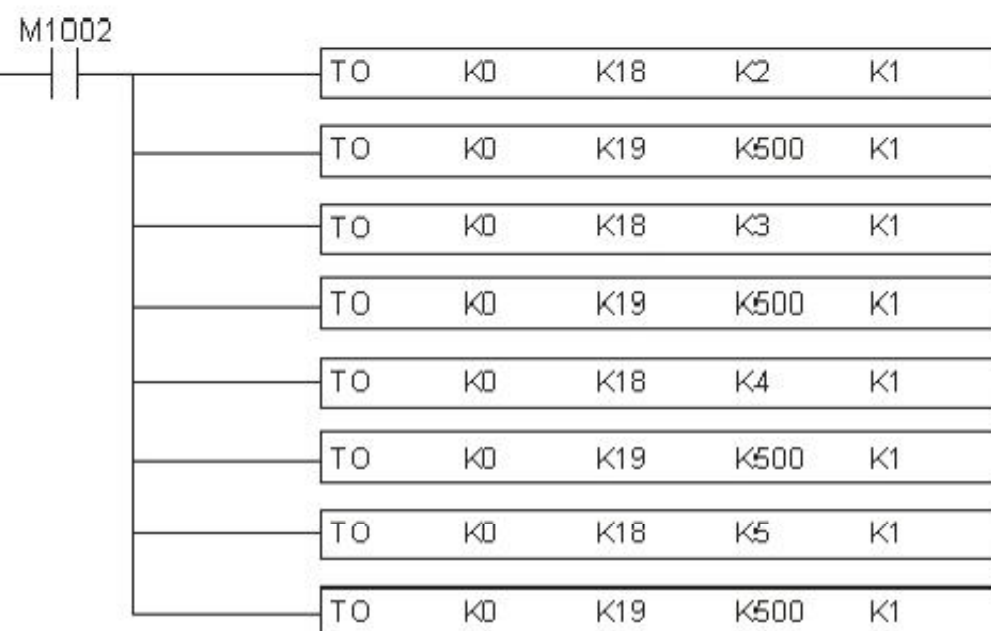
مثال 2

بعد از اینکه PLC را به وضعیت RUN بردید، تابع خودتنظیم اتوماتیک PID برای کانال‌های CH1 ~CH4 فعال خواهد شد.

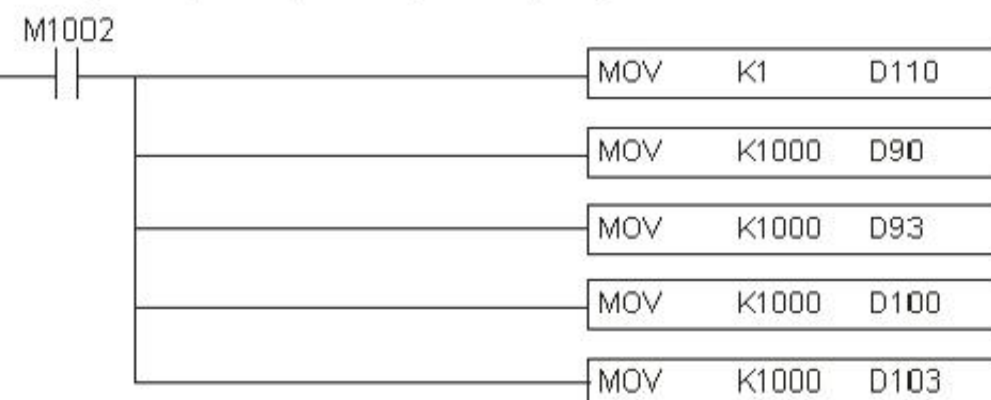
منتظر بمانید که عملکرد تابع خودتنظیم اتوماتیک پارامترهای PID برای کانال‌های CH1~CH4 تکمیل شود (Y10=ON). بعد از اینکه عملکرد این تابع تکمیل شد، برنامه به صورت اتوماتیک مقادیر پارامترهای K_p , K_i , K_d را استخراج خواهد کرد. برای توقف عملکرد تابع خودتنظیم باید فلگ عمومی M0=OFF نمایید.

فقط لازم است که یک مرتبه تابع خودتنظیم اتوماتیک ضرایب PID را اجرا نمایید، و سپس برای تنظیم ضرایب این پارامترها در دفعات بعد باید با فعال کردن فلگ عمومی M1=ON ضرایب PID را به صورت دستی تنظیم کنید.

Set the target temperature of every CH to 50 degree.

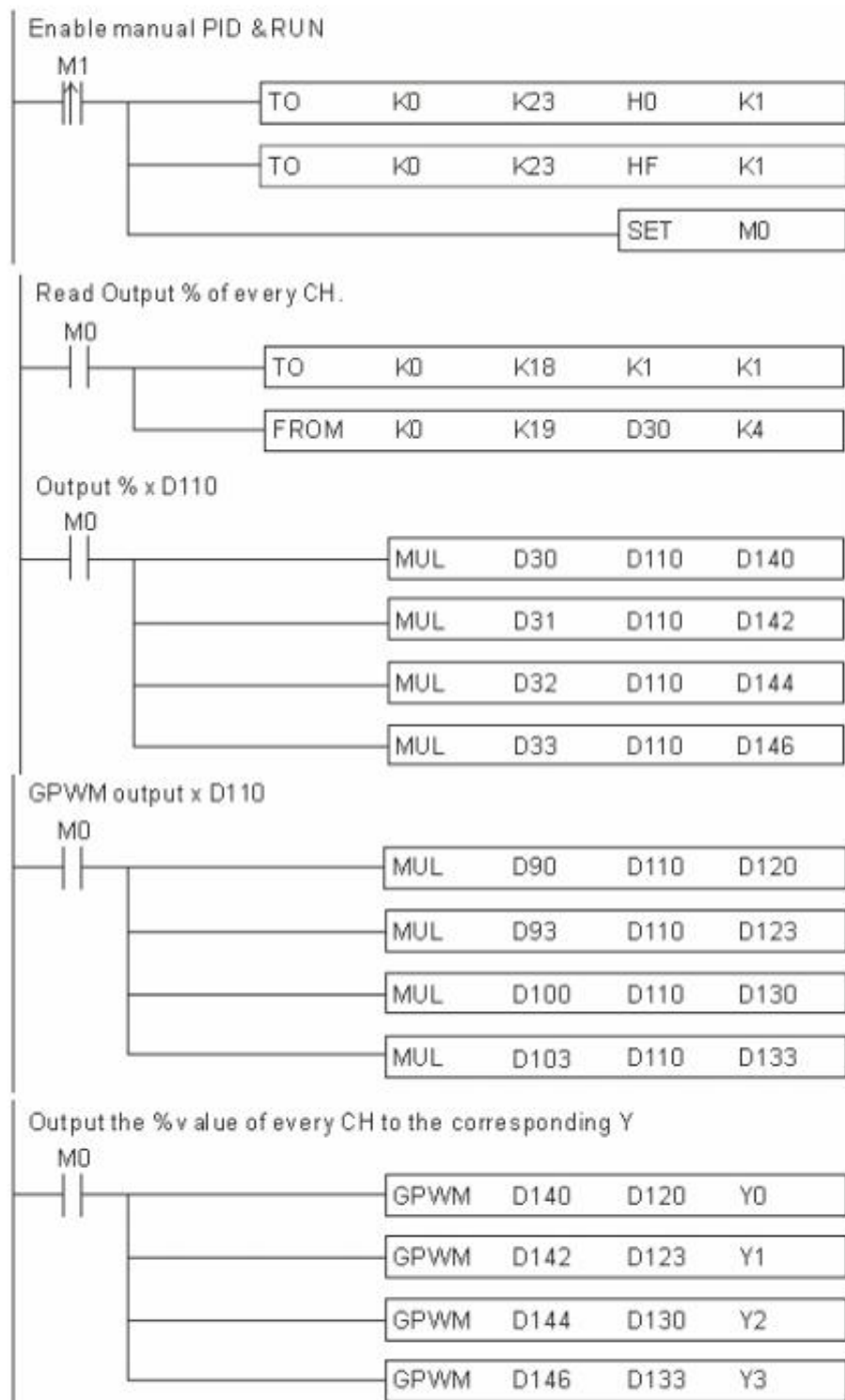


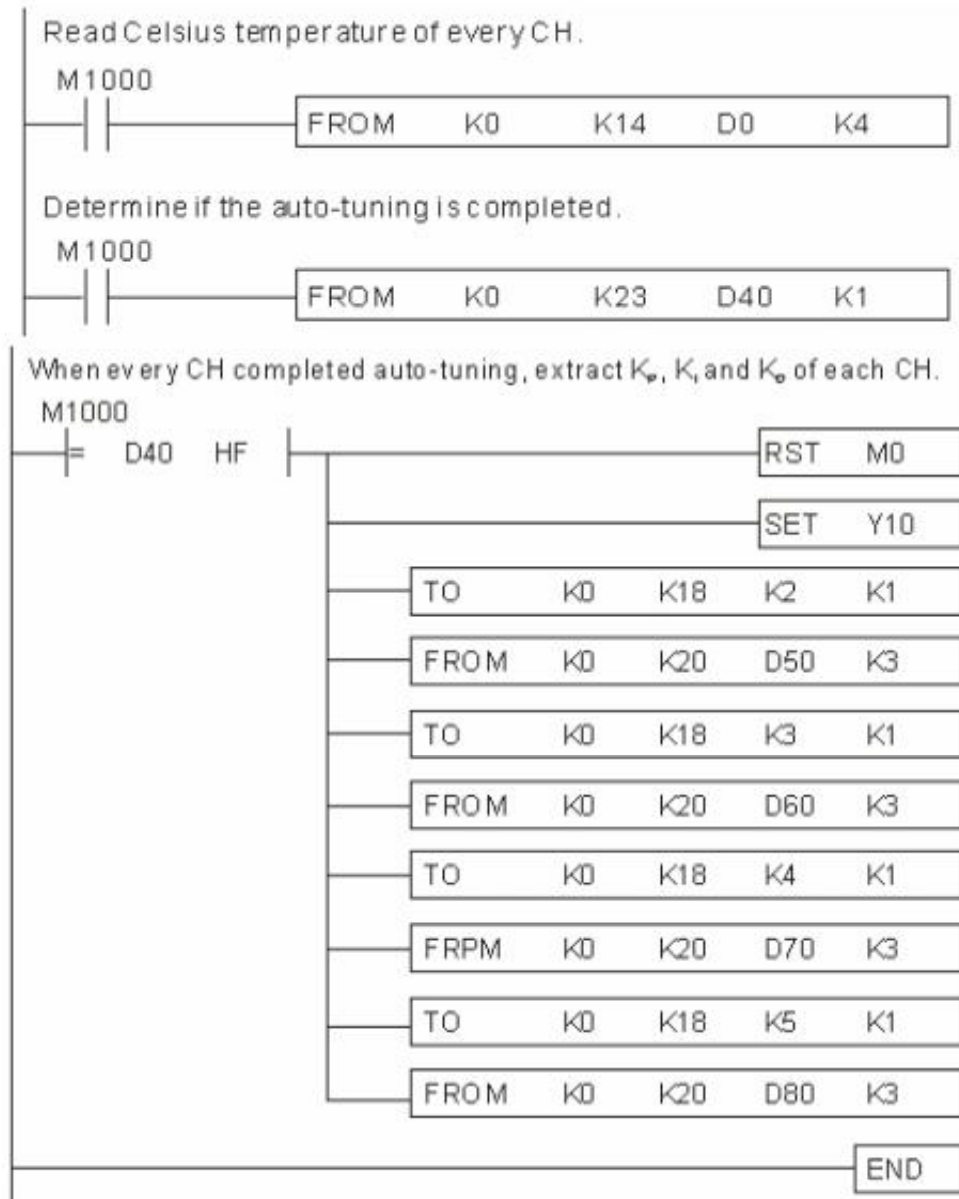
Modify D110 (unit: 1s) to set up the output cycle for GPWM.



Enable auto-tuning & RUN







مثال 3

در اینجا می‌خواهیم یک برنامه‌ی کنترل فرآیند حرارتی را با استفاده از ماژول DVP04PT-H2 بررسی نماییم.

هدف از کنترل در این مثال به شرح زیر است:

مقدار دمای مطلوب را تنظیم نماییم.

تابع خودتنظیم اتوماتیک ضرایب پارامترهای PID را فعال کرده و منتظر بمانیم تا عملکرد این تابع تکمیل شود.

تنها لازم است که فقط یک بار این تابع (خودتنظیم اتوماتیک ضرایب پارامترهای کنترل کننده‌ی PID) را اجرا نماییم، و سپس برای تنظیم مقادیر K_P , K_I , K_D در دفعات بعد باید با فعال کردن حالت دستی کنترل کننده‌ی PID، این ضرایب را به صورت دستی تنظیم کنیم.

Enable M100 to set up target temperature in D500.



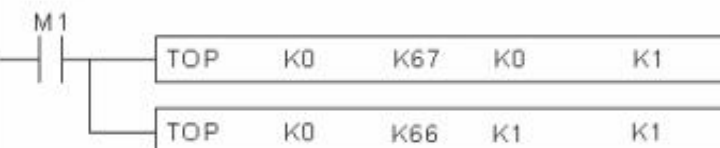
Read average temperature at CH1 every 0.5 second.



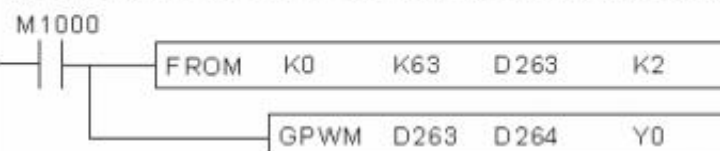
Set up PID auto-tuning and PID RUN at M0.



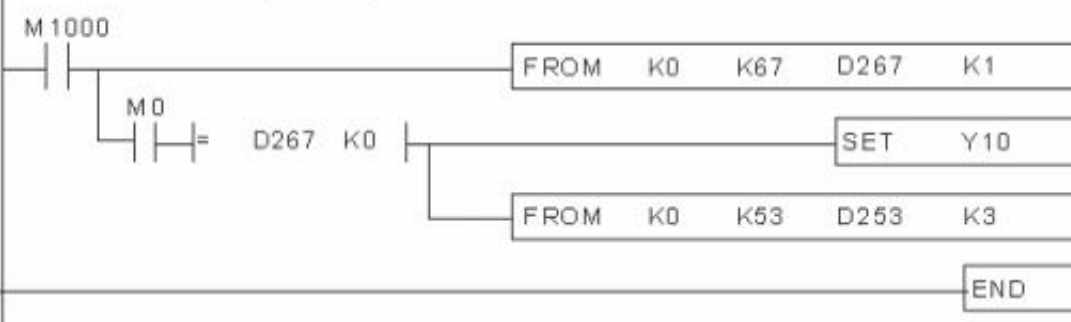
Set up PID manual and PID RUN at M1.



Read output width, output cycle and execute width adjustment instruction.



Auto-tuning CR = K0 indicates that the auto-tuning is completed. Once the auto-tuning is completed, read K_P , K_I and K_D .



« روش کنترل PID

دو روش کنترل خروجی PID وجود دارد. یکی مد کنترل سیکلی بوده و دیگر مد کنترل خروجی آنالوگ می‌باشد.

« مد کنترل سیکلی Cyclic Control Mode

روش کنترل در مد کنترل سیکلی برای ماژول DVP04TC-S به شرح زیر است:

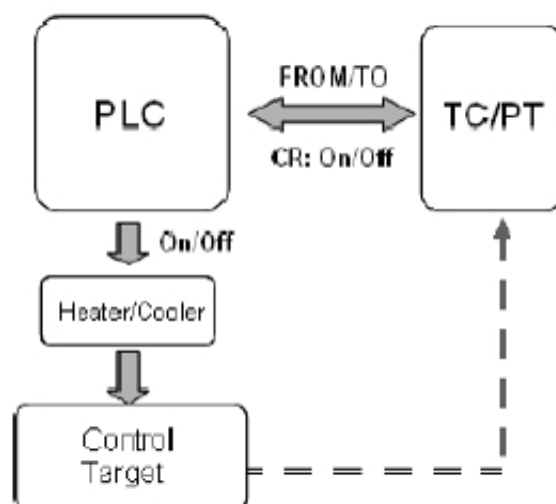
شما می‌توانید سیکل خروجی را مطابق با شرایط محیط کنترلی خود تنظیم نمایید. اگر دما در محیط تحت کنترل خیلی کند تغییر می‌کند، سیکل خروجی را طولانی‌تر تنظیم نمایید. درصد خروجی \times سیکل = پهنای پالس خروجی (Output width = Cycle \times Output %). از پهنای پالس خروجی و سیکل خروجی در دستورالعمل GPWM برای کنترل سیکلی استفاده می‌شود. در اینجا ما هیتر یا فن را به خروجی Y0 متصل کرده‌ایم. توجه داشته باشید که سیم‌بندی این ماژول مانند سیم‌بندی ماژول DVP04TC-H2 می‌باشد.

برای مثال، فرض کنید که می‌خواهیم سیکل خروجی برابر با 3 ثانیه باشد، از این رو، در برنامه کنترلی زیر مقدار D11 را برابر با K3000 تنظیم می‌نماییم. بنابراین $D10 = \text{Output\%} \times K3000/1000$ می‌شود. واحد Output% برابر با 0.1% است. در شکل زیر دیاگرام نردبانی این مثال آورده شده است:

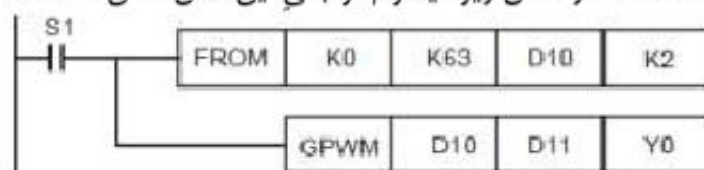


روش کنترل در مد کنترل سیکلی برای ماژول DVP04TC-H2 به شرح زیر است:

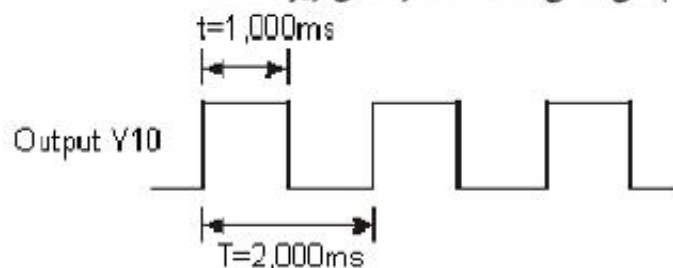
اگر هیتر یا فن استفاده شده در سیستم کنترل دما با قطع و وصل کردن تغذیه ورودی‌شان کنترل می‌شوند، شما باید با استفاده از دستورالعمل GPWM به صورت کنترل سیکلی آنها را کنترل نمایید. برای سیم‌بندی این روش به دیاگرام زیر مراجعه کنید:



شما باید دو رجیستر کنترلی (CR) را از ماژول DVP04TC بخوانید. نخستین رجیستر کنترلی CR سیکل مؤثر و قابل استفاده و دومین رجیستر کنترلی پهنای پالس قابل استفاده را نشان می‌دهد. کنترل سیکلی با دستورالعمل GPWM نوشته شده در DVP-PLC (منظور CPU اصلی می‌باشد) کار می‌کند. برای مثال، سیگنال‌های دریافت شده از ترموکوپل به کانال CH1 ماژول DVP04TC وارد می‌شود. پهنای پالس خروجی CH1 از رجیستر کنترلی CR#63 خوانده شده و سیکل خروجی کانال CH1 از رجیستر CR#64 خوانده می‌شود. شما می‌توانید با استفاده از دستورالعمل FROM محتویات رجیسترهای کنترلی CR#63 و CR#64 را خوانده و توسط پهنای خروجی و مدت زمان سیکل دستورالعمل GPWM کنترل سیکلی را به وجود آورید. در این مثال هیتر یا فن به خروجی دیجیتال Y0 متصل شده است. در شکل زیر دیاگرام نردبانی این مثال نشان داده شده است:



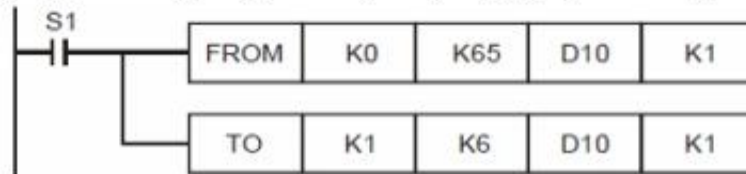
فرض کنید که پهنای پالس خروجی 1000 و زمان یک سیکل برابر با 2000 می‌باشد. پالس‌های خروجی ایجاد شده مانند پالس نشان داده شده در شکل زیر است:



کنترل خروجی آنالوگ

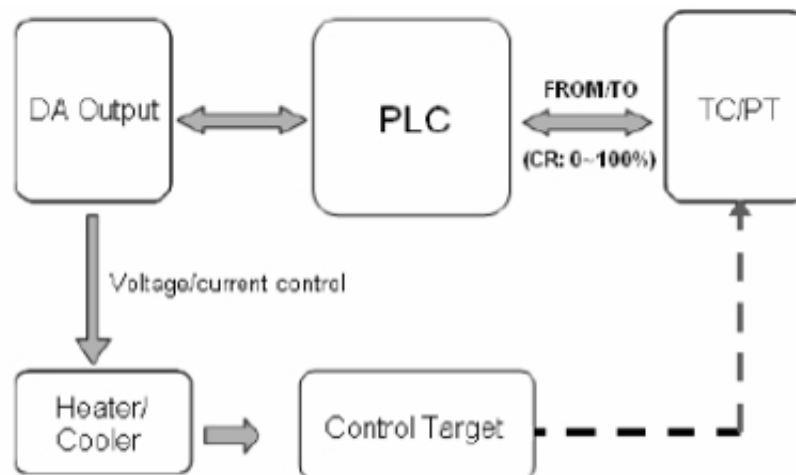
روش کنترل در مد کنترل خروجی آنالوگ برای ماژول DVP04TC-S به شرح زیر است:

در کنترل خروجی آنالوگ ابتدا مقدار درصد خروجی (0~100%) خوانده شده و سپس این مقدار با مقدار خروجی آنالوگ ماژول DVP04DA جهت کنترل یک گرم‌کن (شیر تناسبی یا همان شیر تدریجی گاز مشعل) تطبیق داده می‌شود. برای مثال، اگر مقدار خروجی آنالوگ ماژول DVP04DA-S برابر با مد $0 \sim 4000 (0 \sim 10V)$ تنظیم شده باشد، هیتر (مشعل) متصل شده به ولتاژ خروجی ماژول DVP04DA-S با خواندن مقدار رجیستر کنترلی درصد خروجی Output% کنترل خواهد شد. در اینجا مقدار خروجی طبق معادله‌ی $\text{Output value} = 8000 \times \text{Output} / 1000$ محاسبه شده و واحد خروجی 0.1% می‌باشد. مقدار خروجی آنالوگ توسط دستورالعمل TO کنترل می‌شود.



روش کنترل در مد کنترل خروجی آنالوگ برای ماژول DVP04TC-H2 به شرح زیر است:

اگر سیستم گرمایشی و یا سیستم سرمایشی را توسط ولتاژ و یا جریان کنترل می‌کنید، برای این کار باید از خروجی آنالوگ استفاده نمایید. برای سیم‌بندی سیستم کنترل خروجی آنالوگ به دی‌گرام زیر مراجعه نمایید:



توجه داشته باشید که سیم‌بندی ماژول DVP04TC-S مانند سیم‌بندی ماژول DVP04TC-H2 می‌باشد.

برای کنترل سیستم گرمایشی و یا سرمایشی توسط خروجی آنالوگ، شما باید محدوده‌ی خروجی آنالوگ را تنظیم نمایید. با استفاده از دستورالعمل FROM مقدار خروجی آنالوگ ماژول DVP04TC

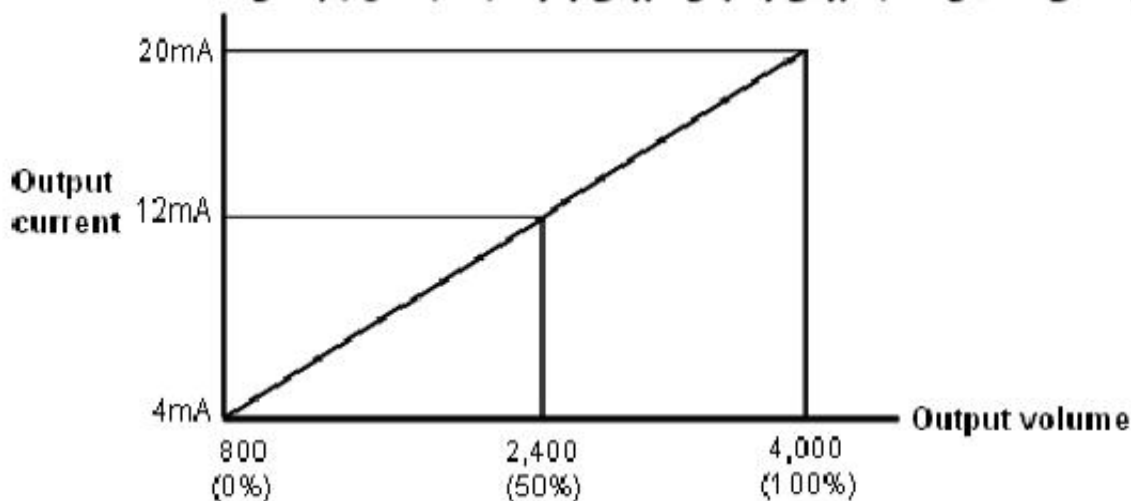
خوانده شده و با استفاده از دستورالعمل TO این مقدار به خروجی آنالوگ ماژول DVP04DA ارسال می‌شود.

مثال

محدوده‌ی خروجی سیستم گرمایشی/ سرمایشی $800 \sim 4000 (4 \sim 20\text{mA})$ می‌باشد. برای کنترل سیستم گرمایشی/ سرمایشی از ماژول خروجی آنالوگ DVP04DA-H2 استفاده شده است. در این ماژول از خروجی آنالوگ جهت کنترل سیستم گرمایشی/ سرمایشی استفاده شده است. از این رو، ما باید محدوده‌ی بالای خروجی آنالوگ را برابر با 4000 تنظیم کرده و محدوده‌ی پایین خروجی آنالوگ را برابر با 800 تنظیم نماییم. همچنین، باید مقدار خروجی دیجیتال از ماژول DVP04TC-H2 را نیز بخوانیم. با استفاده از دستورالعمل FROM مقدار خروجی دیجیتال شده‌ی دما را خوانده و در رجیستر D10 ذخیره می‌نماییم و با استفاده از دستورالعمل TO مقدار خروجی را به ماژول DVP04DA-H2 جهت کنترل شیر آنالوگ مربوط به سیستم گرمایشی/ سرمایشی ارسال می‌کنیم. در شکل زیر، دیاگرام نردبانی این مثال نشان داده شده است:



وابستگی که میان مقدار خروجی و جریان خروجی وجود دارد، در شکل زیر نشان داده شده است:



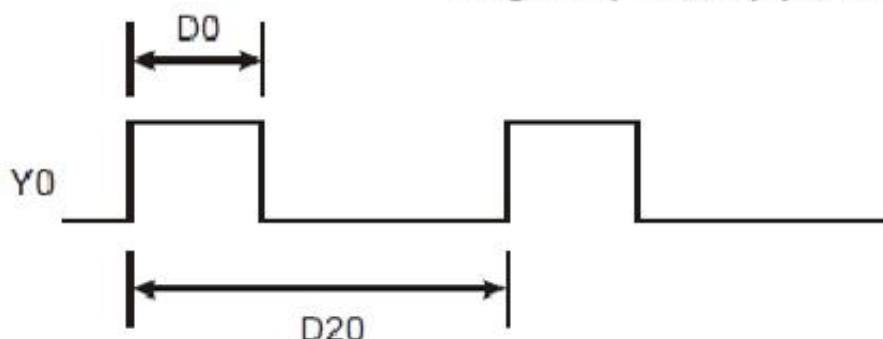
4-5 خودتنظیم اتوماتیک پارامترهای PID کنترلر دما

در این مثال، می‌خواهیم با استفاده از دستورالعمل PID دمای کوره را کنترل نموده و همچنین با استفاده از تابع خودتنظیم اتوماتیک پارامترهای PID را تنظیم نماییم. هدف از کنترل در این مثال به شرح زیر است:

وقتی که دمای محیط کوره ثابت نیست، باید تابع خودتنظیم اتوماتیک دستورالعمل PID را جهت کنترل صحیح دما اجرا کنیم. مقدار دمای مطلوب در این مثال برابر با 80°C می‌باشد.

در این مثال از ماژول DVP04TC-S جهت اندازه‌گیری دمای کوره توسط ترموکوپل تیپ J و انتقال دما به CPU سری DVP12SA2 استفاده شده است. در این برنامه با تنظیم کردن $D204=K3$ تابع خودتنظیم اتوماتیک دستورالعمل PID جهت بهینه تنظیم کردن پارامترهای PID به صورت اتوماتیک اجرا خواهد شد و بعد از اتمام زمان خودتنظیم اتوماتیک، PLC مد کنترل دستورالعمل PID را به صورت اتوماتیک به $D204=K4$ تغییر خواهد داد.

بعد از اجرای دستورالعمل PID و محاسبه‌ی مقادیر و بدست آوردن نتیجه، خروجی بدست آمده از PID به ورودی دستورالعمل GPWM انتقال می‌یابد و دستورالعمل GPWM جهت کنترل هیتر، خروجی Y0 را به صورت خروجی پالس فعال و غیرفعال (متناسب با موج و پهنای پالس تولید شده‌ی PWM) می‌کند. پهنای پالس توسط رجیستر D0 تنظیم می‌شود، از این رو زمان فعال بودن خروجی Y0 وابسته به مقدار موجود در رجیستر D0 می‌باشد.

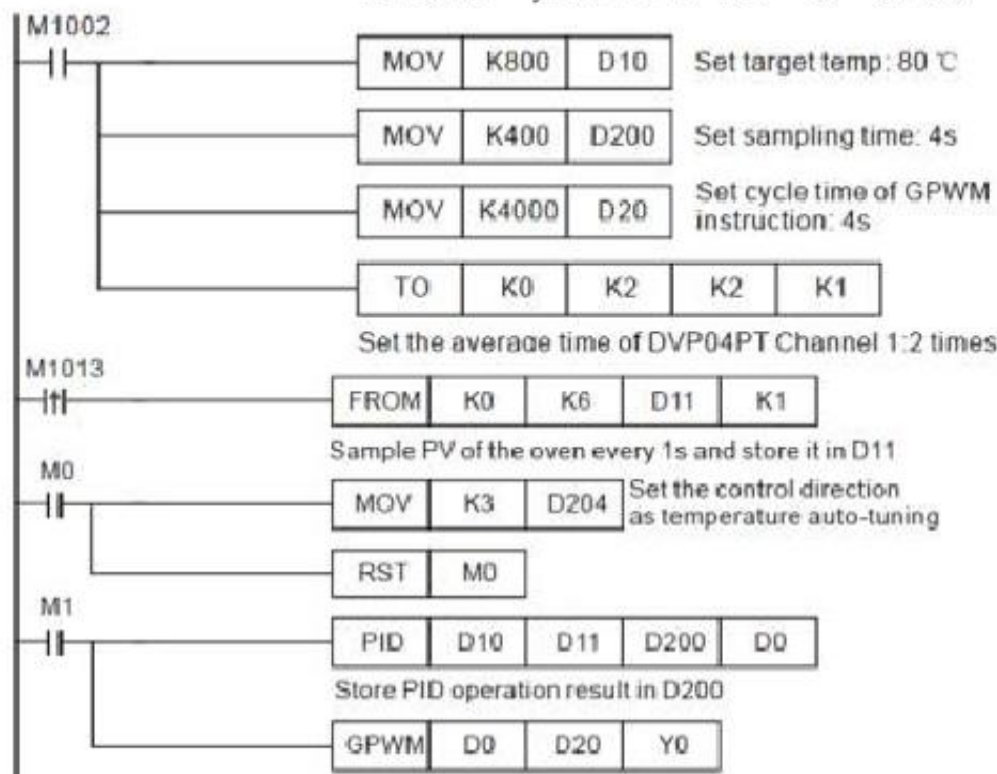


ابزارهای مورد استفاده برای این مثال به شرح زیر می‌باشد:

ابزار	توضیحات
M1	از این فلگ جهت اجرای دستورالعمل PID استفاده می‌شود.
Y0	از این خروجی برای کنترل عملکرد هیتر مطابق با پالس خروجی قابل تنظیم دستورالعمل GPWM استفاده می‌شود.
D0	در این رجیستر نتیجه بدست آمده از اجرای دستورالعمل PID ذخیره می‌شود.
D10	در این رجیستر مقدار دمای هدف (مطلوب، تنظیمی) ذخیره می‌شود.

D11	در این رجیستر دمای واقعی کوره که توسط ترموکوپل تیپ J اندازه‌گیری می‌شود، ذخیره می‌شود.
D20	در این رجیستر سیکل پالس خروجی دستورالعمل GPWM ذخیره می‌شود.
D200	در این رجیستر پارامتر زمان نمونه‌برداری ذخیره می‌شود.

برنامه کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



عملکرد برنامه‌ی کنترلی این مثال به شرح زیر است:

فرمت کلی دستورالعمل PID مطابق با شکل زیر است:

PID	S_1	S_2	S_3	D
-----	-------	-------	-------	---

عملوند S_1 ← مقدار مطلوب (هدف، تنظیمی) دما SV

عملوند S_2 ← مقدار واقعی دما PV

عملوند S_3 ← پارامترهای دستورالعمل PID می‌باشد که این پارامترها را باید شما تنظیم نمایید.

عملوند D ← مقدار خروجی MV. برای این عملوند نباید از ناحیه رجیستر داده با تابع قفل شونده

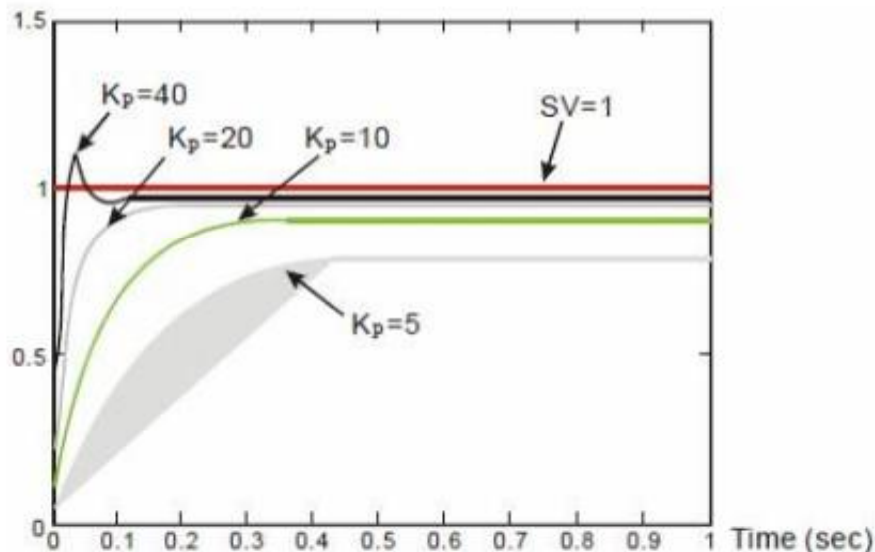
استفاده نمایید.

برای کار با PID باید چندین نکته را رعایت نمایید تا بتوانید تابع کنترلی مناسب را انتخاب نمایید. در این مثال، از پارامتر خودتنظیم اتوماتیک فقط برای کنترل دما استفاده شده و شما نمی‌توانید از آن برای

کنترل سرعت موتور و یا فرآیندهای کنترلی مختلف دیگر استفاده نمایید، زیرا این امر باعث اختلال در عملکرد سیستم کنترلی شما خواهد شد.

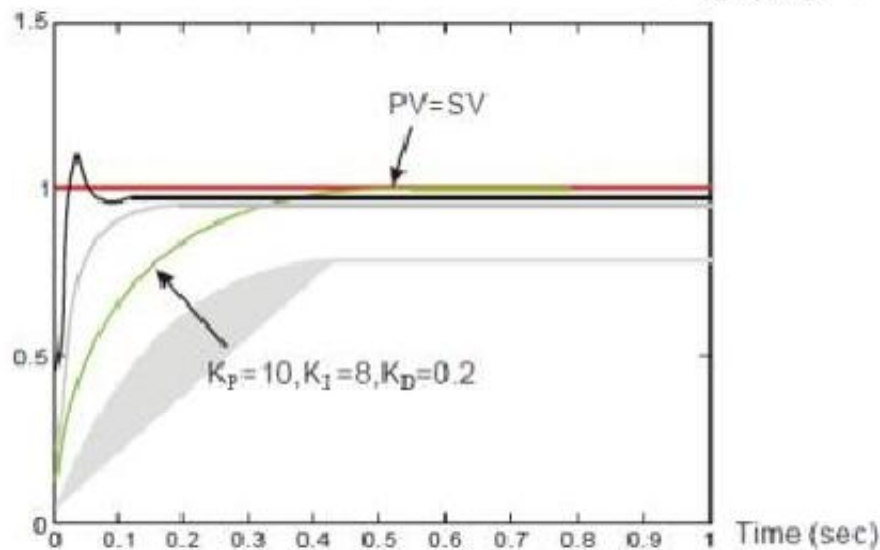
معمولاً تنظیمات پارامترهای کنترلی دستورالعمل PID به صورت تجربی و با آزمایش مکرر سیستم (سعی و خطا) بدست می‌آید. بجز تابع خودتنظیم اتوماتیک در محیط‌های وابسته به کنترل دما، تنظیمات پارامترها در تمام سیستم‌ها مشترک بوده و مطابق با مراحل گفته شده در زیر می‌باشد.

مرحله 1: در این مرحله ابتدا مقادیر K_I و K_D را برابر با صفر و سپس مقدار پارامتر K_P را در چندین مرحله برابر با 5, 10, 20, 40 تنظیم کردیم. مقادیر PV و SV را به ترتیب ثبت کردیم و نتیجه‌ی بدست آمده از این تنظیمات را در یک منحنی آوردیم. نتیجه‌ی بدست آمده مطابق با شکل زیر می‌باشد:



مرحله 2: همانطور که در شکل بالا ملاحظه می‌کنید، در خواهید یافت که وقتی که $K_P = 40$ است، عکس‌العمل سیستم خیلی سریع بوده و دمای واقعی کوره سریع به دمای هدف رسیده است و حتی فرآجهش نیز داشته است. بنابراین، این عملکرد برای کوره به دلیل شک حرارتی مناسب نمی‌باشد. وقتی که $K_P = 20$ است، منحنی عکس‌العمل PV نزدیک به خط ثابت SV می‌باشد و همچنین، در این مقدار فرآجهش نیز وجود ندارد. اما به دلیل راه‌اندازی سریع با مقدار خروجی MV بزرگ و همچنین، شک حرارتی نسبتاً زیاد عملکرد مناسبی برای کوره محسوب نمی‌شود. وقتی که $K_P = 10$ است، منحنی عکس‌العمل PV با یک شیب ملایم و نرم و در مدت زمان بیشتر به مقدار SV رسیده و همچنین فرآجهش نیز در این مقدار وجود ندارد. از این رو، در کوره، این منحنی عملکرد خروجی و کنترل دما مناسب می‌باشد. وقتی که $K_P = 5$ است، به علت کند بودن عکس‌العمل PV دما در مدت زمان بسیار بسیار طولانی به مقدار SV می‌رسد که این سیستم کند بوده و مناسب نمی‌باشد.

مرحله‌ی 3: حال در این مرحله مقدار K_P را برابر با 10 و مقدار K_I را از کم به زیاد (بعنوان مثال، 1, 2, 4, 8) تنظیم می‌کنیم. دقت داشته باشید که مقدار K_I را نباید بزرگتر از K_P تنظیم نمایید. در نهایت، مقدار K_D را از کم به زیاد (بعنوان مثال، 0.01, 0.05, 0.1, 0.2) تنظیم می‌نماییم. توجه داشته باشید که مقدار K_D نباید بیشتر از 10% K_P باشد. با اعمال این تنظیمات عملکرد سیستم را مشاهده کرده و ثبت می‌نماییم. نتایج بدست آمده از تنظیمات اعمال شده برای SV و PV مطابق با منحنی زیر می‌باشد.

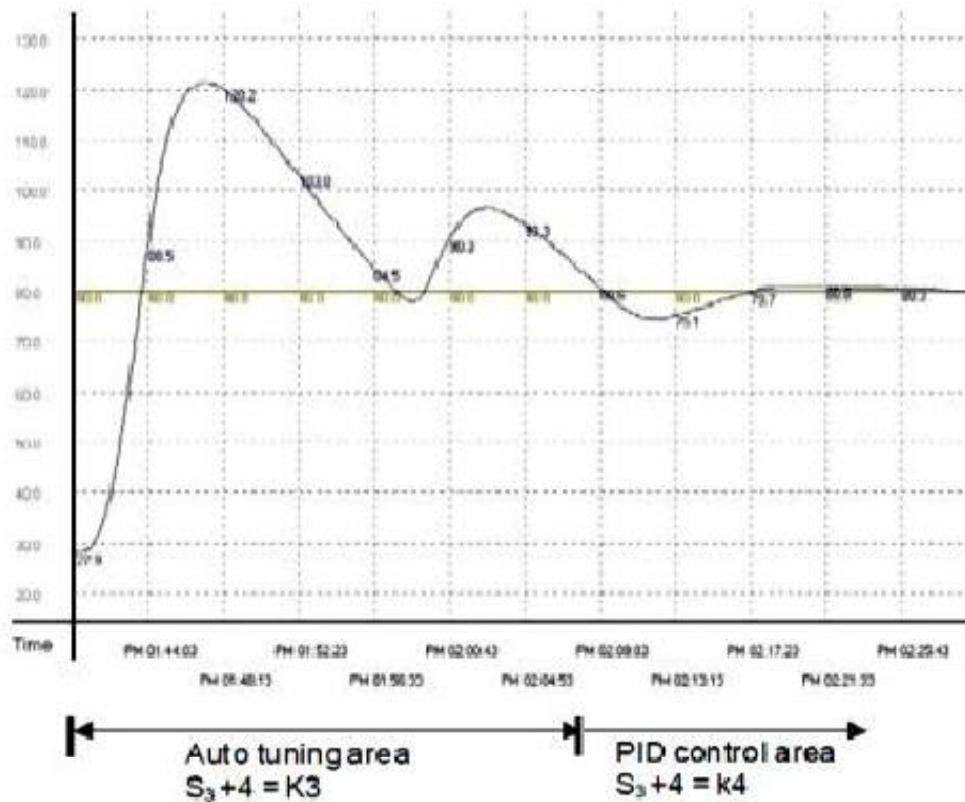


توجه:

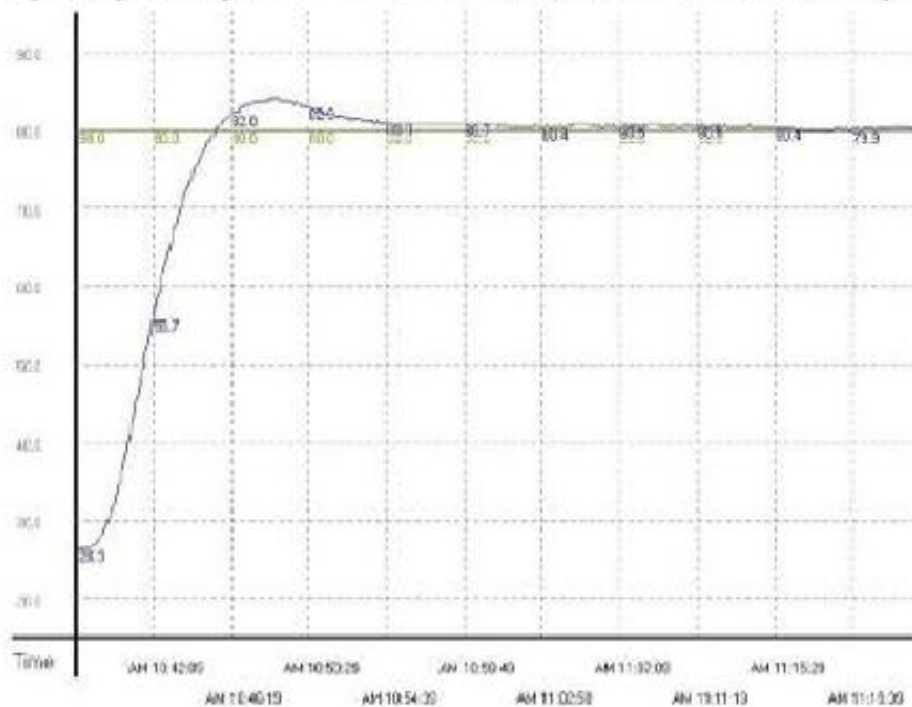
هدف از این مثال، فقط آشنایی شما با عملکرد دستورالعمل PID و تنظیمات پارامترهای آن می‌باشد. بنابراین، در سیستم‌ها و کوره‌های مختلف این تنظیمات متفاوت می‌باشد و نباید از این تنظیمات استفاده نمایید. از این رو، پارامترهای PID را باید مطابق با شرایط واقعی سیستم کنترلی تحت کنترل خود تنظیم کنید.

برای کنترل دمای محیط (از قبیل کوره)، در PLC دلتا تابع خودتنظیم اتوماتیک برای تنظیمات پارامترهای دستورالعمل PID ارائه شده است. بنابراین، شما می‌توانید بدون تنظیمات مرحله‌ای (مراحل توضیح داده شده در بالا) پارامترهای کنترلی خوبی را بر روی سیستم داشته باشید. در زیر، فرآیند خودتنظیم اتوماتیک پارامترها برای این مثال توضیح داده شده است.

تنظیمات مقدماتی: با به کار انداختن سیستم بعد از تنظیم مناسب پارامترهای دستورالعمل کنترل دمای PID و ذخیره کردن نتیجه‌ی بدست آمده در رجیسترهای D200~D219، منحنی عکس‌العمل PV به صورت زیر خواهد بود.



عملکرد سیستم بعد از بدست آمدن مقادیر پارامترهای PID توسط تابع خودتنظیم اتوماتیک برای کنترل دما و ذخیره شدن مقادیر بدست آمده در رجیسترهای D200~D219 مطابق با منحنی زیر می‌باشد.



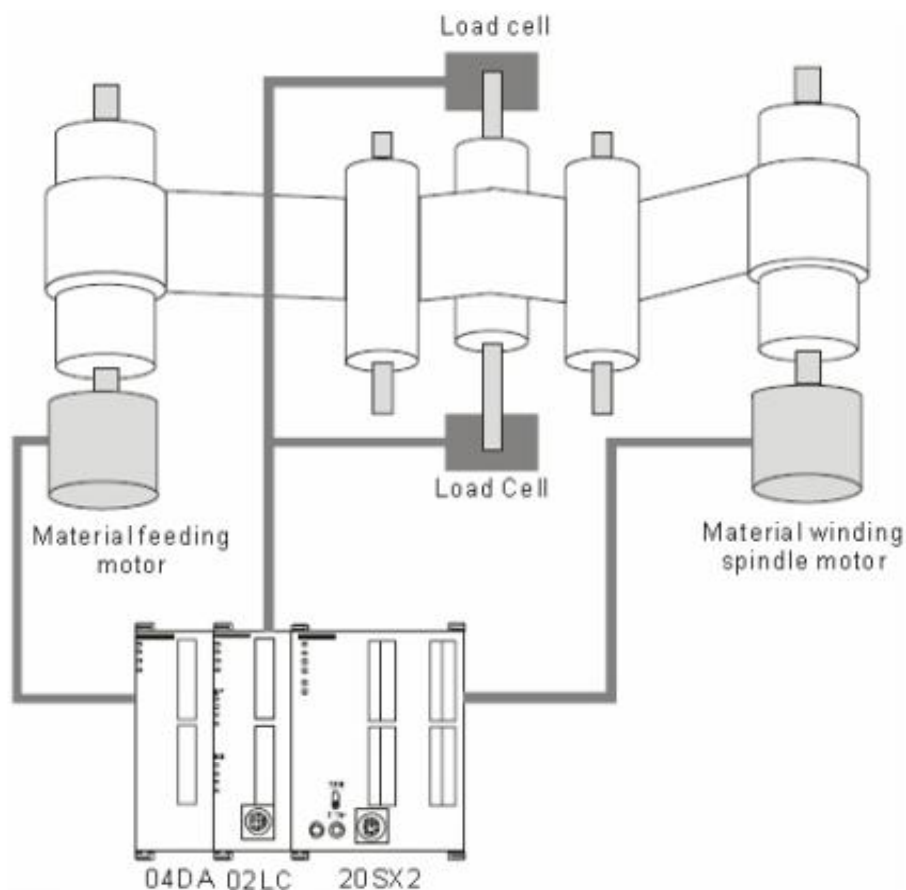
همانطور که در دو شکل بالا ملاحظه می‌کنید، کنترل دما بعد از اجرای تابع خودتنظیم اتوماتیک دارای عملکرد بسیار خوبی می‌باشد. در سیستم کنترل دما، تقریباً حدود 20 دقیقه برای کنترل تابع خودتنظیم اتوماتیک پارامترها در نظر گرفته می‌شود.

زمان نمونه‌برداری دستورالعمل PID باید برابر با زمان سیکل عملکرد دستورالعمل GPWM تنظیم شود، اما واحد دستورالعمل PID برحسب 10ms و واحد دستورالعمل GPWM برحسب 1ms است. بنابراین، نیاز است که این واحدها تبدیل شوند.

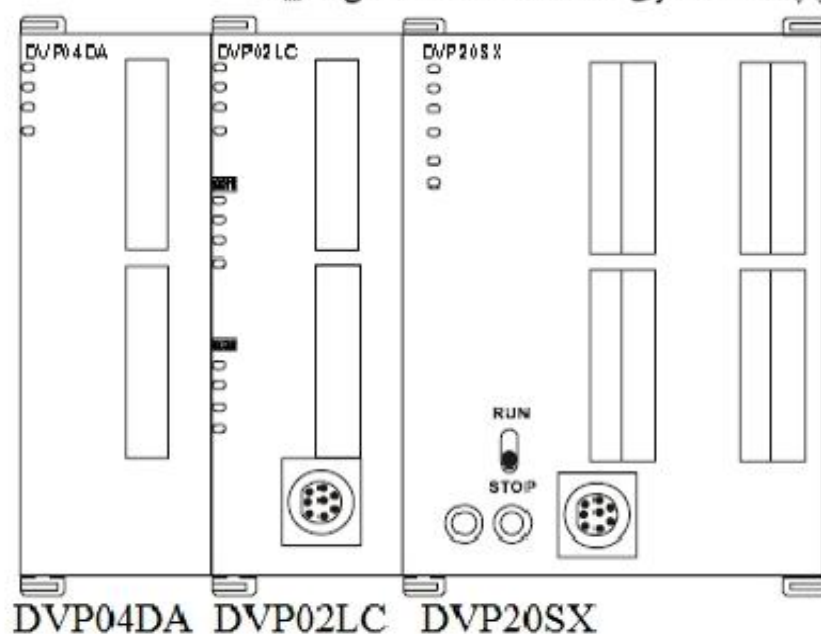
زمان نمونه‌برداری از ترموکوپل یا همان دمای واقعی PV باید دو برابر زمان نمونه‌برداری PID تنظیم شود، معمولاً این زمان بین 2s~6s تنظیم می‌شود.

5-5 کنترل کشش جسمی با استفاده از لودسل توسط دستورالعمل PID

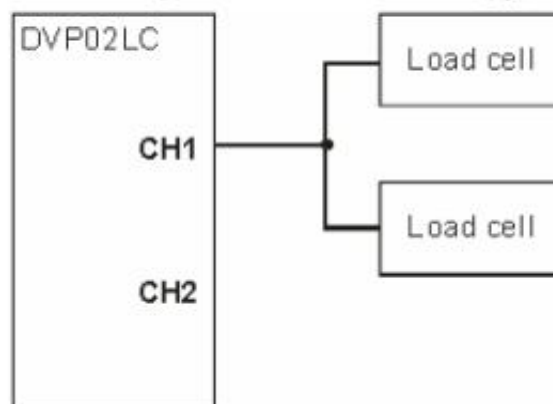
در این مثال، می‌خواهیم با استفاده از ماژول DVP02LC-SL یک برنامه کنترل گشتاور را اجرا نماییم. CPU مورد استفاده در این مثال DVP-SX2 سری (DVP20SX2) PLC می‌باشد. همچنین، از تابع PID در این مثال نیز استفاده شده است. از ماژول DVP02LC-SL جهت آشکارسازی فشار اعمال شده به لودسل استفاده شده می‌شود. از مقدار اندازه‌گیری شده توسط لودسل جهت کنترل سرعت موتور و ترمز سیستم کشش استفاده می‌شود. در این مثال، از ماژول (DVP04DA-SL) خروجی آنالوگ که به سمت چپ CPU متصل می‌شود، استفاده کرده‌ایم. مقدار خروجی آنالوگ توسط دستورالعمل PID جهت کنترل کشش، کنترل خواهد شد. در شکل زیر، سیستم کنترل کشش (گشتاور) نمایش داده شده است.



جهت اجرای سیم‌بندی سخت‌افزاری، ابتدا باید دو ماژول DVP04DA-SL و DVP02LC- را به سمت چپ CPU سری DVP20SX2 متصل نمایید.



در این مثال از دو لودسل 4 سیمه استفاده کرده‌ایم. این دو لودسل را باید با یکدیگر موازی کرده و سپس آنها را به کانال CH1 ماژول DVP02LC-SL متصل نمایید.

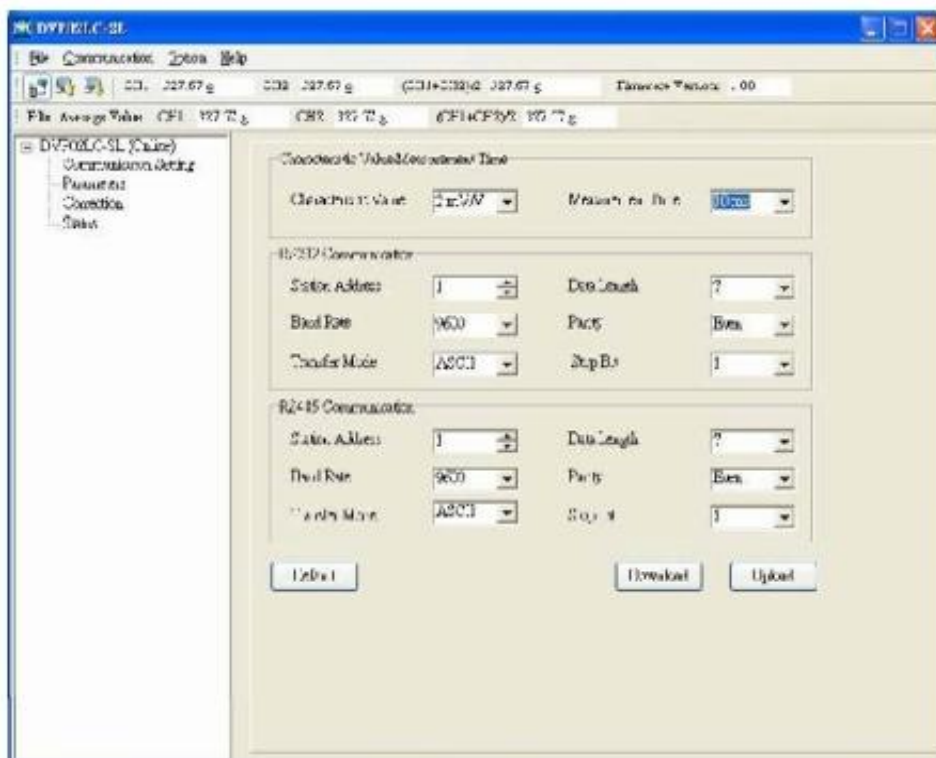


تنظیمات پارامترهای ماژول DVP02LC-SL به شرح زیر می‌باشد:

پارامتر	مقدار تنظیمی	توضیحات
مقدار مشخصه (Eigenvalue)	2(mV/V)	مشخصات مقدار مشخصه لودسل به کار رفته می‌باشد 2mV/V
زمان اندازه‌گیری (Measuring time)	10ms	
تعداد دفعات میانگین‌گیری (CH1 average times)	50	
حداکثر وزن قابل اندازه‌گیری (CH1 max. weight)	32767	حداکثر وزن برابر با 32767 تنظیم شده است.
انتخاب وزن خالص یا ناخالص (CH2 net or gross weight)	Disable	تابع کانال CH2 غیرفعال می‌باشد.

برای تصحیح وزن توسط نرم‌افزار باید مراحل زیر را طی نمایید:

1. مقدار مشخصه‌ی (eigenvalue) لودسل را باید برابر با 2mV/V و زمان اندازه‌گیری را برابر با 10ms تنظیم نمایید.

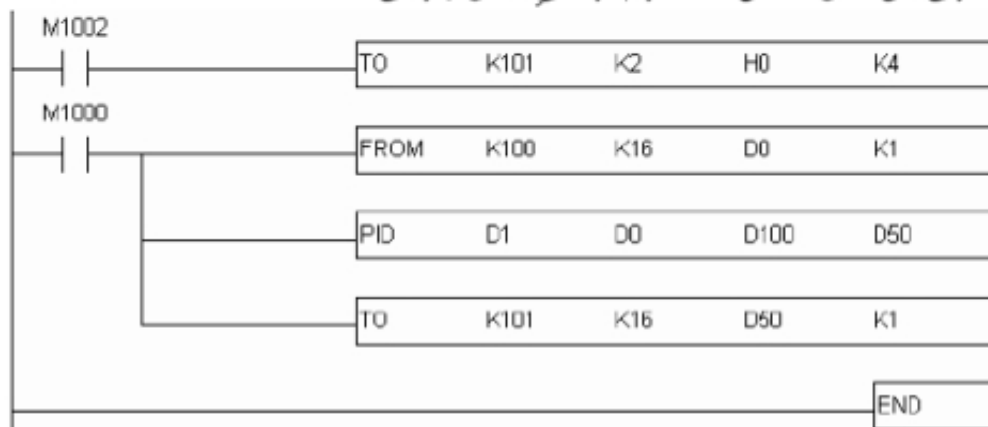


2. در این مرحله باید تعداد دفعات میانگین‌گیری و حداکثر وزن را تنظیم نمایید. به این دلیل که دو لودسل به صورت موازی به یکدیگر متصل شده‌اند و سپس آنها را به کانال CH1 مازول متصل کرده‌ایم و از کانال CH2 استفاده نشده است، باید کانال CH2 را غیرفعال کرد.
3. از یک وزنه استاندارد واقعی برای تصحیح وزن استفاده نمایید.



ما از CPU سری DVP20SX2 برای خواندن مقدار میانگین گرفته شده‌ی وزن از ماژول DVP02LC-SL جهت کنترل عملیات تابع PID و استفاده از خروجی PID برای کنترل ولتاژ خروجی ماژول DVP04DA-SL جهت کنترل و راه‌اندازی اینورتر متصل شده به موتور تغذیه مواد (سیم، نخ، پارچه، کاغذ و ...) استفاده کرده‌ایم.

برنامه کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



ابزارهای مورد استفاده در این مثال به شرح زیر می‌باشند:

D0: از این ابزار برای مشاهده‌ی مقدار میانگین کشش استفاده می‌شود.

D1: از این ابزار برای تنظیم مقدار مطلوب میانگین کشش استفاده می‌شود.

D50: از این رجیستر برای کنترل ولتاژ خروجی ماژول DVP04DA-SL استفاده می‌شود.

D100: از این رجیستر برای تنظیم نقطه‌ی مطلوب دستورالعمل PID استفاده می‌شود.

عملکرد برنامه به شرح زیر است:

بعد از اینکه PLC از وضعیت STOP به وضعیت RUN رفت، برنامه شروع به کار می‌کند، به علت محدوده‌ی 0~10V سیگنال کنترل سرعت موتور توسط اینورتر، باید مد کاری ولتاژ خروجی آنالوگ ماژول DVP04DA-SL را برابر با مد 0 یعنی سیگنال خروجی 10V~ -10V تنظیم نماییم. با استفاده از دستورالعمل FROM می‌توان مقدار میانگین گرفته شده‌ی وزن توسط لودسل را از ماژول DVP02LC-SL خواند.

مقدار خروجی (MV) توسط دستورالعمل PID محاسبه شده و این مقدار خروجی به ماژول DVP04DA-SL ارسال خواهد شد.

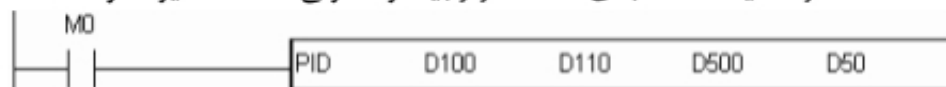
مراحل خودتنظیم اتوماتیک PID به شرح زیر می‌باشد:

1. ابتدا مقدار میانگین گرفته شده‌ی وزن از ماژول DVP02LC-SL خوانده شده و در رجیستر D110 ذخیره خواهد شد.

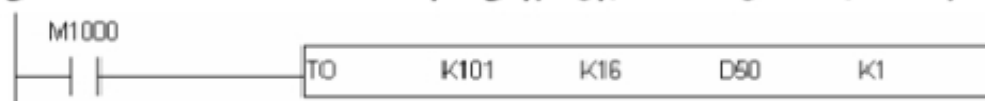


2. برای عملکرد PID باید مقدار رجیسترهای کنترلی PV=0110, SV=D100, PID parameter=D500 را تنظیم نمایید.

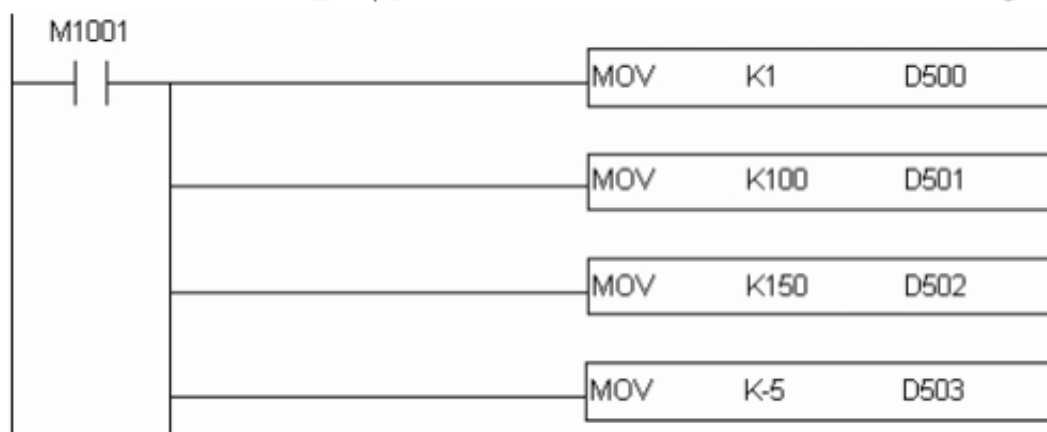
نتیجه‌ی بدست آمده از عملیات محاسبه‌ی PID در رجیستر کنترلی D50 ذخیره خواهد شد.



3. رجیستر D50 را به کانال CH1 ماژول خروجی آنالوگ یا DVP04DA-SL لینک می‌کنیم.



4. زمان نمونه‌برداری PID را برابر با 10ms تنظیم می‌نماییم. پارامترهای PID را باید در رجیسترهای KP=D501, KI=D502, KD=D503 تنظیم نمایید.



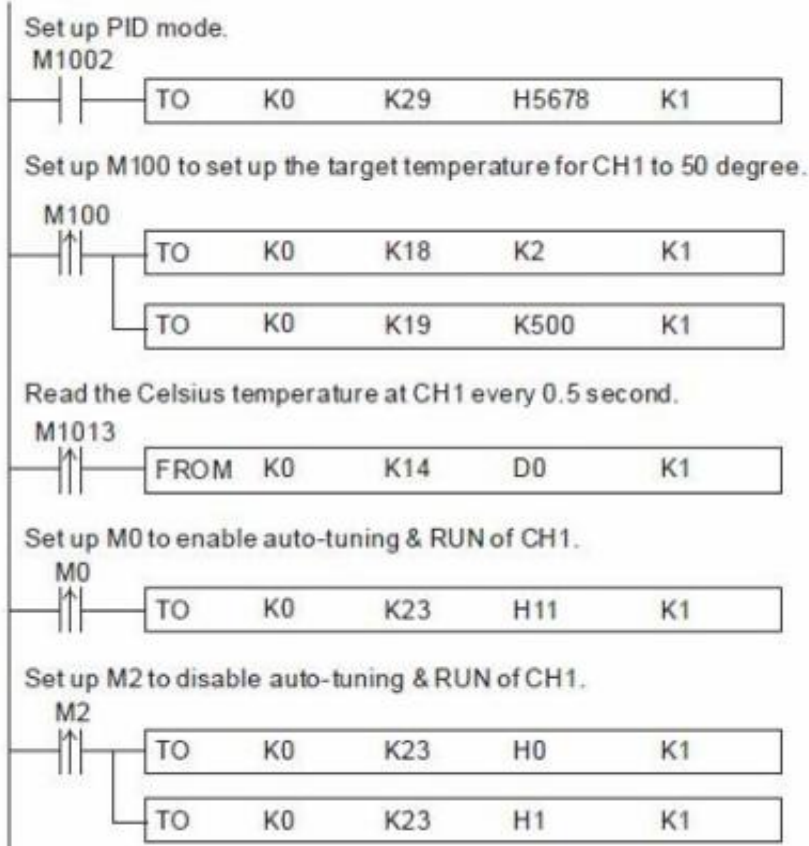
5. بعد از اینکه خودتنظیم اتوماتیک به پایان رسید، بهترین پارامترهای بدست آمده برای ما در این مثال برابر با KP=100, KI=150, KP=-5 بود.

6-5 مثال کاربردی PID توسط ماژول DVP04TC-S

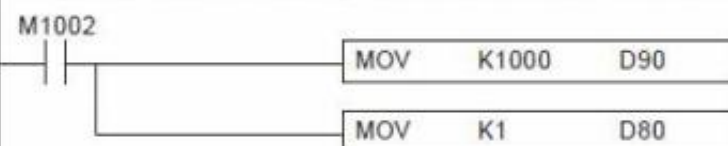
با فعال شدن فلگ M100 (M100=ON) دمای هدف (مطلوب) برای کانال CH1 تنظیم می‌شود.

با فعال شدن فلگ M0=ON تابع خودتنظیم اتوماتیک برای کانال CH1 و تابع PID فعال خواهد شد.

باید برای تکمیل شدن عمل خودتنظیم اتوماتیک کانال CH1 (Y10=ON) کمی منتظر بمانید. در زمان خودتنظیمی اتوماتیک برنامه به صورت اتوماتیک مقادیر صحیح K_D , K_P , K_I را استخراج می‌کند. برای متوقف کردن تابع خودتنظیم اتوماتیک باید فلگ M0=OFF شود. برای بهینه عمل کردن تابع PID فقط یکبار تابع خودتنظیم اتوماتیک را اجرا کنید و منتظر بمانید تا عملکرد این تابع تکمیل شود. برای دفعات بعد فقط لازم است که مد دستی PID را با فعال کردن فلگ M2=ON فعال نمایید.



Change the content in D80 if you would like to modify the output cycle for GPWM. (Unit: 1s)

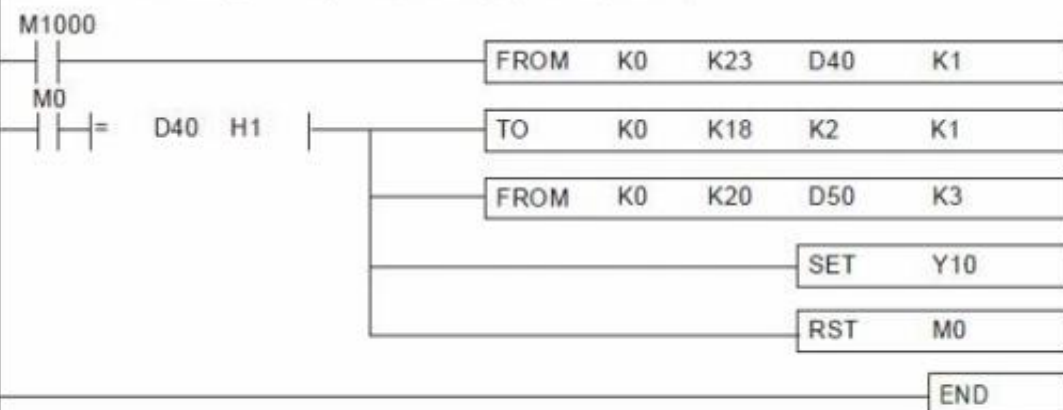


Read the Output % of CH1 and multiply the value by D80. Output the value to Y0.



Determine if the auto-tuning is completed.

Once the auto-tuning is completed, read K_u , K_i and K_d of CH1



با RUN شدن PLC تابع خودتنظیم اتوماتیک کانال‌های CH1~CH4 فعال خواهند شد. منتظر بمانید که عملکرد تابع خودتنظیم اتوماتیک برای کانال‌های CH1~CH4 با فعال شدن خروجی Y10=ON تکمیل شود. در این حالت، برنامه به صورت اتوماتیک ضرایب K_D , K_P , K_I را از کانال‌های CH1~CH4 استخراج می‌کند. برای متوقف کردن عملکرد تابع خودتنظیم اتوماتیک باید فلگ M0=OFF شود.

برای بهینه عمل کردن تابع خودتنظیم PID، فقط باید تابع خودتنظیم یکبار اجرا شده و عملکرد آن تکمیل شود. برای دفعات بعد فقط لازم است که مد دستی PID را با فعال کردن فلگ M1=ON فعال نمایید.

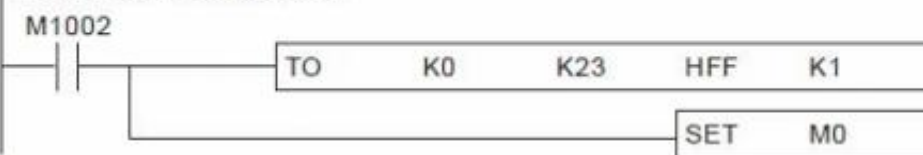
Set the target temperature of every CH to 50 degree.



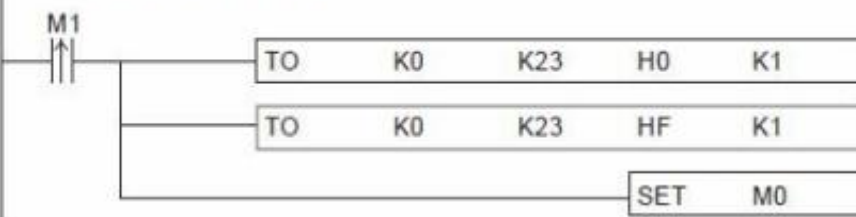
Modify D110 (unit: 1s) to set up the output cycle for GPWM.



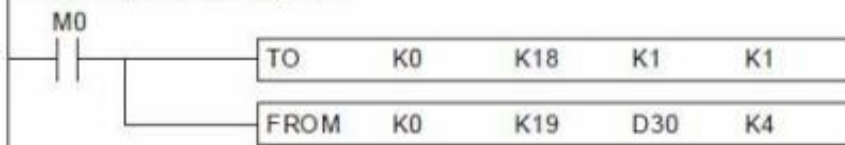
Enable auto-tuning & RUN



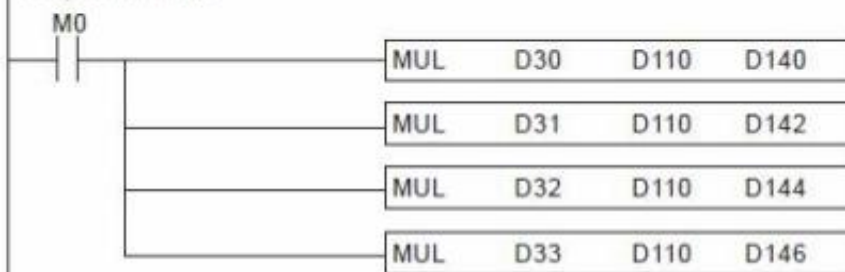
Enable manual PID & RUN



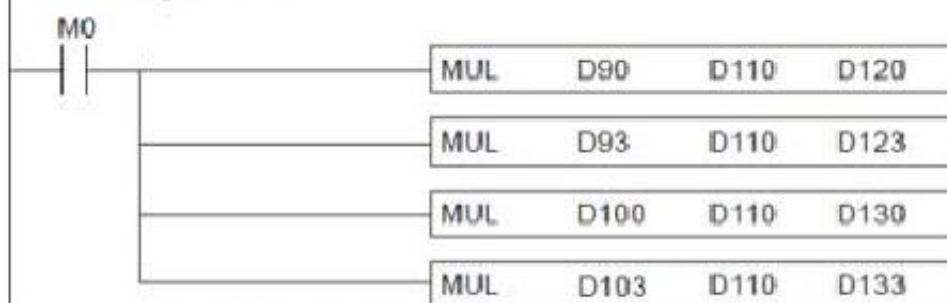
Read Output % of every CH.



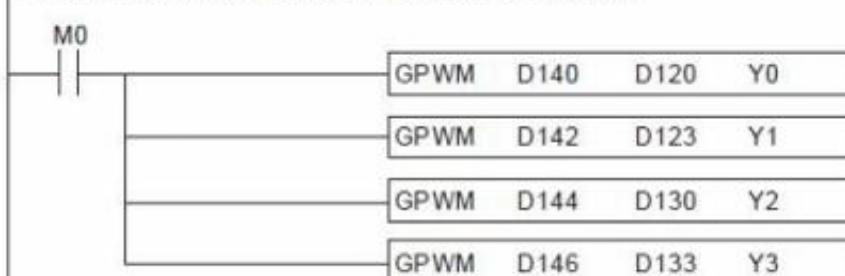
Output % x D110



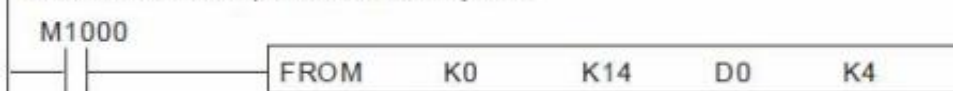
GPWM output x D110



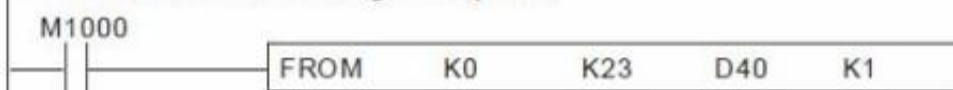
Output the % value of every CH to the corresponding Y

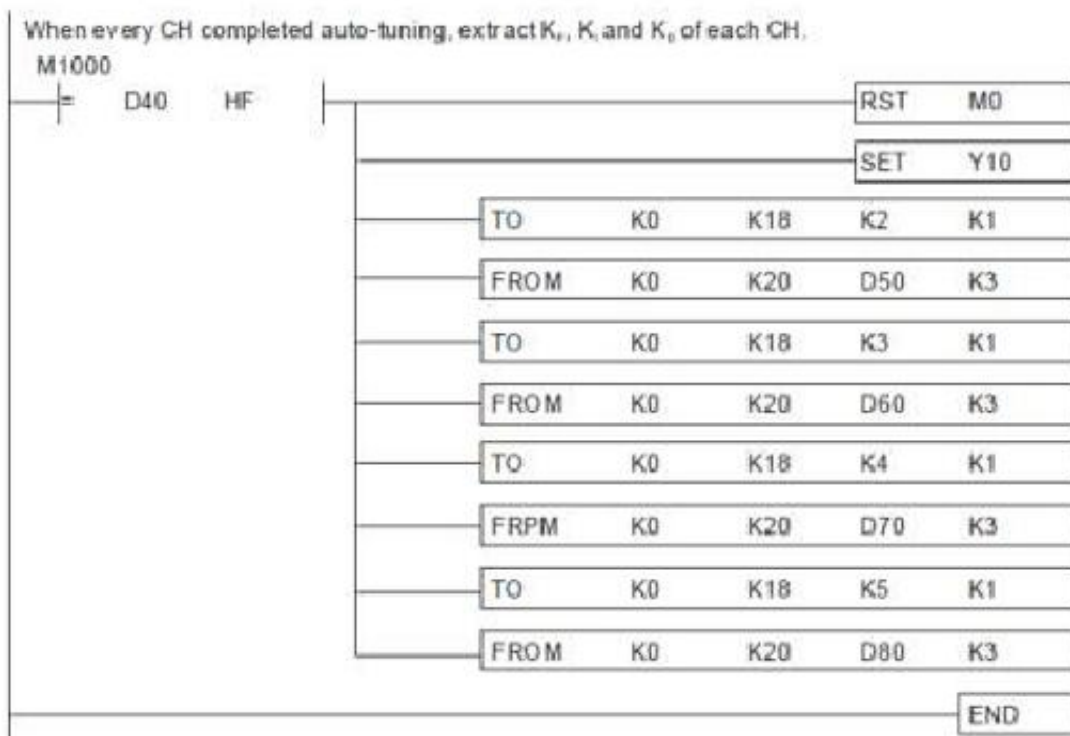


Read Celsius temperature of every CH.



Determine if the auto-tuning is completed.





7-5 مثال کاربردی PID توسط ماژول DVP04TC-H2

در این مثال می‌خواهیم دمای مطلوب را تنظیم نماییم. با فعال شدن تابع PID، تابع خودتنظیم نیز برای تنظیم ضرایب تقویت اجرا می‌شود. در این هنگام منتظر بمانید تا عملکرد تابع خودتنظیم اتوماتیک تکمیل شود. برای بهینه عمل کردن تابع خودتنظیم PID، این تابع فقط باید یکبار اجرا شود. برای دفعات بعد جهت تنظیم ضرایب تقویت باید از مد دستی PID استفاده نمایید.

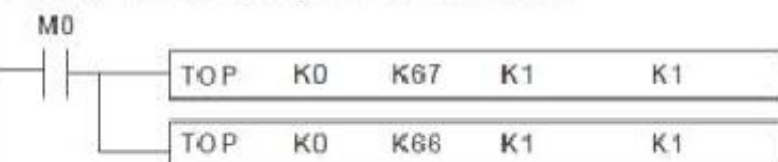
Enable M100 to set up target temperature in D500.



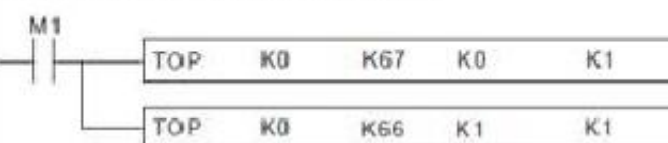
Read average temperature at CH1 every 0.5 second.



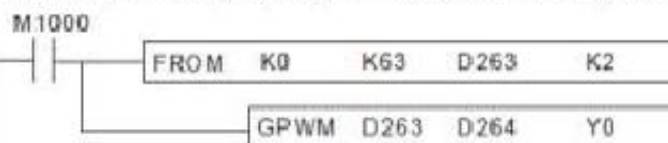
Set up PID auto-tuning and PID RUN at M0.



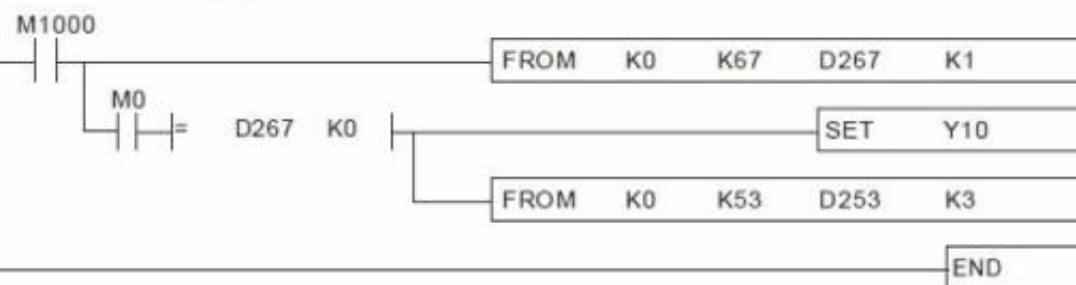
Set up PID manual and PID RUN at M1.



Read output width, output cycle and execute width adjustment instruction.



Auto-tuning CR = K0 indicates that the auto-tuning is completed. Once the auto-tuning is completed, read K_p , K_i and K_d .



8-5 دستورالعمل کنترل فازی دما (FTC)

API	Mnemonic	Operands										Function									
145	FTC	(S ₁)	(S ₂)	(S ₃)	(D)	Fuzzy Temperature Control															

Type	Bit Devices				Word Devices												Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
OP																FTC: 9 steps	
S ₁					*	*							*				
S ₂					*	*							*				
S ₃													*				
D													*				

PULSE										16-bit										32-bit									
ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2	ES	EX	SS	SA	SX	SC	EH	SV	EH3	SV2

این دستورالعمل از چهار عملوند زیر تشکیل شده است:

- عملوند S₁: در این عملوند مقدار مطلوب (تنظیمی و یا هدف) SV تنظیم می‌شود.
- عملوند S₂: در این عملوند مقدار واقعی (PV) تنظیم می‌شود.
- عملوند S₃: در این عملوند پارامترهای کنترل فازی (زمان نمونه برداری) تنظیم می‌شود.
- عملوند D: در این عملوند مقدار خروجی (MV) قرار می‌گیرد.

محدوده‌ی قابل تنظیم برای عملوند S₁ بین 1~5000 می‌باشد که این مقدار برابر با 0.1°C~500°C است. این به آن معنی است که هر واحد (هر عدد) برابر با 0.1°C است. اگر پارامتر S₃+1 برابر با K0 تنظیم شود، محدوده‌ی قابل تنظیم برای عملوند S₁ بین 0.1°C~500°C خواهد شد.

محدوده‌ی قابل نمایش برای عملوند S₂ بین 1~5000 می‌باشد که این مقدار برابر با 0.1°C~500°C است. این به آن معنی است که هر واحد (هر عدد) برابر با 0.1°C است. اگر بیت 0 عملوند S₃+1 برابر با 0 باشد، محدوده‌ی مقدار واقعی بین 0.1°C~500°C خواهد بود. از این رو، وقتی که یک مقدار آنالوگ به دیجیتال (A/D) را از سنسور دما بدست می‌آورید، این مقدار را باید به یک مقدار بین 1~5000 توسط چهار دستورالعمل ریاضی تبدیل نمایید.

اگر عملوند S₃ < K1 باشد، دستورالعمل اجرا نخواهد شد. اگر S₃ > K200 باشد، عملوند S₃ مقدار K200 را به خود اختصاص می‌دهد. عملوند S₃ هفت ابزار (رجیستر) متوالی را به خود اختصاص می‌دهد.

اگر بیت 0 عملوند S₃+1 را برابر با 0 تنظیم نمایید (bit0=0) یعنی واحد اندازه‌گیری دما را برابر با °C (سانتی‌گراد) انتخاب کرده‌اید. اگر بیت 0 عملوند S₃+1 را برابر با 1 تنظیم نمایید (bit0=1) یعنی واحد اندازه‌گیری دما را برابر با °F (فارنهایت) انتخاب کرده‌اید. اگر بیت 1 عملوند S₃+1 را برابر با 0 تنظیم نمایید (bit1=0) یعنی از تابع فیلتر استفاده نکرده‌اید و اگر بیت 1 این عملوند را برابر با 1

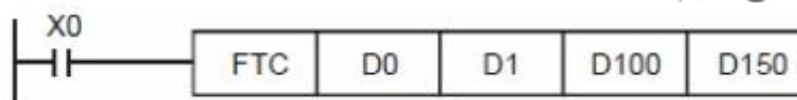
تنظیم نمایید ($bit1=1$) یعنی تابع فیلتر را فعال کرده‌اید. از بیت‌های 2 تا 5 که شامل 4 بیت است T برای تنظیم سرعت گرم کردن محیط تحت کنترل از قبیل کوره‌ها استفاده می‌شود. بیت‌های 6 الی 15 عملوند S_3+1 رزرو شده‌اند.

مقدار عملوند D بین 0 الی مدت زمان نمونه‌برداری $100 \times$ می‌باشد (0~Sampling $time \times 100$). وقتی که از این دستورالعمل استفاده می‌کنید، باید از دیگر دستورالعمل‌ها مطابق با انواع هیتراها (گرم‌کننده‌ها و المنت‌ها) استفاده نمایید. برای مثال، از دستورالعمل FTC می‌توان همراه با دستورالعمل GPWM برای کنترل پالس خروجی استفاده کرد. لازم به ذکر است، Sampling $time \times 100$ سیکل پالس خروجی دستورالعمل GPWM، و MV پهنای پالس تولید شده‌ی دستورالعمل GPWM می‌باشد. برای اطلاع از جزئیات بیشتر، مثال 2 را مشاهده کنید.

هیچ گونه محدودیتی در تعداد دفعات استفاده از دستورالعمل FTC در برنامه‌نویسی وجود ندارد.

مثال 1

در این مثال، می‌خواهیم چگونگی تنظیم پارامترهای دستورالعمل کنترل فازی دما FTC را قبل از اجرا دستورالعمل توضیح دهیم.



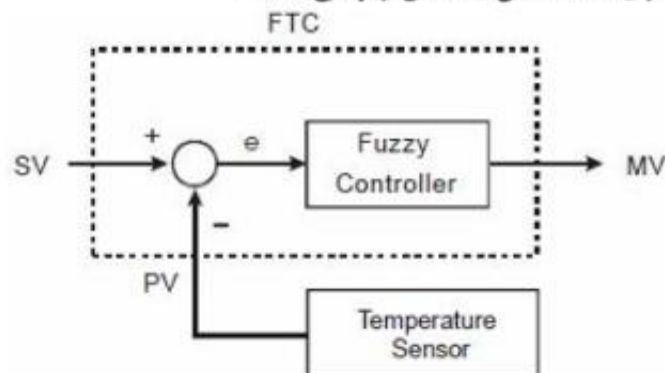
وقتی که ورودی $X0=ON$ شود، دستورالعمل اجرا شده و نتیجه بدست آمده در رجیستر D150 ذخیره می‌شود. وقتی که ورودی $X0=OFF$ شود، دستورالعمل اجرا نخواهد شد و اطلاعات قبلی بدون تغییر باقی خواهد ماند.

تنظیمات پارامترهای دستورالعمل کنترل فازی دما در عملوند S_3 الی S_3+6 صورت خواهد گرفت که عبارتند از:

شماره ابزار	تابع	محدوده‌ی قابل تنظیم	توضیحات
S_3	زمان نمونه‌برداری (T_s) واحد: 100ms	1~200 واحد: 100ms	اگر زمان تنظیمی برای T_s کمتر از زمان یک سیکل اسکن باشد، دستورالعمل FTC برای یک سیکل اسکن برنامه اجرا خواهد شد. اگر $T_s=0$ باشد، دستورالعمل FTC فعال نخواهد شد. حداقل زمان تنظیمی برای T_s باید بزرگتر از زمان یک سیکل اسکن برنامه باشد.
S_3+1	Bit0: واحداندازی‌گیری دما Bit1: تابع فیلترینگ	$B0=0$ سانتی‌گراد °C $B0=1$ فارنهایت °F	از بیت 0 این پارامتر (رجیستر) برای انتخاب واحد اندازه‌گیری دمای واقعی خوانده شده استفاده می‌شود.
	$B2-b5$: تنظیم سرعت گرم کردن محیط $B6-b15$: رزرو شده	$B1=0$ بدون تابع فیلتر $B1=1$ با تابع فیلتر	وقتی که در کنترل دمای فازی از تابع فیلتر استفاده نشده باشد، مقدار واقعی (PV) برابر با مقدار اندازه‌گیری شده در هر لحظه می‌باشد. وقتی که در کنترل دمای فازی از تابع فیلتر

استفاده شده باشد، مقدار واقعی (PV) برابر با مقدار اندازه‌گیری شده در هر لحظه به‌علاوه‌ی مقدار PV قبلی تقسیم بر 2 خواهد بود $PV = (\text{Currently measured value} + \text{previous PV}) / 2$			
گرم کردن دمای محیط تحت کنترل (از قبیل کوره‌ها) به کندی صورت خواهد گرفت.	B2=1		
گرم کردن دمای محیط تحت کنترل (از قبیل کوره‌ها) به صورت معمولی صورت خواهد گرفت.	B3=1		
گرم کردن دمای محیط تحت کنترل (از قبیل کوره‌ها) به سرعت صورت خواهد گرفت.	B4=1		
گرم کردن دمای محیط تحت کنترل (از قبیل کوره‌ها) خیلی سریع صورت خواهد گرفت.	B5=1		
این رجیسترها پارامترهایی می‌باشند که سیستم برای عملکرد خود از آنها استفاده خواهد کرد. از این رو، ما نمی‌توانیم از آنها در برنامه‌نویسی استفاده کنیم. درحقیقت این پارامترها، پارامترهای سیستمی می‌باشند.			$S_3+2 \sim S_3+6$

بلوک دیاگرام کنترل فازی دما مطابق با شکل زیر می‌باشد:



چند نکته مهم را باید متذکر شویم:

پیشنهاد می‌شود که زمان نمونه‌برداری را دو برابر بیشتر از زمان نمونه‌برداری سنسور دما برای کنترل بهتر دما تنظیم نمایید.

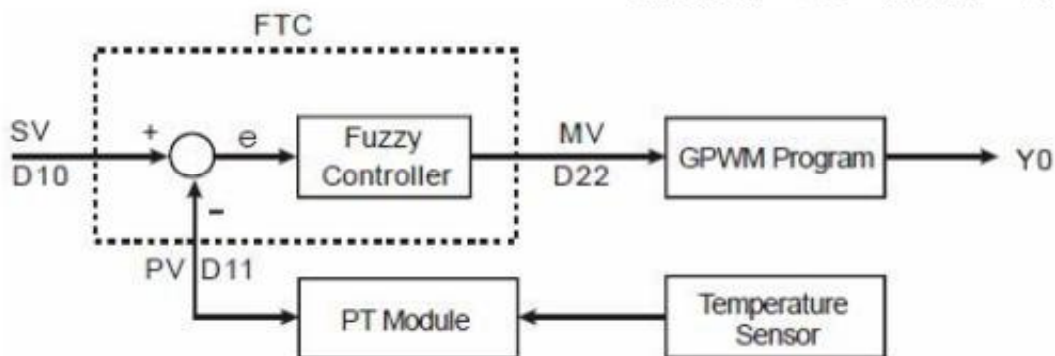
از بیت‌های 2 الی 5 پارامتر S_3+1 برای تعیین سرعت کنترل دمای واقعی استفاده می‌شود. اگر سرعت کنترل دما را تنظیم ننمایید، دستورالعمل FTC به صورت اتوماتیک "گرم کردن دمای محیط تحت کنترل (از قبیل کوره‌ها) به صورت معمولی Bit3=1" را انتخاب خواهد کرد. اگر متوجه شدید که سرعت کنترل رسیدن دمای واقعی (PV) به دمای هدف (SV) خیلی کند است، باید "گرم کردن دمای محیط تحت کنترل (از قبیل کوره‌ها) به کندی" را برای بالا بردن سرعت رسیدن دمای واقعی (PV) به دمای هدف (SV) انتخاب نمایید. بالعکس، وقتی که متوجه شدید کنترل دما و سرعت رسیدن دمای واقعی (PV) به دمای هدف (SV) خیلی سریع است و یا دما خیلی نوسان دارد، باید "گرم کردن دمای

محیط تحت کنترل (از قبیل کوره‌ها) به صورت سریع " را جهت کند کردن سرعت کنترل دما انتخاب نمایید.

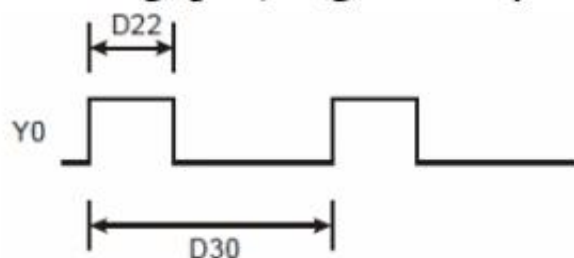
وقتی که بیت‌های 2 الی 5 پارامتر S_3+1 همگی برابر با 1 تنظیم شوند و یا بیشتر از یک نوع سرعت کنترل دما انتخاب شود، دستورالعمل FTC از بیت 2 الی 5 پارامتر S_3+1 را بررسی کرده و تابعی را که برابر با 1 تنظیم شده است را فعال می‌نماید. پارامتر S_3+1 را در هنگام اجرای دستورالعمل FTC را نیز می‌توان تنظیم کرد.

مثال 2

در این مثال می‌خواهیم دمای یک کوره را با استفاده از دستورالعمل‌های کنترل فازی دما FTC و دستورالعمل کنترل پهنای پالس عمومی GPWM کنترل نماییم. دیاگرام کنترلی این مثال به شرح زیر می‌باشد:



خروجی (MV) D22 دستورالعمل کنترل فازی دما FTC به ورودی دستورالعمل تولید موج PWM عمومی با نام GPWM متصل شده است که سیکل کاری آن توسط پالس‌ها قابل تنظیم می‌باشد. در پارامتر D30 زمان سیکل ثابت پالس‌ها تنظیم می‌شود. دیاگرام شکل زیر خروجی Y0 که وابسته به دو دستورالعمل FTC و GPWM می‌باشد را نشان می‌دهد:



تنظیمات پارامترهای پیش‌فرض این مثال به شرح زیر است:

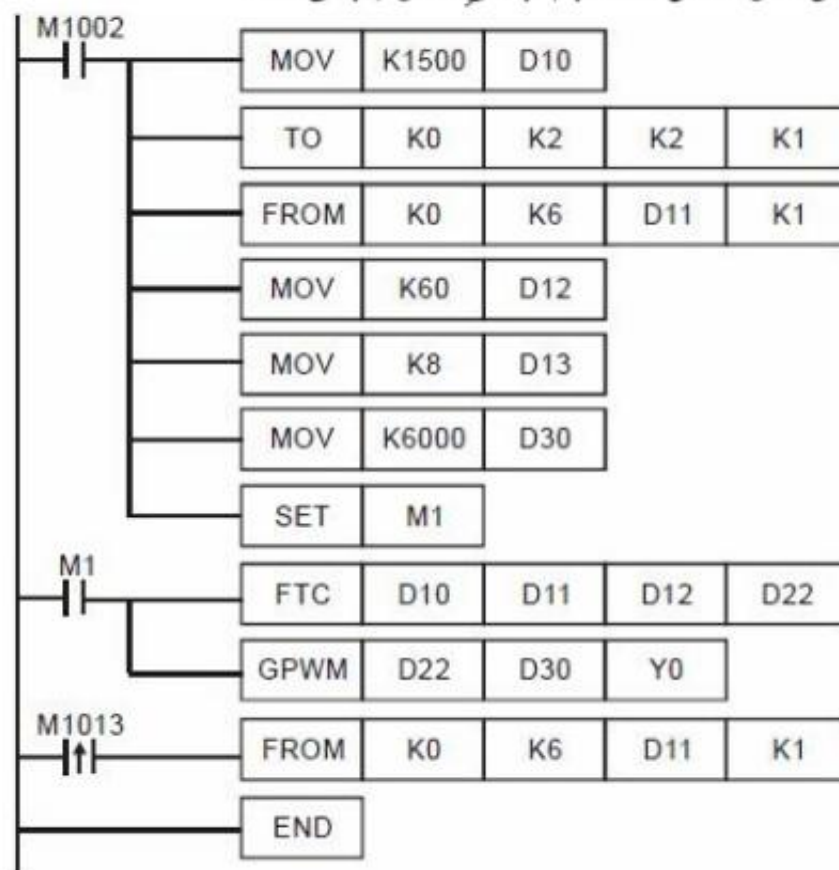
دمای هدف (SV) ← $D10=K1500$

زمان نمونه‌برداری 6 ثانیه T_S ← $D12=K60$

سرعت عملکرد کنترل دما برابر با گرم کردن با سرعت معمولی می‌باشد ← $D13=K8$

زمان یک سیکل ثابت پالس خروجی موج PWM ← $D30 = K6000 (=D12 \times 100)$

برنامه کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می باشد:

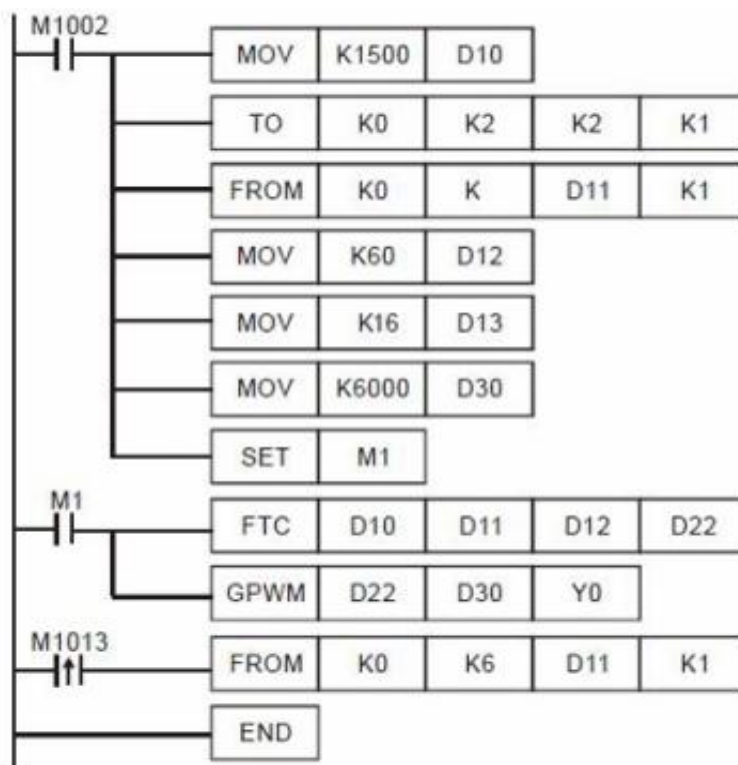


در این مثال، دمای کوره را می توان تا حدود 250 درجه سانتی گراد بالا برد. در شکل زیر، منحنی دمای واقعی کوره (PV) و دمای هدف (SV) ثبت شده در یک ثبات نشان داده شده است. همانطور که در دیاگرام زیر مشاهده می کنید، دمای کوره پس از گذشت 48 دقیقه به دمای هدف رسیده است. تفرانس دمای مطلوب به دمای هدف حدود $\pm 1^{\circ}\text{C}$ و دمای تجاوز شده از دمای هدف در انتهای فرآیند دما حدوداً 10°C است.

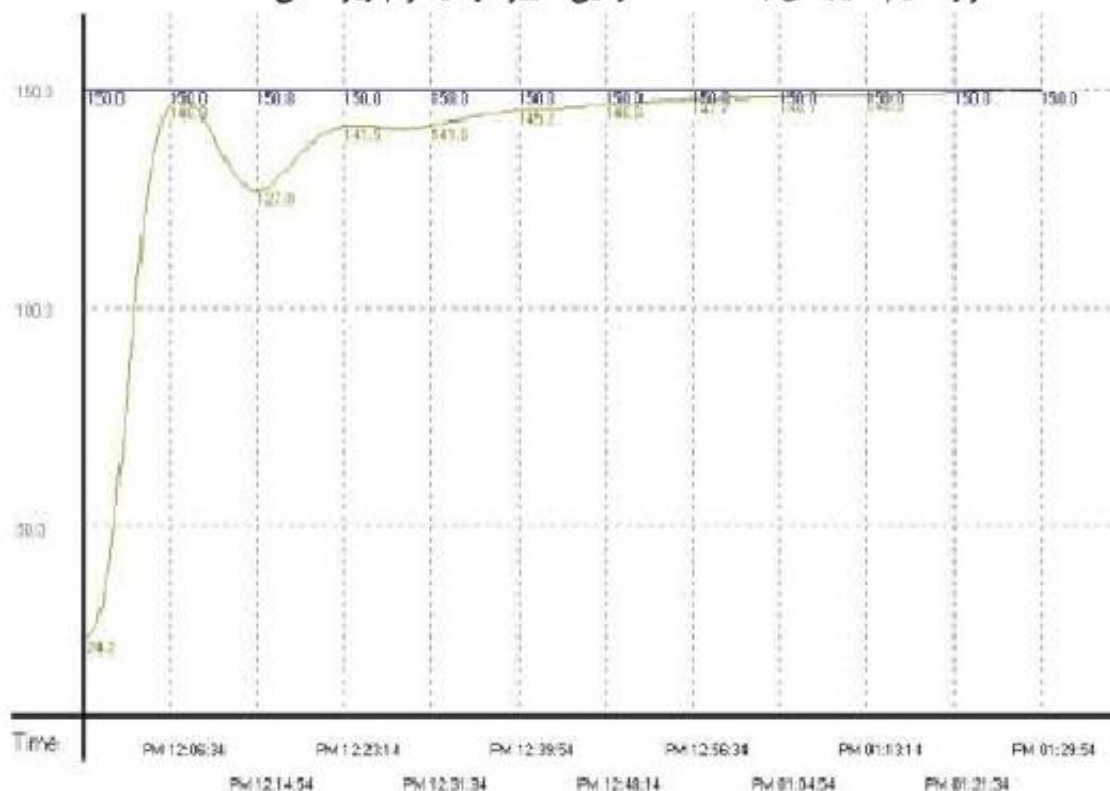


مثال 3

در این مثال، می‌خواهیم برنامه‌ای بنویسیم که دمای واقعی کوره (PV) از دمای هدف (SV) تجاوز نکند.



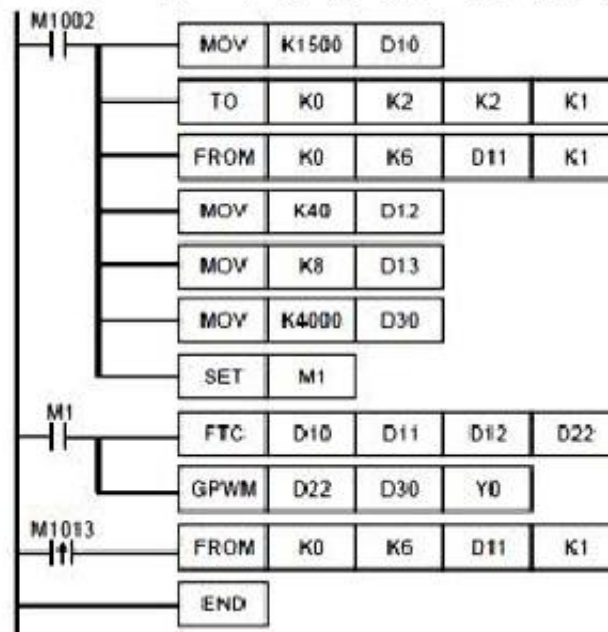
به علت تجاوز یکمرتبه‌ی دمای واقعی کوره (PV) از دمای هدف (SV)، ما باید سرعت گرم کردن محیط کوره را به "گرم کردن دمای داخل کوره با سرعت بالا" تغییر دهیم. برای اینکار باید D13=K16 قرار دهیم. نتیجه‌ی بدست آمده از این تغییر در دیاگرام زیر نشان داده شده است:



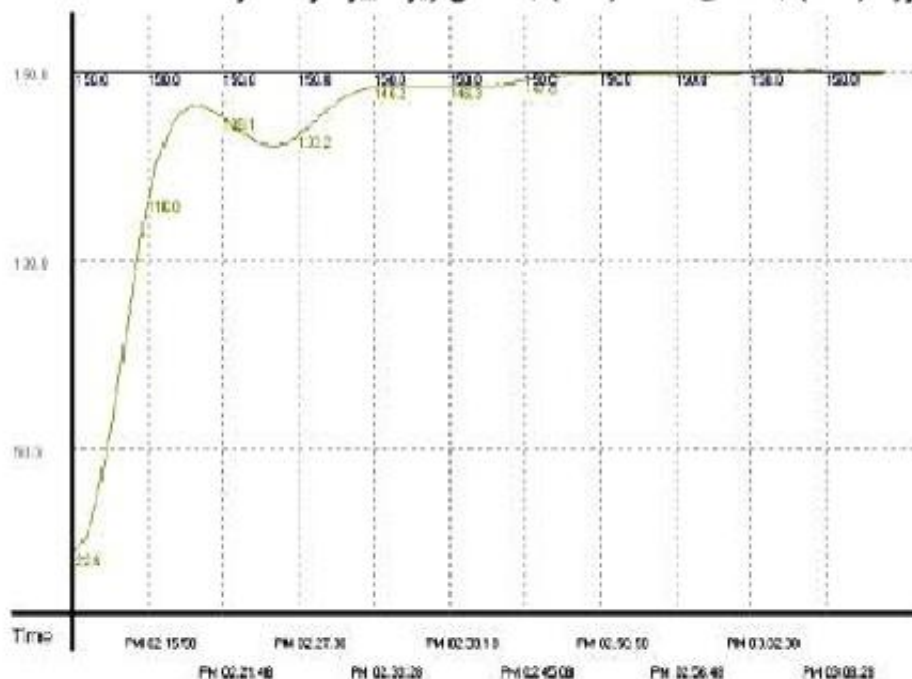
با مشاهده دیاگرام بالا متوجه خواهیم شد که اگرچه دمای واقعی کوره از دمای هدف تجاوز نکرده است، اما زمان رسیدن دمای واقعی به دمای هدف با تolerانس $\pm 1^{\circ}\text{C}$ به 1 ساعت و 15 دقیقه رسیده است. این امر بیانگر آن است که با سریع کردن عملکرد دستورالعمل FTC زمان رسیدن دمای واقعی (PV) به دمای هدف (SV) افزایش خواهد یافت. از این رو، زمان نمونه‌برداری خیلی طولانی می‌باشد، اما نوسان دما کمتر است.

مثال 4

در این مثال، می‌خواهیم زمان رسیدن دمای واقعی کوره (PV) به دمای هدف (SV) را کم کنیم.



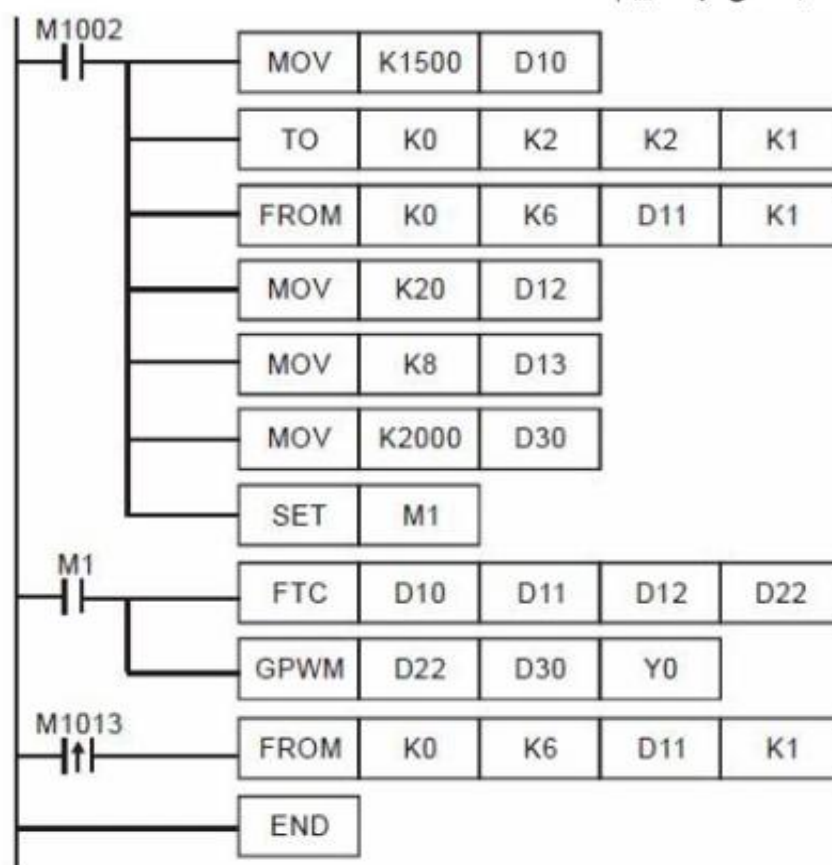
برای سریع کردن سرعت رسیدن دمای واقعی (PV) به دمای هدف (SV)، زمان نمونه‌برداری را باید برابر با 4 ثانیه تنظیم نمایید ($D12=K40$, $D30=K4000$). با تنظیم زمان نمونه‌برداری برابر با 4 ثانیه ($D12=K40$) و پهنای پالس موج PWM برابر با $D30=K4000 (=D12 \times 100)$ منحنی رسیدن دمای واقعی کوره (PV) به دمای هدف (SV) به شکل زیر تغییر خواهد کرد:



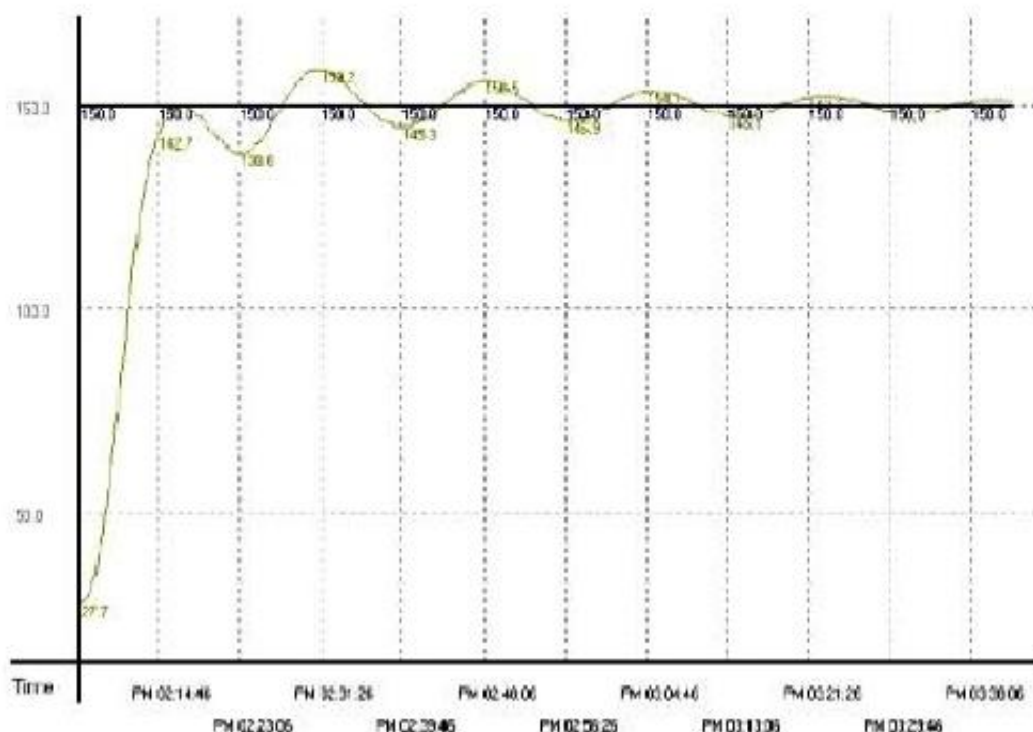
با مشاهده‌ی دیاگرام بالا متوجه خواهیم شد که کل زمان کنترل دما کوتاه‌تر شده و به 37 دقیقه رسیده است. از این رو، متوجه شدیم که با اصلاح و تغییر زمان نمونه‌برداری (T_s) می‌توان زمان و سرعت رسیدن دمای واقعی (PV) به دمای هدف (SV) را تغییر داده و کم و زیاد کرد.

مثال 5

در این مثال، می‌خواهیم زمان رسیدن دمای واقعی کوره (PV) به دمای هدف (SV) را خیلی سریع‌تر (در مقایسه با مثال 4) نماییم.



اگر بخواهیم دمای واقعی را سریع‌تر به دمای هدف برسانیم، باید زمان نمونه‌برداری که در مثال 4 برابر با $D12=40$ و $D30=4000$ تنظیم شده است را به $D12=K20$ و $D30=2000$ تغییر دهیم. دیاگرام خروجی این مثال مطابق با دیاگرام نشان داده شده در شکل زیر می‌باشد:



با مشاهده‌ی دیاگرام بالا متوجه خواهیم شد که با کمتر کردن زمان نمونه‌برداری سیستم کنترلی خیلی حساس شده و نوسانی می‌شود و دما بالا و پایین خواهد شد.

4 روش کنترل PID

در ماژول DVP04PT دو نوع روش کنترل برای کنترل‌کننده‌ی PID وجود دارد.

روش کنترل سیکلی

روش کنترل سیکلی به **Cyclic Control Mode** معروف می‌باشد.

نحوه‌ی عملکرد مد کنترل سیکلی در ماژول DVP04PT-S به شرح زیر است:

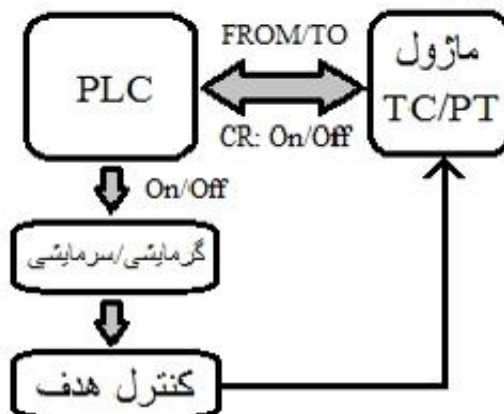
سیکل خروجی را می‌توان مطابق با محیط تحت کنترل تنظیم کرد. اگر دما در محیط تحت کنترل به کندی تغییر کند، شما می‌توانید سیکل خروجی را طولانی‌تر تنظیم نمایید. پهنای پالس خروجی از فرمول $\text{Output Width} = \text{Cycle} \times \text{Output\%}$ بدست خواهد آمد. برای کنترل سیکلی از دستورالعمل **GPWM** استفاده می‌شود. پهنای خروجی و سیکل خروجی را باید در دستورالعمل **GPWM** تنظیم نمایید. در اینجا ما سیستم گرمایشی (هیتر) و یا سیستم سرمایشی (فن) را به خروجی **Y0** متصل می‌نماییم. برای مثال، فرض کنید که زمان سیکل خروجی برابر با 3 ثانیه است، از این رو باید مقدار **K3000** را در رجیستر **D11** بنویسید. $\text{D10} = \text{Output\%} \times \text{K3000} / 1000$. واحد **Output\%** در فرمول بالا 0.1% است.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



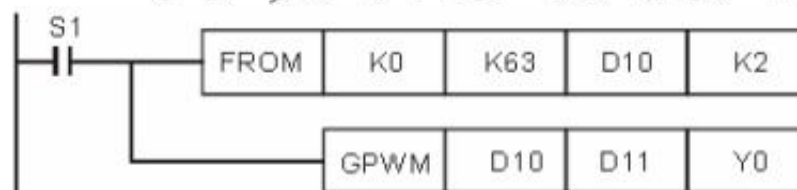
نحوه‌ی عملکرد مدکنترلی سیکلی در ماژول DVP04PT-H2 به شرح زیر است:

اگر کنترل سیستم گرمایشی (هیتر) و یا سرمایشی (فن) از طریق قطع و وصل تغذیه‌ی سیستم می‌باشد، باید از کنترل سیکلی با استفاده از دستورالعمل GPWM استفاده کنید. شکل زیر، روش سیم‌بندی کنترلی سیکلی را نشان می‌دهد:

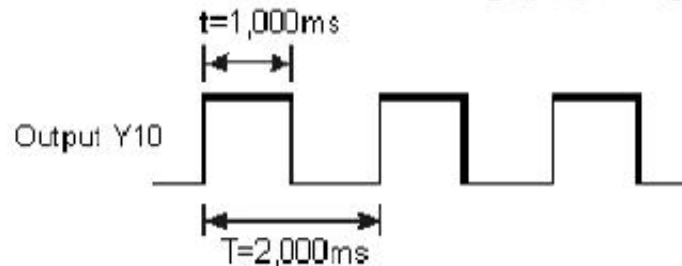


شما باید دو رجیستر کنترلی (CR) را از ماژول DVP04PT بخوانید. نخستین رجیستر کنترلی (CR) زمان سیکل قابل استفاده (مؤثر) و دومین رجیستر کنترلی پهنای قابل استفاده (مؤثر) است. کنترل سیکلی توسط دستورالعمل GPWM در DVP-PLC کار می‌کند. برای مثال، سیگنال‌های دریافتی از سنسور متصل شده به کانال CH1 ماژول DVP04PT را خوانده و پهنای پالس خروجی کانال CH1 را از رجیستر کنترلی CR#63 می‌خوانیم و همچنین باید سیکل خروجی کانال CH1 را از رجیستر کنترلی CR#64 توسط دستورالعمل FROM بخوانیم. مقدار رجیسترهای کنترلی CR#63 و CR#64 را با استفاده از دستورالعمل FROM خوانده و از آنها در دستورالعمل GPWM جهت کنترل خروجی سیستم (Y0) استفاده می‌نماییم. به خروجی Y0 می‌توان هیتر یا فن متصل کرد.

شکل زیر، برنامه‌ی کنترلی این مثال را مطابق با دیاگرام نردبانی نشان می‌دهد:



فرض کنید که پهنای پالس 1000ms بوده و زمان یک سیکل 2000ms است. پهنای پالس خروجی این برنامه مطابق با شکل زیر می‌باشد:



روش کنترل خروجی آنالوگ

روش کنترل خروجی آنالوگ به Analog Output Control معروف است.

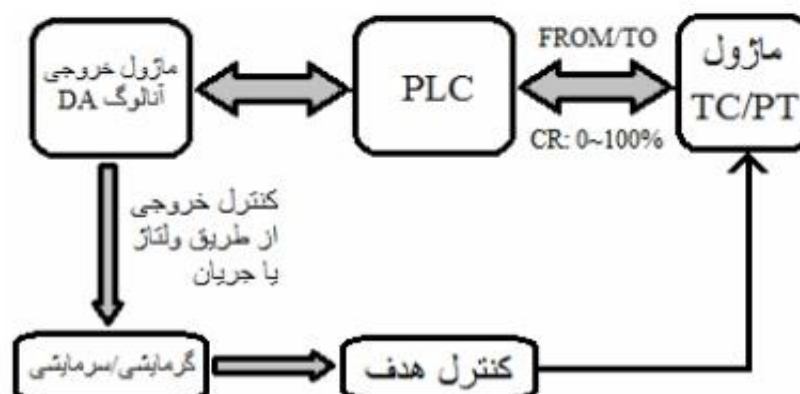
نحوه‌ی عملکرد مد کنترل خروجی آنالوگ در ماژول DVP04PT-S به شرح زیر می‌باشد:

در کنترل خروجی آنالوگ، مقدار درصد خروجی (Output% (0~100%)) از کنترل‌کننده‌ی PID خوانده شده و مطابق با این مقدار، مقدار خروجی آنالوگ در ماژول خروجی آنالوگ DVP04DA جهت کنترل سیستم گرمایشی استفاده خواهد شد. برای مثال، خروجی آنالوگ (0~4000(0~10V)) را از ماژول DVP04DA-S به یک سیستم گرمایشی آنالوگ متصل می‌نماییم و به عبارت دیگر، یعنی یک سیستم گرمایشی آنالوگ را به خروجی ولتاژی ماژول DVP04DA-S متصل کرده‌ایم. برای کنترل خروجی آنالوگ ماژول DVP04DA-S باید مقدار درصد خروجی Output% را از ماژول DVP04PT-S بخوانیم. مقدار خروجی آنالوگ برابر با $\text{Output Value} = 8000 \times \text{Output\%/100}$ می‌باشد. واحد خروجی برابر با 0.1% است. خروجی آنالوگ با این مقدار توسط دستورالعمل TO کنترل خواهد شد.



نحوه‌ی عملکرد مد کنترل خروجی آنالوگ در ماژول DVP04PT-H2 به شرح زیر است:

اگر می‌خواهید یک سیستم گرمایشی یا سرمایشی را با استفاده از سیگنال کنترل ولتاژی یا جریانی کنترل نمایید، نوع مد کنترل خروجی آنالوگ را باید پیاده سازی نمایید. شکل زیر، روش سیم‌بندی مد کنترل خروجی آنالوگ را نشان می‌دهد:



برای کنترل سیستم گرمایشی یا سرمایشی توسط خروجی آنالوگ، باید محدوده‌ی خروجی آنالوگ را تنظیم نمایید. با استفاده از دستورالعمل FROM مقدار خروجی را از ماژول DVP04PT خوانده و با استفاده از دستورالعمل TO مقدار خروجی را به ماژول خروجی آنالوگ DVP04DA ارسال نمایید.

مثال

محدوده‌ی خروجی یک سیستم گرمایشی/سرمایشی (4~20mA) 800~4000 می‌باشد. برای کنترل سیستم گرمایشی/سرمایشی از ماژول خروجی آنالوگ DVP04DA-H2 استفاده کرده‌ایم. سیستم گرمایشی/سرمایشی را به ترمینال خروجی جریانی این ماژول متصل می‌نماییم. محدوده‌ی بالای خروجی ماژول را برابر با 4000 و محدوده‌ی پایین خروجی ماژول را برابر با 800 تنظیم می‌کنیم. مقدار خروجی دیجیتال را باید از ماژول DVP04PT-H2 خواند. با استفاده از دستورالعمل FROM مقدار خروجی دیجیتال را از ماژول دما خوانده و آن را در رجیستر D10 ذخیره می‌نماییم، و با استفاده از دستورالعمل TO مقدار خروجی دیجیتال را به ماژول خروجی آنالوگ ارسال می‌کنیم. در شکل زیر، برنامه کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده است.



در شکل زیر وابستگی بین مقدار خروجی دیجیتال و مقدار خروجی جریانی نشان داده شده است.

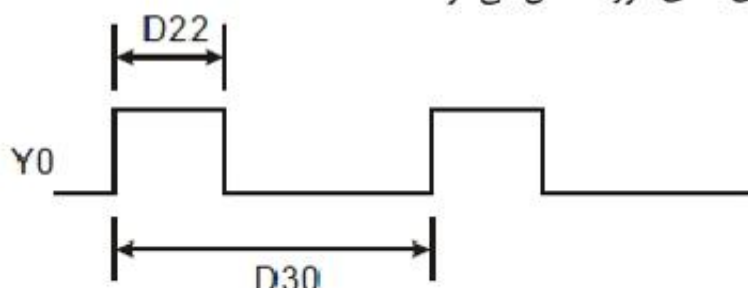


9-5 کنترل فازی دمای کوره

در این مثال، می‌خواهیم با استفاده از دستورالعمل کنترل فازی دما FTC دمای یک کوره را کنترل نماییم. هدف از کنترل در این مثال به شرح زیر است:

می‌خواهیم محیط داخل کوره را سریع گرم کنیم "Fast heating environment". برای این کار باید $D13=K16$ شود و همچنین میزان دمای مطلوب (هدف) در این مثال برابر با 120°C است که برای این کار باید $D10=K1200$ تنظیم کرد. برای بدست آوردن بهترین نتیجه‌ی کنترلی، در برنامه از دستورالعمل FTC به همراه دستورالعمل GPWM برای اجرای کنترل فازی دما استفاده شده است.

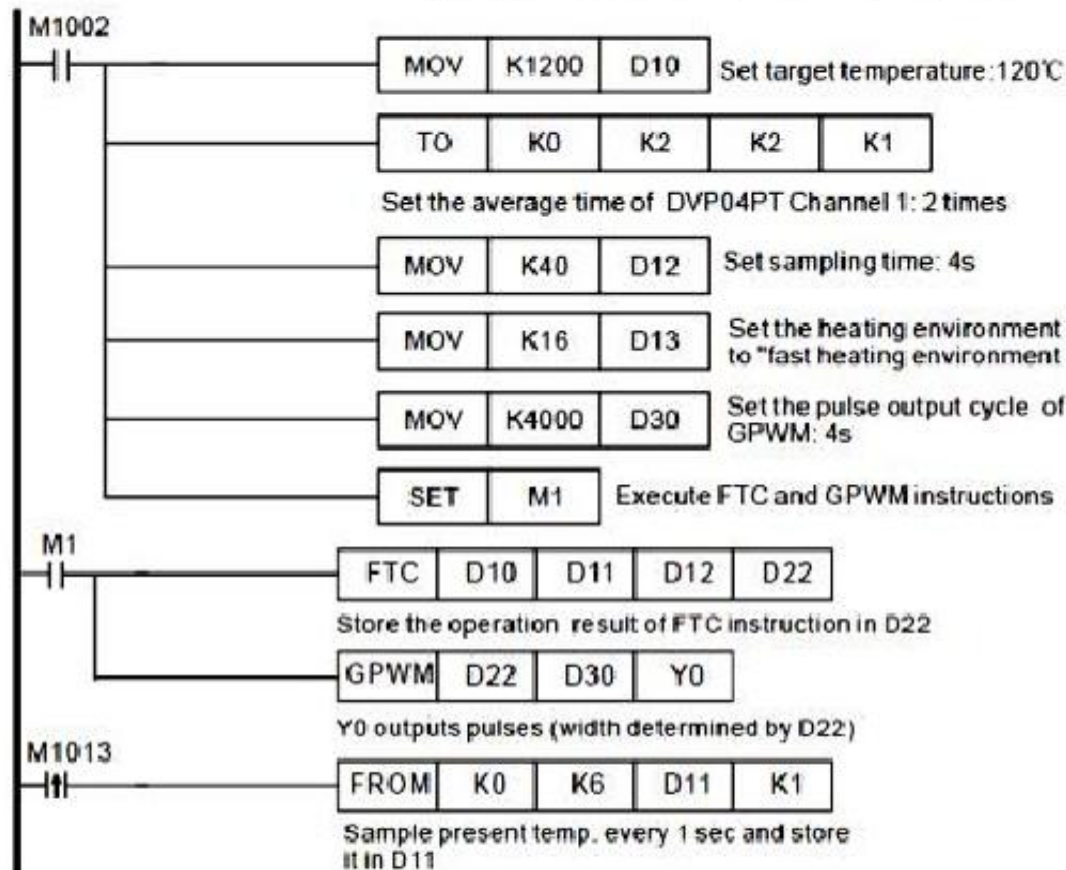
برای اندازه‌گیری دما از سنسور PT100 استفاده شده است که این سنسور به ماژول DVP04PT-S متصل می‌شود. در این مثال، از CPU سری DVP12SA استفاده شده است. بعد از اجرای دستورالعمل FTC، نتیجه عملیات دستورالعمل FTC در خروجی دستورالعمل به ورودی دستورالعمل GPWM اعمال می‌شود. پهنای پالس مدولاسیون دستورالعمل GPWM توسط رجیستر D22 تعیین شده و خروجی دستورالعمل GPWM (پهنای پالس) به خروجی Y0 جهت کنترل گرم‌کن (المنت) و کنترل فازی دمای کوره اعمال می‌شود.



ابزارهای مورد استفاده در این برنامه به شرح زیر می‌باشد:

ابزار	دستورالعمل
M1	از این فلگ برای اجرای دستورالعمل FTC استفاده می‌شود.
Y0	به این خروجی، پالس خروجی ایجاد شده‌ی PWM اعمال می‌شود.
D10	در این رجیستر دمای مطلوب (هدف) تنظیم شده ذخیره می‌شود.
D11	از این رجیستر برای ذخیره کردن دمای واقعی کوره استفاده می‌شود.
D12	در این رجیستر پارامتر زمان نمونه‌برداری دستورالعمل FTC قرار می‌گیرد که زمان نمونه‌برداری در این رجیستر ذخیره می‌شود.
D13	در این رجیستر پارامتر کنترل دمای دستورالعمل FTC ذخیره می‌شود.
D22	در این رجیستر نتیجه‌ی بدست آمده از دستورالعمل FTC ذخیره می‌شود.
D30	در این رجیستر سیکل خروجی پالس دستورالعمل GPWM ذخیره می‌شود.

برنامه کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می باشد:



عملکرد این برنامه کنترلی به شرح زیر است:

دستورالعمل FTC یک دستورالعمل منحصر به فرد برای کنترل دما به صورت دستی می باشد. برخلاف دستورات PID که دارای پارامترهای زیادی جهت کنترل می باشد، در دستورالعمل FTC شما فقط باید چند پارامتر را تنظیم نمایید.

فرمت دستورالعمل FTC مطابق با شکل زیر است:

FTC	S ₁	S ₂	S ₃	D
-----	----------------	----------------	----------------	---

عملوند S₁ ← مقدار مطلوب (هدف) دما SV ← محدوده‌ی قابل تنظیم: 1~5000 مطابق با 0.1°C~500.0°C می باشد، واحد 0.1°C

عملوند S₂ ← مقدار واقعی دما PV ← محدوده‌ی قابل نمایش 1~5000 مطابق با 0.1°C~500.0°C می باشد، واحد 0.1°C

عملوند S₃ ← پارامترهای دستورالعمل FTC که فقط باید S₃ و S₃+1 تنظیم شوند.

عملوند D ← مقدار خروجی MV، محدوده‌ی خروجی 0~S₃×100 می باشد.

تنظیمات پارامترهای S_3 و S_3+1 به شرح زیر می‌باشد:

پارامتر	تابع	محدوده‌ی قابل تنظیم
S_3	زمان نمونه‌برداری T_S	1~200، واحد زمان ms می‌باشد
S_3+1	b_0 : واحد اندازه‌گیری دما b_1 : تابع فیلتر b_2 ~ b_5 : گرم کردن محیط تحت کنترل b_6 ~ b_{15} : رزرو شده	$b_0=0$ به معنی سانتی‌گراد (C) می‌باشد
		$b_0=1$ به معنی فارنهایت (F) می‌باشد
		$b_1=0$ به معنی فعال نبودن تابع فیلتر است.
		$b_1=1$ به معنی فعال بودن تابع فیلتر است.
		$b_2=1$ به معنی گرم کردن آهسته محیط تحت کنترل است.
		$b_3=1$ به معنی گرم کردن معمولی محیط تحت کنترل است.
		$b_4=1$ به معنی گرم کردن سریع محیط تحت کنترل است.
		$b_5=1$ به معنی گرم کردن خیلی سریع محیط تحت کنترل است.

در برنامه‌های کاربردی، معمولاً لازم است که چندین بار پارامترهای S_3 و S_3+1 را تنظیم نمایید تا بهترین نتیجه‌ی کنترل را بدست آورید. قواعد اصلی کار با دستورالعمل FTC در زیر توضیح داده شده است.

- زمان نمونه‌برداری باید دو برابر بزرگتر از زمان نمونه‌برداری سنسور دما تنظیم شود. معمولاً زمان نمونه‌برداری T_S بین $2S$ ~ $6S$ تنظیم می‌شود.

- زمان سیکل کاری دستورالعمل GPWM باید برابر با زمان نمونه‌برداری (T_S) دستورالعمل FTC تنظیم شود، اما دقت داشته باشید که واحد زمان برای زمان سیکل کاری دستورالعمل GPWM برابر با 1ms می‌باشد.

- اگر مدت زمان کنترل دما خیلی طولانی بود، باید زمان نمونه‌برداری از SV را کاهش دهید.
 - اگر دمای تحت کنترل مکرراً دارای نوسان بود باید، زمان نمونه‌برداری از SV را افزایش دهید.
 - سرعت گرم کردن محیط تحت کنترل به صورت پیش‌فرض برابر با $b_3=1$ (گرم کردن معمولی محیط تحت کنترل) می‌باشد.

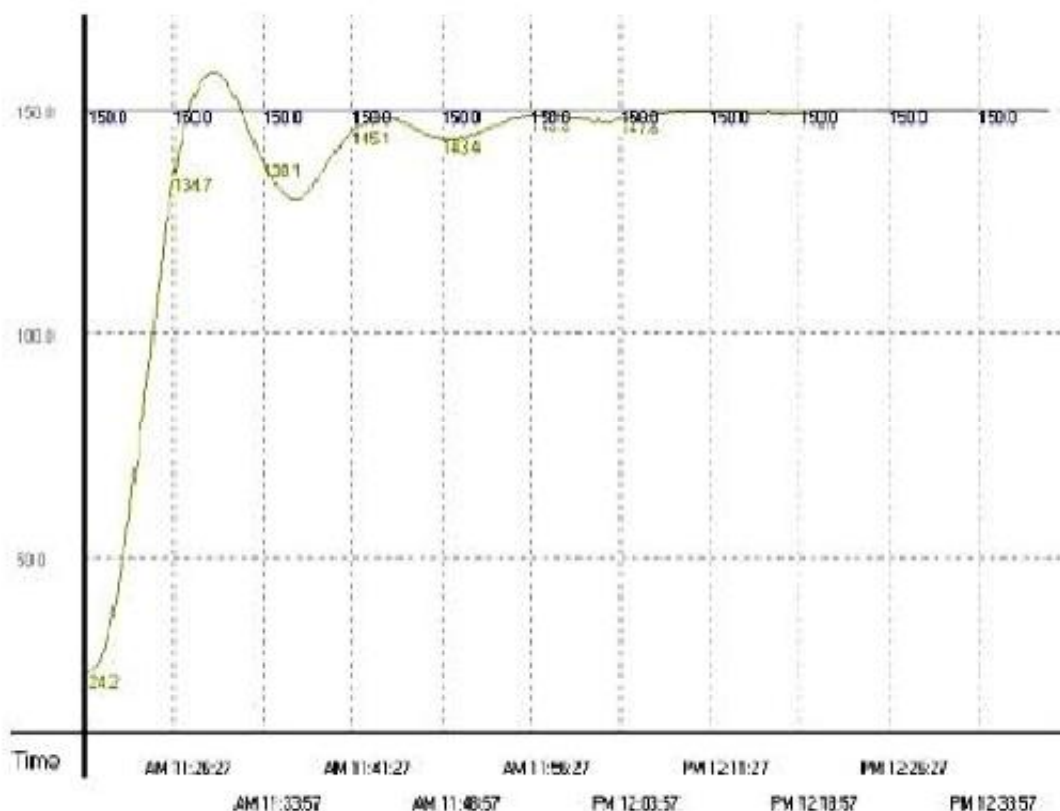
- اگر مدت زمان کنترل دما و رسیدن دمای واقعی به دمای هدف خیلی زیاد بود، باید $b_2=1$ (گرم کردن آهسته محیط تحت کنترل) را انتخاب کنید.

- اگر دمای واقعی محیط تحت کنترل خیلی بالا باشد و یا دمای واقعی محیط تحت کنترل نوسانی بود، باید $b_4=1$ (گرم کردن سریع محیط تحت کنترل) را انتخاب کنید.

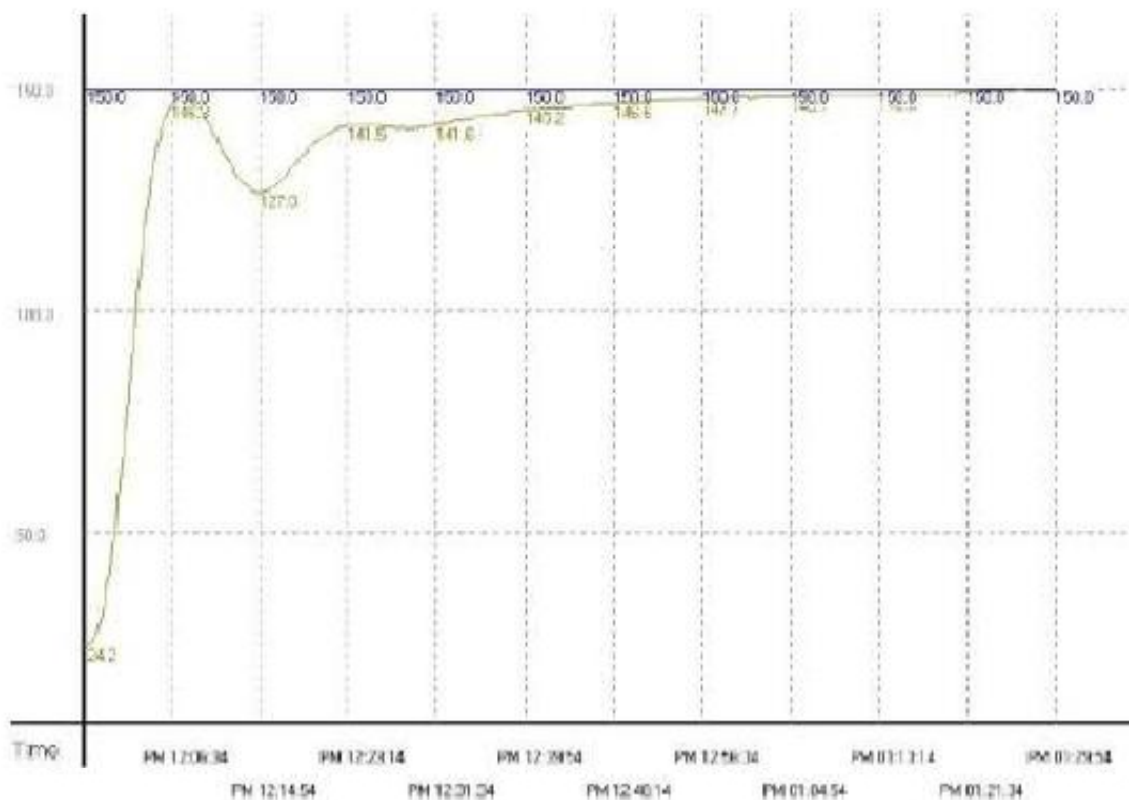
- توجه داشته باشید که بیت‌های b_2 , b_3 , b_4 , b_5 نباید همزمان با هم یک باشند. در یک زمان فقط باید یکی از بیت‌های گفته شده 1 باشد. در غیر این صورت، برنامه بد عمل کرده و دمای کوره را نمی‌توان کنترل کرد.

در این برنامه ما تنظیمات مختلفی را برای پارامترهای S_3+1 و S_3 اجرا کرده‌ایم و نتایج بدست آمده برای هر یک از تنظیمات را ثبت نموده و آنها را برای شما شرح خواهیم داد.

ابتدا پارامترها S_3 و S_3+1 دستورالعمل FTC را برابر با $D12=K60$ (6s) و $D13=K8$ (b3=1) و سیکل زمانی پالس خروجی دستورالعمل GPWM را برابر با $D30=K6000$ ($=D12 \times 100$) تنظیم کردیم، منحنی کنترل دمای بدست آمده در این شرایط مطابق با دیاگرام زیر بود:

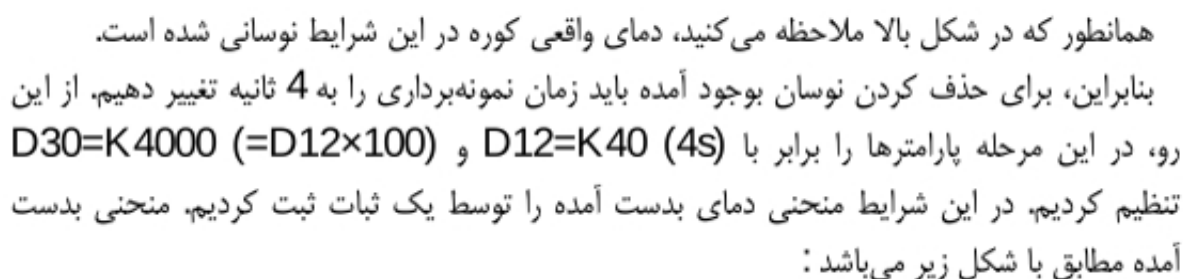


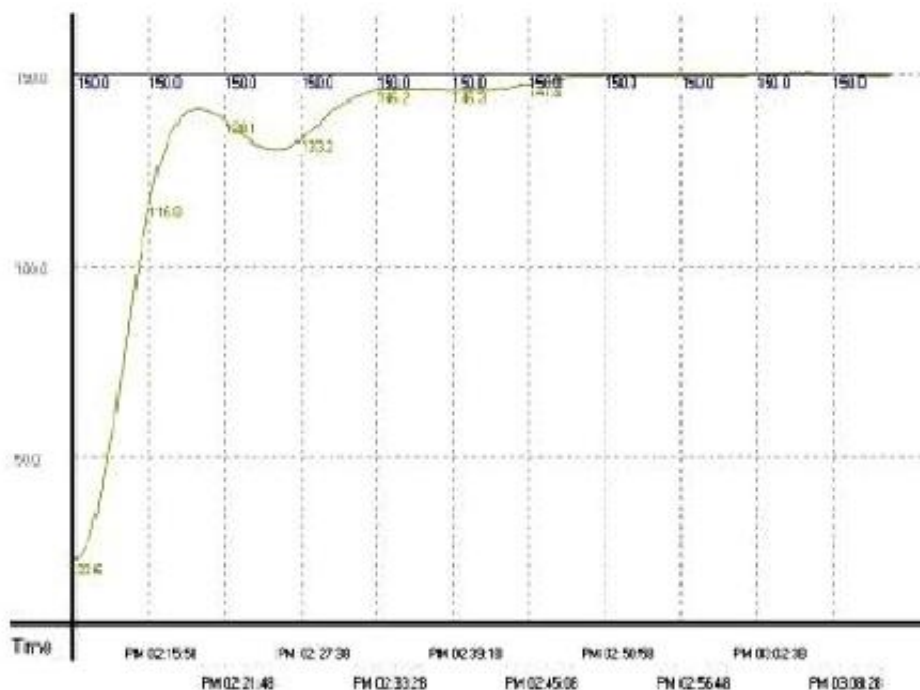
همانطور که در دیاگرام بالا مشاهده می‌کنید، بعد از گذشت 48 دقیقه، دمای واقعی کوره به دمای مطلوب (هدف) با دقت $\pm 1^\circ\text{C}$ و فراجهش 10°C نسبت به دمای هدف رسیده است. به علت تجاوز دمای واقعی از دمای هدف، در این مرحله سرعت گرم کردن کوره را به $D13=K16$ (b4=1) (گرم کردن سریع محیط تحت کنترل) تغییر دادیم. نتیجه بدست آمده از این تغییر را ثبت کردیم. منحنی کنترل دمای بدست آمده با لحاظ این تغییر مطابق با دیاگرام زیر بود:



همانطور که در دیاگرام بالا ملاحظه می‌کنید، در این شرایط، دمای واقعی کوره از دمای مطلوب تجاوز نکرده و مدت زمان رسیدن دمای واقعی به دمای هدف به 1 ساعت و 15 دقیقه با دقت $\pm 1^{\circ}\text{C}$ افزایش یافته است. این تنظیمات برای کنترل دما مناسب است. اما زمان نمونه‌برداری خیلی زیاد است که این امر باعث افزایش زمان گرم کردن و رسیدن دمای واقعی کوره به دمای هدف شده است.

بنابراین، در این مرحله زمان نمونه‌برداری را به 2 ثانیه تغییر دادیم و پارامترها را برابر با $D_{12}=K_{20}$ و $(2s)$ و $D_{30}=K_{2000} (=D_{12} \times 100)$ تنظیم کردیم. نتیجه بدست آمده بعد از تغییر تنظیمات زمان نمونه‌برداری که ثبت شد، مطابق با شکل زیر می‌باشد:

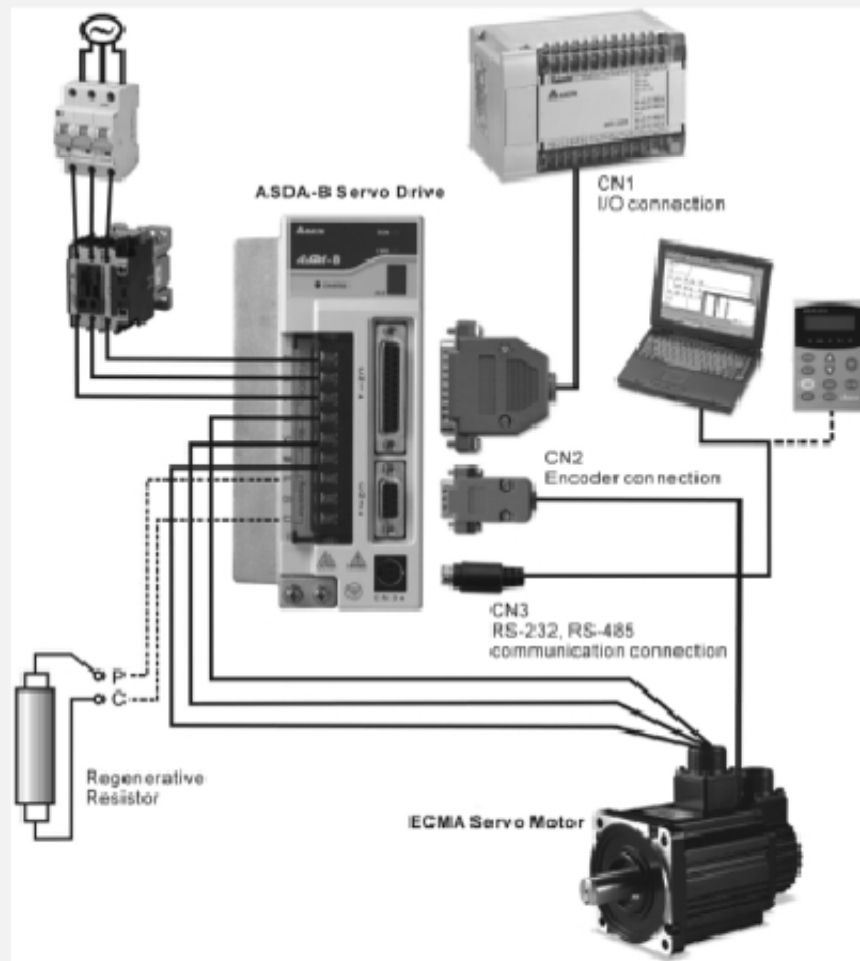




همانطور که در شکل بالا مشاهده می‌کنید، کل زمان کنترل جهت رسیدن دمای واقعی کوره به دمای هدف برابر با 37 دقیقه شده و همچنین دمای واقعی نه دارای نوسان و نه دارای فرابیش (بالا رفتن دمای واقعی از دمای تنظیم شده) می‌باشد. بنابراین، این تنظیمات در این مثال برای ما راضی کننده می‌باشد.

فصل 6

کنترل موقعیت توسط PLC دلتا

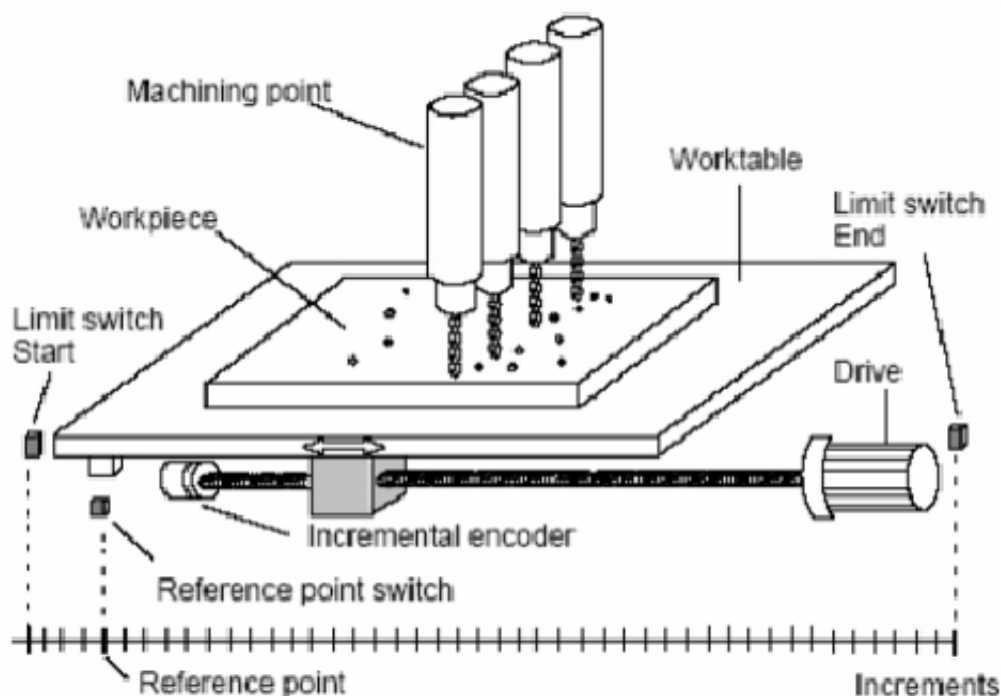


6-1 کنترل موقعیت محور بصورت حلقه باز

برای کنترل سرعت به صورت حلقه باز دو روش وجود دارد:

روش اول ← خروجی آنالوگ برای کنترل سرعت درایو

روش دوم ← خروجی دیجیتال برای کنترل سرعت در حالتی که موتور بصورت کنتاکتور کنترل شود. در کنترل موقعیت نیاز به یک نقطه رفرنس است تا بقیه موقعیت‌ها نسبت به آن سنجیده شوند. این نقطه توسط انگ‌در با قطاری از پالس مشخص می‌گردد. برای فهم بهتر موضوع به ذکر مثالی می‌پردازیم. همانطور که در شکل زیر مشاهده می‌شود، قطعه‌ای لازم است که توسط ماشین ابزار سوراخ کاری شود. قطعه روی میز کار قرار می‌گیرد و سپس میز کار توسط موتور و یک شافت مارپیچ بصورت افقی حرکت می‌کند. ابتدا و انتهای حرکت میز توسط دو لیمیت سوئیچ کنترل می‌شود. نقطه رفرنس نیز توسط یک سنسور مشخص می‌گردد. وقتی که این سنسور یک لبه مثبت بفرستد، نشان دهنده آن است که میز در نقطه رفرنس قرار دارد. انگ‌دري که روی شافت مارپیچ قرار دارد، در نقطه مبنا پالس‌هایی را می‌فرستد که بعنوان مبنا تلقی می‌شود.



باید توجه داشت که دقت در نقطه مبنا بستگی به سرعت حرکت میز و نوع سنسور استفاده شده دارد. عبارات دیگر، ممکن است میز در رفت و برگشت‌های متوالی در نقطه رفرنس‌های مختلف با یک اختلاف

جزئی بایستد. هر چه دقت تکرار سنسور (Repeat Accuracy) بالاتر بوده و سرعت در نزدیکی این نقطه کمتر باشد، اختلاف فوق جزئی‌تر خواهد بود. برای سوئیچ‌های مکانیکی دقت تکرار 10 میکرومتر و برای سوئیچ‌های نوری دقت تکرار 500 میکرومتر است.

از نیازهای دیگر ماشین کاری مد دستی یا Jog Mode است. این مد دو کاربرد مهم دارد. اول اینکه اپراتور در صورت لزوم مثلاً در صورت وقوع اشکال توسط مد دستی میز را به موقعیت مورد نظر هدایت کند. دومین مسئله، سنکرون‌سازی است که برای PLC مهم می‌باشد زیرا وقتی CUP استارت شود، نمی‌تواند موقعیت را محاسبه کند. چون هنوز موقعیت رفرنس را نمی‌شناسد تا سایر موقعیت‌ها را نسبت به آن محاسبه کند. از اینرو، نیاز به شناخت رفرنس دارد که اصطلاحاً به آن سنکرون‌سازی گفته می‌شود. برای سنکرون‌سازی لازم است که محور در حالت Jog Mode حرکت کند تا به نقطه رفرنس برسد و از آنجا سیگنال رفرنس به CUP داده شود.

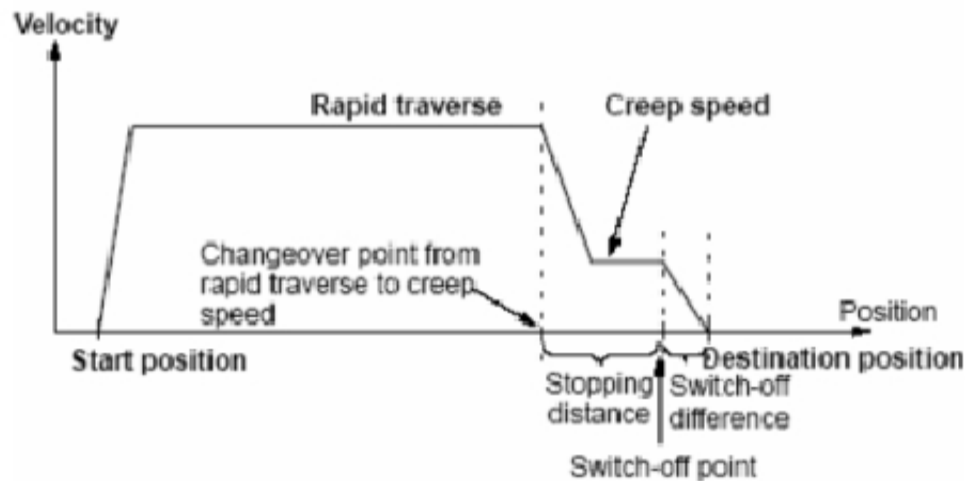
برای سنکرون‌سازی لازم است تعیین شود که حرکت نسبت به نقطه رفرنس Forward یا Backward باشد. این تنظیم توسط CPU انجام می‌شود.

سرعت حرکت محور در حالت Jog Mode قابل تنظیم بوده و شبیه حالت کار عادی محور است. بطور کلی کنترل سرعت بستگی به مدار تغذیه موتور دارد. ممکن است تغذیه بصورت کنتاکتوری باشد یا اینکه از درایو با کنترل فرکانس (برای موتور آسنکرون یا سنکرون) استفاده شده باشد. پس دو روش کنترل سرعت حلقه باز قابل تعریف است:

روش 1 ← روش دو سرعتی که به Rapid Traverse (دور بالا) و Creep Speed (دور پایین) موسوم است.

روش 2 ← استفاده از مبدل فرکانس (اینورتر) برای کنترل سرعت در رنج دلخواه.

در روش 1 که سیستم بصورت کنتاکتوری است، روش کار مطابق با نمودار زیر می‌باشد. این نمودار روش کار را هم برای کار عادی و هم برای حالت Jog Mode نشان می‌دهد. فرض کنیم نقطه مقصد که قرار است محور تا آنجا حرکت کند و بایستد، مشخص است. در اینصورت، ابتدا محرک با بیشترین سرعت از نقطه شروع حرکت می‌کند که این حالت همان Rapid Traverse است. سپس در فاصله مشخصی نسبت به مقصد که این فاصله از قبل مشخص شده است، محرک به حالت سرعت کم یعنی Creep Speed در می‌آید. کمی مانده به نقطه مقصد محرک قطع می‌شود تا نهایتاً محور در موقعیت مورد نظر بایستد. حالت Creep برای دقت بیشتر در رسیدن به نقطه مورد نظر بکار می‌رود.



لازم نیست که نقطه شروع همان نقطه مبدا باشد. از هر نقطه‌ای امکان شروع وجود دارد. تنها محدودیت است که اگر نقطه شروع تا مقصد کوچکتر یا برابر با مسافت Switch Off شدن محرک باشد، عملاً کنترل موقعیت اعمال نمی‌شود.

6-1-1 روش کنترل Rapid/Creep با خروجی دیجیتال برای کنترل سرعت موتور بصورت کنتاکتوری

در این روش 4 خروجی دیجیتال برای عملکرد کنترل در نظر گرفته شده است:

یک خروجی دیجیتال برای فعال‌سازی حالت Rapid

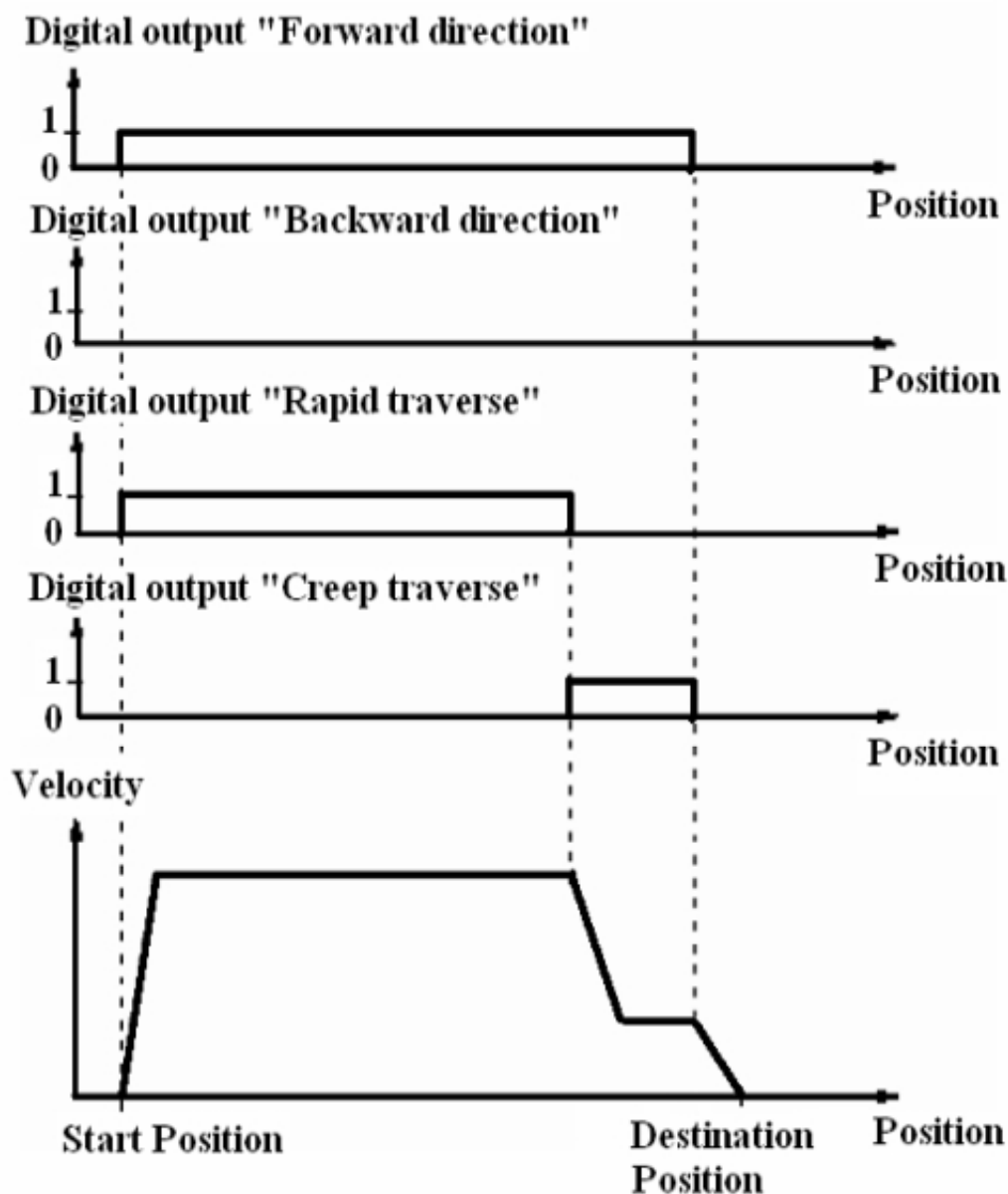
یک خروجی دیجیتال برای فعال‌سازی حالت Creep

یک خروجی دیجیتال برای جهت چرخش Forward

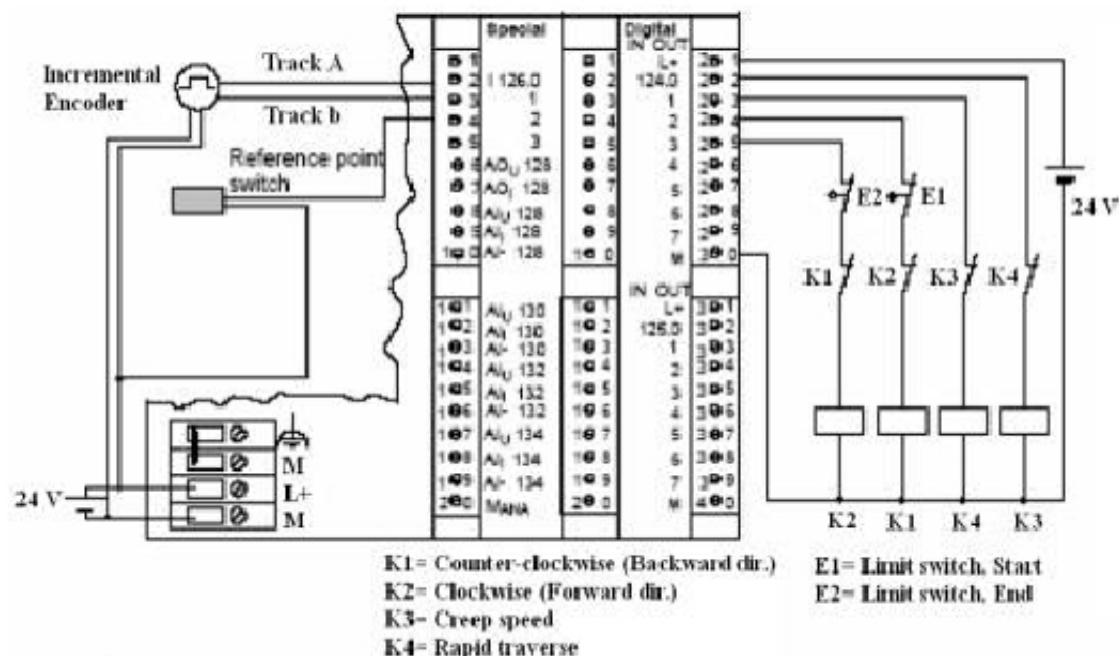
یک خروجی دیجیتال برای جهت چرخش Backward

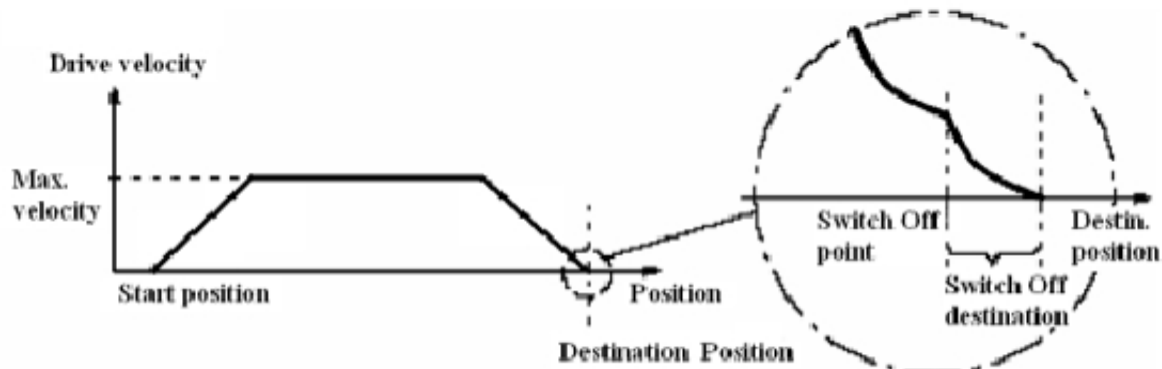
دقت کنید در این نوع کنترل به انگذر هم نیاز می‌باشد.

فرض کنیم لازم است محرک بصورت Forward بچرخد. در این حالت، سیگنال Forward روشن و سیگنال Backward خاموش است. فرض کنیم لازم است محرک بصورت Forward بچرخد. در این حالت سیگنال Forward روشن و سیگنال Backward خاموش است. ابتدا سیگنال Rapid همزمان با Forward فعال شده و به محض رسیدن محور به فاصله معین نسبت به مقصد این سیگنال خاموش شده و سیگنال Creep فعال می‌شود.



از نظر الکتریکی حرکت Forward و حرکت Backward که مربوط به راستگرد و چپگرد موتور هستند، باید با یکدیگر اینترلاک داشته باشند، یعنی با وصل شدن یک کنتاکتور باید کنتاکتور دیگر قطع شود. همین‌طور، کنتاکتورهای حرکت Rapid و حرکت Creep که برای دور تند و دور کند بکار می‌روند نیز باید با یکدیگر اینترلاک داشته باشند.





کنترل مبدل فرکانس (اینورتر) به دو روش انجام می‌شود:

روش 1 ← استفاده از یک خروجی آنالوگ همراه با دو خروجی دیجیتال. سیگنال آنالوگ می‌تواند بین 0-10V یا 0-20mA باشد. این سیگنال سرعت را برای درایو مشخص می‌کند. دو خروجی دیجیتال برای جهت چرخش Forward و Backward بکار می‌رود.

روش 2 ← استفاده از یک سیگنال آنالوگ $\pm 10V$ یا $\pm 20mA$ که در این حالت همزمان سرعت و جهت چرخش مشخص می‌شوند. بدیهی است علامت + برای چرخش Forward و علامت - برای چرخش Backward استفاده می‌شود.

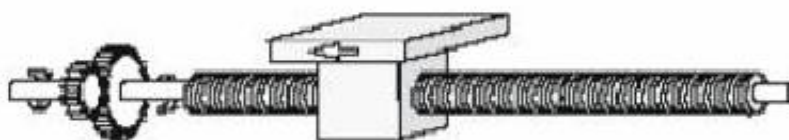
6-2 کنترل موقعیت محور بصورت حلقه بسته

برای کنترل موقعیت حلقه بسته دو روش وجود دارد:

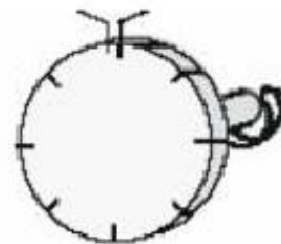
روش 1 ← کنترل موقعیت مبتنی بر خروجی آنالوگ

روش 2 ← کنترل موقعیت مبتنی بر خروجی دیجیتال

با کنترل موقعیت مبتنی بر خروجی آنالوگ می‌توان موتورهای آسنکرون با مبدل فرکانس مجهز به فیدبک توسط انکدر یا سروموتورهای کنترل شده توسط Servo drive را در مد موقعیت کنترل کرد. دو نوع محور قابل کنترل است: Rotary و Liner.



Liner Axis



Rotary Axis

سرعت ماکزیمم بر حسب Pulse/s از حاصل ضرب زیر بدست می‌آید:

$$4 \times (\text{تعداد پالس انگدر در هر دور}) \times (\text{ضریب گیربکس}) \times (\text{سرعت نامی درایو rev/s})$$

بعنوان مثال، اگر دور نامی درایو 3000 rpm (معادل 50 rev/s) و گیربکس نیز وجود نداشته باشد، همچنین تعداد پالس انگدر در هر دور 500 پالس باشد، در این صورت سرعت ماکزیمم برابر است با:

$$50 \times 1 \times 500 \times 4 = 100000 \text{ pulse/s}$$

Creep/Reference Speed : عددی بر حسب پالس بر ثانیه است و می‌تواند تا یک دهم

سرعت ماکزیمم باشد. از این پارامتر برای کاهش سرعت در موقعیت ترمز یا رسیدن به نقطه رفرنس بکار می‌رود.

6-3 لیست PLC های دارای ورودی /خروجی سرعت بالا و آنالوگ و پورت شبکه

« مشخصات PLC های سری DVP-ES2/DVP-EX2

Model name	20EX200T 20EX200R	16ES200T 16ES200R	24ES200T 24ES200R	32ES200T 32ES200R	40ES200T 40ES200R	60ES200T 60ES200R
High-speed input	2 points of 100kHz; 6 points of 10kHz; Max. 8 points for single-phase input; Max. 4 points for 2-phase 2-inputs					
Pulse output	2 points of 100kHz; 2 points of 10kHz					
High-speed comparison interruption	8 points					
External input interruption	8 points					
COM port	Built-in 1 RS-232 port, 2 RS-485 ports					
Built-in analog I/O	Yes	No				
AC motor drive/ Servo drive control commands	Yes					
Extension module connection	Connectable to 8 analog extension modules					
Motion control instructions	Yes, with S-curve acceleration/deceleration function					

« مشخصات PLC سری DVP-EC3

خروجی‌های Y1, Y2 وابسته به پالس سرعت بالا حداکثر تا 10KHz می‌باشند. تعداد ورودی خروجی‌های یکپارچه برای انواع این مدل به صورت 10/14/16/20/24/30/32/40/60 می‌باشد که به عنوان مثال، مدل مورد نظر را با نام DVP-10EC3 می‌شناسند.

Built-in High-Speed Counters

1-phase 1 input		1-phase 2 inputs		2-phase 2 inputs	
Sets	Bandwidth	Sets	Bandwidth	Sets	Bandwidth
2/2	20kHz/10kHz	1	20kHz	1	4kHz

مشخصات PLC سری DVP-EH3

این سری از PLC کنترل حرکت عالی را پشتیبانی می‌کند که فرکانس پالس خروجی سرعت بالا به 200KHz می‌رسد و قادر است 4 دستگاه را کنترل کند. 4 شمارنده سرعت بالا با فرکانس 200KHz را پشتیبانی می‌کند.

Built-in 4 Hardware High-Speed Counters

Standard		Hardware high-speed counter					
1-phase 1 input		1-phase 1 input		1-phase 2 inputs		2-phase 2 inputs	
Sets	Bandwidth	Sets	Bandwidth	Sets	Bandwidth	Sets	Bandwidth
8	10kHz	4	200kHz	2/2	200kHz/20kHz	2/2	200kHz/20kHz

مشخصات PLC سری DVP-SV2

این سری از PLC کنترل حرکت عالی را پشتیبانی می‌کند که فرکانس پالس خروجی سرعت بالا به 200KHz می‌رسد و قادر است 4 دستگاه را کنترل کند. 4 شمارنده سرعت بالا با فرکانس 200KHz را پشتیبانی می‌کند.

Built-in 4 Hardware High-Speed Counters

Standard		Hardware high-speed counter					
1-phase 1 input		1-phase 1 input		1-phase 2 inputs		2-phase 2 inputs	
Sets	Bandwidth	Sets	Bandwidth	Sets	Bandwidth	Sets	Bandwidth
8	10kHz	4	200kHz	2/2	200kHz/20kHz	2/2	200kHz/20kHz

مشخصات PLC سری DVP-SS2

این سری از PLC پروتکل MODBUS ASCII/RTU را ساپورت می‌کند. توابع کنترل حرکت را می‌توان برای 4 دستگاه با خروجی پالس 10KHz و 8 دستگاه ورودی شمارنده سرعت بالا با فرکانس (4 نقطه 20KHz و 4 نقطه 10KHz) اعمال کرد.

Built-in High-Speed Counters

1-phase 1 input		1-phase 2 inputs		2-phase 2 inputs	
Sets	Bandwidth	Sets	Bandwidth	Sets	Bandwidth
4/4	20kHz/10kHz	2	20kHz	2/2	10kHz/5kHz

مشخصات PLC سری DVP-SA2

این سری از PLC ها پروتکل MODBUS ASCII/RTU را ساپورت می کنند. توابع کنترل حرکت را می توان برای 4 دستگاه با خروجی پالس سرعت بالا که 2 دستگاه با فرکانس 100KHz و 2 دستگاه با فرکانس 10KHz کار می کنند و همچنین، 8 دستگاه ورودی شمارنده سرعت بالا که 2 دستگاه با فرکانس 100KHz و 6 دستگاه با فرکانس 10KHz و نیز 1 گروه پالس فازی A/B phase با فرکانس 50KHz کار می کنند، اعمال کرد.

Built-in High-Speed Counters

1-phase 1 input		1-phase 2 inputs		2-phase 2 inputs	
Sets	Bandwidth	Sets	Bandwidth	Sets	Bandwidth
2/6	100kHz/10kHz	2	100kHz	1/3	50kHz/5kHz

« مشخصات PLC سری DVP-SX2

این سری از PLC پروتکل MODBUS ASCII/RTU را ساپورت می کند. توابع کنترل حرکت را می توان برای 4 دستگاه با خروجی پالس سرعت بالا که 2 دستگاه با فرکانس 100KHz و 2 دستگاه با فرکانس 10KHz کار می کنند و همچنین، 8 دستگاه ورودی شمارنده سرعت بالا که 2 دستگاه با فرکانس 100KHz و 6 دستگاه با فرکانس 10KHz کار می کنند، اعمال کرد. این سری دو محور خطی و ترکیبی قوس دار از سروسیستم را ساپورت می کند.

« مشخصات PLC سری DVP-PM

این سری از PLC برای کنترل حرکت از قطار پالس استفاده می نماید.

- مدل DVP 10PM00M کنترل حرکت عمومی را اجرا می کند. این سری می تواند حرکت ترکیبی خطی برای 2، 3، 4، 5 یا 6 محور را کنترل کند. خروجی PWM با دقت بسیار عالی تا 0.3% و فرکانس 200KHz را ساپورت می کند. از پروتکل MODBUS ASCII/RTU برای شبکه کردن وسایل استفاده شده است. توابع کنترل حرکت در سری DVP 10PM00M به شرح زیر است:

خروجی پالس سرعت بالا : کنترل 6 دستگاه با خروجی های فازی A/B phase

کنترل 6 دستگاه شمارنده سرعت بالا و سخت افزار فیلتر دیجیتال برای شمارنده ها تابع کنترل حرکت تک محوره (اولین سرعت، دومین سرعت و تابع MPG را ساپورت می کند) دارای تابع چرخ دنده الکترونیکی است.

- مدل DVP 20PM00D/M کنترل حرکت حرفه ای را ساپورت می کند. این سری از PLC سازگاری کامل با کدهای G و کدهای M، [(G-Code) , (M-Code)] دارد و همچنین، از حرکت 3 محور با حرکت ترکیبی خطی، قوس دار و یا مارپیچ پشتیبانی می کند. در این سری از PLC ها می توان از تابع بادامک الکترونیکی با دقت 2048 نقطه برای کاربردهای از قبیل ماشین برش Flying shear و ماشین برش Rotary cut استفاده کرد. توابع کنترل حرکت در سری DVP 20PM00D/M به قرار زیر است :

سیگنال خروجی تناسبی فاز A/B phase : حداکثر 2 دستگاه برای مدل DVP 20PM00D و حداکثر 3 دستگاه برای DVP 20PM00M که فرکانس خروجی تناسبی 500KHz می باشد. پشتیبانی از ورودی های MPG

تابع کنترل حرکت تک محوره (اولین سرعت، دومین سرعت و تابع MPG را ساپورت می کند) دارای تابع چرخ دنده الکترونیکی است.

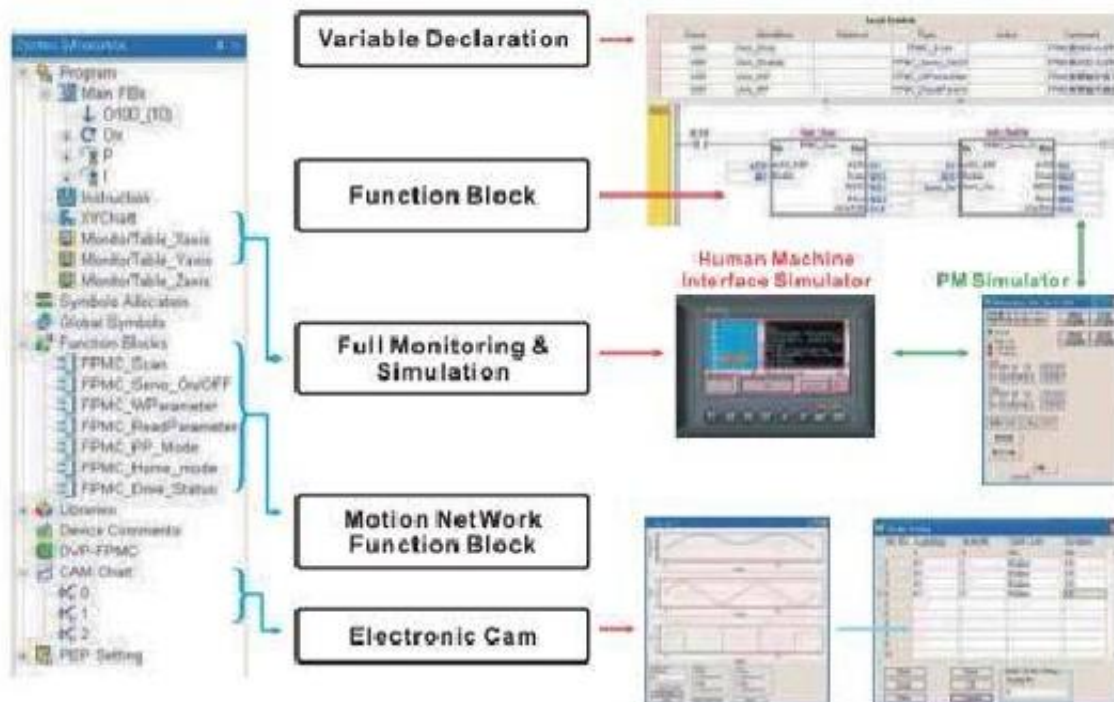
ماژول کنترل حرکت با قطار پالس AH05PM / AM10PM



« ماژول کنترل حرکت ارتباطات شبکه DMCNET در مدل AH20MC



نرم افزار PM Soft برای ویرایش G-Code بوده و همچنین، شبیه ساز خط مسیر حرکت، دستورالعمل فرمان حرکت موقعیتیابی و برقراری بادامک الکترونیکی می باشد.



- در نرم افزار PM Soft توابع کنترل حرکت یک محوره به شرح زیر است :

تابع T_AbsSeg1_U1

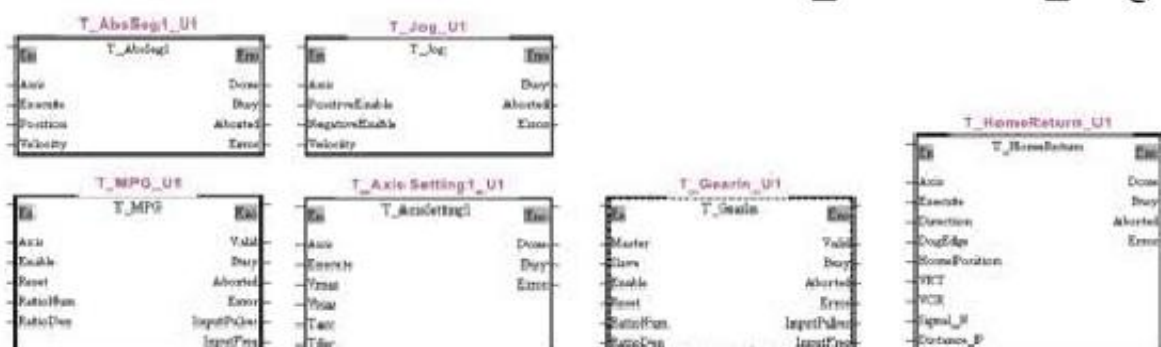
تابع T_Jog_U1

تابع T_MPG_U1

تابع T_Axis Setting1_U1

تابع T_Gearin_U1

تابع T_HomeReturn_U1

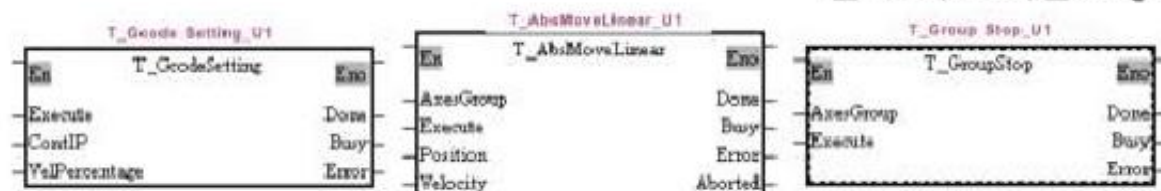


- در نرم افزار PM Soft توابع کنترل حرکت چند محوره به شرح زیر است :

تابع T_Gcode Setting_U1

تابع T_AbsMoveLinear_U1

تابع T_Group Stop_U1



- در نرم افزار PM Soft توابع کنترل حرکت بادامک الکترونیکی به شرح زیر است :

تابع T_CamIn_U1

تابع T_CamCurve_U1

تابع T_CamSyncRatio_U1

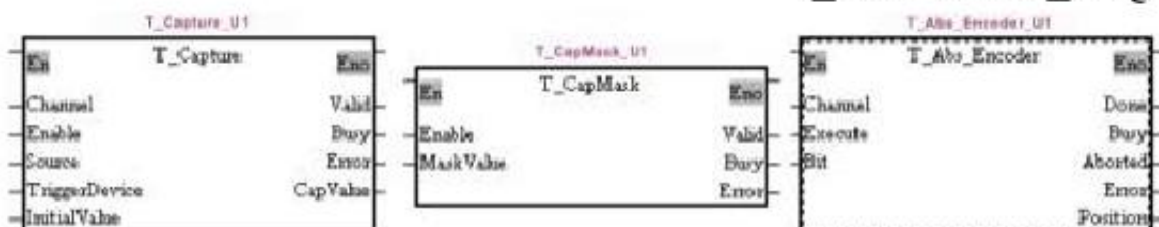


- در نرم افزار PM Soft دیگر توابع کنترل حرکت به شرح زیر است:

تابع T_Capture_U1

تابع T_CapMask_u1

تابع T_Abs Encoder_U1



نکته:

توجه داشته باشید که برای ویرایش ارتباطات شبکه بین PLC و سرور دیو با پروتکل MODBUS باید از نرم افزار SCM Soft استفاده نمایید.

6- 4 مشخصات ورودی خروجی سرعت بالا و توابع کنترل حرکت



ورودی سرعت بالا

در مدل DVP-ES2 هشت ترمینال ورودی سرعت بالا با فرکانس کاری 2 نقطه 100KHz شش نقطه با فرکانس 10KHz ساپورت می‌شود. مدهای شمارش ساپورت شده شامل A/B , U/D , U/Dir می‌باشند.

Counting mode	Counting pulse	
	Up (+1)	Down (-1)
1-phases 2 inputs (U/D, DIR)		
1-phase 2 inputs (U, D)		
2-phase 2 inputs (A, B)		

تنظیم مد شمارش A/B با فرکانس دوپل یا فرکانس 4 برابر در رجیستر مشخص D1022 صورت می‌گیرد.

D1022	Counting diagram
Double frequency (k2)	
4 times frequency (Default: k4)	

لیست سخت افزار شمارنده‌های سرعت بالا به شرح زیر است :

Input	1-phase 1 inputs		1-phase 2 inputs						2-phase 2 inputs			
	C243	C244	C245	C246	C247	C248	C249	C250	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

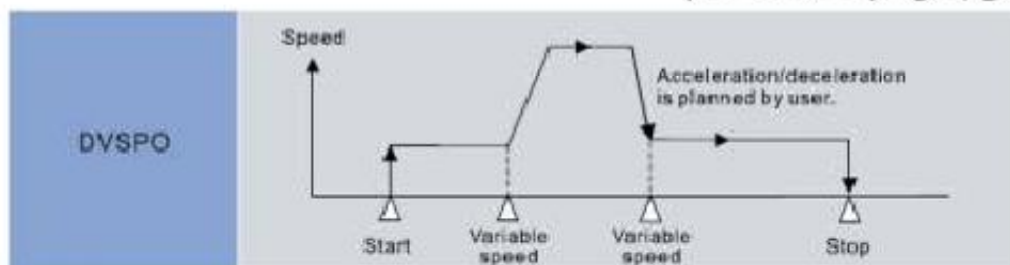
خروجی پالس

در مدل DVP-ES2 چهار محور با خروجی پالس که دو نقطه با حداکثر فرکانس 100KHz و دو نقطه با حداکثر فرکانس 10KHz و خروجی پالس با مدهای Pulse/Dir , A/B , CW/CCW ساپورت می‌شود.

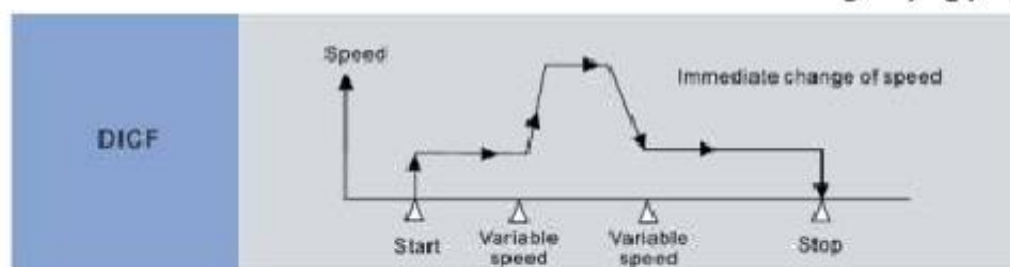
Input	D1220					D1221				
	K0		K1	K2	K3	K0		K1	K2	K3
Y0	Pulse		Pulse	A	CW					
Y1		Pulse	Dir	B						
Y2						Pulse		Pulse	A	CCW
Y3							Pulse	Dir	B	

توابع کنترل حرکت برای خروجی پالس سرعت بالا (HSP) به شرح زیر است:

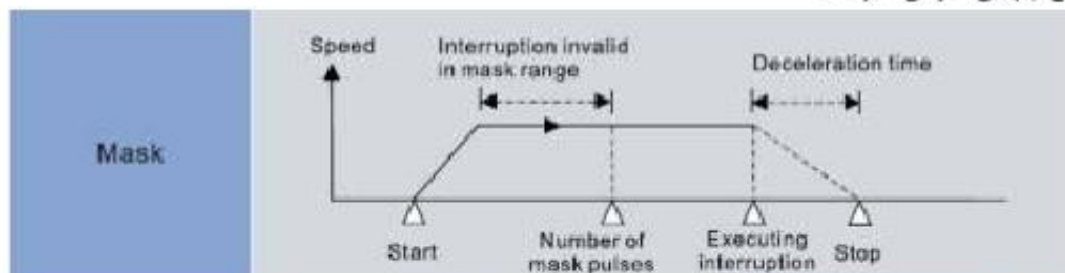
- خروجی پالس سرعت بالای متغیر



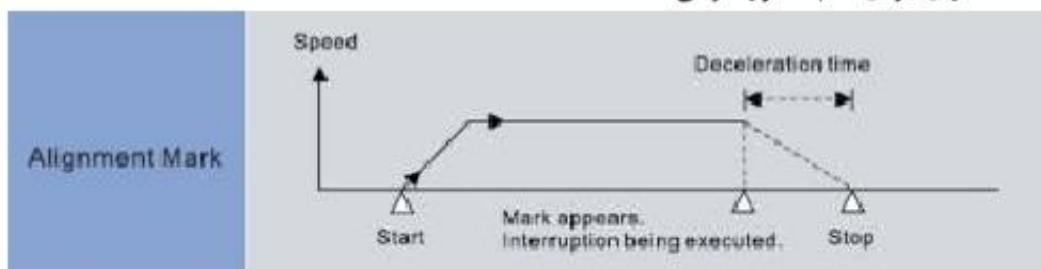
- تغییر سریع فرکانس



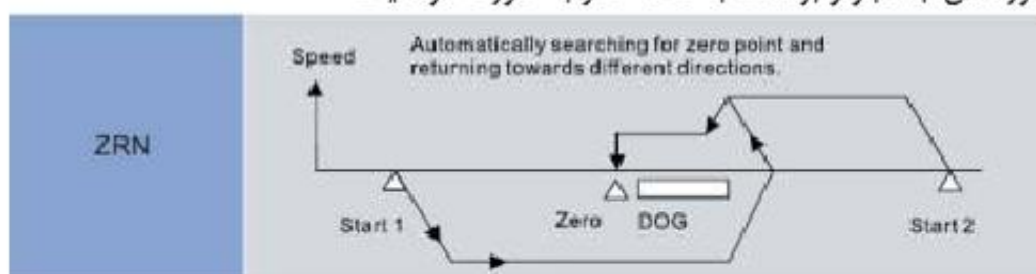
- تابع پنهان کردن حرکت



- علامت‌گذاری برای هم محور کردن



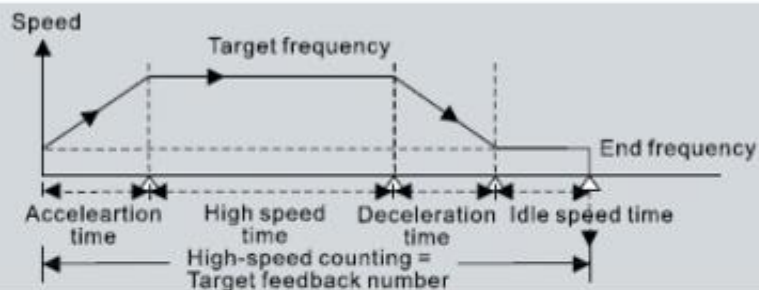
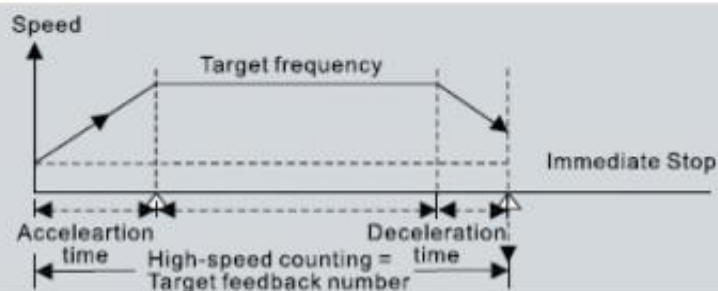
- دستورالعمل جستجو و برگشت به نقطه صفر به صورت اتوماتیک



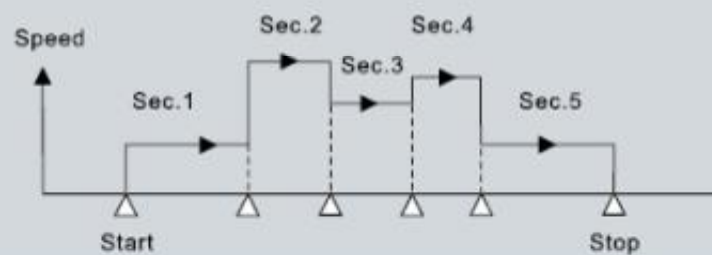
کنترل حلقه بسته موقعیتیابی که در اینجا از انکدر سرودرایو برای ارسال پالس سرعت بالا به PLC برای فیدبک موقعیت استفاده می‌شود.



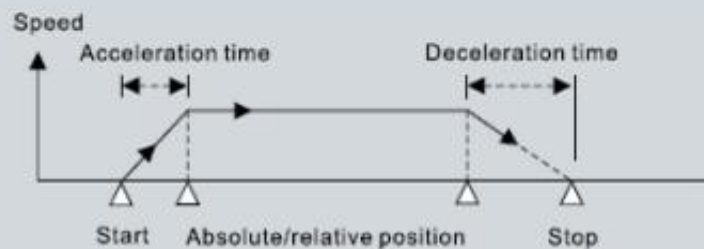
در این نوع برنامه، دستورالعمل موقعیتیابی حلقه بسته، سیگنال فیدبک از انکدر را با تعداد پالس خروجی از PLC مقایسه کرده و تصمیم می‌گیرد که تعداد پالس خروجی را با تعداد پالس ورودی از انکدر یکسان نموده و تفاوت بین پالس‌ها را در پالس خروجی جبران کند.

DCLLM**DCLLM**

- خروجی پالس فهرست شده

DPTPO

- موقعیت یابی وابسته/مطلق

DDRVA/DDRVI

6-5 جدول نام ترمینال‌های PLC برای سیم‌بندی ورودی/خروجی‌ها

ورودی سرعت بالا برای اتصال انگ‌در

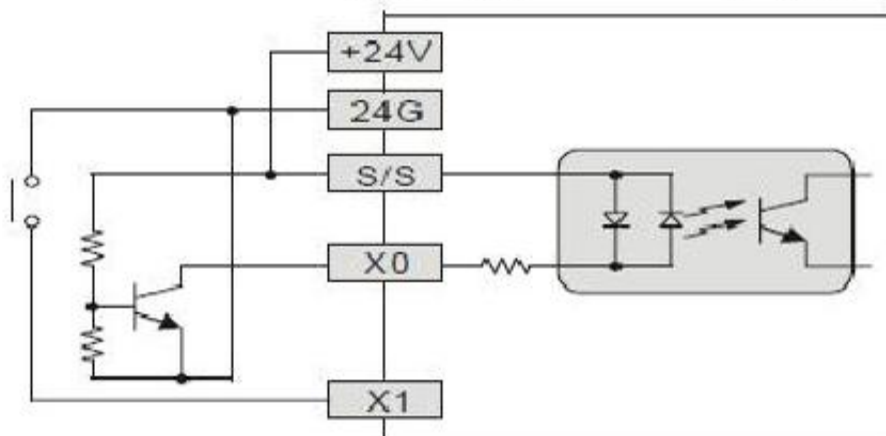
مدل		حداکثر ورودی		محدوده پاسخ	توضیحات	نام ترمینال
20M	20D/ 20DT	ولتاژ	جریان			
*	*	24 V	6 mA	10 ms	فعال کردن ورودی	START0, START1
*	*	24 V	6 mA	10 ms	غیر فعال کردن ورودی	STOP0, STOP1
*	*	24 V	6 mA	10 ms	ورودی محدود کننده چپ و راست	LSP0/LSN0, LSP1/LSN1
*	-	24 V	6 mA	10 ms	ورودی محدود کننده چپ و راست (LSP2/LSN2)	X1/X2 (for Z axis)
*	*	5 to 24 V	15 mA	200 kHz	ورودی پالس +/- فاز A (MPG) سیگنال ورودی تفاضلی	A0+, A0-, A1+, A1- (shared by Y/Z axis)
*	*	5 to 24 V	15 mA	200 kHz	ورودی پالس +/- فاز B (MPG) سیگنال ورودی تفاضلی	B0+, B0-, B1+, B1- (shared by Y/Z axis)
*	*	5 to 24 V	15 mA	200 kHz	سیگنال ورودی نقطه صفر سیگنال ورودی تفاضلی	PG0+, PG0-, PG1+, PG1-
*	-	24 V	6 mA	10 ms	سیگنال ورودی نقطه صفر (PG2)	X3 (for Z axis)
*	*	24 V	6 mA	1 ms	دو مد عملکرد متفاوت وابسته به تغییرات وجود دارد: 1: سیگنال DOG وقتی که صفر برمی گردد 2: وارد کردن سیگنال فعال ساز در اولین یا دومین سرعت	DOG0, DOG1
*	-	24 V	6 mA	10 ms	مانند DOG0, DOG1 (DOG2)	X0 (for Z axis)

خروجی سرعت بالا برای اتصال به سرودرایو

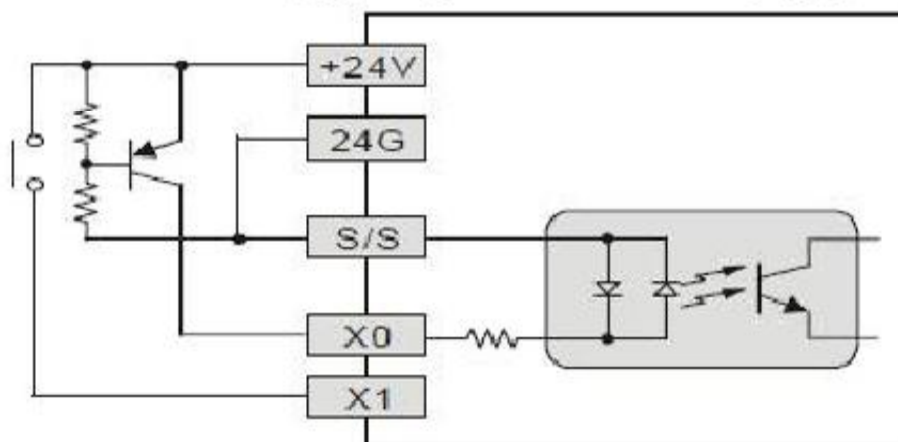
مدل		حداکثر جریان ورودی	محدوده پاسخ	نام ترمینال
20M	20D/20DT			
*	*	20 mA	10 ms	CLR0+, LR0-, CLR1+, CLR1-
*	*	30 mA	10 ms	Y2 (for Z axis)
*	*	40 mA	500 kHz	FP0+, FP0-, FP1+, FP1-
*	-	40 mA	500 kHz	FP2+, FP2-
*	*	40 mA	500 kHz	RP0+, RP0-, RP1+, RP1-
*	-			RP2+, RP2-

6-6 سیم‌بندی ترمینال‌های PLC دلتا

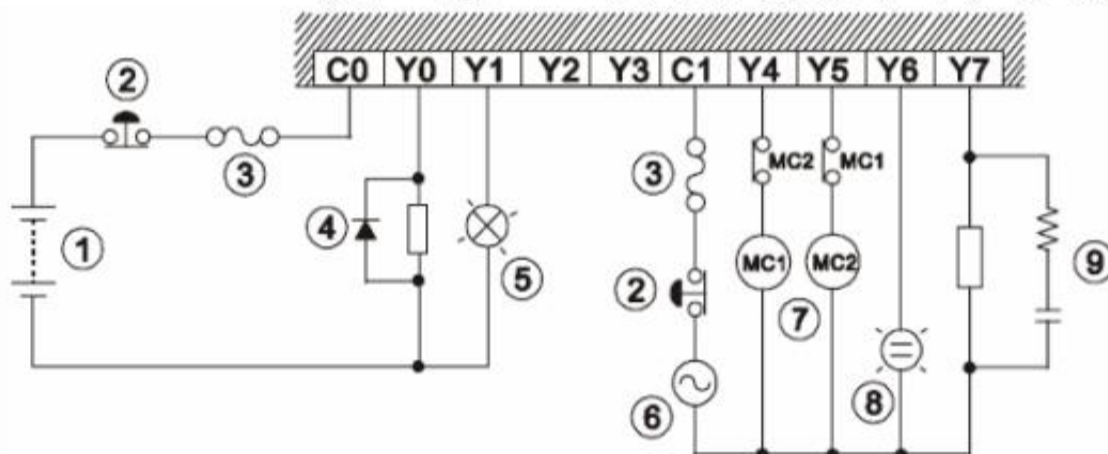
سیم‌بندی سیگنال ورودی با مد SINK مطابق با شکل زیر است:



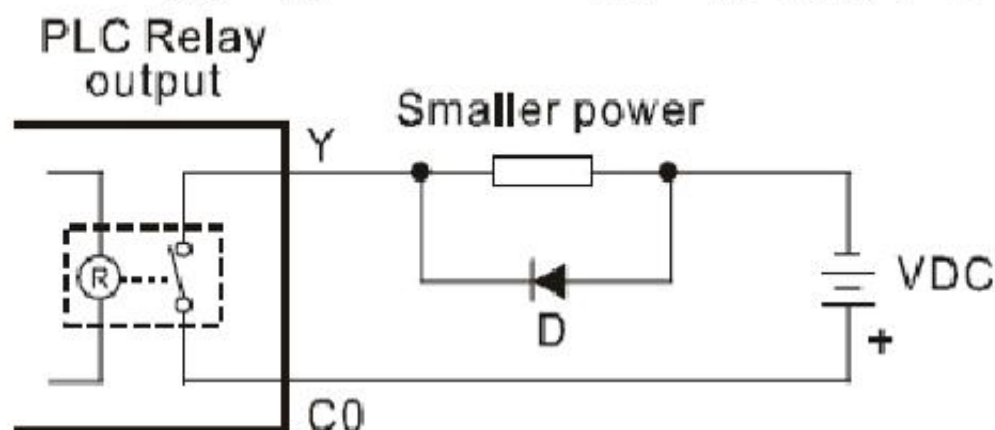
سیم‌بندی سیگنال ورودی با مد SOURCE مطابق با شکل زیر است:



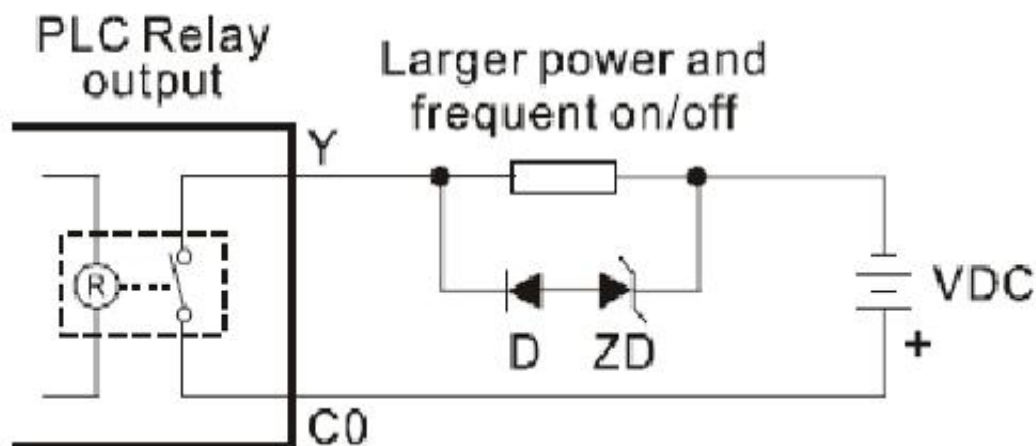
سیم‌بندی مدار سیگنال خروجی رله‌ای (R) مطابق با شکل زیر است:



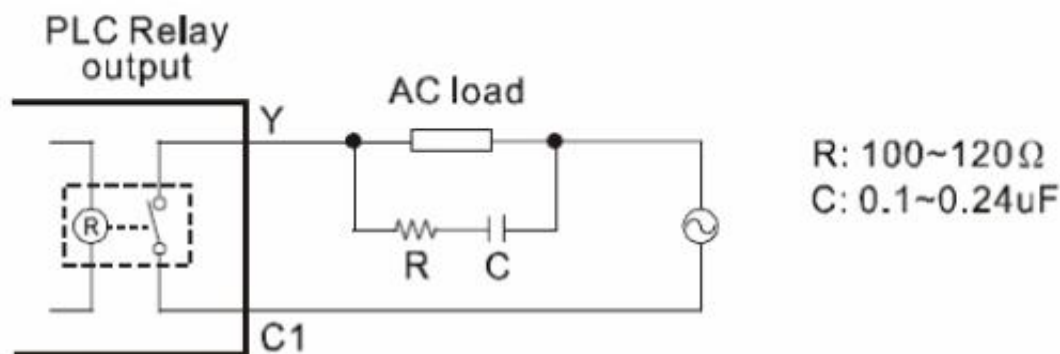
سیم‌بندی مدار خروجی رله‌ای با اتصال دیود D: 1N4001 مطابق با شکل زیر است:



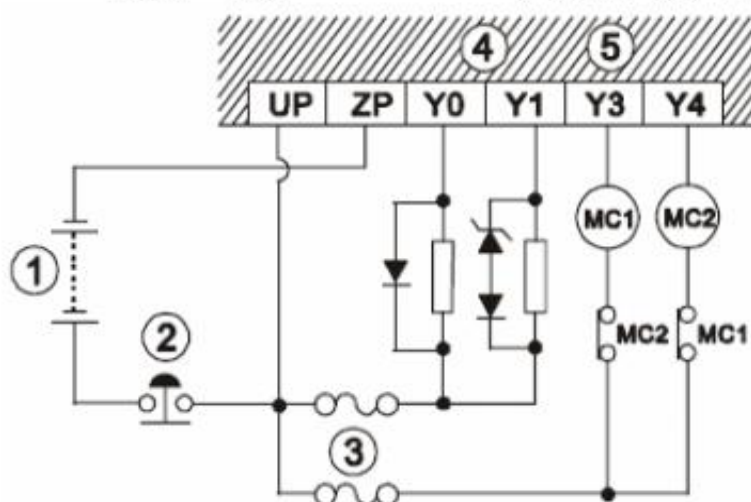
سیم‌بندی مدار خروجی رله‌ای با اتصال دیود ZD : 9V Zener, 5W و دیود زنر D: 1N4001 مطابق با شکل زیر است:



سیم‌بندی مدار خروجی رله‌ای برای بارهای متناوب AC مطابق با شکل زیر است:

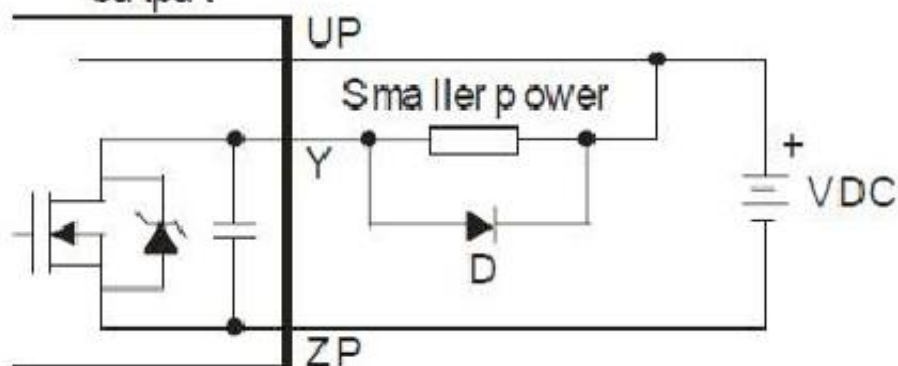


مدار سیگنال خروجی ترانزیستوری (T) مطابق با شکل زیر است :



سیم‌بندی مدار خروجی ترانزیستوری با اتصال دیود D: 1N4001 مطابق با شکل زیر است :

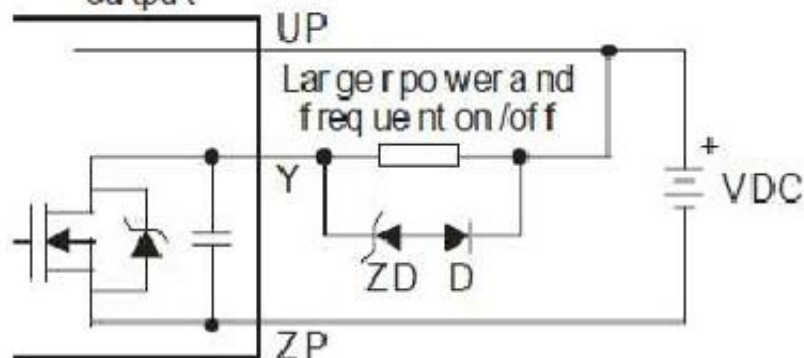
PLC Transistor
output



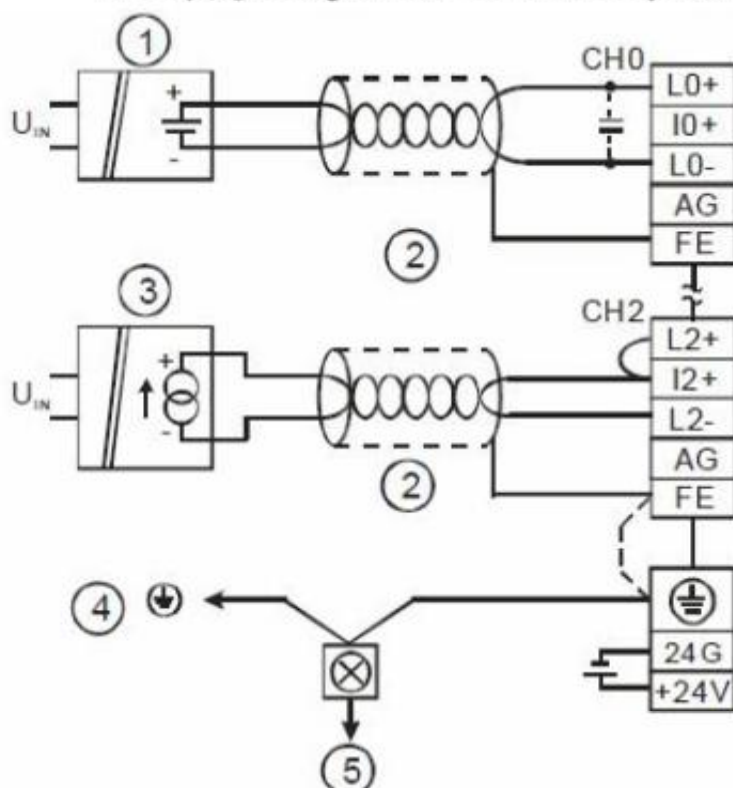
سیم‌بندی مدار خروجی ترانزیستوری با اتصال دیود D: 1N4001 و دیود زنر ZD : 9V Zener

5W مطابق با شکل زیر است :

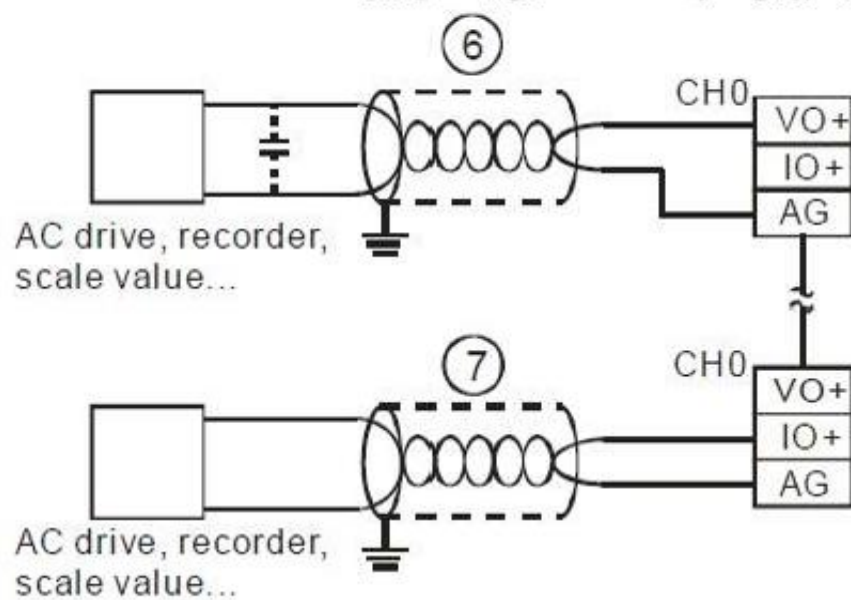
PLC Transistor
output



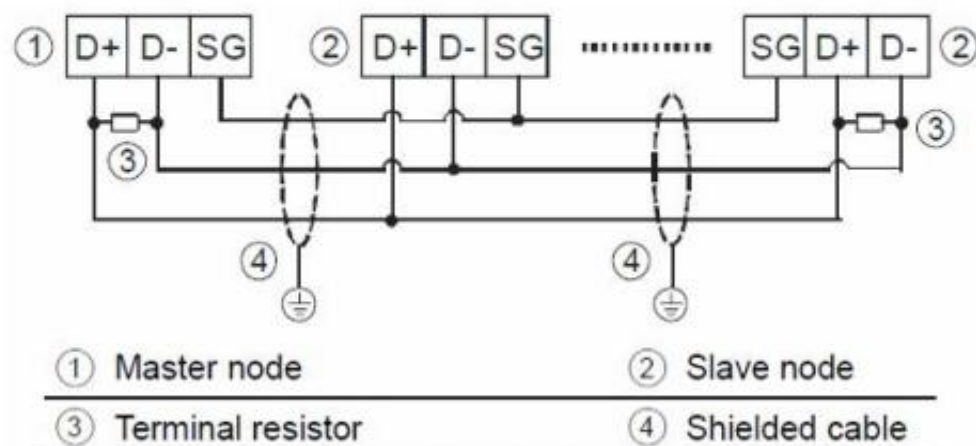
سیم‌بندی مدار ورودی آنالوگ A/D: Active مطابق با شکل زیر است:



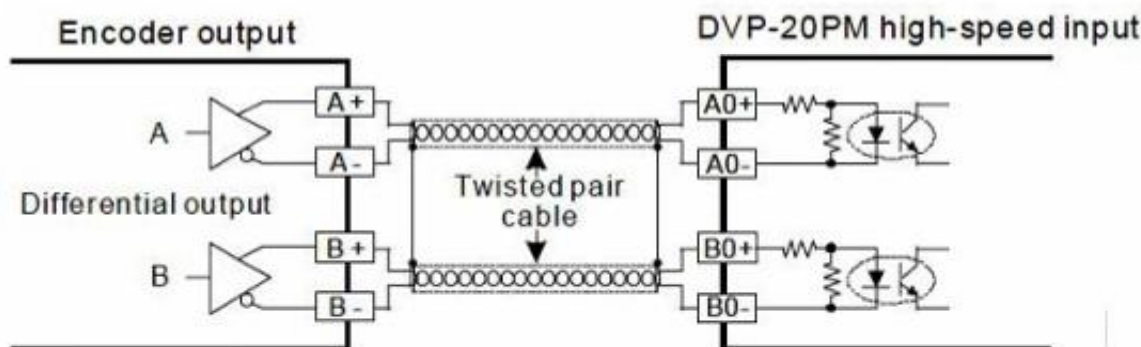
سیم‌بندی مدار خروجی آنالوگ D/A مطابق با شکل زیر است:



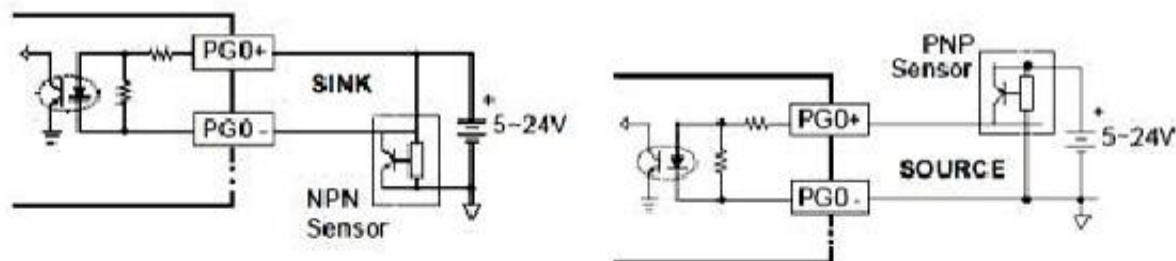
سیم‌بندی ارتباطات سریال RS-485 با مقاومت ترمیناتور 120 اهم در ابتدا و انتهای شبکه مطابق با شکل زیر است:



سیم‌بندی مدار ورودی تفاضلی سرعت بالا (به میزان 200 kHz) در محیط‌های با نویز بالا برای مدار خروجی LINE DRIVE انکدر متصل شده به ورودی PLC مطابق با شکل زیر است:



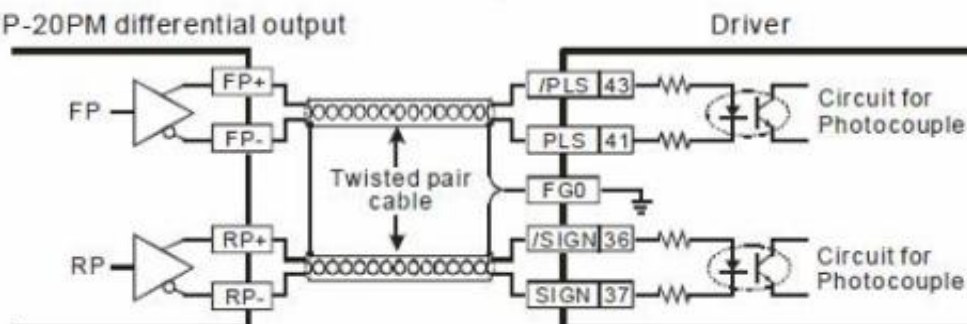
سیم‌بندی مدار ورودی سرعت پایین (کمتر از 50 kHz) در محیط‌های با نویز پایین برای مدار خروجی SINK/SOURCE متصل شده به ورودی PLC مطابق با شکل زیر است:



سیم‌بندی مدار خروجی تفاضلی سرعت بالا برای راه‌اندازی سروسیستم مطابق با شکل زیر است:

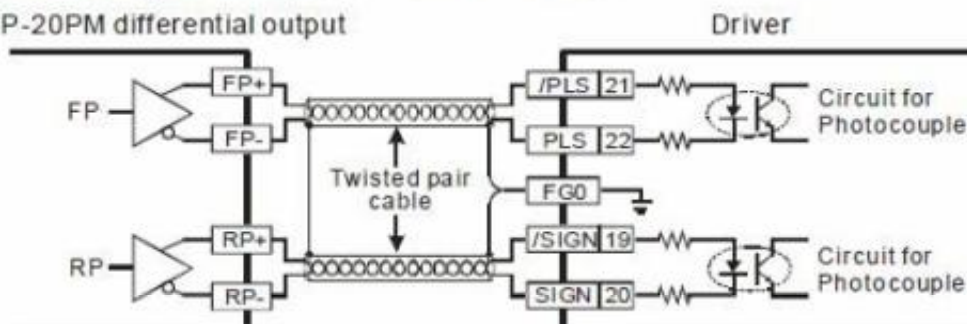
Differential outputs with ASDA-A & A+, ASDA-A2 series driver

DVP-20PM differential output



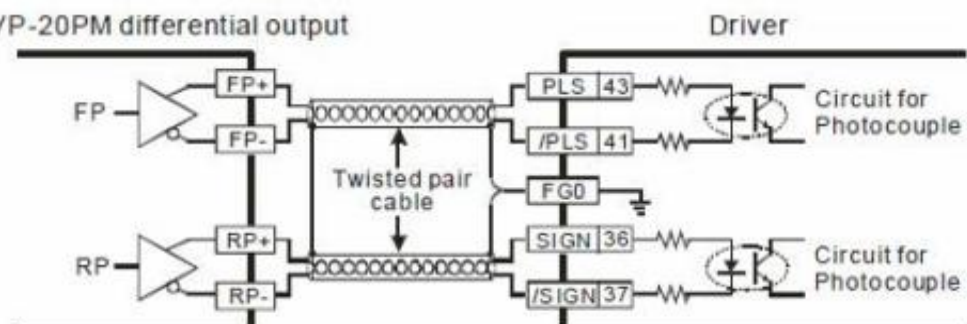
Differential outputs with ASDA-B series driver

DVP-20PM differential output

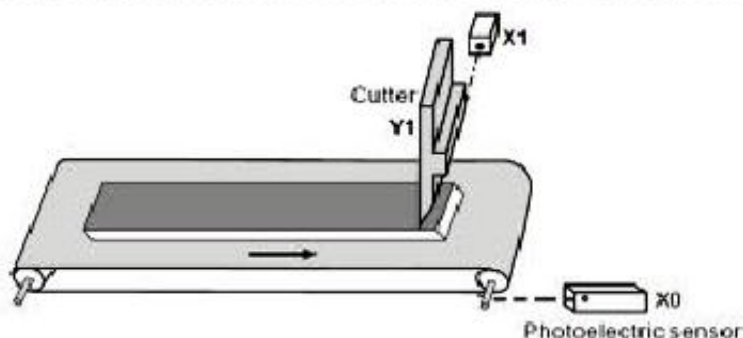


Differential outputs with ASDA-AB series driver

DVP-20PM differential output



6-7 کنترل ماشین برش توسط دستورالعمل DHSCS



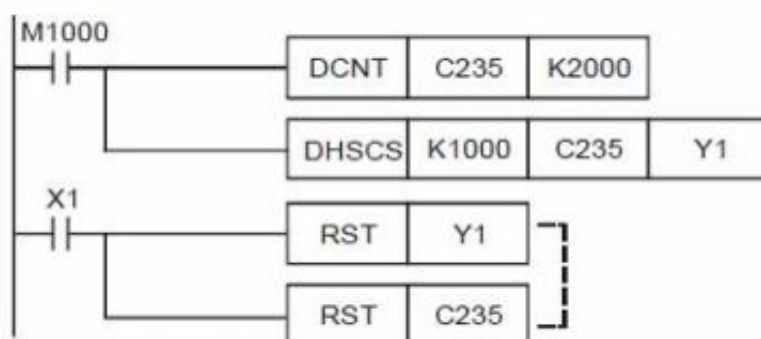
هدف از کنترل در این مثال به شرح زیر است :

هدف این برنامه شمارش تعداد چرخش و کنترل کاتر مطابق با مقدار شمارنده C235 می‌باشد. X0 برای شمارش در هنگام چرخش محور می‌باشد. وقتی که مقدار شمارنده C235 برابر 1000 شد، کاتر فرآیند برش را اجرا می‌کند.

ابزار مورد استفاده در این مثال به شرح زیر است :

ابزار	تابع
X0	سنسور فتوالکتریک، وقتی که محور در حال چرخش است، X0 روشن می‌شود و پالس می‌دهد.
X1	سنسور فتوالکتریک، ورودی X1 در هنگام غیرفعال بودن کاتر فعال ON است. (Y1=OFF)
Y1	کاتر
C235	شمارنده تعداد چرخش محور

برنامه کنترلی این مثال به شرح زیر است :



خط 1 برنامه ← وقتی که PLC در حال کار است، شمارنده سرعت بالای C235 فعال خواهد بود.

خط 2 برنامه ← وقتی که مقدار شمارنده C235 از 999 به 1000 یا از 1001 به 1000 تغییر کرد،

Y1 فعال خواهد شد و خروجی سریعاً تغییر وضعیت می‌دهد.

خط 3 و 4 برنامه ← وقتی که X1=ON شود، کاتر Y1 ریست شده و مقدار شمارنده C235 نیز

پاک خواهد شد.

توضیحات برنامه به شرح زیر است :

از سنسور فتوالکتریک متصل شده به ورودی X0 در برنامه برای ورودی خارجی شمارنده C235

استفاده شده است. وقتی که محور کانوایر شروع به چرخیدن می‌کند، X0 روشن شده و پالس می‌دهد، بنابراین شمارنده C235 شروع به شمارش می‌کند.

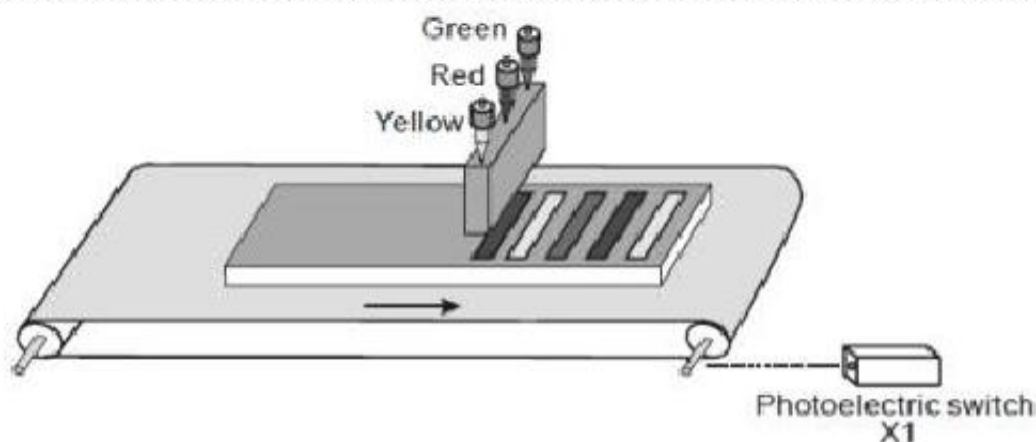
در هنگام اجرای دستورالعمل DHSCS، وقتی که مقدار شمارنده C235 به 1000 رسید (چرخش

محور=1000 بار)، خروجی Y1 فعال (ON) شده و سپس خروجی در ترمینال خروجی خارجی سریعاً

تغییر وضعیت می‌دهد. بنابراین، کاتر فعال خواهد شد.

وقتی که کاتر غیرفعال است، ورودی X1 فعال ON می‌باشد. مقدار شمارنده C235 پاک شده و کاتر Y1 ریست می‌شود. در این موقع، X1 غیرفعال OFF می‌شود. شمارنده C235 برای شمارش بعدی ریست شده و پروسه‌ی بالا مجدداً تکرار خواهد شد.

6-8 کنترل رنگ چندین قسمت توسط دستورالعمل DHSZ/DHSCR

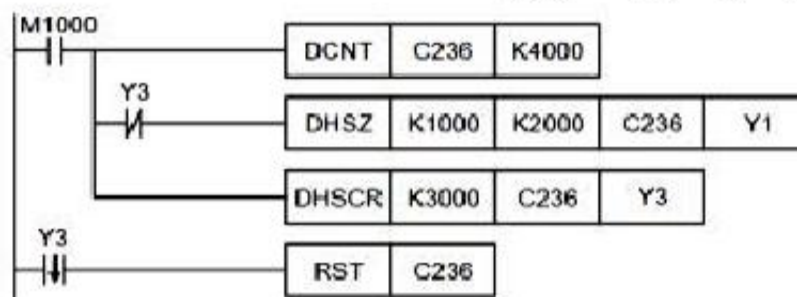


هدف از کنترل در این مثال به شرح زیر است:

رنگ‌های مختلف با ترکیبی از سه رنگ اصلی تولید می‌شوند. این سه رنگ شامل قرمز، زرد و سبز می‌باشند. وقتی که محور کانوایر 1000 بار چرخید، مواد رنگ تغییر خواهد کرد. از این رو پروسه تولید رنگ اجرا شده و رنگ‌های قرمز، زرد، سبز با هم ترکیب خواهند شد. ابزارهای مورد استفاده در این مثال به شرح زیر می‌باشد:

ابزار	تایع
X1	سنسور فتوالکتریک: وقتی که محور چرخید X1 فعال شده و پالس می‌دهد.
Y1	موارد رنگی قرمز برای تولید رنگ
Y2	موارد رنگی زرد برای تولید رنگ
Y3	موارد رنگی سبز برای تولید رنگ
C236	شمارنده برای شمارش تعداد چرخش محور

برنامه کنترلی این مثال مطابق با شکل زیر است:



توضیحات برنامه این مثال به شرح زیر است :

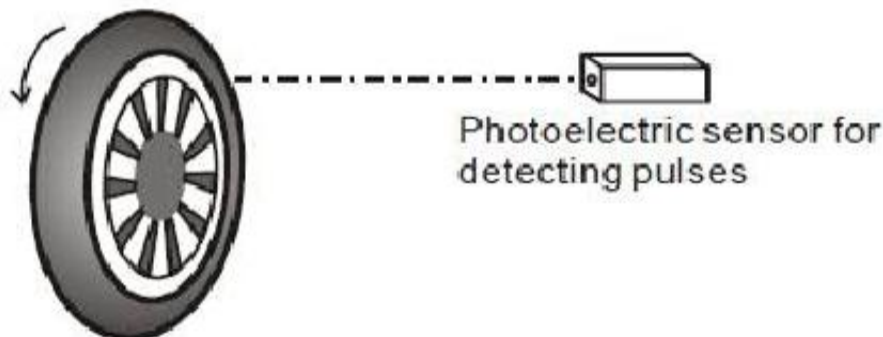
سنسور فتوالکتریک به ورودی X1 متصل شده و این ورودی خارجی به ورودی شمارنده C235 متصل شده است. وقتی که محور کانوایر شروع به چرخیدن می‌کند، X1 روشن شده و پالس می‌دهد. بنابراین، شمارنده C235 شروع به شمارش می‌کند.

وقتی که مقدار جاری PV در شمارنده $K1000 < C236$ (تعداد چرخش محور > 1000) بود، $Y1=ON$ شده و مواد رنگی قرمز شروع به پر کردن جعبه رنگ می‌کند. وقتی که مقدار جاری PV $K1000 \leq C236 \leq 2000$ در شمارنده $2000 \geq C236$ (محور چرخش ≥ 1000) شد، $Y1=OFF$ و سپس $Y2=ON$ می‌شود. مواد رنگی زرد شروع به پر کردن جعبه رنگ می‌کند.

وقتی که مقدار جاری PV $K2000 < C236$ در شمارنده $K3000 < C236$ (> 2000 چرخش محور) شد، $Y1=Y2=OFF$ شده و $Y3=ON$ خواهد شد. مواد رنگی زرد شروع به پر کردن جعبه رنگ می‌کند. وقتی که $Y3=ON$ شود، کنتاکت در حالت عادی بسته (NC) خروجی $Y3$ فعال شده و دستورالعمل DHSZ غیرفعال می‌شود.

وقتی که PV در شمارنده C236 به K3000 رسید، دستورالعمل DHSCR اجرا شده و خروجی $Y3$ ریست می‌شود. مقدار جاری شمارنده C236 به دلیل راه‌اندازی $Y3=ON$ پاک خواهد شد. کنتاکت در حالت عادی بسته (NC) خروجی $Y3$ غیرفعال OFF است و از این رو دستورالعمل DHSZ مجدداً اجرا می‌شود. شمارنده C236 از 0 شروع به شمارش کرده و مواد رنگی مجدداً پر کردن جعبه رنگ را در سیکل مشخص مطابق با قرمز، زرد، سبز، قرمز، زرد، سبز و ... اجرا خواهد کرد.

6- اندازه‌گیری سرعت چرخش چرخ گردان توسط دستورالعمل SPD



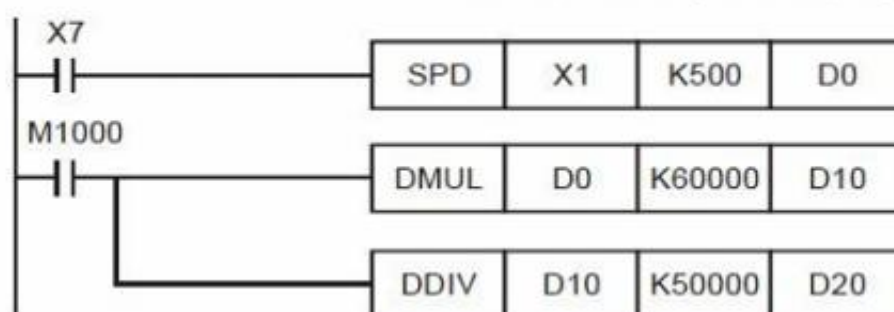
هدف از کنترل در این مثال به شرح زیر است :

محاسبه سرعت چرخش چرخ گردان توسط معادله مبتنی بر شمارش پالس‌های ورودی

ابزارهای مورد استفاده در این مثال به شرح زیر است :

ایزار	تایع
X1	سنسور فتوالکتریک برای آشکارسازی پالس به این ورودی متصل می‌شود.
X7	از این ورودی برای اجرای دستورالعمل SPD استفاده شده است.

برنامه کنترلی این مثال مطابق با شکل زیر است :



توضیحات برنامه به شرح زیر است :

وقتی که $X7=ON$ فعال شود، دستورالعمل SPD اجرا خواهد شد. $D2$ ورودی پالس سرعت بالا $X1$ را محاسبه خواهد کرد و بعد از $500ms$ محاسبه متوقف خواهد شد. نتیجه محاسبه در $D0$ و $D1$ ذخیره خواهد شد.

از معادله زیر برای به دست آوردن سرعت چرخش چرخ اتومبیل استفاده شده است.

$$N = \frac{D0}{nt} \times 60 \times 10^3 (rpm)$$

$N \leftarrow$ تعداد چرخش بر حسب rpm

$N \leftarrow$ تعداد پالس تولید شده در هر دور چرخش

$t \leftarrow$ زمان دریافت پالس (ms)

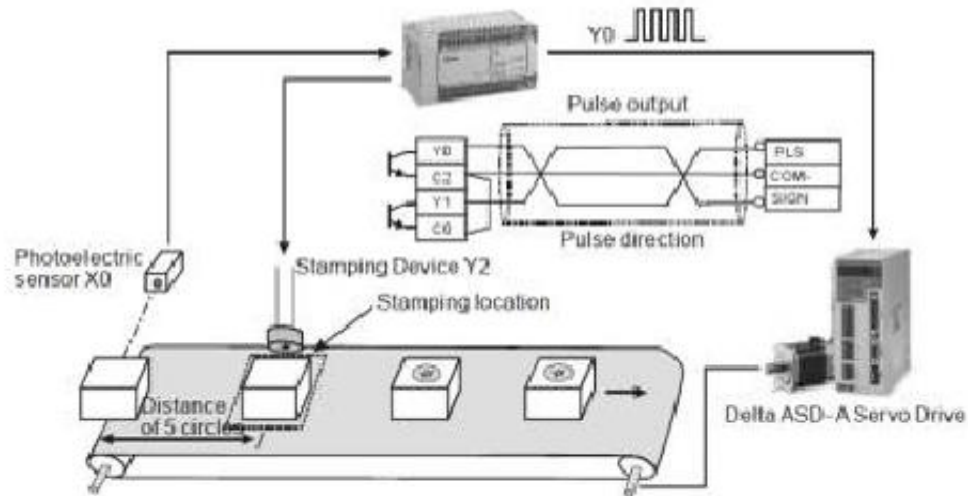
اگر تعداد پالس تولید شده در هر دور چرخش برابر $K100$ بوده و تعداد پالس $K150$ در $500ms$

دریافت شده باشد، سرعت چرخش به قرار زیر است :

$$N = \frac{D0}{nt} \times 60 \times 10^3 = \frac{750 \times 60 \times 10^3}{100 \times 500} \times (rpm) = 900 rpm$$

سرعت چرخش N در $D20$ و $D21$ ذخیره می‌شود.

6-10 برنامه کنترل خط تولید توسط دستورالعمل PLSY

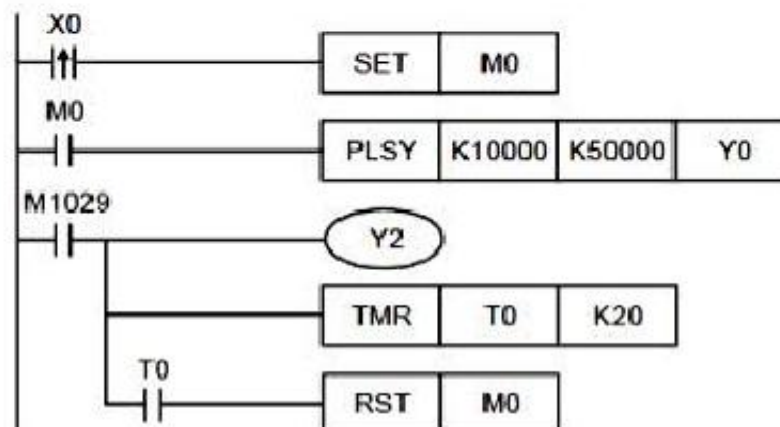


هدف از کنترل در این مثال به شرح زیر است :

وقتی که توسط سنسور فتوالکتریک محصول آشکارسازی شد، سرودرایو تا 5 دور شروع به چرخیدن می‌کند تا محصول را به محل استامپ زدن برده و در عرض 2 ثانیه فرایند استامپ زدن را اجرا می‌کند. ابزارهای مورد استفاده در این مثال به شرح زیر است :

ابزار	تابع
X0	سنسور فتوالکتریک وقتی که بسته در جای خود قرار گرفت، ورودی X0 فعال ON می‌شود.
Y0	خروجی پالس Pulse Output
Y1	جهت پالس Pulse direction
Y2	استامپ زن
T0	تنظیم زمان استامپ زن

برنامه کنترل این مثال مطابق با زیر است :



توضیحات برنامه به شرح زیر است :

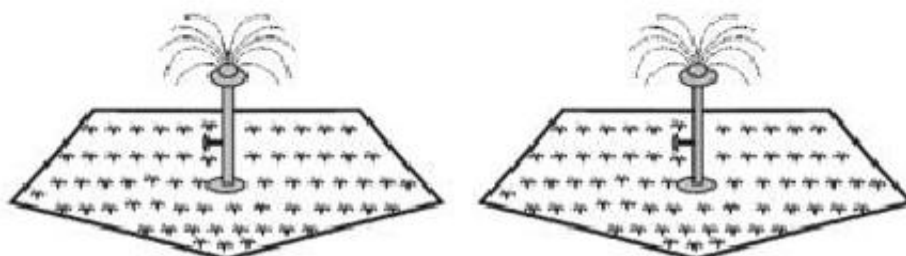
وقتی که یک محصول توسط سنسور فتوالکتریک متصل شده به ورودی X0 آشکار شد، دستورالعمل SET برای ست کردن فلگ M0 و دستورالعمل PLSY برای ارسال پالس به خروجی Y1 با فرکانس 10KHz اجرا می شود.

وقتی که تعداد پالس خروجی به 50000 رسید، سرودرایو برای 5 دور شروع به چرخیدن می کند تا محصول را به محل استامپ زن بفرستد، با فعال شدن ابزار استامپ زن (Y2)، رله داخلی M1027 نیز فعال ON خواهد شد. در این زمان، تایمر T0 برای 2 ثانیه شروع به شمارش می کند. بعد از 2 ثانیه، کنتاکت در حالت عادی باز (NO) تایمر T0 فعال می شود تا M0 را ریست کند. دستورالعمل PLSY نیز مانند M1027 و Y2 ریست خواهد شد. در نهایت، فرآیند استامپ زدن تکمیل شده است. وقتی که X0 مجدداً راه اندازی شود، PLSY مجدداً اجرا شده و Y0 برای پالس خروجی راه اندازی می شود. فرآیند استامپ زدن دوباره تکرار خواهد شد.

توجه:

در این برنامه، زمان راه اندازی ورودی X0 باید بعد از تکمیل شدن فرآیند استامپ زدن باشد. در غیر این صورت، در این فرآیند خطا رخ می دهد.

6- 11 برنامه کنترل شیر برقی اسپری توسط دستورالعمل PWM



هدف از کنترل در این مثال به شرح زیر است :

کنترل عملکرد شیر برقی اسپری به اندازه های 25%، 50% و 100% توسط تنظیمات $t_{on} = t_{off}$ مقدار تکنیک PWM با ولتاژ 24V.

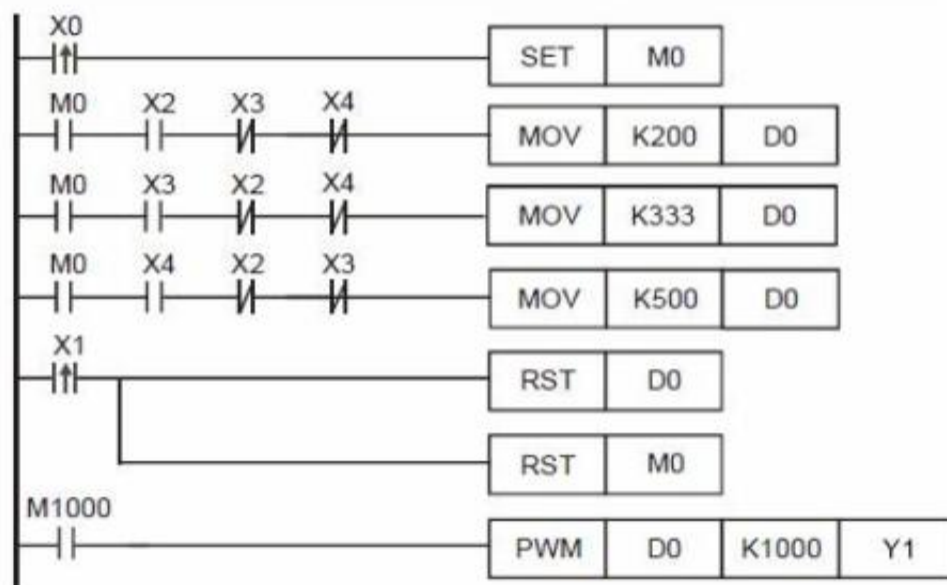
برای کاهش تدریجی انرژی (ولتاژ)، در هنگام قطع و راه‌اندازی مجدد پروسه از روش سوئیچینگ برای اجرای فوری قطع و وصل شیربرقی استفاده شده است. روش سوئیچینگ تا حدودی مانند قطع جریان می‌باشد، بنابراین یک Clipper فراخوانی می‌شود. در برنامه عملی از اعمال یک ترانزیستور بین تغذیه و موتور برای نمایش تابع Clipper استفاده شده است. سیگنال پالس به بیس ترانزیستور برای جاری کردن جریان بین بیس و امیتر اعمال می‌شود. ولتاژ ورودی موتور یک مقدار نسبی $t_{on} = t_{off}$ می‌باشد، بنابراین ولتاژ موتور باید توسط میزان کردن مقدار $t_{on} = t_{off}$ تنظیم شود.

روش‌های مختلفی برای میزان‌سازی این مقدار وجود دارد و بیشترین کاربرد در مدار فرمان تنظیم زمان روشن بودن (t_{on}) با دوره زمانی مختلف است. در این مثال، از روش فراخوانی PWM (Pulse-Width Modulation) استفاده شده است.

ابزارهای مورد استفاده در این مثال به شرح زیر است:

ابزار	تابع
X0	وقتی که شاستی START فشرده شود، X0=ON خواهد شد
X1	وقتی که شاستی STOP فشرده شود، X1=ON خواهد شد
X2	دکمه عملکرد 25%
X3	دکمه عملکرد 50%
X4	دکمه عملکرد 100%
Y1	کنترل اندازه عملکرد شیربرقی
D0	ذخیره اندازه عملکرد شیربرقی

برنامه کنترلی این مثال مطابق با شکل زیر است:



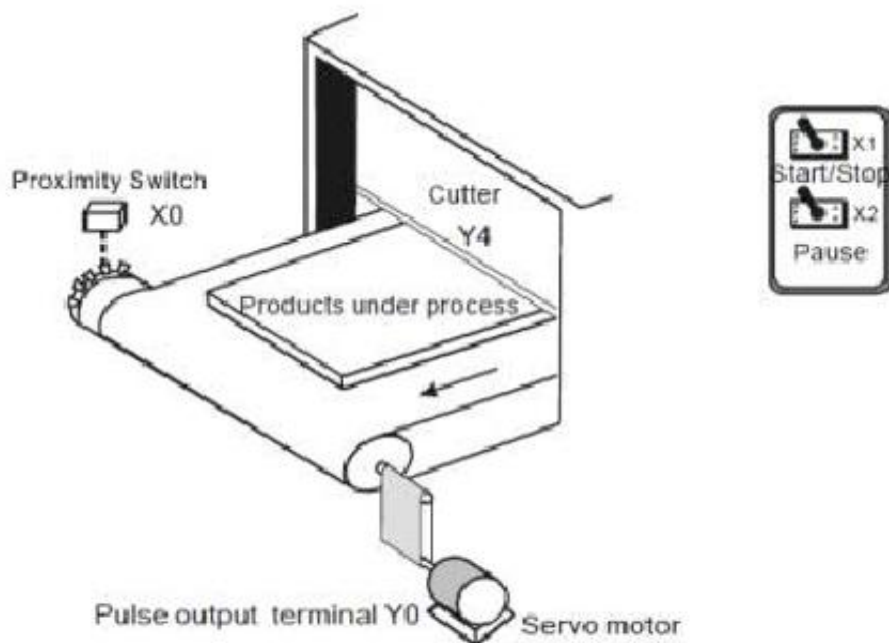
توضیحات برنامه به شرح زیر است :

در این برنامه، اندازه عملکرد شیربرقی اسپری به وسیله مقدار D0 کنترل می‌شود.

وقتی که دکمه START فشرده شود، X0 فعال شده و M0 را Set می‌کند. سیستم اسپری آب به شرط این که شاستی راه‌اندازه عملکرد شیربرقی فشرده شده باشد، شروع به اسپری کردن آب می‌کند. وقتی که دکمه 25% فشرده شده باشد (X2=ON)، مقدار در D0=K200 و D0/(K1000- و D0)=0.25 خواهد شد. بنابراین، اندازه عملکرد شیربرقی 25% می‌شود. وقتی که دکمه 50% فشرده شده باشد (X3=ON)، مقدار DO=K333 و D0/(K1000- و D0)=0.50 خواهد شد. بنابراین، اندازه عملکرد شیربرقی 50% می‌شود. وقتی که دکمه 100% فشرده شده باشد (X4=ON)، مقدار D0=K500 و D0/(K1000- و D0)=1 خواهد شد. بنابراین، اندازه عملکرد شیربرقی 100% می‌شود. وقتی که دکمه STOP فشرده شود، X1 فعال شده و مقدار D0 را پاک کرده و برابر با 0 قرار می‌دهد. یعنی D0/(K1000-D0)=0 خواهد شد. بنابراین، اندازه عملکرد برابر 0 می‌شود. در این زمان، فلگ شروع بکار سیستم یعنی M0 ریست می‌شود.

نکته: هدف از این برنامه فقط آشنایی با تنظیم پهنای پالس PWM می‌باشد.

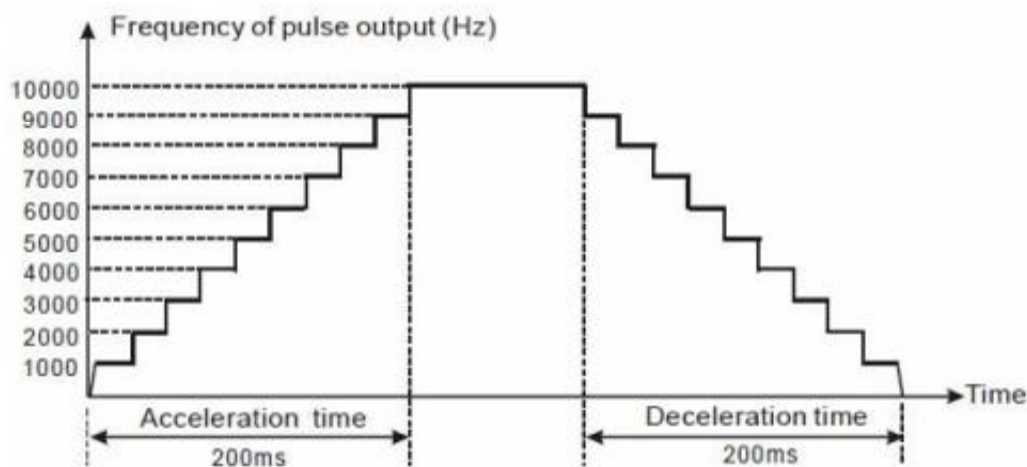
6-12 کنترل زمان شتاب مثبت / شتاب منفی سروموتور توسط دستورالعمل PL SR



هدف از کنترل در این مثال به شرح زیر است :

وقتی که تعداد پالس شمرده شده به تعداد مشخص رسید، شمارش پالس تولید شده توسط سروموتور صورت گرفته و فرآیند برش اجرا می‌شود.

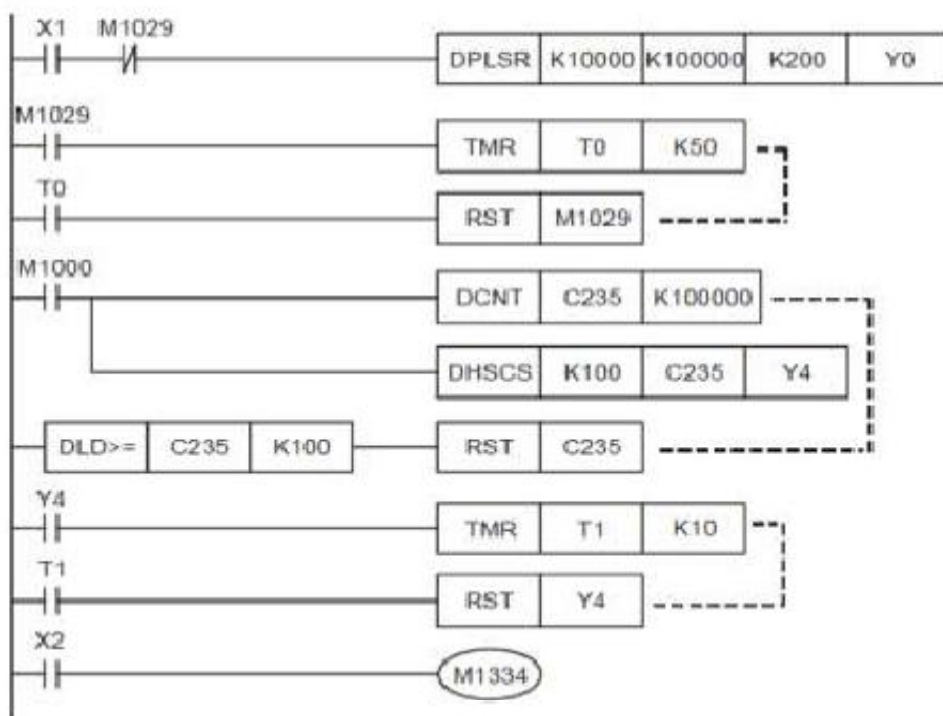
بادامک به چندین دندان به قسمت‌ها و اندازه‌ها و درجه‌های مساوی بر روی محور تقسیم شده است و در سمت دیگر محور سروموتور وجود دارد. از این رو، وقتی که سروموتور شروع به چرخیدن می‌کند، سنسور القایی 10 پالس فرستاده شده توسط ده دندانه‌ای بادامک را آشکارسازی می‌کند. وقتی که سروموتور ده دور چرخید، تعداد پالس برابر با 100 Pulse می‌شود، کانوایر متوقف شده و سیستم فرآیند برش را برای 1 ثانیه اجرا می‌کند. در برنامه از یک سروموتور برای چرخاندن کانوایر استفاده شده است. به دلیل راه‌اندازی باری سنگین نیاز است که جهت راه‌اندازی و توقف سروموتور یک شتاب مثبت و شتاب منفی را تنظیم کرد. زمان برای شتاب مثبت / شتاب منفی مطابق با دیاگرام زیر برابر 200ms تنظیم شده است :



ابزار مورد استفاده در این مثال به شرح زیر است :

ایزار	تابع
X0	سنسور القایی برای آشکارسازی پالس ایجاد شده توسط دندانه‌های بادامک
X1	وقتی که شاستی START فشرده شود، X1=ON می‌شود.
X2	وقتی که شاستی STOP فشرده شود، X2=ON خواهد شد.
Y0	خروجی پالس سرعت بالا برای اعمال به سرودرایو
Y4	کاتر
C235	کاتر سرعت بالا

برنامه کنترلی این مثال مطابق با شکل زیر است :



در خط اول برنامه، خروجی Y2 به عنوان خروجی پالس سرعت بالا برای اعمال پالس به سرودرایو در نظر گرفته شده است. حداکثر فرکانس پالس خروجی برابر 10000HZ تنظیم شده و تعداد پالس خروجی برای یک سیکل کاری برابر 100000 تنظیم شده است. همچنین، زمان شتاب مثبت و شتاب منفی برابر با 200ms تنظیم شده است.

خط دوم و سوم برنامه، سروموتور بعد از ده دور چرخش متوقف می‌شود. M1029=ON شده و تایمر 0 شروع به شمردن زمان می‌کند. بعد از گذشت 5 ثانیه تایمر T0 ریست می‌شود.

در خط 4 و 5 و 6 برنامه، وقتی که تعداد 100 پالس توسط سنسور القایی متصل شده به ورودی X0 آشکار شد، Y4=ON شده و کاتر شروع به بریدن می‌کند سپس محتویات شمارنده C235 پاک می‌شود.

در خط 7 و 8 برنامه، کاتر عمل برش را برای یک ثانیه شروع می‌کند و سپس ریست می‌شود.

در خط 9 برنامه، وقتی که X2=ON شد، خروجی پالس متوقف می‌شود.

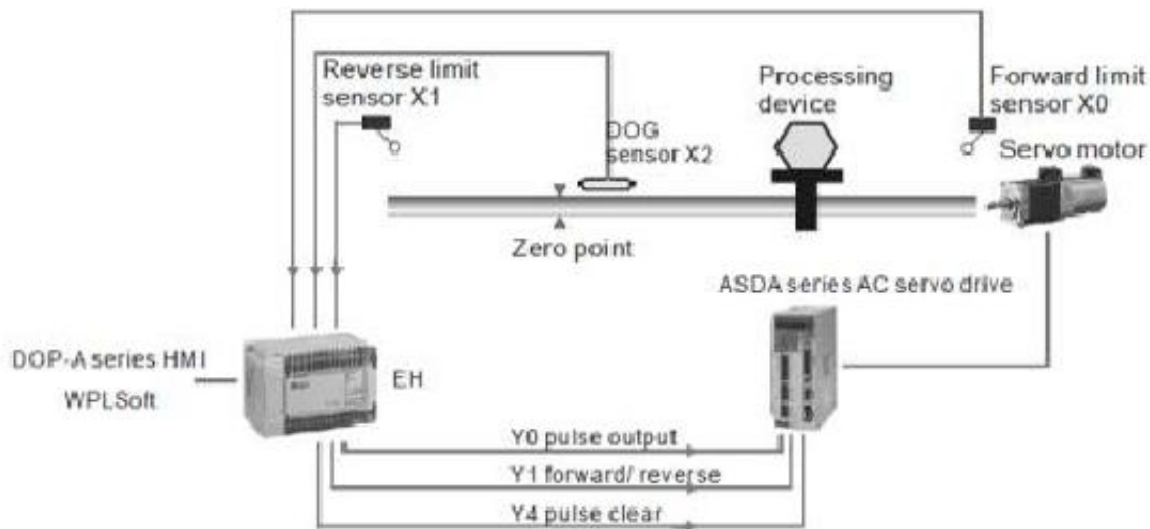
توضیحات برنامه به شرح زیر است :

وقتی که دکمه START فشرده شود، X1=ON خواهد شد، سروموتور با سرعت 0.1 r/s (F=1000Hz) شروع به چرخیدن کرده و سرعت توسط 0.1 r/min در 20ms افزایش

خواهد یافت. بعد از گذشت 200ms، سرعت به $1r/s$ ($F=10000HZ$) رسیده و سپس سرعت ثابت باقی خواهد ماند. وقتی که مقدار پالس خروجی به نزدیک مقدار تنظیم شده رسید، شتاب سروموتور منفی خواهد شد و در نهایت چرخش متوقف می‌شود. در این زمان سروموتور به مقدار تنظیم شده رسیده است.

وقتی که دکمه PAUSE فشرده شود، $X2=ON$ خواهد شد، چرخش سروموتور متوقف شده و مقدار PV در شمارنده C235 ذخیره نخواهد شد. وقتی که $X2=OFF$ شود، سروموتور مجدداً شروع به چرخیدن کرده و وقتی که پالس به مقدار تنظیم شده رسید، سروموتور متوقف می‌شود. وقتی که سروموتور یک دور چرخید، سنسور القایی 10 پالس را آشکارسازی می‌کند. وقتی که سروموتور 10 دور چرخید، سنسور القایی 100 پالس را آشکارسازی می‌کند، چرخش سرو متوقف شده و سیستم برای یک ثانیه فرآیند برش را انجام می‌دهد.

6- 13 سیستم حرکت دمونستراسیون (تجربی) ساده موقعیت سروودرایوهای AC سری ASDA دلتا



هدف از کنترل در این مثال به شرح زیر است :

اجرای برگشت به نقطه صفر که در این سیستم پالس از PLC به سروودرایو ارسال می‌شود.

ابزار مورد استفاده در این مثال به شرح زیر است :

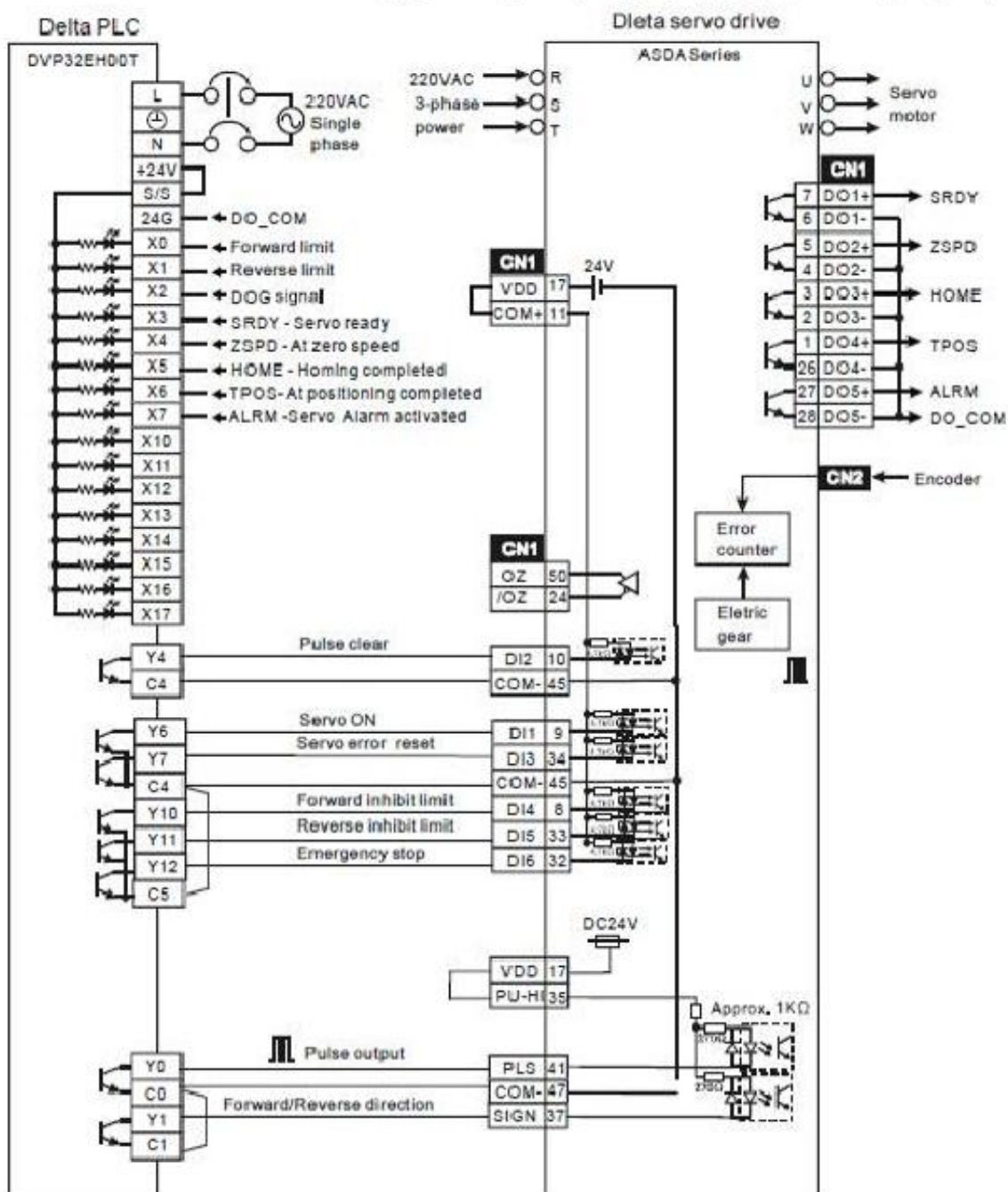
ابزار	تابع
M0	سوئیچ برگشت به نقطه صفر
M1	سوئیچ شروع به کار حرکت به جلو برای 10 دور چرخش
M2	سوئیچ شروع به کار حرکت به عقب برای 10 دور چرخش
M3	سوئیچ تعیین حرکت مطلق: 40000
M4	سوئیچ تعیین حرکت مطلق: -5000
M10	سوئیچ شروع به کار سرو
M11	سوئیچ ریست خطا
M12	سوئیچ توقف خروجی پالس
M13	سوئیچ امرجنسی استوپ
X0	سنسور محدود کردن حرکت به جلو
X1	سنسور محدود کردن حرکت به عقب
X2	سنسور سیگنال DOG
X3	سیگنال آماده به کار سرودرایو ارسال شده از سرودرایو به PLC (مطابق با M20)
X4	دریافت سیگنال در سرعت صفر (مطابق با M21)
X5	دریافت سیگنال تکمیل شدن عمل Homing (مطابق با M22)
X6	دریافت سیگنال تکمیل شدن عمل موقعیت یابی (مطابق با M23)
X7	دریافت سیگنال فعال آلارم (مطابق با M24)
Y0	خروجی پالس
Y1	کنترل مستقیم حرکت به جلو و حرکت به عقب
Y4	پاک کردن رجیستر پالس در سرو
Y6	فعال کردن سرو
Y7	ریست کردن خطا
Y10	منع کردن محدوده حرکت به جلو
Y11	منع کردن محدوده حرکت به عقب
Y12	امرجنسی استوپ
M0	سرو آماده به کار
M21	در سرعت صفر
M22	تکمیل شدن عمل Homing
M23	تکمیل شدن عمل حرکت به موقعیت تنظیم شده
M24	آلارم فعال شده است

تنظیمات پارامترهای سرودرایو AC مدل ASDA-B,B2

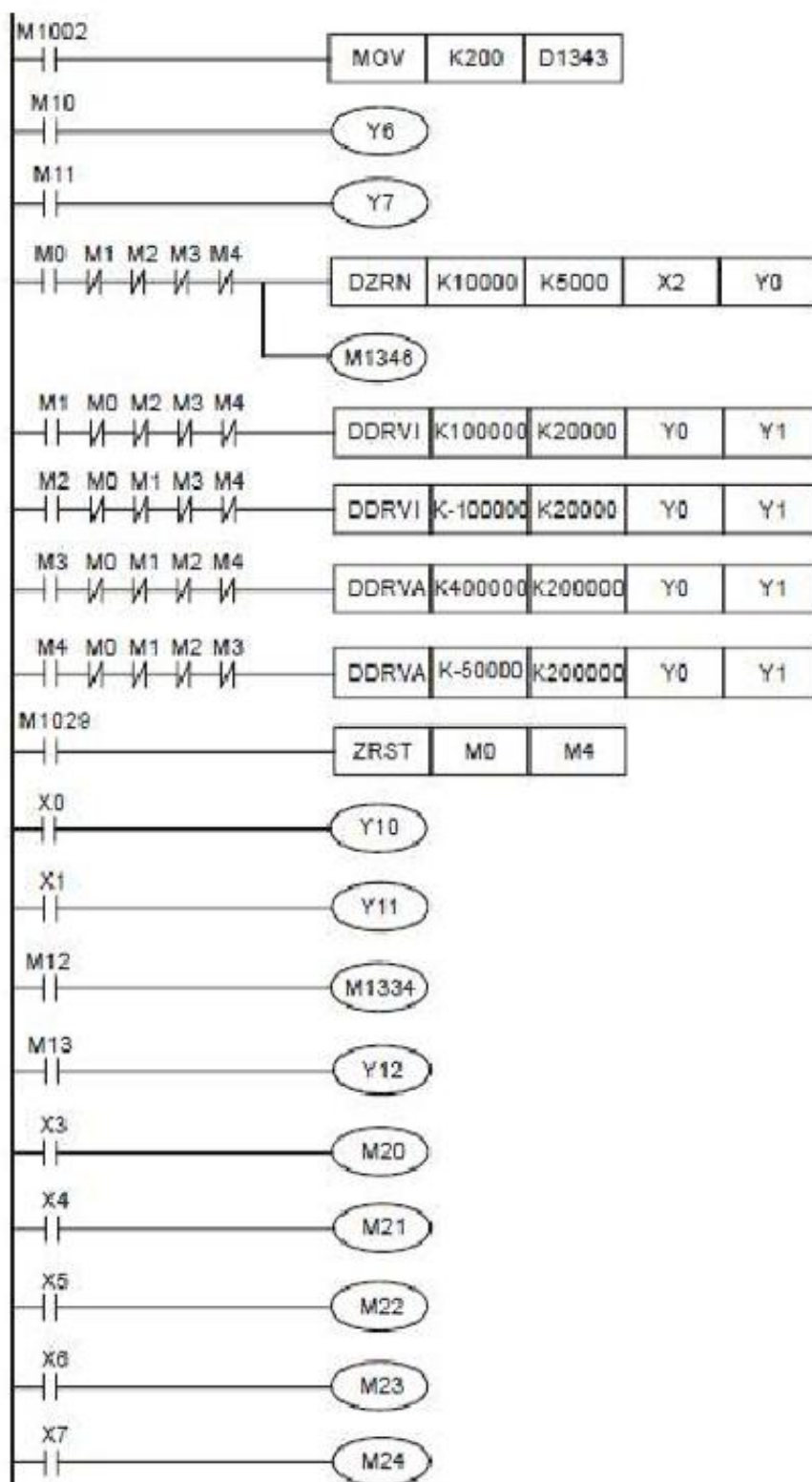
پارامتر	مقدار تنظیم شده	تابع
P0-02	2	وضعیت‌های سرودرایو
P1-00	2	نوع ورودی پالس خارجی: Pulse Direction
P1-01	0	مد کنترل و جهت خروجی
P2-10	101	نخستین ترمینال ورودی دیجیتال (DI1)
P2-11	104	دومین ترمینال ورودی دیجیتال (DI2)
P2-12	102	سومین ترمینال ورودی دیجیتال (DI3)
P2-13	122	چهارمین ترمینال ورودی دیجیتال (DI4)
P2-14	123	پنجمین ترمینال ورودی دیجیتال (DI5)
P2-15	121	ششمین ترمینال ورودی دیجیتال (DI6)
P2-16	0	هفتمین ترمینال ورودی دیجیتال (DI7)
P2-17	0	هشتمین ترمینال ورودی دیجیتال (DI8)
P2-18	101	نخستین ترمینال خروجی دیجیتال (DO1)
P2-19	103	دومین ترمینال خروجی دیجیتال (DO2)
P2-20	103	سومین ترمینال خروجی دیجیتال (DO3)
P2-21	105	چهارمین ترمینال خروجی دیجیتال (DO4)
P2-22	107	پنجمین ترمینال خروجی دیجیتال (DO5)

اگر سرودرایو به صورت عادی نتوانست شروع به کار کند، این به آن معنی است که بعضی از تنظیمات پارامترها اشتباه تنظیم شده‌اند. در این حالت، می‌توان با تنظیم $P2-08=10$ (تنظیمات پیش‌فرض کارخانه) تنظیمات را ریست کرده و مجدداً پارامترهای جدول بالا را تنظیم نمایید.

سیم‌بندی بین PLC و سروودرایو در این مثال مطابق با شکل زیر است :



برنامه کنترلی این مثال مطابق با شکل زیر است :



توضیحات خط به خط برنامه به شرح زیر است:

- خط 1 ← تنظیم زمان شتاب مثبت و شتاب منفی برابر 200ms
- خط 2 ← شروع بکار سرو
- خط 3 ← ریست کردن خطا
- خط 4 ← برگشت به نقطه صفر
- خط 5 ← پاک کردن رجیستر پالس سرودرایو
- خط 6 ← شروع به کار حرکت به جلو برای 10 دور چرخش
- خط 7 ← شروع به کار حرکت به عقب برای 10 دور چرخش
- خط 8 ← تعیین 400000 پالس مطلق برای درایو
- خط 9 ← تعیین 50000 - پالس مطلق برای درایو
- خط 10 ← ریست شدن M0~M4 بعد از تکمیل شدن موقعیت یابی
- خط 11 ← منع کردن محدوده حرکت به جلو
- خط 12 ← منع کردن محدوده حرکت به عقب
- خط 13 ← متوقف کردن خروجی پالس
- خط 14 ← امرجنسی استوپ
- خط 15 ← سرودرایو آماده به کار
- خط 16 ← سرو در سرعت صفر
- خط 17 ← مد Homing در سرو تکمیل شده است.
- خط 18 ← سرودرایو در موقعیت تکمیل شده است.
- خط 19 ← خطا و آلام

توضیحات برنامه در این مثال به شرح زیر است:

ابزارهای M با سوئیچ‌ها کار می‌کنند و نمایش وضعیت‌ها می‌توانند توسط HMI یا برنامه WPL Soft تعیین شده و کار کنند.

با فعال (ON) شدن ورودی X3 (آماده به کار بودن سرو) در صورت عدم وجود سیگنال آلام سرودرایو روشن خواهد شد. با فشردن سوئیچ Servo ON رله داخلی M0 فعال ON شده و خروجی Y6 مربوط به روشن شدن سرودرایو فعال خواهد شد.

وقتی که سوئیچ برگشت به نقطه صفر فعال شد (M0=ON)، سرودرایو عمل برگشت به نقطه صفر را اجرا خواهد کرد. وقتی که سنسور سیگنال DOG فعال شود، سرودرایو از سرعت جاری به سرعت

JOG برابر 5KHz سوئیچ خواهد شد. وقتی که X2 غیرفعال (OFF) شود، سرودرایو در هنگام کار کردن فوراً متوقف شده و برگشت به نقطه صفر تکمیل می‌شود.

وقتی که برای 10 دور چرخیدن حرکت به جلو سوئیچ M1 فشرده شود (M1=ON)، سروموتور با دستورالعمل افزایشی سرودرایو شروع به چرخیدن کرده و بعد از 10 دور چرخیدن در جهت حرکت به جلو، سرو متوقف می‌شود.

وقتی که برای 10 دور چرخیدن حرکت به عقب سوئیچ M2 فشرده شود (M2=ON)، سروموتور با دستورالعمل افزایشی سرودرایو شروع به چرخیدن کرده و بعد از 10 دور چرخیدن در جهت حرکت به عقب، سرو متوقف می‌شود.

وقتی که برای 40000 پالس مطلق تعیین شده سوئیچ M3 فشرده شود (M3=ON)، سروموتور با دستورالعمل مطلق سرودرایو شروع به چرخیدن کرده و بعد از تکمیل شدن موقعیت، سرو متوقف خواهد شد.

وقتی که برای 50000 - پالس مطلق تعیین شده سوئیچ M4 فشرده شود (M4=ON)، سروموتور با دستورالعمل مطلق سرودرایو شروع به چرخیدن کرده و بعد از تکمیل شدن موقعیت، سرو متوقف خواهد شد.

اگر سنسور محدود کننده‌ی حرکت به جلو توسط ابزار مکانیکی متصل شده به مکانیزم حرکتی سروموتور فعال شود (X0=ON, Y10=ON)، سروموتور متوقف شده و آلارم فعال می‌شود (M24=ON).

اگر سنسور محدود کننده‌ی حرکت به عقب توسط ابزار مکانیکی متصل شده به مکانیزم حرکتی سروموتور فعال شود (X1=ON, Y11=ON)، سروموتور متوقف شده و آلارم فعال می‌شود (M24=ON).

اگر آلارم سرو فعال شد، با فشردن سوئیچ M11 خطا را ریست نمایید تا آلارم پاک شود. بعد از ریست آلارم برنامه می‌تواند ادامه برنامه موقعیت‌یابی را انجام دهد بدون آن که کل برنامه ریست شود.

وقتی که سوئیچ توقف خروجی پالس فشرده شود (M12=ON)، PLC خروجی پالس را متوقف کرده و تعداد پالس خروجی در رجیستر ذخیره خواهد شد. وقتی که M12=OFF شود، PLC خروجی پالس را از تعداد پالس ذخیره شده ادامه خواهد داد.

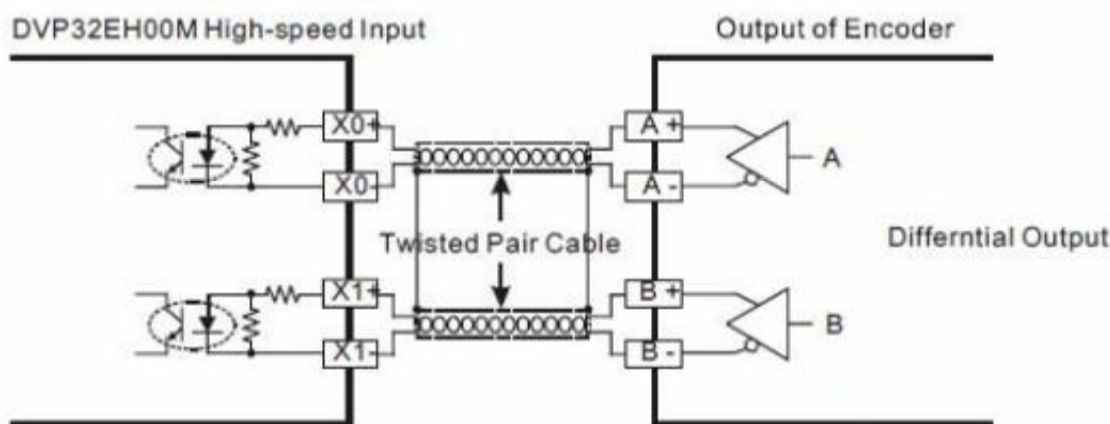
با فشردن دکمه امرجنسی استوپ (M13=ON)، سرودرایو AC سریعاً متوقف خواهد شد. وقتی که M13=OFF شود، برای موقعیت‌یابی، سرودرایو فاصله باقی‌مانده را تکمیل نخواهد کرد.

از M1346 در برنامه برای پاک کردن رجیستر پالس بعد از تکمیل شدن برگشت به نقطه صفر استفاده شده است. وقتی که M1346 فعال شود، خروجی Y4 از PLC یک پالس 20ms برای پاک کردن پالس و نمایش 0 بر روی پنل سرودرایو ارسال خواهد کرد (مطابق با پارامتر P00-02 سرودرایو که برابر 0 تنظیم شده است).

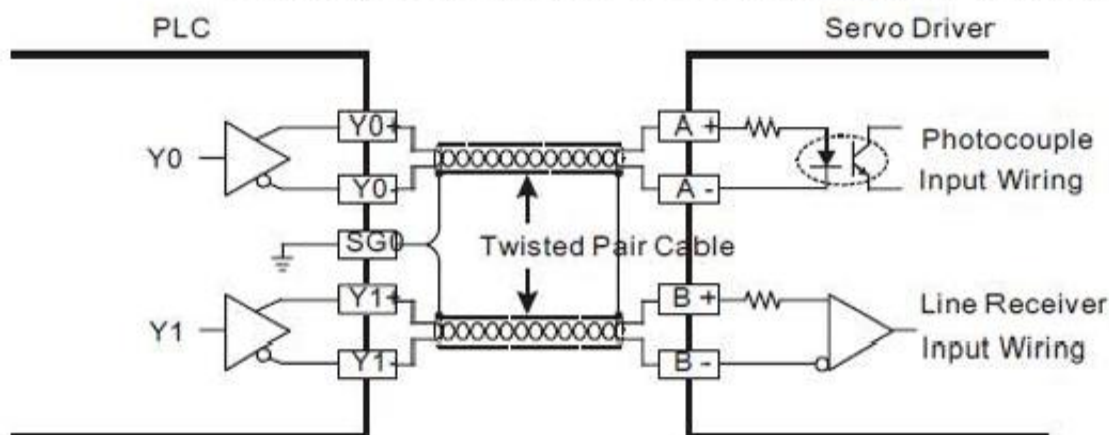
از M1029 در برنامه برای ریست کردن M0~M4 استفاده شده است تا بتوان هر یک از دستورالعمل‌های موقعیتیابی را به درستی اجرا کرد.

6- 14- شمارنده سرعت بالای پالس از جنس AB-phase

سیم‌بندی برای ورودی تفاضلی (پالس سرعت بالا در شرایط نویز بالا) بین PLC و انگ‌در مطابق با شکل زیر است:



سیم‌بندی برای خروجی تفاضلی بین PLC و سرودرایو مطابق با شکل زیر است:



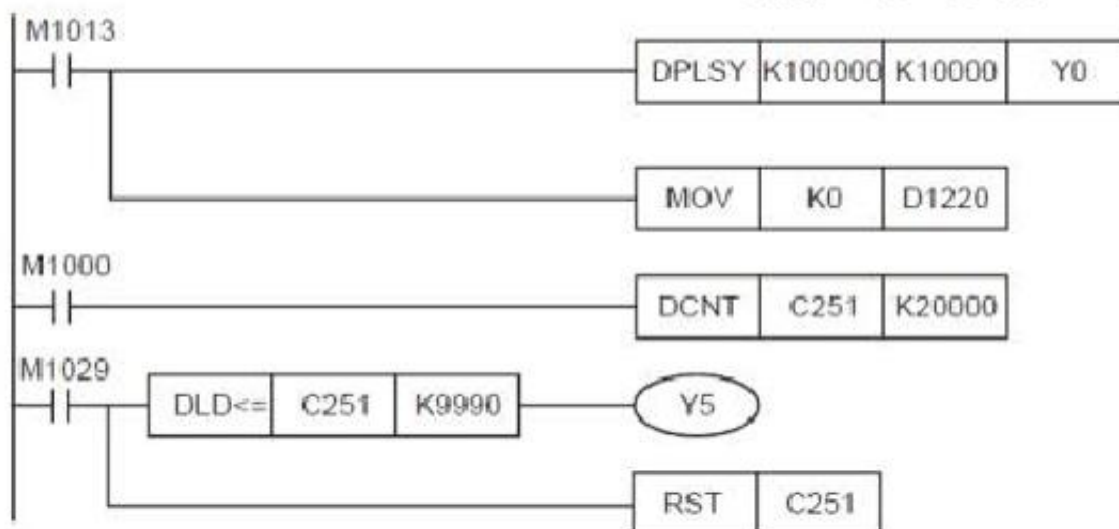
هدف از کنترل در این مثال به شرح زیر است :

PLC مدل DVP32EH00M خروجی پالس سرعت بالای از جنس AB-phase را برای کنترل سرو در سرعت 10000pps ده هزار پالس در هر ثانیه به سرودرایو ارسال می‌کند. سرودر چرخیده و توسط انگ‌در پالس رمزگذاری می‌شود و نتیجه به ترمینال‌های ورودی (ورودی تفاضلی) به شمارنده سرعت بالای PLC ارسال خواهد شد. اگر مقدار شمرده شده‌ی شمارنده سرعت بالای PLC با تعداد پالس ارسال شده توسط دستورالعمل MPG متفاوت باشد، آلام فعال می‌شود.

ابزار مورد استفاده در این مثال به شرح زیر است :

ابزار	تابع
Y0	خروجی پالس با فرکانس 100KHz
Y5	نمایش دهنده آلام
M1013	کلاک پالس 1s
M1029	فلگ تکمیل شدن پالس خروجی
D1220	تنظیم نخستین گروه خروجی فازی (Y0 Y1) CH0
C251	شمارنده سرعت بالا

برنامه کنترلی این مثال به شرح زیر است :



توضیحات برنامه :

در خط اول برنامه، سرعت خروجی Y0 برابر 10000ppr پالس در هر ثانیه با فرکانس 100KHz می‌باشد.

در این مثال، از M1013 برای ارسال پالس به سرودرایو از طریق PLC استفاده شده است. D1220=K0 خروجی Y1 را با خروجی پالس فعال می‌کند. سیگنال فیدبک رمزگذاری شده

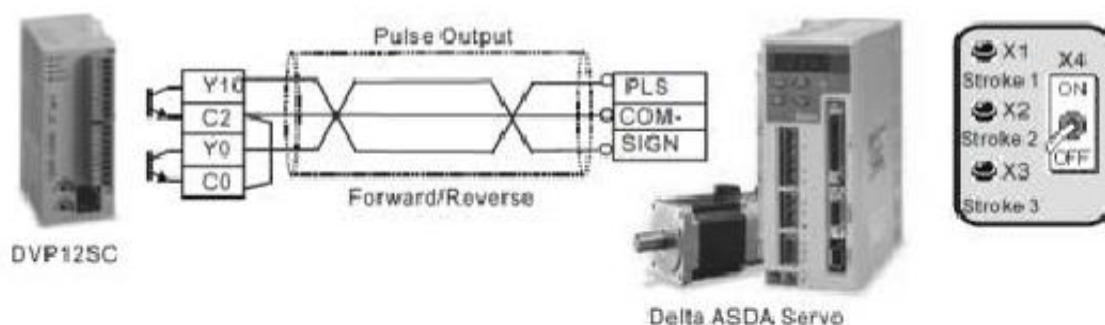
سروموتور از انگ‌در به ورودی سرعت بالای (X0 و X1) ارسال می‌شود. X0 و X1 به شمارنده سرعت بالا C251 متصل شده و حداکثر فرکانس شمارش آن 200KHz است.

وقتی که ارسال پالس تکمیل شد، M1029 فعال (ON) می‌شود. دستورالعمل مقایسه $DLD \leq$ اجرا خواهد شد. اگر اختلاف بین مقدار C251 و تعداد پالس خروجی بالاتر از 10 بود ($C251 \leq K9990$)، آلارم Y5 فعال خواهد شد.

وقتی که M1029=ON شود، دستورالعمل [RST C251] اجرا خواهد شد. مقدار شمارنده C251 پاک شده و شمارنده C251 برای بار بعدی شروع به شمارش می‌کند.

از آنجایی که سیگنال خروجی انگ‌در سرو یک سیگنال تفاضلی می‌باشد، در این مثال نیاز به PLC مدل DVP32EHOOM است که سیگنال ورود تفاضلی را با ترمینال‌های ورودی (X0 و X1) و (X4 و X5) پشتیبانی می‌کند.

6- 15 تنظیمات دستورالعمل کارکرد سرو توسط دستورالعمل CJ



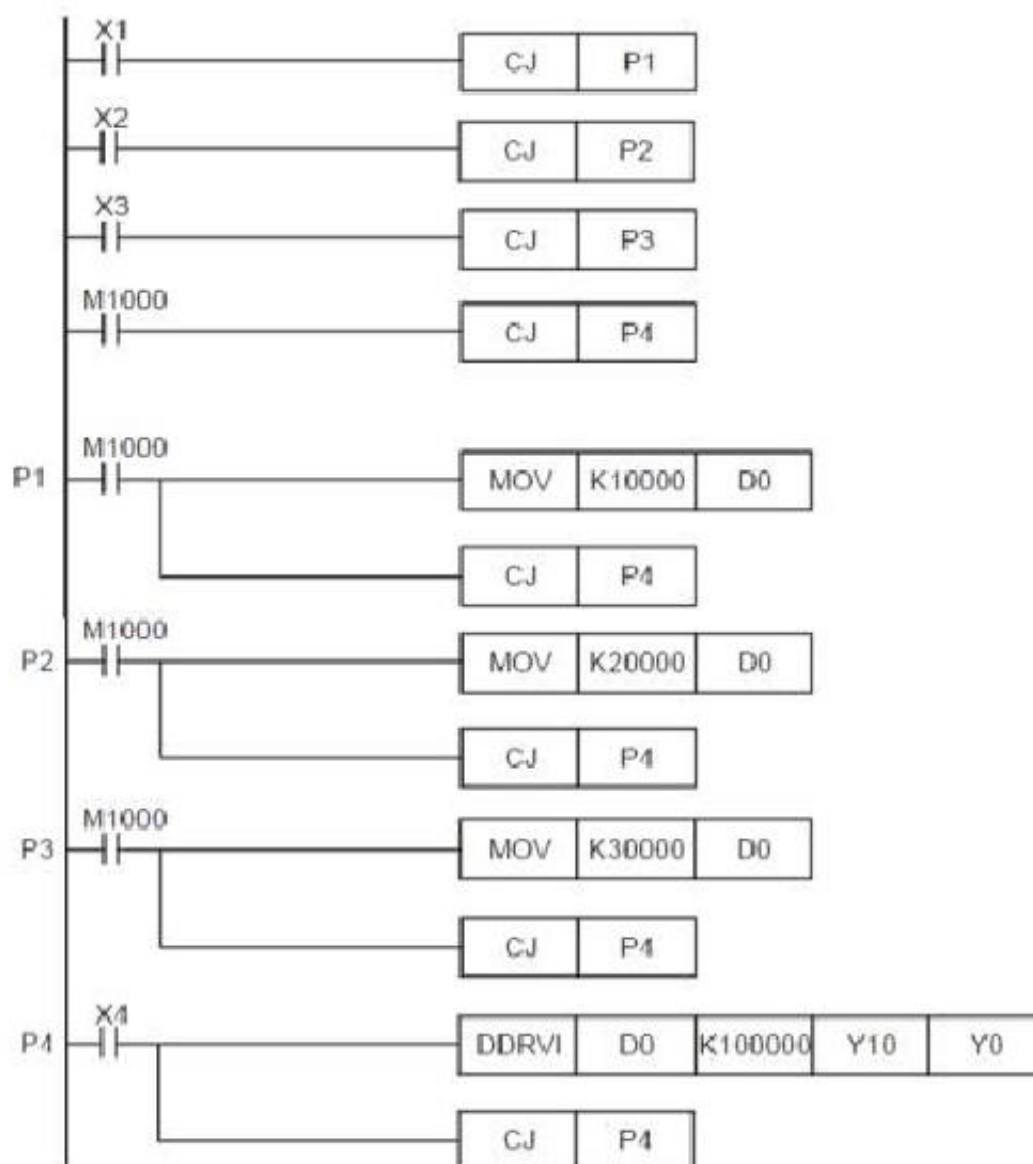
هدف برنامه در این مثال به شرح زیر است:

هدف کنترل سه فاصله حرکتی سرورایو ASDA دلتا توسط فرستادن پالس از PLC سری DVP12SC دلتا به سرورایو می‌باشد. کاربر می‌تواند فاصله حرکتی مناسب را با توجه به نیاز توسط فشردن یکی از 3 سوئیچ منحصر بفرد حرکتی انتخاب نماید.

ابزارهای مورد استفاده در این مثال به شرح زیر است:

ابزار	تابع
X1	وقتی که سوئیچ حرکتی 1 (Stroke 1) فشرده شود ورودی $X1=ON$ فعال می‌شود.
X2	وقتی که سوئیچ حرکتی 2 (Stroke 2) فشرده شود ورودی $X2=ON$ فعال می‌شود.
X3	وقتی که سوئیچ حرکتی 3 (Stroke 3) فشرده شود ورودی $X3=ON$ فعال می‌شود.
X4	وقتی که ورودی $X4=ON$ شود، سرو شروع به حرکت در جهت معینی خواهد شد.
Y0	کنترل جهت پالس (Pulse direction)
Y10	نقطه خروجی پالس (Pulse)

برنامه کنترلی این مثال مطابق با شکل زیر است :



توضیحات برنامه به شرح زیر است :

وقتی که $X1=ON$ و $X2=OFF$ ، $X3=OFF$ باشد، برنامه از CJ P1 به P1 پرش کرده و نخستین فاصله حرکتی انتخاب شده است که این حرکت به صورت محتویات حرکتی K10000 در D0 قرار می‌گیرد. در این زمان، برنامه به آدرس P4 پرش کرده و آماده پذیرش خروجی پالس می‌باشد. وقتی که $X2=ON$ و $X1=OFF$ ، $X3=OFF$ باشد، از CJ P2 به P2 پرش کرده و دومین فاصله حرکتی انتخاب شده است که این حرکت به صورت محتویات حرکتی K20000 در D0 قرار می‌گیرد. در این زمان، برنامه به آدرس P4 پرش کرده و آماده پذیرش خروجی پالس می‌باشد.

وقتی که $X1=OFF$ ، $X2=OFF$ و $X3=ON$ باشد، از CJ P3 به P3 پرش کرده و سومین فاصله حرکتی انتخاب شده است که این حرکت به صورت محتویات حرکتی K30000 در D0 قرار می‌گیرد. در این زمان، برنامه به آدرس P4 پرش کرده و آماده پذیرش خروجی پالس می‌باشد. وقتی که $X1=OFF$ ، $X2=OFF$ و $X3=OFF$ باشد، دستورالعمل CJ P4 اجرا خواهد شد. برنامه مستقیماً به اشاره گر P4 پرش کرده و آماده پذیرش خروجی پالس می‌باشد. وقتی که $X4=ON$ باشد، دستورالعمل DDRVI DO K10000 Y10 Y0 اجرا خواهد شد. در این زمان، خروجی Y10 یک تعداد معین از پالس با فرکانس 100KHz (محتویات در D0 تعداد پالس می‌باشد) به سرودرایو ارسال کرده و خروجی Y0 جهت پالس را کنترل خواهد کرد. هدف از کنترل عملکرد سرو کنترل فاصله حرکت می‌باشد که می‌توان توسط تنظیمات خروجی پالس PLC به این کنترل حرکت دست یافت.

6-16 کنترل حرکت JOG با تنظیمات حرکتی خیلی کم توسط دستورالعمل INC/DEC

هدف از کنترل در این مثال به شرح زیر است:

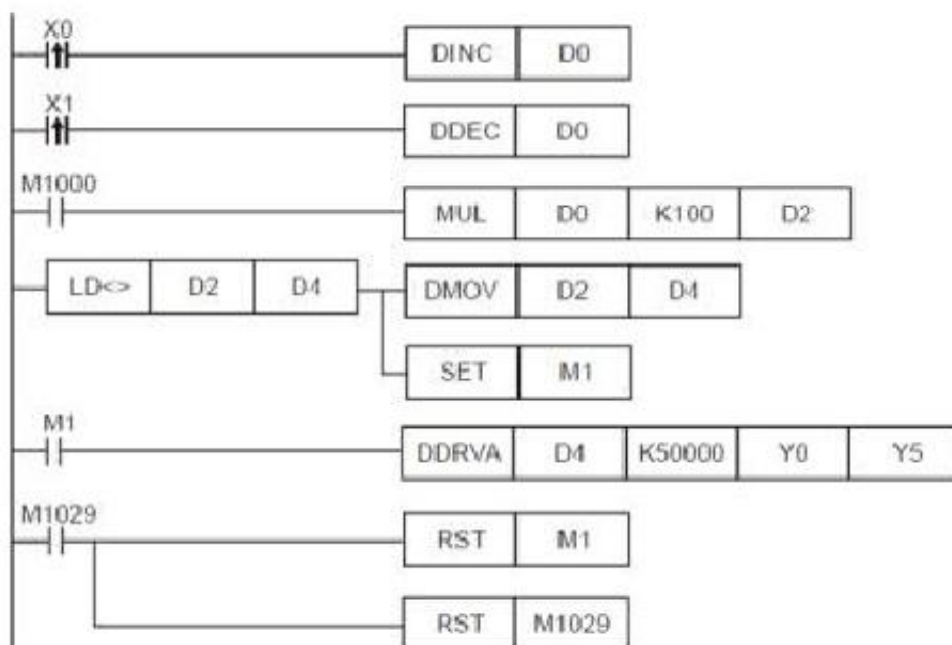
کنترل حرکت بسیار کم و ظریف توسط سوئیچ حرکت به چپ JOG و سوئیچ حرکت به راست JOG.

در این مثال، کنترل موقعیت سیستم با تنظیم حرکت بسیار کم برابر 1mm فرض شده است که می‌توان توسط فرستادن 100 پالس به سرودرایو توسط PLC این جابجایی حرکت را کنترل کرد. وقتی که ورودی X0 فعال شود، جابجایی حرکت به چپ برای دستورالعمل JOG برابر 1mm می‌باشد و وقتی که ورودی X1 فعال شود، جابجایی فاصله حرکتی به راست برای دستورالعمل JOG برابر 1mm می‌شود.

ابزارهای مورد استفاده در این مثال به شرح زیر است:

ابزار	تابع
X0	سوئیچ حرکت به چپ دستورالعمل JOG
X1	سوئیچ حرکت به راست دستورالعمل JOG
D0	موقعیت هدف
D2	تعداد پالس برای موقعیت هدف
Y0	نقطه خروجی پالس (Pulse)
Y5	سیگنال خروجی کنترل جهت پالس (Pulse direction)

برنامه کنترلی این مثال مطابق با شکل زیر است :



توضیحات برنامه به شرح زیر است :

وقتی که سوئیچ حرکت به چپ JOG (ورودی X0) فشرده شود، دستورالعمل DINC با افزایش مقدار در D0 اجرا خواهد شد، و وقتی که سوئیچ حرکت به راست JOG (ورودی X1) فشرده شود، دستورالعمل DDEC با کاهش مقدار در D0 اجرا می‌شود.

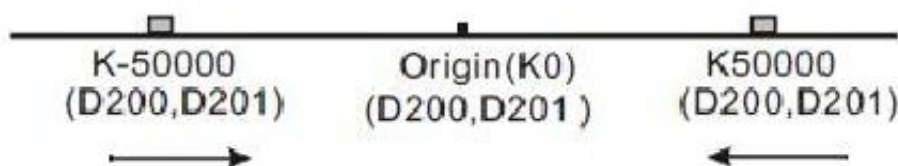
فرض کنید که مقدار اولیه در D0 و D4 برابر K0 می‌باشد. وقتی که سوئیچ حرکت به چپ JOG فشرده شود، مقدار D0 برابر K1 شده و سپس با 100 (تعداد پالس) ضرب می‌شود. تعداد پالس در D2 ذخیره شده، سپس به D4 (مقدار هدف دستورالعمل DDRVA برای موقعیت مطلق) فرستاده می‌شود و با اجرای دستورالعمل DDRVA حافظه M1 فعال می‌شود.

مطابق با اجرای دستورالعمل DDRVA، خروجی Y5 تعداد 100 پالس با فرکانس 50KHZ را به سرودرایو ارسال خواهد کرد و سیستم با سرعت JOG به موقعیت هدف ($D4=D2=K100$) از موقعیت اولیه ($D4=K0$) حرکت کرده و سیستم به سمت چپ با سرعت JOG و با جابجایی 1mm حرکت می‌کند.

اگر X0 مجدداً فعال شود، D2 برابر K200 شده و مقدار فعلی در D4 برابر با K100 است که این مقادیر با هم متفاوت هستند. مقدار در D2 برابر K200 خواهد بود و این مقدار به مقدار هدف D4 برای موقعیت مطلق ارسال می‌شود. با اجرای دستورالعمل DDRVA حافظه M1 فعال خواهد شد. سیستم با سرعت JOG و با جابجایی 1mm حرکت خواهد کرد.

همچنین، فرآیند حرکت به راست با سرعت JOG مشابه حرکت به چپ با سرعت و تغییر موقعیت JOG می‌باشد. با هر بار فشردن سوئیچ حرکت به راست JOG سیستم به اندازه‌ی 1mm با سرعت و تغییر مکان JOG شروع به حرکت خواهد کرد.

6- 17 کنترل تغییر مکان معکوس توسط دستورالعمل ENG

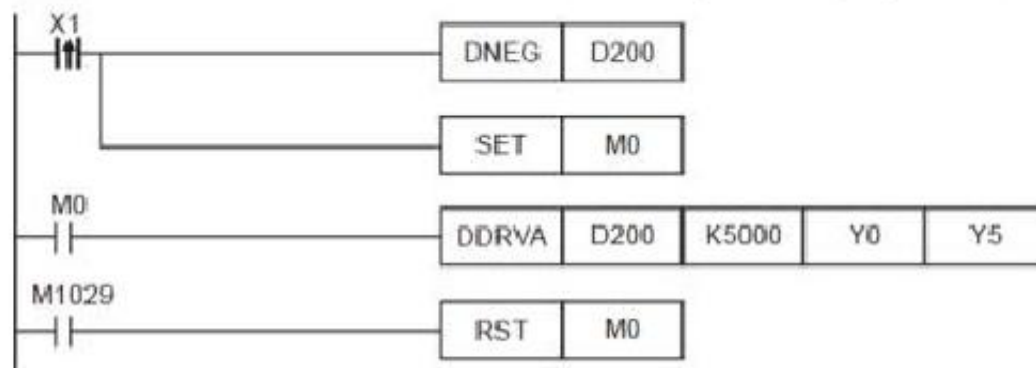


هدف از کنترل در این مثال به شرح زیر است:

نقطه متقارن در این برنامه از مبدأ $D200, D201=K0$ می‌باشد. تغییر مکان با تغییر بین حرکت تا انتهای موقعیت چپ و حرکت تا انتهای موقعیت راست کنترل می‌شود. ابزارهای مورد استفاده در این مثال به شرح زیر است:

ابزار	تابع
X1	شستی START حرکت معکوس
Y0	خروجی پالس (PULSE)
Y5	خروجی کنترل جهت پالس معکوس (pulse direction)
D200,D201	ذخیره مقدار هدف برای موقعیت مطلق

برنامه کنترلی این مثال مطابق با شکل زیر است:



توضیحات برنامه به شرح زیر است :

فرض کنید مقدار اولیه 32 بیتی در D201, D200 برابر K50000 باشد. وقتی که دکمه Reverse START فشرده شود، محتویات در D200 و D201 برابر K50000 خواهد شد.

در مجموع، با اجرای دستورالعمل DDRVA حافظه M0 فعال خواهد شد. برنامه با رسیدن به محل موقعیت هدف K50000 تغییر کرده و K-50000 با فرکانس (K50000) 5KHZ اجرا می‌شود. وقتی که سرو به موقعیت هدف رسید، M1029 فعال شده و M0 ریست خواهد شد و ارسال پالس توسط خروجی Y0 متوقف می‌شود.

وقتی که X1 مجدداً فشرده شود، مقدار D201, D200 از K-50000 به K50000 تغییر خواهد کرد. با اجرای کنترل تغییر مکان معکوس ت و رسیدن به موقعیت مطلق M0 فعال ON خواهد شد.

در عمل بالا، با هر بار فشردن کلید X1 برنامه از محل کنونی به سمت دیگر از نقطه متقارن از مبدأ تغییر مکان خواهد کرد.

فصل 7

پروژه‌های کاربردی شبکه‌ی RS-485 با پروتکل Modbus استاندارد
و غیراستاندارد DVPSCM52-SL



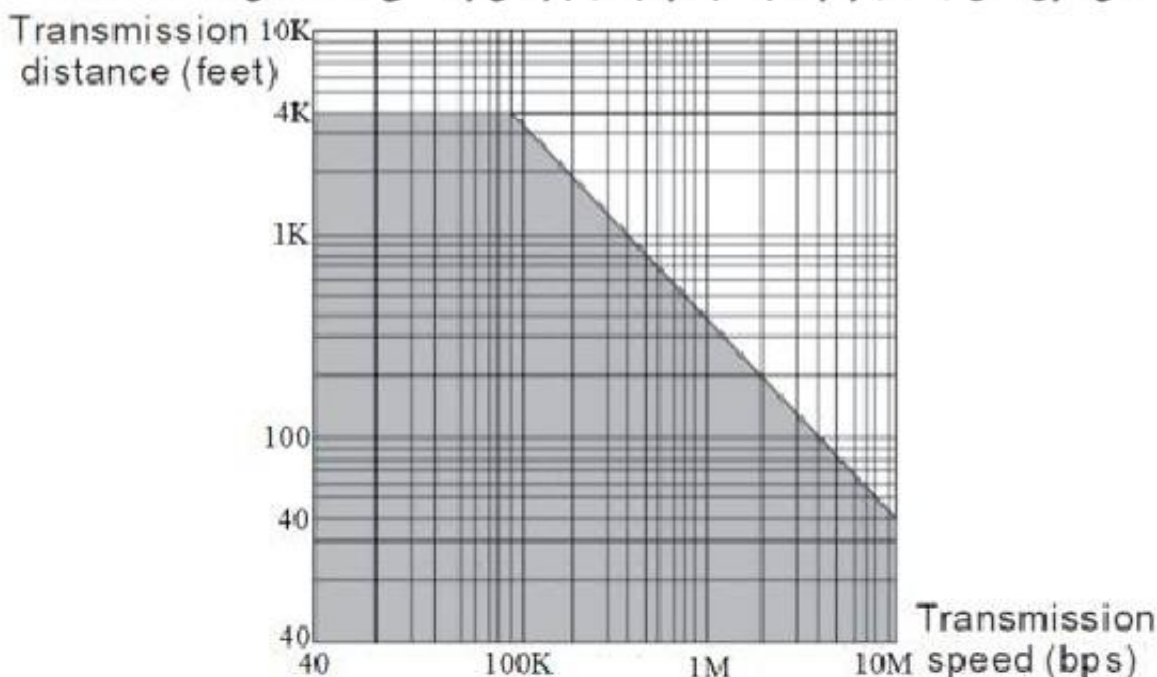
7-1 روش پیشنهادی برقراری شبکه ارتباطات RS-485 با پروتکل MODBUS توسط دلتا

قاعده‌ی کلی سیم‌بندی این پروتکل بر مبنای ارتباطات RS-232/RS-485 است. این گونه اتصال شبکه در برابر اتصال کوتاه حفاظت شده است و همچنین، در برابر نویز تا حدودی مصون می‌باشد. برای حفاظت بیشتر باید سیم‌های شبکه را دور از منابع نویز بالا قرار دهید و همچنین حتی‌الامکان سیم‌های شبکه را اتصال کوتاه نکنید. واسط ارتباطات RS-232 دارای ساختار اتصال یک به یک می‌باشد و معمولاً فاصله ارتباط بین دو وسیله کمتر از 3 متر است. برای اتصال دو وسیله با استفاده از شبکه RS-232 باید از کابل استاندارد دو زوج به هم تابیده شده‌ی شیلددار استفاده کنید. مسافت اتصالات شبکه سرعت بالای RS-485 طولانی‌تر بوده و همچنین، سرعت ارتباطات بین تجهیزات تحت شبکه‌ی RS-485 نیز بالا می‌باشد. تعداد ایستگاه‌هایی که در این شبکه استفاده می‌شود، تا حدود 254 وسیله تحت شبکه می‌باشد. در این شبکه میرایی (تضعیف) سیگنال بالا بوده و ممکن است در اثر اتصال زمین نامناسب مشکلاتی از قبیل افت ولتاژ در تجهیزات شبکه به وجود آید، بنابراین باید مسیر انتقال کابل سیگنال مناسب بوده و شیلد کابل سیگنال و مازول شبکه‌ی RS-485 به درستی زمین شوند. دقت کنید که امپدانس ترمینال‌های مازول شبکه نامناسب نباشد و در محدوده‌ی گفته شده در دیتاشیت شرکت سازنده باشد. تداخل نویز در این شبکه بالا بوده و همچنین، روش سیم‌بندی در این شبکه بسیار مهم است. درنهایت، کیفیت اتصالات در شبکه بسیار مهم بوده و این امر ممکن است در عملکرد شبکه تأثیر بگذارد. بنابراین، باید به تمام مطالب گفته شده با دقت توجه کنیم تا عملکرد مازول شبکه به درستی صورت گیرد. برای سیم‌بندی ارتباطات RS-485 باید به موارد ذکر شده در زیر توجه کنید:

- محدودیت در تعداد ایستگاه‌ها : تعداد ایستگاه‌هایی (تجهیزاتی) که می‌توان به DVP-PLC (PLC های شرکت دلتا) متصل کرد 254 وسیله می‌باشد. برای ارتباطات RS-485، واسط سخت‌افزار (مدار الکترونیکی واسط ارتباطات RS-485) حداکثر با 16 ایستگاه سازگار می‌باشد. اگر در شبکه‌ای بیشتر از 16 ایستگاه مورد نیاز بود، باید از یک ریپیتر (تکرارکننده) RS-485 استفاده کنید. مدل تکرارکننده‌ی RS-485 ساخته شده توسط شرکت دلتا IFD-8510 است. هر ریپیتر 16 ایستگاه را پشتیبانی می‌کند. با این روش کاربر می‌تواند تا 254 ایستگاه را به شبکه ارتباطات RS-485 متصل کند ($16 \times 16 = 256$).

- **محدودیت مسافت بین ایستگاه‌ها:** در ارتباطات RS-485، یک تابع وجود دارد که قادر به ارسال دیتا با همان کیفیت با حداکثر طول کابل می‌باشد که این تابع یک تقویت کننده‌ی سیگنال برای ارسال به مسافت‌های طولانی است. اگر در اتصالات شبکه از حداکثر طول کابل مجاز استفاده کنید، ممکن است که نویز و سیگنال‌های اعوجاج بر روی دیتای ارسال شده تأثیر بد گذاشته و دستگاه بد عمل کند. برای ارتباط بین شبکه‌ها باید از کابل شماره‌ی 24AWG معادل 0.51mm دو زوج تاییده شده بهم از جنس مس استفاده کنید. وقتی که نرخ تبادل داده در شبکه کمتر از 90Kbit/s بود، طول کابل باید کمتر از 1200 متر باشد. اگر نرخ تبادل داده در شبکه برابر با 600Kbit/s بود، طول کابل باید کمتر از 200 متر باشد. قطر کابل (نمره‌ی سیم) در طول کابل بسیار مؤثر است. اگر قطر کابل بیشتر شود، طول کابل اتصالات بین تجهیزات کمتر خواهد شد.

در واسط ارتباطات استاندارد RS-485 یک وابستگی بین سرعت تبادل داده (bps) و مسافت تبادل داده یا همان طول کابل (foot) وجود دارد. در نمودارهای زیر، این وابستگی کاملاً نشان داده شده است:



- **محدودیت در طول کابل:** برای سیم‌بندی شبکه باید از کابل‌های دو زوج تاییده شده بهم شیلدار استفاده کرد. دلیل این امر، کیفیت بالای این کابل بوده و این کابل از تداخل امواج بر روی سیگنال دیتا تا حد بسیار زیادی جلوگیری می‌کند. اگر از کابل‌های با کیفیت پایین‌تر از قبیل کابل‌های دو رشته‌ی بهم تاییده شده‌ی PVC استفاده کنید، تضعیف سیگنال دیتا بزرگتر شده و مسافت (طول کابل) مورد استفاده برای ارتباط ایستگاه‌ها به شدت کم می‌شود. علاوه، نویز به سادگی بر روی دیتای ارسالی توسط کابل تأثیر منفی می‌گذارد. بنابراین، توصیه می‌شود که از کابل‌های با کیفیت بالا از قبیل کابل دو

زوج تاییده شده‌ی پلی اتیلن استفاده کنید. لازم به ذکر است که اگر سرعت تبادل داده پایین بوده و همچنین نویز در محیط نیز کم باشد می‌توان از کابل دو زوج بهم تاییده شده‌ی PVC استفاده کرد. دقت کنید که در این گونه مصارف طول کابل باید کمتر از 5 متر باشد. اگر فاصله بین ایستگاه‌ها برای تبادل دیتا خیلی زیاد باشد، در بین راه (بین دو ایستگاه) باید از ریپتر RS-485 استفاده کنید. یک مدل از این ریپترها IFD-8510 می‌باشد که توسط شرکت دلتا ساخته شده است. در این گونه مصارف ریپتر کار تقویت سیگنال را انجام می‌دهد.

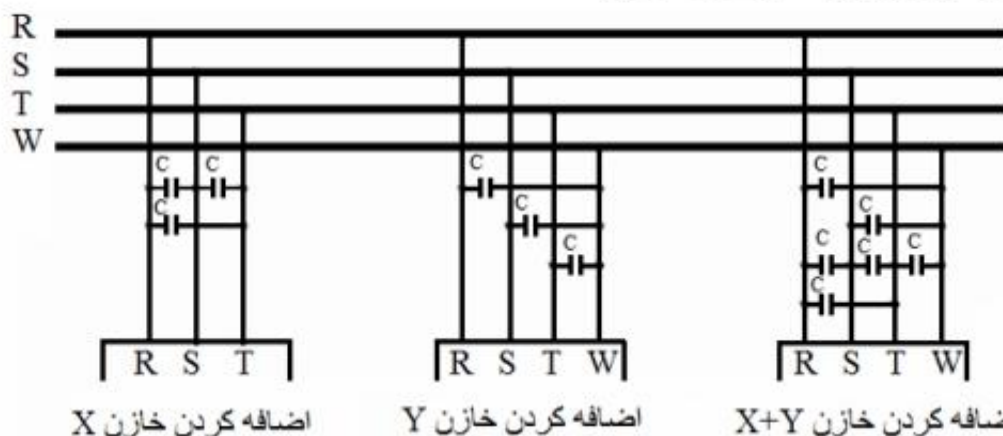
- تکنولوژی سیم‌بندی: برای سیم‌بندی RS-485، گروه‌ها باید نزدیک ایستگاه Master باشند. در کل، پیشنهاد می‌شود که برای سیم‌بندی شبکه‌ی RS-485 از توپولوژی با ساختار زنجیری استفاده کنید. توپولوژی، نوعی ساختار اتصال ایستگاه‌های شبکه به یکدیگر می‌باشد. در توپولوژی RS-485 باید از ساختار ایستگاه به ایستگاه استفاده کرد، این به آن معنی است که ایستگاه‌ها باید به ترتیب ایستگاه 1 به ایستگاه 2 و ایستگاه 2 به ایستگاه 3 و ایستگاه 3 به ایستگاه 4 و ... متصل شوند. شما مجاز به استفاده از توپولوژی با ساختار شاخه و یا حلقه نمی‌باشید.

- زمین کردن سیگنال (SG): اگرچه در اتصالات شبکه‌ی RS-485 از کابل دو زوج بهم تاییده شده استفاده می‌شود، اما تداخل امواج به عنوان نویز به سادگی بر روی دیتای قابل حمل توسط کابل تأثیر بد گذاشته و برای جلوگیری از این تأثیر مخرب باید از CMV (Common Mode Voltage) در اتصالات استفاده کرد. ولتاژ CMV بین ایستگاه‌های شبکه‌ی RS-485 نباید از حد مجاز ولتاژ IC تبادل داده‌ی RS-485 تجاوز کند. اگر CMV از محدوده‌ی ولتاژ کاری IC تجاوز کند، کار شبکه RS-485 متوقف شده و این شبکه استوپ می‌شود.

اما مهم نیست که درجه‌ی CMV چه مقدار باشد، ما می‌توانیم با اتصال به زمین (SG) ایستگاه‌های کاری در شبکه مقدار CMV را کاهش دهیم. برای اتصال به زمین باید از کابل دورشته بهم تاییده شده‌ی شیلددار استفاده کرد، به این صورت که برای کاهش CMV باید شیلد کابل را به زمین متصل نماییم. این روش سیم‌بندی سیستم را در برابر نویز مصون کرده و عملکرد شبکه را بهبود می‌بخشد.

- مقاومت ترمینال: تمام کابل‌ها دارای امپدانس مشخصی می‌باشند. امپدانس کابل دو زوج بهم تاییده شده 120 اهم است. وقتی که سیگنال به ترمینال ارسال می‌شود و امپدانس ترمینال نسبت به امپدانس مشخص شده تفاوت داشته باشد، انعکاس سیگنال از فرم طبیعی خود خارج خواهد شد. این وضعیت برای کابل‌های کوتاه (کمتر 5 متر) رخ نمی‌دهد ولی اگر طول کابل افزایش یابد، این وضعیت را به خوبی می‌توان درک کرد. از این رو، به یک مقاومت ترمینالی (مقاومت ترمینیتور یا همان مقاومت بستن اول و آخر گره در دو سمت شبکه) نیاز داریم تا شبکه به درستی عمل کند.

- **روش‌های کاهش نویز:** وقتی که شبکه‌ی RS-485 را مطابق با مطالب گفته شده در بالا متصل کردید و با مقاومت 120 اهمی (مقاومت ترمینیتور که در دو سمت ابتدا و انتهای شبکه متصل می‌شود) دو سمت باز شبکه را بستید، سپس نوبت به کابل‌های تداخل امواج (نویز) می‌رسد. اگر تداخل امواج پیوسته باشد به آن معنی است که یک منبع نویز قوی در نزدیکی شبکه وجود دارد. بعلاوه، ممکن است که کابل ارسال دیتا نیز در کنار منبع نویز قوی از قبیل شیر برقی، اینورتر (درایور موتورهای AC)، سرودرایوهای AC و دیگر تجهیزات قدرت و همچنین، خط‌های انتقال ولتاژ قدرت با جریان و توان بالا باشد. بهترین روش برای کاهش نویز، اضافه کردن یک جلوگیری کننده‌ی نویز (noise suppressor) به منبع نویز می‌باشد. شکل زیر، یک روش جلوگیری از نویز برای اینورتر، سرودرایو AC و دیگر تجهیزات قدرت را نشان می‌دهد. ظرفیت خازن‌های X، خازن‌های Y، خازن‌های X+Y برای ولتاژ 630 VAC برابر با $C=0.22\mu F \sim 0.47\mu F$ می‌باشد $(C=0.22\mu F \sim 0.47\mu F/AC630V)$.



در مجموع، کابل ارتباطات RS-485 با تکنولوژی کابل دو زوج بهم تابیده شده‌ی شیلدار ساخته شده و ارسال سیگنال توسط اختلاف پتانسیل بین دو زوج کابل انجام می‌شود. از اینرو، به این سیستم، ارسال تفاضلی گفته می‌شود. تداخل امواج در مد تفاضلی بین دو رشته سیم کابل ارسال می‌شود که در این صورت تداخل امواج بین دو سیم متقارن می‌باشد. برای کاهش این گونه تداخل امواج می‌توان از مقاومت تثبیت کننده در کنار کابل دو زوج بهم تابیده شده استفاده کرد. یک روش دیگر، تداخل امواج نامتقارن بوده که تداخل امواج میان کابل ارتباطات و زمین رخ می‌دهد. به این نوع، مد تداخل امواج مشترک گفته می‌شود. برای رفع کردن مد تداخل امواج مشترک باید از روش‌های زیر استفاده کرد:

1. باید از کابل دورشته‌ی بهم تابیده شده‌ی شیلدار استفاده کنید و از اتصال شیلد به زمین اطمینان حاصل نمایید.

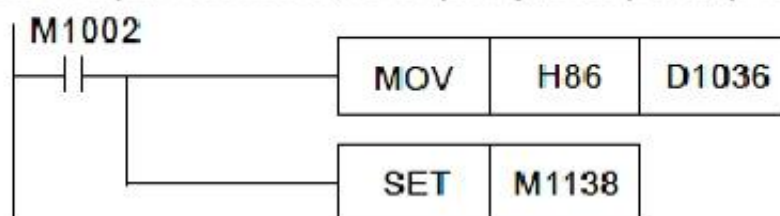
2. برای انتقال کابل از نقاطی که تجهیزات و کابل‌های الکتریکی قوی (آمبر بالا و توان بالا) وجود دارد، باید از لوله گالوانیزه استفاده کنید.
3. در هنگام سیم‌بندی دقت کنید که مازول شبکه‌ی RS-485 و همچنین، کابل ارسال دیتا در کنار خط‌های انتقال ولتاژ بالا نباشند.
4. از یک مدار تثبیت کننده‌ی خطی و یا منبع تغذیه‌ی سوئیچینگ با کیفیت بالا استفاده کنید. دقت کنید که ریبِل (موج) منبع تغذیه سوئیچینگ باید کمتر از 50mV باشد.

7-2 اصلاح فرمت ارتباطات پورت COM

7-2-1 اصلاح فرمت ارتباطات پورت COM1

در این مثال، می‌خواهیم فرمت ارتباطات پورت COM1 را اصلاح نماییم. برای اصلاح کردن فرمت ارتباطات پورت COM1 باید یک قطعه کد را به برنامه‌ی نوشته شده اضافه نماییم. این قطعه کد در شکل زیر به زبان برنامه‌نویسی نردبانی نشان داده شده است. وقتی PLC را از حالت STOP به حالت RUN سوئیچ می‌کنید، فلگ M1138 برای نخستین اسکن برنامه فعال (ON) خواهد شد. اگر فلگ M1138 فعال (ON) شود، برنامه اصلاحات انجام شده برای پورت COM1 را مطابق با مقدار D1036 تنظیم می‌کند.

فرمت ارتباطات اصلاح شده برای پورت COM1 در این مثال به صورت ASCII mode, 115200 bps, 7data bits, even parity, 1stop bit, (115200, 7, E, 1) می‌باشد.



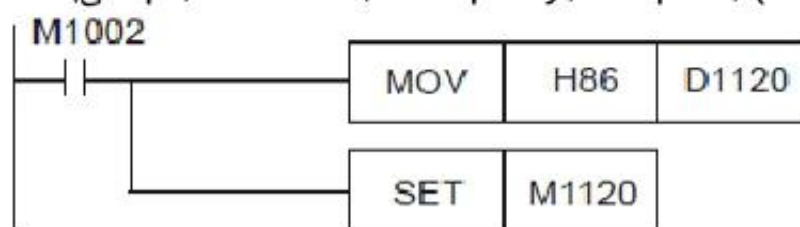
بعد از اینکه فرمت ارتباطات اصلاح شد، وقتی که PLC از حالت RUN به STOP سوئیچ شود، تنظیمات اصلاح شده بطور کامل و دست نخورده در PLC باقی خواهد ماند. اگر تغذیه‌ی PLC را قطع و سپس مجدداً وصل نمایید، فرمت ارتباطات اصلاح شده به تنظیمات پیش‌فرض کارخانه باز خواهد گشت.

7-2-2 اصلاح فرمت ارتباطات پورت COM2

در این مثال، می‌خواهیم فرمت ارتباطات پورت COM2 را اصلاح نماییم.

برای اصلاح کردن فرمت ارتباطات پورت COM2 باید یک قطعه کد را به برنامه‌ی نوشته شده اضافه نماییم. این قطعه کد در شکل زیر به زبان برنامه‌نویسی نردبانی نشان داده شده است. وقتی PLC را از حالت STOP به حالت RUN سوئیچ می‌کنید، فلگ خاص M1120 در نخستین اسکن برنامه فعال (ON) خواهد شد، اگر فلگ M1120 فعال (ON) شود، برنامه اصلاحات انجام شده برای پورت COM2 را مطابق با مقدار D1120 تنظیم خواهد کرد.

اصلاح فرمت ارتباطات پورت COM2 در این مثال به صورت ASCII mode, 576000 bps, 7data bits, even parity, 1stop bit, (57600, 7, E, 1) می‌باشد.



اگر از پورت COM2 به صورت یک ترمینال Slave استفاده شده است، باید از وجود نداشتن دستورالعمل ارتباطات در برنامه مطمئن شوید.

بعد از اینکه فرمت ارتباطات پورت COM2 در PLC اصلاح شد، وقتی که PLC از مد RUN به مد STOP سوئیچ شود، فرمت ارتباطات اصلاح شده بدون هیچ تغییر باقی مانده و فرمت ارتباطات اصلاح شده حفظ خواهد شد.

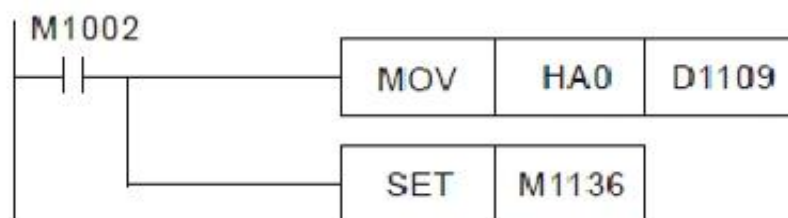
اگر تغذیه PLC را قطع و مجدداً آن را وصل نمایید، فرمت ارتباطات اصلاح شده به تنظیمات پیش‌فرض باز خواهد گشت.

7-2-3 اصلاح فرمت ارتباطات پورت COM3

در این مثال، می‌خواهیم فرمت ارتباطات پورت COM3 را اصلاح نماییم.

فرمت ارتباطات پورت COM3 ثابت بوده و برابر با 7data bits, even parity, 1stop bit می‌باشد. برای اصلاح کردن نرخ تبادل داده برابر با 38400bps برای پورت COM3 باید قطعه کد نوشته شده به زبان برنامه‌نویسی نردبانی را به برنامه نوشته شده اضافه نماییم. وقتی که PLC را از حالت STOP به حالت RUN سوئیچ کردید، فلگ M1136 برای نخستین بار اسکن برنامه فعال (ON) خواهد شد. اگر فلگ M1136 فعال (ON) شود، برنامه اصلاحات انجام شده برای پورت COM3 را مطابق با مقدار D1109 تنظیم می‌کند.

نرخ تبادل داده‌ی (سرعت تبادل داده) اصلاح شده برای پورت COM3 در این مثال برابر با 38400 bps می‌باشد.



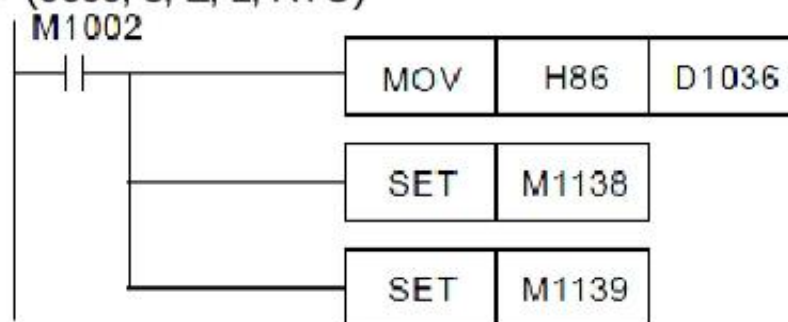
بعد از اینکه فرمت ارتباطات اصلاح شد، وقتی که PLC را از حالت RUN به حالت STOP سوئیچ نماییم، تنظیمات اصلاح شده بدون هیچ تغییر در حافظه‌ی PLC باقی خواهد ماند. اگر تغذیه PLC را قطع و سپس مجدداً وصل نمایید، فرمت ارتباطات اصلاح شده به تنظیمات پیش فرض کارخانه باز خواهد گشت.

7-2-4 فعال کردن مد RTU برای پورت COM1 و COM2

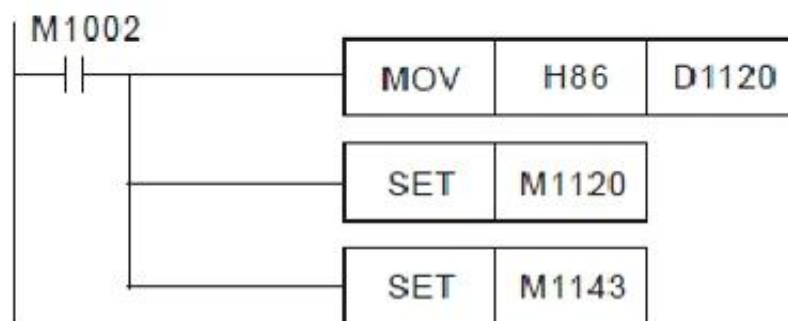
در این مثال، می‌خواهیم مد RTU را برای پورت COM1 و COM2 تنظیم نماییم. هر دو پورت COM1 و COM2 از مدهای ASCII/RTU پشتیبانی می‌کنند. تنظیم مد در پورت COM1 توسط فلگ خاص M1139 و در پورت COM2 توسط فلگ خاص M1143 صورت می‌گیرد. وقتی که وضعیت فلگ‌های M1139، M1143 فعال (ON) باشد، مد RTU برای پورت‌های COM1 و COM2 فعال شده و وقتی که وضعیت این فلگ‌ها OFF باشد، مد ASCII برای این پورت‌ها فعال می‌شود.

در قطعه کدهای نوشته در شکل زیر به زبان نردبانی چگونگی تنظیم مد RTU نشان داده شده است.

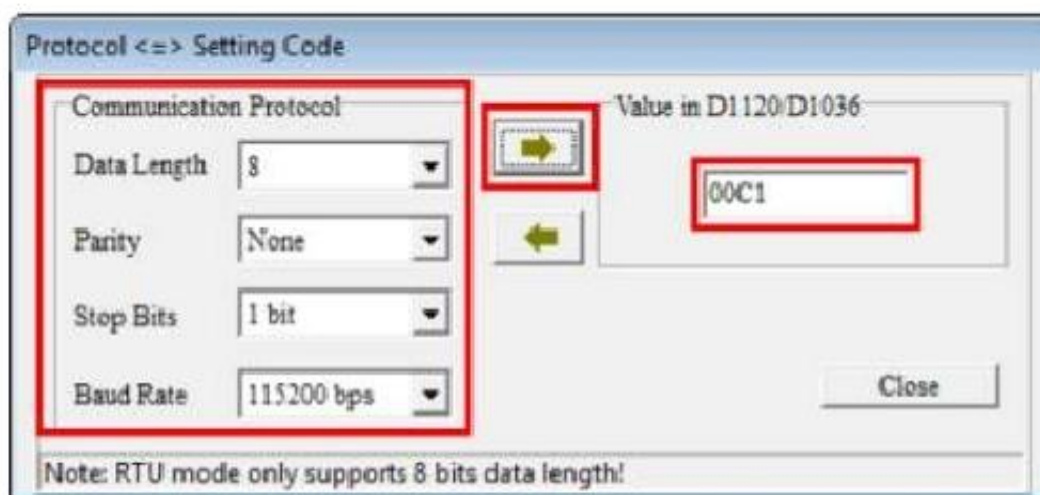
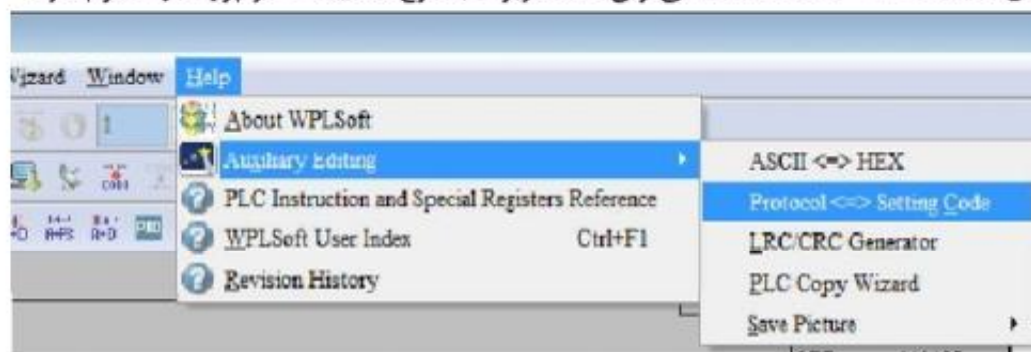
COM1 → (9600, 8, E, 1, RTU)



COM2 → (9600, 8, E, 1, RTU)



5-2-7 تنظیم پورت COM توسط ابزار جادوی نرم‌افزار WPLSoft و ISPSOft
 در نرم‌افزار WPLSoft با استفاده از منوی، Help >> Auxiliary Editing >> Protocol <=> Setting Code می‌توان عدد مربوط به نوع تنظیمات هر پورت را معلوم کرد.



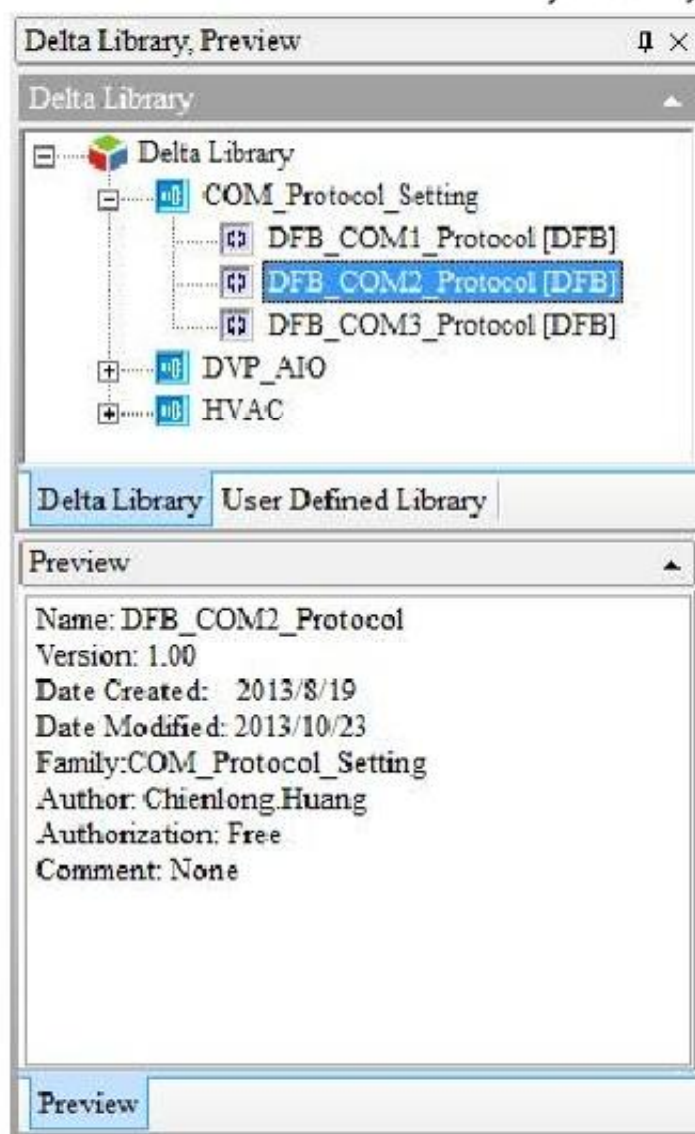
عدد بدست آمده برای پورت COM1 را باید در رجیستر D1036 قرار داده و با یک کردن بیت M1138 این مقدار را در CPU ذخیره کرد.

رجیسترهای مربوط به M1120, D1120, COM2 و برای M1136, D1109, COM3 می‌باشند.

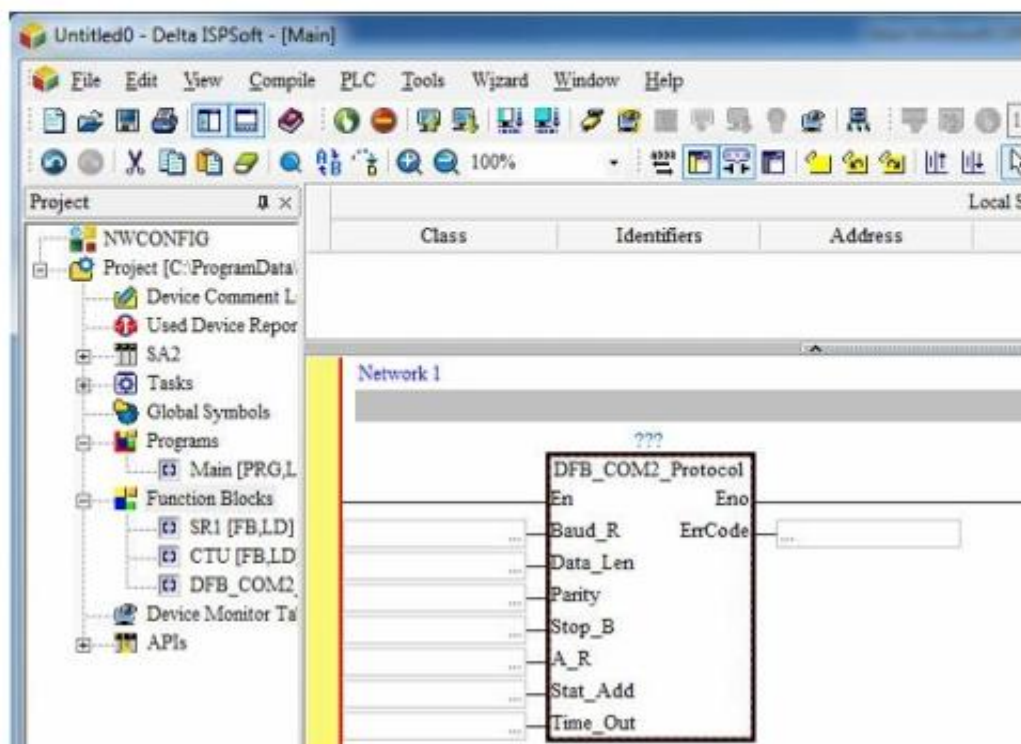
با یک کردن بیت‌های M1139, M1143, M1320 برای COM1, COM2, COM3 می‌توان پروتکل پورت مورد نظر را از ASCII به RTU تبدیل کرد.

برای ارسال دستور، با یک کردن بیت‌های M1312, M1122, M1316 برای COM1, COM2, COM3 تعیین می‌کنیم که دستور از کدام COM ارسال شود.

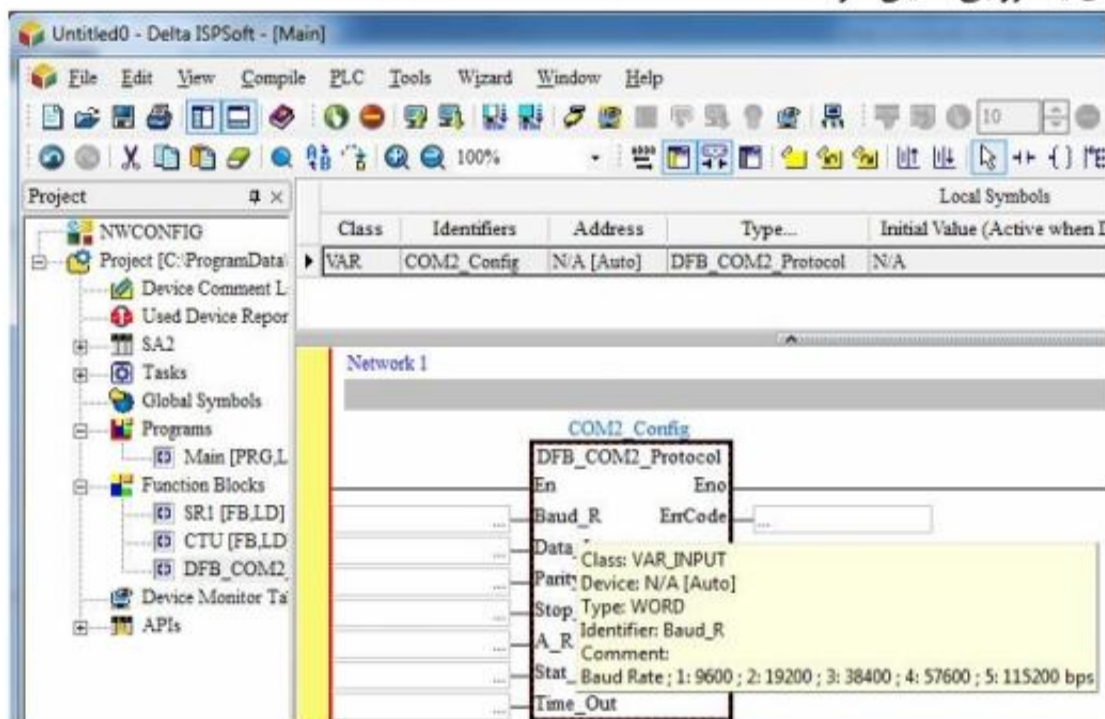
در نرم‌افزار ISPSOft می‌توان از کادر Delta Library فانکشن بلاک‌های از پیش برنامه‌نویسی شده توسط شرکت دلتا را مشاهده کرد.



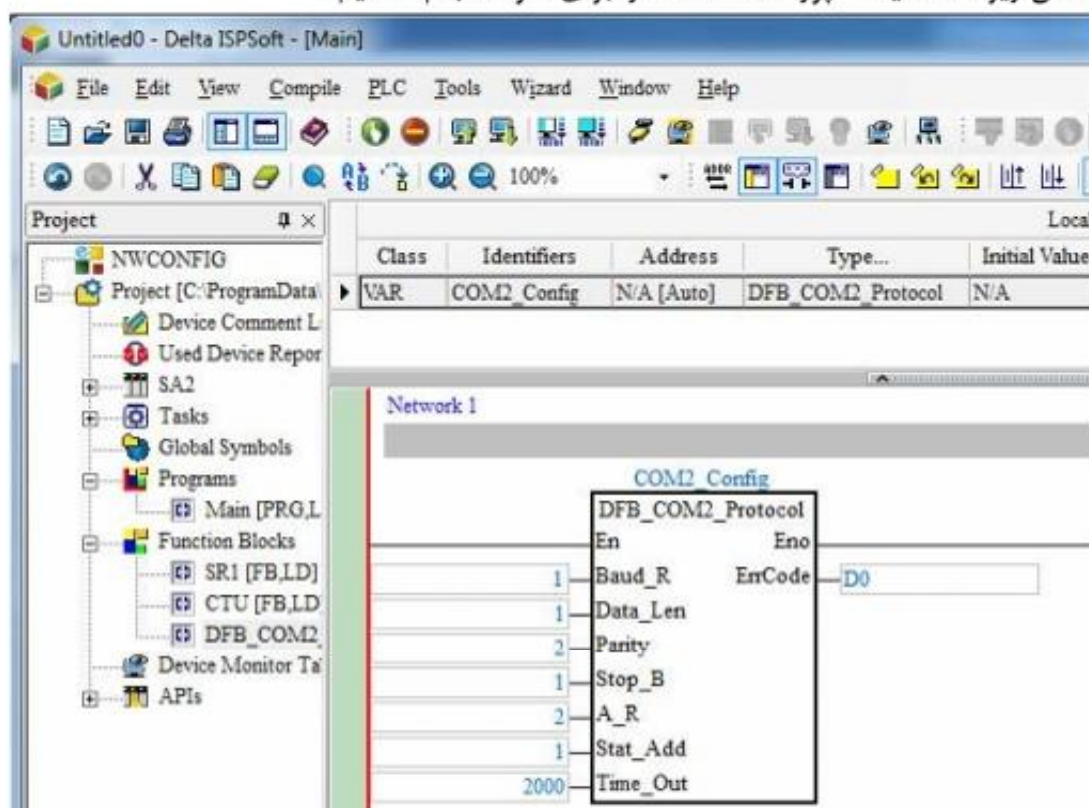
فانکشن بلاک پورت COM مورد نظر را می‌توان با روش کشیدن و رها کردن به برنامه اضافه کرد.



سپس موس را بر روی نوشته‌ی هر ورودی یا خروجی ببرید تا اطلاعات مربوط به نحوه‌ی تنظیم آن ورودی یا خروجی نمایان شود.



با توجه به اطلاعات ارائه شده برای هر ورودی یا خروجی به تنظیم مناسب پورت مورد نظر پردازید. در شکل زیر، ما تنظیمات پورت COM2 را برای نمونه انجام داده‌ایم:

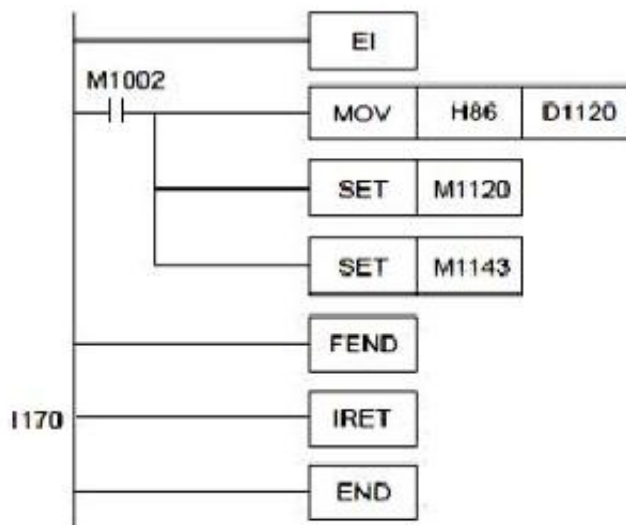


7-2-6 استفاده از وقفه ارتباطات پورت COM2

در PLC های سری E112, SV, EH3, SV2 وقتی که عمل دریافت داده در مد Slave تکمیل شد، وقفه‌ی I170 راه اندازی می‌شود.

معمولاً وقتی که ترمینال ارتباطات PLC در مد Slave است، PLC به صورت آنی مقادیر وارد شده‌ی داده‌ی ارتباطات را پردازش نخواهد کرد. اما داده‌های ارتباطات ورودی بعد از اجرا دستورالعمل END پردازش می‌شوند. از این رو، وقتی که زمان اسکن برنامه خیلی طولانی باشد و اگر شما نیاز به پردازش آنی داده‌ی ارتباطات داشته باشید، باید برای این مهم (موضوع) از وقفه‌ی I170 استفاده کنید.

در زیر، مثالی نحوه‌ی استفاده از وقفه‌ی I170 بعد از تکمیل شدن عمل دریافت داده در مد Slave آورده شده است:



با استفاده از وقفه‌ی I170 در برنامه، وقتی که پورت COM2 در مد Slave بوده و داده‌ی ارتباطات در حال وارد شدن به این پورت باشد، PLC داده‌ی وارد شده را پردازش کرده و فوراً به آن پاسخ می‌دهد.

وقتی که وقفه‌ی I170 در حال استفاده باشد، PLC برنامه را به صورت آنلاین به‌روزرسانی نمی‌کند. در این حالت، زمان اسکن برنامه در PLC طولانی‌تر می‌شود.

7-3 تابع اتصال مودم

از فلگ‌های خاص M1184~M1188 برای تابع اتصال مودم به PLC و ارتباط کامپیوتر با PLC از طریق اینترنت استفاده می‌شود. در شکل زیر اتصال مودم به PLC و ارتباط کامپیوتر با PLC از طریق اینترنت آمده است.



از نرم‌افزار برنامه‌نویسی WPLSoft برای برنامه‌ریزی از راه دور PLC و همچنین، عیب‌یابی می‌توان استفاده کرد. همانطور که در شکل بالا مشاهده می‌کنید، برای اتصال PLC به مودم باید از پورت ارتباطات DVP-F232 که یک واسطه بین PLC و مودم است استفاده کرد و از این رو، باید از مودمی استفاده شود که از پورت ارتباطات RS-232 پشتیبانی می‌کند.

فلگ‌های M خاص برای اتصال مودم به PLC های سری EH2, EH3, SV2 به شرح زیر می‌باشند:

Device No	Function	Note
M1184	Enable MODEM	On: The following actions are valid
M1185	Initialize MODEM	Off: Initialization is completed
M1186	Fail to initialize MODEM	Off: M1185 = On
M1187	MODEM initialization is completed	Off: M1185 = On
M1188	Shows if MODEM is connected	On: Connecting

توجه داشته باشید که وقتی که PLC در وضعیت RUN و یا وضعیت STOP است، فلگ‌های خاص گفته شده در بالا قابل اجرا می‌باشند. این به آن معنی است که اگر PLC در هر یک از وضعیت‌های RUN و یا STOP باشد از طریق مودم و اینترنت می‌توان به PLC متصل شد.

مراحل اتصال به PLC از طریق مودم و شبکه‌ی اینترنت به شرح زیر می‌باشد:

مرحله 1 ← ابتدا باید فلگ M1184 فعال (ON) شود. از این فلگ برای فعال کردن اتصال PLC MODEM استفاده می‌شود.

مرحله 2 ← در این مرحله فلگ M1185 را باید فعال (ON) کرد. از این فلگ برای مقداردهی اولیه و راه‌اندازی مودم متصل شده به PLC استفاده می‌شود.

مرحله 3 ← در این مرحله باید با استفاده از وضعیت فلگ‌های M1187, M1186 چک کرد که آیا مودم با موفقیت راه‌اندازی شده است، یا خیر.

مرحله 4 ← در مرحله‌ی آخر باید منتظر بمانید که اتصال PLC به مودم و در نهایت، به اینترنت برقرار شود.

برای کار با مودم به نکات زیر توجه داشته باشید:

برای اتصال PLC به مودم، به یک کارت توسعه‌ی RS-232 نیاز است. اگر کارت توسعه‌ی RS-232 وجود نداشته باشد، تمام فلگ‌های خاص گفته شده در جدول بالا معتبر نمی‌باشد.

بعد از اینکه مودم فعال شد (M1184=ON)، PLC ابتدا باید برای نخستین بار مودم را راه‌اندازی نماید که این عمل توسط فعال شدن فلگ M1185 صورت می‌گیرد (M1185=ON). اگر PLC موفق به راه‌اندازی مودم نشد، تابع auto-answering مودم فعال نخواهد شد.

بعد از اینکه مودم راه‌اندازی شد، مودم به صورت اتوماتیک وارد مد پاسخ‌دهی اتوماتیک (auto-answering) می‌شود.

اگر کنترل از راه دور کامپیوتر (Remote PC) غیرفعال باشد، مودم به صورت اتوماتیک وارد مد آماده به کار (Stand-by) می‌شود و اگر در این حالت کاربر مودم را خاموش کند، بعد از روشن کردن

مودم و یا فعال کردن کنترل از راه دور کامپیوتر مجدداً باید مودم را با استفاده از فلگ M1185 راه‌اندازی کرد.

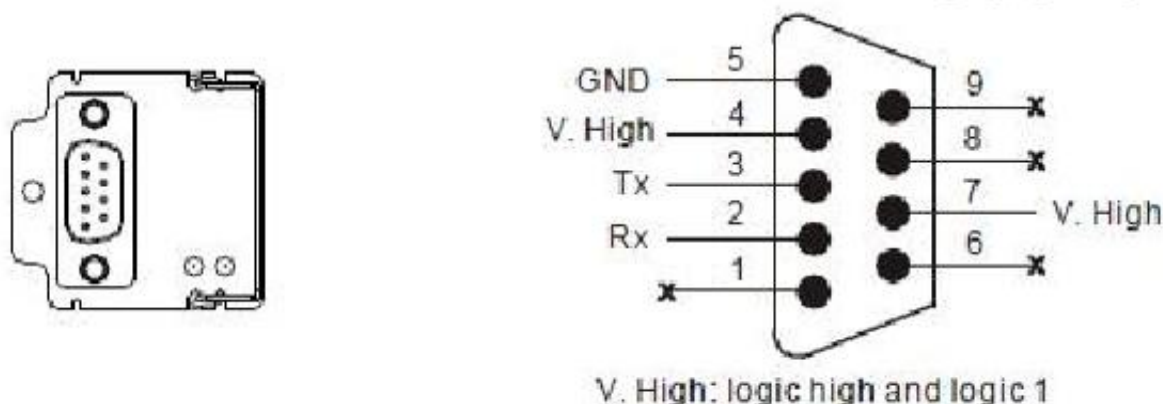
سرعت اتصال توسط PLC برابر با 9600bps تنظیم خواهد شد. این سرعت برای مودم ثابت بوده و تغییر نمی‌کند. بنابراین، باید دقت داشته باشید که مودمی را باید به PLC متصل کنید که از سرعت 9600bps پشتیبانی کند.

فرمت مقداردهی اولیه از PLC به مودم برای پارامترهای ATZ و ATSO برابر با یک است. برای اطلاع از جزئیات این پارامترهای مودم باید به دفترچه‌ی راهنمای مودم مراجعه نمایید.

اگر PLC در فرآیند مقداردهی اولیه (راه‌اندازی) مودم دچار خطا شود، شما باید از سوپر ترمینال در کامپیوتر برای مقداردهی اولیه و راه‌اندازی مودم استفاده کنید. به صورتی که مقادیر پارامترهای ATZ و ATSO را برابر با یک کنید. (برای اطلاع از جزئیات بیشتر super terminal در کامپیوتر به کتاب‌های کامپیوتری مراجعه نمایید).

1-3-7 ماژول DVP-F232 (RS-232 card)

ماژول DVP-F232 یک کارت توسعه RS-232 می‌باشد. در این ماژول از یک کانکتور نری 9 پین (DB-9) جهت ارتباطات با کامپیوتر استفاده شده است. مشخصات ترمینال‌های این کانکتور مطابق با شکل زیر می‌باشد:

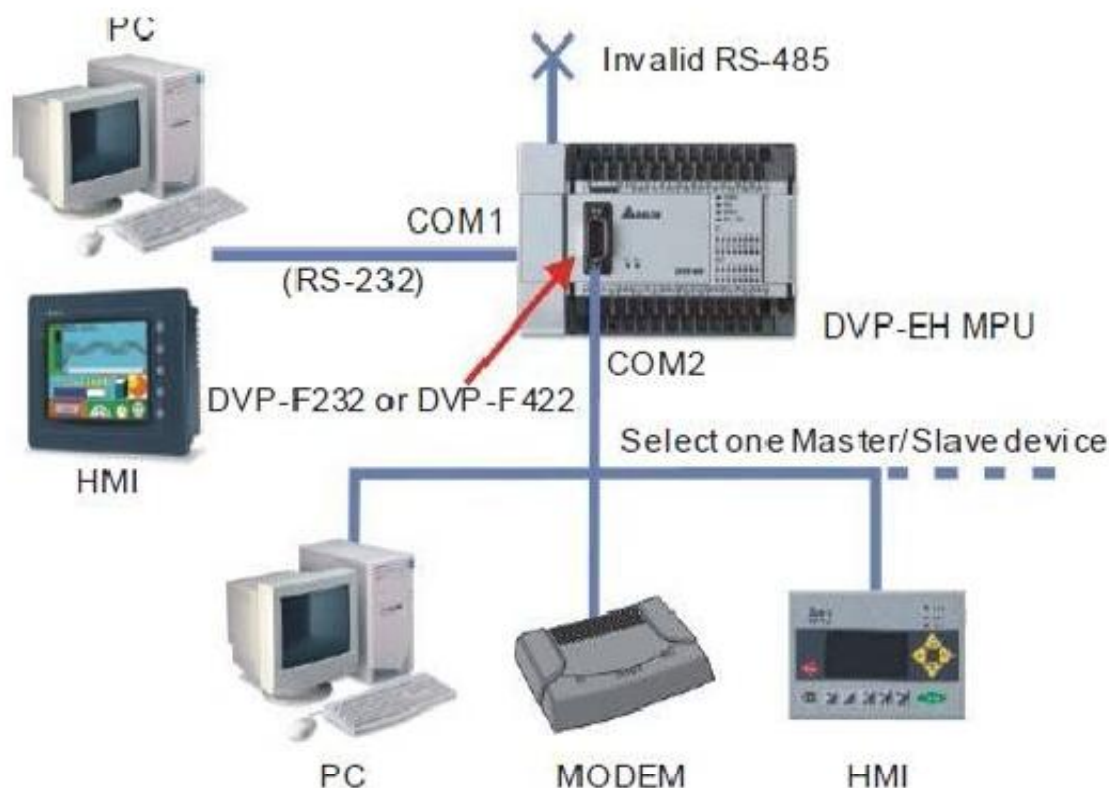


وقتی که از این پورت برای اتصال به کامپیوتر و یا HMI استفاده می‌کنید، فقط پین 2 و پین 3 آن که مربوط به سیگنال می‌باشد، قابل استفاده است.

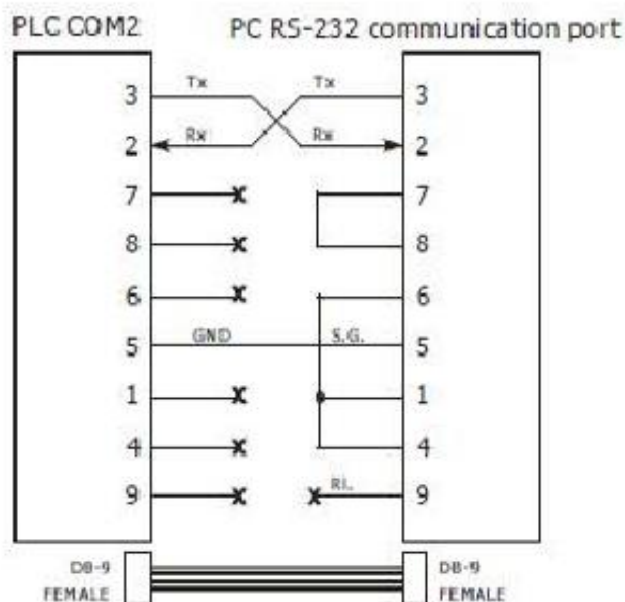
پروتکل ارتباطات پشتیبانی شده توسط این ماژول مطابق با جدول زیر می‌باشد:

آیتم	توضیحات
نام ماژول	DVP-F232
مشخصات ارسال داده	RS-232C
روش ایزوله کردن	این کارت ایزوله‌ای ندارد
فاصله ارسال داده	حداکثر فاصله‌ای که برای کابل ارتباطات می‌توان استفاده کرد، 15 متر می‌باشد
توان مصرفی ماژول	30mA/DC5V تغذیه شده توسط PLC MPU
پارامتر ارتباطات	مد
	سرعت تبادل داده
	بیت‌های داده
	بیت توقف
	بیت توازن
سیگنال ارتباطات	Master/Slave, semi-duplex, bidirectional data transmission / receiving
تجهیزات قابل اتصال به ماژول	110 ~ 115,200 bps (bit/sec)
	7 or 8
	1 or 2
	None, Even, Odd
TX (data transmission), RX (data receiving)	
تمام تجهیزات که با واسط RS-232C ارتباط برقرار می‌کنند	

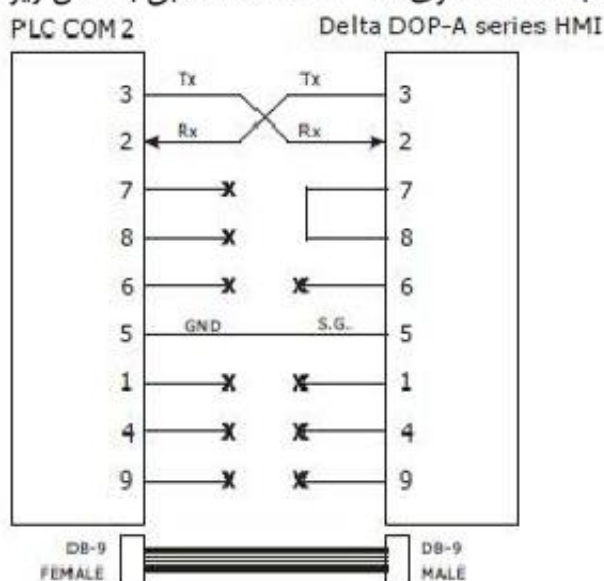
بر روی PLC به صورت پیش‌فرض پورت COM1 (RS-232) و COM2 (RS-485) نصب شده است. اگر می‌خواهید از پورت COM2 بعنوان پورت RS-232 استفاده کرده و به آن کامپیوتر و یا مودم متصل نمایید، باید از ماژول DVP-F232 استفاده نمایید. در این صورت، پورت COM2 (RS-485) پیش‌فرض PLC غیرفعال خواهد شد و این ماژول بعنوان پورت COM2 فعال می‌شود. به غیر از نوع واسط ارتباطات، دیگر توابع ارتباطات پورت COM2 برای ماژول DVP-F232 یکسان بوده و شما می‌توانید از آن بعنوان یک Slave و یا Master با سرعت تبادل داده بین 9600~115200bps استفاده نمایید. در شکل زیر، تجهیزات قابل اتصال به این کارت نشان داده شده است.



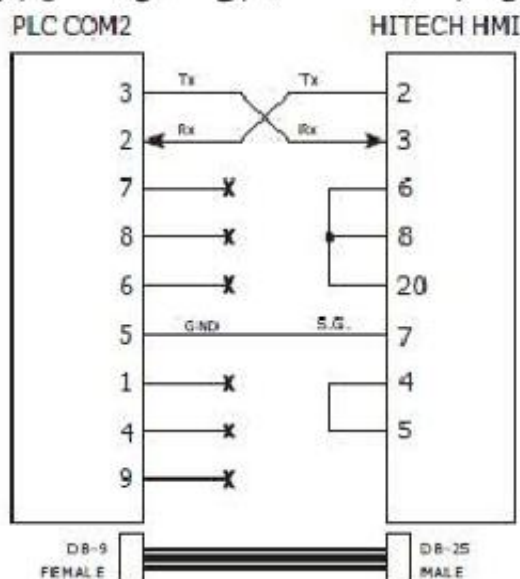
از این پورت می‌توان در کاربردهای با مد **Slave** استفاده کرد. سیم‌بندی اتصال این پورت به کامپیوتر مطابق با شکل زیر است:



نحوه‌ی اتصال این کارت به HMI سری DOP-A دلتا مطابق با شکل زیر است:



نحوه‌ی اتصال و سیم‌بندی این کارت به HMI هیتاچی مطابق با شکل زیر می‌باشد:

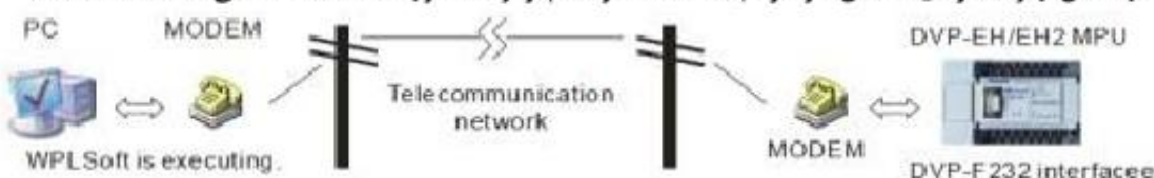


از این پورت می‌توان در کاربردهای با مد Master استفاده کرد. با استفاده از پورت DVP-F233، می‌توان PLC دلتا را از طریق مودم از راه دور کنترل کرد. برای این کار باید به PLC یک مودم متصل کرده و به کامپیوتر نیز یک مودم دیگر متصل نموده و سپس برنامه WPLSoft را اجرا کرد. از این طریق می‌توان برنامه‌ی PLC را از راه دور مانیتور کرده، ویرایش کرده و یا یک برنامه‌ی جدید را به PLC دانلود کرده و یا برنامه‌ی PLC را بر روی کامپیوتر آپلود کرد.

برای این کار باید کامپیوتر (PC Master) را از طریق پورت COM(RS-232) به مودم متصل نماییم.

یک خط تلفن را باید به مودم متصل کرده و همچنین، PLC MPU (Slave) را نیز به مودم متصل نماییم. مودم را در سمت PLC باید به پورت COM1(RS-232) متصل نماییم. این اتصال از طریق کابل RS-232 امکان‌پذیر است. دقت داشته باشید که مودم‌ها باید از پورت RS-232 پشتیبانی کنند تا بتوان کامپیوتر و PLC را به آنها متصل کرد. همچنین، شما باید در هر دو سمت کابل خط تلفن را به سوکت PHONE مودم‌ها متصل نماییم.

در شکل زیر، نحوه‌ی اتصال دو مودم به PLC و کامپیوتر به صورت شماتیک نشان داده شده است:



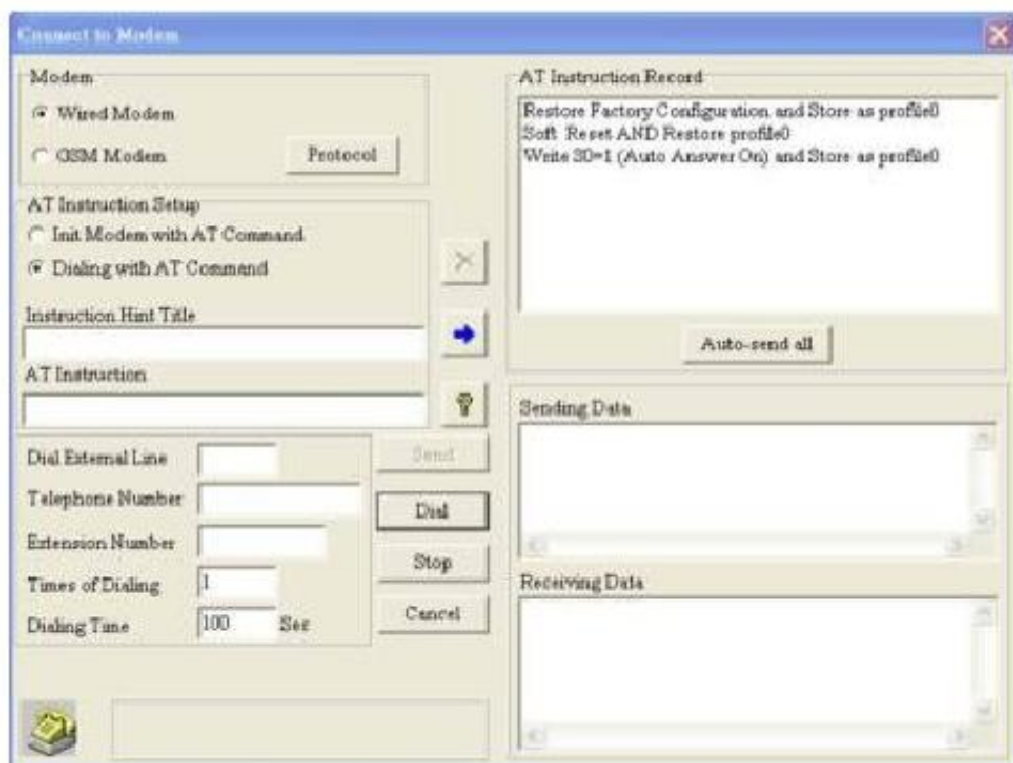
مراحل برقراری ارتباط با مودم در سمت PLC به شرح زیر می‌باشد:

مرحله‌ی 1 ← برای فعال کردن مودم ابتدا باید فلگ خاص M1184 را در سمت PLC فعال (On) نماییم.

مرحله‌ی 2 ← برای فعال کردن مقداره‌ی پارامترهای مودم توسط PLC باید فلگ خاص M1185 را در سمت PLC فعال (ON) نماییم.

مرحله‌ی 3 ← نتیجه‌ی بدست آمده از مودم توسط PLC بررسی خواهد شد. اگر ارتباط بین PLC و مودم با موفقیت برقرار شده باشد، فلگ خاص M1185 فعال (On) خواهد شد. ولی اگر خطایی در برقراری ارتباط بین PLC و مودم رخ داده باشد، فلگ خاص M1187 فعال (ON) خواهد شد.

مرحله‌ی 4 ← بعد از اینکه ارتباط PLC با مودم برقرار شد، نرم‌افزار WPLSoft آماده‌ی برقراری ارتباط با مودم می‌شود. در نرم‌افزار WPLSoft باید از گزینه‌ی Option زیرگزینه‌ی Modem Connection (دقت داشته باشید که قبل از این کار باید نرم‌افزار راه‌انداز سخت‌افزار مودم را بر روی کامپیوتر نصب کرده باشید) را انتخاب نماییم. در این حالت، پنجره‌ی dial-up باز خواهد شد. در شکل زیر، نمونه‌ای از تنظیمات و اطلاعات پنجره‌ی dial-up نشان داده شده است:



1. گزینه‌ی Dial External Line : تنظیم این گزینه تنها زمانی که شما از یک خط تلفن خارجی توسط یک ترمینال تلفن شماره‌گیری می‌نمایید، لازم است.
2. گزینه‌ی Telephone number : در این کادر شماره‌ی تلفن گیرنده (دریافت کننده) وارد می‌شود. هیچ‌گونه فاصله‌ای نباید بین کد شهر یا کشور با شماره تلفن وجود داشته باشد. به عنوان مثال، شماره و کد شهر یا کشور باید به صورت 88633626301 نوشته شود.
3. گزینه‌ی Extension number : شماره تلفن فرعی را فقط در هنگام نیاز تنظیم نمایید.
4. گزینه‌ی Dialing times : از این گزینه برای تنظیم زمان تماس و برقراری ارتباط مجدد در هنگام خطا در برقراری ارتباطات استفاده می‌شود.
5. بعد از اینکه تنظیمات را انجام دادید، برای شروع برقراری ارتباط بر روی دکمه‌ی Dial کلیک نمایید.

وقتی که برقراری ارتباط با dial-up با موفقیت انجام شد، یک پنجره با عنوان Modem is connecting ظاهر خواهد شد و شما می‌توانید برنامه‌ی داخل PLC را از طریق نرم‌افزار WPLSoft از راه دور مانیتور نمایید. وقتی که در سمت PLC یک سیگنال کنترل از راه دور آشکار شود، فلگ خاص M1188 فعال (On) خواهد شد. از این رو، شما می‌توانید برای اطلاع از برقراری ارتباط PLC با مودم وضعیت فلگ خاص M1188 را چک نمایید.

توجه داشته باشید که :

هنگامی که مودم در حال اتصال می‌باشد، نمی‌توانید سرعت تبادل داده را تنظیم نمایید. سرعت تبادل داده‌ی مودم در روی PLC ثابت بوده و قابل تغییر نمی‌باشد. این سرعت برابر با 9600bps است. مودمی که در سمت PLC قرار دارد، باید از تابع پاسخ اتوماتیک (AA) Auto Answer پشتیبانی کند. سرعت تبادل داده‌ی مودم در دو سمت PLC و کامپیوتر باید برابر با و یا کمتر از 9600bps باشد.

فلگ‌های خاص M برای برقراری ارتباط مودم با PLC‌های سری DVP-EH/EH2 به شرح زیر می‌باشند :

فلگ خاص	تابع	توضیحات
M1184	تابع فعال کننده‌ی مودم	اگر M1184=On شود، مودم متصل شده به PLC فعال می‌شود.
M1185	تابع فعال کردن پارامترها و مقداردهی اولیه‌ی مودم	بعد از اینکه مقداردهی و پیکربندی اولیه مودم توسط PLC تکمیل شدند این فلگ OFF می‌شود.
M1186	تابع خطا در پیکربندی و مقداردهی اولیه‌ی مودم	وقتی که خطایی در برقراری ارتباط PLC با مودم بوجود آید این فلگ On می‌شود.
M1187	تابع تکمیل شدن عمل مقداردهی اولیه‌ی مودم	وقتی که پیکربندی مودم توسط PLC تکمیل شد این فلگ On می‌شود.
M1188	تابع نمایش وضعیت جاری مودم	وقتی که PLC با مودم ارتباط برقرار کرد و متصل شد این فلگ On می‌شود.

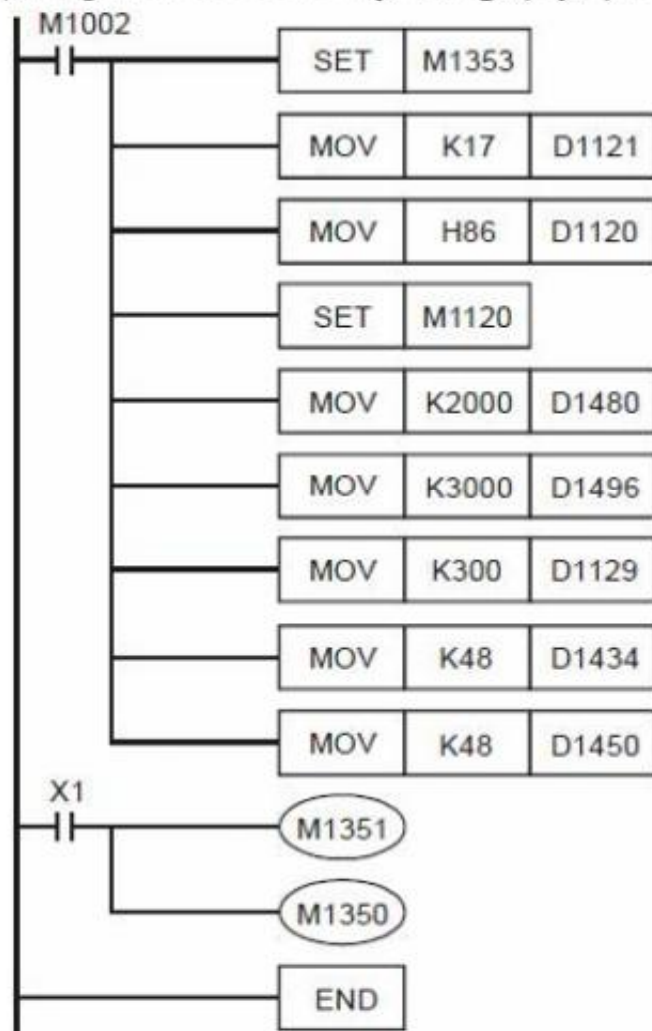
برای برقراری و اتصال PLC به مودم به کارت توسعه‌ی تابعی DVP-F232 نیاز دارید. تمام فلگ‌های خاص M لیست شده در جدول بالا بدون وجود کارت DVP-F232 معتبر نمی‌باشند. بعد از اینکه مودم را فعال کردید (M1184=On)، باید ابتدا مودم را مقداردهی اولیه نمایید (M1185=On). اگر مودم مورد استفاده در سمت PLC دارای تابع پاسخ اتوماتیک نباشد، PLC توانایی مقداردهی اولیه مودم را نخواهد داشت. بعد از اینکه مقداردهی اولیه مودم توسط PLC انجام شد، PLC به صورت اتوماتیک وارد مد پاسخ اتوماتیک خواهد شد.

اگر اتصال بین کامپیوتر از راه دور و مودم برقرار نباشد، مودم به صورت اتوماتیک وارد مد آماده به کار خواهد شد. اگر شما در این وضعیت (مد آماده به کار بودن مودم) مودم را خاموش کردید، باید بعد از روشن کردن مودم مجدداً بار دیگر آن را توسط PLC فعال کرده و مقداردهی اولیه نمایید. فرمت مقداردهی اولیه مودم توسط PLC به صورت ATZ, ATSO=1 می‌باشد.

7-4 پروتکل PLC LINK

7-4-1 تبادل داده در شبکه‌ی PLC LINK توسط فلگ M1353

می‌خواهیم 32 عدد Slave متصل شده به یکدیگر را در شبکه‌ی PLC LINK فعال کرده و تبادل دیتای برابر با 100 عدد داده را نیز در این شبکه توسط فلگ M1353 فعال نماییم.



فلگ خاص M1350 باید قبل از فعال شدن PLC LINK فعال شود. وقتی که PLC LINK اجرا شود، دیگر نمی‌توان فلگ خاص M1353 را ON/OFF کرد.

از رجیسترهای D1576~D1591 و D1480~D1495 برای خواندن داده از Slave ID#1~16 و از رجیسترهای D1592~D1607 و D1496~D1511 برای نوشتن داده در Slave ID#1~16 استفاده می‌شود. شماره‌های گفته شده به ترتیب Slave ID ها است. به عنوان مثال، رجیستر خاص D1480 به ID1، رجیستر خاص D1481 به ID2 و ... اختصاص داده می‌شود.

رجیسترهای خاص D1480~D1495, D1576~D1591, D1496~D1511, D1592~D1607 باید قبل از فعال شدن PLC LINK تنظیم شوند. در موقع اجرا PLC LINK شما فقط می‌توانید محتویات داخل این رجیسترهای خاص را اصلاح کنید و یا تغییر دهید. دقت داشته باشید که تغییرات در این رجیسترهای خاص بلافاصله مؤثر نخواهد بود و در نمونه‌برداری بعدی PLC LINK مؤثر می‌باشد. نگران این موضوع نباشید چون زمان نمونه‌برداری برابر با زمان اسکن برنامه توسط CPU می‌باشد.

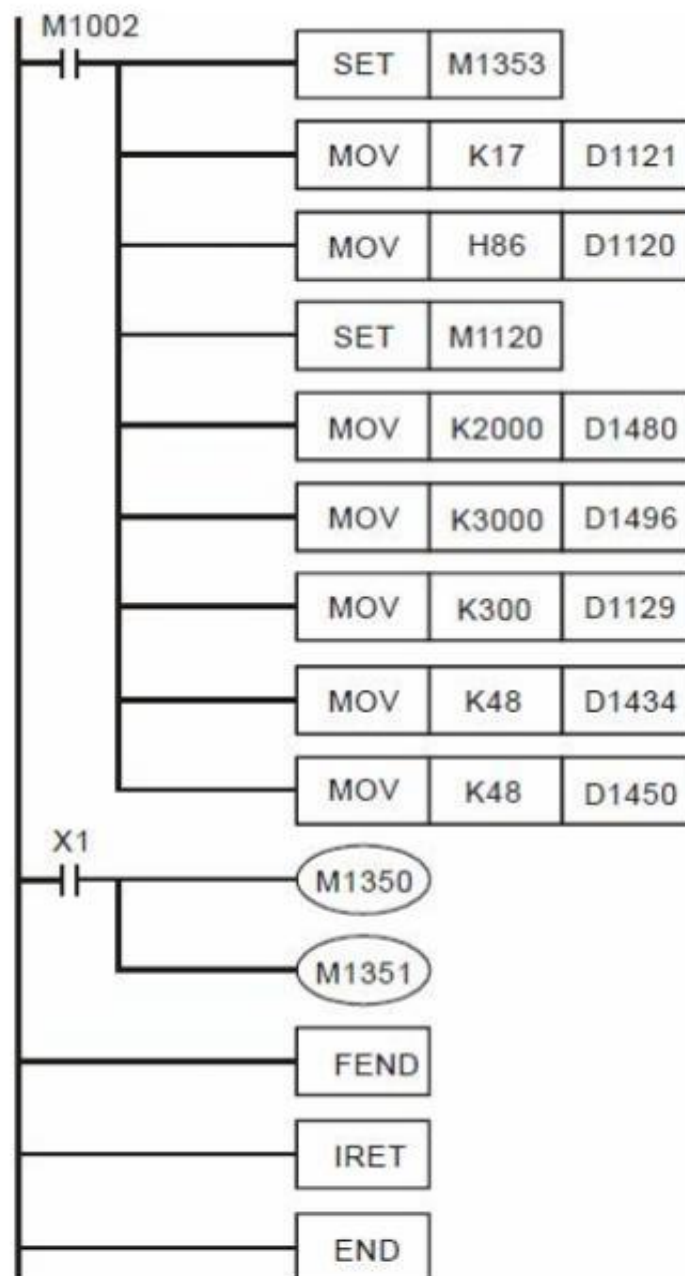
اگر ID# اختصاص داده شده به D1480~D1495, D1576~D1591, D1496~D1511, D1592~D1607 کمتر از 0 و یا بیشتر از 9900 بود، PLC به صورت اتوماتیک ID# را به 9900 اصلاح خواهد کرد.

اگر در هنگام اجرای PLC LINK فلگ خاص M1353 فعال نشود، محدوده‌ی داده برای تبادل دیتا D1434~D1449, D1544~D1559 (شماره‌ی داده خوانده شده از Slave ID#) و D1450~D1465, D1560~D1575 (شماره‌ی داده‌ی نوشته شده در Slave ID#) برابر با 0~16 خواهد شد. اگر تنظیمات رجیسترهای خاص D از این محدوده تجاوز کند، PLC به صورت اتوماتیک آن را به 16 باز خواهد گرداند. وقتی که M1353 فعال شود، محدوده‌ی داده برای تبادل دیتا 0~100 عدد داده خواهد شد. اگر تنظیمات رجیسترهای خاص از این محدوده تجاوز کند، PLC آن را به 100 باز خواهد گرداند.

می‌توانید مقادیر موجود در رجیسترهای خاص D1434~D1449, D1544~D1559, D1450~D1495, D1560~D1575 را در هنگام اجرای PLC LINK تغییر دهید. دقت داشته‌باشید که تغییرات در اجرای بعدی شبکه‌ی PLC LINK معتبر خواهد بود.

7-4-2 تبادل داده در شبکه‌ی PLC LINK توسط فلگ M1353 و وقفه‌ی ارتباطات I170

می‌خواهیم شبکه‌ی PLC LINK را با فلگ خاص M1353 و وقفه‌ی I170 اجرا نماییم.



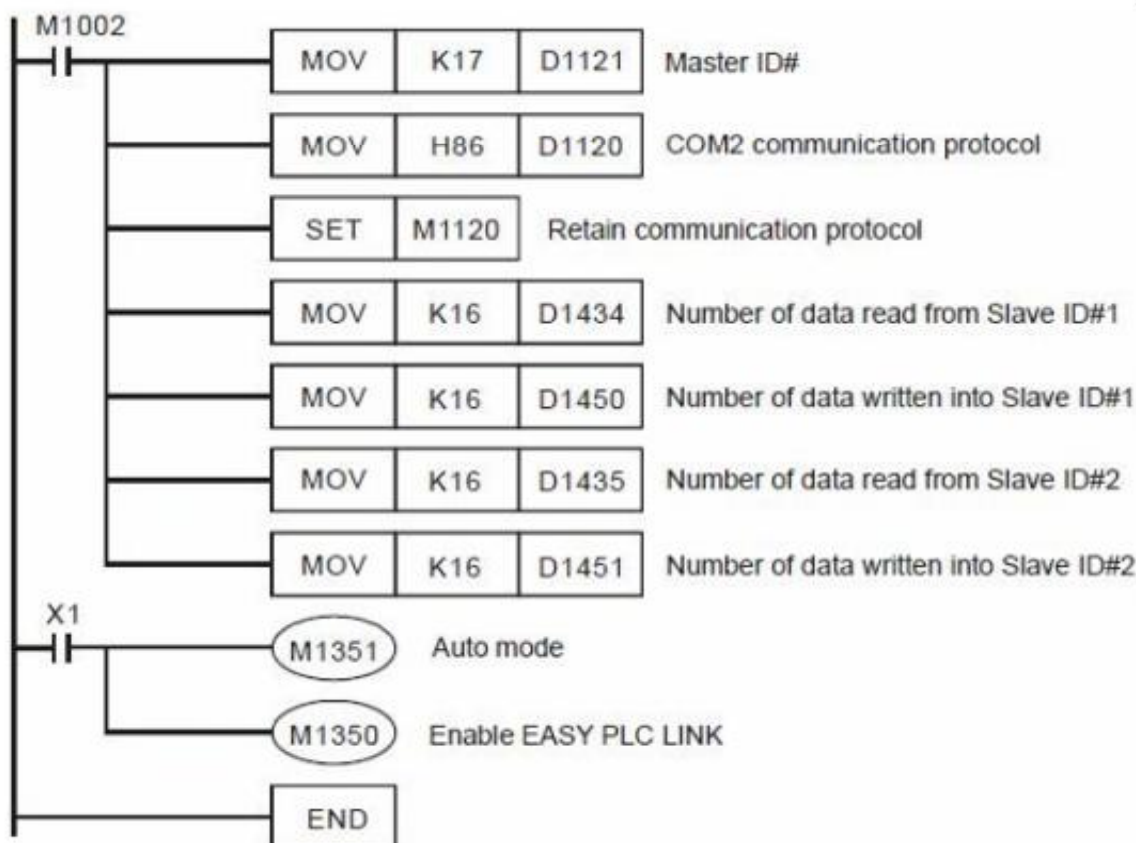
با اجرای وقفه‌ی I170، پردازش داده در شبکه‌ی PLC LINK در خط END اجرا نخواهد شد، اما در هنگام فعال شدن وقفه‌ی I170 فوراً بعد از دریافت داده فرآیند پردازش داده تکمیل خواهد شد. اگر سرعت عکس‌العمل IC (مدار مجتمع) RS-485 جهت کنترل سیگنال Slave کند بود، پیشنهاد می‌شود که برای رفع این مشکل وقفه‌ی I170 را فعال نمایید. می‌توانید رجیستر خاص D1399 (تنظیم شماره Slave ID اختصاص داده شده به PLC LINK) را تنظیم نمایید. در این حالت، ID# پانزده Slave بعدی به ترتیب تنظیم خواهد شد. بعنوان

مثال، وقتی که مقدار رجیستر خاص D1399 را برابر با K20 تنظیم نمایید، Master PLC شماره IDهای 5~20 Slave ID را آشکارسازی خواهد کرد.

3-4-7 اتصال یک Master PLC و دو Slave PLC دلتا بهم توسط شبکه RS-485 و تبادل دیتا توسط پروتکل PLC LINK

در این مثال، یک Master PLC و دو Slave PLC را توسط شبکه RS-485 بهم متصل کرده‌ایم و 16 داده را بین Master و Slave ها را توسط PLC LINK تبادل داده می‌نماییم. در این مثال، فلگ خاص M1353=OFF بوده، تعداد 16 ایستگاه به شبکه قابل اتصال است و مد نوشتن/خواندن 16 داده فعال می‌باشد.

در شکل زیر، برنامه‌ی نوشته شده در داخل Master PLC (ID#17) به زبان دیاگرام نردبانی آورده شده است:



وقتی که ورودی X0=ON شود، تبادل داده بین Master و دو Slave به صورت اتوماتیک در PLC LINK صورت خواهد گرفت. مقادیر موجود در رجیسترهای D100~D115 هر دو Slave توسط Master PLC خوانده شده و در رجیسترهای خاص D1512~D1527،

D1480~D1495 ذخیره خواهد شد. همچنین، مقادیر موجود در رجیسترهای خاص D1496~D1511, D1528~D1495 در داخل رجیسترهای D200~D215 هر دو Slave PLC نوشته خواهد شد.

Master PLC *1		Slave PLC*2
D1480 ~ D1495	Read ←	D100 ~ D115 of Slave ID#1
D1496 ~ D1511	Write →	D200 ~ D215 of Slave ID#1
D1512 ~ D1527	Read ←	D100 ~ D115 of Slave ID#2
D1528 ~ D1543	Write →	D200 ~ D215 of Slave ID#2

فرض کنید از داده‌های موجود در رجیستر D برای تبادل داده بین Master و Slave قبل از فعال شدن PLC LINK (شرط غیرفعال بودن شبکه‌ی PLC LINK غیرفعال بودن فلگ خاص M1350 است) استفاده شده است. در جدول زیر، لیست رجیسترها و مقادیر موجود در آنها در این مثال آورده شده است:

Master PLC	Preset value	Slave PLC	Preset value
D1480 ~ D1495	K0	D100 ~ D115 of Slave ID#1	K5,000
D1496 ~ D1511	K1,000	D200 ~ D215 of Slave ID#1	K0
D1512 ~ D1527	K0	D100 ~ D115 of Slave ID#2	K6,000
D1528 ~ D1543	K2,000	D200 ~ D215 of Slave ID#2	K0

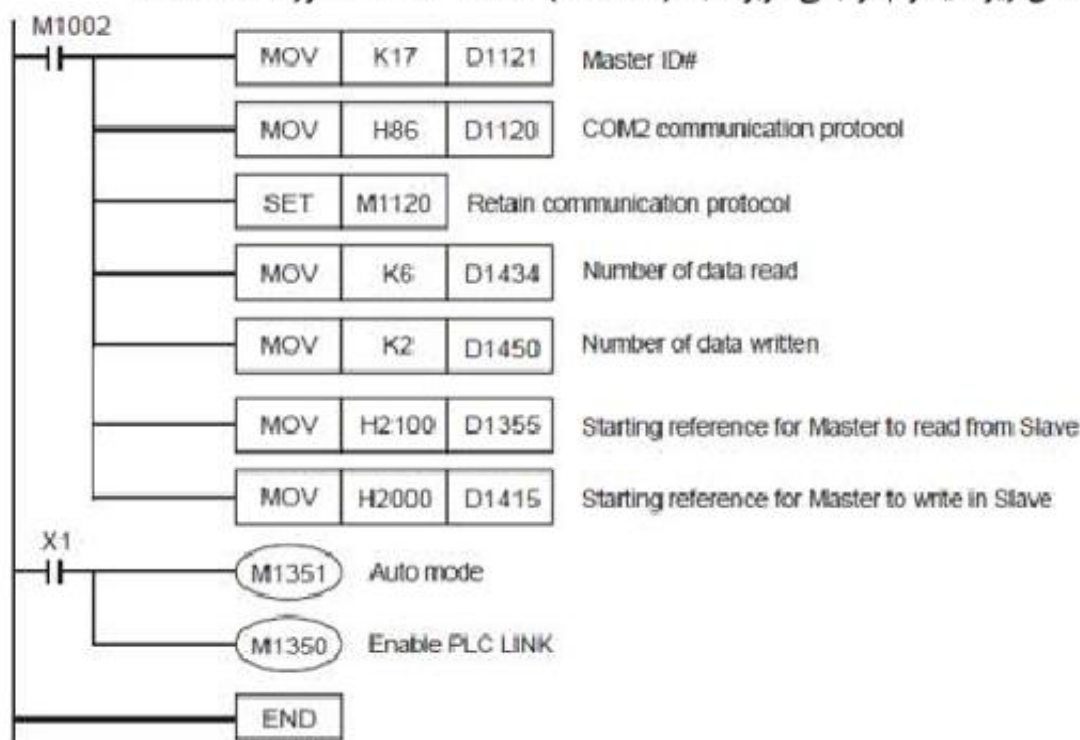
بعد از اینکه شبکه‌ی PLC LINK فعال شد (M1350=ON)، داده‌های موجود در رجیسترهای D برای تبادل داده در این مثال مطابق با جدول زیر خواهد بود:

Master PLC	Preset value	Slave PLC	Preset value
D1480 ~ D1495	K5,000	D100 ~ D115 of Slave ID#1	K5,000
D1496 ~ D1511	K1,000	D200 ~ D215 of Slave ID#1	K1,000
D1512 ~ D1527	K6,000	D100 ~ D115 of Slave ID#2	K6,000
D1528 ~ D1543	K2,000	D200 ~ D215 of Slave ID#2	K2,000

از PLC های SA/ SX/ SC/EH2/ SV/ EH3/ SV2 می‌توان بعنوان Master PLC و از تمام PLC های سری DVP دلتا می‌توان بعنوان Slave PLC استفاده کرد. در اینجا، می‌توان حداکثر از 16 عدد Slave PLC در شبکه PLC LINK استفاده کرد.

4-4-7 تبادل داده در شبکه‌ی PLC LINK بین یک PLC و اینورتر VFD-M دلتا

در این مثال، می‌خواهیم بین PLC دلتا و اینورتر سری VFD-M دلتا شبکه‌ی PLC LINK را برقرار نموده و از طریق این شبکه اینورتر را استوپ، استارت در جهت راستگرد و استارت در جهت چپگرد کرده و همچنین مقدار فرکانس جاری اینورتر را خوانده و یا مقدار فرکانس اینورتر را تنظیم نماییم. در شکل زیر، دیباگرام نردبانی مربوط به Master PLC (ID#17) آورده شده است:



رجیسترهای D1480~D1485 در سمت PLC را به پارامترهای H2100~H2102 اینورتر سری VFD-M اختصاص داده‌ایم. وقتی که ورودی X1=ON شود، PLC LINK فعال شده و دیتای موجود در پارامترهای H2100~H2105 اینورتر در رجیسترهای D1480~D1485 به نمایش درخواهد آمد.

رجیسترهای D1496~D1497 در سمت PLC را به پارامترهای H2001~H2000 اینورتر سری VFD-M دلتا اختصاص داده ایم. وقتی که ورودی X1=ON شود، PLC LINK فعال شده و دیتای موجود در پارامترهای H2001~H2000 اینورتر در رجیسترهای D1496~D1497 به نمایش در خواهد آمد.

از رجیستر D1496 برای دادن فرمان به اینورتر سری VFD-M استفاده شده است. اگر مقدار این رجیستر برابر با $D1496=H12$ شود، اینورتر موتور را در جهت راستگرد به چرخش در خواهد آورد و اگر مقدار این رجیستر برابر با $D1496=H11$ شود، چرخش در جهت چپگرد اینورتر فعال خواهد شد. از رجیستر D1497 برای تغییر فرکانس اینورتر سری VFD-M استفاده شده است. اگر مقدار این رجیستر برابر با K5000 شود، فرکانس خروجی اینورتر برابر با 50HZ خواهد شد.

از PLC های سری SA, SX, SC, EH2, SV, EH3, SV2 می‌توان بعنوان Master PLC در شبکه‌ی PLC LINK و از تمام اینورترهای سری VFD به جز اینورتر VFD-A می‌توان بعنوان Slave در شبکه‌ی PLC LINK استفاده کرد. همچنین، از کنترلرهای دمای سری DTA و سرودرایوهای ASDA دلتا می‌توان بعنوان Slave در شبکه‌ی PLC LINK استفاده کرد. دقت داشته باشید که هر وسیله‌ی که با پروتکل Modbus سازگار باشد، می‌توان از آن در شبکه PLC LINK استفاده کرد. حداکثر وسیله‌ای که می‌توان به شبکه‌ی PLC LINK متصل (LINK) کرد 16 وسیله‌ی تحت پروتکل Modbus است.

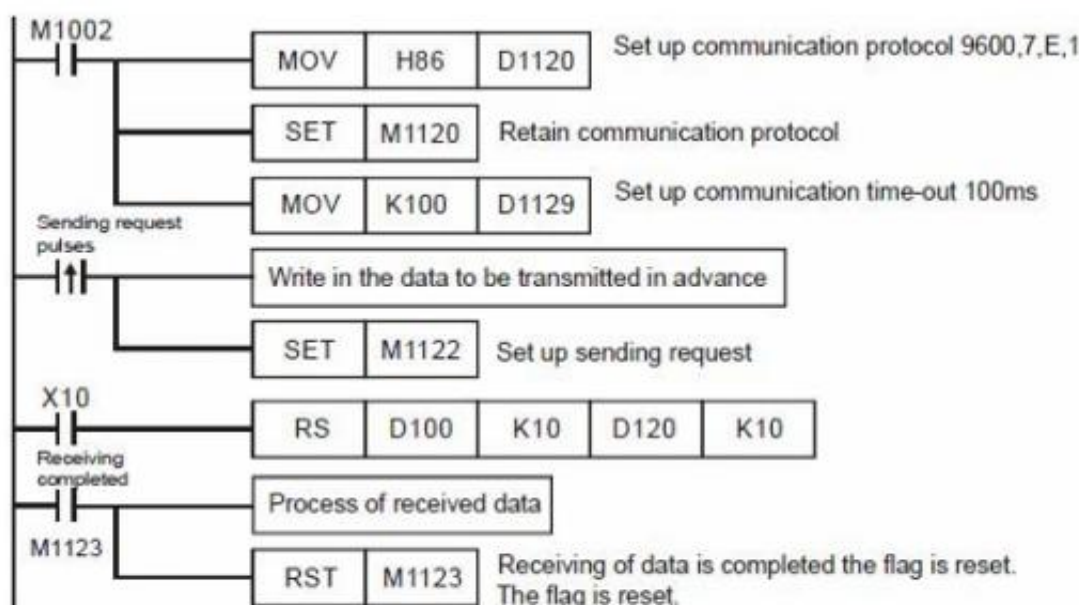
جهت اطلاع از شروع شماره ID برای Slave ها و خواندن و نوشتن مقادیر در Slave ها توسط Master PLC باید به جدول توضیحات فلگ‌های خاص M و رجیسترهای خاص D مراجعه نمایید.

7- 5- تبادل دیتا در شبکه RS-485 با پروتکل Modbus غیراستاندارد

7- 5- 1 تبادل دیتا در شبکه‌ی RS-485 توسط دستورالعمل RS

در این مثال، می‌خواهیم جهت ارسال دیتا، داده را در داخل رجیستر شروع پارامتر مبدأ (D100) بنویسیم.

برای ارسال داده‌ی نوشته شده در پارامتر RS ابتدا باید فلگ درخواست ارسال داده در شبکه‌ی (M1122) را فعال (ON) کنیم.



عملکرد برنامه به شرح زیر است:

وقتی که ورودی $X0=ON$ شود، دستورالعمل RS اجرا شده و PLC شروع به انتظار برای ارسال و دریافت داده خواهد کرد. آدرس شروع داده برای ارسال D100 بوده و طول داده‌ی قابل ارسال برابر با 10 داده می‌باشد. وقتی که عمل ارسال داده توسط دستورالعمل RS پایان یافت، فلگ M1122 به صورت اتوماتیک ریست (OFF) خواهد شد. (به همین دلیل، برای ریست کردن فلگ M1122 از دستورالعمل RST M1122 استفاده شده است). بعد از گذشت 1 میلی‌ثانیه PLC شروع به دریافت 10 عدد داده می‌کند. ذخیره داده در رجیسترهای متوالی به طول 10 رجیستر صورت خواهد گرفت. از آنجایی که آدرس شروع ابزار مقصد D120 می‌باشد، داده‌های دریافتی در رجیسترهای D120~D130 ذخیره خواهند شد.

وقتی که عمل دریافت داده تکمیل شد، فلگ M1123 به صورت اتوماتیک فعال (ON) می‌شود. این فلگ بیانگر تکمیل فرآیند دریافت دیتا در شبکه توسط دستورالعمل RS است. بعد از پایان یافتن فرآیند دریافت دیتا توسط برنامه، فلگ M1123 باید ریست (OFF) شود تا PLC مجدداً آماده‌ی شروع ارسال و یا دریافت دیتا توسط دستورالعمل RS شود. برای ریست کردن فلگ M1123 از دستورالعمل RST M1123 استفاده شده است.

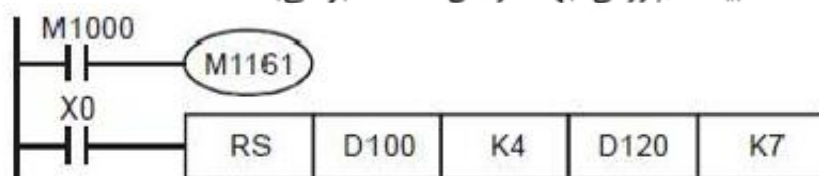
دقت کنید که به دلیل غیرفعال شدن فلگ M1123 به صورت اتوماتیک در هنگام شروع ارسال داده توسط دستورالعمل RS و همچنین، به دلیل فعال شدن فلگ M1123 در هنگام پایان یافتن عمل ارسال داده توسط دستورالعمل RS از هیچ دستورالعملی برای ریست کردن فلگ M1122 و فعال کردن فلگ M1123 در برنامه‌نویسی استفاده نمی‌شود.

7-5-2 سوئیچ بین مد 8 بیتی و مد 16 بیتی دستورالعمل RS

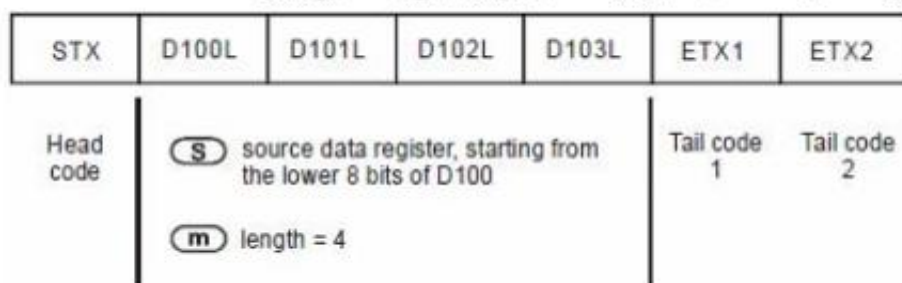
در این مثال، می‌خواهیم بین مد 8 بیتی و مد 16 بیتی سوئیچ کنیم.
برای فعال کردن مد 8 بیتی باید فلگ M1161=ON شود و برای فعال کردن مد 16 بیتی باید فلگ M1161=OFF شود.

روش فعال کردن مد 8 بیتی به شرح زیر است:

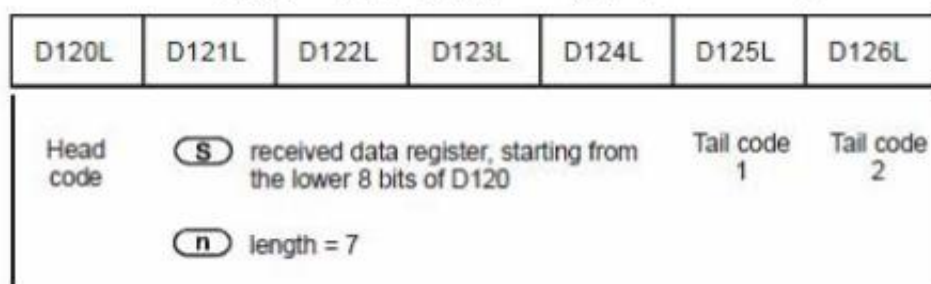
کد شروع (Head Code) و کد پایان (Tail Code) داده توسط ابزارهای خاص M1126 و M1130 به همراه رجیسترهای خاص D1124~D1126 تنظیم می‌شوند. دقت کنید که رله‌های کمکی و رجیسترهای داده‌ی گفته شده همگی خاص بوده و از قبل توسط شرکت دلتا برای کاربرد مقاصد ویژه از پیش برنامه‌ریزی شده‌اند. وقتی که PLC در حال اجرای دستورالعمل RS باشد، کد شروع و کد پایان تنظیم شده توسط کاربر به صورت اتوماتیک به خارج ارسال خواهند شد.
M1161=ON بیانگر این است که PLC در مد تبدیل 8 بیتی می‌باشد. این به آن معنی است که داده‌های 16 بیتی به دو داده‌ی 8 بیتی باارزش و کم‌ارزش تقسیم می‌شوند. در این مد از 8 بیت باارزش صرف‌نظر شده و فقط 8 بیت کم‌ارزش جهت ارسال داده معتبر می‌باشند.



الگوی ارسال داده از PLC به تجهیزات خارجی (جانبی) به شرح زیر است:



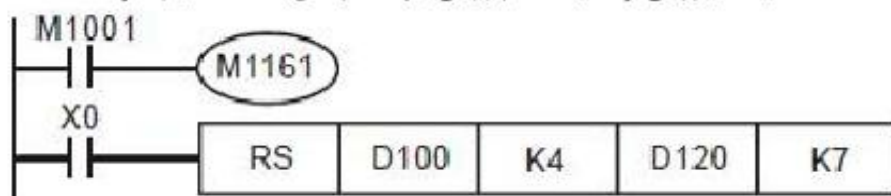
الگوی دریافت داده توسط PLC از تجهیزات خارجی (جانبی) به شرح زیر است:



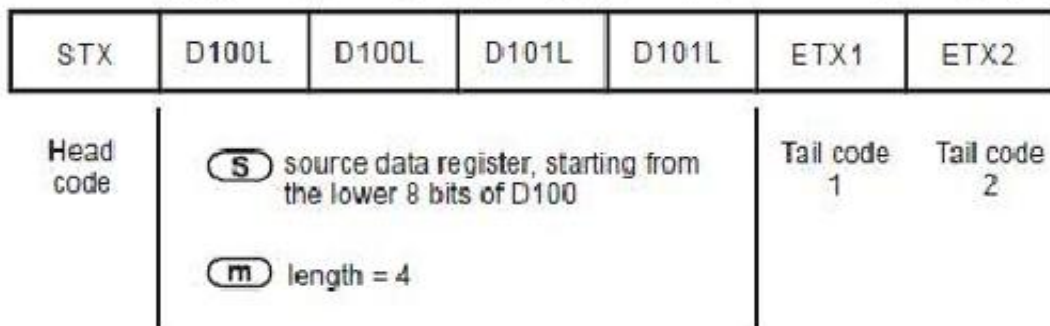
وقتی که داده توسط PLC در حال دریافت است، PLC کد شروع و کد پایان الگوی دریافت داده را از تجهیزات خارجی دریافت خواهد کرد. از این رو، کاربر باید از تنظیمات طول داده‌ی n مطلع شود. روش فعال کردن مد 16 بیتی به شرح زیر می‌باشد:

کد شروع (Head Code) و کد پایان (Tail Code) داده توسط ابزارهای خاص M1126 و M1130 به همراه رجیسترهای خاص D1124~D1126 تنظیم می‌شوند. وقتی که PLC در حال اجرای دستورالعمل RS می‌باشد، کد شروع و کد پایان تنظیم شده توسط کاربر به صورت اتوماتیک به خارج ارسال خواهند شد.

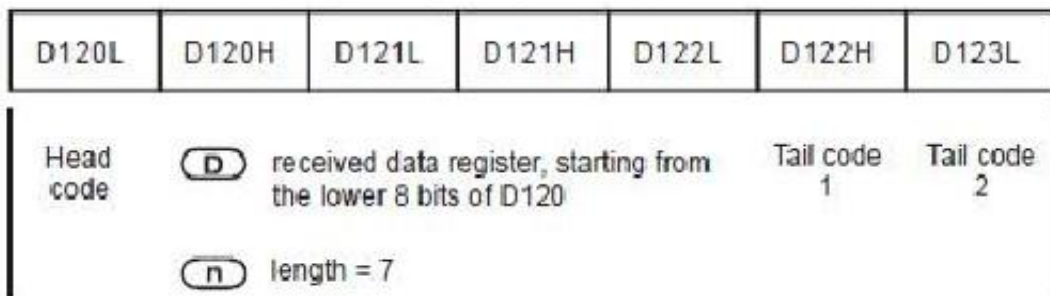
M1161=OFF بیانگر این است که PLC در مد تبدیل 16 بیتی می‌باشد. در این حالت، داده‌ی 16 بیتی به دو قسمت 8 بیت با ارزش و 8 بیت کم ارزش جهت ارسال داده تقسیم خواهد شد.



الگوی ارسال داده از PLC به تجهیزات خارجی (جانبی) مطابق با الگوی شکل زیر است:



الگوی دریافت داده توسط PLC از تجهیزات خارجی (جانبی) به شرح زیر است:

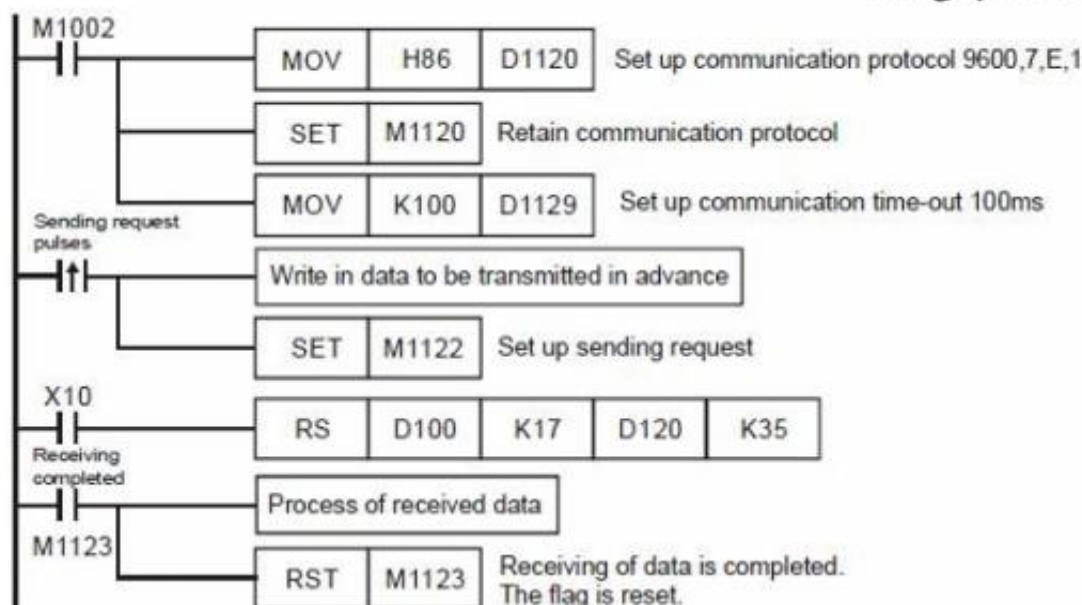


وقتی که داده توسط PLC در حال دریافت است، PLC کد شروع و کد پایان الگوی دریافت داده را از تجهیزات جانبی (خارجی) دریافت خواهد کرد. از این رو، کاربر باید از تنظیمات طول داده مطلع شود.

3-5-7 تبادل دیتا بین PLC و اینورتر VFD-B دلتا توسط دستورالعمل RS در مد ASCII

در این مثال، می‌خواهیم اینورتر AC سری VFD-B دلتا را به PLC دلتا متصل نموده و بین این دو از طریق دستورالعمل RS ارتباط برقرار کنیم. مد تبادل داده در سمت اینورتر برابر با مد اسکی ASCII تنظیم شده است.

در این مثال، در سمت PLC باید از مد 16 بیتی برای تبادل داده استفاده شود. از این رو، باید M1161=OFF باشد. در این مثال طول داده 6 است، بنابراین باید در اینورتر سری VFD-B داده‌ی 16 بیتی را نوشت. آدرس پارامتر شروع نوشتن داده داخل اینورتر توسط شبکه RS-485 برابر با H2101 هگز می‌باشد.



PLC → VFD-B: در این مثال، مقدار PLC "01 03 2101 0006 D4 CR LF" را به اینورتر ارسال می‌کند.

VFD-B → PLC: در این مثال، اینورتر مقدار "01 03 0C 0100 1766 0000 0000" را به PLC ارسال می‌کند.

در جدول زیر، رجیسترهای ارسال داده آورده شده است. این به آن معنی است که PLC پیغامی را به خارج (اینورتر) ارسال خواهد کرد.

Register	Data		Explanation	
D100 low	':'	3A H	STX	
D100 high	'0'	30 H	ADR 1	Address of AC motor drive: ADR (1,0)
D101 low	'1'	31 H	ADR 0	
D101 high	'0'	30 H	CMD 1	Instruction code: CMD (1,0)
D102 low	'3'	33 H	CMD 0	
D102 high	'2'	32 H	Start data address	
D103 low	'1'	31 H		
D103 high	'0'	30 H		
D104 low	'1'	31 H	Number of data (counted by Words)	
D104 high	'0'	30 H		
D105 low	'0'	30 H		
D105 high	'0'	30 H		
D106 low	'6'	36 H		
D106 high	'D'	44 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D107 low	'4'	34 H	LRC CHK 0	
D107 high	CR	D H	END	
D108 low	LF	A H		

در جدول زیر، رجیسترهای دریافت داده آورده شده است. این به آن معنی است که اینورتر VFD-B به پیغام ارسال شده توسط PLC پاسخ داده است.

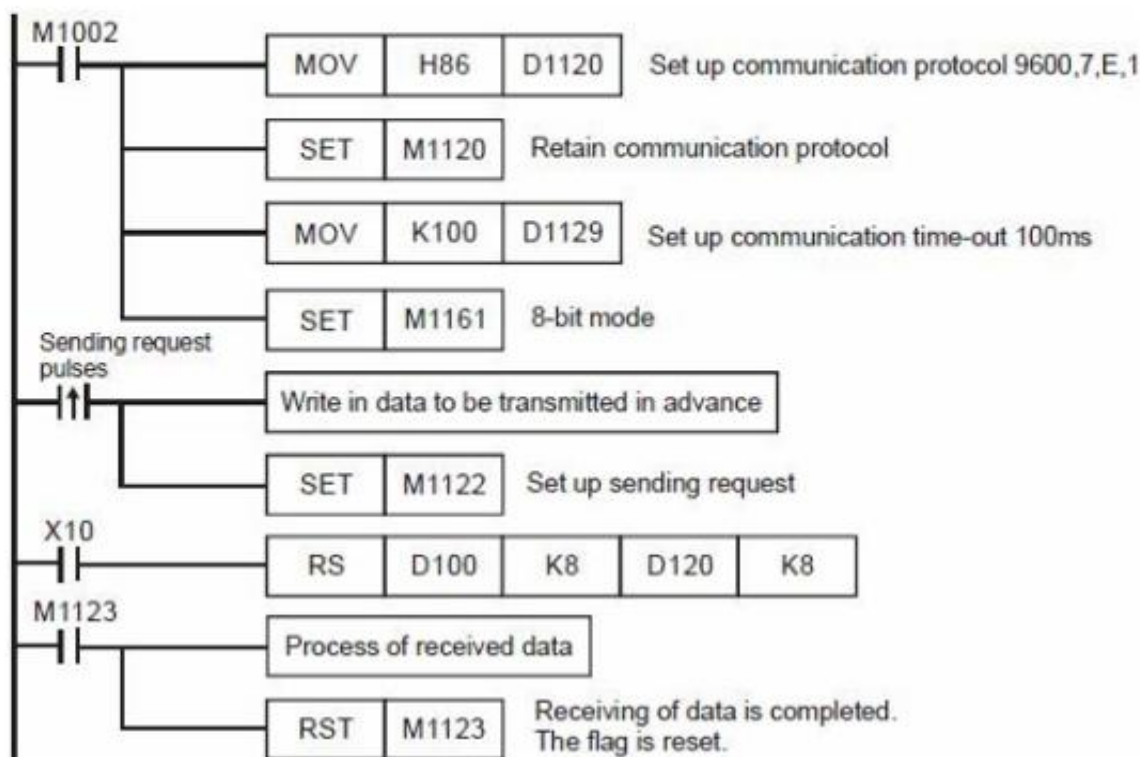
Register	Data		Explanation	
D120 low	':'	3A H	STX	
D120 high	'0'	30 H	ADR 1	
D121 low	'1'	31 H	ADR 0	
D121 high	'0'	30 H	CMD 1	
D122 low	'3'	33 H	CMD 0	
D122 high	'0'	30 H	Number of data (counted by byte)	
D123 low	'C'	43 H		
D123 high	'0'	30 H	Content of address 2101 H	
D124 low	'1'	31 H		
D124 high	'0'	30 H		
D125 low	'0'	30 H	Content of address 2102 H	
D125 high	'1'	31 H		
D126 low	'7'	37 H		
D126 high	'6'	36 H		
D127 low	'6'	36 H	Content of address 2103 H	
D127 high	'0'	30 H		
D128 low	'0'	30 H		
D128 high	'0'	30 H		

D129 low	'0'	30 H	Content of address 2104 H
D129 high	'0'	30 H	
D130 low	'0'	30 H	
D130 high	'0'	30 H	
D131 low	'0'	30 H	Content of address 2105 H
D131 high	'0'	30 H	
D132 low	'1'	31 H	
D132 high	'3'	33 H	
D133 low	'6'	36 H	Content of address 2106 H
D133 high	'0'	30 H	
D134 low	'0'	30 H	
D134 high	'0'	30 H	
D135 low	'0'	30 H	LRC CHK 1
D135 high	'3'	33 H	
D136 low	'B'	42 H	LRC CHK 0
D136 high	CR	D H	
D137 low	LF	A H	END

7-5-4 تبادل دیتا بین PLC و اینورتر VFD-B دلتا توسط دستورالعمل RS در مد RTU

در این مثال، می‌خواهیم اینورتر AC سری VFD-B دلتا را به PLC دلتا متصل نموده و بین این دو از طریق دستورالعمل RS ارتباط برقرار کنیم. مد تبادل داده در سمت اینورتر برابر با مد RTU تنظیم شده است.

در این مثال، در سمت PLC باید از مد 16 بیتی تبادل داده استفاده کنیم. از این رو، باید M1161=ON باشد. می‌خواهیم از طریق پورت ارتباطات شبکه‌ی RS-485 مقدار H12 را توسط PLC در داخل آدرس پارامتر H2000 (فرمت آدرس هگز می‌باشد) اینورتر VFD-B بنویسیم.



ارسال می‌کند. PLC → VFD-B: در این مثال، PLC مقدار 01 06 2000 0012 02 07 را به اینورتر

ارسال می‌کند. VFD-B → PLC: در این مثال، اینورتر مقدار 01 06 2000 0012 02 07 را به PLC

در جدول زیر رجیسترهای ارسال داده آورده شده است. این به آن معنی است که PLC پیغامی را به خارج (اینورتر) ارسال خواهد کرد.

Register	Data	Explanation
D100 low	01 H	Address
D101 low	06 H	Function
D102 low	20 H	Data address
D103 low	00 H	
D104 low	00 H	Data content
D105 low	12 H	
D106 low	02 H	CRC CHK Low
D107 low	07 H	CRC CHK High

در جدول زیر، رجیسترهای دریافت داده آورده شده است. این به آن معنی است که اینورتر VFD-B به پیغام ارسال شده توسط PLC پاسخ داده است.

Register	Data	Explanation
D120 low	01 H	Address

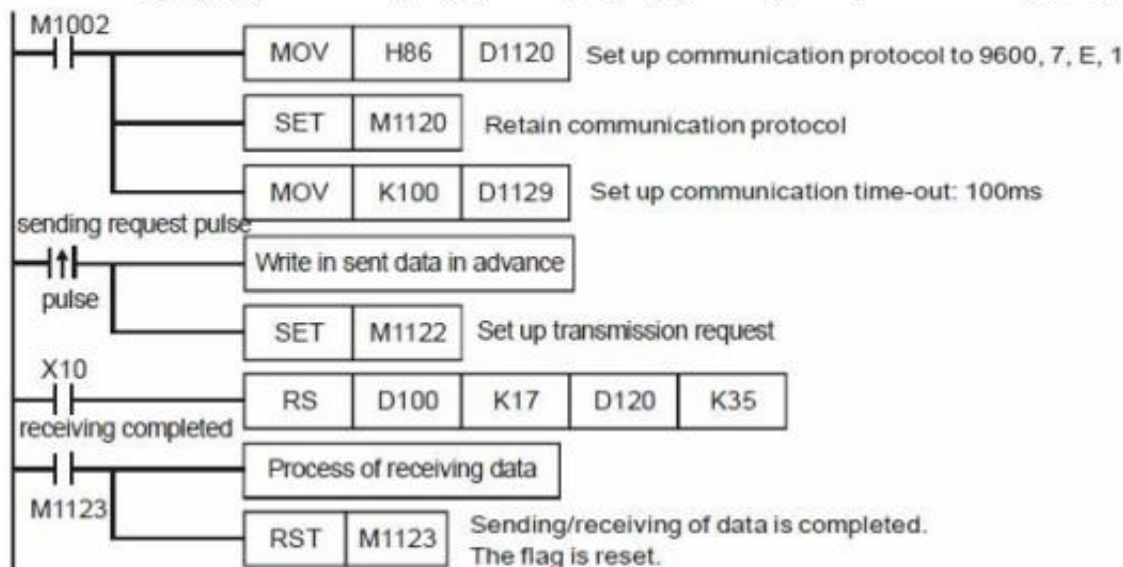
D121 low	06 H	Function
D122 low	20 H	Data address
D123 low	00 H	
D124 low	00 H	
D125 low	12 H	Data content
D126 low	02 H	CRC CHK Low
D127 low	07 H	CRC CHK High

توجه :

1. ارتباطات شبکه‌ی RS-485 از دستورالعمل‌های RS, MODRD, MODWR, FWD, REV, STOP, RDST, RSTEF, MODRW پشتیبانی می‌کند. از این رو، چندین فلگ برای کار کردن این دستورات برنامه‌ریزی شده‌اند. در زیر، لیستی از فلگ‌ها (رله‌های کمکی) خاص رزرو شده برای ارتباطات RS-485 آورده شده است.

5-5-7 ارتباط شبکه بین PLC و اینورتر VFD-S دلتا با مد ASCII

می‌خواهیم بین PLC دلتا و اینورتر سری VFD-S دلتا ارتباط شبکه با مد ASCII برقرار نماییم. برای فعال کردن مد اسکی باید فلگ M1143 غیرفعال (OFF) باشد. برای عمل تبدیل مقدار اسکی، از مد تبدیل 8 بیتی استفاده می‌کنیم. برای فعال کردن مد تبدیل 8 بیتی مقدار اسکی، باید فلگ M1161 فعال (ON) شود. داده‌های ارسالی باید در پارامترهای اینورتر نوشته شوند و 6 داده باید از اینورتر سری VFD-S خوانده شود که شروع خواندن داده از پارامتر H2101 اینورتر می‌باشد.



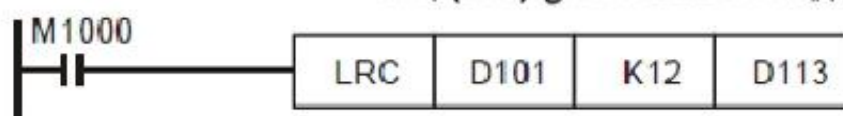
PLC → VFD-S: ارسال مقدار "01 03 2101 0006 D4 CR LF" از PLC دلتا به

اینورتر سری VFD-S دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation	
D100 low	'.'	3A H	STX	
D101 low	'0'	30 H	ADR 1 ADR 0	Address of AC motor drive: ADR (1,0)
D102 low	'1'	31 H		
D103 low	'0'	30 H	CMD 1 CMD 0	Instruction code: CMD (1,0)
D104 low	'3'	33 H		
D105 low	'2'	32 H	Starting data address	
D106 low	'1'	31 H		
D107 high	'0'	30 H		
D108 low	'1'	31 H		
D109 low	'0'	30 H		
D110 low	'0'	30 H	Number of data (counted by Words)	
D111 low	'0'	30 H		
D112 low	'6'	36 H		
D113 low	'D'	44 H	LRC CHK 1 LRC CHK 0	Error checksum: LRC CHK (0,1)
D114 low	'4'	34 H		
D115 low	CR	A H	END	
D116 low	LF	D H		

خطای Checksum LRC CHK(0,1) می‌تواند توسط دستورالعمل LCR محاسبه شود. در مد تبدیل 8 بیتی باید فلگ M1161 فعال (ON) باشد.



$01H + 03H + 21H + 01H + 00H + 06H = 2CH$ LRC Checksum →
متمم 2 بدست آمده برابر با D4H است و مقدار (44H) 'D' در 8 بیت کم‌ارزش رجیستر D113 ذخیره شده و مقدار (34H) '4' در 8 بیت کم‌ارزش رجیستر D114 ذخیره می‌شود.

نکته:

فرمت مد اسکی با اطلاعات ارتباطات شبکه مطابق جدول زیر می‌باشد:

STX	'.'	Start Word = '.' (3AH)
Address Hi	'0'	Communication: 8-bit address consists of 2 ASCII codes
Address Lo	'1'	
Function Hi	'0'	Function code: 8-bit function consists of 2 ASCII codes
Function Lo	'3'	

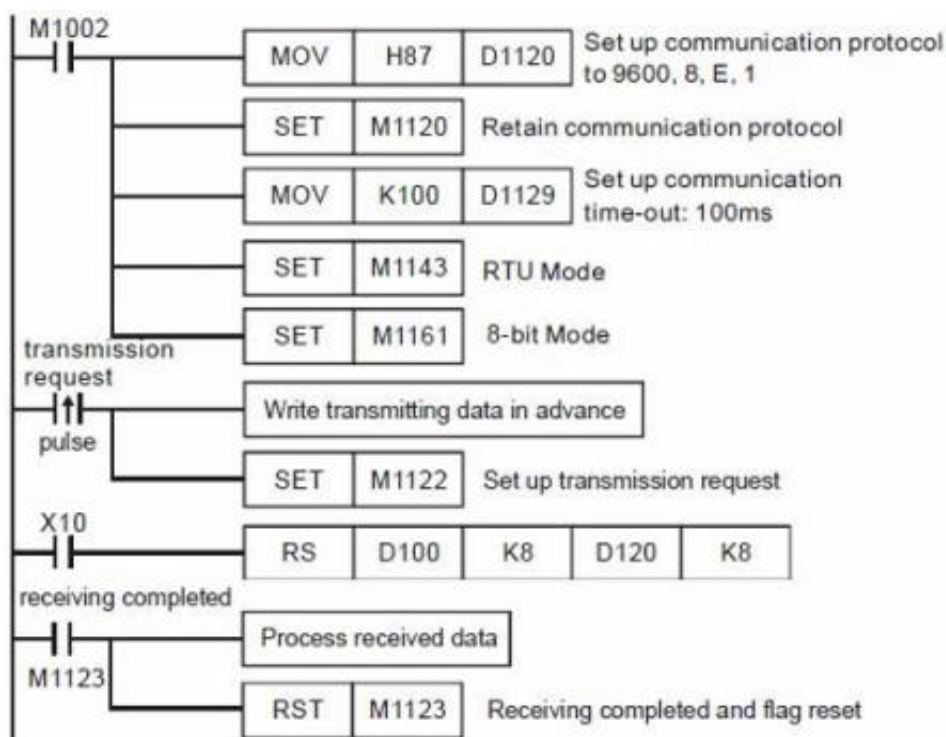
DATA (n-1) DATA 0	'2'	Data content: $n \times 8\text{-bit data consists of } 2n \text{ ASCII codes}$
	'1'	
	'0'	
	'2'	
	'0'	
	'0'	
	'0'	
	'2'	
LRC CHK Hi	'D'	LRC checksum: 8-bit checksum consists of 2 ASCII codes
LRC CHK Lo	'7'	
END Hi	CR	End Word: END Hi = CR (0DH), END Lo = LF(0AH)
END Lo	LF	

LRC Checksum متمم 2 مجموعه مقادیر جمع شده‌ی آدرس و داده‌ی ارتباطات شبکه می‌باشد. برای مثال، فرض کنید که مجموع مقدار یک بسته‌ی اطلاعاتی که شامل آدرس و داده می‌باشد، برابر با $01H + 03H + 21H + 02H + 00H + 02H = 29H$ است. دومین متمم بدست آمده برای مقدار $29H$ برابر $D7H$ می‌باشد.

7-5-6 تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل RS در مد

RTU و بررسی درستی دیتا توسط دستورالعمل CRC

در این مثال، می‌خواهیم بین PLC دلتا و اینورتر سری VFD-S دلتا ارتباط برقرار کنیم. مد ارتباطات در این مثال مد RTU می‌باشد. برای راه‌اندازی مد RTU باید فلگ خاص $M1143=ON$ شود. همچنین، مد تبدیل در این مثال مد تبدیل 16 بیتی است. برای فعال کردن مد تبدیل 16 بیتی باید فلگ خاص $M1161=ON$ شود. در این مثال، می‌خواهیم مقدار هگز $H12$ را داخل پارامتر $H2000$ اینورتر VFD-S دلتا ارسال کرده و بنویسیم.

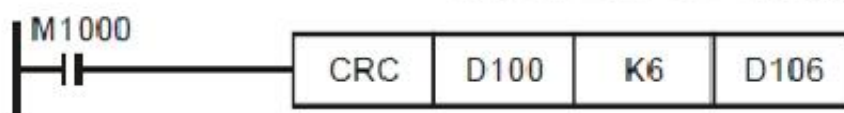


PLC → VFD-S: ارسال مقدار "01 06 20000 0012 02 07" از PLC دلتا به اینورتر سری VFD-S دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیام) مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D100 low	01 H	Address
D101 low	06 H	Function
D102 low	20 H	Data address
D103 low	00 H	
D104 low	00 H	
D105 low	12 H	Data content
D106 low	02 H	
D107 low	07 H	CRC CHK 1

خطای (0, 1) CRC Checksum می‌تواند توسط دستورالعمل CRC محاسبه شود. در مد تبدیل 8 بیتی باید فلگ M1161=ON باشد.



مقدار CRC Checksum برابر با 02H و 07H است که مقدار 02H در 8 بیت کم‌ارزش رجیستر D106 و مقدار 07 در 8 بیت کم‌ارزش رجیستر D107 ذخیره می‌شود.

نکته: فرمت مد RTU با اطلاعات ارتباطات شبکه مطابق با جدول زیر است:

START	Time interval
Address	Communication address: 8-bit binary
Function	Function code: 8-bit binary
DATA (n-1)	Data content: $n \times 8\text{-bit data}$
.....	
DATA 0	
CRC CHK Low	CRC checksum: 16-bit CRC checksum consists of 2 8-bit binaries
CRC CHK High	
END	Time interval

CRC Checksum از Address شروع شده و به END ختم می‌شود. محتویات داده بین Address قرار می‌گیرند.

عملکرد CRC Checksum در 6 مرحله انجام می‌شود که عبارتند از:

مرحله‌ی 1 ← ابتدا یک رجیستر 16 بیتی با مقدار FFFFH هگز به نام CRC register ایجاد می‌شود.

مرحله‌ی 2 ← نخستین 8 بیت از بایت دستورالعمل پیغام و بیت کم‌ارزش رجیستر 16 بیتی CRC با یکدیگر XOR می‌شوند. نتیجه‌ی بدست آمده در رجیستر CRC ذخیره می‌شود.

مرحله‌ی 3 ← در رجیستر CRC یک بیت به سمت راست شیفت می‌یابد و مقدار 0 به بیت باارزش رجیستر CRC سرریز می‌شود.

مرحله‌ی 4 ← مقدار شیفت شده به راست چک می‌شود. اگر این مقدار 0 باشد، در مرحله‌ی 3 در رجیستر CRC یک مقدار جدید ذخیره می‌شود. در غیر این صورت، اگر این مقدار 1 باشد، مقدار A001H و مقدار رجیستر CRC با یکدیگر XOR شده و سپس نتیجه‌ی بدست آمده در رجیستر CRC ذخیره می‌شود.

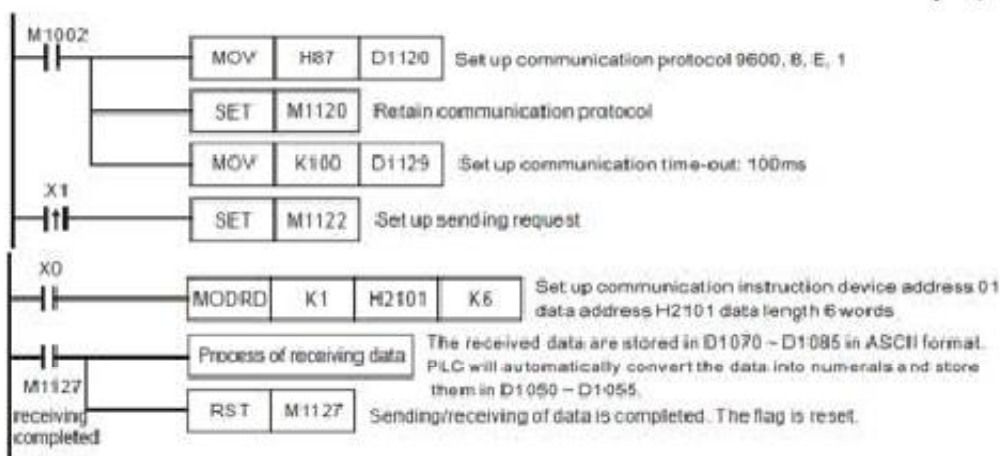
مرحله‌ی 5 ← در این مرحله، مراحل 3 و 4 تکرار می‌شوند تا اینکه محاسبه‌ی هر 8 بیت به پایان برسد.

مرحله‌ی 6 ← در این مرحله، مراحل 2 الی 5 برای بدست آوردن 8 بیت بعدی دستورالعمل پیغام تکرار می‌شوند و تا وقتی که تمام دستورالعمل‌های پیغام محاسبه شوند، مراحل 2 الی 5 تکرار خواهند شد. در پایان، مقدار رجیستر CRC بدست آمده شده همان CRC Checksum می‌باشد. در آخر باید بگوییم که این مقدار CRC Checksum باید در Checksum دستورالعمل پیغام قرار گیرد (جایگزین شود).

7-6 تبادل دیتا در شبکه‌ی RS-485 با پروتکل Modbus استاندارد

7-6-1 تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل MODBUS در مد ASCII

می‌خواهیم بین PLC و اینورتر سری VFD-S دلتا ارتباط برقرار کنیم. در این مثال، مد ارتباط شبکه‌ی MODBUS برابر با ASCII می‌باشد. برای فعال کردن مد اسکی باید فلگ M1143 را غیرفعال (OFF) کرد.



PLC → VFD-S: ارسال مقدار "01 03 2101 0006 D4" از PLC به اینورتر VFD-S.

VFD-S → PLC: ارسال مقدار "01 03 0C 0100 1766 0000 0000 0136 0000"

"3B" از اینورتر VFD-S به PLC.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation	
D1089 low	'0'	30 H	ADR 1	Address of AC motor drive: ADR (1,0)
D1089 high	'1'	31 H	ADR 0	
D1090 low	'0'	30 H	CMD 1	Instruction code: CMD (1,0)
D1090 high	'3'	33 H	CMD 0	
D1091 low	'2'	32 H	Starting data address	
D1091 high	'1'	31 H		
D1092 low	'0'	30 H		
D1092 high	'1'	31 H		
D1093 low	'0'	30 H	Number of data (counted by Words)	
D1093 high	'0'	30 H		
D1094 low	'0'	30 H		
D1094 high	'6'	36 H		
D1095 low	'D'	44 H	LRC CHK 1	Checksum: LRC CHK (0,1)
D1095 high	'4'	34 H	LRC CHK 0	

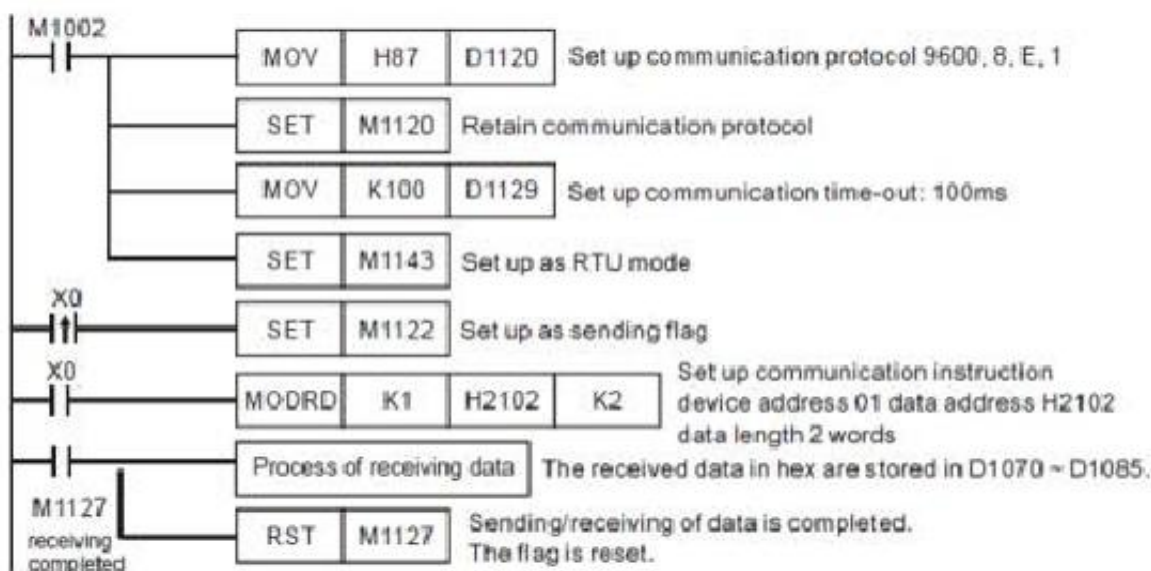
وضعیت رجیسترهای دریافت داده (پاسخ پیغام‌ها) مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation	
D1070 low	'0'	30 H	ADR 1	
D1070 high	'1'	31 H	ADR 0	
D1071 low	'0'	30 H	CMD 1	
D1071 high	'3'	33 H	CMD 0	
D1072 low	'0'	30 H	Number of data (counted by byte)	
D1072 high	'C'	43 H		
D1073 low	'0'	30 H	Content of address 2101 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1050 = 0100 H
D1073 high	'1'	31 H		
D1074 low	'0'	30 H		
D1074 high	'0'	30 H		
D1075 low	'1'	31 H	Content of address 2102 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1051 = 1766 H
D1075 high	'7'	37 H		
D1076 low	'6'	36 H		
D1076 high	'6'	36 H		
D1077 low	'0'	30 H	Content of address 2103 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1052 = 0000 H
D1077 high	'0'	30 H		
D1078 low	'0'	30 H		
D1078 high	'0'	30 H		
D1079 low	'0'	30 H	Content of address 2104 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1053 = 0000 H
D1079 high	'0'	30 H		
D1080 low	'0'	30 H		
D1080 high	'0'	30 H		
D1081 low	'0'	30 H	Content of address 2105 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1054 = 0136 H
D1081 high	'1'	31 H		
D1082 low	'3'	33 H		
D1082 high	'6'	36 H		
D1083 low	'0'	30 H	Content of address 2106 H	PLC automatically convert ASCII codes to numerals and store the numeral in D1055 = 0000 H
D1083 high	'0'	30 H		
D1084 low	'0'	30 H		
D1084 high	'0'	30 H		
D1085 low	'3'	33 H	LRC CHK 1	
D1085 high	'B'	42 H	LRC CHK 0	

7-6-2 تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل

MODBUS در مد RTU

می‌خواهیم بین PLC و اینورتر VFD-S دلتا ارتباط برقرار کنیم. در این مثال، مد ارتباطات شبکه MODBUS برابر با RTU انتخاب خواهد شد. برای فعال کردن مد RTU باید فلگ M1143 را فعال (ON) کرد.



PLC → VFD-S: ارسال مقدار "01 03 2102 0002 6F F7" از PLC به اینورتر VFD-S دلتا.

VFD-S → PLC: ارسال مقدار "01 03 04 1770 0000 FE 5C" از اینورتر VFD-S به PLC دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیام) مطابق با جدول زیر است:

Register	DATA	Explanation
D1089 low	01 H	Address
D1090 low	03 H	Function
D1091 low	21 H	Starting data address
D1092 low	02 H	
D1093 low	00 H	Number of data (counted by Words)
D1094 low	02 H	
D1095 low	6F H	CRC CHK Low
D1096 low	F7 H	CRC CHK High

وضعیت رجیسترهای دریافت داده (پاسخ پیام) مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D1070 low	01 H	Address
D1071 low	03 H	Function
D1072 low	04 H	Number of data (counted by bytes)
D1073 low	17 H	Content of address 2102 H
D1074 low	70 H	
D1075 low	00 H	Content of address 2103 H
D1076 low	00 H	
D1077 low	FE H	CRC CHK Low
D1078 low	5C H	CRC CHK High

7-6-3 دومین مثال از تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط

دستورالعمل MODBUS در مد ASCII

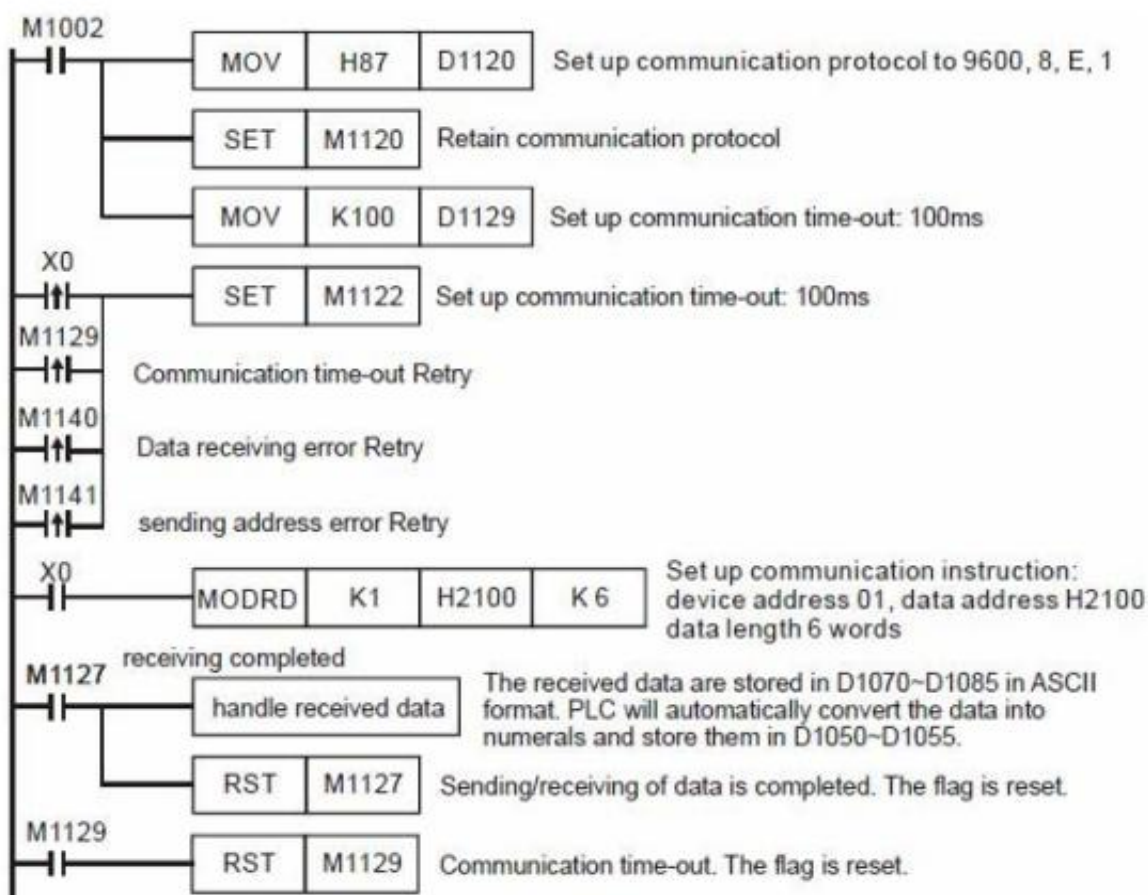
می‌خواهیم بین PLC و اینورتر سری VFD-S دلتا ارتباط شبکه برقرار کنیم. مد ارتباطات شبکه‌ی MODBUS برابر با ASCII می‌باشد. برای فعال کردن مد اسکی فلگ M1143 باید غیرفعال (OFF) باشد. وقتی که خطای time out ارتباطات، خطای دریافت داده و یا خطای ارسال داده رخ دهد، سیستم سعی می‌کند که مجدداً ارتباط شبکه را برقرار کند.

وقتی که X0=ON شود، PLC شروع به خواندن داده از پارامتر H2100 اینورتر VFD-S با آدرس ارتباطات شبکه‌ی 01 کرده و مقدار خوانده شده را به فرمت ASCII در رجیسترهای خاص D1070~D1085 ذخیره می‌کند. سپس PLC به صورت اتوماتیک داده به فرمت اسکی را به عدد تبدیل کرده و در رجیسترهای D1050~D1055 ذخیره می‌کند.

وقتی که خطای time out ارتباطات رخ دهد، فلگ M1129 فعال (ON) خواهد شد. برنامه فلگ M1129 را راه‌اندازی نموده و درخواست خواندن مجدد داده به فلگ M1122 ارسال می‌شود.

وقتی که خطای دریافت داده رخ دهد، فلگ M1140 فعال (ON) خواهد شد. برنامه فلگ M1140 را راه‌اندازی می‌کند و درخواست خواندن مجدد داده به فلگ M1122 ارسال می‌شود.

وقتی که خطای ارسال داده به آدرس مورد نظر اینورتر رخ دهد، فلگ M1141 فعال (ON) خواهد شد. برنامه فلگ M1141 را راه‌اندازی می‌کند و درخواست خواندن مجدد داده به فلگ M1122 ارسال می‌شود.



توجه داشته باشید که :

سه دستورالعمل **MODRW**, **RDST**, **MODRD** را نمی‌توان با کنتاکت‌های با لبه‌ی بالارونده (LDP, ANDP, ORP) و لبه‌ی پایین‌رونده (LDF, ANDF, ORF) راه‌اندازی کرد. در غیر اینصورت، داده‌های ذخیره شده در رجیسترهای دریافت داده صحیح نمی‌باشند.

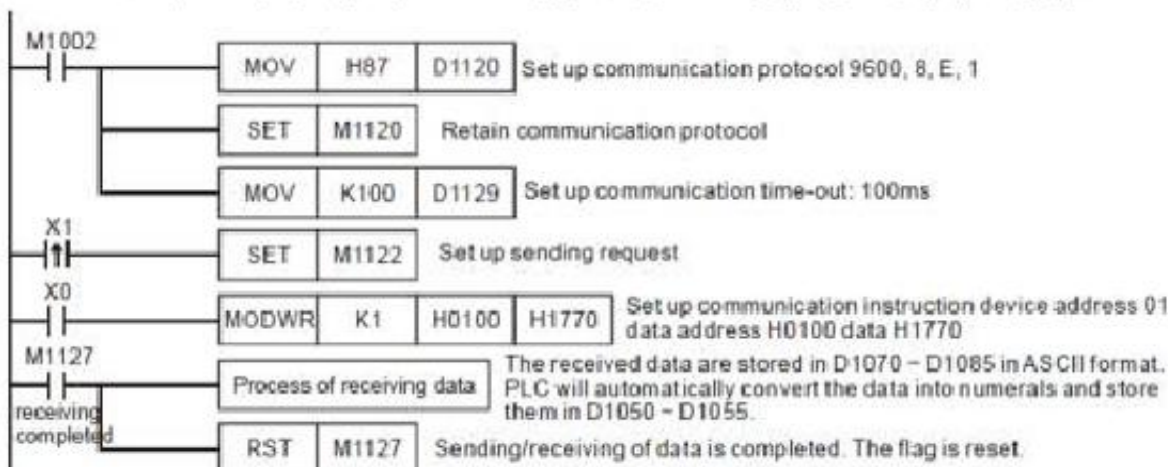
از فلگ **M1127** برای توقف دستورالعمل **MODRD** جهت اعلام تکمیل شدن پاسخ داده استفاده می‌شود. اگر پاسخ داده شده به داده صحیح باشد، فلگ **M1127** فعال (ON) می‌شود. فلگ **M1123** در هنگام پاسخ به داده فعال (ON) می‌شود. برای این فلگ (فلگ **M1123**) درست یا نادرست بودن پاسخ داده اهمیتی ندارد.

هیچ‌گونه محدودیتی برای دفعات استفاده از دستورالعمل **MODRD** در برنامه‌نویسی وجود ندارد. اما فقط در یک زمان می‌توان از یک دستورالعمل **MODRD** استفاده کرده و آن را اجرا کرد. در غیر این صورت، اگر دو دستورالعمل **MODRD** برنامه‌نویسی شده به صورت همزمان با هم اجرا شوند، برنامه خطا خواهد داد.

4-6-7 تبادل دیتا بین PLC و اینورتر VFD-B دلتا توسط دستورالعمل

MODBUS در مد ASCII

می‌خواهیم بین PLC دلتا و اینورتر سری VFD-B دلتا ارتباط شبکه MODBUS با مد ASCII برقرار کنیم. برای فعال کردن مد اسکی باید ابزار M1143 را غیرفعال (OFF) کنید.



PLC → VFD-B : ارسال مقدار "01 06 0100 1770 71" از PLC به اینورتر سری VFD-B دلتا.

VFD-B → PLC : ارسال مقدار "01 06 0100 1770 71" از اینورتر سری VFD-B به PLC دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) مطابق با جدول زیر می‌باشند:

Register	DATA		Explanation	
D1089 low	'0'	30 H	ADR 1	Address of AC motor drive: ADR (1,0)
D1089 high	'1'	31 H	ADR 0	
D1090 low	'0'	30 H	CMD 1	Instruction code: CMD (1,0)
D1090 high	'6'	36 H	CMD 0	
D1091 low	'0'	30 H	Data address	
D1091 high	'1'	31 H		
D1092 low	'0'	30 H		
D1092 high	'0'	30 H		
D1093 low	'1'	31 H	Data contents	
D1093 high	'7'	37 H		
D1094 low	'7'	37 H		
D1094 high	'0'	30 H		
D1095 low	'7'	37 H	LRC CHK 1	Checksum: LRC CHK (0,1)
D1095 high	'1'	31 H	LRC CHK 0	

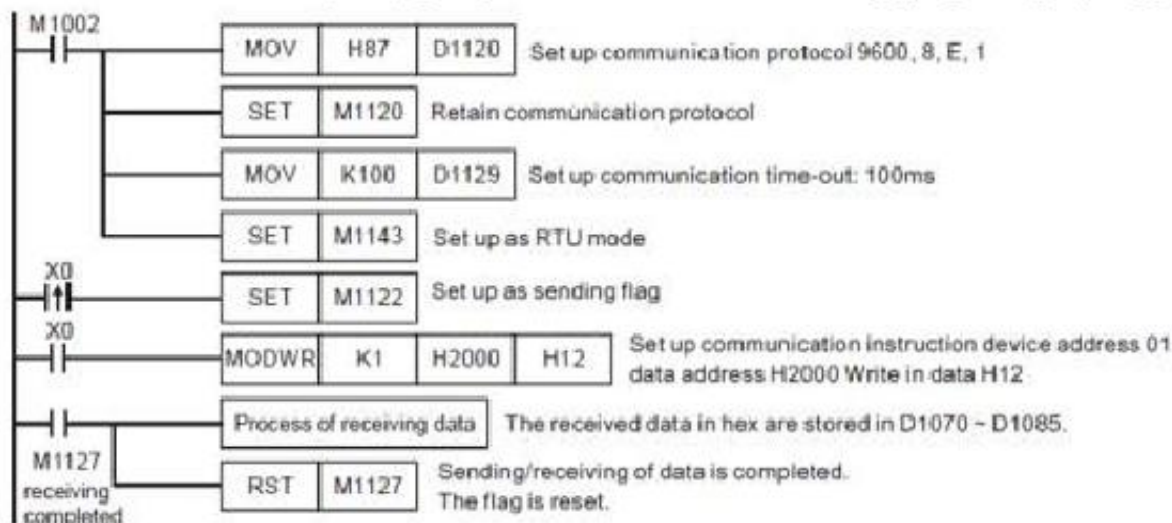
وضعیت رجیسترهای دریافت داده (پاسخ پیغام) مطابق با جدول زیر می‌باشند:

Register	DATA		Explanation
D1070 low	'0'	30 H	ADR 1
D1070 high	'1'	31 H	ADR 0
D1071 low	'0'	30 H	CMD 1
D1071 high	'6'	36 H	CMD 0
D1072 low	'0'	30 H	Data address
D1072 high	'1'	31 H	
D1073 low	'0'	30 H	
D1073 high	'0'	30 H	
D1074 low	'1'	31 H	Data contents
D1074 high	'7'	37 H	
D1075 low	'7'	37 H	
D1075 high	'0'	30 H	
D1076 low	'7'	37 H	LRC CHK 1
D1076 high	'1'	31 H	LRC CHK 0

5-6-7 دومین مثال از تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط

دستورالعمل MODBUS در مد RTU

می‌خواهیم بین PLC دلتا و اینورتر سری VFD-S دلتا ارتباط شبکه MODBUS با مد RTU برقرار کنیم. برای فعال کردن مد RTU باید فلگ M1143 را فعال (ON) کنید.



PLC → VFD-S : ارسال مقدار "01 06 2000 0012 02 07" از PLC دلتا به اینورتر

سری VFD-S دلتا.

VFD-S → PLC : ارسال مقدار "01 06 2000 0012 02 07" از اینورتر سری VFD-

S دلتا به PLC دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D1089 low	01 H	Address
D1090 low	06 H	Function
D1091 low	20 H	Data address
D1092 low	00 H	
D1093 low	00 H	Data contents
D1094 low	12 H	
D1095 low	02 H	CRC CHK Low
D1096 low	07 H	CRC CHK High

وضعیت رجیسترهای دریافت داده (پاسخ پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D1070 low	01 H	Address
D1071 low	06 H	Function
D1072 low	20 H	Data address
D1073 low	00 H	
D1074 low	00 H	Data contents
D1075 low	12 H	
D1076 low	02 H	CRC CHK Low
D1077 low	07 H	CRC CHK High

7-6-6 سومین مثال از تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط

دستورالعمل MODBUS در مد ASCII

می‌خواهیم بین PLC دلتا و اینورتر سری VFD-S دلتا ارتباط شبکه با فرمت MODBUS و مد ASCII برقرار نماییم. برای پیکربندی مد ASCII باید فلگ M1143 را غیرفعال (OFF) کرد.

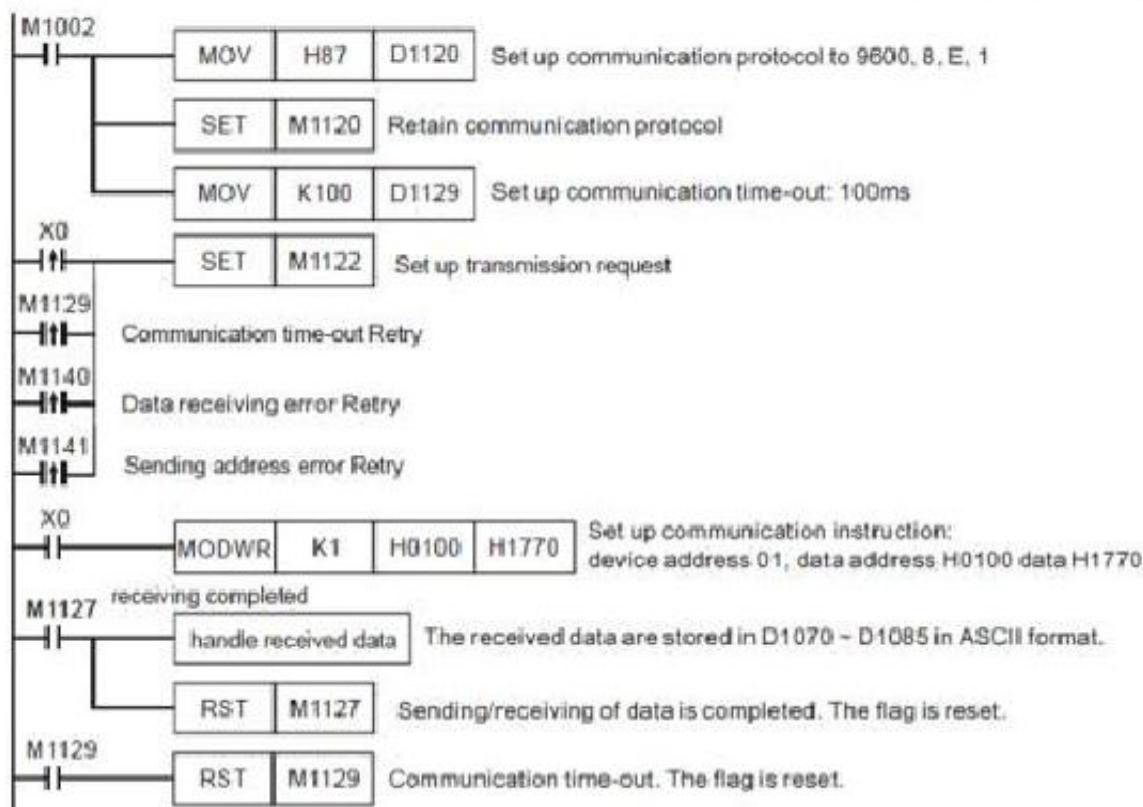
وقتی که خطای **time-out** ارتباطات، خطای دریافت داده در شبکه و یا خطای ارسال داده در شبکه رخ دهد، سیستم سعی می‌کند که مجدداً ارتباط شبکه را برقرار کند.

وقتی که **X0=ON** شود، PLC مقدار **H1770(K 6000)** به معنی عدد دسیمال و **H** به معنی عدد هگزادسیمال می‌باشد) را به اینورتر **VFD-S** دلتا با آدرس گره‌ی ارتباطات **01** ارسال کرده و داخل پارامتر **H0100** (آدرس هگزادسیمال پارامتر اینورتر سری **VFD-S** دلتا) می‌نویسید.

وقتی که خطای **time out** ارتباطات رخ دهد، فلگ خاص **M1129=ON** خواهد شد. برنامه فلگ خاص **M1129** را راه‌اندازی کرده و درخواستی به فلگ **M1122** برای نوشتن مجدد داده داخل پارامتر اینورتر ارسال خواهد کرد.

وقتی که خطای دریافت داده رخ دهد، فلگ خاص **M1140=ON** خواهد شد. برنامه فلگ خاص **M1140** را راه‌اندازی کرده و درخواستی به فلگ **M1122** جهت نوشتن مجدد داده داخل پارامتر اینورتر ارسال خواهد کرد.

وقتی که خطای ارسال داده در شبکه رخ دهد، فلگ **M1141=ON** خواهد شد. برنامه فلگ خاص **M1141** را راه‌اندازی کرده و درخواستی به فلگ **M1122** جهت نوشتن مجدد داده داخل پارامتر اینورتر ارسال خواهد کرد.



7-6-7 چهارمین مثال از تبادل دیتا بین PLC و اینورتر VFD-S دلتا توسط دستورالعمل MODBUS در مد ASCII

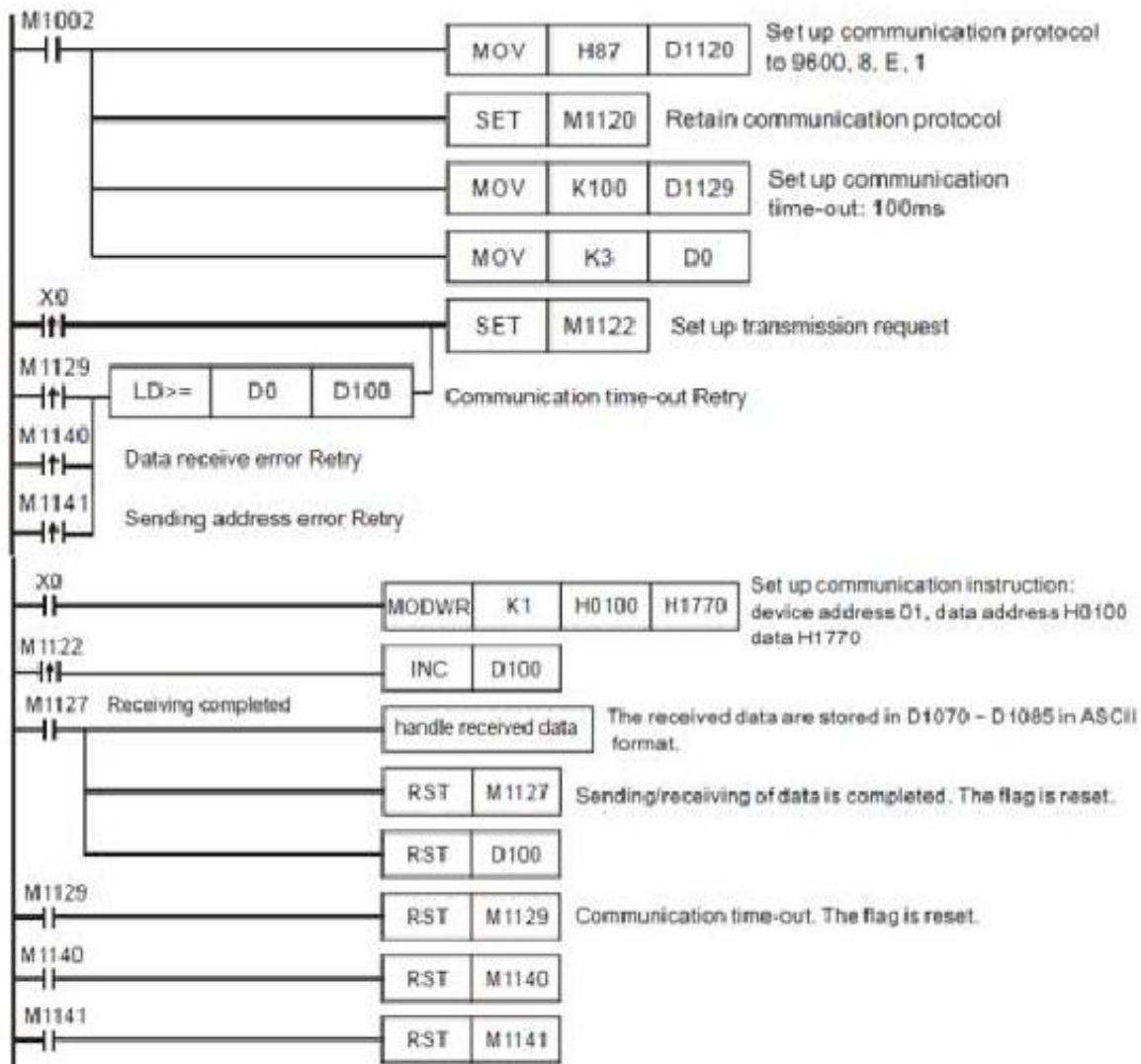
می‌خواهیم بین PLC دلتا و اینورتر سری VFD-S دلتا ارتباط شبکه با فرمت MODBUS و مد ASCII برقرار نماییم. برای پیکربندی مد ASCII باید فلگ M1143 غیرفعال (OFF) باشد. همچنین، وقتی که خطای time-out ارتباطات، خطای دریافت داده در شبکه و یا خطای ارسال داده در شبکه رخ داد، سیستم سعی کند که مجدداً ارتباط شبکه را برقرار نماید. تعداد دفعات سعی در برقراری ارتباطات در هنگام خطا در رجیستر D0 تنظیم می‌شود. مقدار پیش‌فرض این رجیستر برابر با 3 تنظیم شود. وقتی که سعی در برقراری مجدد ارتباطات با موفقیت صورت گرفت، کاربر می‌تواند به کنترل نرمال سیستم بازگردد.

وقتی که ورودی X0=ON شود، PLC مقدار H1770(K6000) را به اینورتر با آدرس گرهی ارتباطات 01 ارسال کرده و داخل پارامتر H0100 اینورتر می‌نویسد.

وقتی که خطای time out ارتباطات رخ دهد، فلگ خاص M1129=ON خواهد شد. برنامه فلگ خاص M1129 را راه‌اندازی کرده و درخواستی به فلگ M1122 برای نوشتن مجدد داده به داخل پارامتر اینورتر ارسال خواهد کرد. در صورت برطرف نشدن خطا، این عمل مجدداً صورت می‌گیرد. مجدداً اگر خطا برطرف نشد، این عمل دوباره صورت می‌گیرد. عمل ریست کردن خطا برای (D0=3) سه بار صورت خواهد گرفت.

وقتی که خطای دریافت داده رخ دهد، فلگ خاص M1140=ON خواهد شد. برنامه فلگ خاص M1140 را راه‌اندازی کرده و درخواستی به فلگ M1122 جهت نوشتن مجدد داده داخل پارامتر اینورتر ارسال خواهد کرد. اگر خطا ریست نشود، عمل ریست کردن مجدداً برای (D0=3) سه بار صورت خواهد گرفت.

وقتی که خطایی در ارسال آدرس رخ دهد، فلگ M1141 فعال (ON) خواهد شد. برنامه فلگ M1141 را راه‌اندازی کرده و درخواستی جهت نوشتن مجدد داده داخل پارامتر اینورتر به فلگ M1122 ارسال خواهد کرد. اگر خطا ریست نشود، عمل ریست کردن خطا برای (D0=3) سه بار تکرار خواهد شد.



توجه داشته باشید که:

برای تنظیمات رجیسترها و فلگ‌های خاص به توضیحات دستورالعمل RS مراجعه نمایید. برای شرایط فعال‌سازی سه دستورالعمل API 100 MODRD, API 101 MODWR and API 150 MODRW (Function Code H06, H10) نمی‌توان از کنتاکت‌های لبه بالا رونده (LDF, ANDF, LDP, ANDP, ORP) و کنتاکت‌های لبه‌ی پایین رونده (ORF) استفاده کرد. همچنین، برای راه‌اندازی سه دستورالعمل گفته شده باید ابتدا فلگ خاص M1122 را فعال کرده و سپس می‌توانید از دیگر شرط‌ها به غیر از کنتاکت‌های حساس به لبه استفاده نمایید.

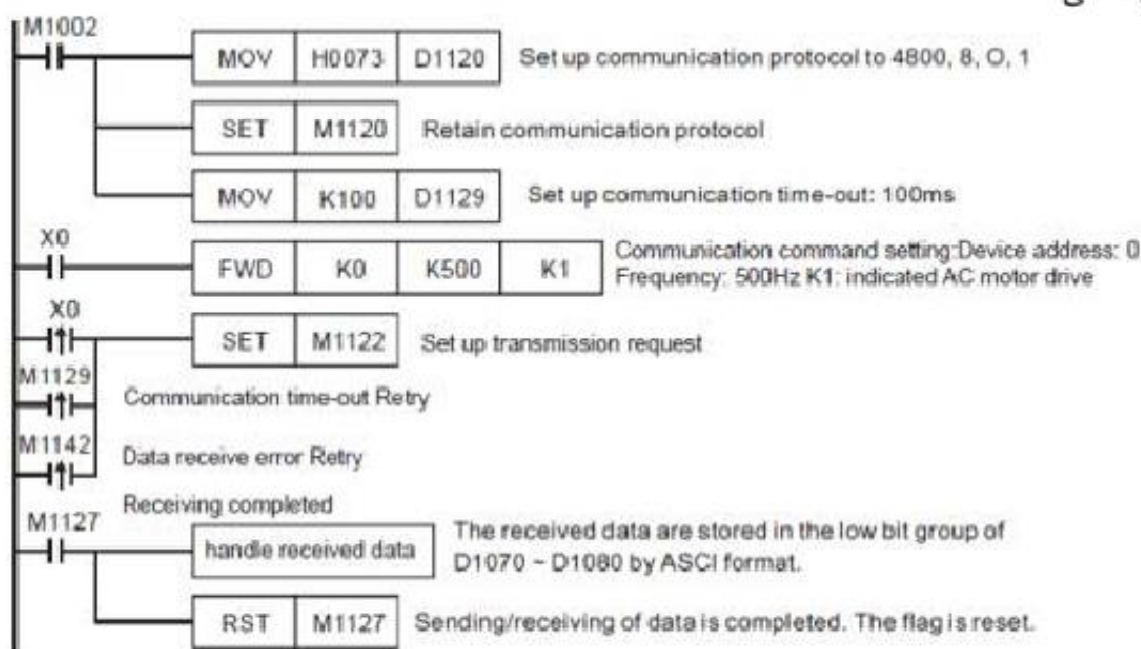
زمانی که پاسخ داده به دستورالعمل MODWR تکمیل شد، فلگ M1127 عملکرد دستورالعمل MODWR را متوقف کرده و یک وقفه در عملکرد آن می‌اندازد. اگر پاسخ داده درست باشد، فلگ

M1127 فعال (ON) خواهد شد. برای فلگ M1123 درست یا نادرست بودن پاسخ و یا وجود داشتن داده اهمیتی ندارد. زمانی که عمل پاسخ داده تکمیل شود، بدون توجه به درست بودن یا خطا داشتن پاسخ داده فلگ M1123 فعال (ON) خواهد شد.

محدودیتی برای تعداد دفعات استفاده از دستورالعمل MODWR در برنامه‌نویسی وجود ندارد. اما توجه داشته باشید که در یک زمان فقط مجاز به یک بار استفاده از دستورالعمل می‌باشید. اگر به صورت همزمان دو بار از دستورالعمل MODWR استفاده شود، برنامه خطا خواهد داد.

7-6-8 تبادل دیتا بین PLC و اینورتر VFD-A دلتا توسط دستورالعمل MODBUS

می‌خواهیم بین PLC دلتا و اینورتر سری VFD-A ارتباطات شبکه‌ی MODBUS برقرار نماییم. اگر خطای time-out شبکه و یا خطای دریافت داده رخ داد، مجدداً سیستم سعی در برقراری ارتباط می‌کند.



PLC → VFD-A : ارسال مقدار "C ♥ ☺ 0001 0500" از PLC دلتا به اینورتر سری VFD-A دلتا.

VFD-A → PLC : ارسال مقدار "C ♥ ♠ 0001 0500" از اینورتر سری VFD-A دلتا به PLC دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation
D1089 low	'C'	43 H	Start Word of instruction
D1090 low	'♥'	03 H	Checksum
D1091 low	'♠'	01 H	Reply authorization (correct: 06H, incorrect: 07 H)
D1092 low	'0'	30 H	Communication address
D1093 low	'0'	30 H	
D1094 low	'0'	30 H	
D1095 low	'1'	31 H	
D1096 low	'0'	30 H	Running instruction
D1097 low	'5'	35 H	
D1098 low	'0'	30 H	
D1099 low	'0'	30 H	

وضعیت رجیسترهای دریافت داده (پاسخ پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation
D1070 low	'C'	43 H	Start Word of instruction
D1071 low	'♥'	03 H	Checksum
D1072 low	'☺'	06 H	Instructed target
D1073 low	'0'	30 H	Communication address
D1074 low	'0'	30 H	
D1075 low	'0'	30 H	
D1076 low	'1'	31 H	
D1077 low	'0'	30 H	Running instruction
D1078 low	'5'	35 H	
D1079 low	'0'	30 H	
D1080 low	'0'	30 H	

توجه داشته باشید که محدودیتی در استفاده از این سه دستورالعمل وجود ندارد، اما از هر دستورالعمل تحت شبکه در یک زمان فقط باید یک بار استفاده کرد. اگر از دو دستورالعمل شبکه به صورت همزمان استفاده نمایید، شبکه خطا خواهد داد.

7- استفاده از کدهای تابعی استاندارد در شبکه Modbus

7-1 خواندن یک دسته دیتا توسط پورت COM2 با استفاده از کد تابع K3(H03)

در این مثال، می‌خواهیم کد تابع K3(H03) را توسط پورت COM2(RS-485) جهت خواندن چندین داده در شبکه ارسال نماییم. بعد از اینکه PLC دلتا به اینورتر سری VFD-S دلتا متصل شد و ارتباط برقرار کرد، برای فعال کردن مد ASCII در PLC باید فلگ خاص M1143 را غیرفعال

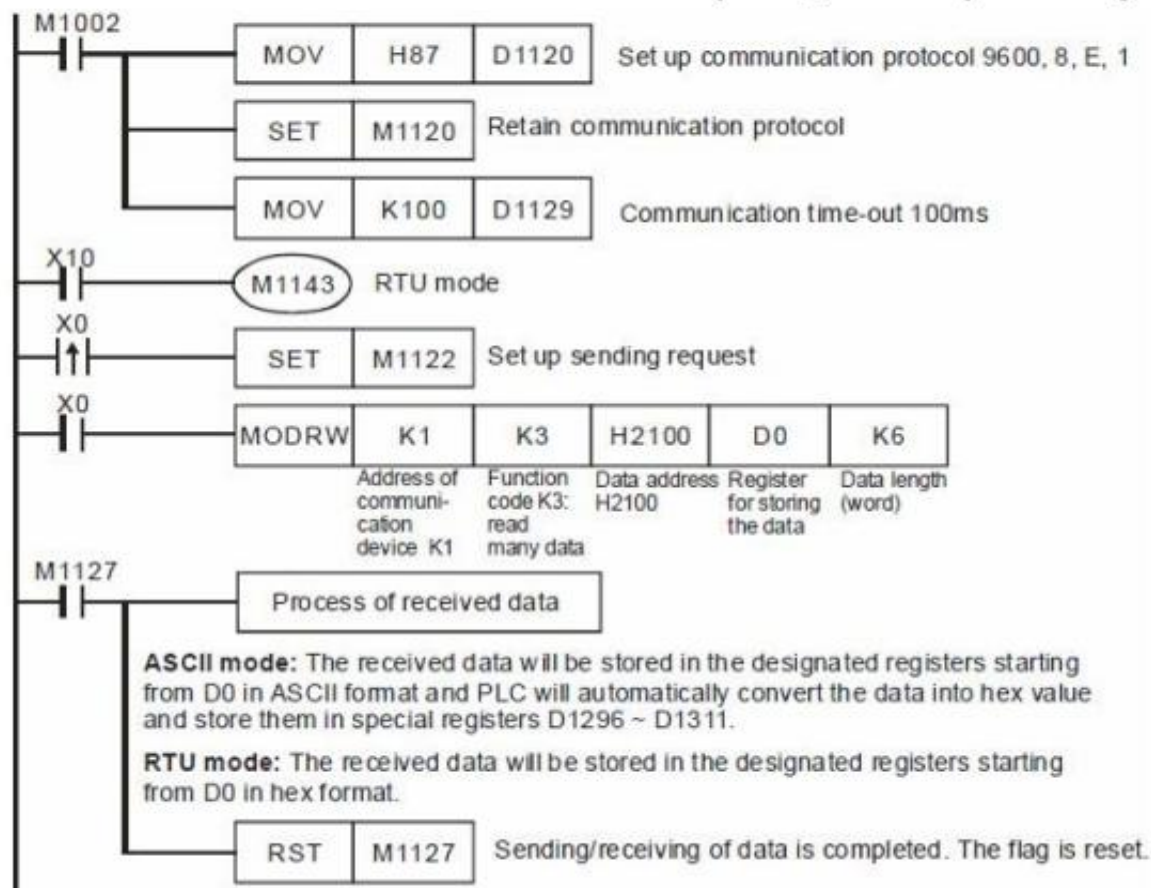
(OFF) کرد. همچنین، برای فعال کردن مد RTU در PLC باید فلگ خاص M1143 را فعال (ON) کرد.

وقتی که مد ASCII فعال باشد، دیتای دریافت شده در رجیسترهای تعیین شده ذخیره خواهند شد. رجیستر شروع از رجیستر D0 در فرمت اسکی می‌باشد که در این فرمت PLC به صورت اتوماتیک مقدار اسکی را به مقدار هگز تبدیل کرده و آن را در رجیسترهای خاص D1131~D1296 ذخیره می‌کند. وقتی که عمل تبدیل داده به هگز شروع شود، فلگ خاص M1131 فعال (ON) خواهد شد و وقتی که عمل تبدیل داده از فرمت اسکی به فرمت هگز تکمیل شود این فلگ خاص غیرفعال (OFF) می‌شود.

در صورت نیاز، می‌توانید مقدار هگز ذخیره شده در رجیسترهای D1131~D1296 را با استفاده از دستورالعمل‌های MOV یا DMOV و یا BMOV به رجیسترهای عمومی دیگر انتقال دهید. وقتی که مد RTU فعال باشد، دیتای دریافت شده در رجیسترهای تعیین شده ذخیره خواهد شد که آدرس رجیستر شروع D0 می‌باشد. در مد RTU داده‌ی دریافتی به فرمت هگز می‌باشد. وقتی که یکی از مدهای ASCII و یا RTU فعال باشد، PLC دیتاهایی را که باید ارسال شوند در رجیسترهای خاص D1256~D1295 ذخیره می‌کند. در صورت نیاز، می‌توانید توسط دستورالعمل‌های MOV یا DMOV و یا BMOV داده را به دیگر رجیسترهای عمومی منتقل کنید.

داده‌های ارسالی از اینورتر به PLC در رجیسترهای تعیین شده توسط کاربر ذخیره می‌شوند. بعد از تکمیل شدن ارسال دیتا، PLC به صورت اتوماتیک چک می‌کند که آیا دیتای دریافتی درست است، یا نادرست. اگر خطایی وجود داشته باشد، فلگ M1140 فعال (ON) خواهد شد. اگر آدرس ابزار اختصاص داده شده به وسیله‌ی تحت شبکه غیرمجاز باشد، آن وسیله‌ی تحت شبکه که به شبکه متصل شده است، پاسخی با عنوان پیغام خطا را به PLC ارسال می‌کند و PLC کد خطا را در رجیستر خاص D1130 ذخیره می‌کند و همچنین، فلگ خاص M1141 جهت اعلام خطا به کار فعال (ON) می‌شود. برای مثال، اگر مقدار 8000H برای اینورتر سری VFD-S دلتا غیرمجاز باشد، فلگ M1141 در PLC فعال (ON) شده و کد خطای 2 در رجیستر D1130 ذخیره می‌شود (D1130=2). برای اطلاع از کدهای خطای اینورتر سری VFD-S دلتا به دفترچه‌ی راهنمای اینورتر مراجعه نمایید.

بعد از اینکه M1140=ON و یا M1141=ON شد، PLC دیتای درست و مجاز دیگری را به اینورتر ارسال خواهد کرد. اگر فیدبک دیتا یا همان دیتای ارسالی از اینورتر به PLC صحیح باشد، فلگ خاص M1140 و M1141 ریست خواهد شد.



مد ASCII ← بعد از اینکه PLC دلتا به اینورتر سری VFD-S دلتا متصل شد، مد ASCII را فعال کنید.

VFD-S → PLC: ارسال مقدار "01 03 2100 0006 D5" از PLC دلتا به اینورتر سری VFD-S دلتا.

VFD-S → PLC: ارسال مقدار "01 03 0C 0100 1766 0000 0000 0136 0000 3B" از اینورتر سری VFD-S دلتا به PLC دلتا.

وضعیت رجیسترهای ارسال دیتا (ارسال پیام) مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation	
D1256 Low	'0'	30 H	ADR 1 ADR 0	Address of AC motor drive: ADR (1,0)
D1256 High	'1'	31 H		
D1257 Low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 High	'3'	33 H		
D1258 Low	'2'	32 H	Starting Data Address	
D1258 High	'11'	31 H		
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'0'	30 H	Number of Data (counted by Words)	
D1260 High	'0'	30 H		
D1261 Low	'0'	30 H		
D1261 High	'6'	36 H		
D1262 Low	'D'	44 H	LRC CHK 1 LRC CHK 0	Error checksum: LRC CHK (0,1)
D1262 High	'5'	35 H		

وضعیت رجیسترهای دریافت دیتای ابزار D (پاسخ پیام) مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation	
D0 Low	'0'	30 H	ADR 1	
D0 High	'1'	31 H	ADR 0	
D1 Low	'0'	30 H	CMD 1	
D1 High	'3'	33 H	CMD 0	
D2 Low	'0'	30 H	Number of Data (counted by byte)	
D2 High	'C'	43 H		
D3 Low	'0'	30 H	Content of address 2100H	PLC automatically convert ASCII codes to numerals and store the numeral in D1296 = H0100
D3 High	'1'	31 H		
D4 Low	'0'	30 H		
D4 High	'0'	30 H		
D5 Low	'1'	31 H	Content of address 2101H	PLC automatically convert ASCII codes to numerals and store the numeral in D1297 = H1766
D5 High	'7'	37 H		
D6 Low	'6'	36 H		
D6 High	'6'	36 H		
D7 Low	'0'	30 H	Content of address	PLC automatically convert ASCII codes to numerals and store the numeral in D1298 = H0000
D7 High	'0'	30 H		
D8 Low	'0'	30 H		

D8 High	'0'	30 H	2102H	
D9 Low	'0'	30 H	Content of address 2103H	PLC automatically convert ASCII codes to numerals and store the numeral in D1299 = H0000
D9 High	'0'	30 H		
D10 Low	'0'	30 H		
D10 High	'0'	30 H		
D11 Low	'0'	30 H	Content of address 2104H	PLC automatically convert ASCII codes to numerals and store the numeral in D1300 = H0136
D11 High	'1'	31 H		
D12 Low	'3'	33 H		
D12 High	'6'	36 H		
D13 Low	'0'	30 H	Content of address 2105H	PLC automatically convert ASCII codes to numerals and store the numeral in D1301 = H0000
D13 High	'0'	30 H		
D14 Low	'0'	30 H		
D14 High	'0'	30 H		
D15 Low	'3'	33 H	LRC CHK 1	
D15 High	'B'	42 H	LRC CHK 0	

مد RTU ← بعد از اینکه PLC دلتا به اینورتر سری VFD-S دلتا متصل شد، مد RTU را فعال کنید.

PLC → VFD-S : ارسال مقدار "01 03 2100 0006 CF F4" از PLC دلتا به اینورتر سری VFD-S دلتا.

VFD-S → PLC : ارسال مقدار "01 03 0C 0000 0503 0BB8 0BB8 0000 012D 8E C5" از اینورتر سری VFD-S دلتا به PLC دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D1256 Low	01 H	Address
D1257 Low	03 H	Function
D1258 Low	21 H	Starting Data Address
D1259 Low	00 H	
D1260 Low	00 H	Number of Data (counted by Words)
D1261 Low	06 H	
D1262 Low	CF H	CRC CHK Low
D1263 Low	F4 H	CRC CHK High

وضعیت رجیسترهای دریافت داده ابزار D (پاسخ پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D0 Low	01 H	Address
D1 Low	03 H	Function
D2 Low	0C H	Number of Data (byte)

D3 Low	00 H	Content of address 2100H	PLC automatically convert ASCII codes to numerals and store the numeral in D1296 = H0000
D4 Low	00 H		
D5 Low	05 H	Content of address 2101H	PLC automatically convert ASCII codes to numerals and store the numeral in D1297 = H0503
D6 Low	03 H		
D7 Low	0B H	Content of address 2102H	PLC automatically convert ASCII codes to numerals and store the numeral in D1298 = H0BB8
D8 Low	B8 H		
D9 Low	0B H	Content of address 2103H	PLC automatically convert ASCII codes to numerals and store the numeral in D1299 = H0BB8
D10 Low	B8 H		
D11 Low	00 H	Content of address 2104H	PLC automatically convert ASCII codes to numerals and store the numeral in D1300 = H0000
D12 Low	00 H		
D13 Low	01 H	Content of address 2105H	PLC automatically convert ASCII codes to numerals and store the numeral in D1301 = H012D
D14 Low	2D H		
D15 Low	8E H	CRC CHK Low	
D16 Low	C5 H	CRC CHK High	

7-2-7 نوشتن مقدار در یک دیتا رجیستر توسط پورت COM2 با استفاده از کد

تابع K6(H06)

در این مثال می‌خواهیم کد تابع K6(H06) را توسط پورت COM2 (RS-485) جهت نوشتن یک داده با طول یک کلمه (Word data) را در داخل یک رجیستر به شبکه ارسال کنیم. بعد از اتصال PLC دلتا به اینورتر سری VFD-S دلتا باید برای فعال کردن مد ASCII در PLC فلگ خاص M1143 غیرفعال (OFF) شود. همچنین، برای فعال کردن مد RTU در PLC فلگ خاص M1143 فعال (ON) شود.

وقتی که مد ASCII فعال است، شما باید داده را جهت نوشتن در رجیستر تعیین شده‌ی D50 به فرمت هگز ذخیره نمایید. دیتای برگشتی از اینورتر در رجیسترهای خاص D1070~D1076 ذخیره می‌شوند.

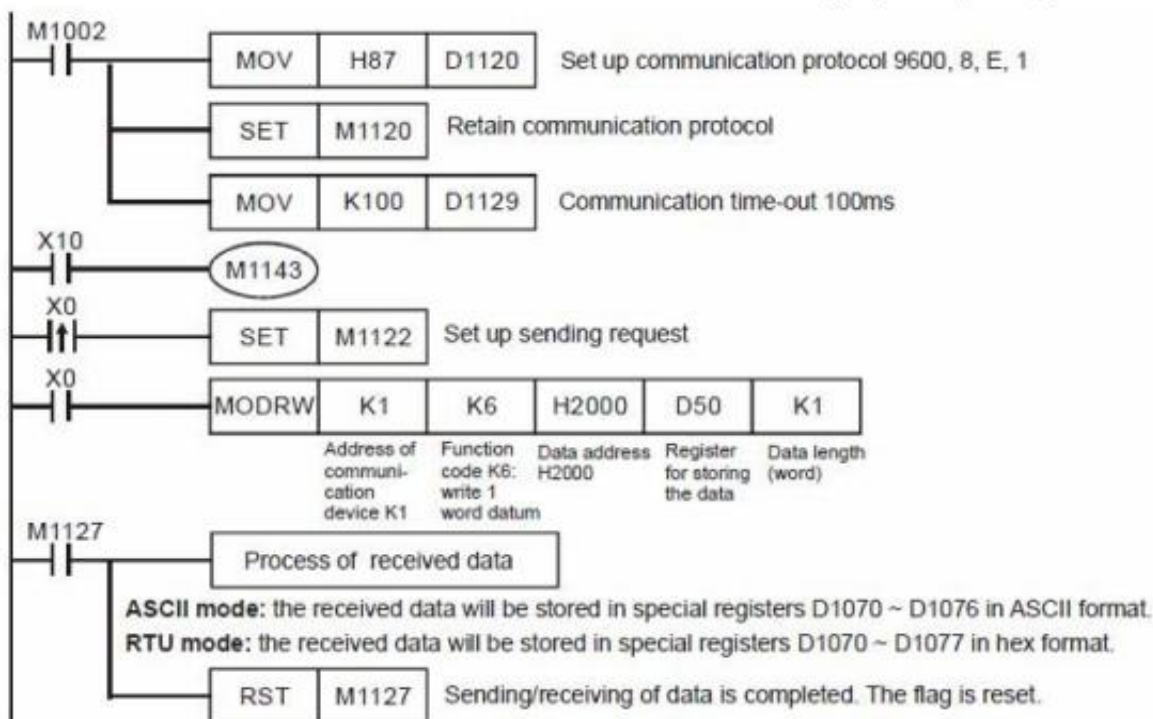
وقتی که مد RTU فعال است، شما باید داده را جهت نوشتن در رجیستر تعیین شده‌ی D50 به فرمت هگز ذخیره نمایید. دیتای برگشتی از اینورتر در رجیسترهای خاص D1070~D1077 ذخیره می‌شوند.

وقتی که یکی از مدهای RTU و ASCII فعال باشد، PLC دیتایی را که باید ارسال شود را در رجیسترهای خاص D1256~d1295 ذخیره خواهد کرد. در صورت لزوم، می‌توانید با استفاده از

دستورالعمل‌های MOV یا DMOV و یا BMOV دیتای تبدیلی در شبکه‌ی RS-485 را در رجیسترهای عمومی دیگر ذخیره نمایید.

بعد از اینکه عمل دریافت دیتای برگشتی از اینورتر تکمیل شد، PLC دیتای دریافتی را به صورت اتوماتیک چک خواهد کرد و از صحیح بودن و یا صحیح نبودن آن مطلع می‌شود. در صورتی که یک خطا در داده‌ی دریافتی توسط PLC وجود داشته باشد، فلگ خاص M1140 فعال (ON) خواهد شد. اگر آدرس گره‌ی ارتباطات وسیله تحت شبکه (در این مثال، اینورتر سری VFD-S دلتا) غیرمجاز باشد، وسیله‌ی تحت شبکه که به شبکه‌ی RS-485 متصل شده است (در این مثال، اینورتر VFD-S دلتا توسط شبکه‌ی ارتباطات RS-485 با پروتکل MODBUS به PLC دلتا متصل شده است) یک پیغام خطا را به PLC ارسال می‌کند و PLC کد خطا را در رجیستر خاص D1130 ذخیره می‌کند و در نهایت، برای اطلاع به کاربر جهت بوجود آمدن خطای ارتباطات شبکه فلگ خاص M1141 فعال (ON) می‌شود. برای مثال، اگر مقدار 8000H برای اینورتر سری VFD-S دلتا غیرمجاز باشد، فلگ M1141=ON شده و کد خطای 2 به رجیستر D1130 ارسال می‌شود. برای اطلاع از کدهای خطای اینورتر به دفترچه‌ی راهنمای اینورتر مراجعه نمایید.

بعد از اینکه فلگ خاص M1140 و یا M1141 فعال (ON) شدند، PLC داده‌ی دیگری را به اینورتر ارسال می‌کند. اگر دیتای برگشتی (فیدبک) از اینورتر صحیح باشد، فلگ‌های خاص M1140 و M1141 ریست (OFF) خواهند شد.



مد ASCII ← بعد از اینکه PLC دلتا به اینورتر VFD-S دلتا متصل شد، مد ASCII را فعال کنید.

VFD-S → PLC : ارسال مقدار "01 06 0100 1770 71" از PLC دلتا به اینورتر سری VFD-S دلتا.

VFD-S → PLC : ارسال مقدار "01 06 0100 1770 71" از اینورتر سری VFD-S دلتا به PLC دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation	
D1256 Low	'0'	30 H	ADR 1 ADR 0	Address of AC motor drive: ADR (1,0)
D1256 High	'1'	31 H		
D1257 Low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 High	'6'	36 H		
D1258 Low	'0'	30 H	Data Address	
D1258 High	'1'	31 H		
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'1'	31 H	Data content	The content of register D50 (H1770 = K6,000)
D1260 High	'7'	37 H		
D1261 Low	'7'	37 H		
D1261 High	'0'	30 H		
D1262 Low	'7'	37 H	LRC CHK 1 LRC CHK 0	LRC CHK (0,1) is error check
D1262 High	'1'	31 H		

وضعیت رجیسترهای دریافت داده (پاسخ پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation
D1070 Low	'0'	30 H	ADR 1 ADR 0
D1070 High	'1'	31 H	
D1071 Low	'0'	30 H	CMD 1 CMD 0
D1071 High	'6'	36 H	
D1072 Low	'0'	30 H	Data Address
D1072 High	'1'	31 H	
D1073 Low	'0'	30 H	
D1073 High	'0'	30 H	
D1074 Low	'1'	31 H	Data content
D1074 High	'7'	37 H	

D1075 Low	'7'	37 H	LRC CHK 1 LRC CHK 0
D1075 High	'0'	30 H	
D1076 Low	'7'	37 H	
D1076 High	'1'	31 H	

مد RTU ← بعد از اینکه PLC دلتا به اینورتر VFD-S دلتا متصل شد، مد RTU را فعال کنید.
 PLC → VFD-S : ارسال مقدار "01 06 2000 0012 02 07" از PLC دلتا به اینورتر سری VFD-S دلتا.
 VFD-S → PLC : ارسال مقدار "01 06 2000 02 07" از اینورتر سری VFD-S دلتا به PLC دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) مطابق با جدول زیر می‌باشد:

Register	DAT A	Explanation	
D1256 Low	01 H	Address	
D1257 Low	06 H	Function	
D1258 Low	20 H	Data Address	
D1259 Low	00 H		
D1260 Low	00 H	Data content	The content of register D50 (H12)
D1261 Low	12 H		
D1262 Low	02 H	CRC CHK Low	
D1263 Low	07 H	CRC CHK High	

وضعیت رجیسترهای دریافت داده (پاسخ پیغام) مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation	
D1070 Low	01 H	Address	
D1071 Low	06 H	Function	
D1072 Low	20 H	Data Address	
D1073 Low	00 H		
D1074 Low	00 H	Data content	
D1075 Low	12 H		
D1076 Low	02 H	CRC CHK Low	
D1077 Low	07 H	CRC CHK High	

7-3- نوشتن مقدار در یک دسته دیتا رجیستر توسط پورت COM2 با استفاده از کد تابع K16(H10)

در این مثال، می‌خواهیم کد تابع K16(H10) را توسط پورت COM2(RS-485) جهت نوشتن چندین داده داخل یک رجیستر را به شبکه ارسال نماییم. بعد از اتصال PLC دلتا به اینورتر سری

VFD-S دلتا باید برای فعال کردن مد ASCII در PLC فلگ خاص M1143 غیرفعال (OFF) شود. همچنین، برای فعال کردن مد RTU در PLC فلگ خاص M1143 فعال (ON) شود. وقتی که مد ASCII فعال است، شما باید داده را جهت نوشتن در رجیستر تعیین شده‌ی D50 به فرمت هگز ذخیره نمایید. دیتای برگشتی از اینورتر در رجیسترهای خاص D1070~D1076 ذخیره می‌شود.

وقتی که مد RTU فعال است، شما باید داده را جهت نوشتن در رجیستر تعیین شده‌ی D50 به فرمت هگز ذخیره نمایید. دیتای برگشتی از اینورتر در رجیسترهای خاص D1070~D1077 ذخیره می‌شود.

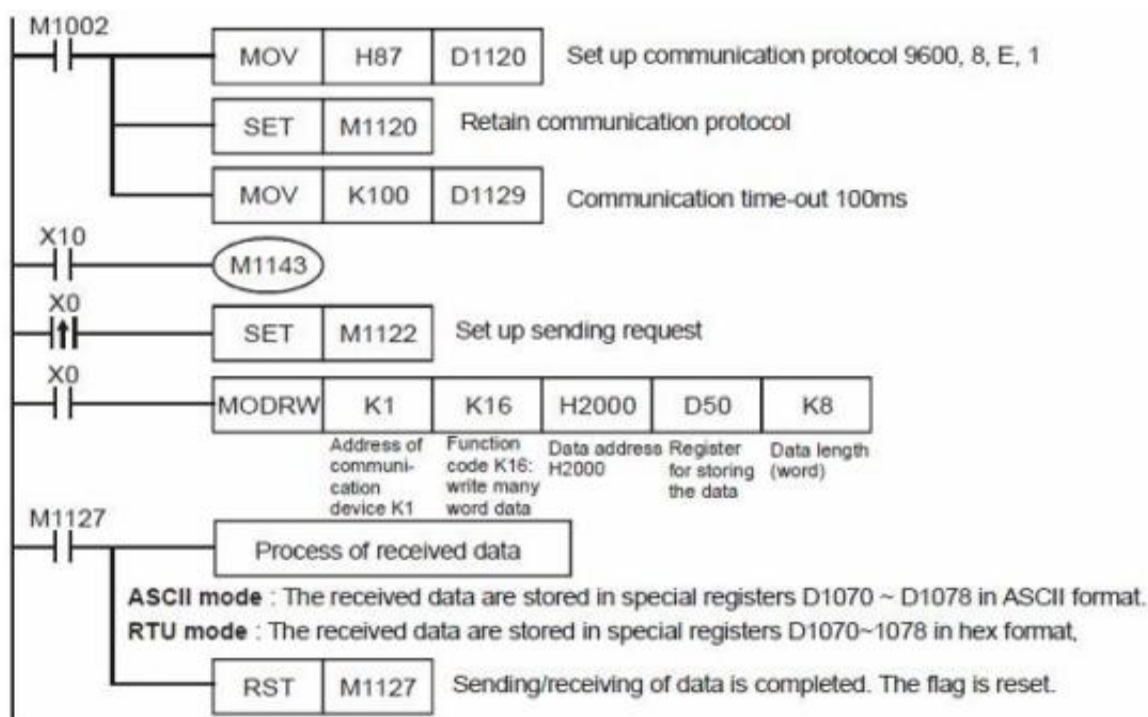
وقتی که یکی از مدهای RTU و یا ASCII فعال باشند، PLC مقدار دیتا را جهت ارسال به شبکه در رجیسترهای خاص D1256~D1295 ذخیره خواهد کرد. در صورت لزوم، می‌توانید با استفاده از دستورالعمل‌های MOV یا DMOV و یا BMOV داده را به رجیسترهای عمومی دیگر ارسال نمایید.

بعد از اینکه عمل دریافت داده برگشتی (فیدبک دیتا) از اینورتر به PLC تکمیل شد، PLC به صورت اتوماتیک داده‌ی دریافتی از اینورتر را چک می‌نماید. اگر خطایی در داده‌ی دریافتی از PLC وجود داشته باشد فلگ خاص M1140 فعال (ON) می‌شود.

اگر آدرس گره‌ی شبکه‌ی وسیله‌ی متصل شده به شبکه (در این مثال، اینورتر سری VFD-S دلتا از طریق شبکه به PLC دلتا متصل شده است) غیرمجاز باشد، آن وسیله ارتباطات یک پیغام خطا را به PLC ارسال می‌کند که در این حالت، کد خطای ارسالی از اینورتر به PLC در رجیستر خاص D1130 ذخیره می‌شود و همچنین، بعد از بروز خطای ارتباطات شبکه فلگ خاص M1141 در PLC فعال (ON) می‌شود. بعنوان مثال، اگر مقدار 8000H برای اینورتر سری VFD-S دلتا مجاز نباشد، فلگ خاص M1143 فعال (ON) شده و مقدار 2 به رجیستر خاص D1130 ارسال می‌شود. برای اطلاع از کدهای خطای اینورتر به دفترچه‌ی راهنمای آن مراجعه نمایید.

بعد از اینکه فلگ خاص M1140 و M1141 فعال (ON) شد، PLC دیتای دیگری را به اینورتر ارسال خواهد کرد.

اگر داده‌ی برگشتی از اینورتر به PLC از طریق شبکه صحیح باشد، فلگ‌های M1140 و M1141 ریست (OFF) خواهند شد.



مد ASCII ← بعد از اینکه PLC دلتا به اینورتر سری VFD-S متصل شد، مد ASCII را فعال نمایید.

PLC → VFD-S : ارسال مقدار "01 10 2000 0002 04 0012 1770 30" از PLC دلتا به اینورتر سری VFD-S دلتا.

VFD-S → PLC : ارسال مقدار "01 10 2000 0002 CD" از اینورتر سری VFD-S دلتا به PLC دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) از PLC به اینورتر مطابق با جدول زیر می‌باشد :

Register	DATA	Explanation
D1256 Low	'0' 30 H	ADR 1 ADR 0 Address of AC motor drive: ADR (1,0)
D1256 High	'1' 31 H	
D1257 Low	'1' 31 H	CMD 1 CMD 0 Command code: CMD (1,0)
D1257 High	'0' 30 H	
D1258 Low	'2' 32 H	Data Address
D1258 High	'0' 30 H	
D1259 Low	'0' 30 H	
D1259 High	'0' 30 H	
D1260 Low	'0' 30 H	Number of Registers
D1260 High	'0' 30 H	
D1261 Low	'0' 30 H	

D1261 High	'2'	32 H	Byte Count	
D1262 Low	'0'	30 H		
D1262 High	'4'	34 H		
D1263 Low	'0'	30 H	Data contents 1	(The content of register D50 (H12
D1263 High	'0'	30 H		
D1264 Low	'1'	31 H		
D1264 High	'2'	32 H		
D1265 Low	'1'	31 H	Data contents 2	(The content of register D51 (H1770 = K6,000
D1265 High	'7'	37 H		
D1266 Low	'7'	37 H		
D1266 High	'0'	30 H		
D1267 Low	'3'	33 H	LRC CHK 1 LRC CHK 0	Error checksum: LRC CHK (0,1)
D1267 High	'0'	30 H		

وضعیت رجیسترهای دریافت داده (پاسخ پیغام) از اینورتر به PLC مطابق با جدول زیر می‌باشند:

Register	DATA		Explanation
D1070 Low	'0'	30 H	ADR 1
D1070 High	'1'	31 H	ADR 0
D1071 Low	'1'	31 H	CMD 1
D1071 High	'0'	30 H	CMD 0
D1072 Low	'2'	32 H	Data Address
D1072 High	'0'	30 H	
D1073 Low	'0'	30 H	
D1073 High	'0'	30 H	
D1074 Low	'0'	30 H	Number of Registers
D1074 High	'0'	30 H	
D1075 Low	'0'	30 H	
D1075 High	'2'	32 H	
D1076 Low	'C'	43 H	LRC CHK 1
D1076 High	'D'	44 H	LRC CHK 0

مد RTU ← بعد از اینکه PLC دلتا به اینورتر VFD-S دلتا متصل شد، مد RTU را فعال کنید.

VFD-S → PLC : ارسال مقدار "01 10 2000 0002 04 0012 1770 C4 7F" از

PLC دلتا به اینورتر سری VFD-S دلتا.

VFD-S → PLC : ارسال مقدار "01 10 2000 0002 4A 08" از اینورتر سری VFD-

S دلتا به PLC دلتا.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) از PLC به اینورتر در سمت PLC مطابق با جدول زیر می‌باشد:

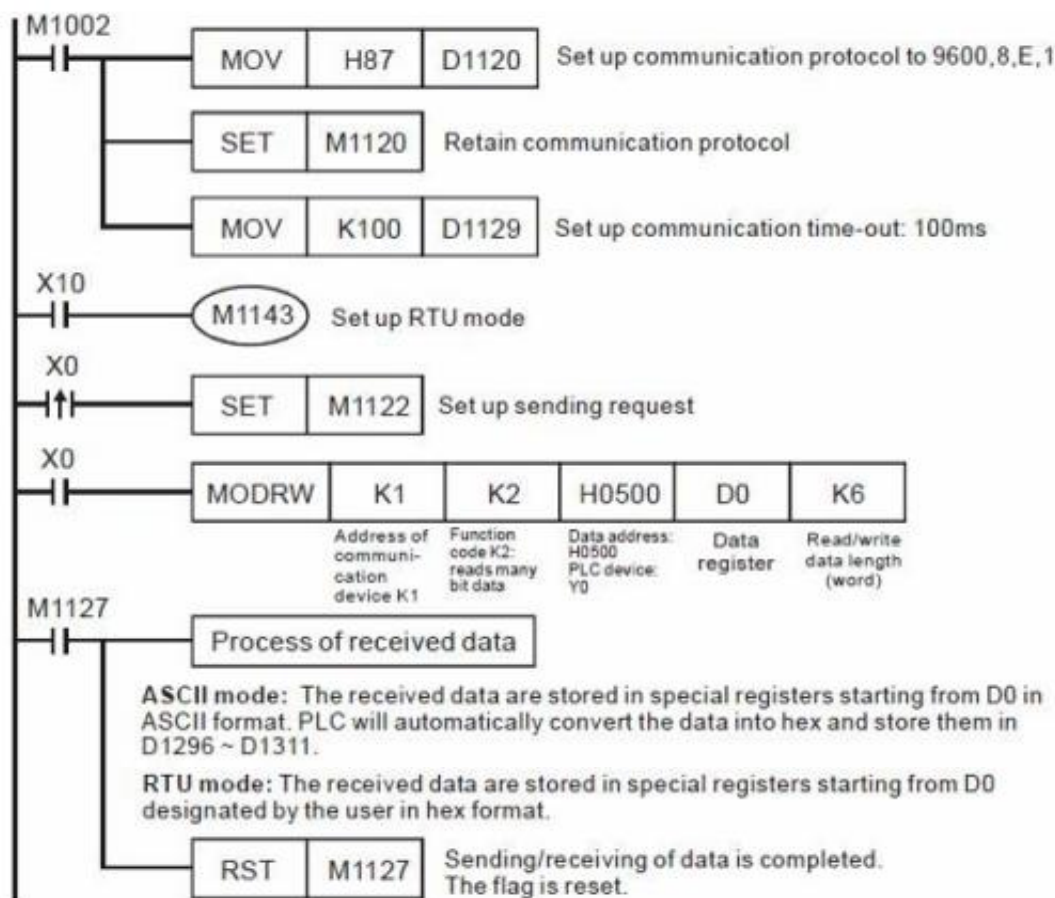
Register	DATA	Explanation	
D1256 Low	01 H	Address	
D1257 Low	10 H	Function	
D1258 Low	20 H	Data Address	
D1259 Low	00 H		
D1260 Low	00 H	Number of Registers	
D1261 Low	02 H		
D1262 Low	04 H	Byte Count	
D1263 Low	00 H	Data content 1	The content of register D50 (H12)
D1264 Low	12 H		
D1265 Low	17 H	Data content 2	The content of register D51 (H1770 = K6,000)
D1266 Low	70 H		
D1267 Low	C4 H	CRC CHK Low	
D1268 Low	7F H	CRC CHK High	

وضعیت رجیسترهای دریافت داده (پاسخ پیغام) از اینورتر به PLC در سمت PLC مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation	
D1070 Low	01 H	Address	
D1071 Low	10 H	Function	
D1072 Low	20 H	Data Address	
D1073 Low	00 H		
D1074 Low	00 H	Number of Registers	
D1075 Low	02 H		
D1076 Low	4A H	CRC CHK Low	
D1077 Low	08 H	CRC CHK High	

7-4 خواندن مقدار در یک دسته ابزار بیتی توسط پورت COM2 با استفاده از کد تابع K2(H02)

در این مثال، می‌خواهیم کد تابع K2(H02) را توسط پورت COM2(RS-485) جهت خواندن چندین ابزار بیتی به شبکه ارسال نماییم. دستورالعمل MODRW دارای پنج عملوند می‌باشد. دستورالعمل MODRW در دومین عملوند خود کد ارتباطات تنظیم شده را می‌خواند. در مثال زیر، K6 طول داده (بیت‌ها) می‌باشد. فرض شده است که وضعیت بیت‌های Y2=Y4=Y5=Y11=Y14 از بیت‌های Y0~Y16 فعال (ON) می‌باشد. در این مثال، کد ارتباطات خوانده شده در رجیستر تعیین شده توسط چهارمین عملوند دستورالعمل MODRW جایگزین خواهد شد.



مد ASCII ← وقتی که اولین PLC را از طریق شبکه RS-485 به دومین PLC متصل کردید، مد ASCII را فعال نمایید.

وقتی که X0 = ON شود، دستورالعمل MODRW با کد تابع 02 شروع به اجرا خواهد کرد. PLC1 → PLC2 : PLC1 مقدار "01 02 0500 0010 E8" را به PLC2 دلتا ارسال خواهد کرد.

PLC1 → PLC2 : PLC1 مقدار "01 02 02 34 12 B5" را از PLC2 دلتا دریافت خواهد کرد.

وضعیت رجیسترهای ارسال داده (ارسال پیام) از PLC1 مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation	
D1256 Low	'0'	30 H	ADR 1 ADR 0	Address of connected device: ADR (1,0)
D1256 High	'1'	31 H		
D1257 Low	'0'	30 H	CMD 1 CMD 0	(Command code: CMD (1,0)
D1257 High	'2'	32 H		
D1258 Low	'0'	30 H		

D1258 High	'5'	35 H	Starting Data Address	
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'0'	30 H	Number of Data (counted by bits)	
D1260 High	'0'	30 H		
D1261 Low	'1'	31 H		
D1261 High	'0'	30 H		
D1262 Low	'E'	45 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1262 High	'8'	38 H	LRC CHK 0	

وضعیت رجیستر D0 برای دریافت داده (پاسخ پیغام) در PLC1 مطابق با جدول زیر می‌باشد :

Register	DATA		Explanation		
D0 Low	'0'	30 H	ADR 1 ADR 0		
D0 High	'1'	31 H			
D1 Low	'0'	30 H	CMD 1 CMD 0		
D1 High	'2'	32 H			
D2 Low	'0'	30 H	Number of data (counted by bytes)		
D2 High	'2'	32 H			
D3 Low	'3'	33 H	Content in address 0500 ~0505	PLC automatically convert ASCII Words and store the result in D1296 = H1234 (b0 ~ b5 are valid)	
D3 High	'4'	34 H			
D4 Low	'1'	31 H			
D4 High	'2'	32 H			
D5 Low	'B'	52 H	LRC CHK 1		
D5 High	'5'	35 H	LRC CHK 0		

مد RTU ← وقتی که اولین PLC را از طریق شبکه RS-485 به دومین PLC متصل کردید،
مد RTU را فعال نمایید.

وقتی که ورودی X0=ON شود، دستورالعمل MODRW با کد تابع 02 شروع به اجرا خواهد
کرد.

ارسال خواهد کرد. PLC1 → PLC2 : دلتا مقدار "01 02 0500 0010 79 0A" را به PLC2 دلتا

دریافت خواهد کرد. PLC1 : PLC2 → PLC1 : دلتا مقدار "01 02 02 34 12 2F 75" را از PLC2 دلتا

وضعیت رجیسترهای ارسال داده (ارسال پیغام) از PLC مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D1256 low	1 H	Address
D1257 low	2 H	Function
D1258 low	5 H	Starting data address
D1259 low	0 H	
D1260 low	0 H	(Number of data (counted by Words
D1261 low	10 H	
D1262 low	79 H	CRC CHK Low
D1263 low	DA H	CRC CHK High

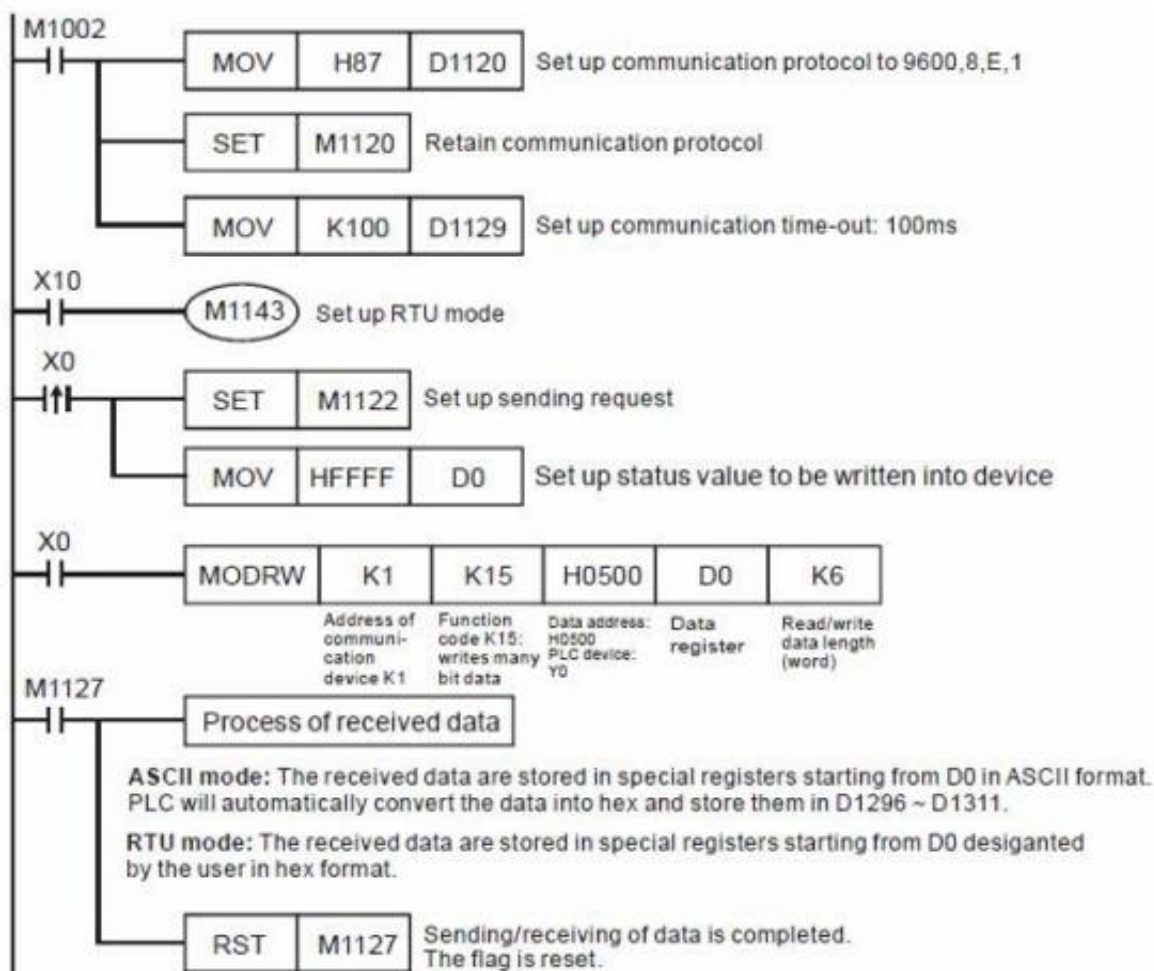
وضعیت رجیستر D0 برای دریافت داده (پاسخ پیغام) در PLC مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D0 low	01 H	Address
D1 low	02 H	Function
D2 low	02 H	Number of data (byte)
D3 low	34 H	Content in address 0500H PLC automatically stores the value in (D1296=H1234 (b0 ~ b5 are valid
D4 low	12 H	
D5 low	2F H	CRC CHK Low
D6 low	75 H	CRC CHK High

7-5- نوشتن مقدار در یک دسته ابزار بیتی توسط پورت COM2 با استفاده از

کد تابع K15(H0F)

در این مثال، می‌خواهیم کد تابع K15(H0F) را توسط پورت COM2(RS-485) جهت نوشتن چند ابزار بیتی در شبکه ارسال نماییم. وضعیت بیت جاری باید در رجیستر تعیین شده در چهارمین عملوند دستورالعمل MODRW در بیت‌های b0~b5 جایگزین شود. یک کلمه (1 Word) شامل 16 بیت وضعیت داده می‌باشد.



مد ASCII ← وقتی که اولین PLC دلتا را از طریق شبکه‌ی RS-485 به دومین PLC دلتا متصل کردید، مد ASCII را فعال نمایید.

وقتی که ورودی X0=ON شود، دستورالعمل MODRW با کد تابع 0F شروع به اجرا خواهد کرد.

ارسال می‌کند. PLC1 → PLC2 : مقدار داده "1 0F 0500 0006 01 3F A5" را به PLC2

دریافت می‌کند. PLC2 → PLC1 : مقدار داده "1 0F 0500 0006 E5" را از PLC2 دریافت

وضعیت رجیسترهای ارسال داده (ارسال پیغام) PLC1 مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation	
D1256 Low	'0'	30 H	ADR 1 ADR 0	Address of connected device: ADR (1,0)
D1256 High	'1'	31 H		
D1257 Low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 High	'F'	46 H		
D1258 Low	'0'	30 H	Data address	
D1258 High	'5'	35 H		
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'0'	30 H	Number of Data (counted by bits)	
D1260 High	'0'	30 H		
D1261 Low	'0'	30 H		
D1261 High	'6'	36 H		
D1262 Low	'0'	30 H	Byte Count	
D1262 High	'1'	31 H		
D1263 Low	'3'	33 H	Data content 1	Content in D0 register (H3F)
D1263 High	'F'	46 H		
D1264 Low	'A'	41 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1264 High	'5'	35 H	LRC CHK 0	

وضعیت رجیسترهای دریافت داده (پاسخ پیغام) PLC1 مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation
D1070 Low	'0'	30 H	ADR 1 ADR 0
D1070 High	'1'	31 H	
D1071 Low	'0'	30 H	CMD 1 CMD 0
D1071 High	'F'	46 H	
D1072 Low	'0'	30 H	Data Address
D1072 High	'5'	35H	
D1073 Low	'0'	30 H	
D1073 High	'0'	30 H	
D1074 Low	'0'	30 H	Number of Registers
D1074 High	'0'	30 H	
D1075 Low	'0'	30 H	
D1075 High	'6'	36 H	
D1076 Low	'E'	45 H	LRC CHK 1 LRC CHK 0
D1076 High	'5'	35 H	

مد RTU ← وقتی که اولین PLC دلتا از طریق شبکه RS-485 به دومین PLC دلتا متصل شد، مد RTU را فعال نمایید.

وقتی که ورودی X0=ON شود، دستورالعمل MODRW با کد تابع 15 شروع به اجرا خواهدکرد.

ارسال می‌کند. PLC1 → PLC2 : مقدار داده "01 0F 0500 0006 01 3F" را به PLC2

دریافت می‌کند. PLC1 : PLC2 → PLC1 مقدار داده "01 0F 0500 0006 D5 05" را از PLC2

وضعیت رجیسترهای ارسال داده (ارسال پیغام) در PLC مطابق با جدول زیر می‌باشد :

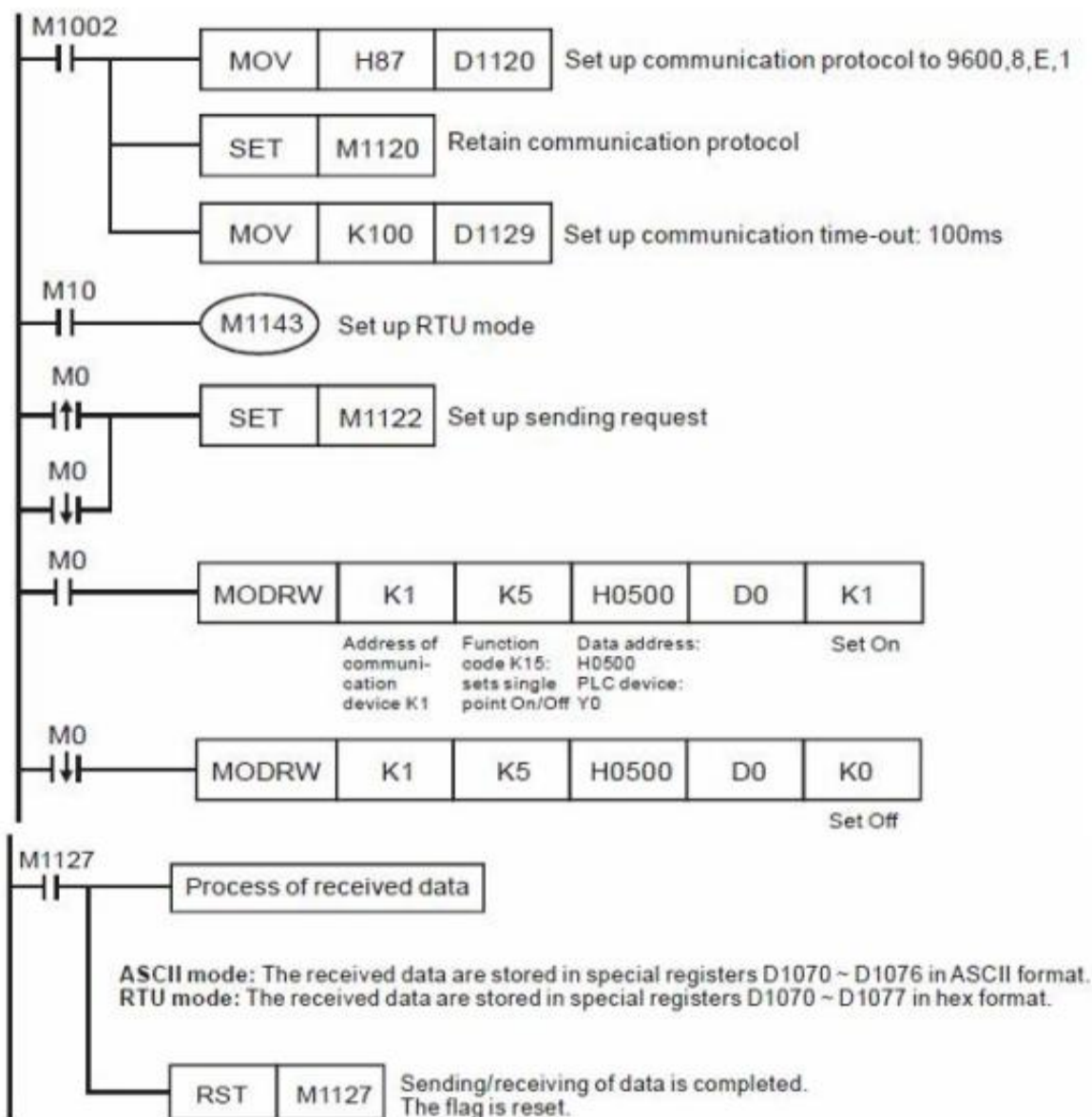
Register	DATA	Explanation		
D1256 low	01 H	Address		
D1257 low	0F H	Function		
D1258 low	05 H	Data address		
D1259 low	00 H			
D1260 low	00 H	Data content	(Content in D0 register (H3F	
D1261 low	06 H			
D1262 low	01 H	CRC CHK Low		
D1263 low	3F H	CRC CHK High		

وضعیت رجیسترهای دریافت داده (پاسخ پیغام) در PLC مطابق با جدول زیر می‌باشد :

Register	DATA	Explanation	
D1070 low	01 H	Address	
D1071 low	0F H	Function	
D1072 low	05 H	Data address	
D1073 low	00 H		
D1074 low	00 H	Data content	
D1075 low	06 H		
D1076 low	D5 H	CRC CHK Low	
D1077 low	05 H	CRC CHK High	

7-7-6 نوشتن مقدار در یک ابزار بیتی توسط پورت COM2 با استفاده از کد تابع K5(H05)

در این مثال، می‌خواهیم کد تابع K5(H05) را توسط COM2(RS-485) جهت نوشتن وضعیت یک ابزار بیتی به شبکه ارسال نماییم. در مثال زیر، وضعیت ابزار K1 را می‌خواهیم on و یا off نماییم.



مد ASCII ← وقتی که اولین PLC دلتا را از طریق شبکه RS-485 به دومین PLC دلتا متصل کردید، مد ASCII را فعال نمایید.

وقتی که M0=ON شود، دستورالعمل MODRW با کد تابع 05 شروع به اجرا کرده و بیت را فعال می‌کند.

PLC1 → PLC2 : مقدار داده "01 05 0500 FF00 F6" را به PLC2 ارسال می‌کند.

PLC1 : PLC2 → PLC1 : مقدار داده "01 05 0500 FF00 F6" را از PLC2 دریافت می‌کند.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) در PLC1 مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation	
D1256 Low	'0'	30 H	ADR 1	Address of connected device: ADR (1,0)
D1256 High	'1'	31 H	ADR 0	
D1257 Low	'0'	30 H	CMD 1	Command code: CMD (1,0)
D1257 High	'5'	35 H	CMD 0	
D1258 Low	'0'	30 H	Starting data address	
D1258 High	'5'	35 H		
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'F'	46 H	Request bit On/Off	
D1260 High	'F'	46 H		
D1261 Low	'0'	30 H		
D1261 High	'0'	30 H		
D1262 Low	'F'	46 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1262 High	'6'	36 H	LRC CHK 0	

وضعیت رجیستر D0 برای دریافت داده (پاسخ پیغام) در PLC1 مطابق با جدول زیر می‌باشد:

Register	DATA		Explanation	
D1070 Low	'0'	30 H	ADR 1	Address of connected device: ADR (1,0)
D1070 High	'1'	31 H	ADR 0	
D1071 Low	'0'	30 H	CMD 1	Command code: CMD (1,0)
D1071 High	'5'	35 H	CMD 0	
D1072 Low	'0'	30 H	Starting data address	
D1072 High	'5'	35H		
D1073 Low	'0'	30 H		
D1073 High	'0'	30 H		
D1074 Low	'F'	46 H	Request bit On/Off	
D1074 High	'F'	46 H		
D1075 Low	'0'	30 H		
D1075 High	'0'	30 H		
D1076 Low	'F'	46 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1076 High	'6'	36 H	LRC CHK 0	

وقتی که $M0=OFF$ شود، دستورالعمل MODRW با کد تابع 05 شروع به اجرا کرده و بیت را غیرفعال می‌کند.

PLC1 → PLC2 : مقدار داده "01 05 0500 FF00 F6" را به PLC2 ارسال می‌کند.

PLC2 → PLC1 : مقدار داده "01 05 0500 FF00 F6" را از PLC2 دریافت می‌کند.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) در PLC1 مطابق با جدول زیر می‌باشد :

Register	DATA		Explanation	
D1256 Low	'0'	30 H	ADR 1 ADR 0	Address of connected device: ADR (1,0)
D1256 High	'1'	31 H		
D1257 Low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1257 High	'5'	35 H		
D1258 Low	'0'	30 H	Starting data address	
D1258 High	'5'	35 H		
D1259 Low	'0'	30 H		
D1259 High	'0'	30 H		
D1260 Low	'0'	30 H	Request bit On/Off	
D1260 High	'0'	30 H		
D1261 Low	'0'	30 H		
D1261 High	'0'	30 H		
D1262 Low	'F'	46 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1262 High	'5'	35 H	LRC CHK 0	

وضعیت رجیستر D0 برای دریافت داده (پاسخ پیغام) در PLC1 مطابق با جدول زیر می‌باشد :

Register	DATA		Explanation	
D1070 Low	'0'	30 H	ADR 1 ADR 0	Address of connected device: ADR (1,0)
D1070 High	'1'	31 H		
D1071 Low	'0'	30 H	CMD 1 CMD 0	Command code: CMD (1,0)
D1071 High	'5'	35 H		
D1072 Low	'0'	30 H	Starting data address	
D1072 High	'5'	35H		
D1073 Low	'0'	30 H		
D1073 High	'0'	30 H		
D1074 Low	'0'	30 H		

D1074 High	'0'	30 H	Request bit On/Off	
D1075 Low	'0'	30 H		
D1075 High	'0'	30 H		
D1076 Low	'F'	46 H	LRC CHK 1	Error checksum: LRC CHK (0,1)
D1076 High	'5'	35 H	LRC CHK 0	

مد RTU ← بعد از اینکه اولین PLC دلتا را از طریق شبکه RS-485 به دومین PLC دلتا متصل کردید، مد RTU را فعال نمایید.

وقتی که M0=ON شود، دستورالعمل MODRW با کد تابع 05 شروع به اجرا کرده و بیت فعال (ON) می‌شود.

PLC1→PLC2 : PLC1 مقدار داده "01 05 0500 FF00 8C F6" را به PLC2 ارسال می‌کند.

PLC2→PLC1 : PLC1 مقدار داده "01 05 0500 FF00 8C F6" را از PLC2 دریافت می‌کند.

وضعیت رجیسترهای ارسال داده (ارسال پیغام) در PLC مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D1256 low	01 H	Address
D1257 low	05 H	Function
D1258 low	05 H	Starting data address
D1259 low	00 H	
D1260 low	FF H	Set bit On/Off
D1261 low	00 H	Request bit ON/OFF
D1262 low	8C H	CRC CHK Low
D1263 low	F6 H	CRC CHK High

وضعیت رجیستر D0 برای دریافت داده (پاسخ پیغام) در PLC مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D1070 low	01 H	Address
D1071 low	05 H	Function
D1072 low	05 H	Starting data address
D1073 low	00 H	
D1074 low	FF H	Set bit On/Off
D1075 low	00 H	Request bit ON/OFF
D1076 low	8C H	CRC CHK Low
D1077 low	F6 H	CRC CHK High

وقتی که M10=OFF شود، دستورالعمل MODRW با کد تابع 05 شروع به اجرا کرده و کد تابع 05 غیرفعال bit off می‌شود.

PLC1 → PLC2 : مقدار داده "01 05 0500 0000 CD 06" را به PLC2 ارسال می‌کند.

PLC2 → PLC1 : مقدار داده "01 05 0500 0000 CD 06" را از PLC2 دریافت می‌کند.

وضعیت رجیسترهای ارسال داده (ارسال پیام) در PLC مطابق با جدول زیر می‌باشد:

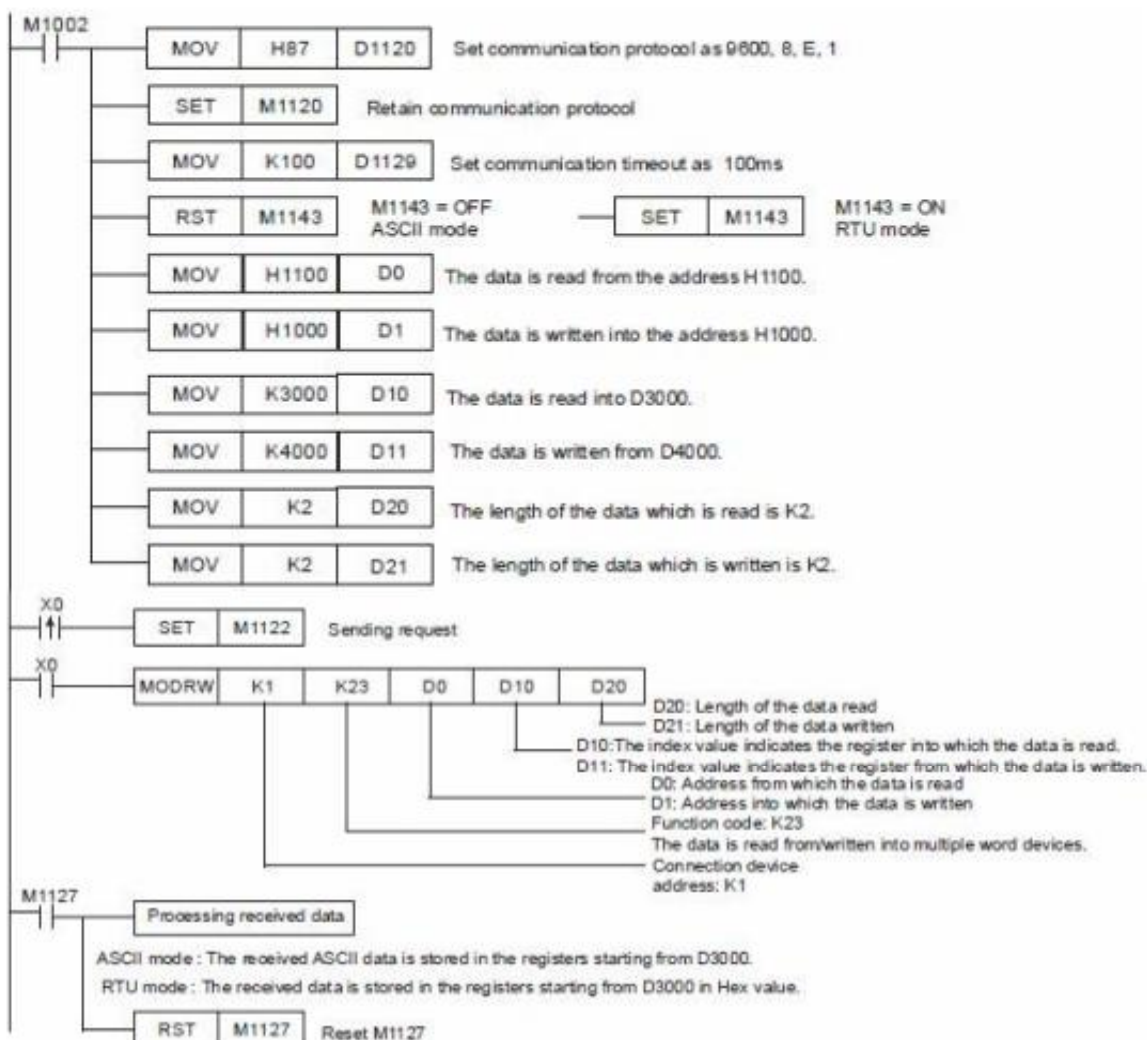
Register	DATA	Explanation
D1256 low	01 H	Address
D1257 low	05 H	Function
D1258 low	05 H	Starting data address
D1259 low	00 H	
D1260 low	00 H	Set bit On/Off
D1261 low	00 H	Request bit ON/OFF
D1262 low	CD H	CRC CHK Low
D1263 low	06 H	CRC CHK High

وضعیت رجیستر D0 برای دریافت داده (پاسخ پیام) در PLC مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D1070 low	01 H	Address
D1071 low	05 H	Function
D1072 low	05 H	Starting data address
D1073 low	00 H	
D1074 low	00 H	Set bit On/Off
D1075 low	00 H	Request bit ON/OFF
D1076 low	CD H	CRC CHK Low
D1077 low	06 H	CRC CHK High

7-7-7 خواندن و نوشتن مقدار در یک دسته ابزار بیتی توسط پورت COM2 با استفاده از کد تابع K23(H17)

در این مثال می‌خواهیم کد تابع K23(H17) را توسط پورت COM2(RS-485) جهت خواندن و نوشتن مقدار در چندین ابزار بیتی به شبکه‌ی RS-485 ارسال نماییم.



مد ASCII ← برای فعال کردن مد ASCII در شبکه فلگ خاص M1143 را غیرفعال OFF نمایید. وقتی که ورودی X0=ON شود، دستورالعمل MODRW با تابع تعیین شده توسط کد تابع H17 اجرا می‌شود.

PLC-A → PLC-B مقدار داده‌ی "01 07 1100 0002 1000 0002 04"

PLC-B را به 1770 0012 06" ارسال می‌کند.

PLC-A : PLC-B → PLC-A مقدار داده‌ی "01 17 04 0100 1766 66" را از PLC-B دریافت می‌کند.

وضعیت رجیسترهای دریافت داده (پاسخ پیغام) در PLC-A مطابق با جدول زیر می‌باشد:

Register	Data		Description
D3000 Low byte	'0'	30 H	ADR 1 ADR 0
D3000 High byte	'1'	31 H	
D3001 Low byte	'1'	31 H	CMD 1 CMD 0
D3001 High byte	'7'	37 H	
D3002 Low byte	'0'	30 H	Number of data (bytes)
D3002 High byte	'4'	34 H	
D3003 Low byte	'0'	30 H	Contents of the address 1100H
D3003 High byte	'1'	31 H	
D3004 Low byte	'0'	30 H	
D3004 High byte	'0'	30 H	
D3005 Low byte	'1'	31 H	Contents of the address 1101H LRC CHK 1 LRC CHK 0
D3005 High byte	'7'	37 H	
D3006 Low byte	'6'	36 H	
D3006 High byte	'6'	36 H	
D3007 Low byte	'6'	36 H	
D3007 High byte	'6'	36 H	

مد RTU ← برای فعال کردن مد RTU در شبکه فلگ خاص M1143 را فعال ON نمایید.
وقتی که ورودی X0=ON شود، دستورالعمل MODRW با تابع تعیین شده توسط کد تابع H17 اجرا می‌شود.

PLC-A → PLC-B : مقدار داده‌ی "01 07 1100 0002 1000 0002 04 1770 0012 A702" را به PLC-B ارسال می‌کند.

PLC-A : PLC-B → PLC-A : مقدار داده‌ی "01 17 04 0100 1766 7701" را از PLC-B دریافت می‌کند.

وضعیت رجیستر D0 برای دریافت داده (پاسخ پیغام) در PLC-A مطابق با جدول زیر می‌باشد:

Register	Data	Description
D3000 Low byte	01 H	Address
D3001 Low byte	17 H	Function
D3002 Low byte	04 H	Number of data (bytes)
D3003 Low byte	01 H	Contents of the address 1100H
D3004 Low byte	00 H	
D3005 Low byte	17 H	Contents of the address 1101H
D3006 Low byte	66 H	
D3007 Low byte	77 H	CRC CHK Low
D3008 Low byte	01 H	CRC CHK High

نکته:

از کنتاکت‌های تشخیص لبه نباید برای شرط راه‌اندازی دستورالعمل MODRW و RDST و MODRD استفاده نمایید. در غیر این صورت، خطایی در داده‌ی ذخیره شده در رجیسترهای دریافتی رخ خواهد داد.

فلگ‌ها و رجیسترهای خاص برای دستورالعمل‌های MODRW در ارتباطات شبکه‌ی RS-485 مطابق با توضیحات گفته شده برای دستورالعمل RS 80 API می‌باشد.

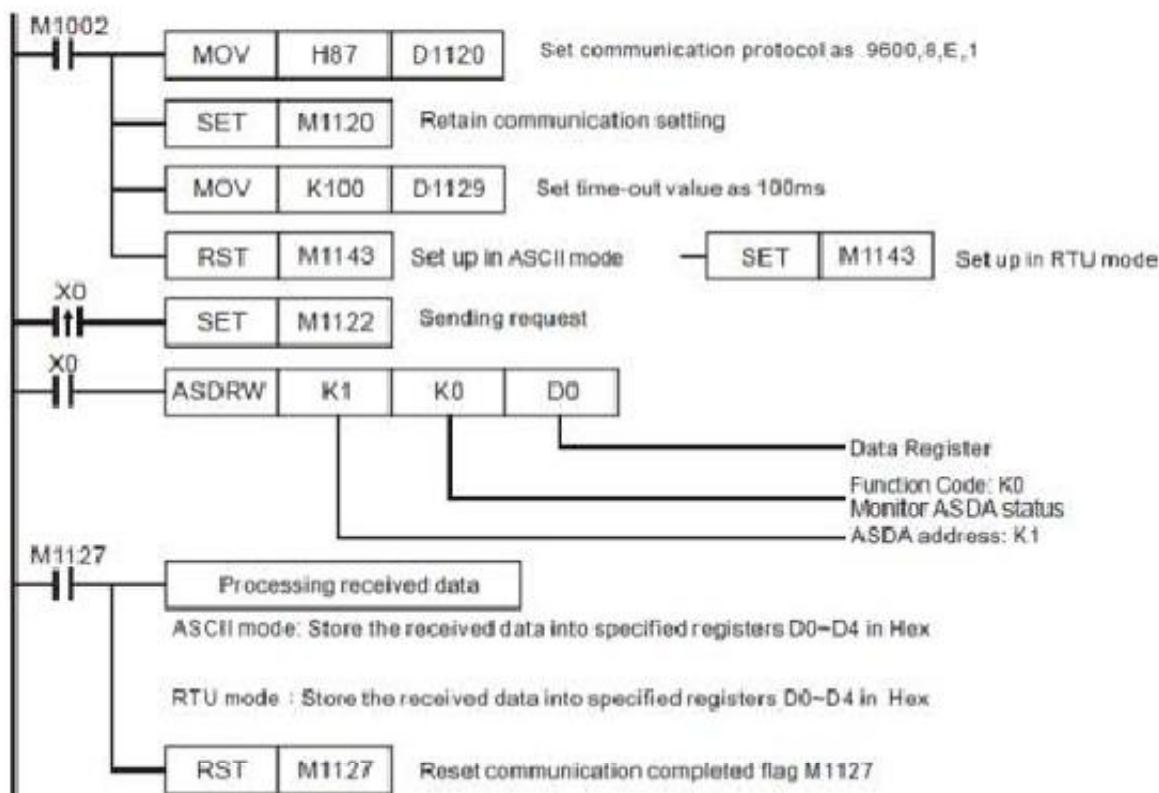
فلگ‌ها و رجیسترها	توضیحات
M1120	از این فلگ برای نگهداشتن و حفظ کردن تنظیمات ارتباطات شبکه استفاده می‌شود. بعد از اینکه تنظیمات شبکه انجام دادید و فلگ M1120 را فعال کردید، تغییرات جدید در رجیستر D1120 معتبر نخواهد بود.
M1121	وقتی که این فلگ غیرفعال OFF شود، دیتا را می‌توان در شبکه RS-485 ارسال کرد.
M1122	از این فلگ برای درخواست ارسال داده در شبکه RS-485 استفاده می‌شود.
M1123	از این فلگ برای اعلام تکمیل شدن عمل دریافت داده در شبکه RS-485 استفاده می‌شود.
M1124	از این فلگ برای اعلام عمل انتظار برای دریافت داده در شبکه RS-485 استفاده می‌شود.
M1125	از این فلگ برای غیرفعال کردن وضعیت دریافت داده در شبکه RS-485 استفاده می‌شود.
M1126	از این فلگ برای انتخاب STX/ETX سیستم در شبکه RS-485 استفاده می‌شود.
M1127	از این فلگ برای اعلام تکمیل شدن عمل ارسال داده و یا دریافت داده توسط دستورالعمل‌های MODRW و RDST و MODRD در شبکه RS-485 استفاده می‌شود.
M1128	از این فلگ برای فعال کردن عمل ارسال داده و یا دریافت داده در شبکه RS-485 استفاده می‌شود.
M1129	از این فلگ برای اعلام خطای time-out در دریافت داده در شبکه RS-485 استفاده می‌شود.
M1130	از این فلگ برای تعریف کردن STX/ETX توسط کاربر و یا سیستم در شبکه RS-485 استفاده می‌شود.
M1131	وقتی که دستورالعمل‌های MODRW و MODWR و MODRD در حال تبدیل داده به فرمت هگز باشند، این فلگ ON می‌شود.
M1140	وقتی که خطایی در دریافت داده توسط دستورالعمل‌های MODRW و MODWR و MODRD رخ دهد، این فلگ ON می‌شود.
M1141	وقتی که خطایی در پارامترهای دستورالعمل‌های MODRW و MODWR و MODRD رخ دهد، این فلگ ON می‌شود.
M1142	وقتی که خطایی در دریافت داده توسط دستورالعمل دریافت دستی داده از اینورتر سری VFD-A رخ دهد، این فلگ ON می‌شود.
M1143	از این فلگ برای انتخاب مد RTU/ASCII برای اجرای دستورالعمل‌های MODRW و MODRD و MODRD استفاده می‌شود. RTU mode → M1143=OFF و ASCII mode → M1143=OFF
D1070~D1085	وقتی که دستورالعمل ارتباطات RS-485 نوشته شده در برنامه اجرا شود و داده‌ای به خارج از PLC ارسال شود، و یا داده‌ای از خارج توسط PLC دریافت شود (پیغام و پاسخ پیغام) ابتدا داده باید در رجیسترهای D1070~D1085 ذخیره شوند. بنابراین، این رجیسترها واسطی بین شبکه و دیگر رجیسترهای PLC

می‌باشند.	
در این رجیستر پروتکل ارتباطات RS-485 تنظیم می‌شود.	D1120
در این رجیستر آدرس گرهی ارتباطات PLC تنظیم می‌شود. آدرس گرهی ارتباطات PLC از نوع حافظه‌دار بوده و پاک شدنی نیست. این آدرس فقط توسط برنامه‌نویس تغییر می‌کند.	D1121
Wordهایی که برای ارسال داده باقی‌مانده‌اند در این رجیستر قرار می‌گیرند.	D1122
Wordهایی که برای دریافت داده باقی‌مانده‌اند در این رجیستر قرار می‌گیرند.	D1123
در این رجیستر متن شروع (STX) تعریف می‌شود.	D1124
در این رجیستر نخستین متن پایان (ETX1) تعریف می‌شود.	D1125
در این رجیستر دومین متن پایان (ETX2) تعریف می‌شود.	D1126
در این رجیستر زمان خطای time-out ارتباطات شبکه‌ی RS-485 برحسب ms تنظیم می‌شود.	D1129
در این رجیستر کدهای خطای برگشتی از شبکه MODBUS ثبت می‌شود.	D1130
وقتی که دستورالعمل MODRW ارتباطات RS-485 برنامه‌نویسی شده اجرا شود، داده‌های ذخیره شده در رجیسترهای خاص D1256~D1295 به خارج از PLC ارسال می‌شوند.	D1256~D1295
PLC به صورت اتوماتیک داده ASCII ذخیره شده در رجیستر تعیین شده توسط کاربر را در این رجیسترها به فرمت هگز تبدیل می‌کند و مجدداً مقدار هگز در رجیسترهای تعیین شده توسط کاربر ذخیره می‌شوند.	D1296~D1311

7-8- دستورات اختصاصی دلتا برای راه‌اندازی سرویس‌های دلتا در شبکه Modbus

7-8-1 راه‌اندازی سرودرایو دلتا از طریق شبکه RS-485 پورت COM2 (RS-485) در این مثال، می‌خواهیم سرودرایو دلتا را از طریق شبکه RS-485 به پورت COM2 (RS-485) از PLC دلتا متصل نموده و سرودرایو را از طریق شبکه RS-485 توسط PLC دلتا راه‌اندازی نماییم.

وقتی که ورودی X0=ON شود، PLC اطلاعاتی را از طریق پورت ارتباطات COM2 برای خواندن وضعیت سرودرایو به سرودرایو ارسال می‌کند. وقتی که PLC فیدبک داده را از سرودرایو ASDA دلتا دریافت کرد، فلگ M1127 فعال شده و داده‌ی خوانده شده از سرودرایو در رجیسترهای D0~D4 ذخیره خواهد شد.

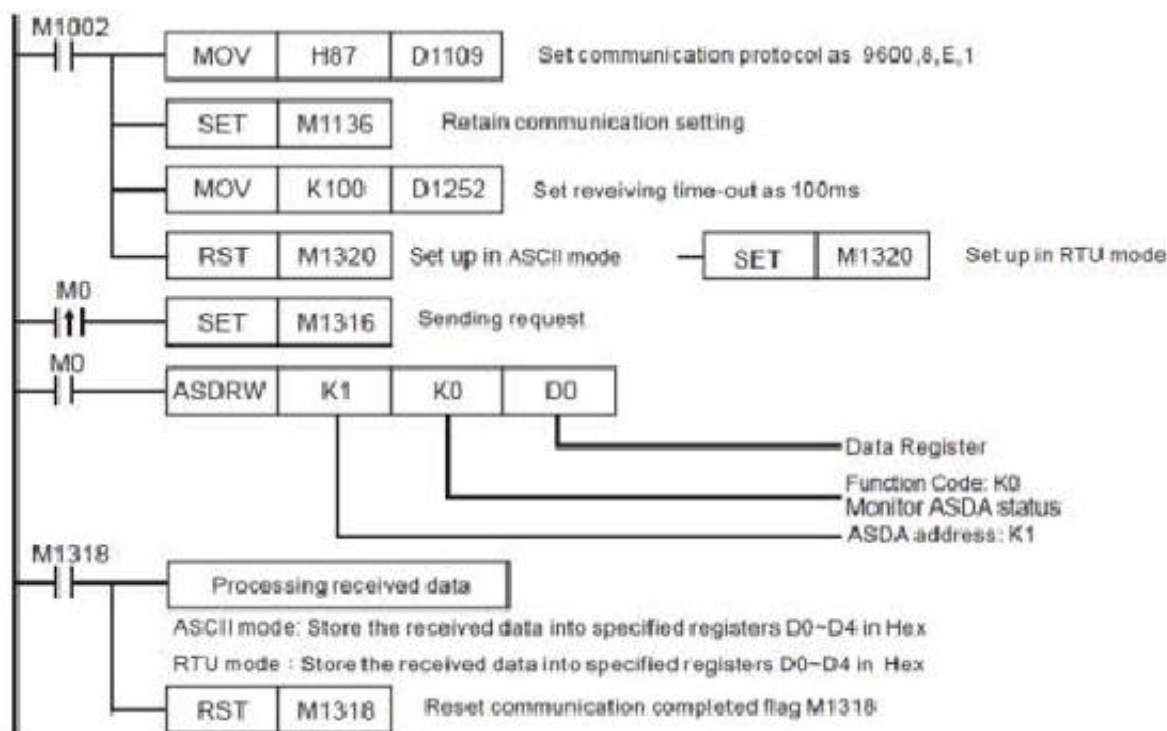


7-8-2 راه‌اندازی سرودرایو دلتا از طریق شبکه RS-485 پورت COM3 (RS-485)

در این مثال، می‌خواهیم سرودرایو سری ASDA دلتا را از طریق شبکه RS-485 به پورت COM3 (RS-485) از PLC دلتا متصل نموده و با استفاده از شبکه‌ی Modbus سرودرایو را توسط PLC کنترل نماییم.

وقتی که M0=ON شود، PLC فرامین را از طریق پورت ارتباطات COM2 برای خواندن وضعیت سرودرایو به سرودرایو ارسال می‌کند.

وقتی که فیدبک داده از سرودرایو ASDA به PLC ارسال شد، فلگ خاص M1318 فعال شده و داده‌های خوانده شده در رجیسترهای D0~D4 ذخیره خواهند شد.

**نکته:**

فلگ‌ها (M) و رجیسترهای (D) خاص وابسته به پورت COM2/COM3 مطابق با جدول زیر

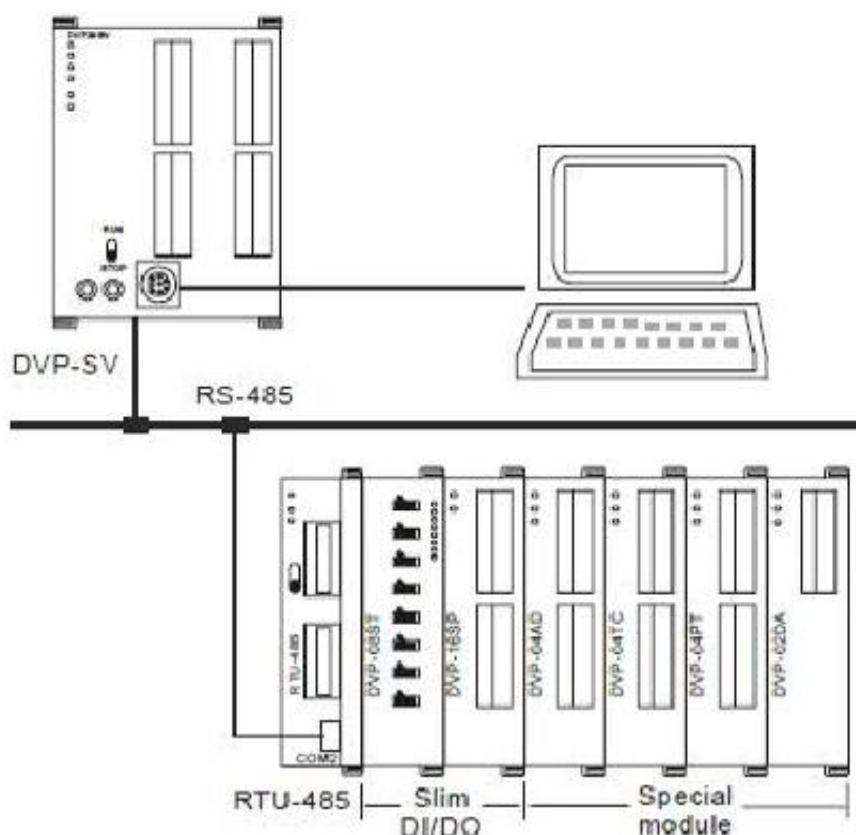
می‌باشند:

توضیحات	پورت COM3	پورت COM2	
از این فلگ‌ها برای نگهداشتن و حفظ کردن تنظیمات ارتباطات شبکه‌ی RS-485 استفاده می‌شود.	M1136	M1120	تنظیمات پروتکل
از این فلگ‌ها برای انتخاب مد ASCII/RTU در ارتباطات شبکه‌ی RS-485 استفاده می‌شود.	M1320	M1143	
از این رجیسترها برای تنظیم پروتکل ارتباطات شبکه‌ی RS-485 استفاده می‌شود.	D1109	D1120	
در این رجیسترها آدرس گره ارتباطات پورت COM تنظیم می‌شود.	D1255	D1121	درخواست ارسال داده
از این فلگ‌ها برای درخواست ارسال داده در ارتباطات شبکه‌ی RS-485 استفاده می‌شود.	M1316	M1122	
تنظیم زمان خطای Time-out ارتباطات شبکه برحسب ms برای پورت‌ها در این رجیستر تنظیم می‌شود.	D1252	D1129	
بعد از اینکه عمل دریافت داده از شبکه تکمیل شد، این فلگ‌ها فعال می‌شوند.	M1318	M1127	تکمیل شدن عمل دریافت داده
اگر خطایی در دریافت داده رخ دهد، این فلگ فعال خواهد شد.	M1319	-	خطاها

در این رجیستر کد خطای ارتباطات شبکه ثبت می‌شود.	D1253	-	
اگر خطای time-out در ارتباطات شبکه رخ دهد، این فلگ فعال (ON) خواهد شد.	-	M1129	
اگر خطای دریافت داده از COM2 (RS-485) MODRD, MODWR, MODRW رخ دهد، این فلگ فعال (ON) خواهد شد.	-	M1140	
اگر خطای پارامترهای دستورالعمل‌های MODRD, MODWR, MODRW به جز کد بوجود آمده در عمل دریافت داده که در رجیستر D1130 ذخیره می‌شود رخ دهد این فلگ فعال (ON) خواهد شد.	-	M1141	
کد خطایی که در پورت COM2 (RS-485) بوجود آمده و از ارتباطات شبکه‌ی Modbus بازگشته است، در این رجیستر ذخیره می‌شود.	-	D1130	

7-9 کاربرد ماژول RTU-485

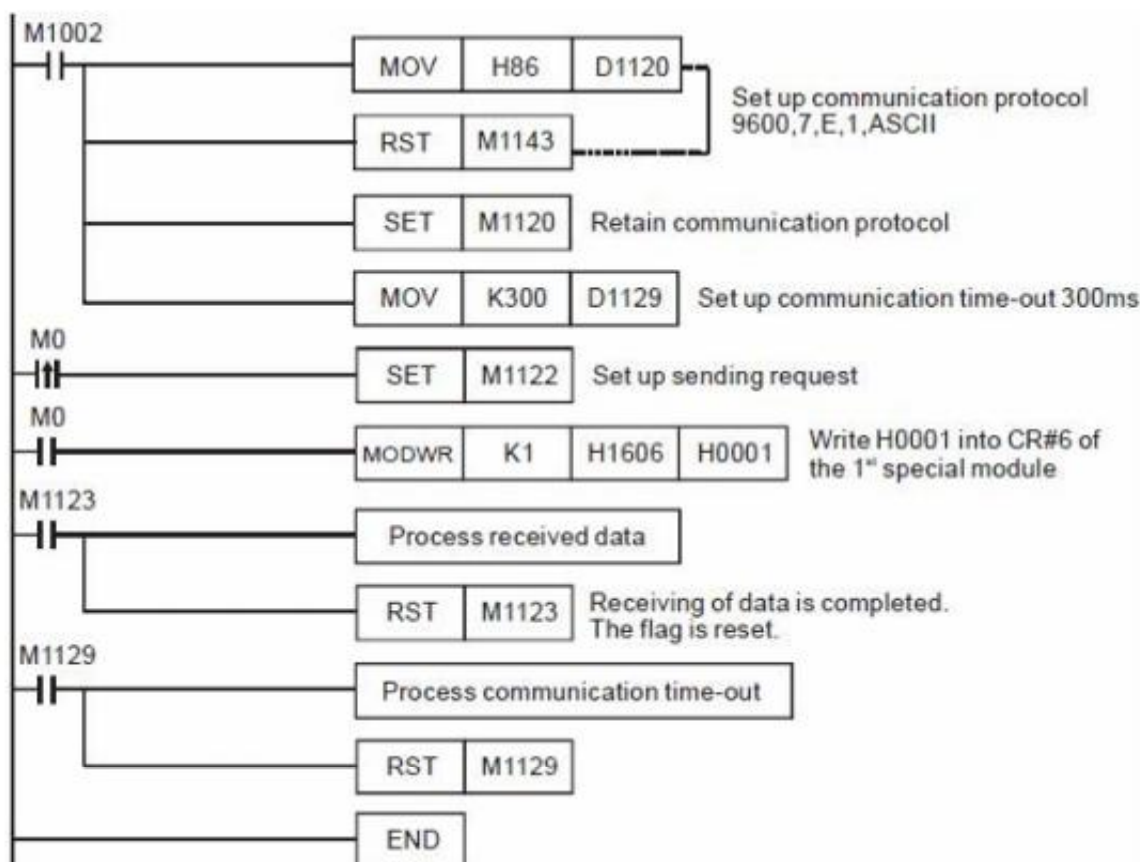
ماژول RTU-485 را می‌توان به دستگاه Master متصل کرد. مانند یک دستگاه Modbus slave استاندارد، ماژول RTU-485 سازگار با کنترلرهای برنامه‌پذیر منطقی (PLC) که از پروتکل مدباس پشتیبانی می‌کنند، می‌باشد. در اینجا مثالی آورده شده است که PLC سری DVP-SV در آن بعنوان PLC Master در نظر گرفته شده است. در این مثال، ما برنامه‌ی نردبانی نوشته شده در کامپیوتر را از طریق پورت ارتباطات RS-232 که پورت COM1 در PLC سری DVP-SV می‌باشد به PLC دانلود می‌کنیم. وقتی که در PLC سری DVP-SV برنامه‌ی نردبانی شروع به اجرا کرد، PLC فرامین مدباس را توسط ارتباطات RS-485 در پورت COM2 به ماژول کنترل از راه دور ورودی/خروجی‌ها (ماژول RTU-485) ارسال خواهد کرد. در شکل زیر، اتصال بین ماژول Slave که ماژول RTU-485 است را با دستگاه Master که PLC سری DVP-SV می‌باشد، نشان داده شده است:



حال می‌خواهیم برنامه‌ای مطابق با دیاگرام نردبانی برای PLC سری DVP-SV بنویسیم که میان PLC و ماژول RTU-485 تبادل داده کرده و برای کار با ورودی / خروجی‌های جدا از PLC از طریق کنترل از راه دور به کار می‌آید.

7-9-1 توسعه تعداد ورودی/خروجی توسط ماژول RTU-485

در این مثال، آدرس ایستگاه ماژول RTU-485 را برابر با 1 تنظیم کرده‌ایم. می‌خواهیم مقدار هگز H'0001 را در رجیستر کنترلی CR#6 نخستین ماژول توسعه‌ی خاص متصل شده به ماژول RTU-485 بنویسیم. در شکل زیر، برنامه‌ی نردبانی نوشته شده برای این مثال آورده شده است:



عملکرد برنامه‌ی این مثال به شرح زیر است :

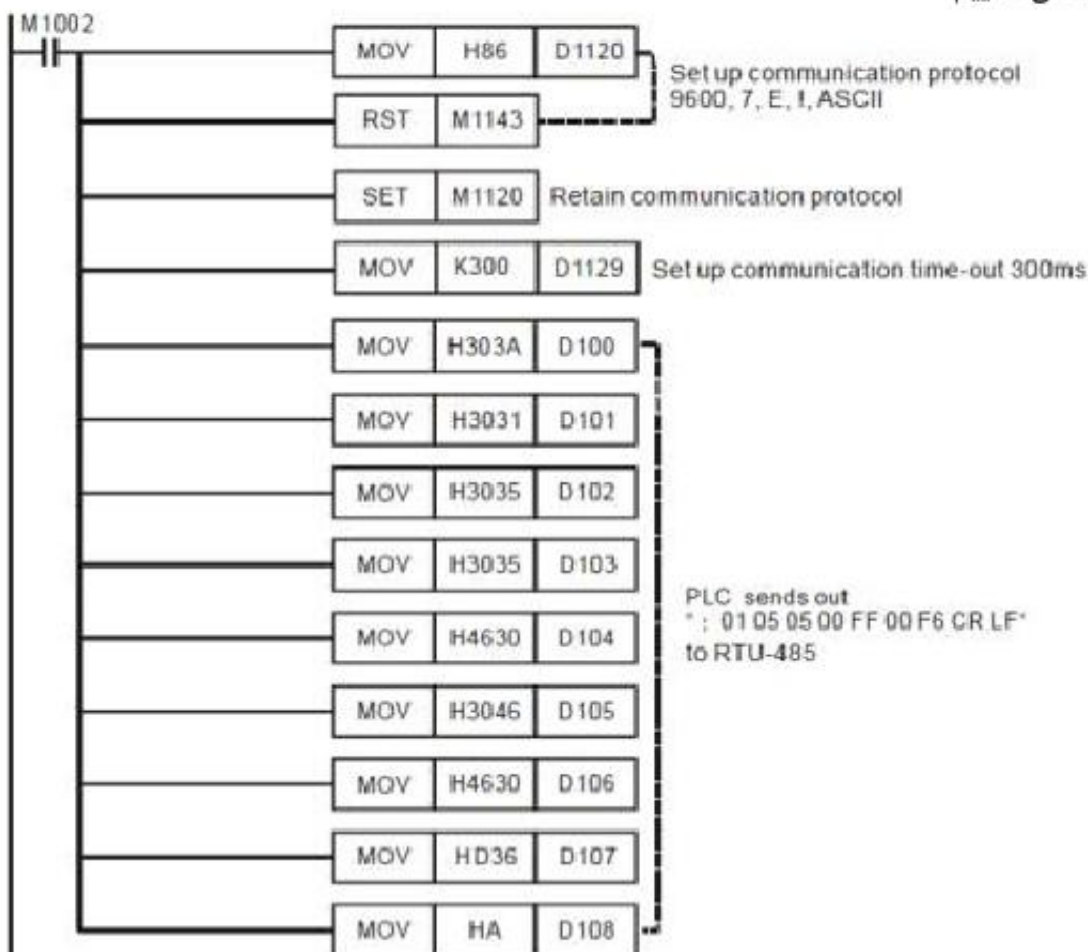
در اولین مرحله، فرمت ارتباطات شبکه‌ی مدباس تنظیم شده است. فرمت ارتباطات مورد استفاده برای دستگاه **Master** و **Slave** باید با یکدیگر یکسان باشند. برای مثال، فرمت ارتباطات تنظیم شده در سمت **PLC** و مازول **RTU-485** در این مثال برابر با **9600, 7, E, 1, ASCII** تنظیم شده است.

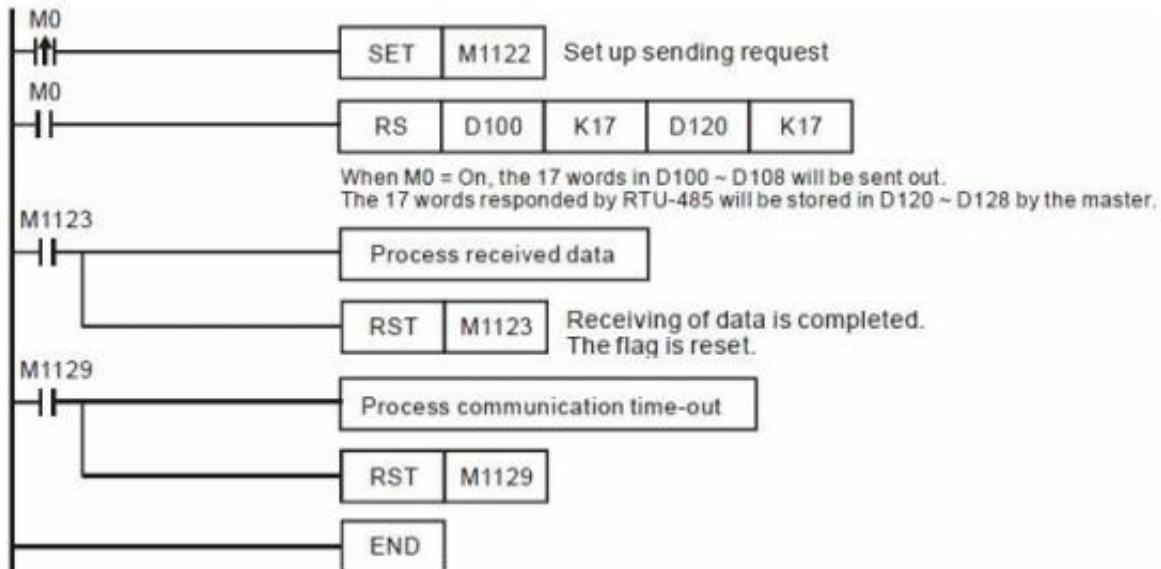
بعد از اینکه تنظیمات فرمت ارتباطات انجام شد، باید تنظیمات انجام شده برای ارتباطات **RS-485** در پورت **COM2** را با استفاده از فلگ **M1120** به صورت ماندگار حفظ نمایید (تنظیمات را به حافظه ماندگار لینک کنیم).

بعد از اینکه فلگ عمومی **M0** فعال (**ON**) شد، فلگ درخواست تنظیمات ارسال خواهد شد و دستگاه **Master** یک درخواست پیغام را به مازول **RTU-485** ارسال می‌کند و مقدار **H'0001** در داخل رجیستر کنترلی **CR#6** نخستین مازول توسعه‌ی خاص متصل شده به سمت راست مازول **RTU-485** نوشته خواهد شد.

7-9-2 کنترل خروجی دیجیتال Y0 متصل شده به ماژول RTU-485

در این مثال، آدرس ایستگاه ماژول RTU-485 را برابر با 1 تنظیم کرده‌ایم. می‌خواهیم خروجی Y0 از ماژول DO متصل شده به سمت راست ماژول RTU-485 را توسط برنامه‌نویسی در CPU مرکزی فعال نماییم.





عملکرد برنامه‌ی کنترلی این مثال به شرح زیر است :

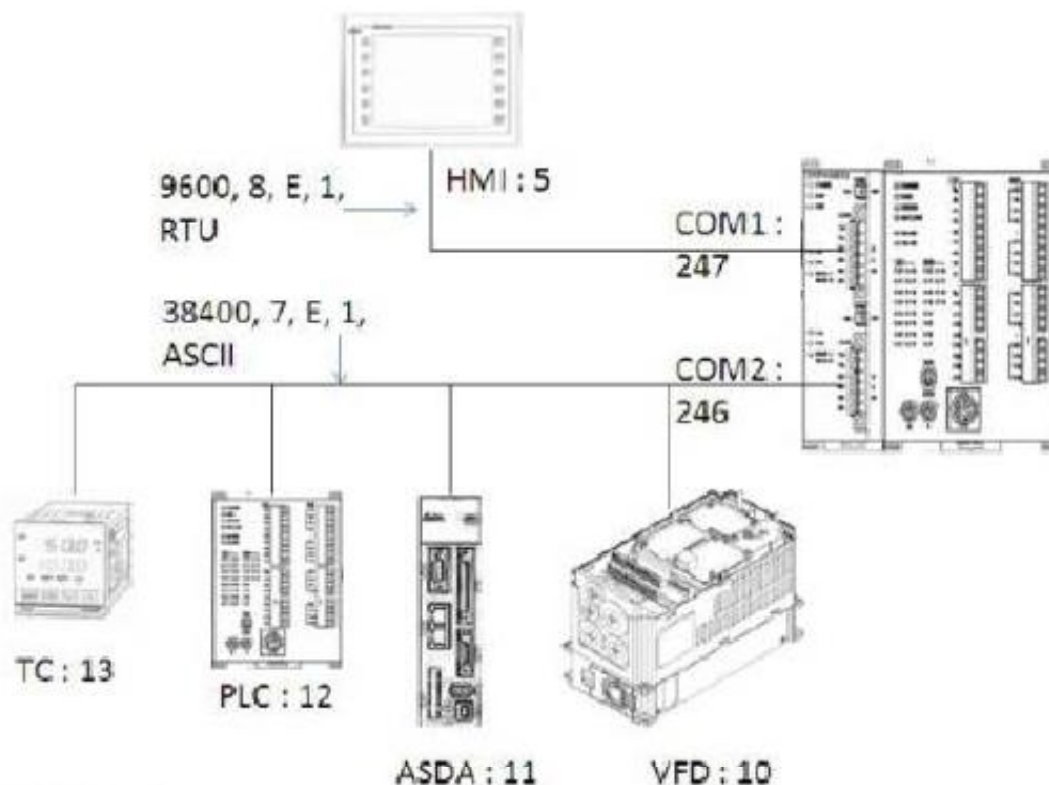
ابتدا باید فرمت ارتباطات شبکه را تنظیم کرد. این عمل در شروع برنامه اجرا خواهد شد. فرمت ارتباطات تنظیم شده برای دستگاه **Master** و دستگاه **Slave** باید با یکدیگر یکسان باشد. در این مثال، فرمت ارتباطات در هر دو سمت **PLC** و مازول **RTU-485** را برابر با 9600, 7, E, 1, **ASCII** تنظیم کرده‌ایم.

Master PLC بسته‌ی اطلاعاتی را به مازول **RTU-485** در مد **ASCII** ارسال می‌کند. شروع بسته‌ی اطلاعاتی از رجیستر **D100** است. بایت بالارزش این رجیستر دارای مقدار 30 بوده که در مد اسکی به معنی 0 می‌باشد و بایت کم‌ارزش رجیستر **D100** دارای مقدار 3A می‌باشد که در مد اسکی به معنی 1 است.

بعد از اینکه ابزار **M0=ON** شد، فلگ خاص **M1112** فعال خواهد شد. در این لحظه **Master PLC** تعداد 17 کلمه (Word) را توسط دستورالعمل **RS** ارسال می‌کند. این 17 کلمه شامل رجیسترهای **D100~D108** است. ابتدا داده‌ی موجود در کلمه‌های (Word) کم‌ارزش ارسال خواهد شد. **Master PLC** پیغام پاسخ ارسال شده توسط مازول **RTU-485** را در رجیسترهای **D120~D128** ذخیره خواهد کرد.

3-9-7 اتصال مازول SCM به شبکه‌ی MODBUS

در این قسمت می‌خواهیم چگونگی اتصال مازول **SCM** به دیگر تجهیزات اتوماسیون دلتا از قبیل واسط بین انسان و دستگاه **HMI**، پنل‌های متنی **TP**، اینورترها، سروسیستم‌ها و **PLC** ها را به شبکه‌ی **MODBUS** استاندارد بررسی نماییم. دیاگرام اتصال این مثال مطابق با شکل زیر است :

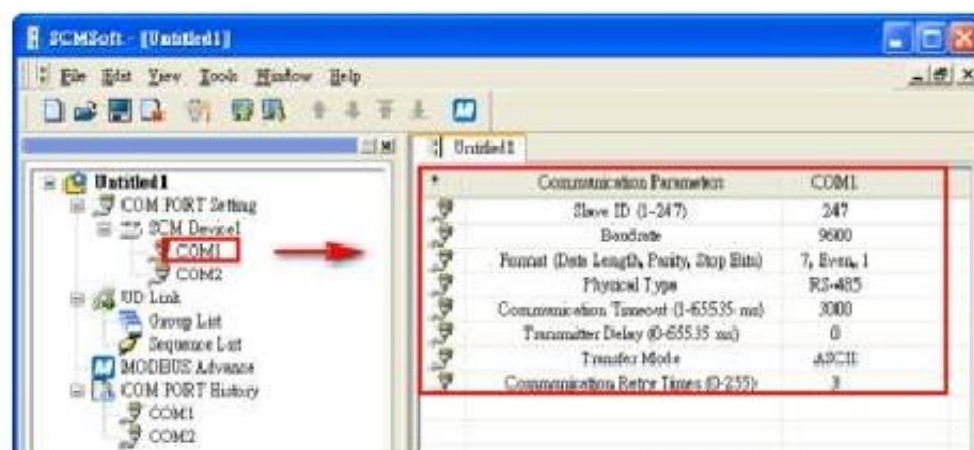
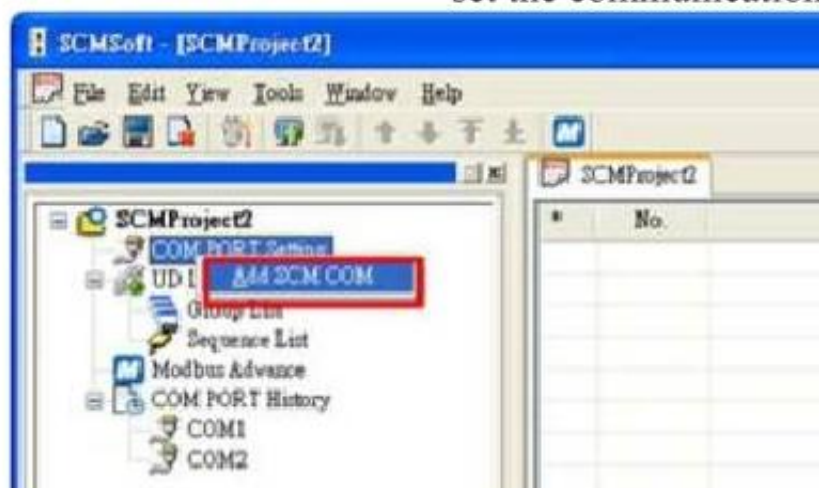


در جدول زیر، اطلاعات مربوط به هر دستگاه جهت برقراری ارتباطات در شبکه‌ی MODBUS در این مثال آورده شده است :

Product	Station Address	Communication Protocol	Address from which the data is read	Register in the MPU	Address into which the data is written	Register in the MPU
HMI	5	9600, RTU, 8, E, 1	-		-	
VFD	10	38400, ASCII, 7, E, 1	2103H	D100	2000H 2001H	D150-D151
ASDA	11	38400, ASCII, 7, E, 1	0101H 020AH	D200, D201	0101H 020AH	D250, D251
PLC	12	38400, ASCII, 7, E, 1	D100-D109	D300-D309	D200-D204	D350-D354
TC	13	38400, ASCII, 7, E, 1	1000H (PV)	D400	1001H (SV)	D451

حال می‌خواهیم ارتباط بین Modbus slave و محصولات دلتا را بررسی نماییم. برای ماژول SCM مانند Modbus slave باید فقط پارامترهایی از قبیل شماره‌ی ایستگاه ماژول SCM و سرعت تبادل داده قابل قبول برای اتصال به یک Master را تنظیم نمایید. ابتدا نرم‌افزار Soft را باز نموده و یک پروژه ایجاد کنید. سپس مسیر گفته شده در زیر را طی نمایید.

SCMSoft → New Project → COM PORT setting: Add SCM COM
→ set the communication parameters



پارامترهای ارتباطات پورت COM1 را به صورت زیر تنظیم نمایید:

Station address 247 (default), Modbus RTU, 9600, 8, Even, 1

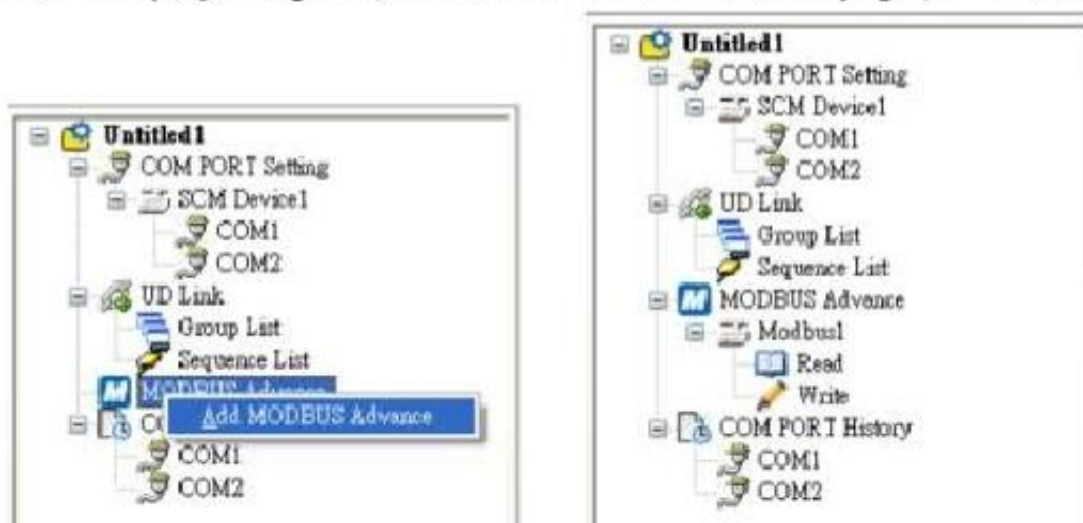
* Communication Parameters	COM1
Slave ID (1-247)	247
Baudrate	9600
Format (Data Length, Parity, Stop Bits)	8, Even, 1
Physical Type	RS-485
Communication Timeout (1-65535 ms)	3000
Transmitter Delay (0-65535 ms)	0
Transfer Mode	RTU
Communication Retry Times (0-255)	3

حال می‌خواهیم ارتباط بین Modbus Master و محصولات دلتا را بررسی نماییم.
مرحله‌ی 1 ← در این مرحله باید پارامترهای ارتباطات پورت COM2 را به صورت زیر تنظیم نماییم.

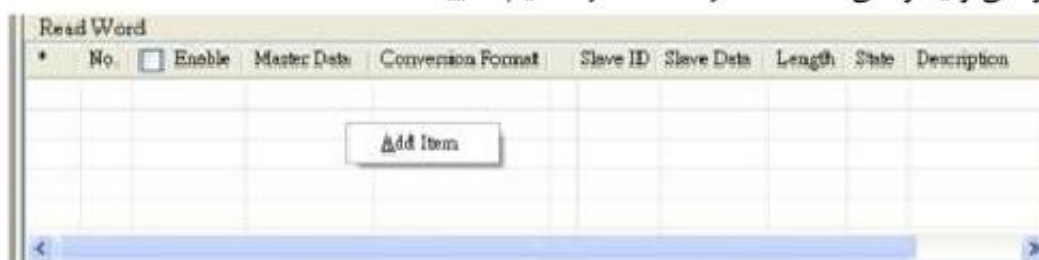
Station address 246 (default), Modbus ASCII, 38400, 7, Even, 1.

*	Communication Parameters	COM2
	Slave ID (1-247)	246
	Baudrate	38400
	Format (Data Length, Parity, Stop Bits)	7, Even, 1
	Physical Type	RS-485
	Communication Timeout (1-65535 ms)	3000
	Transmitter Delay (0-65535 ms)	0
	Transfer Mode	ASCII
	Communication Retry Times (0-255)	3

مرحله‌ی 2 ← در این مرحله باید یک Modbus Advance را مطابق با شکل زیر اضافه کنید.



مرحله‌ی 3 ← در این مرحله باید تبادل داده در Slave را تنظیم نمایید. در اینجا باید یک آتم را برای خواندن و یا نوشتن اطلاعات در Slave را تنظیم نمایید.



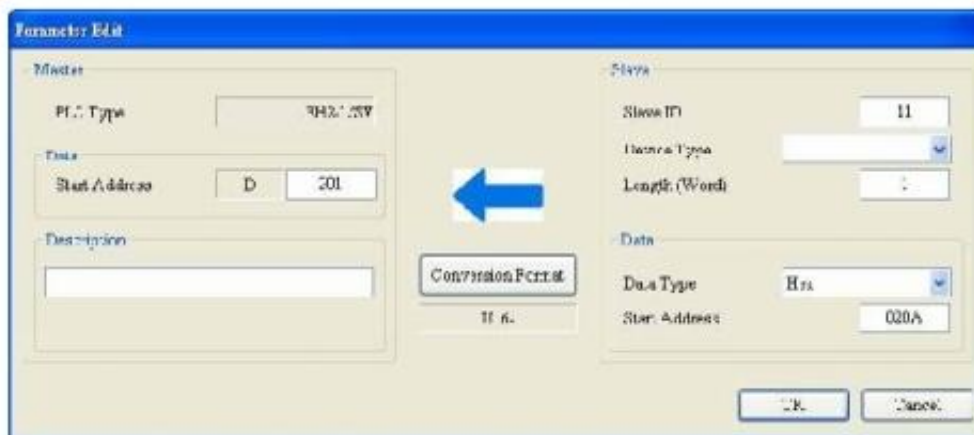
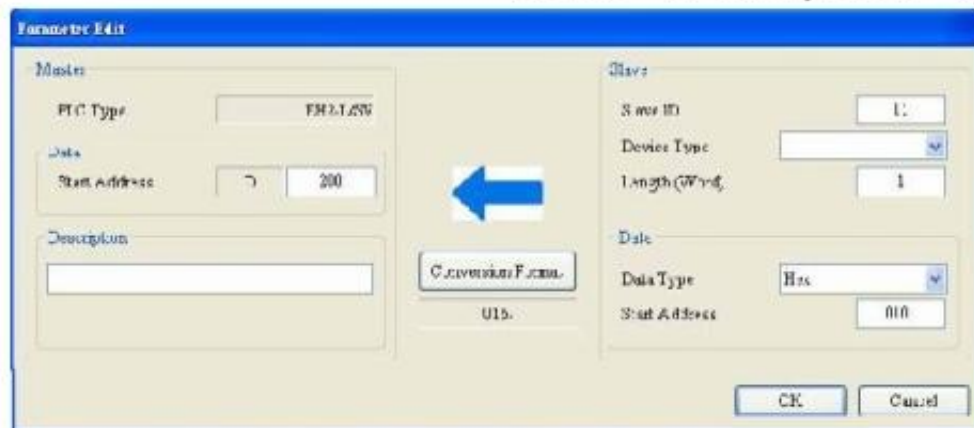
The screenshot shows the 'Parameter Edit' dialog box. On the left, the 'Master' section has 'PLC Type' set to 'EH2-L/SV', 'Data' type set to 'D', and 'Start Address' set to '0000'. On the right, the 'Slave' section has 'Slave ID' set to '1', 'Device Type' as a dropdown, 'Length (Word)' set to '1', 'Data Type' set to 'Hex', and 'Start Address' set to '0000'. A blue arrow points from the Slave 'Start Address' field to the Master 'Start Address' field. In the center, there is a 'Conversion Format' section with 'U16-' selected. At the bottom right are 'OK' and 'Cancel' buttons.

در این مثال، ما اینورتر سری VFD را بعنوان Slave در نظر گرفته ایم. می خواهیم مقدار پارامتر H'2103 اینورتر را خوانده و در رجیستر D100 در سمت PLC ذخیره نماییم. (H'2103 ← D100)، همچنین، می خواهیم مقادیر موجود در رجیسترهای D150, D151 در سمت PLC را در پارامترهای H2000, H2001 اینورتر بنویسیم (D150, D151 → H'2000, H'2001).

This screenshot shows the 'Parameter Edit' dialog box with updated values. The 'Master' section has 'Start Address' set to 'D 100'. The 'Slave' section has 'Start Address' set to '2103'. The 'Conversion Format' remains 'U16-'. A blue arrow points from the Slave 'Start Address' field to the Master 'Start Address' field.

This screenshot shows the 'Parameter Edit' dialog box with further updated values. The 'Master' section has 'Start Address' set to 'D 150'. The 'Slave' section has 'Start Address' set to '2000'. The 'Conversion Format' remains 'U16-'. A blue arrow points from the Slave 'Start Address' field to the Master 'Start Address' field.

در اینجا ما سرودرایو سری ASDA را نیز بعنوان Slave در نظر گرفته‌ایم. می‌خواهیم مقدار پارامتر H'0101 سرودرایو را خوانده و در رجیستر D200 در سمت PLC ذخیره نماییم. همچنین، مقدار پارامتر H'020A سرودرایو را خوانده و در رجیستر D201 در سمت PLC ذخیره کنیم.
(D200 ← H'0101 و D201 ← H'020A).



همچنین، می‌خواهیم مقدار موجود در رجیستر D50 در سمت PLC را در پارامتر H'0101 سرودرایو و مقدار موجود در رجیستر D251 در سمت PLC را در پارامتر H'020A سرودرایو بنویسیم.

Parameter Edit

Master

PLC Type: EH2 LSV

Data

Start Address: D 250

Description:

Slave

Slave ID: 0

Data Type: Hex

Length (Word): 1

Start Address: 300

OK Cancel

Parameter Edit

Master

PLC Type: EH2 LSV

Data

Start Address: D 250

Description:

Slave

Slave ID: 1

Data Type: Hex

Length (Word): 1

Start Address: 300A

OK Cancel

در این مثال، از دو PLC استفاده شده است که یکی Master PLC بوده و به آن مازول SCM متصل شده است و دیگری Slave PLC می‌باشد. می‌خواهیم مقادیر موجود در رجیسترهای D100~D109 در سمت Slave PLC را خوانده و در رجیسترهای D300~309 در سمت Master PLC ذخیره نماییم.

(D300~D309 in the Master PLC ← D100~D109 in the slave PLC)

همچنین، می‌خواهیم مقدار موجود در رجیسترهای D350~D354 در سمت Master PLC را در رجیسترهای D150~D154 در سمت slave PLC بنویسیم.

(D350~D354 in the Master PLC ← D150~D154 in the slave PLC)

Parameter Edit

Master:

PLC Type: EH2-L57

Data:

Start Address: D 200

Description:

Slave:

Slave ID: 12

Device Type: RA25P2

Length (Word): 10

Data:

Data Type: I

Start Address: I000

Conversion Format: U6

OK Cancel

Parameter Edit

Master:

PLC Type: EH2-L57

Data:

Start Address: D 500

Description:

Slave:

Slave ID: 12

Device Type: RA25P2

Length (Word): 1

Data:

Data Type: D

Start Address: D150

OK Cancel

در این مثال، ما کنترل دمای TC دلتا را بعنوان Slave به ماژول SCM متصل کرده‌ایم و می‌خواهیم مقدار موجود در پارامتر H'1000 کنترلر دما را در رجیستر D400 در سمت PLC ذخیره نماییم (D400 ← H'1000). همچنین، می‌خواهیم مقدار موجود در رجیستر D451 در سمت Master PLC را در پارامتر H'1001 کنترلر دمای دلتا بنویسیم. (D451 → H'1001).

Parameter Edit

Master:

PLC Type: EH2-L57

Data:

Start Address: D 400

Description:

Slave:

Slave ID: 13

Device Type: RA25P2

Length (Word): 1

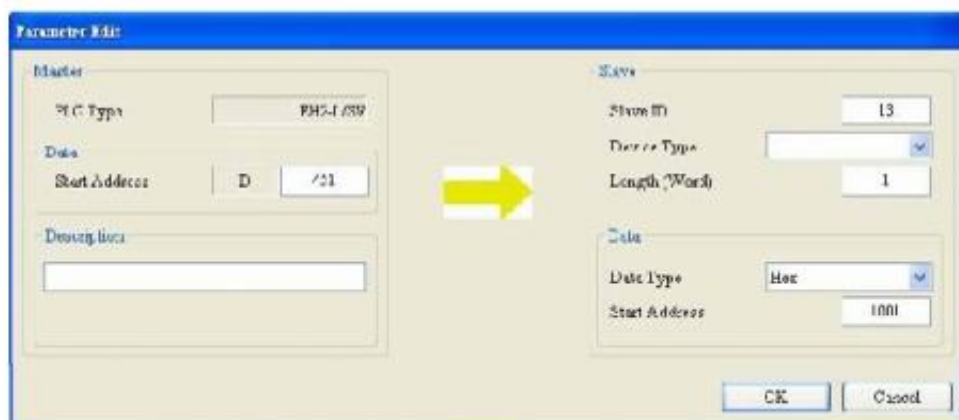
Data:

Data Type: Hex

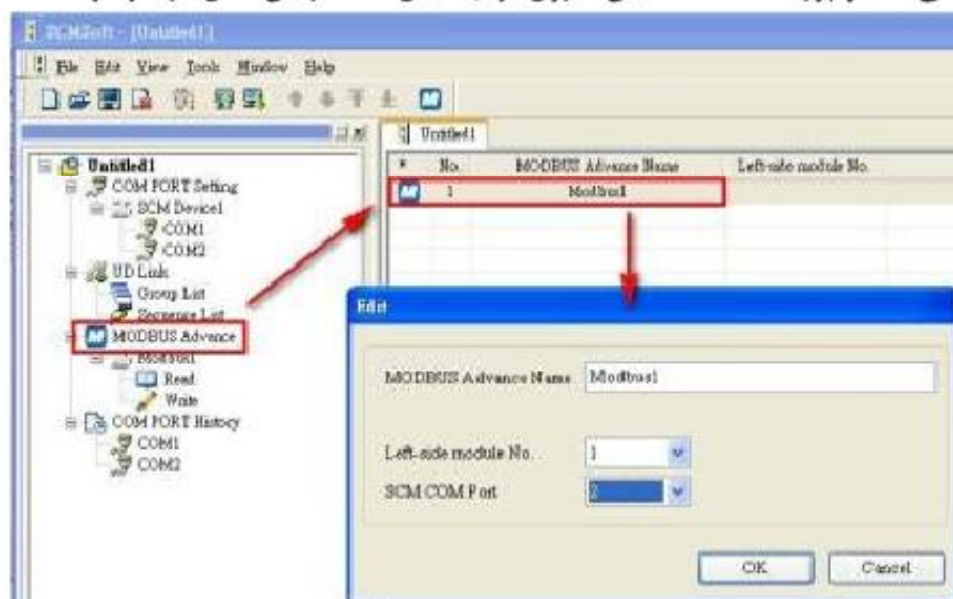
Start Address: I000

Conversion Format: U12

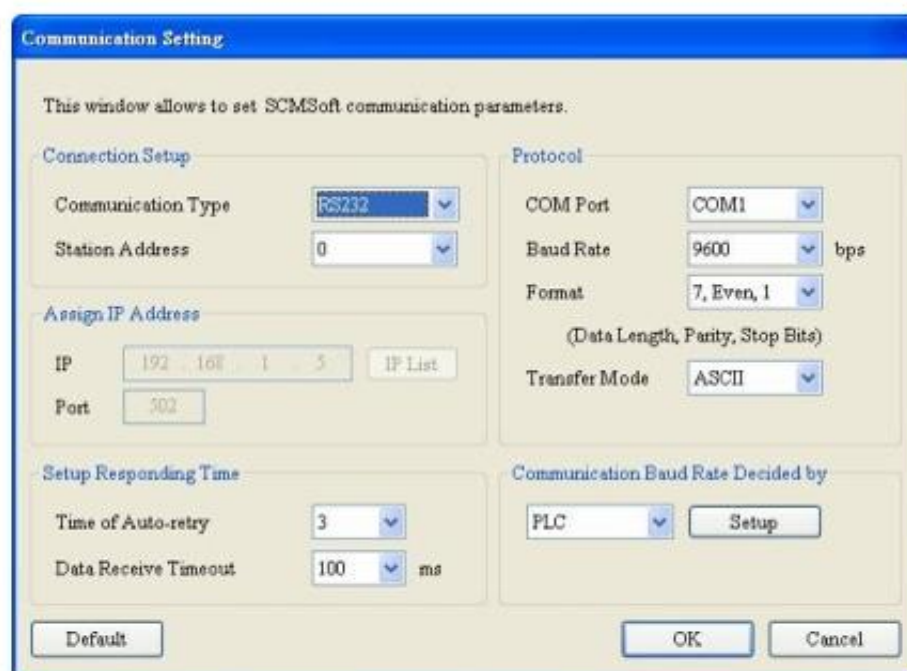
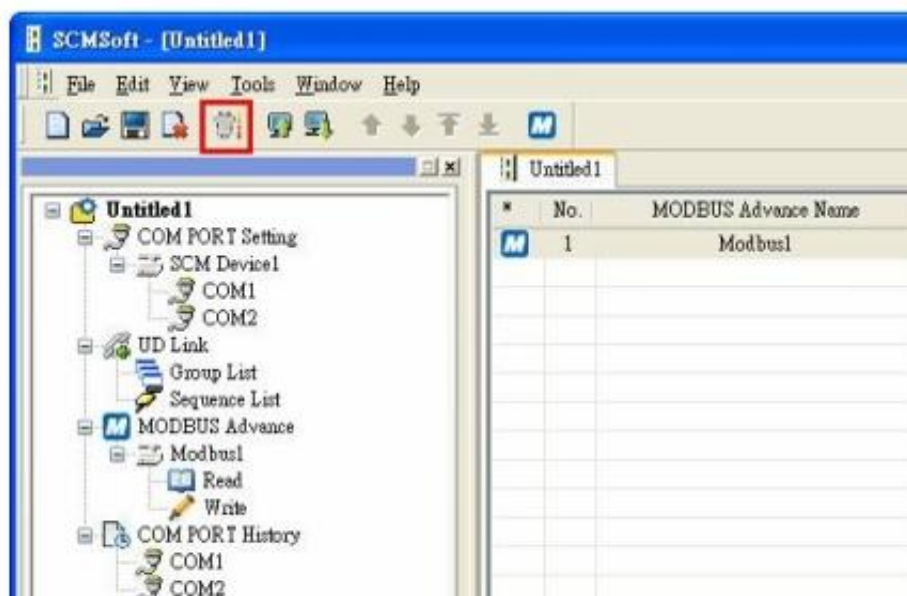
OK Cancel



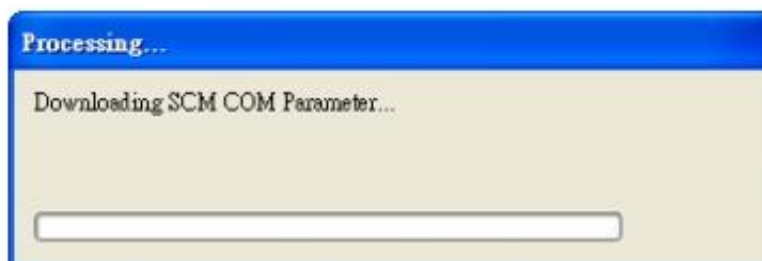
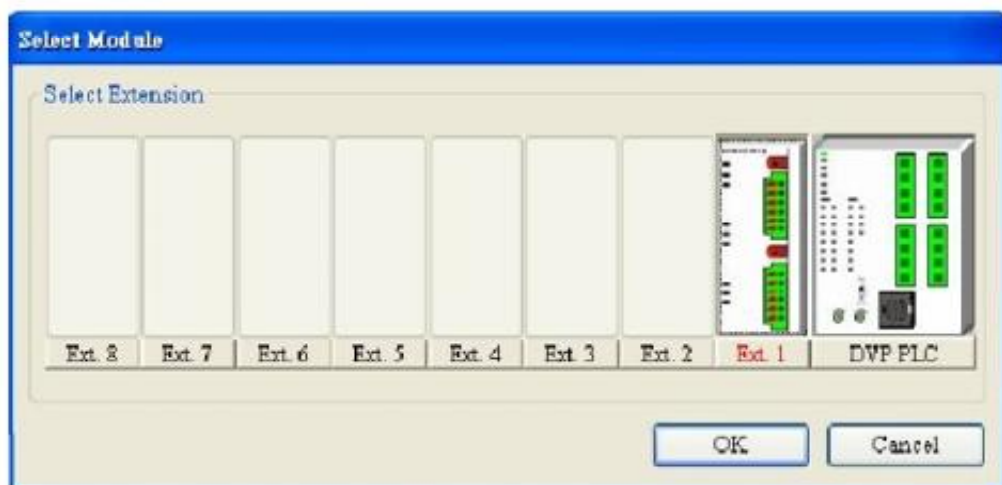
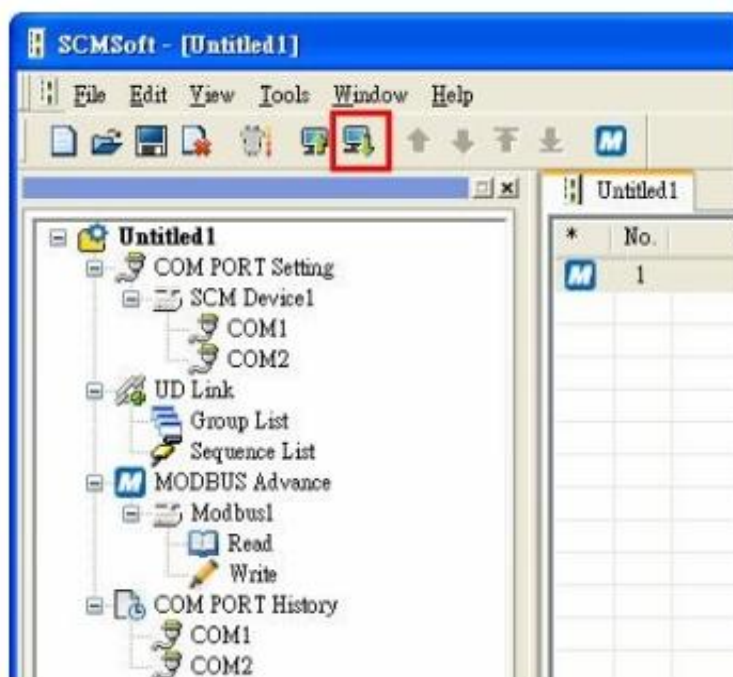
بعد از اینکه تنظیمات را تکمیل کردید، می‌توانید پورت ارتباطات مورد استفاده را برای Modbus Advance در ماژول SCM تعیین کنید. ماژول SCM نخستین ماژول متصل شده به سمت چپ PLC دلتا می‌باشد و پورت COM2 این ماژول برای تبادل داده در این مثال در نظر گرفته شده است.

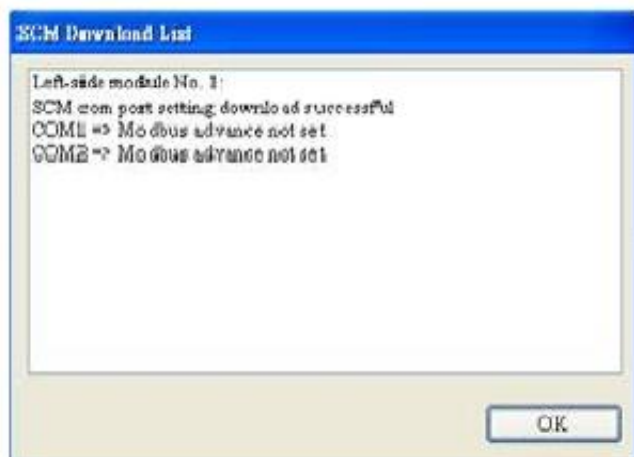


مرحله‌ی 4 ← در این مرحله باید تنظیمات انجام شده برای پارامترها را به ماژول SCM دانلود نمایید. شما می‌توانید پارامترهای ارتباطات را تنظیم نمایید. بعد از اینکه تنظیمات تکمیل شد، بر روی دکمه‌ی OK برای خارج شدن از تنظیمات ارتباطات کلیک کرده و پارامترهای ماژول را تنظیم نمایید.



بر روی آیکن **Download** کلیک نمایید. ماژول سمت چپ را جهت دانلود کردن تنظیمات انتخاب کرده و بر روی دکمه‌ی **OK** کلیک نمایید. اگر فقط یک ماژول (ماژول SCM) به سمت چپ PLC متصل شده بود، مستقیماً بر روی دکمه‌ی **OK** کلیک نمایید.



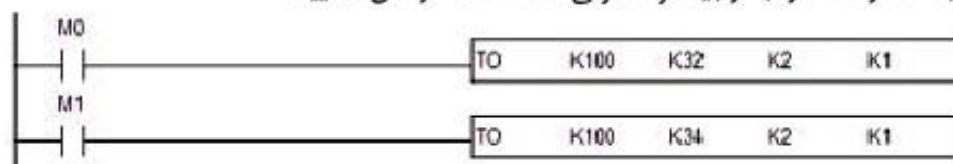


مرحله‌ی 5 ← در این مرحله باید تابع تبادل داده را فعال کنیم. تبادل داده را باید توسط دستورالعمل TO در نرم‌افزار WPLSoft جهت خواندن بیت‌ها / خواندن کلمه‌ها / نوشتن بیت‌ها / نوشتن کلمه‌ها (CR#31~CR#34) کنترل کرد.

#CR	خاصیت	نام رجیستر کنترلی	توضیحات
31	R/W	راه‌اندازی تبادل داده بین پورت COM1 جهت خواندن بیت‌ها یا Wordها	High byte: bit ; low byte: Word اگر مقدار این رجیستر برابر با 0 تنظیم شود، این تابع راه‌اندازی نخواهد شد. اگر مقدار این بیت برابر با 1 تنظیم شود، این تابع فقط برای یکبار راه‌اندازی می‌شود. اگر مقدار این رجیستر برابر با 2 تنظیم شود، این تابع برای همیشه راه‌اندازی خواهد شد.
32	R/W	راه‌اندازی تبادل داده بین پورت COM2 جهت خواندن بیت‌ها یا Wordها	High byte: bit ; low byte: Word اگر مقدار این رجیستر برابر با 0 تنظیم شود، این تابع راه‌اندازی نخواهد شد. اگر مقدار این بیت برابر با 1 تنظیم شود، این تابع فقط برای یکبار راه‌اندازی می‌شود. اگر مقدار این رجیستر برابر با 2 تنظیم شود، این تابع برای همیشه راه‌اندازی خواهد شد.
33	R/W	راه‌اندازی تبادل داده بین پورت COM1 جهت نوشتن بیت‌ها یا Wordها	High byte: bit ; low byte: Word اگر مقدار این رجیستر برابر با 0 تنظیم شود، این تابع راه‌اندازی نخواهد شد. اگر مقدار این بیت برابر با 1 تنظیم شود، این تابع فقط برای یکبار راه‌اندازی می‌شود. اگر مقدار این رجیستر برابر با 2 تنظیم شود، این تابع برای همیشه راه‌اندازی خواهد شد.
34	R/W	راه‌اندازی تبادل داده بین پورت COM2 جهت نوشتن بیت‌ها یا Wordها	High byte: bit ; low byte: Word اگر مقدار این رجیستر برابر با 0 تنظیم شود، این تابع راه‌اندازی نخواهد شد. اگر مقدار این بیت برابر با 1 تنظیم شود، این تابع فقط برای یکبار راه‌اندازی می‌شود. اگر مقدار این رجیستر برابر با 2 تنظیم شود، این تابع برای همیشه راه‌اندازی خواهد شد.

اگر می‌خواهید اجرای عمل خواندن کلمه (Word) را به طور متوالی و پیوسته ادامه دهید، باید مقدار K2 را به رجیستر کنترلی CR#32 ارسال نمایید. اگر می‌خواهید عمل خواندن Wordها را فقط برای یکبار اجرا نمایید، باید مقدار K1 را به رجیستر کنترلی CR#32 ارسال نمایید.

اگر می‌خواهید عمل نوشتن در Word را به صورت متوالی و دائماً اجرا کنید، باید مقدار K2 را به رجیستر کنترلی CR#34 ارسال نمایید. و اگر می‌خواهید عمل نوشتن Wordها را فقط برای یکبار اجرا کنید، باید مقدار K1 را به رجیستر کنترلی CR#34 ارسال نمایید.

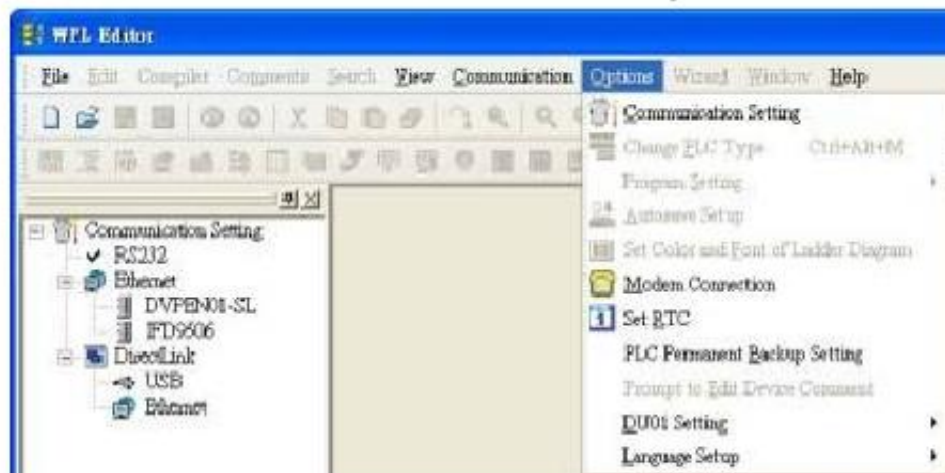


بعد از اینکه M0 راه‌اندازی شد، داده از آدرس Slave ی که برای پورت COM2 در ماژول SCM تنظیم شده است، خوانده خواهد شد. بعد از اینکه فلگ عمومی M1 راه‌اندازی شد، داده در آدرس Slave ی که برای پورت COM2 در ماژول SCM تنظیم شده است، نوشته خواهد شد.

9-4 اتصال ماژول SCM به نرم‌افزار WPL Soft

از ماژول SCM می‌توان بعنوان پورت ارتباطات اضافی در سمت PLC Master استفاده کرد. وقتی که پورت ارتباطات RS-485 در PLC Master در حال اجرا است، شما می‌توانید با استفاده از نرم‌افزار WPL Soft دستگاه Master را توسط ماژول SCM مانیتور نمایید. فرمت ارتباطات پیش‌فرض پورت COM1 در ماژول SCM برابر با 9600, 7, Even, 1, Station address: 247 می‌باشد. برای برقراری ارتباط با ماژول SCM در نرم‌افزار WPL Soft مراحل زیر را طی نمایید:

مرحله‌ی 1 ← نرم افزار WPL Soft را باز کرده، از منوی Option گزینه‌ی Communication Setting را انتخاب نمایید.



مرحله‌ی 2 ← نوع ارتباط را در تنظیمات ارتباطات برابر با RS-232 انتخاب نمایید. شماره‌ی پورت ارتباط در سمت کامپیوتر را در کادر COM PORT تعیین کرده و پارامترهای ارتباطات را وارد نمایید. پارامترهای ارتباطات را باید برابر با تنظیمات پیش‌فرض پورت COM1 ماژول SCM تنظیم نمایید. اگر دیگر پارامترهای ارتباطات مورد نیاز هستند، آنها را نیز باید در تنظیمات COM PORT ماژول SCM اصلاح نمایید. علاوه، تنظیمات گزینه Station Address را برای پورت COM1 ماژول SCM باید مطابق با Station address در سمت PLC تنظیم نمایید.

مرحله‌ی 3 ← برای دانلود/ آپلود برنامه‌ی FROM/TO توسط نرم‌افزار WPLSoft به PLC بر روی OK کلیک نمایید.

9-5- اتصال ماژول SCM به شبکه‌ی ارتباطات RS-485 (مدباس غیر استاندارد)

در این قسمت، می‌خواهیم چگونگی اتصال ماژول SCM به دیگر تولیدات اتوماسیون صنعتی دلتا توسط پورت ارتباطات RS-485 (MODBUS غیر استاندارد) را بررسی نماییم.

در اینجا می‌خواهیم اتصال به متر برقی را بررسی نماییم. دو روش عمومی برای اتصال به یک متر الکتریکی وجود دارد. یکی توسط شبکه‌ی ارتباطات MODBUS استاندارد بوده و دیگری توسط ارتباطات RS-485 می‌باشد. در این بخش، شما با روش چگونگی اتصال ماژول SCM به یک متر الکتریکی توسط RS-485 در UD Link آشنا خواهید شد. مراحل کار به شرح زیر می‌باشد:

مرحله‌ی 1 ← نوع ثبت کردن ← ابتدا باید آدرس ایستگاه ارتباطات متر الکتریکی تحت شبکه‌ی RS-485 را برابر با 5 تنظیم نمایید. (این آدرس اختیاری بوده و باید در کل شبکه منحصر بفرد باشد). متر الکتریکی شامل انواع ثبت کننده‌های اطلاعات می‌باشد که عبارتند از: ثبت کننده‌ی اطلاعات به صورت مختصر (Abbreviated record)، ثبت کننده‌ی اطلاعات به صورت کنترل شده (Control record) و ثبت کننده‌ی اطلاعات به صورت کامل (Full record).

در زیر، جدول انواع ثبت کنندگان اطلاعات در متر الکتریکی آورده شده است:

(Abbreviated)		
Word number	Content	Description
1	10h	Start bit
2	0 ... FAh, FFh	Device address (IA)
3		Function code (FF)
4		(IA+FF) Checksum (CS)
5	16h	End marker

(Control)		
Word number	Content	Description
1	68h	Start bit
2	03h	Length
3	03h	Length (repeat)
4	68h	Start bit (repeat)
5	0 ... FAh, FFh	Device address (IA)
6		Function code (FF)
7		Parameter index (PI)
8		Checksum (CS) (Add from IA to PI.)
9	16h	End marker

(Full)		
Word number	Content	Description
1	68h	Start bit
2		Length
3		Length (repeat)
4	68h	Start bit (repeat)

5	0 ... FAh, FFh	Device address (IA)
6		Function code (FF)
7		Parameter index (PI)
...		n Word, data block
Length+5		Checksum (CS) (Add from IA to the previous item.)
Length+6	16h	End marker

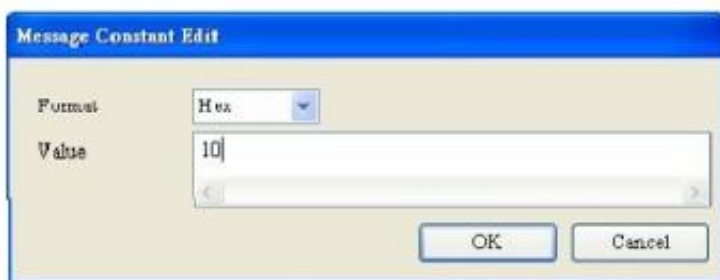
مرحله‌ی 2 ← کاربرد ← 9 نوع کاربرد در مازول ارتباطات SCM با متر الکتریکی وجود دارد که این 9 نوع استفاده را می‌توان توسط ترکیب 3 نوع ثبت کننده‌ی اطلاعات بوجود آورد.

Type	Instruction to the electricity meter	Response (through the record type)
1	Reset Abbreviated record	N/A
2	Query about the status of the device: abbreviated record	Abbreviated record
3	Measured value and error (cyclic data) Abbreviated record	Full record
4	Event data analyzed erroneously Abbreviated record	Full record
5	Measured value Control record	Full record
6	Output parameter: control record	Full record
7	Status: control record	Full record
8	Device specifications: control record	Full record
9	Real-time timing data:	Full record

مرحله‌ی 3 ← ویرایش پروتکل UD Link ← همانطور که گفته شد 9 نوع کاربرد برای متر الکتریکی متصل شده به مازول SCM وجود دارد که ما در اینجا به بررسی برخی از آنها می‌پردازیم.
نوع 1: در این نوع فقط مقدار ثبت کننده‌ی اطلاعات به صورت مختصر (Abbreviated record) به مازول SCM ارسال خواهد شد. فرمت ثبت کننده‌ی اطلاعات به صورت مختصر به شرح زیر می‌باشد.

Start Word + device address (IA) + Function code (FF) + Checksum (CS)
+ End marker → 10h + D0 + 09h + (IA+FF) + 16h

- Word شروع برابر با 10h می‌باشد.



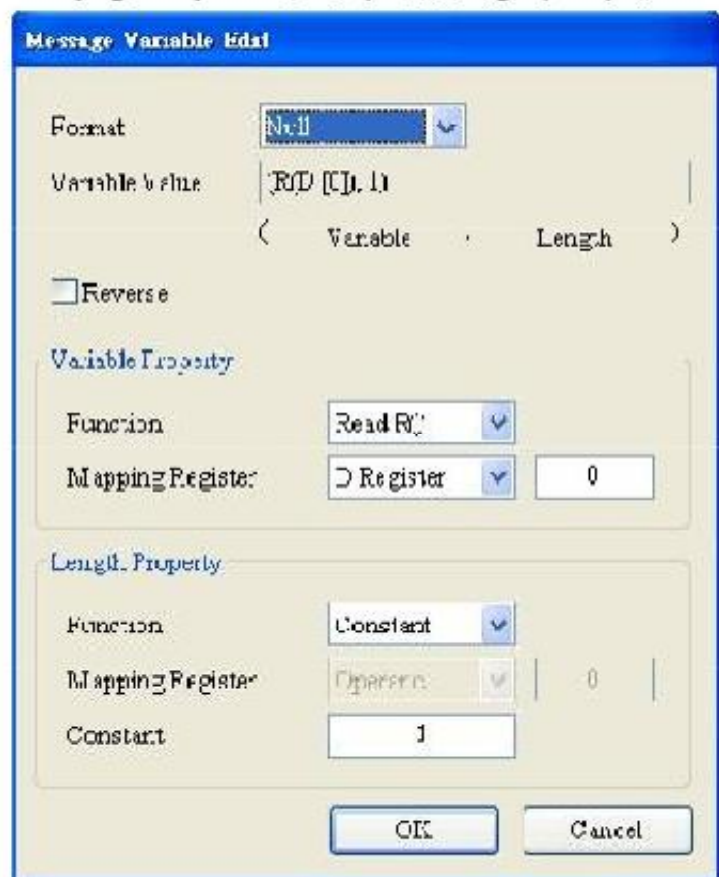
Message Constant Edit

Format: Hex

Value: 10

OK Cancel

- در اینجا آدرس دستگاه (متر الکتریکی) از رجیستر D0(IA) خوانده می‌شود.



Message Variable Edit

Format: Null

Variable Value: RD [I], 1

< Variable Length >

☐ Reverse

Variable Property

Function: Read RC

Mapping Register: Register 0

Length Property

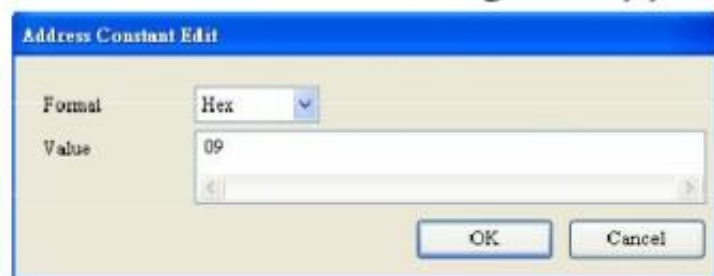
Function: Constant

Mapping Register: Discrete 0

Constant: 1

OK Cancel

- کد تابع FF در اینجا برابر با 09h می‌باشد.




Address Constant Edit

Format: Hex

Value: 09

OK Cancel

- مقدار Checksum برابر با 1byte برای اضافه کردن دو آیتم قبلی به Checksum می‌باشد:



Checksum Edit

Class: SUM (1Byte)

Format: Hex

Initial Value: 0

☐ Reverse

OK Cancel



Checksum

Add No. 2 ~ No. 3

- Word پایان یا همان علامت پایان در اینجا برای خط کش (متر) الکتریکی 16h می‌باشد.



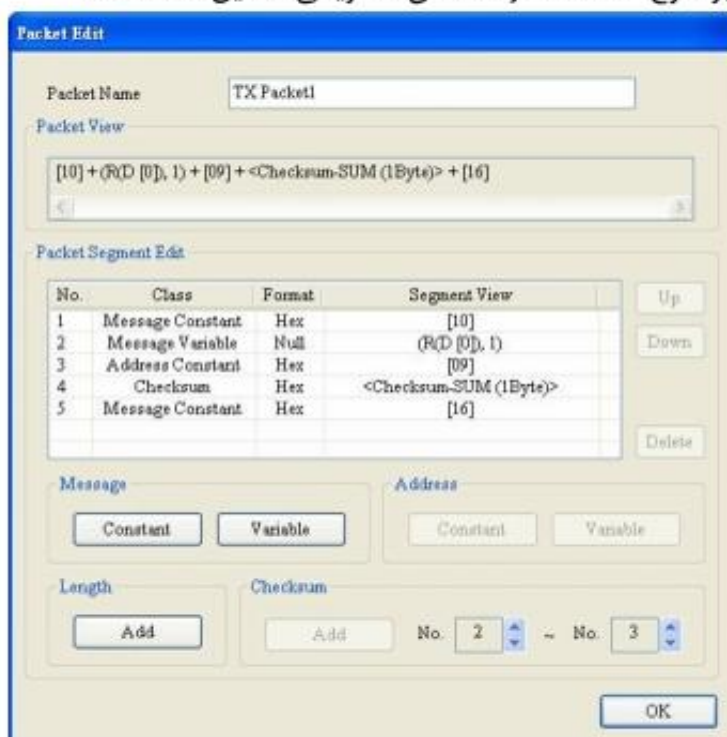
Address Constant Edit

Format: Hex

Value: 16

OK Cancel

- حال ویرایش کاربرد نوع 1 استفاده از خط کش الکتریکی تکمیل شده است.



Packet Edit

Packet Name: TX Packet1

Packet View: [10] + (R(D [0], 1) + [09] + <Checksum-SUM (1Byte)> + [16]

Packet Segment Edit

No.	Class	Format	Segment View
1	Message Constant	Hex	[10]
2	Message Variable	Null	(R(D [0], 1)
3	Address Constant	Hex	[09]
4	Checksum	Hex	<Checksum-SUM (1Byte)>
5	Message Constant	Hex	[16]

Up Down Delete

Message: Constant Variable Address: Constant Variable

Length: Add Checksum: Add No. 2 ~ No. 3

OK

آدرس پاسخ برای کاربرد نوع 1 وجود ندارد، بنابراین، نیازی به ویرایش کد تابع پاسخ (RX) نیست. باید فرمان مخابراتی اطلاعات را برای کاربرد نوع 1 خط کش الکتریکی ویرایش نمایید. از این رو، ما ارسال بسته‌ی 1 از TX را انتخاب می‌نماییم. آدرس پاسخ در این نوع وجود ندارد.

Command Edit

Command No. 1

Command Type Send

Send Packet TX Packet1

Recv Packet

Success End

Fail Abort

Retry 0 (0 - 255)

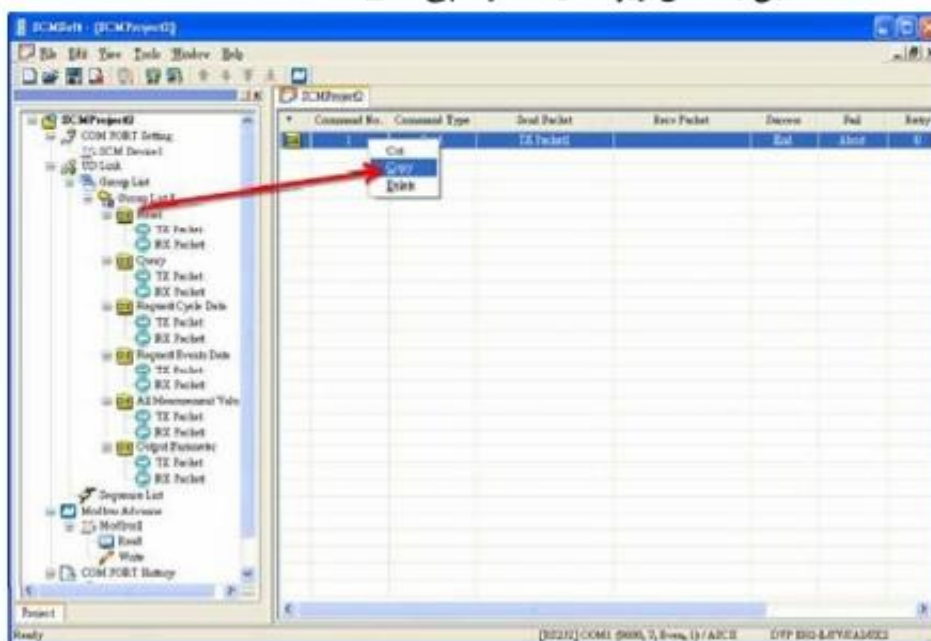
Repeat 0 (0 - 255)

Send Wait 0 (0 - 65535 ms)

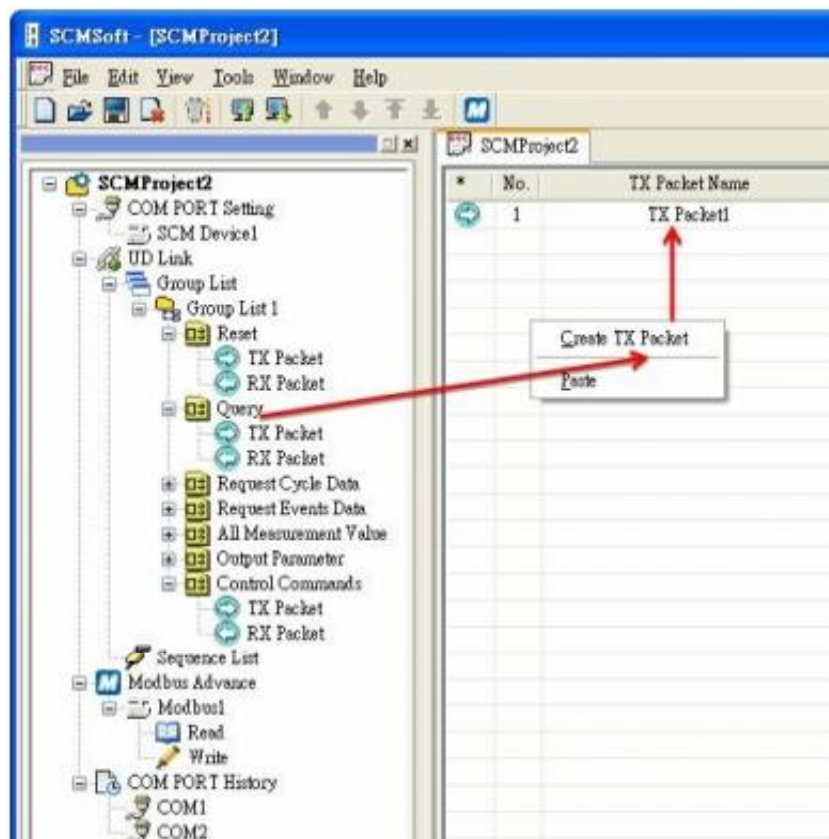
Timeout 50 (0 - 65535 ms)

OK Cancel

نوع 2: در این نوع اطلاعات ثبت شده به صورت مختصر ارسال شده و به ثبت کننده‌ی مختصر اطلاعات پاسخ داده می‌شود. تنظیمات ارسال داده مانند تنظیمات کاربرد نوع 1 می‌باشد. شما می‌توانید تنظیمات را مستقیماً در اینجا کپی نمایید. توجه داشته باشید که تابع در این نوع 29h شده است. - در گروه Reset مطابق با شکل زیر تنظیمات را کپی نمایید.



- تنظیمات کپی شده را در بسته‌ی Tx در گروه Query جهت اجرا تنظیمات بسته‌ی Tx بچسبانید (Paste کنید).



فرمت پاسخ به اطلاعات ارسال شده توسط ثبت کننده‌ی اطلاعات به صورت مختصر مطابق با فرمت زیر می‌باشد:

Start Word + Device address (IA) + Function code (FF) + Checksum (CS) + End marker → 10h + D0 + 09h + (IA+FF) + 16h

- مقدار Word شروع دریافت پاسخ توسط خط‌کش الکتریکی برابر با 10h می‌باشد.



- آدرس پاسخ را بررسی کنید. آدرس دستگاه که از قبل از رجیستر D0(IA) خوانده شده است، باید با آدرس پاسخ برابر باشد.

Message Variable Edit

Format: Null

Variable Value: (R(D [0], 1)
 (Variable * Length)

☐ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 0

Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

OK Cancel

- از کد تابع FF در پاسخ صرفنظر خواهد شد (1, *). از Wordی که طول آن 1 باشد، صرفنظر خواهد شد. اگر قصد دارید که کد تابع را ذخیره نمایید، می‌توانید با مراجعه به تنظیمات آدرس دستگاه (IA) کد تابع را در رجیستر D ذخیره نمایید.

Message Variable Edit

Format: Null

Variable Value: (* , 1)
 (Variable * Length)

☐ Reverse

Variable Property

Function: *

Mapping Register: Operand 0

Length Property

Function: Constant

Mapping Register: Operand 0

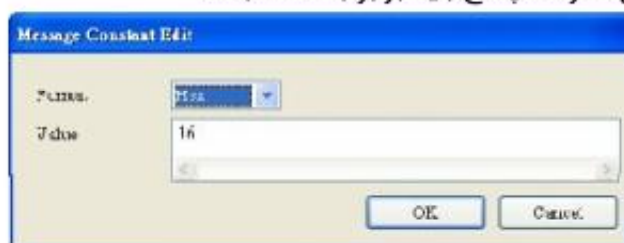
Constant: 1

OK Cancel

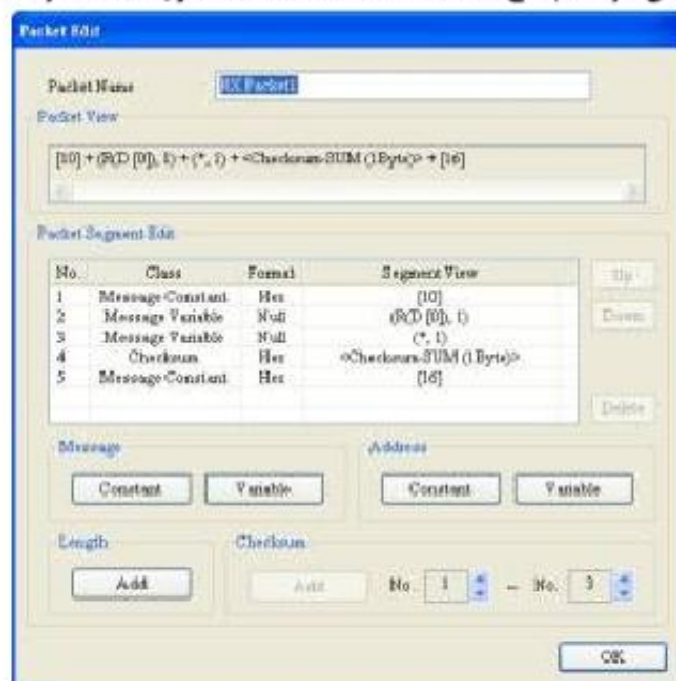
- مقدار Checksum برابر با 1byte بوده و از آن برای اضافه کردن به دو آیتم قبلی استفاده می‌شود.



- علامت Word پایان: فرمت پاسخ باید برابر با 16h باشد.



- در این مرحله، ویرایش فرمت پاسخ به اطلاعات ثبت شده به صورت مختصر تکمیل شده است.



حال نوبت به ویرایش فرمان ارسال رسیده است. در اینجا باید بسته‌ی 1 از TX را ارسال کرده و بسته‌ی 1 از RX را دریافت کرد.

Command Edit

Command No.

Command Type **Send & Receive**

Send Packet **TX Packet1**

Recv Packet **RX Packet1**

Success **End**

Fail **Abort**

Retry (0 - 255)

Repeat (0 - 255)

Send Wait (0 - 65535 ms)

Timeout (0 - 65535 ms)

OK **Cancel**

نوع 3: در این نوع اطلاعات ثبت شده به صورت مختصر ارسال شده و به ثبت کننده‌ی اطلاعات به صورت کامل پاسخ داده می‌شود. برای ارسال اطلاعات ثبت شده به صورت مختصر باید تنظیمات ارسال شده را مانند تنظیمات نوع 1 انجام دهید. توجه داشته باشید کد تابع (FF) در این نوع برابر با 89h می‌باشد.

Packet Edit

Packet Name **RX Packet1**

Packet View **[10] + (R/D (0), 1) + [09] + <Checksum-SUM (1 Byte)> + [16]**

Packet Segment Edit

No.	Class	Format	Segment View
1	Message Constant	Hex	[10]
2	Message Variable	Null	(R/D (0), 1)
3	Address Constant	Hex	[09]
4	Checksum	Hex	<Checksum-SUM (1 Byte)>
5	Message Constant	Hex	[16]

Message **Constant** Variable

Address **Constant** Variable

Length **Add**

Checksum **Add** No. **2** No. **3**

OK

فرمت پاسخ به ثبت کننده‌ی اطلاعات به صورت کامل به شرح زیر است:

Start Word + Length + Length (repeat) + Start Word + Device address (IA) + Function code (FF) + Parameter index (PI) + Data block (DB) + Checksum (CS) + End marker → 68h + (Null) + (Null) + 68h + D0 + (Null) + D100

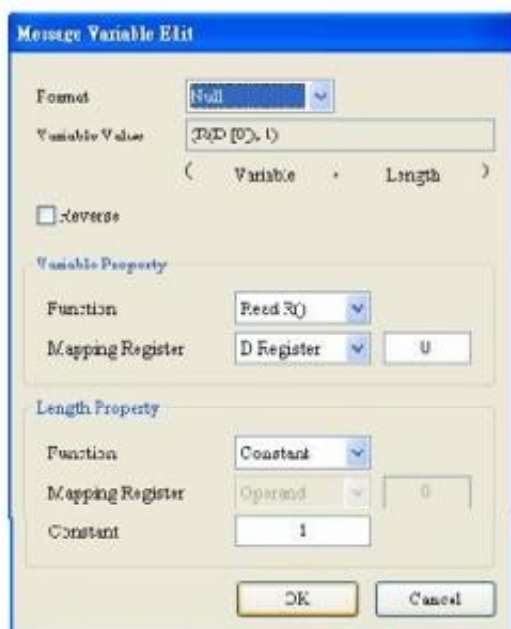
- مقدار Word شروع برابر با 68h می‌باشد.

- تکرار طول داده + طول داده: در این نوع (نوع 3) از این دو Word صرف نظر می‌شود. همچنین، می‌توان از این دو Word چشم پوشی نموده و یا مقدار آنها را ذخیره کرد.

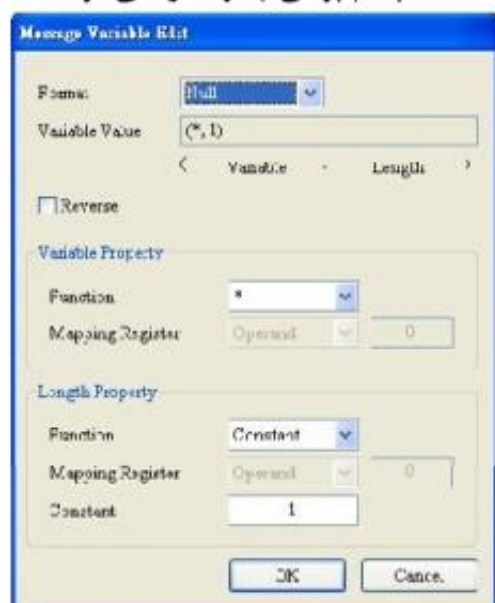
- دومین مقدار Word شروع در نوع 3 برابر با 68h می‌باشد.



- آدرس دستگاه (IA) : این قسمت آدرس پاسخ را بررسی کرده و آدرس دستگاه مکرراً از D0(IA) می‌خواند.



- کد تابع : در نوع 3 از این Word چشم پوشی (صرفنظر) می‌شود.



- داده‌ی بعد از کد تابع در رجیسترهایی که شروع آنها از D100 می‌باشد، ذخیره می‌شود.

No.	Class	Format	Segment Type
1	Message Constant	Hex	[00]
2	Message Variable	Null	C*, 2)
3	Message Constant	Hex	[00]
4	Message Variable	Null	(W(D [0], 1)
5	Message Variable	Null	C*, 1)
6	Message Variable	Null	(W(D [100], *)

توجه داشته باشید که از برخی از Wordهایی که بی‌اهمیت می‌باشند، می‌توان صرف‌نظر کرد. شما می‌توانید فقط داده‌هایی که نیاز دارید را در رجیسترهای (Dx) ذخیره نمایید. (منظور از X شماره‌ی رجیسترها می‌باشد) و با استفاده از این روش می‌توان داده‌ی کد پاسخ را ذخیره نمایید.

ویرایش فرمان در این نوع به صورت ارسال نخستین بسته‌ی RX می‌باشد.

Command Edit

Command No.

Command Type

Send Packet

Recv Packet

Success

Fail

Retry (0 - 255)

Repeat (0 - 255)

Send Wait (0 - 65535 ms)

Timeout (0 - 65535 ms)

نوع 4: در این نوع اطلاعات ثبت شده به صورت مختصر ارسال شده و به اطلاعات ثبت شده به صورت کامل پاسخ داده می‌شود. برای ارسال داده‌ی ثبت شده به صورت مختصر می‌توانید از تنظیمات داده مانند نوع 1 و نوع 2 استفاده نمایید. توجه داشته باشید که کد تابع (FF) در این نوع برابر با A9h است.

Packet Edit

Packet Name

Packet View

Packet Segment Edit

No.	Class	Format	Segment View
1	Message Constant	Hex	[10]
2	Message Variable	Null	(R(D [0], 1)
3	Address Constant	Hex	[A9]
4	Checksum	Hex	<Checksum-SUM (1Byte)>
5	Message Constant	Hex	[16]

Message:

Address:

Length:

Checksum: No. ~ No.

فرمت پاسخ به ثبت کننده‌ی اطلاعات به صورت کامل به شرح زیر است:

Start Word + Length + Length (repeat) + Start Word + Device address (IA) + Function code (FF) + Parameter index (PI) + Data block (DB) + Checksum (CS) + End marker → 68h + 06h + 06h + 68h + D0 + (1 Word) + (3 Words) + (the content gotten from adding from IA to the end) + 16h

- محتویات پیغام به صورت (Start Word-Length-Length-Start Word) و برابر با $68h + 06h + 06h + 68h$ می‌باشد.

- آدرس پاسخ را بررسی کرده و آدرس دستگاه باید مکرراً از D0(IA) خوانده شود. آدرس پاسخ با آدرس دستگاه باید برابر باشد.

- کد تابع (FF): در اینجا (نوع 4) از کد تابع چشم‌پوشی (صرفنظر) شده است.

Message Variable Edit

Format: Null

Variable Value: (*, 1)
(Variable * Length)

☐ Reverse

Variable Property

Function: *

Mapping Register: Operand 0

Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

OK Cancel

- در این نوع، مقدار (بلوک داده (DB) + پارامتر شاخص (PI)) در رجیستر D100 ذخیره می‌شود.

Message Variable Edit

Format: Null

Variable Value: (W(D [100]), 4)
(Variable * Length)

☐ Reverse

Variable Property

Function: Write W()

Mapping Register: D Register 100

Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 4


OK Cancel

- مقدار Checksum مطابق با شکل زیر می‌باشد.



The 'Checksum Edit' dialog box contains the following fields and controls:

- Class:** A dropdown menu set to 'SUM (1Byte)'.
- Format:** A dropdown menu set to 'Hex'.
- Initial Value:** A text input field containing '0'.
- Reverse:** An unchecked checkbox.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.



The 'Checksum' configuration panel includes:

- An 'Add' button.
- A range selector showing 'No. 2' to 'No. 4' with up and down arrows.

- علامت پایان (End Word) در این نوع مطابق با مقدار نشان داده در شکل زیر تنظیم خواهد

شد.

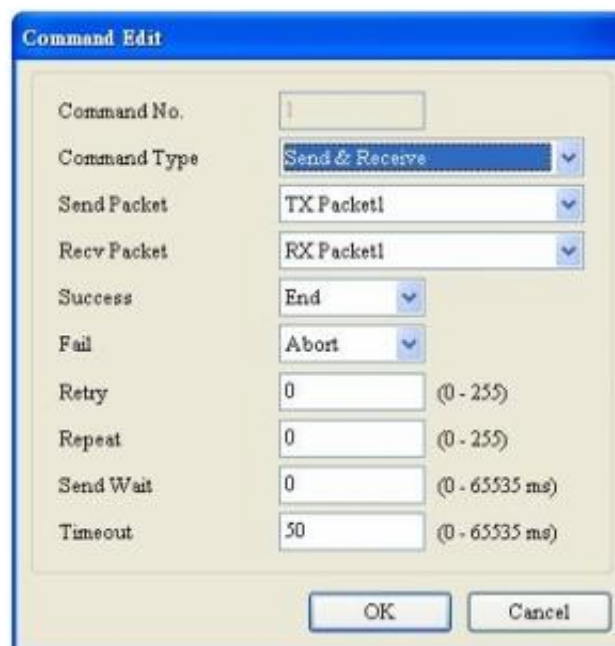


The 'Message Constant Edit' dialog box contains:

- Format:** A dropdown menu set to 'Hex'.
- Value:** A text input field containing '16'.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

- ویرایش فرمان در این نوع برای ارسال نخستین بسته‌ی TX و دریافت نخستین بسته RX مطابق با

شکل زیر است.



The 'Command Edit' dialog box contains the following fields and controls:

- Command No.:** A text input field containing '1'.
- Command Type:** A dropdown menu set to 'Send & Receive'.
- Send Packet:** A dropdown menu set to 'TX Packet1'.
- Recv Packet:** A dropdown menu set to 'RX Packet1'.
- Success:** A dropdown menu set to 'End'.
- Fail:** A dropdown menu set to 'Abort'.
- Retry:** A text input field containing '0' with a range '(0 - 255)'.
- Repeat:** A text input field containing '0' with a range '(0 - 255)'.
- Send Wait:** A text input field containing '0' with a range '(0 - 65535 ms)'.
- Timeout:** A text input field containing '50' with a range '(0 - 65535 ms)'.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

نوع 5: در این نوع اطلاعات ثبت شده به صورت مختصر ارسال می‌شود و به اطلاعات ثبت شده به صورت کامل پاسخ داده می‌شود. وقتی که اطلاعات ثبت شده به صورت کنترل شده (Control record) ارسال شود، کد تابع (FF) برابر با 89h می‌باشد. فرمت داده در این نوع به شکل زیر است.

Start Word + Length + Length (repeat) + Start Word + Device address (IA) + Function code (FF) + Parameter index (PI) + Checksum (CS) + End marker → 68h + 03h + 03h + 68h + D0 + 89h + D1 + (the content gotten from adding from IA to the end) + 16h

- محتویات شروع پیغام به صورت (Start Word-Length-Length-Start Word) و برابر با 68h + 03h + 03h + 68h می‌باشد.



- آدرس خوانده شده از D0 به صورت زیر تنظیم می‌شود.



- مقدار کد تابع (Function code) که برابر با 89h می‌باشد، به صورت زیر تنظیم می‌شود.



The 'Message Constant Edit' dialog box has a 'Format' dropdown menu set to 'Hex' and a 'Value' text field containing '89'. There are 'OK' and 'Cancel' buttons at the bottom right.

- مقدار پارامتر شاخصی که قرار است از D1 خوانده شود، به صورت زیر تنظیم می‌شود.



The 'Message Variable Edit' dialog box contains several sections:

- Format:** A dropdown menu set to 'Null'.
- Variable Value:** A text field containing '(R(D [1]), 1)' with a hint '(Variable , Length)' below it.
- Reverse:** An unchecked checkbox.
- Variable Property:**
 - Function:** A dropdown menu set to 'Read R()'. The text 'Read R()' is also visible in the dropdown.
 - Mapping Register:** A dropdown menu set to 'D Register' and a text field containing '1'.
- Length Property:**
 - Function:** A dropdown menu set to 'Constant'.
 - Mapping Register:** A dropdown menu set to 'Operand' and a text field containing '0'.
 - Constant:** A text field containing '1'.

 There are 'OK' and 'Cancel' buttons at the bottom right.

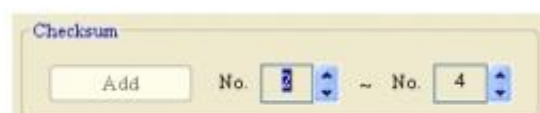
- مقدار Checksum مطابق با شکل زیر می‌باشد:



The 'Checksum Edit' dialog box has:

- Class:** A dropdown menu set to 'SUM (1Byte)'.
- Format:** A dropdown menu set to 'Hex'.
- Initial Value:** A text field containing '0'.
- Reverse:** An unchecked checkbox.

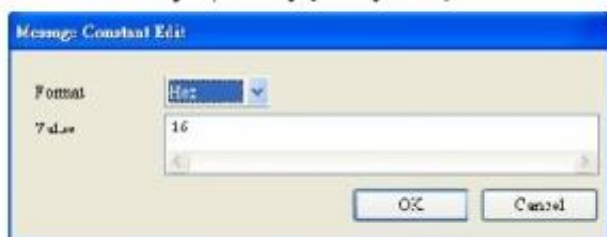
 There are 'OK' and 'Cancel' buttons at the bottom.



A horizontal bar for checksum configuration with the label 'Checksum' on the left. It contains:

- An 'Add' button.
- A 'No.' label followed by a spinner box set to '3'.
- A '~' symbol.
- A 'No.' label followed by a spinner box set to '4'.

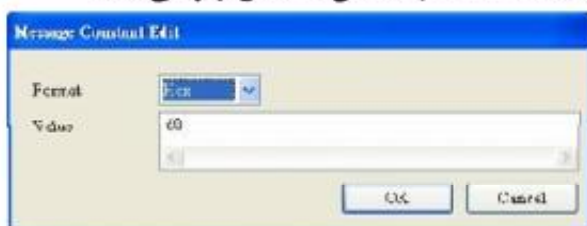
- مقدار علامت پایان (End Word) به فرمت زیر تنظیم خواهد شد.



فرمت پاسخ به ثبت کننده‌ی اطلاعات به صورت کامل به شرح زیر است:

Start Word + Length + Length (repeat) + Start Word + Device address (IA) + Function code (FF) + Parameter index (PI) + Data block (DB) + Checksum (CS) + End marker → 68h + (Null) + (Null) + 68h + D0 + (Null) + D1 + D100 + (the content gotten from adding from IA to the end) + 16h

- تنظیم کلمه شروع (Start Word) مطابق با شکل زیر می‌باشد:



- در این نوع از دو کلمه‌ی طول - طول (Length-Length (two Words)) صرفنظر شده

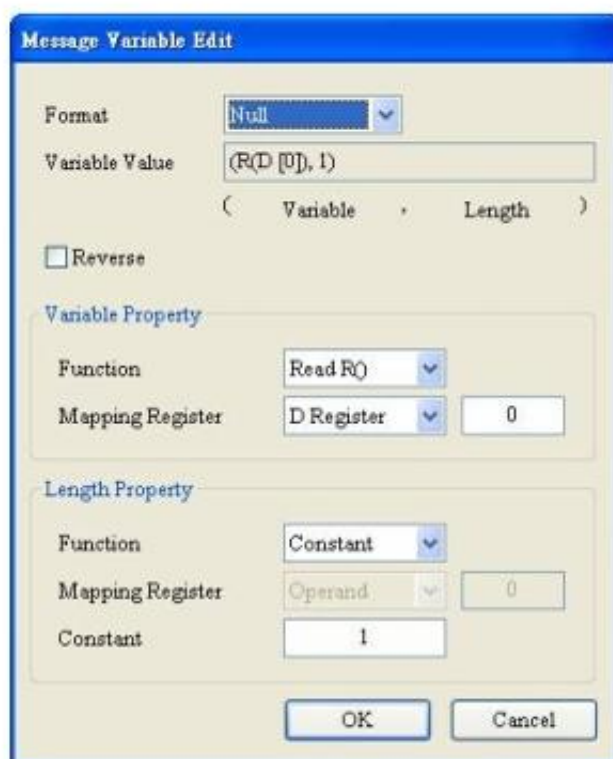
است.



- تنظیم کلمه‌ی شروع (Start Word) مطابق با شکل زیر است :



- باید هر یک از آدرس‌های پاسخ و دستگاه را مکرراً بررسی کرده و از یکسان بودن آنها مطمئن شد. برای این عمل باید تنظیمات زیر را انجام دهید. دیتا باید از رجیستر (IA) D0 خوانده شده و با آدرس پاسخ دستگاه یکسان باشد.



- کد تابع (Function code) این نوع را باید مطابق با شکل زیر تنظیم نمایید.

Message Variable Edit

Format: Null

Variable Value: (*, 1)
(Variable , Length)

☐ Reverse

Variable Property

Function: *

Mapping Register: Operand 0

Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

OK Cancel

- هر یک از پارامترهای شاخص دریافت شده و پارامترهای شاخص ارسال شده باید دائماً بررسی شوند و از یکسان بودن آنها اطمینان حاصل شود. تنظیمات این بخش مطابق با شکل زیر می‌باشد.

Message Variable Edit

Format: Null

Variable Value: (R[D (1), 1)
(Variable , Length)

☐ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 1

Length Property

Function: Constant

Mapping Register: Operand 0

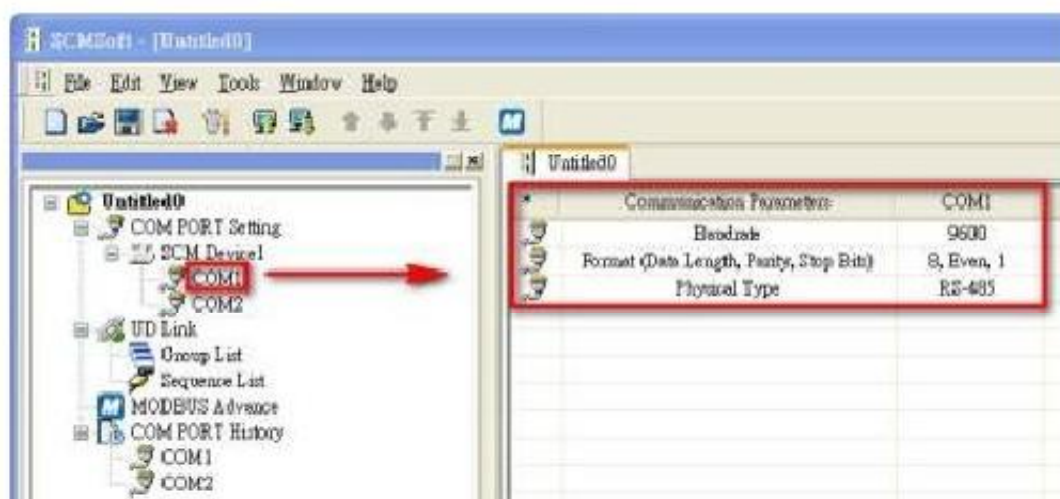
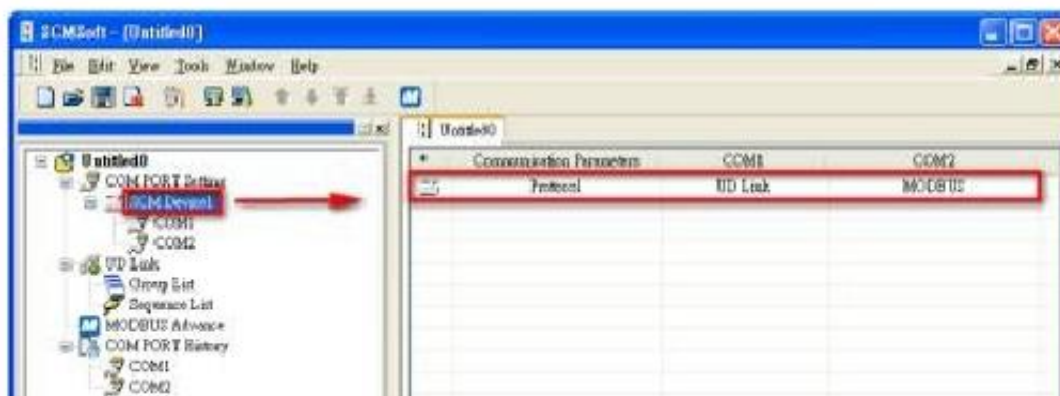
Constant: 1

OK Cancel

- از بلوک داده برای ذخیره کردن مقدار داده‌ی پاسخ داده شده در رجیسترهایی که شروع آنها از D100 می‌باشد، استفاده شده است. برای تنظیم این بخش می‌توانید به شکل زیر مراجعه نمایید.

مرحله‌ی 4 ← بعد از اینکه تنظیمات تمام انواع ارسال و دریافت داده به خط‌کش الکتریکی را انجام دادید، باید این تنظیمات UD Link را به ماژول SCM دانلود کنید. برای دانلود تنظیمات انجام شده به ماژول SCM ابتدا باید نرم‌افزار SCM Soft را باز کنید، یک پروژه‌ی جدید را ایجاد کرده و سپس برای تنظیم پورت COM بر روی COM PORT Setting راست کلیک کرده و گزینه‌ی Add SCM COM را انتخاب نمایید. در نهایت، باید پارامترهای ارتباطات را تنظیم کنید.





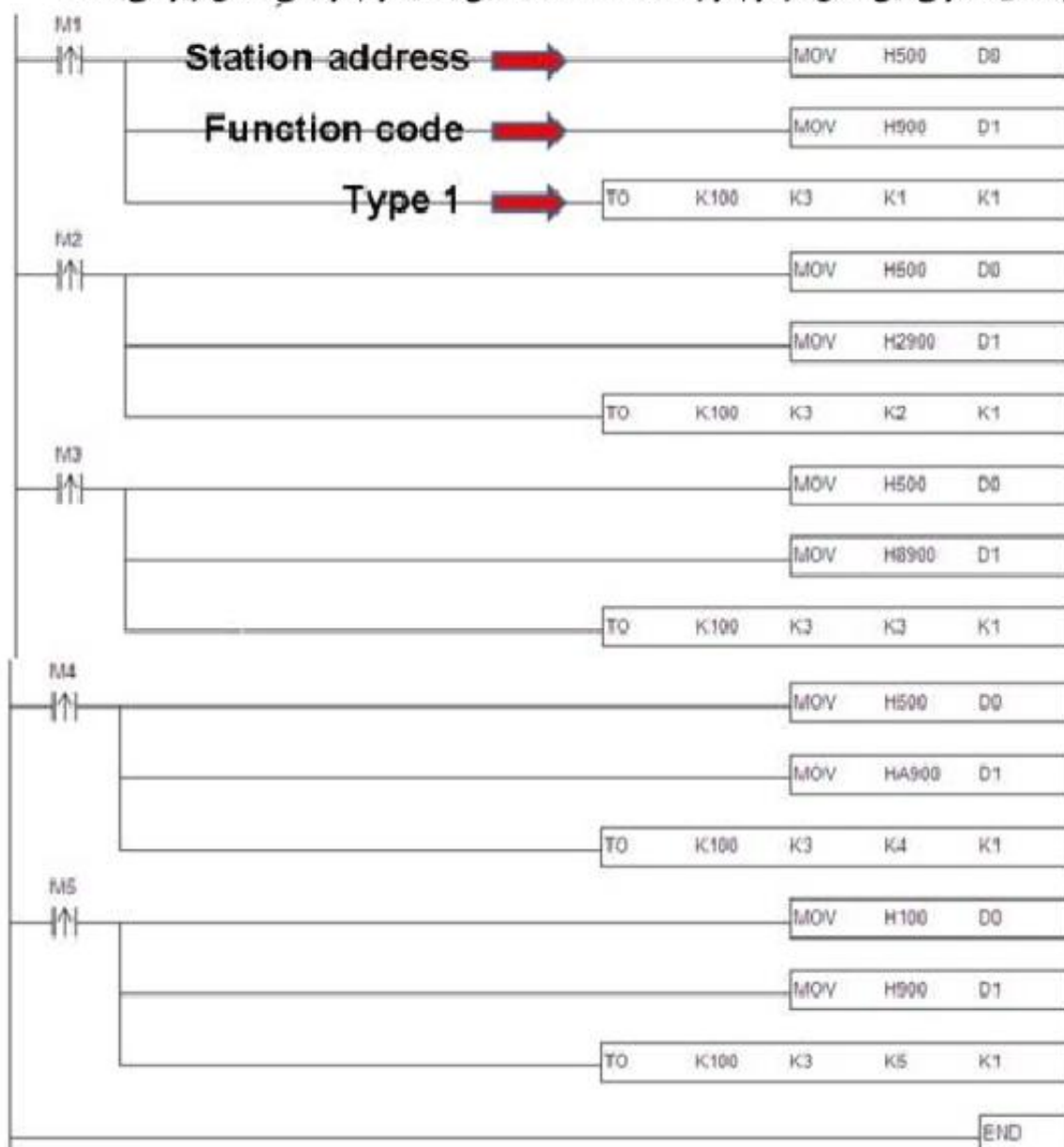
تنظیمات پارامترهای ارتباطات برای پورت COM1 مازول SCM در این مثال به شرح زیر می‌باشد:
 آدرس ایستگاه مازول SCM برابر با 247 تنظیم شده، شبکه UD Link فعال شده، سرعت تبادل داده 9600 انتخاب شده، طول داده 8 بیت باشد، بیت توازن Even بوده و تعداد بیت توقف یک بیت تنظیم شود.

*	Communication Parameters	COM1
	Baudrate	9600
	Format (Data Length, Parity, Stop Bits)	8, Even, 1
	Physical Type	RS-485

مرحله‌ی 5 ← در نهایت، نوبت راه‌اندازی UD Link توسط نرم‌افزار WPLSoft رسیده است.
 توسط دستورالعمل TO باید شماره گروه تنظیم شده برای هر نوع را در نرم‌افزار WPLSoft راه‌اندازی و اجرا کرد. اگر شماره‌ی گروه مورد نظر 1 است، باید مقدار K1 را در رجیستر کنترلی CR#3 نوشته و اگر شماره گروه مورد نظر 2 است، باید مقدار K2 را در رجیستر کنترلی مازول DVP SCM نوشت.

#CR	صفت	نام رجیستر	توضیحات
3	R/W	شماره‌ی گروه راه‌اندازی شده توسط پورت COM1 در شبکه‌ی UD Link	در این رجیستر کنترلی باید شماره‌ی گروه‌ی را که می‌خواهید در پورت COM1 توسط شبکه‌ی UD Link راه‌اندازی نمایید را وارد نمایید.

برنامه‌ی کنترلی این مثال در نرم‌افزار WPL Soft مطابق با دیاگرام نردبانی شکل زیر می‌باشد:



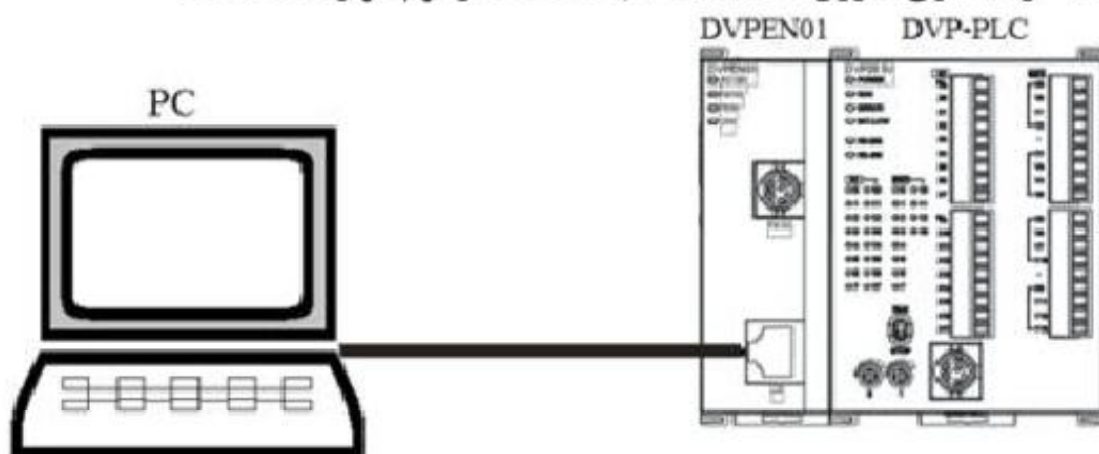
عملکرد برنامه‌ی کنترلی این مثال به شرح زیر است:

توسط فلگ‌های عمومی M1~M5 ارسال نوع 1 تا 5 را می‌توان کنترل کرد. هر راه‌اندازی شامل نوشتن آدرس ایستگاه خط‌کش الکتریکی در شبکه و کد تابع به ترتیب در داخل رجیستر D0 و D1

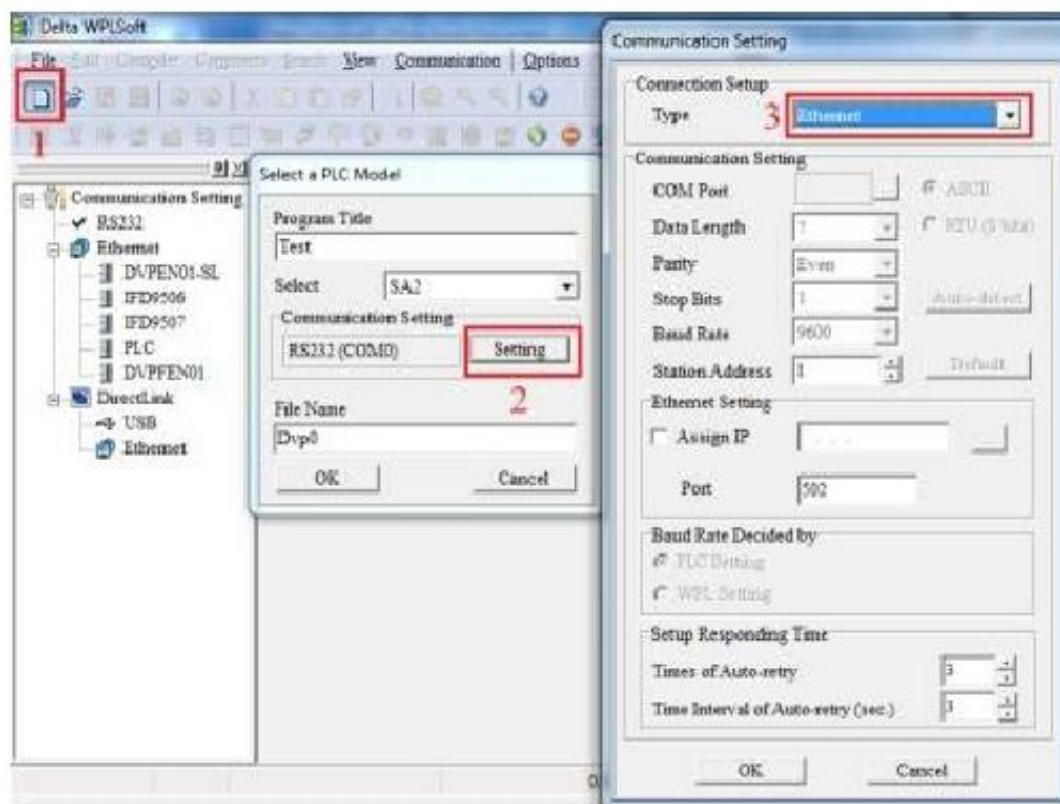
می‌باشد. وقتی که دیتا در رجیسترها نوشته شود، بیت با ارزش مقدم به بیت کم‌ارزش می‌باشد. برای مثال، وقتی که آدرس ایستگاه 5 است، شما باید مقدار H'0555 را وارد نمایید و همین آدرس برای خواندن پاسخ از رجیستر D100 را نیز وارد نمایید.

7- 10- مازول DVPEN01

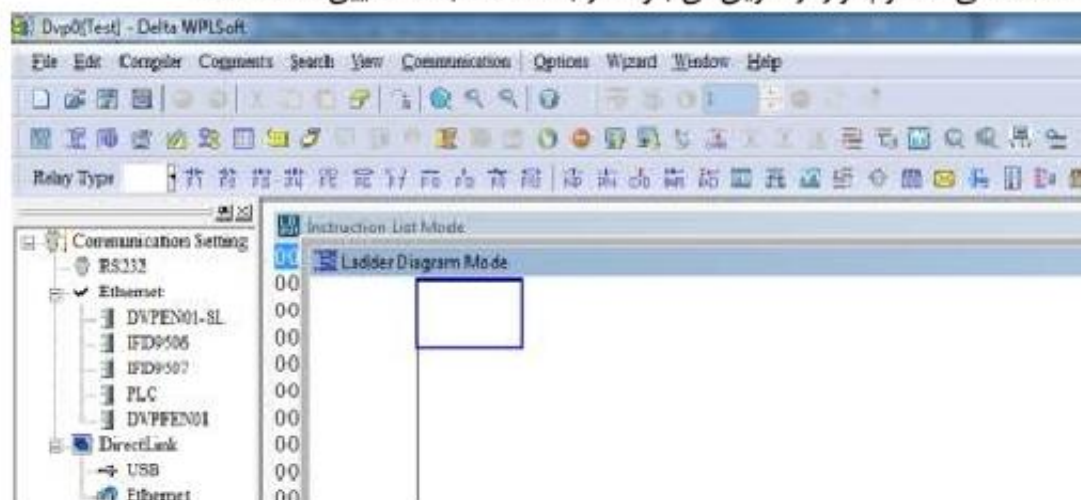
7- 10- 1 راه‌اندازی مازول DVPEN01 با استفاده از نرم‌افزار WPL Soft



برای اینکه بتوانیم از طریق WPL Soft با مازول DVPEN01 ارتباط برقرار کنیم، همزمان با این نرم‌افزار باید نرم‌افزار DCI Soft نیز اجرا شود. باز نبودن همزمان این نرم‌افزار دلیل بسیاری از خطاها و هشدارها مانند "SCM Soft Parameter Error!" در WPL Soft می‌باشد. حال می‌خواهیم پروژه جدیدی در WPL Soft تعریف کنیم. در قسمت Communication Setting نوع یا Type ارتباطات را به صورت Ethernet انتخاب می‌کنیم. البته تنظیم نوع ارتباطات بعد از ساخت پروژه نیز امکان‌پذیر است که در ادامه این مورد شرح داده خواهد شد.

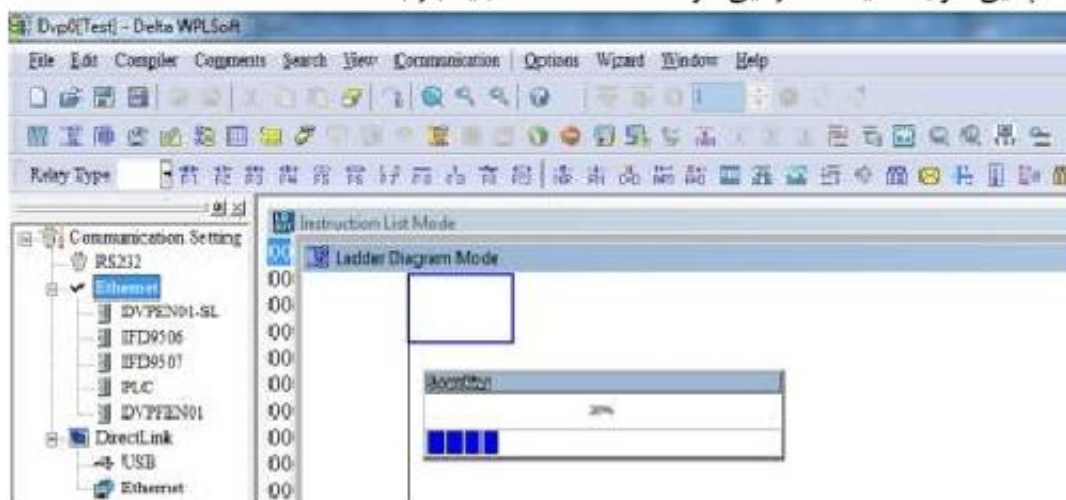


پس از ایجاد پروژه، پنجره آن مانند شکل زیر خواهد شد که در سمت چپ آن بخش تنظیمات ارتباطی قرار دارد. مشخص است که هنوز هیچ روشی برای ارتباط از طریق Ethernet تنظیم نشده است، چرا که IP دستگاهی که نرم‌افزار از طریق آن بتواند ارتباط داشته باشد، تعیین نشده است.

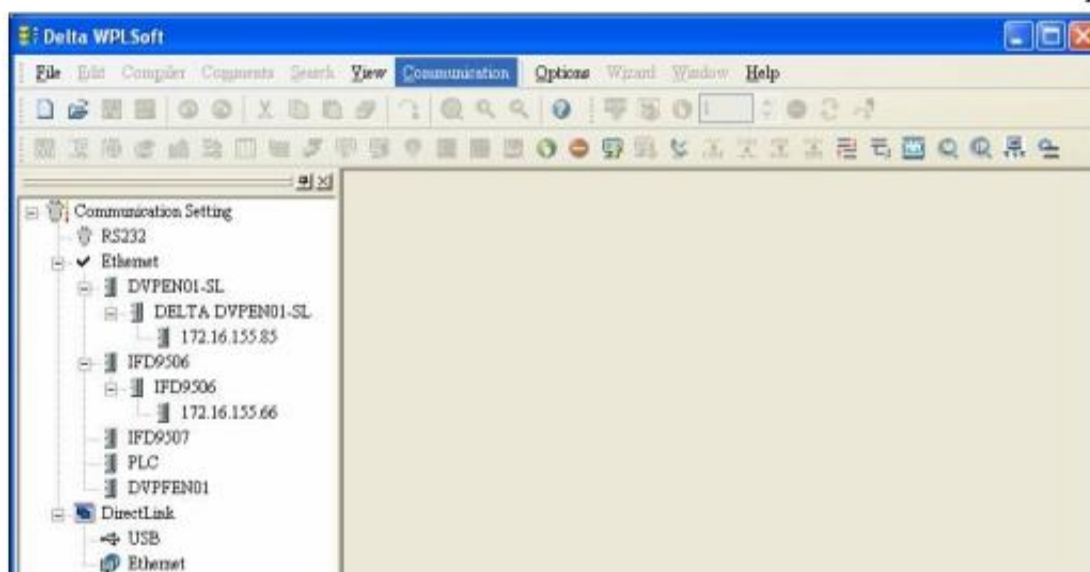


برای اینکه سیستم بتواند ماژول‌های متصل شده به Ethernet را به صورت اتوماتیک جستجو و شناسایی نماید، بر روی آیکون (Auto-Search Ethernet Module) کلیک می‌کنیم. اگر

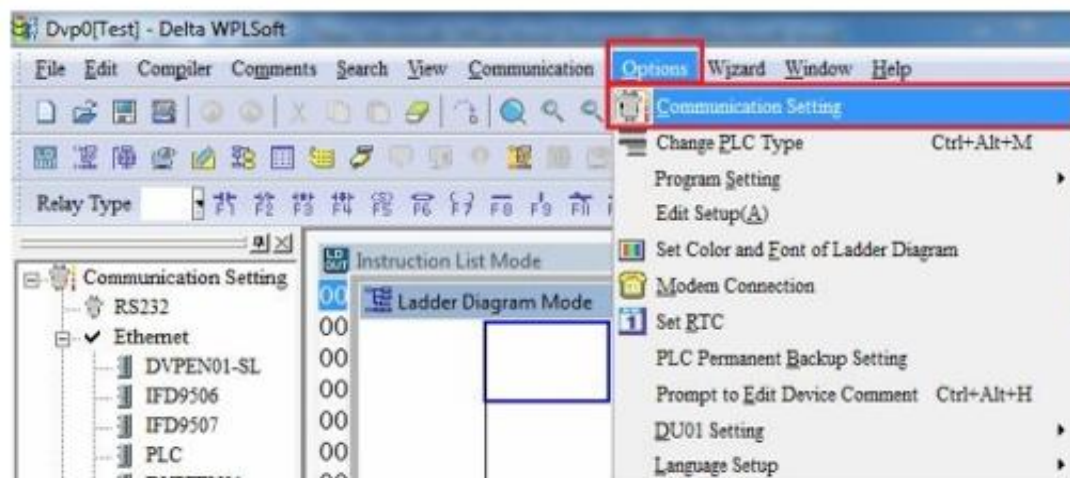
این آیکون غیر فعال است - به صورت - ابتدا بر روی Ethernet در سمت چپ صفحه کلیک کنید. همچنین، توجه کنید که در این مرحله DCI Soft باید باز باشد.



در انتهای جستجو IP ماژول‌های یافت شده در بخش Communication Setting لیست می‌شوند.



حال اگر در ابتدای ساخت پروژه، تنظیمات شبکه Ethernet را انجام ن داده‌اید، می‌توانید با انتخاب Communication Setting از منوی Option نوع (Type) شبکه را به صورت Ethernet انتخاب کنید. دیگر بخش‌ها را به صورت پیش‌فرض قرار داده و بر روی دکمه‌ی OK کلیک نمایید. توجه داشته باشید که با اینکه در راهنمای رسمی کمپانی دلتا گفته شده است در صورتی که از Station Address اطلاعاتی ندارید آن را برابر صفر قرار دهید، ولی مشاهده شده است که گاهی این موضوع باعث ایجاد مشکلاتی در ارتباط نرم‌افزار با شبکه شده است.



پس از طی مراحل فوق، کاربر برای اطمینان از اتصال کامپیوتر به PLC می‌تواند از تستی ساده استفاده کند. در ابتدا موارد زیر را بررسی کنید:

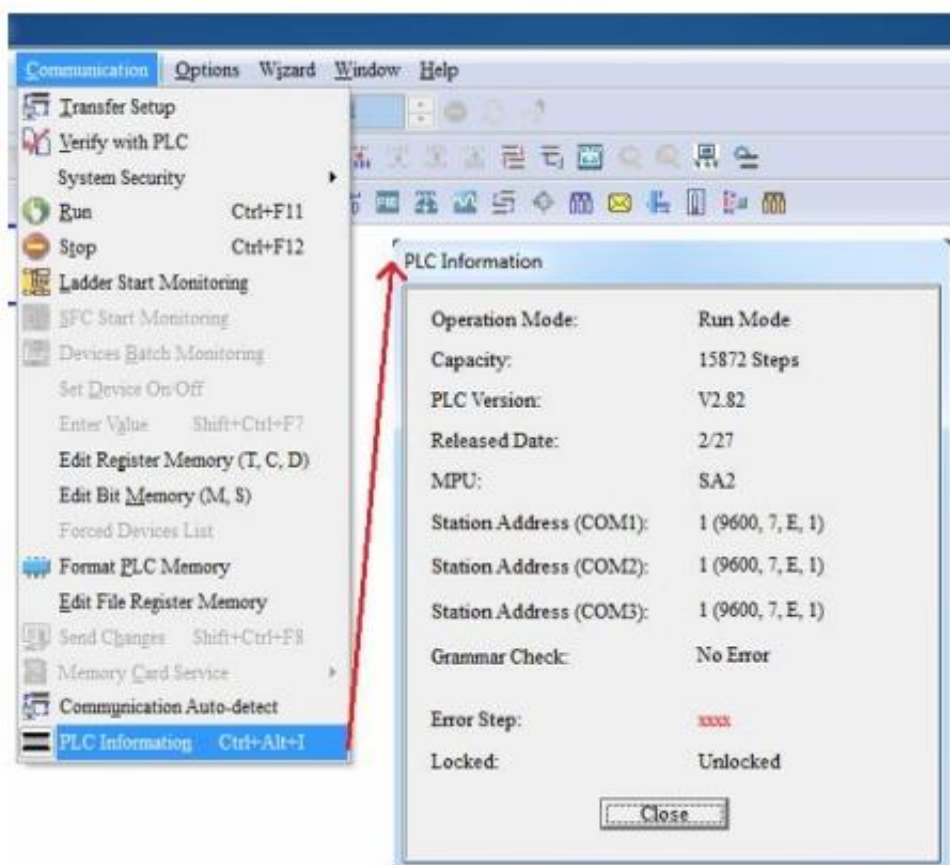
- وضعیت درایور در حالت Start باشد. کابل شبکه RJ-45 هم به ماژول DVPEN01 و هم به کامپیوتر متصل باشد.

- وضعیت کانال ارتباطی شامل کارت شبکه کامپیوتر، Hub و پورت سریال عادی باشد.

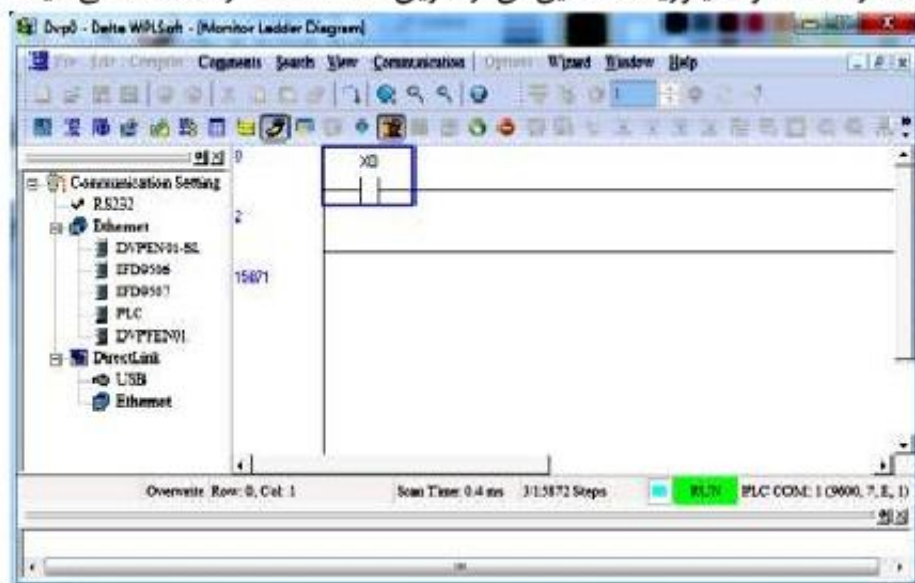
- درایور، آدرس Station و IP در Communication Setting درست تنظیم شده باشد.

- PLC به درستی به DVPEN01 متصل شده باشد. تغذیه PLC متصل و وضعیت آن عادی باشد.

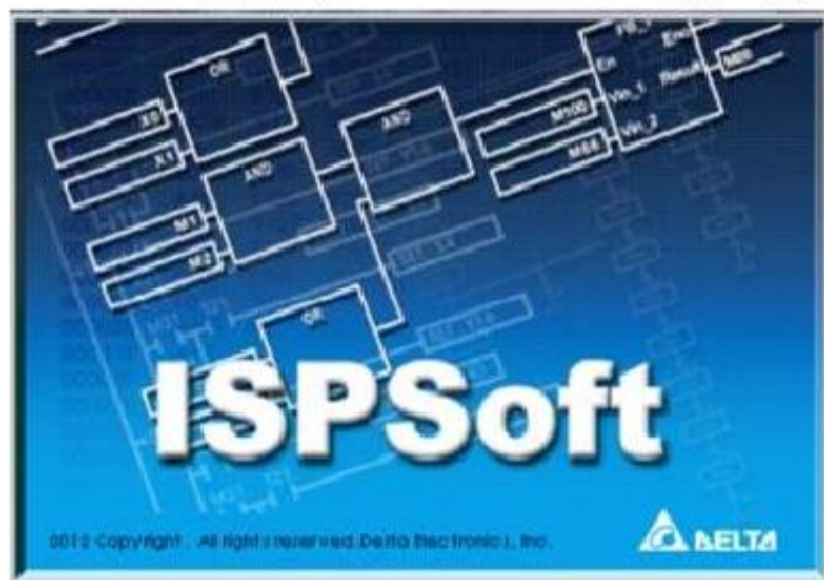
حال می‌توان از منوی Communication گزینه System Information را انتخاب کرد. اگر ارتباط PLC با کامپیوتر به صورت نرمال برقرار شود، آنگاه صفحه System Information ظاهر شده و اطلاعات رسیده از PLC نمایش داده می‌شود.



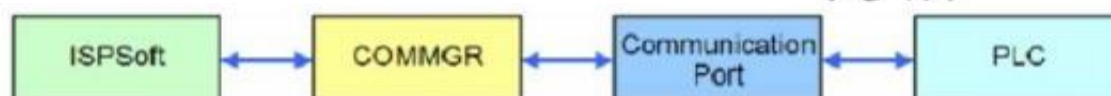
اکنون می‌توان با استفاده از Ethernet برنامه را بر روی PLC بارگذاری یا مانیتور کرد. در زیر، بارگذاری برنامه در PLC و مانیتورینگ آن‌ها از طریق Ethernet را مشاهده می‌کنید:



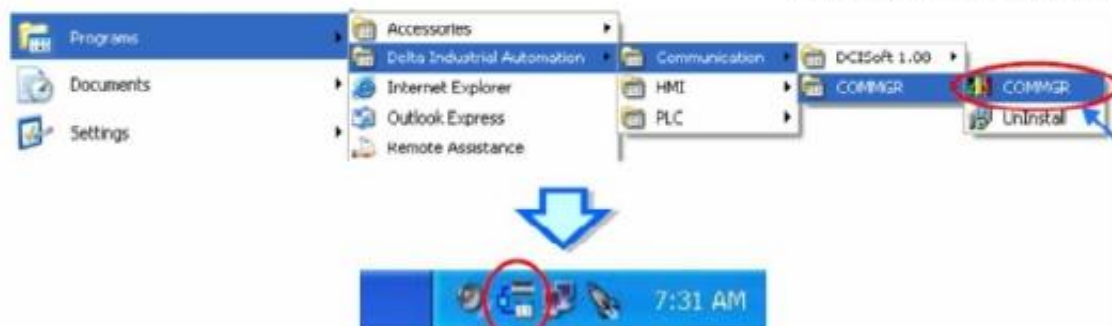
7-10-2 راه اندازی ماژول DVPEN01 با استفاده از نرم افزار ISPSOft



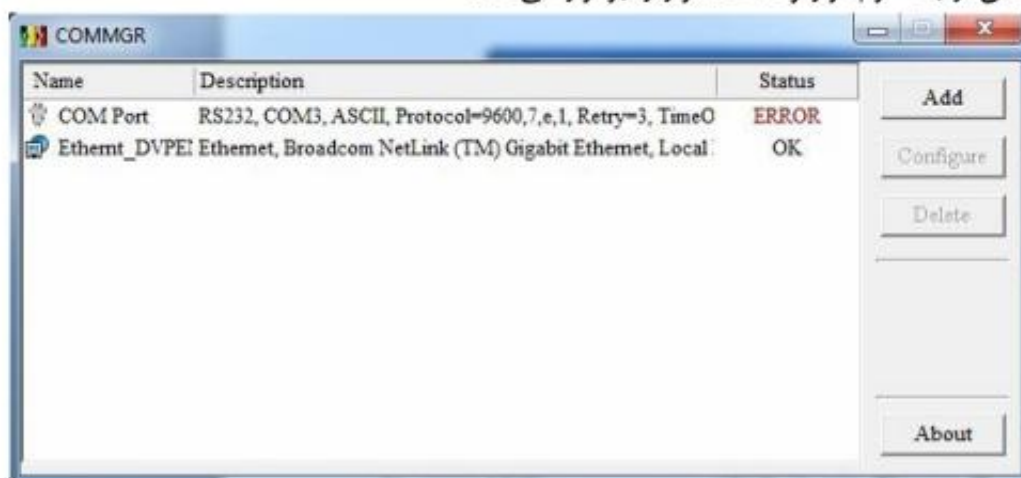
ارتباط بین ISPSOft و PLC های دلتا به صورت بلوک دیاگرام زیر از طریق نرم افزار COMMGR برقرار می شود.



پس از نصب نرم افزار COMMGR آیکون آن در نوار Taskbar ویندوز ظاهر می شود. همچنین، پس از هر بار شروع مجدد ویندوز، این نرم افزار در ابتدای بالا آمدن ویندوز خود به خود فعال می شود. در صورتی که به هر دلیلی این نرم افزار غیر فعال باشد و بخواهیم آن را فعال کنیم، در منوی Start در بخش نرم افزارها طبق مسیر زیر می توانیم COMMGR را فعال کنیم. پس از اجرای COMMGR می توانید با دوبار کلیک کردن بر روی آیکون آن پنجره COMMGR را فعال نمایید.



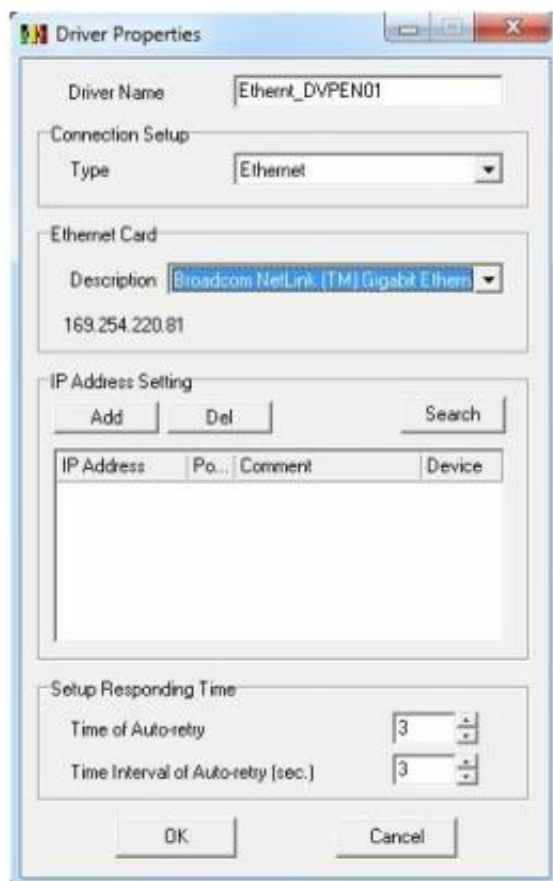
پنجره COMMGR مطابق شکل زیر باز خواهد شد که با گزینه‌های سمت راست می‌توان درایورهای آن را مدیریت کرد. درایورهای لیست شده در COMMGR ارتباط نرم‌افزار و پورت‌های ارتباطی کامپیوتر را برقرار می‌کند. درایور در واقع، یک سری دستورالعمل است که کامپیوتر از آنها پیروی می‌کند تا اطلاعات را برای انتقال به دستگاه جانبی خاص یا بازبانی از آن دوباره قالب‌بندی کند و به این وسیله امکان ارتباط نرم‌افزار و سخت‌افزار را برقرار می‌کند.



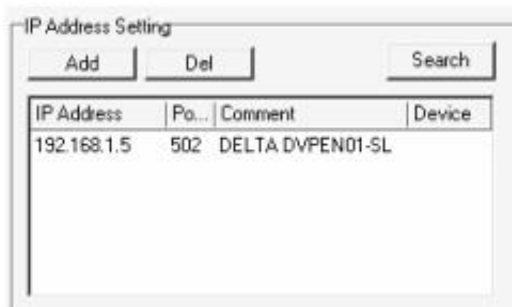
در صورتی که ارتباط نرم‌افزار با پورت‌های مشخص شده به وسیله COMMGR برقرار باشد، در ستون Status وضعیت ارتباط به صورت START مشخص خواهد شد، با این حال، اگر COMMGR نتواند به هر دلیلی با پورت مورد نظر ارتباط برقرار کند (این حالت ممکن است به علت اشغال بودن پورت در اثر استفاده نرم‌افزاری دیگر باشد)، درایور متوقف شده و وضعیت ERROR به نمایش خواهد آمد.

ساخت درایور در COMMGR برای شبکه Ethernet به شرح زیر است:

ابتدا بر روی دکمه‌ی Add در COMMGR کلیک کنید تا صفحه Driver Properties باز شود. در این قسمت می‌توان اسم درایور را به صورت دلخواه در قسمت Driver Name تعیین کرد. توجه داشته باشید که از این نام در آینده برای ارتباط نرم‌افزارهای شرکت دلتا با پورت مورد نظر باید استفاده شود به همین سبب پیشنهاد می‌شود مکانیزم مشخصی برای انتخاب نام درایور شامل در نظر گرفتن نام پورت و ویژگی‌های آن لحاظ شود تا از سردرگمی جلوگیری شود. نوع پورت (پروتکل) ارتباطی مورد نظر خود را نیز می‌توان در قسمت Connection Setup تنظیم کرد. در اینجا، پورت مورد نیاز Ethernet است.



در قسمت Ethernet Card سخت‌افزار کارت شبکه کامپیوتر لیست شده است و IP آن نیز در سمت چپ آن (به عنوان مثال، 192.168.1.1) مشخص شده است. در قسمت بعدی یعنی IP Address Setting باید IP ماژول‌هایی که از طریق Ethernet می‌خواهیم با آنها در ارتباط باشیم را لیست کنیم. البته در صورتی که ماژول‌ها به شبکه متصل باشند، می‌توانیم با کلیک بر روی Search به صورت اتوماتیک به لیست IP آنها دسترسی داشته و آنها را به درایور اضافه کنیم. در شکل زیر، می‌بینیم که پس از Search کردن، نرم‌افزار DVPEN01 را تشخیص و IP آن را به درایور اضافه کرده است.



در قسمت **Setup Responding Time** نیز کاربر می‌تواند حداکثر تعداد تلاش برای ارتباط و حداکثر مدت زمان انتظار برای اینکار را به ترتیب در دو قسمت **Time of Auto-retry** و **Time Interval of Auto-retry** مشخص نماید. پس از تایید و ساخت درایور می‌توان از طریق گزینه‌های سمت راست **COMMGR** آنها را فعال و یا غیرفعال کرد و یا اینکه توسط گزینه **Configure** آنها را دوباره تنظیم کرد.

تنظیم ارتباط بین نرم‌افزار **ISPSOft** و **COMMGR** به شرح زیر است:

پس از تنظیم درایورها در **COMMGR** کاربر می‌تواند در **ISPSOft** نیز درایور را در هر پروژه برای ارتباط با **PLC** تعیین نماید. برای اینکار اگر پروژه گروهی است، ابتدا باید پروژه داخلی مورد نظر خود را فعال نموده و سپس در سربرگ **Tools** گزینه **Communication Settings** را انتخاب کنید.



در پنجره جدید در قسمت **Driver** باید نوع درایور را مشخص کرد. همچنین، **Station Address** متناظر با **PLC** که با **PC** در ارتباط است، باید تعیین شود. اگر **Station Address** را نمی‌دانید، می‌توانید به جای آن صفر را انتخاب نمایید. اگر نوع ارتباط درایور به صورت **Ethernet** باشد، آنگاه باید **IP** تنظیم شده در **COMMGR** مربوط به دستگاهی که می‌خواهید با آن ارتباط برقرار کنید را انتخاب نمایید. توجه داشته باشید که با آنکه در راهنمای رسمی شرکت دلتا گفته شده در صورتی که از **Station Address** اطلاعاتی ندارید آن را برابر صفر قرار دهید، ولی مشاهده شده است که گاهی این موضوع باعث ایجاد مشکلاتی در ارتباط نرم‌افزار با شبکه شده است.

پس از اتمام تنظیمات، اطلاعات در مورد درایور متصل شده در نوار وضعیت نمایش داده می‌شود.



پس از طی مراحل فوق برای اطمینان از اتصال کامپیوتر به PLC می‌توانید از تستی ساده استفاده کنید. در ابتدا موارد زیر را بررسی نمایید :

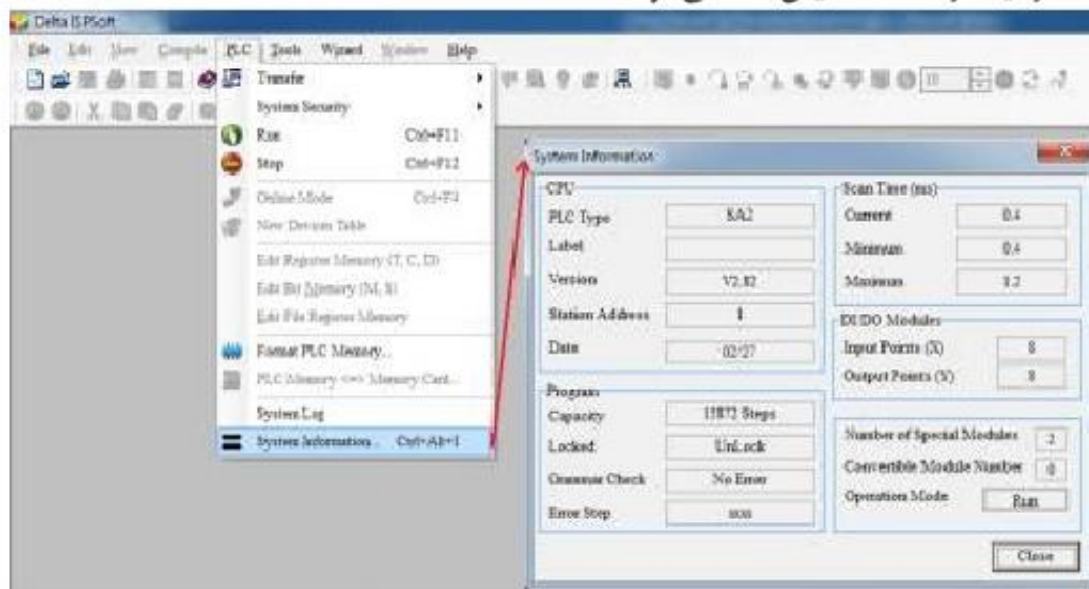
- وضعیت درایور در حالت Start باشد. کابل شبکه RJ-45 هم به ماژول DVPEN01 و هم به کامپیوتر متصل باشد.

- وضعیت کانال ارتباطی شامل کارت شبکه کامپیوتر، Hub و پورت سریال عادی باشد.

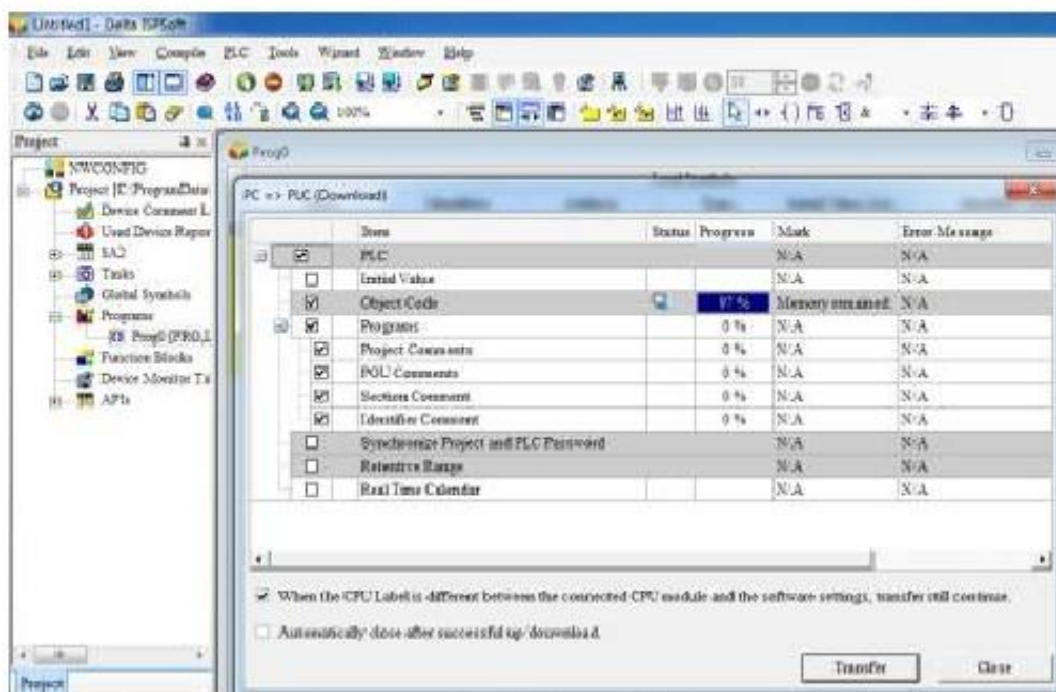
- درایور، آدرس Station و IP در Communication Setting درست تنظیم شده باشد.

- PLC به درستی به DVPEN01 متصل باشد. تغذیه PLC متصل و وضعیت آن عادی باشد.

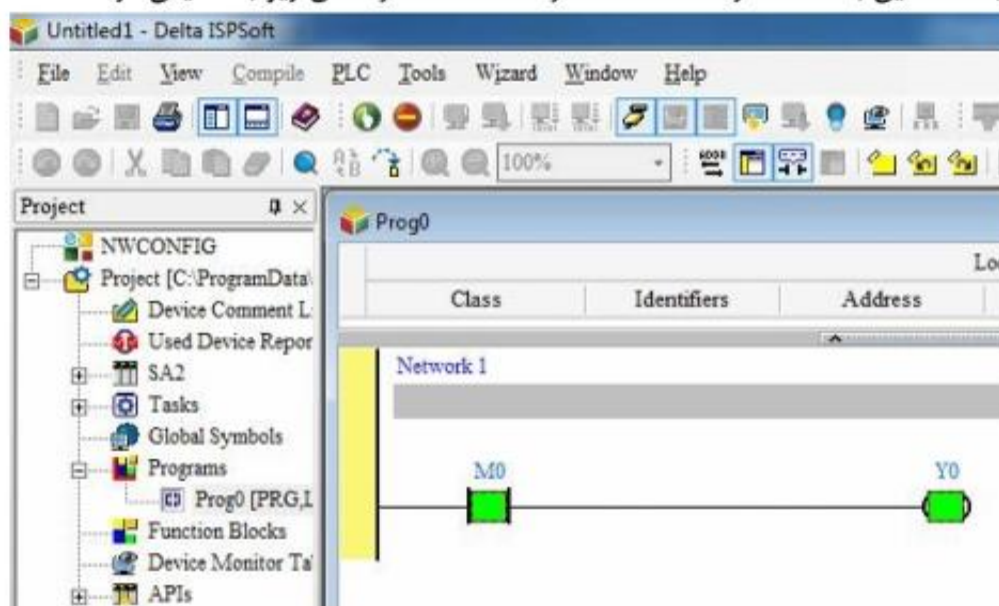
اکنون می‌توان در منوی PLC گزینه System Information را انتخاب کرد. اگر ارتباط PLC با کامپیوتر به صورت نرمال برقرار شود، آنگاه صفحه System Information ظاهر شده و اطلاعات رسیده از PLC نمایش داده می‌شود.



اکنون می‌توان با استفاده از شبکه Ethernet برنامه را بر روی PLC بارگذاری و یا مانیتور کرد. همچنین، در NWCONFIG می‌توان به تنظیم شبکه پرداخت. در زیر، بارگذاری برنامه در PLC از طریق Ethernet را مشاهده می‌کنید.



مانیتورینگ آنلاین با استفاده از Ethernet در ISPSoft در شکل زیر به نمایش در آمده است:



7- 11- مازول DVPEN01-SL

7- 11- 1 برنامه‌ی ارسال ایمیل توسط مازول DVPEN01-SL

در این مثال، می‌خواهیم نحوه‌ی ارسال یک ایمیل به کاربر در هنگام تغییر وضعیت X0 و Y0 را بررسی نماییم. در این مثال، IP address سرور SMTP برابر با 172.16.144.121 است. آدرس

ایمیل کاربر فرضی test@sample.com است. وقتی که ورودی X0 و خروجی Y0 تغییر وضعیت دهند، یک پیام ایمیل تولید خواهد شد.

بعد از این که بین مازول و کامپیوتر ارتباط برقرار کردید، باید در نرم‌افزار DCI Soft پنجره تنظیمات مازول DVPEN01-SL را باز کرده و تب Mail را انتخاب نمایید.

DELTA DVPEN01-SL

Overview | Basic | Mail | SNMP | Data Exchange | MELSEC Protocol | RTU Mapping | IP Filter

SMTP Setting

SMTP Server: Port:

Mail From:

User Name: Password:

E-mail Subject of Event

	Subject of Event						
1	DVPEN01-SL MAIL EVENT 1	<input type="text" value="0"/>	~	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
2	DVPEN01-SL MAIL EVENT 2	<input type="text" value="0"/>	~	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
3	DVPEN01-SL MAIL EVENT 3	<input type="text" value="0"/>	~	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
4	DVPEN01-SL MAIL EVENT 4	<input type="text" value="0"/>	~	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Recipient E-mail Address

	Event-1	Event-2	Event-3	Event-4	Mail Address
1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

در این مرحله باید رویدادی را که قرار است ایمیل شود را انتخاب نماییم. آدرس سرور SMTP را باید وارد نماییم. متن ایمیل‌ها را باید در فیلد مربوطه وارد کرد. نام کاربری و پسورد مورد نظر را باید وارد کرد. آدرس ایمیل فرضی را باید وارد کرد. مقادیر جاری رجیسترهایی را که می‌خواهیم پیوست ایمیل ارسال نماییم را باید در روبروی متن هر ایمیل تنظیم نماییم. این رجیسترها شامل ابزار D، ابزار T و ابزار C حافظه PLC می‌باشند.

DELTA DVPEN01-SL

Overview | Basic | Mail | SNMP | Data Exchange | MELSEC Protocol | RTU Mapping | IP Filter

SMTP Setting

SMTP Server: 172 . 161 . 144 . 122 Port: 25

Mail From: Message@DVPEN01-SL

User Name: AAAA Password: ****

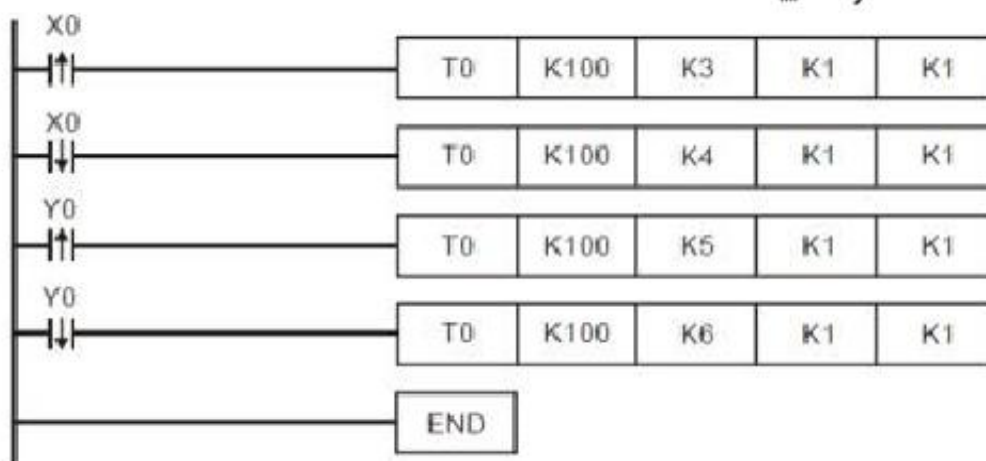
E-mail Subject of Event

	Subject of Event						
1	DVPEN01-SL MAIL EVENT 1	D	0	~	D	99	
2	DVPEN01-SL MAIL EVENT 2	T	0	~	T	99	
3	DVPEN01-SL MAIL EVENT 3	C	0	~	C	99	
4	DVPEN01-SL MAIL EVENT 4	D	1000	~	D	1099	

Recipient E-mail Address

	Event-1	Event-2	Event-3	Event-4	Mail Address
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	test@sample.com
2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

بعد از اینکه تمام تنظیمات ماژول DVPEN01-SL تکمیل شد، یک برنامه برای PLC در نرم‌افزار WPLSoft نوشته و برنامه‌ی نوشته شده برای PLC در نرم‌افزار WPLSoft را کامپایل کرده و به PLC دانلود نمایید.



اگر لبه‌ی بالارونده‌ی ورودی X0 توسط PLC MPU آشکارسازی شود، یعنی ورودی X0 از Off به On تغییر وضعیت داده است. در این حالت، مقدار 1 در داخل رجیستر CR#3 از ماژول DVPEN01-SL نوشته شده و یک ایمیل به خارج ارسال خواهد شد.

اگر لبه‌ی پایین‌رونده‌ی ورودی X0 توسط PLC MPU آشکارسازی شود، یعنی ورودی X0 از On به Off تغییر وضعیت داده است. در این حالت، مقدار 1 در داخل رجیستر CR#4 از ماژول DVPEN01-SL نوشته شده و یک ایمیل به خارج ارسال خواهد شد.

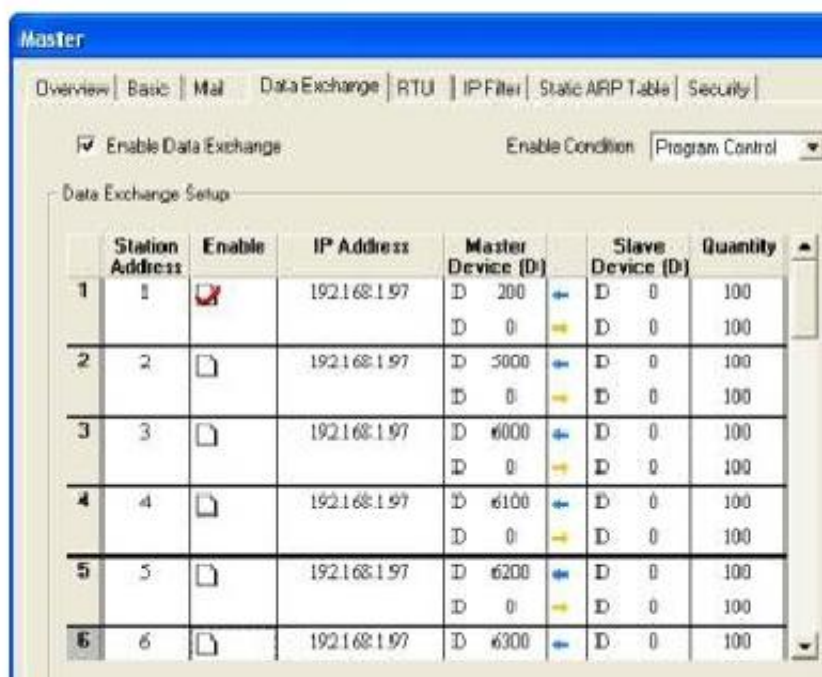
اگر لبه‌ی بالارونده‌ی خروجی Y0 توسط PLC MPU آشکارسازی شود، یعنی خروجی Y0 از Off به On تغییر وضعیت داده است. در این حالت، مقدار 1 در داخل رجیستر CR#5 از ماژول DVPEN01-SL نوشته شده و یک ایمیل به خارج ارسال خواهد شد.

اگر لبه‌ی پایین‌رونده‌ی خروجی Y0 توسط PLC MPU آشکارسازی شود، یعنی خروجی Y0 از On به Off تغییر وضعیت داده است. در این حالت، مقدار 1 در داخل رجیستر CR#6 از ماژول DVPEN01-SL نوشته می‌شود و یک ایمیل به خارج ارسال خواهد شد.

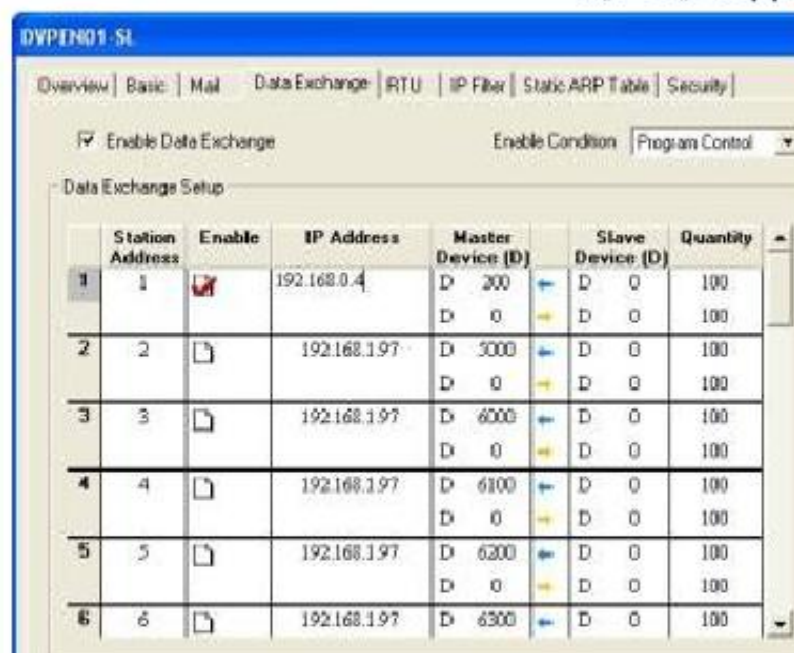
7-11-2 برنامه‌ی تبادل داده در شبکه توسط ماژول DVPEN01-SL

در این مثال، می‌خواهیم نحوه‌ی نوشتن زمان RTC را در PLC_B در رجیسترهای D0~D6 از PLC_A را بررسی نماییم. در اینجا، ما از static IP address بهره برده‌ایم. IP آدرس PLC_A برابر با 192.168.0.4 و IP آدرس PLC_B برابر با 192.168.0.5 تنظیم شده است. به روزرسانی از PLC_B به PLC_A صورت خواهد گرفت.

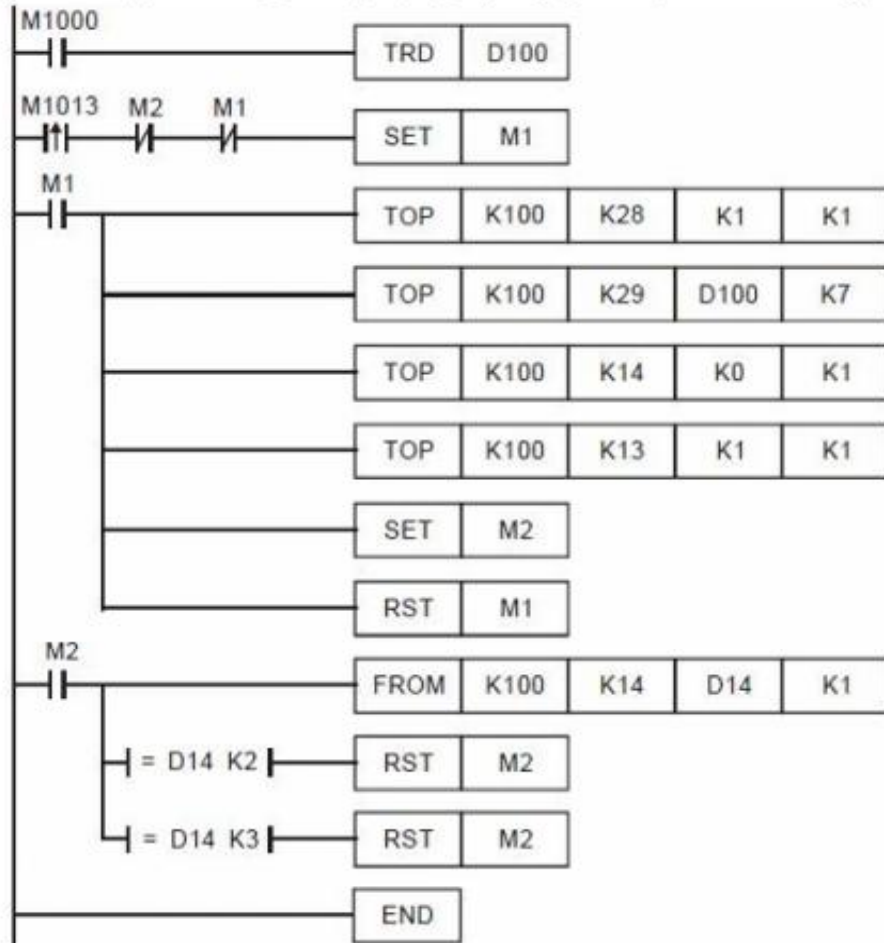
بعد از این که بین ماژول متصل شده به PLC_B و کامپیوتر ارتباط برقرار کردید، باید در نرم‌افزار DCI Soft پنجره تنظیمات ماژول DVPEN01-SL را باز کرده و تب Data Exchange را انتخاب نمایید.



در این مرحله، تیک چک باکس **Enable Data Exchange** را بزنید. در باکس کشویی **Enable Condition** آیتم **Program Control** را انتخاب نمایید. در فیلد **IP address** از سطر **station address 1** باید IP آدرس **PLC_A** را برابر با **"192.168.0.4"** وارد نمایید. سپس بر روی دکمه‌ی **Apply** کلیک کنید.



بعد از اینکه تمام تنظیمات ماژول DVPEN01-SL برای PLC_B تکمیل شد، یک برنامه برای PLC در نرم افزار WPLSoft نوشته، آن را کامپایل کرده و به PLC_B دانلود نمایید.

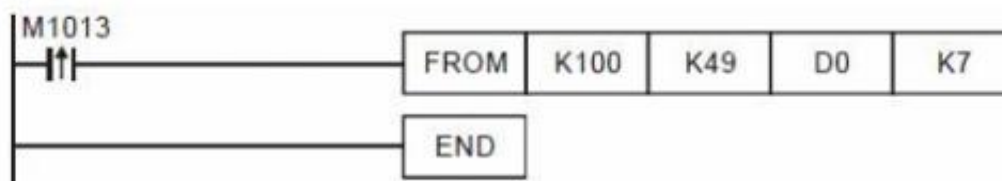


در هر ثانیه یک بار عمل تبادل دیتا بین PLC ها اجرا خواهد شد.

آدرس ارتباطات نوشته شده برای PLC مقصد در رجیستر CR#28 ماژول DVPEN01-SL قرار خواهد گرفت. ماژول DVPEN01-SL به طور اتوماتیک و توسط تنظیمات قبلی نخستین No. 1 IP address را که برابر با "192.168.0.4" می باشد را شناسایی خواهد کرد. نوشتن دیتا برای RTC در داخل رجیسترهای CR#29~CR#35 از همین ماژول صورت خواهد گرفت.

برای شروع تبادل داده باید مقدار 1 را در داخل رجیستر CR#13 نوشت. CR#14=2 به این معنی است که تبادل داده با موفقیت صورت گرفته است و CR#14=3 به معنای آن است که خطای در عمل تبادل دیتا رخ داده است.

بعد از اینکه تمام تنظیمات ماژول DVPEN01-SL برای PLC_A تکمیل شد، یک برنامه برای PLC در نرم‌افزار WPLSoft نوشته، آن را کامپایل کرده و به PLC_A دانلود نمایید.



دیتای دریافت شده در رجیسترهای CR#49~CR#55 ماژول DVPEN01-SL ذخیره خواهد شد.

دیتای دریافت شده هر یک ثانیه یکبار در داخل رجیسترهای D0~D6 نوشته می‌شوند.

3-11-7 دومین برنامه‌ی تبادل داده در شبکه توسط ماژول DVPEN01-SL

در این مثال، می‌خواهیم نحوه‌ی فعال کردن یک تایمر و نوشتن مقادیر تایمر در رجیسترهای D0~D99 را بررسی نماییم. همچنین، مقادیر جاری D0~D99 از PLC_A را به طور پیوسته در داخل رجیسترهای D0~D99 از PLC_B و مقادیر داخل D0~D99 از PLC_B را داخل رجیسترهای D200~D299 از PLC_A بنویسیم. در اینجا، ما از static IP address بهره برده‌ایم. IP آدرس PLC_A برابر با 192.168.1.99 تنظیم شده و IP آدرس PLC_B برابر با 192.168.1.97 تنظیم شده است. به روزرسانی از PLC_A به PLC_B و از PLC_B به PLC_A صورت خواهد گرفت.

بعد از این که بین ماژول متصل شده به PLC_A و کامپیوتر ارتباط برقرار کردید، باید در نرم‌افزار DCI Soft پنجره تنظیمات ماژول DVPEN01-SL را باز کرده و تب Data Exchange را انتخاب نمایید.

DVP1N01-SL

Overview | Basic | Mail | Data Exchange | RTU | IP Filter | Static ARP Table | Security

☒ Enable Data Exchange Enable Condition: **Always Enable**

Data Exchange Setup

	Station Address	Enable	IP Address	Master Device (D)		Slave Device (D)	Quantity
1	1		192.168.1.97	D 200	↔	D 0	100
				D 0	→	D 0	100
2	2		192.168.1.97	D 5000	↔	D 0	100
				D 0	→	D 0	100
3	3		192.168.1.97	D 6000	↔	D 0	100
				D 0	→	D 0	100
4	4		192.168.1.97	D 6100	↔	D 0	100
				D 0	→	D 0	100
5	5		192.168.1.97	D 6200	↔	D 0	100
				D 0	→	D 0	100
6	6		192.168.1.97	D 6300	↔	D 0	100

در این مرحله، تیک چک باکس Enable Data Exchange را بزنید. در باکس کشویی Condition آیتم Always Enable را انتخاب نمایید. در سطر اول station address 1 تیک Enable را بزنید. در فیلد IP address از سطر 1 station address باید IP آدرس PLC_B را برابر با "192.168.1.97" وارد نمایید. سپس بر روی دکمه‌ی Apply کلیک نمایید. در فیلد Master Device باید رجیستر D200 و D0 و در فیلد Slave Device باید رجیستر D0 و D0 را انتخاب نمایید. در فیلد Quantity باید 100 و 100 را انتخاب کنید.

DVP1N01-SL

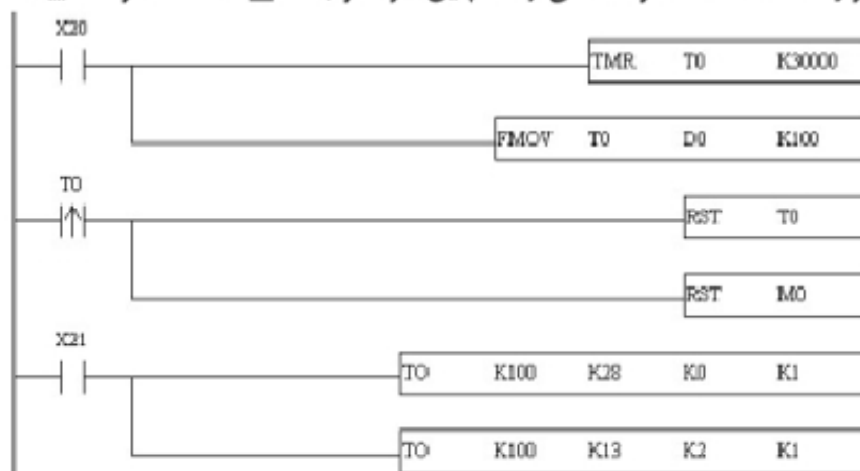
Overview | Basic | Mail | Data Exchange | RTU | IP Filter | Static ARP Table | Security

☒ Enable Data Exchange Enable Condition: **Always Enable**

Data Exchange Setup

	Station Address	Enable	IP Address	Master Device (D)		Slave Device (D)	Quantity
1	1		192.168.1.97	D 200	↔	D 0	100
				D 0	→	D 0	100
2	2		192.168.1.97	D 5000	↔	D 0	100
				D 0	→	D 0	100
3	3		192.168.1.97	D 6000	↔	D 0	100
				D 0	→	D 0	100
4	4		192.168.1.97	D 6100	↔	D 0	100
				D 0	→	D 0	100
5	5		192.168.1.97	D 6200	↔	D 0	100
				D 0	→	D 0	100
6	6		192.168.1.97	D 6300	↔	D 0	100

بعد از اینکه تمام تنظیمات ماژول DVPEN01-SL برای PLC_A تکمیل شد، یک برنامه برای PLC در نرم‌افزار WPLSoft نوشته، آن را کامپایل کرده و به PLC_B دانلود نمایید.



7-11-4 سومین برنامه‌ی تبادل داده در شبکه توسط ماژول DVPEN01-SL

در این مثال، می‌خواهیم نحوه‌ی فعال کردن یک تایمر با ورودی X20 و نوشتن مقادیر تایمر در رجیسترهای D0~D99 را بررسی نماییم. همچنین، برنامه را با ورودی X21 کنترل کرده و مقادیر جاری D0~D99 از PLC_A را در داخل رجیسترهای D0~D99 از PLC_B و مقادیر داخل D0~D99 از PLC_B را داخل رجیسترهای D200~D299 از PLC_A بنویسیم. زمانی که ورودی X21 غیرفعال شد، اجرای برنامه متوقف شود. در اینجا، ما از static IP address بهره برده‌ایم. IP آدرس PLC_A برابر با 192.168.1.99 تنظیم شده و IP آدرس PLC_B برابر با 192.168.1.97 تنظیم شده است. به روزرسانی از PLC_A به PLC_B و از PLC_B به PLC_A صورت خواهد گرفت.

بعد از این که بین ماژول متصل شده به PLC_A و کامپیوتر ارتباط برقرار کردید، باید در نرم‌افزار DCI Soft پنجره تنظیمات ماژول DVPEN01-SL را باز کرده و تب Data Exchange را انتخاب نمایید.

Master

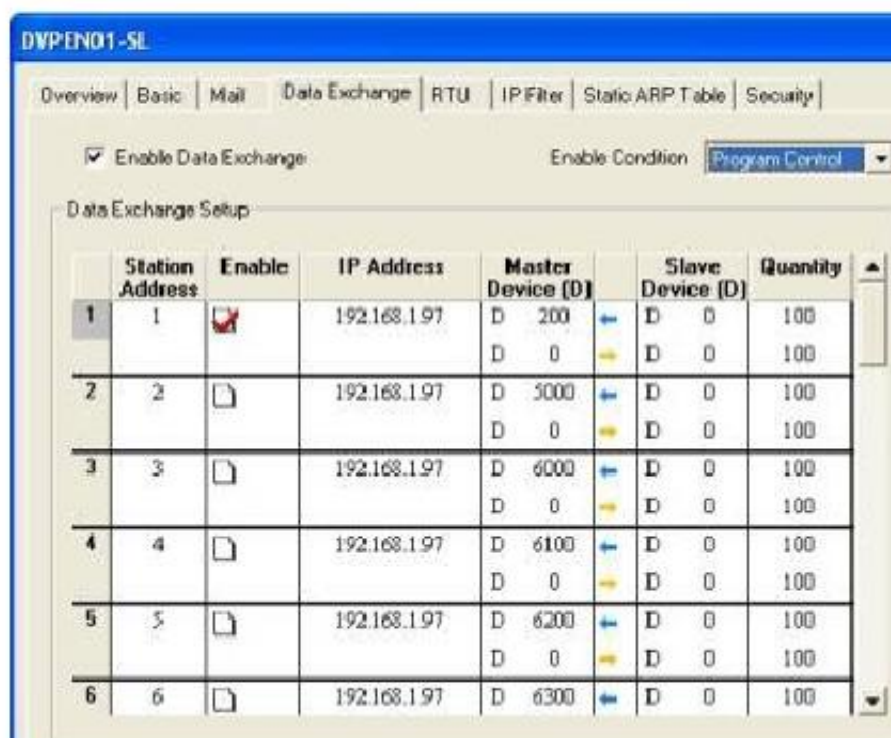
Overview | Basic | Mail | Data Exchange | RTU | IP Filter | Static ARP Table | Security

☒ Enable Data Exchange Enable Condition: Program Control

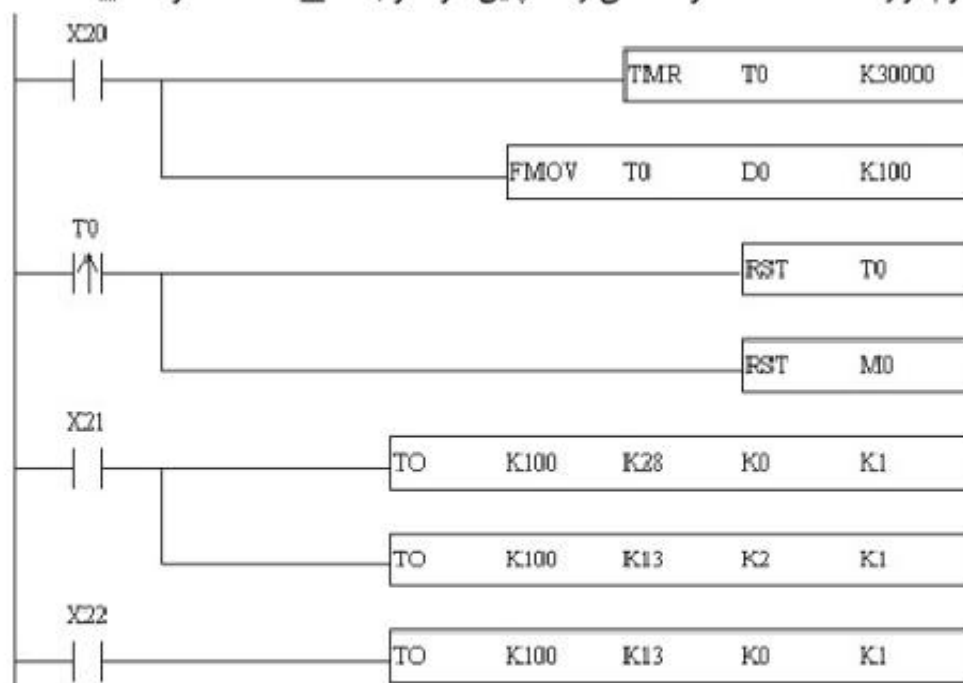
Data Exchange Setup

	Station Address	Enable	IP Address	Master Device (D)		Slave Device (D)	Quantity
1	1	<input checked="" type="checkbox"/>	192.168.1.97	D 200	←	D 0	100
				D 0	→	D 0	100
2	2	<input type="checkbox"/>	192.168.1.97	D 5000	←	D 0	100
				D 0	→	D 0	100
3	3	<input type="checkbox"/>	192.168.1.97	D 6000	←	D 0	100
				D 0	→	D 0	100
4	4	<input type="checkbox"/>	192.168.1.97	D 6100	←	D 0	100
				D 0	→	D 0	100
5	5	<input type="checkbox"/>	192.168.1.97	D 6200	←	D 0	100
				D 0	→	D 0	100
6	6	<input type="checkbox"/>	192.168.1.97	D 6300	←	D 0	100

در این مرحله، تیک چک باکس Enable Data Exchange را بزنید. در باکس کشویی Enable Condition آیتم Program Control را انتخاب نمایید. در سطر اول station address 1 تیک فیلد Enable را بزنید. در فیلد IP address از سطر 1 station باید IP آدرس PLC_B را برابر با "192.168.1.97" وارد نمایید. سپس بر روی دکمه‌ی Apply کلیک نمایید. در فیلد Master Device باید رجیستر D200 و D0 و در فیلد Slave Device باید رجیستر D0 و D0 را انتخاب نمایید. در فیلد Quantity باید 100 و 100 را انتخاب کنید.

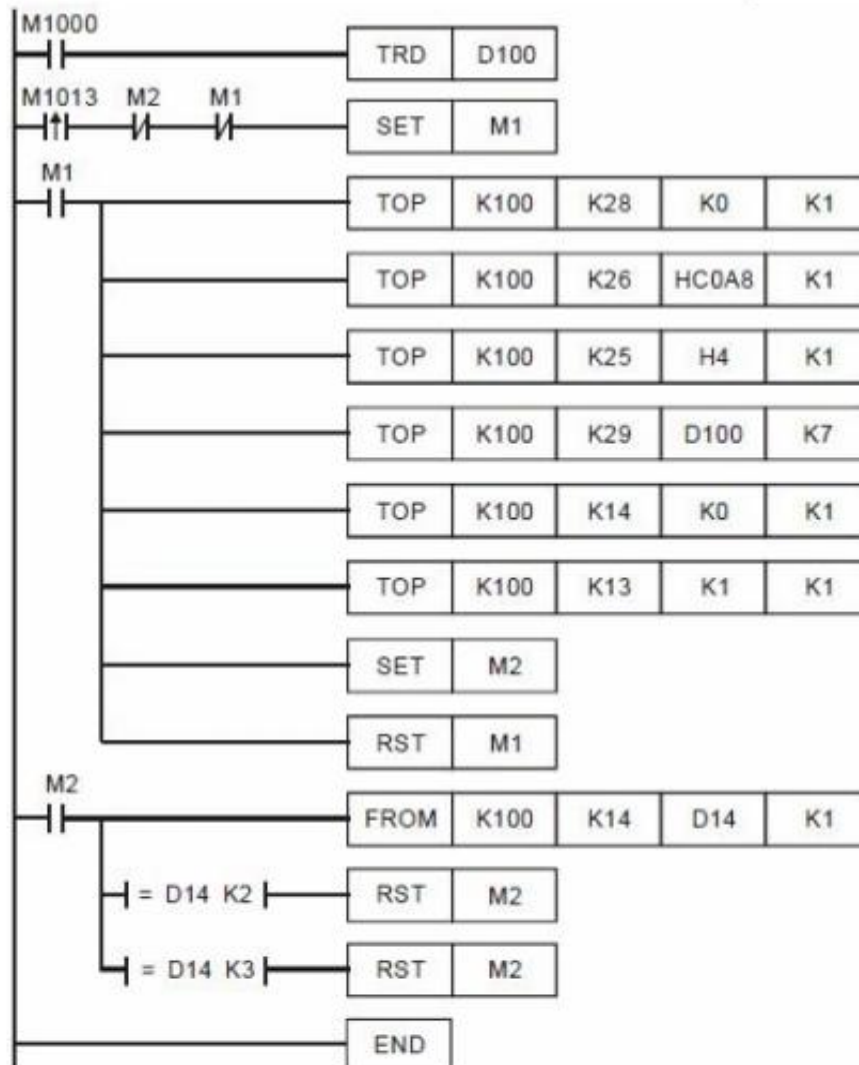


بعد از اینکه تمام تنظیمات مازول DVPEN01-SL برای PLC_A تکمیل شد، یک برنامه برای PLC در نرم‌افزار WPLSoft نوشته، آن را کامپایل کرده و به PLC_B دانلود نمایید.



7-11-5 چهارمین برنامه‌ی تبادل داده در شبکه توسط ماژول DVPEN01-SL

در این مثال، می‌خواهیم نحوه‌ی نوشتن زمان RTC در PLC_B را در رجیسترهای D0~D6 از PLC_A را بررسی نماییم. همچنین، برای اختصاص دادن IP address از برنامه‌ی نردبانی در نرم‌افزار WPLSoft استفاده نماییم. در اینجا ما از static IP address بهره برده‌ایم. IP آدرس PLC_A برابر با 192.168.0.4 و IP آدرس PLC_B برابر با 192.168.0.5 تنظیم شده است. به روزرسانی از PLC_B به PLC_A صورت خواهد گرفت. بعد از این که بین ماژول متصل شده به PLC_B و کامپیوتر ارتباط برقرار کردید، باید یک برنامه به زبان دیاگرام نردبانی بنویسید. برنامه‌ی نوشته شده برای PLC در نرم‌افزار WPLSoft را کامپایل کرده و به PLC_B دانلود نمایید.



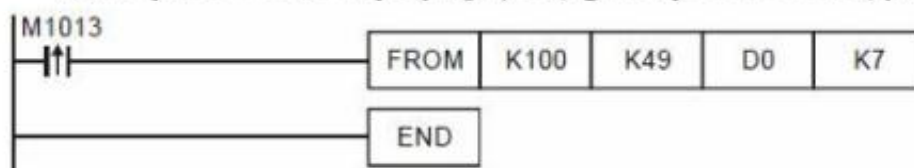
در هر ثانیه یک بار عمل تبادل دیتا بین PLC ها اجرا خواهد شد.

مقدار 0 در رجیستر CR#28 مازول DVPEN01-SL نوشته خواهد شد و PLC_B از رجیسترهای CR#25~CR#26 به عنوان IP address برای PLC مقصد استفاده می‌کند. IP address برای PLC_A در داخل رجیستر CR#25 و CR#26 نوشته می‌شود. همانطور که گفته شد، IP آدرس از چهار بخش تشکیل شده که در این برنامه دو بخش اول آن با مقدار (192.168 = H' C0A8) در داخل رجیستر CR#26 و دو بخش دوم آن (0.4 = H' 0004) در رجیستر CR#25 نوشته می‌شود.

نوشتن دیتا برای RTC در داخل رجیسترهای CR#29~CR#35 از همین مازول صورت خواهد گرفت.

برای شروع تبادل داده باید مقدار 1 را در داخل رجیستر CR#13 نوشت. CR#14=2 به این معنی است که تبادل داده با موفقیت صورت گرفته است و CR#14=3 به این معنی است که خطایی در عمل تبادل دیتا رخ داده است.

بعد از اینکه تمام تنظیمات مازول DVPEN01-SL برای PLC_A تکمیل شد، یک برنامه برای PLC در نرم‌افزار WPLSoft نوشته، آن را کامپایل کرده و به PLC_A دانلود نمایید.



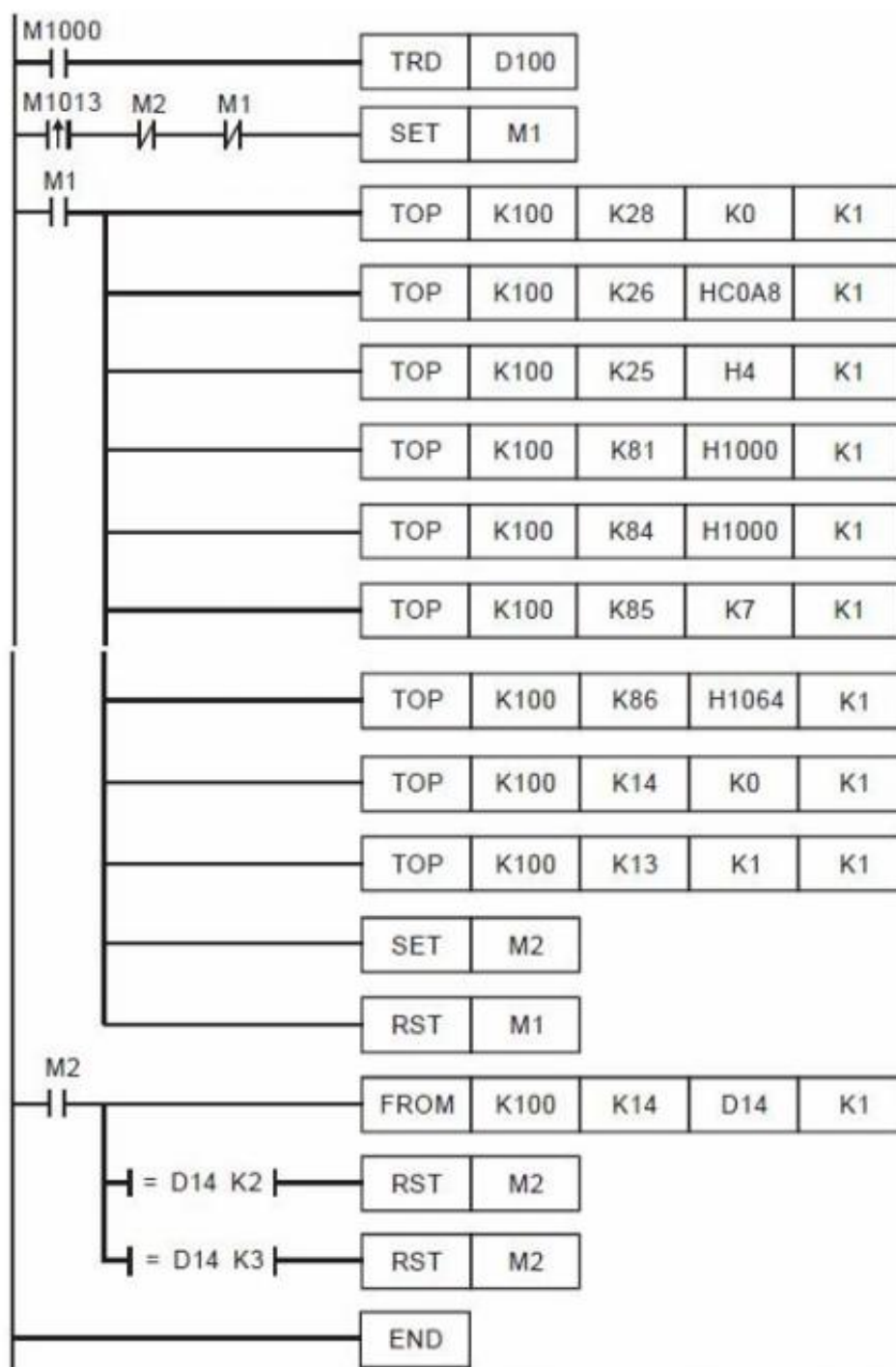
دیتای دریافت شده در رجیسترهای CR#49~CR#55 مازول DVPEN01-SL ذخیره خواهد شد.

دیتای دریافت شده هر یک ثانیه یکبار در داخل رجیسترهای D0~D6 نوشته می‌شوند.

7-11-6 پنجمین برنامه‌ی تبادل داده در شبکه توسط مازول DVPEN01-SL

در این مثال، می‌خواهیم نحوه‌ی نوشتن زمان RTC در PLC_B را مستقیماً در داخل رجیسترهای D0~D6 از PLC_A بدون نوشتن برنامه‌ی دیاگرام نردبانی در PLC_A را بررسی نماییم. در اینجا ما از static IP address بهره برده‌ایم. IP آدرس PLC_A برابر با 192.168.0.4 و IP آدرس PLC_B برابر با 192.168.0.5 تنظیم شده است. به روزرسانی از PLC_B به PLC_A صورت خواهد گرفت.

بعد از این که بین مازول متصل شده به PLC_B و کامپیوتر ارتباط برقرار کردید، باید یک برنامه به زبان دیاگرام نردبانی بنویسید. برنامه‌ی نوشته شده برای PLC در نرم‌افزار WPLSoft را کامپایل کرده و به PLC_B دانلود نمایید.



در هر ثانیه یک بار عمل تبادل دیتا بین PLC ها اجرا خواهد شد.
 مقدار 0 در رجیستر CR#28 مازول DVPEN01-SL نوشته خواهد شد و PLC_B از
 رجیسترهای CR#25~CR#26 به عنوان IP address برای PLC مقصد استفاده می‌کند.

IP address برای PLC_A در داخل رجیستر CR#25 و CR#26 نوشته می‌شود. همانطور که گفته شد، IP آدرس از چهار بخش تشکیل شده که در این برنامه دو بخش اول آن با مقدار $(192.168 = H' C0A8)$ در داخل رجیستر CR#26 و دو بخش دوم آن $(0.4 = H' 0004)$ در رجیستر CR#25 نوشته می‌شود.

آدرس MODBUS رجیستر D0 ($H' 1000$) برای PLC_A را باید در داخل رجیستر CR#81 و CR#84 مازول DVPEN01-SL نوشت.

آدرس MODBUS رجیستر D100 (register of RTC) ($H' 1064$) برای PLC_A را باید در داخل رجیستر CR#86 مازول DVPEN01-SL نوشت.

تعداد رجیسترها را باید به صورت K7 در رجیستر CR#85 مازول DVPEN01-SL نوشت.

برای شروع تبادل داده باید مقدار 1 را در داخل رجیستر CR#13 نوشت.

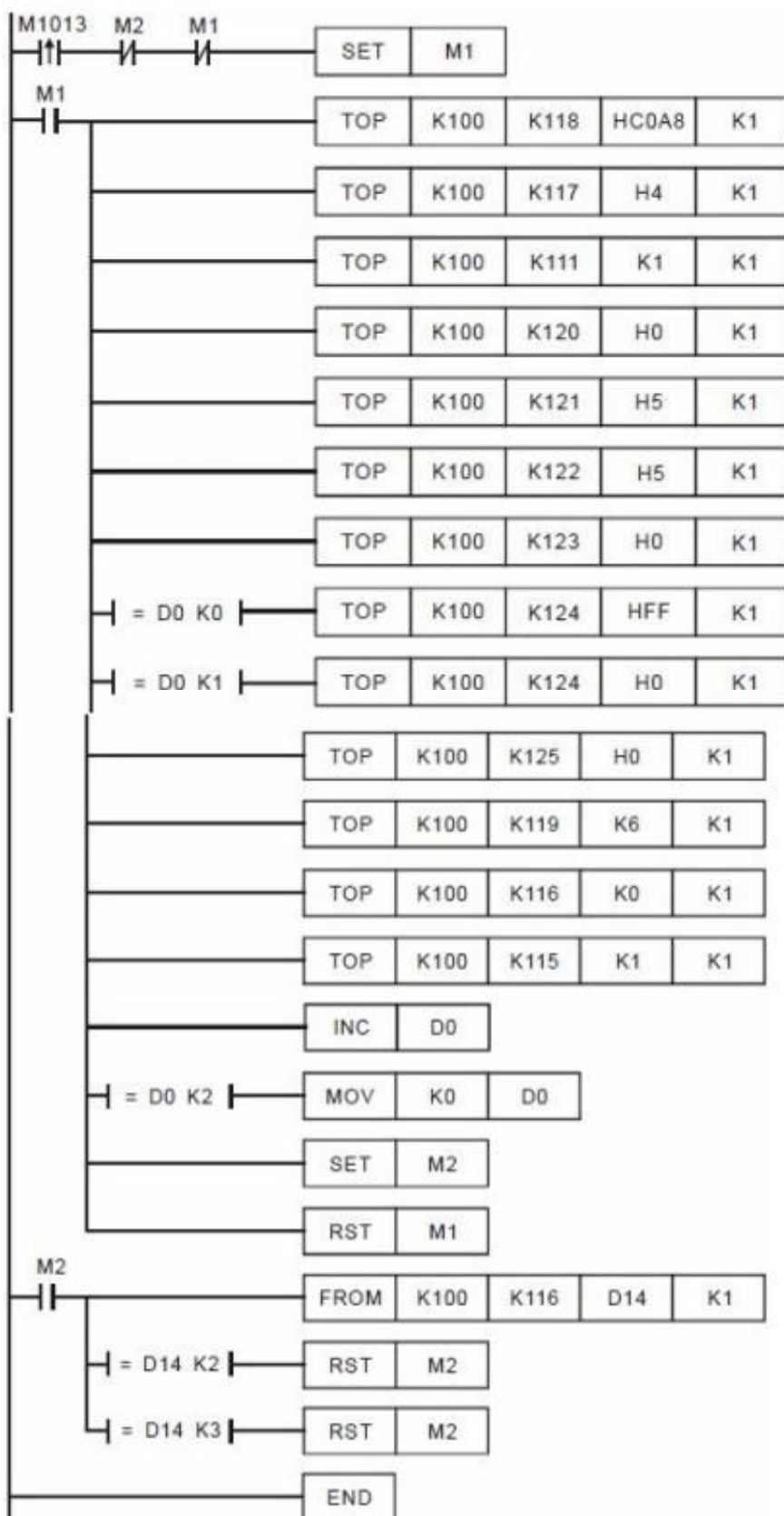
CR#14=2 به این معنی است که تبادل داده با موفقیت صورت گرفته است و CR#14=3 به این معنی است که خطایی در عمل تبادل دیتا رخ داده است.

تبادل دیتا یک مرتبه صورت گرفته و اگر موفقیت‌آمیز باشد، مقادیر موجود در رجیسترهای D1313~D1318 از PLC_B در داخل رجیسترهای D0~D6 از PLC_A ریخته خواهد شد.

7-11-7 راه‌اندازی MODBUS TCP Master

در این مثال، می‌خواهیم با استفاده از دستورالعمل MODBUS در PLC_B خروجی Y0 در PLC_A را به صورت چشمک‌زن راه‌اندازی نماییم. در اینجا، ما از static IP address بهره برده‌ایم. IP آدرس PLC_A برابر با 192.168.0.4 و IP آدرس PLC_B برابر با 192.168.0.5 تنظیم شده است. به روزرسانی از PLC_B به PLC_A صورت خواهد گرفت. توسط دستورالعمل MODBUS با مقدار 050500FF00 می‌خواهیم خروجی Y0 را فعال "On" نماییم. با استفاده از دستورالعمل MODBUS با مقدار 0505000000 می‌خواهیم خروجی Y0 را غیرفعال "Off" نماییم. همچنین، عمل On/Off هر یک ثانیه یک بار تکرار شود.

بعد از این که بین مازول متصل شده به PLC_B و کامپیوتر ارتباط برقرار کردید، باید یک برنامه به زبان دیاگرام نردبانی بنویسید. برنامه‌ی نوشته شده برای PLC در نرم‌افزار WPLSoft را کامپایل کرده و به PLC_B دانلود نمایید. در این مثال، نیازی به نوشتن برنامه‌ی نردبانی برای PLC_A نمی‌باشد.



در هر ثانیه یک بار عمل تبادل دیتا بین PLC ها اجرا خواهد شد.
 IP address برای PLC_A در داخل رجیستر CR#117 و CR#118 نوشته می‌شود.
 همانطور که گفته شد، IP آدرس از چهار بخش تشکیل شده که در این برنامه دو بخش اول آن با مقدار (192.168 = H' C0A8) در داخل رجیستر CR#118 و دو بخش دوم آن (0.4 = H' 0004) در رجیستر CR#117 نوشته می‌شود.

مقدار رجیستر CR#111 را برای فعال کردن مد 8 بیتی باید برابر با 1 تنظیم کرد. دستورالعمل MODBUS مقادیر را در بایت‌های کم‌ارزش CR#120~CR#247 ذخیره خواهد کرد.
 دستورالعمل MODBUS در رجیسترهای CR#120~CR#245 نوشته می‌شوند. در داخل رجیستر CR#120 آدرس MODBUS قرار دارد.
 طول دیتا برای دستورالعمل MODBUS در داخل رجیستر CR#119 نوشته می‌شود.
 با نوشتن مقدار 1 در داخل رجیستر CR#115 دستورالعمل MODBUS TCP شروع به اجرا خواهد کرد.

CR#116=2 به این معنی است که دستورالعمل MODBUS با موفقیت اجرا شده است و
 CR#116=3 به این معنی است که خطایی در اجرای دستورالعمل MODBUS رخ داده است.
 اگر دستورالعمل MODBUS با موفقیت اجرا شود، خروجی Y0 در PLC_A هر یک ثانیه یکبار روشن و خاموش خواهد شد.

7-11-8 مثال از RTU Mapping

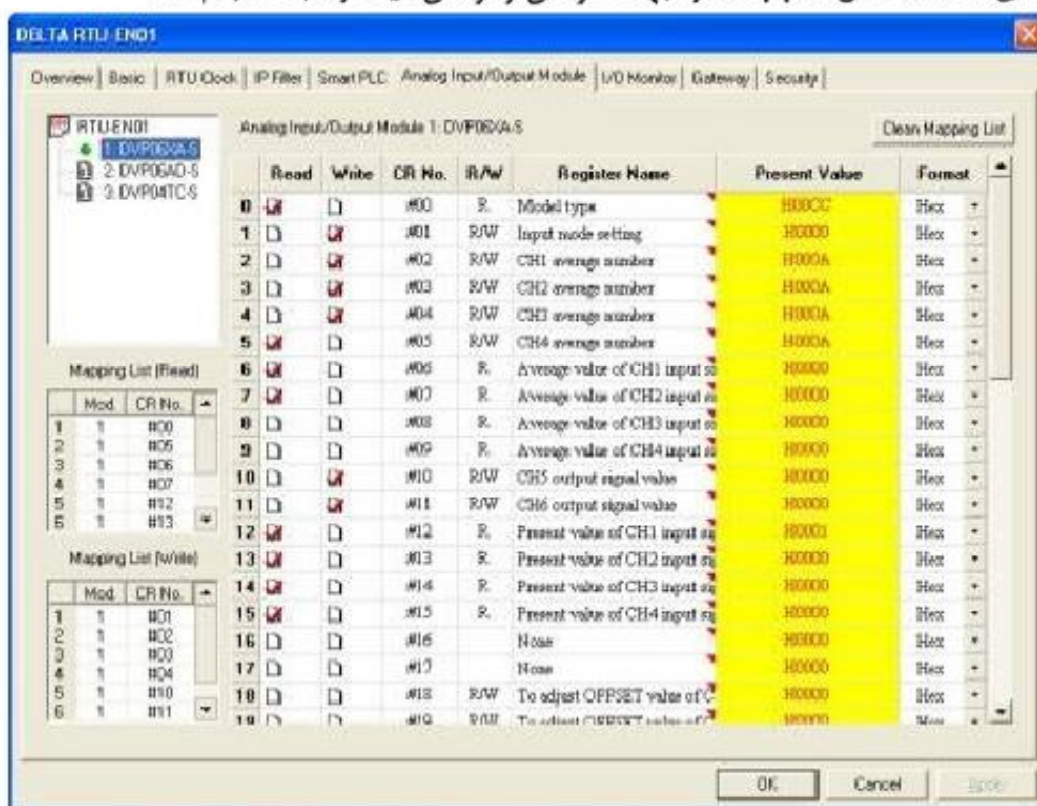
در این مثال، می‌خواهیم با RTU Mapping مقادیری را بر روی رجیسترهای ورودی/خروجی آنالوگ و دیجیتال ریموت نوشته و یا مقادیر را بخوانیم.

DVP28SV+DVPEN01-SL → RTU-EN01+DVP06XA+DVP16SP
 در اینجا، ما از static IP address بهره برده‌ایم. IP آدرس DVPEN01-SL برابر با 192.168.190 تنظیم شده و IP آدرس RTU-EN01 برابر با 192.168.191 تنظیم شده است. با استفاده از نرم‌افزار DCI Soft تنظیمات ماژول RTU-EN01 را انجام داده و 10 تا دیتای مسیره‌دهی شده را برای خواندن مقدارشان و 10 تا دیتای مسیره‌دهی شده را برای نوشتن مقدار در داخلشان بررسی می‌کنیم.

آدرس شروع مسیره‌دهی حافظه "mapping" را تنظیم کرده و تعداد دیتای RX, RY, RCR جهت خواندن مقدارشان و RCR جهت نوشتن مقدار در داخلش را توسط DVPEN01-SL انجام می‌دهیم.

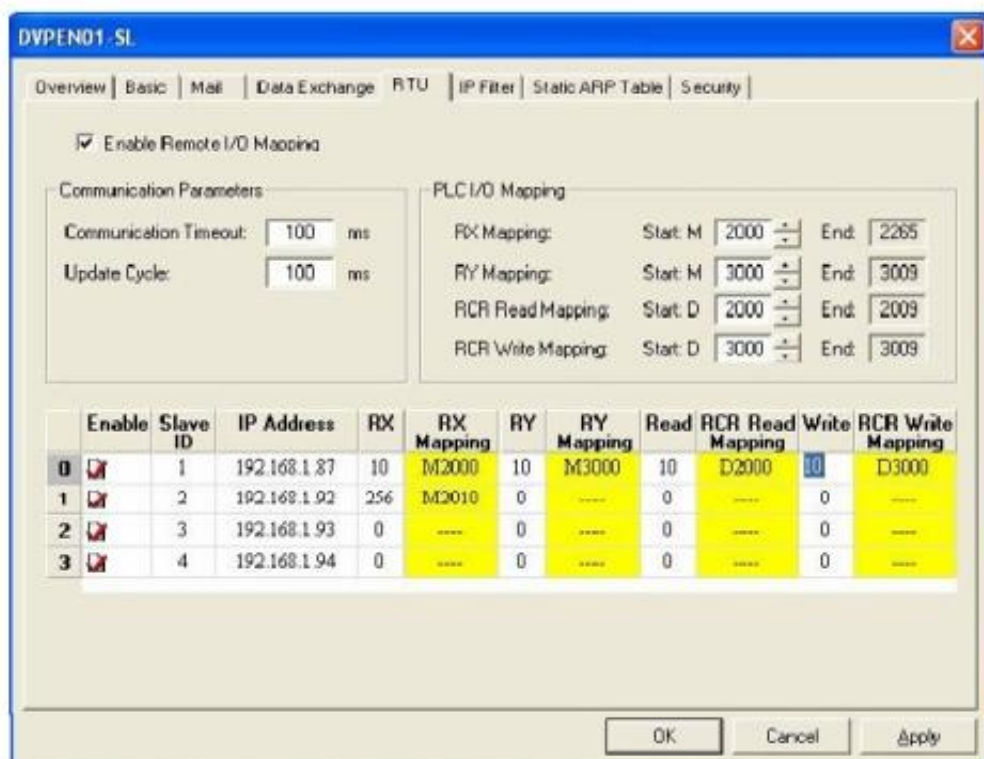
تابع mapping در PLC DVP-SV را توسط DVPEN01-SL باید فعال کرد. از فلگ خاص M2000 و رجیستر خاص D2000 در DVP-SV PLC برای خواندن دیتا و از فلگ خاص M3000 و رجیستر خاص D3000 برای نوشتن مقدار در RTU-EN01 ریموت استفاده می‌کنیم. بعد از این که بین ماژول متصل شده به PLC و کامپیوتر ارتباط برقرار کردید، باید مراحل زیر را طی کنید:

مرحله 1 ← با استفاده از نرم‌افزار DCI Soft باید تنظیمات ماژول RTU-EN01 رجیسترهای mapping control را جهت خواندن و نوشتن دیتا در شبکه انجام داد.

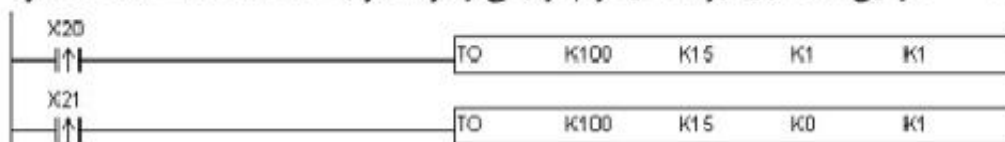


مرحله 2 ← با استفاده از نرم‌افزار DCI Soft باید تنظیمات ماژول DVPEN01-SL را برای تنظیم آدرس شروع و تعداد دیتا انجام داد.

(RX: M2000~M2009; RY: M3000~M3009; RCR (Reading): D2000~D2009; RCR (Writing): D3000~D3009)



مرحله‌ی 3 ← در این مرحله باید برنامه دیاگرام نردبانی را نوشته و به DVPEN01-SL دانلود کرد:



با نوشتن مقدار 1 در داخل رجیستر CR#15 باید تابع mapping را فعال کرد.

با نوشتن مقدار 0 در داخل رجیستر CR#15 می‌توان تابع mapping را غیرفعال کرد.

بعد از اینکه برای فعال کردن تابع mapping مقدار 1 را در داخل رجیستر CR#15 نوشتید، از فلگ‌های M2000~M2009 و رجیسترهای D2000~D2009 برای خواندن دیتا استفاده کرده و مقادیر جاری قبل از نوشتن دیتا از فلگ‌های M3000~M3009 و رجیسترهای D3000~D3009 خوانده خواهند شد.

در هنگام اجرای تابع mapping، دیگر تجهیزات توانایی اصلاح مقادیر رجیسترهای mapping را ندارند.

7-11-9 مثالی از کاربرد پروتکل MELSEC

در این مثال، باید شرط اجرا را در لیست باکس Enable Condition برابر با Always Enable تنظیم نمائید تا بتوان رجیسترهای PLC میتسویشی را توسط PLC دلتا خوانده و یا مقادیری را در رجیسترهای PLC میتسویشی نوشت.

DVP28SV+DVPEN01-SL → Mitsubishi PLC

در اینجا، ما از static IP address بهره برده‌ایم. IP آدرس DVPEN01-SL برابر با 192.168.15 و پورت ارتباطات برای ارسال دیتا برابر با 9002 تنظیم شده و IP آدرس -RTU EN01 برابر با 192.168.139 و پورت ارتباطات برای دریافت دیتا برابر با 9002 تنظیم شده است. مسیریابی دیتا در این مثال به شرح زیر است:

رجیسترهای D100~D199 در DVP28SV در داخل رجیسترهای D100~D199 از Mitsubishi PLC و رجیسترهای D0~D99 از Mitsubishi PLC در داخل رجیسترهای D0~D99 از DVP28SV مسیریابی شده‌اند.

برای اطلاع از نحوه‌ی برنامه‌نویسی Mitsubishi PLC به کتاب چاپ شده توسط انتشارات قدیس (همین انتشارات) و برای اطلاع از پروتکل ارتباطات MELSEC به وب سایت میتسویشی مراجعه نمایید.

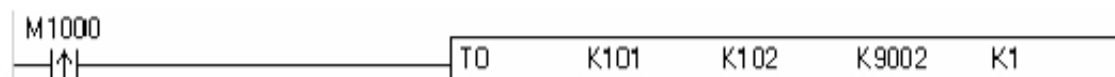
مرحله‌ی 1 ← با استفاده از نرم‌افزار میتسویشی باید تنظیمات IP address را برای Mitsubishi PLC انجام دهید و پارامترهای مربوط به ارتباطات را که در زیر لیست شده‌اند را انجام دهید:

Communication protocol: MC protocol

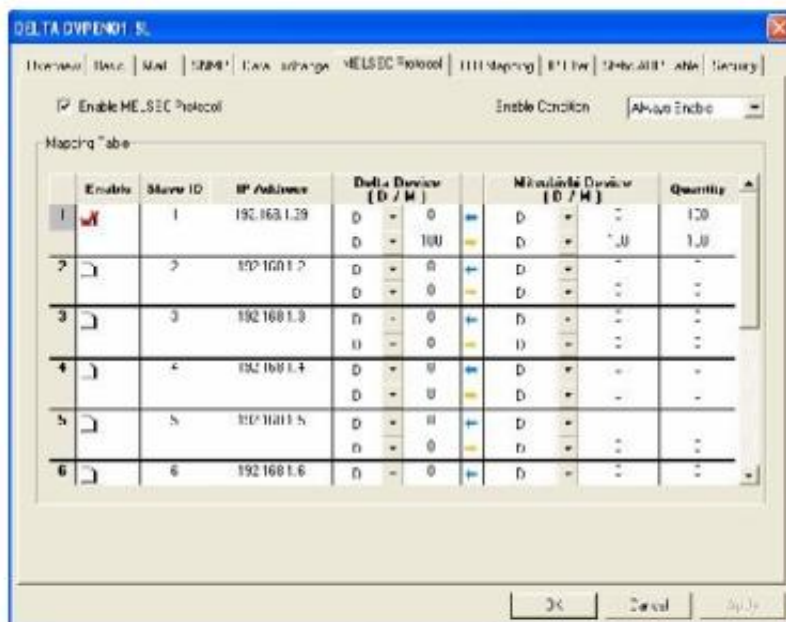
Communication mode: UDP

Communication port: 9002

مرحله‌ی 2 ← در این مرحله باید برنامه دیاگرام نردبانی را نوشته و به DVPEN01-SL دانلود کرد.



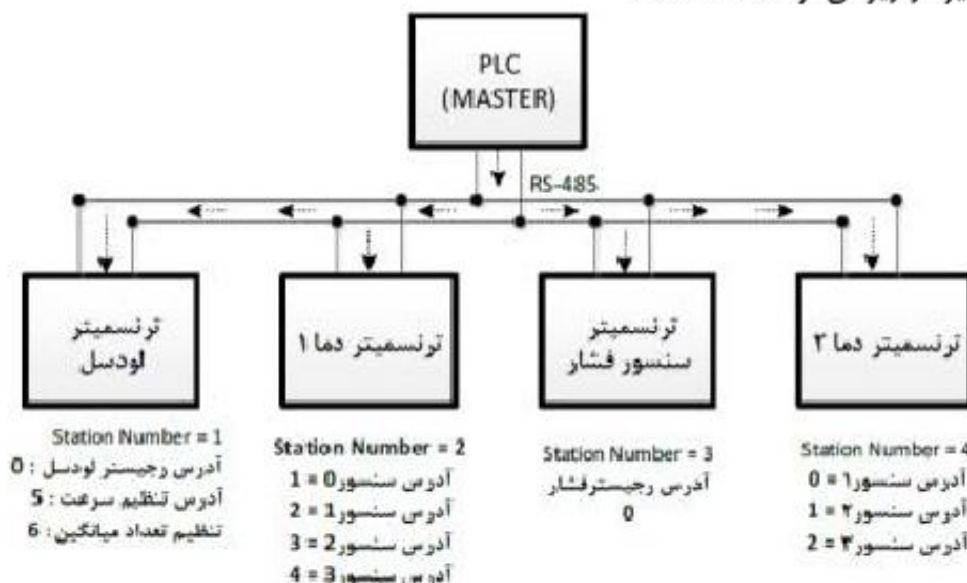
مرحله‌ی 2 ← با استفاده از نرم‌افزار DCISof تنظیمات مربوط به تبادل دیتا برای DVPEN01-SL را انجام دهید.



بعد از اینکه تنظیمات انجام شده به DVPEN01-SL دانلود شد، توسط ماژول DVPEN01-SL می‌توان دیتای موجود در رجیسترهای D0~D99 از Mitsubishi PLC را خوانده و در رجیسترهای D0~D99 از DVP28SV نوشت. همچنین، می‌توان دیتای موجود در رجیسترهای D100~D199 از Mitsubishi PLC را خوانده و در رجیسترهای D100~D199 از DVP28SV نوشت.

7- 12 برقراری ارتباط بین 4 ماژول ترنس미터 با PLC دلتا از طریق پروتکل Modbus

در این مثال، 4 ماژول ترنس미터 Modbus به باس RS-485 متصل هستند. آدرس رجیسترهای هر ماژول نیز در زیر آن نوشته شده است.



می‌خواهیم برنامه‌ای بنویسیم که با استفاده از پورت COM2 مربوط به PLC دلتا فقط از رجیستر شماره 0 مقدار لودسل را بخوانیم و مقادیر سرعت و تعداد میانگین‌گیری آن را تنظیم نماییم. سپس با استفاده از پورت COM2 مربوط به PLC، از رجیستر شماره 0 مقدار لودسل را بخوانیم و مقادیر سرعت و تعداد میانگین‌گیری آن را تنظیم نماییم و 4 رجیستر Slave2 و یک رجیستر از Slave2 و سه رجیستر از Slave3 را بخوانیم.





7-13 برقراری ارتباط بین PLC و اینورتر سری VFD-M دلتا با دستورالعمل MODRD/MODWR

در این مثال، می‌خواهیم ارتباط بین PLC دلتا و درایو موتور AC (اینورتر) سری VFD-M دلتا را با استفاده از دستورات (MODRD/MODWR) بررسی نماییم.

هدف از کنترل در این مثال به شرح زیر می‌باشد:

می‌خواهیم مکرراً فرکانس Master و فرکانس خروجی اینورتر سری VFD-M دلتا را بخوانیم و سپس آن را با استفاده از دستور MODRD در رجیسترهای D0 و D1 ذخیره کنیم. همچنین، توسط

دستورالعمل MODWR مکرراً بتوانیم جهت چرخش و فرکانس راه‌اندازی را در هنگام راه‌اندازی و همچنین، در حال کار تنظیم نماییم. برای مثال، می‌خواهیم تنظیمات درایور موتورهای AC (اینورتر) را برای راه‌اندازی برابر با چرخش در جهت راست با فرکانس 40Hz تنظیم نماییم.

برای تنظیمات پارامترهای درایو موتور AC (اینورتر) سری VFD-M دلتا به جدول زیر مراجعه نمایید :

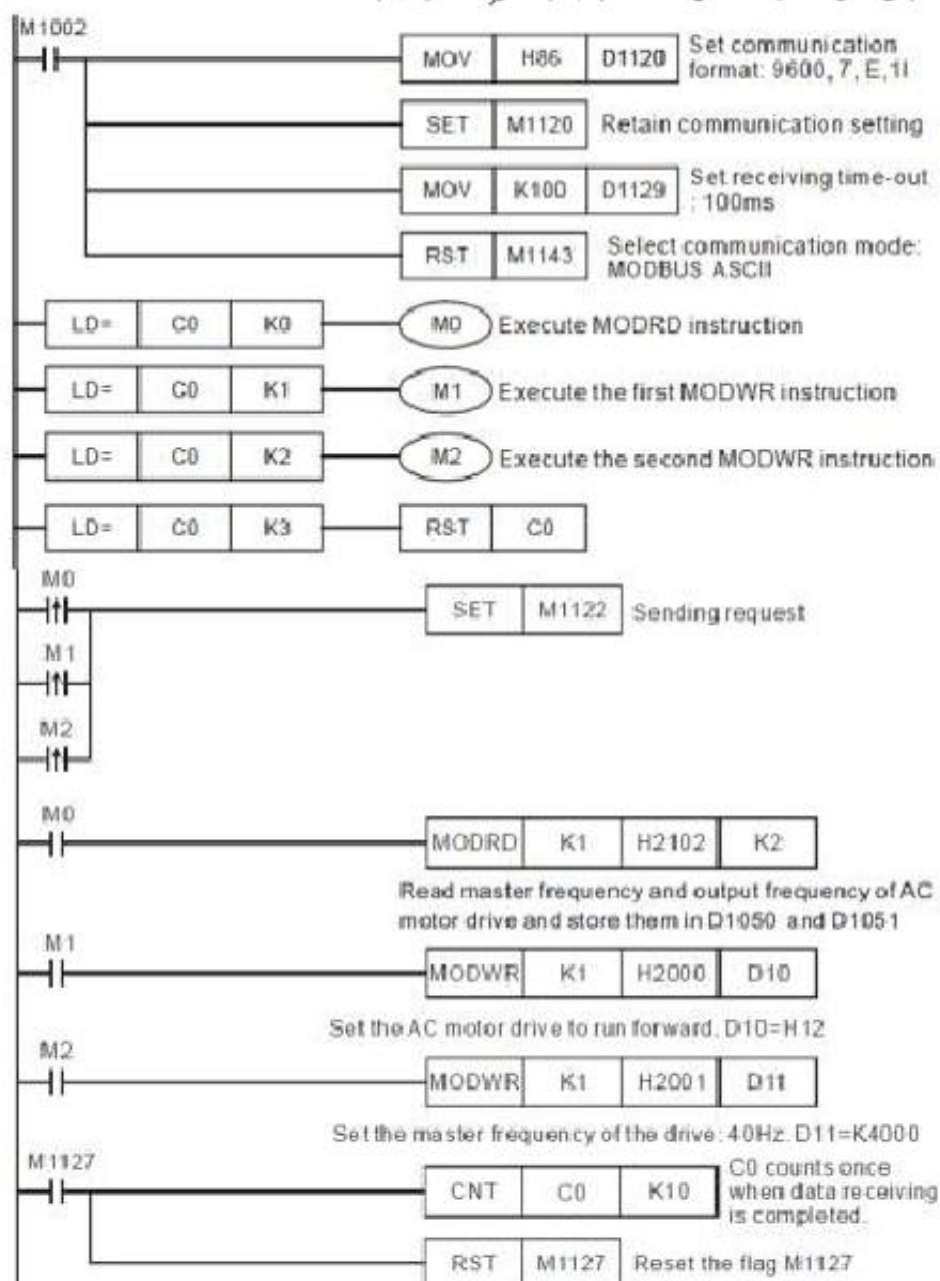
پارامتر	مقدار تنظیمی	توضیحات
P00	03	از این پارامتر برای تعیین کردن فرکانس Master توسط پورت کام RS-485 استفاده می‌شود.
P01	03	از این پارامتر برای تعیین عملکرد اینورتر توسط پورت کام RS-485 استفاده می‌شود، در این مد ابتدا STOP روی کی‌پد اینورتر فعال می‌باشد
P88	01	از این پارامتر برای تنظیم آدرس ارتباطات شبکه استفاده می‌شود آدرس ارتباطات شبکه به صورت پیش‌فرض برابر با 01 است.
P89	01	از این پارامتر برای تنظیم سرعت ارتباطات شبکه برای تبادل دیتا استفاده می‌شود. سرعت ارتباطات شبکه برای تبادل دیتا به صورت پیش‌فرض برابر با 9600 است.
P92	01	از این پارامتر برای تنظیم پروتکل ارتباطات استفاده می‌شود. پروتکل ارتباطات دیتا به صورت پیش‌فرض برابر با <7, E, 1> , MODBUS ASCII mode است.

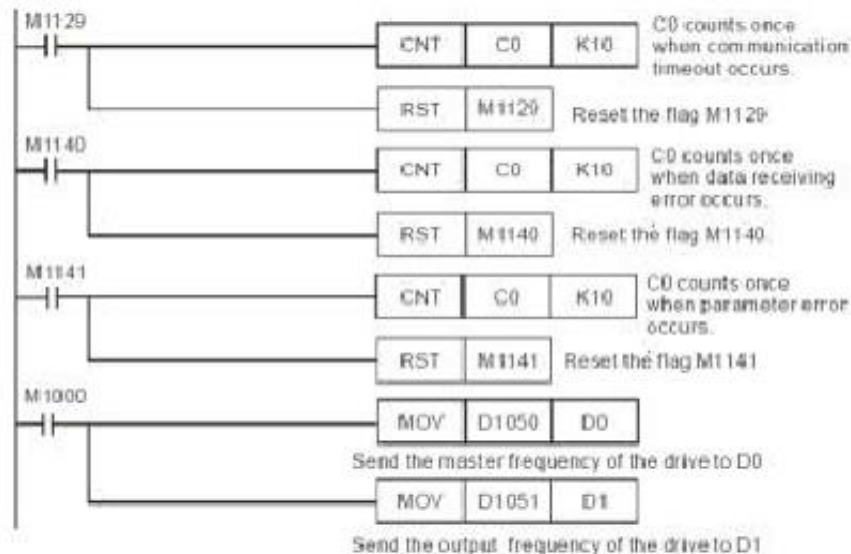
اگر بعد از انجام تنظیمات، اینورتر به درستی راه‌اندازی نشد، ابتدا باید با استفاده از تنظیم پارامتر P79 برابر با 10، تنظیمات پارامترهای اینورتر را به تنظیمات پیش‌فرض کارخانه بازگردانده و سپس تنظیمات ارتباطات شبکه را مجدداً مطابق با جدول بالا انجام دهید.

ابزارهای مورد استفاده در این مثال مطابق با جدول زیر می‌باشد (منظور از ابزار همان آدرس عملوندها می‌باشد) :

ابزارها	عملکرد ابزارها
M0	از این ابزار (فلگ عمومی) برای اجرای دستورالعمل MODRD جهت خواندن فرکانس Master و فرکانس خروجی اینورتر توسط شبکه استفاده می‌شود.
M1	از این فلگ عمومی (فلگ عمومی) برای اجرای نخستین دستورالعمل MODWR برای تنظیم جهت چرخش الکتروموتور توسط شبکه استفاده می‌شود.
M2	از این فلگ عمومی (فلگ عمومی) برای اجرای دومین دستورالعمل MODWR جهت تنظیم فرکانس راه‌اندازی الکتروموتور توسط شبکه استفاده می‌شود.
D10	از این ابزار (رجیستر عمومی) برای تنظیم جهت چرخش مورد نظر (نقطه مطلوب SP) برای تعیین کردن چرخش و یا راستگرد کار کردن الکتروموتور در هنگام راه‌اندازی استفاده می‌شود.
D11	از این ابزار (رجیستر عمومی) برای تنظیم سرعت مورد نظر (نقطه مطلوب SP) تنظیم فرکانس راه‌اندازی الکتروموتور استفاده می‌شود.

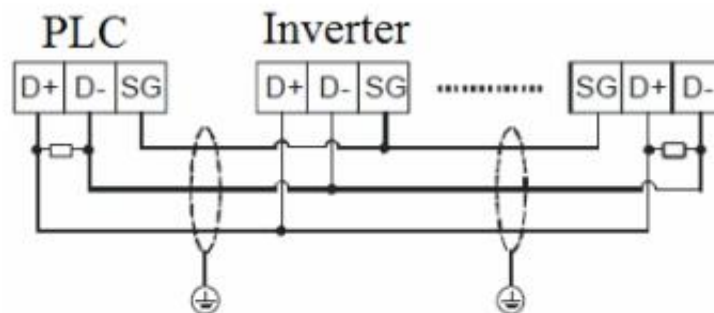
برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر است:





عملکرد این مثال به شرح زیر است:

ابتدا باید اینورتر را از طریق پورت ارتباطات RS-485 به PLC متصل کرده و سپس باید فرمت ارتباطات را برابر با 1, E, 7, 9600, MODBUSASCII تنظیم نماییم. لازم به ذکر است که فرمت ارتباطات اینورتر را نیز مانند فرمت ارتباطات PLC تنظیم نمایید. در غیر این صورت، شبکه خطا داده و کار نمی‌کند.



فقط چهار وضعیت برای ارتباطات شبکه‌ی MODBUS وجود دارد: فلگ M1127 برای ارتباطات نرمال و فلگ‌های M1129, M1140, M1141 برای خطاهای ارتباطات شبکه می‌باشد. وقتی که هر یک از این 4 فلگ یک بار فعال (ON) شوند، یک شماره به مقدار کانتر C0 اضافه می‌شود. از این رو، برنامه‌ای با قابلیت اطمینان بالا برای ارتباطات شبکه نوشته‌ایم که این اطمینان توسط مانیتور کردن وضعیت ON/OFF بودن هر 4 فلگ ارتباطات و اجرای هر سه دستورالعمل MODBUS توسط مقدار موجود در کانتر C0 حاصل شده است.

وقتی که فلگ عمومی M0=ON شود، دستورالعمل [MODRD K1 H2102 K2] اجرا خواهد شد. PLC فرکانس Master و فرکانس خروجی اینورتر را می‌خواند و سپس آن را در ابزارهای

D1073~D1076 به فرمت ASCII ذخیره می‌کند. در مرحله‌ی بعد محتویات داخل ابزارهای D1073~D1076 به صورت اتوماتیک به مقدار هگز تبدیل شده و در ابزارهای D1050 و D1051 ذخیره می‌شوند.

وقتی که فلگ M1=ON شود، دستورالعمل [MODWR K1 H2000 D10] اجرا خواهد شد. D10=H12 شده و اینورتر در جهت راستگرد راه‌اندازی می‌شود. دقت داشته باشید که جهت راه‌اندازی اینورتر را می‌توان توسط محتویات داخل ابزار D10 تغییر داد.

وقتی که فلگ M2=ON شود، دستورالعمل [MODWR K1 H2001 D11] اجرا خواهد شد. D11=K4000 شده و اینورتر با فرکانس 40Hz راه‌اندازی می‌شود. دقت داشته باشید که فرکانس خروجی اینورتر را می‌توان توسط محتویات داخل ابزار D11 تغییر داد.

در انتهای این برنامه، دستورالعمل [MOV D1050 D0] فرکانس Master اینورتر را در ابزار D0 ذخیره کرده و دستورالعمل [MOV D1051 D1] فرکانس خروجی اینورتر را در ابزار D1 ذخیره می‌کند.

با یکبار راه‌اندازی PLC، عمل خواندن/نوشتن مطابق با دستورالعمل‌های [LD=] برای درایور موتور AC (اینورتر) مکرراً اجرا خواهد شد.

7- 14 برقراری ارتباط بین PLC و اینورتر سری VFD-B دلتا با دستورالعمل MODRD/MODWR

در این مثال می‌خواهیم ارتباط بین PLC دلتا و درایور موتور AC (اینورتر) سری VFD-B دلتا را با استفاده از دستورالعمل (MODRD/MODWR) بررسی نماییم.

هدف از کنترل در این مثال به شرح زیر است:

در این مثال می‌خواهیم مکرراً مقدار فرکانس Master و فرکانس خروجی اینورتر سری VFD-B دلتا را توسط دستورالعمل MODRD بخوانیم. وقتی که شستی Start متصل شده به PLC دلتا فشرده شد، اینورتر شروع به کار کرده و الکتروموتور در جهت معکوس شروع به چرخیدن کند. در هنگام روشن شدن، فرکانس خروجی اینورتر در هر یک ثانیه 1Hz بالا رفته تا به 50Hz برسد. در این زمان فرکانس خروجی برابر با 50Hz نگهداشته می‌شود.

وقتی که شستی Stop فشرده شود، الکتروموتور متصل شده به اینورتر متوقف خواهد شد. تمام این عملیات راه‌اندازی و توقف موتور توسط دستورالعمل MODWR انجام می‌شود.

تنظیمات پارامترهای اینورتر مدل VFD-B دلتا برای راه‌اندازی توسط شبکه RS-485 مطابق با جدول زیر است:

پارامتر	مقدار تنظیمی	توضیحات
02-00	04	از این پارامتر برای تنظیم ارتباطات سریال RS-485 و ذخیره‌سازی آخرین فرکانس استفاده شده برای راه‌اندازی الکتروموتور توسط اینورتر استفاده می‌شود.
02-01	03	از این پارامتر برای تنظیم ارتباطات سریال RS-485 و فعال کردن دکمه‌های STOP/RESET بر روی کی‌پد اینورتر استفاده می‌شود.
09-00	01	از این پارامتر برای تنظیم آدرس گرهی ارتباطات اینورتر برای اتصال به شبکه‌ی ارتباطات RS-485 استفاده می‌شود. مقدار پیش‌فرض آدرس گرهی ارتباطات اینورتر برابر با 01 است.
09-01	02	از این پارامتر برای تنظیم سرعت تبادل داده در شبکه‌ی ارتباطات RS-485 استفاده می‌شود. مقدار پیش‌فرض سرعت تبادل داده در شبکه‌ی ارتباطات RS-485 برابر با 19200 می‌باشد.
09-04	03	از این پارامتر برای تنظیم مدارات ارتباطات شبکه‌ی RS-485 با پروتکل MODBUS استفاده می‌شود. مقدار پیش‌فرض این پارامتر برابر با مد MODBUS RTU mode و با پروتکل 8,N,2 می‌باشد.

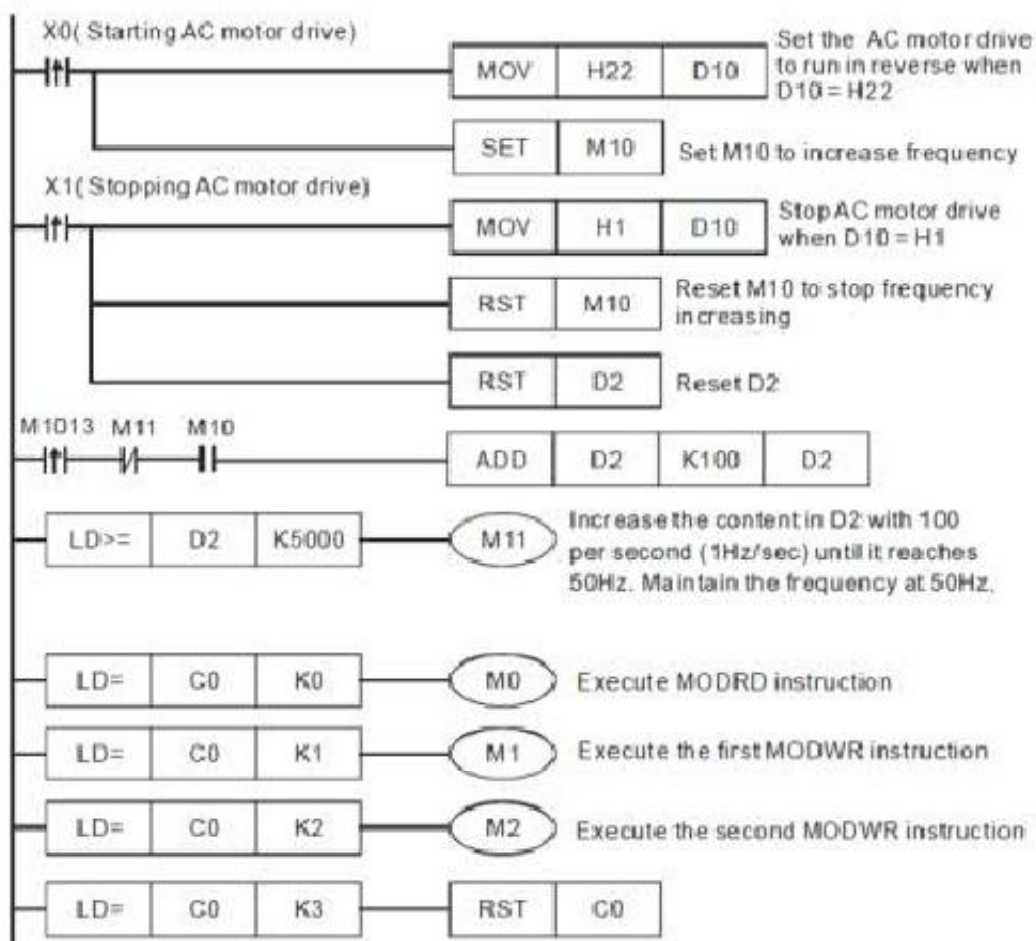
اگر بعد از تنظیمات انجام شده برای اینورتر الکتروموتور به صورت نرمال راه‌اندازی نشد، باید ابتدا تنظیمات پارامترهای اینورتر را ریست کرده و توسط پارامتر $P00-02=10$ به مقدار پیش‌فرض کارخانه بازگردانید. سپس باید تنظیمات بالا را مجدداً برای اینورتر انجام دهید.

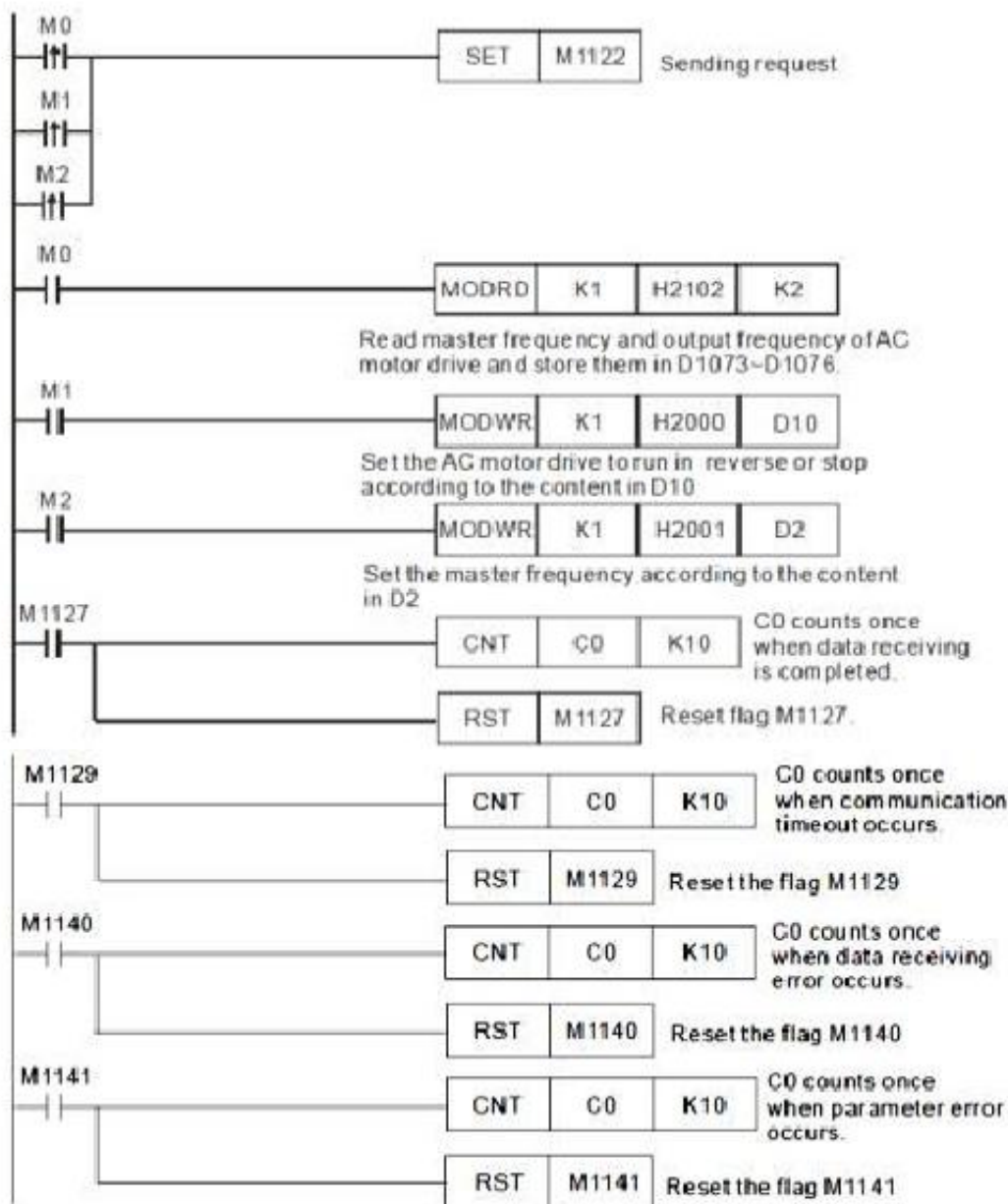
ابزارهای مورد استفاده در این مثال مطابق با جدول زیر می‌باشد:

ابزارها	توضیحات
X0	به این ورودی PLC شستی START برای راه‌اندازی الکتروموتور (اینورتر) متصل می‌شود.
X1	به این ورودی PLC شستی STOP برای توقف الکتروموتور (اینورتر) متصل می‌شود.
M0	از این ابزار برای اجرای دستورالعمل MODRD جهت خواندن فرکانس مستر و فرکانس خروجی اینورتر استفاده می‌شود.
M1	از این ابزار برای اجرای دستورالعمل MODWR جهت راه‌اندازی اینورتر و تعیین جهت حرکت الکتروموتور استفاده می‌شود.
M2	از این ابزار برای اجرای دستورالعمل MODWR جهت راه‌اندازی اینورتر با فرکانس مدنظر (تنظیمی در PLC) استفاده می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر است:







عملکرد این مثال به شرح زیر می‌باشد:

ابتدا باید پورت ارتباطات RS-485 را برای PLC برابر با فرمت MODBUS RTU, 19200, 8, N, 2 تنظیم کنیم که توسط 4 خط اول برنامه امکان‌پذیر است. دقت کنید که فرمت ارتباطات شبکه‌ی RS-485 در سمت اینورتر نیز باید با فرمت ارتباطات PLC یکسان باشد. 4 حالت برای ارتباطات MODBUS وجود دارد که این چهار حالت توسط فلگ‌هایی اجرا می‌شود. این 4 فلگ شامل فلگ M1127 برای ارتباطات نرمال و فلگ‌های M114, M1140 و M1129 برای خطاهای ارتباطات شبکه می‌باشد. وقتی که هر یک از فلگ‌های گفته شده فعال ON شوند، یک

عدد به شمارنده C0 اضافه می‌شود. از این رو، با استفاده از فلگ‌ها، برنامه‌ای مطمئن و با قابلیت اعتماد بالا را برای شبکه نوشته‌ایم. این اطمینان توسط مانیتور کردن ON/OFF بودن وضعیت هر یک از چهار فلگ و اجرای هر سه دستورالعمل MODBUS توسط مقدار موجود در شمارنده C0 حاصل می‌شود.

وقتی که M0=ON شود، دستورالعمل [MODRD K1 H2102 K2] اجرا خواهد شد. در این وضعیت PLC فرکانس مستر و فرکانس خروجی اینورتر را خوانده و در مرحله بعد این مقادیر خوانده شده را در ابزارهای D1073~D1076 به فرمت ASCII ذخیره می‌کند. بعد از ذخیره‌سازی مقادیر به فرمت اسکی، PLC به صورت اتوماتیک مقادیر موجود در ابزارهای D1073~D1076 را به فرمت هگز تبدیل کرده و مقدار جدید را به ابزارهای D1050 و D1051 منتقل می‌کند.

وقتی که M1=ON شود، دستورالعمل [MODWR K1 H2000 D10] اجرا می‌شود. D10=H22 شده و اینورتر در جهت معکوس راه‌اندازی می‌شود. اگر مقدار D10=H1 شود، اینورتر استوپ می‌شود.

وقتی که D2=ON شود، دستورالعمل [MODWR K1 H2001 D2] اجرا می‌شود. با اجرای این دستورالعمل می‌توان با تغییر محتویات داخل ابزار D2 فرکانس کاری موتور را تغییر داد. با یک بار راه‌اندازی PLC، عمل خواندن و نوشتن مقادیر تحت شبکه برای اینورتر مطابق با دستورالعمل [LD=] مکرراً اجرا خواهد شد.

7-15 برقراری ارتباط بین PLC و اینورتر سری VFD-V دلتا با دستورالعمل MODRD/MODWR

در این مثال، می‌خواهیم ارتباط بین PLC دلتا و اینورتر سری VFD-V را توسط دستورالعمل‌های MODRD/MODWR بررسی نماییم.

هدف از کنترل در این مثال به شرح زیر است:

در این مثال، می‌خواهیم مکرراً مقدار فرکانس مستر و فرکانس خروجی اینورتر مدل VFD-V شرکت دلتا را با استفاده از دستورالعمل MODRD بخوانیم. وقتی که شستی متصل شده به ورودی X0 فشرده شود، دستورالعمل MODWR اجرا شده و اینورتر با فرکانس خروجی 30Hz در جهت راستگرد راه‌اندازی می‌شود. وقتی که شستی متصل شده به ورودی X1 فشرده شود، دستورالعمل MODWR اجرا شده و اینورتر با فرکانس خروجی 20Hz در جهت معکوس راه‌اندازی می‌شود. وقتی که شستی متصل شده به ورودی X2 فشرده شود، دستورالعمل MODWR اجرا شده و اینورتر استوپ می‌شود.

تنظیمات پارامترهای اینورتر مدل VFD-V دلتا برای شبکه RS-485 با پروتکل MODBUS مطابق با جدول زیر است:

پارامتر	مقدار تنظیمی	توضیحات
P00-20	1	این پارامتر کنترل فرکانس مستر توسط شبکه‌ی ارتباطات RS-485 را امکان‌پذیر می‌کند.
P00-21	0	این پارامتر کی پد دیجیتال (KPV-CE01) روی اینورتر را فعال می‌کند.
P09-00	01	توسط این پارامتر می‌توان آدرس گره‌ی اینورتر متصل شده به شبکه RS-485 برابر با 01 و یا یک مقدار بین 1 الی 254 تنظیم کرد.
P09-01	9.6	سرعت تبادل داده در شبکه‌ی ارتباطات RS-485 برابر با 9600 تنظیم شده است.
P09-04	02	شبکه‌ی ارتباطات RS-485 برابر با مد ASCII mode و با پروتکل 7,E,1 تنظیم شده است.

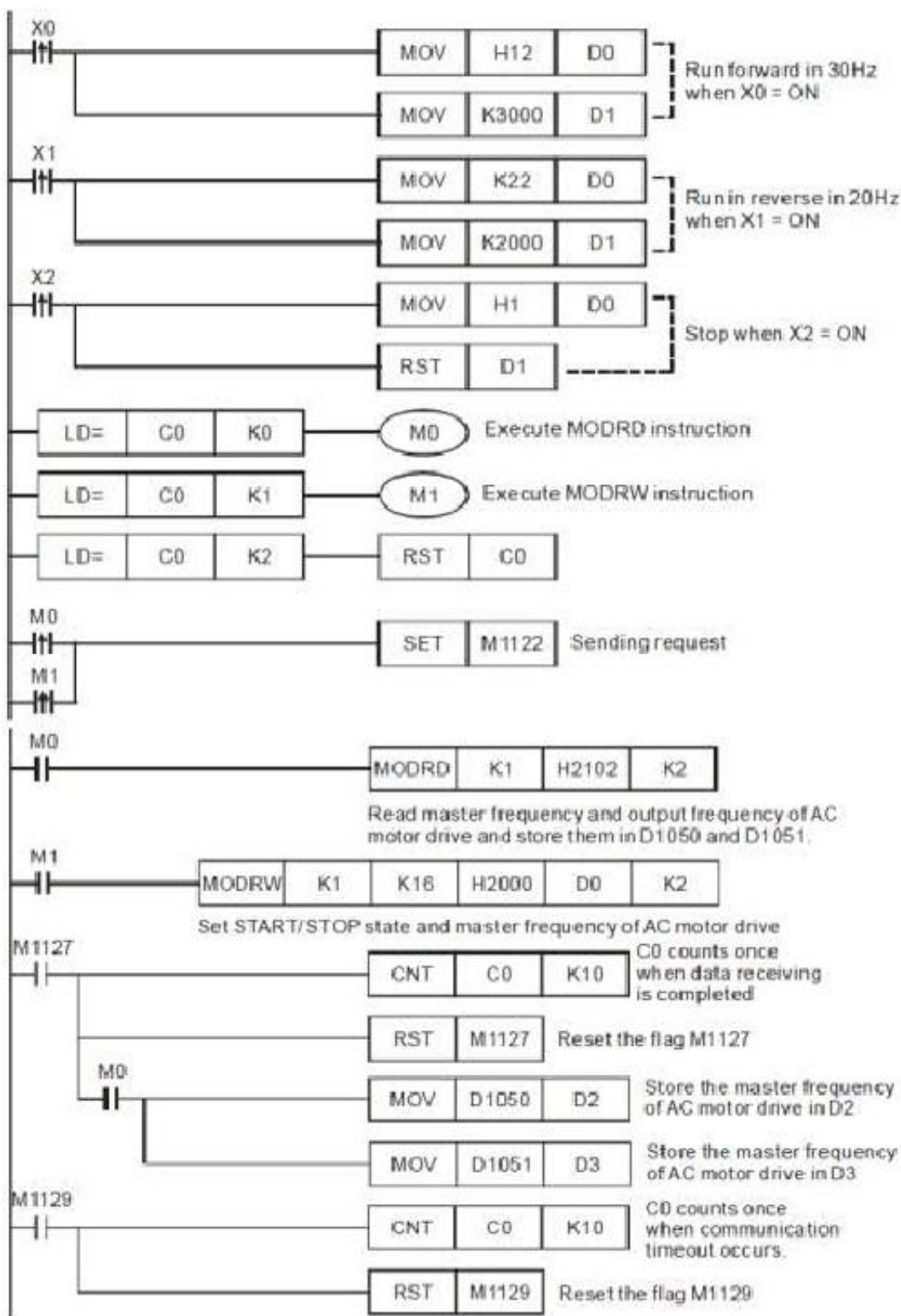
اگر بعد از اجرای دستورالعمل اینورتر به درستی در شبکه راه‌اندازی نشد، ابتدا باید مقدار پارامترهای اینورتر را توسط $P00-02=10$ به مقدار پیش‌فرض کارخانه بازگردانده و سپس پارامترهای گفته شده در جدول بالا را مجدداً تنظیم نمایید.

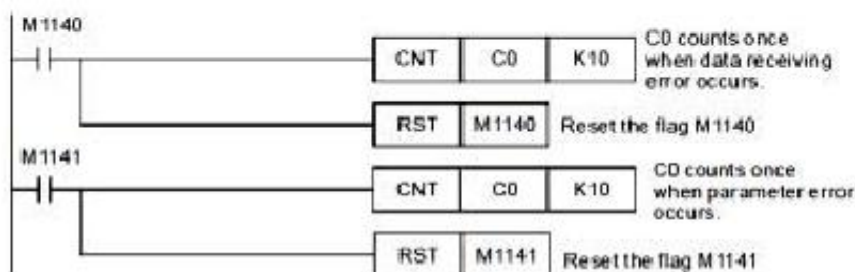
ابزارهای مورد استفاده در برنامه‌نویسی PLC دلتا برای راه‌اندازی اینورتر با استفاده از شبکه RS-485 در این مثال مطابق جدول زیر است:

ابزارها	توضیحات
X0	به این ورودی PLC شستی Forward متصل می‌شود.
X1	به این ورودی PLC شستی Reverse متصل می‌شود.
X2	به این ورودی PLC شستی Stop متصل می‌شود.
M0	از این ابزار برای اجرای دستورالعمل MODRD جهت خواندن فرکانس مستر و فرکانس خروجی اینورتر توسط شبکه استفاده می‌شود.
M1	از این ابزار برای اجرای دستورالعمل MODWR جهت راه‌اندازی و تنظیم فرکانس خروجی اینورتر توسط شبکه استفاده می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نربانی شکل زیر می‌باشد:







عملکرد این مثال به شرح زیر است :

برای برقراری ارتباط بین PLC دلتا و اینورتر سری VFD-V دلتا ابتدا باید تنظیمات پورت RS-485 را برای PLC برنامه‌نویسی کرده و فرمت ارتباطات شبکه را برابر با MODBUS RTU, 1, 7, E, 9600 تنظیم نمایید که این کار توسط چهار خط اول برنامه امکان‌پذیر است. دقت کنید که همین تنظیمات ارتباطات شبکه در سمت اینورتر نیز باید به صورت یکسان اجرا شود. وقتی که تغذیه PLC وصل شود، محتویات راه‌اندازی ایزارهای D0 و D1 ریست شده که این قسمت برای اطمینان از وضعیت استوپ بودن اینورتر در لحظه‌ی وصل تغذیه PLC برنامه‌نویسی شده است.

وقتی که ورودی X0 فعال شود، PLC مقدار D0=H12 را برای راه‌اندازی اینورتر در جهت راستگرد و مقدار D1=3000 را برای فرکانس خروجی 30Hz به اینورتر ارسال می‌کند که این موضوع توسط ارتباطات شبکه صورت می‌گیرد.

وقتی که ورودی X1 فعال شود، PLC مقدار D0=H22 را برای راه‌اندازی اینورتر در جهت چپگرد و مقدار D1=2000 را برای فرکانس خروجی 20Hz از طریق شبکه‌ی ارتباطات به اینورتر ارسال می‌کند.

وقتی که ورودی X2 فعال شود، PLC مقادیر D0=H1 و D1=0 را برای استوپ کردن اینورتر از طریق واسط ارتباطات RS-485 به اینورتر ارسال می‌کند.

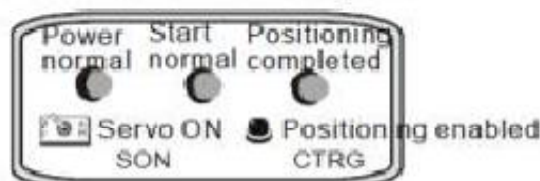
در برنامه چهار وضعیت برای ارتباطات MODBUS در نظر گرفته شده که توسط فلگ‌های خاص می‌توان چهار حالت را برنامه‌نویسی کرد. فلگ M1127 مربوط به نرمال بودن ارتباطات شبکه بوده و فلگ‌های M1141 و M1140 و M1129 مربوط به خطاهای شبکه‌ی ارتباطات می‌باشد. از یک کانتر برای اجرای دو دستورالعمل MODBUS استفاده شده است. وقتی که هر یک از فلگ‌های گفته شده در بالا فعال شوند، یک مقدار به مقدار جاری کانتر C0 اضافه می‌شود. از این رو، با استفاده از کانتر برای شبکه‌ی ارتباطات یک برنامه با قابلیت اعتماد بالا را نوشته‌ایم. این اعتماد توسط مانیتور کردن وضعیت‌های (ON/OFF) هر یک از چهار فلگ گفته شده و اجرای دو دستورالعمل MODBUS توسط مقدار جاری کانتر C0 حاصل شده است.

در این برنامه فرکانس مستر و فرکانس خروجی در ابزارهای D1050 و D1051 ذخیره شده و سپس مقادیر این دو ابزار که مختص ارتباطات شبکه می‌باشند، جهت ذخیره‌سازی نهایی به ابزارهای D2 و D3 منتقل خواهند شد. با یک‌بار راه‌اندازی PLC عمل خواندن و نوشتن مقادیر اینورتر توسط PLC مکرراً توسط دستورالعمل [LD=] اجرا می‌شود.

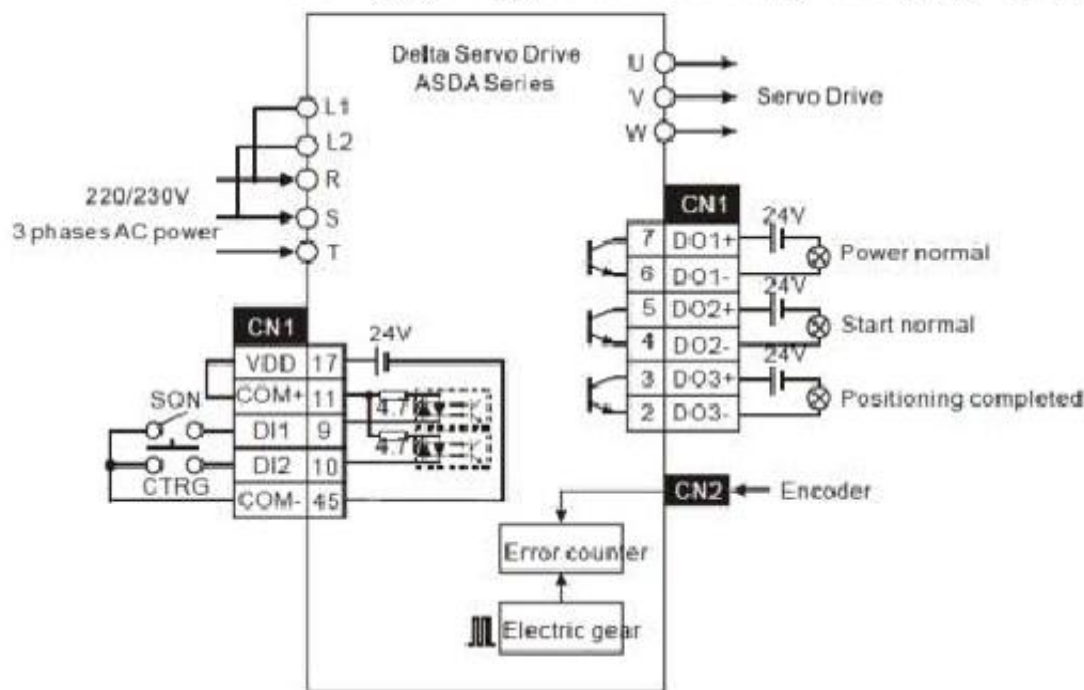
7- 16 برقراری ارتباط بین PLC و سرودرایو ASD-A دلتا با دستورالعمل MODRD/MODRW در مد موقعیت

در این مثال، می‌خواهیم ارتباط بین PLC دلتا و سرودرایو AC سری ASD-A دلتا را برای کنترل موقعیت توسط دستورالعمل MODRD/MODRW در شبکه ارتباطات RS-485 بررسی نماییم. پنل کنترل سرودرایو AC مطابق با شکل زیر می‌باشد:

AC Servo drive control panel



سیم‌بندی سرودرایو AC سری ASD-A دلتا مطابق با شکل زیر است:



هدف از کنترل در این مثال به شرح زیر می‌باشد:

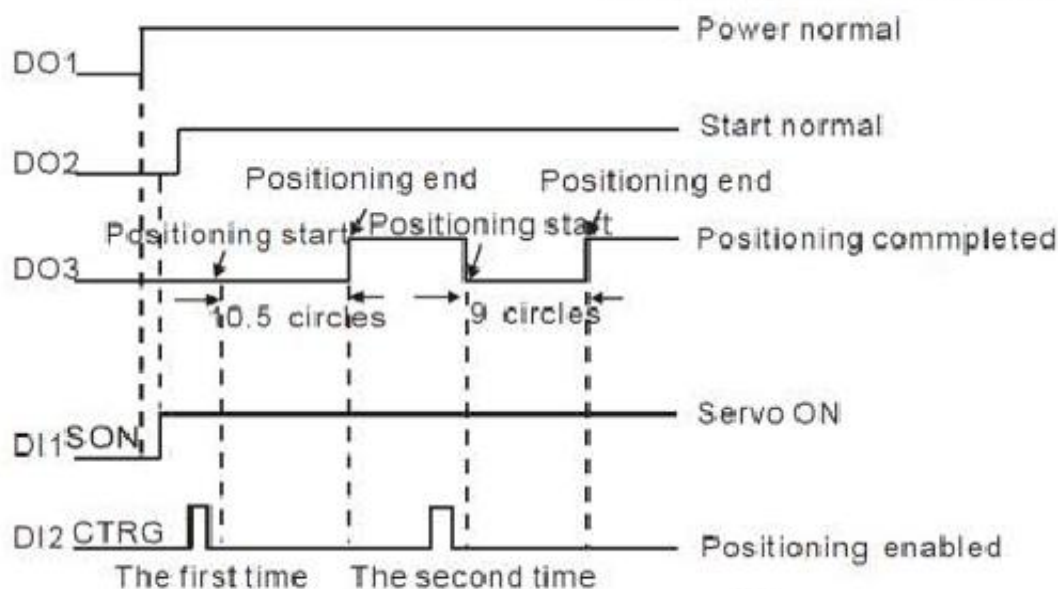
در این مثال، می‌خواهیم موقعیت هدف سروموتور را از سرودرایو AC توسط دستورالعمل MODRD بخوانیم. کنترل موقعیت سروسیستم به صورت افزایشی است. همچنین، می‌خواهیم توسط دستورالعمل MODRW تنظیمات موقعیت هدف سرودرایو AC را از PLC به سرودرایو بفرستیم. به ورودی‌های دیجیتال PLC (DI1~DI2) سه شستی برای فعال کردن Enabling، راه‌اندازی Starting و عمل موقعیت‌یابی Positioning action متصل می‌شود که با فشردن هر کدام عمل مربوط به آن شستی برای عملکرد سرودرایو اجرا شود. توسط نمایشگر متصل شده به خروجی‌های PLC (DO1~DO3) وضعیت‌های مربوط به سرودرایو AC نشان داده می‌شود.

پارامترهای تنظیمی سرودرایو برای ارتباط شبکه‌ی سرودرایو AC سری ASD-A دلتا با PLC دلتا مطابق با جدول زیر می‌باشد.

پارامتر	مقدار تنظیمی	توضیحات
P1-01	1	مد کنترل و جهت خروجی
P1-33	1	مد کنترل موقعیت (Pr)
P2-10	101	ترمینال ورودی دیجیتال یک (DI1)
P2-11	108	ترمینال ورودی دیجیتال دو (DI2)
P2-15	0	ترمینال ورودی دیجیتال شش (DI6)
P2-16	0	ترمینال ورودی دیجیتال هفت (DI7)
P2-17	0	ترمینال ورودی دیجیتال هشت (DI8)
P2-18	101	ترمینال خروجی دیجیتال شماره‌ی یک (DO1)
P2-19	102	ترمینال خروجی دیجیتال شماره‌ی دو (DO2)
P2-20	105	ترمینال خروجی دیجیتال شماره‌ی سه (DO3)
P3-00	1	تنظیم آدرس ارتباطات شبکه‌ی RS-485
P3-01	1	نرخ سرعت تبادل داده در شبکه‌ی RS-485 برابر با 9600 تنظیم شده است.
P3-02	1	پروتکل ارتباطات (7,E,1) MODBUS ASCII mode. Data format:
P3-03	1	اگر خطایی در ارتباطات شبکه رخ داد، سروسیستم خطا داده و استوپ می‌شود.
P3-05	2	فرمت ارتباطات شبکه‌ی RS-485
P3-06	0	تابع ارتباطات ورودی دیجیتال

اگر بعد از تنظیمات پارامترهای سرودرایو، سیستم به درستی عمل نکرد، شما باید ابتدا تنظیمات پارامترهای سرودرایو را به تنظیمات پیش‌فرض کارخانه بازگردانید. این عمل با تنظیم پارامتر P2-08=10 امکان‌پذیر است. حال مجدداً پارامترهای سرودرایو را مطابق با جدول بالا تنظیم نمایید.

مراحل عملکرد سیستم به شرح زیر می‌باشد:



بعد از وصل تغذیه سرودرایو AC پارامترهای سرو را تنظیم نمایید. اگر بعد از وصل تغذیه خطایی رخ ندهد و تغذیه نرمال باشد، نمایشگر متصل شده به خروجی دیجیتال شماره‌ی یک (DO1) سرودرایو فعال ON خواهد شد.

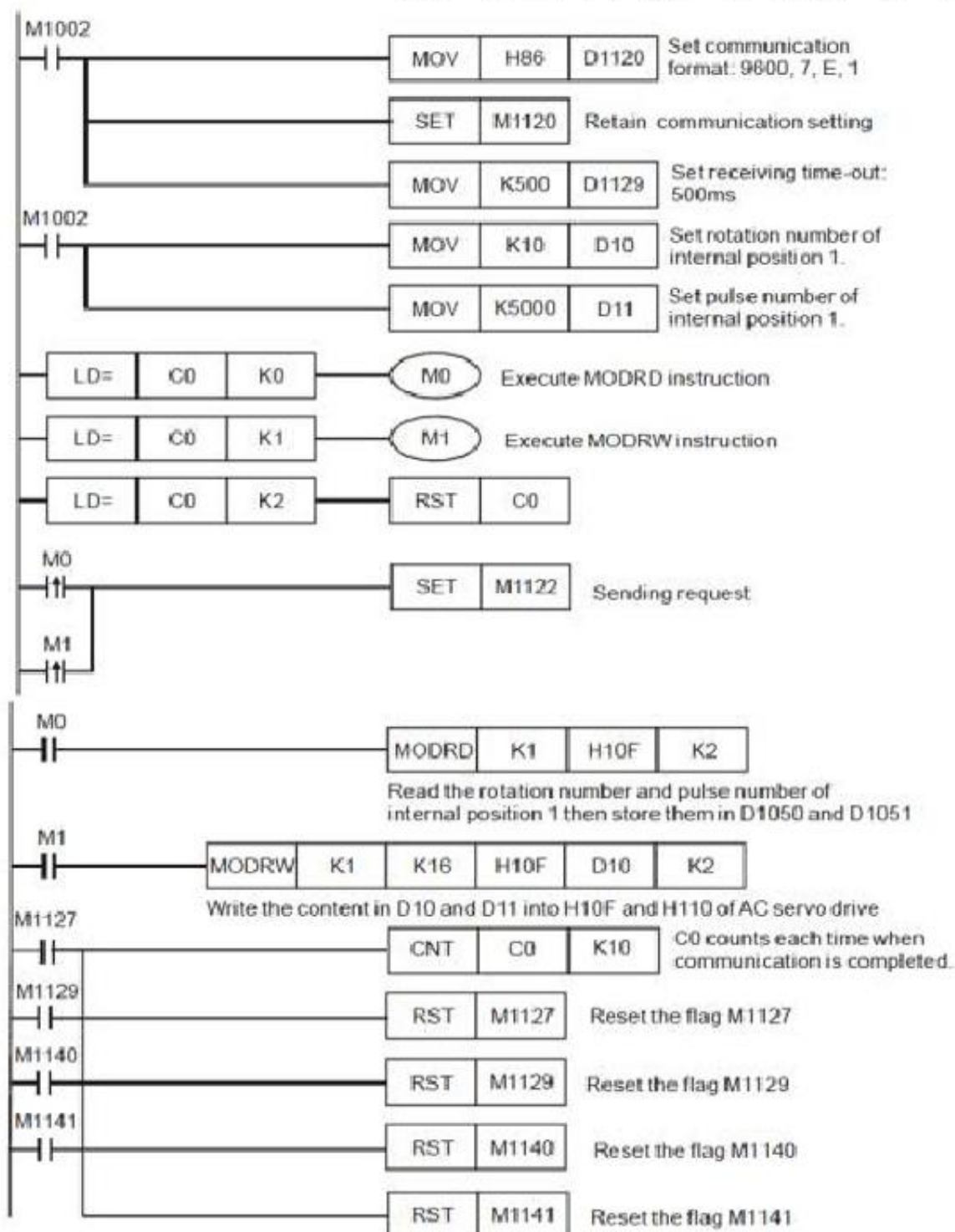
وقتی که نمایشگر نرمال بودن تغذیه فعال (ON) باشد، برای فعال کردن سرو (Servo) SON (ON) باید ورودی دیجیتال یک (DO1) فعال شود. اگر بعد از فعال شدن سرو (SON) خطایی رخ ندهد، نمایشگر Start normal که به خروجی دیجیتال شماره‌ی دو (DO2) سرودرایو متصل شده است، روشن خواهد شد.

وقتی که نمایشگر Start normal فعال ON شود، با راه انداختن ورودی دیجیتال DI1 عمل موقعیت‌یابی سروسیستم (Position enabled) CTRG فعال خواهد شد. سروموتور برای 10.5 سیکل خواهد چرخید و سپس نمایشگر Positioning Completed متصل شده به خروجی دیجیتال DO3 سرودرایو روشن خواهد شد.

ابزارهای مورد استفاده در برنامه‌نویسی PLC دلتا برای این مثال مطابق با جدول زیر می‌باشد:

ابزار	توضیحات
M0	از این ابزار برای اجرای دستورالعمل MODRD جهت خواندن تعداد چرخش موتور و تعداد پالس نخستین موقعیت‌یابی داخلی استفاده می‌شود.
M1	از این ابزار برای اجرای دستورالعمل MODRW جهت تنظیم تعداد چرخش موتور و تعداد پالس نخستین موقعیت‌یابی داخلی استفاده می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر است :



عملکرد این برنامه‌ی کنترلی به شرح زیر است :

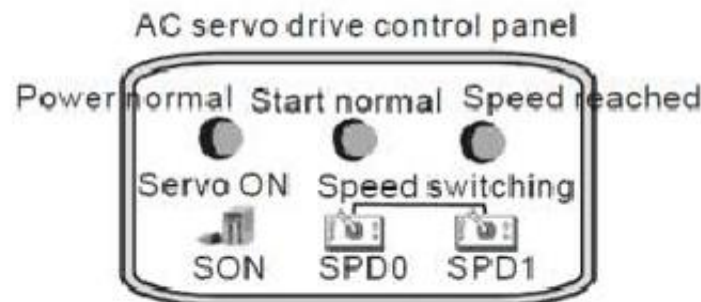
برای برقراری ارتباط بین PLC دلتا و سرودرایو سری ASD-A دلتا ابتدا باید پورت ارتباطات شبکه RS-485 را در سمت PLC مقداردهی نموده و سپس فرمت ارتباطات شبکه را برابر با MODBUS ASCII, 9600, 7, E, 1 تنظیم نمایید. لازم به ذکر است که بگوییم، همین فرمت ارتباطات شبکه را در سمت سرودرایو نیز به صورت یکسان باید تنظیم نمایید. وقتی که M0=ON شود، دستورالعمل [MODRD K1 H10F K2] اجرا خواهد شد و تعداد دور چرخیده شده‌ی سروموتور و تعداد پالس اولین موقعیت‌دهی داخلی توسط PLC خوانده شده و در ابزارهای مختص شبکه‌ی ارتباطات که رجیسترهای D1050 و D1051 می‌باشند ذخیره خواهد شد. وقتی که M1=ON شود، دستورالعمل [MODWR K1 K16 H10F D10 K2] اجرا خواهد شد و محتویات ابزارهای D10 و D11 در آدرس‌های H120F (تعداد چرخش نخستین موقعیت داخلی) و آدرس H10 (تعداد پالس نخستین موقعیت داخلی) سرودرایو نوشته خواهد شد. هر دو سیگنال Start و trigger توسط سوئیچ‌هایی که به صورت خارجی به سرودرایو سیم‌بندی شده‌اند، کنترل می‌شوند. برای اطلاع از سیم‌بندی سرودرایو به دیاگرام سیم‌بندی همین مثال مراجعه نمایید.

از چهار فلگ برای بالا بردن ضریب اطمینان ارتباطات شبکه‌ی MODBUS استفاده شده است که این چهار فلگ عبارتند از : فلگ M1127 بیانگر نرمال بودن ارتباطات شبکه بوده و فلگ‌های M1141 و M1140 و M1129 بیانگر وجود خطا در ارتباطات شبکه می‌باشد. از اینرو، با مانیتور کردن وضعیت‌های ON/OFF این چهار فلگ و اجرای 2 دستورالعمل MODBUS توسط مقدار جاری شمارنده C0 یک ارتباط مطمئن و بی‌خطر را ایجاد کرده‌ایم. با یکبار Run کردن (راه‌اندازی) PLC، عمل خواندن و نوشتن مقادیر از سرودرایو AC توسط اجرا مکرر دستورالعمل [LD=] صورت خواهد گرفت.

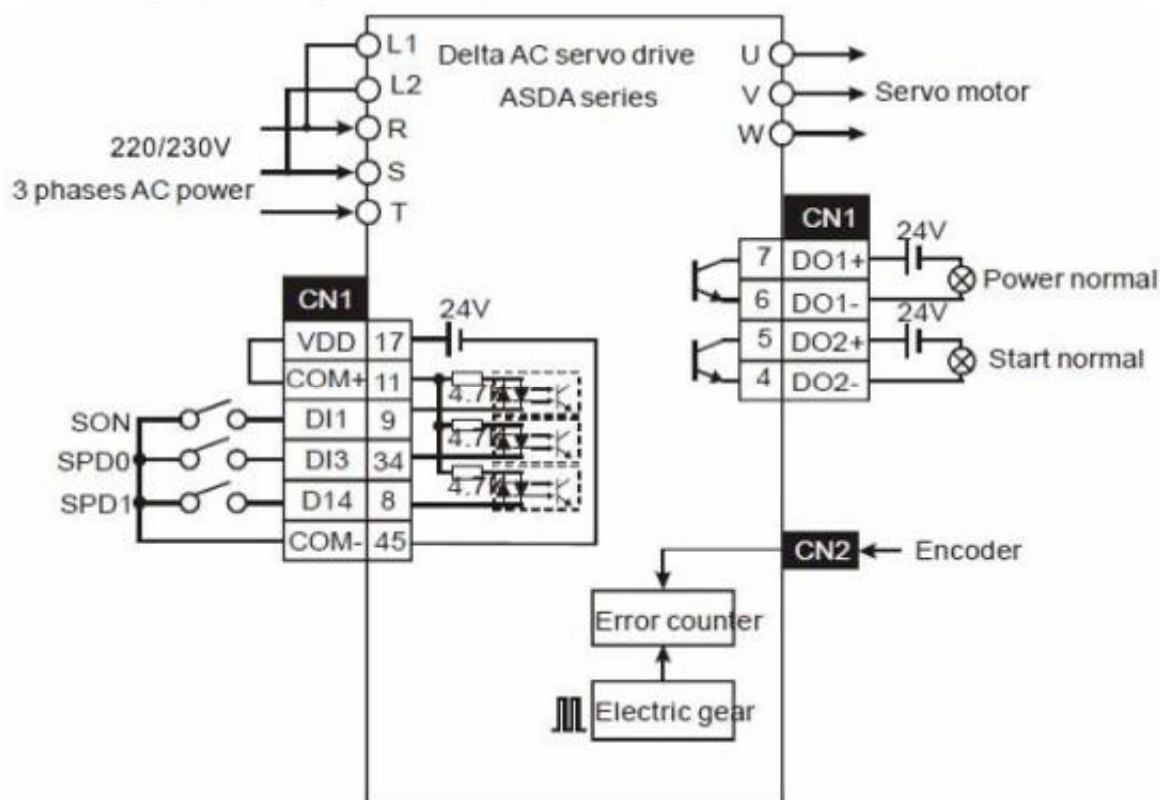
7- 17 برقراری ارتباط بین PLC و سرودرایو ASD-A دلتا با دستورالعمل

MODRD/MODRW در مد سرعت

در این مثال، می‌خواهیم یک ارتباط شبکه RS-485 بین PLC دلتا و سرودرایو سری ASD-A دلتا را برقرار نموده و سپس توسط اجرای دستورات MODRD/ MODRW سرعت سروموتور را در مد سرعت کنترل کنیم.



سیم بندی سرودرایو AC سری ASD-A دلتا در مد کنترل سرعت مطابق شکل زیر می‌باشد:



هدف از کنترل در این مثال به شرح زیر است:

در این مثال، می‌خواهیم سرعت چرخش موتور را از طریق شبکه بخوانیم و آن را در ابزار D0 با استفاده از دستورالعمل MODRD ذخیره کنیم.

همچنین، می‌خواهیم سرعت چرخش موتور را با استفاده از دو سرعت ثابت از پیش تنظیم شده کنترل نماییم و یا توسط دستورالعمل MODWR سرعت مورد نظر را با سوئیچ کردن بین SPD0 و SPD1 تعیین کنیم.

در جدول زیر، سوئیچ‌های انتخاب سرعت ثابت سرودرایو AC آورده شده است که هر وضعیت این دو سوئیچ یک سرعت ثابت از پیش تنظیم شده توسط کاربر در پارامترهای سرودرایو را تعریف می‌کند.

وضعیت SPD0	وضعیت SPD1	توضیحات
ON	OFF	اگر SPD0=ON شود، نخستین سرعت ثابت تنظیم شده در پارامتر P1-09 فعال شده و سروموتور با این سرعت شروع به چرخش می‌کند. در این مثال، محتویات داخل رجیستر D9 که در PLC تنظیم می‌شود، به پارامتر P1-09 سرودرایو ارسال می‌شود. در این مثال، محتوای داخل رجیستر D9 عدد ثابت K1500 می‌باشد. بنابراین سرعت چرخش سروموتور برابر با 1500 r/min در جهت Forward است.
OFF	ON	اگر SPD1=ON شود، دومین سرعت ثابت تنظیم شده در پارامتر P1-10 فعال شده و سروموتور با این سرعت شروع به چرخش می‌کند. در این مثال، محتویات داخل رجیستر D10 که در PLC تنظیم می‌شود، به پارامتر P1-10 سرودرایو ارسال می‌شود. در این مثال، محتوای داخل رجیستر D10 عدد ثابت K-1500 می‌باشد. بنابراین سرعت چرخش سروموتور برابر با 1500 r/min در جهت Reverse است.
ON	ON	اگر SPD0, SPD1=ON شوند، سومین سرعت ثابت تنظیم شده در پارامتر P1-11 فعال شده و سروموتور با این سرعت شروع به چرخش می‌کند. در این مثال، محتویات داخل رجیستر D11 که در PLC تنظیم می‌شود، به پارامتر P1-11 سرودرایو ارسال می‌شود. در این مثال، محتوای داخل رجیستر D11 جهت تعیین سرعت چرخش سروموتور توسط کاربر تعیین می‌شود.

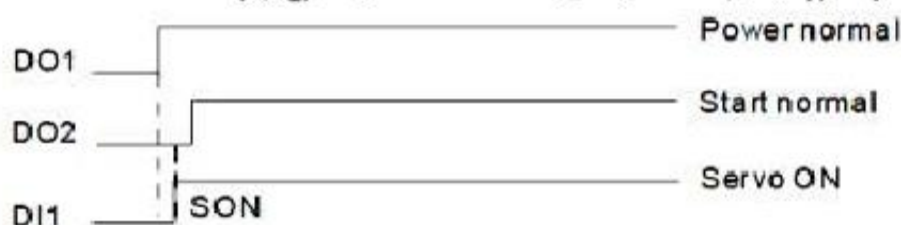
در این مثال، از خروجی‌های دیجیتال D01~D03 در سمت سرودرایو دلتا جهت نمایش هر یک از وضعیت‌های سرودرایو AC استفاده شده است.

برای راه‌اندازی سرودرایو AC سری ASDA دلتا و تنظیمات آن باید پارامترهای نشان داده شده در جدول زیر را برای سرودرایو تنظیم نمایید.

پارامتر	مقدار تنظیمی	توضیحات
P1-01	2	تنظیم مد کنترل سروسیستم و جهت چرخش سروموتور در این پارامتر صورت می‌گیرد
P1-39	1500	تنظیم سرعت هدف سروموتور برابر با 1500 rpm
P2-10	101	تنظیم عملکرد نخستین ترمینال ورودی دیجیتال (DI1) در این پارامتر صورت می‌گیرد
P2-12	114	تنظیم عملکرد سومین ترمینال ورودی دیجیتال (DI3) برابر با SPD0
P2-13	115	تنظیم عملکرد چهارمین ترمینال ورودی دیجیتال (DI4) برابر با SPD1
P2-15	0	برای این پارامتر هیچ تابعی تنظیم نشود
P2-16	0	برای این پارامتر هیچ تابعی تنظیم نشود
P2-17	0	برای این پارامتر هیچ تابعی تنظیم نشود
P2-18	101	اگر هیچ خطایی بعد از وصل تغذیه سروسیستم وجود نداشته باشد، DO1=ON می‌شود.
P2-19	102	اگر هیچ خطایی بعد از راه‌اندازی سروسیستم وجود نداشته باشد، DO2=ON می‌شود.
P2-20	104	وقتی که سرعت سروموتور به سرعت هدف تنظیم شده رسید، DO3=ON می‌شود.
P3-00	1	در این پارامتر باید آدرس گره‌ی ارتباطات شبکه‌ی سرودرایو را برابر با 1 تنظیم نمایید.
P3-01	1	در این پارامتر باید سرعت تبادل داده را برای شبکه‌ی ارتباطات سرودرایو برابر با 9600 تنظیم نمایید.
P3-02	1	در این پارامتر پروتکل ارتباطات شبکه را برابر با MODBUS ASCII mode و فرمت ارتباطات را برابر با 7, 1 تنظیم می‌کنیم.
P3-05	2	تنظیم فرمت ارتباطات RS-485 در این پارامتر صورت می‌گیرد.
P3-06	0	تنظیم ورودی دیجیتال تابع ارتباطات در این پارامتر صورت می‌گیرد.

اگر بعد از تنظیمات پارامترهای گفته شده در جدول بالا برای سرودرایو AC سری ASDA دلتا، سروسیستم به درستی عمل نکرد، باید با استفاده از تنظیم پارامتر $P2-08=10$ برای سرودرایو، تنظیمات سرو را به تنظیمات کارخانه بازگردانید سپس تنظیمات گفته شده را مجدداً برای سرو اجرا نمایید.

مراحل عملکرد سروسیستم تحت شبکه‌ی MODBUS به شرح زیر است:



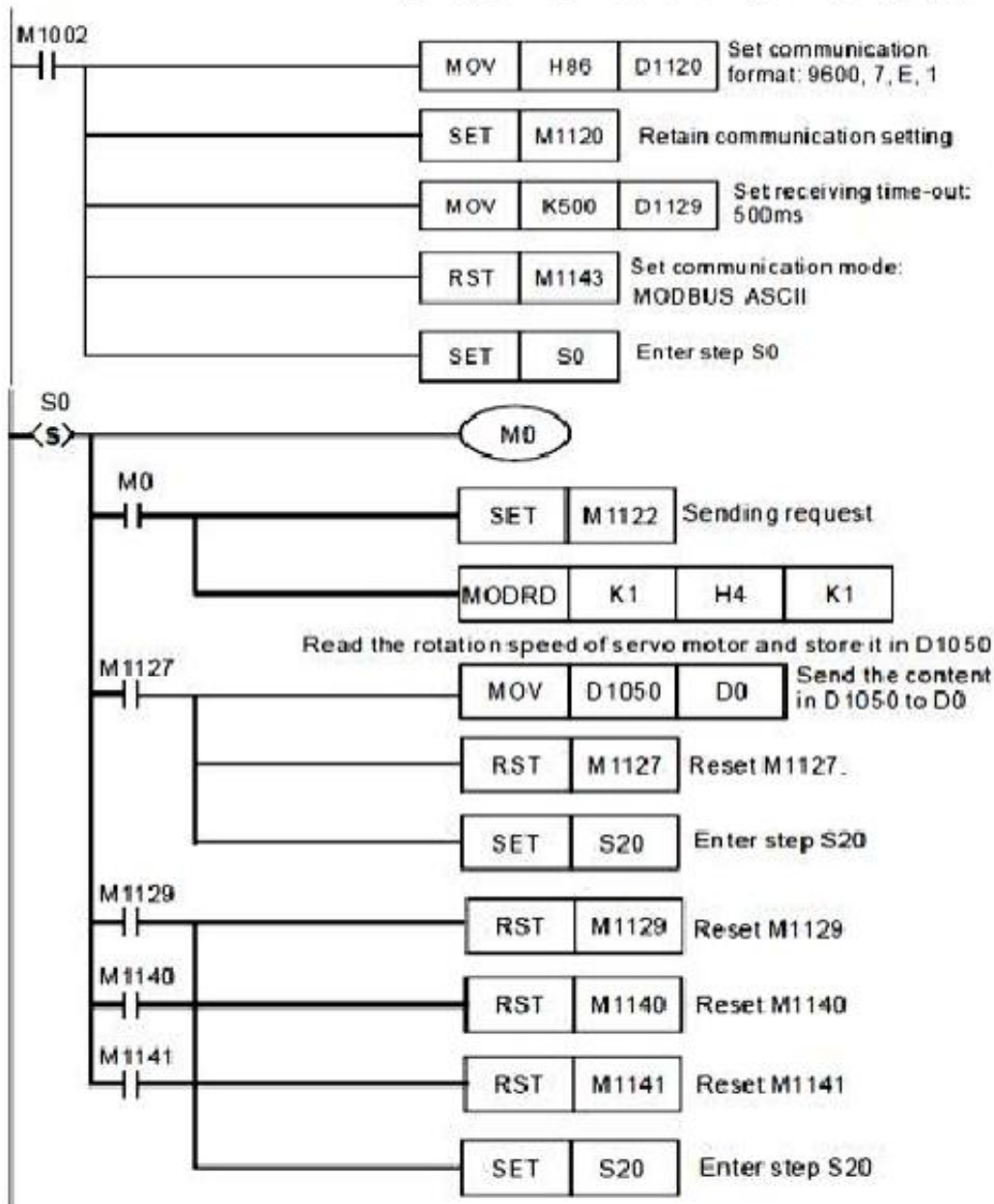
ابتدا باید پارامترهای سرودرایو را مطابق با تنظیمات گفته شده در جدول بالا تنظیم نمایید. سپس یکبار تغذیه سرو را قطع کرده و مجدداً وصل نمایید. اگر هیچ خطایی رخ ندهد، نمایشگر نرمال بودن تغذیه سروسیستم Power normal که به خروجی DO1 سرودرایو متصل شده است، روشن می‌شود. وقتی که نمایشگر نرمال بودن تغذیه روشن شد، می‌توان سیگنال SON (Servo ON) را که به ورودی دیجیتال DI1 سرودرایو متصل شده است را فعال کرد. اگر در این هنگام هیچ خطایی رخ ندهد، نمایشگر راه‌اندازی نرمال سرو Start normal که به خروجی دیجیتال DO2 سرودرایو متصل شده است، روشن می‌شود.

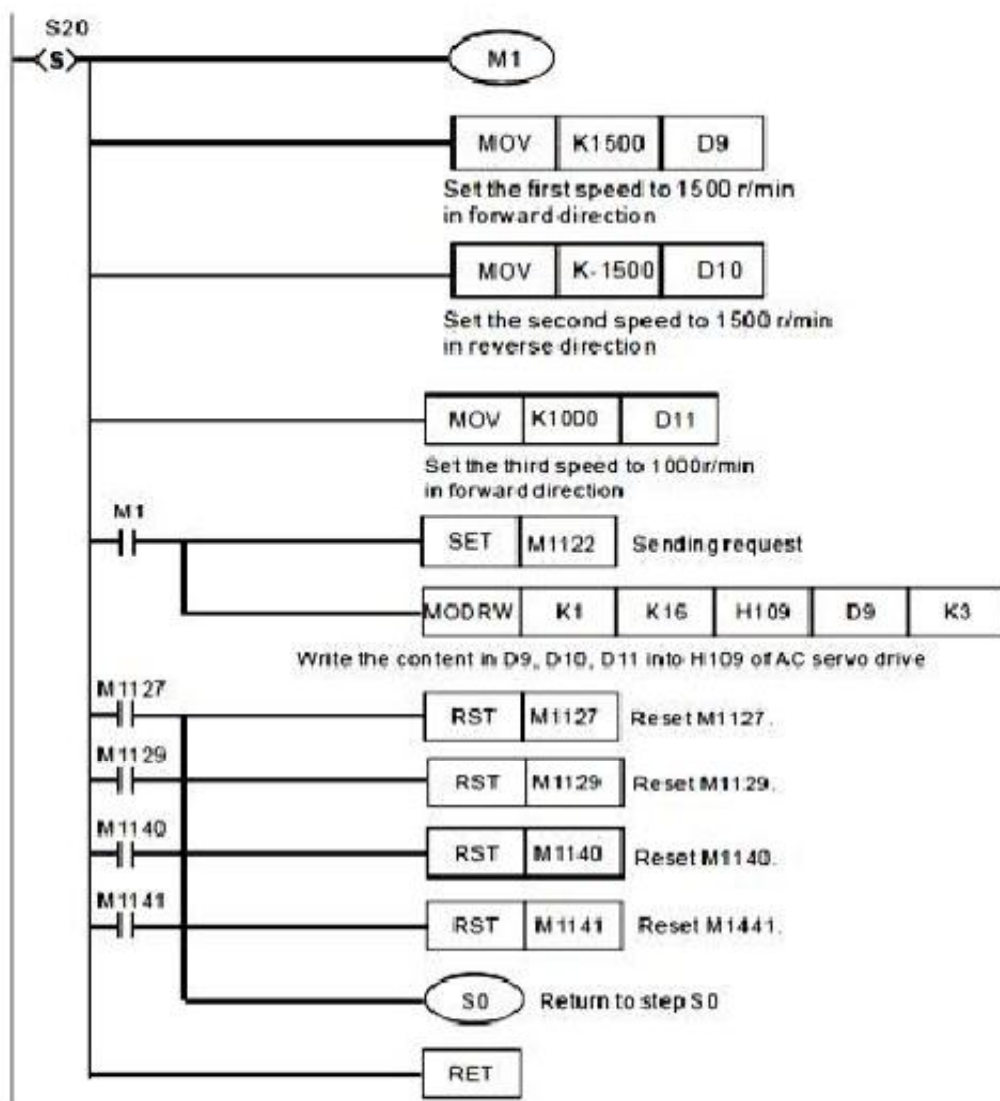
وقتی که SPO0 فعال شود، سرعت تنظیم شده در پارامتر P1-09 فعال خواهد شد. اگر ورودی SPD1 در سمت سرو فعال شود، سرعت تنظیم شده در پارامتر P1-10 سرودرایو فعال خواهد شد. اگر هر دو ورودی SPD0 و SPD1 با هم فعال شوند، سرعت تنظیم شده در پارامتر P1-11 سرودرایو فعال خواهد شد.

ابزارهای مورد استفاده در برنامه‌ی PLC دلتا برای این مثال مطابق با جدول زیر است:

ابزارها	توضیحات
M0	از این ابزار برای اجرای دستورالعمل MODRD جهت خواندن سرعت چرخش سروموتور از سرودرایو استفاده می‌شود.
M1	از این ابزار برای اجرای دستورالعمل MODWR استفاده می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:





عملکرد برنامه به شرح زیر است:

در ابتدا پورت ارتباطات RS-485 را در سمت PLC دلتا راه اندازی کرده و فرمت ارتباطات آن را برابر با 1, E, 7, 9600, ASCII MODBUS تنظیم کرده ایم. دقت کنید که فرمت ارتباطات تنظیم شده در سمت سرودرایو AC سری ASD-A دلتا نیز مانند فرمت ارتباطات PLC می باشد. وقتی که برنامه وارد مرحله S0 شود، ابزار M0=ON شده و دستورالعمل [MODRD K1 H4 K3] اجرا می شود. در این حالت، سرعت چرخش سروموتور توسط PLC از سرودرایو خوانده شده و در ابزار D1050 در سمت PLC ذخیره می شود. سپس دستورالعمل [MOV D1050 D0] اجرا شده و سرعت چرخش سروموتور در ابزار D0 نمایش داده می شود.

وقتی که برنامه وارد مرحله S20 شود، ابزار M1=ON شده و دستورالعمل [MODWR K1 09 H109 K6 اجرا می‌شود. در این حالت، محتویات داخل رجیسترهای D11, D10, D9 در سمت PLC در داخل پارامترهای H10B, H10A, H109 در سمت سرودرایو نوشته می‌شود. دقت کنید که عدد هگز H109 برای پارامتر P1-09 و عدد هگز H10A بیانگر پارامتر P1-10 و عدد هگز H10B بیانگر پارامتر P1-11 سرودرایو است. همچنین، در این دستورالعمل عدد دسیمال K1 بیانگر آدرس شماره گره‌ی ارتباطات شبکه‌ی ارتباطات سرودرایو است. در این مثال، عدد ثابت K1000 را در رجیستر D11 تنظیم کرده‌ایم. شما می‌توانید مقدار داخل ابزار D11 را مناسب با نیازتان تنظیم نمایید.

وقتی که PLC راه‌اندازی شود، ابتدا برنامه وارد مرحله S0 می‌شود سپس برنامه به مرحله S20 منتقل شده و در نهایت، بعد از اتمام برنامه‌ی مرحله S20 مجدداً برنامه‌ی مرحله S0 باز می‌گردد. توسط این فرآیند عمل خواندن و نوشتن برای سرودرایو AC مکرراً اجرا خواهد شد.

7- 18 برقراری ارتباط بین PLC دلتا و کنترلر دمای سری DTA دلتا با دستورالعمل MODRD/MODWR

در این مثال، می‌خواهیم بین PLC دلتا و کنترلر دمای سری DTA دلتا توسط دستورالعمل MODRD/ MODWR ارتباط شبکه برقرار کرده و تبادل اطلاعات نماییم. هدف از کنترل در این مثال به شرح زیر می‌باشد:

خواندن مقدار هدف PV و تنظیم مقدار نقطه مطلوب SP کنترلر دما با سنسور ورودی ترموکوپل (آدرس: H4700 و دستورالعمل مورد استفاده: MODRD)

تنظیم دمای هدف (نقطه مطلوب SV) برابر با 24°C (آدرس: H4701 و دستورالعمل مورد استفاده: MODWR)

تنظیم سیکل کنترل گرمایشی / سرمایشی (آدرس: H4712 و دستورالعمل مورد استفاده: MODWR)



تنظیم مد کنترل برابر با کنترل سرمایشی (آدرس: H4718 و دستورالعمل مورد استفاده: MODWR)



تنظیم پارامترهای کنترلر فرآیند سری DTA دلتا برای این مثال مطابق با جدول زیر است :



پارامتر	تابع	مقدار تنظیمی
CoSH	C WE : از این پارامتر برای فعال یا غیر فعال بودن تابع نوشتن در دستگاه مستر استفاده می‌شود.	ON
C-SL	C-SL : از این پارامتر برای انتخاب فرمت ارتباطات ASCII, RTU استفاده می‌شود.	ASCII
C-no	C NO : از این پارامتر برای تنظیم آدرس ارتباطات شبکه‌ی کنترلر استفاده می‌شود.	1
bPS	BPS : از این پارامتر برای تنظیم نرخ سرعت تبادل داده در شبکه استفاده می‌شود.	9600
LEn	LENGTH : از این پارامتر برای تنظیم طول داده‌ی ارتباطات استفاده می‌شود.	7
Prty	PARITY : از این پارامتر برای تنظیم بیت توازن شبکه استفاده می‌شود.	E
Stop	STOP BIT : از این پارامتر برای تنظیم بیت توقف استفاده می‌شود.	1
tPUn	UNIT : از این پارامتر برای تنظیم واحد اندازه‌گیری دما استفاده می‌شود. واحد قابل نمایش °C یا °F می‌باشد.	°C



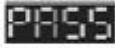

اگر کنترلر نمی‌تواند به صورت نرمال RUN شود، ممکن است این امر ناشی از تنظیمات نامناسب پارامترها باشد. می‌توانید پارامترهای کنترلر را ابتدا به تنظیمات پیش‌فرض کارخانه بازگردانید و سپس تنظیمات بالا را مجدداً انجام دهید.

مراحل بازگرداندن پارامترها به مقدار پیش‌فرض کارخانه برای کنترلر فرآیند سری DTA دلتا به شرح زیر است :

1. با فشردن دکمه  در صفحه‌ی اصلی کنترلر وارد صفحه  شوید. با استفاده از دکمه

 پارامتر  را انتخاب نمایید. با فشردن دکمه **SET** تنظیمات را ذخیره نمایید.

2. دکمه‌های  و  را برای 1 ثانیه با هم فشار دهید تا وارد مد تنظیمات پیش‌فرض شوید. (عملکردهای دیگر در این مد ممنوع شده‌اند. در غیر اینصورت، اگر تنظیمات در این مد انجام شود، خطایی رخ خواهد داد. در این حالت، کنترلر باید به تنظیمات پیش‌فرض کارخانه بازگردد)

3. در صفحه  با فشردن دکمه  پارامتر  را برابر با  تنظیم

نموده و با فشردن دکمه **SET** تنظیمات را ذخیره نمایید.

4. تغذیه کنترلر را قطع کرده سپس مجدداً وصل کنید.

5. پروتکل ارتباطات کنترلر فرآیند سری DTA دلتا را برای این مثال مطابق زیر تنظیم نمایید :

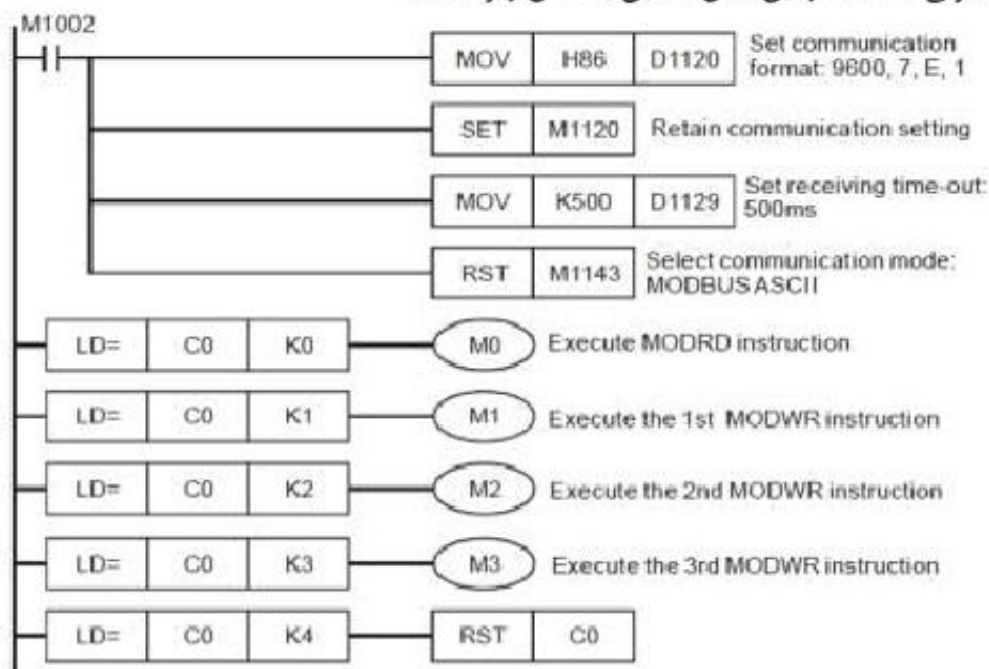
- این کنترلر از MODBUS ASCII/RTU. Baud rate: 2400, 4800, 9600, 19200, 38400 پشتیبانی می‌کند.

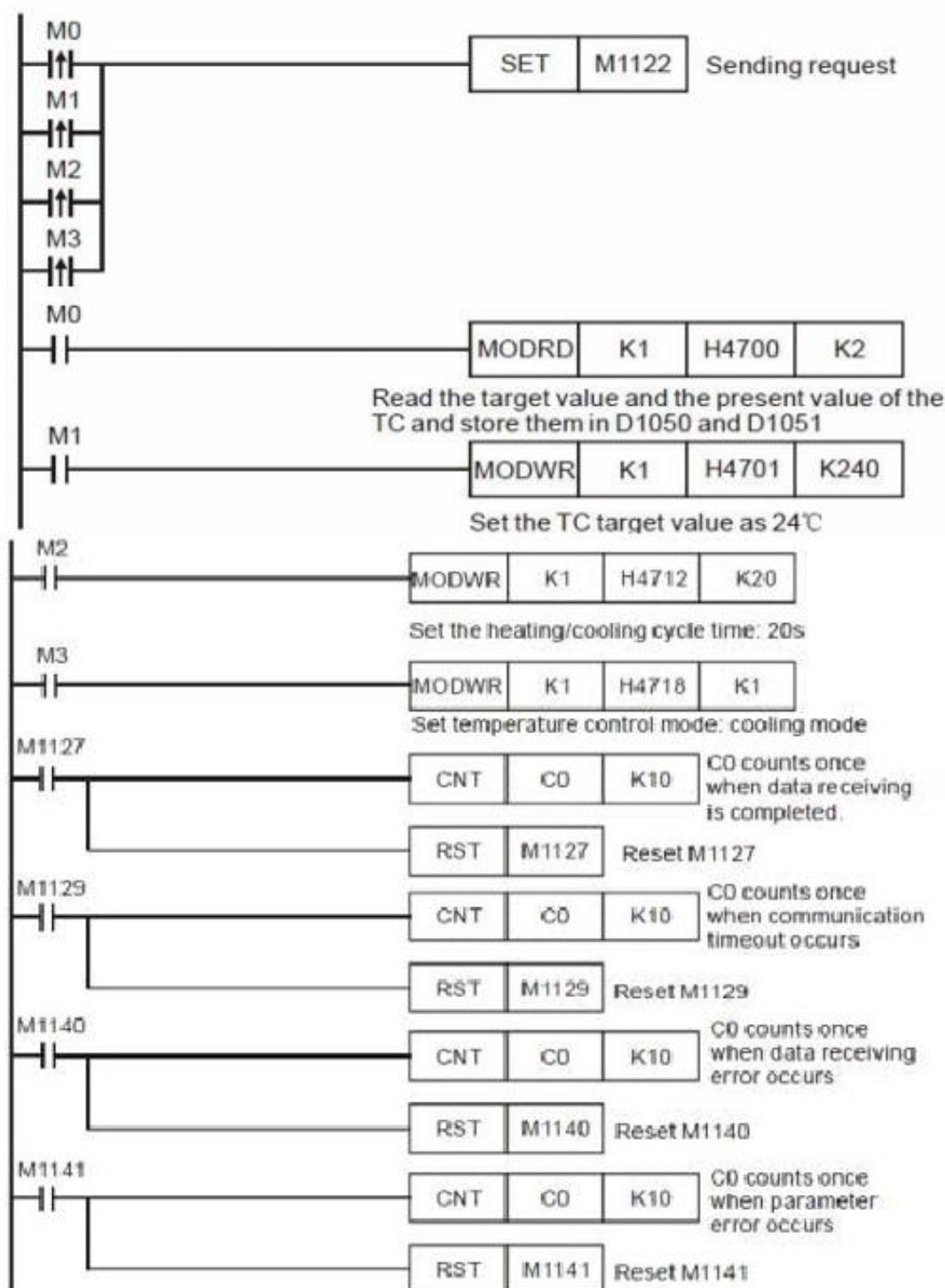
- این کنترلر از کدهای تابع 03H (read multiple Words). 06H (write 1 Word). پشتیبانی نموده و از کد تابع 10H (write multiple Words) پشتیبانی نمی‌کند.

- این کنترلر از فرمت‌های 1, N, 7 یا 2, O, 8 یا 2, E, 8 در مد ASCII پشتیبانی نمی‌کند.
- این کنترلر از فرمت‌های 1, N, 8 یا 2, N, 8 یا 1, O, 8 یا 1, E, 8 در مد RTU پشتیبانی می‌کند.
- شماره آدرس ارتباطات قابل دسترس برای کنترلر از 1 الی 255 می‌باشد و اگر آدرس ارتباطات برابر با 0 تنظیم شود، آدرس ارتباطات Broadcast می‌باشد.
- ابزارهای مورد نیاز برای برنامه‌نویسی PLC در این مثال مطابق جدول زیر است:

ابزارها	توابع
M0	از این فلگ عمومی برای اجرای دستورالعمل MODRD برای خواندن مقدار هدف و مقدار دمای جاری از کنترلر استفاده می‌شود.
M1	از این فلگ عمومی برای اجرای نخستین دستورالعمل MODWR برای تنظیم دمای هدف در داخل کنترلر استفاده می‌شود.
M2	از این فلگ عمومی برای اجرای دومین دستورالعمل MODWR برای تنظیم زمان سیکل گرمایشی/سرمایشی کنترلر استفاده می‌شود.
M3	از این فلگ عمومی برای اجرای سومین دستورالعمل MODWR برای تنظیم مد کنترل برابر با کنترل سرمایشی برای کنترلر استفاده می‌شود.

برنامه کنترلر PLC در این مثال مطابق با شکل زیر است:





توضیحات این برنامه به شرح زیر است:

PLC مورد نظر باید دارای پورت ارتباطات RS-485 باشد و تنظیمات فرمت ارتباطات در آن برای ارتباط با کنترلر DTA باید برابر با 1, E, 7, 9600, ASCII MODBUS تنظیم شوند. فرمت

ارتباطات شبکه RS-485 کنترلر فرآیند سری DTA دلتا باید مانند فرمت ارتباطات PLC تنظیم شود.

از آنجایی که کنترلر فرآیند سری DTA دلتا از کد تابع 10H (Write multiple Words) پشتیبانی نمی‌کند، بنابراین برنامه‌ی نوشته شده در PLC نیاز به 3 دستورالعمل MODWR برای نوشتن داده در 3 آدرس داده را دارد.

فقط 4 حالت برای ارتباطات MODBUS وجود دارد: فلگ M1127 برای ارتباطات نرمال و فلگ‌های M1129, M1140, M1141 برای خطاهای ارتباطات. وقتی که هر یک از این 4 فلگ فعال ON شوند، شمارنده C0 یک عدد شمارش می‌کند. بنابراین، برنامه‌ی ارتباطات قابل اعتمادی توسط مانیتور کردن وضعیت‌های On/Off هر یک از این 4 فلگ از 4 دستورالعمل MODBUS توسط مقدار شمارنده C0 وجود دارد.

زمانی که PLC شروع به کار کرد و یا در اصطلاح RUN شد، عمل خواندن و نوشتن برای کنترلر فرآیند بصورت مکرر مطابق با دستورالعمل‌های [LD=] اجرا می‌شود.

7-19 برقراری ارتباط بین PLC دلتا و کنترلر دمای سری DTB دلتا با دستورالعمل MODRD/MODWR/MODRW

در این مثال، می‌خواهیم بین PLC دلتا و کنترلر دمای سری DTB دلتا توسط دستورالعمل‌های MODRD/ MODWR/ MODRW ارتباط شبکه برقرار نموده و اطلاعات کنترلر را به PLC ارسال کرده و یا اطلاعات را از PLC به کنترلر ارسال نماییم.

هدف از کنترل در این مثال خواندن و نمایش مقدار هدف (نقطه مطلوب SP) و مقدار واقعی فرآیند PV از کنترلر توسط دستورالعمل MODRD است.

تنظیم پارامترهای کنترل فرآیند توسط دستورالعمل‌های MODWR و MODRW به شرح زیر

است :



پارامتر	مقدار	آدرس ارتباطات
تنظیم دمای هدف (نقطه مطلوب)	26°C	1001H
تنظیم محدوده‌ی بالای رنج دما	50°C	1002H
تنظیم محدوده‌ی پایین رنج دما	0°C	1003H
تنظیم نوع خروجی آلارم 1	نخستین نوع آلارم	1020H
تنظیم محدوده‌ی پایین آلارم 1	5°C	1024H
تنظیم محدوده‌ی پایین آلارم 1	3°C	1025H



تنظیم پارامترهای کنترل فرآیند سری DTB دلتا در این مثال به شرح زیر است:



پارامتر	تابع	مقدار تنظیمی
CWE	از این پارامتر برای فعال یا غیر فعال کردن تابع نوشتن در شبکه استفاده می‌شود.	ON
C-SL	از این پارامتر برای انتخاب فرمت ارتباطات ASCII و یا RTU استفاده می‌شود.	RTU
C-NO	از این پارامتر برای تنظیم آدرس ارتباطات شبکه‌ی کنترلر استفاده می‌شود.	1
BPS	از این پارامتر برای تنظیم نرخ سرعت تبادل داده در شبکه برای کنترلر استفاده می‌شود.	9600
LENGTH	از این پارامتر برای تنظیم طول داده استفاده می‌شود.	8
PARITY	از این پارامتر برای تنظیم بیت توازن شبکه استفاده می‌شود.	N
STOP BIT	از این پارامتر برای تنظیم بیت توقف استفاده می‌شود.	2
UNIT	از این پارامتر برای تنظیم واحد اندازه‌گیری دما استفاده می‌شود. واحد قابل نمایش °C یا °F می‌باشد.	°C

اگر کنترلر DTB نمی‌تواند به صورت نرمال RUN شود ممکن است این امر ناشی از تنظیمات نامناسب پارامترها باشد. می‌توانید پارامترهای کنترلر را ابتدا به پیش‌فرض کارخانه بازگردانید و سپس تنظیمات بالا را مجدداً انجام دهید.

مراحل بازگرداندن پارامترها به مقدار پیش‌فرض کارخانه برای کنترلر سری DTB دلتا به شرح زیر است:

1. با فشردن دکمه  در صفحه‌ی اصلی کنترلر وارد صفحه  شوید. با استفاده از دکمه

 پارامتر  را انتخاب کنید. با فشردن دکمه **SET** تنظیمات را ذخیره نمایید.

2. دکمه‌های  و  را برای 1 ثانیه با هم فشار دهید تا وارد مد تنظیمات پیش‌فرض شوید. عملکردهای دیگر در این مد ممنوع شده‌اند، در غیر این صورت با تنظیم مقدار در این مد خطایی رخ خواهد داد. در این حالت کنترلر باید به تنظیمات پیش‌فرض کارخانه بازگردد.

3. در صفحه  با فشردن دکمه  مقدار پارامتر  را برابر با  کنید.

تنظیم نموده و با فشردن دکمه **SET** تنظیمات را ذخیره کنید.

4. تغذیه کنترلر را قطع کرده سپس مجدداً وصل نمایید.

5. پروتکل ارتباطات کنترلر فرآیند سری DTB دلتا را مطابق زیر تنظیم کنید:

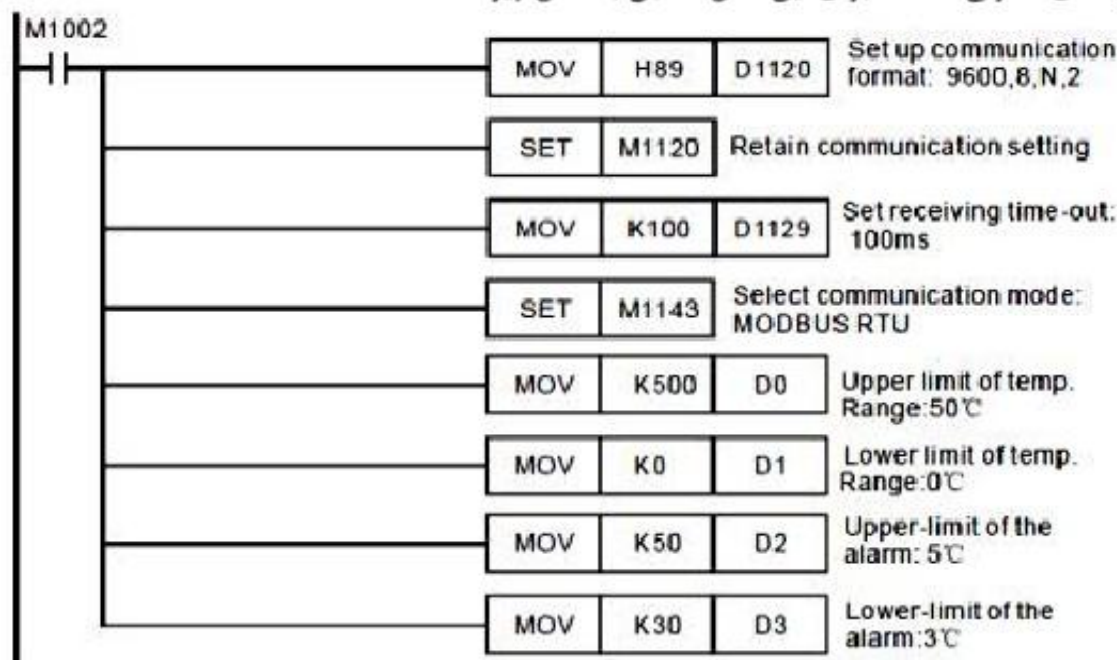
- این کنترلر از MODBUS ASCII/RTU. Baud rate: 2400, 4800, 9600, 19200, 38400 پشتیبانی می‌کند.

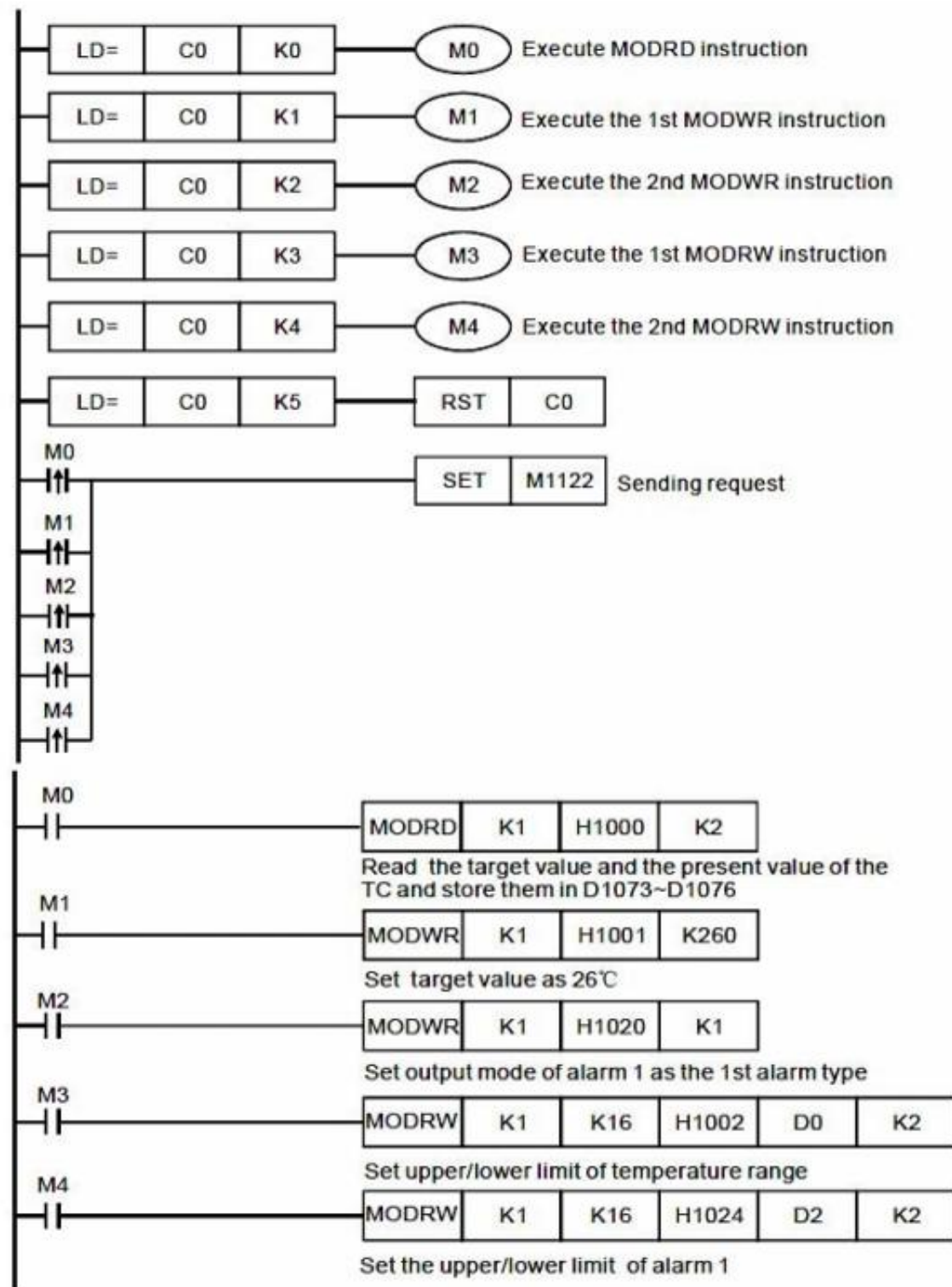
- این کنترلر از کدهای تابع 03H (read multiple Words). 06H (write 1 Word). 10H (write multiple Words) پشتیبانی می‌کند.

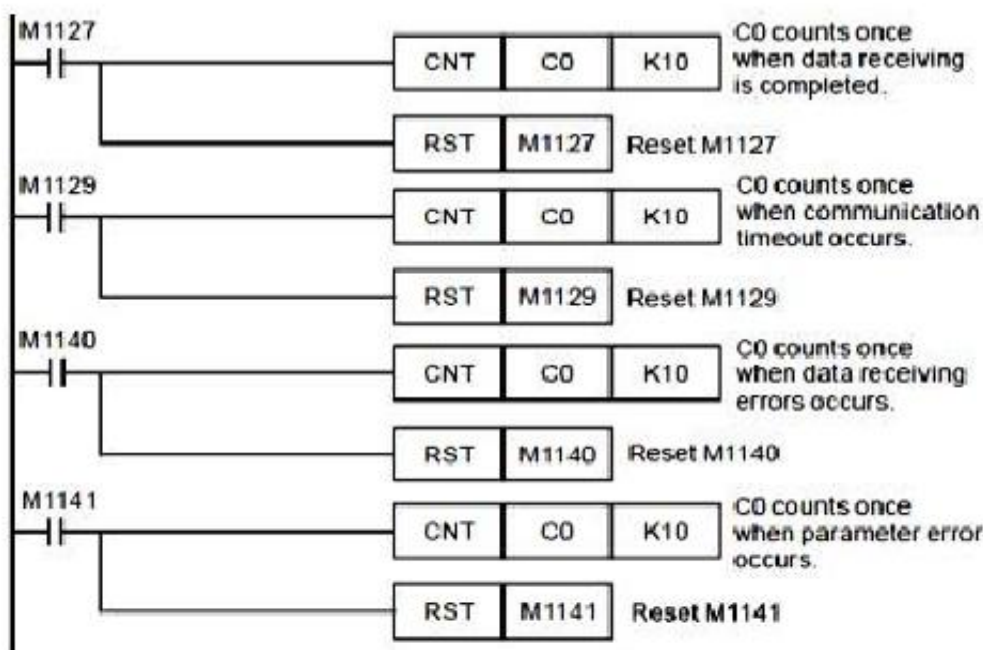
- این کنترلر از فرمت‌های 7, N, 1 یا 8, O, 2 یا 8, E, 2 در مد ASCII پشتیبانی نمی‌کند.
- این کنترلر از فرمت‌های 8, N, 1 یا 8, N, 2 یا 8, O, 1 یا 8, E, 1 در مد RTU پشتیبانی می‌کند.
- شماره آدرس ارتباطات قابل دسترس برای کنترلر از 1 الی 255 می‌باشد و اگر آدرس ارتباطات کنترلر برابر با 0 تنظیم شود، ارتباطات از نوع Broadcast خواهد شد.
- ابزارهای مورد نیاز برای برنامه‌نویسی در PLC برای این مثال به شرح زیر است :

ابزارها	توابع
M0	از این فلگ عمومی برای اجرای دستورالعمل MODRD برای خواندن مقدار هدف و مقدار دمای جاری از کنترلر DTB دلتا استفاده می‌شود.
M1	از این فلگ عمومی برای اجرای نخستین دستورالعمل MODWR برای تنظیم دمای هدف در کنترلر استفاده می‌شود.
M2	از این فلگ عمومی برای اجرا دومین دستورالعمل MODWR برای تنظیم نوع خروجی آلارم در کنترلر استفاده می‌شود.
M3	از این فلگ عمومی برای اجرای نخستین دستورالعمل MODWR برای تنظیم محدوده بالا و پایین رنج دما در کنترلر استفاده می‌شود.
M4	از این فلگ عمومی برای اجرای دومین دستورالعمل MODWR برای تنظیم محدوده بالا و پایین آلارم 1 در کنترلر استفاده می‌شود.

برنامه‌ی کنترلی PLC برای این مثال مطابق با شکل زیر است :







توضیحات برنامه‌ی نشان داده شده در بالا به شرح زیر می‌باشد:

PLC مورد نظر باید دارای پورت ارتباطات RS-485 باشد و تنظیمات فرمت ارتباطات باید برابر با MODBUS RTU, 9800, 8, N, 2 تنظیم شوند. فرمت ارتباطات شبکه RS-485 کنترلر فرآیند سری DTB دلتا باید مانند فرمت ارتباطات PLC تنظیم شود.

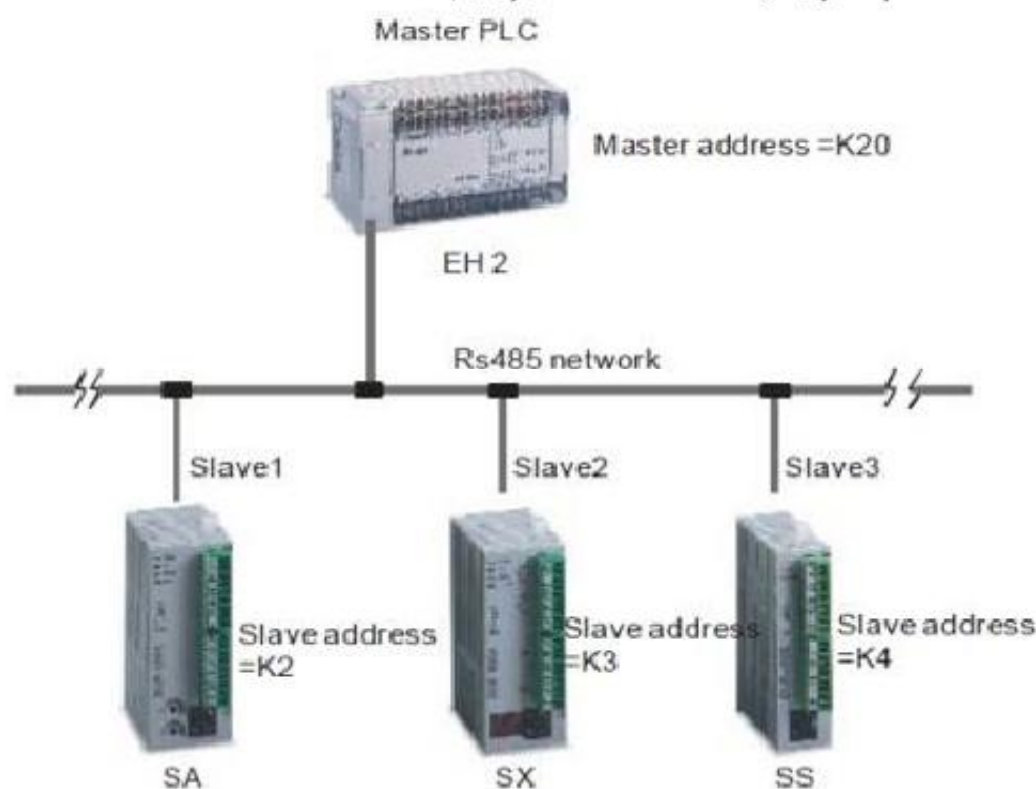
فقط 4 حالت برای ارتباطات MODBUS وجود دارد: فلگ M1127 برای ارتباطات نرمال و فلگ‌های M1129, M1140, M1141 برای خطاهای ارتباطات می‌باشند. وقتی که هر یک از این 4 فلگ فعال ON شوند، شمارنده C0 یک عدد شمارش می‌کند. بنابراین، برنامه ارتباطات قابل اعتمادی توسط مانیتور کردن وضعیت‌های On/Off هر یک از این 4 فلگ از 4 دستورالعمل MODBUS توسط مقدار شمارنده C0 نوشته‌ایم.

از آنجایی که کنترلر فرآیند سری DTB دلتا از کد تابع 10H (Write multiple Words) استفاده کرده‌ایم. در برنامه‌ی نوشته شده در PLC برای نوشتن چندین کلمه داده write multiple Words از دستورالعمل MODWR استفاده کرده‌ایم.

زمانی که PLC شروع به کار کرد (RUN شد)، عمل خواندن و نوشتن برای کنترلر فرآیند بصورت مکرر مطابق با دستورالعمل‌های [LD=] اجرا می‌شود.

20-7 برقراری ارتباط بین PLC سری E دلتا به عنوان Master با PLC سری S دلتا به عنوان Slave

در این مثال می‌خواهیم PLC سری EH2 دلتا را بعنوان Master PLC در شبکه قرار دهیم و سپس 16 ایستگاه بعنوان Slave PLC را به PLC سری EH2 متصل کنیم. در این مثال، از شبکه MODBUS استفاده کرده‌ایم. همچنین، می‌خواهیم در این شبکه 16 داده‌ی از نوع Word را توسط Master PLC خوانده و یا در Slave PLC بنویسیم.



هدف از کنترل در این مثال به شرح زیر می‌باشد:

قصد داریم بین PLC های لینک شده به یکدیگر تبادل داده داشته باشیم و 16 دیتا از نوع Word را بین Master PLC و 3 عدد Slave PLC ارسال و یا دریافت کنیم.

تنظیمات پارامترها برای PLC های تحت شبکه در این مثال به شرح زیر است:

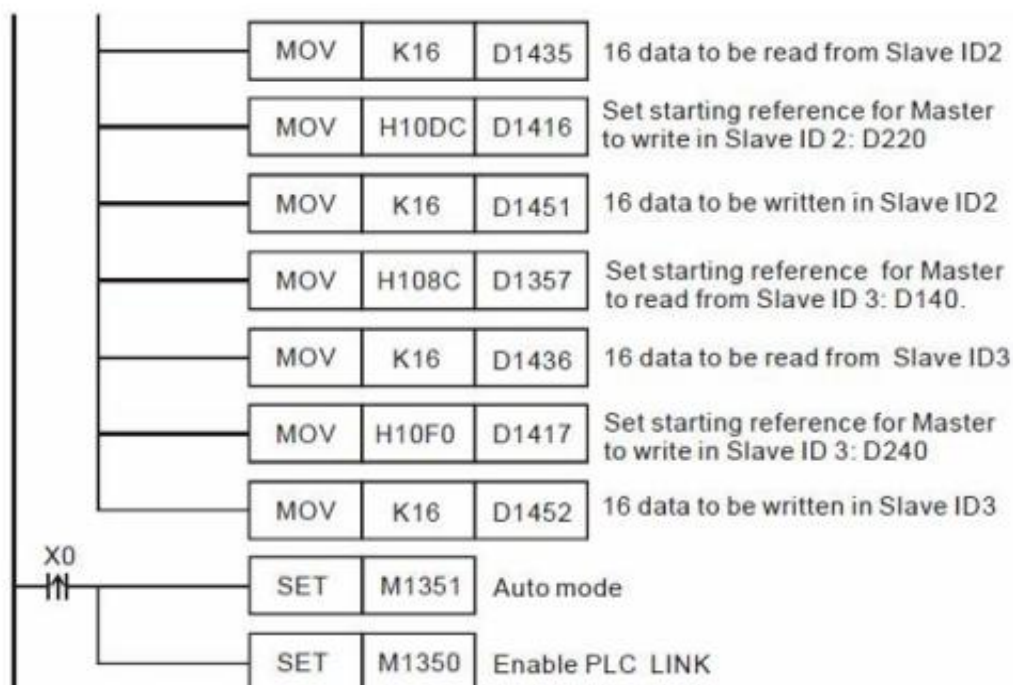
فرمت ارتباطات	شماره ایستگاه	Master/Slave
فرمت ارتباطات تمام PLC های متصل شده به شبکه باید با یکدیگر یکسان باشند. فرمت ارتباطات به شرح زیر می‌باشد: ASCII, 9600, 7, E, 1 (D1120=H86)	K20(D1121=K20)	Master PLC
	K2(D1121=K2)	Slave1
	K3(D1121=K3)	Slave2
	K4(D1121=K4)	Slave3

بعد از تنظیم پارامترهای PLC، اگر یک PLC به درستی راه‌اندازی نشد، ابتدا باید تنظیمات PLC را به تنظیمات پیش‌فرض کارخانه بازگردانده و سپس باید مجدداً تنظیمات را انجام دهید. جهت ریست کردن تنظیمات PLC باید در نرم‌افزار WPLSoft از منو Communication (C) بر روی گزینه Format PLC memory کلیک نمایید. برای تنظیم مجدد پارامترهای ارتباطات شبکه در PLC برای این مثال باید مطابق با جدول زیر عمل کنید:

ابزار	توضیحات
X0	از این فلگ عمومی برای راه‌اندازی شبکه PLC LINK استفاده می‌شود.
M1350	از این فلگ خاص برای فعال کردن ساده‌ی شبکه‌ی EASY PLC LINK استفاده می‌شود.
M1351	از این فلگ خاص برای فعال کردن مد اتوماتیک EASY PLC LINK استفاده می‌شود.
M1352	از این فلگ خاص برای فعال کردن مد دستی EASY PLC LINK استفاده می‌شود.
M1353	از این فلگ خاص برای فعال کردن 32 واحد Slave (از جمله PLC، اینورتر، سروسیستم، کنترل فرایند و ...) استفاده می‌شود.
M1354	از این فلگ خاص برای فعال کردن همزمان عمل خواندن و نوشتن دیتا با استفاده از نمونه‌برداری ساده از لینک‌های PLC متصل شده به شبکه استفاده می‌شود.

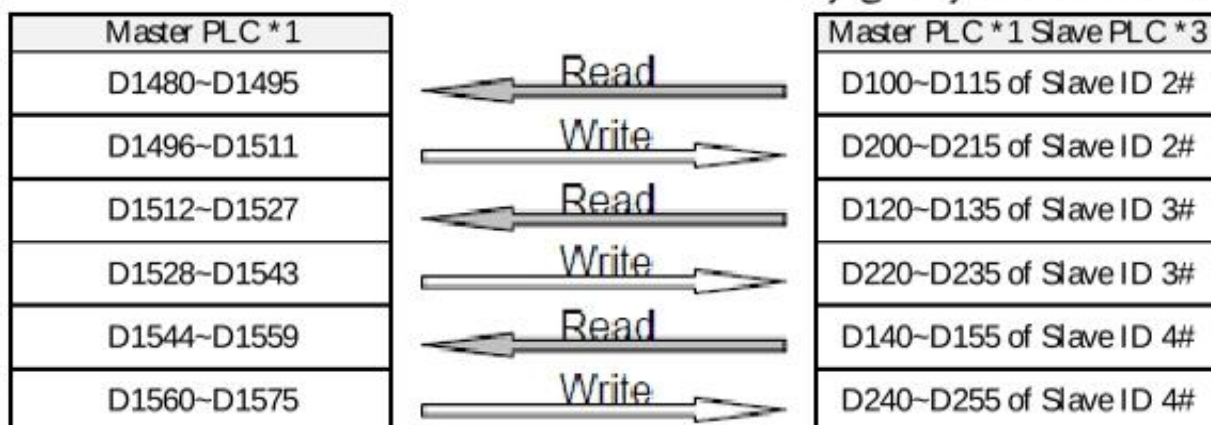
برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر می‌باشد:





عملکرد این برنامه به شرح زیر است:

وقتی که $X0=ON$ شود، تبادل داده بین Master PLC و 3 عدد Slave PLC صورت خواهد گرفت. در این حالت، باید PLC LINK فعال باشد که آن با لبه‌ی بالارونده‌ی ورودی $X1$ فعال می‌شود. داده‌های موجود در ابزارهای $D100\sim D115$ و $D120\sim D135$ و $D140\sim D141$ در هر سه Slave PLC توسط Master PLC مکرراً خوانده شده و در ابزارهای $D1580\sim D1495$ و $D1512\sim D1527$ و $D1544\sim D1559$ موجود در Master PLC نوشته می‌شوند. و داده‌های موجود در ابزارهای $D1496\sim D1511$ و $D1528\sim D1548$ و $D1560\sim D1575$ از Master PLC مکرراً داخل ابزارهای $D200\sim D215$ و $D220\sim D35$ و $D240\sim D255$ هر سه Slave PLC نوشته می‌شوند.



فرض کنید که داده‌های موجود در ابزارهای D برای تعادل داده بین Master PLC و Slave PLC ها قبل از فعال شدن PLC LINK (غیرفعال بودن ابزار M1350) مطابق با جدول زیر می‌باشد:

Master PLC	Set value	Slave PLC	Set value
D1480~D1495	0	D100~D115 of Slave ID#2	1
D1496~D1511	100	D200~D215 of Slave ID#2	0
D1512~D1527	0	D120~D135 of Slave ID#3	2
D1528~D1543	200	D220~D235 of Slave ID#3	0
D1544~D1559	0	D140~D155 of Slave ID#4	3
D1560~D1575	300	D240~D255 of Slave ID#4	0

بعد از فعال شدن EASY PLC LINK با استفاده از فعال کردن ابزار M1350، تبادل داده بین Master PLC و Slave PLC صورت گرفته و داده‌های موجود در ابزارهای D مطابق با جدول زیر خواهند شد:

Master PLC	Set value	Slave PLC	Set value
D1480~D1495	1	D100~D115 of Slave ID#2	1
D1496~D1511	100	D200~D215 of Slave ID#2	100
D1512~D1527	2	D120~D135 of Slave ID#3	2
D1528~D1543	200	D220~D235 of Slave ID#3	200
D1544~D1559	3	D140~D155 of Slave ID#4	3
D1560~D1575	300	D240~D255 of Slave ID#4	300

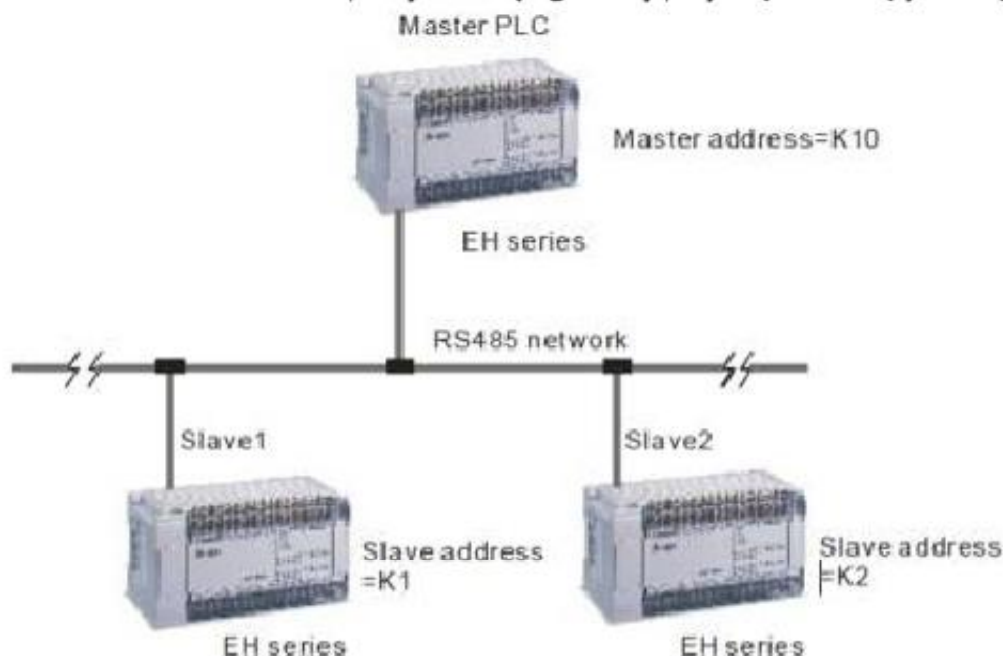
در Master PLC، اولین Slave ID برابر با (D1399=K2) تنظیم شده است. از این رو، Slave ID#2 برابر با Slave PLC1 و Slave ID#3 برابر با Slave PLC2 و Slave ID#4 برابر با Slave PLC3 می‌باشد.

شماره ایستگاه‌های Master PLC و Slave PLC ها باید با یکدیگر متفاوت باشند و در شبکه نباید از یک شماره برای دو ایستگاه استفاده شود. شماره ایستگاه‌های Master PLC و Slave PLC ها باید پشت سرهم و به صوت متوالی باشند. در محصولات دلتا فقط از PLC های سری SA/ SX/ SC/ SV/ EH/ EH2 می‌توان بعنوان Master PLC استفاده کرد، اما از تمام PLC های شرکت دلتا که شامل DVP-PLC می‌باشند، می‌توان بعنوان Slave PLC در شبکه‌ی MODBUS استفاده کرد.

وقتی که X0=ON شود، PLC LINK فعال خواهد شد. اگر عمل فعال کردن شبکه با مشکل مواجه شود، ابزارهای M1350/ M1351 غیرفعال خواهند شد. در این صورت، لازم است که برای فعال کردن شبکه مجدداً ابزار X0 را فعال کنید.

7- 21 برقراری ارتباط بین PLC های سری E دلتا به عنوان Master/Slave

در این مثال، می‌خواهیم با فعال کردن PLC LINC ایستگاه‌های Slave را فعال کرده و داده‌های موجود در 100 ابزار Word را بخوانیم و یا داخل آنها داده بنویسیم.



هدف از کنترل در این مثال به شرح زیر است :

می‌خواهیم دو ایستگاه Slave ID را به ایستگاه Master PLC متصل کنیم و با فعال کردن PLC LINK داده‌های موجود در 100 ابزار Word را بین PLC های تحت شبکه با یکدیگر تبادل نماییم.

تنظیمات پارامترهای PLC ها مطابق با جدول زیر می‌باشد :

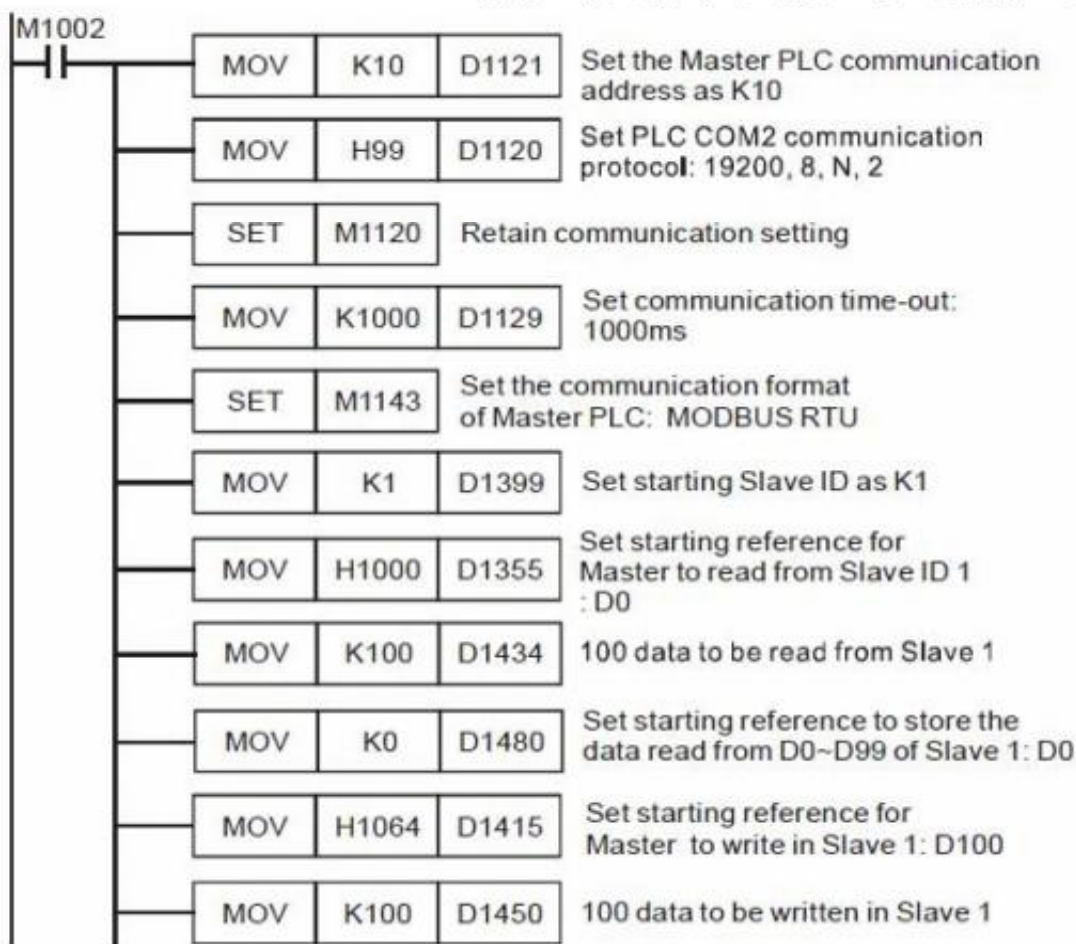
Master/Slave	Station No.	Communication format
Master PLC	K20 (D1121 = K20)	فرمت ارتباطات برای تمام PLC های متصل شده به شبکه باید یکسان باشد
Slave 1	K2 (D1121 = K2)	
Slave 2	K3 (D1121 = K3)	

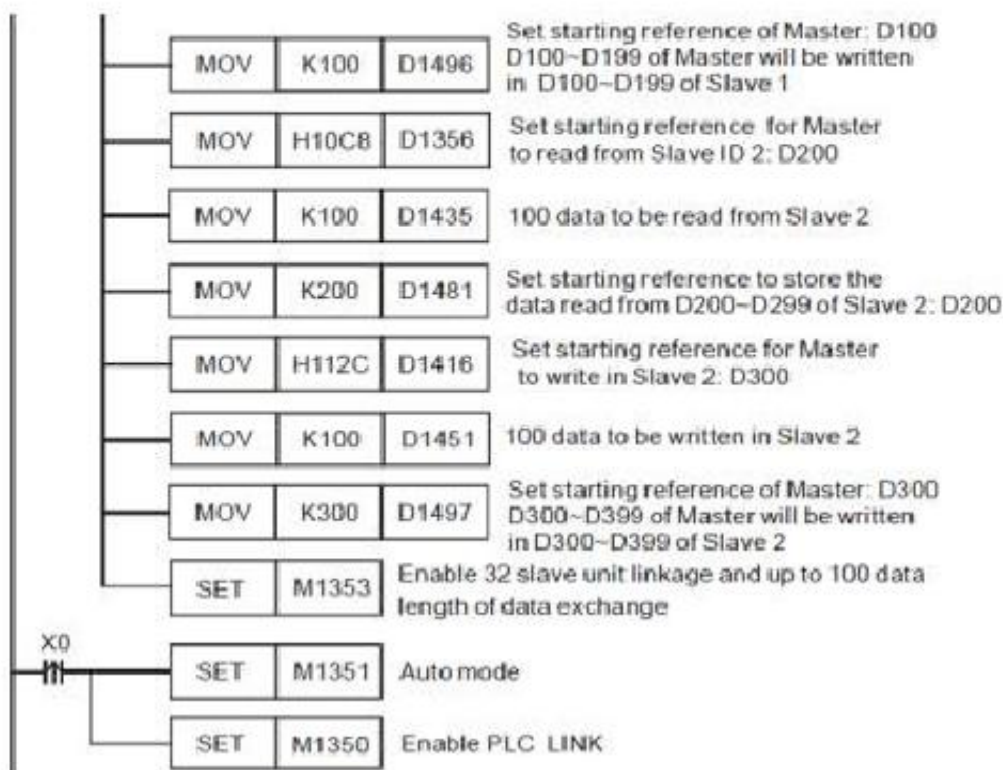
اگر بعد از تنظیمات شبکه‌ی PLC، PLC به صورت نرمال راه‌اندازی نشد، ابتدا باید تنظیمات PLC را به تنظیمات پیش‌فرض کارخانه بازگردانده و سپس مجدداً پارامترهای شبکه را مطابق جدول بالا تنظیم نمایید. برای فرمت کردن PLC دلتا باید از نرم‌افزار WPLSoft استفاده نمایید. در این نرم‌افزار باید از مسیر (C) Communication گزینه‌ی Format PLC memory را انتخاب نمایید.

ابزارهای مورد استفاده در این برنامه مطابق با جدول زیر می‌باشد:

ابزار	توضیحات
X0	از این ورودی برای راه‌اندازی شبکه PLC LINK استفاده می‌شود.
M1350	از این فلگ خاص برای فعال کردن ساده‌ی شبکه‌ی EASY PLC LINK استفاده می‌شود.
M1351	از این فلگ خاص برای فعال کردن مد اتوماتیک شبکه EASY PLC LINK استفاده می‌شود.
M1352	از این فلگ خاص برای فعال کردن مد دستی شبکه EASY PLC LINK استفاده می‌شود.
M1353	از این فلگ خاص برای پیوند دادن 32 واحد Slave به شبکه MODBUS استفاده می‌شود. طول داده برای تبادل دیتا در شبکه باید 100 data باشد.
M1354	از این فلگ خاص برای فعال کردن همزمان عمل خواندن/نوشتن داده در یک نمونه‌گیری از شبکه EASY PLC LINK استفاده می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر است:

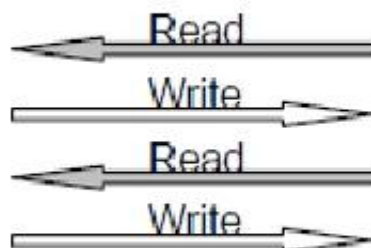




عملکرد برنامه به شرح زیر است:

وقتی که ورودی $X0=ON$ شود، از طریق اجرا شدن برنامه PLC LINK تبادل داده بین Master PLC و 3 عدد Slave PLC صورت خواهد گرفت. اطلاعات ارسالی و دریافتی در شبکه به شرح زیر می‌باشد: داده‌های موجود در ابزارهای $D0-D99$ در سمت Slave 1 توسط Master PLC خوانده شده و در ابزارهای $D0-D99$ ذخیره می‌شوند. داده‌های موجود در ابزارهای $D100-D199$ در سمت Master PLC در سمت Slave 1 نوشته می‌شوند. داده‌های موجود در ابزارهای $D200-D299$ در سمت Slave 2 توسط Master PLC خوانده شده و در ابزارهای $D200-D299$ ذخیره می‌شوند. داده‌های موجود در ابزارهای $D300-D399$ در Master PLC در ابزارهای $D300-D399$ در سمت Slave 2 نوشته می‌شوند.

Master PLC *1
D0-D99
D100-D199
D200-D299
D300-D399



Slave PLC *2
D0-D99 of Slave ID 1#
D100-D199 of Slave ID 1#
D200-D299 of Slave ID 2#
D300-D399 of Slave ID 2#

فرض کنید که داده‌های موجود در ابزارهای D قبل از فعال شدن PLC LINK برای تبادل داده بین Master PLC و Slave PLC ها مطابق با جدول زیر می‌باشد. برای غیرفعال کردن PLC LINK باید ابزار M1350 غیرفعال باشد (M1350=OFF).

Master PLC	Set value	Slave PLC	Set value
D0~D99	0	D0~D99 of Slave ID#1	1
D100~D199	100	D100~D199 of Slave ID#1	0
D200~D299	0	D200~D299 of Slave ID#2	2
D300~D399	200	D300~D399 of Slave ID#2	0

حال بعد از فعال شدن EASY PLC LINK تبادل داده بین شبکه صورت خواهد گرفت و مقدار ابزارهای D مطابق با جدول زیر خواهند شد. برای فعال کردن EASY PLC LINK باید ابزار M1350 را فعال کرد. (M1350=ON).

Master PLC	Set value	Slave PLC	Set value
D0~D99	0	D0~D99 of Slave ID#1	1
D100~D199	100	D100~D199 of Slave ID#1	100
D200~D299	0	D200~D299 of Slave ID#2	2
D300~D399	200	D300~D399 of Slave ID#2	200

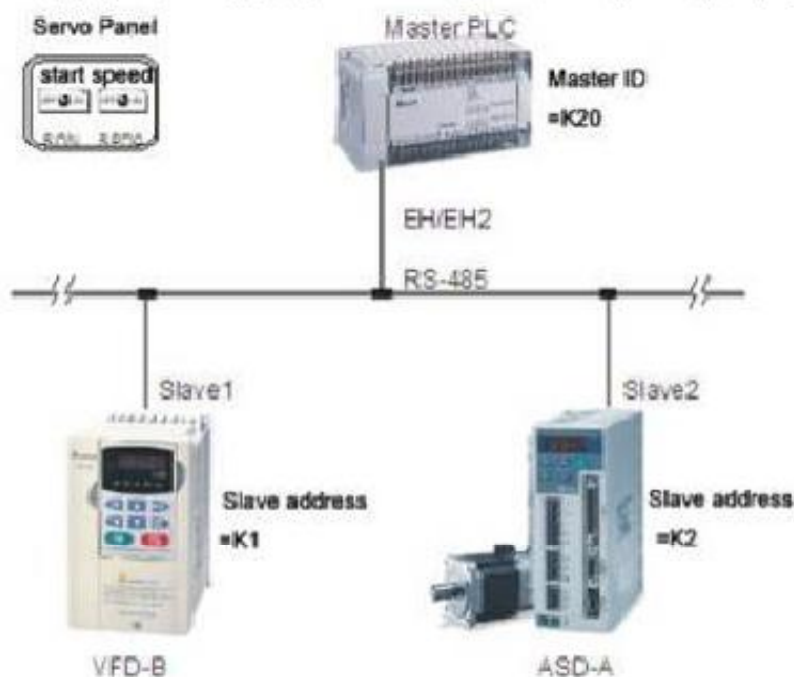
در Master PLC، آدرس شروع شبکه از Slave ID (D1399=K1) می‌باشد. بنابراین شماره شبکه برای Slave PLC1 برابر با Slave ID#1 بوده و شماره شبکه برای Slave PLC2 برابر با Slave ID#2 می‌باشد. شماره‌ی ایستگاه‌های Slave باید به صورت پشت سرهم و متوالی باشند. همچنین، این شماره‌ها باید با یکدیگر متفاوت بوده و یک شماره در شبکه نباید دوبار تکرار شود. اولین شماره باید به Master PLC و شماره‌های بعدی باید به ترتیب به Slave PLC ها اختصاص داده شود.

در محصولات دلتا فقط از PLC های سری SA/ SX/ SC/ SV/ EH/ EH2 می‌توان بعنوان Master PLC در شبکه‌ی MODBUS استفاده کرد. دیگر PLC ها را می‌توان بعنوان Slave PLC در شبکه‌ی MODBUS استفاده کرد.

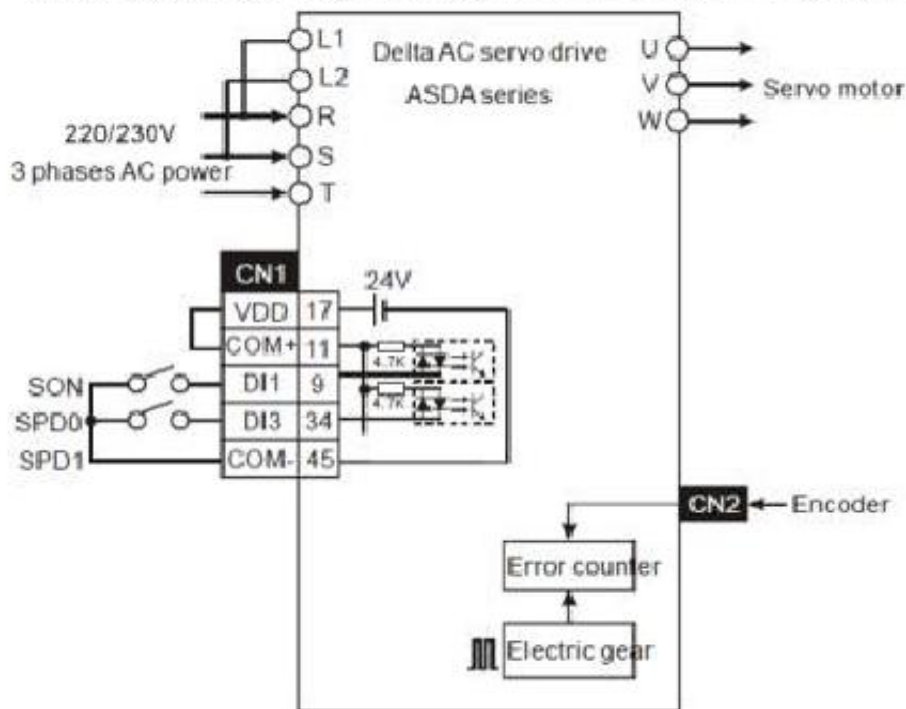
وقتی که ورودی X0=ON شود، لینک ارتباط شبکه PLC LINK فعال خواهد شد. اگر در شبکه خطایی به وجود آید، ابزارهای M1350/M1351=OFF غیرفعال خواهند شد. در این صورت، لازم است که ابزار X0 غیرفعال شده و سپس مجدداً فعال (ON) شود.

22-7 برقراری ارتباط بین PLC دلتا با اینورتر سری VFD-B و سروودرایو سری ASD-A دلتا توسط شبکه Modbus

در این مثال، می‌خواهیم یک ارتباط شبکه بین PLC دلتا و اینورتر دلتا و سروودرایو دلتا برقرار کنیم.



سیم بندی سروودرایو AC سری ASD-A دلتا برای این مثال مطابق شکل زیر می‌باشد:



هدف از کنترل در این مثال به شرح زیر است :

در این مثال، می‌خواهیم فرکانس اینورتر را از طریق PLC تنظیم کرده و بخوانیم و همچنین Start/ Stop آن را نیز از طریق شبکه کنترل کنیم و در نهایت، وضعیت چپگرد و یا راستگرد بودن را نیز از طریق شبکه‌ی MODBUS که یک ارتباط بین PLC و اینورتر است، کنترل نماییم.

همچنین، می‌خواهیم سرعت چرخش سروموتور را نیز از طریق شبکه خوانده و یا تنظیم کنیم. تنظیمات پارامترهای اینورتر سری VFD-B دلتا برای راه‌اندازی از طریق شبکه‌ی MODBUS مطابق جدول زیر است :

پارامتر	مقدار تنظیمی	توضیحات
P02-00	04	شبکه‌ی ارتباطات سریال RS-485. این پارامتر برای ذخیره‌سازی آخرین فرکانس اینورتر رزرو شده است.
P02-01	03	شبکه‌ی ارتباطات سریال RS-485. این پارامتر جهت فعال کردن STOP/RESET کی‌پد می‌باشد.
P09-00	01	آدرس شبکه‌ی ارتباطات اینورتر (گره) برابر با 01 تنظیم شده است.
P09-01	01	سرعت تبادل داده‌ی شبکه‌ی ارتباطات MODBUS برابر با 9600 تنظیم شده است.
P09-04	01	مد شبکه‌ی ارتباطات MODBUS ASCII mode تنظیم شده و پروتکل ارتباطات آن برابر با 7, E, 1 تنظیم شده است.

اگر بعد از تنظیمات پارامترهای اینورتر، موتور به صورت نرمال در شبکه راه‌اندازی نشد، باید ابتدا پارامترهای اینورتر را به پیش‌فرض کارخانه بازگردانده و سپس مجدداً آنها را مطابق با جدول بالا تنظیم نمایید. برای ریست کردن تنظیمات اینورتر سری VFD-B باید پارامتر P00-02 را برابر با 10 تنظیم نمایید.

تنظیمات پارامترهای سرودرایو AC سری ASD-A دلتا برای کار در شبکه MODBUS مطابق با جدول زیر می‌باشد :

پارامتر	مقدار تنظیمی	توضیحات
P0-02	6	وضعیت سرودرایو. نمایش سرعت چرخش سروموتور بر روی پنل سرودرایو.
P0-04	6	وضعیت مانیتورینگ 1. انتقال داده‌ی مربوط به رجیستر داده‌ی سرعت چرخش جاری سروموتور بر روی Monitor 1
P1-01	2	تنظیم مد کنترل و جهت خروجی
P2-02	101	تنظیم تابع اولین ورودی دیجیتال (DI1)
P2-12	114	تنظیم تابع سومین ورودی دیجیتال (DI3)
P2-15~17	0	تنظیم تابع ششمین ورودی دیجیتال (DI6). هیچ تابعی برای این ورودی‌ها تنظیم نشده است.
P3-00	2	تنظیم آدرس ارتباطات شبکه‌ی MODBUS (گره) سرودرایو برابر با 2
P3-01	1	سرعت تبادل داده‌ی ارتباطات شبکه‌ی MODBUS سرودرایو برابر با 9600 تنظیم شود.
P3-02	1	مد ارتباطات شبکه باید MODBUS ASCII mode تنظیم شده و فرمت داده‌ی ارتباطات باید 7, E, 1 تنظیم شود.
P3-05	2	توسط این پارامتر باید فرمت ارتباطات RS-485 تنظیم شود.

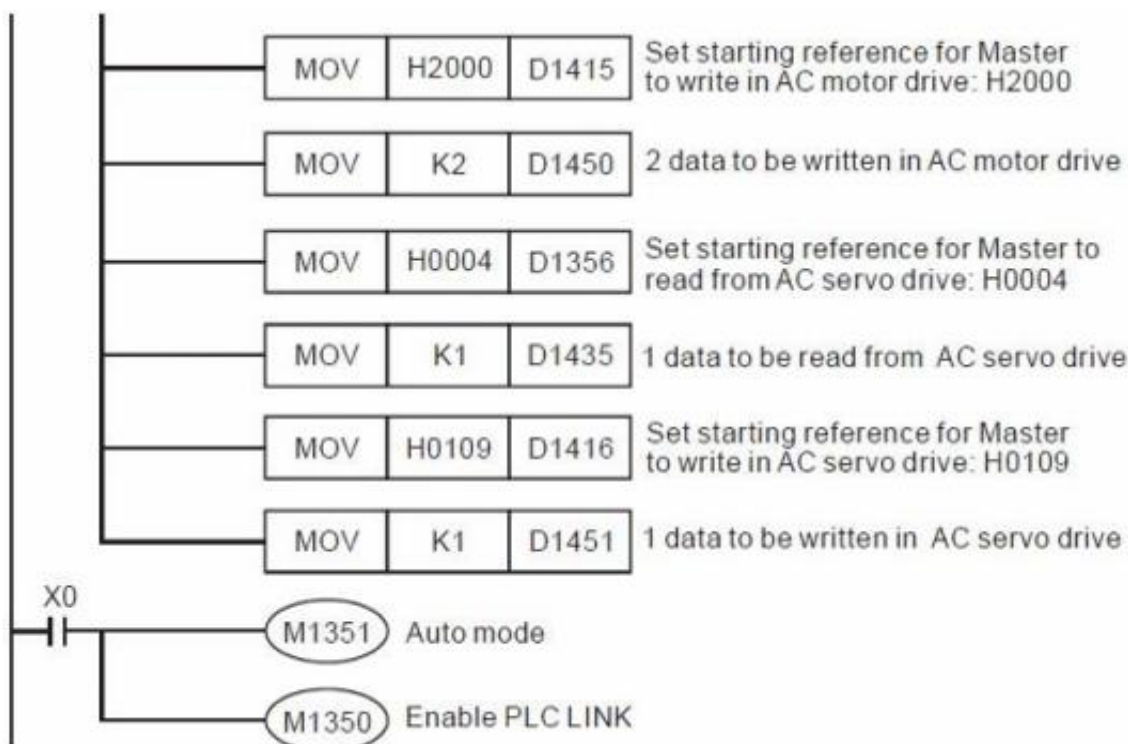
اگر پارامترهای سرودرایو به درستی تنظیم نشود، سروموتور به صورت نرمال نمی‌چرخد. در این صورت، باید مقدار پارامترهای سرودرایو به تنظیمات پیش‌فرض کارخانه بازگردانده شده و سپس مجدداً تنظیمات سرودرایو مطابق با جدول بالا تنظیم شود. برای ریست کردن تنظیمات سرودرایو باید پارامتر P2-08 را برابر با 10 تنظیم نمایید.

ابزارهای مورد استفاده در این برنامه به شرح زیر می‌باشند:

ابزار	توضیحات
X0	از این ورودی برای راه‌اندازی شبکه‌ی PLC LINK استفاده می‌شود.
M1350	از این فلگ خاص برای فعال کردن شبکه‌ی EASY PLC LINK استفاده می‌شود.
M1351	از این فلگ خاص برای فعال کردن مد اتوماتیک شبکه‌ی EASY PLC LINK استفاده می‌شود.
M1352	از این فلگ خاص برای فعال کردن مد دستی شبکه‌ی EASY PLC LINK استفاده می‌شود.
M1353	توسط این فلگ خاص می‌توان 32 واحد Slave متصل شده به شبکه را فعال کرده و طول 100 داده را در شبکه تبادل دیتا کرد.
M1354	با استفاده از این فلگ خاص می‌توان در هر بار نمونه‌برداری EASY PLC LINK مقادیر دیتا را به صورت همزمان خواند و نوشت.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر است:





عملکرد برنامه به شرح زیر است:

برای خواندن پارامترهای H2102~H2103 (به فرمت هگزی) اینورتر سری VFD-B دلتا توسط PLC از رجیسترهای D1480~D1481 در برنامه‌نویسی PLC دلتا استفاده شده است. وقتی که ورودی X0=ON شود، شبکه‌ی PLC LINK فعال شده و مقادیر داخل پارامترهای H2102~H2103 اینورتر توسط PLC خوانده شده و در ابزارهای D1480~D1481 ذخیره خواهد شد.

برای خواندن مقادیر پارامترهای H2000~H2001 (به فرمت هگزی) سرودرایو AC سری ASD-A دلتا توسط PLC دلتا، در برنامه‌نویسی PLC از رجیستر D1496~D1497 استفاده شده است. وقتی که ورودی X0=ON شود، شبکه‌ی PLC LINK فعال شده و مقادیر داخل پارامترهای H2000~H2001 سرودرایو توسط PLC خوانده شده و در رجیسترهای D1496~D1497 ذخیره خواهند شد.

وضعیت اینورتر را می‌توان توسط تنظیمات مقادیر داخل D1496 کنترل کرد. از این رو، اگر D1496=H12 شود، اینورتر در جهت راستگرد راه‌اندازی (Run) شده و اگر D1496=H1 شود، اینورتر متوقف (Stop) خواهد شد. فرکانس اینورتر را می‌توان توسط تنظیمات مقادیر داخل D1497

تغییر داد. از این رو، اگر $D1497=K4000$ تنظیم شود، این به آن معنی است که فرکانس خروجی اینورتر 40Hz خواهد بود.

قبل از فعال کردن شبکه‌ی PLC LINK بین PLC و سروموتور، ابتدا سیگنال SON و سیگنال SDO0 را در سمت سرودرایو فعال کنید تا از کنترل شدن مد سرعت سروسیستم توسط رجیسترهای داخلی مطمئن شوید.

مقدار رجیستر D1512 در PLC دلتا مطابق با پارامتر ارتباطات H0004 (به فرمت هگز) سرودرایو AC دلتا می‌باشد. وقتی که ابزار $X0=ON$ شود، شبکه‌ی PLC LINK فعال شده و مقدار پارامتر H0004 سرودرایو توسط PLC خوانده شده و در ابزار D1512 ذخیره می‌شود.

رجیستر D1528 در PLC برای پارامتر ارتباطات H0109 سرودرایو AC رزرو شده است. وقتی که ابزار $X0=ON$ شود، شبکه‌ی PLC LINK فعال شده و توسط ابزار D1528 می‌توان مقداری را در پارامتر H0109 سرودرایو نوشت.

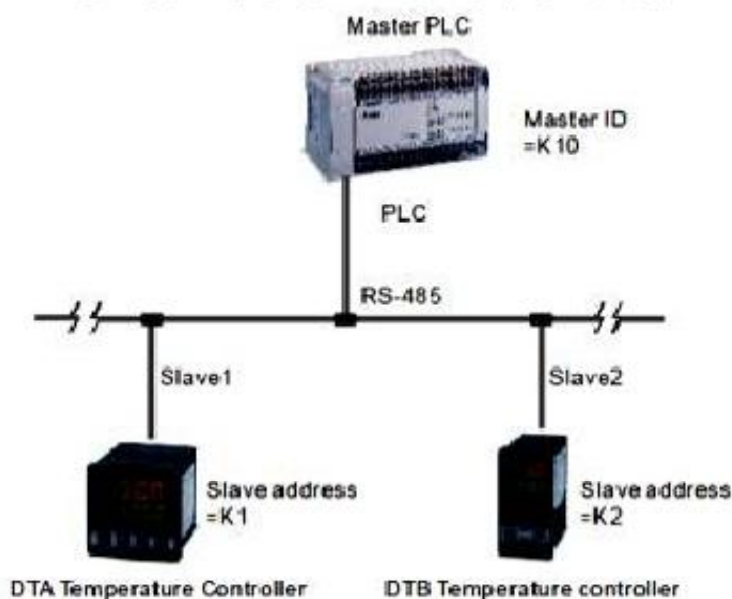
سرعت چرخش سروموتور را می‌توان توسط محتویات داخل ابزار D1528 تغییر داد. وقتی که $D1528=K3000$ تنظیم شود، این به آن معنی است که سرعت چرخش سروموتور برابر با 3000 rpm خواهد بود.

اولین شماره ایستگاه را باید به ایستگاه Master PLC اختصاص داده و سپس شماره ایستگاه‌های بعدی را باید به ایستگاه‌های Slave اختصاص داد. دقت کنید که شماره ایستگاه‌های شبکه باید پشت سرهم و به صورت متوالی باشد. اگر در شبکه از یک شماره برای دو یا چند ایستگاه استفاده شود، شبکه خطا داده و متوقف می‌شود. از PLC های سری SA/ SX/ SC/ SV/ EH/ EH2 دلتا می‌توان بعنوان Master PLC و یا Slave PLC استفاده کرد ولی از PLC های سری ES/ EX/ SS دلتا فقط بعنوان Slave PLC می‌توان استفاده نمود.

وقتی که ورودی $X0=ON$ شود، شبکه‌ی PLC LINK فعال خواهد شد. اگر در هنگام فعال شدن شبکه و یا در زمان فعال بودن شبکه خطایی رخ دهد، ابزارهای M1350/M1351 غیرفعال خواهند شد. در این هنگام برای ریست کردن خطای شبکه ابتدا باید ابزار $X0=OFF$ شود، سپس مجدداً $X0=ON$ شود.

23-7 برقراری ارتباط بین PLC دلتا با کنترلر فرآیند سری DTA و DTB دلتا توسط شبکه‌ی Modbus

در این مثال، می‌خواهیم بین PLC دلتا و کنترلر دمای سری DTB دلتا و DTA دلتا ارتباط شبکه‌ی MODBUS برقرار کرده و سپس بین PLC و این دو کنترلر تبادل داده نماییم:



هدف از کنترل در این مثال به شرح زیر است:

تنظیم دمای هدف و خواندن دمای واقعی (فرآیند) و دمای تنظیم شده (نقطه مطلوب) از کنترلر فرآیند DTA و همچنین، تنظیم دمای هدف، محدوده بالا و پایین رنج دما و خواندن دمای واقعی (فرآیند) و دمای تنظیم شده (نقطه مطلوب) از کنترلر فرآیند DTB از اهداف این مثال است.

تنظیم پارامترهای کنترل فرآیند سری DTA دلتا به شرح زیر است:



پارامتر	تابع	مقدار تنظیمی
CO5H	C WE: فعال یا غیر فعال کردن تابع نوشتن در شبکه توسط این پارامتر تنظیم می‌شود.	ON
C-SL	C-SL: انتخاب فرمت ارتباطات ASCII یا RTU توسط این پارامتر تنظیم می‌شود.	ASCII
C-no	C NO: آدرس ارتباطات شبکه برای کنترلر توسط این پارامتر تنظیم می‌شود.	1
bPS	BPS: نرخ سرعت تبادل داده در شبکه برای کنترلر توسط این پارامتر تنظیم می‌شود.	9600
LEn	LENGTH: طول داده برای کنترلر توسط این پارامتر تنظیم می‌شود.	7
Prty	PARITY: بیت توازن شبکه برای کنترلر توسط این پارامتر تنظیم می‌شود.	E
StoP	STOP BIT: بیت توقف برای کنترلر توسط این پارامتر تنظیم می‌شود.	1
EPUn	UNIT: واحد اندازه‌گیری دما برای کنترلر توسط این پارامتر تنظیم می‌شود. واحد قابل نمایش °C یا °F می‌باشد.	°C

اگر کنترلر نمی‌تواند به صورت نرمال RUN شود، ممکن است این امر ناشی از تنظیمات نامناسب پارامترها باشد. شما می‌توانید پارامترهای کنترلر را ابتدا به پیش‌فرض کارخانه بازگردانده و سپس تنظیمات بالا را مجدداً انجام دهید. کنترلر DTA از تابع نوشتن چند کلمه‌ی داده writing multiple Words پشتیبانی نمی‌کند، بنابراین تعداد نوشتن داده باید برابر 1 تنظیم شود. تنظیم پارامترهای کنترلر فرآیند سری DTB دلتا به شرح زیر است:



پارامتر	تابع	مقدار تنظیمی
CoSH	CWE: فعال یا غیر فعال کردن تابع نوشتن در شبکه توسط این پارامتر تنظیم می‌شود.	ON
C-SL	C-SL: انتخاب فرمت ارتباطات ASCII یا RTU توسط این پارامتر تنظیم می‌شود.	ASCII
C-no	C NO: تنظیم آدرس ارتباطات شبکه برای کنترلر توسط این پارامتر تنظیم می‌شود.	1
bPS	BPS: تنظیم نرخ سرعت تبادل داده در شبکه برای کنترلر توسط این پارامتر تنظیم می‌شود.	9600
LEn	LENGTH: تنظیم طول داده برای کنترلر توسط این پارامتر تنظیم می‌شود.	7
Prty	PARITY: تنظیم بیت توازن شبکه برای کنترلر توسط این پارامتر تنظیم می‌شود.	E
Stop	STOP BIT: تنظیم بیت توقف برای کنترلر توسط این پارامتر تنظیم می‌شود.	1
tPUn	UNIT: تنظیم واحد اندازه‌گیری دما برای کنترلر توسط این پارامتر تنظیم می‌شود. واحد قابل نمایش °C یا °F می‌باشد.	°C

اگر کنترلر نمی‌تواند به صورت نرمال RUN شود ممکن است این امر ناشی از تنظیمات نامناسب پارامترها باشد. شما می‌توانید پارامترهای کنترلر را ابتدا به پیش‌فرض کارخانه بازگردانده و سپس تنظیمات بالا را مجدداً انجام دهید.

مراحل بازگرداندن پارامترها به مقدار پیش‌فرض کارخانه برای کنترلر فرآیندهای دلتا به شرح زیر است:

1. با فشردن دکمه  در صفحه‌ی اصلی کنترلر وارد صفحه  شوید. با استفاده از دکمه

قسمت  را انتخاب نمایید. با فشردن دکمه **SET** تنظیمات را ذخیره کنید.

2. دکمه‌های  و  را برای 1 ثانیه با هم فشار دهید تا وارد مد تنظیمات پیش‌فرض شوید. عملکردهای دیگر در این مد ممنوع شده‌اند، در غیر اینصورت با تنظیم مقادیر در این مد خطایی رخ خواهد داد. در این حالت، کنترلر باید به تنظیمات پیش‌فرض کارخانه بازگردد.

3. در صفحه  با فشردن دکمه  پارامتر  را برابر با  تنظیم نمایید و با فشردن دکمه **SET** تنظیمات را ذخیره کنید.

4. تغذیه کنترلر را قطع کرده سپس مجدداً وصل نمایید.

5. پروتکل ارتباطات کنترلر فرآیند سری DTB دلتا را مطابق زیر تنظیم کنید:

- این کنترلر از MODBUS ASCII/RTU. Baud rate: 2400, 4800, 9600, 19200, 38400 پشتیبانی می‌کند.

- این کنترلر از کدهای تابع 03H (read multiple Words). 06H (write 1 Word). 10H (write multiple Words) پشتیبانی می‌کند.

- این کنترلر از فرمت‌های 1, N, 7 یا 2, O, 8 یا 2, E, 8 در مد ASCII پشتیبانی نمی‌کند.

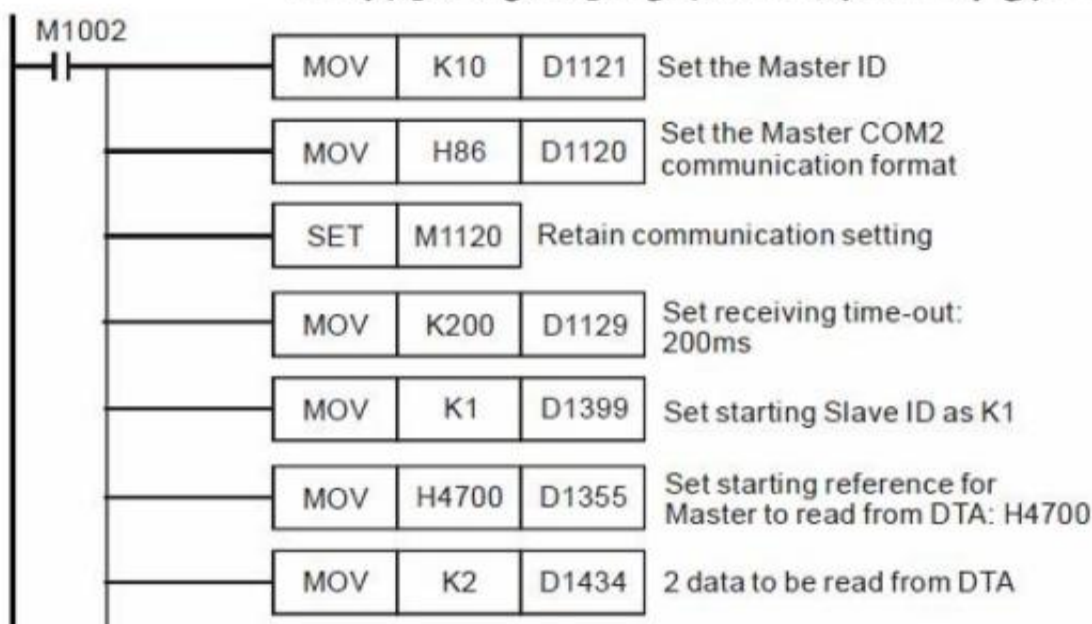
- این کنترلر از فرمت‌های 1, N, 8 یا 2, N, 8 یا 1, O, 8 یا 1, E, 8 در مد RTU پشتیبانی می‌کند.

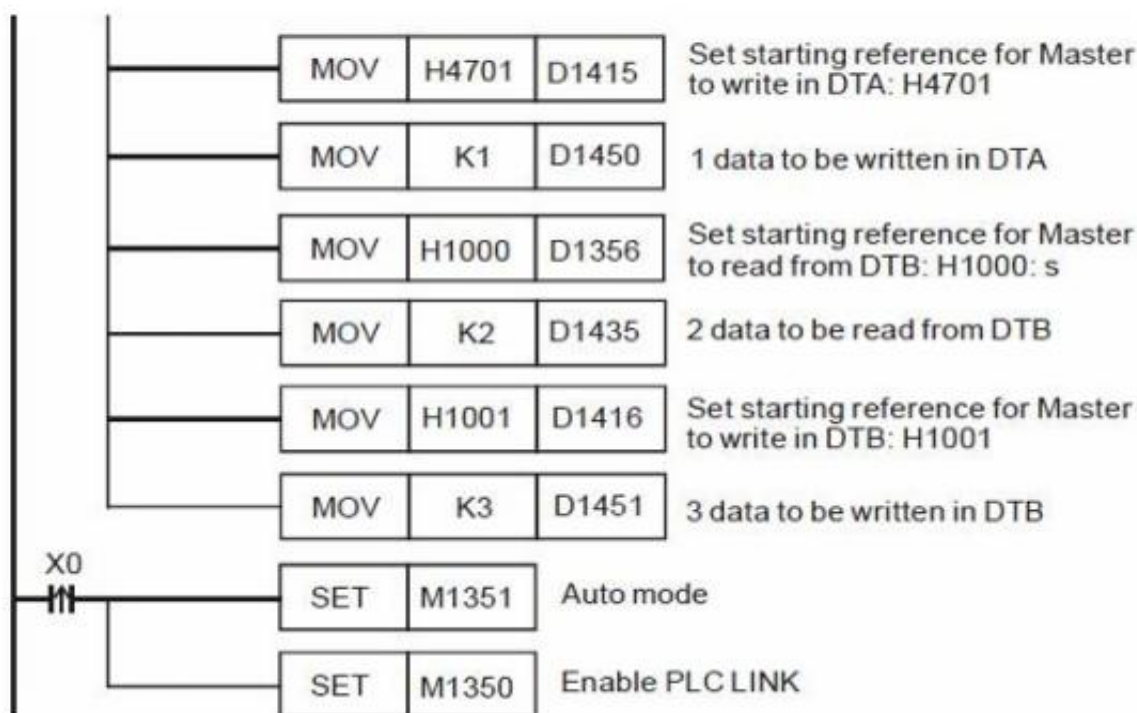
- شماره آدرس ارتباطات قابل دسترس برای کنترلر از 1 الی 255 می‌باشد و اگر آدرس ارتباطات برابر با 0 تنظیم شود، ارتباطات از نوع Broadcast خواهد شد.

ابزارهای مورد نیاز برای برنامه‌نویسی PLC در این مثال به شرح زیر است:

ابزارها	توابع
X0	از این ورودی برای راه‌اندازی PLC لینک استفاده می‌شود.
M1350	از این فلگ خاص برای فعال کردن PLC لینک استفاده می‌شود.
M1351	از این فلگ خاص برای فعال کردن مد اتوماتیک PLC لینک استفاده می‌شود.
M1352	از این فلگ خاص برای فعال کردن مد دستی PLC لینک استفاده می‌شود.
M1353	از این فلگ خاص برای فعال کردن 32 Slave به هم پیوسته در شبکه و تبادل داده با طول داده تا حدود 100 کلمه استفاده می‌شود.
M1354	از این فلگ خاص برای فعال کردن همزمان خواندن و نوشتن داده برای نمونه‌برداری در لینک شبکه PLC استفاده می‌شود.

برنامه کنترلی نوشته شده برای PLC در این مثال مطابق با شکل زیر است:





عملکرد برنامه‌ی بالا به شرح زیر است:

رجیسترهای D1480~D1481 در PLC برابر با پارامترهای H4700~H4701 در کنترلر فرآیند سری DTA دلتا می‌باشد. وقتی که ورودی X0=ON شود، شبکه PLC لینک فعال شده و مقدار H4700~H4701 (دمای هدف (نقطه مطلوب SP) و دمای حاضر (فرآیند PV)) در رجیسترهای D1480~D1481 نشان داده خواهد شد.

رجیستر D1496 در PLC برابر با پارامتر ارتباطات H4701 در کنترلر فرآیند سری DTA دلتا می‌باشد. وقتی که ورودی X0=ON شود، شبکه PLC لینک فعال شده و مقدار H4701 توسط رجیستر D1496 تعیین می‌شود.

وضعیت‌های کنترلر فرآیند DTA می‌تواند توسط تنظیمات مقدار D1496 کنترل شود. (D1496=K300، دمای هدف 30°C خواهد بود.)

رجیسترهای D1512~D1513 در PLC برابر با پارامترهای H1000~H1001 در کنترلر فرآیند سری DTB دلتا می‌باشد. وقتی که ورودی X0=ON شود، شبکه PLC لینک فعال شده و مقدار H1000~H1001 (دمای هدف (نقطه مطلوب SP) و دما حاضر (فرآیند PV)) در رجیسترهای D1512~D1513 نشان داده خواهد شد.

رجیسترهای D1528~D1530 در PLC برابر با پارامترهای ارتباطات H1001~H1003 در کنترلر فرآیند سری DTB دلتا می‌باشد. وقتی که ورودی $X0=ON$ شود، شبکه PLC لینک فعال شده و مقادیر H1001~H1003 توسط رجیسترهای D1528~D1530 تعیین می‌شود. دمای هدف کنترلر فرآیند DTB می‌تواند توسط تنظیمات مقدار D1528 تعیین شود. ($D1528=K400$ ، دمای هدف $40^{\circ}C$ خواهد بود)

محدوده‌ی بالا و پایین رنج دما در کنترلر DTB می‌تواند توسط مقادیر رجیسترهای D1529~D1530 تعیین شود. ($D1529=K500$ و $D1530=K10$ ، محدود بالا برابر با $50^{\circ}C$ و محدوده پایین برابر با $1^{\circ}C$ خواهد بود)

شماره ایستگاه Slave ها باید منحصر بفرد و متفاوت با شماره ایستگاه Master باشند. PLC های سری SA/ SX/ SC/ SV/ EH/ EH2 می‌توانند به عنوان Master باشند. PLC های سری ES/ EX/ SS نمی‌توانند به عنوان Master باشند.

وقتی که ورودی $X0=ON$ شود، شبکه PLC لینک فعال خواهد شد. اگر در عمل فعال شدن خطایی رخ داد، بیت‌های M1350/M1351 غیر فعال OFF خواهند شد و لازم است که برای راه‌اندازی شبکه بیت $X0$ ابتدا غیرفعال شده و مجدداً فعال شود.

7-24 برقراری ارتباط بین دو PLC دلتا با دستورالعمل RS

در این مثال، می‌خواهیم Start/Stop دو عدد DVP PLC تحت شبکه را با استفاده از دستورالعمل RS کنترل نماییم.



هدف از کنترل در این مثال به شرح زیر است :

می‌خواهیم وضعیت START/STOP دو Slave PLC تحت شبکه را توسط Master PLC کنترل نماییم.

پارامترهای تنظیمی برای PLC جهت راه‌اندازی شبکه مطابق با جدول زیر است :

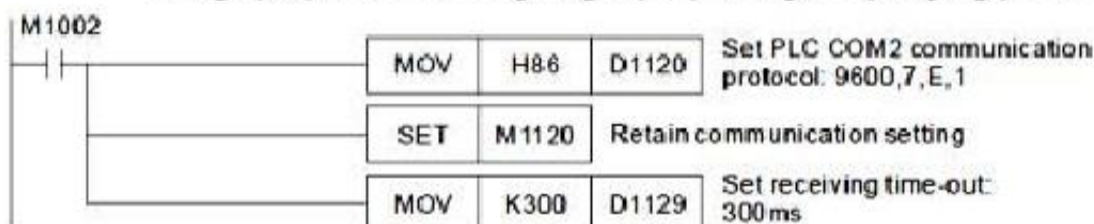
فرمت ارتباطات	شماره ایستگاه	Master/Slave
برای تنظیم فرمت ارتباطات برابر با ASCII, 9600, 7, E, 1 باید D1120=H86 شود. فرمت ارتباطات در تمام تجهیزات متصل شده به شبکه باید با یکدیگر یکسان باشند.	K10(D1121=K10)	Master PLC
	K1(D1121=K1)	Slave 1
	K2(D1121=K2)	Slave 2

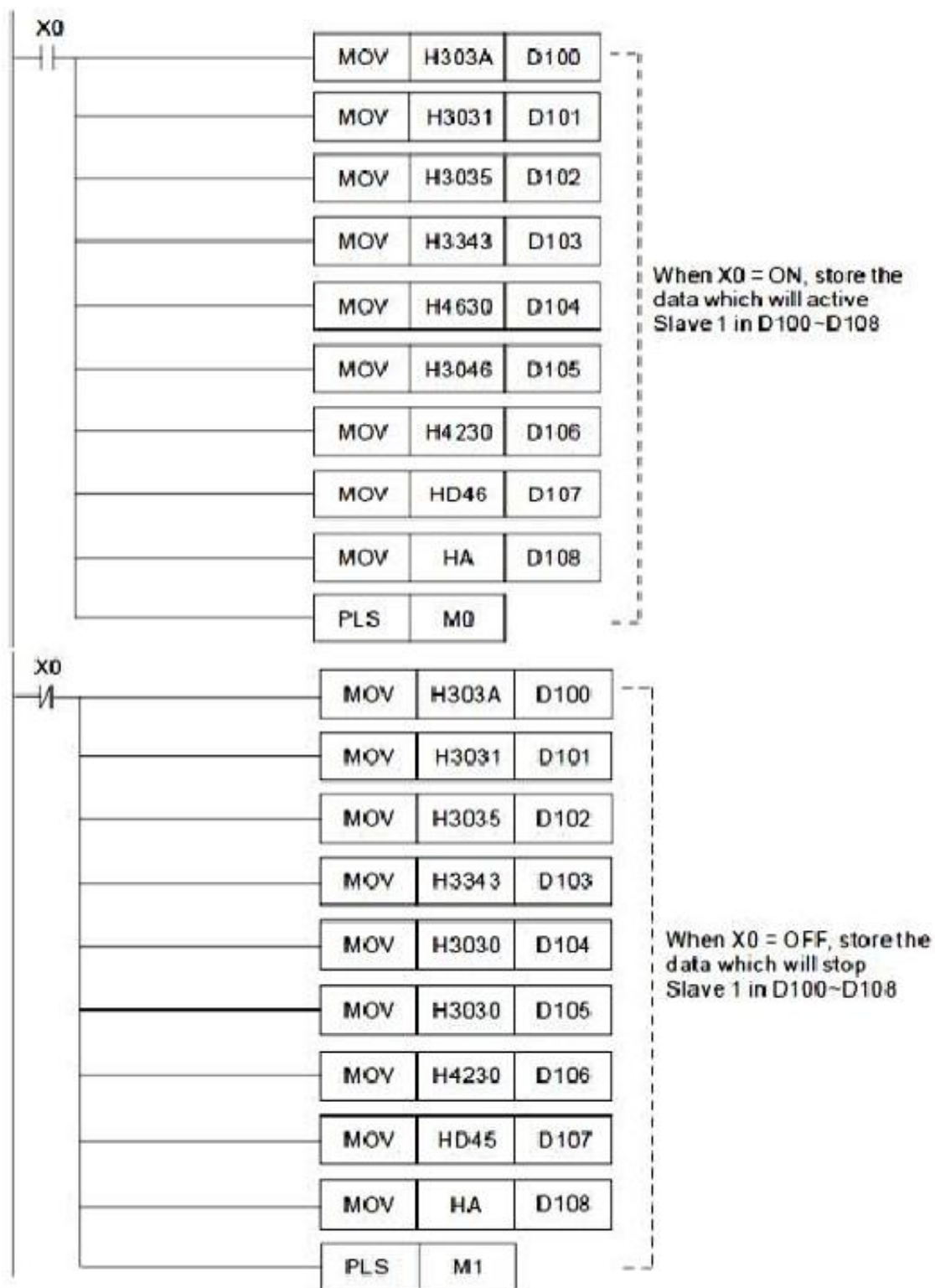
اگر بعد از تنظیم پارامترهای ارتباطات شبکه در تمام PLC ها، PLC ها به صورت نرمال راه‌اندازی نشدند، ابتدا باید تنظیمات PLC ها را به تنظیمات پیش‌فرض کارخانه بازگردانید سپس تنظیمات گفته شده در جدول بالا را مجدداً برای تمام PLC ها اجرا کنید. برای فرمت کردن تنظیمات PLC باید از نرم‌افزار WPLSoft استفاده نموده و از منوی (C) Communication گزینه Format PLC memory را انتخاب نمایید.

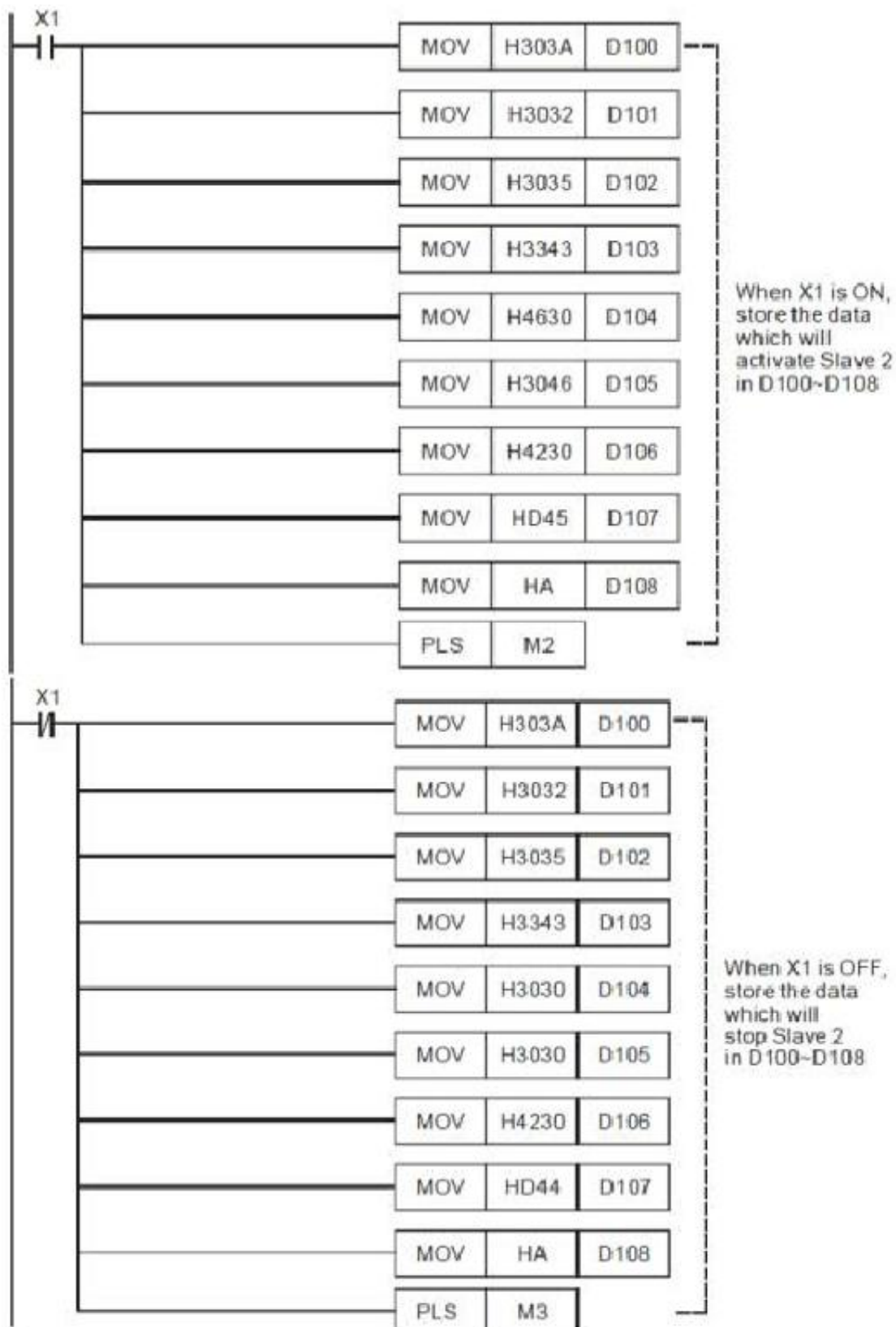
ابزارهای مورد استفاده در این برنامه مطابق با جدول زیر است :

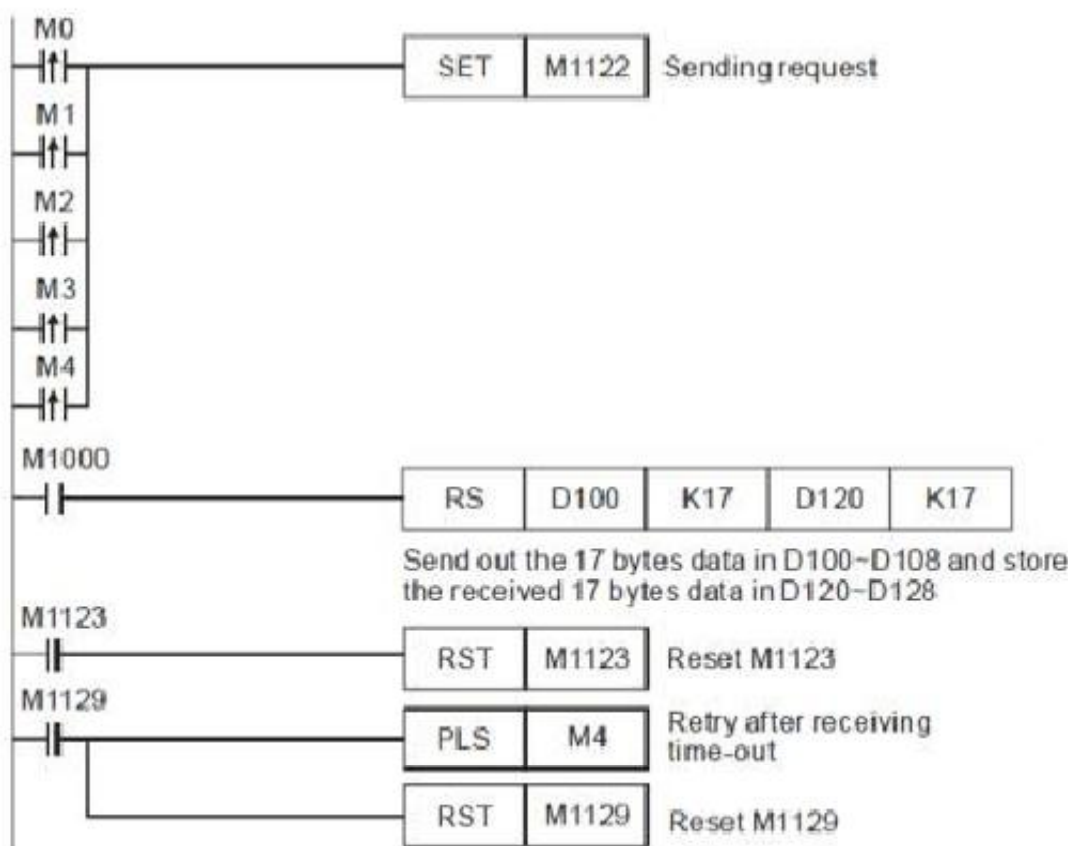
ابزار	توضیحات
X0	از این ورودی برای Start/Stop Slave 1 استفاده می‌شود.
X1	از این ورودی برای Start/Stop Slave 2 استفاده می‌شود.
M0	از این فلگ عمومی برای اجرا نخستین دستورالعمل RS استفاده می‌شود.
M1	از این فلگ عمومی برای اجرا دومین دستورالعمل RS استفاده می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر می‌باشد :









توضیحات برنامه‌ی نوشته شده برای این مثال به شرح زیر است:

ابتدا باید پورت ارتباطات PLC RS-485 را مقداردهی کنیم. فرمت ارتباطات باید برابر با MODBUS ASCII, 9600, 7, E, 1 تنظیم شود. دقت کنید که فرمت ارتباطات RS-485 باید در هر دو سمت Master PLC و Slave ها به صورت یکسان تنظیم شود. دو فلگ خاص برای دستورالعمل RS وجود دارد: فلگ خاص M1123 جهت نرمال بودن ارتباطات رزرو شده و فلگ خاص M1129 برای خطای وقفه افتادن در ارسال و دریافت دیتا در شبکه رزرو شده است. وقتی که خطای وقفه‌ی ارتباطات (Communication time-out) رخ دهد، می‌توان از فلگ عمومی M4 برای برقراری ارتباط مجدد استفاده کرد.

وقتی که ورودی X0=ON شود، ایستگاه Slave 1 راه‌اندازی (Start Running) می‌شود. وقتی که ورودی X0=OFF شود، ایستگاه Slave 1 متوقف (Stop) خواهد شد. وقتی که ورودی X1=ON شود، ایستگاه Slave 2 راه‌اندازی می‌شود، وقتی که ورودی X1=OFF شود، ایستگاه Slave 2 استوپ می‌شود.

7- 25 برقراری ارتباط بین PLC دلتا و اینورتر میکرومستر 420 زیمنس توسط دستورالعمل RS

هدف از کنترل در این مثال به شرح زیر است :

می‌خواهیم یک ارتباط شبکه بین Master PLC که یک PLC دلتا است با Slave device که یک اینورتر میکرومستر 420 زیمنس است، برقرار کرده و از طریق شبکه‌ی MODBUS توسط PLC دلتا با دستورالعمل RS، اینورتر زیمنس را کنترل کرده و Start/Stop نماییم. تنظیمات پارامترهای شبکه برای اینورتر میکرومستر 420 زیمنس مطابق با جدول زیر است :

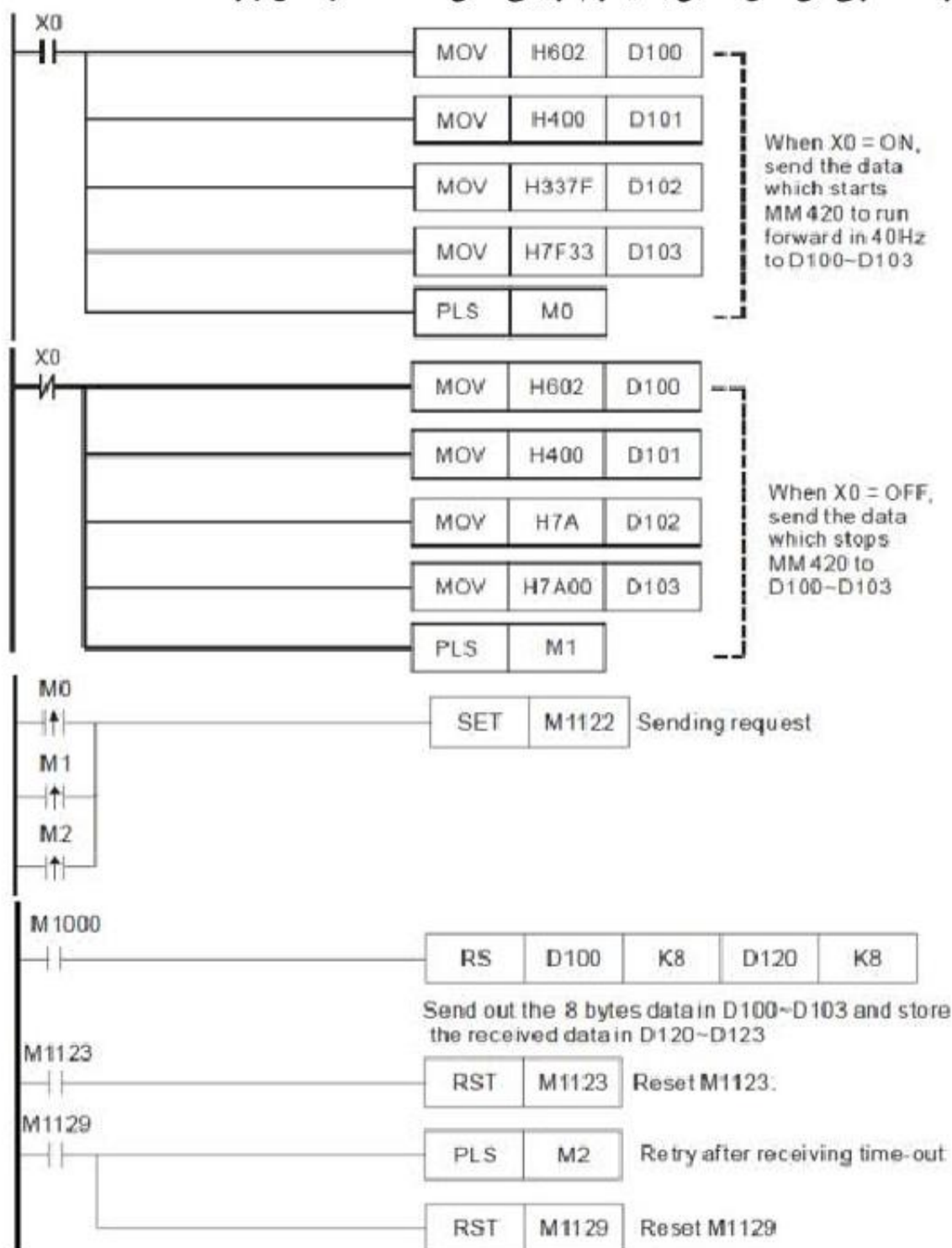
پارامتر	مقدار تنظیمی	توضیحات
P0003	3	توسط این پارامتر سطح پارامترهای قابل دسترس برای کاربر را برابر با expert تنظیم نمایید تا پارامترهای مربوط به شبکه قابل مشاهده شوند.
P0700	5	منبع فرمان (Start/Stop) برابر با USS on COM Link (RS-485) انتخاب شود.
P1000	5	منبع فرکانس (تغییر فرکانس خروجی) برابر با USS on COM Link (RS-485) انتخاب شود.
P2010	6	سرعت تبادل داده در شبکه باید برابر با 9600ps تنظیم شود.
P2011	0	آدرس گره‌ی اینورتر برای قرار گرفتن در شبکه باید برابر با 0 تنظیم شود.

اگر بعد از تنظیمات پارامترهای اینورتر میکرومستر MM420، اینورتر به صورت نرمال در شبکه راه‌اندازی نشد، ابتدا باید تنظیمات پارامترهای اینورتر را به تنظیمات پیش‌فرض کارخانه بازگردانده و سپس مجدداً تنظیمات پارامترهای اینورتر را مطابق با جدول بالا تنظیم نمایید. برای فرمت کردن تنظیمات اینورترهای میکرومستر باید پارامتر $P0010=30$ تنظیم کرده و سپس پارامتر $P0970=1$ تنظیم نمایید.

ابزارهای مورد استفاده در برنامه‌نویسی PLC دلتا در این مثال مطابق با جدول زیر است :

ابزار	توضیحات
X0	به این ورودی سخت افزاری PLC دلتا سوئیچ Start/Stop متصل می‌شود.

برنامه کنترلی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر است:



PLC → MM420, PLC Sends: 02 06 00 047F 3333 7F
MM420 → PLC, PLC receives: 02 06 00 FB34 3333 CB

Register	DATA	Explanation
D100 low	02H	Head. Fixed as 02H. (start of the message)
D100 high	06H	The number of the following bytes
D101 low	00H	Station No. (range: 0~31, corresponding to hex 00H~1FH)
D101 high	04H	Control Word
D102 low	7FH	(starting MM420. Refer to Remarks for definitions.) بخش توجهات را مطالعه نمایید
D102 high	33H	Frequency (4000H = base frequency 50Hz, 3333H = 40Hz)
D103 low	33H	
D103 high	7FH	Tail. (XOR result of all the bytes before this byte)

Register	DATA	Explanation
D120 low	02H	Head. Fixed as 02H (start of the message)
D120 high	06H	The number of the following bytes
D121 low	00H	Station No. (range: 0~31, corresponding to hex 00H~1FH)
D121 high	FBH	Control Word (Refer to Remarks for definitions)
D122 low	34H	بخش توجهات را مطالعه نمایید.
D122 high	33H	Frequency (4000H = base frequency 50Hz, 3333H = 40Hz)
D123 low	33H	
D123 high	CBH	Tail. (XOR result of all the bytes before this byte)

وقتی که ورودی $X0=OFF$ شود، اینورتر میکرومستر MM420 استوپ خواهد شد.

PLC → MM420, PLC Sends 02 06 00 047A 0000 7A
MM420 → PLC, PLC receives: 02 06 00 FB11 0000 EE

رجیسترهای ارسال دیتا (ارسال پیام خروجی از PLC به سمت اینورتر میکرومستر MM420 جهت استوپ کردن اینورتر) مطابق با جدول زیر می‌باشد:

Register	DATA	Explanation
D100 low	02H	Head. Fixed as 02H. (start of the message)
D100 high	06H	The number of the following bytes
D101 low	00H	Station No. (range: 0~31, corresponding to hex 00H~1FH)
D101 high	04H	Control Word (Refer to Remarks for definitions)
D102 low	7AH	بخش توجهات را مطالعه نمایید
D102 high	00H	Frequency (0000H = 0 Hz.)
D103 low	00H	
D103 high	7AH	Tail. (XOR result of all the bytes before this byte)

رجیسترهای دریافت دیتا (پاسخ اینورتر میکرومستر MM420 به پیام ارسالی PLC جهت توقف اینورتر) به شرح زیر می‌باشد:

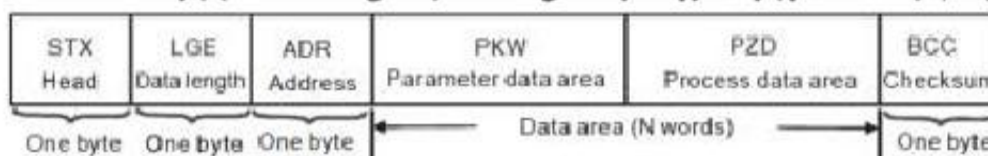
Register	DATA	Explanation
D120 low	02H	Head. Fixed as 02H (start of the message)
D120 high	06H	The number of the following bytes
D121 low	00H	Station No. (range: 0~31, corresponding to hex 00H~1FH)
D121 high	FBH	Control Word (Refer to Remarks for definitions)
D122 low	11H	بخش توجهات را مطالعه نمایید
D122 high	00H	Frequency (0000H = 0 Hz.)
D123 low	00H	
D123 high	EEH	Tail. (XOR result of all the bytes before this byte)

دقت کنید که دو فلگ خاص برای ارتباطات RS وجود دارد. از فلگ خاص M1123 برای وضعیت نرمال بودن ارتباطات شبکه استفاده شده و از فلگ خاص M1129 برای وضعیت خطای وقفه (Time-out) در تبادل داده‌ی تحت شبکه استفاده شده است. وقتی که خطای وقفه‌ی ارتباطات شبکه (Communication time-out) رخ دهد، با استفاده از فلگ عمومی M2 می‌توان خطا را ریست کرده و وضعیت ارتباطات شبکه و تبادل دیتای شبکه را به وضعیت نرمال بازگرداند.

توجه :

در اینورتر میکرومستر MM420 زیمنس از پروتکل ارتباطات USS استفاده شده است. در این پروتکل ارتباطات می‌توان از یک Master و سی و یک Slave استفاده کرد. بنابراین به یک Master PLC می‌توان 31 عدد Slave متصل کرد. شماره‌ی (آدرس) گره Slave ها می‌تواند از 0 الی 31 باشد (Slave ID: 0~31).

ساختار پیغام ارتباطات اینورتر میکرومستر مطابق با ساختار نشان داده شده در زیر است :



برای نواحی STX, LGE, ADR, BCC طول داده برابر با یک بایت می‌باشد. مقدار ناحیه STX یک عدد ثابت به فرمت هگز بوده و برابر با 02H می‌باشد. این به معنی شروع پیغام است.

مقدار ناحیه LGG باید بین مقدار ناحیه ADR و ناحیه BCC بوده و یک عدد از نوع بایت می‌باشد.

ناحیه ADR همان آدرس ارتباطات USS است. رنج این عدد باید بین 0~31 (به فرمت دسیمال) باشد اعداد بین 0~31 دسیمال برابر با اعداد بین 00H~1FH هگز است. بنابراین، همانطور که متوجه شدید، در اینجا، ما از فرمت هگز در برنامه‌نویسی استفاده می‌کنیم.

ناحیه‌ی داده‌ی اینورتر میکرومستر MM420 در داخل دو ناحیه‌ی PKW و ناحیه‌ی PZD تقسیم شده است. از ناحیه‌ی PKW برای خواندن و نوشتن مقادیر پارامترهای اینورتر استفاده می‌شود. ناحیه PKW شامل 4 کلمه (0~4 Word) می‌باشد. (برای اطلاع از این کلمه به تنظیمات پارامتر P2013 اینورتر مراجعه نمایید). از ناحیه PZD برای کنترل اینورتر و تنظیمات فرکانس خروجی اینورتر استفاده می‌شود. ناحیه PZD نیز شامل 4 کلمه (0~4 Word) می‌باشد. (برای اطلاع از دو کلمه‌ی کنترلی PZD به تنظیمات پارامتر P2012 اینورتر میکرومستر MM420 زیمنس مراجعه نمایید). به کلمات کنترلی ناحیه PZD اینورتر Control Word گفته می‌شود. نخستین کلمه PZD همان Control Word کنترل اینورتر (شامل استارت، استوپ، ریست و ...) می‌باشد و دومین کلمه‌ی PZD همان Control Word تنظیم فرکانس خروجی اینورتر میکرومستر توسط شبکه‌ی ارتباطات RS-485 است که این فرکانس از طریق PLC دلتا که در این مثال Master PLC می‌باشد، تنظیم خواهد شد.

از هر یک از نواحی PZD و RKW می‌توان به صورت جداگانه استفاده کرد و همچنین، می‌توان از این دو ناحیه در کنار هم نیز استفاده کرد. معمولاً فقط از ناحیه PZD برای کنترل کردن وضعیت اینورتر و تنظیم فرکانس خروجی اینورتر استفاده می‌شود. دقت داشته باشید که ناحیه PZD همان Control Word می‌باشد که می‌توان از طریق آن اینورتر میکرومستر زیمنس را کنترل کرد. در این برنامه، از ناحیه PZD با طول داده‌ی برابر با 2 کلمه (2 Word) از 4 کلمه استفاده شده است. ساختار ناحیه PZD در این مثال مطابق با شکل زیر است:

02	06	00	047F 3333	7F
STX	LGE	ADR	← DATA(PZD) →	BCC

در شکل بالا، 047FH یک Control Word برای راه‌اندازی اینورتر می‌باشد و 3333H فرکانس است. از آنجایی که H4000 برابر با فرکانس 50HZ می‌باشد، 3333H برابر با فرکانس 40HZ خواهد بود.

BCC Checksum: این قابلیت بایت‌های یک Control Word را از STX تا PZD با یکدیگر XOR کرده و نتیجه را در خود ذخیره می‌کند. برای مثال: $02 \text{ XOR } 06 \text{ XOR } 00 = 04$ XOR 7F XOR 33 XOR 33 = H7F Control Word اینورترهای میکرومستر MM420 زیمنس در ناحیه PZD به شرح زیر می‌باشد و منظور همان ارسال پیغام از اینورتر به PLC است:

Bit	Explanation	Bit status	
00	ON (ramp up enabled)/OFF1 (ramp down disabled)	0 No (OFF1)	1 Yes (ON)
01	OFF2: Coast to standstill	0 Yes	1 No
02	OFF3: Quick ramp down	0 Yes	1 No
03	Pulses enabled	0 No	1 Yes
04	RFG (ramp function generator) enabled	0 No	1 Yes
05	RFG (ramp function generator) start	0 No	1 Yes
06	Set value of frequency enabled	0 No	1 Yes
07	Fault acknowledge	0 No	1 Yes
08	JOG right	0 No	1 Yes
09	JOG left	0 No	1 Yes
10	Controlled by PLC	0 No	1 Yes
11	Reverse	0 No	1 Yes
12	Reserved	—	—
13	MOP (motor potentiometer) up	0 No	1 Yes
14	MOP (motor potentiometer) down	0 No	1 Yes
15	Local/remote control	0 No	1 Yes

توجه داشته باشید که در میان بیت‌های Control Word ارسال شده از PLC به اینورتر میکرومستر، بیت 10 برابر با 1 تنظیم شود، اگر بیت 10 برابر با 0 تنظیم شود Control Word ارسال شده به اینورتر معتبر نمی‌باشد.

Status Word (اطلاعات وضعیت اینورتر) اینورتر میکرومستر MM420 زیمنس در ناحیه PZD تعریف می‌شود، (منظور همان پاسخ اینورتر به پیام ارسال شده از سمت PLC می‌باشد).

Bit	Explanation	Bit status	
00	Drive ready	0 No (OFF1)	1 Yes (ON)
01	Drive ready to run	0 No	1 Yes
02	Drive running	0 No	1 Yes
03	Drive fault active	0 No	1 Yes
04	OFF2 active	0 Yes	1 No
05	OFF3 enabled	0 No	1 Yes
06	Switch on inhibit active	0 No	1 Yes
07	Drive warning active	0 No	1 Yes
08	Excessive deviation	0 Yes	1 No
09	PZDI (process data) control	0 No	1 Yes
10	Maximum frequency reached	0 No	1 Yes
11	Over current alarm	0 Yes	1 No
12	Motor holding brake enabled	0 Yes	1 No
13	Motor overload	0 Yes	1 No
14	Motor running forward	0 No	1 Yes
15	Inverter overload	0 Yes	1 No

7- 26 برقراری ارتباط بین PLC دلتا و اینورتر دانفوس VLT 6000 توسط دستورالعمل RS

در این مثال، می‌خواهیم یک ارتباط شبکه بین PLC دلتا و اینورتر VLT6000 دانفوس سری Adjustable با استفاده از دستورالعمل RS برقرار نماییم.

هدف از کنترل در این مثال به شرح زیر است:

در این مثال، قصد داریم وضعیت Start/Stop اینورتر را توسط PLC کنترل نموده و همچنین، فرکانس خروجی اینورتر را توسط Master PLC خوانده و کنترل نماییم.
پارامترهای قابل تنظیم برای اینورتر سری VLT6000 دانفوس برای ارتباطات شبکه‌ی سریال به شرح زیر است:

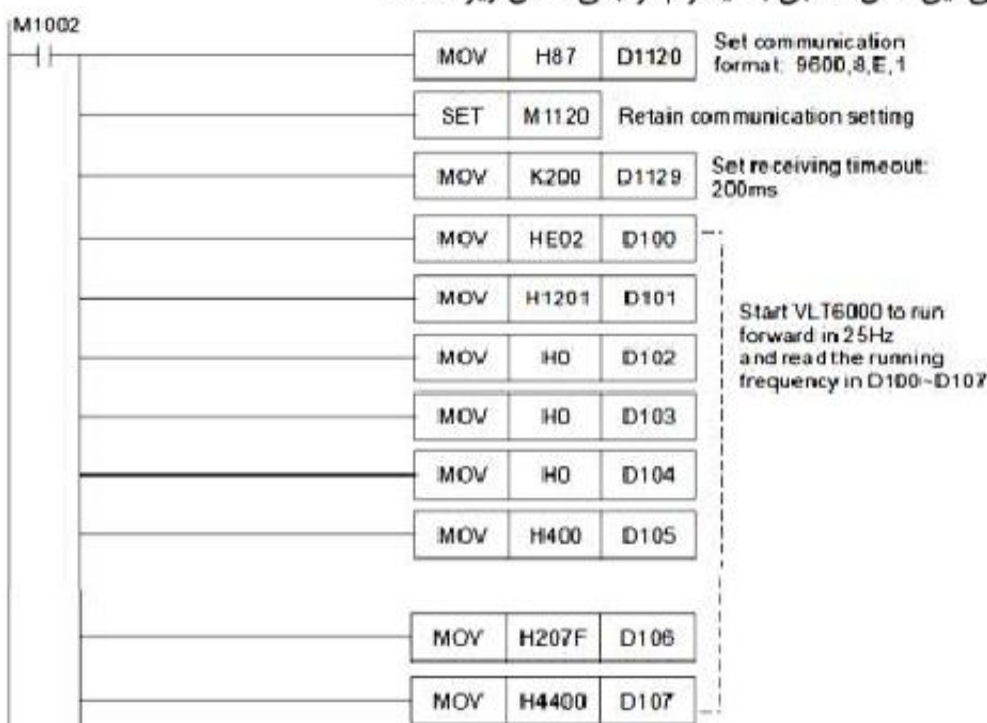
پارامتر	مقدار تنظیمی	توضیحات
P500	0	پروتکل ارتباطات سریال را باید با FC Protocol تنظیم نمایید.
P501	1	آدرس ارتباطات شبکه FC را باید برابر با 1 تنظیم نمایید.
P502	5	سرعت ارتباطات شبکه FC برای تبادل دیتا در شبکه را برابر با 9600 bps تنظیم نمایید.
P503	1	کنترل شیب توقف (Coasting Stop) موتور توسط ارتباطات شبکه‌ی سریال صورت خواهد گرفت.
P504	1	با تنظیم این پارامتر، ترمز DC موتور را می‌توان توسط ارتباطات شبکه‌ی سریال کنترل کرد.
P505	1	با تنظیم این پارامتر، راه‌اندازی (Start) اینورتر را می‌توان توسط ارتباطات شبکه‌ی سریال کنترل کرد.

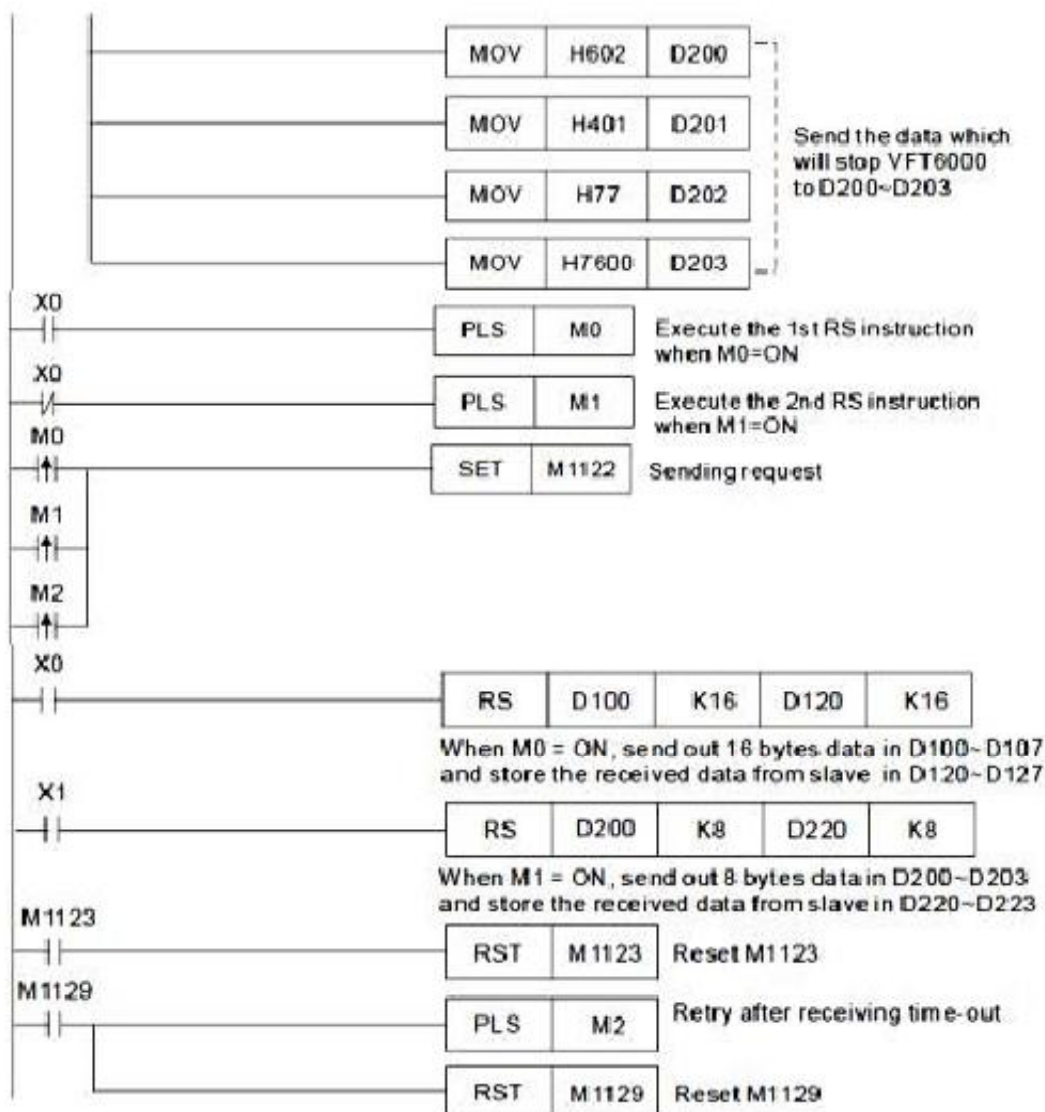
اگر بعد از تنظیمات پارامترهای اینورتر سری VLT6000 دانفوس، اینورتر به درستی راه‌اندازی نشد، ابتدا باید تنظیمات پارامترهای اینورتر را به تنظیمات پیش‌فرض کارخانه بازگردانده و سپس تنظیمات پارامترهای اینورتر را مطابق با تنظیمات گفته شده در جدول بالا انجام دهید. برای بازگرداندن تنظیمات اینورتر به تنظیمات پیش‌فرض کارخانه، باید پارامتر $P620 = 3$ تنظیم شود. سپس باید دکمه OK اینورتر را فشار دهید.

ابزارهای مورد استفاده برای PLC در این مثال مطابق با جدول زیر است:

ابزار	توضیحات
X0	به این ورودی PLC شستی Start/Stop اینورتر متصل می‌شود.
M0	از این فلگ عمومی برای اجرای نخستین دستورالعمل RS استفاده می‌شود.
M1	از این فلگ عمومی برای اجرای دومین دستورالعمل RS استفاده می‌شود.

برنامه‌ی کنترلی این مثال مطابق با دیاگرام نردبانی شکل زیر است:





عملکرد برنامه‌ی نوشته شده برای این مثال به شرح زیر می‌باشد:

در ابتدا تنظیمات پورت ارتباطات RS-485 را برای PLC برابر با فرمت 9600, 8, E, 1 تنظیم کرده‌ایم. لازم به ذکر است که باید تنظیمات ارتباطات RS-485 اینورتر سری VLT6000 دانفوس با تنظیمات فرمت ارتباطات Master PLC یکسان باشد.

وقتی که ورودی X0=ON شود، اینورتر استارت شده و الکتروموتور متصل شده به اینورتر در جهت راستگرد و با فرکانس 40Hz شروع به چرخش می‌کند. در این حالت، فرکانس خروجی به صورت دوطرفه توسط PLC و اینورتر خوانده خواهد شد.

02 0E 01 1200 0000 00000000 مقدار PLC در این حالت، PLC → VLT6000
047F 2000 44 را به اینورتر ارسال خواهد کرد.

02 0E 01 1200 0000 000000FA VLT6000 → PLC در این حالت، اینورتر مقدار
0F07 1FFF 0D را به PLC ارسال کرده و PLC مقدار را جهت خواندن وضعیت اینورتر دریافت
می‌کند.

رجیسترهای ارسال داده (ارسال پیغام‌های خروجی توسط PLC) در سمت PLC جهت راه‌اندازی
اینورتر به شرح زیر می‌باشد:

Register	Data	Explanation	
D100 low	02H	Head, fixed as 02H (start of the message)	
D100 high	0EH	The number of the following bytes	
D101 low	01H	Station No. (range: 0~31, corresponding to hex 00H~1FH)	
D101 high	12H	PKW area	PKE 1H: function code for reading parameters 200H: parameter No. P512 (output frequency)
D102 low	00H		IND Index area (used in indexed parameters, such as P615. Not used in this program.)
D102 high	00H		
D103 low	00H		PWE _{high} Value: 1 (In read status: 0. In write status: high Word will be read)
D103 high	00H		
D104 low	00H		PWE _{low} Value: 2 (In read status: 0. In write status: low Word will be read)
D104 high	00H		
D105 low	00H	PCD1 area	Control Word (starting VLT6000. For the definition, please refer to Remarks.)
D105 high	04H		
D106 low	7FH	PCD2 area	Frequency (4000H corresponds to base frequency 50Hz and 2000H corresponds to 25Hz)
D106 high	20H		
D107 low	00H	BCC area	Tail. (XOR result of all the bytes before this byte)
D107 high	44H		

رجیسترهای دریافت داده (پاسخ اینورتر به پیغام‌های PLC) در سمت PLC جهت فعال بودن
Run (بودن) اینورتر به شرح زیر می‌باشد:

Register	Data	Explanation		
D120 low	02H	Head, fixed as 02H (start of the message)		
D120 high	0EH	The number of the following bytes		
D121 low	01H	Station No. (range: 0~31, corresponding to hex 00H~1FH)		
D121 high	12H	PKW area	PKE	1H: function code for reading parameters 200H: parameter No. P512 (output frequency)
D122 low	00H		IND	Index area (used in indexed parameters, such as P615. Not used in this program.)
D122 high	00H			
D123 low	00H		PWE _{high}	High Word will be read
D123 high	00H			
D124 low	00H		PWE _{low}	Low Word will be read (00FAH corresponds to the decimal value 250 which means the frequency of 25Hz.
D124 high	00H			
D125 low	FAH		PCD1 area	Status Word (For the definition, please refer to Remarks.)
D125 high	0FH			
D126 low	07H			

D126 high	1FH	PCD2 area	Frequency (4000H corresponds to the base frequency 50Hz and 1FFFHz corresponds to 25Hz)
D127 low	FFH		
D127 high	0DH	BCC area	Tail. (XOR result of all the bytes before this byte)

وقتی که ورودی $X0=OFF$ شود، اینورتر استوپ خواهد شد. در این پیام فقط ناحیه PCD به کار برده می‌شود.

ارسال خواهد کرد. $PLC \rightarrow VLT6000$ ، در این حالت PLC مقدار 02 06 01 0477 0000 76 را به اینورتر

ارسال خواهد کرد. $VLT6000 \rightarrow PLC$ ، در این حالت اینورتر مقدار 02 06 01 0603 0000 00 را به PLC

مقادیر داخل رجیسترهای ارسال داده (ارسال پیام استوپ شدن اینورتر توسط PLC) در سمت PLC برای استوپ کردن اینورتر به شرح زیر است:

Register	DATA	Explanation
D200 low	02H	Head. Fixed as 02H. (start of the message)
D200 high	06H	The number of the following bytes
D201 low	01H	Station No. (range: 0~31, corresponding to hex 00H~1FH)
D201 high	04H	Control byte (starting AC motor drive. For the definition, please refer to Remarks.) بخش توجهات را مطالعه نمایید
D202 low	77H	
D202 high	00H	Frequency (0000H corresponding to 0Hz)
D203 low	00H	
D203 high	76H	Tail. (XOR result of all the bytes before this byte)

مقادیر داخل رجیسترهای دریافت داده (پاسخ اینورتر به پیام استوپ شدن اینورتر) در سمت PLC به شرح زیر است:

Register	DATA	Explanation
D220 low	02H	Head. Fixed as 02H (start of the message)
D220 high	06H	The number of the following bytes
D221 low	01H	Station No. (range: 0~31, corresponding to hex 00H~1FH)
D221 high	06H	Control Word (starting AC motor drive. For the definition, please refer to Remarks.) بخش توجهات را مطالعه نمایید
D222 low	03H	
D222 high	00H	Frequency (0000H corresponding to 0Hz)
D223 low	00H	
D223 high	00H	Tail. (XOR result of all the bytes before this byte)

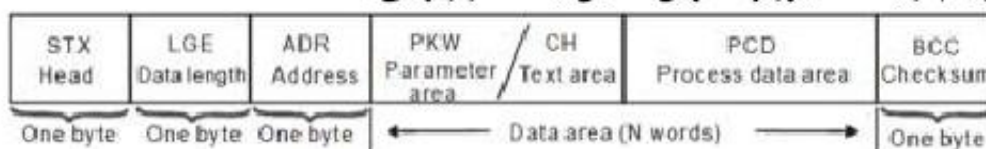
در این برنامه دو وضعیت برای ارتباطات RS وجود دارد. از فلگ خاص M1123 برای وضعیت نرمال بودن ارتباطات شبکه استفاده شده و از فلگ خاص M1129 برای آشکارسازی وقفه رخ داده در ارسال و دریافت پیام توسط شبکه (time-out) استفاده شده است. وقتی که یک وقفه در ارتباطات

شبکه رخ دهد، این وقفه را می‌توان توسط ابزار M2 ریست کرده و مجدداً تبادل دیتای شبکه را راه‌اندازی کرد.

توجهات

سه پروتکل ارتباطات شبکه‌ای برای اینورتر سری VLT6000 دانفوس وجود دارد که عبارتند از: پروتکل FC، پروتکل ارتباطات Metasys NZ و پروتکل ارتباطات LS FLN. در این مثال، از پروتکل ارتباطات FC استفاده شده است. این پروتکل ارتباطات شبکه مانند پروتکل ارتباطات USS اینورترهای میکرومستر زیمنس می‌باشد. در این پروتکل ارتباطات مجاز به استفاده از 32 دستگاه تحت شبکه می‌باشیم که 1 دستگاه بعنوان Master و 32 دستگاه دیگر بعنوان Slave در شبکه متصل می‌شوند. شماره‌ی گره‌ی تنظیم شده برای Slave ها باید از 0 الی 31 باشد. شماره گره همان Slave ID است.

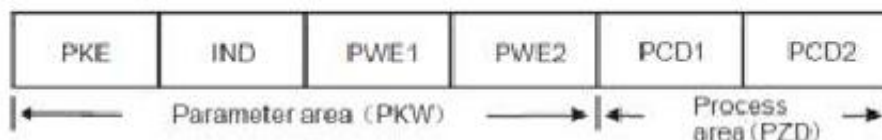
ساختار پیغام ارتباطات اینورتر دانفوس مطابق با ساختار زیر می‌باشد:



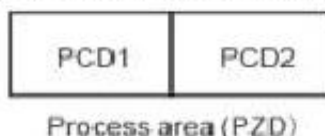
نواحی تعریف شده‌ی STX, LGE, ADR, BCC برای پروتکل FC مشابه نواحی تعریف شده‌ی پروتکل USS اینورترهای میکرومستر زیمنس می‌باشند. برای اطلاعات بیشتر درمورد نواحی داده‌ی گفته شده به مثال‌های قبل مراجعه نمایید.

در ناحیه‌ی داده‌ی اینورتر دانفوس با پروتکل FC می‌توان از 3 نوع پیغام استفاده کرد.

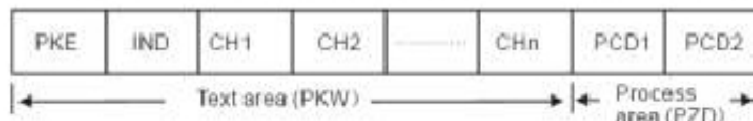
1. پیغام اینورتر شامل ناحیه‌ی پارامتر و ناحیه‌ی فرآیند می‌باشد. از این نوع پیغام برای ارسال پارامترها به سیستم Master-Slave استفاده می‌شود. این نوع پیغام از 6 کلمه (6 Words) تشکیل شده است.



2. پیغام نوع دوم اینورتر فقط شامل ناحیه فرآیند است. این پیغام شامل Control Word (Status Word) و مقدار فرکانس می‌باشد. این پیغام از دو کلمه (2 Words) تشکیل شده است.



3. پیام نوع سوم اینورتر از ناحیه متن (Text area) برای خواندن و نوشتن متن برای ناحیه داده تشکیل شده است. از این ناحیه برای خواندن و نوشتن اطلاعات در پارامتر P621, P631 استفاده می‌شود.



Control Word تعریف شده برای اینورتر در ناحیه‌ی PZD مطابق با جدول زیر می‌باشد:

Bit	Bit status	
	Bit status = 0	Bit status = 1
00	-	Preset ref. lsb
01	-	Preset ref. msb
02	DC braking	-
03	Coasting stop	-
04	Quick stop	-
05	Freeze output frequency	-
06	Ramp stop	Start
07	-	Reset
08	-	JOG
09	No function	
10	Data not invalid	Data valid
11	-	Activate relay 01
12	-	Activate relay 02
13	-	Choice of setup lsb
14	-	Choice of setup msb
15	-	Reversing

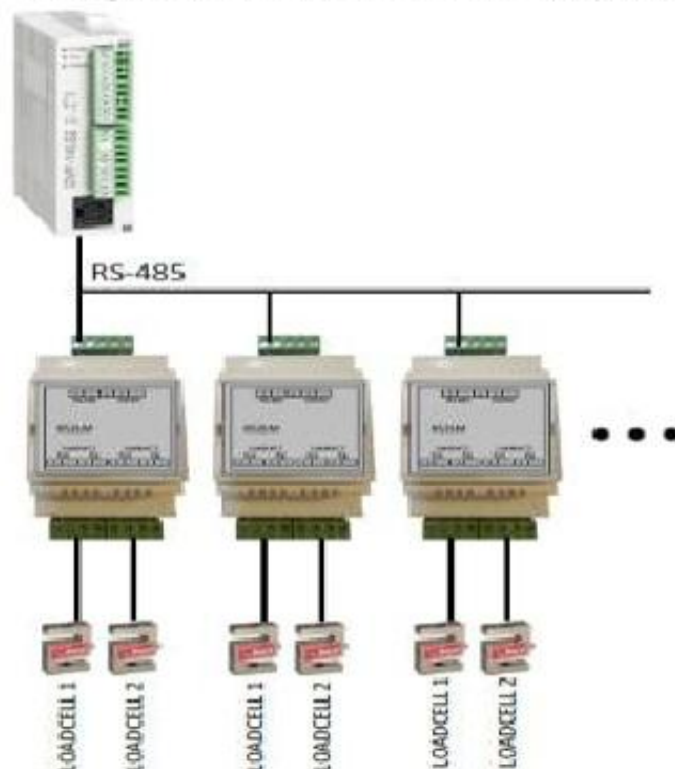
Status Word تعریف شده برای اینورتر دانفوس در ناحیه‌ی PCD مطابق با جدول زیر است:

Bit	Bit status	
	Bit status = 0	Bit status = 1
00	Trip	Control ready
01	-	Drive ready
02	-	Stand by
03	No trip	Trip
04	Not in use	
05	Not in use	
06	Not in use	
07	No warning	Warning
08	Speed/reference	Speed = reference
09	Local operation	Serial comm. Control
10	Out of frequency range	-
11	Disable operation	Operation indication
12	No function	
13	-	Voltage warning high/low
14	-	Current limit
15	--	Thermal warning

7- 27 اتصال لودسل به مبدل مدباس RS2LM و خواندن وزن توسط PLC دلتا

در شبکه مدباس

این ماژول مقدار خروجی لودسل را به صورت عددی در بازه ± 32000 و یا ± 131000 می‌خواند و بوسیله ارتباط RS-485 با پروتکل MODBUS RTU به PLC می‌فرستد.



نحوه اتصال لودسل‌ها در سیستم‌های توزین، که بیشتر از یک لودسل دارند، به شرح زیر است:
برای مثال، اگر 4 لودسل به یک مخزن متصل شده‌اند سیم‌های +Exc , -Exc , +Sig , - به یکدیگر متصل
Sig هر چهار لودسل را به یکدیگر متصل کرده و این چهار سیم را به ورودی ماژول لودسل متصل
نمایید. در واقع، هر چهار لودسل با یکدیگر موازی می‌شوند.

بهترین راه اتصال سیم‌های لودسل‌ها لحیم کردن آنها به یکدیگر است ولی اگر می‌خواهید از جانکشن
باکس استفاده کنید، حتماً از مدل‌های خوب با اتصالات خوب استفاده کنید چون در بعضی موارد مشاهده
شده است که استفاده از جانکشن باکس‌های نامناسب دقت بارگیری را کم می‌کند.

تنظیم مکانیکی لودسل‌ها به شرح زیر است:

برای مثال، اگر وزنه 5 کیلویی را در هر نقطه‌ای از مخزن قرار دهید، وزن اندازه‌گیری شده توسط
لودسل‌ها در هر نقطه، باید 5 کیلوگرم باشد.



مشخصات	RS2LM
تعداد کانال‌های ورودی لودسل	2 کانال
سرعت اسکن	5, 10, 20, 40 Hz
ولتاژ تحریک	5VDC 1A
میزان حساسیت	1/60000, 1/15000 مقدار نامی لودسل
تعداد میانگین‌گیری	1 تا 30 بار
رزولوشن	16 و 18 بیت
فیلتر	فیلتر دیجیتال sinc
توان مصرفی	24 VDC (-15 to +20%) / 3W
حداکثر طول کابل لودسل	100 متر
نحوه اتصال به PLC	دارای پورت RS-485 با پروتکل MODBUS RTU
تعداد نقاط شبکه	(تنظیم توسط دیپ سویچ) 16 مازول از 1 الی 16
سرعت شبکه	(تنظیم توسط دیپ سویچ) 9600, 19200, 57600, 115200
دمای کاری و رطوبت مجاز	5~95%, 0~55°C

آدرس رجیسترهای مودباس در این مازول برای خواندن خروجی لودسل به شرح زیر است:

توسط این رجیستر می‌توان مقدار کانال اول مازول را با دقت 16 بیت و مقدار ± 32000 خواند.	Read	40001 (0H)
توسط این رجیستر می‌توان مقدار کانال دوم مازول را با دقت 16 بیت و مقدار ± 32000 خواند.	Read	40002 (1H)

توسط این رجیستر می‌توان مقدار کانال اول مازول را با دقت 18 بیت و مقدار ± 131000 خواند.	Read (32Bit)	40021 (14H)
توسط این رجیستر می‌توان مقدار کانال دوم مازول را با دقت 18 بیت و مقدار ± 131000 خواند.	Read (32Bit)	40023 (16H)

این رجیستر معادل رجیستر 40021 می‌باشد (فقط 16 بیت کم‌ارزش).	Read	40031 (1EH)
این رجیستر معادل رجیستر 40023 می‌باشد (فقط 16 بیت کم‌ارزش).	Read	40032 (1FH)

25000 - رجیستر 40031 (رجیستر 40031 منهای عدد 25000)	Read	40033 (20H)
25000 - رجیستر 40032 (رجیستر 40032 منهای عدد 25000)	Read	40034 (21H)

آدرس رجیسترهای مودباس برای تنظیم سرعت و تعداد میانگین‌گیری به شرح زیر است:

تنظیم سرعت اسکن لودسل برای کانال اول (5 یا 10 یا 20 یا 40 بار در ثانیه)	Read/Write	40006 (5H)
تعداد میانگین‌گیری برای کانال اول را می‌توان از 1 الی 30 بار تنظیم کرد.	Read/Write	40007 (6H)
تنظیم سرعت اسکن لودسل برای کانال دوم (5 یا 10 یا 20 یا 40 بار در ثانیه)	Read/Write	40008 (7H)
تعداد میانگین‌گیری برای کانال دوم را می‌توان از 1 الی 30 بار تنظیم کرد.	Read/Write	40009 (8H)
تنظیم سرعت اسکن لودسل برای هر دو کانال (5 یا 10 یا 20 یا 40 بار در ثانیه)	Read/Write	40010 (9H)
تعداد میانگین‌گیری برای هر دو کانال را می‌توان از 1 الی 30 بار تنظیم کرد.	Read/Write	40011 (AH)

تنظیمات شبکه در حالت پیش‌فرض بصورت 1, NONE, 8, 115200, Station و number=1 می‌باشد. در صورت نیاز، پارامترهای Station number و Baud rate را می‌توان بوسیله دیپ سویچ‌های داخلی تنظیم کرد. لازم به ذکر است که برای این منظور ابتدا پوشش پلاستیکی ماژول را باید باز کنید.

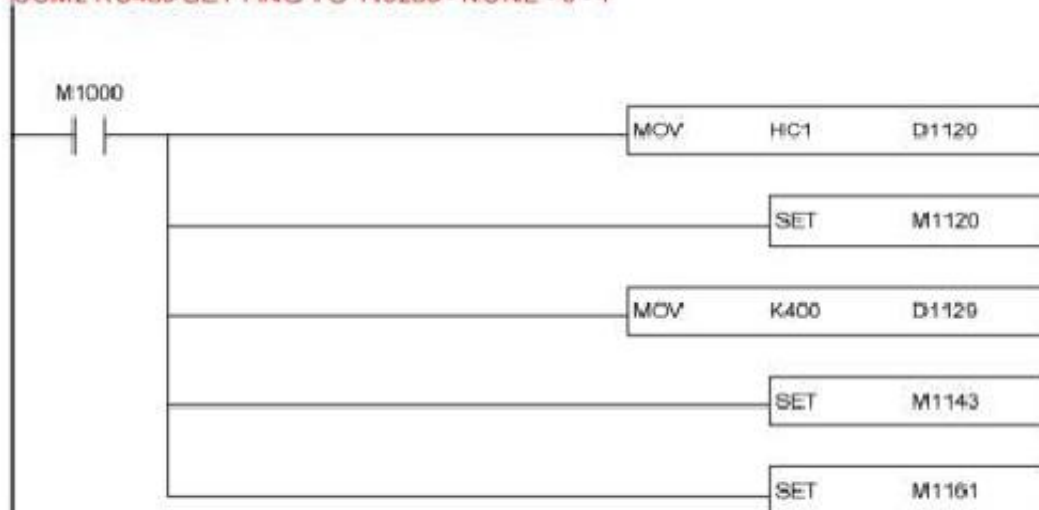
تنظیم دیپ سویچ‌های مربوط به سرعت ارتباط RS-485 و تنظیم دیپ سویچ‌های مربوط به Station number مطابق با جدول زیر است:

Station number	دیپ سویچ 6	دیپ سویچ 5	دیپ سویچ 4	دیپ سویچ 3	Baud rate	دیپ سویچ 2	دیپ سویچ 1
1	OFF	OFF	OFF	OFF	115200	OFF	OFF
2	ON	OFF	OFF	OFF	57600	ON	OFF
3	OFF	ON	OFF	OFF	38400	OFF	ON
4	ON	ON	OFF	OFF	9600	ON	ON
5	OFF	OFF	ON	OFF			
6	ON	OFF	ON	OFF			
7	OFF	ON	ON	OFF			
8	ON	ON	ON	OFF			
9	OFF	OFF	OFF	ON			
10	ON	OFF	OFF	ON			
11	OFF	ON	OFF	ON			
12	ON	ON	OFF	ON			
13	OFF	OFF	ON	ON			
14	ON	OFF	ON	ON			
15	OFF	ON	ON	ON			
16	ON	ON	ON	ON			

برای استفاده از این ماژول توسط PLC کفایت مقدار رجیستر 40001 را خوانده و مقدارهای سرعت و تعداد میانگین‌گیری را در رجیسترهای 40006 و 40007 قرار داده و در برنامه PLC مقدار خوانده شده را کالیبره کرد.

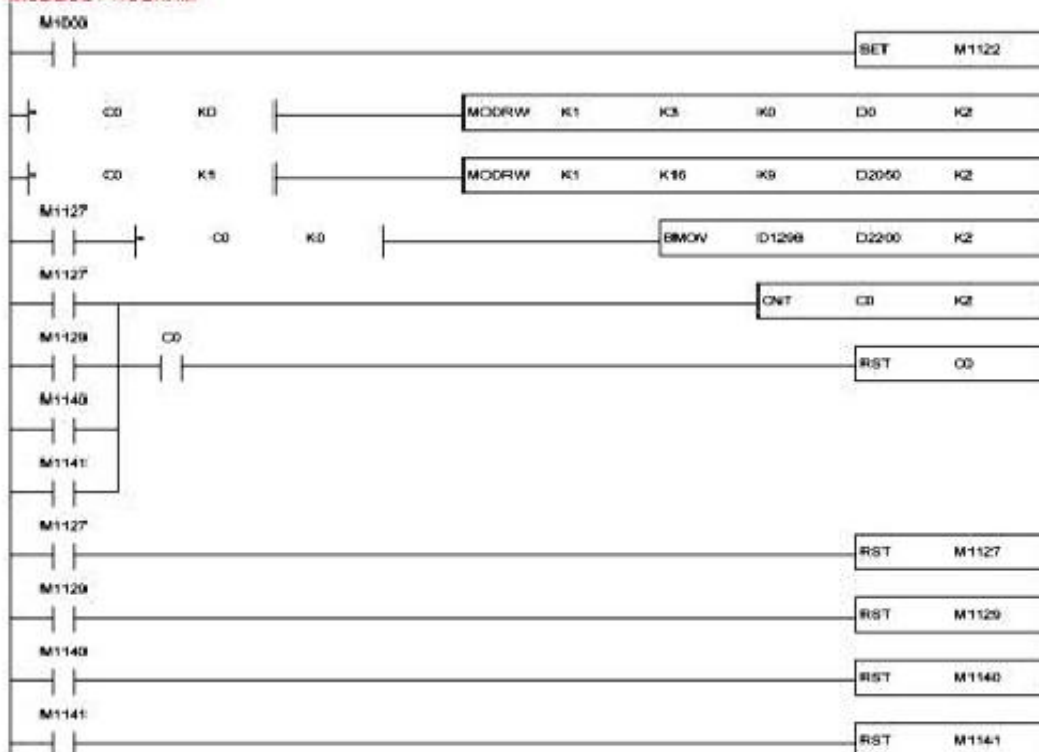
نحوه‌ی برنامه‌نویسی و راه‌اندازی این ماژول در نرم‌افزار WPLSoft به روش دیاگرام نردبانی مطابق با شکل زیر است:

COM2 RS485 SETTING TO 115200 - NONE - 8 - 1



با استفاده از برنامه زیر می‌توان مقدار لودسل‌های متصل شده به کانال اول را در رجیستر D2200 و مقدار لودسل‌های متصل شده به کانال دوم را در رجیستر D2201 خواند و همین‌طور توسط رجیستر D2050 سرعت خواندن از لودسل‌ها و توسط رجیستر D2051 تعداد میانگین‌گیری را تنظیم نمود:

MODBUS PROGRAM

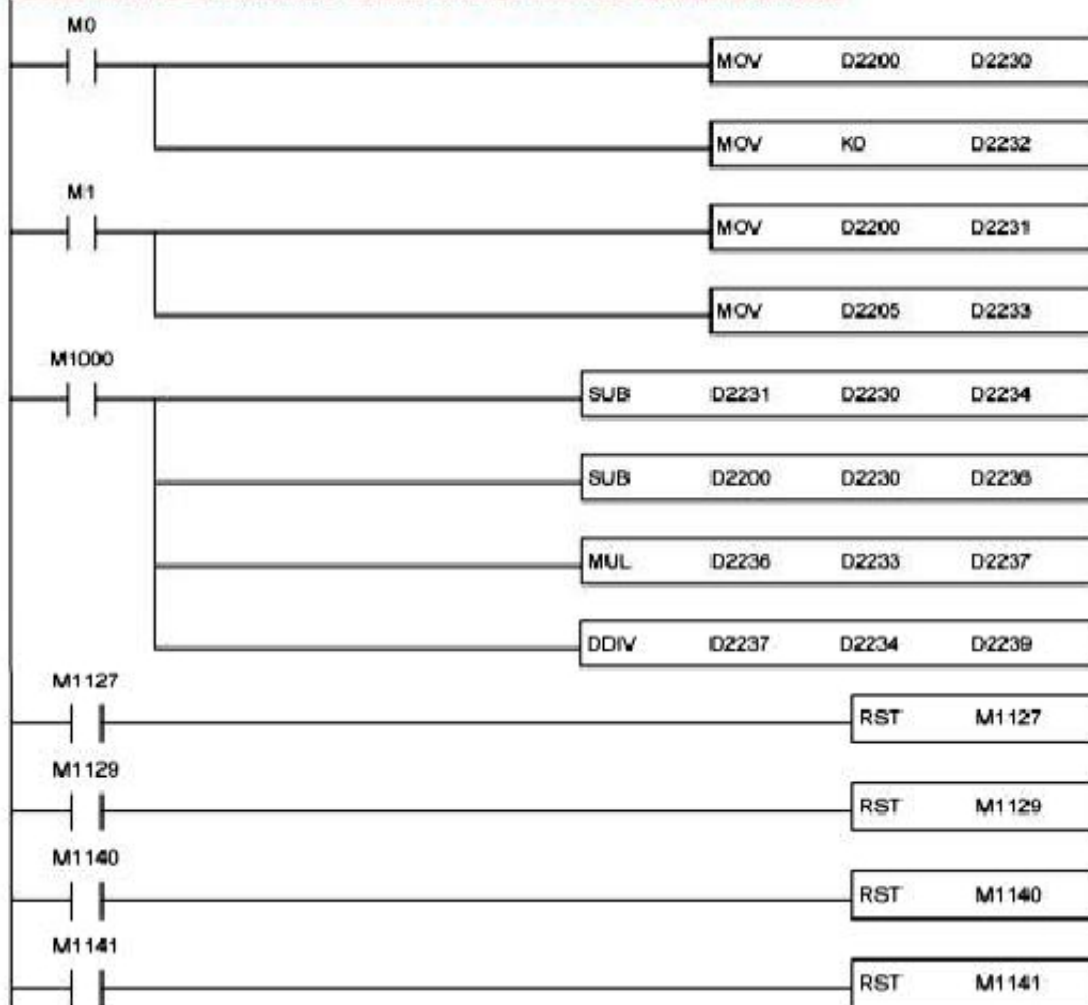


برنامه زیر برای کالیبره کردن کانال اول می‌باشد:

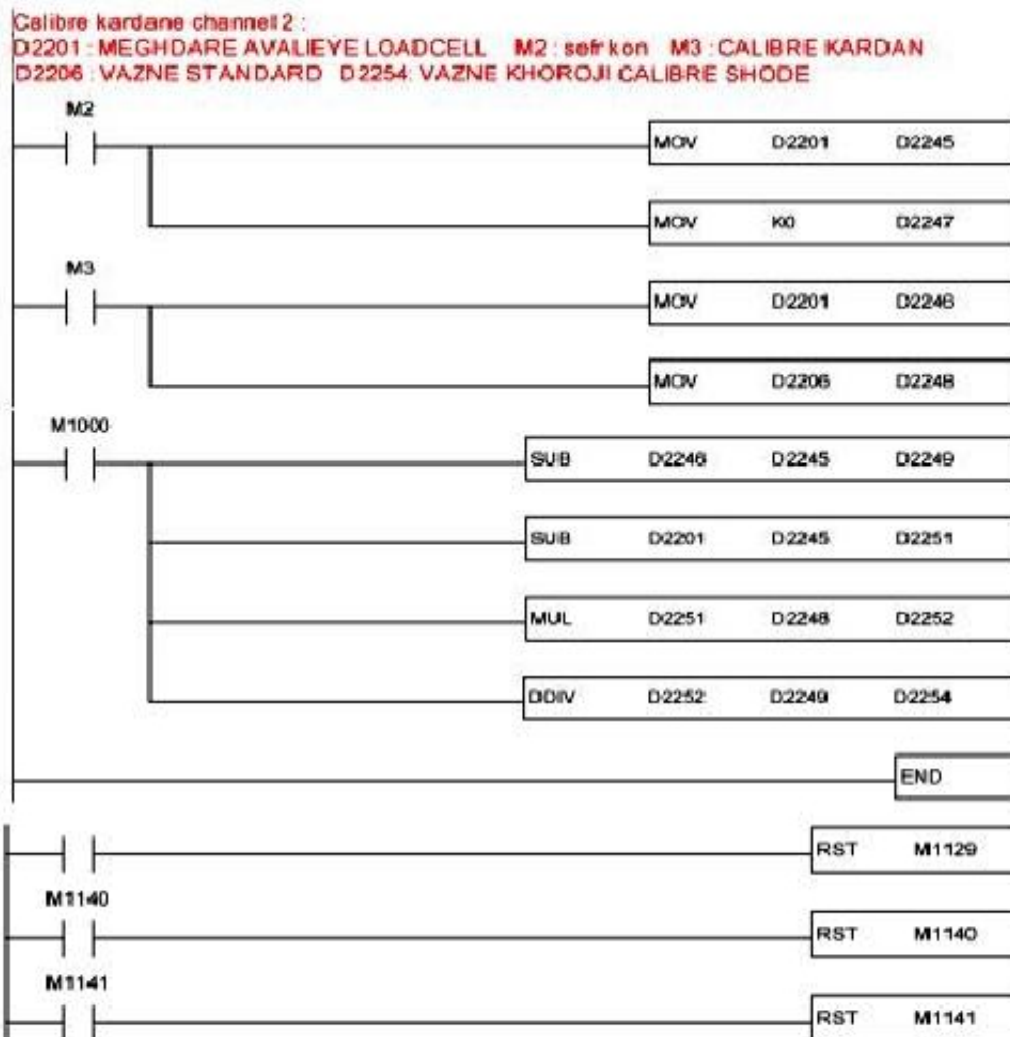
Calibre kardane channel 2 :

D2200 : MEGHDARE AVALIYE LOADCELL M0 : sefr koni M1 : CALIBRE KARDAN

D2205 : VAZNE STANDARD D2230 : VAZNE KHOROJI CALIBRE SHODE

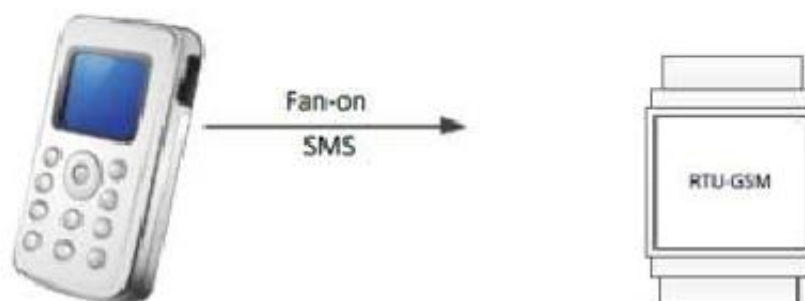


برنامه زیر برای کالیبره کردن کانال دوم است :

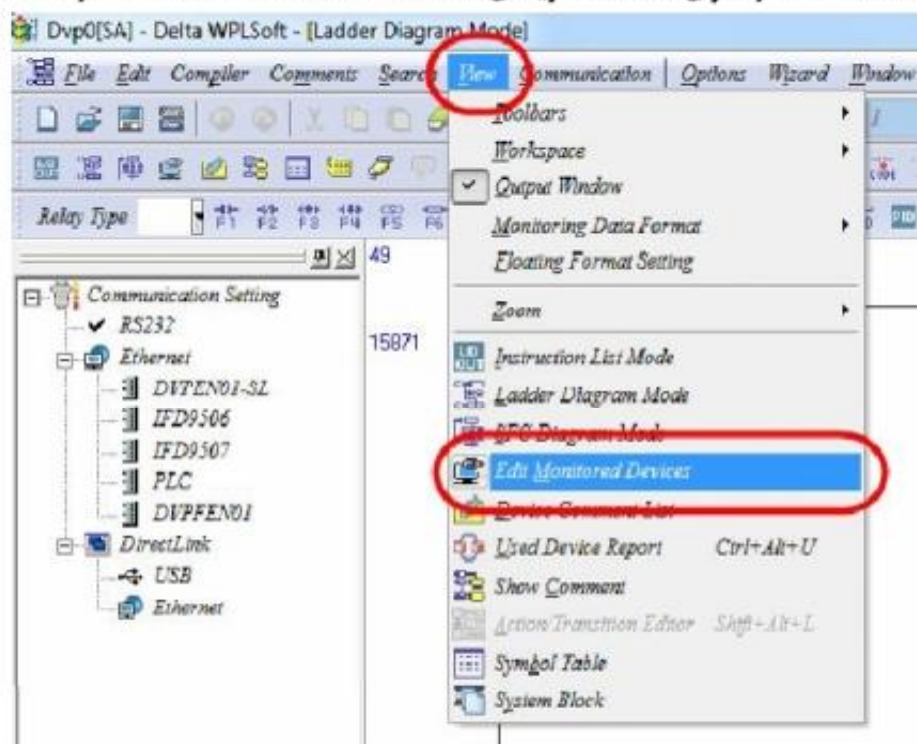


7- 28 دریافت اس ام اس از طریق ماژول GSM توسط شبکه Modbus در PLC های دلتا

در نظر داشته باشید که سیم کارت مورد استفاده برای ماژول GSM نباید "پین کد" داشته باشد. این ماژول از طریق COM2 (RS-485) به PLC وصل می‌شود. چنانچه ماژول GSM آماده باشد (تمام موارد لازم مانند آنتن‌دهی، ارتباط با PLC و ... برقرار باشد) چراغ LED بر روی ماژول GSM خاموش و روشن می‌شود.



در نرم‌افزار WPLSoft از منوی View گزینه‌ی Edit Monitored Device را انتخاب کنید:



در این صفحه در حالی که به PLC بصورت Online متصل می‌باشید، می‌توانید مقادیر رجیسترها را مشاهده یا عوض کنید.

پس از دریافت SMS، شماره موبایل و متن SMS در رجیسترهایی که از قبل در نظر گرفته شده نمایش داده می‌شود.

برای مثال، در برنامه نمونه شماره رجیسترها به ترتیب زیر می‌باشد:

شماره موبایل مربوط به ارسال کننده‌ی SMS به این مازول در رجیسترهای D78~D91 و متن SMS دریافت شده در رجیسترهای D40~D69 نمایش داده می‌شوند.

Device Name	Comment	Present Value	Format
D71		K1	Signed Decimal
D73		K100	Signed Decimal
D75		K1	Signed Decimal
D77		K1	Signed Decimal
D78		*9	ASCII
D79		*8	ASCII
D80		*9	ASCII
D81		*1	ASCII
D82		*9	ASCII
D83		*4	ASCII
D84		*5	ASCII
D85		*9	ASCII
D86		*7	ASCII
D87		*7	ASCII
D88		*6	ASCII
D89		*2	ASCII
D90		**	ASCII
D91		**	ASCII
D93		K39	Signed Decimal

وضعیت مازول GSM

درصد آنتن دهی

وجود اعلام بوجود سیم کارت

چنانچه این رجیستر یک شود به معنای دریافت SMS جدید می باشد.

شماره موبایل اسنسی دریافت شده
989194597792

Device Name	Comment	Present Value	Format
D40		*F	ASCII
D41		*a	ASCII
D42		*n	ASCII
D43		*-	ASCII
D44		*o	ASCII
D45		*h	ASCII
D46		**	ASCII
D47		**	ASCII

هتین اسنسی دریافت شده

کد اسکی هر کاراکتر وقتی
مقدار رجیستر با فرمت دسیمال
نمایش داده شود

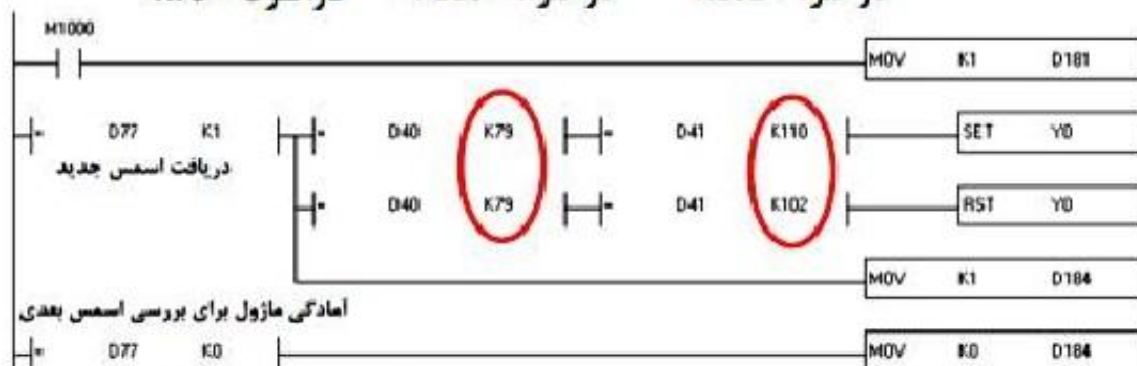
Device Name	Comment	Present Value	Format
D40		K70	Signed Decimal
D41		K97	Signed Decimal
D42		K110	Signed Decimal
D43		K45	Signed Decimal
D44		K111	Signed Decimal
D45		K110	Signed Decimal
D46		K0	Signed Decimal
D47		K0	Signed Decimal

نحوه برنامه‌نویسی در PLC برای دریافت SMS جدید مطابق با فلوجارت زیر می‌باشد:



در برنامه زیر چنانچه SMS ، On دریافت شود خروجی Y0 روشن شده و چنانچه SMS ، Off دریافت شود، خروجی Y0 خاموش خواهد شد.

کاراکتر f = K102 کاراکتر n = K110 کاراکتر O = K79



برای معرفی هر کاراکتر یک کد در نظر گرفته شده است. مثلاً حرف A با عدد 65 نمایش داده می‌شود.

در این مازول همه رجیسترها با کدهای اسکی، کاراکتر خود را نشان می‌دهند. با توجه به جدول زیر می‌بینیم که حتی اعداد 0~9 کد اسکی دارند. به عنوان مثال، عدد 9 با کد اسکی 57 نشان داده می‌شود.

ASCII Codes	Symbol
10	Enter
32	(Space)
33	!
34	"
35	#
36	\$
37	%
38	&
40	(
41)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>

ASCII Codes	Symbol
63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z
92	\
95	_

ASCII Codes	Symbol
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
126	~
127	DEL

نحوه ارسال SMS توسط PLC به شرح زیر است:



The Fan is off
SMS



در این ماژول GSM می‌توان 6 شماره موبایل را تعریف کرد. رجیسترهای تعریف شماره موبایل در این مثال به شرح زیر است:

رجیسترهای D150~D163 برای ذخیره شماره موبایل نفر اول

رجیسترهای D164~D173 برای ذخیره شماره موبایل نفر دوم

رجیسترهای D200~D213 برای ذخیره شماره موبایل نفر سوم

رجیسترهای D215~D228 برای ذخیره شماره موبایل نفر چهارم

رجیسترهای D230~D243 برای ذخیره شماره موبایل نفر پنجم

رجیسترهای D245~D268 برای ذخیره شماره موبایل نفر ششم می‌باشند.

با یک کردن هر یک از بیت‌های رجیستر (D183) 400094 می‌توان تعیین کرد که SMS به کدام شماره موبایل ارسال شود. از بیت دوم این رجیستر برای جواب دادن به آخرین نفری که به ماژول SMS فرستاده است، استفاده می‌شود.

هر بیت از این رجیستر، برای فعال کردن شماره موبایل مورد نظر برای ارسال SMS می‌باشد. 0 به معنی نفرستادن و 1 به معنی فرستادن می‌باشد.

برای ارسال یک SMS به چندین نفر باید مطابق با مراحل گفته شده در زیر اقدام نمایید:

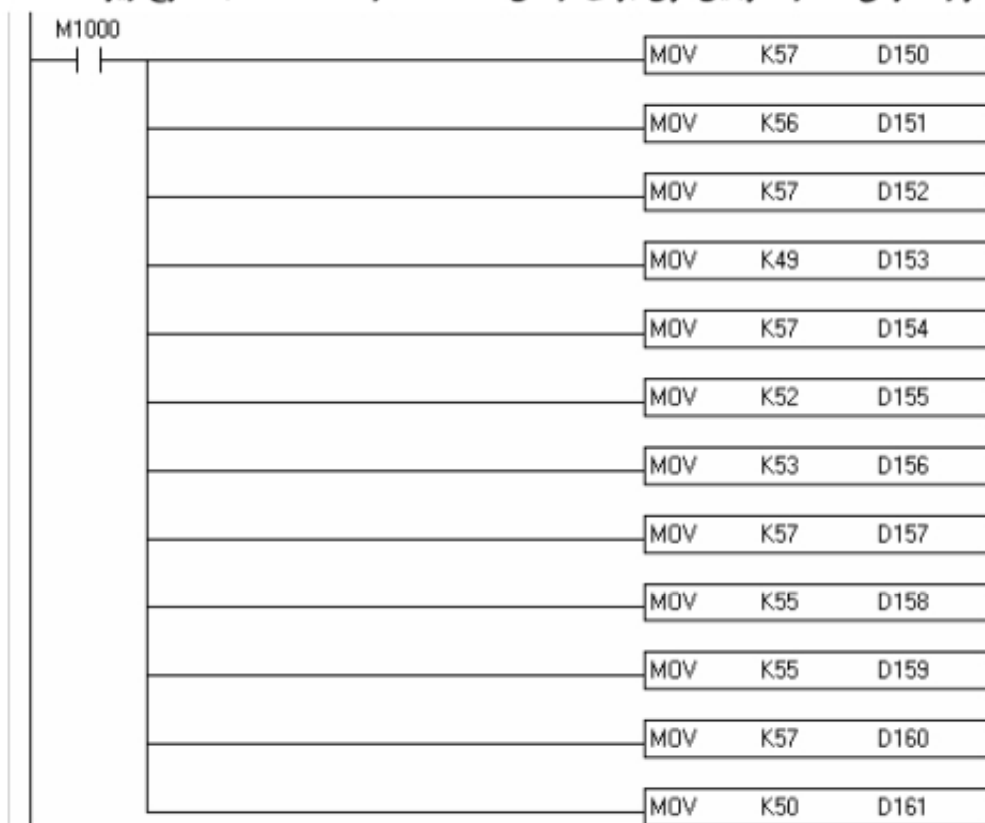


شماره موبایل را باید با کد کشور وارد کرد. برای مثال، 989194597792.

در برنامه زیر، کدهای کاراکتری این شماره موبایل در رجیسترهای D150~D161 بعنوان شماره موبایل 1 وارد شده‌اند. در نظر داشته باشید که بهتر است با تابع MOV مقدار کد اسکی هر کاراکتر را

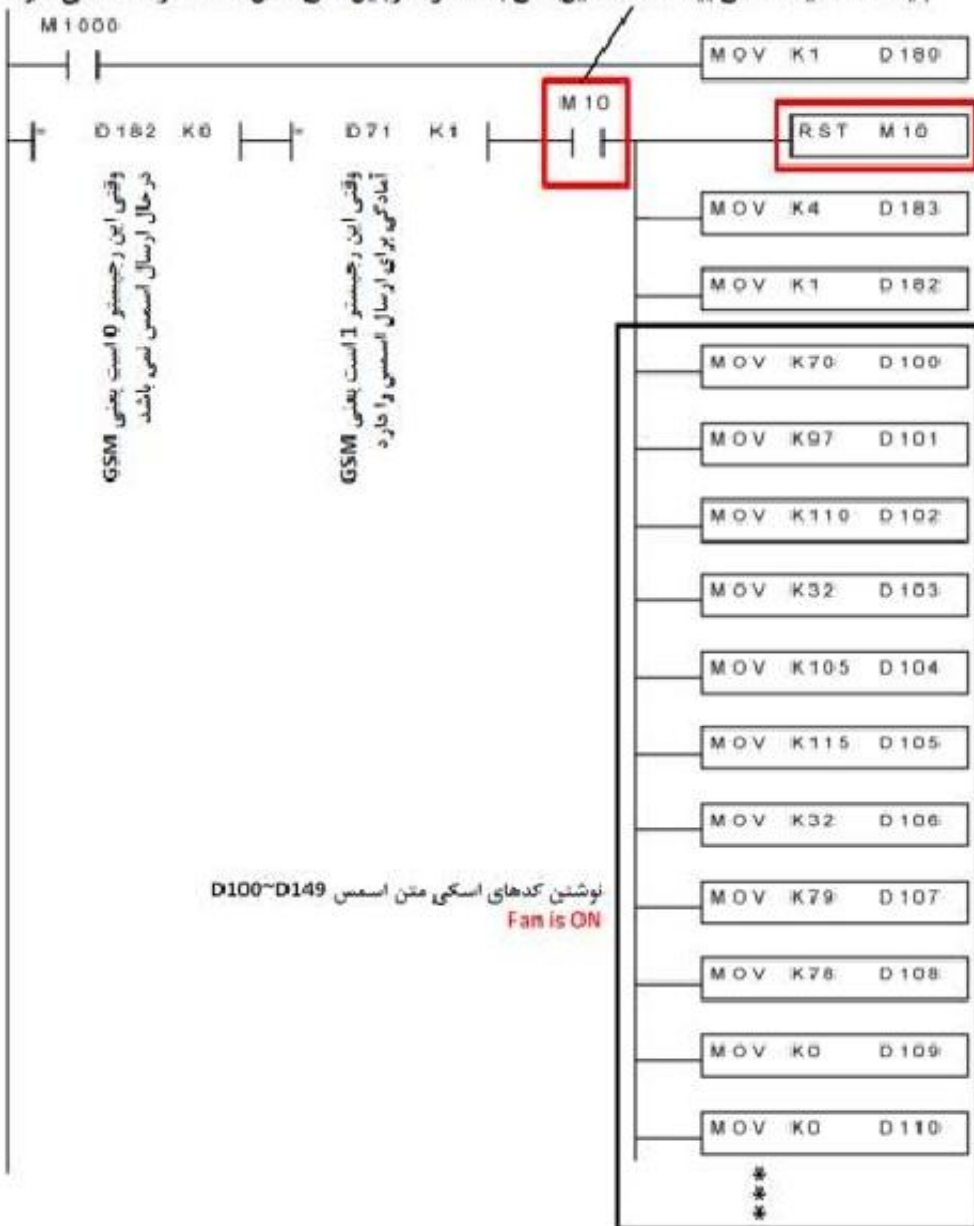
در رجیستر مورد نظر قرار دهید تا بعد از قطع و وصل برق یا تعویض PLC اطلاعات در برنامه ذخیره شوند.

برنامه‌ی وارد کردن شماره موبایل اول برای ارسال SMS در PLC دلتا به شرح زیر است :



چنانچه در رجیستر 400091 (D180) عدد 1 را قرار دهید، فقط از 8 بیت کم‌ارزش رجیسترها، برای ارسال کاراکترها استفاده می‌شود. در این صورت، می‌توان مقدار کد اسکی هر کاراکتر را در هر رجیستر قرار داد (بهتر است همیشه 1 باشد).

با یک لحظه یک شدن بیت M10. این متن به شماره موبایل‌های فعال شده، فرستاده می‌شود



از رجیستر (D71) 400151 برای نمایش وضعیت GSM استفاده شده است:

اگر D71=0 بود، یعنی ماژول آماده ارسال SMS نمی‌باشد.

اگر D71=1 بود، یعنی ماژول آماده ارسال SMS است.

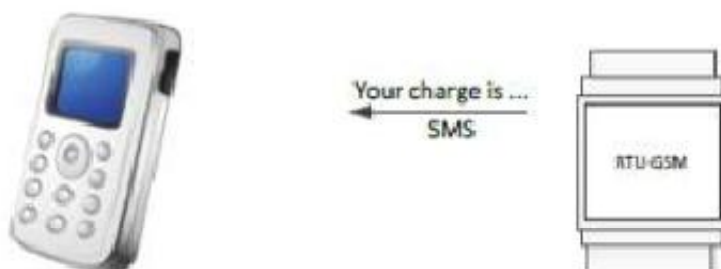
اگر D71=2 بود، یعنی ماژول در حال ارسال SMS است.

اگر D71=3 بود، یعنی ارسال SMS به پایان رسیده است.

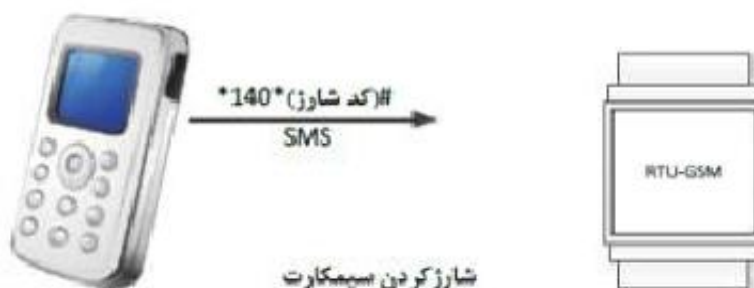
وقتی D71=3 باشد، باید رجیستر D182 صفر شود.



برای مشاهده‌ی مقدار شارژ موجود در سیم کارت نصب شده در مازول GSM باید مراحل شکل زیر را طی کرد:



برای شارژ کردن سیم کارت از طریق SMS باید به روش گفته شده در زیر اقدام کرد:
همان متنی که برای شارژ کردن سیم کارت در موبایل استفاده می‌کنید را به شماره‌ای که در مازول GSM قرار دارد، SMS کنید.



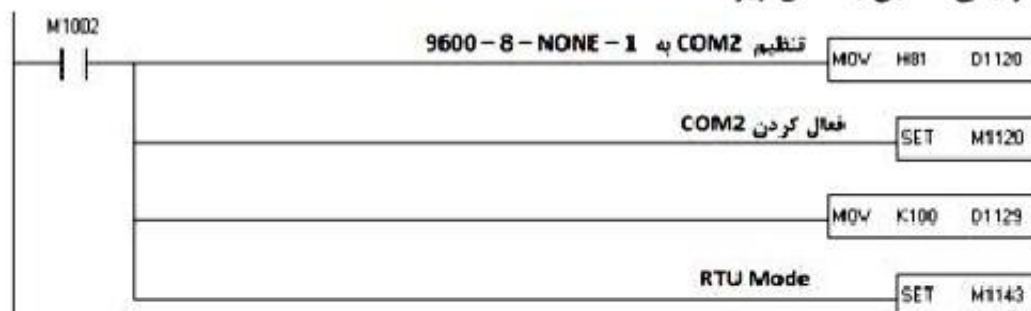
رجیسترهای مورد استفاده برای ارتباط Modbus در این برنامه با مازول GSM در سمت PLC جهت نوشتن دیتا در شبکه Modbus به شرح زیر است:

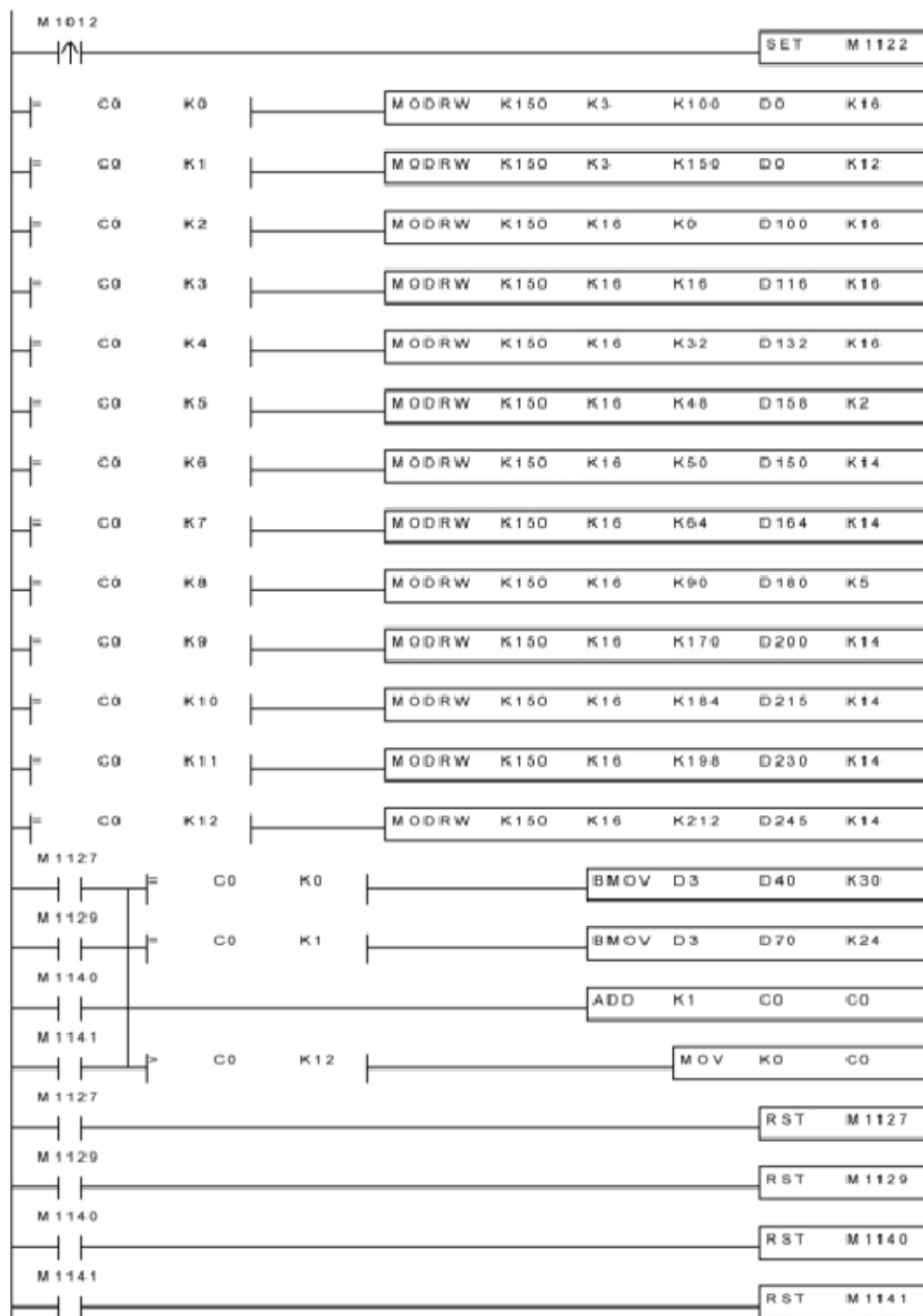
توضیحات	آدرس رجیسترهای مودباس	آدرس رجیسترهای PLC در این مثال	توضیحات
کاراکترهایی که در این رجیسترها نوشته شوند؛ ارسال خواهند شد.	400001~400050 (0 ~ 49)	D100~D149	متن ارسال شونده
شماره موبایل را برای ارسال SMS به صورت کاراکترها را بصورت زیر وارد کنید: 989123689654	400051~400064 (50 ~ 63)	D150~D163	شماره موبایل 1
	400065~400078 (64 ~ 77)	D164~D177	شماره موبایل 2
	400171~400184 (170 ~ 183)	D200~D213	شماره موبایل 3
	400185~400198 (184 ~ 197)	D215~D228	شماره موبایل 4
	400199~400212 (198 ~ 211)	D230~D243	شماره موبایل 5
	400213~400226 (212 ~ 225)	D245~D258	شماره موبایل 6
-	400079~400090 (78 ~ 89)	-	دزرو
0: از تمام 16 بیت یک رجیستر استفاده می‌شود. 1: فقط از 8 بیت کم‌ارزش رجیسترها استفاده می‌شود.	400091 (90)	D180	نحوه ارسال کاراکترها از GSM به PLC
0: ابتدا بایت با ارزش سپس کم‌ارزش ارسال می‌شود. 1: ابتدا بایت کم‌ارزش و سپس با ارزش ارسال می‌شود.	400092 (91)	D181	ترتیب ارسال کاراکترها از GSM به PLC
چنانچه مقدار این رجیستر از C به عددی در بازه‌های زیر تغییر کند، مازول شروع به ارسال SMS می‌کند. ارسال متن بعنوان SMS به موبایل: 1~50 ارسال متن بعنوان مقادیر رجیستر به مازول GSM دیگر: 51~100	400093 (92)	D182	شماره ارسال متن SMS
با یک کردن هر یک از بیت‌های این رجیستر، می‌توان تعیین کرد که ا سمس به کدام شماره موبایل ارسال شود.	400094 (93)	D183	ارسال به شماره موبایل مورد نظر
در برنامه PLC، پس از این که SMS دریافت شد، باید با یک کردن این بیت تایید دریافت SMS را به مازول اعلام کرد.	400095 (94)	D184	تایید دریافت SMS

رجیسترهای مورد استفاده برای ارتباط Modbus در این برنامه با مازول GSM در سمت PLC جهت خواندن دیتا از شبکه Modbus به شرح زیر است:

توضیحات	آدرس رجیسترهای مودباس	آدرس رجیسترهای PLC در این مثال
متن SMS دریافت شده است.	400101~400150	D40-D69
0: به معنی عدم آمادگی ماژول GSM است. 1: به معنی ماژول GSM آماده ارسال SMS است. 2: به معنی ماژول GSM در حال ارسال SMS است. 3: به معنی پایان ارسال SMS است.	400151	D71
-	400152	D73
0: به معنی عدم وجود سیم کارت در ماژول GSM است. 1: به معنی وجود سیم کارت در ماژول GSM است.	400153	D75
اگر این رجیستر مساوی عدد 1 شود بمعنای دریافت یک SMS جدید می‌باشد	400154	D77
شماره موبایل آخرین SMS دریافتی در این رجیستر ذخیره می‌شود. (مثلاً، 98912654145)	400155~400161	D78~D91
-	400162	D93
این رجیستر ورژن ماژول GSM را نشان می‌دهد.		D94
		Os version

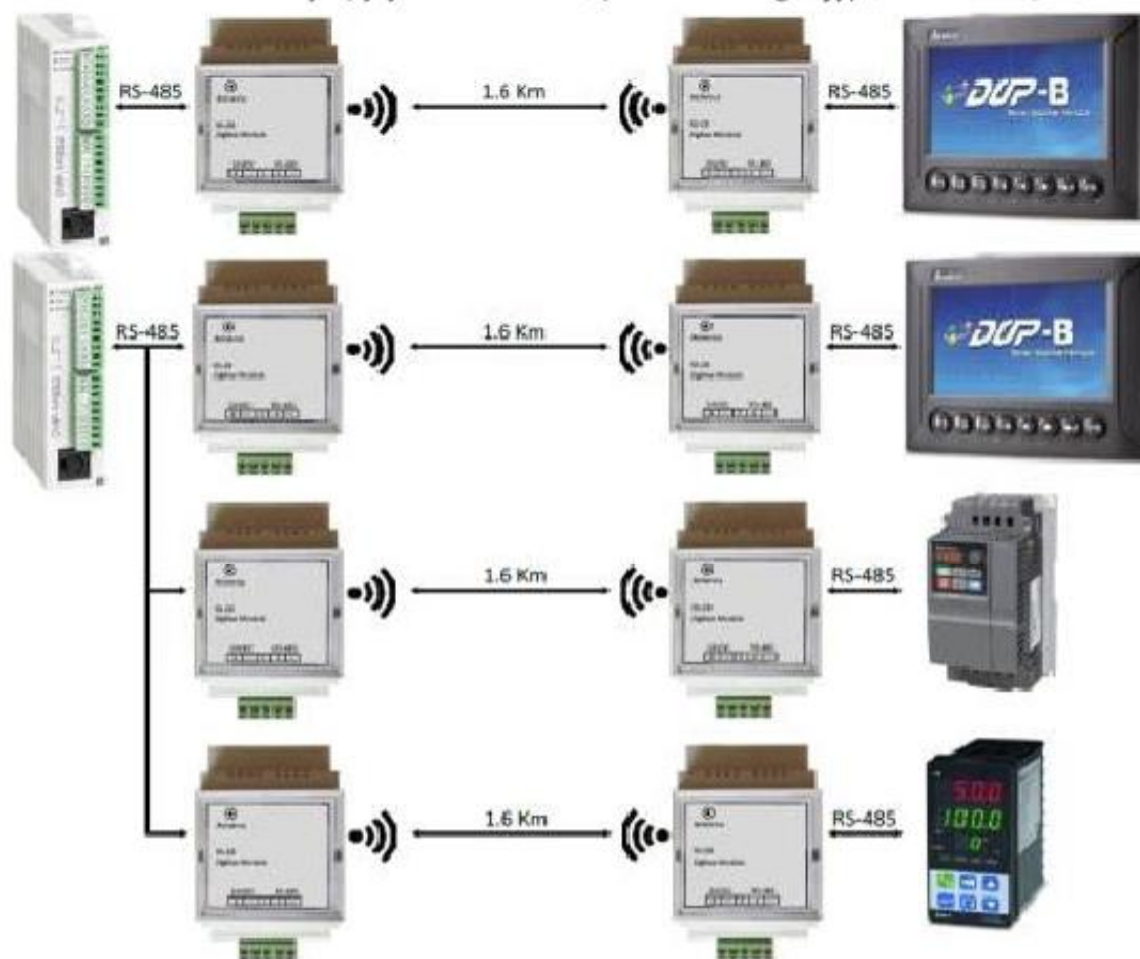
برنامه‌ی کنترلی این مثال برای راه‌اندازی و ارتباط مودباس در PLC دلتا با ماژول GSM به روش دیاگرام نردبانی مطابق با شکل زیر است:





7- 29 ارتباط وایرلس چندین وسیله در شبکه‌ی RS-485 با پروتکل Modbus

توسط ماژول RS-ZB که یک ماژول ارتباط سریال RS-485 بی‌سیم است، می‌توان بین تجهیزات تحت شبکه‌ی RS-485 با پروتکل Modbus ارتباط Wireless برقرار کرد.



همانطور که در شکل بالا مشاهده می‌کنید، برای هر دستگاه تحت شبکه از دو ماژول بی‌سیم استفاده شده است. دلیل آن این است که در این ماژول از واسطه Zigbee برای ارتباط بی‌سیم استفاده شده است. واسطه Zigbee یک درگاه بسیار امن می‌باشد که نفوذ در آن تقریباً غیرممکن است. به همین دلیل، برای بالا بردن سطح امنیت و صفر کردن تداخل امواج و نویزپذیری باید در هر دستگاه برای ارتباط بی‌سیم از یک جفت ماژول بی‌سیم استفاده کرد.

مشخصات این ماژول به شرح زیر است:

RS-ZB	مشخصات
IEEE 802.15.4	استاندارد
2.410~2.475GHz	فرکانس کاری
20dBm	توان ارسال
1.6 km (1 mile)	فاصله بین دو ماژول (فضای باز)
DSSS (Direct Sequence Spread Spectrum)	روش انتقال اطلاعات
RS-485	پورت ارتباطی
24VDC, -15%/+20%, 2W	تغذیه ورودی

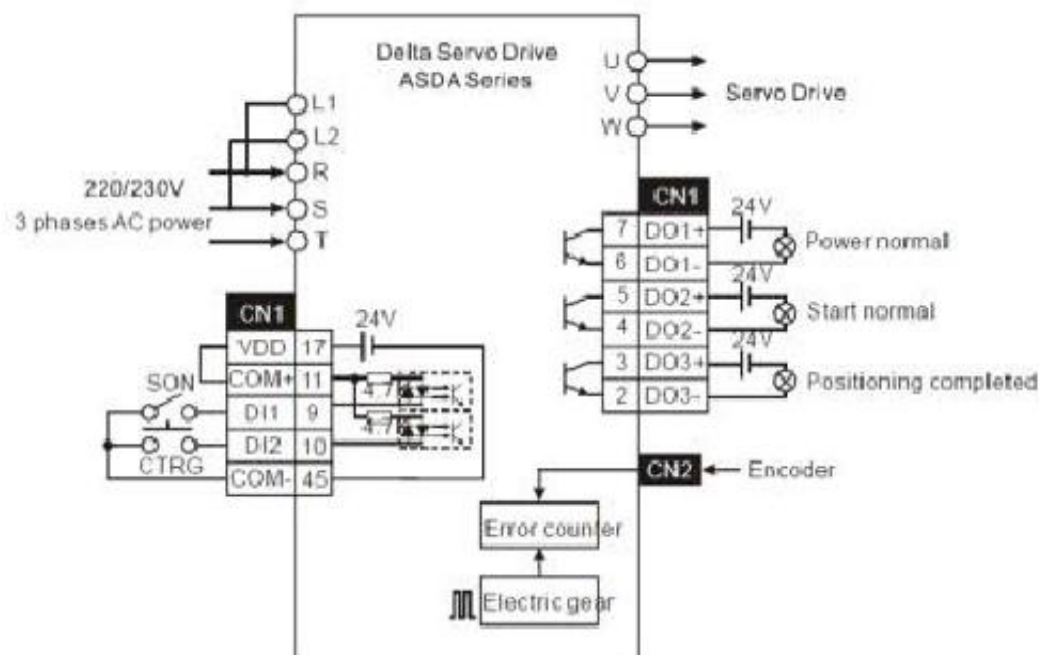
برای کار با این ماژول تنها نکته‌ای که باید رعایت شود این است که نرخ تبادل دیتا (Baud rate) تمام تجهیزات تحت شبکه را برابر با 9600 تنظیم نمایید. حال به راحتی و با استفاده از مطالبی که در کتاب فوق برای شبکه‌ی RS-485 با پروتکل Modbus گفته شد، می‌توانید یک شبکه بی‌سیم را راه‌اندازی نمایید.

7-30 ارتباط شبکه بین PLC و سروودرایو سری ASDA دلتا

AC Servo drive control panel



سیم‌بندی سروودرایو AC سری ASDA دلتا مطابق با شکل زیر است:



هدف از کنترل در این مثال به شرح زیر است:

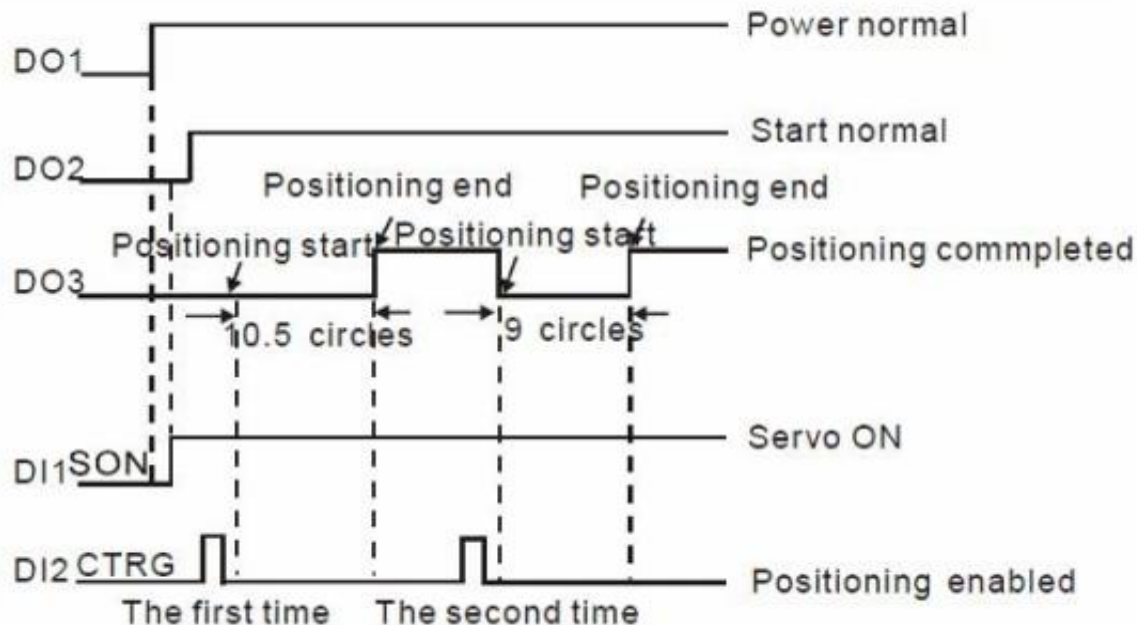
خواندن موقعیت هدف از سرودرایو AC (موقعیت افزایشی) توسط دستورالعمل MODRD.
تنظیم موقعیت هدف سرودرایو AC (موقعیت افزایشی) توسط دستورالعمل MODRW.
فعال کردن شروع به کار و عمل موقعیت‌یابی سرودرایو توسط نقاط ورودی DI1~DI2 وقتی که هر یک از شستی‌ها فشرده شود.

نشان دادن وضعیت سرودرایو توسط نمایشگرها با استفاده از نقاط خروجی DO1~DO3.
تنظیمات پارامترهای سرودرایو AC سری ASDA دلتا به شرح زیر است:

پارامتر	مقدار تنظیمی	توضیحات
P1-01	1	مد کنترل و جهت خروجی
P1-33	1	مد کنترل موقعیت Pr
P2-10	101	نخستین ترمینال ورودی دیجیتال (DI1)
P2-11	10E	دومین ترمینال ورودی دیجیتال (DI2)
P2-15	0	ششمین ترمینال ورودی دیجیتال (DI6)
P2-16	0	هفتمین ترمینال ورودی دیجیتال (DI7)
P2-17	0	هشتمین ترمینال ورودی دیجیتال (DI8)
P2-18	101	نخستین ترمینال خروجی دیجیتال (DO1)
P2-19	10E	دومین ترمینال خروجی دیجیتال (DO2)
P2-20	10E	سومین ترمینال خروجی دیجیتال (DO3)
P3-00	1	تنظیمات آدرس ارتباطات
P3-01	1	سرعت ارتباطات و Baud rate برابر 9600
P3-02	1	مد ASCII پروتکل MODBUS و فرمت داده: (7, E, 1)
P3-03	1	اخطار و استوپ در صورت رخ دادن خطای ارتباطات
P3-05	2	فرمت ارتباطات RS-485
P3-06	0	تابع ارتباطات ورودی دیجیتال

اگر سرو درایو AC به صورت عادی شروع به کار نکرد، تنظیم پارامترها اشتباه می‌باشد. برای رفع این عیب، پارامتر P02-08 را برابر 10 تنظیم نمایید تا تنظیمات تمام پارامترها ریست شده و به مقدار پیش فرض کارخانه بازگردد. سپس پارامترها را مطابق جدول بالا تنظیم نمایید.

مراحل عملکرد به شرح زیر است :



پس از وصل کردن مجدد برق سرودرایو AC پارامترها را تنظیم نمایید. اگر خطایی رخ نداد، نمایشگر خروجی 1 (DO1) فعال می‌شود.

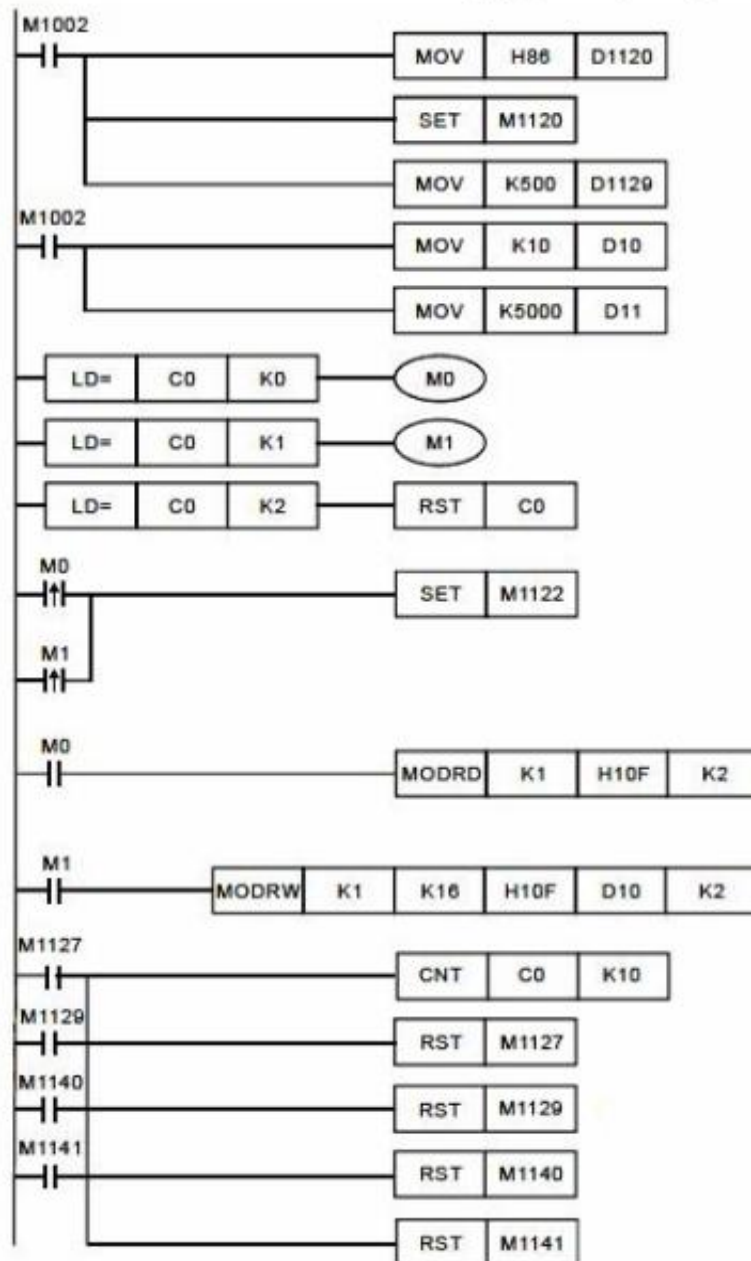
وقتی که نمایشگر حالت عادی سرو فعال شد، با فعال کردن ورودی دیجیتال DI1 سرو فعال (SON) می‌شود. اگر خطایی رخ نداد و سرو به صورت نرمال شروع به کار کرد، نمایشگر خروجی دیجیتال 2 فعال می‌شود.

وقتی که نمایشگر START NORMAL فعال شد، با فعال کردن ورودی دیجیتال 2 موقعیتیابی CTRG فعال می‌شود. سروموتور برای 10/5 دور شروع به چرخیدن کرده و سپس نمایشگر تکمیل شدن موقعیت (POSITIONING COMPLETED) در خروجی دیجیتال 3 فعال می‌شود.

ابزارهای مورد استفاده در این مثال به شرح زیر است :

ابزار	تابع
M0	اجرای دستورالعمل MODRD با خواندن تعداد چرخش و تعداد پالس از ورودی داخلی موقعیت 1
M1	اجرای دستورالعمل MODRW با تنظیم تعداد چرخش و تعداد پالس از ورودی داخلی موقعیت 1

برنامه کنترلی این مثال مطابق با شکل زیر است :



توضیحات خط به خط برنامه به شرح زیر است:

- خط 1 ← تنظیم ارتباطات سریال با فرمت 1 . E . 7 . 96000
- خط 2 ← حفظ کردن و نگهداری تنظیمات ارتباطات سریال
- خط 3 ← تنظیم زمان وقفه در عمل تبادل دیتا (time out) برابر با 500ms
- خط 4 ← تنظیم تعداد چرخش موقعیت‌یابی داخلی 1
- خط 5 ← اجرای دستورالعمل MODRD

خط 6 ← اجرای دستورالعمل MODRW

خط 8 ← درخواست ارسال داده

خط 10 ← خواندن تعداد چرخش و تعداد پالس از موقعیتیابی داخلی 1 و سپس ذخیره کردن آن در D1050 , D1051

خط 11 ← نوشتن محتویات D11 , D12 در داخل کدهای هگز H10F و H110 سرودرایو AC

خط 12 ← وقتی که ارتباطات شبکه تکمیل شد، در هر بار شمارنده C0 یک بار شمرده می‌شود.

خط 13 ← ریست کردن فلگ M1127

خط 14 ← ریست کردن فلگ M1129

خط 15 ← ریست کردن فلگ M1140

خط 16 ← ریست کردن فلگ M1141

توضیحات برنامه به شرح زیر است:

مقداردهی اولیه پورت ارتباطات سریال RS-485 از سمت PLC و تنظیم فرمت ارتباطات شبکه MODBUS با پروتکل 1 , E , 7 , 9600 , ASC11 که باید فرمت ارتباطات شبکه سریال RS-485 در سمت سرودرایو AC مانند PLC باشد.

وقتی که M1=ON شود، با خواندن تعداد چرخش و تعداد پالس موقعیت داخلی 1 و ذخیره آن در D1050 , D1051 دستورالعمل MODRD K1 H10F K2J اجرا خواهد شد.

وقتی که M1=ON شود، دستورالعمل MODWR K1 K16 H10F D10 K2 اجرا خواهد شد. در این صورت، محتویات D10 در داخل H10F (تعداد چرخش موقعیت داخلی 1) و D11 در داخل H10 (تعداد پالس از موقعیتیابی داخلی) سرودرایو نوشته می‌شود.

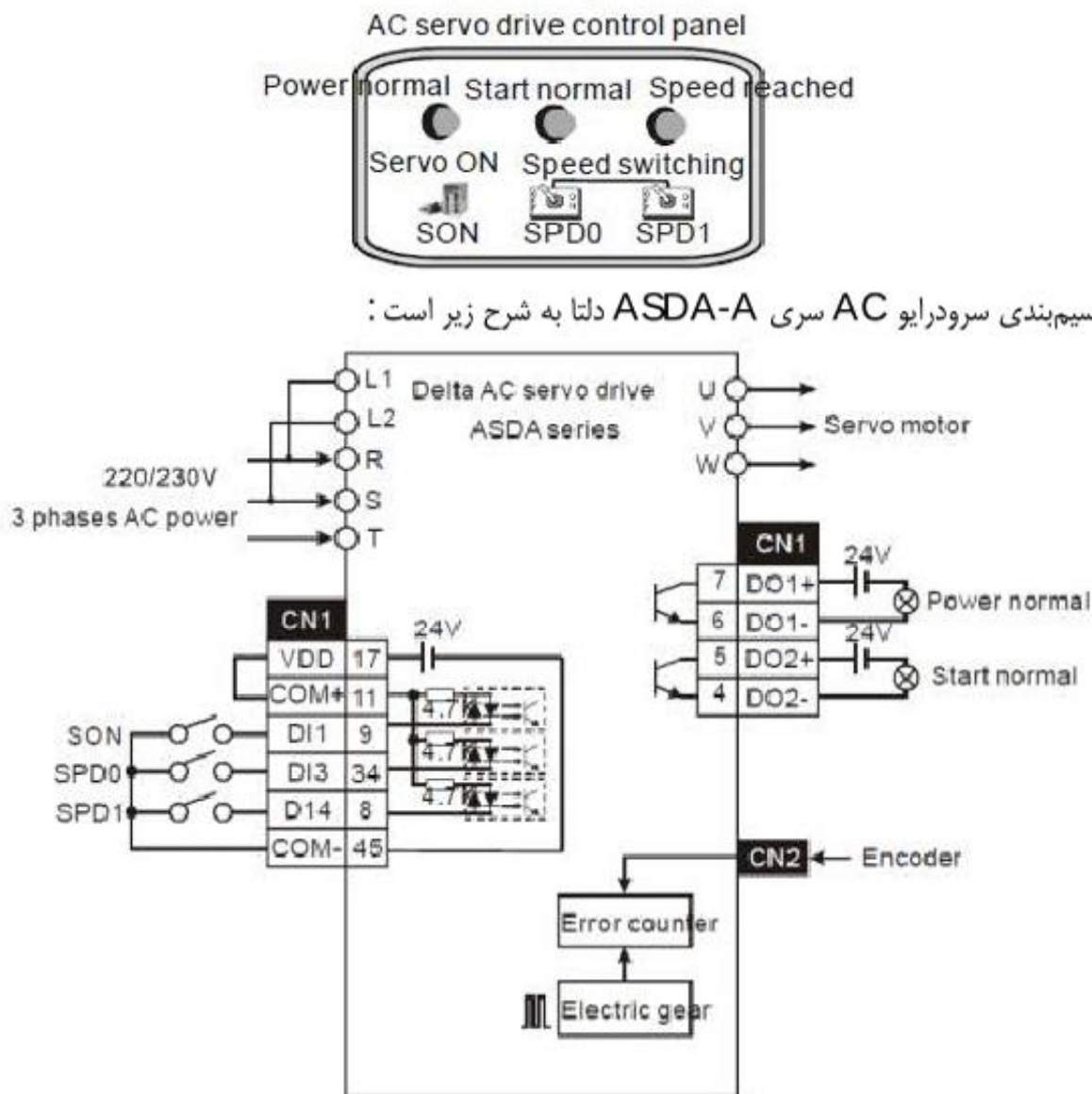
هر دو سیگنال استارت و سیگنال راه‌اندازی توسط سوئیچ‌های از سرودرایو AC و سیم‌بندی خارجی کنترل خواهند شد. برای روش سیم‌بندی به دیاگرام سیم‌بندی مراجعه نمایید.

فقط 4 فلگ خاص برای ارتباطات سریال با پروتکل MODBUS وجود دارد: فلگ M1127 برای ارتباطات نرمال و M1141 , M1140 , M1129 برای خطاهای ارتباطات سریال می‌باشند. وقتی که هر یک از 4 فلگ فعال باشند، شمارنده C0 یک بار شمرده می‌شود. بنابراین، برنامه به شما اطمینان می‌دهد که ارتباطات سریال توسط مانیتور کردن وضعیت‌های on/off چهار فلگ و اجرای دو دستورالعمل MODBUS توسط تنظیم مقدار شمارنده C0 قابل اعتماد می‌باشد.

با راه‌اندازی PLC و RUN کردن آن، عمل READ / WRITE برای سرودرایو AC مکرراً مطابق با دستورالعمل LD= اجرا خواهد شد.

نکته: دقت کنید که این برنامه فقط برای سرودرایو AC سری ASDA دلتا قابل اجرا می‌باشد.

7- 31- ارتباطات سریال بین PLC و سروودرایو AC سری ASDA دلتا



هدف از کنترل در این مثال به شرح زیر است:

خواندن سرعت چرخش سروودرایو و ذخیره آن در D0 توسط دستورالعمل MODRD.
کنترل چرخش سرووموتور در 3 سرعت ثابت یا سرعت مشخص توسط دستورالعمل MODRW با سوئیچ کردن SPD0 , SPD1.

تعریف سوئیچ‌های سرعت سرودرایو AC در این مثال به شرح زیر است:

نمایش وضعیت سرودرایو AC توسط خروجی دیجیتال سرودرایو از DO1 الی DO3 انجام می‌شود.

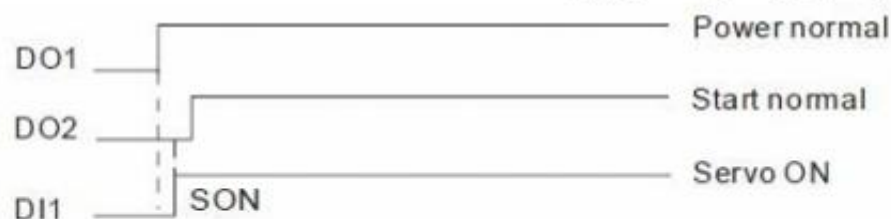
وضعیت SPD0	وضعیت SPD1	تابع
OFF	OFF	سرعت صفر
ON	OFF	انتخاب نخستین سرعت توسط تنظیم پارامتر P01-09 (تعیین کردن سرعت توسط محتویات داخل D9، در این برنامه عدد ثابت K1500 می‌باشد. سرعت چرخش سروموتور برابر 1500RPM و جهت آن راستگرد است.)
OFF	ON	انتخاب دومین سرعت توسط تنظیم پارامتر P01-10 (تعیین کردن سرعت توسط محتویات داخل D10، در این برنامه عدد ثابت K-1500 می‌باشد. سرعت چرخش سروموتور برابر 1500RPM و جهت آن چپگرد است.)
ON	ON	انتخاب سومین سرعت توسط تنظیم پارامتر P01-11 (تعیین کردن سرعت توسط محتویات داخل D11، در این برنامه سرعت چرخش سروموتور توسط کاربر تعیین می‌شود که این مقدار در D11 قرار می‌گیرد.)

تنظیمات پارامتر سرودرایو AC سری ASDA دلتا در این مثال به شرح زیر است :

پارامتر	مقدار تنظیمی	توضیحات
P1-01	2	مد کنترل و جهت خروجی
P1-39	1500	سرعت هدف سروموتور: 1500RPM
P2-10	101	ترمینال ورودی دیجیتال 1 برای فعال کردن سروسیستم: (DI1)
P2-12	114	DI3: ترمینال ورودی برای SPD0
P2-13	115	DI4: ترمینال ورودی برای SPD1
P2-15 ~ P2-17	0	استفاده نشده
P2-18	101	اگر بعد از وصل برق هیچ خطایی رخ ندهد، DO1=ON می‌شود.
P2-19	102	اگر بعد از راه‌اندازی سرو هیچ خطایی رخ ندهد، DO2=ON می‌شود.
P2-20	104	وقتی که سرعت سرو به سرعت هدف رسید، DO3=ON می‌شود.
P3-00	1	تنظیم آدرس ارتباطات شبکه
P3-01	1	سرعت تبادل داده، 9600: BAUD RATE
P3-02	1	پروتکل MODBUS با مد ASCII و فرمت داده: (7, E, 1)
P3-05	2	فرمت ارتباطات سریال RS-485
P3-06	0	تابع ارتباطات سریال برای راه‌اندازی ورودی‌های دیجیتال

اگر بعد از تنظیمات پارامترها، سرو راه‌اندازی نشد، این به آن معنی است که بعضی از پارامترها اشتباه تنظیم شده‌اند. در این صورت باید پارامتر P02-08 را برابر 10 (تنظیمات پیش‌فرض کارخانه) تنظیم نموده و سپس پارامترها را مطابق با جدول بالا تنظیم نمایید.

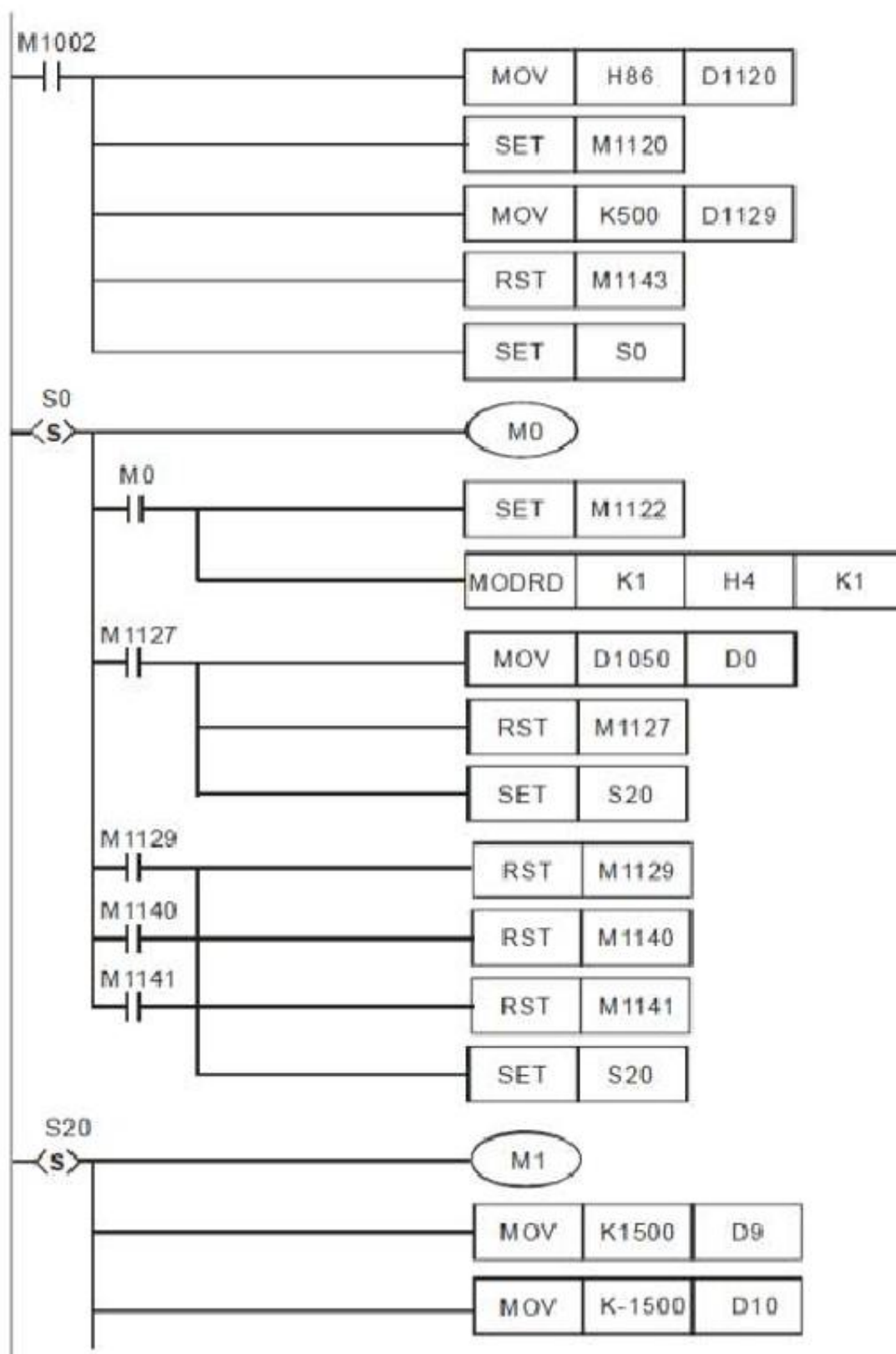
مراحل عملکرد در این مثال به شرح زیر است :

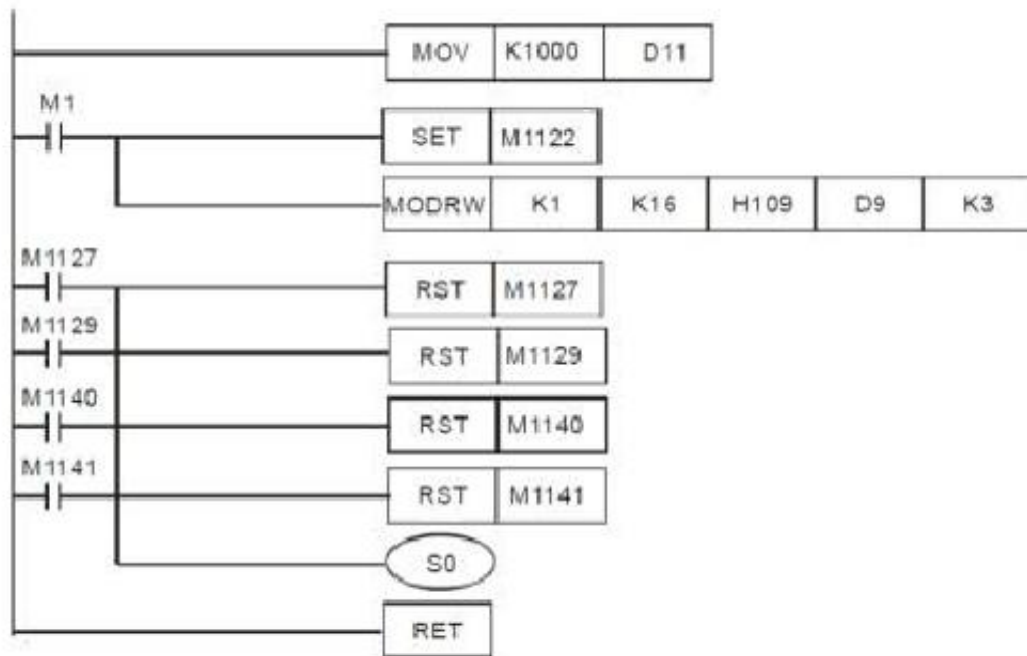


بعد از وصل مجدد برق سرودرایو، پارامترها را تنظیم نمایید. اگر هیچ خطایی رخ نداد، نمایشگر Power normal در خروجی D01 فعال می‌شود. وقتی نمایشگر Power normal فعال شود، با استفاده از DI1 سرودرایو را فعال SON نمایید. اگر هیچ خطایی رخ نداد، نمایشگر Start normal در خروجی DO2 فعال می‌شود. وقتی که SPD0 فعال شود، سرعت تنظیم شده در پارامتر P01-09 فعال می‌شود. وقتی که SPD1 فعال شود، سرعت تنظیم شده در پارامتر P01-10 فعال می‌شود. وقتی که هر ورودی SPD1 و SPD0 فعال شوند، سرعت تنظیم شده در پارامتر P01-11 فعال می‌شود. ابزارهای مورد استفاده در این مثال به شرح زیر است:

ابزار	تابع
M0	اجرای دستورالعمل MODRD با خواندن سرعت چرخش سروموتور
M1	اجرای دستورالعمل MODWR

برنامه کنترلی این مثال مطابق با شکل زیر است :





توضیح خط به خط برنامه به شرح زیر است:

- خط 1 ← تنظیم فرمت ارتباطات 1 , E , 7 , 9600
- خط 2 ← تنظیمات نگهداری و حفظ کردن ارتباطات شبکه
- خط 3 ← تنظیم زمان time-out برای تشخیص وقفه در دریافت اطلاعات: 500MS
- خط 4 ← تنظیم مد ارتباطات شبکه سریال: پروتکل MODBUS مد ASCII
- خط 5 ← ورود به مرحله S0
- خط 7 ← درخواست ارسال داده
- خط 8 ← خواندن سرعت سروموتور و ذخیره آن در D1050
- خط 9 ← ارسال محتویات D1050 به D0
- خط 10 ← ریست کردن M1127
- خط 11 ← ورود به مرحله S20
- خط 12 ← ریست کردن فلگ خاص M1129
- خط 13 ← ریست کردن فلگ خاص M1140
- خط 14 ← ریست کردن فلگ خاص M1141
- خط 15 ← ورود به مرحله S20
- خط 17 ← تنظیم نخستین سرعت برابر 1500RPM در جهت راستگرد
- خط 18 ← تنظیم دومین سرعت برابر 1500RPM در جهت چپگرد

خط 19 ← تنظیم سومین سرعت برابر 1000RPM در جهت راستگرد

خط 20 ← درخواست ارسال داده

خط 21 ← نوشتن محتویات D9 , D10 , D11 در داخل آدرس هگز H109 سرودرایو AC

خط 22 ← ریست کردن فلگ خاص M1127

خط 23 ← ریست کردن فلگ خاص M1129

خط 24 ← ریست کردن فلگ خاص M1140

خط 25 ← ریست کردن فلگ خاص M1141

خط 26 ← برگشت به مرحله S0

توضیحات برنامه به شرح زیر است :

تنظیمات اولیه پورت ارتباطات سریال RS-485 در سمت PLC و تنظیم فرمت ارتباطات سریال با پروتکل MODBUS برابر با مد ASCII و سرعت تبادل داده 9600 و همچنین 1 , E , 7 است. فرمت ارتباطات سریال RS-485 در سمت سرودرایو AC باید مانند فرمت ارتباطات سریال در سمت PLC باشد.

وقتی که برنامه وارد مرحله S0 می‌شود، M0=ON خواهد شد. با اجرای دستورالعمل MODRD K1 H4 K1 سرعت چرخش سروموتور خوانده شده و در حافظه D1050 ذخیره می‌شود. سپس برای نشان دادن سرعت چرخش در D0 دستورالعمل MOR D1050 D0 اجرا خواهد شد.

وقتی که برنامه وارد مرحله S20 می‌شود، M1=ON خواهد شد. با اجرا دستورالعمل MODWR K1 K16 H109 D9 K3 محتویات D9 , D10 , D11 در داخل H10A , H109 H10B برابر آدرس ارتباطات شبکه سرو و در پارامترهای سرو نوشته می‌شود.

تنظیمات اولیه در D11 برابر K1000 است. کاربر می‌تواند مقدار D11 را برای کاربردهای واقعی تنظیم نماید.

وقتی که PLC استارت شود، برنامه وارد مرحله S0 خواهد شد. سپس برنامه به مرحله S20 منتقل شده و در آخر مجدداً به مرحله S0 باز می‌گردد. توسط این پروسه عمل خواندن/نوشتن برای سرودرایو AC مکرراً اجرا خواهد شد.

فصل 8

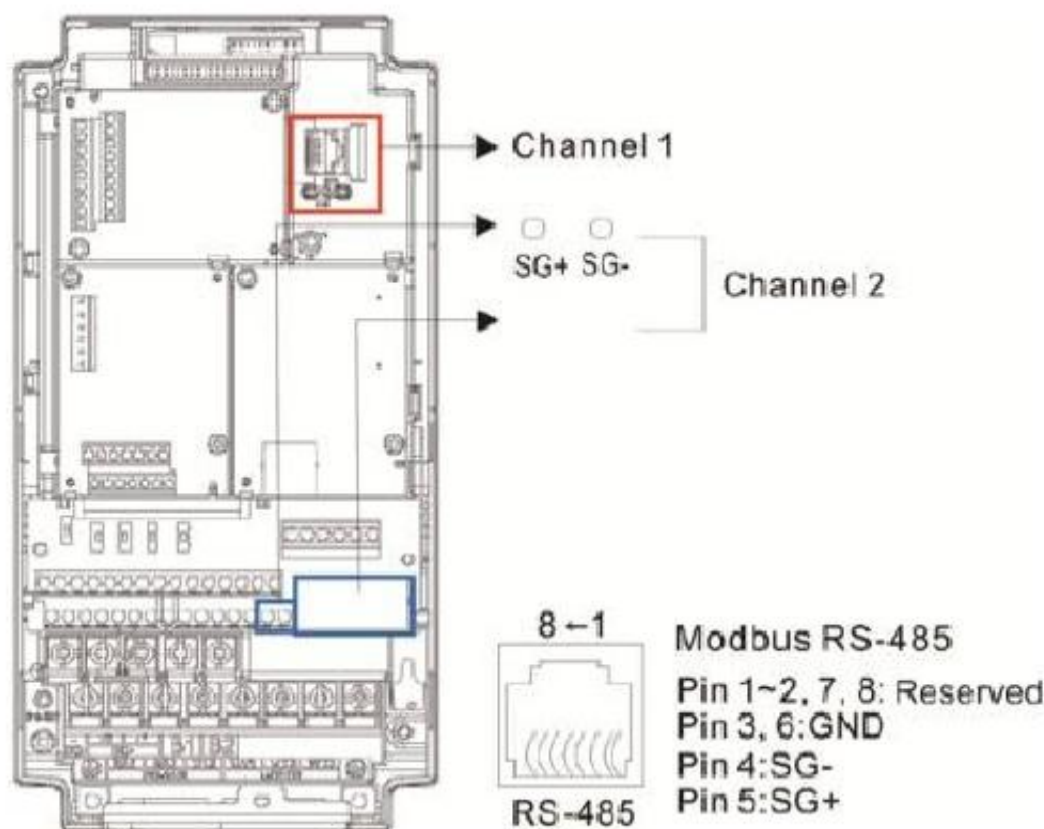
مثال های PLC داخلی اینورتر VFD-C2000



8- 1- ملاحظات قبل از استفاده از PLC داخلی اینورتر

فرمت ارتباطات PLC برای ارتباط با کامپیوتر 9600, N, 2, 7 بوده و شماره Node آن 2 است. توسط پارامتر Pr.09-35 می‌توان PLC node اینورتر را تغییر داد. دقت داشته باشید که این آدرس نباید با آدرس تنظیم شده در پارامتر Pr.09-00 یکسان باشد.

در اینورتر VFD-C2000 دو پورت ارتباطات سریال وجود دارد که از آنها می‌توان برای دانلود برنامه‌ی نوشته شده برای PLC داخلی به PLC استفاده کرد. لازم به ذکر است که فرمت ارتباطات کانال 1 ثابت بوده و قابل تغییر نیست. فرمت ارتباطات کانال 1 برابر با 19200, 8, N, 2, RTU



Client می‌تواند به صورت همزمان به دیتای کانورتر و PLC داخلی دسترسی داشته باشد. این قابلیت دسترسی توسط node تعیین هویت انجام می‌شود. بعنوان مثال، اگر Converter node برابر با 1 و Internal PLC node برابر با 2 باشد، سپس فرمان Client با مقدار 01(node) 0001(1 data item) 0400(address) 03(read) خواهد بود که این مقدار بیانگر این است

که باید دیتای پارامتر Pr.04-00 کانورتر خوانده شود و فرمان Client با مقدار 02(node) 0001(1 data item) 0400(address) 03(read) بیانگر این است که باید دیتای ورودی X0 از PLC داخلی خوانده شود.

وقتی که برنامه‌ای را از PLC بر روی کامپیوتر آپلود می‌کنید و یا برنامه‌ای را از کامپیوتر به PLC دانلود می‌کنید، برنامه‌ی PLC غیرفعال خواهد شد.

وقتی که از فرمان WPR برای نوشتن مقدار در پارامترها استفاده می‌کنید، مقادیر را می‌توان تا 10^9 بار اصلاح کرد. در غیر این صورت، یک خطای نوشتن مقدار در حافظه رخ خواهد داد. محاسبات اصلاح دیتا مبتنی بر تغییرات مقادیر ورودی می‌باشد. اگر مقدار وارد شده بدون تغییر باقی بماند، کانتر مربوط به افزایش مقدار اصلاح شده بدون تغییر باقی خواهد ماند. اما اگر مقدار ورودی با مقدار قبلی تفاوت داشته باشد، یک واحد به کانتر مربوط به افزایش مقدار اصلاح شده اضافه خواهد شد.

وقتی که پارامتر Pr.00-04 برابر با 28 تنظیم شده باشد، مقدار رجیستر D1043 که مربوط به PLC داخلی است، بر روی نمایشگر کی‌پد اینورتر نمایش داده خواهد شد.

نمایش مقدار 0~9999 بر روی نمایشگر کی‌پد دیجیتال KPC-CC01 	کی‌پد دیجیتال مدل KPC-CC01 توانایی نمایش مقدار 0~65535 را دارد
وقتی که مقدار نمایش داده شده بیشتر از 9999 شود، 	

در مد PLC Run و PLC Stop مقدار پارامتر Pr.00-02 را نمی‌توان برابر با 9 و 10 تنظیم کرد و همچنین، نمی‌توان به مقدار پیش‌فرض کارخانه بازگرداند.

وقتی که مقدار پارامتر Pr.00-02 برابر با 6 تنظیم شده باشد، PLC داخلی را می‌توان به پیش‌فرض کارخانه بازگرداند.

مطابق با تابعی که به ورودی دیجیتال (MI) اینورتر اختصاص داده شده است، می‌توان نوشتن مقدار به ورودی X از PLC را غیرفعال کرد.

وقتی که PLC فرامین عملکرد کانورتر را کنترل می‌کند، فرامین کنترلی وارد شده توسط PLC کنترل می‌شوند و تحت تأثیر تنظیمات پارامتر Pr.00-21 نخواهد بود.

وقتی که PLC فرامین فرکانس کانورتر (فرامین FREQ) را کنترل می‌کند، فرامین فرکانس وارد شده توسط PLC کنترل می‌شوند و تحت تأثیر تنظیمات پارامتر Pr.00-20 یا پیکربندی دستی ON/OFF نخواهد بود.

وقتی که PLC فرکانس کانورتر (فرامین TORQ) را کنترل می‌کند، فرامین گشتاور وارد شده توسط PLC کنترل می‌شوند و تحت تأثیر تنظیمات پارامتر Pr.11-33 یا پیکربندی دست ON/OFF نخواهند بود.

وقتی که PLC فرکانس کانورتر (فرامین POS) را کنترل می‌کند، فرامین موقعیت وارد شده توسط PLC کنترل می‌شوند و تحت تأثیر تنظیمات پارامتر Pr.11-40 یا پیکربندی دستی ON/OFF نخواهند بود.

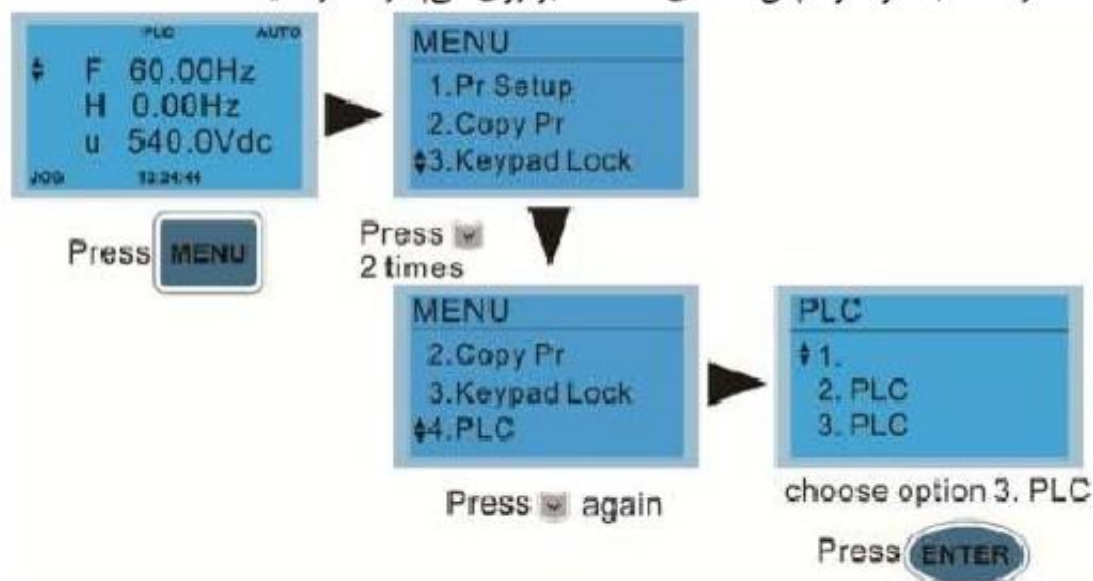
وقتی که عملکرد کانورتر توسط PLC کنترل می‌شود، اگر تنظیمات Stop کی‌پد معتبر باشد، اینورتر یک خطای FStp را تریگر خواهد کرد.

8-2 روشن کردن PLC

8-2-1 متصل شدن به کامپیوتر

برای اینکه توابع PLC شروع به کار نمایند، باید 4 مرحله‌ی زیر انجام شود:

مرحله‌ی 1 ← بعد از اینکه در کی‌پد دیجیتال KPC-CC01 اینورتر وارد Menu شدید آیتم PLC:4 را انتخاب نموده و سپس دکمه‌ی Enter بر روی کی‌پد را فشار دهید.



اگر از کی‌پد دیجیتال KPC-CE01 استفاده می‌کنید، باید از روش گفته شده در زیر استفاده نمایید:

بعد از اینکه تغذیه‌ی اینورتر را متصل کردید، دکمه‌ی **Enter** بر روی کی‌پد KPC-CE01 را یک بار فشار دهید تا به تابع **Screen** سوئیچ شوید، که در این صورت **PrSET** نمایش داده خواهد شد. با استفاده از دکمه‌های بالا و پایین کی‌پد وارد صفحه‌ی **PLC** شده و سپس دکمه‌ی **Enter** را برای وارد شدن به تنظیمات تابع **PLC** فشار دهید. سپس دکمه‌ی بالا را فشار دهید تا به **PLC2** سوئیچ شود. حال دکمه‌ی **Enter** را فشار دهید. حال صفحه‌ی **PLCn** نمایش داده خواهد شد که بیانگر این است که **PLC** داخلی برنامه ندارد که از این پیغام خطا می‌توان صرف‌نظر کرد. اگر **PLC** دارای یک برنامه‌ی ویرایش شده باشد، صفحه‌ی **End** نمایش داده خواهد شد و بعد از 1 یا 2 ثانیه صفحه به **PLC2** بر می‌گردد. وقتی که برنامه‌ای به **PLC** داخلی اینورتر دالود نشده باشد، حتی اگر یک پیغام خطای **PLC** ظاهر شود، برنامه به صورت نرمال کار کرده و **Run** خواهد ماند.



مرحله‌ی 2 ← واسطه ارتباطات RJ-45 اینورتر را به یک واسطه RS-485 کامپیوتر متصل نمایید.



مرحله‌ی 3 ← برای استفاده از **PLC** باید طبق جدول زیر عمل کنید:

<div style="border: 1px solid black; padding: 5px; background-color: #00a0e3; color: white; text-align: center;"> PLC 1. Disable 2. PLC Run 3. PLC Stop </div>	<p>توابع PLC در شکل بالا نشان داده شده است. با انتخاب آیتم 2 توابع PLC اجرا خواهند شد.</p>
<p>1: No Function (Disable) 2: Enable PLC (PLC Run) 3: Stop PLC Function (PLC Stop)</p>	<p>نحوه‌ی نمایش توابع PLC بر روی کی‌پد دیجیتال KPC-CE01</p>
<p>PLC 0 ← توابع PLC اجرا نمی‌شوند. PLC 1 ← PLC در حال کار می‌باشد (PLC Run). PLC 2 ← PLC استوپ است (PLC Stop).</p>	

مرحله‌ی 4 ← وقتی که تابع اختصاص داده شده به ترمینال‌های ورودی قابل برنامه‌ریزی خارجی (MI1~MI8) برابر با (51) PLC Mode select bit 0 یا (52) PLC Mode select bit 1 (در حالت عادی باز) یا N.C. (در حالت عادی بسته) بود، به اجبار اینورتر به PLC mode سوئیچ شده و عملکرد کی‌پد بی‌اثر خواهد بود. در جدول زیر، عملکرد بیت‌های گفته شده برای PLC Mode آورده شده است.

PLC mode		PLC Mode select bit1(52)	PLC Mode select bit0 (51)
Using KPC-CC01	Using KPC-CE01		
Disable	PLC 0	OFF	OFF
PLC Run	PLC 1	OFF	ON
PLC Stop	PLC 2	ON	OFF
Maintain previous state	Maintain previous state	ON	ON

نحوه‌ی استفاده از کی‌پد دیجیتال KPC-CE01 برای اجرای توابع PLC به شرح زیر است:
وقتی که صفحه‌ی PLC به صفحه‌ی PLC1 سوئیچ شود، PLC داخلی به وضعیت Run رفته و Start/Stop کردن برنامه‌ی PLC را می‌توان با استفاده از نرم‌افزار WPLSoft یا SPSoft کنترل کرد.

وقتی که صفحه‌ی PLC به صفحه‌ی PLC2 سوئیچ شود، PLC به وضعیت استوپ رفته و Start/Stop کردن برنامه را می‌توان با استفاده از نرم‌افزار WPLSoft یا SPSoft کنترل کرد.
روش کنترل وضعیت PLC داخلی اینورتر توسط ترمینال‌های ورودی دیجیتال در جدول بالا نشان داده شده است.

نکته:

وقتی که از ترمینال‌های ورودی خروجی خارجی (FWD, REV, MI1~MI8, MI10~MI15, Relay1, Relay2, RY10~RY1, MO1, MO2, MO10, MO11) در برنامه‌ی PLC استفاده شود، از ترمینال‌های ورودی خروجی فقط در PLC داخلی استفاده خواهد شد. بعنوان مثال، وقتی که برنامه‌ی PLC در هنگام عملکرد PLC (عملکرد PLC شامل PLC1 (وضعیت Run) و PLC2 (وضعیت stop) می‌باشد) ترمینال خروجی Y0 را کنترل می‌کند، رله‌ی ترمینال خروجی (RA/RB/RC) مطابق با برنامه‌ی PLC عمل خواهد کرد. در این هنگام، تنظیمات ترمینال‌های ورودی خروجی قابل برنامه‌ریزی برای اینورتر بی‌اثر خواهد بود. زیرا توابع ترمینال‌ها برای PLC رزرو شده است. وضعیت ترمینال‌های ورودی خروجی دیجیتال و آنالوگ استفاده شده توسط PLC را می‌توان توسط پارامترهای Pr.02-52, Pr.02-53, Pr.03-30 بررسی کرد.

رجیستر خاص D1040 در PLC به خروجی آنالوگ AFM1 اینورتر اختصاص داده شد. همچنین، رجیستر خاص D1045 در PLC داخلی به خروجی آنالوگ AFM2 اینورتر اختصاص داده شده است. بنابراین، برای کنترل این دو خروجی آنالوگ اینورتر می‌توان از رجیسترهای خاص نام برده شده در برنامه‌نویسی PLC استفاده کرد. توسط پارامتر Pr.03-30 می‌توان وضعیت ترمینال خروجی آنالوگ تابع PLC را مانیتور کرد. بیت صفر مطابق با وضعیت عملکرد ترمینال AFM1 بوده و بیت یک مطابق با وضعیت عملکرد ترمینال AFM2 می‌باشد.

8-2-2 ابزارهای ورودی خروجی اینورتر

ابزارهای مورد استفاده در اینورتر مطابق با جدول زیر است:

Serial No.	X0	X1	X2	X3	X4	X5	X6	X7	X10	X11	X12	X13	X14	X15	X16	X17
1	FWD	REV	MI1	MI2	MI3	MI4	MI5	MI6	MI7	MI8						
2											MI10	MI11	MI12	MI13	MI14	MI15
3											MI10	MI11	MI12	MI13		

شماره سریال 1 بیانگر ورودی‌های خود اینورتر می‌باشد.

شماره سریال 2 بیانگر ورودی‌های کارت توسعه‌ی (D1022=4) EMC-D611A است.

شماره سریال 3 بیانگر ورودی‌های کارت توسعه‌ی (D1022=5) EMC-D42A است.

ابزارهای خروجی مورد استفاده در اینورتر مطابق با جدول زیر است:

Serial No.	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y10	Y11	Y12	Y13	Y14	Y15	Y16	Y17
1	RY1	RY2		MO1	MO2											
2						MO10	MO11									
3						RY10	RY11	RY12	RY13	RY14	RY15					

شماره سریال 1 بیانگر خروجی‌های خود اینورتر می‌باشد.
 شماره سریال 2 بیانگر خروجی‌های کارت توسعه‌ی EMC-D42A (D1022=5) است.
 شماره سریال 3 بیانگر خروجی‌های کارت توسعه‌ی EMC-R6AA (D1022=6) است.

8- 3 مثال کاربردی برای ماژول CMC-MOD01

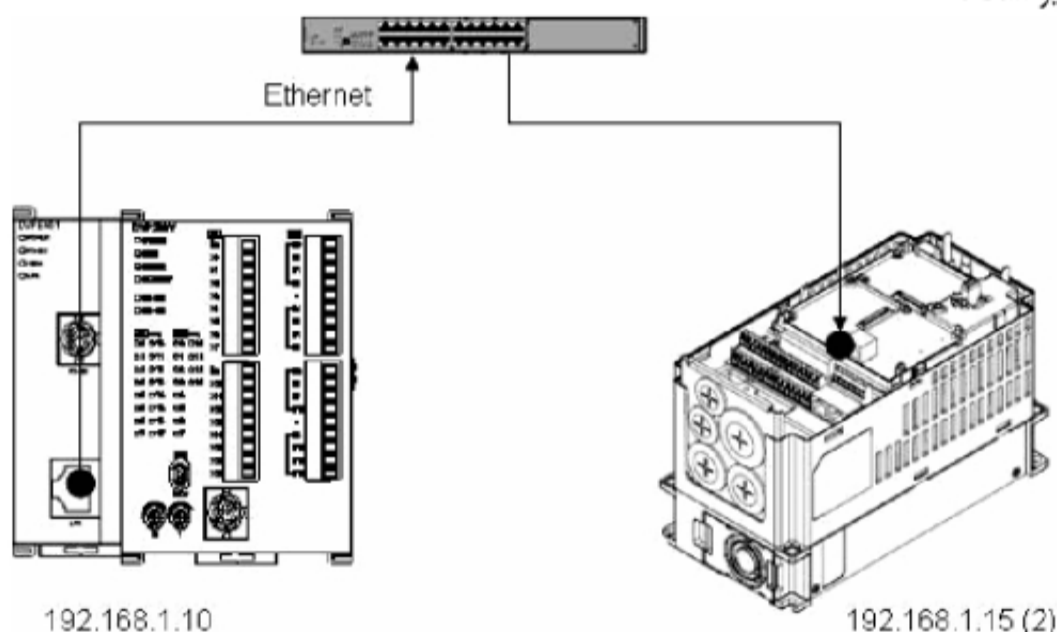
می‌خواهیم فرکانس اینورتر VFD-C2000 را توسط شبکه از طریق PLC سری DVP-SV دلتا و ماژول شبکه‌ی DVPEN01 دلتا کنترل نماییم. برای این کار باید ماژول شبکه‌ی DVPEN01 به سمت چپ PLC سری DVP-SV متصل شود. در این مثال، قصد داریم فرکانس 60 هرتز را داخل پارامتری با آدرس H' 2001 توسط شبکه‌ی اترنت از طریق PLC بنویسیم.

Parameter Address: H' 2001, 60Hz, H' 1770

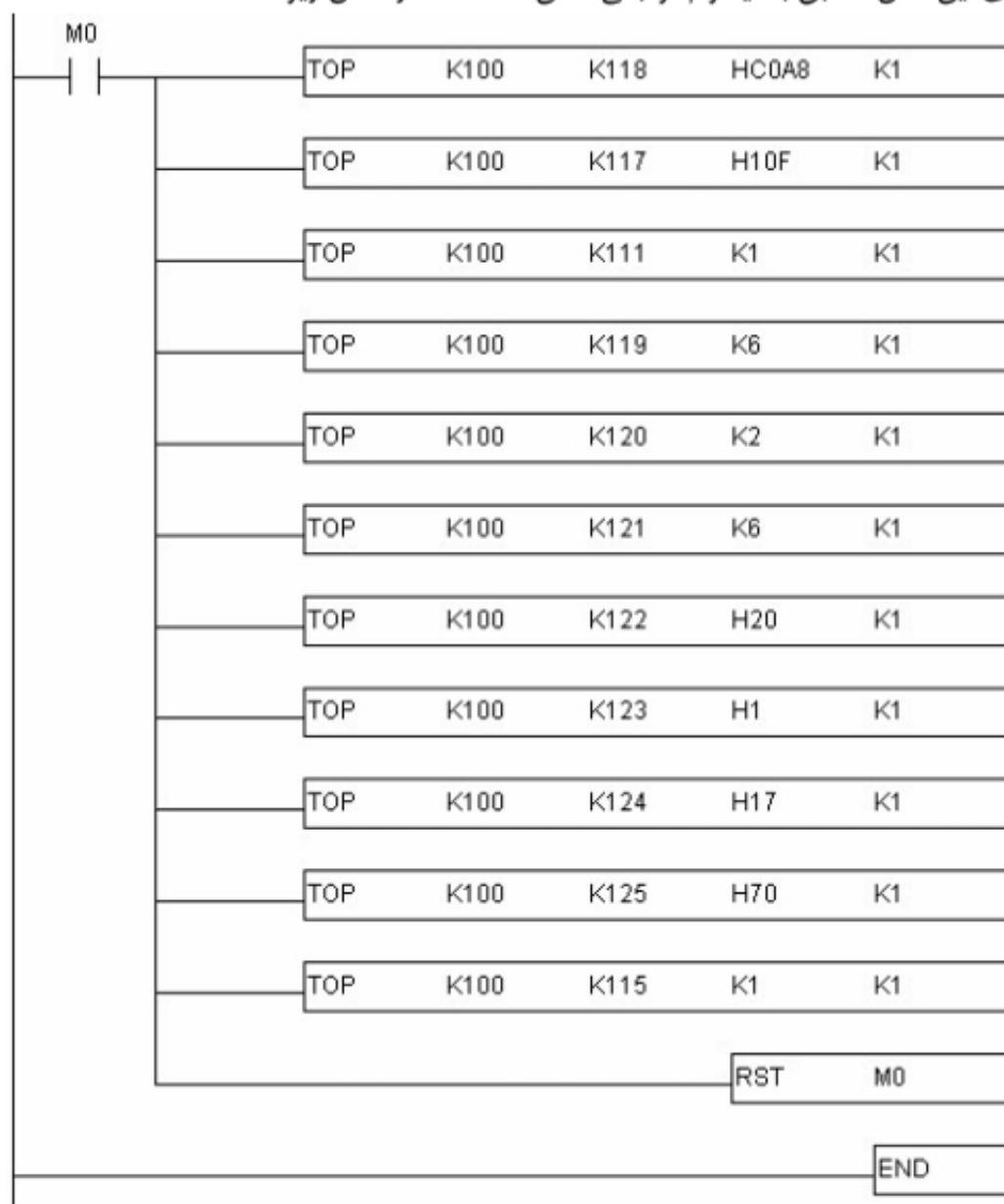
در نخستین مرحله باید IP آدرس ماژول DVPEN01-SL را برابر با 192.168.1.10 تنظیم نمایید. سپس باید IP آدرس کارت شبکه‌ی اترنت (CMC-MOD01) اینورتر VFD-C2000 را برابر با 192.168.1.20 تنظیم کرده و شماره ایستگاه اینورتر (Station NO.) را برابر با 2 تنظیم نمایید.

بعد از اجرای برنامه، PLC مقدار 06020620011770 را به شبکه‌ی MODBUS TCP/IP ارسال خواهد کرد.

نحوه‌ی اتصال اینورتر VFD-C2000 و PLC مجهز به کارت DVPEN01-SL مطابق با شکل زیر است:



برنامه‌ی این مثال مطابق با دیاگرام نردبانی نشان داده شده در شکل زیر است :

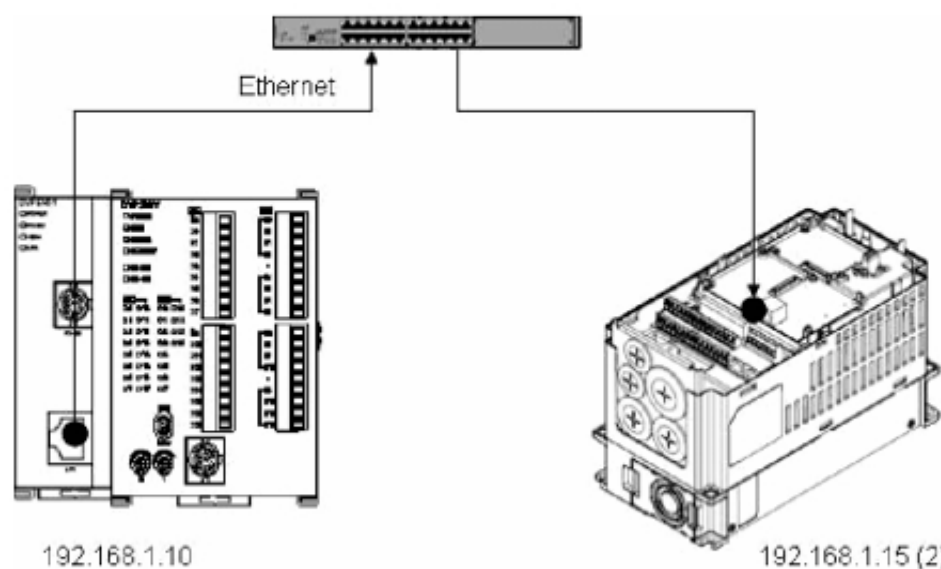


همانطور که گفته شد، PLC سری DVP 28 SV اطلاعات را به کارت شبکه‌ی اترنت سری DVPEN01-SL ارسال کرده و این کارت مقدار 06 02 06 20 01 17 70 را از طریق فرمان MODBUS TCP به اینورتر سری VFD-C2000 ارسال می‌کند. ابتدا IP آدرس 192.168.1.15 به فرمت هگز H' C0A8, H' 010F تبدیل شده و سپس آنها را به پارامترهای K117, K118 در ماژول DVPEN01-SL می‌نویسید. طول فرمان "06" به پارامتر K119 در ماژول DVPEN01-SL ارسال خواهد شد.

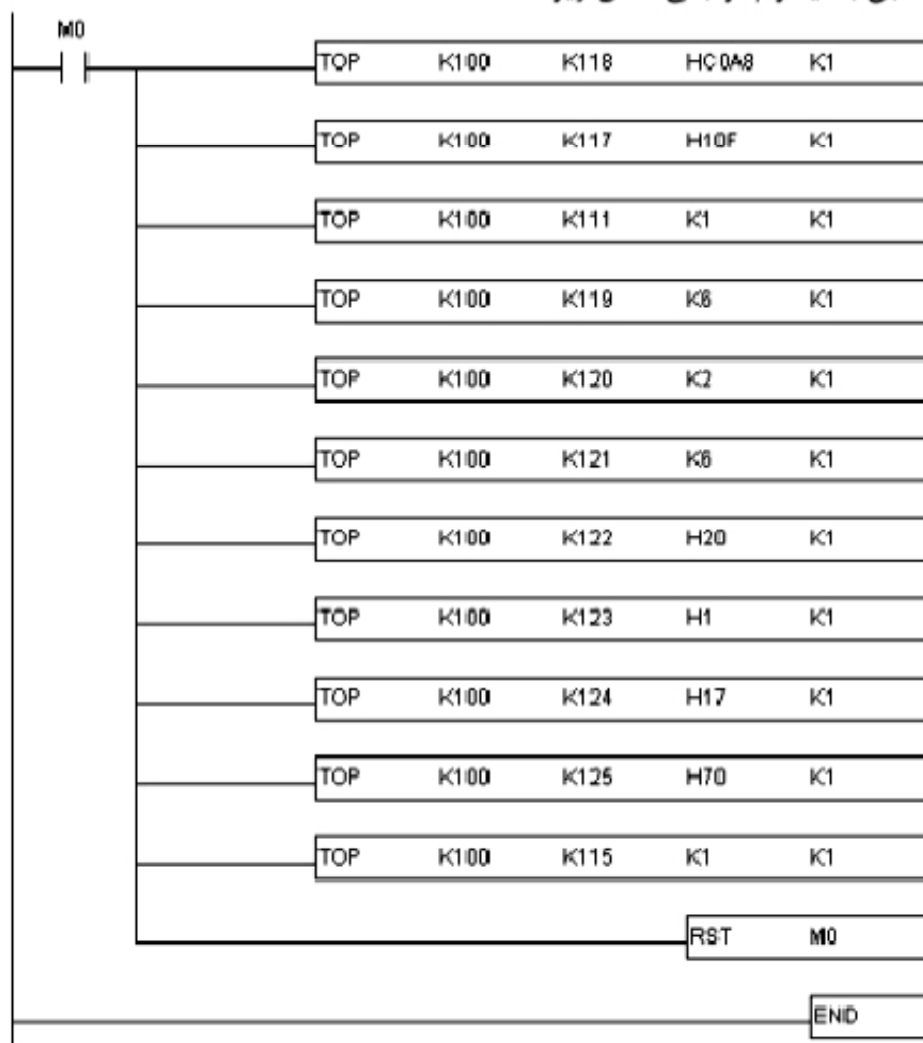
آدرس ایستگاه "02" به پارامتر K120 در ماژول DVPEN01-SL ارسال خواهد شد.
 کد تابع "06" به پارامتر K121 در ماژول DVPEN01-SL ارسال خواهد شد.
 آدرس هگز پارامتر "H' 2001" اینورتر VFD-C2000 دلتا به پارامترهای K122, K123 در ماژول DVPEN01-SL ارسال خواهد شد.
 آدرس هگز پارامتر "H' 1770" اینورتر VFD-C2000 دلتا به پارامترهای K124, K125 در ماژول DVPEN01-SL ارسال خواهد شد.
 برای فعال کردن عمل ارسال دیتا توسط ماژول DVPEN01-SL مقدار "1" داخل پارامتر K115 نوشته می‌شود.
 برنامه‌ی نوشته شده را به PLC دالود کرده و سپس فلگ M0 را برای ارسال مقدار 060206200 به دستگاهی با IP آدرس 192.168.1.15 ارسال کنید.
 برای اطلاع از عملکرد و جزئیات ماژول شبکه‌ی اترنت با پروتکل MODBUS TCP/IP به کتاب مرجع کامل شبکه‌ی مدباس با استفاده از PLC های دلتا چاپ شده توسط همین انتشارات مراجعه نمایید.

8- مثال کاربردی از ماژول CMC-EIP01

می‌خواهیم فرکانس اینورتر VFD-C2000 دلتا را توسط شبکه MODBUS TCP از طریق PLC سری DVP-SV دلتا و ماژول شبکه‌ی DVPEN01-SL دلتا کنترل نماییم. برای این کار باید ماژول شبکه‌ی DVPEN01-SL را به سمت چپ PLC سری DVP-SV متصل نماییم. در این مثال، قصد داریم فرکانس 60 هرتز را داخل پارامتری با آدرس H' 2001 توسط شبکه‌ی MODBUS TCP از طریق PLC بنویسیم (Parameter Address: H' 2001, 60Hz=H' 1770).
 در نخستین مرحله باید IP آدرس ماژول DVPEN01-SL را برابر با 192.168.1.10 تنظیم کنیم. سپس باید IP آدرس کارت شبکه‌ی اترنت CMC-EIP01 را برابر با 192.168.1.15 تنظیم کرده و شماره ایستگاه اینورتر را برابر با 2 تنظیم نماییم.
 بعد از اجرای برنامه، PLC DVP-SV مقدار 06020620011770 را توسط ماژول DVPEN01-SL از طریق شبکه‌ی MODBUS TCP به اینورتر مدل VFD-C2000 ارسال خواهد کرد.
 نحوه‌ی اتصال اینورتر VFD-C2000 و PLC DVP-SV که مجهز به کارت شبکه‌ی DVPEN01-SL است، مطابق با شکل زیر می‌باشد:



برنامه‌ی این مثال مطابق با دیاگرام نردبانی شکل زیر است :



ابتدا IP آدرس ماژول CMC-EIP01 توسط ماژول DVPEN01-SL داخل رجیستر CR#17(1.5=H'010E) و رجیستر CR#18(192.168.=H'C0A8) از ماژول CMC-EIP01 نوشته می‌شود.

برای استفاده از مد 8 بیتی باید مقدار رجیستر CR#111 را برابر با 1 کرد.

طول فرمان داخل رجیستر CR#19 نوشته می‌شود.

فرمان مدباس با مقدار 06020620011770 در داخل رجیسترهای CR#120~CR#125 نوشته می‌شود. دقت داشته باشید که رجیستر CR#120 برای شماره ایستگاه شبکه مدباس رزرو شده است.

مقدار 1 در رجیستر CR#115 برای شروع اجرای فرمان MODBUS TCP نوشته می‌شود.

برنامه‌ی نوشته شده برای DVP-SV PLC در نرم‌افزار WPLSoft را به PLC دانلود نمایید. با فعال شدن فلگ M0 مقدار 06020620011770 که یک فرمان مدباس است به اینورتر VFD-C2000 دلتا ارسال خواهد شد. این فرمان فرکانس اینورتر را برابر با 60Hz تنظیم خواهد کرد.

برای اطلاع از نحوه‌ی عملکرد و جزئیات ماژول شبکه‌ی ات‌رنت DVPEN01-SL با پروتکل MODBUS TCP/IP به کتاب مرجع کامل شبکه‌ی مدباس با استفاده از PLC های دلتا که توسط همین انتشارات چاپ شده مراجعه نمایید.

8- 5 مثال کاربردی از ماژول CMC-PD01

برای اتصال اینورتر سری VFD-C2000 به شبکه‌ی Profibus DP ابتدا باید پارامترهای مربوط به پروفیباس را در اینورتر تنظیم کرد. پارامترهای وابسته به شبکه‌ی Profibus DP را باید مطابق با جدول زیر برای اینورتر VFD-C2000 تنظیم نمایید. دقت داشته باشید که دستگاه Profibus DP Master تنها بعد از تنظیم پارامترهای ارتباطات توانایی خواندن و نوشتن مقدار در پارامترهای اینورتر VFD-C2000 را دارد.

پارامتر	نام تابع	مقدار تنظیمی	توضیحات
Pr.00-20	منبع فرمان فرکانس اینورتر	8	با تنظیم مقدار 8 در این پارامتر، فرکانس فرمان را می‌توان توسط کارت ارتباطات کنترل کرد.
Pr.00-21	منبع فرمان عملکرد اینورتر	5	با تنظیم مقدار 5 در این پارامتر، عملکرد اینورتر را می‌توان توسط کارت ارتباطات کنترل کرد.
Pr.09-30	روش دیکد کردن برای پورت ارتباطات	0/1	0: روش دیکد کردن قدیمی برای اینورتر (20xx) 1: روش دیکد کردن جدید برای اینورتر (60xx)
Pr.09-70	آدرس کارت ارتباطات	-	آدرس کارت شبکه‌ی Profibus DP برای اینورتر VFD-C2000 باید یک آدرس منحصر بفرد در کل شبکه باشد.

توجه داشته باشید که آدرس تعریف شده در پارامتر Pr.09-70 باید در کل شبکه منحصر بفرد بوده و در رنج 1 الی 127 قرار داشته باشد. در هنگام پیکربندی GSD فایل این اینورتر در نرم افزار Master PLC مانند PLC زمینس باید آدرس GSD فایل اینورتر با آدرس تنظیم شده در پارامتر Pr.09-70 یکسان باشد. بعد از تغییر پارامتر Pr.09-70 باید یک بار تغذیه‌ی اینورتر را قطع و وصل کنید تا این آدرس معتبر شود. برای تعیین فاصله (مسافت) تبادل داده در شبکه باید بر روی سرعت تبادل دیتا (Baud Rate) کار کرد. محدوده‌ی Baud Rate برای شبکه‌ی Profibus DP بین 9.6Kbps~12Mbps است. طول کابل برای تبادل دیتا بین دستگاه‌های متصل شده به شبکه وابسته به سرعت تبادل دیتای تنظیمی در شبکه است. طول این کابل بین 100m~1200m است. در جدول زیر سرعت‌های تبادل دیتا (Baud Rate) پشتیبانی شده توسط کارت CMC-PD01 و همچنین طول کابل مورد استفاده برای هر سرعت تنظیمی آورده شده است:

Baud rate (bps)	9.6k	19.2k	93.75k	187.5k	500k	1.5M	3M	6M	12M
Distance (m)	1,200	1,200	1,200	1,000	400	200	100	100	100

برای خواندن مقدار فرکانس خروجی جاری از اینورتر VFD-C2000 دلتا توسط شبکه‌ی پروفیباس، ابتدا مقدار کد درخواست (Request ID) را باید برابر با 1 (خواندن پارامتر) تنظیم نمایید. آدرس فرکانس خروجی جاری اینورتر برابر با 2103 به فرمت هگز و 8451 به فرمت دسیمال است. به دلیل اینکه محدوده‌ی مقدار PNU بین 1 الی 1999 بوده، شماره‌ی صفحه‌ی PNU برابر با 4 تنظیم شده و مقدار IND برابر با 2000 می‌باشد، بنابراین داریم:

$$PNU = 8451 - 4 \times 2000 = 451 \text{ (01C3 (hex))}$$

فرمت دیتا برای PKW به شکل زیر است:

Master → VFD-C2000 : 11C3 2000 0000 0000

VFD-C2000 → Master : 11C3 2000 0000 1070

درخواست دیتا توسط دستگاه Master برای هر یک از Word های PKW مطابق با جدول زیر است:

Parameters in PKW	Parameters in PKW (hex)
Word 1 (PKE)	11C3
Word 2 (IND)	2000
Word 3 (PWE1)	0000
Word 4 (PWE2)	0000

مقدار دیتای پاسخ توسط دستگاه Slave (اینورتر) برای هر یک از Word های PKW مطابق با جدول زیر است:

Parameters in PKW	Parameters in PKW (hex)
Word 1 (PKE)	11C3
Word 2 (IND)	2000
Word 3 (PWE1)	0000

برای نوشتن مقدار 2 داخل پارامتری با آدرس 2000 هگز در اینورتر VFD-C2000 دلتا توسط دستگاه Master در شبکه‌ی پروفیباس، ابتدا باید کد درخواست (Request ID) را برابر با 2 (اصلاح مقدار پارامتر) تنظیم نمایید. آدرس Control Word (کلمه کنترلی) برای اینورتر برابر با 2000 به فرمت هگز و 8192 به فرمت دسیمال است. شماره صفحه‌ی PNU برابر با 4، مقدار IND برابر با 2000 تنظیم می‌شود. بنابراین مقدار PNU برابر است با:

$$PNU = 8192 - 4 \times 2000 = 192 \text{ (0C0 (hex))}$$

فرمت دیتا برای PKW به شکل زیر است:

Master → VFD-C2000 : 20C0 2000 0000 0002

VFD-C2000 → Master : 10C0 2000 0000 0002

مقدار درخواست دیتا توسط دستگاه Master برای هر یک از Word های PKW مطابق با جدول زیر است:

Parameters in PKW	Parameters in PKW (hex)
Word 1 (PKE)	20C0
Word 2 (IND)	2000
Word 3 (PWE1)	0000
Word 4 (PWE2)	0002

مقدار دیتای پاسخ توسط دستگاه Slave برای هر یک از Word های PKW مطابق با جدول زیر است:

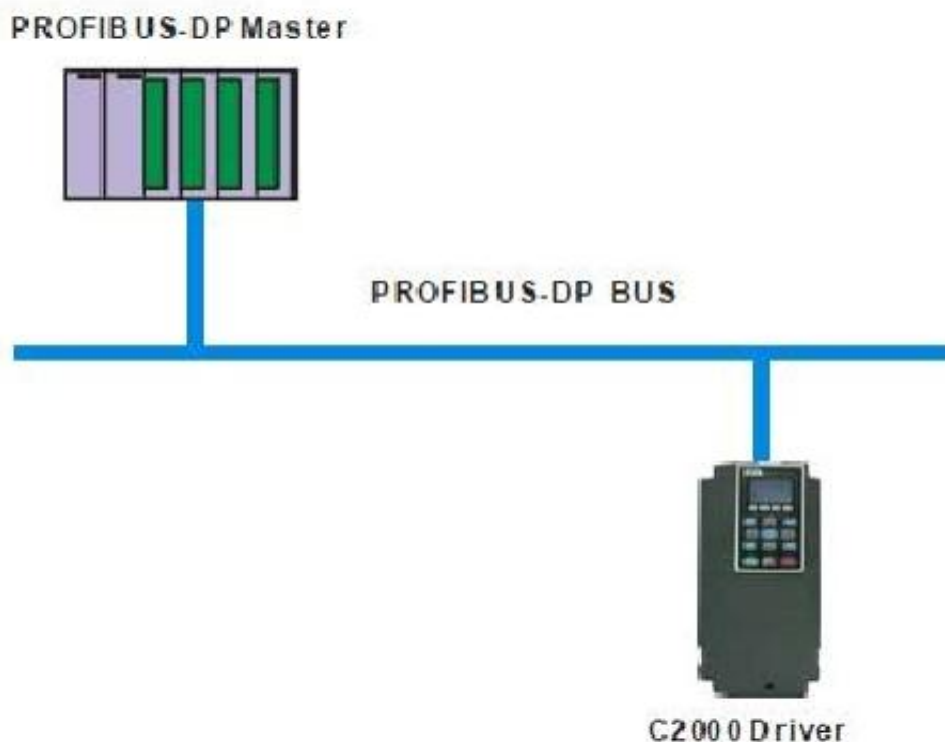
Parameters in PKW	Parameters in PKW (hex)
Word 1 (PKE)	10C0
Word 2 (IND)	2000
Word 3 (PWE1)	0000
Word 4 (PWE2)	0002

GSD File یک فایل متنی است که می‌توان از آن برای شناسایی دستگاه Master یا Slave در شبکه‌ی Profibus DP در نرم‌افزار برنامه‌نویسی دستگاه Master استفاده کرد. یک GSD فایل معمولاً شامل اطلاعات تهیه کننده، نرخ تبادل داده‌ی (Baud rate) پشتیبانی شده و پیغام‌های ورودی/خروجی کاربردی است. وقتی که از اینورتر VFD-C2000 بعنوان یک دستگاه Profibus DP Slave استفاده می‌کنید، ابتدا باید GSD File مربوط به اینورتر VFD-C2000 را جهت

پیکربندی نرم‌افزاری برای دستگاه Profibus DP Master به نرم‌افزار برنامه‌نویسی دستگاه Master وارد کنید. برای تهیه‌ی GSD File مربوط به اینورتر VFD-C2000 می‌توانید به وب سایت دلتا با آدرس <http://www.delta.com.tw/> مراجعه کرده و آن را دانلود نمایید. در این مثال، می‌خواهیم بین یک PLC سری S7-300 ساخت شرکت زیمنس و اینورتر VFD-C2000 دلتا توسط شبکه‌ی Profibus DP تبادل دیتا نماییم.

برای اتصال اینورتر VFD-C2000 به شبکه‌ی Profibus DP باید مراحل زیر را طی کنید:

1. در اینجا PLC S7-300 را بعنوان دستگاه Profibus DP Master و اینورتر VFD-C2000 را بعنوان دستگاه Profibus DP Slave در نظر می‌گیریم.



2. پارامترهای اینورتر VFD-C2000 را باید مطابق با مقادیر گفته شده در زیر تنظیم کنید.

Pr.00-20 = 8

Pr.00-21 = 5

Pr.09-30 = 0

Pr.09-70 = 8

مقدار پارامتر Pr.09-70 آدرس اینورتر VFD-C2000 بر روی شبکه‌ی Profibus DP است. وقتی که مقدار پارامتر Pr.00-70 را تغییر دادید، تغذیه اینورتر را باید یکبار قطع و وصل کنید تا آدرس جدید معتبر شود.

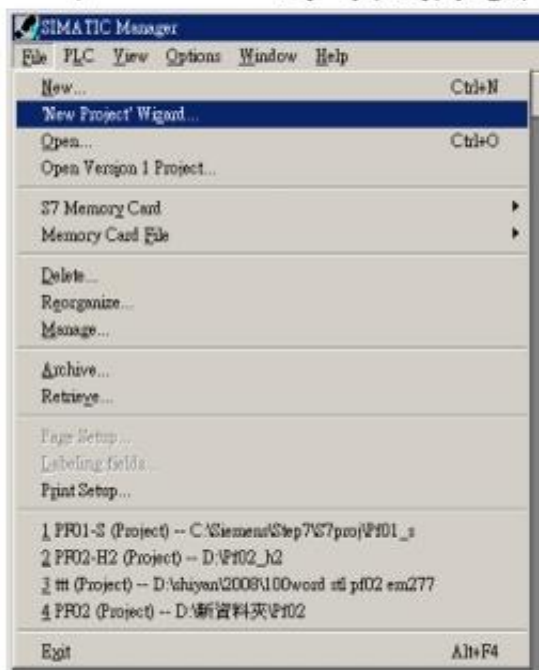
3. کارت CMC-PD01 را به اینورتر VFD-C2000 متصل نموده و سپس سیم‌بندی کابل شبکه را بدرستی انجام دهید.

پیکربندی نرم‌افزاری اینورتر VFD-C2000 در شبکه‌ی پروفیباس به شرح زیر است:

1. نرم‌افزار SIMATIC Manager را باز نمایید. این نرم‌افزار، نرم‌افزار برنامه‌نویسی و پیکربندی PLC های سری S7-300 و S7-400 شرکت زیمنس است.



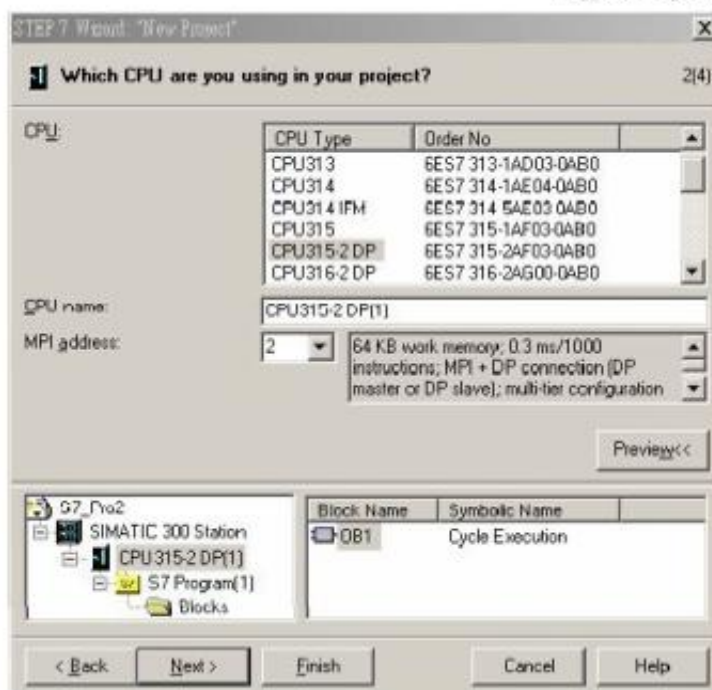
2. بر روی منوی File و سپس بر روی زیر منوی New Project Wizard کلیک نمایید.



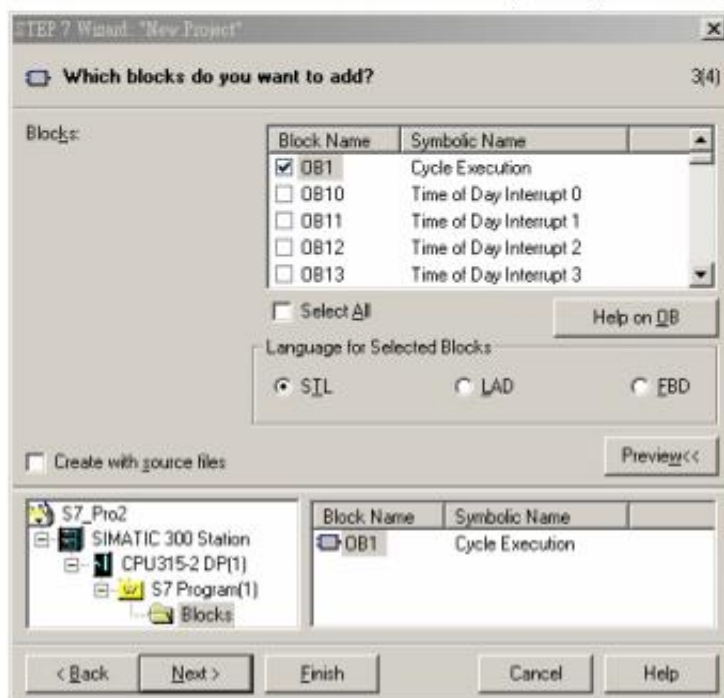
3. در پنجره‌ی Wizard باز شده بر روی دکمه‌ی Next کلیک کنید.



4. در کادر CPU باید مدل CPU ی S7-300 را برابر با CPU مورد استفاده خود تنظیم نمایید. در این مثال، CPU ی مورد استفاده‌ی ما مدل CPU 315-2 DP است. سپس بعد از انتخاب مدل CPU بر روی دکمه‌ی Next کلیک نمایید.



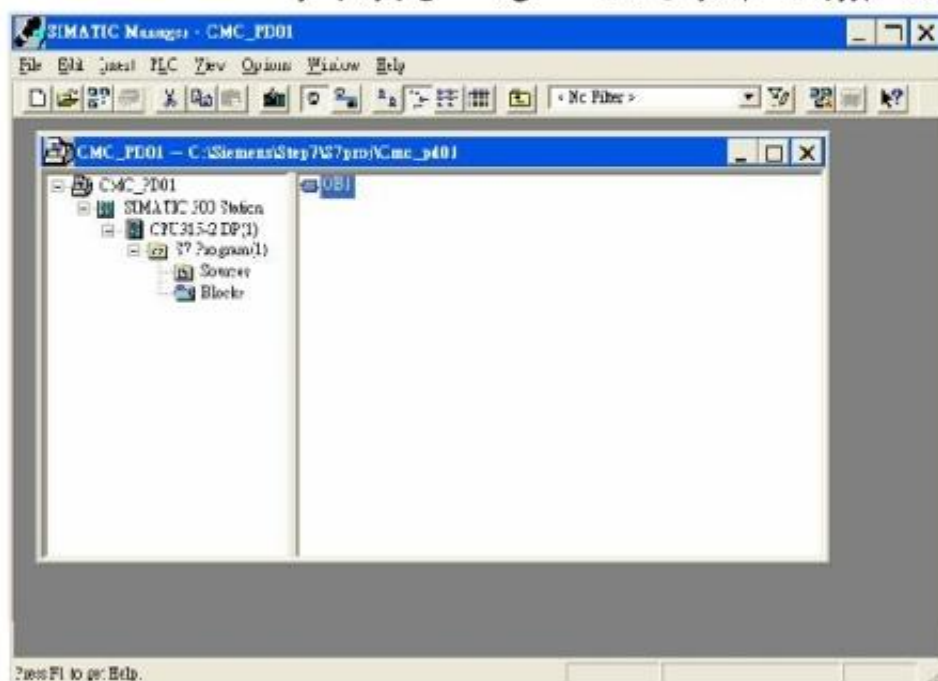
5. در کادر Blocks باید بلوک برنامه‌نویسی و زبان برنامه‌نویسی مورد نظر خود را انتخاب نمایید. سپس بر روی دکمه‌ی Next کلیک کنید.



6. در کادر Project Name باید نام پروژه را وارد کرده و در نهایت، بر روی دکمه‌ی Next کلیک نمایید.

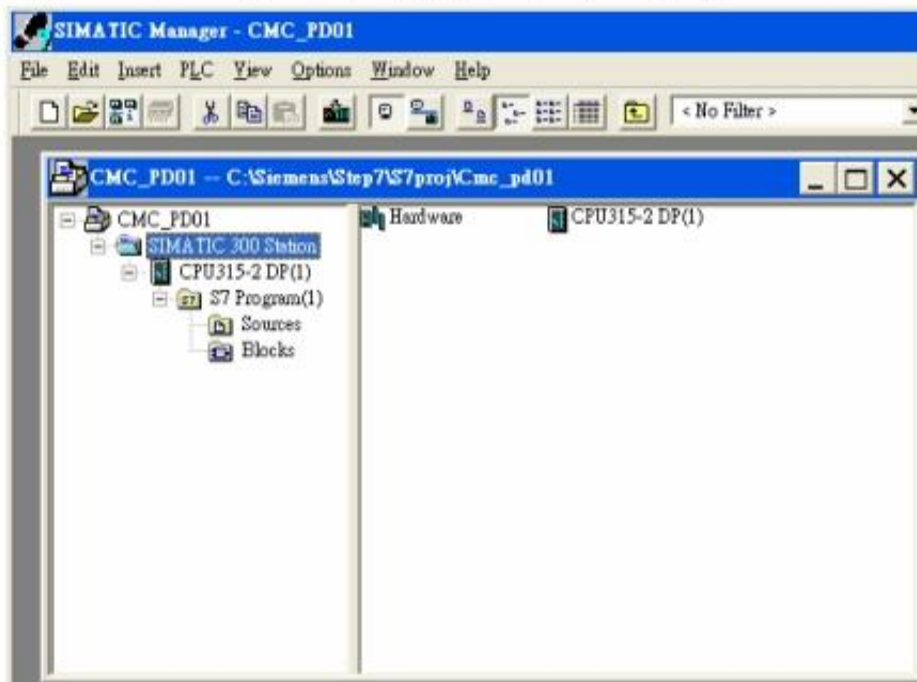


7. بعد از ایجاد پروژه یک پنجره‌ی جدید مطابق با شکل زیر باز خواهد شد.

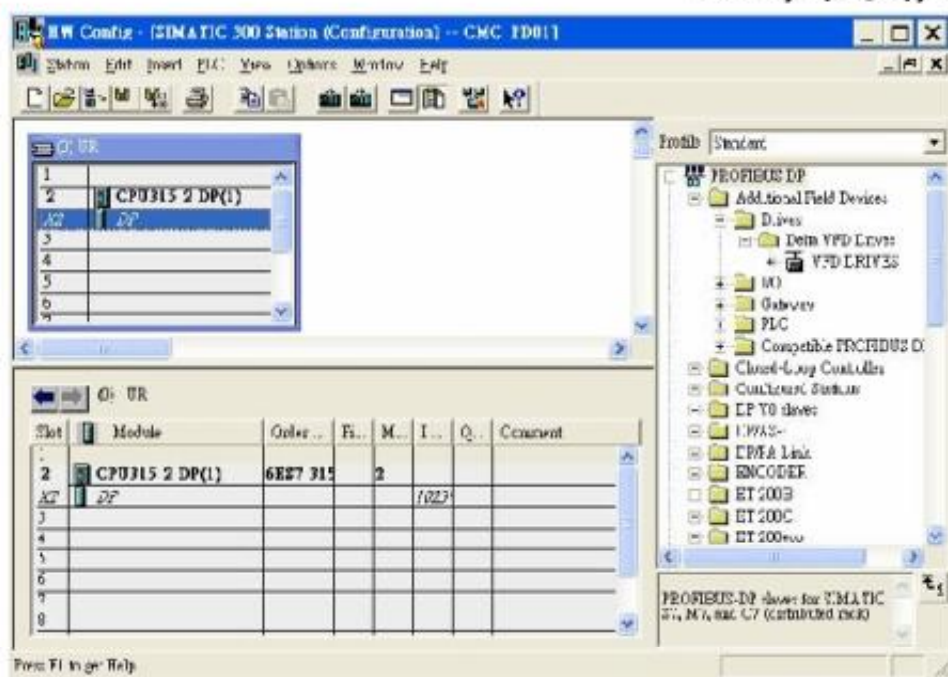


اضافه کردن باس Profibus DP به PLC

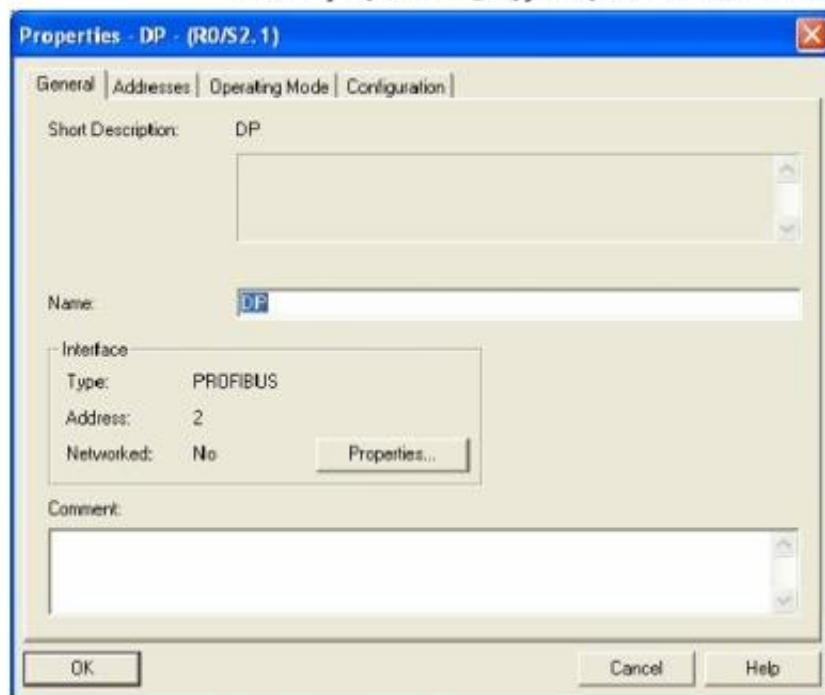
1. در پروژه‌ی ایجاد شده بر روی آیت **SIMATIC 300 Station** کلیک کنید. در پنجره‌ی باز شده بر روی آیکن **Hardware** دابل کلیک کنید تا یک پنجره‌ی جدید به نام **HW-Config** باز شود.



2. در پنجره‌ی HW-Config در ستون سمت چپ بر روی DP دابل کلیک نمایید. در این حالت، یک کادر محاوره‌ای باز خواهد شد.



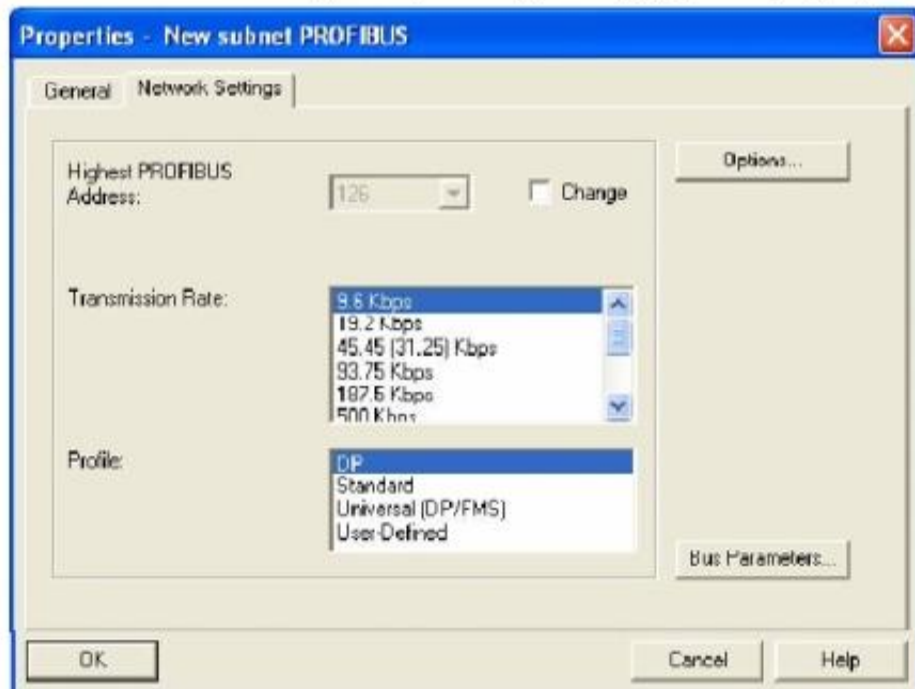
3. در کادر محاوره‌ای باز شده در تب General در باکس Interface بر روی دکمه‌ی Properties کلیک نمایید. یک کادر محاوره‌ای جدید باز خواهد شد.



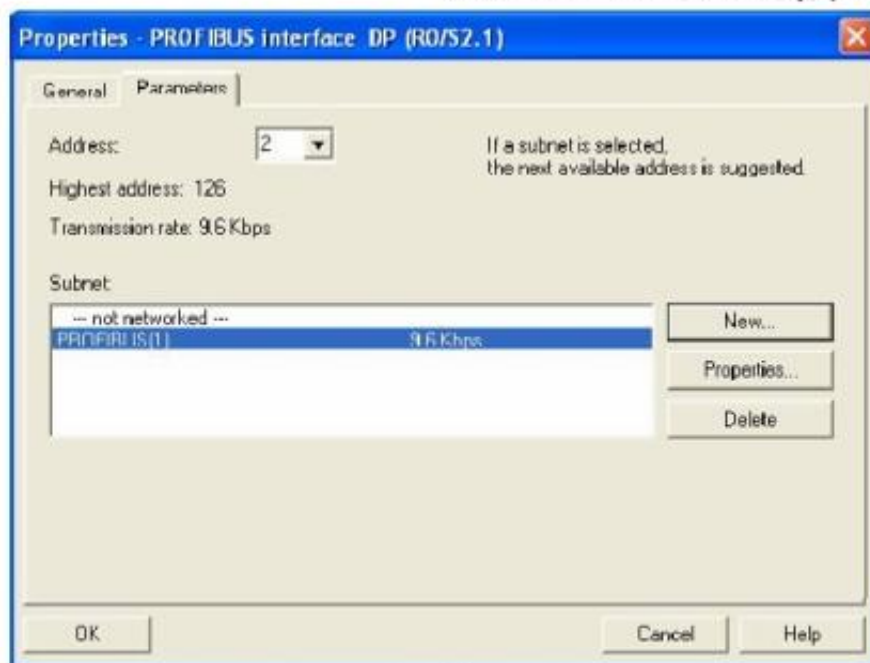
4. در کادر محاوره‌ای باز شده بر روی تب **Parameters** کلیک کرده و در قسمت **Address** آدرس دستگاه **Master** (همان CPU انتخابی برای PLC) را انتخاب نمایید. سپس بر روی دکمه‌ی **New** کلیک کنید تا به کادر محاوره‌ای بعدی برود.



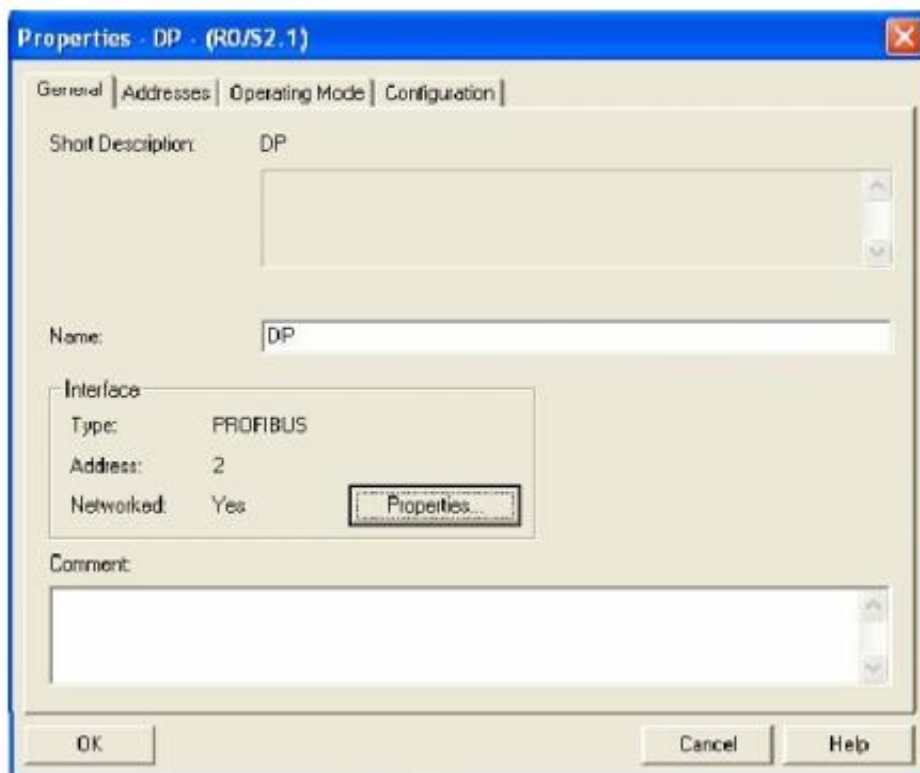
5. در پنجره‌ی باز شده بر روی تب **Network Setting** کلیک نمایید. سپس سرعت تبادل دیتا و نوع باس را انتخاب کرده و آنگاه بر روی دکمه‌ی **OK** کلیک نمایید.



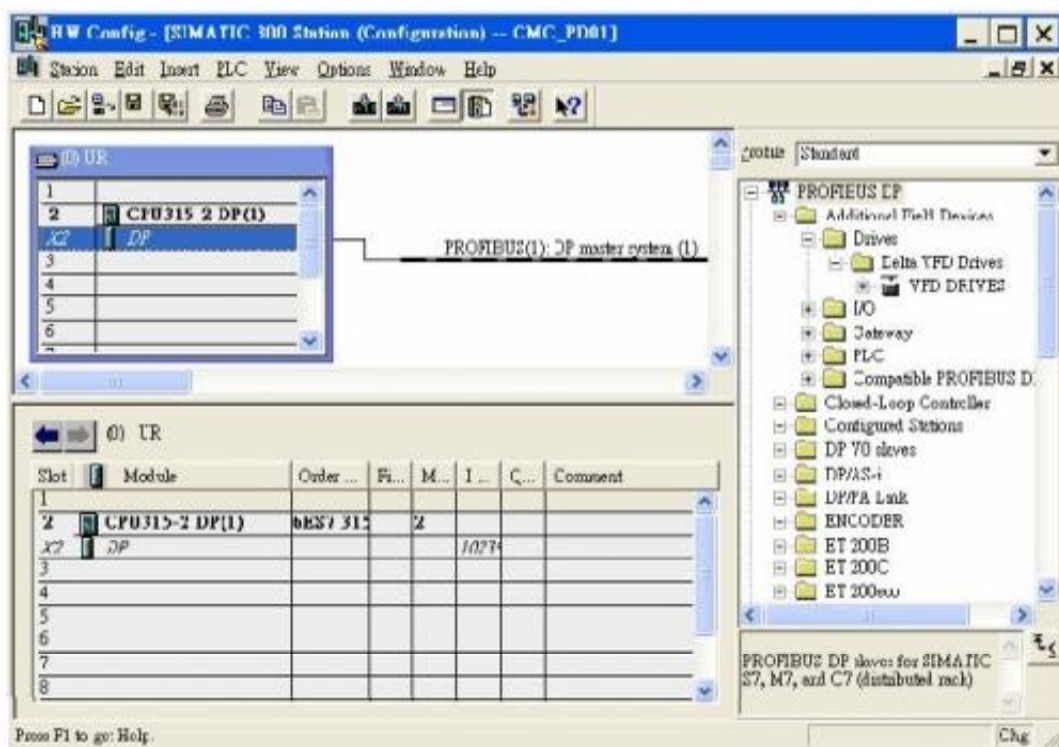
6. برای تأیید نهایی سرعت تبادل داده و آدرس دستگاه Master برای باس Profibus DP در پنجره‌ی باز شده بر روی دکمه‌ی OK کلیک نمایید.



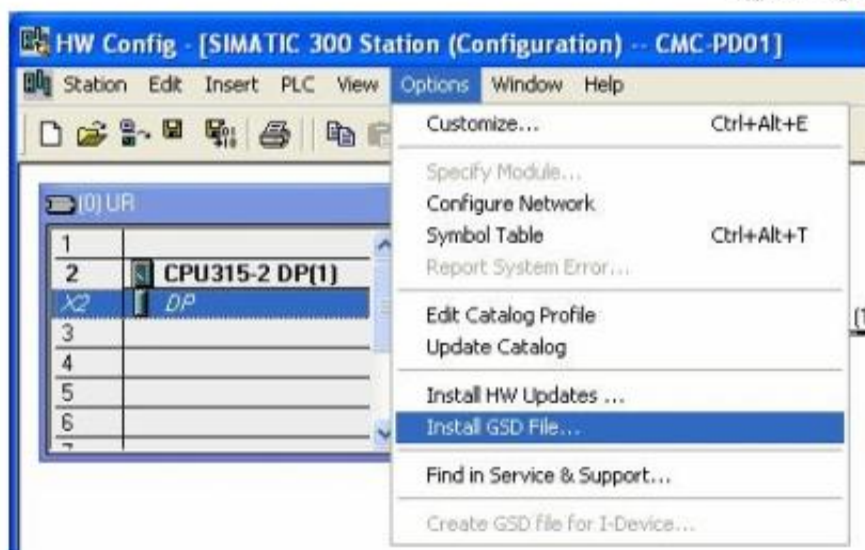
7. برای تأیید نهایی اطلاعات باس Profibus DP در کادر محاوره‌ای باز شده بر روی دکمه‌ی OK کلیک نمایید.



8. با یکبار تنظیم تمام پارامترها، یک باس Profibus DP در کنار سطر DP در جدول UR ظاهر خواهد شد.



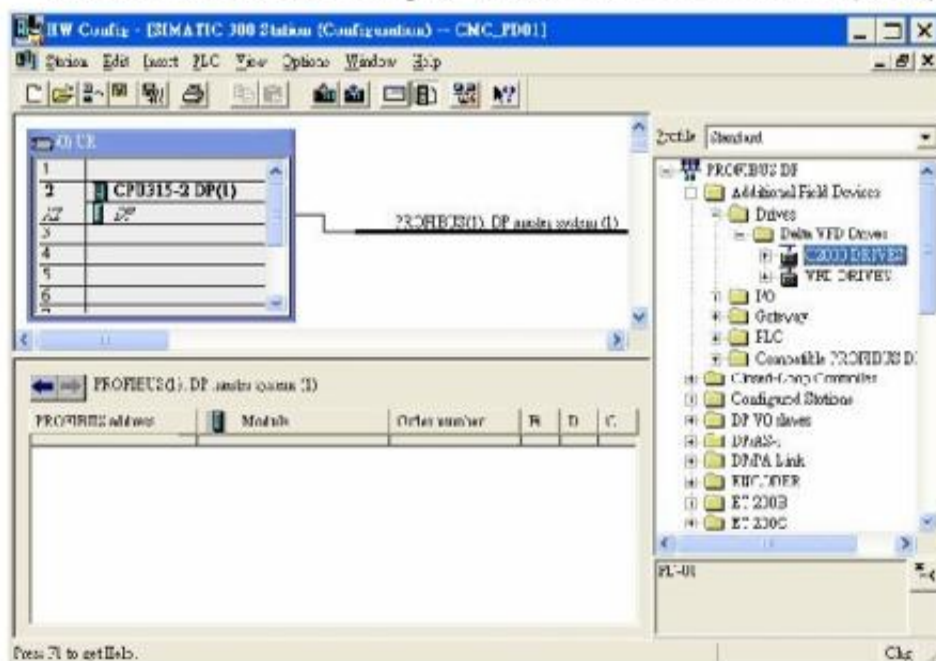
1. در پنجره‌ی HW-Config بر روی منوی Option و سپس بر روی زیر منوی Install GSD File کلیک نمایید.



2. با کلیک بر روی دکمه‌ی . . Browse در پنجره‌ی باز شده مسیر GSD File را که در آنجا ذخیره شده است را یافته و بر روی GSD File اینورتر کلیک نمایید سپس برای وارد کردن آن به نرم‌افزار SIMATIC Manager بر روی دکمه‌ی Open کلیک نمایید.

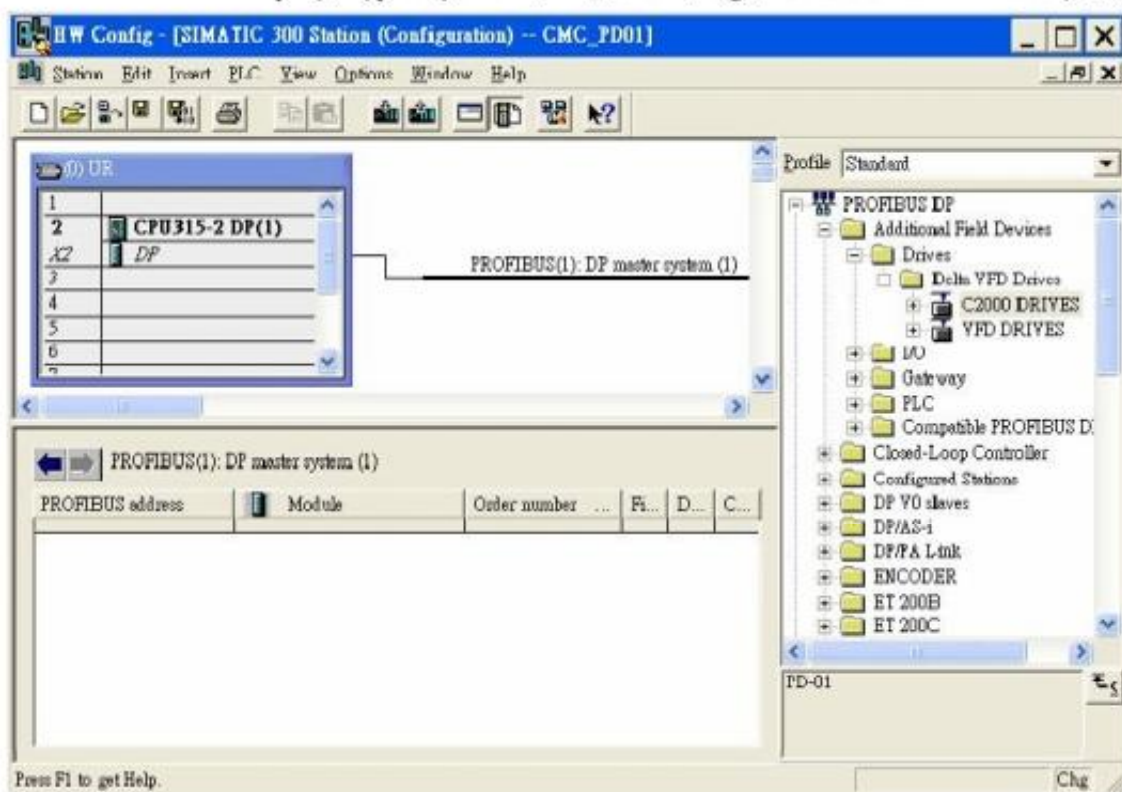


3. در ستون سمت راست پنجره‌ی HW-Config می‌توان GSD File اینورتر VFD-C2000 را مشاهده کرد. فایل C2000 DRIVES را باید به باس Profibus DP اضافه کنید.



اضافه کردن VFD-C2000 Slave به باس پروفیباس و تنظیم پارامترهای آن

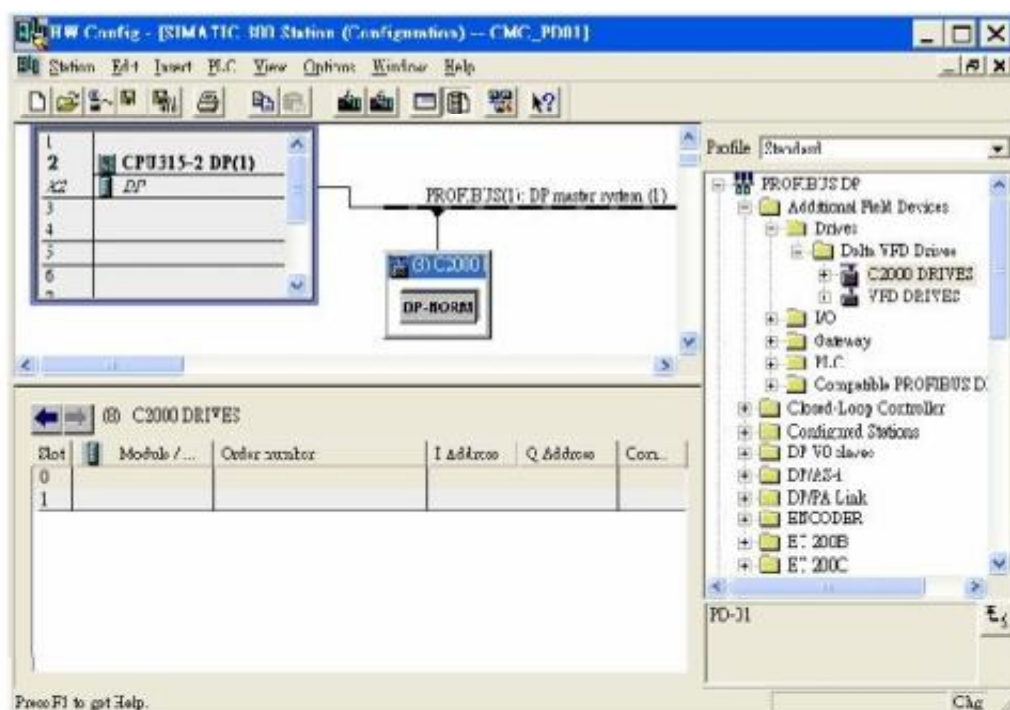
1. در پنجره‌ی HW-Config بر روی باس Profibus DP کلیک کنید تا انتخاب شود. با کلیک بر روی این باس در سمت راست، این باس کاملاً سیاه و توپر می‌شود. سپس در سمت چپ پنجره بر روی آیتم C2000 DRIVES دابل کلیک نمایید تا یک کادر محاوره باز شود.



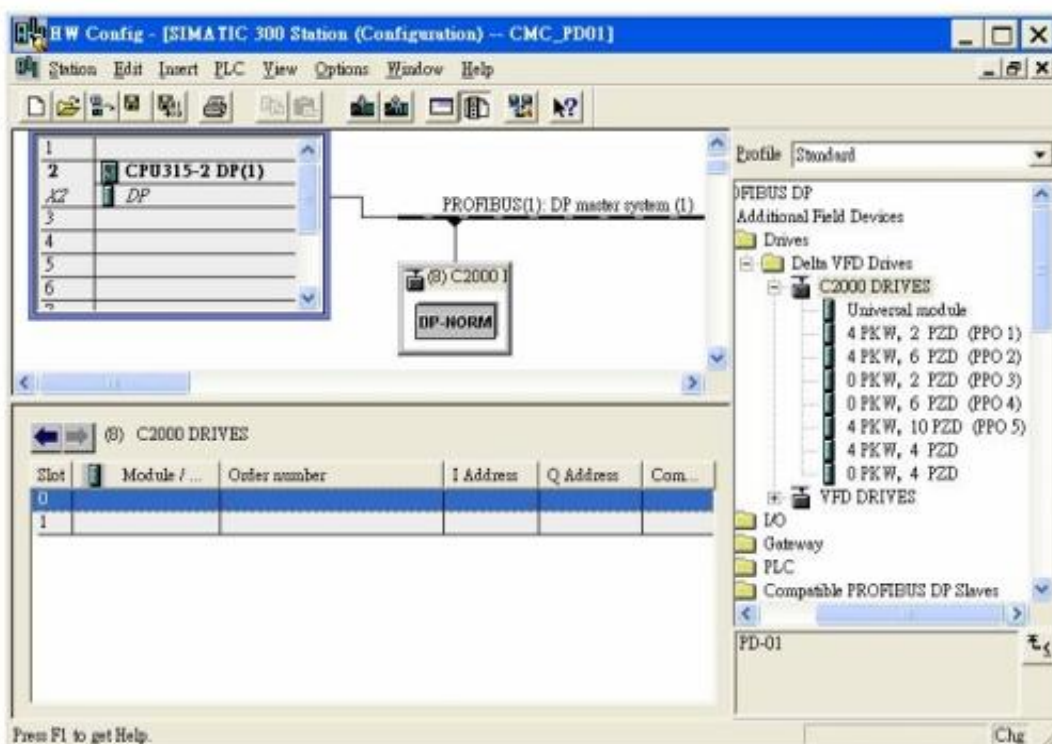
2. در کادر محاوره‌ای باز شده باکس آدرس اینورتر VFD-C2000 Slave که به فرمت دسیمال است را انتخاب کنید. این آدرس باید با آدرس تنظیم شده در پارامتر Pr.09-70 اینورتر یکسان باشد. در این مثال، ما آدرس اینورتر را برابر با 8 تنظیم کرده‌ایم. بنابراین، در این پنجره آدرس را باید برابر با 8 تنظیم نمایید. بعد از تنظیم آدرس بر روی دکمه‌ی OK کلیک نمایید.



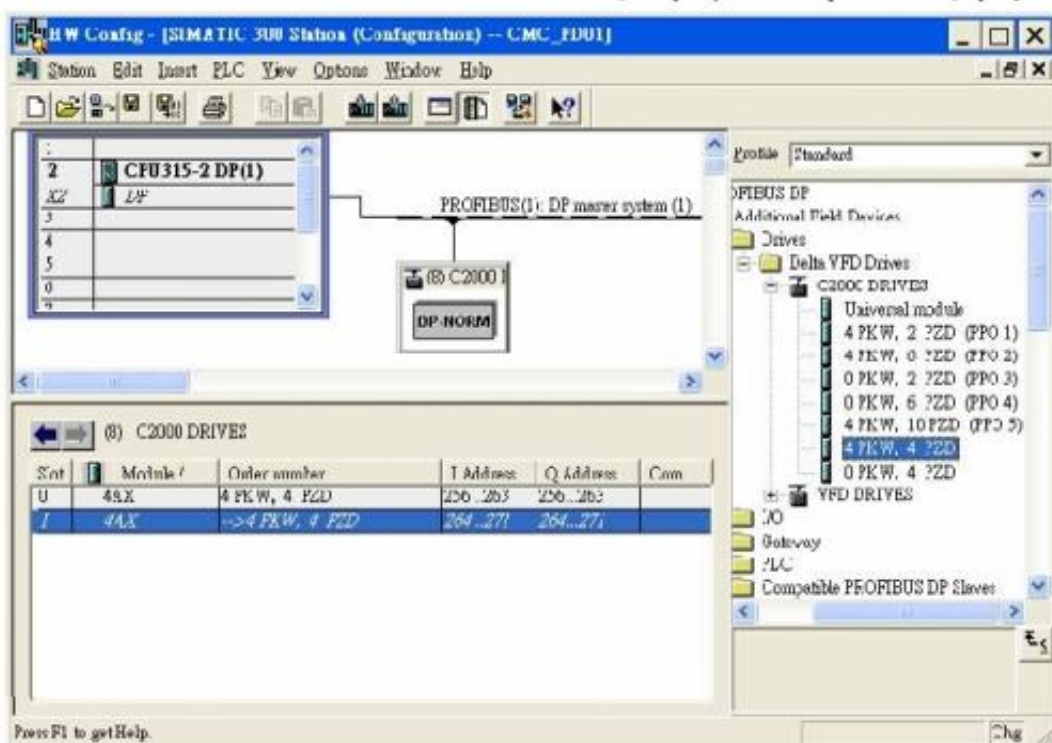
3. همانطور که در شکل زیر مشاهده می‌کنید، فایل VFD-C2000 به باس Profibus DP دستگاه Master اضافه شده است.



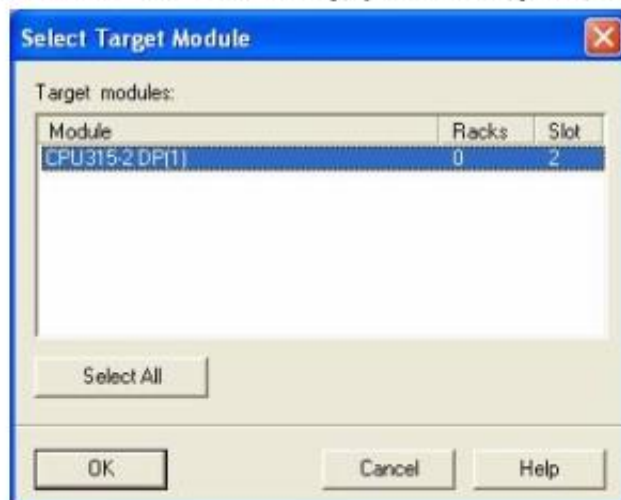
4. بر روی فایل VFD-C2000 در پنجره‌ی سمت چپ که به باس پروفیباس اضافه شده است، کلیک نمایید تا انتخاب شود. یک پنجره در پایین باز خواهد شد که مربوط به فایل C2000 DRIVES است. در این پنجره بر روی Slot 0 کلیک کنید. سپس در سمت راست پنجره بر روی علامت + کنار فایل C2000 DRIVES کلیک نمایید تا یک بخش کشویی باز شود. آنگاه بر روی آیتم 4PKW 4PZD دابل کلیک نمایید تا به Slot 0 اضافه شود.



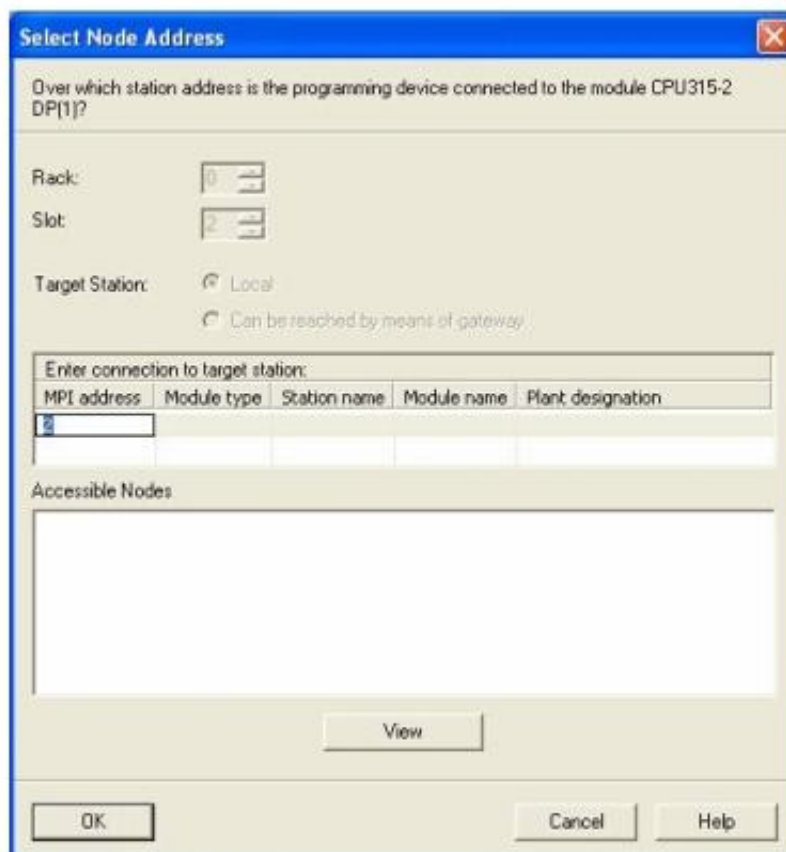
5. مطابق با روش گفته شده در بخش 4 برای Slot 1 نیز این کار را انجام دهید. یعنی 4PKW را برای Slot 1 و Slot 0 پیکربندی کنید.



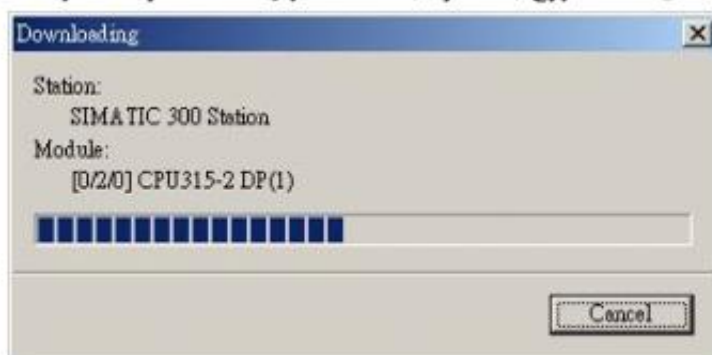
6. بعد از اینکه تمام پارامترها را پیکربندی کردید، برای دانلود پیکربندی انجام شده به PLC در نوار ابزار پنجره‌ی HW-Config بر روی آیکن  کلیک کنید. کادر محاوره‌ای مطابق با شکل زیر باز خواهد شد. برای رفتن به کادر محاوره‌ای بعدی بر روی دکمه‌ی OK کلیک کنید.



7. در کادر محاوره‌ای باز شده برای دانلود پارامترهای پیکربندی شده به PLC باید بر روی دکمه‌ی OK کلیک نمایید.



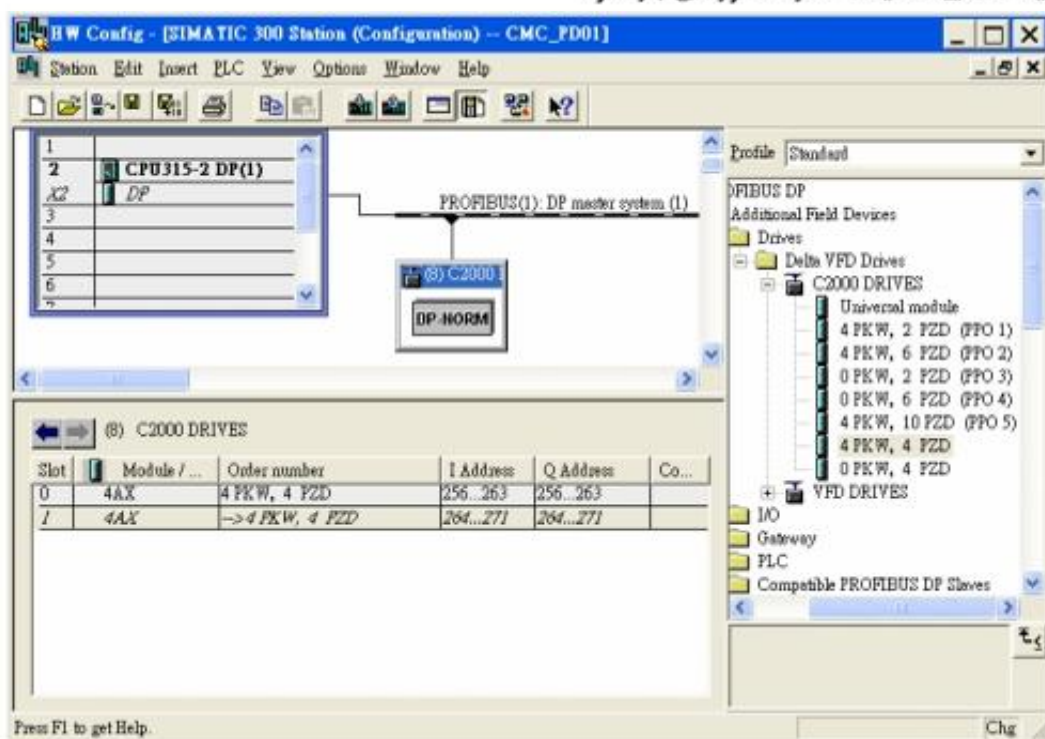
8. پارامترهای پیکربندی شده شروع به دانلود به سخت‌افزار PLC خواهند کرد.



9. بعد از اینکه دانلود به پایان رسید، نمایشگر NET LED بر روی کارت شبکه‌ی CMC-PD01 به رنگ سبز روشن خواهد شد.

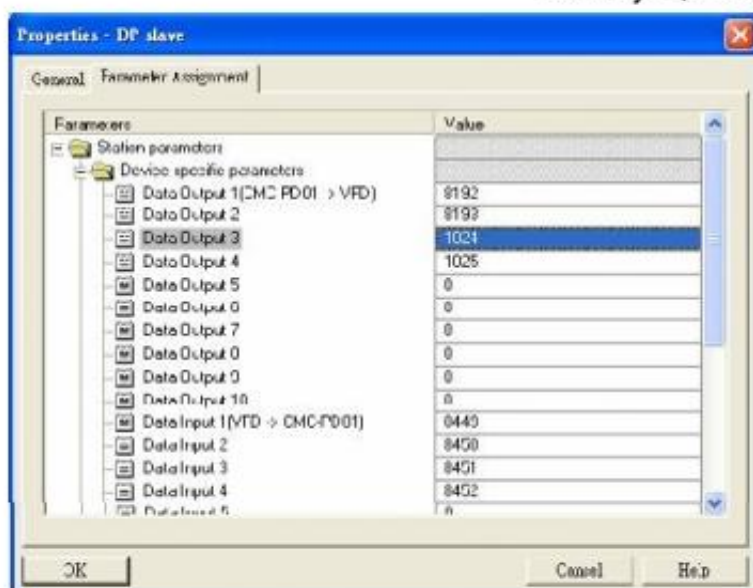
مسیردهی دیتا (Data Mapping)

بر روی آیکن C2000 DRIVES Slave که در زیر باس Profibus DP قرار گرفته است، دابل کلیک نمایید تا یک کادر محاوره‌ای باز شود.

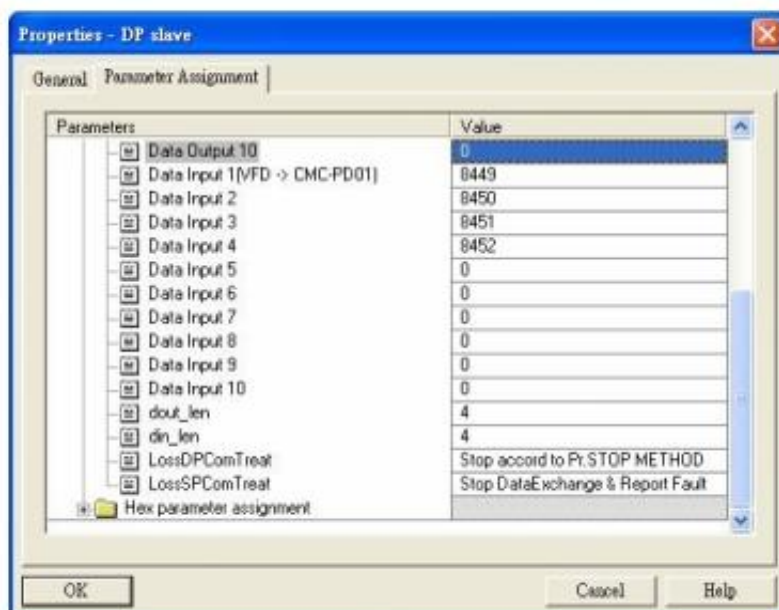


نواحی دیتای ورودی (Data Input areas) و نواحی دیتای خروجی (Data Output areas) در PZD برای آدرس‌های Slave مسیردهی شده‌اند. فرمت دیتا برای دیتای ورودی و خروجی از نوع دسیمال می‌باشد و می‌توان آنها را به صورت دستی وارد کرد. برای مثال، اگر مقدار 1024 دسیمال را در

سومین دیتای خروجی (Data Output 3) وارد کنیم، آدرس اینورتر (hex) 0400 یعنی پارامتر Pr.04-00 اینورتر انتخاب خواهد شد.



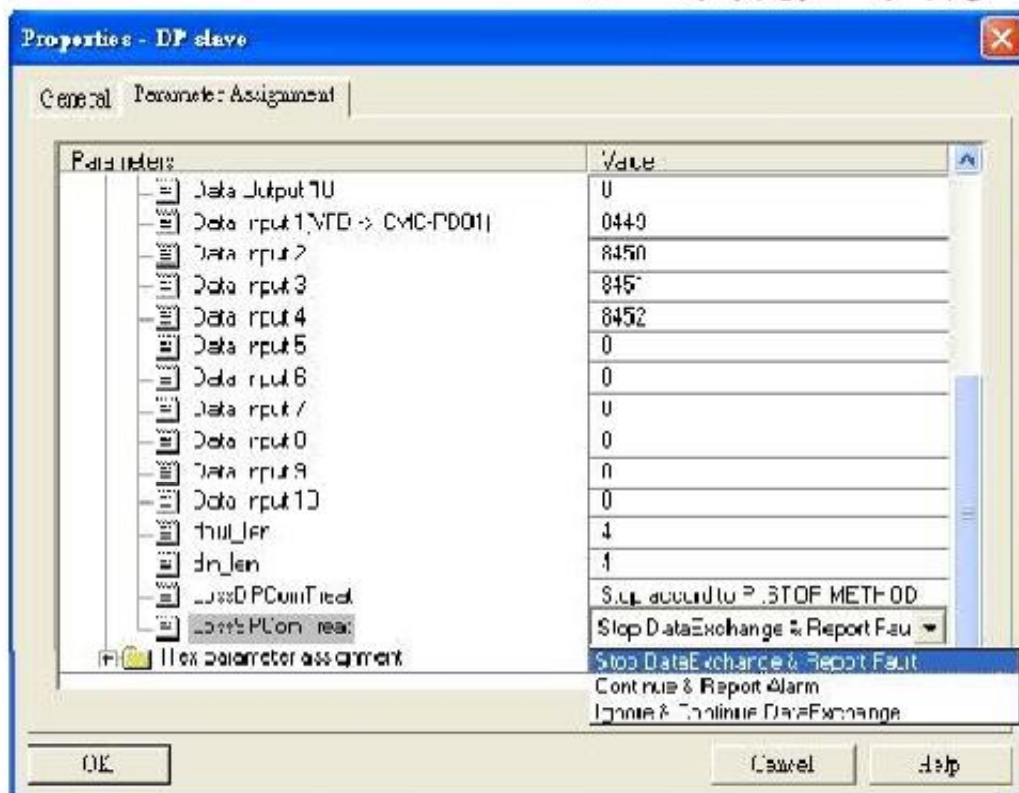
در شکل بالا، نوار اسکرول سمت راست را به پایین بکشید، تا وارد شکل پایین شود. `din_len` به معنی طول دیتا در دیتای ورودی و `dout_len` به معنی طول دیتا در دیتای خروجی (Data Output) می‌باشد.



در بخش پارامترها آیتم `LossDPCom Treat` را انتخاب می‌کنیم. برای اطلاع از جزئیات جدول زیر را بررسی کنید.

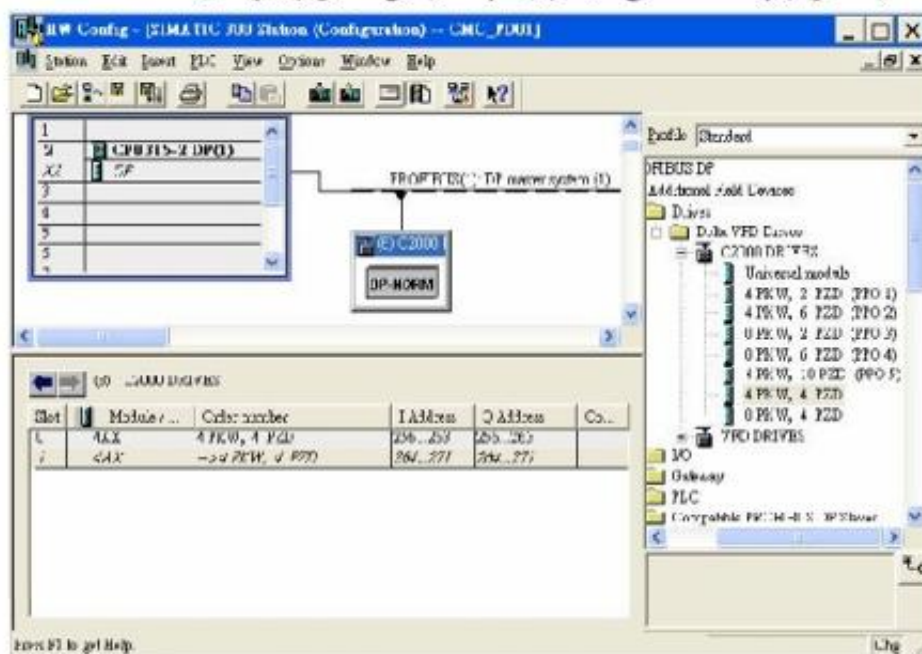
عملکرد اینورتر در هنگامی که ارتباط کارت CMC-PD01 با دستگاه Profibus DP Master قطع شود.	
بعد از قطع ارتباط شبکه اینورتر به کارش ادامه خواهد شد.	Ignore and countinue
بعد از قطع ارتباط شبکه اینورتر مطابق با روش استوپ تنظیم شده متوقف خواهد شد.	Stop accord to Pr.STOP METHOD

مطابق با شکل زیر، در بخش پارامترها آیتم **LossSPComTreat** را انتخاب می‌کنیم. برای اطلاع از جزئیات این پارامتر به جدول زیر مراجعه نمایید.



عملکرد باس Profibus DP در هنگام قطع ارتباط کارت CMC-PD01 با اینورتر VFD-C2000	
کارت CMC-PD01 تبادل دیتا با دستگاه Profibus DP Master را متوقف کرده و گزارش خطا خواهد داد.	Stop Data Exchange & Report Fault
کارت CMC-PD01 تبادل دیتا با دستگاه Profibus DP Master را حفظ کرده و گزارش آلارم را خواهد داد.	Continue & Report Alarm
کارت CMC-PD01 تبادل دیتا با دستگاه Profibus DP Master را حفظ کرده و گزارش آلارم را خواهد داد.	Ignore & Continue Data Exchange

همانطور که در شکل زیر مشاهده می‌کنید، پارامترها در این شکل پیکربندی شده‌اند:



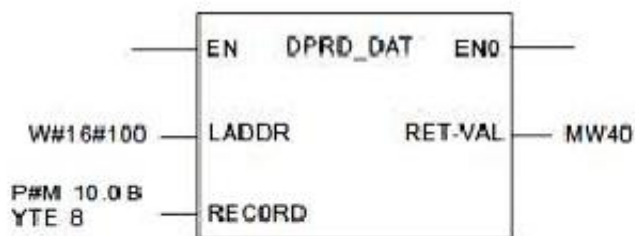
در جدول زیر، مسیریابی دیتای وابسته به پیکربندی پارامترهای اینورتر که در شکل بالا نشان داده شده است آورده شده است.

PKW/PZD	External I/O word for S7-300	Data transmission direction in PROFIBUS DP network	Parameter address of AC motor drive (hex)
PKW	PQW256	➔	4 words combined make it able to read/write one AC motor drive parameter.
	PQW258		
	PQW260		
	PQW262		
	PIW256	➔	PKW returning data
	PIW258		
	PIW260		
	PIW262		

PKW/PZD	External I/O word for S7-300	Data transmission direction in PROFIBUS DP network	Parameter address of AC motor drive (hex)
PZD	PQW264	➔	2000 (Data Output 1)
	PQW266		2001 (Data Output 2)
	PQW268		0400 (Data Output 3)
	PQW270		0401 (Data Output 4)
	PIW264	➔	2101 (Data Input 1)
	PIW266		2102 (Data Input 2)
	PIW268		2103 (Data Input 3)
	PIW270		2104 (Data Input 4)

برای برنامه‌نویسی این مثال در بلوک OB1 باید ابتدا فرامین SFC14, SFC15 را فراخوانی نمایید.

SFC15 ← از تابع SFC15 برای نوشتن پیوسته دیتا در دستگاه استاندارد Profibus Slave استفاده می‌شود.



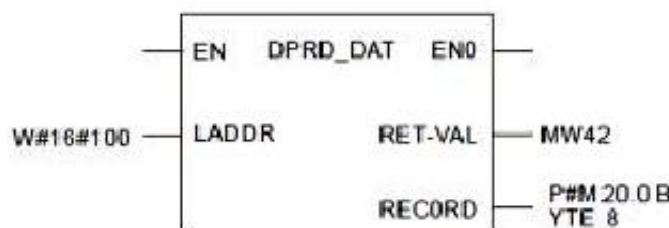
در ورودی LADDR آدرس شروع ناحیه‌ی مسیردهی شده دستگاه Slave برای خروجی دستگاه Master (به عنوان مثال، PQW256 تعیین شده در بخش پیکربندی پارامترها در پنجره‌ی HW-Config) نوشته می‌شود. فرمت این ورودی هگزادسیمال است. بعنوان مثال، از مقدار هگزادسیمال W#16#100 در ورودی LADDR تابع SFC15 برای نمایش PQW256 استفاده می‌شود. معادل دسیمال 100hex برابر با 256 است.

در ورودی RECORD آدرس شروع و طول دیتا نوشته می‌شود. طول دیتای RECORD باید سازگار با طول دیتا در پارامتر پیکربندی شده بوده و واحد آن باید بایت باشد. فرمت ورودی این پارامتر به عنوان مثال P#M 10.0 BYTE 8 است.

در خروجی RET_VAL این تابع، آدرس کد خطا نوشته می‌شود. وقتی که ورودی EN تابع SFC15 فعال (True) شود، تابع SFC15 اجرا خواهد شد. اگر در هنگام اجرای تابع هر خطایی رخ دهد، کد خطای تولید شده در خروجی RET_VAL تابع فوق ذخیره خواهد شد. وقتی که خطایی در تابع رخ ندهد، مقدار 0 در خروجی RET_VAL ذخیره خواهد شد.

همانطور که در شکل بالا مشاهده می‌کنید، ما توانستیم دیتایی را که در رجیسترهای MW10, MW12, MW16, MW18 وجود دارد را در داخل رجیسترهای PQW256, PQW258, PQW260, PQW262 بنویسیم و دیتای موجود در PQW به صورت اتوماتیک به دستگاه Slave ارسال خواهد شد.

SFC14 ← از این تابع برای نوشتن پیوسته دیتا در دستگاه استاندارد Profibus DP Slave استفاده می‌شود.



در ورودی LADDR تابع SFC14 آدرس شروع ناحیه‌ی مسیردهی شده دستگاه Slave برای ورودی دستگاه Master (عنوان مثال، PIW256 تعیین شده در بخش پیکربندی پارامترها در پنجره‌ی HW-Config) نوشته می‌شود. فرمت این ورودی هگزادسیمال است. به عنوان مثال، از مقدار هگزادسیمال W#16#100 در ورودی LADDR تابع SFC14 برای نمایش PIW256 استفاده می‌شود. معادل دسیمال 100 hex برابر با 256 است.

در خروجی RECORD این تابع آدرس شروع و طول دیتا برای خواندن مقدار آن نوشته می‌شود. طول دیتای RECORD باید سازگار با طول دیتا در پارامتر پیکربندی شده بوده و واحد آن باید برحسب بایت باشد. بعنوان مثال، فرمت این پارامتر به شکل P#M20.0 BYTE 8 است.

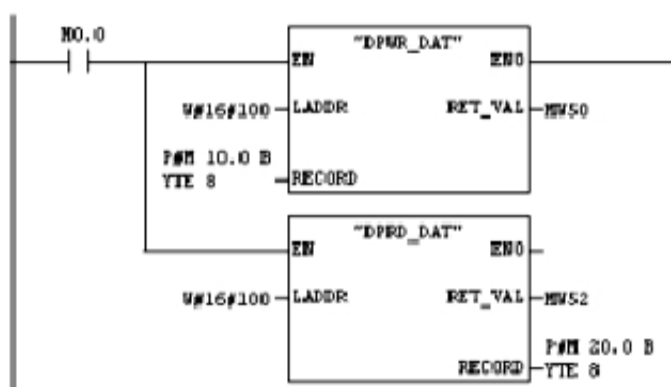
در خروجی RET_VAL این تابع آدرس کد خطا نوشته می‌شود. وقتی که ورودی EN تابع SFC14 فعال (True) شود، تابع فوق اجرا خواهد شد. اگر در هنگام اجرای تابع هر خطایی رخ دهد، کد خطای تولید شده در خروجی RET_VAL تابع فوق ذخیره خواهد شد. وقتی که خطایی در تابع رخ ندهد، مقدار 0 در خروجی RET_VAL ذخیره خواهد شد.

همانطور که در شکل بالا مشاهده می‌کنید، ما توانستیم دیتای ارسال شده از دستگاه Slave را که در رجیسترهای PIW256, PIW258, PIW260, PIW261 ذخیره شده است، را داخل رجیسترهای MW20, MW22, MW24, MW26 بنویسیم.

برنامه‌ی بلوک OB1 دستگاه Master برای این مثال مطابق با دیاگرام نردبانی شکل زیر است:

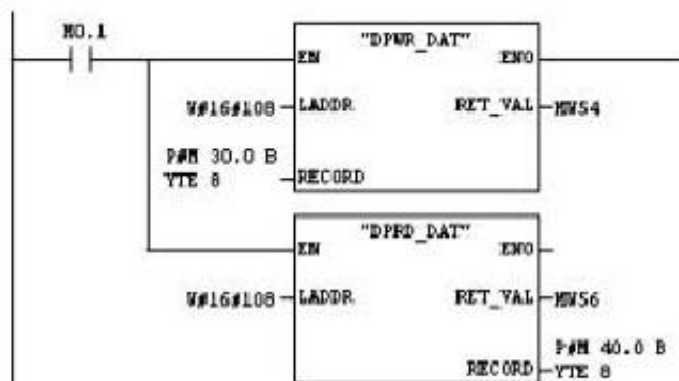
Network 1: Title:

Comments:



Network 2: Title:

Comments:





وقتی که $M0.0=ON$ شود، PKW اجرا خواهد شد. دیتای PKW مطابق با جدول زیر می‌باشد:

Modifying value in parameter 04-02 (hex) of AC motor drive by PKW (the address of parameter 04-02 is 402 (hex))				
Data from master to Slave	MW10	MW12	MW14	MW16
Data value (hex)	2402	0	0	500
Data from Slave to master	MW20	MW22	MW24	MW26
Data value (hex)	1402	0	0	500
Reading value in parameter 04-02 (hex) of AC motor drive by PKW (the address of parameter 04-02 is 402 (hex))				
Data from master to Slave	MW10	MW12	MW14	MW16
Data value (hex)	1402	0	0	0
Data from Slave to master	MW20	MW22	MW24	MW26
Data value (hex)	1402	0	0	500

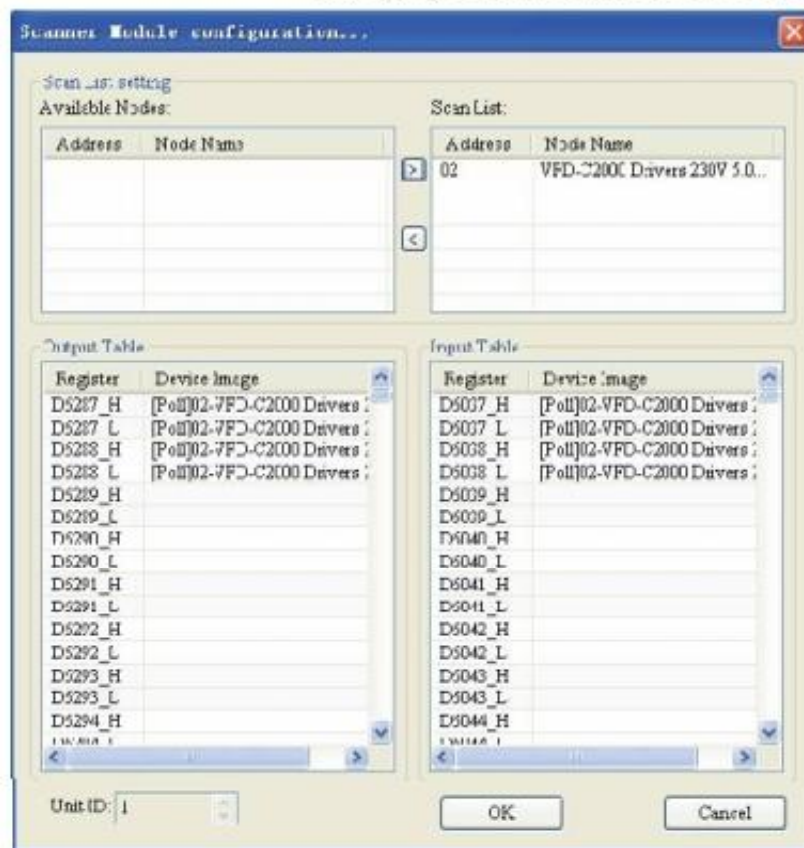
وقتی که $M0.1=ON$ شود، PZD اجرا خواهد شد. در جدول زیر دیتای ارسال به PZD آورده

شده است:

Register in master	Data transmission of PROFIBUS DP bus	Parameter address of AC motor drive
MW30 (PQW264)		2000 (hex)
MW32 (PQW266)		2001 (hex)
MW34 (PQW268)		0400 (hex)
MW36 (PQW270)		0401 (hex)
MW40 (PIW264)		2101 (hex)
MW42 (PIW266)		2102 (hex)
MW44 (PIW268)		2103 (hex)
MW46 (PIW280)		2104 (hex)

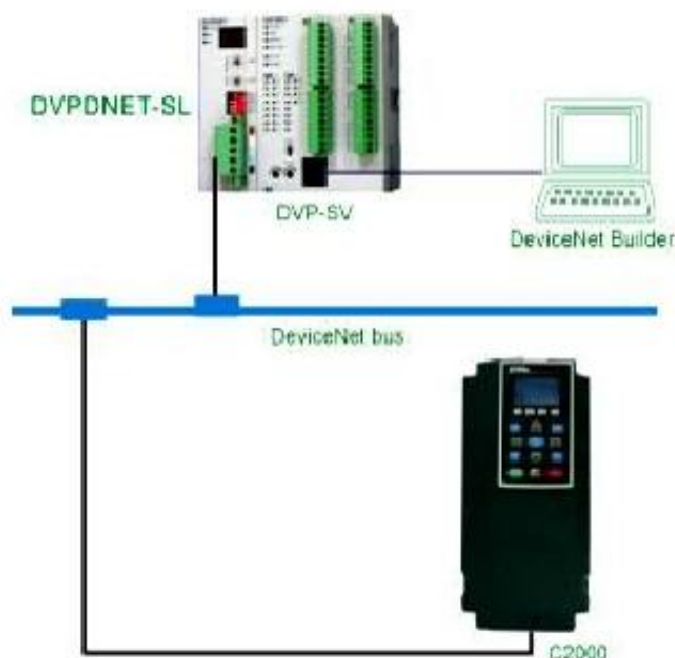
8- 6 مثال کاربردی برای مازول CMC-DN01

برای برقراری اتصال ورودی خروجی ابتدا باید در نرم افزار DeviceNet Builder صفحه‌ی Scanner Modul configuration ... را باز کنید.



در این صفحه می‌توان پیکربندی اینورتر در دستگاه DeviceNet Master را مشاهده کرد. از رجیسترهای موجود در Output Table و Input Table برای تبادل دیتا بین اینورتر و دستگاه DeviceNet Master استفاده می‌شود. PLC سری DVP-SV دلتا و مازول DPVDNET-SL دلتا با یکدیگر توسط این رجیسترها تبادل دیتا می‌کنند. رجیسترهای کنترلی D6287, D6288, D6037, D6039 رجیسترهای موجود در DVP-SV هستند. توسط رجیسترهای کنترلی موجود در DVP-SV می‌توان پارامترهای اینورتر را کنترل کرده و حتی مانیتور کرد. در این بخش، می‌خواهیم نحوه‌ی پیکربندی اینورتر سری VFD-C2000 را تحت یک مثال بررسی نماییم.

1. ساختار یک شبکه‌ی DeviceNet مانند شکل زیر است. مطابق با شکل زیر، باید یک شبکه‌ی DeviceNet را ایجاد کنید.



2. تنظیمات مازول DVPDNET-SL و اینورتر VFD-C2000 برای پیکربندی باس شبکه‌ی DeviceNet به شرح زیر است :

نام مازول	آدرس گرهی دستگاه در شبکه	نرخ (سرعت) تبادل دیتای باس
DVPDNET-SL	1	500kbps
VFD-C2000	2	500kbps

برای تنظیم آدرس گرهی اینورتر در شبکه (Node address) و سرعت تبادل دیتای اینورتر روی باس شبکه (Baud rate) پارامترهای اینورتر را باید مطابق با جدول زیر تنظیم نمایید.

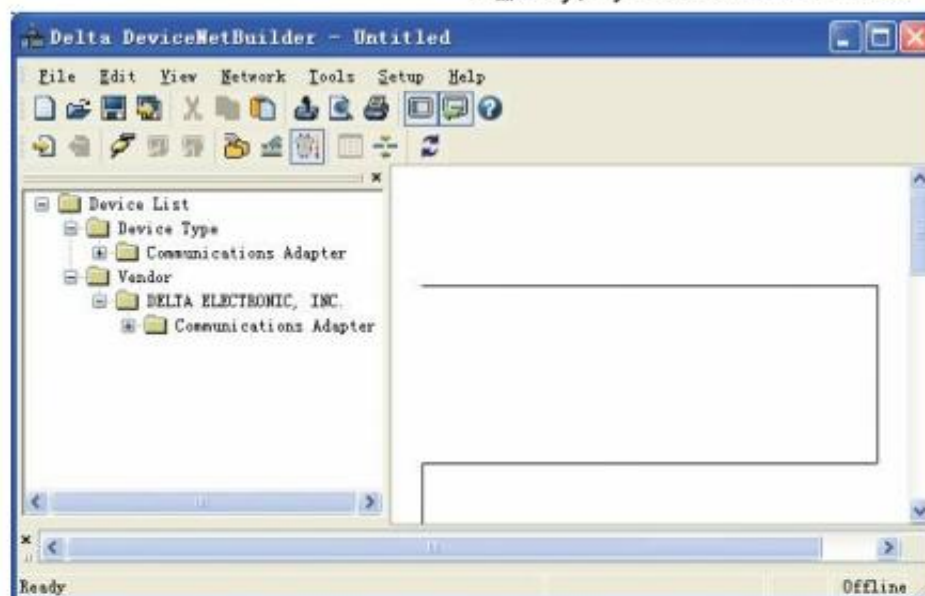
پارامتر	نام پارامتر	محدوده‌ی قابل تنظیم	مقدار تنظیمی
Pr.00-20	منبع تنظیم فرکانس اینورتر		8
Pr.00-21	منبع فرمان اینورتر		5
Pr.09-30	روش دیکد کردن ارتباطات شبکه		0
Pr.09-70	آدرس گرهی اینورتر در شبکه DeviceNet	0~63	2
Pr.09-71	سرعت تبادل دیتا بر روی باس شبکه‌ی DeviceNet	Standard mode	2
		Extended mode	
		0:125 kbps 1:250 kbps 2:500 kbps	
		0:10 kbps 1:20 kbps 2:50 kbps 3:125 kbps 4:250 kbps 5:500 kbps 6:800 kbps 7:1 Mbps	

0	اگر مقدار این پارامتر 0 تنظیم گردد، پارامتر Pr.09-71 Standard mode وارد مد خواهد شد اگر مقدار این پارامتر برابر با 1 تنظیم گردد، پارامتر Pr.09-71 Extended mode وارد مد خواهد شد	انتخاب مد برای پارامتر Pr.09-71	Pr.09-72
---	---	---------------------------------	----------

3. ماژول DVPDNET-SL دلتا و اینورتر هر دو باید به صورت نرمال کار کنند. سیم‌بندی شبکه را به درستی و با دقت انجام دهید. در صورت صحیح بودن سیم‌بندی، تغذیه‌ی شبکه‌ی DeviceNet باید نرمال باشد. در صورتی که خطایی رخ داد، باید وضعیت نمایشگرهای LED را بررسی کرده و خطای بوجود آمده را رفع نمایید.

در این بخش، می‌خواهیم نحوه‌ی پیکربندی شبکه‌ی DeviceNet را در نرم‌افزار DeviceNet Builder که مختص شرکت دلتا است، بررسی نماییم.

1. نرم‌افزار DeviceNet Builder را اجرا نمایید.



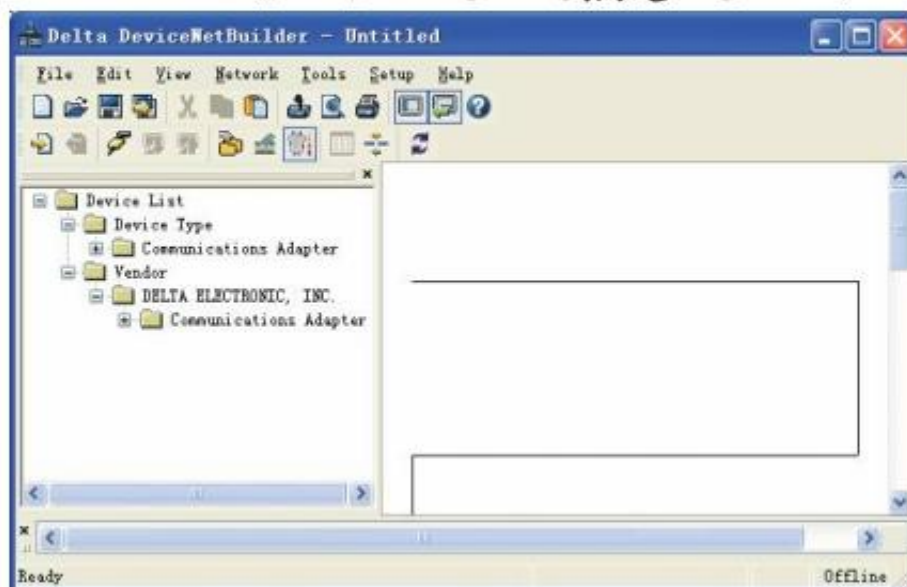
2. از منوی Setup بر روی زیرمنوی Communication Setting کلیک کرده و سپس برای تنظیم پورت سریال بر روی آیتم Serial Port Setting کلیک نمایید.



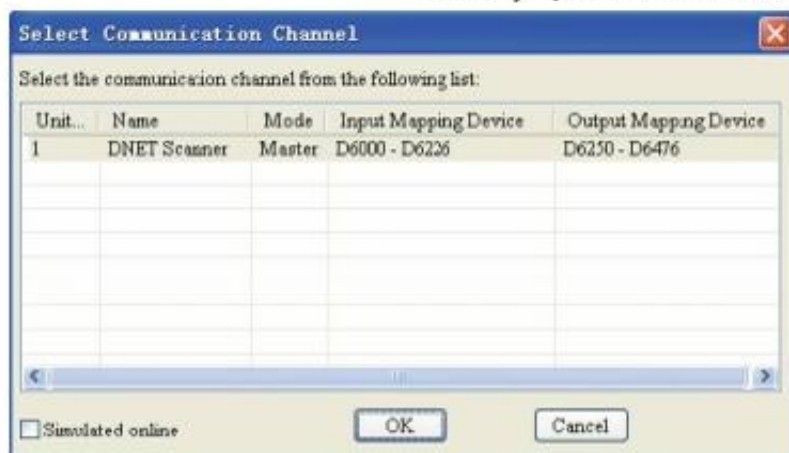
3. پارامترهای ارتباطات را مطابق با جدول زیر تنظیم نمایید:

پارامتر	توضیحات	مقدار پیش فرض
COM Port	پورت ارتباطات کامپیوتر را که می‌خواهید با آن به PLC متصل شوید را انتخاب نمایید	COM1
Address	ورودی Modbus Node ID را برای PLC انتخاب نمایید	01
Boud rate	سرعت تبادل دیتا بین کامپیوتر و PLC را تنظیم نمایید	9600 bps
Data bits	پروتکل ارتباطات بین کامپیوتر و PLC را انتخاب نمایید	7
Parity		Even parity
Stop bit		1
Mode	مد ارتباطات بین کامپیوتر و PLC را انتخاب نمایید	ASCII

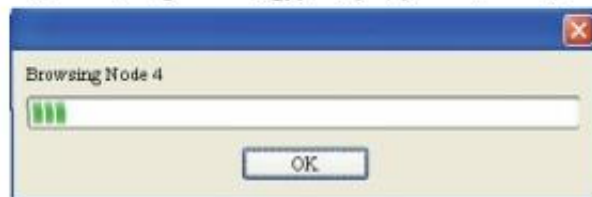
4. برای برگشت به صفحه‌ی اصلی بر روی دکمه‌ی OK کلیک نمایید.



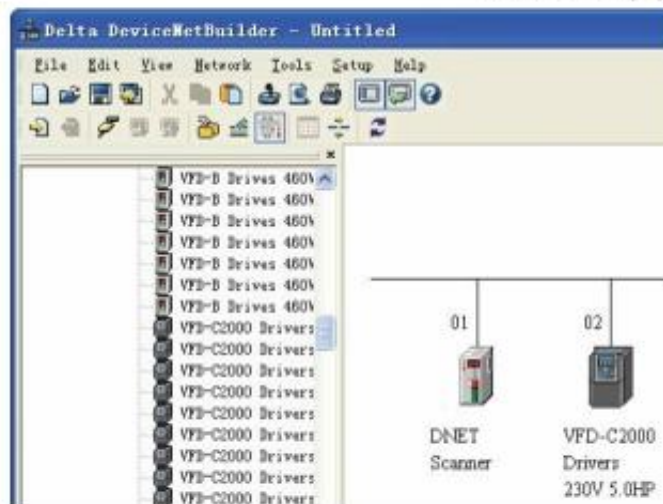
5. از منوی Network بر روی گزینه‌ی Online کلیک کنید پنجره‌ی Select Communication Channel باز خواهد شد.



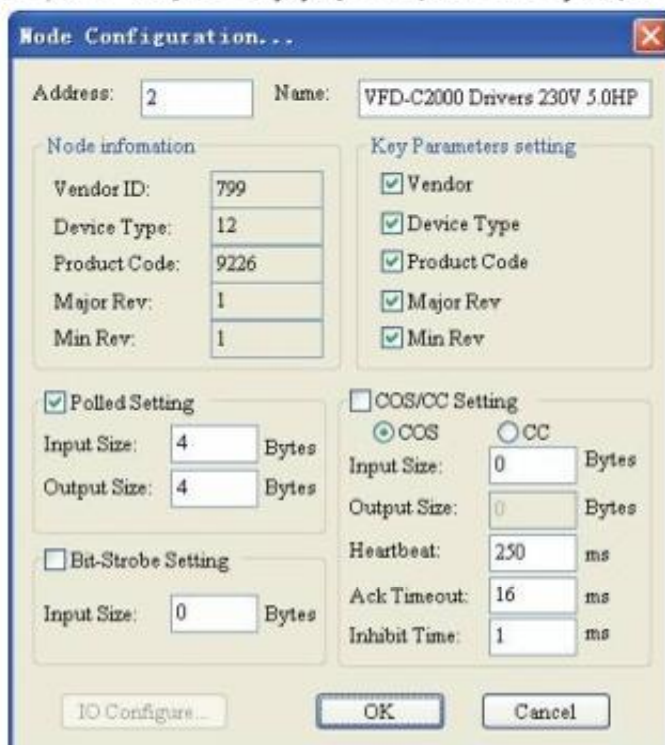
6. بر روی دکمه‌ی OK کلیک کنید تا نرم‌افزار شروع به اسکن شبکه نماید.



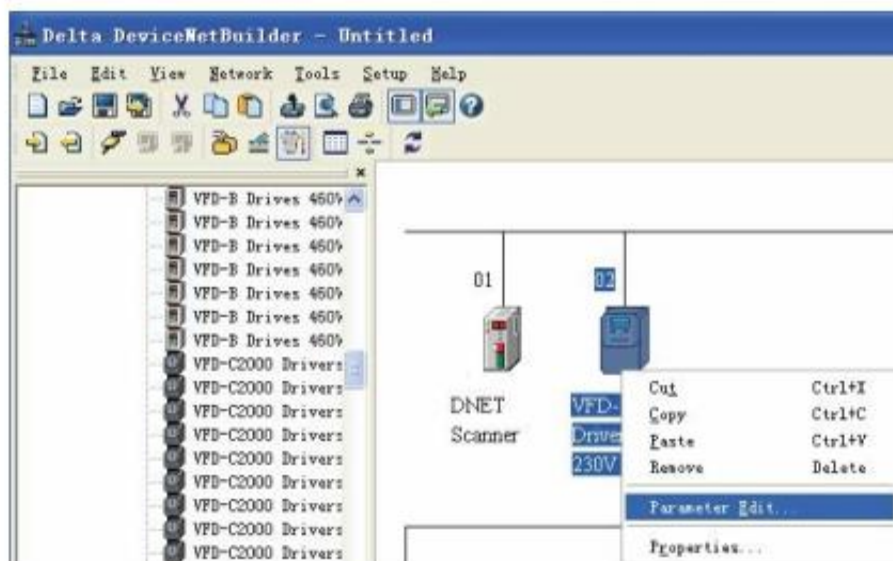
7. اگر پروسس بار شروع به پیروی نکرد، یعنی ارتباط بین کامپیوتر و PLC برقرار نبوده و یا اینکه پورت سریال مورد استفاده در این نرم‌افزار توسط برنامه‌ی دیگری اشغال شده و باز می‌باشد. وقتی که اسکن به پایان رسید، یک کادر محاوره‌ای باز می‌شود که بیانگر تکمیل شدن عمل اسکن می‌باشد. سپس تمام آیکن‌های مربوط به هر Node و نام تمام دستگاه‌های متصل شده به شبکه در نرم‌افزار نمایش داده خواهد شد. در این مثال، Node Address (آدرس گره‌ی دستگاه در شبکه) برای ماژول DVPDNET-SL برابر با 01 است.



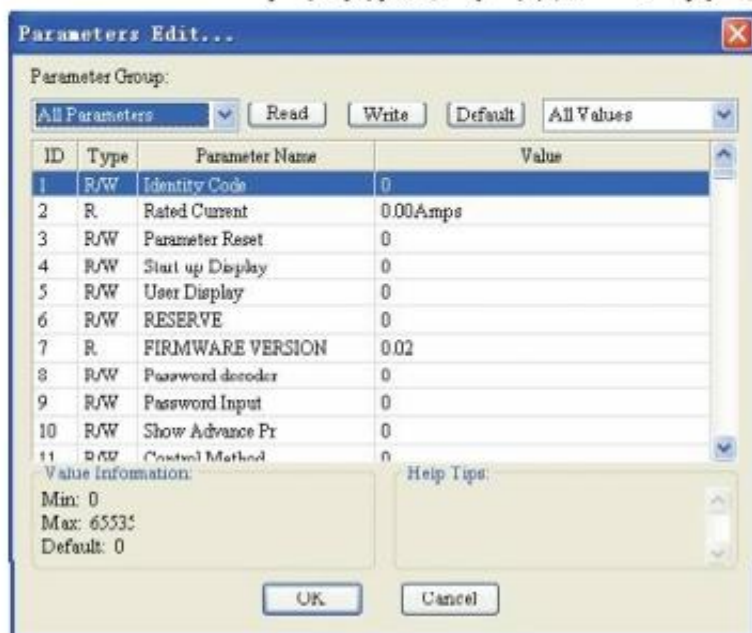
8. بر روی آیکن VFD-C2000 Drives (ماژول CMC-DN01 متصل شده به اینورتر VFD-C2000) دابل کلیک کنید تا پنجره‌ی محاوره‌ای Node Configuration باز شود. در این پنجره مقدار Input Size و Output Size را برابر با 4 bytes تنظیم نمایید.



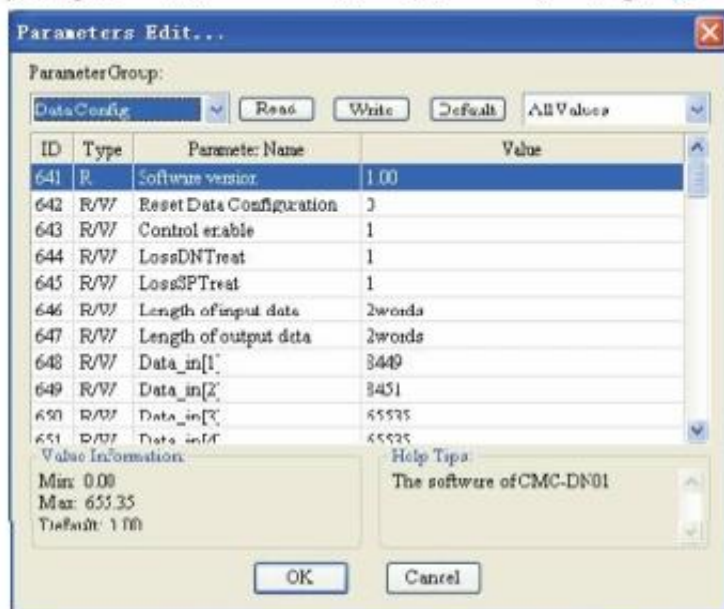
9. سپس بر روی آیکن VFD-C2000 Drives راست کلیک کرده و . . . Parameter Edit را انتخاب نمایید.



10. کادر محاوره‌ای مربوط به تنظیم پارامترهای اینورتر باز خواهد شد.



11. در بخش Parameter Group گزینه‌ی Data Config را انتخاب نمایید. سپس پارامترهای مربوط به مسیریابی دیتا در حافظه (Data mapping) به نمایش درخواهد آمد.



12. در پنجره‌ی فوق باید پارامترها را مطابق با مقادیر گفته شده در زیر تنظیم نمایید :

Length of input data → 2 words

Length of output data → 2 words

Data_in [1] → 8449 (Converted from VFD-C2000 status word H2101)

Data_in [2] → 8451 (Converted from VFD-C2000 output frequency H2103)

Data_out [1] → 8192 (Converted from VFD-C2000 control word H2000)

Data_out [2] → 8193 (Converted from VFD-C2000 given frequency H2001)

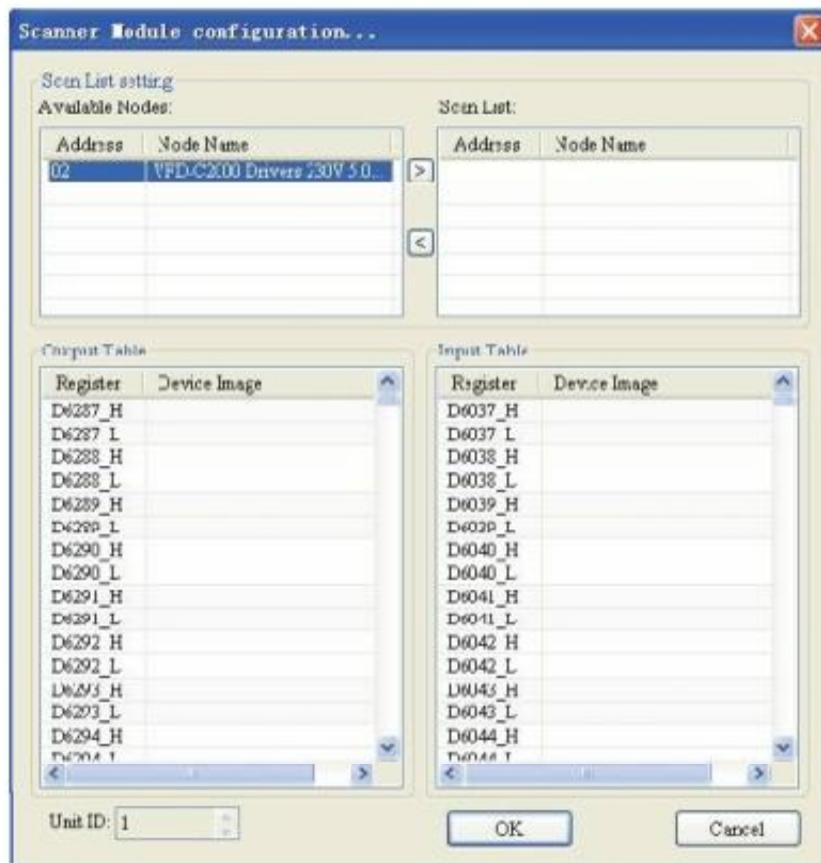
بعد از اینکه تمام تنظیمات را انجام دادید، گزینه‌ی **All values** را مطابق با شکل زیر انتخاب کرده و سپس بر روی دکمه‌ی **Write** کلیک نمایید.



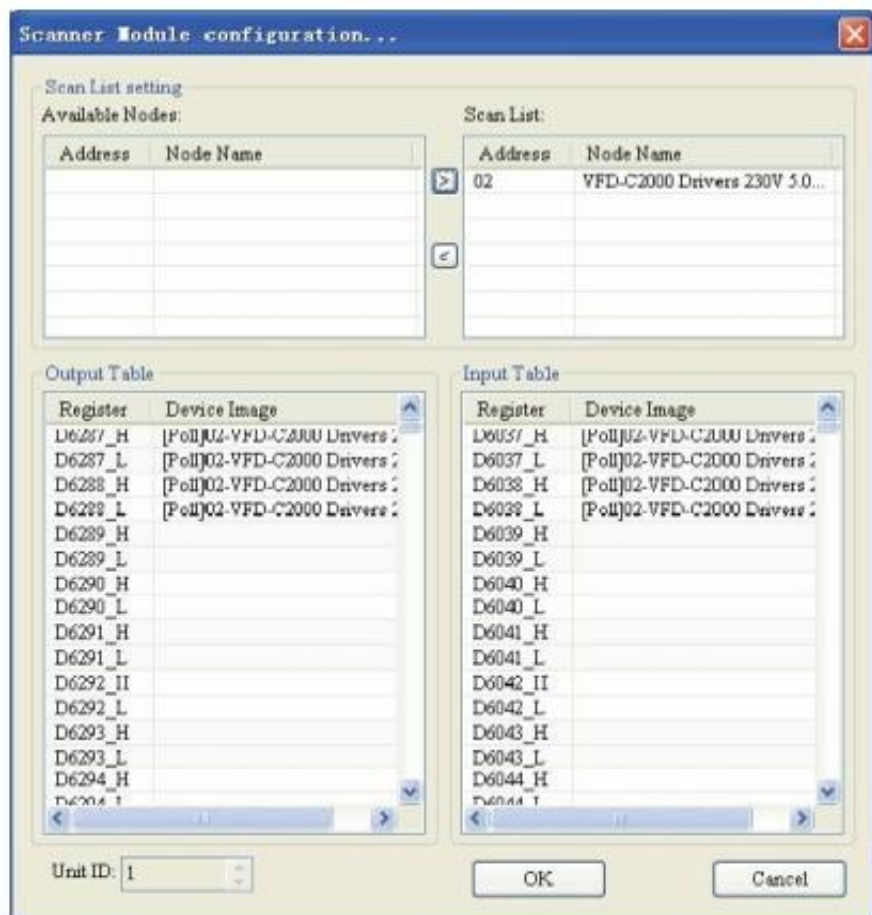
13. بعد از اینکه دانلود تنظیمات به اینورتر تکمیل شد، یک بار تغذیه‌ی اینورتر **VFD-C2000** را قطع کرده و سپس وصل نمایید.

پیکربندی ماژول DVPDNET-SL scanner

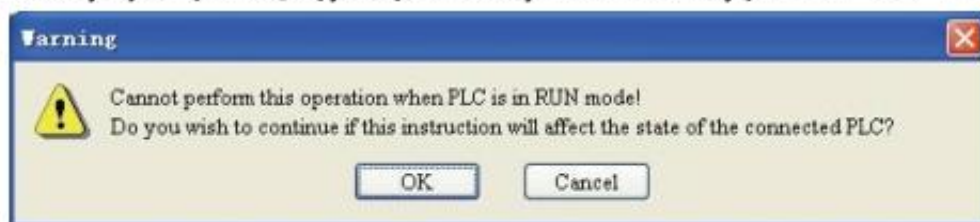
1. برای باز کردن کادر محاوره‌ای **Scanner Module Configuration** بر روی **DNET Scanner (Node 01)** دابل کلیک نمایید. در سمت چپ جدول می‌توان **Node** (گره‌ی) مربوط به **VFD-C2000 Drives** را مشاهده کرد. در سمت راست نرم‌افزار، جدول **Scan List** خالی است.



2. VFD-C2000 Drives (DeviceNet Slave) را باید به جدول Scan List اضافه کنید. برای این کار در سمت چپ نرم افزار Node مربوط به VFD-C2000 Drives را انتخاب کرده و سپس بر روی دکمه‌ی  کلیک نمایید.



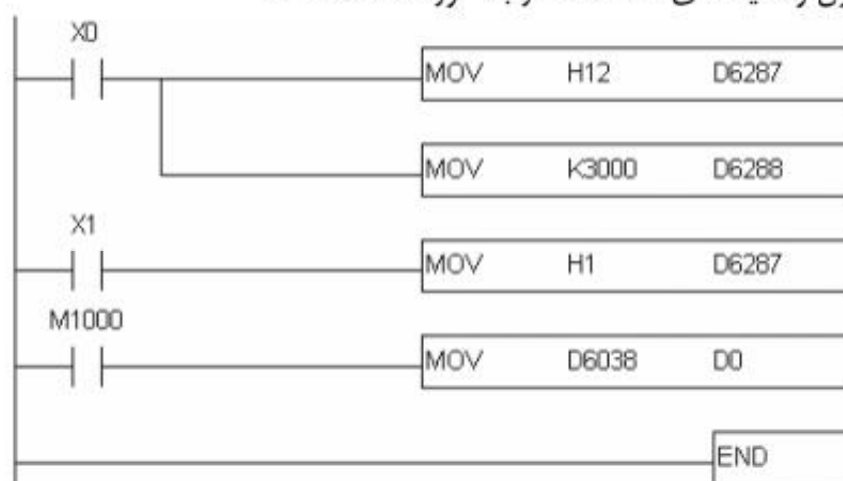
3. از صحیح بودن تنظیمات اطمینان حاصل نمایید. سپس برای دانلود پیکربندی انجام شده به ماژول DVPDNET-SL بر روی دکمه‌ی OK کلیک نمایید. اگر در هنگام دانلود کردن تنظیمات به ماژول، DVP-SV PLC در وضعیت RUN بود، یک کادر محاوره‌ای هشدار ظاهر خواهد شد.



4. برای اینکه PLC در هنگام دانلود تنظیمات به ماژول در وضعیت RUN باقی بماند، باید بر روی دکمه‌ی OK کلیک نمایید. حال باید نمایشگرهای وضعیت MS LED و NS LED را بر روی ماژول CMC-DN01 بررسی کرده و از روشن بودن آنها به رنگ سبز اطمینان حاصل نمایید. به مسيردهی دیتا در حافظه Data Mapping گفته می‌شود. بعد از اینکه شبکه‌ی DeviceNet را مطابق با مراحل گفته شده در بالا پیکربندی کردید، باید مسيردهی دیتا در حافظه را برای PLC مطابق با جدول زیر انجام دهید.

DVPDNET-SL		VFD-C2000 series AC motor Drive
D6287	➡	H2000
D6288		H2001
D6037	➡	H2101
D6038		H2103

دیتای ورودی خروجی شامل، Control word, Status word, Given frequency, Output frequency اینورتر می‌باشند. بنابراین، از طریق دیتای ورودی خروجی اینورتر می‌توان وضعیت Run/Stop, Forward Running, Reverse Running, Running speed اینورتر را توسط یک برنامه به زبان LD در PLC کنترل کرد. در شکل زیر، مثالی از یک برنامه نردبانی برای کنترل وضعیت‌های گفته شده در بالا آورده شده است:



در خط اول وقتی که ورودی X0 فعال (ON) شود، فرمان Forward Running اینورتر VFD-C2000 را فعال خواهد کرد.

در خط دوم وقتی که ورودی X0 فعال (ON) شود، اینورتر VFD-C2000 با فرکانس 30 Hz شروع به کار خواهد کرد.

در خط سوم وقتی که ورودی X1 فعال (ON) شود، فرمان Stop اینورتر VFD-C2000 را غیرفعال خواهد کرد.

در خط چهارم وقتی که ورودی X1 فعال (ON) شود، فرکانس خروجی اینورتر VFD-C2000 در رجیستر D0 از PLC ذخیره خواهد شد.

8- 7- مثالی از تقویم PLC داخلی اینورتر

PLC داخلی اینورتر از توابع تقویم پشتیبانی می‌کند، ولی از این تابع تنها زمانی می‌توان استفاده کرد که کی‌پد KPC-CC01 به اینورتر متصل باشد. در غیر این صورت، نمی‌توان از سالنامه‌ی PLC داخلی اینورتر استفاده کرد. دستورالعمل‌های وابسته به سالنامه شامل دستورالعمل TCMP (مقایسه‌ی مقادیر سالنامه)، TZCP (تنظیم محدوده‌ای برای مقایسه‌ی سالنامه)، TADD (اضافه کردن به تاریخ سالنامه)، TSUB (کم کردن از زمان یا تاریخ سالنامه) و TRD (خواندن تاریخ یا همان سالنامه) می‌باشد.

در کاربردهای واقعی، PLC داخلی اینورتر توانایی تشخیص فعال شدن توابع سالنامه را دارد. اگر توابع سالنامه فعال شده باشند، کدهای هشدار تقویم فعال خواهند شد. مقدار زمان و تاریخ جاری سالنامه در رجیسترهای خاص D1063~D1069 قرار دارد. بنابراین، برای اطلاع از مقدار زمان و تاریخ جاری سیستم باید مقدار این رجیسترها را خواند. در جدول زیر، تعریف و کاربرد هر یک از این رجیسترها لیست شده است.

Special	Item		Attributes
D1063	Year (Western)	20xx (2000~2099)	RO
D1064	Weeks	1~7	RO
D1065	Month	1~12	RO
D1066	Day	1~31	RO
D1067	Hour	0~23	RO
D1068	Minute	0~59	RO
D1069	Second	0~59	RO

در جدول زیر، فلگ‌های خاص وابسته به سالنامه (تقویم) در PLC داخلی اینورتر لیست شده است:

فلگ M خاص	توضیحات	خاصیت فلگ
M1068	خطای زمان تقویم (سالنامه)	فقط خواندنی RO
M1076	خطای زمان تقویم (سالنامه) یا وقفه در به‌روزرسانی تقویم	فقط خواندنی RO
M1036	نادیده گرفتن هشدار تقویم	خواندنی و نوشتنی RW

وقتی که یک برنامه را با استفاده از دستورالعمل‌های TCMP, TZCP, TADD, TSUB می‌نویسید، اگر یک مقدار از مقدار مجاز تجاوز کند، فلگ خاص M1026 فعال (ON) خواهد شد. وقتی که بر روی کی‌پد PLra (هشدار تصحیح زمان RTC) یا PLrt (هشدار وقفه در زمان RTC) رخ دهد، فلگ M1076 فعال خواهد شد. وقتی که فلگ خاص M1036 فعال (ON) شود، PLC از هشدار تقویم صرف‌نظر خواهد کرد.

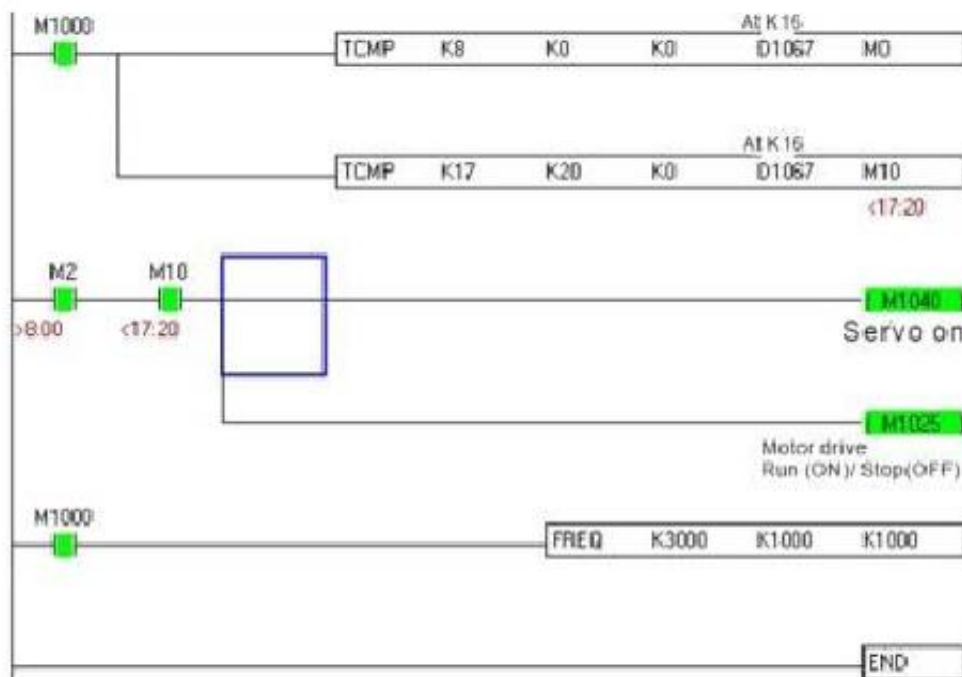
وقتی که توابع سالنامه‌ی PLC در حال اجرا باشند، اگر کی‌پد از اینورتر جدا شده و کی‌پد دیگری جایگزین آن شود، هشدار PLra فعال خواهد شد.

وقتی که تغذیه‌ی کی‌پد بیشتر از 7 روز قطع باشد یا کی‌پد بیشتر از 6 دقیقه از اینورتر جدا شود، مقدار زمان تنظیم شده برای تقویم اینورتر پاک شده و بر روی کی‌پد آلارم PLra راه‌اندازی خواهد شد. وقتی که بیشتر از 10 ثانیه ارتباط کی‌پد با اینورتر قطع شود و یا اینکه زمان راه‌اندازی بیشتر از 10 ثانیه شود، هشدار PLrt فعال خواهد شد.

برای استفاده از اینورتر و همچنین، استفاده از توابع تقویم در PLC داخلی اینورتر، ابتدا زمان صحیح را بر روی کی‌پد تنظیم نمایید. دکمه Enter بر روی کی‌پد را فشار دهید تا وارد منوی آن شوید. آیتم 9.Time Setup را انتخاب کرده و وارد صفحه تنظیمات تقویم شوید. سپس زمان جاری سیستم را مطابق زمان حاضر تنظیم نمایید.



در مثال زیر، ما فرمان 8:00-17:20 را بعنوان یک دوره‌ی زمانی برای روشن کردن اینورتر تنظیم کرده‌ایم.



8-8 مثال‌هایی از دستورات اختصاصی PLC داخلی اینورتر VFD-C2000

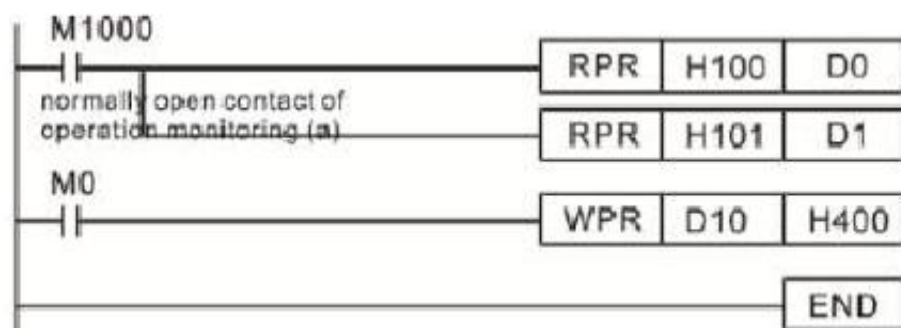
8-8-1 مثال از نوشتن مقدار از PLC داخلی اینورتر به CPU اینورتر

از دستورالعمل WPR برای نوشتن مقدار از PLC داخلی اینورتر به پارامترهای اینورتر استفاده می‌شود. این دستورالعمل دارای 2 عملوند است. در عملوند S1 آدرس دیتا رجیستری که باید محتوای آن داخل پارامتر نوشته شود و یا عدد ثابتی که باید داخل پارامتر نوشته شود، قرار می‌گیرد و در عملوند S2 آدرس پارامتری را که می‌خواهیم مقداری را از PLC داخلی اینورتر در آن پارامتر بنویسیم را قرار می‌دهیم.

در این مثال، با RUN شدن PLC داخلی اینورتر مقدار پارامتر H0100 خوانده شده و در دیتا رجیستر D0 ذخیره می‌شود. همچنین، مقدار پارامتر H0101 از اینورتر خوانده شده و در دیتا رجیستر D1 ذخیره می‌شود.

وقتی که فلگ M0=ON شود، محتوای دیتا رجیستر D10 در پارامتر H0400 اینورتر (نخستین سرعت دیجیتال چند سرعتی) نوشته می‌شود.

وقتی که عمل نوشتن مقدار در پارامترها با موفقیت انجام شود، فلگ M1017=ON خواهد شد. دقت داشته باشید که دستورالعمل WPR از نوشتن مقدار در پارامترهایی با آدرس $H20 \times x$ پشتیبانی نمی‌کند. اما دستورالعمل RPR از خواندن مقدار پارامترها با آدرس $H21 \times x$, $H22 \times x$ پشتیبانی می‌کند.



پیشنهاد می‌شود که از دستورالعمل WPR تنها در مواقع نیاز استفاده نمایید، زیرا تنها 10^9 بار می‌توان عمل نوشتن و تغییر پارامترها را انجام داد. اگر مقدار تغییرات پارامترها از 10^9 بار بیشتر شود، یک خطایی مبنی بر عدم نوشتن مقدار در حافظه رخ خواهد داد.

برخی از پارامترها در اینورتر وجود دارد که محدودیت در نوشتن مقدار در داخل آنها وجود ندارد. این پارامترها را می‌توان به دفعات نامحدود تغییر داد بدون اینکه خطای عدم نوشتن مقدار در حافظه رخ دهد. در زیر، این پارامترها لیست شده‌اند:

P00-10: Control method
 P00-11: Speed mode selection
 P00-12: P2P position mode
 P00-13: Torque mode select
 P00-27: User-defined value
 P01-12: Acceleration time 1
 P01-13: Deceleration time 1
 P01-14: Acceleration time 2
 P01-15: Deceleration time 2
 P01-16: Acceleration time 3
 P01-17: Deceleration time 3
 P01-18: Acceleration time 4
 P01-19: Deceleration time 4

P02-12: Select MI Conversion Time mode
 P02-18: Select MO Conversion Time mode

P04-50 ~ P04-69: PLC register parameter 0 – 19

P08-04: Upper limit of integral
 P08-05: PID output upper limit

P10-17: Electronic gear A
 P10-18: Electronic gear B

P11-34: Torque command
 P11-43: P2P highest frequency
 P11-44: Position control acceleration time
 P11-45: Position control deceleration time

وقتی که یک برنامه برای PLC می‌نویسید، اگر در استفاده از دستورالعمل WPR تردید دارید پیشنهاد می‌شود که از دستورالعمل WPRP استفاده ننمایید.

2-8-8 مثالی از راه‌اندازی مد کنترل PID اینورتر توسط PLC داخلی اینورتر

از دستورالعمل FPID برای راه‌اندازی مد کنترل PID اینورتر در PLC داخلی اینورتر استفاده می‌شود. این دستورالعمل متشکل از 4 عملوند است. در عملوند S1 باید ترمینال ورودی برای قرار دادن

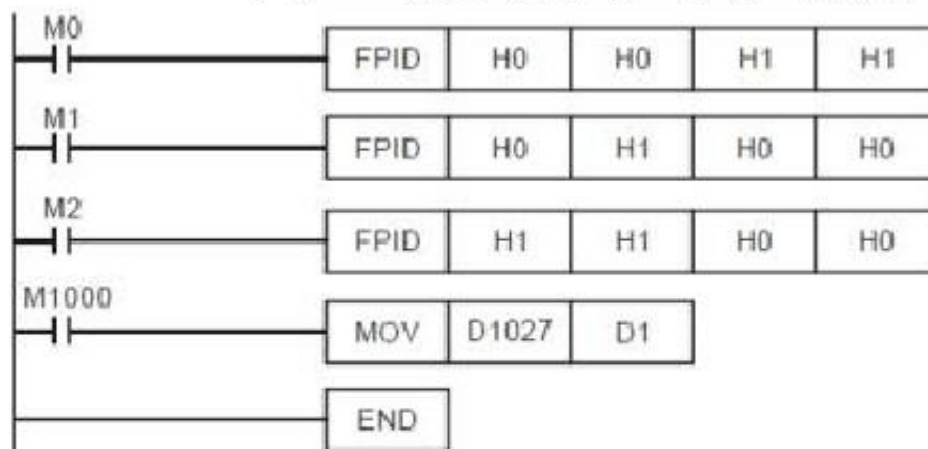
مقدار آن بعنوان مقدار رفرنس PID انتخاب شود. در عملوند S2 ضریب تناسبی تابع PID تنظیم می‌شود. در عملوند S3 زمان انتگرال‌گیری تابع PID و در عملوند S4 زمان مشتق‌گیری تابع PID تنظیم می‌شود.

دستورالعمل FPID را می‌توان مستقیماً توسط پارامتر Pr.08-00 (انتخاب ترمینال ورودی به عنوان مقدار رفرنس یا همان فیدبک PID)، پارامتر Pr.08-01 (ضریب تقویت تناسبی تابع PID)، پارامتر Pr.08-02 (زمان انتگرال‌گیری تابع PID) و پارامتر Pr.08-03 (زمان مشتق‌گیری تابع PID) کنترل کرد. در این حالت، نیاز به برنامه‌نویسی PLC داخلی اینورتر نیست.

در این مثال، وقتی که ورودی X0=ON شود، ترمینال ورودی مربوط به فیدبک یا همان رفرنس PID برابر با 0 انتخاب می‌شود ضریب تناسبی (P) تابع PID برابر با 0 (واحد ضریب تناسبی 0.01 است)، زمان انتگرال‌گیری (I) برابر با 1 (واحد زمان انتگرال‌گیری 0.01 ثانیه است) و زمان مشتق‌گیری (D) برابر با 1 (واحد زمان مشتق‌گیری 0.01 است) تنظیم خواهد شد.

وقتی که فلگ M1=ON شود، ترمینال ورودی رفرنس PID برابر با 0 انتخاب می‌شود. ضریب تناسبی (P) برابر با 1، زمان انتگرال‌گیری (I) برابر با 0 و زمان مشتق‌گیری (D) برابر با 0 تنظیم خواهد شد.

وقتی که فلگ M2=On شود، ترمینال ورودی رفرنس PID برابر با 1 انتخاب می‌شود. در این صورت فرکانس ورودی از طریق کی‌پد کنترل خواهد شد. ضریب تناسبی (P) برابر با 1، زمان انتگرال‌گیری (I) برابر با 0 و زمان مشتق‌گیری (D) برابر با 0 تنظیم خواهد شد.



8-3-8 مثالی از راه‌اندازی مد کنترل سرعت اینورتر توسط PLC داخلی اینورتر

از دستورالعمل FREQ برای راه‌اندازی مد کنترل سرعت اینورتر استفاده می‌شود. این دستورالعمل دارای 3 عملوند است. در عملوند S1 منبع فرکانس تنظیم می‌شود. در عملوند S2 زمان شتاب‌گیری

تنظیم شده و در عملکرد S3 زمان کاهش سرعت تنظیم می‌شود. برای تنظیم شتاب‌گیری و کاهش شتاب به پارامتر Pr.01-45 اینورتر رجوع نمایید.

به عنوان مثال، وقتی که مقدار پارامتر Pr.01-45=0 تنظیم شده باشد، واحد تنظیم شتاب‌گیری و کاهش شتاب 0.01 ثانیه خواهد شد. پس اگر در عملکرد S2 مقدار 50 (به فرمت دسیمال) را قرار دهید، زمان شتاب‌گیری برابر با 0/5 ثانیه خواهد شد و اگر در عملکرد S3 مقدار 60 (به فرمت دسیمال) را قرار دهید، زمان کاهش شتاب برابر با 0/6 ثانیه خواهد شد.

توسط دستورالعمل FREQ می‌توان فرامین فرکانس اینورتر را کنترل کرد. به همراه این دستورالعمل می‌توان از رجیسترها و فلگ‌های خاص لیست شده در زیر استفاده کرد:

M1026: Control driver operating direction FWD(Off)/REV(On)

M1040: Control Servo ON/Servo Off.

M1042: Trigger quick stop (ON)/does not trigger quick stop (Off).

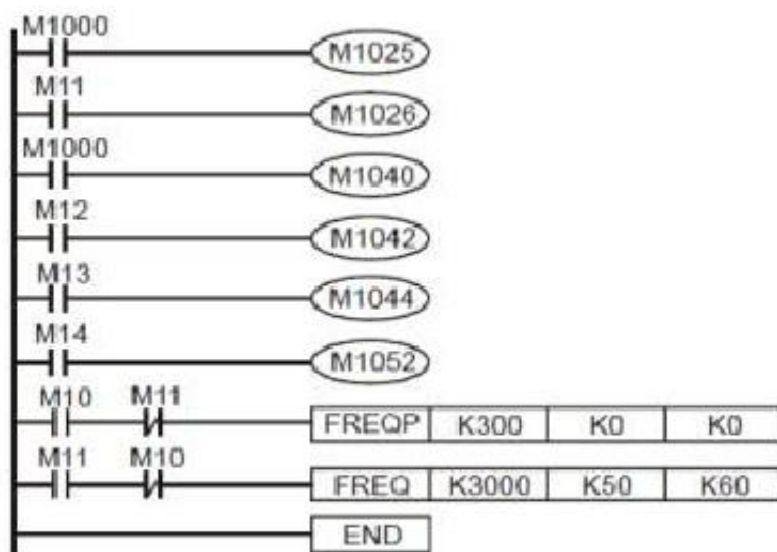
M1044: Pause (On)/release pause (Off)

M1052: Lock frequency (On)/release lock frequency (Off)

از فلگ‌های M1025: Driver RUN(On)/STOP(Off), M1026: driver operating direction FWD(Off)/REV(On), M1015: frequency reached در این مثال استفاده شده است. وقتی که M10=ON شود، فرکانس اینورتر K300(3.00Hz) تنظیم خواهد شد. در همین حین زمان شتاب‌گیری و کاهش شتاب 0 ثانیه خواهد بود.

وقتی که فلگ M11=ON شود، فرکانس اینورتر برابر با K3000(30.00 Hz) شده و زمان شتاب‌گیری برابر با 50(0.5sec) و زمان کاهش شتاب 60(0.6 sec) خواهد شد. این در صورتی است که پارامتر Pr.01-45=0 باشد.

وقتی که فلگ M11=OFF شود، فرکانس خروجی اینورتر برابر با 0 خواهد شد.

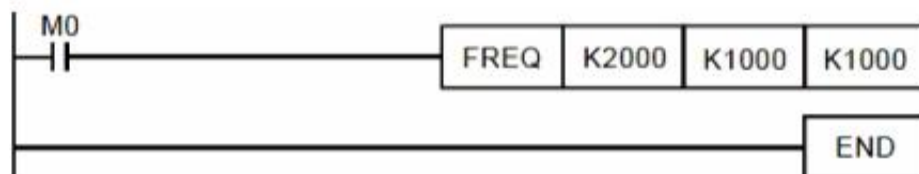


از پارامتر Pr.09-33 برای انتخاب روش پاک کردن فرامین فرکانس قبل از عملکرد PLC استفاده می‌شود.

بیت صفر از پارامتر Pr.09-33 ← در ابتدای فرآیند سیکل اسکن PLC، فرکانس هدف برابر با صفر می‌شود. یعنی وقتی که PLC روشن شود، این بیت فرمان FREQ را برابر با صفر خواهد کرد. بیت یک از پارامتر Pr.09-33 ← در ابتدای فرآیند سیکل اسکن PLC، گشتاور هدف برابر با صفر خواهد شد. یعنی وقتی که PLC روشن شود، این بیت فرمان TORQ را برابر با صفر خواهد کرد.

بیت دو از پارامتر Pr.09-33 ← در ابتدای فرآیند سیکل اسکن PLC، محدوده‌ی سرعت در مد کنترل گشتاور برابر با صفر می‌شود. یعنی وقتی که PLC روشن شود، این بیت فرمان TORQ را برابر با صفر خواهد کرد.

به عنوان مثال، وقتی که از دستورالعمل FREQ برای نوشتن یک برنامه استفاده می‌کنید، این فرآیند صورت می‌گیرد.



اگر ما فلگ M0 را به اجبار 1 کنیم، فرکانس فرمان برابر با 20.00Hz خواهد شد، اما وقتی که فلگ M0 را 0 نماییم، باید یک تغییر در وضعیت اینورتر ایجاد کنیم. برای اعمال این تغییر باید از پارامتر Pr.09-33 استفاده نماییم.

وقتی که بیت 0 از پارامتر Pr.09-33 برابر با 0 تنظیم شده باشد، با صفر شدن فلگ M0، فرکانس فرمان در وضعیت 20.00 Hz باقی خواهد ماند. این به آن دلیل مهم است که وقتی که سیکل اسکن برنامه مجدداً شروع شود، فرکانس به صفر بر نخواهد گشت.

وقتی که بیت 0 از پارامتر Pr.09-33 برابر با 1 تنظیم شده باشد، با صفر شدن فلگ M0، فرکانس فرمان برابر با 0.00Hz خواهد شد. دلیل این مهم این است که وقتی که سیکل اسکن برنامه مجدداً شروع شود، فرکانس به 0 بازخواهد گشت.

4-8-8 مثالی از به کارگیری اینورتر در مد کنترل سرعت توسط PLC داخلی

اینورتر

PLC داخلی اینورتر مدهای کنترلی سرعت، گشتاور، جایگاه‌یابی و موقعیت‌یابی را پشتیبانی می‌کند. مد کنترل سرعت، مد کنترل گشتاور و مد کنترل موقعیت بر مبنای کنترل برداری FOC کنترل می‌شوند. بنابراین، برای انجام کنترل دقیق و موفقیت آمیز باید پارامترهای الکتروموتور را به دقت بررسی کرده و در اینورتر تنظیم نمایید تا بتوانید به مد کنترل گشتاور، مد کنترل سرعت و مد کنترل موقعیت مبتنی بر FOC با دقت بالا دست یابید.

بعلاوه، الکتروموتورها به دو نوع دسته‌بندی می‌شوند. الکتروموتورهای IM و الکتروموتورهای PM. بنابراین، باید ابتدا پارامترهای الکتروموتور IM را به دقت بررسی و مطالعه نمایید. برای الکتروموتورهای PM، بعد از به پایان رسیدن بررسی و مطالعه پارامترهای الکتروموتور، باید به طور کامل مبدأ زاویه انحراف (خطای) موتور را نیز بررسی و مطالعه نمایید. برای اطلاع از جزئیات بیشتر پارامترهای Pr.05-00 را مطالعه نمایید.

اگر از یک الکتروموتور PM سری ECMA دلتا استفاده می‌کنید، پارامترهای موتور را می‌توانید مستقیماً از کاتالوگ سروموتور مطالعه کرده و تنظیم نمایید. در این صورت، نیاز به بررسی و مطالعه پارامترهای موتور نمی‌باشد.

در PLC داخلی اینورتر تعدادی فلگ خاص، رجیستر خاص برای لینک کردن PLC با رجیسترهای CPU اینورتر در نظر گرفته شده است. این فلگ‌ها و رجیسترهای خاص عمل برنامه‌نویسی را بسیار ساده کرده و نیاز ما را در کنترل ساده‌ی اینورتر توسط PLC داخلی اینورتر مرتفع ساخته‌اند.

در جدول زیر، فلگ‌های خاص M که کار کنترلی انجام می‌دهند لیست شده‌اند:

Special M	Description of Function	Attributes
M1025	Driver frequency = set frequency (ON)/driver frequency = 0 (OFF)	RW
M1026	Driver operating direction FWD(OFF)/REV(ON)	RW
M1040	Hardware power (Servo ON)	RW
M1042	Quick stop	RW
M1044	Pause (Halt)	RW
M1052	Lock frequency (lock, frequency locked at the current operating frequency)	RW

در جدول زیر، فلگ‌های خاص M که از آنها برای مانیتور کردن وضعیت اینورتر استفاده می‌شود لیست شده‌اند:

Special M	Description of Function	Attributes
M1015	Frequency attained (when used together with M1025)	RO
M1056	Servo ON Ready	RO
M1058	On Quick Stopping	RO

در جدول زیر، رجیستر خاص D که از آن برای تنظیم مد کنترل توسط PLC داخلی استفاده می‌شود آورده شده است:

Special D	Description of Function	Attributes
D1060	Mode setting (speed mode is 0)	RW

در جدول زیر، رجیسترهای خاص D که از آنها برای مانیتور کردن وضعیت اینورتر استفاده می‌شود لیست شده‌اند:

Special D	Description of Function	Attributes
D1037	Converter output frequency (0.00~600.00)	RO
D1050	Actual operating mode (speed mode is 0)	RO

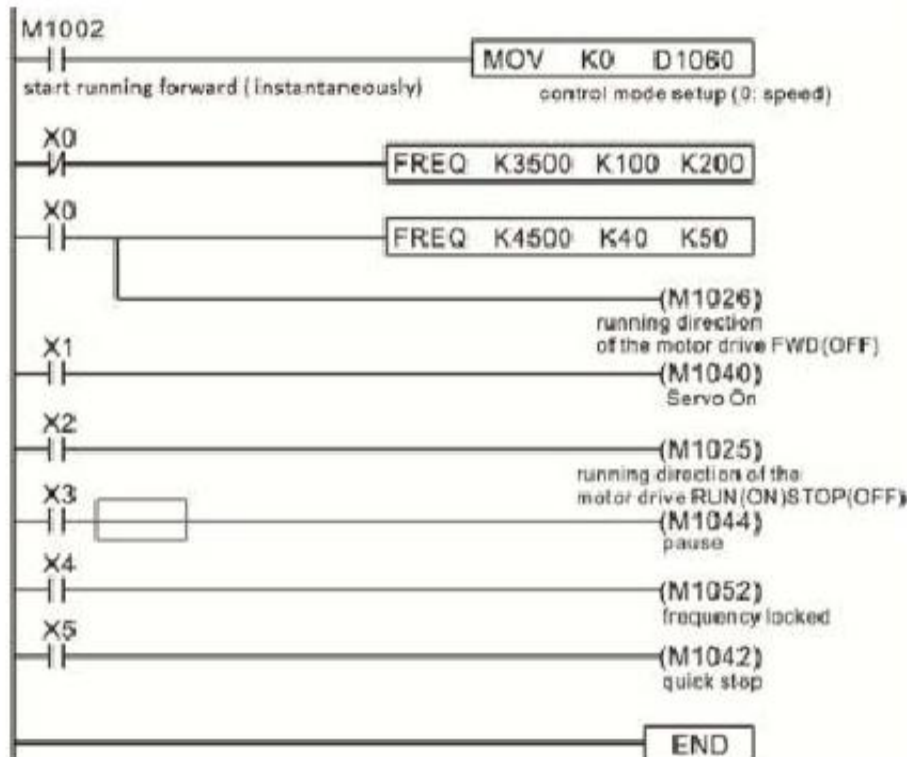
در PLC داخلی اینورتر VFD-C2000 دستورالعملی کاملاً اختصاصی به نام $FREQ(P)$ وجود دارد. از این دستورالعمل برای کنترل مد سرعت اینورتر استفاده می‌شود. این دستورالعمل دارای 3 عملوند است. در عملوند S1 سرعت هدف (مدنظر) تنظیم خواهد شد. در عملوند S2 مقدار نخستین زمان شتاب‌گیری (شتاب مثبت) و در عملوند S3 مقدار نخستین زمان کاهش سرعت (شتاب منفی) تنظیم می‌شود.

در ادامه، مثالی از برنامه‌نویسی مد کنترل سرعت برای PLC داخلی اینورتر با دستورالعمل اختصاصی $FREQ$ آورده شده است:

قبل از اجرای مد کنترل سرعت، باید روش کنترل FOC (magnetic field orientation) را تنظیم کرد، که برای این کار ابتدا باید پارامترهای الکترومکانیکی را به طور کامل تنظیم نمایید.
- برای تنظیم مد کنترل سرعت باید پارامتر (رجیستر خاص) D1060 را برابر با 0 تنظیم نمایید.

- برای کنترل فرکانس خروجی اینورتر از دستورالعمل **FREQ** استفاده نمایید. این دستورالعمل علاوه بر کنترل فرکانس خروجی اینورتر، زمان نخستین شتاب‌گیری و کاهش سرعت را نیز تنظیم می‌نماید.
- برای فعال کردن اینورتر باید فلگ خاص **M1040** را برابر با 1 تنظیم نمایید. در این صورت، اینورتر **RUN** شده ولی فرکانس خروجی آن برابر با صفر خواهد بود.
- برای فعال کردن فرکانس خروجی اینورتر باید فلگ خاص **M1025** را برابر با 1 تنظیم نمایید. در این صورت اینورتر به فرکانس تنظیم شده در عملوند **S1** دستورالعمل **FREQ** پرش کرده و آن را با زمان شتاب‌گیری تنظیم شده در عملوند **S2** ی‌همین دستورالعمل به خروجی ارسال می‌کند.
- از فلگ خاص **M1052** برای قفل کردن فرکانس جاری اینورتر استفاده می‌شود.
- از فلگ خاص **M1044** برای متوقف کردن موقتی عملکرد اینورتر با اعمال زمان کاهش سرعت تنظیم شده استفاده می‌شود.
- از فلگ خاص **M1042** می‌توان برای اجرای توقف سریع اینورتر استفاده کرد. در عمل توقف سریع ممکن است یک خطا رخ دهد. اگر بار متصل شده به موتور خیلی بزرگ باشد، ممکن است یک خطای دیگر نیز رخ دهد.

- فلگ‌های خاص کنترلی مورد استفاده در این مثال فلگ فعال کردن اینورتر (**Servo ON**) **M1040**، فلگ توقف سریع (**Quick Stop**) **M1042**، فلگ توقف موقتی با اعمال زمان کاهش سرعت (**Halt**) **M1044** و فلگ خاص قفل کردن عملکرد اینورتر (**LOCK**) **M1052** می‌باشند.



8-5-8 مثالی از راه‌اندازی مد کنترل گشتاور اینورتر توسط PLC داخلی اینورتر

از دستورالعمل TORQ برای راه‌اندازی مد کنترل گشتاور اینورتر استفاده می‌شود. این دستورالعمل دارای 2 عملوند است. در عملوند S1 فرمان گشتاور نوشته می‌شود. مقدار این عملوند یک رقمی می‌باشد و در عملوند S2 محدوده‌ی سرعت تنظیم می‌شود. بنابراین توسط دستورالعمل TORQ می‌توان فرکانس گشتاور اعمالی و در یک محدوده‌ی سرعت معین را کنترل کرد. به همراه این دستورالعمل باید از رجیسترهای خاص لیست شده در زیر استفاده کرد.

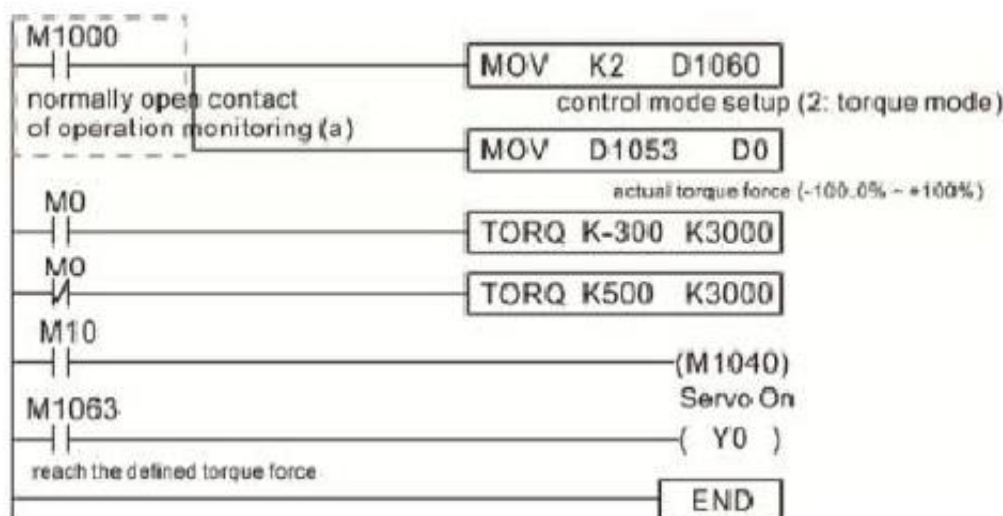
فلگ خاص M1040 : از این فلگ برای کنترل فرمان Servo ON/ Servo OFF استفاده می‌شود. وقتی که سیگنال Servo در وضعیت ON است، اگر دستورالعمل TORQ اجرا شود، گشتاور خروجی اعمال شده به موتور از طریق دستورالعمل TORQ تعیین شده و فرکانس خروجی توسط دستورالعمل TORQ کنترل شده و محدود خواهد شد.

در این مثال، توسط فلگ خاص M1040 می‌توانیم فرمان Servo ON/ Servo OFF را به اینورتر اعمال کرده و آن را کنترل نماییم. همچنین، توسط فلگ خاص M1063 می‌توان منبع تنظیم گشتاور را انتخاب کرد. توسط رجیستر D1060 می‌توان مد کنترلی را انتخاب کرده و توسط رجیستر خاص D1053 می‌توان گشتاور واقعی را خواند.

وقتی که فلگ M0=OFF باشد، فرمان گشتاور راه‌اندازی برابر با $K+500(+50.0\%)$ تنظیم شده و سرعت چرخش موتور به $3000(30\text{Hz})$ محدود می‌شود.

وقتی که فلگ M0=ON باشد، فرمان گشتاور راه‌اندازی برابر با $K-300(-30.0\%)$ تنظیم شده و سرعت چرخش موتور به $3000(30\text{Hz})$ محدود می‌شود.

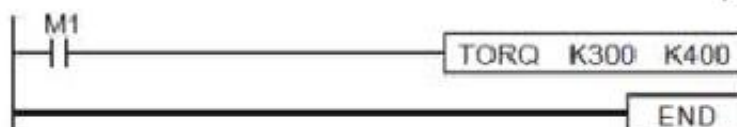
وقتی که فلگ M10=ON شود، به دلیل فعال شدن تابع Servo ON، دستورالعمل گشتاور خروجی راه‌اندازی می‌شود. وقتی که گشتاور خروجی به گشتاور تنظیم شده رسید، فلگ خاص M1063 فعال خواهد شد. دقت داشته باشید که این فلگ به طور پیوسته ON و OFF خواهد شد. دلیل این مهم آن است که گشتاور خروجی هیچ گاه به طور پیوسته با گشتاور تنظیمی برابر نخواهد بود.



از پارامتر Pr.09-33 برای تعریف مرجع پاک کردن فرکانس قبل از عملکرد PLC استفاده می‌شود.

بیت صفر از پارامتر Pr.09-33 ← در ابتدای فرآیند سیکل اسکن PLC، فرکانس هدف برابر با صفر می‌شود. یعنی وقتی که PLC روشن شود، این بیت فرمان **FREQ** را برابر با صفر خواهد کرد. بیت یک از پارامتر Pr.09-33 ← در ابتدای فرآیند سیکل اسکن PLC، گشتاور هدف برابر با صفر خواهد شد. یعنی وقتی که PLC روشن شود، این بیت فرمان **TORQ** را برابر با صفر خواهد کرد.

بیت دو از پارامتر Pr.09-33 ← در ابتدای فرآیند سیکل اسکن PLC، محدوده‌ی سرعت در مد کنترل گشتاور برابر با صفر می‌شود. یعنی وقتی که PLC روشن شود، این بیت فرمان **TORQ** را برابر با صفر خواهد کرد. به عنوان مثال داریم:



اگر فلگ M1 را به اجبار 1 نماییم، فرمان گشتاور $K+300(+30\%)$ شده و محدوده‌ی سرعت برابر با $400(40\text{Hz})$ می‌شود. اما وقتی که فلگ M1 برابر با 0 شود، یک تغییر در وضعیت اینورتر باید رخ دهد. برای اعمال این تغییر باید از پارامتر Pr.09-33 استفاده کرد.

وقتی که بیت 1 و بیت 2 پارامتر Pr.09-33 هر دو برابر با 0 منطقی باشند و فلگ M1 در این هنگام 0 شود، فرمان گشتاور در $+30\%$ باقی مانده و سرعت موتور در فرکانس 40Hz محدود خواهد شد.

وقتی که بیت 1 و بیت 2 پارامتر Pr.09-33 هر دو برابر با 1 منطقی باشند و فلگ M1 در این هنگام 0 شود، فرمان گشتاور 0% شده و فرکانس سرعت موتور 0Hz خواهد شد.

8-6-8 مثالی از به کارگیری اینورتر در مد کنترل گشتاور توسط PLC داخلی اینورتر

PLC داخلی اینورتر از مدهای کنترلی سرعت، گشتاور، جایگاه‌یابی و موقعیت‌یابی پشتیبانی می‌کند. مد کنترل سرعت، مد کنترل گشتاور و مد کنترل موقعیت بر مبنای کنترل برداری FOC کنترل می‌شوند. بنابراین، برای انجام کنترل دقیق و موفقیت آمیز باید پارامترهای الکتروموتور را به دقت بررسی کرده و در اینورتر تنظیم نمایید تا بتوانید به مد کنترل گشتاور، مد کنترل سرعت و مد کنترل موقعیت مبتنی بر FOC با دقت بالا دست یابید.

بعلاوه، الکتروموتورها به دو نوع الکتروموتورهای IM و الکتروموتورهای PM دسته‌بندی می‌شوند. بنابراین، باید ابتدا پارامترهای الکتروموتور IM را به دقت بررسی و مطالعه نمایید. برای الکتروموتورهای PM، بعد از به پایان رسیدن بررسی و مطالعه پارامترهای الکتروموتور، باید به طور کامل مبدأ زاویه انحراف (خطای) موتور را نیز بررسی و مطالعه نمایید. برای اطلاع از جزئیات بیشتر پارامترهای Pr.05-00 را مطالعه نمایید.

اگر از یک الکتروموتور PM سری ECMA دلتا استفاده می‌کنید، پارامترهای موتور را می‌توانید مستقیماً از کاتالوگ سروموتور مطالعه کرده و تنظیم نمایید. در این صورت، نیاز به بررسی و مطالعه پارامترهای موتور نمی‌باشد.

ابتدا به بررسی رجیسترها و فلگ‌های خاص مربوط به مد کنترل گشتاور در حافظه‌ی PLC داخلی اینورتر خواهیم پرداخت.

در جدول زیر، فلگ خاص مربوط به کنترل مد گشتاور توسط PLC داخلی اینورتر آورده شده است:

Special M	Description of Function	Attributes
M1040	Servo ON	RW

در جدول زیر، فلگ‌های خاص مربوط به بررسی وضعیت مد کنترل گشتاور توسط PLC داخلی اینورتر آورده شده است:

Special M	Description of Function	Attributes
M1056	Servo ON Ready	RO
M1063	Torque attained	RO

در جدول زیر، رجیستر خاص مربوط به کنترل مد گشتاور توسط PLC داخلی اینورتر آورده شده است:

Special D	Description of Function	Attributes
D1060	Operating mode setting (torque mode is 2)	RW

در جدول زیر، رجیسترهای خاص مربوط به بررسی وضعیت مد گشتاور توسط PLC داخلی اینورتر آورده شده است:

Special D	Description of Function	Attributes
D1050	Actual operating mode (speed mode is 0)	RO
D1053	Actual torque	RO

در PLC داخلی اینورتر VFD-C2000 دستورالعملی کاملاً اختصاصی به نام TORQ(P) وجود دارد. از این دستورالعمل برای کنترل مد گشتاور اینورتر استفاده می‌شود. این دستورالعمل دارای دو عملوند است. در عملوند S1 مقدار گشتاور هدف (مورد نظر) تنظیم می‌شود و در عملوند S2 باید محدودیت فرکانس خروجی را تنظیم کرد تا در هنگام عملکرد فرکانس خروجی از مقدار تنظیم شده در عملوند S2 بالاتر نرود. در ادامه، مثالی از برنامه‌نویسی مد کنترل گشتاور برای PLC داخلی اینورتر VFD-C2000 با دستورالعمل اختصاصی TORQ آورده شده است.

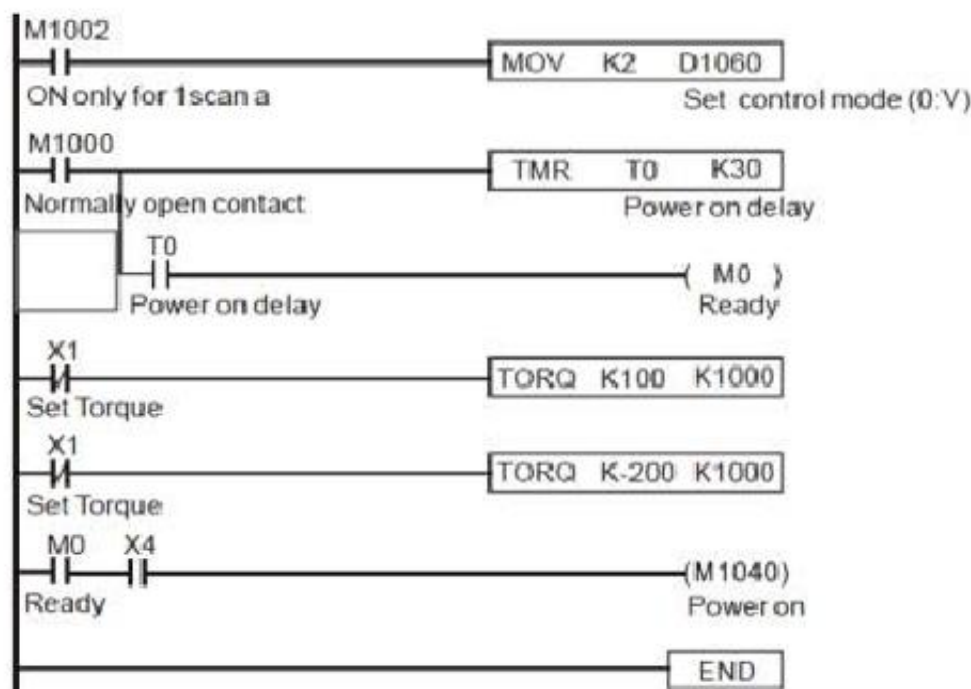
تنظیم پارامترهای الکترومکانیکی موتور برای مد کنترل گشتاور بسیار دشوار است. بنابراین، پارامترهای الکترومکانیکی را باید قبل از اجرای مد کنترل سرعت به دقت بررسی کرده و تنظیم نمایید تا سیستم با بالاترین دقت، عملکرد خود را به درستی انجام دهد. سپس باید تنظیمات PLC داخلی را برای مد کنترل گشتاور انجام دهید.

- در PLC داخلی اینورتر ابتدا باید مد گشتاور را تنظیم نمایید. برای این کار، رجیستر خاص D1060 را برابر با 2 تنظیم نمایید.

- برای اجرای مد کنترل گشتاور و محدود کردن سرعت الکتروموتور از دستورالعمل TORQ استفاده نمایید.

- فلگ خاص M1040 را برابر با 1 تنظیم نمایید. در این حالت، اینورتر RUN شده و سیستم گشتاور هدف را اجرا کرده و سرعت را محدود خواهد کرد.

- توسط رجیستر خاص D1053 می‌توانید مقدار گشتاور واقعی (جاری) سیستم را مشاهده کنید.



8-8-7 مثالی از راه‌اندازی مد کنترل موقعیت نقطه به نقطه اینورتر توسط PLC داخلی اینورتر

از دستورالعمل DPOS برای راه‌اندازی مد کنترل موقعیت نقطه به نقطه استفاده می‌شود. این دستورالعمل دارای یک عملوند است. در عملوند S1 مقدار هدف (مورد نظر) نوشته می‌شود. به طور دقیق‌تر می‌توان گفت که توسط دستورالعمل DPOS می‌توان فرامین موقعیت را کنترل کرد. برای کنترل فرمان موقعیت نیاز به استفاده از برخی رجیسترها و فلگ‌های خاص می‌باشد که در زیر لیست شده‌اند:

M1040 ← فرمان کنترل Servo ON/ Servo OFF

M1055 ← فلگ اعلام وضعیت رسیدن موتور به موقعیت تعیین شده برای مبدأ حرکت (Origin)

M1048 ← فرمان حرکت به موقعیت جدید

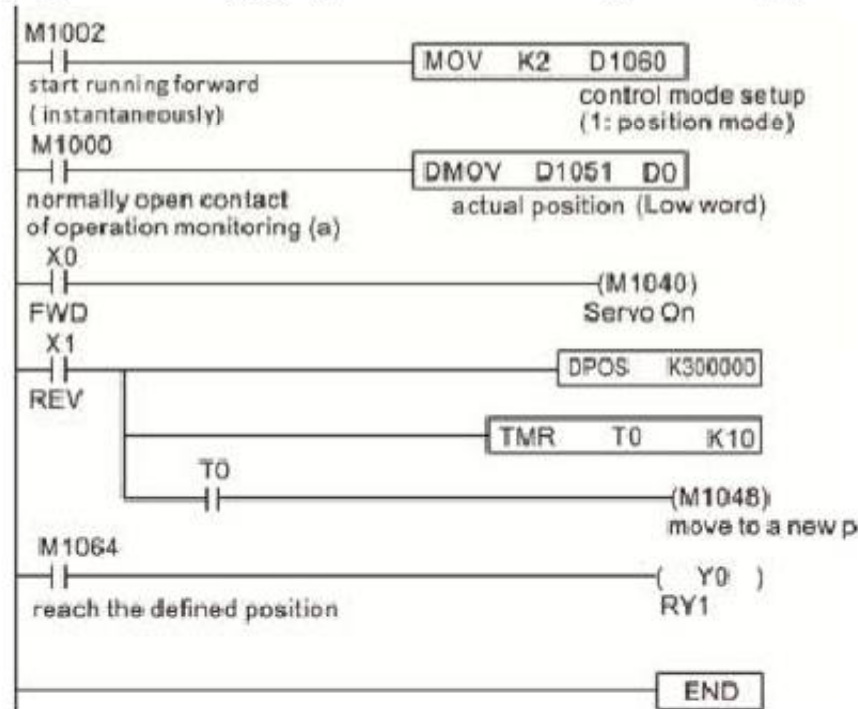
D1060 ← تعیین مد کنترل اینورتر

اگر مد کنترل اینورتر را برابر با مد موقعیت ($D1060=1$) تنظیم کرده باشید و اینورتر در وضعیت Servo ON ($M1040=1$) باشد، با اجرا شدن دستورالعمل DPOS و با یکبار فعال شدن فلگ خاص M1048 موتور به سمت موقعیت جدید حرکت خواهد کرد.

در این مثال، توسط فلگ M1040 فرمان Servo ON را صادر کرده و توسط فلگ M1064 می‌توان از رسیدن موتور به موقعیت تنظیم شده مطلع شد. با نوشتن مقدار 1 در رجیستر خاص D1060 می‌توان مد کنترل را برابر با مد موقعیت انتخاب کرد.

از رجیسترهای خاص D1051(L), D1052(H) برای خواندن موقعیت جاری شفت موتور استفاده می‌شود. وقتی که ورودی X0=ON شود، فلگ خاص M1040 فعال (ON) شده و تابع Servo ON اجرا می‌شود.

وقتی که ورودی X10=ON شود، دستورالعمل DPOS اجرا شده و موقعیت جدید برابر با +300000 تنظیم خواهد شد و هنگامی که فلگ خاص M1048 به وضعیت ON تغییر کند، بعد از یک تأخیر 1 ثانیه‌ای موتور به سمت موقعیت جدید حرکت خواهد کرد. با خواندن مقدار رجیستر خاص D1051 می‌توان تغییر موقعیت لحظه به لحظه‌ی جاری موتور را بررسی کرد. وقتی که موتور به موقعیت تنظیم شده رسید، فلگ خاص M1064=ON شده و خروجی Y0=ON می‌شود.



8-8-8 مثالی از به کارگیری اینورتر در مد کنترل موقعیت و جایگاه‌یابی توسط PLC داخلی اینورتر

PLC داخلی اینورتر از مدهای کنترلی سرعت، گشتاور، جایگاه‌یابی و موقعیت‌یابی پشتیبانی می‌کند. مد کنترل سرعت، مد کنترل گشتاور و مد کنترل موقعیت بر مبنای کنترل برداری FOC کنترل

می‌شوند. بنابراین، برای انجام کنترل دقیق و موفقیت آمیز باید پارامترهای الکتروموتور را به دقت بررسی کرده و در اینورتر تنظیم نمایید تا بتوانید به مد کنترل گشتاور، مد کنترل سرعت و مد کنترل موقعیت مبتنی بر FOC با دقت بالا دست یابید.

بعلاوه، الکتروموتورها به دو نوع الکتروموتورهای IM و الکتروموتورهای PM دسته‌بندی می‌شوند. بنابراین باید ابتدا پارامترهای الکتروموتور IM را به دقت بررسی و مطالعه نمایید. برای الکتروموتورهای PM، بعد از به پایان رسیدن بررسی و مطالعه پارامترهای الکتروموتور، باید به طور کامل مبدأ زاویه انحراف (خطای) موتور را نیز بررسی و مطالعه نمایید. برای اطلاع از جزئیات بیشتر پارامترهای Pr.05-00 را مطالعه نمایید.

اگر از یک الکتروموتور PM سری ECMA دلتا استفاده می‌کنید، پارامترهای موتور را می‌توانید مستقیماً از کاتالوگ سروموتور مطالعه کرده و تنظیم نمایید. در این صورت، نیاز به بررسی و مطالعه پارامترهای موتور نمی‌باشد.

ابتدا به بررسی رجیسترها و فلگ‌های خاص مربوط به مد کنترل جایگاه‌یابی (Homing) و مد کنترل موقعیت‌یابی (Positioning) در حافظه‌ی داخلی PLC خواهیم پرداخت. در جدول زیر، فلگ‌های خاص مربوط به کنترل مد جایگاه‌یابی و مد موقعیت‌یابی توسط PLC داخلی اینورتر آورده شده است:

Special M	Description of Function	Attributes
M1040	Servo ON	RW
M1048	Move to new position, must use control mode as position mode and M1040 = 1 (D1060 = 1)	RW
M1050	Absolute position/relative position (0: relative/1: absolute)	RW
M1055	Search for origin (home start), must use control mode as position mode (D1060 = 3) and M1040 = 1	RW

در جدول زیر، فلگ‌های خاص مربوط به وضعیت مد جایگاه‌یابی و مد موقعیت‌یابی توسط PLC داخلی اینورتر آورده شده است:

Special M	Description of Function	Attributes
M1064	Target reached	RO
M1070	Return home complete	RO
M1071	Homing error	RO

در جدول زیر، رجیستر خاص مربوط به کنترل مد جایگاه‌یابی و مد موقعیت‌یابی توسط PLC داخلی اینورتر آورده شده است:

Special D	Description of Function	Attributes
D1060	Operating mode setting (position mode is 1, homing mode is 3)	RW

در جدول زیر، رجیسترهای خاص مربوط به وضعیت مد جایگاه‌یابی و مد موقعیت‌یابی توسط PLC داخلی اینورتر آورده شده است:

Special D	Description of Function	Attributes
D1050	Actual operating mode (speed mode is 0)	RO
D1051	Actual position (Low word)	RO
D1052	Actual position (High word)	RO

دقت داشته باشید که از ترکیب رجیسترهای D1051, D1052 باید برای بدست آوردن محل جاری شفت الکتروموتور استفاده کرد. یعنی این رجیستر 32 بیتی می‌باشد و باید از دستورالعمل 32 بیتی استفاده شود.

در PLC داخلی اینورتر VFD-C2000 دستورالعملی کاملاً اختصاصی به نام DPOS(P) وجود دارد. از این دستورالعمل برای کنترل مد موقعیت اینورتر استفاده می‌شود. این دستورالعمل دارای یک عملوند است. در عملوند S1 این دستورالعمل مقدار موقعیت هدف (مورد نظر) تنظیم می‌شود. در ادامه، مثالی از برنامه‌نویسی مد کنترل جایگاه‌یابی و مد کنترل موقعیت‌یابی برای PLC داخلی اینورتر با دستورالعمل اختصاصی DPOS آورده شده است:

قبل از اجرای مد کنترل جایگاه‌یابی یا مد کنترل سرعت باید پارامترهای الکترومکانیکی مربوط به موقعیت‌یابی را در اینورتر VFD-C2000 تنظیم نمایید.

- برای انتخاب روش جایگاه‌یابی از پارامتر Pr.00-40 استفاده نمایید. در روش جایگاه‌یابی باید از 2 سنسور محدود کننده حرکت به چپ و راست و یک سنسور مبدأ (نقطه‌ی صفر) به نام Origin استفاده نمایید. برای این سه سنسور ابتدا باید توابع مربوط به ترمینال‌های ورودی دیجیتال (MI) اینورتر را تنظیم نمایید. به یکی از ترمینال‌های ورودی دیجیتال (MI) تابع 44 (محدود کردن حرکت بیش از اندازه به چپ) را اختصاص دهید. به یک ترمینال ورودی دیجیتال (MI) دیگر تابع 45 (محدود کردن حرکت بیش از اندازه به راست توسط سنسور) را اختصاص داده و به یک ترمینال ورودی دیجیتال (MI) دیگر تابع 46 (سنسور مبدأ Origin) را اختصاص دهید. به دلیل اینکه اینورتر VFD-C2000 برای مبدأ‌یابی از فاز Z انکدر پشتیبانی می‌کند، باید فاز Z انکدر به کارت انکدر متصل شود.

- برای تنظیم مد کنترل موقعیت‌یابی باید رجیستر خاص D1060 را برابر با 3 تنظیم نمایید.
 - مقدار فلگ خاص M1040 را برابر با 1 تنظیم نمایید. در مدهای VF/ SVC/ VFPG، سیستم به مد Standby خواهد رفت. توسط پارامتر Pr.01-34 می‌توان به مدهای Standby دسترسی داشت. در مد FOC+PG، سرعت الکتروموتور در سرعت صفر نگه داشته خواهد شد.

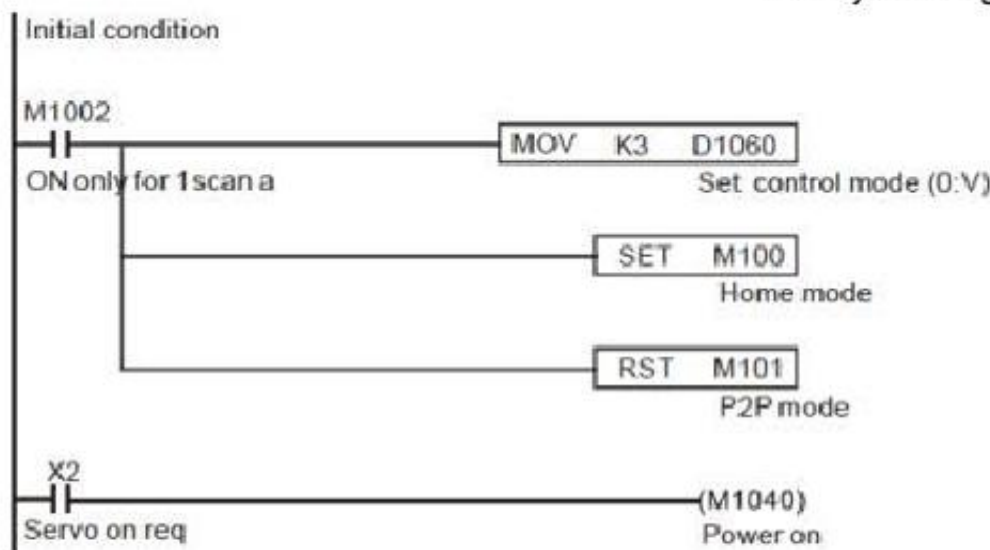
- فلگ خاص M1055 را برابر با 1 تنظیم نمایید. در این صورت اینورتر شروع به جستجوی مبدأ (Origin) خواهد کرد.

- وقتی که عمل جایگاه‌یابی به پایان رسید، فلگ خاص M1070 به وضعیت ON تغییر خواهد کرد. با فعال شدن فلگ M1070 می‌توانید رجیستر خاص D1060 را برابر با 1 تنظیم نمایید تا مد کنترل اینورتر از مد جایگاه‌یابی به مد موقعیت‌یابی تغییر کند. دقت داشته باشید که در این هنگام فلگ خاص M1040 تغییر وضعیت نداده و OFF یا 0 نخواهد شد.

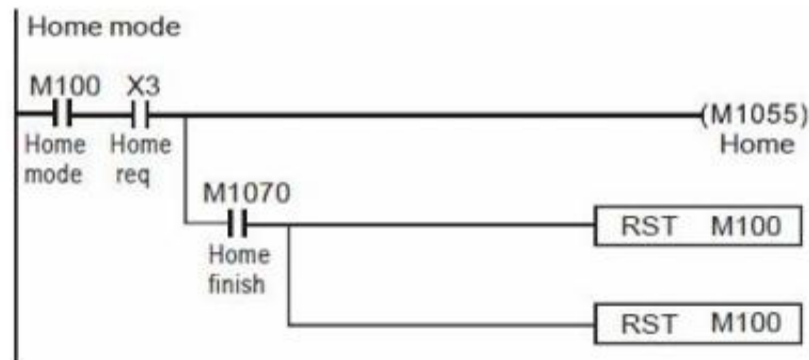
- حال می‌توان با استفاده از دستورالعمل DPOS موقعیت مکانیزم متصل شده به اینورتر را تغییر داده و به موقعیت مورد نظر (مقصد) حرکت داد. از فلگ خاص M1050 یا پارامتر Pr.00-12 می‌توان برای تنظیم نوع موقعیت‌یابی استفاده کرد. موقعیت‌یابی به دو صورت افزایشی (وابسته) Relative position یا مطلق Absolute position صورت می‌گیرد.

- با یک بار ON کردن فلگ خاص M1048 به صورت پالسی (مدت زمان این پالس باید بیشتر از 1 میلی‌ثانیه باشد)، اینورتر شروع به حرکت دادن شفت الکتروموتور به جلو تا موقعیت هدف خواهد کرد. توجه داشته باشید که فلگ خاص M1040 در این هنگام باید 1 باشد تا فرمان شروع حرکت به جلو اجرا شود. موقعیت جاری را می‌توان با استفاده از رجیستر خاص D1051, D1052 خواند.

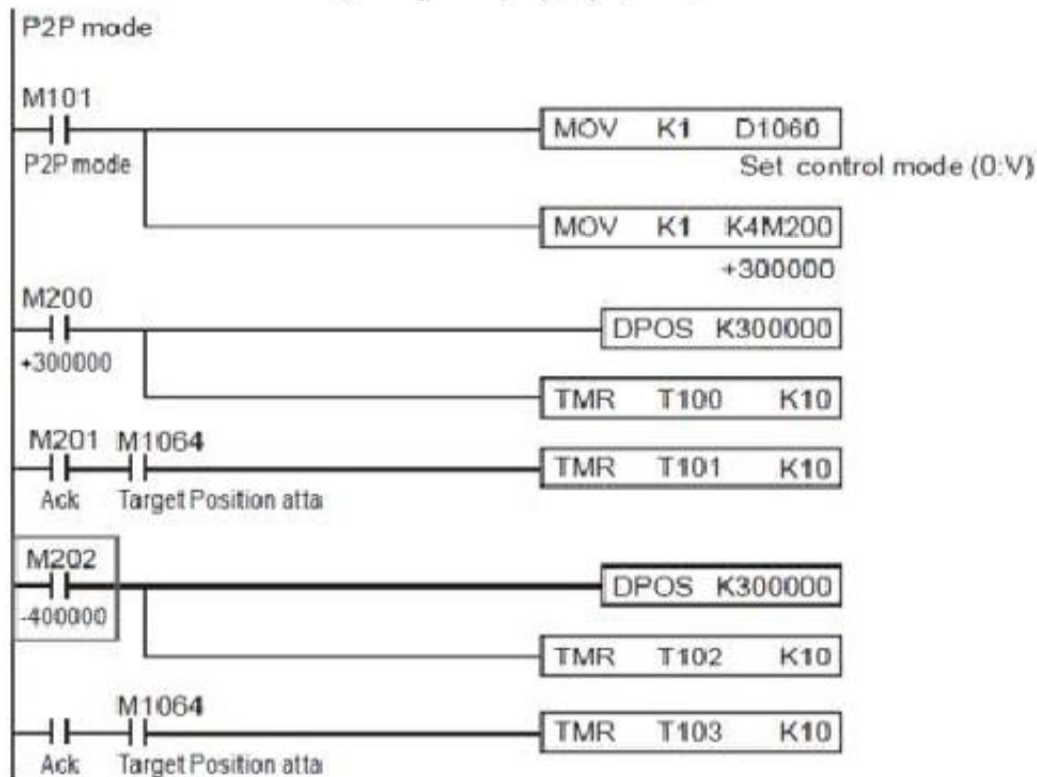
بخش 1 ← در این بخش برنامه مربوط به مد جایگاه‌یابی و مقداردهی اولیه موقعیت سیستم را خواهیم نوشت. در این مد رجیستر خاص D1060=3 تنظیم شده و از ورودی X2 برای اجرای فرمان جایگاه‌یابی استفاده خواهد شد.

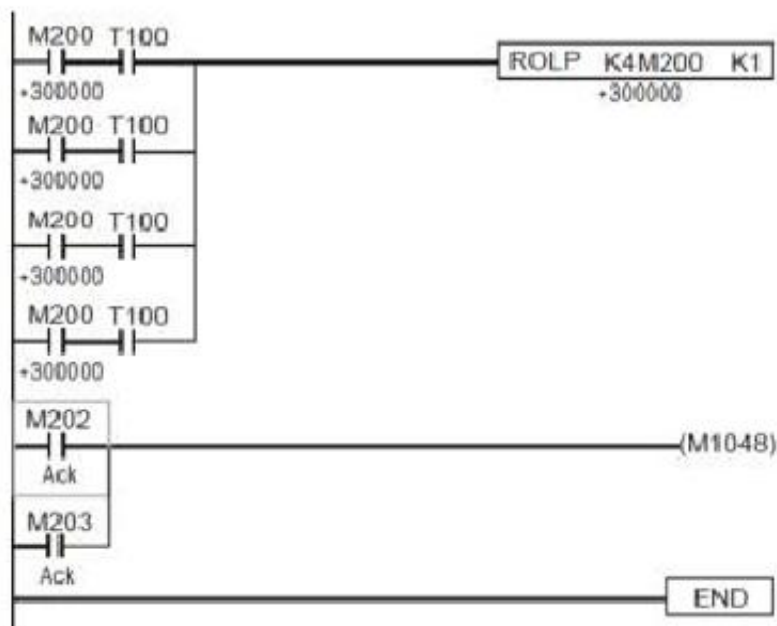


بخش 2 ← از ورودی X3 برای تریگر کردن عمل جایگاه‌یابی استفاده شده است. بعد از تکمیل شدن و به پایان رسیدن عمل جایگاه‌یابی، مد جایگاه‌یابی به صورت اتوماتیک به مد موقعیت‌یابی سوئیچ خواهد شد.



بخش 3 ← در این بخش، برنامه‌ی مربوط به جابجایی نقطه به نقطه را خواهیم نوشت. همانطور که گفته شد، مد جایگاه‌یابی به مد موقعیت‌یابی ($D1060=1$) سوئیچ شده و مکانیزم متصل شده به موتور بین نقاط $-300000 \sim +300000$ به عقب و جلو حرکت خواهد کرد.





اگر در یک پروژه نیاز به مد جایگاه‌یابی نداشتید، نیاز به برنامه‌نویسی بخش 1 و بخش 2 نمی‌باشد. توجه داشته باشید که تنها باید شرط فعال شدن فلگ خاص M1040 در بخش یک را به بخش سه اضافه کنید. شرط فعال شدن فلگ M1040 هم وابسته به فعال شدن ورودی دیجیتال X2 می‌باشد. بعلاوه، در بخش 3 از فلگ M101 برای تنظیم مد کنترل استفاده شده که می‌توان این فلگ را حذف کرده و به جای آن از فلگ خاص M1002 استفاده کرد. به این ترتیب، وقتی که PLC داخلی اینورتر وارد وضعیت RUN شود، مد کنترل موقعیت فعال خواهد شد.

8-9 مثال‌هایی از دستورات اختصاصی شبکه در PLC داخلی اینورتر VFD-C2000

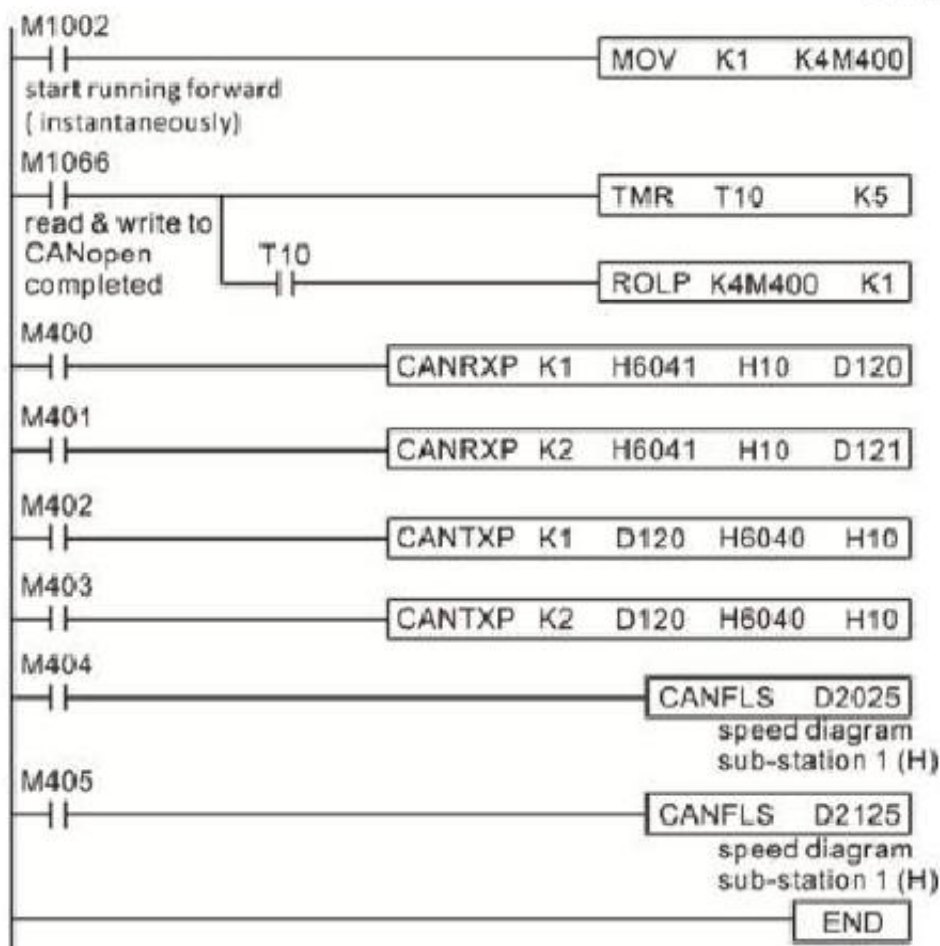
8-9-1 مثالی از راه‌اندازی شبکه‌ی CANopen اینورتر توسط PLC داخلی اینورتر

از دستورالعمل CANRX برای خواندن دیتا از ایستگاه CANopen Slave استفاده می‌شود. این دستورالعمل دارای 4 عملوند است. در عملوند S1 شماره ایستگاه دستگاه Slave تنظیم می‌شود. در عملوند S2 شماره‌ی Index اصلی (گروه پارامتر) تنظیم می‌شود. در عملوند S3 شماره Subindex+bit length (پارامترهای زیرگروه) تنظیم می‌شود و در عملوند D آدرس جاری از حافظه‌ی PLC نوشته می‌شود. بنابراین، می‌توان گفت که از دستورالعمل CANRX برای خواندن مقدار از یک index از یک دستگاه Slave در شبکه‌ی CANopen استفاده می‌شود.

وقتی که این دستورالعمل اجرا شود، یک پیغام SDO به دستگاه Slave مورد نظر ارسال خواهد کرد. در همین حین، هر دو فلگ خاص M1066 و M1067 برابر با 0 هستند و بعد از عمل خواندن

فلگ خاص M1066 برابر با 1 خواهد شد. اگر ایستگاه Slave پاسخ صحیح به دستگاه Master دهد، دستگاه Master مقدار را در رجیستر خاص اختصاص داده شده به عملوند D خواهد نوشت و در همین هنگام فلگ خاص M1067=1 خواهد شد. اگر دستگاه Slave یک پاسخ خطا به دستگاه Master دهد، فلگ خاص M1067=0 شده و یک پیغام خطا در رجیسترهای خاص D1076~D1079 ثبت خواهد شد.

به عنوان مثال، وقتی که PLC در وضعیت RUN است و یک بار دستورالعمل CANRX اجرا شود، K4M400=K1 خواهد شد. سپس، هر گاه که M1066=1 شود، او به یک پیغام دیگر سوئیچ خواهد شد.



از دستورالعمل CANTX برای نوشتن دیتا در دستگاه CANopen Slave استفاده می‌شود. این دستورالعمل دارای 4 عملوند است. در عملوند S1 شماره ایستگاه Slave تنظیم می‌شود. در عملوند S2 آدرس پارامتر برای نوشتن مقدار در آن پارامتر تنظیم می‌شود. در عملوند S3 شماره Index اصلی (گروه پارامتر) تنظیم شده و در عملوند S4 شماره Subindex+bit length (پارامترهای زیر گروه)

تنظیم می‌شود. بنابراین، می‌توان گفت که از دستورالعمل CANTX برای نوشتن مقدار در یک index از یک دستگاه Slave در شبکه‌ی CANopen استفاده می‌شود.

وقتی که این دستورالعمل اجرا شود، یک پیام SDO به دستگاه Slave مورد نظر ارسال خواهد کرد. در همین حین، هر دو فلگ خاص M1066 و M1067 برابر با 0 هستند و بعد از عمل نوشتن فلگ خاص M1066 برابر با 1 خواهد شد. اگر ایستگاه Slave پاسخ صحیح به دستگاه Master دهد، دستگاه Master مقدار را در رجیستر خاص اختصاص داده شده به عملوند D خواهد نوشت و در همین هنگام فلگ خاص M1067=1 خواهد شد. اگر دستگاه Slave یک پاسخ خطا به دستگاه Master دهد، فلگ خاص M1067=0 شده و یک پیام خطا در رجیسترهای خاص D1076~D1079 ثبت خواهد شد.

از دستورالعمل CANFLS برای به‌روز کردن دیتای موجود در رجیسترهای خاص D که برای شبکه‌ی CANopen رزرو شده‌اند و برای تبادل دیتا در شبکه از آنها استفاده می‌شود، می‌توان استفاده کرد. این دستورالعمل تنها یک عملوند دارد. در عملوند S1 شماره رجیسترهای خاص D برای به‌روز کردن دیتای آن نوشته می‌شود.

وقتی که صفت (ویژگی یا همان خاصیت) پارامتر فقط خواندنی باشد، با اجرا شدن این دستورالعمل یک پیام معادل با CANRX به دستگاه Slave ارسال خواهد شد و دستگاه Slave یک مقدار را برگشت داده و رجیستر خاص D را به‌روزرسانی خواهد کرد.

وقتی که صفت پارامتر خواندنی و نوشتنی باشد، با اجرا شدن این دستورالعمل یک پیام معادل با CANTX به دستگاه Slave ارسال شده و مقدار رجیستر خاص D در پارامتر مورد نظر در دستگاه Slave نوشته می‌شود.

وقتی که فلگ‌های خاص M1066 و M1067 هر دو 0 باشند و بعد از عمل خواندن وضعیت فلگ M1066 برابر با 1 شود، اگر ایستگاه Slave پاسخ صحیح به دستگاه Master دهد، مقدار در رجیستر تعیین شده نوشته شده و فلگ خاص M1067 برابر با 1 خواهد شد. اگر پاسخ دستگاه Slave شامل یک کد خطا بود، فلگ خاص M1067 برابر با 0 شده و یک پیام خطا در رجیسترهای خاص D1076~D1079 ثبت خواهد شد.

8-9-2- مثال‌ی از ارتباط داخلی CPU اینورتر و PLC CPU داخلی اینورتر

از دستورالعمل ICOMR برای خواندن مقدار در ارتباطات داخلی استفاده می‌شود. این دستورالعمل دارای 4 عملوند می‌باشد. توسط عملوند S1 می‌توان دستگاه Slave را انتخاب کرد. توسط عملوند S2 می‌توان CPU داخلی را انتخاب کرد. اگر مقدار عملوند S2 برابر با 0 تنظیم شود، یعنی CPU مربوط

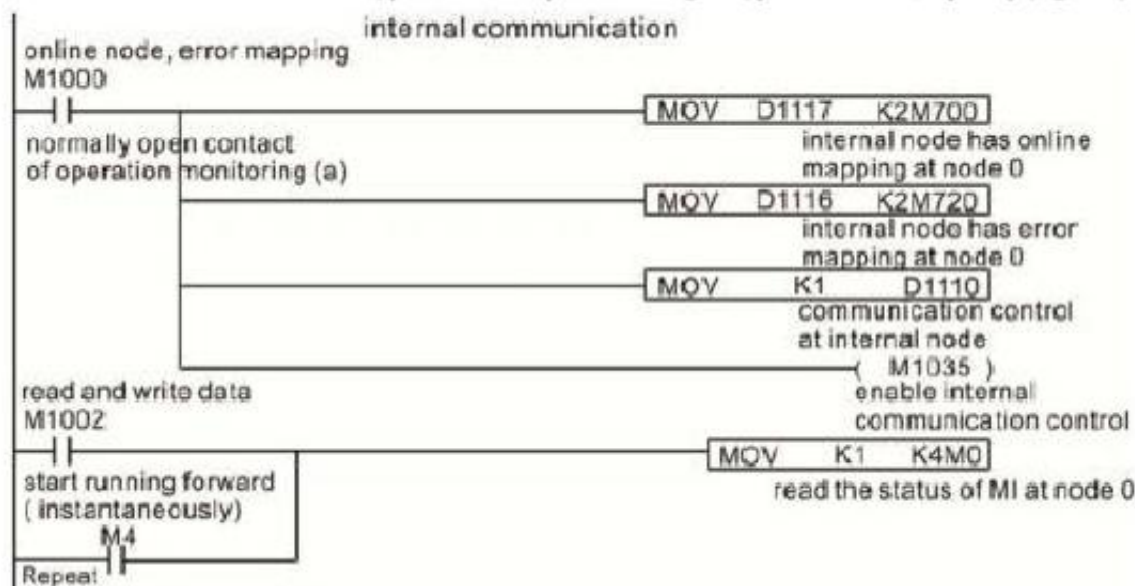
اینورتر انتخاب شده ولی اگر 1 تنظیم شود، یعنی CPU مربوط به PLC داخلی اینورتر انتخاب شده است. در عملکرد S3 آدرس مربوط به خواندن مقدار پارامتر اینورتر یا رجیستر PLC داخلی اینورتر تنظیم شده و در عملکرد D مقدار خوانده شده ذخیره می‌شود.

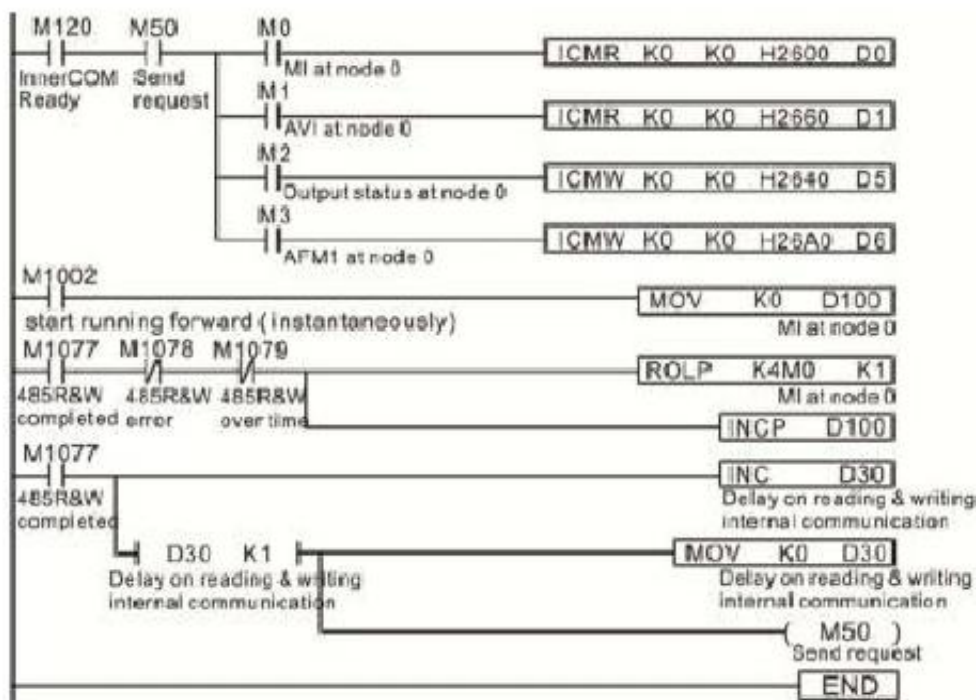
دستورالعمل ICOMR توانایی بدست آوردن ایستگاه‌های اینورتر Slave و مقدار رجیستر PLC های داخلی را دارد.

از دستورالعمل ICOMW برای نوشتن مقدار در ارتباطات داخلی استفاده می‌شود. این دستورالعمل دارای 4 عملکرد است. توسط عملکرد S1 می‌توان دستگاه Slave را انتخاب کرد. توسط عملکرد S2 می‌توان CPU داخلی را انتخاب کرد. اگر مقدار عملکرد S2 برابر با 0 تنظیم شود، یعنی CPU مربوط به اینورتر انتخاب شده ولی اگر 1 تنظیم شود یعنی CPU مربوط به PLC داخلی اینورتر انتخاب شده است. در عملکرد S3 آدرس مربوط به نوشتن مقدار در پارامتر اینورتر یا رجیستر PLC داخلی اینورتر تنظیم شده و در عملکرد D مقدار خوانده شده ذخیره می‌شود.

دستورالعمل ICOMW یک مقدار را در CPU ی اینورتر Slave و یا رجیستر PLC های داخلی اینورتر می‌نویسد.

در شکل زیر، نحوه‌ی استفاده دستورالعمل ICMW و ICMR آورده شده است :





8-9-3 به کارگیری CANopen Master Control در PLC داخلی اینورتر

در جایی که نیاز باشد چند محور توسط شبکه‌ی CANopen کنترل شود، می‌توان از اینورتر VFD-C2000 دلتا استفاده کرد. اینورتر VFD-C2000 از پروتکل CANopen Master پشتیبانی می‌کند. بنابراین، در شبکه می‌توان یک اینورتر را به عنوان CANopen Master پیکربندی کرده و اینورترهای دیگر را به عنوان CANopen Slave پیکربندی کرد. توسط اینورتر VFD-C2000 با پروتکل CANopen Master می‌توان اینورترها را در مد سرعت، موقعیت، گشتاور و جایگاه‌یابی کنترل کرد. برای پیکربندی شبکه‌ی CANopen می‌توان از 7 مرحله‌ی ذکر شده در زیر استفاده کرد:

مرحله 1 ← فعال کردن توابع CANopen Master

1. برای مقاردهی اولیه‌ی تابع Master باید پارامتر $Pr.09-45=1$ تنظیم شود. بعد از اینکه تنظیمات اینورتر به پایان رسید، برای اعمال تنظیمات باید یکبار تغذیه اینورتر را قطع و وصل نمایید. در نوار ابزار کی‌پد دیجیتال KPC-CC01 باید عنوان CAN Master به نمایش درآید.
2. برای ریست کردن PLC و باز گرداندن مقادیر دیتا رجیسترهای PLC به مقادیر پیش فرض باید پارامتر $Pr.00-02$ را برابر با 6 تنظیم نمایید.
3. تغذیه اینورتر را قطع کرده و مجدداً وصل نمایید.

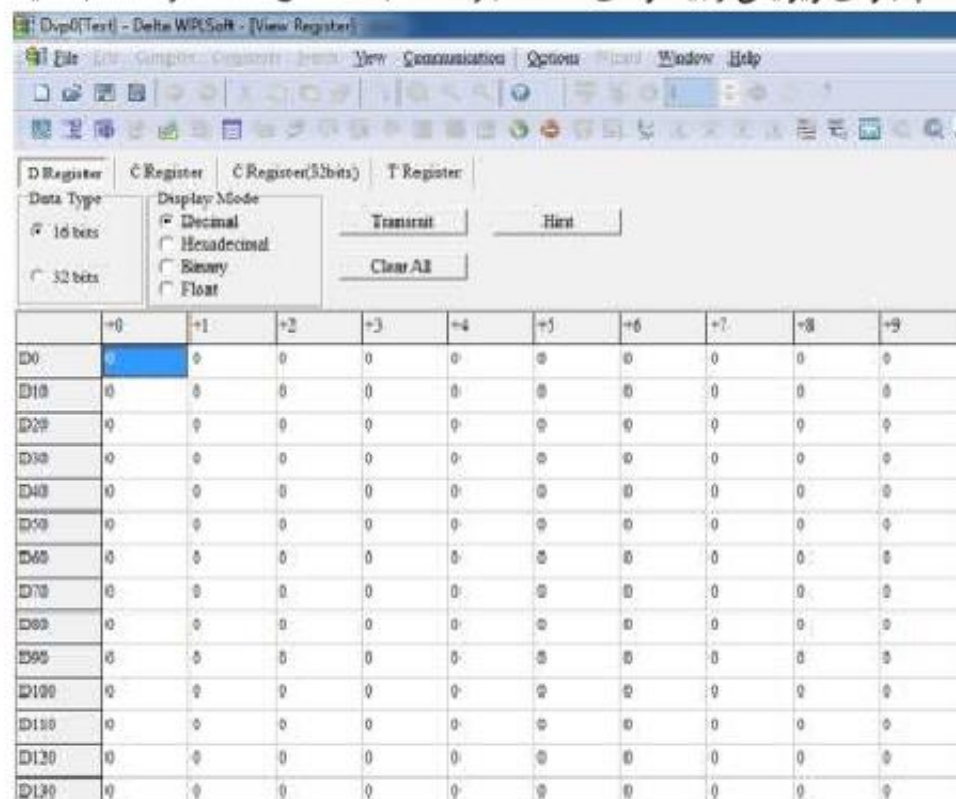
4. برای تنظیم مد کنترل PLC برابر با PLC Stop باید از کی‌پد دیجیتال KPC-CC01 استفاده نمایید. اگر از کی‌پد دیجیتال KPC-CE01 استفاده می‌نمایید، برای تنظیم مد کنترل PLC برابر با مد PLC Stop باید پارامتر PLC2 را تنظیم نمایید. اگر برای اولین بار از اینورتر VFD-C2000 استفاده می‌کنید، به دلیل خالی بودن حافظه‌ی برنامه‌ی PLC داخلی، کد هشدار PLFF به نمایش درخواهد آمد.

مرحله‌ی 2 ← تنظیم حافظه‌ی Master

1. بعد از وصل کابل ارتباطات RS-485 به اینورتر و کامپیوتر، نرم‌افزار WPLSoft را اجرا کرده و یک پروژه جدید باز کرده و در این پروژه کنترلر را VFD-C2000 انتخاب نمایید. با استفاده از نرم‌افزار WPLSoft وضعیت PLC داخلی اینورتر را به وضعیت Stop تغییر دهید. توجه داشته باشید که اگر مد PLC داخلی اینورتر برابر با مد PLC Stop بود، باید وضعیت PLC را در نرم‌افزار نیز Stop نمایید.

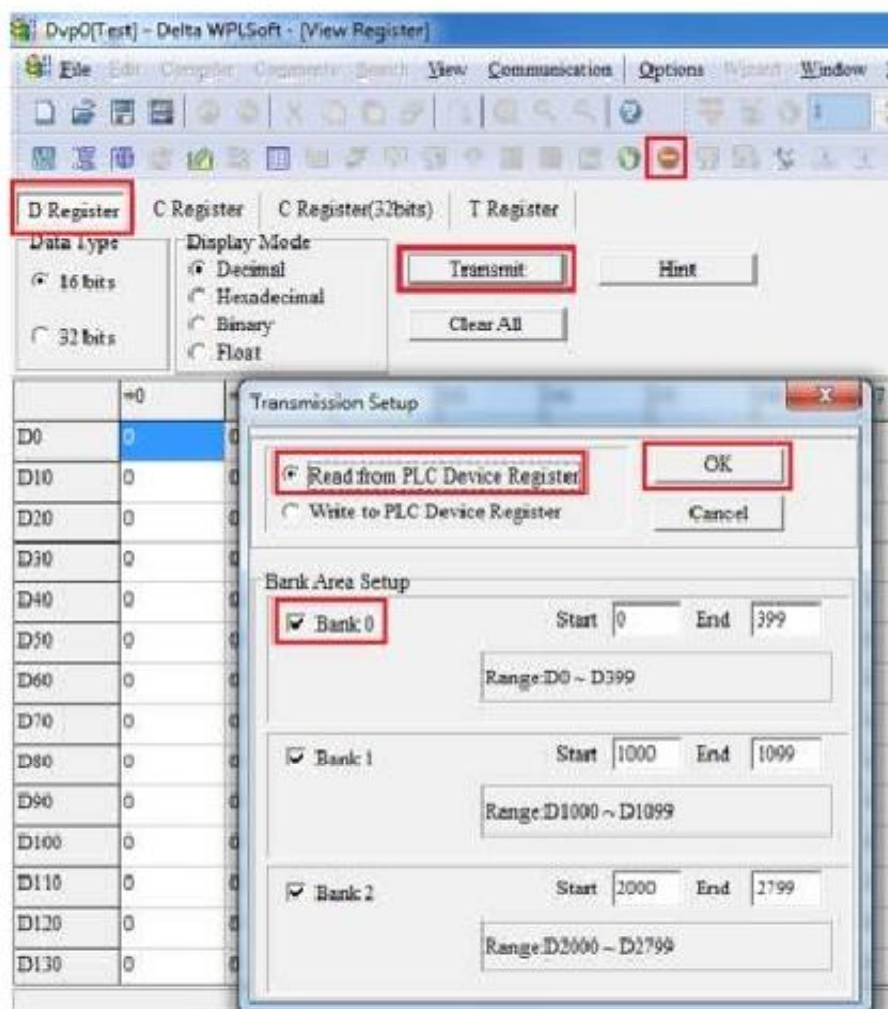
2. در این مرحله باید آدرس و شماره ایستگاه دستگاه Slave کنترل شده توسط دستگاه Master را تنظیم نمایید. به عنوان مثال، اگر می‌خواهید دو دستگاه Slave را توسط Master کنترل نمایید و شماره ایستگاه این دو دستگاه برابر با 21 و 22 است، فقط لازم است که دیتا رجیستر D2000 را برابر با 20 و دیتا رجیستر D2100 را برابر با 21 تنظیم نمایید. از آنجایی که توسط یک اینورتر VFD-C2000 پیکربندی شده به عنوان CANopen Master می‌توان 8 اینورتر VFD-C2000 پیکربندی شده به عنوان CANopen Slave را کنترل کرد، باید در نرم‌افزار WPLSoft برای تنظیم شماره ایستگاه این اینورترها از دیتا رجیسترهای D2000, D2100, D2200, D2300, D2400, D2500, D2600, D2700 استفاده شود. چون در این مثال ما تنها از دو دستگاه Slave استفاده کرده‌ایم، فقط کافی است که دیتا رجیسترهای D2000 و D2100 را تنظیم کرده و دیگر دیتا رجیسترها (D2200, D2300, D2400, D2500, D2600, D2700) را برابر با صفر قرار دهیم. در ادامه، نحوه‌ی تنظیم کردن این دیتا رجیسترها در نرم‌افزار WPLSoft توضیح داده شده است.

ابتدا نرم‌افزار WPLSoft را باز کرده و بعد از انتخاب کنترلر VFD-C2000 در یک پروژه جدید، بر روی نوار ابزار Communication زیرمنوی (T, C, D) Edit Register Memory کلیک نمایید.



اگر برنامه‌ی ایجاد شده برای PLC جدید بود و تنظیماتی برای PLC انجام نشده بود، می‌توان دیتای پیش‌فرض اینورتر را خوانده و دیتا رجیسترهای PLC داخلی اینورتر را مطابق با نیاز تنظیم کرد. اگر قبلاً تنظیماتی برای اینورتر انجام شده بود، در دیتا رجیسترهای خاص (D) مربوط به ناحیه CANopen مقادیر تنظیم شده که از قبل در حافظه‌ی PLC داخلی ذخیره شده است، به نمایش درخواهد آمد. دیتا رجیسترهای خاص D1090~D1099 و D2000~D2799 برای تنظیمات دستگاه یا همان اینورتر Slave1 و Slave2 در شبکه‌ی CANopen برنامه‌ریزی شده‌اند. توجه داشته باشید که این رجیسترها توسط شرکت دلتا برنامه‌ریزی شده و ما قادر به استفاده از آنها در شبکه‌ی CANopen و تنظیمات شبکه‌ی CANopen هستیم. فرض کنید که یک برنامه جدید باز کرده‌ایم، ابتدا باید مقادیر پیش‌فرض PLC داخلی اینورتر را بخوانیم تا بتوانیم فرمت ارتباطات را بررسی نماییم. اگر لینک ارتباطات وجود نداشته باشد، شماره ایستگاه PLC داخلی اینورتر به طور پیش‌فرض برابر با 2,9600, 7,N,2, ASCII خواهد بود. مراحل گفته شده در زیر را طی نمایید:

1. PLC داخلی اینورتر را به وضعیت PLC Stop ببرید.
2. دکمه‌ی Transmit را کلیک نمایید.
3. بعد از خارج شدن از پنجره بر روی دکمه‌ی Read memory کلیک نمایید.
4. از دیتا رجیسترهای D0~D399 صرف‌نظر کرده و آنها را نخوانید.
5. بر روی دکمه‌ی Confirm کلیک نمایید.



بعد از خواندن دیتا، لازم است که برخی از تنظیمات را توسط رجیسترهای خاص D انجام دهید. قبل از شروع تنظیمات ابتدا باید هر یک از رجیسترهای خاص را بررسی کرده و نام هر یک و همچنین، محدوده‌ی قابل تنظیم آنها را نیز باید بدانید. رجیسترهای خاص D مربوط به تنظیمات CANopen Master رجیسترهای D1070~D1099 و رجیسترهای D2000~D2799 می‌باشد. این رجیسترها به سه قسمت تقسیم شده‌اند.

- از اولین قسمت برای نمایش وضعیت جاری شبکه‌ی CANopen استفاده می‌شود که شامل رجیسترهای D1070~D1089 می‌باشد
- از دومین قسمت برای تنظیمات اصلی شبکه‌ی CANopen استفاده می‌شود که شامل رجیسترهای D1090~D1099 می‌باشد
- از سومین قسمت برای مسيردهی حافظه و نواحی کنترلی دستگاه Slave استفاده می‌شود که شامل رجیسترهای D2000~D2799 می‌باشد.

در ادامه، به شرح هر یک از این سه ناحیه خواهیم پرداخت:

وقتی که اینورتر **Master** یک اینورتر **Slave** را مقداردهی اولیه می‌کند، از طریق رجیستر خاص **D1070** می‌توان متوجه شد که پیکربندی کدام یک از اینورترهای **Slave** انجام شده و به پایان رسیده است. همچنین، توسط رجیستر **D1071** می‌توان متوجه شد که در فرآیند پیکربندی اولیه در کدام دستگاه **Slave** خطایی رخ داده است و توسط رجیستر **D1074** می‌توان متوجه شد که پیکربندی کدام دستگاه **Slave** به درستی و بطور کامل انجام نشده است.

بعد از اینکه پیکربندی به درستی انجام شد، توسط رجیستر **D1073** می‌توان متوجه شد که کدام اینورتر در شبکه آف‌لاین است. علاوه، می‌توان با استفاده از فرامین **CANRX**, **CANTX**, **CANFLS** بررسی کرد که دیتا در حال خواندن یا نوشتن در دستگاه **Slave** است، یا خیر. با استفاده از رجیسترهای **D1076~D1079** در صورت بروز خطا در عملیات نوشتن و خواندن دیتا از دستگاه **Slave** می‌توان به خطای رخ داده دسترسی داشت.

در جدول زیر، به صورت مختصر در رابطه با هر یک از رجیسترها توضیحاتی آورده شده و قابلیت دسترسی (فقط خواندنی، خواندنی/نوشتنی) آن نیز تعیین گردیده است.

Special D	Description of Function	R/W
D1070	Channel opened by CANopen initialization (bit0=Machine code0)	R
D1071	Error channel occurring in CANopen initialization process (bit0=Machine code0)	R
D1072	Reserved	-
D1073	CANopen break channel (bit0=Machine code0)	R
D1074	Error code of master error 0: No error 1: Slave station setting error 2: Synchronizing cycle setting error (too small)	R
D1075	Reserved	-
D1076	SDO error message (main index value)	R
D1077	SDO error message (secondary index value)	R
D1078	SDO error message (error code L)	R
D1079	SDO error message (error code H)	R

دومین ناحیه‌ی رجیسترهای خاص شبکه‌ی **CANopen** به تنظیمات اصلی شبکه‌ی **CANopen** تخصیص داده شده است. دقت داشته باشید که وقتی که از این ناحیه استفاده می‌کنید و در حال تنظیم آن هستید باید **PLC** داخلی اینورتر در وضعیت **Stop** قرار گیرد.

ابتدا باید زمان تبادل دیتا را برای اینورتر **Master** و اینورتر **Slave** تنظیم کنید. از طریق رجیستر **D1090** می‌توان زمان سیکل سنکرون سازی شبکه‌ی **CANopen** برای اینورترهای تحت شبکه را تنظیم کرد.

Special D	Description of Function	Default	R/W
D1090	Synchronizing cycle setting	4	RW

برای تنظیم این زمان باید از رابطه‌ی گفته شده در زیر استفاده نمایید:

$$Sync\ time \geq \frac{1M}{Rate} \times \frac{N}{4}$$

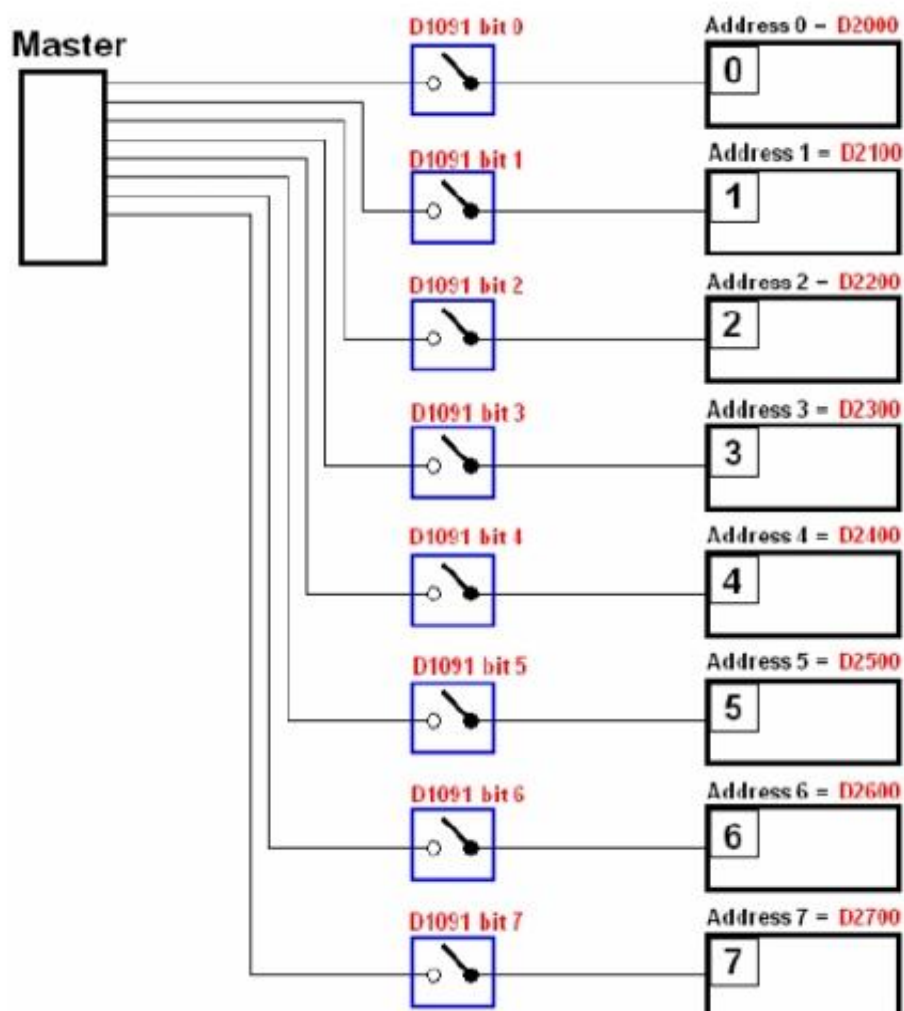
$N:TXPDO + RXPDO$

بعنوان مثال، وقتی که سرعت تبادل دیتا 500K بوده و TXPDO, RXPDO به 8 دستگاه متصل باشد، زمان سنکرون سازی مورد نیاز بیشتر از 4 میلی‌ثانیه خواهد بود.

همچنین باید تعداد اینورترهای Slave را که می‌خواهیم قابل دسترس باشند را تعریف کنیم. رجیستر خاص D1091 کانالی برای تعریف ایستگاه‌های باز و رجیستر $D2000+100 \times n$ شماره ایستگاه تعریف شده برای این کانال می‌باشند. در جدول زیر، جزئیات مربوطه توضیح داده شده است:

رجیستر D	توضیحات تابع	قابلیت دسترسی
D1091	بیت‌های 0~7 این رجیستر مربوط به شماره ایستگاه 0~7 متصل شده به شبکه‌ی CANopen می‌باشد. با ON کردن هر یک از این بیت‌ها می‌توانید شماره ایستگاه متناظر آن را باز کنید و با OFF کردن هر یک از این بیت‌ها می‌توانید شماره ایستگاه متناظر آن را ببندید. در صورت باز بودن ایستگاه (اینورتر) Slave می‌توان از آن اینورتر در شبکه استفاده کرد.	RW
$D2000+100 \times n$	شماره ایستگاه (اینورتر) Slave 0~7.	RW

در شکل زیر نمایی از باز و بسته کردن هر یک از ایستگاه‌های (اینورترهای) Slave متصل شده به دستگاه Master آورده شده است.



اگر اینورتر Slave دارای یک راه‌انداز کند باشد، دستگاه Master می‌تواند تأخیری برای یک مدت کوتاه قبل از انجام پیکربندی ایستگاه Slave ایجاد کند. این تأخیر زمانی را می‌توان در رجیستر خاص D1092 تنظیم کرد.

رجیستر D	توضیحات تابع	مقدار پیش فرض	قابلیت دسترسی
D1092	توسط این رجیستر می‌توان یک تأخیر زمانی برای قبل از شروع مقداردهی اولیه اینورترهای Slave توسط دستگاه Master را در دستگاه Master تنظیم کرد.	0	RW

صرفنظر از مقداردهی اولیه‌ی دستگاه Slave، یک تأخیر زمانی را می‌توان برای تشخیص اینکه کدام دستگاه Slave دارای خطا است، تنظیم کرد. اگر سرعت تبادل دیتا کم است، تأخیر زمانی را می‌توان برای تشخیص اینکه مقداردهی اولیه کدام دستگاه تکمیل شده است، تنظیم کرد. از این زمان برای اطمینان از اینکه در این مدت مقداردهی اولیه دستگاه Slave انجام می‌شود، استفاده خواهد شد.

رجیستر D	توضیحات تابع	مقدار پیش‌فرض	قابلیت دسترسی
D1099	از این رجیستر برای تنظیم یک تأخیر زمانی جهت تکمیل کردن مقداردهی اولیه دستگاه‌های Slave توسط دستگاه Master استفاده می‌شود. محدوده‌ی قابل تنظیم برای این رجیستر بین 1 الی 60000 ثانیه است.	15 ثانیه	RW

بعد از اینکه ارتباطات شبکه با موفقیت به پایان رسید، سیستم باید شناسایی کند که آیا این یک وقفه در ارتباط با دستگاه Slave است، یا خیر. از رجیستر خاص D1093 برای تنظیم زمان شناسایی (آشکارسازی) و از رجیستر خاص D1094 برای تنظیم تعداد خطاهای متوالی که در یک خطای وقفه راه‌اندازی خواهد شد، استفاده می‌شود.

رجیستر D	توضیحات تابع	مقدار پیش‌فرض	قابلیت دسترسی
D1093	از این رجیستر خاص برای تنظیم زمان شناسایی (آشکارسازی) وقفه‌ی بوجود آمده در ارتباطات شبکه استفاده می‌شود.	1000ms	RW
D1094	از این رجیستر خاص برای تنظیم تعداد وقفه‌های متوالی رخ داده شده در هنگام شناسایی (آشکارسازی) وقفه‌ی ارتباطات استفاده می‌شود.	3	RW

نوع بسته‌ی ارسالی توسط PDO قبل از برقراری ارتباط نرمال تنظیم خواهد شد و معمولاً نیاز به تنظیم مجدد آن نمی‌باشد.

رجیستر D	توضیحات تابع	مقدار پیش‌فرض	قابلیت دسترسی
D1097	نوع بسته‌ی ارسالی توسط PDO مطابق با زمان بلادرنگ (Real-time) می‌باشد و مقدار آن از 1 الی 240 قابل تنظیم است.	1	RW
D1098	نوع بسته‌ی دریافتی توسط PDO مطابق با زمان بلادرنگ (Real-time) می‌باشد و مقدار آن از 1 الی 240 قابل تنظیم است.	1	RW

سومین قسمت از تنظیمات شبکه‌ی CANopen به مسيردهی حافظه‌ی ایستگاه (اینورتر) Slave و ناحیه‌ی کنترلی آن اختصاص دارد.

در شبکه‌ی CANopen یک روش به نام PDO برای انجام مسيردهی حافظه دستگاه‌های Slave و Master ارائه شده است که به دستگاه Master امکان می‌دهد که بتواند مستقیماً به دیتا برای خواندن مقدارش و نوشتن مقدار در آن در نواحی از حافظه‌ی دستگاه Slave دسترسی داشته باشد. دستگاه Master به صورت اتوماتیک عمل تبادل دیتا را با دستگاه Slave انجام داده و می‌تواند بعد از برقراری ارتباط و انجام تبادل دیتا در زمان بلادرنگ ($M1034=1time$) مقادیر خوانده شده را

مستقیماً در رجیسترهای خاص D ذخیره کرده و مقادیر را در رجیسترهای خاص D در دستگاه Slave بنویسد. اینورتر VFD-C2000 دلتا مستقیماً از مسیریابی بلادرنگ حافظه برای 4 عدد PDO پشتیبانی می‌کند. دو نوع PDO برای تبادل دیتا وجود دارد که اولین نوع آن RxPDO بوده که اطلاعات را از دستگاه Slave می‌خواند و دومین نوع آن TxPDO می‌باشد که اطلاعات را در دستگاه Slave می‌نویسد. بعلاوه، به منظور ساده کردن عمل کنترل، اینورتر VFD-C2000 توانایی انجام مسیریابی رجیسترهای عمومی را ندارد. در جدول زیر، خلاصه‌ای از وضعیت مسیریابی PDO آورده شده است:

TX PDO							
PDO4		PDO3		PDO2		PDO1	
Description	Special D	Description	Special D	Description	Special D	Description	Special D
Controller Word	D2008+100*n	Controller Word	D2008+100*n	Slave device DO	D2027+100*n	Controller Word	D2008+100*n
Target torque	D2017+100*n	Target torque	D2020+100*n D2021+100*n	Slave device AO1	D2031+100*n	Target speed	D2012+100*n
Control method	D2010+100*n	Control method	D2010+100*n	Slave device AO2	D2032+100*n		
				Slave device AO3	D2033+100*n		

TX PDO							
PDO4		PDO3		PDO2		PDO1	
Description	Special D	Description	Special D	Description	Special D	Description	Special D
Mode Word	D2009+100*n	Mode Word	D2009+100*n	Slave device DI	D2026+100*n	Mode Word	D2009+100*n
Actual torque	D2018+100*n	Actual position	D2022+100*n D2023+100*n	Slave device AI1	D2028+100*n	Actual frequency	D2013+100*n
Actual mode	D2011+100*n	Actual mode	D2011+100*n	Slave device AI2	D2029+100*n		
				Slave device AI3	D2030+100*n		

به دلیل ساده کردن استفاده از اینورترهای VFD-C2000 در شبکه‌ی CANopen بدون PLC خارجی و با استفاده از PLC داخلی خود اینورتر برای باز کردن PDO مربوط به اینورترهای متصل شده به شبکه برای TxPDO باید رجیسترهای خاص $D2034+100 \times n$ و برای RxPDO باید رجیستر خاص $D2067+100 \times n$ را تنظیم کرد.

این دو ناحیه برای رجیسترهای خاص D به صورت زیر تعریف خواهند شد:

Default Definition	PDO4 Torque		PDO3 Position		PDO2 Remote I/O		PDO1 Speed	
Bit	15	14 ~ 12	11	10 ~ 8	7	6 ~ 4	3	2 ~ 0
Definition	En	Length:	En	Length:	En	Length:	En	Length:

در جدول بالا En بیانگر این است که کدام PDO استفاده شده است.

Length بیانگر این است که مسیره‌ی برای چندین متغیر انجام شده است.

به عنوان یک مثال ساده، اگر بخواهیم یک اینورتر VFD-C2000 را که به عنوان یک دستگاه Slave در شبکه پیکربندی شده است را در مد کاری سرعت کنترل کنیم، باید تنظیماتی مطابق با تنظیمات زیر انجام دهیم:

$D2034+100 \times n = 000Ah$

Length	TX PDO							
	PDO4		PDO3		PDO2		PDO1	
	Description	Special D	Description	Special D	Description	Special D	Description	Special D
1	Controller Word	$D2008+100 \times n$	Controller Word	$D2008+100 \times n$	Slave device DO	$D2027+100 \times n$	Controller Word	$D2008+100 \times n$
2	Target torque	$D2017+100 \times n$	Target torque	$D2020+100 \times n$ $D2021+100 \times n$	Slave device AO1	$D2031+100 \times n$	Target speed	$D2012+100 \times n$
3	Control method	$D2010+100 \times n$	Control method	$D2010+100 \times n$	Slave device AO2	$D2032+100 \times n$		
4					Slave device AO3	$D2033+100 \times n$		

	PDO4		PDO3		PDO2		PDO1	
Definition	Torque		Position		Remote I/O		Speed	
Bit	15	14 ~ 12	11	10 ~ 8	7	6 ~ 4	3	2 ~ 0
Definition	0	0	0	0	0	0	1	2

D2067+100×n=000Ah

Length	TX PDO							
	PDO4		PDO3		PDO2		PDO1	
	Description	Special D	Description	Special D	Description	Special D	Description	Special D
1	Controller Word	D2009+100*n	Controller Word	D2009+100*n	Slave device DI	D2026+100*n	Controller Word	D2009+100*n
2	Actual torque	D2018+100*n	Actual position	D2022+100*n D2023+100*n	Slave device AI1	D2028+100*n	Actual frequency	D2013+100*n
3	Actual mode	D2011+100*n	Actual mode	D2011+100*n	Slave device AI2	D2029+100*n		
4					Slave device AI3	D2030+100*n		

	PDO4		PDO3		PDO2		PDO1	
Definition	Torque		Position		Remote I/O		Speed	
Bit	15	14 ~ 12	11	10 ~ 8	7	6 ~ 4	3	2 ~ 0
Definition	0	0	0	0	0	0	1	2

بعد از اینکه تنظیمات مربوط به شبکه‌ی CANopen به پایان رسید، PLC داخلی اینورتر را به وضعیت RUN سوئیچ کنید. سپس منتظر بمانید تا مقداردهی اولیه شبکه CANopen با موفقیت به پایان برسد. در این حالت، باید فلگ خاص M1059=1 شده و فلگ خاص M1061=0 شود. همچنین، با یک شدن فلگ خاص M1034 متوجه خواهید شد که مقداردهی اولیه مسيردهی حافظه‌ی PLC برای شبکه‌ی CANopen با موفقیت به پایان رسیده است. Control word و Frequency Command به صورت اتوماتیک و مطابق با وضعیت دستگاه Slave... (رجیستر خاص D2008+n×100 و رجیستر خاص D2012+n×100) به روزرسانی خواهند شد و همچنین، مقدار Status word و فرکانس جاری دستگاه (اینورتر) Slave (رجیستر D2009+n×100 و رجیستر D2013+n×100) به صورت اتوماتیک از دستگاه Master به دستگاه Slave ارسال خواهد گردید. در ادامه، با ذکر مثال توضیح خواهیم داد که دستگاه Master چگونه می‌تواند این وظایف (Task)ها را برای عملیات خواندن/نوشتن در ناحیه رجیسترهای خاص D به کار گیرد.

بعلاوه، باید به این نکته توجه داشت که از ریموت I/O های PDO2 می‌توان برای بدست آوردن وضعیت ورودی‌های دیجیتال (DI) و ورودی‌های آنالوگ (AI) دستگاه‌های (اینورترهای) Slave استفاده کرد و همچنین، می‌توان توسط ریموت I/O های PDO2 وضعیت خروجی‌های دیجیتال (DO) و آنالوگ

(AO) را کنترل کرد. بعد از معرفی کامل مسيردهی اتوماتیک حافظه برای رجیسترهای خاص D باید به این نکته اشاره کرد که اینورتر VFD-C2000 CANopen Master اطلاعات رجیسترهای خاص را دائماً به روزرسانی می‌کند. برای مثال، در مد کنترل سرعت، تنظیمات مربوط به شتاب‌گیری و کاهش سرعت ممکن است دائماً تغییر کند که در این صورت، نیاز به به روزرسانی دائمی دارد. دیتای مربوط به رجیسترهای خاص D به صورت بلادرنگ در حافظه ذخیره می‌شوند. با استفاده از فرمان CANFLS می‌توان فرامین شبکه را به روزرسانی کرد. در ادامه، نواحی تغییرات دیتای مربوط به دستگاه VFD-C2000 CANopen Master آورده شده است. این تغییر در رجیسترهای $D2001+100 \times n \sim D2033+100 \times n$ صورت می‌گیرد.

1. محدوده‌ی تغییر n از 0 تا 7 می‌باشد که بیانگر تعداد اینورترهای Slave متصل شده به دستگاه (اینورتر) Master است.

2. در جداول زیر، رجیسترهای خاص D می‌توانند با استفاده از فرمان CANFLS به روزرسانی شوند. همچنین ● بیانگر PDOTx بوده و ▲ بیانگر PDORx می‌باشد.

Special D	Description of Function	Default	PDO Default				R/W
			1	2	3	4	
$D2000+100 \times n$	Station number n of slave station Setting range: 0~127 0: No CANopen function	0					RW
$D2002+100 \times n$	Manufacturer code of slave station number n (L)	0					R
$D2003+100 \times n$	Manufacturer code of slave station number n (H)	0					R
$D2004+100 \times n$	Manufacturer's product code of slave station number n (L)	0					R
$D2005+100 \times n$	Manufacturer's product code of slave station number n (H)	0					R

رجیسترهای مربوط به تنظیمات اصلی برای هر Slave متصل شده به Master در جدول زیر آورده شده است:

Special D	Description of Function	Default	PDO Default				R/W
			1	2	3	4	
$D2006+100 \times n$	Communications break handling method of slave station number n	0					RW
$D2007+100 \times n$	Error code of slave station number n error	0					R
$D2008+100 \times n$	Control word of slave station number n	0	●		●	●	RW
$D2009+100 \times n$	Status word of slave station number n	0	▲		▲	▲	R
$D2010+100 \times n$	Control mode of slave station number n	2					RW
$D2011+100 \times n$	Actual mode of slave station number n	2					R

رجیسترهای مربوط به مد کنترل سرعت برای هر اینورتر **Slave** متصل شده به اینورتر **Master** در شبکه‌ی **CANopen** در جدول زیر آورده شده است:

Special D	Description of Function	Default	PDO Default				R/W
			1	2	3	4	
D2001+100*n	Torque restriction on slave station number n	0					RW
D2012+100*n	Target speed of slave station number n (rpm)	0	●				RW
D2013+100*n	Actual speed of slave station number n (rpm)	0	▲				R
D2014+100*n	Error speed of slave station number n (rpm)	0					R
D2015+100*n	Acceleration time of slave station number n (ms)	1000					RW
D2016+100*n	Deceleration time of slave station number n (ms)	1000					RW

رجیسترهای مربوط به مد کنترل گشتاور برای هر اینورتر **Slave** متصل شده به اینورتر **Master** در شبکه‌ی **CANopen** در جدول زیر آورده شده است:

Special D	Description of Function	Default	PDO Default				R/W
			1	2	3	4	
D2017+100*n	Target torque of slave station number n (-100.0%~+100.0%)	0				●	RW
D2018+100*n	Actual torque of slave station number n (XX.X%)	0				▲	R
D2019+100*n	Actual current of slave station number n (XX.XA)	0					R

رجیسترهای مربوط به مد کنترل موقعیت برای هر اینورتر **Slave** متصل شده به اینورتر **Master** در شبکه‌ی **CANopen** در جدول زیر آورده شده است:

Special D	Description of Function	Default	PDO Default				R/W
			1	2	3	4	
D2020+100*n	Target of slave station number n (L)	0			●		RW
D2021+100*n	Target of slave station number n (H)	0					RW
D2022+100*n	Actual position of slave station number n (L)	0			▲		R
D2023+100*n	Actual position of slave station number n (H)	0					R
D2024+100*n	Speed chart of slave station number n (L)	10000					RW
D2025+100*n	Speed chart of slave station number n (H)	0					RW

رجیسترهای مربوط به ریموت I/O برای هر اینورتر Slave متصل شده به اینورتر Master در شبکه‌ی CANopen در جدول زیر آورده شده است:

Special D	Description of Function	Default	PDO Default				R/W
			1	2	3	4	
D2026+100*n	MI status of slave station number n	0		▲			R
D2027+100*n	MO setting of slave station number n	0		●			RW
D2028+100*n	AI1 status of slave station number n	0		▲			R
D2029+100*n	AI2 status of slave station number n	0		▲			R
D2030+100*n	AI3 status of slave station number n	0		▲			R
D2031+100*n	AO1 setting of slave station number n	0		●			RW
D2032+100*n	AO2 setting of slave station number n	0		●			RW
D2033+100*n	AO3 setting of slave station number n	0		●			RW

بعد از شناخت رجیسترهای خاص، به توضیح مراحل تنظیم حافظه PLC داخلی اینورتر Master خواهیم پرداخت. بعد از وارد کردن مقادیر مورد نظر (تنظیمات مورد نیاز) در رجیسترهای D1090~D1099, D2000+100*n, D2034+100*n, D2067+100*n نمی‌توان شروع به دانلود برنامه کرد. رجیسترهای خاص مربوط به حافظه PLC داخلی اینورتر Master را مطابق با تنظیمات گفته شده در زیر تنظیم نمایید.

- رجیستر خاص D2000=20 و رجیستر خاص D2100=21 تنظیم نموده و رجیسترهای خاص D2200, D2300, D2400, D2500, D2600, D2700 را برابر با 0 تنظیم نمایید. اگر تنظیم این رجیسترها برابر با 0 خطایی ایجاد کرد، باید رجیستر خاص D1091 را برابر با 3 تنظیم نمایید که در این صورت ایستگاه‌های (اینورترهای) Slave از شماره‌ی 2 تا 7 غیرفعال خواهند شد.

- PLC داخلی اینورتر را به وضعیت Stop سوئیچ نمایید.

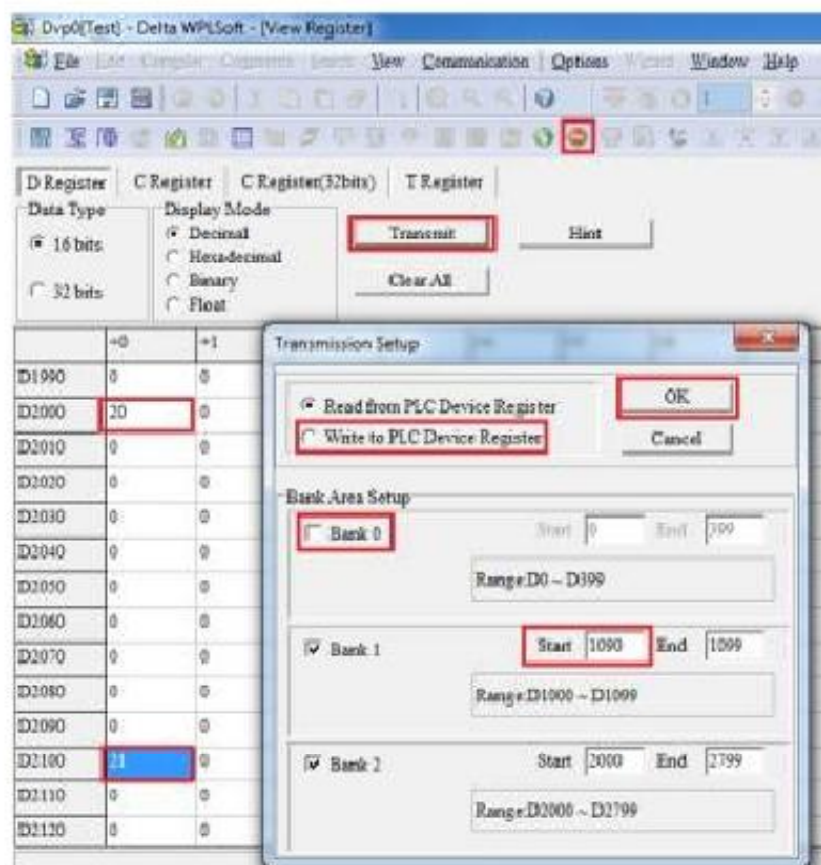
- دکمه‌ی انتقال (Transmit) را فشار دهید.

- بعد از خروج از پنجره‌ی فوق به روی Write memory کلیک نمایید.

- از رجیسترهای D0~D399 صرف‌نظر کنید.

- دومین محدوده‌ی حافظه را به D1090~1099 تغییر دهید.

- بر روی دکمه‌ی Confirm کلیک کنید.



برای تنظیم رجیستر خاص D1091 می‌توان از روش دیگری استفاده کرد. در این روش می‌توان تعیین کرد که ایستگاه‌های (اینورترهای) Slave از شماره‌ی 0 تا 7 مورد نیاز نمی‌باشد که برای این کار با غیرفعال کردن اینورترهای Slave مورد نظر باید بیت متناظر آن دستگاه را در رجیستر D1091 برابر با صفر کرد. به عنوان مثال، ما در شبکه به اینورترهای Slave شماره 2 تا 7 نیاز داریم که در این صورت باید مقدار رجیستر D1091 را برابر با 003B هگز تنظیم نماییم. این روش، همان کاری را انجام می‌دهد که روش گفته شده در بالا را انجام می‌دهد. برای مقداردهی اولیه در نرم‌افزار WPLSoft باید از منوی Communications زیر منوی use register edit (T C D) برای اجرای تنظیمات انتخاب نمایید.

مرحله‌ی 3 ← تنظیم تعداد ایستگاه ارتباطات Master و سرعت تبادل دیتا
وقتی که شماره ایستگاه Master را تنظیم می‌کنید، مطمئن شوید که از شماره‌ی یکسان با یک ایستگاه Slave استفاده نکرده باشید. این به آن معنی است که شماره ایستگاه تمام اینورترها در شبکه باید منحصر بفرد بوده و تکراری نباشد. توجه داشته باشید که مقدار پیش‌فرض پارامتر Pr.09-46 برابر با 100 است.

برای تنظیم سرعت تبادل دیتا در شبکه‌ی CANopen باید از پارامتر Pr.09-37 استفاده نمایید. صرف‌نظر از اینکه اینورتر VFD-C2000 به عنوان Master یا Slave تعریف شده باشد، باید سرعت تبادل دیتا در کل شبکه یکسان باشد. بنابراین، سرعت تبادل دیتای تنظیم شده در پارامتر Pr.09-37 باید در تمام اینورترها یکسان باشد.

مرحله‌ی 4 ← نوشتن کد برنامه

در قابلیت دسترسی بلادرنگ می‌توان مستقیماً ناحیه رجیسترهای D را از PLC خوانده و یا مقادیری را در ناحیه رجیسترهای D نوشت. در قابلیت دسترسی با وقفه (Non Real-time) می‌توان از فرامین نوشتن، خواندن و به‌روزرسانی استفاده کرد که در ادامه، هر یک را توضیح خواهیم داد.

فرمان Read ← برای خواندن مقادیر ناحیه‌ی رجیسترهای D می‌توان از فرمان CANRX استفاده کرد.

وقتی که در این وضعیت عمل خواندن مقادیر به پایان رسید، فلگ خاص M1066 برابر با 1 خواهد شد. وقتی که عمل خواندن مقادیر با موفقیت به پایان رسید، فلگ M1067 برابر با 1 خواهد شد و هنگامی که در هنگام خواندن دیتا خطایی رخ داد، فلگ M1067 برابر با 1 خواهد شد.

فرمان Write ← برای نوشتن مقدار در ناحیه‌ی رجیسترهای D می‌توان از فرمان CANTX استفاده کرد.

وقتی که در این وضعیت عمل نوشتن مقادیر به پایان رسید، فلگ خاص M1066 برابر با 1 خواهد شد. وقتی که عمل نوشتن مقادیر با موفقیت به پایان رسید، فلگ M1067 برابر با 1 خواهد شد و هنگامی که در هنگام نوشتن دیتا خطایی رخ داد، فلگ M1067 برابر با 1 خواهد شد.

فرمان Refresh ← برای به‌روزرسانی مقادیر ناحیه رجیسترهای D می‌توان از فرمان CANFLS استفاده کرد. اگر رجیستر D دارای خاصیت RW بود، اینورتر Master مقادیری را در اینورتر Slave خواهد نوشت. اگر رجیستر D دارای خاصیت RO باشد، اینورتر Slave مقدار خوانده شده را به اینورتر Master باز خواهد گرداند.

وقتی که در این وضعیت عمل به‌روزرسانی مقادیر به پایان رسید، فلگ خاص M1066 برابر با 1 خواهد شد. وقتی که عمل به‌روزرسانی مقادیر با موفقیت به پایان رسید، فلگ M1067 برابر با 1 خواهد شد و هنگامی که در هنگام به‌روزرسانی دیتا خطایی رخ داد، فلگ M1067 برابر با 1 خواهد شد.

به این نکته باید توجه داشت باشید که در صورت استفاده از فرامین CANRX, CANTX, CANFLS برای اجرای فرامین بعدی باید منتظر بمانید تا فلگ خاص M1066 برابر با 1 شود

سپس می‌توانید فرمان بعدی CANRX یا CANTX یا CANFLS را اجرا کنید. این روند تا زمانی که PLC در وضعیت RUN است، ادامه دارد.

سپس برنامه را به اینورتر دالود کنید. به این نکته باید توجه داشته باشید که فرمت پیش‌فرض ارتباطات PLC داخلی اینورتر برابر با ASCII 7N2 9600 بوده و شماره ایستگاه اینورتر در شبکه برابر با 2 می‌باشد. بنابراین، فرمت ارتباطات اینورتر باید در نرم‌افزار WPLSoft و از مسیر Setting→Communications Setting تنظیم شود.

مرحله 5 ← تنظیم شماره ایستگاه اینورترهای Slave و سرعت تبادل دیتا و منبع کنترل و منبع فرمان

اینورتر مدل VFD-C2000 دلتا و اینورتر سری EC از شبکه‌ی CANopen موسوم به CANopen Communications interface driver پشتیبانی می‌کنند. پارامترهای مربوط به تنظیمات شماره ایستگاه اینورتر در شبکه سرعت تبادل دیتا و غیره در جدول زیر آورده شده است:

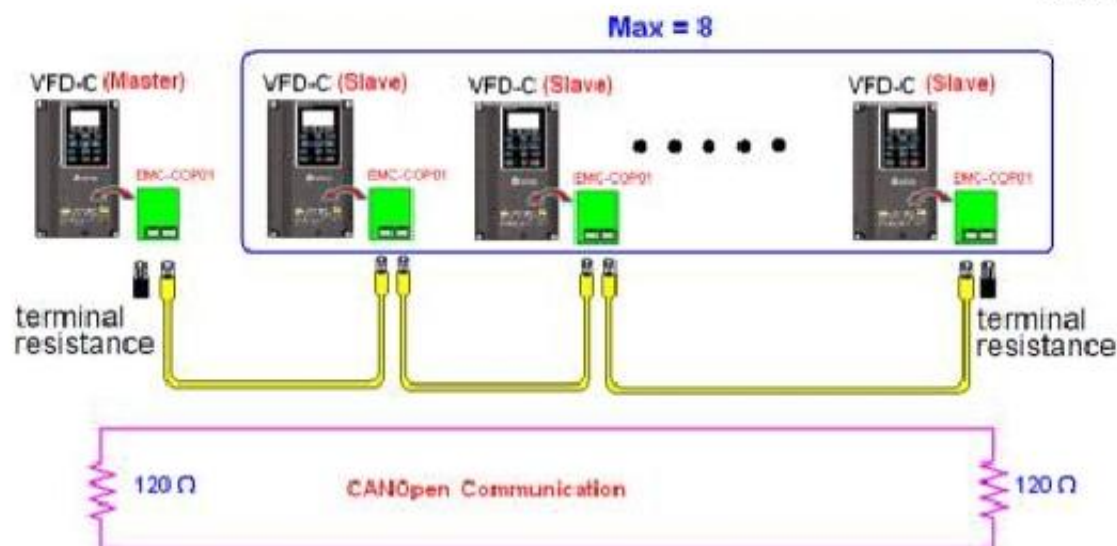
	Corresponding device parameters		Value	Definition
	C2000	E-C		
Slave station Address	09-36	09-20	0	Disable CANopen hardware interface
			1~127	CANopen Communication address
Communication Speed	09-37	09-21	0	1M
			1	500K
			2	250K
			3	125K
			4	100K
			5	50K
Control source	00-21	-	3	
	-	02-01	5	
Frequency source	00-20	-	6	
	-	02-00	5	
Torque source	11-33	-	3	
	-	-	-	
Position source	11-40	-	3	
	-	-	-	

سرودرایو سری ASDA-A2 دلتا نیز از واسط ارتباطات CANopen پشتیبانی می‌کند. پارامترهای مربوط به تنظیمات شبکه‌ی CANopen سرودرایو A2 در جدول زیر آورده شده است:

	Corresponding device Parameters	Value	Definition
	A2		
Slave station Address	03-00	1~127	CANopen Communication address
Communication speed	03-01 bit 8-11 XRX	R= 0	125K
		R= 1	250K
		R= 2	500K
		R= 3	750K
		R= 4	1M
Control/command source	01-01	B	

مرحله‌ی 6 ← اتصالات و سیم‌بندی سخت‌افزاری

در هنگام سیم‌بندی شبکه‌ی CANopen به این نکته باید توجه داشت باشید که دو انتهای باز شبکه را باید با ترمینال مقاومتی موسوم به ترمیناتور (ترمینیتور) ببندید تا شبکه‌ی CANopen به درستی عمل کند. روش اتصال شبکه‌ی CANopen به اینورتر VFD-C2000 در شکل زیر آورده شده است:



مرحله‌ی 7 ← کنترل و مقداردهی اولیه

بعد از نوشتن برنامه و دانلود و نوشتن آن در PLC داخلی اینورتر، PLC داخلی اینورتر را به مد RUN کلیک نمایید. تنها اینورتر Master را یک بار ریستارت نمایید. در این حالت، باید اینورترهای

Slave خاموش باشند، یعنی تغذیه‌ی آنها قطع باشد. این امر باعث می‌شود که پروتکل CANopen Master در اینورتر Master مقداردهی اولیه شود. در ادامه، می‌خواهیم مثالی از راه‌اندازی اینورتر VFD-C2000 در شبکه‌ی CANopen را مرحله به مرحله و به شکل جامع توضیح دهیم.

مثال

1. فعال کردن توابع CANopen Master

پارامتر Pr.09-45 که مربوط به مقداردهی اولیه‌ی توابع Master می‌باشد را برابر با 1 تنظیم نمایید. بعد از تکمیل شدن تنظیمات یک بار اینورتر را ریستارت نمایید. در نوار وضعیت کی‌پد مدل KPC-CC01 باید متن CAN Master نمایش داده شود. برای ریست کردن PLC داخلی اینورتر پارامتر Pr.00-02 را برابر با 6 تنظیم نمایید. دقت داشته باشید که با این کار برنامه‌ی موجود در PLC ریست شده و مقدار رجیسترهای PLC به مقدار پیش‌فرض باز خواهد گشت.

مجدداً یکبار اینورتر را ریستارت نمایید.

از کی‌پد دیجیتال KPC-CC01 برای تنظیم مد کاری PLC به مد PLC Stop استفاده نمایید. اگر از کی‌پد دیجیتال KPC-CE01 استفاده می‌کنید، باید مد کاری PLC را برابر با PLC2 تنظیم کنید تا PLC استوپ شود. اگر برای اولین بار از اینورتر استفاده می‌کنید، PLC داخلی اینورتر بدون برنامه بوده و با استوپ کردن PLC کد هشدار PLFF نمایش داده خواهد شد که این کد بیانگر بدون برنامه بودن PLC داخلی اینورتر می‌باشد.

2. حافظه‌ی ارتباطات اینورتر Master

نرم‌افزار WPLSoft را در کامپیوتر خود اجرا نمایید.

با استفاده از کی‌پد مد کاری PLC را برابر با Stop (PLC2) تنظیم نمایید.

توسط نرم‌افزار WPLSoft مقادیر رجیسترهای D2000~D2799, D1070~D1099 را بخوانید.

مقدار رجیستر D2000=10 و D2100=11 تنظیم نمایید.

مقدار رجیسترهای خاص D2100, D2200, D2300, D2400, D2500, D2600, D2700 را برابر با صفر تنظیم نمایید.

تنظیمات انجام شده برای رجیسترهای D2000~D2799 را به PLC دانلود نمایید.

3. تنظیم شماره ایستگاه ارتباطات و سرعت تبادل دیتا برای اینورتر Master شماره ایستگاه اینورتر Master باید توسط پارامتر Pr.09-46 تنظیم شود. مقدار پیش‌فرض این پارامتر 100 است. توجه داشته باشید که نباید این شماره با شماره ایستگاه اینورترهای Slave یکسان باشد. این به آن معنی است که شماره ایستگاه اینورتر Master و اینورترهای Slave باید در کل برنامه منحصر بفرد باشند.

سرعت تبادل دیتای شبکه‌ی CANopen توسط پارامتر Pr.09-37 تنظیم خواهد شد. مقدار پیش‌فرض این پارامتر برابر با 0 است. این به آن معنی است که سرعت تبادل دیتا به صورت پیش‌فرض 1M می‌باشد. توجه داشته باشید که سرعت تبادل دیتا در کل شبکه برای تمام اینورترها اعم از Master و Slave ها باید یکسان باشد.

4. نوشتن کد برنامه

در قابلیت دسترسی بلادرنگ می‌توان مستقیماً ناحیه رجیسترهای D را از PLC خوانده و یا مقداری را در ناحیه رجیسترهای D نوشت. در قابلیت دسترسی با وقفه (Non Real-time) می‌توان از فرامین نوشتن، خواندن و به‌روزرسانی استفاده کرد که در ادامه، هر یک را توضیح خواهیم داد. فرمان Read ← برای خواندن مقادیر ناحیه‌ی رجیسترهای D می‌توان از فرمان CANRX استفاده کرد.

وقتی که در این وضعیت عمل خواندن مقادیر به پایان رسید، فلگ خاص M1066 برابر با 1 خواهد شد. وقتی که عمل خواندن مقادیر با موفقیت به پایان رسید، فلگ M1067 برابر با 1 خواهد شد و هنگامی که در هنگام خواندن دیتا خطایی رخ داد، فلگ M1067 برابر با 1 خواهد شد. فرمان Write ← برای نوشتن مقدار در ناحیه‌ی رجیسترهای D می‌توان از فرمان CANTX استفاده کرد.

وقتی که در این وضعیت عمل نوشتن مقادیر به پایان رسید، فلگ خاص M1066 برابر با 1 خواهد شد. وقتی که عمل نوشتن مقادیر با موفقیت به پایان رسید، فلگ M1067 برابر با 1 خواهد شد و هنگامی که در هنگام نوشتن دیتا خطایی رخ داد، فلگ M1067 برابر با 1 خواهد شد. فرمان Refresh ← برای به‌روزرسانی مقادیر ناحیه رجیسترهای D می‌توان از فرمان CANFLS استفاده کرد. اگر رجیستر D دارای خاصیت RW بود، اینورتر Master مقداری را در اینورتر Slave خواهد نوشت. اگر رجیستر D دارای خاصیت RO باشد، اینورتر Slave مقدار خوانده شده را به اینورتر Master باز خواهد گرداند.

وقتی که در این وضعیت عمل به‌روزرسانی مقادیر به پایان رسید فلگ خاص M1066 برابر با 1 خواهد شد. وقتی که عمل به‌روزرسانی مقادیر با موفقیت به پایان رسید، فلگ M1067 برابر با 1 خواهد شد و هنگامی که در هنگام به‌روزرسانی دیتا خطایی رخ داد، فلگ M1067 برابر با 1 خواهد شد. به این نکته باید توجه داشته باشید که در صورت استفاده از فرامین CANRX, CANTX, CANFLS برای اجرای فرامین بعدی باید منتظر بمانید تا فلگ خاص M1066 برابر با 1 شود سپس می‌توانید فرمان بعدی CANRX یا CANTX یا CANFLS را اجرا کنید. این روند تا زمانی که PLC در وضعیت RUN است ادامه دارد.

سپس برنامه را به اینورتر دالود کنید. به این نکته باید توجه داشته باشید که فرمت پیش‌فرض ارتباطات PLC داخلی اینورتر برابر با ASCII 7N2 9600 بوده و شماره ایستگاه اینورتر در شبکه برابر با 2 می‌باشد. بنابراین، فرمت ارتباطات اینورتر باید در نرم‌افزار WPLSoft و از مسیر Setting→Communications Setting تنظیم شود.

5. تنظیم شماره ایستگاه و سرعت تبادل دیتا برای اینورترهای Slave

شماره ایستگاه نخستین اینورتر Slave را برابر با 10 تنظیم نمایید (Pr.09-36=10, Node ID 10) و سرعت تبادل دیتای آن را برابر با 1Mbps تنظیم نمایید (Pr.09-37=0, Speed 1M). شماره ایستگاه دومین اینورتر Slave را برابر با 11 تنظیم نمایید (Pr.09-36=11, Node ID 11) و سرعت تبادل دیتای آن را برابر با 1Mbps تنظیم نمایید (Pr.09-37=0, Speed 1M).

6. سیم‌بندی سخت‌افزاری شبکه‌ی CANopen برای اینورتر VFD-C2000

در هنگام سیم‌بندی شبکه‌ی CANopen به این نکته باید توجه داشته باشید که دو انتهای باز شبکه را باید با ترمینال مقاومتی موسوم به ترمیناتور (ترمینیتور) ببندید تا شبکه‌ی CANopen به درستی عمل کند. روش اتصال شبکه‌ی CANopen به اینورتر VFD-C2000 در شکل زیر آورده شده است:



7. مقداردهی اولیه‌ی بخش کنترل اینورتر Master

بعد از نوشتن برنامه و دانلود و نوشتن آن در PLC داخلی اینورتر، PLC داخلی اینورتر را به مد RUN کلیک نمایید. تنها اینورتر Master را یک بار ریستارت نمایید. در این حالت باید اینورترهای Slave خاموش باشند یعنی تغذیه‌ی آنها قطع باشد. این امر باعث می‌شود که پروتکل CANopen Master در اینورتر Master مقداردهی اولیه شود.

در ادامه، می‌خواهیم مثالی از راه‌اندازی اینورتر VFD-C2000 در شبکه‌ی CANopen را مرحله به مرحله بصورت جامع توضیح دهیم.

8-9-4 کنترل گره (Node) اصلی در ارتباطات داخلی

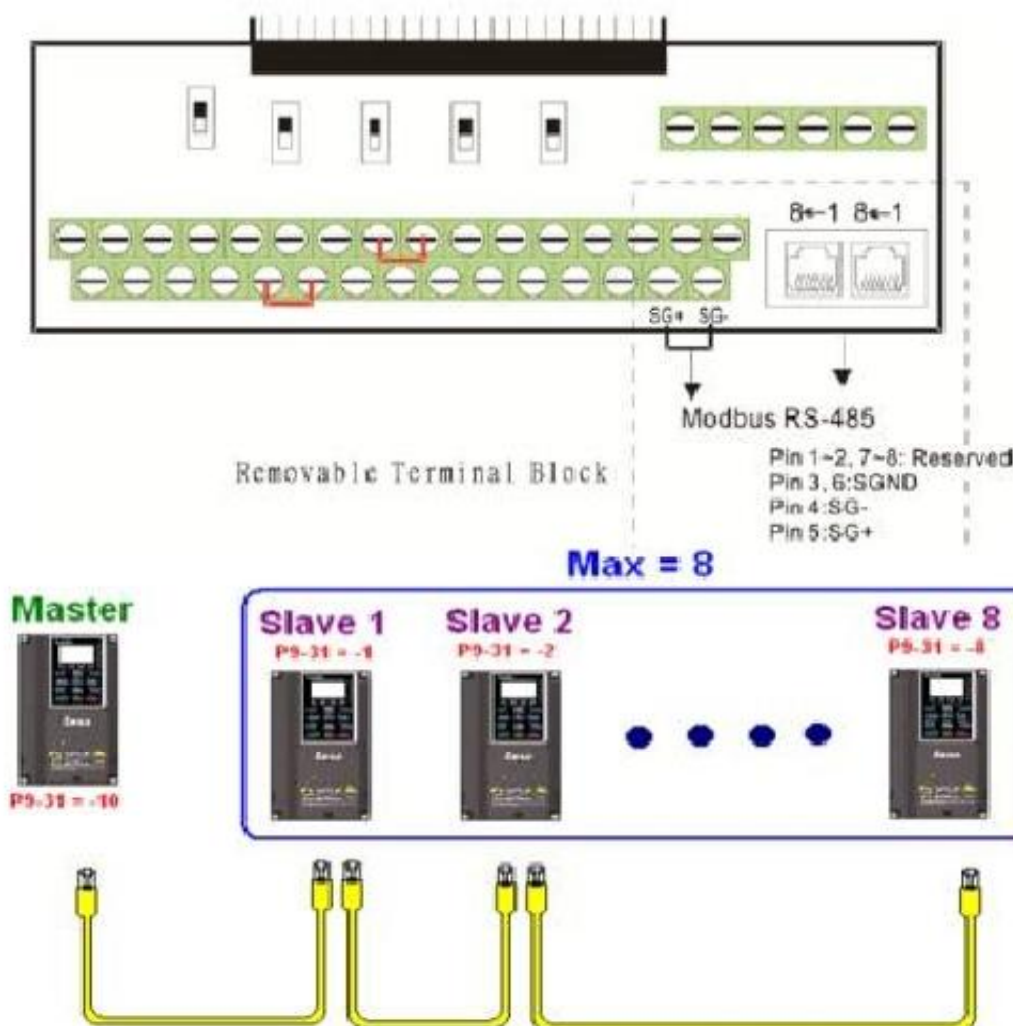
به منظور ساده کردن استفاده از پروتکل CANopen پروتکلی به نام 485 در این اینورتر توسعه داده شده است. این پروتکل تنها در اینورتر VFD-C2000 و VFD-CT2000 استفاده شده است. حداکثر تعداد دستگاه Slave قابل اتصال به این پروتکل 8 دستگاه می‌باشد.

ارتباطات داخلی دارای یک ساختار Master-Slave است. در این پروتکل روش مقداردهی اولیه بسیار ساده می‌باشد.

دستگاه Slave ← برای اینورترهای Slave باید پارامتر Pr.09-31 را برابر با 1 - الی 8 - تنظیم نمایید. بنابراین، از 1 - الی 8 - می‌توان به 8 اینورتر Slave دسترسی داشت. برای تعریف منبع کنترل توسط پروتکل 485 باید پارامتر Pr.00-20=1 تنظیم شود. همچنین، برای فرمان سرعت Pr.00-21=2، فرمان گشتاور Pr.11-33=1 و فرمان موقعیت Pr.11-40=2 تنظیم شود. در این صورت، تنظیمات اینورترهای Slave به پایان خواهد رسید. توجه داشته باشید که به غیر از پارامتر Pr.09-31 باید مقدار پارامترهای ذکر شده در این بخش در تمام اینورترهای Slave یکسان باشد. توجه داشته باشید که نیاز به فعال کردن توابع PLC نیست.

سیستم ← تنظیمات اینورتر Master نیز مانند اینورترهای Slave می‌باشد، با این تفاوت که مقدار پارامتر Pr.09-31 را باید برابر با 10 - تنظیم نمایید تا تابع PLC فعال شود.

سیم‌بندی سخت‌افزاری ← توسط پورت سریال 485 باید اینورتر Master و اینورترهای Slave را به یکدیگر متصل نمایید. در اینورتر VFD-C2000 دلتا دو نوع واسطه پورت سریال 485 وجود دارد. در شکل زیر، می‌توانید این دو واسطه سریال 485 را مشاهده کنید.



برنامه‌نویسی اینورتر Master ← در برنامه‌نویسی PLC داخلی اینورتر می‌توانید از رجیستر خاص D1110 برای تعریف یک ایستگاه Slave برای کنترل کردن آن ایستگاه استفاده کنید. همانطور که گفته شد، هشت اینورتر Slave را از 1 الی 8 می‌توان به Master متصل کرد. اگر مقدار رجیستر خاص D1110=0 تنظیم شود، برنامه بین هر 8 ایستگاه پرش خواهد کرد. سپس، فلگ خاص M1035 را برابر با 1 منطقی (ON) نمایید. در این صورت، موقعیت‌های حافظه‌ی اینورتر Master و اینورترهای Slave با هم منطبق (برابر) خواهند شد. در این هنگام Master تنها نیاز به ارسال فرامین به آدرس ایستگاه Slave جهت کنترل موقعیت Slave دارد. در جداول زیر، رجیسترهای متصل شده به ارتباطات داخلی لیست شده است.

در جدول زیر، فلگ خاص M مربوط به کنترل ارتباطات داخلی با پروتکل 485 توسط PLC داخلی اینورتر لیست شده است:

Special M	Description of Function	Attributes
M1035	Initiates internal communications control	RW

در جدول زیر، رجیسترهای خاص D مربوط به کنترل ارتباطات داخلی 485 توسط PLC داخلی اینورتر لیست شده است:

Special D	Description of Function	Attributes
D1110	Internal node communications number 1-8 (set the station number of the slave station to be controlled)	RW

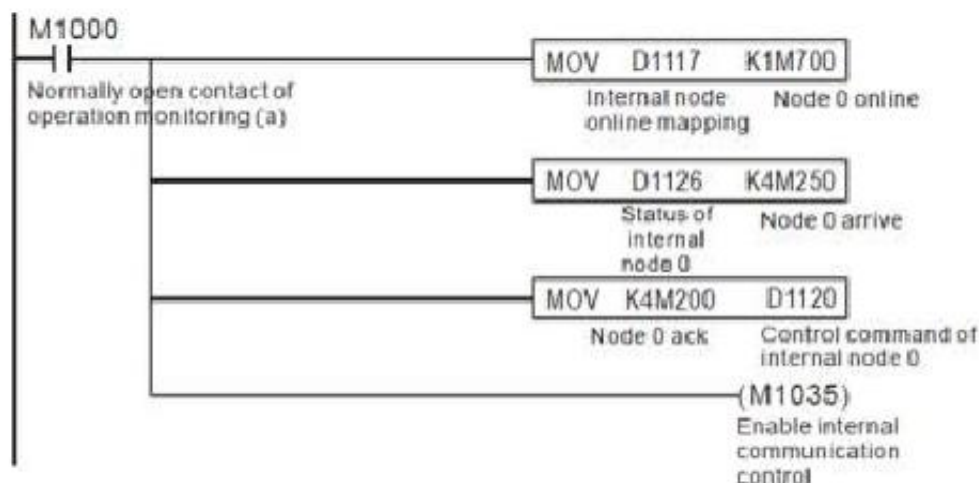
Special D	Description of Function							Attributes
	Definition	bit	User rights	Speed mode	Location mode	Torque mode	Homing mode	
D1120 + 10*N	Internal node N control command	0	4	Command Functions	-	-	Homing Origin	RW
		1	4	Reverse rotation requirements	Immediate Change	-	-	
		2	4	-	-	-	-	
		3	3	Temporary Pause	Temporary Pause	-	-	
		4	4	Frequency Locking	-	-	Temporary pause	
		5	4	JOG	-	-	-	
		6	2	Quick Stop	Quick Stop	Quick Stop	Quick Stop	
		7	1	Servo ON	Servo ON	Servo ON	Servo ON	
		11~8	4	Speed interval Switching	Speed interval Switching	-	-	
		13~12	4	Deceleration time change	-	-	-	
		14	4	Enable Bit 13~ 8	Enable Bit 13~ 8	-	-	
		15	4	Clear error code	Clear error code	Clear error code	Clear error code	
D1121 + 10*N	Internal node N control Mode			0	1	2	3	RW
D1122 + 10*N	Internal node N reference command L			Speed command (no number)	Position command (with numbers)	Torque command (with numbers)	-	RW
D1123 + 10*N	Internal node N reference command H			-			-	RW

در جدول زیر رجیسترهای خاص D مربوط به نمایش وضعیت ارتباطات داخلی 485 توسط PLC داخلی اینورتر لیست شده است:

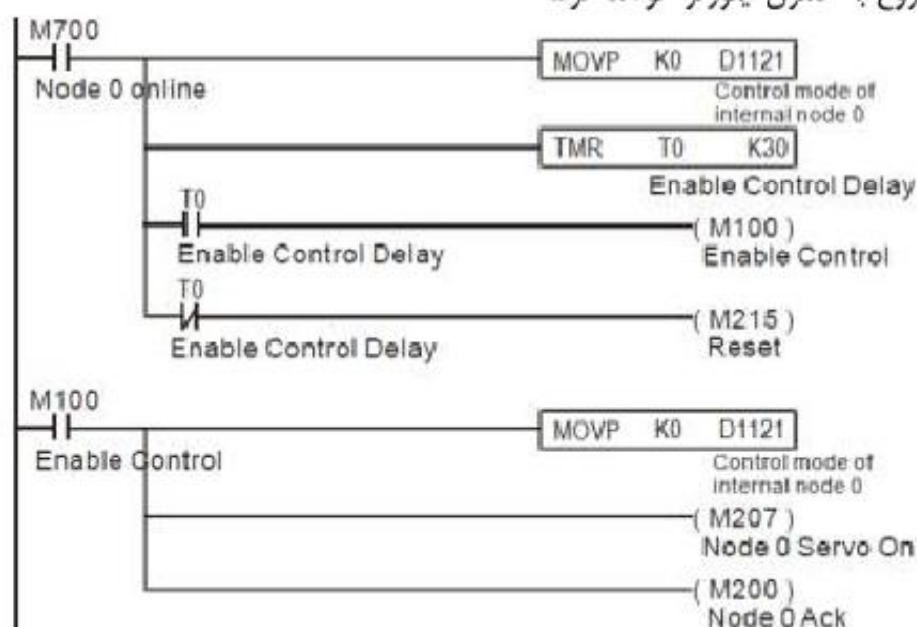
Special D	Description of Function	Attributes
D1115	Internal node synchronizing cycle (ms)	RO
D1116	Internal node error (bit0 = slave device 1, bit1 = slave device 2,...bit7 = slave device 8)	RO
D1117	Internal node online correspondence (bit0 = slave device 1, bit1 = slave device 2,...bit7 = slave device 8)	RO

Special ID	Description of Function					Attributes
	bit	Speed mode	Location mode	Torque mode	Homing mode	
D1126 + 10*N	0	Frequency command Arrival	Position command attained	Torque command Attained	Zero command Completed	RO
	1	Clockwise	Clockwise	Clockwise	Clockwise	
		Counterclockwise	Counterclockwise	Counterclockwise	Counterclockwise	
		Warning	Warning	Warning	Warning	
	3	Error	Error	Error	Error	
	5	JOG				
	6	Quick Stop	Quick Stop	Quick Stop	Quick Stop	
	7	Servo ON	Servo ON	Servo ON	Servo ON	
D1127 + 10*N		Actual frequency	Actual position (with numbers)	Actual torque (with numbers)	-	RO
D1128 + 10*N		-		-	-	

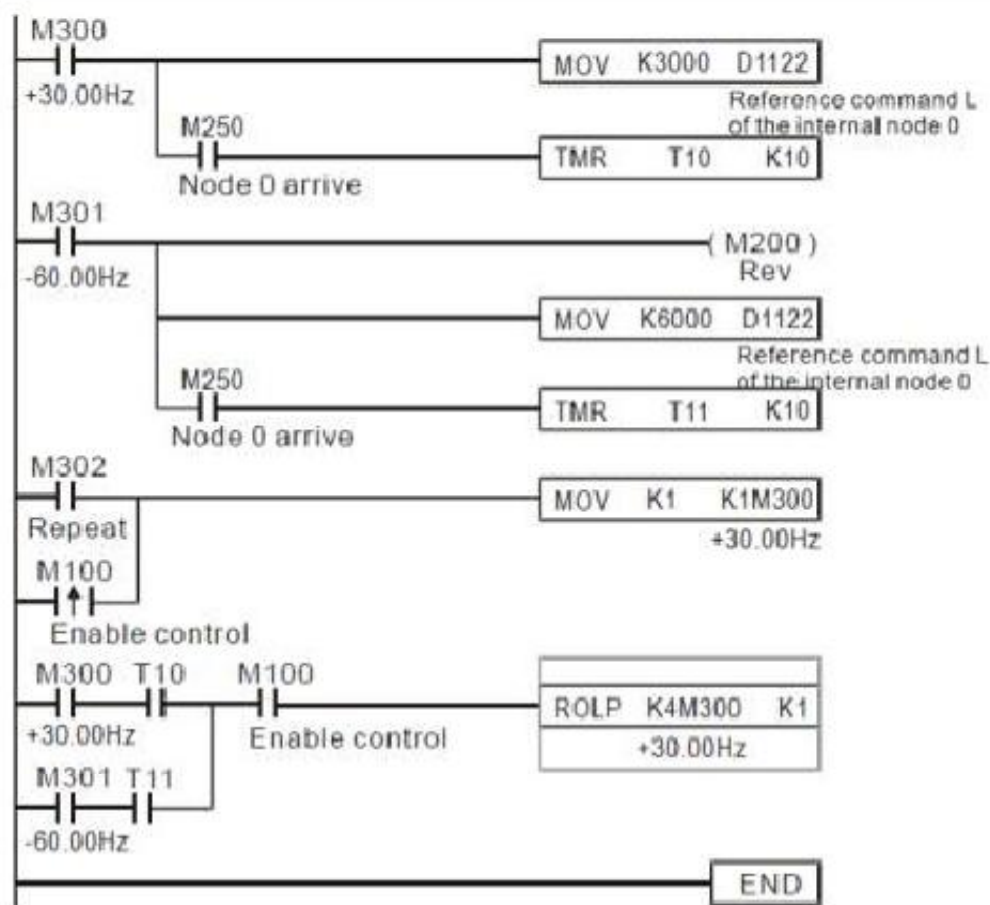
فرض کنید می‌خواهیم عملکرد اینورتر **Slave** با آدرس 1 را در فرکانس‌های 60.00Hz, 30.00Hz وضعیت اینورتر و **Node** آنلین را بررسی نماییم. برای این کار برنامه‌ای مطابق با شکل زیر خواهیم نوشت:



وقتی که اینورتر Master وضعیت آنلاین بودن اینورتر Slave شماره 1 را بررسی کرد، بعد از 3 ثانیه تأخیر شروع به کنترل اینورتر خواهد کرد.



اکنون بعد از 1 ثانیه اینورتر Slave شماره 1 با فرکانس 30.00Hz و در جهت چرخش به جلو شروع و بعد از 1 ثانیه با فرکانس 60.00Hz و در جهت چرخش به عقب شروع به کار می‌کند و این حالت متناوباً تکرار خواهد شد.



8-9-5 شبکه‌ی Modbus و کنترل از راه دور ورودی/خروجی‌ها توسط

دستورالعمل MODRW

PLC داخلی اینورتر VFD-C2000 از توابع خواندن و نوشتن پورت سریال RS-485 پشتیبانی می‌کند. بنابراین، می‌توان از دستورالعمل MODRW در این PLC استفاده کرد. قبل از نوشتن برنامه در نرم‌افزار PLC باید قابلیت دسترسی PLC داخلی اینورتر به پورت سریال 485 را تعریف کرد. برای این کار باید ابتدا پارامتر Pr.09-31 را برابر با 12 - تنظیم نمایید. بعد از تکمیل عمل تنظیمات از توابع استاندارد تعریف شده برای پورت سریال RS-485 می‌توان برای اجرای دستور خواندن و یا نوشتن مقدار از/ یا در دستگاه‌های متصل شده به شبکه‌ی مدباس استفاده کرد. سرعت تبادل دیتا در شبکه را می‌توان در پارامتر Pr.09-01 تعریف کرد. فرمت ارتباطات شبکه‌ی مدباس در پارامتر Pr.09-04 و شماره‌ی ایستگاه جاری PLC در پارامتر Pr.09-35 تعریف می‌شود. به غیر از شماره ایستگاه جاری PLC باید تنظیمات پارامترهای گفته شده در کل شبکه یکسان باشد، ولی شماره ایستگاه PLC باید در کل شبکه منحصر بفرد بوده و تکراری نباشد. اینورتر VFD-C2000 از توابع ذکر شده در زیر پشتیبانی می‌کند.

Read coil	→	(0×01)
Read input	→	(0×02)
Read register	→	(0×03)
Write to single register	→	(0×06)
Write to several coils	→	(0×0F)
Write to several registers	→	(0×10)

نحوه‌ی استفاده از این توابع در جدول زیر توضیح داده شده است :

MODRW command					General meaning	Slave device is Delta's PLC meaning	Slave device is Delta's converter meaning
S1	S2	S3	S4	S5			
Node ID	Command	Address	Return: D area	Length			
K3	H01	H500	D0	K18	Read coil (Bit)	18 بیت دیتا از ایستگاه Slave شماره 3 که یک PLC دلتا است، خوانده خواهد شد. این دیتا Y0~Y21 می‌باشند. این دیتا بعد از خوانده شدن در بیت 0~15 رجیستر D0 و بیت 0~3 رجیستر D1 در همین ایستگاه (PLC داخلی اینورتر) ذخیره می‌شوند.	Does not support this function
K3	H02	H400	D10	K10	Read input (Bit)	10 بیت دیتا از ایستگاه Slave شماره 3 که یک PLC دلتا می‌باشد، خوانده شده که این دیتا X0~X11 می‌باشند. این دیتا بعد از خوانده شدن در بیت 0~9 رجیستر شماره D10 همین ایستگاه (PLC داخلی اینورتر) ذخیره می‌شوند.	Does not support this function
K3	H03	H600	D20	K3	Read register (word)	3 کلمه (word) دیتا از ایستگاه Slave شماره 3 که یک PLC دلتا می‌باشد، خوانده شده که این دیتا T0~T2 می‌باشند. این دیتا بعد از خوانده شدن در رجیسترهای D20~D22 همین ایستگاه (PLC داخلی اینورتر) ذخیره می‌شوند.	Read 3 words of data corresponding to slave station 3 converter parameters 06-00 to 06-02. This data is stored by D20 to D22
K3	H06	H610	D30	XX	Write to single register (word)	PLC داخلی اینورتر مقدار موجود در رجیستر D30 را در تایمر T16 ایستگاه شماره 3 که یک PLC دلتا است، خواهد نوشت.	Write slave station 3 converter 06 to 16 parameter to this station's D30 value
K3	H0F	H509	D40	K10	Write to multiple coils (Bit)	PLC داخلی اینورتر مقدار موجود بیت‌های 0~9 رجیستر D40 را در بوبین‌های Y11~Y22 ایستگاه شماره 3 که یک PLC دلتا است، خواهد نوشت.	Does not support this function

K3	H10	H602	D50	K4	Write to multiple registers (word)	PLC داخلی اینورتر مقدار موجود در رجیسترهای D50~D53 را در تایمرهای T2~T5 ایستگاه شماره 3 که یک PLC دلتا است، خواهد نوشت.	Write slave station 3 converter 06-02 to 06-05 parameters to this station's D50 to D53
----	-----	------	-----	----	------------------------------------	---	--

در جدول بالا ×× بیانگر مهم نبودن طول داده می‌باشد.

بعد از اجرای دستورالعمل MODRW، وضعیت عملکرد مورد نظر را می‌توان در فلگ خاص M1077 (تکمیل شدن عمل خواندن یا نوشتن مقدار در شبکه‌ی RS-485)، فلگ M1078 (خطا در عمل خواندن یا نوشتن در شبکه‌ی RS-485)، و فلگ خاص M1079 (وقفه در عمل خواندن یا نوشتن شبکه‌ی RS-485) را بررسی کرد. بعد از اینکه اجرای دستورالعمل MODRW به پایان رسید، وضعیت فلگ M1077 برابر با 0 خواهد شد. توجه داشته باشید که در هنگام اجرای دستورالعمل MODRW فلگ M1077 فعال ON خواهد شد، اگر هیچ خطایی و یا یک خطا و یا وقفه در عملکرد خواندن یا نوشتن در شبکه‌ی RS-485 رخ دهد نیز، این فلگ در هنگام اجرای دستورالعمل MODRW فعال ON خواهد بود.

به عنوان مثال، می‌خواهیم برنامه‌ای بنویسیم که توابع مختلف را تست کند.

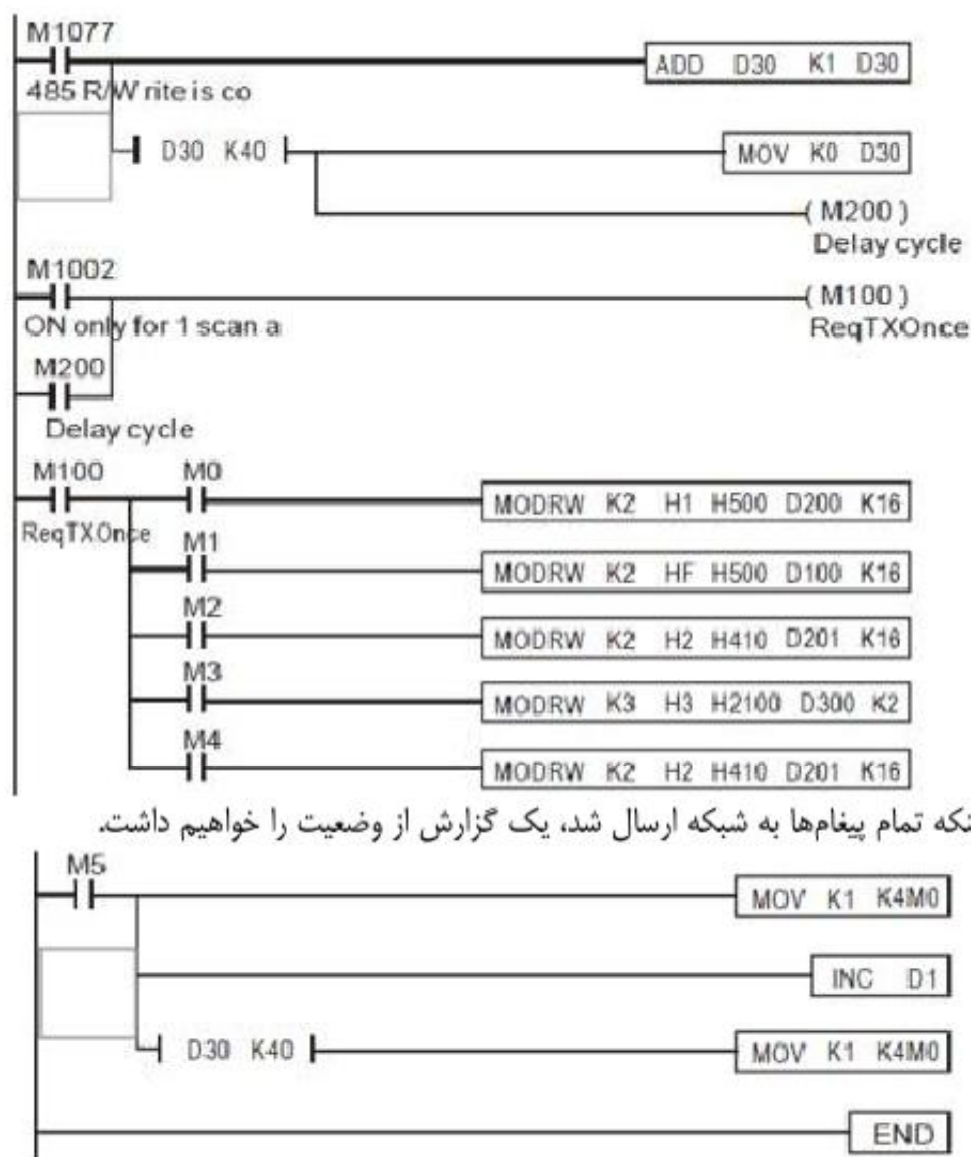
در هنگام شروع به کار (RUN شدن) PLC داخلی اینورتر می‌خواهیم به ترتیب فلگ M0~M15 را فعال کنیم.



وقتی که پیام گزارش شده در شبکه‌ی MODBUS بیانگر عدم وجود خطا بود، برنامه به دستورالعمل بعدی منتقل خواهد شد.



اگر در هنگام تبادل دیتا وقفه‌ای رخ داد و یا اینکه خطایی رخ داد، فلگ M1077 به وضعیت ON تغییر خواهد کرد. در این هنگام، بعد از گذشت 30 بار سیکل اسکن برنامه مجدداً دستورالعمل پیام را ارسال خواهد کرد.



بعد از اینکه تمام پیام‌ها به شبکه ارسال شد، یک گزارش از وضعیت را خواهیم داشت.

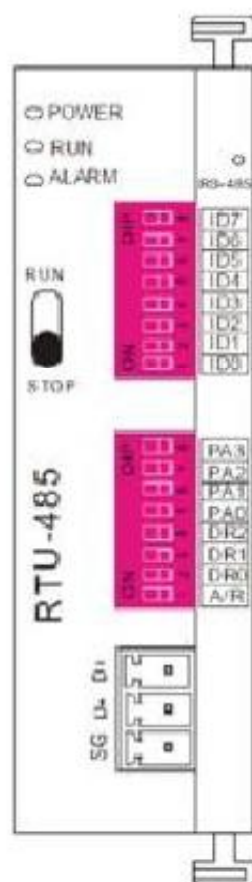
یک مثال کاربردی از شبکه‌ی Modbus که در آن از اینورتر VFD-C2000 استفاده شده است را بررسی خواهیم کرد.

معمولاً برای کنترل اینورتر VFD-C2000 دلتا توسط PLC‌های دلتا از ماژول RTU-485 در کنار PLC دلتا استفاده می‌شود. در ادامه، مرحله به مرحله پیکربندی و برنامه‌نویسی این مثال را بررسی خواهیم کرد:

مرحله‌ی 1 ← فرمت ارتباطات را تنظیم نمایید. در این مثال، فرض شده فرمت ارتباطات برابر با 115200, 8, N, 2, RTU است. در اینورتر VFD-C2000 شماره ایستگاه پیش‌فرض PLC داخلی برابر با 2 می‌باشد. این شماره را می‌توانید در پارامتر Pr.09-35 تغییر دهید. با تنظیم پارامتر

Pr.09-31 برابر با 12 - پورت COM1 توسط PLC داخلی اینورتر کنترل خواهد شد. با تنظیم پارامتر Pr.09-01 برابر با 115.2 سرعت تبادل دیتای اینورتر در شبکه‌ی مدباس برابر با 115200 شده و در نهایت، با تنظیم پارامتر Pr.09-04 برابر با 13 فرمت ارتباطات مدباس برای اینورتر برابر با 8, N, 2, RTU تنظیم خواهد شد.

در ماژول RTU 485 که در کنار PLC دلتا متصل می‌شود، شماره ایستگاه را برابر با 8 تنظیم می‌نماییم. در شکل زیر، نحوه‌ی تنظیم شماره ایستگاه ماژول RTU-485 و فرمت ارتباطات آن آورده شده است.



Communication station #:
ID0 ~ ID7 are defined as $2^0, 2^1, 2^2 \dots 2^6, 2^7$

Communication protocol

PA3	PA2	PA1	PA0	A/R	Communication Protocol
OFF	OFF	OFF	OFF	ON	7.E.1 - ASCII
OFF	OFF	OFF	ON	ON	7.O.1 - ASCII
OFF	OFF	ON	OFF	ON	7.E.2 - ASCII
OFF	OFF	ON	ON	ON	7.O.2 - ASCII
OFF	ON	OFF	OFF	ON	7.N.2 - ASCII
OFF	ON	OFF	ON	ON	8.E.1 - ASCII
OFF	ON	ON	OFF	ON	8.O.1 - ASCII
OFF	ON	ON	ON	ON	8.N.1 - ASCII
ON	OFF	OFF	OFF	ON	8.N.2 - ASCII
OFF	ON	OFF	ON	OFF	8.E.1 - RTU
OFF	ON	ON	OFF	OFF	8.O.1 - RTU
OFF	ON	ON	ON	OFF	8.N.1 - RTU
ON	OFF	OFF	OFF	OFF	8.N.2 - RTU

DR2	DR1	DR0	Communication Speed
OFF	OFF	OFF	1,200 bps
OFF	OFF	ON	2,400 bps
OFF	ON	OFF	4,800 bps
OFF	ON	ON	9,600 bps
ON	OFF	OFF	19,200 bps
ON	OFF	ON	38,400 bps
ON	ON	OFF	57,600 bps
ON	ON	ON	115,200 bps

مرحله‌ی 2 ← در این مرحله، باید تجهیزات کنترلی را نصب کرد. در این مثال، ما به ترتیب تجهیزات گفته شده را در کنار ماژول RTU-485 به هم متصل می‌کنیم.

DVP16-SP (8 IN 8 OUT), DVP-04AD (4channels AD), DVP02DA (2channels DA), DVP-08ST (8 switches)

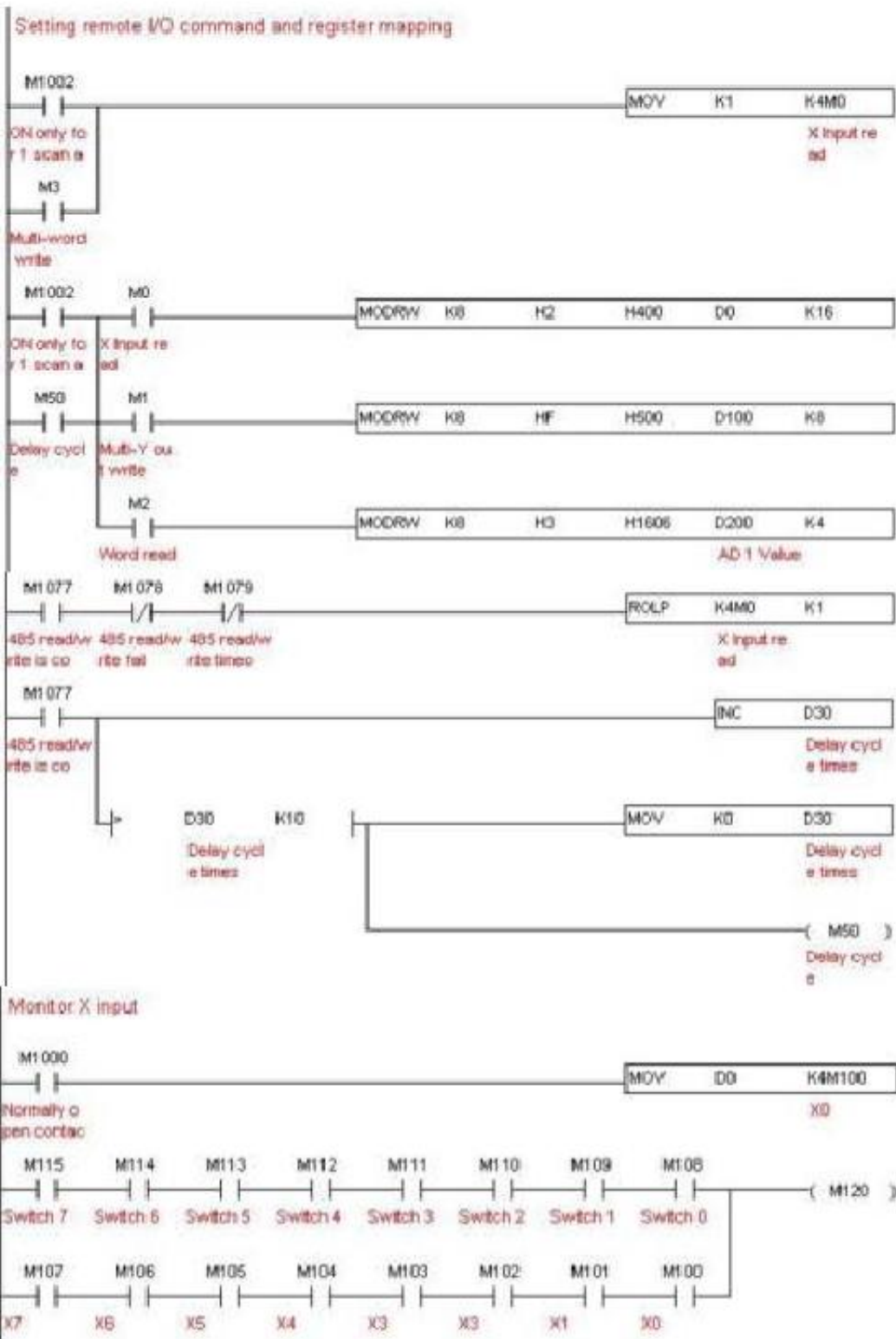
با این ترکیب اتصالات ماژول‌های دلتا در کنار هم و اتصال آنها به PLC داخلی اینورتر به ورودی خروجی‌های دیجیتال و آنالوگ توسعه یافته دست خواهیم یافت. آدرس هر یک از این ورودی خروجی‌های توسعه یافته برای PLC داخلی اینورتر مطابق با جدول زیر است:

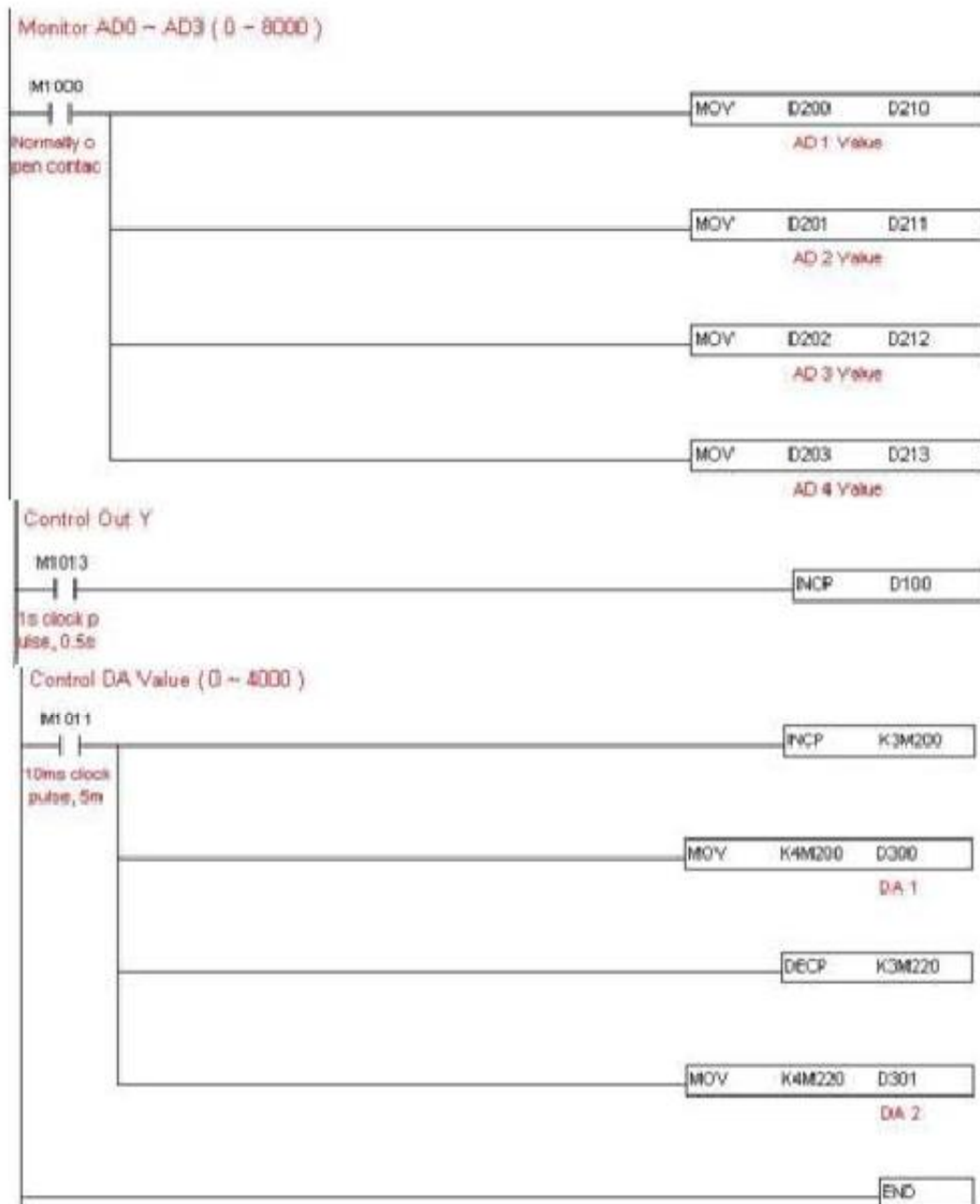
Module	Terminals	Address 485
DVP16-SP	X0 ~ X7	0400H ~ 0407H
	Y0 ~ Y7	0500H ~ 0507H
DVP-04AD	AD0 ~ AD3	1600H ~ 1603H
DVP02DA	DA0 ~ DA1	1640H ~ 1641H
DVP-08ST	Switch 0 ~ 7	0408H ~ 040FH

مرحله‌ی 3 ← پیکربندی سخت‌افزاری (فیزیکی) ماژول‌های توسعه با استفاده از ماژول RTU-485 برای PLC داخلی اینورتر VFD-C2000 دلتا را انجام می‌دهیم.

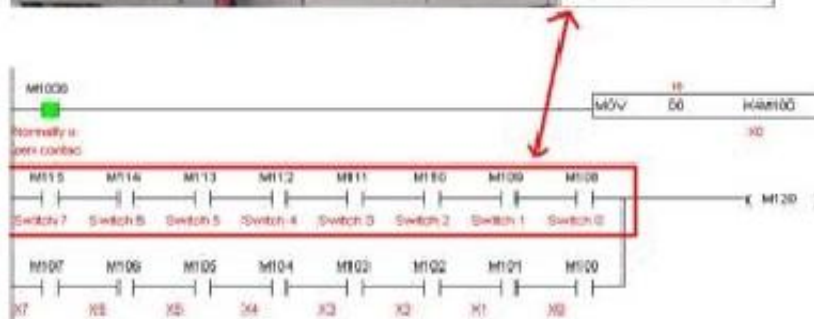
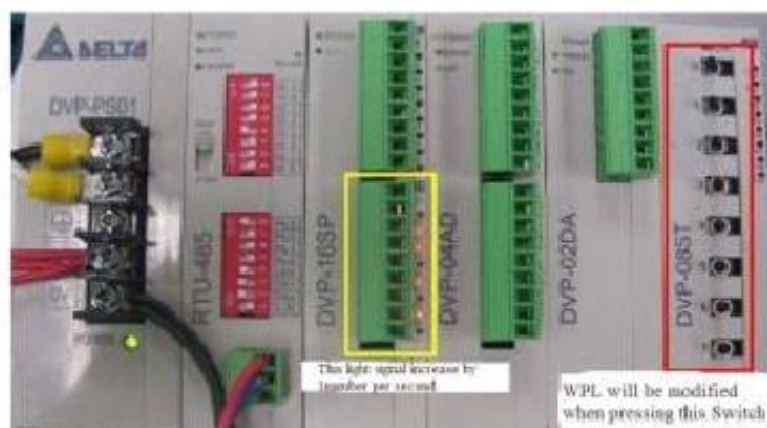


مرحله‌ی 4 ← برنامه‌ای مطابق با شکل زیر به زبان دیاگرام نردبانی در نرم‌افزار WPLSoft برای PLC داخلی اینورتر VFD-C2000 می‌نویسیم.

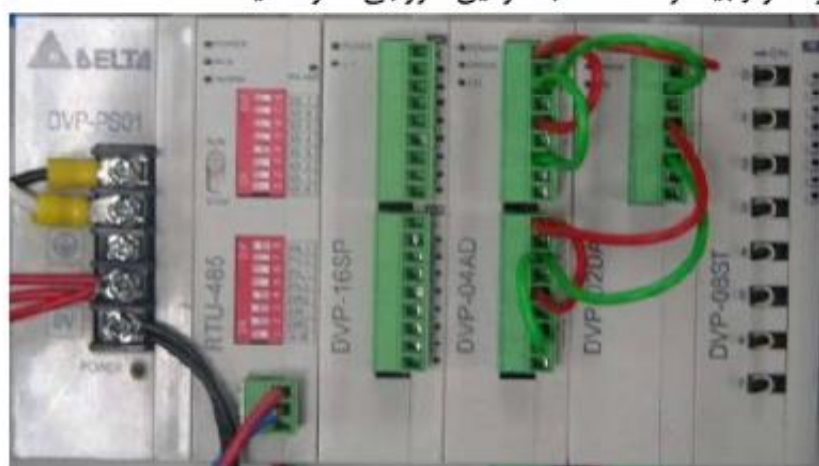




مرحله‌ی 5 ← اکنون می‌خواهیم ورودی خروجی‌های توسعه یافته‌ی دیجیتال اینورتر را تست نماییم. وقتی که سوئیچی را بر روی کارت توسعه‌ی DVP-08ST فعال کنیم، PLC داخلی اینورتر با استفاده از فلگ‌های تعریف شده M108~M115 برای ماژول فوق، تغییر وضعیت را در برنامه آشکارسازی می‌نماید. مطابق با برنامه نوشته شده برای PLC داخلی اینورتر می‌توانیم روشن شدن LEDهای وضعیت خروجی‌های کارت DVP-16SP را در هر ثانیه یک‌بار مشاهده کنیم. این نحوه‌ی نمایش LEDها به فرمت باینری می‌باشد.

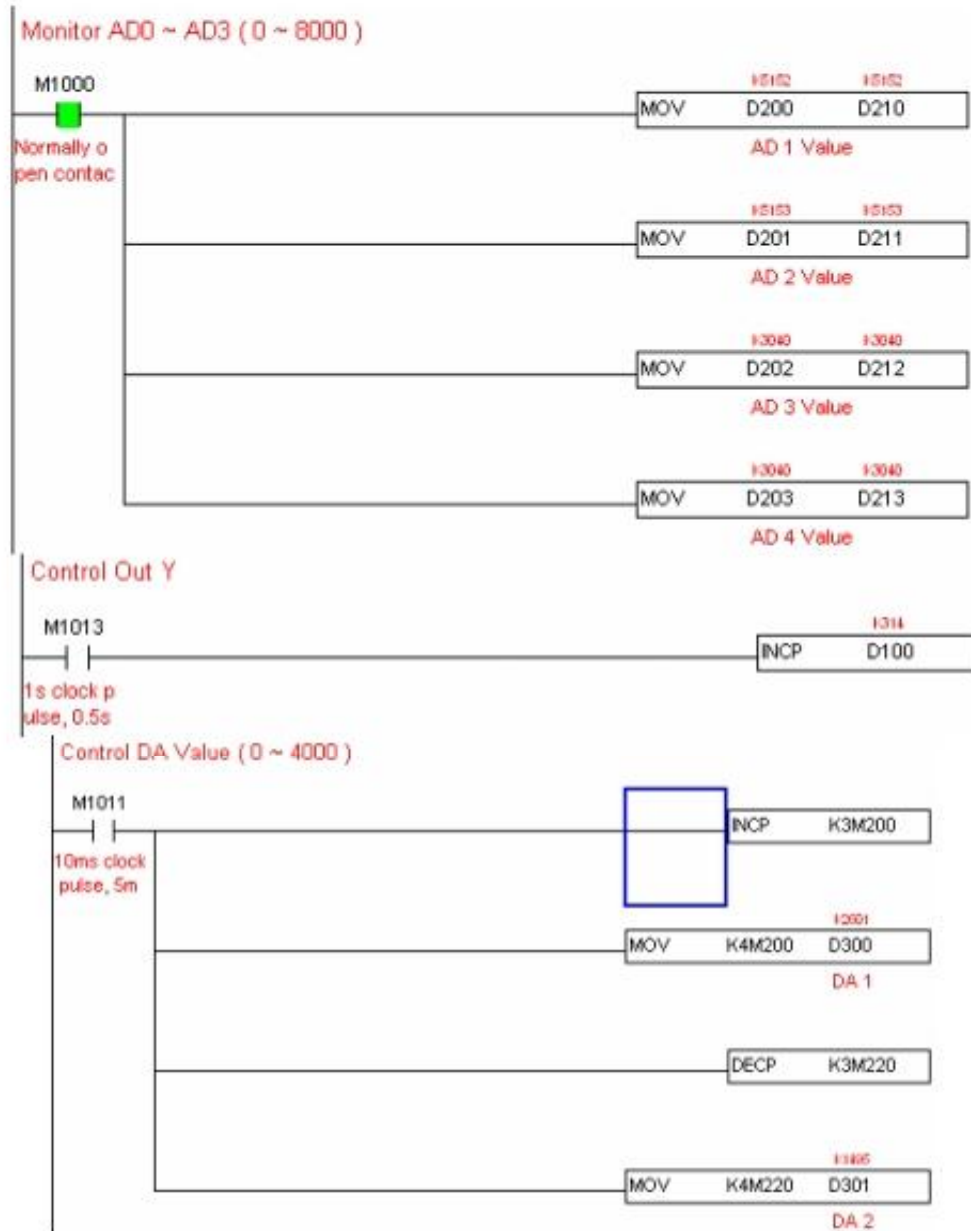


اکنون می‌خواهیم ورودی خروجی‌های آنالوگ توسعه یافته را بررسی نماییم. همانطور که در شکل زیر مشاهده می‌کنید، دو ورودی ابتدای آنالوگ را به اولین خروجی آنالوگ، و دو ورودی بعدی آنالوگ را به دومین خروجی آنالوگ متصل کرده‌ایم. رجیستر D200 به اولین ورودی آنالوگ، D201 به دومین ورودی آنالوگ، D202 به سومین ورودی آنالوگ، D203 به چهارمین ورودی آنالوگ، D300 به اولین خروجی آنالوگ و رجیستر D301 به دومین خروجی آنالوگ لینک شده است.



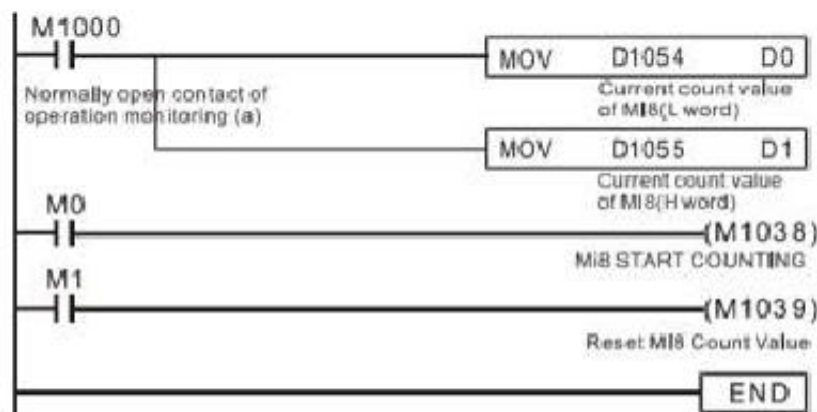
AD 1 — DA1
AD2 —

AD3 — DA 2
AD4 —



8- 10 تابع شمارش سرعت بالا با استفاده از ورودی دیجیتال M18

ورودی دیجیتال M18 اینورتر VFD-C2000 دلتا از روش شمارش پالس با حداکثر سرعت 100KHz پشتیبانی می‌کند. متد شروع شمارش بسیار ساده بوده و تنها نیاز به تنظیم فلگ خاص M1038 می‌باشد. دیتا رجیستر مربوط به شمارش پالس 32 بیتی بوده و مقدار شمرده شده در رجیسترهای خاص D1054, D1055 ذخیره خواهد شد. با 0 کردن فلگ M1039 مقدار شمرده شده ریست شده و رجیسترهای D1054, D1055 برابر با 0 خواهند شد.



وقتی که ورودی دیجیتال MI8 در برنامه‌نویسی PLC داخلی اینورتر به عنوان شمارنده‌ی سرعت بالا تعریف شود، باید مقداری را در فلگ‌های خاص M1038 یا M1039 نوشته و توابع اصلی ورودی MI8 را غیرفعال کرد.

8- 11- تابع محاسبه‌ی فرکانس

وظیفه‌ی دیگر شمارنده‌ی سرعت بالای مربوط به ورودی دیجیتال MI8 اینورتر VFD-C2000 دلتا تبدیل پالس دریافت شده به فرکانس می‌باشد. همانطور که در شکل زیر مشاهده می‌کنید، تضادی بین تبدیل فرکانس و محاسبات شمارش وجود ندارد. بنابراین، از این دو تابع می‌توان به طور همزمان استفاده کرد.

در فرمول محاسبه‌ی سرعت PLC از رجیسترهای خاص لیست شده در زیر استفاده شده است:

D1057 ← سرعت

D1058 ← محاسبات بین مدت زمان (فاصله)

D1059 ← نقطه‌ی اعشاری

فرض کنید که در هر ثانیه 5 عدد پالس از ورودی MI8 دریافت می‌شود و همچنین، مقدار رجیستر D1058 را برابر با $1000\text{ms}=10\text{sec}$ برای محاسبه‌ی فرکانس در این مدت زمان (فاصله) طی شده تنظیم کرده‌ایم. در این صورت، سیستم برای تبدیل فرکانس در هر ثانیه این 5 پالس را به فرمول ارسال می‌کند.



فرض کنید که فرکانس این 5 پالس در هر ثانیه برابر با 1Hz است و ما مقدار رجیستر خاص D1057 را برابر با 5 تنظیم کرده‌ایم.

اکنون فرض کنید می‌خواهیم دو رقم اعشار را نیز نمایش دهیم. در این صورت، باید مقدار رجیستر خاص D1059 را برابر با 2 تنظیم نماییم. در این صورت، فرکانس به صورت 1.00Hz نمایش داده خواهد شد. این مقدار در رجیستر D1056 بدون اعشار نمایش داده می‌شود. یعنی 1.00Hz در رجیستر D1056 به صورت 100 نشان داده می‌شود که دو رقم یکان و دهگان این مقدار مربوط به اعشار می‌باشد. فرمول تبدیل فرکانس به فرم زیر می‌باشد:

$$D1058 = \frac{\text{Pulse per second}}{D1057} \times \frac{1000}{D1057} \times 10^{D1059}$$

فصل 9

مانیتورینگ و کنترل تجهیزات آنالوگ در HMI دلتا



9-1 مقدمه

اتوماسیون صنعتی، به معنای استفاده از سیستم‌های کنترلی از قبیل کامپیوترها برای کنترل فرآیندها و ماشین آلات صنعتی و کاهش دخالت انسان در آن است. در اتوماسیون صنعتی از کامپیوترهای ویژه‌ای با نام کنترل کننده‌های منطقی برنامه‌پذیر (Programmable Logic Controller-PLC) در سطح وسیعی برای هماهنگ کردن جریان ورودی‌ها از حسگرها (Sensors) با جریان خروجی‌ها به عملگرها (Actuators) و رویدادها (Events) استفاده می‌شود. به این ترتیب، فرمان‌های کنترل شده‌ی دقیق، می‌توانند کنترل هر آنچه را که در فرآیند صنعتی است، به دنبال داشته باشند.

از واسطه‌های بین انسان و ماشین (Human Machine Interface-HMI) معمولاً برای ارتباط PLC‌ها با بقیه کامپیوترها به منظور پردازش داده‌ها، مانیتورینگ و وارد کردن پارامترهای کنترلی برای کنترل و پاسخگویی به شرایط اضطراری استفاده می‌شوند. یک HMI معمولاً به پایگاه داده و برنامه‌های نرم‌افزاری یک سیستم اتوماسیون (Automation System) اتصال می‌یابد تا ترسیم نمودارها، مدیریت آلام و مدیریت اطلاعات را برعهده گیرد.

در سیستم HMI، اطلاعات فرآیند، معمولاً به شکل دیاگرامی برگرفته از فرآیند، به صورت گرافیکی نشان داده می‌شوند. به این معنی که اپراتور، نمایش شماتیکی از فرآیند در حال کنترل را می‌تواند ببیند. به عنوان مثال، تصویری از ترموکوپلی که در داخل کوره قرار گرفته شده است، می‌تواند به اپراتور اعلام کند که دمای اندازه‌گیری شده‌ی کوره توسط ترموکوپل چقدر است.

یک بخش مهم در هر سیستم اتوماسیون، پیاده سازی شرایط آلام در آن است. آلام یک وضعیت دو حالته است که می‌تواند در وضعیت عادی و یا در وضعیت آلام باشد. وضعیت آلام زمانی که شرایط آن برقرار شود، اتفاق می‌افتد. به عنوان مثال، وقتی که دمای داخل کوره از یک حد معینی بالاتر رفته و یا پایین تر رود، آلام می‌تواند اتفاق بیفتد. در این حالت، اپراتور متوجه آلام و مکانی که آلام در آن ایجاد شده است، می‌شود و می‌تواند تمهیدات لازم را در نظر بگیرد.

9-2 برنامه‌نویسی HMI دلتا

HMI‌های دلتا به دو دسته‌ی متنی و گرافیکی تقسیم می‌شوند که HMI‌های متنی در خانواده‌ی DOP/TP و HMI‌های گرافیکی در خانواده‌ی DOP-A و DOP-B قرار گرفته‌اند. نرم‌افزار برنامه‌نویسی HMI‌های متنی TP Editor Software بوده و نرم‌افزار برنامه‌نویسی HMI‌های

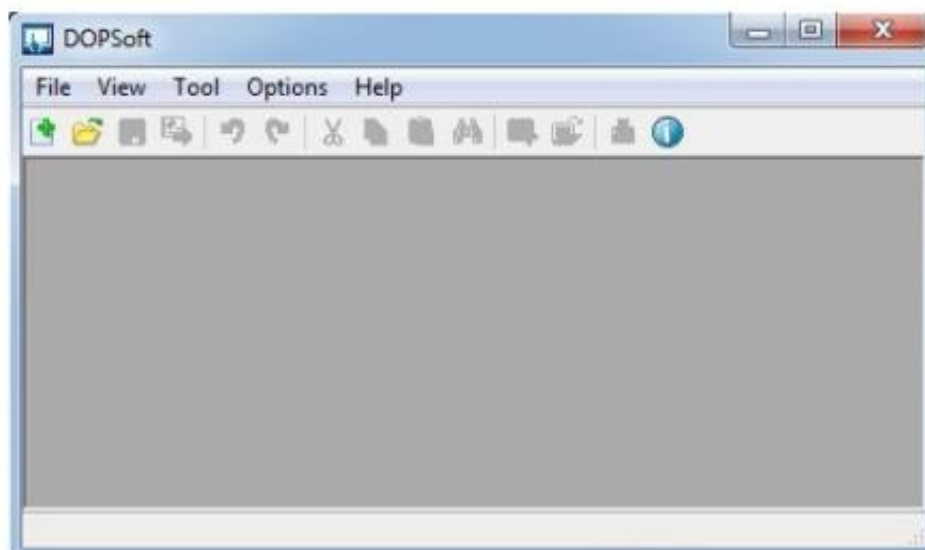
گرافیکی ScreenEditor_DOP-A و ScreenEditor_DOP-B می‌باشد. همچنین، دلتا HMI های سری HMC را نیز ارائه کرده که این سری دارای دو پردازنده‌ی سرعت بالا می‌باشد. اولین پردازنده CPU بوده و کار پردازش HMI را انجام می‌دهد و دومین پردازنده DSP می‌باشد که در حقیقت، یک PLC بزرگ محسوب شده و کار پردازش کنترل حرکت ترتیبی را انجام می‌دهد. همچنین، از HMI های سری HMC نیز می‌توان در اتوماسیون خانگی برای هوشمند کردن ساختمان استفاده کرد. نرم‌افزار برنامه‌نویسی HMI های سری HMC نرم‌افزار DOPSoft بوده که هر دو نرم‌افزار HMI Screen Editor و PLC Programming Software را پوشش می‌دهد.

لازم به ذکر است که برنامه‌نویسی در نرم‌افزارهای ScreenEditor_DOP-B و DOPSoft برای HMI کاملاً شبیه به یکدیگر بوده و می‌توان از هر دو نرم‌افزار برای برنامه‌نویسی HMI های سری DOP-B استفاده کرد. توجه داشته باشید که نرم‌افزار DOPSoft دیتابیس تمام HMI های سری DOP-B را پوشش داده و شامل هر 28 نوع تمام HMI سری DOP-B می‌باشد. ولی نرم‌افزار ScreenEditor_DOP-B تنها 13 نوع از HMI سری DOP-B پوشش می‌دهد.

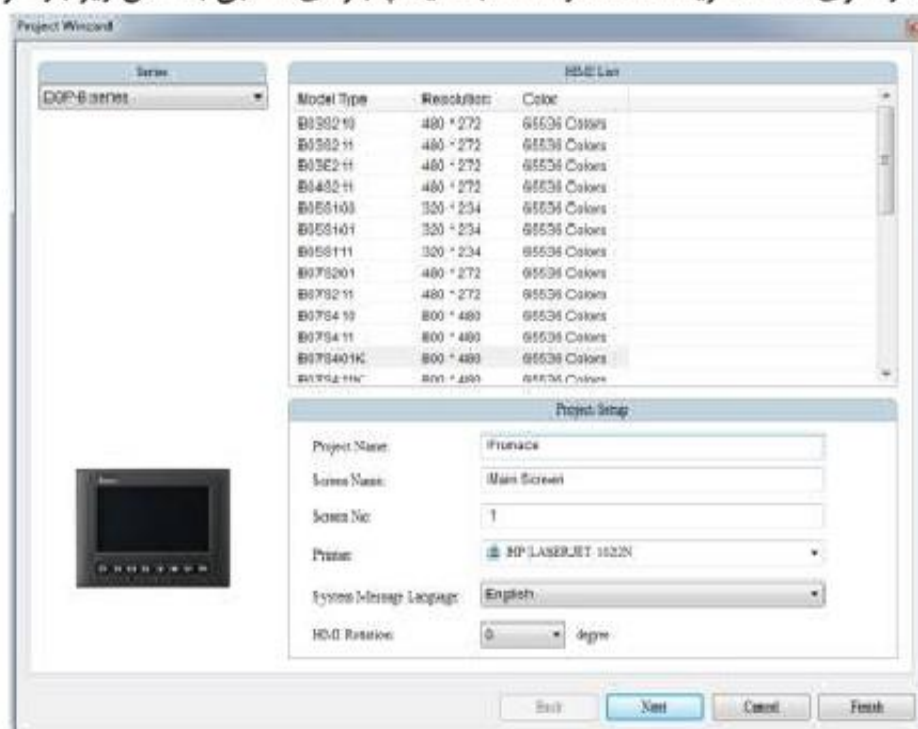
در این کتاب، به دلیل استفاده از HMI سری DOP-B07S401K 65536 Colors برای تست مثال‌های گفته شده، از نرم‌افزار DOPSoft استفاده کرده‌ایم. دلیل استفاده از این نرم‌افزار به جای نرم‌افزار ScreenEditor_DOP-B نداشتن دیتابیس HMI سری DOP-B07S401K 65536 Colors در نرم‌افزار ScreenEditor_DOP-B می‌باشد. همانطور که گفته شد، برنامه نویسی توسط این دو نرم‌افزار مشابه یکدیگر بوده و هیچ جای نگرانی برای کار با این دو نرم‌افزار وجود ندارد. یکی از دلایل مهم دیگر انتخاب این نرم‌افزار آن است که این نرم‌افزار توانایی برقراری ارتباط با سیستم عامل اندروید، مانیتورینگ و کنترل از راه دور از طریق اینترنت، کنترل حرکت ترتیبی و ترکیبی سروسیستم به صورت مستقیم و بدون واسط PLC، توانایی وارد کردن کدهای حرکتی از قبیل G Code و M Code را دارد.

9-3 ایجاد یک پروژه‌ی جدید

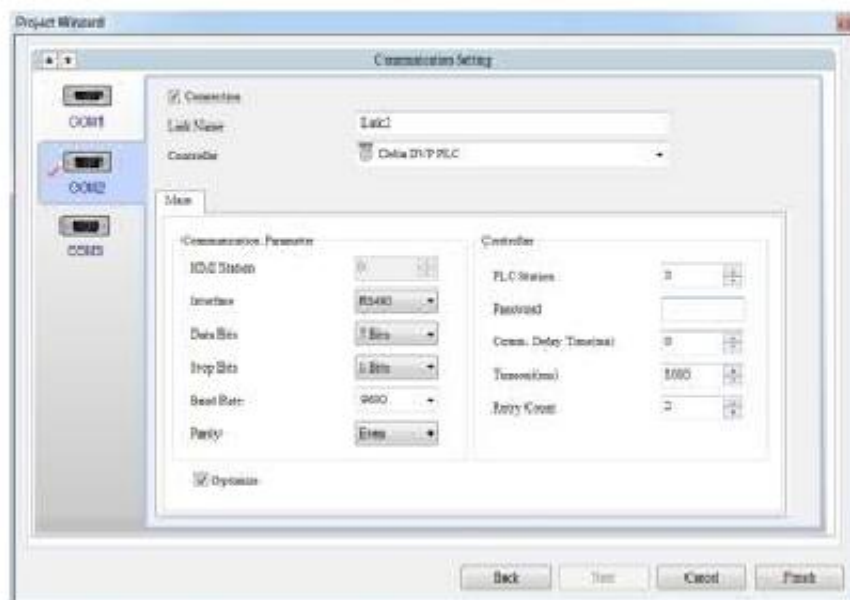
بعد از نصب نرم‌افزار DOPSoft بر روی کامپیوتر، نرم‌افزار را اجرا نموده و مراحل گفته شده در زیر را طی کنید.



مرحله 1: از منوی File گزینه New را انتخاب کنید. پنجره‌ای مطابق با شکل زیر باز خواهد شد.




مرحله 2: در کادر Series سری HMI را برابر با DOP-B series انتخاب نمایید. در پنل HMI List مدل HMI مورد نظر خود را انتخاب کنید. در پنل Project Setup یک نام برای پروژه، یک نام برای صفحه اصلی، شماره صفحه برای صفحه اصلی، نوع پرینتری را که می‌خواهید به HMI متصل نمایید، زبان نرم‌افزار و در نهایت زاویه نمایش تصویر بر روی HMI را تنظیم کنید. سپس بر روی دکمه‌ی Next کلیک نمایید. در این زمان، پنجره‌ی مطابق با پنجره‌ی زیر باز خواهد شد.



مرحله 3: این پنجره مربوط به تنظیمات ارتباطات HMI می‌باشد. پورتی را که می‌خواهید با آن پورت به PLC متصل شوید را انتخاب کنید. در این مثال، ما پورت COM2 را برای اتصال به PLC انتخاب کرده‌ایم. دقت داشته باشید که برای فعال بودن پورت مورد نظر تیک Connection باید خورده باشد. یک نام را برای لینک اختصاص دهید. در کادر Controller نام PLC و پروتکل ارتباطی مورد نظر خود را انتخاب کنید. در این کتاب، ما از PLC سری DVP-SA2 استفاده کرده‌ایم به همین خاطر در کادر Controller از پوشه Delta گزینه Delta DVP PLC را انتخاب می‌کنیم. در پنل Main باید پارامترهای ارتباطات مربوط به HMI و پارامترهای مربوط به PLC را تنظیم کنید. توجه داشته باشید که پارامترهای Interface, Data Bits, Stop Bits, Baud Rate, Parity در هر دو سمت PLC و HMI باید با یکدیگر یکسان باشند، در غیر اینصورت، ارتباط بین این دو برقرار نخواهد شد. ولی پارامتر HMI Station و PLC Station باید با یکدیگر متفاوت بوده و در کل شبکه باید منحصر بفرد باشند. اگر در یک شبکه دو و یا چند آدرس ایستگاه برای دستگاه‌های متصل شده به شبکه باهم یکسان باشند، شبکه خطا داده و ارتباط بین دستگاه‌ها برقرار نخواهد شد. دقت داشته باشید که دقیقاً باید همان آدرس شبکه‌ای را که برای PLC تنظیم کرده‌اید نیز، در کادر PLC Station وارد کنید، در غیر اینصورت اگر آدرس‌ها متفاوت باشند، ارتباط بین این دو برقرار نخواهد شد. در این مثال، ما آدرس گرهی پورت ارتباطات RS-485 مربوط به PLC را در نرم‌افزار WPLSoft برابر با 3 تنظیم کرده‌ایم. اگر برای PLC ی که به HMI متصل کرده‌اید پسورد قرار داده‌اید، باید پسورد PLC را در پنل Controller و در کادر Password وارد کنید. در غیر اینصورت، HMI با PLC ارتباط برقرار نخواهد کرد.

در پایان، برای اینکه تنظیمات انجام شده را بخواهیم ذخیره نماییم باید بر روی دکمه‌ی Finish کلیک کنیم.

9-4 اجرای تنظیمات پورت RS-485 در نرم‌افزار WPL Soft برای PLC

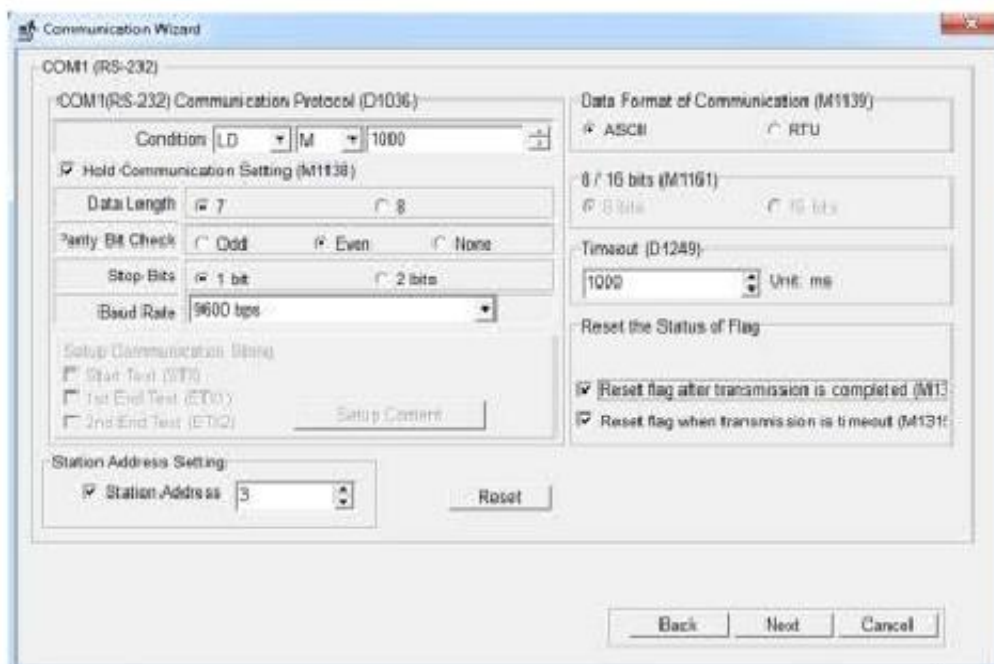
همانطور که در سمت HMI تنظیمات پورت ارتباط بین HMI و PLC را انجام دادیم، باید در سمت PLC نیز تنظیمات پورت ارتباطات برای برقراری ارتباط بین PLC و HMI را نیز انجام دهیم. برای اینکار باید در نرم‌افزار WPL Soft آیکن  را انتخاب کنید.



اکنون، پنجره‌ای مطابق با پنجره‌ی زیر باز خواهد شد.



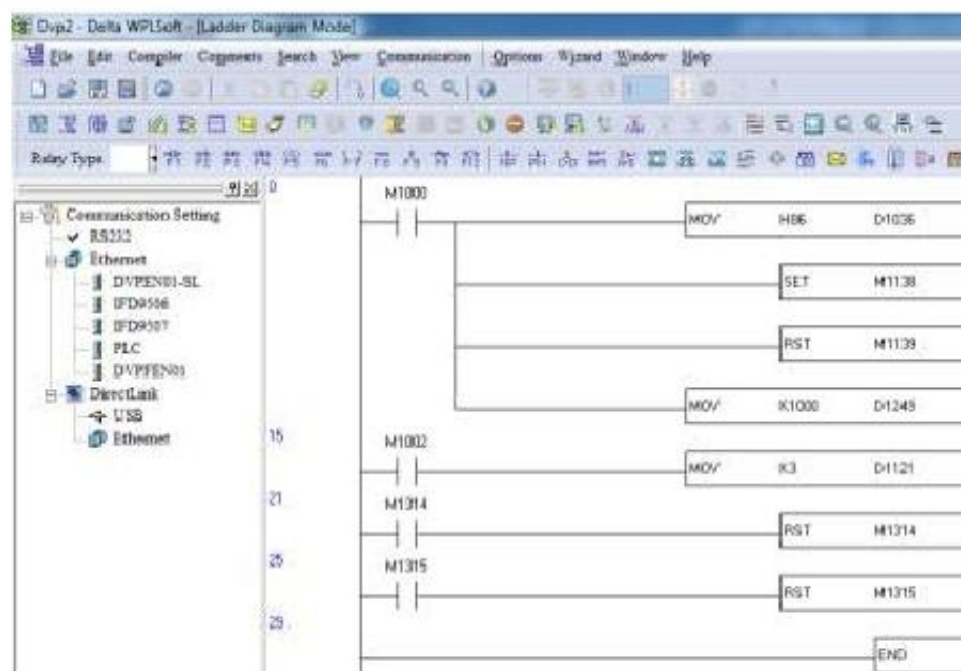
در این پنجره باید گزینه‌ی COM2 (RS-485) را انتخاب کنید. سپس بر روی دکمه‌ی Next کلیک کنید تا به پنجره‌ی بعدی بروید.



همانطور قبلاً اشاره شد، باید تنظیمات پارامترهای Interface, Data Bits, Stop Bits, Baud Rate, Parity و HMI در هر دو سمت PLC و HMI یکسان باشند. همانطور که مشاهده می‌کنید، در کادر PLC Station آدرس گرهی ارتباطات پورت RS-485 را برابر با 3 تنظیم کرده‌ایم. از این رو، HMI در شبکه توانایی برقراری ارتباط با این PLC را که آدرس آن 3 است را دارد. سپس بر روی دکمه‌ی Next کلیک کنید تا به پنجره‌ی بعدی بروید.



در این پنجره فقط بر روی دکمه‌ی Next کلیک کنید تا برنامه‌ی تنظیمات انجام شده توسط کامپایلر نرم‌افزار تولید شود.



در مرحله‌ی بعد، باید برنامه‌ی اصلی کنترل و مانیتورینگ دما را به برنامه‌ی بالا اضافه کنیم. ادامه‌ی برنامه‌ی PLC را در بخش‌های بعد خواهیم نوشت.

9- 5 سیم‌بندی پورت ارتباطات PLC و HMI

در شکل زیر، روش سیم‌بندی پورت ارتباطات RS-485 بین PLC سری DVP ی دلتا و HMI سری DOP-B دلتا نشان داده شده است.



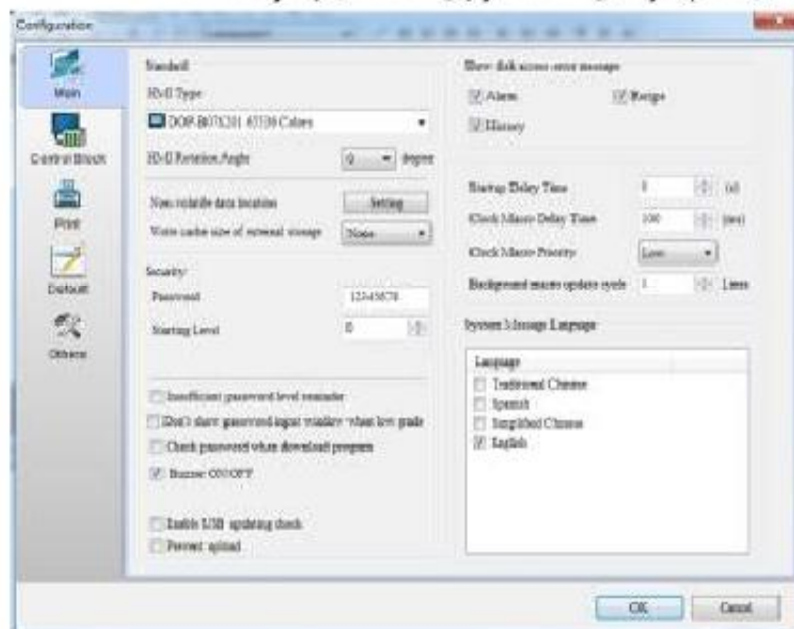
9- 6 اصلاح تنظیمات HMI

بعد از اینکه تنظیمات HMI را تکمیل کردید و زمانی که متوجه شدید در جای اشتباه کرده‌اید می‌توانید تنظیمات انجام شده را اصلاح کنید.

برای اینکار، باید از نوار ابزار Options گزینه Configuration ... را برای پیکربندی HMI انتخاب نماییم.



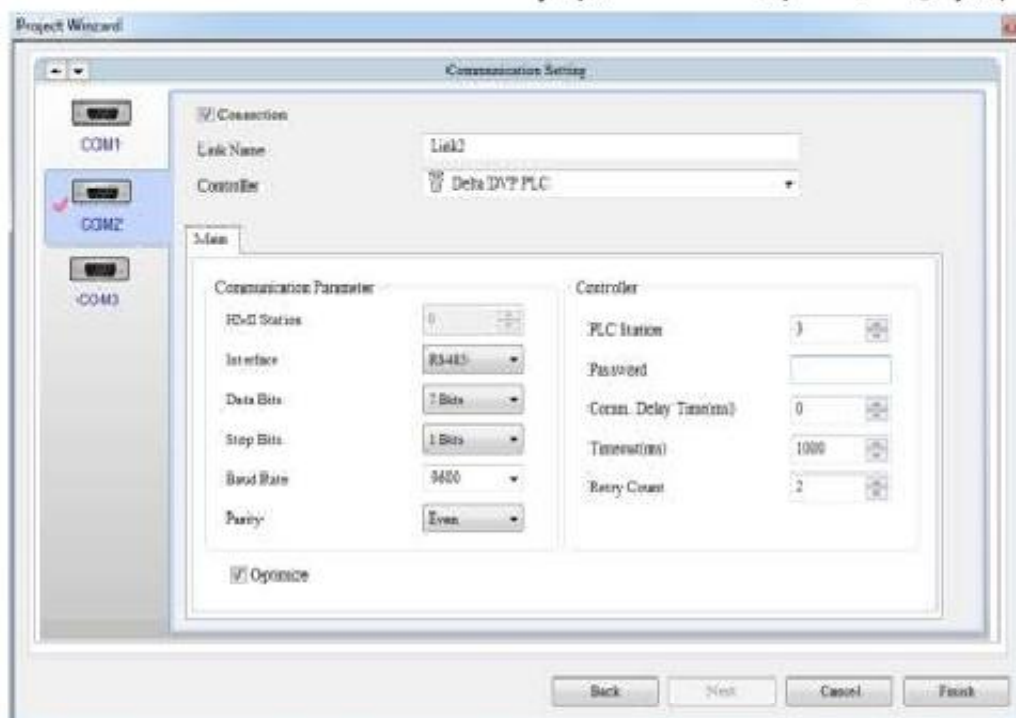
اکنون پنجره‌ی تنظیمات پیکربندی سخت‌افزاری HMI باز خواهد شد.



همچنین، شما می‌توانید از نوار ابزار Options گزینه Communication Setting را برای تنظیمات واسط ارتباطات HMI انتخاب نمایید.



اکنون، پنجره‌ی تنظیمات ارتباطات HMI باز خواهد شد.



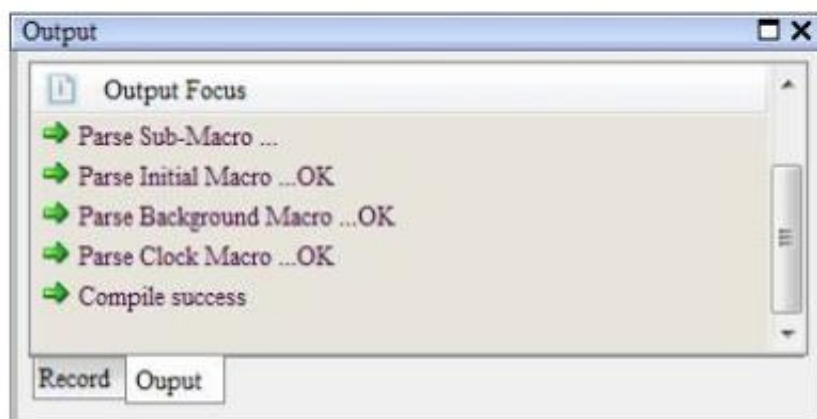
برای اطلاع از جزئیات تنظیمات این دو صفحه می‌توانید به کتاب "راهنمای استفاده سریع از HMI دلتا" که توسط همین انتشارات "انتشارات قدیس" به چاپ رسیده است، استفاده نمایید.

7-9 دانلود/آپلود برنامه به HMI

بعد از اینکه برنامه‌ی HMI را نوشته و تکمیل کردید، نوبت به کامپایل کردن برنامه می‌رسد. برای کامپایل کردن برنامه باید از نوار ابزار Tools گزینه‌ی Compile را انتخاب کنید.



در صورتیکه برنامه درست و بدون خطا باشد، در پنجره‌ی Output باز شده باید Compile success را مشاهده کنید.

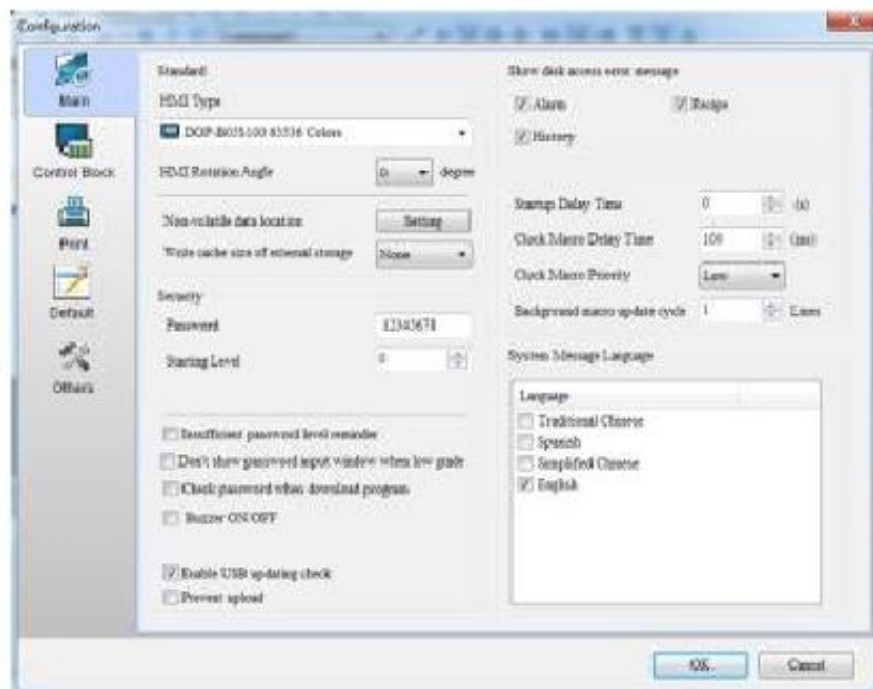


اکنون برای دانلود برنامه‌ی نوشته شده به HMI، بر روی نوار ابزار **Tools** کلیک کنید و یکی از سه گزینه‌ی **Download All Data**, **Download Screen**, **Download Recipe** را با توجه به نیاز خود انتخاب کنید.

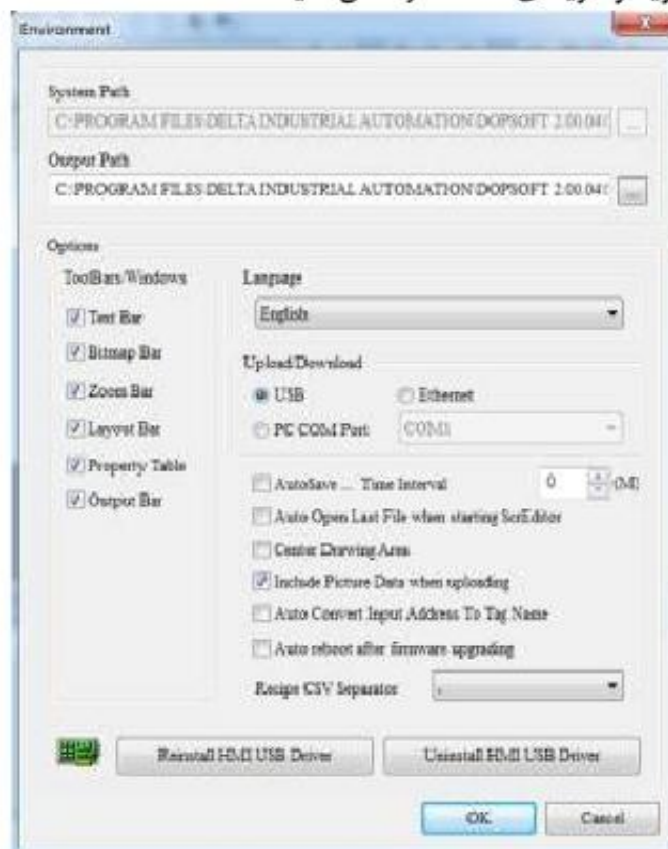


در صورتی که کابل ارتباط کامپیوتر به HMI به درستی متصل شده و تنظیمات انجام شده درست باشد، برنامه‌ی نوشته شده برای HMI در نرم‌افزار DOPSoft به HMI دانلود خواهد شد. اگر ارتباط بین کامپیوتر و HMI برقرار نشد، یک پیغام خطا نمایش داده شده و عدم ارتباط را به کاربر هشدار خواهد داد. دانلود کردن و یا آپلود کردن برنامه که از طریق **USB** یا پورت **RS-232** و یا اترنت انجام می‌شود، دارای تنظیمات ویژه‌ای می‌باشد. تنظیمات مربوط به دانلود/آپلود از طریق زیر انجام می‌شود.

از منوی **Options** گزینه‌ی **Configuration** را انتخاب کرده و از پنجره‌ی باز شده تیک گزینه‌ی **Enable USB updating check** را بزنید.



از منوی Options گزینه‌ی Environment را انتخاب کرده و از پنجره‌ی باز شده از نصب بودن درایور USB مطمئن شوید و گزینه‌ی USB را فعال کنید.



برای آپلود کردن برنامه از HMI به کامپیوتر در نرم افزار DOPSoft با توجه به نیاز خود باید یکی از دو گزینه‌ی Upload All Data و یا Upload Recipe را از نوار ابزار Tools انتخاب کنید.



9-8 برنامه‌ی مانیتورینگ و کنترل I/O آنالوگ PLC در HMI

در این مثال، می‌خواهیم کنترل و مانیتورینگ مقادیر پیوسته‌ی آنالوگ یک کوره‌ی ذوب را بررسی کنیم. در این مثال هم برنامه‌ی PLC و هم برنامه‌ی HMI قدم به قدم توضیح داده خواهد شد.

9-8-1 ساخت افزار مورد نیاز

سخت افزارهای مورد نیاز در این پروژه عبارتند از :

یک DVP-PLC سری DVP-12SA2

یک ماژول ورودی آنالوگ سری DVP-04AD_S

یک ماژول خروجی آنالوگ سری DVP-04DA_S

یک ماژول ترموکوپل سری DVP-04TC_S

یک HMI سری DOP-B07S401K 65536

یک ترانسدیوسر فشار 0-600 mbar با سیگنال کنترلی 0-10 V

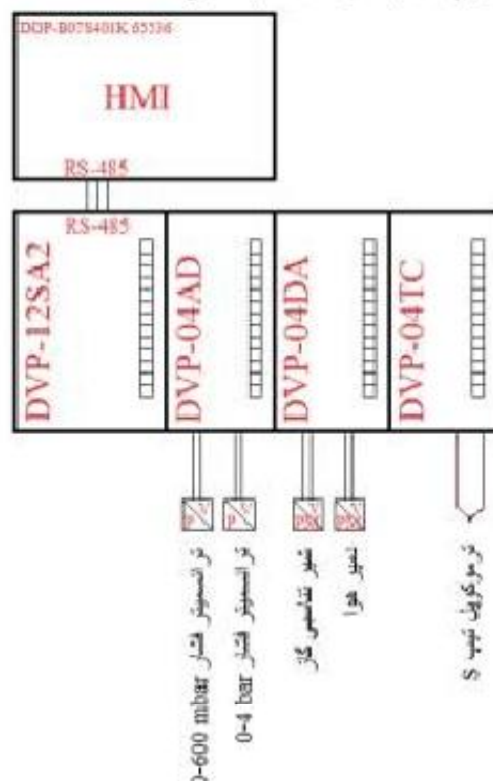
یک ترانسدیوسر فشار 0-4 bar با سیگنال کنترلی 0-10 V

یک شیر تدریجی گاز با لوله‌ی 3 اینچ، فشار 0-500 mbar، سیگنال کنترلی 0-10 V

یک دمپر هوا با سیگنال کنترلی 0-10 V

یک ترموکوپل تیپ S

در شکل زیر، نحوه‌ی پیکربندی و سیم‌بندی سیستم نشان داده شده است:



ترانسدیوسر فشار 0-600 mbar به ترمینال‌های V+، COM، کانال CH1 ماژول DVP-04AD_S و ترانسدیوسر فشار 0-4 bar به ترمینال‌های V+، COM، کانال CH2 ماژول DVP-04AD_S متصل شده است.

شیر تدبیری گاز به ترمینال‌های V+، COM، کانال CH1 ماژول DVP-04DA_S و دمپر هوا به ترمینال‌های V+، COM، کانال CH2 ماژول DVP-04DA_S متصل شده است. ترموکوپل تیپ S را به ترمینال‌های L-، L+، کانال CH1 ماژول DVP-04TC_S متصل کرده‌ایم.


9-8-2 آدرس‌دهی و برنامه‌نویسی در سمت PLC

خود PLC به صورت اتوماتیک به هر یک از کانال‌های هر ماژول توسعه‌ی خاص آنالوگ یک آدرس را اختصاص می‌دهد که این آدرس‌دهی مطابق با شکل زیر می‌باشد:



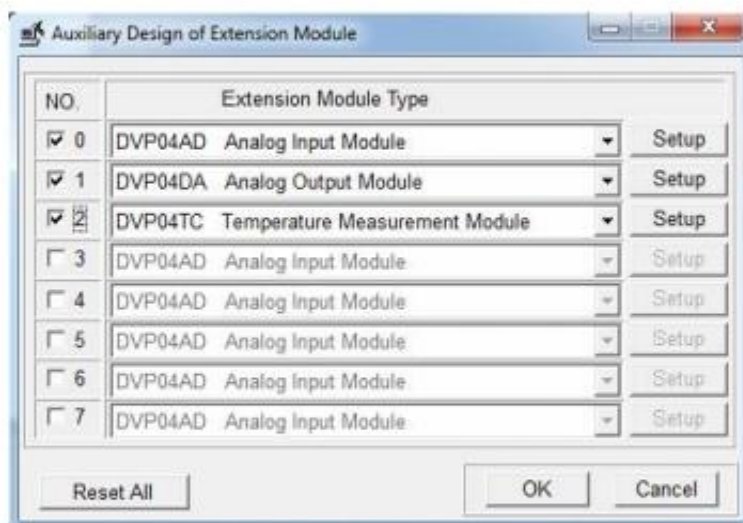
DVP-12SA2	DVP-04AD	DVP-04DA	DVP-04TC
	↓	↓	↓
	First module	Second module	Third module
CH1 AIO conversion value	D9900	D9910	D9920
CH2 AIO conversion value	D9901	D9911	D9921
CH3 AIO conversion value	D9902	D9912	D9922
CH4 AIO conversion value	D9903	D9913	D9923
CH5 AIO conversion value	X	X	X
CH6 AIO conversion value	X	X	X

9-8-3 برنامه‌نویسی برای PLC

همانطور که در فصل‌های قبل ذکر شد، شما می‌توانید این آدرس‌دهی را تغییر داده و از رجیسترهای دیگری برای نمایش و کنترل مقادیر ورودی خروجی آنالوگ استفاده نمایید. برای این کار باید در نرم‌افزار WPLSoft آیکن  را انتخاب کنید.

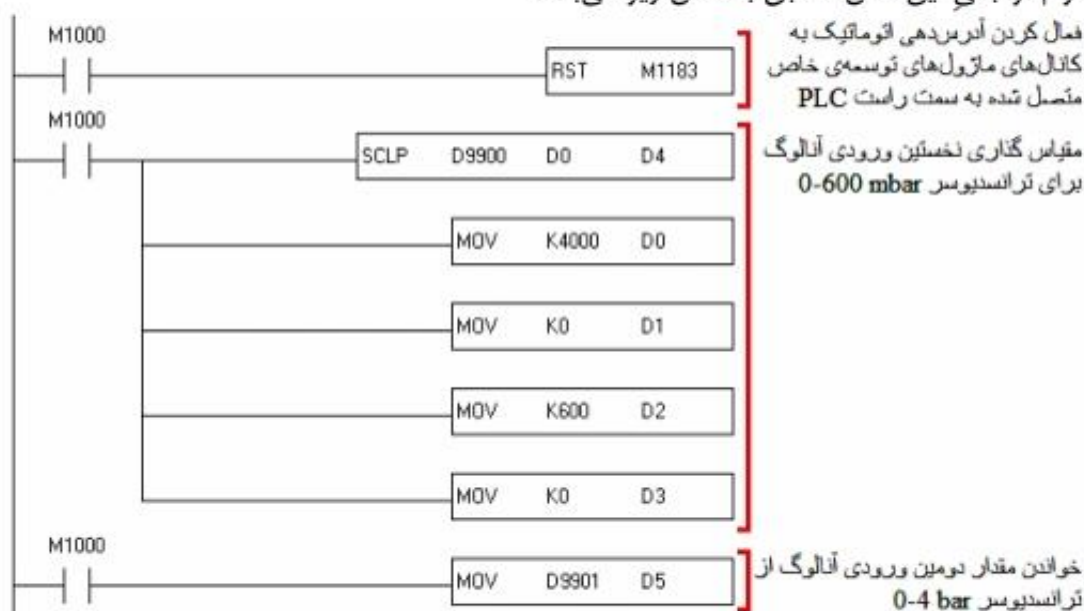


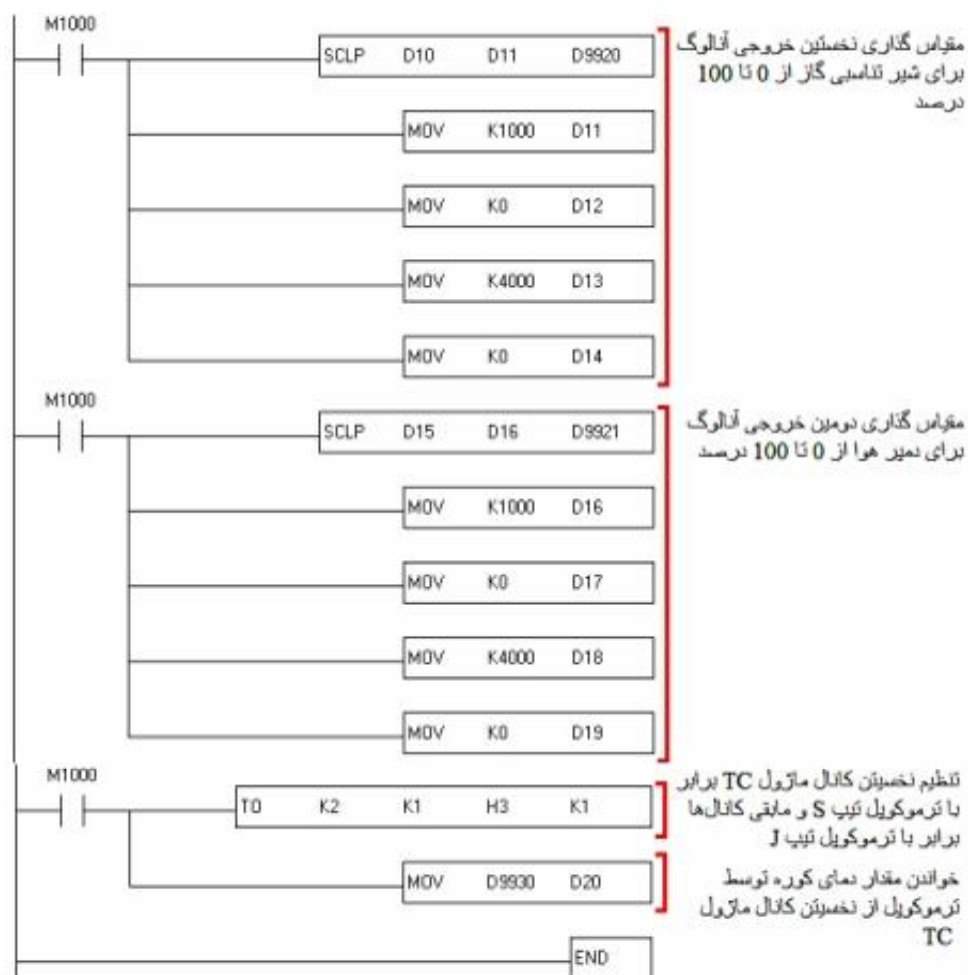
پنجره‌ای مطابق با شکل زیر باز خواهد شد که در این پنجره می‌توانید ماژول توسعه‌ی خاص مدنظرتان را انتخاب کرده و تنظیمات گرافیکی آن را انجام دهید.



در اینجا، ما به دلیل حجم تر نشدن تعداد صفحات کتاب از توضیح تنظیمات گرافیکی صرف نظر می‌کنیم و فقط برنامه‌ی ایجاد شده در نرم‌افزار WPLSoft را بعد از تنظیمات گرافیکی پارامترهای ماژول‌های توسعه‌ی آنالوگ نشان خواهیم داد. توجه داشته باشید که ما در این مثال آدرس‌دهی پارامترهای ماژول آنالوگ به صورت ترکیبی استفاده خواهیم کرد. یعنی از آدرس‌های اختصاص داده شده به صورت اتوماتیک توسط PLC به کانال‌های ماژول‌ها برای خواندن مقادیر از ورودی‌های آنالوگ و نوشتن مقادیر در خروجی‌های آنالوگ استفاده کرده و برای انجام تنظیمات دیگر پارامترهای ماژول‌های آنالوگ از آدرس‌دهی توسط کاربر به صورت برنامه‌نویسی استفاده کرده‌ایم. همانطور که متوجه شدید، آدرس‌دهی قابل انعطاف بوده و از هر نوع آدرس‌دهی می‌توانید استفاده کنید.

دیagram نردبانی این مثال مطابق با شکل زیر می‌باشد.





از توضیح عملکرد این برنامه‌ی کنترلی به دلیل اینکه هدف فقط آموزش برنامه‌نویسی HMI برای کار با مقادیر آنالوگ می‌باشد، صرف‌نظر شده است.

لازم به ذکر است که در این مثال از سیستم عددی، اعداد اعشاری باینری **binary floating point** برای محاسبات و ذخیره‌سازی داده‌ی بدست آمده از عملیات ریاضی استفاده شده نشده است. توجه داشته باشید که در صورت استفاده از این سیستم عددی در محاسبات ریاضی، باید برای محاسبات از دستورالعمل‌های ریاضی اعداد اعشاری باینری (**binary floating point operation instruction**) استفاده کنید.

همچنین، ممکن است از اعداد اعشاری و اعداد صحیح به صورت ترکیبی در کنار یکدیگر استفاده شود. بنابراین، اگر مقدار پارامتری قبل از محاسبات به صورت عدد اعشاری باینری نبود، ابتدا باید مقدار آن عدد را با استفاده از دستورالعمل **FLT** به عدد اعشاری باینری تبدیل کنید تا عمل محاسبات و مانیتورینگ به سادگی صورت گیرد. توجه داشته باشید که مقدار بدست آمده در محاسبات اعداد اعشاری

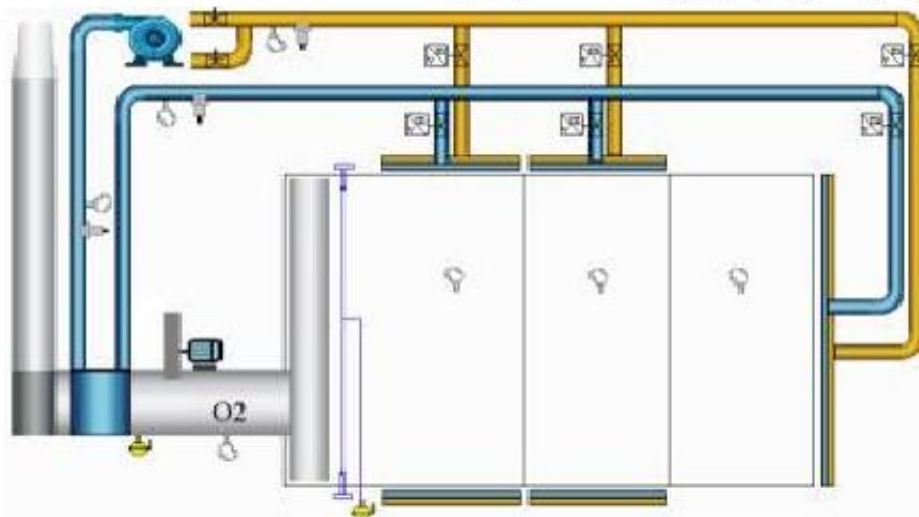
باینری قابل فهم نمی‌باشد. از اینرو، مقدار اعداد اعشاری باینری بدست آمده از محاسبات را باید به اعداد اعشاری دسیمال تبدیل کرد.

توجه داشته باشید که تمام پارامترها باید به فرمت اعداد صحیح باینری **binary floating integers** باشند (به عنوان مثال، ... **DDIV, DFLT, DEDIV, DEMUL**). در غیر اینصورت، از دستورالعمل **FLT** برای تبدیل سیستم اعداد استفاده نمایید.

برای مثال، فرض کنید که در یک برنامه مقادیر باینری را به مقادیر دسیمال تبدیل کرده‌ایم. روش محاسبه به این صورت است که ابتدا مقادیر باینری موجود در رجیسترهای (به عنوان مثال **D105, D104**) به مقدار دسیمال تبدیل شده و در رجیسترهای (به عنوان مثال **D107, D106**) قرار می‌گیرد. فرض کنید که مقادیر به صورت **D106 = K6250, D107 = K-4** است. بنابراین مقدار عدد اعشاری دسیمال آن برابر با $0.625 = 6250 \times 10^{-4}$ خواهد بود.

9-8-4 برنامه‌نویسی برای HMI

اکنون نوبت کار با نرم‌افزار **DOPSoft** رسیده است. این نرم‌افزار با شمایی که در نرم‌افزارهای طراحی گرافیکی از قبیل **Paint**، فتوشاپ و ... که فرمت **Bitmap image (.bmp)** را تولید می‌کنند، سازگار است. در اینجا، ما در نرم‌افزار **Paint** یک طرح کلی از محیط تحت کنترل را کشیده و در هنگام ذخیره کردن طرح کشیده شده، فرمت آن را برابر با **.bmp** تنظیم کرده‌ایم. لازم به ذکر است که در هنگام کشیدن طرح‌تان باید ابتدا ابعاد صفحه نمایش **HMI** را شناسای کرده و سپس ابعاد صفحه‌ی طراحی‌تان را مطابق با آن تنظیم نمایید. در غیر اینصورت، ممکن است طرح‌تان از صفحه نمایش **HMI** بیرون بزند. به دلیل اینکه مثال ما در مورد مانیتورینگ و کنترل دمای یک کوره است، طرح خود را بر آن اساس در نرم‌افزار **Paint** کشیده‌ایم.

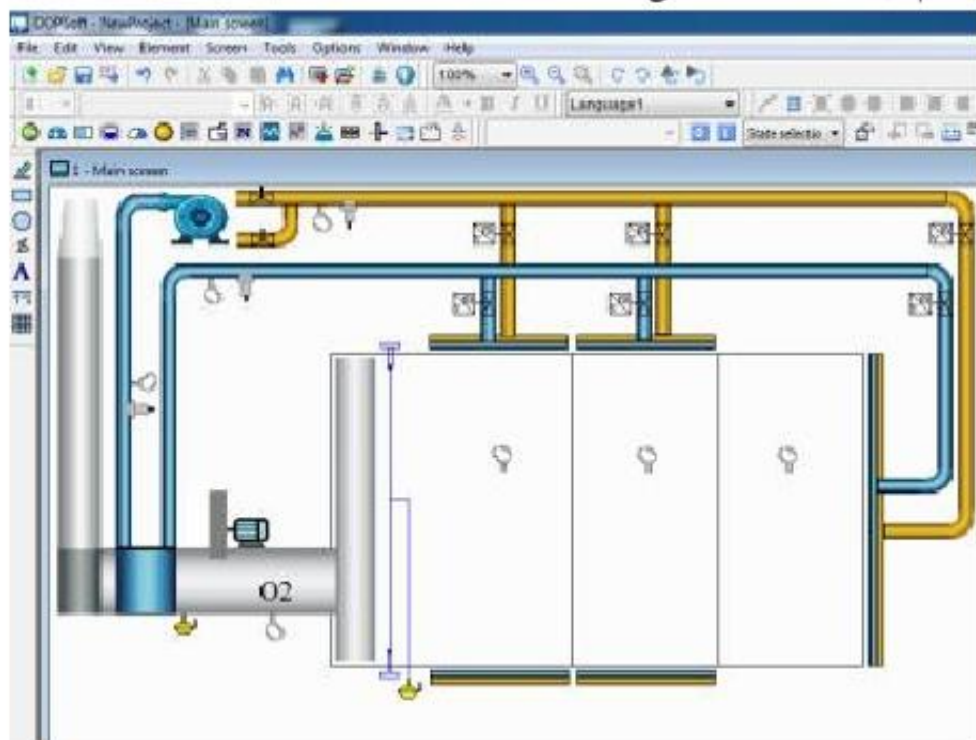


لازم به ذکر است که ما در شماتیک فوق لوله‌ی گاز را مطابق با استاندارد اروپا برابر با رنگ زرد در نظر گرفته‌ایم و همچنین، لوله‌ی هوا را با رنگ آبی کشیده‌ایم.

برای وارد کردن این طرح به صفحه‌ی ایجاد شده در نرم‌افزار DOPSoft، باید از منوی Screen گزینه Import را انتخاب کنید.



در شکل زیر، شمایل طراحی شده که به صفحه‌ی Main screen ایجاد شده در نرم‌افزار DOPSoft چسبانده شده است، نشان داده شده است.



اکنون نوبت به قرار دادن المان‌های نمایشی و کنترلی به صفحه‌ی طراحی شده رسیده است. ابتدا می‌خواهیم یک نمایشگر مقدار جاری برای نمایش مقدار فشار گاز و فشار هوا را بر روی صفحه نمایش

قرار دهیم. برای این کار از منوی View گزینه‌ی Element Tool را تیک می‌زنیم تا پنجره‌ی Element Tool بر روی صفحه‌ی اصلی به نمایش درآید. سپس در زیر مجموعه‌ی Data گزینه‌ی Numeric Display را کلیک کرده و با روش کشیدن و رها کردن آن را بر روی صفحه قرار می‌دهیم. این کار را دو بار برای گاز و هوا انجام می‌دهیم.



اکنون بر روی این دو آیتم به صورت جداگانه دابل کلیک کنید تا صفحه‌ی تنظیمات آن باز شود.



در صفحه‌ی باز شده بر روی دکمه‌ی Read Address کلیک کنید تا صفحه‌ی تنظیم آدرس رجیسترهای PLC باز شود. به نمایشگر فشار گاز آدرس D4 را اختصاص داده و به نمایشگر فشار هوا آدرس D5 را اختصاص می‌دهیم.

اکنون باید نوع داده و مقدار اعشار را برای این دو رجیستر تعیین کنیم. تغییراتی برای نمایشگر فشار گاز مصرفی نیاز نمی‌باشد. ولی برای فشار هوای مصرفی باید تنظیمات گفته شده در زیر را انجام دهیم. صفحه‌ی تنظیمات نمایشگر فشار هوای مصرفی را باز کنید. اکنون باید پارامترهای پنل **Detail** را تنظیم کنیم. پارامتر **Data Type** را باید برابر با یک رجیستر یا همان **Word** تنظیم کنیم. پارامتر **Memory** را باید به دلیل علامت‌دار نبودن مقدار خوانده شده برابر با **Unsigned Decimal** تنظیم کنیم. پارامتر **Integer Digits** را به دلیل داشتن یک رقم صحیح فشار هوا (رقم‌های 0 و 1 و 2 و 3 و 4) باید برابر با 1 تنظیم کنیم. پارامتر **Fractional** را به دلیل داشتن 3 رقم اعشار (رقم‌های 0.000 الی 4.000) باید برابر با 3 تنظیم کنیم.



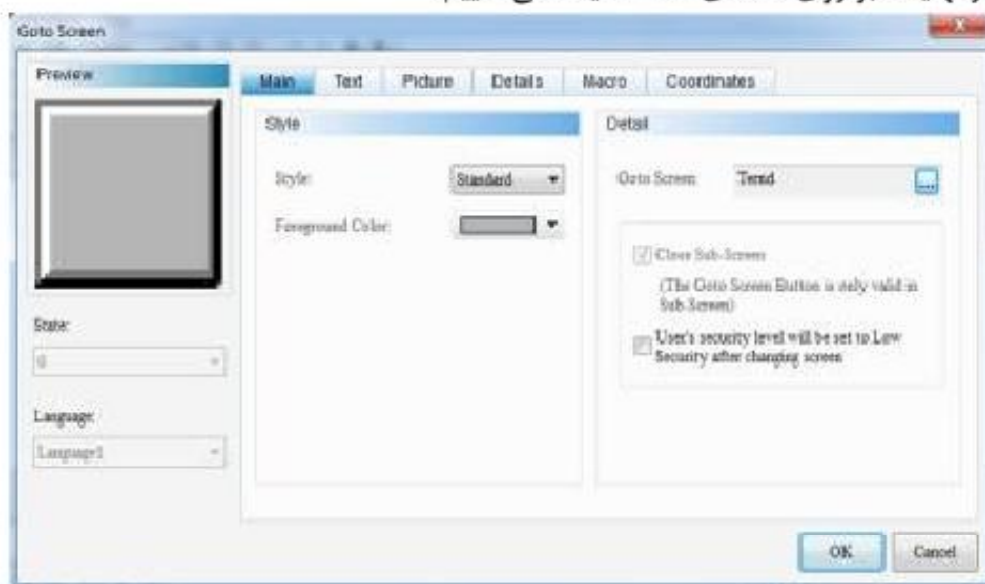
همانطور که متوجه شدید، ما در برنامه‌نویسی PLC از سیستم اعداد اعشاری استفاده نکردیم و در برنامه‌نویسی HMI توانسیم مقدار عدد صحیح داخل PLC را به مقدار عدد اعشاری تبدیل کنیم. از اینرو، برنامه‌نویسی به این روش بسیار ساده‌تر شده و قابل درک می‌باشد.

در این مرحله، می‌خواهیم دمای اندازه‌گیری شده توسط ترموکوپل را به صورت بارگراف و بر روی نمودار نمایش دهیم. برای این کار لازم است که ابتدا از منوی Screen گزینه‌ی New screen را برای ایجاد یک صفحه‌ی جدید انتخاب کنیم. نام صفحه را برابر با Trend تنظیم کرده و نوع صفحه را برابر با Sub screen تنظیم می‌کنیم. سپس برای ایجاد شدن یک زیر صفحه‌ی جدید باید بر روی دکمه‌ی OK کلیک شود.

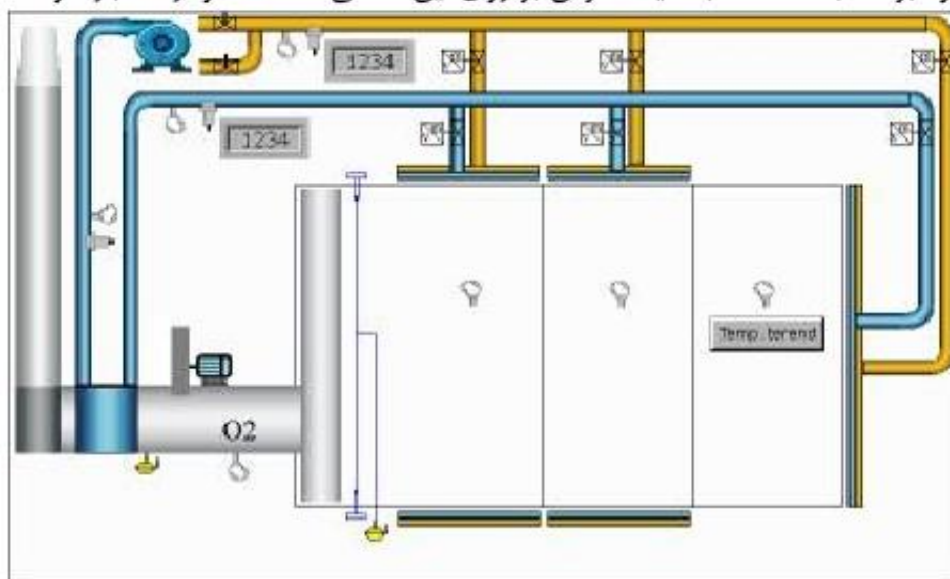


اکنون زیر صفحه‌ای را با نام Trend ایجاد کرده‌ایم. برای باز کردن این زیر صفحه باید از شاخه‌ی Button در پنجره‌ی Element Tool شستی Goto screen را با روش گرفتن، کشیدن و رها

کردن بر روی صفحه‌ی اصلی قرار دهیم. سپس بر روی شستی قرار داده شده بر روی صفحه‌ی اصلی دابل کلیک می‌کنیم تا صفحه‌ی تنظیمات آن باز شود. در این صفحه گزینه‌ی **Goto screen** را برابر با صفحه‌ی **Trend** تنظیم می‌کنیم. نام این شستی را از تب **Text** برابر با **Temp. trend** تنظیم می‌کنیم. در نهایت، بر روی دکمه‌ی **OK** کلیک می‌نماییم.

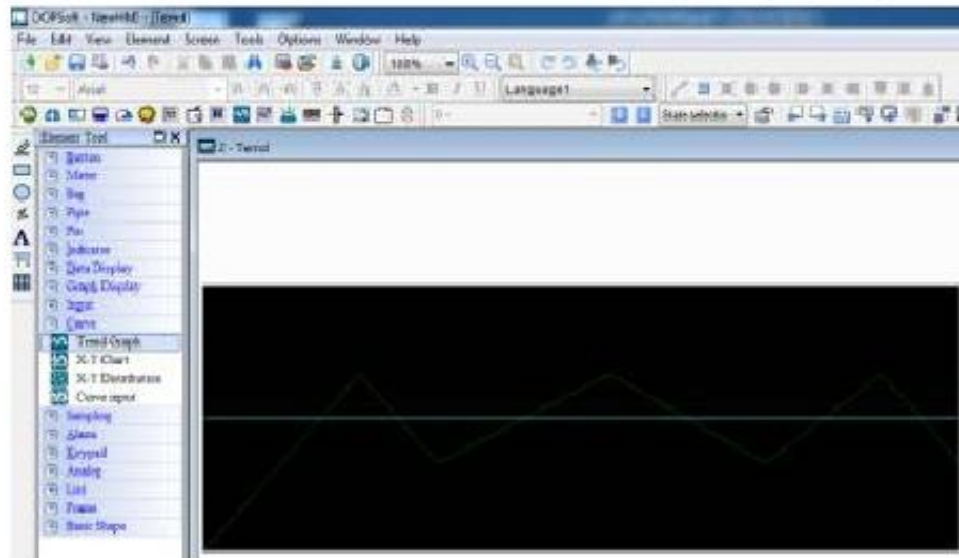


بعد از دانلود برنامه به HMI، با کلیک کردن بر روی این شستی صفحه نمودار دما باز خواهد شد.



اکنون صفحه‌ی **Trend** ایجاد شده را از منوی **Screen** و گزینه **Open screen** انتخاب کرده و آن را باز کنید. از پنجره‌ی **Element Tool** گزینه‌ی **Curve** را باز کرده و **Trend Graph**

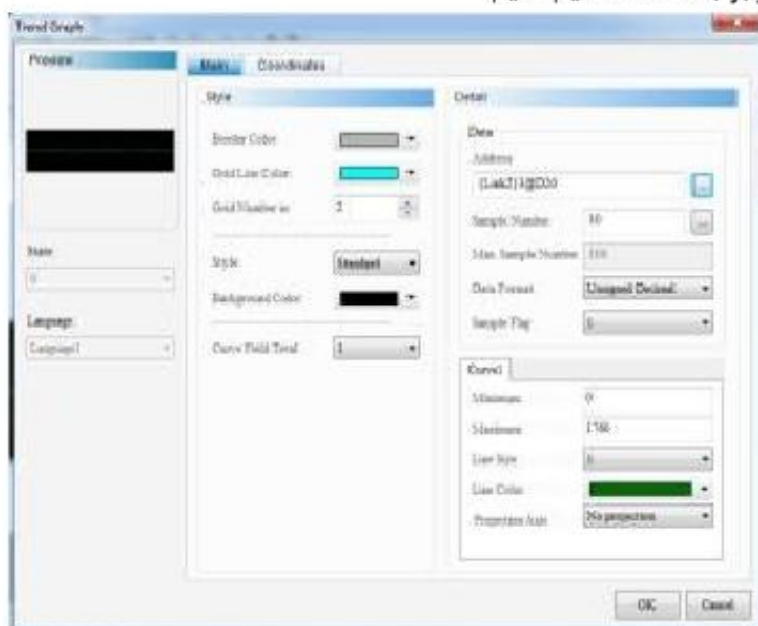
را با روش کشیدن و رها کردن به صفحه‌ی Trend ایجاد شده اضافه کنید. سپس ابعاد المان Trend Graph را مطابق با نیاز خود تنظیم کنید.



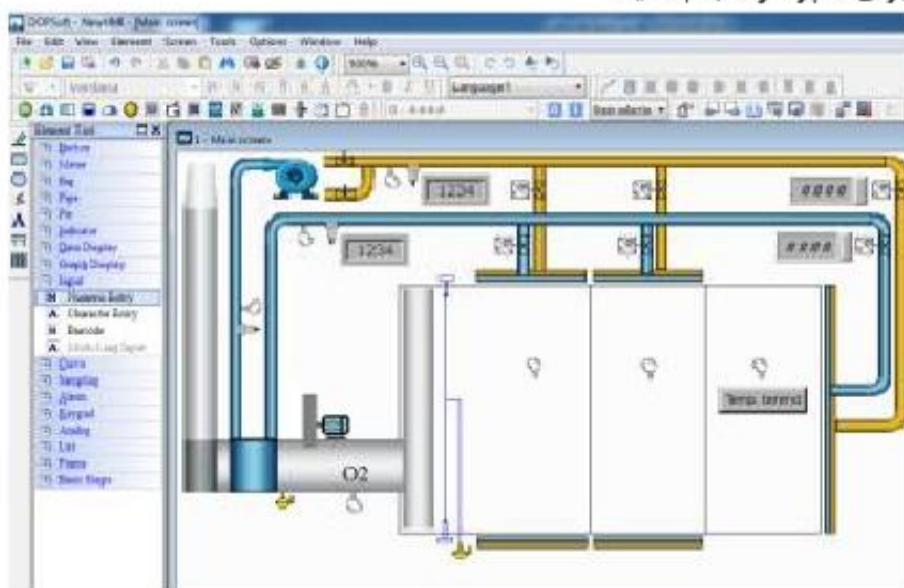
همچنین شما می‌توانید هر المان دیگری را که می‌خواهید، مطابق با نیاز خود به این پنجره اضافه کنید. فرض کنید می‌خواهیم در این پنجره المان Numeric Display را برای نمایش عددی دما نیز قرار دهیم. دقت اندازه‌گیری دما توسط مازول DVP-04TC برابر با 0.1°C می‌باشد و همچنین، دامنه‌ی اندازه‌گیری دما توسط ترموکوپل تیپ S از 0 تا 1760 درجه‌ی سانتی‌گراد است. به همین دلیل، اگر از المان Numeric Display برای نمایش دما استفاده می‌کنید، علاوه بر تنظیم آدرس باید در پنل Detail پارامتر Integer Digits را به دلیل داشتن چهار رقم صحیح دمای قابل اندازه‌گیری (رقم‌های 0 الی 1760) برابر با 4 تنظیم کرده و پارامتر Fractional را به دلیل داشتن 1 رقم اعشار (رقم‌های 0.0 الی 1760.9) برابر با 1 تنظیم کنیم.

اکنون برای تنظیم آدرس برای المان Trend Graph باید بر روی آن دابل کلیک کنید تا پنجره‌ی تنظیمات آن باز شود. بر روی دکمه‌ی Address کلیک کنید تا پنجره‌ی تنظیم آدرس این المان باز شود. برای خواندن مقدار دما و مانیتور کردن آن بر روی Trend Graph باید آدرس D20 را به این المان اختصاص دهید تا بتوانید دمای خوانده شده توسط PLC را به صورت گرافیکی بر روی HMI نمایش دهید. در نهایت، در پنل Curve باید مقدار Minimum را برابر با 0 تنظیم کرده و مقدار Maximum را برابر با 1760 تنظیم کنیم. توجه داشته باشید که تنظیم مقدار حداقل و حداکثر قابل نمایش بر روی محور Y اختیار بوده و شما می‌توانید محدوده‌ی دمای مورد نظرتان را برای مقیاس‌گذاری تنظیم کنید. به عنوان مثال، فرض می‌کنیم که دامنه‌ی اندازه‌گیری دمای محیط تحت کنترل ما بین

1200 الی 1300 می‌باشد. از این رو، ما می‌توانیم مقدار Minimum را برابر با 1100 و مقدار Maximum را برابر با 1400 تنظیم کنیم.



در آخرین مرحله، نوبت به باز و بسته کردن شیر تدریجی گاز و دمپر هوا به صورت 0 تا 100 درصد رسیده است. برای این کار باید وارد صفحه‌ی Main screen ایجاد شده شویم. سپس از پنجره‌ی Element Tool گزینه‌ی Input را باز کرده و المان Numeric Entry را با روش کشیدن و رها کردن به صفحه‌ی Main screen ایجاد شده اضافه کنید. این کار را یک بار برای شیر تدریجی گاز و یک بار دیگر برای دمپر هوا انجام دهید.



سپس بر روی آیتم‌های اضافه شده به صفحه دابل کلیک کنید تا پنجره‌ی تنظیمات آن باز شود.

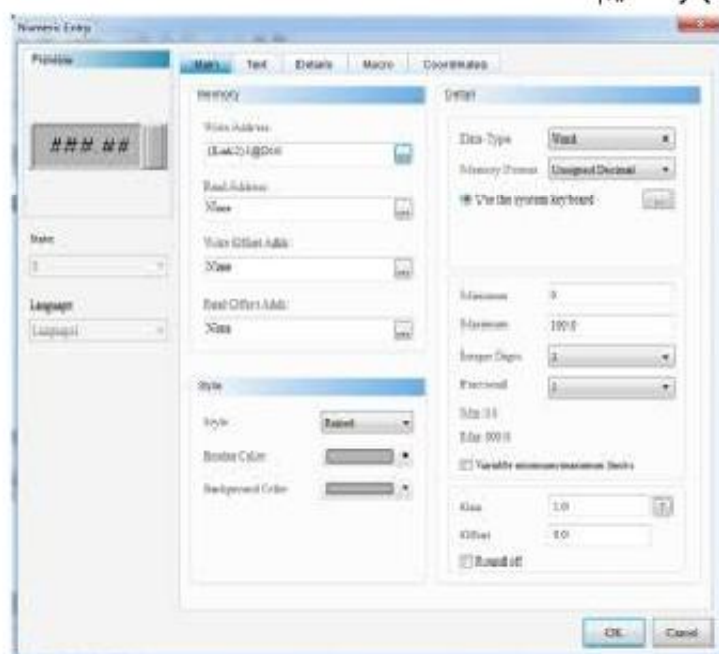


در پنجره‌ی تنظیمات المان Numeric Entry سه پارامتر مهم برای تنظیم وجود دارد که اولین پارامتر Write Address می‌باشد. با کلیک بر روی دکمه‌ی آن، پنجره‌ی تنظیم آدرس باز خواهد شد که در این پنجره می‌توان آدرسی از رجیستر PLC را که قرار است در آن مقداری را توسط HMI بنویسیم، وارد می‌کنیم.



در این مثال، برای تنظیم درصد باز بودن شیر برقی تناسبی (تدریجی) گاز به صورت دستی توسط اپراتور باید آدرس D10 را وارد کرده و برای باز و بسته کردن مقدار درصد دمپر هوا توسط اپراتور به صورت دستی باید آدرس D15 را وارد نماییم.

دومین و سومین پارامتر مهم در تنظیمات المان Numeric Entry تنظیم مقدار Minimum و مقدار Maximum قابل تنظیم این المان توسط اپراتور بر روی HMI می‌باشد. در این مثال، ما مقدار Minimum را برابر با 0.0 تنظیم کرده و مقدار Maximum را برابر با 100.0 تنظیم می‌کنیم. همچنین، برای تنظیم مکان نقطه‌ی اعشاری باید مقدار پارامتر Integer Digits را به دلیل داشتن سه رقم صحیح (رقم‌های 0 الی 100) برابر با 3 و پارامتر Fractional را به دلیل داشتن 1 رقم اعشار (رقم‌های 0.0 الی 100.0) برابر با 1 تنظیم کنیم. از این رو، اپراتور قادر است اعدادی بین این دو مقدار (رقم 0.0 الی 100.0) را تنظیم کند.



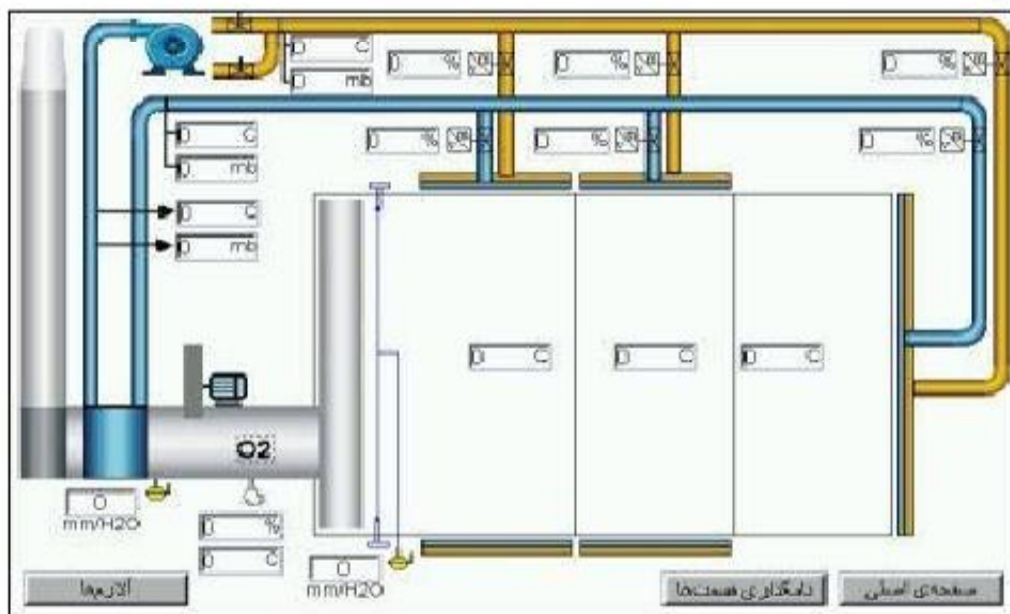
در آخرین مرحله، باید برنامه را کامپایل کرده و در صورت بدون خطا بودن برنامه‌ی نوشته شده، آن را به HMI دانلود کنید. اکنون می‌توانید از طریق HMI به PLC متصل شده و مقادیر آنالوگ جاری را از طریق HMI مانیتور و کنترل کنید.

9-9 نمونه برنامه‌ی مانیتورینگ یک کوره

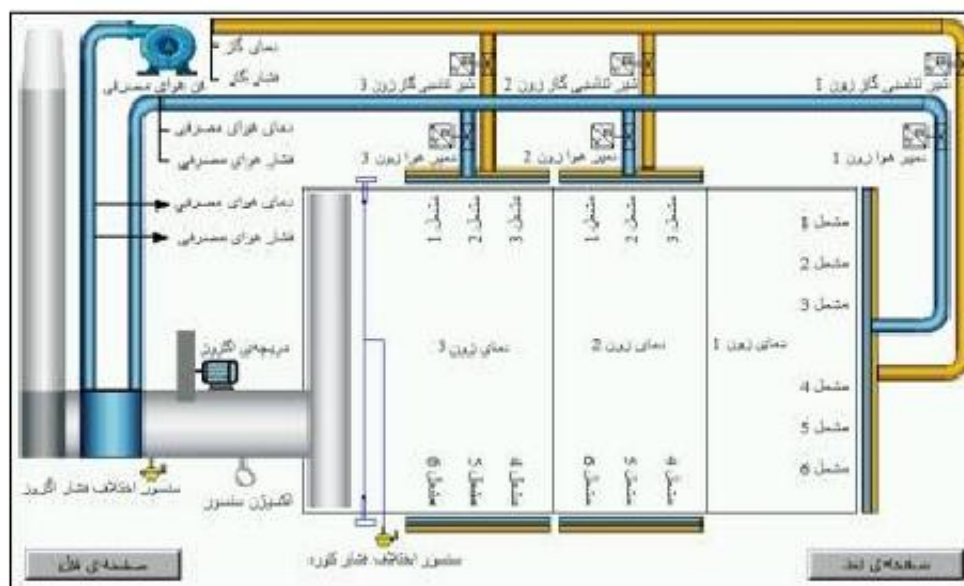
در شکل زیر، نخستین صفحه‌ی طراحی شده برای اتوماتیک کردن یک کوره جهت کنترلر و مانیتورینگ دما و دیگر پارامترهای کوره آورده شده است.



در شکل زیر، صفحه‌ی طراحی شده برای مانیتور کردن پارامترهای کوره جهت نمایش به اپراتور آورده شده است.



در شکل زیر صفحه‌ی طراحی شده برای نامگذاری قسمت‌های مختلف پارامترهای کوره جهت نمایش به اپراتور آورده شده است.



در شکل زیر، صفحه‌ی طراحی شده برای تنظیم پارامترهای زون 1 کوره جهت تنظیم توسط تکنسین و یا مهندس برق آورده شده است.

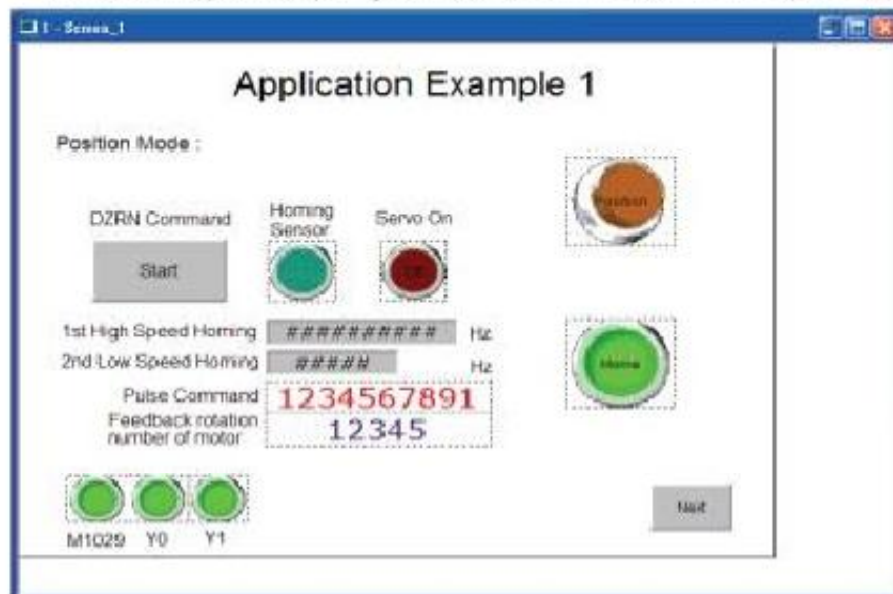


همانطور که در شکل بالا مشاهده می‌کنید، تنظیمات پارامترهای زون 1 همان تنظیمات پارامترهای PID کنترلر اختصاص داده شده به کنترل دمای زون 1 به صورت اتوماتیک می‌باشد.

9-10 یک برنامه‌ی کاربردی جهت مانیتور کردن دو مد موقعیت و سرعت ثابت سروموتور

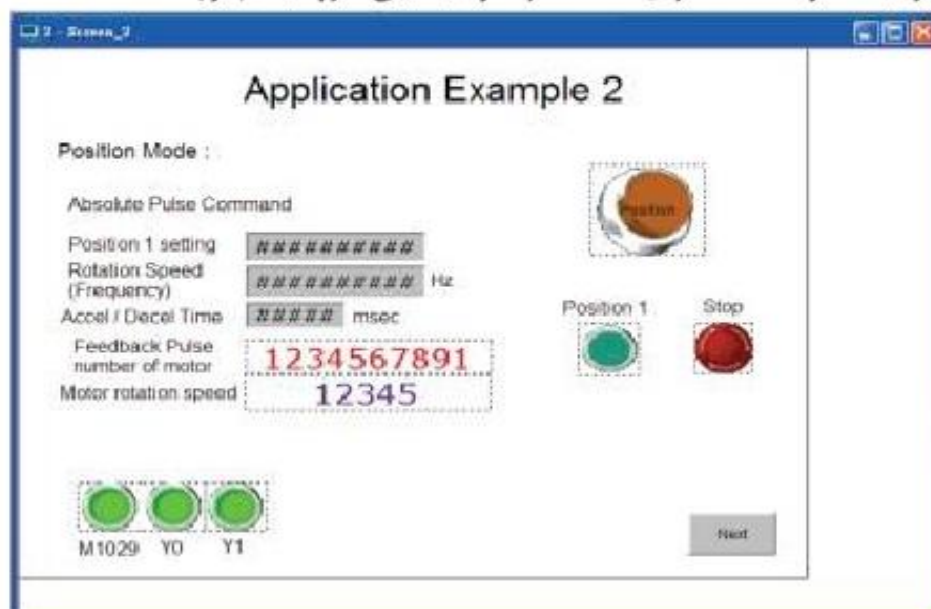
9-10-1 طراحی صفحه‌ی Homing Operation

در شکل زیر، یک نمونه صفحه برای مد عملکرد جایگاه‌یابی سروسیستم آورده شده است.



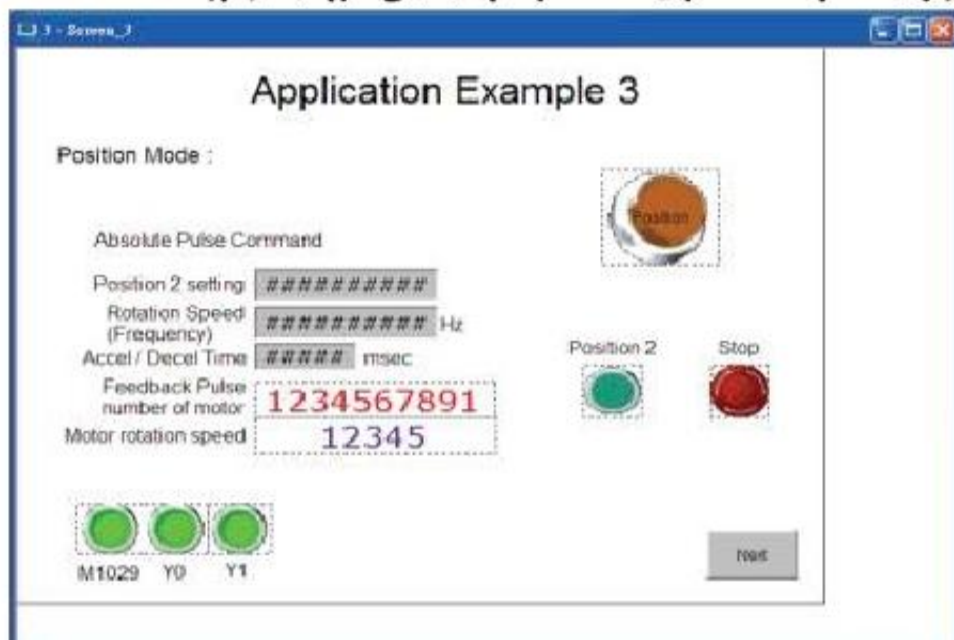
9-10-2 طراحی صفحه‌ی Position Control 1

در شکل زیر، یک نمونه صفحه برای مد عملکرد موقعیت‌یابی سروسیستم آورده شده است.



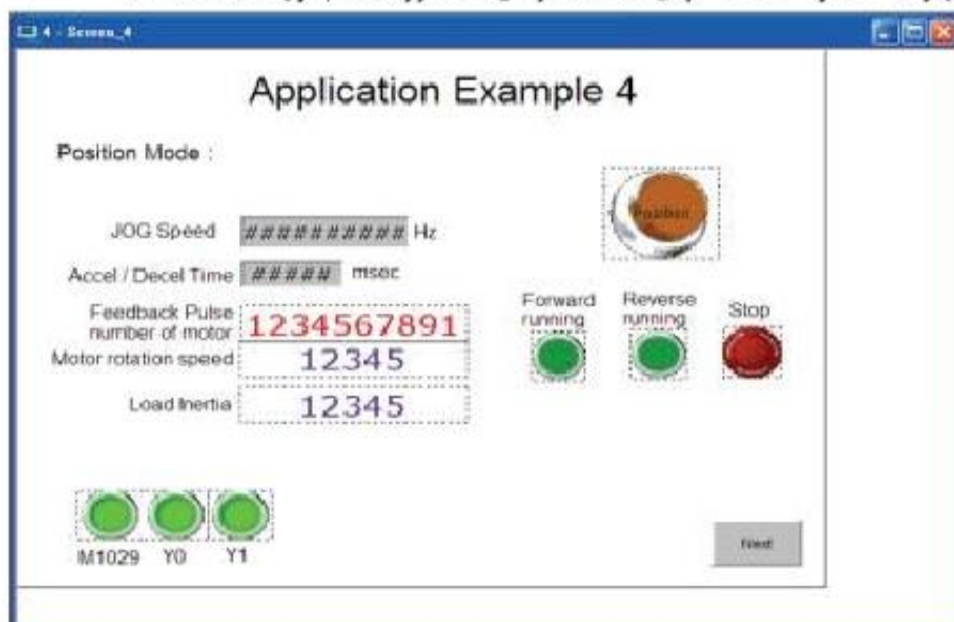
9-10-3 طراحی صفحه‌ی Position Control 2

در شکل زیر، یک نمونه صفحه برای مد عملکرد موقعیتیابی سروسیستم آورده شده است.



9-10-4 طراحی صفحه‌ی JOG Operation

در شکل زیر، یک نمونه صفحه برای مد عملکرد Jog سروسیستم آورده شده است.



9-10-5 طراحی صفحه‌ی Speed Control

در شکل زیر، یک نمونه صفحه برای مد کنترل سرعت سروسیستم آورده شده است.

