

IABU Headquarters

Delta Electronics, Inc.
Taoyuan1
31-1, Xingbang Road, Guishan Industrial Zone,
Taoyuan County 33370, Taiwan, R.O.C.
TEL: 886-3-362-6301 / FAX: 886-3-362-7267

Asia

Delta Electronics (Jiang Su) Ltd.
Wujiang Plant3
1688 Jiangxing East Road,
Wujiang Economic Development Zone,
Wujiang City, Jiang Su Province,
People's Republic of China (Post code: 215200)
TEL: 86-512-6340-3008 / FAX: 86-512-6340-7290

Delta Greentech (China) Co., Ltd.
238 Min-Xia Road, Cao-Lu Industry Zone, Pudong, Shanghai,
People's Republic of China
Post code : 201209
TEL: 021-58635678 / FAX: 021-58630003

Delta Electronics (Japan), Inc.
Tokyo Office
Delta Shibadaimon Building, 2-1-14
Shibadaimon, Minato-Ku, Tokyo, 105-0012,
Japan
TEL: 81-3-5733-1111 / FAX: 81-3-5733-1211

Delta Electronics (Korea), Inc.
234-9, Duck Soo Building 7F, Nonhyun-Dong,
Kangnam-Gu, Seoul, Korea 135-010
TEL: 82-2-515-5305 / FAX: 82-2-515-5302

Delta Electronics Int'l (Singapore) Pte Ltd
4 Kaki Bukit Ave 1, # 05 – 05, Singapore 417939
TEL: 65-6747-5155 / FAX: 65-6744-9228

Delta Electronics (India) Pvt. Ltd.
Plot No. 43, Sector – 35, HSIIDC,
Gurgaon122001, Haryana, India
TEL: 91-124-416-9040 / FAX: 91-124-403-6045

Americas

Delta Products Corporation (USA)
Raleigh Office
P.O. Box 12173, 5101 Davis Drive,
Research Triangle Park, NC 27709, U.S.A.
TEL: 1-919-767-3813 / FAX: 1-919-767-3969

Delta Greentech (Brasil) S/A
Sao Paulo Office
Rua Itapeva, N° 26, 3° andar, Bela vista
ZIP: 01332-000 - São Paulo - SP - Brasil
TEL: 55-11-3568-3875 / FAX: 55-11-3568-3865

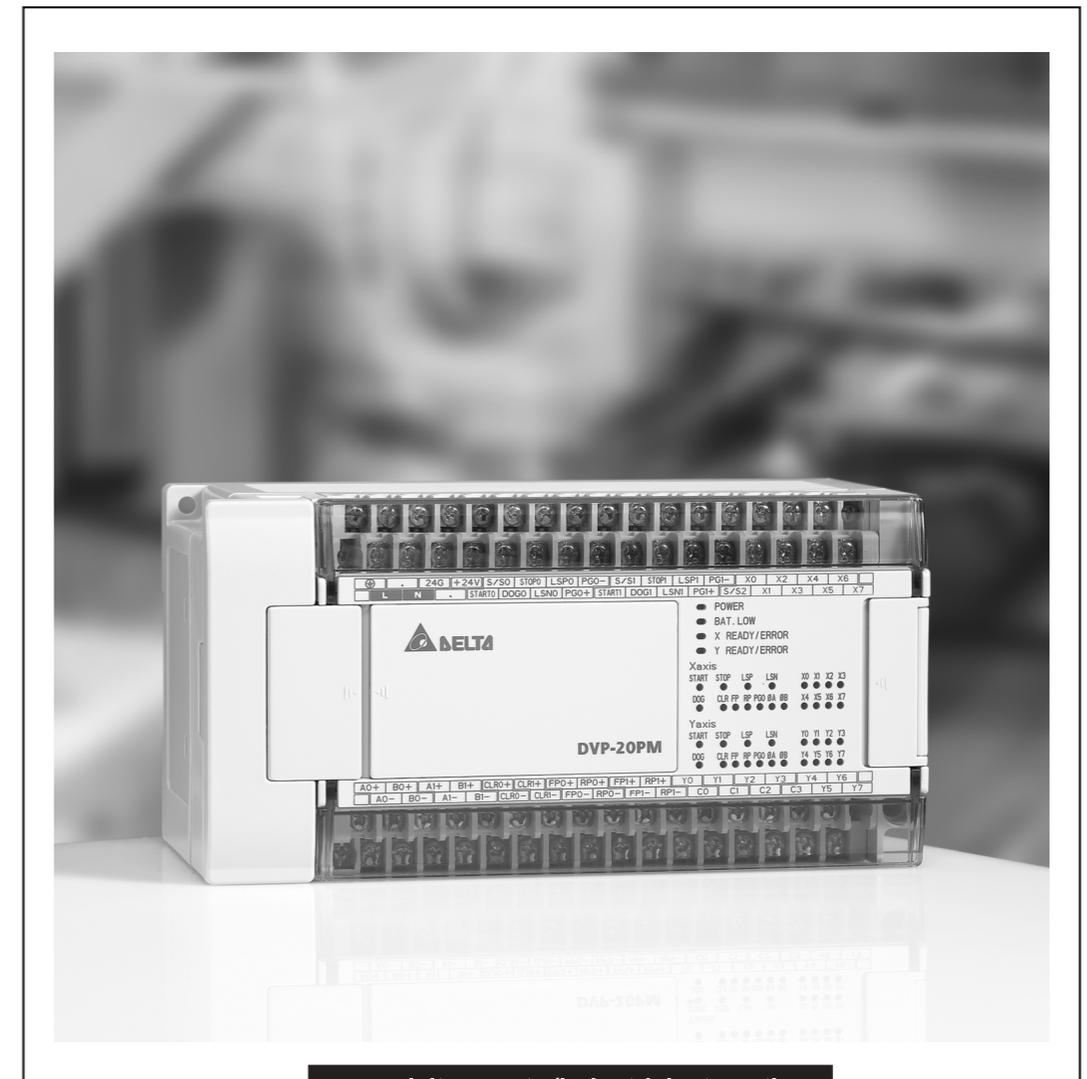
Europe

Deltronics (The Netherlands) B.V.
Eindhoven Office
De Witbogt 15, 5652 AG Eindhoven, The Netherlands
TEL: 31-40-2592850 / FAX: 31-40-2592851

DVP-PM

Application Manual

Programming



DVP-PM APPLICATION MANUAL

Table of Contents

Chapter 1: Program Structure of DVP-PM

1.1	O100 Main Program	1-1
1.2	Structure of OX Motion Subroutine.....	1-3
1.3	Structure of Pn Subroutine	1-5
1.4	Structure of O100, OX and Pn Program Design	1-7
1.4.1	The Program Structure	1-7

Chapter 2: Hardware Specifications and Wiring

2.1	Hardware Specifications.....	2-1
2.1.1	Electrical Specifications	2-1
2.1.2	I/O Point Specifications	2-1
2.1.3	Dimension	2-6
2.2	Installation and Wiring.....	2-8
2.2.1	Wiring	2-8
2.2.2	Power Input Wiring	2-9
2.2.3	Safety Wiring.....	2-9
2.2.4	I/O Point Wiring	2-10
2.2.5	Wiring with Drives.....	2-18
2.3	Communication Ports	2-28
2.3.1	COM1 (RS-232)	2-28
2.3.2	COM2 (RS-485)	2-29
2.3.3	COM2 (RS-232/RS-485).....	2-29

Chapter 3: Functions of Devices in DVP-PM

3.1	Device in DVP-PM	3-1
3.2	Values, Constants [K]/[H], Floating Points [F]	3-4
3.3	Numbering and Functions of External Input/Output Contacts [X]/[Y].....	3-6
3.4	Numbering and Functions of Auxiliary Relays [M]	3-7
3.5	Numbering and Functions of Step Relays [S].....	3-8
3.6	Numbering and Functions of Timers [T].....	3-8

3.7	Numbering and Functions of Counters [C]	3-9
3.8	Numbering and Functions of Registers [D]]	3-16
3.8.1	Data Register [D]	3-16
3.8.2	Index Registers [V], [Z]	3-16
3.9	Pointer [N], Pointer [P _n].....	3-17
3.10	Special Auxiliary Relays [M], Special Data Register [D]	3-18
3.11	Functions of Special Auxiliary Relays and Special Registers.....	3-35
3.12	Special Registers for Manual Motion Settings	3-57
3.12.1	Functions of Special Registers for Manual Mode Settings	3-59
3.12.2	Manual Modes	3-77
3.12.3	Position & Speed Control Registers for Manual Modes.....	3-78

Chapter 4: Basic Instructions

4.1	Basic Instructions	4-1
4.2	Explanations on Basic Instructions	4-3

Chapter 5: Categories and Use of Basic Application Instructions

5.1	List of Instructions	5-1
5.2	Instruction Composition.....	5-5
5.3	Numeric Values for Data Processing	5-7
5.4	Index Register V, Z	5-10
5.5	Instruction Index	5-11
5.6	Application Instructions	5-15
•	(API 00 ~ 09) Loop Control	5-15
•	(API 10 ~ 19) Transmission Comparison.....	5-23
•	(API 20 ~ 29) Four Arithmetic Operations	5-36
•	(API 30 ~ 39) Rotation and Displacement.....	5-48
•	(API 40 ~ 49) Data Processing.....	5-59
•	(API 50) High Speed Processing.....	5-72
•	(API 61 ~ 69) Handy Instructions	5-74
•	(API 78 ~ 87) I/O.....	5-80
•	(API 100 ~ 109) Communications.....	5-85
•	(API 110 ~ 175) Floating Point Operation	5-94
•	(API 215 ~ 223) Contact Type Logic Operation	5-126
•	(API 224 ~ 246) Contact Type Comparison	5-129

• (API 147 ~ 260) Other Instructions.....	5-132
---	-------

Chapter 6: Motion Instructions and G-Code Instructions

6.1 List of Motion Instructions and G-Code Instructions.....	6-1
6.2 Composition of Motion Instructions and G-Code Instructions.....	6-3
6.2.1 Motion Instructions.....	6-3
6.2.2 G-Code Instructions.....	6-4
6.3 Motion Instructions.....	6-6
• (MON 00 ~ 19) Motion Instructions.....	6-6
6.4 O Pointer / M Pointer.....	6-40
6.5 G-Code Instructions.....	6-44
• (G0 ~ 4, 90 ~ 91) G-Code Instructions.....	6-44

Chapter 7: Use DVP-PM As Slave

7.1 Access between DVP-EH2, DVP-PM (as Master) and DVP-PM (as Slave)	7-1
7.1.1 The Structure.....	7-1
7.1.2 Example of Master-Slave Data Exchange.....	7-2

Chapter 8: Application Examples

8.1 Draw the Trajectories by Using Motion Instructions and G-Codes.....	8-1
8.1.1 Trajectory.....	8-1
8.1.2 Design Procedure.....	8-3
8.2 Applying Application Examples in PMSoft.....	8-7
8.2.1 Program Structure.....	8-7
8.2.2 Design Example Program.....	8-8
8.3 Planning Variable Speed Operation.....	8-10
8.3.1 Program structure.....	8-10
8.3.2 Design Example Program.....	8-11
8.4 Connect DVP-PM (Master) with DVP01PU-H2 (Slave) for 3 rd Axis Control.....	8-14

Chapter 9: Electrical Cam

9.1 Introduction to Electrical CAM (E-CAM).....	9-1
9.2 E-CAM Application.....	9-2
9.2.1 Initial Settings.....	9-2
9.2.2 Obtain Master Position.....	9-6

9.2.3 Start / Stop E-CAM	9-9
9.3 Explanations on Special Flags and Registers.....	9-16
9.4 Set up E-CAM Data	9-21
9.4.1 Use PMsoft CAM Chart to Set up E-CAM Data.....	9-21
9.4.2 Use DTO / DFROM Instructions to Set up E-CAM Data	9-27
9.5 Multi-axis E-CAM	9-53
9.6 Field Applications of E-CAM.....	9-57
9.6.1 E-CAM Application on Winding Machine	9-57
9.6.2 E-CAM Application on Rotary Cut and Flying Saw Devices	9-64

Chapter 10: Encrypting User Program

10.1 Password Setting	10-1
10.1.1 System Information	10-1
10.1.2 Downloading Program	10-3
10.1.3 Uploading Program.....	10-6

Chapter 11: G-code Application

11.1 DVP-PM00M Program Download with PEP setting	11-1
11.2 DVP-PM00M G-code Download Methods	11-2
11.2.1 PMGDL Software	11-2
11.2.2 Download by B Type HMI	11-8
11.2.3 Download by HMI other than B type through D registers.....	11-13
11.3 G-code Application in Dispensers	11-17
11.3.1 Operation steps.....	11-17
11.3.2 Storage of G/M code.....	11-22

Chapter 12: POU Editing Mode

12.1 POU Ladder Editing Environment	12-1
12.1.1 Function of Symbols	12-2
12.1.2 Editing Symbol Tables	12-4
12.1.3 Create POU Function Block	12-15
12.1.4 Create POU Folders	12-18
12.1.5 Export POU.....	12-23
12.1.6 Import POU.....	12-30
12.1.7 Symbols Allocation	12-33

12.1.8 Ladder Find	12-36
12.1.9 Ladder Replace	12-40
12.2 Edit POU Ladder Diagram	12-41
12.2.1 Replace Devices with Symbols.....	12-41
12.2.2 Applying POU Function Blocks.....	12-43
12.3 Monitor POU program.....	12-50
12.4 Hint Function on Symbols and Function Blocks.....	12-54

Chapter 13: CANopen Communication Card

13.1 Introduction to DVP-FPMC: CANopen Communication Card	13-1
13.2 Specifications.....	13-1
13.3 Product Profile & Installation.....	13-2
13.4 Parameters for Control Register (CR).....	13-3
13.5 How to Set Up ASDA-A2 Servo Drive	13-5
13.6 DVP-FPMC Mode Settings	13-20
13.7 DVP-FPMC Ethernet Applications	13-24
13.7.1 Communication between DVP-FPMC and HMI	13-24
13.7.2 Communication between DVP-FPMC and PMSOFT	13-27
13.8 LED Indicator & Troubleshooting.....	13-30

Chapter 14: High Speed Compare and Capture

14.1 High Speed Compare and Capture Function.....	14-1
14.2 High Speed Compare Function.....	14-4
14.3 Capture Function	14-8

Chapter 15: Appendix

15.1 Appendix A: Error Codes	14-1
15.2 Appendix B: Manual Revision History	14-2
15.3 Appendix C: Function Comparison Table of DVP-PM Models.....	14-2

1 Program Structure of DVP-PM

Delta's DVP-PM series MPU is a high-speed positioning and multi-functional programmable logic controller with 2-axis linear/arc interpolation, featuring functions as basic instructions, application instruction (API), motion instruction (MON) and G-code instructions, making the editing and compiling of program more diverse.

This chapter will introduce the program structure of DVP-PM series MPU. DVP-PM combines the sequential control and 2-axis interpolation positioning control; therefore, the program is in three types: O100 main program, OX motion subroutine and Pn subroutine, which will be illustrated in this chapter.

The basic instructions, application instructions (API), motion instructions (MON) and G-Code instructions will be given in Chapter 4 ~ 6.

Here we provide a specification comparison table of different model PLCs for your reference:

Specifications	PLC Models	
	DVP-10PM	DVP-20PM
High speed output	1000kHz 4 sets	500kHz 3 sets
PWM	Accuracy 0.3%@200KHz	
Hardware high speed counter	6 sets (differentialx2, open collector x4)	2 sets
Program capacity	64K Steps	64K Steps
Execution speed	LD: 0.14us MOV: 2us DMUL: 7.6us DEMUL: 6.1us	

1.1 O100 Main Program

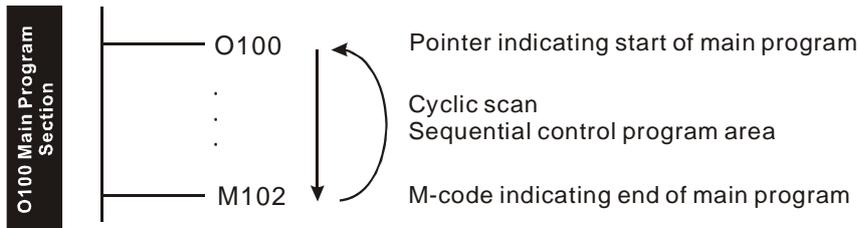
O100 main program is the PLC sequential control program for DVP-PM series MPU. The O100 main program section only supports basic instructions and application instructions. Besides processing I/O signals and calling Pn subroutine, basic instructions and application instructions also control 100 OX motion subroutines which enable OX0 ~ OX99. Therefore, O100 main program establishes the main control program, and the main control program sets up and activates motion subroutines. This is the control structure of the operation of DVP-PM. See below the operation procedure and features of O100 main program.

1. There are two ways to activate O100 main program
 - When DVP-PM is powered, and the AUTO/MANU switch goes from MANU to AUTO, M1072 will be ON automatically, and O100 main program will be in RUN status.
 - When DVP-PM is powered, you can set M1072 to be ON or O100 main program to be in RUN status by communication.

1 Program Structure of DVP-PM



- O100 main program operates in cyclic scans. When O100 main program is enabled, the scan will start at the start flag of O100. When the scan reaches M102 (main program ends instruction), it will return to the start flag of O100 and repeat the scan, as shown in the figure below:



- There are three ways to stop the operation of O100 main program:
 - When DVP-PM is powered, and the AUTO/MANU switch goes from AUTO to MANU, M1072 will be OFF automatically, and O100 main program will be in STOP status. The operation of OX and Pn subroutines will stop at this moment.
 - When DVP-PM is powered, you can set M1072 to be OFF or O100 main program to be in STOP status by communication. The operation of OX and Pn subroutines will stop immediately.
 - When errors occur during compiling or operation of the program, O100 main program will stop automatically. See Chapter 14 Index for the table of the error codes and their causes.
- O100 main program supports basic instructions and application instructions; therefore, you can design the program according to your actual needs. Besides, you can further activate OX0 ~ OX99 motion subroutines by setting up the parameters in motion subroutines and the activation No. in the motion subroutines.
 - O100 main program does not support motion instructions and G-Code instructions. Please design motion instructions and G-Code instructions in OX0 ~ OX99 subroutines. See 1.2 for more details.
 - O100 main program is able to call Pn subroutines. See 1.3 for more details.
- The above explanations are sorted in the table below:

O100 main program	Explanation
Start of O100	O100, pointer indicates the start of O100 main program (*It will be inserted by PMSOft when compiling to IL instructions, therefore you don't have to add it into ladder diagram.)
End of O100	M102, instruction indicates the end of O100 main program (*It will be inserted by PMSOft when compiling to IL instructions, therefore you don't have to add it into ladder diagram.)
Activation	1. MANU → AUTO by MANU/AUTO switch ON DVP-PM 2. M1072 OFF → ON by communication
Execution	Operated in cyclic scans
Instruction supported	Basic instructions and application instructions
Quantity	Only one O100 program is allowed in the program

1 Program Structure of DVP-PM

Features & functions	<ol style="list-style-type: none">1. Performs sequential control of PLC2. Able to activate OX0 ~ OX99 motion subroutines and call Pn subroutines3. Can be placed in front of or after OX0 ~ OX99 motion subroutines and Pn subroutines.
----------------------	---

6. Manual Motion Mode in O100 Main Program

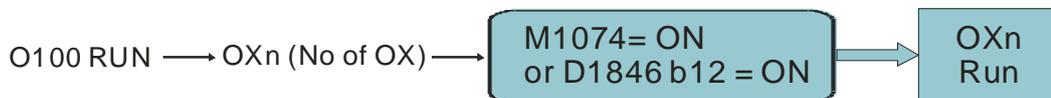
In O100 main program, you can use special registers for designing Manual Motion Mode. (Please see 3.12 for details of how to set up).

1.2 Structure of OX Motion Subroutine

OX0 ~ OX99 motion subroutines are motion control programs for controlling the motions ON X/Y/Z axes in DVP-PM. The OX0 ~ OX99 motion subroutines support basic instructions, application instructions, motion instructions and G-Code instructions. Also OX subroutines are able to call Pn subroutines. OX0 ~ OX99 are instructions provided specially for controlling the moving path ON X/Y/Z axes. See below the operation procedure and features of OX motion subroutines.

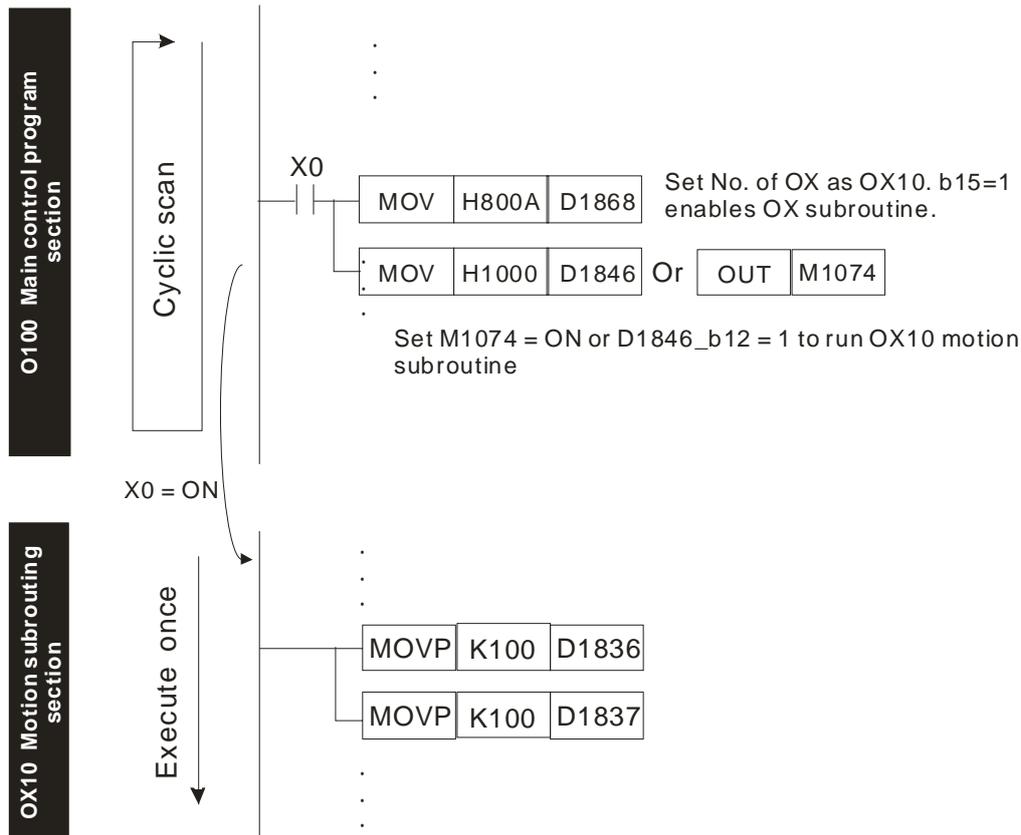
1. How to activate OX0 ~ OX99 motion subroutines:

- When O100 main program is in RUN status, you can set up the execution No. of OX00~OX99 (D1868: H8000 ~ H8063) in O100 main program. After number of OX subroutine is specified, set M1074 (Enabling OX subroutines) as ON or set b12 of D1846 as 1 to enable OX motion subroutine.
- Before enabling OX subroutine, make sure there are no other motion subroutines in operation.



2. OX subroutine executes once when it is activated. When O100 main program activates OX subroutine, the execution will start from the start flag of OX motion subroutine and end at M2 (end of OX subroutine) as shown in the figure below:

1 Program Structure of DVP-PM



When X0 = On, OX10 motion subroutine will execute once and stop when the execution reaches M2. If you need to execute it again, activate X0 again and OX10 subroutine will be executed again.

3. There are four ways to stop OX motion subroutine:
 - When DVP-PM is powered, and the AUTO/MANU switch goes from AUTO to MANU, M1072 will be OFF automatically. O100 main program will be in STOP status. The operation of OX motion subroutines will stop at this moment.
 - You can also stop OX subroutine by controlling the input signals of the external terminal (STOP0).
 - When DVP-PM is powered, you can also stop OX subroutine through communication by setting M1074 to be OFF or D1846 as 0.
 - When errors occur during compiling or operation of the program, OX subroutines will stop automatically. See Chapter 11 Index for the table of the error codes and their causes.
4. OX motion subroutines support basic instructions, application instructions (API), motion instructions (MON) and G-Code instructions. Therefore, you can design your own motion program by using these instructions and setting up X-Y-Z axis parameters for your desired X-Y-Z motion control.
 - The instructions mentioned above shall be designed in OX0 ~ OX99 motion subroutines.
 - OX motion subroutine is able to call Pn subroutines. See 1.3 for more details.
5. The above explanations are sorted in the table below:

OX motion subroutine	Explanation
Start of OX	OXn, pointer indicates the start of OX subroutine. OX0 ~ OX99, 100 motion subroutines

1 Program Structure of DVP-PM

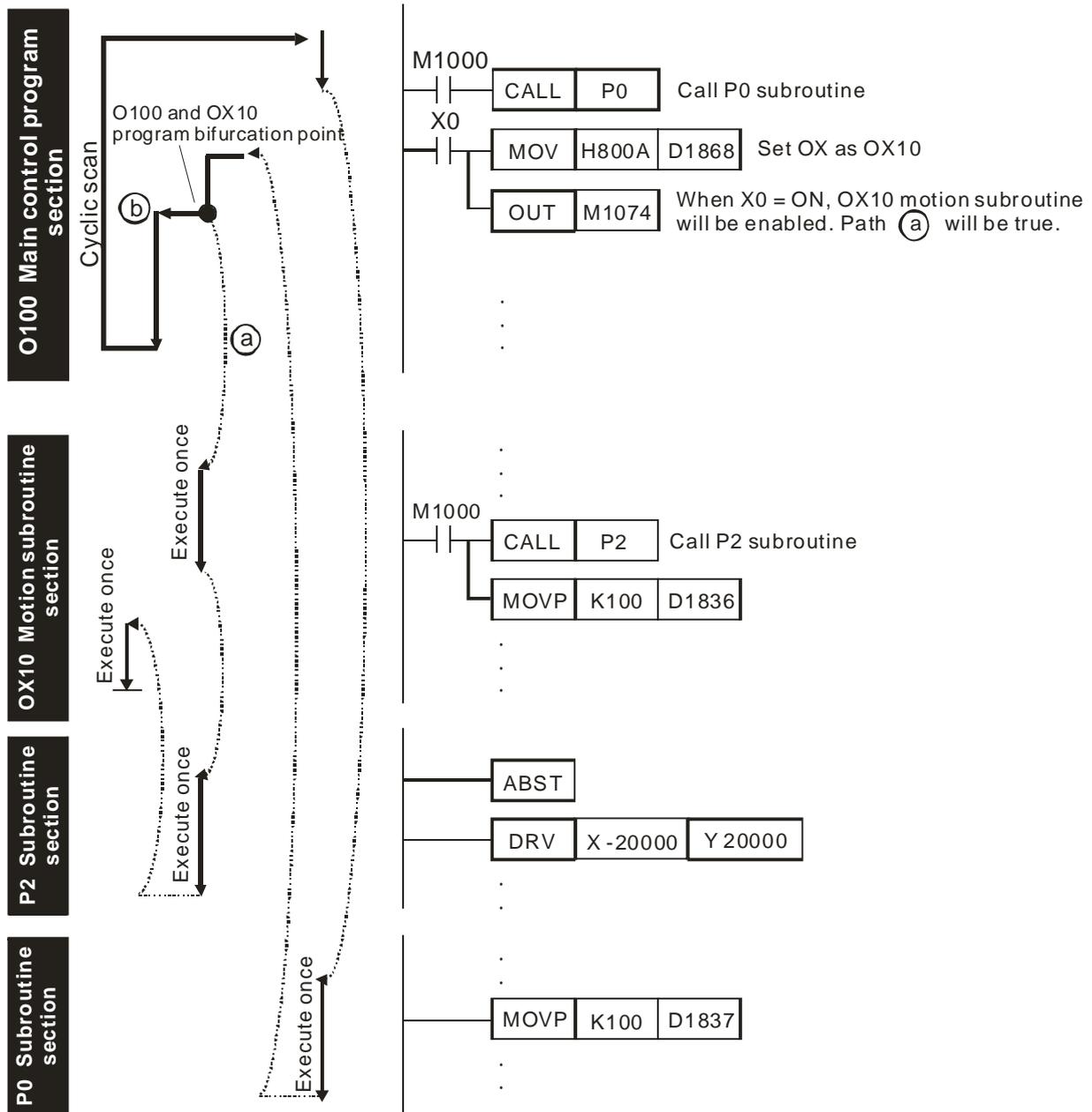
	(*It will be inserted by PMSoft when compiling to IL instructions, therefore you don't have to add it into ladder diagram.)
End of OX	M2, instruction indicates the end of OX subroutine (*It will be inserted by PMSoft when compiling to IL instructions, therefore you don't have to add it into ladder diagram.)
Activation	<ol style="list-style-type: none"> 1. When O100 main program is in RUN status, set M1074 = ON or D1846_b12 = 1 to enable OX motion subroutine by PMSoft. 2. When O100 main program is in RUN status, set M1074 = ON or D1846_b12 = 1 to enable OX motion subroutine through communication. 3. OX subroutine can be stopped by input signals at external control terminal (STOP0). <p>☐ Note: When you need to enable OX motion subroutine, make sure there are no other motion subroutines in operation.</p>
Execution	Execute once whenever the subroutine is enabled. Re-enable it for the re-execution.
Instruction supported	<p>Basic instructions, application instructions (API), motion instructions (MON), and G-Code instructions.</p> <p>☐ Note: Avoid pulse-type instruction when using basic instructions and application instructions.</p>
Quantity	100 OX motion subroutines are available. To activate more OX subroutines, please set up the number of OX (OXn) in D1868 and set M1074 = ON or D1846_b12 = 1 again.
Features & functions	<ol style="list-style-type: none"> 1. Motion subroutine especially for motion control. (Can be activated only by O100) 2. Offers the third axis (Z) control. See 6.4 G00 and G01 instructions for more details. 3. Can be enabled / disabled by controlling the external terminals, program design and communication. 4. Able to call Pn subroutine. 5. Can be placed in front of or after O100 main program and Pn subroutines.

1.3 Structure of Pn Subroutine

Pn subroutine is a general-purpose subroutine which can be called in O100 main program and OX motion subroutines. When Pn subroutine is called in O100 main program, the Pn subroutine area supports basic instructions and application instructions. When Pn subroutines is called in OX0 ~ OX99 motion subroutines, the Pn subroutine area supports basic instructions, application instructions (API), motion instructions (MON) and G-Code instructions.

1. How to enable Pn subroutine:
 - Call Pn subroutine in O100 main program.
 - Call Pn subroutine in OX motion subroutine.
2. No matter Pn subroutine is called in O100 or OX, the execution will jump to Pn subroutine to execute the Pn subroutine once. When SRET (Pn subroutine ends instruction) is reached, the execution will return to the row under CALL Pn instruction and proceed with the remaining program as below.

1 Program Structure of DVP-PM



- P0 subroutine section called by O100 supports only basic instructions and application instructions. However, P2 subroutine section called by OX subroutine supports basic instructions, application instructions (API), motion instructions (MON) and G-Code instructions.

3. There are three ways to stop Pn subroutine:

- When DVP-PM is powered, and the AUTO/MANU switch goes from AUTO to MANU, M1072 will be OFF automatically, and O100 main program will be in STOP status. The operation of OX and Pn subroutines will also stop at this moment.
- When DVP-PM is powered, you can also stop OX subroutine by setting M1074 to be OFF or D1846 as 0 through communication, so as to stop the Pn subroutine called by OX subroutine.
- When errors occur during compiling or operation of the program, Pn subroutines will stop automatically. See Chapter 14 Index for the table of the error codes and their causes.

4. The above explanations are sorted in the table below:

1 Program Structure of DVP-PM

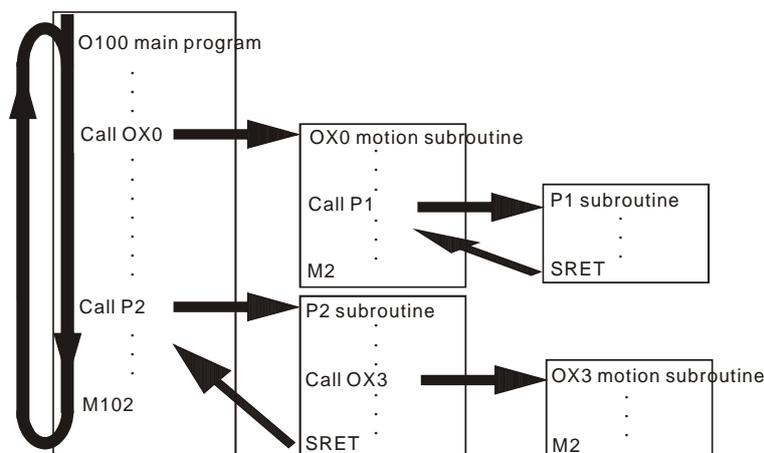
Pn subroutine	Explanation
Start of the program	Pn, start flag of Pn subroutine (P0 ~ P255) (*It will be inserted by PMSoft when compiling to IL instructions, therefore you don't have to add it into ladder diagram.)
End of the program	SRET, instruction indicating the end of Pn subroutine (*It will be inserted by PMSoft when compiling to IL instructions, therefore you don't have to add it into ladder diagram.)
Activation	1. Call Pn subroutine in O100 main program. 2. Call Pn subroutine in OX motion subroutine.
Execution	Execute once whenever the subroutine is enabled. Re-enable it for the re-execution.
Instructions supported	1. When called in O100: supports basic instructions and application instructions 2. When called in OX: supports basic instructions, application instructions (API), motion instructions (MON) and G-Code instructions. Note: When you need to call Pn in OX and use basic instructions and application instructions, please avoid pulse-type instructions.
Quantity	256 Pn subroutines are available
Features & functions	1. A general-purpose subroutine 2. Can be called in O100 main program and OX motion subroutine 3. Can be placed in front of or after O100 main program and OX subroutines.

1.4 Program Design Structure of O100, OX and Pn

O100 main program, OX motion subroutine and Pn subroutine are introduced in 1.1 ~ 1.3. In this section, we will further illustrate the program structure mixed with the 3 types of programs.

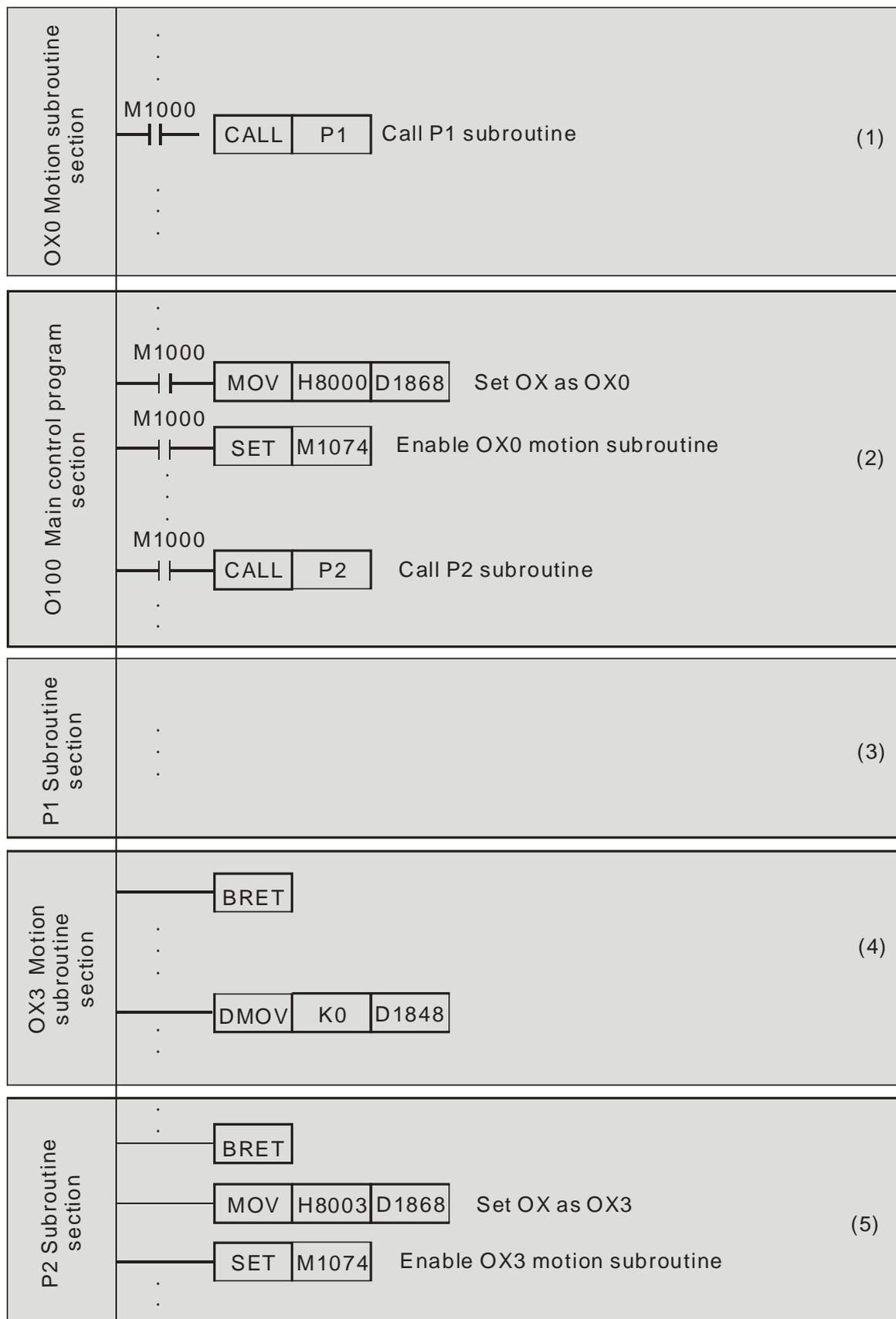
1.4.1 The Program Structure

Assume we would like to design an O100 main program, an OX0 motion subroutine, an OX3 motion subroutine, a P1 subroutine and a P2 subroutine (5 program sections) as the following sequential control process:



To explain the example in an easier way, the program design will be given in section (1) ~ (5), as shown below:

1 Program Structure of DVP-PM



1 Program Structure of DVP-PM

Explanations ON the program design:

1. The design order is from (1) to (5) in above example. However, there is no rule for the placing order of program sections, so you can place the 5 sections according to your needs.
2. There should be only one O100 main program (2), and it cannot be called by other programs or subroutines. O100 can call OX motion subroutines and Pn subroutines.
3. OX motion subroutine can be called by O100 main program and Pn subroutine, and it can also call a Pn subroutine.
4. Pn subroutine can be called by O100 main program and OX motion subroutine, and it can also call a OX motion subroutine.

Note:

1. Only one OX motion subroutine can be executed at a time. Therefore, when OX0 motion subroutine is executed, OX3 will not be able to work, and vice versa.
2. Once O100 main program or Pn subroutine enables an OX motion subroutine, it will continue to execute the next row of the program without paying attention to the OX motion subroutine being executed.
3. The enabled OX motion subroutine will only execute once. If you want it to execute again, you have to re-enable it.

Instructions supported in each program section: (O: supported; X: not supported)

Section	O100 main program	OX motion subroutine (OX0, OX3)	P1 subroutine	P2 subroutine
Basic instruction	O	O	O	O
Application instruction	O	O	O	O
Motion instruction	X	O	O	X
G-Code instruction	X	O	O	X
Explanation	Instructions supported are fixed	Instructions supported are fixed	Called by OX motion subroutine; therefore, motion instructions and G-Code instructions are supported.	Called by O100 main program; therefore, motion instructions and G-Code instructions are not supported.

Remarks:

	Main program	Subroutine	Motion subroutine
Placing order	No limitation	No limitation	No limitation
Activation	In RUN Status	Called by main program or motion subroutine	Called by main program or subroutine
Execution	In cycles	Execute once whenever being called once	Execute once whenever being called once
Quantity	1	256, depending ON user's demand.	100, depending ON user's demand.

2 Hardware Specifications and Wiring

2.1 Hardware Specifications

This chapter only provides information on electrical specification and wiring. For detailed information on program design and instructions, please refer to Chapter 5 ~ 6. For how to purchase its peripheral devices, please refer to the instruction sheet enclosed with the product.

2.1.1 Electrical Specifications

	20PM	10PM
Power supply voltage	100 ~ 240V AC (-15% ~ 10%), 50/60Hz ± 5%	
Power supply fuse	2A/250VAC	
Power consumption	60VA	
DC24V current output	500mA	
Power supply protection	DC24V; output short-circuit protection	
Voltage withstand	1,500VAC (Primary-secondary); 1,500VAC (Primary-PE); 500VAC (Secondary-PE)	
Insulation resistance	> 5MΩ (all I/O point-to-ground: 500VDC)	
Noise immunity	ESD: 8KV Air Discharge; EFT: Power Line: 2KV, Digital I/O: 1KV, Analog & Communication I/O: 250V	
Grounding	The diameter of grounding wire shall not be less than that of L, N terminal of the power. (When many PLCs are in use at the same time, please make sure every PLC is properly grounded.)	
Environment	Operation: 0°C ~ 55°C (temperature), 50 ~ 95% (humidity), pollution degree 2 Storage: -25°C ~ 70°C (temperature), 5 ~ 95% (humidity)	
Vibration/shock resistance	International standards: IEC61131-2, IEC 68-2-6 (TEST Fc)/IEC61131-2 & IEC 68-2-27 (TEST Ea)	
Weigh (approx. g.)	478/688	

2.1.2 I/O Point Specifications

Input point specifications:

■ DVP20PM00D

Terminal	Description	Response time	Max. input	
			Current	Voltage
START0, START1	For input signals of Start	10ms	6mA	24V
STOP0, STOP1	For input signals of Stop	10ms	6mA	24V
LSP0/LSN0, LSP1/LSN1	Right limit input/left limit input	10ms	6mA	24V
A0+, A0-, A1+, A1-	MPG A-phase pulse input +, - (differential signal input) (A1+, A1- are shared by Y and Z)	200kHz	15mA	5 ~ 24V
B0+, B0-, B1+, B1-	MPG B-phase pulse input +, - (differential signal input) (B1+, B1- are shared by Y and Z)	200kHz	15mA	5 ~ 24V

2 Hardware Specifications and Wiring

Terminal	Description	Response time	Max. input	
			Current	Voltage
PG0+, PG0-, PG1+, PG1-	Zero point signal input +, - (differential signal input)	200kHz	16mA	5 ~ 24V
DOG0, DOG1	There are two variations according to different operation modes: 1. DOG signal in zero return mode 2. Start signal for inserting single speed or 2-speed operation	1ms	6mA	24V
X0~X7	General input points	200kHz	15mA	24V

■ DVP20PM00M

Terminal	Description	Response time	Max. input	
			Current	Voltage
START0, START1	For input signals of Start	10ms	6mA	24V
STOP0, STOP1	For input signals of Stop	10ms	6mA	24V
LSP0/LSN0, LSP1/LSN1	Right limit input/left limit input	10ms	6mA	24V
X1/X2	Right limit input/left limit input (COM) (for Z axis)	10ms	6mA	24V
A0+, A0-, A1+, A1-	MPG A-phase pulse input +, - (differential signal input) (A1+, A1- are shared by Y and Z)	200kHz	15mA	5 ~ 24V
B0+, B0-, B1+, B1-	MPG B-phase pulse input +, - (differential signal input) (B1+, B1- are shared by Y and Z)	200kHz	15mA	5 ~ 24V
PG0+, PG0-, PG1+, PG1-	Zero point signal input +, - (differential signal input)	200kHz	16mA	5 ~ 24V
X3	Zero point signal input (COM) (for Z axis)	100ms	6mA	24V
DOG0, DOG1	There are two variations according to different operation modes: 1. DOG signal in zero return mode 2. Start signal for inserting single speed or 2-speed operation	1ms	6mA	24V
X0	Same as DOG0, DOG1 (COM) (for Z axis)	10ms	6mA	24V
X4~X7	General input points	200kHz	15mA	24V

■ DVP10PM00M

Terminal	Description	Response time	Max. input	
			Current	Voltage
X0~X7	1. single phase / A-B phase input 2. DOG signal for X-Y-Z-A axis: X0, X2, X4, X6 3. PG signal for X-Y-Z-A axis: X1, X3, X3, X7	200kHz	15mA	24V
X10+, X10-, X11+, X11-	MPG+/- pulse input (differential input)	200kHz	15mA	5 ~ 24V
X12+, X12-, X13+, X13-	Differential input	200kHz	15mA	5 ~ 24V

Output point specifications:

2 Hardware Specifications and Wiring

■ DVP20PM00D

Terminal	Description	Response time	Max. output current
CLR0+, CLR0-, CLR1+, CLR1-	Clear signals (for clearing the error counter in servo drive)	10ms	20mA
FP0+, FP0-, FP1+, FP1-	Forward / reverse running mode: Forward pulse output Pulse / direction mode: Pulse output terminal A, B phase mode: A-phase output	500kHz	40mA
RP0+, RP0-, RP1+, RP1-	Forward / reverse running mode: Reverse pulse output Pulse / direction mode: Direction output terminal A, B phase mode: B-phase output	500kHz	40mA
Y0 ~ Y7	General output points	200kHz	40mA

■ DVP20PM00M

Terminal	Description	Response time	Max. output current
CLR0+, CLR0-, CLR1+, CLR1-	Clear signals (for clearing the error counter in servo drive)	10ms	20mA
Y2	Same as clear signals (for Z axis)	10ms	30mA
FP0+, FP0-, FP1+, FP1- FP2+, FP2-	Forward / reverse running mode: Forward pulse output Pulse / direction mode: Pulse output terminal A, B phase mode: A-phase output	500kHz	40mA
RP0+, RP0-, RP1+, RP1- RP2+, RP2-	Forward / reverse running mode: Reverse pulse output Pulse / direction mode: Direction output terminal A, B phase mode: B-phase output	500kHz	40mA
Y3 ~ Y7	Open collector type high-speed pulse output	200kHz	40mA

■ DVP10PM00M

Terminal	Description	Response time	Max. output current
Y0 ~ Y3	Open collector type high-speed pulse output	200kHz	40mA
Y10+, Y10-, Y12+, Y12-, Y14+, Y14-, Y16+, Y16-	Forward / reverse running mode: Forward pulse output Pulse / direction mode: Pulse output terminal A, B phase mode: A-phase output	1000kHz	40mA
Y11+, Y11-, Y13+, Y13- Y15+, Y15-, Y17+, Y17-	Forward / reverse running mode: Reverse pulse output Pulse / direction mode: Direction output terminal A, B phase mode: B-phase output	1000kHz	40mA

Digital input points:

2 Hardware Specifications and Wiring

■ DVP20PM

Item		24V DC single-ended common point input		Note
		Low speed	High speed (200kHz) ^{#1}	
Spec				
Input wiring type		Change wiring from S/S to SINK or SOURCE		#1: Input point A, B and PG are high-speed inputs; others are low-speed input.
Input indicator		LED display; light on = ON, light off = OFF		
Input voltage		-		
Active level	OFF→ON	20us		For 10PM, X0 ~ X7 and X10+ ~ X13+ can be used as general high-speed input points.
	ON→OFF	30us		
Response time / noise immunity ^{#2}		10ms	0.5us	#2: Input point X0 ~ X7 can conduct 10 ~ 60ms digital filter adjustment.

■ DVP10PM

Item		Double-ended differential input	24V DC single-ended common point input	Note
		High speed (200kHz)		
Spec				
Input wiring type		Independent wiring	Change wiring from S/S to SINK or SOURCE	#1: Input points support 10 ~ 60ms digital filter adjustment, and 20~85KHz high-frequency digital filter adjustment.
Input indicator		LED display; light on = ON, light off = OFF		
Input voltage		5~24 VDC	24 VDC	
Input current		15mA		
Active level	OFF→ON	20us		
	ON→OFF	30us		
Response time / noise immunity ^{#1}		10ms / 0.5us		

Digital output points:

■ DVP-20PM

Item		Single-ended common point transistor output ^{#1}		Single-ended common point ^{#1} relay output
		Low speed	High speed	
Spec				
Maximum frequency		10kHz	200kHz	For ON/OFF control of load
Output indicator		LED display; light on = ON, light off = OFF		
Minimum load		-		2mA/DC power supply
Working voltage		5 ~ 30VDC		< 250V AC, 30VDC
Isolation		Photocoupler isolation		Electromagnetic isolation
Current spec.	Resistive	0.5A/1 point (2A/COM)		2A/1 point (5A/COM)
	Inductive	9W (24V DC)		^{#2}
	Lamp	2W (24V DC)		20W DC/100W AC
Max. output	OFF→ON	20us	0.2us	10ms

2 Hardware Specifications and Wiring

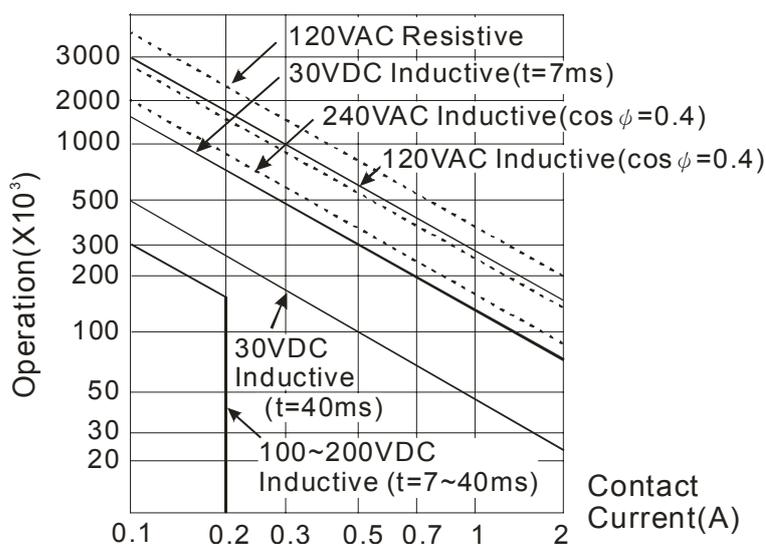
Item		Single-ended common point transistor output ^{#1}		Single-ended common point ^{#1} relay output
		Low speed	High speed	
Spec				
delay time	ON→OFF	30us		
Over-current protection		N/A		

■ DVP-10PM

Item		Double-ended differential output	Single-ended common point relay output
		Spec	
Maximum frequency		1 MHz	200 kHz
Output indicator		LED display; light on = ON, light off = OFF	
Output point configuration		Y10 ~ Y17	Y0 ~ Y3
Working voltage		5 VDC	5 ~ 30VDC
Max. output current		40 mA	40 mA
Isolation		Line driver	Photocoupler isolation
Current spec.	Resistive	< 25 mA	0.5A/1 點 (4A/COM)
	Inductive	--	12W (24 VDC)
	Lamp	--	2 W (24 VDC)
Max. output delay time	OFF→ON	0.2 us	
	ON→OFF		
Over-current protection		N/A	

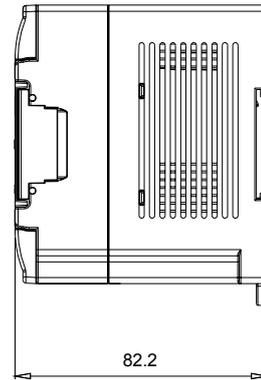
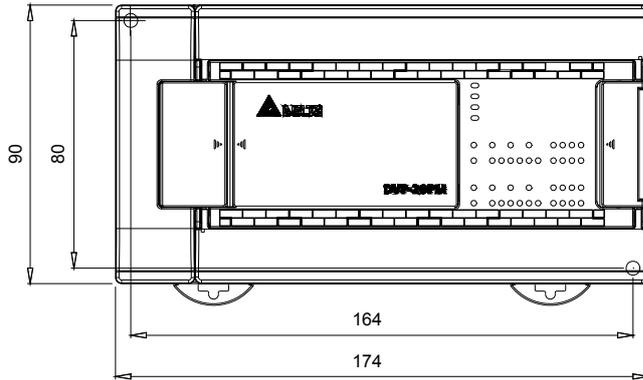
#1: Y0 ~ Y7 on DVP20PM00D are relay output; FP2+, FP2- on DVP20PM00M are high-speed transistor output, Y2 and Y3 are low-speed transistor output, Y4 ~ Y7 are relay output. Common point for Y0: C0; common point for Y1: C1; common point for Y2 and Y3: C2; common point for Y4~Y7: C3. Output points of DVP10PM are all high-speed transistor outputs, and Y0~Y3 can be used as general output points.

#2: The life cycle curve:



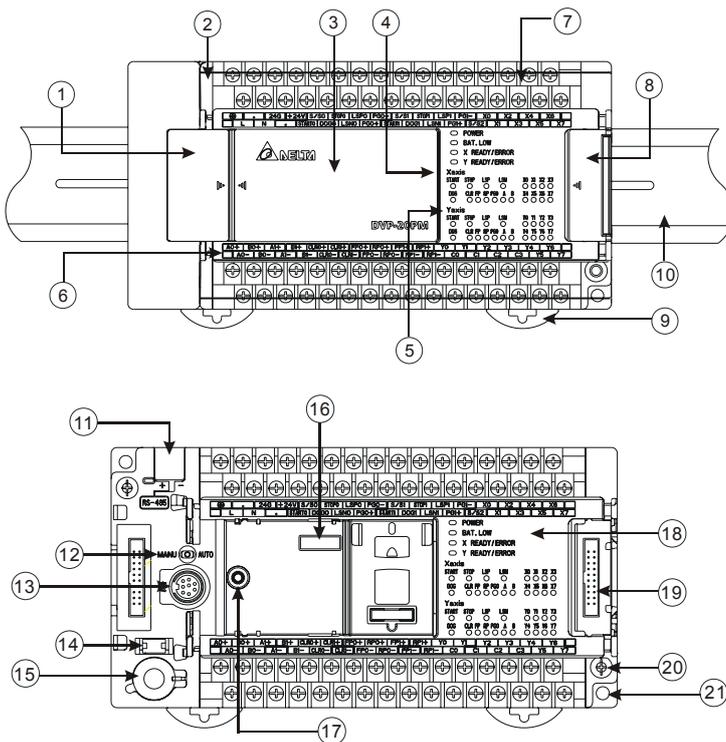
2 Hardware Specifications and Wiring

2.1.3 Dimension



(Unit: mm)

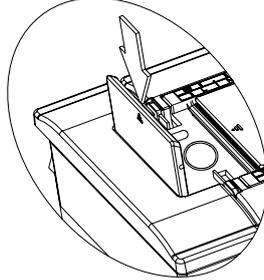
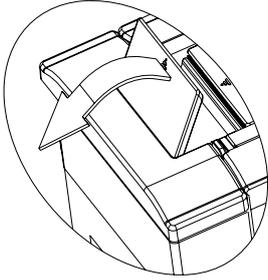
Product Profile & Outline:



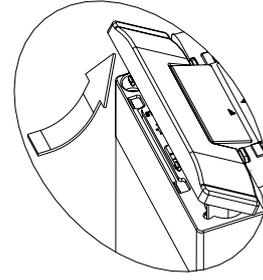
- | | |
|----|----------------------------------|
| 1 | Communication port cover |
| 2 | I/O terminal cover |
| 3 | Function card cover |
| 4 | Input indicator |
| 5 | Output indicator |
| 6 | I/O terminal No. |
| 7 | I/O terminals |
| 8 | I/O module connection port cover |
| 9 | DIN rail clip |
| 10 | DIN rail (35mm) |
| 11 | COM2 (RS-485) |
| 12 | MANU/AUTO (STOP/RUN) switch |
| 13 | COM1 (RS-232) |
| 14 | Battery socket |
| 15 | Battery |
| 16 | Function card port |
| 17 | Function card fixing hole |
| 18 | POWER/ERROR/BAT.LOW indicators |
| 19 | I/O module connection port |
| 20 | Case fixing screw |
| 21 | Direct mounting hole |

2 Hardware Specifications and Wiring

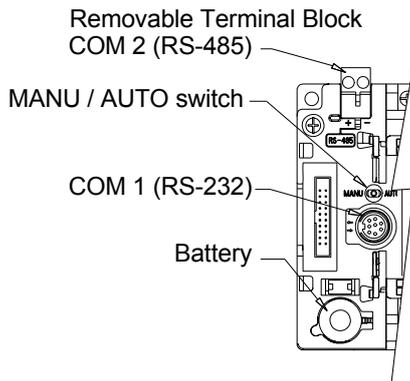
COM1 cover



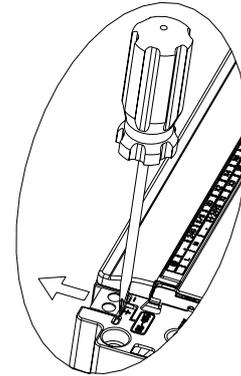
Left-side port cover



Ports under left-side port cover



Screw driver is required for removing RS-485 terminal



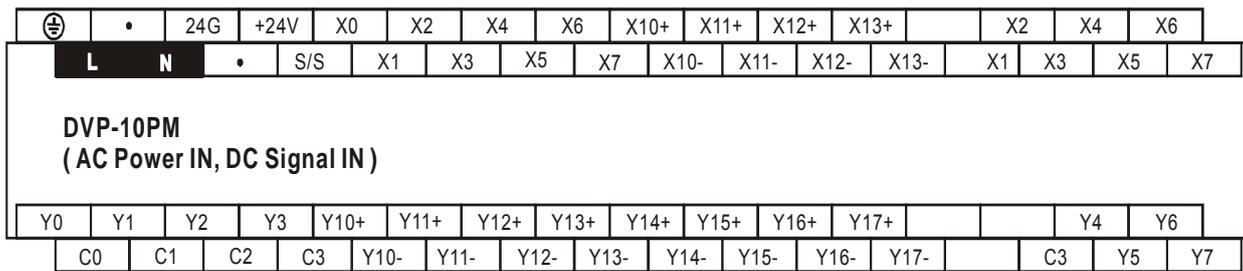
Part	Description
COM2 (RS-485)	For both master and slave modes
MANU/AUTO (STOP/RUN) switch	RUN/STOP control
COM1 (RS-232)	Slave mode (can be used with COM2 at the same time)
Battery	The battery shall be changed within 1 minute

Wiring Terminals: See 2.1.2 for detailed electrical specifications

⊕	•	24G	+24V	S/S0	STOP0	LSP0	PG0-	S/S1	STOP1	LSP1	PG1-	X0	X2	X4	X6
L	N	•	START0	DOG0	LSN0	PG0+	START1	DOG1	LSN1	PG1+	S/S2	X1	X3	X5	X7
DVP-20PM00D (AC Power IN, DC Signal IN)															
A0+	B0+	A1+	B1+	CLR0+	CLR1+	FP0+	RP0+	FP1+	RP1+	Y0	Y1	Y2	Y3	Y4	Y6
A0-	B0-	A1-	B1-	CLR0-	CLR1-	FP0-	RP0-	FP1-	RP1-	C0	C1	C2	C3	Y5	Y7

⊕	•	24G	+24V	S/S0	STOP0	LSP0	PG0-	S/S1	STOP1	LSP1	PG1-	X0	X2	X4	X6
L	N	•	START0	DOG0	LSN0	PG0+	START1	DOG1	LSN1	PG1+	S/S2	X1	X3	X5	X7
DVP-20PM00M (AC Power IN, DC Signal IN)															
A0+	B0+	A1+	B1+	CLR0+	CLR1+	FP0+	RP0+	FP1+	RP1+	FP2+	RP2+	Y2	Y3	Y4	Y6
A0-	B0-	A1-	B1-	CLR0-	CLR1-	FP0-	RP0-	FP1-	RP1-	FP2-	RP2-	C2	C3	Y5	Y7

2 Hardware Specifications and Wiring



2.2 Installation & Wiring

DVP-PM is an OPEN-TYPE device and therefore should be installed in an enclosure free of airborne dust, humidity, electric shock and vibration. The enclosure should prevent non-maintenance staff from operating the device (e.g. key or specific tools are required for opening the enclosure) in case danger and damage on the device may occur.

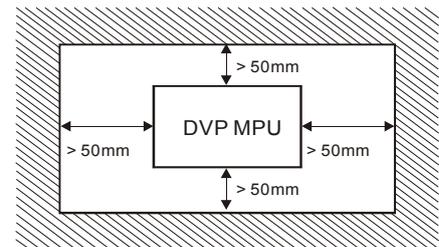
DO NOT connect AC power input to any of the I/O terminals; otherwise serious damage may occur. Check all the wiring again before switching on the power. To prevent electromagnetic interferences, make sure the PLC is properly grounded by ground terminal ⊕.

2.2.1 Mounting and Wiring Notes

Please install DVP-PM in an enclosure with sufficient space around it to allow heat dissipation, as shown in the figure.

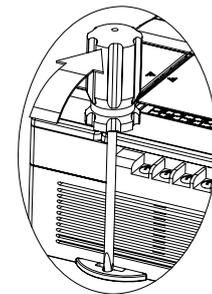
Direct Mounting:

Please use M4 screw according to the dimension of the product.

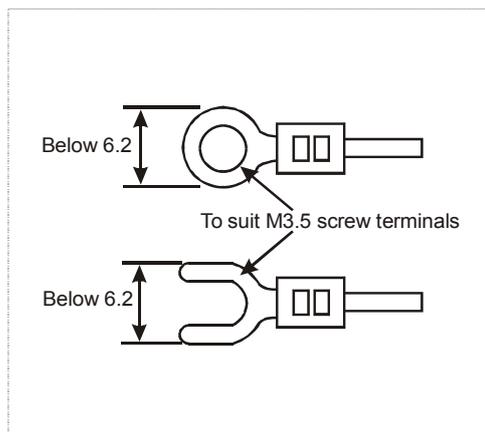


DIN-rail Mounting:

DVP-PM can be secured to a cabinet by using the 35mm DIN rail. When mounting the PLC to 35mm DIN rail, be sure to use the retaining clip to stop any side-to-side movement of the PLC and reduce the chance of wires being loose. The retaining clip is at the bottom of the PLC. To secure the PLC to DIN rail, pull down the clip, place it onto the rail and gently push it up. To remove the PLC, pull the retaining clip down with a flat screwdriver and gently remove the PLC from DIN rail.



Wiring notes:



1. Use O-type or Y-type terminals. See the figure in the left hand side for its specification. PLC terminal screws should be tightened to 9.50 kg-cm (8.25 in-lbs). Use 60/75°C copper conductor only.
2. DO NOT wire empty terminal. DO NOT place the input signal cable and output power cable in the same wire duct.
3. DO NOT drop tiny metallic conductor into the PLC while screwing and wiring. Please attach the dustproof sticker to the PLC before the installation to prevent conductive objects from dropping in. Tear off the sticker before running the PLC to ensure normal heat dissipation.

2 Hardware Specifications and Wiring

2.2.2 Power Input Wiring

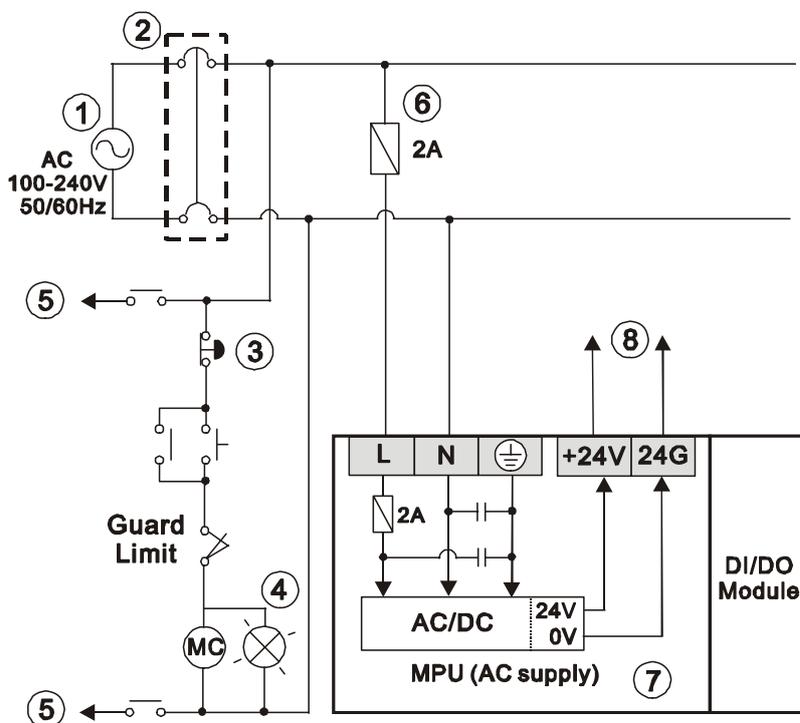
The power input type for DVP-PM is AC input. When operating DVP-PM, please note the following points:

1. The range of the input voltage should be 100 ~ 240VAC. The power input should be connected to L and N terminals. Please note that wiring AC110V or AC220V to +24V output terminal or digital input points will result in serious damage on the PLC.
2. The AC power inputs for the MPU and the digital I/O module should be ON or OFF at the same time.
3. Use wires of 1.6mm diameter (or bigger size) for the grounding of PLC.
4. The power shutdown of less than 10ms will not affect the operation of the PLC. However, power shutdown time that is too long or the drop of power supply voltage will stop the running of the PLC, and all outputs will go "OFF". When the power returns to normal status, the PLC will automatically resume operation. (Care should be taken on the latched auxiliary relays and registers inside the PLC when programming.)
5. The +24V output is rated at 0.5A from MPU. DO NOT connect other external power supplies to this terminal. Every input terminal requires 5 ~ 7mA to be driven; e.g. the 16-point input will require approximately 100mA. Therefore, +24V terminal cannot give output to the external load that is more than 400mA.

2.2.3 Safety Wiring

In PLC control system, many devices are controlled at the same time and actions of any device could influence each other, i.e. breakdown of any device may cause danger or even the breakdown of the entire auto-control system.

Therefore, we suggest you wire a protection circuit at the power supply input terminal. See the figure below.



① AC power supply: 100 ~ 240VAC, 50/60Hz

② Circuit breaker

③ Emergency stop. This button cuts off the system power supply when accidental emergency occurs.

④ Power indicator

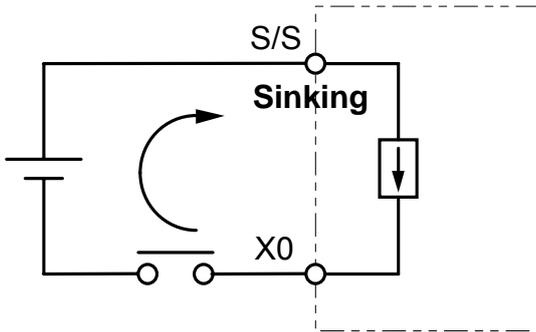
2 Hardware Specifications and Wiring

- ⑤ AC power supply load
- ⑥ Power supply circuit protection fuse (3A)
- ⑦ DVP-PM MPU
- ⑧ DC Power supply output: 24VDC, 500mA

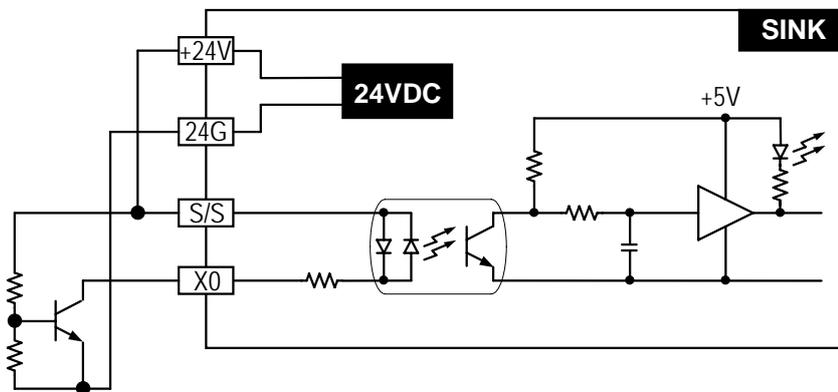
2.2.4 I/O Point Wiring

1. The type of input signal is DC input and there are two types of DC input: **SINK** and **SOURCE**.

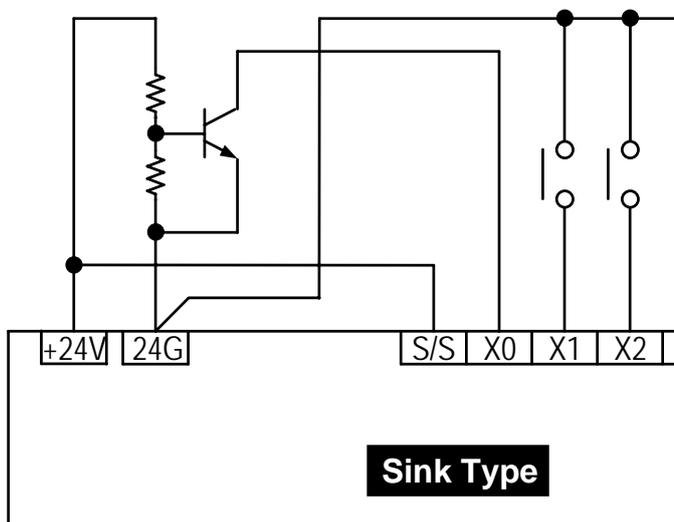
- DC Signal IN - SINK mode:



Equivalent Circuit for Input Point:

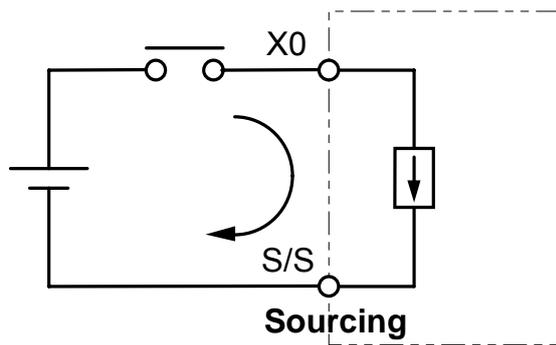


Wiring Circuit:

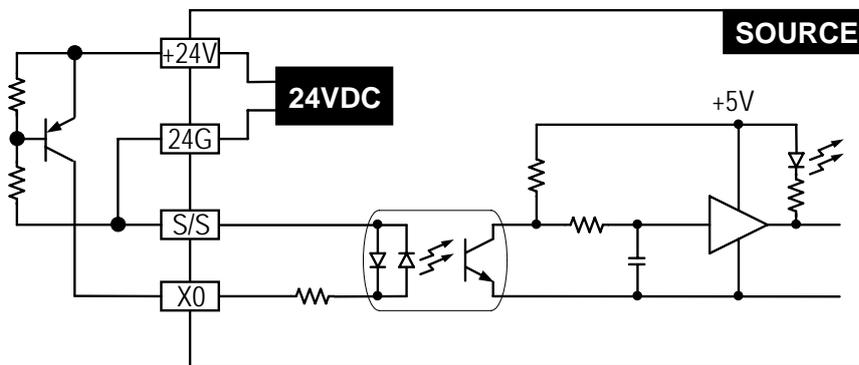


2 Hardware Specifications and Wiring

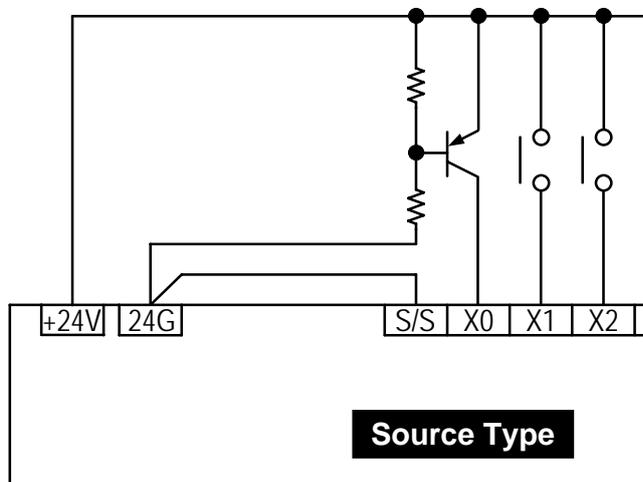
- DC Signal IN - SOURCE mode:



Equivalent Circuit for Input Point



Wiring Circuit:

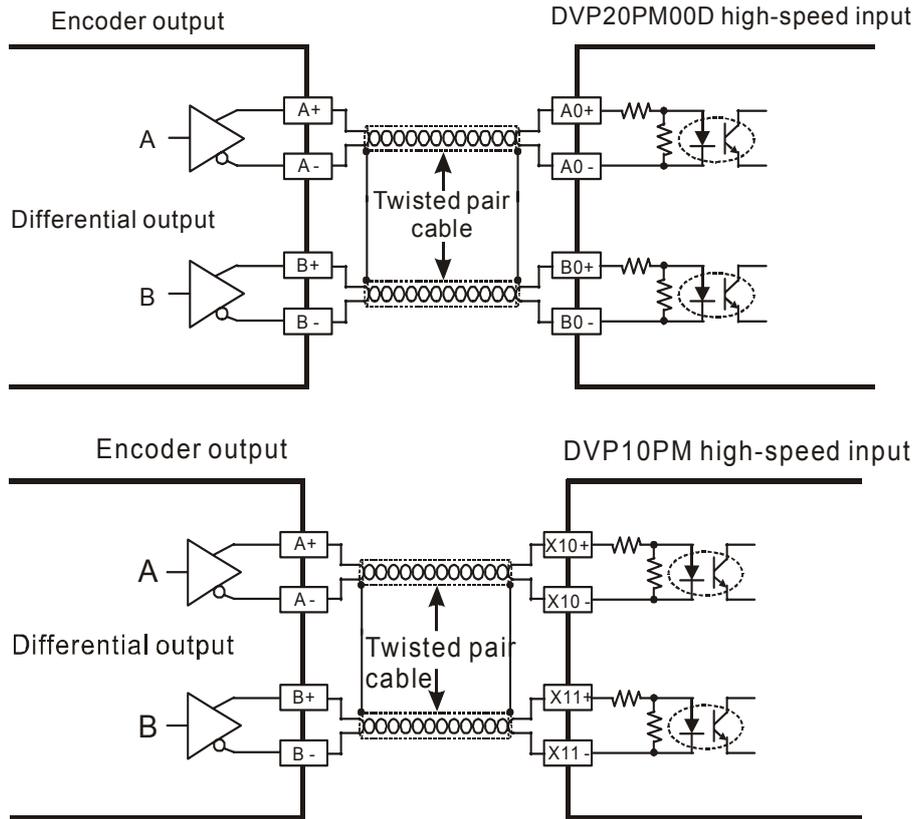


2. Wiring of Differential Input:

A0 ~ A1 and B0 ~ B1 of DVP-PM series are all DC5V ~ 24V high-speed input circuit and others are DC24V input. The working frequency of high-speed input circuit can reach up to 200kHz and is applied mainly for connecting to output circuit with dual differential line driver.

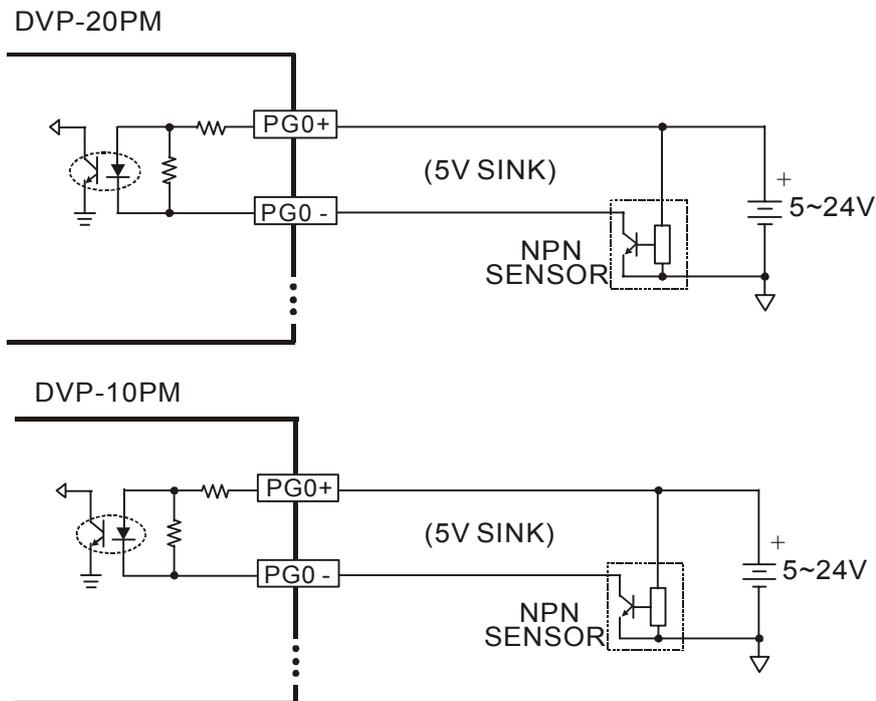
Wiring in a high-speed, high-noise environment

2 Hardware Specifications and Wiring



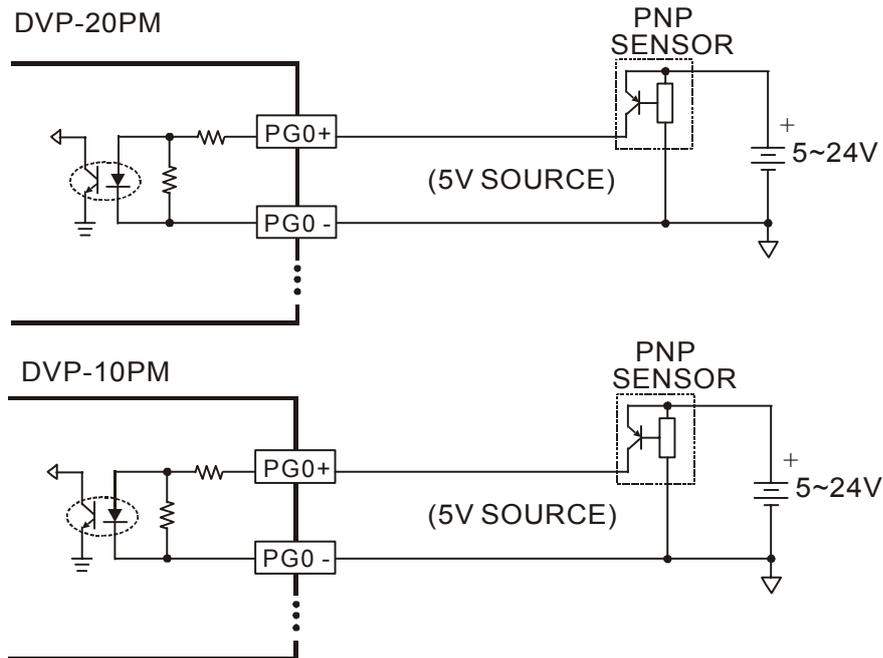
In low-noise and low-frequency (less than 50kHz) environment, you can also use single-ended DC 5V ~ 24V SINK/SOURCE input. See below for the input wirings:

- Wiring of DVP-PM DC5V SINK

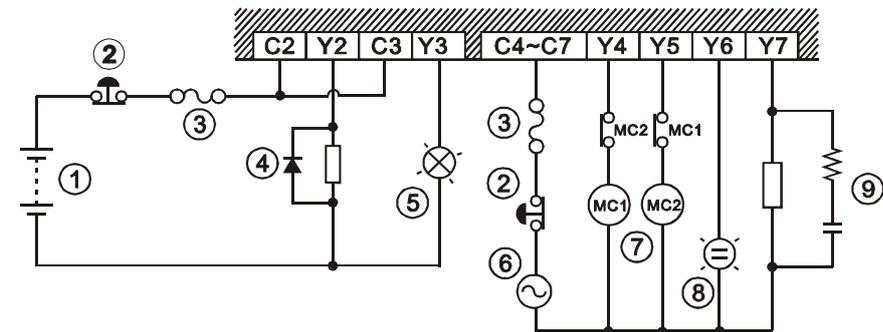


2 Hardware Specifications and Wiring

- Wiring of DVP-PM DC5V SOURCE



3. Wiring of Relay (R) Output Circuit



① DC power supply

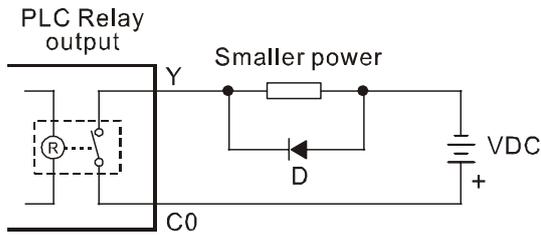
② Emergency stop: Uses external switch

③ Fuse: Uses 5 ~ 10A fuse at the shared terminal of output contacts to protect the output circuit

2 Hardware Specifications and Wiring

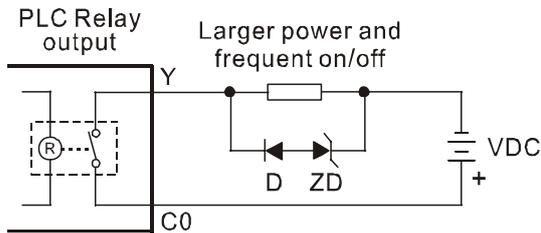
④ Transient voltage suppressor: To extend the life span of contact.

1. Diode suppression of DC load: Used when in smaller power.



D: 1N4001 diode or equivalent component

2. Diode + Zener suppression of DC load: Used when in larger power and frequent ON/OFF operation



D: 1N4001 diode or equivalent component
ZD: 9V Zener, 5W

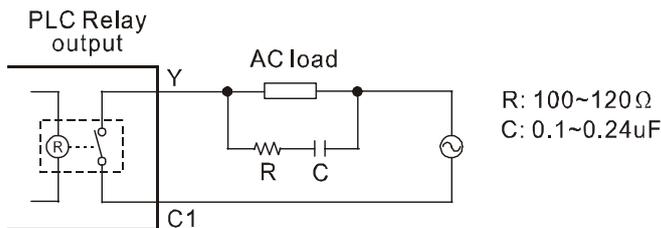
⑤ Incandescent light (resistive load)

⑥ AC power supply

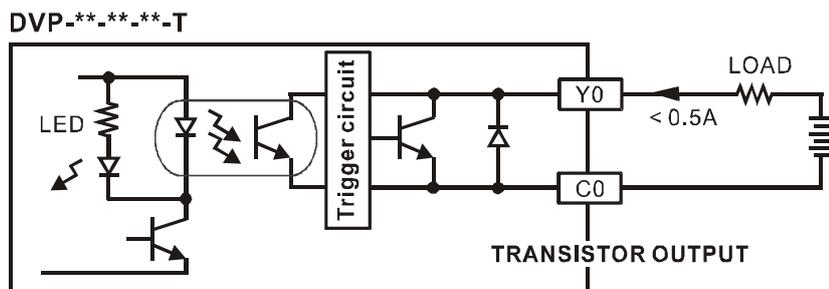
⑦ Manually exclusive output: For example, Y4 and Y5 control the forward running and reverse running of the motor, forming an interlock for the external circuit, together with the PLC internal program, to ensure safe protection in case of any unexpected errors.

⑧ Neon indicator

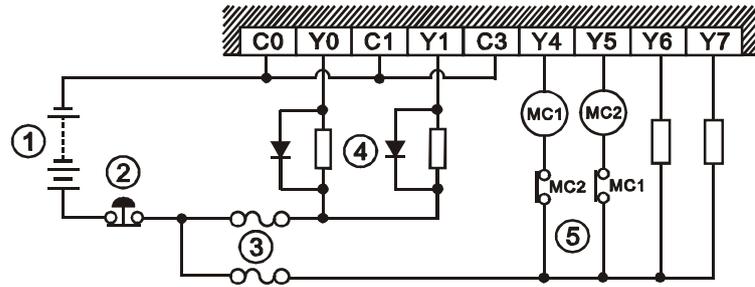
⑨ Surge Absorber: To reduce the interference on AC load.



4. Wiring of Transistor (T) Output Circuit



2 Hardware Specifications and Wiring

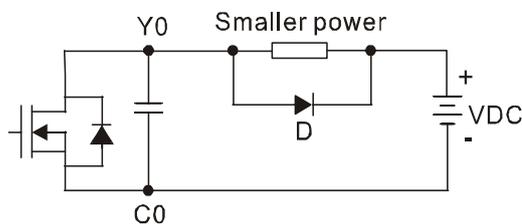


- ① DC power supply ② Emergency stop ③ Circuit protection fuse

④ The transistor output model of DVP-PM applies “open collector output”. Therefore if Y0/Y1 is set as pulse output which requires higher operation frequency, the output current passes pull-up resistor has to be bigger than 0.1A to ensure normal operation of PLC.

1. Diode suppression: Used when in smaller power.

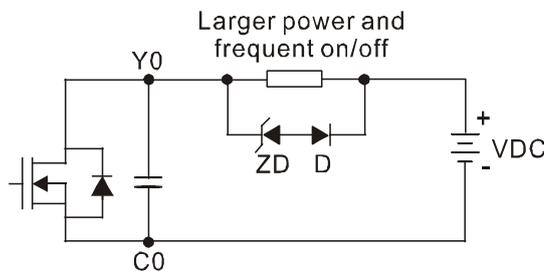
PLC Transistor output



D: 1N4001 diode or equivalent component

2. Diode + Zener suppression: Used when in larger power and frequent On/Off.

PLC Transistor output



D: 1N4001 diode or equivalent component

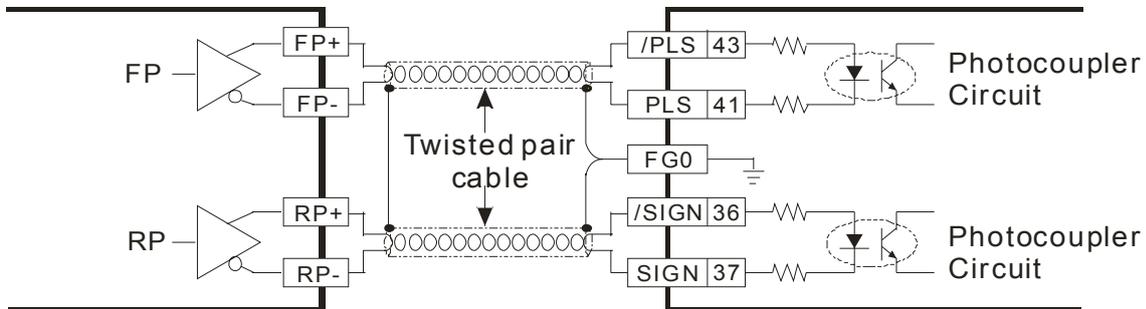
⑤ Manually exclusive output: For example, assume that Y4 and Y5 control the forward running and reverse running of the motor. In this case, you can design an interlock for the external circuit together with the PLC internal program to ensure safe operation in case of any unexpected errors.

2 Hardware Specifications and Wiring

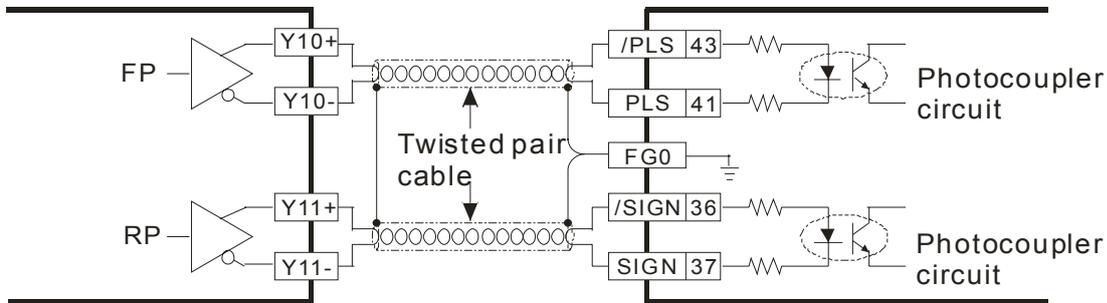
5. Wiring of Differential Output

- Wiring of DVP-PM differential output with ASDA-A, ASDA-A+ or ASDA-A2 series servo drive

DVP-20PM differential output

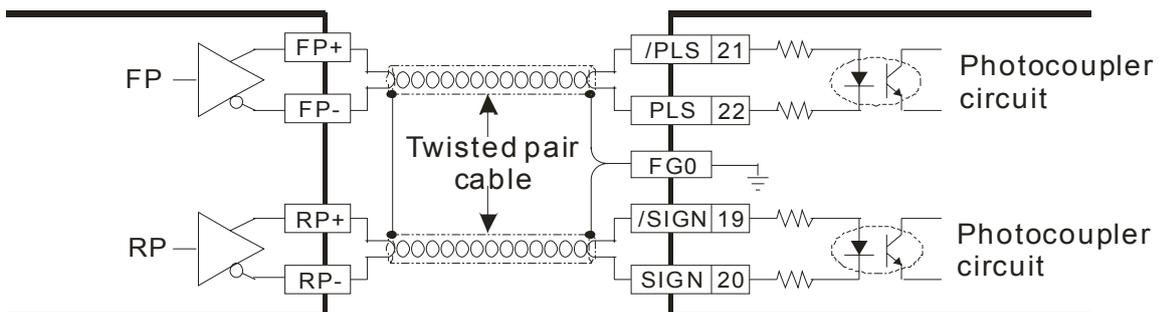


DVP-10PM differential output

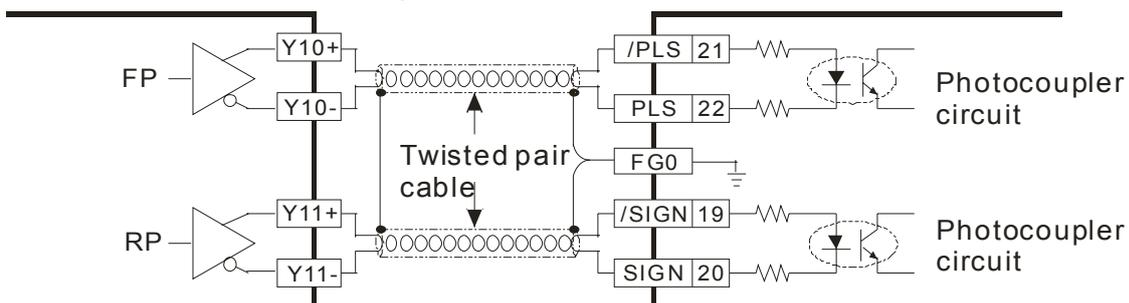


- Wiring of DVP-PM differential output with ASDA-B series servo drive

DVP-20PM differential output

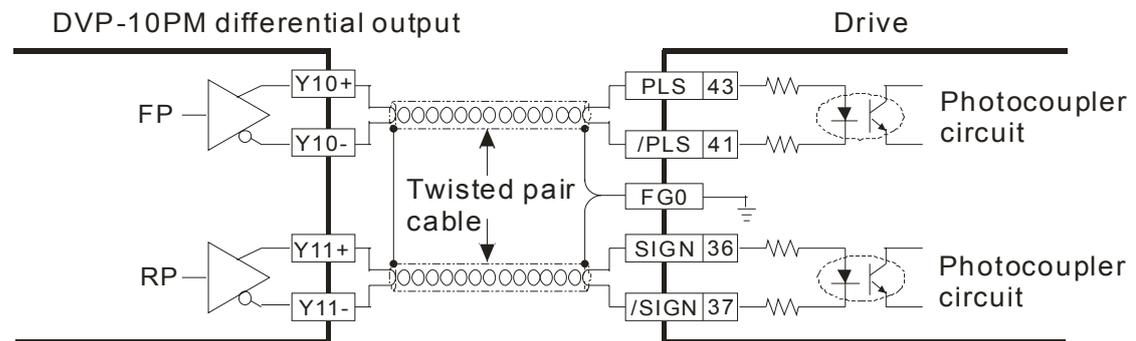
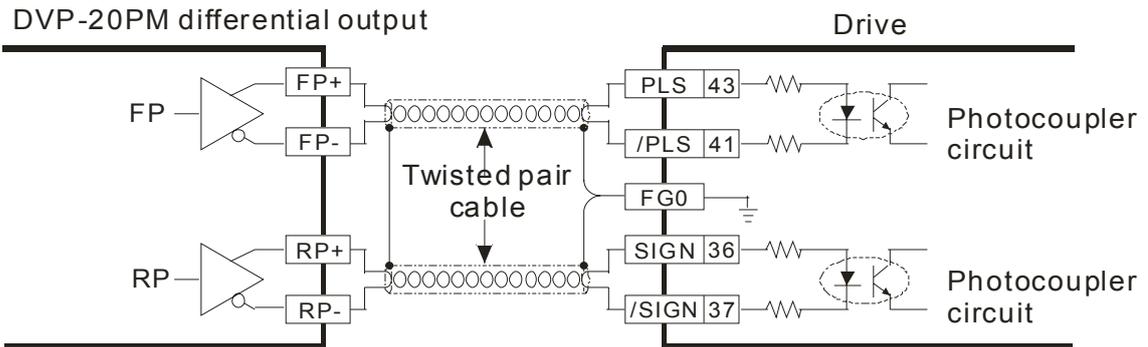


DVP-10PM differential output



2 Hardware Specifications and Wiring

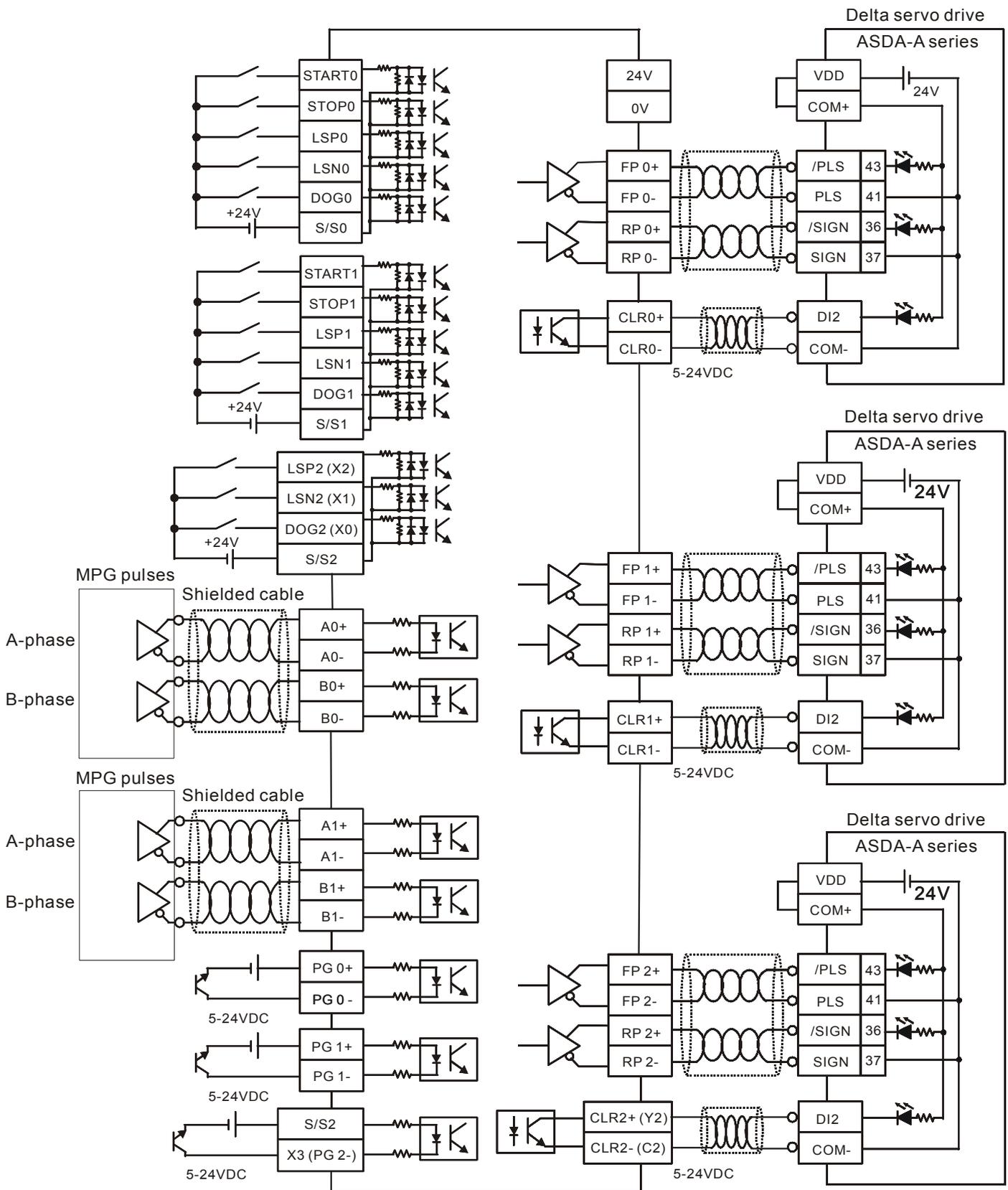
- Wiring of DVP-PM differential output with ASDA-AB series servo drive



2 Hardware Specifications and Wiring

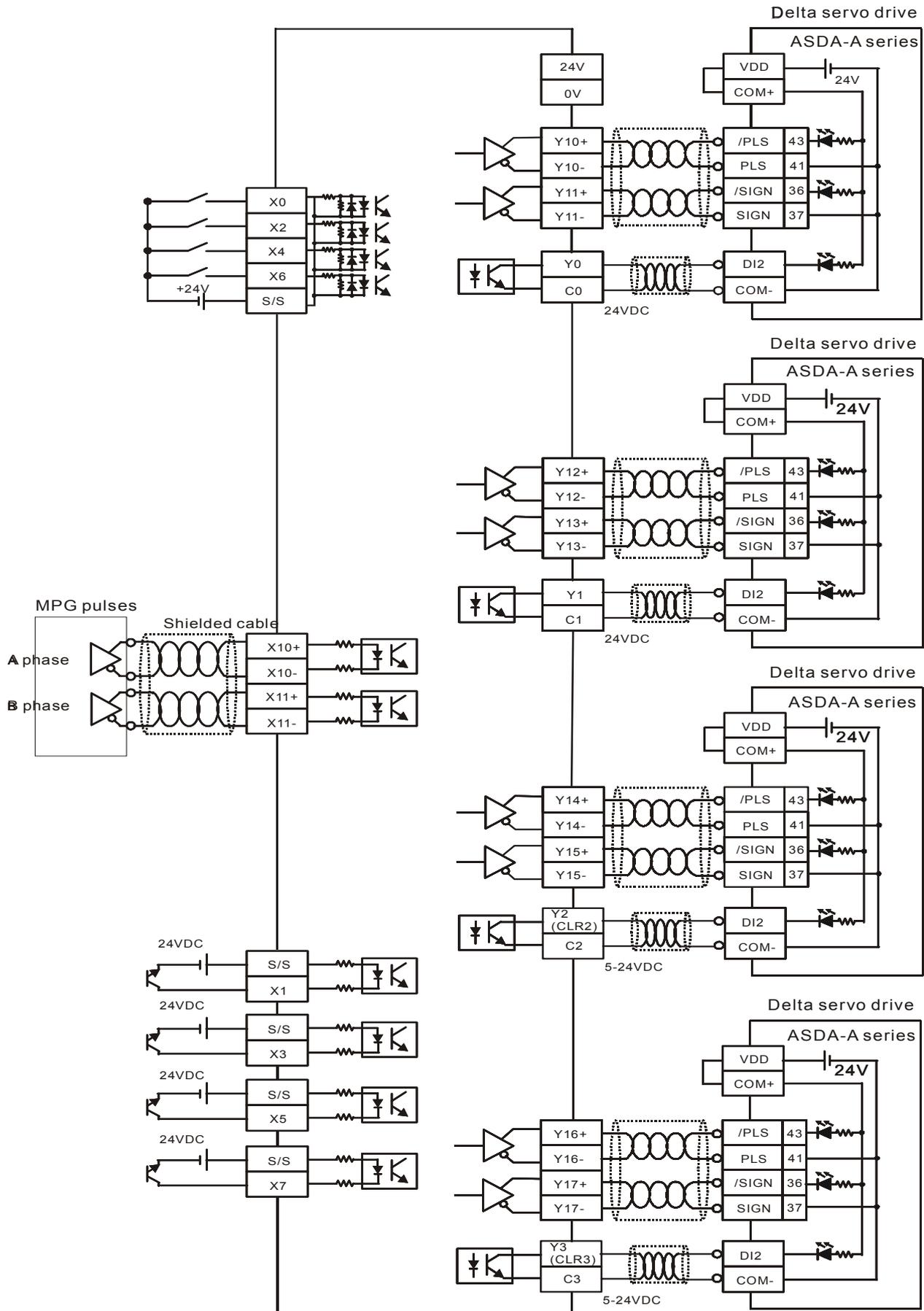
2.2.5 Wiring with Drives

Wiring of DVP-20PM with Delta ASDA-A series servo drive:



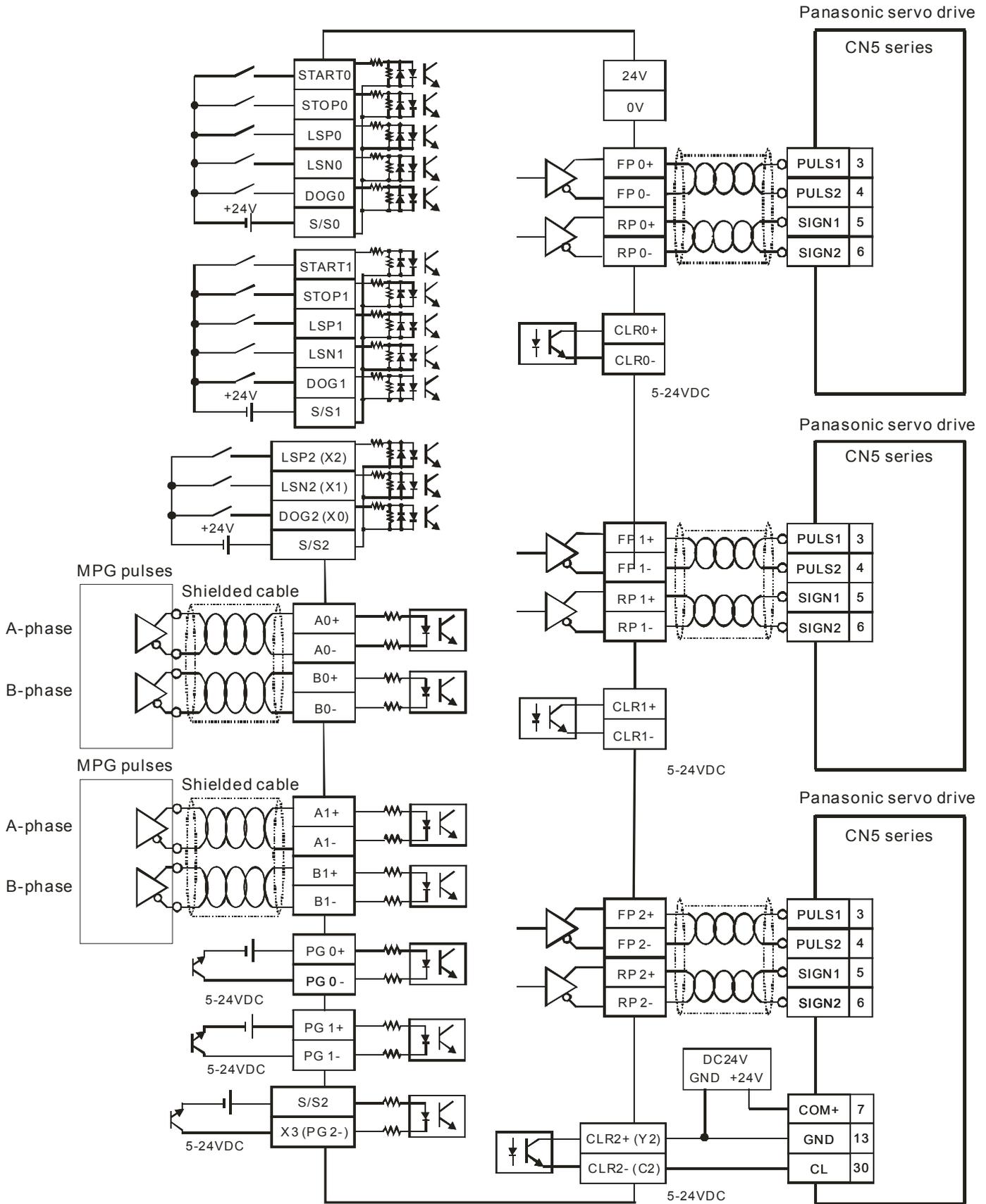
2 Hardware Specifications and Wiring

Wiring of DVP-10PM with Delta ASDA-A series servo drive:



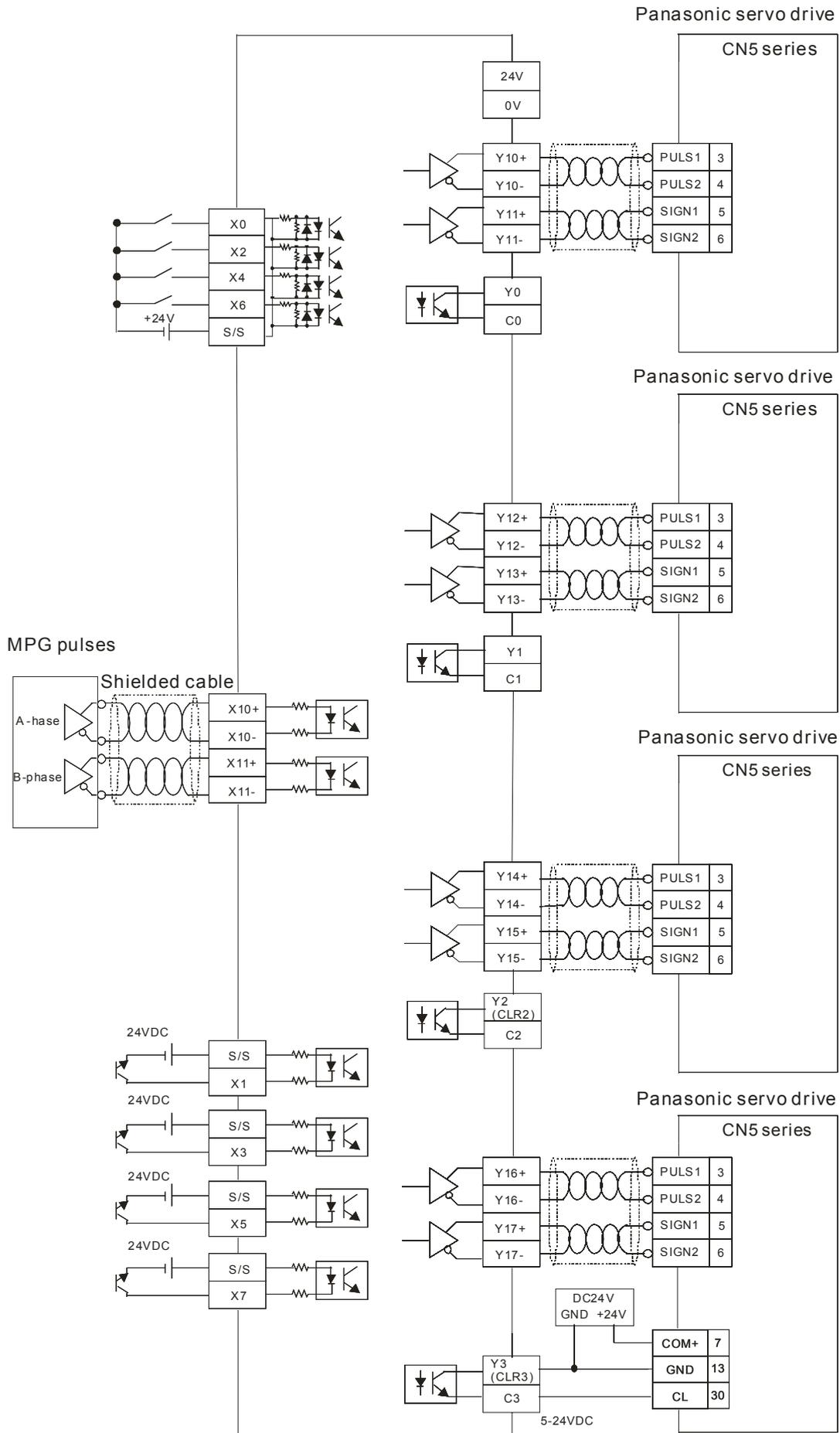
2 Hardware Specifications and Wiring

Wiring of DVP-20PM with Panasonic CN5 series servo drive:



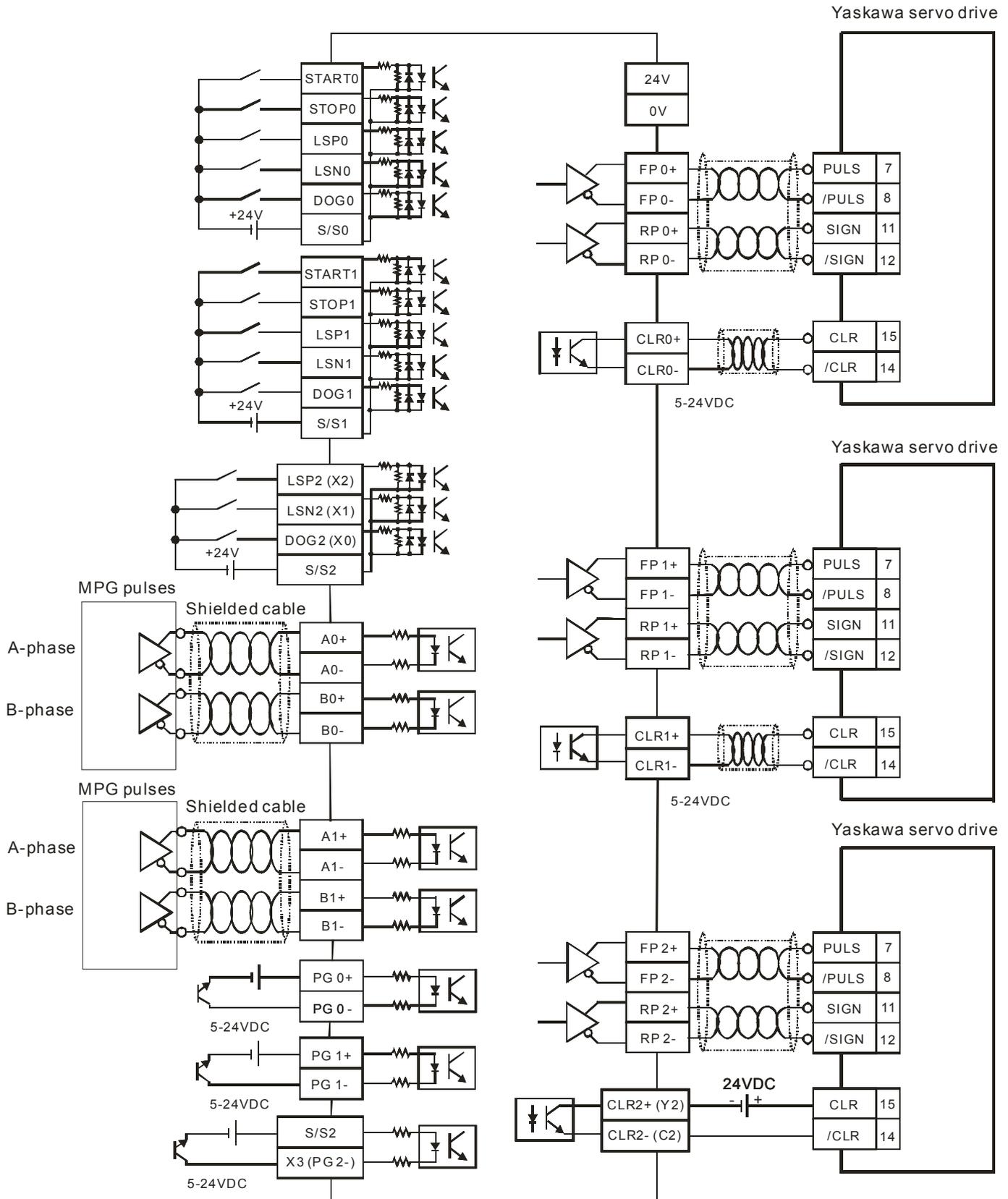
2 Hardware Specifications and Wiring

Wiring of DVP-10PM with Panasonic CN5 series servo drive:



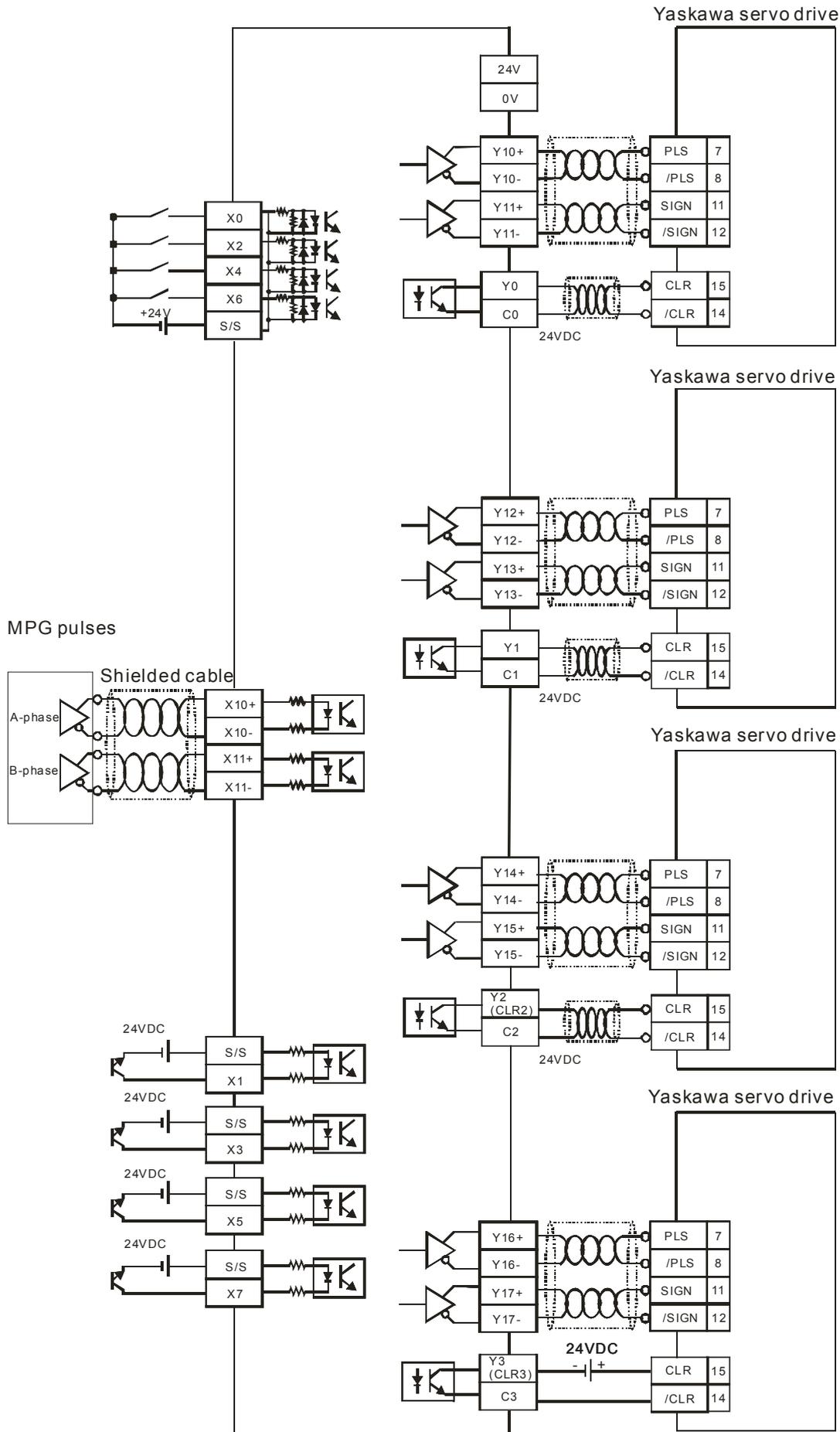
2 Hardware Specifications and Wiring

Wiring of DVP-20PM with Yaskawa servo drive:



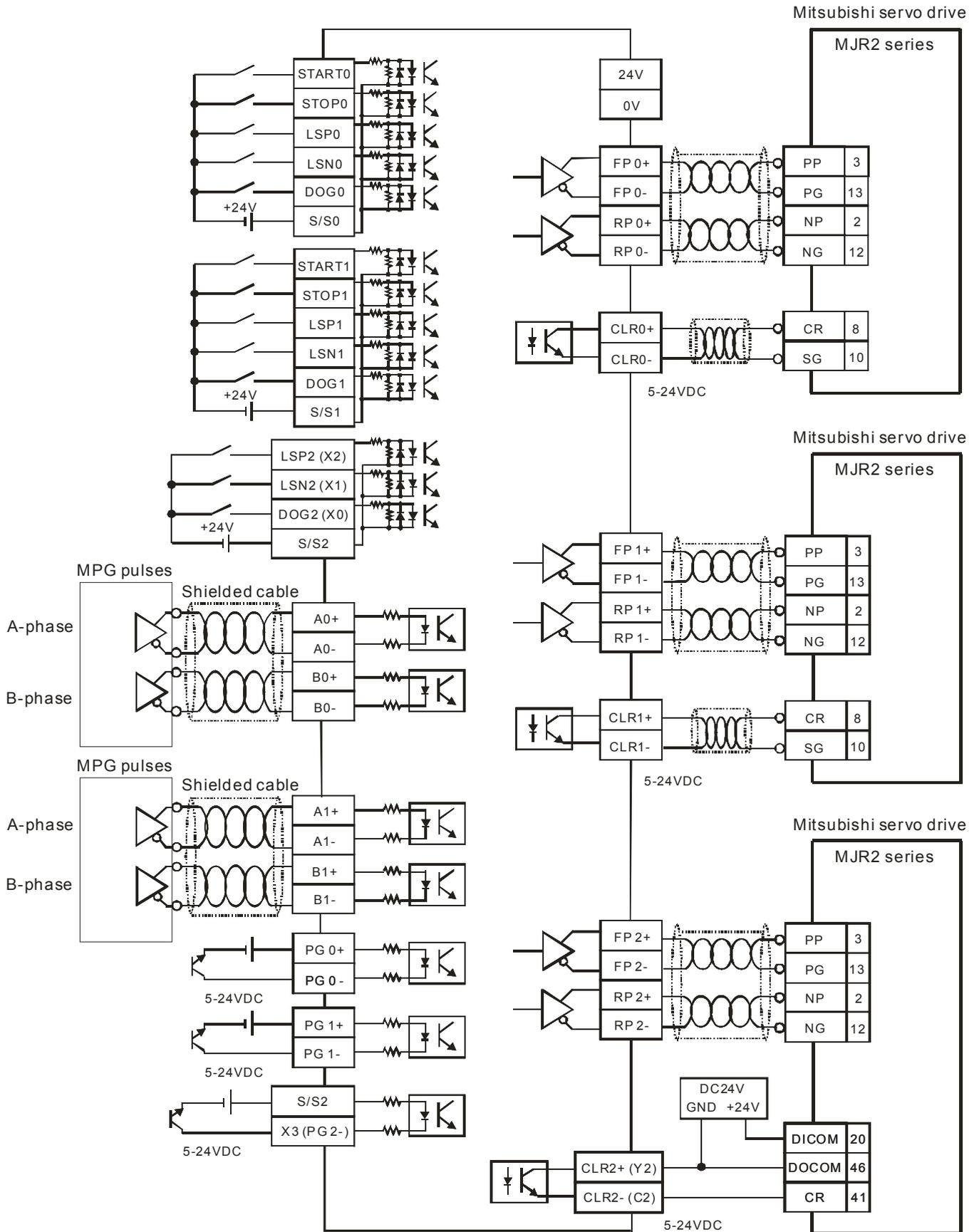
2 Hardware Specifications and Wiring

Wiring of DVP-10PM with Yaskawa servo drive:



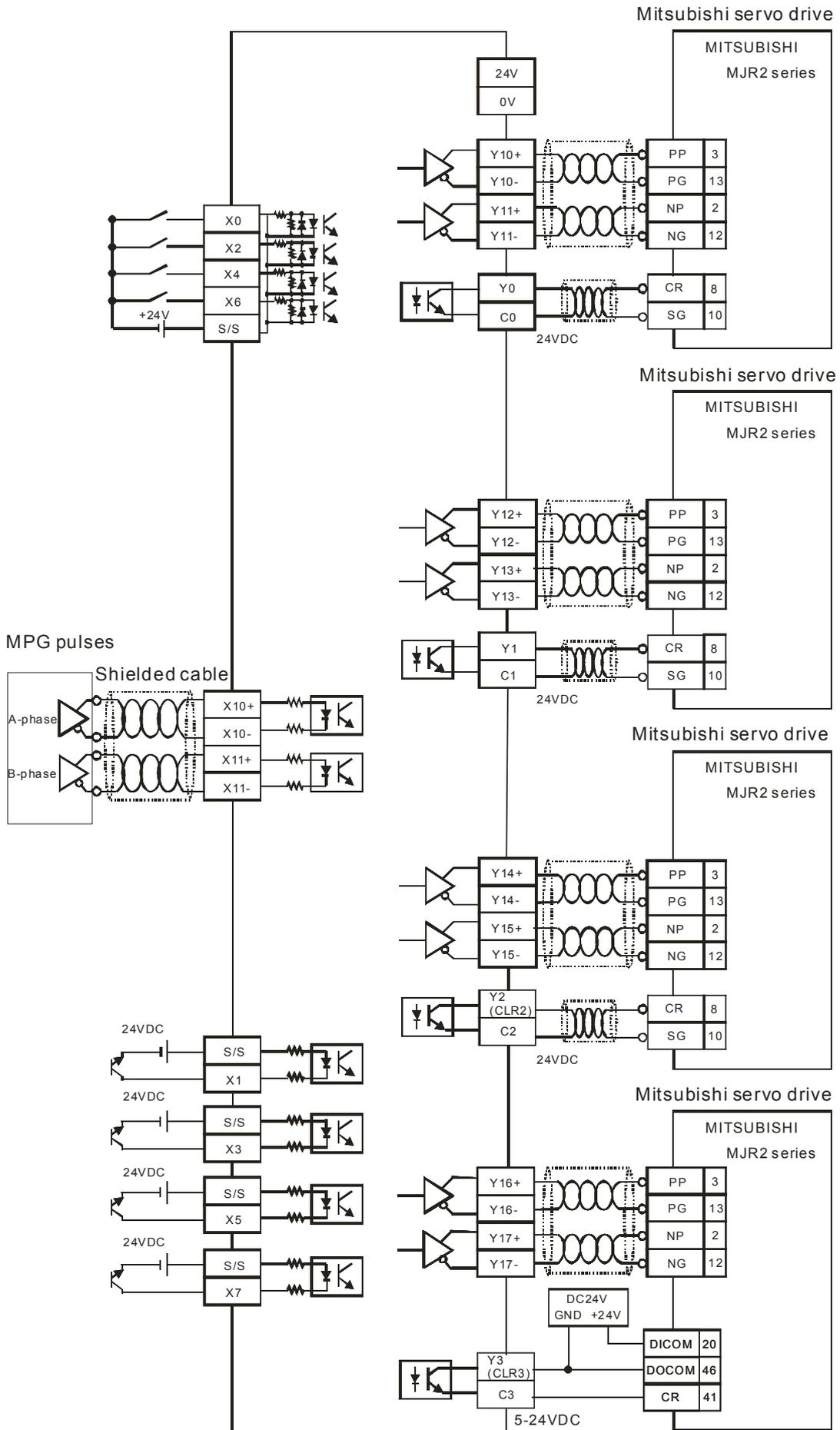
2 Hardware Specifications and Wiring

Wiring of DVP-20PM with Mitsubishi MJR2 series servo drive:



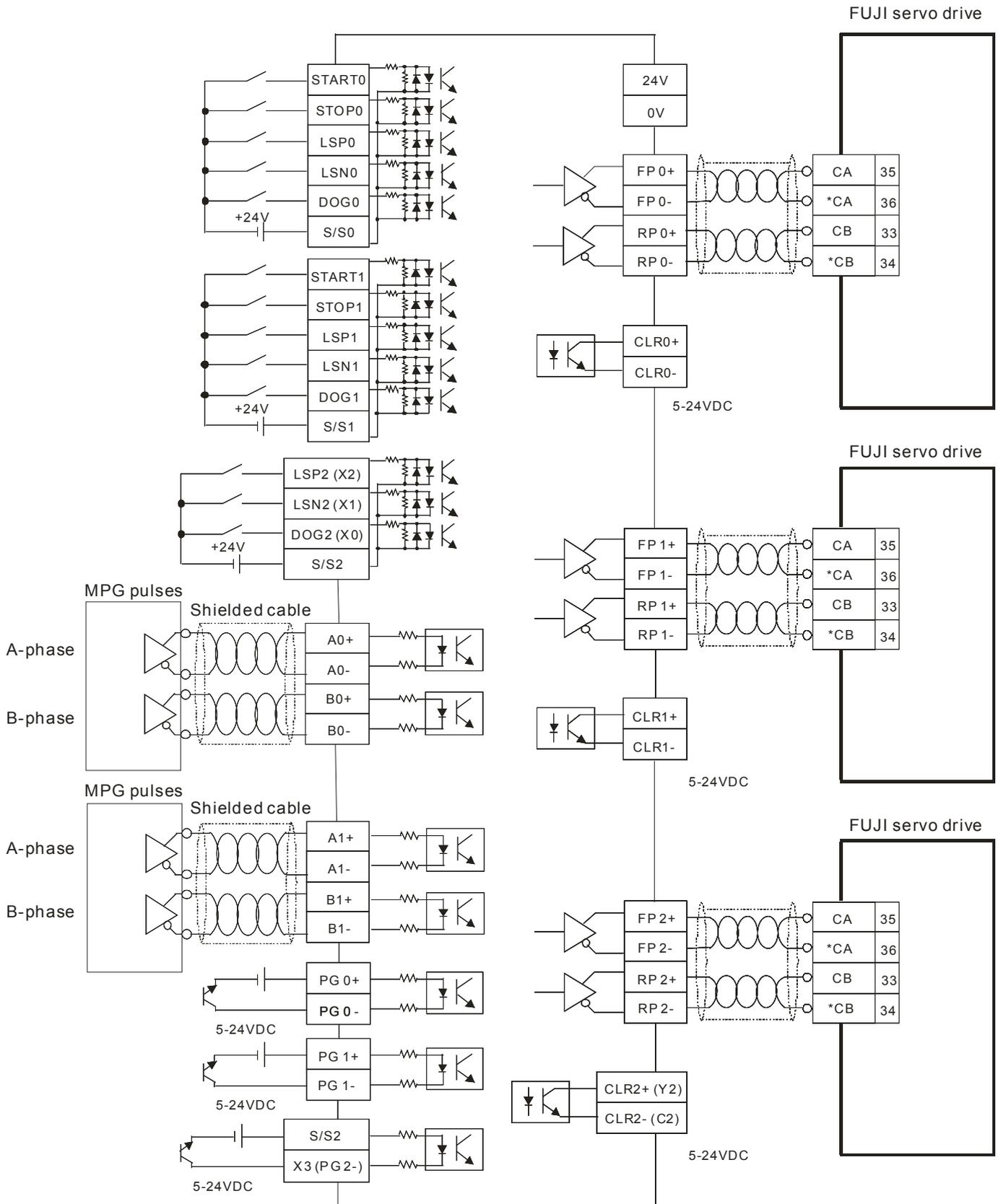
2 Hardware Specifications and Wiring

Wiring of DVP-10PM with Mitsubishi MJR2 series servo drive:



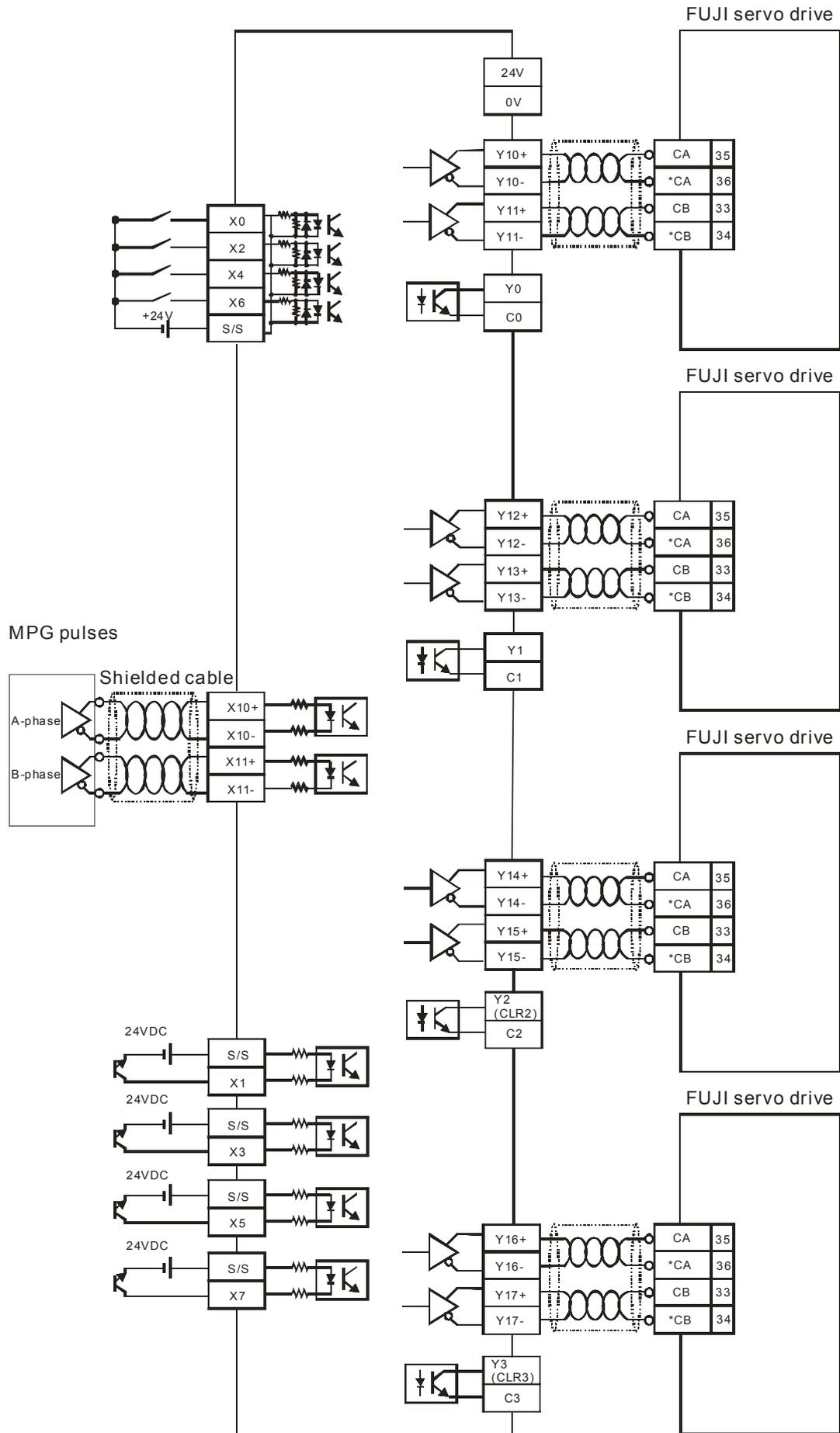
2 Hardware Specifications and Wiring

Wiring of DVP-20PM with FUJI servo drive:



2 Hardware Specifications and Wiring

Wiring of DVP-10PM with FUJI servo drive:



2 Hardware Specifications and Wiring

2.3 Communication Ports

DVP-PM offers two built-in communication ports, COM1 (RS-232 communication) COM2 (RS-485 communication) and one optional communication card, COM3 (RS-232/RS485.communication)

COM1: RS-232 communication port. Main communication port for programming. Can be used as Slave in Modbus ASCII or RTU mode.

COM2: RS-485 communication port. Can be used as Slave / Master in Modbus ASCII or RTU mode.

COM3: RS-232/RS-485 communication port. Can be used as Slave in Modbus ASCII mode.

Communication Format:

COM port	RS-232 (COM1)	RS-485 (COM2)	RS-232/RS-485 (COM3)
Parameter			
Baud rate	110~115,200 bps		110~38,400 bps
Data length	7~8bits		
Parity	Even/Odd/None parity check		
Length of stop bit	1~2 bits		
Register for Setting	D1036	D1120	D1109
ASCII mode	Available for Slave	Available for Master/Slave	Available for Slave
RTU mode	Available for Slave	Available for Master/Slave	-
Data length for access (ASCII)	100 registers		32 registers
Data length for access (RTU)	100 registers		32 registers

Default communication settings:

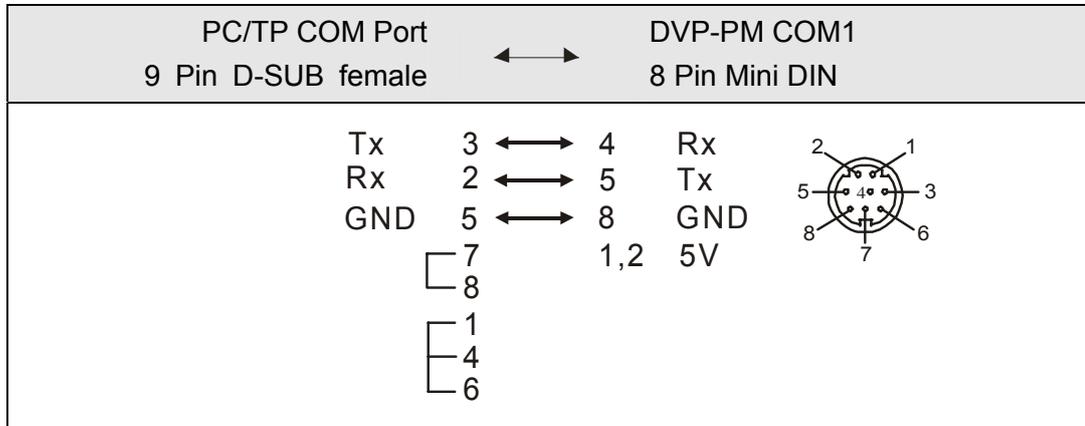
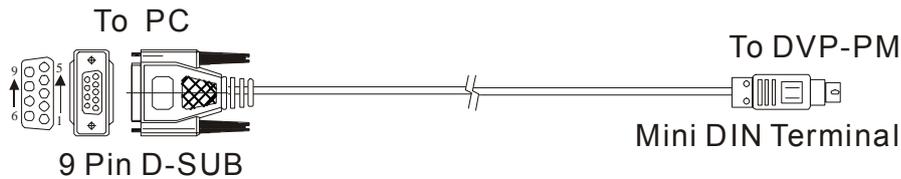
- Modbus ASCII
- 7 bits
- 1 stop bit
- EVEN parity
- Baud rate: 9600bps

2.3.1 COM1 (RS-232)

1. COM1 supports RS-232 data transmission which uploads and downloads PLC program. The communication format is Modbus (ASCII/RTU mode) and the baud rate is 9,600 ~ 115, 200bps.

Communication cable(DVPACAB2A30):

2 Hardware Specifications and Wiring



(For detailed model names or specifications of accessories, please see the catalog of Delta PLCs or download the latest product information from Delta's website.)

- COM1 is used for Slave mode. Therefore, it can be connected to a HMI for monitoring purposes.

2.3.2 COM2 (RS-485)

- COM2 supports RS-485 communication which can be used on connecting Master with multiple Slaves. The communication format is Modbus (ASCII / RTU) and the baud rate is 9,600 ~ 115, 200bps.
- COM2 can work in Master or Slave mode. In Master mode, it can be connected to lower position machines such as Delta PLCs or motor drives (e.g. Delta servo drive, temperature controller, AC motor drive, and so on) for accessing data. In Slave mode, it can be connected to HMI (e.g. Delta's TP and DOP series HMI) for monitoring purposes.

2.3.3 COM3 (RS-232/RS-485)

- When the built-in COM ports, COM1 (RS-232) and COM2 (RS-485), can not fulfill the communication requirements, users can apply additional communication cards (DVP-F232S, DVP-F485S) as COM3. The common functions of the additional RS-232/RS-485 communication card are the same as COM1, however, the baud rate for COM3 only supports 9600/19200/38400 bps in ASCII mode.
- COM3 can only be used as Slave, therefore it can be connected with HMI (Human Machine Interface) for monitoring purpose.

2 *Hardware Specifications and Wiring*

MEMO

3 Functions of Devices in DVP-PM

3.1 Devices in DVP-PM

■ Function Specifications:

		Specifications			Note
		20PM		10PM	
Control system		2-axis synchronous linear/arc interpolation and independent 2-axis control (*5)		4-axis synchronous interpolation and 4-axis independent control	
Program storage		Built-in 64k steps storage device			
Control units		Motor system (pulse)	Combined system(pulse/length)	Machine system (length)	
Access method for I/O module		Access CR (Control register) of the I/O module by FROM/TO instructions. System uses 2 CRs for 32-bit data.			
Serial connection with MPU		When used as an I/O module, MPU accesses the built-in CR0 ~ CR199 (maps to D1500 ~ D1699) of DVP-PM			
Pulse output system		Differential output in 3 modes: Pulse/Dir, FP(CW) / RP(CCW), A/B phase			
Maximum speed		For single axis: 500K PPS. For interpolation axis: 500K PPS			
Input signal	Operation switch	AUTO/MANU (auto/manual switch), START, STOP		STOP/RUN (auto/manual switch)	
	Detector	DOG, LSP (positive limit), LSN (negative limit), PG0 (zero point)		X0, X2, X4, X6 (DOG signal) X1, X3, X5, X7 (Zero point signal)	
	General input point	X0 ~ X7, expandable up to 256 points (MPU + I/O module)			
Output signal	Servo output	FP (forward pulse), RP (reverse pulse), CLR (clear signal)		Y10+, Y10-, Y12+, Y12-, Y14+, Y14-, Y16+, Y16- (forward pulse) Y11+, Y11-, Y13+, Y13-, Y15+, Y15-, Y17+, Y17- (reverse pulse)	
	General output point	Y0 ~ Y7, expandable up to 256 points (MPU + I/O module)			
	COM port	COM1: RS-232 (Slave only); COM2: RS-485 (can be Master or Slave) COM3: Optional RS-232 / RS-485 communication card (Slave only)			
Special I/O module	Optional	Shares all right-side modules with DVP-EH2 including AD, DA, PT, TC, XA, PU, etc (maximum 8 modules expandable). The left side of MPU can connect to high-speed I/O modules (maximum 8 modules extendable).			Extended modules will not occupy any digital I/O points.
Basic instruction		27		27	
Application instruction		130		130	
Motion instruction		22			
M -Code		<ul style="list-style-type: none"> ● M02: End of OX0 ~ 99 motion subroutine ● M00 ~ M01, M03 ~ M101, M103 ~ M65535: used as pause of the program, can be applied according to actual needs. ● M102: End of O100 main program 			
G-Code		G0 (hi-speed positioning), G1 (linear interpolation), G2 (clockwise arc interpolation), G3 (counter-clockwise arc interpolation), G4 (pause), G17(X-Y plane), G18(X-Z plane), G19(YZ plane), G90 (absolute positioning), G91 (relative positioning)		N/A	
Self-diagnosis		Displaying errors such as parameter error, program error and external errors			

3 Functions of Devices in DVP-PM

Relay (bit)	X	External inputs		X0 ~ X377, octal number system, 256 points (corresponding to physical input points)		Total 512 points	
	Y	External outputs		Y0 ~ Y377, octal number system, 256 points (corresponding to physical output points)			
	M	Auxiliary relay	General	M0 ~ M499, 500 points (*2)			Total 4,096 points. Can be driven (ON/OFF) in program.
				M3000 ~ M4095, 1,096 points (*3)			
			Latched	M500 ~ M999, 500 points (*3)			
			Special	M1000 ~ M2999, 2,000 points (some are latched)			
	T	Timer	10ms	T0 ~ T255, 256 points (*2)		Total 256 points. The associated contact = ON when timer reaches preset value.	
	C	Counter	16-bit count up	C0 ~ C99, 100 points (*2)		Total 250 points. The associated contact = ON when counter reaches preset value	
				C100 ~ C199, 100 points (*3)			
			32-bit count up/down	C208 ~ C219, 12 points (*2)			
C220 ~ C255, 36 points (*3)							
	32-bit high speed	C200, C204, 2 points (*2)		C200, C204, C208, C212, C216, C220, 6 points			
S	Step point	General	S0 ~ S499, 500 points (*2)			Total 1,024 points. Can be driven (ON/OFF) in program.	
		Latched	S500 ~ S1023, 524 points (*3)				
Register (word)	T	Current value		T0 ~ T255, 16-bit timer. 256 words		The associated contact = ON when timer reaches preset value.	
	C	Current value		C0 ~ C199, 16-bit counter, 200 words		The associated contact = ON when counter reaches preset value.	
				C200 ~ C255, 32-bit counter, 56 words			
	D	Data register	General	D0 ~ D199, 200 words (*2)			Total 10,000 words. General storage for word data. V/Z can be used for indirect designation.
			Latched	D200 ~ D999, 800 words (*3)			
D3000 ~ D9999, 7,000 words (*3)							
Special			D1000 ~ D2999, 2,000 words (some are latched)				
	Index	V0 ~ V7 (16-bit), Z0 ~ Z7, 16 words (32-bit) (*1)					
Pointer	P	For CJ, CJN, CALL, JMP instructions		P0 ~ P255, 256 points		Position pointer of CJ, CJN, CALL and JMP	
Constant	K	Decimal		K-32,768 ~ K32,767 (16-bit operation)			
				K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)			

3 Functions of Devices in DVP-PM

	H	Hex	H0000 ~ HFFFF (16-bit operation); H00000000 ~ HFFFFFFF (32-bit operation)	
	F	Floating point	Displaying floating point value by 32-bit data complying with IEEE754 standard. $\pm 1.1755 \times 10^{-38} \sim \pm 3.4028 \times 10^{+38}$	

*1: Non-latched area cannot be modified.

*2: Non-latched area, can be modified as latched area by changing the parameter settings

*3: Latched area, can be modified as non-latched area by changing the parameter settings

*4: Latched area, cannot be modified

*5: DVP20PM00M supports 3-axis linear interpolation and helical interpolation

■ Settings of latched and non-latched memory devices:

M (Auxiliary relay)	General purpose			Special auxiliary relay
	M0 ~ M499	M500 ~ M999	M3000 ~ M4095	M1000 ~ M2999
	Default:: not latched	Default:: latched	Default:: not latched	Allocated in the range of "general purpose" M relays
	Start:: D1200 (K500)*1 End: D1201 (K999) *1			Some are latched and cannot be modified.
T (Timer)	10ms			
	T0 ~ T255			
	Default:: not latched			
	Start:: D1202 (K-1) *2 End: D1203 (K-1) *2			
C (Counter)	16-bit count up		32-bit count up/down	
	C0 ~ C99	C100 ~ C199	C200, C204 C208 ~ C219	C220 ~ C255
	Default:: not latched	Default:: latched	Default:: not latched	Default:: latched
	Start:: D1204 (K100) End: D1205 (K199)		Start:: D1206 (K220) End: D1207 (K255)	
S (Step relay)	Initial	General purpose		Latched
	S0 ~ S9	S10 ~ S499		S500 ~ S1023
	Default:: non-latched			Default:: latched
	Start:: D1208 (K500), End: D1209 (K1023)			
D (Register)	General purpose	Latched		Special register
	D0 ~ D999	D3000 ~ D9999		D1000 ~ D2999 Allocated in the range of "general purpose" registers
	Default:: not latched	Default:: latched		Some are latched and cannot be modified.
	Start:: D1210 (K200) *3 End: D1211 (K9999) *3			

*1: If you set D1200 = 0 and D1201 = 4095, DVP-PM will automatically ignore M1000 ~ M2999 and set M0 ~ M999 and M3000 ~ M4095 as latched area.

*2: K-1 refers to default = not latched.

*3: If you set D1210 = 0 and D1211 = 9999, DVP-PM will automatically ignore D1000 ~ D2999 and set D0 ~ D999 and D3000 ~ D9999 as latched areas.

3 Functions of Devices in DVP-PM

■ Memory status between ON/OFF operation

Memory type	Power OFF->ON	MANU->AUTO (STOP->RUN)	AUTO->MANU (RUN->STOP)	Clear all non-latched areas (M1031=ON)	Clear all latched area (M1032=ON)	Default
Non-latched	Cleared	Unchanged	Cleared when M1033 = OFF	Cleared	Unchanged	0
			Unchanged when M1033 = ON			
Latched	Unchanged			Unchanged	Cleared	0

3.2 Values, Constants [K] / [H], Floating Points [F]

Constant	K	Decimal	K-32,768 ~ K32,767 (16-bit operation) K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)
	H	Hexadecimal form	H0000 ~ HFFFF (16-bit operation) H00000000 ~ HFFFFFFFF (32-bit operation)
Floating point	F	32 bits	Displaying floating point value by 32-bit data complying with IEEE754 standard. $\pm 1.1755 \times 10^{-38} \sim \pm 3.4028 \times 10^{+38}$

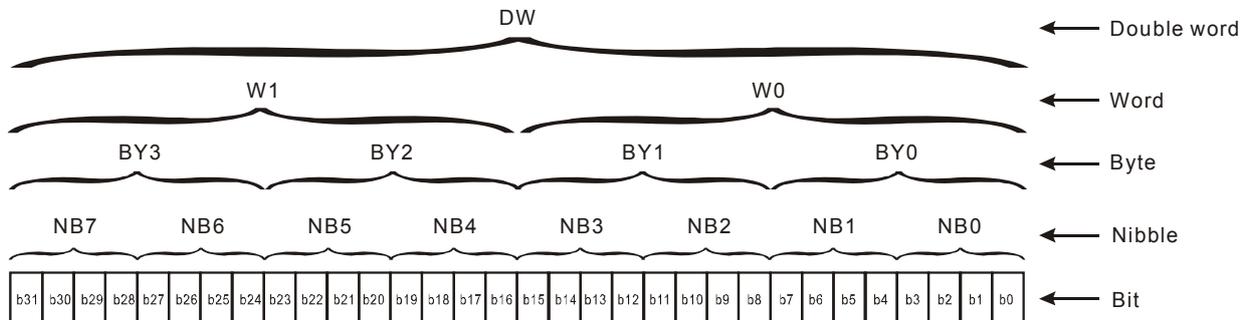
For fulfilling different kinds of internal manipulation, DVP-PLC applies 5 formats of number systems. Each number system has its specific purpose and function described as below.

1. Binary value (BIN)

DVP-PM internally calculates, operates, and stores the value in Binary format.

Bit	Bit is the basic unit of a binary number system. Range is 0 or 1
Nibble	Consists of 4 consecutive bits, e.g. b3~b0. Range 0 ~ 9 in Decimal or 0~F in Hex
Byte	Consists of 2 consecutive nibbles, e.g. b7~b0. Range 00 ~ FF in Hex
Word	Consists of 2 consecutive bytes, e.g. b15~b0. Range 0000 ~ FFFF in Hex
Double Word	Consists of 2 consecutive words, e.g. b31~b1. Range 00000000 - FFFFFFFF in Hex

Bit, nibble, byte, word and double word in a binary system:



2. Octal value (OCT)

The external I/O points of DVP-PLC are numbered in octal format.

e.g.

External inputs: X0~X7, X10~X17, ..., X377. (No. of device)

External outputs: Y0~Y7, Y10~Y17, ..., Y377. (No. of device)

3 Functions of Devices in DVP-PM

3. Decimal value (DEC)

DVP-PLC applies decimal operation in situations below:

- Set value for timers and counters, e.g. TMR C0 K50. (K value)
- No. of S, M, T, C, D, E, F, P, I devices, e.g. M10, T30. (No. of device)
- For use of operand in API instructions, e.g. MOV K123 D0. (K value)

4. Binary coded decimal (BCD)

BCD format takes 1 digit or 4 bits to indicate a Decimal value, so that data of consecutive 16 bits indicates a 4-digit decimal value. Used mainly for reading values from DIP switches or sending data to 7-segment displays.

5. Hexadecimal value (HEX)

Occasion of using hexadecimal values:

- For use of operand in API instructions, e.g. MOV H1A2B D0 (H value)

Constant K:

Decimal value in PLC operation is attached with an "K", e.g. K100 indicates the value 100 in Decimal format.

Exception:

When constant K is used with bit devices X, Y, M or S, the value specified after K indicates the groups of 4-bit unit, which forms a digit(4-bit), byte(8 bit), word(16bit), or double word(32-bit) data, e.g. K2Y10, K4M100, representing Y10 ~ Y17 and M100~M115.

Constant H:

Hexadecimal value in PLC operation is attached with an "H", e.g. H100 indicates the value 256 in Hex format.

Floating point F:

Floating point value in PLC operation is attached with an "F", e.g. F3.123 refers to a floating point value, 3.123. 23.

Reference table:

Binary (BIN)		Octal (OCT)	Decimal (DEC)	Binary Code Decimal (BCD)		Hexadecimal (HEX)
For DVP-PM internal operation		No. of device X, Y	Constant K and No. of device M, S, T, C, D, V, Z, P	For DIP switch and 7-segment display		Constant H
0 0 0 0	0 0 0 0	0	0	0 0 0 0	0 0 0 0	0
0 0 0 0	0 0 0 1	1	1	0 0 0 0	0 0 0 1	1
0 0 0 0	0 0 1 0	2	2	0 0 0 0	0 0 1 0	2
0 0 0 0	0 0 1 1	3	3	0 0 0 0	0 0 1 1	3
0 0 0 0	0 1 0 0	4	4	0 0 0 0	0 1 0 0	4
0 0 0 0	0 1 0 1	5	5	0 0 0 0	0 1 0 1	5
0 0 0 0	0 1 1 0	6	6	0 0 0 0	0 1 1 0	6
0 0 0 0	0 1 1 1	7	7	0 0 0 0	0 1 1 1	7
0 0 0 0	1 0 0 0	10	8	0 0 0 0	1 0 0 0	8
0 0 0 0	1 0 0 1	11	9	0 0 0 0	1 0 0 1	9
0 0 0 0	1 0 1 0	12	10	0 0 0 1	0 0 0 0	A
0 0 0 0	1 0 1 1	13	11	0 0 0 1	0 0 0 1	B

3 Functions of Devices in DVP-PM

Binary (BIN)		Octal (OCT)	Decimal (DEC)	Binary Code Decimal (BCD)		Hexadecimal (HEX)
For DVP-PM internal operation		No. of device X, Y	Constant K and No. of device M, S, T, C, D, V, Z, P	For DIP switch and 7-segment display		Constant H
0 0 0 0	1 1 0 0	14	12	0 0 0 1	0 0 1 0	C
0 0 0 0	1 1 0 1	15	13	0 0 0 1	0 0 1 1	D
0 0 0 0	1 1 1 0	16	14	0 0 0 1	0 1 0 0	E
0 0 0 0	1 1 1 1	17	15	0 0 0 1	0 1 0 1	F
0 0 0 1	0 0 0 0	20	16	0 0 0 1	0 1 1 0	10
0 0 0 1	0 0 0 1	21	17	0 0 0 1	0 1 1 1	11
:		:	:	:		:
0 1 1 0	0 0 1 1	143	99	1 0 0 1	1 0 0 1	63

3.3 Numbering and Function of External Input/Output Contacts [X] / [Y]

■ Input relay X0 ~ X377

The numbering of input relay (or input terminal) is in octal format. DVP-PM is designed for up to 256 points, and the range is: X0 ~ X7, X10 ~ X17, ...X370 ~ X377.

■ Output relay Y0 ~ Y377

The numbering of output relay (or output terminal) is in octal format. DVP-PM is designed for up to 256 points, and the range is: Y0 ~ Y7, Y10 ~ Y17, ...Y370 ~ Y377.

■ Function of input contact X

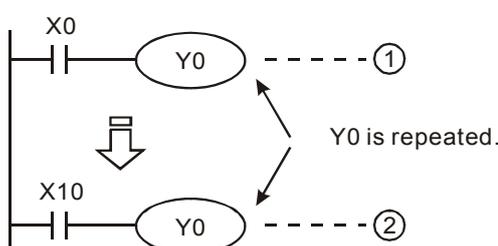
The input contact X is connected to input devices and reads the input signals into DVP-PM. There is no limitation on the times of using NO (Normally Open) / NC (Normally Closed) contacts in the program. ON/OFF of the input contact X only changes with ON/OFF status of the connected devices. PMSOft can not force ON/OFF the input contact X.

■ M1304, Force ON/OFF contact X

When M1304 = OFF, input contact can not be force ON/OFF by PMSOft; when M1304 = ON, input contact X can be force ON/OFF by PMSOft. However, the function of updating input signals will be disabled in this case.

■ Function of output contact Y

Output contact Y sends out ON/OFF signals to drive the load connected to output contacts. There are two types of output contacts, relay and transistor. There is no limitation on the times of using Y contacts (NO / NC) in the program. However, we suggest users do not repeat the No. of output coil Y; otherwise, according to the scan principle of ladder diagram, the status of output coil will be determined by the status of the last Y in the program.

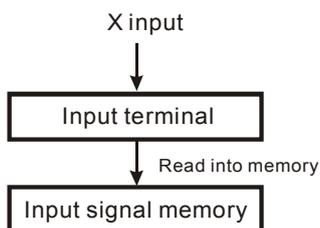


The output of Y0 will be determined by Y0 in ②, i.e. ON/OFF of X10 will determine the output status of Y0.

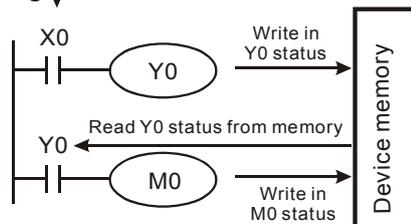
3 Functions of Devices in DVP-PM

The handling process of DVP-PM program:

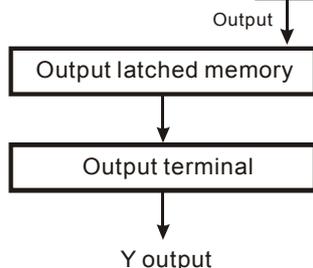
Update input signal



Program processing



Update output



Update input signal

1. Before the execution of the program, DVP-PM reads the ON/OFF status of the external input signals into the input signal memory.
2. The ON/OFF status of input signal changed during program execution will not be read into input signal memory. The new ON/OFF status will be read in the next scan.
3. There will be approximately a 10ms delay from external ON/OFF switching to the internal ON/OFF of contacts. The delay time may be affected by program scan time.

Program processing

After DVP-PM reads the ON/OFF status stored in the input signal memory, it executes every instruction from address 0 and stores the status of devices into the device memory.

Update output

1. When the execution reaches M102 instruction, DVP-PM sends the ON/OFF status of Y in the device memory to the output latched memory. The output latched memory is the coil of the output relay.
2. There will be a 10ms delay from the ON/OFF switching of output relay to the actual ON/OFF of physical output contacts.
3. For transistor output MPU, there will be a 10 ~ 20us delay from the ON/OFF switching to the actual ON/OFF of physical output contacts.

3.4 Numbering and Functions of Auxiliary Relays [M]

No. of auxiliary relays (in decimal):

Auxiliary relay M	General purpose	M0 ~ M499, 500 points, can be modified as latched area by setting up parameters.	Total 4,096 points
	Latched	M500 ~ M999, M3000 ~ M4095, 1,596 points, can be modified as non-latched area by setting up parameters.	
	Special purpose	M1000 ~ M2999, 2,000 points, some are latched.	

Functions of auxiliary relays:

Both auxiliary relay M and output relay Y have output coils and NO, NC contacts, and there is no limitation on the times of using the contacts. You can use auxiliary relay M in the program but it cannot directly drive the external load. There are three types of auxiliary relays.

3 Functions of Devices in DVP-PM

- General auxiliary relay:** If the relay encounters power OFF during the operation of DVP-PM, its status will be reset to OFF and remains OFF when the power is ON again.
- Latched auxiliary relay:** If the relay encounters power OFF during the operation of DVP-PM, its status will be retained and resumes the status when the power is ON again.
- Special auxiliary relay:** Every relay of this kind has its specific function. DO NOT use undefined special purpose auxiliary relay. See 3.10 for special relay and special registers and 3.11 for their functions.

3.5 Numbering and Functions of Step Relays [S]

■ No. of step relay (in decimal):

Step relay S	General purpose	S0 ~ S499, 490 points, can be modified as latched area by setting up parameters.	Total 1,024 points
	Latched	S500 ~ S1023, 524 points, can be modified as non-latched area by setting up parameters.	

■ Functions of step relays:

The device No. of S is S0 ~ S1023 (total 1,024 points). Both step relay S and output relay Y have output coils and NO, NC contacts, and there is no limitation on the times of using the contacts. You can use auxiliary relay M in the program but it cannot directly drive the external load. Step relay can be used as general auxiliary relays.

3.6 Numbering and Functions of Timers [T]

■ No. of timers (in decimal):

Timer T	10ms general purpose	T0 ~ T255, 256 points, can be modified as latched area by setting up parameters.	
------------	----------------------	--	--

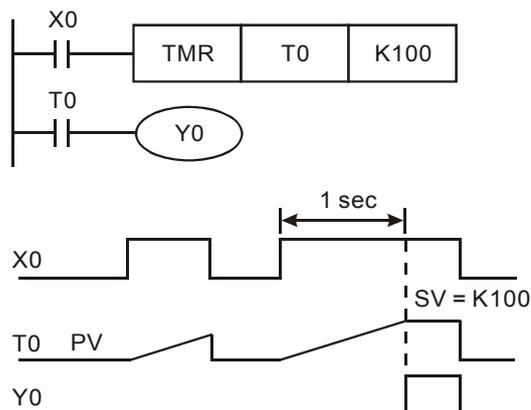
■ Functions of timers:

The timer resolution is 10ms and counts up when activated. The associated contacts of timer will be driven when the current value reaches the set value in timer. Set value can be K value (decimal) or D register.

The actual set time in the timer = timer resolution × set value

General purpose timer:

General timer counts once when TMR instruction is driven once. When TMR instruction is executed and the present value reaches the set value, the associated output coil will be ON.



- When X0 = ON, The PV in timer T0 will count up with timer resolution 10ms. When the PV = SV (K100), the output coil T0 will be ON.
- When X0 = OFF or the power is OFF, the PV in timer T0 will be cleared as 0, and the output coil will be OFF.

3 Functions of Devices in DVP-PM

How to designate SV: The actual set time in the timer = timer resolution × set value

1. Designating constant K: set a K value as SV directly
2. Designating D register: designate a data register in TMR instruction and the timer will take the value in the register as the SV.

3.7 Numbering and Functions of Counters [C]

■ **No. of counters (in decimal):**

20PM:

Counter C	16-bit count up	C0 ~ C199, 200 points	Total 248 points	The associated contact = ON when counter reaches preset value
	32-bit count up/down	C208 ~ C255, 48 points (accumulative)		
	32-bit high speed	C200, C204	Total 2 points	<ul style="list-style-type: none"> ● Input point of C200: MPGA0/MPGB0 ● Input point of C204: MPGA1/MPGB1

10PM:

Counter C	16-bit count up	C0 ~ C199, 200 points	Total 236 points	The associated contact = ON when counter reaches preset value
	32-bit count up/down	C220 ~ C255, 36 points (accumulative)		
	32-bit high speed	C200, C204, C208, C212, C216, C220. 6 points	Total 6 points	<ul style="list-style-type: none"> ● Input point of C200: X0/X1 ● Input point of C204: X2/X3 ● Input point of C208: X4/X5 ● Input point of C212: X6/X7 ● Input point of C216: X10+ / X10- / X11+ / X11- ● Input point of C220: X12+ / X12- / X13+ / X13-

■ **Features of counter:**

	16-bit counter	32-bit counter
Type	General purpose	General purpose
Counting direction	Count up	Count up, count down
Range of SV	0 ~ 32,767	-2,147,483,648 ~ +2,147,483,647
Designating SV	Constant K or data register D	Constant K or data register D (2 consecutive words)
Operation when SV is reached	Counting will stop after SV is reached.	Counting will continue after SV is reached.
Output contact	The associated contact = ON and latched when counter reaches SV.	The associated contact = ON and latched when counter reaches preset value in counting-up mode The associated contact is reset as OFF when counter reaches preset value in counting-down mode
Reset	When RST instruction is executed, PV of counter is cleared and the contact is OFF.	
Contact Action	Contacts act when the scan is completed.	Contacts act when the scan is completed.

3 Functions of Devices in DVP-PM

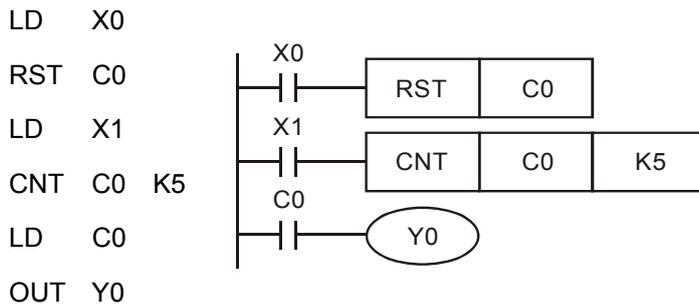
■ Functions of counters:

The counters count once when CNT instruction is driven once. When CNT instruction is executed and the PV reaches SV, the associated output coil will be ON. SV can be K value (decimal) or D register

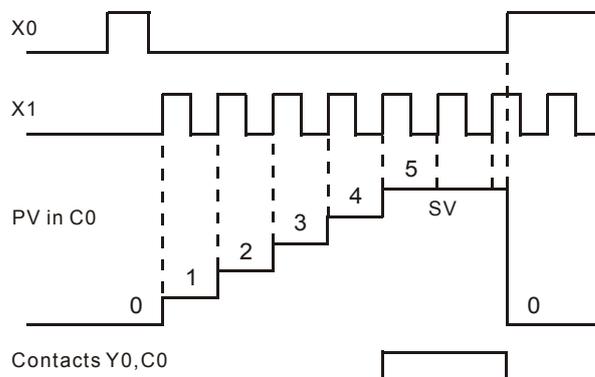
16-bit counters C0 ~ C199:

1. The set range of 16-bit counter: K0 ~ K32,767. SV=K0 is the same as SV=K1, i.e. the associated output contact will be ON immediately when counter is activated.
2. PV in the general purpose counter will be cleared when the power of DVP-PM is OFF. If the counter is a latched (accumulative) type, PV and the contact status will be retained, and PLC will resume counting after the power is ON again.
3. If you use MOV instruction or PMSOFT to send a value bigger than SV to the present value register of C0, when X1 goes from OFF to ON, the associated contact of counter C0 will be ON and its PV will equal SV.
4. SV can be K value (decimal) or D register (Special data register D1000~D2999 is not included).
5. Constant K should be a positive value for SV, however, values in data register D can be positive or negative. When PV reaches up to 32,767, the next PV will turn to -32,768.

Example:



1. When X0 = ON, RST instruction is executed, PV in C0 will be cleared and the associated contact will be reset.
2. When X1 goes from OFF to ON, PV in the counter will count up (plus 1).
3. When the counting of C0 reaches SV = K5, the contact of C0 will be ON. X1 input signal comes afterwards will not be accepted by C0, and PV of C0 will stay at K5.



32-bit general count-up/count-down counters: C208 ~ C255(20PM) / C220~C255(10PM)

1. The setup range of 32-bit counter: K-2,147,483,648 ~ K2,147,483,647. Counting mode (count-up/count-down) of the counter is designated by ON/OFF status of special auxiliary relay M1208 ~ M1255. For example, when M1255 = OFF, C255 will be a count-up counter; when M1255 = ON, C255 will be a count-down counter.
2. SV can be K value (decimal) or D register (Special data register D1000~D2999 is not included). If a D register is

3 Functions of Devices in DVP-PM

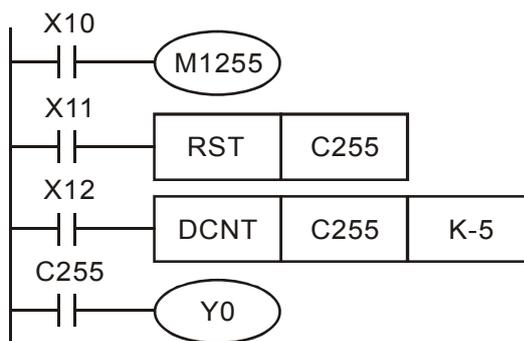
specified as SV, it will occupy 2 consecutive data registers.

- PV in the general purpose counter will be cleared when power of DVP-PM is OFF. If the counter is a latched (accumulative) type, PV and the contact status will be retained, and PLC will resume counting after power is ON again.
- When PV reaches up to 2,147,483,647, the next PV will turn to -2,147,483,648. When PV reaches down to -2,147,483,648, the next PV will turn to 2,147,483,647.

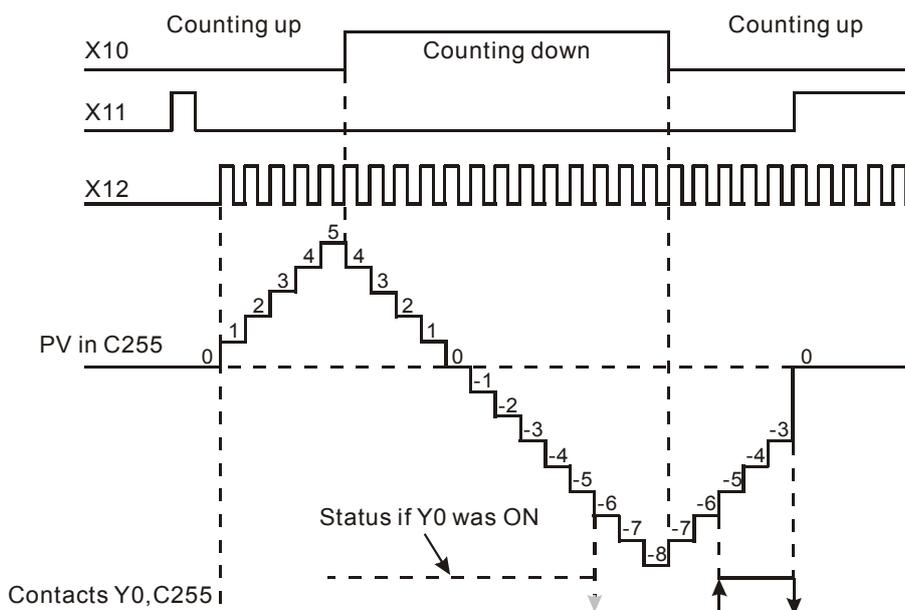
Example:

```

LD    X10
OUT   M1255
LD    X11
RST   C255
LD    X12
DCNT  C255 K-5
LD    C255
OUT   Y0
    
```



- X10 drives M1255 to determine whether C255 is a count-up or count-down counter.
- When X11 goes from OFF to ON, RST instruction will be executed, PV in C255 will be cleared, and the associated contact will be reset.
- When X12 goes from OFF to ON, PV in the counter will count up (plus 1) or down (minus 1).
- When PV in C255 changes from K-6 to K-5 ($PV \geq SV$), the contact of C255 will go from OFF to ON. When PV in C255 changes from K-5 to K-6 ($PV \leq SV$), the contact of C255 will go from ON to OFF.
- If you use MOV instruction or PMSoft to send a value bigger than SV to the present value register of C0, when X1 goes from OFF to ON, the associated contact of counter C0 will be ON and its PV will equal SV.



3 Functions of Devices in DVP-PM

32-bit High Speed Counter:

■ 20PM: C200, C204

1. The setup range of 32-bit high speed counter: K-2,147,483,648 ~ K2,147,483,647.
2. C200/C204 counting mode setting:

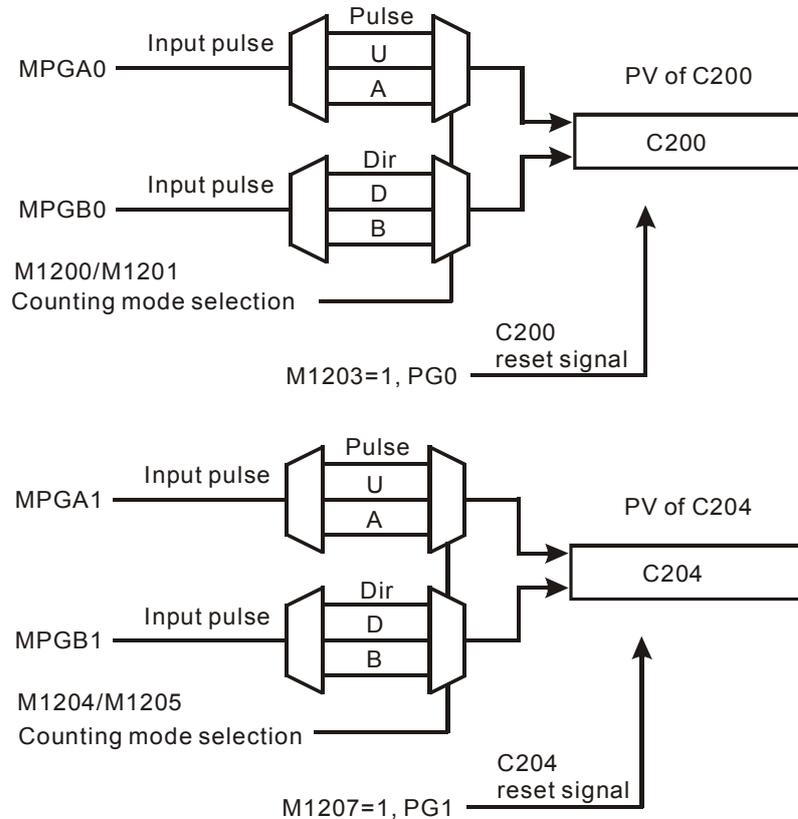
	Counter	Counting mode		Reset signal (Valid only when b3 = 1)	Input signal
		Device	b0 / b1		
1 st set	C200	K1M1200	0: U/D* 1: P/D* 2: A/B* (Single frequency)	PG0 (M1203)	A0+, A0-, B0+, B0-
2 nd set	C204	K1M1204	3:4A/B (Quadruple frequency)	X11- (M1207)	X2, X3, S/S

Note:

- U/D: Counting up / Counting down, P/D: Pulse / Direction, A/B: A-phase / B-phase

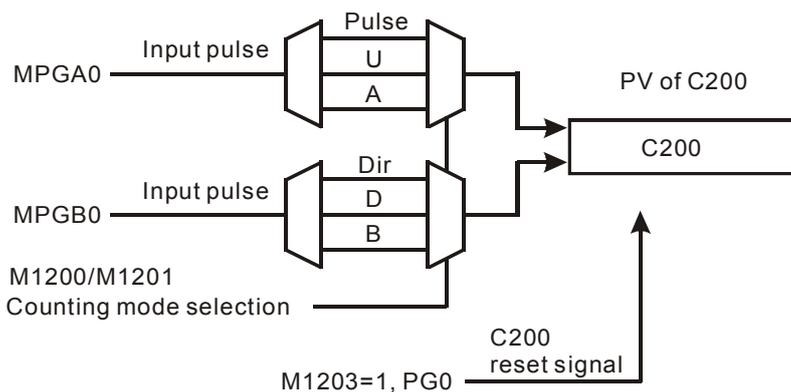
3. M1908: Input signal control of C200 / C204

When M1908 = OFF, the input signal of C200 is controlled by MPG A0/B0 and reset signal is controlled by PG0. In addition, Input signal of C204 is controlled by MPG A1/B1 and reset signal is controlled by PG1.



When M1908 = ON, the 3 axes X/Y/Z share the same high speed counter C200. The input signal of C200 is controlled by MPG A0/B0. C200 reset signal is controlled by PG0.

3 Functions of Devices in DVP-PM



4. SV can be K value (decimal) or D register (Special data register D1000~D2999 is not included). SV can be a positive or negative value. If a D register is specified as SV, it will occupy 2 consecutive data registers.
5. PV in the general purpose counter will be cleared when power of DVP-PM is OFF. If the counter is a latched (accumulative) type, PV and the contact status will be retained, and PLC will resume counting after power is ON again.
6. When PV reaches up to 2,147,483,647, the next PV will turn to -2,147,483,648. When PV reaches down to -2,147,483,648, the next PV will turn to 2,147,483,647.

■ 10PM: C200, C204, C208, C216, C220

1. The setup range of 32-bit high speed counter: K-2,147,483,648 ~ K2,147,483,647.
2. C200/C204/C208/C216/C220 counting mode settings:

	Counter	Counting mode		Reset signal	Reset terminal	Input signal
		Device	b0 / b1			
1 st set	C200	K1M1200	0: U/D*	M1203	X10	X0, X1, S/S
2 nd set	C204	K1M1204	1: P/D*	M1207	X11	X2, X3, S/S
3 rd set	C208	K1M1208	2: A/B* (Single frequency)	M1211	X12	X4, X5, S/S
4 th set	C212	K1M1212		M1215	X13	X6, X7, S/S
5 th set	C216	K1M1216	3:4A/B (Quadruple frequency)	M1219	X0	X10+, X10-, X11+, X11-
6 th set	C220	K1M1220		M1223	X1	X12+, X12-, X13+, X13-

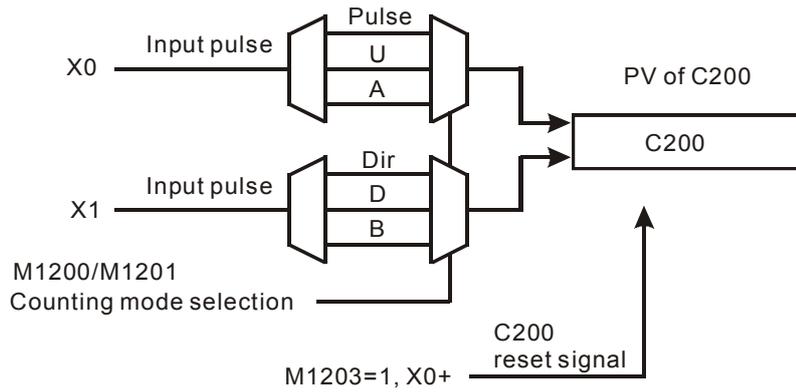
Note:

- U/D: Counting up / Counting down, P/D: Pulse / Direction, A/B: A-phase / B-phase

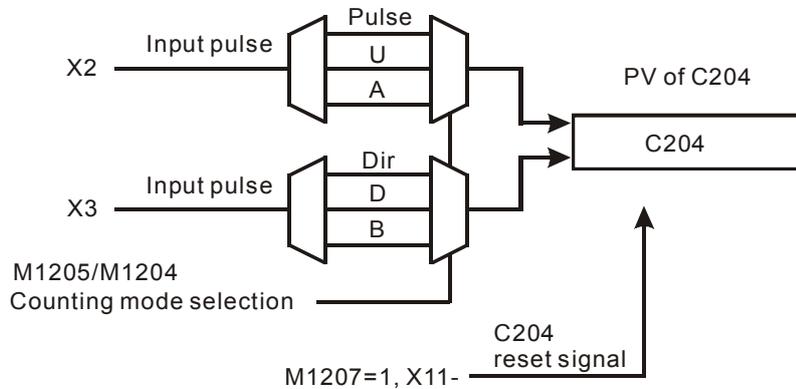
3. Operation diagram:

- Counting mode of C200 is selected by M1200/M1201. Input signal of C200 is controlled by X0/X1. Reset signal of C200 is enabled by M1203 and triggered by X10.

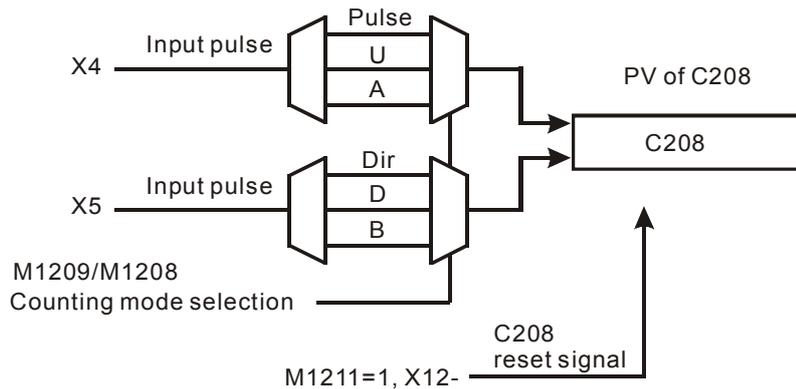
3 Functions of Devices in DVP-PM



- Counting mode of C204 is selected by M1204/M1205. Input signal of C204 is controlled by X2/X3. Reset signal of C204 is enabled by M1207 and triggered by X11.

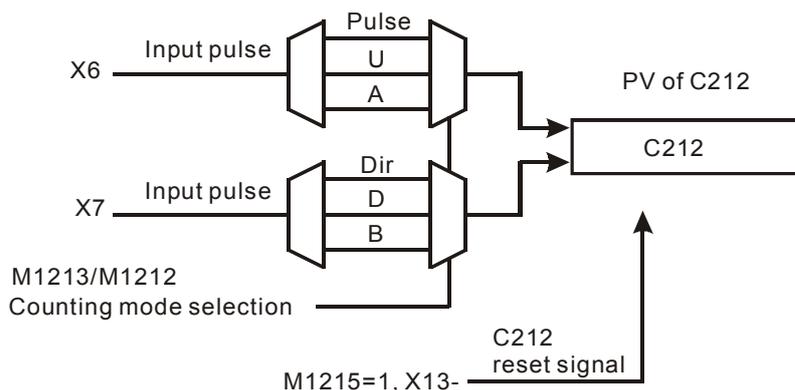


- Counting mode of C208 is selected by M1208/M1209. Input signal of C208 is controlled by X4/X5. Reset signal of C208 is enabled by M1211 and triggered by X12.

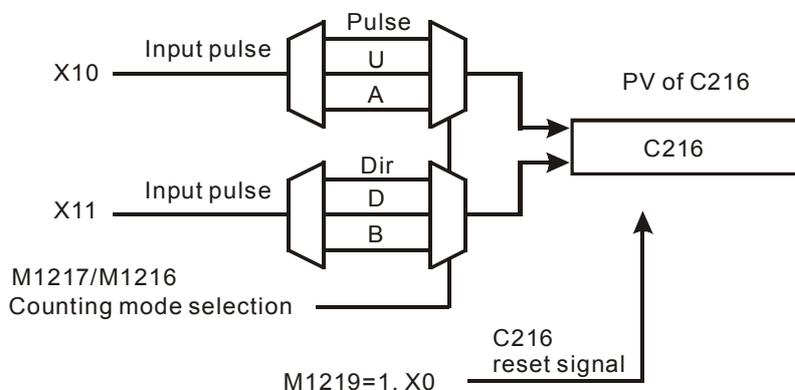


- Counting mode of C212 is selected by M1212/M1213. Input signal of C212 is controlled by X6/X7. Reset signal of C212 is enabled by M1215 and triggered by X13.

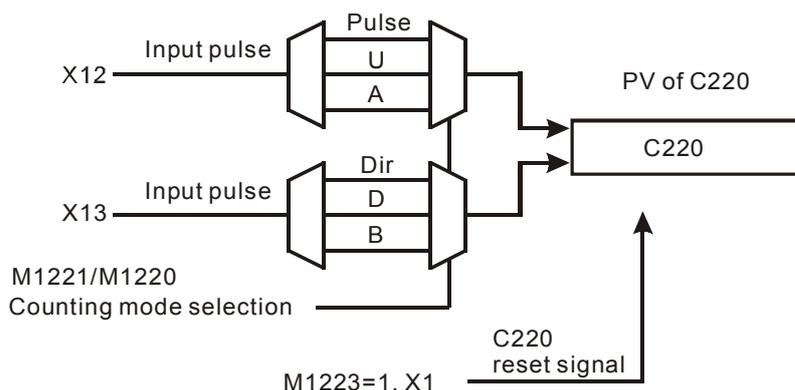
3 Functions of Devices in DVP-PM



- Counting mode of C216 is selected by M1216/M1217. Input signal of C216 is controlled by X10/X11. Reset signal of C216 is enabled by M1219 and triggered by X0.



- Counting mode of C220 is selected by M1220/M1221. Input signal of C220 is controlled by X12/X13. Reset signal of C220 is enabled by M1223 and triggered by X1.



- SV can be K value (decimal) or D register (Special data register D1000~D2999 is not included). SV can be a positive or negative value. If a D register is specified as SV, it will occupy 2 consecutive data registers.
- PV in the general purpose counter will be cleared when power of DVP-PM is OFF. If the counter is a latched (accumulative) type, PV and the contact status will be retained, and PLC will resume counting after power is ON again.
- When PV reaches up to 2,147,483,647, the next PV will turn to -2,147,483,648. When PV reaches down to -2,147,483,648, the next PV will turn to 2,147,483,647

3 Functions of Devices in DVP-PM

3.8 Numbering and Functions of Registers [D]

3.8.1 Data Register [D]

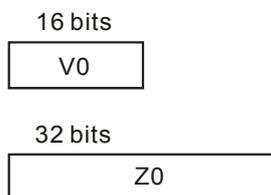
16-bit data register stores the value between -32,768 ~ +32,767. The MSB (Most Significant Bit) is also the sign bit indicating the value as positive “+” or negative “-“. Two 16-bit registers can be combined into a 32-bit register (D+1, D; register with smaller number is low word) storing the value between -2,147,483,648 ~ +2,147,483,647. Also, MSB is the sign bit.

Data register D	General purpose	D0 ~ D199, 200 points, can be modified into latched area by setting up parameters.	Total 10,000 points
	Latched	D200 ~ D999, D3000 ~ D9999, 7,800 points, can be modified into non-latched area by setting up parameters.	
	Special purpose	D1000 ~ D2999, 2,000 points, some are latched	
	Index register V, Z	V0 ~ V7, Z0 ~ Z7, 16 points	
File register		K0 ~ K9,999, MPU 10,000 points, fixed as latched	10,000 points

There are 4 types of registers:

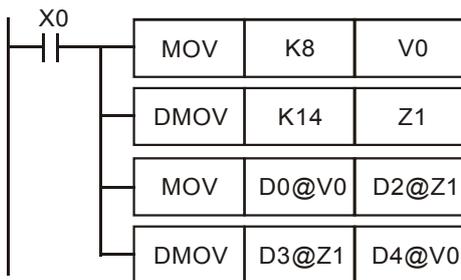
- General purpose register:** When DVP-PM goes from AUTO to MANU (STOP->RUN) or the power is switched off, the data in the register will be cleared to “0”. When M1033 = ON and DVP-PM goes from AUTO to MANU (STOP->RUN), the data will not be cleared but will still be cleared to “0” when the power is OFF.
- Latched register:** When the power of DVP-PM is switched off, the data in the register will not be cleared and will retain the value before the power is OFF. You can use RST or ZRST instruction to clear the data in the latched register.
- Special purpose register:** Every register of this kind has its special definition and purpose, mainly for storing the system status, error messages and monitoring status. See 3.10 and 3.11 for more details.
- Index register V, Z:** V is a 16-bit register, and Z is a 32-bit register. V0 ~ V7, Z0 ~ Z7, total 16 points.

3.8.2 Index Registers [V], [Z]



Register V is a 16-bit register and can be written and read as a general data register. V as a general register can only be used in 16-bit instructions.

Z is a 32-bit register. Z as a general data register can only be used in 32-bit instructions.



When X0 = ON, V0 = 8, Z1 = 14.

D0@V0 = D8, D2@Z1 = D16

The content in D8 will be moved to D16

The content in D17 will be moved to D12

V and Z for modifying the operands can be used in both 16-bit and 32-bit instructions (as the example).

The index register can be used as normal operands for moving or comparison on word devices (KnX, KnY, KnM, KnS, T, C, D) and bit devices (X, Y, M, S).

3 Functions of Devices in DVP-PM

V0 ~ V7, Z0 ~ Z7, total 16 point

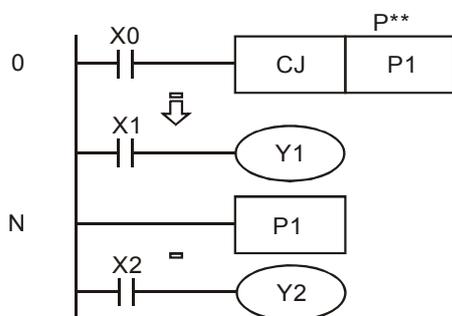
- Constant and some instructions do not support index registers. For how to use index register V, Z to modify the operands, see Ch5, 5.4 section for more details.
- When you use index register V, Z to modify the operands, the modification range CANNOT exceed the area of special purpose registers D1000 ~ D2999 and special auxiliary relays M1000 ~ M2999 in case errors may occur.

3.9 Pointer [N], Pointer [P_n]

Pointer	N	For nested level structure	N0 ~ N7, 8 points	Control point of nested level
	P	For CJ, CJK, JMP instructions	P0 ~ P255, 256 points	Position pointer of CJ, CJK, JMP

Pointer P: Used with API 00 CJ, API 256 CJK, and API 257 JMP. See Chapter 5 for explanations on CJ, CJK and JMP instructions for more details.

CJ conditional jump:



- When X0 = ON, the program will jump from address 0 to N (Pointer P1) and keep on the execution. The instructions between CJ instruction and P1 will be ignored.
- When X0 = OFF, the program will execute from address 0 and keep on executing. In this case, CJ instruction will not be executed.

3 Functions of Devices in DVP-PM

3.10 Special Auxiliary Relays [M], Special Data Register [D]

The types and functions of special auxiliary relays (special M) are listed in the table below. Special M and special D marked with "*" will be further illustrated in 3.11. Columns marked with "R" refers to "read only", "R/W" refers to "read and write", "-" refers to the status remains unchanged and "#" refers to that system will set it up according to the status of the PLC..

Special M	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
M1000*	Monitoring normally open contact:: ON when AUTO is activated	○	○	○	OFF	ON	OFF	R	NO	OFF	3-35
M1001*	Monitoring normally closed contact: OFF when AUTO is activated	○	○	○	ON	OFF	ON	R	NO	ON	3-35
M1002*	Enable single positive pulse at the moment when AUTO is activated (Normally OFF). Pulse width = scan time.	○	○	○	OFF	ON	OFF	R	NO	OFF	3-35
M1003*	Enable single negative pulse at the moment when AUTO is activated (Normally ON). Pulse width = scan time.	○	○	○	ON	OFF	ON	R	NO	ON	3-35
M1008	Watchdog timer (ON: PLC WDT time out)	○	○	○	OFF	OFF	-	R	NO	OFF	-
M1009	Indicate that LV status occurred.	○	○	○	OFF	-	-	R	NO	OFF	
M1011	10ms clock pulse, 5ms ON / 5ms OFF	○	○	○	OFF	-	-	R	NO	OFF	-
M1012	100ms clock pulse, 50ms ON / 50ms OFF	○	○	○	OFF	-	-	R	NO	OFF	-
M1013	1s clock pulse, 0.5s ON / 0.5s OFF	○	○	○	OFF	-	-	R	NO	OFF	-
M1014	1min clock pulse, 30s ON / 30s OFF	○	○	○	OFF	-	-	R	NO	OFF	-
M1025	Indicate incorrect request for communication (When PC or HMI is connected with DVP-PM, and DVP-PM receives illegal request for communication service during the data transmission, M1025 will be set, and the error code will be stored in D1025.)	○	○	○	OFF	OFF	-	R	NO	OFF	-
M1026	RAMP mode selection	○	○	○	OFF	-	-	R/W	NO	OFF	
M1029	CH0 (Y0, Y1) pulse output execution completed.	○	○	○	OFF	-	-	R	NO	OFF	
M1031	Clear all non-latched memory	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1032	Clear all latched memory	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1033	Memory latched when not in operation	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1034	Disable all Y outputs	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1035	Enable STOP0/START0 as external input points	○	○	○	OFF	OFF	OFF	R/W	NO	OFF	-
M1036*	Enable continuous interpolation on application instructions	○	○	○	OFF	OFF	OFF	R/W	NO	OFF	3-46
M1039*	Fix scan time	○	○	○	OFF	-	-	R/W	NO	OFF	3-39
M1048	Indicating alarm status	○	○	○	OFF	-	-	R	NO	OFF	-
M1049	Enabling alarm monitoring	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1072	PLC status (AUTO / MANU), ON = AUTO (communication)	○	○	○	OFF	ON	OFF	R/W	NO	OFF	-
M1074*	Enable OX motion subroutine	×	○	○	OFF	-	-	R/W	NO	OFF	3-39
M1077	Battery in low voltage, malfunction or no battery	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1087	Indicate current LV status	○	○	○	OFF	-	-	R/W	NO	OFF	-

3 Functions of Devices in DVP-PM

Special M	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
M1120*	Retain the communication setting of COM2 (RS-485). Modifying D1120 will be invalid when M1120 is set.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R/W	NO	OFF	3-36
M1121	For COM2(RS-485), data transmission ready	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	ON	-	R	NO	OFF	-
M1122	For COM2(RS-485), sending request	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R/W	NO	OFF	-
M1123	For COM2(RS-485), data receiving completed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R/W	NO	OFF	-
M1124	For COM2(RS-485), data receiving ready	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R	NO	OFF	-
M1125	For COM2(RS-485), communication ready status reset	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R/W	NO	OFF	-
M1127	For COM2(RS-485), data sending / receiving completed.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R/W	NO	OFF	-
M1128	For COM2(RS-485), Transmitting/Receiving status Indication	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R	NO	OFF	-
M1129	For COM2(RS-485), receiving time out	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R/W	NO	OFF	-
M1136*	For COM3(RS-485), retain communication setting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	3-36
M1138*	For COM1 (RS-232), retain communication setting. Modifying D1036 will be invalid when M1138 is set.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-36
M1139*	For COM1(RS-232), ASCII/RTU mode selection when in Slave mode (OFF: ASCII; ON: RTU)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-36
M1140	For COM2 (RS-485), MODRD / MODWR data receiving error	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R	NO	OFF	-
M1141	For COM2 (RS-485), MODRD / MODWR parameter error	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R	NO	OFF	-
M1143*	For COM2(RS-485), ASCII/RTU mode selection when in Slave mode (OFF: ASCII; ON: RTU) For COM2(RS-485), ASCII/RTU mode selection when in Master mode (used together with MODRD/MODWR instructions OFF: ASCII; ON: RTU)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-36
M1161	8/16 bit mode (ON = 8 bit mode)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1162	Switching between decimal integer and binary floating point for SCLP instruction ON: binary floating point; OFF: decimal integer	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1168	Designating work mode of SMOV	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	
M1200*	C200 counting mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-40
M1201	C200 counting mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1203	C200 reset signal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1204*	C204 counting mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-40
M1205	C204 counting mode	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1207	C204 reset signal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1208*	C208 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-40
M1209	C209 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1210	C210 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1211	C211 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1212*	C212 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-40
M1213	C213 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-

3 Functions of Devices in DVP-PM

Special M	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
M1214	C214 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1215	C215 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1216*	C216 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	3-40
M1217	C217 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1218	C218 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1219	C219 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1220*	C220 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	3-40
M1221	C221 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1222	C222 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1223	C223 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1224	C224 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1225	C225 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1226	C226 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1227	C227 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1228	C228 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1229	C229 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1230	C230 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1231	C231 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1232	C232 counter monitor (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1233	C233 counter monitor (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1234	C234 counter monitor (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1235	C235 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1236	C236 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1237	C237 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1238	C238 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1239	C239 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1240	C240 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1241	C241 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1242	C242 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1243	C243 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1244	C244 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1245	C245 counting mode (ON: count down)	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1246	C246 counting mode (ON: count down)	○	○	○	OFF	-	-	R	NO	OFF	-
M1247	C247 counting mode (ON: count down)	○	○	○	OFF	-	-	R	NO	OFF	-
M1248	C248 counting mode (ON: count down)	○	○	○	OFF	-	-	R	NO	OFF	-
M1249	C249 counting mode (ON: count down)	○	○	○	OFF	-	-	R	NO	OFF	-

3 Functions of Devices in DVP-PM

Special M	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
M1250	C250 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	-
M1251	C251 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	-
M1252	C252 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	-
M1253	C253 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	-
M1254	C254 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	-
M1255	C255 counting mode (ON: count down)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	-
M1303	High / low bits exchange for XCH instruction	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	
M1304*	Enable force-ON/OFF of input point X	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-43
M1744*	Reset M-code of OX subroutine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R/W	NO	OFF	3-44
M1745	Disable zero return of X axis in OX subroutine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1746	Select the start signal of acyclic CAM 0: START0, 1: PGO	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF			R/W	NO	OFF	
M1748	Indicate CAM completion	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF			R	NO	OFF	
M1749	CAM max frequency control	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1751*	Enable current position write-in function on X-axis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-44
M1752	Enable activation at non-zero position	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>							
	Enable acyclic E-CAM immediate stop function (Stop signal: DOG)	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1755	Switch the source of Master of acyclic CAM	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1756	Enable reference function of synchronized output. X0: CLR0=OFF signal. X1: CLR1=OFF signal	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1757	Remain current speed until meeting limit switch	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1760	Radian / degree selection in OX subroutine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-
M1761*	Stop X axis movement at fixed position	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-44
M1792*	X axis ready flag	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ON	ON	ON	R	NO	ON	3-44
M1793*	Error flag of motion subroutine on X axis; (automatically cleared when X axis is re-activated)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	3-45
M1794*	Indicate OX M-code is being executed (automatically cleared when OX is re-activated)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	OFF	R	NO	OFF	3-44
M1795	Indicate OX M0 code is being executed (automatically cleared when OX is re-activated)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	-
M1796	Indicate OX M2 code is being executed (automatically cleared when OX is re-activated)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	ON	-	R	NO	OFF	-
M1808	Zero flag for OX subroutine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	-
M1809	Borrow flag for OX subroutine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	-
M1810	Carry flag for OX suborutine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R	NO	OFF	-
M1811	Error flag for OX floating point operation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	OFF	-	R	NO	OFF	-
M1812	Indicate the completion of CAM initialization	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF			R/W	NO	OFF	
M1813	Indicate the completion of CAM	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF			R/W	NO	OFF	
M1825	Disable zero return of Y axis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	OFF	-	-	R/W	NO	OFF	-

3 Functions of Devices in DVP-PM

Special M	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
M1831*	Enable current position write in function on Y-axis	○	○	○	OFF	-	-	R/W	NO	OFF	3-44
M1841*	Stop Y axis movement at fixed position	×	○	○	OFF	-	-	R/W	NO	OFF	3-44
M1872*	Y axis ready flag	○	○	○	ON	ON	ON	R	NO	ON	3-44
M1873*	Error flag of motion subroutine on Y axis; (automatically cleared when Y axis is re-activated)	○	○	○	OFF	-	-	R	NO	OFF	3-45
M1908	Input signal control of C200/C204	×	○	○	OFF	-	-	R/W	NO	OFF	-
M1909	Virtual Master setting 1	×	○	○	OFF	-	-	R/W	NO	OFF	
M1910	Virtual Master setting 2	×	○	○	OFF	-	-	R/W	NO	OFF	
M1920	Radian / degree selection in O100	○	○	○	OFF	-	-	R/W	NO	OFF	-
M1952	O100 ready flag	○	○	○	ON	OFF	ON	R	NO	ON	-
M1953*	Error flag of O100	○	○	○	OFF	OFF	-	R/W	NO	OFF	3-50
M1957	Indicate AUTO / MANU (STOP / RUN) status	○	○	○	OFF	ON	-	R	NO	OFF	-
M1968	Zero flag for O100	○	○	○	OFF	-	-	R	NO	OFF	-
M1969	Borrow flag for O100	○	○	○	OFF	-	-	R	NO	OFF	-
M1970	Carry flag for O100	○	○	○	OFF	-	-	R	NO	OFF	-
M1971	Error flag for O100 floating point operation	○	○	○	OFF	-	-	R	NO	OFF	-
M1985	Disable zero return of Z axis	×	×	○	OFF	-	-	R/W	NO	OFF	-
M1991*	Enable current position write-in function on Z-axis	○	○	○	OFF	-	-	R/W	NO	OFF	3-44
M2001*	Stop Z axis movement at fixed position	○	○	○	OFF	-	-	R/W	NO	OFF	3-44
M2032*	Z axis ready flag	×	×	○	OFF	ON	ON	R	NO	ON	3-44
M2033*	Error flag of motion subroutine on Z axis; (automatically cleared when Y axis is re-activated)	×	×	○	OFF	-	-	R/W	NO	OFF	3-45
M2065	Disable zero return of A axis	○	×	×	OFF	-	-	R/W	NO	OFF	-
M2071*	Enable current position write-in function on A-axis	○	×	×	OFF	-	-	R/W	NO	OFF	3-44
M2112*	A axis ready flag	○	×	×	ON	ON	ON	R	NO	ON	3-44
M2113*	Error flag of motion subroutine on Z axis; (automatically cleared when Y axis is re-activated)	○	×	×	OFF	-	-	R/W	NO	OFF	3-45

Note: 10M = DVP10PM, 20D = DVP20PM00D, 20M = DVP20PM00M

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
D1000*	Scanning watch dog timer (Unit: 1ms)	○	○	○	200	-	-	R/W	NO	200	3-35
D1005	Display the firmware version of DVP-PM (initial factory setting)	○	○	○	#	-	-	R	NO	#	-
D1002	Program capacity	○	○	○	65,535	-	-	R	NO	65,535	-
D1003	Sum of the PLC internal program memory	○	○	○	-	-	-	R	YES	0	-
D1008	Step address when WDT is ON	○	○	○	0	-	-	R	NO	0	-
D1010	Current scan time (unit: 1ms)	○	○	○	0	-	-	R	NO	0	-
D1011	Minimum scan time (unit: 1ms)	○	○	○	0	-	-	R	NO	0	-
D1012	Maximum scan time (unit: 1ms)	○	○	○	0	-	-	R	NO	0	-
D1020*	X0 ~ X7 input filter (unit: ms)	○	○	○	10	-	-	R/W	NO	10	3-36
D1025	Error code of communication request error	○	○	○	0	0	-	R	NO	0	-
D1036*	COM1 (RS-232) Communication protocol	○	○	○	H'86	-	-	R/W	NO	H'86	3-36
D1038*	Delay time setting for data response when PLC is Slave in RS-485 communication. Range:0~3000 (unit: 10ms)	○	○	○	-	-	-	R/W	YES	0	3-39
D1039*	Fixed scan time (ms)	○	○	○	0	-	-	R/W	NO	0	3-39
D1040	The cycle time setting to synchronize with DVP-FPMC	○	○	○	0	-	-	R/W	NO	0	
D1050 ↓ D1055	Converted data for Modbus communication data processing. PLC automatically converts the ASCII data in D1070~D1085 into Hex data and stores the 16-bit Hex data into D1050~D1055	○	○	○	0	-	-	R	NO	0	-
D1056	CH0 present value of function card 2AD	○	○	○	0	#	-	R	NO	0	-
D1057	CH1 present value of function card 2AD	○	○	○	0	#	-	R	NO	0	-
D1070 ↓ D1085	Feedback data (ASCII) of Modbus communication. When PLC's RS-485 communication instruction receives feedback signals, the data will be saved in the registers D1070~D1085. Users can check the received data in these registers.	○	○	○	0	-	-	R	NO	0	-
D1089 ↓ D1099	The sent data of Modbus communication. When PLC's RS-485 communication instruction sends out data, the data will be stored in D1089~D1099. Users can check the sent data in these registers.	○	○	○	0	-	-	R	NO	0	-
D1109*	COM3 (RS-485/RS232) Communication protocol	○	○	○	H'86	-	-	R/W	NO	H'86	3-36
D1110	CH0 average value of function card 2AD	○	○	○	0	#	-	R	NO	0	-
D1111	CH1 average value of function card 2AD	○	○	○	0	#	-	R	NO	0	-
D1116	CH0 output value of function card 2DA	○	○	○	0	-	-	R/W	NO	0	-
D1117	CH1 output value of function card 2DA	○	○	○	0	-	-	R/W	NO	0	-
D1120*	COM2 (RS-485) communication protocol	○	○	○	H'86	-	-	R/W	NO	H'86	3-36
D1121	DVP-PM communication address (latched)	○	○	○	-	-	-	R/W	YES	1	-
D1122	COM2(RS-485) Residual number of words of transmitting data	○	○	○	0	0	-	R	NO	0	-
D1123	COM2(RS-485) Residual number of words of the	○	○	○	0	0	-	R	NO	0	-

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
	receiving data										
D1124	COM2(RS-485) Definition of start character (STX)	○	○	○	H'3A	-	-	R/W	NO	H'3A	
D1125	COM2(RS-485) Definition of first ending character (ETX1)	○	○	○	H'0D	-	-	R/W	NO	H'0D	
D1126	COM2(RS-485) Definition of second ending character (ETX2)	○	○	○	H'0A	-	-	R/W	NO	H'0A	
D1129	COM2 (RS-485) Communication time-out setting (ms)	○	○	○	0	-	-	R/W	NO	0	-
D1130	COM2 (RS-485) Error code returning from Modbus	○	○	○	0	0	-	R	NO	0	-
D1140*	Number of right-side modules (max. 8)	○	○	○	0	-	-	R	NO	0	3-42
D1142*	Number of input points (X) on DIO modules	○	○	○	0	-	-	R	NO	0	3-42
D1143*	Number of output points (Y) on DIO modules	○	○	○	0	-	-	R	NO	0	3-42
D1149	ID of function card (0: no card inserted; 3: COM3 card; 8: 2AD card; 9: 2DA card)	○	○	○	0	-	-	R	NO	0	-
D1200*	Start address of latched memory for auxiliary relays M	○	○	○	-	-	-	R/W	YES	500	3-42
D1201*	End address of latched memory for auxiliary relays M	○	○	○	-	-	-	R/W	YES	999	3-42
D1202*	Start address of latched memory for timer T	○	○	○	-	-	-	R/W	YES	-1	3-42
D1203*	End address of latched memory for timer T	○	○	○	-	-	-	R/W	YES	-1	3-42
D1204*	Start address of latched memory for 16-bit counter C	○	○	○	-	-	-	R/W	YES	100	3-42
D1205*	End address of latched memory for 16-bit counter C	○	○	○	-	-	-	R/W	YES	199	3-42
D1206*	Start address of latched memory for 32-bit counter C	○	○	○	-	-	-	R/W	YES	220	3-42
D1207*	End address of latched memory for 32-bit counter C	○	○	○	-	-	-	R/W	YES	255	3-42
D1208*	Start address of latched memory for step relay S	○	○	○	-	-	-	R/W	YES	500	3-42
D1209*	End address of latched memory for step relay S	○	○	○	-	-	-	R/W	YES	1,023	3-42
D1210*	Start address of latched memory for data register D	○	○	○	-	-	-	R/W	YES	200	3-42
D1211*	End address of latched memory for data register D	○	○	○	-	-	-	R/W	YES	9,999	3-42
D1320*	ID of the 1 st right-side module	○	○	○	0	-	-	R	NO	0	3-43
D1321*	ID of the 2 nd right-side module	○	○	○	0	-	-	R	NO	0	3-43
D1322*	ID of the 3 rd right-side module	○	○	○	0	-	-	R	NO	0	3-43
D1323*	ID of the 4 th right-side module	○	○	○	0	-	-	R	NO	0	3-43
D1324*	ID of the 5 th right-side module	○	○	○	0	-	-	R	NO	0	3-43
D1325*	ID of the 6 th right-side module	○	○	○	0	-	-	R	NO	0	3-43
D1326*	ID of the 7 th right-side module	○	○	○	0	-	-	R	NO	0	3-43
D1327*	ID of the 8 th right-side module	○	○	○	0	-	-	R	NO	0	3-43
D1328*	Target position (low word) of the third axis control by G-code (G00, G01)	×	×	○	0	-	-	R/W	NO	NO	6-46
D1329*	Target position (high word) of the third axis control by G-code (G00, G01)	×	×	○	0	-	-	R/W	NO	NO	6-46
D1330*	Max positioning speed (D1330 = K-1) of the third axis by G-code, G00 (low word)	×	×	○	0	-	-	R/W	NO	NO	6-46

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
	Interpolation speed of the third axis by G-code, G01. (low word)										6-48
D1331*	Max positioning speed (D1330 = K-1) of the third axis by G-code, G00 (high word)	×	×	○	0	-	-	R/W	NO	NO	6-46
	Interpolation speed of the third axis by G-code, G01. (high word)										6-48
D1400*	Enable interruption	○	○	○	0	-	-	R/W	NO	0	3-43
D1401*	Interrupt interval for timer interrupt (ms)	○	○	○	0	-	-	R/W	NO	0	3-43
D1500	FROM/TO data area, corresponding to CR#0	○	○	○	H'6260	-	-	R	NO	H'6260	-
D1501 ↓ D1699	FROM/TO data area, corresponding to CR#1 ~ CR#199	○	○	○	0	-	-	R/W	NO	0	-
D1700	The executing No. of OX	○	○	○	0	-	-	R	NO	0	-
D1702	Step No. of the executing OX subroutine	○	○	○	0	-	-	R	NO	0	-
D1703*	The executing M-code	○	○	○	0	-	-	R	NO	0	3-44
D1704	Pause time of OX subroutine	○	○	○	0	-	-	R	NO	0	-
D1705	Present pause time of OX	○	○	○	0	-	-	R	NO	0	-
D1706	Set value of RPT instruction in OX subroutine	○	○	○	0	-	-	R	NO	0	-
D1707	Present value of RPT instruction in OX subroutine	○	○	○	0	-	-	R	NO	0	-
D1708	Offset value for target position of X-axis (low word)	○	○	○	0	-	-	R	NO	0	-
D1709	Offset value for target position of X-axis (high word)										
D1710	Offset for distance to center of arc on X-axis (low word)	×	○	○	0	-	-	R	NO	0	-
D1711	Offset for distance to center of arc on X-axis (high word)										
D1712	Offset value of arc radius (low word)	×	○	○	0	-	-	R	NO	0	-
D1713	Offset value of arc radius (high word)										
D1724	Offset value for target position of Y-axis (low word)	×	○	○	0	-	-	R	NO	0	-
D1725	Offset value for target position of Y-axis (high word)										
D1726	Offset for distance to center of arc on Y-axis (low word)	×	○	○	0	-	-	R	NO	0	-
D1727	Offset for distance to center of arc on Y-axis (high word)										
D1736	Pause time (TIM) of O100	○	○	○	0	-	-	R	NO	0	-
D1737	Present pause time (TIM) of O100	○	○	○	0	-	-	R	NO	0	-
D1738	Set value of RPT instruction in O100	○	○	○	0	-	-	R	NO	0	-
D1739	Present value of RPT instruction in O100	○	○	○	0	-	-	R	NO	0	-
D1796*	Deceleration speed for continuous interpolation	○	○	○	0	-	-	R	NO	0	3-46
D1798*	Percentage value of G-Code moving speed	○	○	○	0	-	-	R	NO	0	3-47
D1799*	Input terminal polarity (for X/Y-axis)	○	○	○	0	-	-	R/W	NO	0	3-47
D1800*	Status of input terminal	○	○	○	0	-	-	R	NO	0	3-48

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
D1802*	Error code of O100	○	○	○	0	-	-	R/W	NO	0	3-50
D1803*	Error STEP of O100	○	○	○	0	0	-	R/W	NO	0	3-50
D1804*	Input terminal polarity 2 (for Z-axis)	×	×	○	0	-	-	R/W	NO	0	3-48
D1805*	Input terminal status 2 (for Z-axis)	×	×	○	0	-	-	R	NO	0	3-49
D1806*	High Byte: Filter coefficient of MPG0 / MPG1 Low Byte: Filter coefficient of other input points other than X points. Filter frequency: 85000/2N+4(kHz)	○	○	○	0	-	-	R/W	NO	0	3-49
D1816*	Parameter setting of X axis	○	○	○	-	-	-	R/W	YES	0	3-50
D1818	Pulses per revolution (PPR) on X axis (low word)	○	○	○	-	-	-	R/W	YES	2,000	-
D1819	Pulses per revolution (PPR) on X axis (high word)	○	○	○	-	-	-	R/W	YES	2,000	-
D1820	Displacement per revolution (DPR) of motor on X axis (low word)	○	○	○	-	-	-	R/W	YES	1,000	-
D1821	Displacement per revolution (DPR) of motor on X axis (high word)	○	○	○	-	-	-	R/W	YES	1,000	-
D1822	Maximum speed on X axis: V _{MAX} (low word)	○	○	○	-	-	-	R/W	YES	500K	-
D1823	Maximum speed on X axis: V _{MAX} (high word)	○	○	○	-	-	-	R/W	YES	500K	-
D1824	Bias speed of X axis: V _{BIAS} (low word)	○	○	○	-	-	-	R/W	YES	0	-
D1825	Bias speed of X axis: V _{BIAS} (high word)	○	○	○	-	-	-	R/W	YES	0	-
D1826	JOG speed of X axis: V _{JOG} (low word)	○	○	○	-	-	-	R/W	YES	5,000	-
D1827	JOG speed of X axis: V _{JOG} (high word)	○	○	○	-	-	-	R/W	YES	5,000	-
D1828	Zero return speed of X axis: V _{RT} (low word)	○	○	○	-	-	-	R/W	YES	50K	-
D1829	Zero return speed of X axis: V _{RT} (high word)	○	○	○	-	-	-	R/W	YES	50K	-
D1830	Zero return deceleration speed of X axis: V _{CR} (low word)	○	○	○	-	-	-	R/W	YES	1,000	-
D1831	Zero return deceleration speed of X axis: V _{CR} (high word)	○	○	○	-	-	-	R/W	YES	1,000	-
D1832*	Number of Zero point signals (PG0) on X axis:N	○	○	○	-	-	-	R/W	YES	0	3-51
	Times of acyclic CAM repetitions	×	○	○	-	-	-	R/W	YES	0	-
D1833*	Number of pulses for zero return on X axis: P	○	○	○	-	-	-	R/W	YES	0	3-51
	Distribution of remaining acyclic CAM pulses	×	○	○	-	-	-	R/W	YES	0	-
D1834	Definition of zero point on X axis: HP (low word)	○	○	○	-	-	-	R/W	YES	0	3-51
	Pulses for delay start of acyclic CAM	×	○	○	-	-	-	R/W	YES	0	-
D1835	Definition of zero point on X axis: HP (high word)	○	○	○	-	-	-	R/W	YES	0	-
	Pulses for delay start of CAM	×	○	○	-	-	-	R/W	YES	0	-
D1836	Acceleration time of X axis: T _{ACC}	○	○	○	-	-	-	R/W	YES	500	-
D1837	Deceleration time of X axis: T _{DEC}	○	○	○	-	-	-	R/W	YES	500	-
D1838	Target position (I) of X axis: P(I) (low word)	○	○	○	○	-	-	R/W	NO	0	-
	Lower bound of CAM synchronized output (low word)	×	○	○	○	-	-	R/W	NO	0	-

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
	PWM pulse width setting on X axis (low word)	○	×	×	○-	-	-	R/W	NO	0	-
D1839	Target position (I) of X axis: P(I) (high word)	○	○	○	○	-	-	R/W	NO	0	-
	Lower bound of CAM synchronized output (high word)	×	○	○	○-	-	-	R/W	NO	0	-
	PWM pulse width setting on X axis (Ihigh word)	○	×	×	○-	-	-	R/W	NO	0	-
D1840	Operation speed (I) of X axis: V(I) (low word)	○	○	○	1,000	-	-	R/W	NO	1,000	-
	CAM max frequency (low word)	×	○	○	1,000	-	-	R/W	NO	1,000	-
D1841	Operation speed (I) of X axis: V(I) (high word)	○	○	○	1,000	-	-	R/W	NO	1,000	-
	CAM max frequency (high word)	×	○	○	1,000	-	-	R/W	NO	1,000	-
D1842	Target position (II) of X axis: P(II) (low word)	○	○	○	0	-	-	R/W	NO	0	-
	Upper bound of CAM synchronized output (low word)	×	○	○	0	-	-	R/W	NO	0	-
	PWM cycle setting on X axis (low word)	○	×	×	0	-	-	R/W	NO	0	-
D1843	Target position (II) of X axis: P(II) (high word)	○	○	○	0	-	-	R/W	NO	0	-
	Upper bound of CAM synchronized output (high word)	×	○	○	0	-	-	R/W	NO	0	-
	PWM cycle setting on X axis (high word)	○	×	×	0	-	-	R/W	NO	0	-
D1844	Operation speed (II) of X axis: V(II) (low word)	○	○	○	2,000	-	-	R/W	NO	2,000	-
D1845	Operation speed(II) of X axis: V(II) (high word)	○	○	○	2,000	-	-	R/W	NO	2,000	-
D1846*	Operation commands for X axis	○	○	○	0	-	0	R/W	NO	0	3-53
	The enable setting of CAM (cyclic / acyclic)	○	○	○	0	-	0	R/W	NO	0	-
D1847*	Work modes for X axis	×	○	○	0	-	-	R/W	NO	0	3-53
D1848	Current position of X axis: CP (pulse) (low word)	○	○	○	0	-	-	R/W	NO	0	-
	Slave position (low word)	○	○	○	0	-	-	R/W	NO	0	-
D1849	Current position of X axis: CP (pulse) (high word)	○	○	○	0	-	-	R/W	NO	0	-
	Slave position (high word)	○	○	○	0	-	-	R/W	NO	0	-
D1850	Current speed of X axis: CS (pps) (low word)	○	○	○	0	0	0	R/W	NO	0	-
D1851	Current speed of X axis: CS (pps) (high word)	○	○	○	0	0	0	R/W	NO	0	-
D1852	Current position of X axis: CP (unit) (low word)	○	○	○	0	-	-	R/W	NO	0	-
D1853	Current position of X axis: CP (unit) (high word)	○	○	○	0	-	-	R/W	NO	0	-
D1854	Current speed of X axis: CS (unit) (low word)	○	○	○	0	0	0	R/W	NO	0	-
D1855	Current speed of X axis:CS (unit) (high word)	○	○	○	0	0	0	R/W	NO	0	-
D1856*	Execution status of X axis	○	○	○	0	-	-	R	NO	0	3-54
D1857*	Error code of X axis	○	○	○	0	-	-	R	NO	0	3-45
D1858	Numerator of electronic gear ratio of X axis	○	○	○	-	-	-	R/W	YES	1	-
D1859	Denominator of electronic gear ratio of X axis	○	○	○	-	-	-	R/W	YES	1	-
D1860	MPG input frequency on X axis (low word)	○	○	○	0	0	-	R/W	NO	0	-
	Master frequency (low word)	○	○	○	0	0	-	R/W	NO	0	-
D1861	MPG input frequency on X axis (high word)	○	○	○	0	0	-	R/W	NO	0	-

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
	Master frequency (high word)										
D1862	Number of accumulated MPG input pulses on X axis (low word)	○	○	○	0	-	-	R/W	NO	0	-
	Master position (low word)										
D1863	Number of accumulated MPG input pulses on X axis (high word)	○	○	○	0	-	-	R/W	NO	0	-
	Master position (high word)										
D1864	Response time of MPG input on X axis	○	○	○	-	-	-	R/W	YES	5	-
	Setting MPG input pulse pattern	×	○	○	-	-	-	R/W	YES	5	3-61
D1865	Stop mode for OX0 ~ 99. (K1 → complete remaining distance in next activation, K2 → execute the next instruction in next activation, Others → restart)	○	○	○	-	-	-	R/W	YES	0	-
D1866	Electrical zero point address on X axis (low word)	○	○	○	-	-	-	R/W	YES	0	-
	The position for switching the source of Master of acyclic CAM (low word)	×	○	○	-	-	-	R/W	YES	0	-
D1867	Electrical zero point address on X axis (high word)	○	○	○	-	-	-	R/W	YES	0	-
	The position for switching the source of Master of acyclic CAM (high word)	×	○	○	-	-	-	R/W	YES	0	-
D1868*	Setting up the No. of OX	○	○	○	0	0	-	R/W	YES	0	3-39
	Selecting CAM chart	×	○	○	0	0	-	R/W	YES	0	3-39
D1869	Error Step of OX	○	○	○	0	-	-	R/W	NO	0	-
D1872	Enabling Y output when OX is ready										
	High byte: K1; low byte: designating start No. of Y output	○	○	○	0	-	-	R/W	NO	0	-
D1873	Enabling Y output when OX executes M-code										
	High byte: K1; low byte: designating start No. of Y output	○	○	○	-	-	-	R/W	YES	0	-
D1874	No. of input point X for clearing OX M-code status	○	○	○	0	-	-	R/W	NO	0	-
D1875*	Enable External Input for X-Axis (ZRN, MPG, JOG-, JOG+)	○	○	○	-	-	-	R/W	YES	0	3-55
D1877*	Enable/Disalbe PID control on X axis	×	○	○	-	-	-	R/W	NO	0	3-55
D1878*	Set value for PID control on X axis (Low word)	×	○	○	-	-	-	R	NO	0	3-55
D1879	Set value for PID control on X axis (High word)	×	○	○	-	-	-	R	NO	0	3-55
D1880*	Proportional gain (K _P) value on X axis (Low word)	×	○	○	-	-	-	R/W	NO	0	3-55
D1881	Proportional gain (K _P) value on X axis (High word)	×	○	○	-	-	-	R/W	NO	0	3-55
D1882*	Intsgral gain (K _i) value on X axis (Low word)	×	○	○	-	-	-	R/W	NO	0	3-55
D1883	Intsgral gain (K _i) value on X axis (High word)	×	○	○	-	-	-	R/W	NO	0	3-55
D1884	Error value (difference) on X axis (Low word)	×	○	○	-	-	-	R	NO	0	-
D1885	Error value (difference) on X axis (High word)	×	○	○	-	-	-	R	NO	0	-
D1886	Accumulated error value on X axis (Low word)	×	○	○	-	-	-	R	NO	0	-

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
D1887	Accumulated error value on X axis (High word)	×	○	○	-	-	-	R	NO	0	-
D1888*	The max accumulated error value on X axis (Low word)	×	○	○	-	-	-	R	NO	0	3-55
D1889	The max accumulated error value on X axis (High word)	×	○	○	-	-	-	R	NO	0	3-55
D1896*	Parameter setting of Y axis	×	○	○	-	-	-	R/W	YES	0	3-50
D1898	Pulses per revolution (PPR) on Y axis (low word)	×	○	○	-	-	-	R/W	YES	2,000	-
D1899	Pulses per revolution (PPR) on Y axis (high word)										
D1900	Displacement per revolution (DPR) of motor on Y axis (low word)	○	○	○	-	-	-	R/W	YES	1,000	-
D1901	Displacement per revolution (DPR) of motor on Y axis (high word)										
D1902	Maximum speed on Y axis: V_{MAX} (low word)	○	○	○	-	-	-	R/W	YES	500K	-
D1903	Maximum speed on Y axis: V_{MAX} (high word)										
D1904	Bias speed of Y axis: V_{BIAS} (low word)	○	○	○	-	-	-	R/W	YES	0	-
D1905	Bias speed of Y axis: V_{BIAS} (high word)										
D1906	JOG speed of Y axis: V_{JOG} (low word)	○	○	○	-	-	-	R/W	YES	5,000	-
D1907	JOG speed of Y axis: V_{JOG} (high word)										
D1908	Zero return speed of Y axis: V_{RT} (low word)	○	○	○	-	-	-	R/W	YES	50K	-
D1909	Zero return speed of Y axis: V_{RT} (high word)										
D1910	Zero return deceleration speed of Y axis: V_{CR} (low word)	○	○	○	-	-	-	R/W	YES	1,000	-
D1911	Zero return deceleration speed of Y axis: V_{CR} (high word)										
D1912	Number of Zero point signals (PG0) on Y axis: N	○	○	○	-	-	-	R/W	YES	0	-
D1913	Number of pulses for zero return on Y axis: P	○	○	○	-	-	-	R/W	YES	0	-
D1914	Definition of zero point on Y axis: HP (low word)	○	○	○	-	-	-	R/W	YES	0	-
D1915	Definition of zero point on Y axis: HP (high word)										
D1916	Acceleration time of Y axis: T_{ACC}	○	○	○	-	-	-	R/W	YES	500	-
D1917	Deceleration time of Y axis: T_{DEC}	○	○	○	-	-	-	R/W	YES	500	-
D1918	Target position (I) of Y axis: P(I) (low word)	○	○	○	0	-	-	R/W	NO	0	-
	PWM pulse width setting on Y axis (low word)	○	×	×	0	-	-	R/W	NO	0	-
D1919	Target position (I) of Y axis: P(I) (high word)	○	○	○	0	-	-	R/W	NO	0	-
	PWM pulse width setting on Y axis (high word)	○	×	×	0	-	-	R/W	NO	0	-
D1920	Operation speed (I) of Y axis: V(I) (low word)	○	○	○	1,000	-	-	R/W	NO	1,000	-
D1921	Operation speed (I) of Y axis: V(I) (high word)										
D1922	Target position (II) of Y axis: P(II) (low word)	○	○	○	0	-	-	R/W	NO	0	-
	PWM cycle setting on Y axis (low word)	○	×	×	0	-	-	R/W	NO	0	-
D1923	Target position (II) of Y axis: P(II) (high word)	○	○	○	0	-	-	R/W	NO	0	-

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
	PWM cycle setting on Y axis (high word)	○	×	×	0	-	-	R/W	NO	0	-
D1924	Operation speed (II) of Y axis: V(II) (low word)	○	○	○	2,000	-	-	R/W	NO	2,000	-
D1925	Operation speed (II) of Y axis: V(II) (high word)	○	○	○							
D1926*	Operation commands for Y axis	○	○	○	0	-	0	R/W	NO	0	3-53
D1927*	Work modes for Y Axis	×	○	○	0	-	-	R/W	NO	0	3-53
D1928	Current position of Y axis: CP (pulse) (low word)	○	○	○	0	-	-	R/W	NO	0	-
D1929	Current position of Y axis: CP (pulse) (high word)	○	○	○							
D1930	Current speed of Y axis: CS (pps) (low word)	○	○	○	0	0	0	R/W	NO	0	-
D1931	Current speed of Y axis: CS (pps) (high word)	○	○	○							
D1932	Current position of Y axis: CP (unit) (low word)	○	○	○	0	-	-	R/W	NO	0	-
D1933	Current position of Y axis: CP (unit) (high word)	○	○	○							
D1934	Current speed of Y axis: CS (unit) (low word)	○	○	○	0	0	0	R/W	NO	0	-
D1935	Current speed of Y axis:CS (unit) (high word)	○	○	○							
D1936*	Execution status of Y axis	○	○	○	0	-	-	R	NO	0	3-54
D1937*	Error code of OX on Y axis	○	○	○	0	-	-	R	NO	0	3-45
D1938	Numerator of electronic gear ratio of Y axis	○	○	○	-	-	-	R/W	YES	1	-
D1939	Denominator of electronic gear ratio of Y axis	○	○	○	-	-	-	R/W	YES	1	-
D1940	MPG input frequency on Y axis (low word)	×	○	○	0	-	0	R/W	NO	0	-
D1941	MPG input frequency on Y axis (high word)	○	○	○							
D1942	Number of accumulated MPG input pulses on Y axis (low word)	○	○	○	0	-	-	R/W	NO	0	-
D1943	Number of accumulated MPG input pulses on Y axis (high word)	○	○	○							
D1944	Response time of MPG input on Y axis	×	○	○	-	-	-	R/W	YES	5	-
D1946	Electrical zero point address on Y axis (low word)	○	○	○	-	-	-	R/W	YES	0	-
D1947	Electrical zero point address on Y axis (high word)	○	○	○							
D1955*	Enable External Input for Y-Axis (ZRN, MPG, JOG-, JOG+)	○	○	○	-	-	-	R/W	YES	4	3-55
D1957	Enable/Disalbe PID control on Y axis	×	○	○	-	-	-	R/W	NO	0	
D1958	Set value for PID control on Y axis (Low word)	×	○	○	-	-	-	R	NO	0	
D1959	Set value for PID control on Y axis (High word)	×	○	○	-	-	-	R	NO	0	
D1960	Proportional gain (K _P) value on Y axis (Low word)	×	○	○	-	-	-	R/W	NO	0	
D1961	Proportional gain (K _P) value on Y axis (High word)	×	○	○	-	-	-	R/W	NO	0	
D1962	Intsgral gain (K _i) value on Y axis (Low word)	×	○	○	-	-	-	R/W	NO	0	
D1963	Intsgral gain (K _i) value on Y axis (High word)	×	○	○	-	-	-	R/W	NO	0	
D1964	Error value (difference) on Y axis (Low word)	×	○	○	-	-	-	R	NO	0	
D1965	Error value (difference) on Y axis (High word)	×	○	○	-	-	-	R	NO	0	
D1966	Accumulated error value on Y axis (Low word)	×	○	○	-	-	-	R	NO	0	

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
D1967	Accumulated error value on Y axis (High word)	×	○	○	-	-	-	R	NO	0	
D1968	The max accumulated error value on Y axis (Low word)	×	○	○	-	-	-	R	NO	0	
D1969	The max accumulated error value on Y axis (High word)	×	○	○	-	-	-	R	NO	0	
D1976*	Parameter setting of Z axis	○	×	○	-	-	-	R/W	YES	0	3-50
D1978	Pulses per revolution (PPR) at Z axis (low word)	○	×	○	-	-	-	R/W	YES	2,000	-
D1979	Pulses per revolution (PPR) at Z axis (high word)	○	×	○	-	-	-	R/W	YES	2,000	-
D1980	Displacement per revolution (DPR) of motor at Z axis (low word)	○	×	○	-	-	-		YES	1,000	-
D1981	Displacement per revolution (DPR) of motor at Z axis (high word)	○	×	○	-	-	-		YES	1,000	-
D1982	Maximum speed on Z axis: V_{MAX} (low word)	○	×	○	-	-	-	R/W	YES	500K	-
D1983	Maximum speed on Z axis: V_{MAX} (high word)	○	×	○	-	-	-	R/W	YES	500K	-
D1984	Bias speed of Z axis: V_{BIAS} (low word)	○	×	○	-	-	-	R/W	YES	0	-
D1985	Bias speed of Z axis: V_{BIAS} (high word)	○	×	○	-	-	-	R/W	YES	0	-
D1986	JOG speed of Z axis: V_{JOG} (low word)	○	×	○	-	-	-	R/W	YES	5,000	-
D1987	JOG speed of Z axis: V_{JOG} (high word)	○	×	○	-	-	-	R/W	YES	5,000	-
D1988	Zero return speed of Z axis: V_{RT} (low word)	○	×	○	-	-	-	R/W	YES	50K	-
D1989	Zero return speed of Z axis: V_{RT} (high word)	○	×	○	-	-	-	R/W	YES	50K	-
D1990	Zero return deceleration speed of Z axis: V_{CR} (low word)	○	×	○	-	-	-	R/W	YES	1,000	-
D1991	Zero return deceleration speed of Z axis: V_{CR} (high word)	○	×	○	-	-	-	R/W	YES	1,000	-
D1992	Number of Zero point signals (PG0) on Z axis: N	○	×	○	-	-	-	R/W	YES	0	-
D1993	Number of pulses for zero return on Z axis: P	○	×	○	-	-	-	R/W	YES	0	-
D1994	Definition of zero point on Z axis: HP (low word)	○	×	○	-	-	-	R/W	YES	0	-
D1995	Definition of zero point on Z axis: HP (high word)	○	×	○	-	-	-	R/W	YES	0	-
D1996	Acceleration time of Z axis: T_{ACC}	○	×	○	-	-	-	R/W	YES	500	-
D1997	Deceleration time of Z axis: T_{DEC}	○	×	○	-	-	-	R/W	YES	500	-
D1998	Target position (I) of Z axis: P(I) (low word)	○	○	○	0	-	-	R/W	NO	0	-
	PWM pulse width setting on Z axis (low word)	○	×	×	0	-	-	R/W	NO	0	-
D1999	Target position (I) of Z axis: P(I) (high word)	○	○	○	0	-	-	R/W	NO	0	-
	PWM pulse width setting on Z axis (high word)	○	×	×	0	-	-	R/W	NO	0	-
D2000	Operation speed (I) of Z axis: V(I) (low word)	○	×	○	1000	-	-	R/W	NO	1,000	-
D2001	Operation speed (I) of Z axis: V(I) (high word)	○	×	○	1000	-	-	R/W	NO	1,000	-
D2002	Target position (II) of Z axis: P(II) (low word)	○	○	○	0	-	-	R/W	NO	0	-
	PWM cycle setting on Z axis (low word)	○	×	×	0	-	-	R/W	NO	0	-
D2003	Target position (II) of Z axis: P(II) (high word)	○	○	○	0	-	-	R/W	NO	0	-
	PWM cycle setting on Z axis (high word)	○	×	×	0	-	-	R/W	NO	0	-

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
D2004	Operation speed (II) of Z axis: V(II) (low word)	○	×	○	2000	-	-	R/W	NO	2000	-
D2005	Operation speed(II) of Z axis: V(II) (high word)										
D2006*	Operation commands for Z axis	○	×	○	0	-	0	R/W	NO	0	3-53
D2007	Work modes for Z axis	×	×	○	0	-	-	R/W	NO	0	3-53
D2008	Current position of Z axis: CP (pulse) (low word)	○	×	○	0	-	-	R/W	NO	0	-
D2009	Current position of Z axis: CP (pulse) (high word)										
D2010	Current speed of Z axis: CS (pps) (low word)	○	×	○	0	0	0	R/W	NO	0	-
D2011	Current speed of Z axis: CS (pps) (high word)										
D2012	Current position of Z axis: CP (unit) (low word)	○	×	○	0	-	-	R/W	NO	0	-
D2013	Current position of Z axis: CP (unit) (high word)										
D2014	Current speed of Z axis: CS (unit) (low word)	○	×	○	0	0	0	R/W	NO	0	-
D2015	Current speed of Z axis:CS (unit) (high word)										
D2016	Execution status of Z axis	○	×	○	0	-	-	R	NO	0	3-54
D2017*	Error code of Z axis	○	×	○	0	-	-	R	NO	0	3-45
D2018	Numerator of electronic gear ratio of Z axis	○	×	○	-	-	-	R/W	YES	1	-
D2019	Denominator of electronic gear ratio of Z axis	○	×	○	-	-	-	R/W	YES	1	-
D2020	MPG input frequency on Z axis (low word)	○	×	○	0	-	0	R/W	NO	0	-
D2021	MPG input frequency on Z axis (high word)										
D2022	Number of accumulated MPG input pulses at Z axis (low word)	○	×	○	0	-	-	R/W	NO	0	-
D2023	Number of accumulated MPG input pulses at Z axis (high word)										
D2024	Response time of MPG input at Z axis	○	×	○	-	-	-	R/W	YES	5	-
D2026	Electrical zero point address on Z axis (low word)	○	×	○	-	-	-	R/W	YES	0	-
D2027	Electrical zero point address on Z axis (high word)										
D2029	Error Step of OZ (reserved, currently not available)	○	×	○	-	-	-	R/W	YES	0	-
D2037	Enable/Disalbe PID control on Z axis	×	○	○	-	-	-	R/W	NO	0	-
D2038	Set value for PID control on Z axis (Low word)	×	○	○	-	-	-	R	NO	0	-
D2039	Set value for PID control on Z axis (High word)	×	○	○	-	-	-	R	NO	0	-
D2040	Proportional gain (K _P) value on Z axis (Low word)	×	○	○	-	-	-	R/W	NO	0	-
D2041	Proportional gain (K _P) value on Z axis (High word)	×	○	○	-	-	-	R/W	NO	0	-
D2042	Intsgral gain (K _i) value on Z axis (Low word)	×	○	○	-	-	-	R/W	NO	0	-
D2043	Intsgral gain (K _i) value on Z axis (High word)	×	○	○	-	-	-	R/W	NO	0	-
D2044	Error value (difference) on Z axis (Low word)	×	×	○	-	-	-	R	NO	0	-
D2045	Error value (difference) on Z axis (High word)	×	×	○	-	-	-	R	NO	0	-
D2046	Accumulated error value on Z axis (Low word)	×	×	○	-	-	-	R	NO	0	-
D2047	Accumulated error value on Z axis (High word)	×	×	○	-	-	-	R	NO	0	-

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
D2048	The max accumulated error value on Z axis (Low word)	×	×	○	-	-	-	R	NO	0	-
D2049	The max accumulated error value on Z axis (High word)	×	×	○	-	-	-	R	NO	0	-
D2056*	Parameter setting of A axis	○	×	×	-	-	-	R/W	YES	0	3-50
D2058	Pulses per revolution (PPR) on A axis (low word)	○	×	×	-	-	-	R/W	YES	2000	-
D2059	Pulses per revolution (PPR) on A axis (high word)										
D2060	Displacement per revolution (DPR) of motor on A axis (low word)	○	×	×	-	-	-		YES	1000	
D2061	Displacement per revolution (DPR) of motor on A axis (high word)										
D2062	Maximum speed on A axis: V_{MAX} (low word)	○	×	×	-	-	-	R/W	YES	500K	-
D2063	Maximum speed on A axis: V_{MAX} (high word)										
D2064	Bias speed of A axis: V_{BIAS} (low word)	○	×	×	-	-	-	R/W	YES	0	-
D2065	Bias speed of A axis: V_{BIAS} (high word)										
D2066	JOG speed of A axis: V_{JOG} (low word)	○	×	×	-	-	-	R/W	YES	5000	-
D2067	JOG speed of A axis: V_{JOG} (high word)										
D2068	Zero return speed of A axis: V_{RT} (low word)	○	×	×	-	-	-	R/W	YES	50K	-
D2069	Zero return speed of A axis: V_{RT} (high word)										
D2070	Zero return deceleration speed of A axis: V_{CR} (low word)	○	×	×	-	-	-	R/W	YES	1000	-
D2071	Zero return deceleration speed of A axis: V_{CR} (high word)										
D2072	Number of Zero point signals (PG0) on A axis:N	○	×	×	-	-	-	R/W	YES	0	-
D2073	Number of pulses for zero return on A axis: P	○	×		-	-	-	R/W	YES	0	-
D2074	Definition of zero point on A axis: HP (low word)	○	×	×	-	-	-	R/W	YES	0	-
D2075	Definition of zero point on A axis: HP (high word)										
D2076	Acceleration time of A axis: T_{ACC}	○	×	×	-	-	-	R/W	YES	500	-
D2077	Deceleration time of A axis: T_{DEC}	○	×	×	-	-	-	R/W	YES	500	-
D2078	Target position (I) of A axis: P(I) (low word)	○	×	×	0	-	-	R/W	NO	0	-
	PWM pulse width setting on A axis (low word)	○									
D2079	Target position (I) of A axis: P(I) (high word)	○	×	×							
	PWM pulse width setting on X axis (high word)	○									
D2080	Operation speed (I) of A axis: $V(I)$ (low word)	○	×	×	1000	-	-	R/W	NO	1000	-
D2081	Operation speed (I) of A axis: $V(I)$ (high word)										
D2082	Target position (II) of A axis: P(II) (low word)	○	×	×	0	-	-	R/W	NO	0	-
	PWM cycle setting on A axis (low word)	○									
D2083	Target position (II) of A axis: P(II) (high word)	○	×	×							
	PWM cycle setting on A axis (high word)	○									

3 Functions of Devices in DVP-PM

Special D	Function	10M	20D	20M	OFF ↓ ON	MANU ↓ AUTO	AUTO ↓ MANU	Attrib.	Latched	Default	Page number
D2084	Operation speed (II) of A axis: V(II) (low word)	○	×	×	2000	-	-	R/W	NO	2K	-
D2085	Operation speed(II) of A axis: V(II) (high word)										
D2086	Operation commands for A axis	○	×	×	0	-	0	R/W	NO	0	3-53
D2088	Current position of A axis: CP (pulse) (low word)	○	×	×	0	-	-	R/W	NO	0	-
D2089	Current position of A axis: CP (pulse) (high word)										
D2090	Current speed of A axis: CS (pps) (low word)	○	×	×	0	0	0	R/W	NO	0	-
D2091	Current speed of A axis: CS (pps) (high word)										
D2092	Current position of A axis: CP (unit) (low word)	○	×	×	0	-	-	R/W	NO	0	-
D2093	Current position of A axis: CP (unit) (high word)										
D2094	Current speed of A axis: CS (unit) (low word)	○	×	×	0	0	0	R/W	NO	0	-
D2095	Current speed of A axis:CS (unit) (high word)										
D2096	Execution status of A axis	○	×	×	0	-	-	R	NO	0	3-54
D2097*	Error code of A axis	○	×	×	0	-	-	R	NO	0	-
D2098	Numerator of electronic gear ratio of A axis	○	×	×	-	-	-	R/W	YES	1	-
D2099	Denominator of electronic gear ratio of A axis	○	×	×	-	-	-	R/W	YES	1	-

Note: 10M = DVP10PM, 20D = DVP20PM00D, 20M = DVP20PM00M

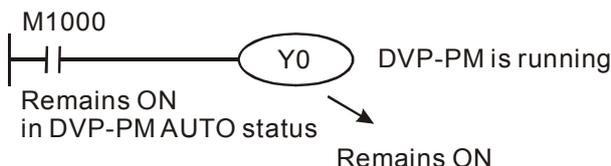
3.11 Functions of Special Auxiliary Relays and Special Registers

Function Group: DVP-PM Operation Flag

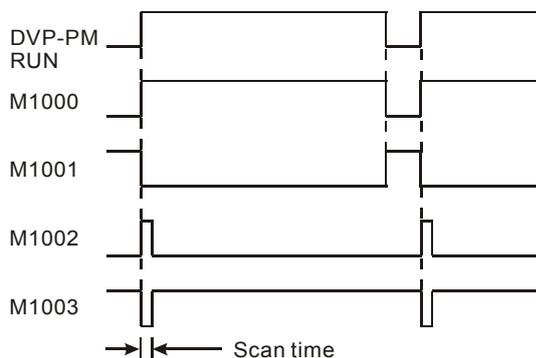
Number: M1000 ~ M1003

Explanations:

1. **M1000:** NO contact for monitoring DVP-PM status. M1000 remains “ON” when DVP-PM is running.



2. **M1001:** NC contact for monitoring PLC status. M1001 remains “OFF” when PLC is running.
3. **M1002:** Enables single positive pulse for the first scan when DVP-PM AUTO is activated. Remains OFF after the initial pulse. Used to initialize registers, outputs, or counters when AUTO is executed. Pulse width = one scan time
4. **M1003:** Enables single negative pulse for the first scan when DVP-PM AUTO is activated. Remains ON after the initial pulse. Used to initialize registers, outputs, or counters when AUTO is executed. Pulse width = one scan time

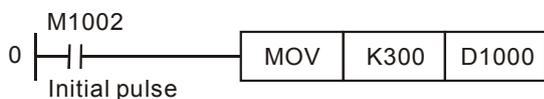


Function Group: Monitor Timer

Number: D1000

Explanations:

1. Monitor timer is used for monitoring PLC scan time. When the scan time exceeds the set value (SV) in the monitor timer, the red ERROR LED will be ON and all outputs will be “OFF”.
2. The default in the monitor timer is 200ms. If the program is long or the operation is too complicated, MOV instruction can be used to modify SV. See the example below for setting SV as 300ms.



3. The maximum SV in the monitor timer is 32,767ms. However, care should be taken when adjusting SV. If SV in D1000 is too big, it cost much longer to detect operation errors. Therefore, SV is suggested to be shorter than 200ms.

3 Functions of Devices in DVP-PM

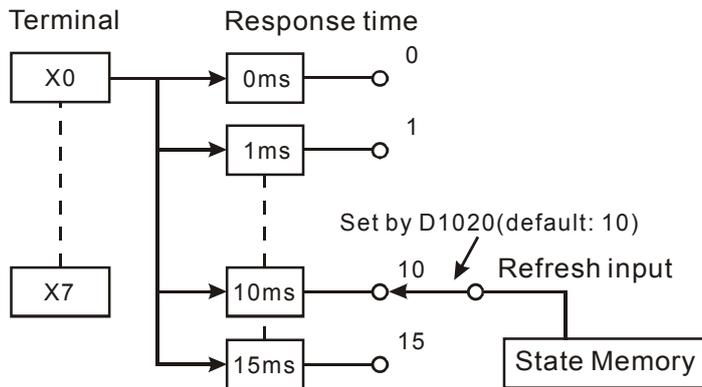
4. Scan time could be prolonged due to: 1. too many complicated instructions in the program or, 2. too many I/O modules are being connected. Check D1010 ~ D1012 to see if the scan time exceeds the SV in D1000. In addition, SV in D1000 can be modified to fit the proper scan time.

Function Group: Input filter (Response time setting)

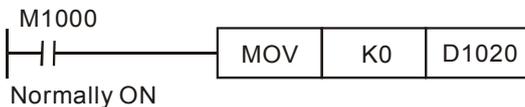
Number: D1020

Explanations:

1. Input terminals X0~X7 support response time setting by D1020. Range: 0~20, unit: ms.
2. When PLC is powered from OFF to ON, the content in D1020, D1021 will be set as 10 automatically.



3. If the program executes the instruction below with the response time setting as 0ms, the fastest response time will be 50μs due to a series connection with RC filter circuit.



4. Input filter time requires no modification when high speed counters or interrupts are applied in program.

Function Group: COM Port Function

Number: M1120, M1136, M1138, M1139, M1143, D1036, D1120, D1109

Explanations:

COM ports (COM1: RS-232, COM2: RS-485) in DVP-PM support Modbus ASCII/RTU format with communication speed of up to 115,200bps. Optional communication card (COM3:RS-232/RS-485) supports Modbus ASCII format with speed of up to 38,400bps. COM1, COM2 and COM3 can be used at the same time.

COM1: Can only be used as Slave. COM1 supports ASCII/RTU communication format, adjustable baud rate with speed of up to 115,200bps, and modification on data length (data bits, parity bits, stop bits).

COM2: Can be used as Master or Slave. COM2 supports ASCII/RTU communication format, adjustable baud rate with speed of up to 115,200bps, and modification on data length (data bits, parity bits, stop bits).

COM3: Can only be used as Slave. COM3 supports ASCII communication format, adjustable baud rate with speed of up to 38,400bps. COM2 and COM3 can not be used as Slave at the same time.

■ Communication format settings:

COM1: 1. D1036: communication protocol, b8 ~ b15 are fixed when COM1 (RS-232) is used as Slave

3 Functions of Devices in DVP-PM

2. M1138: retain communication setting.

3. M1139: ASCII/RTU mode selection.

COM2: 1. D1120: RS-232/RS-485/RS-422 communication protocol when used as Master or Slave

2. M1120: retain communication setting.

3. M1143: ASCII/RTU mode selection.

COM3: 1. D1109: communication protocol, b0~b3、b8~b15 are fixed

2. M1136: retain communication setting

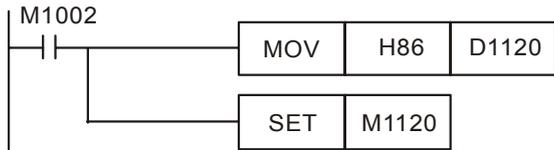
■ Communication format:

	Content	
b0	Data Length	0: 7 data bits, 1: 8 data bits
b1 b2	Parity bit	00: None 01: Odd 11: Even
b3	Stop bits	0: 1 bit, 1: 2bits
b4 b5 b6 b7	Baud rate	0001(H1): 110 0010(H2): 150 0011(H3): 300 0100(H4): 600 0101(H5): 1200 0110(H6): 2400 0111(H7): 4800 1000(H8): 9600 1001(H9): 19200 1010(HA): 38400 1011(HB): 57600 1100(HC): 115200
b8	Start bit (STX)	0: None 1: D1124
b9	The 1 st end bit (ETX1)	0: None 1: D1125
b10	The 2 nd end bit (ETX2)	0: None 1: D1126
b11~b15	Undefined	

Example 1: Modifying communication format of COM2

1. Add the below instructions on top of the program to modify the communication format of COM2. When PLC switches from MANU to AUTO, the program will detect whether M1120 is ON in the first scan. If M1120 is ON, the program will modify the communication settings of COM2 according to the value set in D1120.
2. Modify COM2 communication format to ASCII mode, 9600bps, 7 data bits, even parity, 1 stop bits (9600, 7, E, 1)

3 Functions of Devices in DVP-PM

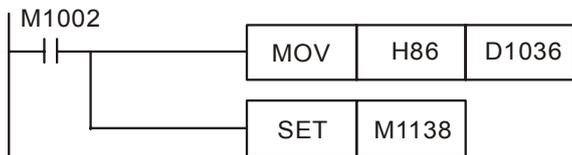


Notes:

1. Do NOT write any communication instruction in the program when COM2 is used as slave..
2. After the communication format is modified, the format will stay intact when PLC switches from AUTO to MANU.
3. Communication format will be reset to default setting after power is shut down.

Example 2: Modifying the communication format of COM1

1. Add the below instructions on top of the program to modify the communication format of COM1. When PLC switches from MANU to AUTO, the program will detect whether M1138 is ON in the first scan. If M1138 is ON, the program will modify the communication settings of COM1 according to the value set in D1036
2. Modify COM1 communication format to ASCII mode, 9600bps, 7 data bits, even parity, 1 stop bits (9600, 7, E, 1)

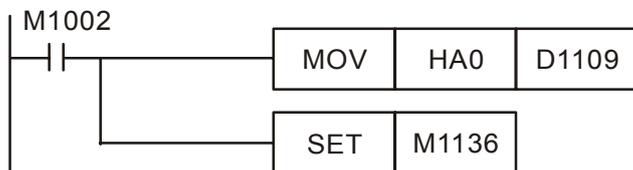


Note:

1. After the communication format is modified, the format will stay intact when PLC switches from AUTO to MANU.
2. Communication format will be reset to default setting after power is shut down.

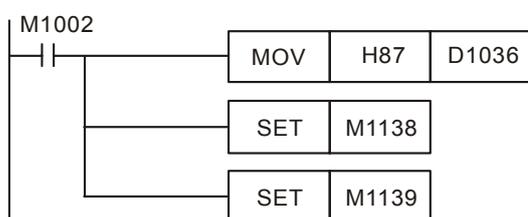
Example 3: Modifying the communication format of COM3

1. Communication format of COM3 is fixed as 7 Data bits, Even parity, 1 Stop bit. If users need to modify the communication speed of COM3 into 38,400bps, add the below instructions on top of the program. When PLC switches from MANU to AUTO, the program will detect whether M1136 is ON in the first scan. If M1136 is ON, the program will modify the communication speed of COM3 according to the value set in D1109.
2. Modify COM3 communication speed to 38,400 bps.



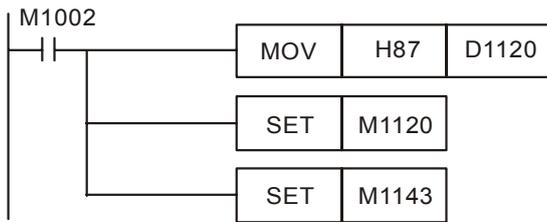
Example 4: Setting up RTU mode of COM1 and COM2

COM1:



COM2:

3 Functions of Devices in DVP-PM



Function Group: Communication Response Delay

Number: D1038

Explanations:

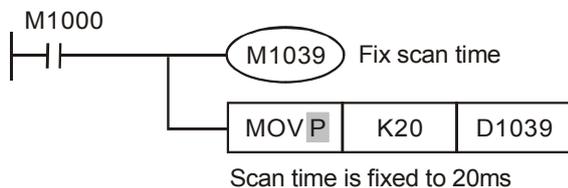
Data response delay time can be set when DVP-PM used as Slave in COM2 RS-485 communication. Unit: 10ms. 0~3,000 (0~30 sec) adjustable. Set value should be smaller than the value in D1000. If the set value is out of the valid range, D1038 will be set as 0.

Function Group: Fixed Scan Time

Number: M1039, D1039

Explanations:

1. When M1039 is ON, program scan time is determined by D1039. When program execution is completed, next scan will be activated only when the fixed scan time is reached. If D1039 is less than actual scan time, it will scan by the actual program scan time.



2. The fixed scan time will also be stored in D1010 ~ D1012.

Function Group: Setting up the No. of OX Subroutine

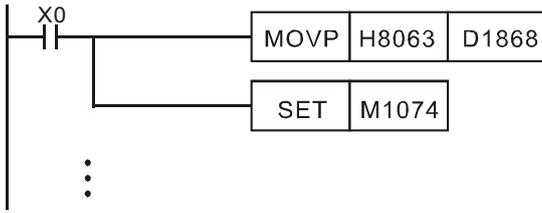
Number: M1074, D1868

Explanations:

D1868 designates the No. of OX subroutine to be executed, please refer to the following example for the setting steps.

1. Set b14 or b15 of D1868 to be "1", or b14 = b15 = 1 (only one of the three conditions to be met). In addition, write K99 (= H'63) into b0 ~ b13 of D1868, i.e. set OX as OX99. To conclude, we write H'8063 into D1868.
2. After step 1 is finished, set up M1074, and the OX program designated by D1868 will be enabled.
3. Program example:

3 Functions of Devices in DVP-PM



In O100 main program, X0 enables OX99 subroutine and executes the instructions in OX99

Function Group: High Speed Counting (DVP-10PM)

Number: M1200/C200, M1204/C204, M1208/C208, M1212/C212, M1216/C216, M1220/C220

Explanations:

DVP-10PM provides 6 sets of high speed counters as the table below:

	Counter	Counting mode		Reset signal (Valid only when b3 = 1)	Input signal
		Device	b0 / b1		
1 st set	C200	K1M1200	0: U/D*	X10- (M1203)	X0, X1, S/S
2 nd set	C204	K1M1204	1: P/D*	X11- (M1207)	X2, X3, S/S
3 rd set	C208	K1M1208	2: A/B* (Single frequency)	X12- (M1211)	X4, X5, S/S
4 th set	C212	K1M1212	3:4A/B (Quadruple frequency)	X13- (M1215)	X6, X7, S/S
5 th set	C216	K1M1216		X0 (M1219)	X10+, X10-, X11+, X11-
6 th set	C220	K1M1220		X1 (M1223)	X12+, X12-, X13+, X13-

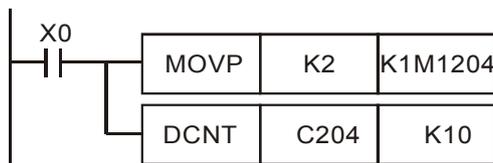
Note:

- U/D: Counting up / Counting down, P/D: Pulse / Direction, A/B: A-phase / B-phase
- Counters from set 1~4 apply open collector circuit; set 5~6 apply differential circuit.

The setting steps for using the 2nd set high speed counter are as below:

1. Design the 2nd set of high speed counter with A/B phase counting mode by setting K1M1204 = K2
2. Execute the instruction to start the counting.

The program of step 1 and step 2 is as below:



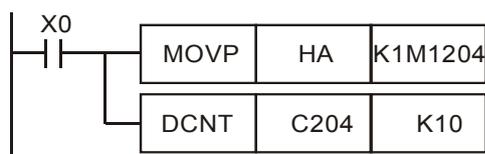
3. If the reset signal is to be triggered from external signal, set K1M1024=H'A

M1027	M1026	M1025	M1024
1	0	1	0

4. Execute the instruction to start the counting of C204. When X0 is ON, C204 will be cleared immediately.

3 Functions of Devices in DVP-PM

The program of step 3 and step 4 is as below:



In O100 main program, design the 2nd set of high speed counter with both A/B phase counting mode and reset function by setting K1M1024 = HA, i.e. b0/b1 = K2 and b3(M1207) = 1. When the instructions execute, C204 will start the counting. If the reset signal input X11- is triggered, the value in C204 will be reset immediately.

Function Group: High Speed Timing

Number: M1200/C201, M1204/C205, M1208/C209, M1212/C213, M1216/C217, M1220/C221

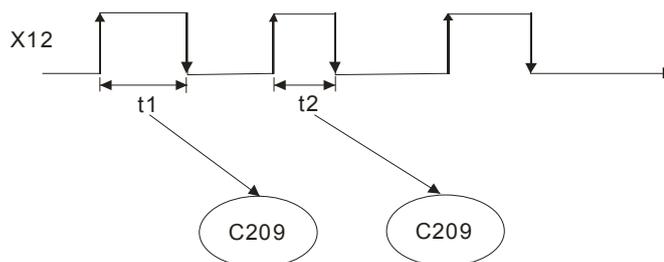
Explanations:

DVP-10PM provides 6 sets of high speed timers as the table below:

	Timer	Timing mode				Start signal	Storage timer	
		Device	b0 / b2					
1 st set	C200	K1M1200	Bit3	Bit2	Bit1	Bit0	X10	C201
2 nd set	C204	K1M1204	x	Start timing	x	Mode setting	X11	C205
3 rd set	C208	K1M1208	b0 = 0: Normal mode. The timing interval is from the rising edge to the falling edge. b0 = 1: Cycle mode. The timing interval is from one rising edge to the next rising edge b2 = 1: Enabling high speed timing function				X12	C209
4 th set	C212	K1M1212					X13	C213
5 th set	C216	K1M1216					X0	C217
6 th set	C220	K1M1220					X1	C221

Example 1: Use the 3rd set of timer and set up the timing mode as Normal mode:

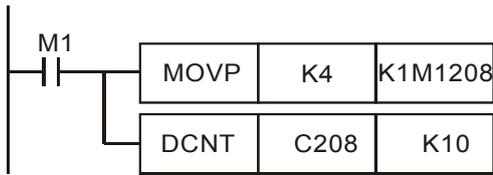
- In O100 main program, design the 3rd set of high speed timer with normal timing mode by setting K1M1208 = K4, i.e. b0=0, b2 = 1.
- Execute the program and C208 will start to time the interval from rising edge to falling edge of the start signal X12. The timed interval will be stored in C209 with the timer resolution 0.01us as the timing diagram below:



Normal mode. Unit: 0.01μs

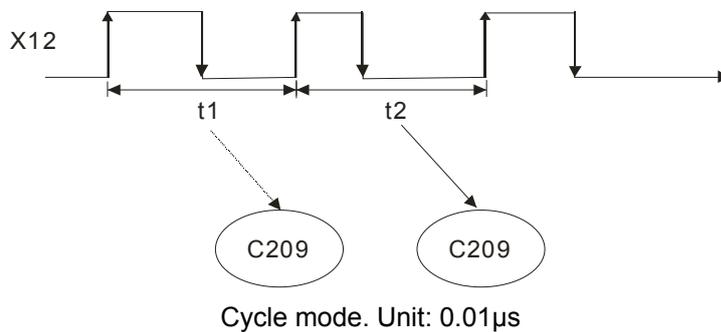
The example program of step 1 and step 2 is as below:

3 Functions of Devices in DVP-PM

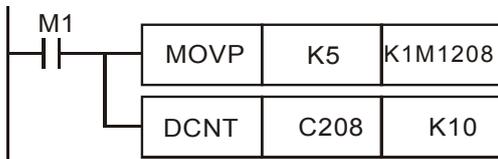


Example 2: Use the 3rd set of timer and set up the timing mode as Cycle mode:

1. In O100 main program, design the 3rd set of high speed timer with cycle timing mode by setting K1M1208 = K5, i.e. b0 = 1, b2 = 1.
2. Execute the program and C208 will start to time the interval between rising edges of the start signal X12. The timed interval will be stored in C209 with the timer resolution 0.01 μ s. Please see the below diagram for the timing diagram:



The example program of step 1 and step 2 is as below:



3. Cycle mode is mainly used in frequency measuring application.

Function Group: I/O Modules Detection

Number: D1140, D1142, D1143

Explanations:

D1140: Number of right-side modules (AIO, PT, TC, etc.), max. 8 modules can be connected.

D1142: Number of input points (X) on DIO modules.

D1143: Number of output points (Y) on DIO modules

Function Group: Setting Up Latched Area

Number: D1200 ~ D1211

Explanations:

1. The specified range between D1200 (start) and D1211 (end) will be set up as the latched area of DVP-PM.
2. Please refer to the tables in 3.1 for more details.

3 Functions of Devices in DVP-PM

Function Group: Enabling force-ON/OFF of input point X

Number: M1304

Explanations:

When M1304 = ON, the peripheral devices, e.g. PMSoft, can force input points X0~X17 to be ON/OFF. However, the LED will not respond to the ON/OFF state in this case.

Function Group: ID of right-side modules

Number: D1320 ~ D1327

Explanations:

1. When I/O modules are connected, the ID of each I/O module will be stored in D1320~D1327 in connection order.
2. ID of Special I/O modules for DVP-PM:

Module Name	Module ID (hex)	Module Name	Module ID (hex)
DVP04AD-H2	H'6400	DVP01PU-H2	H'6110
DVP04DA-H2	H'6401	DVP04PT-H2	H'6402
DVP04TC-H2	H'6403	DVP06XA-H2	H'6604
DVPPM	H'6260	DVP01HC-H2	H'6120

Function Group: Interruption

Number: D1400, D1401

Explanations:

1. To enable the interrupts below, please set ON the corresponding bits of D1400.

bit#	Interruption	No.
0	Timer interrupt	10
1	External interrupt Start0 / X0*	11
2	External interrupt Stop0 / X1*	12
3	External interrupt Start1 / X2*	13
4	External interrupt Stop1 / X3*	14
5	External interrupt X4*	15
6	External interrupt X5*	16
7	External interrupt X6*	17
8	External interrupt X7*	18

Note: X0*~X7* are applicable only for DVP-10PM

2. If timer interrupt is enabled, please enter the interrupt interval in D1401
3. There are 2 types of interrupts including external interrupt and timer interrupt:
 - External interrupt: When the external terminal is triggered by the rising/falling edge of an input signal, the current executing program will be interrupted and the interrupt subroutine will be executed immediately

3 Functions of Devices in DVP-PM

regardless of the scan cycle. When the interrupt subroutine is completed, the program execution will return to the previous program before interrupt occurs.

- Timer interrupt: The interrupt subroutine will be executed repeatedly by the interrupt interval set up in D1401.

Function Group: Clearing M-code In Execution

Number: M1744, M1794, D1703

Explanations:

1. Set M1744 = 1 to clear the executing M-code instruction. When M1744 is executed, D1703 will be cleared and M1794 will be reset.
2. M1794 is the flag indicating OX subroutines are being executed. D1703 is the register storing the executing M-code.

Function Group: Ready flags

Number: M1792, M1872, M2032, M2112

Explanations:

1. There are 4 ready flags indicating the ready status of X-Y-Z-A axes. X-axis: M1792, Y axis: M1872, Z axis: M2032 and A axis: M2112.
2. For example, if X-axis is ready before operation, M1792 will be ON. During the operation of X-axis M1792 will be OFF and then turn ON again when the operation on X-axis is completed.

Function Group: Current position write in function

Number: M1751, M1831, M1991, M2071

Explanations:

1. If the user needs to write in (change) the current position of X-Y-Z-A axes, set ON flags M1751, M1831, M1991, M2071 and the attribute of current position register will be changed from read only to read/write available. For example, when M1751 = ON, its corresponding current position register D1848 will become available for changing current position.
2. In addition, if D1848 is written when M1751 = OFF, the value in D1848 will be cleared.

Function Group: Stop Single Axis Movement at Fixed Position

Number: M1761, M1841, M2001

Explanations:

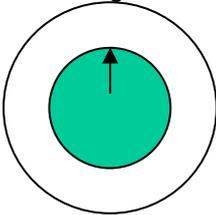
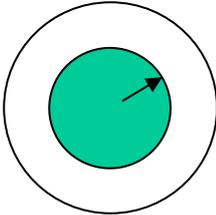
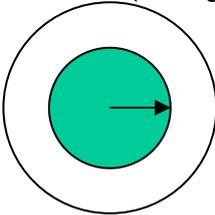
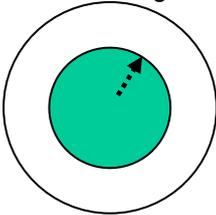
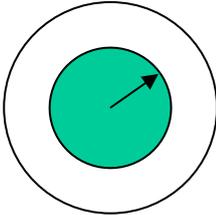
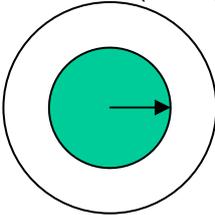
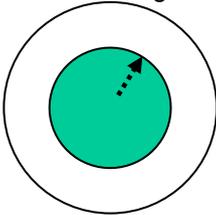
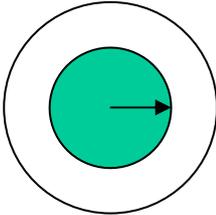
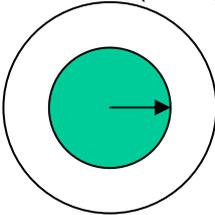
1. The function can be used on controlling the stop position of X/Y/Z axis and should be enabled with JOG+ function of manual mode.
2. How to set stop position: If start position is 0 degree, pulses per round is set as 20,000, and the target stop angle is 90 degree, the stop position should be set as 5,000.

3 Functions of Devices in DVP-PM

3. The relative registers are listed as below.

X - axis	Y- axis	Z - axis	Function
M1761	M1841	M2001	Stop at fixed position
D1839, 1838	D1919..1918	D1999..1998	Stop position
D1843, 1842	D1923..1922	D2003, 2002	Pulses per round

4. For example, if pulses per round is set as 20,000 and the target stop angle is 90 degree, the setting of stop position could be the cases below:

Start position	JOG+ to stop	Stop position
Case 1: If the operation starts at 0 degree and JOG+ signal is triggered at position 4000, it should stop at position 5000		
0 degree 	Position: 4000 	Position: 5000 (90 degree) 
Case 2: If the operation starts at random degree and JOG+ signal is triggered at position 63500, it should stop at position 65000.		
Random degree 	Position: 63500 	Position: 65000 (90 degree) 
Case 3: If the operation starts at random degree and JOG+ signal is triggered at position 25001, it should go 19999 pulses and stop at position 45000.		
Random degree 	Position: 25001 	Position: 45000 (90 degree) 

Function Group: Error Status of Motion Control

Number: M1793, D1857, M1873, D1937, M2033, D2017

Explanations:

- When errors occur on X, Y or Z axis, the corresponding error flags M1793, M1873 and M2033 will be ON, and the error messages will be stored in D1857, D1937 and D2017.
- To eliminate the error, please clear the error message registers and reset the error flags.

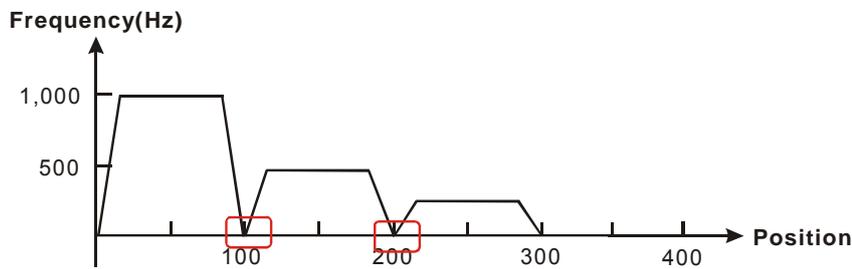
3 Functions of Devices in DVP-PM

Function Group: Continuous Interpolation

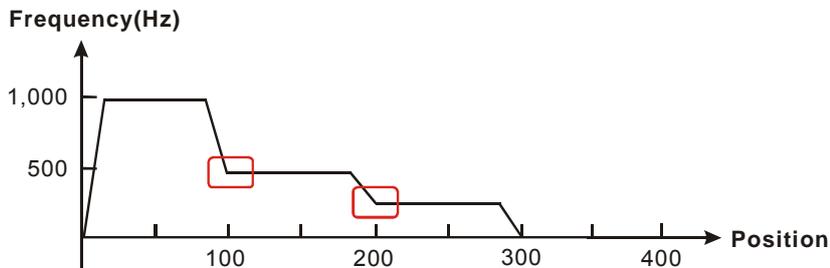
Number: D1796, M1036

Explanations:

1. When you use D1796 to set up the deceleration speed (kHz), DVP-PM will compare the set value in D1796 with the actual deceleration frequency then takes the smaller frequency as the transition frequency of continuous interpolation.
2. When $D1796 > 0$, the continuous interpolation is enabled to form a continuous path. However, the function is only effective to G1/G2/G3/LIN/CW/CCW instructions. You can set $M1036 = ON$ ($M1036 = K1$) to enable continuous interpolation on general application instructions.
3. When $D1796 = K0$ (continuous interpolation disabled) the speed will decelerate to 0kHz between each deceleration step regardless of the actual deceleration frequency. See the figure below.



4. When $D1796 = K500$ (continuous interpolation enabled) DVP-PM will compare K500 with the actual deceleration frequency and takes the smaller frequency as the transition speed. See the figure below.



5. Continuous path demonstration

When D1796 (continuous interpolation) has not been set, the $X \rightarrow Y$ path will be a right-angle turn. After D1796 is set, the $X \rightarrow Y$ path will perform a smooth transition.

Program example:

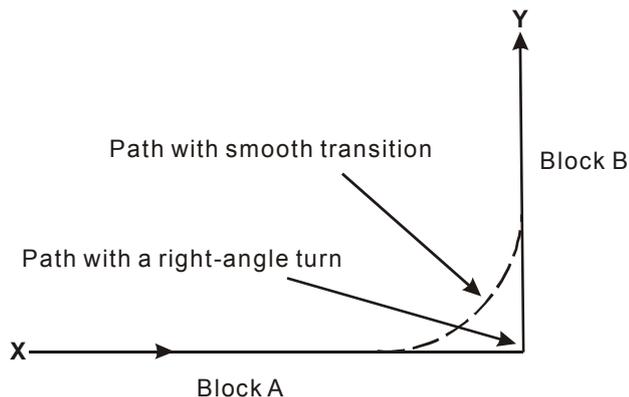
G01 X100 F1000; (Block A)

Y100; (Block B)

Block A: execute moving path of X-axis

Block B: execute moving path of Y-axis

3 Functions of Devices in DVP-PM



Function Group: Setting up Percentage Value of G-Code Moving Speed

Number: D1798

Explanations:

1. When D1798 is set to 100, the G-Code moving speed will be the original speed. When D1798 is set to 1,000, the G-Code moving speed will become 10 times faster than the original speed. When D1798 is set to 50, the G-Code moving speed will decrease to half of the original speed, and so forth.
2. The upper limit of G-Code moving speed is 500k(Hz). When the G-Code moving speed is set to be faster than 500k(Hz), the speed will be set as 500k(Hz) automatically

Function Group: Setting up Input Terminal Polarity (for X/Y-axis)

Number: D1799

Explanations:

For input terminals below, setting the corresponding bit# to be ON will define the input terminal as NO contact; setting the corresponding bit# to be OFF will define the input terminal as NC contact.

20PM:

bit#	Input terminal polarity on X axis	bit#	Input terminal polarity on Y axis
0	PG0	8	PG01
1	MPGB0	9	MPGB1
2	MPGA0	10	MPGA1
3	LSN0	11	LSN1
4	LSP0	12	LSP1
5	DOG0	13	DOG1
6	STOP0	14	STOP1
7	START0	15	START1

10PM:

bit#	Input terminal polarity
0	DOG0

3 Functions of Devices in DVP-PM

bit#	Input terminal polarity
1	PG0
2	DOG1
3	PG1
4	DOG2
5	PG2
6	DOG3
7	PG3

Function Group: Reading the Status of Input Terminal

Number: D1800

Explanations:

When there is signal input at the input terminal, the corresponding bit# = ON. If not, the corresponding bit# = OFF.

20PM:

bit#	Input terminal status on X axis	bit#	Input terminal status on Y axis
0	PG0	8	PG0
1	MPGB0	9	MPGB1
2	MPGA0	10	MPGA1
3	LSN	11	LSN
4	LSP	12	LSP
5	DOG	13	DOG
6	STOP	14	STOP
7	START	15	START

10PM:

bit#	Input terminal status
0	DOG0
1	PG0
2	DOG1
3	PG1
4	DOG2
5	PG2
6	DOG3
7	PG3

Function Group: Setting up Input terminal polarity 2 (for Z-axis)

Number: D1804

Explanations:

3 Functions of Devices in DVP-PM

For input terminals below, setting the corresponding bit# to be ON will define the input terminal as NO contact; setting the corresponding bit# to be OFF will define the input terminal as NC contact.

20PM :

bit#	Polarity of input terminal
0	PG2
1	MPGB1
2	MPGA1
3	LSN2
4	LSP2
5	DOG2

10PM :

bit#	Polarity of input terminal
0	X3(DOG3)
1	MPGB1
2	MPGA1
3	X2(DOG2)
4	X1(DOG1)
5	X0(DOG0)

Function Group: Reading Input terminal status 2 (for Z-axis)

Number: D1805

Explanations:

When there is signal input at the input terminal, the corresponding bit# = ON. If not, the corresponding bit# = OFF

20PM :

bit#	Input terminal status
0	PG2
1	MPGB1
2	MPGA1
3	LSN2
4	LSP2
5	DOG2

10PM :

bit#	Input terminal status
0	X3(DOG3)
1	MPGB1
2	MPGA1
3	X2(DOG2)
4	X1(DOG1)
5	X0(DOG0)

Function Group: Filter coefficient

Number: D1806

Explanations:

1. The low byte of D1806 sets the filter coefficient of the input terminals START, STOP, DOG, LSN, LSP and PG
2. The high byte of D1806 sets the filter coefficient of the input terminal MPG0/MPG1.
3. Filter coefficient = $\frac{85000}{2^{N+4}}$ (KHz), N = 1~19; the conversion results are as below:

N	kHz	N	kHz
1	2656.25	11	2.593994
2	1328.125	12	1.296997
3	664.0625	13	0.648499
4	332.0313	14	0.324249

3 Functions of Devices in DVP-PM

5	166.0156	15	0.162125
6	83.00781	16	0.081062
7	41.50391	17	0.040531
8	20.75195	18	0.020266
9	10.37598	19	0.010133
10	5.187988		

4. When D1806 = 0, the filter function on external terminals will be disabled.
5. For example, if D1806 is set as H0A0A, the filter coefficient on input terminals START, STOP, DOG, LSN, LSP and PG and MPG0/1 will be: $\frac{85000}{2^{10+4}} = 5.187988$ (KHz), i.e. input frequency higher than 5.187988KHz will be filtered out.

Function Group: Error Status of O100
Number: M1953, D1802, D1803

Explanations:

- When errors occur in O100 program, O100 error flag M1953 will be ON, the error code will be stored in D1802, and error step will be stored in D1803.
- See the table of error messages in Appendix of Chapter 14.

Function Group: Parameter Setting of X-Y-Z-A Axis
Number: D1816, D1896, D1976, D2056

Explanations:

Parameter setting of X axis: D1816, Y axis: D1896, Z axis: D1976 and A axis: D2056. See the tables below:

bit#	X-Y-Z axis parameter setting	bit#	X-Y-Z axis parameter setting
0	Unit (*1)	8	Zero return direction (*4)
1		9	Zero return mode (*4)
2	Multiplication of position data (*2)	10	DOG signal rising/falling edge triggered (*4)
3		11	Reverse the displacement direction in same pulse output polarity (*4)
4	Pulse type (*3)	12	Relative/absolute positioning (*4)
5		13	DOG signal rising/falling edge triggered (for single/ 2 speed positioning interruption mode) (*4)
6*	PWM mode (*4)	14	Curve selection (*4)
7		15	

Note: PWM mode is only applicable for DVP-10PM

Note *1:

3 Functions of Devices in DVP-PM

b1	b0	Unit	Motor unit	Combined unit	Machine unit
0	0	Motor	pulse	um	
0	1	Machine	pulse	m deg	
1	0	Combined	pulse	10 ⁻⁴ inch	
1	1				
			pulse/sec		cm/min
			pulse/sec		10deg/min
			pulse/sec		inch/min

Note *2:

b3	b2	Multiplication of position data
0	0	10 ⁰
0	1	10 ¹
1	0	10 ²
1	1	10 ³

Note *3:

b5	b4	Description
0	0	Forward pulse + reverse pulse
0	1	Pulse + direction
1	0	A/B phase pulse
1	1	

Note *4:

bit#	Explanation
6	b[6]=1: Enable PWM mode (1) When the operation command "JOG+ operation" is enabled, PWM will be executed on Y0~Y3. (2) When the operation command "single speed positioning" is enabled, single speed positioning will be executed on Y0~Y3 with single-phase pulses. (3) PWM pulse width setting: D1838, D1918, D1998, D2078 (4) PWM cycle setting: D1842, D1922, D2002, D2082
8	b[8]=0: Zero return executes in positive direction, i.e. current position(CP) increases b[8]=1: Zero return executes in negative direction, i.e. current position(CP) decreases
9	b[9]=0: normal mode b[9]=1: overwrite mode
10	b[10]=0: DOG signal falling edge triggered b[10]=1: DOG signal rising edge triggered
11	b[11]=0: CP value increases when in forward running; decreases when in reverse running b[11]=1: CP value decreases when in forward running; increases when in reverse running
12	b[12]=0: Absolute positioning b[12]=1: Relative positioning
13	b[13]=0: DOG signal rising edge triggered b[13]=1: DOG signal falling edge triggered (Valid for single-speed / 2-speed positioning interruption mode)
14	b[14]=0: Trapezoid curve b[14]=1: S curve

Function Group: Setting up Acyclic E-CAM

Number: D1832, D1833, D1834 (not supported by 10PM)

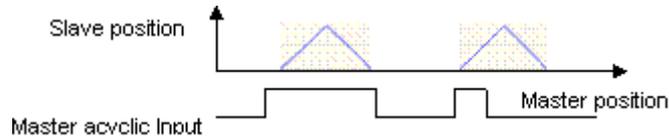
3 Functions of Devices in DVP-PM

Explanations:

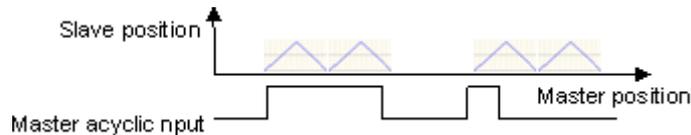
1. D1832 is the register for setting times of CAM repetitions.

D1832 controls the repetition times of CAM data cycle execution. When the value in D1832 exceeds H8000 (b15 = 1), cyclic CAM will start to execute. See the example below. When D1832 = 0, the CAM data will not be executed repeatedly. However, when D1832 = 1, the CAM data will be repeated once, and so forth.

D1832 = 0



D1832 = 1

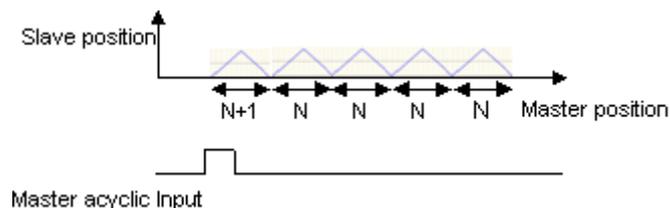


2. D1833 is the register for distributing remaining pulses.

When the operation of the main shaft is fixed to 202 pulses, and 50 pulses are required to execute a cycle of CAM data. The execution result will be 4 CAM data with 2 remaining pulses. In this case, set D1832 to 3 (repeat CAM for 3 times) and D1833 to 2. DVP-PM will execute one more pulse between the first and the second CAM cycle as well as the second and the third CAM cycle, so as to properly distribute the remaining 2 pulses. See example (1) below. When D1832 = 4 and D1833 = 1, the CAM data will be executed 5 times. Between the first and the second CAM cycle, the main shaft (master) will execute one more pulse. In example (2) below, when D1832 = 4 and D1833 = 3, the main shaft (master) will execute one more pulse after the first, second, and third CAM cycle.

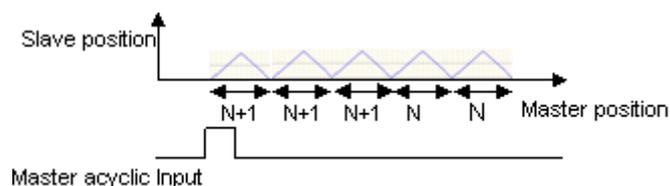
- Example (1)

D1832 = 4, D1833 = 1



- Example (2)

D1832 = 4, D1833 = 3



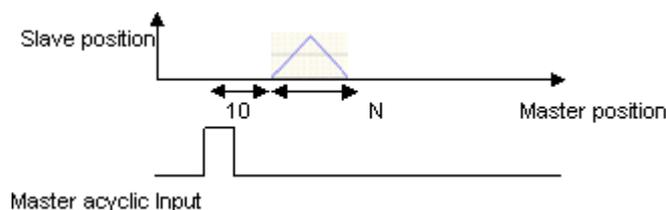
*The value in D1833 cannot be bigger than that in D1832.

3. D1834 is the register for setting pulses for delay start of CAM.

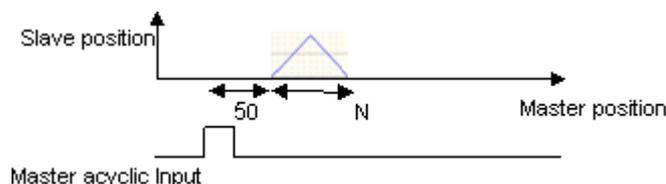
When executing the acyclic CAM, you can perform a delay start of CAM data by setting up value in D1834. The set up value is the number of pulses executed by main shaft between the master acyclic input signal and the start of an CAM cycle. As the examples below, when the system receives a master acyclic input signal, which enables CAM, the main shaft will execute the CAM data after the pulses set in D1834 are executed.

3 Functions of Devices in DVP-PM

D1834 = 10



D1834 = 50



Function Group: Operation Commands for X-Y-Z-A Axis

Number: D1846, D1926, D2006, D2086

Explanations:

Operation commands for X axis: D1846, Y axis: D1926, Z axis: D2006 and A axis: D2086

bit#	X-Y-Z-A operation setting	bit#	X-Y-Z-A operation setting
0	STOP	8	Enabling single-speed positioning mode
1	START	9	Enabling single-speed positioning with additional distance
2	JOG+ operation	10	Enabling 2-speed positioning mode
3	JOG- operation	11	Enabling 2-speed positioning with additional distance
4	Enabling variable speed operation	12	0: Stop OX; 1: Start OX
5*	MPG input operation	13*	Enabling cyclic CAM
6	Enabling zero return mode	14*	Enabling acyclic CAM
7*	Externally triggering single-speed positioning	15	-

*Note:

1. MPG input for 00D: X axis (A0+/A0-, B0+/B0-), Y axis (A1+/A1-, B1+/B1-)
MPG input for 00M: X axis (A0+/A0-, B0+/B0-), Y axis and Z axis (A1+/A1-, B1+/B1-)
2. Only bit 7 of D1926 supports the mode “Externally triggering single-speed positioning.”
3. Only bit 13 and bit 14 of D1846 support the CAM enabling function.

Function Group: Work Modes of X-Y-Z Axis

Number: D1847, D1927, D2007

Explanations:

Work mode setting of X axis: D1847, Y axis: D1927 and Z axis: D2007.

bit#	Work mode of X-Y-Z	bit#	Work mode of X-Y-Z
0	-	8	MASK selection

3 Functions of Devices in DVP-PM

1	-	9	
2	CLR signal output mode	10	
3	CLR output ON/OFF control	11	01: Insert CAM data in multiple axes
4	CLR polarity setting	12	10: Table-Output setting of CAM control
5	STOP mode setting	13	-
6	Range for MPG	14	-
7	LSP/LSN stop mode	15	Returning to default setting

bit#	Explanation
2	When b[2] = 0, CLR will output 130ms signal to the servo as the clear signal when zero return is completed. When b[2] = 1, CLR will be a general output point, and its status will be controlled by ON/OFF status of b[3].
3	When b[3] = 0, output point CLR will be OFF. When b[3] = 1, output point CLR will be ON.
4	b[4] = 0: CLR is NO contact. b[4] = 1: CLR is NC contact.
5	b[5] = 0: When STOP signal comes in during the motor execution, the motor will decelerate to stop. When the next motion instruction is activated, the motor will <u>ignore</u> the unfinished distance and immediately execute the next positioning step. b[5] = 1: When STOP signal comes in during the motor execution, the motor will decelerate to stop. When the next motion instruction is activated, the motor will <u>complete</u> the unfinished distance before executing the next positioning step.
6	b[6] = 0: No limitation on the range of MPG pulse output b[6] = 1: The range for MPG pulse output is limited within P(I) and P(II). When the range is exceeded, the pulse output will decelerate to stop.
7	b[7] = 0: During the running of motor, the motor decelerates to stop when encountering LSP/LSN signal input b[7] = 1: During the running of motor, the motor stops immediately when encountering LSP/LSN signal input.
8	MASK(Stop) settings (single-speed positioning, 2-speed positioning, single-speed positioning interruption, 2-speed positioning interruption) b[10~8] = K0 (000) or other values: MASK function disabled.
9	b[10~8] = K1 (001): Triggering MASK by the rising edge of input signal on terminal ΦA_{\pm}
	b[10~8] = K2 (010): Triggering MASK by the falling edge of input signal on terminal ΦA_{\pm}
10	b[10~8] = K3 (011): Triggering MASK by the rising edge of input signal on terminal ΦB_{\pm}
	b[10~8] = K4 (100): Triggering MASK by the falling edge of input signal on terminal ΦB_{\pm} (Enable the trigger signal of ΦA_{\pm} or ΦB_{\pm} as the stop signal when using a MPG)
11	b[12~11]=K1 (01) : Insert CAM data in multiple axes
12	b[12~11]=K2 (10) : Table-Output setting of CAM control

Function Group: Execution Status of X-Y-Z Axis

Number: D1856, D1936, D2016, D2096

Explanations:

Execution status of X axis: D1856, Y axis: D1936 and Z axis: D2016..

bit#	Execution status of X-Y-Z
0	Forward pulse output in progress
1	Reverse pulse output in progress
2	Operation in progress
3	Error occurs

3 Functions of Devices in DVP-PM

4	Operation pauses
5	Forward MPG input
6	Reverse MPG input
7	-

Function Group: Enabling External Input for X-Y Axis

Number: D1875, D1955

Explanations:

1. When high byte of D1875 / D1955 = H'01, external input is enabled. When high byte of D1875 / D1955 = H'00 external input is disabled.
2. The low byte of D1875 and D1955 indicates the starting number of 4 consecutive external inputs (X), e.g., H'00 indicates X0 ~ X3 is to be enabled corresponding to JOG+, JOG-, MPG and ZRN.
3. Example: D1875 and D1955 = H'0110 refers to JOG+, JOG-, MPG and ZRN is enabled on X10 ~ X13.

Function Group: Close-loop PID control (not supported by DVP-10PM)

Number: D1877, D1878, D1880, D1882, D1888

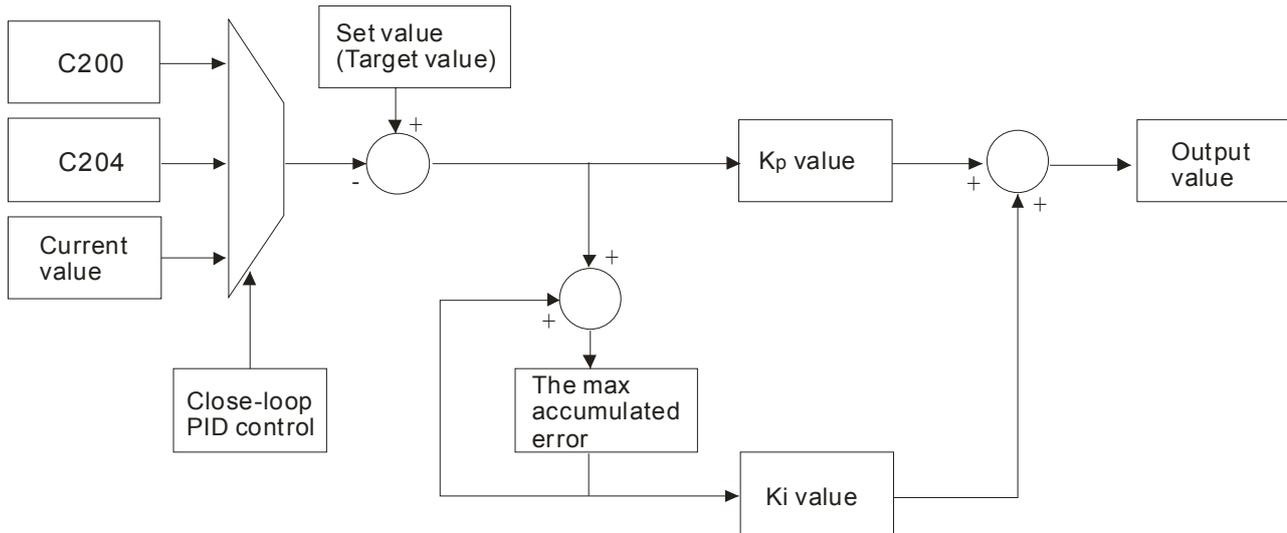
Explanations:

1. D1877 is the register for setting up close-loop PID control. The user can enable and select the feedback source of PID control system.

Value of D1877	Feedback source
1	C200
2	C204
3	Current value (D1848 · D1928 · D2008)
Other	Disable close-loop PID control function

2. D1878, D1879: Set value (target value) of PID control. PID control system adjusts the output value to the set value by referencing the value from feedback source.
3. D1880, D1881: K_p value, i.e. proportional gain for close-loop PID control.
4. D1882, D1883: K_i value, i.e. integral gain for close-loop PID control.
5. D1888, D1889: The tolerable range for error(difference) between input and output value. The value set up in D1888 will limit the accumulated error within the tolerable range.
6. PID control diagram:

3 Functions of Devices in DVP-PM



After the feedback source is selected, the control system adjusts the output value to the set value by comparing between set value and the feedback value. The error (difference) between set value and the feedback value is saved in the error register. The tolerable range for the error can be set up to limit the max error within proper range. If the error exceeds the range, the output will be stopped.

3 Functions of Devices in DVP-PM

3.12 Special Registers for Manual Mode Settings

Below are the types and functions of special registers (special D) for motion modes. See the next section for more details on the functions. You can set the parameters according to explanations of each register and also know more about the system information by these registers.

Special D								Content	Range	Default
X axis		Y axis		Z axis		A axis				
HW*1	LW*1	HW	LW	HW	LW	HW	LW			
	D1816		D1896		D1976		D2056	Parameter setting	b0 ~ b15	H0
D1819	D1818	D1899	D1898	D1979	D1978	D2059	D2058	Pulses per revolution (PPR) of motor	1 ~ +2,147,483,647 PLS/REV	K2,000
D1821	D1820	D1901	D1900	D1981	D1980	D2061	D2060	Displacement per revolution (DPR) of motor	1 ~ +2,147,483,647 *2	K1,000
D1823	D1822	D1903	D1902	D1983	D1982	D2063	D2062	Maximum speed	0 ~ +2,147,483,647 *3	K500,000
D1825	D1824	D1905	D1904	D1985	D1984	D2065	D2064	Bias speed: V _{BIAS}	0 ~ +2,147,483,647 *3	K0
D1827	D1826	D1907	D1906	D1987	D1986	D2067	D2066	JOG speed: V _{JOG}	0 ~ +2,147,483,647 *3	K5,000
D1829	D1828	D1909	D1908	D1989	D1988	D2069	D2068	Zero return speed: V _{RT}	0 ~ +2,147,483,647 *3	K50,000
D1831	D1830	D1911	D1910	D1991	D1990	D2071	D2070	Zero return deceleration speed: V _{CR}	0 ~ +2,147,483,647 *3	K1,000
	D1832		D1912		D1992		D2072	Number of PG0 signals: N	0 ~ +32,767 PLS	K0
	D1833		D1913		D1993		D2073	Number of pulses for zero return: P	-32,768 ~ +32,767 PLS	K0
D1835	D1834	D1915	D1914	D1995	D1994	D2075	D2074	Definition of zero point: HP	0 ~ ±999,999 *2	K0
	D1836		D1916		D1996		D2076	Acceleration time: T _{ACC}	10 ~ +32,767 ms	K100
	D1837		D1917		D1997		D2077	Deceleration time: T _{DEC}	10 ~ +32,767 ms	K100
D1839	D1838	D1919	D1918	D1999	D1998	D2079	D2078	Target position (I): P(I)	-2,147,483,648 ~ +2,147,483,647 *2	K0
D1841	D1840	D1921	D1920	D2001	D2000	D2081	D2080	Operation speed (I): V(I)	0 ~ +2,147,483,647 *2	K1,000
D1843	D1842	D1923	D1922	D2003	D2002	D2083	D2082	Target position (II): P(II)	-2,147,483,648 ~ +2,147,483,647 *2	K0
D1845	D1844	D1925	D1924	D2005	D2004	D2085	D2084	Operation speed (II): V(II)	0 ~ +2,147,483,647 *2	K2,000
	D1846		D1926		D2006		D2086	Operation commands	b0 ~ b15	H0
	D1847		D1927		D2007		D2087	Work modes	b0 ~ b15	H0
D1849	D1848	D1929	D1928	D2009	D2008	D2089	D2088	Current position: CP (pulse)	-2,147,483,648 ~ +2,147,483,647 *2	K0
D1851	D1850	D1931	D1930	D2011	D2010	D2091	D2090	Current speed: CS (pulse)	0 ~ +2,147,483,647 PPS	K0
D1853	D1852	D1933	D1932	D2013	D2012	D2093	D2092	Current position: CP (unit *3)	-2,147,483,648 ~ +2,147,483,647 *2	K0
D1855	D1854	D1935	D1934	D2015	D2014	D2095	D2094	Current speed: CS (unit *3)	0 ~ +2,147,483,647 PPS	K0
	D1856		D1936		D2016		D2096	Execution status	b0 ~ b15	H0
	D1857		D1937		D2017		D2097	Error code	See the error code table	H0

3 Functions of Devices in DVP-PM

Special D								Content	Range	Default
X axis		Y axis		Z axis		A axis				
HW*1	LW*1	HW	LW	HW	LW	HW	LW			
	D1858		D1938		D2018		D2098	Electronic gear ratio (numerator)	1 ~ +32,767	K1
	D1859		D1939		D2019		D2099	Electronic gear ratio (denominator)	1 ~ +32,767	K1
D1861	D1860	D1941	D1940	D2021	D2020	D2101	D2100	MPG input frequency	Pulse frequency by MPG input	K0
D1863	D1862	D1943	D1942	D2023	D2022	D2103	D2102	Number of accumulated MPG input pulses	Number of input pulses from MPG	K0
	D1864		D1944		D2024		D2104	Response time of MPG input	Response time of MPG input	K5
D1865								Stop mode (OX0-99)	Depending on the demand	K0
D1867	D1866	D1947	D1946	D2027	D2026	D2107	D2106	Coordinate of electrical zero point	Depending on the demand	K0
D1868								Number of OX subroutine	Depending on the demand	K0
D1869								Error step of OX subroutine	Depending on the demand	K0
D1872								Enabling Y output when OX is ready High byte: K1; low byte: designating start No. of Y output	Depending on the demand	K0
D1873								Enabling Y output when OX executes M-code High byte: K1; low byte: designating start No. of Y output	Depending on the demand	K0
D1874								No. of input point X for clearing OX M-code status	Depending on the demand	K0
D1875		D1955						Enable external input (ZRN, MPG, JOG-, JOG+)	Depending on the demand	K0

*1: HW: High word; LW: Low word

*2: Units available: $\mu\text{m}/\text{rev}$, $\text{m deg}/\text{rev}$ and 10^{-4} inch/rev.

*3: The unit setting follows the settings of b0 and b1 of D1816, D1896 and D1976 (Parameter setting)

3.12.1 Functions of Special Registers for Manual Mode Settings

X axis		Y axis		Z axis		A axis		Parameter Setting
HW	LW	HW	LW	HW	LW	HW	LW	
	D1816		D1896		D1976		D2056	

See the tables below for the explanations of b0 ~ b15.

■ **b0 and b1 of D1816 (D1896, D1976, D2056): setting of unit**

b1	b0	Unit	Explanation
0	0	Motor	Unit: pulse
0	1	Machine	Unit: length, angle
1	0	Combined	Unit for position: length, angle (machine)
1	1		Unit for speed: pulse (motor)

	Motor Unit	Combined unit	Machine Unit
Position	pulse	um	
	pulse	m deg	
	pulse	10-4inch	
Speed	pulse/sec		cm/min
	pulse/sec		10deg/min
	pulse/sec		inch/min

- Position data: zero point position (HP), target position (I) (P(I)), target position (II) (P(II)), current position (CP).
- Speed data: maximum speed (V_{MAX}), bias speed (V_{BIAS}), JOG speed (V_{JOG}), zero return speed (V_{RT}), zero return deceleration speed (V_{CR}), operation speed (I) ($V(I)$), operation speed (II) ($V(II)$).

• **Example 1:**

b[1:0] = 00: Motor unit ⇒ unit of position: pulse; unit of speed: pulse/sec (PPS)

Assume: target position P(I): 10,000 (pulse); operation speed V(I): 10K (PPS)

Explanation: When the positioning controller sends out 10,000 pulses (frequency at 10K PPS), the target position can be achieved. The displacement moved by every pulse is calculated by users according to the parameter of connected devices.

• **Example 2:**

b[1:0] = 01: Machine unit ⇒ unit of position: um; unit of speed: cm/min

Assume: D1818 (D1898/D1978) = 1,000 (Pulse/REV), D1820 (D1900/D1980) = 100 (um/REV), target position P(I) = 10,000 (um), and operation speed V(I) = 6 (cm/min). Solve the number of pulses and the frequency required by pulse instruction of the positioning controller.

Solution:

$$1. \quad \text{Total Distance } P(I) = \underbrace{\frac{\text{Distance}}{\text{Revolution}}}_B \times \underbrace{\frac{\text{Revolution}}{\text{Number of pulses}}}_{\frac{1}{A}} \times \text{Total number of pulses}$$

A: Pulses per revolution (PPR) of motor; **B:** Displacement per revolution (DPR) of motor

3 Functions of Devices in DVP-PM

$$\text{Total number of pulses required} = \frac{P(I)}{B/A} = P(I) \times \frac{A}{B} = 100,000 \text{ pulses}$$

2. Operation speed V(I): 6 (cm/min) = 60,000/60 (um/sec)

$\text{Speed(Frequency)} = \frac{\text{Distance}}{\text{Time}} = \frac{\text{Distance}}{\underbrace{\text{Revolution}}_B} \times \frac{\text{Revolution}}{\underbrace{\text{Number of pulses}}_{1/A}} \times \frac{\text{Number of pulses}}{\underbrace{\text{Time}}_{\text{PPS, pulse/sec}}}$
--

Calculate the by the positioning controller

$$\text{Frequency (PPS)} = V(I) \times \frac{A}{B} = \frac{60,000}{60} \times \frac{1,000}{100} = 10,000 \text{ PPS}$$

• **Example 3:**

b[1:0] = 01 = 10, 11 ⇒ Combined unit: unit for position: um; unit for speed: pulse/sec (PPS)

Assume: D1818 (D1898/D1978) = 2,000 (Pulse/REV), D1820 (D1900/D1980) = 100 (um/REV), target position P(I): 10,000 (um), and operation speed V(I): 10K (PPS). Solve the number of pulses required by pulse instruction of the positioning controller.

Solution:

$$\text{Total number of pulses required} = \frac{P(I)}{B/A} = P(I) \times \frac{A}{B} = 200,000 \text{ pulses}$$

■ **b2 and b3 of D1816 (D1896, D1976, D2056): setting of multiplication of position data**

The position data, i.e. zero point position (HP), target position (I) (P(I)), target position (II) (P(II)), current position (CP), have to be multiplied by the multiplication values listed in the table below.

b3	b2	Multiplication
0	0	Position data × 10 ⁰
0	1	Position data × 10 ¹
1	0	Position data × 10 ²
1	1	Position data × 10 ³

■ **b4 and b5 of D1816 (D1896, D1976, D2056): pulse output type**

b5	b4	Pulse output type (positive logic)	Explanation
0	0	FP forward pulses RP reverse pulses	Double-phase
0	1	FP pulses RP direction (DIR)	Single-phase
1	0	FPA-phase pulses 	A/B phase



3 Functions of Devices in DVP-PM

b5	b4	Pulse output type (positive logic)	Explanation
1	1		

■ **b6 of D1816 (D1896, D1976, D2056): PWM mode (for DVP-10PM only)**

B[6] = 1, PWM executes.

- (1) When the operation command “JOG+ operation” is enabled, PWM will be executed on Y0~Y3.
- (2) When the operation command “single speed positioning” is enabled, single speed positioning will be executed on Y0~Y3 with single-phase pulses.
- (3) PWM pulse width setting: D1838, D1918, D1998, D2078
- (4) PWM cycle setting: D1842, D1922, D2002, D2082

■ **b8 of D1816 (D1896, D1976, D2056): zero return direction**

b[8] = 0: Zero return executes in positive direction, i.e. current position(CP) increases.

b[8] = 1: Zero return executes in negative direction, i.e. current position(CP) decreases.

■ **b9 of D1816 (D1896, D1976, D2056): zero return mode**

b[9] = 0: normal mode. When DOG signal is generated, after the specified PG0 signals (N) then the following pulse signals (P), the motor will stop immediately.

b[9] = 1: overwrite mode. When DOG signal is generated, the motor will stop according to either PG0 signals (N) or pulse signals (P), which is finished first.

■ **b10 of D1816 (D1896, D1976, D2056): DOG signal rising/falling edge triggered**

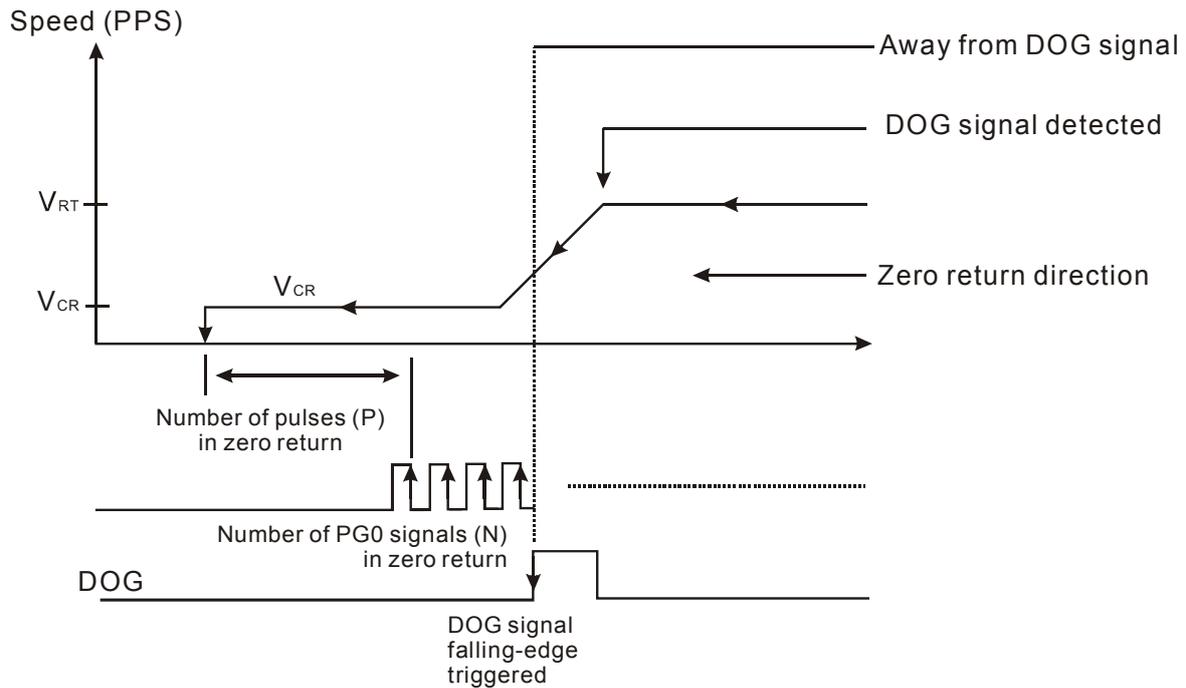
b[10] = 0: DOG signal falling edge triggered

b[10] = 1: DOG signal rising edge triggered

- b [9:10] = 00: normal mode; DOG signal falling-edge triggered

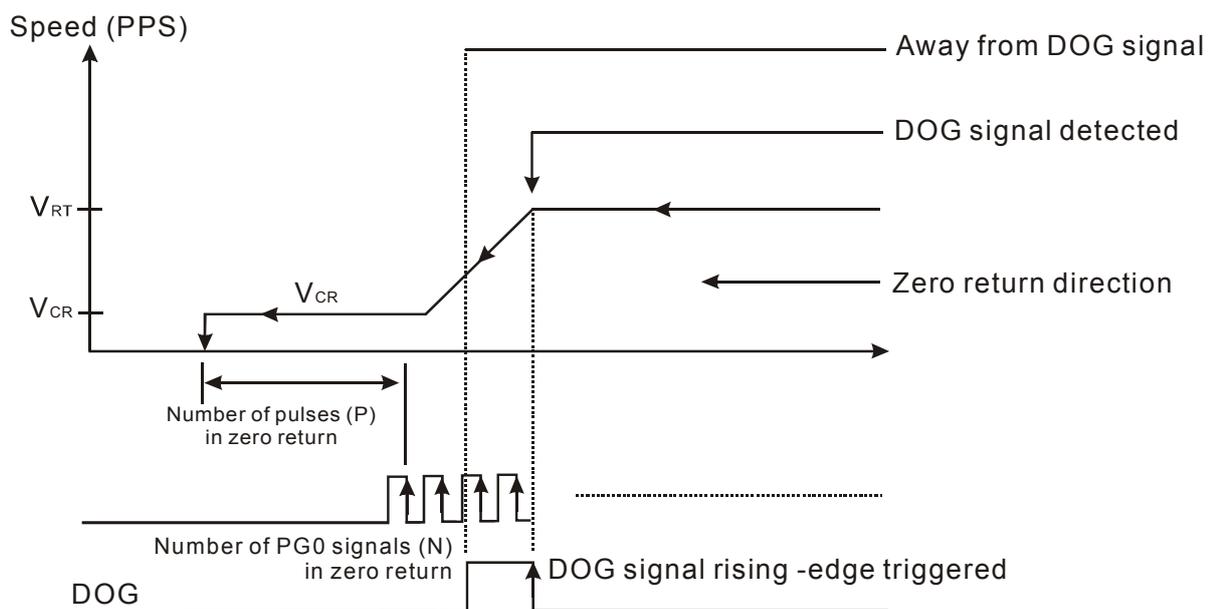
1. Zero return: The motor operates at zero return speed V_{RT} . When it encounters DOG signal, it will decelerate to zero return deceleration speed V_{CR} . If the falling edge of DOG signal is triggered, the motor will stop after the specified PG0 signals (N) plus the specified pulse signals (P).
2. If the specified N or P is too small, when the falling edge of DOG signal is triggered, the motor will stop immediately after the specified PG0 signals (N) plus the specified pulse signals (P), even if the zero return deceleration speed is not yet reached.
3. If N and P are both set as “0”, the motor will stop immediately when the falling-edge of DOG signal is triggered.

3 Functions of Devices in DVP-PM



- **b[9:10] = 01**: normal mode; DOG signal rising-edge triggered

1. Zero return: The motor operates at zero return speed V_{RT} . When it encounters DOG signal, it will decelerate to zero return deceleration speed V_{CR} . At the same time, rising edge of DOG signal is triggered to enable the specified PG0 signals (N) plus the specified pulse signals (P). After both N signal and P pulses are finished, the motor stops.
2. If the specified N or P is too small, when the rising edge of DOG signal is triggered, the motor will stop immediately after the specified PG0 signals (N) plus the specified pulse signals (P), even if the zero return deceleration speed (V_{CR}) is not yet reached.
3. If N and P are both set as "0", the motor will stop immediately when the rising-edge of DOG signal is triggered.



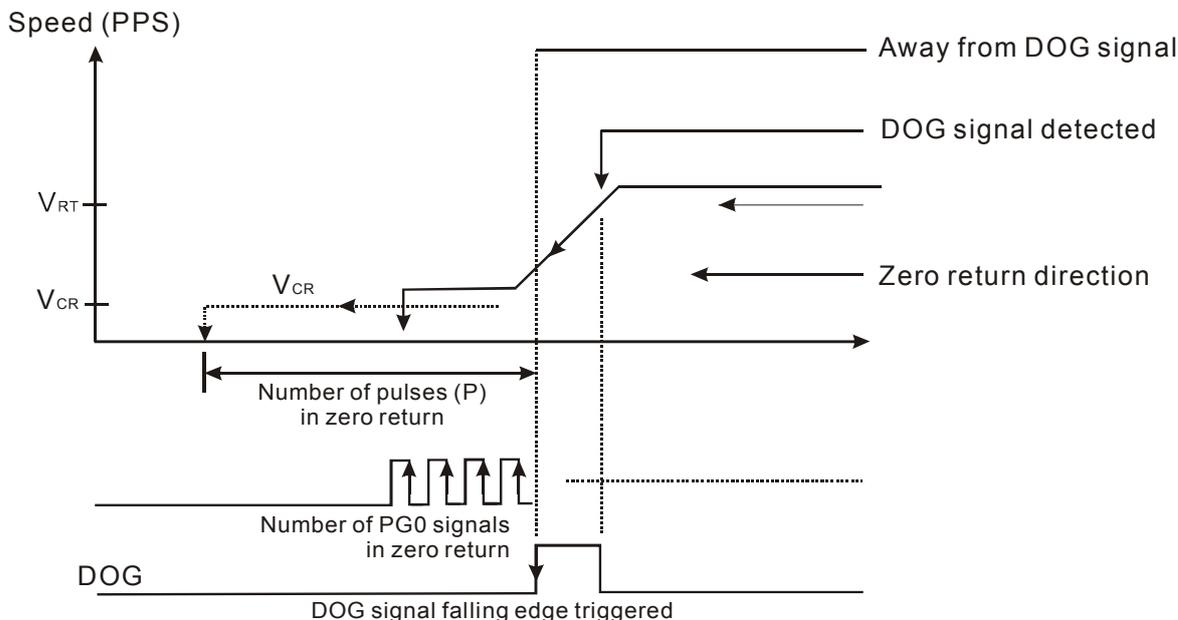
- **b[9:10] = 10**: overwrite mode; DOG signal falling-edge triggered

1. Zero return: The motor operates at zero return speed V_{RT} . When it encounters DOG signal, it will decelerate to

3 Functions of Devices in DVP-PM

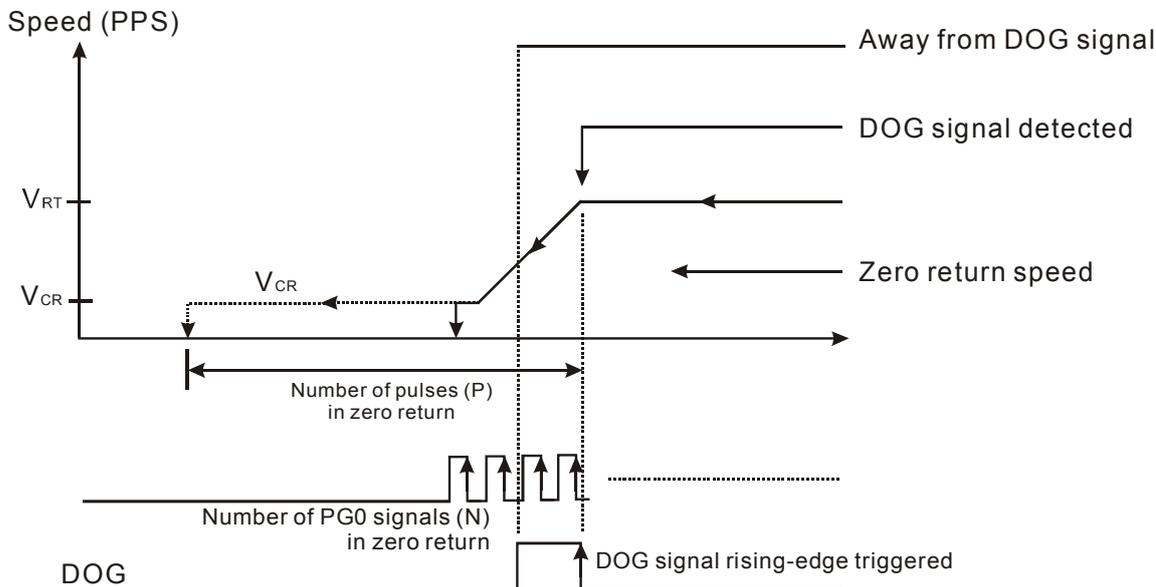
zero return deceleration speed V_{CR} . When the falling edge of DOG signal is triggered, the motor will stop according to either PG0 signals (N) or pulse signals (P), of which is finished first.

2. If the specified N or P is too small, when the falling edge of DOG signal is triggered, the motor will stop immediately after the specified PG0 signals (N) or the specified pulse signals (P) of which is finished first, even if the zero return deceleration speed is not yet reached..
3. If N and P are both set as "0", the motor will stop immediately when the falling-edge of DOG signal is triggered.



• **b[9:10] = 11**: overwrite mode; DOG signal rising-edge triggered

1. Zero return: The motor operates at zero return speed V_{RT} . When it encounters DOG signal, it will decelerate to zero return deceleration speed V_{CR} . At the same time, rising edge of DOG signal is triggered to enable the specified PG0 signals (N) and the specified pulse signals (P). The motor will stop according to either PG0 signals (N) or pulse signals (P), of which is finished first.
2. If the specified N or P is too small, when the rising edge of DOG signal is triggered, the motor will stop immediately according to either PG0 signals (N) or pulse signals (P), of which is finished first.
3. If N and P are both set as "0", the motor will stop immediately when the rising-edge of DOG signal is triggered



3 Functions of Devices in DVP-PM

■ **b11 of D1816 (D1896, D1976, D2056): Reverse the displacement direction in same pulse output polarity**

b[11] = 0: CP value increases when in forward running; CP value decreases when in reverse running

b[11] = 1: CP value decreases when in forward running; CP value increases when in reverse running

■ **b12 of D1816 (D1896, D1976, D2056): absolute/relative positioning setting**

b[12] = 0: absolute positioning

b[12] = 1: relative positioning

■ **b13 of D1816 (D1896, D1976, D2056): DOG signal rising/falling edge triggered**

(For single-speed positioning interruption and 2-speed positioning interruption)

b[13] = 0: DOG signal rising-edge triggered

b[13] = 1: DOG signal falling-edge triggered

■ **b14 of D1816 (D1896, D1976, D2056): acceleration/deceleration curve selection**

b[14] = 0: trapezoid curve

b[14] = 1: S curve

X axis		Y axis		Z axis		A axis		Pulses Per Revolution (PPR) (A)
HW	LW	HW	LW	HW	LW	HW	LW	
D1819	D1818	D1899	D1898	D1979	D1978	D2059	D2058	

- Users can set up electronic gear ratio for general servo drives. Therefore, the pulses per revolution (A) set up in PLC may not equal to the actual pulses generated by the encoder for servo motor.

$$\text{PPR(A)} \times \text{electronic gear ratio (CMX/CDV)} = \text{PPR generated by encoder}$$

- The system unit varies according to the settings of b0 and b1 in D1816 (D1896, D1976, D2056). Parameter (A) is valid only when the system unit is set to be machine unit or combined unit. Parameter A is invalid when the system unit is set as motor unit.

X axis		Y axis		Z axis		A axis		Displacement Per Revolution (DPR) (B)
HW	LW	HW	LW	HW	LW	HW	LW	
D1821	D1820	D1901	D1900	D1981	D1980	D2061	D2060	

- The system unit varies according to the settings of b0 and b1 in D1816 (D1896, D1976). There are three units available for displacement per revolution. Valid range: 1 ~ +2,147,483,647 (um/Rev, mdeg/Rev, 10⁻⁴ inch/Rev)
- Parameter (B) is valid only when the system unit is set to be machine unit or combined unit. Parameter (B) is invalid when the system unit is set as motor unit.

X axis		Y axis		Z axis		A axis		Maximum Speed (V _{MAX})
HW	LW	HW	LW	HW	LW	HW	LW	
D1823	D1822	D1903	D1902	D1983	D1982	D2063	D2062	

- V_{MAX} is the maximum speed for all kinds of operation modes. Range: 0 ~ +2,147,483,647; the unit is set by b0 and b1 of D1816 (D1896, D1976, D2056).
- V_{MAX} should be specified within the valid speed range of motion instructions: 10 ~ 500kPPS. If the value is bigger than 500k, the output will be set as 500k; if the value is smaller than 10, the output will be set as 10.

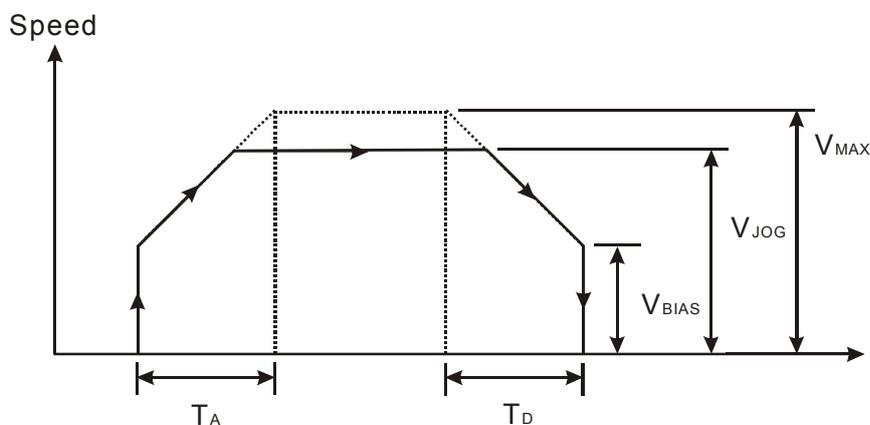
3 Functions of Devices in DVP-PM

X axis		Y axis		Z axis		A axis		Bias Speed (V_{BIAS})
HW	LW	HW	LW	HW	LW	HW	LW	
D1825	D1824	D1905	D1904	D1985	D1984	D2065	D2064	

- V_{BIAS} is the start speed for pulse output. Range: 0 ~ +2,147,483,647; the unit is set by b0 and b1 of D1816 (D1896, D1976, D2056).
- V_{BIAS} should be specified within the valid bias speed range of motion instructions: 0 ~ 500kPPS. If the value is bigger than 500k, the output will be set as 500k; if the value is smaller than 0, the output will be set as 0.
- If you apply a step drive system, please pay attention to the resonance frequency of the step motor. Set the bias speed higher than the resonance frequency for a safe startup.

X axis		Y axis		Z axis		A axis		JOG Speed (V_{JOG})
HW	LW	HW	LW	HW	LW	HW	LW	
D1827	D1826	D1907	D1906	D1987	D1986	D2067	D2066	

- Range: 0 ~ +2,147,483,647; the unit is set by b0 and b1 of D1816 (D1896, D1976, D2056).
- V_{JOG} should be specified within the valid speed range of motion instructions: 10 ~ 500kPPS. If the value is bigger than 500k, the output will be set as 500k; if the value is smaller than 10, the output will be set as 10.
- Setup range limitation: $V_{MAX} > V_{JOG} > V_{BIAS}$. If $V_{JOG} > V_{MAX}$, set value of V_{JOG} will be same as V_{MAX} . If $V_{JOG} < V_{BIAS}$, set value of V_{JOG} will be same as V_{BIAS} .
- V_{JOG} cannot be modified during the execution.



X axis		Y axis		Z axis		A axis		Zero Return Speed V_{RT}
HW	LW	HW	LW	HW	LW	HW	LW	
D1829	D1828	D1909	D1908	D1989	D1988	D2069	D2068	

- V_{RT} is the speed for returning to mechanical zero point. Range: 0 ~ +2,147,483,647; the unit is set by b0 and b1 of D1816 (D1896, D1976, D2056).
- V_{RT} should be specified within the valid speed range of motion instructions: 10 ~ 500kPPS. If the value is bigger than 500k, the output will be set as 500k; if the value is smaller than 10, the output will be set as 10.
- Setup range limitation: $V_{MAX} > V_{RT} > V_{BIAS}$
- V_{RT} cannot be modified during the execution.

3 Functions of Devices in DVP-PM

X axis		Y axis		Z axis		A axis		Zero Return Deceleration Speed V_{CR}
HW	LW	HW	LW	HW	LW	HW	LW	
D1831	D1830	D1911	D1910	D1991	D1990	D2071	D2070	

1. Range: 0 ~ +2,147,483,647; the unit is set by b0 and b1 of D1816 (D1896, D1976, D2056).
2. V_{CR} should be specified within the valid speed range of motion instructions: 10 ~ 500kPPS. If the value is bigger than 500k, the output will be set as 500k; if the value is smaller than 10, the output will be set as 10.
3. When zero return is executed, the motor will operate at zero return speed V_{RT} . When DOG signal is triggered, the motor will decelerate to zero return deceleration speed V_{CR} .
4. To perform precise positioning result at zero point, we suggest you set up V_{CR} in low speed.
5. V_{CR} cannot be modified during the execution.

X axis		Y axis		Z axis		A axis		Number of Zero Point Signals (PG0) (N)
HW	LW	HW	LW	HW	LW	HW	LW	
	D1832		D1912		D1992		D2072	

1. Range: -32,768 ~ 32,767 (pulse). Positive values indicate the number of positive PG0 signals (N). Negative values indicate the number of negative PG0 signals (N).
2. For details about the reference signals for zero return (N, P), please refer to explanations of b9, 10 (zero return mode) of D1816 (D1896, D1976, D2056).

X axis		Y axis		Z axis		A axis		Number of Pulses for Zero Return (P)
HW	LW	HW	LW	HW	LW	HW	LW	
	D1833		D1913		D1993		D2073	

1. Range: -32,768 ~ 32,767 (pulse). Positive values indicate the number of positive zero return pulses (P). Negative values indicate the number of negative zero return pulses (P).
2. For details about the reference signals for zero return (N, P), please refer to explanations of b9, 10 (zero return mode) of D1816 (D1896, D1976, D2056)

X axis		Y axis		Z axis		A axis		Definition of Zero Point (HP)
HW	LW	HW	LW	HW	LW	HW	LW	
D1835	D1834	D1915	D1914	D1995	D1994	D2075	D2074	

1. Range: 0 ~ ±999,999; the unit is set by b0 and b1 of D1816 (D1896, D1976).
2. After zero return is completed, current position (CP) will be updated with zero point (HP).

X axis		Y axis		Z axis		A axis		Acceleration Time T_{ACC}
HW	LW	HW	LW	HW	LW	HW	LW	
	D1836		D1916		D1996		D2076	

1. T_{ACC} is the time required from bias speed V_{BIAS} (D1824, D1904, D1984, D2064) to maximum speed V_{MAX} (D1822, D1902, D1982, D2062).
2. If the value is smaller than 10ms, the output will be set as 10ms; if the value is bigger than 32767 ms, the output will be set as 32,767 ms.
3. If you need a complete S curve, please set the operation speed (V) as maximum speed V_{MAX} .

3 Functions of Devices in DVP-PM

X axis		Y axis		Z axis		A axis		Deceleration Time T_{DEC}
HW	LW	HW	LW	HW	LW	HW	LW	
	D1837		D1917		D1997		D2077	

- T_{DEC} is the time required from maximum speed V_{MAX} (D1822, D1902, D1982, D2062) to bias speed V_{BIAS} (D1824, D1904, D1996, D2064).
- If the value is smaller than 10ms, the output will be set as 10ms; if the value is bigger than 32767 ms, the output will be set as 32,767 ms.
- If you need a complete S curve, please set the operation speed (V) as maximum speed V_{MAX} .

X axis		Y axis		Z axis		A axis		Target Position (I) (P(I))
HW	LW	HW	LW	HW	LW	HW	LW	
D1839	D1838	D1919	D1918	D1999	D1998	D2079	D2078	

- Range: -2,147,483,648 ~ +2,147,483,647; the unit is set by b0 and b1 of D1816 (D1896, D1976).
- Attribute of target position P(I):
 - ◆ Absolute coordinate: b12 of D1816 (D1896, D1976, D2056) = 0
Absolute coordinate is the coordinate starting from "0." When the target position P(I) > current position (D1848, D1928, D2008, D2088), the motor will conduct forward running. When the target position P(I) < current position, the motor will conduct reverse running.
 - ◆ Relative coordinate: b12 of D1816 (D1896, D1976, D2056) = 1
Relative coordinate is the relative displacement moved from current position (D1848, D1928, D2008, D2088). When the relative coordinate is a positive value, the motor will conduct forward running. When the relative coordinate is a negative value, the motor will conduct reverse running.
- The multiplication of position data P(I) varies according to the settings of b2 and b3 in D1816 (D1896, D1976, D2056).

X axis		Y axis		Z axis		A axis		Operation Speed (I) (V(I))
HW	LW	HW	LW	HW	LW	HW	LW	
D1841	D1840	D1921	D1920	D2001	D2000	D2081	D2080	

- Range: 0 ~ +2,147,483,647; the unit is set by b0 and b1 of D1816 (D1896, D1976, D2056).
- $V(I)$ should be specified within the valid speed range of motion instructions: 10 ~ 500kPPS. If the value is bigger than 500k, the output will be set as 500k; if the value is smaller than 10, the output will be set as 10.
- Setup range limitation: $V_{MAX} > V(I) > V_{BIAS}$.
- When b4 of D1846 (D1926, D2006, D2086) = 1, which enables variable speed operation, the operation speed $V(I)$ can be modified during operation. When the sign of $V(I)$ is "+", the motor will conduct forward running; when the sign of $V(I)$ is "-", the motor will conduct reverse running.

X axis		Y axis		Z axis		A axis		Target Position (II) (P(II))
HW	LW	HW	LW	HW	LW	HW	LW	
D1843	D1842	D1923	D1922	D2003	D2002	D2083	D2082	

- Range: -2,147,483,648 ~ +2,147,483,647; the unit is set by b0 and b1 of D1816 (D1896, D1976, D2056).

3 Functions of Devices in DVP-PM

2. Attribute of target position P(II):

- ◆ Absolute coordinate: b12 of D1816 (D1896, D1976, D2056) = 0

Absolute coordinate is the coordinate starting from “0.” When the target position P(II) > current position (D1848, D1928, D2008, D2088), the motor will conduct forward running. When the target position P(II) < current position, the motor will conduct reverse running.

- ◆ Relative coordinate: b12 of D1816 (D1896, D1976, D2056) = 1

Relative coordinate is the relative displacement moved from current position (D1848, D1928, D2008, D2088). When the relative coordinate is a positive value, the motor will conduct forward running. When the relative coordinate is a negative value, the motor will conduct reverse running..

3. The multiplication of position data P(II) varies according to the settings of b2 and b3 in D1816 (D1896, D1976, D2056).

X axis		Y axis		Z axis		A axis		Operation Speed (II) (V(II))
HW	LW	HW	LW	HW	LW	HW	LW	
D1845	D1844	D1925	D1924	D2005	D2004	D2085	D2084	

1. Range: 0 ~ +2,147,483,647; the unit is set by b0 and b1 of D1816 (D1896, D1976, D2056).

2. V(I) should be specified within the valid speed range of motion instructions: 10 ~ 500kPPS. If the value is bigger than 500k, the output will be set as 500k; if the value is smaller than 10, the output will be set as 10.

3. Setup range limitation: $V_{MAX} > V(II) > V_{BIAS}$.

X axis		Y axis		Z axis		A axis		Operation Commands
HW	LW	HW	LW	HW	LW	HW	LW	
	D1846		D1926		D2006		D2086	

1. b0 of D1846 (D1926, D2006, D2086): STOP

- ◆ When b0 is rising edge triggered (0→1), the positioning controller decelerate to stop.
- ◆ The function is the same as the external STOP input signal. The positioning controller will decelerate to stop in both situations..

2. b1 of D1846 (D1926, D2006, D2086): START

- ◆ When b1 is rising edge triggered (0→1), the positioning controller will operates according to operation conmmands set in D1846 (D1926, D2006, D2086).

3. b2 of D1846 (D1926, D2006, D2086): enabling JOG+

- ◆ b[2] = 1: JOG+ sends out forward pulses (CW)
- ◆ When PWM mode is enabled (Parameter Setting, D1816/D1896/D1976/D2056, b[6] = 1), and the operation command “JOG+ operation” is enabled (b[2] = 1), PWM will be executed on Y0~Y3.

4. b3 of D1846 (D1926, D2006, D2086): enabling JOG-

- ◆ b[3] = 1: JOG- sends out reverse pulses (CCW)

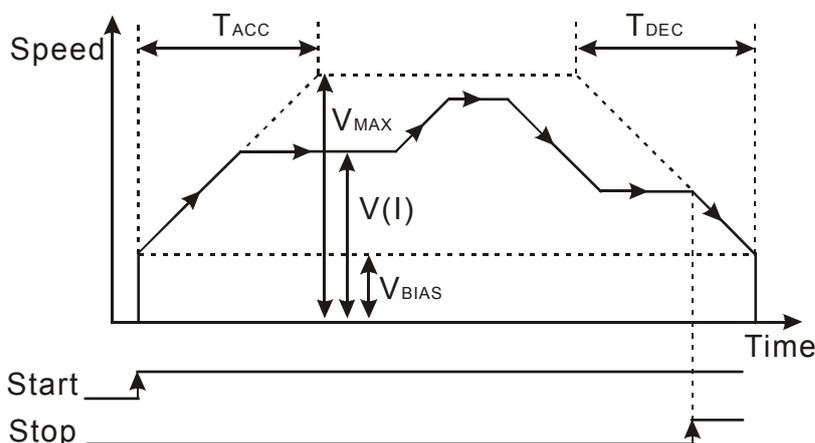
5. b4 of D1846 (D1926, D2006, D2086): variable speed operation

- ◆ When b[4] = 1 and operation is started (START terminal is ON), DVP-PM will start to send out pulses for performing variable speed operation on positioning controller.
- ◆ The operation speed will be stable after the speed is accelerated from V_{BIAS} to the expected V(I). When

3 Functions of Devices in DVP-PM

operation speed achieves stable status, you can modify $V(I)$, and the pulse output from DVP-PM will accelerate or decelerate according to the modification. In this case, the external STOP input signal cannot stop the pulse output from DVP-PM. To stop the pulse output, you have to trigger the STOP function (b0 of D1846 (D1926, D2006).

- ◆ Timing diagram:



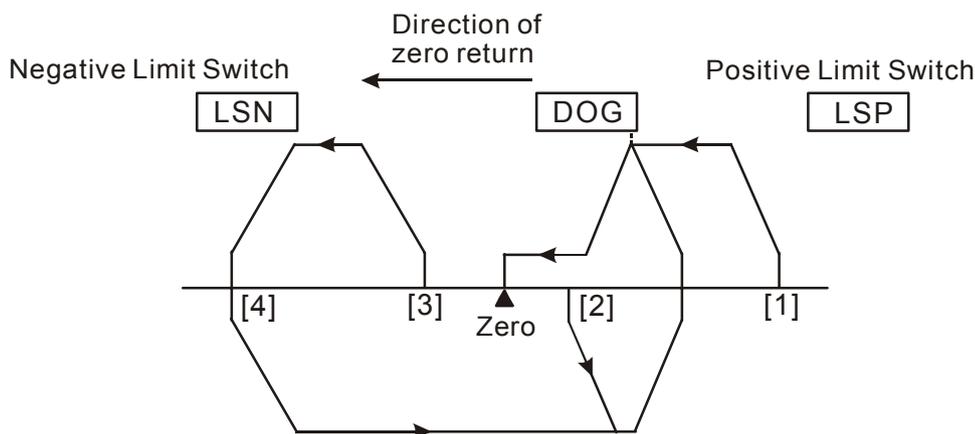
- b5 of D1846 (D1926, D2006, D2086): manual pulse generator (MPG) input

- ◆ b[5] = 1: enabling MPG input. See D1858 ~ D1864 (D1938 ~ D1944, D2018 ~ D2024) for more details.

- b6 of D1846 (D1926, D2006): enabling zero return mode

- ◆ b[6] = 0→1: starting zero return. The motions of zero return vary depending on different current positions (CP)

Zero return path:



CP [1]: On the right of zero point and DOG sensor; DOG = OFF.

CP [2]: On the right of zero point; DOG = ON.

CP [3]: On the left of zero point and DOG sensor; DOG = OFF, LSN = OFF.

CP [4]: On the left of zero point; DOG = OFF, LSN = ON.

Note: DVP-10PM do not have LSN/LSP signals, therefore the motions on CP[3] and CP[4] are not available.

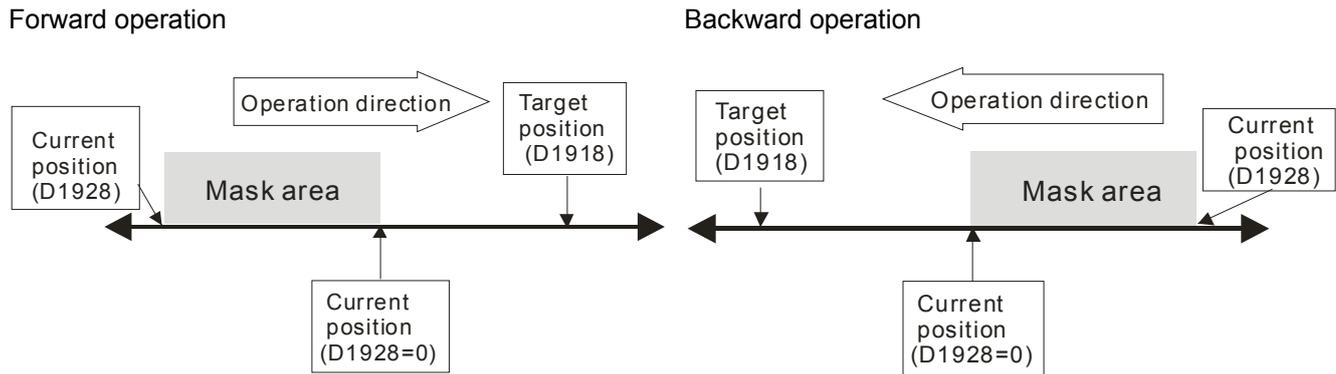
- b7 of D1926: Externally triggering single-speed positioning

- ◆ b[7] = 0→1: Enabling single-speed positioning triggered by external signal(DOG1). When b7 of D1926 is ON and DOG1 is triggered, single-speed positioning will be executed. In this case, a mask area will

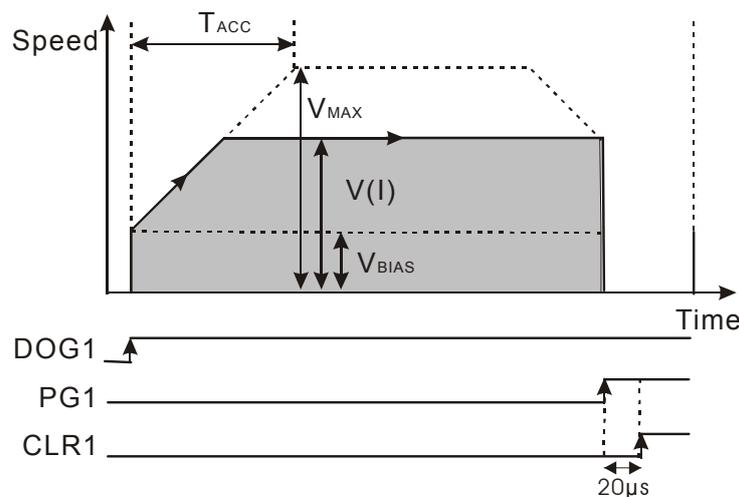
3 Functions of Devices in DVP-PM

automatically be defined, and PG1 signal which stops the positioning immediately is only valid when it is out of the mask area. When PG1 is triggered, the clear signal CLR1 will be triggered automatically after 20 μ s. In addition, the positioning speed is operated according to V(I), and the pulses are sent out by the pulse generator.

- ◆ The mask area is defined by current position and the target position as below diagrams. If the current position is smaller than the target position, the positioning will be operated in forward direction. If the current position is bigger than the target position, the positioning will be operated in backward direction. In both cases, the mask area will be the range between current position and 0.



- ◆ Operation direction: Direction of relative positioning is determined by the sign bit of the value for P(I) (set in D1838, D1918, D1998). Direction of absolute positioning is determined by current position (CP). Forward running occurs when P(I) is bigger than CP. Reverse running occurs when P(I) is smaller than CP.
- ◆ The operation speed will be stable after the speed is accelerated from V_{BIAS} to the expected V(I). When PG1 is triggered out of the mask area, the positioning stops immediately.
- ◆ The registers involved: D1824 (D1904, D1984) (V_{BIAS}), D1840 (D1920, D2000) (V(I)), D1822 (D1902, D1982) (V_{MAX}), D1838 (D1918, D1998) (P(I)) and D1836 (D1916, D1996) (T_{ACC})



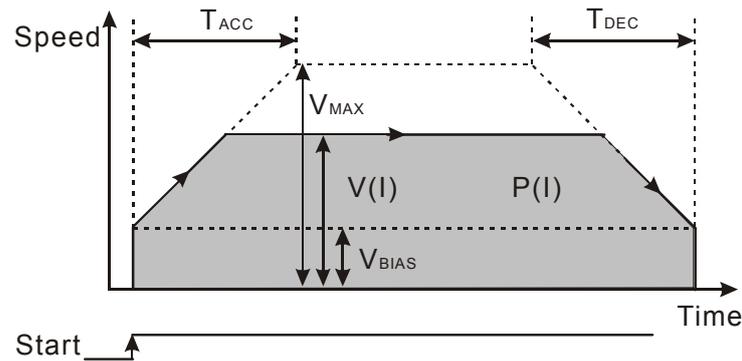
9. b8 of D1846 (D1926, D2006, D2086): enabling single-speed positioning

- ◆ When b[8] is triggered, i.e. single-speed positioning is enabled, and START = ON, the single-speed positioning will be started. The pulse generator will send out pulses according to the setting of P(I) and V(I).
- ◆ Operation direction: Direction of relative positioning is determined by the sign bit of the value for P(I) (set in D1838, D1918, D1998, D2078). Direction of absolute positioning is determined by current position (CP).

3 Functions of Devices in DVP-PM

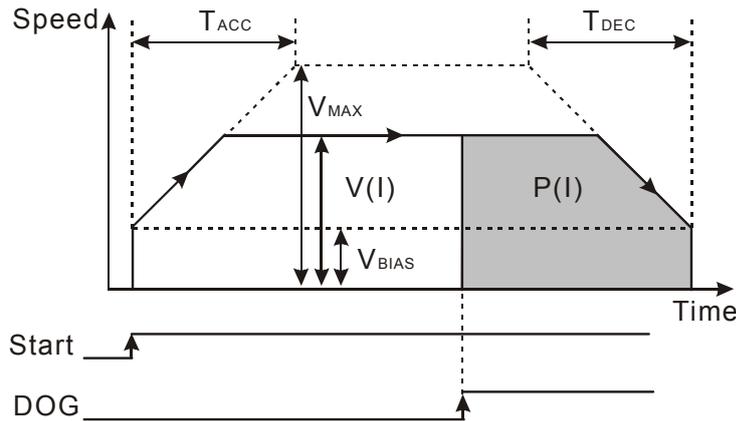
Forward running occurs when P(I) is bigger than CP. Reverse running occurs when P(I) is smaller than CP.

- ◆ The operation speed will be stable after the speed is accelerated from V_{BIAS} to the expected $V(I)$. When it is approaching the P(I) value set in the register, the positioning will start to decelerate to V_{BIAS} and stop. There are P(I) pulses generated during the positioning.
- ◆ The registers involved: D1824 (D1904, D1984, D2054) (V_{BIAS}), D1840 (D1920, D2000, D2080) ($V(I)$), D1822 (D1902, D1982, D2062) (V_{MAX}), D1838 (D1918, D1998, D2078) (P(I)), D1836 (D1916, D1996, D2076) (T_{ACC}) and D1837 (D1917, D1997, D2077) (T_{DEC}).



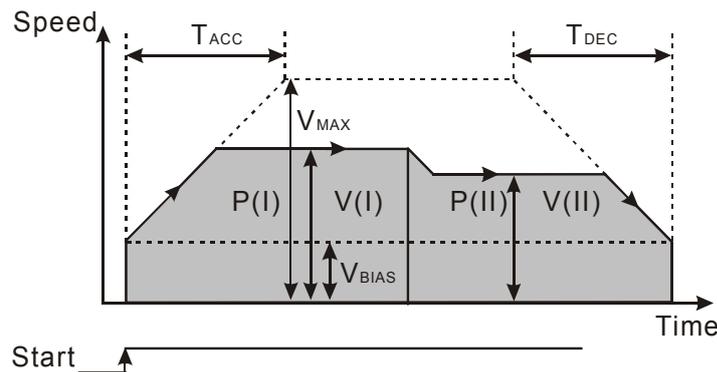
- ◆ When PWM mode is enabled (Parameter Setting, D1816/D1896/D1976/D2056, b[6] = 1), and the operation command “Single-speed positioning” is enabled (b[8] = 1), single-speed positioning will be executed on Y0~Y3 with single-phase pulses.
10. b9 of D1846 (D1926, D2006, D2086): enabling single-speed positioning interruption
- ◆ When b[9] is triggered and START = ON, the single-speed positioning interruption mode will be started. The single speed insterruption will be triggered to execute target position P(I) when DOG sensor is detected.
 - ◆ Operation direction: Direction of relative positioning is determined by the sign bit of the value for P(I) (set in D1838, D1918, D1998). Direction of absolute positioning is determined by current position (CP). Forward running occurs when P(I) is bigger than CP. Reverse running occurs when P(I) is smaller than CP.
 - ◆ The operation speed will be stable after the speed is accelerated from V_{BIAS} to the expected $V(I)$. When DOG sensor is detected, the single speed insterruption will be triggered to send out pulses set in P(I) register. Also, when it is approaching the P(I) value set in the register, the positioning will start to decelerate to V_{BIAS} and stop.
 - ◆ The registers involved: D1824 (D1904, D1984, D2054) (V_{BIAS}), D1840 (D1920, D2000, D2080) ($V(I)$), D1822 (D1902, D1982, D2062) (V_{MAX}), D1838 (D1918, D1998, D2078) (P(I)), D1836 (D1916, D1996) (T_{ACC}) and D1837 (D1917, D1997, D2077) (T_{DEC}).

3 Functions of Devices in DVP-PM



11. b10 of D1846 (D1926, D2006, D2086): enabling 2-speed positioning

- ◆ When b[10] is triggered and START = ON, the 2-speed positioning mode will be started. The 2nd-speed positioning will start immediately after the 1st-speed positioning reaches P(I).
- ◆ Operation direction: Direction of relative positioning is determined by the sign bit of the value for P(I) (set in D1838, D1918, D1998, D2078). Direction of absolute positioning is determined by current position (CP). Forward running occurs when P(I) is bigger than CP. Reverse running occurs when P(I) is smaller than CP.
- ◆ The operation speed will be stable after the speed is accelerated from V_{BIAS} to the expected $V(I)$. When P(I) is reached, $V(I)$ will accelerate/decelerate to $V(II)$ and operates in stable speed. Also, when it is approaching P(II) value set in the register, the positioning will start to decelerate to V_{BIAS} and stop. There are P(I)+P(II) pulses generated during the positioning
- ◆ The registers involved: D1824 (D1904, D1984, D2054) (V_{BIAS}), D1840 (D1920, D2000, D2080) ($V(I)$), D1844 (D1924, D2004, D2084) ($V(II)$), D1822 (D1902, D1982, D2062) (V_{MAX}), D1838 (D1918, D1998, D2078) (P(I)), D1842 (D1922, D2002, D2082) (P(II)), D1836 (D1916, D1996, D2076) (T_{ACC}) and D1837 (D1917, D1997, D2077) (T_{DEC}).



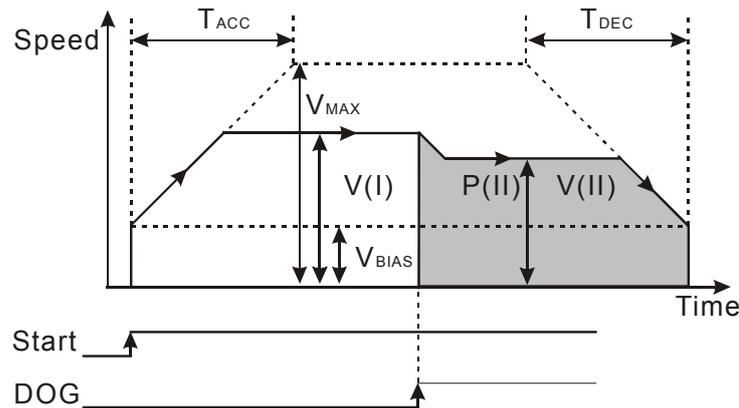
12. b11 of D1846 (D1926, D2006, D2086): enabling 2-speed positioning interruption

- ◆ When b[11] is triggered and START = ON, the 2-speed positioning interruption mode will be started. The 2nd-speed positioning will start immediately when DOG sensor is detected during the 1st-speed positioning process.
- ◆ Operation direction: Direction of relative positioning is determined by the sign bit of the value for P(I) (set in D1838, D1918, D1998, D2078). Direction of absolute positioning is determined by current position (CP). Forward running occurs when P(I) is bigger than CP. Reverse running occurs when P(I) is smaller than CP.
- ◆ The operation speed will be stable after the speed is accelerated from V_{BIAS} to the expected $V(I)$. When

3 Functions of Devices in DVP-PM

DOG sensor is detected during the 1st-speed positioning process, the 2nd-speed positioning will start immediately to accelerate/decelerate to V(II) and operates in stable speed. Pulse output can be stopped by the input signal from external STOP terminal.

- ◆ The registers involved: D1824 (D1904, D1984, D2054) (V_{BIAS}), D1840 (D1920, D2000, D2080) ($V(I)$), D1844 (D1924, D2004, D2084) ($V(II)$), D1822 (D1902, D1982, D2062) (V_{MAX}), D1838 (D1918, D1998, D2078) ($P(I)$), D1842 (D1922, D2002, D2082) ($P(II)$), D1836 (D1916, D1996, D2076) (T_{ACC}) and D1837 (D1917, D1997, D2077) (T_{DEC}).



13. b12 of D1846 (D1926, D2006): enabling OX subroutine

- ◆ b[12] = 1: start OX subroutine; b[12] = 0: stop OX subroutine

X axis		Y axis		Z axis		Work Modes
HW	LW	HW	LW	HW	LW	
	D1847		D1927		D2007	

1. b2 of D1847 (D1927, D2007): CLR signal output mode

- ◆ b[2] = 0: CLR will output 130ms signal to the servo as the clear signal when zero return is completed,
- ◆ b[2] = 1: CLR will be a general output point, and its status will be controlled by ON/OFF status of b[3].

2. b3 of D1847 (D1927, D2007): CLR output ON/OFF control

- ◆ b[3] = 0: output point CLR will be OFF.
- ◆ b[3] = 1: output point CLR will be ON.

3. b4 of D1847 (D1927, D2007): CLR polarity setting

- ◆ b[4] = 0: CLR is NO contact.
- ◆ b[4] = 1: CLR is NC contact.

4. b5 of D1847 (D1927, D2007): STOP mode setting

- ◆ b[5] = 0: When STOP signal comes in during the motor execution, the motor will decelerate to stop. When the next motion instruction is activated, the motor will ignore the unfinished distance and immediately execute the next positioning step.
- ◆ b[5] = 1: When STOP signal comes in during the motor execution, the motor will decelerate to stop. When the next motion instruction is activated, the motor will complete the unfinished distance before executing the next positioning step.

5. b6 of D1847 (D1927, D2007): range for MPG

- ◆ b[6] = 0: No limitation on the range of MPG pulse output

3 Functions of Devices in DVP-PM

- ◆ b[6] = 1: The range for MPG pulse output is limited within P(I) and P(II). When the range is exceeded, the pulse output will decelerate to stop.
6. b7 of D1847 (D1927, D2007): LSP/LSN stop mode
- ◆ b[7] = 0: During the running of motor, the motor decelerates to stop when encountering LSP/LSN signal input.
 - ◆ b[7] = 1: During the running of motor, the motor stops immediately when encountering LSP/LSN signal input.
7. b8 ~ b10 of D1847 (D1927, D2007): MASK(Stop) settings
- ◆ MASK settings include single-speed positioning, 2-speed positioning, single-speed positioning interruption and 2-speed positioning interruption.
 - ◆ b[10~8] = K1 (001): Triggering MASK by the rising edge of input signal on terminal ΦA_{\pm} .
 - ◆ b[10~8] = K2 (010): Triggering MASK by the falling edge of input signal on terminal ΦA_{\pm} .
 - ◆ b[10~8] = K3 (011): Triggering MASK by the rising edge of input signal on terminal ΦB_{\pm} .
 - ◆ b[10~8] = K4 (100): Triggering MASK by the falling edge of input signal on terminal ΦB_{\pm} .
8. b15 of D1847 (D1927, D2007): returning to default setting
- ◆ b[15] = 1: Return all parameters to default settings.

X axis		Y axis		Z axis		A axis		Current Position (CP) (pulse)
HW	LW	HW	LW	HW	LW	HW	LW	
D1849	D1848	D1929	D1928	D2009	D2008	D2089	D2088	

1. Range: -2,147,483,648 ~ +2,147,483,647
2. Current position (motor unit: pulse). The unit setting can be modified by b0 and b1 of D1816 (D1896, D1976).
When zero return is completed, current position registers will be updated by the definition of zero point (HP) (D1834 D1914, D1994)

X axis		Y axis		Z axis		A axis		Current Speed (CS) (pps)
HW	LW	HW	LW	HW	LW	HW	LW	
D1851	D1850	D1931	D1930	D2011	D2010	D2091	D2090	

1. Range: 0 ~ +2,147,483,647
2. Unit: pulse/sec (PPS)

X axis		Y axis		Z axis		A axis		Current Position (CP) (unit)
HW	LW	HW	LW	HW	LW	HW	LW	
D1853	D1852	D1933	D1932	D2013	D2012	D2093	D2092	

1. Range: -2,147,483,648 ~ +2,147,483,647
2. Current position (combined unit / machine unit). The unit setting can be modified by b0 and b1 of D1816 (D1896, D1976). When zero return is completed, current position registers will be updated by the definition of zero point (HP) (D1834 D1914, D1994).

X axis		Y axis		Z axis		A axis		Current Speed (CS) (unit)
HW	LW	HW	LW	HW	LW	HW	LW	
D1855	D1854	D1935	D1934	D2015	D2014	D2095	D2094	

3 Functions of Devices in DVP-PM

1. Range: 0 ~ +2,147,483,647
2. Unit: machine unit (cm/min, 10deg/min, inch/min). The unit setting can be modified by b0 and b1 of D1816 (D1896, D1976).

X axis		Y axis		Z axis		A axis		Execution Status
HW	LW	HW	LW	HW	LW	HW	LW	
	D1856		D1936		D2016		D2096	

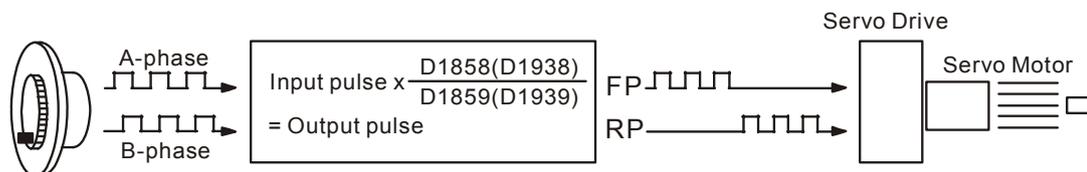
bit#	D1856 (D1936, D2016)	bit#	D1856 (D1936, D2016)
0	Forward pulse output in progress	8	Not defined
1	Reverse pulse output in progress	9	Not defined
2	Operation in progress	10	Not defined
3	Error occurs	11	Not defined
4	Operation pauses	12	Not defined
5	Forward MPG input	13	Not defined
6	Reverse MPG input	14	Not defined
7	Not defined	15	Not defined

X axis		Y axis		Z axis		A axis		Error Code
HW	LW	HW	LW	HW	LW	HW	LW	
	D1857		D1937		D2017		2097	

Please see Appendix in Chapter 14 for details.

X axis		Y axis		Z axis		A axis		Electronic Gear Ratio
HW	LW	HW	LW	HW	LW	HW	LW	
	D1858		D1938		D2018		D2098	
	D1859		D1939		D2019		D2099	Denominator of electronic gear ratio

1. When b5 of D1846 (D1926, D2006) is ON, MPG input operation will be enabled.
2. Users can generate A/B phase input pulse by MPG. See the figure below for the relation between input and output pulses.



3. If LSP is enabled during the operation, the forward pulse will be disabled but reverse pulse will be enabled. If LSN is enabled during the operation, the reverse pulse will be disabled but forward pulse will be enabled.
4. The output frequency is the multiplication result between MPG input pulses and the electronic gear ratio. (D1858 (D1938, D2018), D1859 (D1939, D2019)).

3 Functions of Devices in DVP-PM

X axis		Y axis		Z axis		A axis		MPG Input Frequency
HW	LW	HW	LW	HW	LW	HW	LW	
D1861	D1860	D1941	D1940	D2021	D2020	D2101	D2100	

The frequency of MPG input is not affected by the electronic gear ratio.

X axis		Y axis		Z axis		A axis		Number of accumulated MPG Input Pulses
HW	LW	HW	LW	HW	LW	HW	LW	
D1863	D1862	D1943	D1942	D2023	D2022	D2103	D2102	

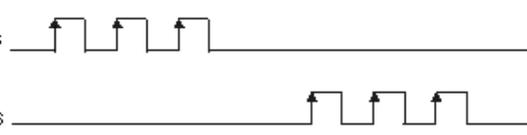
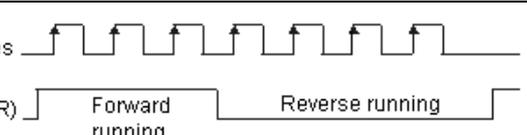
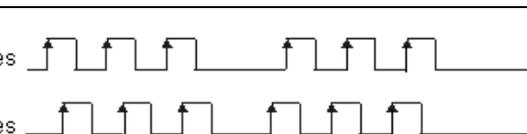
1. Number of MPG input pulses is accumulated and stored by corresponding registers. The accumulated value is plused by 1 when a forward pulse comes in, and minused by 1 when reverse pulse comes in.
2. The accumulated value is not affected by the electronic gear ratio (D1858 (D1938, D1918), D1859 (D1939, D2019)).

X axis		Y axis		Z axis		A axis		Response Time of MPG Input
HW	LW	HW	LW	HW	LW	HW	LW	
	D1864		D1944		D2024		D2104	

1. The faster the response time, the more synchronous between pulse output and MPG input.
2. The slower the response time, the more possible the pulse output lags behind MPG input.

Set value	ResponseTime
≥ 5	4ms (default)
4	32ms
3	108ms
2	256ms
1 or 0	500ms

3. b8 and b9 of D1864 (D1944, D2024): setting MPG input pulse pattern

b8	b9	Input pulse patterns (positive logic)	Description
0	0	FP forward pulses 	2-phase pulse
0	1	FP pulses 	1-phase pulse + direction
1	0	FP A-phase pulses 	A/B phase
1	1	RP B-phase pulses 	

3.12.2 Manual Modes

1. There are 8 manual modes in DVP-PM

- | | |
|--|-------------------------------------|
| 1. Mechanical zero return | 6. 2-speed positioning |
| 2. JOG mode | 7. 2-speed positioning interruption |
| 3. Single-speed positioning | 8. Variable speed mode |
| 4. Single-speed positioning interruption | 9. MPG input |
| 5. Externally triggering single-speed positioning* | |

Note*: Only DVP-20PM supports this function

2. When several manual modes are enabled at the same time, they will be processed in the following order.

- | | |
|---------------------------|---|
| 1. STOP | 7. Single-speed positioning |
| 2. Mechanical zero return | 8. Single-speed positioning interruption |
| 3. JOG+ mode | 9. Externally triggering single-speed positioning |
| 4. JOG- mode | 10. 2-speed positioning |
| 5. MPG input | 11. 2-speed positioning interruption |
| 6. Variable speed mode | |

When one of the manual modes is being executed and another manual mode is enabled, DVP-PM will remain the operation of the executing manual mode.

3. There are two types of pulse acceleration/deceleration curves.

- | | |
|--------------------|------------|
| 1. Trapezoid curve | 2. S curve |
|--------------------|------------|

3 Functions of Devices in DVP-PM

3.12.3 Position & Speed Control Registers for Manual Modes

Registers for the Motion								Parameter Name	Manual Mode								
X axis		Y axis		Z axis		A axis			JOG	Zero return	Single-speed positioning	Single-speed positioning interruption	Externally triggering single-speed positioning	2-speed positioning	2-speed positioning interruption	Variable speed	MPG input
HW	LW	HW	LW	HW	LW	HW	LW										
D1819	D1818	D1899	D1898	D1979	D1978	D2059	D2058	Pulses per revolution (PPR) (A)	No need to be set up if the unit (b0, b1 of D1816, D1896, D1976) is motor unit. Needs to be set up if the unit is machine unit or combined unit.								
D1821	D1820	D1901	D1900	D1981	D1980	D2061	D2060	Displacement per revolution(DPR) (B)									
	D1816		D1896		D1976		D2056	Parameter setting	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	
D1823	D1822	D1903	D1902	D1983	D1982	D2063	D2062	Maximum speed (V _{MAX})	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	
D1825	D1824	D1905	D1904	D1985	D1984	D2065	D2064	Bias speed (V _{BIAS})	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	
D1827	D1826	D1907	D1906	D1987	D1986	D2067	D2066	JOG speed (V _{JOG})	⊙	-	-	-	-	-	-	-	
D1829	D1828	D1909	D1908	D1989	D1988	D2069	D2068	Zero return speed (V _{RT})									
D1831	D1830	D1911	D1910	D1991	D1990	D2071	D2070	Zero return deceleration speed (V _{CR})									
	D1832		D1912		D1992		D2072	Number of PG0 signals for zero return (N)	-	⊙	-	-	-	-	-	-	
	D1833		D1913		D1993		D2073	Number of pulses for zero return (P)									
D1835	D1834	D1915	D1914	D1995	D1994	D2075	D2074	Definition of zero point (HP)									
	D1836		D1916		D1996		D2076	Acceleration time (T _{ACC})	⊙	⊙	⊙	⊙	⊙	⊙	⊙	-	
	D1837		D1917		D1997		D2077	Deceleration time (T _{DEC})	⊙	⊙	⊙	⊙	⊙	⊙	⊙	-	
D1839	D1838	D1919	D1918	D1999	D1998	D2079	D2078	Target position(I) (P(I))	-	-	⊙	⊙	⊙	⊙	⊙	-	
D1841	D1840	D1921	D1920	D2001	D2000	D2081	D2080	Operation speed (I) (V(I))	-	-	⊙	⊙	⊙	⊙	⊙	-	
D1843	D1842	D1923	D1922	D2003	D2002	D2083	D2082	Target position (II) (P(II))	-	-	-	-	-	⊙	⊙	-	
D1845	D1844	D1925	D1924	D2005	D2004	D2085	D2084	Operation speed (II) (V(II))	-	-	-	-	-	⊙	⊙	-	
	D1846		D1926		D2006		D2086	Operation Commands	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	
	D1847		D1927		D2007		D2087	Work modes	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	
D1849	D1848	D1929	D1928	D2009	D2008	D2089	D2088	Current position (CP) (PLS)	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	

3 Functions of Devices in DVP-PM

Registers for the Motion								Parameter Name	Manual Mode								
X axis		Y axis		Z axis		A axis			JOG	Zero return	Single-speed positioning	Single-speed positioning interruption	Externally triggering single-speed positioning	2-speed positioning	2-speed positioning interruption	Variable speed	MPG input
HW	LW	HW	LW	HW	LW	HW	LW										
D1851	D1850	D1931	D1930	D2011	D2010	D2091	D2090	Current speed (CS) (PPS)	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
D1853	D1852	D1933	D1932	D2013	D2012	D2093	D2092	Current position (CP) (unit)	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
D1855	D1854	D1935	D1934	D2015	D2014	D2095	D2094	Current speed (CS) (unit)	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
	D1858		D1938		D2018		D2098	Numerator of electronic gear	-	-	-	-	-	-	-	-	⊙
	D1859		D1939		D2019		D2099	Denominator of electronic gear	-	-	-	-	-	-	-	-	⊙
D1861	D1860	D1941	D1940	D2021	D2020	D2101	D2100	Frequency of MPG input	-	-	-	-	-	-	-	-	⊙
D1863	D1862	D1943	D1942	D2023	D2022	D2103	D2102	Number of accumulated MPG input pulses	-	-	-	-	-	-	-	-	⊙
	D1864		D1944		D2024		D2104	MPG response time	-	-	-	-	-	-	-	-	⊙
D1865								Stop mode (OX0-99)	-	-	-	-	-	-	-	-	-
D1867	D1866	D1947	D1946	D2027	D2026		D2106	Electrical zero point	-	-	-	-	-	-	-	-	-
D1868								Setting up the No. of OX	-	-	-	-	-	-	-	-	-
D1869								Error step of OX	-	-	-	-	-	-	-	-	-
D1872								Enabling Y output when OX is ready High byte: K1; low byte: designating start No. of Y output	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
D1873								Enabling Y output when OX executes M-code High byte: K1; low byte: designating start No. of Y output	-	-	-	-	-	-	-	-	-
D1874								No. of input point X for clearing OX M-code status	-	-	-	-	-	-	-	-	-
D1875		D1955						Enable external input (ZRN, MPG, JOG-, JOG+)	⊙	-	-	-	-	-	-	-	⊙

⊙ indicates that the register is applicable for the corresponding manual mode.

3 Functions of Devices in DVP-PM

MEMO

4.1 Basic Instructions

General Instructions

Instruction	Function	Operands	Execution speed (us)	Step	Page number
LD	Load NO contact	X, Y, M, S, T, C	3.3	3	4-3
LDI	Load NC contact	X, Y, M, S, T, C	3.3	3	4-3
AND	Connect NO contact in series	X, Y, M, S, T, C	3.3	3	4-3
ANI	Connect NC contact in series	X, Y, M, S, T, C	3.3	3	4-4
OR	Connect NO contact in parallel	X, Y, M, S, T, C	3.3	3	4-4
ORI	Connect NC contact in parallel	X, Y, M, S, T, C	3.3	3	4-4
ANB	Connect a block in series	N/A	2.3	3	4-5
ORB	Connect a block in parallel	N/A	2.3	3	4-5

Output Instructions

Instruction	Function	Operands	Execution speed (us)	Step	Page number
OUT	Output coil	Y, M, S	7.3	3	4-6
SET	Latches the ON status	Y, M, S	5.6	3	4-6
RST	Resets contacts, registers or coils	Y, M, S, T, C, D, V, Z	6.9	3	4-7

Timers, Counters

API	Instruction	Function	Operands	Execution speed (us)	Step	Page number
96	TMR	16-bit timer	T-K or T-D	19	5	4-7
97	CNT	16-bit counter	C-K or C-D (16 bits)	16	5	4-8
97	DCNT	32-bit counter	C-K or C-D (32 bits)	16.5	6	4-8

Rising / Falling-Edge Triggers

API	Instruction	Function	Operands	Execution speed (us)	Step	Page number
90	LDP	Load Rising edge trigger	X, Y, M, S, T, C	12.3	3	4-9
91	LDF	Load Falling edge trigger	X, Y, M, S, T, C	12.3	3	4-9
92	ANDP	Connect Rising edge in series	X, Y, M, S, T, C	12.3	3	4-9
93	ANDF	Connect Falling edge in series	X, Y, M, S, T, C	12.3	3	4-10
94	ORP	Connect Rising edge in parallel	X, Y, M, S, T, C	12.6	3	4-10
95	ORF	Connect Falling edge in parallel	X, Y, M, S, T, C	12.6	3	4-11

4 Basic Instructions

Rising-/Falling-Edge Output Instruction

API	Instruction	Function	Operands	Execution speed (us)	Step	Page number
89	PLS	Rising-edge output	Y, M	20.7	3	4-11
99	PLF	Falling-edge output	Y, M	20.9	3	4-12

Other Instructions

API	Instruction	Function	Operands	Execution speed (us)	Step	Page number
-	P	Pointer	P0 ~ P255	-	1	4-12

4.2 Explanations on Basic Instructions

Mnemonic	Function						Controllers	
LD	Load NO contact						20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	
	✓	✓	✓	✓	✓	✓	-	

Explanations:

The LD instruction is used to load NO contact which connects to left side bus line or starts a new block of program connecting in series or parallel connection.

Program Example:

Ladder diagram:



Instruction:

LD	X0	Load NO contact X0
AND	X1	Connect NO contact X1 in series
OUT	Y1	Drive coil Y1

Operation:

Mnemonic	Function						Controllers	
LDI	Load NC contact						20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	
	✓	✓	✓	✓	✓	✓	-	

Explanations:

The LDI instruction is used to load NC contact which connects to left side bus line or starts a new block of program connecting in series or parallel connection.

Program Example:

Ladder diagram:



Instruction:

LDI	X0	Load NC contact X0
AND	X1	Connect NO contact X1 in series
OUT	Y1	Drive coil Y1

Operation:

Mnemonic	Function						Controllers	
AND	Connect NO contact in series						20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	
	✓	✓	✓	✓	✓	✓	-	

Explanations:

The AND instruction is used to connect NO contact in series.

Program Example:

4 Basic Instructions

Ladder diagram:



Instruction:

LD X1
AND X0
 OUT Y1

Operation:

Load NC contact X1
 Connect NO contact X0 in series
 Drive Y1 coil

Mnemonic	Function								
ANI	Connect NC contact in series							Controllers	
								20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999		
	✓	✓	✓	✓	✓	✓	-		

Explanations:

The ANI instruction is used to connect NC contact in series.

Program Example:

Ladder diagram:



Instruction:

LD X1
ANI X0
 OUT Y1

Operation:

Load NO contact X1
 Connect NC contact X0 in series
 Drive Y1 coil

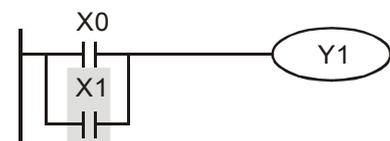
Mnemonic	Function								
OR	Connect NO contact in parallel							Controllers	
								20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999		
	✓	✓	✓	✓	✓	✓	-		

Explanations:

The OR instruction is used to connect NO contact in parallel.

Program Example:

Ladder diagram:



Instruction:

LD X0
OR X1
 OUT Y1

Operation:

Load NO contact X0
 Connect NO contact X1 in parallel
 Drive Y1 coil

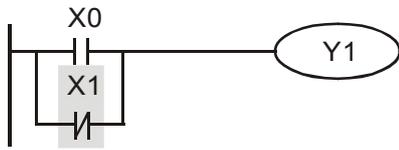
Mnemonic	Function								
ORI	Connect NC contact in parallel							Controllers	
								20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999		
	✓	✓	✓	✓	✓	✓	-		

Explanations:

The ORI instruction is used to connect NC contact in parallel.

Program Example:

Ladder diagram:



Instruction:

LD X0

ORI X1

OUT Y1

Operation:

Load NO contact X0

Connect NC contact X1 in parallel

Drive Y1 coil

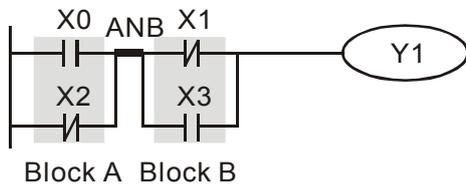
Mnemonic	Function	Controllers	
ANB	Connect a block in series	20PM	10PM
Operand	N/A		

Explanations:

The ANB instruction is used to connect a circuit block to the preceding block in series. Generally, the circuit block to be connected in series consists of several contacts which form a parallel connection structure

Program Example:

Ladder diagram:



Block A Block B

Instruction:

LD X0

ORI X2

LDI X1

OR X3

ANB

OUT Y1

Operation:

Load NO contact X0

Connect NC contact X2 in parallel

Load NC contact X1

Connect NO contact X3 in parallel

Connect circuit block in series

Drive Y1 coil

Mnemonic	Function	Controllers	
ORB	Connect a block in parallel	20PM	10PM
Operand	N/A		

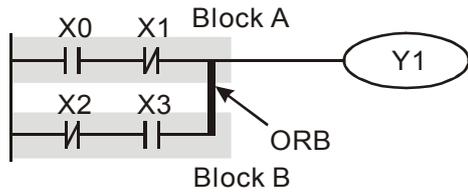
Explanations:

The ORB instruction is used to connect a circuit block to the preceding block in parallel. Generally, the circuit block to be connected in parallel consists of several contacts which form a serial connection structure.

Program Example:

4 Basic Instructions

Ladder diagram:



Instruction:

LD X0
 ANI X1
 LDI X2
 AND X3
ORB
 OUT Y1

Operation:

Load NO contact X0
 Connect NC contact X1 in series
 Load NC contact X2
 Connect NO contact X3 in series
Connect circuit block in parallel
 Drive Y1 coil

Mnemonic	Function												
OUT	Output Coil												
	<table border="1"> <thead> <tr> <th colspan="2">Controllers</th> </tr> <tr> <th>20PM</th> <th>10PM</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>							Controllers		20PM	10PM		
Controllers													
20PM	10PM												
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999						
	—	✓	✓	✓	-	-	-						

Explanations:

1. Output the program evaluation results before OUT instruction to the designated device.
2. Status of coil contact:

Evaluation result	OUT instruction		
	Coil	Associated Contacts	
		NO contact (normally open)	NC contact (normally close)
FALSE	OFF	Current blocked	Current flows
TRUE	ON	Current flows	Current blocked

Program Example:

Ladder diagram:



Instruction:

LDI X0
 AND X1
OUT Y1

Operation:

Load NC contact X0
 Connect NO contact X1 in series
Drive Y1 coil

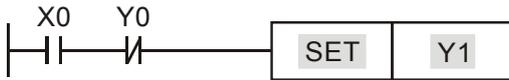
Mnemonic	Function												
SET	Latches the ON status												
	<table border="1"> <thead> <tr> <th colspan="2">Controllers</th> </tr> <tr> <th>20PM</th> <th>10PM</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>							Controllers		20PM	10PM		
Controllers													
20PM	10PM												
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999						
	-	✓	✓	✓	-	-	-						

Explanations:

When the SET instruction is driven, its designated device will be ON and latched whether the SET instruction is still driven. In this case, RST instruction can be applied to turn off the device.

Program Example:

Ladder diagram:



Instruction:

LD X0

ANI Y0

SET Y1

Operation:

Load NO contact X0

Connect NC contact Y0 in series

Drive Y1 and latch the status

Mnemonic	Function							
RST	Resets contacts, registers or coils							
	Controllers							
	20PM				10PM			
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	V, Z
	-	✓	✓	✓	✓	✓	✓	✓

Explanations:

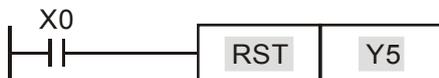
- When RST instruction is driven, the actions of the designated devices are:

Device	Status
S, Y, M	Coil and contact are set to OFF.
T, C	Current value is cleared. Associated contacts or coils are reset .
D, V, Z	The content is set to 0.

- Status of designated devices remains the same when RST instruction is not executed.

Program Example:

Ladder diagram:



Instruction:

LD X0

RST Y5

Operation:

Load NO contact X0

Reset contact Y5

Mnemonic	Function	
TMR	16-bit Timer	
	Controllers	
	20PM 10PM	
Operand	T-K	T0 ~ T255, K0 ~ K32,767
	T-D	T0 ~ T255, D0 ~ D9999

Explanations:

When TMR instruction is executed, the designated timer coil will be ON, and the timer will be enabled. When the set value in the timer is reached (present value \geq set value), the associated contact will be ON and act as follows:

NO (normally open) contact	Current flows
NC (normally closed) contact	Current blocked

Program Example:

Ladder diagram:



Instruction:

LD X0

TMR T5 K1,000

Operation:

Load NO contact X0

Set value in timer T5 as K1,000

4 Basic Instructions

Remarks:

See the specification of DVP-PM for the range of operand T.

Mnemonic	Function		Controllers	
CNT	16-bit Counter		20PM	10PM
Operand	C-K	C0 ~ C199, K0 ~ K32,767		
	C-D	C0 ~ C199, D0 ~ D9999		

Explanations:

- When CNT instruction goes from OFF to ON, the designated counter coil will be driven, and the present value in the counter will plus 1. When the counting reaches the set value (present value = set value), the associated contact will be ON and act as follows:

NO (normally open) contact	Current flows
NC (normally closed) contact	Current blocked

- If there are other counting pulse inputs after the counting reaches its target, the contact and present value will stay intact. Use RST instruction to restart or reset the counting.

Program Example:

Ladder diagram:



Instruction:

LD X0

Operation:

Load NO contact X0

CNT C20 K100 Set value in counter C20 as K100

Mnemonic	Function		Controllers	
DCNT	32-bit Counter		20PM	10PM
Operand	C-K	C200, C204, C208 ~ C255, K-2,147,483,648 ~ K2,147,483,647		
	C-D	C200, C204, C208 ~ C255, D0 ~ D9999		

Explanations:

- DCNT is the instruction for enabling the 32-bit counters C200 ~ C255.
- For general count-up / count-down counters C200 ~ C255, when DCNT instruction goes from OFF to ON, the present value in the counter will plus 1 (counting up) or minus 1 (counting down) according to the modes set by M1200 ~ M1234.

Program Example:

Ladder diagram:



Instruction:

LD M0

Operation:

Load NO contact M0

DCNT C254 K1,000 Set value in counter C254 as K1,000

Mnemonic	Function						Controllers	
LDP	Load Rising-edge Trigger						20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	
	✓	✓	✓	✓	✓	✓	-	

Explanations:

Similar to LD instruction, LDP is the rising-edge contact which is triggered only at the rising edge of the signal.

Program Example:

Ladder diagram:



Instruction:

Operation:

LDP	X0	Load rising-edge contact X0
AND	X1	Connect NO contact X1 in series
OUT	Y1	Drive coil Y1

Remarks:

1. See the specification of DVP-PM for the range of operands.
2. If the status of a rising edge is ON before DVP-PM is powered, the rising edge contact will be activated after DVP-PM is powered.

Mnemonic	Function						Controllers	
LDF	Load Falling-edge Trigger						20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	
	✓	✓	✓	✓	✓	✓	-	

Explanations:

Similar to LD instruction, LDF is the falling-edge contact which is triggered only at the falling edge of the signal.

Program Example:

Ladder diagram:



Instruction:

Operation:

LDF	X0	Load falling-edge contact X0
AND	X1	Connect NO contact X1 in series
OUT	Y1	Drive coil Y1

Mnemonic	Function						Controllers	
ANDP	Connect Rising-edge in Series						20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	
	✓	✓	✓	✓	✓	✓	-	

Explanations:

4 Basic Instructions

Similar to AND instruction, ANDP instruction connects rising-edge trigger in series with another device or block.

Program Example:

Ladder diagram:



Instruction:

LD X0

Operation:

Load NO contact X0

ANDP X1

Connect rising-edge contact X1 in series

OUT Y1

Drive coil Y1

Mnemonic	Function						Controllers	
	ANDF	Connect Falling-edge in Series						20PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	
	✓	✓	✓	✓	✓	✓	-	

Explanations:

Similar to AND instruction, ANDF instruction connects falling edge trigger in series with another device or block.

Program Example:

Ladder diagram:



Instruction:

LD X0

Operation:

Load NO contact X0

ANDF X1

Connect falling-edge contact X1 in series

OUT Y1

Drive coil Y1

Mnemonic	Function						Controllers	
	ORP	Connect Rising-edge in Parallel						20PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	
	✓	✓	✓	✓	✓	✓	-	

Explanations:

Similar to OR instruction, ORP instruction connects rising edge triggers in parallel with another device or block.

Program Example:

Ladder diagram:



Instruction:

LD X0

Operation:

Load NO contact X0

ORP X1

Connect rising-edge contact X1 in parallel

OUT Y1

Drive coil Y1

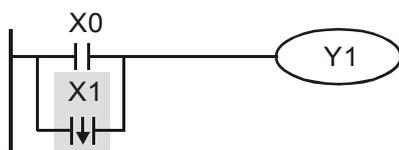
Mnemonic	Function						
ORF	Connect Falling-edge in Parallel						
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999
	✓	✓	✓	✓	✓	✓	-

Explanations:

Similar to OR instruction, ORF instruction connects falling edge triggers in parallel with another device or block.

Program Example:

Ladder diagram:



Instruction:

LD X0 Load NO contact X0

ORF X1 Connect Falling-edge contact X1 in parallel

OUT Y1 Drive coil Y1

Operation:

Load NO contact X0

Connect Falling-edge contact X1 in parallel

Drive coil Y1

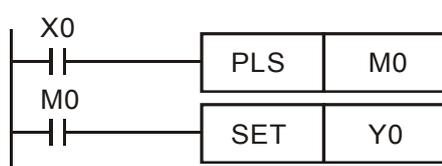
Mnemonic	Function							Controllers	
PLS	Rising-Edge Output							20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999		
	-	✓	✓	-	-	-	-		

Explanations:

When X0 goes from OFF to ON (Rising-edge triggered), PLS instruction executes and M0 generates a pulse with the pulse width of one scan time.

Program Example:

Ladder diagram:



Instruction:

LD X0 Load NO contact X0

PLS M0 M0 rising-edge output

LD M0 Load NO contact M0

SET Y0 Y0 latched (ON)

Operation:

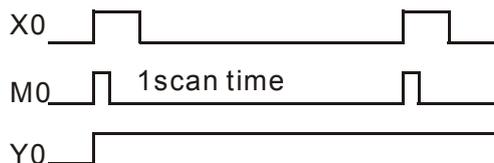
Load NO contact X0

M0 rising-edge output

Load NO contact M0

Y0 latched (ON)

Timing diagram:



4 Basic Instructions

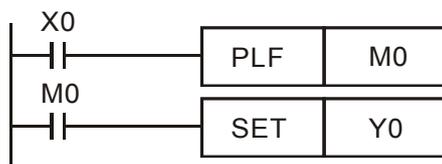
Mnemonic	Function						Controllers	
PLF	Falling-Edge Output						20PM	10PM
Operand	X0~X377	Y0~Y377	M0~M4095	S0~S1023	T0~T255	C0~C255	D0~D9999	
	-	✓	✓	-	-	-	-	

Explanations:

When X0 goes from ON to OFF (falling-edge triggered), PLS instruction executes and M0 generates a pulse with the pulse width of one scan time.

Program Example:

Ladder diagram:

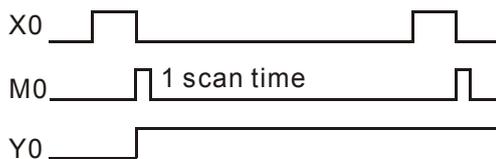


Instruction:

LD	X0	Load NO contact X0
PLF	M0	M0 falling-edge output
LD	M0	Load NO contact M0
SET	Y0	Y0 latched (ON)

Operation:

Timing diagram:



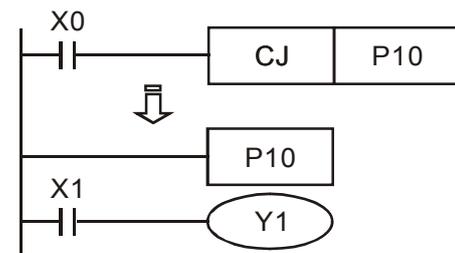
Mnemonic	Function						Controllers	
P	Pointer						20PM	10PM
Operand	P0 ~ P255							

Explanations:

Pointer P is used in API 00 CJ, API 01 CALL, API 256 CJN and API 257 JMP instructions. Pointer P does not need to be used starting from P0. However, number of P cannot be repeated, otherwise unexpected errors may occur.

Program Example:

Ladder diagram:



Instruction:

LD	X0	Load NO contact X0
CJ	P10	Jump to P10 subroutine
:		
P10		Pointer P10
LD	X1	Load NO contact X1
OUT	Y1	Drive coil Y1

Operation:

5 Categories and Use of Basic Application Instructions

5.1 List of Instructions

Category	API	Mnemonic		Pulse	Function	20PM	10PM	STEPS		Page
		16-bit	32-bit					16-bit	32-bit	
Loop Control	00	CJ	-	✓	Conditional Jump	✓	✓	3	-	5-15
	01	CALL	-	✓	Call Subroutine	✓	✓	3	-	5-18
	02	SRET	-	-	Subroutine Return	✓	✓	1	-	5-18
	07	WDT	-	✓	Watchdog timer refresh	✓	✓	1	-	5-20
	08	RPT	-	-	Start of a Nested RPT-RPE loop (Only 1 level available)	✓	✓	3	-	5-21
	09	RPE	-	-	End of a Nested RPT-RPE loop	✓	✓	1	-	5-21
Transmission Comparison	10	CMP	DCMP	✓	Compare	✓	✓	7	9	5-23
	11	ZCP	DZCP	✓	Zone Compare	✓	✓	9	12	5-24
	12	MOV	DMOV	✓	Move	✓	✓	5	6	5-25
	13	SMOV	-	✓	Shift move	✓	✓	11	-	5-26
	14	CML	DCML	✓	Complement	✓	✓	5	9	5-29
	15	BMOV	-	✓	Block move	✓	✓	7	-	5-30
	16	FMOV	DFMOV	✓	Fill move	✓	✓	7	13	5-32
	17	XCH	DXCH	✓	Exchange	✓	✓	5	9	5-33
	18	BCD	DBCD	✓	Convert BIN to BCD	✓	✓	5	5	5-34
	19	BIN	DBIN	✓	Convert BCD to BIN	✓	✓	5	5	5-35
Four Arithmetic Operations	20	ADD	DADD	✓	Addition	✓	✓	7	9	5-36
	21	SUB	DSUB	✓	Subtraction	✓	✓	7	9	5-38
	22	MUL	DMUL	✓	Multiplication	✓	✓	7	9	5-39
	23	DIV	DDIV	✓	Division	✓	✓	7	9	5-40
	24	INC	DINC	✓	Increment	✓	✓	3	3	5-41
	25	DEC	DDEC	✓	Decrement	✓	✓	3	3	5-42
	26	WAND	DWAND	✓	Logical Word AND	✓	✓	7	9	5-43
	27	WOR	DWOR	✓	Logical Word OR	✓	✓	7	9	5-44
	28	WXOR	DWXOR	✓	Logical XOR	✓	✓	7	9	5-45
	29	NEG	DNEG	✓	2's Complement (Negation)	✓	✓	3	3	5-46
Rotation and Displacement	30	ROR	DROR	✓	Rotate right	✓	✓	5	9	5-48
	31	ROL	DROL	✓	Rotate left	✓	✓	5	9	5-49
	32	RCR	DRCR	✓	Rotate right with carry	✓	✓	5	9	5-50
	33	RCL	DRCL	✓	Rotate left with carry	✓	✓	5	9	5-51
	34	SFTR	-	✓	Bit shift right	✓	✓	9	-	5-52
	35	SFTL	-	✓	Bit shift left	✓	✓	9	-	5-53
	36	WSFR	-	✓	Word shift right	✓	✓	9	-	5-54
	37	WSFL	-	✓	Word shift left	✓	✓	9	-	5-56
	38	SFWR	-	✓	Shift register write	✓	✓	7	-	5-57
	39	SFRD	-	✓	Shift register read	✓	✓	7	-	5-58
Data Processing	40	ZRST	-	✓	Zone Reset	✓	✓	5	-	5-59
	41	DECO	-	✓	Decode	✓	✓	7	-	5-60
	42	ENCO	-	✓	Encode	✓	✓	7	-	5-61

5 Categories and Use of Basic Application Instructions

Category	API	Mnemonic		Pulse	Function	20PM	10PM	STEPS		Page
		16-bit	32-bit					16-bit	32-bit	
	43	SUM	DSUM	✓	Sum of Active bits	✓	✓	5	9	5-63
	44	BON	DBON	✓	Check specified bit status	✓	✓	7	13	5-64
	45	MEAN	DMEAN	✓	Mean	✓	✓	7	13	5-65
	46	ANS	-	-	Timed Annunciator Set	✓	✓	7	-	5-66
	47	ANR	-	✓	Annunciator Reset	✓	✓	1	-	5-67
	48	SQR	DSQR	✓	Square Root	✓	✓	5	9	5-69
	49	-	DFLT	✓	Floating Point	✓	✓	-	6	5-70
High Speed Processing	50	REF	-	✓	Refresh	✓	✓	5	-	5-72
Handy Instructions	61	SER	DSER	✓	Search a data stack	✓	✓	9	17	5-74
	66	ALT	-	✓	Alternate state	✓	✓	3	-	5-75
	67	RAMP	DRAMP	-	Ramp variable value	✓	✓	9	17	5-76
	69	SORT	DSORT	-	Data sort	✓	✓	11	21	5-78
I/O	78	FROM	DFROM	✓	Read CR Data from Special Modules	✓	✓	9	12	5-80
	79	TO	DTO	✓	Write CR Data into Special Modules	✓	✓	9	13	5-81
	87	ABS	DABS	✓	Absolute value	✓	✓	3	5	5-84
Basic Instructions	89	PLS	-	-	Rising-edge Output	✓	✓	3	-	4-11
	90	LDP	-	-	Rising-edge Detection Operation	✓	✓	3	-	4-9
	91	LDF	-	-	Falling-edge Detection Operation	✓	✓	3	-	4-9
	92	ANDP	-	-	Rising-edge Series Connection	✓	✓	3	-	4-9
	93	ANDF	-	-	Falling-edge Series Connection	✓	✓	3	-	4-10
	94	ORP	-	-	Rising-edge Parallel Connection	✓	✓	3	-	4-10
	95	ORF	-	-	Falling-edge Parallel Connection	✓	✓	3	-	4-11
	96	TMR	-	-	16-bit Timer	✓	✓	5	-	4-7
	97	CNT	DCNT	-	16-bit/32-bit Counter	✓	✓	5	6	4-8
	99	PLF	-	-	Falling-edge Output	✓	✓	3	-	4-12
Communications	100	MODRD	-	-	Read Modbus Data	✓	✓	7	-	5-85
	101	MODWR	-	-	Write Modbus Data	✓	✓	7	-	5-89
Floating Point Operation	110	-	DECMP	✓	Floating Point Compare	✓	✓	7	9	5-94
	111	-	DEZCP	✓	Floating Point Zone Compare	✓	✓	9	12	5-95
	112		DMOV	✓	Move floating point data	✓	✓		9	5-96
	116	-	DRAD	✓	Degree → Radian	✓	✓	-	6	5-97
	117	-	DDEG	✓	Radian → Degree	✓	✓	-	6	5-98
	120	-	DEADD	✓	Floating Point Addition	✓	✓	7	9	5-99
	121	-	DESUB	✓	Floating Point Subtraction	✓	✓	7	9	5-100
	122	-	DEMUL	✓	Floating Point Multiplication	✓	✓	7	9	5-101

5 Categories and Use of Basic Application Instructions

Category	API	Mnemonic		Pulse	Function	20PM	10PM	STEPS		Page
		16-bit	32-bit					16-bit	32-bit	
	123	-	DEDIV	✓	Floating Point Division	✓	✓	7	9	5-102
	124	-	DEXP	✓	Float Exponent Operation	✓	✓	-	6	5-103
	125	-	DLN	✓	Float Natural Logarithm Operation	✓	✓	-	6	5-104
	126	-	DLOG	✓	Float Logarithm Operation	✓	✓	-	9	5-105
	127	-	DESQR	✓	Floating Point Square Root	✓	✓	5	6	5-106
	128	-	DPOW	✓	Floating Point Power Operation	✓	✓	-	9	5-107
	129	-	DINT	✓	Float to Integer	✓	✓	-	6	5-109
	130	-	DSIN	✓	Sine	✓	✓	5	6	5-110
	131	-	DCOS	✓	Cosine	✓	✓	5	6	5-112
	132	-	DTAN	✓	Tangent	✓	✓	5	6	5-114
	133	-	DASIN	✓	Arc Sine	✓	✓	-	6	5-116
	134	-	DACOS	✓	Arc Cosine	✓	✓	-	6	5-117
	135	-	DATAN	✓	Art Tangent	✓	✓	-	6	5-118
	136	-	DSINH	✓	Hyperbolic Sine	✓	✓	-	6	5-119
	137	-	DCOSH	✓	Hyperbolic Cosine	✓	✓	-	6	5-120
	138	-	DTANH	✓	Hyperbolic Tangent	✓	✓	-	6	5-121
	172	-	DADDR	✓	Floating point addition	✓	✓	-	13	5-122
	173	-	DSUBR	✓	Floating point subtraction	✓	✓	-	13	5-123
	174	-	DMULR	✓	Floating point multiplication	✓	✓	-	13	5-124
	175	-	DDIVR	✓	Floating point division	✓	✓	-	13	5-125
Contact Type Logic Operation	215	LD&	DLD&	-	S1 & S2	✓	✓	5	7	5-126
	216	LD	DLD	-	S1 S2	✓	✓	5	7	5-126
	217	LD^	DLD^	-	S1 ^ S2	✓	✓	5	7	5-126
	218	AND&	DAND&	-	S1 & S2	✓	✓	5	7	5-127
	219	AND	DAND	-	S1 S2	✓	✓	5	7	5-127
	220	AND^	DAND^	-	S1 ^ S2	✓	✓	5	7	5-127
	221	OR&	DOR&	-	S1 & S2	✓	✓	5	7	5-128
	222	OR	DOR	-	S1 S2	✓	✓	5	7	5-128
	223	OR^	DOR^	-	S1 ^ S2	✓	✓	5	7	5-128
Contact Type Comparison	224	LD=	DLD=	-	S1 = S2	✓	✓	5	7	5-129
	225	LD>	DLD>	-	S1 > S2	✓	✓	5	7	5-129
	226	LD<	DLD<	-	S1 < S2	✓	✓	5	7	5-129
	228	LD<>	DLD<>	-	S1 ≠ S2	✓	✓	5	7	5-129
	229	LD≤	DLD≤	-	S1 ≤ S2	✓	✓	5	7	5-129
	230	LD≥	DLD≥	-	S1 ≥ S2	✓	✓	5	7	5-129
	232	AND=	DAND=	-	S1 = S2	✓	✓	5	7	5-130
	233	AND>	DAND>	-	S1 > S2	✓	✓	5	7	5-130
	234	AND<	DAND<	-	S1 < S2	✓	✓	5	7	5-130
	236	AND<>	DAND<>	-	S1 ≠ S2	✓	✓	5	7	5-130
	237	AND≤	DAND≤	-	S1 ≤ S2	✓	✓	5	7	5-130

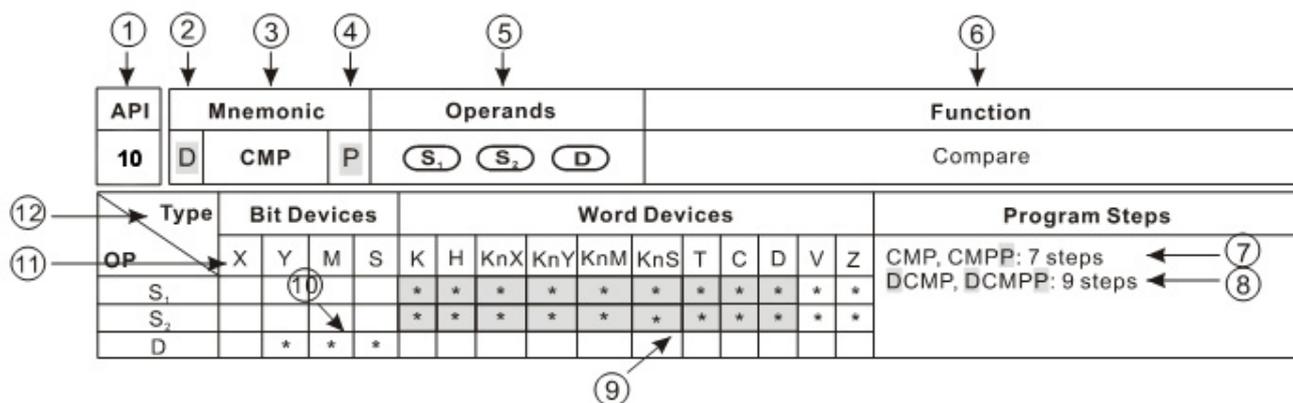
5 Categories and Use of Basic Application Instructions

Category	API	Mnemonic		Pulse	Function	20PM	10PM	STEPS		Page
		16-bit	32-bit					16-bit	32-bit	
	238	AND>=	DAND>= =	-	$S1 \geq S2$	✓	✓	5	7	5-130
	240	OR=	DOR=	-	$S1 = S2$	✓	✓	5	7	5-131
	241	OR>	DOR>	-	$S1 > S2$	✓	✓	5	7	5-131
	242	OR<	DOR<	-	$S1 < S2$	✓	✓	5	7	5-131
	244	OR<>	DOR<>	-	$S1 \neq S2$	✓	✓	5	7	5-131
	245	OR<=	DOR<=	-	$S1 \leq S2$	✓	✓	5	7	5-131
	246	OR>=	DOR>=	-	$S1 \geq S2$	✓	✓	5	7	5-131
Other Instructions	147	SWAP	DSWAP	✓	Byte swap	✓	✓	3	5	5-132
	154	RAND	DRAND	✓	Random number	✓	✓	7	13	5-133
	202	SCAL	-	✓	Proportional value calculation	✓	✓	9	-	5-134
	203	SCLP	DSCLP	✓	Parameter proportional value calculation	✓	✓	7	13	5-136
	256	CJN	-	✓	Negated Conditional Jump	✓	✓	3	-	5-140
	257	JMP	-	-	Unconditional Jump	✓	✓	3	-	5-141
	258	BRET	-	-	Return to Bus Line	✓	✓	1	-	5-142
	259	MMOV	-	✓	16-bit → 32-bit Conversion	✓	✓	6	-	5-143
260	RMOV	-	✓	32-bit → 16-bit Conversion	✓	✓	6	-	5-144	

5 Categories and Use of Basic Application Instructions

5.2 Instruction Composition

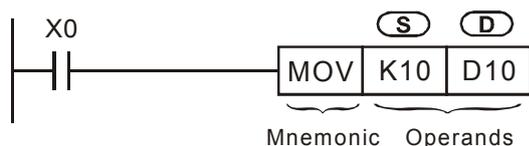
- ◆ Instructions consist of either just the instruction or the instruction followed by operands for parameter settings
 - Mnemonic: Indicates the name and the function of the instruction
 - Operand: The parameter setting for the instruction
- Mnemonic of an instruction usually occupies 1 step, and each operand occupies 2 steps for 16-bit instruction or 3 steps for 32-bit instruction.
- ◆ Format of an application instruction



- ① API number for instruction
- ② Mnemonic code with a prefix "D" indicates 32-bit instruction, e.g. API 10 DCMP.
- ③ The core mnemonic code of instruction
- ④ Mnemonic code with a suffix "P" indicates pulse instruction, e.g. API 12 MOV_P.
- ⑤ Operands
- ⑥ Function of the instruction
- ⑦ Steps occupied by the 16-bit/32-bit instruction
- ⑧ Steps occupied by the 16-bit/32-bit instruction
- ⑨ Column marked with * and in grey refers to V, Z index register modification is applicable.
- ⑩ Column marked with * is the device applicable for the operand.
- ⑪ Device name
- ⑫ Device type

- ◆ Specify an application instruction:

The application instructions of DVP-PM are specified by API numbers 0~260 and each has its mnemonic. When designing the user program with ladder editing program (PMSOFT), users only need to key in the mnemonic, e.g. MOV, and the instruction will be inserted. Take MOV instruction for example:



MOV instruction moves the operand designated in **S** to the operand designated in **D**.

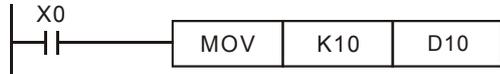
S	Source: if there are more than one source is required, it will be indicated as S ₁ , S ₂ ...etc.
D	Destination: if there are more than one destination is required, it will be indicated as D ₁ , D ₂ ...etc.
If the operand can only be constant K/H or a register, it will be represented as m , m₁ , n , n₁ , n₂ ...etc	

5 Categories and Use of Basic Application Instructions

◆ Length of Operand (16-bit or 32-bit instruction)

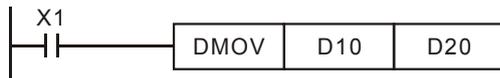
The length of operand can be divided into two groups, 16-bit and 32-bit, for processing data of different length. A prefix "D" indicates 32-bit instructions.

16-bit MOV instruction



When X0 = ON, K10 will be sent to D10.

32-bit DMOV instruction

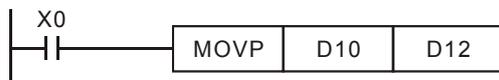


When X1 = ON, the content in (D11, D10) will be sent to (D21, D20).

◆ Continuous execution vs. Pulse execution

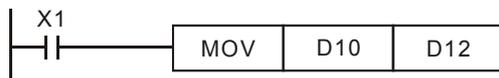
1. There are two execution types for instructions: continuous execution and pulse execution. Program scan time is shorter when instructions are not executed. Therefore, using the pulse execution instruction can reduce the scan time of the program.
2. The 'pulse' function allows the associated instruction to be activated on the rising edge of the drive contact. The instruction is driven ON for the duration of one program scan.
3. In addition, while the control input remains ON, the associate instruction will not be executed for the second time. To re-execute the instruction the control input must be turned from OFF to ON again.

Pulse execution instruction



MOV P instruction will be executed once in present scan period only when X0 goes from OFF to ON.

Continuous execution instruction



When X1 = ON, MOV instruction will be re-executed in every scan of program. This is called continuous execution instruction.

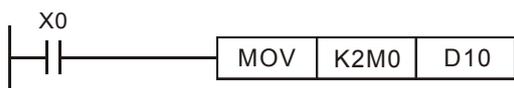
◆ Operands

1. Bit devices X, Y, M, and S can be combined into word device, storing values and data for operations in the form of KnX, KnY, KnM and KnS in an application instruction.
2. Data register D, timer T, counter C and index register E, F are designated by general operands.
3. A data register D consists of 16 bits, i.e. a 32-bit data register consists of 2 consecutive D registers.
4. If an operand of a 32-bit instruction designates D0, 2 consecutive registers D1 and D0 will be occupied. D1 is the high word and D0 is the low word. This principal also applies to timer T, 16-bit counters C0 ~ C199.
5. When the 32-bit counters C200 ~ C255 are used as data registers, they can only be designataed by the operands of 32-bit instructions.

◆ Operand Data format

1. X, Y, M, and S are defined as bit devices which indicate ON/OFF status.
2. 16-bit (or 32-bit) devices T, C, D, and registers V, Z are defined as word devices.
3. "Kn" can be placed before bit devices X, Y, M and S to make it a word device for performing word-device operations. (n = 1 refers to 4 bits. For 16-bit instruction, n = K1 ~ K4; for 32-bit instruction, n = K1 ~ K8). For example, K2M0 refers to 8 bits, M0 ~ M7

5 Categories and Use of Basic Application Instructions



When X0 = ON, the contents in M0 ~ M7 will be moved to b0 ~b7 in D10 and b8 ~ b15 will be set to "0".

◆ Kn values

16-bit instruction		32-bit instruction	
Designated value: K-32,768 ~ K32,767		Designated value: K-2,147,483,648 ~ K2,147,483,647	
Values for designated K1 ~ K4		Values for designated K1 ~ K8	
K1 (4 bits)	0 ~ 15	K1 (4 bits)	0 ~ 15
K2 (8 bits)	0 ~ 255	K2 (8 bits)	0 ~ 255
K3 (12 bits)	0 ~ 4,095	K3 (12 bits)	0 ~ 4,095
K4 (16 bits)	-32,768 ~ +32,767	K4 (16 bits)	0 ~ 65,535
		K5 (20 bits)	0 ~ 1,048,575
		K6 (24 bits)	0 ~ 167,772,165
		K7 (28 bits)	0 ~ 268,435,455
		K8 (32 bits)	-2,147,483,648 ~ +2,147,483,647

◆ Flags

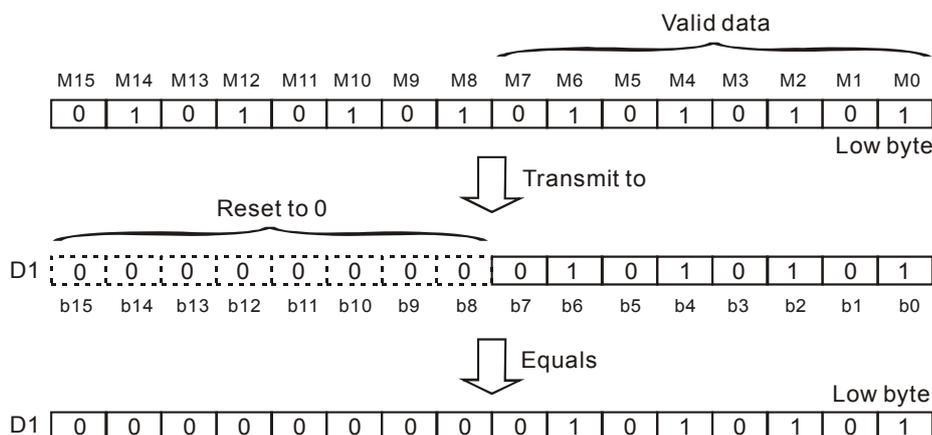
The flags listed below are used for indicating the operation result of the application instruction:

M1968: zero flag M1969: borrow flag M1970: carry flag

All flags will turn ON or OFF according to the operation result of an instruction. For example, the execution result of instructions ADD/SUB/MUL/DVI in O100 main program will affect the status of M1968 ~ M1970. When the instruction is not executed, the ON/OFF status of the flag will be held. The status of the four flags relates to many instructions. See relevant instructions for more details.

5.3 Numeric Values for Data Processing

- ◆ Devices indicates ON/OFF status are called bit devices, e.g. X, Y, M and S. Devices used for storing values are called word devices, e.g. T, C, D, E and F. Although bit device can only be ON/OFF for a single point, they can also be used as numeric values in the operands of instructions if the data type declaration device Kn is added in front of the bit device.
- ◆ For 16-bit data, K1~K4 are applicable. For 32-bit data, K1~K8 are applicable. For example, K2M0 refers to a 8-bit value composed of M0 ~ M7.



- ◆ Transmit K1M0, K2M0, K3M0 to 16-bit registers. ONLY the valid bit data will be transmitted and the upper bits in

5 Categories and Use of Basic Application Instructions

the 16-bit register will all be filled with 0. The same rule applies when sending K1M0, K2M0, K3M0, K4M0, K5M0, K6M0, K7M0 to 32-bit registers.

- ◆ When the Kn value is specified as K1~K3 (K4~K7) for a 16-bit (32-bit) operation, the empty upper bits of the target register will be filled with “0.” Therefore, the operation result in this case is positive since the MSB(Most significant bit) is 0.



- ◆ When assigning the number of bit devices, a multiple of 10 is preferred, e.g. X0, X10, ...Y0, Y10...M0, M10, etc.
- ◆ Assign Continuous Bit Devices

As already explained, bit devices can be grouped into 4 bit units. The “n” in Kn defines the number of groups of 4 bits to be combined for data operation. For data register D, consecutive D refers to D0, D1, D2, D3, D4...; For bit devices with Kn, consecutive No. refers to:

K1X0	K1X4	K1X10	K1X14.....
K2Y0	K2Y10	K2Y20	Y2X30.....
K3M0	K3M12	K3M24	K3M36.....
K4S0	K4S16	K4S32	K4S48.....

Note: To avoid errors, please follow the order listed above when assigning continuous bit devices. In addition, when K4Y0 is used in 32-bit operation, the high word is defined as 0. Therefore, it is recommended to use K8Y0 in 32bit operation.

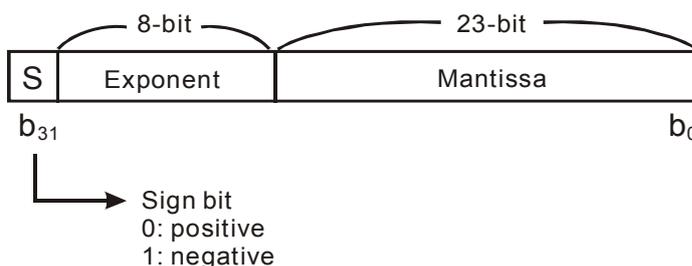
- ◆ Floating Point Operation

The operations in DVP-PM are conducted in BIN integers. When the integer performs division, e.g. $40 \div 3 = 13$, the remainder will be 1. When the integer performs square root operations, the decimal point will be left out. To obtain the operation result with decimal point, please use floating point instructions.

API 110 (DECMP)	API 111 (DEZCP)	API 116 (DRAD)	API 117 (DDEG)
API 120 (DEADD)	API 121 (DESUB)	API 122 (DEMUL)	API 123 (DEDIV)
API 124 (DEXP)	API 125 (DLN)	API 126 (DLOG)	API 127 (DESQR)
API 128 (DPOW)	API 129 (DINT)	API 130 (DSIN)	API 131 (DCOS)
API 133 (DASIN)	API 133 (DASIN)	API 134 (DACOS)	API 135 (DATAN)
API 136 (DSINH)	API 137 (DCOSH)	API 138 (DTANH)	

Binary Floating Point

DVP-PM represents floating point value in 32 bits, following the IEEE754 standard:



5 Categories and Use of Basic Application Instructions

Equation $(-1)^S \times 2^{E-B} \times 1.M; B = 127$

Therefore, the range of 32-bit floating point value is from $\pm 2^{-126}$ to $\pm 2^{+128}$, i.e. from $\pm 1.1755 \times 10^{-38}$ to $\pm 3.4028 \times 10^{+38}$

Example 1: Represent “23” in 32-bit floating point value

Step 1: Convert “23” into a binary value: $23.0 = 10111$

Step 2: Normalize the binary value: $10111 = 1.0111 \times 2^4$, in which 0111 is mantissa and 4 is exponent.

Step 3: Obtain the exponent: $\because E - B = 4 \rightarrow E - 127 = 4 \therefore E = 131 = 10000011_2$

Step 4: Combine the sign bit, exponent and mantissa into a floating point

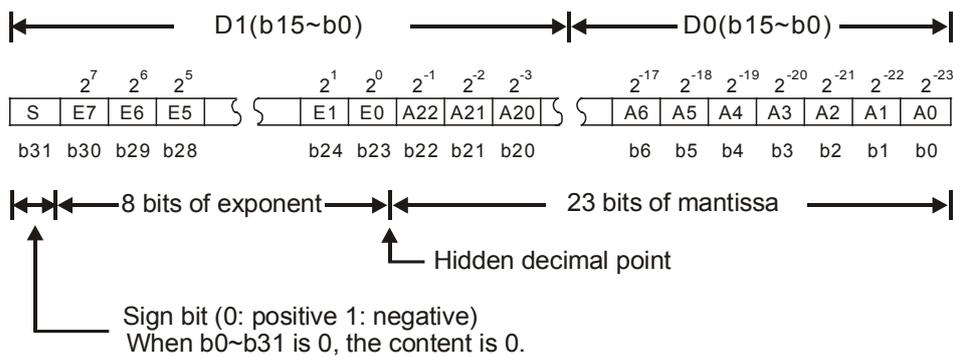
$$0\ 10000011\ 01110000000000000000000_2 = 41B80000_{16}$$

Example 2: Represent “-23.0” in 32-bit floating point value

The steps required are the same as those in Example 1 and only differs in modifying the sign bit into “1”.

$$1\ 10000011\ 01110000000000000000000_2 = C1B80000_{16}$$

DVP-PM uses registers of 2 continuous No. to store a 32-bit floating point value. For example, we use registers (D1, D0) for storing a binary floating point value as below:



Decimal Floating Point

- ◆ Since the binary floating point value is not very user-friendly, we can convert it into a decimal floating point value for use. However, the floating point operation in DVP-PM is still operated in binary floating point format..
- ◆ The decimal floating point is represented by 2 continuous registers. The register of smaller number is for the constant while the register of bigger number is for the exponent.

Example: Store a decimal floating point in registers (D1, D0)

$$\text{Decimal floating point} = [\text{constant D0}] \times 10^{[\text{exponent D1}]}$$

Constant D0 = $\pm 1,000 \sim \pm 9,999$

Exponent D1 = $-41 \sim +35$

Note: The constant 100 does not exist in D0 because 100 is represented as $1,000 \times 10^{-1}$. The range of decimal floating point is $\pm 1175 \times 10^{-41} \sim \pm 3402 \times 10^{+35}$..

- ◆ The decimal floating point value can be used in the following instructions:
 - DEBCD: Convert binary floating point into decimal floating point
 - DEBIN: Convert decimal floating point into binary floating point
- ◆ In O100 ~ M102 main programs, the execution result will affect the status of M1968 ~ M1970 when using ADD/SUB/MUL/DIV instructions. Zero flag (M1968), borrow flag (M1970) and carry flag (M1969) will also be

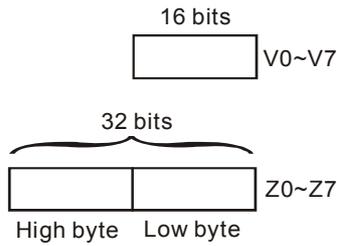
5 Categories and Use of Basic Application Instructions

affected by the execution result of floating point instructions:

- Zero flag: M1968 = ON if the operational result is “0”.
- Borrow flag: M1970 = ON if the operational result exceeds the minimum unit.
- Carry flag: M1969 = ON if the absolute value of the operational result exceeds the range of use.

5.4 Index Register V, Z

The index registers are 16-bit registers. V is 16-bit register, and Z is 32-bit register. There are 16 devices for DVP-PM (V0 ~ V7 and Z0 ~ Z7).



V is 16-bit register, which can be read and written. If you need a 32-bit register, you have to designate Z.

Devices modifiable in DVP-PM: P, I, X, Y, M, S, KnX, KnY, KnM, KnS, T, C, D.

V and Z can modify the devices listed above but cannot modify themselves and Kn, e.g. K4M0@Z0 is valid and K0@Z0M0 is invalid. In addition, V and Z cannot modify constant K and H, e.g. K4@Z0 is invalid. Grey columns in the table of operand at the beginning page of each application instruction indicate the operands modifiable by V and Z.

When you use instruction mode in PMSoft to modify constant K and H, you have to use @, for example, “MOV K10@Z0 D0V0”.

5 Categories and Use of Basic Application Instructions

5.5 Instruction Index

Sorted by alphabetic order:

	API	Mnemonic		P Instruction	Function	STEPS		Page
		16-bit	32-bit			16-bit	32-bit	
A	20	ADD	DADD	✓	Addition	7	9	5-34
	46	ANS	-	-	Timed Annunciator Set	7	-	5-64
	47	ANR	-	✓	Annunciator Reset	1	-	5-65
	66	ALT	-	✓	Alternate state	3	-	5-73
	87	ABS	DABS	✓	Absolute value	3	5	5-82
	92	ANDP	-	-	Rising-Edge Series Connection	3	-	4-9
	93	ANDF	-	-	Falling-Edge Series Connection	3	-	4-10
	133	-	DASIN	✓	Arc Sine	-	6	5-114
	134	-	DACOS	✓	Arc Cosine	-	6	5-115
	135	-	DATAN	✓	Arc Tangent	-	6	5-74
	218	AND&	DAND&	-	S1 & S2	5	7	5-125
	219	AND	DAND	-	S1 S2	5	7	5-125
	220	AND^	DAND^	-	S1 ^ S2	5	7	5-125
	232	AND=	DAND=	-	S1 = S2	5	7	5-128
	233	AND>	DAND>	-	S1 > S2	5	7	5-128
	234	AND<	DAND<	-	S1 < S2	5	7	5-128
	236	AND<>	DAND<>	-	S1 ≠ S2	5	7	5-128
	237	AND<=	DAND<=	-	S1 ≤ S2	5	7	5-128
238	AND>=	DAND>=	-	S1 ≥ S2	5	7	5-128	
B	15	BMOV	-	✓	Block move	7	-	5-28
	18	BCD	DBCD	✓	Convert BIN to BCD	5	5	5-32
	19	BIN	DBIN	✓	Convert BCD to BIN	5	5	5-33
	44	BON	DBON	✓	Check specified bit status	7	13	5-62
	258	BRET	-	-	Return to Bus Line	1	-	5-140
C	00	CJ	-	✓	Conditional Jump	3	-	5-13
	01	CALL	-	✓	Call Subroutine	3	-	5-16
	10	CMP	DCMP	✓	Compare	7	9	5-21
	14	CML	DCML	✓	Complement	5	9	5-27
	97	CNT	DCNT	-	16-bit/32-bit Counter	5	6	4-8
	131	-	DCOS	✓	Cosine	5	6	5-110
	137	-	DCOSH	✓	Hyperbolic Cosine	-	6	5-118
	256	CJN	-	✓	Negated Conditional Jump	3	-	5-138
D	23	DIV	DDIV	✓	Division	7	9	5-38
	25	DEC	DDEC	✓	Decrement	3	3	5-40
	41	DECO	-	✓	Decode	7	-	5-58
	117	-	DDEG	✓	Radian → Degree	-	6	5-96
E	42	ENCO	-	✓	Encode	7	-	5-59
	110	-	DECMP	✓	Floating Point Compare	7	9	5-92
	111	-	DEZCP	✓	Floating Point Zone Compare	9	12	5-52
	112	-	DMOV	✓	Move floating point data	-	9	5-94

5 Categories and Use of Basic Application Instructions

	API	Mnemonic		P Instruction	Function	STEPS		Page
		16-bit	32-bit			16-bit	32-bit	
	120	-	DEADD	✓	Floating Point Addition	7	9	5-97
	121	-	DESUB	✓	Floating Point Subtraction	7	9	5-98
	122	-	DEMUL	✓	Floating Point Multiplication	7	9	5-99
	123	-	DEDIV	✓	Floating Point Division	7	9	5-100
	124	-	DEXP	✓	Float Exponent Operation	-	6	5-101
	127	-	DESQR	✓	Floating Point Square Root	5	6	5-104
	172	-	DADDR	✓	Floating point addition	-	13	5-120
	173	-	DSUBR	✓	Floating point subtraction	-	13	5-121
	174	-	DMULR	✓	Floating point multiplication	-	13	5-122
	175	-	DDIVR	✓	Floating point division	-	13	5-123
F	16	FMOV	DFMOV	✓	Fill move	7	13	5-30
	49	-	DFLT	✓	Floating Point	-	6	5-68
	78	FROM	DFROM	✓	Read CR Data from Special Modules	9	12	5-78
I	24	INC	DINC	✓	Increment	3	3	5-39
	129	-	DINT	✓	Float to Integer	-	6	5-107
J	257	JMP	-	-	Unconditional Jump	3	-	5-139
L	90	LDP	-	-	Rising-Edge Detection Operation	3	-	4-9
	91	LDF	-	-	Falling-Edge Detection Operation	3	-	4-9
	125	-	DLN	✓	Float Natural Logarithm Operation	-	6	5-102
	126	-	DLOG	✓	Float Logarithm Operation	-	9	5-103
	215	LD&	DLD&	-	S1 & S2	5	7	5-124
	216	LD	DLD	-	S1 S2	5	7	5-124
	217	LD^	DLD^	-	S1 ^ S2	5	7	5-124
	224	LD=	DLD=	-	S1 = S2	5	7	5-127
	225	LD>	DLD>	-	S1 > S2	5	7	5-127
	226	LD<	DLD<	-	S1 < S2	5	7	5-127
	228	LD<>	DLD<>	-	S1 ≠ S2	5	7	5-127
	229	LD<=	DLD<=	-	S1 ≤ S2	5	7	5-127
230	LD>=	DLD>=	-	S1 ≥ S2	5	7	5-127	
M	12	MOV	DMOV	✓	Move	5	6	5-23
	22	MUL	DMUL	✓	Multiplication	7	9	5-37
	45	MEAN	DMEAN	✓	Mean	7	13	5-63
	100	MODRD	-	-	Read Modbus Data	7	-	5-83
	101	MODWR	-	-	Write Modbus Data	7	-	5-87
	259	MMOV	-	✓	16-bit → 32-bit Conversion	6	-	5-141
N	29	NEG	DNEG	✓	2's Complement (Negation)	3	3	5-44
O	94	ORP	-	-	Rising-Edge Parallel Connection	3	-	4-10
	95	ORF	-	-	Falling-Edge Parallel Connection	3	-	4-11
	221	OR&	DOR&	-	S1 & S2	5	7	5-126
	222	OR	DOR	-	S1 S2	5	7	5-126
	223	OR^	DOR^	-	S1 ^ S2	5	7	5-126
	240	OR=	DOR=	-	S1 = S2	5	7	5-127
	241	OR>	DOR>	-	S1 > S2	5	7	5-127
	242	OR<	DOR<	-	S1 < S2	5	7	5-127

5 Categories and Use of Basic Application Instructions

	API	Mnemonic		P Instruction	Function	STEPS		Page
		16-bit	32-bit			16-bit	32-bit	
	244	OR<>	DOR<>	-	S1 ≠ S2	5	7	5-127
	245	OR<=	DOR<=	-	S1 ≤ S2	5	7	5-127
	246	OR>=	DOR>=	-	S1 ≥ S2	5	7	5-127
P	89	PLS	-	-	Rising-Edge Output	3	-	4-11
	99	PLF	-	-	Falling-Edge Output	3	-	4-12
	128	-	DPOW	✓	Floating Point Power Operation	-	9	5-105
R	08	RPT	-	-	Start of a Nested RPT-RPE loop (Only 1 level available)	3	-	5-19
	09	RPE	-	-	End of a Nested RPT-RPE loop	1	-	5-19
	30	ROR	DROR	✓	Rotate right	5	9	5-46
	31	ROL	DROL	✓	Rotate left	5	9	5-47
	32	RCR	DRCR	✓	Rotate right with carry	5	9	5-48
	33	RCL	DRCL	✓	Rotate left with carry	5	9	5-49
	50	REF	-	✓	Refresh	5	-	5-70
	67	RAMP	DRAMP	-	Ramp variable value	9	17	5-74
	116	-	DRAD	✓	Degree → Radian	-	6	5-95
	154	RAND	DRAND	✓	Random number	7	13	5-131
	260	RMOV	-	✓	32-bit → 16-bit Conversion	6	-	5-142
S	02	SRET	-	-	Subroutine Return	1	-	5-16
	13	SMOV	-	✓	Shift move	11	-	5-24
	21	SUB	DSUB	✓	Subtraction	7	9	5-36
	34	SFTR	-	✓	Bit shift right	9	-	5-50
	35	SFTL	-	✓	Bit shift left	9	-	5-51
	38	SFWR	-	✓	Shift register write	7	-	5-55
	39	SFRD	-	✓	Shift register read	7	-	5-56
	43	SUM	DSUM	✓	Sum of Active bits	5	9	5-61
	48	SQR	DSQR	✓	Square Root	5	9	5-67
	61	SER	DSER	✓	Search a data stack	9	17	5-72
	69	SORT	DSORT	-	Data sort	11	21	5-76
	147	SWAP	DSWAP	✓	Byte swap	3	5	5-130
	130	-	DSIN	✓	Sine	5	6	5-108
	136	-	DSINH	✓	Hyperbolic Sine	-	6	5-117
	202	SCAL	-	✓	Proportional value calculation	9	-	5-132
203	SCLP	DSCLP	✓	Parameter proportional value calculation	7	13	5-134	
T	79	TO	DTO	✓	Write CR Data into Special Modules	9	13	5-79
	96	TMR	-	-	16-bit Timer	5	-	4-7
	132	-	DTAN	✓	Tangent	5	6	5-112
	138	-	DTANH	✓	Hyperbolic Tangent	-	6	5-119
W	07	WDT	-	✓	Watchdog timer refresh	1	-	5-18
	26	WAND	DWAND	✓	Logical Word AND	7	9	5-41
	27	WOR	DWOR	✓	Logical Word OR	7	9	5-42
	28	WXOR	DWXOR	✓	Logical XOR	7	9	5-43
	36	WSFR	-	✓	Word shift right	9	-	5-52
	37	WSFL	-	✓	Word shift left	9	-	5-54

5 Categories and Use of Basic Application Instructions

	API	Mnemonic		P Instruction	Function	STEPS		Page
		16-bit	32-bit			16-bit	32-bit	
X	17	XCH	DXCH	✓	Exchange	5	9	5-31
Z	11	ZCP	DZCP	✓	Zone Compare	9	12	5-22
	40	ZRST	-	✓	Zone Reset	5	-	5-57

5 Categories and Use of Basic Application Instructions

5.6 Application Instructions

API	Mnemonic		Operands	Function	Controllers	
					20PM	10PM
00	CJ	P	(S)	Conditional Jump		
OP	Range			Program Steps		
(S)	P0 ~ P255			CJ, CJP: 3 steps		

Operands:

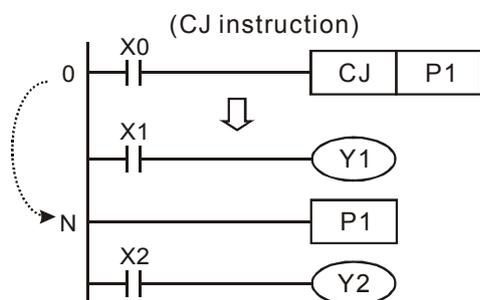
S: The destination pointer P of the conditional jump.

Explanations:

- P cannot be modified by index registers V, Z.
- If you need to skip a particular part of PLC program to shorten the scan time or execute dual outputs, CJ instruction or CJ P instruction can be adopted.
- When the program designated by pointer P is prior to CJ instruction, WDT timeout will occur and PLC will stop running. Please use it carefully.
- CJ instruction can designate the same pointer P repeatedly. However, CJ and CALL cannot designate the same pointer P; otherwise operation error will occur.
- Actions of all devices while conditional jump is being executed:
 - Y, M and S remain their previous status before the conditional jump takes place.
 - The executing 10ms timer stops.
 - General counters stop executing and general application instructions are not executed.
 - If timer is reset before CJ instruction executes, the timer will still be in the reset status while CJ instruction is being executed.

Program Example 1:

- When X0 = ON, the program will skip from address 0 to N (Pointer P1) automatically and keep on executing. Instructions between address 0 and N will be skipped..
- When X0 = OFF, program flow will proceed with the row immediately after the CJ instruction.



Program Example 2:

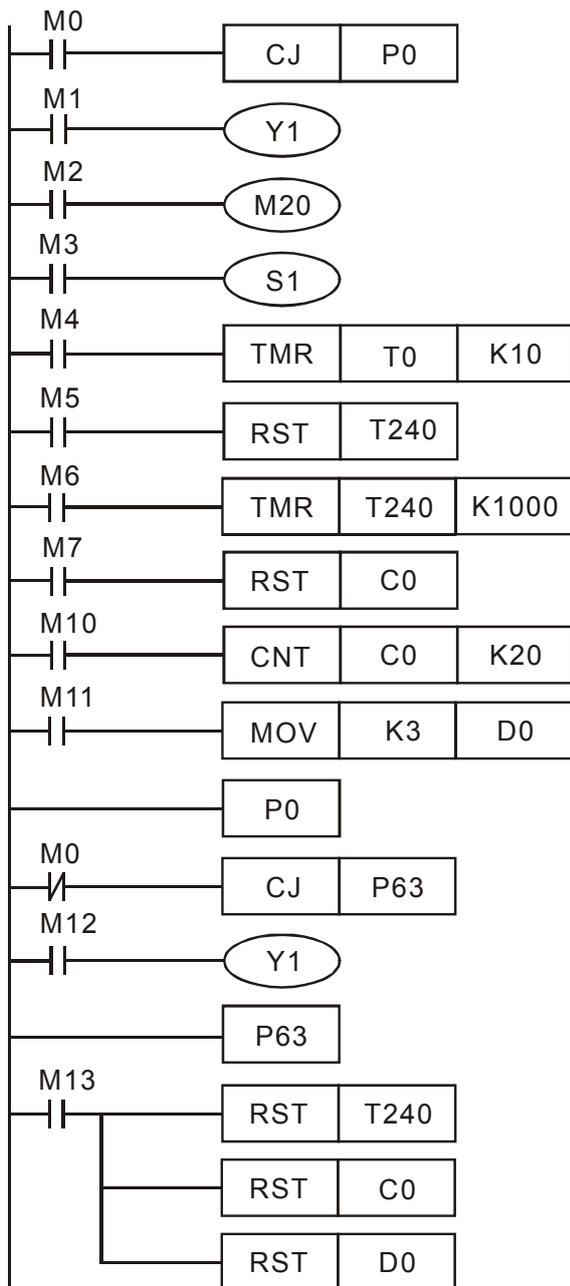
The table explains the device status in the ladder diagram below.

5 Categories and Use of Basic Application Instructions

Device	Contact state before CJ execution	Contact state during CJ execution	Output coil state during CJ execution
Y, M, S	M1, M2, M3 OFF	M1, M2, M3 OFF → ON	Y1*1, M20, S1 OFF
	M1, M2, M3 ON	M1, M2, M3 ON → OFF	Y1*1, M20, S1 ON
10ms Timer	M4 OFF	M4 OFF → ON	Timer T0 is not enabled.
	M4 ON	M4 ON → OFF	Timer T0 immediately stops and is latched. When M0 ON → OFF, T0 will be reset.
	M6 OFF	M6 OFF → ON	Timer T240 is not enabled.
	M6 ON	M6 ON → OFF	Timer T240 immediately stops and is latched. When M0 ON → OFF, T240 will be reset.
C0 ~ C234	M7, M10 OFF	M10 is ON/OFF triggered	Counter C0 stops
	M7 OFF, M10 is ON/OFF triggered	M10 is ON/OFF triggered	Counter C0 stops and latched. When M0 is OFF, C0 resumes counting.
Application instruction	M11 OFF	M11 OFF → ON	Application instructions will not be executed.
	M11 ON	M11 ON → OFF	The skipped application instruction will not be executed but API 53~59, API 157~159 keep executing.

*1: Y1 is dual output. When M0 is OFF, Y1 is controlled by M1. When M0 is ON, Y1 is controlled by M12.

5 Categories and Use of Basic Application Instructions



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands	Function	Controllers	
					20PM	10PM
01	CALL	P	(S)	Call Subroutine		
OP	Range			Program Steps		
(S)	P0 ~ P255			CALL, CALLP: 3 steps		

Operands:

S: The destination pointer P of the call subroutine

Explanations:

1. When the CALL instruction is active it forces the program to run the subroutine associated with the called pointer
2. P cannot be modified by index register V, Z.
3. The subroutine designated by the pointer should be programmed after M102, M2 and SRET instructions.
4. The number of pointer P, when used by CALL, cannot be the same as the number designated by CJ, CJN and JMP instructions.
5. If only CALL instruction is in use, it can call subroutines of the same pointer number with no limits on the times.
6. CALL instruction is not allowed to be used in subroutines.

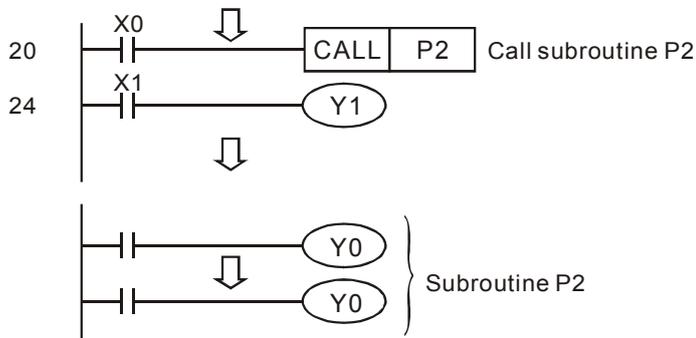
API	Mnemonic	Function	Controllers	
			20PM	10PM
02	SRET	Subroutine Return		
OP	Descriptions		Program Steps	
N/A	No operand. No contact to drive the instruction is required		SRET: 1 steps	

Explanations:

SRET indicates the end of subroutine program. When SRET is executed, program execution will automatically return to the address after CALL instruction in O100 main program.

Program Example 1:

When X0 = ON, the CALL instruction will jump to P2 and run the subroutine. With the execution of the SRET instruction, it will jump back to address 24 and continue the execution.

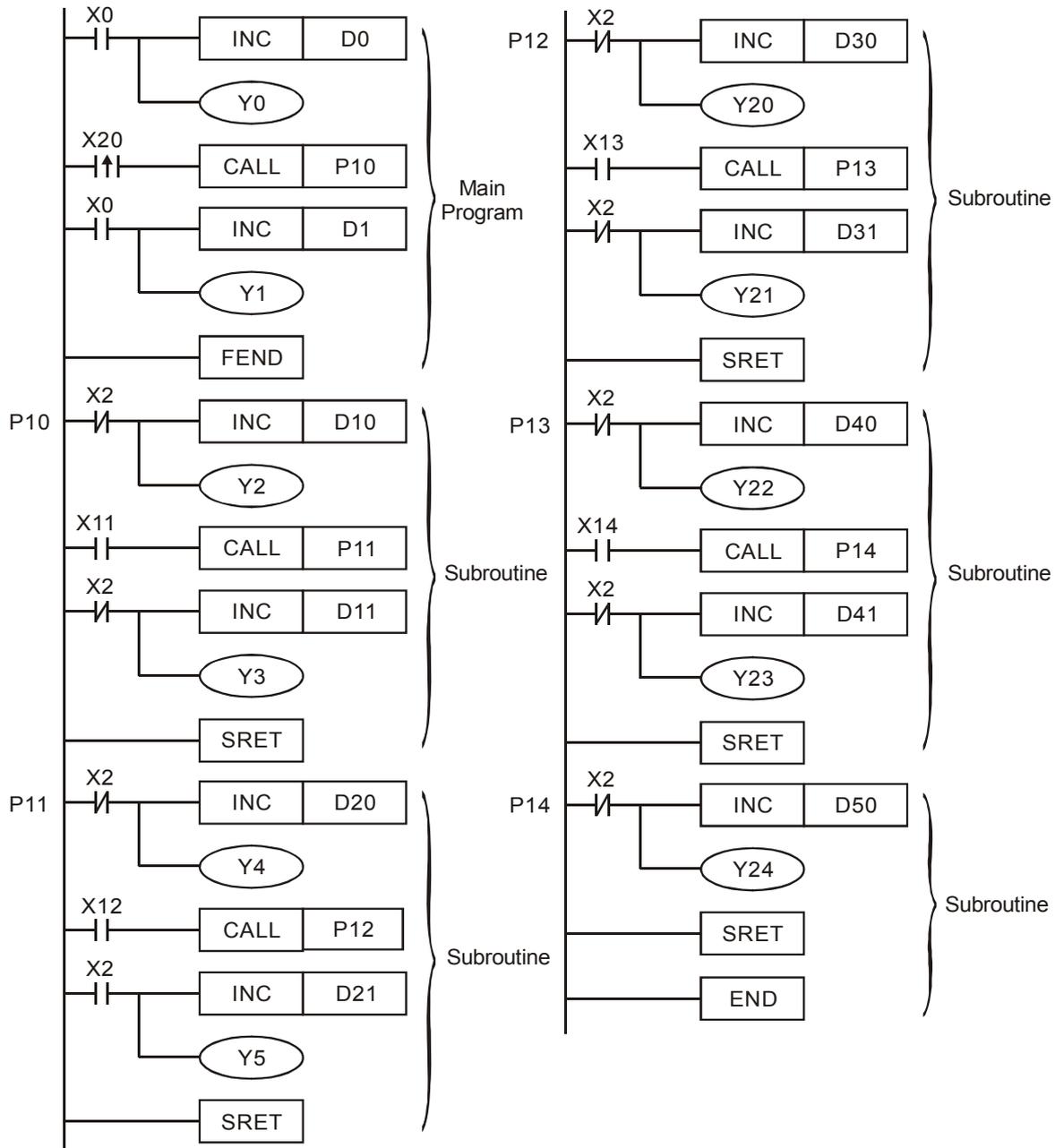


Program example 2:

1. When the rising-edge contact X20 is triggered, CALL P10 instruction will transfer execution to subroutine P10.
2. When X11 is ON, execute CALL P11, jump to and run subroutine P11.
3. When X12 is ON, execute CALL P12, jump to and run subroutine P12.

5 Categories and Use of Basic Application Instructions

4. When X13 is ON, execute CALL P13, jump to and run subroutine P13.
5. When X14 is ON, execute CALL P14, jump to and run subroutine P14. When the SRET instruction is reached, jump back to the last P subroutine to finish the remaining instructions.
6. The execution of subroutines will go backwards to the subroutine of upper level until SRET instruction in P10 subroutine is executed. After this program execution will return to the main program.

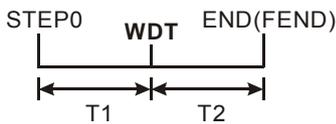


5 Categories and Use of Basic Application Instructions

API	Mnemonic		Function	Controllers	
	07	WDT		P	Watchdog Timer Refresh
OP	Descriptions			Program Steps	
N/A	No operand. No contact to drive the instruction is required			WDT, WDTP: 1 step	

Explanations:

- WDT instruction can be used to reset the Watch Dog Timer. If the PLC scan time (from address 0 to END or FEND instruction) is more than 200ms, the ERROR LED will flash. In this case, users have to turn the power OFF and then ON to clear the fault. PLC will determine the status of RUN/STOP according to RUN/STOP switch. If there is no RUN/STOP switch, PLC will return to STOP status automatically.
- Time to use WDT:
 - When error occur in PLC system.
 - When the scan time of the program exceeds the WDT value in D1000. It can be modified by using the following two methods.
 - Use WDT instruction



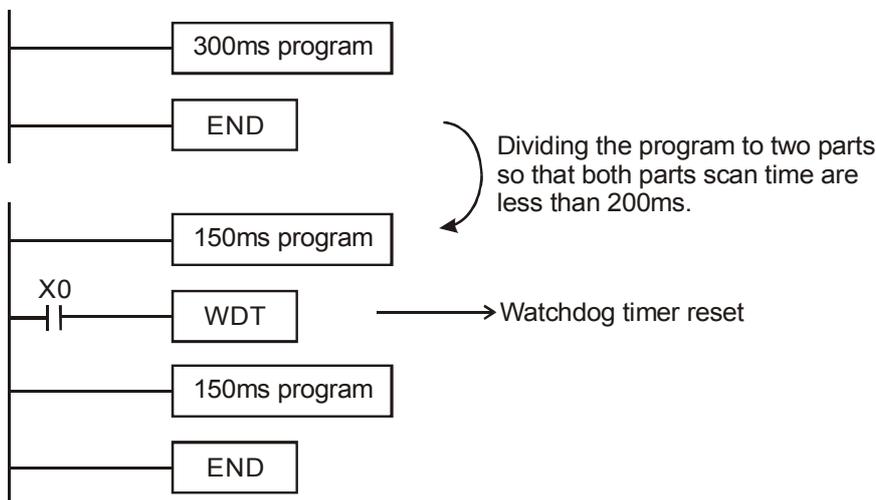
- Use the set value in D1000 (Default: 200ms) to change the time for watchdog.

Points to note:

- When the WDT instruction is used it will operate on every program scan as long as its input condition has been made. To force the WDT instruction to operate for only ONE scan, users have to use the pulse (P) format of the WDT instruction, i.e. WDTP.
- The watchdog timer has a default setting of 200ms. This time limit can be customized to users requirement by editing the content in D1000, the wathdog timer register.

Program example:

If the program scan time is over 300ms, users can divide the program into 2 parts. Insert the WDT instruction in between, making scan time of the first half and second half of the program being less than 200ms.



5 Categories and Use of Basic Application Instructions

API	Mnemonic	Operands	Function	Controllers													
08	RPT	(S)	Start of a Nested RPT-RPE loop	20PM	10PM												
OP	Type	Bit Devices		Word Devices							Program Steps						
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	RPT: 3 steps
S					*	*	*	*	*	*	*	*	*	*	*		

Operands:

S: The number of times for the loop to be repeated

Explanations:

1. No contact to drive the instruction is required.
2. The nested RPT ~ RPE loop can only be 1 level. Errors will occur when the number of levels is more than 1.

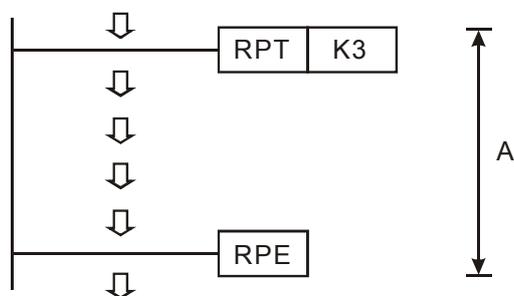
API	Mnemonic	Function	Controllers	
09	RPE	End of a Nested RPT-RPE loop	20PM	10PM
OP	Descriptions		Program Steps	
N/A	No operand. No contact to drive the instruction is required.		RPE: 1 steps	

Explanations:

1. RPT instruction executes the RPT-RPE loop back and forth for “N” times then return to the instruction after the RPT-RPE loop.
2. $N = K1 \sim K32,767$. N is regarded as K1 when $N \leq K1$.
3. CJ instruction can be used to skip or jump out of RPT-RPE loop.
4. An error will occur in the following conditions:
 - RPE instruction is placed before RPT instruction.
 - RPT instruction exists, but RPE instruction does not exist.
 - Number of RPT instructions differs from that of RPE instructions.
5. The nested RPT ~ RPE loop can only be 1 level. Errors will occur when the number of levels is more than 1.

Program Example 1:

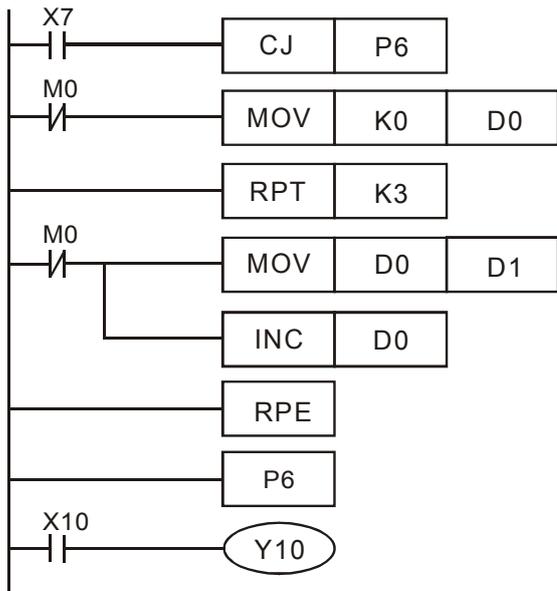
If you would like program section A to execute for 3 times, you can use RPT-RPE written as follows:



Program Example 2:

When X7 = OFF, PLC will execute the program between RPT and RPE. When X7 = ON, CJ instruction jumps to P6 and avoids executing the instructions between RPT and RPE.

5 Categories and Use of Basic Application Instructions



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function			Controllers	
	10	D	CMP	P	S ₁	S ₂	D	Compare			20PM

OP	Type	Bit Devices				Word Devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z			
S ₁					*	*	*	*	*	*	*	*	*	*	*			CMP, CMPP: 7 steps
S ₂					*	*	*	*	*	*	*	*	*	*	*			DCMP, DCMPP: 9 steps
D		*	*	*														

Operands:

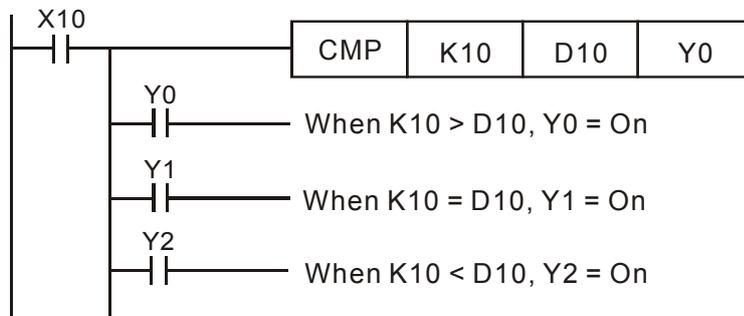
S₁: Comparison value 1 S₂: Comparison value 2 D: Comparison result

Explanations:

- The contents of S₁ and S₂ are compared and D stores the comparison result.
- When CMP is used as 16-bit instruction, Z device cannot be adopted; when CMP is used as 32-bit instruction, V device cannot be adopted.
- D will occupy 3 consecutive devices. D, D + 1, D + 2 hold the comparison results: D = ON if S₁ > S₂, D + 1 = ON if S₁ = S₂, D + 2 = ON if S₁ < S₂.

Program Example:

- If D is set as Y0, then Y0, Y1, Y2 will display the comparison results as shown below..
- When X10 = ON, CMP instruction is executed and one of Y0, Y1, Y2 will be ON. When X10 = OFF, CMP instruction is not executed and Y0, Y1, Y2 remain in their previous condition.
- If you need to obtain a comparison result with \geq , \leq and \neq , make a series/parallel connection between Y0 and Y2.



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands				Function				Controllers	
	D	ZCP	P	(S ₁)	(S ₂)	(S)	(D)	Zone Compare				20PM	10PM
11	D	ZCP	P	(S ₁)	(S ₂)	(S)	(D)	Zone Compare				20PM	10PM

OP	Type	Bit Devices				Word Devices										Program Steps		
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z		
S ₁						*	*	*	*	*	*	*	*	*	*	*	ZCP, ZCPP: 9 steps	
S ₂						*	*	*	*	*	*	*	*	*	*	*	DZCP, DZCPP: 12 steps	
S						*	*	*	*	*	*	*	*	*	*	*		
D		*	*	*														

Operands:

S₁: Lower bound of zone comparison **S₂**: Upper bound of zone comparison **S**: Comparison value

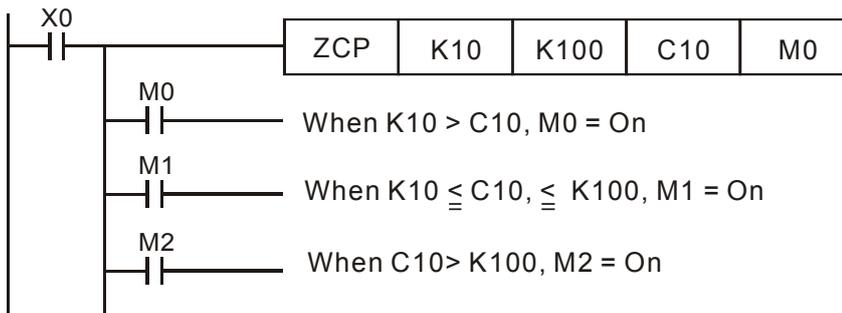
D: Comparison result

Explanations:

- S** is compared with its lower bound **S₁** and upper bound **S₂** and **D** stores the comparison results.
- ZCP instruction supports V and Z. When ZCP is used as 16-bit instruction, Z device cannot be adopted; when ZCP is used as 32-bit instruction, V device cannot be adopted.
- Operand **S₁** should be smaller than operand **S₂**. When **S₁ > S₂**, the instruction takes **S₁** as the 1st comparison value and performs normal comparison similar to CMP instruction.
- Operand **D** occupies 3 consecutive devices. **D**, **D + 1**, **D + 2** hold the comparison results, **D = ON** if **S₁ > S**, **D + 1 = ON** if **S₁ ≤ S ≤ S₂**, **D + 2 = ON** if **S₂ < S**.

Program Example:

- If **D** is set as M0, then M0, M1, M2 will work as the program example below.
- When X0 = ON, ZCP instruction is driven and one of M0, M1, M2 is ON. When X0 = OFF, ZCP instruction is not driven and M0, M1, M2 remain in the previous status.



- Use RST or ZRST instruction to reset the comparison results

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers			
	D	MOV	P	S	D	Move										20PM	10PM		
12																			
OP	Type	Bit Devices				Word Devices										Program Steps			
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z			
S						*	*	*	*	*	*	*	*	*	*	*	MOV, MOV _P : 5 steps		
D								*	*	*	*	*	*	*	*	DMOV, DMOV _P : 6 steps			

Operands:

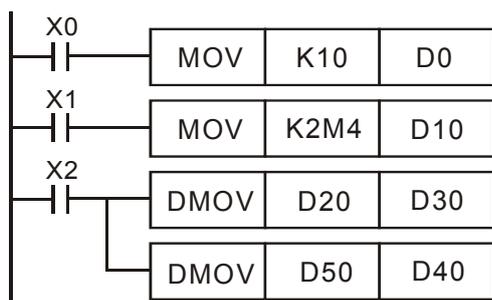
S: Source of data **D:** Destination of data

Explanations:

- When this instruction is executed, the content of **S** will be moved directly to **D**. When this instruction is not executed, the content of **D** remains unchanged
- When MOV is used as 16-bit instruction, Z device cannot be adopted; when MOV is used as 32-bit instruction, V device cannot be adopted.
- For moving 32-bit operation results of certain instructions (e.g. MUL), **DMOV** instruction has to be applied.

Program Example:

- MOV will move a 16-bit value from the source location to the destination.
 - When X0 = OFF, the content of D0 remains unchanged. If X0 = ON, the data K10 is moved to D0.
 - When X1 = OFF, the content of D10 remains unchanged. If X1 = ON, the data of K2M4 is moved to D10 data register.
- DMOV** will move a 32-bit value from the source location to the destination.
 - When X2 = OFF, the content of (D31, D30) and (D41, D40) remain unchanged.
 - When X2 = ON, the data of (D21, D20) is moved to (D31, D30) data register. Meanwhile, the data of C235 is moved to (D41, D40) data register.



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands					Function					Controllers				
13	SMOV	P	(S)	(m ₁)	(m ₂)	(D)	(n)	Shift Move					20PM	10PM			
OP	Type	Bit Devices				Word devices										Program Steps	
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SMOV, SMOVP: 11 steps
S								*	*	*	*	*	*	*	*	*	
m ₁					*	*											
m ₂					*	*											
D								*	*	*	*	*	*	*	*	*	
n					*	*											

Operands:

S: Source device **m₁:** Start digit to be moved from source device **m₂:** Number of digits to be moved **D:** Destination device **n:** Start digit of the destination device for the moved digits

Explanation:

- This instruction is able to re-allocate or combine data. When the instruction is executed, **m₂** digits of contents starting from digit **m₁** (from high digit to low digit) of **S** will be sent to **m₂** digits starting from digit **n** (from high digit to low digit) of **D**
- When SMOV is used as 16-bit instruction, Z device cannot be adopted; when SMOV is used as 32-bit instruction, V device cannot be adopted.
- When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
- M1168 is used for designating SMOV working mode. When M1168 = ON, the instruction is in BIN mode. When M1168 = OFF, the instruction is in BCD mode.

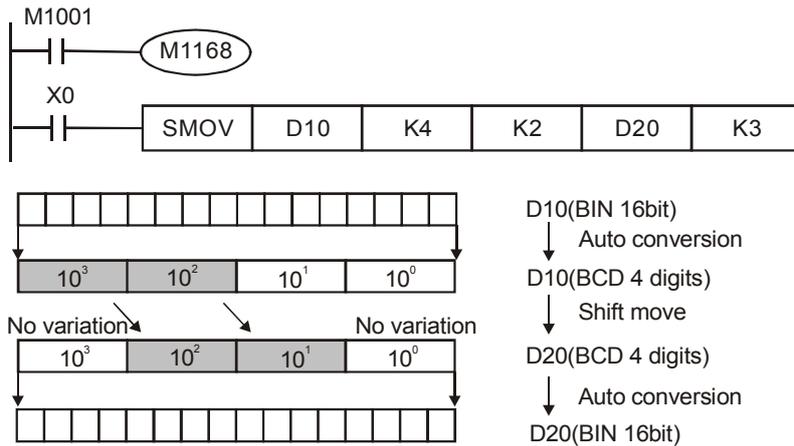
Points to note:

- The range of **m₁**: 1 – 4
- The range of **m₂**: 1 – **m₁**
- The range of **n**: **m₂** – 4

Program example 1:

- When M1168 = OFF (in BCD mode) and X0 = ON, the 4th (thousand) and 3rd (hundred) digit of the decimal value in D10 start to move to the 3rd (hundred) and 2nd (ten) digit of the decimal value in D20. 10³ and 10⁰ of D20 remain unchanged after this instruction is executed.
- When the BCD value exceeds the range of 0 ~ 9,999, PLC detects an operation error and will not execute the instruction. M1067, M1068 = ON and D1067 stores the error code OE18 (hex).

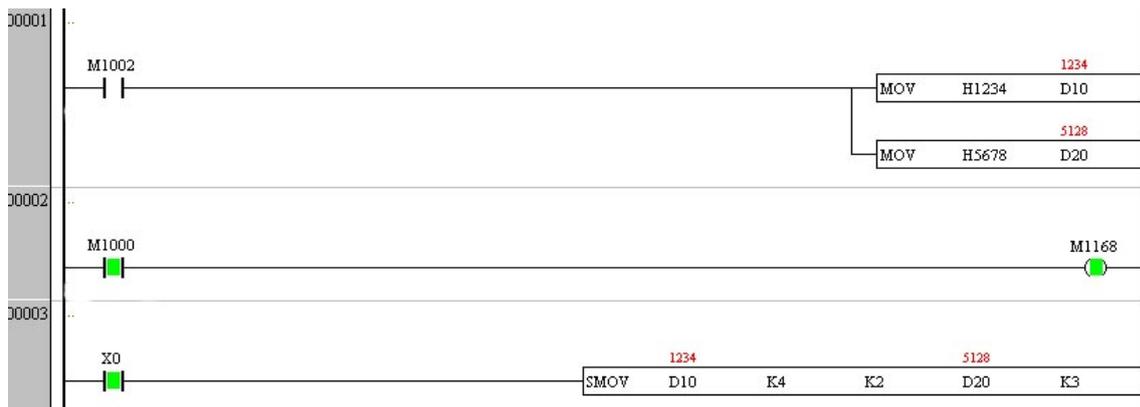
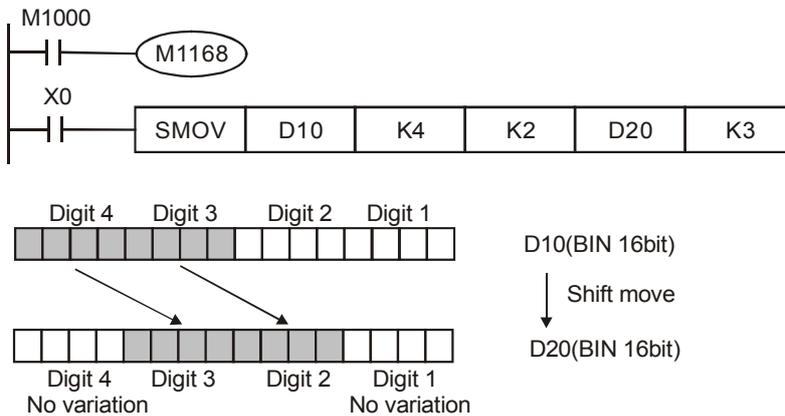
5 Categories and Use of Basic Application Instructions



If D10 = K1234, D20 = K5678 before execution, D10 remains unchanged and D20 = K5128 after execution.

Program example 2:

When M1168 = ON (in BIN mode) and SMOV instruction is in use, D10 and D20 will not be converted in BCD format but be moved in BIN format (4 digits as a unit)..



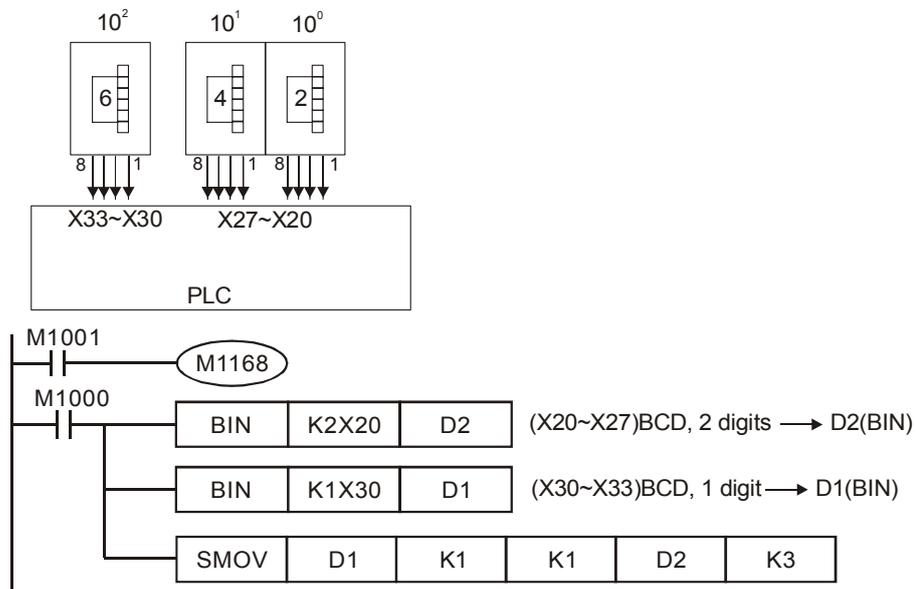
If D10 = H1234, D20 = H5678 before execution, D10 remains unchanged and D20 = H5128 after execution

Program example 3:

1. This instruction can be used to combine the DIP switches connected to the input terminals without continuous numbers.
2. Move the 2 digits of the right DIP switch (X27~X20) to the 2 digits of D2, and the 1 digit of the DIP switch (X33~X30) to the 1st digit of D1.

5 Categories and Use of Basic Application Instructions

Use SMOV instruction to move the 1st digit of D1 to the 3rd digit of D2 and combine the values from two DIP switches into one set of value.



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function								Controllers			
	D	CML	P	S	D	Compliment								20PM	10PM		
14																	
Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CML, CMLP: 5 steps DCML, DCMLP: 9 steps	
S					*	*	*	*	*	*	*	*	*	*	*		
D							*	*	*	*	*	*	*	*	*		

Operands:

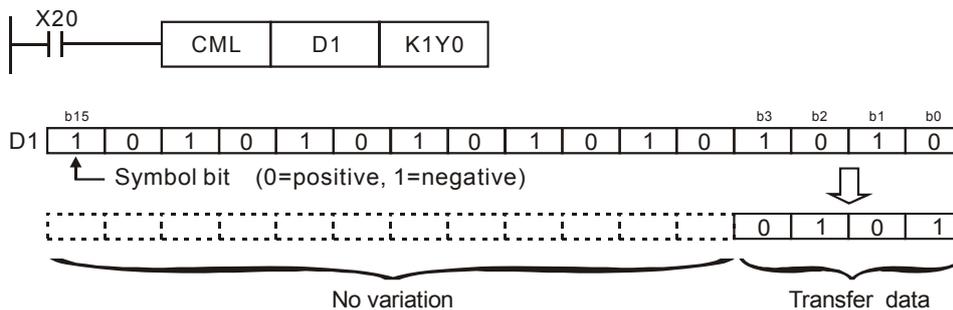
S: Source of data D: Destination device

Explanations:

- The instruction reverses the bit pattern (0→1, 1→0) of all the contents in **S** and sends the contents to **D**.
- When CML is used as 16-bit instruction, Z device cannot be adopted; when CML is used as 32-bit instruction, V device cannot be adopted.
- When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).

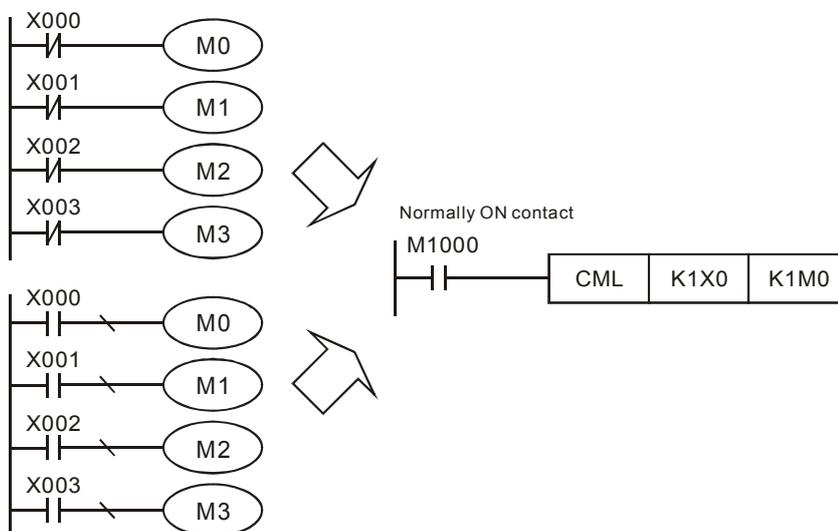
Program example 1:

When X10 = ON, b0 ~ b3 in D1 will be inverted and sent to Y0 ~ Y3



Program example 2:

The diagram below can be substituted by the instruction on the right.



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands			Function							Controllers				
	15	BMOV	P	(S)	(D)	(n)	Block Move							20PM	10PM		
Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	BMOV, BMOVP: 7 steps	
S							*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n					*	*					*	*	*				

Operands:

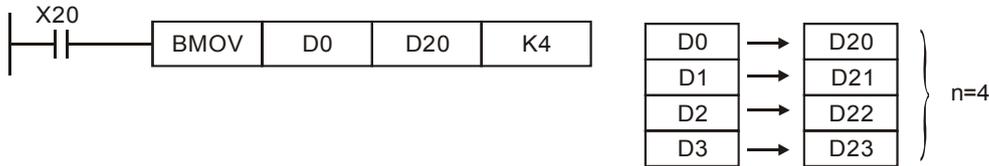
S: Start of source devices **D:** Start of destination devices **n:** Number of data to be moved

Explanations:

- The program copies a specified block of devices to another destination. Contents in **n** registers starting from **S** will be moved to **n** registers starting from **D**. If **n** exceeds the actual number of available source devices, only the devices that fall within the valid range will be used
- When BMOV is used as 16-bit instruction, Z device cannot be adopted; when BMOV is used as 32-bit instruction, V device cannot be adopted.
- When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
- Range of **n**: 1 ~ 512.

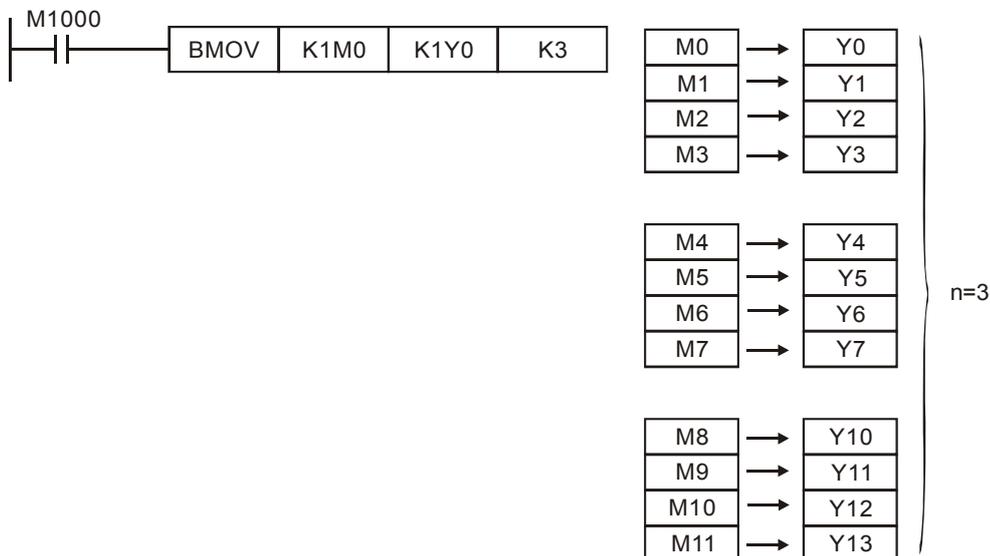
Program example 1:

When X20 = ON, the contents in registers D0 ~ D3 will be moved to the 4 registers D20 ~ D23



Program example 2:

Assume the bit devices KnX, KnY, KnM and KnS are designated for moving, the number of digits of **S** and **D** has to be the same, i.e. their **n** has to be the same.



5 Categories and Use of Basic Application Instructions

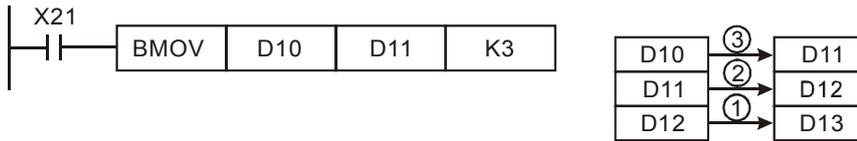
Program example 3:

The BMOV instruction will operate differently, automatically, to prevent errors when **S** and **D** coincide.

5. When **S** > **D**, the BMOV instruction is processed in the order ①→②→③.



6. When **S** < **D**, the BMOV instruction is processed in the order ③→②→①, then D11~D13 all equal to D10.



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands			Function								Controllers			
	D	FMOV	P	S	D	n	Fill Move								20PM	10PM	
16																	
Type	Bit Devices				Word devices								Program Steps				
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	FMOV, FMOV P: 7 steps	DFMOV, DFMOV P: 13 steps
S					*	*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*				
n					*	*											

Operands:

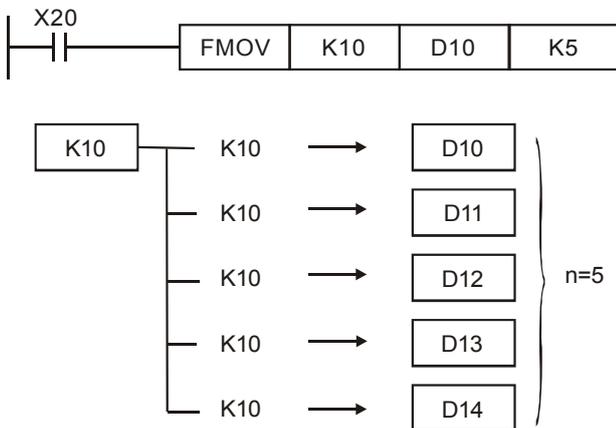
S: Source of data **D:** Destination of data **n:** Number of data to be moved

Explanations:

- The contents in n registers starting from the device designated by **S** will be moved to n registers starting from the device designated by **D**. If n exceeds the actual number of available source devices, only the devices that fall within the valid range will be used
- When FMOV is used as 16-bit instruction, Z device cannot be adopted; when FMOV is used as 32-bit instruction, V device cannot be adopted.
- When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
- The range of n: 1~ 512

Program example:

When X20 = ON, K10 will be moved to the 5 consecutive registers starting from D10



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands	Function										Controllers		
17	D	XCH	P	(D ₁) (D ₂)	Exchange										20PM	10PM	
Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	XCH, XCHP: 5 steps DXCH, DXCHP: 9 steps	
D ₁							*	*	*	*	*	*	*	*	*		
D ₂							*	*	*	*	*	*	*	*	*		

Operands:

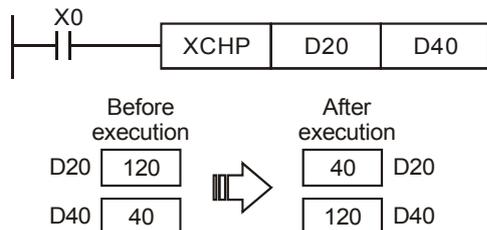
D₁: Device to be exchanged 1 D₂: Device to be exchanged 2

Explanations:

1. The contents in the devices designated by D₁ and D₂ will exchange
2. It is better to apply a pulse execution for this instruction (XCHP).
3. When XCH is used as 16-bit instruction, Z device cannot be adopted; when XCH is used as 32-bit instruction, V device cannot be adopted.
4. When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).

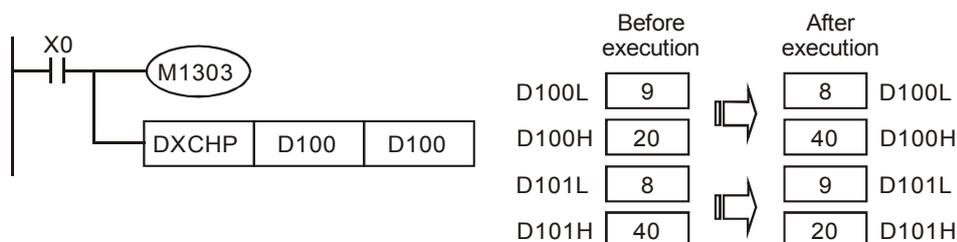
Program example:

When X0=OFF→ON, the contents of D20 and D40 exchange with each other.



Points to note:

1. As a 16-bit instruction, when the devices designated by D₁ and D₂ are the same and M1303 = ON, the upper and lower 8 bits of the designated devices exchange with each other.
2. As a 32-bit instruction, when the devices designated by D₁ and D₂ are the same and M1303 = ON, the upper and lower 16 bits in the designated device exchange with each other.
3. When X0 = ON and M1303 = ON, 16-bit contents in D100 and those in D101 will exchange with each other.



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers		
	D	BCD	P	S	D	Convert BIN to BCD										20PM	10PM	
18	D	BCD	P	S	D	Convert BIN to BCD										20PM	10PM	
OP	Type	Bit Devices				Word Devices										Program Steps		
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	BCD, BCDP: 5 steps	DBC, DBCDP: 6 steps
S								*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*	*	*		

Operands:

S: Source of data **D:** Conversion result

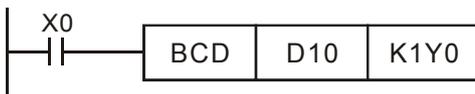
Explanations:

- The content in **S** (BIN value) is converted into BCD value and stored in **D**
- When BCD is used as 16-bit instruction, Z device cannot be adopted; when BCD is used as 32-bit instruction, V device cannot be adopted.
- If the conversion result exceeds the range of 0 ~ 9,999, BCD will not be executed. If the conversion result exceeds the range of 0 ~ 99,999,999, **DBC** will not be executed.
- BCD instruction converts the BIN data in the positioning unit into BCD data for displaying data on external device (e.g., 7-segment display).
- Flags:

	OX	O100
Operation error	M1793	M1953

Program Example:

- When X0 = ON, the binary value of D10 will be converted into BCD value, and the 1s digit of the conversion result will be stored in K1Y0 (Y0 ~ Y3, the 4 bit devices).



- If D10=001E (Hex) = 0030 (decimal), the result will be Y0~Y3 = 0000(BIN).

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
	19	D	BIN	P	S	D	Convert BCD to BIN										20PM

OP	Type	Bit Devices				Word Devices								Program Steps						
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z				
S								*	*	*	*	*	*	*	*	*	BIN, BINP: 5 steps			
D								*	*	*	*	*	*	*	*	*	DBIN, DBINP: 6 steps			

Operands:

S: Source of data **D:** Result of conversion

Explanations:

- The content in **S** (BCD value) is converted into BIN value and stored in **D**
- When BIN is used as 16-bit instruction, Z device cannot be adopted; when BIN is used as 32-bit instruction, V device cannot be adopted.
- The valid range of source **S**: BCD (0 to 9,999), **D** BCD (0 to 99,999,999)
- Constant K and H will automatically be converted into BIN format, therefore BIN instruction do not need to be adopted for constants.
- Flags:

	OX	O100
Operation error	M1793	M1953

Program Example:

When X0 = ON, the BCD value in K1M0 will be converted into BIN value and stored in D10.



Points to note::

- When DVP-PM needs to read an external DIP switch in BCD format, BIN instruction has to be first adopted to convert the read data into BIN value and store the data in PLC.
- On the contrary when DVP-PM needs to display a value on a BCD format 7-segment displayer, BCD instruction is required to convert the internal data into BCD value then sent the value to the displayer.
- When X0 = ON, the BCD value of K4X0 is converted into BIN value and sent to D100. The BIN value of D100 will then be converted into BCD value and sent to K4Y20.



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands			Function										Controllers		
	D	ADD	P	S ₁	S ₂	D	Addition										20PM	10PM
20	Type		Bit Devices				Word Devices										Program Steps	
	OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	ADD, ADDP: 7 steps	
		S ₁				*	*	*	*	*	*	*	*	*	*	*	*	DADD, DADDP: 9 steps
		S ₂				*	*	*	*	*	*	*	*	*	*	*	*	
D							*	*	*	*	*	*	*	*	*			

Operands:

S₁: Summand S₂: Addend D: Sum

Explanations:

- This instruction adds S₁ and S₂ in BIN format and store the result in D
- When ADD is used as 16-bit instruction, Z device cannot be adopted; when ADD is used as 32-bit instruction, V device cannot be adopted.
- The most significant bit (MSB) is the sign bit of the data. 0 indicates positive and 1 indicates negative. All calculations is algebraically processed, e.g. 3 + (-9) = -6.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

Flag changes in binary addition

In 16-bit BIN addition,

- If the operational result = 0, zero flag will be ON.
- If the operational result < -32,768, borrow flag will be ON.
- If the operational result > 32,767, carry flag will be ON.

In 32-bit BIN addition,

- If the operational result = 0, zero flag will be ON.
- If the operational result < -2,147,483,648, borrow flag will be ON.
- If the operational result > 2,147,483,647, carry flag will be ON.

Program Example 1:

In 16-bit BIN addition:

When X0 = ON, the content in D0 will plus the content in D10, and the sum will be stored in D20.



Program Example 2:

In 32-bit BIN addition:

When X1 = ON, the content in (D31, D30) will plus the content in (D41, D40), and the sum will be stored in (D51, D50). D30, D40 and D50 are low words; D31, D41 and D51 are high words. .

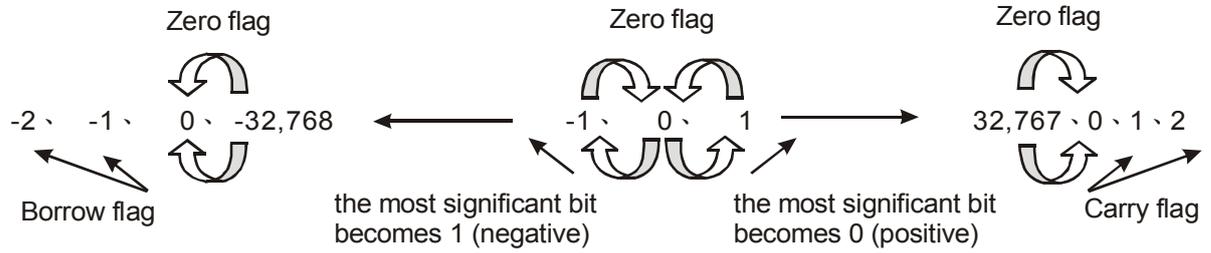


5 Categories and Use of Basic Application Instructions

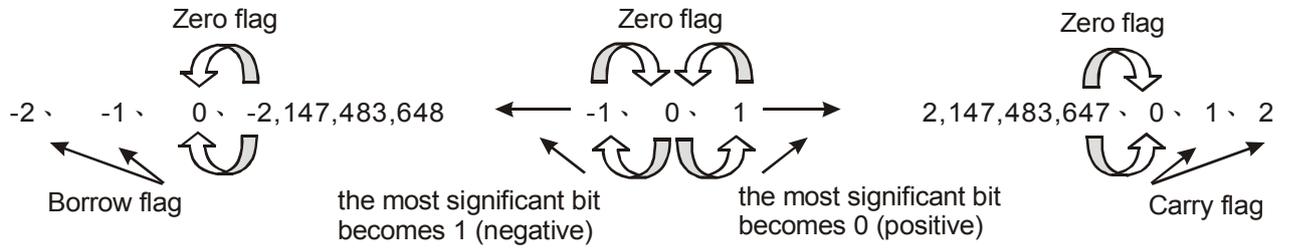
Points to note:

Operations of flags:

16-bit instruction:



32-bit instruction:



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function			Controllers	
	21	D	SUB	P	S ₁	S ₂	D	Subtraction			20PM

OP	Type	Bit Devices				Word Devices										Program Steps	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z		
S ₁					*	*	*	*	*	*	*	*	*	*	*	SUB, SUBP: 7 steps	
S ₂					*	*	*	*	*	*	*	*	*	*	*	DSUB, DSUBP: 9 steps	
D							*	*	*	*	*	*	*	*	*		

Operands:

S₁: Minuend S₂: Subtrahend D: Remainder

Explanations:

- SUB instruction subtracts S₁ and S₂ in BIN format and stores the result in D
- When SUB is used as 16-bit instruction, Z device cannot be adopted; when SUB is used as 32-bit instruction, V device cannot be adopted.
- The MSB is the sign bit. 0 indicates positive and 1 indicates negative. All calculation is algebraically processed
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

Flag changes in binary subtraction

In 16-bit DIN subtraction,

- If the operational result = 0, zero flag will be ON.
- If the operational result < -32,768, borrow flag will be ON.
- If the operational result > 32,767, carry flag will be ON.

In 32-bit BIN subtraction,

- If the operational result = 0, zero flag will be ON.
- If the operational result < -2,147,483,648, borrow flag will be ON.
- If the operational result > 2,147,483,647, carry flag will be ON.

- The operation of flags for ADD instruction can also be applied to SUB instruction.

Program Example 1:

In 16-bit BIN subtraction:

When X0 = ON, the content in D0 will minus the content in D10, and the remainder will be stored in D20.



Program Example 2:

In 32-bit BIN subtraction:

When X1 = ON, the content in (D31, D30) will minus the content in (D41, D40), and the remainder will be stored in (D51, D50). D30, D40 and D50 are low word; D31, D41 and D51 are high word.



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function			Controllers	
	22	D	MUL	P	(S ₁)	(S ₂)	(D)	Multiplication			20PM

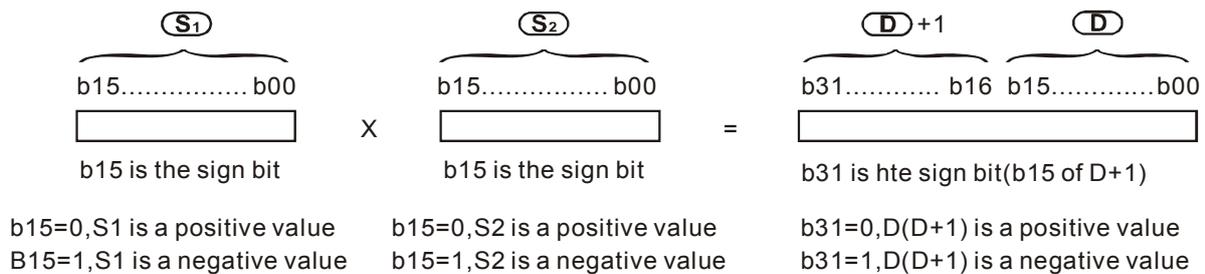
OP	Type	Bit Devices				Word Devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z			
S ₁					*	*	*	*	*	*	*	*	*	*	*			MUL, MULP: 7 steps
S ₂					*	*	*	*	*	*	*	*	*	*	*			DMUL, DMULP: 9 steps
D											*	*	*					

Operands:

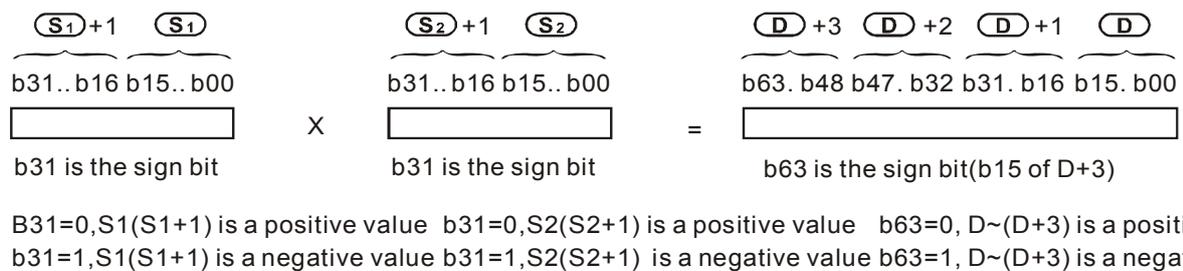
S₁: Multiplicand S₂: Multiplier D: Product

Explanations:

- MUL instruction multiplies S₁ by S₂ in BIN format and stores the result in D. Care should be taken on positive/negative signs of S₁, S₂ and D when doing 16-bit and 32-bit operations.
- When MUL is used as 16-bit instruction, Z device cannot be adopted; when MUL is used as 32-bit instruction, V device cannot be adopted.
- In 16-bit BIN multiplication:

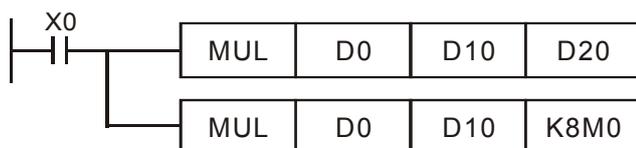


- In 32-bit BIN multiplication:



Program Example:

The 16-bit D0 is multiplied by the 16-bit D10 and brings forth a 32-bit product. The higher 16 bits are stored in D21 and the lower 16-bit are stored in D20. ON/OFF of MSB indicates the positive/negative status of the operation result.



5 Categories and Use of Basic Application Instructions

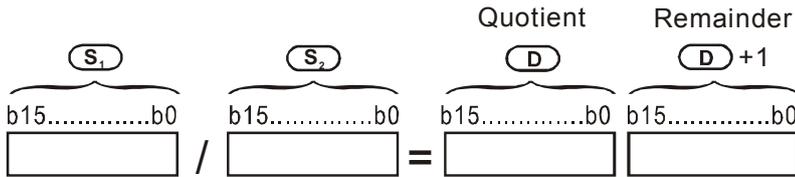
API	Mnemonic			Operands			Function							Controllers				
	D	DIV	P	S ₁	S ₂	D	Division							20PM	10PM			
23																		
OP	Type	Bit Devices				Word Devices										Program Steps		
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DIV, DIVP: 7 steps DDIV, DDIVP: 9 steps	
	S ₁					*	*	*	*	*	*	*	*	*	*	*		
	S ₂					*	*	*	*	*	*	*	*	*	*	*		
	D											*	*	*				

Operands:

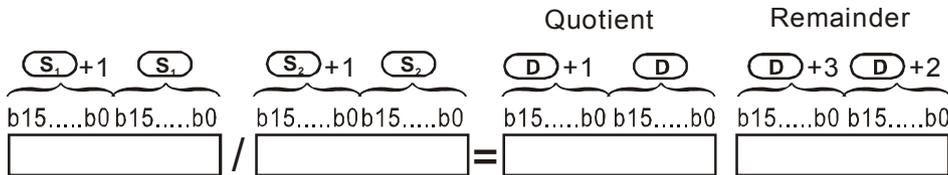
S₁: Dividend S₂: Divisor D: Quotient and remainder

Explanations:

- DIV instruction divides S₁ and S₂ in BIN format and stores the result in D. Care should be taken on positive/negative signs of S₁, S₂ and D when doing 16-bit and 32-bit operations.
- When DIV is used as 16-bit instruction, Z device cannot be adopted; when DIV is used as 32-bit instruction, V device cannot be adopted.
- DIV will not be executed when the divisor is 0.
- In 16-bit BIN division:

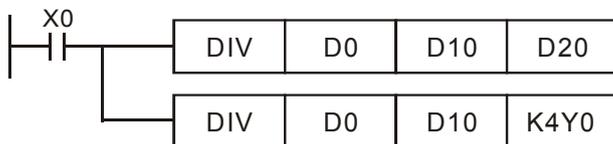


- In 32-bit BIN division:



Program Example:

When X0 = ON, D0 will be divided by D10, and the quotient will be stored in D20 while remainder in D21. ON/OFF of MSB indicates the positive/negative status of the result.



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands	Function	Controllers	
24	D	INC	P	(D)	Increment	20PM	10PM

OP	Type	Bit Devices				Word Devices										Program Steps				
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z				
D									*	*	*	*	*	*	*	*	INC, INCP: 3 steps DINC, DINCP: 3 steps			

Operands:

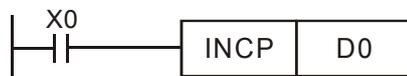
D: Destination device

Explanations:

1. If the instruction is not used in pulse execution mode, the content in the designated device **D** will plus "1" in every scan period
2. When INC is executed, the content in **D** will be incremented. However, in 16-bit instruction, if +32,767 is reached and "1" is added, it will write a value of -32,768 to the destination. In 32-bit instruction, if +2,147,483,647 is reached and "1" is added, it will write a value of -2,147,483,648 to the destination.
3. When INC is used as 16-bit instruction, Z device cannot be adopted; when INC is used as 32-bit instruction, V device cannot be adopted.
4. This instruction is generally used in pulse execution mode (INC P, DINC P).

Program Example:

When X0 is triggered, the content of D0 will be incremented by 1.



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands	Function	Controllers	
25	D	DEC	P	(D)	Decrement	20PM	10PM

Type OP	Bit Devices				Word Devices								Program Steps			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
D								*	*	*	*	*	*	*	*	DEC, DECP: 3 steps DDEC, DDECP: 3 steps

Operands:

D: Destination device

Explanations:

1. If the instruction is not used in pulse execution mode, the content in the designated device D will minus “1” in every scan period.
2. When DEC is executed, the content in D will be decremented. However, in 16-bit instruction, if -32,768 is reached and “1” is minused, it will write a value of +32,767 to the destination. In 32-bit instruction, if -2,147,483,648 is reached and “1” is minused, it will write a value of +2,147,483,647 to the destination
3. When DEC is used as 16-bit instruction, Z device cannot be adopted; when DEC is used as 32-bit instruction, V device cannot be adopted.
4. This instruction is generally used in pulse execution mode (DEC P, DDEC P)

Program Example:

When X0 is triggered, the value in D0 will be decremented by 1.



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function			Controllers							
26	D	WAND	P	(S ₁)	(S ₂)	(D)	Logical Word AND			20PM	10PM						
OP	Type	Bit Devices				Word Devices								Program Steps			
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	WAND, WANDP: 7 steps DWAND, DWANDP: 9 steps
S ₁					*	*	*	*	*	*	*	*	*	*	*	*	
S ₂					*	*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	*	

Operands:

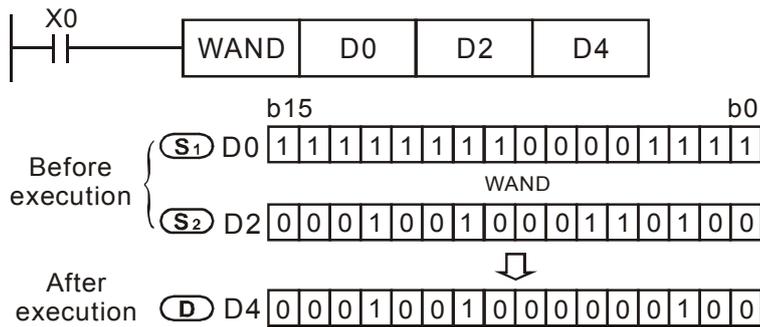
S₁: Source data device 1 S₂: Source data device 2 D: Operation result

Explanations:

1. WAND instruction conducts logical AND operation of S₁ and S₂ and stores the result in D
2. When WAND is used as 16-bit instruction, Z device cannot be adopted; when WAND is used as 32-bit instruction, V device cannot be adopted.

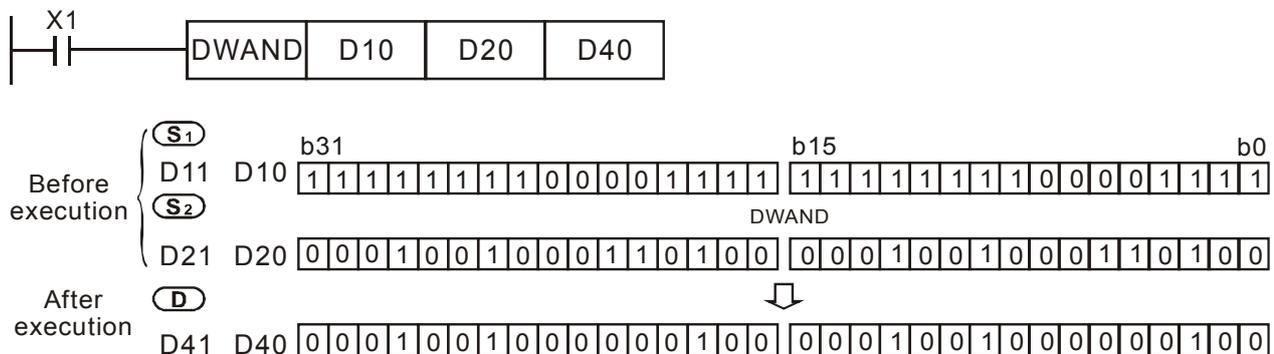
Program Example 1:

When X0 = ON, the 16-bit source D0 and D2 are analyzed and the operation result of the logical AND operation is stored in D4.



Program Example 2:

When X1 = ON, the 32-bit source (D11, D10) and (D21, D20) are analyzed and the operation result of the logical AND operation is stored in (D41, D40).



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function			Controllers							
	D	WOR	P	(S ₁)	(S ₂)	(D)	Logical Word OR			20PM	10PM						
27																	
Type OP	Bit Devices				Word Devices								Program Steps				
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	WOR, WOP: 7 steps DWOR, DWOP: 9 steps	
S ₁					*	*	*	*	*	*	*	*	*	*	*		
S ₂					*	*	*	*	*	*	*	*	*	*	*		
D							*	*	*	*	*	*	*	*	*		

Operands:

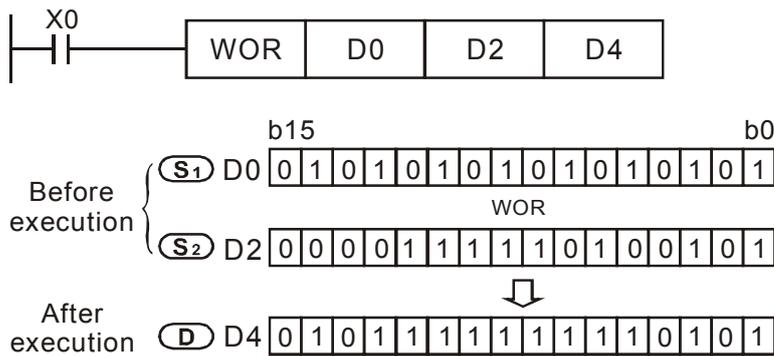
S₁: Source data device 1 S₂: Source data device 2 D: Operation result

Explanations:

1. WOR instruction conducts logical OR operation of S₁ and S₂ and stores the result in D
2. When WOR is used as 16-bit instruction, Z device cannot be adopted; when WOR is used as 32-bit instruction, V device cannot be adopted.

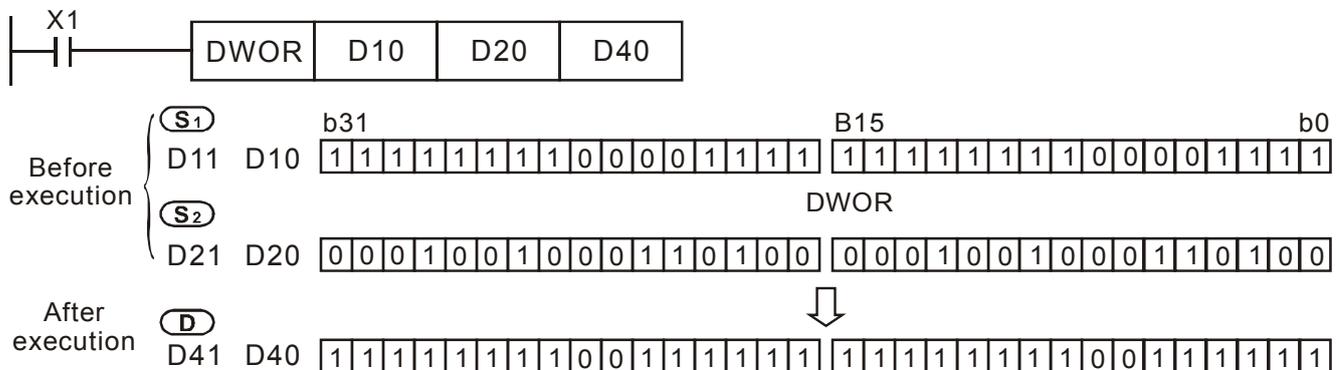
Program Example 1:

When X0 = ON, the 16-bit data source D0 and D2 are analyzed and the result of the logical OR is stored in D4.



Program Example 2:

When X1 is ON, the 32-bit data source (D11, D10) and (D21, D20) are analyzed and the operation result of the logical OR is stored in (D41, D40).



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function			Controllers							
28	D	WXOR	P	(S ₁)	(S ₂)	(D)	Logical Word XOR			20PM	10PM						
OP	Type	Bit Devices				Word Devices								Program Steps			
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	WXOR, WXORP: 7 steps DWXOR, DWXORP: 9 steps
	S ₁					*	*	*	*	*	*	*	*	*	*	*	
	S ₂					*	*	*	*	*	*	*	*	*	*	*	
	D							*	*	*	*	*	*	*	*	*	

Operands:

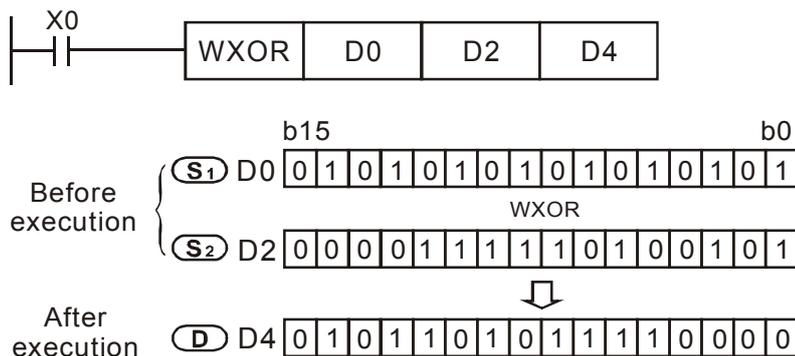
S₁: Source data device 1 S₂: Source data device 2 D: Operation result

Explanations:

1. WXOR conducts logical XOR operation of S₁ and S₂ in 16-bit mode and stores the result in D
2. When WXOR is used as 16-bit instruction, Z device cannot be adopted; when WXOR is used as 32-bit instruction, V device cannot be adopted.

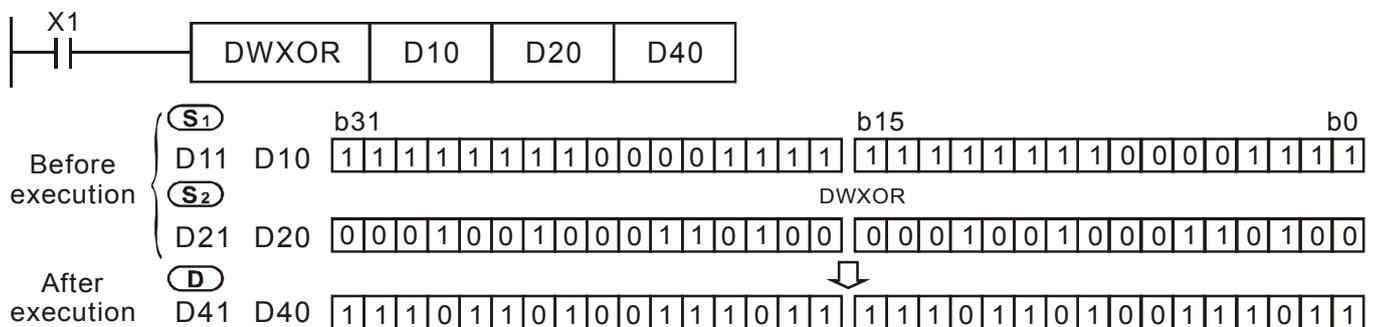
Program Example 1:

When X0 = ON, the 16-bit data source D0 and D2 are analyzed and the operation result of the logical XOR is stored in D4.



Program Example 2:

When X1 = ON, the 32-bit (D11, D10) and (D21, D20) will perform DWXOR logical XOR operation, and the result will be stored in (D41, D40).



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands	Function	Controllers										
	D	NEG	P			20PM	10PM									
29	D	NEG	P	D	2's Complement (Negation)											
OP	Type	Bit Devices				Word Devices								Program Steps		
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
	D							*	*	*	*	*	*	*	*	

Operands:

D: Device to store the operation result of 2's Compliment

Explanations:

1. NEG instruction conducts operation of 2's complement and can be used for converting a negative BIN value into an absolute value
2. When NEG is used as 16-bit instruction, Z device cannot be adopted; when NEG is used as 32-bit instruction, V device cannot be adopted.
3. This instruction is generally used in pulse execution mode (NEGP, DNEGP).

Program Example 1:

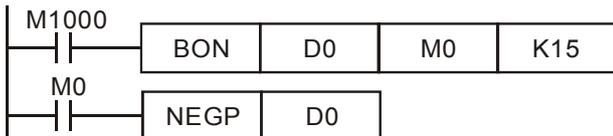
When X0 goes from OFF to ON, the phase of each bit in D10 will be reversed (0→1, 1→0) and then 1 will be added to the Least Significant Bit (LSB) of the register. Operation result will then be stored in D10.



Program Example 2:

To obtain the absolute value of a negative value:

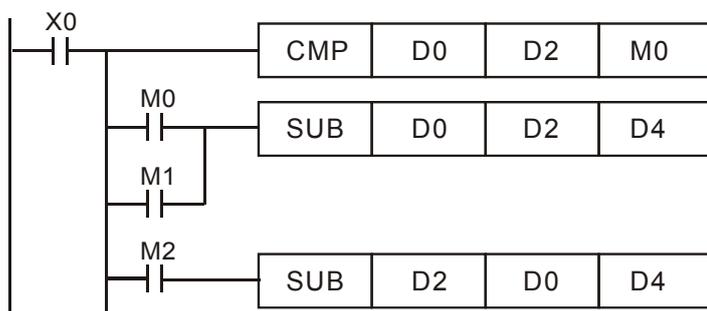
1. When MSB (b15) of D0 is "1", M0 = ON. (D0 is a negative value)
2. When M0 = ON, the absolute value of D0 can be obtained by NEG instruction.



Program Example 3:

Obtain the absolute value of the remainder of the subtraction. When X0 = ON,

1. If D0 > D2, M0 = ON.
2. If D0 = D2, M1 = ON.
3. If D0 < D2, M2 = ON.
4. D4 is then able to remain positive.



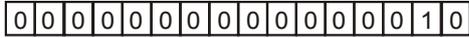
5 Categories and Use of Basic Application Instructions

Points to note:

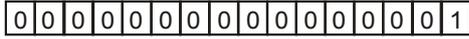
Detailed explanations on negative value and its absolute value:

1. MSB = 0 indicates the value is positive while MSB = 1 indicates the value is negative.
2. NEG instruction can be applied to convert a negative value into its absolute value

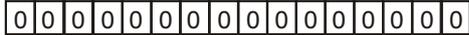
(D0=2)



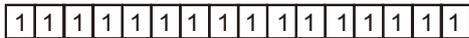
(D0=1)



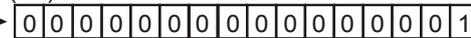
(D0=0)



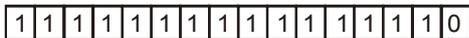
(D0=-1)



$\overline{(D0)+1=1}$



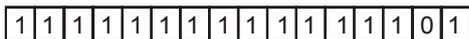
(D0=-2)



$\overline{(D0)+1=2}$



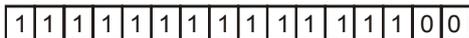
(D0=-3)



$\overline{(D0)+1=3}$



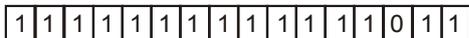
(D0=-4)



$\overline{(D0)+1=4}$



(D0=-5)



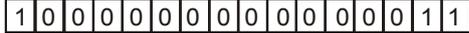
$\overline{(D0)+1=5}$



⋮

⋮

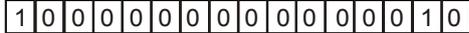
(D0=-32,765)



$\overline{(D0)+1=32,765}$



(D0=-32,766)



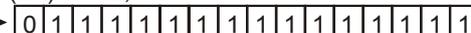
$\overline{(D0)+1=32,766}$



(D0=-32,767)



$\overline{(D0)+1=32,767}$



(D0=-32,768)



$\overline{(D0)+1=-32,768}$



↖
Max. absolute value is 32,767

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
30	D	ROR	P	D	n	Rotation Right										20PM	10PM
OP	Type	Bit Devices				Word devices										Program Steps	
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ROR, RORP: 5 steps DROR, DRORP: 9 steps
	D							*	*	*	*	*	*	*	*		
	n					*	*										

Operands:

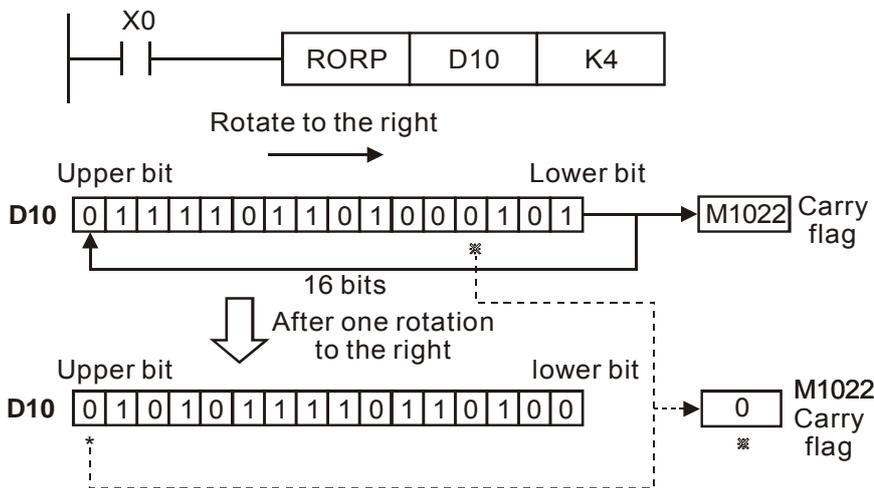
D: Device to be rotated **n:** Number of bits to be rotated in 1 rotation

Explanations:

1. This instruction rotates bit status of the device **D** to the right for **n** bits
2. The status of the last bit rotated (marked with ※) is copied to the carry flag M1970 (Carry flag)
3. This instruction is generally used in pulse execution mode (RORP, DRORP).
4. When ROR is used as 16-bit instruction, Z device cannot be adopted; when ROR is used as 32-bit instruction, V device cannot be adopted.
5. When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
6. If operand **D** is specified as KnY, KnM or KnS, only K4 (16-bit) or K8 (32-bit) is valid.
7. Valid range of operand **n**: $1 \leq n \leq 16$ (16-bit), $1 \leq n \leq 32$ (32-bit)

Program Example:

When X0 goes from OFF to ON, the 16 bits (4 bits as a group) in D10 will rotate to the right, as shown in the figure below. The bit marked with ※ will be sent to carry flag M1970..



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers		
32	D	R	C	P	(D)	(n)	Rotation Right with Carry										20PM	10PM
Type OP	Bit Devices				Word devices										Program Steps			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	R	C	
D								*	*	*	*	*	*	*	*	5	5	
n					*	*										9	9	

Operands:

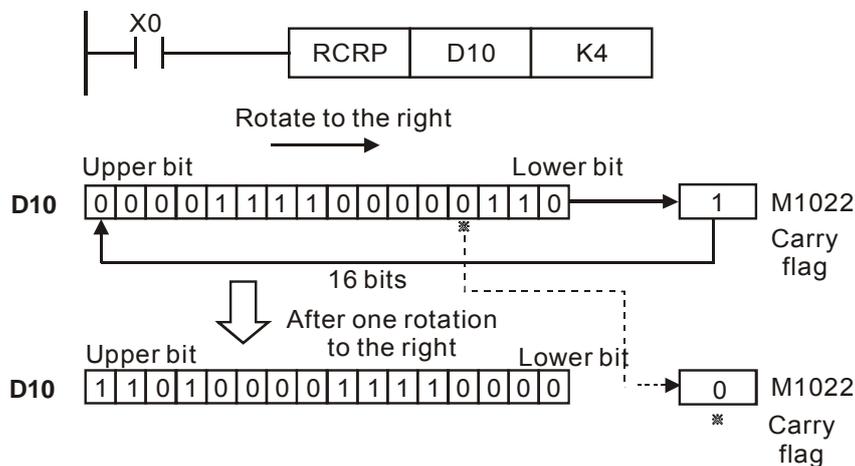
D: Device to be rotated n: Number of bits to be rotated in 1 rotation

Explanation:

1. This instruction rotates bit status of the device **D** together with M1970 to the right for **n** bits.
2. The status of the last bit rotated (marked with ※) is moved to the carry flag M1970.
3. This instruction is generally used in pulse execution mode (RCRP, DRCRP).
4. When RCR is used as 16-bit instruction, Z device cannot be adopted; when RCR is used as 32-bit instruction, V device cannot be adopted..
5. When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
6. Valid range of operand **n**: $1 \leq n \leq 16$ (16-bit), $1 \leq n \leq 32$ (32-bit)

Program Example:

When X0 goes from OFF to ON, the 16 bits (4 bits as a group) in D10 together with carry flag M1970 (total 17 bits) will rotate to the right, as shown in the figure below. The bit marked with ※ will be moved to carry flag M1970



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands				Function				Controllers					
	SFTR	P	S	D	n ₁	n ₂	Bit Shift Right				20PM	10PM				
34																
Type OP	Bit Devices				Word devices								Program Steps			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SFTR, SFTRP: 9 steps
S	*	*	*	*												
D		*	*	*												
n ₁					*	*										
n ₂					*	*										

Operands:

S: Start No. of source device **D:** Start No. of destination device **n₁:** Length of data to be shifted **n₂:** Number of bits to be shifted as a group

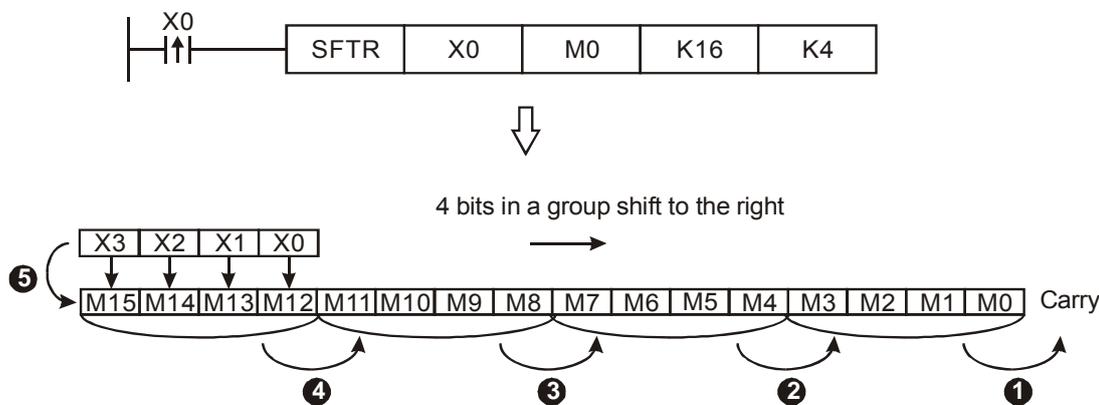
Explanation:

- This instruction performs a right shift from source device of **n₂** bits starting from **S** to destination device of **n₁** bits starting from **D**.
- This instruction is generally used in pulse execution mode (SFTRP).
- When SFTR is used as 16-bit instruction, Z device cannot be adopted; when SFTR is used as 32-bit instruction, V device cannot be adopted.
- Valid range of operand **n₁**, **n₂** : $1 \leq n_2 \leq n_1 \leq 1024$

Program Example:

- When X0 is rising edge triggered, SFTR instruction shifts X0~X4 into 16 bit data M0~M15 and M0~M15 also shift to the right with a group of 4 bits.
- The figure below illustrates the right shift of the bits in one scan.

- ① M3~M0 → Carry
- ② M7~M4 → M3~M0
- ③ M11~M8 → M7~M4
- ④ M15~M12 → M11~M8
- ⑤ X3~X0 → M15~M12 completed



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands				Function				Controllers	
35	SFTL	P	(S)	(D)	(n1)	(n2)	Bit Shift Left				20PM	10PM

Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F	
S	*	*	*	*													SFTL, SFTLP: 9 steps
D		*	*	*													
n ₁					*	*											
n ₂					*	*											

Operands:

S: Start No. of source device **D:** Start No. of destination device **n₁:** Length of data to be shifted **n₂:** Number of bits to be shifted as a group

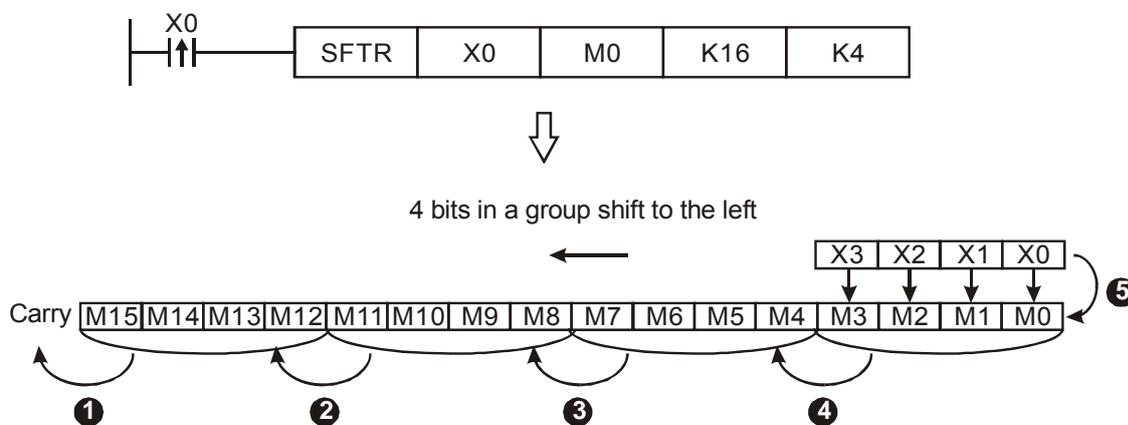
Explanations:

- This instruction performs a left shift from source device of **n₂** bits starting from **S** to destination device of **n₁** bits starting from **D**
- This instruction is generally used in pulse execution mode (SFTLP).
- When SFTL is used as 16-bit instruction, Z device cannot be adopted; when SFTL is used as 32-bit instruction, V device cannot be adopted
- Valid range of operand **n₁, n₂** : $1 \leq n_2 \leq n_1 \leq 1024$

Program Example:

- When X0 is rising edge triggered, SFTL instruction shifts X0~X4 into 16-bit data M0~M15 and M0~M15 also shift to the left with a group of 4 bits.
- The figure below illustrates the left shift of the bits in one scan

- ❶ M15~M12 → Carry
- ❷ M11~M8 → M15~M12
- ❸ M7~M4 → M11~M8
- ❹ M3~M0 → M7~M4
- ❺ X3~X0 → M3~M0 completed



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands				Function				Controllers						
36	WSFR	P	(S)	(D)	(n ₁)	(n ₂)	Word Shift Right				20PM	10PM					
Type OP	Bit Devices				Word devices								Program Steps				
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WSFR, WSFRP: 9 steps	
S							*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n ₁					*	*											
n ₂					*	*											

Operands:

S: Start No. of source device **D:** Start No. of destination device **n₁:** Length of data to be shifted **n₂:** Number of devices to be shifted as a group

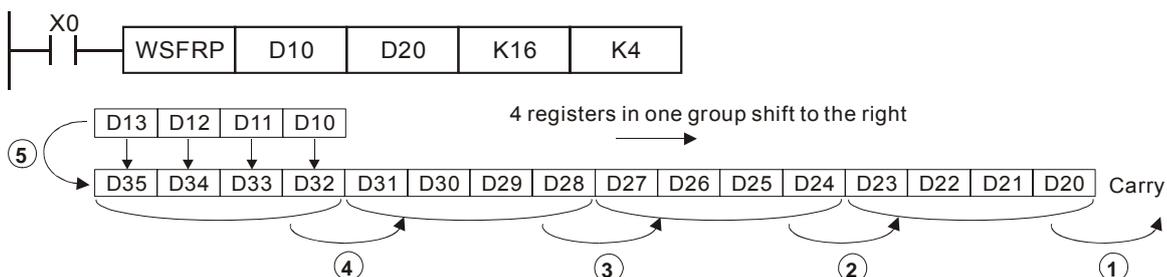
Explanations:

- This instruction performs a right shift from source device of **n₂** registers starting from **S** to destination device of **n₁** registers starting from **D**.
- This instruction is generally used in pulse execution mode (WSFRP).
- The type of devices designated by **S** and **D** has to be the same, e.g. **K_nX**, **K_nY**, **K_nM**, and **K_nS** as a category and **T**, **C**, and **D** as another category
- Provided the devices designated by **S** and **D** belong to **K_n** type, the number of digits of **K_n** in **S** and **D** has to be the same.
- When specifying the bit device of **K_nX(Y/M/S)**, the starting No. should be a multiple of 16 in octal or decimal format. For example, **K1X0**, **K4Y20** (octal); **K1M0**, **K4S16** (decimal).
- When WSFR is used as 16-bit instruction, **Z** device cannot be adopted; when WSFR is used as 32-bit instruction, **V** device cannot be adopted
- Valid range of operand **n₁**, **n₂** : $1 \leq n_2 \leq n_1 \leq 512$

Program Example 1:

- When **X0** is triggered, WSFRP instruction shifts **D10~D13** into data stack **D20~D35** and **D20~D35** also shift to the right with a group of 4 registers.
- The figure below illustrates the right shift of the registers in one scan.

- ① **D23~D20** → Carry
- ② **D27~D24** → **D23~D20**
- ③ **D31~D28** → **D27~D24**
- ④ **D35~D32** → **D31~D28**
- ⑤ **D13~D10** → **D35~D32** completed

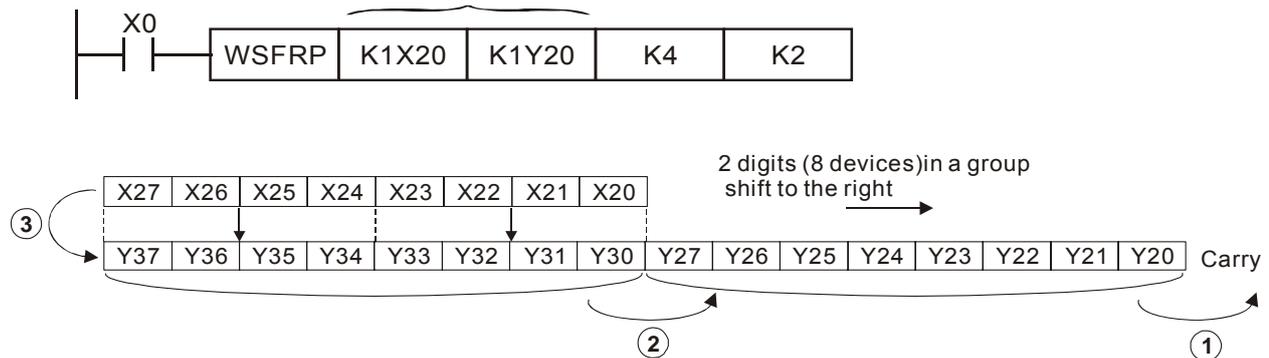


Program Example 2:

5 Categories and Use of Basic Application Instructions

1. When X0 is triggered, WSFRP instruction shifts X20~X27 into data stack Y20~Y37 and Y20~Y37 also shift to the right with a group of 4 devices.
2. The figure below illustrates the right shift of the devices in one scan
 - ❶ Y27~Y20 → carry
 - ❷ Y37~Y30 → Y27~Y20
 - ❸ X27~X20 → Y37~Y30 completed

When using Kn device, the specified Kn value (digit) must be the same.



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands				Function				Controllers						
	WSFL	P	S	D	n ₁	n ₂	Word Shift Left				20PM	10PM					
37																	
Type OP	Bit Devices				Word devices								Program Steps				
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WSFL, WSFLP: 9 steps	
S							*	*	*	*	*	*	*				
D								*	*	*	*	*	*				
n ₁					*	*											
n ₂					*	*											

Operands:

S: Start No. of source device **D:** Start No. of destination device **n₁:** Length of data to be shifted **n₂:** Number of devices to be shifted as a group

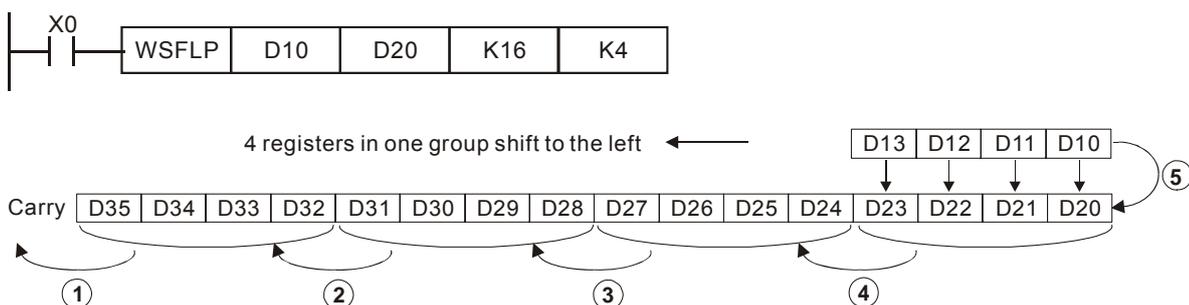
Explanations:

- This instruction performs a left shift from source device of **n₂** registers starting from **S** to destination device of **n₁** registers starting from **D**.
- This instruction is generally used in pulse execution mode (WSFLP).
- The type of devices designated by **S** and **D** has to be the same, e.g. K_nX, K_nY, K_nM, and K_nS as a category and T, C, and D as another category
- Provided the devices designated by **S** and **D** belong to K_n type, the number of digits of K_n in **S** and **D** has to be the same.
- When specifying the bit device of K_nX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
- When WSFL is used as 16-bit instruction, Z device cannot be adopted; when WSFL is used as 32-bit instruction, V device cannot be adopted.
- Valid range of operand **n₁**, **n₂** : 1 ≤ **n₂** ≤ **n₁** ≤ 512

Program Example:

- When X0 is triggered, WSFLP instruction shifts D10~D13 into data stack D20~D35 and D20~D35 also shift to the left with a group of 4 registers.
- The figure below illustrates the left shift of the words in one scan

- ① D35~D32 → Carry
- ② D31~D28 → D35~D32
- ③ D27~D24 → D31~D28
- ④ D23~D20 → D27~D24
- ⑤ D13~D10 → D23~D20 completed



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands			Function				Controllers	
38	SFWR	P	(S)	(D)	(n)	Shift Register Write				20PM	10PM

Type OP	Bit Devices				Word devices										Program Steps	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SFWR, SFWRP: 7 steps
S					*	*	*	*	*	*	*	*	*	*	*	
D							*	*	*	*	*	*				
n					*	*										

Operands:

S: Source device **D:** Head address of data stack **n:** Length of data stack

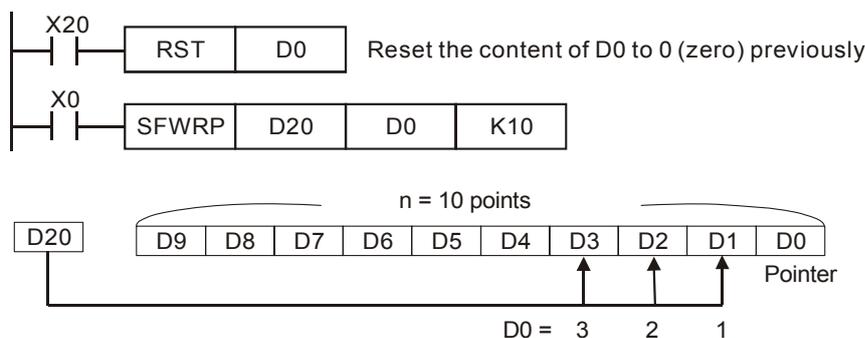
Explanations:

- This instruction defines the data stack of **n** words starting from **D** as a “first-in, first out (FIFO)” data stack and specifies the first device as the pointer (**D**). When SFWRP is executed, content in pointer pluses 1, and the content in **S** will be written into the device designated by the pointer. When the content in pointer exceeds **n-1**, the instruction stops and carry flag M1970 = ON.
- This instruction is generally used in pulse execution mode (SFWRP).
- When SFWR is used as 16-bit instruction, Z device cannot be adopted; when SFWR is used as 32-bit instruction, V device cannot be adopted.
- When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
- Valid range of operand **n**: $2 \leq n \leq 512$
- Flags:

	OX	O100
Carry flag	M1810	M1970

Program Example:

- First, reset the content of D0. When X0 goes from OFF to ON, the content of D0 (pointer) becomes 1, and D20 is written into D1. If the content of D20 is changed and X0 is triggered again, pointer D0 becomes 2, and the content of D20 is then written into D2.
- P The figure below illustrates the shift and writing process of the instruction.
 - The content of D0 becomes 1.
 - The content of D20 is written into D1.



Points to note:

This instruction can be used together with API 39 SFRD for the reading/writing of “first-in, first-out” stack data

5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands			Function							Controllers				
39	SFRD	P	S	D	n	Shift Register Read							20PM	10PM			
Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SFRD, SFRDP: 7 steps	
S					*	*		*	*	*	*	*	*				
D								*	*	*	*	*	*	*	*		
n					*	*											

Operands:

S: Head address of data stack **D:** Destination device **n:** Length of data stack

Explanation:

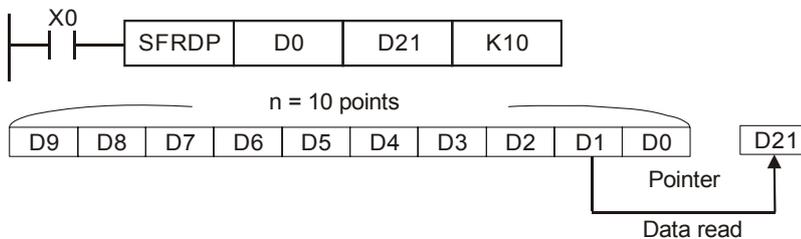
- This instruction defines the data stack of **n** words starting from **S** as a FIFO data stack and specifies the first device as the pointer (**S**). The content of pointer indicates current length of the stack. When SFRDP is executed, first data (**S+1**) will be read out to **D**, all data in this stack moves up to fill the read device and content in pointer minuses 1. When the content in pointer = 0, the instruction stops and carry flag M1970= ON
- This instruction is generally used in pulse execution mode (SFRDP).
- When SFRD is used as 16-bit instruction, Z device cannot be adopted; when SFRD is used as 32-bit instruction, V device cannot be adopted.
- When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
- Valid range of operand **n**: $2 \leq n \leq 512$
- Flags:

	OX	O100
Zero flag	M1808	M1968

Program Example:

- When X0 goes from OFF to ON, D9~D2 are all shifted to the right and the pointer D0 is decremented by 1 when the content of D1 is read and moved to D21.
- The figure below illustrates the shift and reading of the instruction.

- ❶ The content of D1 is read and moved to D21.
- ❷ D9~D2 are all shifted to the right.
- ❸ The content of D0 is decremented by 1.



5 Categories and Use of Basic Application Instructions

API	Mnemonic	Operands	Function	Controllers	
40	ZRST	P	(D ₁) (D ₂)	Zone Reset	
				20PM	10PM

OP	Type	Bit Devices				Word Devices										Program Steps				
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	ZRST, ZRSTP: 5 steps			
	D ₁		*	*	*							*	*	*						
	D ₂		*	*	*							*	*	*						

Operands:

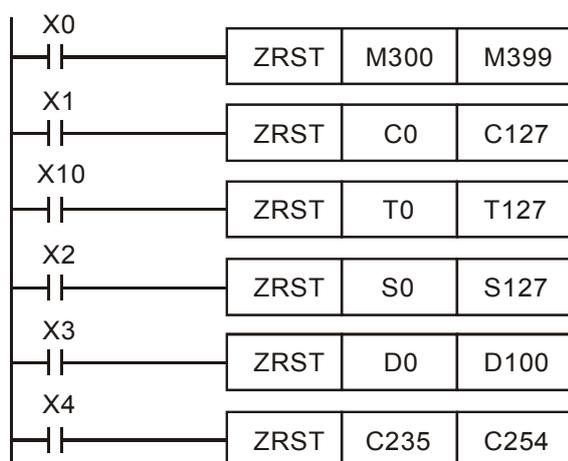
D₁: Starting device of the reset range **D₂**: End device of the reset range

Explanations:

1. When the instruction is executed, range **D₁** to **D₂** will be reset.
2. Operand **D₁** and **D₂** must be the same data type, Valid range: **D₁ ≤ D₂**
3. 16-bit counters and 32-bit counters can be reset at the same by executing ZRST instruction.
4. When **D₁ > D₂**, only operands designated by **D₂** will be reset.
5. This instruction is generally used in pulse execution mode (ZRSTP)

Program Example:

1. When X0 = ON, M300 to M399 will be reset..
2. When X1 = ON, C0 to C127 will all be reset, i.e. present value = 0 and the associated contacts / outputs will be reset as well.
3. When X20 = ON, T0 to T127 will all be reset, i.e. present value = 0 and the associated contacts/ outputs will be reset as well.
4. When X2 = ON, the steps of S0 to S127 will be reset.
5. When X3 = ON, the data of D0 to D100 will be reset.
6. When X4 = ON, C235 to C254 will all be reset, i.e. present value = 0 and the associated contacts/ outputs will be reset as well.



Points to note:

Bit devices Y, M, S and word devices T, C, D can be individually reset by RST instruction.

5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands			Function						Controllers						
41	DECO	P	(S)	(D)	(n)	Decode						20PM	10PM					
OP	Type	Bit Devices				Word devices								Program Steps				
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DECO, DECOP: 7 steps	
S		*	*	*	*	*	*					*	*	*	*	*		
D			*	*	*							*	*	*	*	*		
n						*	*											

Operands:

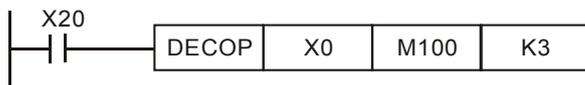
S: Source device to be decoded **D:** Device for storing the result **n:** Number of consecutive bits of **S**

Explanation:

1. The instruction decodes the lower “n” bits of **S** and stores the result of “2ⁿ” bits in **D**.
2. When DECO is used as 16-bit instruction, Z device cannot be adopted; when DECO is used as 32-bit instruction, V device cannot be adopted.
3. This instruction is generally used in pulse execution mode (DECOP).
4. When operand **D** is a bit device, n = 1~8, when operand **D** is a word device, n = 1~4

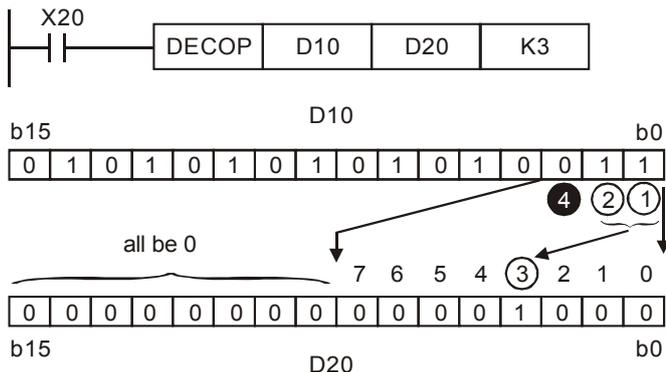
Program Example 1:

1. When **D** is used as a bit device, n = 1 ~ 8. Errors will occur if n = 0 or n > 8.
2. If n = 8, the decoded data is 2⁸= 256 bits data.
3. When X20 goes from OFF to ON, the data of X0~X2 will be decoded to M100~M107.
4. If the source data is 3, M103 (third bit from M100) = ON.
5. After the execution is completed, X20 is turned OFF. The decoded results or outputs will retain their operation.



Program Example 2:

1. When **D** is used as a word device, n = 1 ~ 4. Errors will occur if n = 0 or n > 4.
2. When n = 4, the decoded data is 2⁴ = 16 bits.
3. When X20 goes from OFF to ON, the data in D10 (b2 to b0) will be decoded and stored in D20 (b7 to b0). The unused bits in D20 (b15 to b8) will be set to 0.
4. The lower 3 bits of D10 are decoded and stored in the lower 8 bits of D20. The higher 8 bits of D20 are all 0.
5. After the execution is completed, X20 is turned OFF. The decoded results or outputs will retain their operation.



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands			Function						Controllers	
42	ENCO	P	(S)	(D)	(n)	Encode						20PM	10PM

Type OP	Bit Devices				Word devices										Program Steps	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*								*	*	*	*	*
D												*	*	*	*	*
n					*	*										

DECO, DECOP: 7 steps

Operands:

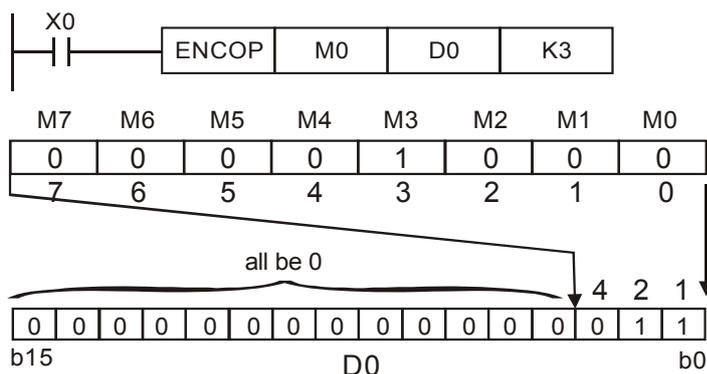
S: Source device to be encoded **D:** Device for storing the result **n:** Number of consecutive bits of **S**

Explanation:

1. The instruction encodes the lower "2ⁿ" bits of source **S** and stores the result in **D**.
2. The highest active bit in **S** has the priority for encoding operation.
3. When ENCO is used as 16-bit instruction, Z device cannot be adopted; when ENCO is used as 32-bit instruction, V device cannot be adopted.
4. This instruction is generally used in pulse execution mode (ENCOP).
5. When operand **S** is a bit device, n=1~8, when operand **S** is a word device, n=1~4
6. If no bits in **S** is active (1), M1067, M1068 = ON and D1067 records the error code 0E1A (hex).

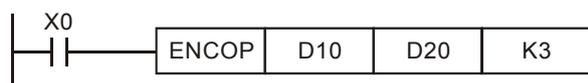
Program Example 1:

1. When **S** is used as a bit device, n = 1 ~ 8. Errors will occur if n = 0 or n > 8.
2. If n = 8, the decoded data is 2⁸ = 256 bits data.
3. When X0 goes from OFF to ON, the data in (M0 to M7) will be encoded and stored in lower 3 bits of D0 (b2 to b0). The unused bits in D0 (b15 to b3) will be set to 0.
4. After the execution is completed, X0 is turned OFF and the data in **D** remains unchanged.

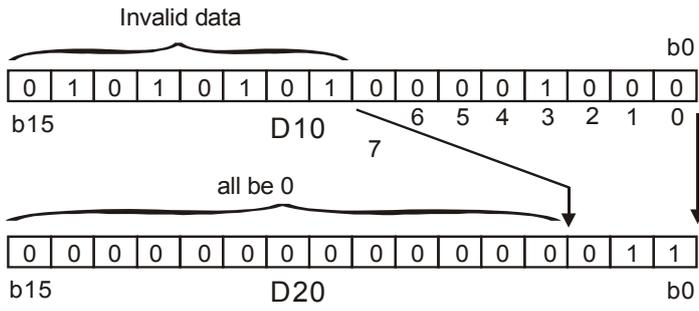


Program Example 2:

1. When **S** is used as a word device, n = 1 ~ 4. Errors will occur if n = 0 or n > 4.
2. When n = 4, the decoded data is 2⁴ = 16 bits data.
3. When X0 goes from OFF to ON, the 2³ bits (b0 ~ b7) in D10 will be encoded and the result will be stored in the lower 3 bits of D20 (b2 to b0). The unused bits in D20 (b15 to b3) will be set to 0.
4. After the execution is completed, X0 is turned OFF and the data in **D** remains unchanged



5 Categories and Use of Basic Application Instructions



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
43	D	SUM	P	S	D	Sum of Active bits										20PM	10PM

Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
S					*	*	*	*	*	*	*	*	*	*	*	SUM, DSUMP: 5 steps	
D											*	*	*	*	*	DSUM, DSUMP: 9 steps	

Operands:

S: Source device **D:** Destination device for storing counted value

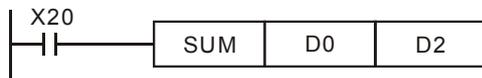
Explanation:

- This instruction counts the total active bits in **S** and store the value in **D**.
- D** will occupy two registers when using in 32-bit instruction.
- When SUM is used as 16-bit instruction, Z device cannot be adopted; when SUM is used as 32-bit instruction, V device cannot be adopted.
- When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
- If there is no active bits, zero flag M1968 =ON.
- Flags:

	OX	O100
Zero flag	M1808	M1968

Program Example:

When X20 = ON, all active bits in D0 will be counted and the result will be stored in D2.



0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	3
D0																D2	

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function							Controllers	
	45	D	MEAN	P	S	D	n	Mean							20PM

Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	MEAN, MEANP: 7 steps DMEAN, DMEANP: 13 steps	
S					*	*	*	*	*	*	*	*	*	*	*		
D							*	*	*	*	*	*	*	*	*		
n					*	*	*	*	*	*	*	*	*	*	*		

Operands:

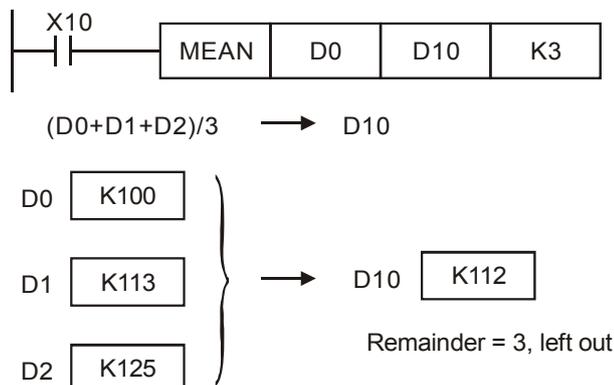
S: Source device **D:** Destination for storing result **n:** Number of consecutive device from **S**

Explanations:

- The instruction obtains the mean value from **n** consecutive registers from **S** and stores the value in **D**.
- Remainders in the operation will be ignored.
- If **S** is not within the valid range, only those addresses within the valid range will be processed.
- If **n** is out of the valid range (1~64), PLC will determine it as an “instruction operation error”.
- When MEAN is used as 16-bit instruction, Z device cannot be adopted; when MEAN is used as 32-bit instruction, V device cannot be adopted.
- When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal)
- Valid range of operand **n** : **n** = 1~64

Program Example:

When X10 = ON, the contents in 3 (**n** = 3) registers starting from D0 will be summed and then divided by 3 to obtain the mean value. The result will be stored in D10 and the remainder will be left out



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands			Function										Controllers	
46	ANS	P	S	m	D	Timed Annunciator Set										20PM	10PM
Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ANS: 7 steps	
S											*						
m					*												
D				*													

Operands:

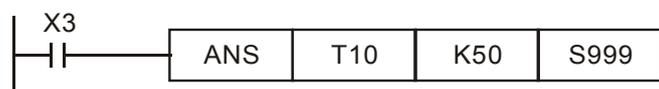
S: Alarm timer **m:** Time setting **D:** Alarm

Explanations:

1. ANS instruction is used to drive the output alarm device in designated time.
2. Operand **S** valid range: T0~T183
 Operand **m** valid range: K1~K32,767 (unit: 100 ms)
 Operand **D** valid range: S912~S1023
3. Flag: M1048 (ON: Alarm is active), M1049 (ON: Alarm monitoring is enabled)
4. See ANR instruction for more information

Program Example:

If X3 = ON for more than 5 sec, alarm step relay S999 will be ON. S999 will remains ON after X3 is reset. (T10 will be reset, present value = 0)



5 Categories and Use of Basic Application Instructions

API	Mnemonic	P	Function	Controllers	
47	ANR	P	Annunciator Reset	20PM	10PM

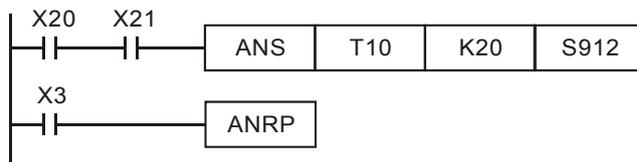
OP	Descriptions	Program Steps
N/A	No operand. No contact to drive the instruction is required	ANR, ANRP: 1 step

Explanations:

1. ANR instruction is used to reset an alarm.
2. When several alarm devices are ON, the alarm with smaller number will be reset.
3. This instruction is generally used in pulse execution mode (ANRP).

Program Example:

1. If X20 and X21 are ON at the same time for more than 2 sec, the alarm S912 will be ON. If X20 or X21 is reset, alarm S912 will remain ON but T10 will be reset and present value is cleared.
2. If X20 and X21 are ON less than 2 sec, the present value of T10 will be cleared.
3. When X3 goes from OFF → ON, activated alarms S912 will be reset.
4. When X3 goes from OFF → ON again, the alarm device with second lower number will be reset.



Points to note:

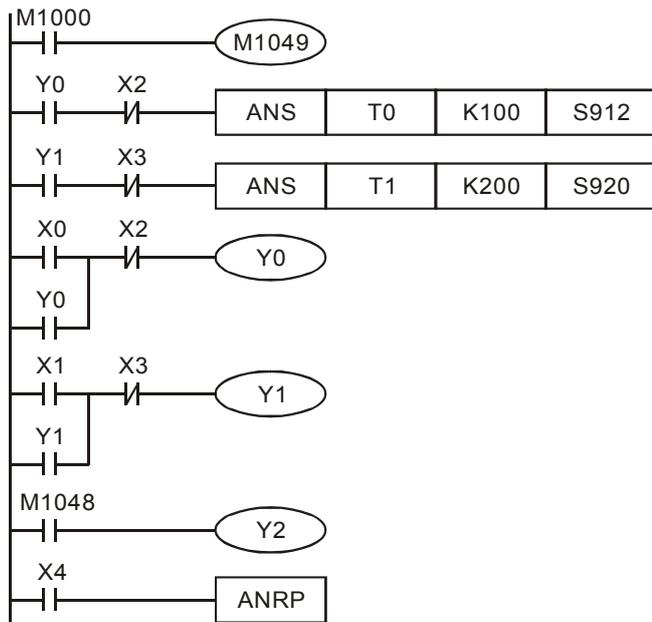
Flags:

1. M1048 (indicating alarm status): When M1049 = ON, enabling any of the alarm S912~S1023 turns M1048 ON.
2. M1049 (Enabling alarm monitoring): When M1049 = ON, D1049 will automatically hold the lowest alarm number in active alarms.

Application example of alarm device (production line):

- | | |
|----------------------------|---------------------------|
| X0 = Forward switch | X1 = Backward switch |
| X2 = Front position switch | X3 = Back position switch |
| X4 = Alarm reset button | |
| Y0 = Forward | Y1 = Backward |
| Y2 = Alarm indicator | |
| S912 = Forward alarm | S920 = Backward alarm |

5 Categories and Use of Basic Application Instructions



1. M1048 and D1049 are valid only when M1049 = ON.
2. When Y0 = ON for more than 10 sec and the product fails to reach the front position X2, S912 = ON
3. When Y1 = ON for more than 10 sec and the product fails to reach the back position X3, S920= ON.
4. When backward switch X1 = ON and backward device Y1 = ON, Y1 will go OFF only when the product reaches the back position switch X3.
5. Y2 is ON when any alarm is enabled.
6. Whenever X4 is ON, 1 active alarm will be reset. If there are several active alarms, the reset will start from the alarm with the lowest number and then the alarm with second lower number, etc.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
	48	D	SQR	P	S	D	Square Root										20PM

Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SQR, SQRP: 5 steps DSQR, DSQRP: 9 steps	
S											*						
m					*												
D				*													

Operands:

S: Source device **D:** Device for storing the result

Explanation:

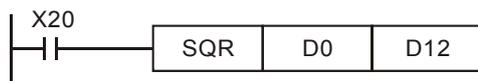
- This instruction performs a square root operation on **S** and stores the result in **D**.
- S** can only be a positive value. Performing a square root operation on a negative value will result in an error and the instruction will not be executed.
- The operation result **D** should be integer only, and the decimal will be left out. When decimal is left out, borrow flag M1969 = ON.
- When the operation result **D** = 0, zero flag M1968 = ON.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the absolute value of the conversion result > max. floating point value, carry flag will be ON.
- If the absolute value of the conversion result < min. floating point value, borrow flag will be ON.
- If the conversion result is "0", zero flag will be ON.

Program Example:

When X20 = ON, square root of D0 will be stored in D12.



$$\sqrt{D0} \rightarrow D12$$

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function								Controllers			
	D	FLT	P	S	D	Floating Point								20PM	10PM		
49																	
OP	Type	Bit Devices				Word Devices								Program Steps			
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DFLT, DFLTP: 6 steps
	S					*	*							*			
	D												*				

Operands:

S: Source device **D:** Device for storing the conversion result

Explanations:

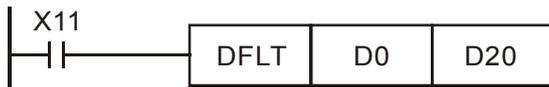
- Only 32-bit instructions **DFLT** and **DFLTP** are applicable
- FLT instruction converts BIN integer in **S** into binary floating point value in **D**
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the absolute value of the conversion result > max. floating point value, carry flag will be ON.
- If the absolute value of the conversion result < min. floating point value, borrow flag will be ON.
- If the conversion result is "0", zero flag will be ON.

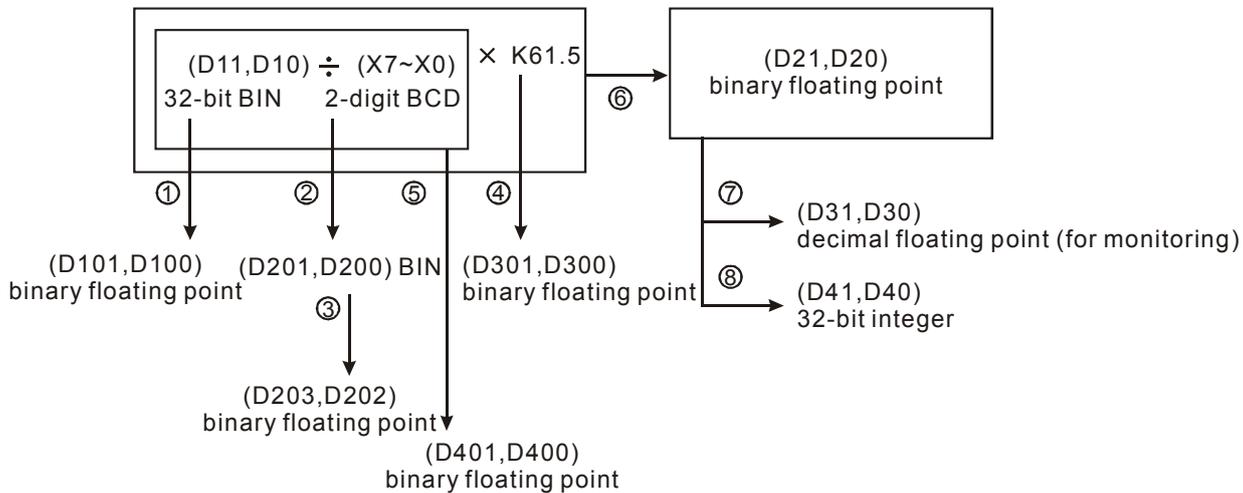
Program Example 1:

- When X11 = ON, D1 and D0 (BIN integers) are converted into D21 and D20 (binary floating point values).
- Assume 32-bit register D0 (D1) = K100,000. When X11 = ON, the converted 32-bit floating point value will be H'4735000 and stored in the 32-bit register D20 (D21).

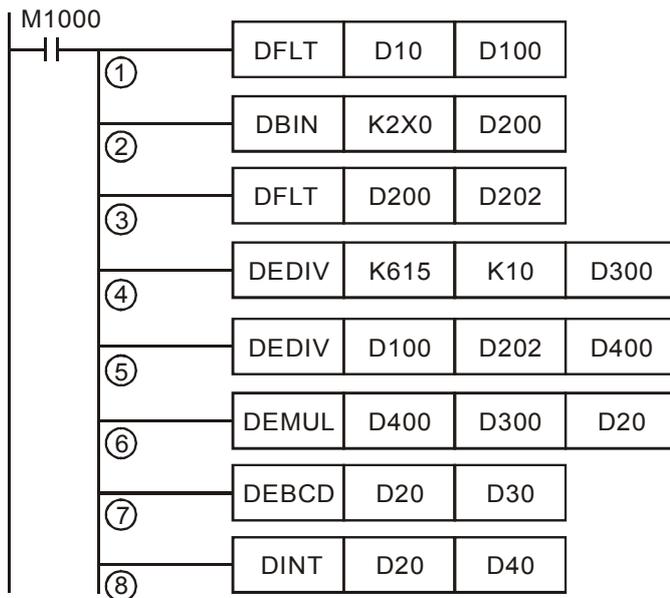


Program Example 2:

Apply FLT instruction to complete the following operation:



5 Categories and Use of Basic Application Instructions



- ① Convert D11, D10 (BIN integer) to D101, D100 (floating point).
- ② Convert the value of X7~X0 (BCD value) to D201, D200 (BIN value).
- ③ Convert D201, D200 (BIN integer) to D203, D202 (floating point).
- ④ Save the result of $K615 \div K10$ to D301, D300 (floating point).
- ⑤ Divide the floating point:
Save the result of $(D101, D100) \div (D203, D202)$ to D401, D400 (floating point)..
- ⑥ Multiply floating point:
Save the result of $(D401, D400) \times (D301, D300)$ to D21, D20 (floating point)
- ⑦ Convert floating point (D21, D20) to decimal floating point (D31, D30)..
- ⑧ Convert floating point (D21, D20) to BIN integer (D41, D40).

5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands		Function										Controllers	
	50	REF	P	D	n	Refresh										20PM

OP	Type	Bit Devices				Word devices										Program Steps		
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	REF, REFP: 5 steps	
D		*	*															
n						*	*											

Operands:

D: Start device for I/O refresh **n:** Number of devices for I/O refresh

Explanations:

- PLC updates I/O status between END instruction and the start of next program scan. If an immediate I/O refresh is needed, REF can be applied for performing I/O refresh immediately.
- D** can only be a multiple of 10, i.e. X0 or Y0, and the instruction is NOT applicable for I/O points on DIO modules.
- Only the I/O points on MPU can be specified for operand D for I/O refresh.
 - When **D** specifies X0 and $n \leq 8$, only X0~X7 will be refreshed. If $n > 8$, all I/O points on MPU will be refreshed.
 - When **D** specifies Y0 and $n = 4$, only Y0~X3 will be refreshed. If $n > 4$, all I/O points on MPU will be refreshed.
 - When **D** specifies X10 or Y10, I/O points on MPU starting from X10 or Y4 will all be refreshed regardless of **n** value, i.e. status of X0~X7 or Y0~Y3 remains.
- For EX2 MPU only: If M1180 = ON and REF instruction executes, PLC will read the A/D value and update the read value to D1110~D1113. If M1181 = ON and REF instruction executes, PLC will output the D/A value in D1116 and D1117 immediately. When A/D or D/A values are refreshed, PLC will reset M1180 or M1181 automatically.
- Range for **n**: 4 ~ total I/O points on MPU. **n** should always be a multiple of 4.

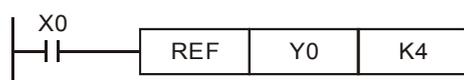
Program Example 1:

When X0 = ON, PLC will refresh the status of input points X0 ~ X7 immediately without delay.



Program Example 2:

When X0 = ON, the 4 output signals on Y0 ~ Y3 will be sent to output terminals immediately before the program proceeds to END instruction.



5 Categories and Use of Basic Application Instructions

Program Example 3:

When X0 = ON, I/O points starting from X10 or Y4 will all be refreshed.

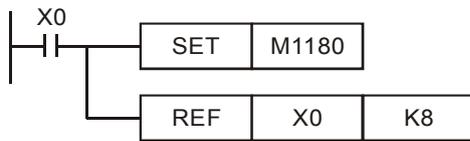


Or



Program Example 4:

For DVP-EX2 only: When X0 = ON and M1180 = ON, A/D signal in D1110~D1113 will be refreshed immediately regardless of the settings of operands **D** and **n**



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands				Function				Controllers	
	61	D	SER	P	S ₁	S ₂	D	n	Search a Data Stack				20PM

OP	Type	Bit Devices				Word devices										Program Steps		
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F	
S ₁								*	*	*	*	*	*	*	*	*		SER, SERP: 9 steps DSER, DSERP: 17 steps
S ₂					*	*	*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*	*	*		
N					*	*								*				

Operands:

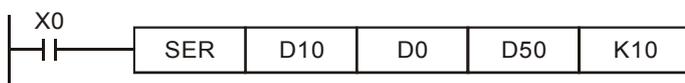
S₁: Start device of data stack **S₂**: Device to be searched **D**: Start device for storing search result (occupies 5 consecutive devices) **n**: Stack length

Explanations:

- SER instruction searches for the value stored in **S₂** from the data stack starting with **S₁**, with a stack length **n**. The search results are stored in the 5 registers starting from **D**
- D** stores the total of the matched results; **D+1** stores the No. of device storing the first matched result; **D+2** stores the No. of device storing the last matched result; **D+3** stores the No. of device storing the smallest value; **D+4** stores the No. of device storing the biggest value..
- When SER is used as 16-bit instruction, Z device cannot be adopted; when SER is used as 32-bit instruction, V device cannot be adopted.
- If the instruction applied 32-bit instruction, operands **S₁**, **S₂**, **D**, **n** will specify 32-bit registers.
- When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
- The range of operand **n**: **n** = 1~256 (16-bit instruction), **n** = 1~128 (32-bit instruction)

Program Example:

- When X0 = ON, the data stack D10~D19 are compared with D0 and the result is stored in D50~D54. If there is no matched result, the content of D50~D52 will all be 0.
- D53 and D54 store the location of the smallest and biggest value. When there are more than one smallest and biggest values, the devices with bigger No. will be recorded.



S ₁	Content	Data to be compared	Data No.	Result	D	Content	Explanation
D10	88	S ₂ D0=K100	0		D50	4	The total data numbers of equal value
D11	100		1	Equal	D51	1	The number of the first equal value
D12	110		2		D52	8	The number of the last equal value
D13	150		3		D53	7	The number of the smallest value
D14	100		4	Equal	D54	9	The number of the largest value
D15	300		5				
D16	100		6	Equal			
D17	5		7	Smallest			
D18	100		8	Equal			
D19	500		9	Largest			

5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands	Function	Controllers												
66	ALT	P	D	Alternate State	20PM	10PM											
OP	Type	Bit Devices				Word devices								Program Steps			
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ALT, ALTP: 3 steps
D		*	*	*													

Operands:

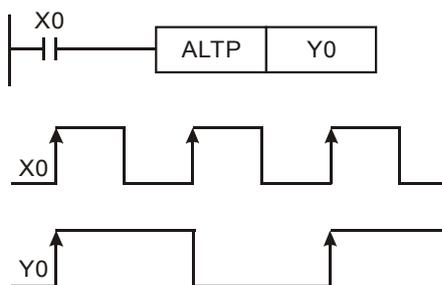
D: Destination device

Explanations:

1. The status of **D** is alternated every time when the ALT instruction is executed.
2. When ALT instruction is executed, ON/OFF state of **D** will be switched which is usually applied on switching two operation modes, e.g. Start/Stop
3. This instruction is generally used in pulse execution mode (ALTP).

Program Example 1:

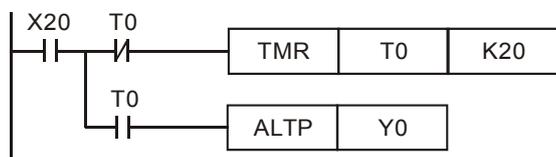
When X0 goes from OFF to ON, Y0 will be ON. When X0 goes from OFF to ON for the second time, Y0 will be OFF.



Program Example 2:

Creating a flashing circuit by applying ALTP with a timer

When X20 = ON, T0 will generate a pulse every two seconds and output Y0 will be switched between ON and OFF by the pulses from T0.



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands				Function								Controllers	
	67	D	RAMP	(S ₁)	(S ₂)	(D)	(n)	Ramp variable Value								20PM

OP	Type	Bit Devices				Word devices										Program Steps		
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F		
	S ₁													*			RAMP: 9 steps DRAMP: 17 steps	
	S ₂													*				
	D													*				
	n					*	*							*				

Operands:

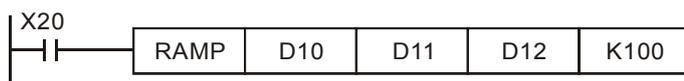
S₁: Start of ramp signal **S₂**: End of ramp signal **D**: Current value of ramp signal (occupies 2 consecutive devices) **n**: Times for scan (n: 1~32,767)

Explanations:

- This instruction creates a ramp output. A ramp output linearity depends on a consistent scan time. Therefore, scan time has to be fixed before executing RAMP instruction.
- When RAMP instruction is executed, the ramp signal will vary from **S₁** to **S₂**. Current value of ramp signal is stored in **D** and **D+1** stores the current number of accumulated scans. When ramp signal reaches **S₂**, or when the drive contact of RAMP instruction turns OFF, the content in **D** varies according to the setting of M1026 which is explained later in **Points to note**.
- When **n** specifies a D register, the value in D cannot be modified during the execution of the instruction. Please modify the content of D when the instruction is stopped.
- When this instruction is applied with analog output function, Ramp start and Ramp stop function can be achieved.

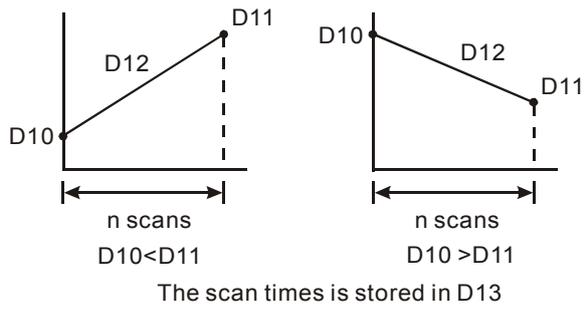
Program example:

- Before executing the instruction, first drive M1039 = ON to fix the scan time. Use MOV instruction to write the fixed scan time to the special data register D1039. Assume the scan time is 30ms and take the below program for example, n = K100, the time for D10 to increase to D11 will be 3 seconds (30ms × 100).
- When X20 goes OFF, the instruction will stop its execution. When X10 goes ON again, the content in D12 will be reset to 0 for recalculation
- When M1026 = OFF, M1029 will be ON to indicate the completion of ramp process and the content in D12 will be reset to the set value in D10.
- Set the Start and End of ramp signal in D10 and D11. When X20 = ON, D10 increases towards D11, the current value of the variation is stored in D12 and the number of current scans is stored in D13.



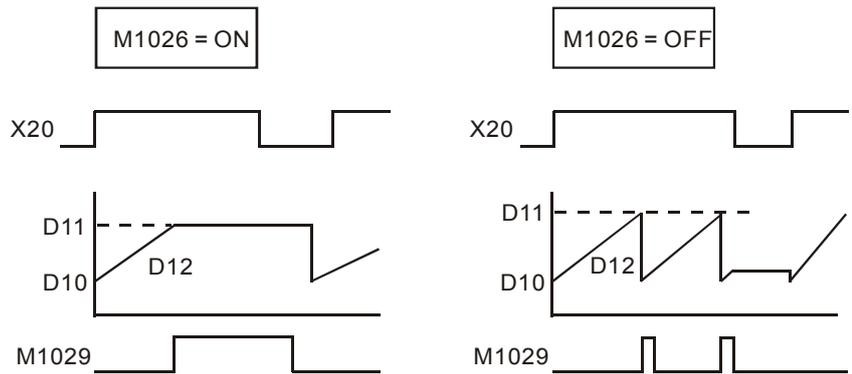
If X20 = ON,

5 Categories and Use of Basic Application Instructions



Points to note:

The variation of the content in D12 according to ON/OFF state of M1026:



5 Categories and Use of Basic Application Instructions

API	Mnemonic	Operands	Function	Controllers	
69	SORT	(S) (m ₁) (m ₂) (D) (n)	Data sort	20PM	10PM

Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E		F	
S																	SORT: 11 steps DSORT: 21 steps
m ₁					*	*											
m ₂					*	*											
D													*				
n					*	*							*				

Operands:

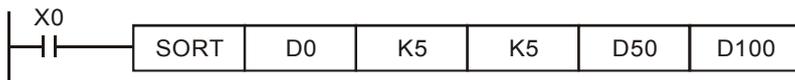
S: Start device for the source data **m₁:** Groups of data to be sorted (m₁=1~32) **m₂:** Number of columns in the table (m₂=1~6) **D:** Start device for the sorted data **n:** The No. of column to be sorted. (n=1~ m₂)

Explanations:

- The sorted data is stored in the m₁ × m₂ registers starting from the device designated in D. Therefore, if S and D designate the same register, the sorted results will be the same.
- It is suggested to specify the operand S with a number which is a multiple of 10, e.g. D0, D10, ...etc.
- SORT instruction is completed after m₁ times of scan. Once the SORT instruction is completed, the Flag M1029 (Execution completed flag) = ON.
- There is no limitation on the times of using this instruction in the program. However, only one instruction can be executed at a time

Program Example:

When X0 = ON, the sorting process starts. When the sorting is completed, M1029 will be ON. DO NOT change the data to be sorted during the execution of the instruction. If the sorting needs to be executed again, turn X0 from OFF to ON again.



Example table of data sort

		Columns of data: m ₂				
		Data Column				
Row	Column	1	2	3	4	5
		Students No.	English	Math.	Physics	Chemistry
Groups of data: m ₁	1	(D0) 1	(D5) 90	(D10) 75	(D15) 66	(D20) 79
	2	(D1) 2	(D6) 55	(D11) 65	(D16) 54	(D21) 63
	3	(D2) 3	(D7) 80	(D12) 98	(D17) 89	(D22) 90
	4	(D3) 4	(D8) 70	(D13) 60	(D18) 99	(D23) 50
	5	(D4) 5	(D9) 95	(D14) 79	(D19) 75	(D24) 69

5 Categories and Use of Basic Application Instructions

Sort data table when D100 = K3

Columns of data: m_2

		Data Column				
Column		1	2	3	4	5
Row		Students No.	English	Math.	Physics	Chemistry
Groups of data: m_1	1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
	2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
	3	(D52) 1	(D57) 90	(D62) 75	(D67) 66	(D72) 79
	4	(D53) 5	(D58) 95	(D63) 79	(D68) 75	(D73) 69
	5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

Sort data table when D100 = K5

Columns of data: m_2

		Data Column				
Column		1	2	3	4	5
Row		Students No.	English	Math.	Physics	Chemistry
Groups of data: m_1	1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
	2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
	3	(D52) 5	(D57) 95	(D62) 79	(D67) 75	(D72) 69
	4	(D53) 1	(D58) 90	(D63) 75	(D68) 66	(D73) 79
	5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands				Function							Controllers		
	D	FROM	P	m ₁	m ₂	D	n	Read CR data from Special Modules							20PM	10PM	
78	Type		Bit Devices				Word Devices							Program Steps			
	OP	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	FROM, FROMP: 9 steps DFROM, DFROMP: 17 steps
		m ₁				*	*					*	*	*	*	*	
		m ₂				*	*					*	*	*	*	*	
		D										*	*	*	*	*	
n					*	*					*	*	*	*	*		

Operands:

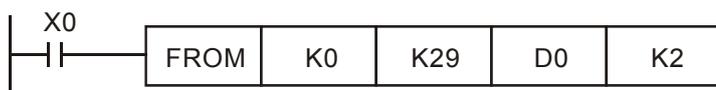
m₁: No. of special module m₂: CR# in special module to be read D: Device for storing read data n: Number of data to be read at a time

Explanations:

- DVP-PM uses this instruction to read CR (Control register) data from special modules
- Range of m₁ (16-bit and 32-bit): 0 ~ 255
- Range of m₂ (16-bit and 32-bit): 0 ~ 499
- Range of n:
 - 16-bit: 1 ~ (500 - m₂)
 - 32-bit: 1 ~ (500 - m₂)/2
- When FROM is used as 16-bit instruction, Z device cannot be adopted; when FROM is used as 32-bit instruction, V device cannot be adopted.
- See **Points to note** of API 79 TO for the numbering of special modules.

Program Example:

- Read out the data in CR#29 of special module N0.0 to register D0 in PLC, and CR#30 of special module No.0 to register D1 in PLC. 2 consecutive 16-bit data are read at one time (n = 2).
- When X0 = ON, the instruction executes; when X0 = OFF, the previous content in D0 and D1 won't be changed.



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands				Function						Controllers	
79	D	TO	P	m_1	m_2	S	n	Write CR data into Special Modules						20PM	10PM

OP	Type	Bit Devices				Word Devices								Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	TO, TOP: 9 steps DTP, DTP: 17 steps
m_1					*	*					*	*	*	*	*	
m_2					*	*					*	*	*	*	*	
S					*	*					*	*	*	*	*	
n					*	*					*	*	*	*	*	

Operands:

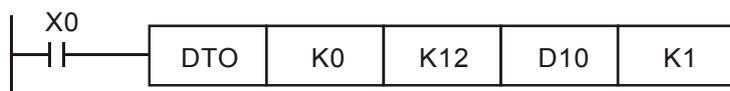
m_1 : No. of special module m_2 : CR# in special module to be written S: Data to be written in CR
n: Number of data to be written at a time

Explanations:

- DVP-PM uses this instruction to write data into CR (Control register) on special modules
- Range of m_1 (16-bit and 32-bit): 0 ~ 255
- Range of m_2 (16-bit and 32-bit): 0 ~ 499
- Range of n:
16-bit: 1 ~ (500 - m_2)
32-bit: 1 ~ (500 - m_2)/2
- When TO is used as 16-bit instruction, Z device cannot be adopted; when TO is used as 32-bit instruction, V device cannot be adopted.

Program Example:

- Use 32-bit instruction DTO to write the content in D11 and D10 into CR#13 and CR#12 of special module No.0. One 32-bit data is written at a time (n = 1).
- When X0 = ON, the instruction executes; when X0 = OFF, the previous content in D10 and D11 won't be changed..



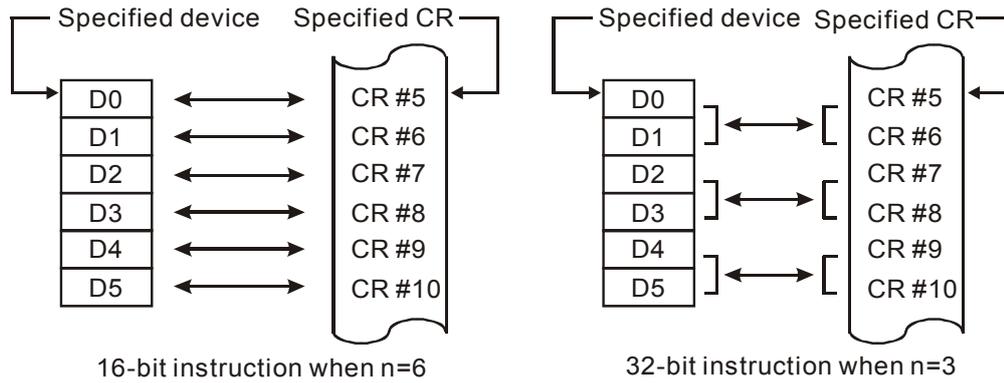
Points to note:

- m_1 : number of special module. The modules are numbered from 0 (closest to MPU) to 7 automatically by their distance from MPU. Maximum 8 modules are allowed to connect to MPU and will not occupy any digital I/O points.
- m_2 : number of CR (Control Register). CR is the 16-bit memory built in the special module for control or monitor purpose, numbering in decimal. All operation status and settings of the special module are recorded in the CR.
- FROM/TO instruction reads/writes 1 CR at a time. DFROM/DTO instruction reads/writes 2 CRs at a time.
Higher 16-bit Lower 16-bit



- n: Number of data to be written at a time. n = 2 in 16-bit instruction has the same operation results as n = 1 in 32-bit instruction.

5 Categories and Use of Basic Application Instructions



Application Example 1:

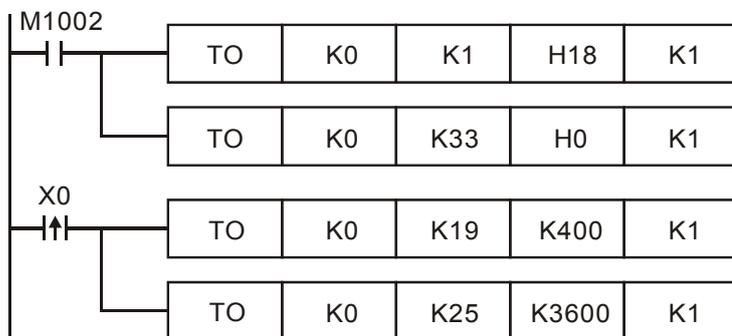
Adjust the A/D conversion curve of DVP04AD-H2. Set the OFFSET value of CH1 as 0V (= $K0_{LSB}$) and GAIN value as 2.5V (= $K2,000_{LSB}$).



1. Write H'0 into CR#1 of analog input module No. 0 to set CH1 as mode 0 (voltage input: -10V ~ +10V).
2. Write H'0 into CR#33 to allow OFFSET/GAIN tuning in CH1 ~ CH4.
3. When X0 goes from OFF to ON, write the OFFSET value $K0_{LSB}$ into CR#18 and GAIN value $K2,000_{LSB}$ into CR#24.

Application Example 2:

Adjust the A/D conversion curve of DVP04AD-H2. Set the OFFSET value of CH2 as 2mA (= $K400_{LSB}$) and GAIN value as 18mA (= $K3,600_{LSB}$).

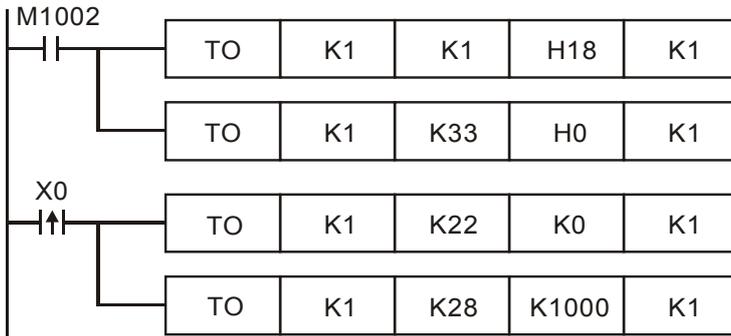


1. Write H'18 into CR#1 of analog input module No. 0 to set CH2 as mode 3 (current input: -20mA ~ +20mA).
2. Write H'0 into CR#33 to allow OFFSET/GAIN tuning in CH1 ~ CH4.
3. When X0 goes from OFF to ON, write the OFFSET value $K400_{LSB}$ into CR#19 and the GAIN value $K3,600_{LSB}$ into CR#25.

FROM/TO Application Example 3:

5 Categories and Use of Basic Application Instructions

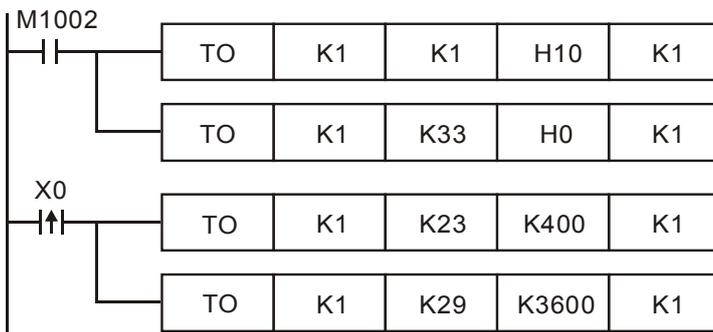
Adjust the D/A conversion curve of DVP02DA-H2. Set the OFFSET value of CH2 as 0mA (= $K0_{LSB}$) and GAIN value as 10mA (= $K1,000_{LSB}$).



1. Write H'18 into CR#1 of analog output module No. 1 to set CH2 as mode 3 (current output: 0mA ~ +20mA).
2. Write H'0 into CR#33 to allow OFFSET/GAIN tuning in CH1 and CH2.
3. When X0 goes from OFF to ON, write the OFFSET value $K0_{LSB}$ into CR#22 and the GAIN value $K1,000_{LSB}$ into CR#28.

FROM/TO Application Example 4:

Adjust the D/A conversion curve of DVP02DA-H2. Set the OFFSET value of CH2 as 2mA (= $K400_{LSB}$) and GAIN value as 18mA (= $K3,600_{LSB}$).



1. Write H'10 to CR#1 of analog output module No. 1 to set CH2 as mode 2 (current output: +4mA ~ +20mA).
2. Write H'0 to CR#33 to allow OFFSET/GAIN tuning in CH1 and CH2.
3. When X0 goes from OFF to ON, write the OFFSET value $K400_{LSB}$ into CR#23 and GAIN value $K3,600_{LSB}$ into CR#29.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands	Function										Controllers		
	D	ABS	P		Absolute Value										20PM	10PM	
87	D	ABS	P	D													
Type OP	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ABS, ABSP: 3 steps	DABS, DABSP: 5 steps
D								*	*	*	*	*	*	*	*		

Operands:

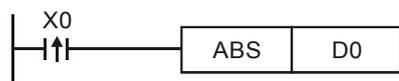
D: Device for absolute value operation

Explanation

1. The instruction conducts absolute value operation on **D**
2. This instruction is generally used in pulse execution mode (ABSP, DABSP).
3. When ABS is used as 16-bit instruction, Z device cannot be adopted; when ABS is used as 32-bit instruction, V device cannot be adopted.
4. When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).

Program Example:

When X0 goes from OFF to ON, ABS instruction obtains the absolute value of the content in D0.



5 Categories and Use of Basic Application Instructions

DVP-PM ⇒ VFD-B, DVP-PM transmits: “01 03 2101 0006 D4”

VFD-B ⇒ DVP-PM, DVP-PM receives: “01 03 0C 0100 1766 0000 0000 0136 0000 3B”

Registers for data to be sent (sending messages)

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	Command 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 (count by word)	
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	

Registers for received data (responding messages)

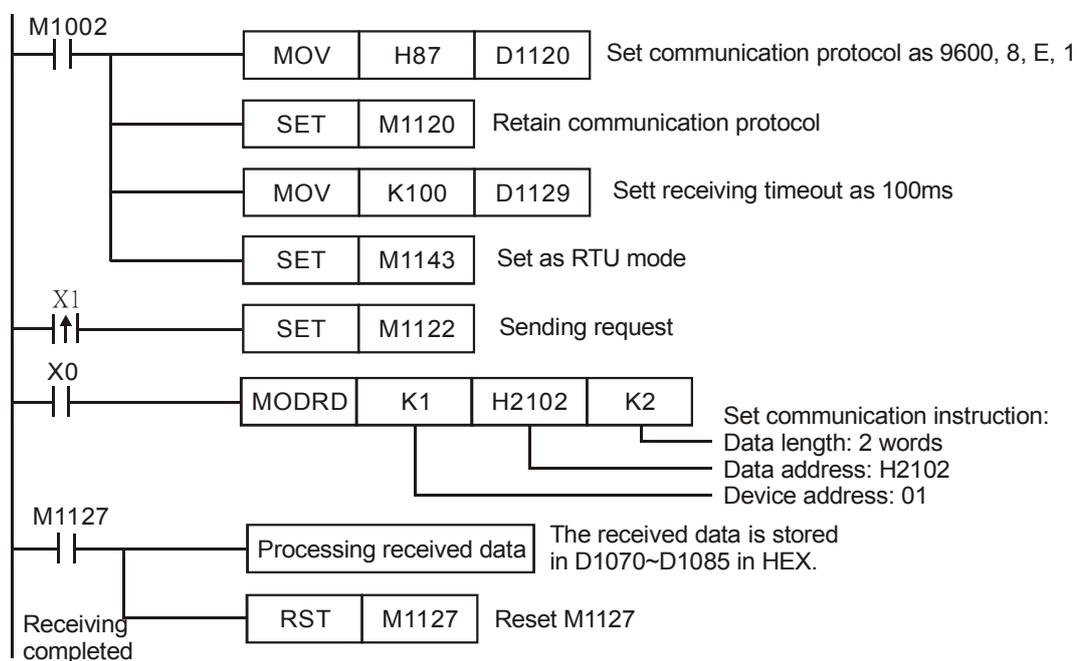
Register	DATA		Explanation	
D1070 low	'0'	30 H	ADR 1	
D1070 high	'1'	31 H	ADR 0	
D1071 low	'0'	30 H	CMD 1	
D1071 high	'3'	33 H	CMD 0	
D1072 low	'0'	30 H	Number of Data (count by byte)	
D1072 high	'C'	43 H		
D1073 low	'0'	30 H	Content of address 2101 H	0100 H PLC automatically converts ASCII codes and store the converted value in D1050
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	1766 H PLC automatically converts ASCII codes and store the converted value in D1051
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	0000 H PLC automatically converts ASCII codes and store the converted value in D1052
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	0000 H PLC automatically converts ASCII codes and store the converted value in D1053
D1079 high	'0'	30 H		
D1080 low	'0'	30 H		
D1080 high	'0'	30 H		
D1081 low	'0'	30 H	Content in address	0136 H

5 Categories and Use of Basic Application Instructions

Register	DATA		Explanation	
D1081 high	'1'	31 H	2105 H	converts ASCII codes and store the converted value in D1054
D1082 low	'3'	33 H		
D1082 high	'6'	36 H		
D1083 low	'0'	30 H	Content in address 2106 H	0000 H PLC automatically converts ASCII codes and store the converted value in D1055
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	

Program Example 2:

Communication between DVP-PM and VFD-B series AC motor drive (RTU mode, M1143 = ON)



DVP-PM ⇒ VFD-B, DVP-PM transmits: **"01 03 2102 0002 6F F7"**

VFD-B ⇒ DVP-PM, DVP-PM receives: **"01 03 04 1770 0000 FE 5C"**

Registers for data to be sent (sending messages)

Register	DATA	Explanation
D1089 low	01 H	Address of AC motor drive
D1090 low	03 H	Command code of AC motor drive
D1091 low	21 H	Starting data address
D1092 low	02 H	
D1093 low	00 H	Number of data (count by word)
D1094 low	02 H	
D1095 low	6F H	CRC CHK Low
D1096 low	F7 H	CRC CHK High

Registers for received data (responding messages)

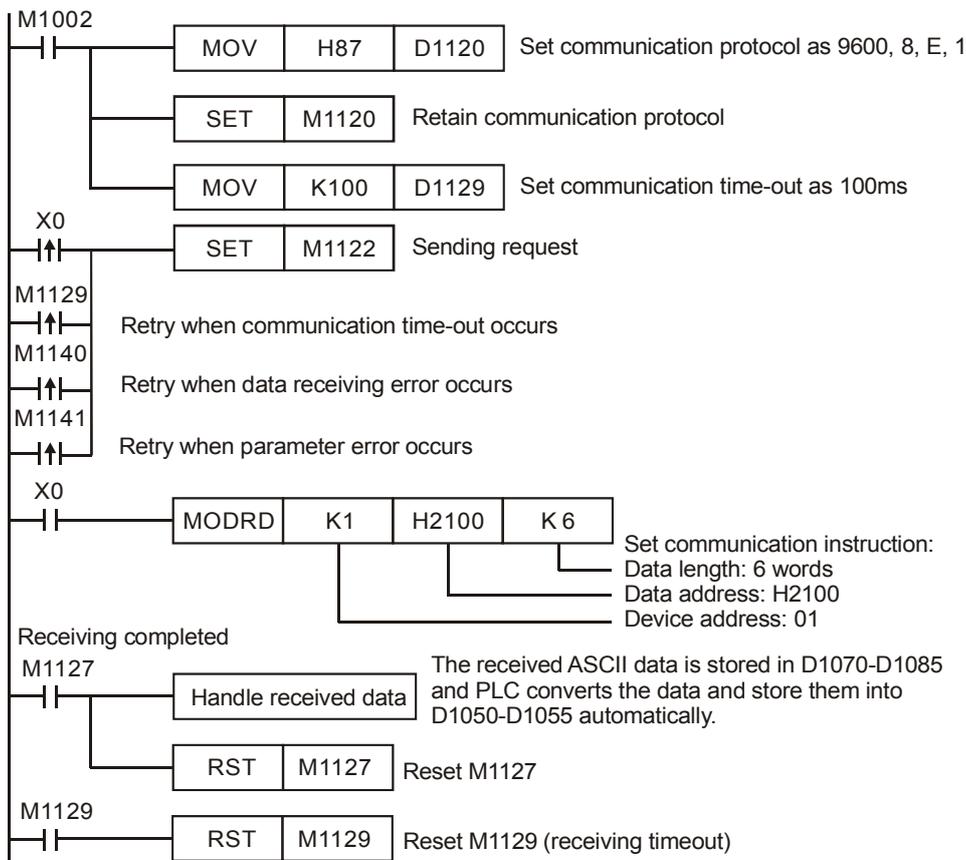
Register	DATA	Explanation
D1070 low	01 H	Address of AC motor drive

5 Categories and Use of Basic Application Instructions

D1071 low	03 H	Command code of AC motor drive
D1072 low	04 H	Number of data (count by byte)
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

Program Example 3:

1. In the communication between DVP-PM and VFD-B series AC motor drive (ASCII Mode, M1143 = OFF), executes Retry when communication time-out, data receiving error or parameter error occurs.
2. When X0 = ON, DVP-PM will read the data of address H2100 in device 01(VFD-B) and stores the data in ASCII format in D1070 ~ D1085. PLC will automatically convert the data and store them in D1050 ~ D1055.
3. M1129 will be ON when communication time-out occurs. The program will trigger M1129 and send request for reading the data again.
4. M1140 will be ON when data receiving error occurs. The program will trigger M1140 and send request for reading the data again.
5. M1141 will be ON when parameter error occurs. The program will trigger M1141 and send request for reading the data again.



Points to note:

1. Rising-edge contact (LDP, ANDP, ORP) and falling-edge contact (LDF, ANDF, ORF) can not be used with MODRD instruction, otherwise the data stored in the receiving registers will be incorrect.
2. There is no limitation on the times of using this instruction, but only one instruction can be executed at a time.

5 Categories and Use of Basic Application Instructions

API	Mnemonic	Operands	Function		Controllers	
			Write Modbus Data		20PM	10PM
101	MODWR	(S ₁) (S ₂) (n)				

OP	Type	Bit Devices				Word Devices								Program Steps			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D		V	Z	
	S ₁					*	*						*				MODWR: 7 steps
	S ₂					*	*						*				
	n					*	*						*				

Operands:

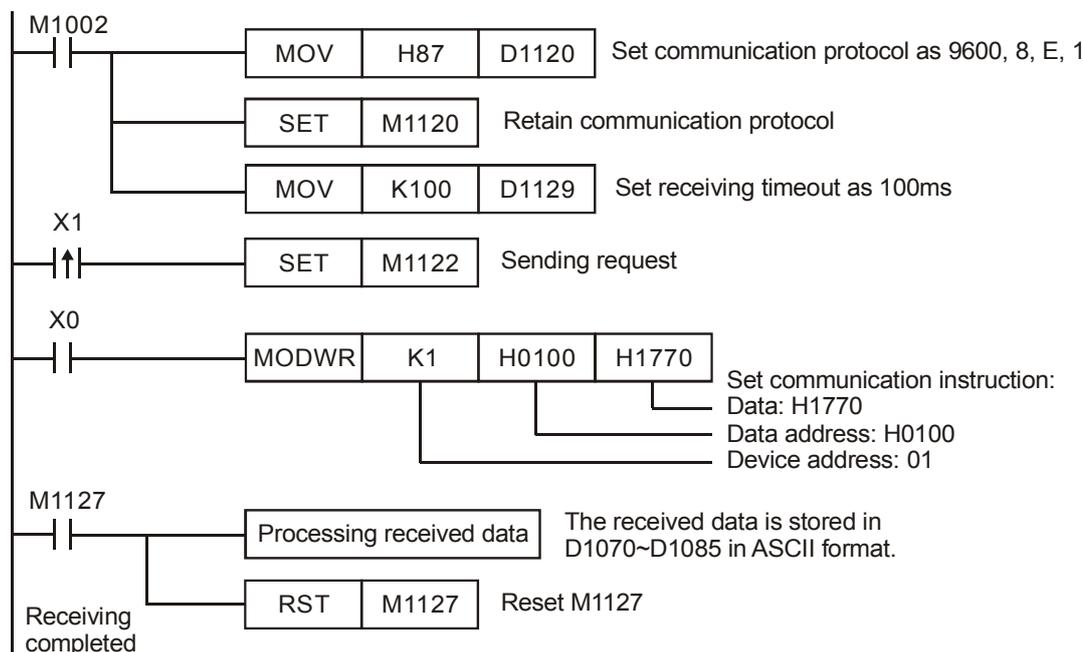
S₁: Device address S₂: Data address n: Data to be written

Explanations:

1. Range of S₁: K0 ~ K254.
2. MODWR is an instruction exclusively for peripheral communication equipment in MODBUS ASCII/RTU mode. The built-in RS-485 communication ports in Delta VFD drives (except for VFD-A series) are all compatible with MODBUS communication format. MODRD can be used for communication (write data) of Delta drives.
3. If the address of S₂ is illegal for the designed communication device, the device will respond with an error, PLC will record the error code in D1130 and M1141 will be ON. For example, if 8000H is invalid to VFD-B, M1141 will be ON and D1130 = 2. For error code explanations, please see the user manual of VFD-B.
4. The feedback (returned) data from the peripheral equipment will be stored in D1070 ~ D1085. After data receiving is completed, PLC will check the validity of the data automatically. If there is an error, M1140 will be ON.
5. If peripheral device receives a correct record (data) from PLC after M1140/M1141 = ON, the peripheral device will send out feedback data and PLC will reset M1140/M1141 after the validity of data is confirmed.
6. Flags: M1120 ~ M1129, M1140 ~ M1143.

Program Example 1:

Communication between DVP-PM and VFD-B series AC motor drive (ASCII mode, M1143 = OFF)



5 Categories and Use of Basic Application Instructions

DVP-PM ⇒ VFD-B, DVP-PM transmits: “01 06 0100 1770 71”

VFD-B ⇒ DVP-PM, DVP-PM receives: “01 06 0100 1770 71”

Registers for data to be sent (sending messages)

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	Command code of AC motor drive: 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	

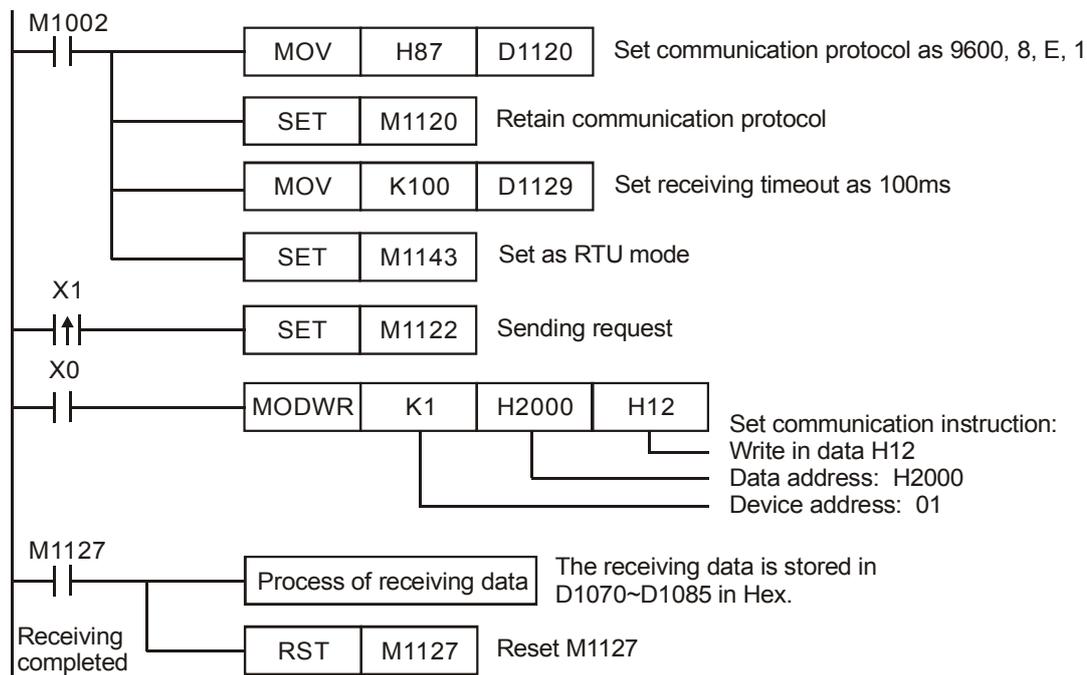
Registers for received data (responding messages)

Register	DATA		Explanation	
D1070 low	'0'	30 H	ADR 1	ADR 0
D1070 high	'1'	31 H	ADR 0	
D1071 low	'0'	30 H	CMD 1	CMD 0
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 content	
D1074 high	'7'	37 H		
D1075 low	'7'	37 H		
D1075 high	'0'	30 H		
D1076 low	'7'	37 H	LRC CHK 1	LRC CHK 0
D1076 high	'1'	31 H	LRC CHK 0	

5 Categories and Use of Basic Application Instructions

Program Example 2:

Communication between DVP-PM and VFD-B series AC motor drive (RTU mode, M1143 = ON)



DVP-PM ⇒ VFD-B, DVP-PM transmits: "01 06 2000 0012 02 07"

VFD-B ⇒ DVP-PM, DVP-PM receives: "01 06 2000 0012 02 07"

Registers for sent data (sending messages)

Register	DATA	Explanation
D1089 low	01 H	Address of AC motor drive
D1090 low	06 H	Command code of AC motor drive
D1091 low	20 H	Data Address
D1092 low	00 H	
D1093 low	00 H	Data content
D1094 low	12 H	
D1095 low	02 H	CRC CHK Low
D1096 low	07 H	CRC CHK High

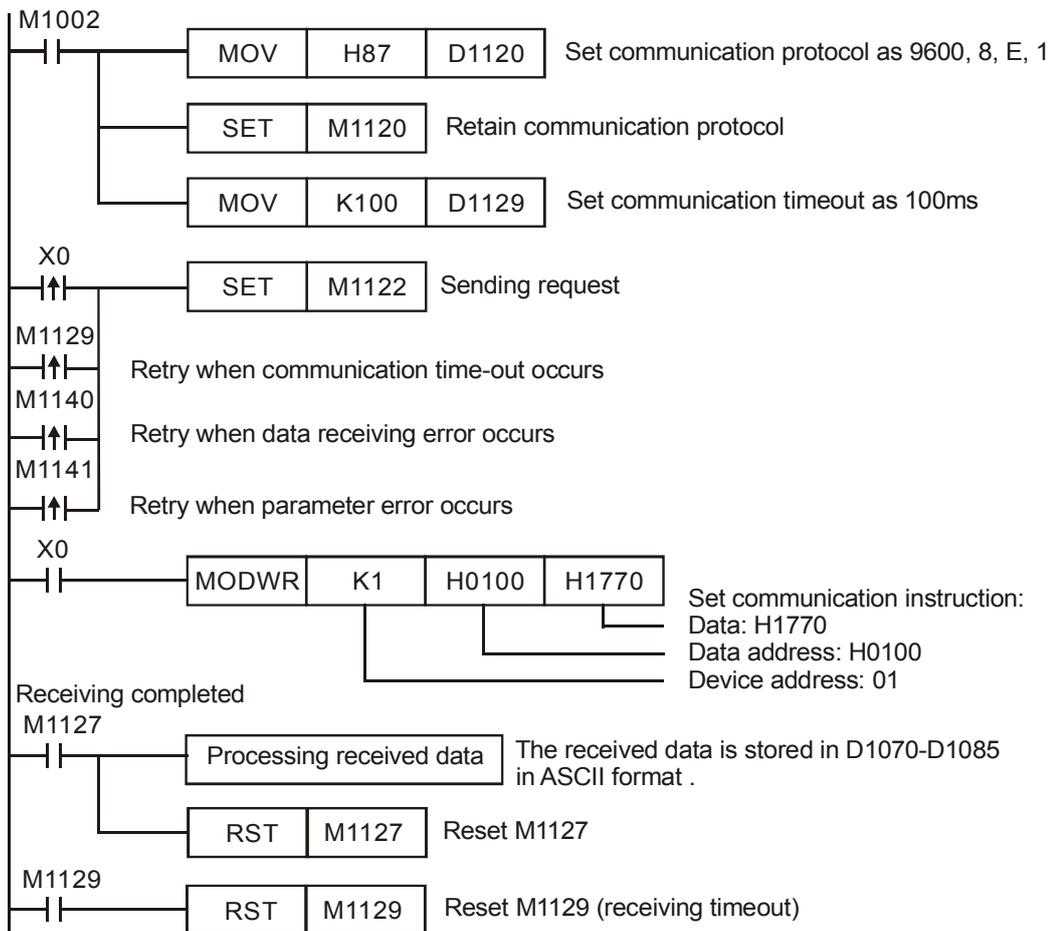
Registers for received data (responding messages)

Register	DATA	Explanation
D1070 low	01 H	Address of AC motor drive
D1071 low	06 H	Command code of AC motor drive
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

5 Categories and Use of Basic Application Instructions

Program Example 3:

1. In the communication between DVP-PM and VFD-B series AC motor drive (ASCII Mode, M1143 = OFF), executes Retry when communication time-out, data receiving error or parameter error occurs.
2. When X0 = ON, DVP-PM will write data H1770 K6000) into address H0100 in device 01 (VFD-B).
3. M1129 will be ON when communication time-out occurs. The program will trigger M1129 and send request for reading the data again.
4. M1140 will be ON when data receiving error occurs. The program will trigger M1140 and send request for reading the data again.
5. M1141 will be ON when parameter error occurs. The program will trigger M1141 and send request for reading the data again.

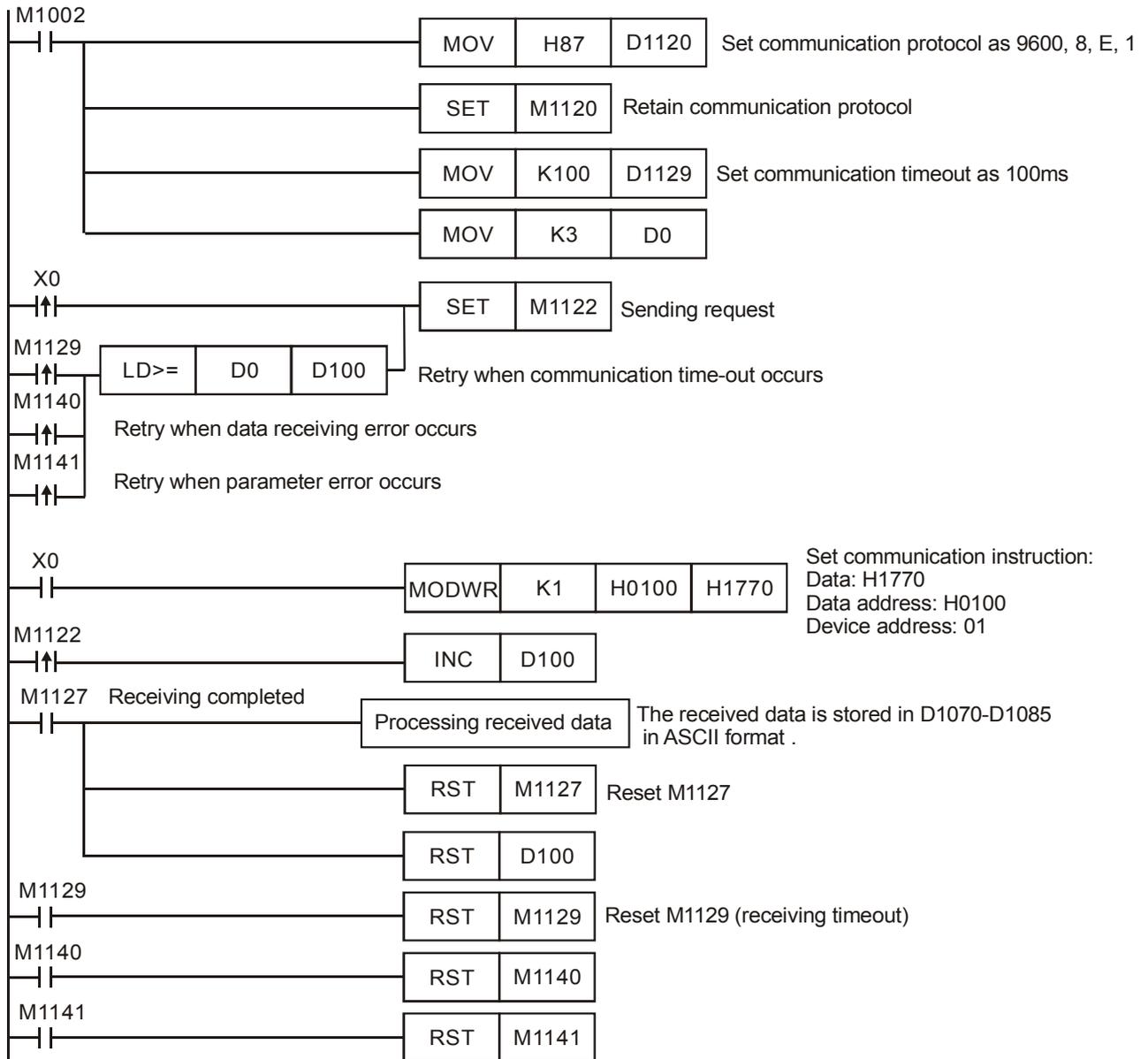


Program Example 4:

1. In the communication between DVP-PM and VFD-B series AC motor drive (ASCII Mode, M1143 = OFF), executes Retry when communication time-out, data receiving error or parameter error occurs. Set the times of retry as 3 in D0. When communication retry is successful, the communication can again be triggered by users.
2. When X0 = ON, PLC will write data H1770 (K6000) into address H0100 in device 01 (VFD-B).
3. M1129 will be ON when communication time-out occurs. The program will trigger M1129 and send request for reading the data again. The specified times of retry: 3 (D0 = 3).
4. M1140 will be ON when data receiving error occurs. The program will trigger M1140 and send request for reading the data again. The specified times of retry: 3 (D0 = 3).
5. M1141 will be ON when parameter error occurs. The program will trigger M1141 and send request for reading

5 Categories and Use of Basic Application Instructions

the data again. The specified times of retry: 3 (D0 = 3).



Points to note:

1. If rising-edge contacts (LDP, ANDP, ORP) or falling-edge contacts (LDF, ANDF, ORF) is used before MODWR instruction, sending request flag M1122 has to be executed as a requirement..
2. There is no limitation on the times of using this instruction, but only one instruction can be executed at a time on the same COM port.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands				Function				Controllers			
	D	EZCP	P	S ₁	S ₂	S	D	Floating Point Zone Compare				20PM	10PM		
111															
Type OP	Bit Devices				Word Devices								Program Steps		
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S ₁					*								*		
S ₂					*								*		
S					*								*		
D		*	*	*											
DEZCP, DEZCPP: 12 steps															

Operands:

S₁: Lower bound of zone comparison S₂: Upper bound of zone comparison

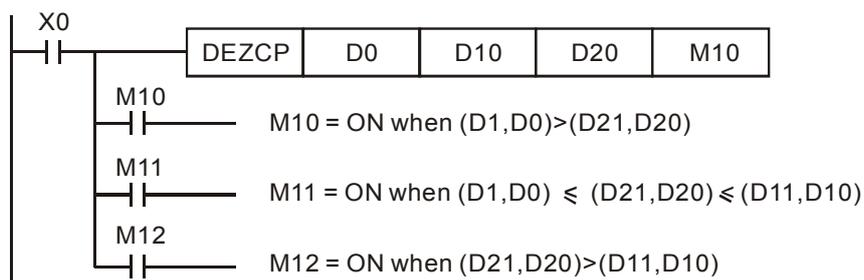
S: Comparison value D: Comparison result

Explanations:

- The data of **S** is compared to the data range of **S₁ ~ S₂** and the result (>, =, <) is indicated by three bit devices in **D**
- D** occupies 3 consecutive devices.
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions **DEZCP** and **DEZCPP** are applicable.
- If **S₁** or **S₂** is specified with floating point value F, the instruction will operate in binary floating point.
- Operand **S₁** should be smaller than operand **S₂**. When **S₁ > S₂**, the instruction takes **S₁** as the 1st comparison value and performs normal comparison similar to ECMP instruction.

Program Example:

- If the specified device is M10, M10~M12 will automatically be used.
- When X0 = ON, one of M10~M12 will be ON. When X0 = OFF, **DEZCP** instruction is not executed, M10~M12 will retain their previous state before X0 = OFF.
- RST or ZRST instruction is required if users need to reset the comparison result.



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
	116	D	RAD	P	S	D	Degree → Radian										20PM

Type OP	Bit Devices				Word Devices										Program Steps	
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V		Z
S					*								*			DRAD, DRADP: 6 steps
D													*			

Operands:

S: Source device (degree) **D:** Conversion result (radian)

Explanations:

- Use the following formula to convert degree to radian:

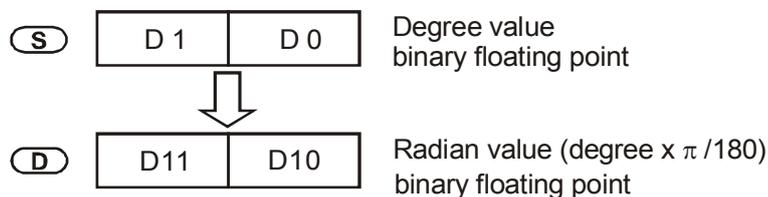
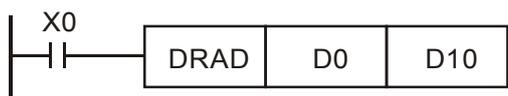
$$\text{Radian} = \text{degree} \times (\pi/180)$$
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions **DRAD** and **DRADP** are applicable.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the absolute value of the result exceeds the max. floating point value, carry flag will be ON.
- If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- If the conversion result is 0, zero flag will be ON.

Program Example:

When X0 = ON, convert degree value of the binary floating point in (D1, D0) to radian and save the binary floating point result in (D11, D10).



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers			
	D	DEG	P	S	D	Radian → Degree										20PM	10PM		
117																			
Type OP	Bit Devices				Word Devices										Program Steps				
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DDEG, DDEGP: 6 steps			
S					*								*						
D													*						

Operands:

S: Source device (radian) **D:** Conversion result (degree)

Explanations:

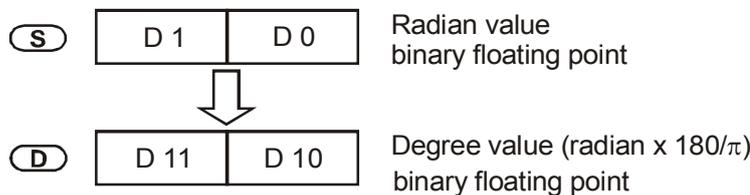
- Use the following formula to convert radian to degree:
Degree = Radian × (180/π)
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions **DDEG** and **DDEGP** are applicable.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the absolute value of the result exceeds the max. floating point value, carry flag will be ON.
- If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- If the conversion result is 0, zero flag will be ON

Program Example:

When X0 = ON, convert the radian of the binary floating point in (D1, D0) to degree and save the binary floating point result in (D11, D10).



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function							Controllers	
	120	D	EADD	P	S ₁	S ₂	D	Binary Floating Point Addition							20PM

Type OP	Bit Devices				Word Devices										Program Steps	
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V		Z
S ₁					*								*			DEADD, DEADDP: 9 steps
S ₂					*								*			
D													*			

Operands:

S₁: Augend S₂: Addend D: Addition result

Explanations:

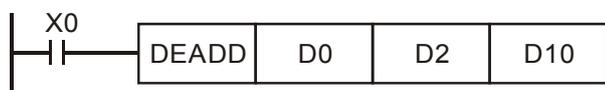
1. $S_1 + S_2 = D$. The floating point value in S_1 and S_2 are added and the result is stored in D
2. F refers to floating point input. Be sure to add a decimal point when using it.
3. Only 32-bit instructions DEADD and DEADDP are applicable..
4. If S_1 or S_2 is specified with floating point value F, the instruction will operate in binary floating point.
5. S_1 and S_2 can designate the same register. In this case, if the instruction is specified as “continuous execution instruction” (generally DEADDP instruction) and the drive contact is ON, the register will be added once in every scan.
6. Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- a) If the absolute value of the result exceeds the max. floating point value, carry flag will be ON.
- b) If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- c) If the conversion result is 0, zero flag will be ON

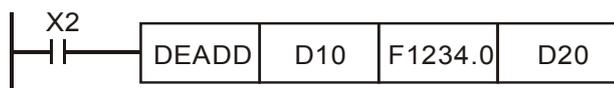
Program Example 1:

When X0 = ON, add the binary floating point value (D1, D0) with binary floating point value (D3, D2) and store the result in (D11, D10).



Program Example 2:

When X2 = ON, add the binary floating point value of (D11, D10) with F1234.0 (automatically converted to binary floating point value) and store the result in (D21, D20).



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function							Controllers				
	D	ESUB	P	S ₁	S ₂	D	Binary Floating Point Subtraction							20PM	10PM			
121																		
OP	Type	Bit Devices				Word Devices								Program Steps				
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DESUB, DESUBP: 9 steps	
	S ₁					*								*				
	S ₂					*								*				
	D													*				

Operands:

S₁: Minuend S₂: Subtrahend D: Subtraction result

Explanations:

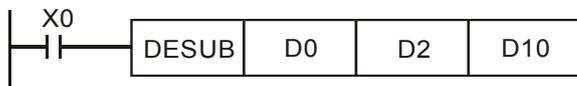
- S₁ – S₂ = D. The floating point value in S₂ is subtracted from the floating point value in S₁ and the result is stored in D. The subtraction is conducted in binary floating point format.
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions **DEADD** and **DEADDP** are applicable.
- If S₁ or S₂ is specified with floating point value F, the instruction will operate in binary floating point.
- S₁ and S₂ can designate the same register. In this case, if the instruction is specified as “continuous execution instruction” (generally **DESUBP** instruction) and the drive contact is ON, the register will be subtracted once in every scan.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the absolute value of the result exceeds the max. floating point value, carry flag will be ON.
- If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- If the conversion result is 0, zero flag will be ON.

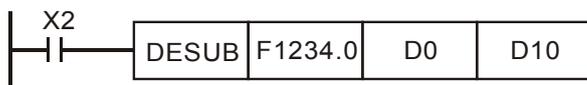
Program Example 1:

When X0 = ON, binary floating point value (D1, D0) minuses binary floating point value (D3, D2) and the result is stored in (D11, D10).



Program Example 2:

When X2 = ON, F1234.0 (automatically converted into binary floating point value) minuses binary floating point (D1, D0) and the result is stored in (D11, D10).



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function							Controllers				
	D	EMUL	P	S ₁	S ₂	D	Binary Floating Point Multiplication							20PM	10PM			
122																		
OP	Type	Bit Devices				Word Devices								Program Steps				
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DEMUL, DEMULP: 9 steps	
	S ₁					*								*				
	S ₂					*								*				
	D													*				

Operands:

S₁: Multiplicand **S₂**: Multiplier **D**: Multiplication result

Explanations:

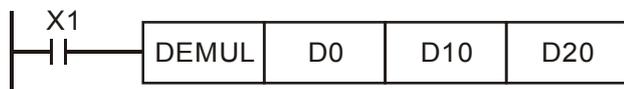
- S₁ × S₂ = D**. The floating point value in **S₁** is multiplied with the floating point value in **S₂** and the result is **D**. The multiplication is conducted in binary floating point format
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions **DEMUL** and **DEMULP** are applicable.
- If **S₁** or **S₂** is specified with floating point value F, the instruction will operate in binary floating point
- S₁** and **S₂** can designate the same register. In this case, if the instruction is specified as “continuous execution instruction” (generally **DEMULP** instruction) and the drive contact is ON, the register will be multiplied once in every scan.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the absolute value of the result exceeds the max floating point value, carry flag will be ON.
- If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- If the conversion result is 0, zero flag will be ON.

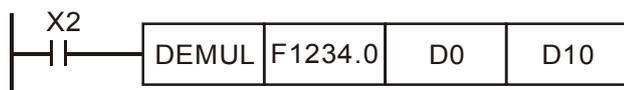
Program Example 1:

When X1 = ON, binary floating point (D1, D0) multiplies binary floating point (D11, D10) and the result is stored in (D21, D20).



Program Example 2:

When X2 = ON, F1234.0 (automatically converted into binary floating point value) multiplies binary floating point (D1, D0) and the result is stored in (D11, D10).



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function							Controllers	
	123	D	EDIV	P	S ₁	S ₂	D	Binary Floating Point Division							20PM

OP	Type	Bit Devices				Word Devices								Program Steps			
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S ₁						*								*			DEDIV, DEDIVP: 9 steps
S ₂						*								*			
D														*			

Operands:

S₁: Dividend S₂: Divisor D: Quotient and remainder

Explanations:

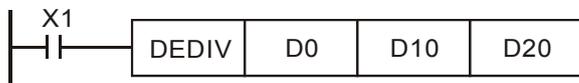
- S₁ ÷ S₂ = D. The floating point value in S₁ is divided by the floating point value in S₂ and the result is stored in D. The division is conducted in binary floating point format
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions DEDIV and DEDIVP are applicable.
- If S₁ or S₂ is specified with floating point value F, the instruction will operate in binary floating point.
- If S₂ = 0, operation error will occur, the instruction will not be executed. The operational error flag will be ON, and the error code H'0E19 will be recorded.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970
Operation error	M1793	M1953

- If the absolute value of the result exceeds the max floating point value, carry flag will be ON.
- If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- If the conversion result is 0, zero flag will be ON.

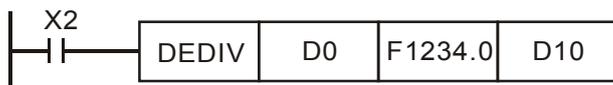
Program Example 1:

When X1 = ON, binary floating point value of (D1, D0) is divided by binary floating point (D11, D10) and the quotient and remainder is stored in (D21, D20).



Program Example 2:

When X2 = ON, binary floating point value of (D1, D0) is divided by F1234.0 (automatically converted to binary floating point value) and the result is stored in (D11, D10).



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
	124	D	EXP	P	S	D	Float Exponent Operation										20PM

OP	Type	Bit Devices				Word Devices										Program Steps		
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DEXP, DEXPP: 6 steps	
S						*								*				
D														*				

Operands:

S: Exponent **D:** Operation result

Explanations:

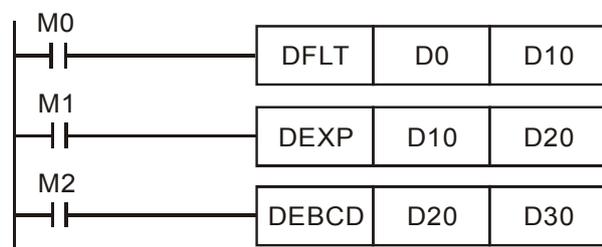
- $e = 2.71828$ as the base and **S** as exponent for EXP operation: $EXP^{[D+1, D]} = [S + 1, S]$
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions **DEXP** and **DEXPP** are applicable.
- Both positive and negative values are valid for **S**. Register **D** has to be 32-bit format. Operation is conducted in floating point value, so the value in **S** needs to be converted into floating value before exponent operation.
- The content in **D**: e^S , $e = 2.71828$ and **S** is the specified exponent.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the absolute value of the result exceeds the max floating point value, carry flag will be ON.
- If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- If the conversion result is 0, zero flag will be ON.

Program Example:

- When M0 = ON, convert (D1, D0) to binary floating value and save the result in (D11, D10).
- When M1 = ON, perform exponent operation with (D11, D10) as the exponent. The value is saved in register (D21, D20) in binary floating format.
- When M2 = ON, convert the value in (D21, D20) into decimal floating point value and save the result in (D31, D30). (At this time, D31 indicates powers of 10 for D30)



Points to note:

For floating point operations, see "5.3 Numeric Values for Data Processing"

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function								Controllers			
	D	LOG	P	S ₁	S ₂	D	Float Logarithm Operation								20PM	10PM		
126																		
OP	Type	Bit Devices				Word Devices										Program Steps		
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DLOG, DLOGP: 9 steps	
	S ₁					*								*				
	S ₂					*								*				
	D													*				

Operands:

S₁: Base S₂: Antilogarithm D: Operation result

Explanations:

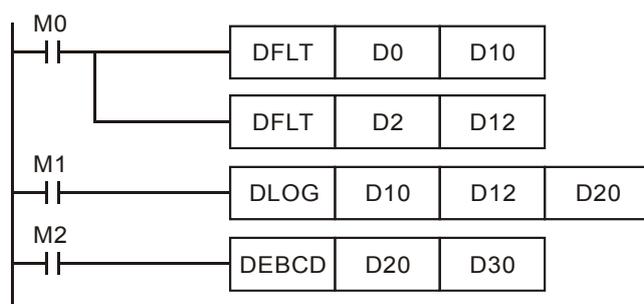
1. Perform logarithm operation with S₁ as the base and S₂ as the antilogarithm and save the result in D.
2. F refers to floating point input. Be sure to add a decimal point when using it.
3. Only 32-bit instructions DLOG and DLOGP are applicable.
4. Only a positive number is valid for S. Register D has to be 32-bit format. Operation is conducted in floating point value, so the value in S needs to be converted into floating value before logarithm operation.
5. Logarithm operation: $S_1^D = S_2, D = ? \rightarrow \log_{S_1} S_2 = D$
Example: Assume S₁ = 5, S₂ = 125, D = $\log_5^{125} = ?$ S₁^D = S₂ → 5^D = 125 → D = $\log_5^{125} = 3$
6. Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- a) If the absolute value of the result exceeds the max floating point value, carry flag will be ON.
- b) If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- c) If the conversion result is 0, zero flag will be ON.

Program Example:

1. When M0 = ON, convert (D1, D0) and (D3, D2) to binary floating value and save the result in register (D11, D10) and (D13, D12) individually.
2. When M1 = ON, perform natural logarithm operation with (D11, D10) as base and (D13, D12) as antilogarithm. The value is saved in register (D21, D20) in binary floating format.
3. When M2 = ON, convert the value in (D21, D20) into decimal floating point value and save the result in (D31, D30). (At this time, D31 indicates powers of 10 for D30).



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
	127	D	ESQR	P	S	D	Floating Point Square Root										20PM

OP	Type	Bit Devices				Word Devices										Program Steps	
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DESQR, DESQRP: 6 steps
S						*								*			
D														*			

Operands:

S: Source device **D:** Operation result

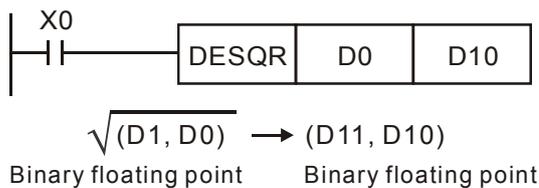
Explanations:

- This instruction performs a square root operation on the floating point value in **S** and stores the result in **D**. All data will be operated in binary floating point format and the result will also be stored in floating point format
- S** can only be a positive value. Performing any square root operation on a negative value will result in an “operation error” and instruction will not be executed. Operation error flag will be ON and error code “0E19” will be recorded.
- F refers to floating point input. Be sure to add a decimal point when using it.
- If **S** is specified with floating point value F, the instruction will operate in binary floating point.
- Only 32-bit instructions **DESQR** and **DESQRP** are applicable.
- If operation result of **D** is 0 (zero), Zero flag will be ON.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Operation error	M1793	M1953

Program Example 1:

When X0 = ON, the square root of binary floating point (D1, D0) is stored in (D11, D10) after the operation of square root.



Program Example 2:

When X2 = ON, the square root of F1234.0 (automatically converted to binary floating value) is stored in (D11, D10)



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function							Controllers	
	128	D	POW	P	S ₁	S ₂	D	Floating Point Power Operation							20PM

OP	Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S ₁					*								*			DPOW, DPOWP: 9 steps
S ₂					*								*			
D													*			

Operands:

S₁: Base S₂: Exponent D: Operation result

Explanations:

1. Perform power operation on binary floating value S₁ and S₂ and save the result in D.
POW [S₁+1, S₁]^[S₂+1, S₂] = D
2. Example of power operation:
When S₁^{S₂} = D, D = ? Assume S₁ = 5, S₂ = 3, D = 5³ = 125
3. F refers to floating point input. Be sure to add a decimal point when using it.
4. Only 32-bit instructions DPOW and DPOWP are applicable.
5. Only a positive number is valid for S₁ and both positive and negative values are valid for S₂. Register D has to be 32-bit format. Operation is conducted in floating point value, so the value in S₁ and S₂ needs to be converted into floating value before power operation
6. If the value in S₁ or S₂ is invalid, operation error flag will be ON and error code "0E19" will be recorded.
7. Flags:

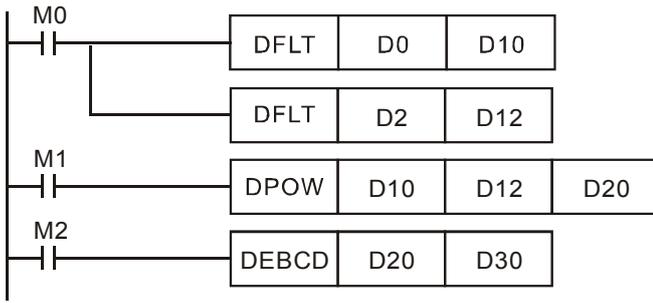
	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970
Operation error	M1793	M1953

- a) If the absolute value of the result exceeds the max floating point value, carry flag will be ON.
- b) If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- c) If the conversion result is 0, zero flag will be ON.

Program Example:

1. When M0 = ON, convert (D1, D0) and (D3, D2) to binary floating value and save the result in register (D11, D10) and (D13, D12) individually.
2. When M1 = ON, perform power operation with (D11, D10) as base and (D13, D12) as exponent. The value is saved in register (D21, D20) in binary floating format.
3. When M2 = ON, convert the value in (D21, D20) into decimal floating point value and save the result in (D31, D30). (At this time, D31 indicates powers of 10 for D30).

5 Categories and Use of Basic Application Instructions



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
	129	D	INT	P	S	D	Float to Integer										20PM

OP	Type	Bit Devices				Word Devices										Program Steps	
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DINT, DINTP: 5 steps
S													*				
D													*				

Operands:

S: Source device **D:** Operation result

Explanations:

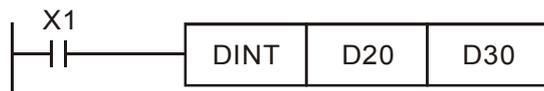
- The binary floating point value in the register **S** is converted to BIN integer and stored in register **D**. The decimal of the operation result will be left out
- This instruction is the opposite of the API 49 (FLT) instruction
- Only 32-bit instructions **DINT** and **DINTP** are applicable.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the conversion result is 0, zero flag will be ON.
- If there is any decimal left out, borrow flag will be ON.
- If the result exceeds the range (-2,147,483,648~2,147,483,647), carry flag will be ON.

Program Example:

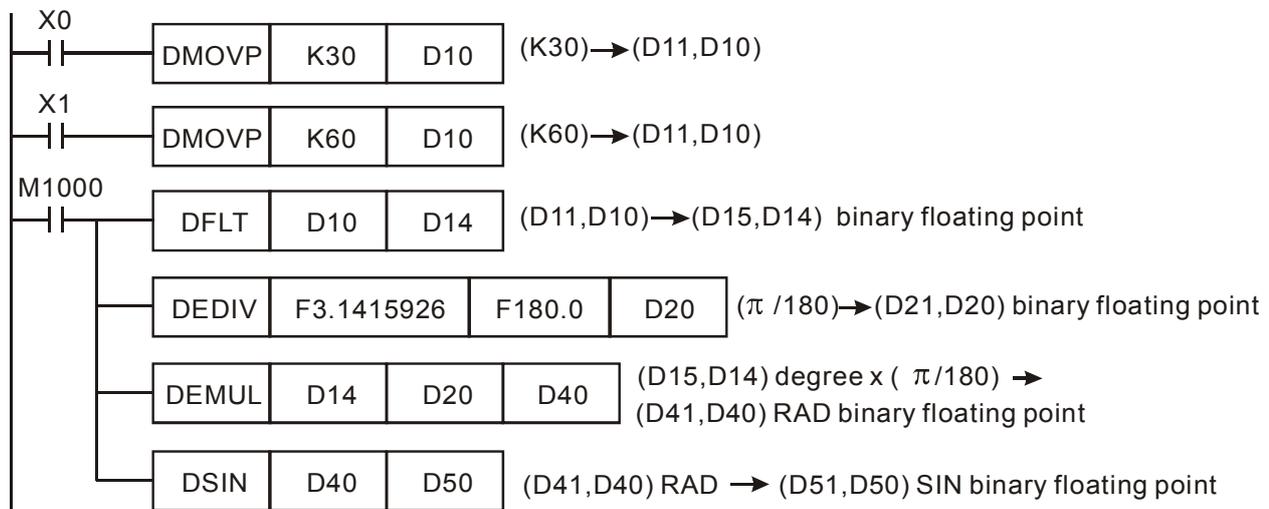
When X1 = ON, the binary floating point value of (D21, D20) will be converted to BIN integer and the result is stored in (D31, D30). The decimal of the result will be left out.



5 Categories and Use of Basic Application Instructions

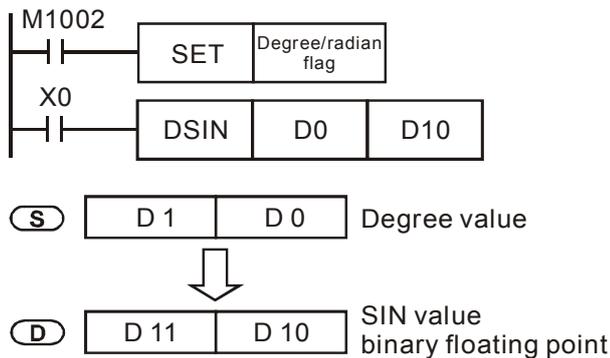
Program Example 2:

Radian/degree selection flag = OFF indicates radian mode. Select the degree value from inputs X0 and X1 and convert it to RAD value for further sine operation.



Program Example 3:

Radian/degree selection flag = ON indicates degree mode. When X0 = ON, DSIN instruction performs sine operation on the degree value ($0^\circ \leq \text{degree} < 360^\circ$) in (D1, D0) and stores the SIN value in (D11, D10) in binary floating point format



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing”.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function											Controllers	
	D	COS	P	S	D	Cosine											20PM	10PM
131				(S)	(D)													
OP	Type	Bit Devices				Word Devices											Program Steps	
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DCOS, DCOSP: 6 steps	
	S					*								*				
	D													*				

Operands:

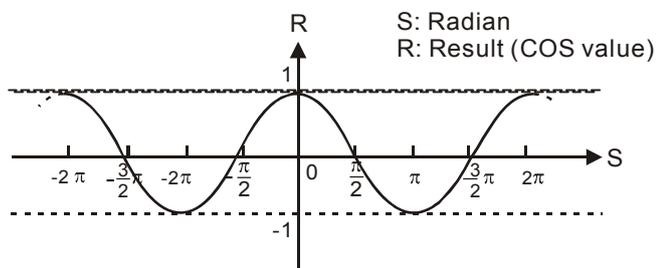
S: Source device ($0^\circ \leq S < 360^\circ$) **D:** Operation result

Explanations:

- COS instruction performs cosine operation on **S** and stores the result in **D**
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions **DCOS** and **DCOSP** are applicable.
- The value in **S** can be set as radian or degree by Radian/Degree selection flag
M1760/M1920 = OFF indicates radian mode. $RAD = degree \times \pi / 180$.
M1760/M1920 = ON indicates degree mode. Degree range: $0^\circ \leq degree < 360^\circ$
- If the result = 0, zero flag will be ON.
- Flags:

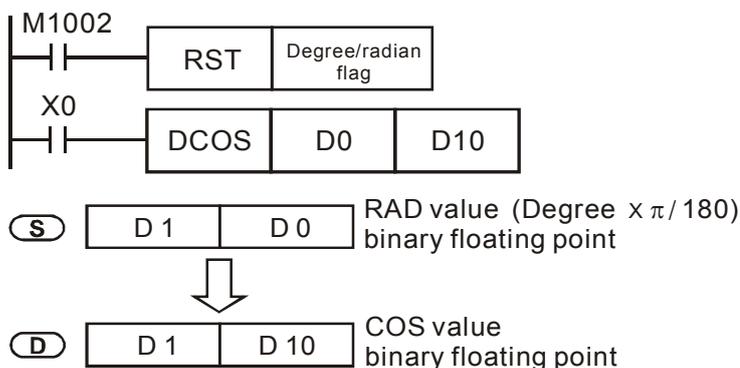
	OX	O100
Zero flag	M1808	M1968
Radian/Degree flag	M1760	M1920

- See the figure below for the relation between the radian and the operation result:



Program Example 1:

Radian/degree selection flag = OFF indicates radian mode. When X0 = ON, DCOS instruction conducts cosine operation on binary floating value in (D1, D0) and stores the COS value in (D11, D10) in binary floating format.

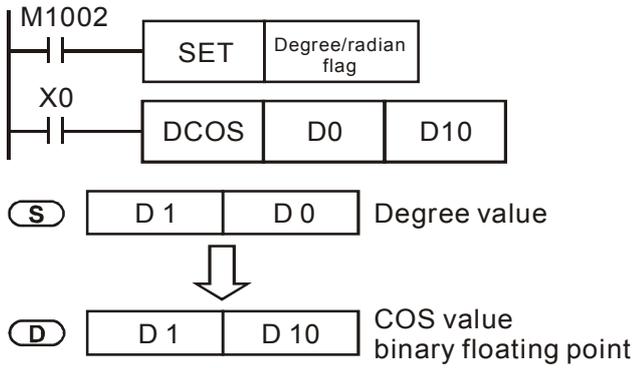


Program Example 2:

Radian/degree selection flag = ON indicates degree mode. When X0 = ON, DCOS instruction performs cosine

5 Categories and Use of Basic Application Instructions

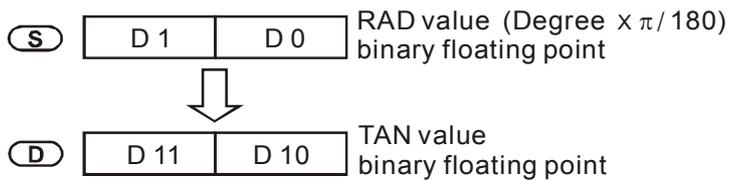
operation on the degree value ($0^\circ \leq \text{degree} < 360^\circ$) in (D1, D0) and stores the COS value in (D11, D10) in binary floating format.



Points to note:

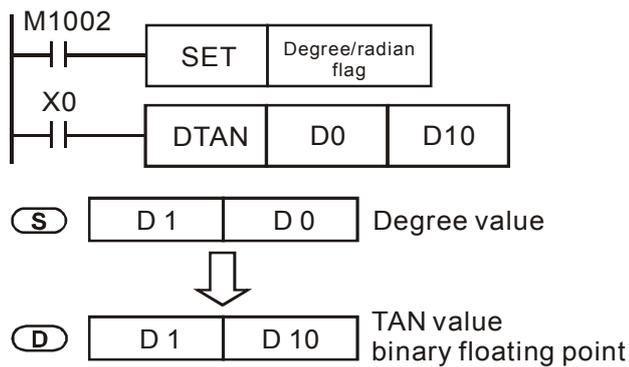
For floating point operations, see “5.3 Numeric Values for Data Processing”.

5 Categories and Use of Basic Application Instructions



Program Example 2:

Radian/degree selection flag = ON indicates degree mode. When X0 = ON, DTAN instruction performs tangent operation on the degree value ($0^\circ \leq \text{degree} < 360^\circ$) in (D1, D0) and stores the TAN value in (D11, D10) in binary floating format.



Points to note:

For floating point operations, see "5.3 Numeric Values for Data Processing".

5 Categories and Use of Basic Application Instructions

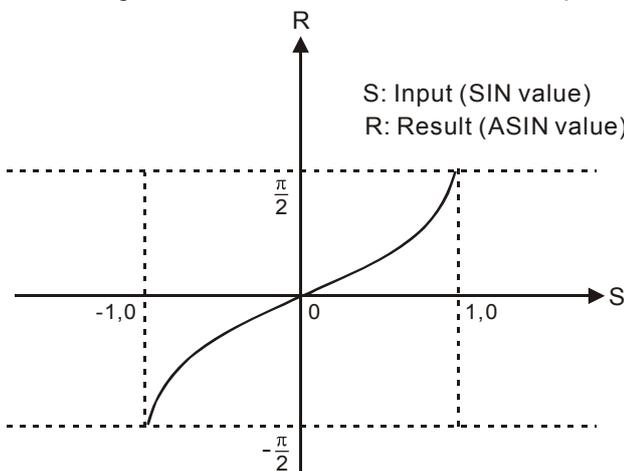
API	Mnemonic			Operands		Function										Controllers	
	133	D	ASIN	P	S	D	Arc Sine										20PM
OP	Type	Bit Devices				Word Devices										Program Steps	
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DASIN, DASINP: 6 steps
	S					*							*				
	D												*				

Operands:

S: Source device (binary floating value) **D**: Operation result

Explanations:

- ASIN value = \sin^{-1} . ASIN instruction performs arc sine operation on **S** and stores the result in **D**.
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions **DASIN** and **DASINP** are applicable.
- See the figure below for the relation between input **S** and the result:

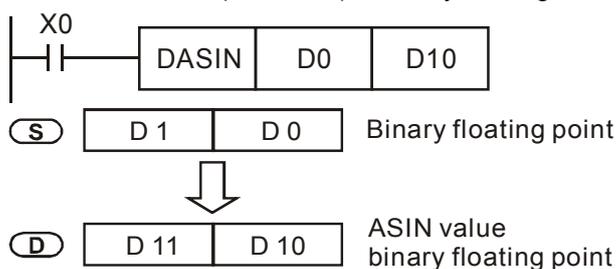


- The decimal value of the SIN value designated by **S** should be within -1.0 ~ +1.0. If the value exceeds the range, operation error flag will be ON and instruction will be disabled.
- If the result = 0, zero flag will be ON.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Operation error flag	M1793	M1953

Program Example:

When X0 = ON, DASIN instruction performs arc sine operation on the binary floating value in (D1, D0) and stores the ASIN value in (D11, D10) in binary floating format..



Points to note:

For floating point operations, see "5.3 Numeric Values for Data Processing".

5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands		Function								Controllers	
134	D	ACOS	P	(S) (D)	Arc Cosine								20PM	10PM

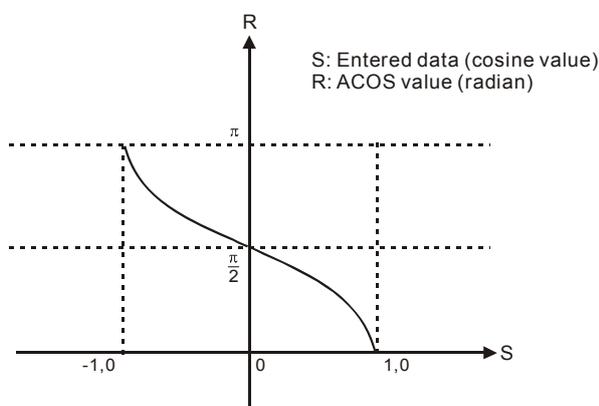
OP	Type	Bit Devices				Word Devices								Program Steps			
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DACOS, DACOSP: 6 steps
S						*								*			
D														*			

Operands:

S: Source device (binary floating value) **D:** Operation result

Explanations:

- ACOS value = COS^{-1} . ACOS instruction performs arc cosine operation on **S** and stores the result in **D**.
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions **DACOS** and **DACOSP** are applicable.
- See the figure below for the relation between the input **S** and the result:

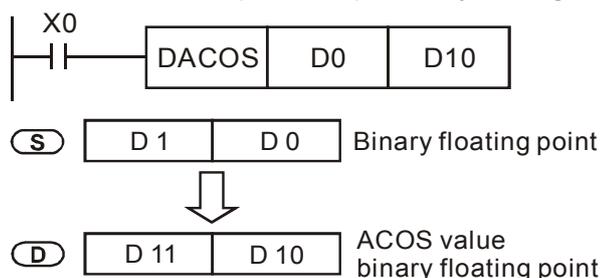


- The decimal value of the COS value designated by **S** should be within $-1.0 \sim +1.0$. If the value exceeds the range, M1067 and M1068 will be ON and instruction will be disabled.
- If operation result in **D** is 0, zero flag will be ON.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Operation error	M1793	M1953

Program Example:

When X0 = ON, DACOS instruction performs arc cosine operation on the binary floating value in (D1, D0) and stores the ACOS value in (D11, D10) in binary floating format.



Points to note:

For floating point operations, see "5.3 Numeric Values for Data Processing".

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
	135	D	ATAN	P	S	D	Arc Tangent										20PM

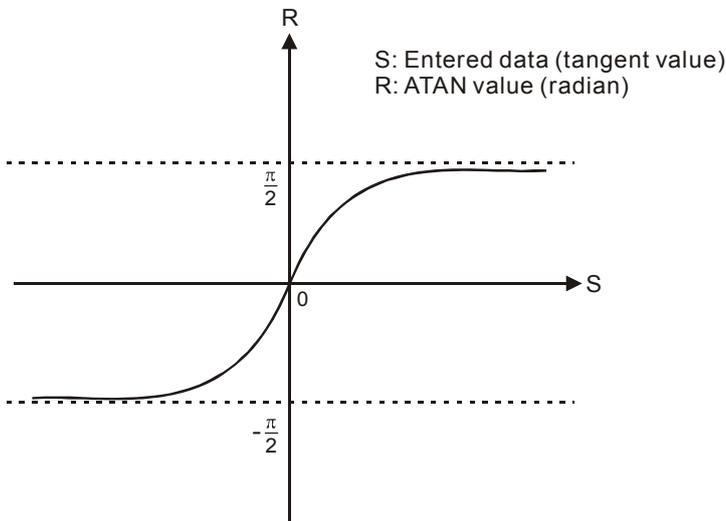
Type OP	Bit Devices				Word Devices										Program Steps	
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DATAN, DATANP: 6 steps
S					*								*			
D													*			

Operands:

S: Source device (binary floating value) **D:** Operation result

Explanations:

- ATAN value = TAN^{-1} . ATAN instruction performs arc tangent operation on **S** and stores the result in **D**
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions **DATAN** and **DATANP** are applicable.
- See the figure below for the relation between the input and the result:

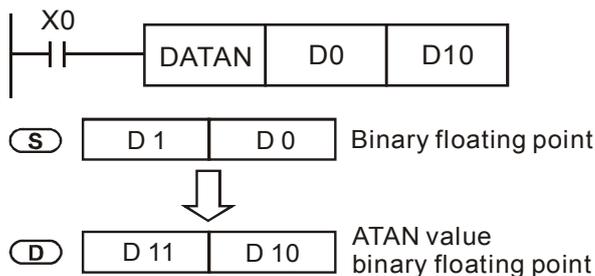


- If operation result in **D** is 0, zero flag will be ON.
- Flags:

	OX	O100
Zero flag	M1808	M1968

Program Example:

When X0 = ON, DATAN instruction performs arc tangent operation on the binary floating value in (D1, D0) and stores the ATAN value in (D11, D10) in binary floating format..



Points to note:

For floating point operations, see "5.3 Numeric Values for Data Processing".

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
	136	D	SINH	P	S	D	Hyperbolic Sine										20PM

Type OP	Bit Devices				Word Devices										Program Steps		
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DSINH, DSINH P: 6 steps	
S					*								*				
D													*				

Operands:

S: Source device (binary floating value) D: Operation result

Explanations:

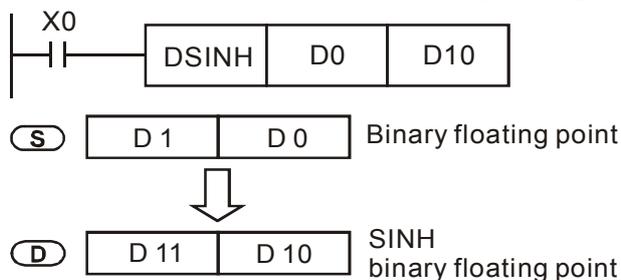
- SINH value = $(e^s - e^{-s}) / 2$. SINH instruction performs hyperbolic-sine operation on **S** and stores the result in **D**.
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions DSINH and DSINH P are applicable.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the absolute value of the result exceeds the max floating point value, carry flag will be ON.
- If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- If the conversion result is 0, zero flag will be ON.

Program Example:

When X0 = ON, DSINH instruction performs hyperbolic-sine operation on the binary floating value in (D1, D0) and stores the SINH value in (D11, D10) in binary floating format.



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing.”

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers		
	D	COSH	P	S	D	Hyperbolic Cosine										20PM	10PM	
137																		

Type OP	Bit Devices				Word Devices										Program Steps		
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DCOSH, DCOSHP: 6 steps	
S					*								*				
D													*				

Operands:

S: Source device (binary floating value) D: Operation result

Explanations:

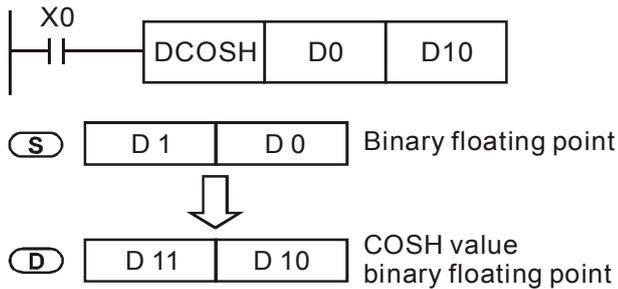
- COSH value = $(e^s + e^{-s}) / 2$. COSH instruction performs hyperbolic-cosine operation on S and stores the result in D
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions DCOSH and DCOSHP are applicable.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the absolute value of the result exceeds the max floating point value, carry flag will be ON.
- If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- If the conversion result is 0, zero flag will be ON.

Program Example:

When X0 = ON, DCOSH instruction performs hyperbolic-cosine operation on the binary floating value in (D1, D0) and stores the COSH value in (D11, D10) in binary floating format.



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing.”.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands		Function										Controllers	
138	D	TANH	P	S	D	Hyperbolic Tangent										20PM	10PM
Type OP	Bit Devices				Word Devices										Program Steps		
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	DTANH, DTANHP: 6 steps	
S					*								*				
D													*				

Operands:

S: Source device (binary floating value) D: Operation result

Explanations:

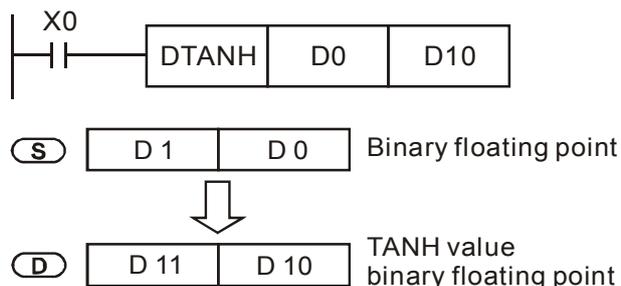
- TANH value = $(e^s - e^{-s}) / (e^s + e^{-s})$. TANH instruction performs hyperbolic-tangent operation on **S** and stores the result in **D**
- F refers to floating point input. Be sure to add a decimal point when using it.
- Only 32-bit instructions DTANH and DTANHP are applicable.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

- If the absolute value of the result exceeds the max floating point value, carry flag will be ON.
- If the absolute value of the result is less than min. floating point value, borrow flag will be ON.
- If the conversion result is 0, zero flag will be ON.

Program Example:

When X0 = ON, DTANH instruction performs hyperbolic-tangent operation on the binary floating value in (D1, D0) and stores the TANH value in (D11, D10) in binary floating format..



Points to note:

For floating point operations, see “5.3 Numeric Values for Data Processing.”

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function										Controllers	
	D	ADDR	P	(S ₁)	(S ₂)	(D)	Floating point addition										20PM	10PM
172																		
OP	Type	Bit Devices				Word devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DADDR, DADDRP: 13 steps		
S ₁													*					
S ₂													*					
D													*					

Operands:

S₁: Floating point summand S₂: Floating point addend D: Sum

Explanations:

- DADDR instruction adds the floating point summand S₁ with floating point addend S₂ and stores the operation result in D.
- In DADDR instruction, floating point values can be directly entered into S₁ and S₂.
- In DADDR instruction, floating point values (e.g. F1.2) can be either entered directly into S₁ and S₂ or stored in data registers for operation.
- When S₁ and S₂ is specified as data registers, the function of DADDR instruction is the same as API 120 EADD instruction.
- S₁ and S₂ can designate the same register. In this case, if the instruction is specified as “continuous execution instruction” (generally DADDRP instruction) and the drive contact is ON, the register will be added once in every scan.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

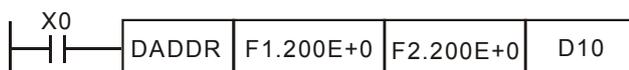
If absolute value of the result exceeds max floating point value, carry flag will be ON.

If absolute value of the result is less than min. floating point value, borrow flag will be ON.

If the conversion result is 0, zero flag will be ON

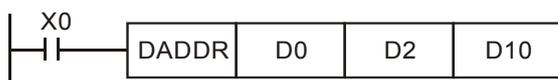
Program Example 1:

When X0 = ON, add floating point number F1.200E+0 (Input F1.2, and scientific notation F1.200E+0 will be displayed on ladder diagram. Users can set monitoring data format as float on the function View) with F2.200E+0 and store the obtained result F3.400E+0 in register D10 and D11.



Program example 2:

When X0 = ON, add floating point value (D1, D0) with (D3, D2) and store the result in (D11, D10).



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function							Controllers	
	173	D	SUBR	P	(S ₁)	(S ₂)	(D)	Floating point subtraction							20PM

OP	Type	Bit Devices				Word devices										Program Steps		
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DSUBR: 13 steps	
	S ₁													*				
	S ₂													*				
	D													*				

Operands:

S₁: Floating point minuend S₂: Floating point subtrahend D: Remainder

Explanations:

- DSUBR instruction subtracts S₁ with S₂ and stores the operation result in D.
- In DSUBR instruction, floating point values can be directly entered into S₁ and S₂.
- In DSUBR instruction, floating point values (e.g. F1.2) can be either entered directly into S₁ and S₂ or stored in data registers for operation.
- When S₁ and S₂ is specified as data registers, the function of DSUBR instruction is the same as API 121 ESUB instruction.
- S₁ and S₂ can designate the same register. In this case, if the instruction is specified as “continuous execution instruction” (generally DSUBRP instruction) and the drive contact is ON, the register will be subtracted once in every scan.
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

If absolute value of the result exceeds max floating point value, carry flag will be ON.

If absolute value of the result is less than min. floating point value, borrow flag will be ON.

If the conversion result is 0, zero flag will be ON

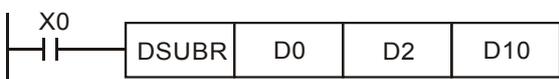
Program example 1:

When X0 = ON, subtract floating point number F1.200E+0 (Input F1.2, and scientific notation F1.200E+0 will be displayed on ladder diagram. Users can set monitoring data format as float on the function View) with F2.200E+0 and store the obtained result F-1.000E+0 in register D10 and D11.



Program example 2:

When X0 = ON, subtract the floating point value (D1, D0) with (D3, D2) and store the result in (D11, D10).



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function							Controllers	
	174	D	MULR	P	(S ₁)	(S ₂)	(D)	Floating point multiplication							20PM

Type OP	Bit Devices				Word devices										Program Steps	
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	E		F
S ₁													*			DMULR, DMULRP: 13 steps
S ₂													*			
D													*			

Operands:

S₁: Floating point multiplicand **S₂**: Floating point multiplier **D**: Product

Explanations:

- DMULR instruction multiplies **S₁** with **S₂** and stores the operation result in **D**.
- In DMULR instruction, floating point values can be directly entered into **S₁** and **S₂**.
- In DMULR instruction, floating point values (e.g. F1.2) can be either entered directly into **S₁** and **S₂** or stored in data registers for operation.
- When **S₁** and **S₂** is specified as data registers, the function of DMULR instruction is the same as API 122 EMUL instruction.
- S₁** and **S₂** can designate the same register. In this case, if the instruction is specified as “continuous execution instruction” (generally DMULRP instruction) and the drive contact is ON, the register will be multiplied once in every scan
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

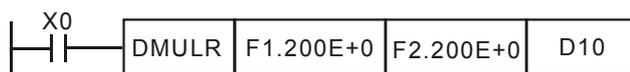
If absolute value of the result exceeds max floating point value, carry flag will be ON.

If absolute value of the result is less than min. floating point value, borrow flag will be ON.

If the conversion result is 0, zero flag will be ON

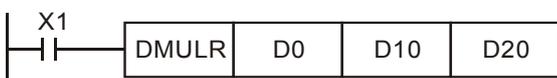
Program Example 1:

When X0 = ON, multiply floating point number F1.200E+0 (Input F1.2, and scientific notation F1.200E+0 will be displayed on ladder diagram. Users can set monitoring data format as float on the function View) with F2.200E+0 and store the obtained result F2.640E+0 in register D10 and D11.



Program example 2:

When X1= ON, multiply the floating point value (D1, D0) with (D11, D10) and store the result in (D21, D20).



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function							Controllers			
	D	DIVR	P	S ₁	S ₂	D	Floating point division							20PM	10PM		
175																	
OP	Type	Bit Devices				Word devices										Program Steps	
		X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DDIVR: 13 steps
	S ₁													*			
	S ₂													*			
	D													*			

Operands:

S₁: Floating point n dividend S₂: Floating point divisor D: Quotient

Explanations:

- DDIVR instruction divides S₁ by S₂ and stores the operation result in D
- In DDIVR instruction, floating point values can be directly entered into S₁ and S₂.
- In DDIVR instruction, floating point values (e.g. F1.2) can be either entered directly into S₁ and S₂ or stored in data registers for operation.
- When S₁ and S₂ is specified as data registers, the function of DDIVR instruction is the same as API 123 EDIV instruction.
- If S₂ = 0, operation error occurs and M1067 = ON, M1068 = ON. D1067 stores the error code 0E19 (HEX).
- Flags:

	OX	O100
Zero flag	M1808	M1968
Borrow flag	M1809	M1969
Carry flag	M1810	M1970

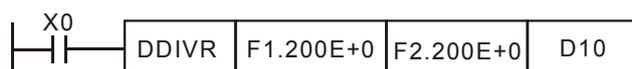
If absolute value of the result exceeds max floating point value, carry flag will be ON.

If absolute value of the result is less than min. floating point value, borrow flag will be ON.

If the conversion result is 0, zero flag will be ON.

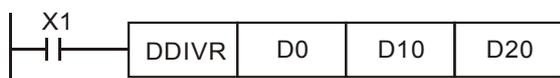
Program example 1:

When X0 = ON, divide floating point number F1.200E+0 (Input F1.2, and scientific notation F1.200E+0 will be displayed on ladder diagram. Users can set monitoring data format as float on the function View) with F2.200E+0 and store the obtained result F0.545E+0 in D10 and D11.



Program example 2:

When X1 = ON, divide the floating point number value (D1, D0) by (D11, D10) and store the obtained quotient into registers (D21, D20).



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands		Function										Controllers		
	D	LD#	S ₁	S ₂	Contact Logical Operation LD#										20PM	10PM	
215~217																	
Type OP	Bit Devices				Word Devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	LD#	DLD#
S ₁					*	*	*	*	*	*	*	*	*	*	*	5	7
S ₂					*	*	*	*	*	*	*	*	*	*	*		

Operands:

S₁: Data source device 1 S₂: Data source device 2

Explanations:

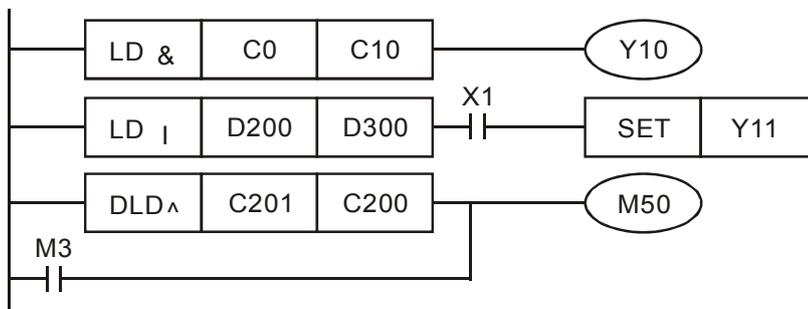
- See the specifications of DVP-PM for its range of use.
- #: &, |, ^
- LD# instruction compares the content in S₁ and S₂. If the result is not "0", the instruction will be ON. If the result is "0", instruction will be OFF.
- LD# can be connected directly with bus.

API No.	16-bit instruction	32-bit instruction	"ON" condition	"OFF" condition
215	LD&	DLD&	S ₁ & S ₂ ≠ 0	S ₁ & S ₂ = 0
216	LD	DLD	S ₁ S ₂ ≠ 0	S ₁ S ₂ = 0
217	LD^	DLD^	S ₁ ^ S ₂ ≠ 0	S ₁ ^ S ₂ = 0

- &: Logical 'AND' operation
- |: Logical 'OR' operation
- ^: Logical 'XOR' operation
- When 32-bit counters (C200 ~ C255) are used in this instruction for operation, please adopt 32-bit instruction (DLD#). If 16-bit instruction (LD#) is adopted, "program error" will occur, and the ERROR LED indicator on the panel of DVP-PM will flash.

Program Example:

- When the result of logical AND operation of C0 and C10 ≠ 0, Y10 will be ON.
- When the result of logical OR operation of D200 and D300 ≠ 0 and X1 = ON, Y11 will be ON and held.
- When the result of logical XOR operation of C201 and C200 ≠ 0 or M2 = ON, M50 will be ON.



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands		Function										Controllers			
218~220	D	AND#		(S ₁) (S ₂)	Contact Logical Operation AND#										20PM	10PM		
OP	Type	Bit Devices				Word Devices										Program Steps		
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	AND#: 5 steps	DAND#: 7 steps
	S ₁					*	*	*	*	*	*	*	*	*	*	*		
	S ₂					*	*	*	*	*	*	*	*	*	*	*		

Operands:

S₁: Data source device 1 S₂: Data source device 2

Explanations:

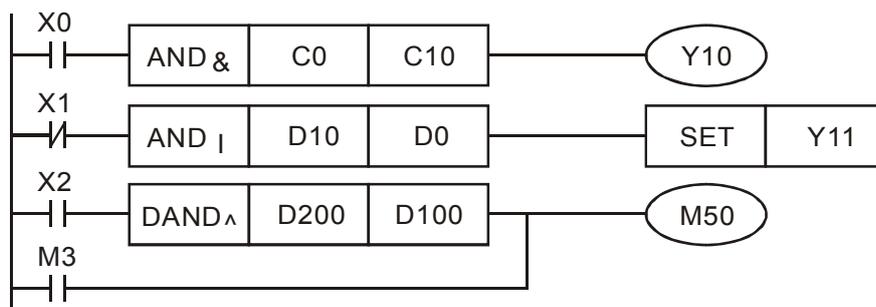
- See the specifications of DVP-PM for its range of use.
- #: &, |, ^
- AND# instruction compares the content in S₁ and S₂. If the result is not "0", the instruction will be ON. If the result is "0", instruction will be OFF.
- AND# is series connected to contacts:

API No.	16-bit instruction	32-bit instruction	"ON" condition	"OFF" condition
218	AND&	DAND&	S ₁ & S ₂ ≠ 0	S ₁ & S ₂ = 0
219	AND	DAND	S ₁ S ₂ ≠ 0	S ₁ S ₂ = 0
220	AND^	DAND^	S ₁ ^ S ₂ ≠ 0	S ₁ ^ S ₂ = 0

- &: Logical 'AND' operation
- |: Logical 'OR' operation
- ^: Logical 'XOR' operation
- When 32-bit counters (C200 ~ C255) are used in this instruction for operation, please adopt 32-bit instruction (DAND#). If 16-bit instruction (AND#) is adopted, "program error" will occur, and the ERROR LED indicator on the panel of DVP-PM will flash.

Program Example:

- When the result of logical AND operation of C0 and C10 ≠ 0, Y10 will be ON.
- When the result of logical OR operation of D10 and D0 ≠ 0 and X1 = OFF, Y11 will be ON and held.
- When X2 = ON and the result of logical XOR operation of 32-bit register D200 (D201) and 32-bit register D100 (D101) ≠ 0 or M3 = ON, M50 will be ON.



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands		Function								Controllers		
	D	OR#	S ₁	S ₂	Contact Logical operation OR#								20PM	10PM	
221~223															
Type OP	Bit Devices				Word Devices								Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S ₁					*	*	*	*	*	*	*	*	*	*	*
S ₂					*	*	*	*	*	*	*	*	*	*	*
															OR#: 5 steps
															DOR#: 7 steps

Operands:

S₁: Data source device 1 S₂: Data source device 2

Explanations:

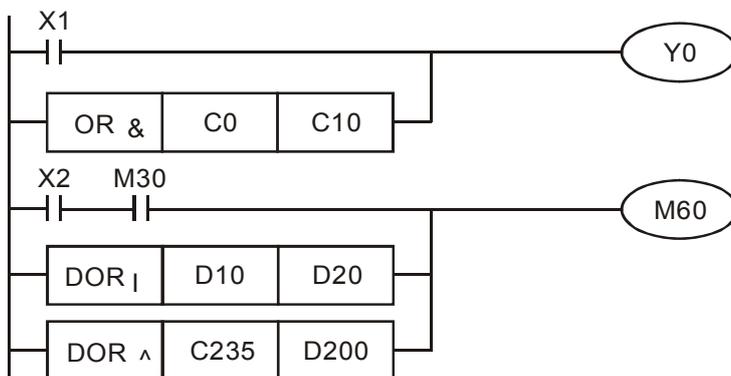
- See the specifications of DVP-PM for its range of use.
- #: &, |, ^
- OR# instruction compares the content in S₁ and S₂. If the result is not "0", the instruction will be ON. If the result is "0", instruction will be OFF.
- OR# is parallel connected to contacts:

API No.	16-bit instruction	32-bit instruction	"ON" condition	"OFF" condition
221	OR&	DOR&	S ₁ & S ₂ ≠ 0	S ₁ & S ₂ = 0
222	OR	DOR	S ₁ S ₂ ≠ 0	S ₁ S ₂ = 0
223	OR^	DOR^	S ₁ ^ S ₂ ≠ 0	S ₁ ^ S ₂ = 0

- &: Logical 'AND' operation
- |: Logical 'OR' operation
- ^: Logical 'XOR' operation
- When 32-bit counters (C200 ~ C255) are used in this instruction for operation, please adopt 32-bit instruction (DOR#). If 16-bit instruction (OR#) is adopted, "program error" will occur, and the ERROR LED indicator on the panel of DVP-PM will flash.

Program Example:

- When X1 = ON, or the result of logical AND operation of C0 and C10 ≠ 0, Y0 will be ON
- M60 will be ON when X2 = ON, M30 = ON, or the result of logical OR operation of 32-bit register D10 (D11) and 32-bit register D20 (D21) ≠ 0, or the result of logical XOR operation of 32-bit register D200 (D201) and 32-bit counter C235 ≠ 0.



5 Categories and Use of Basic Application Instructions

API	Mnemonic	Operands	Function	Controllers	
224~230	D LD※	(S ₁) (S ₂)	Contact Type Comparison	20PM	10PM

Type OP	Bit Devices				Word Devices										Program Steps		
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	LD※: 5 steps	DLD※: 7 steps
S ₁					*	*	*	*	*	*	*	*	*	*	*		
S ₂					*	*	*	*	*	*	*	*	*	*	*		

Operands:

S₁: Source device 1 S₂: Source device 2

Explanations:

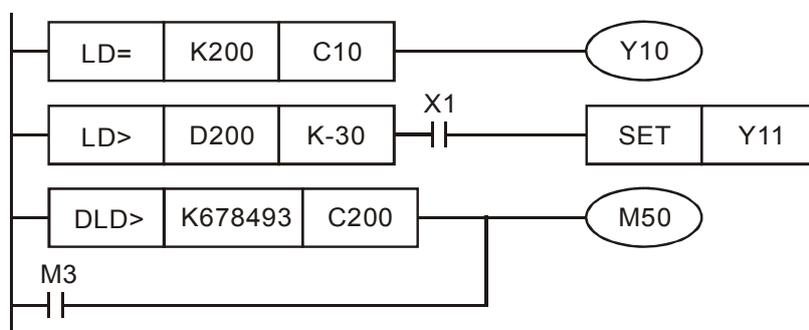
- This instruction compares the content in S₁ and S₂. Take API224 (LD=) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
- ※: =, >, <, <>, ≤, ≥
- LD※ (※: =, >, <, <>, ≤, ≥) instruction is used for direct connection with left hand bus bar

API No.	16-bit instruction	32-bit instruction	Continuity condition	Discontinuity condition
224	LD=	DLD=	S ₁ =S ₂	S ₁ ≠S ₂
225	LD>	DLD>	S ₁ >S ₂	S ₁ ≤S ₂
226	LD<	DLD<	S ₁ <S ₂	S ₁ ≥S ₂
228	LD<>	DLD<>	S ₁ ≠S ₂	S ₁ =S ₂
229	LD<=	DLD<=	S ₁ ≤S ₂	S ₁ >S ₂
230	LD>=	DLD>=	S ₁ ≥S ₂	S ₁ <S ₂

- When 32-bit counters (C200 ~ C255) are used in this instruction, make sure to adopt 32-bit instruction (DLD※). If 16-bit instruction (LD※) is adopted, a "program error" will occur and the ERROR indicator on the panel of DVP-PM will flash.

Program Example:

- When the content in C10 = K200, Y10 = ON.
- When the content in D200 > K-30 and X1 = ON, Y11 = ON and latched.
- When the content in C200 < K678,493 or M3 = ON, M50 will be ON.



5 Categories and Use of Basic Application Instructions

API	Mnemonic	Operands	Function	Controllers												
232~238	D AND※	(S ₁) (S ₂)	Serial Type Comparison	20PM	10PM											
Type OP	Bit Devices				Word Devices								Program Steps			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C		D	V	Z
S ₁					*	*	*	*	*	*	*	*	*	*	*	AND※: 5 steps
S ₂					*	*	*	*	*	*	*	*	*	*	*	DAND※: 7 steps

Operands:

S₁: Source device 1 S₂: Source device 2

Explanations:

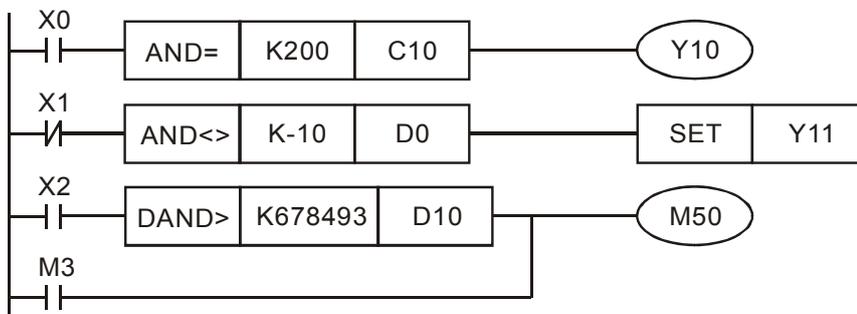
- This instruction compares the content in S₁ and S₂. Take API232 (AND =) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
- ※: =, >, <, <>, ≤, ≥
- AND※ (※: &, |, ^) instruction is used for serial connection with contacts.

API No.	16-bit instruction	32-bit instruction	Continuity condition	Discontinuity condition
232	AND =	DAND =	S ₁ = S ₂	S ₁ ≠ S ₂
233	AND >	DAND >	S ₁ > S ₂	S ₁ ≤ S ₂
234	AND <	DAND <	S ₁ < S ₂	S ₁ ≥ S ₂
236	AND <>	DAND <>	S ₁ ≠ S ₂	S ₁ = S ₂
237	AND ≤	DAND ≤	S ₁ ≤ S ₂	S ₁ > S ₂
238	AND ≥	DAND ≥	S ₁ ≥ S ₂	S ₁ < S ₂

- When 32-bit counters (C200 ~ C254) are used in this instruction, make sure to adopt 32-bit instruction (DAND※). If 16-bit instruction (AND※) is adopted, a "program error" will occur and the ERROR indicator on the panel DVP-PM will flash..

Program Example:

- When X0 = ON and the present value in C10 = K200, Y10 = ON.
- When X1 = OFF and the content in register D10 ≠ K-10, Y11 = ON and latched.
- When X2 = ON and the content in the 32-bit register D0 (D11) < 678,493 or M3 = ON, M50 will be ON.



5 Categories and Use of Basic Application Instructions

API	Mnemonic	Operands	Function	Controllers												
240~246	D OR※	(S ₁) (S ₂)	Parallel Type Comparison	20PM	10PM											
Type OP	Bit Devices				Word Devices								Program Steps			
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C		D	V	Z
S ₁					*	*	*	*	*	*	*	*	*	*	*	*
S ₂					*	*	*	*	*	*	*	*	*	*	*	*

Operands:

S₁: Source device 1 S₂: Source device 2

Explanations:

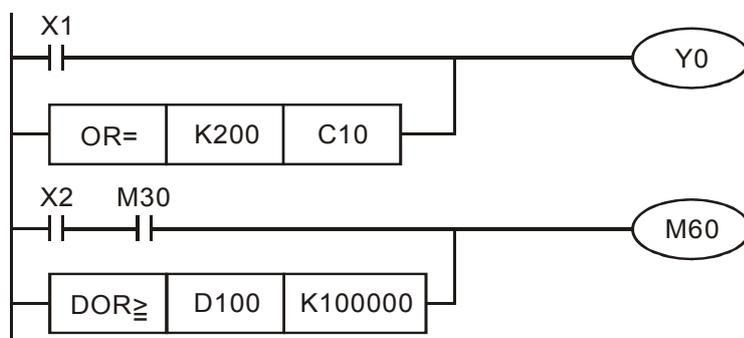
- This instruction compares the content in S₁ and S₂. Take API240 (OR =) for example, if the result is "=", the continuity of the instruction is enabled. If the result is "≠", the continuity of the instruction is disabled.
- ※: =, >, <, <>, ≤, ≥
- OR※ (※: &, |, ^) instruction is used for parallel connection with contacts.

API No.	16-bit instruction	32-bit instruction	Continuity condition	Discontinuity condition
240	OR =	DOR =	S ₁ = S ₂	S ₁ ≠ S ₂
241	OR >	DOR >	S ₁ > S ₂	S ₁ ≤ S ₂
242	OR <	DOR <	S ₁ < S ₂	S ₁ ≥ S ₂
244	OR <>	DOR <>	S ₁ ≠ S ₂	S ₁ = S ₂
245	OR ≤	DOR ≤	S ₁ ≤ S ₂	S ₁ > S ₂
246	OR ≥	DOR ≥	S ₁ ≥ S ₂	S ₁ < S ₂

- When 32-bit counters (C200 ~ C254) are used in this instruction, make sure to adopt 32-bit instruction (DOR※). If 16-bit instruction (OR※) is adopted, a "program error" will occur and the ERROR indicator on the panel of DVP-PM will flash.

Program Example:

- When X1 = ON and the present value in C10 = K200, Y0 = ON.
- M60 will be ON either when both X2 and M30 are "ON", or when the content in 32-bit register D100 (D101) ≥ K100,000.



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands	Function										Controllers		
147	D	SWAP	P	(S)	Byte swap										20PM	10PM	
OP	Type	Bit Devices				Word devices										Program Steps	
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	SWAP, SWAPP: 3 steps
S								*	*	*	*	*	*	*	*		

Operands:

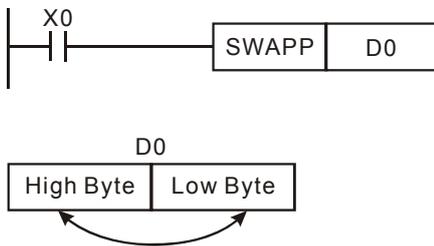
S: Device for byte swap.

Explanations:

- For 16-bit instruction, high byte and low byte of the register will be swapped.
- For 32-bit instruction, byte swap is conducted on the 2 registers separately.
- When SWAP is used as 16-bit instruction, Z device cannot be adopted; when SWAP is used as 32-bit instruction, V device cannot be adopted.
- When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal).
- This instruction adopts pulse execution instructions (SWAPP, DSWAPP)

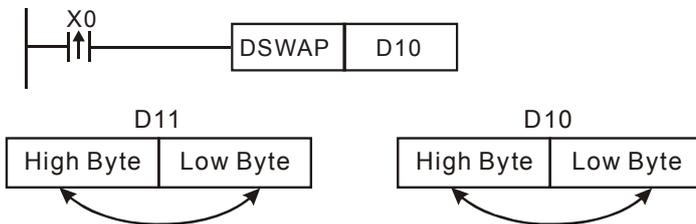
Program Example 1:

When X0 = ON, high byte and low byte of D0 will be swapped.



Program Example 2:

When X0 = ON, high byte and low byte of D11 will be swapped as well as the high byte and low byte of D10.



5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function								Controllers	
	154	D	RAND	P	S ₁	S ₂	D	Random number								20PM

OP	Type	Bit Devices				Word devices										Program Steps		
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z		
S ₁						*	*	*	*	*	*	*	*	*	*	*	RAND, RANDP: 7 steps	
S ₂						*	*	*	*	*	*	*	*	*	*	*	DRAND, DRANDP: 13 steps	
D								*	*	*	*	*	*	*	*			

Operands:

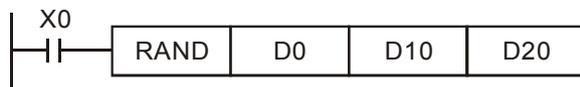
S₁: Lower bound of the random number S₂: Upper bound of the random number D: Operation result

Explanations:

1. The range of 16-bit operands S₁, S₂: $K0 \leq S_1, S_2 \leq K32,767$; the range of 32-bit operands S₁, S₂: $K0 \leq S_1, S_2 \leq K2,147,483,647$.
2. Entering S₁ > S₂ will result in operation error. The instruction will not be executed at this time, M1067, M1068 = ON and D1067 records the error code 0E1A (HEX)
3. When RAND is used as 16-bit instruction, Z device cannot be adopted; when RAND is used as 32-bit instruction, V device cannot be adopted.
4. When specifying the bit device of KnX(Y/M/S), the starting No. should be a multiple of 16 in octal or decimal format. For example, K1X0, K4Y20 (octal); K1M0, K4S16 (decimal)

Program Example:

When X10 = ON, RAND will produce the random number between the lower bound D0 and upper bound D10 and store the result in D20.



5 Categories and Use of Basic Application Instructions

API	Mnemonic	Operands	Function	Controllers	
202	SCAL P	(S ₁) (S ₂) (S ₃) (D)	Proportional calculation	20PM	10PM

OP	Type	Bit Devices				Word devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	SCAL, SCLAP: 9 steps
S ₁					*	*							*			
S ₂					*	*							*			
S ₃					*	*							*			
D													*			

Operands:

S₁: Source value S₂: Slope (unit: 0.001) S₃: Offset D: Operation result

Range of operands S₁, S₂, S₃: -32767~32767.

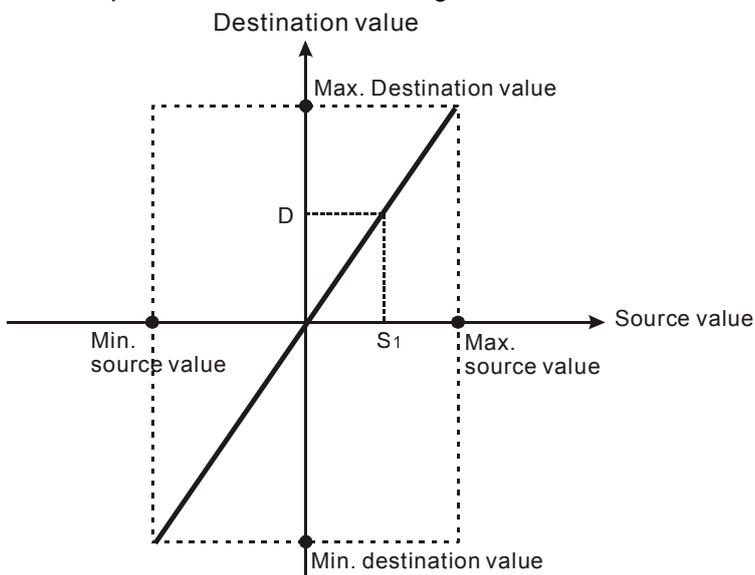
Explanations:

- SCAL instruction performs a proportional calculation according to the internal slope equation.
- Operation equation in the instruction: $D = (S_1 \times S_2) \div 1000 + S_3$
- Users have to obtain S₂ and S₃ (decimals are rounded up into 16-bit integers) by using the slope and offset equations below.

Slope equation: $S_2 = [(max. destination value - min. destination value) \div (max. source value - min. source value)] \times 1,000$

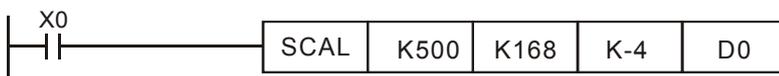
Offset equation: $S_3 = min. destination value - min. source value \times S_2 \div 1,000$

- The output curve is shown as the figure:

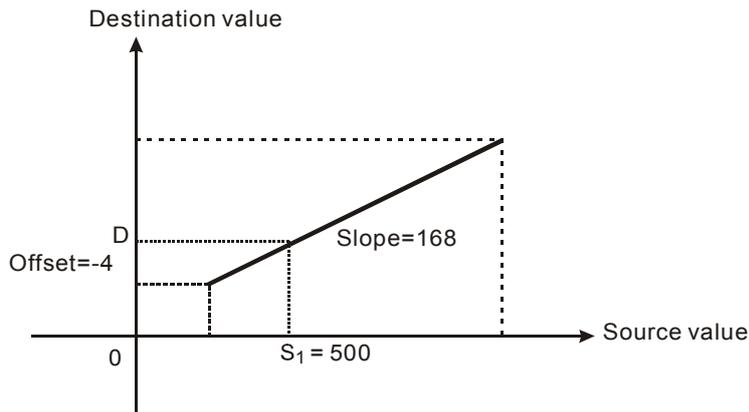


Program Example 1:

- Assume S₁ = 500, S₂ = 168 and S₃ = -4. When X0 = ON, SCAL instruction executes and the result of proportional calculation will be stored in D0.
- Equation: $D0 = (500 \times 168) \div 1000 + (-4) = 80$

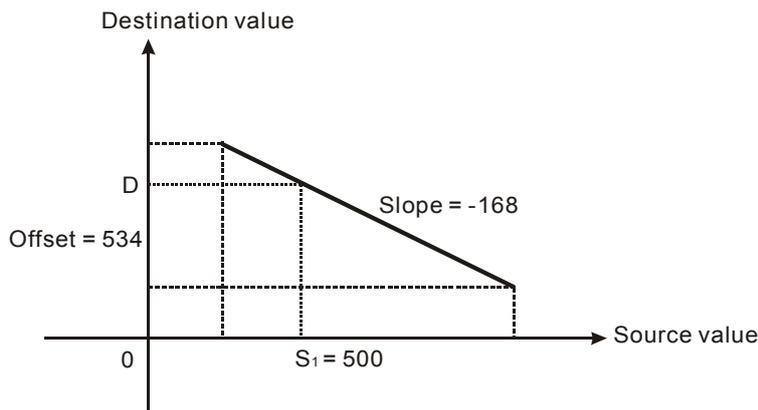
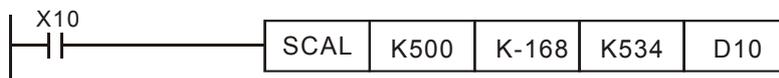


5 Categories and Use of Basic Application Instructions



Program Example 2:

1. Assume $S_1 = 500$, $S_2 = -168$ and $S_3 = 534$. When $X0 = ON$, SCAL instruction executes and the result of proportional calculation will be stored in D10..
2. Equation: $D10 = (500 \times -168) \div 1000 + 534 = 450$



Points to note:

1. This instruction is applicable for known slope and offset. If slope and offset are unknown, please use SCLP instruction for the calculation.
2. S_2 has to be within the range $-32,768 \sim 32,767$. If S_2 exceeds the applicable range, use SCLP instruction instead.
3. When adopting the slope equation, please note that the Max. source value must be larger than the min. source value. However the max. destination value does not need to be larger than the min. destination value.
4. If $D > 32,767$, D will be set as 32,767. If $D < -32,768$, D will be set as -32,768.

5 Categories and Use of Basic Application Instructions

API	Mnemonic			Operands			Function										Controllers	
203	D	SCLP	P	(S ₁)	(S ₂)	(D)	Parameter proportional value calculation										20PM	10PM

OP	Type	Bit Devices				Word devices										Program Steps	
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	SCLP, SCLPP: 9 steps DSCLP, DSCLPP: 13 steps	
S ₁					*	*							*				
S ₂					*	*							*				
D													*				

Operands:

S₁: Source value S₂: Parameters D: Operation result

Explanations:

- SCLP instruction performs a proportional calculation according to the internal slope equation as well as the parameters set in this instruction.
- Settings of S₂ for 16-bit instruction (occupies 4 consecutive devices):

Device No.	Parameter	Range
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

- Settings of S₂ for 32-bit instruction (occupies 8 consecutive devices).

Device No.	Parameter	Range	
		Integer	Floating point number
S ₂ 、S ₂ +1	Max. source value	-2,147,483,648~2,147,483,647	Range of 32-bit floating point number
S ₂ +2、3	Min. source value		
S ₂ +4、5	Max. destination value		
S ₂ +6、7	Min. destination value		

- Operation equation in the instruction: $D = [(S_1 - \text{min. source value}) \times (\text{max. destination value} - \text{min. destination value})] \div (\text{max. source value} - \text{min. source value}) + \text{min. destination value}$
- The equation to obtain the operation equation of the instruction:

$$y = kx + b$$

where

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

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

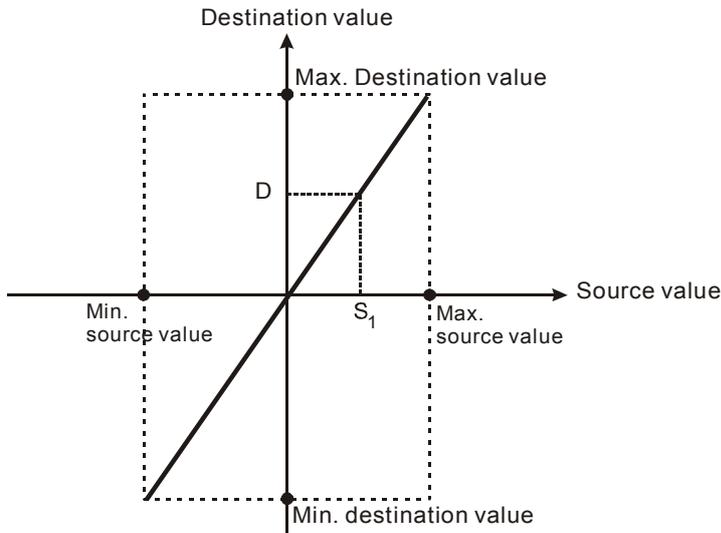
$$x = \text{Source value (S}_1\text{)}$$

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

- Substitute the above parameters into $y = kx + b$ and the operation instruction can be obtained. $y = kx + b = D = k S_1 + b = \text{slope} \times S_1 + \text{offset} = \text{slope} \times S_1 + \text{min. destination value} - \text{min. source value} \times \text{slope} = \text{slope} \times (S_1 - \text{min. source value}) + \text{min. destination value} = (S_1 - \text{min. source value}) \times (\text{max. destination value} - \text{min. destination value}) \div (\text{max. source value} - \text{min. source value}) + \text{min. destination value}$

5 Categories and Use of Basic Application Instructions

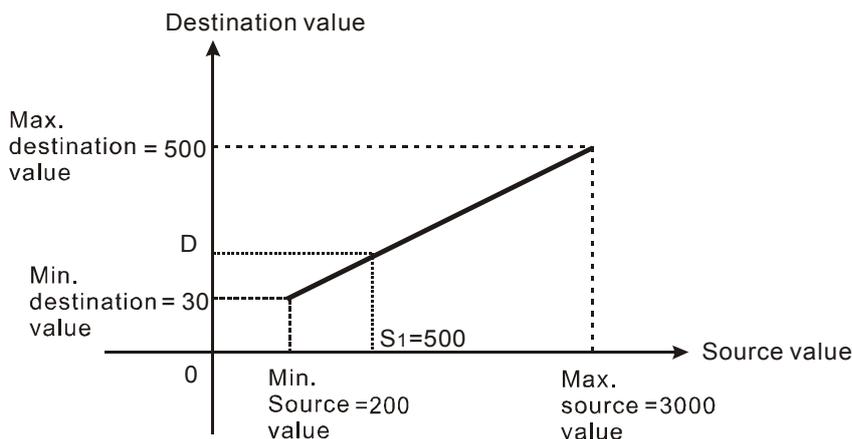
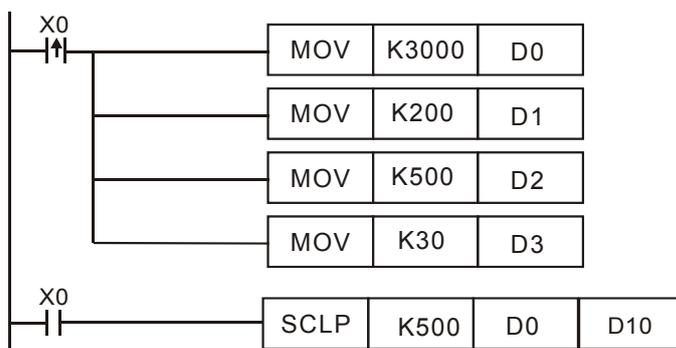
7. If $S_1 > \text{max. source value}$, S_1 will be set as max. source value. If $S_1 < \text{min. source value}$, S_1 will be set as min. source value. When the source value and parameters are set, the following output figure can be obtained:



8. Flag: M1126. Flag for switching between decimal integer and binary floating point for SCLP instruction
ON: binary floating point; OFF: decimal integer

Program Example 1:

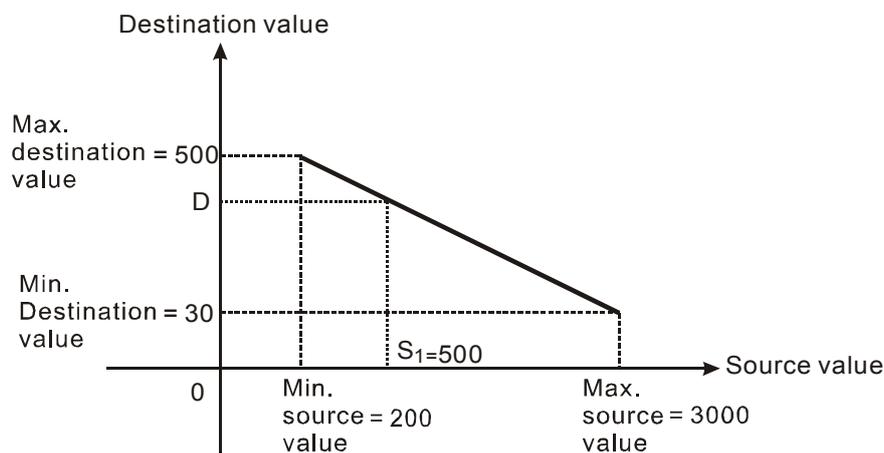
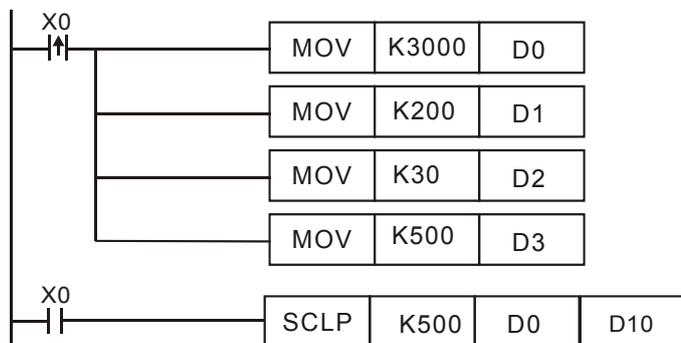
- Assume source value $S_1 = 500$, max. source value $D0 = 3000$, min. source value $D1 = 200$, max. destination value $D2 = 500$, and min. destination value $D3 = 30$. When $X0 = \text{ON}$, SCLP instruction executes and the result of proportional calculation will be stored in D10.
- Equation: $D10 = [(500 - 200) \times (500 - 30)] \div (3000 - 200) + 30 = 80.35$. Rounding off the result into an integer, $D10 = 80$.



5 Categories and Use of Basic Application Instructions

Program Example 2:

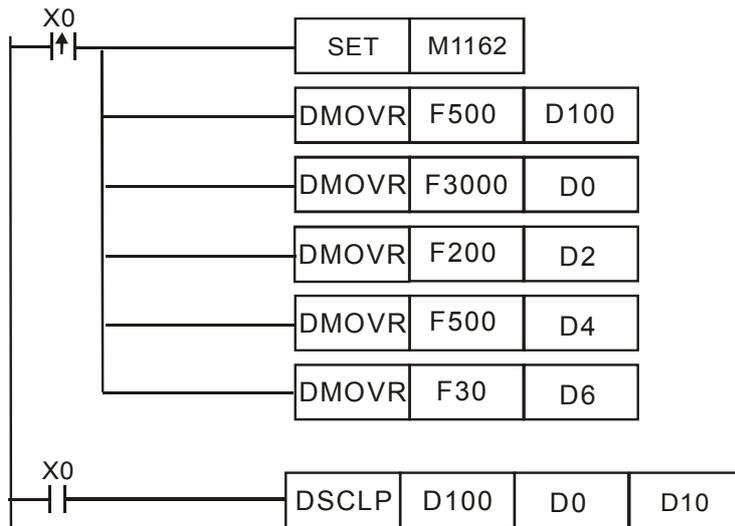
1. Assume source value $S_1 = 500$, max. source value $D0 = 3000$, min. source value $D1 = 200$, max. destination value $D2 = 30$, and min. destination value $D3 = 500$. When $X0 = ON$, SCLP instruction executes and the result of proportional calculation will be stored in $D10$.
2. Equation: $D10 = [(500 - 200) \times (30 - 500)] \div (3000 - 200) + 500 = 449.64$. Rounding off the result into an integer, $D10 = 450$.



Program Example 3:

1. Assume the source value S_1 , $D100 = F500$, max. source value $D0 = F3000$, min. source value $D2 = F200$, max. destination value $D4 = F500$, and min. destination value $D6 = F30$. When $X0 = ON$, M1162 is set up to adopt floating point operation. DSCLP instruction executes and the result of proportional calculation will be stored in $D10$.
2. Equation: $D10 = [(F500 - F200) \times (F500 - F30)] \div (F3000 - F200) + F30 = F80.35$. Round off the result into an integer, $D10 = F80$.

5 Categories and Use of Basic Application Instructions



Points to note:

1. Range of S_1 for 16-bit instruction: max. source value $\geq S_1 \geq$ min. source value; -32,768 ~ 32,767. If the value exceeds the bounds, the bound value will be used for calculation.
2. Range of integer S_1 for 32-bit instruction: max. source value $\geq S_1 \geq$ min. source value; -2,147,483,648 ~ 2,147,483,647. If the value exceeds the bounds, the bound value will be used for calculation.
3. Range of floating point S_1 for 32-bit instruction: max. source value $\geq S_1 \geq$ min. source value; adopting the range of 32-bit floating point. If the value exceeds the bounds, the bound value will be used for calculation.
4. When adopting the slope equation, please note that the Max. source value must be larger than the min. source value. However the max. destination value does not need to be larger than the min. destination value.

5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands	Function	Controllers	
	CJN	P			20PM	10PM
256			(S)	Negated Conditional Jump		
OP	Range			Program Steps		
(S)	P0 ~ P255			CJN, CJNP: 3 steps		

Operands:

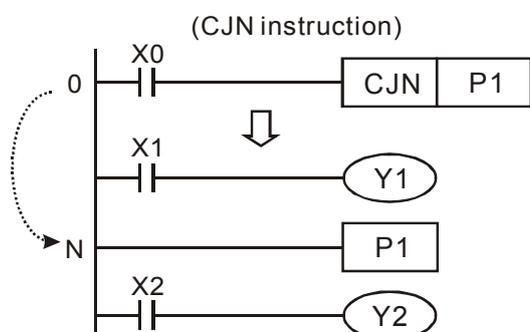
S: The destination pointer P of the negated conditional jump

Explanations:

- P cannot be modified by index registers V, Z.
- When CJN instruction is ON, the program flow will proceed with the row after the CJN instruction. When CJN instruction is OFF, program flow will jump to the address designated by pointer P.
- If you need to skip a particular part of PLC program to shorten the scan time or execute dual outputs, CJN instruction or CJN P instruction can be adopted.
- When the program designated by pointer P is prior to CJN instruction, WDT timeout will occur and PLC will stop running. Please use it carefully.
- CJN instruction can designate the same pointer P repeatedly. However, CJN and CALL cannot designate the same pointer P; otherwise operation error will occur.
- Actions of all devices while CJN is OFF and conditional jump is being executed:
 - Y, M and S remain their previous status before the conditional jump takes place.
 - The executing 10ms timer stops.
 - General counters stop executing and general application instructions are not executed.
 - If timer is reset before CJ instruction executes, the timer will still be in the reset status while CJ instruction is being executed.

Program Example 1:

- When X0 = OFF, the program will skip from address 0 to N (Pointer P1) automatically and keep on executing. Instructions between address 0 and N will be skipped.
- When X0 = ON, program flow will proceed with the row immediately after the CJN instruction.



5 Categories and Use of Basic Application Instructions

API	Mnemonic	Operands	Function	Controllers	
257	JMP	(S)	Unconditional Jump	20PM	10PM
OP	Range		Program Steps		
(S)	P0 ~ P255		JMP: 3 steps		

Operands:

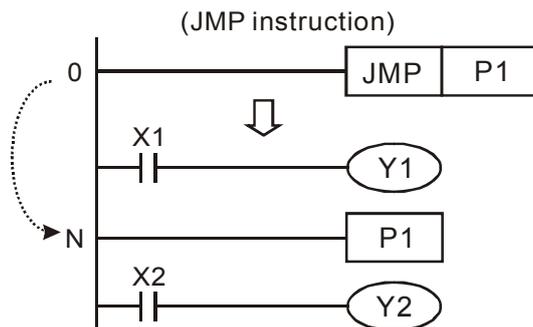
S: The destination pointer P of the conditional jump

Explanations:

1. P cannot be modified by index registers V, Z.
2. No contact to drive the instruction is required.
3. JMP instruction is similar to CJ instruction. CJ instruction requires a drive contact as the condition, but JMP instruction requires no drive contact.
4. JMP does not support pulse execution instruction (JMPP).

Program Example:

When program scan reaches address 0, the program will skip from address 0 to N (Pointer P1) automatically no matter there is a drive contact or not. Instructions between address 0 and N will be skipped.



5 Categories and Use of Basic Application Instructions

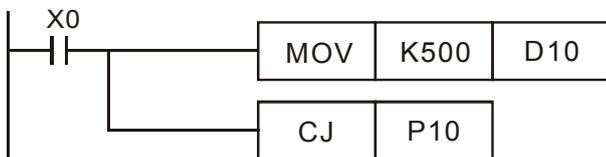
API	Mnemonic	Function	Controllers	
			20PM	10PM
258	BRET	Return to Bus Line		
OP	Descriptions		Program Steps	
N/A	N/A		BRET: 1 steps	

Explanations:

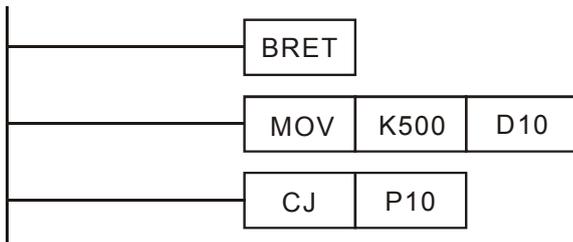
1. No contact to drive the instruction is required.
2. When BRET instruction is executed, the instructions which require drive contacts can be connected to bus line directly, i.e. instructions can be driven without drive contacts.

Program Example:

1. In general programs, the instructions will be executed only when drive contact X0 = ON.



2. When BRET instruction is added into the program, the instructions will be executed directly without drive contacts.



5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands	Function	Controllers	
259	MMOV	P	(S) (D)	16-bit→32-bit Conversion	20PM	10PM

OP	Type	Bit Devices				Word Devices										Program Steps
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	
S					*	*	*	*	*	*	*	*	*	*	*	
D							*	*	*	*	*	*	*	*	*	

MMOV, MMOVP: 5 steps

Operands:

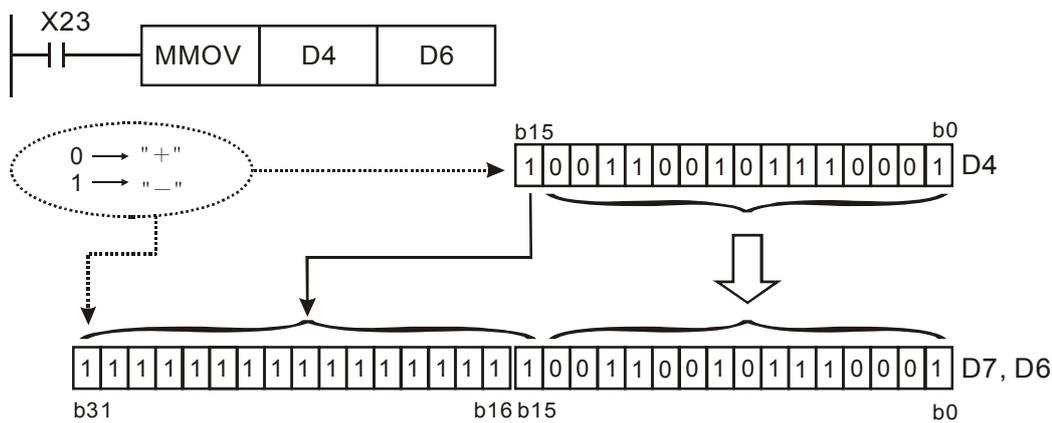
S: Source device (16-bit) **D:** Destination device (32-bit)

Explanations:

MMOV instruction sends the data in 16-bit device **S** to 32-bit device **D**. Sign bit (MSB) of source device will be copied to every bit in the high byte of **D**.

Program Example:

When X23 = 0N, 16-bit data in D4 will be sent to D6 and D7.



In the example above, b15 in D4 will be sent to b15~b31 of D7/D6, therefore all bits from b15~b31 will be “negative.”

5 Categories and Use of Basic Application Instructions

API	Mnemonic		Operands		Function										Controllers			
260	RMOV	P	(S)	(D)	32-bit → 16-bit Conversion										20PM	10PM		
OP	Type	Bit Devices				Word Devices										Program Steps		
		X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	RMOV, RMOVP: 5 steps	
	S					*	*	*	*	*	*	*	*	*	*	*		
	D							*	*	*	*	*	*	*	*	*		

Operands:

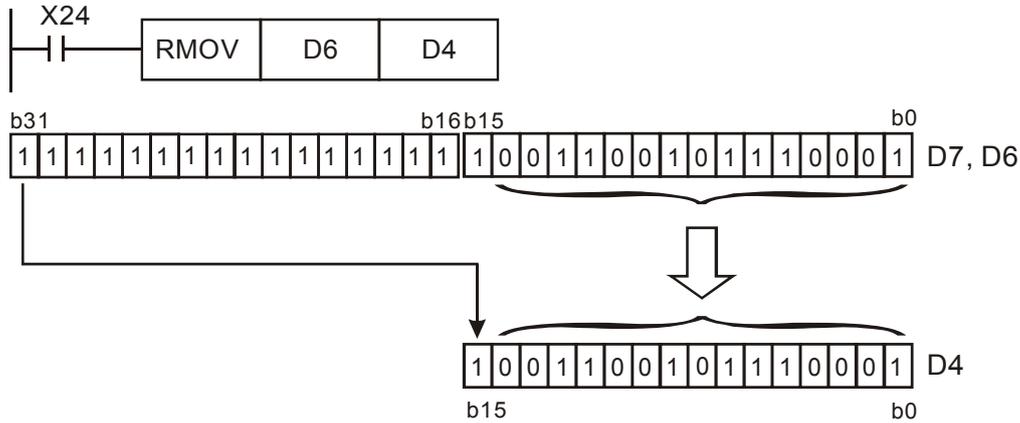
S: Source device (32-bit) **D:** Destination device (16-bit)

Explanations:

RMOV instruction sends the data in 32-bit device **S** to 16-bit device **D**. Sign bit (MSB) of source device will be copied to MSB of destination device.

Program Example:

When X24 = ON, the data in D6 and D7 will be sent to D4.



When X24 = ON, b31 (MSB) of D7 (**S**) will be sent to b15 (MSB) of D4 (**D**). In addition, b0~b14 of D6 will also be sent to b0~b14 of D4 and b15 ~ b30 will be ignored.

6 Motion Instructions and G-Code Instructions

6.1 List of Motion Instructions and G-Code Instructions

Motion Instructions

MON	Mnemonic	Function	Model			Response Time	Page
			20D	20M	10M		
00	DRV	2-Axis High-Speed Positioning	✓	✓		20 ~ 25ms	6-6
		3-Axis High-Speed Positioning		✓		20 ~ 25ms	6-6
01	LIN	2-Axis Synchronous Linear Interpolation (considering remaining distance)	✓	✓		20 ~ 22ms	6-9
		3-Axis Synchronous Linear Interpolation (considering remaining distance)		✓		20 ~ 25ms	6-9
02	CW	Clockwise Arc Movement (set distance to center of arc, considering remaining distance)	✓	✓		20 ~ 24ms	6-12
		Clockwise Helical Movement (set distance to center of arc, considering remaining distance)		✓		20 ~ 25ms	6-12
03	CCW	Counterclockwise Arc Movement (set distance to center of arc, considering remaining distance)	✓	✓		20 ~ 24ms	6-12
		Counterclockwise Helical Movement (set distance to center of arc, considering remaining distance)		✓		20 ~ 25ms	6-12
04	CW	Clockwise Arc Movement (set radius, considering remaining distance)	✓	✓		20 ~ 24ms	6-18
		Clockwise Helical Movement (set radius, considering remaining distance)		✓		20 ~ 25ms	6-18
05	CCW	Counterclockwise Arc Movement (set radius, considering remaining distance)	✓	✓		20 ~ 24ms	6-18
		Counterclockwise Helical Movement (set radius, considering remaining distance)		✓		20 ~ 25ms	6-18
06	TIM	Pause Time	✓	✓		-	6-22
07	DRVZ	Return to Mechanical Zero Point (zero return)	✓	✓		20 ~ 25ms	6-23
08	SETR	Set up Electrical Zero Point	✓	✓		-	6-26
09	DRVR	Return to Electrical Zero Point	✓	✓		20 ~ 25ms	6-27
10	INTR	2-axis Single-Speed Linear Interpolation (ignoring remaining distance)	✓	✓		20 ~ 25ms	6-28
11	SINTR	1-axis Single-speed Positioning with Additional Distance	✓	✓		20 ~ 25ms	6-29
12	DINTR	1-axis Double-speed Positioning with Additional Distance	✓	✓		20 ~ 25ms	6-31
13	MOVC	Set up Offset of Linear Movement	✓	✓		-	6-33
14	CNTC	Set up Offset for Distance to Center of Arc	✓	✓		-	6-34
15	RADC	Set up Offset of Arc Radius	✓	✓		-	6-35
16	CANC	Cancel Compensation	✓	✓		-	6-36
17	ABST	Set up Absolute Coordinate	✓	✓		-	6-37
18	INCT	Set up Relative Coordinate	✓	✓		-	6-37
19	SETT	Set up Current Position	✓	✓		-	6-38

6 Motion Instructions & G-Code Instructions

O Pointer / M Pointer (M code)

Pointer	Function	Explanation	Model			Page
			20D	20D	10M	
O	Main program / Motion Subroutine pointer	Main Program:O100, Motion subroutine: OX0 ~ OX99	✓	✓		6-40
M	M pointer / M code	M0 ~ M65535 M102: End of O100 main program M2: end of OX0 ~ OX99 motion subroutine	✓	✓		6-41

G-Code Instructions

G-Code	Mnemonic	Function	Model			Page
			20D	20M	10M	
00	DRV	2-Axis High-Speed Positioning	✓	✓		6-47
00	DRV	3-Axis High-Speed Positioning		✓		6-44
01	LIN	2-Axis Synchronous Linear Interpolation (considering remaining distance)	✓	✓		6-50
01	LIN	3-Axis Synchronous Linear Interpolation (considering remaining distance)		✓		6-48
02	CW	Clockwise Arc Movement (set distance to center of arc, considering remaining distance)	✓	✓		6-51
02	CW	Clockwise Helical Movement (set distance to center of arc, considering remaining distance)		✓		6-51
03	CCW	Counterclockwise Arc Movement (set distance to center of arc, considering remaining distance)	✓	✓		6-51
03	CCW	Counterclockwise Helical Movement (set distance to center of arc, considering remaining distance)		✓		6-51
02	CW	Clockwise Arc Movement (set radius, considering remaining distance)	✓	✓		6-52
02	CW	Clockwise Helical Movement (set radius, considering remaining distance)		✓		6-52
03	CCW	Counterclockwise Arc Movement (set radius, considering remaining distance)	✓	✓		6-52
03	CCW	Counterclockwise Helical Movement (set radius, considering remaining distance)		✓		6-52
04	TIM	Pause Time	✓	✓		6-53
17	NULL	Set up X-Y plane		✓		6-54
18	NULL	Set up X-Z plane		✓		6-54
19	NULL	Set up Y-Z plane		✓		6-54
90	ABST	Set up Absolute Coordinate	✓	✓		6-54
91	INCT	Set up Relative Coordinate	✓	✓		6-54

6 Motion Instructions and G-Code Instructions

6.2 Composition of Motion Instructions and G-Code Instructions

6.2.1 Motion Instructions

- ◆ A motion instruction has two parts: the mnemonic and the operand

Mnemonic		Function of the instruction
Operand	Function index	Indexes X, Y, Z, F, FX, FY, FZ, R, I, J, K. indicate axis, frequency, radius, and coordinates.
	Parameter value	Parameter value for each function index

- “Function index” of the operand must not be ignored.
- “Parameter value” of the operand can be specified as follows:
 1. Enter arabic numerals and 32-bit register (DD). The operands will all be operated in 32-bit format.
For example: DRV X1000 FX1000 Y1000 FYDD1000
 2. Enter single K, H, D with arabic numerals. The operands will be operated in 16-bit format.
For example: DRV XK1000 FXH1000 YK1000 FYD1000
 3. Enter single K, H, D or double K, H, D with arabic numerals. The operands can be operated in both 16-bit and 32-bit format according to the specified value index (K, H, or D).
For example: DRV X1000 FXHH1000 YK1000 FY1000

- ◆ Program steps occupied by motion instruction

- Mnemonic in 20D model: 1 step;
Mnemonic in 20M model: DRV, LIN, CW, CCW and SETT occupy 2 steps, others 1 step..
- Operands with only arabic numerals: 3 steps per operand
- Operands with K, H, D, DD and arabic numerals: 2 steps per operand
- Operands with KK, HH and arabic numerals: 3 steps per operand

- ◆ Format of a motion instruction:

MON	Mnemonic	Operands						Function
00	DRV	X(P ₁)	FX(V ₁)	Y(P ₂)	FY(V ₂)			High-Speed Positioning
Type	Bit Devices			Double-Word Devices			Notes	
OP	K	H	D	KK	HH	DD	<ul style="list-style-type: none"> ▪ RADC instruction supports V, Z index register modification on the devices. ▪ See specifications of DVP-PM for the range of use. ▪ You can place an M-Code instruction after RADC. 	
P ₁	*	*	*	*	*	*		
V ₁	*	*	*	*	*	*		
P ₂	*	*	*	*	*	*		
V ₂	*	*	*	*	*	*		

- ① MON (motion) instruction No.
- ② Mnemonic
- ③ Function index of operands: X, Y, Z, F, FX, FY, FZ, R, I, J, K. Indicate axis, frequency, radius, and coordinates.
- ④ Parameter value of operands
- ⑤ Function of the MON instruction
- ⑥ Device type
- ⑦ Device name

6 Motion Instructions & G-Code Instructions

- ⑧ Parameter columns marked with * indicate the applicable device type for the parameter.
- ⑨ Parameter columns marked with * and in grey refers to V, Z index modification is applicable.
- ⑩ Notes for the instruction

- Points to note:

Some motion instructions are only composed of the instruction part (mnemonic), e.g. DRVZ, SETR, or ABS. However, most motion instructions are composed of mnemonic with operands. In addition, no drive contact is required to be placed before a motion instruction.

6.2.2 G-Code Instructions

◆ A G-Code instruction also has two parts: the mnemonic and the operand

Mnemonic		Function of the instruction
Operand	Function index	Indexes X, Y, Z, I, J, K, R, F indicate axis, frequency, radius, and coordinates.
	Parameter value	Parameter value for each function index

- “Function index” of the operand must not be ignored.
- “Parameter value” of the operand can be specified by entering arabic numerals or decimals. All operands will all be operated in 32-bit format..

For example: G00 X100 Y100 or G00 X100.0 Y100.0

- For 20M model, 16-bit register (D) and 32-bit register (DD) are also applicable for operands, for example:

G0 XD11 YDD20 FDD25;

XDD30 YD40;

G1 X100.0 Y25.0 FD50;

G02 XD60 Y50.0 ID100 JDD80;

G03 YDD90 RD70 F300.0;

- Operand with decimal is 1,000 times larger than it being without decimal point.

For example: G00 X100.0 Y100.0 = G00 X100000 Y100000

◆ Program steps occupied by G-Code instruction

- Mnemonic in 20D model: 1 step;
- Mnemonic in 20M model: G00, G01, G02 and G03 occupy 2 steps, others 1 step..
- Operands with arabic numerals: 3 steps per operand

◆ Format of a G-Code instruction:

G-Code	① ↓	② ↓	③ ↓	④ ↓
	Mnemonic	Operands		Function
	G0	X(P ₁) Y(P ₂) Z(P ₃)		High-Speed Positioning

- ① G-Code instruction No.
- ② Function index of operands: X, Y, Z, F, R, I, J, K. Indicate axis, frequency, radius, and coordinates.
- ③ Parameter value of operands
- ④ Function of the G-Code instruction

- Points to note:

6 Motion Instructions & G-Code Instructions

6.3 Motion Instructions

MON	Mnemonic	Operands	Function	Controllers	
00	DRV	X (P ₁) FX (V ₁) Y (P ₂) FY (V ₂) Z (P ₃) FZ (V ₃)	High-Speed Positioning	20PM	-

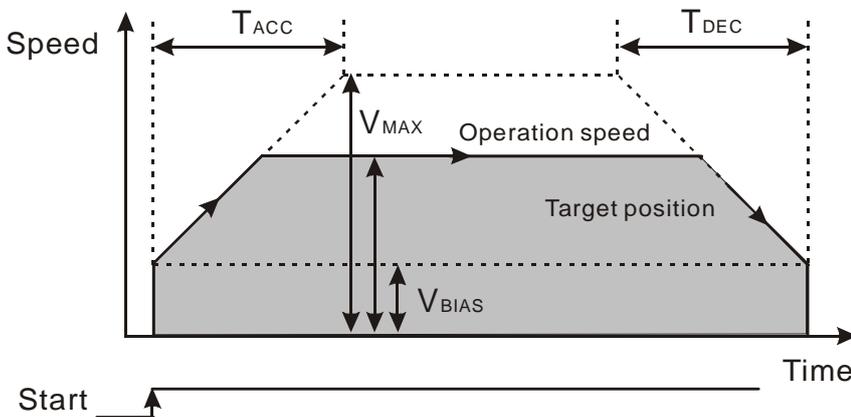
Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
P ₁	*	*	*	*	*	*	<ul style="list-style-type: none"> DRV instruction supports V, Z index register modification on the devices. See specifications of DVP-PM for the applicable range of each device. You can place an M-Code instruction after DRV.
V ₁	*	*	*	*	*	*	
P ₂	*	*	*	*	*	*	
V ₂	*	*	*	*	*	*	
P ₃	*	*	*	*	*	*	
V ₃	*	*	*	*	*	*	

Operands:

P₁: Target position on X axis **V₁:** Moving speed on X axis **P₂:** Target position on Y axis
V₂: Moving speed on Y axis **P₃:** Target position on Z axis **V₃:** Moving speed on Z axis

Explanations:

- Maximum **V₁**, **V₂**, **V₃** = **V_{MAX}**
- Range of parameters: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
- Acceleration/deceleration time and bias speed can be set up in special D.
- Acceleration/deceleration time increases or decreases in proportional to the setting of **V_{MAX}**.
- Timing diagram:



- 16-bit devices and 32-bit devices can be used together.
- If you set up the moving speed on an axis, you have to set up the target position on the axis. However, if you set up the target position, it is not a must to set up the moving speed.
- If you set up the target position on the axis without setting up the moving speed, the operation will run at **V_{MAX}**.
- 20D model supports 2-axis (X/Y) high-speed positioning, and therefore there are 8 operand combinations for DRV instruction.

NO.	Instruction	Operand combination
1	DRV	X (P ₁)
2		X (P ₁) FX (V ₁)
3		Y (P ₂)

6 Motion Instructions and G-Code Instructions

4		Y (P ₂) FY (V ₂)
5		X (P ₁) Y (P ₂)
6		X (P ₁) Y (P ₂) FY (V ₂)
7		X (P ₁) FX (V ₁) Y (P ₂)
8		X (P ₁) FX (V ₁) Y (P ₂) FY (V ₂)

10. 20M model supports 3-axis (X, Y or Z) high-speed positioning, and therefore there are 26 operand combinations for DRV instruction.

NO.	Instruction	Operand combination
1	DRV	X (P ₁)
2		X (P ₁) FX (V ₁)
3		Y (P ₂)
4		Y (P ₂) FY (V ₂)
5		X (P ₁) Y (P ₂)
6		X (P ₁) Y (P ₂) FY (V ₂)
7		X (P ₁) FX (V ₁) Y (P ₂)
8		X (P ₁) FX (V ₁) Y (P ₂) FY (V ₂)
9		Z (P ₃)
10		X (P ₁) Z (P ₃)
11		X (P ₁) FX (V ₁) Z (P ₃)
12		Y (P ₂) Z (P ₃)
13		Y (P ₂) FY (V ₂) Z (P ₃)
14		X (P ₁) Y (P ₂) Z (P ₃)
15		X (P ₁) Y (P ₂) FY (V ₂) Z (P ₃)
16		X (P ₁) FX (V ₁) Y (P ₂) Z (P ₃)
17		X (P ₁) FX (V ₁) Y (P ₂) FY (V ₂) Z (P ₃)
18		Z (P ₃) FZ (V ₃)
19		X (P ₁) Z (P ₃) FZ (V ₃)
20		X (P ₁) FX (V ₁) Z (P ₃) FZ (V ₃)
21		Y (P ₂) Z (P ₃) FZ (V ₃)
22		Y (P ₂) FY (V ₂) Z (P ₃) FZ (V ₃)
23		X (P ₁) Y (P ₂) Z (P ₃) FZ (V ₃)
24		X (P ₁) Y (P ₂) FY (V ₂) Z (P ₃) FZ (V ₃)
25		X (P ₁) FX (V ₁) Y (P ₂) Z (P ₃) FZ (V ₃)
26		X (P ₁) FX (V ₁) Y (P ₂) FY (V ₂) Z (P ₃) FZ (V ₃)

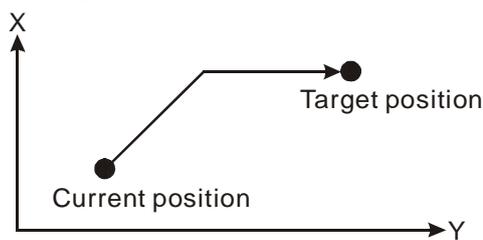
Program Example:

1. DRV XK12345 YH7567 FYKK40000

The instruction performs high speed positioning on 2 axes to the target position (K12345, H7567). The target position can be an absolute coordinate or relative coordinate, which is determined by ABST / INCT (G90/G91) instruction above and closest to DRV. The moving speed on X axis is not set (i.e. output by V_{MAX}), and Y axis outputs at 40K pulses per second.

6 Motion Instructions & G-Code Instructions

2. Moving path



3. Other valid combinations of operands:

DRV_XKK-345289_FXD100_YDD10Z5_FYDD102

DRV_XDD20_FXHH2345_YK456@V4_FYDD0

The above instructions are legal, and device D is the register storing the value to be set up.

Remarks:

Relevant special registers:

D1822, D1823	Maximum speed on X axis. D1822 (low word); D1823 (high word).
D1824, D1825	Bias speed of X axis. D1824 (low word); D1825 (high word).
D1836	Acceleration time of X axis
D1837	Deceleration time of X axis
D1902, D1903	Maximum speed on Y axis. D1902 (low word); D1903 (high word).
D1904, D1905	Bias speed of Y axis. D1904 (low word); D1905 (high word).
D1916	Acceleration time of Y axis
D1917	Deceleration time of Y axis
D1982, D1983	Maximum speed on Z axis. D1982 (low word); D1983 (high word)
D1984, D1985	Bias speed of Z axis. D1984 (low word); D1985 (high word)
D1996	Acceleration time of Z axis
D1997	Deceleration time of Z axis

6 Motion Instructions and G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
01	LIN	X (P ₁) Y (P ₂) Z (P ₃) F (V)	Linear Interpolation (considering remaining distance)	20PM	-

Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
P ₁	*	*	*	*	*	*	<ul style="list-style-type: none"> LIN instruction supports V, Z index register modification on the devices. See specifications of DVP-PM for the applicable range of each device. You can place an M-Code instruction after LIN.
P ₂	*	*	*	*	*	*	
P ₃	*	*	*	*	*	*	
F	*	*	*	*	*	*	

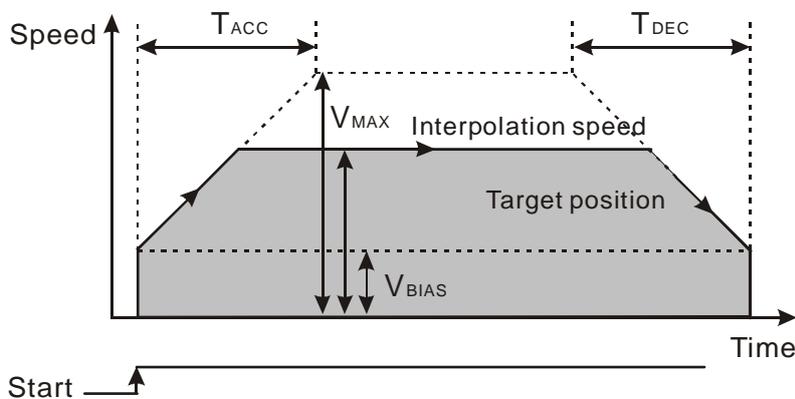
Operands:

P₁: Target position on X axis P₂: Target position on Y axis P₃: Target position on Z axis

V: Speed for linear interpolation

Explanations:

- Maximum $V = V_{MAX}$.
- Range of parameters: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
- Acceleration/deceleration time and bias speed can be set up in special D.
- Acceleration/deceleration time increases or decreases in proportional to the setting of V_{MAX} .
- Timing diagram:



- The interpolation speed can be monitored by special registers: D1850 ~ D1851 for X axis; D1930 ~ D1931 for Y axis; D2010 ~ D2011 for Z axis.
- D1865 is for setting up stop mode with the consideration on the remaining distance (see Remarks for more information).
- 16-bit devices and 32-bit devices can be used together.
- Target position is required, but moving speed could be left out.
- If you set up the target position on the axis without setting up the moving speed, the operation will run at V_{MAX} .
- 20D model only supports 2-axis (X/Y) synchronous interpolation positioning, and therefore there are 6 operand combinations for LIN instruction.

NO.	Instruction	Operand combination
1	LIN	X (P ₁)
2		X (P ₁) F (V)

6 Motion Instructions & G-Code Instructions

3		Y (P ₂)
4		Y (P ₂) F (V)
5		X (P ₁) Y (P ₂)
6		X (P ₁) Y (P ₂) F (V)

12. 20M model support 3-axis (X, Y or Z) high-speed positioning, and therefore there are 14 operand combinations for LIN instruction.

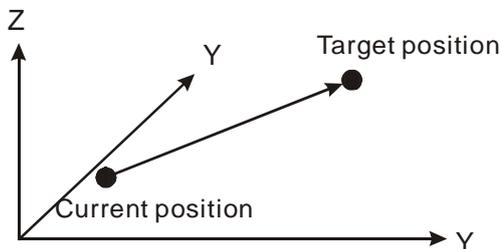
NO.	Instruction	Operand combination
1	LIN	X (P ₁)
2		X (P ₁) F (V)
3		Y (P ₂)
4		Y (P ₂) F (V)
5		X (P ₁) Y (P ₂)
6		X (P ₁) Y (P ₂) F (V)
7		Z (P ₃)
8		Z (P ₃) F (V)
9		X (P ₁) Z (P ₃)
10		X (P ₁) Z (P ₃) F (V)
11		Y (P ₂) Z (P ₃)
12		Y (P ₂) Z (P ₃) F (V)
13		X (P ₁) Y (P ₂) Z (P ₃)
14		X (P ₁) Y (P ₂) Z (P ₃) F (V)

Program Example:

1. LIN XK12345 YH7567 ZKK3280 FKK40000

The instruction performs linear interpolation on 3 axes to the target position (K12345, H7567, ZKK3280). The target position can be an absolute coordinate or relative coordinate, which is determined by the ABST / INCT (G90/G91) instructions above and closest to LIN. The linear movement operates at speed 40kHz.

2. Moving path:



3. Other valid combinations of operands:

LIN XKK-345289 YDD10Z5 FD100

LIN XDD20 Y456@V4

The above instructions are legal, and device D is the register storing the value to be set up.

6 Motion Instructions and G-Code Instructions

Remarks:

Relevant special registers

D1822, D1823	Maximum speed on X axis. D1822 (low word); D1823 (high word).
D1824, D1825	Bias speed of X axis. D1824 (low word); D1825 (high word).
D1836	Acceleration time of X axis
D1837	Deceleration time of X axis
D1865	Stop mode for OX0 ~ 99 K1: complete remaining distance in next activation => considering remaining distance K2: execute the next instruction in next activation Others: restart
D1902, D1903	Maximum speed on Y axis. D1902 (low word); D1903 (high word).
D1904, D1905	Bias speed of Y axis. D1904 (low word); D1905 (high word).
D1916	Acceleration time of Y axis
D1917	Deceleration time of Y axis
D1982, D1983	Maximum speed on Z axis. D1982 (low word); D1983 (high word)
D1984, D1985	Bias speed of Z axis. D1984 (low word); D1985 (high word)
D1996	Acceleration time of Z axis
D1997	Deceleration time of Z axis

6 Motion Instructions & G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
02 03	CW / CCW	X (P ₁) Y (P ₂) Z (P ₃) I (P ₄) J (P ₅) K (P ₆) F (V)	Clockwise Arc / Clockwise Helical / Counterclockwise Arc / Counterclockwise Helical Movement (distance to center of arc)	20PM	-

Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
P ₁	*	*	*	*	*	*	<ul style="list-style-type: none"> ■ CW/CCW instruction supports V, Z index register modification on the devices. ■ You can place an M-Code instruction after CW/CCW.
P ₂	*	*	*	*	*	*	
P ₃	*	*	*	*	*	*	
P ₄	*	*	*	*	*	*	
P ₅	*	*	*	*	*	*	
P ₆	*	*	*	*	*	*	
V	*	*	*	*	*	*	

Operands:

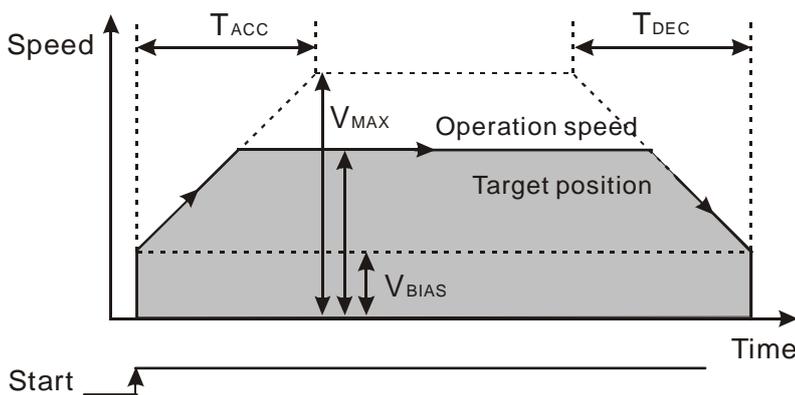
P₁: Target position on X axis P₂: Target position on Y axis P₃: Target position on Z axis

P₄: Distance to the center of arc/helix on X axis P₅: Distance to the center of arc/helix on Y axis

P₆: Distance to the center of arc/helix on Z axis V: Arc/helical interpolation speed

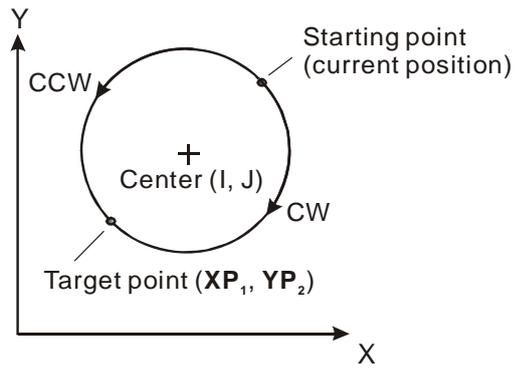
Explanations:

1. P₄, P₅, P₆: The vector-valued coordinate of center of arc or center of helix, i.e. distance to the center of arc/helix on X-Y-Z axes.
2. V_{MAX} = 500kHz
3. Range of parameters P₁, P₂, P₃, P₄, P₅, P₆: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
4. Acceleration/deceleration time and bias speed can be set up in special D.
5. Acceleration/deceleration time increases or decreases in proportional to the setting of V_{MAX}.
6. Timing diagram:

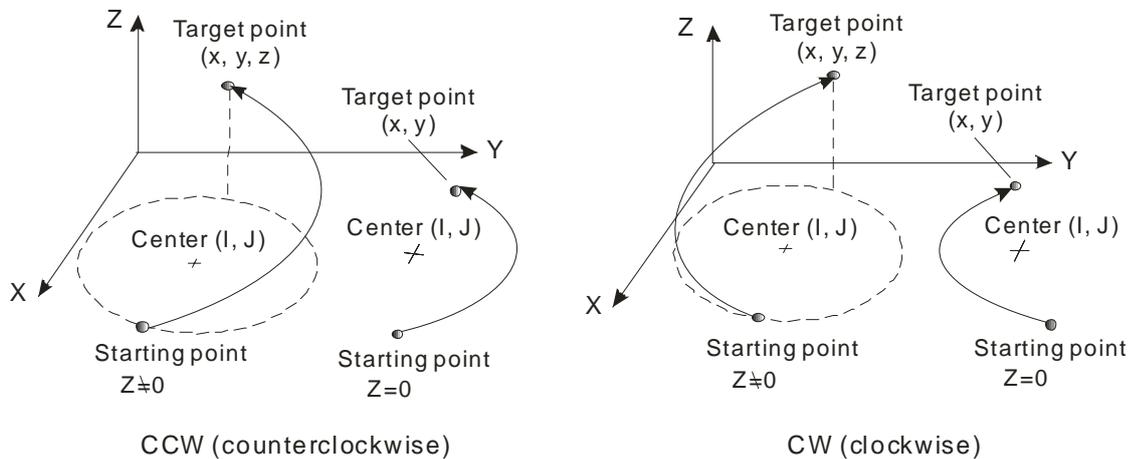


7. 2-axis arc interpolation: 20D model only supports arc interpolation on X-Y plane. 20M model supports arc / helical interpolation on 3 planes by G-Code instructions: G17 (X-Y plane), G18 (X-Z plane) or G19 (Y-Z plane). See the figure below for X-Y arc interpolation.

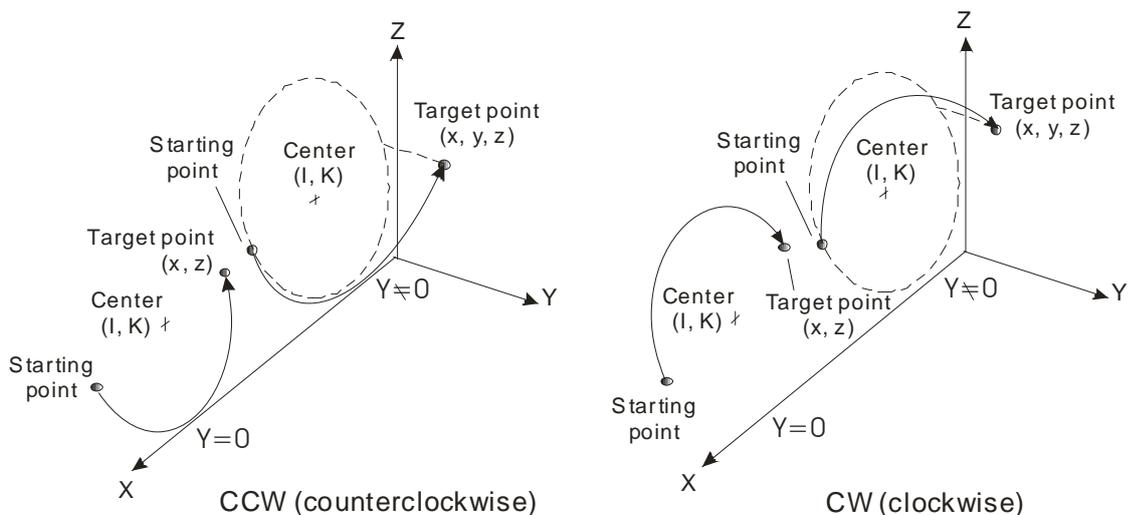
6 Motion Instructions and G-Code Instructions



8. 3-axis helical interpolation: Applicable to 20M model only. 3-axis helical interpolation is an advanced function to 2-axis arc interpolation. When a 3rd axis is added to arc interpolation, a helical interpolation can be performed, i.e. if the vertical value on Z axis is set to 0, the helical interpolation will simply perform 2-axis arc interpolation.
9. The figure below illustrates the moving paths of the helical interpolation with and without the value on Z-axis. (G17 mode, X-Y plane)

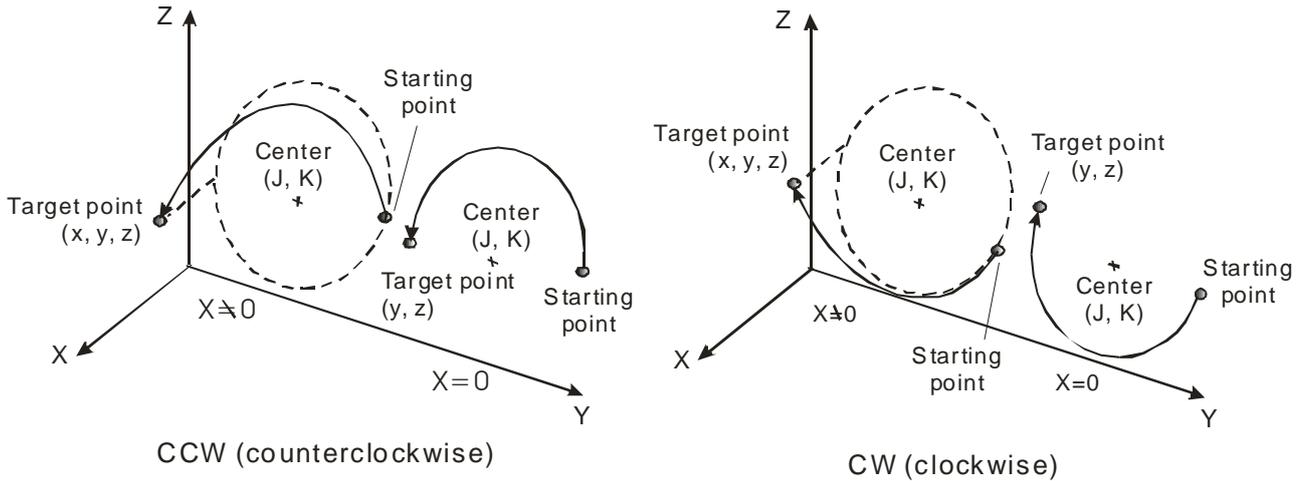


10. The figure below illustrates the moving paths of the helical interpolation with and without the value on Y-axis. (G18 mode, X-Z plane)



11. The figure below illustrates the moving paths of the helical interpolation with and without the value on X-axis. (G19 mode, Y-Z plane)

6 Motion Instructions & G-Code Instructions



12. 16-bit devices and 32-bit devices can be used together.
13. Programming rule: (1) Target position and distance to center is a must while the moving speed is not; (2) If the coordinate of X/Y/Z axis of target position is the same with that of current position, the same X/Y/Z value could be left out; (3) If the distance to center of arc/helix has no variation, it could also be left out.
14. If you set up the target position on the axis without setting up the moving speed, the operation will run at V_{MAX} .
15. The arc/helical movement can reach 360° .
16. 20D model only supports 2-axis (X-Y plane) arc interpolation, and according to the programming rule, there are 18 operand combinations for CW/CCW instruction.

NO.	Instruction	Operand combination
1	CW/CCW	X (P ₁) I (P ₃)
2		X (P ₁) I (P ₃) F (V)
3		X (P ₁) J (P ₄)
4		X (P ₁) J (P ₄) F (V)
5		X (P ₁) I (P ₃) J (P ₄)
6		X (P ₁) I (P ₃) J (P ₄) F (V)
7		Y (P ₂) I (P ₃)
8		Y (P ₂) I (P ₃) F (V)
9		Y (P ₂) J (P ₄)
10		Y (P ₂) J (P ₄) F (V)
11		Y (P ₂) I (P ₃) J (P ₄)
12		Y (P ₂) I (P ₃) J (P ₄) F (V)
13		X (P ₁) Y (P ₂) I (P ₃)
14		X (P ₁) Y (P ₂) I (P ₃) F (V)
15		X (P ₁) Y (P ₂) J (P ₄)
16		X (P ₁) Y (P ₂) J (P ₄) F (V)
17		X (P ₁) Y (P ₂) I (P ₃) J (P ₄)
18		X (P ₁) Y (P ₂) I (P ₃) J (P ₄) F (V)

17. 20M model supports 3-zxis arc/helical interpolation, and according to the programming rule, there are 78 operand combinations for CW/CCW instruction.

6 Motion Instructions and G-Code Instructions

NO.	Instruction	Operand combination	G17	G18	G19
1	CW/CCW	X(P ₁) I(P ₃)	✓	✓	
2		X(P ₁) I(P ₃) F(V)	✓	✓	
3		X(P ₁) J(P ₄)	✓		
4		X(P ₁) J(P ₄) F(V)	✓		
5		X(P ₁) I(P ₃) J(P ₄)	✓		
6		X(P ₁) I(P ₃) J(P ₄) F(V)	✓		
7		Y(P ₂) I(P ₃)	✓		
8		Y(P ₂) I(P ₃) F(V)	✓		
9		Y(P ₂) J(P ₄)	✓		✓
10		Y(P ₂) J(P ₄) F(V)	✓		✓
11		Y(P ₂) I(P ₃) J(P ₄)	✓		
12		Y(P ₂) I(P ₃) J(P ₄) F(V)	✓		
13		X(P ₁) Y(P ₂) I(P ₃)	✓	✓	
14		X(P ₁) Y(P ₂) I(P ₃) F(V)	✓	✓	
15		X(P ₁) Y(P ₂) J(P ₄)	✓		✓
16		X(P ₁) Y(P ₂) J(P ₄) F(V)	✓		✓
17		X(P ₁) Y(P ₂) I(P ₃) J(P ₄)	✓		
18		X(P ₁) Y(P ₂) I(P ₃) J(P ₄) F(V)	✓		
19		X(P ₁) Z(P ₃) I(P ₃)	✓	✓	
20		X(P ₁) Z(P ₃) I(P ₃) F(V)	✓	✓	
21		X(P ₁) Z(P ₃) J(P ₄)	✓		✓
22		X(P ₁) Z(P ₃) J(P ₄) F(V)	✓		✓
23		X(P ₁) Z(P ₃) I(P ₃) J(P ₄)	✓		
24		X(P ₁) Z(P ₃) I(P ₃) J(P ₄) F(V)	✓		
25		Y(P ₂) Z(P ₃) I(P ₃)	✓	✓	
26		Y(P ₂) Z(P ₃) I(P ₃) F(V)	✓	✓	
27		Y(P ₂) Z(P ₃) J(P ₄)	✓		✓
28		Y(P ₂) Z(P ₃) J(P ₄) F(V)	✓		✓
29		Y(P ₂) Z(P ₃) I(P ₃) J(P ₄)	✓		
30		Y(P ₂) Z(P ₃) I(P ₃) J(P ₄) F(V)	✓		
31		X(P ₁) Y(P ₂) Z(P ₃) I(P ₃)	✓	✓	
32		X(P ₁) Y(P ₂) Z(P ₃) I(P ₃) F(V)	✓	✓	
33		X(P ₁) Y(P ₂) Z(P ₃) J(P ₄)	✓		✓
34		X(P ₁) Y(P ₂) Z(P ₃) J(P ₄) F(V)	✓		✓
35		X(P ₁) Y(P ₂) Z(P ₃) I(P ₃) J(P ₄)	✓		
36		X(P ₁) Y(P ₂) Z(P ₃) I(P ₃) J(P ₄) F(V)	✓		
37		X(P ₁) K(P ₆)			✓
38		X(P ₁) K(P ₆) F(V)			✓
39		X(P ₁) I(P ₃) K(P ₆)			✓
40		X(P ₁) I(P ₃) K(P ₆) F(V)			✓
41		Z(P ₃) I(P ₃)			✓
42		Z(P ₃) I(P ₃) F(V)			✓

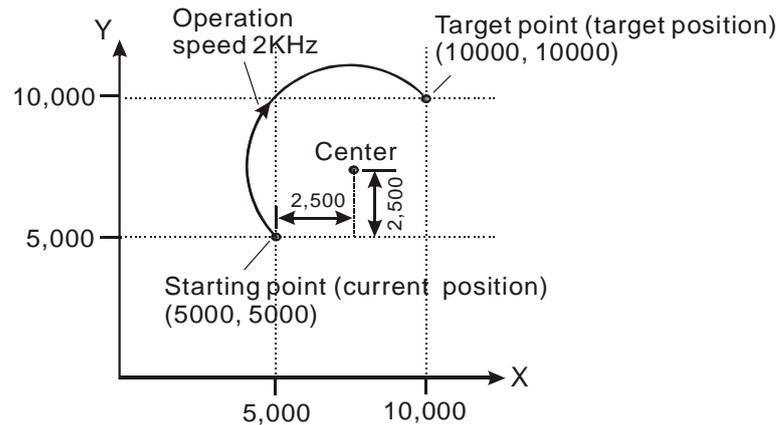
6 Motion Instructions & G-Code Instructions

NO.	Instruction	Operand combination	G17	G18	G19
43		Z (P ₃) K (P ₆)		✓	✓
44		Z (P ₃) K (P ₆) F (V)		✓	✓
45		Z (P ₃) I (P ₃) K (P ₆)		✓	
46		Z (P ₃) I (P ₃) K (P ₆) F (V)		✓	
47		X (P ₁) Z (P ₃) K (P ₆)		✓	✓
48		X (P ₁) Z (P ₃) K (P ₆) F (V)		✓	✓
49		X (P ₁) Z (P ₃) I (P ₃) K (P ₆)		✓	
50		X (P ₁) Z (P ₃) I (P ₃) K (P ₆) F (V)		✓	
51		X (P ₁) Y (P ₂) K (P ₆)		✓	✓
52		X (P ₁) Y (P ₂) K (P ₆) F (V)		✓	✓
53		X (P ₁) Y (P ₂) I (P ₃) K (P ₆)		✓	
54		X (P ₁) Y (P ₂) I (P ₃) K (P ₆) F (V)		✓	
55		Y (P ₂) Z (P ₃) K (P ₆)		✓	✓
56		Y (P ₂) Z (P ₃) K (P ₆) F (V)		✓	✓
57		Y (P ₂) Z (P ₃) I (P ₃) K (P ₆)		✓	
58		Y (P ₂) Z (P ₃) I (P ₃) K (P ₆) F (V)		✓	
59		X (P ₁) Y (P ₂) Z (P ₃) K (P ₆)		✓	✓
60		X (P ₁) Y (P ₂) Z (P ₃) K (P ₆) F (V)		✓	✓
61		X (P ₁) Y (P ₂) Z (P ₃) I (P ₃) K (P ₆)		✓	
62		X (P ₁) Y (P ₂) Z (P ₃) I (P ₃) K (P ₆) F (V)		✓	
63		Y (P ₂) K (P ₆)			✓
64		Y (P ₂) K (P ₆) F (V)			✓
65		Y (P ₂) J (P ₄) K (P ₆)			✓
66		Y (P ₂) J (P ₄) K (P ₆) F (V)			✓
67		Z (P ₃) J (P ₄)			✓
68		Z (P ₃) J (P ₄) F (V)			✓
69		Z (P ₃) J (P ₄) K (P ₆)			✓
70		Z (P ₃) J (P ₄) K (P ₆) F (V)			✓
71		Y (P ₂) Z (P ₃) J (P ₄) K (P ₆)			✓
72		Y (P ₂) Z (P ₃) J (P ₄) K (P ₆) F (V)			✓
73		X (P ₁) Y (P ₂) J (P ₄) K (P ₆)			✓
74		X (P ₁) Y (P ₂) J (P ₄) K (P ₆) F (V)			✓
75		X (P ₁) Z (P ₃) J (P ₄) K (P ₆)			✓
76		X (P ₁) Z (P ₃) J (P ₄) K (P ₆) F (V)			✓
77		X (P ₁) Y (P ₂) Z (P ₃) J (P ₄) K (P ₆)			✓
78		X (P ₁) Y (P ₂) Z (P ₃) J (P ₄) K (P ₆) F (V)			✓

Program Example:

- Example (1): Set up absolute coordinate system. Apply CW clockwise arc instruction and set target position of arc as (10000, 10000), distance to center of arc as (2500, 2500), and output speed as 2,000Hz.

6 Motion Instructions and G-Code Instructions

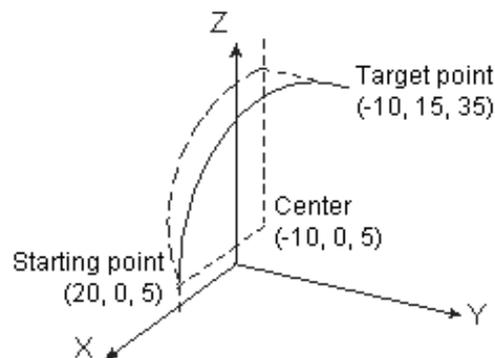


The program should be written as:

ABST

CW XK10000 YK10000 IK2500 JK2500 FK2000

- Example (2): Set up absolute coordinate system on X-Z plane (G18). Apply CW (clockwise) helical instruction and set target position of arc as (-10, 15, 35), distance to center of arc as (-10, 0, 5), and output speed as 2,000Hz.



The program should be written as:

ABS;

G18;

CW XK-10 YJ15 ZK35 IK-30 JK0 (can be ignored) KK0 (can be ignored) FK2000;

- Other valid combinations of operands: The below 5 instructions are all legal, and device D is the register storing the value to be set up. However, in actual execution, the user has to set up the corresponding coordinate plane. In general when there is no coordinate plane setting, X-Y plane will be applied.

CW XK123 YDD10V7 I450000 JD10 FKK50000

CCW XHAABB YDD100 Z30 IK4500 KKK3500 FK4000@V5

CW XHAABB YDD100 IK4500 JD500

CW XHAABB YDD100 ZD30 IK4500 KD200 FK4000@V5

CCW XHAABB YDD100 IK4500 JK3500

6 Motion Instructions & G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
04 05	CW / CCW	$X(P_1) \quad Y(P_2) \quad Z(P_3)$ $R(L) \quad F(V)$	Clockwise Arc / Clockwise Helical / Counterclockwise Arc / Counterclockwise Helical Movement (radius)	20PM	-

Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
P ₁	*	*	*	*	*	*	<ul style="list-style-type: none"> CW/CCW instruction supports V, Z index register modification on the devices. You can place an M-Code instruction after CW/CCW.
P ₂	*	*	*	*	*	*	
P ₃	*	*	*	*	*	*	
L	*	*	*	*	*	*	
V	*	*	*	*	*	*	

Operands:

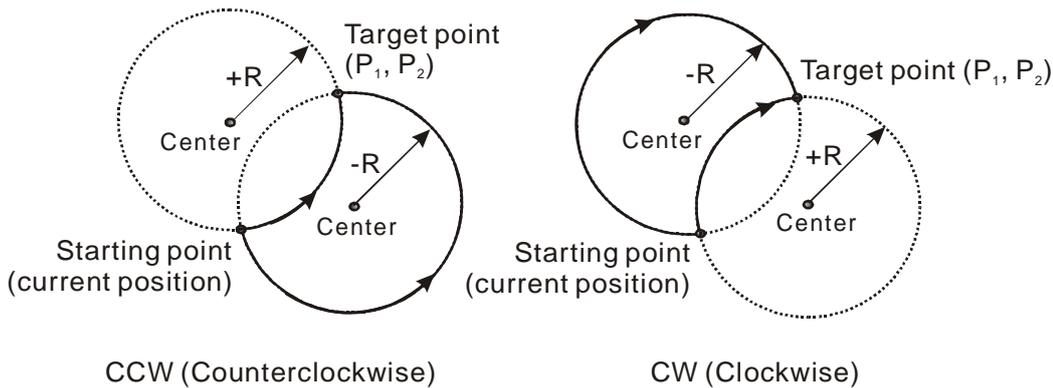
P₁: Target position on X axis P₂: Target position on Y axis P₃: Target position on Z axis

L: Radius of arc / helix (R = "+" when radian < 180°; R = "-" when radian > 180°)

V: Arc/helical interpolation speed

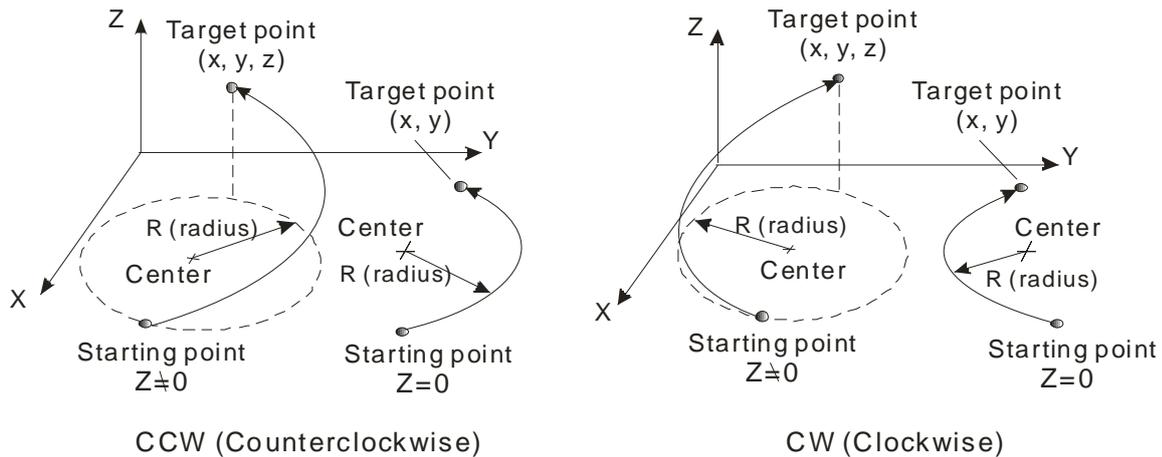
Explanations:

- Maximum $V = V_{MAX}$.
- Range of parameters: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
- Acceleration/deceleration time and bias speed can be set up in special D.
- Acceleration/deceleration time increases or decreases in proportional to the setting of V_{MAX} .
- 2-axis arc interpolation: 20D model only supports arc interpolation on X-Y plane. 20M model supports arc / helical interpolation on 3 planes by G-Code instructions: G17 (X-Y plane), G18 (X-Z plane) or G19 (Y-Z plane). See the figure below for X-Y arc interpolation

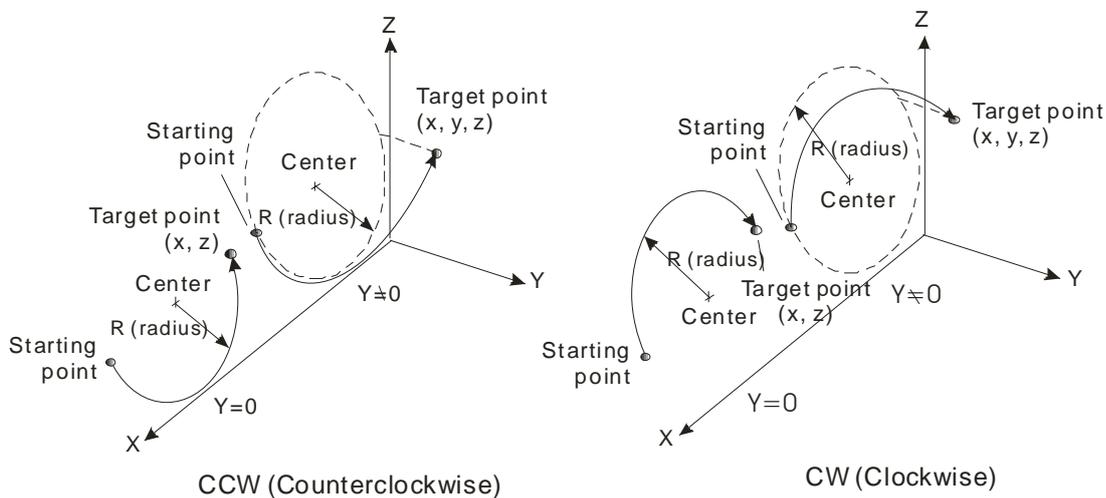


- 3-axis helical interpolation: Applicable to 20M model only. 3-axis helical interpolation is an advanced function to 2-axis arc interpolation. When a 3rd axis is added to arc interpolation, a helical interpolation can be performed, i.e. if the vertical value on Z axis is set to 0, the helical interpolation will simply perform 2-axis arc interpolation
- The figure below illustrates the moving paths of the helical interpolation with and without the value on Z-axis. (G17 mode, X-Y plane).

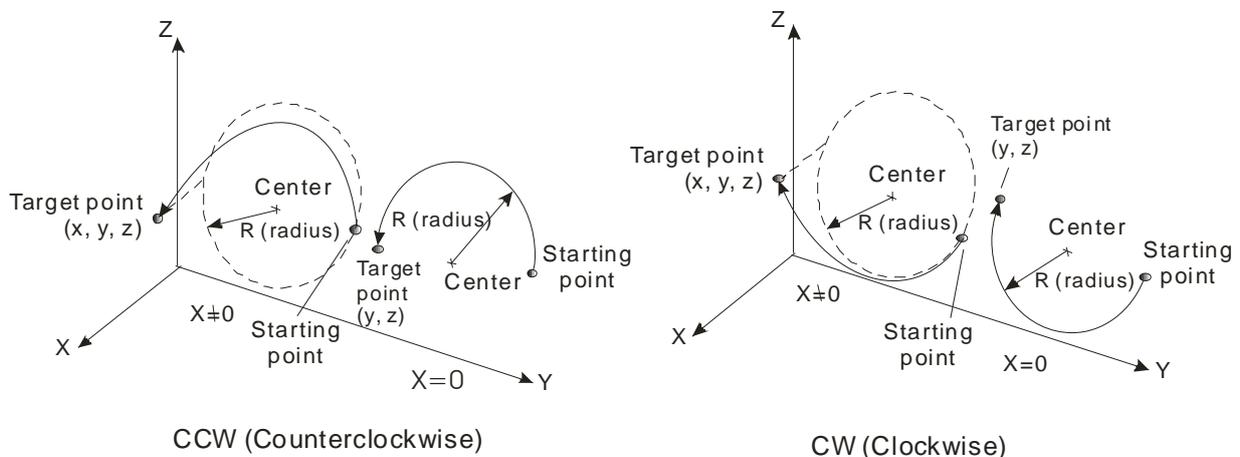
6 Motion Instructions and G-Code Instructions



8. The figure below illustrates the moving paths of the helical interpolation with and without the value on Y-axis. (G18 mode, X-Z plane).



9. The figure below illustrates the moving paths of the helical interpolation with and without the value on X-axis. (G19 mode, Y-Z plane)



10. 16-bit devices and 32-bit devices can be used together.
11. Programming rule: (1) Target position and radius is a must while the moving speed is not; (2) If the coordinate of X/Y/Z axis of target position is the same with that of current position, the same X/Y/Z value could be left out
12. If you set up the target position on the axis without setting up the moving speed, the operation will run at V_{MAX}

6 Motion Instructions & G-Code Instructions

13. 20D model only supports 2-axis (X, Y plane) arc interpolation, and according to the programming rule, there are 6 operand combinations for CW/CCW instruction.

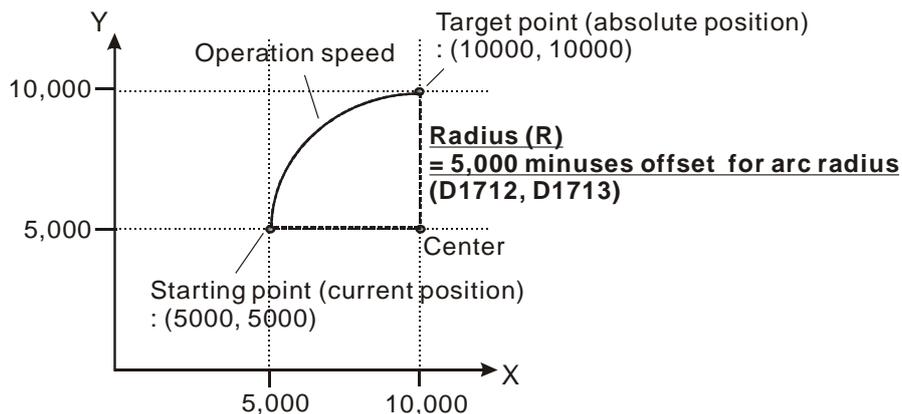
NO.	Instruction	Operand combination
1	CW/CCW	X (P ₁) R (L)
2		X (P ₁) R (L) F (V)
3		Y (P ₂) R (L)
4		Y (P ₂) R (L) F (V)
5		X (P ₁) Y (P ₂) R (L)
6		X (P ₁) Y (P ₂) R (L) F (V)

14. 20M model supports 3-axis arc/helical interpolation, and according to the programming rule, there are 14 operand combinations for CW/CCW instruction.

NO.	Instruction	Operand combination	G17	G18	G19
1	CW/CCW	X (P ₁) R (L)	✓	✓	
2		X (P ₁) R (L) F (V)	✓	✓	
3		Y (P ₂) R (L)	✓		✓
4		Y (P ₂) R (L) F (V)	✓		✓
5		X (P ₁) Y (P ₂) R (L)	✓	✓	✓
6		X (P ₁) Y (P ₂) R (L) F (V)	✓	✓	✓
7		X (P ₁) Z (P ₃) R (L)	✓	✓	✓
8		X (P ₁) Z (P ₃) R (L) F (V)	✓	✓	✓
9		Y (P ₂) Z (P ₃) R (L)	✓	✓	✓
10		Y (P ₂) Z (P ₃) R (L) F (V)	✓	✓	✓
11		X (P ₁) Z (P ₃) Y (P ₂) R (L)	✓	✓	✓
12		X (P ₁) Z (P ₃) Y (P ₂) R (L) F (V)	✓	✓	✓
13		Z (P ₃) R (L)		✓	✓
14		Z (P ₃) R (L) F (V)		✓	✓

Program Example:

1. Example (1): Set up absolute coordinate system. Apply CW clockwise arc instruction and set target position of arc as (10000, 10000), radius = 500 (R = "+" when radian < 180°), and output speed as 100 pulses per second.



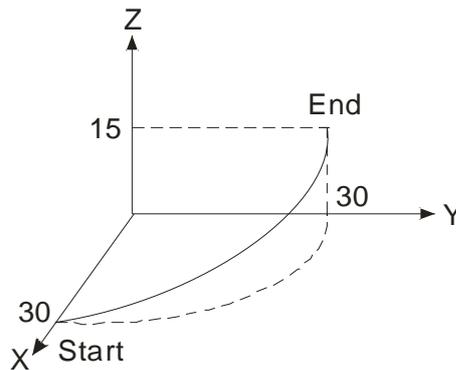
6 Motion Instructions and G-Code Instructions

The program should be written as:

ABS

CW XK10000 YK10000 RK5000 FK1000

- Example (2): Set up absolute coordinate system. Apply CCW counterclockwise helical instruction and set target position of helix as (0, 30, 15), radius = 30.0 (R = "+" when radian < 180°), and output speed as 100 pulses per second.



The program should be written as:

ABS;

G17;

CCW XK0 YK30 ZK15 RK30 FK100

Points to note

- For how to set up arc radius compensation, please refer to MON 15 RADC instruction.
- The arc/helical movement cannot reach 360°
- Other valid combinations of operands: The below 5 instructions are all legal, and device D is the register storing the value to be set up.

CW XK123 YDD10 RKK450000 FKK50000

CCW XHAABB R350000 F400000@V5

6 Motion Instructions & G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
06	TIM	(T)	Pause Time	20PM	10PM

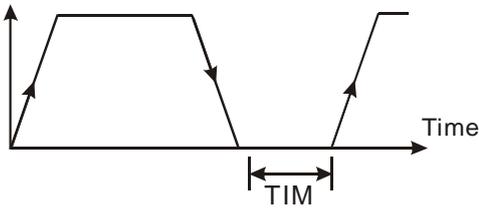
Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
T	*	*	*	*	*	*	<ul style="list-style-type: none"> TIM instruction supports V, Z index register modification on the devices. See the specifications of DVP-PM for the applicable range of each device. You can place an M-Code instruction after TIM.

Operands:

T: Pause time (unit: 10ms => K100 refers to pausing for 1 second)

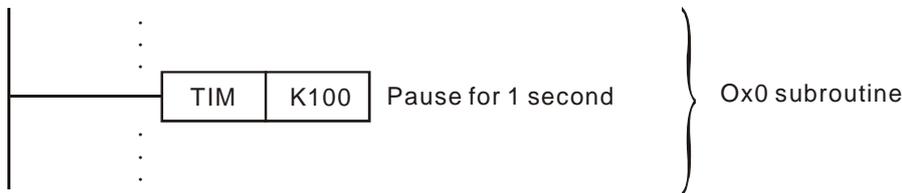
Explanations:

- TIM instruction is used for setting up the pause time between one instruction and another instruction.



- Range of parameters: (16-bit) K = 0 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.

Program Example:



6 Motion Instructions and G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
07	DRVZ	N/A	Return to Mechanical Zero Point (Zero Return)	20PM	-

Explanations:

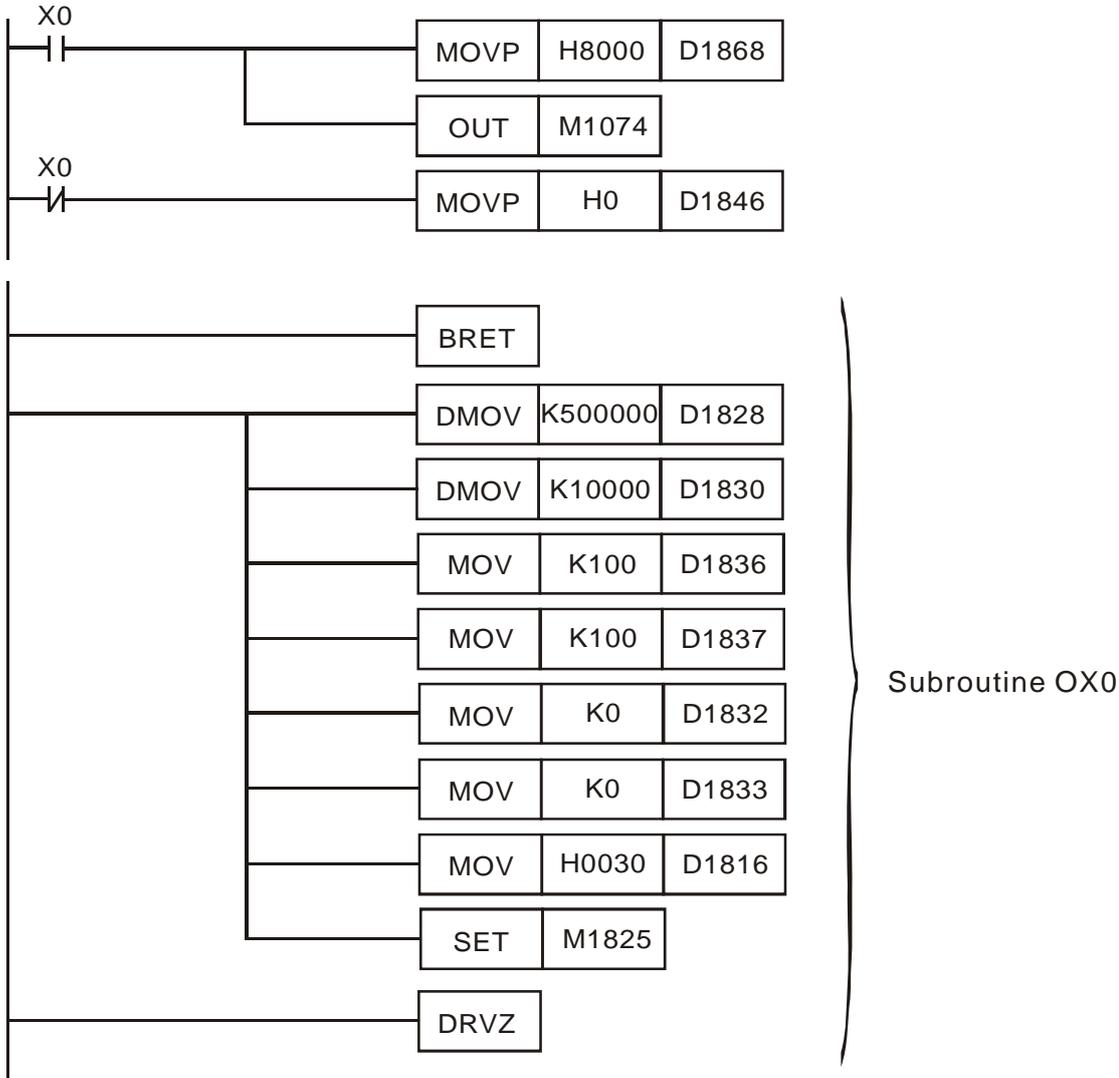
1. You can place an M-Code instruction after DRVZ.
2. Before enabling DRVZ, you have to first set up the following parameters (for relative D registers please refer to **Remarks** explained later):
 - a) Zero return speed (V_{RT}): When zero return is triggered, the system will operate at zero return speed (V_{RT}) until zero point signal (DOG) is triggered. V_{RT} cannot be modified during the execution. Range: 0 ~ 500kHz. Limitation: $V_{MAX} > V_{RT} > V_{BIAS}$
 - b) Zero return deceleration speed (V_{CR}): When zero point signal (DOG) is triggered, the system will decelerate to zero return deceleration speed (V_{CR}). In order to accurately stop at zero point, it is suggested that you set V_{CR} at low speed. Range: 0 ~ 500kHz. Limitation: $V_{CR} < V_{RT}$. V_{CR} cannot be modified during deceleration..
 - c) Acceleration time: The time spent on accelerating to V_{RT} .
 - d) Deceleration time: The time spent on decelerating from V_{RT} to V_{CR} and from V_{CR} to zero speed.
 - e) Number of zero point signals (PG0) in zero return (N): Reference signal for the motor to decelerate and stop. When a DOG signal is detected, the program will start to count the number of PG0 signals for the reference to stop. Range: 0 ~ +32,767 pulses.
 - f) Number of pulse signals in zero return (P): Reference signal for the motor to decelerate and stop. Positive set value refers to forward-running pulses, and negative set value refers to reverse-running pulses. Range: -32,768 ~ 32,767 pulses
 - g) Confirm that the flags for disabling zero return on X, Y axis are correctly set ON/OFF.
3. In addition, parameters below should be set up in special registers. D1816 for X axis; D1896 for Y axis, and D1976 for Z axis.
 - a) Zero return direction
 - b[8] = 0: Zero return executes in positive direction, i.e. current position(CP) increases.
 - b[8] = 1: Zero return executes in negative direction, i.e. current position(CP) decreases.
 - b) Zero return mode
 - b[9] = 0: normal mode. When DOG signal is generated, after the specified PG0 signals (N) then the following pulse signals (P), the motor will stop immediately.
 - b[9] = 1: overwrite mode. When DOG signal is generated, the motor will stop according to either PG0 signals (N) or pulse signals (P), which is finished first.
 - c) Detecting DOG signal rising/falling edge triggered
 - b[10] = 0: On. DOG signal falling edge triggered.
 - b[11] = 1: Off. DOG signal rising edge triggered.
 - d) According to the 4 combinations of parameters b) and c), there are four zero return modes. See 3.12 for how they work.

Program Example:

When X0 is ON, DRVZ instruction in OX0 will be executed, and the zero return on Y axis will be disabled. The

6 Motion Instructions & G-Code Instructions

system accelerates to V_{RT} (500kHz) by 100ms and operates at V_{RT} until DOG is triggered. When DOG is triggered, the system decelerates to V_{CR} (10kHz) by 100ms. In this case, zero return mode is set as normal mode and PG0 signal is set as falling-edge triggered by DOG. Therefore, once the falling edge of DOG signal is triggered, the system will stop after the specified PG0 signals (N) plus the specified pulse signals (P) is reached. However, in this example both PG0 signals and pulse signals are set to 0, so the zero return will complete as soon as falling edge of DOG is triggered.



Remarks:

1. Relevant flags:

M1745	Disable zero return of X axis in OX
M1825	Disable zero return of Y axis in OX
M1985	Disable zero return of Z axis in OX
M1074	Enable OX motion subroutine

2. Relevant special registers:

D1816	Parameter setting of X axis
D1846	Operation commands for X axis
D1868	Setting up the No. of OX
D1828	Zero return speed of X axis: V_{RT} (low word)
D1829	Zero return speed of X axis: V_{RT} (high word)

6 Motion Instructions and G-Code Instructions

D1830	Zero return deceleration speed of X axis: V_{CR} (low word)
D1831	Zero return deceleration speed of X axis: V_{CR} (high word)
D1832	Number of zero point signals (PG0) on X axis: N
D1833	Number of pulses for zero return on X axis: P
D1836	Acceleration time of X axis: T_{ACC}
D1837	Deceleration time of X axis: T_{DEC}
D1896	Parameter setting of Y axis
D1908	Zero return speed of Y axis: V_{RT} (low word)
D1909	Zero return speed of Y axis: V_{RT} (high word)
D1910	Zero return deceleration speed of Y axis (low word)
D1911	Zero return deceleration speed of Y axis (high word)
D1912	Number of zero point signals (PG0) on Y axis: N
D1913	Number of pulses for zero return on Y axis: P
D1916	Acceleration time of Y axis: T_{ACC}
D1917	Deceleration time of Y axis: T_{DEC}
D1976	Parameter setting of Z axis
D1988	Zero return speed of Z axis: V_{RT} (low word)
D1989	Zero return speed of Z axis: V_{RT} (high word)
D1990	Zero return deceleration speed of Z axis (low word)
D1991	Zero return deceleration speed of Z axis (high word)
D1992	Number of zero point signals (PG0) on Z axis: N
D1993	Number of pulses for zero return on Z axis: P
D1996	Acceleration time of Z axis: T_{ACC}
D1997	Deceleration time of Z axis: T_{DEC}

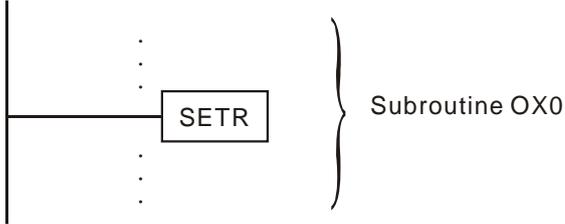
6 Motion Instructions & G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
				20PM	-
08	SETR	N/A	Set up Electrical Zero Point		

Explanations:

1. You can place an M-Code instruction after SETR
2. When SETR is executed, you can set the current position of X/Y/Z axis as the electrical zero point, i.e. move the content in the current position (CP) register into the register for electrical zero point.

Program Example:



Remarks:

Relevant special registers

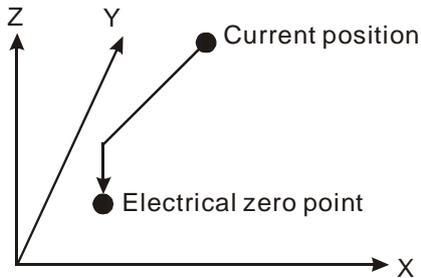
D1848	Current position of X axis: CP (low word)
D1849	Current position of X axis: CP (high word)
D1866	Electrical zero point address on X axis (low word)
D1867	Electrical zero point address on X axis (high word)
D1928	Current position of Y axis: CP (low word)
D1929	Current position of Y axis: CP (high word)
D1946	Electrical zero point address on Y axis (low word)
D1947	Electrical zero point address on Y axis (high word)
D2008	Current position of Z axis: CP (low word)
D2009	Current position of Z axis: CP (high word)
D2026	Electrical zero point address on Z axis (low word)
D2027	Electrical zero point address on Z axis (high word)

6 Motion Instructions and G-Code Instructions

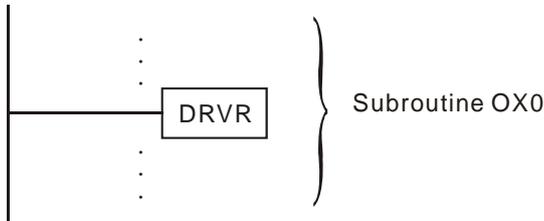
MON	Mnemonic	Operands	Function	Controllers	
				20PM	-
09	DRVR	N/A	Return to Electrical Zero Point		

Explanations:

1. You can place an M-Code instruction after DRVR.
2. When DRVR instruction is executed, X / Y / Z axes will return to electrical zero point at V_{MAX} (0 ~ 500kHz).
3. Moving path:



Program Example:



Remarks:

Relevant special registers

D1822	Maximum speed on X axis: V_{MAX} (low word)
D1823	Maximum speed on X axis: V_{MAX} (high word)
D1848	Current position of X axis: CP (low word)
D1849	Current position of X axis: CP (high word)
D1866	Electrical zero point address on X axis (low word)
D1867	Electrical zero point address on X axis (high word)
D1902	Maximum speed on Y axis: V_{MAX} (low word)
D1903	Maximum speed on Y axis: V_{MAX} (high word)
D1928	Current position of Y axis: CP (low word)
D1929	Current position of Y axis: CP (high word)
D1946	Electrical zero point address on Y axis (low word)
D1947	Electrical zero point address on Y axis (high word)
D1982	Maximum speed of Z axis: V_{MAX} (low word)
D1983	Maximum speed of Z axis: V_{MAX} (high word)
D2008	Current position of Z axis: CP (low word)
D2009	Current position of Z axis: CP (high word)
D2026	Electrical zero point address on Z axis (low word)
D2027	Electrical zero point address on Z axis (high word)

6 Motion Instructions & G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
10	INTR	X (P ₁) Y (P ₂) F (V)	2-axis Single-Speed Linear Interpolation (ignoring remaining distance)	20PM	-

Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
P ₁	*	*	*	*	*	*	<ul style="list-style-type: none"> INTR instruction supports V, Z index register modification on the devices. See specifications of DVP-PM for the applicable range of each device. You can place an M-Code instruction after INTR.
P ₂	*	*	*	*	*	*	
V	*	*	*	*	*	*	

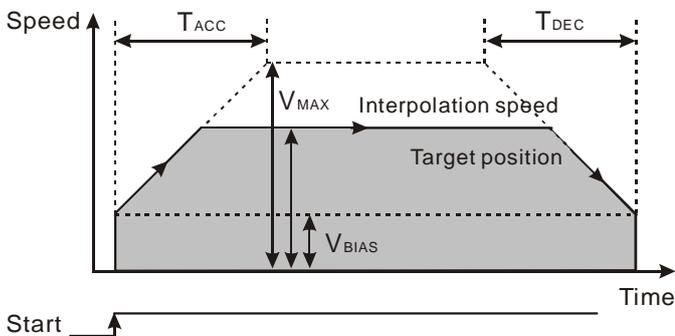
Operands:

P₁: Target position on X axis P₂: Target position on Y axis

V: Speed for 2-axis linear interpolation

Explanations:

- The functions of LIN and INTR are the same, except that LIN can set up stop mode, considering remaining distance.
- Maximum $V = V_{MAX}$.
- Range of parameters: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
- Acceleration/deceleration time and bias speed can be set up in special D.
- Acceleration/deceleration time increases or decreases in proportional to the setting of V_{MAX} .
- Timing diagram:



- The interpolation speed is monitored by special registers: D1850 ~ D1851 for X axis; D1930 ~ D1931 for Y axis.
- Target position is required, but moving speed could be left out, and therefore there are 6 operand combinations for INTR instruction.

NO.	Instruction	Operand combination
1	INTR	X (P ₁)
2		X (P ₁) F (V)
3		Y (P ₂)
4		Y (P ₂) F (V)
5		X (P ₁) Y (P ₂)
6		X (P ₁) Y (P ₂) F (V)

Remarks:

See Remarks of LIN instruction for relevant special registers.

6 Motion Instructions and G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
11	SINTR	X (P ₁)/Y (P ₂), F (V)	1-axis Single-speed Positioning with Additional Distance	20PM	-

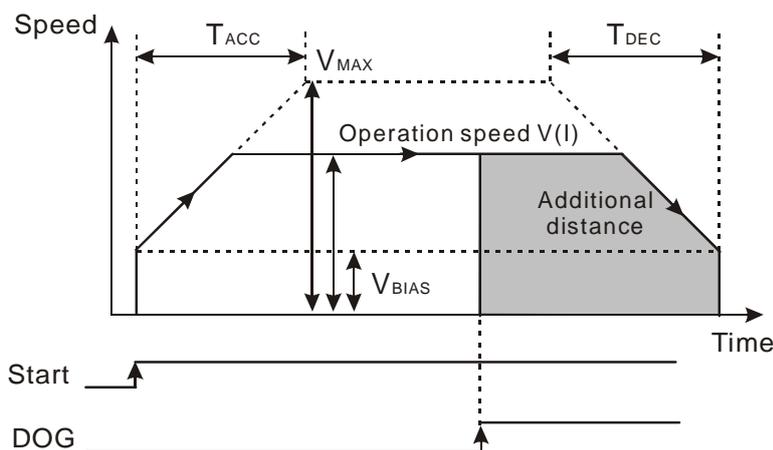
Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
P ₁	*	*	*	*	*	*	<ul style="list-style-type: none"> SINTR instruction supports V, Z index register modification on the devices. See specifications of DVP-PM for the applicable range of each device. You can place an M-Code instruction after SINTR.
P ₂	*	*	*	*	*	*	
V	*	*	*	*	*	*	

Operands:

P₁: Additional distance on X axis P₂: Additional distance on Y axis V: Operation speed V(l)

Explanations:

- When SINTR instruction is enabled, the operation speed will start from V_{BIAS} and accelerate to $V(l)$ and then operate stably. When the execution encounters DOG signal, it will follow the additional distance set by the instruction and complete the positioning process.
- Maximum $V = V_{MAX}$.
- The first operand of SINTR can be either X axis or Y axis for single speed positioning with additional distance.
- Range of parameters: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
- Acceleration/deceleration time and bias speed can be set up in special D.
- Acceleration/deceleration time increases or decreases in proportional to the setting of V_{MAX} .
- The 16-bit devices and 32-bit devices can be used together.
- Timing diagram:



- Both additional distance and moving speed are required, and therefore there are 2 operand combinations for SINTR instruction.

NO.	Instruction	Operand combination
1	SINTR	X (P ₁) F (V)
2		Y (P ₂) F (V)

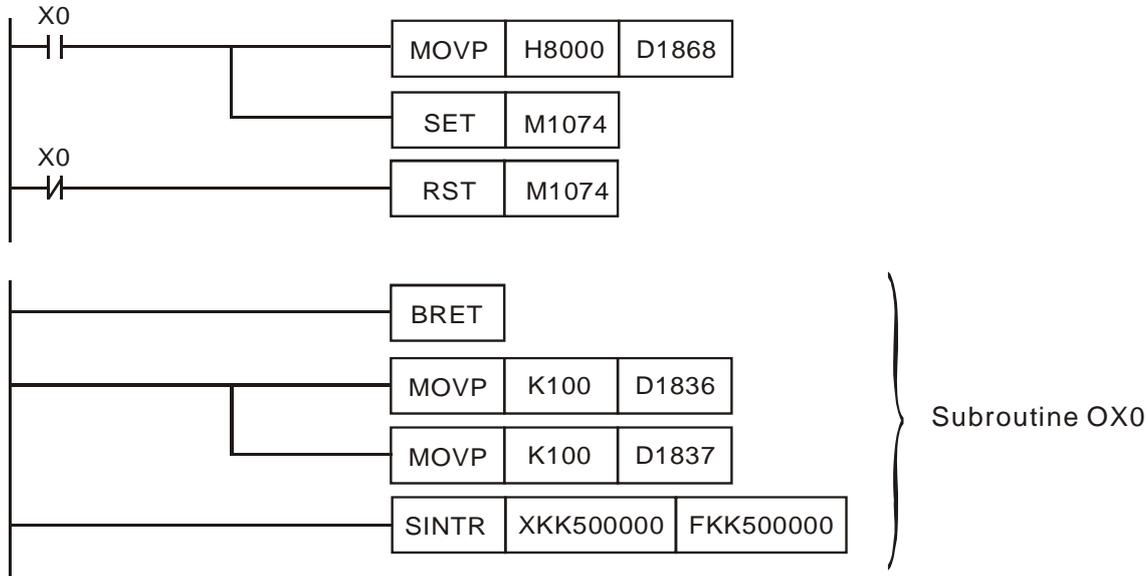
Program Example:

- When X0 = ON, SINTR instruction in OX0 subroutine will be executed. X axis accelerates to 500kHz in 100ms.

6 Motion Instructions & G-Code Instructions

When DOG signal is triggered, SINTR instruction will finish the single speed positioning on X axis with the additional 500,000 pulses set by the first operand.

- When X0 = OFF, program OX0 and SINTR instruction will be disabled.



Remarks:

- Even if absolute coordinate system is adopted in the program, the displacement will be regarded as additional distance once SINTR is executed,
- Relevant special registers:

D1836	Acceleration time of X axis: T_{ACC}
D1837	Deceleration time of X axis: T_{DEC}
D1848	Current position of X axis: CP (low word)
D1849	Current position of X axis: CP (high word)
D1916	Acceleration time of Y axis: T_{ACC}
D1917	Deceleration time of Y axis: T_{DEC}
D1928	Current position of Y axis: CP (low word)
D1929	Current position of Y axis: CP (high word)

6 Motion Instructions and G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
12	DINTR	X (P ₁)/Y (P ₂), F (V ₁) F (V ₂)	1-axis Double-speed Positioning with Additional Distance	20PM	-

Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
P ₁	*	*	*	*	*	*	<ul style="list-style-type: none"> ■ DINTR instruction supports V, Z index register modification on the devices. ■ See specifications of DVP-PM for the applicable range of each device. ■ You can place an M-Code instruction after DINTR.
P ₂	*	*	*	*	*	*	
V ₁	*	*	*	*	*	*	
V ₂	*	*	*	*	*	*	

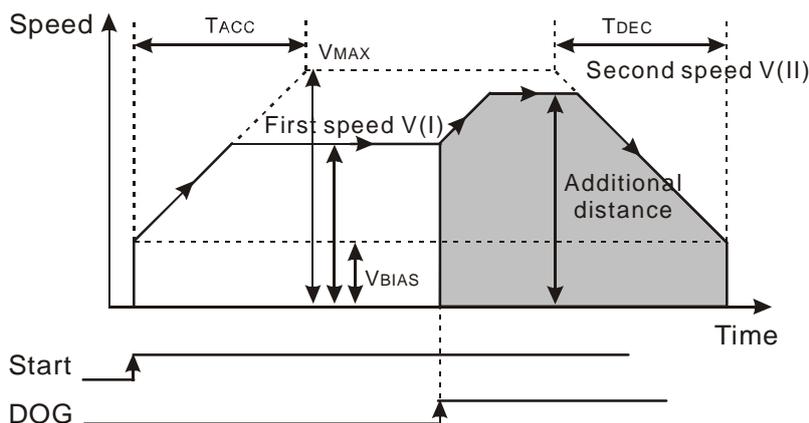
Operands:

P₁: Additional distance on X axis P₂: Additional distance on Y axis V₁: The first speed

V₂: The second speed

Explanations:

1. When DINTR instruction is enabled, the operation speed will start from V_{BIAS} and accelerate to V(I) and then operate stably. When the execution encounters DOG signal, it will further accelerate to V(II) then finish the additional distance set by the instruction.
2. Maximum V₁, V₂ = V_{MAX}.
3. The first operand of DINTR can be either X axis or Y axis for double speed positioning with additional distance.
4. Range of parameters: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
5. Acceleration/deceleration time and bias speed can be set up in special D.
6. Acceleration/deceleration time increases or decreases in proportional to the setting of V_{MAX}.
7. The 16-bit devices and 32-bit devices can be used together.
8. Timing diagram:



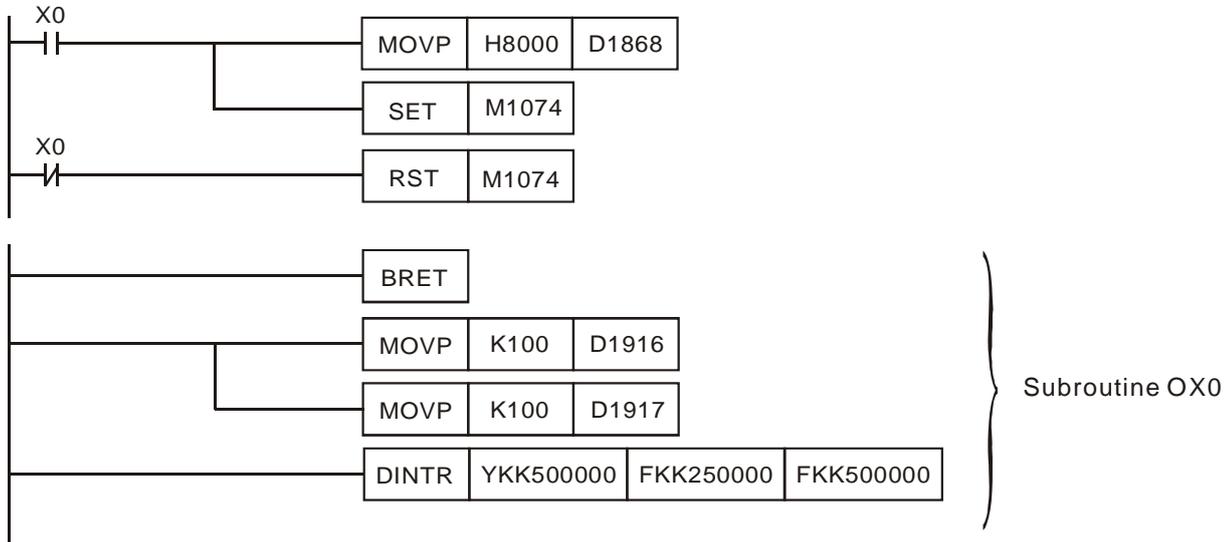
9. Both additional distance and moving speed are required, and therefore there are 2 operand combinations for SINTR instruction.

NO.	Instruction	Operand combination
1	DINTR	X (P ₁) F (V ₁) F (V ₂)
2		Y (P ₂) F (V ₁) F (V ₂)

6 Motion Instructions & G-Code Instructions

Program Example:

1. When X0 = ON, DINTR instruction in OX0 subroutine will be executed. Y axis accelerates to the first speed 250kHz in 100ms and operate at the speed stably. When DOG signal is triggered, it will further accelerate to the second speed 500kHz, and finish the double speed positioning on Y axis with the additional 500,000 pulses.
2. When X0 = OFF, program OX0 and DINTR instruction will be disabled.



Remarks:

Relevant special registers

D1836	Acceleration time of X axis: T_{ACC}
D1837	Deceleration time of X axis: T_{DEC}
D1848	Current position of X axis: CP (low word)
D1849	Current position of X axis: CP (high word)
D1916	Acceleration time of Y axis: T_{ACC}
D1917	Deceleration time of Y axis: T_{DEC}
D1928	Current position of Y axis: CP (low word)
D1929	Current position of Y axis: CP (high word)

6 Motion Instructions and G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
13	MOVC	X(L ₁) Y(L ₂)	Set up Offset of Linear Movement	20PM	-

Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
L ₁	*	*	*	*	*	*	<ul style="list-style-type: none"> ■ MOVC instruction supports V, Z index register modification on the devices. ■ See specifications of DVP-PM for the applicable range of each device. ■ You can place an M-Code instruction after MOVC.
L ₂	*	*	*	*	*	*	

Operands:

L₁: Offset for target position of X axis L₂: Offset for target position of Y axis

Explanations:

- When MOVC instruction is executed, the offset value will automatically be written into special registers: D1708 ~ D1709 for X axis; D1724 ~ D1725 for Y axis. MOVC instruction will adjust the target position of linear movement according to the offset values.
- You can set up only the offset value of X axis, e.g. MOVC XDD0.
- The offset of linear movement can be adopted for DRV, LIN and INTR instructions..
- Range of parameters: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
- The 16-bit parameter devices and 32-bit parameter devices can be used together.

NO.	Instruction	Operand combination
1	MOVC	X(L ₁)
2		X(L ₁) Y(L ₂)

Remarks:

Relevant special registers

D1708	Offset value for target position of X-axis (low word)
D1709	Offset value for target position of X-axis (high word)
D1724	Offset value for target position of Y-axis (low word)
D1725	Offset value for target position of Y-axis (high word)

6 Motion Instructions & G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
14	CNTC	I (L ₁) J (L ₂)	Set up Offset for Distance to Center of Arc	20PM	-

Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
L ₁	*	*	*	*	*	*	<ul style="list-style-type: none"> ■ CNTC instruction supports V, Z index register modification on the devices. ■ See specifications of DVP-PM for the applicable range of each device. ■ You can place an M-Code instruction after CNTC.
L ₂	*	*	*	*	*	*	

Operands:

L₁: Offset of distance to center of arc on X axis L₂: Offset of distance to center of arc on Y axis

Explanations:

1. When CNTC instruction is executed, the offset value will automatically be written into special registers: D1710 ~ D1711 for X axis; D1726 ~ D1727 for Y axis. CNTC instruction will adjust the distance to center of arc according to the offset values.
2. The offset of distance to center of arc can be adopted in CW and CCW instructions.
3. Range of parameters: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
4. The 16-bit devices and 32-bit devices can be used together.

Remarks:

Relevant special registers

D1710	Offset for distance to center of arc on X-axis (low word)
D1711	Offset for distance to center of arc on X-axis (high word)
D1726	Offset for distance to center of arc on Y-axis (low word)
D1727	Offset for distance to center of arc on Y-axis (high word)

6 Motion Instructions and G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
15	RADC	R(L)	Set up Offset of Arc Radius	20PM	-

Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
L	*	*	*	*	*	*	<ul style="list-style-type: none"> ■ RADC instruction supports V, Z index register modification on the devices. ■ See specifications of DVP-PM for the applicable range of each device. ■ You can place an M-Code instruction after RADC.

Operands:

L: Offset of arc radius

Explanations:

1. When RADC instruction is executed, the offset value will automatically be written into special registers: D1712 ~ D1713. RADC instruction will adjust the arc radius according to the offset value.
2. The offset of arc radius can be adopted in CW and CCW instructions.
3. Range of parameters: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
4. The 16-bit devices and 32-bit devices can be used together.

Remarks:

Relevant special registers

D1712	Offset value of arc radius (low word)
D1713	Offset value of arc radius (high word)

6 Motion Instructions & G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
				20PM	-
16	CANC	N/A	Cancel Compensation		

Explanations:

1. You can place an M-Code instruction after CANC.
2. When CANC instruction is executed, all motion compensations (offset values) will be cancelled, i.e. special registers D1708 ~ D1709, D1724 ~ D1725, D1710 ~ D1711, D1726 ~ D1727, and D1712 ~ D1713 will all be cleared automatically.

Remarks:

D1708	Offset value for target position of X-axis (low word)
D1709	Offset value for target position of X-axis (high word)
D1710	Offset for distance to center of arc on X-axis (low word)
D1711	Offset for distance to center of arc on X-axis (high word)
D1712	Offset value of arc radius (low word)
D1713	Offset value of arc radius (high word)
D1724	Offset value for target position of Y-axis (low word)
D1725	Offset value for target position of Y-axis (high word)
D1726	Offset for distance to center of arc on Y-axis (low word)
D1727	Offset for distance to center of arc on Y-axis (high word)

6 Motion Instructions and G-Code Instructions

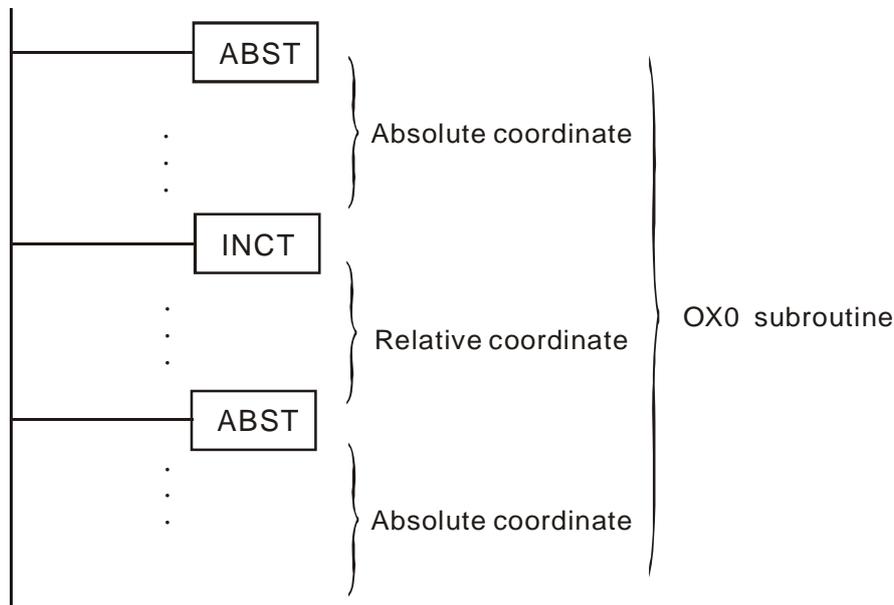
MON	Mnemonic	Operands	Function	Controllers	
17	ABST	N/A	Set up Absolute Coordinate	20PM	-
MON	Mnemonic	Operands	Function	Controllers	
18	INCT	N/A	Set up Relative Coordinate	20PM	-

Explanations:

1. ABST enables absolute coordinate system. Absolute coordinate is the coordinate starting from "0." When the target position $P(I) >$ current position (DD1848, DD1928, DD2008), the motor will conduct forward running. When the target position $P(I) <$ current position, the motor will conduct reverse running.
2. INCT enables relative coordinate system. Relative coordinate is the relative displacement moved from current position (DD1848, DD1928, DD2008). When the relative coordinate is a positive value, the motor will conduct forward running. When the relative coordinate is a negative value, the motor will conduct reverse running.
3. Even if absolute coordinate system is adopted, the distance to center of arc (I/J), the arc radius (R), and the additional distance set by SINTR and DINTR instructions are all regarded as relative displacement.

Program Example:

When DVP-PM switches from MANU to AUTO without designating coordinate system, the default setting for the program will be absolute coordinate system (ABST). After INCT instruction is executed, the motion instructions starting from the next row (e.g. DRV, LIN, CW, CCW) will be operated in relative coordinate system as below.



6 Motion Instructions & G-Code Instructions

MON	Mnemonic	Operands	Function	Controllers	
				20PM	-
19	SETT	X (P ₁) Y (P ₂) Z (P ₃)	Set up Current Position		

Type	Bit Devices			Double-Word Devices			Notes
	K	H	D	KK	HH	DD	
P ₁	*	*	*	*	*	*	<ul style="list-style-type: none"> SETT instruction supports V, Z index register modification on the devices. See specifications of DVP-PM for the applicable range of each device. You can place an M-Code instruction after SETT.
P ₂	*	*	*	*	*	*	
P ₃	*	*	*	*	*	*	

Operands:

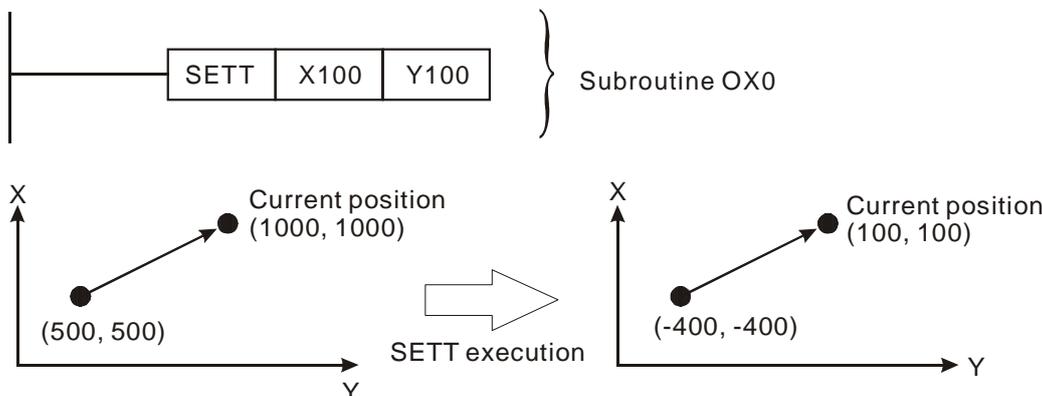
P₁: Current position on X axis P₂: Current position on Y axis P₃: Current position on Z axis

Explanations:

- When SETT instruction is executed, the current position set in SETT instruction will be written automatically into current position registers: D1848 ~ D1849 for X axis; D1928 ~ D1929 for Y axis and D2008~D2009 for Z axis.
- Range of parameters: (16-bit) K = -32,768 ~ 32,767; H = 0 ~ FFFF; D = 0 ~ 9,999; (32-bit) KK = -2,147,483,648 ~ 2,147,483,647; HH = 0 ~ FFFFFFFF; DD = 0 ~ 9,998.
- The 16-bit devices and 32-bit devices can be used together.
- Due to that current position is changed when SETT instruction is executed, the mechanical zero point and electrical zero point will also be changed.
- At least one current position on one axis should be specified, therefore there are 7 operand combinations for SETT instruction.

NO.	Instruction	Operand combination	20D	20M
1	SETT	X (P ₁)	✓	✓
2		Y (P ₂)	✓	✓
3		X (P ₁) Y (P ₂)	✓	✓
4		Z (P ₃)	-	✓
5		X (P ₁) Z (P ₃)	-	✓
6		Y (P ₂) Z (P ₃)	-	✓
7		X (P ₁) Y (P ₂) Z (P ₃)	-	✓

Program Example:



6 Motion Instructions and G-Code Instructions

Remarks:

Relevant special registers

D1848	Current position of X axis: CP (low word)
D1849	Current position of X axis: CP (high word)
D1928	Current position of Y axis: CP (low word)
D1929	Current position of Y axis: CP (high word)
D2008	Current position of Z axis: CP (low word)
D2009	Current position of Z axis: CP (high word)

6 Motion Instructions & G-Code Instructions

6.4 O Pointer / M Pointer

Mnemonic	Function	Controllers	
O	Pointer for Main Program and Motion Subroutine	20PM	10PM
Explanation	Main program pointer: O100 Motion subroutine pointer: OX0 ~ OX99		

Explanations:

- O100 is the start pointer of main control program. You need the main control program to activate OX0 ~ OX99 motion subroutines. O100 main control program ends when M102 instruction is executed.
- OX0 ~ OX99 are the pointers for the 100 motion control subroutines and can be applied by the programming engineer for different motion routes. Register D1868 (CR72) stores the No. of subroutine, in which b14 or b15 has to be "1." In addition, activate b12 of D1846 (CR50) or set M1074 = ON, and the motion subroutine will be enabled. See the example below.

Example: 2 steps to activate OX99 motion subroutine:

- Set up the No. to be activated: D1868 = H'4063 (or H'8063, H'C063)
 - Enable OX99: D1846 = H'1000 or M1074 = ON
- M2 is the instruction to end OX0 ~ OX99 motion subroutines.

Program Example:

In the program example below, N0000 ~ N0100 are O100 main control program; N0102 ~ N0304 are OX50 motion subroutine.

No. of row	Instruction List
N0000	O100
N0001	LD M1000
N0002	MOV H8063 D1868
N0007	OUT M1074
:	:
:	:
N0099	OUT Y30
N0100	M102
N0101	N
N0102	OX50
N0103	DRVZ
N0104	ABS
:	:
:	:
N0304	M2

6 Motion Instructions and G-Code Instructions

Mnemonic	Function		
M	M-code Instructions	Controllers	
		20PM	-
Explanation	M0 ~ M65535 M102: End of O100 main program M2: End of OX0 ~ OX99 motion subroutine		

Explanations:

1. M-code instruction is one type of motion instruction. To execute M-code, first store the No. of M-code into D1703. When M-code is enabled, M1794 will be "ON" automatically. If M1744 is set "ON", M1794 will become "OFF", indicating that the execution of M-code is completed.
2. Y output can be enabled by executing M-code. The high byte of D1873 is used to enable Y output, and the low byte is used to designate the start No. of Y output. When M1794 is "ON" (i.e. M-code is being executed), the Y output terminals designated by D1873 will be "ON". When M1794 is "OFF", the Y output terminals will be "OFF". See Program Example 1.
3. M-code generally is used in the sections of OX00 ~ OX00 subroutines.
4. There are two modes for using M-code instruction: "after" mode and "with" mode. The difference between the two modes is the timing of executing M-code instruction. See Program Example 2.
5. When the execution of M-code is completed, M1794 will turn from ON to OFF in two ways:
 - (1) Set M1794 to be "OFF" directly to reset the status.
 - (2) Set M1744 to be "ON" directly.

Note: There are two special M-codes exclusively for stopping O100 main program (M102) and OX0~OX99 motion subroutine (M2). Users should avoid the 2 M-codes when applying M-codes for motion subroutines.

Program Example 1:

1. Display the current No. of M-code being executed, e.g. M6, by status of Y terminals:
 - (1) First set the parameter in D1873 as below:

```
MOV H0□□□ D1873
      |
      |→ N: Start No. of Y output
      |→ 0: Disable
      |→ 1: Enable
```

- (2) Execute M-code (M6). In this case, DVP-PM will automatically write K6 (0110) into D1703 (The executing M code) and map the value in D1703 into K2Y_N. N is the start No. of Y output.

```
MOV H6 D1703
MOV D1703 K2YN
```

When M-code is enabled, the above instructions will run automatically. Therefore, you do not need to enter the two MOV instructions manually.

- (3) The number of M-code (M6) will be displayed by Y outputs according to the settings of N value in D1873:

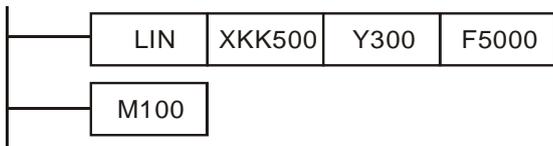
D1873	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
H00□□	No Y output							
H0100	0	0	0	0	0	1	1	0

6 Motion Instructions & G-Code Instructions

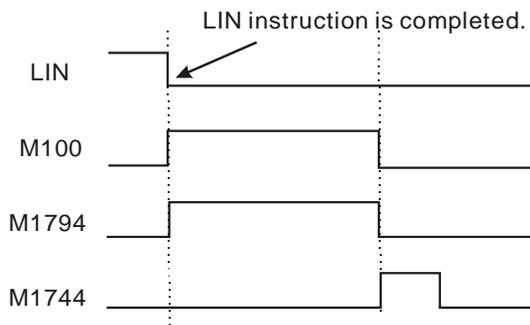
H0101	0	0	0	0	1	1	0	-
H0102	0	0	0	1	1	0	-	-
H0103	0	0	1	1	0	-	-	-
.								
.								
.								

Program Example 2:

1. "After" mode: connect a single M-code instruction to left-hand bus bar after a motion instruction.

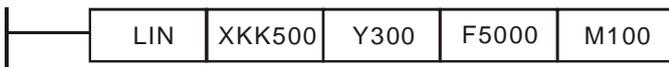


Timing diagram:

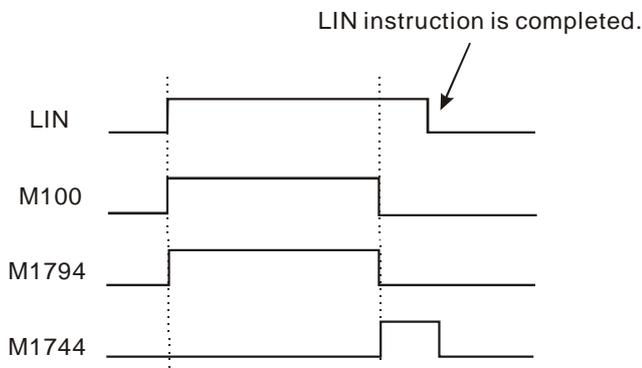


When LIN instruction is completed, M-code (M100) will be enabled automatically and M1794 will be ON as well. To stop M100 subroutine, set M1744 = ON. In addition, M-code can be re-activated by other instructions.

2. "With" mode: Place an M-code instruction at the end of a motion instruction.



Timing diagram:



When LIN instruction is activated, M-code (M100) will be automatically enabled at the same time and M1794 will be ON as well. To stop M100 subroutine, set M1744 = ON. If you need to re-activate M-code in "with" mode, you have to wait until LIN instruction is finished then modify the parameter to re-execute the desired M-code instruction.

6 Motion Instructions and G-Code Instructions

Program Example 3:

The instruction list below shows the usages of special M-codes, "with" mode and "after" mode. N0100 and N0301 are special M-codes for stopping O100 and OX subroutine. N0105 indicates "with" mode; N0250 indicates "after" mode.

No. of row	Program
N000	O100
N001	LD M1000
:	:
:	:
N0099	OUT Y30
N0100	M102
N0101	N
N0102	OX50
N0103	DRVZ
N0104	ABS
N0105	DRV XD10 FXD12 M20
:	:
N0250	M08
:	:
N0304	M2

6 Motion Instructions & G-Code Instructions

6.5 G-Code Instructions

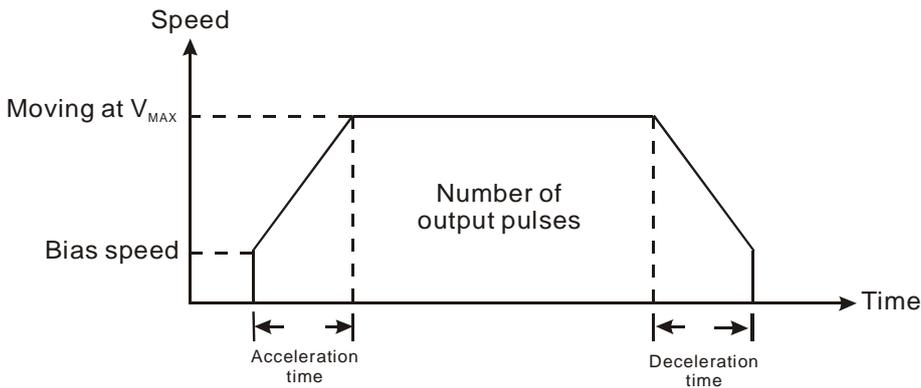
G-Code	Mnemonic	Operands	Function	Controllers	
	G00	X (P ₁) Y (P ₂) Z (P ₃)	3-axis High-speed Positioning	20PM	-

Operands:

P₁: Target position on X axis **P₂**: Target position on Y axis **P₃**: Target position on Z axis (built-in 3rd axis)

Explanations:

- For 20D model, the target position on Z axis (**P₃**) is operated by the built-in 3rd axis control. Please refer to p6-46.
- Range of parameters: -2,147,483,648 ~ 2,147,483,647 (without decimal point); -2,147,483.648 ~ 2,147,483.647 (with decimal point)
- The operand parameters for 20M model can also be 16-bit register (D) or 32-bit register (DD).
- For relevant special registers, see Remarks of MON 00 DRV.
- The moving speed of G00 instruction is fixed at maximum speed V_{MAX} .
- The settings of position have continuity. See Remarks.
- Acceleration/deceleration time and bias speed can be set up in special D.
- Acceleration/deceleration time and bias speed increase or decrease in proportional to the setting of V_{MAX} .
- Timing diagram:



- 20D model does not support 3-axis positioning control. For 3rd axis control, an additional single-axis positioning module DVP-01PU is required. Therefore, you have to design a 2-axis high-speed positioning in X-Y axis and independent high-speed positioning for Z axis. For the safety of the mechanical operation, when G00 instruction is executed, Z-axis high-speed positioning will be executed first before the X-Y-axis high-speed positioning. That is to say, when DVP-PM is executing G00 instruction with X-Z, Y-Z, X-Y-Z combinations, the program will automatically be divided as:

G00 Z P₃ (A)

G00 X P₁ Y P₂ (B)

See Remarks for more explanations on row (A) and (B)

Remarks:

- The settings of position have continuity, for example:

G00 X500.0 Y500.0 Z500.0;

X1000.0 Y1000.0 Z1000.0;

After the row with G00 instruction is executed, the program will execute the next row. The second row of the

6 *Motion Instructions and G-Code Instructions*

program will reach the target position automatically by G00.

2. For 20D model, when G00 adopts Z-axis target position (built-in 3rd axis control) as:

```
G00 X1000 Y1000 Z100;
```

The instruction will automatically be compiled as:

```
G00 Z100; ... (A)
```

```
G00 X1000 Y1000; ... (B)
```

(A) is first executed, and at this time the operation fast moves to position K100 on Z axis. (B) is next executed and the operation moves to target position (1000, 1000) at the maximum speed.

6 Motion Instructions & G-Code Instructions

G-Code	Mnemonic	Operands	Function	Controllers	
	G00	Z (P ₃)	3-axis High-speed Positioning (3 rd axis control)	20PM	-

Explanations:

- 20D model does not support 3-axis positioning control. For 3rd axis control, an additional single-axis positioning module DVP-01PU is required
- For 20D model, when G00 sets Z-axis target position (built-in 3rd axis control), the 3rd axis control will be independently executed first.

For example:

G00 X1000 Y1000 Z100

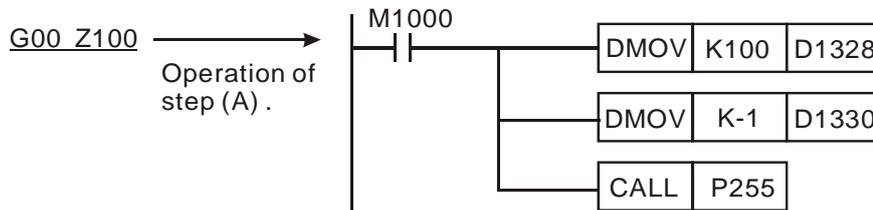
↓

G00 Z100 (A)

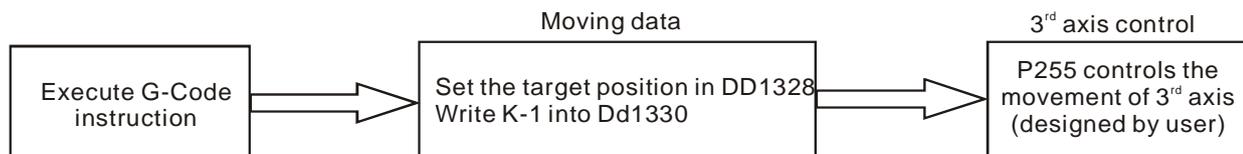
G00 X1000 Y1000 (B)

In step (A), DVP-PM writes target position K100 on Z axis into registers D1328, D1329. The moving speed of G00 instruction is fixed at maximum speed V_{MAX}. Therefore, PLC writes K-1 into D1330, D1331 (K-1 refers to positioning at max speed). After this, call and execute P255 subroutine.

Operation of step (A):



P255 is a subroutine for controlling the 3rd axis (e.g. pen lifting, clipping and release, and so on). When the target position is achieved by the control data set in DD1328, DD1330, the operation in P255 will be executed.



Because D1328~D1331, and P255 could be occupied by 3rd axis control, care should be taken on using these registers, and we suggest you avoid using these registers and also P255 repeatedly.

G-Code	Mnemonic	Operands	Function	Controllers	
	G00	X (P ₁) Y (P ₂)	2-axis High-speed Positioning	20PM	-

Explanations:

The instruction is mainly for high-speed positioning on 20D model without specifying Z-axis. See Remarks of MON 00 DRV for usages and relevant special registers.

6 Motion Instructions and G-Code Instructions

G-Code	Mnemonic	Operands	Function	Controllers	
	G01	X (P ₁) Y (P ₂) Z (P ₃) F (V)	3-axis Linear Interpolation (considering remaining distance)	20PM	-

Operands:

P₁: Target position on X axis **P₂**: Target position on Y axis **P₃**: Target position on Z axis

V: Speed for linear interpolation

Explanations:

- For 20D model, the target position on Z axis (**P₃**) is controlled by the built-in 3rd axis. Please refer to p6-49.
- Range of **P₁**, **P₂**, **P₃**: -2,147,483,648 ~ 2,147,483,647 (without decimal point); -2,147,483.648 ~ 2,147,483.647 (with decimal point)
- Range of **V**: 0 ~ 500,000 (without decimal point); 0 ~ 500.0 (with decimal point)
- The operand parameters for 20M model can also be 16-bit register (D) and 32-bit register (DD).
- The speed for linear interpolation (**V**) has continuity. See Remarks.
- For positioning process, see explanations of MON 01 LIN instruction.
- 20D model does not support 3-axis synchronous interpolation. For 3rd axis control, an additional single-axis positioning module DVP-01PU is required. Therefore, you have to design a 2-axis high-speed interpolation in X-Y axis and independent high-speed positioning for Z axis. For the safety of the mechanical operation, when G01 instruction is executed, Z-axis movement will be executed first before the X-Y-axis interpolation. That is to say, when DVP-PM is executing G01 instruction with X-Z, Y-Z, X-Y-Z combinations, the program will automatically be divided as:

G01 Z **P₃** **FV** (A)

G01 X **P₁** Y **P₂** **FV** (B)

Please refer to p6-49 for more explanations on row (A) and (B)

Remarks:

- The settings of speed have continuity, for example:

G01 X100 Y100 Z400 F200;

X200 Y200;

After the row with G01 instruction is executed, the program will execute the next row. The second row of the program will reach the target position automatically by speed F200 set in the first row.

- For 20D model, when G01 adopts Z-axis target position (built-in 3rd axis control) as:

G01 X1000 Y1000 **Z100** F200;

The instruction will automatically be compiled as:

G01 **Z100** F200; ... (A)

G01 X1000 Y1000 F200; ... (B)

(A) is first executed, and at this time the operation on Z-axis fast moves to target position K100 at speed K200.

(B) is next executed and the operation on X-Y-axis moves to target position (1000, 1000) at speed K200.

6 Motion Instructions & G-Code Instructions

G-Code	Mnemonic	Operands	Function	Controllers	
	G01	Z (P ₃) F (V)	3-axis Linear Interpolation (3 rd Axis Control)	20PM	-

Explanations:

- 20D model does not support 3-axis synchronous interpolation. For 3rd axis control, an additional single-axis positioning module DVP-01PU is required.
- For 20D model, when G01 sets Z-axis target position (built-in 3rd axis control), the 3rd axis control will be independently executed first.

For example:

G01 X1000 Y1000 Z100 F200

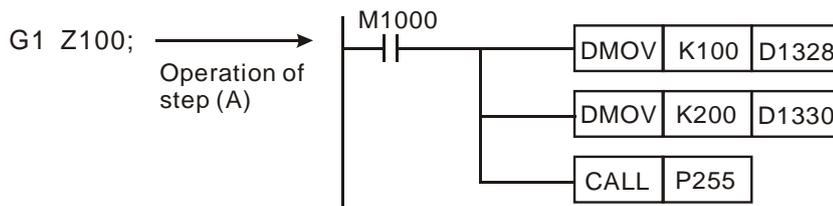


G01 Z100 F200 (A)

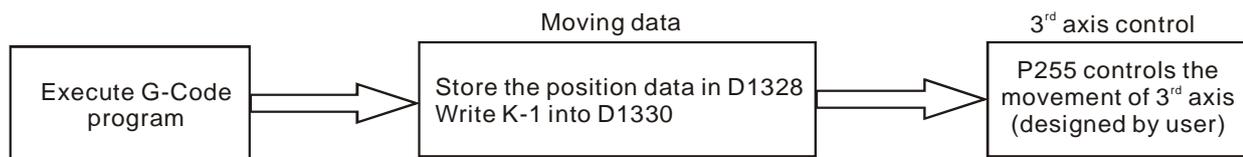
G01 X1000 Y1000 (B)

In step (A), DVP-PM writes target position K100 on Z axis into registers D1328, D1329. The moving speed of G01 instruction K200 is written into D1330, D1331. After this, call and execute P255 subroutine.

Operation of step (A):



P255 is a subroutine for controlling the 3rd axis (e.g. pen lifting, clipping and release, and so on). When the target position is achieved by the control data set in DD1328, DD1330, the operation in P255 will be executed.



Because D1328~D1331, and P255 could be occupied by 3rd axis control, care should be taken on using these registers, and we suggest you avoid using these registers and also P255 repeatedly.

G-Code	Mnemonic	Operands	Function	Controllers	
	G01	X (P ₁) Y (P ₂) F (V)	2-Axis Synchronous Linear Interpolation (considering remaining distance)	20PM	-

Explanations:

The instruction is mainly for linear interpolation on 20D model without specifying Z-axis. See Remarks of MON 01 LIN for usages and relevant special registers.

6 Motion Instructions and G-Code Instructions

G-Code	Mnemonic	Operands	Function	Controllers	
	G02 G03	X (P ₁) Y (P ₂) Z (P ₃) I (P ₄) J (P ₅) K (P ₆) F (V)	Clockwise Arc / Clockwise Helical / Counterclockwise Arc / Counterclockwise Helical Movement (distance to center of arc)	20PM	-

Operands:

P₁: Target position on X axis **P₂**: Target position on Y axis **P₃**: Target position on Z axis

P₄: Distance to the center of arc/helix on X axis **P₅**: Distance to the center of arc/helix on Y axis

P₆: Distance to the center of arc/helix on Z axis **V**: Arc/helical interpolation speed

Explanations:

- P₄, P₅, P₆**: The vector-valued coordinate of center of arc or center of helix, i.e. distance to the center of arc/helix on X-Y-Z axes
- Range of parameters **P₁, P₂, P₃, P₄, P₅, P₆**: -2,147,483,648 ~ 2,147,483,647 (without decimal point);
-2,147,483.648 ~ 2,147,483.647 (with decimal point)
- Range of **V**: 0 ~ 500,000 (without decimal point); 0 ~ 500.0 (with decimal point)
- The operand parameters for 20M model can also be 16-bit register (D) and 32-bit register (DD).
- The settings of speed has continuity. See Remarks.
- For detailed explanation of the positioning process, see MON 02 CW and MON 03 CCW.

Remarks:

- The settings of speed have continuity, for example:
G02 X0.0 Y100.0 I0.0 J50.0 F100.0;
G02 X0.0 Y0.0 I0.0 J50.0;
- After the row with G02 instruction is executed, the program will execute the next row. The second row of the program will reach the target position automatically by speed F100 set in the first row.

6 Motion Instructions & G-Code Instructions

G-Code	Mnemonic	Operands	Function	Controllers	
	G02 G03	X (P ₁) Y (P ₂) Z (P ₃) R (L) F (V)	Clockwise Arc / Clockwise Helical / Counterclockwise Arc / Counterclockwise Helical Movement (radius)	20PM	-

Operands:

P₁: Target position on X axis **P₂:** Target position on Y axis **P₃:** Target position on Z axis

L: Radius of arc / helix (R = "+" when radian < 180°; R = "-" when radian > 180°)

V: Arc/helical interpolation speed

Explanations:

1. Range of parameters **P₁, P₂, P₃, R:** -2,147,483,648 ~ 2,147,483,647 (without decimal point); -2,147,483.648 ~ 2,147,483.647 (with decimal point)
2. Range of **V:** 0 ~ 500,000 (without decimal point); 0 ~ 500.0 (with decimal point)
3. The operand parameters for 20M model can also be 16-bit register (D) and 32-bit register (DD).
4. For detailed explanation of the positioning process, see MON 04 CW and MON 05 CCW.

6 Motion Instructions and G-Code Instructions

G-Code	Mnemonic	Operands	Function	Controllers	
	G04	X(<u>T</u>) / P(<u>T</u>)	Pause Time	20PM	-

Operands:

XT: Pause time (unit: 1 sec). G4 X1 refers to pausing for 1 second; G4 X2.5 refers to pausing for 2.5 seconds.

PT: Pause time (unit: 1 ms). G4 P100 refers to pausing for 0.1 second; G4 P4500 refers to pausing for 4.5 seconds.

Explanations:

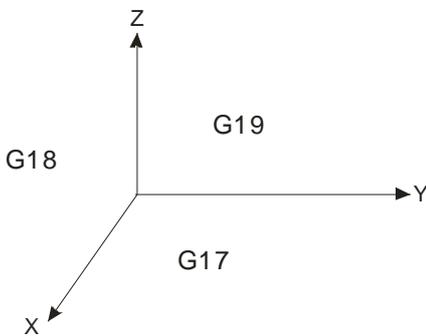
1. 10ms is the minimum setting for **PT**. PT value smaller than 10ms will be regarded as 0ms, i.e. if **PT** = 23ms, it will be regarded as 20ms.
2. The operand parameters for 20M model can also be 16-bit register (D) and 32-bit register (DD).
3. For detailed explanation of the instruction, see MON 06 TIM.

6 Motion Instructions & G-Code Instructions

G-Code	Mnemonic	Operands	Function	Controllers	
	G17	N/A	Select X-Y Plane	20PM	-
	G18		Select X-Z Plane		
	G19		Select Y-Z Plane		

Explanations:

1. The three instructions decide the work planes for arc interpolation and helical interpolation and have no effects on linear interpolation.
2. The three instructions can be switched to each other during the execution of instruction. If no work plane is set up, X-Y plane (G17) will be automatically selected as the work plane..
3. Work planes:



G-Code	Mnemonic	Operands	Function	Controllers	
	G90	N/A	Set up Absolute Coordinate	20PM	-

See MON 17 ABS for the operation of G90

G-Code	Mnemonic	Operands	Function	Controllers	
	G91	N/A	Set up Relative Coordinate	20PM	-

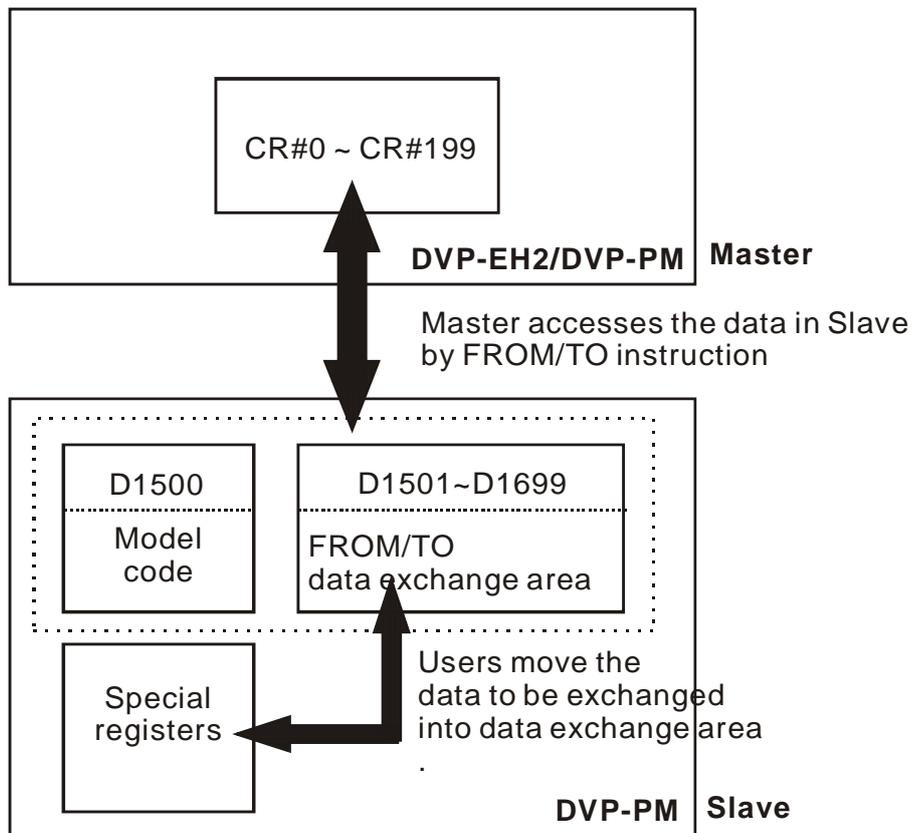
See MON 18 INC for the operation of G91

7.1 Access between DVP-EH2, DVP-PM (as Master) and DVP-PM (as Slave)

When DVP-PM is used as Slave, there is a data exchange area in DVP-PM which corresponds to the control registers (CRs) in the Master. The data exchange area is consisted of consecutive special registers, and users can utilize the data exchange area for accessing data between Master and Slave as well as performing motion control functions through a Slave DVP-PM.

7.1.1 The Structure

- ◆ DVP-EH2 and Master DVP-PM apply FROM/TO instructions to transmit commands to Slave DVP-PM for executing motion control subroutines.
- ◆ DVP-EH2 and Master DVP-PM apply FROM/TO instructions to access the control registers (CR#0 ~ CR#199, corresponding to special registers D1500 ~ D1699 in the Slave) in Slave DVP-PM.



7 Use DVP-PM As Slave

7.1.2 Example of Master-Slave Data Exchange

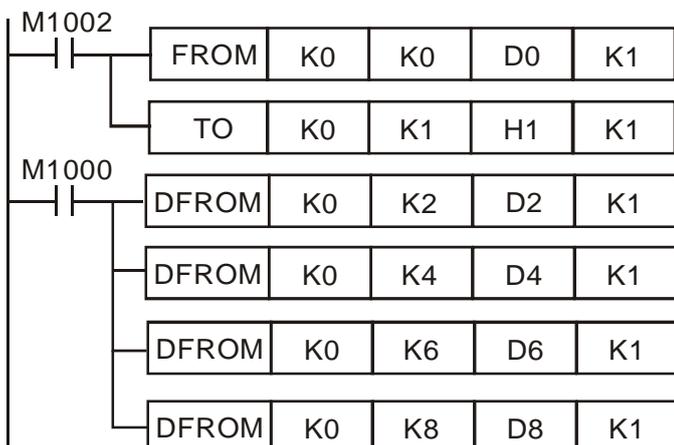
- ◆ Set up: design the data exchange programs in Master and Slave respectively.
 - Slave DVP-PM: Move the data to be accessed by Master into the data exchange area through MOV instruction
 - Master: Plan the CRs (on Slave) for Master to access.
- ◆ Example 1
 - Control Purpose:
DVP-EH2 applies FROM/TO instructions to access special registers D1500 ~ D1699 in Slave DVP-PM and executes manual mode position control on X/Y axis. (For relative registers see 3.12.3)
 - Table for CRs in the Master and corresponding special registers in the Slave:

Master	Slave		Content
	Planned internally	Planned by user	
CR#0	D1500	Set up by the system	Model code of Slave DVP-PM
CR#1	D1501	D1846	Operation commands for X axis
CR#2 ~ 3	D1502 ~ D1503	D1848 ~ D1849	Current position of X axis CP (Pulse)
CR#4 ~ 5	D1504 ~ D1505	D1850 ~ D1851	Current speed of X axis (PPS)
CR#6 ~ 7	D1506 ~ D1507	D1860 ~ D1861	MPG input frequency on X axis
CR#8 ~ 9	D1508 ~ D1509	D1862 ~ D1863	Accumulated number of MPG input pulses at X axis

1. If you need to use other functions of Slave DVP-PM, please refer to Chapter 3 and fill the relative special registers into the “Planned by user” column. After this you can add the relative registers to the example program, so that the desired motion control functions can be performed.
2. D1500 ~ D1699 are the special registers planned internally in the Slave, among which D1500 is the read-only register for storing the model code of DVP-PM (H'6260). Therefore, D1501 ~ D1699 are the applicable registers.

Program in Master DVP-EH2

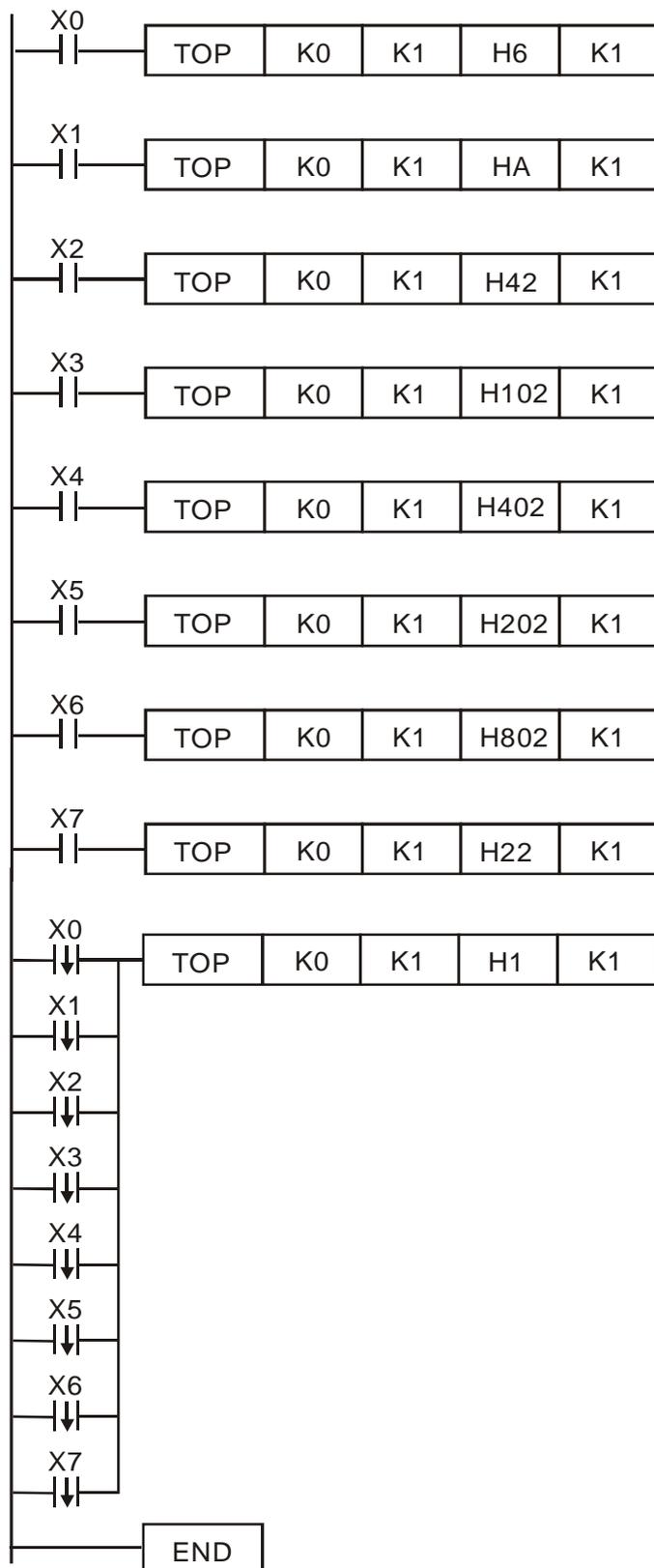
Ladder diagram:



Operation:

- When Master DVP-EH2 is in RUN, read out CR#0 of Slave, corresponding to D1500 in Slave.
- Write in CR#1 of Slave, corresponding to D1501 in Slave, to STOP X axis in Slave.
- Read out CR#2 of Slave, corresponding to D1502 ~ D1503 in Slave.
- Read out CR#4 of Slave, corresponding to D1504 ~ D1505 in Slave.
- Read out CR#6 of Slave, corresponding to D1506 ~ D1507.
- Read out CR#8 of Slave, corresponding to D1508 ~ D1509.

7 Use DVP-PM As Slave



When X0 = ON, write in CR#1 of Slave, corresponding to D1501 in Slave, to enable JOG+ operation on X axis in Slave.

When X1 = ON, write in CR#1 of Slave, corresponding to D1501 in Slave, to enable JOG- operation on X axis in Slave.

When X2 = ON, write in CR#1 of Slave, corresponding to D1501 in Slave, to enable zero return on X axis in Slave.

When X3 = ON, write in CR#1 of Slave, corresponding to D1501 in Slave, to enable single-speed positioning on X axis in Slave.

When X4 = ON, write in CR#1 of Slave, corresponding to D1501 in Slave, to enable 2-speed positioning on X axis in Slave.

When X5 = ON, write in CR#1 of Slave, corresponding to D1501 in Slave, to enable single-speed positioning with additional distance on X axis in Slave.

When X6 = ON, write in CR#1 of Slave, corresponding to D1501 in Slave, to enable 2-speed positioning with additional distance on X axis in Slave.

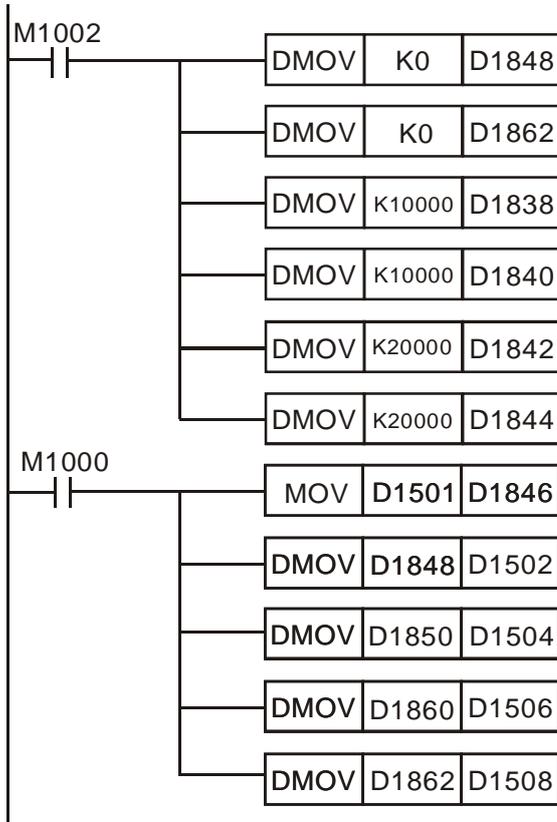
When X7 = ON, write in CR#1 of Slave, corresponding to D1501 in Slave, to enable MPG input operation of X axis in Slave.

When X0 ~ X7 = OFF, write in CR#1 of Slave, corresponding to D1501 in Slave, to STOP X axis in Slave.

7 Use DVP-PM As Slave

Program in DVP-PM Slave

Ladder diagram:



Operation:

Enable O100 in Slave, and clear the current position of X axis as "0".

Clear the number of accumulated MPG pulses of X axis as "0".

Set up the target position (I) of X axis P(I)

Set up the operation speed (I) of X axis V(I)

Set up the target position (II) of X axis P(II)

Set up the operation speed (II) of X axis V(II)

Move D1501, corresponding to CR#1, to X axis for setting up operation commands.

Move the current position of X axis D1848 ~ D1849 to D1502 ~ D1503, corresponding to CR#2 ~ CR#3.

Move the current speed of X axis D1850 ~ D1851 to D1504 ~ D1505, corresponding to CR#4 ~ CR#5.

Move MPG input frequency of X axis D1860 ~ D1861 to D1502 ~ D1503, corresponding to CR#6 ~ CR#7.

Move the number of MPG pulses of X axis D1862 ~ D1863 to D1508 ~ D1509, corresponding to CR#8 ~ CR#9.

◆ Example 2

- Control Purpose:

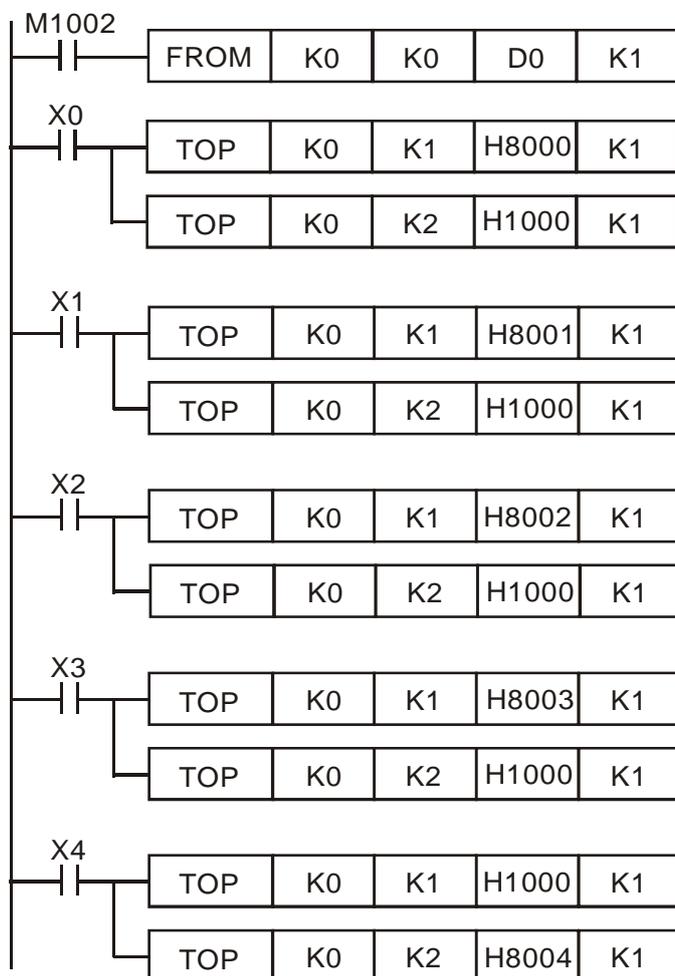
DVP-EH2 applies FROM/TO instructions to access special registers D1500 ~ D1699 in Slave DVP-PM and executes motion instructions in OX subroutines (see Chapter 6 for how to use motion instructions).

- Table for CRs in the Master and corresponding special registers in the Slave:

Master	Slave		Content
	Planned internally	Planned by user	
CR#0	D1500	-	Model code of DVP-PM Slave
CR#1	D1501	D1868	No. of OX subroutine
CR#2	D1502	D1846	Operation commands for X axis (OX)

Program in DVP-EH2 Master

Ladder diagram:



Operation:

When DVP-EH2 Master is in RUN, read out CR#0 of Slave, corresponding to D1500 in Slave.

Write in CR#1 of Slave, corresponding to D1501 in Slave, to enable OX00 and execute DRV instruction in Slave.

Write in CR#2 of Slave, corresponding to D1502 in Slave, to enable OX subroutine in Slave.

Write in CR#1 of Slave, corresponding to D1501 in Slave, to enable OX01 and execute LIN instruction in Slave.

Write in CR#2 of Slave, corresponding to D1502 in Slave, to enable OX subroutine in Slave.

Write in CR#1 of Slave, corresponding to D1501 in Slave, to enable OX02 and execute CW instruction in Slave.

Write in CR#2 of Slave, corresponding to D1502 in Slave, to enable OX subroutine in Slave.

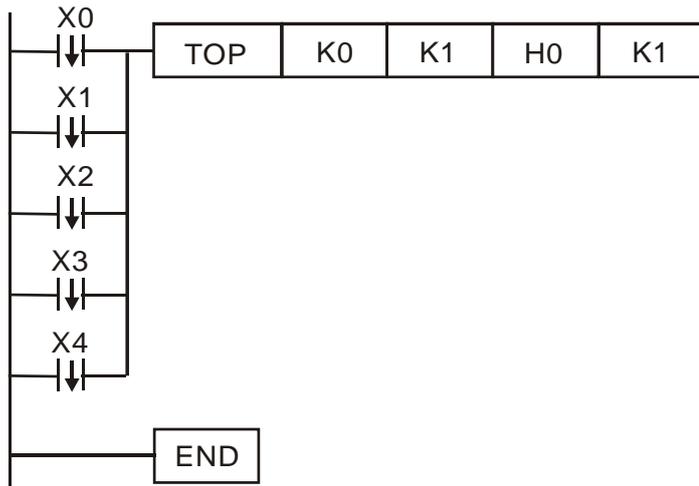
Write in CR#1 of Slave, corresponding to D1501 in Slave, to enable OX03 and execute CCW instruction in Slave.

Write in CR#2 of Slave, corresponding to D1502 in Slave, to enable OX subroutine in Slave.

Write in CR#1 of Slave, corresponding to D1501 in Slave, to enable OX04 and execute DRVZ instruction in Slave.

Write in CR#2 of Slave, corresponding to D1502 in Slave, to enable OX subroutine in Slave.

7 Use DVP-PM As Slave



When X0 ~ X4 = OFF, write in CR#1 of Slave, corresponding to D1501 in Slave, to disable OX subroutine in Slave.

Program in DVP-PM Slave

Instruction mode:

```
O100
LD M1002
DMOV K0 D1848
DMOV K0 D1928
LD M1000
MOV D1501 D1868
MOV D1502 D1846
M102
```

```
OX00
DRV X200000 FX100000 Y200000 FY100000
M2
```

```
OX01
LIN X100000 Y100000 F200000
M2
```

```
OX02
CW X0 Y100000 IO J50000 F200000
M2
```

```
OX03
CCW X0 Y100000 IO J50000 F200000
M2
```

```
OX04
BRET
DMOV K200000 D1828
DMOV K100000 D1830
DMOV K200000 D1908
DMOV K100000 D1910
DRVZ
M2
```

Operation:

Place the instructions for initializing current position of X/Y axis in O100 main program. Enable O100 in Slave then call the motion subroutine.

Place motion instruction DRV in OX00 subroutine.

Place motion instruction LIN in OX01 subroutine.

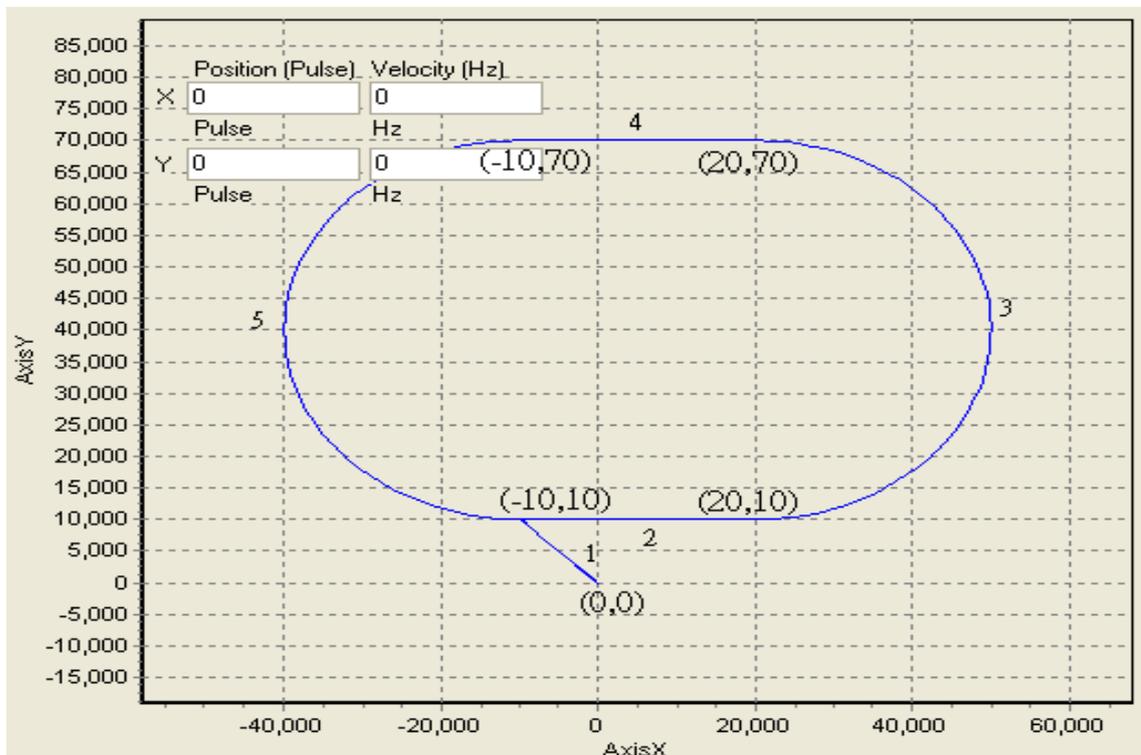
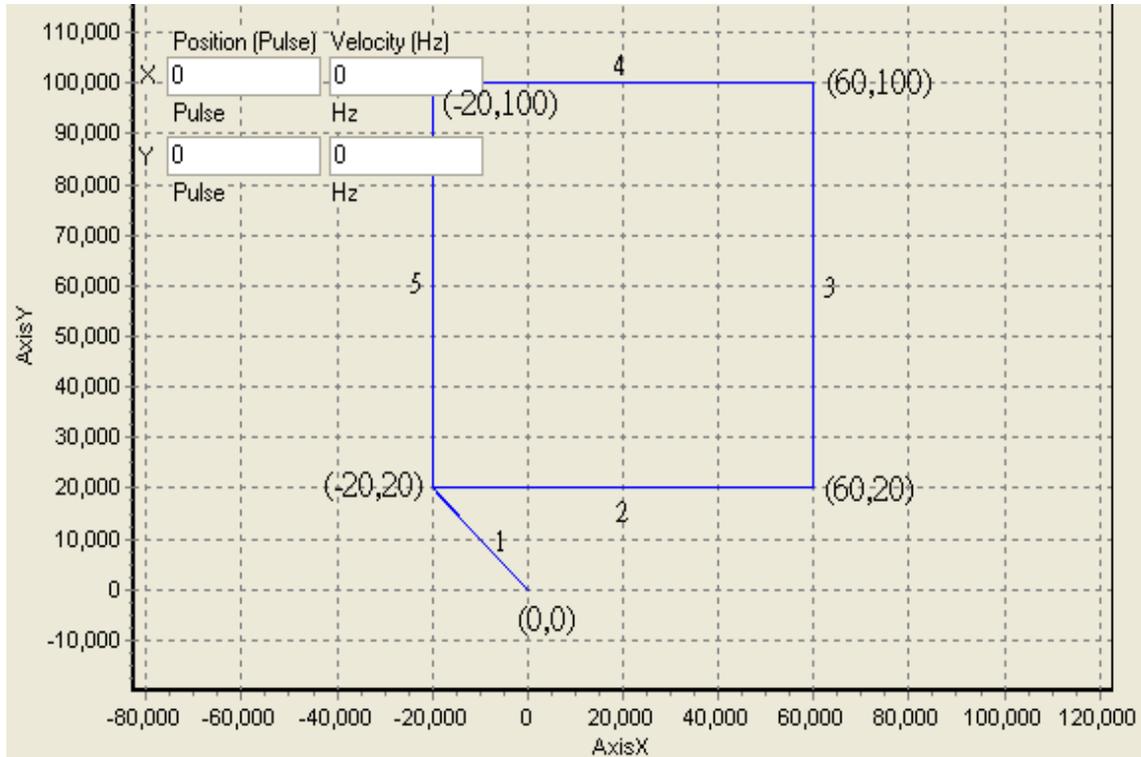
Place motion instruction CW in OX02 subroutine.

Place motion instruction CCW in OX03 subroutine.

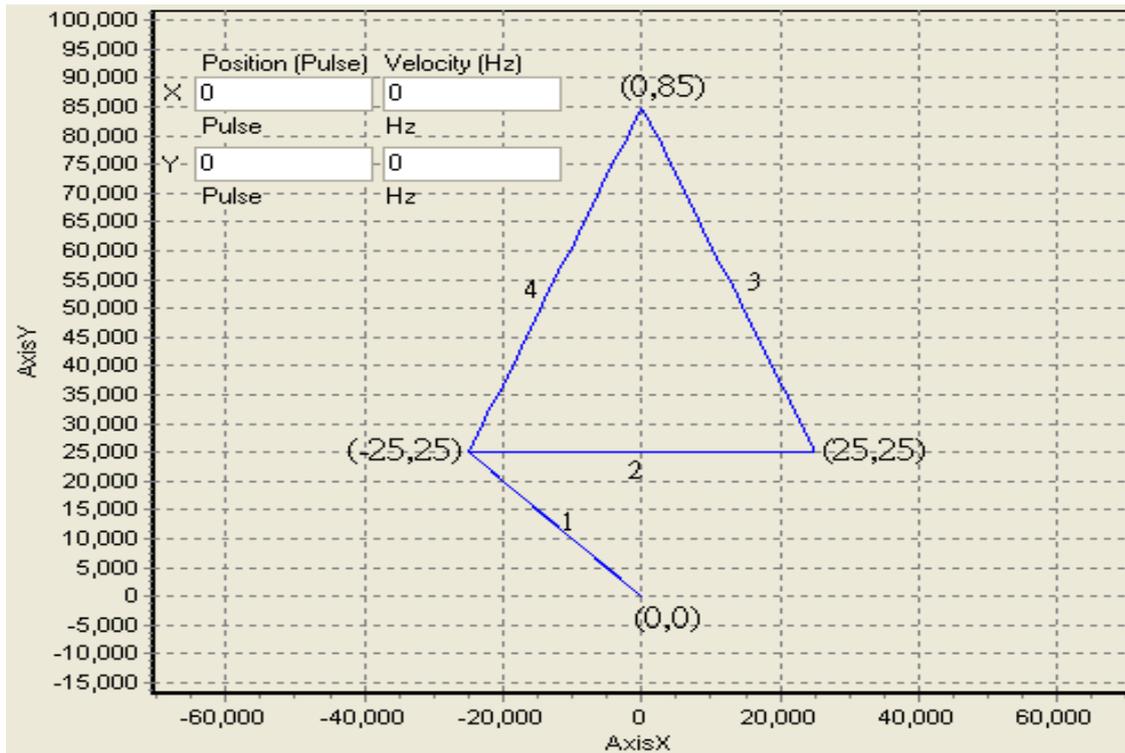
Place motion instruction DRVZ in OX04 subroutine, and set up relevant parameters for DRVZ.

8.1 Draw the Trajectories Below by Motion Instructions and G-Codes

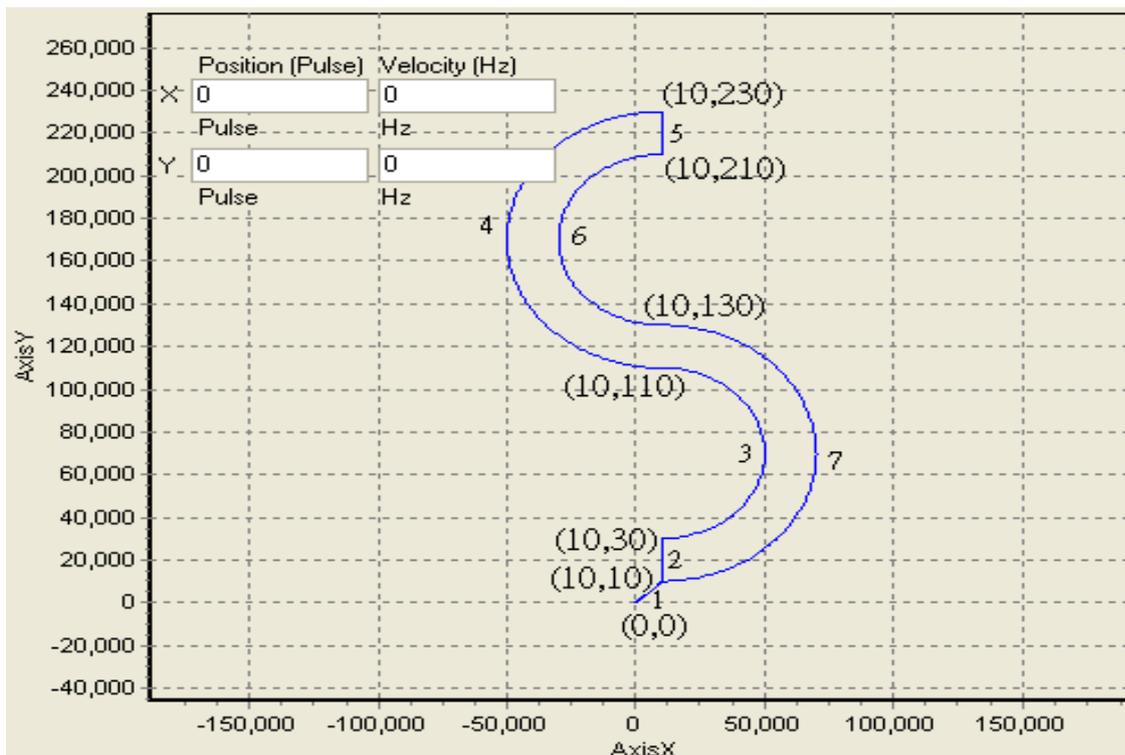
8.1.1 Trajectory



8 Application Examples



Trajectory 3



Trajectory 4

8.1.2 Design Procedure

1. Trajectory 1: Set up the absolute coordinates of the four points (-20, 20), (60, 20), (60, 100) and (-20, 100). Start from (0, 0).
2. Trajectory 2: Set up the absolute coordinates of the four points (-10, 10), (20, 10), (20, 70) and (-10, 70). Start from (0, 0).
3. Trajectory 3: Set up the absolute coordinates of the three points (-25, 25), (25, 25) and (0, 85). Start from (0, 0).
4. Trajectory 4: Set up the absolute coordinates of the seven points (10, 10), (10, 30), (10, 110), (10, 230), (10, 210), (10, 130) and (10, 10). Start from (0, 0).
5. Instruction List of Motion instructions or G-codes

O100 main program:

Design instructions for initializing current status of X/Y axis and enabling OX0 motion subroutine

O100			O100 main program
LD	M1002		
DMOV	K0	D1848	Set the current position of X axis to 0
DMOV	K0	D1928	Set the current position of Y axis to 0
RST	M1074		Disable OX motion subroutine
MOV	H8000	D1868	Write the No. (0) of OX to be enabled
SET	M1074		Enable OX motion subroutine
M102			

OX0 subroutine: Call pointer P0 in subroutine

OX0			OX motion subroutine
BRET			Return to bus line
CALL	P0		Call P0 subroutine
M2			

Example of motion instructions for trajectory 1

P0				P0 subroutine
ABST				Set-up absolute coordinate
DRV	X-20000	Y20000		Fast move to designated position
LIN	X60000	Y20000	F20000	Move to designated position by linear interpolation. Can also be written as LIN X60000 F20000
LIN	X60000	Y100000	F20000	Move to designated position by linear interpolation. Can also be written as LIN Y100000
LIN	X-20000	Y100000	F20000	Move to designated position by linear interpolation. Can also be written as LIN X-20000
LIN	X-20000	Y20000	F20000	Move to designated position by linear interpolation. Can also be written as LIN Y20000
SRET				

8 Application Examples

Example of G-code instructions for trajectory 1.

P0					P0 subroutine
G90					Set up absolute coordinate
G00	X-20.0	Y20.0			Fast move to designated position
G01	X60.0	Y20.0	F20.0		Move to designated position by linear interpolation. Can also be written as G01 X60.0 F20.0
G01	X60.0	Y100.0	F20.0		Move to designated position by linear interpolation. Can also be written as G01 Y100.0
G01	X-20.0	Y100.0	F20.0		Move to designated position by linear interpolation. Can also be written as G01 X-20.0
G01	X-20.0	Y20.0	F20.0		Move to designated position by linear interpolation. Can also be written as G01 Y20.0
SRET					

Example of motion instructions for trajectory 2

P0					P0 subroutine
ABST					Set up absolute coordinate
DRV	X-10000	Y10000			Fast move to designated position
LIN	X20000	Y10000	F40000		Move to designated position by linear interpolation. Can also be written as LIN X20000 F40000
CCW	X20000	Y70000	J30000	F20000	Move to designated position by arc interpolation. Can also be written as CCW Y70000 J30000 F20000
LIN	X-10000	Y70000	F20000		Move to designated position by linear interpolation. Can also be written as LIN X-10000
CCW	X-10000	Y10000	J-30000	F20000	Move to designated position by arc interpolation. Can also be written as CCW Y10000 J-30000
SRET					

Example of G-code instructions for trajectory 2

P0					P0 subroutine
G90					Set up absolute coordinate
G00	X-10.0	Y10.0			Fast move to designated position
G01	X20.0	Y10.0	F40.0		Move to designated position by linear interpolation. Can also be written as G01 X20.0 F40.0
G03	X20.0	Y70.0	J30.0	F20.0	Move to designated position by arc interpolation. Can also be written as G03 Y70.0 J30.0 F20.0
G01	X-10.0	Y70.0	F20.0		Move to designated position by linear interpolation. Can also be written as G01 X-10.0
G03	X-10.0	Y10.0	J-30.0	F20.0	Move to designated position by arc interpolation. Can also be written as G03 Y10.0 J-30.0
SRET					

Example of motion instructions for trajectory 3

P0					P0 subroutine
INCT					Set up relative position
DRV	X-25000	Y25000			Fast move to designated position
LIN	X50000	Y0	F20000		Move to designated position by linear interpolation.
LIN	X-25000	Y60000	F20000		Move to designated position by linear interpolation. Can also be written as LIN X-25000 Y60000
LIN	X-25000	Y-60000	F20000		Move to designated position by linear interpolation. Can also be written as LIN X-25000 Y-60000
DRV	X25000	Y-25000			Fast move to designated position
SRET					

Example of G-code instructions for trajectory 3

P0					P0 subroutine
G91					Set up relative coordinate
G00	X-25.0	Y25.0			Fast move to designated position
G01	X50.0	Y0	F20.0		Move to designated position by linear interpolation.
G01	X-25.0	Y60.0	F20.0		Move to designated position by linear interpolation. Can also be written as G01 X-25.0 Y60.0
G01	X-25.0	Y-60.0	F20.0		Move to designated position by linear interpolation. Can also be written as G01 X-25.0 Y-60.0
G00	X25.0	Y-25.0			Fast move to designated position
SRET					

Example of motion instructions for trajectory 4

P0					P0 subroutine
ABST					Set up absolute coordinate
DRV	X10000	Y10000			Fast move to designated position
LIN	X10000	Y30000	F20000		Move to designated position by linear interpolation. Can also be written as LIN Y30000 F20000
CCW	X10000	Y110000	J40000	F20000	Move to designated position by arc interpolation. Can also be written as CCW Y110000 J40000
CW	X10000	Y230000	R60000	F15000	Move to designated position by arc interpolation. Can also be written as CW Y230000 R60000 F15000
LIN	X10000	Y210000	F15000		Move to designated position by linear interpolation. Can also be written as LIN Y210000
CCW	X10000	Y130000	J-40000	F15000	Move to designated position by arc interpolation. Can also be written as CCW Y130000 J-40000
CW	X10000	Y10000	R60000	F20000	Move to designated position by arc interpolation. Can also be written as CW Y10000 R60000 F20000
SRET					

8 Application Examples

Example of G-code instructions for trajectory 4

P0					P0 subroutine
G90					Set up absolute coordinate
G00	X10.0	Y10.0			Fast move to designated position
G01	X10.0	Y30.0	F20.0		Move to designated position by linear interpolation. Can also be written as G01 Y30.0 F20.0
G03	X10.0	Y110.0	J40.0	F20.0	Move to designated position by arc interpolation. Can also be written as G03 Y110.0 J40.0
G02	X10.0	Y230.0	R60.0	F15.0	Move to designated position by arc interpolation. Can also be written as G02 Y230.0 R60.0 F15.0
G01	X10.0	Y210.0	F15.0		Move to designated position by linear interpolation. Can also be written as G01 Y210.0
G03	X10.0	Y130.0	J-40.0	F15.0	Move to designated position by arc interpolation. Can also be written as G03 Y130.0 J-40.0
G02	X10.0	Y10.0	R60.0	F20.0	Move to designated position by arc interpolation. Can also be written as G02 Y10.0 R60.0 F20.0
SRET					

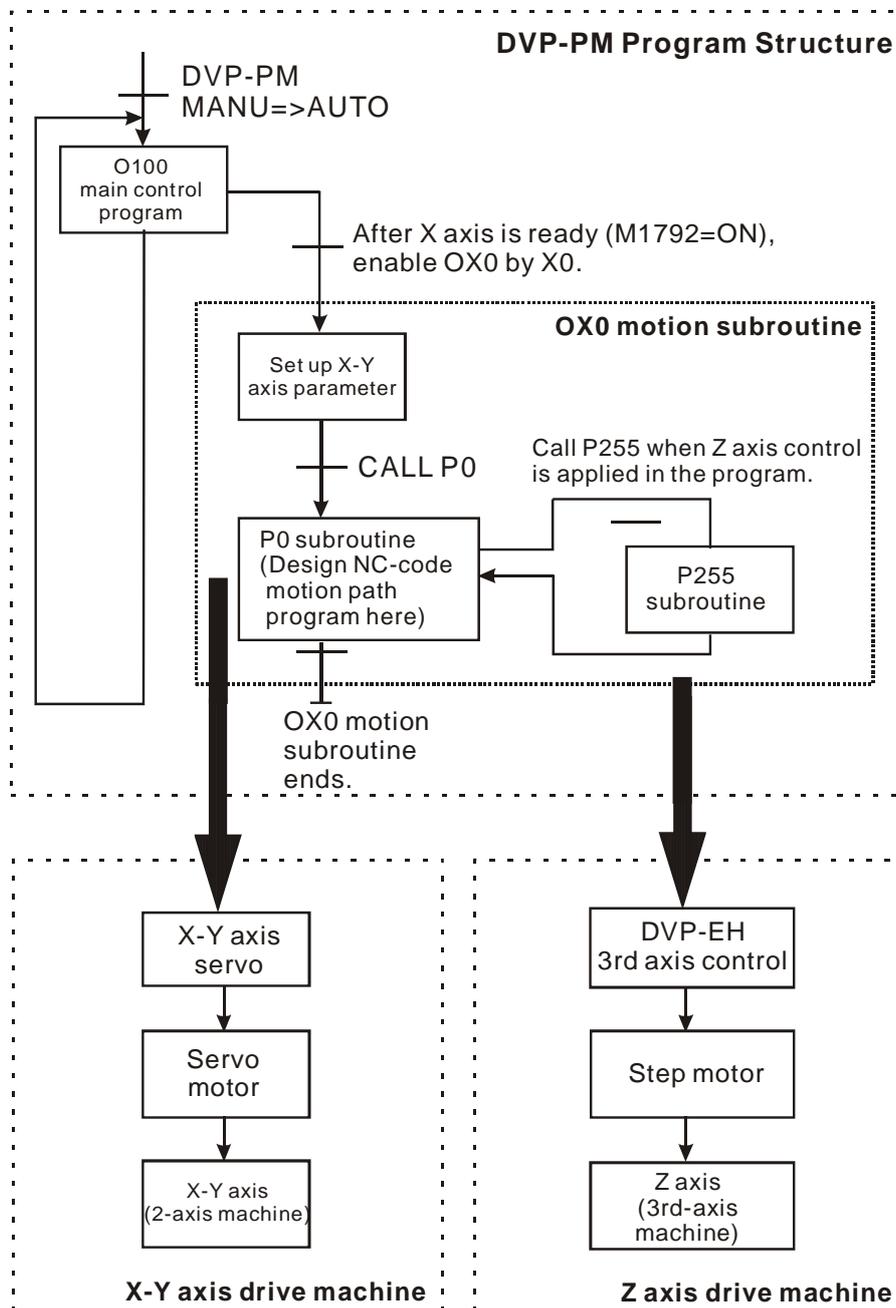
6. When M1072 in DVP-PM is ON, DVP-PM will be in AUTO status and the above positioning instructions will be executed.

8.2 Applying Application Examples in PMSoft

You can apply the application example “motionSample” to draw English letters, any graph or text. If you wish to apply this function to any 2-axis control equipment, you can modify the example program below for you to realize more diverse control purposes. Path: Open PMSoft => File => Open Examples => select “motionSample_26Letter” file to open the example program.

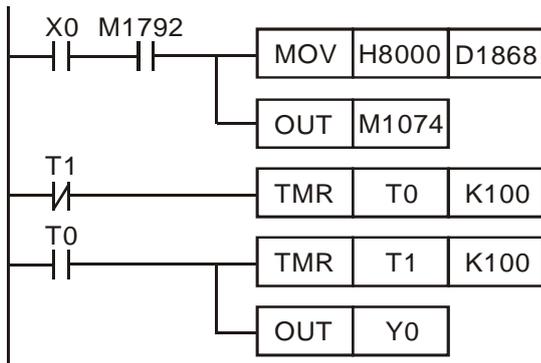
8.2.1 Program Structure

In order to apply DVP-PM for drawing, we need to convert letters or graphs into G-Codes (i.e. NC code) before designing the main control program. In addition, DVP-PM 20D only offers X/Y axis interpolation, we have to apply additional Z axis control for the “pen-lifting” function. In this example, we use DVP-EH to complete the third axis control. (Other controllers are also applicable).



further place other operations in the main program.

Ladder diagram:



Operation:

When OX is ready (M1792 = ON), OX0 motion subroutine can be enabled by X0

Enable OX0 motion subroutine

Other operations

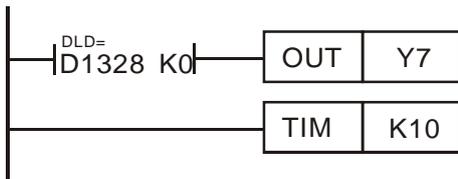
Other operations

Other operations

3. P255 ~ SRET: set up 3rd axis (Z) control system

The 3rd axis control will activate according to the parameter setting of G00 and G01 instructions. When G00/G01 sets target position of Z-axis (D1328), P255 will be enabled to drive Y7 (3rd axis control signal) and enable DVP-EH to control the lift or drop of a pen (i.e. up/down movement of Z axis). For more details, please refer to G00 and G01 instructions in Chapter 6.

Ladder diagram:



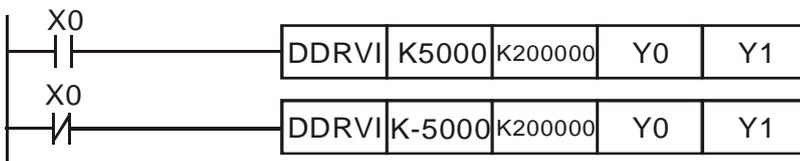
Operations:

Y7 (control signals for pen lifting) is controlled by the target position (D1328) of Z axis.

Pause for 0.1 second

3-axis control signal (Y7) in P255 of DVP-PM drives the input contact X1 of DVP-EH. When X1 is ON, the step motor controlled by pulses will lift the pen through DDRVI instruction; when X1 is OFF the step motor will drop the pen through DDRVI instruction.

The program of DVP-EH:



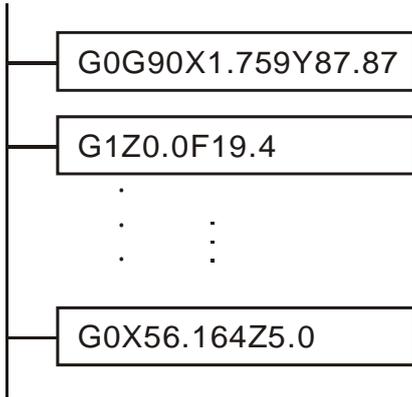
Connect the input devices Y0 and Y1 in DVP-EH to the pulse input terminals on the step motor.

4. P0 ~ SRET: 2-axis (X, Y) interpolation control

After converting the letters or graphs into G-Codes (NC-Codes), place the G-Code into OX0 subroutine but into P0 subroutine in order to simplify the program. Operate the P0~SRET section with the 3 sections above and the drawing of letters or graphs can be performed.

8 Application Examples

Ladder diagram:



Operations:

G-Code (NC-Code) which draws trajectories

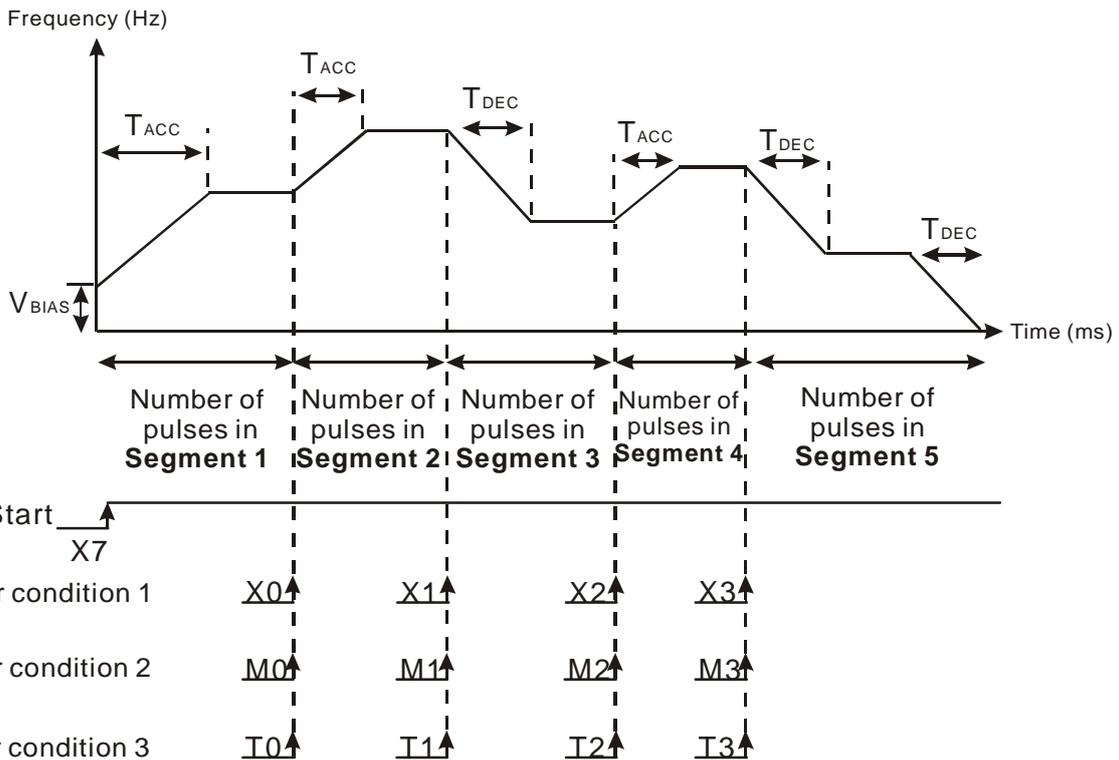
When the program sections 1 ~ 4 are completed, drawing of letters, graphs or any texts by DVP-PM can be performed by DVP-PM

8.3 Planning Variable Speed Operation

This section introduces how to trigger many segments of speed (variable speed) in a fixed route by using single-speed positioning mode.

8.3.1 Program structure

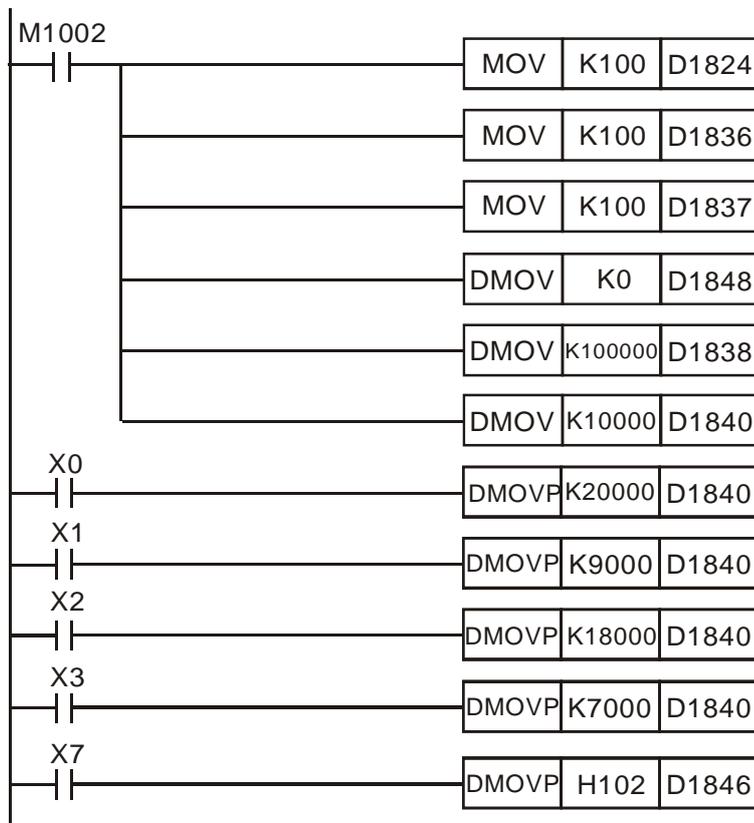
1. Trigger condition 1: external input signal. X0 ~ X3 switches to the 2nd ~ 5th speed.
2. Trigger condition 2: comparison results of CP. M0 ~ M3 switches to the 2nd ~ 5th speed.
3. Trigger condition 3: Timer. T0 ~ T3 switches to the 2nd ~ 5th speed.



DD1838 (total number of output pulses) = number of pulses in Segment 1 + Segment 2 + ... + Segment 5

8.3.2 Design Example Program

Ladder diagram of trigger condition 1:



Operations:

Set up bias speed of X axis (V_{BIAS})

Set up acceleration time of X axis (T_{ACC})

Set up deceleration time of X axis (T_{DEC})

Clear the current position of X axis as 0

Set up the total distance of all segments for X axis

Set up the operation speed for the 1st segment on X axis

X0 = ON, modify the operation speed into 20,000Hz

X1 = ON, modify the operation speed into 9,000Hz

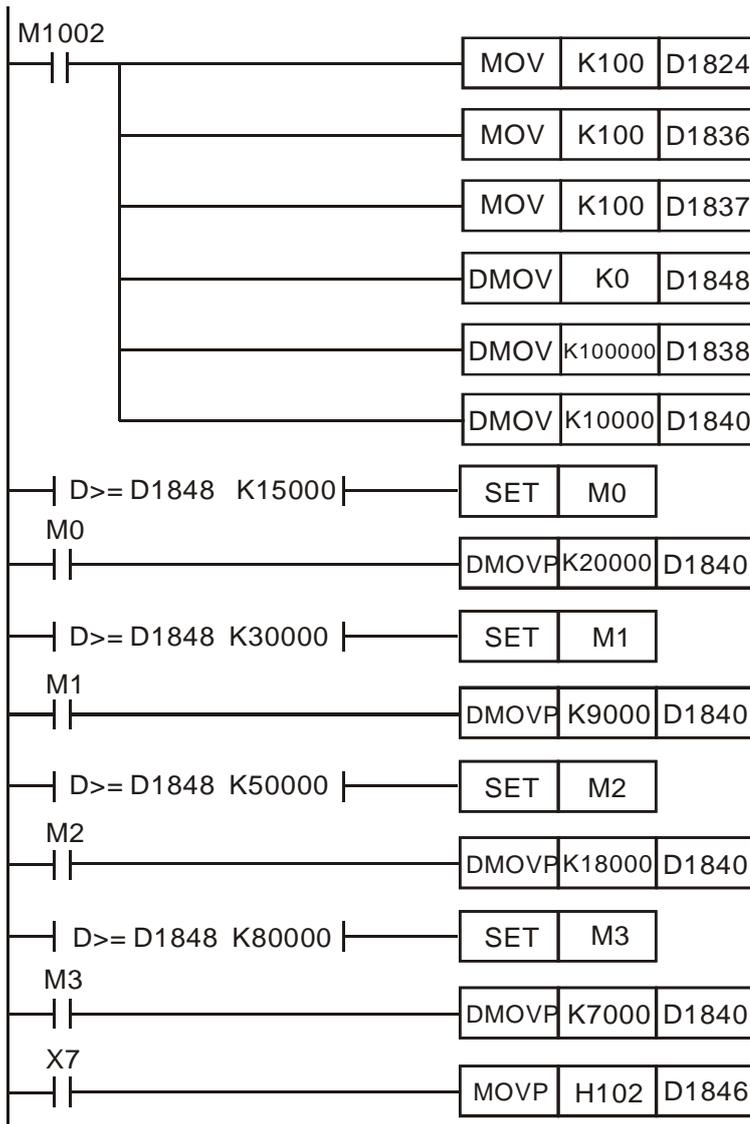
X2 = ON, modify the operation speed into 18,000Hz

X3 = ON, modify the operation speed into 7,000Hz. Pulse output will be finished in this segment.

X7 = ON, enable single speed positioning mode of X axis.

8 Application Examples

Ladder diagram for trigger condition 2:



Operations:

Set up the bias speed of X axis (V_{BIAS})

Set up the acceleration time of X axis (T_{ACC})

Set up the deceleration time of X axis (T_{DEC})

Clear the current position of X axis as 0

Set up the total distance of all segments for X axis

Set up the operation speed for the 1st segment on X axis

Compare K15000 with the current position
M0 = ON, modify the operation speed into 20,000Hz

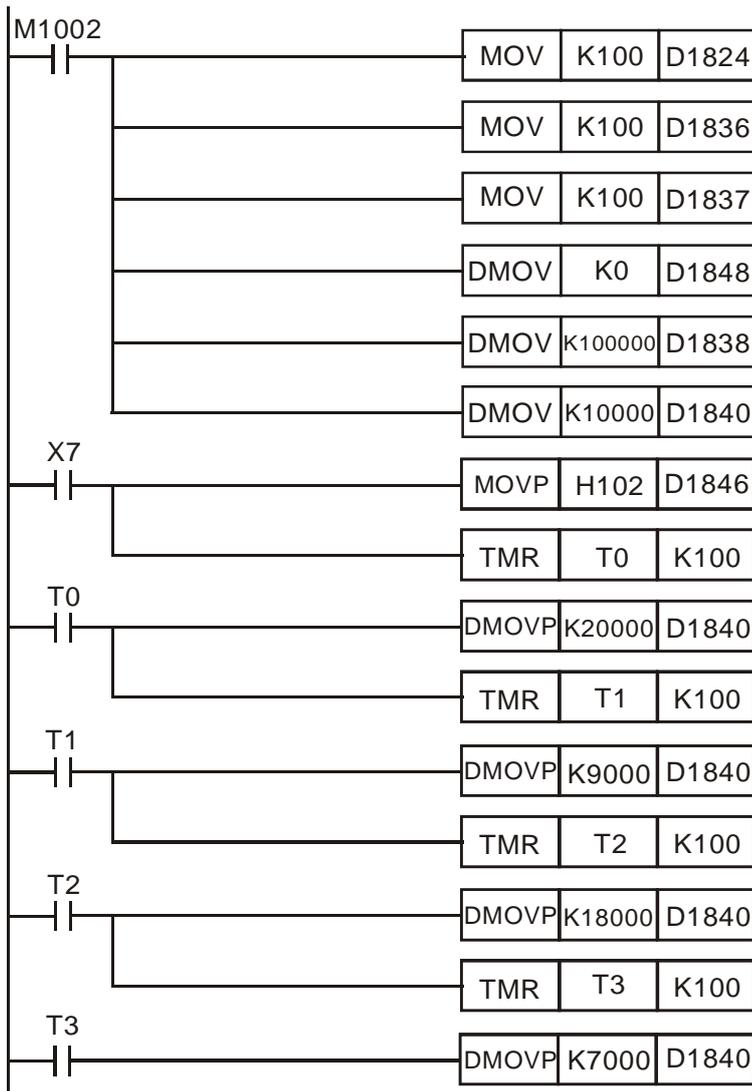
Compare K30000 with the current position
M1 = ON, modify the operation speed into 9,000Hz

Compare K50000 with the current position
M2 = ON, modify the operation speed into 18,000Hz

Compare K80000 with the current position
M3 = ON, modify the operation speed into 7,000Hz. Pulse output will be finished in this segment.

X7 = ON, enable single speed positioning mode of X axis.

Ladder diagram for trigger condition 3:



Operations:

Set up the bias speed of X axis (V_{BIAS})

Set up the acceleration time of X axis (T_{ACC})

Set up the deceleration time of X axis (T_{DEC})

Clear the current position of X axis as 0

Set up the total distance of all segments for X axis

Set up the operation speed for the 1st segment on X axis

X7 = ON, enable single speed positioning mode on X axis and T0 executes, preparing for switching to the 2nd segment.

T0 = ON, modify the operation speed into 20,000Hz and T1 executes, preparing for switching to the 3rd segment.

T1 = ON, modify the operation speed into 9,000Hz and T2 executes, preparing for switching to the 4th segment.

T2 = ON, modify the operation speed into 18,000Hz and T3 executes, preparing for switching to the 5th segment.

T3 = ON, modify the operation speed into 7,000Hz. Pulse output will be finished in this segment.

8 Application Examples

8.4 Connect DVP-PM (as Master) and DVP01PU-H2 (as Slave) for 3rd Axis Control

1. Enable O100 and execute OX0.
2. When the execution encounters G01 Z-25000 F10000 in OX0 subroutine, the program will call P255.
3. In P255, when D1328 < 0, drive DVP01PU-H2 to control the 3rd axis. Move to target position K1,000 in operation speed K10,000.
4. Return to OX0 after the execution of P255 is completed. Wait for 10 seconds.
5. When the execution encounters G01 Z10000 F20000 in OX0 subroutine, the program will call P255.
6. In P255, when D1328 > 0, drive DVP01PU-H2 to control the 3rd axis. Move to target position K2,000 in operation speed K20,000.
7. Return to OX0 after the execution of P255 is completed.

Instruction List of O100 main program and subroutines:

O100 main program:

Design instructions for initializing current status of X/Y axis and enabling OX0 motion subroutine

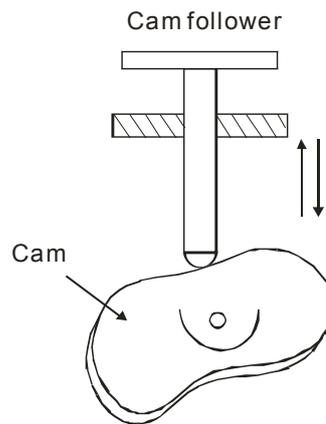
LD	M1002				
MOV	H8000	D1868			Write in the No. (0) of OX to be enabled
SET	M1074				Enable OX motion subroutine
M102					
OX0 subroutine					
G01	Z-25000	F10000			G01 3 rd axis control
TIM	K1000				Pause for 10 seconds
G01	Z10000	F20000			G01 3 rd axis control
M2					
P255 subroutine					
BRET					
TO	K0	K31	H2	K1	Stop DVP01PU
DLD>=	K0	D1328			Compare 0 with D1328
DTO	K0	K23	K1000	K1	Set up DVP01PU target position
DTO	K0	K25	D1330	K1	Set up DVP01PU operation speed
TO	K0	K32	H1	K1	Set up DVP01PU single speed positioning mode
TO	K0	K31	H100	K1	Enable DVP01PU
DLD<	K0	D1328			Compare 0 with D1328
DTO	K0	K23	K2000	K1	Set up DVP01PU target position
DTO	K0	K25	D1330	K1	Set up DVP01PU operation speed
TO	K0	K32	H1	K1	Set up DVP01PU single speed positioning mode
TO	K0	K31	H100	K1	Enable DVP01PU
SRET					

Note: This application is only applicable for DVP20PM00D

9.1 Introduction to Electrical CAM (E-CAM)

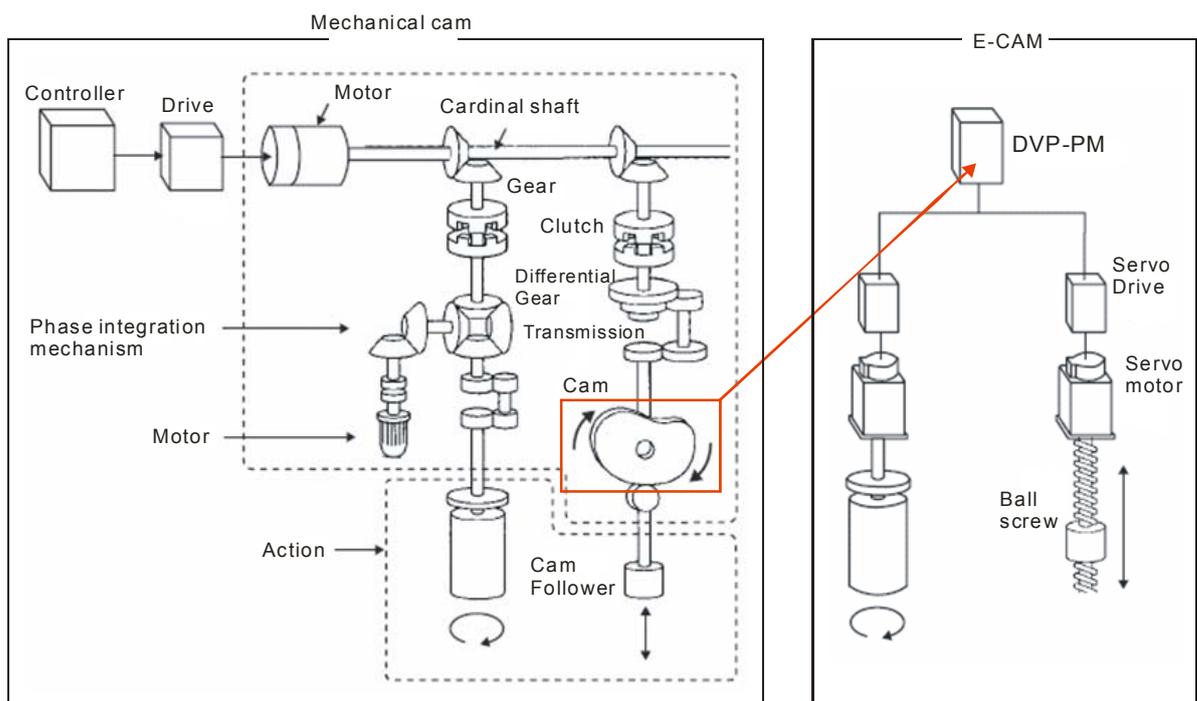
Mechanical cam system consists of 3 parts including a cam, a cam follower, and a fixed linkage.

- Mechanical cam is a rotating piece with irregular form, usually an input device with steady rotation rate, operates cam follower to produce a smooth reciprocating motion through the fixed linkage between the cam and the follower.
- Cam follower is a type of roller or needle bearing designed to be driven by the cam. Usually the cam follower is an output device for producing discrete signals.
- Fixed linkage is the frame for linking and fixing the cam the the follower.



In comparison with mechanical cam system, there are many advantages of using E-CAM

- E-CAM provides a more user-friendly interface.
- Not necessary to modify machinery system when different production process is required. Users only need to modify the E-CAM data in CAM chart by software.
- Higher acceleration speed can be achieved.
- Operation is smoother.

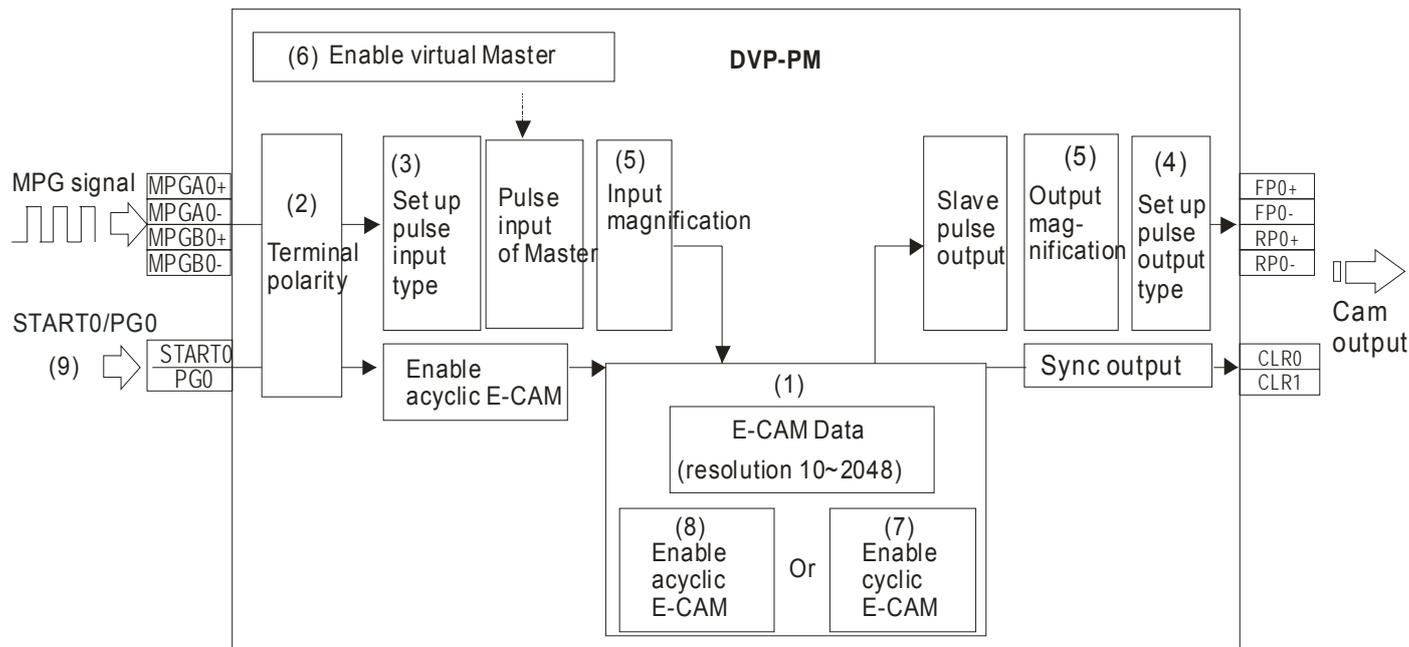


9 Electrical CAM

9.2 E-CAM Application

In electrical cam system of DVP-PM, we have to set up the master shaft (Master) which simulates mechanical cam, and also the slave shaft (Slave) which simulates the cam follower. In this section we will introduce the methods to obtain Master position and relationship between Master and Slave (E-CAM data).

■ Structure Map



■ Operation Process

STEP1	STEP2	STEP3
Initial Settings	Select Source of Master	Enable/Disable E-CAM
(1) Set E-CAM data (2) Set terminal polarity (not applicable for virtual Master) (3) Set up input pulse type (4) Set up output pulse type (5) Input/output magnification	<ul style="list-style-type: none"> ● MPG ● Encoder of Master servo ● Output pulses of DVP-PM (6) Enable virtual Master	(7) Enable cyclic E-CAM (8) Enable acyclic E-CAM

9.2.1 Initial Settings

9.2.1.1 Set E-CAM Data

- There are 2 methods applicable for setting up E-CAM data in DVP-PM:

Method 1: Use DTO instruction to create E-CAM data directly.

Method 2: Use E-CAM Chart of PMSOFT to draw the CAM curve

For detailed explanation on the setting methods of E-CAM data, please refer to section 9.4

9.2.1.2 Set Terminal Polarity

- Input Terminals

1. MPGA0/MPGB0: Pulse input terminal for Master. Max allowable frequency: 200kHz.

2. START0/PG0: Input terminal for enabling acyclic E-CAM

■ Output Terminals

1. FP0/RP0: Output terminal for pulse output of E-CAM. Max output frequency: 500kHz.
2. CLR0/CLR1: Output terminal for E-CAM synchronized output signal. When D1839, D1838 (PI) \leq CP (Current Position) of Master (X axis) \leq D1843, D1842 (PII), CLR0/CLR1 will be ON. (For special application of CLR0/CLR1, please refer to Reference function of synchronized output in section 9.3.

■ Input Terminal Polarity Setting

The polarity of input terminals is set by the corresponding bits of D1799. bit # = ON sets the input terminal as NO contact while OFF sets the input terminal as the NC contact. For example, to set MPGA0/MPGB0 as NO contacts, you have to set ON b1 and b2 of D1799, i.e. specify D1799 = 6.

D1799 (Input Terminal Polarity)

Bit#	X-axis Input Terminal Polarity
0	PG0
1	MPGB0
2	MPGA0
7	START0

■ Input Terminal Digital Filter

D1806 High Byte: Filter coefficient of MPG0 / MPG1. D1806 Low Byte: Filter coefficient of other input points other than X points. Filter frequency: $85000/2^{N+4}$ (kHz).

N	kHz	N	kHz
1	2656.25	11	2.593994
2	1328.125	12	1.296997
3	664.0625	13	0.648499
4	332.0313	14	0.324249
5	166.0156	15	0.162125
6	83.00781	16	0.081062
7	41.50391	17	0.040531
8	20.75195	18	0.020266
9	10.37598	19	0.010133
10	5.187988		

9.2.1.3 Input/Output Pulse Type Setting

When Virtual Master is applied or the input pulses for Master are from Y axis of PM, settings of D1864 (MPG pulse input type) and D1816/D1896/D1976 (pulse output type) should match with each other, i.e. if the source of Master is set as single phase (P/D), the output pulse type of Y axis should be single phase as well. For wirings, simply connect MPGA0 (A0+, A0-) with Y axis output terminals (FP1+, FP1-) and the setting is completed.

9 Electrical CAM

■ Pulse Input Type

MPG Pulse Input Type Setting: D1864

b9	b8	Pulse input type (positive logic)	Explanation
0	0		Double-phase (FP/RP)
0	1		Single-phase (P/D)
1	0		A/B phase (Single frequency)
1	1		A/B phase (Quadruple frequency)

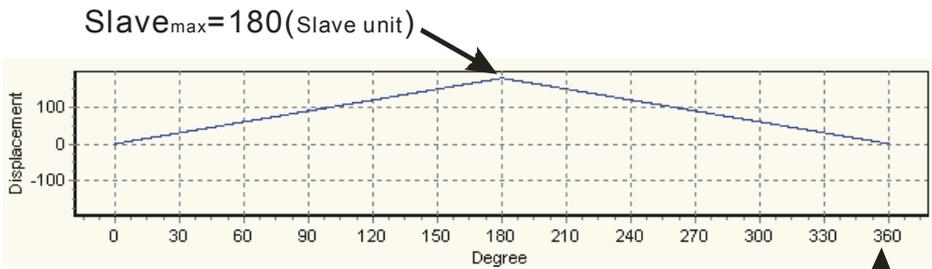
■ Pulse output Type

Pulse Output Type Setting: D1816/D1896/D1976

B5	B4	Pulse output type (positive logic)	Explanation
0	0		Double-phase (FP/RP)
0	1		Single-phase (P/D)
1	0		A/B phase (AB)

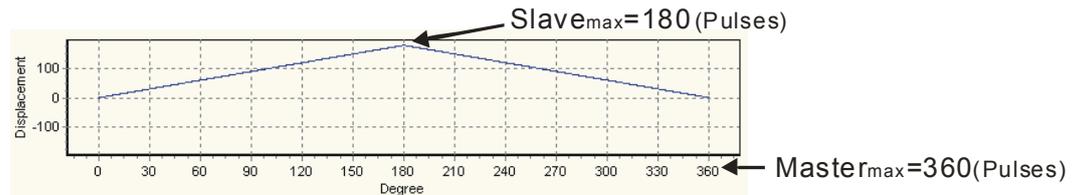
9.2.1.4 Input /Output Magnification Setting

DVP-PM provides E-CAM Data magnification function. When you input E-CAM Data as the below diagram, E-CAM Data can be magnified or minified by setting b0/b1 of D1816 (unit setting) as well as DD1858 (Electronic Gear Ratio), respectively controlling the output magnification and the input magnification



D1816

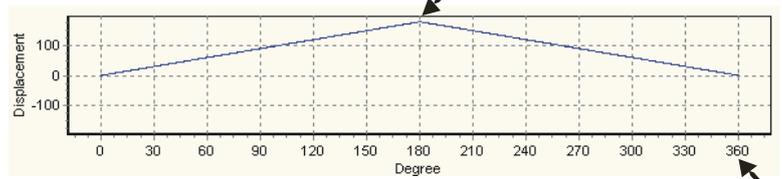
b1	b0	Unit
0	0	Motor system



D1816

b1	b0	Unit
0	1	Motor system
1	0	Combined system
1	1	

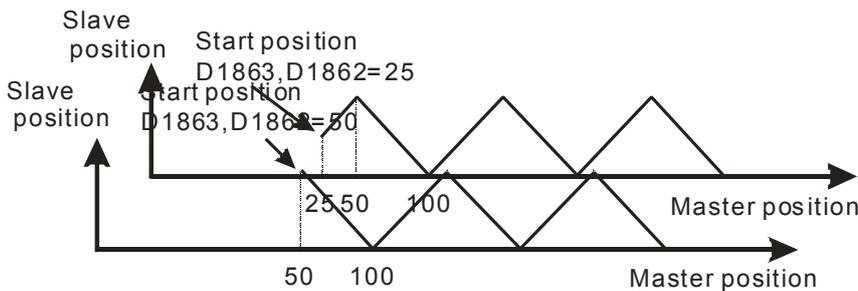
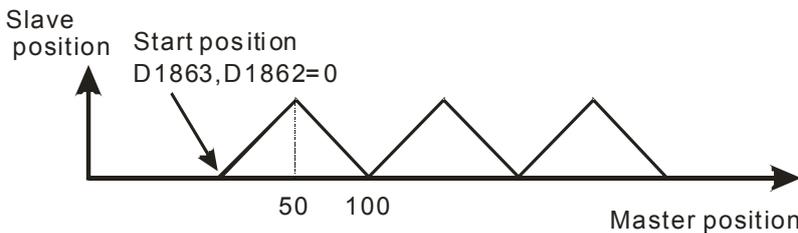
$$\frac{\text{Slave}_{\text{max}} = 180 \times \text{D1819} \cdot \text{D1818 (Pulses per revolution)}}{\text{D1821} \cdot \text{D1820 (Displacement per revolution)}} \text{ (Pulses) = Output magnification}$$



$$\frac{\text{Master}_{\text{max}} = 360 \times \text{D1858 Numerator of electronic gear ratio}}{\text{D1859 Denominator of electronic gear ratio}} \text{ (Pulses) = Input magnification}$$

9.2.1.5 E-CAM Activation at Non-zero Position

You can set M1752 = ON to enable E-CAM activation at non-zero position, and write the start position of Master in D1862, D1863.

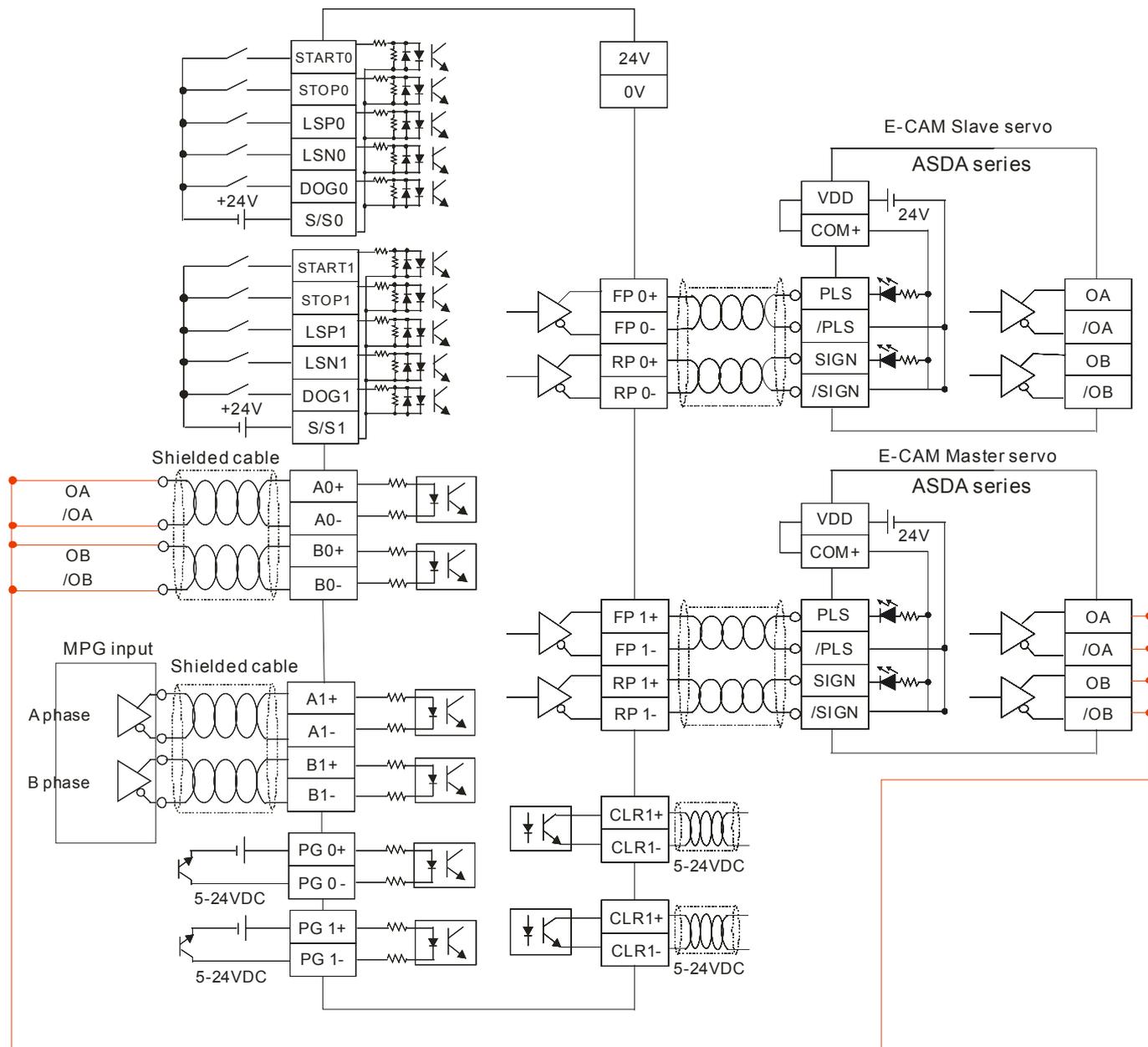


9 Electrical CAM

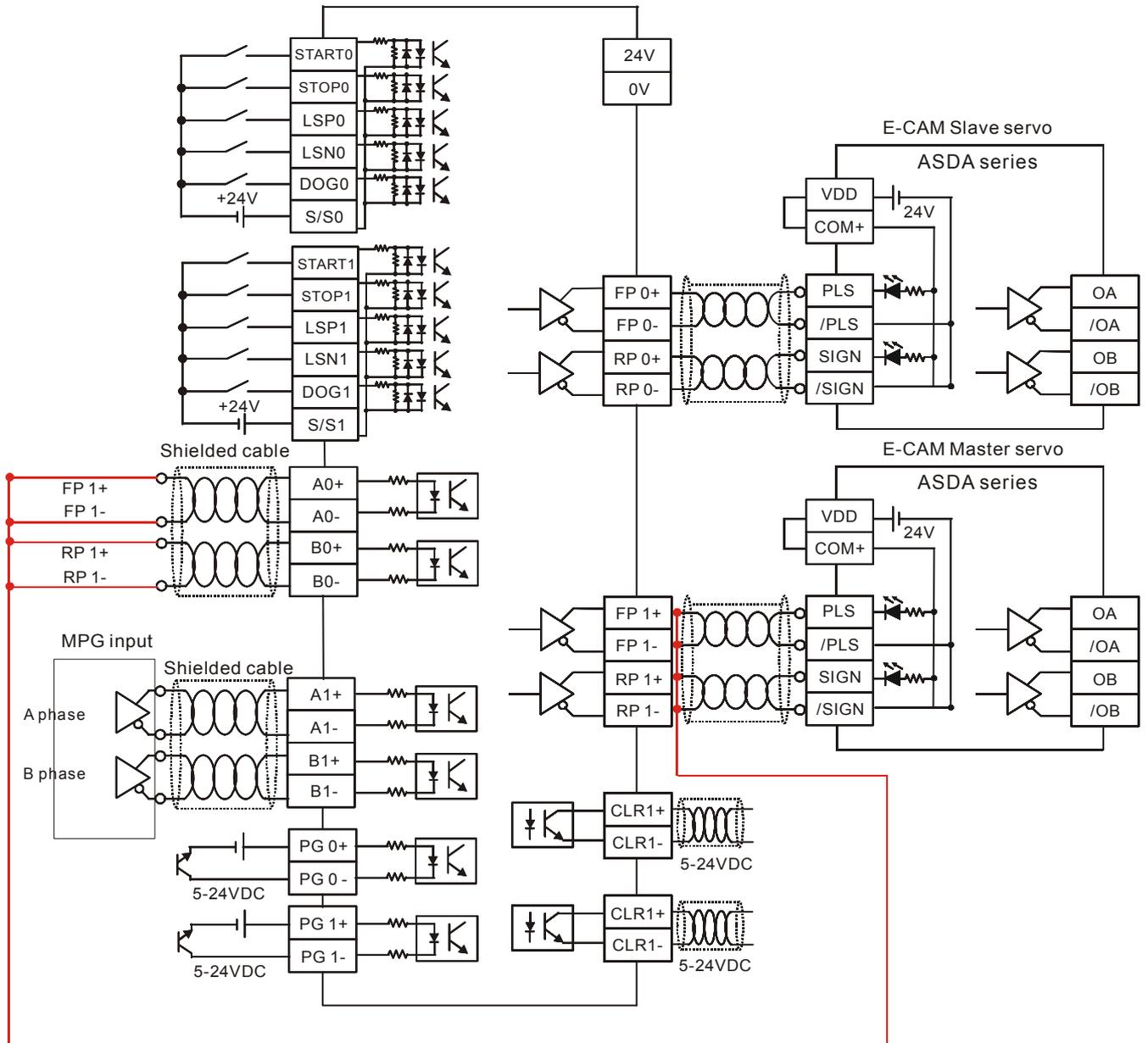
9.2.2 Obtain Master Position

There are 3 methods to obtain Master position.

- Method 1: Apply internal virtual Master – simple settings and high accuracy. (Refer to 9.2.3 in P9-4)



- Method 2: Obtain Master position from the encoder of Master servo. Receive the signals from the encoder then convert the signals into Master position.



9 Electrical CAM

Method 3: Apply Virtual Master

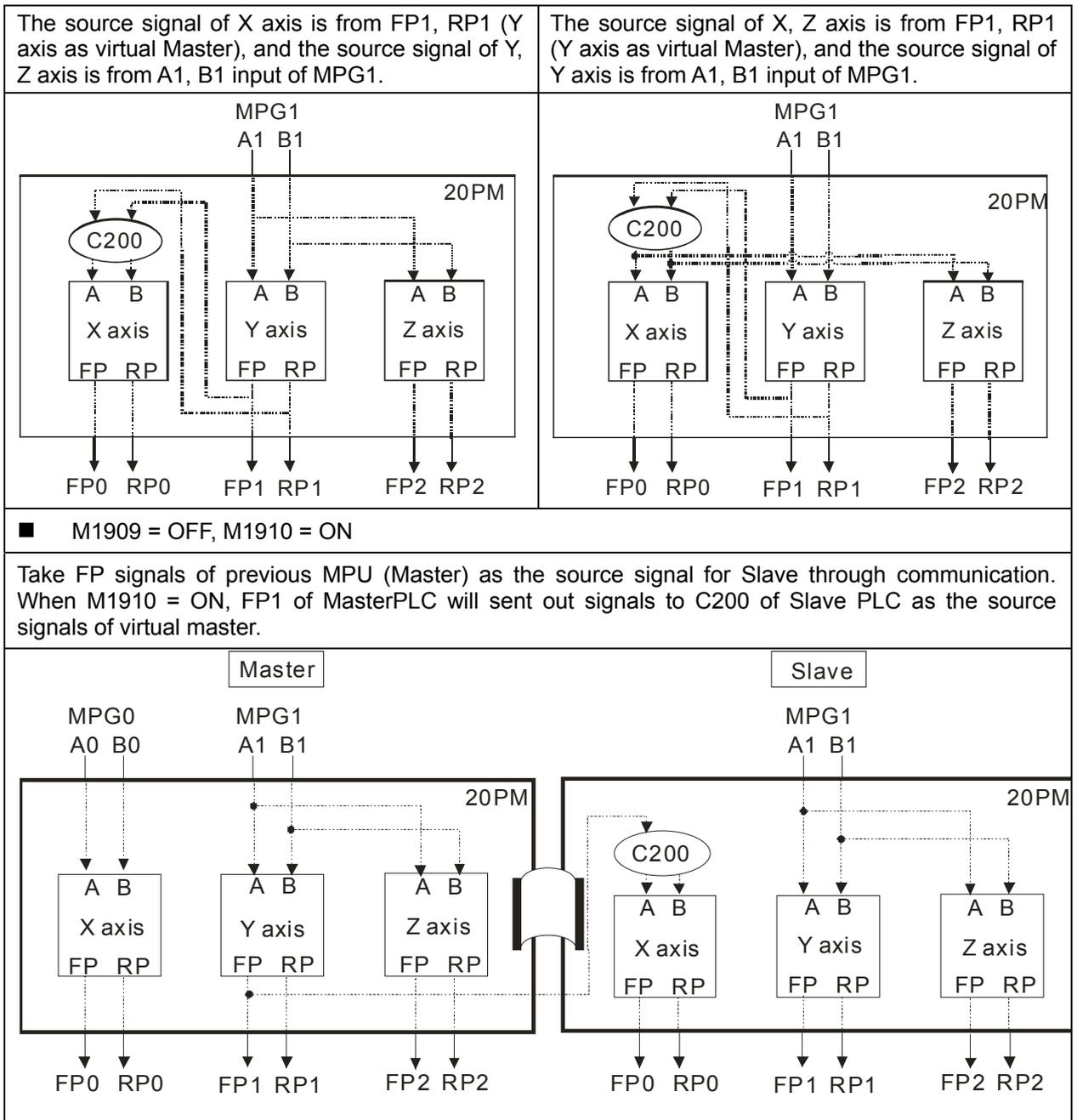
In virtual Master mode, you can catch the output signals from internal Y axis as the signal source of Master for X axis or Z axis. In this case, you don't have to apply additional wirings.

To apply virtual Master mode, set ON M1909 and A0 will be internally linked with FP of Y axis; set ON M1910 and B0 will be internally linked with RP of Y axis. After the 2 flags are ready, DVP-PM will automatically apply the internal counter C200 as the source of counting pulses for virtual Master.

However, when M1909 is OFF and M1910 is ON, the source of Master will be the FP1 signals of Y axis on the previous MPU (DVP-PM).

In addition, care should be taken when setting the pulse type of MPG and the pulse type of Y axis. Pulse types of the two sides should be the same. Please refer to 9.2.1.3 for the setting.

Single CAM	Multiple CAM
<p>■ M1909 = OFF, M1910 = OFF</p>	
<p>The source signal of X axis is from A0, B0 input of MPG0, and the source signal of Y, Z axis is from A1, B1 input of MPG1.</p>	<p>The source signal of X, Y, Z axis is from A0, B0 input of MPG0.</p>
<p>■ M1909 = ON, M1910 = OFF</p>	
<p>The source signal of X axis is from FP1 (Y axis as virtual Master), and the source signal of Y, Z axis is from A1, B1 input of MPG1.</p>	<p>The source signal of X, Z axis is from FP1 (Y axis as virtual Master), and the source signal of Y axis is from A1, B1 input of MPG1.</p>
<p>■ M1909 = ON · M1910 = ON</p>	



Note: According to our measurement, delay time for connecting single DVP-PM is 0.07us. Connection with up to approximately 29 MPUs will delay only 2us, equaling to a 500kHz pulse. Therefore, signal attenuation for this application is insignificant.

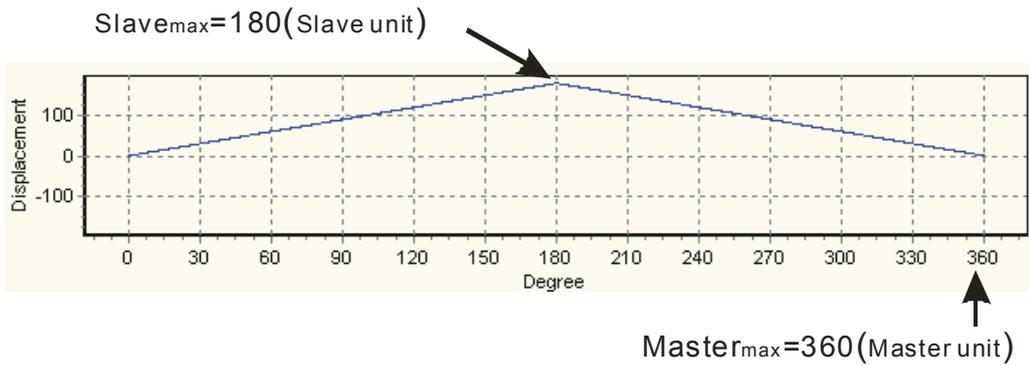
9.2.3 Start / Stop E-CAM

9.2.3.1 Start / Stop Cyclic E-CAM

In cyclic E-CAM, Master and Slave operate according to the user-defined E-CAM Data and repeat the E-CAM Data constantly.

E-CAM Data

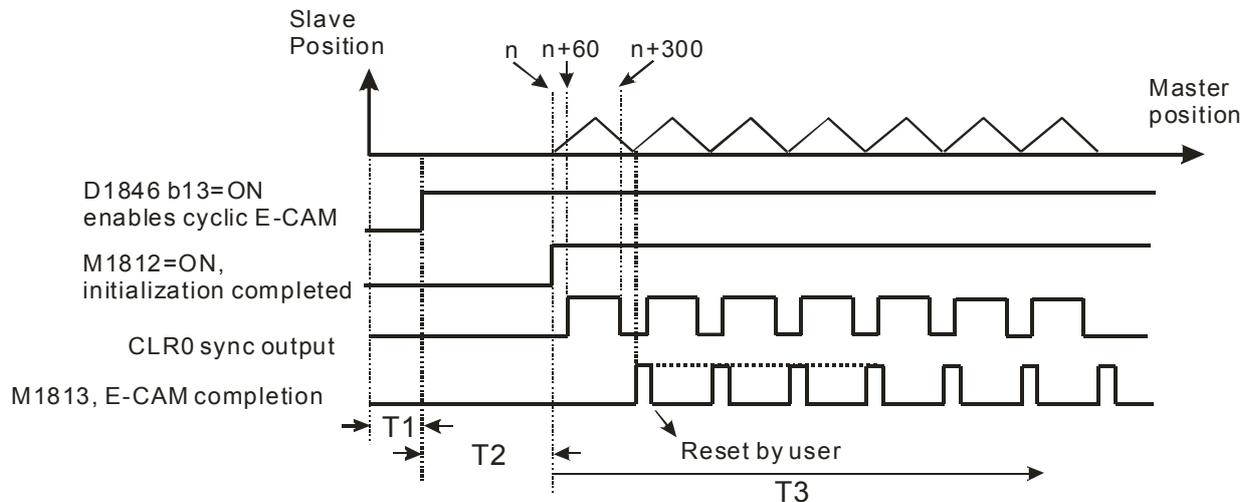
9 Electrical CAM



Start Cyclic E-CAM

The timing diagram below indicates an operation process of cyclic E-CAM operation with synchronized output position between 60~300.

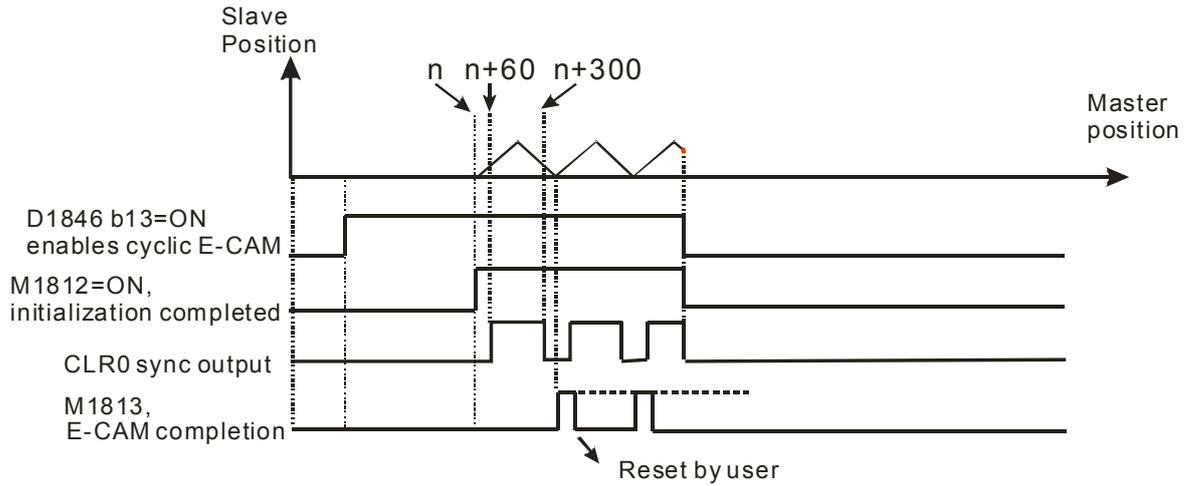
1. In T1, D1846 bit 13 = ON, cyclic E-CAM is enabled.
2. After the initialization interval of T2, M1812 = ON to indicate the completion of E-CAM initialization.
3. In T3, cyclic operation starts after M1812 = ON. Slave executes with Master constantly according to the E-CAM Data.
4. At the same time, CLR0 outputs according to the synchronized output range. When M1813 = ON, an E-CAM cycle is completed. User has to reset the flag for indicating the completion of other E-CAM cycles.



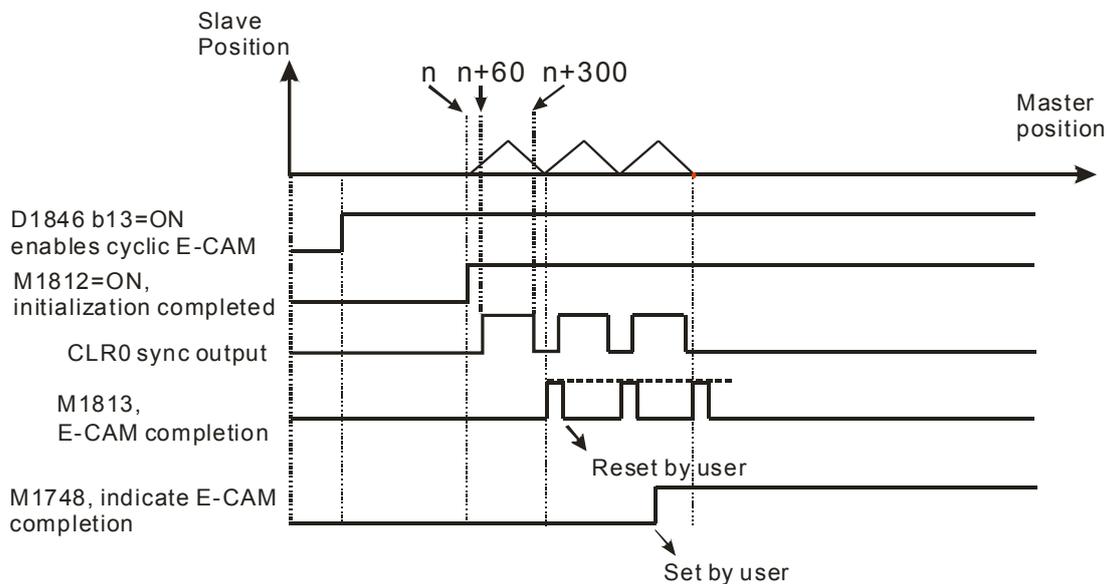
Stop Cyclic E-CAM

There are two ways to stop cyclic E-CAM:

1. When D1846 bit 13 (enabling cyclic E-CAM) = OFF, Slave of E-CAM stops immediately.



- When M1748 (indicate E-CAM completion) is ON during E-CAM execution, Slave of E-CAM will stop until the current E-CAM cycle is completed.

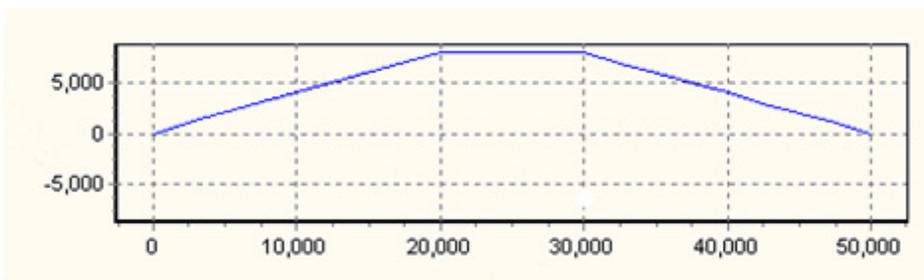


■ Application Example:

Setup:

Set up E-CAM Data as below:

E-CAM cycle: Master position 0~50000; Control unit: motor system; Sync output: Master position 20000~30000, CLR0 = ON as control signal.



Set up E-CAM Data as the above curve. E-CAM Data can be set up by PMSOft or DTO instruction (Please refer to explanations in 9.4). Connect pulse output terminals (FP+, FP-, RP+, RP-) to X axis MPG input terminals (A0+, A0-, B0+, B0-). In this case, input signals are from output signals of Y axis.

9 Electrical CAM

Control steps:

Step 1: Initialization

- (1) Clear the content in registers D1848, D1849, D1862, D1863, D1868.
- (2) Set up input pulse type as A/B phase (D1864 = H200)
- (3) Set up pulse output type of Y axis as A/B phase (H30)
- (4) Set up D1799 (input terminal polarity setting) = 6. MPGA/MPGB are NO contacts
- (5) Set up operation speed of Y axis variable speed operation (in this case Y axis outputs signals as input signals of Master).
- (6) Set up lower bound of CAM sync output D1839, D1838 = 20000 and upper bound D1843, D1842 = 30000. Between Master position 20000~30000 (can be monitored by D1862), CLR0 is ON. However, the CLR LED on the MPU will not respond to it. To monitor CLR signal, you can connect CLR0- with 0V and CLR0+ with Xn input contact then monitor the ON/OFF status of Xn input.

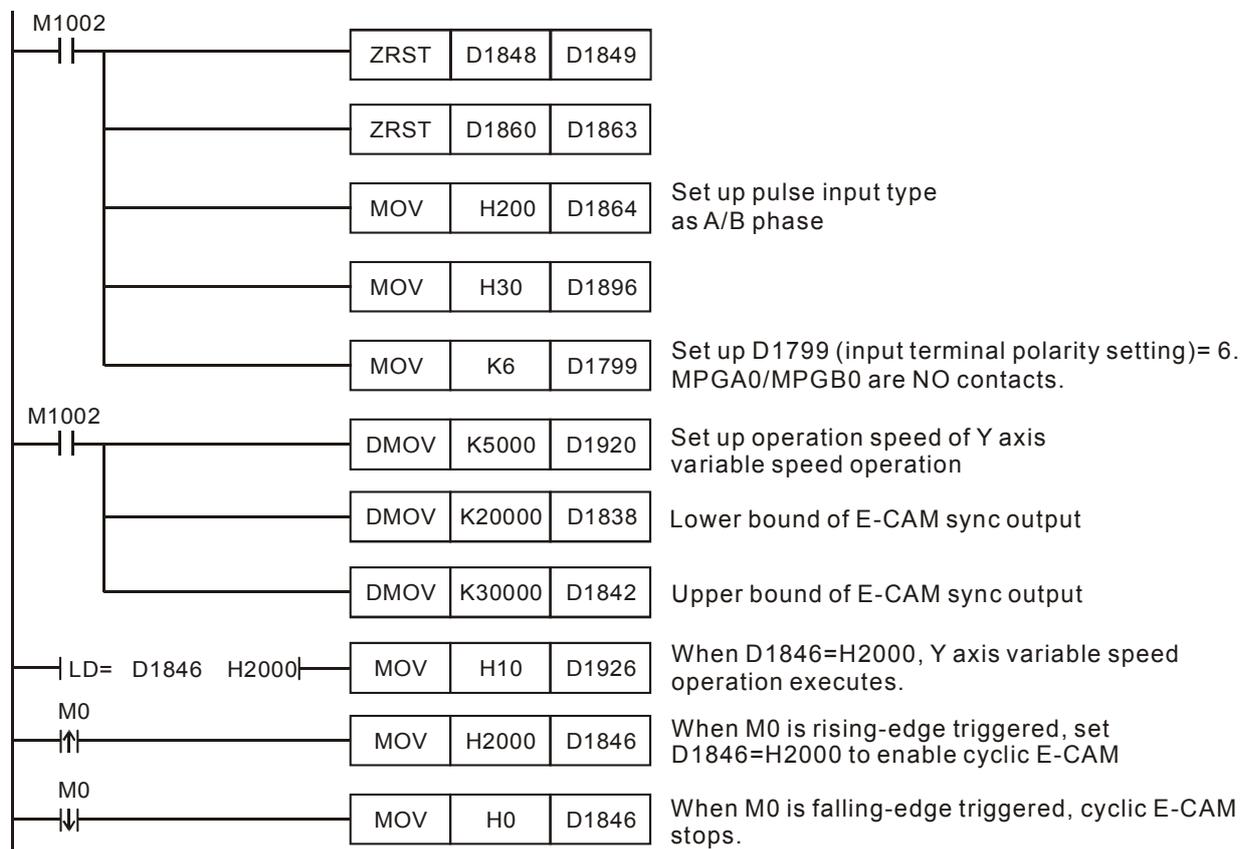
Step 2: Set On M0 to enable cyclic E-CAM.

When M0 is rising-edge triggered, Master starts to receive the variable speed pulses from Y axis, and Slave operates according to the above E-CAM Data curve. In addition, CLR0 = ON during Master position 20000~30000.

Step 3: Reset M0.

When M0 is falling-edge triggered, E-CAM stops.

Ladder Diagram:

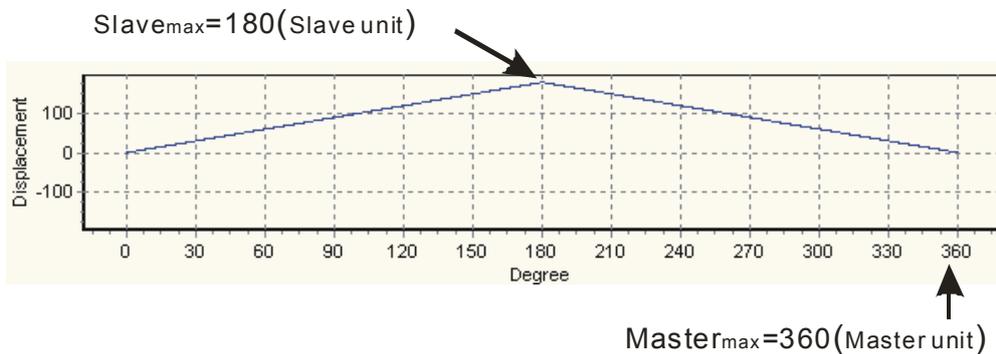


9.2.3.2 Start / Stop Acyclic E-CAM

In cyclic E-CAM, Master and Slave operate according to the user-defined E-CAM Data only when E-CAM start signal (START0/PG0) is triggered. Unlike cyclic E-CAM, acyclic E-CAM operates only one cycle for each triggered signal, i.e. E-CAM Data operates only once for one triggered signal.

Before selecting START0 as the start signal, M1035 has to be set ON for setting STOP0/START0 as the external input. After M1035 is set OFF, M1746 = OFF will enable START0 as the start signal of acyclic E-CAM rather than PG0.

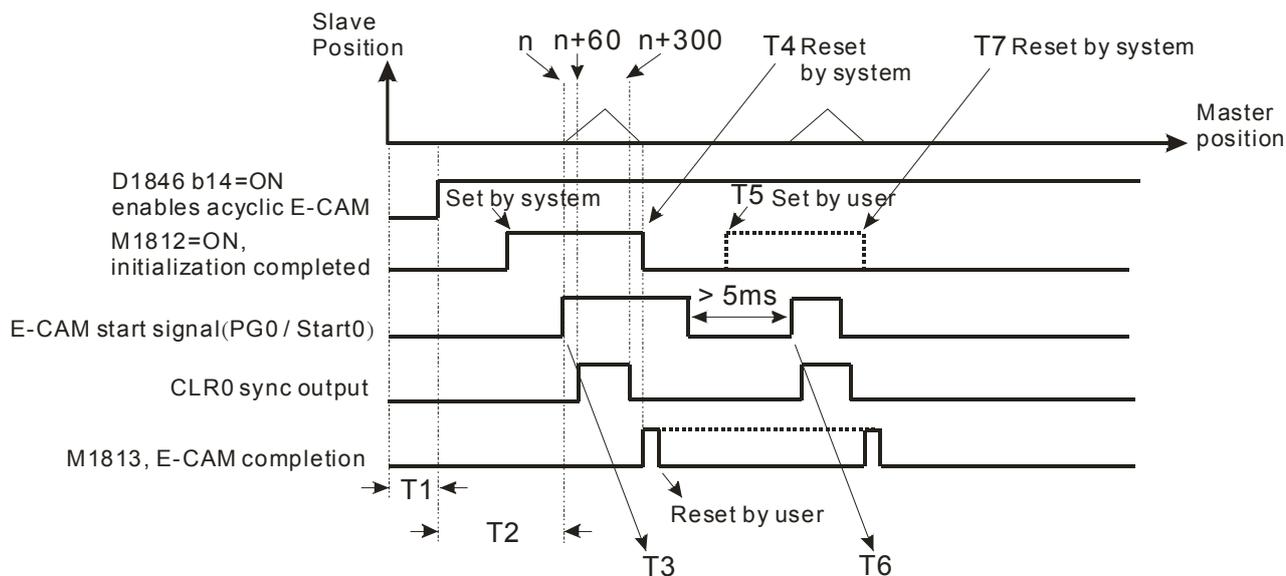
E-CAM Data



■ Start Acyclic E-CAM

The timing diagram below indicates an operation process of acyclic E-CAM operation with synchronized output position between 60~300.

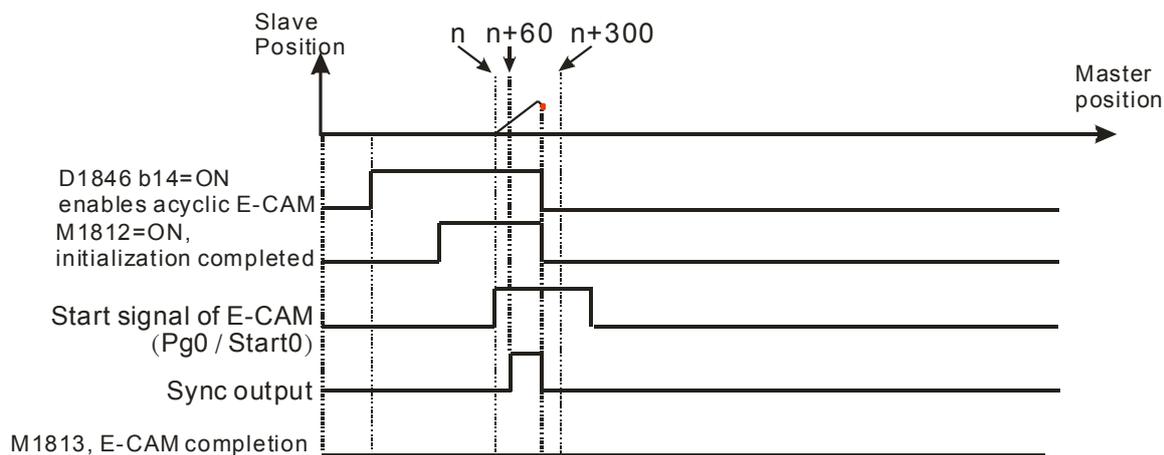
1. In T1, D1846 bit 14 = ON, acyclic E-CAM is enabled
2. After the initialization interval of T2, M1812 = ON to indicate the completion of E-CAM initialization. At this time Slave is still not activated.
3. In T3, acyclic operation starts after START0/PG0 (Input terminal for enabling acyclic E-CAM) is ON. When M1746 = ON, PG0 is the start signal of acyclic E-CAM; when M1746 = OFF and M1035 = ON, START0 is the start signal of acyclic E-CAM. In this case Slave starts to operate a cycle according to the E-CAM Data. ("M1035 = ON" enables START0 / STOP0 as the external input point).
4. When the cycle is completed at T4, DVP-PM will clear the state of M1812 = ON. In addition, users can also confirm the E-CAM completion by ON state of M1813.
5. In T5, you can decide whether to set ON M1812 for confirming the next E-CAM completion.
6. In T6 and T7, actions of T3~T4 will be repeated. Please note that the interval for enabling acyclic E-CAM should be longer than 5ms.
7. CLR0 outputs according to the upper/lower bound of sync output during each E-CAM cycle.



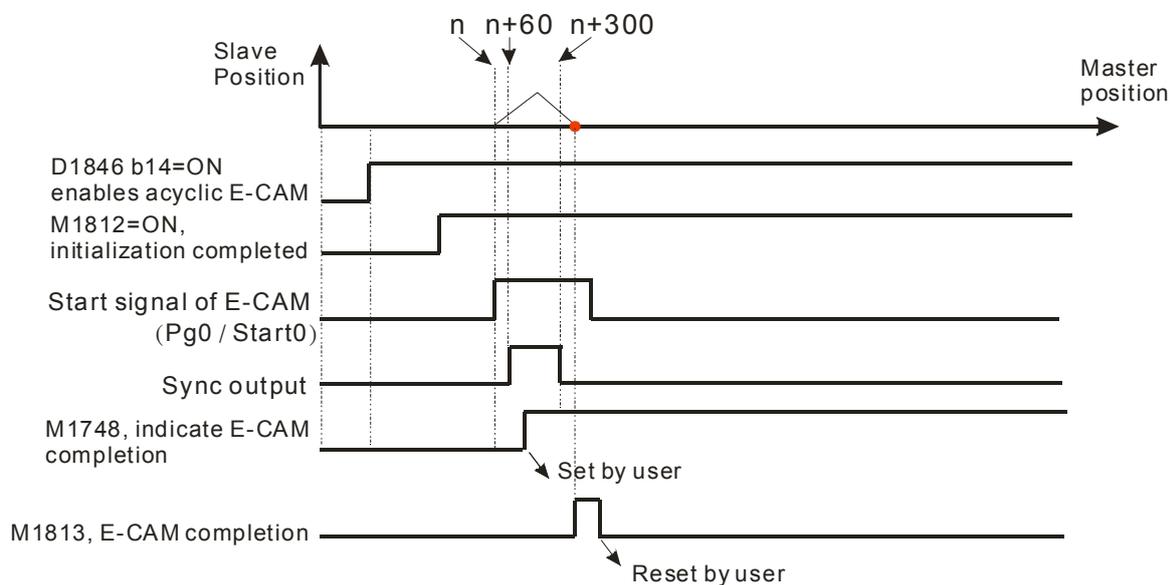
■ Stop Acyclic E-CAM

There are two ways to stop acyclic E-CAM:

1. When D1846 bit 14 (enabling acyclic E-CAM) = OFF, Slave of E-CAM stops immediately.



2. When M1748 (indicate E-CAM completion) is ON during E-CAM execution, Slave of E-CAM will stop until the current E-CAM cycle is completed



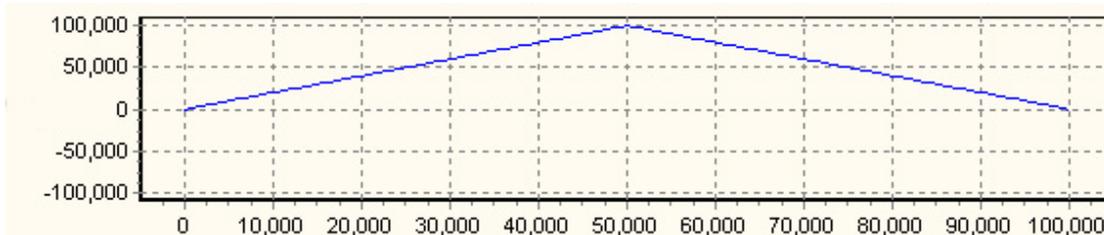
■ Application Example

Setup:

Set up E-CAM Data as below:

E-CAM cycle: Master position 0~100000; Control unit: motor system;

Every time when START0 is triggered, execute 3 acyclic E-CAM Data (D1832 = 2)



Set up E-CAM Data as the above curve. E-CAM Data can be set up by PMSoft or DTO instruction (Please refer to explanations in 9.4). Connect pulse output terminals (FP+, FP-, RP+, RP-) to X axis MPG input terminals (A0+, A0-, B0+, B0-). In this case, input signals are from output signals of Y axis.

Control Steps:

Step 1: Initialization

- (1) Clear the content in registers D1848, D1849, D1862, D1863, D1868.
- (2) Set up input pulse type as A/B phase (D1864 = H200)
- (3) Set up pulse output type of Y axis as A/B phase (D1896 = H30)
- (4) Set up D1799 (input terminal polarity setting) = 6, MPGA0 and MPGB0 are NO contacts
- (5) Set M1035 = ON to enable STOP0/START0 as external input point
- (6) Set M1746 = OFF to enable START0 as the start signal of acyclic E-CAM.
- (7) Set D1832 = 2 (Repeat E-CAM Data for two times)
- (8) Set up operation speed of Y axis variable speed operation (in this case Y axis outputs signals as Input signals of Master).

Step 2: Set On M0 to enable acyclic E-CAM.

When D1846=H4000, Y axis variable speed operation is enabled but X axis is not yet executed.

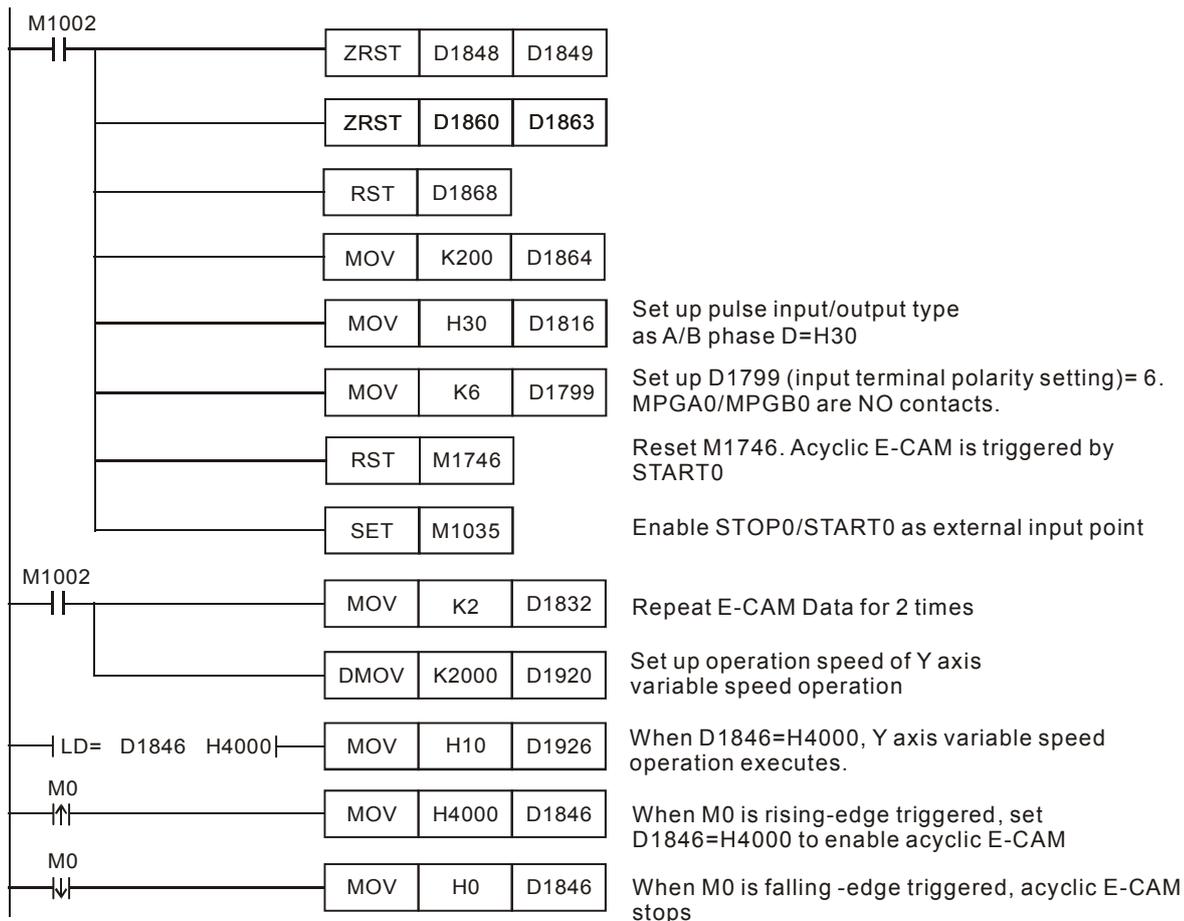
Step 3: Trigger START0.

X axis will be activated and executes the E-CAM data for 3 cycles.

Step 4: Reset M0.

When M0 is reset during the execution of E-CAM, X axis will be stopped.

Ladder Diagram:

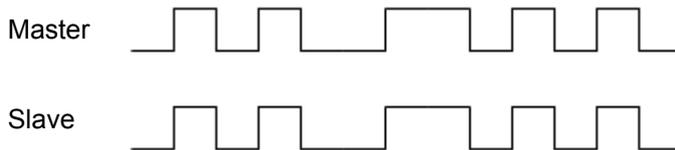


9.3 Explanations on Special Flags and Registers

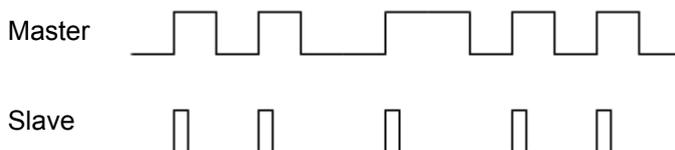
■ E-CAM max frequency: DD1840, M1749

1. D1841, D1840 (VI): E-CAM max frequency. If the registers are not set up properly, output function will not operate normally.
2. M1749: E-CAM max frequency control

M1749 = OFF, the Slave frequency will follow the Master frequency, but the max frequency will be 300kHz.



M1749 = ON, the max frequency of Slave will be specified by D1841, D1840(VI)



■ Enabling cyclic E-CAM / Completion of cyclic E-CAM: D1846 / M1813

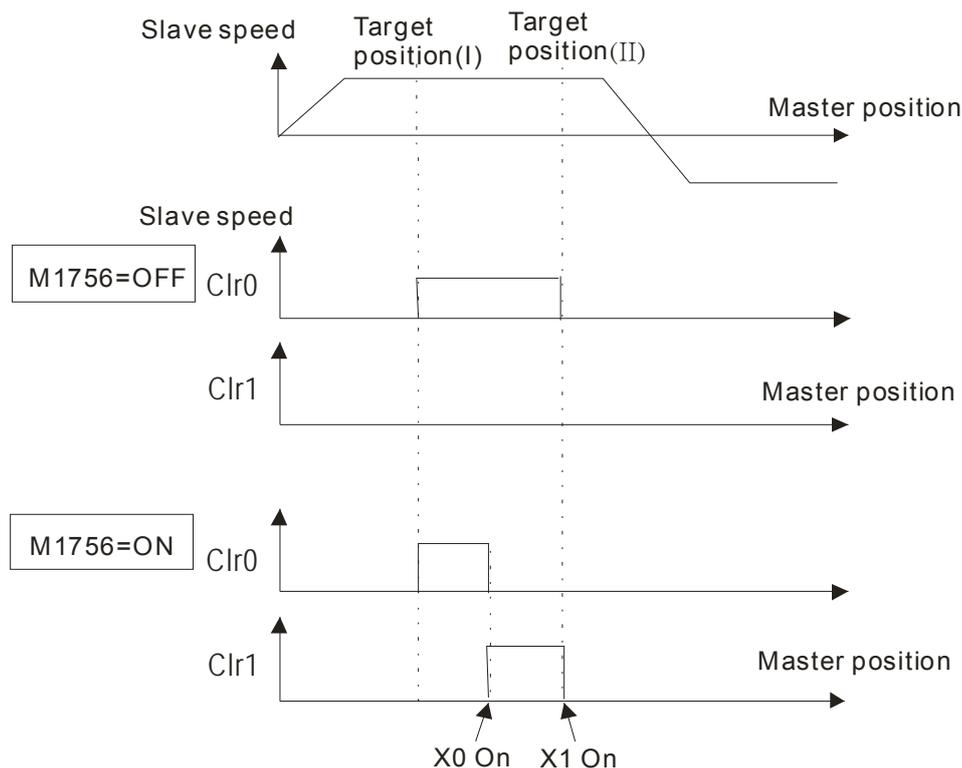
1. D1846: Enabling cyclic E-CAM. Bit 13 = ON, cyclic E-CAM is enabled; bit 13 = OFF, cyclic E-CAM is disabled.
2. M1813: Completion of cyclic E-CAM. M1813 will be ON when cyclic E-CAM is completed. To restart the cyclic E-CAM, the user needs to reset this flag.

■ Enabling acyclic E-CAM: D1846

D1846: bit 14 = ON, acyclic E-CAM enabled; bit 14 = OFF, acyclic E-CAM disabled.

■ Lower/higher bound of E-CAM synchronized output function: M1756, DD1838, DD1842

1. Operation Process when M1756 = ON: In flying saw / rotary cut application, set M1756 = ON to enable reference function. When Master position enters synchronized output section between PI and PII, CLR0 will be ON. When X0 is ON later, CLR0 switches off while CLR1 switches on. After this when X1 is ON, CLR1 will be OFF.
2. When M1756 is OFF with the condition: “D1839, D1838 (PI) \leq CP of Master \leq D1843, 1842 (PII)”, CLR0 will be ON.

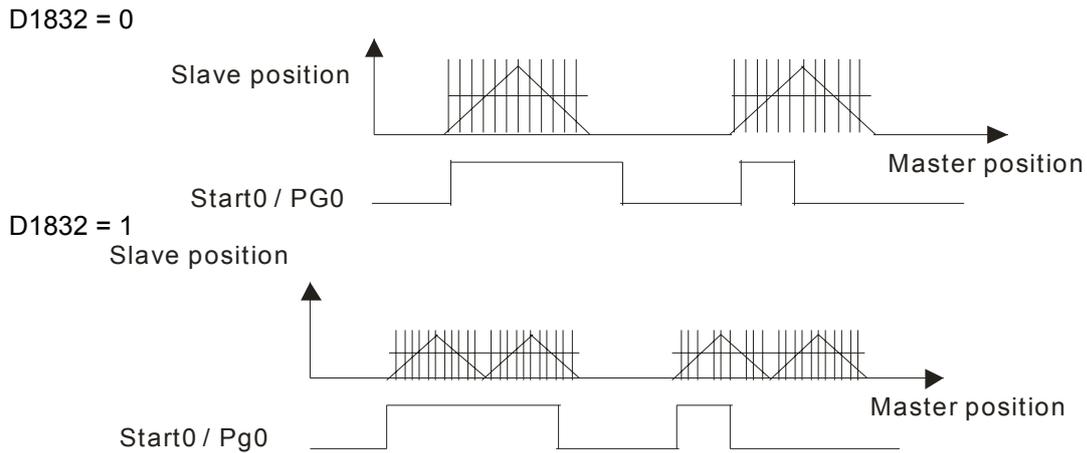


■ Completion of E-CAM initialization: M1812

When E-CAM is enabled, DVP-PM will initialize the settings for E-CAM operation and M1812 will be ON when initialization is completed. The flag needs to be reset by the user.

■ Times of acyclic E-CAM repetitions: D1832

Times of acyclic E-CAM repetitions can be set in D1832. However, when the set value is bigger than H8000 (b15=1), acyclic E-CAM will be switched into cyclic E-CAM automatically. In addition, if set value is 0, E-CAM Data will not be repeated. When set value is 1, E-CAM Data will be repeated once.

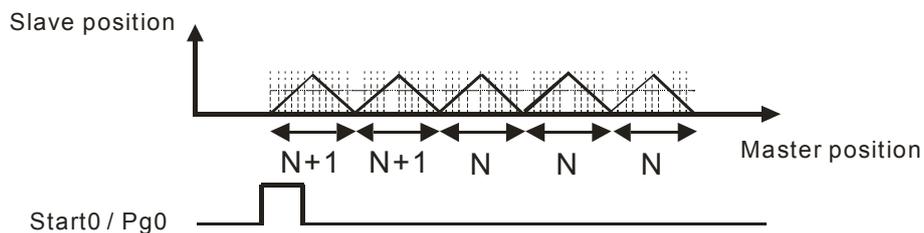


■ Distribution of remaining acyclic E-CAM pulses: D1833

Assume that Master operates 202 pulses and a single E-CAM Data requires 50 pulses, the execution results will be 4 cycles of E-CAM Data and 2 pulses remaining. In this case, set D1832 = 4(repeat for 4 times) and D1833 = 2. DVP-PM will execute one more pulse at the first and second intervals between the repeated E-CAM Data cycles, so as to properly distribute the remaining pulses. E-CAM Data in this condition will operate 5 times and only the first and the second cycle will be added with one more pulse as below.

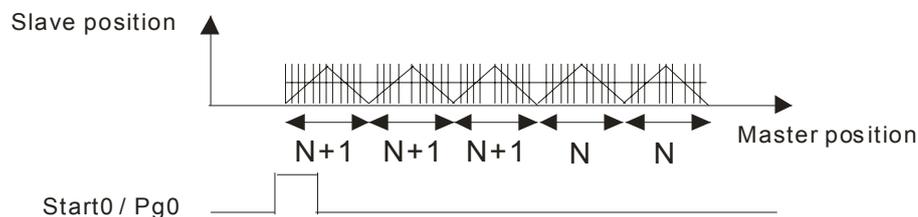
Note: The value in D1833 should be smaller or equal to D1832.

D1832 = 4, D1833 = 2



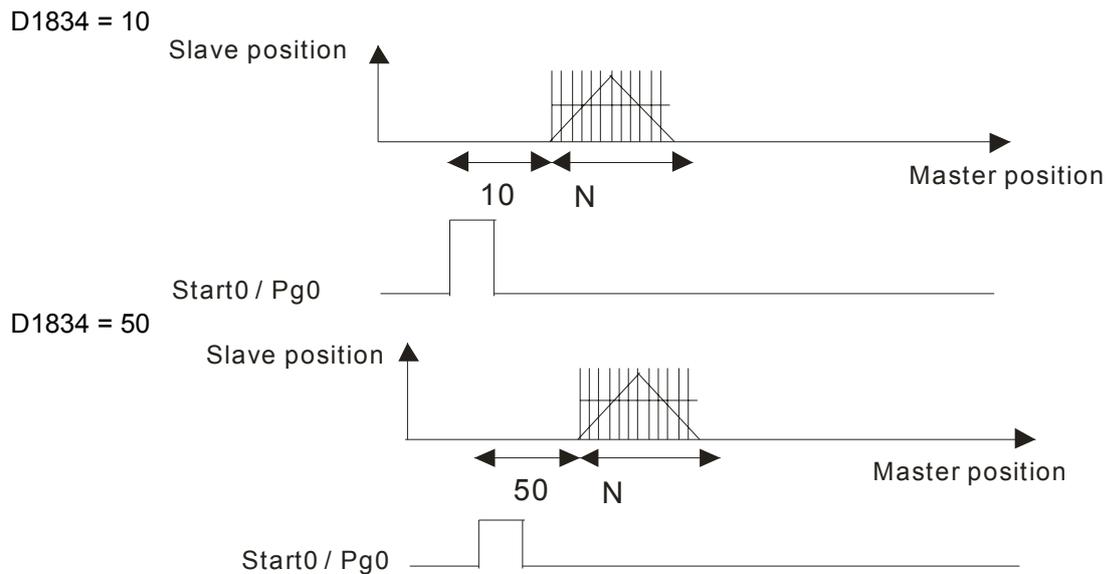
D1832 = 4, D1833 = 3

In the same way, DVP-PM will execute one more pulse at the 1st, 2nd and 3rd intervals between the repeated E-CAM Data cycles, so as to properly distribute the remaining pulses.



■ Pulses for delay start of acyclic E-CAM: D1834

The execution of acyclic E-CAM Data can be delayed by the number of pulses set in D1834. In diagram below, when PG0/START0 is triggered, E-CAM Data delayed for 10 or 50 pulses before its execution.



- Select the start signal of acyclic E-CAM: M1746
 1. When M1746 is OFF, the start signal of acyclic E-CAM will be START0; when M1746 is ON, PG0 will be the start signal of acyclic E-CAM.
 2. Before selecting START0 as the start signal, M1035 has to be set ON for setting STOP0/START0 as the external input. After M1035 is set OFF, M1746 = OFF will enable START0 as the start signal of acyclic E-CAM rather than PG0.

- Indicate E-CAM completion: M1748

When M1748 = ON, E-CAM stops immediately when E-CAM data is completed.

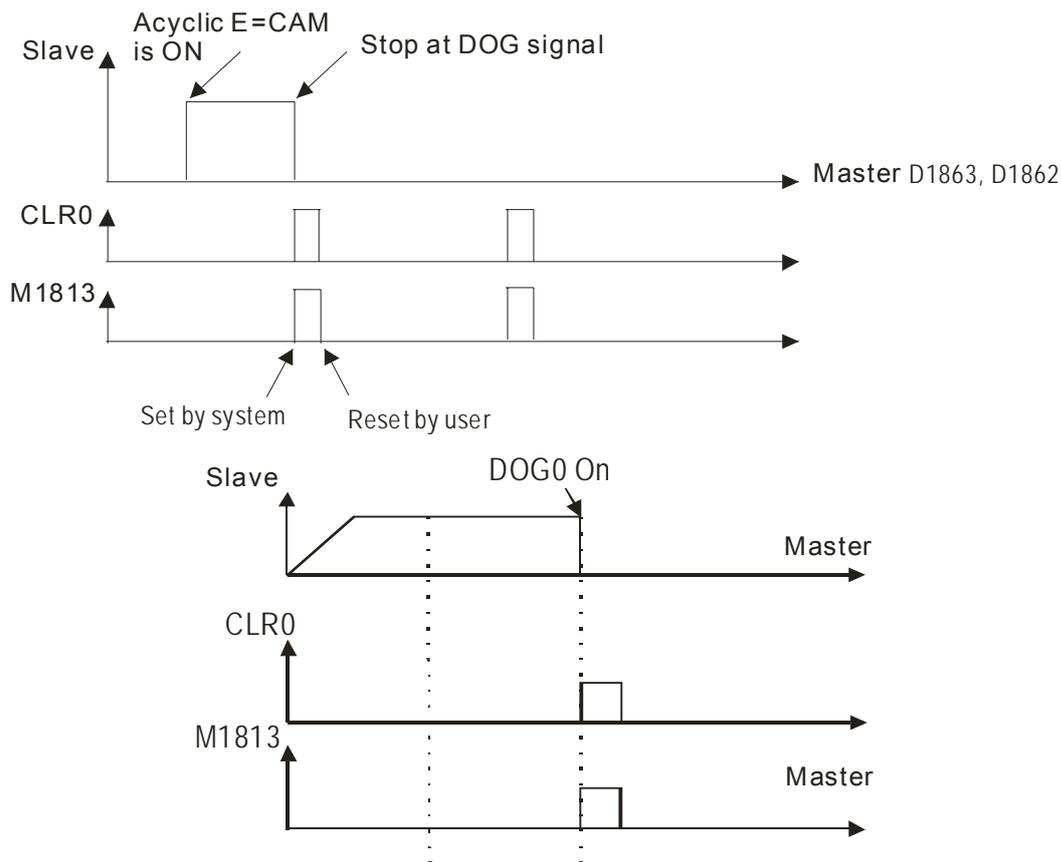
- Position and frequency of Master and Slave: DD1862, DD1860, DD1848
 1. Number of accumulated MPG input pulses / Master position: DD1862.
 2. MPG input frequency / Master frequency: DD1860.
 3. Current position / Slave position: DD1848.

- Select E-CAM Chart
 1. Every E-CAM chart includes a set of E-CAM data. There are 3 E-CAM charts available for 3 sets of E-CAM data.
 2. Before E-CAM executes, set up D1868 to select the E-CAM Chart 0~2 to be used.
 3. E-CAM Chart-0: D1868 = 0; E-CAM Chart-1: D1868 = H'8001,; E-CAM Chart-2: D1868 = H'8002.

- Acyclic E-CAM immediate stop function (Stop signal: DOG)

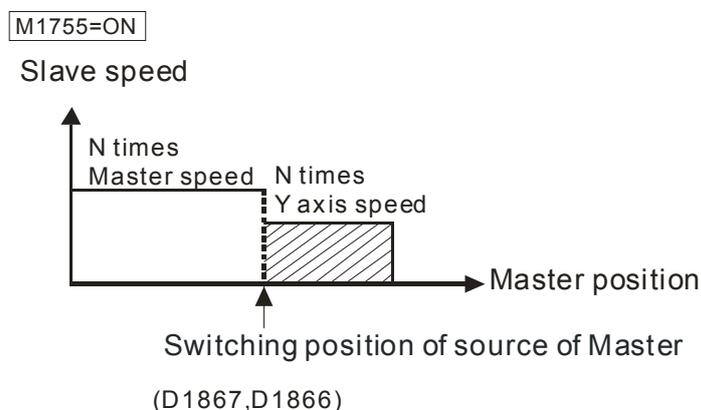
Acyclic E-CAM immediate stop function can be used together with M1755 for performing cutting with equalled deceleration speed and correct stop position at DOG. When M1752 is ON, acyclic E-CAM stops at DOG signal and CLR0 will be ON to clear the error of servo. Users have to reset M1813, so that CLR0 will be reset for next cutting operation.

9 Electrical CAM



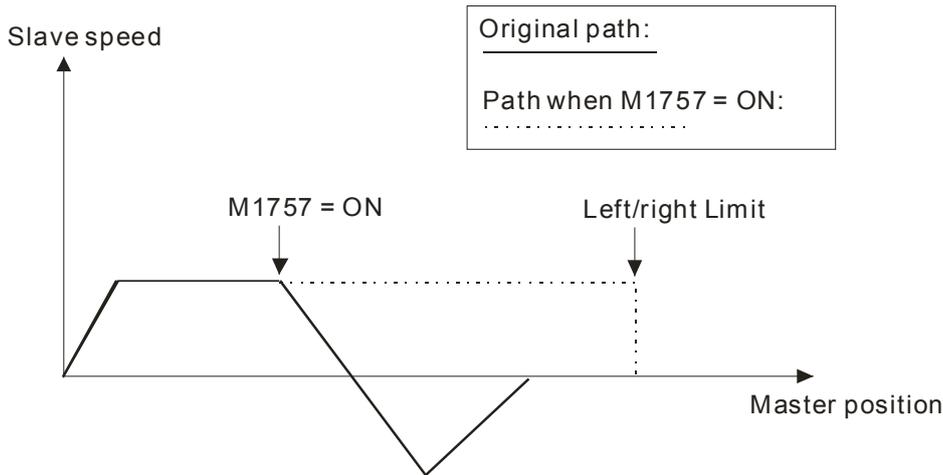
■ Switch the source of Master of acyclic E-CAM

For acyclic E-CAM, the source of Master can be switched to Y axis during the execution by setting up M1755. With M1755 = ON, source of Master will be switched automatically when Master position (DD1862) is bigger than the switching position (DD1866). Source of Master will be switched back to MPG0 after acyclic E-CAM cycle is completed. (Not applicable for virtual Master mode: M1909 / M1910). The function is generally used for obtaining equalled deceleration speed as well as correct stop position, regardless of the error resulted from Master speed or the characteristics of photo sensor.



■ Remain current speed until meeting limit switch: M1757

When M1757 is ON, Slave will remain current speed and continue operation until the right/left limit is reached.



9.4 Set up E-CAM Data

9.4.1 Use PMsoft CAM Chart to Set up E-CAM Data

The relationship defined by users between Master and Slave is called E-CAM Data.

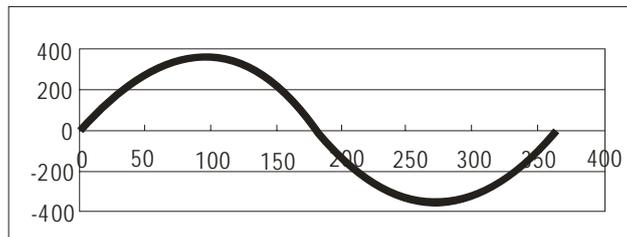
For describing E-CAM Data there are 2 methods:

- Method 1: Point-to-point relationship on X and Y axes.
- Method 2: Function, e.g. $Y = f(X)$

To obtain E-CAM data there are also 2 methods:

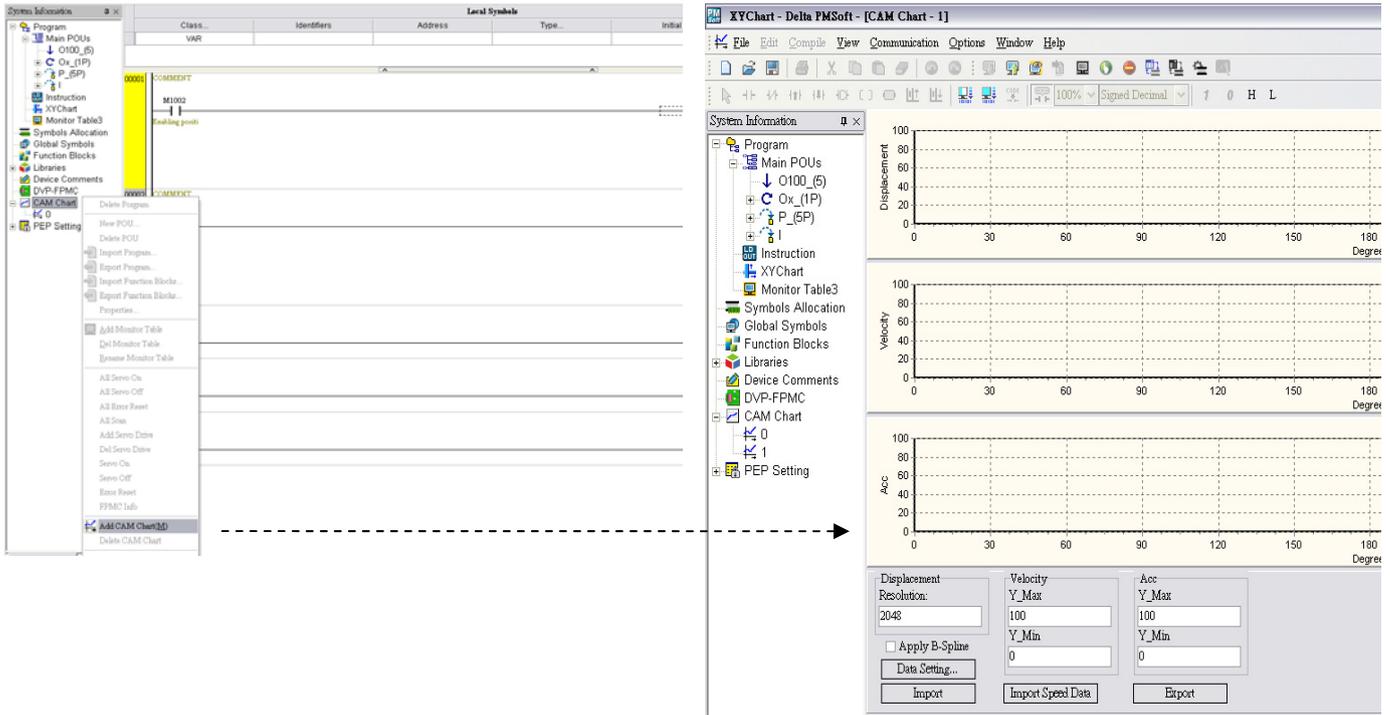
- Method 1: Obtain E-CAM Data according to the measured point-to-point values.
- Method 2: Obtain E-CAM Data according to the E-CAM Data function

E-CAM Data can be used to describe various CAM curves. After the relationship between Master and Slave is confirmed, we can obtain Slave position according to the Master position. For example, the below diagram illustrates an E-CAM Data of SIN wave.



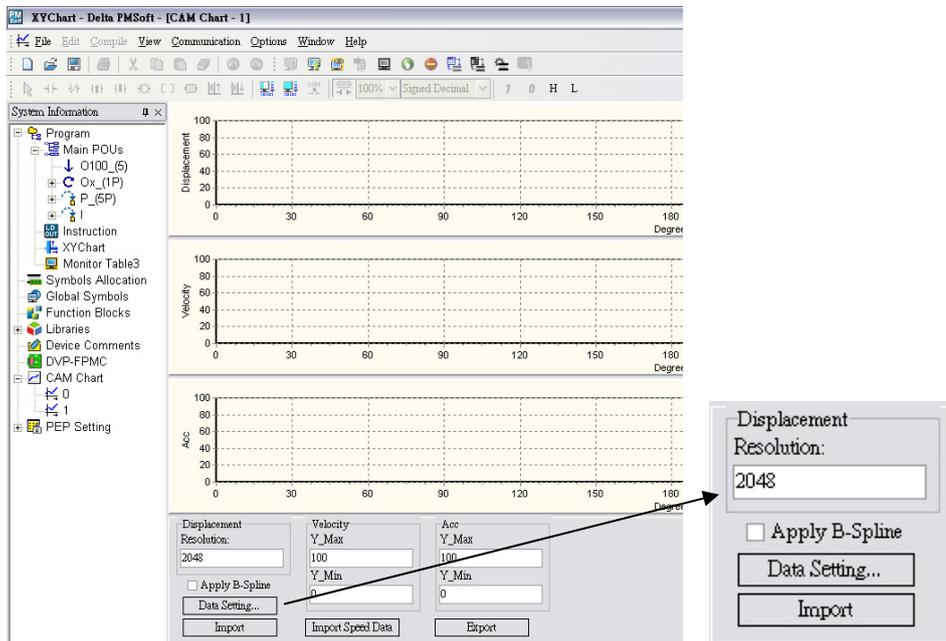
In PMSoft, E-CAM Data is represented as CAM Chart. You can add CAM Chart or delete CAM Chart by right-clicking at CAM Chart in System Information column.

9 Electrical CAM



There are 4 parts in CAM Chart: Displacement / Degree (relative position of Master/Slave), Velocity / Degree (relative speed of Master/Slave), Acc Speed / Degree (relative acceleration speed of Master/Slave), and Data Setting Area in the bottom.

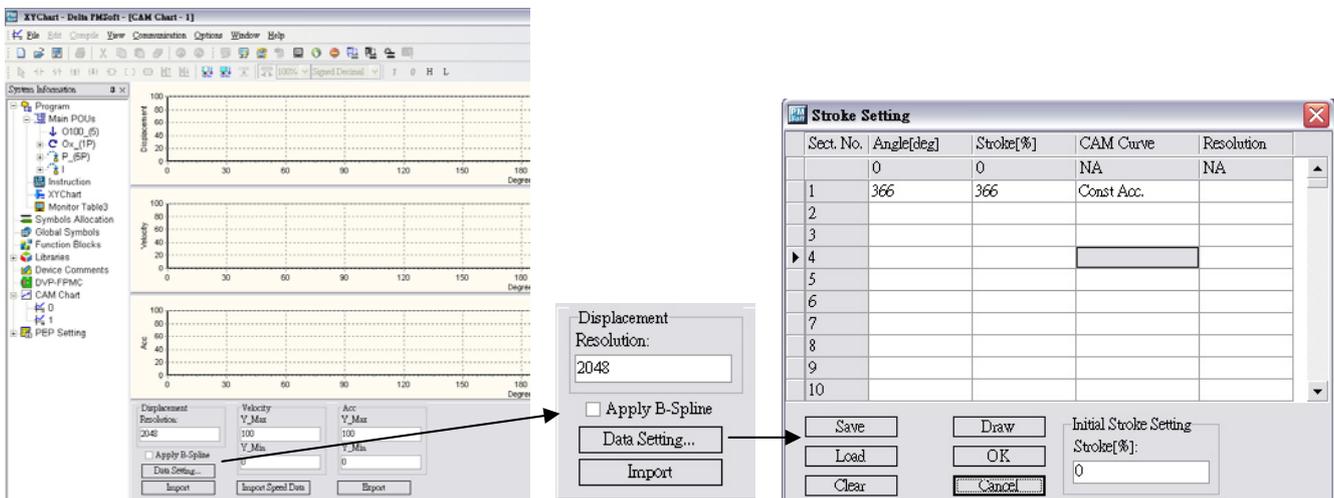
- The first 3 charts indicate user-defined E-CAM Data with horizontal axis as Master position, and vertical axis as Slave position, speed ratio of Slave/Master, and acc speed ratio of Slave/Master.
- Data Setting Area:
 1. Displacement Resolution: set up E-CAM Data resolution. Set range:10~2048 points. 1 point of E-CAM Data occupies 10 steps of user program area (Total 64k steps). Max 2048 points can be set up for a CAM Chart as below.



2. Velocity: automatically calculated max/min speed value
3. Acc: :automatically calculated max/min acc speed value
4. Data Setting: describe the variation of displacement between Master and Slave by a function.
5. Import: describe the variation of displacement between Master and Slave by point-to-point method.
6. Export: export the point-to-point relationship between Master and Slave as a file
7. Import Speed Data: describe the variation of speed between Master and Slave by point-to-point method.

9.4.1.1 Design the Variation of Displacement by CAM Curves

1. Click “Data Setting” in data setting area and the Stroke Setting window will pop up. You can describe the CAM curve by a Function by this window so that point-to-point description is not required here. In addition, DVP-PM currently provides users with 5 CAM curves: Const Speed, Const Acc., Single Hypot., Cycloid, and BSpline.

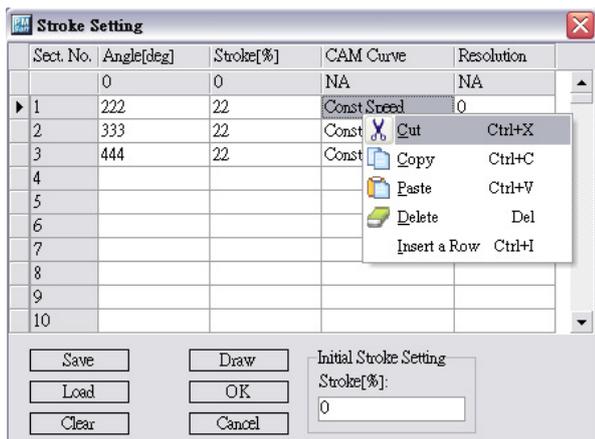


2. In Stroke Setting window, there are many sections which compose a complete CAM curve. Every section defines a part of the CAM curve and multiple sections form a full CAM curve. Settings for every section include Angle, Stroke, CAM Curve and Resolution as explained below:

- Angle: Degree of Master, must be an incremental value bigger than 0.
- Stroke: Displacement of Slave, can be a positive or negative value.
- CAM Curve: The applied function for current section
- Resolution: set up E-CAM Data resolution for current section. Set range for a full E-CAM Data:10~2048 points. 1 point of E-CAM Data occupies 10 steps of user program area (Total 64k steps). If resolution is not specified, PMSoft will automatically distribute the remaining points into blank columns. In addition, resolution should be specified according to the requirement of control devices. The operation is smoother with higher resolution but occupies more capacity (steps).

3. The editing on Stroke Setting window is similar to that on Excel of Microsoft Office, therefore you can copy, paste, or cut the cells by right-clicking on the sections.

9 Electrical CAM



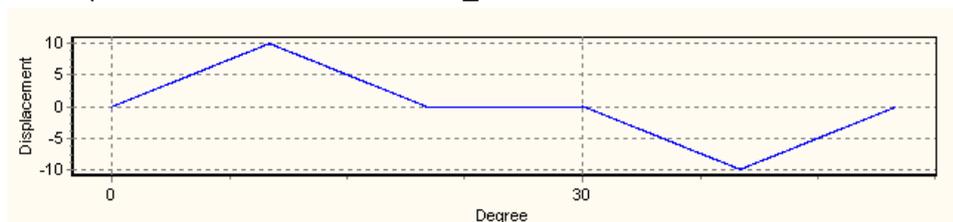
Explanations on buttons:

- Save: save the setting of current section
- Load: load the data into temporary data buffer
- Clear: clear all sections
- Draw: compile the sections and draw CAM Curves on CAM Charts
- OK: compile the sections, draw CAM Curves on CAM Chart and close the window..
- Cancel: close the Stroke Setting window.

- When the setting is completed, click OK to close the setting window and go back to the main page of CAM Chart.
- Click "Export", and PMSoft will store the CAM Data containing displacement data, velocity data and acc data in the subfolder "CAMData" under the main folder "PMSoft" as below:
 - <PMSoft>\CAMData\Data_S.txt: point-to-point displacement data
 - <PMSoft>\CAMData\Data_V.txt: point-to-point velocity data
 - <PMSoft>\CAMData\Data_A.txt: point-to-point acc data
- The CAM curves of each CAM data will also be drawn by PMSoft as below:

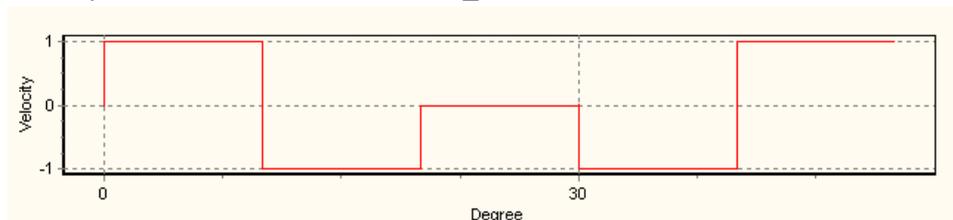
Displacement

Folder path: <PMSoft>\CAMData\Data_S.txt



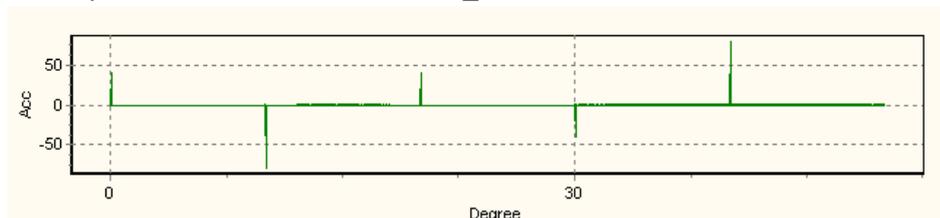
Velocity

Folder path: <PMSoft>\CAMData\Data_V.txt



Acc

Folder path: <PMSoft>\CAMData\Data_A.txt



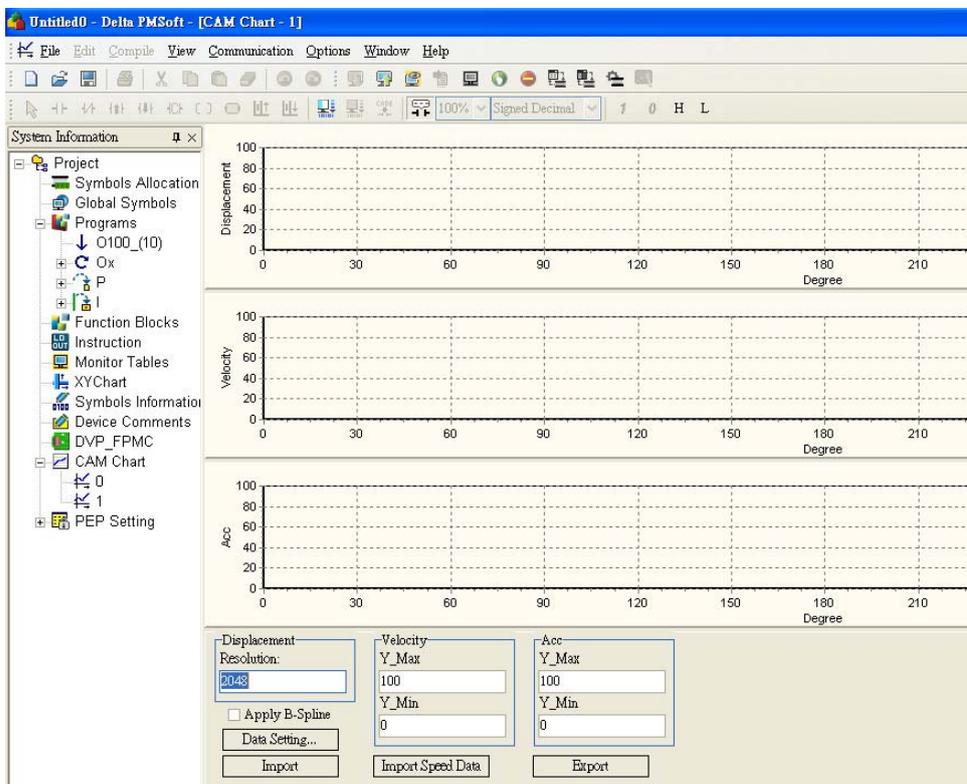
9.4.1.2 Design the Variation of Displacement by Point-to-point Data

You can input or modify the point-to-point data in the exported CAM data files and import the modified CAM data in PMSOft. In this way, the variation of displacement of the user-designed CAM curves will also be drawn in the CAM Chart page. The design process is as follows:

1. Input the point-to-point displacement data in the file Data_S.txt under the folder path:
<PMSOft>\CAMData\Data_S.txt



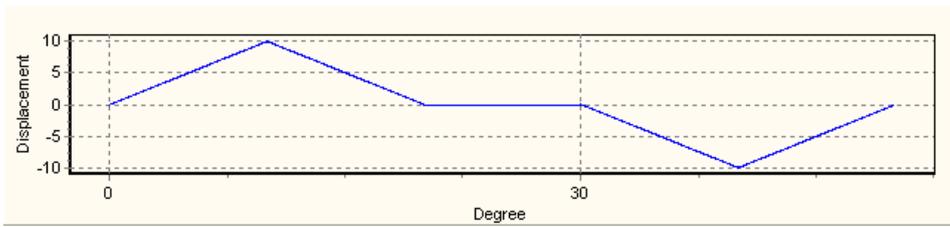
2. Save the modified file. Add a new CAM chart in PMSOft and enter the resolution.



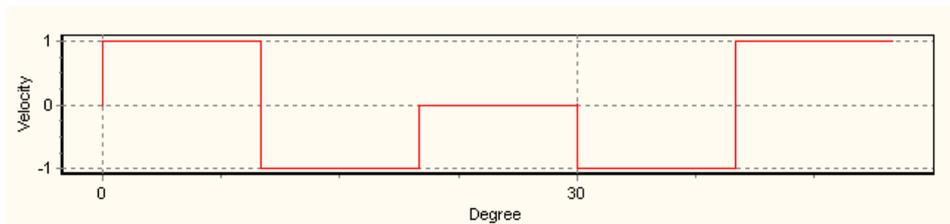
3. Click "Import". PMSOft will read out the content in Data_S.txt and draw the data on Displacement chart. Also, PMSOft will draw the Velocity and Acc charts based on the data of Data_S.txt.

9 Electrical CAM

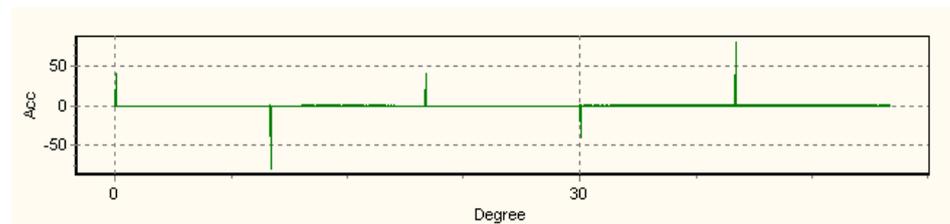
<PMSoft>\CAMData\Data_S.txt



Displacement



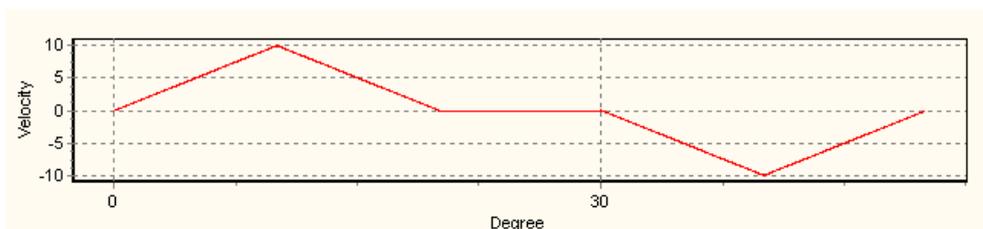
Velocity



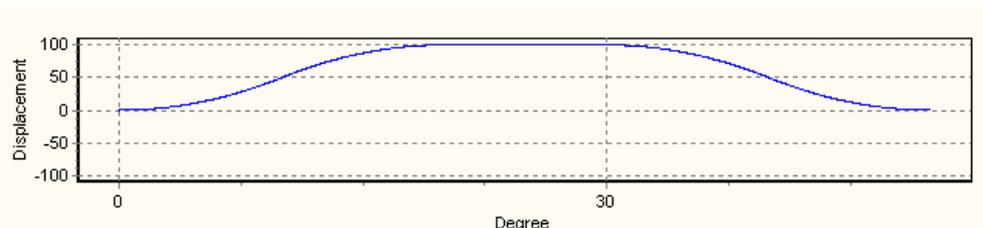
Acc

- Click "Import Speed Data". PMSoft will read out the content in Data_S.txt and draw the data on Velocity chart. Also, PMSoft will convert the velocity data and draw the converted Displacement and Acc charts

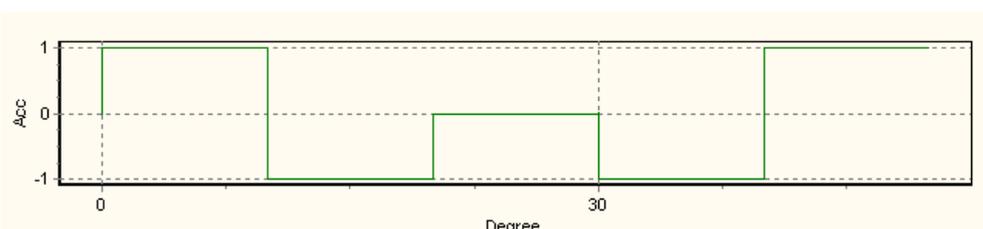
<PMSoft>\CAMData\Data_S.txt



Velocity



Displacement



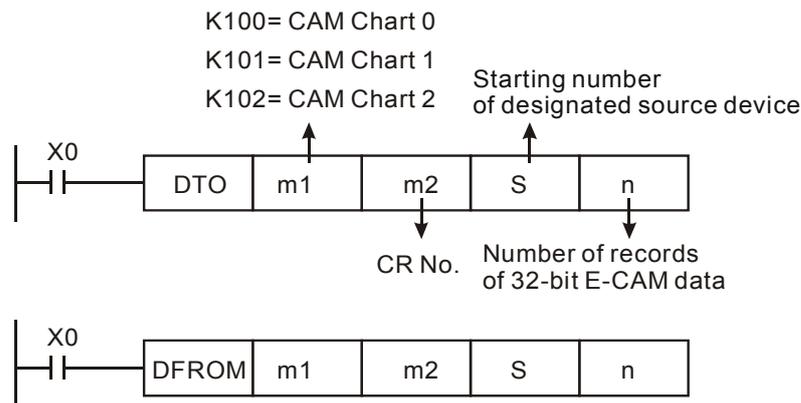
Acc

9.4.2 Use DTO / DFROM Instructions to Set up E-CAM Data

DVP-20PM is designed with 3 virtual modules exclusively for E-CAM0~ECAM2, and the numbers of each E-CAM chart are K100, K101 and K102. Through DTO/DFROM instructions, you can set up or modify the E-CAM data in the user program.

E-CAM Data created by PMSOft has been converted into floating point format because E-CAM data operates in floating point format. Therefore, the real-time modification data should be converted into binary floating point value first by DFLT instruction. In addition, floating point format should also be adopted when monitoring E-CAM Data in Monitor Table.

The format of DFROM/DTO instruction for setting up E-CAM data is as below. In E-CAM application, E-CAM0~E-CAM2 is regarded as #100~#102 special modules, i.e. for accessing E-CAM0, operand m_1 has to be set as K100. Operand m_2 represents the starting CR number, and the total number of CR will be a multiple of 4. Operand S refers to the starting number of designated source device. Operand n refers to the number of records to be written into CR. Every record of data consists of 2 words, and the set value of n should be a multiple of 2. Please note that the 16-bit TO/FROM instructions are not supported currently.



■ CR table of E-CAM:

CR#	Function	Data type	Data length
0~9999	Create / Modify single E-CAM data	DWord	1
10000	Create rotary cut E-CAM Data	DWord	10
10001	Read E-CAM status	DWord	7
10002	E-CAM Chart Real-time Change	DWord	1
10003	Smooth curve interpolation	DWord	3
10005	Read E-CAM status in Capture mode	DWord	5

■ Explanations on CR:

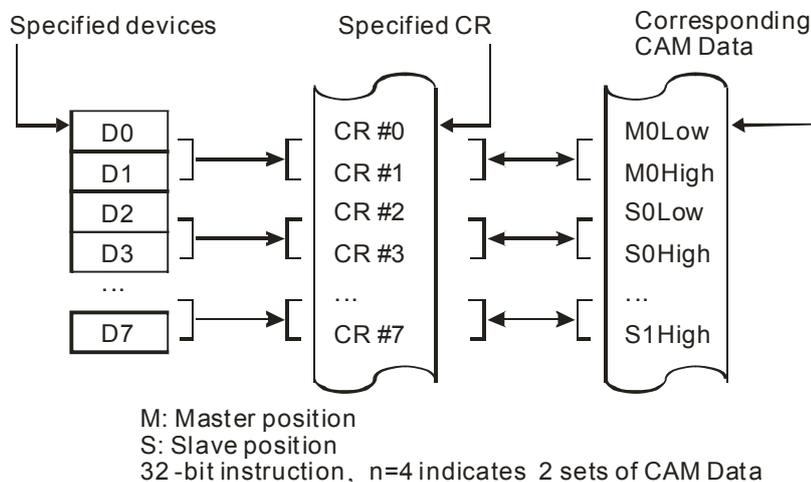
CR#0~9999: Create / Modify single E-CAM data

[Explanations]

DVP-20PM is built in with 3 sets of E-CAM charts: E-CAM Chart 0~E-CAM Chart 2. Every E-CAM chart supports

9 Electrical CAM

E-CAM data of 10,000 points and is set up by CR#0~9999. One set of E-CAM data consists of 2 points including Master and Slave position, and the unit of every point is Dword. For example, 2 sets of E-CAM data include 4 Master/Slave points which requires 8 CRs as the below structure. In addition, for details about single-data real time modification, please refer to 9.4.2.1.



CR#10000: Create rotary cut E-CAM Data

[Explanations]

DVP-20PM provides CR#10000 exclusively for creating rotary cut E-CAM. The user can create and apply the rotary-cut E-CAM by setting up the related machinery parameters according to actual needs. Parameters for rotary-cut E-CAM are as below:

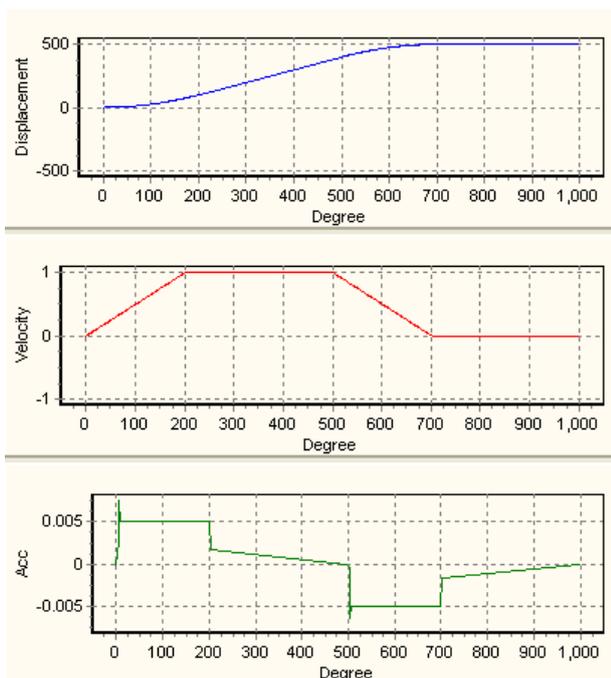
Parameters	Data	Format	Explanations
P1	Length of Master	Integer	Length of the material feeding axis. Unit:mm
P2	Length of Slave	Integer	Length of the cutting axis (cutter length included). Unit: mm.
	Sync start position		When sync function is enabled, the register sets up the sync start position.
P3	Length of sync area	Integer	Length of sync area. Unit: mm.
	Sync end position		When sync function is enabled, the register sets up the sync end position.
P4	Magnification ratio	Floating point	The speed of master and slave is synchronized in sync area. Calculation method of magnification ratio: $V_1 = V_2 \Rightarrow \frac{F_1 \times 3.14 \times D_1}{R_1} = \frac{F_2 \times 3.14 \times D_2}{R_2} \Rightarrow \frac{F_2}{F_1} = \frac{R_2 / D_2}{R_1 / D_1}$ where, V ₁ (V ₂) = Master(Slave) speed (mm/sec); F ₁ (F ₂) = Master(Slave) frequency (Hz); D ₁ (D ₂) = Master(Slave) diameter; R ₁ (R ₂) = Master(Slave) pulses of single round.
P5	Max magnification ratio	Floating point	
P6	Acceleration	Integer	E-CAM acceleration setting:

	curve (Low word)		0:const speed 1:const Acc 2:SingleHypot 3:Cycloid
	CAM curve (High word)	Integer	CAM curve selection for different sections in sync area, e.g. start, end: 0: leftCAM: Sync area is on the left side of CAM curve 1: midCAMall: Sync area is in the middle 2: midCAMbegin 3: midCAMend: (When the curve is selected, E-CAM stops automatically when completed) 5: rightCAM: Sync area is on the right side of CAM curve. Note: when CAM curve 2 or 3 is selected, the length of Master will be calculated by system. Sync area setting b[13]=1: Create sync area setting. When the function is enabled, P2 and P3 should be filled with start and end position of sync area. b[14]=1: Only single data real-time modification is applicable. b[15]=1: Borrow the data setting of previous E-CAM
P7	Results	Integer	Display the rotary cut E-CAM creating results (when read back): 0: OK 1: Input data is not proper 2: Insufficient length of E-CAM.

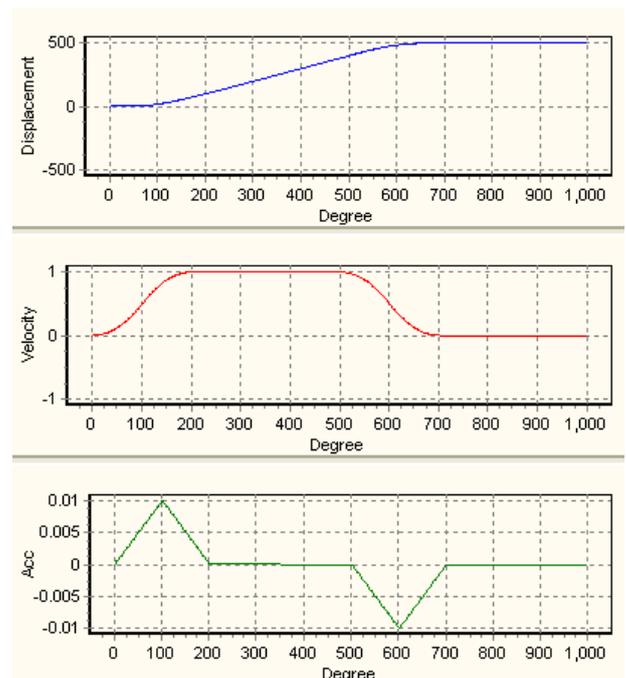
P6 sets up the rotary cut acceleration curves and the E-CAM curves, and the examples of each curve are as below:

Acceleration curves

0: const speed

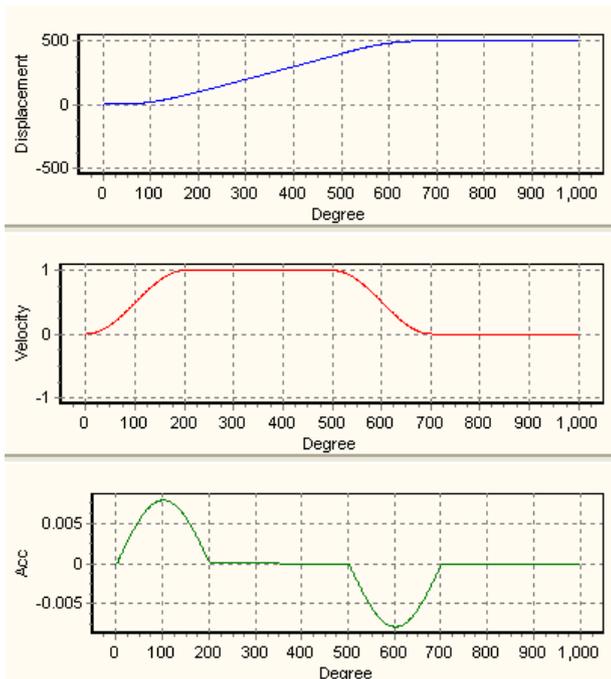


1: const Acc

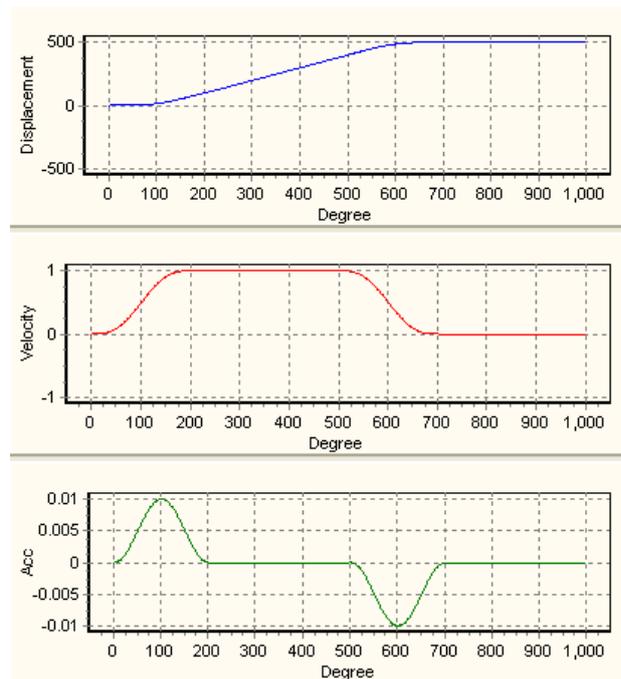


9 Electrical CAM

2: SingleHypot

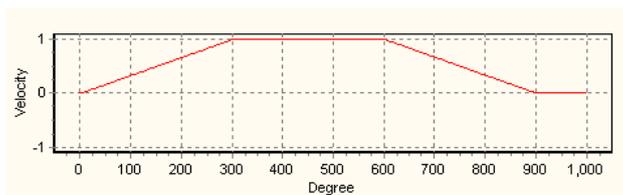


3: Cycloid

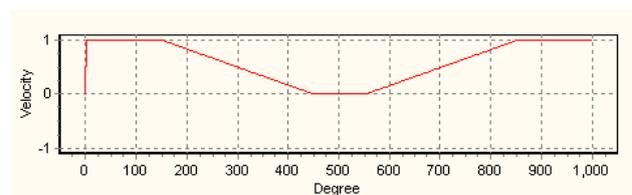


CAM curves

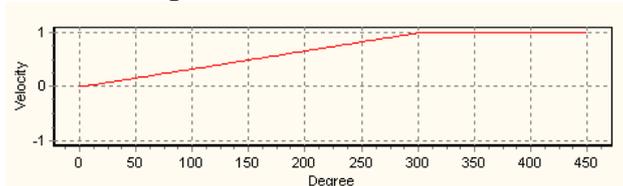
0: leftCAM



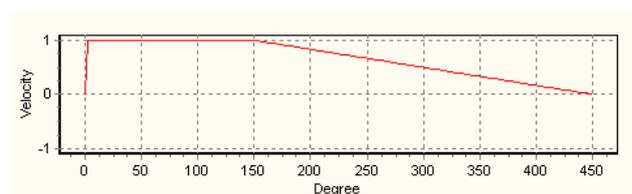
1: midCAMall



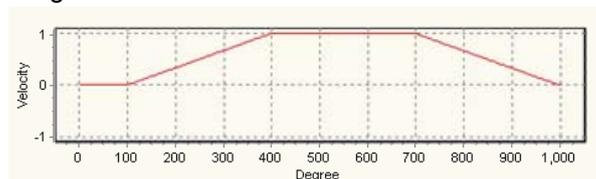
2: midCAMbegin



3: midCAMend



5: right CAM



For detailed explanation of creating rotary cut E-CAM curves, please refer to 9.4.2.3.

The rotary cut E-CAM data can be used for general rotary cut applications. However, for think material cutting application, the cutting angle and the cutter speed should be additionally considered on setting E-CAM.

DVP-20PM provides parameters for thick material cutting application as below:

Parameters	Data	Format	Explanations
P1	Thick-material cutting format	Integer	Thick-material cutting format setting: 0: divide non-sync area (when pulses can be equally divided by the drop times of cutter) 1: specify non-sync area (when pulses can NOT be

			equally divided by the drop times of cutter) b[2]=1: Specify the number of data points in sync area b[3]=1: Borrow the data setting of previous E-CAM b[4]=1: Non-smooth acceleration/deceleration for fixing the resolution.
P2	Pulses per round of Master	Integer	Unit: Pulse
P3	Length per round of Master	Integer	Unit: mm
P4	Pulses per round of Slave	Floating point	Unit: Pulse
P5	Length per round of Slave	Floating point	Unit: mm
P6	Sync start angle of Slave	Integer	Set range: 180~360 degree
P7	Sync end angle of Slave	Integer	Set range: 180~360 degree
P8	Length of cutting process on Master	Integer	Unit: mm
P9	Number of cutters on Slave	Integer	Number of cutters applied in rotary cut application
P10	Length of Master before sync area	Integer	When P1 is set as 1, the parameter sets the length of Master before sync area.
P11	Angle between start position and sync start: angle θ_1 (degree)	Integer	When P1 is set as 1, the parameter sets the angle before Sync Start.
P12	Number of data points in sync area	Integer	When b[2] of P1 is 1, the parameter sets the number of data points in sync area.

For detailed explanation of creating thick material cutting E-CAM curves, please refer to 9.4.2.4.

CR#10001: Read E-CAM status

[Explanations]

The function is used to read current E-CAM status when E-CAM is executing. Only DFROM instruction is applicable for this function. The read data will be as follows:

Parameters	Data	Format	Explanations
P1	Current position of Master	Integer	Reset at DOG signal
P2	Length of Master 1	Integer	DOG signal to DOG signal
P3	Current E-CAM curve	Integer	
P4	Current position of Slave	Integer	Updated at DOG signal
P5	Error of Slave	Integer	Error between DOG signal and the end point of CAM cycle.
P6	Length of Master 2	Integer	START0/PG0 signal to START0/PG0 signal
P7	Error of Master	Integer	Error between START0, PG0 signal and the end point of CAM cycle

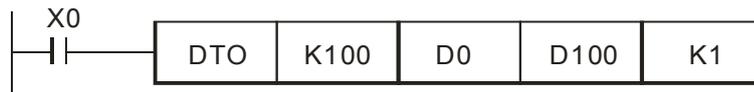
The function of reading E-CAM status can be applied in offset compensation to Slave error. For the methods of Slave error compensation, please refer to 9.4.2.5.

9 Electrical CAM

CR#10002: E-CAM Chart Real-time Change

[Explanations]

There are 3 built-in E-CAM charts in PMSoft: E-CAM Chart-0~E-CAM Chart-2. If you need to conduct real-time change of E-CAM chart during program execution, set up this register to designate the E-CAM chart to be enable in the next cycle. Care should be taken on the number of E-CAM charts, which should be as least 2 sets available for the change. The instruction below is the application example. Set D0 = K10002 before the E-CAM Chart real time change. When X0 = ON, set value (0, 1, 2) in D100 indicating the CAM Chart to be selected (E-CAM Chart-0~E-CAM Chart-2) will be written into DVP-PM.



CR#10003: Smooth Curve Interpolation

[Explanations]

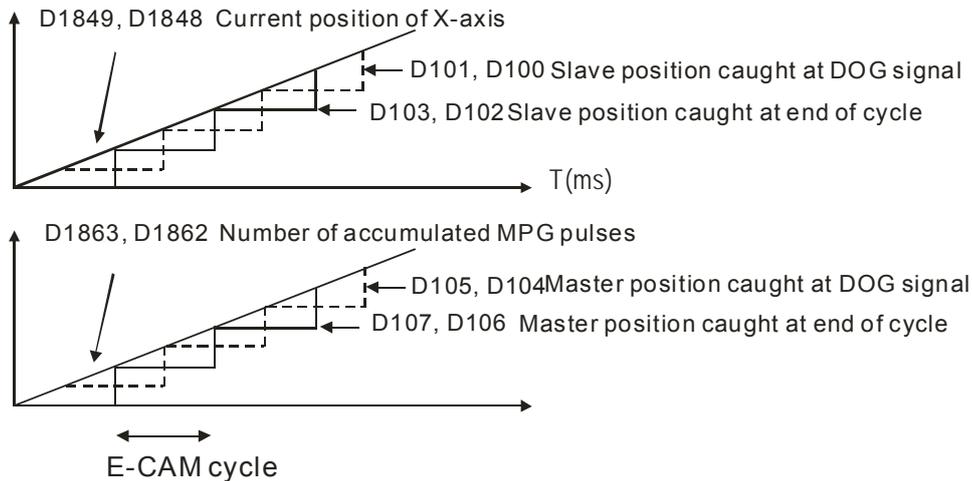
When the original data points are not sufficient for a smoother operation, you can apply Smooth Curve Interpolation. The detailed content of smooth curve interpolation will be explained in 9.4.2.2. Parameters for setting up the function are listed as below.

Parameters	Data	Format	Explanations
P1	Length of Master	Integer	Length of master can be modified according to actual needs.
P2	Original data points	Integer	Data points before interpolation
	Total points to be inserted	Integer	Number of points to be equally inserted into the sections between original points.
P3	Smoothing coefficient	Integer	Times of smoothing curve operation 3: 3 times 4: 4 times 5: 5 times 6: 6 times Other: 3 times
	Cycloid types	Integer	0: Uniform B-Spline 1: Non Uniform B Spline 2: C-spline (original data points should be less than 20 points) 3: CC Spline 4: Cycloid

CR#10005: Read E-CAM status in Capture mode

[Explanations]

Parameteres	Data	Format	Explanations
P1	Slave position 1	Integer	Caught at DOG signal
P2	Slave position 2	Integer	Caught at end of cycle
P3	Master position 1	Integer	Caught at DOG signal
P4	Master position 2	Floating point	Caught at end of cycle
P5	Master position 3	Floating point	Caught at START0(M1746=OFF) or PG0(M1746=ON)



The function of reading E-CAM status in capture mode should be used with M1757 (Remain current speed), and can be applied for checking the cutter position in flying saw applications. For details please refer to 9.4.2.6.

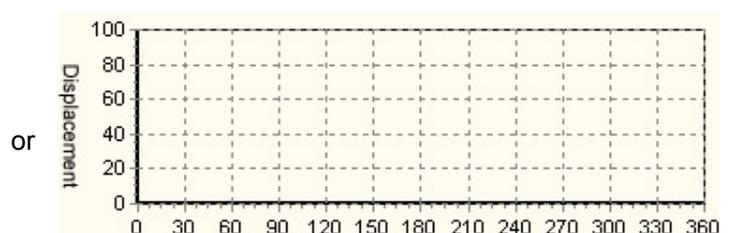
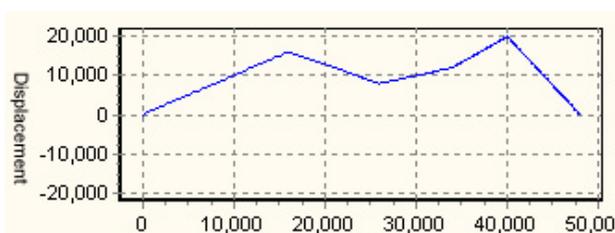
9.4.2.1 E-CAM Data Real-time Modification

E-CAM Data set in E-CAM Chart will be downloaded into DVP-PM with the user program. Due to that E-CAM Data can only be modified by PMSOFT, users have to download E-CAM Data again after modification. Aiming at this problem DVP-PM also provides users with E-CAM Data real-time Modification function by DTO/DFROM instruction

Data length of E-CAM Data can be real-time modified. When 4 words of one point (set) of E-CAM Data are all 0 and identified, DVP-PM will take it as E-CAM completion. Therefore, if you want to modify 10 points E-CAM Data into 5 points, DTO instruction can be applied to write 0 into the 4 words of the 6th point.

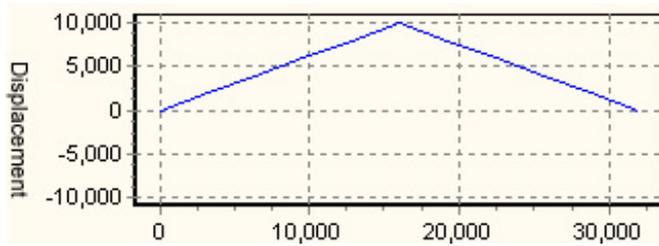
■ Application Example

Modify the original E-CAM Data (existed data or no data) as diagram (a) into the 3-point E-CAM Data as Diagram (b) with coordinate (0, 0), (16000, 10000) and (32000, 0).



9 Electrical CAM

(a) Multiple-point E-CAM Data or no data



(b) 3-point E-CAM Data

Control Steps:

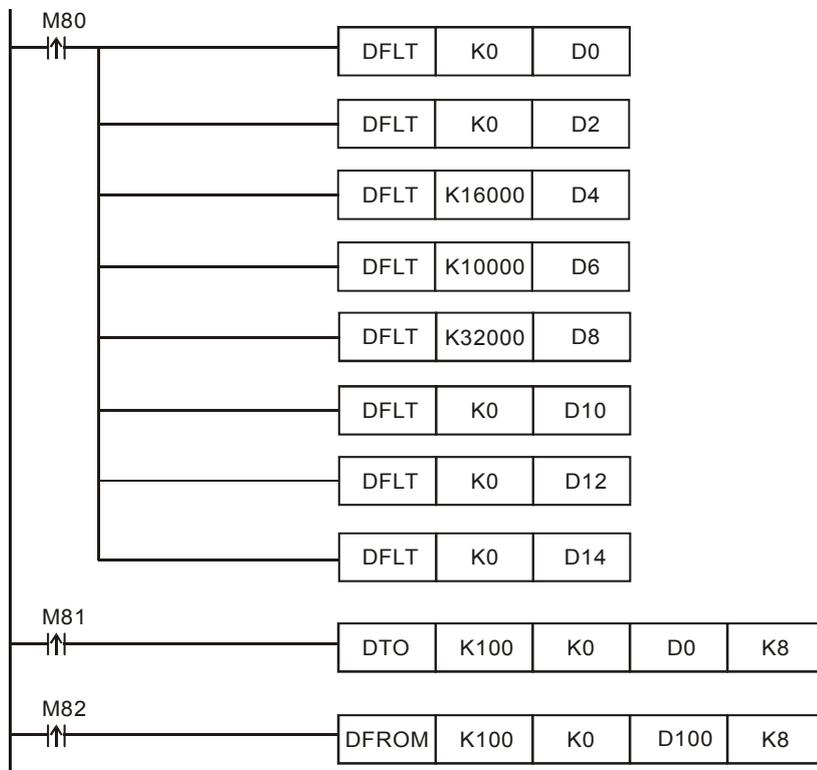
Step 1: Set M80 to convert the 4-point E-CAM data (3-point data with an additional point 0) into binary floating point format, and store the results in D0~D15.

Step 2: Set M81 to write the data in D0~D15 into E-CAM chart-0 (K100).

Step 3: Set M82 to read the written data back into D100~D115 so as to confirm the data is correctly written into E-CAM chart-0.

Step 4: Use Monitor Table to confirm the data read back in D100~D115. Data format (Radix) of D100~D115 should be set as floating point.

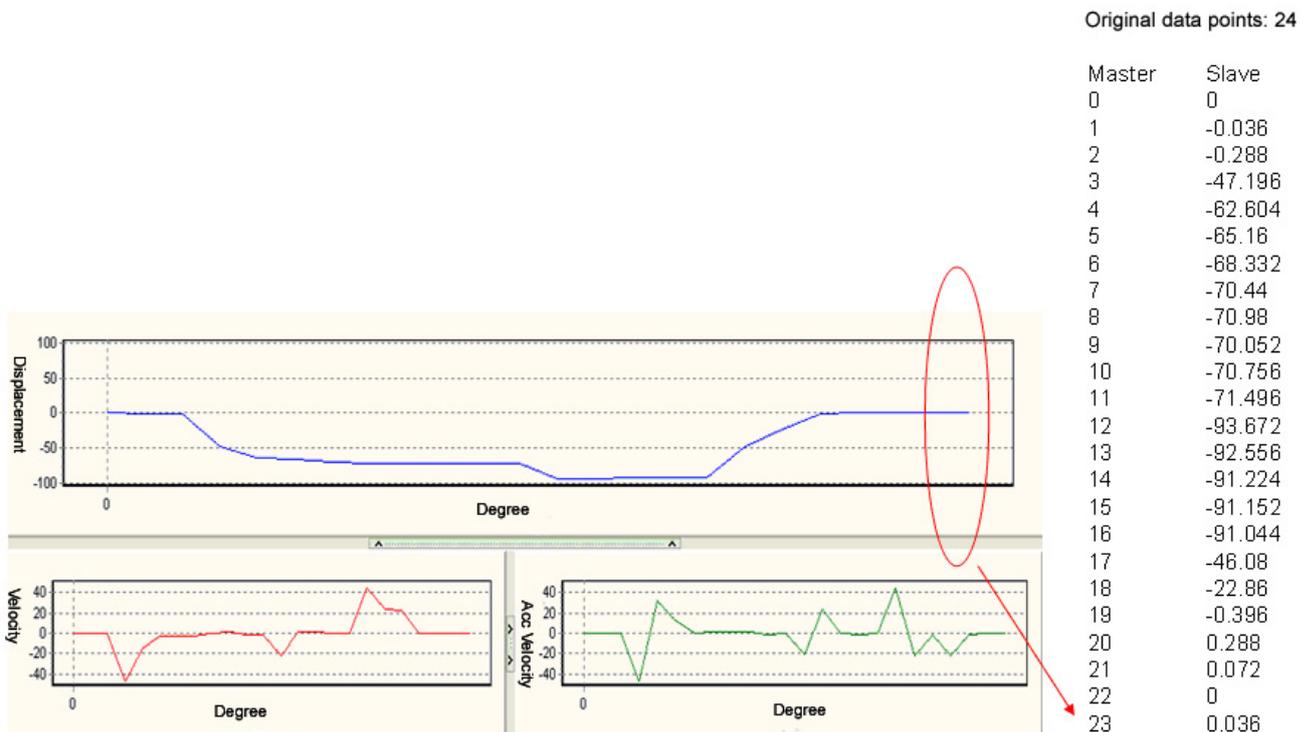
Ladder Diagram:



9.4.2.2 Smooth Curve Interpolation

Smooth curve interpolation conducts interpolation and magnification on data points of Master, so as to smoothen the modified curve with higher resolution. The interpolation results will elevate the stability and smoothness for machine operation.

In the sample below, the original data points are 24 points, and the modified length of Master will be 10,000 points. In this case, the magnification rate of point-to-point section will be $10,000/23=434$. On this basis, PMSOft will automatically calculate the data of the inserting points according to the number of inserting points and the smoothing coefficient. After this, the modified 10,000 points E-CAM curve will be drawn according to the calculated results in the same proportion of the previous E-CAM curve.



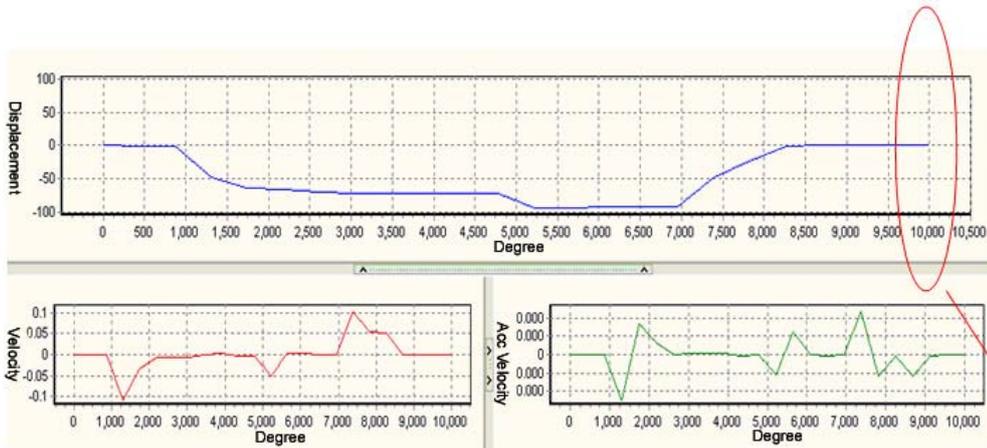
After the interpolation, the original points of Master 0~23 are modified as 10,000 points as below.

9 Electrical CAM

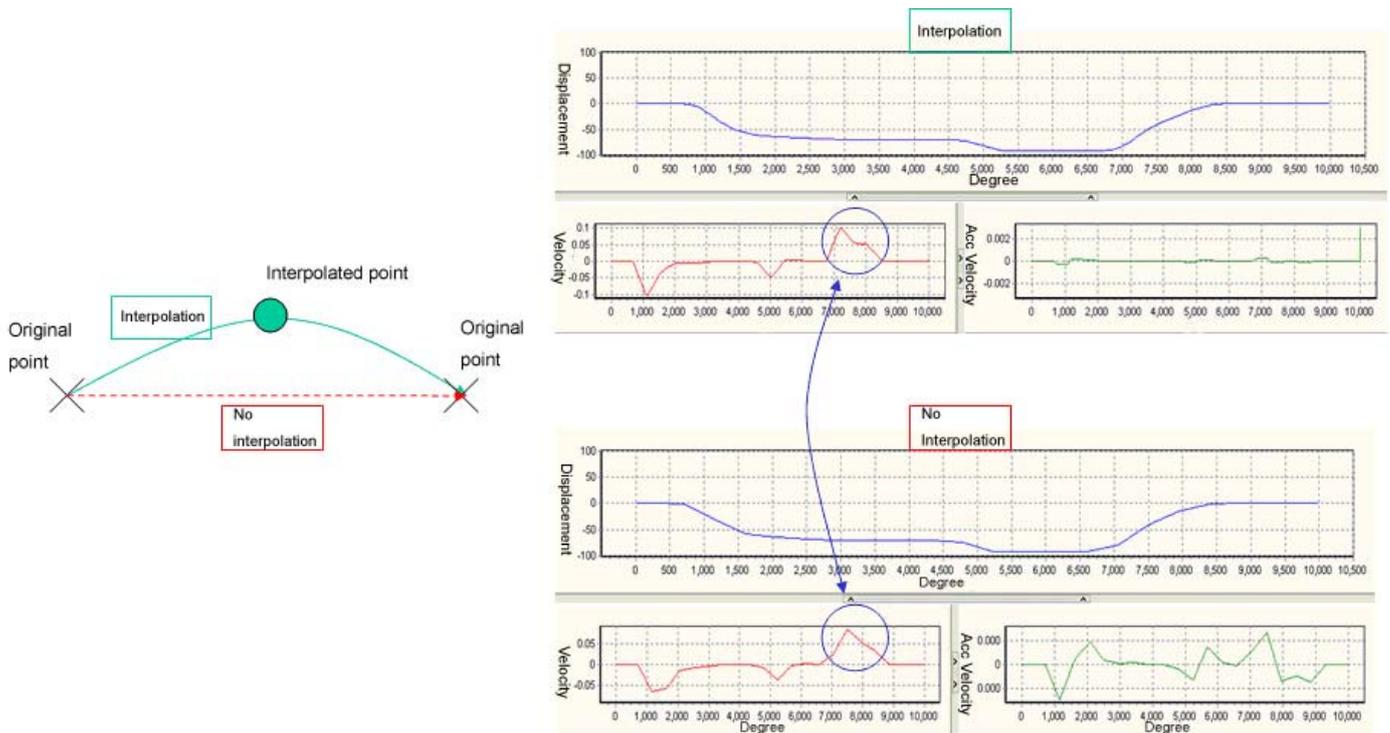
Original data points: 24

Modified data points: 10000

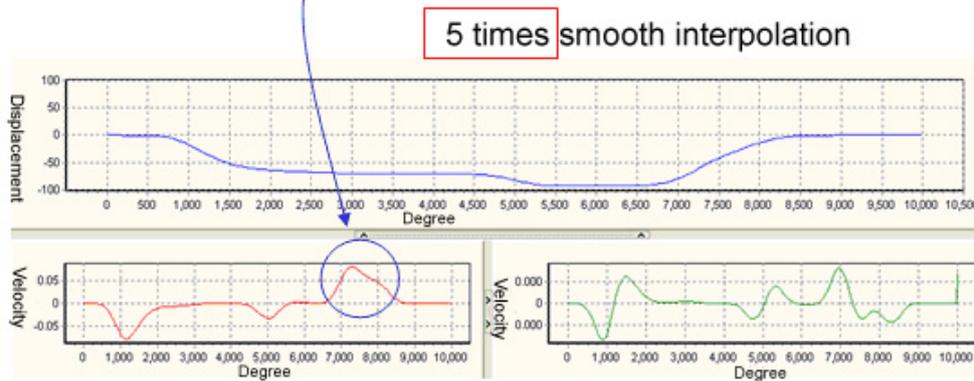
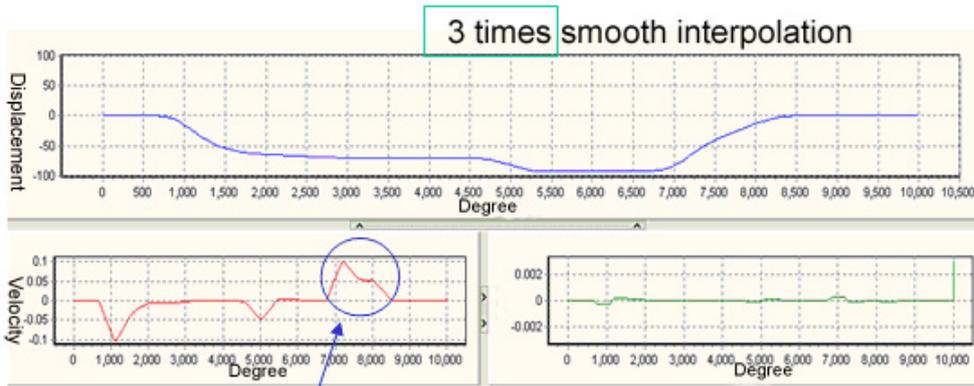
Master	Slave
0	0
434	-0.036
869	-0.288
1304	-47.196
1739	-62.604
2173	-65.16
2608	-68.332
3043	-70.44
3478	-70.98
3913	-70.052
4347	-70.756
4782	-71.496
5217	-93.672
5652	-92.556
6086	-91.224
6521	-91.152
6956	-91.044
7391	-46.08
7826	-22.86
8260	-0.396
8695	0.288
9130	0.072
9565	0
10000	0.036



The below diagram illustrates the difference between the original curve and the curve with interpolation.



With higher smoothing coefficient, the generated curve will be smoother as well. The below diagrams show the results of 3-times and 5-times interpolation respectively.

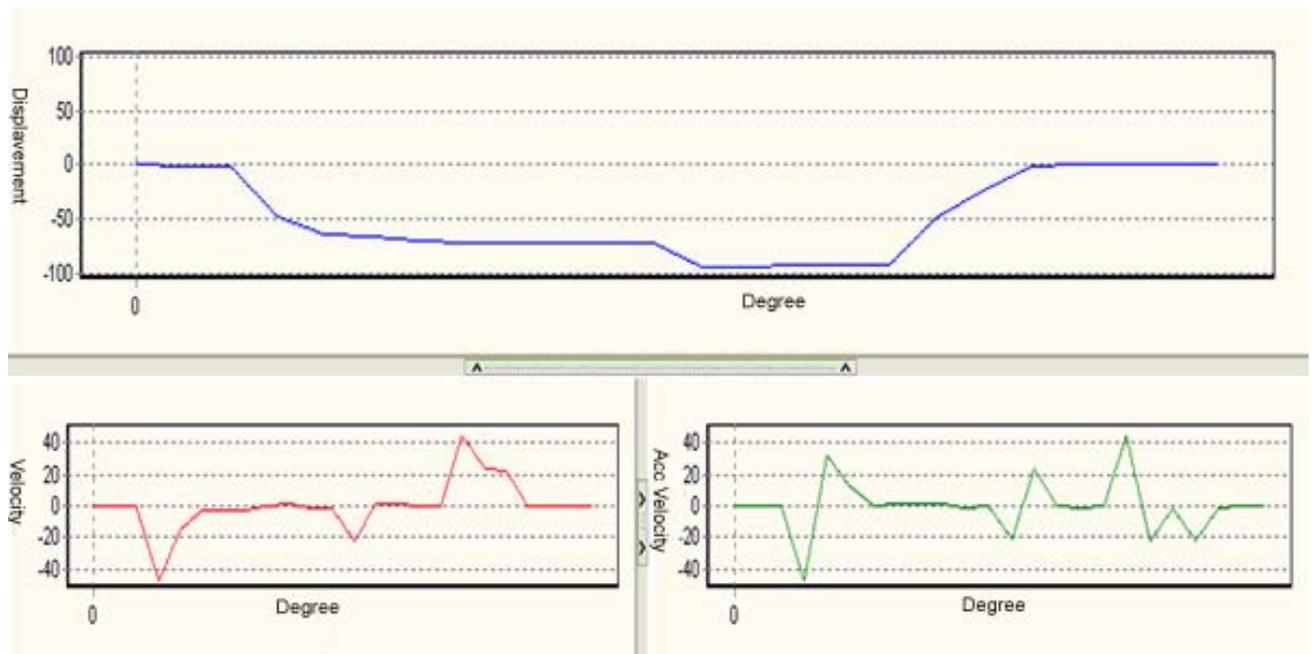


■ Application Example

Conduct smooth curve interpolation on the original curve of E-CAM Chart-0:

Original data points: 24; Length of Master: 10,000; Total points to be inserted: 200 points; Smoothing coefficient: 5 times; Cycloid type: Uniform B-Spline.

The original curve is as below:



Control Steps:

Step 1: Create a 23-point E-CAM curve in E-CAM Chart-0. Use the import function to set up the E-CAM data

9 Electrical CAM

as below. For the set up methods please refer to section 9.4.

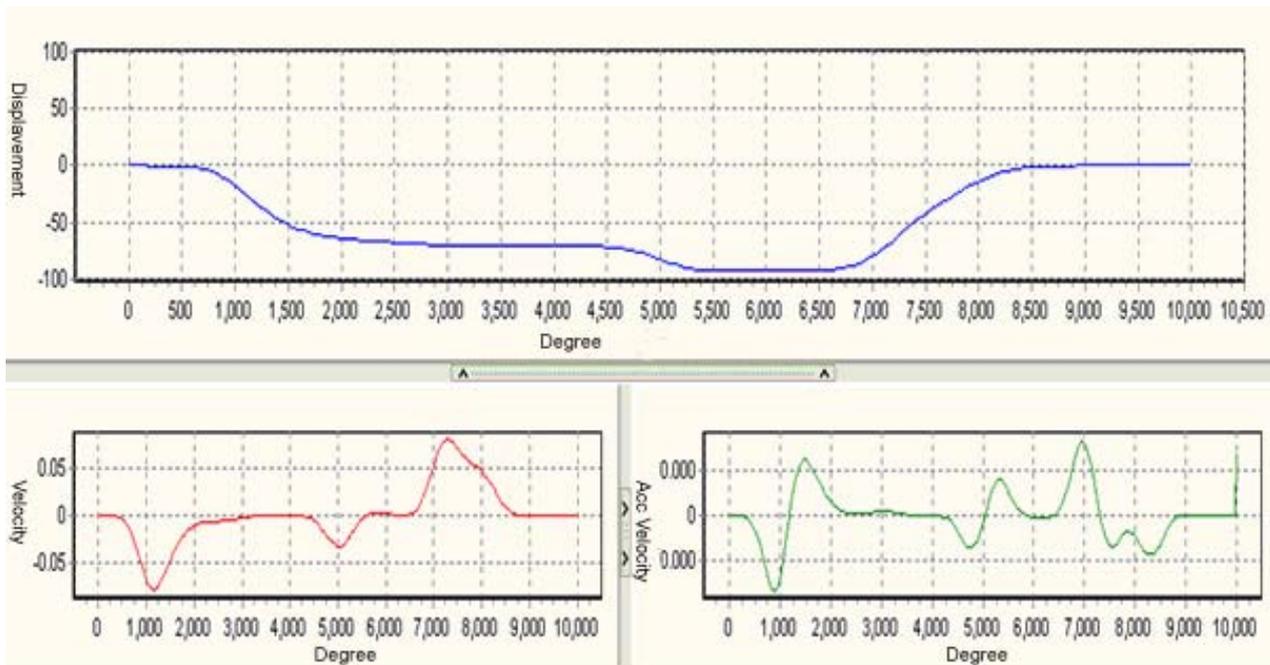
X	Y	x	y	x	y	x	y
0	0	6	-68.332	12	-93.672	18	-22.86
1	-0.036	7	-70.44	13	-92.556	19	-0.396
2	-0.288	8	-70.98	14	-91.224	20	0.288
3	-47.196	9	-70.052	15	-91.152	21	0.072
4	-62.604	10	-70.756	16	-91.044	22	0
5	-65.16	11	-71.496	17	-46.08	23	0.036

Step 2: Set the resolution of E-CAM Chart-0. The resolution should be equal or bigger than “Original data points + Total points to be inserted.”

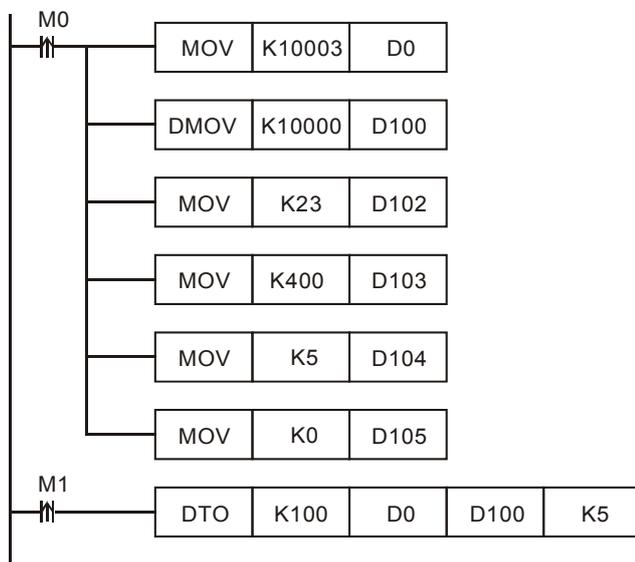
Step 3: Set M0 to write the parameters for smooth curve interpolation into registers D100~D105

Step 4: Set M1 to write in the data in D100~D105 for executing smooth curve interpolation.

Step 5: Execute the program and monitor the E-CAM curve in E-CAM Chart-0. The curves after interpolation will be as below:



Ladder Diagram:



9.4.2.3 E-CAM Data for Rotary Cut Function

E-CAM Data for rotary cut function is set up according to the parameters of connected devices and the control requirements. Users should input sufficient data into corresponding registers and DVP-PM will establish E-CAM Data for rotary cut function. There are 2 steps for setting up rotary cut E-CAM Data.

Step 1: Create E-CAM Charts

Use PMSoft to create 2 E-CAM Data of resolution 300 in E-CAM charts. Download the 2 E-CAM Data to DVP-PM. This step is to create 2 sets of E-CAM Data space of resolution 300 for storing the operating rotary cut E-CAM data.

The screenshot shows the PMSoft software interface. On the left, a tree view shows 'DVP_FPMC' with sub-items 'CAM Chart' and 'PEP Setting'. A context menu is open over 'CAM Chart', listing options like 'Delete Program', 'New POU...', 'Delete POU', 'Import Program...', 'Export Program...', 'Import Function Blocks...', 'Export Function Blocks...', 'Properties...', 'New Folder', 'Delete Folder', 'Rename Folder...', 'Add Monitor Table', 'Del Monitor Table', 'Rename Monitor Table', 'All Servo On', 'All Servo Off', 'All Error Reset', 'All Scan', 'Add Servo Drive', 'Del Servo Drive', 'Servo On', 'Servo Off', 'Error Reset', 'Servo Drive Info', and 'Add CAM Chart'. A white arrow points to the 'Add CAM Chart' option. To the right, two empty E-CAM charts are displayed. The top chart is for Velocity (Y-axis: 0 to 100) vs. Degree (X-axis: 0 to 180). The bottom chart is for Acceleration (Y-axis: 0 to 100) vs. Degree (X-axis: 0 to 180). Below the charts, there are input fields for 'Displacement Resolution' (set to 300), 'Velocity Y_Max' (100), 'Velocity Y_Min' (0), 'Acc Y_Max' (100), and 'Acc Y_Min' (0). There are also buttons for 'Data Setting...', 'Import', 'Import Speed Data', and 'Export'.

Step 2: Set up and input the rotary cut E-CAM Data

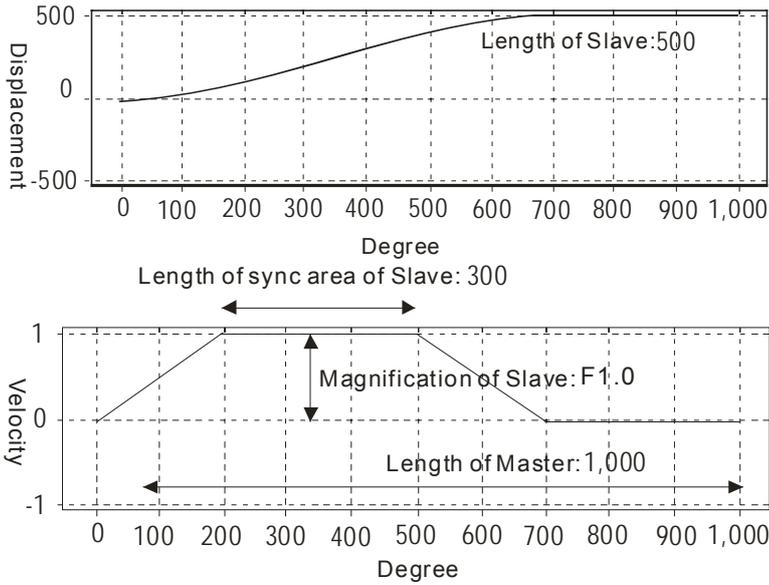
Input the machine parameters of field devices together with cut length and sync area length into CR#10000 of special module K100. After this, execute DTO instruction to generate the rotary cut E-CAM curve.

■ Application Example 1: Rotary Cut

Setup:

The example explains the steps to setup rotary cut E-CAM curve. Parameters for setting include: Length of Master = 1000; Length of Slave = 500; Length of synchronized area = 300; Magnification ratio = 100. In addition, set b[14] of high word of P6 = 1 for E-CAM curve setting, indicating only single data real-time modification is applicable. This is to prevent rotary cut E-CAM curves from being modified during program execution. The below diagram explains the generated rotary cut E-CAM curve and the required parameters:

9 Electrical CAM



Parameters	SV
Length of Master	1000
Length of Slave	500
Length of sync area	200
Magnification ratio	1.0
Max magnification ratio	10.0
Acceleration curve	0
CAM curve	0
CAM curve setting	Only single data real-time modification is applicable
Results	0

Control Steps:

Step 1: Add 2 sets of E-CAM charts and set the resolution as 300.

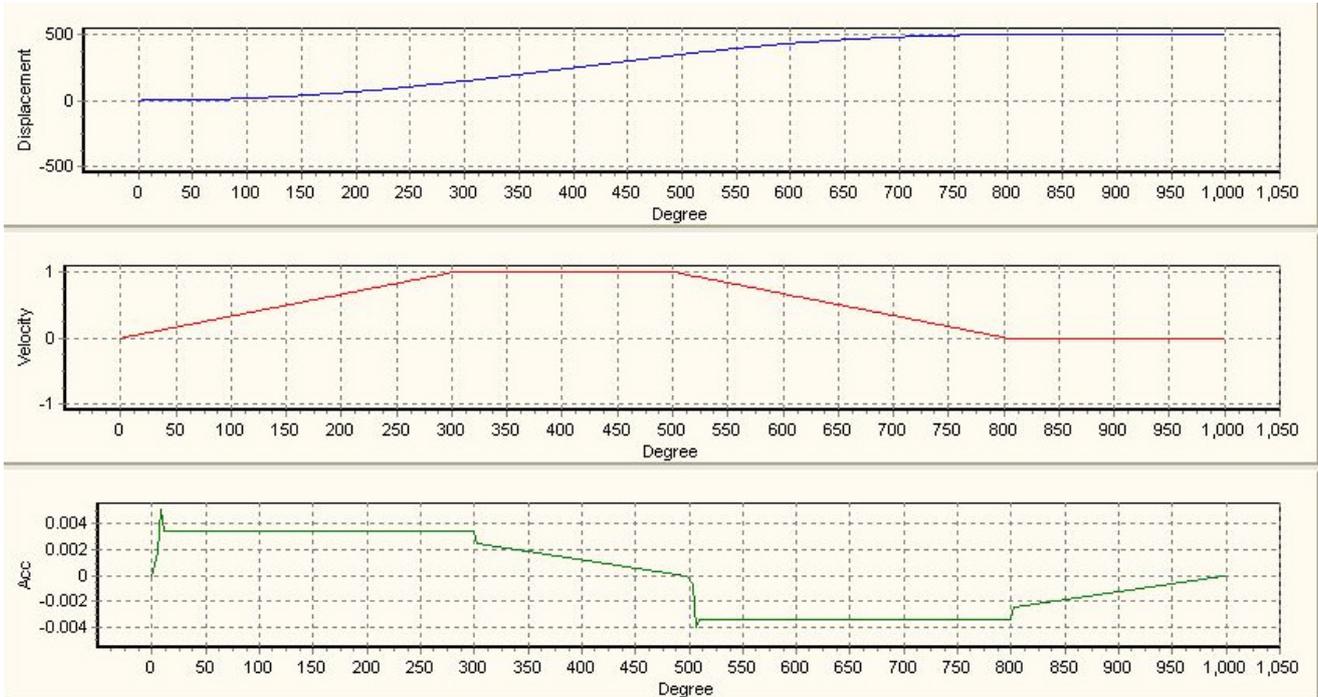
Step 2: Download the below program in DVP-20PM and execute.

Step 3: Set M1 to write the parameters for rotary cut E-CAM into D100~D112 and CR#10000 into D0

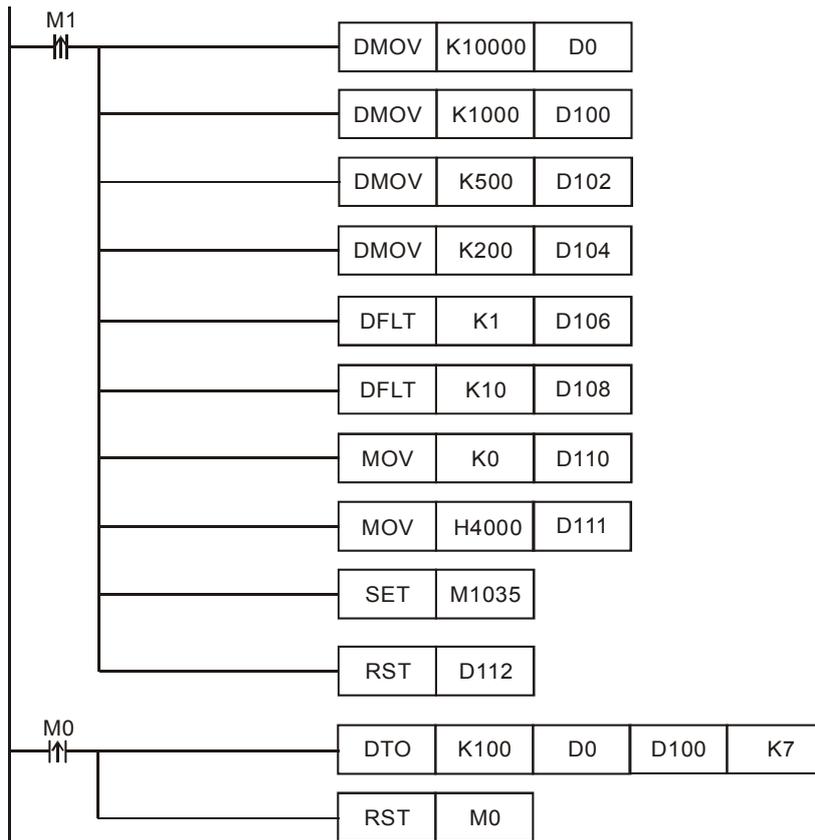
Step 4: Set M0 to write the data in D100~D112 into special register CR#10000 to generate the rotary cut E-CAM curve.

Step 5: Stop the DVP-PM and upload the program.

Step 6: Monitor the uploaded E-CAM Chart-0, which is the generated rotary cut E-CAM curves.



Ladder Diagram:



■ Application Example 2: Flying Saw Application

Setup:

Flying saw operation can be performed by applying 2 rotary cut E-CAM curves. First, we need to create the 1st E-CAM curve for rotary cut. Second, design the 2nd E-CAM curve, in which length of Slave is specified with a negative value of the previous E-CAM curve. In addition, set up the 6th data (D111 in this example) to borrow the previous E-CAM data setting and combine the 2 E-CAM curves, so that a reciprocating flying saw operation can be performed.

To create the above flying saw CAM curve, 2 blank E-CAM data spaces of resolution 300 (Total 600 points) should be created before running the program. The set values of the 2 E-CAM are as below.

Parameters	SV of 1 st rotary cut E-CAM	SV of 2 nd rotary cut E-CAM
Length of Master	1000	1000
Length of Slave	500	-500
Length of sync area	200	200
Magnification ratio	1.0	1.0
Max magnification ratio	10.0	10.0
Acceleration curve	0	0
CAM curve	0	0
CAM curve setting	Only single data real-time modification is applicable	1. Only single data real-time modification is applicable 2. Borrow the previous E-CAM data setting
Results	0	0

Control Steps:

9 Electrical CAM

Step 1: Add 2 sets of E-CAM charts and set the resolution as 600.

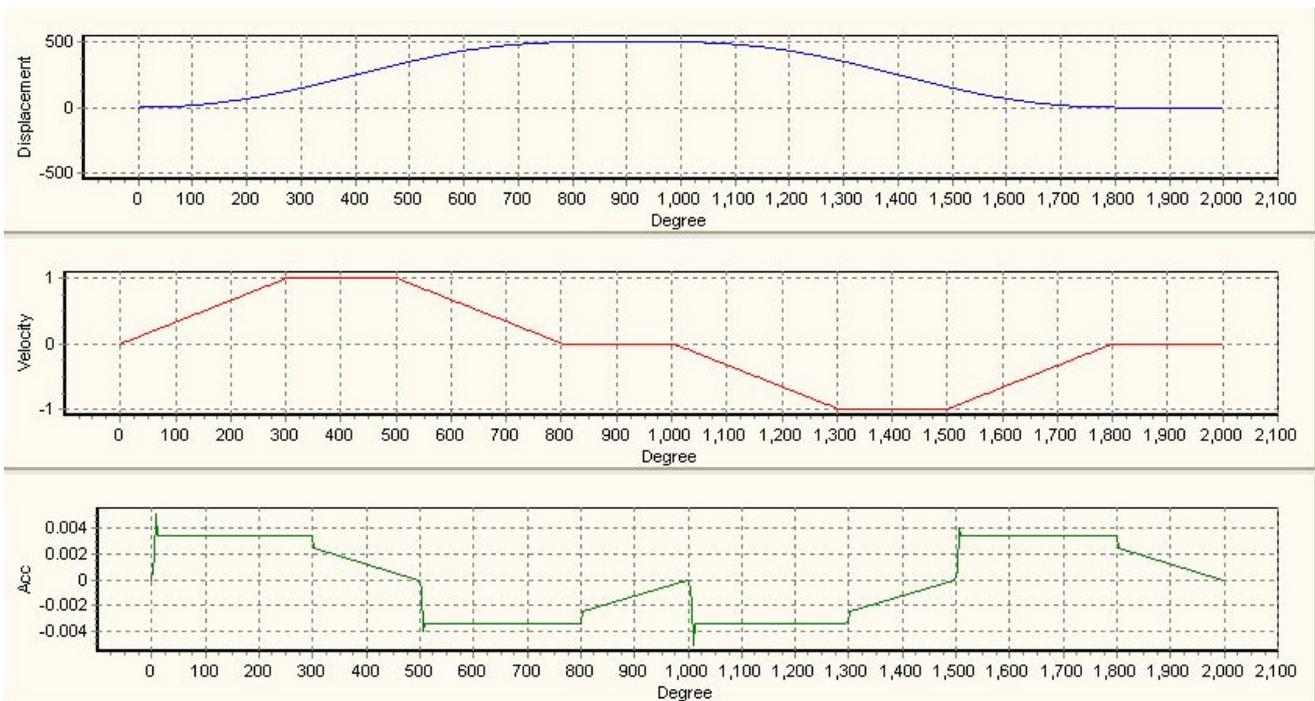
Step 2: Download the below program in DVP-20PM and execute.

Step 3: Set M0 to write the parameters for rotary cut E-CAM into D100~D112 and CR#10000 into D0. In addition, write the data in D100~D112 into special register CR#10000 to generate the 1st rotary cut E-CAM curve

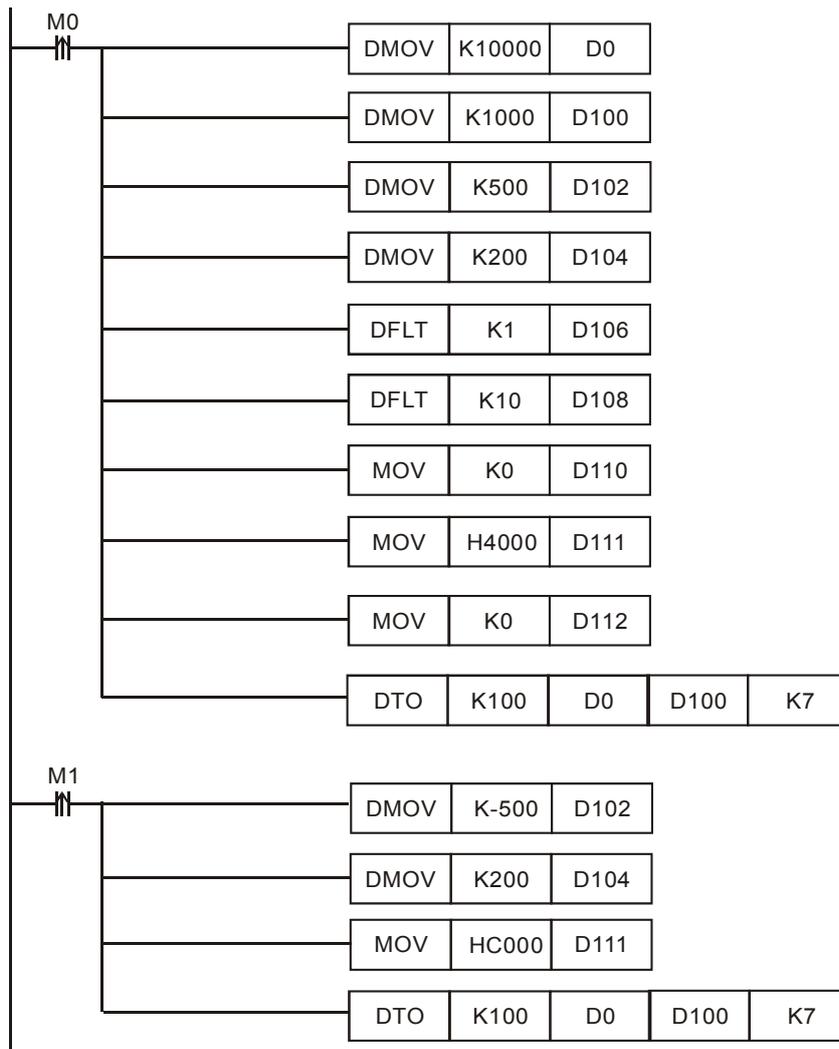
Step 4: Set M1 to change the data in D102, D104 and D111. Write the modified data into special register CR#10000 to generate the 2nd rotary cut E-CAM curve.

Step 5: Stop the DVP-PM and upload the program.

Step 6: Monitor the uploaded E-CAM Chart-0, which is the generated flying saw E-CAM curves.



Ladder Diagram:



■ Application Example 3: Multi-cutter application

Setup:

In multi-cutter application, such as rotary cut with 3 cutters, the pulses per round of Slave can not be evenly distributed to each cutter and the error of Slave will increase as the execution times increase. In this case, the CAM data of each cutter should be designed respectively and the sum of the data length of 3 cutters should equal to total pulses per round.

To meet this requirement, we only need to design the data setting of the first E-CAM once and set up the associated register for borrowing the data setting from the previous E-CAM curve. By this function, the 3 CAM curves can be well connected and the pulses per round of Slave can be fulfilled. The set values of the 3 E-CAM are as below.

Parameters	SV of 1 st rotary cut E-CAM	SV of 2 nd rotary cut E-CAM	SV of 3 rd rotary cut E-CAM
Length of Master	1000	1000	1000
Length of Slave	333	333	334
Length of sync area	200	200	200
Magnification ratio	1.0	1.0	1.0
Max magnification ratio	10.0	10.0	10.0
Acceleration curve	0	0	0

9 Electrical CAM

CAM curve	0	0	0
CAM curve setting	Only single data real-time modification is applicable	Borrow the previous E-CAM data setting	Borrow the previous E-CAM data setting
Results	0	0	0

Control Steps:

Step 1: Add 2 sets of E-CAM charts and set the resolution as 900.

Step 2: Download the below program in DVP-20PM and execute.

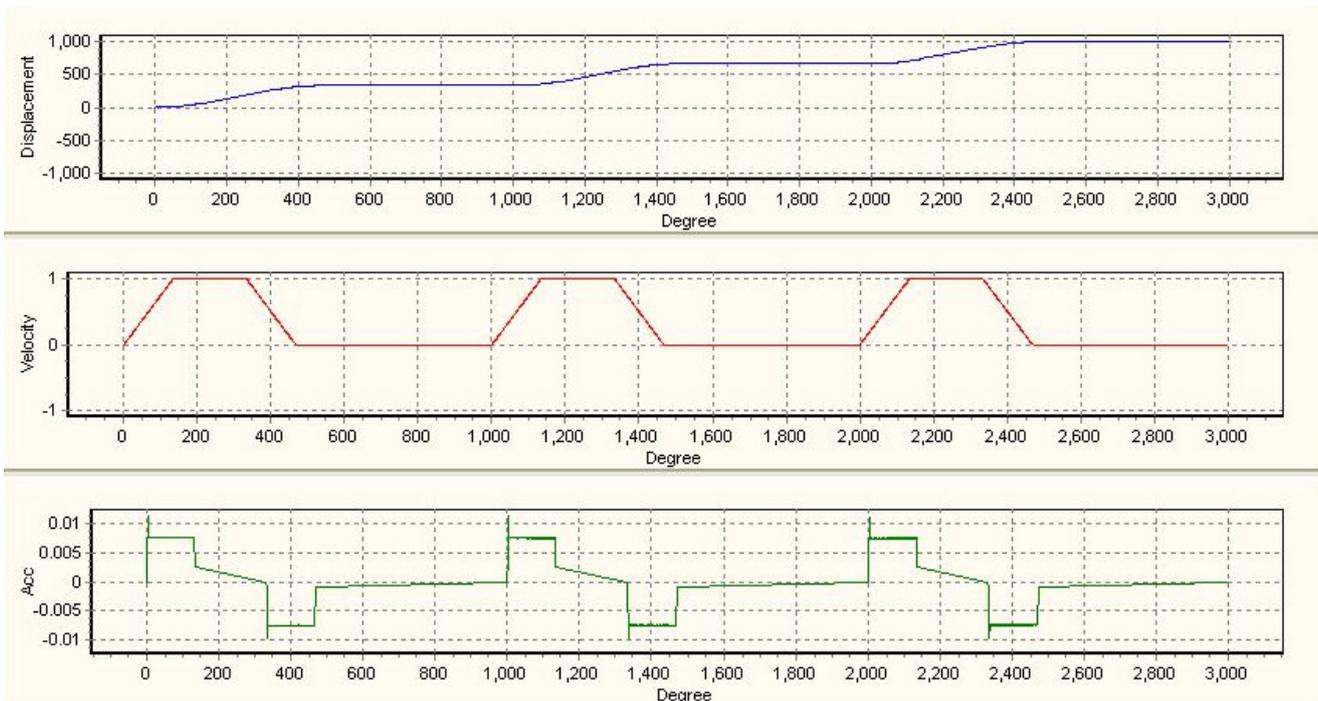
Step 3: Set M100 to write the parameters for rotary cut E-CAM into D100~D112 and CR#10000 into D0. In addition, write the data in D100~D112 into special register CR#10000 to generate the 1st rotary cut E-CAM curve

Step 4: Set M101 to change the data in D111. Write the modified data into special register CR#10000 to generate the 2nd rotary cut E-CAM curve.

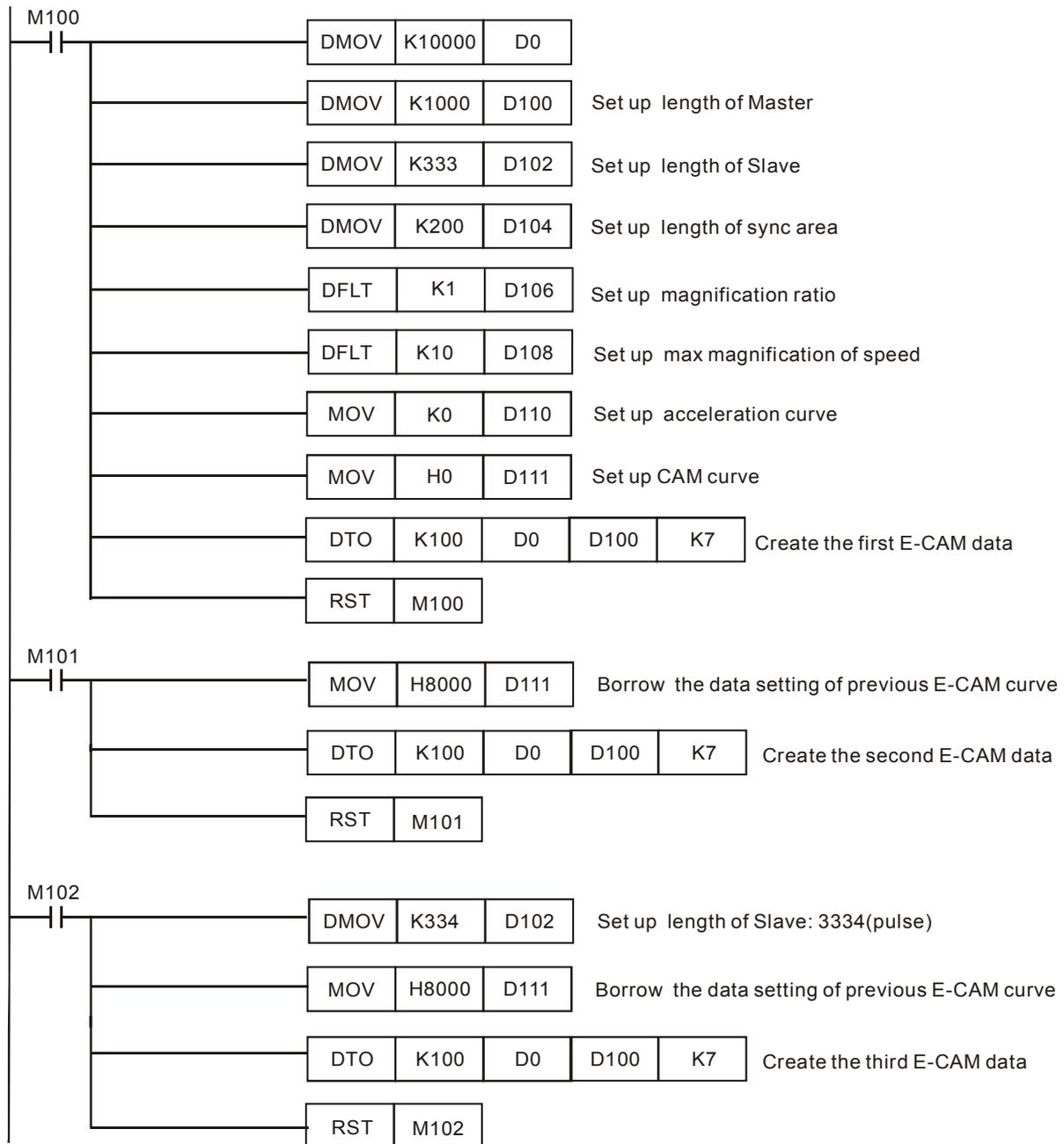
Step 5: Set M102 to change the data in D102 and D111. Write the modified data into special register CR#10000 to generate the 3rd rotary cut E-CAM curve.

Step 6: Stop the DVP-PM and upload the program.

Step 7: Monitor the uploaded E-CAM Chart-0, which is the generated multi-cutter E-CAM curves.



Ladder Diagram:



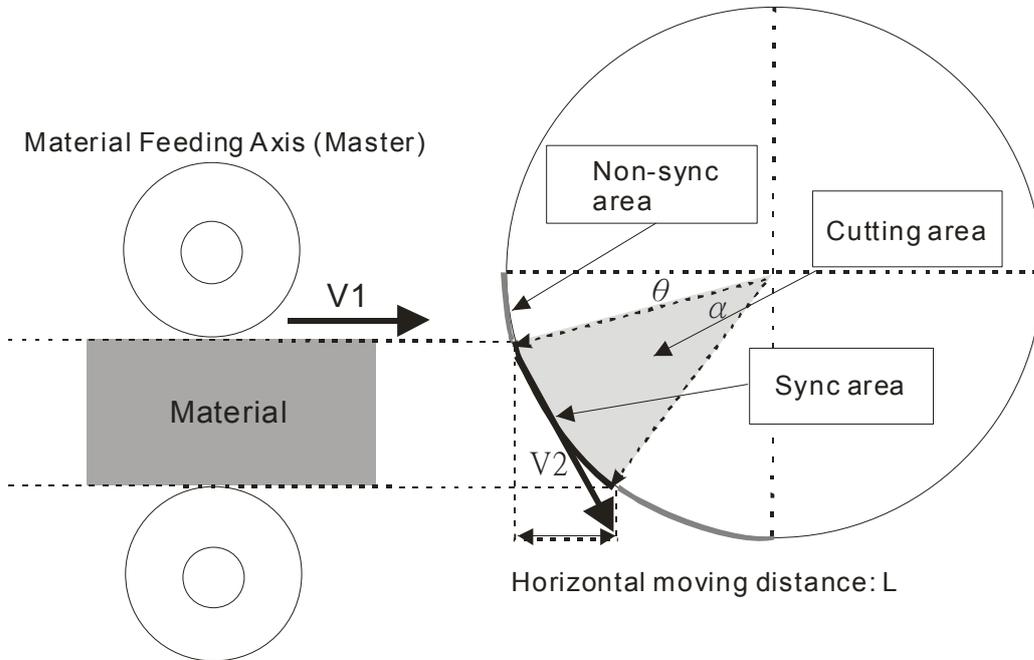
9.4.2.4 Thick Material Cutting

In above rotary cut application, there is only one contact point for the cutter and the material, regardless of the thickness of the material. In this way, only one synchronizing point is required when designing the E-CAM. For thick material cutting, such as soap or steel plate, the velocity relationship between cutter dropping and the material moving should be considered. When the material enters the sync area, the material moving speed will equal to the moving speed of cutter, ensuring that the cutting plane is flat and vertical to the conveyor. The below diagram illustrates the operation angles between the cutter the material. V1 is the material moving speed; V2 is the cutter dropping speed; and the grey curves indicate sync area (cutting area). When the thickness of material is equal, the contact angle between the cutter and the material (angle θ) is a fixed value. In addition, the cutting angle α varies during the cutting process. The velocity

9 Electrical CAM

relationship can be described by the equation below: $\frac{V2}{V1} = \frac{1}{\sin(\alpha + \theta)}$. According to the diagram below,

the cutter rolls through non-sync area and sync area. In sync area, the horizontal moving distance (L) of the cutter should be the same as the moving distance of the material, and the horizontal moving distance can be obtained by the equation: $L = R \cos \theta - R \sin \alpha$. In non-sync area, users can set up the acceleration/deceleration speed according to the cut length of material.



Furthermore, the starting position (angle) of the cutter should also be considered by the number of cutters on the cutting axis. If the total pulses of cutting axis can be equally divided by the cutters, i.e. even number of cutters shares even number of pulses, the pulses required for single cutter can be obtained. Convert the obtained pulses into degrees and the starting position as well as the end position can be obtained, which are the required parameters for setting thick material cutting. On the other hand, if the total pulses of cutting axis can not be equally divided by the cutters, i.e. even number of pulses for odd number of cutters, angle θ and the length of Master before sync area for single cutter should be set up respectively to obtain the starting and end position of each cutter.

Also, care should be taken on setting the contact angle, which is related to the thickness of the material and the vertical position of the cutting device. When setting the contact angle, make sure the contact angle is a little smaller or equal to the actual contact angle otherwise an arc cutting plane will be generated. On the other hand, the sync end angle should also be a little bigger or equal to the actual sync end angle for ensuring that a smooth and flat cutting plane can be obtained.

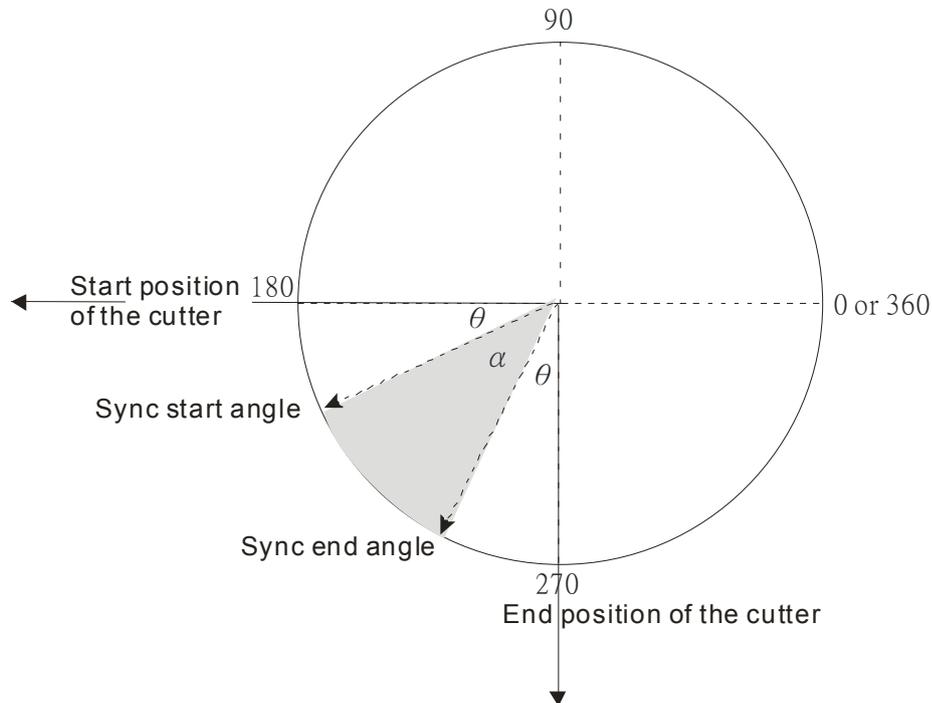
The settings of two cases are explained here below.

- Case 1: Pulses can be equally divided:

For even number of cutters, select "Equally divide non-sync area" function by bit0 of Data 1, and DVP-PM will automatically calculate the required pulses for each cutter. As the below diagram, in actual settings only 2 parameters including "Sync start angle" and Sync end angle" are required for calculating the angle of sync area α and the angle of non-sync area θ .

$\alpha = \text{Sync end angle of Slave} - \text{Sync start angle of Slave}$

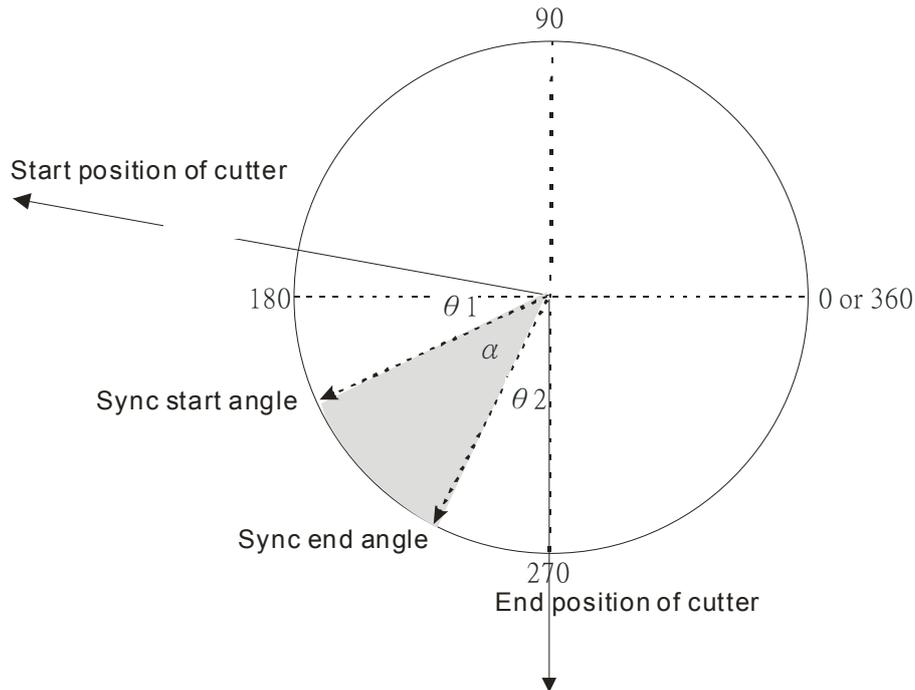
$$\theta = \left(\frac{360}{\text{Pulses per round}} \right) \times \left(\frac{\text{Pulses per round}}{\text{Number of cutters}} - \alpha \cdot \frac{\text{Pulses per round}}{360} \right) \div 2$$



The actual adjustment steps are: 1. select the sync start and sync end angle; 2. align the cutter with the start position of the cutter; 3. Execute E-CAM operation; 4. Inspect the cutting plane on material. If the cutting plane is bended, conduct fine tuning on the sync start/end angle until the cutting plane is flat and smooth.

- Case 2: Pulses can NOT be equally divided:

If the total pulses of Slave can not be equally divided by the cutters, angle θ_1 and the length of Master before sync area should be specified respectively in E-CAM setting, so that the starting and end position of each cutter can be obtained. As the diagram below, parameters including angle θ_1 , length of Master before sync area, sync start angle and sync end angle should be specified respectively. In addition, the pulses required for each cutter should also be specified in E-CAM setting. When the above parameters are ready, run the thick material cutting E-CAM and conduct fine-tuning on the parameters of each cutter until the best results are obtained.



■ Application Example:

Setup:

The example is the application of using DTO instruction for creating thick material cutting E-CAM. The required parameters are listed in the table below. We assume that the diameter of material feeding axis is 170(mm), diameter of cutting axis is 240(mm), number of cutters: 4, pulses per round: 200,000, sync start angle of Slave: $\theta_1 = 210$ degree, and sync end angle of Slave: $\theta_2 = 250$ degree. The cutting length can be calculated according to the parameters above: $170 \times 314 \div 4 = 13345$. Parameters for thick-material cutting are listed as below.

Data	Set value
E-CAM format	1
Pulses per round of Master	200000
Length per round of Master	53380
Pulses per round of Slave	200000
Length per round of Slave	75360
Sync start angle of Slave:180~360	210
Sync end angle of Slave:180~360	250
Length of cutting process on Master	13345
Number of cutters on Slave	4
Length of Master before sync area	-
Angle of Slave between start position and sync start: angle θ_1 (degree)	-
Number of data points in sync area	-

Control Steps:

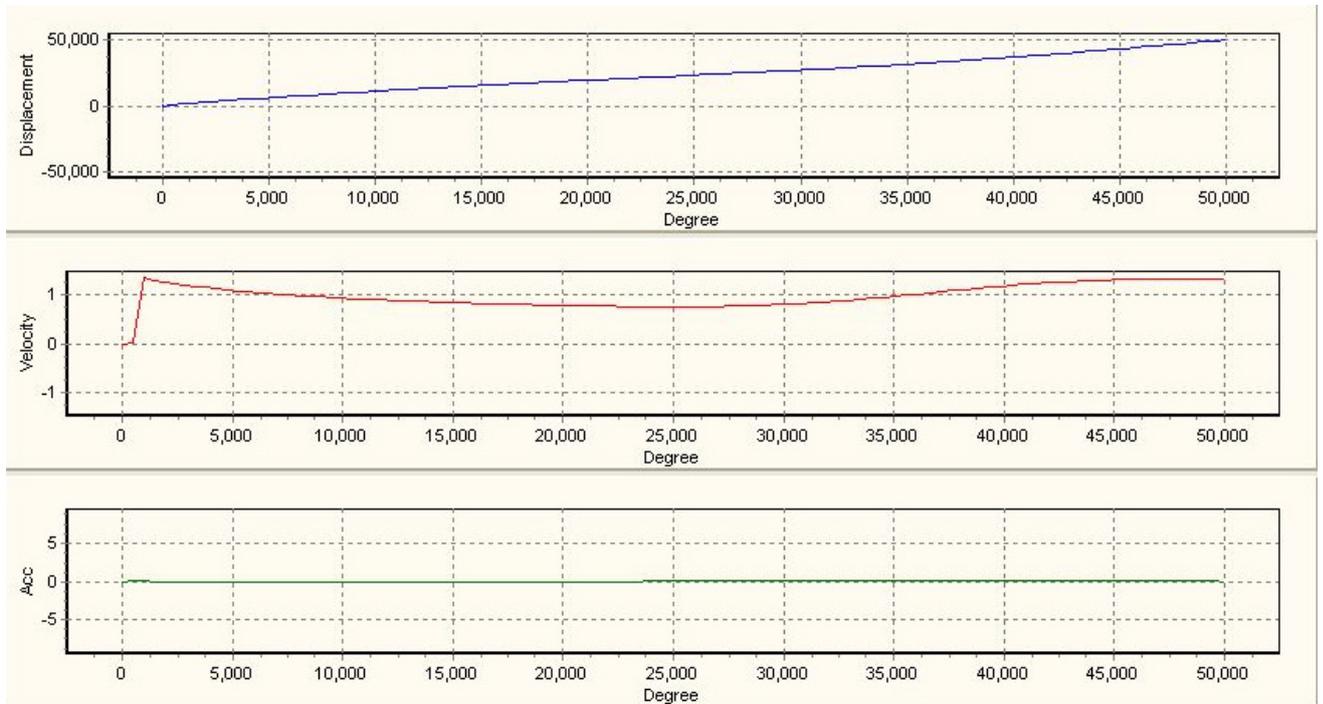
Step 1: Add 2 sets of E-CAM charts and set the resolution as 407.

Step 2: Download the below program in DVP-20PM and execute.

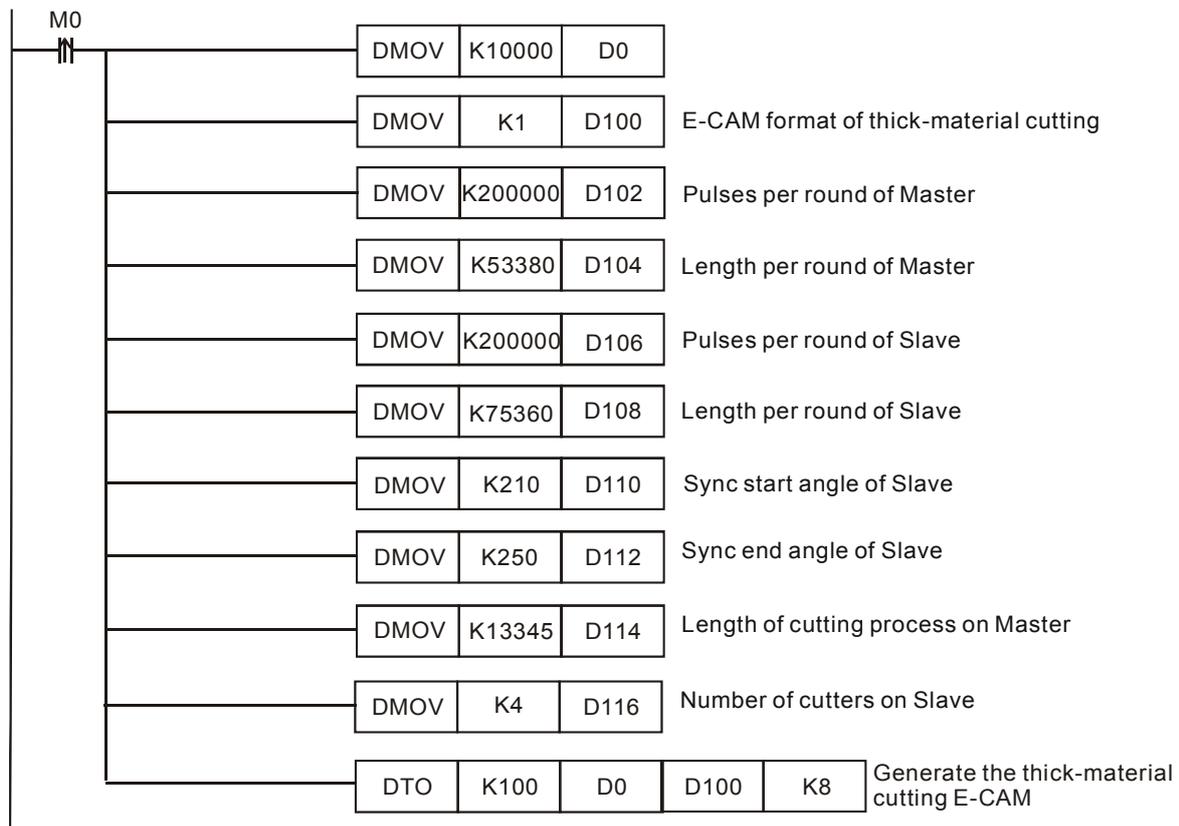
Step 3: Set M0 to write the parameters for thick-material cutting E-CAM into D100~D116 and CR#10000 into D0. In addition, write the data in D100~D116 into special register CR#10000 to generate the thick-material cutting E-CAM curve

Step 4: Stop the DVP-PM and upload the program.

Step 5: Monitor the uploaded E-CAM Chart-0, which is the generated thick-material cutting E-CAM curves.

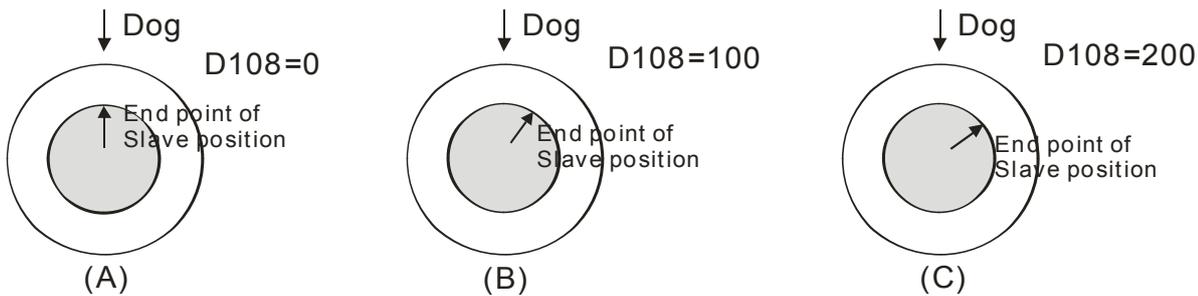


Ladder Diagram:



9.4.2.5 Offset Compensation to Slave Error

In field application, errors occur between the actual output position on device and the target output position on E-CAM curve. In addition, the error could be larger as the increase of execution times. Aiming at this problem we need to apply offset compensation according to the error value between DOG signal and the end point of Slave position (D108). First, backup the initial value of D108 (Diagram A). Second, compare the value in backup register (D128) with D108, which has errors to the initial value (Diagram B, C). Third, take the difference between the 2 registers as the reference value for offset compensation.



Application Methods:

1. Read initial Slave error. The initial error is the value obtained by subtracting the captured value at DOG with the captured value at E-CAM completion.
2. When the Slave error increases in the next cycle, subtract the increased Slave error with the initial Slave error, and we can have the real Slave error.
3. Subtract Length per round of Slave, which is one of the E-CAM parameter, with the real Slave error and we can have the real error between the results of E-CAM output and the device output. On this basis, we conduct offset compensation for the error.
4. When the real error is between “ \pm half of Slave length”, take the real error as the offset value directly.
5. When the real error exceeds “ \pm half of Slave length”, follow the below rule:

$$\left\{ \begin{array}{l} \text{Real error} > \frac{1}{2} \text{ length of Slave} \Rightarrow \text{Offset} = (\text{Real error} - \text{length of Slave}) \times \text{Offset ratio} \\ \text{Real error} < -\frac{1}{2} \text{ length of Slave} \Rightarrow \text{Offset} = (\text{Real error} + \text{length of Slave}) \times \text{Offset ratio} \end{array} \right.$$

6. In the next cycle, input the modified length of Slave (with offset) by DTO instruction and the offset compensation is completed.

Application Example

Setup:

The example is used for judging if real error between DOG switch and E-CAM completion exists. When error occurs, calculate offset value for compensation on the length of Slave. In this case, D302 refers to the modified length of Slave.

Control Steps:

Step 1: Run the program and read the E-CAM status by DFROM instruction. Read the data in CR#10001 of special module K100 to D100~D108.

Step 2: Set M3 to write the length of Slave into D126 and the initial Slave error into D128. Calculate the real error, i.e. difference between the increased error and the initial error, and take the half of Slave length as the reference.

Step 3: When the real error is between “± half of Slave length”, take the real error as the offset value directly.

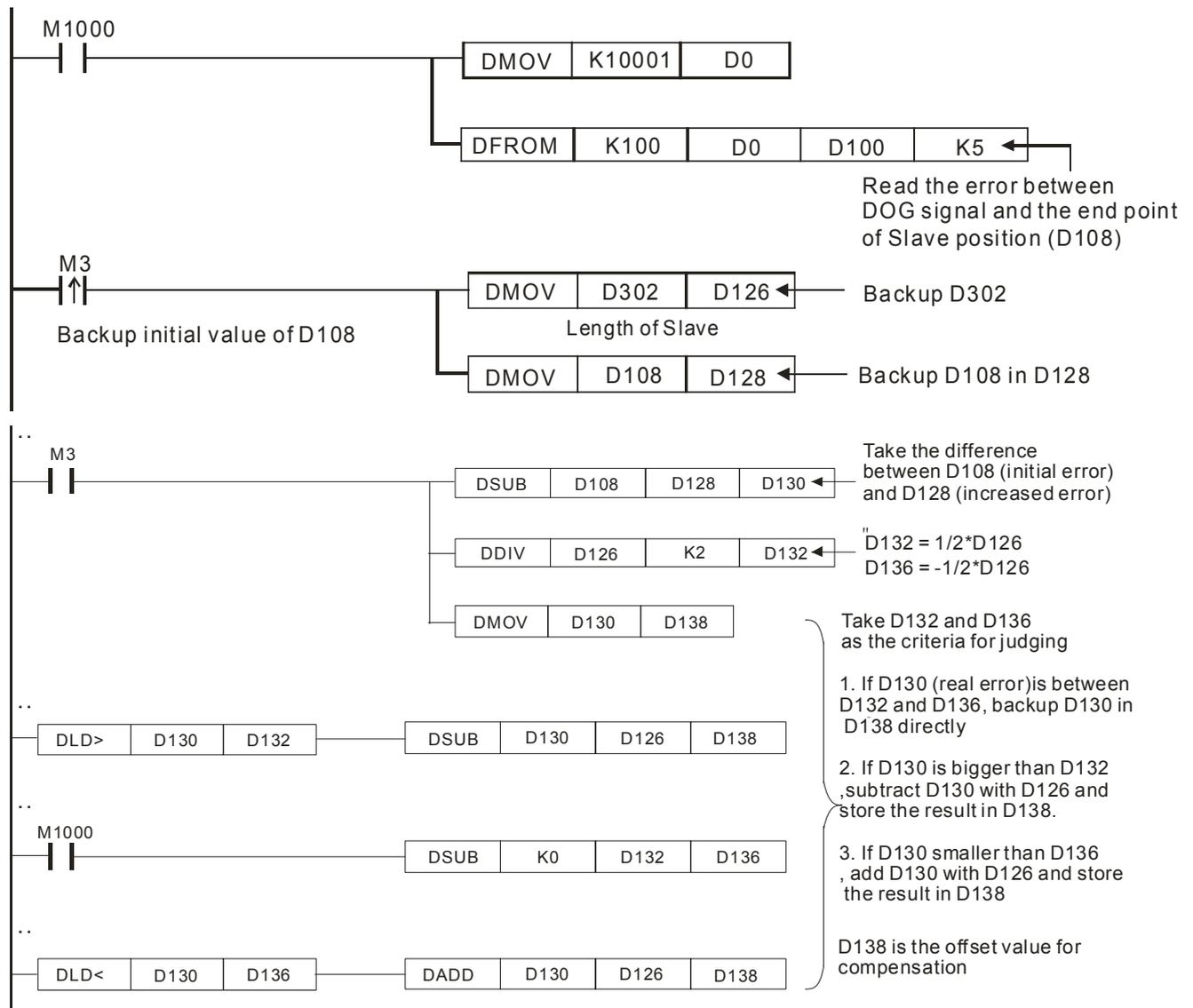
Step 4: When the real error exceeds “± half of Slave length”, follow the below rule:

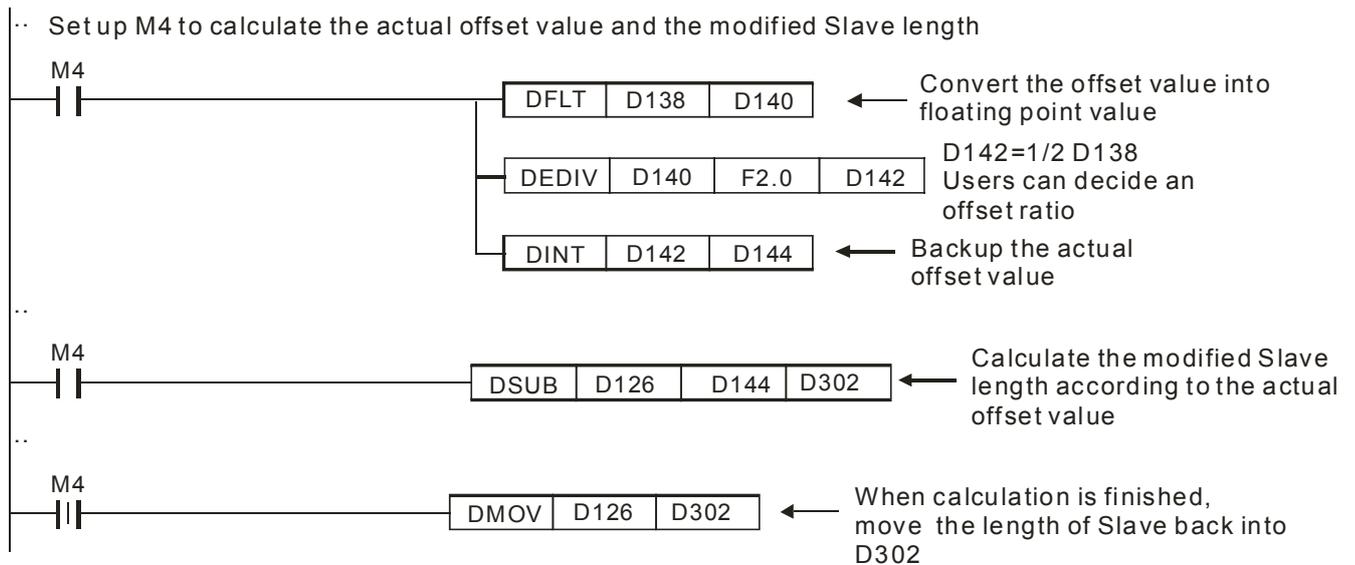
$$\begin{cases} \text{Real error} > \frac{1}{2} \text{ length of Slave} \Rightarrow \text{Offset} = (\text{Real error} - \text{length of Slave}) \times \text{Offset ratio} \\ \text{Real error} < -\frac{1}{2} \text{ length of Slave} \Rightarrow \text{Offset} = (\text{Real error} + \text{length of Slave}) \times \text{Offset ratio} \end{cases}$$

Step 5: Set M4 to calculate the modified length of Slave with offset compensation.

Step 6: When M4 is reset, fill the original length of Slave back to D302.

Ladder Diagram:





Points to note:

1. Users can modify the offset ratio ($0.5 \times D138$ in this example) according to actual needs
2. D132 has to be applied in DTO instruction for creating rotary cut E-CAM Data with offset compensation

9.4.2.6 Flying Saw Application – Check cutter position

In flying saw application, it is important to check if the cutter is back in designated position when moving back. If the cutter is not at designated position, it should remain current speed until limit switch is met, otherwise the mechanical system might be damaged during the moving back process of cutter. M1757 can be applied to remain current speed of Slave and continue operation until the limit. Users can compare the designated position with Master position in current cycle to decide if M1757 needs to be enabled. Master position in current cycle can be obtained by subtracting “Master position captured at the end of cycle” from accumulated Master position (DD1862). Master position captured at end of cycle can be read out by using DFROM instruction (D0 = K10005, Capture mode).

■ Application Example

Set up:

Check if the cutter position is correct and automatically determine the execution of M1757.

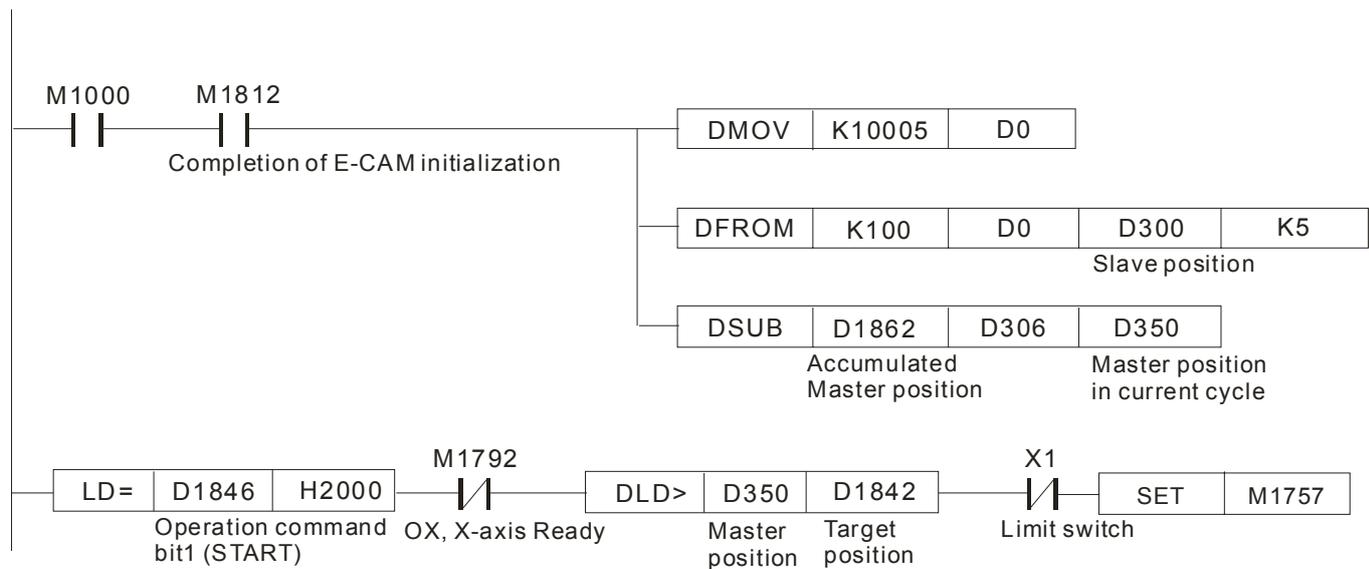
Control Steps:

Step 1: Run the program and read the E-CAM status in capture mode by DFROM instruction. Read the data in CR#10005 (Capture mode) of special module K100 to D300~D308 at the end of cycle.

Step 2: Subtract the captured Master position (D306) from the number of accumulated MPG input pulses (D1862), and we can obtain the actual Master position in current cycle (D350).

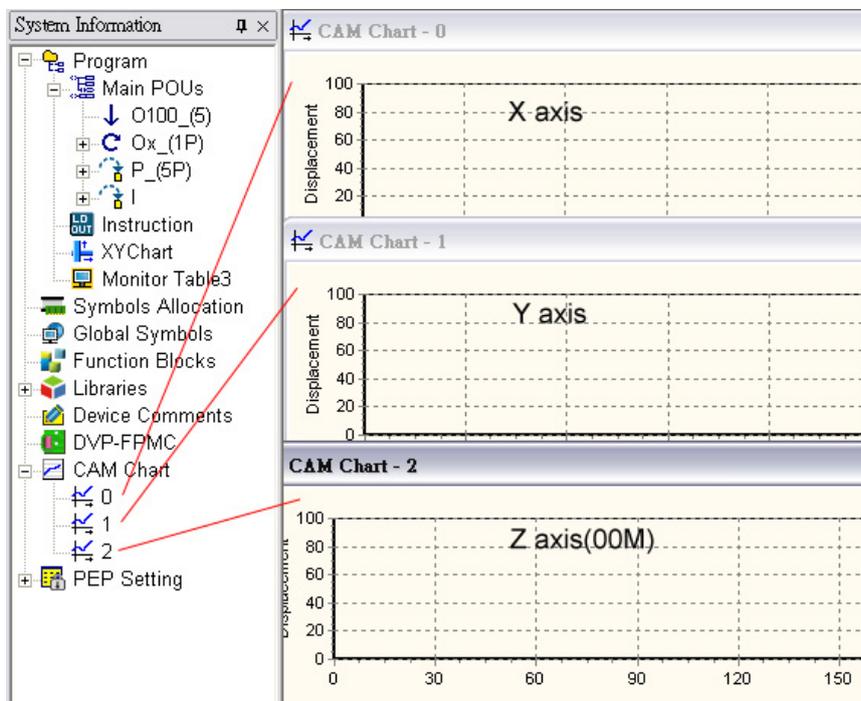
Step 3: Compare the actual Master position (D350) with the target position (D1842). If the actual Master position is bigger than the target position and the limit switch has not been triggered, execute M1757 then stop the CAM operation.

Ladder Diagram:



9.5 Multi-axis E-CAM

Single-axis E-CAM can be executed in cyclic operation and acyclic operation. Unlike single-axis E-CAM, multi-axis E-CAM can only be executed in cyclic operation. For multi-axis E-CAM, only one set of E-CAM Charts can be applied, i.e. CAM Chart 0 ~ CAM Chart 2 correspond to X, Y, Z axis.



Same as single-axis cyclic E-CAM, multi-axis E-CAM is enabled by setting b13 of D1846 as ON. In addition to setting D1846, work modes of each axis should also be set up and enabled, and C200 should be activated for counting pulses of Master.

1. D1846

Set b13 of D1846 as ON (D1846 =H'2000) to enable multi-axis E-CAM.

9 Electrical CAM

2. Set up work modes (X-D1847 Y-D1927 Z-D2007)

For multi-axis E-CAM, b12 and b11 of D1847 should be set as 01 (bit11=1). When Y, Z axis are applied, bit11 of D1927 and D2007 should also be set as 1.

3. C200/C204 counter

Enable C200/C204 after setting up counting mode of C200/C204. Multi-axis E-CAM applies C200/C204 as the input signal of Master. Counting mode of C200/C204 should match the pulse input pattern of Master.

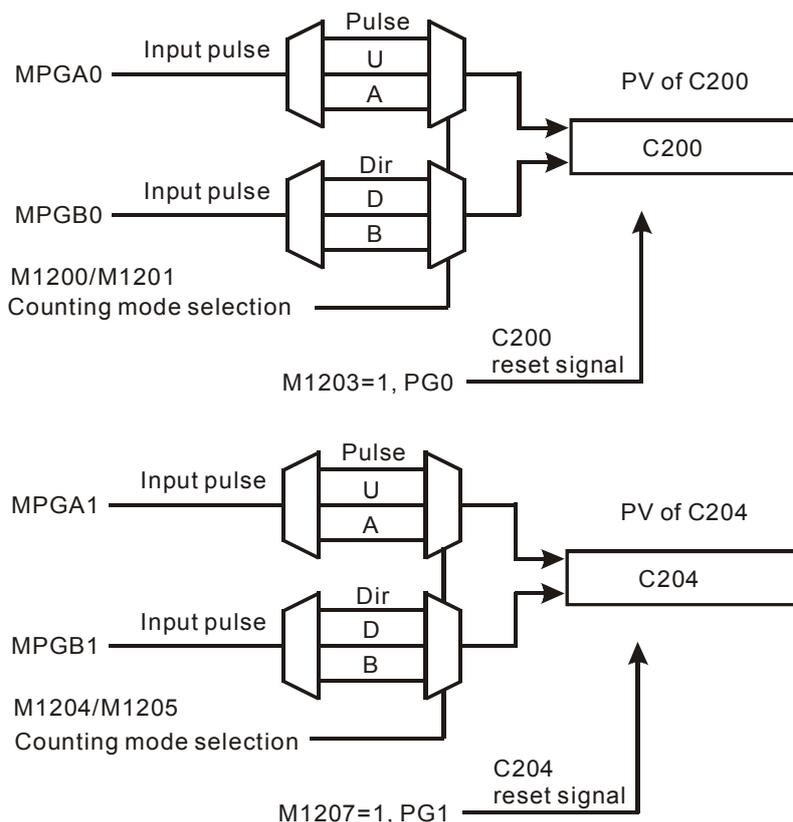
C200/C204 counting mode setting:

Device	C200 Counting Mode	Device	C204 Counting Mode
M1200=0, M1201=0	U/D*	M1204=0, M1205=0	U/D
M1200=1, M1201=0	P/D*	M1204=1, M1205=0	P/D
M1200=0, M1201=1	A/B*(Single frequency)	M1204=0, M1205=1	A/B(Single frequency)
M1200=1, M1201=1	4A/B(Quadruple frequency)	M1204=1, M1205=1	4A/B(Quadruple frequency)
Deivce	C200 Counting Mode	Device	C204 Counting Mode
M1203=1	PG0	M1207=1	PG1

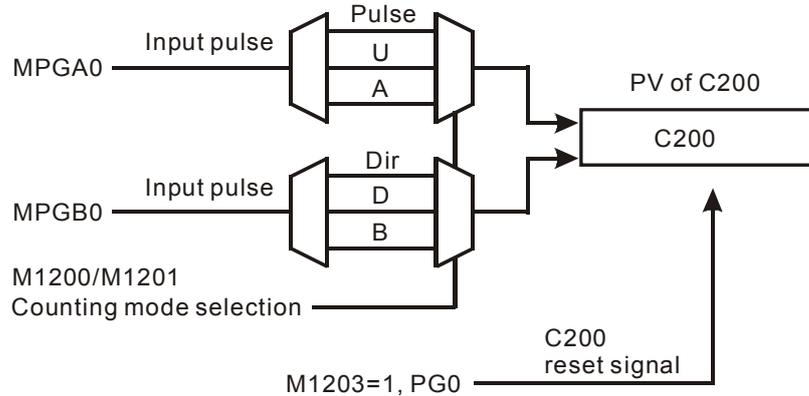
Note:

- U/D: Counting up / Counting down, P/D: Pulse / Direction, A/B: A-phase / B-phase
- M1908: Input signal control of C200 / C204

When M1908 = OFF, the input signal of C200 is controlled by MPG A0/B0 and reset signal is controlled by PG0. In addition, Input signal of C204 is controlled by MPG A1/B1 and reset signal is controlled by PG1.



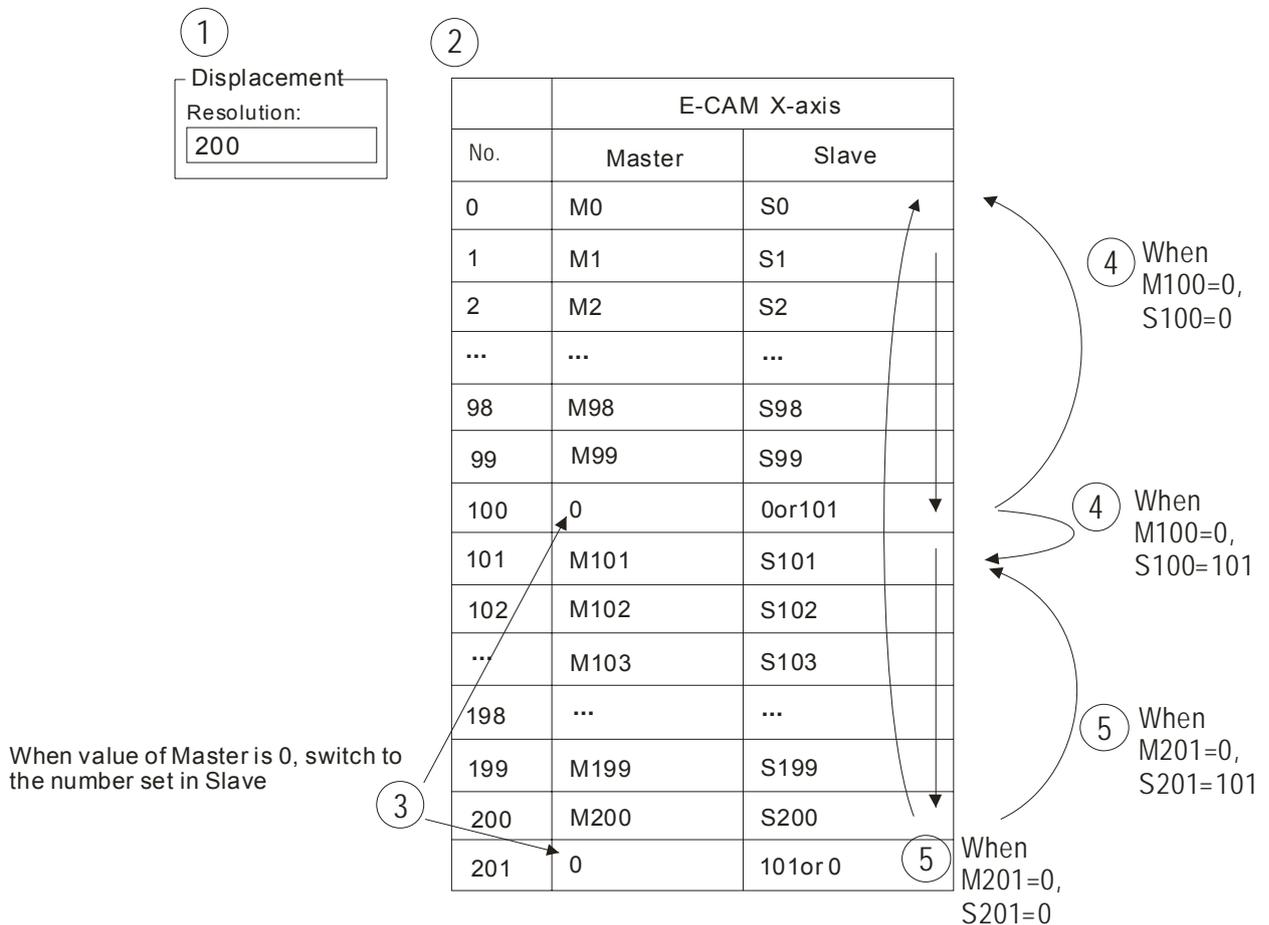
When M1908 = ON, the 3 axes X/Y/Z share the same high speed counter C200. The input signal of C200 is controlled by MPG A0/B0. C200 reset signal is controlled by PG0.



There are 2 execution patterns of multi-axis E-CAM:

1. Point-to-point Switching Pattern

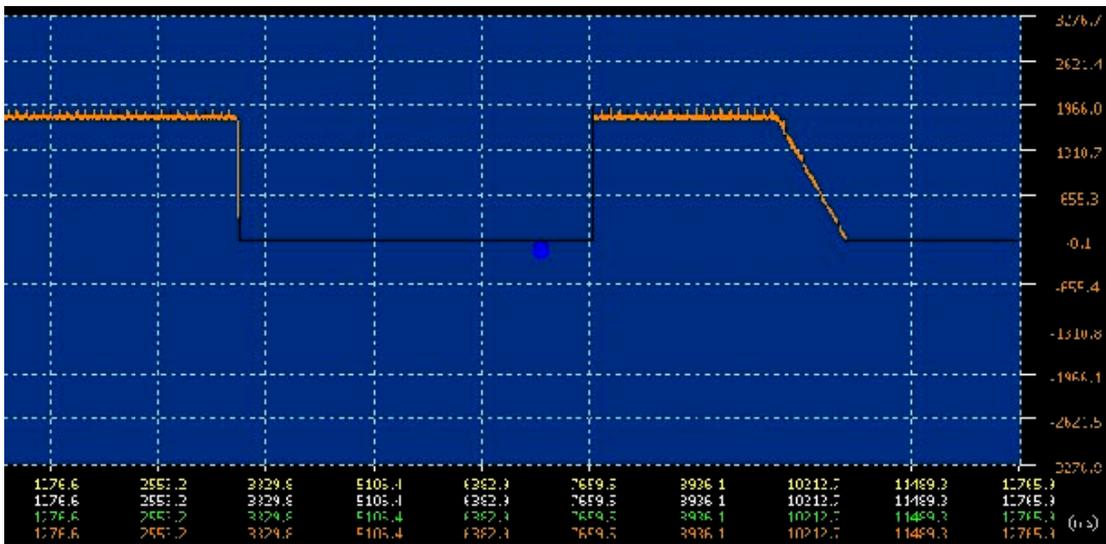
In multi-axis E-CAM, we can separate an M-points E-CAM Data into several sections for realizing real-time CAM data switching between sections. Only one section will be executed for one time and users can add new data into the next section. For switching to next section, insert an additional point of (0, n1) between the sections. When Master position is detected as 0, set value n1 of Slave position will be the target number for poin-to-point switching. When single section is completed, M1813/M1893/M2053 will be ON and has to be clear by user for Indicating next completion. In the below diagram we use an N-point E-CAM Data and separate the beginning 202 points into 2 sections.



9 Electrical CAM

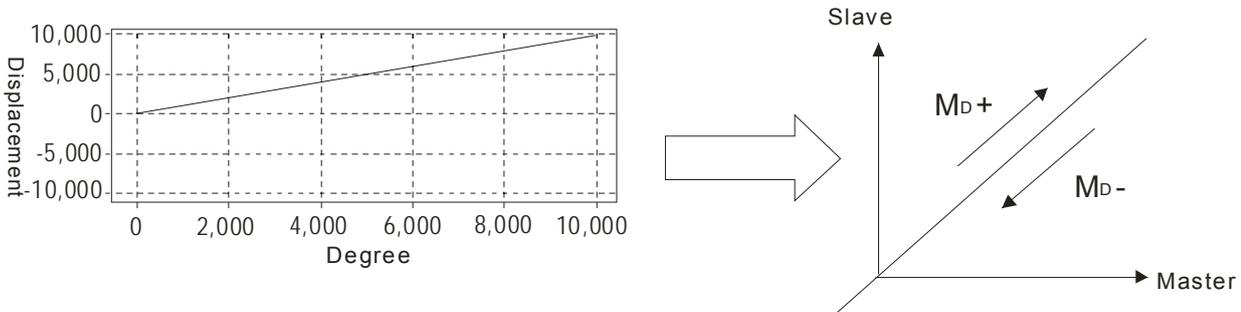
Set value of Slave in No.100 and No.201 indicates the target number for next execution. Therefore, before the switching point is reached users can insert or modify the data in the next section. However, care should be taken on setting the starting value of the second section. For second section, if the starting value of Master (M101) is set as 0, Slave will continue the execution based on the ending value of previous section, i.e. the starting value will be Master position: M99+M102, and Slave position: S99+S101. On the other hand, if the starting value of Master is not "0", the starting value of Slave will be based on "0", i.e. Master position: M99+M102 and Slave position: 0+S101.

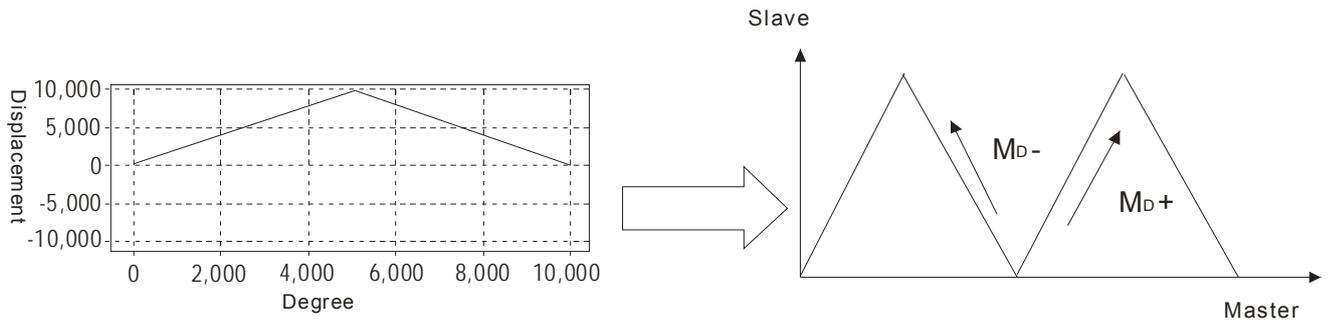
In addition, for point-to-point switching pattern, if E-CAM max frequency (DD1840) is set as 0Hz and the value of deceleration time T_{DEC} (D1837) is specified, E-CAM will decelerate to stop. The below diagram indicates the difference between normal stop operation and stop operation with deceleration.



2. Forward / reverse Running Pattern

To enable forward / reverse running pattern, bit 12, 11 of D1847/D1927/D1927 (work mode setting) should be set ON (K3). In forward / reverse running pattern of multi-axis E-CAM, the execution of Slave will be in the same direction as Master. In reverse running operation, the execution will start again from the end of the data when Master position 0 is reached. Therefore, please note that forward/reverse running pattern can not be used with point-to-point switching pattern otherwise error will occur. The below diagrams illustrate the execution of 2 E-CAM Charts.





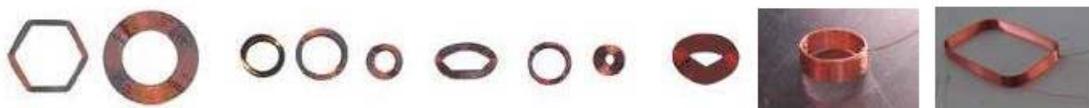
In forward / reverse running pattern, master position 0 other than the start point should not exist.

0	0
100	200
200	400
300	600
400	800
500	1000
600	1200
700	1400
800	1600
900	1800
1000	2000
1100	2200
1200	2400
1300	2600
1400	2800
1500	3000
1600	3200
1700	3400
1800	3600
1900	3800
2000	4000
0	0
0	0
0	0
0	0
0	0

9.6 Field Applications of E-CAM

9.6.1 E-CAM Application on Winding Machine

In this application example we apply DVP-PM for controlling the auto bobbinless winding machine, which can be used on winding air-cored coils of different specifications such as drive coils, speaker coils, antenna coils and other common bobbinless coils. The winding system with DVP-PM has better reliability and is highly automated with high working speed and efficiency as well. Therefore, it can achieve high quality results of batch production of coils. Here are some of the coils:



Normal winding machines apply small PLCs with built-in pulse function. Though high speed input points, the PLC receives pulses from the encoder of winding shaft, and roughly synchronizes the speed between the winding

9 Electrical CAM

shaft and the coil shaft. Due to the possible delay of CPU operation, this kind of winding process has lower accuracy on speed synchronization. Therefore, the coil winding quality could be significantly lowered because the uneven winding results. For example, generally PLC executes the reverse winding operation through interruption when the set up position from the winding shaft is reached. In this case, error will occur due to the delay of CPU operation. In low speed situation, the basic demanded quality could be fulfilled. However, in high speed and multi-layered winding situation, the surface of coil will become uneven and the winding quality will drop down.

DVP-PM PLC is designed specifically for motion control applications. DVP-PM applies duo CPU structure with one CPU exclusively processing motion control algorithm. Therefore, motion control operations such as trajectory drawing, linear/arc interpolation can be well performed with common logical control. In this high speed winding application we apply the E-CAM function to solve the switching delay problem of reverse winding situation explained above.

9.6.1.1 Explanations on High Speed Winding Devices

There are 9 parts of devices in a high speed winding system as below:



(1) Machine stand

The machine stand consists of steel angle frames and stainless platform with castors at the bottom. When the machine stand is located in proper position, users can lower the bottom stands to fix the

machine stand.

(2) Tensioning device

Tensioning device is installed in wire-feeding component to provide proper tension for coil winding. Tension can be modified by the knobs on the device. Tension will be automatically adjusted after the modification is completed.

(3) Winding device

Winding device mainly consists of Delta ASDA-B type servo motor (200W), timing belt and winding flyer and is the winding shaft (Master) of E-CAM operation. Copper wires are wound on the coil die through the winding flyer. Winding device is the major part of windig operation.

(4) Coil device

Coil device mainly consists of Delta ASDA-B type servo motor (100W), precise ball screw, guide rail and pneumatic sliding forks and is the coil shaft (Slave) of E-CAM operation. Coil device follows the winding device and reciprocates to perform the coil spacing action. Coil device is also the major part of winding operation.

(5) Work stage

Work stage mainly consists of index stepping motor, rotation stage, wire fork and coil die. The work stage supports multi-task processing, executing die preheating, wire cutting heating, mold releasing while coil winding is in process.

(6) Cutting device

Cutting device is pneumatic device used in cutting off the wires on the two ends of the completed coil.

(7) Die releasing device

Die releasing device consists of index stepping motor and pneumatic die releasing device, taking the completed coil off the coil die.

(8) Hot air system

Hot air system has 2 temperature adjustable 220V heat guns. Users can preheat the coil die before winding and heat up the completed coil to release the coil from the coil die.

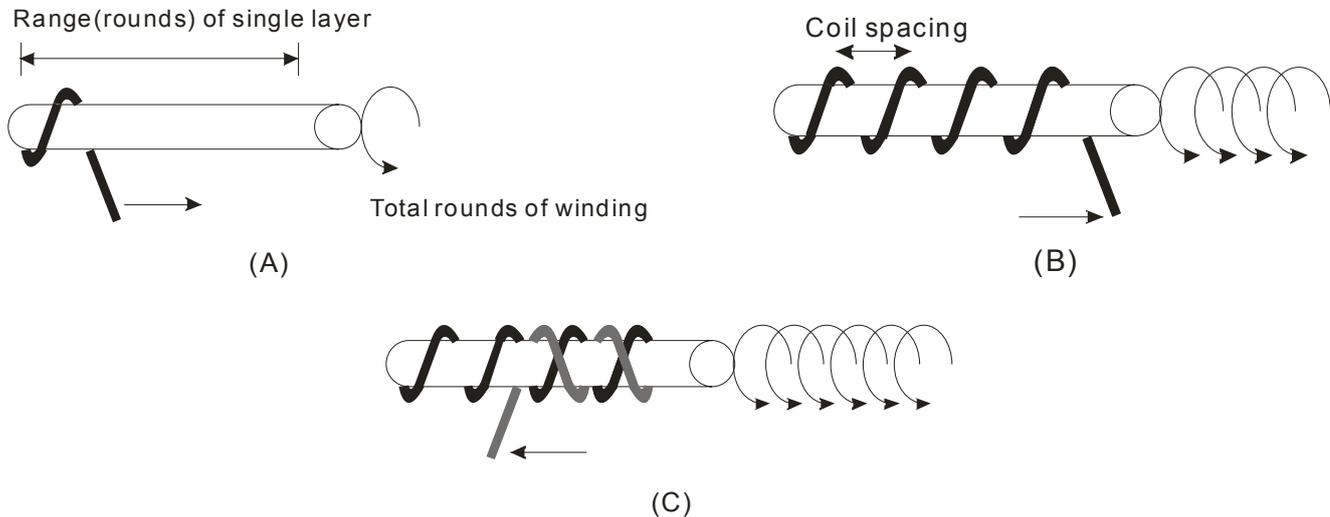
(9) Electric control system

Electric control system has an electric control box and usually a touch panel. In electric control box we apply DVP-20PM00D as the core of motion control, the touch panel as the human-machine interface and the servo motor as working device. By this combination, we can perform precise motion control between winding shaft and the coil shaft, so that the accuracy of coil winding can be obtained.

High speed winding on master shaft and slave shaft can be performed by executing single speed positioning on winding device(3), together with E-CAM operation on coil device(4). Other operations are similar to general sequential ladder control and can be executed by general functions of PLC.

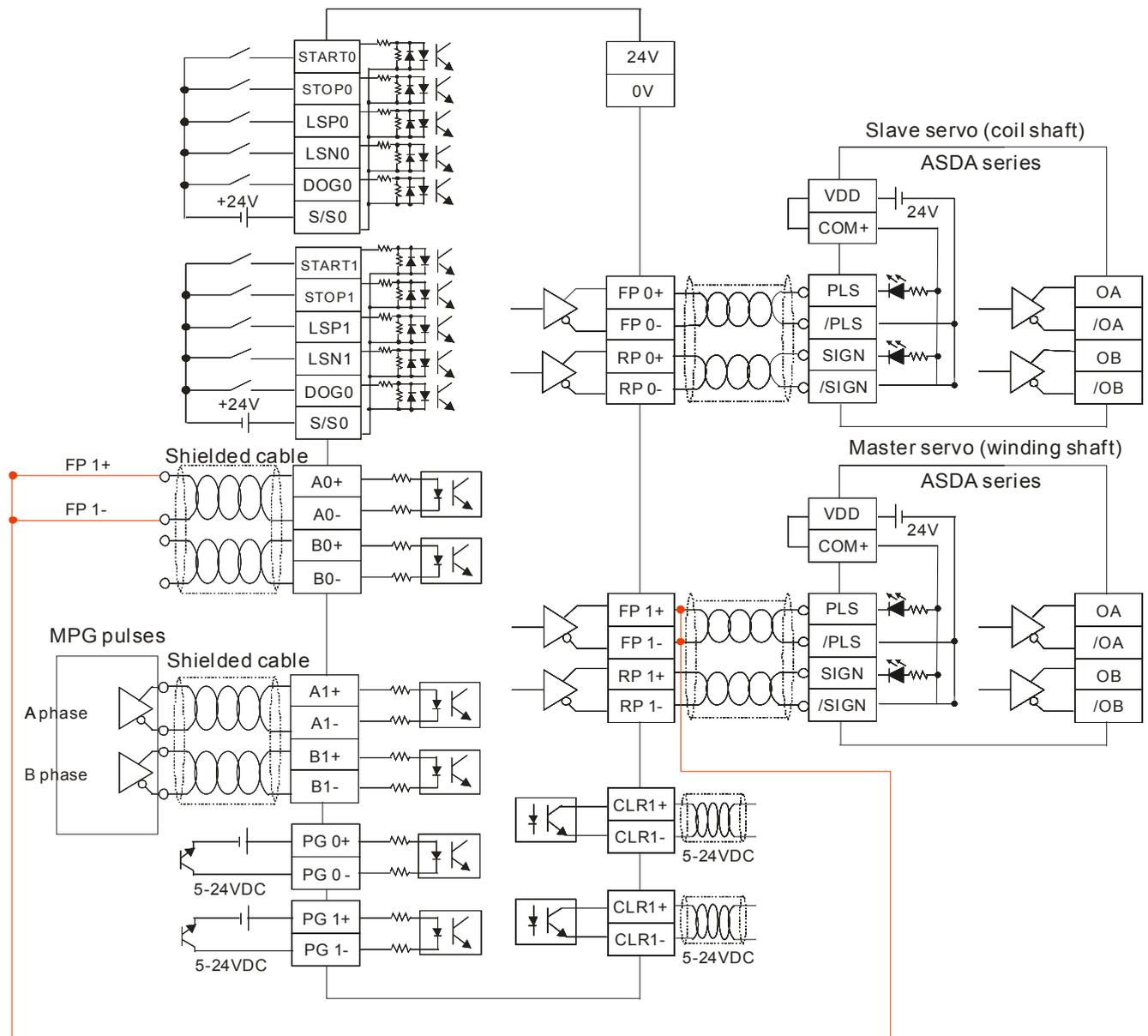
9 Electrical CAM

The movement of winding shafts and coil shafts are explained as below. The coil shaft (Slave) follows the winding shaft (Master) and reciprocates within the range of single layer in certain proportion with winding shaft. At the beginning, Slave starts at the left end and moves one coil space as diagram (A). When Slave moves to the right end of the single layer range as indicated in diagram (B), the moving direction of Slave reverses as diagram (C). When Slave reaches the left end of single layer range, the moving direction of Slave reverses again.



From the above winding movements, there are 3 major input parameters: 1. rounds per layer, 2. total rounds of winding, and 3. coil spacing (The value of coil spacing is calculated by adding coil diameter with the space between each round). In addition, Mechanical parameter (mm/pulses) is required for obtaining the Master/Slave (winding shaft/coil shaft) proportion. Mechanical parameter consists of mechanism parameter and servo parameter. Mechanism parameter is the the moving distance per round (mm/revolution) and servo parameter is the pulses per round (pulses/revolution) obtained by multiplying the electronic gear ratio. Mechanism parameter (mm/revolution) divided by servo parameter (pulses/revolution) equals the mechanical parameter (mm/pulses). After the parameters are ready, apply cyclic E-CAM on X axis (Slave) because the coil shaft reciprocates cycle by cycle, and single speed positioning on Y axis (Master) because the winding shaft moves on in the sme direction.

9.6.1.2 Hardware Wiring Diagram

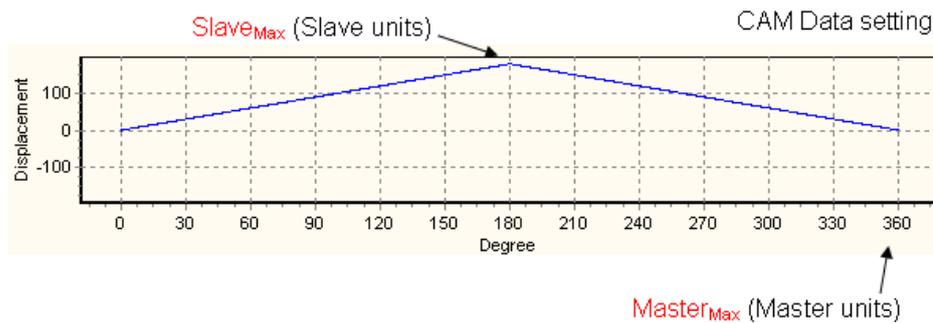


In above wiring diagram, we connect output terminal of Y axis (FP1+ · FP1-) to the MPG input terminals (A0+ · A0-) as the source of Master. By this wiring method, winding operation of coil shaft (Slave) will not be influenced no matter the winding direction of winding shaft (Master) is clockwise or counter-clockwise. Therefore, in this case we only connect single direction for Master.

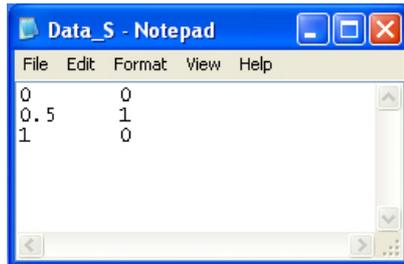
9.6.1.3 Create E-CAM Curve

After the wiring, create E-CAM Data in PMSOft as below.

9 Electrical CAM



To create E-CAM Data, import the file "Data_S.txt".



Firstly in this file we set the base unit of E-CAM by specifying SlaveMax=1 and MasterMax=1. Second, select motor unit for E-CAM operation by setting up b1/b0 of D1816. Third, specify the magnification of E-CAM Data according to the input parameters below. In addition, you can also select motor unit then real-time modify the SlaveMax/MasterMax value by DTO instruction. When input parameters are ready, we can obtain 2 settings: 1. Pulses of Winding shaft, 2. SlaveMax / MasterMax. The below table explains how to obtain the 2 settings.

Input parameters	Rounds per layer		N1
	Total rounds of winding		N2
	Coil spacing (mm)		D
	Winding shaft (Master/Y axis)	Mechanism parameter (mm/revolution)	There is no actual moving distance of Master because winding shaft is directly driven by signals. The winding principle is that Slave moves certain distance when Master rotates a round, therefore mechanism parameter of Master can be regarded the same as that of Slave. $A_{Master} = A_{Slave}$
		Servo parameter (pulses/revolution)	B_{Master}
		Mechanical parameter (mm/pulses)	$C_{Master} = A_{Master}/B_{Master}$
	Coil shaft (Slave/X axis)	Mechanism parameter (mm/revolution)	A_{Slave}
		Servo parameter (pulses/revolution)	B_{Slave}
		Mechanical parameter (mm/pulses)	$C_{Slave} = A_{Slave}/B_{Slave}$
Setting	Winding shaft (Master/Y axis)	Length of single speed positioning (pulses)	$=N2 \times B_{Master}$
	Coil shaft (Slave/X axis)	Master _{Max} (pulses)	$=2 \times N1 \times B_{Master}$ "2" indicates double layer winding
		Slave _{max} (pulses)	$=N1 \times D/C_{Slave}$

Here are 2 examples of obtaining the settings of Master and Slave

Example 1

Input parameters	Rounds per layer		$N1 = 10$	
	Total rounds of winding		$N2 = 80$	
	Coil spacing (mm)		$D = 0.2$	
	Winding shaft (Master/Y axis)	Mechanism parameter (mm/revolution)	<p>There is no actual moving distance of Master because winding shaft is directly driven by signals. The winding principle is that Slave moves certain distance when Master rotates a round, therefore mechanism parameter of Master can be regarded the same as that of Slave.</p> $A_{Master} = A_{Slave}$	
		Servo parameter (pulses/revolution)		$B_{Master} = 3600$
		Mechanical parameter (mm/pulses)		$C_{Master} = A_{Master}/B_{Master}$
	Coil shaft (Slave/X axis)	Mechanism parameter (mm/revolution)	$A_{Slave} = 10$	
		Servo parameter (pulses/revolution)	$B_{Slave} = 10000$	
		Mechanical parameter (mm/pulses)	$C_{Slave} = A_{Slave}/B_{Slave}$	
Settings	Winding shaft (Master/Y axis)	Length of single speed positioning (pulses)	$=N2 \times B_{Master} = 80 \times 3600 = 288000$	
	Coil shaft (Slave/X axis)	Master _{max} (pulses)	$=2 \times N1 \times B_{Master} = 2 \times 10 \times 3600 = 72000$ "2" indicates double layer winding	
		Slave _{max} (pulses)	$=N1 \times D/C_{Slave} = 10 \times 0.2 / (0.1/100)$ $=2000$	

Example 2

Input parameters	Rounds per layer		$N1 = 20$	
	Total rounds of winding		$N2 = 100$	
	Coil spacing (mm)		$D = 0.3$	
	Winding shaft (Master/Y axis)	Mechanism parameter (mm/revolution)	<p>There is no actual moving distance of Master because winding shaft is directly driven by signals. The winding principle is that Slave moves certain distance when Master rotates a round, therefore mechanism parameter of Master can be regarded the same as that of Slave.</p> $A_{Master} = A_{Slave}$	
		Servo parameter (pulses/revolution)		$B_{Master} = 3600$
		Mechanical parameter (mm/pulses)		$C_{Master} = A_{Master}/B_{Master}$
	Coil shaft (Slave/X axis)	Mechanism parameter (mm/revolution)	$A_{Slave} = 10$	
		Servo parameter (pulses/revolution)	$B_{Slave} = 10000$	
		Mechanical parameter	$C_{Slave} = A_{Slave}/B_{Slave}$	

9 Electrical CAM

		(mm/pulses)	
Settings	Winding shaft (Master/Y axis)	Length of single speed positioning (pulses)	$=N2 \times B_{Master} = 100 \times 3600 = 360000$
	Coil shaft (Slave/X axis)	Master _{Max} (pulses)	$=2 \times N1 \times B_{Master} = 2 \times 20 \times 3600 = 144000$ "2" indicates double layer winding
		Slave _{max} (pulses)	$=N1 \times D/C_{Slave} = 20 \times 0.3 / (0.1/100)$ $=6000$

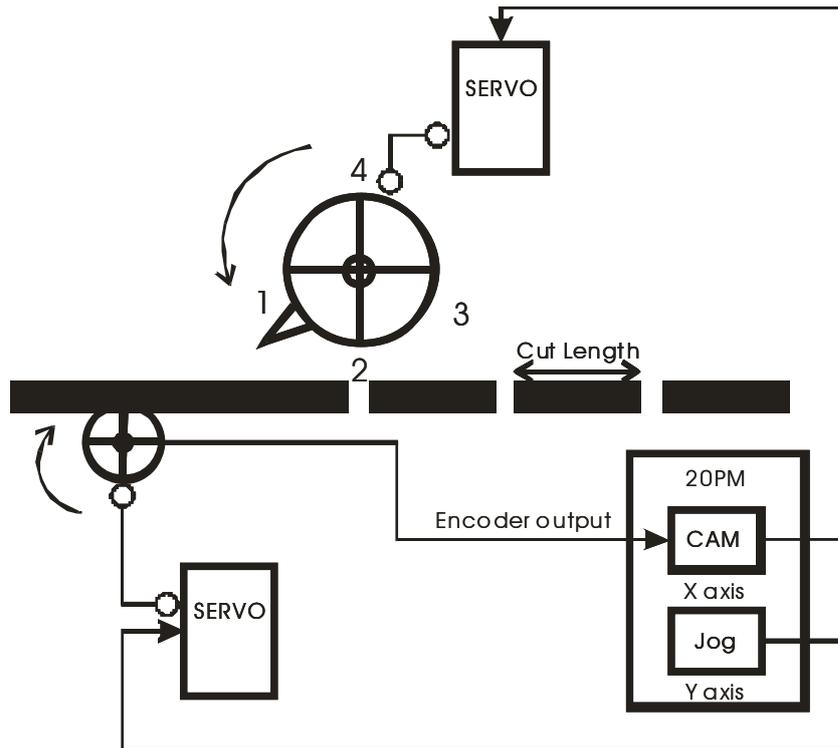
9.6.2 E-CAM Application on Rotary Cut and Flying Saw Devices

The traditional material feeding and cutting applications apply go-and-stop pattern. The feeding control axis moves the material for a fixed distance and the cutting control axis performs the cutting process. In this pattern, feeding control axis repeats the move-and-stop actions while cutting control axis repeats cut-and-stop actions. Therefore, a go-and-stop working pattern is formed. However, the production efficiency of the pattern is low due to the deceleration and acceleration time generated in go-and-stop process. Aiming at this problem, most current cutting applications apply continuous cutting patterns, such as rotary cut and flying saw, which have some difference with each other on E-CAM data design. Flying saw operates in reciprocating movements while rotary cut operates in the same direction. In addition, thick material cutting also works in the same direction as rotary cut. The difference between thick material cutting and rotary cut is the E-CAM curve variation in cutting process. Unlike the constant proportion curve of rotary cut's sync area, the E-CAM curve of thick material cutting varies during the cutting process between the start and the end of cutting. The 3 E-CAM applications are explained as below.

9.6.2.1 Rotary Cut

When the rotary cutter performs cutting action, the feeding conveyor does not slow down and stop. When the cutter touches the material, the moving speed of cutter should be the same as that of the feeding conveyor. If the cutter is slower than the conveyor, the material will be squeezed and piled; if the cutter is faster than the conveyor, the material will be extended and damaged.

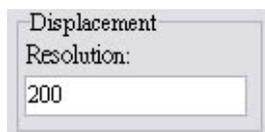
The operation and simple wiring of rotary cut is illustrated as below. Positions 1, 2, 3 and 4 respectively indicate starting position, middle position, end position and ready position of speed synchronization. When Master (Y axis) executes, Slave (X axis) accelerates from position 4 to position 1, reaching the synchronizing speed. The sync speed is maintained from position 1 to position 3. After position 3, Slave decelerates and returns to position 4. The cycle repeats for continuous rotary cut operation.



■ Create E-CAM Curve

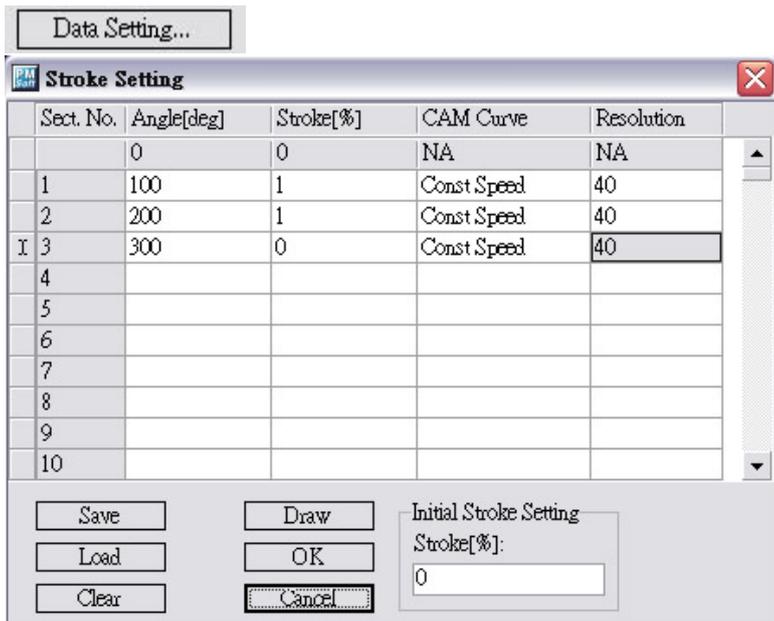
In E-CAM design process, PMSoft creates E-CAM data by operating displacement data to generate the velocity data and the acc data. However, in this case we need to control the velocity between Slave and Master. To obtain the velocity, we can input the velocity data in the displacement table for creating the velocity curve. After the velocity curve is created with displacement and acc curve, export the velocity data by clicking “Export” and import the velocity data by clicking “Import Speed Data”, so that the velocity relationship between Slave and Master can be obtained.

1. Set the max resolution as 200.

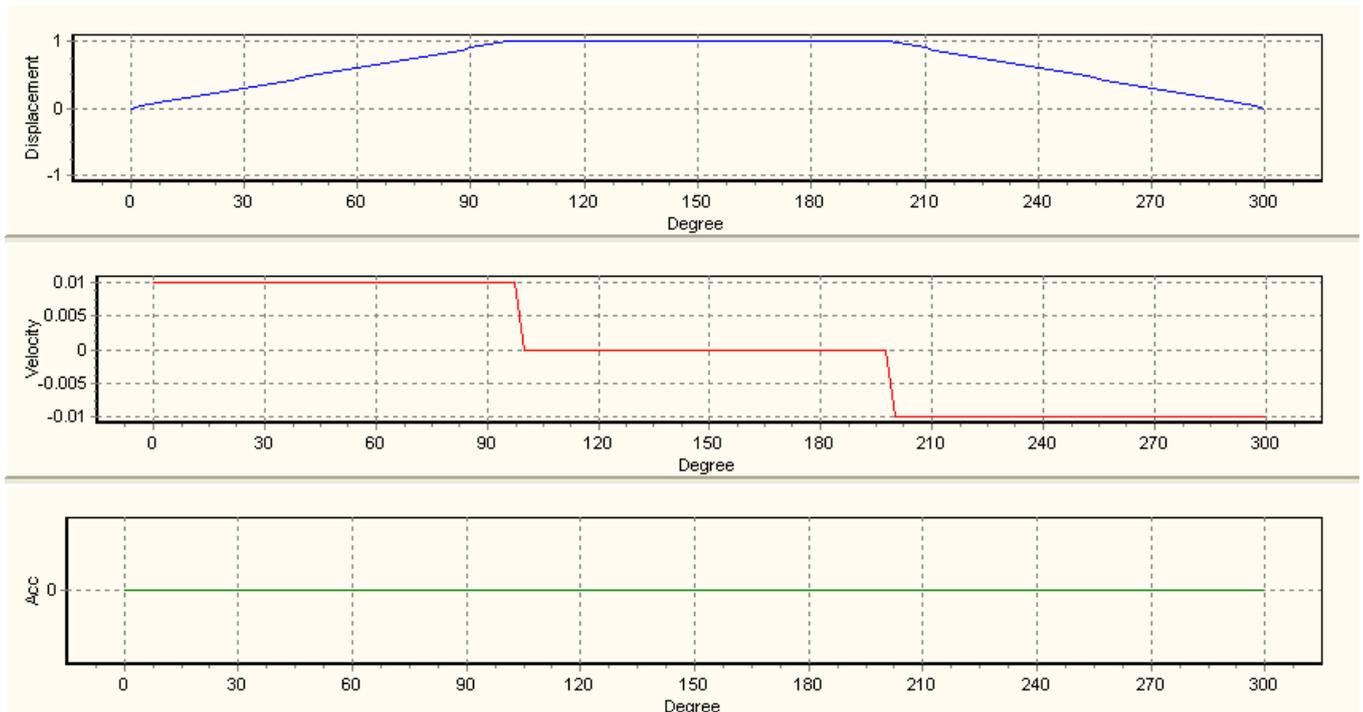


2. Input the velocity data in the displacement table.
Position 1: 100 degrees, position 3: 200 degrees, position 4: 300 degrees.

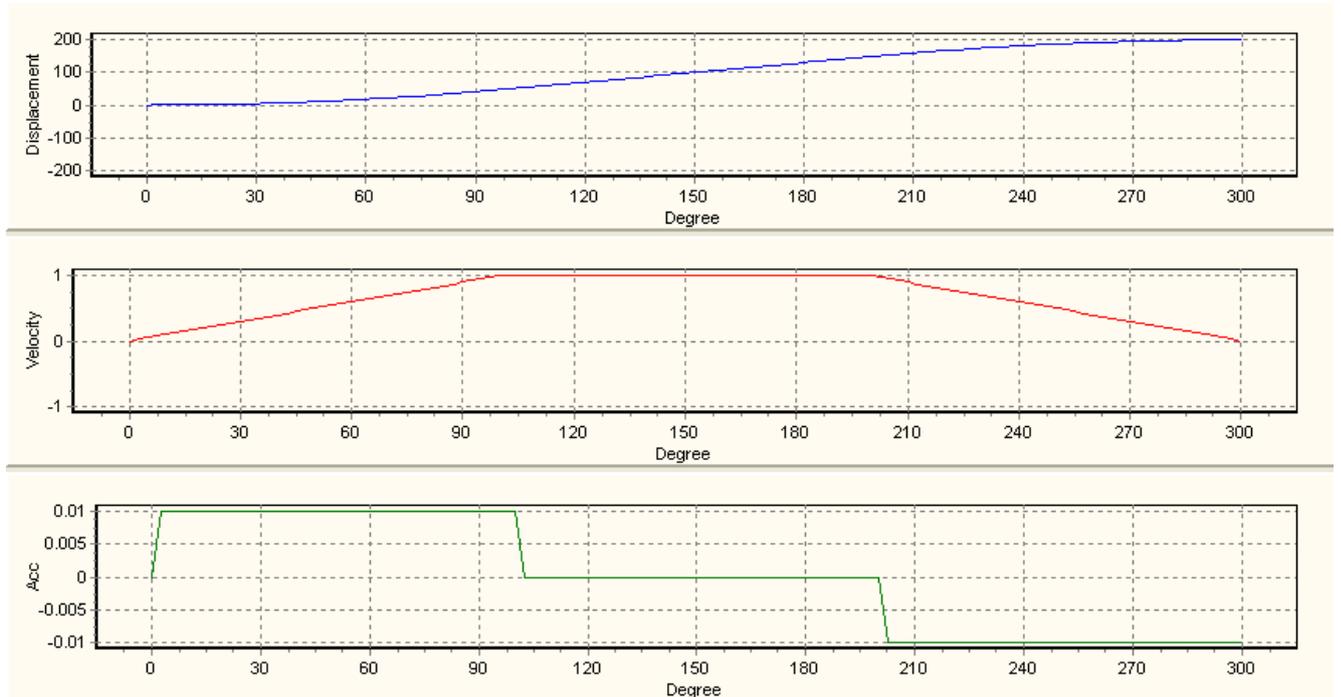
9 Electrical CAM



Click "Draw" and the displacement data can be obtained as below.



3. Click "Export" then "Import Speed Data" and the velocity relationship between Slave and Master can be obtained.

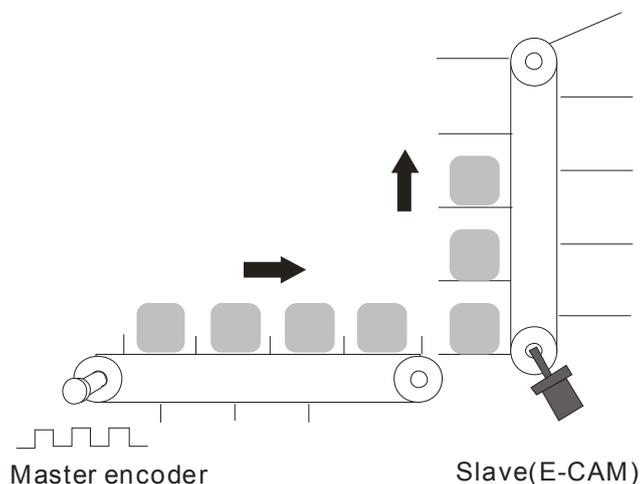


According to the above E-CAM Chart, we can calculate the pulses per round of Slave (Slavemax=200) by the square measure of velocity data.

$$(1 \times 100 / 2 + 1 \times (200 - 100) + 1 \times (300 - 200) / 2) = 200$$

Pulses per round of Master equal to the cut length. You can apply input /output magnification setting at 9.3.1.7 in this chapter to obtain the proper set value.

The below diagram is another application on synchronizing conveyers. The E-CAM operation of the example is similar to flying saw. However, care should be taken on setting the CAM curve of Slave. Constant speed for Slave is recommended. If the CAM curve is set the same as the flying saw, the vertical conveyor can not keep smooth moving speed due to the acceleration and deceleration process of flying saw.

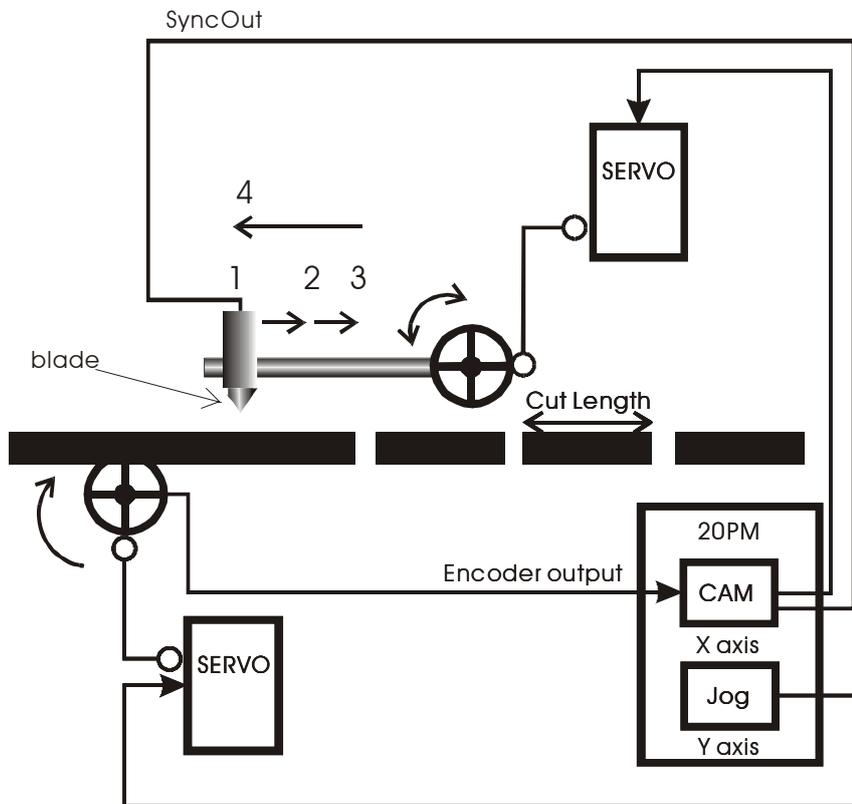


9 Electrical CAM

9.6.2.2 Flying Saw

When the flying cutter performs cutting action, the feeding conveyor does not slow down and stop, and the moving speed of cutter should be the same as that of the feeding conveyor. In addition, the synchronizing time should be long enough for the cutter to finish the cutting process and return to the safe position. The Slave (E-CAM) controls the cutter as well as the whole cutting device to move forward and back and synchronize with the Master (conveyor) during cutting process.

The operation and simple wiring of flying saw is illustrated as below. Positions 1, 2, 3 and 4 respectively indicate starting position, sync-start position, sync-end position and ready position. When Master (Y axis) executes, Slave (X axis) accelerates from position 1 to position 2, reaching the synchronizing speed. The sync speed is maintained from position 2 to position 3. After position 3, Slave decelerates in reverse direction and returns to position 4 (Same as position 1). The cycle repeats for continuous flying saw operation.



■ Create E-CAM Curve

In E-CAM design process, PMSOFT creates E-CAM data by operating displacement data to generate the velocity data and the acc data. However, in this case we need to control the velocity between Slave and Master. To obtain the velocity, we can input the velocity data in the displacement table for creating the velocity curve. After the velocity curve is created with displacement and acc curve, export the velocity data by clicking "Export" and import the velocity data by clicking "Import Speed Data", so that the velocity relationship between Slave and Master can be obtained.

1. Set the max resolution as 200.

Displacement
Resolution:
200

2. Input the velocity data in the displacement table.

Position 2: 100 degrees, position 3: 200 degrees, position 4: 700 degrees.

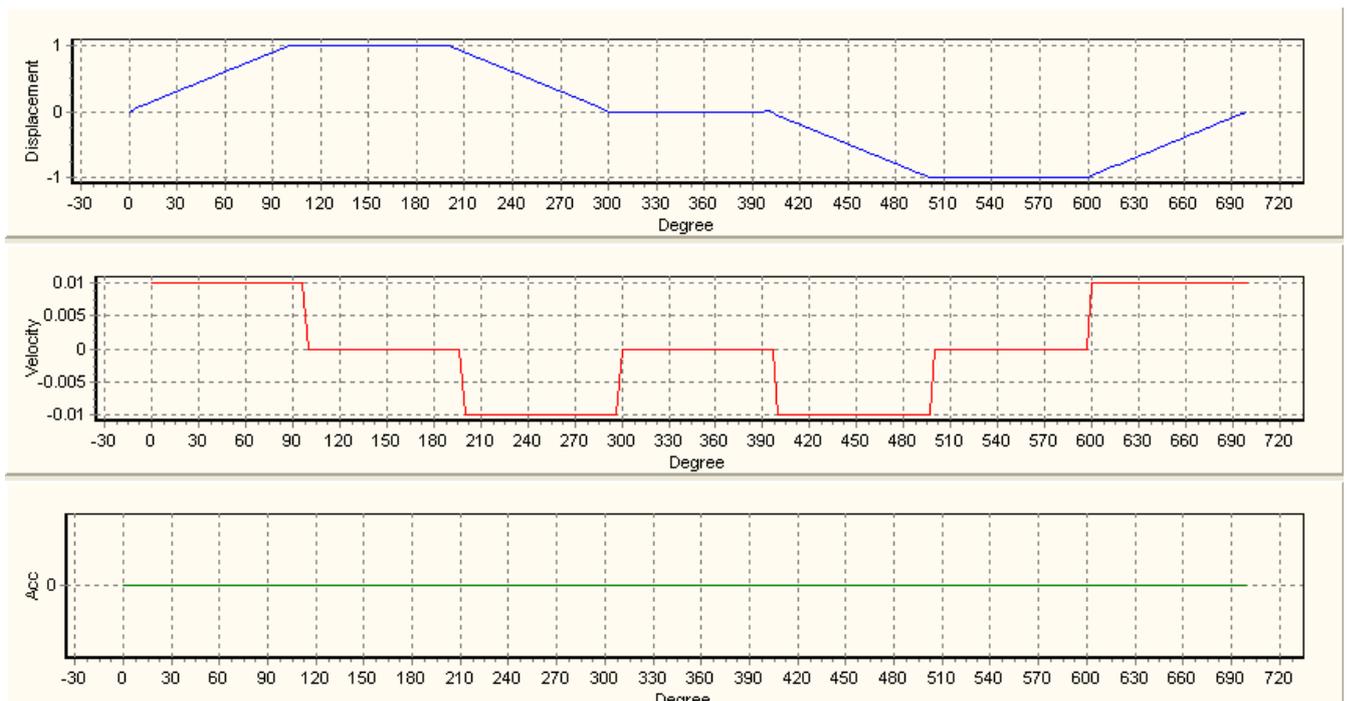
Data Setting...

Stroke Setting

Sect. No.	Angle[deg]	Stroke[%]	CAM Curve	Resolution
	0	0	NA	NA
1	100	1	Const Speed	
2	200	1	Const Speed	
3	300	0	Const Speed	
4	400	0	Const Speed	
5	500	-1	Const Speed	
6	600	-1	Const Speed	
7	700	0	Const Speed	
8				
9				
10				

Save Draw Initial Stroke Setting
 Load OK Stroke[%]:
 Clear Cancel 0

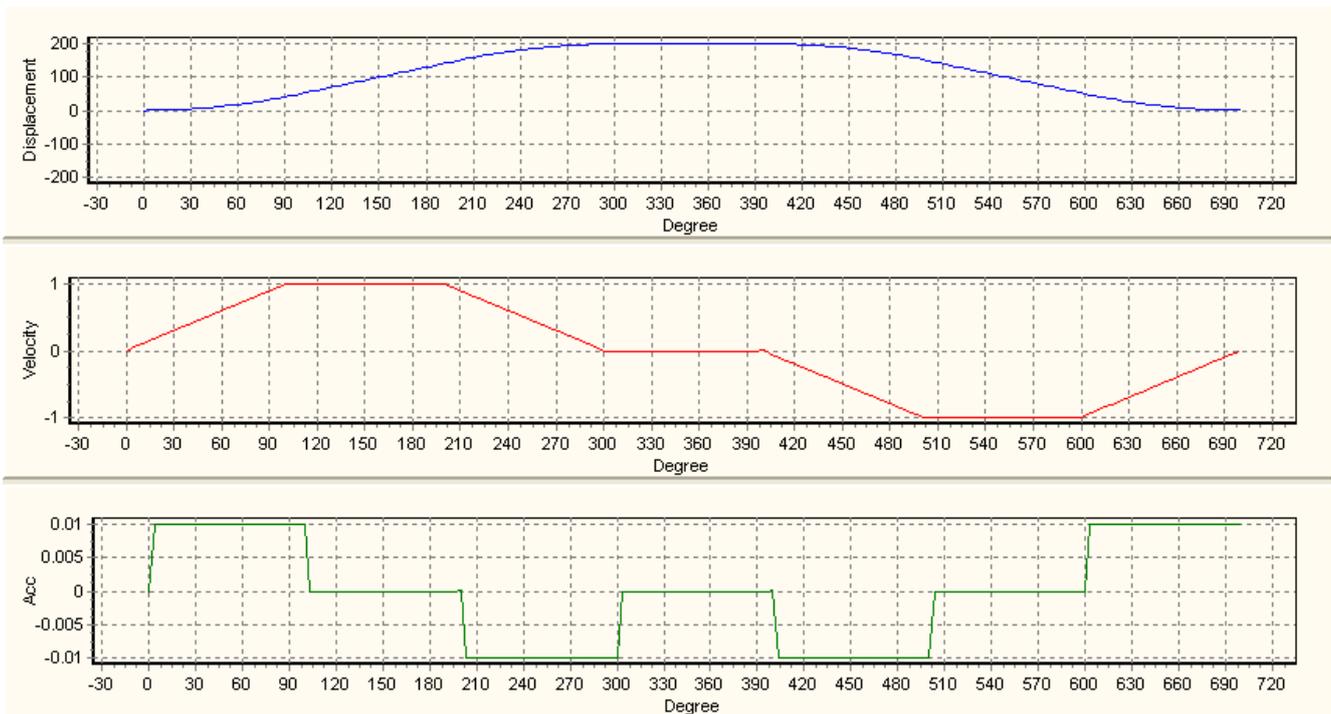
Click "Draw" and the displacement data can be obtained as below.



3. Click "Export" then "Import Speed Data" and the velocity relationship between Slave and Master can be obtained.

9 Electrical CAM

Export → Import Speed Data →



According to the above E-CAM Chart, we can calculate the pulses per round of Slave (Slavemax=400) by the square measure of velocity data.

$$[(1 \cdot 100 / 2 + 1 \cdot (200 - 100) + 1 \cdot (300 - 200) / 2)] \cdot 2 = 400$$

In addition, for pulses per round of Master you can apply input /output magnification setting at 9.3.1.7 in this chapter to obtain the proper set value. Different from the setting of rotary cut, lower/higher bound of E-CAM synchronized output section should be specified. Please refer to point 4 of 9.3.1.3 in this chapter. Also, to save the time of operation cycle you can set a higher speed for the process when cutter returns to position 4. See below example.

Sect. No.	Angle[deg]	Stroke[%]	CAM Curve	Resolution
	0	0	NA	NA
1	100	1	Const Speed	33
2	200	1	Const Speed	33
3	300	0	Const Speed	33
4	400	0	Const Speed	33
5	500	-2	Const Speed	33
I 6	600	0	Const Speed	35
7				
8				
9				
10				

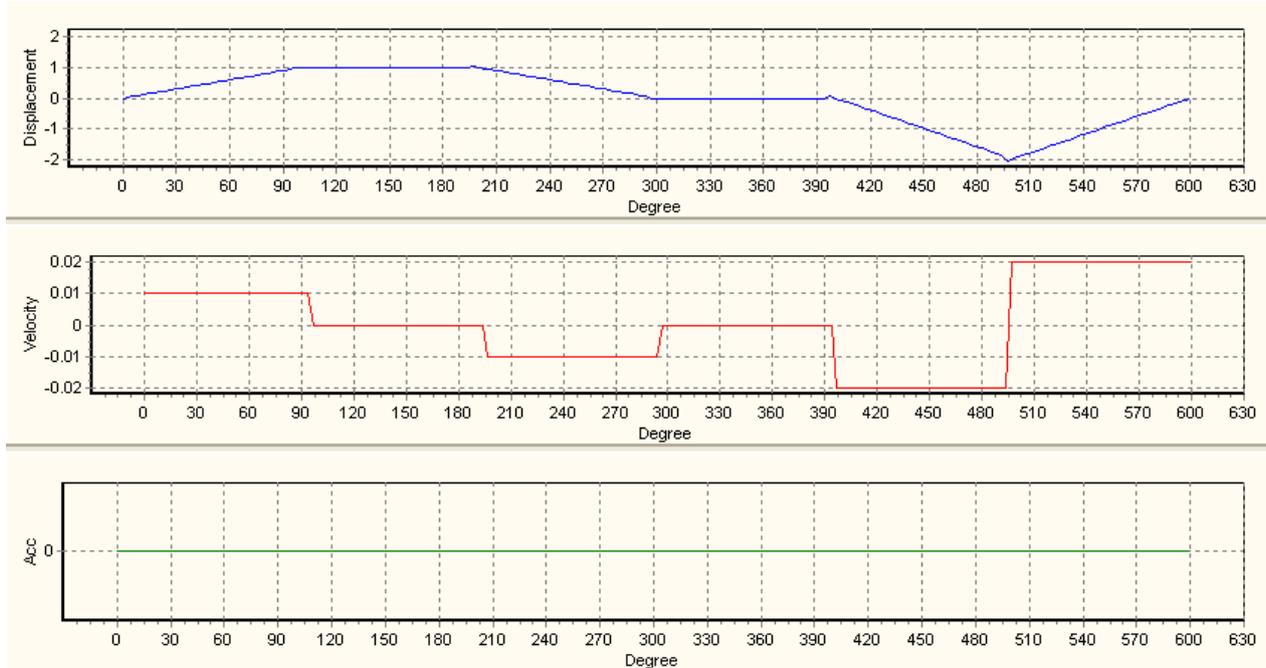
Buttons: Save, Draw, Load, OK, Clear, Cancel

Initial Stroke Setting
Stroke[%]: 0

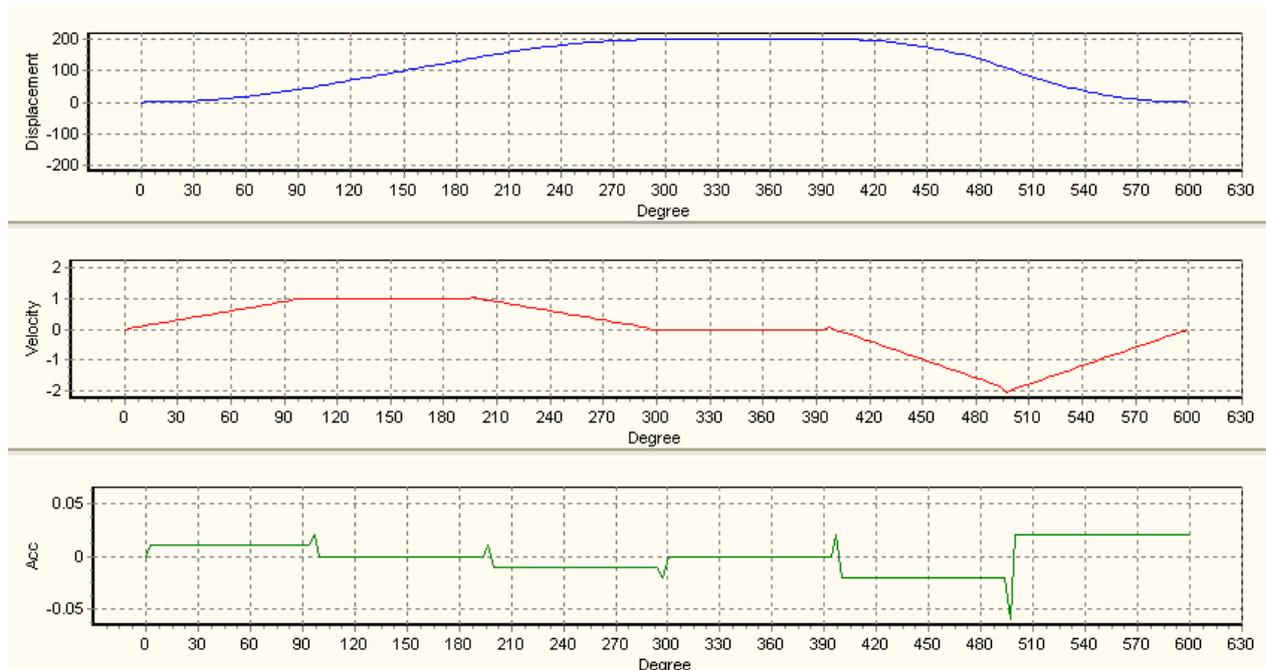
In this example, modify the angle of position 4 from 700 to 600 degrees and set the stroke at section 5 as -2,

indicating doubling the reverse running speed. In this case, 100 pulses (700-600) can be reduced at operation speed 100kHz, i.e. approximately 1ms can be saved from the whole operation cycle. However, please note that the square measure of positive pulses should be the same as that of the negative pulses if the ready position (position 4) is required to match starting position (position 1).

The modified displacement data:



The imported speed data:



Positive square measure:

$$(1 \cdot 100 / 2 + 1 \cdot (200 - 100) + 1 \cdot (300 - 200) / 2) = 200$$

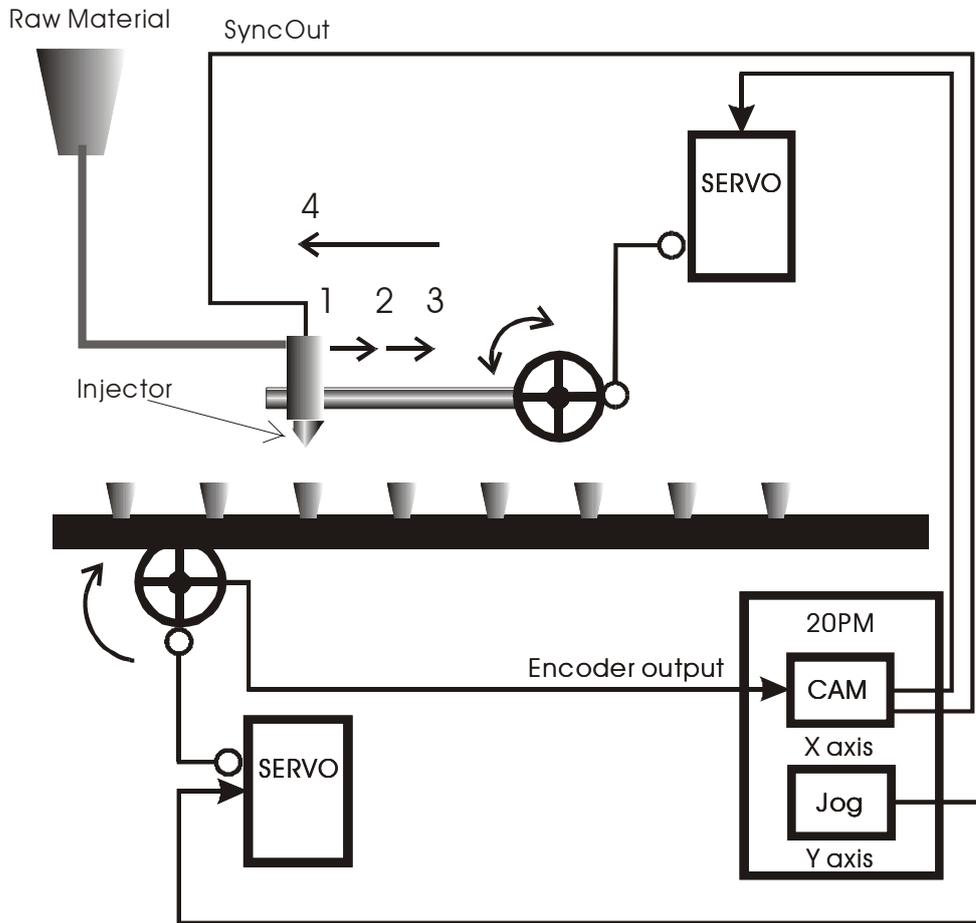
Negative square measure:

$$(-2 \cdot (600 - 400) / 2) = -200$$

9 Electrical CAM

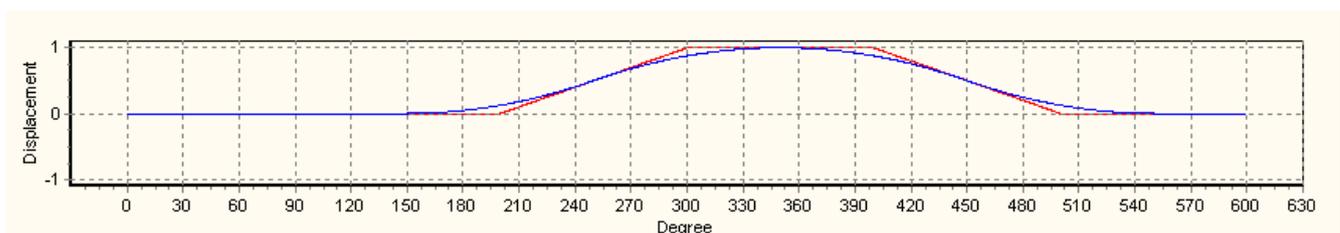
9.6.2.3 Flying Saw – Liquid Filling Machine

The below diagram is another flying saw application: liquid filling machine. The E-CAM operation of the example is similar to the above case of flying saw. The only difference is that the cutter is replaced by the injector. See the simple wiring diagram as below:



Filling machine

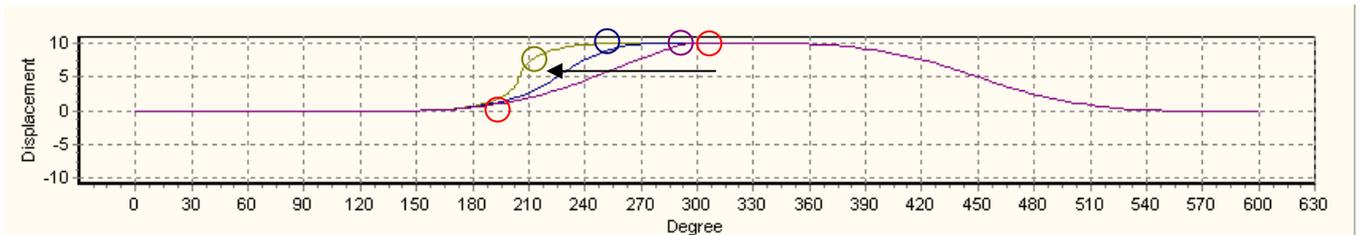
PMSOft provides B-Spline function for CAM curve setting. B-Spline can smooth the CAM curve as well as the positioning process. By selecting B-Spline as the CAM curve, we can reduce the abrasion of the mechanism and extend its life span. The B-Spline in PMSOft ensures the smoothness and continuity of the designed CAM curve in 2nd order differentiation, i.e. selecting B-Spline for displacement curve also ensures the continuity of velocity curve and acc curve. The results of the same E-CAM data with 2 different CAM curves are as below. Red curve indicates Const Speed and blue curve indicates B-Spline.



Sect. No.	Angle[deg]	Stroke[%]	CAM Curve	Resolution
	0	0	NA	NA
▶ 1	100	0	Const Speed	341
2	200	0	Const Speed	341
3	300	1	Const Speed	341
4	400	1	Const Speed	341
5	500	0	Const Speed	341
6	600	0	Const Speed	343
7				
8				
9				
10				

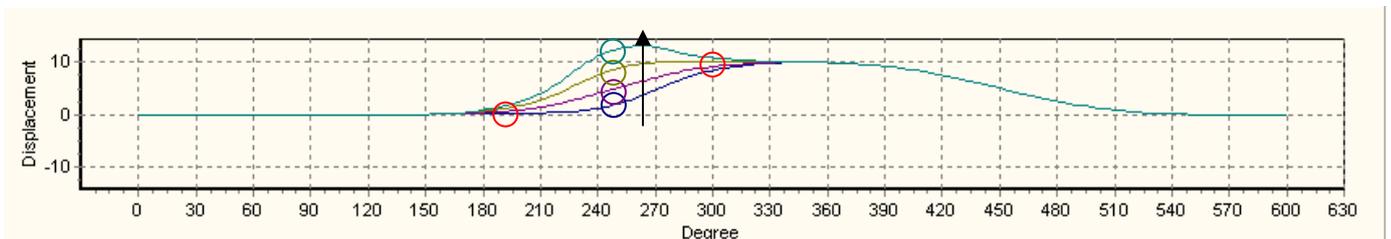
Sect. No.	Angle[deg]	Stroke[%]	CAM Curve	Resolution
	0	0	NA	NA
▶ 1	100	0	BSpline	341
2	200	0	BSpline	341
3	300	1	BSpline	341
4	400	1	BSpline	341
5	500	0	BSpline	341
6	600	0	BSpline	343
7				
8				
9				
10				

The property of B-Spline is to modify only partial curve rather than the whole curve, i.e. if additional points between 2 Master positions are inserted, only the curve in the section is modified, and the curve besides the section remains in the same shape. Here is the example of inserting 210, 250, and 290 between Master position 200 and 300. We can observe the curve variation of each point.

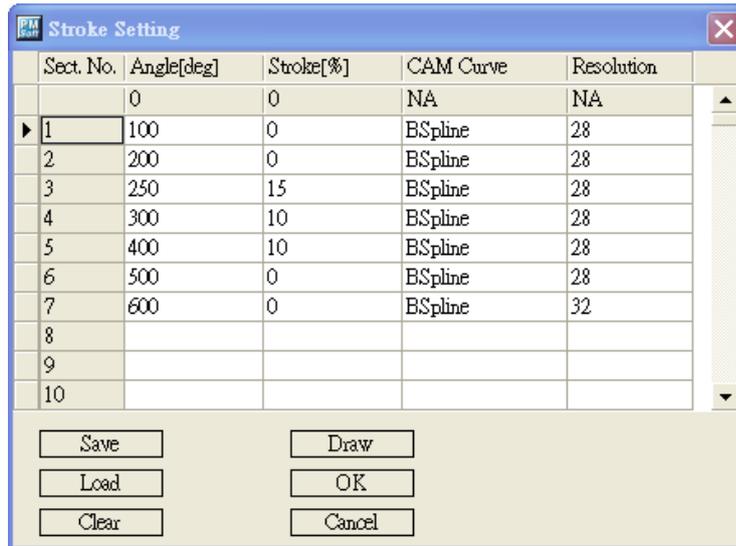


Sect. No.	Angle[deg]	Stroke[%]	CAM Curve	Resolution
	0	0	NA	NA
▶ 1	100	0	BSpline	28
2	200	0	BSpline	28
3	210	10	BSpline	28
4	300	10	BSpline	28
5	400	10	BSpline	28
6	500	0	BSpline	28
7	600	0	BSpline	32
8				
9				
10				

The above example illustrates the horizontal variation of inserting different Master positions. Here is another example explaining the vertical variation of inserting different Slave positions 1, 5, 10 and 15.



9 Electrical CAM



The image shows a 'Stroke Setting' dialog box with a table of parameters and a set of control buttons. The table has five columns: Sect. No., Angle[deg], Stroke[%], CAM Curve, and Resolution. The data is as follows:

Sect. No.	Angle[deg]	Stroke[%]	CAM Curve	Resolution
	0	0	NA	NA
▶ 1	100	0	BSpline	28
2	200	0	BSpline	28
3	250	15	BSpline	28
4	300	10	BSpline	28
5	400	10	BSpline	28
6	500	0	BSpline	28
7	600	0	BSpline	32
8				
9				
10				

Below the table are six buttons arranged in two columns: Save, Load, Clear on the left; Draw, OK, Cancel on the right.

10.1 How to Set

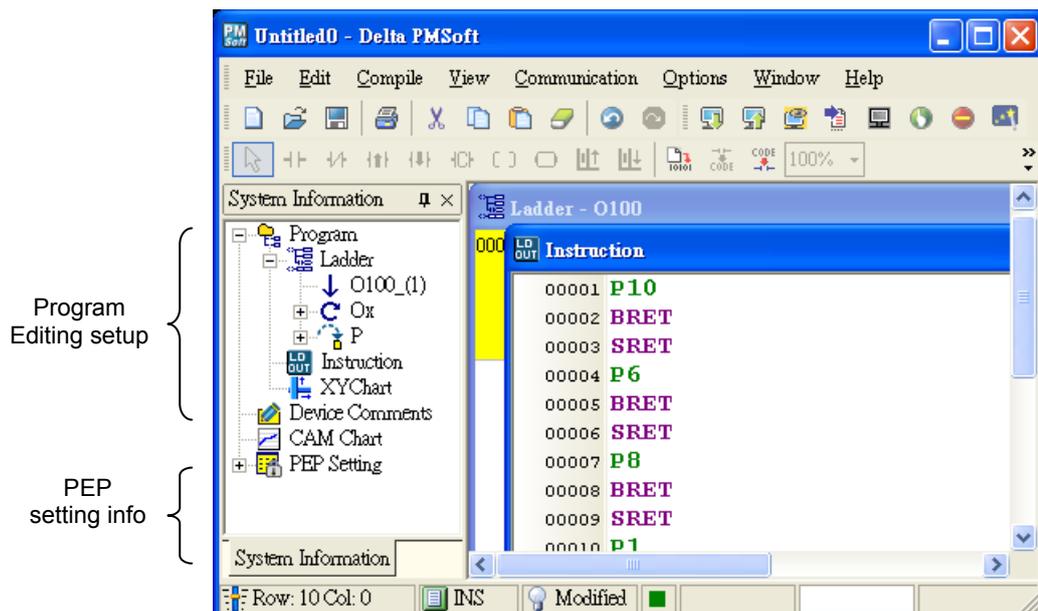
DVP-PM allows users to set up passwords to protect user program through PMSOft. Set up the password before downloading the program into DVP-PM so that the program is protected from being illegally intercepted or modified. Therefore, before uploading the encrypted program to PMSOft for modifications, please enter the correct password first. In PMSOft, you can encrypt O100 main program, 100 Oxn ($n = 0 \sim 99$) motion subroutines and 256 Pm ($m = 0 \sim 255$) subroutines.

The setup process contains three parts: system information, download and upload.

10.1.1 System Information

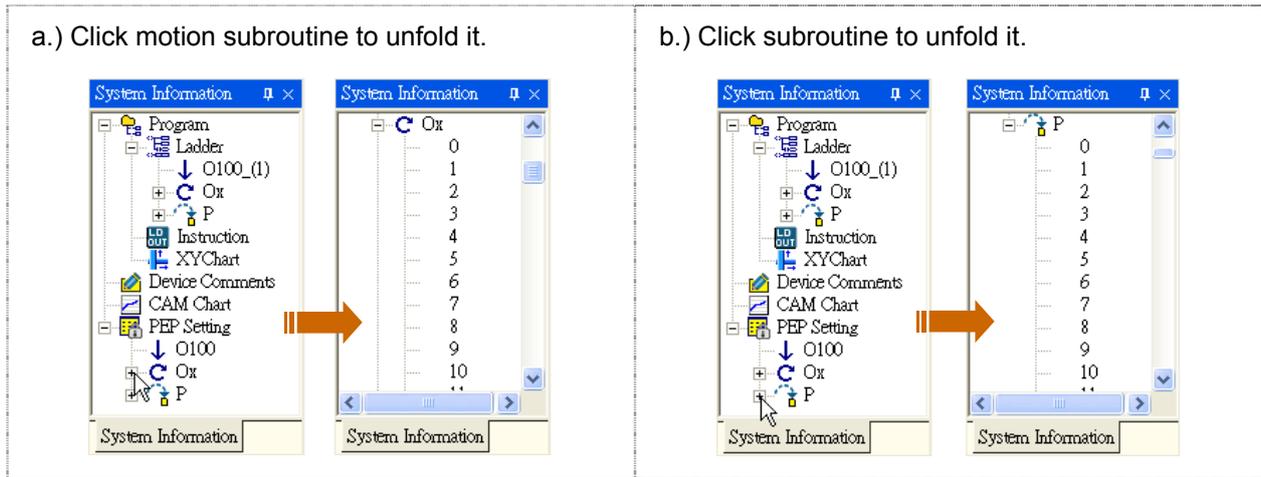
The system information area contains 2 parts: (1) Program, for switching between editing windows and acquiring the current program status; (2) PEP Setting, the encryption to protect the program. In the editing window, there are total 357 programs can be protected by encryption, including 1 O100 main program, 100 Oxn motion subroutines and 256 Pm subroutines. See paragraphs below for the explanations on the functions in the system information area.

- Displaying system information and program encryption setting



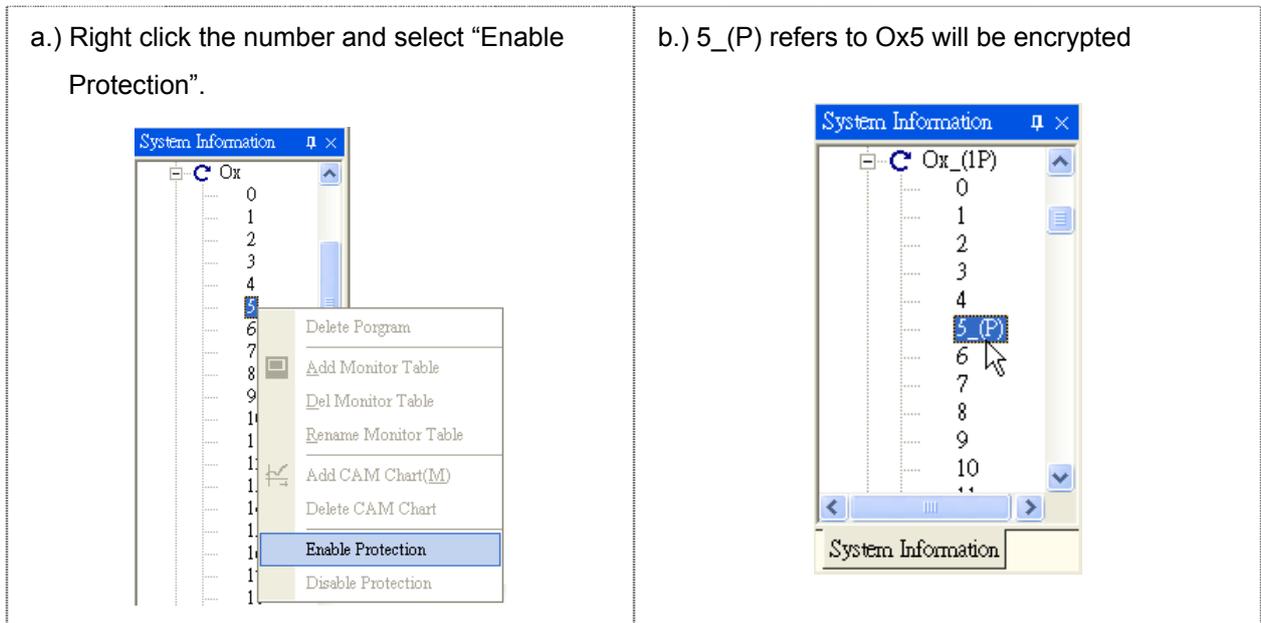
- Unfolding motion subroutine and P subroutine: Click the motion subroutine or P subroutine to unfold the subroutine number. There are 100 motion subroutines and 256 P subroutines.

10 Encrypting User Program



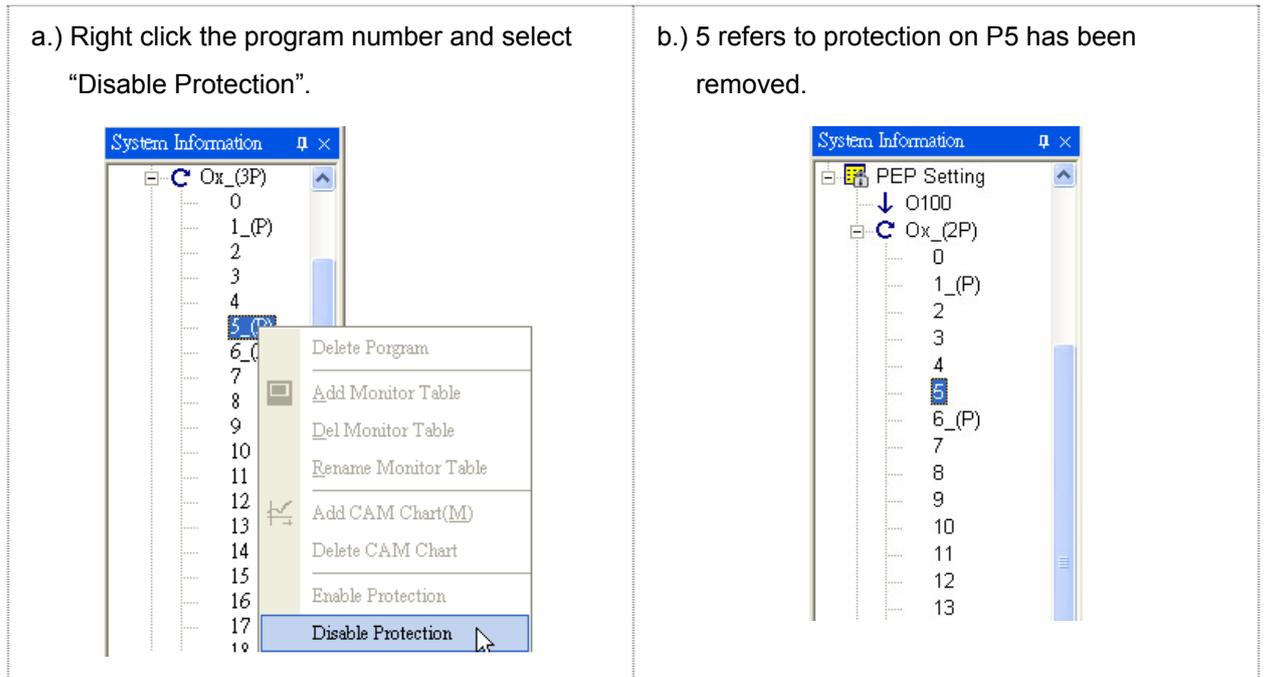
■ Applying encryption to specific program:

1. Right click the motion subroutine or P subroutine number to be encrypted and select “Enable Protection” as shown in a.) below.
2. The encrypted program number will have a “_(P)” mark after it, meaning the program has been protected, as shown in b.) below. Ox_(1P) means “1” motion subroutine will be encrypted, and Ox_4(P) means “4” motion subroutines will be encrypted. The same rule also applies to P subroutines. However, there is only 1 main program; therefore, enable the protection directly on O100 and make it O100_(P).



■ Removing the protection

1. Right click the motion subroutine or P subroutine and select “Disable Protection”, as shown in a.) below.
2. The “_(P)” mark after the encrypted program number will be removed as shown in b.) below. P_(3P) refers to 3 subroutines are encrypted. After 1 protection is removed, it will become P_(2P).

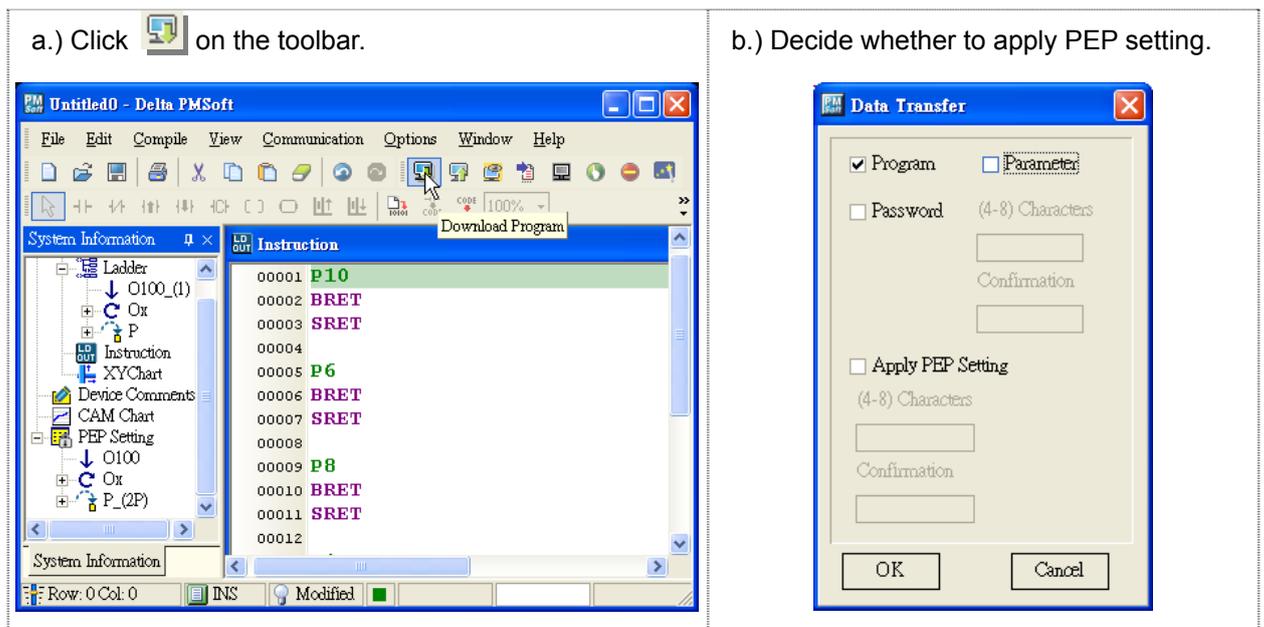


10.1.2 Downloading Program

When you download the program, you will see a window asking whether to apply PEP setting. Check “Apply PEP Setting, and PMSOft will ask you to set up the encryption password. If not, PMSOft will download all programs to the general program storage area in DVP-PM. Encrypted programs or E-CAM data will be placed in the protection area.

■ Check “Apply PEP Setting”

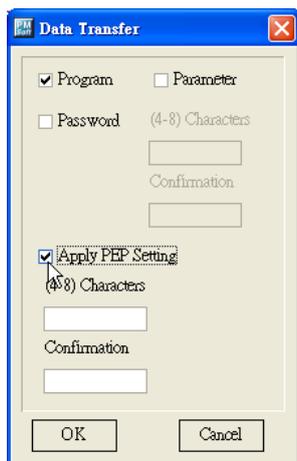
1. Select “Communication (C)” and click “PC→PM(D)”, or click  on the toolbar, as shown in a.) below, and decide whether to apply PEP setting, as shown in b.) below.



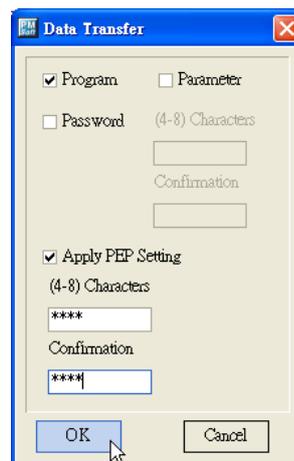
2. Check “Apply PEP Setting”, as a.) below, and you will need to enter 4 ~ 8 characters of password and confirm the password, as b.) If this is the first time you set up the password, PMSOft will directly download the program or E-CAM data to DVP-PM during the program transmission, as c.).

10 Encrypting User Program

a.) Check "Apply PEP Setting".



b.) Enter password and confirm password.



c.) Start downloading the program if this is the first time for password setup.



3. During the program transmission process, if the program has been encrypted, PMSoft will ask you to enter the password, as a.). Click "OK" to start downloading the program or E-CAM data to DVP-PM, as b.).

a.) Enter the password.



b.) Press "OK" and start to download the program.

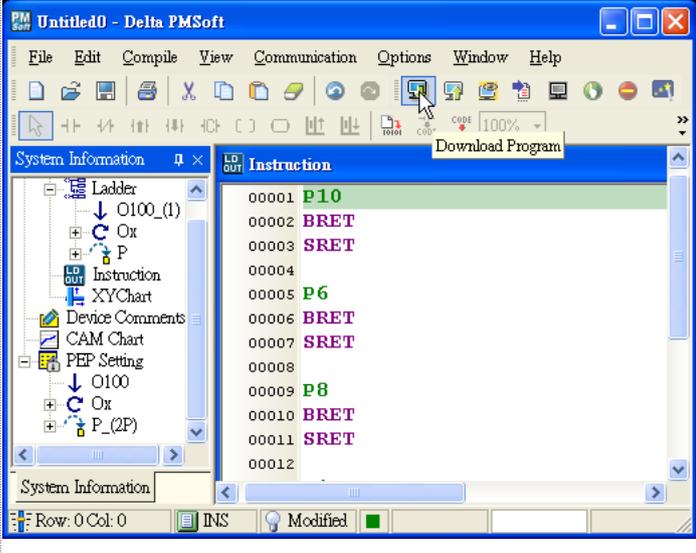


■ No "Apply PEP Setting"

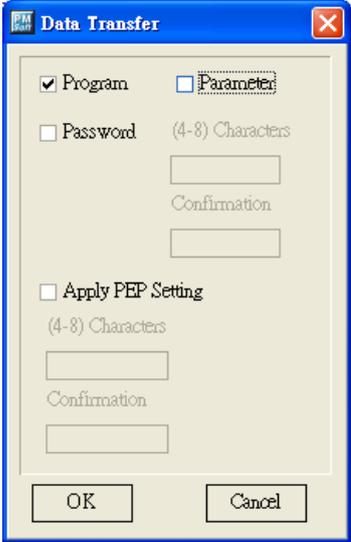
1. Select "Communication (C)" and click "PC→PM(D)", or click  on the toolbar, as shown in a.) below, and decide whether to apply PEP setting, as shown in b.) below.

10 Encrypting User Program

a.) Click  on the toolbar

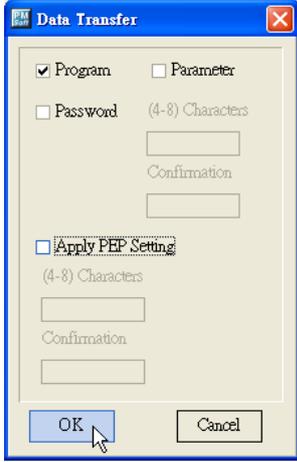


b.) Decide whether to apply PEP setting.



2. Leave “Apply PEP Setting” box blank, as a.) If DVP-PM is currently set with a password and there are E-CAM data in the program to be downloaded, PMSOft will ask you to enter the password (to unlock the protection for storing E-CAM data). Click “OK”, as b.) to start downloading the program to DVP-PM, as c.).

a.) Leave “Apply PEP Setting” box blank.



b.) Enter the password (if DVP-PM is set with a password and the downloading program contains E-CAM data).



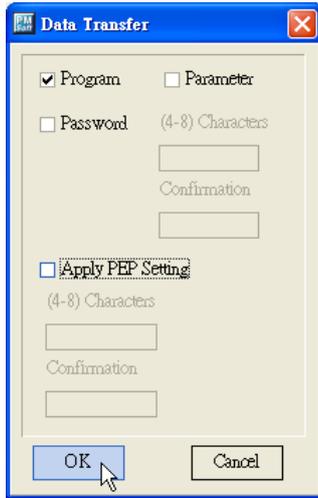
c.) Start downloading the program.



3. During the program transmission process, if DVP-PM is currently set with a password but there are no E-CAM data in the program to be downloaded, the system will directly download the program to DVP-PM, as b.).

10 Encrypting User Program

a.) Leave “Apply PEP Setting” box blank.

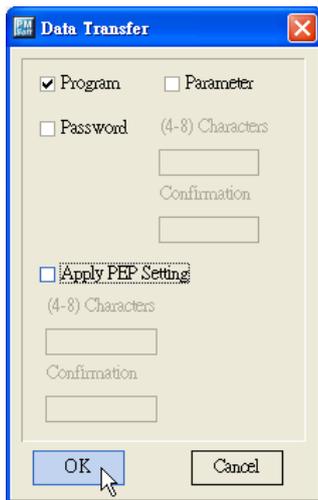


b.) Start downloading the program (if DVP-PM is set with a password and the downloading program contains no E-CAM data).



4. During the program transmission process, if DVP-PM is not set with a password, the system will directly download the program to DVP-PM whether there are E-CAM data or not, as b.).

a.) Leave “Apply PEP Setting” box blank.



b.) Start downloading the program when DVP-PM is not set with a password. (whether there is E-CAM data or not)



10.1.3 Uploading Program

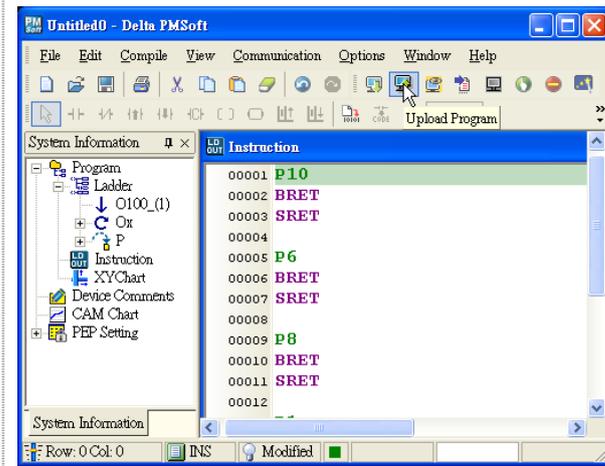
When you upload the program, you will see a window asking whether to read PEP protection area in DVP-PM. If “PEP” box is not checked, only programs in general program area of DVP-PM will be uploaded. If “PEP” box is checked, you have to enter the protection password so that the programs in general program area as well as the programs in PEP protection area will be uploaded. See the explanations on the setting steps of the three cases.

■ No protection set in DVP-PM

1. Select “Communication (C)” and click “PM→PC (U)”, or click  on the toolbar, as shown in a.) below, and decide whether to read the program in PEP protection area, as shown in b.) below.

10 Encrypting User Program

a.) Click  on the toolbar.

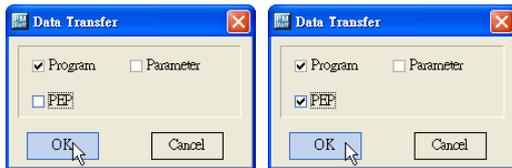


b.) Decide whether to read the program in the protection area.



2. Whether you decide to read PEP protection area or not, once you click “OK”, as a.), the general program or E-CAM data will be directly read and uploaded to PC, as b.).

a.) Decide whether to read PEP. Click “OK”.



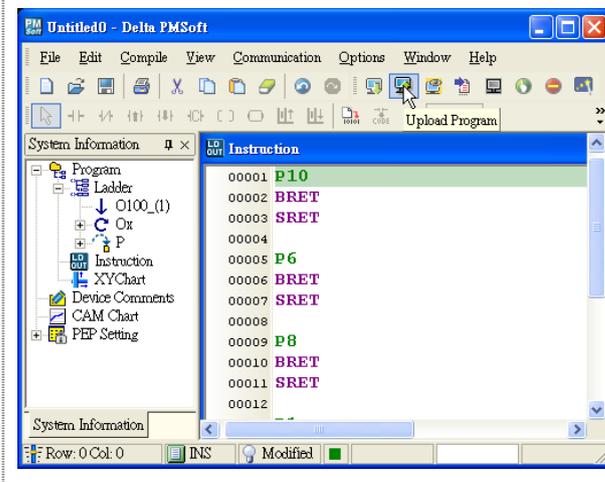
b.) Start downloading the program.



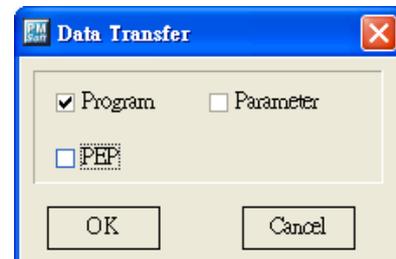
■ DVP-PM is set with protection but contains no E-CAM data

1. Select “Communication (C)” and click “PM→PC (U)”, or click  on the toolbar, as shown in a.) below, and decide whether to read the program in PEP protection area, as shown in b.).

a.) Click  on the toolbar.



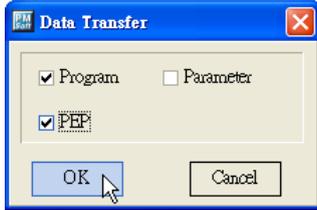
b.) Decide whether to read the program in the protection area.



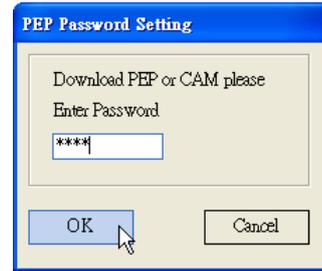
2. Check “PEP” to read and click “OK”, as a.). Next, enter the password and confirm it, as b.), to start uploading the protected program to PC, as c.).

10 Encrypting User Program

a.) Check “PEP” to read the protected program.
Click “OK”.



b.) Enter password and confirm it.



c.) Start downloading the program.



3. If “PEP” is not checked, click “OK”, as a.), and the system will upload the program directly to PC, as b.).

a.) Leave “PEP” box blank.



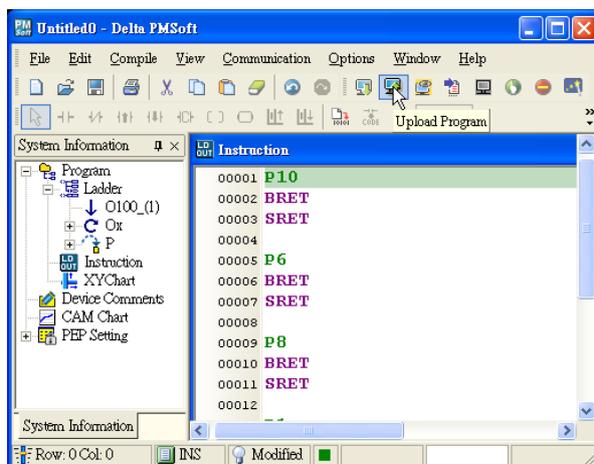
b.) Start uploading the program when “PEP” is not checked



■ DVP-PM is set with protection and contains E-CAM data

1. Select “Communication (C)” and click “PM→PC (U)”, or click  on the toolbar, as shown in a.) below, and decide whether to read the program in PEP protection area, as shown in b.).

a.) Click  on the toolbar.



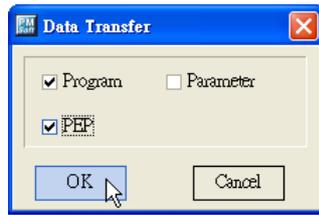
b.) Decide whether to read the program in the protection area.



2. Check “PEP” to read the protection area and click “OK”, as a.). Next, enter the password and confirm it, as b.), to start uploading general program, protected program and E-CAM data to PC, as c.).

10 Encrypting User Program

a.) Check "PEP" to read it. Click "OK".



b.) Enter password and confirm it.

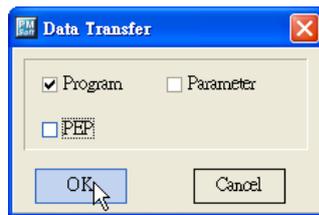


c.) Start uploading the program.

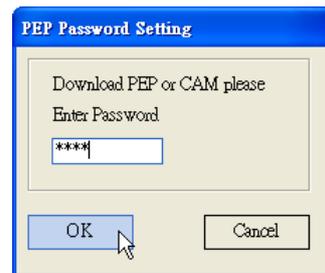


3. Leave "PEP" box blank and click "OK", as a.). Next, enter the password and confirm it, as b.), to start uploading general program and E-CAM data to PC, as c.)

a.) Leave "PEP" box blank. Click "OK".



b.) Enter password and confirm it.



c.) Start uploading the program.



10 Encrypting User Program

MEMO

Details about G-code instructions are explained in previous chapters. In this chapter, we will focus on G-code applications which improve the usability by various G-code download methods and enhance the operation stability by advanced G-code application (mainly used on dispensing machine).

The CNC (Computer Numerical Control) machines applying G-codes have the same operation methods and functions, therefore the PLC program for CNC machine do not need to be modified for different CNC machines.

Users only need to preset the source of G-codes:

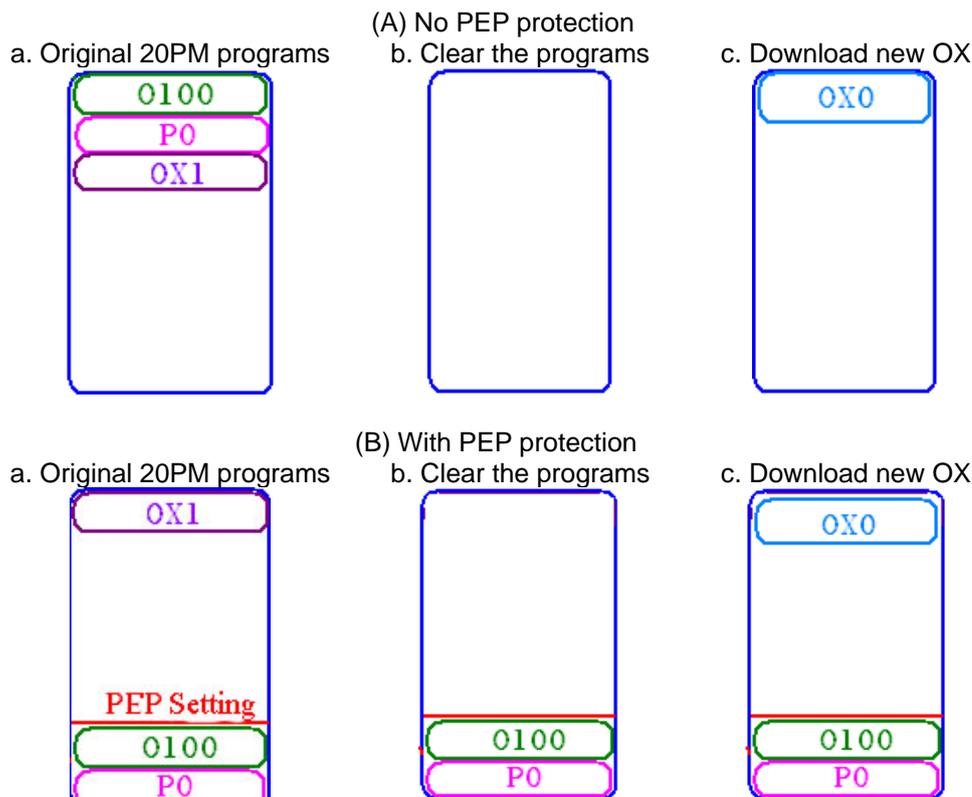
1. HMI: editing and storing G-codes on HMI,
2. Memory card: convert the file on CAM software into G-codes. Store the G-codes in memory card and insert the card into the slot on CNC machine,
3. PC(NB): directly download the G-codes into CNC machine from PC(NB).

When the source is preset, you can select one of the 3 sources (HMI, memory card or PC) by HMI and perform the control requirement.

To achieve the above G-code download requirements on different sources, DVP-PM00M provides 2 methods for downloading G-codes to CNC: PMGDL software and specific HMI. Before explaining the two download methods, we need to know the program download settings on DVP-PM00M.

11.1 DVP-PM00M Program Download with PEP setting

When programs are downloaded to DVP-PM, the DVP-PM programs without PEP protection will be cleared first then the new program will be downloaded to the program area without PEP protection as below [OX0 in (A), (B)]. Motion subroutines (G-codes) in DVP-PM are directly called by O100 main program or P subroutines which are called by O100 main program. Therefore, to prevent O100 or P subroutines from being cleared and making G-codes invalid, PEP settings are usually enabled on O100 and P subroutine to protect the constant programs.



11 G-code Application

With PEP protection as above, constant programs O100 and P are maintained in 20PM, and OX0 (G-codes) which needs to be constantly updated is downloaded in the general program area without PEP protection.

11.2 DVP-PM00M G-code Download Methods

In this section, we will introduce G-code download methods by applying PMGDL software or HMI.

1. Use PMGDL to download G-codes to DVP-PM
2. Use Delta B-Type HMI to download G-Codes to DVP-PM
3. Use Delta HMI and convert G-codes through D registers

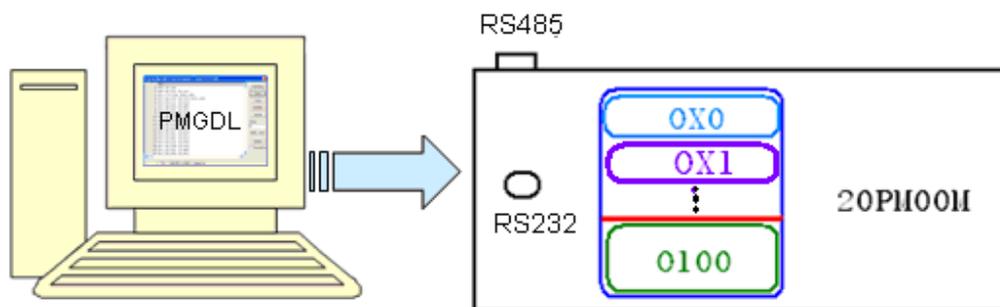
11.2.1 PMGDL Software

G-codes can be downloaded through PMGDL by 2 modes:

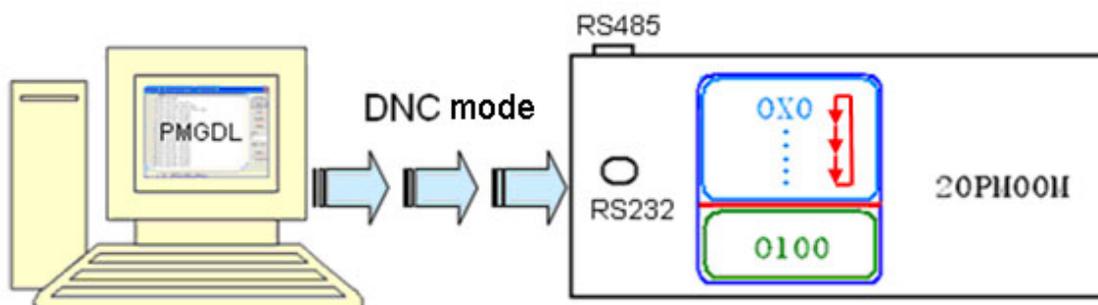
1. Common mode: Download G-codes to DVP-PM then execute OXn motion subroutines.
2. DNC mode: Execute OXn subroutines while file transmission is processing.

The application of Common mode and DNC mode is mainly decided according to the G-code file size. In Common mode, G-codes will be completely downloaded to PM before executing OXn motion subroutine. In DNC mode, generally applied for huge G-code files, transmission proceeds while motion subroutine executes. In diagram (A), PMGDL in Common mode downloads all G-codes into OX0, OX1 in 20PM00M. In diagram (B), PMGDL in DNC mode transmits G-codes while motion subroutines in 20PM00M is operating. Users should set up PEP protection for O100 main program before download O100 to 20PM00M. After this, executes O100 to set up COM1(RS232) or COM2(RS485) for communication with PMGDL, and PMGDL will download the G-codes.

Note: Please note that PMSOft supports ASCII mode only, and PMGDL software supports RTU mode only.



(A) Common mode G code download



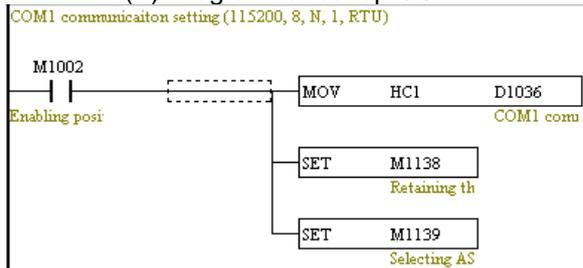
(B) DNC mode G code download

■ Common mode:

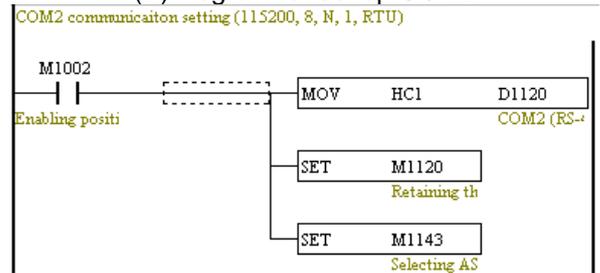
Step 1: Edit O100 main program. M-code (G-code) processing and COM port (COM1/COM2) should be set

up in O100 to communicate with PMGDL. Communication format: 115200, 8, N, 1(RTU).

(A) Program for set up COM1

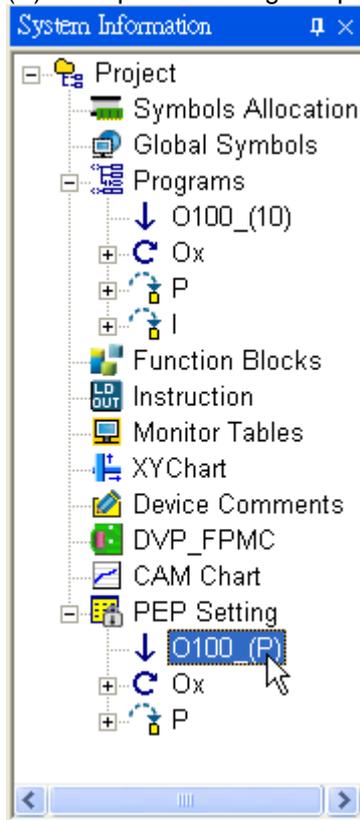


(B) Program for set up COM2

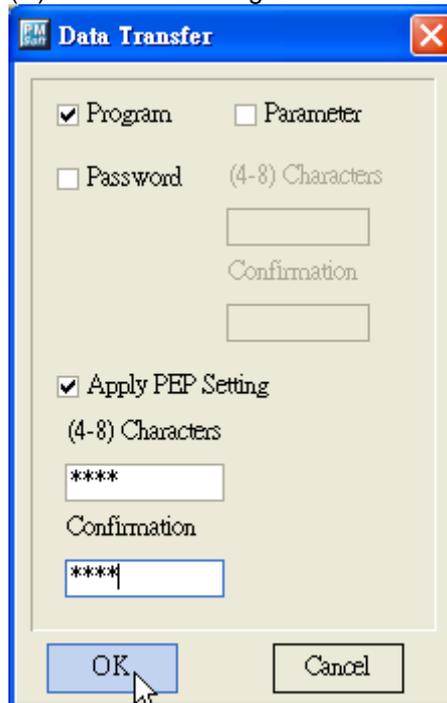


Step 2: Set up PEP settings in PMSOft to enable protection on O100 or other constant programs and download the program to 20PM00M as below.

(A) Set up PEP settings on programs



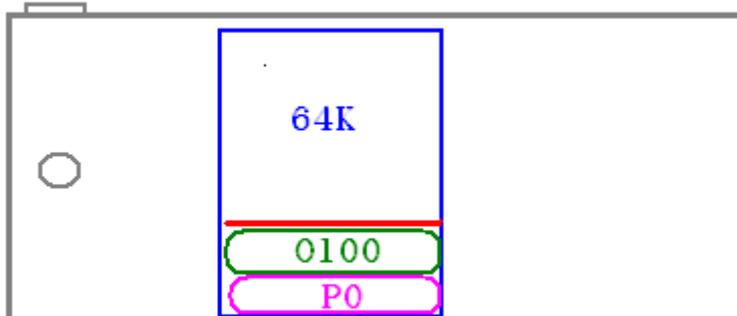
(C) Download settings



(B) Click Download Program

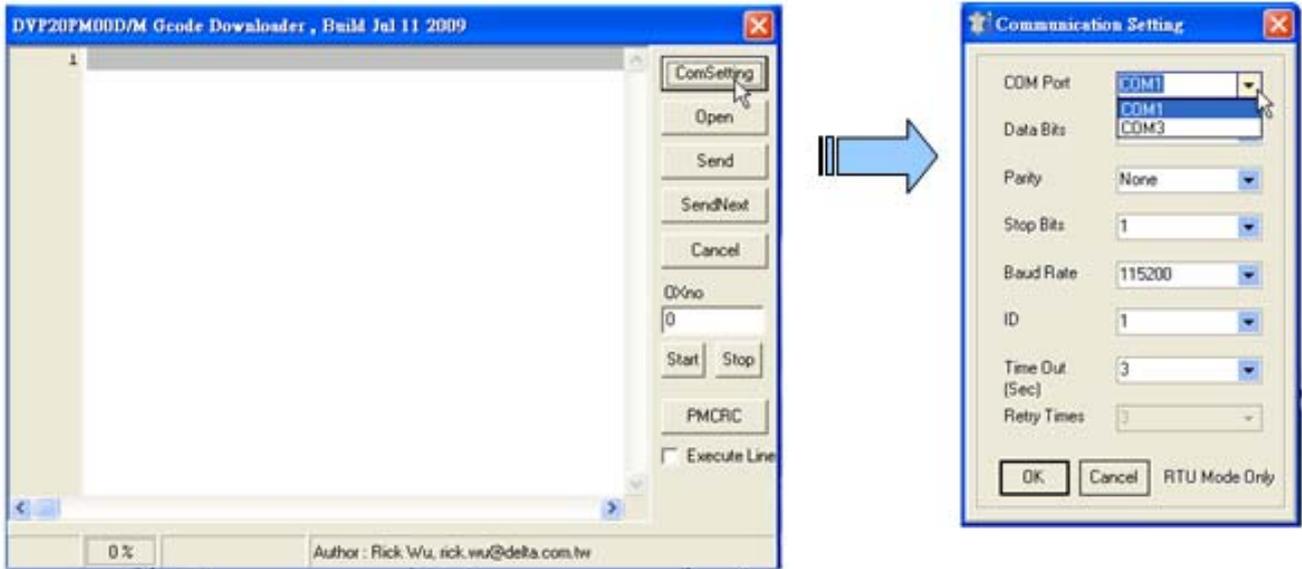


(D) Results



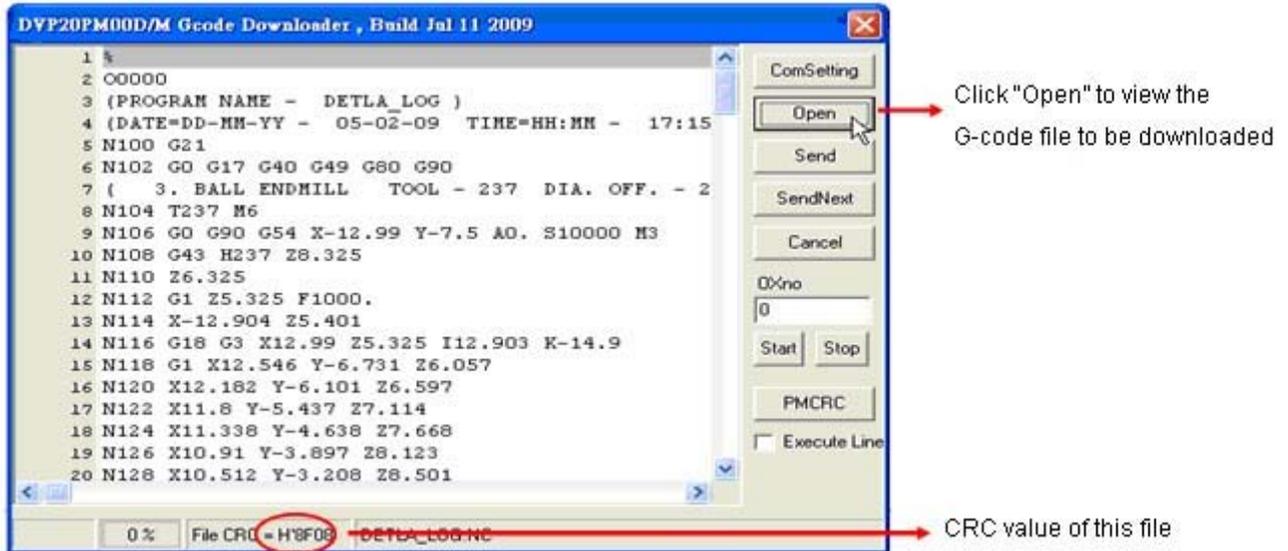
11 G-code Application

Step 3: Execute PMGDL software and set up COM Port (between PC and 20PM00M COM port) as below. Communication format 115200, 8, N, 1 should not be changed.

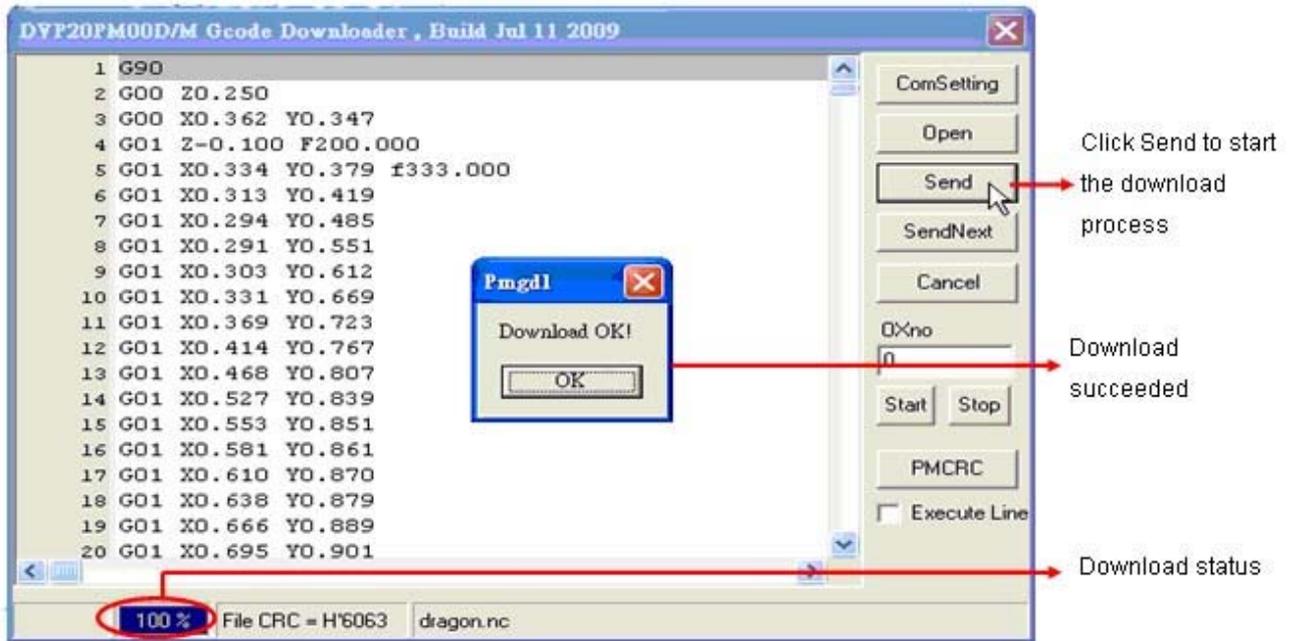


Step 4: Click "Open" to view the G-code file to be downloaded. In the bottom-left corner, File CRC will be displayed. Click "Send" to download the G-code file.

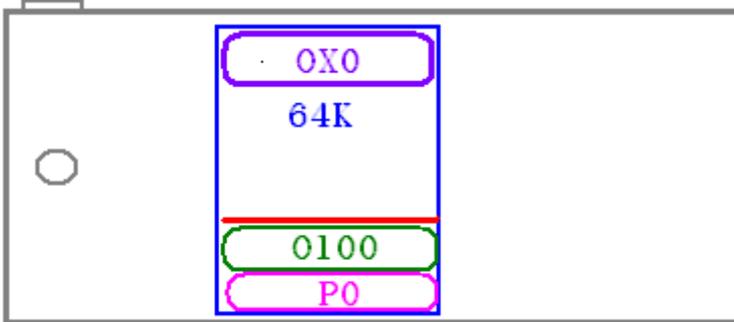
(1) Click "Open" to view the G-code file to be downloaded.



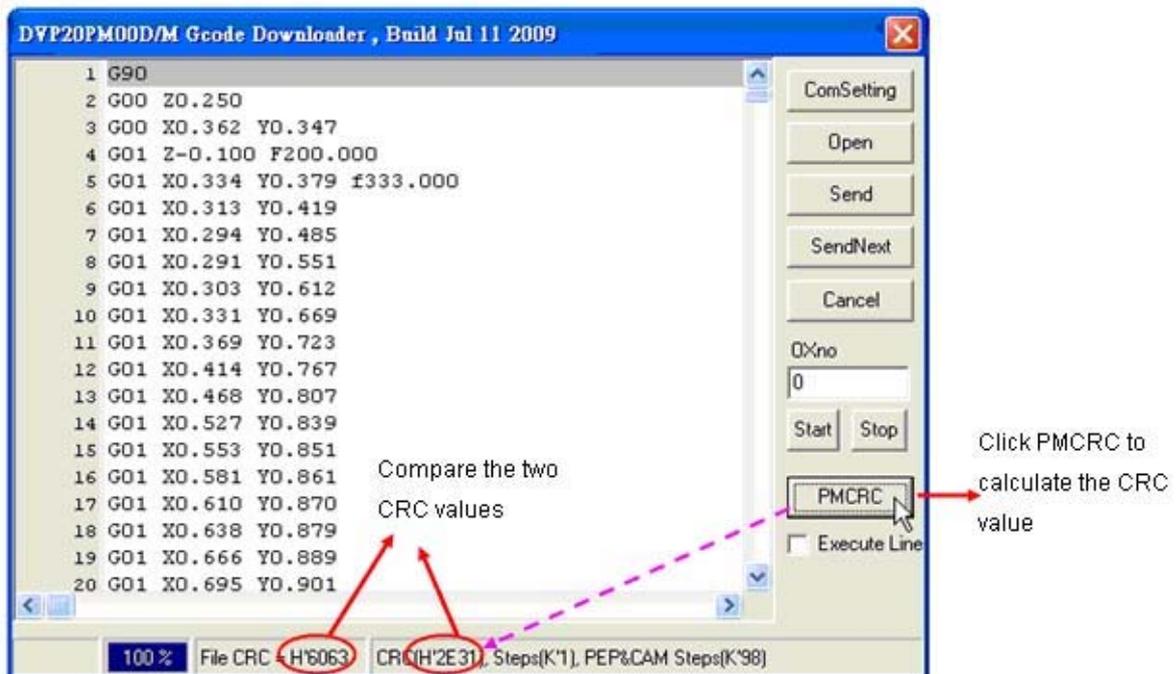
(2) Click "Send" to download the G-code file.



(3) Download completed

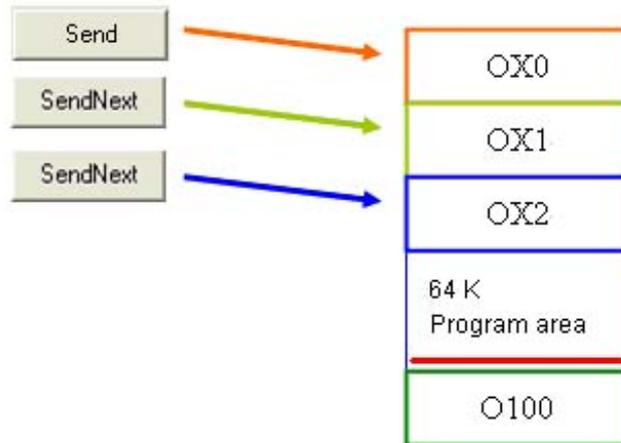


Step 5: You can click "PMCRC" to calculate the CRC value and compare the CRC value with File CRC. The download is succeeded if the two CRC values are the same. Step 5 can be skipped if "Download OK" message was showed in step 4.



11 G-code Application

Step 6: PMGDL can download multiple G-code files to DVP-PM00M. Click “Send” and the file will be downloaded to OX0. Click “Send Next”, and the file will be downloaded to OXn (n increases 1 at a time). If only OX0 is required, the step can be skipped.



PS: Next time when G-codes need to be updated, only steps after step 3 are required.

■ DNC mode:

DNC mode is applied for G-code file with over 5000 rows of instructions. In DNC mode, O100 do not need to be executed for calling OX0 motion subroutine. When DNC mode is select by PMGDL, DVP-PM will automatically execute OX0 when G-codes are downloaded to DVP-PM.

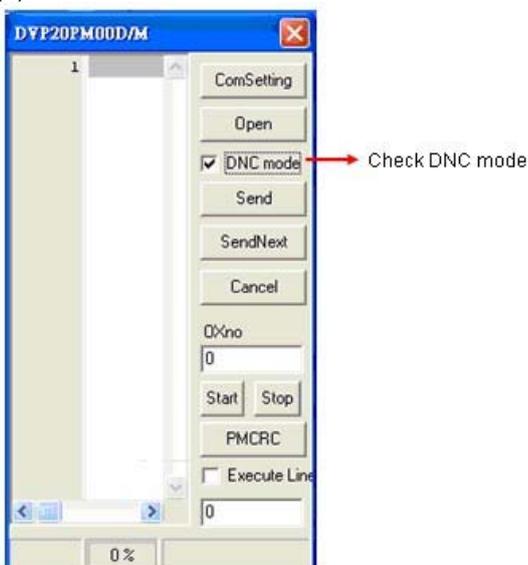
Step 1: The program downloaded to DVP-PM should contain OX0 motion subroutine, and G-codes do not need to be designed in OX0. To create OX0, click OX in System Information of PMSoft. Double click at 0 and OX0 editing window will pop up automatically. After OX0 is created, follow step 1 of Common mode to set up O100. ◦

Step 2: same as Step 2 of Common mode

Step 3: same as Step 3 of Common mode

Step 4: (1) check DNC mode, (2) Click “Open” to view the G-code file to be downloaded, (3)Click “Send” to download the G-code file. OX0 will automatically be enabled if download is succeeded.

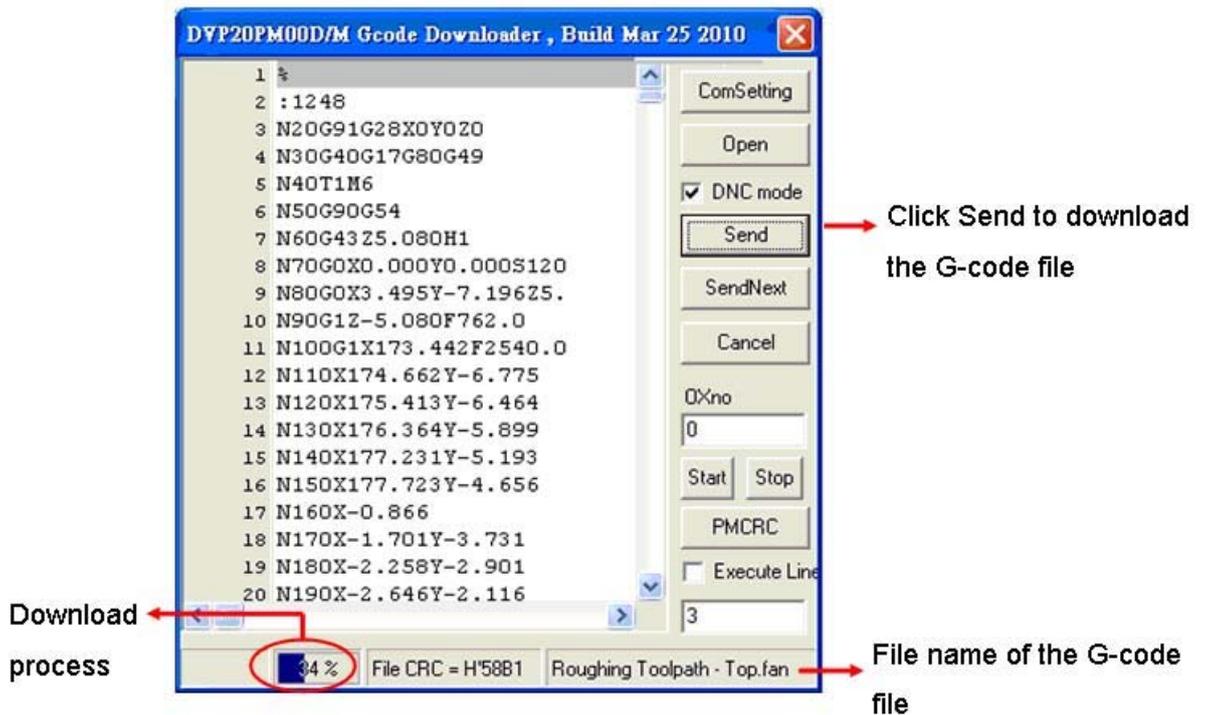
(1) Check DNC mode



(2) Click “Open” to view the G-code file to be downloaded



(3) Click "Send" to download the G-code file.

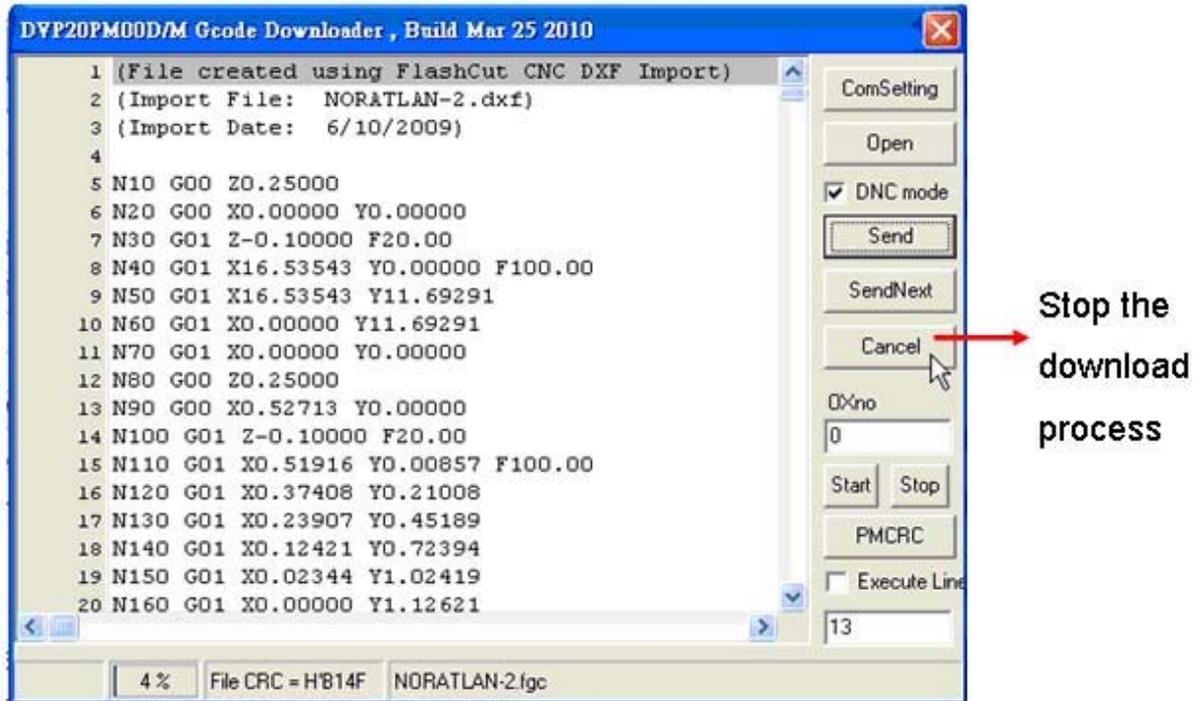


Step 5: On PMSoft, you can monitor D1733 for number of rows of received G-code on DVP-PM, and D1701 for currently executed rows of G-code. Check if the content of the registers increases and OX0 executes automatically as below.

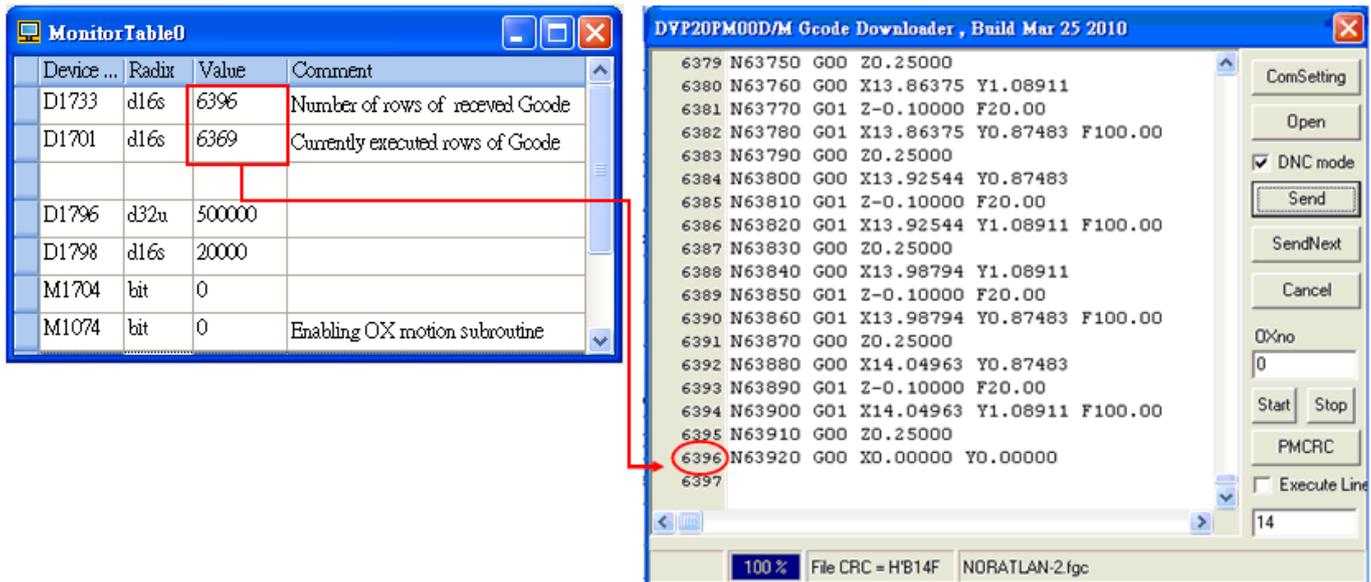
Device No.	Radix	Value	Comment
D1733	d16s	169	Number of rows of received Goode
D1701	d16s	39	Currently executed rows of Goode
D1796	d32u	500000	
D1798	d16s	20000	
M1704	bit	0	
M1074	bit	1	Enabling OX motion subroutine

Step 6: If you want to stop the operation in DNC mode, click "Cancel". G-codes will stop when the executed rows equal to the received rows. If you need to reset and download the file again, stop O100 and activate O100 again. The step can be skipped if there is no need for stopping the operation.

11 G-code Application



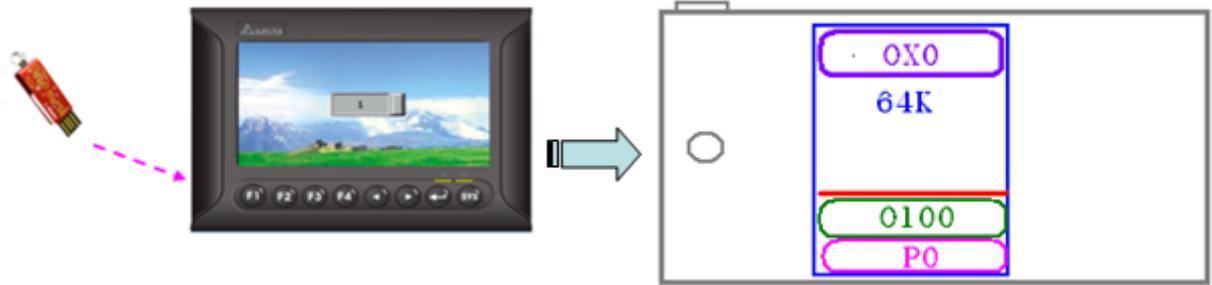
Step 7: When the G-code operation is completed, the executed rows will be the same as the received rows in monitor table. In addition, the number of received rows also equals to the number of rows displayed in PMGDL. Also, OX0 stops automatically when the operation is finished as below.



Step 8: If you want to execute other G-codes, repeat the steps from step 4.

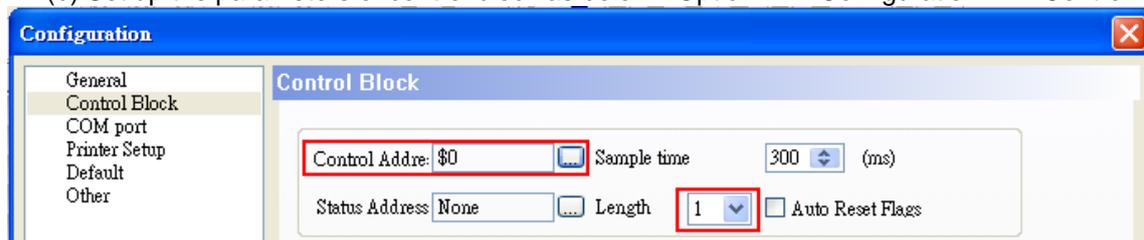
11.2.2 Download by B Type HMI

The below diagrams illustrate the steps to download G-codes to OX0 through HMI. Users should set up PEP protection for O100 main program (also includes P subroutines) before download O100 to 20PM00M. After this, executes O100 to set up COM1(RS232) or COM2(RS485) for communication with HMI, and HMI will download the G-codes to DVP-PM00M.



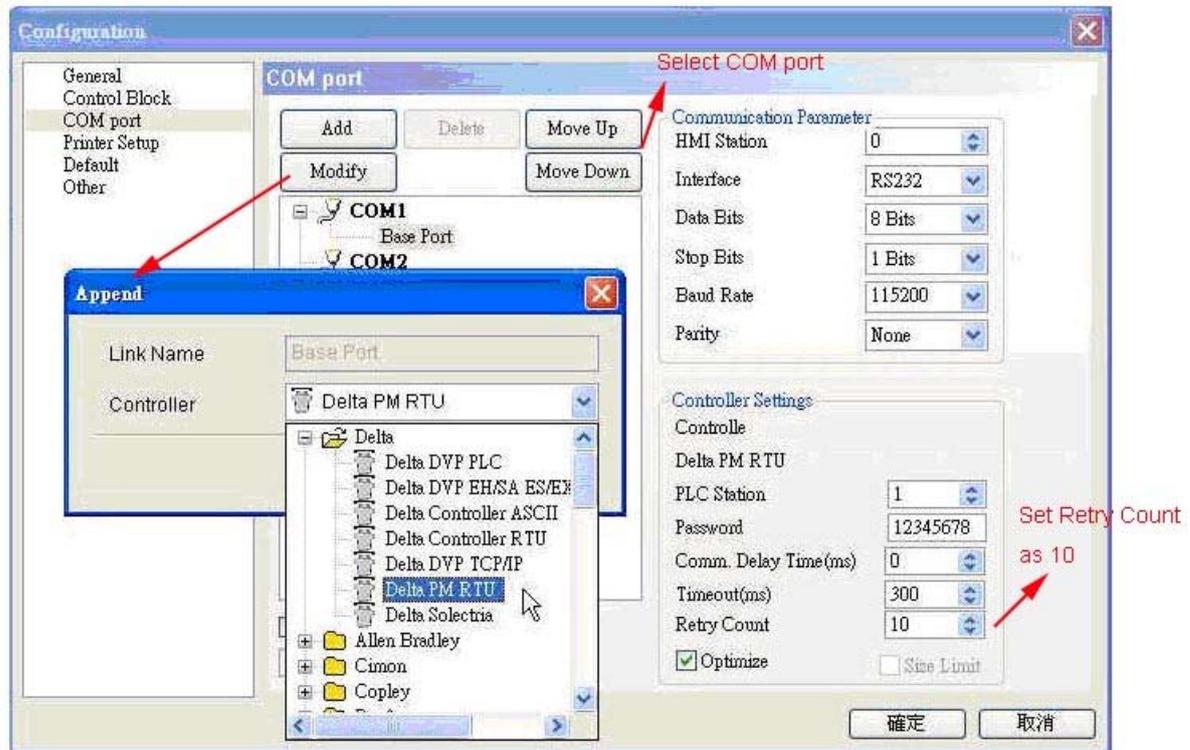
Step 1: set up an HMI supports G-code (NC file) download function.

- (1) HMI model: **DOP-BS0722P00**
- (2) Install the HMI programming software: Screen Editor 2.00.12
- (3) Execute Patch_DOPB_072909 to update the software version to Screen Editor 2.00.12
- (4) Edit HMI program:
 - (a) Execute Screen Editor 2.00.12, open a new file
 - (b) Set up the parameters of control block as below: "Option" >> "Configuration" >> "Control Block"



- (c) Set up communication parameters as below: "Option" >> "Configuration" >> "Communication"

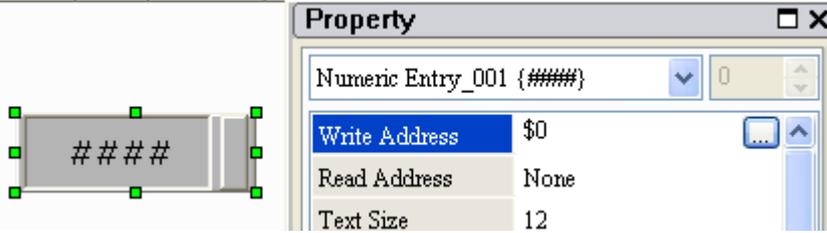
Controller: Delta PM RTU
 COM: COM1/COM2/COM3
 Interface: RS232/RS485
 Data Bits: 8(RTU)
 Stop Bits: 1
 Baud Rate: 9600/19200/38400/57600/115200
 Parity: None



- (5) Set the path to enter G-code download window: write 65535 into HMI internal memory address \$0 and the G-code download window will be displayed. There are 2 ways to set up the path:

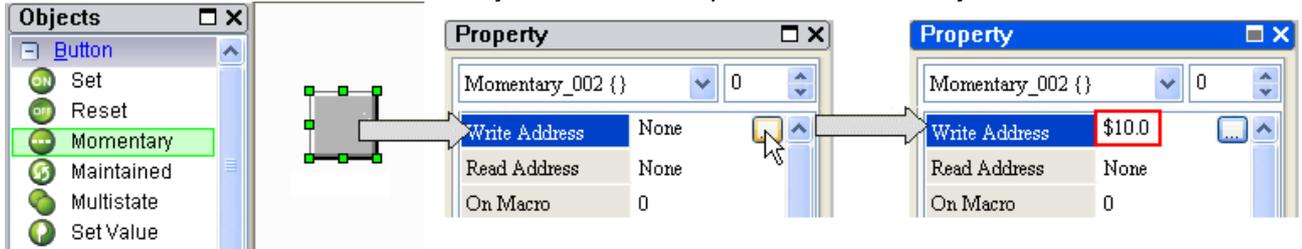
11 G-code Application

(a) Use a value input object, set up Write Address as \$0 and the address value as 65535.

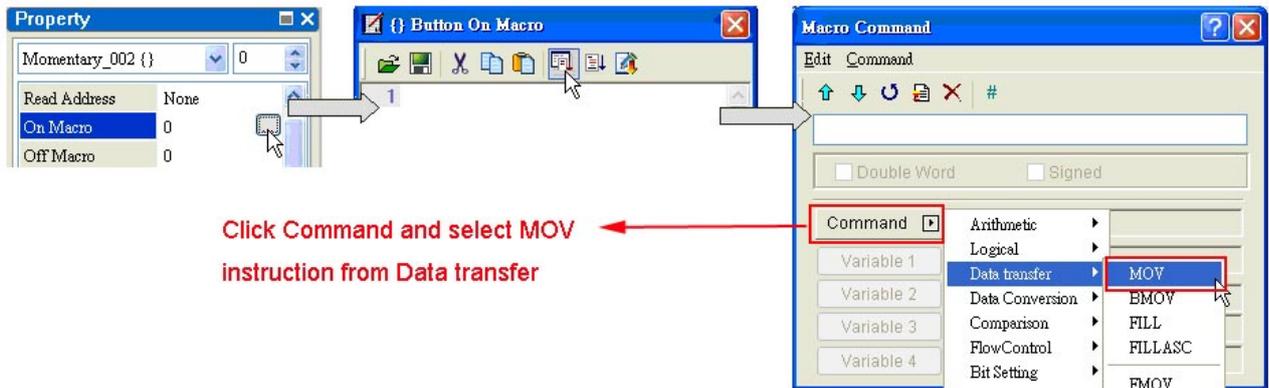


(b) Use a button object. Set up On Macro. When the button is pressed, 65535 will be written into \$0.

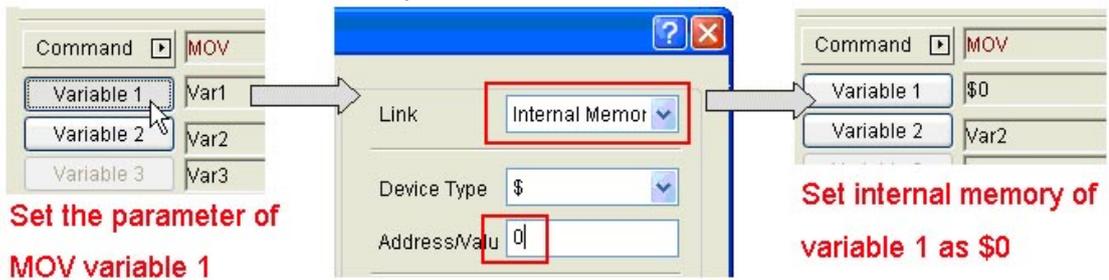
Select a Momentary Button and set up the internal memory address



Edit On Macro and use MOV instruction



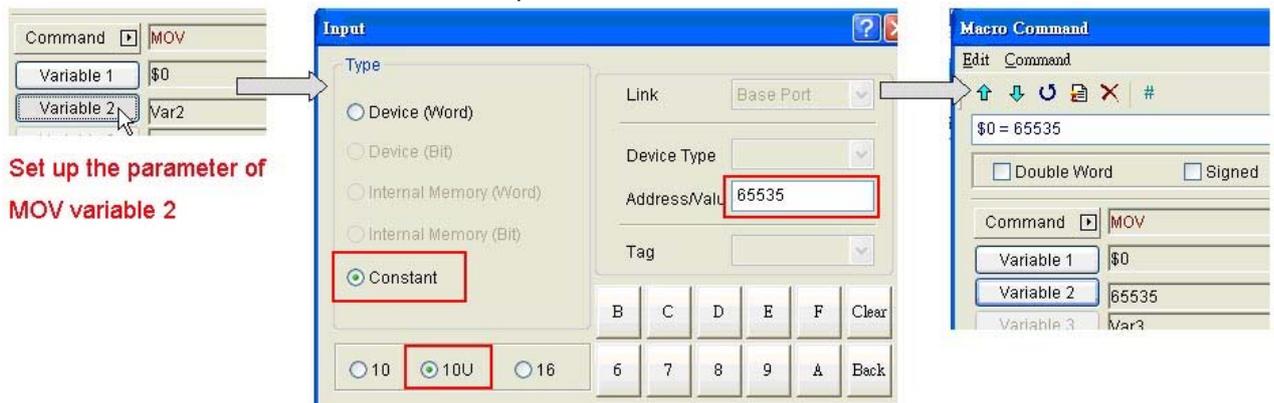
Set up variable 1 of MOV instruction



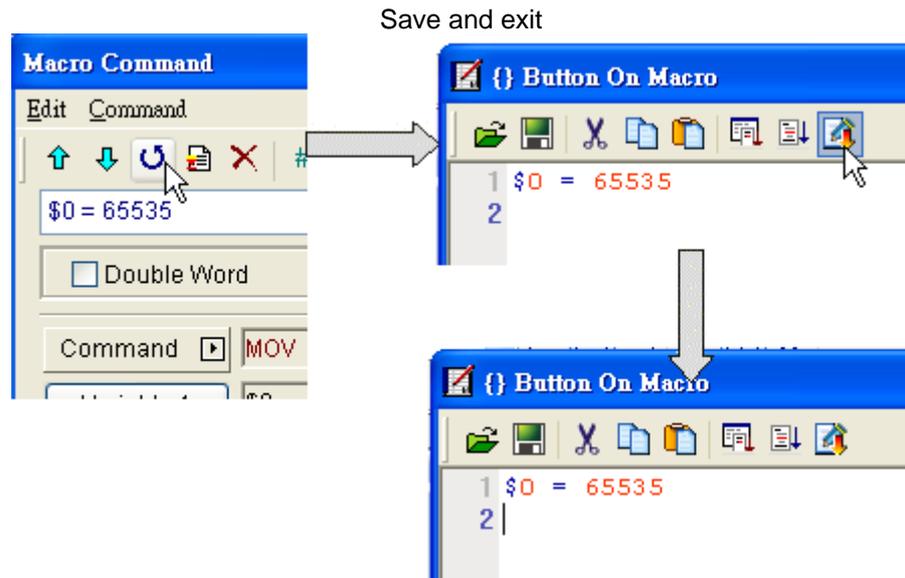
Set the parameter of MOV variable 1

Set internal memory of variable 1 as \$0

Set up variable 2 of MOV instruction



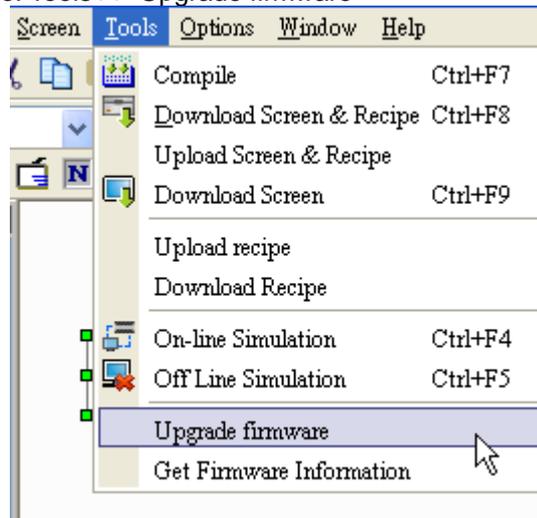
Set up the parameter of MOV variable 2



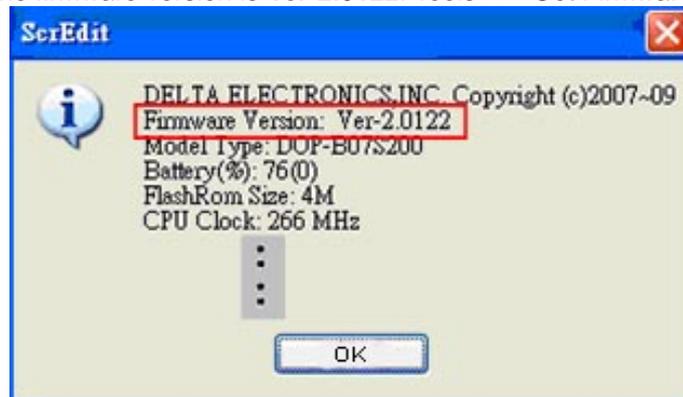
(6) Download the edited program to HMI

(7) Update the firmware version of HMI (can be skipped if updated):

(a) Upgrade the firmware: Tools >> "Upgrade firmware"

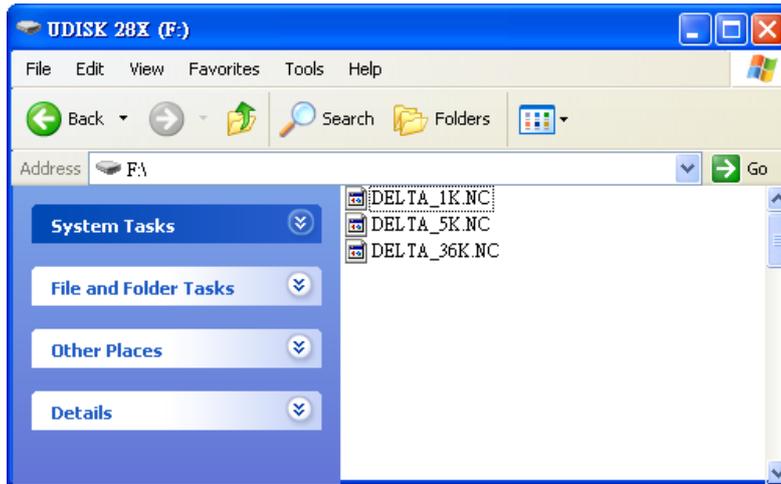


(b) Confirm that the firmware version is **Ver-2.0122**: Tools >> "Get Firmware Information"



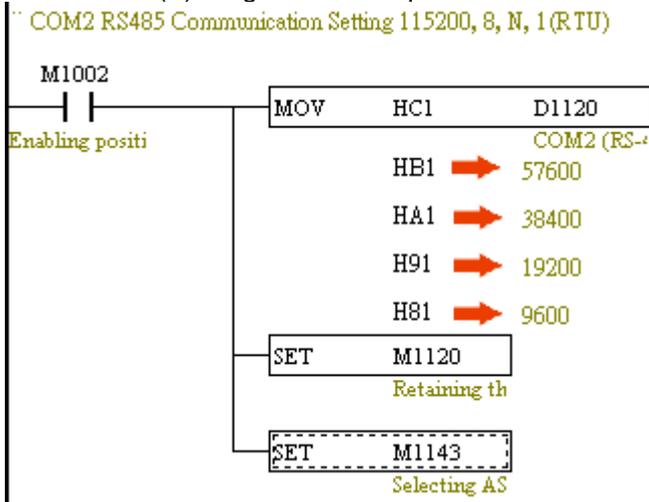
Step 2: Edit and copy the G-code file, and paste the file to a USB drive (The data format of USB should be FAT32). Insert the USB to the host USB port in the back of HMI.

11 G-code Application

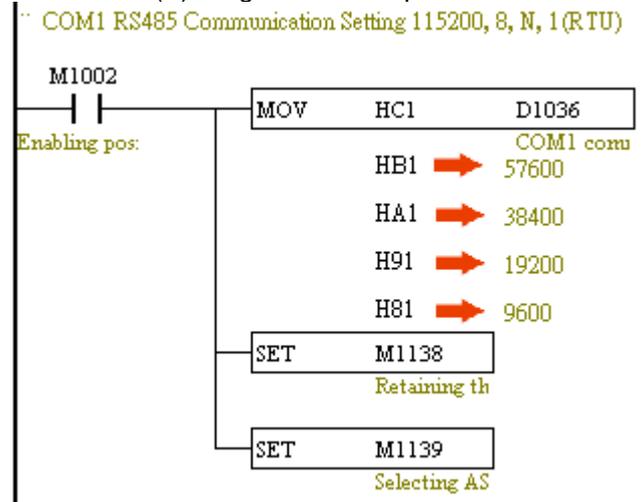


Step 3: Edit O100 main program. M-code (G-code) processing and COM port (COM1/COM2) should be set up in O100 to communicate with HMI. Communication format: 115200(57600/38400/19200/9600), 8, N, 1(RTU). The communication format here should match the format set up in HMI.

(A) Program for set up COM1



(B) Program for set up COM2



Step 4: Use PMSOft to enable PEP protection on O100 or other constant programs and download the program to DVP-PM. The set up steps are the same as Step 2 of PMGDL download method.

Step 5: connect DVP-20PM00M with B type HMI by communication cable.

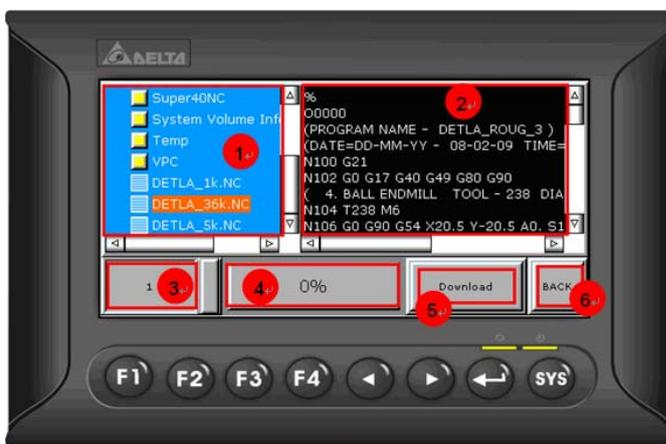
(a) Enter 65535 in the value input object on the screen of HMI.



(b) Press ENTER on the screen to enter the G-code download window.



Step 6: Enter the G-code download window. The functions of each block are explained as below. Select the G-code file to be downloaded in block (1). When download is finished, press BACK to return to previous page. If the file is too big, CRC value error will be displayed on the screen.



1	G-code file selection window: Select the G-code file to be downloaded in this window
2	G-code preview window: The selected G-code can be previewed in this window
3	Station number: Set up the station number of 20PM00M
4	Download: Click Download to start the G-code download process
5	Back: Click Back to return to previous page

Step 7: If you want to change the G-code file, repeat the process of step 5 and step 6.

11.2.3 Download by HMI other than B type through D registers

Users can also download G-codes by HMIs other than B-type. Set up the recipe on HMI and design the user program for receiving G-codes on DVP-PM00M. When both sides are ready, the G-codes can be downloaded through HMI directly.

The G-code download program designed in DVP-PM00M should operate in below steps: 1. initialize download process, 2. design the size of recipe according to the recipe format set up in HMI, 3. execute the program and the G-codes will be downloaded to designated registers in PLC according to the recipe set up in HMI. Please note that the recipe sets up the total number of G-codes but the whole G-code file requires several times of transmission for a complete download process. Therefore, in actual application the G-code file will be separated into many groups, and only one group will be transmitted by one time. File conversion will be executed every time when one group of G-code is received. If the conversion results are correct, the filtered G-codes will be displayed in the designated register. If an error in file conversion, error code 0xffff and the error row number will be displayed in designated registers

After one group are received, DVP-PM sends another value to change the group number and repeats the

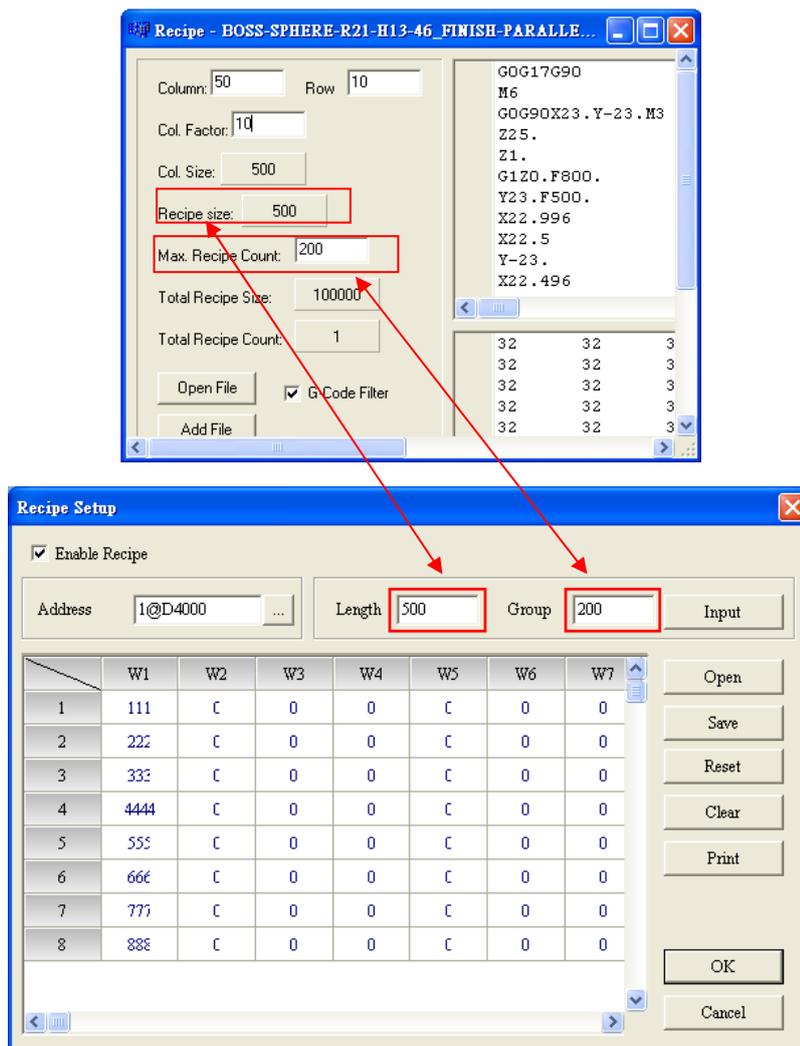
11 G-code Application

above process until all groups of G-code file are received, which is indicated by 20H(K32) in D3002, D3003. When all G-codes are downloaded, the conversion results will be checked again. Before the download begins, a blank motion subroutine should be created for storing the received G-codes. In addition, PEP protection should also be enabled on O100 main program, so that the original O100 on PC will not be cleared after uploading the program from DVP-PM.

Set up recipe on HMI:

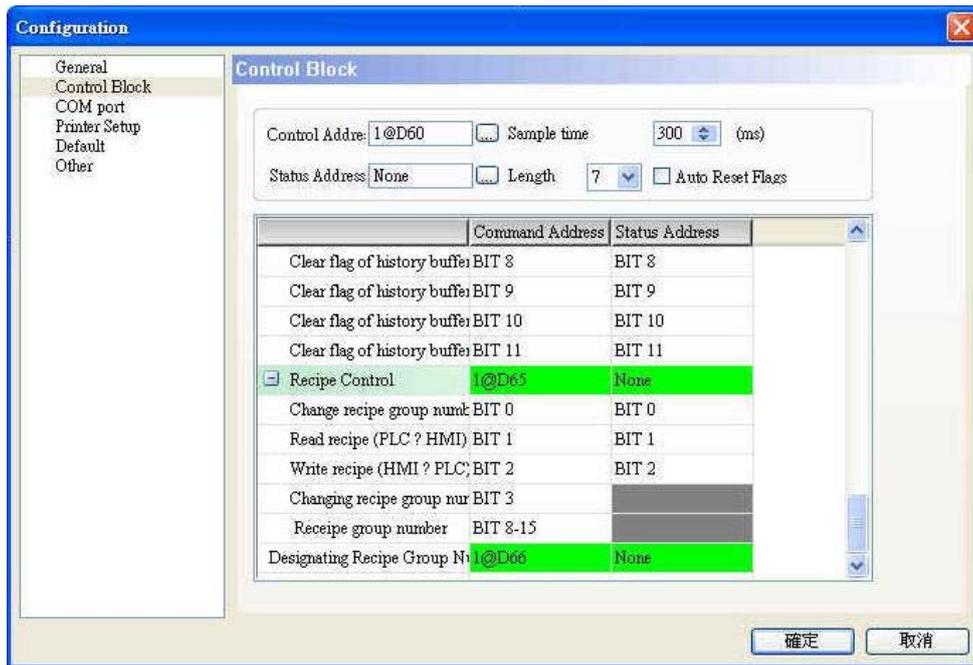
Users can convert the G-codes of CNC into recipe of HMI through the software "Recipe". Recipe refers to the data format for transferring G-codes, including Length (Recipe size) and Group (Max Recipe Count). After the recipe file is ready, store the file in memory card and insert the memory card in the slot on HMI. Configure "Recipe Setup" in Screen Editor according to the recipe settings in Recipe software and the G-codes can be downloaded when the program in DVP-PM00M executes.

Step 1: Set up the recipe in Screen Editor. The below diagrams indicates the recipe setting in both Recipe and Screen Editor. The format of Length and Group in Screen Editor is specified with 500 words and 200 groups corresponding to Recipe size and Max Recipe Count in Recipe software.



Step 2: Designate registers for commanding recipe and setting recipe group number in Screen Editor. The designated registers correspond to the registers on DVP-PM-00M. As the diagram below, DVP-PM sends recipe commands, such as read /write recipe, by sending values to the control register D65. Sending value

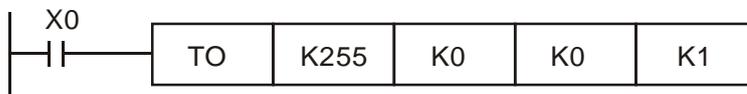
to D65 will change the recipe group number.



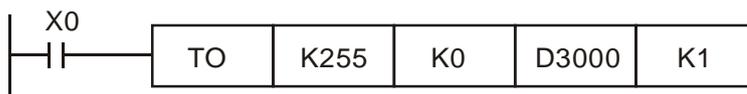
Design instructions in DVP-PM00M:

DTO/DFROM instructions are used for downloading G-codes. The applied instructions are listed as below:

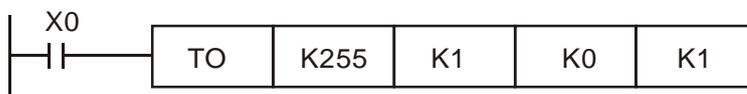
Initialization:



File conversion



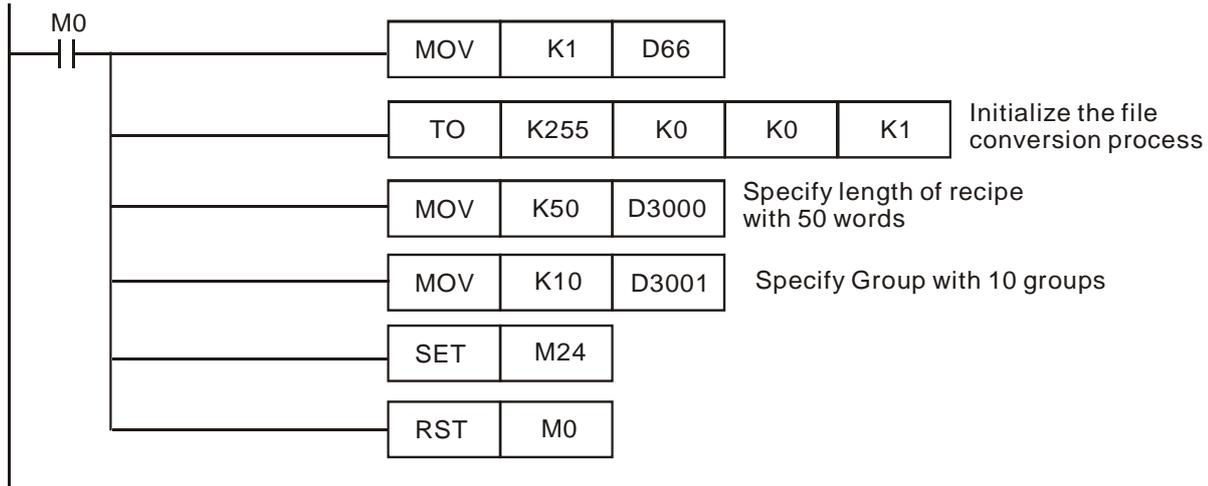
Check the completed conversion results



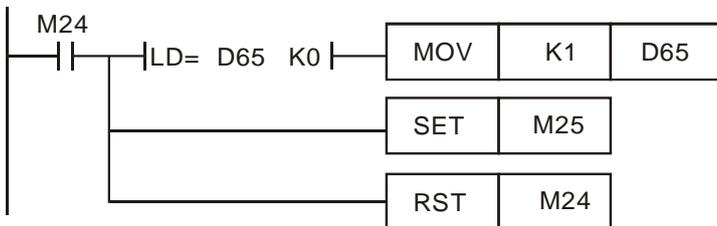
The below example explains the instructions for receiving G-codes. Specify length of recipe with 50 words and groups of recipe with 10. Execute file conversion and check the conversion results. If the conversion results are correct, the downloaded file will be stored in D3002~D3501. If errors occur, 0xffff will be recorded in D3000, and the number of error group (row) will be displayed in D3001. After uploading the program to PMSOFT, you can check OX0 motion subroutine for the received G-codes.

Step 1: Move K1 to D66 to designate the recipe group number. Initialize the file conversion process and setting the format of recipe according to the recipe setup in Screen Editor.

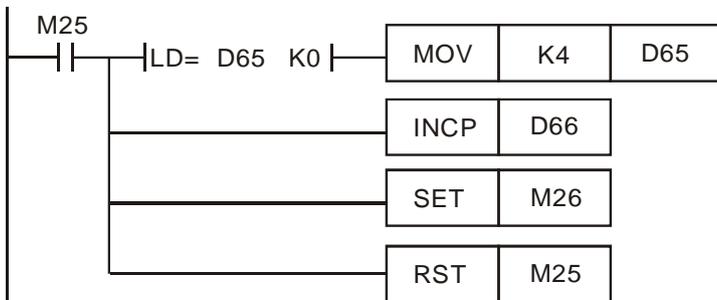
11 G-code Application



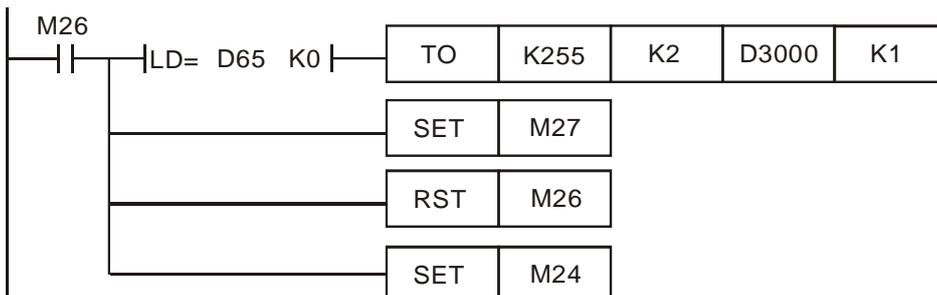
Step 2: HMI reads the value in D65 as the command to control the recipe. D65 = 1 indicates changing the recipe group number.



Step 3: When recipe group number is changed, the value in D65 becomes 0. After this, write in the recipe command to download next group and increase the recipe group number in D66.

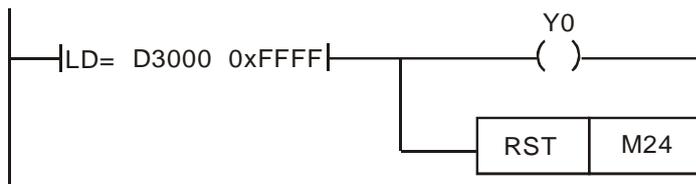


Step 4: The value in D65 will be cleared as 0 every time when modified. When one group of recipe is received, conduct file conversion to convert the received data into ASCII codes and store the converted data in D3002~D3502. When the above process is completed, repeat the step to download the next group.

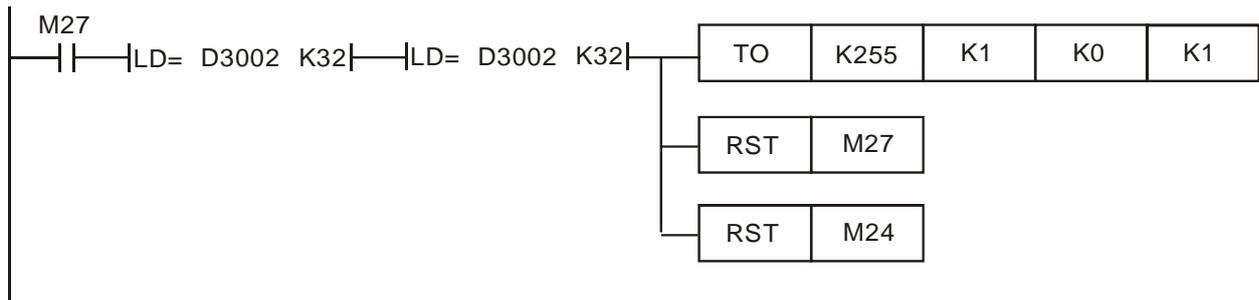


Step 5: If errors occur in file conversion, D3000 will store the error code 0xffff. Users can stop the next

download by monitoring D3000.



Step 6: When the values in D3002 and D3003 are both 20H (K32), the complete recipe download process are finished. Check the completed conversion results by To instruction as below.



11.3 G-code Application in Dispensers

For G-code application in automatic dispensers, DVP-PM00M provides a specific function which performs smooth movement with equal speed, avoiding pauses which lead to uneven dispensing results during operation.

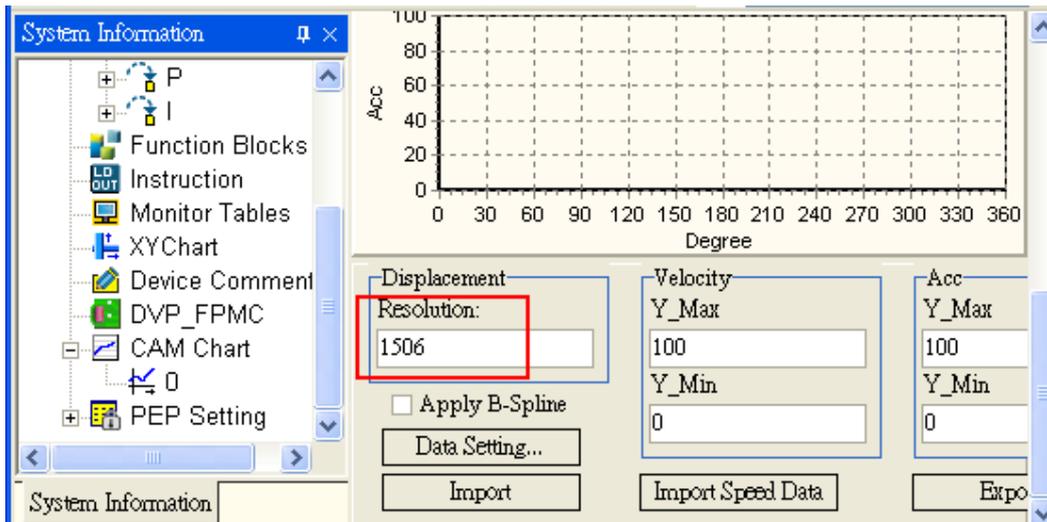
The function utilizes CAM charts for storing the G/M codes and applies E-CAM operation to execute the data in CAM charts. DVP-PM are designed with three CAM charts storing 3 sets of G/M code programs including OX0, OX1 and OX2. Max 2048 points are allowed for each CAM chart, and each CAM chart should accommodate displacement and speed data of 3 axes. In addition, only G00, G01, G04 and M codes applied in "With mode" are supported in this function. Therefore, the total available instruction capacity for G-codes (G00/G01/G04) and M-codes in single CAM chart is 680, excluding the beginning and ending instructions.

11.3.1 Operation steps

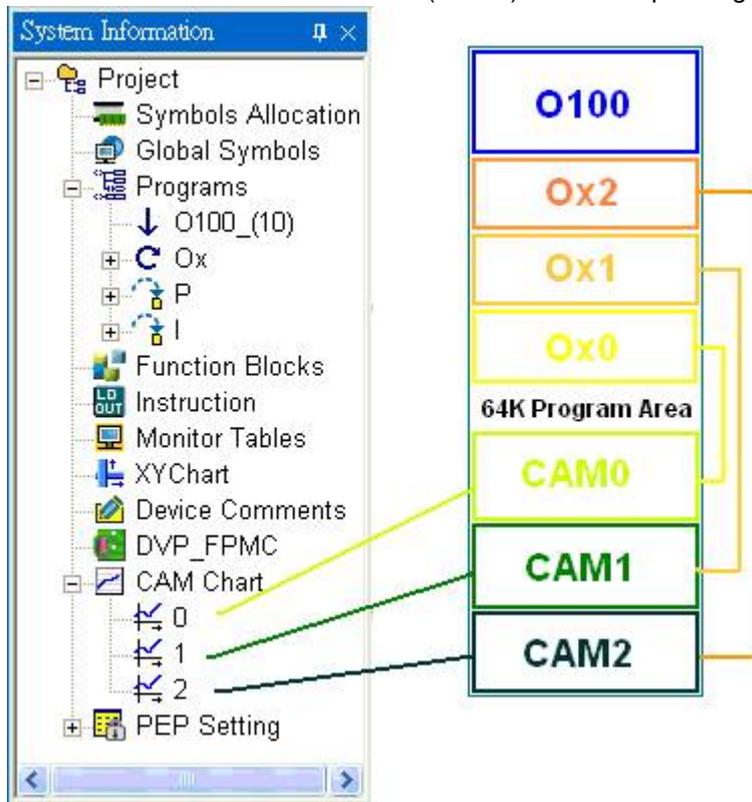
The 4 operation steps are 1. parameter setting on servo, 2. Create a blank CAM chart, 3. Load data into CAM chart and 4. Enable the dispensing operation.

1. Parameter setting on servo: if a Delta's servo is applied, set the parameter p1-08 as 5~8 according to the mechanism.
2. Create a blank CAM chart: create a blank CAM chart with proper resolution. For example, if OX0 for dispensing operation requires 500 points including G04 and M-codes, the resolution of CAM chart-0 should be set as 1506 as below. The same rule also applies on OX1 and OX2.

11 G-code Application



3. Load data into CAM chart: Load the G/M codes in OXn (n: 0~2) into corresponding CAM charts.



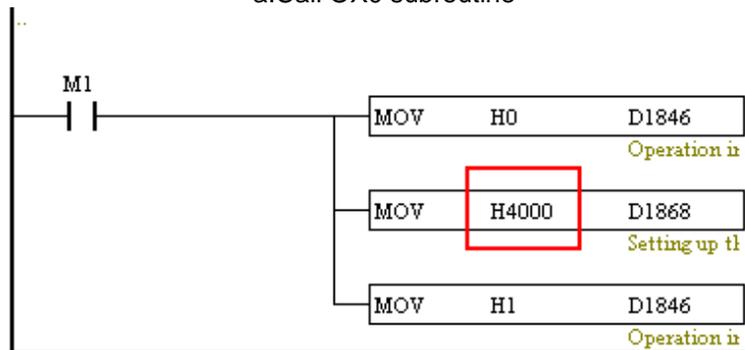
To load the data into CAM charts, work modes of 3 axes should be set up first then Call OXn.

- (1) Set up work modes of 3 axes:

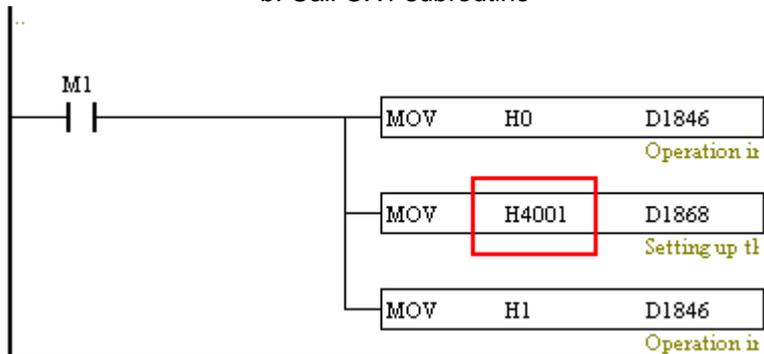
X axis	Y axis	Z axis	
D1847=H1000.	D1927=H1000.	D2007=H1000.	Work mode (bit12)

- (2) Call OXn: Call OXn to load the G/M codes in corresponding CAM charts, without executing OXn. When the G/M codes are loaded, M1792 will be rising-edge triggered. The below diagrams are examples demonstrating how to call OXn subroutines.

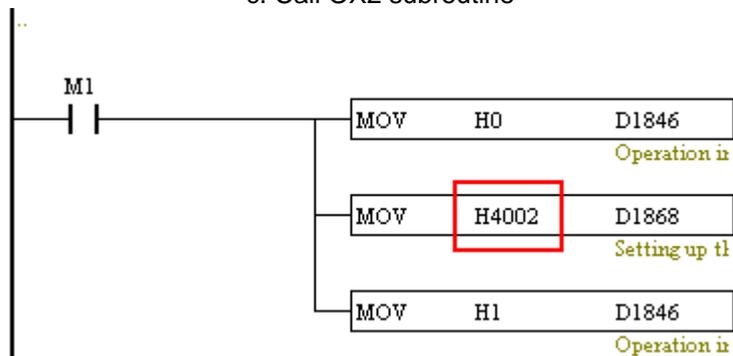
a. Call OX0 subroutine



b. Call OX1 subroutine



c. Call OX2 subroutine



Note: Do not set up M1036 (continuous interpolation) before this step, otherwise the loaded data will be G/M codes with continuous interpolation.

4. Enable the dispensing operation: Execute the G/M codes in CAM chart. Work modes of 3 axes should be set up first and select the CAM chart to be executed. Set D1846 = H2000 to enable cyclic E-CAM. When the operation in CAM chart is completed, M1793 will be rising-edge triggered.

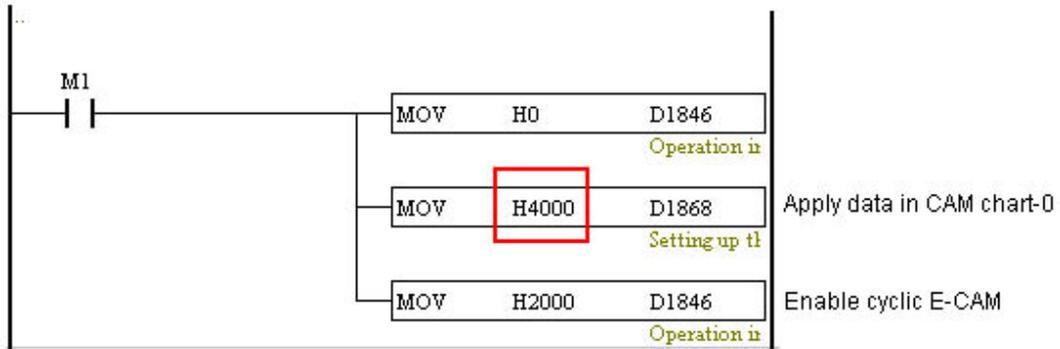
(1) Set up work modes of 3 axes: if the work modes are set up as H1000 already, skip this step.

X axis	Y axis	Z axis	
D1847=H1000	D1927=H1000	D2007=H1000	Work mode (bit12)

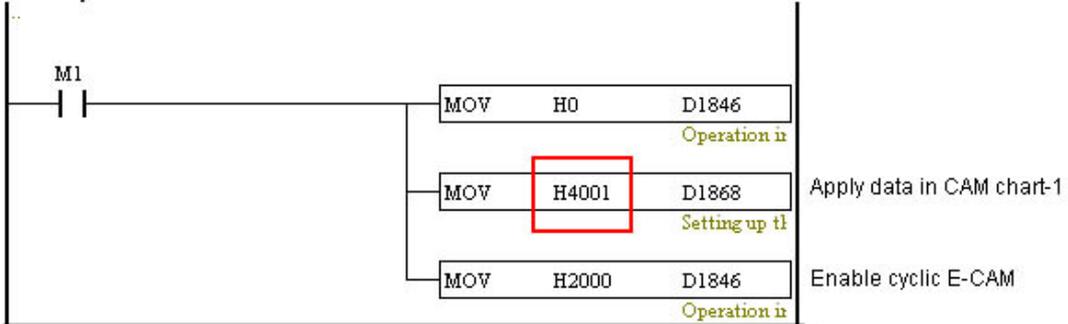
(2) Enable the dispensing operation by executing OXn subroutine.

11 G-code Application

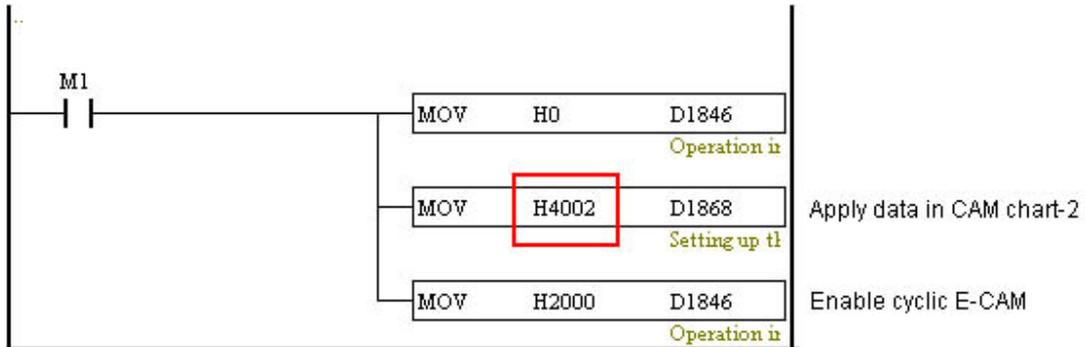
a. Execute OX0 subroutine



b Execute OX1 subroutine



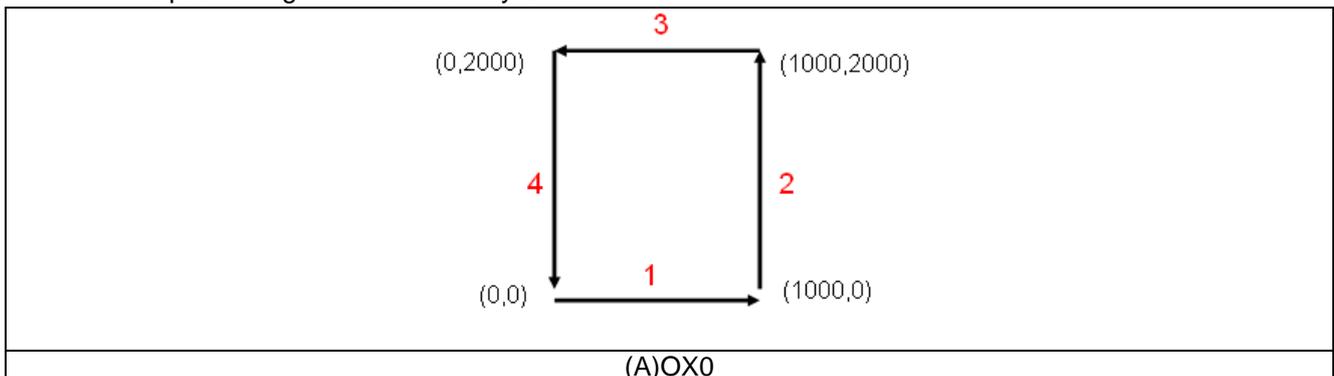
c. Execute OX2 subroutine



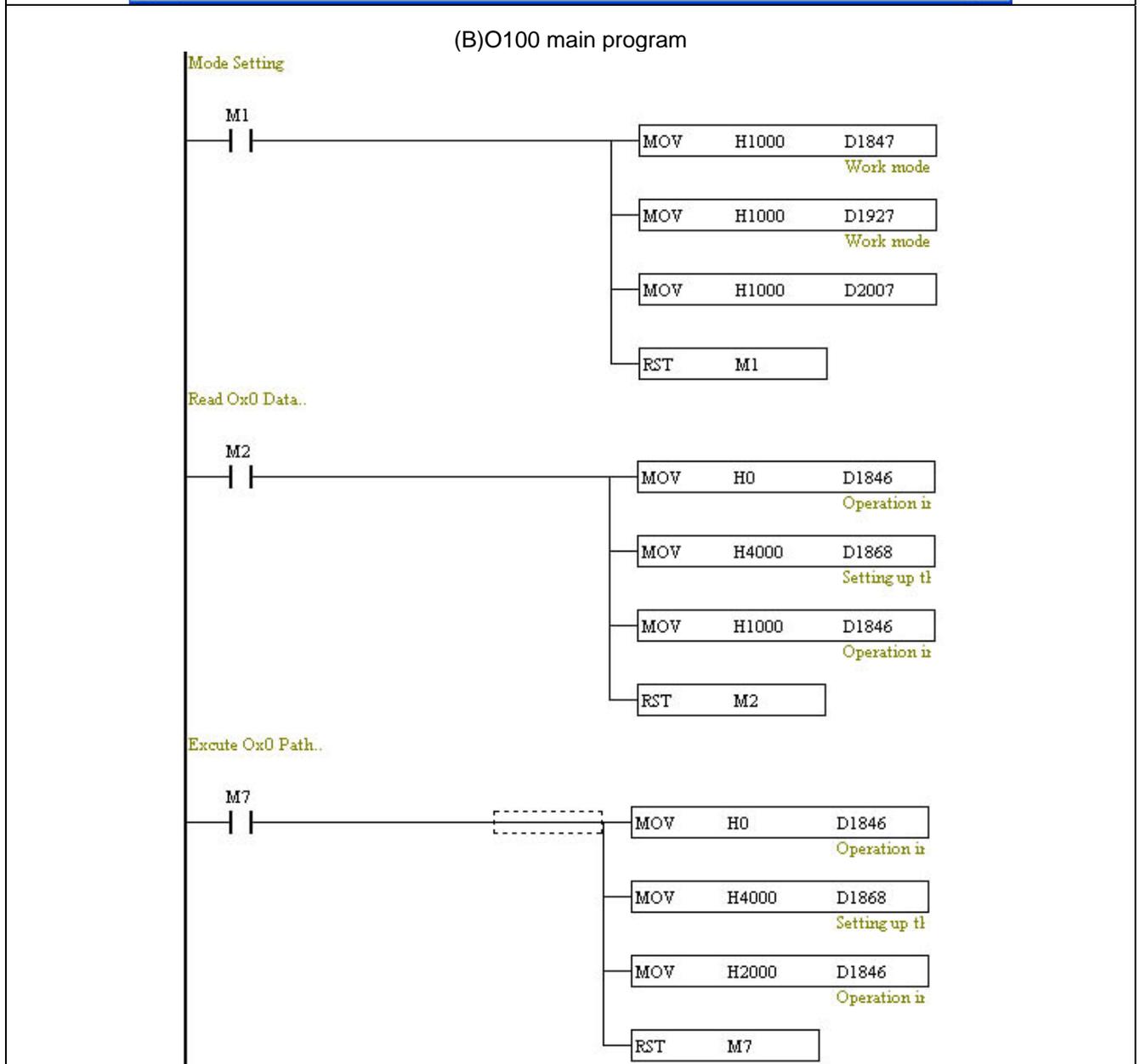
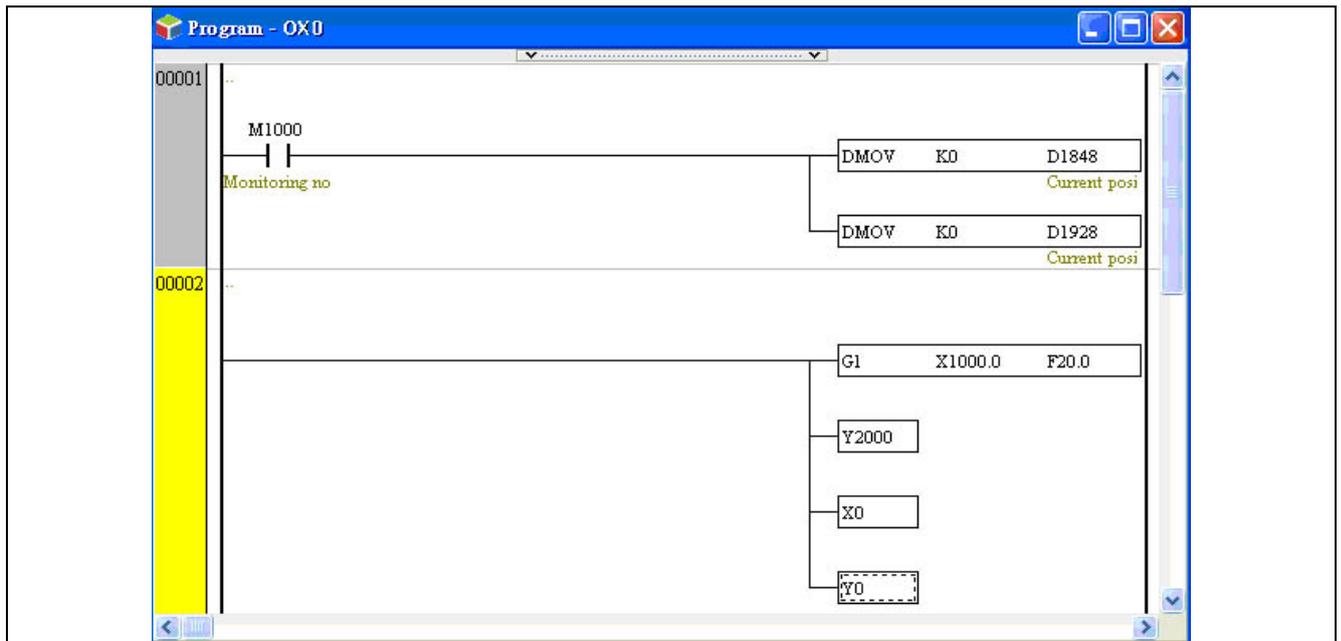
Note: When the function is not used, make sure the work mode is set as 0 (D1847 = 0, D1927 = 0 and D2007 = 0), otherwise errors will occur when executing single speed positioning.

Application example:

Perform the positioning route as below by OX0.



11 G-code Application



11 G-code Application

The program in O100 is listed as the diagram(B)

Step 1: M1 = ON, set up work modes of 3 axes.

- (1) D1847=H1000
- (2) D1927=H1000
- (3) D2007=H1000

Step 2: M2 = ON, load the data in OX0 into CAM chart-0:

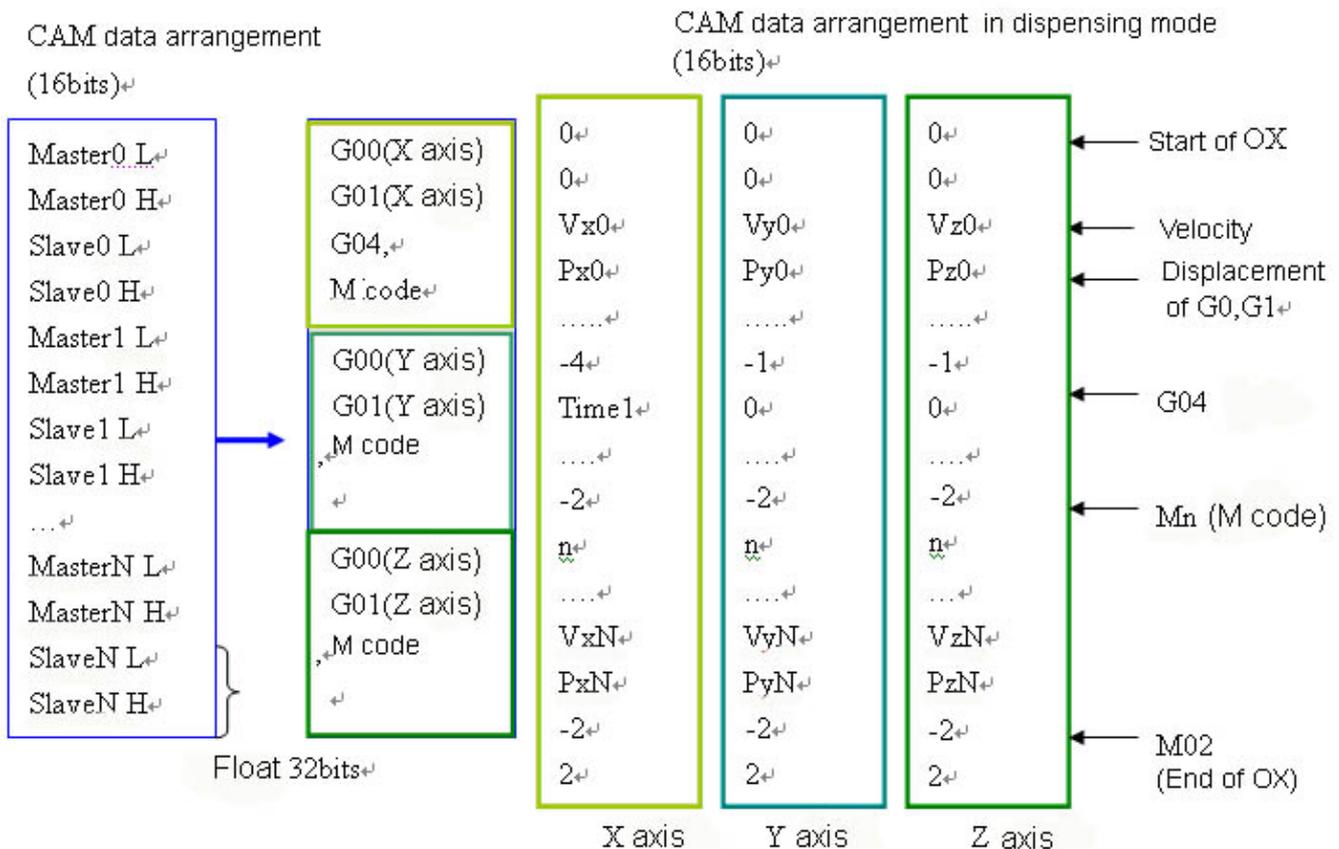
- (1) Select OX0 by setting D1868 = H4000.
- (2) Load the data in OX0 into CAM chart-0

Step 3: M7 = ON, execute OX0 (The data source is in CAM chart-0)

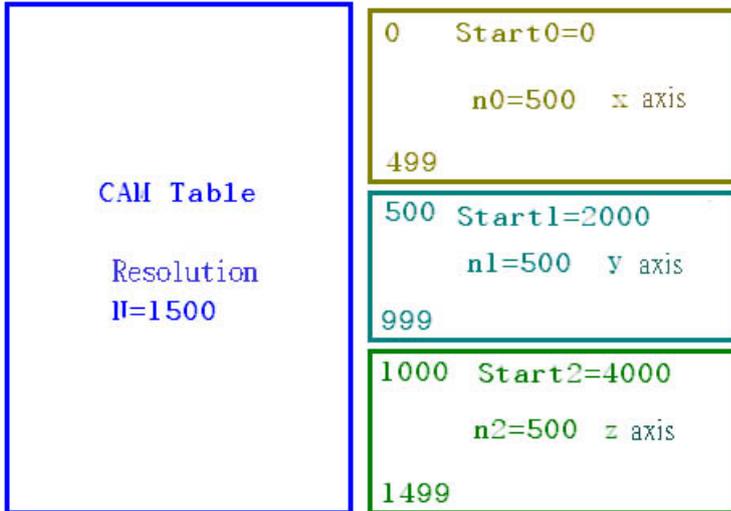
- (1) Select OX0 by setting D1868 = H4000.
- (2) Enable cyclic E-CAM

11.3.2 Storage of G/M code

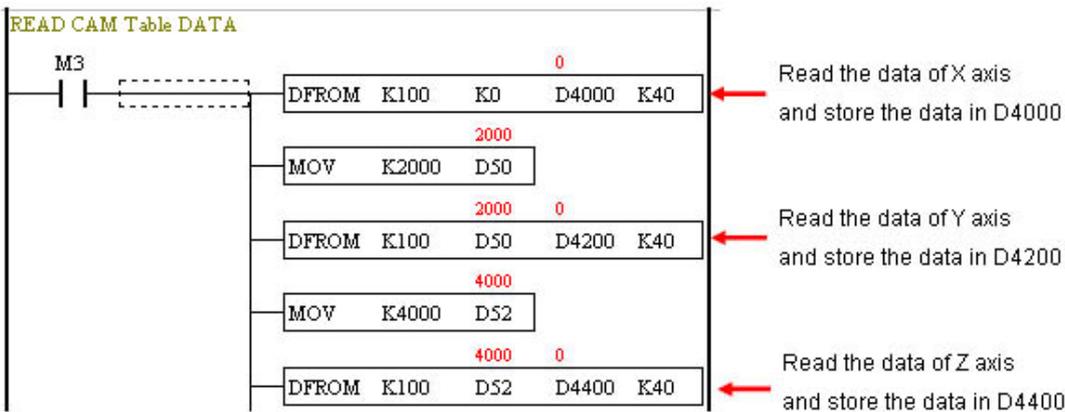
In this function, data in OXn is loaded to the corresponding CAM chart, and only G00, G01, G04 and M codes applied in "With mode" are supported for this function. The below diagram explains the data allocation in CAM chart.



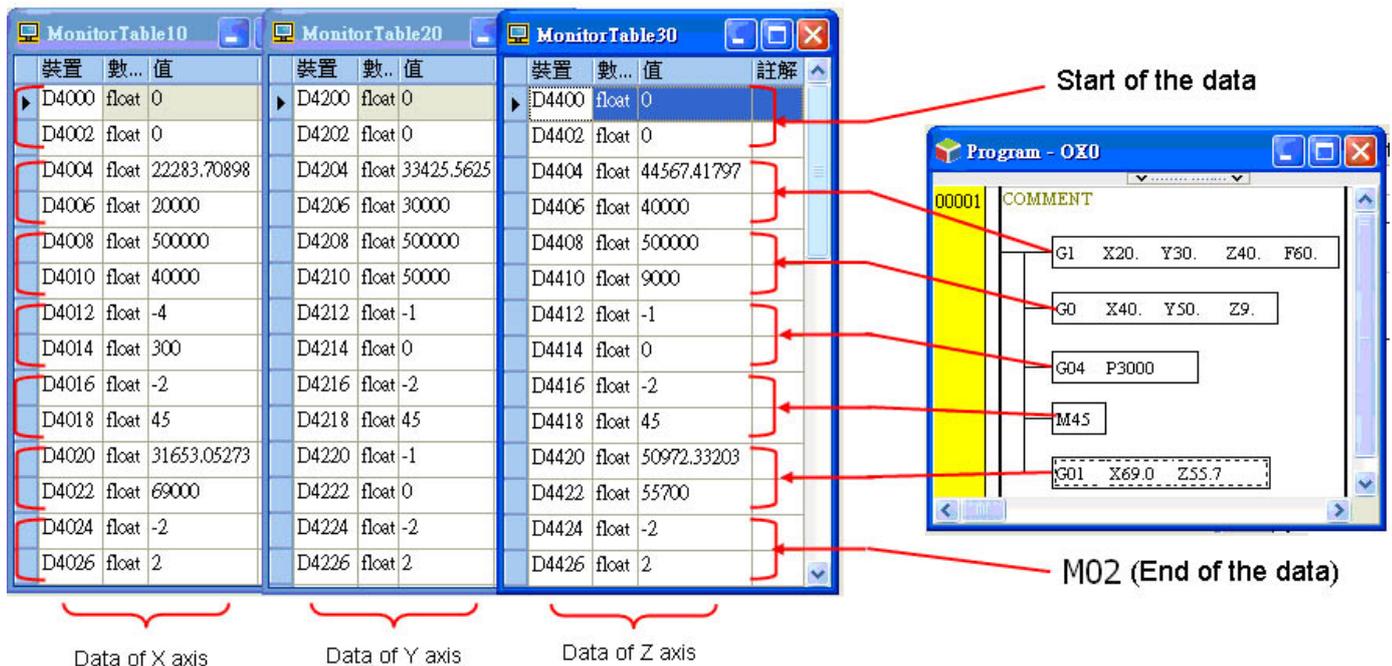
11 G-code Application



The example program uses CAM chart-0 with max resolution 1500. The below program are applied for reading the data of 3 axes in CAM chart.



Note: In the instruction DFROM (DTO) K100 K0 D4000 K40, set value K100 indicates access in CAM chart-0, set value K101 indicates access in CAM chart-1, set value K102 indicates access in CAM chart-2.



Explanations on the above table:

- (1) The first data of each axis is 0.

11 *G-code Application*

- (2) G01: The upper data is speed; the lower data is position. "-1" will be set if the speed data is not specified.
- (3) G00: Speed data is fixed as 500K. The lower data is position. "-1" will be set if the speed data is not specified.
- (4) Stop G00/G01: Write -1 into the speed data of G00/G01 on specific axis, and the instruction on the axis will be stopped.
- (5) G04: The value -4 in upper data indicates G04. The lower data is pause time, unit: 10ms.
- (6) M code: The value -2 indicates M code. The lower data is M code number.
- (7) End of the data: -2 and 2 indicates M02, which is the end of the CAM data.
- (8) G04 can only be placed in X axis.
- (9) M code instruction should be placed in all axes.

12 POU Editing Mode

PMSoft version released after 1.03V (1.03V not included) supports POU (Program Organization Unit) editing mode, which is also applicable to the program edited by older versions. POU editing mode can only be operated under ladder editing environment and mainly performs two functions as below:

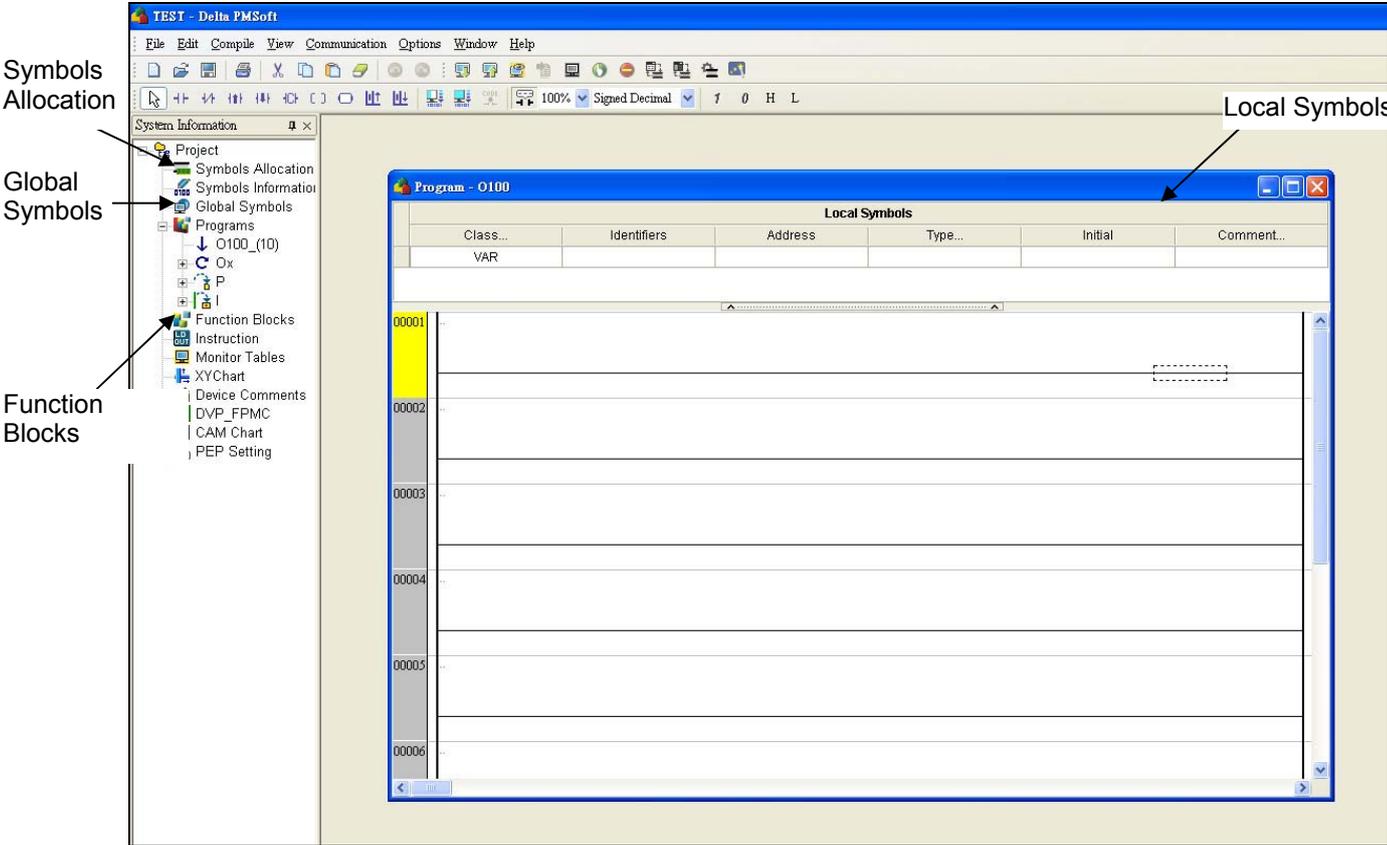
Function 1: User-defined function blocks (FB). FB can be used for O100 main program, OXn motion subroutines, Pn subroutines, and FB subroutines. PMSoft can import or export FB subroutines.

Function 2: Devices in ladder diagram can be replaced by symbols. Symbols in ladder diagram or FB can be allocated with devices by users or by system.

Functions of POU help users on simplifying programs and modulizing program structure, so as to save the time spent on developing program. In this chapter, we will introduce POU ladder editing environment, POU ladder editing methods, POU online monitoring and POU Hint function.

12.1 POU Ladder Editing Environment

When PMSoft creates a new project or opens an old project, it enters ladder editing environment. POU functions including Local Symbols, Global Symbols, Symbols Allocation and Function Blocks will be listed under System Information as below.



12 POU Editing Mode

12.1.1 Function of Symbols

Users can declare and define the symbols by local symbol table and global symbol table. Local symbols are applicable for the current editing window and global symbols are applicable for all POU in the program.

Local Symbols:

POU function operates in both Program and Function Blocks. Therefore, local symbols exist in both editing windows of program (O100, Pn subroutine, OXn subroutine) and function blocks. However, usage of local symbols differs in the two editing windows and is explained below.

1. Local symbols in Program:

(1) Only VAR is available for Class and the setting is only for internal use of current program.

Local Symbols					
Class...	Identifiers	Address	Type...	Initial	Comment...
VAR	test1		SET_FB		
VAR	result		BOOL	FALSE	
VAR	BIT1		BOOL	FALSE	

(2) All columns of Address should be filled.

2. Local symbols in Function Blocks

(1) VAR, INPUT, OUTPUT and INOUT are available for the column of Class.

VAR: available only in current function block.

INPUT: stores the input data of current function block

OUTPUT: stores the output data of current function block

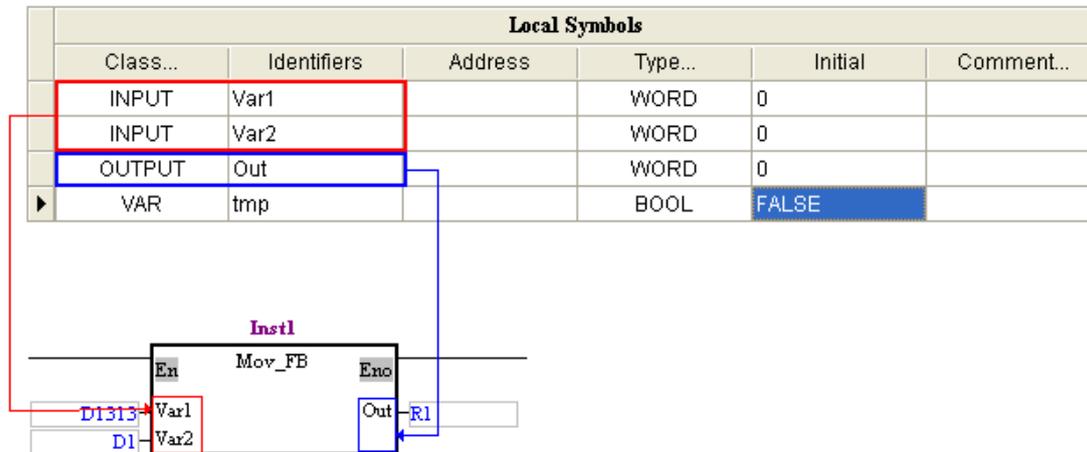
INOUT: operates the input data and output the data after the calculation.

Local Symbols					
Class...	Identifiers	Address	Type...	Initial	Comment...
VAR	TEST_1		BOOL	FALSE	
INPUT	TEST_2		BOOL	FALSE	
OUTPUT	TEST_3		BOOL	FALSE	
INOUT	TEST_4		BOOL	FALSE	

(2) The column of Address can be blank.

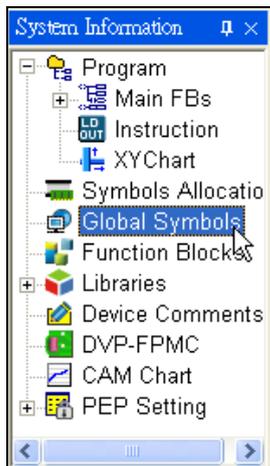
(3) When INPUT or OUTPUT is specified as the Class of symbols, the function block will utilize the symbols as the input or output interface as the diagram below.

12 POU Editing Mode



Global Symbols:

Unlike local symbols which exist in both program and function blocks, there is only one group of global symbols applicable for all POU. Double click the icon  under system information as Diagram (A), and the Global Symbol Table will pop up as Diagram (B).



(A) System Information

Global Symbols				
Identifiers	Address	Type...	Initial	Comment...

(B) Global Symbol Table

Rules for Naming Identifiers of Symbols (in circled columns):

Local Symbol					
Class...	Identifiers	Address	Type...	Initial	Comment...
VAR			DWORD	0	
VAR	OUTDATA1		COUNTER[4]		

1. Symbols such as "~ , ! , @ , # , \$, % , ^ , & , * , (,) , " are not allowed.
2. Case-insensitive.
3. Must be consisted of English letters, digits or underlines.
4. Two consecutive underlines are not permitted.

12 POU Editing Mode

5. The last character cannot be an underline.
6. Max length is 20 characters.
7. Spaces are not allowed.
8. Identifiers cannot be the same as device names.
9. Identifiers cannot be constants.
10. Identifiers contain decimal values starting with "DD" are illegal.

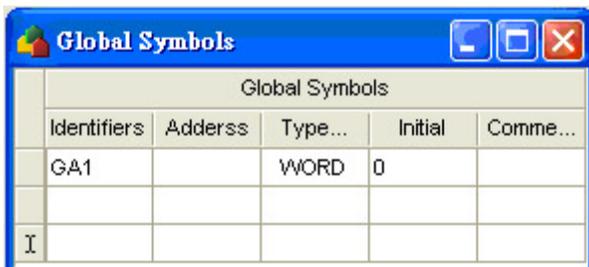
■ Rules for Using Symbol Table:

1. Identifiers, Type and Initial of the same symbol should be specified in coordination with each other.
2. Identifiers, Type, and Initial of the same symbol can not be set up individually.
3. Symbols can not be repeated in the same symbol table.
4. If symbols are specified with devices (Address), the specified devices should be in valid range.
5. G-code instructions do not support the application of symbols.
6. Symbols of index (V, Z) in local symbol table should be specified within the range of V0~V7 or Z0~Z7 as below.
7. Index V (V0~V7) should be specified with WORD in Type column as below.
8. Index Z (Z0~Z7) should be specified with DWORD in Type column as below.

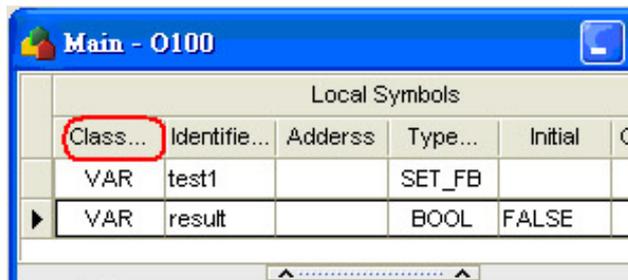
Class...	Identifiers	Addresss	Type...	Initial	Comment...
VAR	shift_v	V0	WORD	0	
VAR	shift_z	Z7	DWORD	0	

12.1.2 Editing Symbol Tables

There are 2 types of symbol tables in PMSoft editing environment: Global Symbol Table and Local Symbol Table. The content and editing methods of the two tables are basically the same except that Local symbol table provides the column of Class. The editing methods of symbol tables are explained as below.



Global Symbol Table



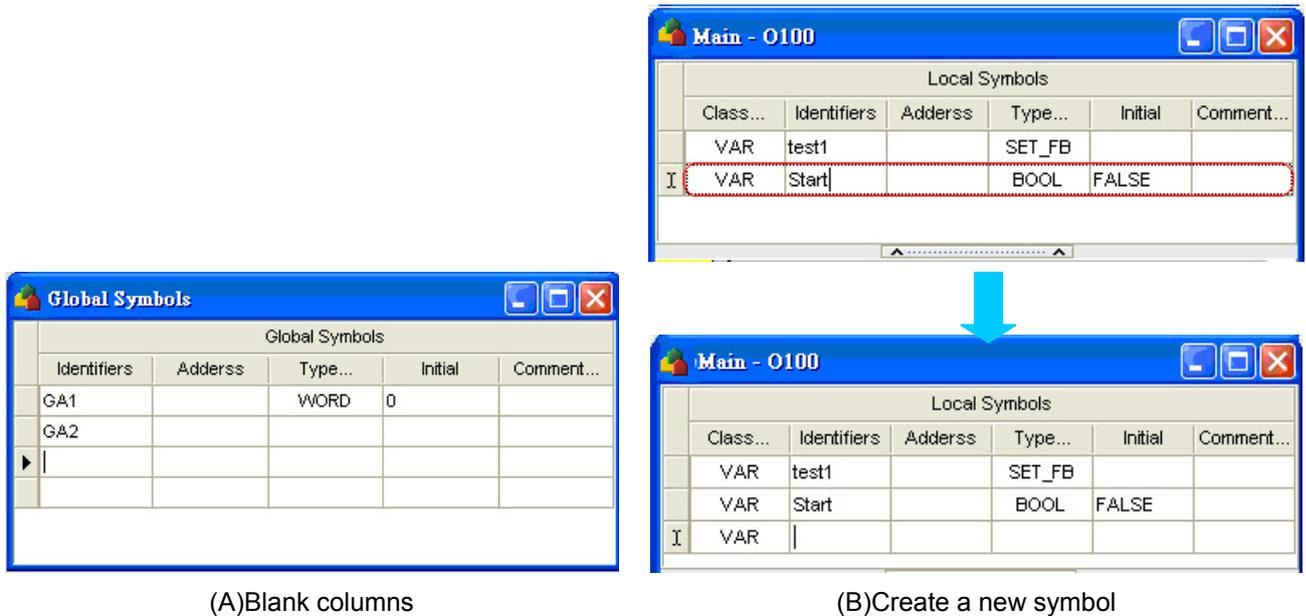
Local Symbol Table

■ Create New Symbols and Declare Symbols

12 POU Editing Mode

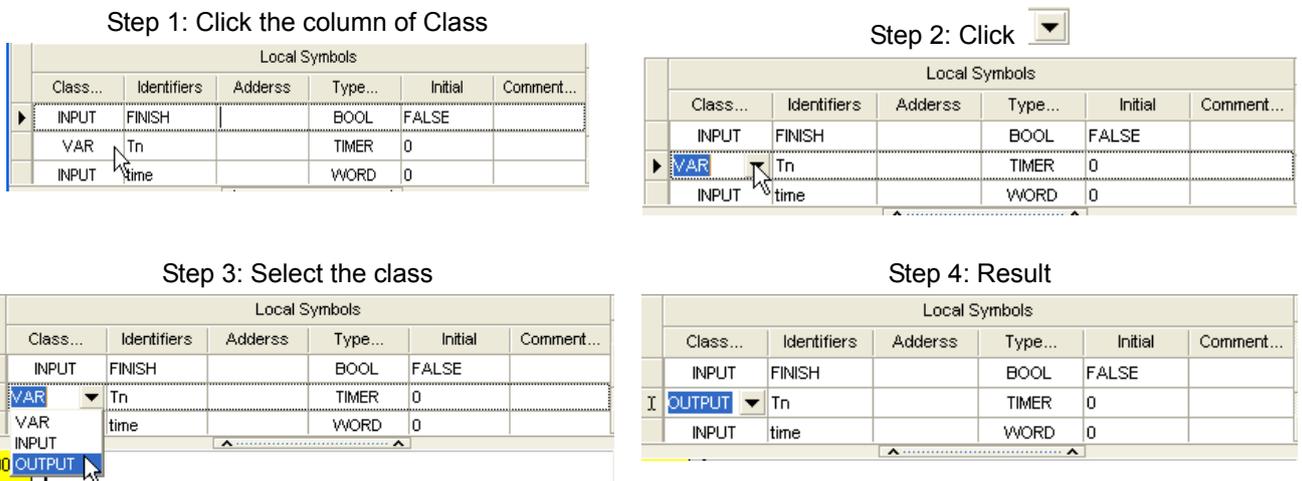
1. Create New Symbols:

Click the blank columns in symbol table to declare a symbol (Diagram A). If a new column is required, click the last column in the table and press “Enter” to create a new one (Diagram B), so that a new symbol can be created and declared.



2. Declare Symbols:

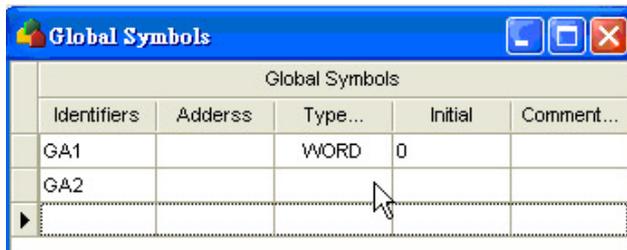
- **Class:** Class of a symbol. Only local symbol table provides the column of Class. 3 classes including VAR, INPUT and OUTPUT are available for local symbol table of function blocks; only VAR is available for symbol table of other programs or subroutines. Users can key in the class directly, or click the column of Class to generate the arrow  and select the desired class from the drop down list as below.



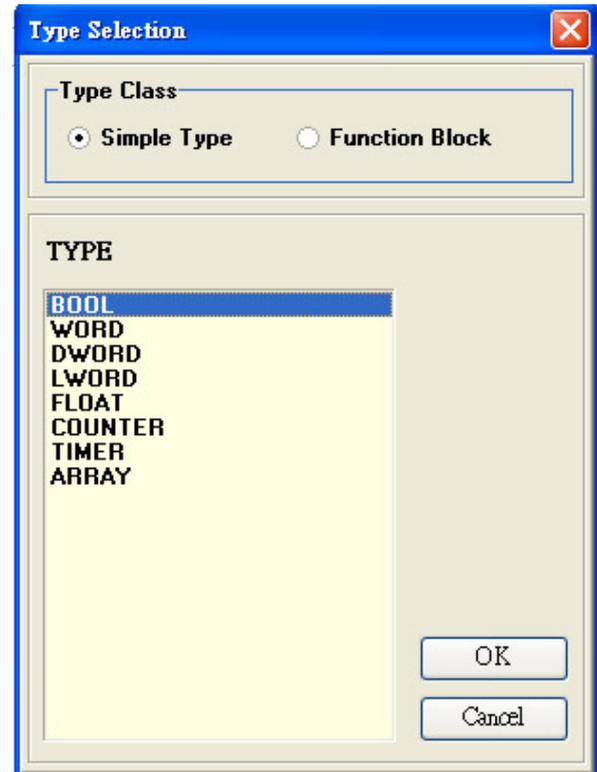
12 POU Editing Mode

- Identifiers: Click the column of Identifiers and input the proper names of identifiers.
- Address: The column of Address can be specified with devices by users or allocated by system.
- Type: Data type of a Symbol. Two types of symbols are available for this column: Simple Type and Function Block. Users can key in the data type directly, or click the column of Type to generate the arrow "▼". Click the arrow and a Type Selection window will pop up as below.

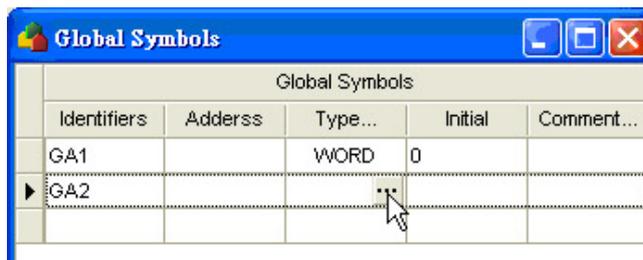
Step 1 : Click the column of Type



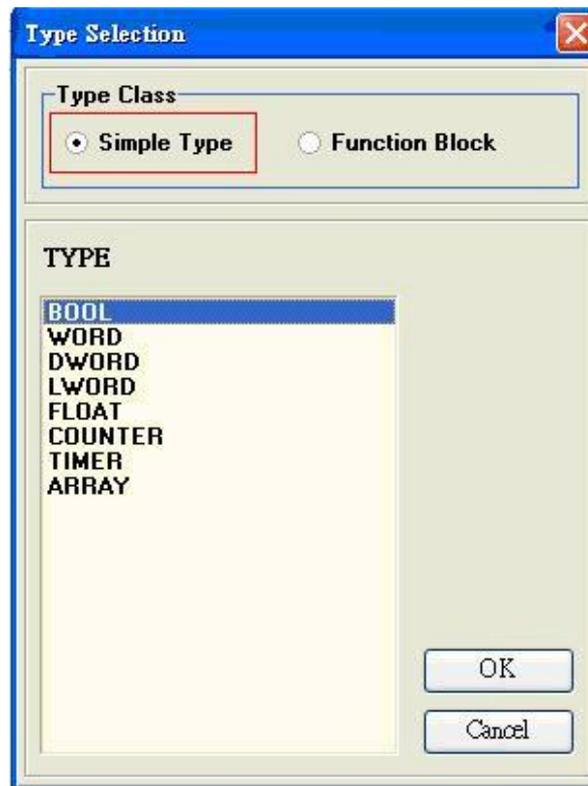
Step3 : Select types



Step 2 : Click ...



(a) **Simple Type:** Declare the Type of a symbol as general devices.

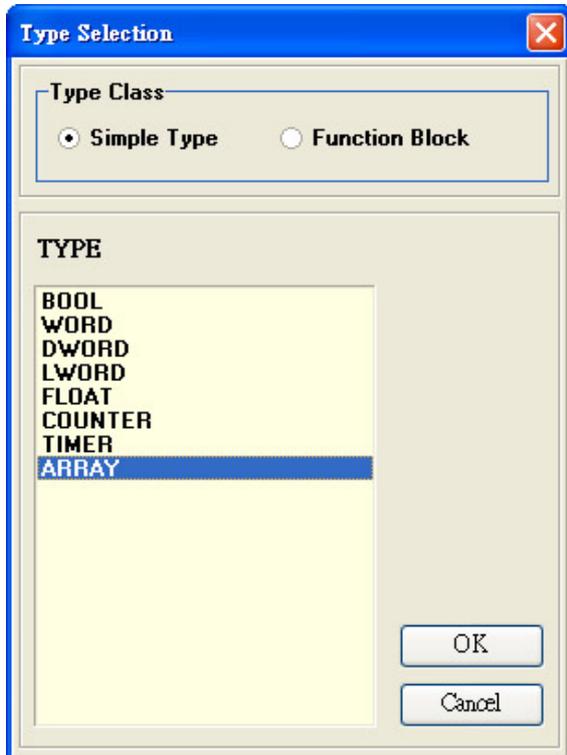


- ◆ **BOOL**: Declare the data type as Boolean logic (1-bit), and system will allocate M devices.
- ◆ **WORD**: Declare the data type as 16-bit, and system will allocate one D device.
- ◆ **DWORD**: Declare the data type as 32-bit, and system will allocate two D devices.
- ◆ **LWORD**: Declare the data type as 64-bit, and system will allocate four D devices.
- ◆ **FLOAT**: Declare the data type as 32-bit float, and system will allocate two D devices dor floating point instructions.
- ◆ **COUNTER**: Declare the data type as 16-bit counter, and system will allocate C devices.
- ◆ **TIMER**: Declare the data type as 16-bit timer, and system will allocate T devices.
- ◆ **ARRAY**: Declare the data type as array. Select ARRAY then click OK to enter the Array Type window.

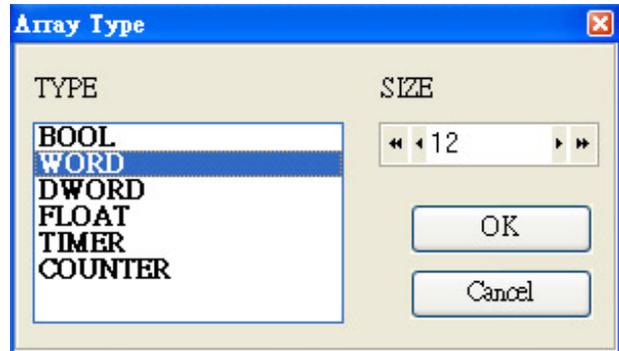
Select data type and size in the window. The below diagram shows steps to declare the current array as WORD format and the size as 2.

12 POU Editing Mode

Step 1: Select ARRAY



Step 2: Specify data type and size

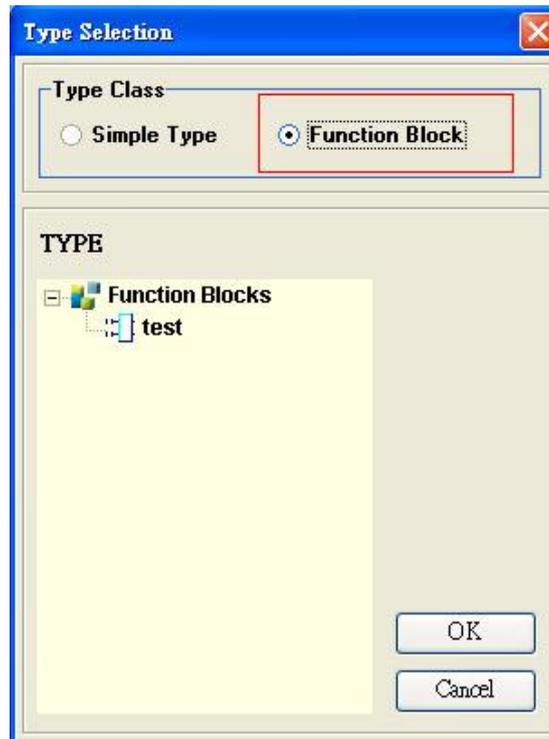


Step3: Results

Local Symbols					
	Class...	Identifiers	Address	Type...	I
	VAR	TEST_1		BOOL	FALSE
	INPUT	TEST_2		BOOL	FALSE
	OUTPUT	TEST_3		BOOL	FALSE
▶	INOUT	TEST_4		WORD[12]	

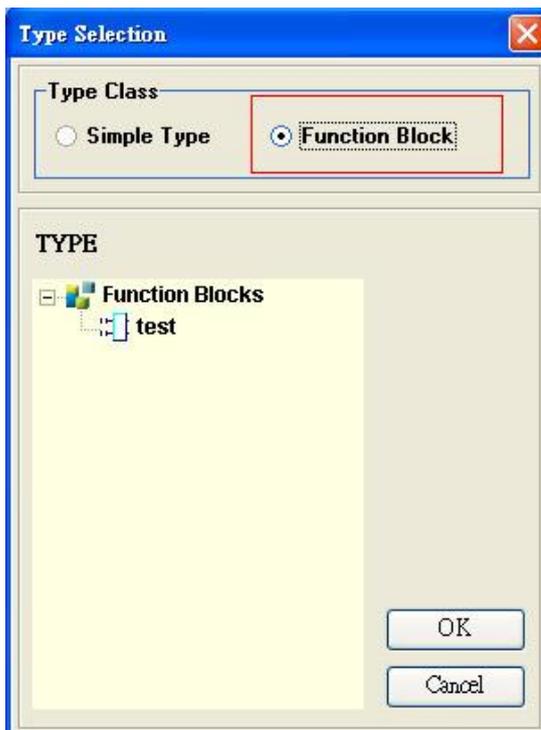
(b) Function Block: Declare the Type Class of a symbol as function block. The names of available function-block symbols will be listed as below.

12 POU Editing Mode

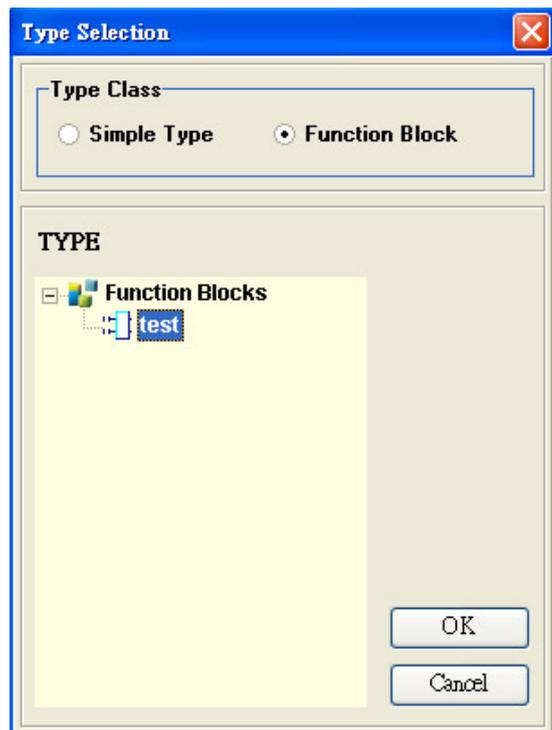


Function blocks have to be declared with a function-block symbol in the symbol list. Please follow the steps below to declare the function-block symbol.

Step 1: Select Function Block as the type class



Step 2: Select the symbol to be declared for the function block



12 POU Editing Mode

Step3: Results

Local Symbols						
Class...	Identifiers	Address	Type...	Initial	Comment...	
VAR			test			

- Initial: Initial value of the symbol. The value will be written into PLC with the program.
- Comment: The information or explanations of the symbol.

Initial Setting of Symbols

After data type is specified, click the column "Initial" in symbol table to configure the initial setting for symbols. If the symbol does not need to be initialized, clear the content in the column.

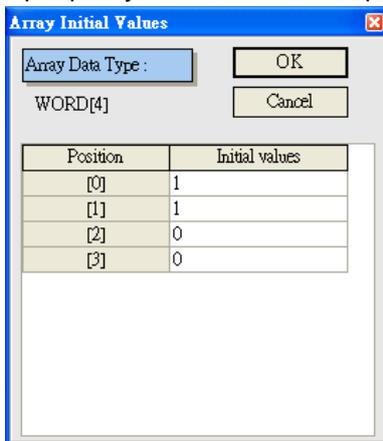
1. When the data type of symbol is specified as BOOL, the initial setting has to be TRUE or FALSE.

Local Symbols						
Class...	Identifiers	Address	Type...	Initial	Comment...	
VAR	Bit		BOOL	FALSE		

2. When the data type of symbol is specified as WORD or DWORD, a prefix K or H respectively indicating decimal or hex value can be added in front of the initial value, e.g. K100, H2000.

Local Symbols						
Class...	Identifiers	Address	Type...	Initial	Comment...	
VAR	word1		WORD	0		
VAR	word2		WORD	K100		
VAR	word3		WORD	H2000		

3. When the data type of symbol is specified as Array, click Initial and the Array Initial Values window will pop up. Specify initial values in the positions of Array then click OK, the initial values of array will be created.



Results

Local Symbols						
Class...	Identifiers	Address	Type...	Initial	Comment...	
VAR	data	D7000	DWORD	0		
VAR	outdata		WORD[4]	[1(2),0(2)]		
VAR	dmul_32		DMUL_FB			

4. The format of Initial value:
 - The value should be placed between brackets.

12 POU Editing Mode

- Commas should be inserted between each position, e.g. [1,2,3,4]
- If there are repeated values, apply the following format “[value (times of repetitions)]”, e.g. 1(2) indicates the value 1 is repeated for 2 times.

Local Symbols						
Class...	Identifiers	Address	Type...	Initial	Comment...	
VAR	outdata		WORD[4]	[1(2),0(2)]		

■ Modify Symbols in Symbol Table

The methods to modify symbols are the same as specifying new symbols. Users can modify symbols directly in the symbol table as below.

Local Symbols						
Class...	Identifiers	Address	Type...	Initial	Comment...	
OUTPUT	DATA		WORD	0		
▶ OUTPUT	FINISH	D200	BOOL	FALSE		
VAR	Tn		TIMER	0		
INPUT	time		WORD	0		
OUTPUT						

Notes:

1. For function-block POU, the data type of symbol can not be specified as function block.
2. For function-block POU, Address should be allocated by system if Class of the symbol is specified as INPUT or OUTPUT

■ Move Symbols in Symbol Table

1. Select a symbol to be moved

Local Symbols						
Class...	Identifiers	Address	Type...	Initial	Comment...	
VAR	DATA		WORD	0		
▶ INPUT	FINISH		BOOL	FALSE		
INPUT	Tn		TIMER	0		
INPUT	time		WORD	0		

2. Press keys [Alt + ↓] or [Alt + ↑] to shift the location of the symbol.

Local Symbols						
Class...	Identifiers	Address	Type...	Initial	Comment...	
VAR	DATA		WORD	0		
INPUT	Tn		TIMER	0		
▶ INPUT	FINISH		BOOL	FALSE		
INPUT	time		WORD	0		

■ Delete Symbols in Symbol Table

Users can select the symbols to be deleted and delete symbols by the following methods:

Method 1: Click "Edit" then click "Delete" in the drop down list.

12 POU Editing Mode

Method 2: Click the icon  on the toolbar.

Method 3: Right click the mouse and select “Delete” from the pop up menu.

Method 4: Press the key [Delete] on the keyboard.

Deletion can be operated on single row or multiple rows.

- Single-row deletion: Click the row to be deleted and the selected row will be indicated by  in the front. Apply one of the 4 methods to delete the row as below (GA2 is deleted).

Step 1: Click the row to be deleted

Global Symbols				
Identifiers	Address	Type...	Initial	Comment...
GA1		WORD	0	
GA2		DWORD	0	
GA3		LWORD	0	
GA4		FLOAT	0	

Step 2: Delete

Global Symbols				
Identifiers	Address	Type...	Initial	Comment...
GA1		WORD	0	
▶ GA2		DWORD	0	
GA3		LWORD	0	
GA4		FLOAT	0	

Step 3: Result

Global Symbols				
Identifiers	Address	Type...	Initial	Comment...
GA1		WORD	0	
▶ GA3		LWORD	0	
GA4		FLOAT	0	
GA5		COUNTER	0	

- Multi-row deletion: Click the first row to be deleted. Press Shift on the keyboard and click the last row to be deleted. Apply one of the 4 methods to delete the selected rows as below (Symbol GA2~GA5 are deleted).

Step1: Select the first row to be deleted

Global Symbols				
Identifiers	Address	Type...	Initial	Comment...
GA1		WORD	0	
GA2		DWORD	0	
GA3		LWORD	0	
GA4		FLOAT	0	
GA5		COUNTER	0	
GA6		TIMER	0	

Step2: Shift+ the last row to be deleted

Global Symbols				
Identifiers	Address	Type...	Initial	Comment...
GA1		WORD	0	
▶ GA2		DWORD	0	
GA3		LWORD	0	
GA4		FLOAT	0	
GA5		COUNTER	0	
GA6		TIMER	0	

Step3: Delete

Step4: Result

12 POU Editing Mode

Global Symbols				
Identifiers	Address	Type...	Initial	Comment...
GA1		WORD	0	
GA2		DWORD	0	
GA3		LWORD	0	
GA4		FLOAT	0	
GA5		COUNTER	0	
GA6		TIMER	0	

Global Symbols				
Identifiers	Address	Type...	Initial	Comment...
GA1		WORD	0	
GA6		TIMER	0	
GA8		BOOL[3]		
GA9		TIMER[2]		
GA10		COUNTER[4]		
GA11		BOOL[1]		

Export Symbols:

Symbols can be exported to a file with an extension name "*.csv".

1. Move the cursor to the symbol table and right click to select Export Symbols from the pop-up list.

Step 1: move the cursor to the symbol table

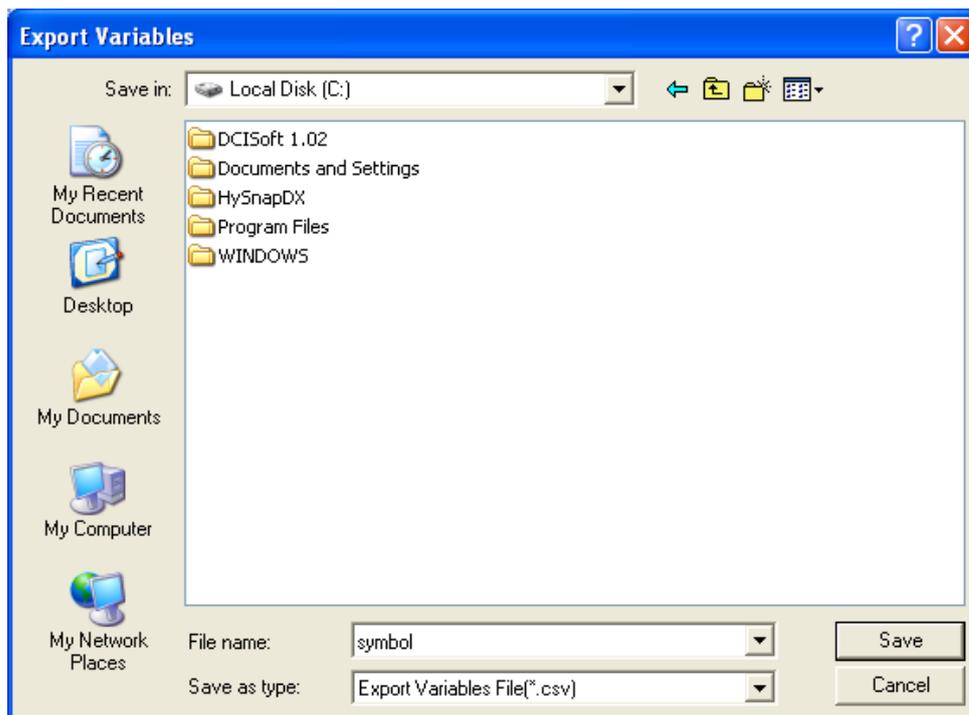
Local Symbols			
Class...	Identifiers	Address	Type...
VAR	tt		WORD
VAR	add_16		add_fb
VAR	ee		WORD

Step 2: Right click to select "Export Symbols" from the pop-up list.

Local Symbols					
Class...	Identifiers	Address	Type...	Initial	Con
VAR	tt		WORD		
VAR	add_16		add_fb		
VAR	ee		WORD		

✂	Cut	Ctrl+X
📄	Copy	Ctrl+C
📄	Paste	Ctrl+V
🗑	Delete	Del
🔍	Select All	Ctrl+A
➕	Insert a Row	Ctrl+I
🔍	Find	Ctrl+F
➡	Find Next	F3
⬅	Find Previous	Ctrl+F3
📄	Import Symbols...	
📄	Export Symbols...	

2. set up the name and the path for the file to be exported

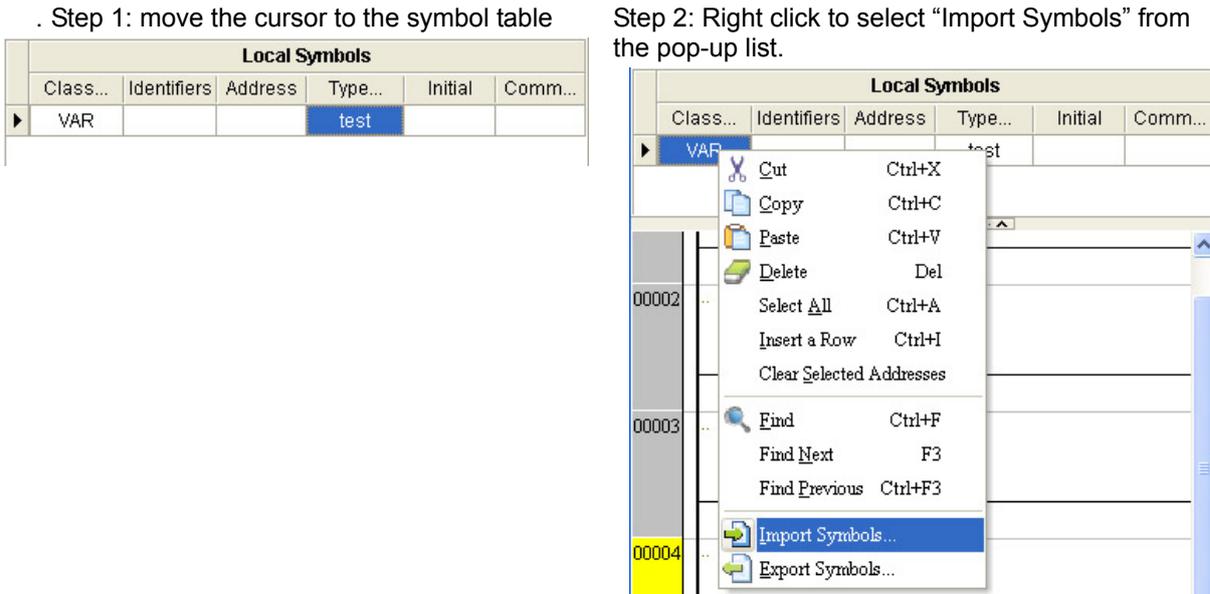


12 POU Editing Mode

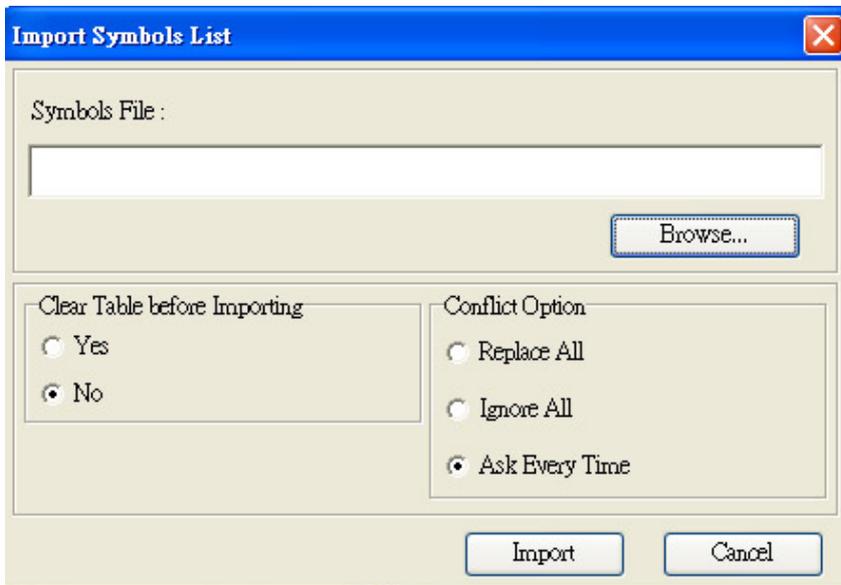
■ Import Symbols

The exported symbols can be imported by selecting the file with an extension name “*.cvs”. After the symbol file is imported, the symbols in the file will be displayed in the symbol table.

1. Move the cursor to the symbol table and right click to select Import Symbols from the pop-up list.



2. Select the file to be imported and set up the preference setting for importing symbols.



- Symbols File: select the file to be imported.
- Clear Table before Importing: The symbols in current symbol table will be cleared if “Yes” is selected and be held if “No” is selected.
- Conflict Option: When “Replace All” is selected, the repeated symbols will be replaced during the import

12 POU Editing Mode

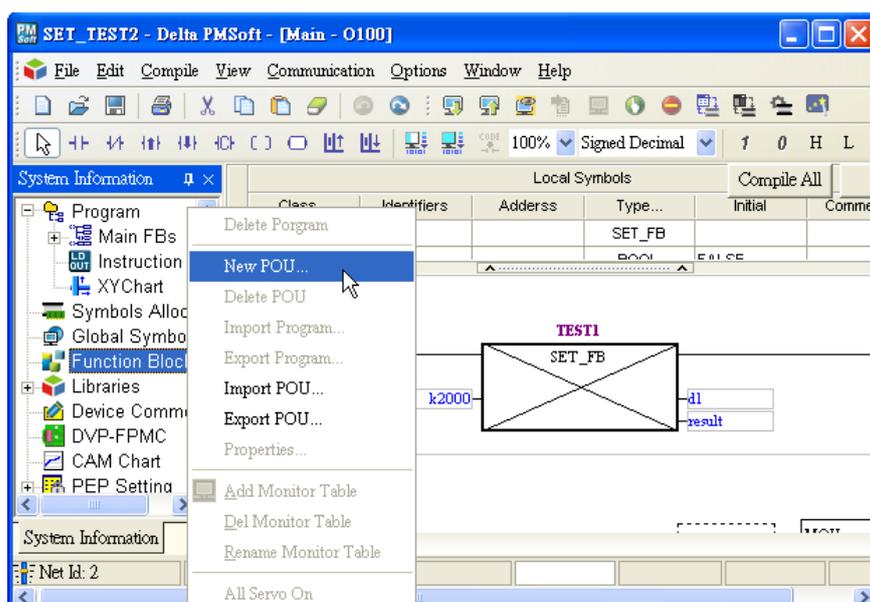
process. When “Ignore All” is selected, the repeated symbols will not be replaced. If “Ask Every Time” is selected, a confirming window will pop-up every time when conflict (repeated symbols) happens.

12.1.3 Create POU Function Block

Function block can be used many times in a program, and it also benefits users by simplifying user program. The input symbol and output symbol of a function block are declared by local symbol table of the function block. If Class in local symbol table is specified as INPUT, the symbol functions as the input interface of the function block’s data source. If Class in local symbol table is specified as OUTPUT, the symbol functions as the output interface of the function block’s operation results. Function blocks can be modified and applied according to users’ requirements.

■ When a new program is opened, users need to create a new POU function block as program editing area. Methods for creating new POU are explained as below:

1. Right click at the node of “Function Blocks” under “System Information” and select “New POU.”



2. The window of “Create a New POU” will pop up as below.

12 POU Editing Mode



Rules for Naming a POU:

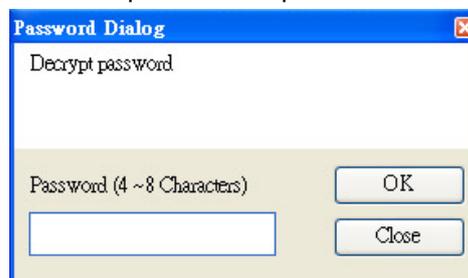
- Symbols such as “~, !, @, #, \$, %, ^, &, *, (,), “ are not allowed.
- POU names should not be repeated in the same program.
- Case-insensitive.
- Max length is 20 characters.

Protection:

When password is set up, users need to enter correct password for accessing the POU. Take the following case for example. DMOV_FB has been locked by password, so users need to enter correct password for accessing the internal program in POU. Error message will show up if an incorrect password is entered.

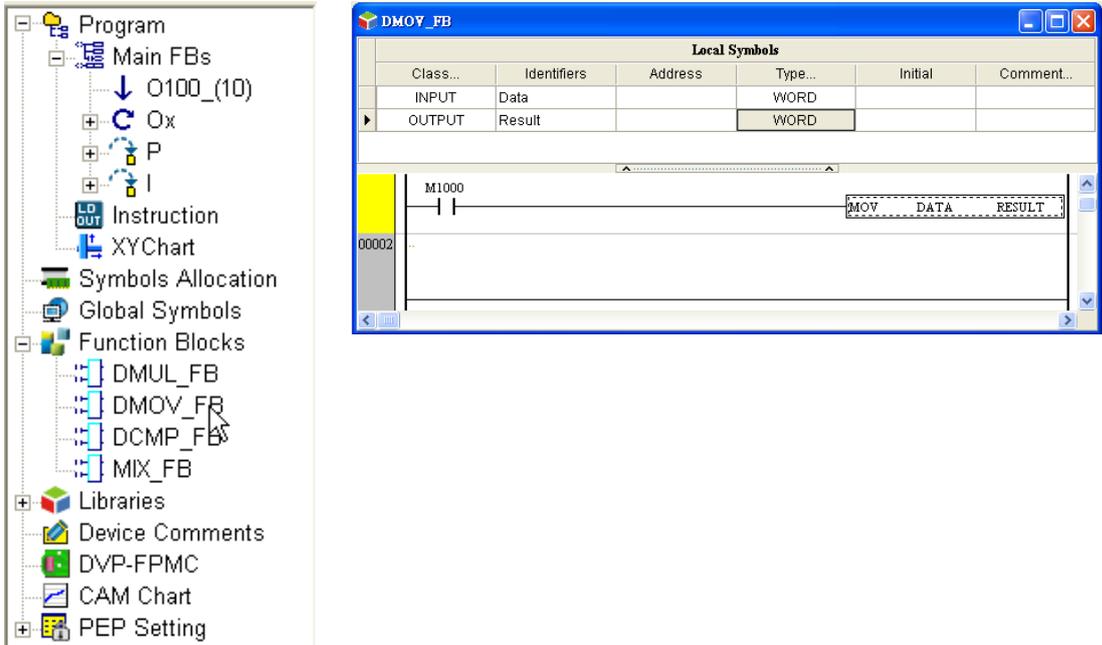
Step1: double-click on
DMOV_FB

Step2: enter the password.



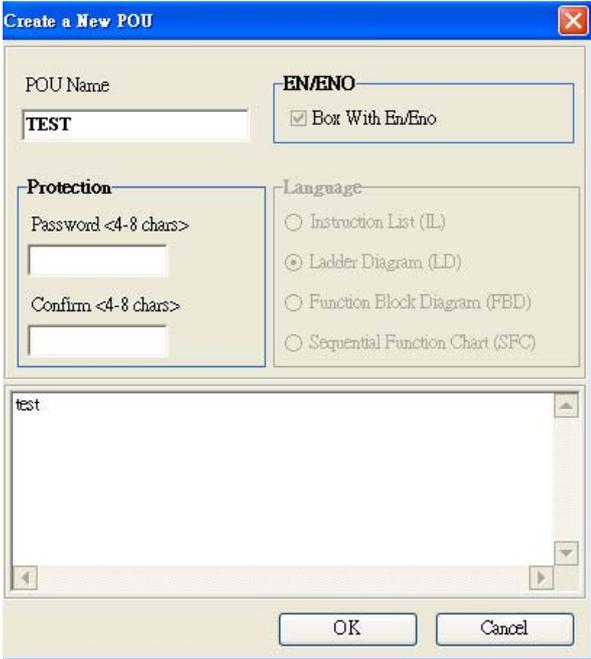
Step3: Results

12 POU Editing Mode



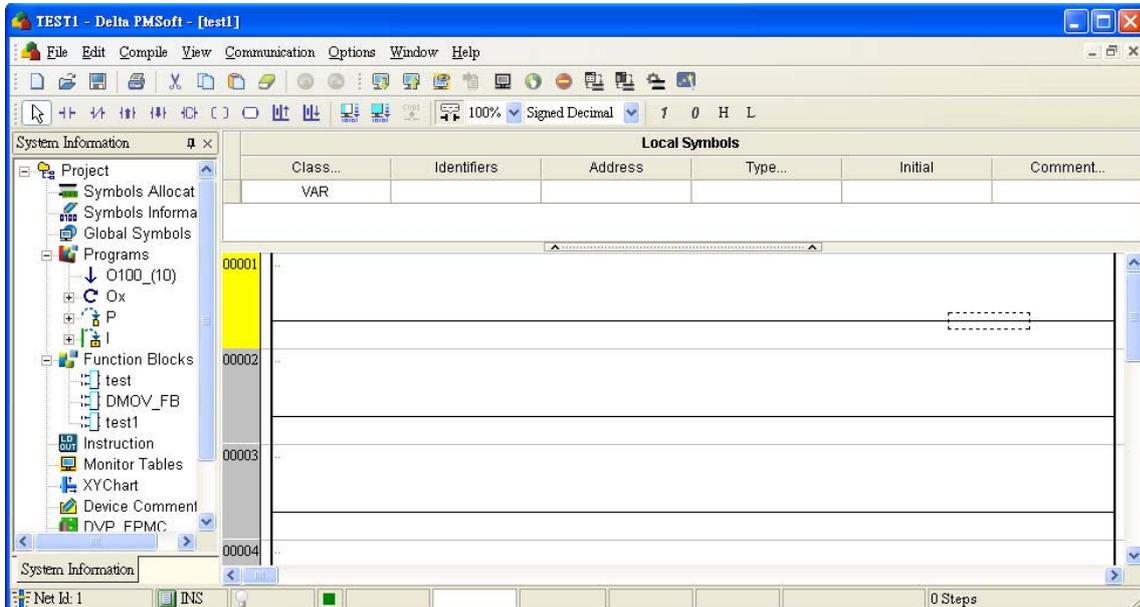
POU Comment:

Edit the POU comment as a POU explanation in the blank area under the window of "Create a New POU"



- 3. Click OK and the ladder editing window will pop up as below.

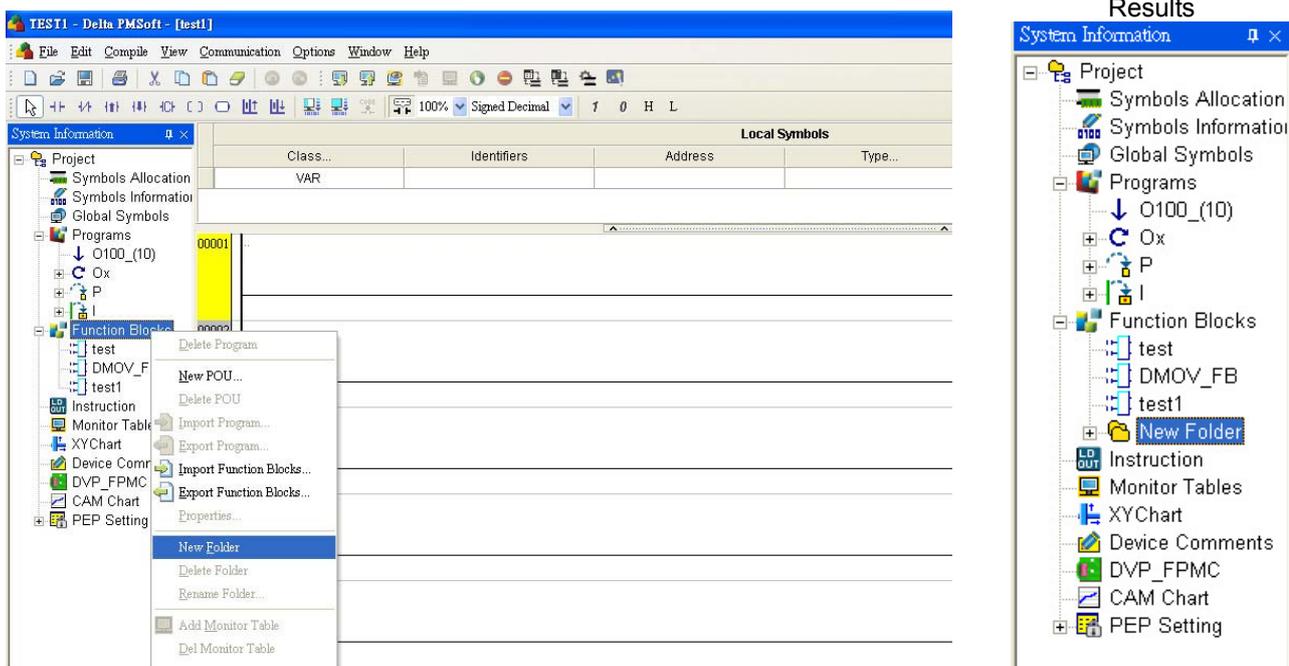
12 POU Editing Mode



12.1.4 Create POU Folders

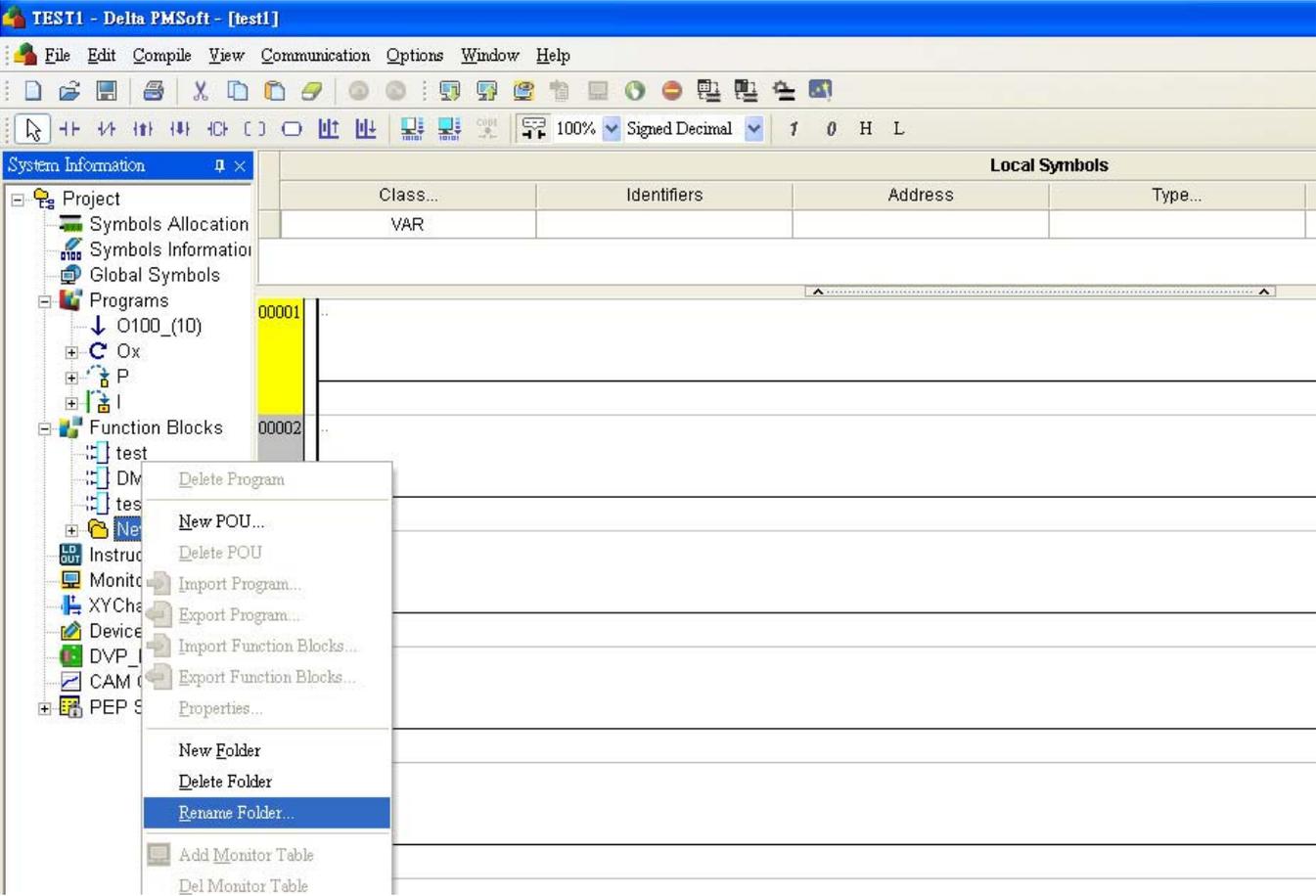
Users can create new folders under Function Blocks for further classification. The methods to create and edit folders are explained as below:

1. Create new folder: Move the cursor to Function Blocks under System Information. Right click and select New Folder from the drop-down list and a New Folder will automatically be created under Function Blocks.

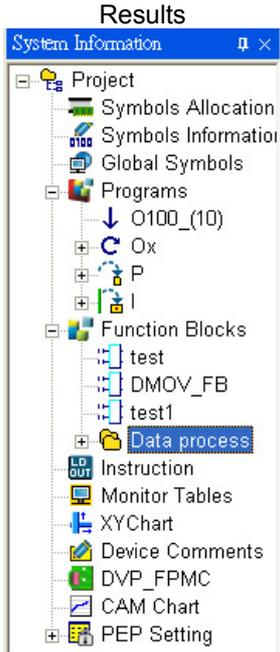


2. Rename the folder: Move the cursor to New Folder. Right click and select Rename Folder.

12 POU Editing Mode

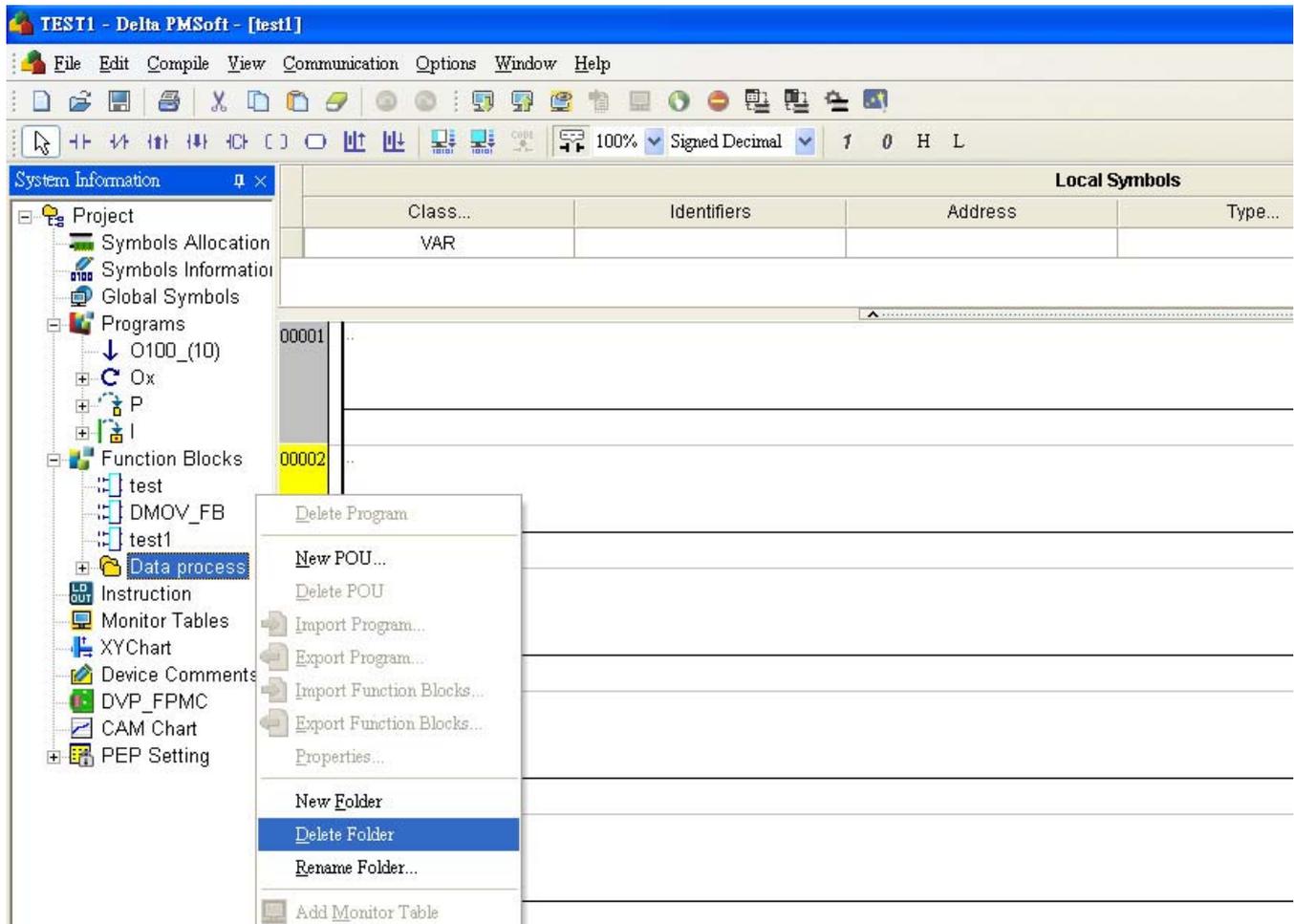


The Rename Folder window will pop up. Enter the new name for this folder and click OK to finish.

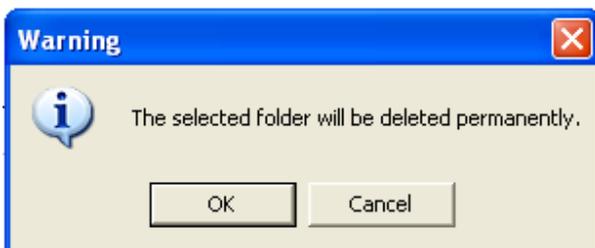


3. Delete the folder: Move the cursor to the folder to be deleted. Right click and select Delete Folder.

12 POU Editing Mode

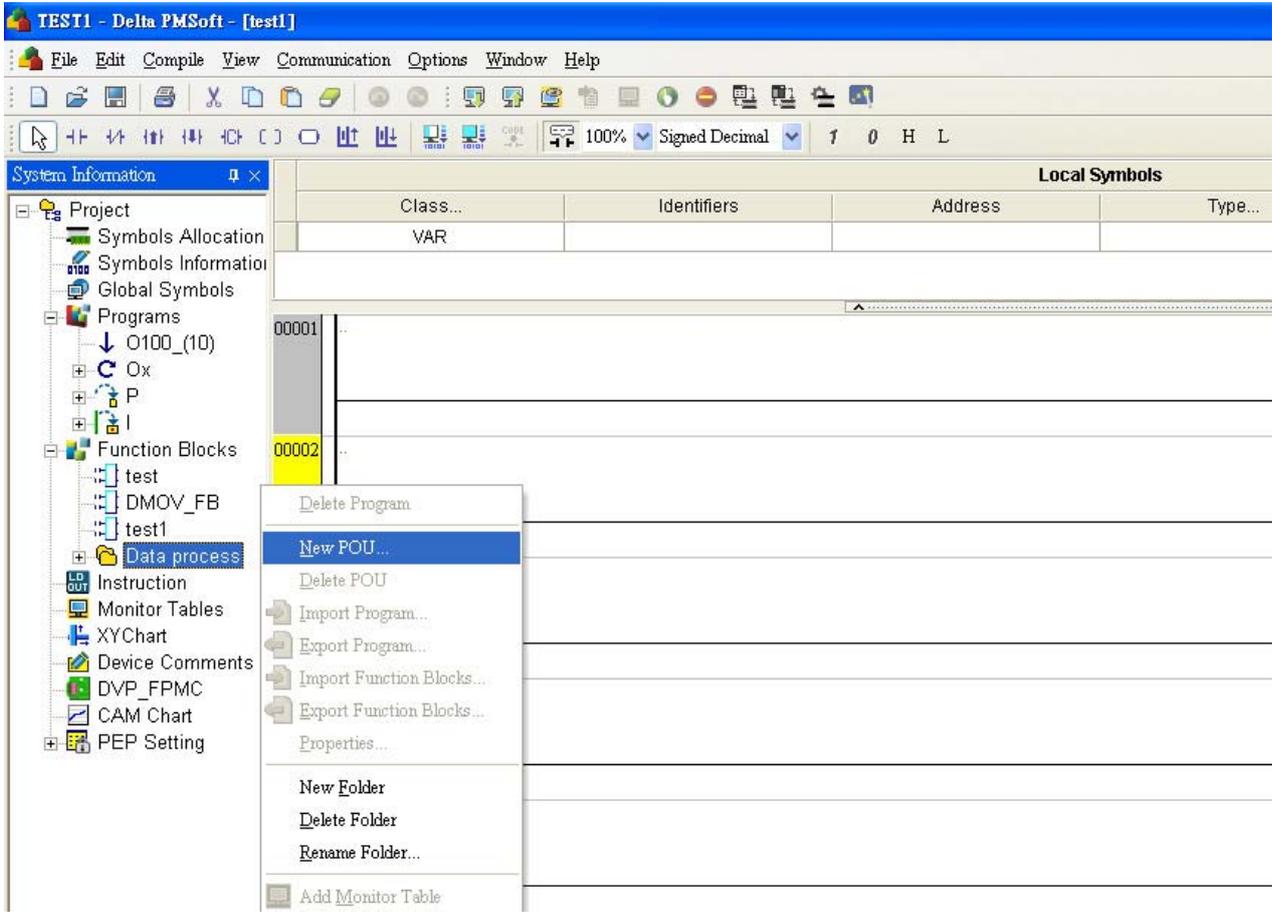


A warning window will pop up to confirm the deletion. Click OK and the selected folder will be deleted.



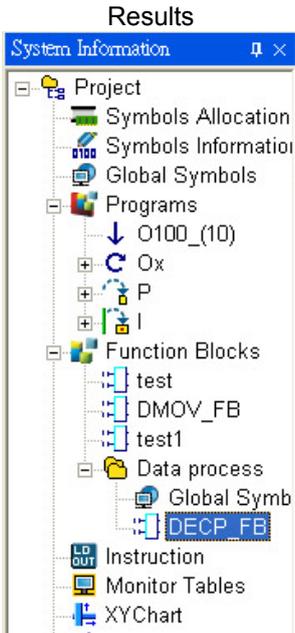
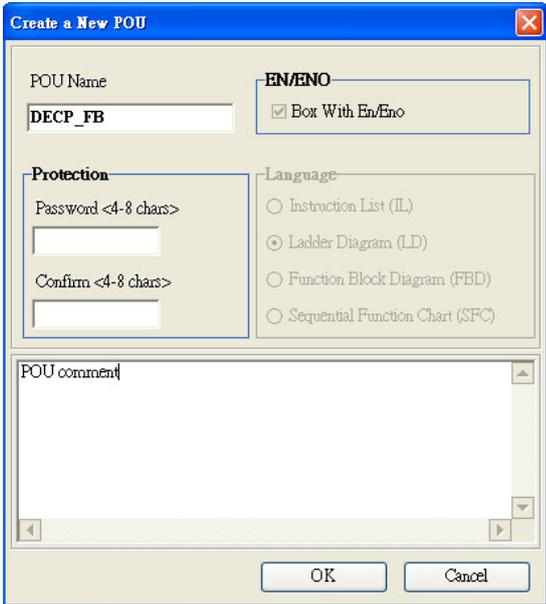
4. Create new POU in the folder: Move the cursor to the target folder and right click to select New POU.

12 POU Editing Mode



Create a New POU window will be displayed. For POU naming rules please refer to Rules for Naming a POU in

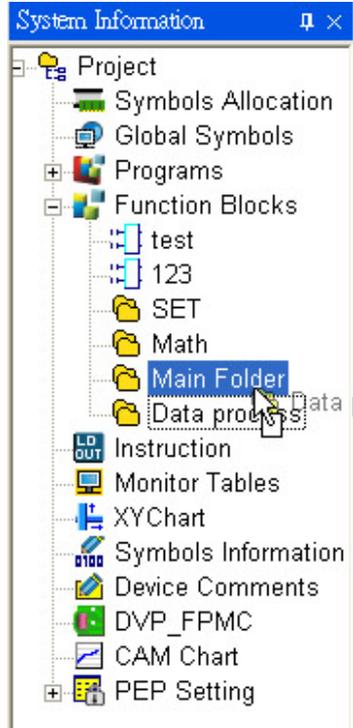
12.1.3. Enter the POU name and click OK. The new POU will be displayed under the target folder.



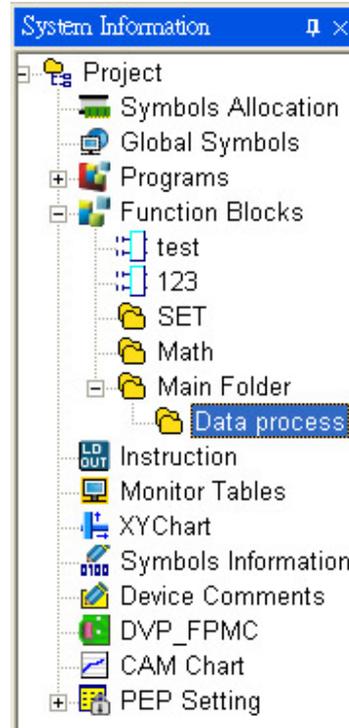
12 POU Editing Mode

5. Move the folders: click on the folder to be moved or POU to be moved and drag it to the target folder. In this way, a folder or POU can be moved into other folders.

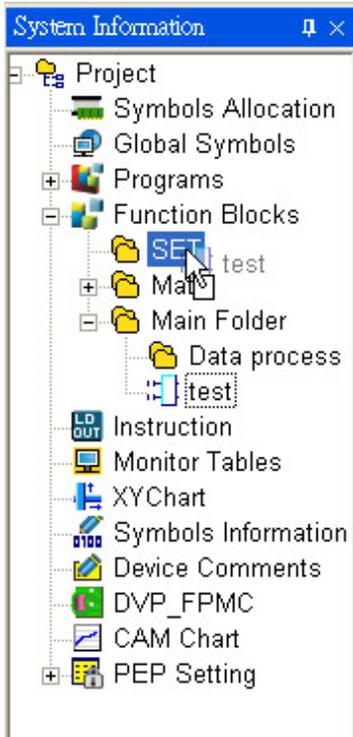
Move a folder into another folder



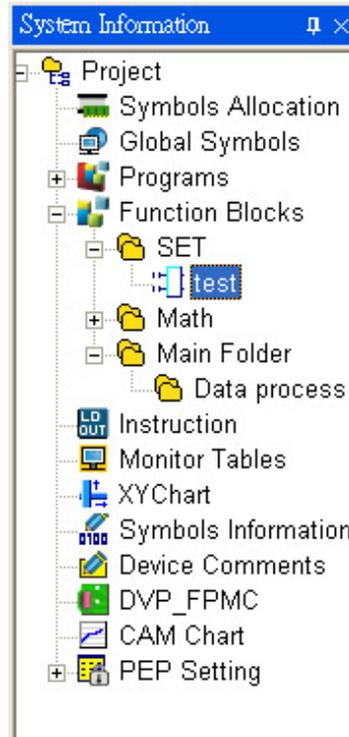
Results



Move a POU into another folder



Results



Note: when a folder is dragged, all the folders under the dragged folder will be moved as well. If the folders under the

12 POU Editing Mode

target folder have the same folder name as the source folders, a warning message will be displayed to inform users of the existence of the folder name.

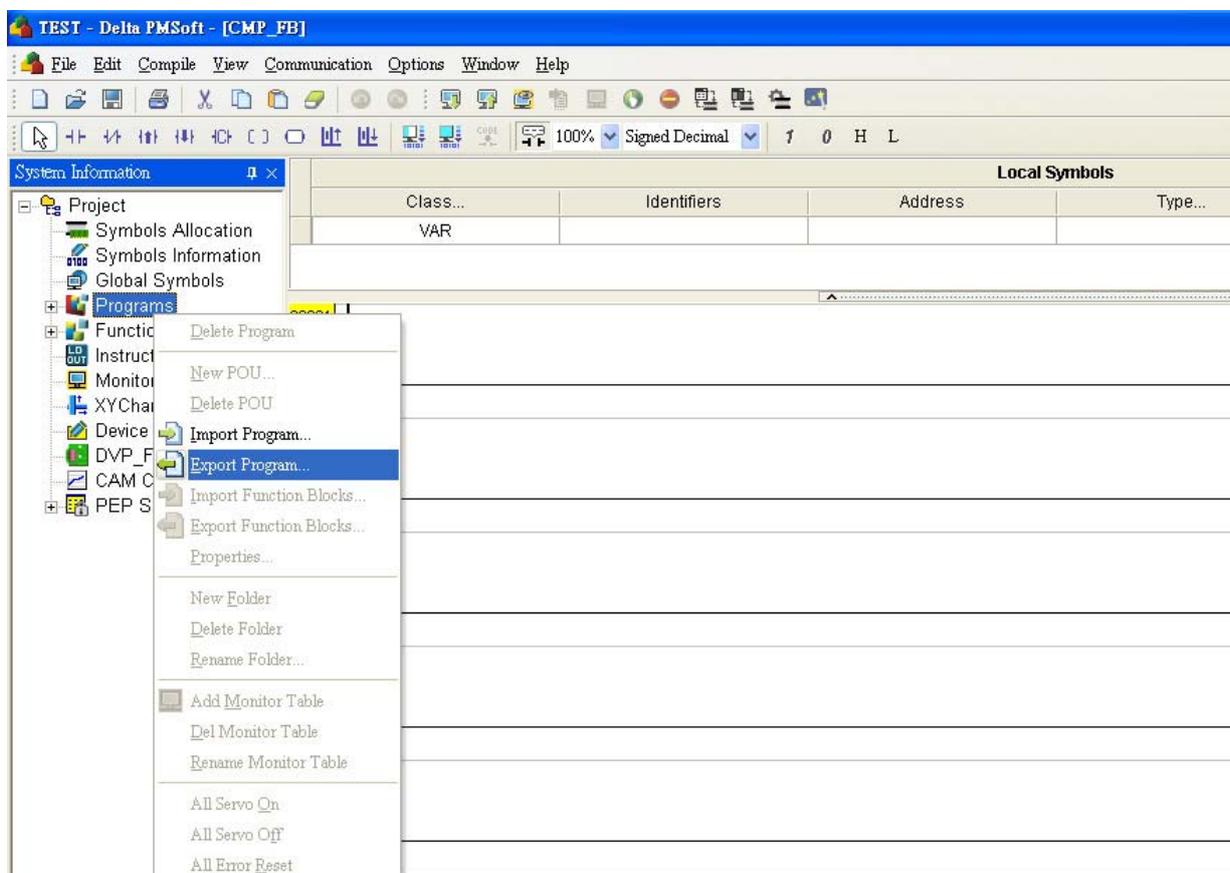


12.1.5 Export POU

Users can export POU's in the program so that the POU will be available to be imported when editing other programs. There are two types of POU including program POU and function-block POU. The steps to export POU are explained as below:

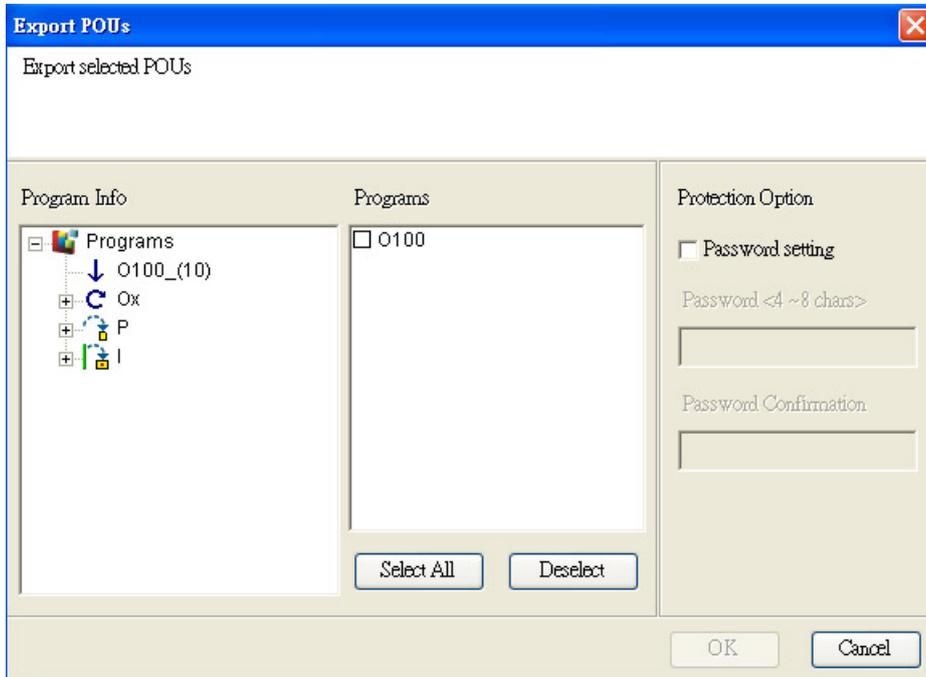
- Export program POU:

1. Right click at the node of "Programs" under "System Information" and select "Export Program."



12 POU Editing Mode

- Program information and all programs under program area will be displayed. If O100, Ox or P is selected, only programs under the selected program area will be displayed. You can click on the program POU to be exported in the program list.



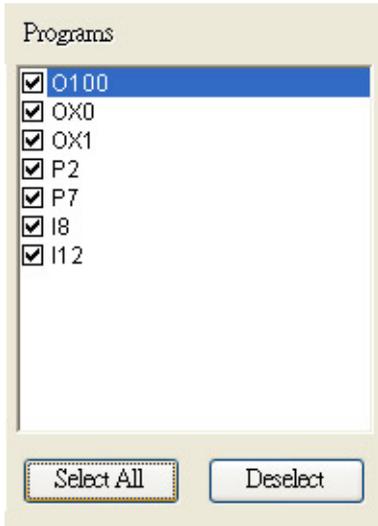
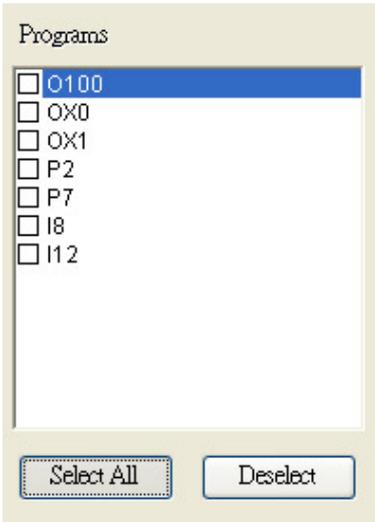
- The methods for selecting programs are explained as below:

(1) Select by clicking the program to be exported. In below diagram, "OX1" is selected.

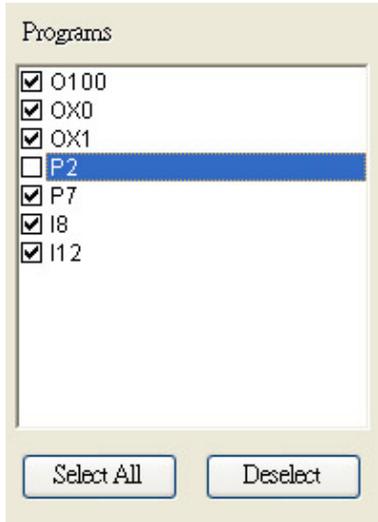
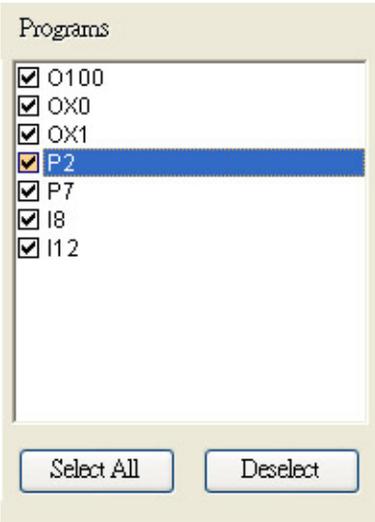


(2) Select All. Click the button  and all programs will be selected as below.

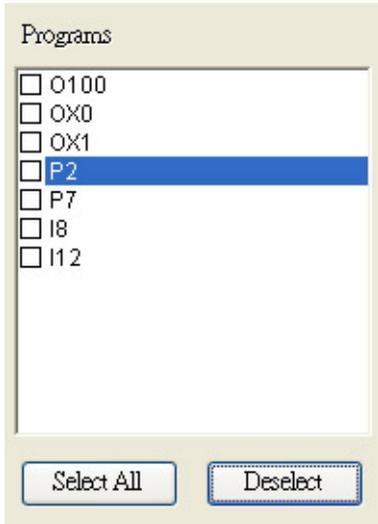
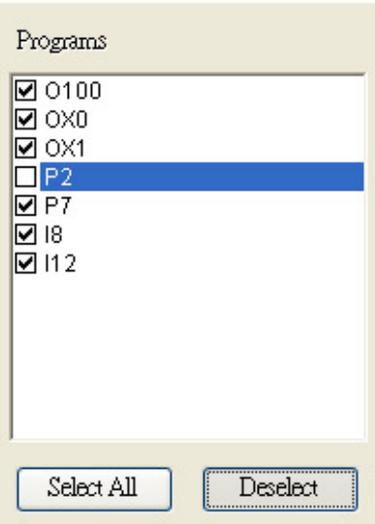
12 POU Editing Mode



(3) Deselect. Clicking the selected program will deselect it.

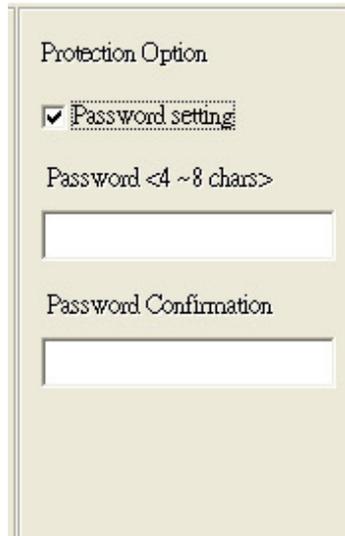


(4) Deselect All. Click the button "Deselect" and all programs will be deselected as below.



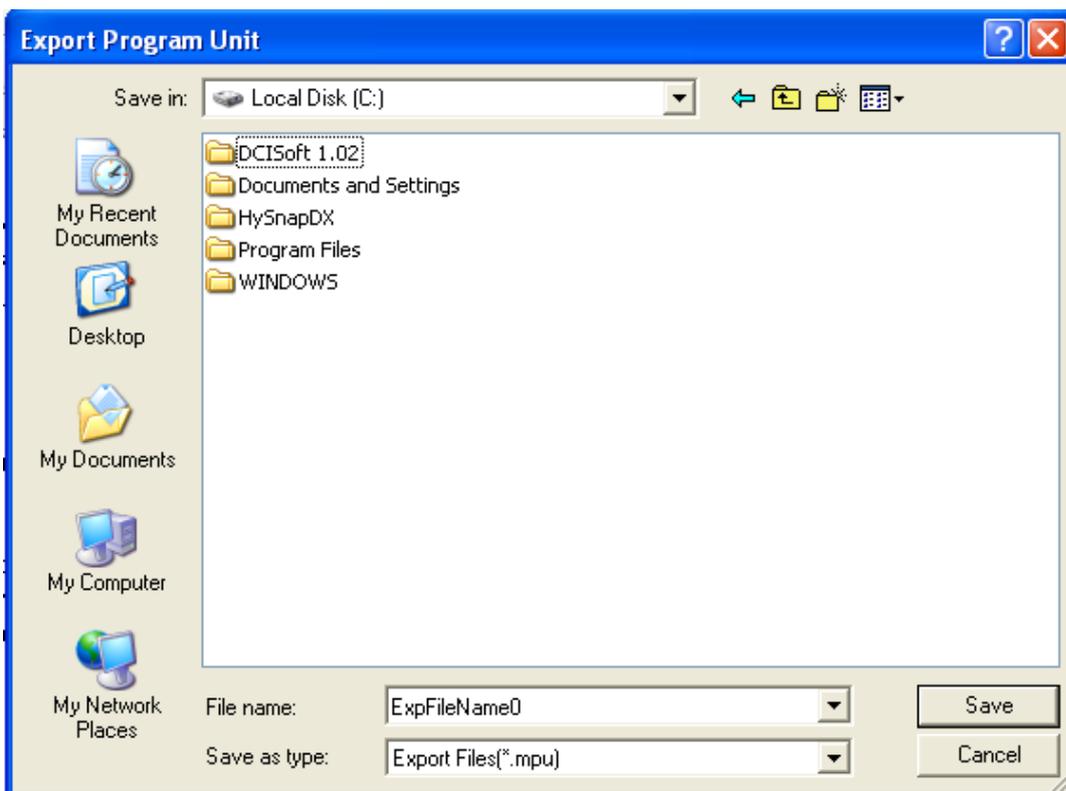
12 POU Editing Mode

3. Protection Option: Password protection on POU. Click “Password setting” to set up and confirm password.



The screenshot shows a dialog box titled "Protection Option". It contains a checked checkbox labeled "Password setting". Below this is a text input field labeled "Password <4 ~8 chars>". At the bottom, there is another text input field labeled "Password Confirmation".

4. When all options are set, click OK and the file saving window will pop up. Set up the path and the file name with the file extension “*.mpu”.



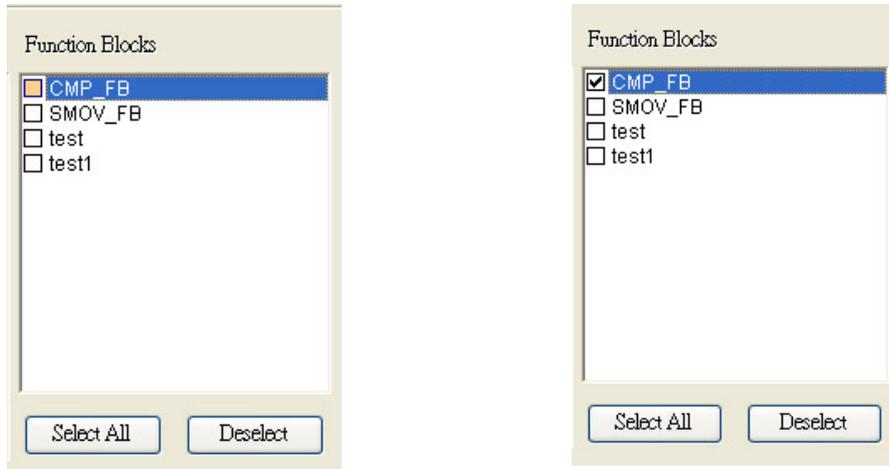
■ Export Function Block POU:

1. Right click at the node of “Function Blocks” under “System Information” and select “Export Function Blocks.”

12 POU Editing Mode

- The methods for selecting function blocks are explained as below:

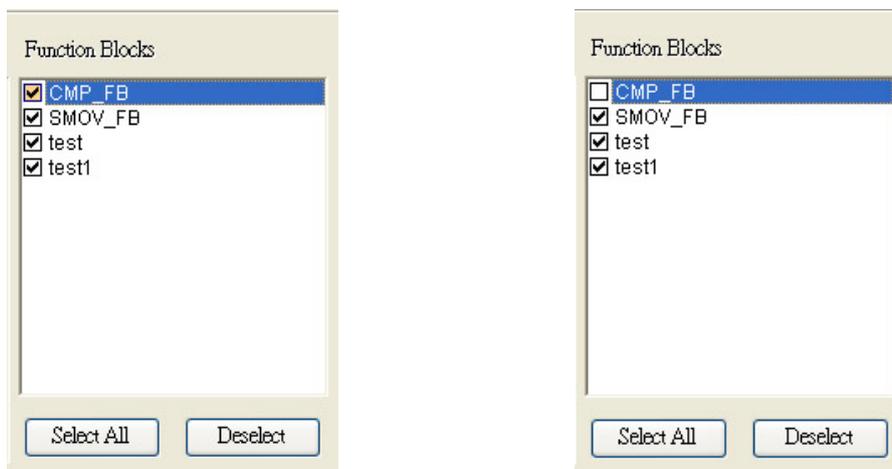
(1) Select by clicking the function block to be exported. In below diagram, "CMP_FB" is selected.



(2) Select All. Click the button  and all function blocks will be selected as below.

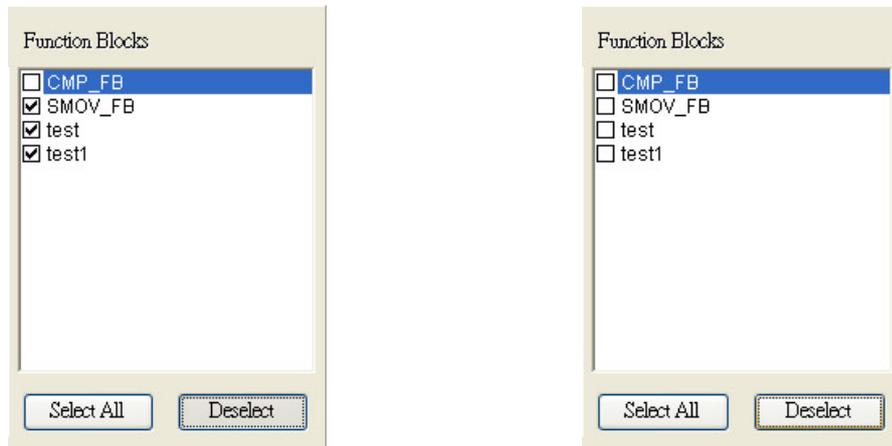


(3) Deselect. Clicking the selected function block will deselect it.



(4) Deselect All. Click the button "Deselect" and all function blocks will be deselected as below.

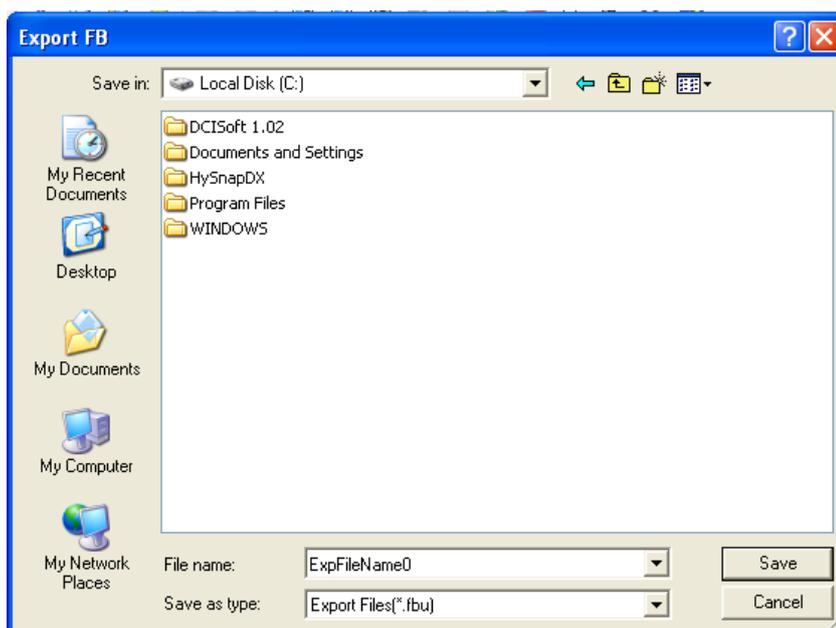
12 POU Editing Mode



- (5) Protection Option: Password protection on POU. Click "Password setting" to set up and confirm password.



- (6) When all options are set, click OK and the file saving window will pop up. Set up the path and the file name with the file extension ".fbu".



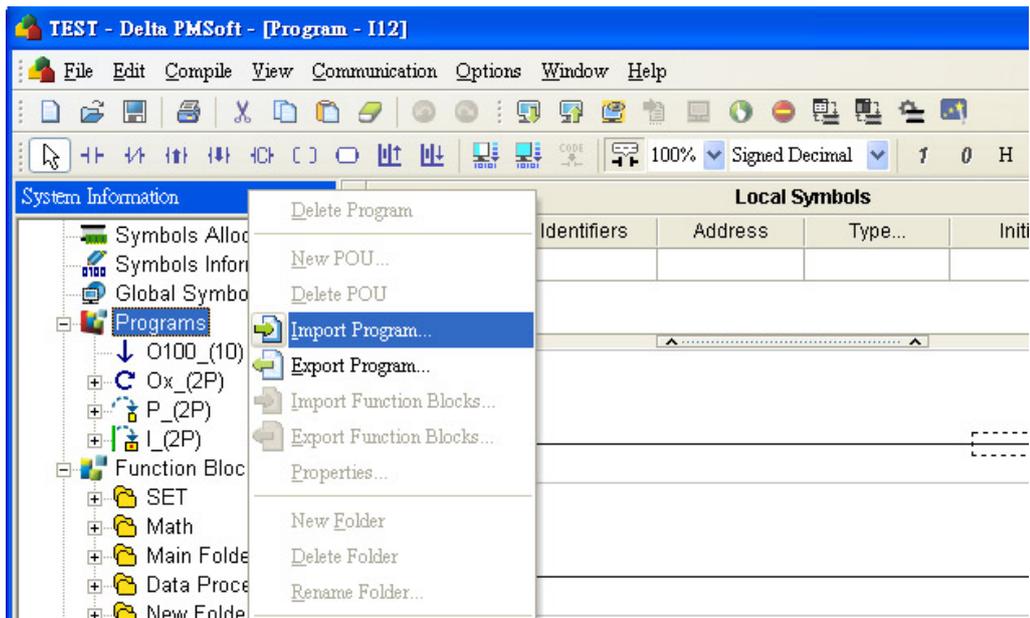
12 POU Editing Mode

12.1.6 Import POU

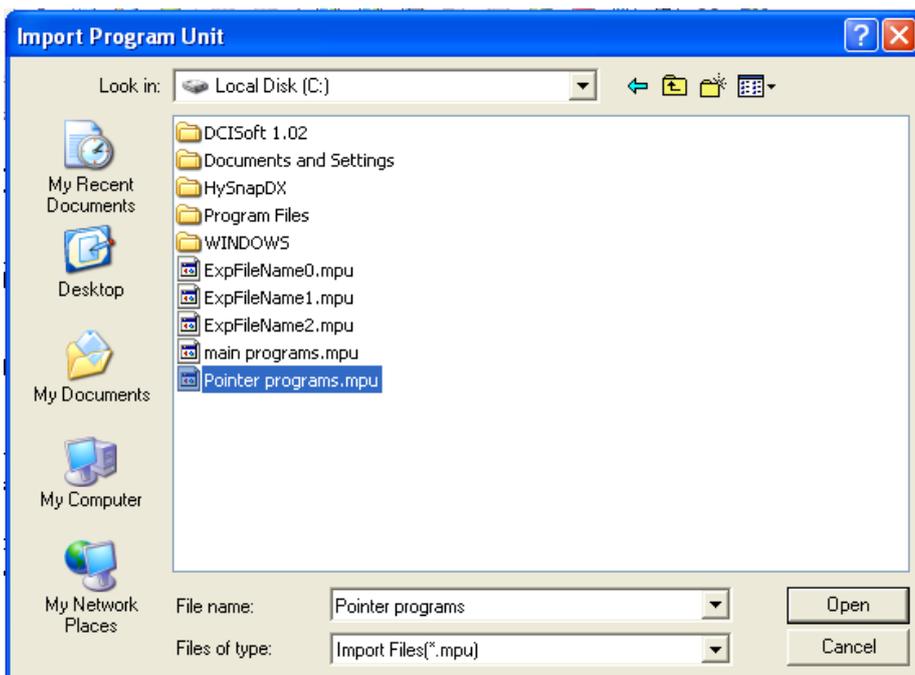
POUs can be imported so that users can save time on editing POUs with the same function. The steps to import POUs are explained as below:

- Import program POU:

1. Right click at the node of “Programs” under “System Information” and select “Import Program.”

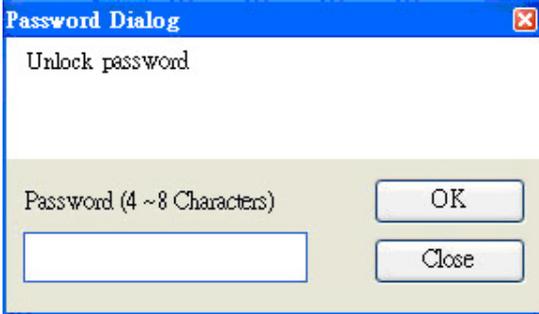


2. PMSOFT will display program POU files with the extension “*.mpu”. Select a file and click Open to import this file.

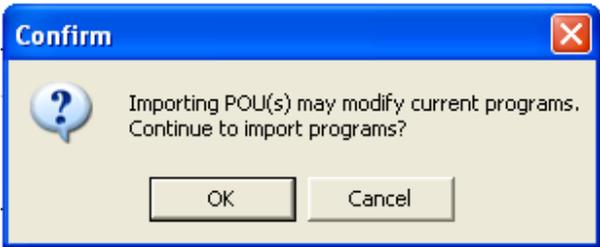


12 POU Editing Mode

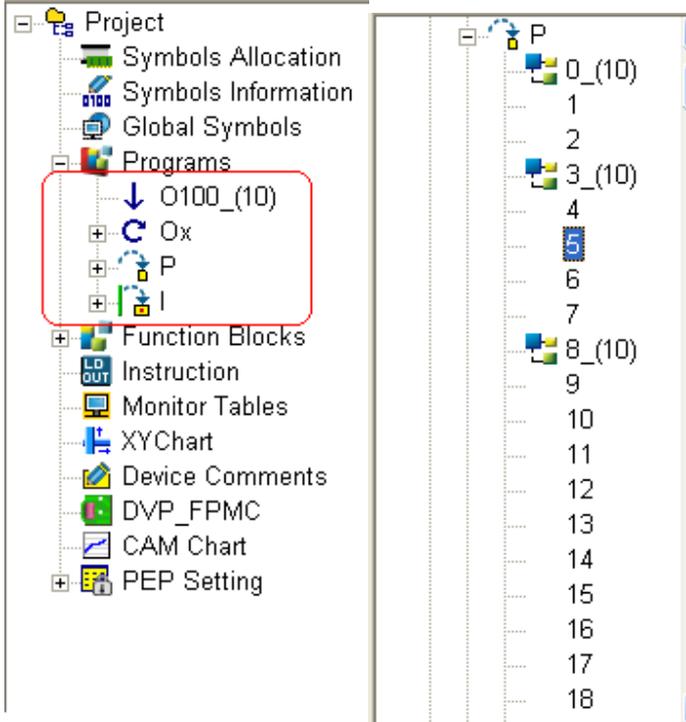
- 3. Unlock password: If the POU is protected by password, users need to enter the correct password for importing the POU file. If the file is not protected by password, system will ignore this step.



- 4. Confirm: Because importing POU(s) will modify current programs, a Confirm window will be displayed to ask for further confirmation. Click OK to import the program POU.



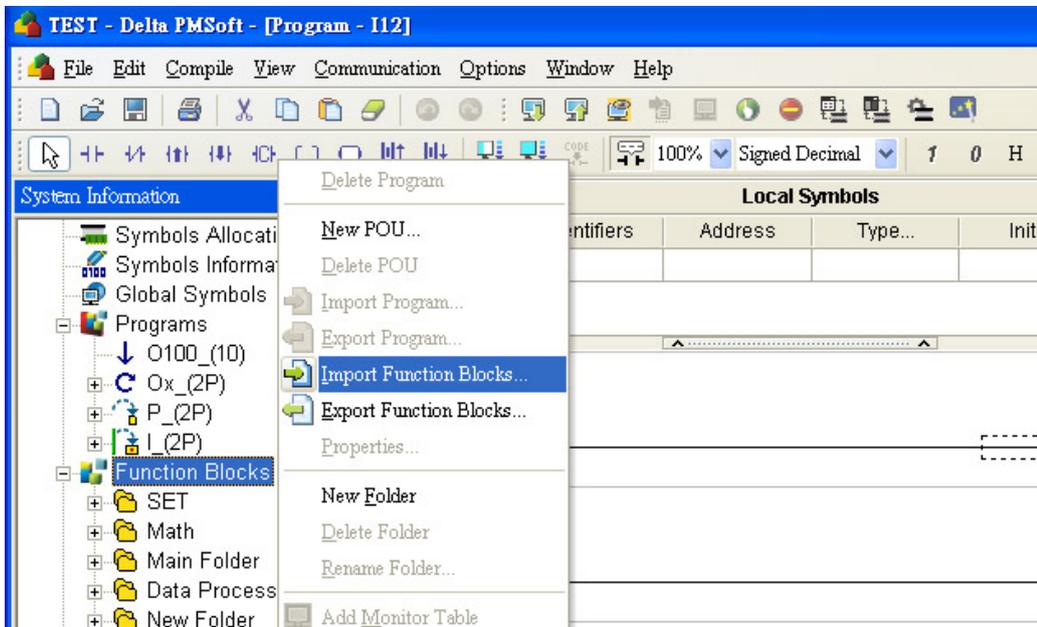
- 5. Results: The imported program POUs are circled as below.



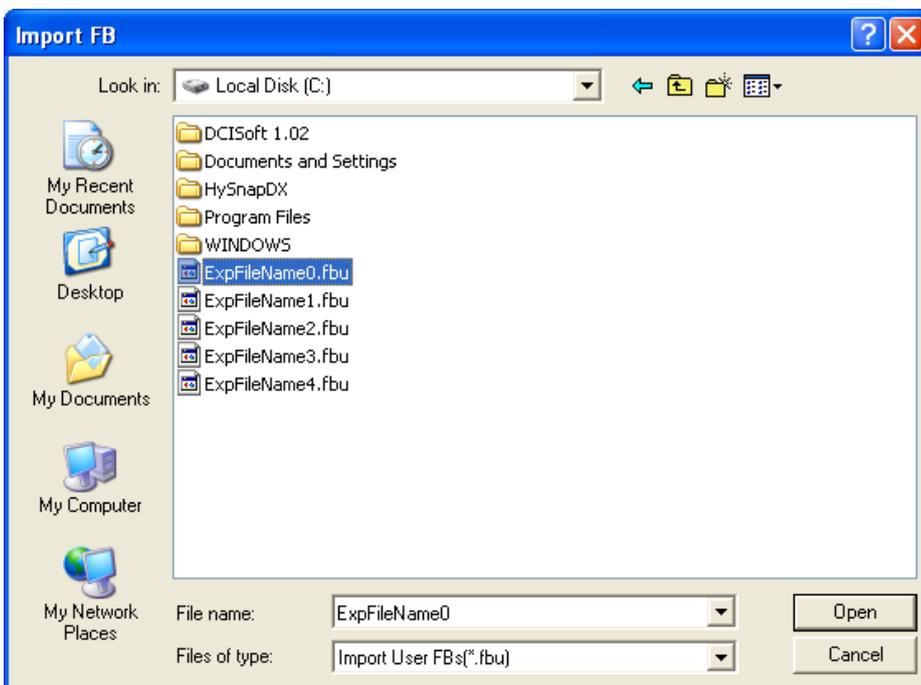
12 POU Editing Mode

■ Import function block POU:

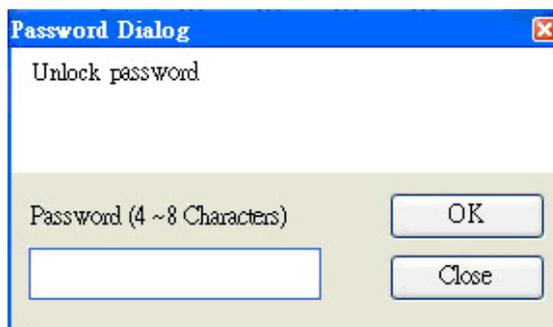
1. Right click at the node of “Function Blocks” under “System Information” and select “Import Function Blocks.”



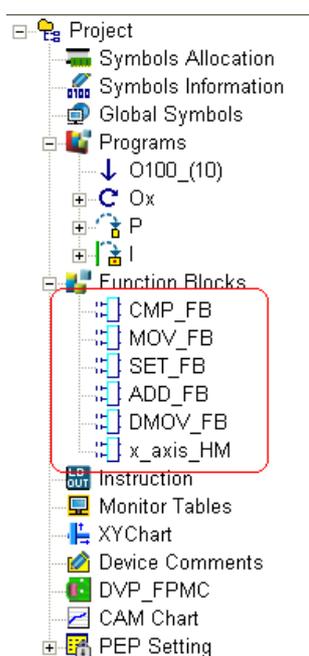
2. PMSOFT will display the function block files with the extension name “*.fbu”. Select a file and click Open to import this file.



3. Unlock password: If the function block POU is protected by password, users need to enter the correct password for importing the POU file. If the file is not protected by password, system will ignore this step.



- Results: the imported function blocks will be displayed as below.

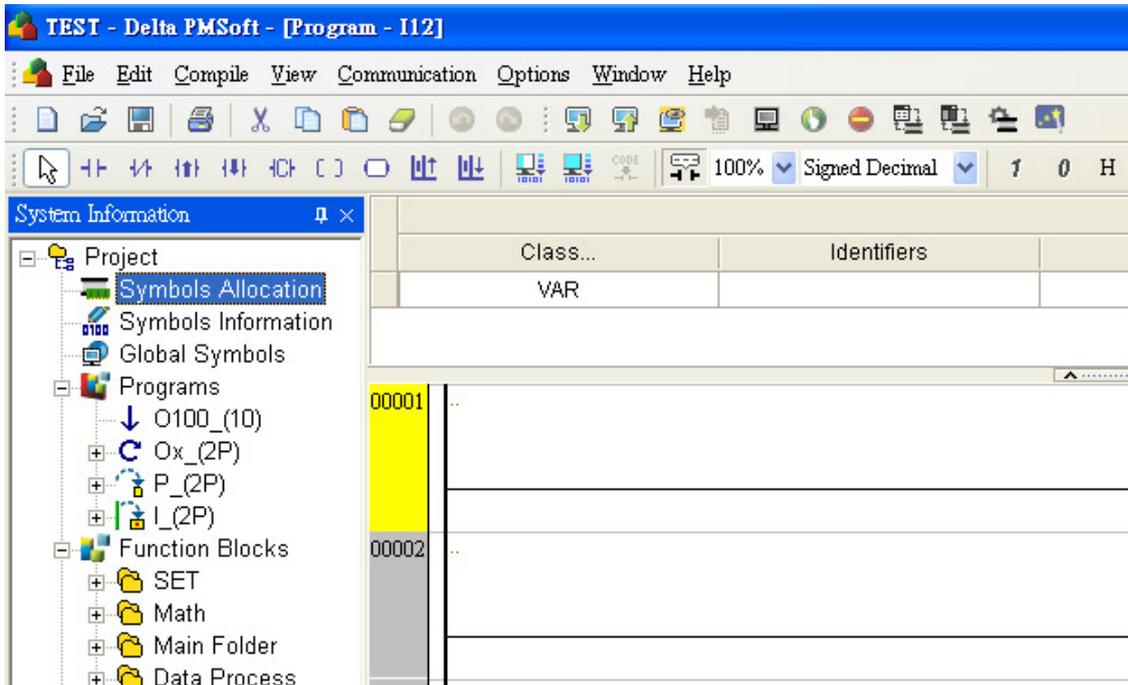


12.1.7 Symbols Allocation

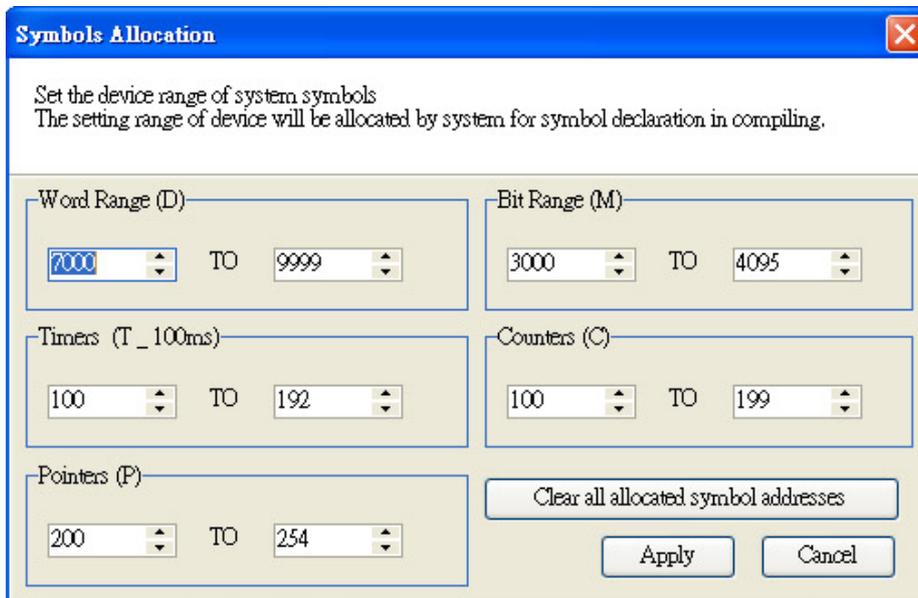
In Symbols Allocation window, users can set up the usable range of devices including data registers(D), auxiliary relays(M), timers(T), counters(C) and pointers(P) for system symbols. The setting range of device will be allocated by system for symbol declaration in compiling. The ways of setting device range are explained below.

- Double-click "Symbols Allocation" under "System Information" as below.

12 POU Editing Mode



2. Symbols Allocation window will pop up as below.

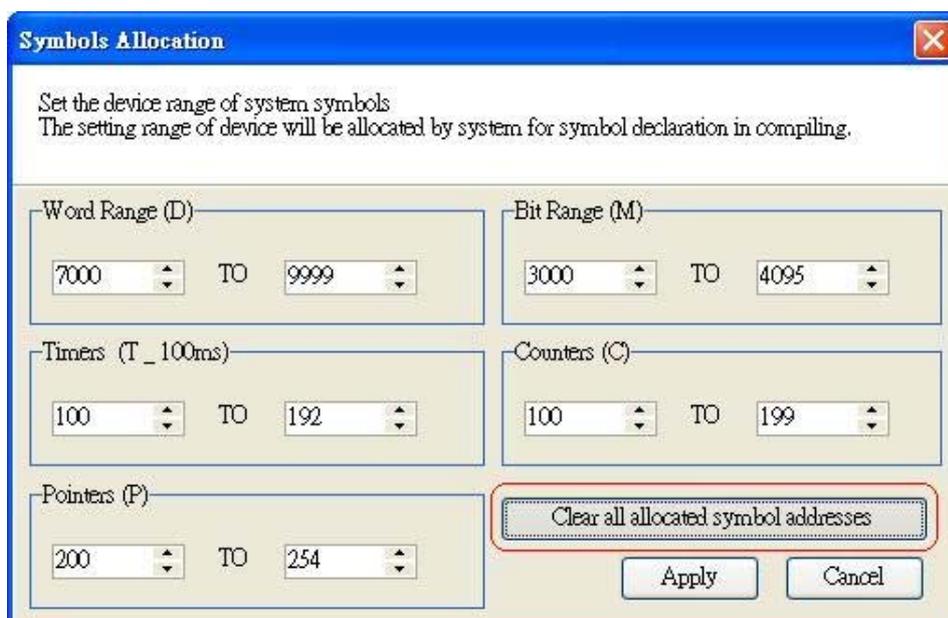


Set the usable range of the following devices:

- Data register(D): Default: D7000~D9999. Valid range: D0 ~ D9999. Care should be taken that special D registers should not be specified within the set range.
- Timers(T): Default: T100~T192. Valid range: T0 ~ T255.
- Counters(C): Default: C100~C199. Valid range: C0 ~ C255 ◦

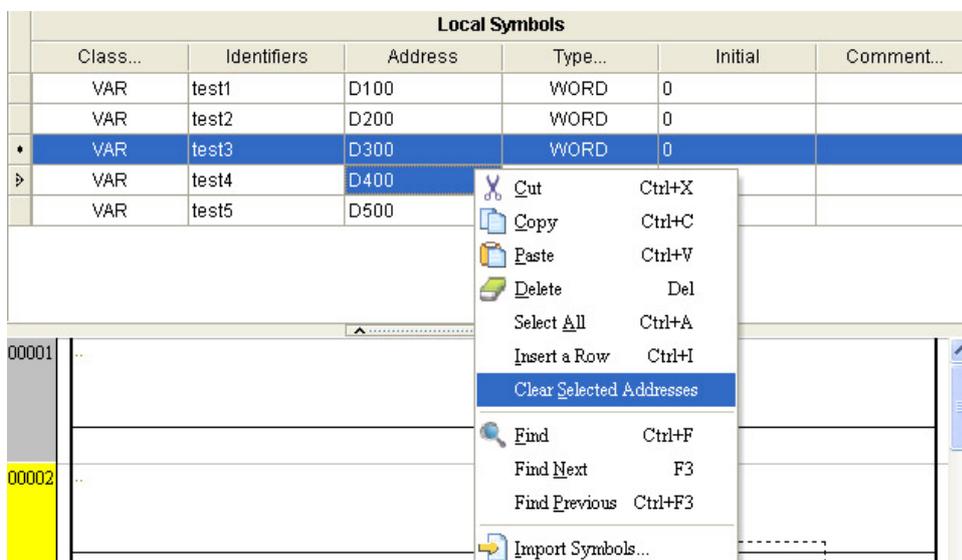
12 POU Editing Mode

- Auxiliary Relays(M): Default: M3000~M4095. Valid range: M0 ~ M4095 Care should be taken that special M relays should not be specified within the set range.
 - Pointers(P): Default: P200~P254. Valid range for 20PM00M: P0 ~ P255. Valid range for 20PM00D: P0 ~ P254.
3. Clicking “Clear all allocated symbol addresses” will reset all address values in global symbol table and local symbol table.



4. To clear the symbol address, select the symbols to be cleared and right click to select “Clear Selected Address”.

The address values of the selected symbols will be cleared.



12 POU Editing Mode

5. In the table below, address values of test3 and test4 are cleared.

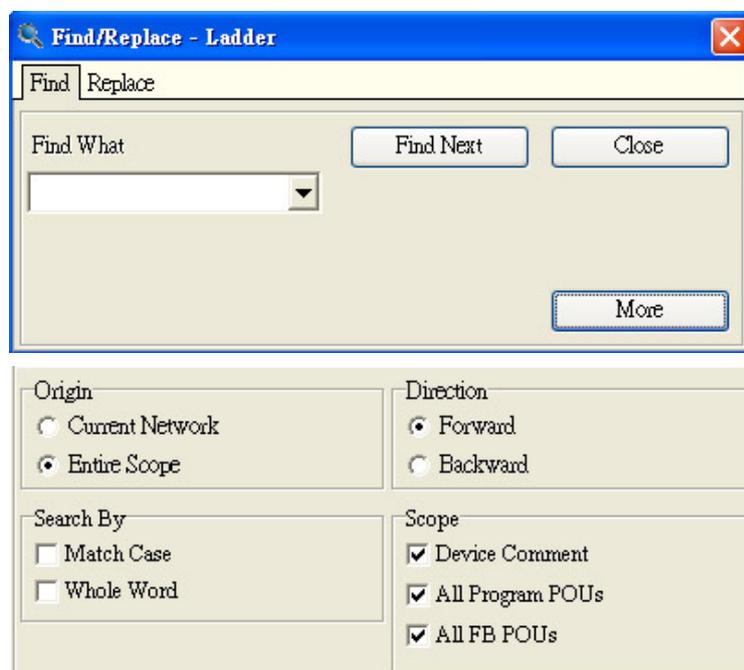
Local Symbols						
	Class...	Identifiers	Address	Type...	Initial	Comment...
	VAR	test1	D100	WORD	0	
	VAR	test2	D200	WORD	0	
•	VAR	test3		WORD	0	
▶	VAR	test4		WORD	0	
	VAR	test5	D500	WORD	0	

12.1.8 Ladder Find

There are 2 types of Find function in PMSOft including “Ladder Find” and “IL Find.” Ladder Find function is similar to general IL find function which is explained in “IL Find” in chapter 4. To find instructions, devices or symbols in POU ladders or Symbol Tables, Ladder Find function is required.

1. Find / Replace – Ladder

Right click at current POU ladder diagram and select Find from the drop-down list. You can find instructions, devices or symbols by this function. Matched results, if any, will be indicated. You can click “More” at the bottom right corner of Find window for more search options as below.

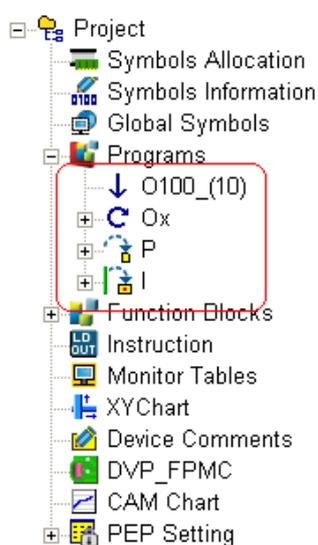


Available options for searching:

- Current Network: The searching starts from the current network section you are editing.

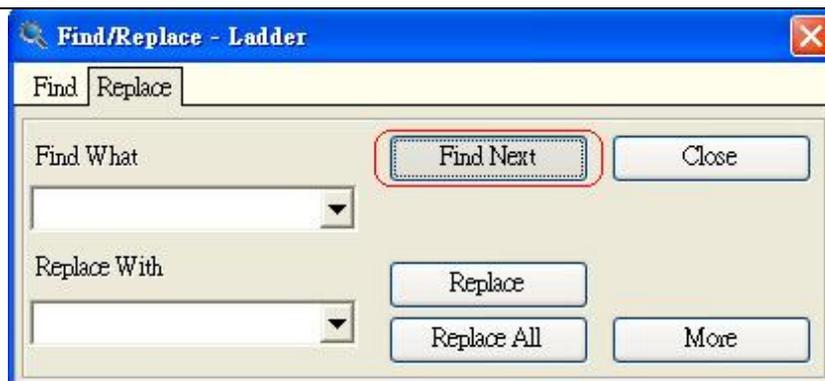
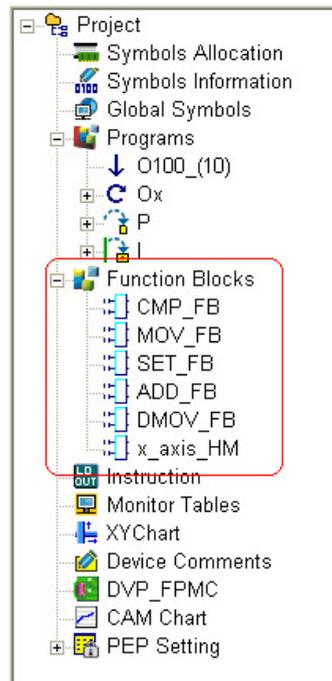
12 POU Editing Mode

- Beginning: The searching starts from the beginning of the window you are currently editing.
- Match Case: The instruction or device name you are looking for has to match exactly the capital or small letters you enter.
- Whole Word: The instruction or device name you are looking for has to match the whole string of words you enter.
- Forward: The searching is conducted from top to down.
- Backward: The searching is conducted from down to top.
- Device comment: The searching is only conducted in device comment areas.
- All Program POUs: Available range for searching includes O100, Ox0~Ox99 and P0-P255.



- All FB POUs: search all user-defined function blocks.

12 POU Editing Mode



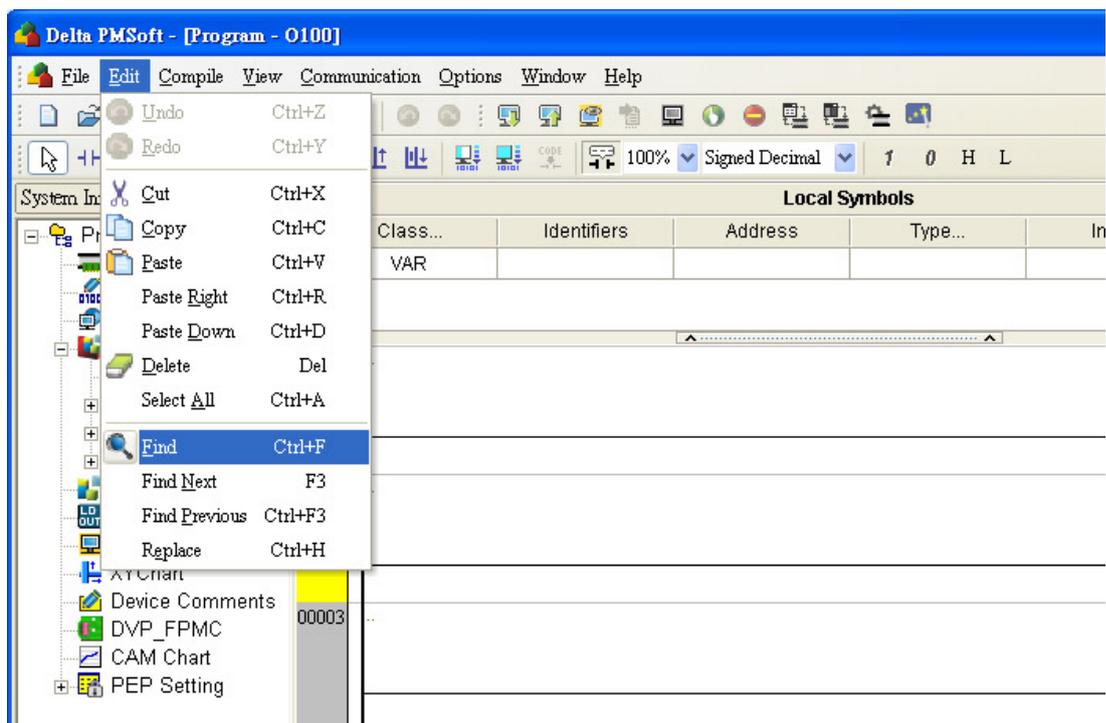
After search conditions are set, click “Find Next” to start the searching. The found item will be highlighted. Click “Find Next” again, and the searching will jump to the next found item. If there are no matched results, “Not found” window will appear, indicating there is no other item found.



2. How to open Find window in POU ladders:

Method 1: Select “Find” in the Edit menu

12 POU Editing Mode



Method 2: Right click at current POU ladder diagram and select Find from the drop-down list.

Method 3: Press Ctrl + F on the keyboard.

3. How to conduct Find Next function:

Find Next function is based on the last item found and looks downward for the next item in the ladder diagram.

The searching is conducted according to the search options below.

Origin <input type="radio"/> Current Network <input checked="" type="radio"/> Entire Scope	Direction <input checked="" type="radio"/> Forward <input type="radio"/> Backward
Search By <input type="checkbox"/> Match Case <input type="checkbox"/> Whole Word	Scope <input checked="" type="checkbox"/> Device Comment <input checked="" type="checkbox"/> All Program POU's <input checked="" type="checkbox"/> All FB POU's

Method 1: Select "Find Next" in the Edit menu.

Method 2: Press "F3" on the keyboard.

Method 3: Right click the mouse and select "Find Next" from the drop-down list.

4. How to conduct Find Previous function:

12 POU Editing Mode

Find Previous function is based on the last item found and looks upward for the next item in the ladder diagram.

The searching is conducted according to search options.

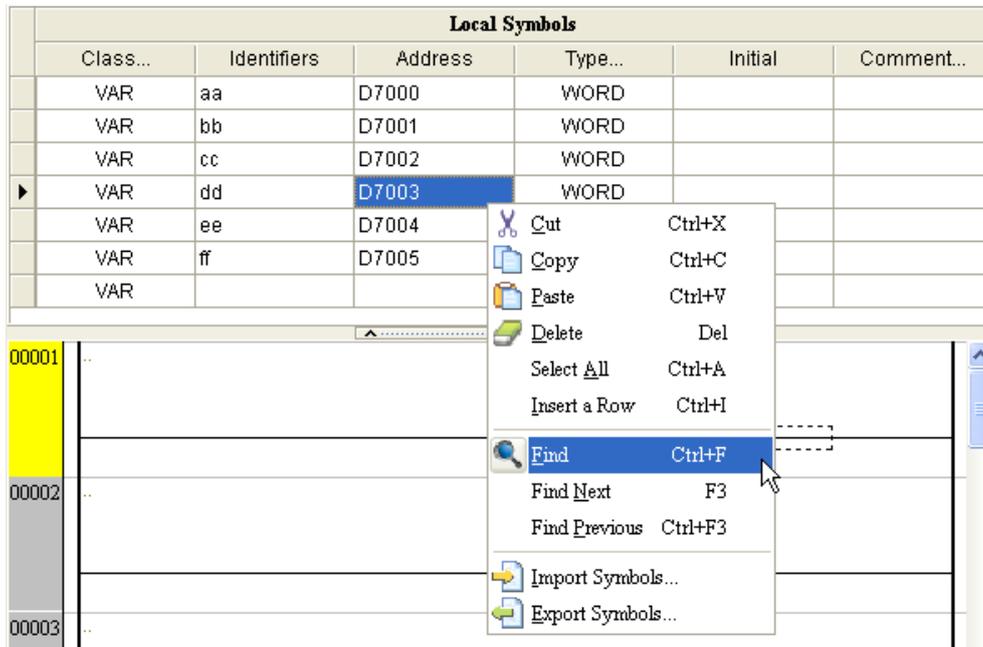
Method 1: Select “Find Previous” in the Edit menu.

Method 2: Press Ctrl + F3 on the keyboard.

Method 3: Right click the mouse and select “Find Previous” from the drop-down list.

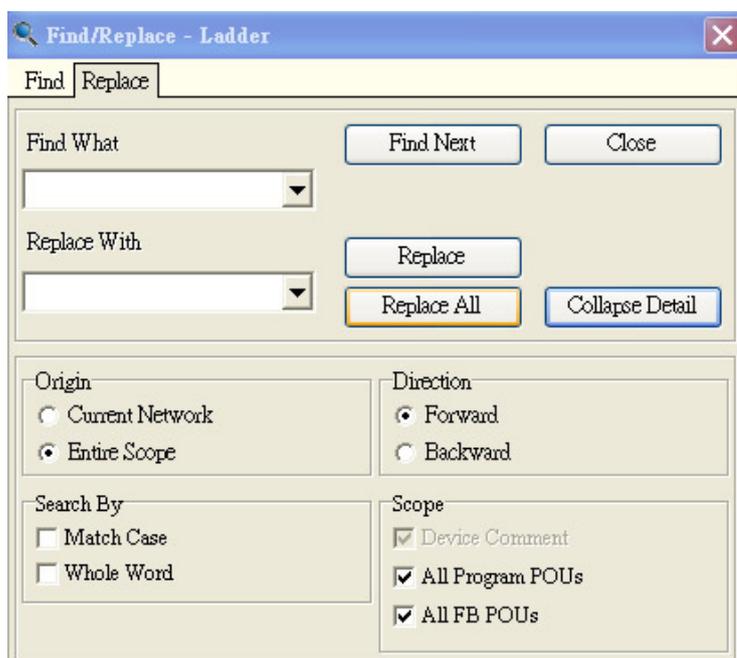
5. Find / Replace – Local Symbols

In addition to searching in POU ladder diagram, you can also search devices and symbols in Local Symbol Table as well as Global Symbol Table by right-clicking the symbol table and select Find from the drop-down list. However, the searching is conducted only at the symbol table which is right-clicked by user. To search other symbol tables you need to move the cursor and right click on them again.



12.1.9 Ladder Replace

Replace function can be conducted only on instructions, devices or symbols in POU ladders. Symbol tables are not supported by Replace function. In addition, replace function has to be used together with Find function. Replace function works only when matched results are found by Find function. Care should be taken when using Replace or Replace All function. Please make sure no instruction error, illegal device or incorrect instruction combination will occur after replacement is completed. See below for the details of the Replace window.



- How to open Replace window in POU ladders:

Method 1: Select “Replace” in the Edit menu.

Method 2: Right click the mouse and select “Replace” from the drop-down list.

Method 3: Press Ctrl + F on the keyboard.

12.2 Edit POU Ladder Diagram

POU ladder diagram is partly different from general ladder diagram in programming interface. However, when users apply POU ladder diagram for performing certain operation, which can also be performed by general ladder diagrams, they will have the same instruction list (IL) after compiling and achieve the same results as well. POU ladder diagram provides two advanced functions comparing with general ladder diagrams: 1. Replacing devices with symbols; 2. Applying POU function blocks.

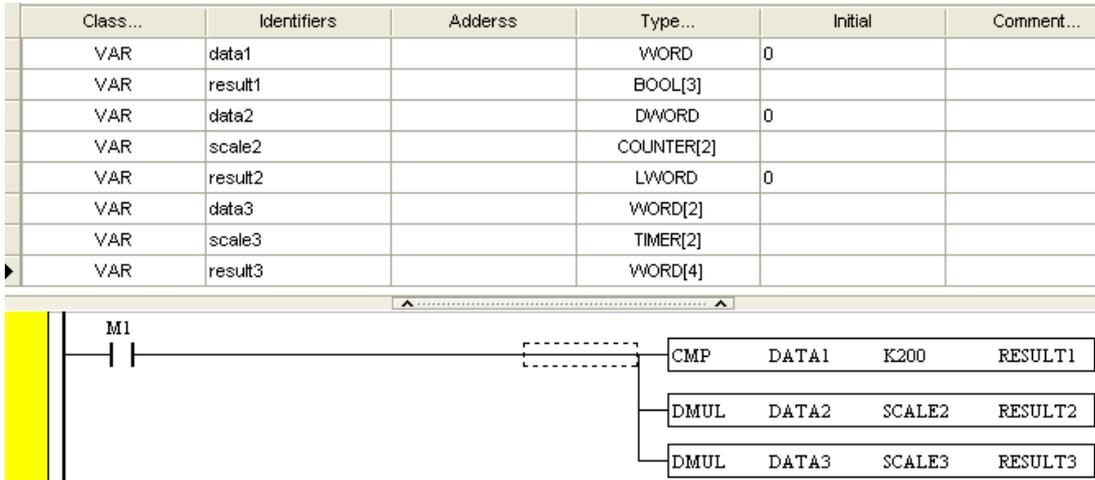
12.2.1 Replace Devices with Symbols

All devices in O100 main program, OX motion subroutines and Pn pointer subroutines can be replaced by symbols declared in local symbol table or global symbol table. Users should pay attention to the following rules when replacing devices in program.

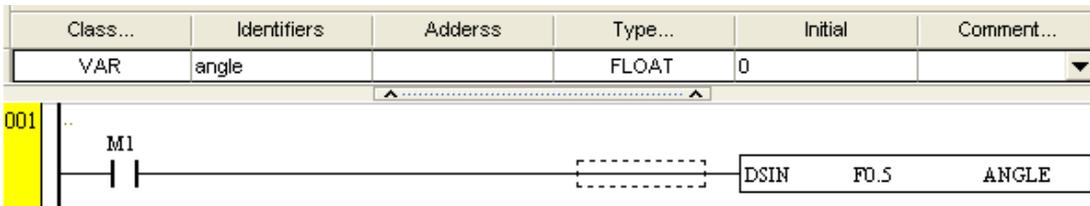
- Type of symbols should be specified in coordination with the limitation of operands in instruction. For example, the symbol “data2” should be specified according to the valid range of the instruction DMUL DATA2 SCALE2

12 POU Editing Mode

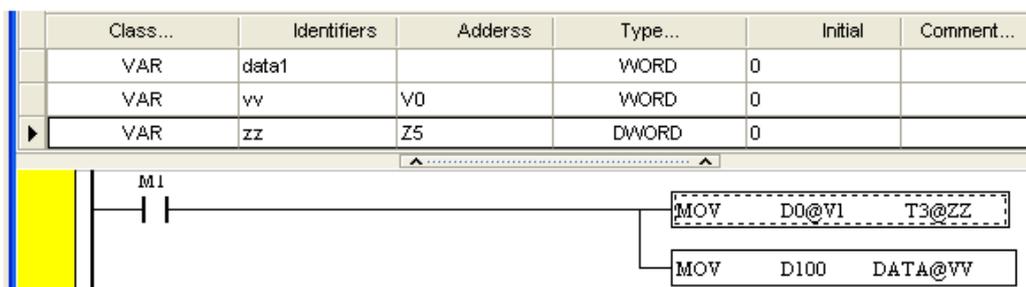
RESULT2. DATA2. In this case, only 32-bit data type including DWORD, WORD[2], TIMER[2], COUNTER[2] can be specified for “data2” as below. This rule applies to other symbols and instructions. With correct combinations between symbols and instructions, PMSOft can allocate proper devices when compiling.



- For floating point instructions, the operand D should be specified with a symbol declaring data type as FLOAT. Take the below diagram for example, The symbol “angle” in floating point instruction DSIN should be specified as FLOAT in Type column.

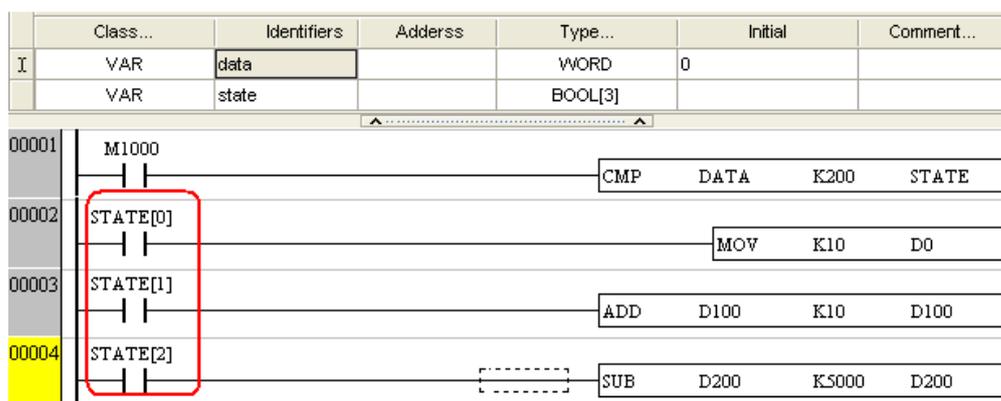


- Character “@” should be added when using indexes V or Z as below.

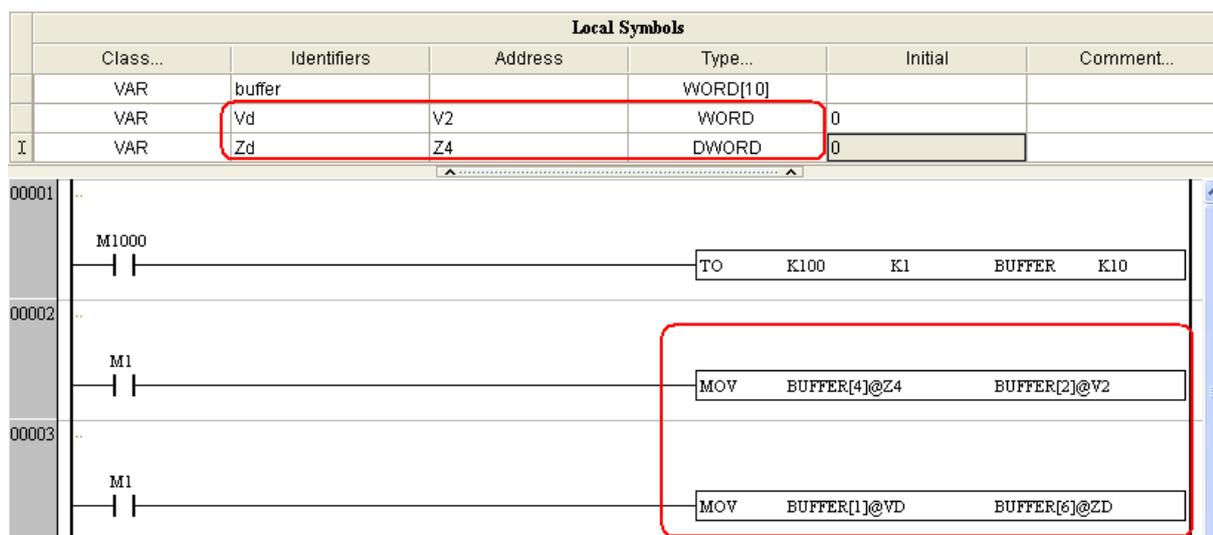


- Users can add index number in brackets “[]” after data type to apply designated data in the data string. For example, the last operand (STATE) of CMP instruction occupies 3 bits, so “state” in the symbol table should be declared as BOOL[3], creating 3 conditions including STATE[0], STATE[1] and STATE[3] for processing different operation results of CMP instruction.

12 POU Editing Mode



- Users can also attach index registers V, Z together with the index number in brackets “[]” for applying the designated data in a data string. For example, if data type of the symbol “buffer” is set as WORD[10], indicating a data string of 10 data, “buffer[m]@Vn” or “buffer[m]@Zn” (m: 0~9, n: 0~7) can be used for applying the data of index number “m+Vn” or “m+Zn” ($0 \leq m+Vn(Zn) \leq 9^*$). In this case, Vn and Zn can also be declared in symbol table. Application examples are demonstrated in below diagram. BUFFER[2]@V2, BUFFER [4]@Z4, BUFFER [1]@VD and BUFFER [6]@ZD of MOV instructions are used for applying different data in the data string of “buffer” with VD declared as V2 and ZD declared as Z2.



*Note: The data specified by index number plus index register (m+Vn, m+Zn) should not exceed the applicable range declared by data type in the symbol table.

12.2.2 Applying POU Function Blocks

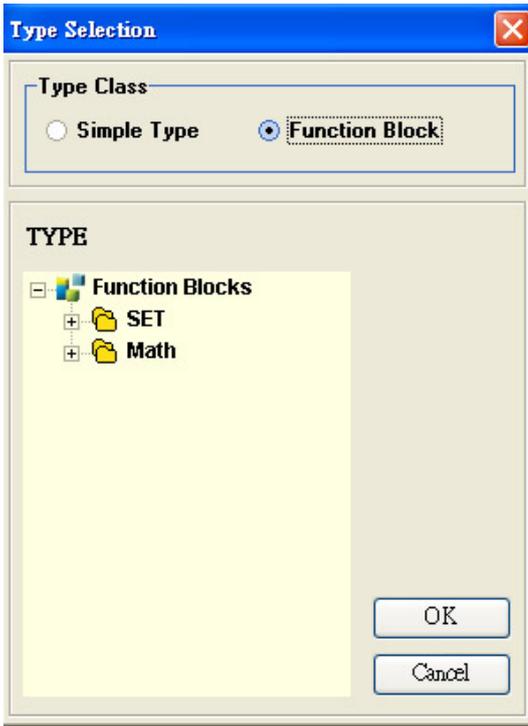
POU ladder differs from traditional ladder diagram in some way including the application of function blocks. Function blocks can be applied in POU ladder diagram and have some features that can help users achieve a more convenient programming process. The steps of applying POU function blocks and the features of POU ladders will be introduced below.

12 POU Editing Mode

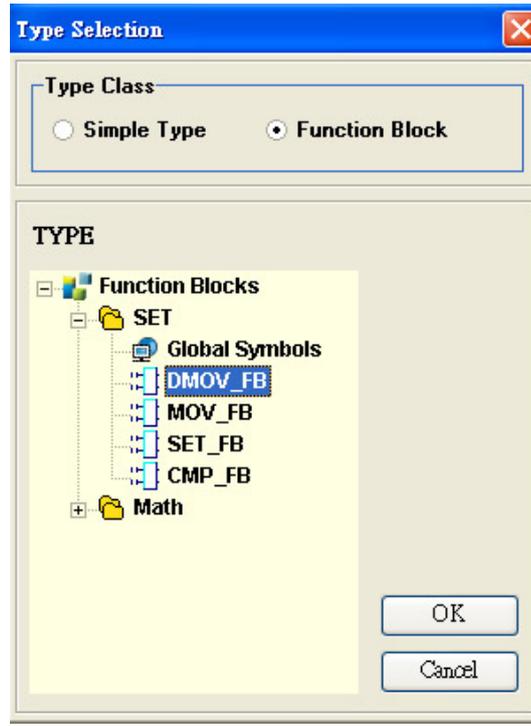
- Applying Function Blocks: there are 3 steps for using function blocks: declare, assign and edit the pin of FBs. The operation results are the same no matter declaration or assignment is performed first.

Step 1: To declare.

1: Click Type in symbol table and select Function Block as the Type Class.



2: Select the function block to be declared for the symbol



3: Results

Local Symbols						
	Class...	Identifiers	Address	Type...	Initial	Comment...
▶	VAR			DMOV_FB		

Step 2: To assign

There are 3 ways to assign the function blocks.

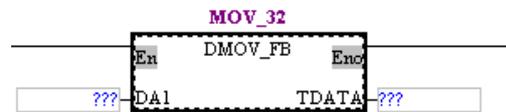
- Same as the input method for general instructions, input "FB name" + "." + "Identifier" (DMOV_FB.MOV_32)

12 POU Editing Mode

a. Enter DMOV_FB.MOV_32



b. Results

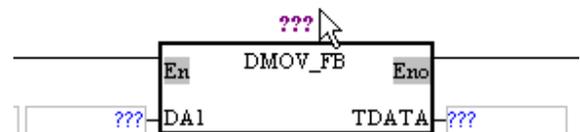


(2) Enter FB name (DMOV_FB) first. Click “???” and enter the identifier (MOV_32).

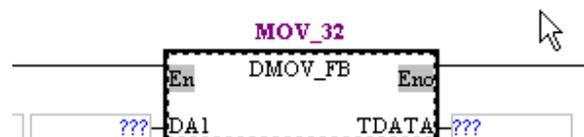
a. Enter DMOV_FB



b. Click “???” then enter the identifier MOV_32



c. Results

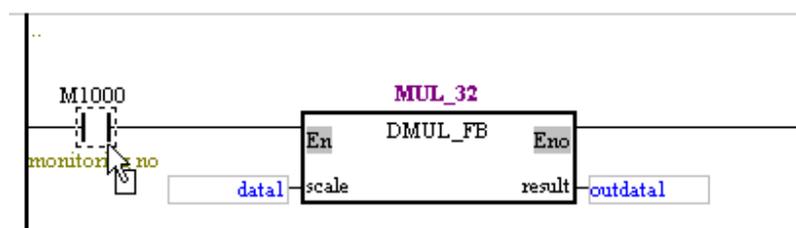


(3) Drag and drop. Click “???” and enter the identifier (MOV_32).

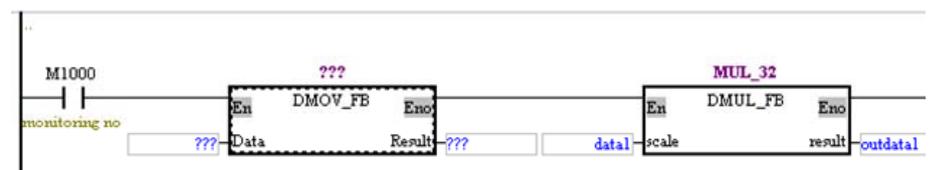
a.

Select the function block to be applied.

b. Drag the function block and drop at the desired location.

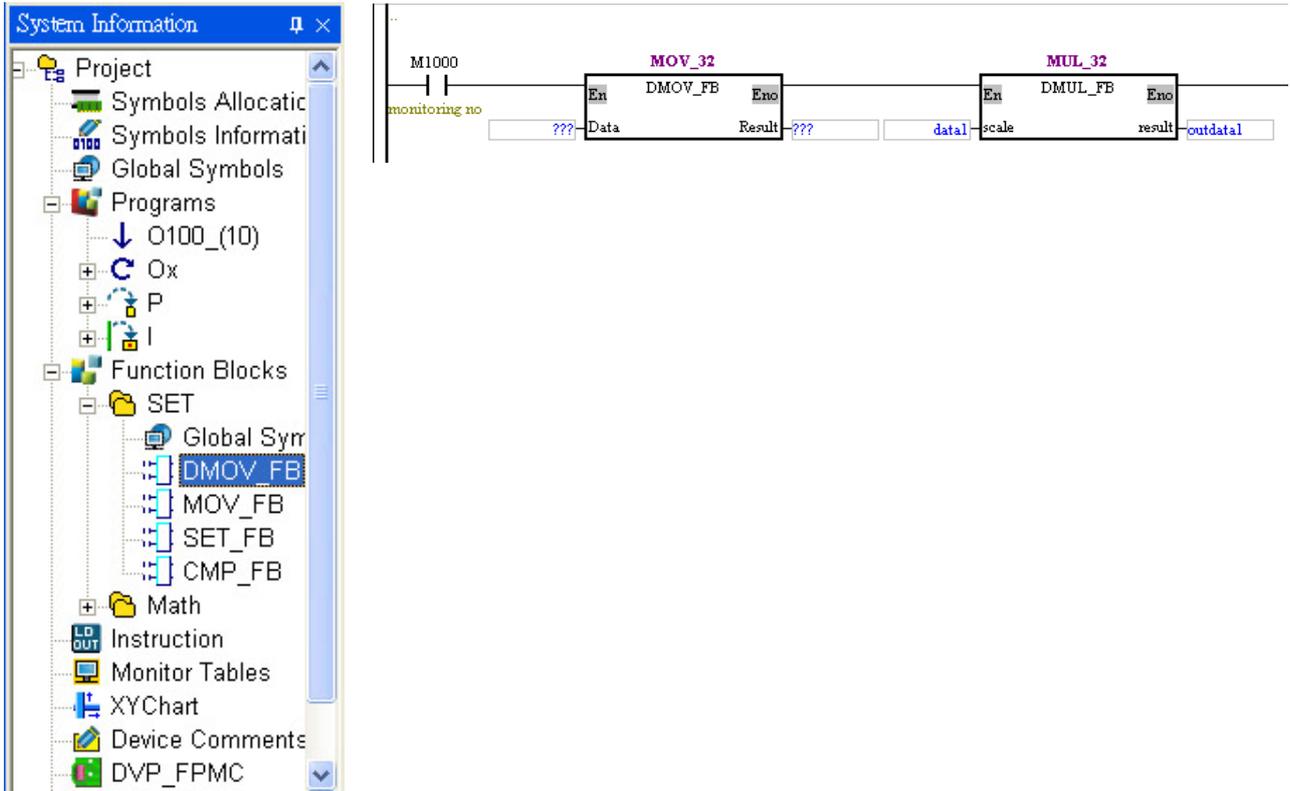


c. Click “???” and enter the identifier MOV_32



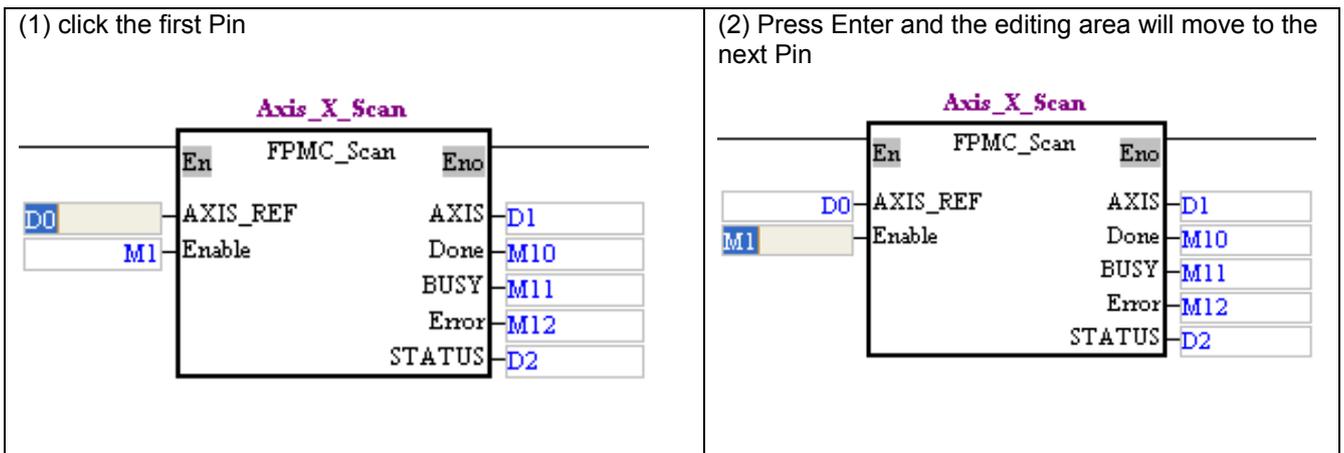
d. Results

12 POU Editing Mode



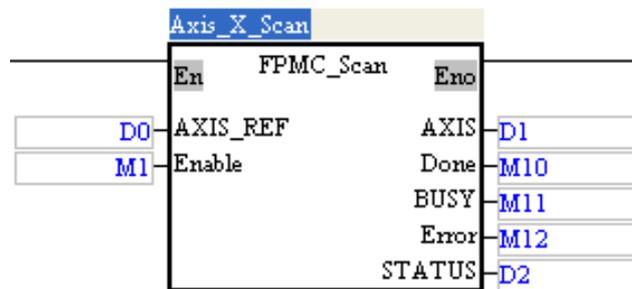
Step 3: Edit the Pin of function block

To enter the pin editing area, click the pin directly or click the function block then press Tab on the keyboard. Specify the value for the editing pin and press Enter. The editing area will move to next pin by the following order: input pins, individual name of current function block, output pins.

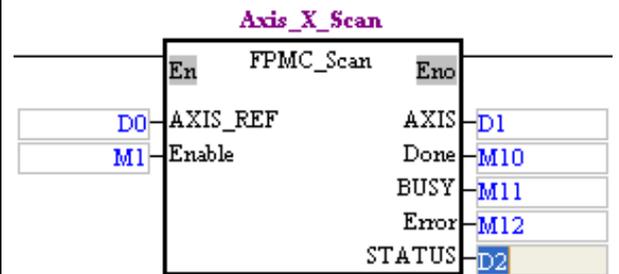


12 POU Editing Mode

(3) Press Enter again, the editing area will move to the individual name of current function block.

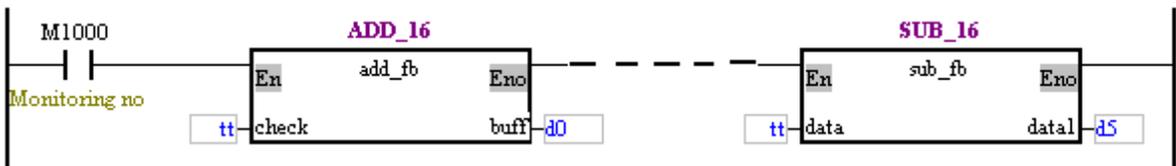


(4) The editing area will move to the last output Pin

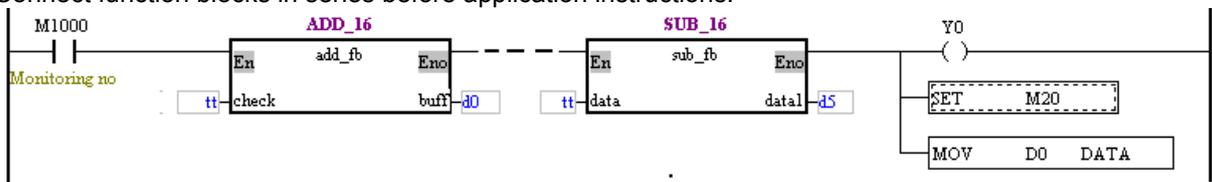


- Function blocks function as devices in ladder diagram. However, there are only two usages when applying function blocks in ladder diagram.

1. Connect function blocks in series.

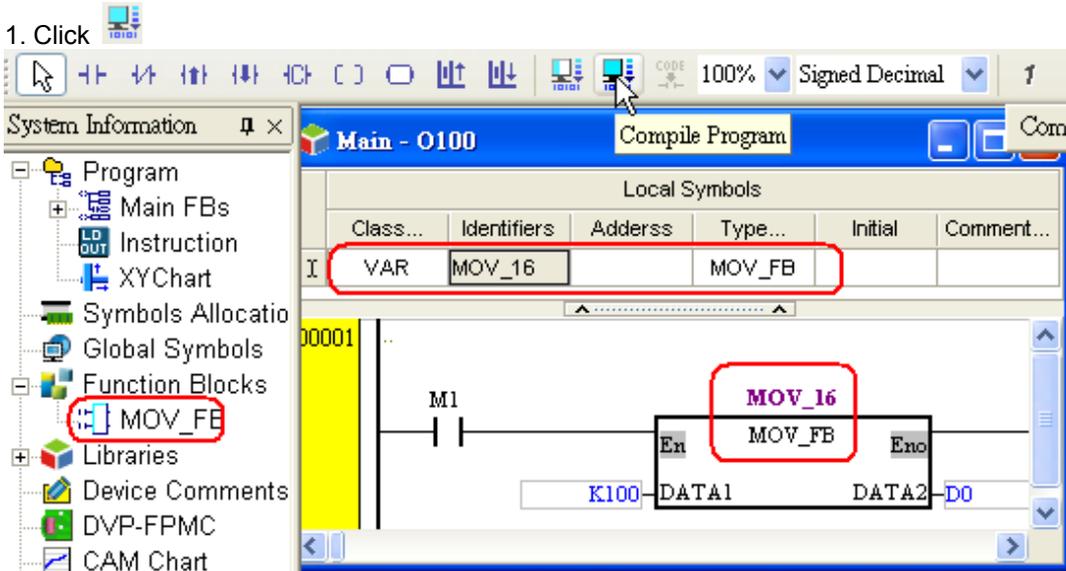


2. Connect function blocks in series before application instructions.

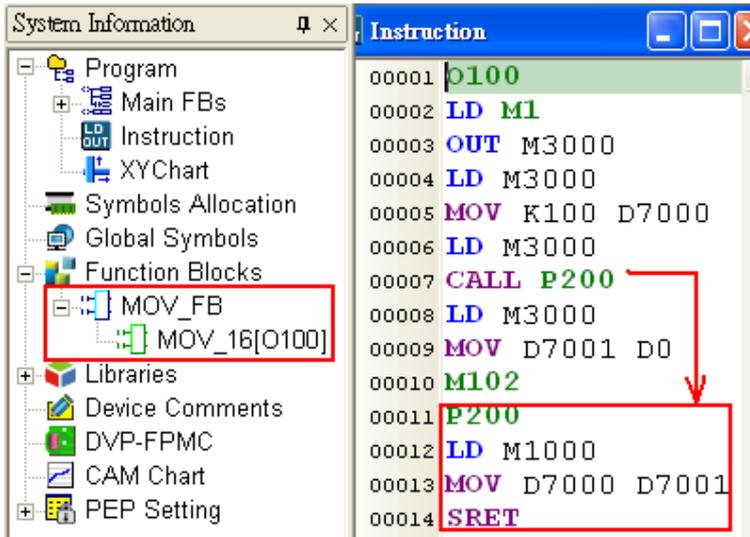


- Function block is regarded as Pointer Subroutine(Pn) by PMSOft. Therefore, the compiled results of function blocks in instruction list (IL) will be indicated by P pointers. The function block below is indicated as P200 in IL.

12 POU Editing Mode

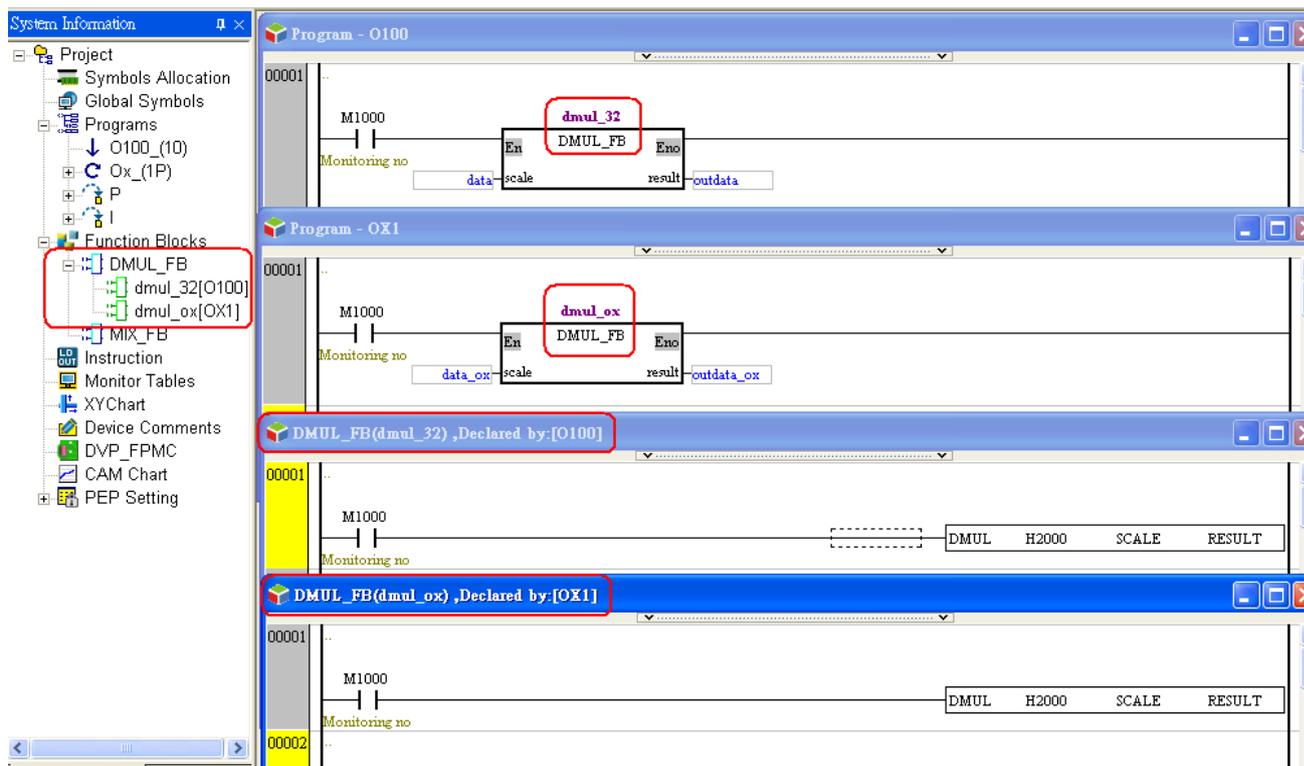


2. Results

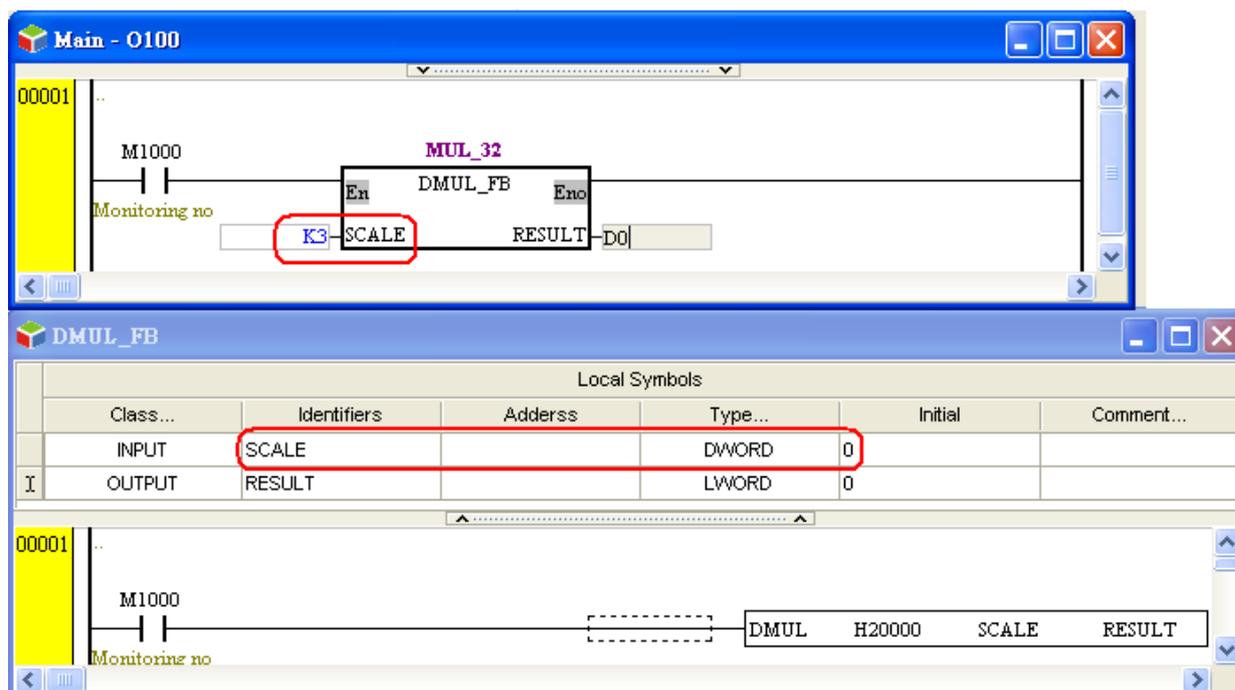


- Function blocks can be used repeatedly. Applying different symbols on the same function block generates different Pn subroutines and different instance diagrams after compiling. Function blocks and Pn subroutines should not be applied over the valid range of 256. Take below diagram as example, Function block DMUL_FB is applied in both O100 main program and OX1 motion subroutine while its symbol is declared in both global symbol table and OX1 local symbol table. Therefore, after compiling, 2 different subroutines with 2 pointers P will be generated as well as different instance ladder diagrams.

12 POU Editing Mode

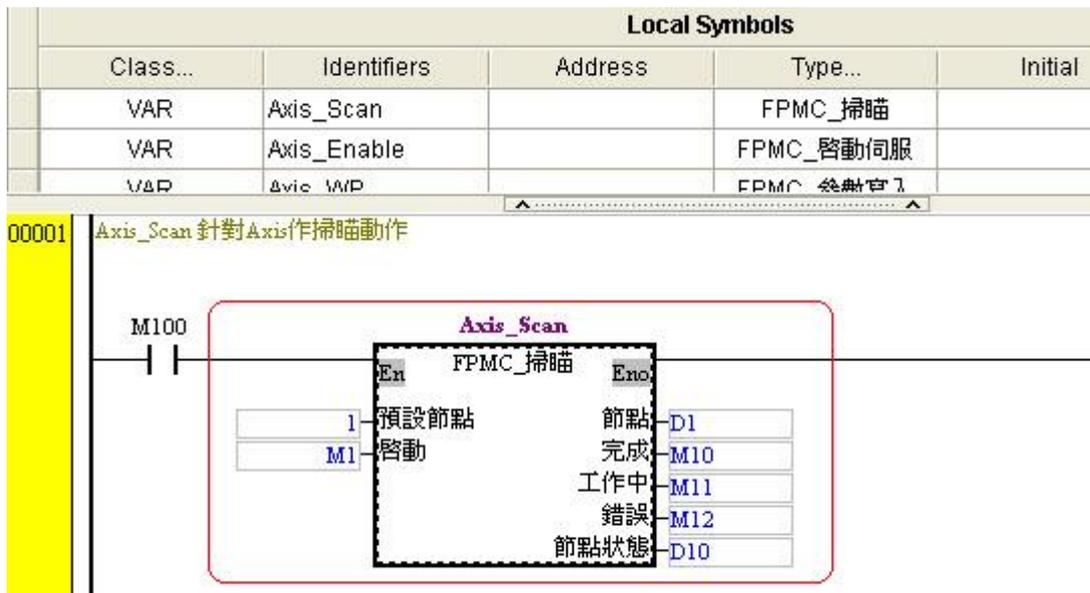


- The input/output data should be specified in coordination with input/output data type of the symbol, otherwise errors will occur after Compile  or Check  is clicked. However, when data type of input interface in function block is specified as WORD or DWORD, the input data allows constant K or H as below.

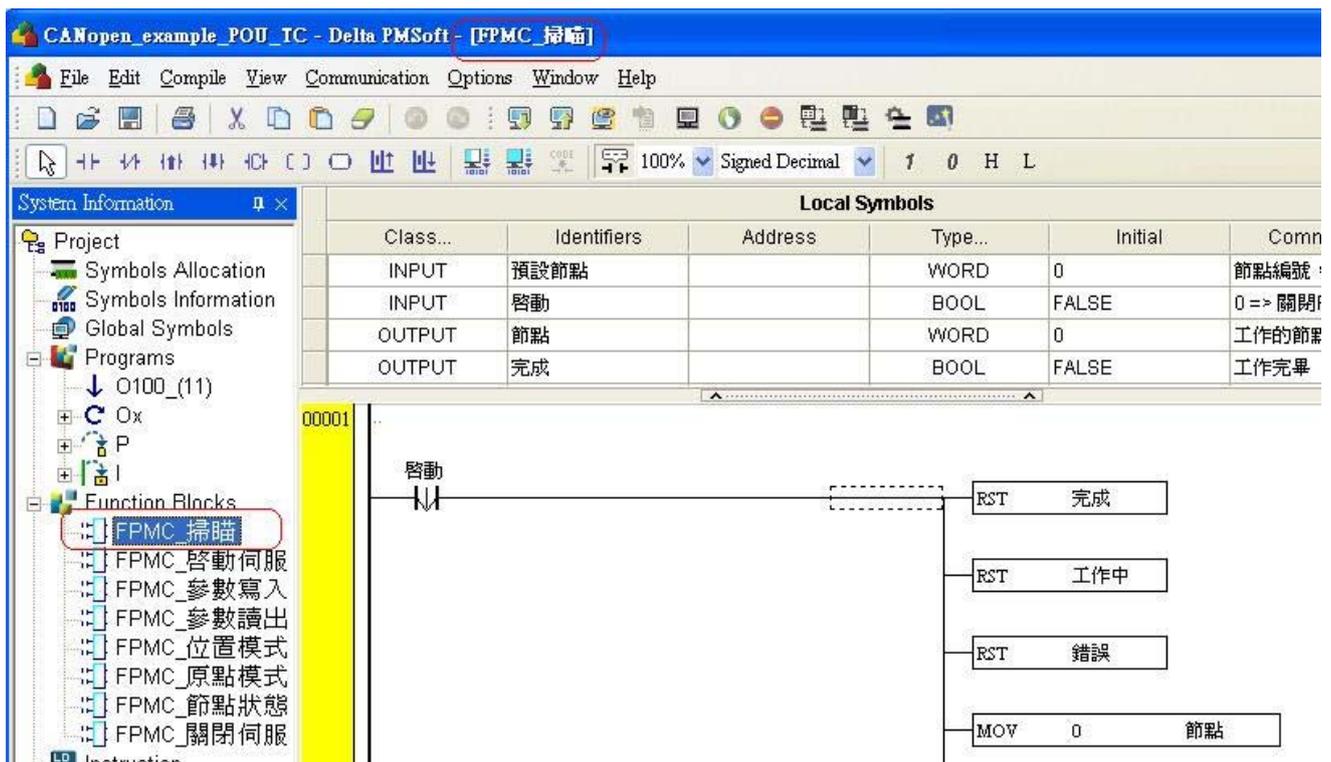


- Double click the function block directly and the editing window of the function block will be displayed.

12 POU Editing Mode



Results:

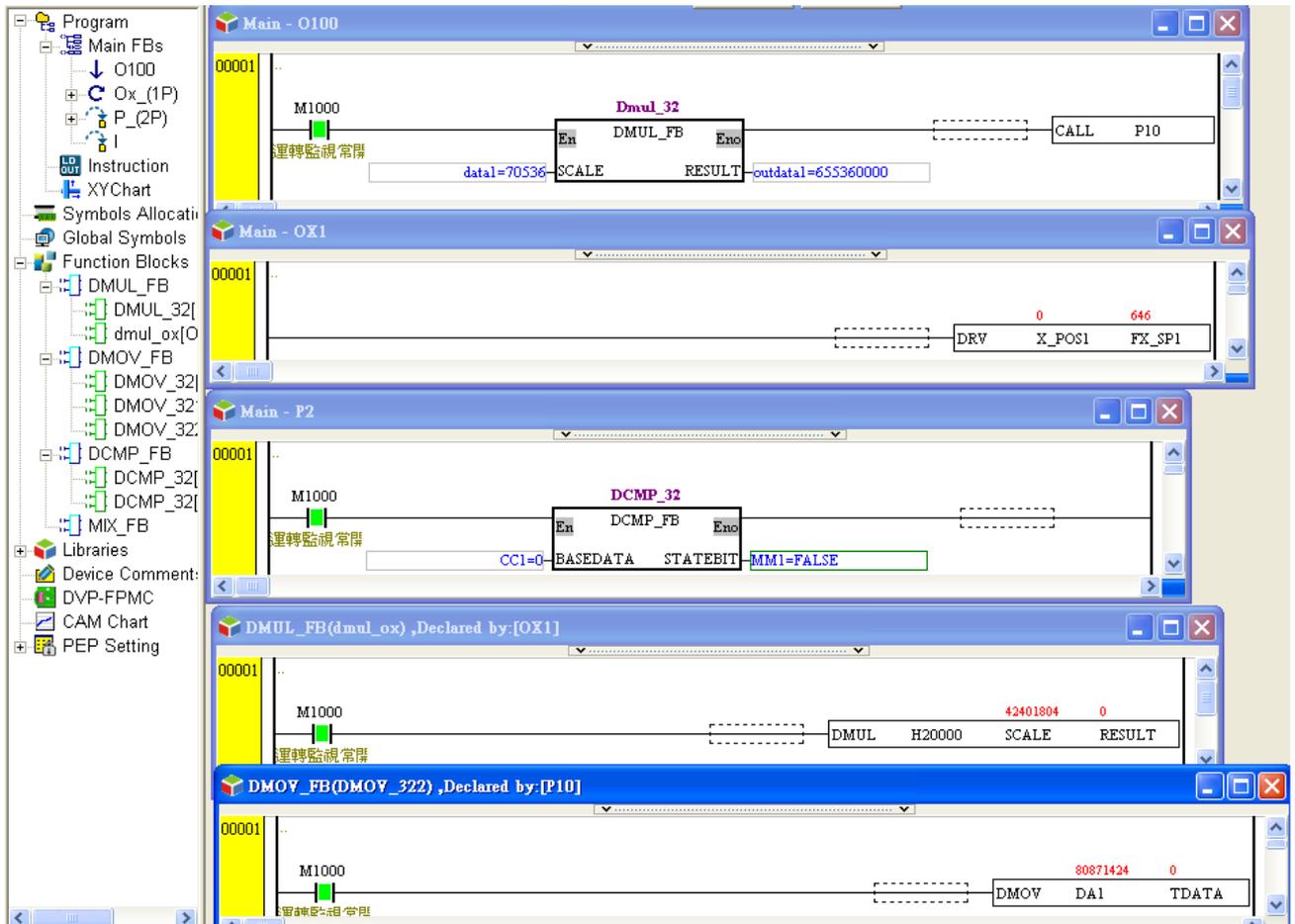


12.3 Monitor POU program

POU program can be directly modified in ladder diagram or monitor table under Monitoring mode.

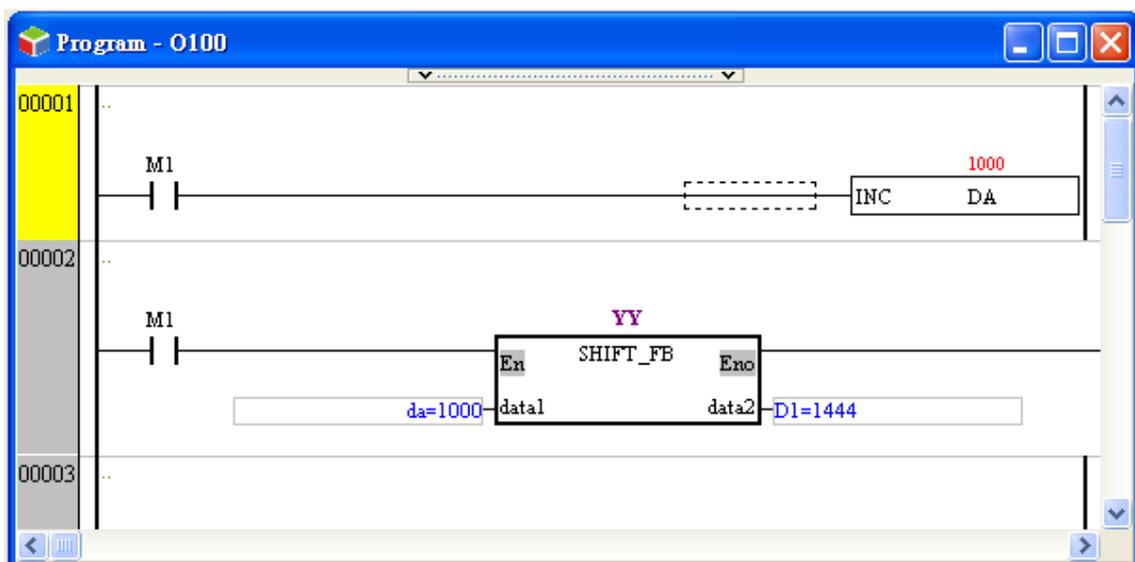
- Monitor ladder diagrams: click  and monitoring mode will be started to monitor O100 main program, P subroutines, OX subroutines and the instance ladder diagrams compiled by function blocks as below.

12 POU Editing Mode



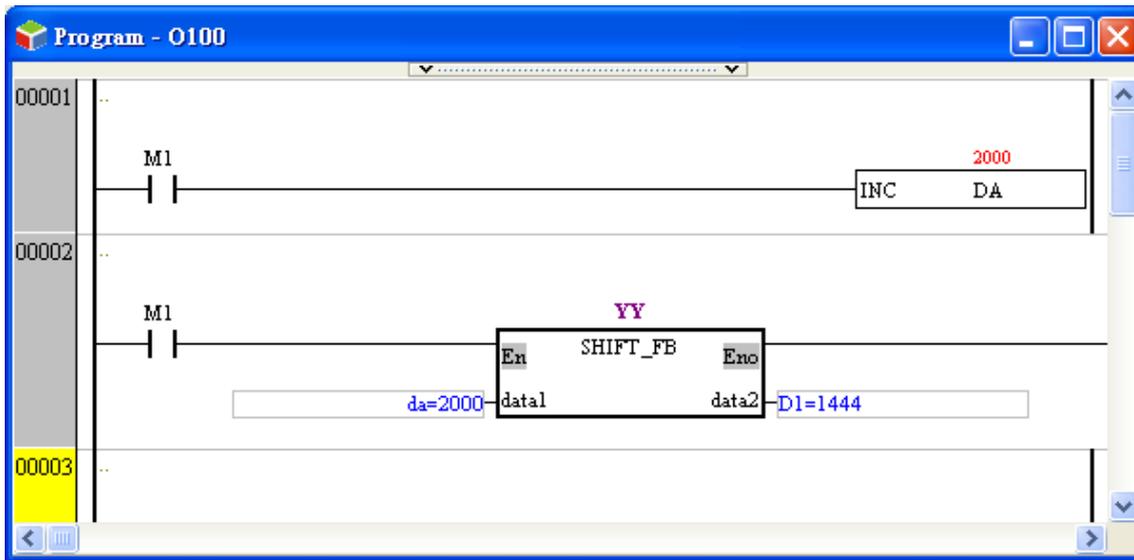
- Modify data in monitoring mode: Modify the DA value in O100 main program from 1000 to 2000 as below.

1: Click the input symbol (DA) of function block SHIFT and enter 2000.



2: The symbol DA displays DA = 2000.

12 POU Editing Mode



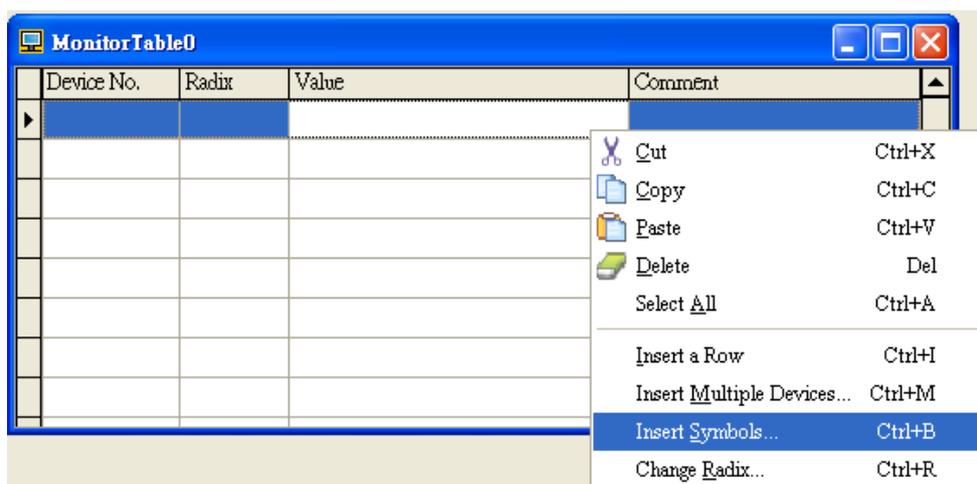
- Monitor by MonitorTable: Users can monitor symbols by Monitor Table. Click the column “Device No.” and input the symbol to be monitored in the format: Name of symbol table (O100, OXn, Pn, Global) + “.” + Name of symbol. In the diagram below, MonitorTable10 monitors the symbol TIMES in O100 main program and other symbols as well.

Device No.	Radix	Value	Comment
O100.TIMES	d16s	1	
OX2.X_POSIT	d32s	42336268	
P5.STOR	d16s	5000	
GLOBAL.BUFFER	d16s	5000	

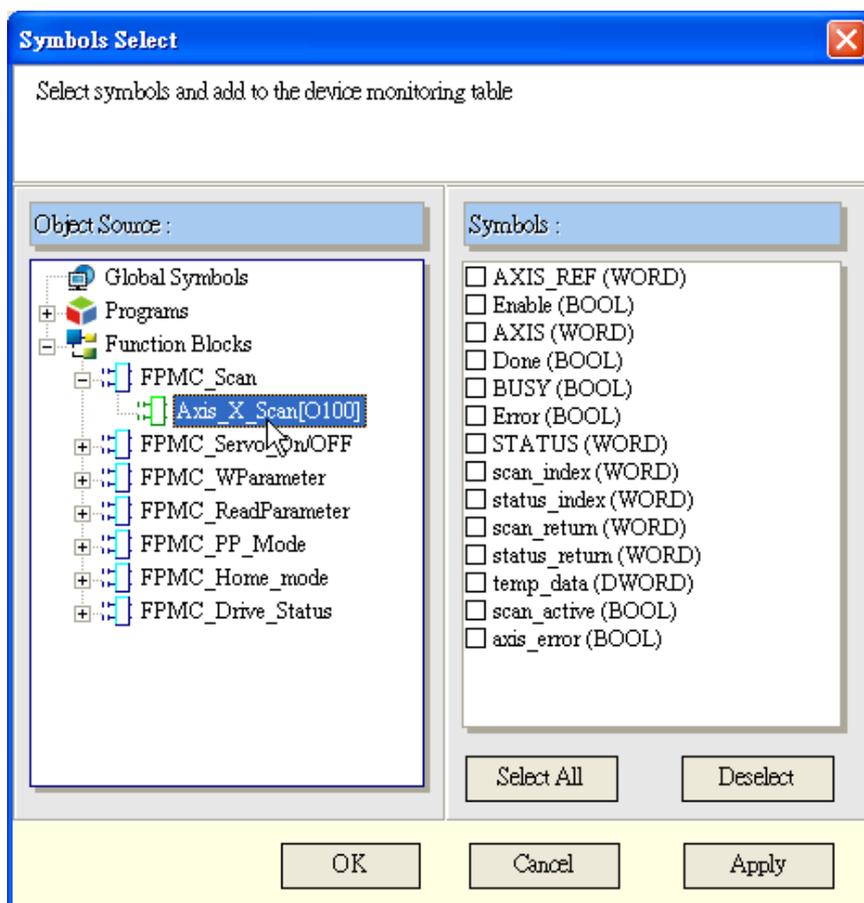
- Quickly Insert Symbols: Right click the monitor table and select Insert Symbols. Symbol Select window will pop up with all declared symbols in the project. Click the Object Source on the left and all symbols related to the selected function block will be listed in the Symbols column. Select the symbol to be inserted and click OK to add the symbols to the monitor table.

1: Right click at the monitor table and select Insert Symbols in the drop-down list.

12 POU Editing Mode

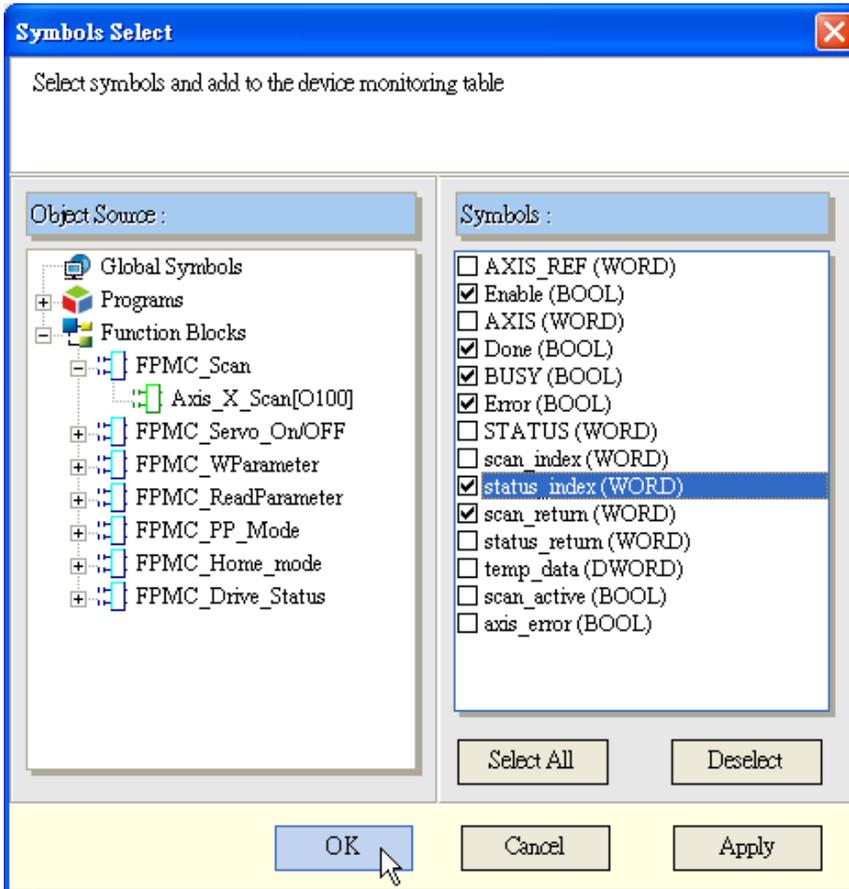


2: Click the program to be monitored (O100 main program, P subroutine, OX subroutine or function blocks) and the related symbols will be displayed in the Symbols column.



3: Click the symbol to be monitored directly or click Select All to select all symbols. After symbols are selected, click OK to finish.

12 POU Editing Mode



4: The selected symbols will be listed in the monitor table

Device No.	Radix	Value	Comment
0100.AXIS_X_SCAN.ENABLE	bit		
0100.AXIS_X_SCAN.DONE	bit		
0100.AXIS_X_SCAN.BUSY	bit		
0100.AXIS_X_SCAN.ERROR	bit		
0100.AXIS_X_SCAN.STATUS_INDEX	d16s		
0100.AXIS_X_SCAN.SCAN_RETURN	d16s		

12.4 Hint Function on Symbols and Function Blocks

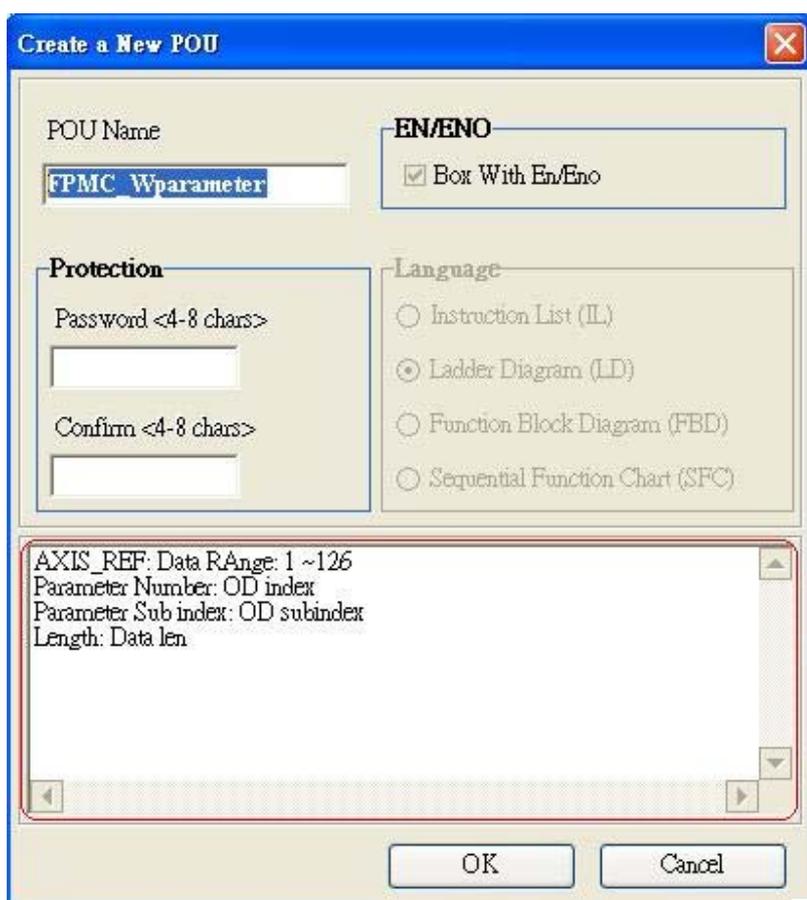
Users can display the information of symbols and function blocks by setting up Hint function. Before enabling Hint function, users need to enter relative information into the comment areas of Symbol Table(A) and Create a New POU window(B). When the cursor is moved to symbols or function blocks, the edited information such as current POU function, input/output interface definition, or data types of symbols will be displayed.

12 POU Editing Mode

(A) Comment Area of Local Symbol Table

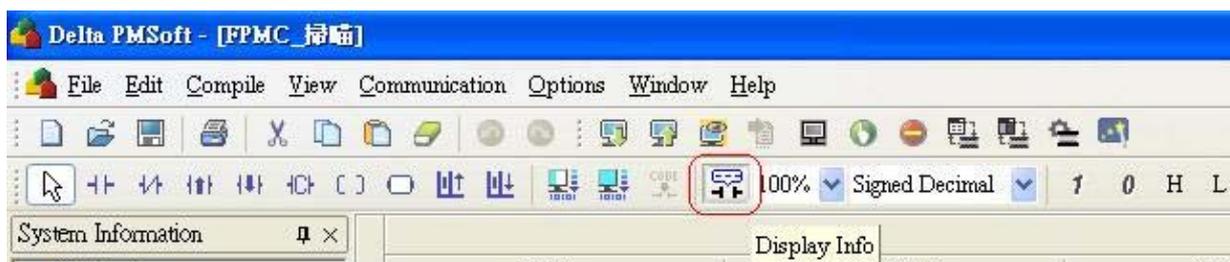
Local Symbols						
Class...	Identifiers	Address	Type...	Initial	Comment...	
INPUT	AXIS_REF		WORD	0	1~4	
INPUT	Enable		BOOL	FALSE		
OUTPUT	AXIS		WORD	0		
OUTPUT	Done		BOOL	FALSE		

(B) Comment Area of POU



- Display Info : Hint function can be enabled by clicking the icon  on toolbar「Diagram (A)」. Diagrams (B), (C), (D) indicate the information displayed on symbol, interface of function block and function block.

(A) Icon for Enabling Hint Function

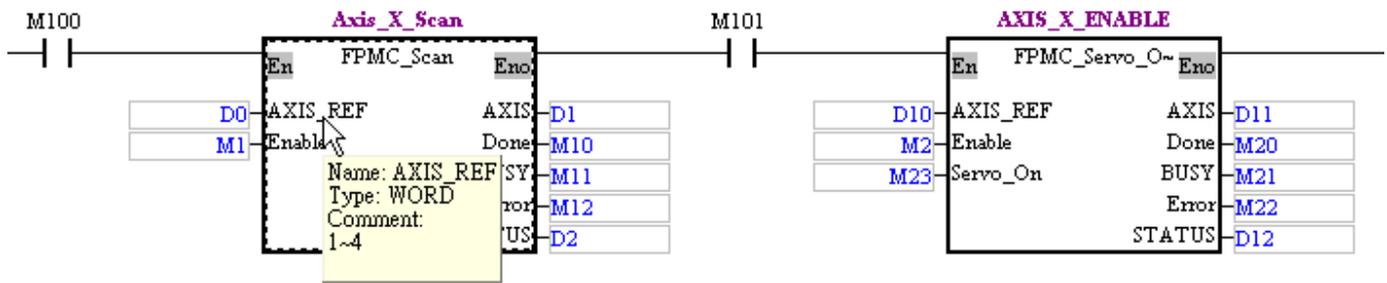


12 POU Editing Mode

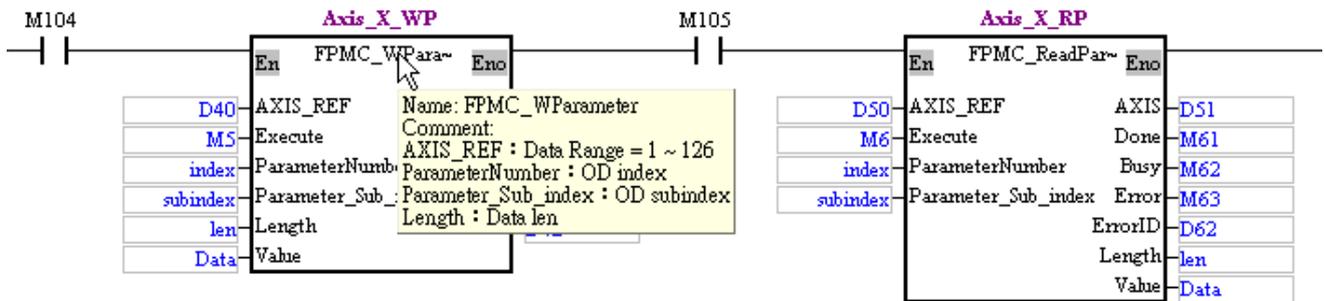
(B) Information of Symbol



(C) Information of Function Block Interface



(D) Information of Function Block



13 CANopen Communication Card

13.1 Introduction to DVP-FPMC: CANopen Communication Card

DVP-FPMC is a CANopen communication card for DVP-PM series PLC to conduct data exchange, featuring:

- ◆ Compliant with CANopen standard protocol, DS301 v4.02
- ◆ Supports NMT protocol
- ◆ Supports SDO protocol
- ◆ Supports CANopen standard protocol DS402 v2.0: Max. 4 motion axes
- ◆ The motion axes support Profile Position mode

13.2 Specifications

■ CANopen connector

Transmission method	CAN
Electrical isolation	500VDC
Type	Removable connector (5.08mm)
Transmission cable	2 communication cables, 1 shielded cable and 1 ground wire

■ Ethernet connector

Transmission method	Ethernet
Electrical isolation	500VDC
Type	Removable connector (5.08mm)
Transmission cable	2 communication cables, 1 shielded cable and 1 ground wire

■ Communication

Protocol type	PDO, SDO, SYNC (synchronous object), EMCY (emergency object), NMT, Heartbeat
Serial transmission speed	500k, 1M bps (bits per second)
Product code	254
Equipment type	0 (Non-Profile)
Company ID	477 (Delta Electronics, Inc.)

■ Electrical specification

Power supply voltage	Supplied by MPU through internal bus, 24VDC (-15% ~ 20%)
----------------------	--

13 CANopen Communication Card

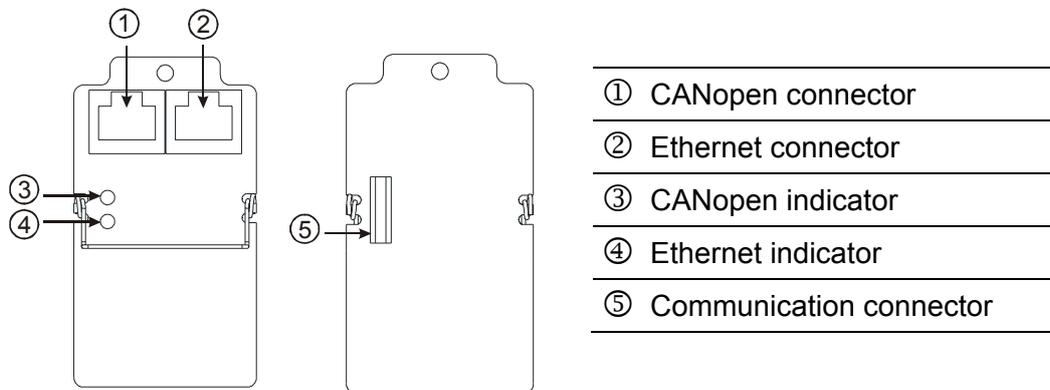
Power consumption	1.7W
Insulation voltage	500V
Weight (g)	66

■ Environment

Noise immunity	ESD (IEC 61131-2, IEC 61000-4-2): 8kV Air Discharge, 4kV Contact Discharge EFT (IEC 61131-2, IEC 61000-4-4): Power Line: 2kV, Digital I/O: 1kV Analog & Communication I/O: 1kV Damped-Oscillatory Wave: Power Line: 1kV, Digital I/O: 1kV RS (IEC 61131-2, IEC 61000-4-3): 80MHz ~ 1000MHz , 1.4GHz ~ 2.0GHz , 10V/m
Operation/storage	Operation: 0°C ~ 55°C (temperature), 50 ~ 95% (humidity), pollution degree 2 Storage: -25°C ~ 70°C (temperature), 5 ~ 95% (humidity)
Shock/vibration immunity	International standards: IEC 61131-2 · IEC 68-2-6 (TEST Fc)/IEC 61131-2 & IEC 68-2-27 (TEST Ea)
Standards	IEC 61131-2

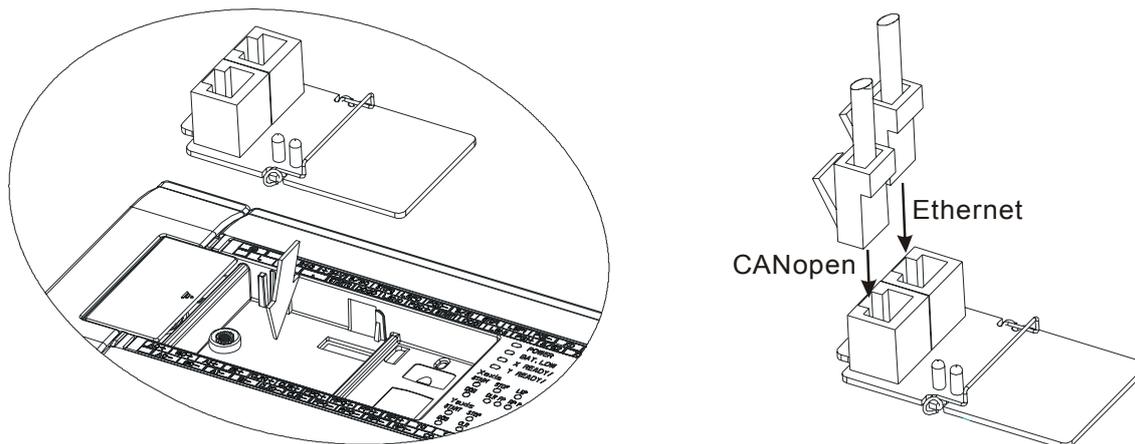
13.3 Product Profile & Installation

Product Profile:



Installing DVP-FPMC on DVP-PM MPU and connecting it to communication wire:

13 CANopen Communication Card



13.4 Parameters for Control Register (CR)

■ Normal Mode

CR#	Function	Attribute	Data type	Data length
001	Firmware version of DVP-FPMC	R	Word	1
052	CANopen synchronous packet sending setting	R/W	Word	1
053	CANopen node ID setting	R/W	Word	1
054	CANopen transmission speed setting	R/W	Word	1
055	CANopen SDO / NMT timeout waiting time	R/W	Word	1
056	DVP-FPMC error status	R/W	Word	1
059	Network IP and port setting for DVP-FPMC	R/W	Word	3
062	Ethernet connection command and status	R/W	Word	1
063	IP and port setting for ethernet master	R/W	Word	3
064	Length of data to be transmitted through ethernet	R/W	Word	1
065	Content of data to be transmitted through ethernet	R/W	Word	512
066	Length of data to be received through ethernet	R	Word	1
067	Content of data to be received through ethernet	R	Word	512
070	Node ID of SDO server	R/W	Word	1
071	SDO access command and status	R/W	Word	1
072	SDO OD index	R/W	Word	1
073	SDO OD sending/receiving datum 1	R/W	Word	512
074	SDO OD sending/receiving datum 2	R/W	Word	512
075	SDO OD sending/receiving datum 3	R/W	Word	512
076	SDO OD sending/receiving datum 4	R/W	Word	512
080	NMT command	R/W	Word	1

■ A2 Mode Parameters

13 CANopen Communication Card

CR#	Function	Attribute	Data type	Data length
010	CANopen bus scan	R/W	Word	1
020	CANopen bus communication status	R	Word	1
040	Error status of node	R	Word	1
050	CANopen bus control command	R/W	Word	1
090	Data written in DVP-FPMC QBuffer	R/W	Word	32
091	Address of data written in DVP-FPMC QBuffer	R/W	Word	32
092	Data read from DVP-FPMC QBuffer	R/W	Word	32
093	Address of data read from DVP-FPMC QBuffer	R/W	Word	32
n00	Node ID	R	Word	1
n01	Manufacturer ID	R	Word	1
n02				
n03	Product ID	R	Word	1
n04				
n05	Firmware version	R	Word	1
n06				
n07	Product type	R	Word	1
n08				
n09	CANopen node communication status	R	Word	1
n10	Emergency error code	R	Word	1
n11	Manufacturer's error code	R	Word	1
n12				
n20	Servo drive status	R	Word	1
n21	Current operating mode of servo drive	R	Word	1
n22	Servo drive position	R	Word	1
n23				
n40	Node control command	R/W	Word	1
n50	SDO access command and status	R/W	Word	1
n51	SDO OD(Object Dictionary) index	R/W	Word	1
n52	SDO sending or receiving datum 1	R/W	Word	512
n53	SDO sending or receiving datum 2	R/W	Word	512
n54	SDO sending or receiving datum 3	R/W	Word	512
n55	SDO sending or receiving datum 4	R/W	Word	512
n60	Servo drive control	R/W	Word	1
n61	Operating mode selection	R/W	Word	1
n70	Target position of profile position mode	R/W	Word	1
n71				
n72	Target speed of profile position mode	R/W	Word	1
n73				
n74	Acceleration time(ms) of profile position mode	R/W	Word	1
n75				

13 CANopen Communication Card

CR#	Function	Attribute	Data type	Data length
n76	Deceleration time(ms) of profile position mode	R/W	Word	1
n77				
n78	Profile position settings	R/W	Word	1
n80	Homing method	R/W	Word	1
n81	Home offset	R/W	Word	1
n82				
n83	Homing speed	R/W	Word	1
n84				
n85	Speed for Homing after hitting DOG	R/W	Word	1
n86				
n87	Homing acceleration time	R/W	Word	1
n88				
n89	Enable homing	R/W	Word	1
n90	Target position of interpolation position mode	R/W	Word	1
n91				
n92	Enable interpolation positioning	R/W	Word	1

■ CANopen Common Mode

CR#	Function	Attribute	Data type	Data length
500	CANopen mode switch	R/W	Word	1
504	Enable heartbeat protocol	R/W	Word	1
505	Execution status of heartbeat protocol	R	Word	1
506	Heartbeat status	R	Word	1

■ Object Dictionary Parameters

CR#	Function	Attribute	Data type	Data length
1006h	Synchronization cycle setting	R/W	DWord	1
1017h	DVP-FPMC heartbeat cycle setting	R/W	Word	1
1400h~143Fh	Parameter settings for RPDO	R/W	Word	3
1600h~163Fh	Parameter settings for RPDO data mapping	R/W	DWord	4
1800h~183Fh	Parameter settings for TPDO	R/W	Word	3
1A00h~1A3Fh	Parameter settings for TPDO data mapping	R/W	DWORD	4
2000h~207Fh	PDO data registers	R/W	Word	4

13.5 Explanations on Control Registers

■ Parameters for Normal Mode

13 CANopen Communication Card

CR#001: Firmware version of DVP-FPMC

[Explanations]

The firmware version of DVP-FPMC is displayed in hex values, e.g. H'8161 indicates the firmware version issue date as "Afternoon, August 16."

CR#052: CANopen synchronous packet sending setting

[Explanations]

- Low Byte of CR052 sets up the CANopen synchronous function. When the value of the low byte is 1, DVP-FPMC will send out synchronous packet. If the value is 0, the function will be disabled.
- High Byte of CR052 sets up the synchronous cycle time with DVP-PM. Care should be taken on setting the high-byte value. 5 times of the set value should equal to the value in D1040 of DVP-PM. Synchronization between DVP-PM and DVP-FPMC will be enabled when the high-byte value is bigger than 1.

Bit	b15~ b8	B7~b0
Function	Synchronous cycle time setting	Enable synchronous packet sending

CR#053: DVP-FPMC CANopen node ID setting

[Explanations]

Set up the CANopen node ID within the range 5~127. Default: 127.

CR#054: DVP-FPMC CANopen transmission speed setting

[Explanations]

Set up the CANopen transmission speed. The setting status is indicated by b15; b15=1 indicates that setting is under progress and b15=0 indicates that setting is completed. For example, specify K1000 in CR#054 directly and the CANopen transmission speed will be set up as 1000kb/s.

Bit	b15	b14~b0
Function	Setting status 0: completed 1: under progress	1000: CANopen speed = 1000kb/s 500: CANopen speed = 500kb/s

CR#055: CANopen SDO / NMT timeout waiting time

[Explanations]

Set up the waiting time for CANopen SDO / NMT timeout. Unit: ms. Default: 1000.

CR#056: DVP-FPMC error status

[Explanations]

Display the error status of DVP-FPMC. Please refer to the below table for the error status and the actions.

13 CANopen Communication Card

Error status	Content	Actions
CANopenconnection error	C1	Confirm the CANopen nodes of the current Slave.
Ethernetconnection error	E1	Confirm the connection between the communication module and Ethernet

CR#059: Network IP and port setting for DVP-FPMC

[Explanations]

Set up the network IP and the port number for DVP-FPMC. Data length: 3 words. Default IP: 192.168.0.100; port: 1024. Refer to the table below for the setting format.

Word 0		Word 1		Word 2
H-byte	L-byte	H-byte	L-byte	1024
192	168	0	100	

CR#062: Ethernet connection command and status

[Explanations]

Set up Ethernet connection commands and obtain the connection status.

- H'0: indicates "Disconnected".
- H'30: indicates "Connected"
- H'10: sends out connection command.
- H'20: sends out disconnection command.

CR#063: IP and port setting for ethernet master

[Explanations]

Set up the network IP and the port number for ethernet master. Data length: 3 words. Default IP: 192.168.0.100; port: 1024. Refer to the table below for the setting format.

Word 0		Word 1		Word 2
H-byte	L-byte	H-byte	L-byte	1024
192	168	0	100	

CR#064~CR#67: Data length and content to be accessed through ethernet

[Explanations]

Set up the data to be accessed through Ethernet. (Max 1024 bytes)

- In data sending cases, after the user enters the length of data into CR#064 and the data content into CR#65, DVP-FPMC will send out the data and clear the two registers automatically.
- In data receiving cases, the user reads the length of data in CR#066 first, and then the data content matched with the length of data in CR#067.

CR#070: Node ID of SDO server

13 CANopen Communication Card

[Explanations]

Set up the node ID of the SDO server within the range 1~127.

CR#071: SDO access command and status

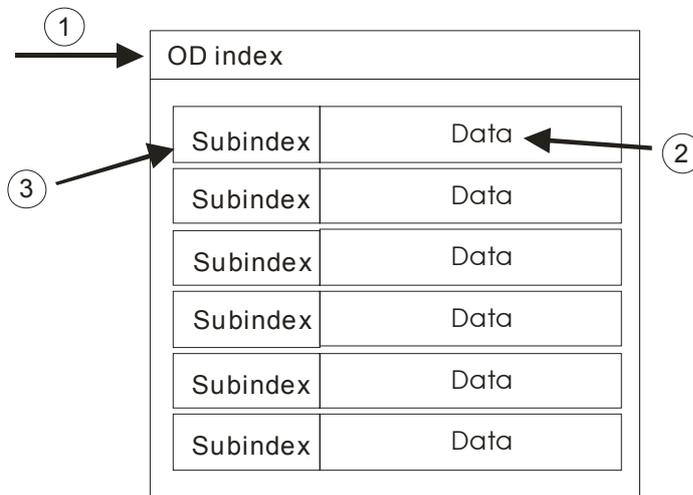
[Explanations]

Set up the SDO access commands and obtain the status. Please refer to the table below for the setting format.

Bit	b15~b8	b7~b4	b3	b2~b0
Function	Subindex of the the target OD index.	Data length (unit: byte) Range: 1~8 Length of data should be specified in write-in cases	Error flag	Commands: 0: completed 1: write in (check up) 2: read out (check up) 3: write in (No check up) 4: read out (No check up)

SDO data transmission diagram:

1. Specify the OD index of SDO server (CR#070) in CR#072.
2. Set up the data to be transmitted in CR#073~CR#076.
3. Refer to the above table. Specify the subindex in b15~b8 of CR#071 and the SDO access commands as well.



CR#072: SDO OD(Object Dictionary) index

[Explanations]

Specify the target OD index. Range: H'0000~H'FFFF.

CR#073~CR#076: SDO OD sending/receiving data 1~4

[Explanations]

The 4 registers store the data to be accessed (max 1024 bytes) through SDO protocol. When an error occurs

13 CANopen Communication Card

during SDO data transmission, the error code will be stored in CR#073 and CR#074. If CR#073~CR#076 are used at one time, CR#073 serves as LSB and CR#076 as MSB.

CR#080: NMT command

[Explanations]

When DVP-FPMC is Master, NMT commands can be used to change the network status. Please refer to the table below for the setting format.

Bit	b15~b8	b7~b0
Function	Network management command 1: enable node communication 2: disable node communication 128: switch the operation mode 129: reset node communication	Node ID of Slave

■ Parameters for A2 mode

A2 mode is one of the applications of DVP-FPMC specifically for delta servo drive: ASDA-A2. In A2 mode, there are 4 CANopen nodes ID1~ID4 planned for ASDA-A2, and control registers CR#100~CR#499 corresponding to servo parameters. CR#100~CR#199 are CRs for Node ID1, CR#200~CR#299 are CRs for Node ID2, CR#300~CR#399 are CRs for Node ID3, CR#400~CR#499 are CRs for Node ID4. Care should be taken on the special index "n" in the CR number. The value of index n could be 1, 2, 3 or 4 indicates 100, 200, 300 or 400 corresponding to the 4 CANopen nodes for ASDA-A2. Control registers for ASDA-A2 application are applicable only in A2 mode.

CR#010: CANopen bus scan

[Explanations]

Scan CANopen nodes ID1~ID4. b0~b3 of CR#010 correspond to node1~node4. When the bit is 1, the corresponding node will be scanned and the content of the register will be cleared automatically. Please refer to the table below.

Bit	b15~b4	b3	b2	b1	b0
Node ID	Reserved	Node 4	Node 3	Node 2	Node 1

CR#020: CANopen bus communication status

[Explanations]

Display the node communication status by 2 consecutive bits: 00: disconnected, 01: connected, 11: ready. Please refer to the table below.

Bit	b15~b8	b7~b6	b5~b4	b3~b2	b1~b0
Node ID	Reserved	Node 4	Node 3	Node 2	Node 1

13 CANopen Communication Card

CR#040: Error status of node

[Explanations]

Display the error status of servo drive. b0~b3 of CR#010 correspond to node1~node4. When error occurs, the corresponding bit will be 1. When error reset command is executed, the content in this register will be cleared automatically. Please refer to the table below.

Bit	b15~b4	b3	b2	b1	b0
Node ID	Reserved	Node 4	Node 3	Node 2	Node 1

CR#050: CANopen bus control command

[Explanations]

Send out bus control commands to the connected nodes on CANopen. Set value = 1 indicates "All Servo On". Set value = 128 indicates "Servo Off." Set value = 129 indicates "Error Reset." When the execution is completed, the content in the register will be cleared automatically. Please refer to the table below for the setting format.

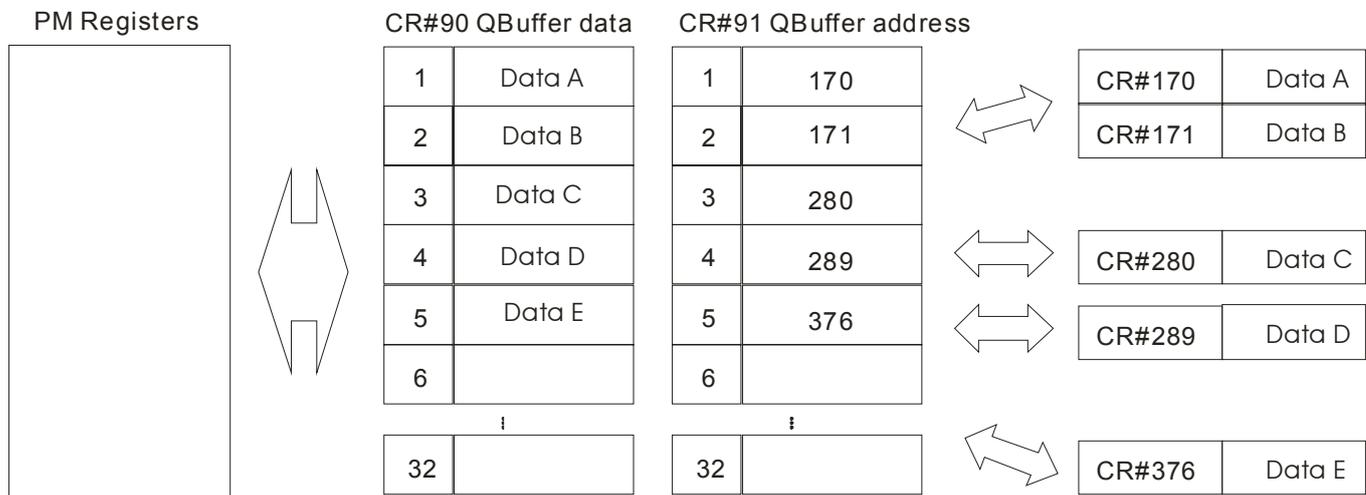
Bit	b15~b8]	b7~b0]
Command	Reserved	ALL Servo On: 1 ALL Servo Off: 128 Error Reset: 129

CR#090~CR#93: DVP-FPMC QBuffer data / data address

[Explanations]

QBuffer is designed for accessing several discontinuous control registers within CR100~CR499 by single TO/FROM accessing. PLC scan time can be reduced by decreasing the number of TO/FROM instructions through QBuffer application. The QBuffer access operation is explained as the diagram below.

For example, write the discontinuous CR number (CR#170, 171, 280, 289, 376) into the QBuffer address CR#91, and DVP-FPMC will map the content in these registers into QBuffer data CR#90 automatically. The user only needs to access CR#90 and the content in QBuffer address (CR#170, 171, 280, 289, 376) can be modified at one time.



13 CANopen Communication Card

CRn00: Node ID

[Explanations]

Display the Node ID of servo drive on CANopen.

Node ID = 1: CR#100 = 1

Node ID = 2: CR#200 = 2

Node ID = 3: CR#300 = 3

Node ID = 4: CR#400 = 4

CRn01~CRn02: Manufacturer ID

[Explanations]

Display the manufacturer ID of the ASDA-A2 in DWORD format.

CRn03~CRn04: Product ID

[Explanations]

Display the product ID of the ASDA-A2 in DWORD format.

CRn05~CRn06: Firmware version

[Explanations]

Display the firmware version of the ASDA-A2 in DWORD format.

CRn07~CRn08: Product type

[Explanations]

Display the product type of the ASDA-A2 in DWORD format.

CRn09: CANopen node communication status

[Explanations]

Display the node communication status on CANopen. Please refer to the table below.

Status	Value
Disconnected	H'1
Connected	H'2
Operation mode	H'5
Error	H'6
Resetting	H'7

CRn10: Emergency error code

[Explanations]

Display the error code defined by CANopen protocol when error occurs on certain node.

13 CANopen Communication Card

CRn11~CRn12: Manufacturer's error code

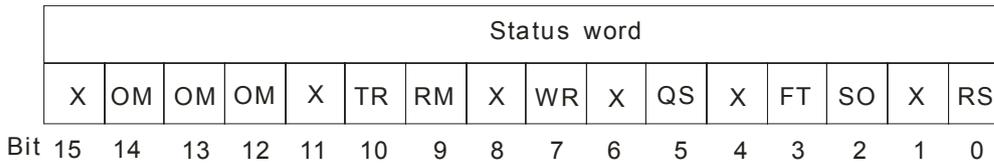
[Explanations]

Display the error code defined by the manufacturer when error occurs on certain servo drive. For error code details please refer to the manual of Delta ASDA-A2 servo drive.

CRn20: Servo drive status

[Explanations]

Display the current status of ASDA-A2. For the status codes, please refer to the table below.



- RS: Servo ready. When servo initialization is completed, the bit will be 1, indicating the ready status.
- SO: Servo On. The bit will be 1 when the servo drive is ON.
- FT: Error flag. When error occurs in the servo drive, the flag will be 1.
- QS: Quick stop. The user can execute quick stop on the servo when the bit is 1.
- WR: Warning. When the servo sends warning messages, the bit will be 1.
- RM: Remote monitor. Remote monitor is applicable when the bit is 1.
- TR: Motion command completed. When motion command is completed, the bit will be 1.
- OM[14~12]: Indicates status of each operating mode. Please see the table below.

	Profile position mode	Homing mode	Interpolation position mode
OM[12]	Position setting completed	Executing Homing	Executing interpolation
OM[13]	Following error	Homing error	X
OM[14]	X	X	SYNC executes

Note: "X" indicates "Reserved"

CRn21: Current operating mode of servo drive

[Explanations]

Display the current operating mode of the servo.

Content	Operating mode
0x01	Profile position mode
0x06	Homing mode
0x07	Interpolation position mode

CRn22~CRn23: Servo drive position

[Explanations]

Display the current position of servo drive in DWORD format.

13 CANopen Communication Card

CRn40: Node control command

[Explanations]

Send out node control commands to the connected nodes. Set value = 1 indicates "Servo On". Set value = 128 indicates "Servo Off." Set value = 129 indicates "Error Reset." For the setting format, please refer to the table below.

Bit	b15~b8]	b7~b0]
Content	reserved	Servo On: 1 Servo Off: 128 Error Reset: 129

CRn50: SDO access command and status

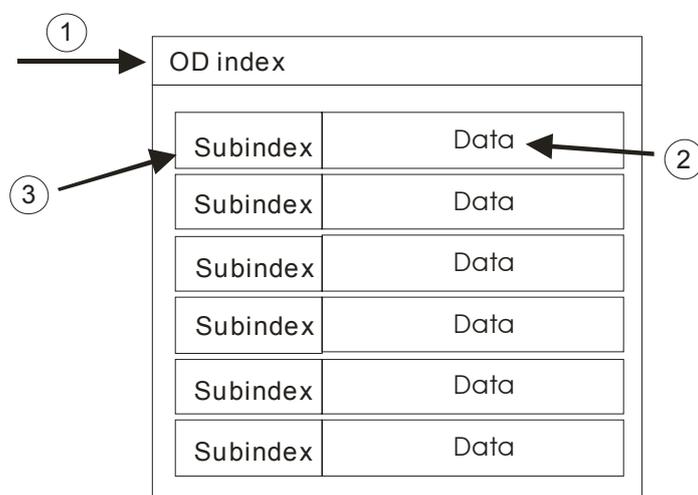
[Explanations]

Set up the SDO access commands on the node and obtain the status. Please refer to the table below for the setting format.

Bit	b15~b8	b7~b4	b3	b2~b0
Function	Subindex of the the target OD index.	Data length (unit: byte) Range: 1~8 Length of data should be specified in write-in cases	Error flag	Commands: 0: completed 1: write in (check up) 2: read out (check up) 3: write in (No check up) 4: read out (No check up)

SDO data transmission diagram:

1. Specify the OD index of SDO server in CRn51.
2. Set up the data to be transmitted in CRn52~CRn55.
3. Refer to the above table. Specify the subindex in b15~b8 of CRn50 and the SDO access commands as well.



13 CANopen Communication Card

CRn51: SDO OD(Object Dictionary) index

[Explanations]

Specify the OD index of the node. Range: H'0000~H'FFFF.

CRn52~CRn55: SDO OD sending/receiving data 1~4

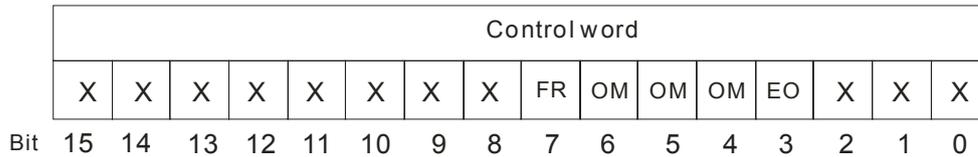
[Explanations]

The 4 registers store the data to be accessed (max 1024 bytes) through SDO protocol. When an error occurs during SDO data transmission, the error code will be stored in CRn52 and CRn53. If CRn52~CRn55 are used at one time, CRn52 serves as LSB and CRn55 as MSB.

CRn60: Servo drive control

[Explanations]

Sent out control commands to ASDA-A2. For the control codes, please refer to the table below.



- EO: Servo On. The servo drive will be ON when the bit is 1.
- FR: Error reset. The servo drive will reset the error when the bit is 1.
- OM: control commands for each operating mode. Please refer to the table below.

	Profile position mode	Homing mode	Interpolation position mode
OM[4]	Set up new target position	Homing On	X
OM[5]	Enable target position changing during positioning process	X	X
OM[6]	Absolute / Relative positioning	X	X

Note: "X" indicates "Reserved"

CRn61: Operating mode selection

[Explanations]

Select the operating mode of the servo. For the set values please refer to the table below.

Set value	Operating mode
0x01	Profile position mode
0x06	Homing mode
0x07	Interpolation position mode

CRn70~CRn71: Target position of profile position mode

[Explanations]

Set up the target position in DWORD format.

13 CANopen Communication Card

CRn72~CRn73: Target speed of profile position mode

[Explanations]

Set up the target speed in DWORD format.

CRn74~CRn75: Acceleration time(ms) of profile position mode

[Explanations]

Set up the acceleration time in DWORD format.

CRn76~CRn77: Deceleration time(ms) of profile position mode

[Explanations]

Set up the deceleration time in DWORD format.

CRn78: Profile position settings

[Explanations]

Absolute / relative positioning setting of profile position mode.

- 0: Positioning completed
- 1: Absolute positioning. The content of this CR will be cleared automatically when positioning is completed.
- 2: Relative positioning. The content of this CR will be cleared automatically when positioning is completed.
- 3: Absolute positioning. The content of this CR will be reserved when positioning is completed.

CRn80: Homing method

[Explanations]

Set up Homing method. Range: 1~35. For the details of each Homing method, please refer to Chapter 13 Homing Modes in CiA specification: CiA 402 V2.0.0 CANopen device profile for drives and motion control.

CRn81~CRn82: Home offset

[Explanations]

Set up the Home offset in DWORD format. Range: -2,147,483,648~2,147,483,647.

CRn83~CRn84: Homing speed

[Explanations]

Set up the Homing speed in DWORD format. Range: 0~2,147,483,647.

13 *CANopen Communication Card*

CRn85~CRn86: Speed for Homing after hitting DOG

[Explanations]

Set up the speed for Homing after hitting DOG in DWORD format. Range: 0~2,147,483,647

CRn87~CRn88: Homing acceleration time

[Explanations]

Set up the the Homing acceleration time in DWORD format. Range: 0~2,147,483,647

CRn89: Enable Homing

[Explanations]

Enable Homing. Homing executes when the set value is 1. When Homing is completed, the content of the register will be cleared automatically.

CRn90~CRn91: Target position of interpolation position mode

[Explanations]

Set up the target position of interpolation position mode in DWORD format. Range: -2,147,483,648 ~ 2,147,483,647.

CRn92: Enable interpolation positioning

[Explanations]

Enable interpolation positioning. Interpolation positioning executes when the set value is 1. When the set value is 0, interpolation positioning will be disabled.

■ CANopen Common Mode

CR#500: CANopen mode switch

[Explanations]

CANopen mode switch for DVP-FPMC. Set value = 1 indicates A2 mode; 2 indicates Normal mode. Default: 1. Control registers for A2 mode will be unavailable when the mode is set as Normal mode, and control registers for Normal mode will be unavailable either when A2 mode is selected.

CR#504: Enable heartbeat protocol

[Explanations]

When heartbeat protocol is enabled in common mode, the heartbeat mechanism will be enabled by Master on node ID1~16(Slave). Set value = 0 indicates disable heartbeat and 1 indicates enable heartbeat protocol.

13 CANopen Communication Card

CR#505: Execution status of heartbeat protocol

[Explanations]

Display the execution status of heartbeat protocol on node ID1~16. Content = 0 indicates that execution is completed and 1 indicates that heartbeat protocol is executing.

CR#506: Heartbeat status

[Explanations]

Display the heartbeat status of node ID1~16. The node ID1~16 is corresponding to b0~b15. Heartbeat status is ON when the corresponding bit is 1; heartbeat status is 0 when the corresponding bit is 0. In addition, the corresponding bit will also be 0 when the Master is disconnected with the Slave.

		Heartbeat status of Slave															
Node ID		16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

■ Parameters for Object Dictionary

CR#H'1006: Synchronization cycle setting

[Explanations]

Set up the time interval (unit: μ s) for CANopen synchronization packet sending in normal mode. Data format: DWORD. Default: 5000. The time interval for synchronization cycle can be millisecond accuracy, i.e. values smaller than millisecond will be ignored. We recommend the user to set the minimum sync cycle time as 1ms for 4 PDOs, i.e. set the minimum sync cycle time as 1ms for PDOs between 1~4 and 2ms for PDOs between 4~8.

CR#H'1017: Heartbeat cycle setting

[Explanations]

Set up the heartbeat cycle (unit: ms, default: 0). When the heartbeat cycle is set to 0, the heartbeat function will be disabled.

CR#H'1400~CR#H'143F: Parameter settings for RPDO

[Explanations]

Set up the parameters for RPDO in normal mode. The register provides 3 words for the RPDO parameters. For the setting format please refer to the table below.

Transmission method	PDO ID	
Word2	Word1(High)	Word0(Low)

- PDO ID: CANopen POD ID specified in 2 words. Default:

13 CANopen Communication Card

CR#H'1400 = H180 + DVP-FPMC node ID(CR#053)

CR#H'1401 = H280 + DVP-FPMC node ID(CR#053)

CR#H'1403 = H380 + DVP-FPMC node ID(CR#053)

CR#H'1404 = H480 + DVP-FPMC node ID(CR#053)

- Transmission method:

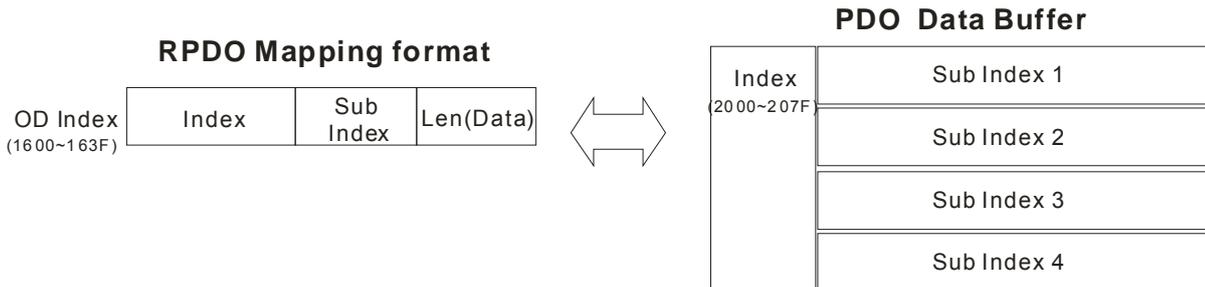
Set vlalue 1~240: PDO synchronize with CANopen packet sending and executes after every interval (CR#H'1006).

Set value 241~255: no action. Default: 241

CR#H'1600~CR#H'163F: Parameter settings for RPDO data mapping

[Explanations]

Set up parameters for RPDO data mapping in normal mode. The data format of RPDO parameter is DWORD. The 1st Word sets up OD Index, the high byte of the 2nd Word sets up subindex and low byte sets up the data length(unit: bit). For the setting format and the corresponding PDO data buffer, please refer to the table below.



CR#H'1800~CR#H'183F: Parameter settings for TPDO

[Explanations]

Set up the parameters for TPDO in normal mode. The register provides 3 words for the TPDO parameters. For the setting format please refer to the table below.

Transmission method	PDO ID	
Word2	Word1(High)	Word0(Low)

- PDO ID: CANopen POD ID specified in 2 words. Default:

CR#H'1800 = H200 + DVP-FPMC node ID(CR#053)

CR# H'1801 = H300 + DVP-FPMC node ID(CR#053)

CR#H'1803 = H400 + DVP-FPMC node ID(CR#053)

CR#H'1804 = H500 + DVP-FPMC node ID(CR#053)

- Transmission method:

Set vlalue 1~240: PDO synchronize with CANopen packet sending and executes after every interval (CR#H'1006).

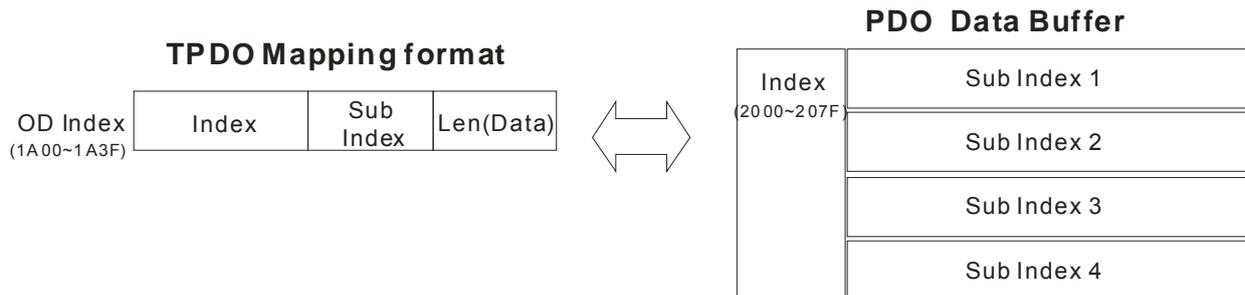
13 CANopen Communication Card

Set value 241~255: no action. Default: 241

CR#H'1A00~CR#H'1A3F: Parameter settings for TPDO data mapping

[Explanations]

Set up parameters for TPDO data mapping in normal mode. The data format of TPDO parameter is DWORD. The 1st Word sets up OD Index, the high byte of the 2nd Word sets up subindex and low byte sets up the data length(unit: bit). For the setting format and the corresponding PDO data buffer, please refer to the table below.



CR#H'2000~CR#H'207F: PDO data registers

[Explanations]

Data registers storing the data for PDO access. DVP-FPMC currently provides 2000H~207FH of OD as the data registers. Every index contains 4 subindexes for storing data and each subindex is a word. If the data to be accessed is bigger than a word, use the subindex area for the data transmission.

■ Object Dictionary for DVP-FPMC

Index (hex)	Object Name	Number of sub-indices	Name	Data type	Attribute	Allowed mapping target
1000	VAR	1	Product type	UNSIGNED32	R	N
1006	VAR	1	Synchronization cycle setting	UNSIGNED32	RW	N
1018	ARRAY	5	Product info.	UNSIGNED32	R	N
1200	ARRAY	3	SDO parameter of master	UNSIGNED32	R	N
1280	ARRAY	4	SDO parameter of slave	UNSIGNED32	R	N
:	:	:	:	:	:	:
128F	ARRAY	4	SDO parameter of slave	UNSIGNED32	R	N
1400	ARRAY	6	RPDO parameter	UNSIGNED32	RW	N
:	:	:	:	:	:	:
143F	ARRAY	6	RPDO parameter	UNSIGNED32	RW	N
1600	ARRAY	9	RPDO mapping parameter	UNSIGNED32	RW	N

13 CANopen Communication Card

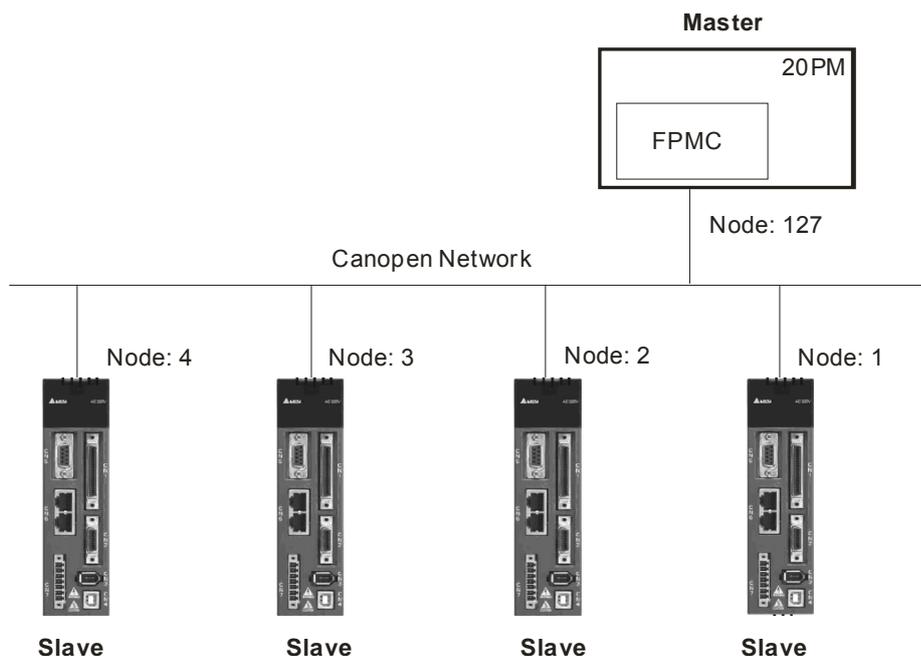
:	:	:	:	:	:	:
163F	ARRAY	9	RPDO mapping parameter	UNSIGNED32	RW	N
1800	ARRAY	6	TPDO parameter	UNSIGNED32	RW	N
:	:	:	:	:	:	:
183F	ARRAY	6	TPDO parameter	UNSIGNED32	RW	N
1A00	ARRAY	9	TPDO mapping parameter	UNSIGNED32	RW	N
:	:	:	:	:	:	:
1A3F	ARRAY	9	TPDO mapping parameter	UNSIGNED32	RW	N
2000	ARRAY	5	PDO data register	UNSIGNED32	RW	Y
:	:	:	:	:	:	:
207F	ARRAY	5	PDO data register	UNSIGNED32	RW	Y
6000	ARRAY	5	Mode switch	UNSIGNED8	R	Y
6100	ARRAY	17	Servo drive control	UNSIGNED16	R	Y
6120	ARRAY	17	Parameter of profile position mode	UNSIGNED32	R	Y
6200	ARRAY	5	Current operating mode of servo drive	UNSIGNED8	RW	Y
6300	ARRAY	5	Servo drive status	UNSIGNED16	RW	Y
6320	ARRAY	5	Servo drive position	UNSIGNED32	RW	Y

13.6 DVP-FPMC Mode Settings

■ A2 mode

In A2 mode, DVP-FPMC communicates with 4 ASDA-A2 through CANopen network. In the communication, DVP-FPMC serves as Master and servo drives serve as Slaves. The communication structure is illustrated as the diagram below. Default node ID of DVP-FPMC is 127. Set up the node ID of servo drives as ID1~4, which are the planned node IDs for A2 Slaves, and the data exchange between Master and Slave can be executed.

13 CANopen Communication Card



In A2 mode, there are 6 sets of PDOs for setting servo parameters. The user can monitor servo status directly by accessing the control registers through CANopen network. Additional setting for PDO parameters is not required. The PDO allocation for DVP-FPMC (Master) and servo drives (Slaves) is listed as the table below: 4 sets of PDO for DVP-FPMC and 2 sets of PDO for servo data transmission.

PDO	Master (transmit)	Slave (receive)
1	Target position of profile position mode (CRn70~CRn71) Target speed of profile position mode (CRn72~CRn73)	
2	Acceleration time(ms) of profile position mode (CRn74~CRn75) Deceleration time(ms) of profile position mode (CRn76~CRn77)	
3	Servo drive control (CRn60)	
4	Target position of interpolation position mode (CRn90~CRn91) Enable interpolation positioning (CRn92)	
5		Servo drive status (CRn20) Current operating mode of servo drive (CRn21)
6		Servo drive position (CRn22~ CRn23)

A2 mode communication setting:

- ASDA-A2 settings

Before setting up CANopen connection, set the servo drives in CANopen mode by the following steps:

1. Set A2 Keypad P1-01as 0x0B (Set ASDA-A2 in CANopen mode)

13 CANopen Communication Card

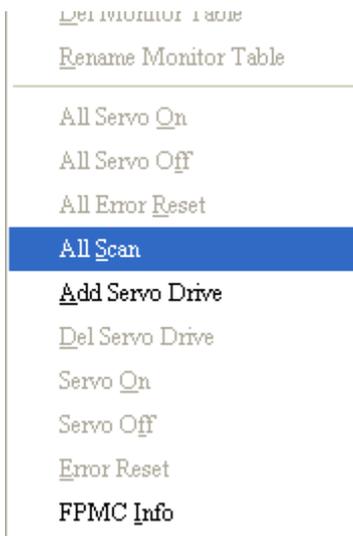
2. Set A2 Keypad P3-00 as 0x01~0x04 (Node ID of servo drives)
 3. Set up A2 Keypad P3-01 as 0x0403(1M bps) for baud rate setting. (2: 500 kbps; 4: 1Mbps) DVP-FPMC currently supports communication speed up to 1M (default) and 500k.
- DVP-FPMC settings

When all parameter settings for CANopen in connected drives are completed, create a CANopen network by DVP-FPMC as below:

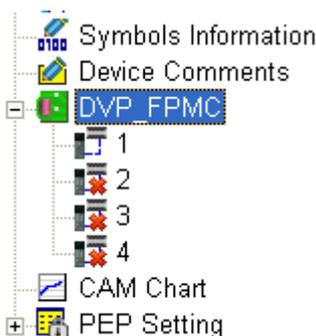
1. Set the content of CR#500 as 1 to set up DVP-FPMC in A2 mode.
2. Set up CR#053 for DVP-FPMC node ID setting. Default: 127
3. Set the content of CR#010 as H'FFFF to scan all connected drives.
4. Read the content of CR#010 by FROM instruction to check if the content is cleared.
5. Set the content of CR#050 as 1 to set On all connected drives.

Step 2 ~ step 4 can be performed on PMSOft. Right click at the icon of DVP-FPMC in the column of System Information, select All Scan and the connected drives will be displayed under DVP-FPMC. Please see the steps below for further setting.

1. Right click at DVP-FPMC icon and select All Scan.



2. Display the connection status of nodes after the scan is completed.



3. Right click at DVP-FPMC and select All Servo On. When the connected servos are enabled, a confirmation message will pop up. In addition, servo running status will be added after each node.

13 CANopen Communication Card



■ Normal mode

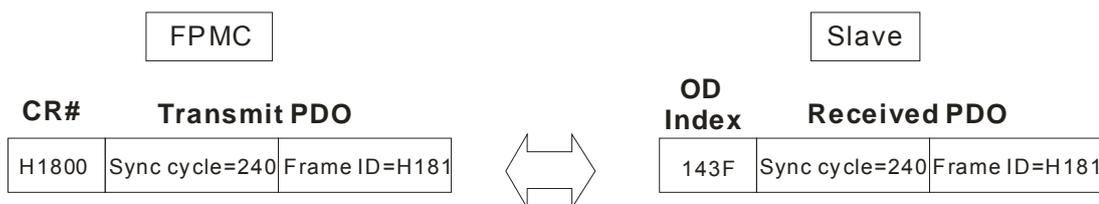
In normal mode, the user has to set up PDO parameters between DVP-FPMC (Master) and the Slave. In addition, use FROM/TO instructions to set up the control registers of DVP-FPMC and apply SDO protocol to write in the PDO parameters of servo. Please follow the PDO setting process below:

● PDO transmission parameter settings

PDO parameters contain Frame ID and synchronization cycle. Set range of frame ID is between 181h~578h. Please note that Frame ID and synchronization cycle should be the same between the corresponding PDOs of Master and Slave. The 2 types of PDO, the transmitting PDO (TPDO) and the receiving PDO (RPDO) will be explained as below:

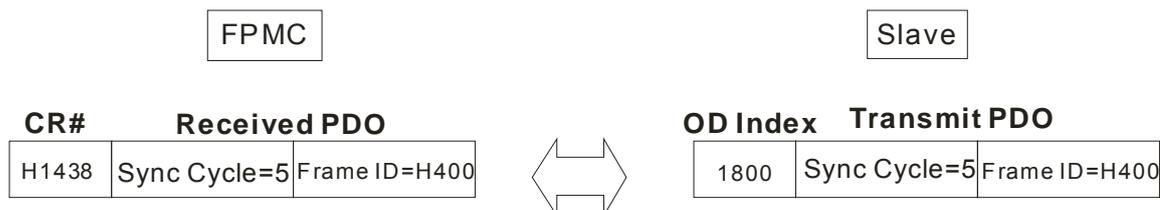
1. TPDO settings

1800H~183FH of DVP-FPMC serves as TPDO, corresponding to RPDO of the Slave. For example, set up the OD index H1800 (TPDO) of Master to communicate with the OD index 143F(RPDO) of Slave with the frame ID H181 and sync cycle time 240 as the diagram below.



2. RPDO settings

1400H~143FH of DVP-FPMC serves as RPDO, corresponding to TPDO of the Slave. For example, set up the OD index H1438 (RPDO) of Master to communicate with the OD index 1800 (TPDO) of Slave with the frame ID H400 and sync cycle time 5 as the diagram below.

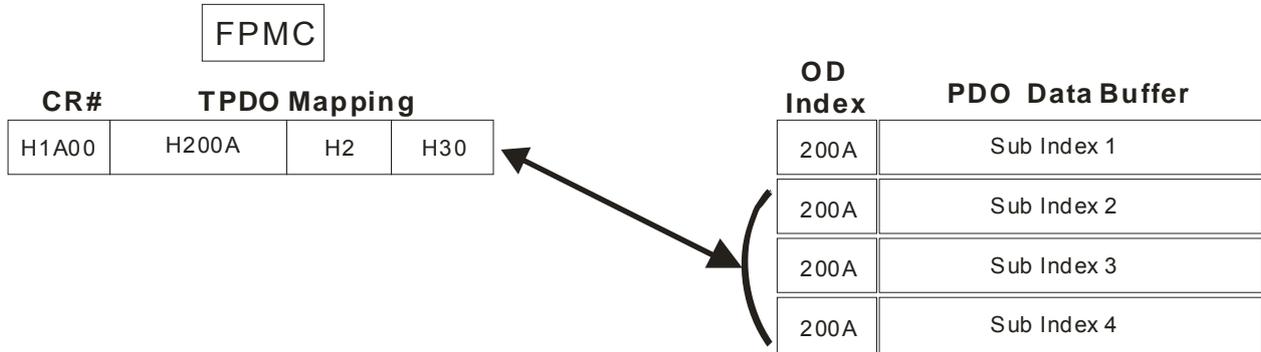


● PDO mapping parameter settings

Set up the mapping target and the data length of PDO data buffer. The max data length can be 64 bits,

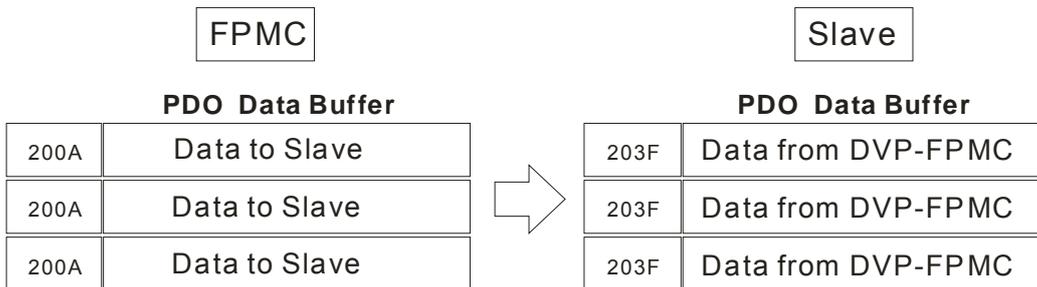
13 CANopen Communication Card

i.e. 4 subindexes in PDO data buffer can be occupied at one time. For example, set up the TPDO mapping parameter of DVP-FPMC as H200A, H2, H30, indicating the 2nd subindex of H200A with data length 48 bits. In this case, the subindex 2 ~4 in PDO data buffer will be occupied.



- Set up PDO data

Entre the data to be transmitted by PDO in the above data buffer of OD index. For example, set up the TPDO data buffer of DVP-FPMC as subindex 2~4 of OD index 200A, and the RPDO data buffer of Slave as subindex 1~3 of OD index 203F. When the communication executes, data transmission will be executed according to the specified synchronization cycle setting.



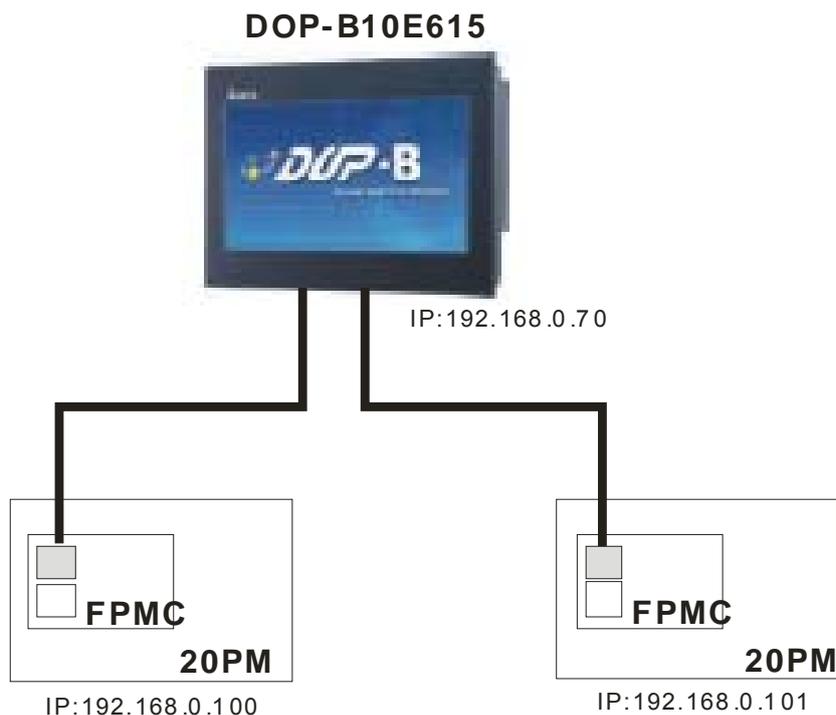
13.7 DVP-FPMC Ethernet Applications

DVP-FPMC supports Ethernet communication between DVP-FPMC and Ethernet devices, or between DVP-FPMC and PC through PMSOft for program upload/download and monitoring. For external wirings, the user only needs to connect the Ethernet COM ports on DVP-FPMC and the device to be connected through an Ethernet cable. Please also refer to detailed hardware configurations on 13.3 Product Profile & Installation. When connecting with PC, the Ethernet indicator will be ON in normal connection status. If the indicator is OFF, please check the hardware configuration or PC settings.

13.7.1 Communication between DVP-FPMC and HMI

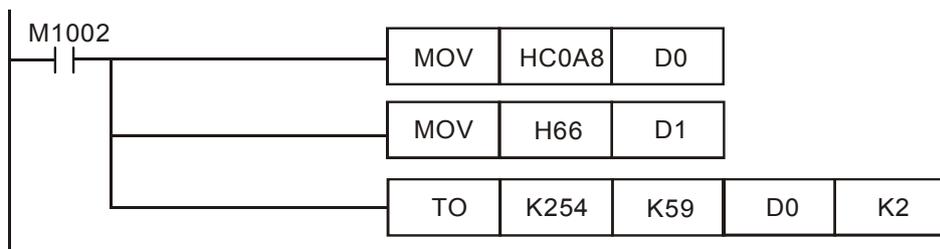
- Configuration

The example applies 2 sets of DVP-20PM with DVP-FPMC for data exchange between PLCs and HMI (DOP-B10E615) through Ethernet. The hardware configuration is as the diagram below. The control purpose is to control Y0~Y7 on both PLCs by HMI.

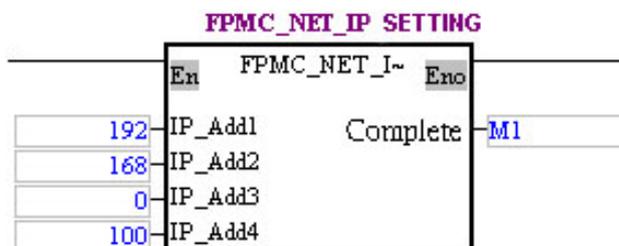


■ DVP- FPMC Setting

In this example, DVP-FPMC functions as Slave, therefore IP address setting is required. The user does not need to set up the IP address of the device to be connected and the Ethernet settings. To set up the IP of DVP-FPMC, the user needs to fill the IP into CR#59 according to the setting format illustrated in explanations of CR#59 in 13.5 Explanations on Control Registers. Take IP 192.168.0.100 for example, the program in DVP-PM will be as follows:



In addition, the user can also take advantage of the function block as below to set up the IP address. The input interfaces IP_Addr1~IP_Addr4 refer to the IP setting format. When the setting is completed, the output interface will be ON.

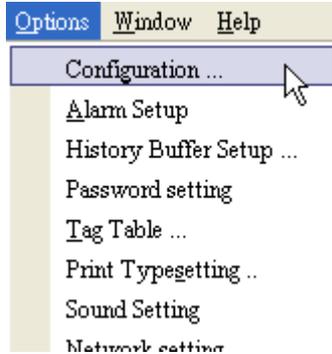


13 CANopen Communication Card

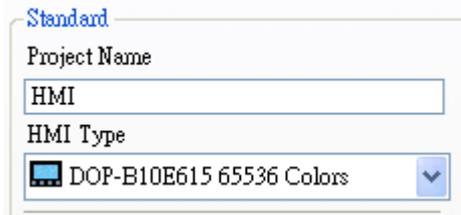
■ HMI Setting

HMI functions as Master, therefore the Ethernet connection with the 2 Slaves should be set up in addition to its IP address. To set up the connection between HMI and DVP-FPMC, please follow the procedure below:

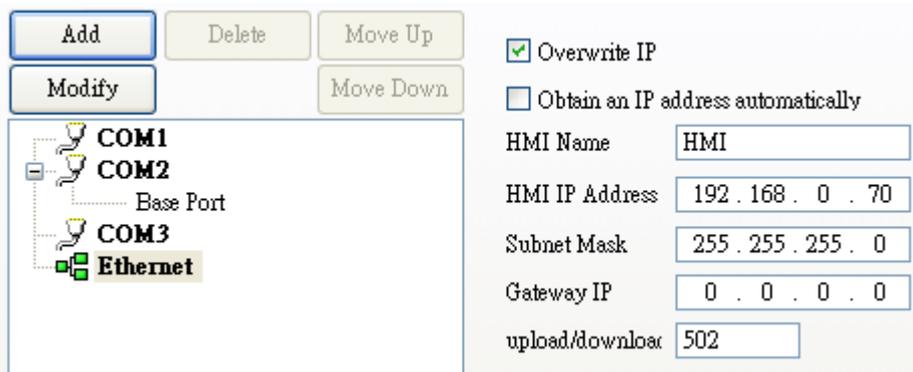
1. Open or create a file in Screen Editor. Select Configuration from **Options** in menu bar.



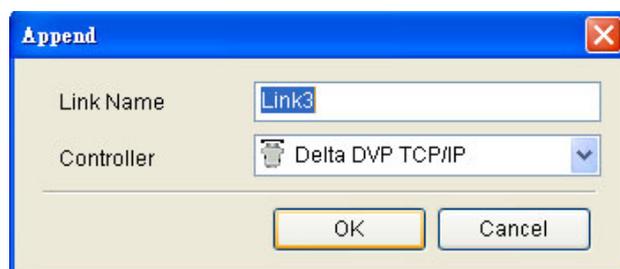
2. In "General" window, select DOP-B10E615 as the HMI type from the drop-down list in "Standard" column.



3. In "COM port" window, select "Ethernet" and check "Overwrite IP" in the right. Enter the IP address of HMI.

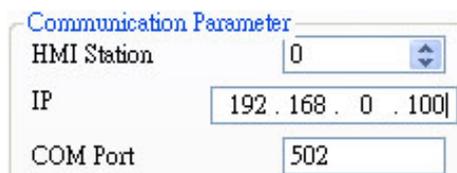


4. Add the connections for the devices to be connected. Click "Add" and enter the Link Name for the device. Select "Delta VP TCP/IP" in the list of Controller and click OK, the device will be created with the Link Name.



13 CANopen Communication Card

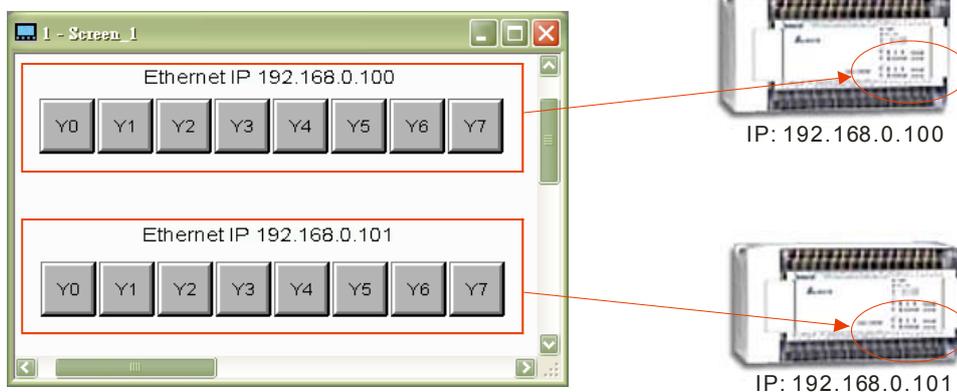
- Set up the IP of the created device. Click the created device and enter its IP in “Communication Parameter” column to configure the IP address of one DVP-FPMC. The setting procedure of the other DVP-FPMC is the same. Create Link4 and specify its IP as well.



- After the devices are created, create an object in the editing area. Click the object and click the icon  in the Property column to open the input window of Write Address / Read Address. Select the created device (Link3) in “Link”. In this way, the object can be operated to control the devices of Slave through Ethernet.



Design the buttons Y0~Y7 for both DVP-PM according to the above procedure as the diagram below. Buttons Y0~Y7 correspond to Y0~Y7 of Link3 (Slave1) and Link4 (Slave2). When the above settings are completed, the Ethernet connection between HMI and the 2 DVP-PM can be performed.



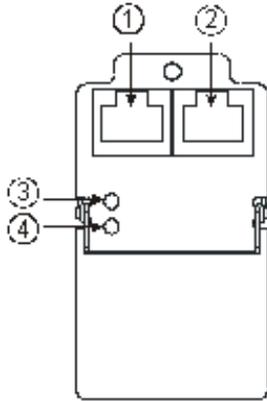
13.7.2 Communication between DVP-FPMC and PMSOft

PMSOft is featured with Ethernet function through DVP-FPMC. The ethernet communication interface between DVP-PM and PC supports software functions including program download/upload and monitoring on PMSOft.

■ Hardware Settings

For external wirings, the user only needs to connect the Ethernet COM ports on DVP-FPMC and PC through an Ethernet cable. When connecting with PC, the Ethernet indicator on DVP-FPMC will be ON in normal connection status. If the indicator is OFF, please check the hardware configuration or PC settings.

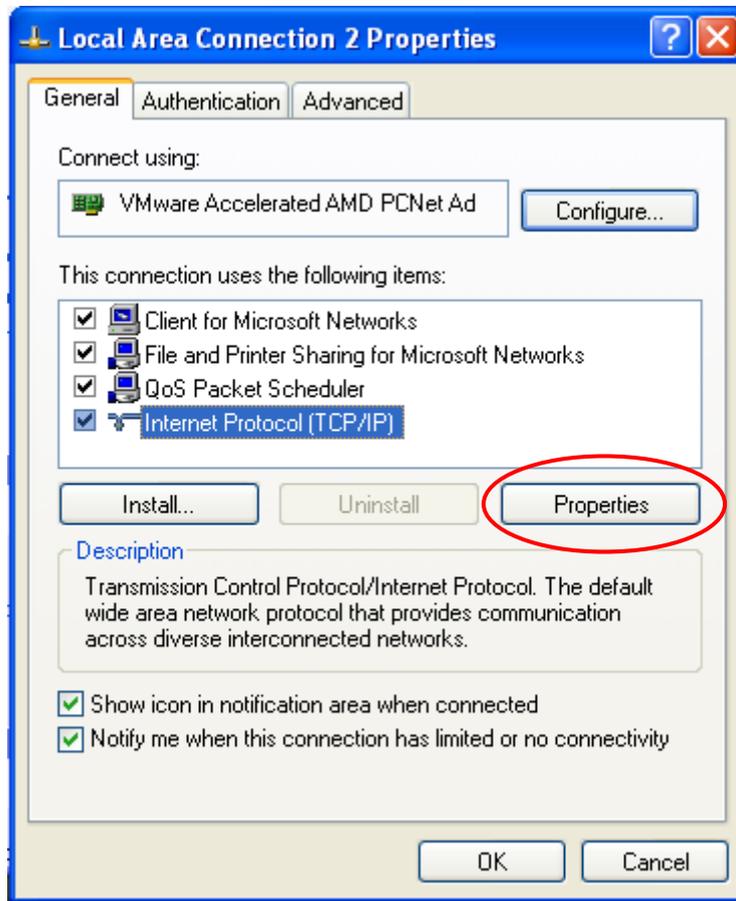
13 CANopen Communication Card



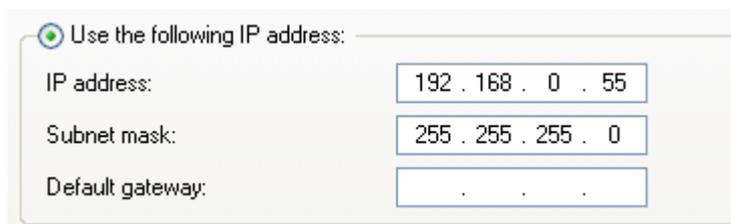
①	CANopen connector
②	Ethernet connector
③	CANopen indicator
④	Ethernet indicator

■ PC Setting

1. In “Local Area Connection Properties” window, select “Internet Protocol [TCP/IP]” and click Properties to open the IP setting window.



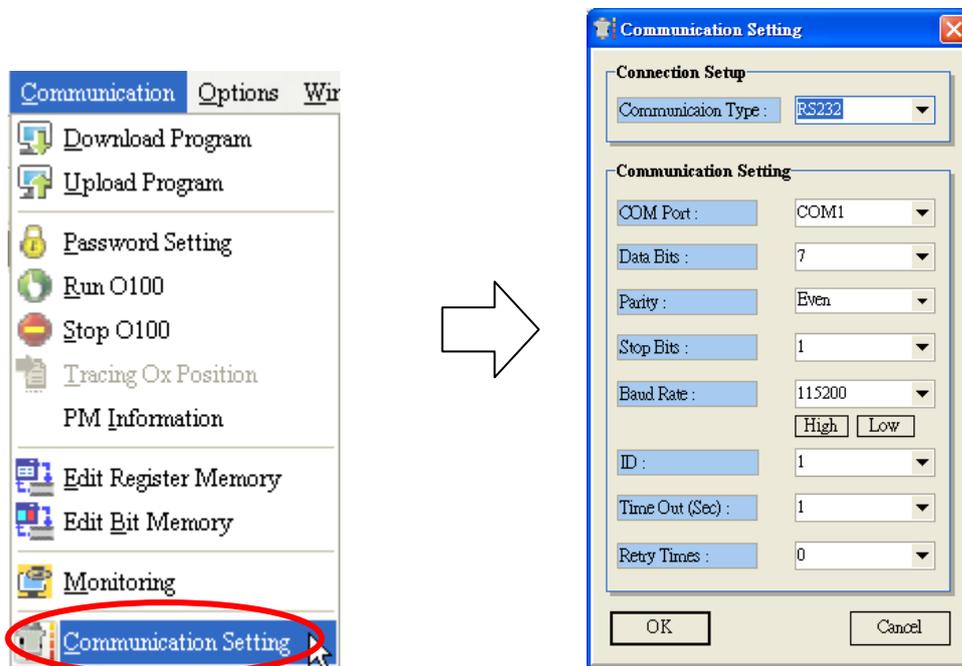
2. In “Properties” window, check “Use the following IP address” and enter 192.168.0.55 in IP address, where the last value of IP address can be replaced by values 1~255 other than 100. For Subnet mask, enter 255.255.255.0 and click OK.



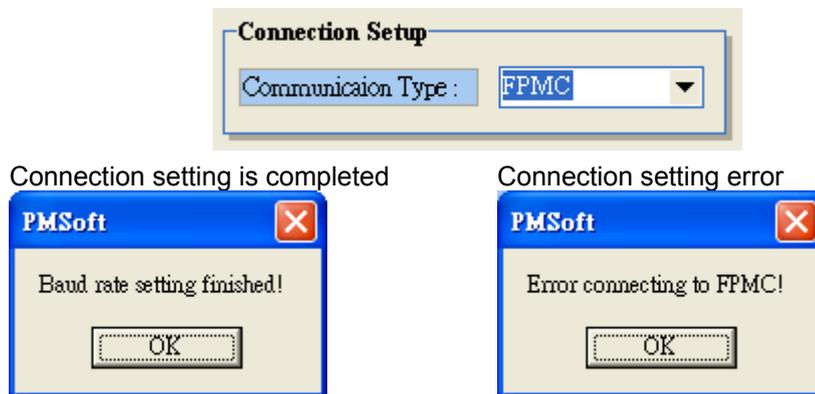
13 CANopen Communication Card

■ PMSOft Setting

1. Select “Communication Setting” from “Communication” in menu bar to enter the communication setting window.



2. Select FPMC in “Communication Type” and click OK.



When connection setting is completed, program download/upload or monitoring can be performed through Ethernet.

3. Program Download: To download the program, click the icon  (PC->PM) in the toolbar or select PC->PM from “Communications” in the menu bar. The download process through Ethernet interface is the same as the process through other COM ports.
4. Program Upload: To upload the program, click the icon  (PM->PC) in the toolbar or select PM->PC from “Communications” in the menu bar. The upload process through Ethernet interface is the same as the process through other COM ports.
5. Monitoring: To monitor the program operation, click the icon  (Monitoring) in the toolbar or select

13 CANopen Communication Card

Monitoring from “Communications” in the menu bar. The monitoring process through Ethernet interface is the same as the process through other COM ports.

13.8 LED Indicator & Troubleshooting

■ CANopen LED

LED status	Indication	How to correct
Green light off	Not connected to CAN bus cable	Check if the connection is correctly connected.
Green light on	Connection with CAN bus cable is normal.	--

■ Ethernet LED

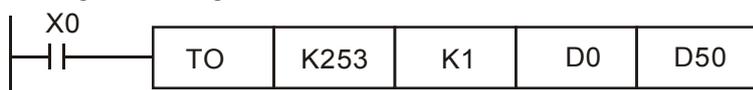
LED status	Indication	How to correct
Green light off	Not connected to the network	Check if the network cable is correctly connected.
Green light on	The network connection is normal.	--
Green light flashes	Network in operation	

14 High Speed Compare and Capture

14.1 High Speed Compare and Capture Function

DVP-PM supports 8 sets of registers for high-speed Compare and Capture function. Through FROM/TO instructions we can access CR253 for setting the control status or reading the data for high speed Compare or Capture function. The below table explains the settings of FROM/TO instructions and the relative registers.

■ Setting control registers



- D0 The number of the starting set: n (n=0~7, 8 sets)
- D1 0
- D3,D2 Control register of the n set: CRn
- D5,D4 Data register of the n set: DRn
- D7,D6 Control register of the n+1 set: CRn+1
- D9,D8 Data register of the n+1 set: DRn+1
- :
- D31,D30 Control register of the n+7 set: CRn+7
- D33,D32 Data register of the n+7 set: DRn+7
- D50 Data length: 2+m*4 (1 set of registers: 4 words), m: number of sets. Max: 8 sets.

(1) Setting format of control registers (CRn) for high-speed Compare function

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Settings					Output terminals and counters				Output action		Compare method		Data source for comparison			

	Bit	SV	20PM	10PM
Data source for comparison	[0-3]	0	CP of X axis	CP of X axis
		1	CP of Y axis	CP of Y axis
		2	CP of Z axis	CP of Z axis
		3	C200 count value	CP of A axis
		4	C204 count value	C200 count value
		5	—	C204 count value
		6	—	C208 count value
Compare method	[5-4]	1	=	=
		2	≥	≥
		3	≤	≤
Output action	[7-6]	0	Set	Set
		1	Reset	Reset
		2, 3	—	No action
Output terminal and counters	[11-8]	0	CLR0	Y0
		1	CLR1	Y1
		2	Y2	Y2

14 High Speed Compare and Capture

	3	Y3	Y3
	4	Reset C200	Reset C200
	5	Reset C204	Reset C204
	6	Finish Lead state of CAM	Reset C208
	7	Engage CAM	Reset C212

(2) Setting format of control registers (CRn) for high-speed Capture function

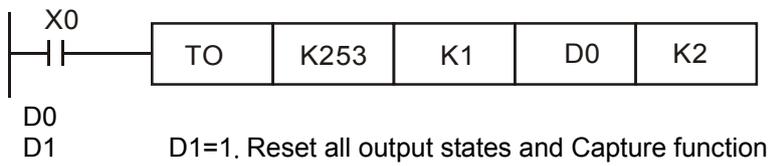
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Settings	Trigger switch of Capture function				Output terminals and counters				Output action		Capture function		Data source of Capture function			

	Bit	SV	20PM	10PM
Data source for capture function	[3-0]	0	CP of X axis	CP of X axis
		1	CP of Y axis	CP of Y axis
		2	CP of Z axis	CP of Z axis
		3	C200 count value	CP of A axis
		4	C204 count value	C200 count value
		5	—	C204 count value
		6	—	C208 count value
		7	—	C212 count value
Capture function	[5-4]	0	Default:0	
Output action	[7-6]	0	—	No action
		1	—	
		2,	—	Set
		3	—	Reset
Output terminal and counters	[11-8]	0	—	Y0
		1	—	Y1
		2	—	Y2
		3	—	Y3
		4	—	Reset C200
		5	—	Reset C204
		6	—	Reset C208
		7	—	Reset C212
Trigger switch of Capture function	[15-12]	0	X axis PG0	X0
		1	X axis MPGB	X1
		2	X axis MPGA	X2
		3	X axis LSN	X3
		4	X axis LSP	X4
		5	X axis DOG	X5

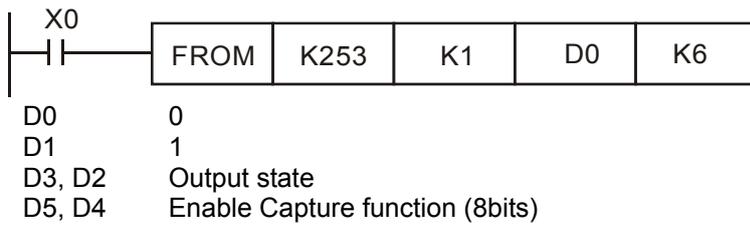
14 High Speed Compare and Capture

	6	X axis STOP	X6
	7	X axis START	X7
	8	Y axis PG0	X8
	9	Y axis MPGB	X9
	10	Y axis MPGA	X10
	11	Y axis LSN	X11
	12	Y axis LSP	—
	13	Y axis DOG	—
	14	Y axis STOP	—
	15	Y axis START	—

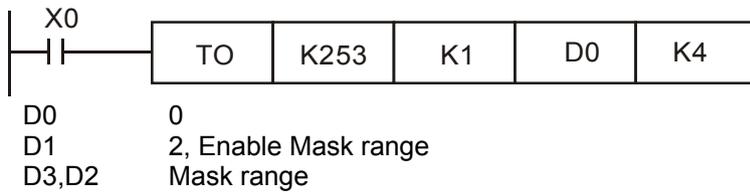
■ Setting initial status



■ Read initial status

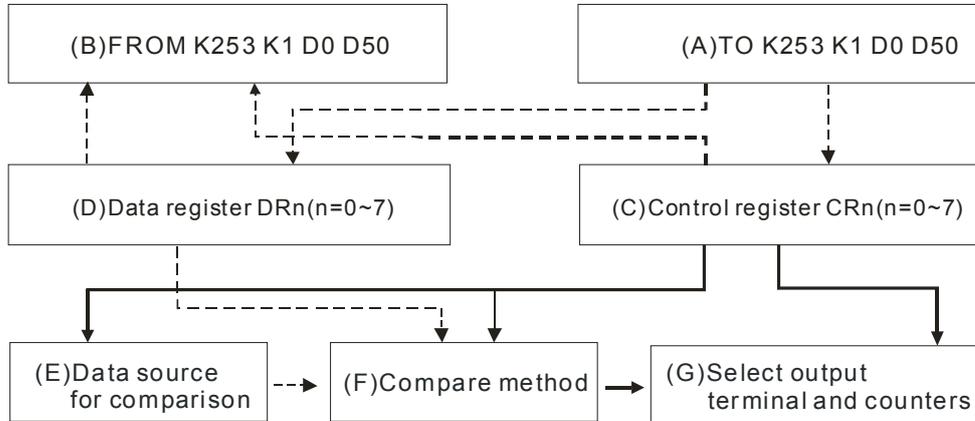


■ Setting Mask area



14.2 High Speed Compare Function

High speed Compare applies FROM/TO instructions for accessing the comparison method and comparison data. The comparison function is illustrated as the below diagram.



※Dashed lines indicate data access process and solid lines indicate the control setting process

Block A: Use TO instruction to write data into control registers (Block C) and data registers (Block D)

Block B: Use FROM instruction to read data from control registers (Block C) and data registers (Block D)

Block C: According to the data specified by TO instruction, control registers set up the data source (Block E), Compare method (Block F) and output terminal setting (Block G).

Block D: Data registers store the data specified by TO instruction and Compare the data with the data source in Block E.

Block E: There are 5 sets of data source including current position of X/Y/Z axis, count value of C200 (current position of Master / MPG of X-axis) and count value of C204 (MPG of Y-axis). For settings of high speed counters C200 and C204, please refer to Ch3 of this manual.

Block F: Compare method: compare the data in Block D and Block E by 3 conditions: \geq , \leq , and $=$. When one of the conditions is met and the trigger method is set as "Set" (Bit7-6=0), the output terminals (CLR0, CLR1, Y2, Y3) and counters C200/C204 (when RST signal is triggered, count value will be cleared and cannot resume counting) will be set ON. When trigger method is set as "Rst" (Bit7-6=1), the output terminals and counters C200/C204 (ON state will be reset and can resume counting) will be reset.

Note: In this case, the LEDs will not respond when CLR0, CLR1, Y2 or Y3 is ON. To identify their status, connect CLR0-, CLR1-, C2 with 24G; CLR0+, CLR1+, Y2 and Y3 with Xn, and S/S2 with +24V. By this wiring method the LED of the corresponding X point will be ON when one of the terminals are ON.

Block G: When the conditions of comparison are met, one of the 6 output settings including CLR0, CLR1, Y2 (00M), Y3 (00M), C200 and C204 will be set ON or reset.

Process of comparison:

Set up compare settings and comparison data (Block A) → Compare specified data (Block D) with data source (Block E) by the conditions set in Block F → Set/Rst output terminals or counters C200/C204 according to settings in Block G.

14 High Speed Compare and Capture

Program example:

In this example high speed counter C204 is used. When count value exceeds K100, CLR1 will be ON; when count value exceeds K300, CLR1 will be reset. Therefore, 2 sets of comparison is designed in this example, one set triggers CLR1 and another resets CLR1. In comparison application, the LED of CLR1 will not respond when CLR1 is ON. To monitor CLR1 state, connect CLR1- with 24G, CLR1+ with X7, S/S2 with +24V and an additional MPG is required. Please see below for the example program.

Comparison initial setting: apply 2 sets of Compare (n=0, n=1). FROM/TO 10 pieces of data.

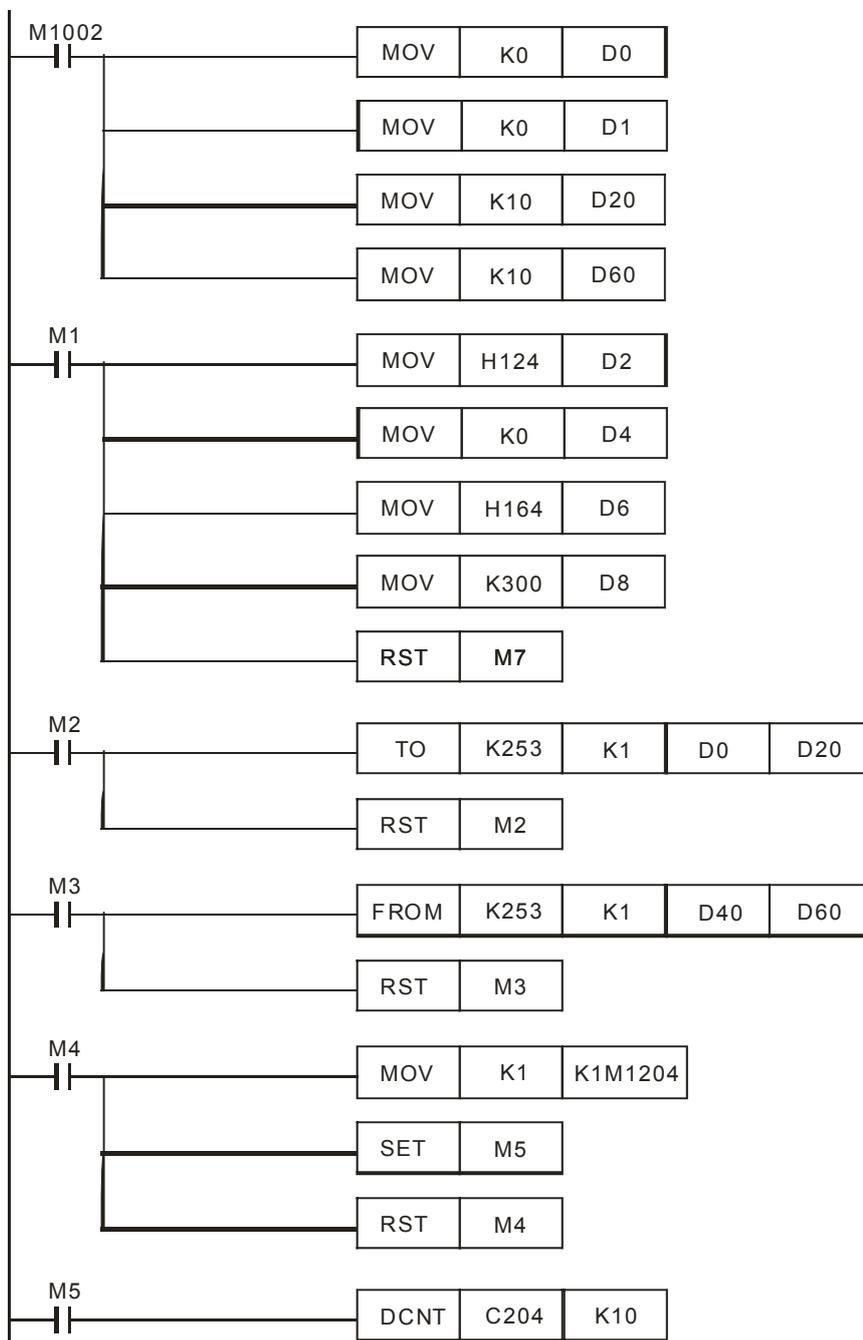
Compare setting:

n=0, Source: C204(b[3:0]=4); Compare method: \geq (b[5:4]=2); SET(b[7:6]=0); output terminal: CLR1(b[11:8]=1);

Compare value: 100

n=1, Source: C204(b[3:0]=4); Compare method: \geq (b[5:4]=2); RST(b[7:6]=1); output terminal: CLR1(b[11:8]=1);

Compare value: 300



14 High Speed Compare and Capture

Sequential control process:

Step 1: Initial setting for high speed Compare when O100 is activated:

- (1) D0 = 0 → Specify the number of the starting set. n = 0 indicates the first set.
- (2) D1 = 0
- (3) D20 = 10 → Set up data length for TO instruction: 10 words, including 2 sets of comparison data
- (4) D60 = 10 → Set up data length for FROM instruction: 10 words, including 2 sets of comparison data.

Step 2: Set M1. Specify the comparison data in control registers and data registers.

- (1) Control registers of the 1st set: D3, D2 = H124 → Data source: C204(Bit3-0=4); compare method: \geq , SET (Bit5-4=2) (Bit7-6=0); output terminal: CLR0 (Bit11-8=1).
- (2) Data registers of the 1st set: D5, D4 = K100 → output terminal will be triggered when source value \geq K100.
- (3) Control registers of the 2nd set: D7, D6 = H164 → Data source: C204(Bit3-0=4); compare method: \geq , RST (Bit5-4=2) (Bit7-6=1); output terminal: CLR0 (Bit11-8=1).
- (4) Data registers of the 2nd set: D9, D8 = K300 output terminal will be triggered when source value \geq K300.

Step 3: Set M2. Activate the 2 sets of high speed Compare

Step 4: Set M3. Read the data set up in D0~D9 as below:

Device No.	Radix	Value	Comment
C204	d32u	0	MPG value of Y axis
D44	d32u	100	Read the 1st set of comparison data registers: D4, D5
D48	d32u	300	Read the 2nd set of comparison data registers: D8, D9
D40	d16u	0	Read D0
D41	d16u	0	Read D1
D42	h32	00000124	Read the 1st set of comparison control registers: D2, D3
D44	d32u	100	Read the 1st set of comparison data registers: D4, D5
D46	h32	00000164	Read the 2nd set of comparison control registers: D6, D7
D48	d32s	300	Read the 2nd set of comparison data registers: D8, D9

Step 5: Set M4 to set C204 counting mode as P/D (Pulse/direction) then set M5 to enable C204.

Step 6: Rotate the MPG to check if C204 is counting as below.

Devic...	Ra...	Value	Comment
C204	d32u	95	MPG value of Y axis
D44	d32u	100	Read the 1st set of comparison data registers: D4, D5
D48	d32u	300	Read the 2nd set of comparison data registers: D8, D9

14 High Speed Compare and Capture

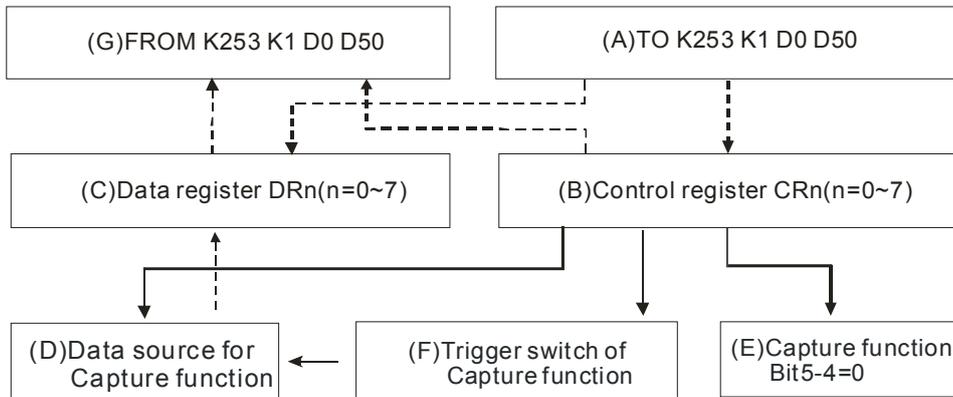
Step 7: Keep rotating the MPG. When the value of C204 exceeds K100, monitor the LED of X7 to check if CLR1 is triggered.

Step 8: Keep rotating the MPG. When the value of C204 exceeds K300, monitor the LED of X7 to check if CLR1 is reset.

14 High Speed Compare and Capture

14.3 Capture Function

Reading current position of X/Y/Z axis or count value of C200/C204 in program scan results in error due to the program scan time. To avoid the error, we can use Capture function to read data immediately through external input signal. The Capture function is illustrated as the below diagram.



Block A: Use TO instruction to write data into control registers (Block B) and data registers (Block C).

Block B: According to the data specified by TO instruction, control registers set up the data source (Block D), enable the Capture function (Block E) and select the trigger switch (Block F).

Block C: Data registers store the data specified by TO instruction or the caught value from the data source specified in Block D.

Block D: There are 5 sets of data source including current position of X/Y/Z axis, count value of C200 (current position of Master / MPG of X-axis) and count value of C204 (MPG of Y-axis). For settings of high speed counters C200 and C204, please refer to Ch3 of this manual.

Block E: Enable Capture function

Block F: Select trigger switch of Capture function

Block G: Use FROM instruction to read data from control registers (Block B) and data registers (Block C) which stores the caught value.

Process of Capture function:

Write in settings of Capture function by TO instruction (Block A). → External input signal triggers ON (Block F) → Capture the value from Block D and store the caught data in Block C → Read the caught data by FROM instruction (Block G).

Program example:

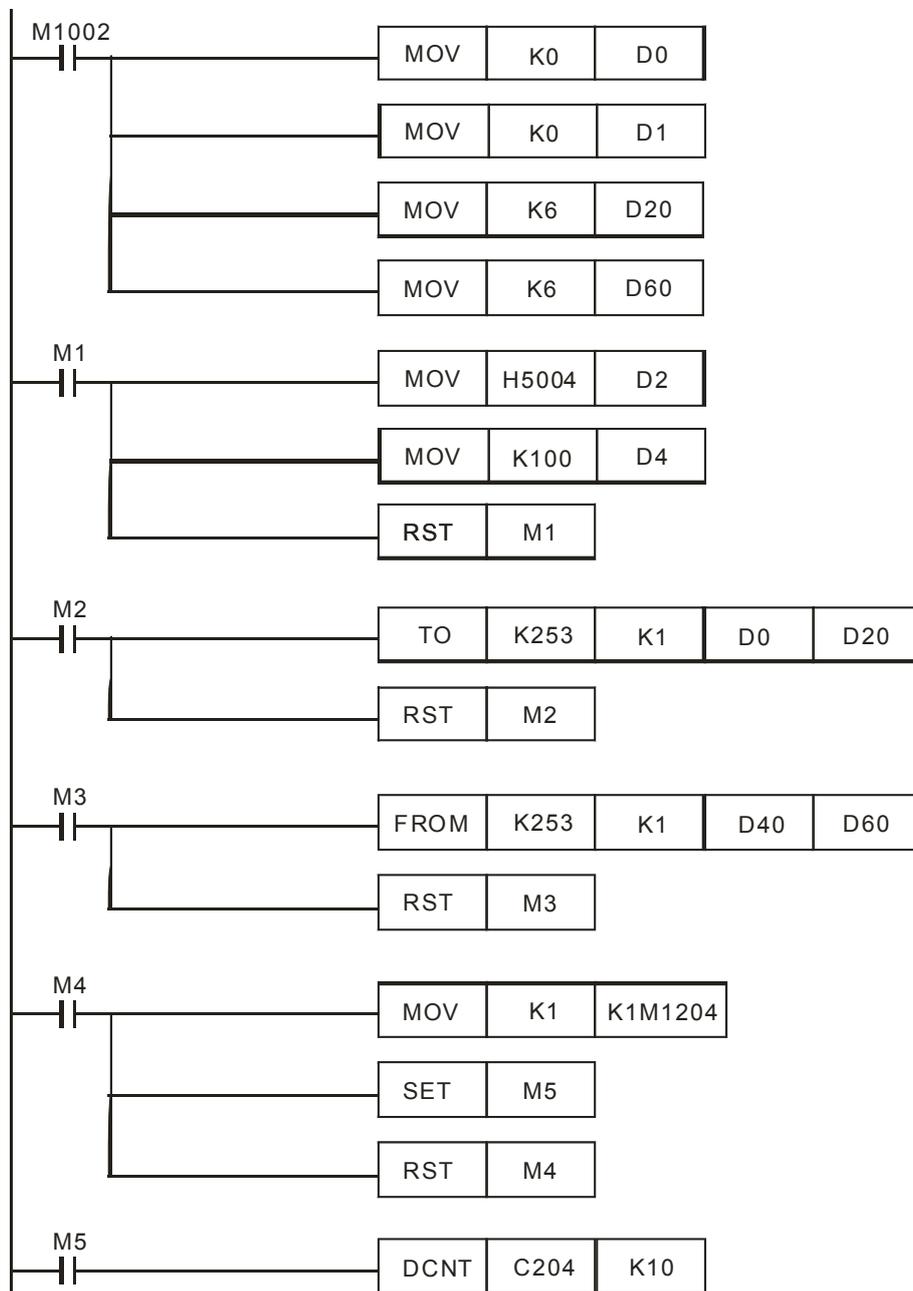
In this example high speed counter C204 is used and high speed counting is enabled. Trigger signal is set as DOG0 to capture the count value of C204 when MPG is rotating. Please see below for the example program.

14 High Speed Compare and Capture

Capture initial setting: apply 1 set of Capture (n=0). FROM/TO 6 pieces of data.

Capture setting:

n=0, Source: C204(b[3:0]=4); Enable Capture function(b[5:4]=0); Trigger switch of Capture function: X_DOG(b[15:12]=5).



Sequential control process:

Step 1: Initial setting for Capture function when O100 is activated to set ON M1002:

- (1) D0 = 0 → Specify the number of the starting set. n = 0 indicates the first set.
- (2) D1 = 0
- (3) D20 = 6 → Set up data length for TO instruction: 6 words, 1 set of data for Capture function
- (4) D60 = 6 → Set up data length for FROM instruction: 6 words, 1 set of data for Capture function.

Step 2: Set M1. Specify the data for Capture function in control registers and data registers.

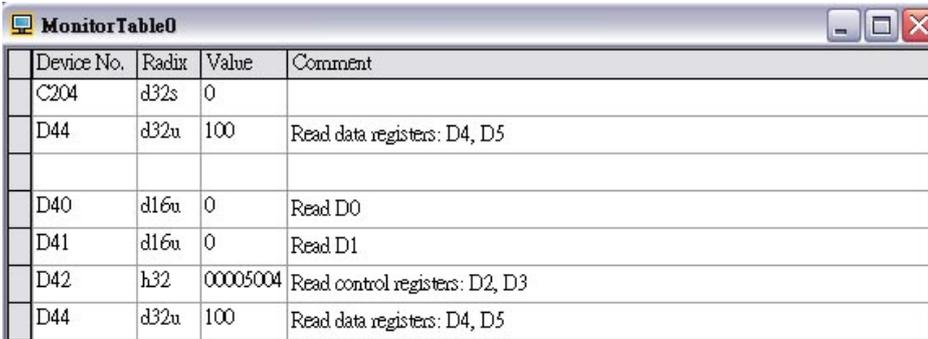
14 High Speed Compare and Capture

(1) Control registers: D3, D2 = H5004 → Data source: C204(Bit3-0=4); Enable Capture function (Bit5-4=0); Trigger switch: X axis _DOG0(Bit15-12=5).

(2) Data registers: D5, D4 = K100 → can be specified with a random value.

Step 3: Set M2. Active Capture function by TO instruction

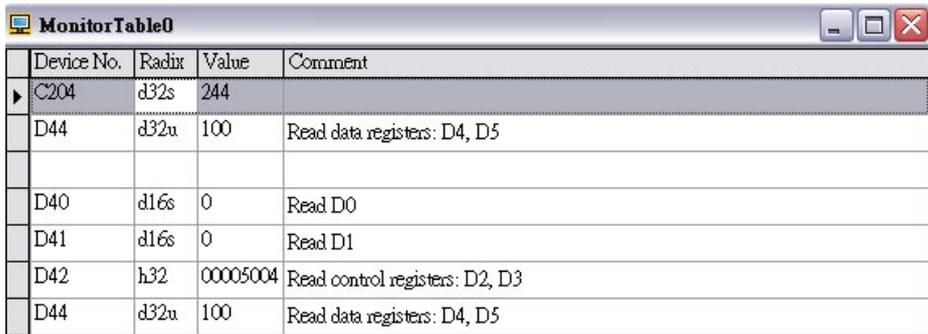
Step 4: Set M3. Read the data set up in D0~D5 as below:



Device No.	Radix	Value	Comment
C204	d32s	0	
D44	d32u	100	Read data registers: D4, D5
D40	d16u	0	Read D0
D41	d16u	0	Read D1
D42	h32	00005004	Read control registers: D2, D3
D44	d32u	100	Read data registers: D4, D5

Step 5: Set M4 to set C204 counting mode as P/D (Pulse/direction) then set M5 to enable C204.

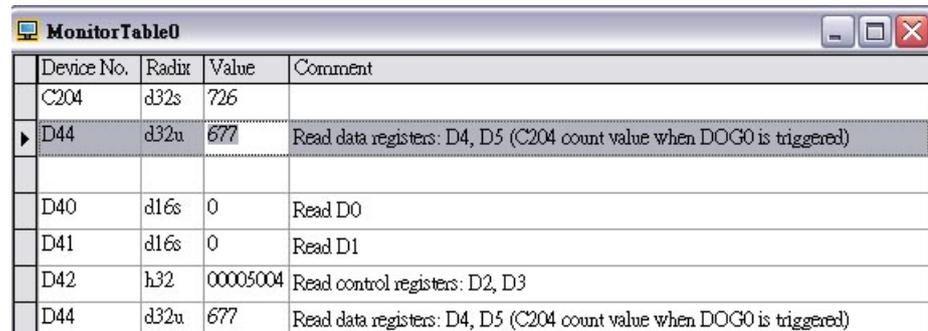
Step 6: Rotate the MPG to check if C204 is counting as below.



Device No.	Radix	Value	Comment
C204	d32s	244	
D44	d32u	100	Read data registers: D4, D5
D40	d16s	0	Read D0
D41	d16s	0	Read D1
D42	h32	00005004	Read control registers: D2, D3
D44	d32u	100	Read data registers: D4, D5

Step 7: Keep rotating the MPG and trigger DOG0 to capture the data immediately.

Step 8: Set M3 to read the the caught value. D44 stores C204 count value when DOG0 is triggered.



Device No.	Radix	Value	Comment
C204	d32s	726	
D44	d32u	677	Read data registers: D4, D5 (C204 count value when DOG0 is triggered)
D40	d16s	0	Read D0
D41	d16s	0	Read D1
D42	h32	00005004	Read control registers: D2, D3
D44	d32u	677	Read data registers: D4, D5 (C204 count value when DOG0 is triggered)

15.1 Appendix A: Error Codes

If error occurs in O100 or OX program area in DVP-PM, the error flag will be ON and ERROR indicator will flash. The causes of errors could be illegal use of instruction operand (device), program syntax error, or improper setting of motion subroutine parameters. Users can monitor the error register and obtain the cause of error according to the error code in error registers.

◆ Error codes in all program areas and motion axes & devices for displaying steps

Program area	O100				OX			
Error type	Program error	Motion error			Program error	Motion error		
		X axis	Y axis	Z axis		X axis	Y axis	Z axis
Error flag	M1953	M1793	M1873	M2033	M1793	M1793	M1873	M2033
Error register	D1802	D1857	D1937	D2017	D1857	D1857	D1937	D2017
Number of error step	D1803	D1869			D1869	D1869		

◆ Error codes for program area and motion axis (in hex)

Code	Cause	Code	Cause
0002	No content in the applied subroutine	0031	Forward pulses prohibited
0003	No corresponding Pn for CJ, CJN, JMP	0032	Reverse pulses prohibited
0004	There is a subroutine flag in main program	0033	Left/right limit achieved
0005	No subroutine	0040	Incorrect use range of device
0006	Repeated index in the same program	0041	MODRD and MODWR communication time-out
0007	Repeated subroutine index	0044	V/Z modification error
0008	Repeated index of jump instructions in different subroutines	0045	Floating point conversion error
0009	JMP instruction and CALL instruction are using the same flag.	0E18	BCD conversion error
000A	The index is the same as subroutine index	0E19	Arithmetic error in division (divisor = 0)
0011	Incorrect target position (I) setting	C401	General loop error
0012	Incorrect target position (II) setting	C402	Continuous use of LD/LDI for more than 9 times
0021	Incorrect running speed (I) setting	C404	RPT ~ RPE are used more than 1 layer
0022	Incorrect running speed (II) setting	C405	SRET is used between RPT ~ RPE
0023	Incorrect zero return speed (V_{RT}) setting	C4EE	There is no end instruction (M102, M2) in the program
0024	Incorrect zero return deceleration speed (V_{CR}) setting	C4FF	No such instruction / Incorrect operand format or incorrect range
0025	Incorrect JOG speed setting		

15.2 Appendix B: Manual Revision History

Item	Revision	Chapter
1	Added Z axis into hardware wiring diagram.	2
2	Added maximum input current and output current in electrical specifications.	2
3	Added and modified explanations on special D registers: D1798, D1864, D1846, D1832 ~ D1834, D1945, D1948 ~ D1949, D1952 ~ D1954, D2025, D2028 ~ D2029, D2032 ~ D2035 Added and modified explanations on special M registers: M1035, M1036, M1811, M1905, M1956, M2000, M2036, M2048 ~ M2051	3
4	Added explanations on DRV (3-axis high-speed positioning), LIN (3-axis synchronous linear interpolation), CW/CCW (helical movement) and X-Y/X-Z/Y-Z surface setup.	6
5	Added 20D and 20M model indications.	6
6	Added G-Code adoptable variable designated by D register.	6
7	Added Ch. 8.4 Connecting DVP-PM (Master) to DVP01PU-H2 (Slave) for 3 rd Axis Control (For DVP20PM00D only).	8
8	Added Ch. 9 E-CAM.	9
9	Modified wiring diagram of DVP-PM with servo drives of different brands.	2
10	Modified explanations on the wiring of differential input.	2
11	Modified Menu→Auto of M1796 to On.	3
12	Deleted O100, M102, Oxn/M2, Pn/SRET in the ladder diagram example.	1, 3, 5, 6, 7, 8
13	Deleted NOP and its explanation in other instructions.	4
14	Deleted “not including V, Z devices” in instruction table head.	6
15	Added explanations on PEP setting in Ch.10.	10
16	Added explanations on G-code application in Ch 11	11
17	Added explanations on POU editing mode in Ch 12	12
18	Added explanations on CANopen communication card in Ch 13	13

15.3 Appendix C: Function Comparison Table of DVP-PM Models

Function		DVP-20PM 00D	DVP-20PM 00M	DVP-10PM	Chapter/Section
Program	O100 main program	○	○	○	1.1
	OX motion subroutine	○	○	X	1.2
	P subroutine	○	○	○	1.3
Instruction	Basic / API	130	130	130	Ch4
	Motion (OX)	22	22	X	6.2
	G-Code	10	10	X	6.4
	M-Code	65536	65536	X	6.3

Single axis motion	JOG	2 axes	3 axes	4 axes	3.11
	Zero return	2 axes	3 axes	4 axes	3.11
	Variable speed	2 axes	3 axes	4 axes	3.11
	Single speed positioning	2 axes	3 axes	4 axes	3.11
	Single-speed positioning interruption	2 axes	3 axes	4 axes	3.11
	2-speed positioning	2 axes	3 axes	4 axes	3.11
	2-speed positioning interruption	2 axes	3 axes	4 axes	3.11
	Externally triggering single-speed positioning	X	3 axes	X	3.11
	MPG input	2 axes	3 axes	4 axes	3.11
	Cyclic/Acyclic E-CAM	2 axes	3 axes	X	3.11
PWM	X	X	4 axes	3.11	
High speed Counter	High speed capture	8 sets	8 sets	8 sets	14.2
	High speed compare	8 sets	8 sets	8 sets	14.3
	High speed timing	X	X	6 sets	3.11
Expansion cards	FPMC	○	○	○	Ch13
	FPMSD	○	○	○	-

MEMO