Smarter. Greener. Together.

Industrial Automation Headquarters

Delta Electronics, Inc.
Taoyuan Technology Center
No.18, Xinglong Rd., Taoyuan City,
Taoyuan County 33068, Taiwan
TEL: 886-3-362-6301 / FAX: 886-3-371-6301

Asia

Delta Electronics (Jiangsu) Ltd.
Wujiang Plant 3
1688 Jiangxing East Road,
Wujiang Economic Development Zone
Wujiang City, Jiang Su Province, P.R.C. 215200
TEL: 86-512-6340-3008 / FAX: 86-769-6340-7290

Delta Greentech (China) Co., Ltd.
238 Min-Xia Road, Pudong District,
ShangHai, P.R.C. 201209
TEL: 86-21-58635678 / FAX: 86-21-58630003

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

Delta Electronics (Korea), Inc.
1511, Byucksan Digital Valley 6-cha, Gasan-dong,
Geumcheon-gu, Seoul, Korea, 153-704
TEL: 82-2-515-5303 / FAX: 82-2-515-5302

Delta Electronics Int'l (S) 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
Gurgaon, PIN 122001, Haryana, India
TEL : 91-124-4874900 / FAX : 91-124-4874945

Americas

Delta Products Corporation (USA)
Raleigh Office
P.O. Box 12173,5101 Davis Drive,
Research Triangle Park, NC 27709, U.S.A.
TEL: 1-919-767-3800 / FAX: 1-919-767-8080

Delta Greentech (Brasil) S.A.
Sao Paulo Office
Rua Itapeva, 26 - 3° andar Edificio Itapeva One-Bela Vista
01332-000-São Paulo-SP-Brazil
TEL: 55 11 3568-3855 / FAX: 55 11 3568-3865

Europe

Deltronics (The Netherlands) B.V.
Eindhoven Office
De Witbogt 20, 5652 AG Eindhoven, The Netherlands
TEL: 31-40-2592850 / FAX: 31-40-2592851

DVP-0099720-01

2014-02-12

DVP-20PM **Application Manual (Programming)**

DVP-20PM
Application Manual (Programming)

www.deltaww.com

Smarter. Greener. Together.

DVP-20PM Application Manual

Contents

Chapter 10 Application of G-codes

Chapter 11 CANopen Communication Card

Delta DVP-PM series motion controllers can put axes in particular positions at high-speeds, create linear interpolations, and circular interpolations. They can execute basic instructions, applied instructions, motion instructions, and G-codes. Different DVP-PM series motion controllers support different program frameworks and functions. The functions that DVP-PM series motion controllers support are shown in the table below.

Function		DVP-20PM	DVP-10PM
Program	Main program O100	○	○
	Ox motion subroutines	○	×
	P subroutines	○	○
Instruction	General instructions/Applied instructions	○	○
	Motion instructions	○	×
	G-codes	○	×
	M-codes	○	×
Uniaxial motion	JOG motion	○	○
	Returning home	○	○
	Variable motion	○	○
	Single-speed motion	○	○
	Inserting single-speed motion	○	○
	Two-speed motion	○	○
	Inserting two-speed motion	○	○
	Triggering single-speed motion by means of an external signal	○	×
	Manual pulse generator mode	○	○
	Cyclic/Noncyclic electronic cam motion	○	×

In this chapter, the basic program frameworks of DVP-PM series motion controllers are described. Owing to the fact that the functionality of a DVP-PM series motion controller is composed of sequence control and positioning control, a program comprises O100, Ox motion subroutines, and P subroutines. O100, Ox motion subroutines, and P subroutines are described in this chapter. Basic instructions, applied instructions, motion instructions, and G-codes will be introduced in other chapter 4~chapter 6. The specifications for DVP-PM series motion controllers are shown in the table below.

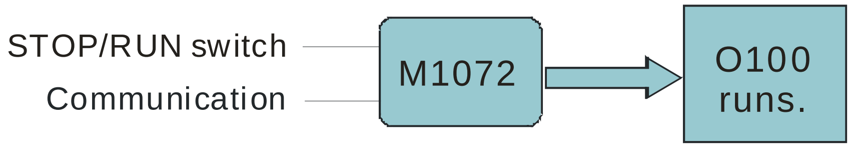
Specifications	DVP-10PM	DVP-20PM
High-speed output	4 axes (1000 kHz)	3 axes (500 kHz)
PWM	Precision: 0.3%@200 kHz	-
High-speed counter	6 input terminals (2 differential input terminals, and 4 input terminals whose collectors are open collectors)	2 input terminals
Program capacity	64K steps	64K steps
Execution speed	LD: 0.14 us MOV: 2 us DMUL: 7.6 us DEMUL: 6.1 us	-

1.1 Structure of O100

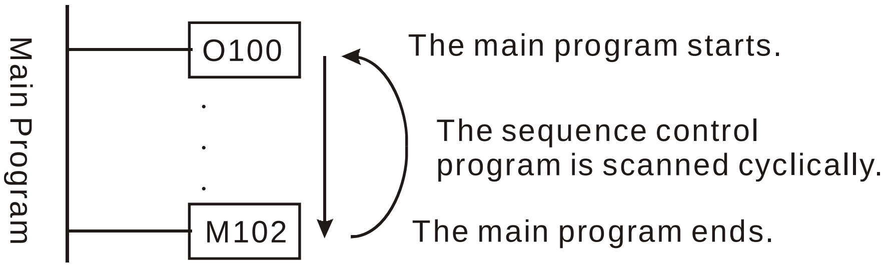
O100 is a sequence control program. It is the main program in a DVP-PM series motion controller. It only supports basic instructions and applied instructions. Users can use these two types of instructions to process I/O data, call P subroutines, and enable Ox motion subroutines (Ox0~Ox99). O100 functions as a main program. Motion subroutines are enabled through O100. There is hierarchical relation between O100 and motion subroutines. The characteristics of O100 are described below.

1. There are two methods of enabling O100.
   - If the STOP/RUN switch of a DVP-PM series motion controller is turned from the “STOP” position to the “RUN” position when the DVP-PM series motion controller is powered, M1072 will be ON, and O100 will run.

- If a DVP-PM series motion controller is powered, users can use communication to set M1072 to ON, and to run O100.



2. O100 is scanned cyclically. The scan of the main program O100 starts from the starting flag O100. After the ending instruction M102 is scanned, the scan of the main program O100 will go back to the starting flag O100.



3. There are three methods of disabling O100.
   - If the STOP/RUN switch of a DVP-PM series motion controller is turned form the “RUN” position to the “STOP” position when the DVP-PM series motion controller is powered, M1072 will be OFF, and O100 will stop. If O100 stops, Ox motion subroutines and P subroutines will not be executed.
   - If a DVP-PM series motion controller is powered, users can use communication to set M1072 to OFF, and to stop O100. If O100 stops, Ox motion subroutines and P subroutines will not be executed.
   - If an error occurs when O100 is compiled or when O100 runs, O100 will stop automatically. Please refer to appendix A in chapter 15 for more information about error codes.
4. O100 supports basic instructions and applied instructions. Users can write a control program according to their needs. They can set the parameters of motion instructions, and motion subroutine numbers (Ox0~Ox99) in O100.
   - O100 does not support motion instructions and G-codes. Motion instructions and G-codes must be used in the motion subroutines Ox0~Ox99. Please refer to section 1.2 for more information.
   - O100 can call P subroutines. Please refer to section 1.3 for more information.
5. The description of O100 is shown below.

O100	Description
Enabling O100	Starting flag O100 (If O100 is a ladder diagram in PMSoft, the starting flag in O100 will be set automatically, and users do not have to write the starting flag.)
Disabling O100	Ending instruction M102 (If O100 is a ladder diagram in PMSoft, the ending instruction M102 will be set automatically, and users do not have to write the ending instruction M102.)
Executing O100	1. The STOP/RUN switch of a DVP-PM series motion controller is turned form the “STOP” position to the “RUN” position. 2. Users use communication to set M1072 to ON.
Operation characteristic	O100 is scanned cyclically.
Instructions supported	Basic instructions and applied instructions are supported.
Number	There is only one O100 in a program.
Characteristic and function	1. It is a sequence control program. 2. It can enable the motion subroutines Ox0~Ox99, and call P subroutines. 3. If O100 is used with Ox motion subroutines and P subroutines, O100, the Ox motion subroutines, and the P subroutines can be arranged in any order.

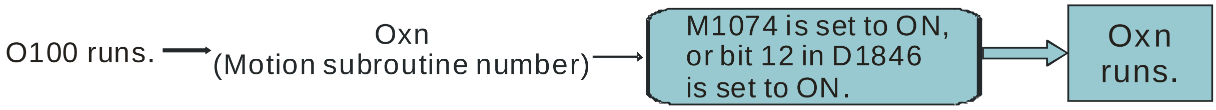
1.1.1 Manual Function of O100

Users can set manual motion modes by means of special registers in O100. (Please refer to section 3.12 for more information.)

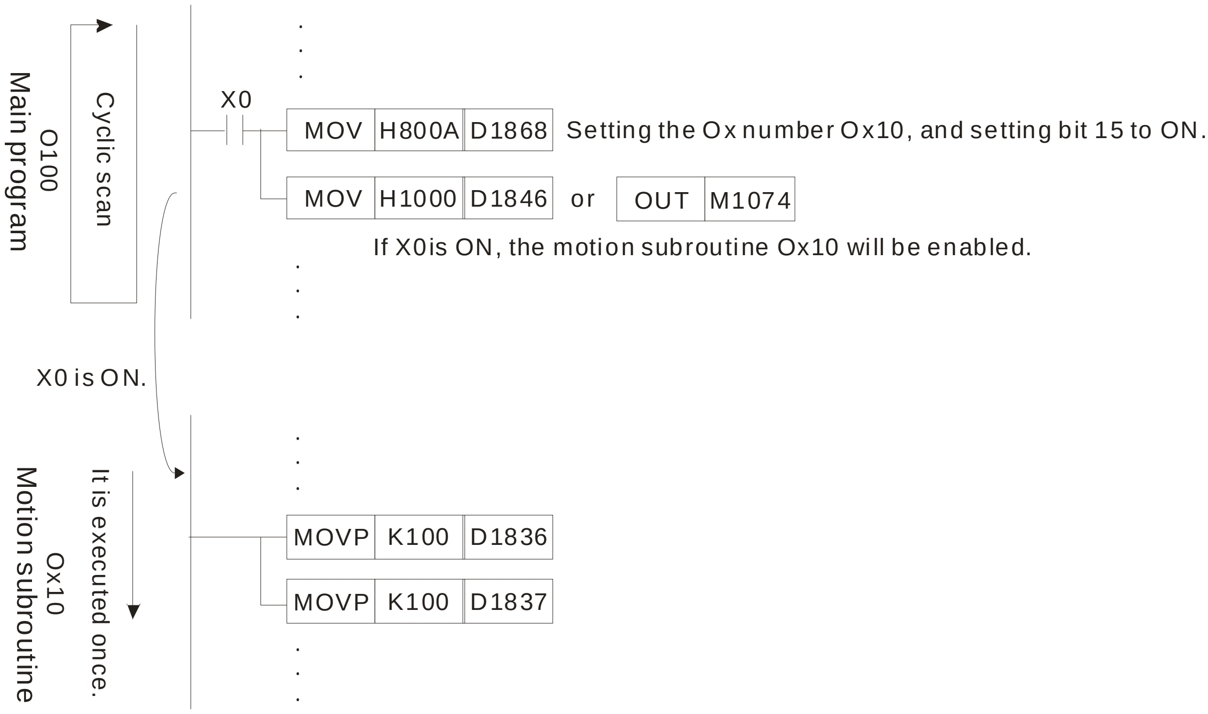
1.2 Structure of Ox Motion Subroutines

The motion subroutines Ox0~Ox99 are motion control programs. They are subroutines which control the motion of the axes of a DVP-PM series motion controller. Ox0~Ox99 support basic instructions, applied instructions, motion instructions, and G-codes. They can call P subroutines. Users can control the paths of the axes of a DVP-PM series motion controller through Ox motion subroutines. The characteristics of Ox motion subroutines are described below.

1. There are two methods of enabling an Ox motion subroutine.
   - When O100 runs, users can set motion subroutine numbers in O100. (The motion subroutine numbers must be in the range of Ox0 to Ox99. The users can set a motion subroutine number in O100 by setting D1868. The value in D1868 must be in the range of H8000 to H8063.) If the users want to enable an Ox motion subroutine, they have to set M1074 to ON or set bit 12 in D1846 to ON.
   - Before an Ox motion subroutine is enabled, users have to make sure that no Ox motion subroutine runs.



2. Whenever an Ox motion subroutine is enabled, it is executed once. After O100 enables an Ox motion subroutine, the execution of the Ox motion subroutine will start from the starting flag in the Ox motion subroutine. After the ending instruction M2 in the Ox motion subroutine is executed, the execution of the Ox motion subroutine will stop.



   If X0 is ON, the motion subroutine Ox10 will be enabled. After the ending instruction M2 in Ox10 is executed, the execution of Ox10 will stop. (Ox10 is executed once. If Ox10 needs to be executed again, X0 has to be set to ON.)

3. There are four methods of disabling an Ox motion subroutine.
   - If the STOP/RUN switch of a DVP-PM series motion controller is turned form the “RUN” position to the “STOP” position when the DVP-PM series motion controller is powered, M1072 will be OFF, O100 will stop, and Ox motion subroutines will not be executed.
   - Users can stop the execution of Ox motion subroutines by means of the external terminal Stop0.
   - If a DVP-PM series motion controller is powered, users can use communication to set the value in D1846 to 0, or to set M1074 to OFF, and to stop the execution of Ox motion subroutines.
   - If an error occurs when an Ox motion subroutine is compiled or when an Ox motion subroutine is

executed, the execution of the Ox motion subroutine will stop automatically. Please refer to appendix A in chapter 15 for more information about error codes.

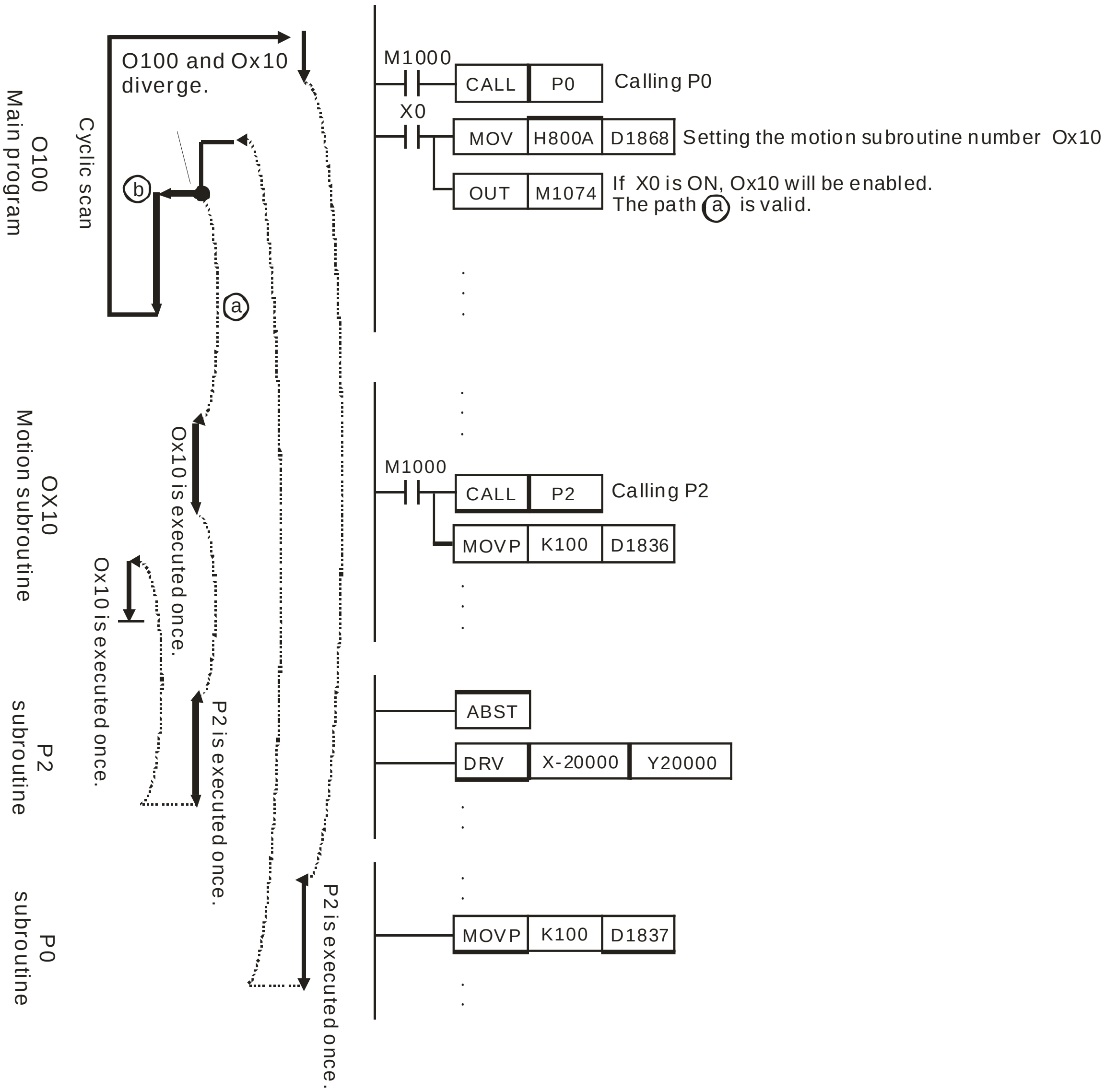
4. An Ox motion subroutine supports basic instructions, applied instructions, motion instructions, and G-codes. Users can write a motion program according to their needs. They can control the motion of the axes of a DVP-PM series motion controller by setting the parameters of the axes.
   - Basic instructions, applied instructions, motion instructions and G-codes must be used in the motion subroutines Ox0~Ox99.
   - Ox motion subroutines can call P subroutines. Please refer to section 1.3 for more information.
5. The description of Ox motion subroutines is shown below.

Ox motion subroutine	Description
Enabling an Ox motion subroutine	There are 100 Ox motion subroutines (Ox0~Ox99). (If an Ox motion subroutine is a ladder diagram in PMSoft, the starting flag in the Ox motion subroutine will be set automatically, and users do not have to write the starting flag.)
Disabling an Ox motion subroutine	Ending instruction M2 (If an Ox motion subroutine is a ladder diagram in PMSoft, the ending instruction M2 will be set automatically, and users do not have to write the ending instruction M2.)
Executing an Ox motion subroutine	1. If users set bit 12 in D1846 or M1074 to ON when O100 runs, an Ox motion subroutine will be enabled. 2. If users use communication to set bit 12 in D1846 or M1074 to ON when O100 runs, an Ox motion subroutine will be enabled. 3. Users can stop the execution of Ox motion subroutines by means of the external terminal Stop0. Note: Before an Ox motion subroutine is enabled, users have to make sure that no Ox motion subroutine runs.
Operation characteristic	Whenever an Ox motion subroutine is enabled, it is executed once. If an Ox motion subroutine needs to be executed again, it has to be enabled again.
Instructions supported	Basic instructions, applied instructions, motion instructions, and G-codes are supported. Note: Users have to avoid using pulse instructions.
Number	There are 100 Ox motion subroutines in a program. If users want to enable a motion subroutine number, they have to set D1868, and set bit 12 in D1846 or M1074 to ON.
Characteristic and function	1. Ox0~Ox99 are motion subroutines. (They can only be enabled by O100.) 2. Ox motion subroutines can be used to control the third axis (the Z-axis). Please refer to section 6.4 for more information about G00 and G01. 3. An Ox motion subroutine can be enabled/disabled by an external terminal, a program, or communication. 4. Ox motion subroutines can call P subroutines. 5. If Ox motion subroutines are used with O100 and P subroutines, the Ox motion subroutines, O100, and the P subroutines can be arranged in any order.

1.3 Structure of P Subroutines

P subroutines are general subroutines. They can be called by O100 and Ox motion subroutines. If P subroutines are called by O100, the P subroutines will support basic instructions and applied instructions. If P subroutines are called by Ox0~Ox99, the P subroutines will support basic instructions, applied instructions, motion instructions, and G-codes. After O100 or an Ox motion subroutine calls a P subroutine, the P subroutine will be executed. After SRET in the P subroutine is executed, the lines under the instruction which calls the P subroutine will be executed.

1. There are two methods of enabling a P subroutine.
   - O100 can call P subroutines.
   - Ox motion subroutines can call P subroutines.
2. Whenever a P subroutine is called, it is executed once. After O100 or an Ox motion subroutine calls a P subroutine, the P subroutine will be executed. After the ending instruction SRET in the P subroutine is executed, the execution of the P subroutine will stop, and the lines under the instruction which calls the P subroutine will be executed.



The subroutine P0 supports basic instructions and applied instructions. The subroutine P2 supports basic instructions, applied instructions, motion instructions, and G-codes.

3. There are three methods of disabling a P subroutine.
   - If the STOP/RUN switch of a DVP-PM series motion controller is turned form the “RUN” position to the “STOP” position when the DVP-PM series motion controller is powered, M1072 will be OFF, O100 will stop, and Ox motion subroutines and P subroutines will not be executed.
   - If a DVP-PM series motion controller is powered, users can use communication to set the value in D1846 to 0, or to set M1074 to OFF, to stop the execution of Ox motion subroutines, and to stop the execution of P subroutines.
   - If an error occurs when a P subroutine is executed, the execution of the P subroutine will stop automatically. Please refer to appendix A in chapter 15 for more information about error codes.
4. If P subroutines are called by O100, the P subroutines will support basic instructions and applied instructions. If P subroutines are called by Ox0~Ox99, the P subroutines will support basic instructions, applied instructions, motion instructions, and G-codes.
5. The description of P subroutines is shown below.

P subroutine	Description
Enabling a P subroutine	There are 256 P subroutines (P0~P255). (If a P subroutine is a ladder diagram in PMSoft, the starting flag in the P subroutine will be set automatically, and users do not have to write the starting flag.)
Disabling a P subroutine	Ending instruction SRET (If a P subroutine is a ladder diagram in PMSoft, the ending instruction SRET will be set automatically, and users do not have to write the ending instruction SRET.)

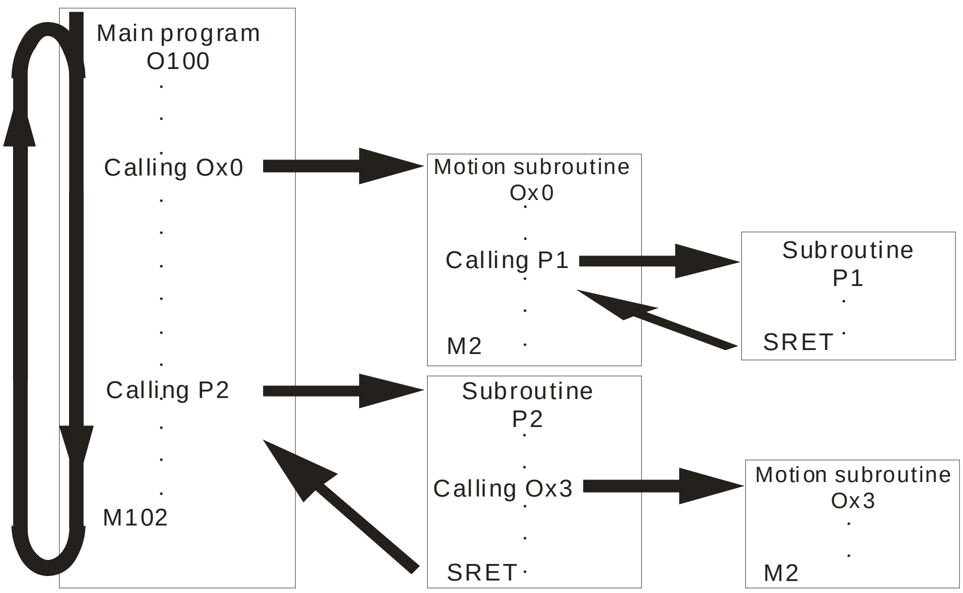
P subroutine	Description
Executing a P subroutine	1. O100 can call P subroutines. 2. Ox motion subroutines can call P subroutines.
Operation characteristic	Whenever a P subroutine is enabled, it is executed once. If a Pn subroutine needs to be executed again, it has to be enabled again.
Instruction supported	1. If P subroutines are called by O100, the P subroutines will support basic instructions and applied instructions. 2. If P subroutines are called by Ox motion subroutines, the P subroutines will support basic instructions, applied instructions, motion instructions, and G-codes. Note: If P subroutines are called by Ox motion subroutines, users have to avoid using pulse instructions.
Number	There are 256 P subroutines in a program.
Characteristic and function	1. P subroutines are general subroutines. 2. P subroutines can be called by O100 and Ox motion subroutines. 3. If P subroutines are used with O100 and Ox motion subroutines, the P subroutines, O100, and the Ox motion subroutines can be arranged in any order.

1.4 Using O100, Ox Motion Subroutines, and P Subroutines

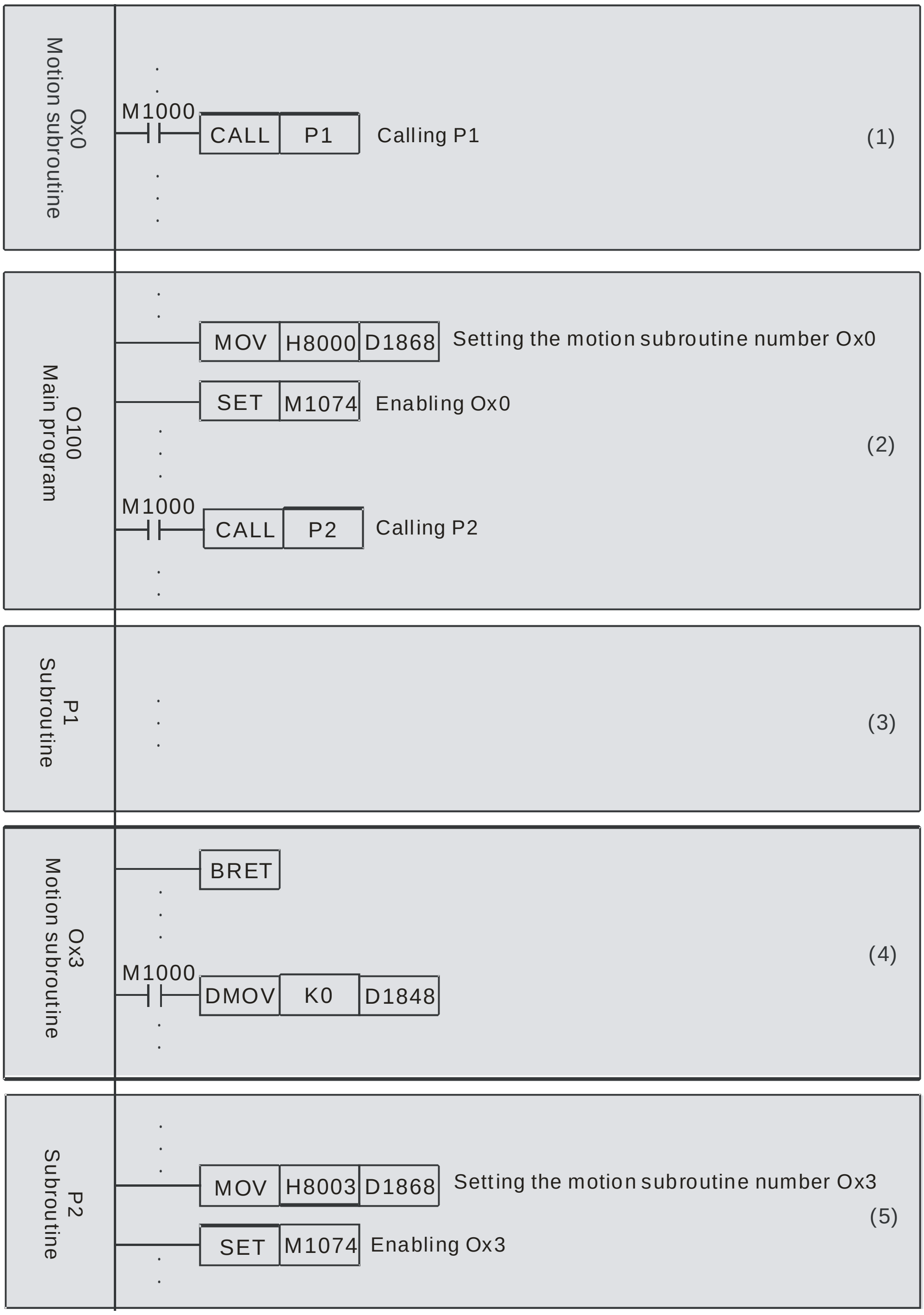
O100, Ox motion subroutines, and P subroutines are introduced in section 1.1~section 1.3. In this section, a program composed of O100, Ox motion subroutines, and P subroutines is described.

1.4.1 Structure of a Program

Suppose a program is composed of O100, Ox0, Ox3, P1, and P2. The five program blocks are shown below.



In order to describe the program, the program is divided into 5 sections (section (1)~section (5)).



The program is described below.

1. Section (1)~section (5) are created in numerical order, but they can be arranged in any order.
2. There is only one O100. O100 can not be called by another program, but it can freely call Ox motion subroutines and P subroutines.
3. Ox motion subroutines can be called by O100 and P subroutines, and it can call P subroutines.
4. P subroutines can be called by O100 and Ox motion subroutines, and it can call Ox motion subroutines.

Note:

1. One Ox motion subroutine is executed at a time. If Ox0 is executed, Ox3 can not be executed. If Ox3 is executed, Ox0 can not be executed.
2. After O100 or a P subroutine enables an Ox motion subroutine, the next line will be executed, and the execution of the Ox motion subroutine will be ignored.

3. Whenever an Ox motion subroutine is enabled, it is executed once. If an Ox motion subroutine needs to be executed again, it has to be enabled again.

The instructions supported by O100, Ox0, Ox3, P1 and P3 are described below. (O: Supported; X: Not supported)

Section	O100	Ox0 and Ox3	P1	P2
Basic instruction	O	O	O	O
Applied instruction	O	O	O	O
Motion instruction	X	O	O	X
G-code	X	O	O	X
Description	-	-	P1 is called by Ox0, and therefore it supports motion instructions and G-codes.	P2 is called by O100, and therefore it does not support motion instructions and G-codes.

Additional remark:

	Main program	Subroutine	Motion subroutine
Order	In any order	In any order	In any order
Execution	It runs normally.	P subroutines can be called by O100 or Ox motion subroutines.	Ox motion subroutines can be called by O100 or P subroutines.
Operation	It is scanned cyclically.	Whenever a subroutine is called, it is executed once.	Whenever a motion subroutine is called, it is executed once.
Number	1 main program	256 subroutines They can be used according to users' needs.	100 motion subroutines They can be used according to users' needs.

2.1 Hardware Specifications

Electrical specifications and wiring are described in this chapter. Please refer to chapter 5~chapter 6 for more information about the writing of a program and the use of instructions. For more information about the peripherals purchased, please refer to the manuals attached to them.

2.1.1 Specifications for Power

Item	20PM
Supply voltage	100~240 V AC (-15%~10%), 50/60 Hz±5%
Fuse	2 A/250 V AC
Power Consumption	60 V A
24 V DC power	500 mA
Power protection	24 V DC output is equipped with a short circuit protection.
Surge voltage withstand level	1500 V AC (Primary-secondary), 1500 V AC (Primary-PE), 500 V AC (Secondary-PE)
Insulation impedance	Above 5 MΩ (The voltage between all input terminals/output terminals and the ground is 500 V DC.)
Noise immunity	ESD: 8 kV air discharge EFT: Power line: 2 kV; digital I/O: 1 kV; analog & communication I/O: 250 V
Ground	The diameter of the ground should not be less than the diameters of the cables connected to the terminals L and N. (If several DVP-20PM series motion controllers are used, please use single-point ground.)
Operation/Storage	Operation:0°C~55°C (Temperature), 5~95% (Humidity), pollution degree 2 Storage: -25°C ~70°C (Temperature), 5~95% (Humidity)
Vibration/Shock resistance	International standards IEC 61131-2, IEC 68-2-6 (TEST Fc)/IEC 61131-2 & IEC 68-2-27 (TEST Ea)
Weight	Approximately 478/688 g

2.1.2 Electrical Specifications for Input Terminals/Output Terminals

Electrical specifications for input terminals:

- DVP20PM00D

Terminal	Description	Response	Maximum input	
			Current	Voltage
START0 and START1	Input terminals for starting DVP20PM00D	10 ms	6 mA	24 V
STOP0 and STOP1	Input terminals for stopping DVP20PM00D	10 ms	6 mA	24 V
LSP0/LSN0 and LSP1/LSN1	Positive limit switches/Left limit switches (for the X-axis and the Y-axis)	10 ms	6 mA	24 V
A0+, A0-, A1+, and A1-	A-phase input terminals for manual pulse generators (differential input terminals)	200 kHz	15 mA	5~24 V
B0+, B0-, B1+, and B1-	B-phase input terminals for manual pulse generators (differential input terminals)	200 kHz	15 mA	5~24 V
PG0+, PG0-, PG1+, and PG1-	PG signals (differential input terminals)	200 kHz	15 mA	5~24 V
DOG0 and DOG1	The use of the input terminals varies with the mode used. 1. DOG signals for returning home 2. Signals for starting the insertion of single-speed motion or the insertion of two-speed motion	1 ms	6 mA	24 V
X0~X7	General input terminals	200 kHz	15 mA	24 V

■ DVP20PM00M

Terminal	Description	Response	Maximum input	
			Current	Voltage
START0 and START1	Input terminals for starting DVP20PM00M	10 ms	6 mA	24 V
STOP0 and STOP1	Input terminals for stopping DVP20PM00M	10 ms	6 mA	24 V
LSP0/LSN0 and LSP1/LSN1	Positive limit switches/Left limit switches (for the X-axis and the Y-axis)	10 ms	6 mA	24 V
X1/X2	Positive limit switch/Left limit switch (common-point ground) (for the Z-axis)	10 ms	6 mA	24 V
A0+, A0-, A1+, and A1-	A-phase input terminals for manual pulse generators (differential input terminals) (A1+ and A1- are for the Y-axis and the Z-axis.)	200 kHz	15 mA	5~24 V
B0+, B0-, B1+, and B1-	B-phase input terminals for manual pulse generators (differential input terminals) (B1+ and B1- are for the Y-axis and Z-axis.)	200 kHz	15 mA	5~24 V
PG0+, PG0-, PG1+, and PG1-	PG signals (differential input terminals)	200 kHz	15 mA	5~24 V
X3	PG signal (common-point ground) (for the Z-axis)	10 ms	6 mA	24 V
DOG0 and DOG1	The use of the input terminals varies with the mode used. 1. DOG signals for returning home 2. Signals for starting the insertion of single-speed motion or the insertion of two-speed motion	1 ms	6 mA	24 V
X0	The use of the input terminal varies with the mode used. 1. DOG signal for returning home (common-point ground) (for the Z-axis) 2. Signal for starting the insertion of single-speed motion or the insertion of two-speed motion (common-point ground) (for the Z-axis)	10 ms	6 mA	24 V
X4~X7	General input terminals	200 kHz	15 mA	24 V

Electrical specifications for output terminals:

■ DVP20PM00D

Terminal	Description	Response	Maximum current output
CLR0+, CLR0-, CLR1+, and CLR1-	CLR signals (for clearing the present positions of servo drives which are stored in registers in the servo drives)	10 ms	20 mA
FP0+, FP0-, FP1+, and FP1-	U/D: Counting up P/D: Pulse A/B: A phase	500 kHz	40 mA
RP0+, RP0-, RP1+, and RP1-	U/D: Counting down P/D: Direction A/B: B phase	500 kHz	40 mA
Y0~Y7	General output terminals	200 kHz	40 mA

■ DVP20PM00M

Terminal	Description	Response	Maximum current output
CLR0+, CLR0-, CLR1+, and CLR1-	CLR signals (for clearing the present positions of servo drives which are stored in registers in the servo drives)	10 ms	20 mA
Y2	CLR signal (for the Z-axis)	10 ms	30 mA

Terminal	Description	Response	Maximum current output
FP0+, FP0-, FP1+, FP1-, FP2+, and FP2-	U/D: Counting up P/D: Pulse A/B: A phase	500 kHz	40 mA
RP0+, RP0-, RP1+, RP1-, RP2+, and RP2-	U/D: Counting down P/D: Direction A/B: B phase	500 kHz	40 mA
Y3~Y7	General output terminals	200 kHz	40 mA

Digital input terminals:

■ DVP-20PM series motion controller

Specifications \ Item		24 V DC common terminal		Remark
		Low speed	High speed of 200 kHz[#1]	
Wiring type		A current flows into the terminal S/S (sinking), or a current flows from the terminal S/S.		#1: The input terminals A0+, A0-, A1+, A1-, B0+, B0-, B1+, B1-, PG0+, PG0-, PG1+, and PG1- are high-speed input terminals, and the other input terminals are low-speed input terminals. #2: Users can filter pulses by setting the input terminals X0~X7 to ON after the pulses in 10 ms~60 ms are received.
Input indicator		LED indicator (If the LED indicator corresponding to an input terminal is ON, the input terminal is ON. If the LED indicator corresponding to an input terminal is OFF, the input terminal is OFF.)		
Input voltage		-		
Action level	**Off→On**	20 us		
	On→Off	30 us		
Response time/Noise reduction[#2]		10 ms/0.5 us 0.5us		

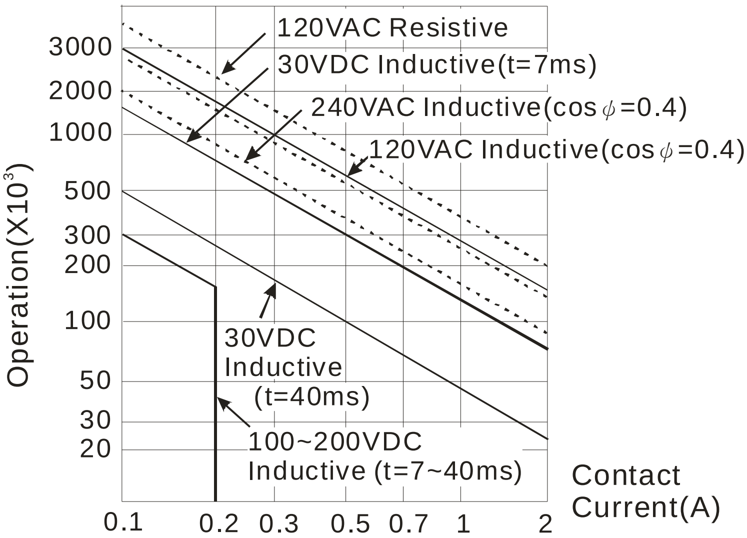
Digital output terminals:

■ DVP-20PM series motion controller

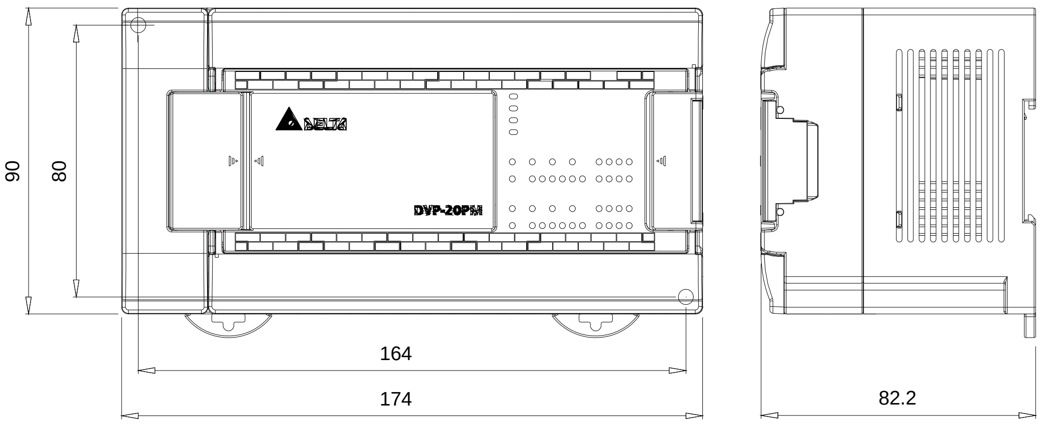
Specifications \ Item		Transistor output terminal connected to a common terminal [#1]		Relay output terminal connected to a common terminal[#1]
		Low speed	High speed	
Maximum frequency of output signals		10 kHz	200 kHz	100 Hz
Output indicator		LED indicator (If the LED indicator corresponding to an output terminal is ON, the output terminal is ON. If the LED indicator corresponding to an output terminal is OFF, the output terminal is OFF.)		
Minimum load		-		2 mA/DC power
Working voltage		5~30 V DC		<250 V AC, 30 V DC
Isolation		Optocoupler		Electromagnetic isolation
Current specifications	**Resistance**	0.5A/output terminal (2A/COM)		2 A/ output terminal (5 A/COM)
	Inductance	9 W (24 V DC)		#2
	Bulb	2 W (24 V DC)		20 W DC/100 W AC
Response time	**Off→On**	20 us	0.2 us	10 ms
	On→Off	30 us		
Overcurrent protection		N/A		

#1: Y0~Y7 on DVP20PM00D are relay output terminals. FP2+ and FP2- on DVP20PM00M are high-speed transistor output terminals, Y2 and T3 on DVP20PM00M are low-speed transistor output terminals, and Y4~Y7 on DVP20PM00M are relay output terminals. Y0 is connected to the common terminal C0, Y1 is connected to the common terminal C1, Y2 and Y3 are connected to the common terminal C2, and Y4~Y7 are connected to the common terminal C3.

#2: Life cycle curve



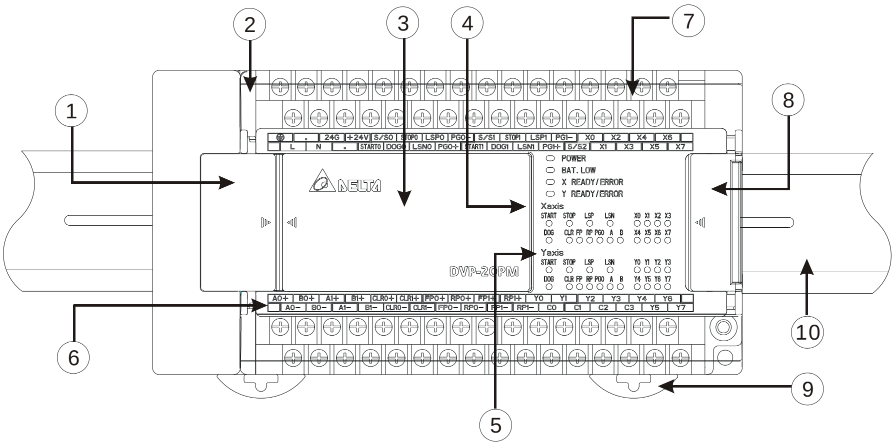
2.1.3 Dimensions

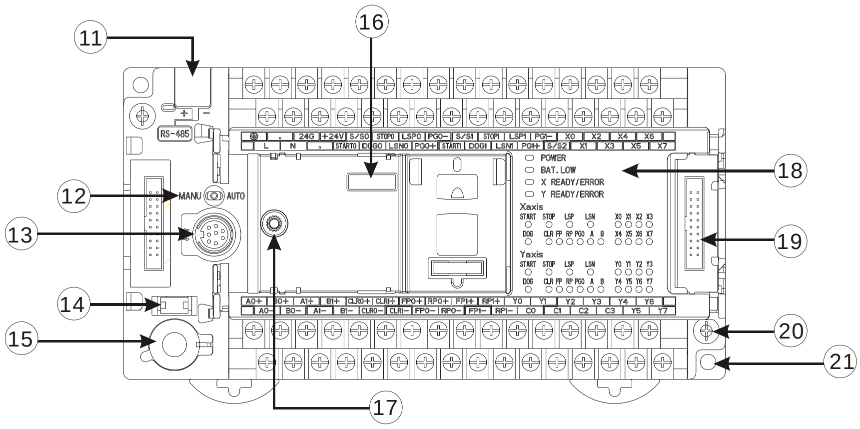


(Unit: mm)

Profile

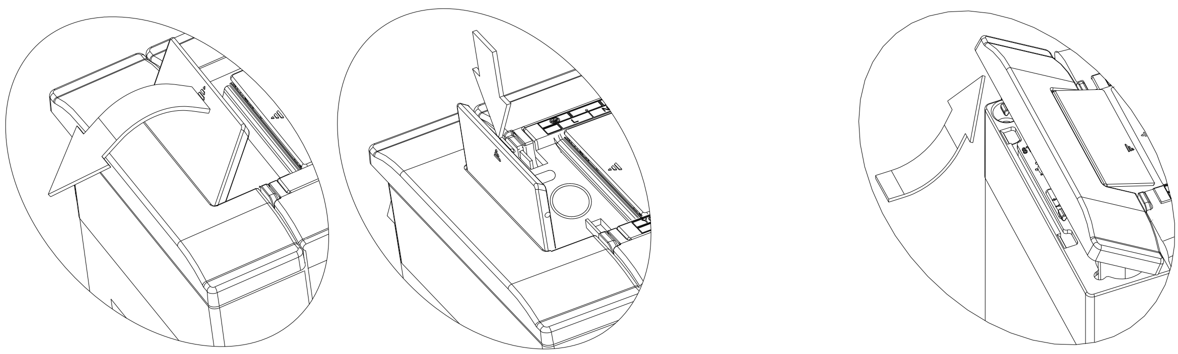
- DVP-20PM series motion controller



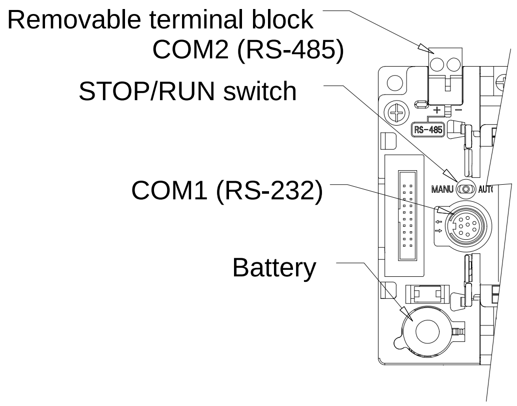


①	Communication port cover
②	Input/Output terminal cover
③	Function card cover
④	Input LED indicators
⑤	Output LED indicators
⑥	Input/Output terminal numbers
⑦	Input/Output terminals
⑧	Connector cover
⑨	DIN rail mounting clip
⑩	DIN rail (35 mm)
⑪	COM2 (RS-485 port)
⑫	STOP/RUN switch
⑬	COM1 (RS-232 port)
⑭	Battery compartment
⑮	Battery
⑯	Connector
⑰	Mounting hole
⑱	POWER LED indicator, BATTERY LED indicator, and ERROR LED indicators
⑲	Connector
⑳	Set screw
㉑	Mounting hole

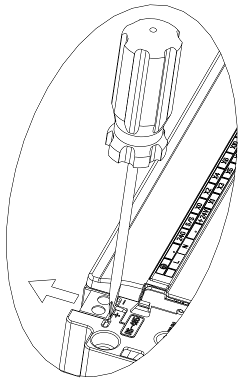
Open the COM1 cover.



Please change the battery in a minute.



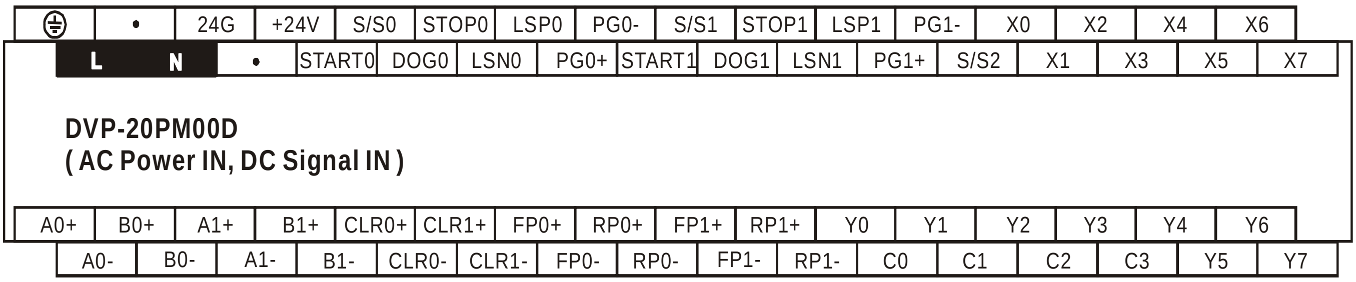
Take out the RS-485 terminals.



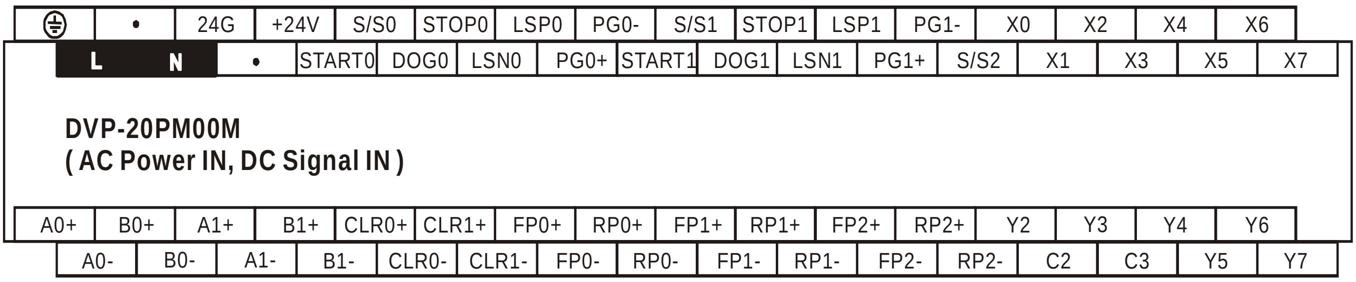
Part	Description
COM2 (RS-485 port)	Master/Slave mode
STOP/RUN switch	Running/Stopping the DVP-20PM series motion controller
COM1 (RS-232 port)	Slave mode (It can be used with COM2 at the same time.)

Arrangement of terminals: Please refer to section 2.1.2 for more information.

- DVP20PM00D



- DVP20PM00M



2.2 Wiring

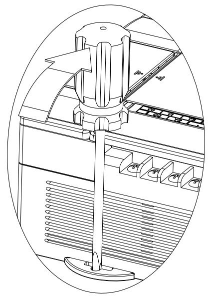
A DVP-20PM series motion controller is an OPEN-TYPE device. It should be installed in a control cabinet free of airborne dust, humidity, electric shock and vibration. To prevent non-maintenance staff from operating a DVP-10PM series motion controller, or to prevent an accident from damaging a DVP-10PM series motion controller, the control cabinet in which a DVP-10PM series motion controller is installed should be equipped with a safeguard. For example, the control cabinet in which a DVP-10PM series motion controller is installed can be unlocked with a special tool or key.

DO NOT connect AC power to any of I/O terminals, otherwise serious damage may occur. Please check all wiring again before a DVP-20PM series motion controller is powered up. Make sure that the ground terminal ⏚ on a DVP-20PM series motion controller is correctly grounded in order to prevent electromagnetic interference.

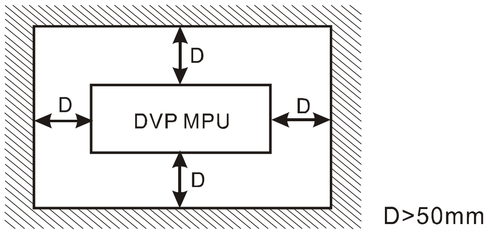
2.2.1 Installation of a DVP-20PM Series Motion Controller in a Control Box

Installing a DIN rail:

The installation is applicable to a 35 millimeter DIN rail. Before users hang a DVP-10PM series motion controller on a DIN rail, they have to insert a slotted screw into the slots on the mounting clips, and pull out the mounting clips. After the users hang the DVP-10PM series motion controller on the DIN rail, they have to push the mounting clips back. If the users want to remove the DVP-10PM series motion controller, they have to insert a slotted screw into the slots on the mounting clips, and pull out the mounting clips. After the mounting clips are pulled out, they will not move back.

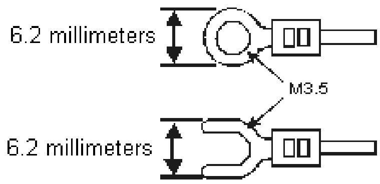


1. Using screws: Please mount a DVP-20PM series motion controller on a DIN rail by means of M4 screws.
2. A DVP-20PM series motion controller has to be installed in a closed control box. In order to ensure that the DVP-20PM series motion controller radiates heat normally, there should be space between the DVP-20PM series motion controller and the control box.



Points for attention:

1. Please use O-type terminals or Y-type terminals. The specifications for terminals are on the right. The torque applied to the terminal screws used should be 9.50 kg-cm (8.25 Ib-in). Please use copper conducting wires. The temperature of the copper conducting wires used should be 60/75°C.
2. Please do not wire NC. Please do not put the cables connected to input terminals and the cables connected to output terminals in the same cable tray.
3. Users have to make sure that there are no tiny metal conductors inside a DVP-10PM series motion controller when they tighten screws and wire terminals. In order to ensure that the DVP-10PM series motion controller radiates heat normally, the users have to remove the sticker on the heat hole.



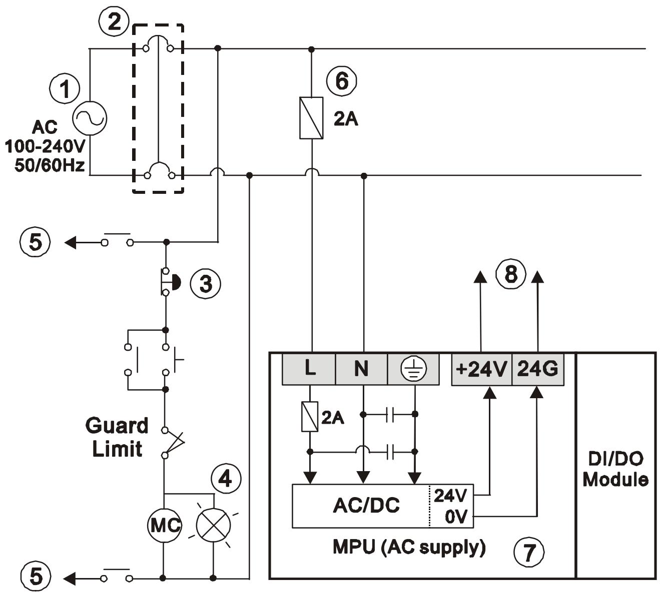
2.2.2 Wiring Power Input

The power input of a DVP-20PM series motion controller is AC input. Users have to pay attention to the following points.

1. The voltage of AC power input is in the range of 100 V AC to 240 V AC. A live wire and a neutral wire are connected to L and N. If 110 V AC power or 220 V AC power is connected to +24V or an input terminal on a DVP-20PM series motion controller, the DVP-20PM series motion controller will be damaged.
2. The AC power input of a DVP-20PM series motion controller, and the AC power input of the I/O module connected to the DVP-20PM series motion controller must be ON or OFF at the same time.
3. The length of the cable connected to the ground terminal on a DVP-20PM series motion controller is at least 1.6 millimeters.
4. If a power cut lasts for less than 10 milliseconds, the DVP-20PM series motion controller used will keep running without being affected. If a power cut lasts for long, or if the voltage of the power input of DVP-20PM series motion controller decreases, the DVP-20PM series motion controller will stop running, and the output terminals will be OFF. When the power input returns to normal, the DVP-20PM series motion controller will resume. (Users have to notice that there are latching auxiliary relays and registers in a DVP-20PM series motion controller when they write a program.)
5. The maximum current which can flows from the power output terminal +24V is 0.5 A. Please do not connect any external power to +24V. The current flows into any input terminal must be in the range of 6 mA to 7 mA. If there are 16 input terminals, 100 mA will be required. As a result, the current that flows from +24 V to an external load can not be greater than 400 mA.

2.2.3 Safety Wiring

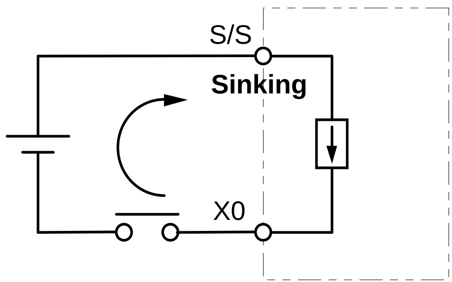
A DVP-20PM series motion controller controls many devices, and the activity of any device affects the activity of other devices. If any device breaks down, the whole automatic control system will go out of control, and dangers will occur. As a result, it is suggested that users should create the protection circuit shown below when they wire power input.



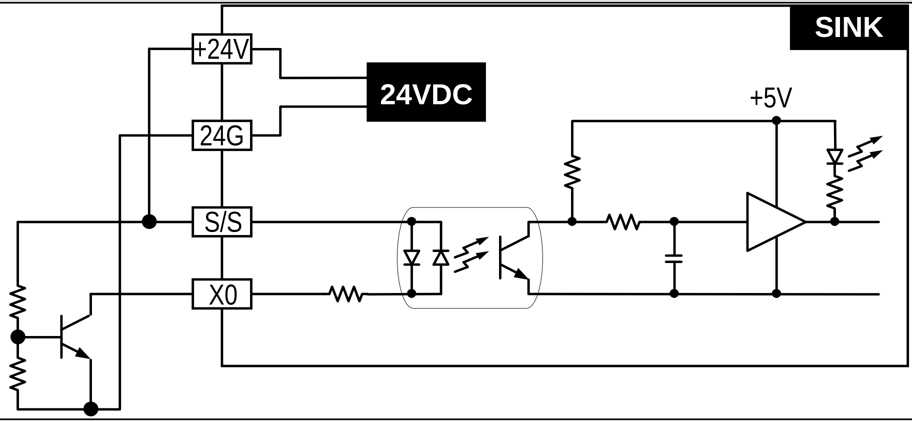
①	Alternating-current power input: 100~240 VAC, 50/60 Hz
②	Circuit breaker
③	Emergency stop: The emergency stop button can be used to cut off power when an emergency occurs.
④	Power indicator
⑤	Load through which a alternating current passes
⑥	3 A fuse
⑦	DVP-20PM series motion controller
⑧	Direct-current power output: 24 V DC, 500 mA

2.2.4 Wiring Input/Output Terminals

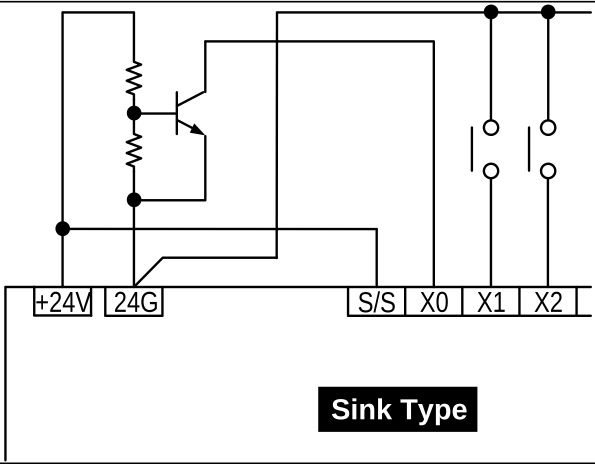
1. The power input of a DVP-20PM series motion controller is DC power input. Sinking and sourcing are current driving capabilities of a circuit. They are defined as follows.

Sinking:
DC power input  Sinking: The current flows into the common terminal S/S.

Equivalent circuit of an input circuit

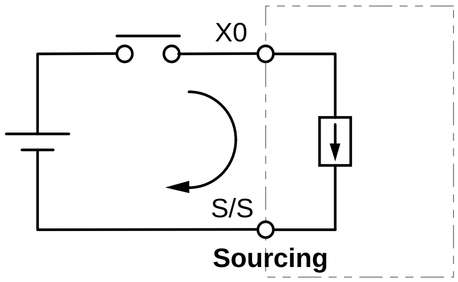


Wiring



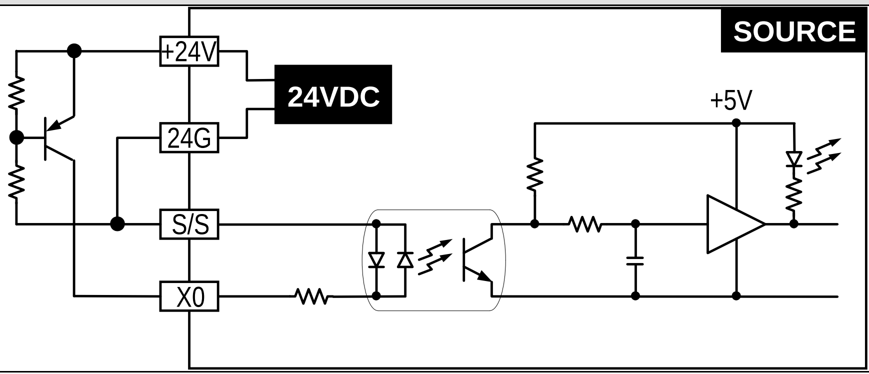
Sourcing:

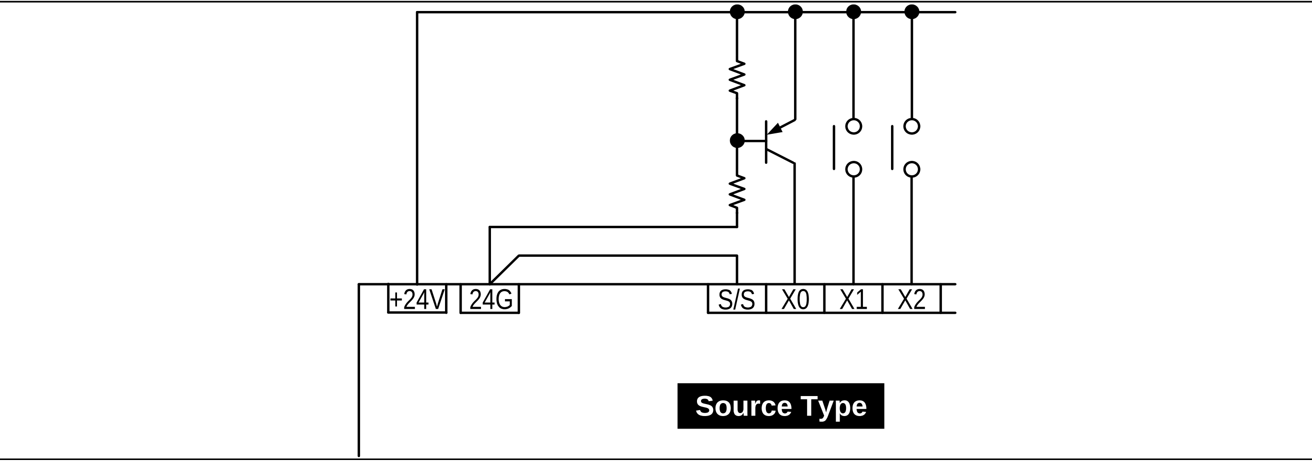
DC power input



Souring: The current flows from the common terminal S/S.

Equivalent circuit of an input circuit

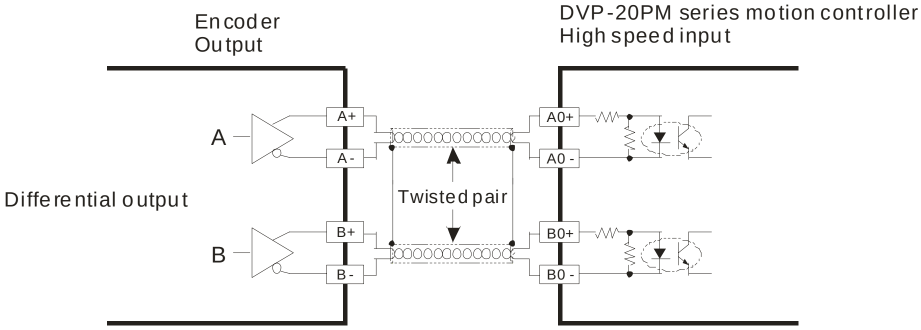


Wiring


2. Wiring differential input terminals

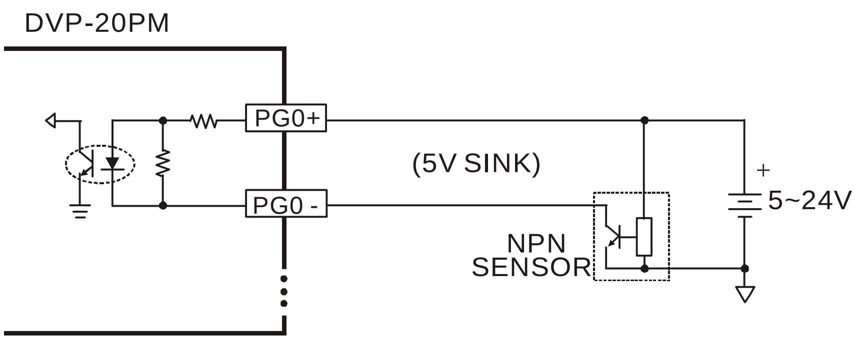
   The direct-current signals ranging in voltage from 5 V to 24 V can pass through the high-speed input terminals A0~A1 and B0~B1 on a DVP-20PM series motion controller. The frequency of input signals can be up to 200 kHz. These high-speed input terminals are connected to a differential (two-wire) line driver.

   Wiring of differential input terminals (used for high speed and noise):

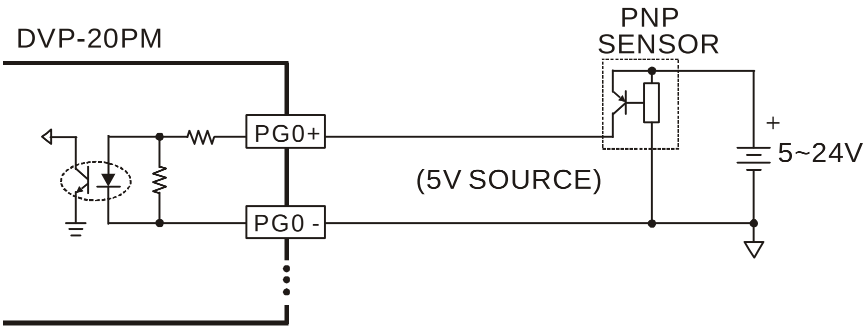


If the frequency of input signals is less than 50 kHz and there is not much noise, these high-speed input terminals can be connected to the direct-current power input whose voltage is in the range of 5 V to 24 V, as shown below.

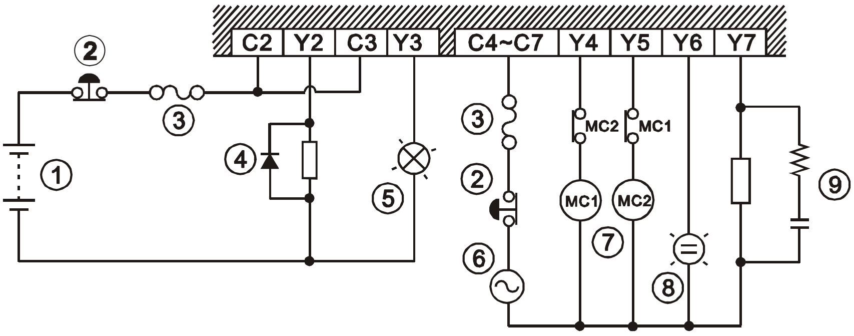
Sinking:



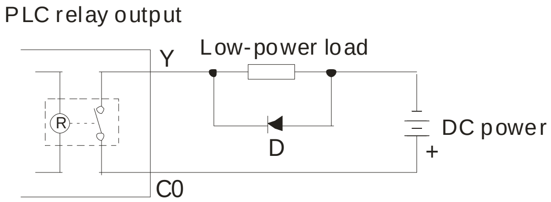
Sourcing:



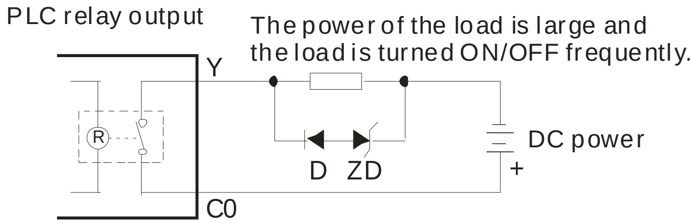
3. Relay output circuit



①	Direct-current power input
②	Emergency stop: An external switch is used.
③	Fuse: To protect the output circuit, a fuse having a breaking capacity in the range of 5 A to 10 A is connected to a common terminal.
④	A diode is used to absorb the surge voltage which occurs when the load connected is OFF. It can lengthen the lifespan of a terminal. 1. A diode is connected to a load through which a direct current passes. It is used when the power of the load connected is small. [Figure 2] 2. A diode and a zener diode are connected to a load through which a direct current passes. They are used when the power of the load is large and the load is turned ON/OFF frequently. [Figure 3]



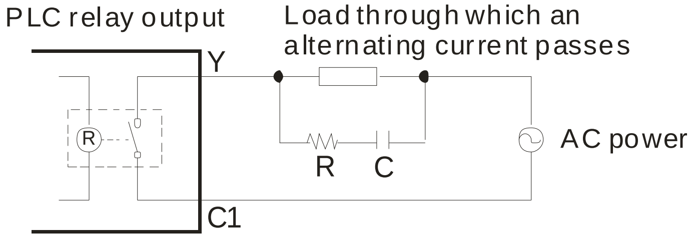
D: 1N4001 diode



D: 1N4001 diode
ZD: 9 V zener diode, 5 W

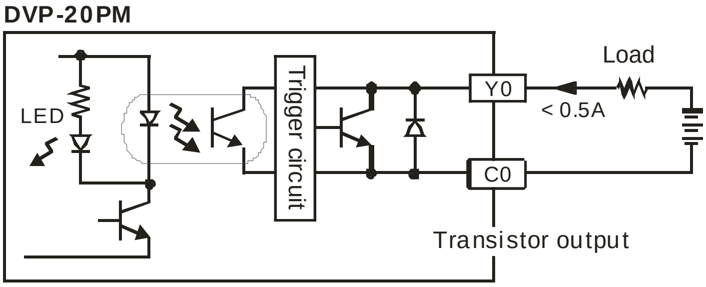
⑤	Incandescent lamp: Resistive load
⑥	Alternating-current power input
⑦	Mutually exclusive output: Y4 controls the clockwise rotation of a motor, and Y5 controls the counterclockwise rotation of a motor. The interlock circuit which is formed, and the program in the DVP-20PM series motion controller ensure that there will be protective measures if an abnormal condition occurs.
⑧	Indicator: Neon lamp

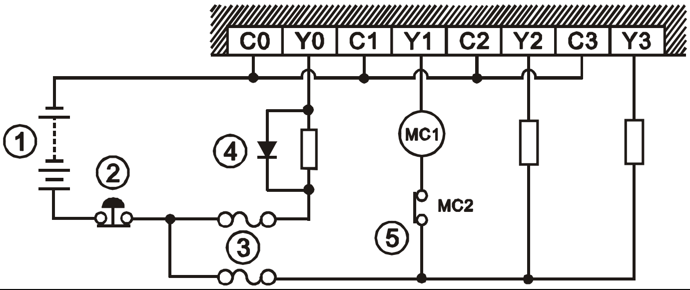
⑨ Surge absorber: It can be used to reduce the noise of a load through which an alternating current passes.



R: 100~120Ω
C: 0.1~0.24uF

4. Transistor output circuit

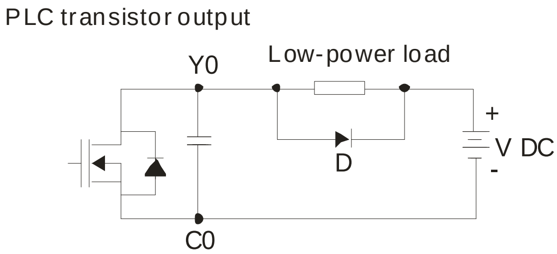




① Direct-current power ② Emergency stop ③ Fuse

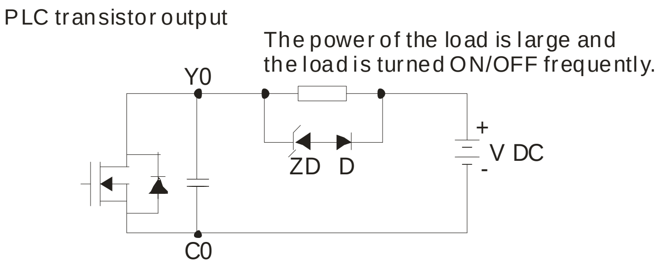
④ Transistor output terminals are open collectors. If Y0/Y1 is a pulse output terminal, the output current passing through an output pull-up resistor must be larger than 0.1 A to ensure that transistor output terminals operate normally.

1. Diode: It is used when the power of the load connected is small.



D: 1N4001 diode

2. Diode and zener diode: They are used when the power of the load connected is large and the load is turned ON/OFF frequently.

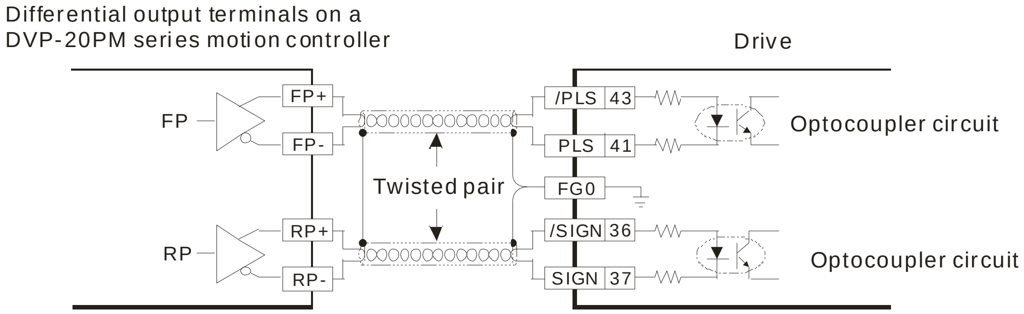


D: 1N4001 diode

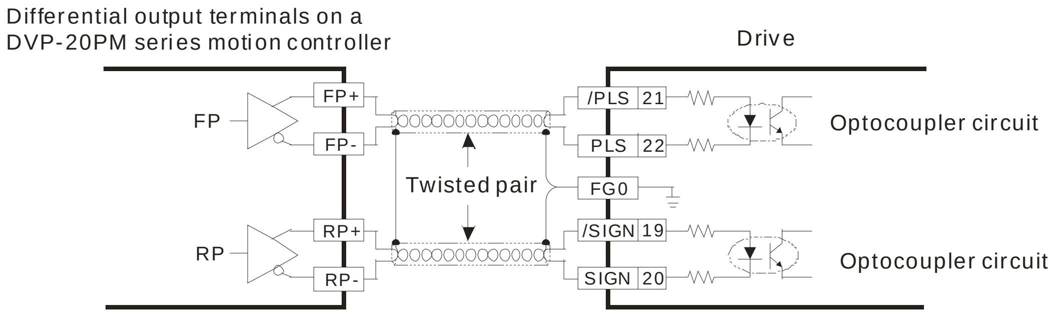
⑤ Mutually exclusive output: Y4 controls the clockwise rotation of a motor, and Y5 controls the counterclockwise rotation of a motor. The interlock circuit which is formed, and the program in the DVP-10PM series motion controller ensure that there will be protective measures if an abnormal condition occurs.

5. Wiring differential output terminals

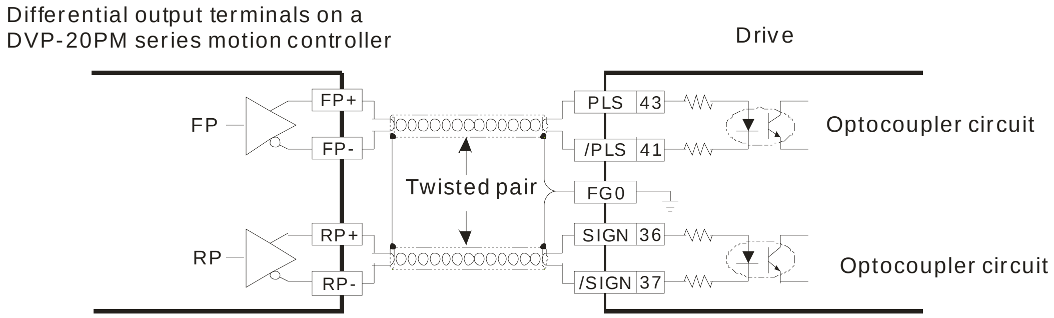
- Wiring differential output terminals on a DVP-20PM series motion controller and an ASDA-A series AC servo drive/ASDA-A+ series AC servo drive/ASDA-A2 series AC servo drive



- Wiring differential output terminals on a DVP-20PM series motion controller and an ASDA-B series AC servo drive

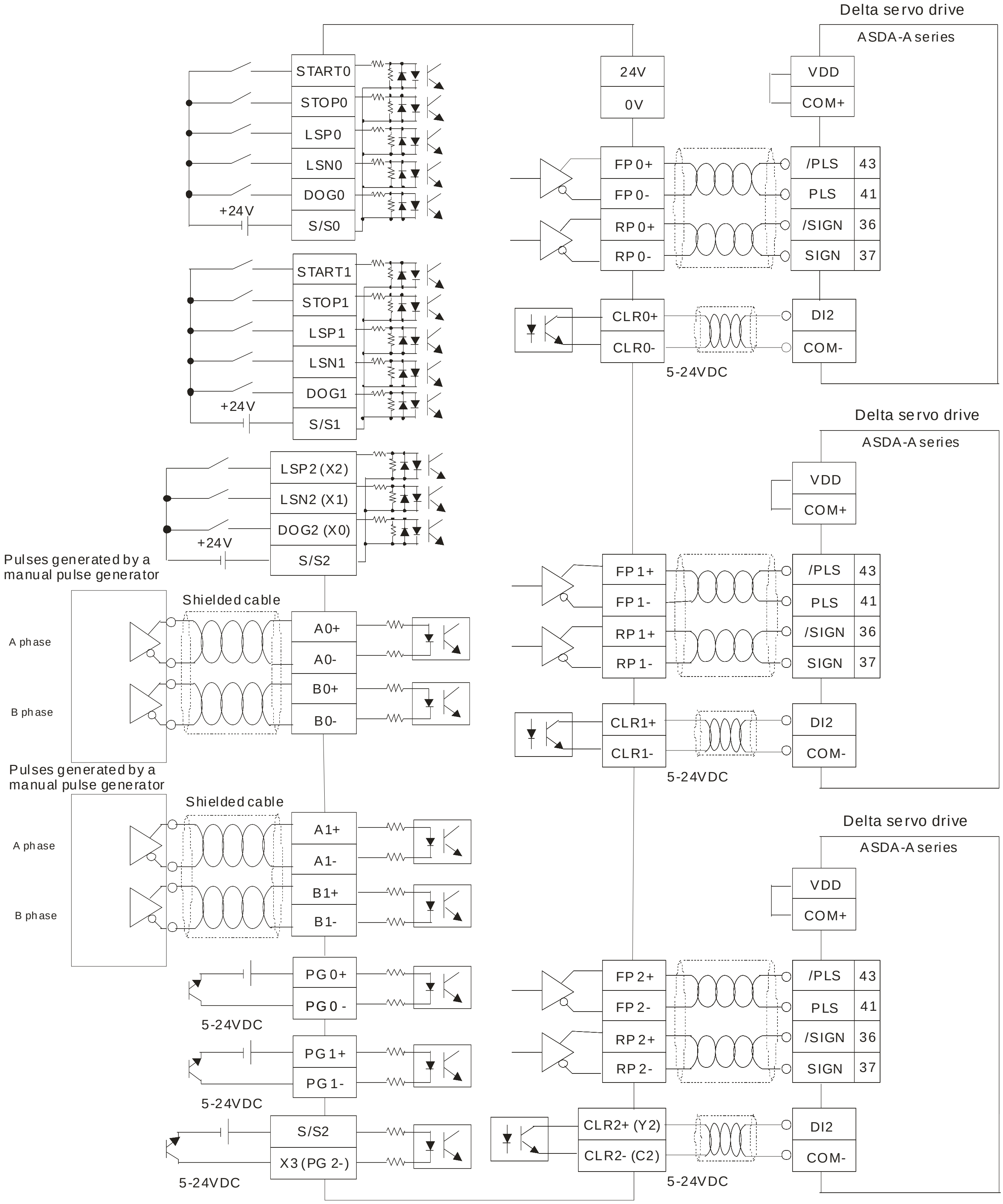


- Wiring differential output terminals on a DVP-20PM series motion controller and an ASDA-AB series AC servo drive

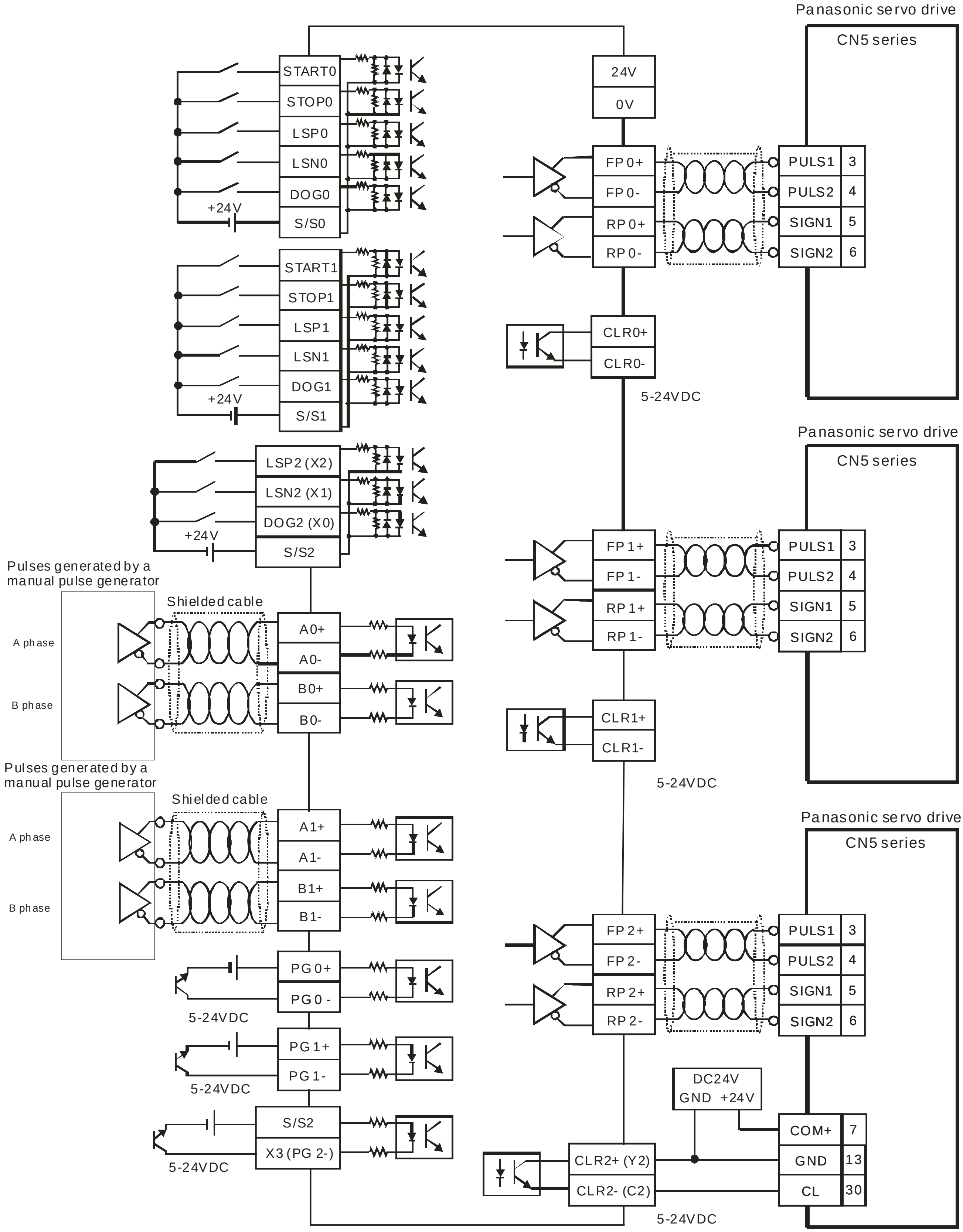


2.2.5 Wiring a DVP-20PM Series Motion Controller and an Inferior Servo Drive

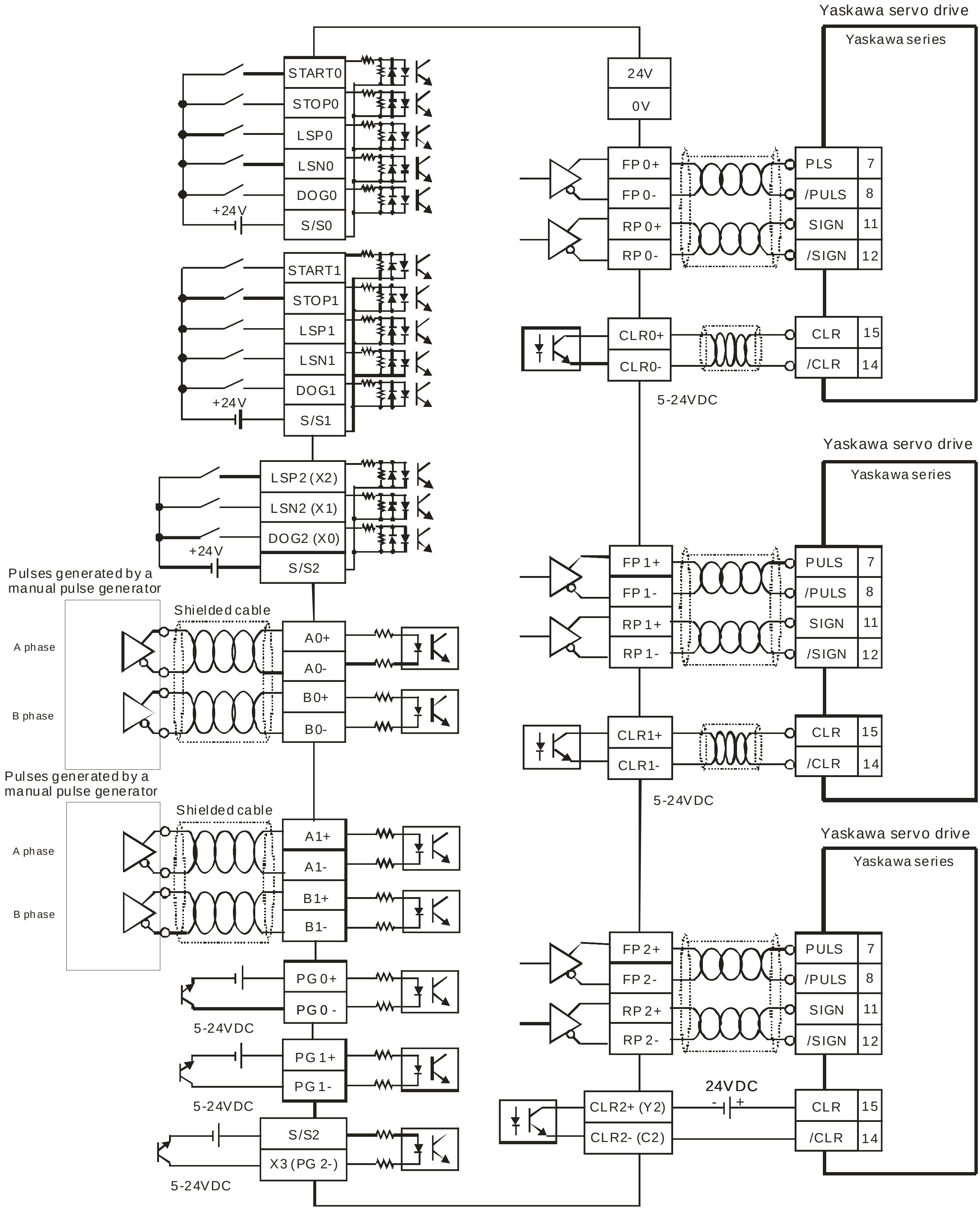
Wiring a DVP-20PM series motion controller and a Delta ASDA-A series AC servo drive:



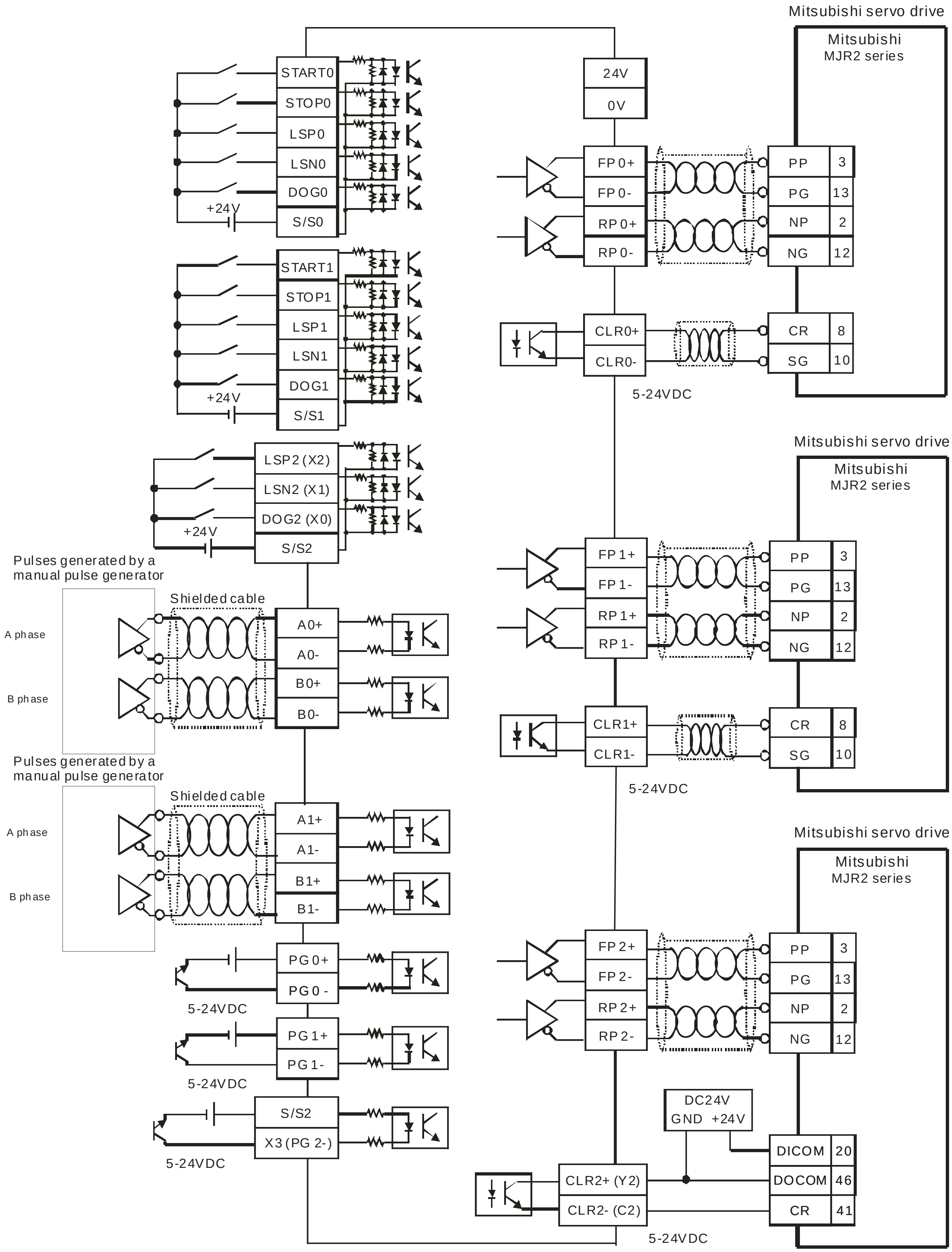
Wiring a DVP-20PM series motion controller and a Panasonic CN5 series servo drive:



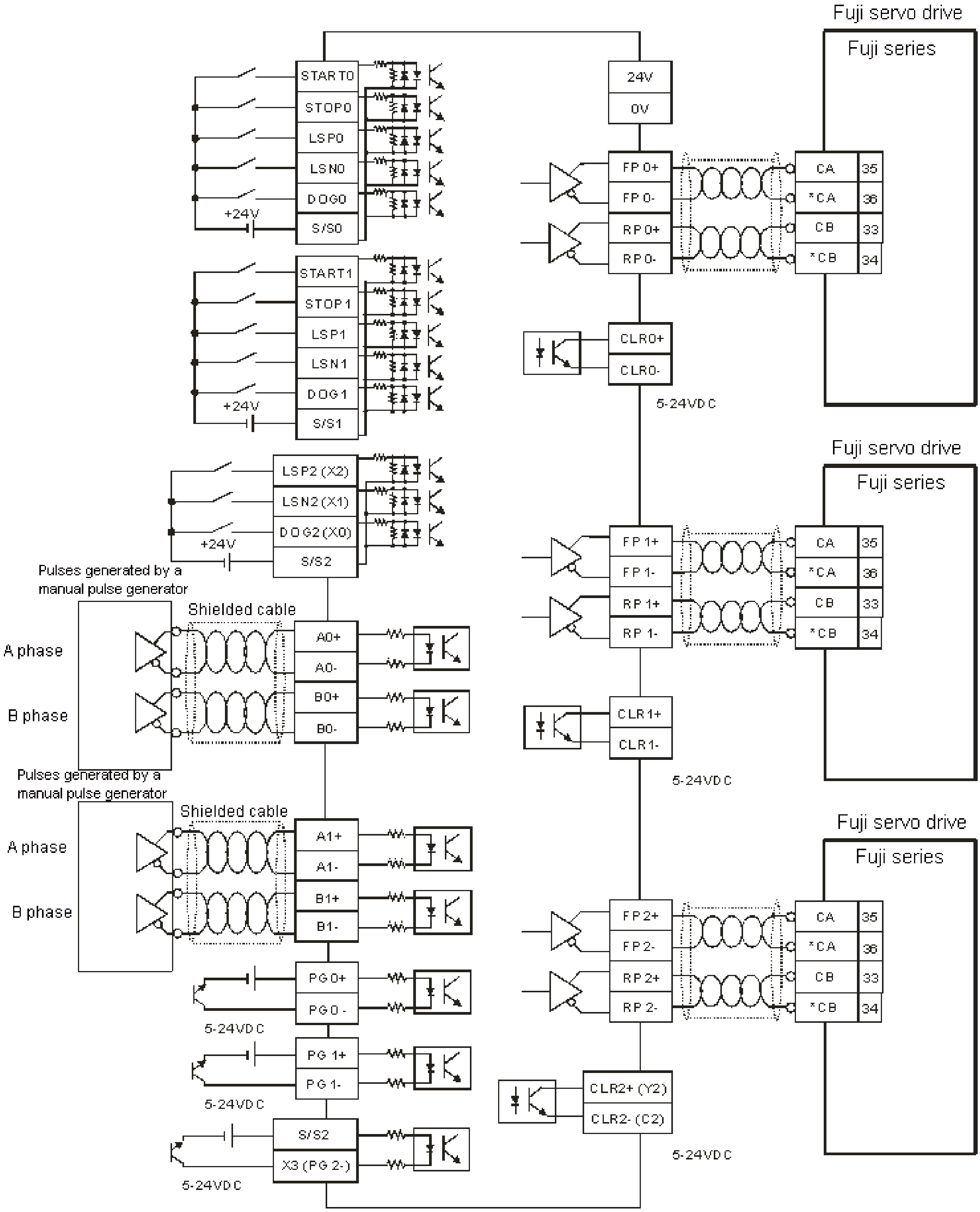
Wiring a DVP-20PM series motion controller and a Yaskawa servo drive:



Wiring a DVP-20PM series motion controller and a Mitsubishi MJR2 series servo drive:



Wiring a DVP-20PM series motion controller and a Fuji servo drive:



2.3 Communication Ports

A DVP-20PM series motion controller is equipped with COM1 (RS-232 port), COM2 (RS-485 port), and a communication card (COM3 (RS-232 or RS-485 communication)).

COM1: It is an RS-232 port. It can function as a slave station. A program is edited through this port. COM1 can be used in a Modbus ASCII mode or an RTU mode.

COM2: It is an RS-485 port. It can function as a master station or a slave station. It can be used in a Modbus ASCII mode or an RTU mode.

COM3: It is an RS-232/RS-485 port. It can function as a slave station. It can be used in a Modbus ASCII mode.

Communication architecture:

Communication port / Communication parameter	RS-232 port (COM1)	RS-485 port (COM2)	RS-232/RS-485 port (COM3)
Serial transmission rate	110~115,200 bps		110~38,400 bps
Number of data bits	7 bits~8 bits		
Parity bit	Even/Odd parity bit/None		
Number of stop bits	1 data bit~2 data bits		
Register where a communication format is stored	D1036	D1120	D1109
ASCII mode	Slave station	Master station/ Slave station	Slave station
RTU mode	Slave station	Master station/ Slave station	-
Quantity of data read/written (ASCII mode)	100 registers		32 registers
Quantity of data read/written (RTU mode)	100 registers		32 registers

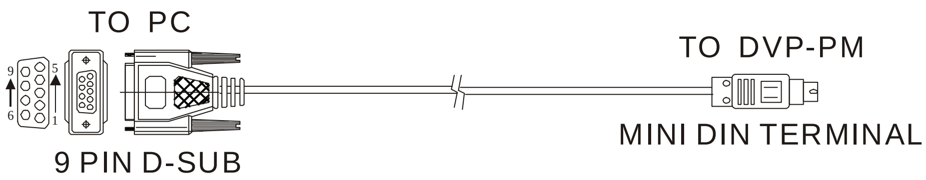
Default communication protocol supported by a communication port

- Modbus ASCII mode
- 7 data bits
- 1 stop bit
- Even parity bit
- Serial transmission rate: 9600 bps

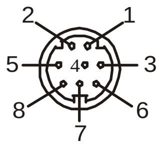
2.3.1 COM1 (RS-232 Port)

1. COM1 is an RS-232 port. Users can upload the program in a DVP-20PM series motion controller through COM1, and download a program to DVP-20PM series motion controller through COM1. The communication protocols that COM1 supports are Modbus ASCII and Modbus RTU, and the transmission rate supported is in the range of 9,600 bps to 115,200 bps.

   The communication cable DVPACAB2A30 is described below.



Communication port on a PC/TP 9-pin D-SUB female connector		↔	COM1 on a DVP-20PM series motion controller 8-pin Mini-DIN connector	
Tx	3	↔	4	Rx
Rx	2	↔	5	Tx
GND	5	↔	8	GND
	7, 8 (connected)		1,2	5V
	1, 4, 6 (connected)			



2. COM1 functions as a slave station. It can be connected to a human-machine interface.

2.3.2 COM2 (RS-485 Port)

1. COM2 is an RS-485 port. It can function as a master station or a slave station. The communication protocols that COM2 supports are Modbus ASCII and Modbus RTU, and the transmission rate supported is in the range of 9,600 bps to 115,200 bps.
2. COM2 can function as a master station or a slave station. If it functions as a master station, it can be connected to a Delta PLC, or an inferior drive such as a Delta servo drive, a Delta AC motor drive, or a temperature controller, and read/write data. If it functions as a slave station, it can be connected to a human-machine interface such as a Delta TP series HMI or DOP series HMI.

2.3.3 COM3 (RS-232 / RS-485 Port)

1. If COM1 (RS-232 port) and COM2 (RS-485 port) can not fulfill a communication requirement, users can use the function card DVP-F232S or DVP-F485S to add a communication interface called COM3 (RS-232/RS485 interface). The functions of DVP-F232S/DVP-F485S is the same as those of COM1, but the transmission rate that DVP-F232S/DVP-F485 supports is 9600/19200/38400 bps in an ASCII mode.
2. COM3 functions as a slave station. It can be connected to a human-machine interface.

3.1 Device Lists

■ Functional specifications

<table>
<tr><th colspan="2">Item</th><th colspan="3">Specifications</th><th>Remark</th></tr>
<tr><td colspan="2">Operation of axes</td><td colspan="3">Two axes operating synchronously control linear/circular interpolation or two axes operate independently (*5)</td><td></td></tr>
<tr><td colspan="2">Storage</td><td colspan="3">The capacity of a built-in storage is 64K steps.</td><td></td></tr>
<tr><td colspan="2">Unit</td><td>Motor unit</td><td>Compound unit</td><td>Mechanical unit</td><td></td></tr>
<tr><td colspan="2">Master mode</td><td colspan="3">Users can read the data in control registers in an I/O module by means of the instruction FROM and write data into control registers in an I/O module by means of the instruction TO. If the data read or written is 32-bit data, two control registers will be used.</td><td></td></tr>
<tr><td colspan="2">Slave mode</td><td colspan="3">If a DVP-20PM series motion controller is used as an I/O module, a master can read the data in CR0~CR199 in the DVP-20PM series motion controller or write data into CR0~CR199 in the DVP-20PM series motion controller. (CR0~CR199 correspond to D1500~D1699.)</td><td></td></tr>
<tr><td colspan="2">Pulse output</td><td colspan="3">There are three types of pulse output modes. These modes adopt differential output.
1. Pulse/Direction
2. Counting up/Counting down
3. A/B-phase output</td><td></td></tr>
<tr><td colspan="2">Maximum speed</td><td colspan="3">Single axis: 500K PPS
Multi-axis interpolation: 500K PPS</td><td></td></tr>
<tr><td rowspan="3">Input signal</td><td>Switch</td><td colspan="3">STOP/RUN switch (Manual/Automatic switch), START switch, STOP switch</td><td></td></tr>
<tr><td>Detector</td><td colspan="3">DOG, LSP, LSN, PG0</td><td></td></tr>
<tr><td>General input terminal</td><td colspan="3">X0~X7
They can be connected to I/O modules. The maximum number of expansion input terminals is 256, including the number of input terminals on a DVP-20PM series motion controller.</td><td></td></tr>
<tr><td rowspan="3">Output signal</td><td>Servo output signal</td><td colspan="3">FP, RP, CLR</td><td></td></tr>
<tr><td>General output terminal</td><td colspan="3">Y0~Y7
They can be connected to I/O modules. The maximum number of expansion output terminals is 256, including the number of output terminals on a DVP-20PM series motion controller.</td><td></td></tr>
<tr><td>Serial communication port</td><td colspan="3">The communication ports which can be used for the reading/writing of a program are as follows.
COM1: RS-232 port (It can function as a slave station.)
COM2: RS-485 port (It can function as a master station or a slave station.)
COM3 (Communication card): RS-232/RS-485 port (It can function as a slave station, and it is optionally required.)</td><td></td></tr>
<tr><td>Special I/O module</td><td>Optional purchase</td><td colspan="3">The EH2 series special right-side modules which are supported are AD, DA, PT, TC, XA, and PU. (Eight special right-side modules can be connected at most, and they do not occupy I/O devices.)</td><td></td></tr>
<tr><td>Special function card</td><td>Optional purchase</td><td colspan="3">The function cards which are supported are 02AD, 02DA, and COM3.</td><td></td></tr>
<tr><td colspan="2">Number of basic instructions</td><td colspan="3">27</td><td></td></tr>
<tr><td colspan="2">Number of applied instructions</td><td colspan="3">130</td><td></td></tr>
<tr><td colspan="2">Number of motion instructions</td><td colspan="3">22</td><td></td></tr>
</table>

Item				Specifications	Remark
M-code				● Ox0~Ox99 (motion subroutine/positioning program): M02 (The execution of a program stops. (END)) ● M00~M01, M03~M101, and M103~M65535: The execution of a program pauses. (WAIT) (Users can use them freely.) ● O100 (main program in a DVP-20PM series motion controller/subtask program): M102 (The execution of a program stops. (END))	
G-code				G0 (rapid positioning), G1 (linear interpolation), G2 (circular interpolation, clockwise), G3 (circular interpolation, counterclockwise), G4 (dwell), G17 (XY plane selection), G18 (ZX plane selection), G19 (YZ plane selection), G90 (absolute programming), and G91 (incremental programming)	
Self-diagnosis				Parameter errors, program errors, external errors and so on are displayed.	
Relay (Bit device)	X	External input relay		X0~X377; octal numbers; 256 external input relays (corresponding to external input terminals)	512 relays in total
	Y	External output relay		Y0~Y377, octal numbers, 256 external output relays (corresponding to external output terminals)	
	M	Auxiliary relay	General	M0~M499; 500 general auxiliary relays (*2)	There are 4,096 auxiliary relays in total. They can be set to ON/OFF in a program.
				M3000~M4095; 1096 general auxiliary relays (*3)	
			Latching	M500~M999; 500 latching auxiliary relays (*3)	
			Special	M1000~M2999; 2000 special auxiliary relays (Some special auxiliary relays are latching auxiliary relays.)	
	T	Timer	10 ms	T0~T255; 256 timers (*2)	There are 256 timers in total. If the present value of the timer specified by the instruction TMR matches the value set, the contact of the timer will be ON.
	C	Counter	16-bit up counter	C0~C99; 100 16-bit up counters (*2)	There are 250 counters in total. If the present value of the counter specified by the instruction CNT (DCNT) matches the value set, the contact of the counter will be ON.
				C100~C199; 100 16-bit up counters (*3)	
			32-bit up/down counter	C208~C219; 12 32-bit up/down counters (*2)	
				C220~C255; 36 32-bit up/down counters (*3)	
			32-bit high-speed counter	C200 and C204; 2 32-bit high-speed counters (*2)	
	S	Stepping relay	General	S0~S499; 500 stepping relays (*2)	There are 1,024 stepping relays in total. They can be set to ON/OFF in a program.
			Latching	S500~S1023; 524 stepping relays (*3)	

Item				Specifications	Remark
Register (Word device)	T	Present value of a timer		T0~T255; 16-bit timers; 256 timers	If the present value of a timer matches the value set, the contact of the timer will be ON.
	C	Present value of a counter		C0~C199; 16-bit counters; 200 counters	If the present value of a counter matches the value set, the contact of the counter will be ON.
				C200~C255; 32-bit counters; 56 counters	
	D	Data register	General	D0~D199; 200 general data registers (*2)	There are 10,000 registers in total. Users can store data in data registers. V/Z registers are index registers.
			Latching	D200~D999; 800 latching data registers (*3)	
				D3000~D9999; 7000 latching data registers (*3)	
			Special	D1000~D2999; 2000 special data registers (Some special data registers are latching data registers.)	
			Index	V0~V7 (16-bit registers); Z0~Z7 (32-bit registers); 16 index registers (*1)	
Pointer	P	Used with CJ, CJN, CALL, or JMP		P0~P255; 256 pointers	It is used with CJ, CJN, CALL, or JMP.
Constant	K	Decimal system		K-32,768~K32,767 (16-bit operation)	
				K-2,147,483,648~K2,147,483,647 (32-bit operation)	
	H	Hexadecimal system		H0000~HFFFF (16-bit operation); H00000000~HFFFFFFFF (32-bit operation)	
	F	Floating-point number		32-bit operation: $\pm 1.1755X10^{-38} \sim \pm 3.4028X10^{+38}$ (The IEEE 754 standard is used.)	

*1: They are non-latching devices, and can not be changed.
*2: They are non-latching devices. Users can change them to latching devices by setting parameters.
*3: They are latching devices. Users can change them to non-latching devices by setting parameters.
*4: They are latching devices, and can not be changed.
*5: DVP20PM00M can create three-axis linear/helical interpolation.

■ Latching and non-latching memory devices

Auxiliary relay (M)	General auxiliary relays			Special auxiliary relays
	M0~M499	M500~M999	M3000~M4095	M1000~M2999
	Non-latching	Latching	Non-latching	(They are in the general auxiliary relay range.)
	Start: D1200 (K500)*1 End: D1201 (K999) *1			Some special auxiliary relays are latching auxiliary relays. They can not be changed.

Timer (T)	10 ms
	T0~T255
	Non-latching
	Start: D1202 (K-1) *2; End: D1203 (K-1) *2

Counter (C)	16-bit up counters		32-bit up/down counters	
	C0~C99	C100~C199	C200, C204, and C208~C219	C220~C255
	Non-latching	Latching	Non-latching	Latching
	Start: D1204 (K100) End: D1205 (K199)		Start: D1206 (K220) End: D1207 (K255)	

Stepping relay (S)	Initial stepping relays	General stepping relay	Latching stepping relay
	S0~S9	S10~S499	S500~S1023
	Non-latching		Latching
	Start: D1208 (K500); End: D1209 (K1023)		

Data register (D)	General data registers	Latching data registers	Special data registers
	D0~D999	D3000~D9999	D1000 ~ D2999 (They are between the general data register range and the latching data register range.)
	Non-latching	Latching	Some special data registers are latching data registers. They can not be changed.
	Start: D1210 (K200) *3 End: D1211 (K9999) *3		

*1: If the value in D1200 is 0, and the value in D1201 is 4095, the DVP-20PM series motion controller used will automatically skip M1000~M2999, and M0~M999 and M3000~M4095 will be changed to latching devices.

*2: K-1 indicates that the timers are non-latching devices.

*3: If the value in D1210 is 0, and the value in D1211 is 9999, the DVP-20PM series motion controller used will automatically skip D1000~D2999, and D0~M999 and D3000~D9999 will be changed to latching devices.

- When power is switched ON/OFF, or when the DVP-20PM series motion controller used switches between a manual mode and an automatic mode, the action of general devices are as shown below.

Memory type	Power OFF=>ON	STOP=>RUN	RUN=>STOP	Clearing all non-latching devices (M1031 is ON.)	Clearing all latching devices (M1032 is ON.)	Factory setting
Non-latching	Cleared	Unchanged	Cleared when M1033 is OFF Unchanged when M1033 is ON	Cleared	Unchanged	0
Latching	Unchanged			Unchanged	Cleared	0

3.2 Values, Constants, and Floating-point Numbers

Constant	**K**	Decimal system	16-bit operation: K-32,768~K32,767 32-bit operation: K-2,147,483,648~K2,147,483,647
	H	Hexadecimal system	16-bit operation: H0~HFFFF 32-bit operation: H0~HFFFFFFFF
Floating-point number	**F**	32-bit number	32-bit operation: $\pm 1.1755X10^{-38}$~$\pm 3.4028X10^{+38}$ (The IEEE 754 standard is used.)

A DVP-20PM series motion controller performs operations on five types of values according to various control purposes. The functions of the five types of values are described below.

1. Binary number (BIN)

   The values on which a DVP-20PM series motion controller performs operations, and the values stored in the DVP-20PM series motion controller are binary numbers. Binary numbers are described below.

   Bit: A bit is the basic unit of information in the binary system. Its state is either 1 or 0.

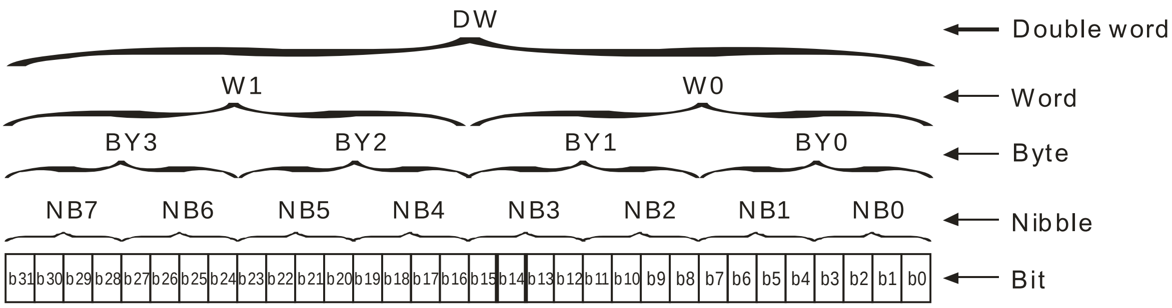
   Nibble: A nibble is composed of four consecutive bits (e.g. b3~b0). Nibbles can be used to represent 0~9 in the decimal system, or 0~F in the hexadecimal system.

   Byte: A byte is composed of two consecutive nibbles (i.e. 8 bits, b7~b0). Bytes can be used to represent 00~FF in the hexadecimal system.

   Word: A word is composed of two consecutive bytes (i.e. 16 bits, b15~b0). Words can be used to represent 0000~FFFF in the hexadecimal system.

   Double word: A double word is composed of two consecutive words (i.e. 32 bits, b31~b0). Double words can be used to represent 00000000~FFFFFFFF in the hexadecimal system.

The relation among bits, nibbles, bytes, words, and double words in the binary system is shown below.



2. Octal number (OCT)

   The external input terminal numbers and the external output terminal numbers on a DVP-20PM series motion controller are octal numbers.

   - External input terminals: X0~X7, X10~X17… (Device numbers)
   - External output terminals: Y0~Y7, Y10~Y17… (Device numbers)

3. Decimal number (DEC)

   - A decimal number can be used as the setting value of a timer, or the setting value of a counter, e.g. TMR T0 K50 (K indicates that the value following it is a constant.).
   - A decimal number can be used as an S/M/T/C/D/V/Z/P device number, e.g. M10 and T30.
   - A decimal number can be used as an operand in an applied instruction, e.g. MOV K123 D0 (K indicates that the value following it is a constant.).

4. Binary-coded decimal number (BCD)

   A decimal value is represented by a nibble or four bits, and therefore sixteen consecutive bits can represent a four-digit decimal value. A binary-coded decimal number is mainly used as the input value of a DIP switch, or the value displayed on a seven-segment display.

5. Hexadecimal Number (HEX)

   - A hexadecimal number can be used as an operand in an applied instruction, e.g. MOV H1A2B D0 (H indicates that the value following it is a constant.).

Constant (K): A decimal number in a DVP-20PM series motion controller is generally preceded by K. For example, K100 represents the decimal number 100.

> Exception:
>
> If K is used with an X/Y/M/S device, a nibble device, a byte device, a word device, or a double word device will be formed.
>
> Example:
>
> K1Y10 represents a device composed of 4 bits, K2Y10 represents a device composed of 8 bits, K3Y10 represents a device composed of 12 bit, and K4Y10 represents a device composed of 16 bits. K1M100 represents a device composed of 4 bits, K2M100 represents a device composed of 8 bits, K3M100 represents a device composed of 12 bit, and K4M100 represents a device composed of 16 bits.

Constant (H): A hexadecimal number in a DVP-20PM series motion controller is generally preceded by H. For example, the hexadecimal number H100 represents the decimal number 256.

Floating-point number (F): A floating-point number in a DVP-20PM series motion controller is generally preceded by F. For example, the floating-point number F3.123 represents 3.123.

Value table:

Binary number (BIN)		Octal number (OCT)	Decimal number (DEC)	Binary-coded decimal number (BCD)		Hexadecimal Number (HEX)
For internal operations in a DVP-20PM series motion controller		X/Y device number	Constant (K) M/S/T/C/D/V/Z/P device number	Input value of a DIP switch, and the value displayed on a seven-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
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 External Input Devices and External Output Devices

- Input devices: X0~X377

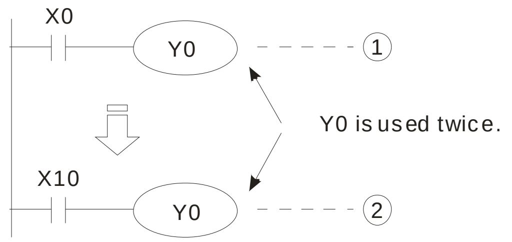
  Input device numbers are octal numbers. A DVP-20PM series motion controller has 256 input devices at most (X0~X7, X10~X17, ……, X370~X377).
- Output devices: Y0~Y377

  Output device numbers are octal numbers. A DVP-20PM series motion controller has 256 output devices at most (YX0~Y7, Y10~Y17, ……, Y370~Y377).
- Functions of input devices:

  After X devices in a DVP-20PM series motion controller are connected to an input device, the input signals sent to the DVP-20PM series motion controller will be read. There is no limitation on the number of times the Form A contact/the Form B contact of an X device can be used in a program. The state of an X device varies with the state of the input device to which the X device is connected.
- Users can turn X devices ON/OFF by means of M1304.

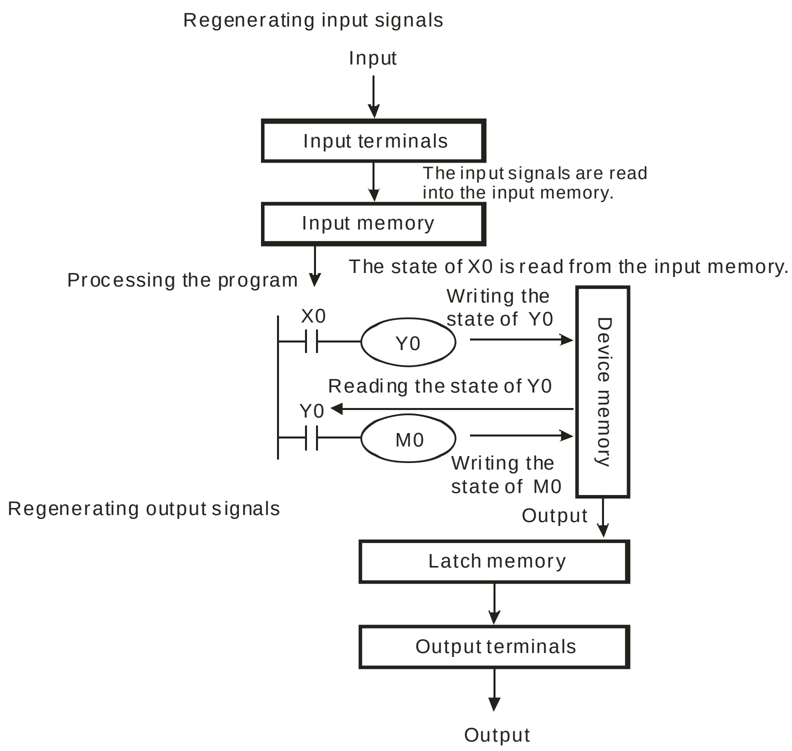
  If M1304 is OFF, X devices can not be turned ON/OFF by means of PMSoft. If M1304 is ON, X devices can be turned ON/OFF by means of PMSoft. However, if users use PMSoft to turn ON/OFF X devices in a DVP-20PM series motion controller when M1304 is ON, the function of updating input signals will be disabled.
- Functions of output devices:

  A Y device sends a signal to drive the load connected to it. There are two types of output devices. They are relays and transistors. There is no limitation on the number of times the Form A contact/the Form B contact of a Y device can be used in a program. However, it is suggested that a Y device should be used once in a program. If a Y device is used more than once in a program, the state of the Y device depends on the Y device used last time.



The state of Y0 depends on circuit ②, that is, the state of X10 determines the state of Y0.

The procedure for processing the program in a DVP-20PM series motion controller is described below.



- Regenerating an input signal:
  1. Before a DVP-20PM series motion controller executes a program, it reads the states of the input signals sent to it into its input memory.
  2. If the states of the input signals change during the execution of the program, the states of input signals stored in the input memory will not change until the DVP-20PM series motion controller reads the states of the input signals sent to it next time.
  3. The time it takes for an input device in the program to receive the state of an external signal is about 10 milliseconds. (The time it takes for a contact in the program to receive the state of an external signal may be affected by the time it takes for the program to be scanned.)
- Processing a program:

  After the DVP-20PM series motion controller reads the states of the input signals stored in the input memory, the execution of the instructions in the program will start from the beginning of the program. After the program is executed, the states of the Y devices used in the program will be stored in the device memory in the DVP-20PM series motion controller.
- Regenerating an output signal:
  1. After M102 is executed, the states of the Y devices stored in the device memory will be sent to the latch memory in the DVP-20PM series motion controller.
  2. The time it takes for a relay to be turned form ON to OFF or turned from OFF to ON is about 10.
  3. The time it takes for a transistor to be turned form ON to OFF or turned from OFF to ON is about 10~20 milliseconds.

3.4 Auxiliary Relays

Auxiliary relay (M): Auxiliary relay numbers are decimal numbers.

Auxiliary relay (M)	General auxiliary relay	M0~M499 (500 general auxiliary relays) Users can change M0~M499 to latching devices by setting parameters.	4,096 auxiliary relays in total
	Latching auxiliary relay	M500~M999 and M3000~M4095 (1,596 latching auxiliary relays) Users can change M0~M499 to non-latching devices by setting parameters.	
	Special auxiliary relay	M1000~M2999 (2,000 special auxiliary relays) Some of them are latching devices.	

Functions of auxiliary relays:

An M device has an output coil and a Form A contact/Form B contact. There is no limitation on the number of times an M device can be used in a program. Users can combine control loops by means of M devices, but can not drive external loads by means of M devices. There are three types of auxiliary relays.

1. General auxiliary relay: If a power cut occurs when a DVP-20PM series motion controller runs, the general auxiliary relays in the DVP-20PM series motion controller will be reset to OFF. When the supply of electricity is restored, the general auxiliary relays are still OFF.
2. Latching auxiliary relay: If a power cut occurs when a DVP-20PM series motion controller runs, the latching auxiliary relays will remain in their last states. When the supply of electricity is restored, the latching auxiliary relays remain unchanged.
3. Special auxiliary relay: Every auxiliary relay has its own specific function. Please do not use the auxiliary relays which are not defined. Users can refer to section 3.10 for more information about special auxiliary relays and special data registers, and refer to section 3.11 for more information about the functions of special auxiliary relays and functions of special data registers.

3.5 Stepping Relays

Stepping relay (S): Stepping relay numbers are decimal numbers.

Stepping relay (S)	General stepping relay	S0~S499 (490 general stepping relays) Users can change S0~S499 to latching devices by setting parameters.	1,024 stepping relays in total
	Latching stepping relay	S500~S1023 (524 latching stepping relays) Users can change S500~S1023 to non-latching devices by setting parameters.	

Functions of stepping relays:

There are 1024 stepping relays (S0~S1023). An S device has an output coil and a Form A contact/Form B contact. There is no limitation on the number of times an S device can be used in a program. Users can not drive external loads by means of S devices. An S device can be used as a general auxiliary relay.

3.6 Timers

Timer (T): Timer numbers are decimal numbers.

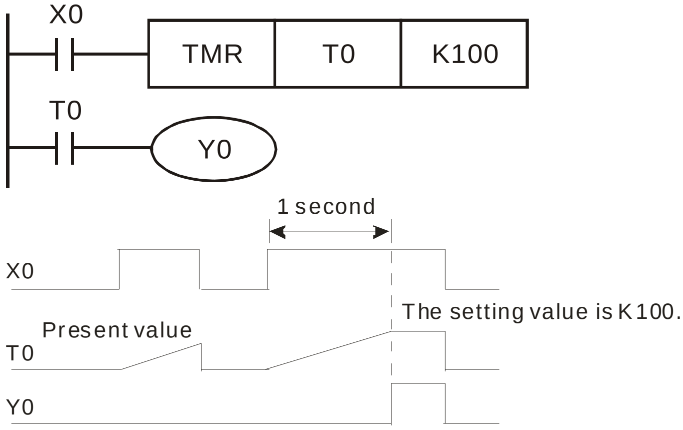
Timer (T)	10 ms General timer	T0~T255 (256 general timers) Users can change T0~T255 to latching devices by setting parameters.

Functions of timers:

10 milliseconds are a unit of measurement for time. A timer counts upwards for measuring time which elapses. If the present value of a timer is equal to the value set, the output coil of the timer will be ON. The value set can be a decimal value preceded by K, or the value in a data register.

Actual time measured by a timer= Unit of measurement for time x Setting value

1. If the instruction TMR is executed, a timer will count for measuring time which elapses once. If the value of a timer matches the value set, the output coil of the timer will be ON.



- If X0 is ON, the timer T0 will count upwards from the present time value every 10 milliseconds. If the present timer value matches the setting value K100, the output coil T0 will be ON.
- If X0 is OFF, or there is a power cut, the present value in T0 will become 0, and the output coil T0 will be OFF.

Setting value: Actual time measured by a timer= Unit of measurement for time x Setting value

1. Constant preceded by K: A setting value can be a constant preceded by K.
2. Value in a data register: A setting value can be the value in a data register.

3.7 Counters

Counter (C): Counter numbers are decimal numbers.

C	Counter	16-bit up counter	C0~C199 (200 16-bit up counters)	248 counters in total	If the present value of the counter specified by the instruction CNT (DCNT) matches the value set, the contact of the counter will be ON.
		32-bit up/down counter	C208~C255 (48 32-bit up/down counters) (Accumulation)		
		32-bit high-speed counter	C200 and C204 (232-bit high-speed counters)	2 counters in total	Input contact of C200: A0±/B0± Input contact of C204: A1±/B1±

Characteristics of counters:

Item	16-bit counter	32-bit counter	
Type	General counter	General counter	High-speed counter
Direction	Counting up	Counting up/down	
Setting value	0~32,767	-2,147,483,648~+2,147,483,647	
Specification of a setting value	Constant preceded by K, or value stored in a data register	Constant preceded by K, or value stored in two consecutive data registers	
Change of the present value	If the present value matches the setting value, the counter will stop counting.	Even if the present value matches the setting value, the counter will keep counting.	
Output contact	If the present value matches the setting value, the output contact will be ON.	Counting up: If the present value matches the setting value, the output contact will be ON. Counting down: If the present value matches the setting value, the output contact will be reset to OFF.	
Resetting of a contact	If the instruction RST is executed, the present value will becomes zero, and the contact will be reset to OFF.		
Actions of contacts	After the scan of a program is complete, the contacts will act.	After the scan of a program is complete, the contacts will act.	If the present value matches the setting value, the contact will be ON.

Functions of counters:

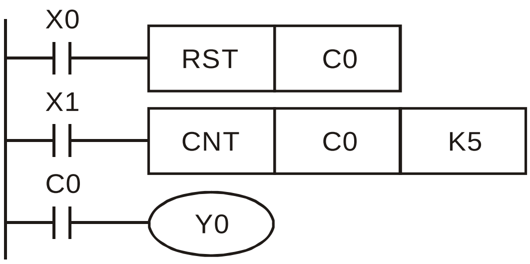
If the input signal of a counter is turned from OFF to ON, and the present value of the counter matches the value set, the output coil of the counter will be ON. A setting value can be a constant preceded by K, or the value stored in a data register.

16-bit counter:

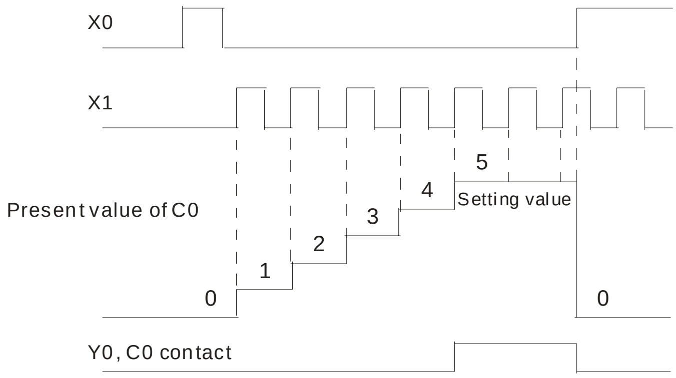
1. The setting value of a 16-bit counter must be in the range of K0 to K32,767. (K0 is equal to K1. If the setting value of a counter is K0 or K1, the output contact of the counter will be ON after the counter counts for the first time.)
2. If a power cut occurs when a general counter in a DVP-20PM series motion controller counts, the present value of the counter will be cleared. If a power cut occurs when a latching counter counts, the present value of the counter and the state of the contact of the counter will be retained, and the latching counter will not continue counting until power is restored.
3. If users move a value greater than the setting value of C0 to C0 by means of the instruction MOV, the contact C0 will be ON, and the present value of the counter will become the setting value next time X1 is turned from OFF to ON.
4. The setting value of a counter can be a constant preceded by K, or the value stored in a data register. (The special data registers D1000~D2999 can not be used.)
5. If the setting value of a counter is a value preceded by K, the setting value can only be a positive value. If the setting value of a counter is the value stored in a data register, the setting value can be a positive value or a negative value. If a counter counts up from the present value 32,767, the next value following 32,767 will be -32,768.

Example:

```
LD    X0
RST   C0
LD    X1
CNT   C0   K5
LD    C0
OUT   Y0
```



1. If X0 is ON, the instruction RST will be executed, the present value of C0 will become zero, and the output contact will be reset to OFF.
2. If X01 is turned from OFF to ON, the present value of the counter will increase by one.
3. If the present value of C0 matches the setting value K5, the contact C0 will be ON (Present value of C0=Setting value=K5). K5 will be retained even if X1 is turned from OFF to ON again.



32-bit up/down counter:

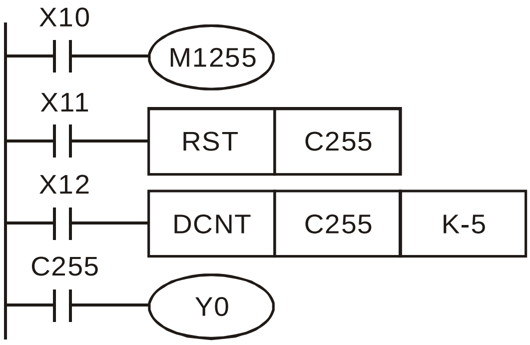
1. The setting value of a 32-bit general up/down counter must be in the range of K-2,147,483,648 to K2,147,483,647. The states of the special auxiliary relays M1208~M1255 determine whether the 32-bit general up/down counters C220~C255 count up or count down. For example, C208 will count up if M1208 is OFF, and C208 will count down if M1208 is ON.
2. The setting value of a 32-bit up/down counter can be a constant preceded by K, or the value stored in two consecutive data registers. (The special data registers D1000~D2999 can not be used.) A setting value can be a positive value, or a negative value.
3. If a power cut occurs when a general counter in a DVP-20PM series motion controller counts, the

present value of the counter will be cleared. If a power cut occurs when a latching counter counts, the present value of the counter and the state of the contact of the counter will be retained, and the latching counter will not continue counting until power is restored.

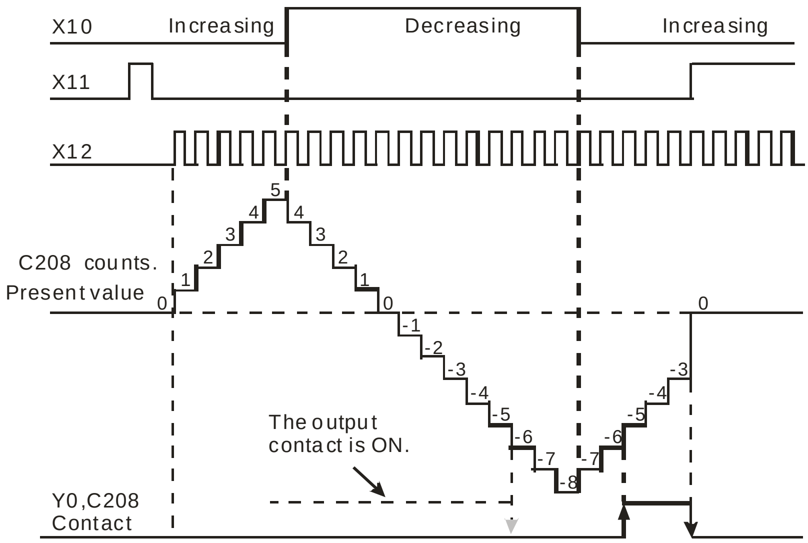
4. If a counter counts up from the present value 2,147,483,647, the next value following 2,147,483,647 will be -2,147,483,648. If a counter counts down from the present value -2,147,483,648, the next value following -2,147,483,648 will be 2,147,483,647.

Example:

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



1. M1255 is driven by X10. The state of M1255 determines whether C255 counts up or counts down.
2. If X11 is turned form OFF to ON, the instruction RST will be executed, the present value of C255 will become 0, and the contact will be OFF.
3. If X12 is turned form OFF to ON, the present value of the counter will increase by one or decrease by one.
4. If the present value of the counter C255 increases from K-6 to K-5, the contact C255 will be turned form OFF to ON. If the present value of the counter C255 decreases from K-5 to K-6, the contact C255 will be turned from ON to OFF.
5. If users move a value greater than the setting value of C255 to C255 by means of the instruction MOV, the contact C255 will be ON, and the present value of the counter will become the setting value next time X11 is turned from OFF to ON.



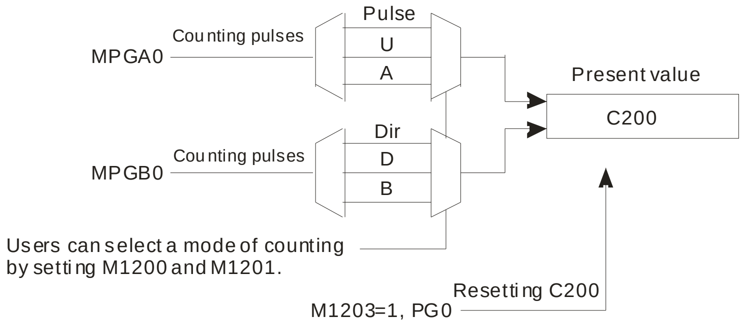
32-bit high-speed counter:

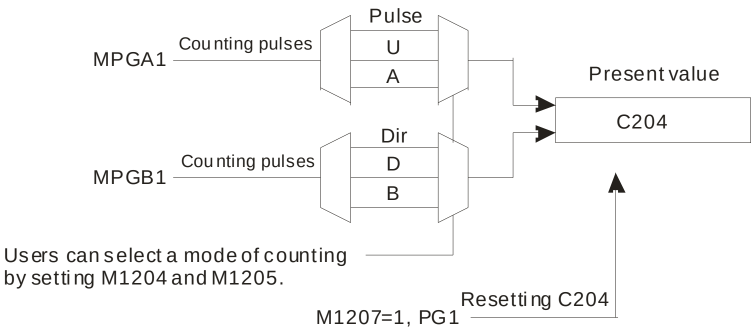
- DVP-20PM series motion controller (C200 and C204)
  1. The setting value of a 32-bit high-speed counter must be in the range of K-2,147,483,648 to K2,147,483,647.
  2. Mode of counting:

Counter	Mode of counting		Resetting a counter	External reset terminal	External input terminal
	Device	Setting value			
C200	K1M1200	0: U/D* 1: P/D* 2: A/B* (One time the frequency of A/B-phase inputs) 3: 4A/B (Four times the frequency of A/B-phase inputs)	M1203	PG0	A0+, A0-, B0+, and B0-
C204	K1M1204		M1207	PG1	A1+, A1-, B1+, and B1-

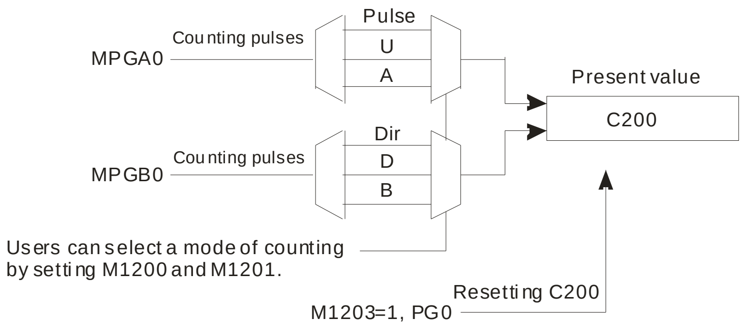
Note: U/D: Counting up/Counting down; P/D: Pulse/Direction; A/B: A phase/B phase

  3. The state of M1908 determines the terminal which controls the input signals for C200/C204.
     - M1908=Off: The input signals for C200 are controlled by A0±/B0±, and the reset signals for C200 are controlled by PG0. The input signals for C204 are controlled by A1±/B1±, and the reset signals for C204 are controlled by PG1.





     - M1908=On: C200 are for the X-axis, the Y-axis, and the Z-axis. The input signals for C200 are controlled by A0±/B0±, and the reset signals for C200 are controlled by PG0.



4. The setting value of a 32-bit high-speed counter can be a constant preceded by K, or the value stored in two consecutive data registers. (The special data registers D1000~D2999 can not be used.) A setting value can be a positive value, or a negative value.
5. If a power cut occurs when a general counter in a DVP-20PM series motion controller counts, the present value of the counter will be cleared. If a power cut occurs when a latching counter counts, the present value of the counter and the state of the contact of the counter will be retained, and the latching counter will not continue counting until power is restored.
6. If a counter counts up from the present value 2,147,483,647, the next value following 2,147,483,647 will be -2,147,483,648. If a counter counts down from the present value -2,147,483,648, the next value following -2,147,483,648 will be 2,147,483,647.

3.8 Registers

Registers are classified according to their characters. There are four types of registers.

1. General register: If the STOP/RUN switch on a DVP-20PM series motion controller is turned from the STOP position to the RUN position, or a DVP-20PM series motion controller is disconnected, the values in the general registers will become 0. If M1033 in a DVP-20PM series motion controller is turned ON, the values in the general registers will be retained after the STOP/RUN switch on the DVP-20PM series motion controller is turned from the RUN position to the STOP position, and will become 0 after the module is disconnected.
2. Latching register: If a module is disconnected, the values in the latching registers will be retained.

   If users want to clear the value in a latching register, they can use the instruction RST or ZRST.
3. Special data register: Every special data register has its definition and purposes. System states, error messages, and states monitored are stored in special data registers. Please refer to section 3.10 and section 3.11 for more information about special auxiliary relays and special data registers.
4. Index register (V)/(Z): V devices are 16-bit registers, and Z devices are 32-bit registers. There are 8 V devices (V0~V7), and 8 Z devices (Z0~Z7) in a DVP-20PM series motion controller.

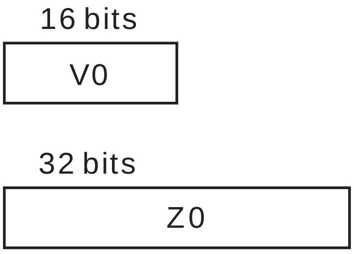
3.8.1 Data Registers

The value in a data register is a 16-bit value. The highest bit in a 16-bit data register represents an algebraic sign. The value stored in a data register must be in the range of -32,768 to +32,767. Two 16-bit data registers can be combined into one 32-bit data register (D+1, D). The highest bit in a 32-bit data register represents an algebraic sign. The value stored in a 32-bit data register must be in the range of -2,147,483,648 to +2,147,483,647.

<table>
<tr><td rowspan="3">Data register (D)</td><td>General data register</td><td>D0~D199 (200 general data registers in total)
Users can change them to latching devices by setting parameters.</td><td rowspan="3">10,000 data registers in total</td></tr>
<tr><td>Latching data register</td><td>D200~D999 and D3000~D9999 (7,800 latching data register in total)
Users can change them to non-latching devices by setting parameters.</td></tr>
<tr><td>Special data register</td><td>D1000~D2999 (2,000 special data registers in total)
Some of them are latching devices.</td></tr>
</table>

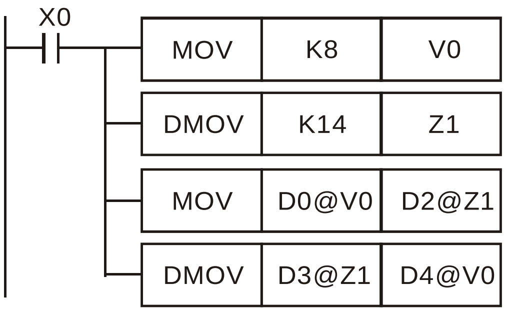
3.8.2 Index Registers

Index register (V)/(Z)	V0~V7	16 index registers in total
	Z0~Z7	



V devices are 16-bit registers. Data can be freely written into a V device, and data can be freely read from a V device. If a V device is used as a general register, it can only be used in a 16-bit instruction.

Z devices are 32-bit registers. If a Z device is used as a general register, it can only be used in a 32-bit instruction.



If X0 is ON, the value in V0 will be 8, and the value in Z1 will be 14, the value in D8 will be moved to D16, and the value in D17 will be moved to D12.

If a V device or a Z device is an index register used to modify an operand, the V device or the Z device can be used in a 16-bit instruction and a 32-bit instruction.

Index registers are like general operands in that they can be used in movement instructions and comparison instructions. They can be used to modify word devices (KnX/KnY/KnM/KnS/T/C/D devices) and bit devices (X/Y/M/S devices).

<u>There are 8 V devices (V0~V7), and 8 Z devices (Z0~Z7) in a DVP-20PM series motion controller.</u>

※Constants and some instructions do not support the use of index registers. Please refer to section 5.4 for more information about using index registers to modify operands.

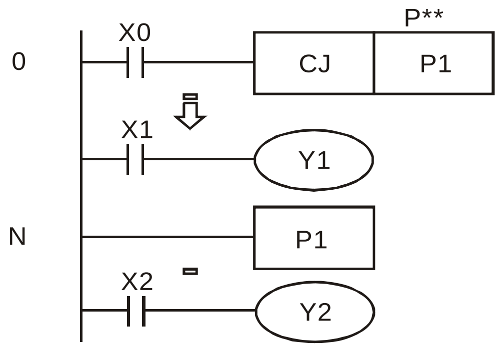
※To prevent error from occurring, if a V device or a Z device is used to modify an operand, the effective address which is formed can not be in the range of D1000 to D2999, and can not be in the range of M1000 to M2999.

3.9 Pointers

Pointer	N	Used with a master control loop	N0~N7 (8 pointers)	Used with a master control loop
	P	Used with CJ, CJN, or JMP	P0~P255 (256 pointers)	Used with CJ, CJN, or JMP

Pointer (P): A pointer is used with API 00 CJ, API 256 CJN, or API 257 JMP. Please refer to chapter 5 for more information about the use of CJ/CJN/JMP.

- Conditional jump (CJ):



- If X0 is ON, the execution of the program will jump from address 0 to address N, and the part of the program between address 0 and address N will not be executed.
- If X0 is OFF, the execution of the program starts from address 0, and the instruction CJ will not be executed.

3.10 Special Auxiliary Relays and Special Data Registers

Special auxiliary relays (special M devices) and special data registers (special D devices) are shown in the tables below. Some device numbers in the tables are marked with *. Users can refer to section 3.11 for more information. If the attribute of a device is “R”, the users can only read data from the device. If the attribute of a device is “R/W”, the users can read data from the device, and write data into the device. In addition, “-” indicates that the state of a special auxiliary relay is unchanged, or the value in a special data register is unchanged. “#” indicates that a special auxiliary relay or a special data register in a DVP-20PM series motion controller is set according to the state of the DVP-20PM series motion controller. The users can read a setting value, and refer to the manual for more information.

Special M device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
M1000*	If the motion controller runs, M1000 will be a normally-open contact (Form A contact). When the motion controller runs, M1000 is ON.	○	○	Off	On	Off	R	No	Off	3-30
M1001*	If the motion controller runs, M1001 will be a normally-closed contact (Form B contact). When the motion controller runs, M1001 is OFF.	○	○	On	Off	On	R	No	On	3-30
M1002*	A positive-going pulse is generated at the time when the motion controller runs. The width of the pulse is equal to the scan cycle.	○	○	Off	On	Off	R	No	Off	3-30
M1003*	A negative-going pulse is generated at the time when the motion controller runs. The width of the pulse is equal to the scan cycle.	○	○	On	Off	On	R	No	On	3-30
M1008	The watchdog timer is ON.	○	○	Off	Off	-	R	No	Off	-
M1009	The low voltage signal has ever occurred.	○	○	Off	-	-	R	No	Off	-
M1011	10 millisecond clock pulse (The pulse is ON for 5 milliseconds, and is OFF for 5 milliseconds.)	○	○	Off	-	-	R	No	Off	-
M1012	100 millisecond clock pulse (The pulse is ON for 50 milliseconds, and OFF for 50 milliseconds.)	○	○	Off	-	-	R	No	Off	-
M1013	1 second clock pulse (The pulse is ON for 0.5 seconds, and OFF for 0.5 seconds.)	○	○	Off	-	-	R	No	Off	-
M1014	1 minute clock pulse (The pulse is ON for 30 seconds, and OFF for 30 seconds.)	○	○	Off	-	-	R	No	Off	-
M1025	Incorrect request for communication (If a PC or an HMI is connected to a DVP-20PM series motion controller, and the DVP-20PM series motion controller receives illegal request for communication during data transmission, M1025 will be set to ON, and an error code will be stored in D1025.)	○	○	Off	Off	-	R	No	Off	-
M1026	Selecting a RAMP mode	○	○	Off	-	-	R/W	No	Off	
M1029	The sending of pulses through CH0 (Y0, Y1) is complete.	○	○	Off	-	-	R	No	Off	
M1031	All the non-latching devices are cleared.	○	○	Off	-	-	R/W	No	Off	-
M1032	All the latching devices are cleared.	○	○	Off	-	-	R/W	No	Off	-
M1033	Data is retained when the DVP-20PM series motion controller does not run.	○	○	Off	-	-	R/W	No	Off	-
M1034	All the outputs are disabled.	○	○	Off	-	-	R/W	No	Off	-
M1035	Using STOP0/START0 as external I/O terminals.	○	○	Off	Off	Off	R/W	No	Off	-
M1036*	Creating a continuous path by means of subroutines	○	○	Off	Off	Off	R/W	No	Off	3-37
M1039*	The scan time for the program is fixed.	○	○	Off	-	-	R/W	No	Off	3-34
M1048	Status of the alarm	○	○	Off	-	-	R	No	Off	-
M1049	Monitoring the alarm	○	○	Off	-	-	R/W	No	Off	-
M1072	The DVP-20PM series motion controller is made to run. (Communication)	○	○	Off	On	Off	R/W	No	Off	-
M1074*	Enabling the Ox motion subroutine specified	○	○	Off	-	-	R/W	No	Off	3-15
M1077	The battery voltage is low, or malfunctions, or there is no battery.	○	○	Off	-	-	R/W	No	Off	-
M1087	The low voltage signal occurs.	○	○	Off	-	-	R/W	No	Off	-

Special M device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
M1120*	The setting of the communication through COM2 (RS-485 port) is retained. After M1120 is set to ON, changing the value in D1120 will be invalid.	○	○	Off	Off	-	R/W	No	Off	3-31
M1121	The transmission of the RS-485 data is ready.	○	○	Off	On	-	R	No	Off	-
M1122	Request for sending the data	○	○	Off	Off	-	R/W	No	Off	-
M1123	The reception of the data is complete.	○	○	Off	Off	-	R/W	No	Off	-
M1124	The reception of the data is ready.	○	○	Off	Off	-	R	No	Off	-
M1125	The reception of the data is reset.	○	○	Off	Off	-	R/W	No	Off	-
M1127	The sending/reception of the data is complete.	○	○	Off	Off	-	R/W	No	Off	-
M1128	The data is being sent/received.	○	○	Off	Off	-	R	No	Off	-
M1129	Reception timeout	○	○	Off	Off	-	R/W	No	Off	-
M1136	The setting of the communication through COM3 (communication card) is retained.	○	○	Off	-	-	R	No	Off	3-31
M1138*	The setting of the communication through COM1 (RS-232 port) is retained. After M1138 is set to ON, changing the value in D1036 will be invalid.	○	○	Off	-	-	R/W	No	Off	3-31
M1139*	Selecting an ASCII mode or an RTU mode when COM1 (RS-232 port) is in a slave mode. (OFF: ASCII mode; ON: RTU mode)	○	○	Off	-	-	R/W	No	Off	3-31
M1140	The data that users receive by means of MODRD/MODWR is incorrect.	○	○	Off	Off	-	R	No	Off	-
M1141	The values of parameters of MODRD/MODWR are incorrect.	○	○	Off	Off	-	R	No	Off	-
M1143*	Selecting an ASCII mode or an RTU mode when COM2 (RS-485 port) is in a slave mode. (OFF: ASCII mode; ON: RTU mode) Selecting an ASCII mode or an RTU mode when COM2 (RS-485 port) is in a master mode. (M1143 is used with MODRD/MODWR.) (OFF: ASCII mode; ON: RTU mode)	○	○	Off	-	-	R/W	No	Off	3-31
M1161	8-bit mode (ON: 8-bit mode; OFF: 16-bit mode)	○	○	Off	-	-	R/W	No	Off	-
M1162	Using decimal integers or binary floating-point values when SCLP is executed. ON: Binary floating-point values OFF: Decimal integers	○	○	Off	-	-	R/W	No	Off	-
M1168	SMOV: Mode of operation	○	○	Off	-	-	R/W	No	Off	-
M1200	C200: Selecting a mode of counting	○	○	Off	-	-	R/W	No	Off	-
M1201	C200: Selecting a mode of counting	○	○	Off	-	-	R/W	No	Off	-
M1203	Resetting C200	○	○	Off	-	-	R/W	No	Off	-
M1204	C204: Selecting a mode of counting	○	○	Off	-	-	R/W	No	Off	-
M1205	C204: Selecting a mode of counting	○	○	Off	-	-	R/W	No	Off	-
M1207	Resetting C204	○	○	Off	-	-	R/W	No	Off	-
M1208	C208: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1209	C209: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1210	C210: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1211	C211: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1212	C212: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1213	C213: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1214	C214: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1215	C215: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-

Special M device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
M1216	C216: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1217	C217: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1218	C218: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1219	C219: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1220	C220: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1221	C221: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1222	C222: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1223	C223: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1224	C224: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1225	C225: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1226	C226: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1227	C227: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1228	C228: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1229	C229: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1230	C230: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1231	C231: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1232	C232: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1233	C233: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1234	C234: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1235	C235: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1236	C236: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1237	C237: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1238	C238: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1239	C239: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1240	C240: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1241	C241: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1242	C242: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1243	C243: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1244	C244: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-
M1245	C245: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R/W	No	Off	-

Special M device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
M1246	C246: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R	No	Off	-
M1247	C247: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R	No	Off	-
M1248	C248: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R	No	Off	-
M1249	C249: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R	No	Off	-
M1250	C250: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R	No	Off	-
M1251	C251: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R	No	Off	-
M1252	C252: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R	No	Off	-
M1253	C253: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R	No	Off	-
M1254	C254: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R	No	Off	-
M1255	C255: Selecting a mode of counting (On: Counting down)	○	○	Off	-	-	R	No	Off	-
M1303	Interchanging high bits with low bits when XCH is executed	○	○	Off	-	-	R/W	No	Off	
M1304*	The input terminals can be set to ON or OFF.	○	○	Off	-	-	R/W	No	Off	3-35
M1744*	Resetting the M-code in the Ox motion subroutine	○	○	Off	Off	-	R/W	No	Off	3-36
M1745	Disabling the X-axis from returning home in the Ox motion subroutine	○	○	Off	-	-	R/W	No	Off	-
M1746	Setting the signal for starting the noncyclic electronic cam motion	○	○	Off			R/W	No	Off	-
M1748	Stopping the electronic cam motion	○	○	Off			R	No	Off	-
M1749	Controlling the maximum frequency of pulses sent by the electronic cam	○	○	Off	-	-	R/W	No	Off	-
M1751	Enabling the writing of the present command position of the X-axis	○	○	Off	-	-	R/W	No	Off	3-36
M1752	The electronic cam does not start at zero. Immediately stopping the noncyclic electronic cam motion when there is a transition in DOG’s signal from low to high or form high to low	○	○	Off	-	-	R/W	No	Off	-
M1755	Switching the source of the master axis of the noncyclic electronic cam	○	○	Off	-	-	R/W	No	Off	-
M1756	The reference terminals for the synchronization output terminals are X0 (CLR0=Off) and X1 (CLR1= Off).	○	○	Off	-	-	R/W	No	Off	-
M1757	The present speed of the electronic cam remains unchanged.	○	○	Off	-	-	R/W	No	Off	-
M1760	Using a radian or a degree in the Ox motion subroutine	○	○	Off	-	-	R/W	No	Off	-
M1761	The X-axis stops at the angle specified.	○	○	Off	-	-	R/W	No	Off	3-36
M1792	The X-axis is ready.	○	○	On	On	On	R	No	On	3-36
M1793*	X-axis motion error flag (It is reset at the time when the X-axis operates.)	○	○	Off	-	-	R/W	No	Off	3-37
M1794*	If an M code in an Ox motion subroutine is executed, M1794 will be ON. (M1974 is reset to OFF at the time when the Ox motion subroutine is executed.)	○	○	Off	-	Off	R	No	Off	3-36
M1795	If M0 in an Ox motion subroutine is executed, M1795 will be ON. (M1795 is reset to OFF at the times when the Ox motion subroutine is executed.)	○	○	Off	-	-	R	No	Off	-
M1796	If M02 in an Ox motion subroutine is executed, M1796 will be ON. (M1796 is reset to OFF at the time when the Ox motion subroutine is executed.)	○	○	Off	On	-	R	No	Off	-

Special M device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
M1808	Zero flag in the Ox motion subroutine	○	○	Off	-	-	R	No	Off	-
M1809	Borrow flag in the Ox motion subroutine	○	○	Off	-	-	R	No	Off	-
M1810	Carry flag in the Ox motion subroutine	○	○	Off	-	-	R	No	Off	-
M1811	An error occurs in a floating-point operation in the Ox motion subroutine	○	○	Off	Off	-	R	No	Off	-
M1812	Initialization of the electronic cam	○	○	Off			R/W	No	Off	
M1813	The electronic cam cycle is complete.	○	○	Off			R/W	No	Off	
M1825	Disabling the Y-axis from returning home	○	○	Off	-	-	R/W	No	Off	-
M1831	Enabling the writing of the present command position of the Y-axis	○	○	Off	-	-	R/W	No	Off	3-36
M1841	The Y-axis stops at the angle specified.	○	○	Off	-	-	R/W	No	Off	3-36
M1872	The Y-axis is ready.	○	○	On	On	On	R	No	On	3-36
M1873*	Y-axis motion error flag (It is reset at the time when the Y-axis operates.)	○	○	Off	-	-	R	No	Off	3-37
M1908	Determining the terminal which controls the input signals for C200/C204	○	○	Off	-	-	R/W	No	Off	
M1909	Virtual axis mode 1	○	○	Off	-	-	R/W	No	Off	
M1910	Virtual axis mode 2	○	○	Off	-	-	R/W	No	Off	
M1920	Using a radian or a degree in O100	○	○	Off	-	-	R/W	No	Off	-
M1952	O100 is ready.	○	○	On	Off	On	R	No	On	-
M1953*	An error occurs in O100.	○	○	Off	-	-	R/W	No	Off	3-39
M1957	State of the STOP/RUN switch	○	○	Off	On	-	R	No	Off	-
M1968	Zero flag in O100.	○	○	Off	-	-	R	No	Off	-
M1969	Borrow flag in O100.	○	○	Off	-	-	R	No	Off	-
M1970	Carry flag in O100.	○	○	Off	-	-	R	No	Off	-
M1971	An error occurs in a floating-point operation in O100.	○	○	Off	-	-	R	No	Off	-
M1985	Disabling the Z-axis from returning home	×	○	Off	-	-	R/W	No	Off	-
M1991	Enabling the writing of the present command position of the Z-axis	○	○	Off	-	-	R/W	No	Off	3-36
M2001	The Z-axis stops at the angle specified.	○	○	Off	-	-	R/W	No	Off	3-36
M2032	The Z-axis is ready.	×	○	On	On	On	R	No	On	3-36
M2033*	Z-axis motion error flag (It is reset at the time when the Z-axis operates.)	×	○	Off	-	-	R/W	No	Off	3-37

Additional remark: 20D=DVP20PM00D; 20M=DVP20PM00M

Special D device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
D1000*	Watchdog timer (Unit: ms)	○	○	200	-	-	R/W	No	200	3-30
D1002	Size of the program	○	○	65535	-	-	R	No	65535	-
D1003	Checksum of the program	○	○	-	-	-	R	Yes	0	-
D1005	Firmware version of the DVP-20PM series motion controller (factory setting)	○	○	#	-	-	R	No	#	-
D1008	Step address at which the watchdog timer is ON	○	○	0	-	-	R	No	0	-
D1010	Present scan time (Unit: 1 millisecond)	○	○	0	-	-	R	No	0	-
D1011	Minimum scan time (Unit: 1 millisecond)	○	○	0	-	-	R	No	0	-
D1012	Maximum scan time (Unit: 1 millisecond)	○	○	0	-	-	R	No	0	-
D1020	Filtering the inputs X0~X7 (Unit: ms)	○	○	10	-	-	R/W	No	10	3-31
D1025	Code for a communication request error	○	○	0	0	-	R	No	0	-
D1036*	Communication protocol of COM1	○	○	H’86	-	-	R/W	No	H’86	3-31
D1038*	Delay which is allowed when an RS-485 port on the DVP-20PM series motion controller functions as a slave station (Setting range: 0~3000; Unit: 10 ms)	○	○	-	-	-	R/W	Yes	0	3-34
D1039*	Fixed scan time (Unit: ms)	○	○	0	-	-	R/W	No	0	-
D1050 ↓ D1055	Modbus communication data is processed. The DVP-20PM series motion controller automatically converts the ASCII data in D1070~D1085 to hexadecimal values.	○	○	0	-	-	R	No	0	-
D1056	Present value of CH0 in the function card 2AD	○	○	0	#	-	R	No	0	-
D1057	Present value of CH1 in the function card 2AD	○	○	0	#	-	R	No	0	-
D1070 ↓ D1085	Modbus communication data is processed. A DVP-20PM series motion controller has an RS-485 communication instruction. After a receptor receives the command sent by an RS-485 communication instruction, it will reply with a message which will be stored in D1070~D1085. Users can view the message by D1070~D1085.	○	○	0	-	-	R	No	0	-
D1089 ↓ D1099	Modbus communication data is processed. A DVP-20PM series motion controller has an RS-485 communication instruction. The command sent by the RS-485 communication instruction is stored in D1089~D1099. Users can check whether the command is correct by viewing the values in D1089~D1099.	○	○	0	-	-	R	No	0	-
D1109	Communication protocol of COM3 (communication card)	○	○	H’86	-	-	R/W	No	H’86	3-31
D1110	Number by which the sum of several values of CH0 in the function card 2AD is divided	○	○	0	#	-	R	No	0	-
D1111	Number by which the sum of several values of CH1 in the function card 2AD is divided	○	○	0	#	-	R	No	0	-
D1116	Present value of CH0 in the function card 2DA	○	○	0	-	-	R/W	No	0	-
D1117	Present value of CH1 in the function card 2DA	○	○	0	-	-	R/W	No	0	-
D1120*	Communication protocol of COM2 (RS-485 port)	○	○	H’86	-	-	R/W	No	H’86	3-31
D1121	Communication address of the DVP-20PM series motion controller	○	○	-	-	-	R/W	Yes	1	-
D1122	Number characters which remain to be sent	○	○	0	0	-	R	No	0	-
D1123	Number of characters which remain to be received	○	○	0	0	-	R	No	0	-
D1124	Start-of-text character (STX)	○	○	H’3A	-	-	R/W	No	H’3A	
D1125	First terminator (END High)	○	○	H’0D	-	-	R/W	No	H’0D	
D1126	Second terminator (END Low)	○	○	H’0A	-	-	R/W	No	H’0A	
D1129	Communication timeout (Unit: ms)	○	○	0	-	-	R/W	No	0	-

Special D device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
D1130	Error code that a slave station sends by means of Modbus when the RS-485 port on the DVP-20PM series motion controller functions as a master station	○	○	0	0	-	R	No	0	-
D1140*	Number of right-side modules (8 right-side modules at most)	○	○	0	-	-	R	No	0	3-35
D1142*	Number of X devices in a digital module	○	○	0	-	-	R	No	0	3-35
D1143*	Number of Y devices in a digital module	○	○	0	-	-	R	No	0	3-35
D1149	ID of a function card (0: No card inserted; 3: COM3; 8: 2AD; 9: 2DA)	○	○	0	-	-	R	No	0	-
D1200*	Starting latching auxiliary relay address	○	○	-	-	-	R/W	Yes	500	3-35
D1201*	Terminal latching auxiliary relay address	○	○	-	-	-	R/W	Yes	999	3-35
D1202*	Starting latching timer address	○	○	-	-	-	R/W	Yes	-1	3-35
D1203*	Terminal latching timer address	○	○	-	-	-	R/W	Yes	-1	3-35
D1204*	Starting latching 16-bit counter address	○	○	-	-	-	R/W	Yes	100	3-35
D1205*	Terminal latching 16-bit counter address	○	○	-	-	-	R/W	Yes	199	3-35
D1206*	Starting latching 32-bit counter address	○	○	-	-	-	R/W	Yes	220	3-35
D1207*	Terminal latching 32-bit counter address	○	○	-	-	-	R/W	Yes	255	3-35
D1208*	Starting latching stepping relay address	○	○	-	-	-	R/W	Yes	500	3-35
D1209*	Terminal latching stepping relay address	○	○	-	-	-	R/W	Yes	1023	3-35
D1210*	Starting latching data register address	○	○	-	-	-	R/W	Yes	200	3-35
D1211*	Terminal latching data register address	○	○	-	-	-	R/W	Yes	9999	3-35
D1313*	Value of the second in the real-time clock (RTC): 00~59	×	×	-	-	-	R/W	Yes	0	-
D1314*	Value of the minute in the real-time clock (RTC): 00~59	×	×	-	-	-	R/W	Yes	0	-
D1315*	Value of the hour in the real-time clock (RTC): 00~23	×	×	-	-	-	R/W	Yes	0	-
D1316*	Value of the day in the real-time clock (RTC): 1~31	×	×	-	-	-	R/W	Yes	1	-
D1317*	Value of the month in the real-time clock (RTC): 01~12	×	×	-	-	-	R/W	Yes	1	-
D1318*	Value of the week in the real-time clock (RTC): 1~7	×	×	-	-	-	R/W	Yes	2/5	-
D1319*	Value of the year in the real-time clock (RTC): 00~99 (A.D.)	×	×	-	-	-	R/W	Yes	8/10	-
D1320*	ID of the first right-side module	○	○	0	-	-	R	No	0	3-35
D1321*	ID of the second right-side module	○	○	0	-	-	R	No	0	3-35
D1322*	ID of the third right-side module	○	○	0	-	-	R	No	0	3-35
D1323*	ID of the fourth right-side module	○	○	0	-	-	R	No	0	3-35
D1324*	ID of the fifth right-side module	○	○	0	-	-	R	No	0	3-35
D1325*	ID of the sixth right-side module	○	○	0	-	-	R	No	0	3-35
D1326*	ID of the seventh right-side module	○	○	0	-	-	R	No	0	3-35
D1327*	ID of the eighth right-side module	○	○	0	-	-	R	No	0	3-35
D1328*	Target position of the third axis controlled by G00 and G01 (low word)	○	×	0	-	-	R/W	No	0	-
D1329*	Target position of the third axis controlled by G00 and G01 (High word)									
D1330*	Maximum speed of the third axis controlled by G00 (low word)	○	×	0	-	-	R/W	No	0	-
	Speed of the interpolation executed by the third axis controlled by G01 (low word)									-
D1331*	Maximum speed of the third axis controlled by G00 (high word)	○	×	0	-	-	R/W	No	0	-
	Speed of the interpolation executed by the third axis controlled by G01 (high word)									-
D1400	Enabling the interrupt	○	○	0	-	-	R/W	No	0	3-35
D1401	Cycle of the time interrupt (Unit: ms)	○	○	0	-	-	R/W	No	0	3-35

Special D device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
D1500	Data block used by FROM/TO It corresponds to CR#0.	○	○	H6260	-	-	R	No	H6260	-
D1501 ↓ D1699	Data block used by FROM/TO They correspond to CR#1~CR#199.	○	○	0	-	-	R/W	No	0	-
D1700	Ox motion subroutine which is executed	○	○	0	-	-	R	No	0	-
D1702	Step address which is executed in the Ox motion subroutine	○	○	0	-	-	R	No	0	-
D1703*	M-code which is executed in the Ox motion subroutine	○	○	0	-	-	R	No	0	3-36
D1704	Dwell duration of the Ox motion subroutine which is set	○	○	0	-	-	R	No	0	-
D1705	Present dwell duration of the Ox motion subroutine	○	○	0	-	-	R	No	0	-
D1706	Number of times the instruction RPT in the Ox motion subroutine is executed	○	○	0	-	-	R	No	0	-
D1707	Number of times the instruction RPT in the Ox motion subroutine has been executed	○	○	0	-	-	R	No	0	-
D1708	Compensation value for the X-axis (low word)	○	○	0	-	-	R	No	0	-
D1709	Compensation value for the X-axis (high word)									
D1710	Compensation value for the center of the arc created by the X-axis (low word)	○	○	0	-	-	R	No	0	-
D1711	Compensation value for the center of the arc created by the X-axis (high word)									
D1712	Compensation value for the radius (low word)	○	○	0	-	-	R	No	0	-
D1713	Compensation value for the radius (high word)									
D1724	Compensation value for the Y-axis (low word)	○	○	0	-	-	R	No	0	-
D1725	Compensation value for the Y-axis (high word)									
D1726	Compensation value for the center of the arc created by the Y-axis (low word)	○	○	0	-	-	R	No	0	-
D1727	Compensation value for the center of the arc created by the Y-axis (high word)									
D1736	Dwell duration of O100 which is set	○	○	0	-	-	R	No	0	-
D1737	Present dwell duration of O100	○	○	0	-	-	R	No	0	-
D1738	Number of times the instruction RPT in O100 is executed	○	○	0	-	-	R	No	0	-
D1739	Number of times the instruction RPT in O100 has been executed	○	○	0	-	-	R	No	0	-
D1796*	Speed to which the speed of the continuous interpolation decreases	○	○	0	-	-	R	No	0	3-37
D1798*	Percentage for the values of the speed parameters of the G-codes	○	○	0	-	-	R	No	0	3-38
D1799*	Polarities of the input terminals	○	○	0	-	-	R/W	No	0	3-38
D1800*	States of the input terminals	○	○	0	-	-	R	No	0	3-38
D1802*	O100 error code	○	○	0	-	-	R/W	No	0	3-39
D1803*	Step address in O100 at which an error occurs	○	○	0	0	-	R/W	No	0	3-39
D1804*	Polarities of the input terminals	×	○	0	-	-	R/W	No	0	3-39
D1805*	States of the input terminals	×	○	0	-	-	R	No	0	3-39
D1806	Filter coefficient for the input terminals	○	○	0	-	-	R/W	No	0	3-39
D1816*	Setting the parameters of the X-axis	○	○	-	-	-	R/W	Yes	0	3-40
D1818	Number of pulses it takes for the motor of the X-axis to rotate once (low word)	○	○	-	-	-	R/W	Yes	2000	-
D1819	Number of pulses it takes for the motor of the X-axis to rotate once (high word)									

Special D device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
D1820	Distance generated after the motor of the X-axis rotate once (low word)	○	○	-	-	-	R/W	Yes	1000	-
D1821	Distance generated after the motor of the X-axis rotate once (High word)									
D1822	Maximum speed (V_{MAX}) at which the X-axis rotates (low word)	○	○	-	-	-	R/W	Yes	500K	-
D1823	Maximum speed (V_{MAX}) at which the X-axis rotates (high word)									
D1824	Start-up speed (V_{BIAS}) at which the X-axis rotates (low word)	○	○	-	-	-	R/W	Yes	0	-
D1825	Start-up speed (V_{BIAS}) at which the X-axis rotates (high word)									
D1826	JOG speed (V_{JOG}) at which the X-axis rotates (low word)	○	○	-	-	-	R/W	Yes	5000	-
D1827	JOG speed (V_{JOG}) at which the X-axis rotates (high word)	○	○	-	-	-	R/W	Yes	5000	-
D1828	Speed (V_{RT}) at which the X-axis returns home (low word)	○	○	-	-	-	R/W	Yes	50K	-
D1829	Speed (V_{RT}) at which the X-axis returns home (high word)									
D1830	Speed (V_{CR}) to which the speed of the X-axis decreases when the X-axis returns home (low word)	○	○	-	-	-	R/W	Yes	1000	-
D1831	Speed (V_{CR}) to which the speed of the X-axis decreases when the X-axis returns home (high word)									
D1832*	Number of PG pulses for the X-axis	○	○	-	-	-	R/W	Yes	0	-
	Setting the number of times the noncyclic electronic cam motion is repeated	○	○	-	-	-	R/W	Yes	0	3-41
D1833*	Supplementary pulses for the X-axis	○	○	-	-	-	R/W	Yes	0	3-41
	Number of remaining pulses sent by the master axis of the electronic cam	○	○	-	-	-	R/W	Yes	0	3-41
D1834*	Home position of the X-axis (low word)	○	○	-	-	-	R/W	Yes	0	3-41
	Number of pulses sent by the master axis of the electronic cam before the electronic cam is started	○	○	-	-	-	R/W	Yes	0	3-41
D1835	Home position of the X-axis (high word)	○	○	-	-	-	R/W	Yes	0	-
D1836	Time (T_{ACC}) it takes for the X-axis to accelerate	○	○	-	-	-	R/W	Yes	500	-
D1837	Time (T_{DEC}) it takes for the X-axis to decelerate	○	○	-	-	-	R/W	Yes	500	-
D1838	Target position of the X-axis (P (I)) (low word)	○	○	○	-	-	R/W	No	0	-
	Upper limit for a synchronization zone (low word)	○	○	○	-	-	R/W	No	0	-
D1839	Target position of the X-axis (P (I)) (high word)	○	○	○	-	-	R/W	No	0	-
	Lower limit for a synchronization zone (high word)	○	○	○	-	-	R/W	No	0	-
D1840	Speed at which the X-axis rotates (V (I)) (low word)	○	○	1000	-	-	R/W	No	1000	-
	Maximum frequency of pulses sent by the electronic cam	○	○	1000	-	-	R/W	No	1000	-
D1841	Speed at which the X-axis rotates (V (I)) (high word)	○	○	1000	-	-	R/W	No	1000	-
	Maximum frequency of pulses sent by the electronic cam	○	○	1000	-	-	R/W	No	1000	-
D1842	Target position of the X-axis (P (II)) (low word)	○	○	0	-	-	R/W	No	0	-
	Upper limit for a synchronization zone (low word)	○	○	0	-	-	R/W	No	0	-
D1843	Target position of the X-axis (P (II)) (high word)	○	○	0	-	-	R/W	No	0	-
	Upper limit for a synchronization zone (high word)	○	○	0	-	-	R/W	No	0	-

Special D device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
D1844	Speed at which the X-axis rotates (V (II)) (low word)	○	○	2000	-	-	R/W	No	2000	-
D1845	Speed at which the X-axis rotates (V (II)) (high word)									
D1846*	Operation command for the X-axis	○	○	0	-	0	R/W	No	0	3-42
	Starting the electronic cam	○	○	0	-	0	R/W	No	0	3-42
D1847*	X-axis's mode of operation	○	○	0	-	-	R/W	No	0	3-43
D1848	Present command position of the X-axis (pulse) (low word)	○	○	0	-	-	R/W	No	0	-
	Position of the slave axis									
D1849	Present command position of the X-axis (pulse) (high word)	○	○	0	-	-	R/W	No	0	-
	Position of the slave axis									
D1850	Present command speed of the X-axis (PPS) (low word)	○	○	0	0	0	R/W	No	0	-
D1851	Present command speed of the X-axis (PPS) (high word)									
D1852	Present command position of the X-axis (unit) (low word)	○	○	0	-	-	R/W	No	0	-
D1853	Present command position of the X-axis (unit) (high word)									
D1854	Present command speed of the X-axis (unit) (low word)	○	○	0	0	0	R/W	No	0	-
D1855	Present command speed of the X-axis (unit) (high word)									
D1856*	State of the X-axis	○	○	0	-	-	R	No	0	3-44
D1857*	X-axis error code	○	○	0	-	-	R	No	0	3-37
D1858	Electronic gear ratio of the X-axis (numerator)	○	○	-	-	-	R/W	Yes	1	-
D1859	Electronic gear ratio of the X-axis (denominator)	○	○	-	-	-	R/W	Yes	1	-
D1860	Frequency of pulses generated by the manual pulse generator for the X-axis (low word)	○	○	0	0	-	R/W	No	0	-
	Frequency of pulses sent by the master axis									
D1861	Frequency of pulses generated by the manual pulse generator for the X-axis (high word)	○	○	0	0	-	R/W	No	0	-
	Frequency of pulses sent by the master axis									
D1862	Number of pulses generated by the manual pulse generator for the X-axis (low word)	○	○	0	-	-	R/W	No	0	-
	Position of the master axis									
D1863	Number of pulses generated by the manual pulse generator for the X-axis (high word)	○	○	0	-	-	R/W	No	0	-
	Position of the master axis									
D1864*	Response speed of the manual pulse generator for the X-axis	○	○	-	-	-	R/W	Yes	5	-
	Pulse type received by the master axis of the electronic cam	○	○	-	-	-	R/W	Yes	5	-
D1865	Mode of stopping Ox0~Ox99 (K1: The execution of Ox0~Ox99 will resume next time Ox0~Ox99 are started. K2: The next instruction will be executed next time Ox0~Ox99 are started. Others: Ox0~Ox99 are executed again.)	○	○	-	-	-	R/W	Yes	0	-
D1866	Electrical zero of the X-axis (low word)	○	○	-	-	-	R/W	Yes	0	-
	Number of pulses for switching the source of the master axis of the noncyclic electronic cam (low word)	○	○	-	-	-	R/W	Yes	0	-
D1867	Electrical zero of the X-axis (high word)	○	○	-	-	-	R/W	Yes	0	-
	Number of pulses for switching the source of the master axis of the noncyclic electronic cam (high word)	○	○	-	-	-	R/W	Yes	0	-

Special D device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
D1868*	Setting an Ox motion subroutine number	○	○	0	0	-	R/W	Yes	0	-
	Specifying an electronic cam chart	○	○	0	0	-	R/W	Yes	0	-
D1869	Step address in the Ox motion subroutine at which an error occurs	○	○	0	-	-	R/W	No	0	-
D1877	Enabling the PID closed-loop control of the X-axis	○	○	-	-	-	R/W	No	0	3-44
D1878	Present value of the command for the PID closed-loop control of the X-axis (low word)	○	○	-	-	-	R	No	0	3-44
D1879	Present value of the command for the PID closed-loop control of the X-axis (high word)	○	○	-	-	-	R	No	0	-
D1880	Proportional gain for the PID closed-loop control of the X-axis (low word)	○	○	-	-	-	R/W	No	0	3-44
D1881	Proportional gain for the PID closed-loop control of the X-axis (high word)	○	○	-	-	-	R/W	No	0	-
D1882	Integral gain for the PID closed-loop control of the X-axis (low word)	○	○	-	-	-	R/W	No	0	3-44
D1883	Integral gain for the PID closed-loop control of the X-axis (high word)	○	○	-	-	-	R/W	No	0	-
D1884	Error for the PID closed-loop control of the X-axis (low word)	○	○	-	-	-	R	No	0	-
D1885	Error for the PID closed-loop control of the X-axis (high word)	○	○	-	-	-	R	No	0	-
D1886	Cumulative error for the PID closed-loop control of the X-axis (low word)	○	○	-	-	-	R	No	0	-
D1887	Cumulative error for the PID closed-loop control of the X-axis (high word)	○	○	-	-	-	R	No	0	-
D1888	Maximum cumulative error for the PID closed-loop control of the X-axis (low word)	○	○	-	-	-	R	No	0	3-44
D1889	Maximum cumulative error for the PID closed-loop control of the X-axis (high word)	○	○	-	-	-	R	No	0	
D1896*	Setting the parameters of the Y-axis	○	○	-	-	-	R/W	Yes	0	3-40
D1898	Number of pulses it takes for the motor of the Y-axis to rotate once (low word)	○	○	-	-	-	R/W	Yes	2000	-
D1899	Number of pulses it takes for the motor of the Y-axis to rotate once (high word)									
D1900	Distance generated after the motor of the Y-axis rotate once (low word)	○	○	-	-	-	R/W	Yes	1000	-
D1901	Distance generated after the motor of the Y-axis rotate once (High word)									
D1902	Maximum speed (V_{MAX}) at which the Y-axis rotates (low word)	○	○	-	-	-	R/W	Yes	500K	-
D1903	Maximum speed (V_{MAX}) at which the Y-axis rotates (high word)									
D1904	Start-up speed (V_{BIAS}) at which the Y-axis rotates (low word)	○	○	-	-	-	R/W	Yes	0	-
D1905	Start-up speed (V_{BIAS}) at which the Y-axis rotates (high word)									
D1906	JOG speed (V_{JOG}) at which the Y-axis rotates (low word)	○	○	-	-	-	R/W	Yes	5000	-
D1907	JOG speed (V_{JOG}) at which the Y-axis rotates (high word)									
D1908	Speed (V_{RT}) at which the Y-axis returns home (low word)	○	○	-	-	-	R/W	Yes	50K	-
D1909	Speed (V_{RT}) at which the Y-axis returns home (high word)									

Special D device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
D1910	Speed (V_{CR}) to which the speed of the Y-axis decreases when the Y-axis returns home (low word)	○	○	-	-	-	R/W	Yes	1000	-
D1911	Speed (V_{CR}) to which the speed of the Y-axis decreases when the Y-axis returns home (high word)									
D1912	Number of PG pulses for the Y-axis	○	○	-	-	-	R/W	Yes	0	-
D1913	Supplementary pulses for the Y-axis	○	○	-	-	-	R/W	Yes	0	-
D1914	Home position of the Y-axis (low word)	○	○	-	-	-	R/W	Yes	0	-
D1915	Home position of the Y-axis (high word)									
D1916	Time (T_{ACC}) it takes for the Y-axis to accelerate	○	○	-	-	-	R/W	Yes	500	-
D1917	Time (T_{DEC}) it takes for the Y-axis to decelerate	○	○	-	-	-	R/W	Yes	500	-
D1918	Target position of the Y-axis (P (I)) (low word)	○	○	0	-	-	R/W	No	0	-
D1919	Target position of the Y-axis (P (I)) (high word)	○	○	0	-	-	R/W	No	0	-
D1920	Speed at which the Y-axis rotates (V (I)) (low word)	○	○	1000	-	-	R/W	No	1000	-
D1921	Speed at which the Y-axis rotates (V (I)) (high word)									
D1922	Target position of the Y-axis (P (II)) (low word)	○	○	0	-	-	R/W	No	0	-
D1923	Target position of the Y-axis (P (II)) (high word)	○	○	0	-	-	R/W	No	0	-
D1924	Speed at which the Y-axis rotates (V (II)) (low word)	○	○	2000	-	-	R/W	No	2000	-
D1925	Speed at which the Y-axis rotates (V (II)) (high word)									
D1926*	Operation command for the Y-axis	○	○	0	-	0	R/W	No	0	3-42
D1927*	Y-axis's mode of operation	○	○	0	-	-	R/W	No	0	3-43
D1928	Present command position of the Y-axis (pulse) (low word)	○	○	0	-	-	R/W	No	0	-
D1929	Present command position of the X-axis (pulse) (high word)									
D1930	Present command speed of the Y-axis (PPS) (low word)	○	○	0	0	0	R/W	No	0	-
D1931	Present command speed of the Y-axis (PPS) (high word)									
D1932	Present command position of the Y-axis (unit) (low word)	○	○	0	-	-	R/W	No	0	-
D1933	Present command position of the Y-axis (unit) (high word)									
D1934	Present command speed of the Y-axis (unit) (low word)	○	○	0	0	0	R/W	No	0	-
D1935	Present command speed of the Y-axis (unit) (high word)									
D1936*	State of the Y-axis	○	○	0	-	-	R	No	0	3-44
D1937*	Y-axis error code	○	○	0	-	-	R	No	0	3-37
D1938	Electronic gear ratio of the Y-axis (numerator)	○	○	-	-	-	R/W	Yes	1	-
D1939	Electronic gear ratio of the Y-axis (denominator)	○	○	-	-	-	R/W	Yes	1	-
D1940	Frequency of pulses generated by the manual pulse generator for the Y-axis (low word)	○	○	0	-	0	R/W	No	0	-
D1941	Frequency of pulses generated by the manual pulse generator for the Y-axis (high word)									
D1942	Number of pulses generated by the manual pulse generator for the Y-axis (low word)	○	○	0	-	-	R/W	No	0	-
D1943	Number of pulses generated by the manual pulse generator for the Y-axis (high word)									
D1944	Response speed of the manual pulse generator for the Y-axis	○	○	-	-	-	R/W	Yes	5	-
D1946	Electrical zero of the Y-axis (low word)	○	○	-	-	-	R/W	Yes	0	-
D1947	Electrical zero of the Y-axis (high word)									

Special D device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
D1957	Enabling the PID closed-loop control of the Y-axis	○	○	-	-	-	R/W	No	0	-
D1958	Present value of the command for the PID closed-loop control of the Y-axis (low word)	○	○	-	-	-	R	No	0	-
D1959	Present value of the command for the PID closed-loop control of the Y-axis (high word)	○	○	-	-	-	R	No	0	-
D1960	Proportional gain for the PID closed-loop control of the Y-axis (low word)	○	○	-	-	-	R/W	No	0	-
D1961	Proportional gain for the PID closed-loop control of the Y-axis (high word)	○	○	-	-	-	R/W	No	0	-
D1962	Integral gain for the PID closed-loop control of the Y-axis (low word)	○	○	-	-	-	R/W	No	0	-
D1963	Integral gain for the PID closed-loop control of the Y-axis (high word)	○	○	-	-	-	R/W	No	0	-
D1964	Error for the PID closed-loop control of the Y-axis (low word)	○	○	-	-	-	R	No	0	-
D1965	Error for the PID closed-loop control of the Y-axis (high word)	○	○	-	-	-	R	No	0	-
D1966	Cumulative error for the PID closed-loop control of the Y-axis (low word)	○	○	-	-	-	R	No	0	-
D1967	Cumulative error for the PID closed-loop control of the Y-axis (high word)	○	○	-	-	-	R	No	0	-
D1968	Maximum cumulative error for the PID closed-loop control of the Y-axis (low word)	○	○	-	-	-	R	No	0	-
D1969	Maximum cumulative error for the PID closed-loop control of the Y-axis (high word)	○	○	-	-	-	R	No	0	-
D1976	Setting the parameters of the Z-axis	×	○	-	-	-	R/W	Yes	0	-
D1978	Number of pulses it takes for the motor of the Z-axis to rotate once (low word)	×	○	-	-	-	R/W	Yes	2000	-
D1979	Number of pulses it takes for the motor of the Z-axis to rotate once (high word)									
D1980	Distance generated after the motor of the Z-axis rotate once (low word)	×	○	-	-	-		Yes	1000	-
D1981	Distance generated after the motor of the Z-axis rotate once (High word)									
D1982	Maximum speed (V_{MAX}) at which the Z-axis rotates (low word)	×	○	-	-	-	R/W	Yes	500K	-
D1983	Maximum speed (V_{MAX}) at which the Z-axis rotates (high word)									
D1984	Start-up speed (V_{BIAS}) at which the Z-axis rotates (low word)	×	○	-	-	-	R/W	Yes	0	-
D1985	Start-up speed (V_{BIAS}) at which the Z-axis rotates (high word)									
D1986	JOG speed (V_{JOG}) at which the Z-axis rotates (low word)	×	○	-	-	-	R/W	Yes	5000	-
D1987	JOG speed (V_{JOG}) at which the Z-axis rotates (high word)									
D1988	Speed (V_{RT}) at which the Z-axis returns home (low word)	×	○	-	-	-	R/W	Yes	50K	-
D1989	Speed (V_{RT}) at which the Z-axis returns home (high word)									
D1990	Speed (V_{CR}) to which the speed of the Z-axis decreases when the Z-axis returns home (low word)	×	○	-	-	-	R/W	Yes	1000	-
D1991	Speed (V_{CR}) to which the speed of the Z-axis decreases when the Z-axis returns home (high word)									
D1992	Number of PG pulses for the Z-axis	×	○	-	-	-	R/W	Yes	0	-
D1993	Supplementary pulses for the Z-axis	×	○	-	-	-	R/W	Yes	0	-

Special D device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
D1994	Home position of the Z-axis (low word)	×	○	-	-	-	R/W	Yes	0	-
D1995	Home position of the Z-axis (high word)									
D1996	Time (T_{ACC}) it takes for the Z-axis to accelerate	×	○	-	-	-	R/W	Yes	500	-
D1997	Time (T_{DEC}) it takes for the Z-axis to decelerate	×	○	-	-	-	R/W	Yes	500	-
D1998	Target position of the Z-axis (P (I)) (low word)	○	○	○	-	-	R/W	No	0	-
D1999	Target position of the Z-axis (P (I)) (high word)	○	○	○	-	-	R/W	No	0	-
D2000	Speed at which the Z-axis rotates (V (I)) (low word)	×	○	1000	-	-	R/W	No	1000	-
D2001	Speed at which the Z-axis rotates (V (I)) (high word)									
D2002	Target position of the Z-axis (P (II)) (low word)	○	○	0	-	-	R/W	No	0	-
D2003	Target position of the Z-axis (P (II)) (high word)	○	○	0	-	-	R/W	No	0	-
D2004	Speed at which the Z-axis rotates (V (II)) (low word)	×	○	2000	-	-	R/W	No	2K	-
D2005	Speed at which the X-axis rotates (V (II)) (high word)									
D2006	Operation command for the Z-axis	×	○	0	-	0	R/W	No	0	3-42
D2007	Z-axis's mode of operation	×	○	0	-	-	R/W	No	0	3-43
D2008	Present command position of the Z-axis (pulse) (low word)	×	○	0	-	-	R/W	No	0	-
D2009	Present command position of the Z-axis (pulse) (high word)									
D2010	Present command speed of the Z-axis (PPS) (low word)	×	○	0	0	0	R/W	No	0	-
D2011	Present command speed of the Z-axis (PPS) (high word)									
D2012	Present command position of the Z-axis (unit) (low word)	×	○	0	-	-	R/W	No	0	-
D2013	Present command position of the Z-axis (unit) (high word)									
D2014	Present command speed of the Z-axis (unit) (low word)	×	○	0	0	0	R/W	No	0	-
D2015	Present command speed of the Z-axis (unit) (high word)									
D2016	State of the Z-axis	×	○	0	-	-	R	No	0	3-44
D2017*	Z-axis error code	×	○	0	-	-	R	No	0	3-37
D2018	Electronic gear ratio of the Z-axis (numerator)	×	○	-	-	-	R/W	Yes	1	-
D2019	Electronic gear ratio of the Z-axis (denominator)	×	○	-	-	-	R/W	Yes	1	-
D2020	Frequency of pulses generated by the manual pulse generator for the Z-axis (low word)	×	○	0	-	0	R/W	No	0	-
D2021	Frequency of pulses generated by the manual pulse generator for the Z-axis (high word)									
D2022	Number of pulses generated by the manual pulse generator for the Z-axis (low word)	×	○	0	-	-	R/W	No	0	-
D2023	Number of pulses generated by the manual pulse generator for the Z-axis (high word)									
D2024	Response speed of the manual pulse generator for the Z-axis	×	○	-	-	-	R/W	Yes	5	-
D2026	Electrical zero of the Z-axis (low word)	×	○	-	-	-	R/W	Yes	0	-
D2027	Electrical zero of the Z-axis (high word)									
D2029	Step address at which an Oz error occurs (The register is not available presently, and is reserved.)	×	○	-	-	-	R/W	Yes	0	-
D2037	Enabling the PID closed-loop control of the Z-axis	×	○	-	-	-	R/W	No	0	

Special D device	Function	20D	20M	Off ⇩ On	STOP ⇩ RUN	RUN ⇩ STOP	Attribute	Latching	Default	Page
D2038	Present value of the command for the PID closed-loop control of the Z-axis (low word)	×	○	-	-	-	R	No	0	
D2039	Present value of the command for the PID closed-loop control of the Z-axis (high word)	×	○	-	-	-	R	No	0	
D2040	Proportional gain for the PID closed-loop control of the Z-axis (low word)	×	○	-	-	-	R/W	No	0	
D2041	Proportional gain for the PID closed-loop control of the Z-axis (high word)	×	○	-	-	-	R/W	No	0	
D2042	Integral gain for the PID closed-loop control of the Z-axis (low word)	×	○	-	-	-	R/W	No	0	
D2043	Integral gain for the PID closed-loop control of the Z-axis (high word)	×	○	-	-	-	R/W	No	0	
D2044	Error for the PID closed-loop control of the Z-axis (low word)	×	○	-	-	-	R	No	0	
D2045	Error for the PID closed-loop control of the Z-axis (high word)	×	○	-	-	-	R	No	0	
D2046	Cumulative error for the PID closed-loop control of the Z-axis (low word)	×	○	-	-	-	R	No	0	
D2047	Cumulative error for the PID closed-loop control of the Z-axis (high word)	×	○	-	-	-	R	No	0	
D2048	Maximum cumulative error for the PID closed-loop control of the Z-axis (low word)	×	○	-	-	-	R	No	0	
D2049	Maximum cumulative error for the PID closed-loop control of the Z-axis (high word)	×	○	-	-	-	R	No	0	

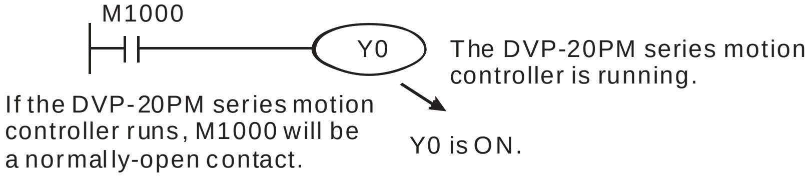
Additional remark: 20D=DVP20PM00D; 20M=DVP20PM00M

3.11 Functions of Special Auxiliary Relays and Special Data Registers

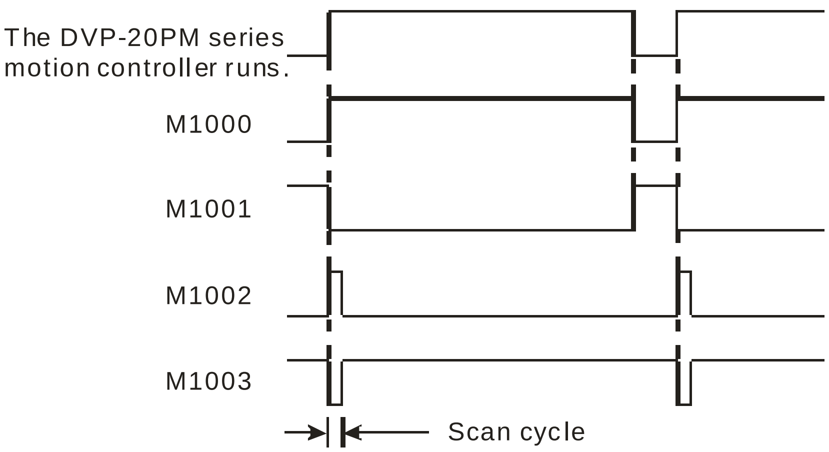
Operation flags

M1000~M1003

1. M1000: If the DVP-20PM series motion controller runs, M1000 will be a normally-open contact (Form A contact). When the DVP-20PM series motion controller runs, M1000 is ON.



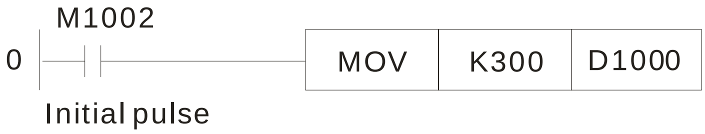
2. M1001: If the DVP-20PM series motion controller runs, M1001 will be a normally-closed contact (Form B contact). When the DVP-20PM series motion controller runs, M1001 is OFF.
3. M1002: A positive-going pulse is generated at the time when the DVP-20PM series motion controller runs. The width of the pulse is equal to the scan cycle. If users want to initialize the DVP-20PM series motion controller, they can use the contact.
4. M1003: A negative-going pulse is generated at the time when the DVP-20PM series motion controller runs. The width of the pulse is equal to the scan cycle.



Watchdog timer

D1000

1. The watchdog timer is used to monitor a scan cycle. If the scan cycle is greater than the watchdog timer value, the ERROR LED indicator on the DVP-20PM series motion controller will be turned ON, and all the output devices will be turned OFF.
2. The watchdog timer is initailly set to 200. If the program is long, or the operation is complex, users can change the watchdog timer value by means of the instruction MOV. In the example below, the watchdog timer value is changed to 300.

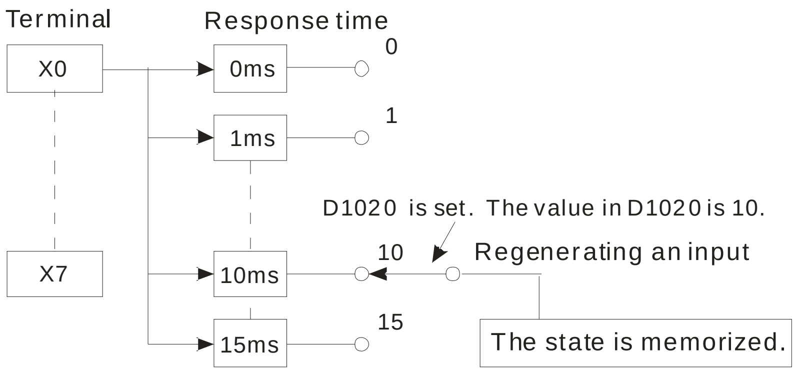


3. The maximum value which can be stored in the watchdog timer is 32,767. However, the larger the watchdog timer value is, the more time it takes to detect any operation error. As a result, if there is no complex operation resulting in a scan cycle longer than 200 milliseconds, it is suggested that the watchdog timer value should be less than 200.
4. If an opration is complex, the scan cycle may be long. Users can check whether the scan cycle is greater than the value stored in D1000 by monitoring D1010~D1012. If the scan cycle is greater than the value stored in D1000, the users can change the value in D1000.

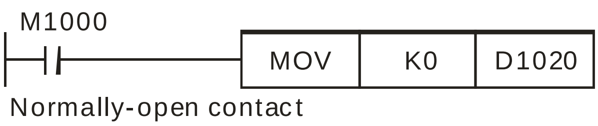
Input filter

D1020

1. Users can set the time it takes for the input terminals X0~X7 to respond by setting D1020. The value in D1020 must be in the range of 0 to 20. (Unit: ms)
2. If the DVP-20PM series motion controller is turned form OFF to ON, the value in S D1020 will automatically become 10.



3. If the program below is executed, the time is takes for the input terminals X0~X7 to respond will be 0 milliseconds. Owing to the fact that the input terminals are connected to resistor-capacitor circuits in series, the shortest time it takes for the input terminals to respond is 50 microseconds.



4. If high-spedd counters and interrupts are used in a program, the value in D1020 does not have any effect.

Communication port

M1120, M1136, M1138, M1139, M1143, D1036, D1109, and D1120

The communication ports with which a DVP-20PM series motion controller is equipped are COM1 (RS-232 port) and COM2 (RS-485 port). They support Modbus ASCII/RTU. The maximum speed available is 115,200 bps. COM3 (RS-232/RS-485 communication card) supports Modbus ASCII. The maximum speed available is 38,400 bps. COM1, COM2, and COM3 can be used simultaneously.

COM1 COM1 can only be used as a slave station. It supports ASCII/RTU, and the adjustment of a communication speed. The maximum speed available is 115,200 bps. It supports the modification of the number of data bits.

COM2 COM2 can be used as a master station or a slave station. It supports ASCII/RTU, and the adjustment of a communication speed. The maximum speed available is 115,200 bps. It supports the modification of the number of data bits.

COM3 COM3 can only be used as a slave station. It supports the ASCII communication format 7, E, 1 (7 data bits, even parity bit, 1 stop bit), and the adjustment of a communication speed. The maximum speed available is 38,400 bps. COM2 and COM3 can not be used as slave stations simultaneously.

◎ Setting a communication format:

COM1
1. D1036 is used to set a communication format. Bit 8~bit 15 in D1036 do not support the setting of a communicaiton format.
2. M1138: The setting of the communication through COM1 is retained.
3. M1139: Selecting an ASCII mode or an RTU mode

COM2
1. D11120 is used to set a communication format.
2. M1120: The setting of the communication through COM2 is retained.
3. M1143: Selecting an ASCII mode or an RTU mode

COM3
1. D1109 is used to set a communication format. Bit 0~bit 3 and Bit 8~bit 15 in D1036 do not support the setting of a communicaiton format.
2. M1136: The setting of the communication through COM3 is retained.

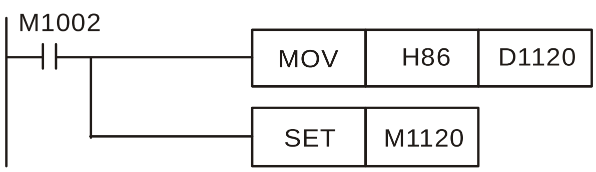
Communication format:

	Contents	0	1
b0	Data length	b0=0: 7	b0=1: 8
b1 b2	Parity bit	b2, b1=00 : None b2, b1=01 : Odd b2, b1=11 : Even	
b3	Stop bit	b3=0: 1 bit	b3=1: 2 bits
b7~b4	b7~b4=0001 (H1) : 110 bps b7~b4=0010 (H2) : 150 bps b7~b4=0011 (H3) : 300 bps b7~b4=0100 (H4) : 600 bps b7~b4=0101 (H5) : 1,200 bps b7~b4=0110 (H6) : 2,400 bps b7~b4=0111 (H7) : 4,800 bps b7~b4=1000 (H8) : 9,600 bps b7~b4=1001 (H9) : 19,200 bps b7~b4=1010 (HA) : 38,400 bps b7~b4=1011 (HB) : 57,600 bps b7~b4=1100 (HC) : 115,200 bps		
b8	Start character	b8=0: None	b8=1: D1124
b9	First terminator	b9=0: None	b9=1: D1125
b10	Second terminator	b10=0: None	b10=1: D1126
b15~b11	Undefined		

Example 1: Modifying the communication format of COM2

If users want to modify the communication format of COM2 on a DVP-20PM series motion controller, they have to add the program shown below to the top of the program in the DVP-20PM series motion controller. After the STOP/RUN switch on the DVP-20PM series motion controller is turned from the STOP position to the RUN position, the state of M1120 will be detected during the first scan cycle. If M1120 is ON, the setting of COM2 will be changed in accordance with the value in D1120.

The communication format of COM2 is changed to the ASCII format 9600, 7, E 1 (9,600bps, 7 data bits, even parity bit, 1 stop bit).



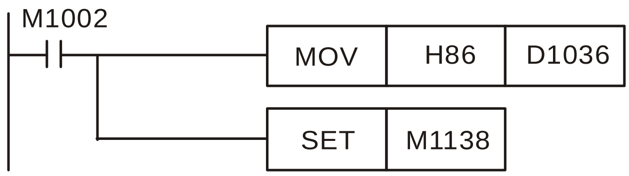
Notes:

1. If COM2 on a DVP-20PM series motion controller is used as a slave station, no communication instruction can exist in the program in the DVP-20PM series motion controller.
2. If the STOP/RUN switch on a DVP-20PM series motion controller is turned from the RUN position to the STOP position after the communication format of COM2 is modified, the new communication format of COM2 will not be changed.
3. If users disconnect a DVP-20PM series motion controller and then power it up after they modify the communication format of COM2, the new communication format of COM2 will be restored to its factory setting.

Example 2: Modifying the communication format of COM1

If users want to modify the communication format of COM1 on a DVP-20PM series motion controller, they have to add the program shown below to the top of the program in the DVP-20PM series motion controller. After the STOP/RUN switch on the DVP-20PM series motion controller is turned from the STOP position to the RUN position, the state of M1138 will be detected during the first scan cycle. If M1138 is ON, the setting of COM1 will be changed in accordance with the value in D1036.

The communication format of COM1 is changed to the ASCII format 9600, 7, E 1 (9,600bps, 7 data bits, even parity bit, 1 stop bit).



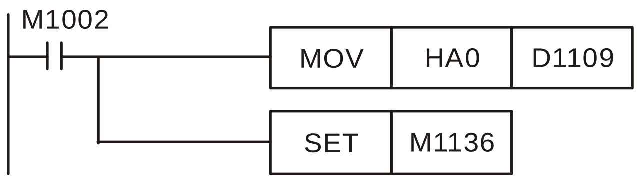
Notes:

1. If the STOP/RUN switch on a DVP-20PM series motion controller is turned from the RUN position to the STOP position after the communication format of COM1 is modified, the new communication format of COM1 will not be changed.
2. If users disconnect a DVP-20PM series motion controller and then power it up after they modify the communication format of COM1, the new communication format of COM1 will be restored to its factory setting.

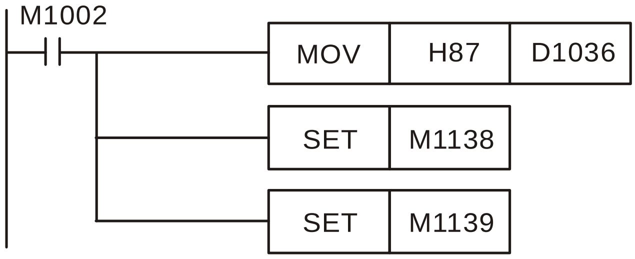
Example 3: Modifying the communication format of COM3

The communication format of COM3 is 7, E 1 (7 data bits, even parity bit, 1 stop bit). If users want to change the communication speed of COM3 on a DVP-20PM series motion controller to 38,400 bps, they have to add the program shown below to the top of the program in the DVP-20PM series motion controller. After the STOP/RUN switch on the DVP-20PM series motion controller is turned from the STOP position to the RUN position, the state of M1136 will be detected during the first scan cycle. If M1136 is ON, the setting of COM3 will be changed in accordance with the value in D1109.

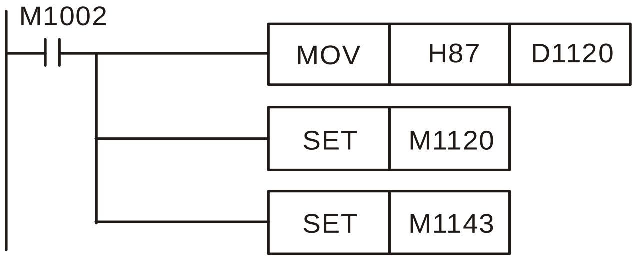
The communication speed of COM3 is changed to 38,400 bps.



Example 4: Using COM1/COM2 in an RTU mode
COM1: (9,600, 8, E, 1, RTU)



COM2: (9,600, 8, E, 1, RTU)



Communication timeout

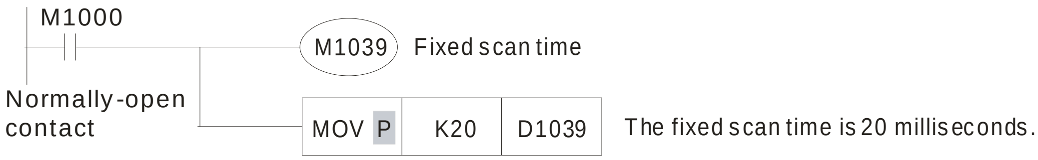
D1038

If an RS-485 port on a DVP-20PM series motion controller functions as a slave station, users can set a communication timeout. The value in D1038 is in the range of 0 to 3,000 (0 to 30 seconds). The unit used is 10 milliseconds. If the value in D1038 is not in the range of 0 to 3,000, the value in D1038 will become 0. The value in D1038 must be less than the value in D1000.

Fixed scan time

M1039 and D1039

1. If M1039 is ON, the time it takes for the program to be scanned will depend on the value in D1039. If the execution of a program is complete, the program will not be scanned again until the fixed scan time set elapses. If the value in D1039 is less than the time it takes for a program to be scanned, the time it takes for a program to be scanned, will be given priority.



2. The values stored in D1010~D1012 include the value stored in D1039.

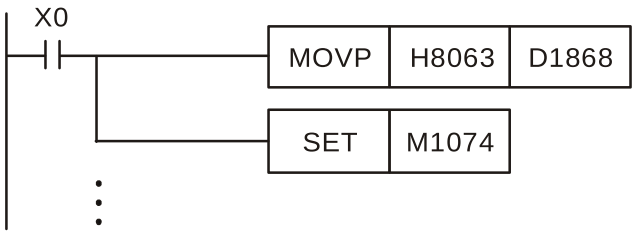
Setting an OX motion subroutine number

M1074 and D1868

Users can specify an Ox motion subroutine by setting D1868. The steps of setting D1868 are as follows.

1. The users have to set bit 14 in D1868 to 1, set bit 15 in D1868 to 1, or set bit 14 and bit 15 in D1868 to 1. Besides, the users have to write K99 (H63) into bit 0~bit 13 in D1868, that is, the Ox motion subroutine number specified is Ox99. To sum up, the users have to write H8063 into D1868.
2. After M1074 is set to ON, the Ox motion subroutine specified by D1868 will be executed.

The program is shown below.



In the main program O100, X0 starts the motion subroutine Ox99.

Detecting expansion

D1140, D1142, and D1143

1. D1140: Number of right-side modules (AD, DA, XA, PT, TC, RT, HC, PU) (8 right-side modules at most)
2. D1142: Number of X devices in a digital module
3. D1143: Number of Y devices in a digital module

Latching device range

D1200~D1211

1. Users can set latching device ranges. The devices in the range of a starting latching device address and a terminal latching device address are latching devices.
2. Please refer to section 3.1 for more information.

Turning the X devices ON/OFF

M1304

If M1304 in a DVP-20PM series motion controller is ON, the X devices (X0~X17) in the DVP-20PM series motion controller can be turned ON/OFF by means of PMSoft.

Right-side special I/O module ID

D1320~D1327

1. If a DVP-20PM series motion controller is connected to special I/O modules, the IDs of the special I/O modules will be stored in D1320~D1327.
2. ID's of the special I/O modules which can be connected to a DVP-20PM series motion controller:

I/O module	ID (Hexadecimal value)	I/O module	ID (Hexadecimal value)
DVP04AD-H2	H'6400	DVP01PU-H2	H'6110
DVP04DA-H2	H'6401	DVP04PT-H2	H'6402
DVP04TC-H2	H'6403	DVP06XA-H2	H'6604
DVP-PM	H'6260	DVP01HC-H2	H'6120

Interrupt register

D1400 and D1401

1. D1400 is an interrupt register. If users set a bit in D1400 to ON, an interrupt will be enabled.

Bit#	Interrupt	Interrupt number
0	Time interrupt	I0
1	External terminal START0/X0*	I1
2	External terminal STOP0/X1*	I2
3	External terminal START1/X2*	I3
4	External terminal STOP1/X3*	I4
5	External terminal X4*	I5
6	External terminal X5*	I6
7	External terminal X6*	I7
8	External terminal X7*	I8

*: Only applicable to DVP-10PM series motion controllers

2. If an interrupt enabled is a time interrupt, users can write the cycle of the interrupt into D1401.
3. There are two types of interrupts.
   - External interrupt: If an interrupt is triggered by the rising edge/falling edge of a pulse received through an external terminal, the execution of the present program will stop, and the interrupt will be executed. After an interrupt is executed, the program which is executed before the interrupt is triggered will be executed.
   - Time interrupt: The execution of the present program stops at regular intervals. Whenever the execution of the present program stops, an interrupt is executed.

Clearing the M-code which is executed

M1744, M1794, and D1703

- If users want to clear the M-code in D1703, they have to set M1744 to ON. If M1744 is set to ON, the value in D1703 will be cleared, and M1794 will be reset.
- If an M code in an Ox motion subroutine is executed, M1794 will be ON. The M-code which is executed is stored in D1703.

Ready flag

M1792, M1872, and M2032

1. Every motion axis uses a ready flag. The X-axis uses M1792, the Y-axis uses M1872, and the Z-axis uses M2032. Users can use the ready flags to judge whether the axes operate.
2. Description of the ready flag for the X-axis: Before the X-axis operates, M1792 is ON. When the X-axis operates, M1792 is OFF. After the first axis finishes operating, M1792 is ON.

Enabling the writing of the present command position

M1751, M1831, and M1991

1. If users want to write the present command position of the X-axis/Y-axis/Z-axis, they have to set M1751/M1831/M1991 to ON. After M1751/M1831/M1991 is set to ON, a value can be written into D1848/D1928/D2008.
2. If users write a value into D1848 when M1751 is OFF, the value in D1848 will be cleared to 0.

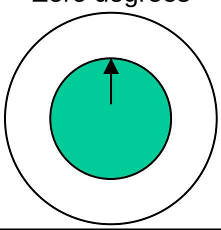
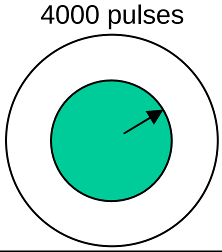
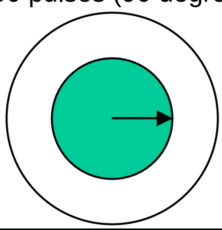
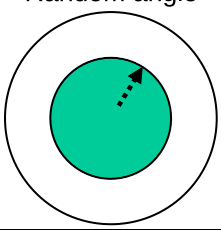
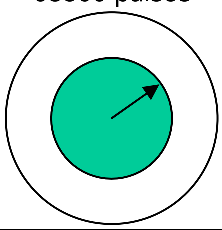
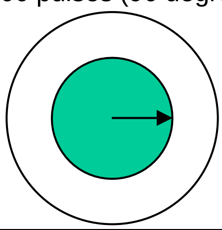
Stopping the uniaxial motion at the angle specified

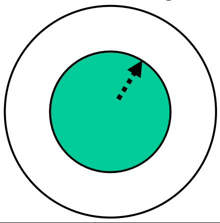
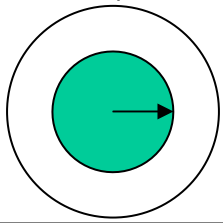
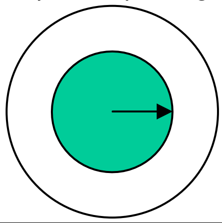
M1761, M1841, and M2001

1. M1761, M1841, and M2001 are used to stop the X-axis, the Y-axis, and the Z-axis at the angles specified in positive JOG motion.
2. Users have to set M1761 to ON, write the number of pulses per cycle into (D1843, D1842), write an angle into (D1839, D1838), and start positive JOG motion.
3. Stopping uniaxial motion at an angle specified

Parameter	X-axis	Y-axis	Z-axis
Stopping at the angle specified	M1761	M1841	M2001
Angle	(D1838, D1839)	(D1918, D1919)	(D1998, D1999)
Number of pulses per cycle	(D1842, D1843)	(D1922, D1923)	(D2002, D2003)

4. If the number of pulses it takes for the motor of an axis to rotate once is 20000, and the angle at which users want to stop the axis is 90 degrees, there will be the states shown below.

Starting position	Stop position of the JOG motion	Final stop position
Zero degrees 	4000 pulses 	5000 pulses (90 degrees) 
Random angle 	63500 pulses 	65000 pulses (90 degrees) 

Starting position	Stop position of the JOG motion	Final stop position
Random angle 	25001 pulses 	45000 pulses (90 degrees) 

Clearing the motion error

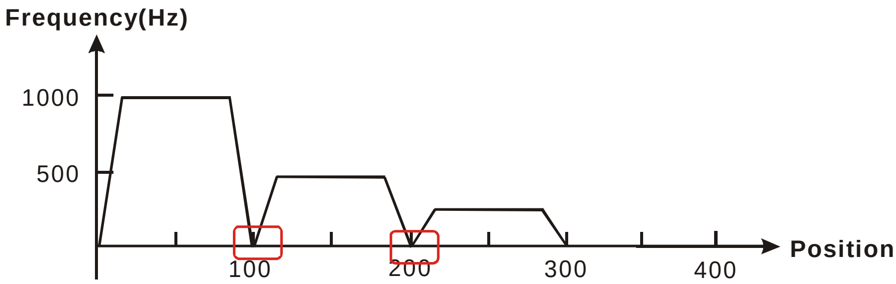
M1793, D1857, M1873, D1937, M2033, and D2017

1. If errors occur in the X-axis, the Y-axis, and the Z-axis, M1793, M1873, and M2033 will be ON, and the error messages which appear will be stored in D1857, D1937, and D2017.
2. If users want to eliminate the error occuring in an axis, they have to clear the error code in the special data register corresponding to the axis, and reset the special auxiliary relay corresponding to the axis.

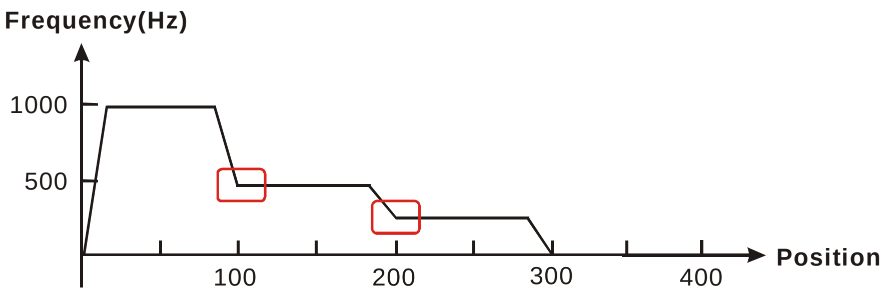
Continuous interpolation

D1796 and M1036

1. If users set the minimum speed to which the speed of continuous interpolation decreses in D1796, the smaller speed will be taken as a turning point after the setting value in D1796 is compared with the acutal speed to which the speed of continuous interpolation decreses.
2. If the value in D1796 is greater than 0, a continuous path can be created, and only G1/G2/G3/LIN/CW/CCW can be used to create a continuous created. If M1036 is set to ON, other applied instructions can be used to create a continuous path.
3. If the value in D1796 is K0 (there is no continuous interpolation), the speed of motion will decrease to 0 Hz no matter what the actual deceleration is.



4. The value in D1796 is K500 (there is continuous interpolation). After the value in D1796 is compared with the acutal deceleration, the smaller deceleration will be taken as a turning point.

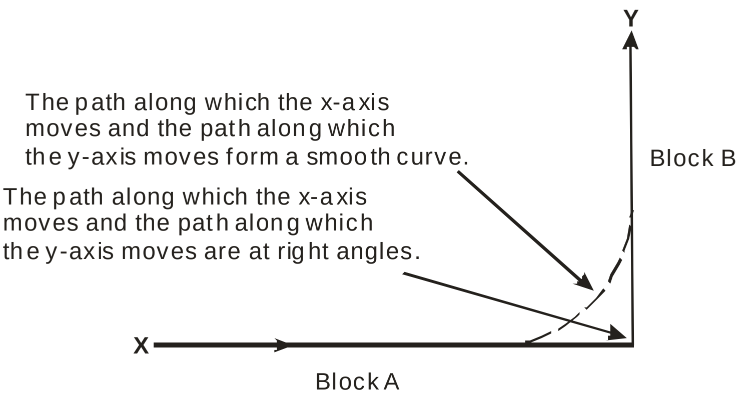


5. Continuous path

   If D1796 is not set, the path along which the X-axis moves and the path along which the Y-axis moves will be at right angles. If D1796 is set, the path along which the X-axis moves and the path along which the Y-axis moves will form a smooth curve.

   G01 X100 F1000; (Block A)

   Y100; (Block B)



Block A: Path along which the X-axis moves; Block B: Path along which the Y-axis moves

Setting the percentage for the values of the speed parameters of the G-codes

D1798

1. If the value in D1798 is 100, the speeds of the G-codes used will be the orginial speeds. If the value in D1798 is 1000, the speeds of the G-codes used will be multipled by 10. If the value in D1798 is 50, the speeds of the G-codes used will be half the original speed.
2. If the result gotten from the multiplication of the speed of a G-code by the percentage set in D1798 is greater than 500000 Hz, the G-code will move the axes used at a speed of 500000 Hz.

Setting the polarities of the input terminals

D1799

If users want to turn an input terminal into a Form A contact, they have to set the bit corresponding to the input terminal to ON. If the users want to turn an input terminal into a Form B contact, they have to set the bit corresponding to the input terminal to OFF.

Bit#	Polarity of an input terminal of the X-axis	Bit#	Polarity of the input terminal of the Y-axis
0	PG0	8	PG1
1	B0	9	B1
2	A0	10	A1
3	LSN0	11	LSN1
4	LSP0	12	LSP1
5	DOG0	13	DOG1
6	STOP0	14	STOP1
7	START0	15	START1

Reading the states of the input terminals

D1800

If a bit in D1800 is ON, the input terminal corresponding to the bit receives a signal. If a bit in D1800 is OFF, the input terminal corresponding to the bit does not receive a signal.

Bit#	State of an input terminal of the X-axis	Bit#	State of an input terminal of the Y-axis
0	PG0	8	PG0
1	B0	9	B1
2	A0	10	A1
3	LSN	11	LSN
4	LSP	12	LSP
5	DOG	13	DOG
6	STOP	14	STOP
7	START	15	START

Setting the polarities of the input terminals

D1804

If users want to turn an input terminal into a Form A contact, they have to set the bit corresponding to the input terminal to ON. If the users want to turn an input terminal into a Form B contact, they have to set the bit corresponding to the input terminal to OFF.

Bit#	Polarity of an input terminal of the Z-axis
0	PG2
1	B1
2	A1
3	LSN2
4	LSP2
5	DOG2

Reading the states of the input terminals

D1805

If a bit in D1805 is ON, the input terminal corresponding to the bit receives a signal. If a bit in D1805 is OFF, the input terminal corresponding to the bit does not receive a signal.

Bit#	State of an input terminal of the Z-axis
0	PG2
1	B1
2	A1
3	LSN2
4	LSP2
5	DOG2

Setting a filter coefficient for the input terminals

D1806

1. Users can set the hardware filter for the input terminals START, STOP, DOG, LSN, LSP, and PG by means of D1806.
2. Users can set a filter coefficient for the input terminal MPG0/1 by setting the high byte in D1806.
3. Filter coefficeint $= \frac{85000}{2^{N+4}}$ (kHz); N=1~19

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		

4. If the value in D1806 is 0, no signals will be filtered.
5. If the value in D1806 is H0A0A, the filter coefficient for START, STOP, DOG, LSN, LSP, and PG $= \frac{85000}{2^{10+4}} = 5.187988$, and the signals whose frequencies are higher than 5.187988 kHz will be removed.

O100 error

M1953, D1802, and D1803

1. If an error occurs in O100, M1953 will be ON, the error code corresponding to the error will be stored in D1802, and the step address at which the error occurs will be stored in D1803.
2. Please refer to appendix A in chapter 15 for more information about error codes.

Setting the parameters of the X-axis/ Y-axis/z-axis

D1816, D1896, and D1976

D1816 is for the X-axis, D1896 is for the Y-axis, and D1976 is for the Z-axis.

Bit#	Parameter of the X-axis/Y-axis/Z-axis	Bit#	Parameter of the X-axis/Y-axis/Z-axis
0	Unit[*1]	8	Direction in which the axis returns home[*4]
1		9	Mode of returning home[*4]
2	Ratio[*2]	10	Mode of triggering the return to home[*4]
3		11	Direction in which the motor rotates[*4]
4	Output type[*3]	12	Relative/Absolute coordinates[*4]
5		13	Mode of triggering the calculation of the target position[*4]
6		14	Curve[*4]
7		15	

*1:

b1	b0	Unit
0	0	Motor unit
0	1	Mechanical unit
1	0	Compound unit
1	1	

	Motor unit	Compound unit	Mechanical unit
Position	pulse	μm	
	pulse	mdeg	
	pulse	10^{-4} inches	
Speed	pulse/second		centimeter/minute
	pulse/second		10 degrees/minute
	pulse/second		inch/minute

*2:

b3	b2	Ratio
0	0	10^0
0	1	10^1
1	0	10^2
1	1	10^3

*3:

b5	b4	Description
0	0	Positive-going pulse+Negative-going pulse
0	1	Pulse+Direction
1	0	A/B-phase pulse (two phases and two inputs)
1	1	

*4:

Bit#	Description
8	Bit 8=0: The value indicating the present command position of the axis decreases progressively. Bit 8=1: The value indicating the present command position of the axis increases progressively.
9	Bit 9=0: Normal mode ; bit 9=1: Overwrite mode
10	Bit 10=0: The return to home is triggered by a transition in DOG's signal from high to low. Bit 10=1: The return to home is triggered by a transition in DOG's signal from low to high.
11	Bit 11=0: When the motor rotates clockwise, the value indicating the present command position of the axis increases. Bit 11=1: When the motor rotates clockwise, the value indicating the present command position of the axis decreases.
12	Bit 12=0: Absolute coordinates Bit 12=1: Relative coordinates

Bit#	Description
13	Bit 13=0: The calculation of the target position of the axis is triggered by a transition in DOG's signal from low to high. Bit 13=1: The calculation of the target position of the axis is triggered by a transition in DOG's signal from high to low. (The setting of bit 13 is applicable to the insertion of single-speed motion, and the insertion of two-speed motion.)
14	Bit 14=0: Trapezoid curve Bit 14=1: S curve

Setting the noncyclic electronic cam

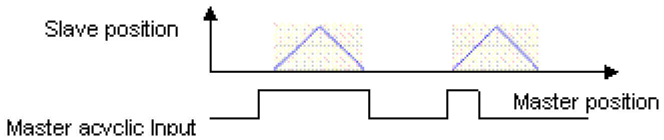
D1832, D1833, and D1834

Only DVP-20PM series motion controllers support this function.

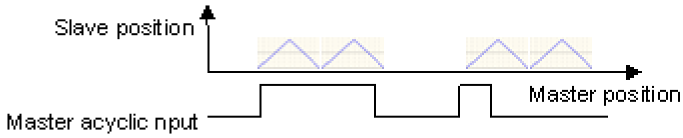
1. D1832: Setting the number of times noncyclic electronic cam motion is repeated

   Users can set the number of times noncyclic electronic cam motion is repeated by means of D1832. If the value in D1832 is greater than H8000 (bit 15=1), there will be cyclic electronic cam motion. If the value in D1832 is 0, noncyclic electronic cam motion will not be repeated. If the value in D1832 is 1, noncyclic electronic cam motion will be repeated once.

   D1832=0



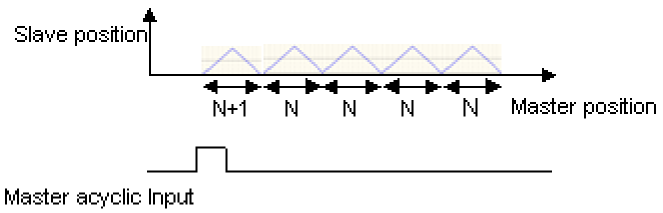
   D1832=1



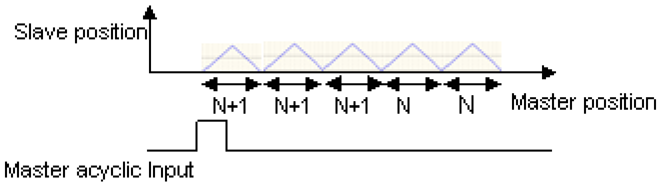
2. D1833: Number of remaining pulses sent by the master axis of an electronic cam

   If the number of pulses sent by the master axis of an electronic cam is not divisible by the number of pulses per ccycle, users can divide the number that is left by setting D1833. If the number of pulses sent by the master axis of an electronic cam is 202, and the number of pulses per cycle is 50, two pulses will be left after four cycles. If the value in D1832 is 3, and the value in D1833 is 2, one pulse will be added to the first cycle and the second cycle.

   D1832=4 and D1833=1

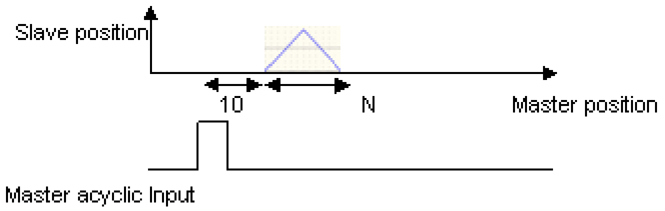


   D1832=4 and D1833=3

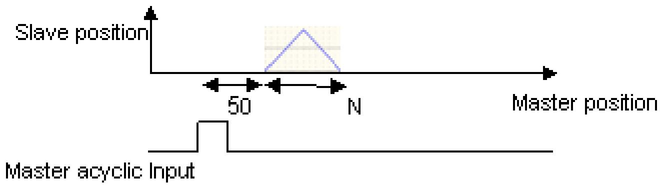


*The value in D1833 can not be greater than the value in D1832.

3. D1834: Number of pulses sent by the master axis of the electronic cam before the electronic cam is started
   Users can delay the sending of pulses by the slave axis of a noncyclic electronic cam by setting D1834. When the number of pulses sent by the master of a noncyclic electronic cam sends is equal to the value in D1834, the slave axis of the noncyclic electronic cam sends pulses.
   D1834=10
   (The slave axis sends pulses after the master axis sends 10 pulses.)



   D1834=50
   (The slave axis sends pulses after the master axis sends 50 pulses.)



Operation command for the X-axis/ Y-axis/Z-axis

D1846, D1926, and D2006

D1846 is for the X-axis, D1926 is for the Y-axis, and D2006 is for the Z-axis.

Bit#	Operation command for the X-axis/Y-axis/Z-axis	Bit#	Operation command for the X-axis/Y-axis/Z-axis
0	The motion of the axis specified is stopped by software.	8	A mode of single-speed motion is activated.
1	The motion of the axis specified is started by software.	9	A mode of inserting single-speed motion is activated.
2	The axis specified operates in a JOG+ mode.	10	A mode of two-speed motion is activated.
3	The axis specified operates in a JOG- mode.	11	A mode of inserting two-speed motion is activated.
4	A mode of variable motion is activated.	12	0: The execution of the Ox motion subroutine set stops. 1: The execution of the Ox motion subroutine set starts.
5	A manual pulse generator is operated.	13*2	Starting the cyclic electronic cam specified
6	A mode of triggering the return to home is activated.	14*2	Starting the noncyclic electronic cam specified
7*1	Triggering single-speed motion by means of an external signal	15	-

Input terminals on DVP20PM00D for manual pulse generators: X-axis (A0+/A0-, B0+/B0-); Y-axis (A1+/A1-, B1+/B1-)

Input terminals on DVP20PM00M for manual pulse generators: X-axis (A0+/A0-, B0+/B0-); Y-axis and Z-axis (A1+/A1-, B1+/B1-)

*1: Only the Y-axis supports this function. (D1926)

*2: Only the X-axis supports this function. (D1846)

Mode of operation

D1847, D1927, and D2007

D1847 is for the X-axis, D1927 is for the Y-axis, D2007 is for the Z-axis, D2087 is for the A-axis, D2167 is for the B-axis, and D2247 is for the C-axis.

Bit#	Mode of operation	Bit#	Mode of operation
0	-	8	Setting a stop signal
1	-	9	
2	Mode of sending a CLR signal	10	
3	Setting the CLR output to ON/OFF	11	01: The motion of the axis specified is interrupted, and an electronic cam chart is inserted. 10: G-codes can be created in a cam chart. 11: The axis specified can operate clockwise and operate counterclockwise.
4	Setting the polarity of the CLR output	12	
5	Mode of stopping the motor used when STOP is ON	13	-
6	Limitation on the present position of the slave axis controlled by the manual pulse generator used	14	-
7	Mode of stopping the motor used when the motor used comes into contact with a positive limit switch/negative limit switch	15	Restoring the DVP-20PM series motion controller to the factory settings

Bit#	Description
2	Bit 2=0: After the axis specified returns home, the CLR output will send a 130 millisecond signal to the servo drive used, and the present position of the servo drive which is stored in a register in the servo drive will be cleared. Bit 2=1: The CLR output functions as a general output. Its state is controlled by bit 3.
3	Bit 3=0: The CLR output is OFF. Bit 3=1: The CLR output is ON.
4	Bit 4=0: The CLR output is a Form A contact. Bit 4=1: The CLR output is a Form B contact.
5	Bit 5=0: If STOP is ON when the motor used operates, the motor will decelerate and stop. If a motion command is sent, the motor will ignore the previous unfinished distance, and rotate for the distance specified. Bit 5=1: If STOP is ON when the motor used operates, the motor will decelerate and stop. If a motion command is sent, the motor will complete the previous unfinished distance, and then rotate for the distance specified.
6	Bit 6=0: There is no limitation on the present position of the slave axis controlled by the manual pulse generator used Bit 6=1: The present position of the slave axis controlled by the manual pulse generator used has to be in the range of the P (I) set to the P (II) set. If the present position of the slave axis controlled by the manual pulse generator used is not in the range of the P (I) set to the P (II) set, the slave axis will decelerate and stop.
7	Bit 7=0: If the motor used comes into contact with a positive limit switch/negative limit switch when it rotates, it will decelerate and stop. Bit 7=1: If the motor used comes into contact with a positive limit switch/negative limit switch when it rotates, it will stop immediately.

Bit#	Description
8 9 10	When single-speed motion, two-speed motion, inserting single-speed motion, or inserting two-speed motion is executed, A0 or B0 can be used to stop the motion. Bit 10~bit 8=K0 (000) or other values: No function Bit 10~bit 8=K1 (001): Motion is stopped immediately when there is a transition in A0's signal from low to high. Bit 10~bit 8=K2 (010): Motion is stopped immediately when there is a transition in A0's signal from high to low. Bit 10~bit 8=K3 (011): Motion is stopped immediately when there is a transition in B0's signal from low to high. Bit 10~bit 8=K4 (100): Motion is stopped immediately when there is a transition in B0's signal from high to low.
11 12	Bit 12~bit 11=K1 (01): The motion of the axis specified is interrupted, and an electronic cam chart is inserted. Bit 12~bit 11=K2 (10): G-codes can be created in a cam chart. Bit 12~bit 11=K3 (11): The axis specified can operate clockwise and operate counterclockwise.

State of the axis

D1856, D1936, and D2016

D1856 is for the X-axis, D1936 is for the Y-axis, and D2016 is for the Z-axis.

Bit#	State of the X-axis/Y-axis/Z-axis
0	Positive-going pulses are being output.
1	Negative-going pulses are being output.
2	The axis specified is operating.
3	An error occurs.
4	The axis specified pauses.
5	The manual pulse generator used generates positive-going pulses.
6	The manual pulse generator used generates negative-going pulses.
7	-

Enabling the PID closed-loop control

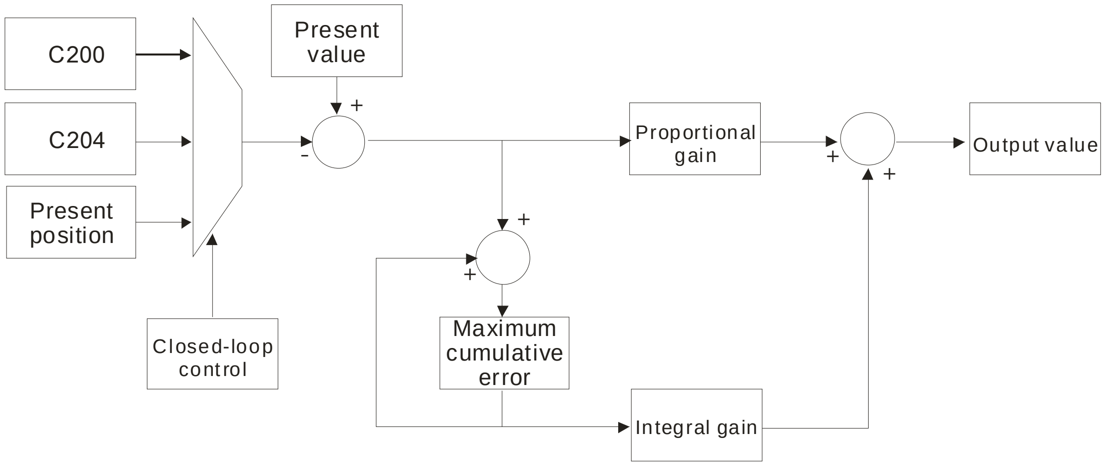
D1877, D1878, D1880, D1882, and D1888

1. D1877 is used to enable the PID closed-loop control of the X-axis. The source of the feedback given to a closed loop is shown below.

Value	Source
1	C200
2	C204
3	Present position (D1848, D1928, D2008)
Others	Disabling PID closed-loop control

2. The value in (D1879, D1878) indicates the present value of the command for the PID closed-loop control of the X-axis. The source of the feedback given to a closed loop is controlled by the closed loop so that the output value of the command for the closed loop is the same as the setting value of the command for the closed loop.
3. The value in (D1881, D1880) indicates the proportional gain for the PID closed-loop control of the X-axis.
4. The value in (D1883, D1882) indicates the integral gain for the PID closed-loop control of the X-axis.
5. The value in (D1889, D1888) indicates the maximum cumulative error for the PID closed-loop control of the X-axis. D1888 and D1889 are used to limit the difference between an input value and an output value.

6. Closed-loop control



After the closed-loop control is used to select the source of the feedback given to the closed loop, the source value of the feedback will be compared with the present value of the command sent to the closed loop. Owing to the PID closed-loop control, the source value of the feedback is the same as the present value of the command. The maximum cumulative error set is used to limit the difference between the source value of the feedback and the present value of the command. If the difference is greater than the maximum cumulative error, there will no output.

3.12 Special Data Registers for Motion Axes

The special data registers for the X-axis, the Y-axis, and the Z-axis in a DVP-20PM series motion controller are described below. Please refer to this section for more information about the setting of the special data registers.

Special D device number						Special data register	Setting range	Default value
X-axis		Y-axis		Z-axis				
HW*1	LW*1	HW	LW	HW	LW			
-	D1816	-	D1896	-	D1976	Setting the parameters of the axis specified	Bit 0~bit 15	H0
D1819	D1818	D1899	D1898	D1979	D1978	Number of pulses it takes for the motor of the axis specified to rotate once (A)	1~+2,147,483,647 pulses/revolution	K2,000
D1821	D1820	D1901	D1900	D1981	D1980	Distance generated after the motor of the axis specified rotate once (B)	1~+2,147,483,647 *2	K1,000
D1823	D1822	D1903	D1902	D1983	D1982	Maximum speed (V_{MAX}) at which the axis specified rotates	0~+2,147,483,647 *3	K500,000
D1825	D1824	D1905	D1904	D1985	D1984	Start-up speed (V_{BIAS}) at which the axis specified rotates	0 ~ +2,147,483,647 *3	K0
D1827	D1826	D1907	D1906	D1987	D1986	JOG speed (V_{JOG}) at which the axis specified rotates	0~+2,147,483,647 *3	K5,000
D1829	D1828	D1909	D1908	D1989	D1988	Speed (V_{RT}) at which the axis specified returns home	0~+2,147,483,647 *3	K50,000
D1831	D1830	D1911	D1910	D1991	D1990	Speed (V_{CR}) to which the speed of the axis specified decreases when the axis returns home	0~+2,147,483,647 *3	K1,000
-	D1832	-	D1912	-	D1992	Number of PG0 signals for the axis specified	0~+32,767 PLS	K0
-	D1833	-	D1913	-	D1993	Number of supplementary pulses for the axis specified	-32,768~+32,767 PLS	K0
D1835	D1834	D1915	D1914	D1995	D1994	Home position of the axis specified	0~±999,999 *1	K0
-	D1836	-	D1916	-	D1996	Time (T_{ACC}) it takes for the axis specified to accelerate	10~+32,767 ms	K100
-	D1837	-	D1917	-	D1997	Time (T_{DEC}) it takes for the axis specified to decelerate	10~+32,767 ms	K100
D1839	D1838	D1919	D1918	D1999	D1998	Target position of the axis specified (P (I))	-2,147,483,648~+2,147,483,647 *1	K0
D1841	D1840	D1921	D1920	D2001	D2000	Speed at which the axis specified rotates (V (I))	0~+2,147,483,647 *1	K1000
D1843	D1842	D1923	D1922	D2003	D2002	Target position of the axis specified (P (II))	-2,147,483,648~+2,147,483,647 *1	K0
D1845	D1844	D1925	D1924	D2005	D2004	Speed at which the axis specified rotates (V (II))	0~+2,147,483,647 *2	K2,000
-	D1846	-	D1926	-	D2006	Operation command	Bit 0~bit 15	H0
-	D1847	-	D1927	-	D2007	Mode of operation	Bit 0~bit 15	H0
D1849	D1848	D1929	D1928	D2009	D2008	Present command position of the axis specified (Pulse)	-2,147,483,648~+2,147,483,647 *1	K0
D1851	D1850	D1931	D1930	D2011	D2010	Present command speed of the axis specified (PPS)	0~+2,147,483,647 PPS	K0
D1853	D1852	D1933	D1932	D2013	D2012	Present command position of the axis specified (unit *3)	-2,147,483,648~+2,147,483,647 *1	K0
D1855	D1854	D1935	D1934	D2015	D2014	Present command speed of the axis specified (unit *3)	0~+2,147,483,647 PPS	K0
-	D1856	-	D1936	-	D2016	State of the axis specified	Bit 0~bit 15	H0
-	D1857	-	D1937	-	D2017	Axis error code	Please refer to appendix A for more information.	H0
-	D1858	-	D1938	-	D2018	Electronic gear of the axis specified (Numerator)	1~+32,767	K1

Special D device number						Special data register	Setting range	Default value
X-axis		Y-axis		Z-axis				
HW*1	LW*1	HW	LW	HW	LW			
-	D1859	-	D1939	-	D2019	Electronic gear of the axis specified (Denominator)	1~+32,767	K1
D1861	D1860	D1941	D1940	D2021	D2020	Frequency of pulses generated by the manual pulse generator for the axis specified	Frequency of pulses generated by the manual pulse generator for the axis specified	K0
D1863	D1862	D1943	D1942	D2023	D2022	Number of pulses generated by the manual pulse generator for the axis specified	Number of pulses generated by the manual pulse generator for the axis specified	K0
-	D1864	-	D1944	-	D2024	Response speed of the manual pulse generator for the axis specified	Response speed of the manual pulse generator for the axis specified	K5
D1865	-	-	-	-	-	Mode of stopping Ox0~Ox99	Users have to set a value according to their needs.	K0
D1867	D1866	D1947	D1946	D2027	D2026	Electrical zero of the axis specified	Users have to set a value according to their needs.	K0
D1868	-	-	-	-	-	Setting an Ox motion subroutine number	Users have to set a value according to their needs.	K0
D1869	-	-	-	-	-	Step address in the Ox motion subroutine executed at which an error occurs	Users have to set a value according to their needs.	K0

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

*2: Unit: μm/rev, mdeg/rev, and 10^{-4} inches/rev

*3: The unit used varies with the setting of bit 0 and bit 1 in D1816/D1896/D1976.

3.12.1 Descriptions of the Special Data Registers Related to Motion

X-axis		Y-axis		Z-axis		Setting the parameters of the axis specified
HW	LW	HW	LW	HW	LW	
	D1816		D1896		D1976	

[Description]

Bit 0~bit 15 in D1816 (D1896, D1976) are described below.

1. Bit 0 and bit 1 in D1816 (D1896, D1976): Unit

b1	b0	Unit	Description
0	0	Motor unit	A pulse is a unit.
0	1	Mechanical unit	A micrometer, 10-4 inches, or a degree is a unit.
1	0	Compound unit	Position: A micrometer, 10-4 inches, or a degree is a unit. (Mechanical unit) Speed: A pulse is a unit. (Motor unit)
1	1		

<table>
<tr><th></th><th>Motor unit</th><th>Compound unit</th><th>Mechanical unit</th></tr>
<tr><th rowspan="3">Position</th><td>pulse</td><td colspan="2">μm</td></tr>
<tr><td>pulse</td><td colspan="2">mdeg</td></tr>
<tr><td>pulse</td><td colspan="2">10^-4 inches</td></tr>
<tr><th rowspan="3">Speed</th><td colspan="2">pulse/second</td><td>centimeter/minute</td></tr>
<tr><td colspan="2">pulse/second</td><td>10 degrees/minute</td></tr>
<tr><td colspan="2">pulse/second</td><td>inch/minute</td></tr>
</table>

- Position: Home position of the axis specified, target position of the axis specified (P (I)), target position of the axis specified (P (II)), and present command position of the axis specified
- Speed: Maximum speed (V_{MAX}) at which the axis specified rotates, start-up speed (V_{BIAS}) at which the axis specified rotates, JOG speed (V_{JOG}) at which the axis specified rotates, speed (V_{RT}) at which the axis specified returns home, speed (V_{CR}) to which the speed of the axis specified decreases when the axis returns home, speed at which the axis specified rotates (V (I)), and speed at which the axis specified rotates (V (II))

- Example 1:
  Bit [1:0]=00⇨Motor unit
  Position: Pulse
  Speed: Pulse/second (PPS)
  Target position of the axis specified (P (I)): 10,000 pulses
  Speed at which the axis specified rotates: 10K PPS
  After the DVP-20PM series motion controller sends 10,000 pulses, the axis specified can move to the target position specified. (The frequency of pulses is 10K PPS.) The distance for which the axis specified can move after a pulse is sent is calculated according to the physical quantity used.
- Example 2:
  Bit [1:0]=01⇨Mechanical unit
  Position: μm
  Speed: Centimeter/minute
  N=0
  D1818 (D1898, D1978)=1,000 (pulses/revolution)
  D1820 (D1900, D1980)=100 (micrometers/revolution)
  P (I)=10,000 (micrometers)
  V (I)=6 (centimeters/minute)
  The number of pulses sent by the DVP-20PM series motion controller and the frequency of pulses are calculated below.

$$\text{Distance} = \underbrace{\frac{\text{Distance}}{\text{Revolution}}}_{B} \times \underbrace{\frac{\text{Revolution}}{\text{Number of pulses}}}_{1/A} \times \text{Number of pulses}$$

  Number of pulses it takes for the axis specified to move to the target position specified $= \frac{\text{P (I)}\,\mu\text{m}}{B/A} = \text{P (I)} \times \frac{A}{B} = 100{,}000\ (\text{pulses})$

  Speed at which the axis specified rotates (V (I)): 6 (centimeters/minute)=60,000/60 (micrometers/second)

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

  The frequency of pulses calculated by the DVP-20PM series motion controller

$$= V(I) \times \frac{10^4}{60} \times \frac{A}{B} = \frac{60{,}000}{60} \times \frac{1{,}000}{100} = 10{,}000 \quad \text{(PPS)}$$

- Example 3
  Bit [1:0]=10 or 11⇨Compound unit
  Position: Micrometer
  Speed: Pulse/second (PPS)
  N=0
  D1818 (D1898, D1978)=2,000 (pulses/revolution)
  D1820 (D1900, D1980)=100 (micrometers/revolution)
  P (I)=10,000 (micrometers)
  V (I)=10K (PPS)
  The number of pulses sent by the DVP-20PM series motion controller is calculated below.
  Number of pulses it takes for the axis specified to move to the target position specified

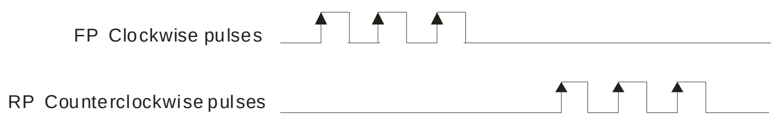
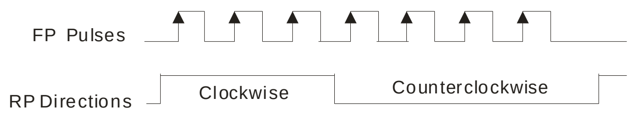
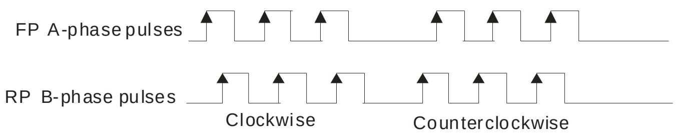
$$= \frac{\text{P (I)}\,\mu\text{m}}{B/A} = \text{P (I)} \times \frac{A}{B} = 200{,}000\ (\text{pulses})$$

2. Bit 2 and bit 3 in D1816 (D1896, D1976): Ratio

   Position: The home position of the axis specified, the target position of the axis specified (P (I)), the target position of the axis specified (P (II)), and the present command position of the axis specified must be multiplied by a ratio.

b3	b2	Ratio
0	0	Position×10^0
0	1	Position×10^1
1	0	Position×10^2
1	1	Position×10^3

3. Bit 4 and bit 5 in D1816 (D1896, D1976): Output type

<table>
<tr><th>b5</th><th>b4</th><th>Output type (positive logic)</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>
</td><td>Counting up/down</td></tr>
<tr><td>0</td><td>1</td><td>
</td><td>Pulses+Directions</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">
</td><td>A/B-phase pulses</td></tr>
<tr><td>1</td><td>1</td><td>Four times the frequency of A/B-phase pulses</td></tr>
</table>

4. Bit 6 in D1816 (D1896, D1976): Setting a PWM mode

   Bit 6=1: If positive JOG motion is started, Y0~Y3 will execute PWM.

5. Bit 8 in D1816 (D1896, D1976): Direction in which the axis specified returns home

   Bit 8=0: The value indicating the present command position of an axis specified decreases, and the axis returns home in the negative direction.

   Bit 8=1: The value indicating the present command position of an axis specified increases, and the axis returns home in the positive direction.

6. Bit 9 in D1816 (D1896, D1976): Mode of returning home

   Bit 9=0: Normal mode

   After DOG's signal is generated, the motor used will rotate for a specific number of PG0 pulses, then rotate for a specific number of supplementary pulses, and finally stop.

   Bit 9=1: Overwrite mode

   After DOG's signal is generated, the motor used will rotate for a number of PG0 pulses or rotate for a number of supplementary pulses, and then stop.

7. Bit 10 in D1816 (D1896, D1976): Mode of triggering the return to home

   Bit 10=0: The return to home is triggered by a transition in DOG's signal from high to low.

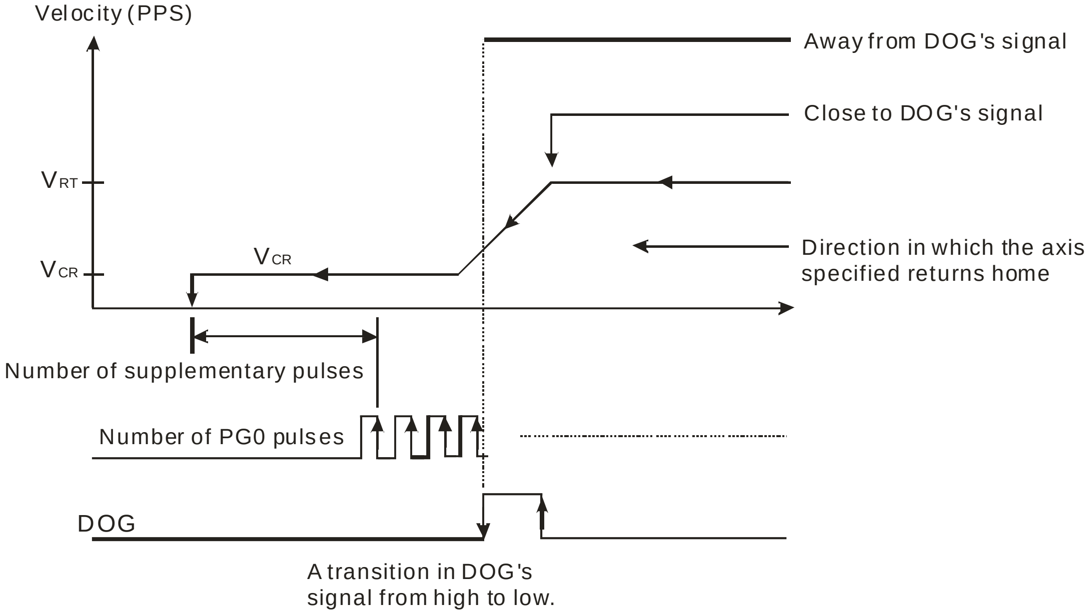
   Bit 10=1: The return to home is triggered by a transition in DOG's signal from low to high.

   - Bit [9:10] in D1816 (D1896, D1976) is 00.⇨The mode of returning home is a normal mode, and the return to home is triggered by a transition in DOG's signal from high to low.

     Steps: The motor used rotates at the speed V_{RT}. When DOG's signal is generated, the speed of the motor begins to decrease to the speed V_{CR}. After DOG's signal goes from high to low, the motor will rotate for a specific number of PG0 pulses, and then rotate for a specific number of supplementary pulses, and finally stop.

     If the number of PG0 pulses or the number of supplementary pulses is not large, the speed of the motor used will decrease to the speed V_{CR} after DOG's signal is generated. After DOG's signal goes from high to low, the motor will rotate for a specific number of PG0 pulses, and then rotate for a specific number of supplementary pulses, and finally stop whether the its speed is V_{CR}.

If the number of PG0 pulses is 0, and the number of supplementary pulses is 0, the motor used will stop after DOG's signal is generated and there is a transition in DOG's signal from high to low.

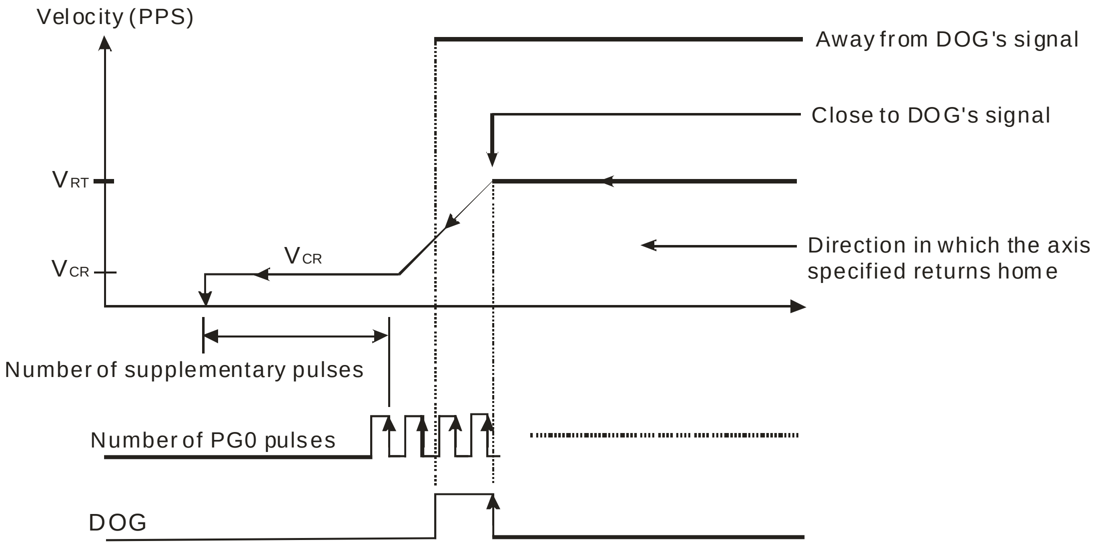


- Bit [9:10] in D1816 (D1896, D1976) is 01.⇨The mode of returning home is a normal mode, and the return to home is not triggered by a transition in DOG's signal from high to low.

  Steps: The motor used rotates at the speed V_{RT}. When DOG's signal is generated, the speed of the motor begins to decrease to the speed V_{CR}. After the motor rotates for a specific number of PG0 pulses, and rotate for a specific number of supplementary pulses, it will stop.

  If the number of PG0 pulses or the number of supplementary pulses is not large, the speed of the motor used will decrease to the speed V_{CR} after DOG's signal is generated. After the motor rotates for a specific number of PG0 pulses, and rotates for a specific number of supplementary pulses, it will stop whether its speed is V_{CR}.

  If the number of PG0 pulses is 0, and the number of supplementary pulses is 0, the motor used will stop after DOG's signal is generated.

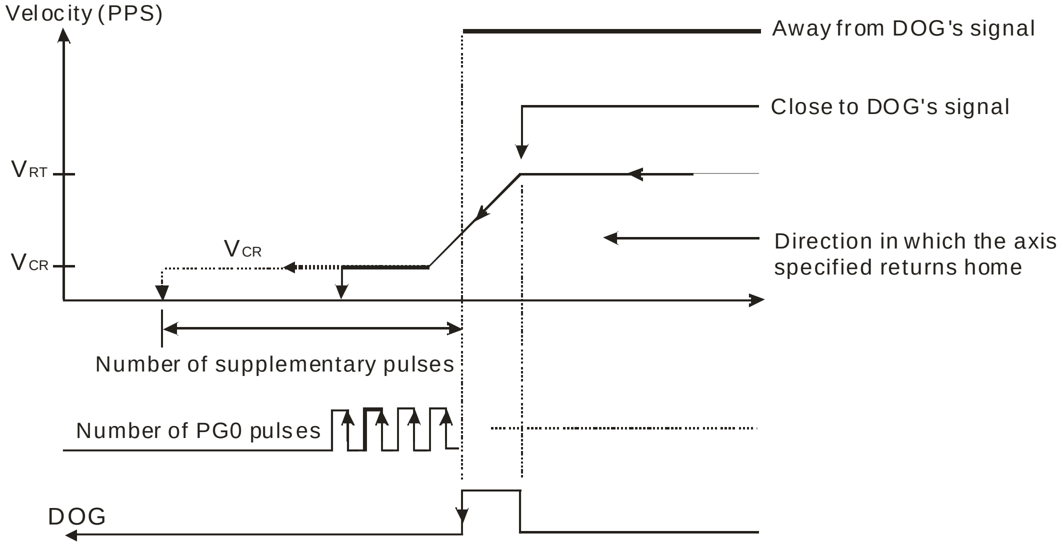


- Bit [9:10] in D1816 (D1896, D1976) is 10.⇨ The mode of returning home is an overwrite mode, and the return to home is triggered by a transition in DOG's signal from high to low.

  Steps: The motor used rotates at the speed V_{RT}. When DOG's signal is generated, the speed of the motor begins to decrease to the speed V_{CR}. After DOG's signal goes from high to low, the motor will rotate for a specific number of PG0 pulses, or rotate for a specific number of supplementary pulses, and then stop.

  If the number of PG0 pulses or the number of supplementary pulses is not large, the speed of the motor used will decrease to the speed V_{CR} after DOG's signal is generated. After DOG's signal goes from high to low, the motor will rotate for a specific number of PG0 pulses, or rotate for a specific number of supplementary pulses, and then stop whether the its speed is V_{CR}.

If the number of PG0 pulses is 0, and the number of supplementary pulses is 0, the motor used will stop after DOG's signal is generated and there is a transition in DOG's signal from high to low.

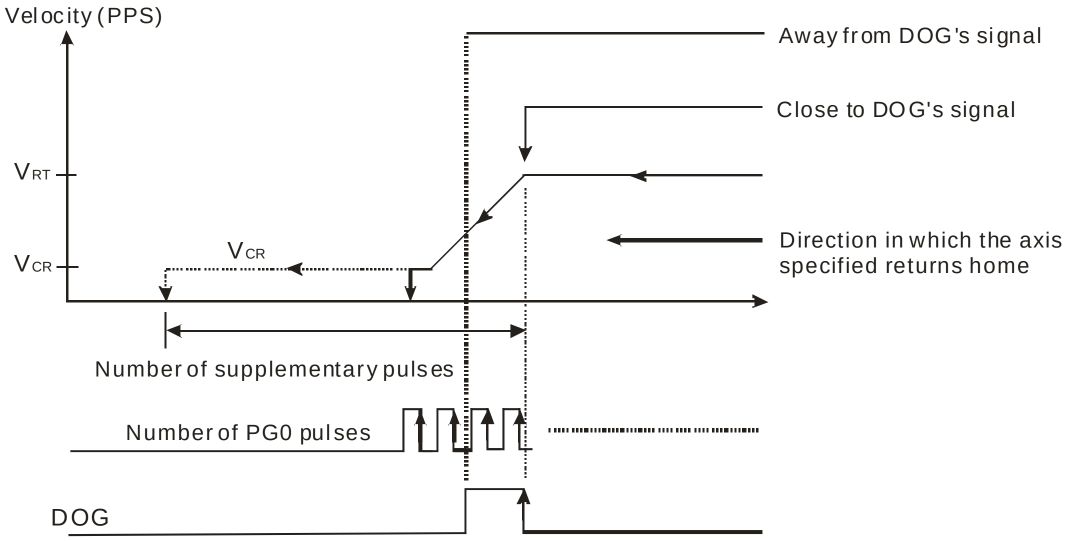


- Bit [9:10] in D1816 (D1896, D1976) is 11.➩ The mode of returning home is an overwrite mode, and the return to home is not triggered by a transition in DOG's signal from high to low.

  Steps: The motor used rotates at the speed V_{RT}. When DOG's signal is generated, the speed of the motor begins to decrease to the speed V_{CR}. After the motor rotates for a specific number of PG0 pulses, or rotate for a specific number of supplementary pulses, it will stop.

  If the number of PG0 pulses or the number of supplementary pulses is not large, the speed of the motor used will decrease to the speed V_{CR} after DOG's signal is generated. After the motor rotates for a specific number of PG0 pulses, or rotates for a specific number of supplementary pulses, it will stop whether its speed is V_{CR}.

  If the number of PG0 pulses is 0, and the number of supplementary pulses is 0, the motor used will stop after DOG's signal is generated.



8. Bit 11 in D1816 (D1896, D1976): Direction in which the motor used rotates
   - Bit 11=0: When the motor rotates clockwise, the value indicating the present command position of the axis specified increases.
   - Bit 11=1: When the motor rotates clockwise, the value indicating the present command position of the axis specified decreases.

9. Bit 12 in D1816 (D1896, D1976): Relative/Absolute coordinates
   - Bit 12=0: Absolute coordinates
   - Bit 12=1: Relative coordinates
10. Bit 13 in D1816 (D1896, D1976): Mode of triggering the calculation of the target position
    - Bit 13=0: The calculation of the target position of the axis specified is triggered by a transition in DOG's signal from low to high.
    - Bit 13=1: The calculation of the target position of the axis specified is triggered by a transition in DOG's signal from high to low. (The setting of bit 13 is applicable to the insertion of single-speed motion, and the insertion of two-speed motion.)
11. Bit 14 in D1816 (D1896, D1976): Curve
    - Bit 14=0: Trapezoid curve
    - Bit 14=1: S curve

X-axis		Y-axis		Z-axis		Number of pulses it takes for the motor of the axis specified to rotate once (A)
HW	LW	HW	LW	HW	LW	
D1819	D1818	D1899	D1898	D1979	D1978	

[Description]

1. Owing to the fact that users can set an electronic gear ratio for a servo drive, the number of pulses it takes for a servo motor to rotate once is not necessarily equal to the number of pulses which will be generated after a decoder rotates once. The relation between the number of pulses it takes for a servo drive to rotate once and an electronic gear ratio is described below.

   Number of pulses it takes for a motor to rotate once (A) x Electronic gear ratio (CMX/CDV)=Number of pulses which will be generated after a decoder rotates once
2. The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976). If the unit selected is a mechanical unit or a compound unit, users need to set D1818 (D1898, D1978) and D1819 (D1899, D1979). If the unit selected is a motor unit, users do not need to set D1818 (D1898, D1978) and D1819 (D1899, D1979).

X-axis		Y-axis		Z-axis		Distance generated after the motor of the axis specified rotate once (B)
HW	LW	HW	LW	HW	LW	
D1821	D1820	D1901	D1900	D1981	D1980	

[Description]

1. Three units are available. They are μm/revolution, mdeg/revolution, and 10^{-4} inches/revolution. The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976). The value in (D1821, D1820) ((D1901, D1900), (D1981, D1980)) is in the range of 1 to 2,147,483,647.
2. The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976). If the unit selected is a mechanical unit or a compound unit, users need to set D1820 (D1900, D1980) and D1821 (D1901, D1981). If the unit selected is a motor unit, users do not need to set D1820 (D1900, D1980) and D4821 (D1901, D1981).

X-axis		Y-axis		Z-axis		Maximum speed (V_{MAX}) at which the axis specified rotates
HW	LW	HW	LW	HW	LW	
D1823	D1822	D1903	D1902	D1983	D1982	

[Description]

1. Users can set the maximum speed of motion. The value in (D1823, D1822) ((D1903, D1902), (D1983, D1982)) is in the range of 0 to 2,147,483,647. (The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).)
2. The frequency of pulses generated by motion is in the range of 10 PPS to 500K PPS. If the value in (D1823, D1822) ((D1903, D1902), (D1983, D1982)) is greater than 500K, the frequency of pulses generated will be 500K PPS. If the value in (D1823, D1822) ((D1903, D1902), (D1983, D1982)) is less than 10, the frequency of pulses generated will be 10 PPS.

X-axis		Y-axis		Z-axis		Start-up speed (V_{BIAS}) at which the axis specified rotates
HW	LW	HW	LW	HW	LW	
D1825	D1824	D1905	D1904	D1985	D1984	

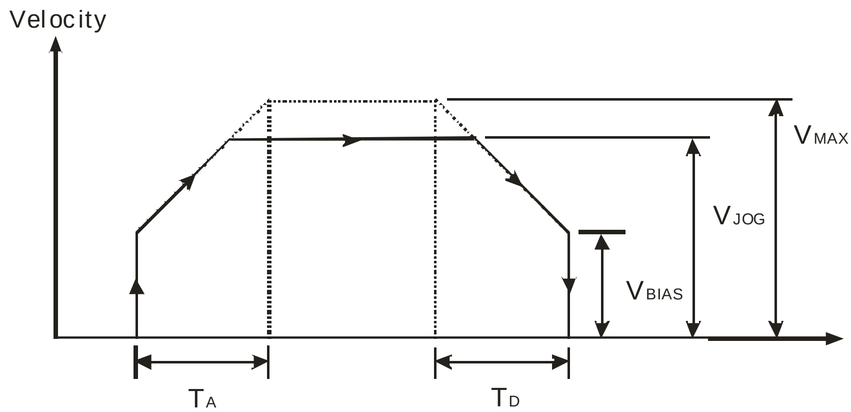
[Description]

1. Users can set the start-up speed of motion. The value in (D1825, D1824) ((D1905, D1904), (D1985, D1984)) is in the range of 0 to 2,147,483,647. (The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).)
2. The frequency of pulses generated by motion is in the range of 0 PPS to 500K PPS. If the value in (D1825, D1824) ((D1905, D1904), (D1985, D1984)) is greater than 500K, the frequency of pulses generated will be 500K PPS. If the value in (D1825, D1824) ((D1905, D1904), (D1985, D1984)) is less than 0, the frequency of pulses generated will be 0 PPS.
3. If a stepper motor system is used, the start-up speed that users set must be greater than the motor resonance frequency generated.

X-axis		Y-axis		Z-axis		JOG speed (V_{JOG}) at which the axis specified rotates
HW	LW	HW	LW	HW	LW	
D1827	D1826	D1907	D1906	D1987	D1986	

[Description]

1. Users can set the JOG speed (V_{JOG}) at which the axis specified rotates. The value in (D1827, D1826) ((D1907, D1906), (D1987, D1986)) is in the range of 0 to 2,147,483,647. (The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).
2. The frequency of pulses generated by motion is in the range of 10 PPS to 500K PPS. If the value in (D1827, D1826) ((D1907, D1906), (D1987, D1986)) is greater than 500K, the frequency of pulses generated will be 500K PPS. If the value in (D1827, D1826) ((D1907, D1906), (D1987, D1986)) is less than 10, the frequency of pulses generated will be 10 PPS.
3. $V_{MAX}>V_{JOG}>V_{BIAS}$
   If the V_{JOG} set is greater than the V_{MAX} set, the actual V_{JOG} will be equal to the V_{MAX}.
   If the V_{JOG} set is less than the V_{BIAS} set, the actual V_{JOG} will be equal to the V_{BIAS}, and an error will occur.
4. When an axis operates, users can not modify the JOG speed of the axis



X-axis		Y-axis		Z-axis		Speed (V_{RT}) at which the axis specified returns home
HW	LW	HW	LW	HW	LW	
D1829	D1828	D1909	D1908	D1989	D1988	

[Description]

1. Users can set the speed at which the axis specified returns home. The value in (D1829, D1828) ((D1909, D1908), (D1989, D1988)) is in the range of 1 to 2,147,483,647. (The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).)
2. The frequency of pulses generated by motion is in the range of 10 PPS to 500K PPS. If the value in (D1829, D1828) ((D1909, D1908), (D1989, D1988)) is greater than 500K, the frequency of pulses generated will be 500K PPS. If the value in (D1829, D1828) ((D1909, D1908), (D1989, D1988)) is less than 10, the frequency of pulses generated will be 10 PPS.

3. $V_{MAX}>V_{RT}>V_{BIAS}$
4. When an axis returns home, the speed at which the axis returns home can not be changed.

X-axis		Y-axis		Z-axis		Speed (V_{CR}) to which the speed of the axis specified decreases when the axis returns home
HW	LW	HW	LW	HW	LW	
D1831	D1830	D1911	D1910	D1991	D1990	

[Description]

1. The value in (D1831, D1830) ((D1911, D1910), (D1991, D1990)) is in the range of 1 to 2,147,483,647. (The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).
2. The frequency of pulses generated by motion is in the range of 10 PPS to 500K PPS. If the value in (D1831, D1830) ((D1911, D1910), (D1991, D1990)) is greater than 500K, the frequency of pulses generated will be 500K PPS. If the value in (D1831, D1830) ((D1911, D1910), (D1991, D1990)) is less than 10, the frequency of pulses generated will be 10 PPS.
3. When motion of returning home is executed, the speed of the motor used is the V_{RT} set. When there is a transition in DOG's signal from low to high or from high to low, the speed of the motor used decreases to the V_{CR} set.
4. In order for the axis specified to returns home precisely, it is suggested that the V_{CR} set should be a low speed.
5. When the motion of returning home is executed, the V_{CR} set can not be changed.

X-axis		Y-axis		Z-axis		Number of PG0 pulses for the axis specified
HW	LW	HW	LW	HW	LW	
-	D1832	-	D1912	-	D1992	

[Description]

1. The value in D1832 (D1912, D1992) is in the range of -32,768 to 32,767. If the value in D1832 (D1912, D1992) is a positive value, the axis specified will move in the direction in which it returns home. If the value in D1832 (D1912, D1992) is a negative value, the axis specified will move in the direction which is opposite to the direction in which it returns home.
2. Please refer to the descriptions of bit 9 and bit10 in D1816 (D1896, D1976) for more information about decelerating and stopping the motor used.

X-axis		Y-axis		Z-axis		Supplementary pulses for the axis specified
HW	LW	HW	LW	HW	LW	
-	D1833	-	D1913	-	D1993	

[Description]

1. The value in D1833 (D1913, DD1993) is in the range of -32,768 to 32,767. If the value in D1833 (D1913, DD1993) is a positive value, the axis specified will move in the direction in which it returns home. If the value in D1833 (D1913, DD1993) is a negative value, the axis specified will move in the direction which is opposite to the direction in which it returns home.
2. Please refer to the descriptions of bit 9 and bit10 in D1816 (D1896, D1976) for more information about decelerating and stopping the motor used.

X-axis		Y-axis		Z-axis		Home position of the axis specified
HW	LW	HW	LW	HW	LW	
D1835	D1834	D1915	D1914	D1995	D1994	

[Description]

1. The value in (D1835, D1834) ((D1915, D1914), (D1995, D1994)) is in the range of 0 to ±999,999. (The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).
2. After the axis specified returns home, the value in (D1835, D1834) ((D1915, D1914), (D1995, D1994)) will be written into (D1849, D1848) ((D1929, D1928), (D2009, D2008)).

X-axis		Y-axis		Z-axis		Time (T_{ACC}) it takes for the axis specified to accelerate
HW	LW	HW	LW	HW	LW	
-	D1836	-	D1916	-	D1996	

[Description]

1. Users can set the times it takes for the speed of the axis specified to increase from its start-up speed to its maximum speed. The value in D1836 (D1916, D1996) is in the range of 0 to 32,767. A millisecond is a unit.
2. If the value in D1836 (D1916, D1996) is less than 10, it will be counted as 10. If the value in D1836 (D1916, D1996) is greater than 32,767, it will be counted as 32,767.
3. If users want to have a complete S curve, the maximum speed which is set must be the same as the speed at which the axis specified operates.

X-axis		Y-axis		Z-axis		Time (T_{DEC}) it takes for the axis specified to decelerate
HW	LW	HW	LW	HW	LW	
	D1837		D1917		D1997	

[Description]

1. Users can set the times it takes for the speed of the axis specified to decrease from its maximum speed to its start-up speed. The value in D1837 (D1917, D1997) is in the range of 0 to 32,767. A millisecond is a unit.
2. If the value in D1837 (D1917, D1997) is less than 10, it will be counted as 10. If the value in D1837 (D1917, D1997) is greater than 32,767, it will be counted as 32,767.
3. If users want to have a complete S curve, the maximum speed which is set must be the same as the speed at which the axis specified operates.

X-axis		Y-axis		Z-axis		Target position of the axis specified (P (I))
HW	LW	HW	LW	HW	LW	
D1839	D1838	D1919	D1918	D1999	D1998	

[Description]

1. The value in (D1839, D1838) ((D1919, D1918), (D1999, D1998)) is in the range of -2,147,483,648 to +2,147,483,647. (The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).)
2. Target position (P (I))
   - Absolute coordinates: Bit 12 in D1816 (D1896, D1976) is 0.

     The target position of the axis specified indicates a distance from 0. If the target position of an axis is greater than its present command position, the motor used will rotate clockwise. If the target position of an axis is less than its present command position, the motor used will rotate counterclockwise.
   - Relative coordinates: Bit 12 in D1816 (D1896, D1976) is 1.

     The target position of an axis indicates a distance from its present command position. If the target position specified is a positive value, the motor used will rotate clockwise. If the target position specified is a negative value, the motor used will rotate counterclockwise.
3. The ratio used is determined by bit 2 and bit 3 in D1816 (D1896, D1976).

X-axis		Y-axis		Z-axis		Speed at which the axis specified rotates (V (I))
HW	LW	HW	LW	HW	LW	
D1841	D1840	D1921	D1920	D2001	D2000	

[Description]

1. The value in (D1841, D1840) ((D1921, D1920), (D2001, D2000)) is in the range of -2,147,483,647 to +2,147,483,647. (The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).)
2. The frequency of pulses generated by motion is in the range of 10 PPS to 500K PPS. If the value in (D1841, D1840) ((D1921, D1920), (D2001, D2000)) is greater than 500K, the frequency of pulses generated will be 500K PPS. If the value in (D1841, D1840) ((D1921, D1920), (D2001, D2000)) is less than 10, the frequency of pulses generated will be 10 PPS.

3. $V_{MAX} > V (I) > V_{BIAS}$
4. When bit 4 in D1846 (D1926, D2006) is ON, the speed at which the axis specified rotates (V (I)) can be changed. If the Speed at which the axis specified rotates is a positive value, the motor used will rotate clockwise. If the Speed at which the axis specified rotates is a negative value, the motor used will rotate counterclockwise.

X-axis		Y-axis		Z-axis		Target position of the axis specified (P (II))
HW	LW	HW	LW	HW	LW	
D1843	D1842	D1923	D1922	D2003	D2002	

[Description]

1. The value in (D1843, D1842) ((D1923, D1922), (D2003, D2002)) is in the range of -2,147,483,648 to +2,147,483,647. (The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).)
2. Target position (P (II))
   - Absolute coordinates: Bit 12 in (D1816 (D1896, D1976) is 0.
     The target position of the axis specified indicates a distance from 0. If the target position of an axis is greater than its present command position, the motor used will rotate clockwise. If the target position of an axis is less than its present command position, the motor used will rotate counterclockwise.
   - Relative coordinates: Bit 12 in (D1816 (D1896, D1976, D2056, D2136, D2216) is 1.
     The target position of an axis indicates a distance from its present command position. If the target position specified is a positive value, the motor used will rotate clockwise. If the target position specified is a negative value, the motor used will rotate counterclockwise.
3. The ratio used is determined by bit 2 and bit 3 in D1816 (D1896, D1976).

X-axis		Y-axis		Z-axis		Speed at which the axis specified rotates(V (II))
HW	LW	HW	LW	HW	LW	
D1845	D1844	D1925	D1924	D2005	D2004	

[Description]

1. The value in (D1845, D1844) ((D1925, D1924), (D2005, D2004)) is in the range of 0 to 2,147,483,647. (The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).)
2. The frequency of pulses generated by motion is in the range of 10 PPS to 500K PPS. If the value in (D1845, D1844) ((D1925, D1924), (D2005, D2004)) is greater than 500K, the frequency of pulses generated will be 500K PPS. If the value in (D1845, D1844) ((D1925, D1924), (D2005, D2004)) is less than 10, the frequency of pulses generated will be 10 PPS.
3. $V_{MAX} > V (II) > V_{BIAS}$

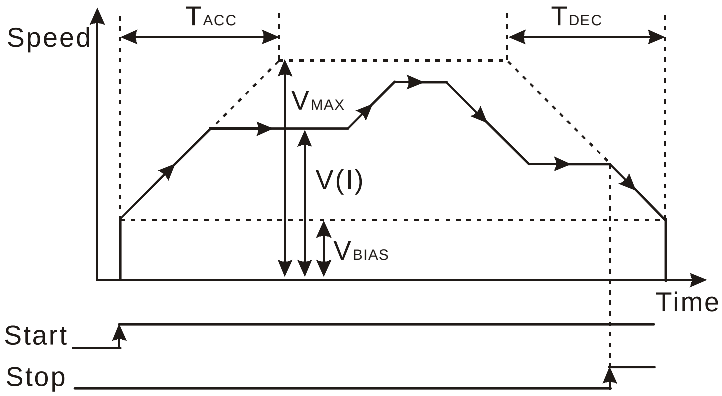
X-axis		Y-axis		Z-axis		Operation command
HW	LW	HW	LW	HW	LW	
-	D1846	-	D1926	-	D2006	

[Description]

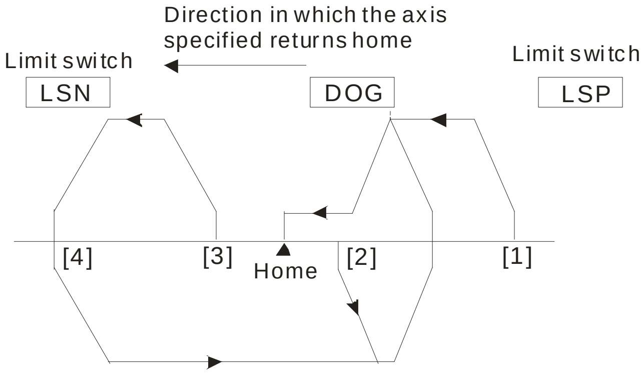
1. Bit 0 in D1846 (D1926, D2006): The motion of the axis specified is stopped by software.
   - The motion of the axis specified is stopped by software when bit 0 in D1846 (D1926, D2006) is turned from OFF to ON.
   - The function of bit 0 in D1846 (D1926, D2006) is the same as the function of STOP. Bit 0 in D1846 (D1926, D2006) and STOP can be used to decelerate and stop the motion controller used.
2. Bit 1 in D1846 (D1926, D2006): The motion of the axis specified is started by software.
   - The motion of the axis specified is started by software when bit 1 in D1846 (D1926, D2006) is turned from OFF to ON.
3. Bit 2 in D1846 (D1926, D2006): The axis specified operates in a JOG+ mode.
   - When bit 2 in D1846 (D1926, D2006), clockwise pulses are generated at the JOG speed set.
4. Bit 3 in D1846 (D1926, D2006): The axis specified operates in a JOG- mode.

- When bit 3 in D1846 (D1926, D2006) is ON, counterclockwise pulses are generated at the JOG speed set.

5. Bit 4 in D1846 (D1926, D2006): A mode of variable motion is activated.
   - After bit 4 in D1846 (D1926, D2006) is set to 1, the DVP-20PM series motion controller will execute variable motion, and it will send pulses by a pulse generator.
   - After a mode of variable motion is activated, the V_{BIAS} of the axis specified will increase to its V (I). When the axis operates, users can change its V (I) at will. The DVP-20PM series motion controller accelerates or decelerates according to the V (I) set.
   - Users can stop variable motion by setting bit 0 in D1846 (D1926, D2006,) to 1, or by setting bit 4 in D1846 (D1926, D2006) to 0.
   - Diagram



6. Bit 5 in D1846 (D1926, D2006): A manual pulse generator is operated.
   - If bit 5 in D1846 (D1926, D2006) is set to ON, a manual pulse generator mode will be activated. Please refer to the descriptions of D1858~D1864 (D1938~D1944, D2018~D2024) for more information.
7. Bit 6 in D1846 (D1926, D2006): A mode of triggering the return to home is activated.
   - When bit 6 in D1846 (D1926, D20066) is turned from OFF to ON, a mode of triggering the return to home is activated. The mode of triggering the return to home varies with the present command position of the axis specified. There are two situations.



Position (1): Position [1] is at the right side of the home and DOG, and DOG is OFF.

Position (2): Position [2] is at the right side of the home, and DOG is ON.

Position (3)*: Position [3] is at the left side of the home and DOG. DOG is OFF, and LSN is OFF.

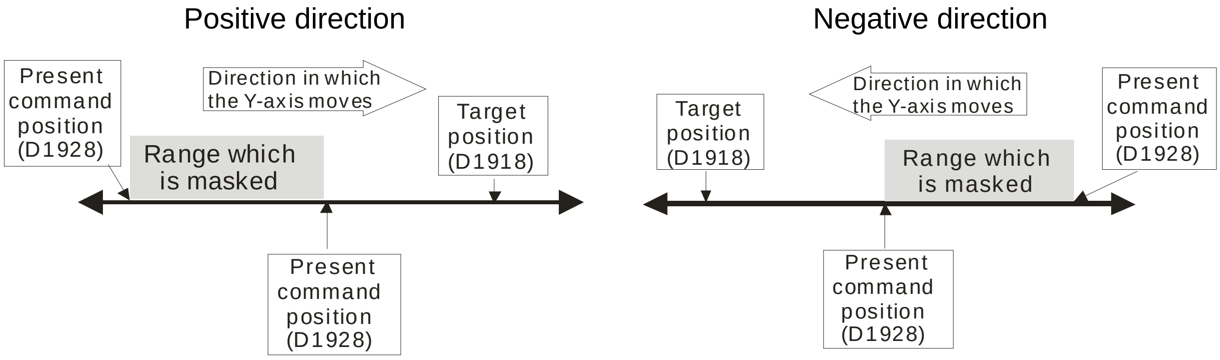
Position (4)*: Position [4] is at the left side of the home and DOG. DOG is OFF, and LSN is ON.

*: A DVP-10PM series motion controller does not have LSN and LSP, and therefore does not support position (3) and position (4).

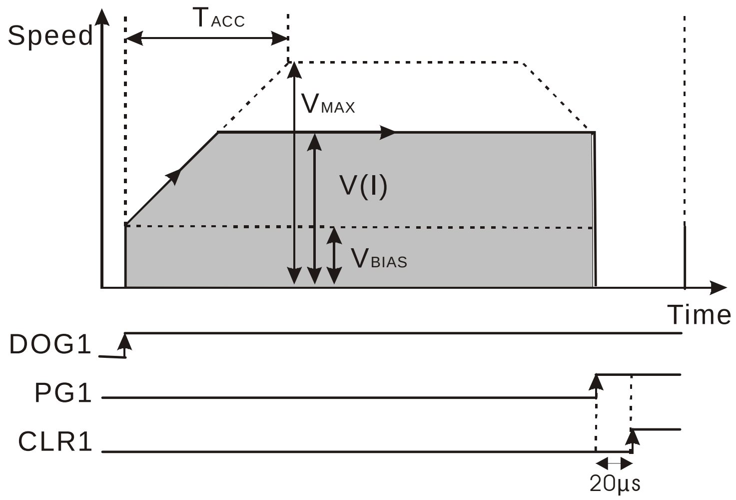
8. Bit 7 in D1926: Triggering single-speed motion by means of an external signal
   - If bit 7 in D1926 is set to ON, single-speed motion can be triggered by an external signal. After DOG1's signal goes from low to high or from high to low, the single-speed motion will be executed. If PG1 is turned ON when the Y-axis moves to the range which is not masked, the motion of the Y-axis will stop immediately, and CLR1 will be ON twenty microseconds after PG1's signal goes from low to high or from high to low. The speed of the single-speed motion depend on the V (I)

which are set by users, and the DVP-20PM series motion controller sends pulses by a pulse generator.

- The range which is masked is shown below. It depends on the present command position of the Y-axis and the target position of the Y-axis. If the present command position of the Y-axis is less than its target position, t the motor used will rotate clockwise. If the present command position of the Y-axis is greater than its target position, the motor used will rotate counterclockwise, and the Y-axis leaves the range which is masked when the value in D1928 is 0.

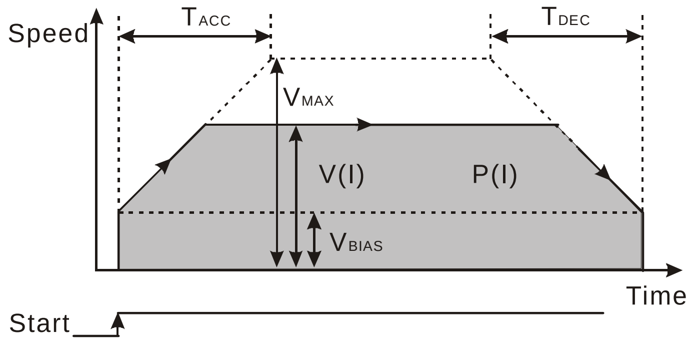


- If relative single-speed motion is activated, the sign bit of the P (I) set by users will determine the direction of the relative single-speed motion.
- Absolute single-speed motion: If the target position of the axis specified is greater than its present command position, the motor used will rotate clockwise. If the target position of the axis specified is less than its present command position, the motor used will rotate counterclockwise.
- The speed of the Y-axis increases from the V_{BIAS} set to the V (I) set. After PG1 is turned ON, the Y-axis will stop outputting pulses.
- V_{BIAS}: D1824 (D1904, D1984); V (I): D1840 (D1920, D2000); V_{MAX}: D1822 (D1902, D1982); T_{ACC}: D1836 (D1916, D1996)



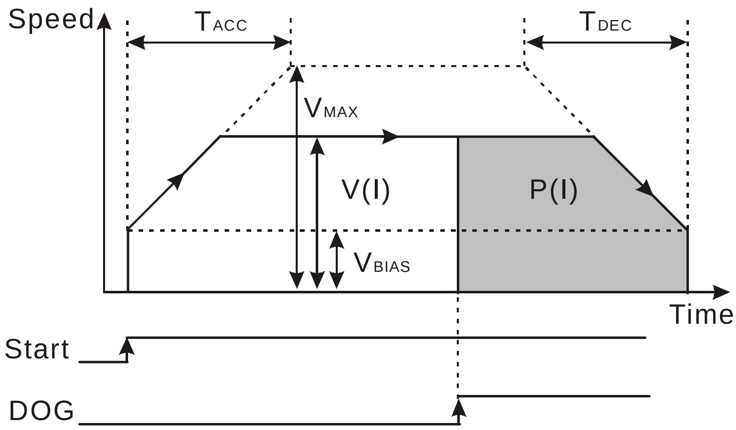
9. Bit 8 in D1846 (D1926, D2006): A mode of single-speed motion is activated.
   - After bit 8 in D1846 (D1926, D2006) is set to 1, a mode of single-speed motion will be activated. The target position of the single-speed motion and the speed of the single-speed motion depend on the P (I) and the V (I) which are set by users, and the DVP-20PM series motion controller sends pulses by a pulse generator.
   - If relative single-speed motion is activated, the sign bit of the P (I) set by users will determine the direction of the relative single-speed motion.
   - Absolute single-speed motion: If the target position of the axis specified is greater than its present command position, the motor used will rotate clockwise. If the target position of the axis specified is less than its present command position, the motor used will rotate counterclockwise.
   - After single-speed motion is activated, the speed of the absolute single-speed motion will increase from the V_{BIAS} set to the V (I) set. The speed of the absolute single-speed motion will not decrease from the V (I) set to the V_{BIAS} set until the number of pulses output is near the P (I) set.

- V_{BIAS}: D1824 (D1904, D1984); V (I): D1840 (D1920, D2000); V_{MAX}: D1822 (D1902, D1982); P (I): D1838 (D1918, D1998); T_{ACC}: D1836 (D1916, D1996); T_{DEC}: D1837 (D1917, D1997)



- If bit 6 in D1816 (D1896, D1976) is ON, and bit 8 in D1846 (D1926, D2006) is ON, Y0~Y3 will output unidirectional pulses.

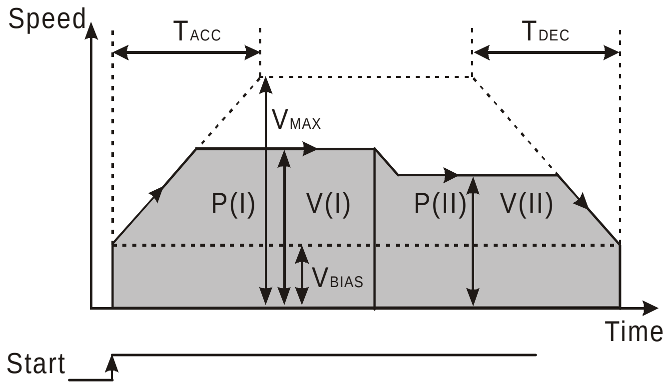
10. Bit 9 in D1846 (D1926, D2006): A mode of inserting single-speed motion is activated.
    - After bit 9 in D1846 (D1926, D2006) is set to 1, a mode of inserting single-speed motion will be activated, and the DVP-20PM series motion controller will send pulses by a pulse generator. After DOG's signal goes from low to high or from high to low, the axis specified will move to the target position indicated by the P (I) set.
    - If relative single-speed motion is activated, the sign bit of the P (I) set by users will determine the direction of the relative single-speed motion.
    - Absolute single-speed motion: If the target position of the axis specified is greater than its present command position, the motor used will rotate clockwise. If the target position of the axis specified is less than its present command position, the motor used will rotate counterclockwise.
    - The speed of motion will increase from the V_{BIAS} set to the V (I) set. After DOG's signal goes from low to high or from high to low, the DVP-20PM series motion controller used will continue sending pulses. The speed of the motion will not decrease from the V (I) set to the V_{BIAS} set until the number of pulses output is near the P (I) set.
    - V_{BIAS}: D1824 (D1904, D1984); V (I): D1840 (D1920, D2000); V_{MAX}: D1822 (D1902, D1982); P (I): D1838 (D1918, D1998); T_{ACC}: D1836 (D1916, D1996); T_{DEC}: D1837 (D1917, D1997)



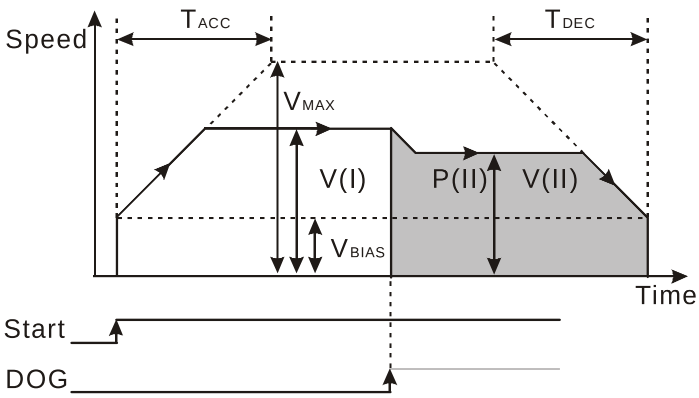
11. Bit 10 in D1846 (D1926, D2006): A mode of two-speed motion is activated.
    - After bit 10 in D1846 (D1926, D2006) is set to 1, a mode of two-speed motion will be activated. The axis specified moves at the V (I) set. After it moves to the P (I) set, it will move to the P (II) set at the V (II) set.
    - Relative coordinates: The sign bit of the P (I) set by users determines the direction of motion. If the P (I) specified is a positive value, the motor used will rotate clockwise. If the P (I) specified is a negative value, the motor used will rotate counterclockwise.
    - Absolute coordinates: If the target position (P (I)) of an axis is greater than its present command position, the motor used will rotate clockwise. If the target position (P (I)) of an axis is less than its present command position, the motor used will rotate counterclockwise.
    - After motion is started, the speed of the motion will increase from the V_{BIAS} set to the V (I) set. The speed of the motion will not increase/decrease from the V (I) set to the V (II) set until the number of pulses output is near the P (I) set. The speed of the motion will not decrease from the V (II) to

the V_{BIAS} set until the present command position of the axis specified is near the P (II) set.

- V_{BIAS}: D1824 (D1904, D1984); V (I): D1840 (D1920, D2000); V (II): D1844 (D1924, D2004,); V_{MAX}: D1822 (D1902, D1982); P (I): D1838 (D1918, D1998); P (II): D1842 (D1922, D2002); T_{ACC}: D1836 (D1916, D1996); T_{DEC}: D1837 (D1917, D1997)



12. Bit 11 in D1846 (D1926, D2006): A mode of inserting two-speed motion is activated.
    - After bit 11 in D1846 (D1926, D2006) is set to 1, a mode of inserting two-speed motion will be activated. The axis specified moves at the V (I) set. After DOG's signal goes from low to high or from high to low, the axis will move to the target position indicated by the P (II) set at the V (II) set.
    - Relative coordinates: The sign bit of the P (I) set by users determines the direction of motion.
    - Absolute coordinates: If the target position (P (I)) of an axis is greater than its present command position, the motor used will rotate clockwise. If the target position (P (I)) of an axis is less than its present command position, the motor used will rotate counterclockwise.
    - After motion is started, the speed of the motion will increase from the V_{BIAS} set to the V (I) set. After DOG's signal goes from low to high or from high to low, the speed of the motion will increase/decrease from the V (I) set to the V (II) set. If there is a transition in STOP's signal from low to high or from high to low when the axis specified moves at the V (II) set, the axis will stop outputting pulses.
    - V_{BIAS}: D1824 (D1904, D1984); V (I): D1840 (D1920, D2000); V (II): D1844 (D1924, D2004); V_{MAX}: D1822 (D1902, D1982); P (I): D1838 (D1918, D1998); P (II): D1842 (D1922, D2002); T_{ACC}: D1836 (D1916, D1996); T_{DEC}: D1837 (D1917, D1997)



13. Bit 12 inD1846 (D1926, D2006): The execution of the Ox motion subroutine set starts.
    - Bit 12=1: The execution of the Ox motion subroutine set starts.
    - Bit 12=0: The execution of the Ox motion subroutine set stops.

X-axis		Y-axis		Z-axis		Mode of operation
HW	LW	HW	LW	HW	LW	
-	D1847	-	D1927	-	D2007	

[Description]

1. Bit 2 in D1847 (D1927, D2007): Mode of sending a CLR signal
    - Bit 2=0: After the axis specified returns home, the CLR output will send a 130 millisecond signal to the servo drive used, and the present position of the servo drive which is stored in a register in the

servo drive will be cleared.
- Bit 2=1: The CLR output functions as a general output. Its state is determined by bit 3 in D1847 (D1927, D2007).

2. Bit 3 in D1847 (D1927, D2007): Setting the CLR output to ON/OFF
   - Bit 3=0: The CLR output is OFF.
   - Bit 3=1: The CLR output is ON.
3. Bit 4 in D1847 (D1927, D2007): Setting the polarity of the CLR output
   - Bit 4=0: The CLR output is a Form A contact.
   - Bit 4=1: The CLR output is a Form B contact.
4. Bit 5 in D1847 (D1927, D2007): Mode of stopping the motor used when STOP is ON.
   - Bit 5=0: If STOP is ON when the motor used operates, the motor will decelerate and stop. If a motion command is sent, the motor will ignore the previous unfinished distance, and rotate for the distance specified.
   - Bit 5=1: If STOP is ON when the motor used operates, the motor will decelerate and stop. If a motion command is sent, the motor will complete the previous unfinished distance, and then rotate for the distance specified.
5. Bit 6 in D1847 (D1927, D2007,): Limitation on the present position of the slave axis controlled by the manual pulse generator used
   - Bit 6=0: There is no limitation on the present position of the slave axis controlled by the manual pulse generator used.
   - Bit 6=1: The present position of the slave axis controlled by the manual pulse generator used has to be in the range of the P (I) set to the P (II) set. If the present position of the slave axis controlled by the manual pulse generator used is not in the range of the P (I) set to the P (II) set, the slave axis will decelerate and stop.
6. Bit 7 in D1847 (D1927, D2007): Mode of stopping the motor used when the motor used comes into contact with a positive limit switch/negative limit switch
   - Bit 7=0: If the motor used comes into contact with a positive limit switch/negative limit switch when it rotates, it will decelerate and stop.
   - Bit 7=1: If the motor used comes into contact with a positive limit switch/negative limit switch when it rotates, it will stop immediately.
7. Bit 8~bit 10 in D1847 (D1927, D2007): Setting a stop signal
   - When single-speed motion, two-speed motion, inserting single-speed motion, or inserting two-speed motion is executed, A0 or B0 can be used to stop the motion.
   - Bit 10~bit 8=K0 (000) or other values: No function
     Bit 10~bit 8=K1 (001): Motion is stopped immediately when there is a transition in A0's signal from low to high.
     Bit 10~bit 8=K2 (010): Motion is stopped immediately when there is a transition in A0's signal from high to low.
     Bit 10~bit 8=K3 (011): Motion is stopped immediately when there is a transition in B0's signal from low to high.
     Bit 10~bit 8=K4 (100): Motion is stopped immediately when there is a transition in B0's signal from high to low.
8. Bit 15 in D1847 (D1927, D2007): Restoring the DVP-20PM series motion controller to the factory settings
   - Bit 15=1: The values of parameters are restored to factory settings.

X-axis		Y-axis		Z-axis		Present command position of the axis specified (Pulse)
HW	LW	HW	LW	HW	LW	
D1849	D1848	D1929	D1928	D2009	D2008	

[Description]
1. The value in (D1849, D1848) ((D1929, D1928), (D2009, D2008)) is in the range of -2,147,483,648 to +2,147,483,647.
2. The present command position of the axis specified is indicated by the number of pulses. The unit used

is determined by bit 0 and bit 1 in D1816 (D1896, D1976). After the axis specified returns home, the value in (D1835, D1834) ((D1915, D1914), (D1995, D1994)) will be written into (D1849, D1848) ((D1929, D1928), (D2009, D2008)).

X-axis		Y-axis		Z-axis		Present command speed of the axis specified (PPS)
HW	LW	HW	LW	HW	LW	
D1851	D1850	D1931	D1930	D2011	D2010	

[Description]

1. The value in (D1851, D1850) ((D1931, D1930), (D2011, D2010)) is in the range of 0 to 2,147,483,647.
2. The present command speed of the axis specified is indicated by the number of pulses.

X-axis		Y-axis		Z-axis		Present command position of the axis specified (Unit)
HW	LW	HW	LW	HW	LW	
D1853	D1852	D1933	D1932	D2013	D2012	

[Description]

1. The value in (D1853, D1852) ((D1933, D1932), (D2013, D2012)) is in the range of -2,147,483,648 to +2,147,483,647.
2. The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976). After the axis specified returns home, the value in (D1835, D1834) ((D1915, D1914), (D1995, D1994)) will be written into (D1853, D1852) ((D1933, D1932), (D2013, D2012)).

X-axis		Y-axis		Z-axis		Present command speed of the axis specified (Unit)
HW	LW	HW	LW	HW	LW	
D1855	D1854	D1935	D1934	D2015	D2014	

[Description]

1. The value in (D1855, D1854) ((D1935, D1934), (D2015, D2014)) is in the range of 0 to 2,147,483,647.
2. The unit used is determined by bit 0 and bit 1 in D1816 (D1896, D1976).

X-axis		Y-axis		Z-axis		State of the axis specified
HW	LW	HW	LW	HW	LW	
-	D1856	-	D1936	-	D2016	

[Description]

Bit#	D1856 (D1936, D2016)
0	Positive-going pulses are being output.
1	Negative-going pulses are being output.
2	The axis specified is operating.
3	An error occurs.
4	The axis specified pauses.
5	The manual pulse generator used generates positive-going pulses.
6	The manual pulse generator used generates negative-going pulses.
7	-

X-axis		Y-axis		Z-axis		Axis error code
HW	LW	HW	LW	HW	LW	
-	D1857	-	D1937	-	D2017	

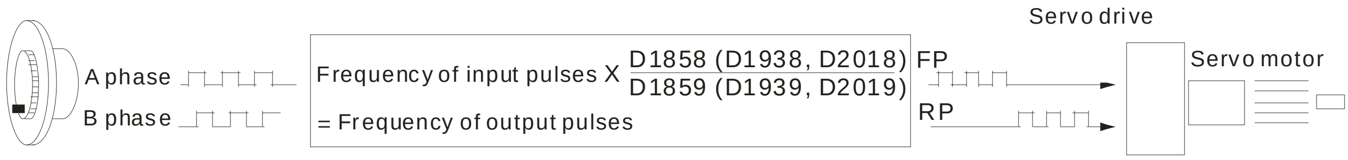
[Description]

Please refer to chapter 15 for more information.

X-axis		Y-axis		Z-axis		Electronic gear ratio
HW	LW	HW	LW	HW	LW	
-	D1858	-	D1938	-	D2018	Electronic gear ratio (Numerator)
-	D1859	-	D1939	-	D2019	Electronic gear ratio (Denominator)

[Description]

1. If bit 5 in D1846 (D1926, D2006) is set to ON, a manual pulse generator mode will be activated.
2. A manual pulse generator generates A/B-phase pulses that are sent to the input terminals A0±/B0± (A1±/B1±). The relation between the position of the axis specified and the input pulses generated by the manual pulses used is shown below.



   If a positive limit switch or a negative limit switch is enabled when a manual pulse generator is operated, the generation of pulses will stop. If a positive limit switch is enabled, positive-going pulses will be inhibited, and negative-going will be allowed. If a negative limit switch is enabled, negative-going pulses will be inhibited, and positive-going switch will be allowed.
3. The speed output is determined by the frequency of input pulses generated by a manual pulse generator and an electronic gear ratio.

X-axis		Y-axis		Z-axis		Frequency of pulses generated by the manual pulse generator for the axis specified
HW	LW	HW	LW	HW	LW	
D1861	D1860	D1941	D1940	D2021	D2020	

[Description]

1. The value in (D1861, D1860) ((D1941, D1940), (D2021, D2020)) indicates the frequency of pulses generated by the manual pulse generator for the axis specified. It does not vary with the values in D1858 (D1938, D2018) and D1859 (D1939, D2019).

X-axis		Y-axis		Z-axis		Number of pulses generated by the manual pulse generator for the axis specified
HW	LW	HW	LW	HW	LW	
D1863	D1862	D1943	D1942	D2023	D2022	

[Description]

1. The value in (D1863, D1862) ((D1943, D1942), (D2023, D2022)) indicates the number of pulses generated by the manual pulse generator for the axis specified. If the pulses generated by the manual pulse generator for the axis specified are clockwise pulses, the value in (D1863, D1862) ((D1943, D1942), (D2023, D2022)) will increase. If the pulses generated by the manual pulse generator for the axis specified are counterclockwise pulses, the value in (D1863, D1862) ((D1943, D1942), (D2023, D2022)) will decrease.
2. The value in (D1863, D1862) ((D1943, D1942), (D2023, D2022)) does not vary with the values in D1858 (D1938, D2018) and D1859 (D1939, D2019).

X-axis		Y-axis		Z-axis		Response speed of the manual pulse generator for the axis specified
HW	LW	HW	LW	HW	LW	
-	D1864	-	D1944	-	D2024	

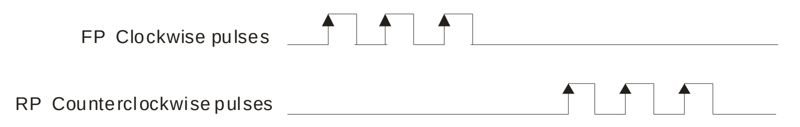
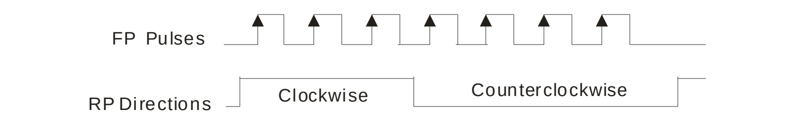
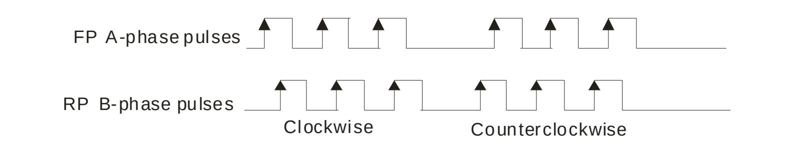
[Description]

1. If the response speed set is high, the pulses output happen almost at the same time as the pulses input by the manual pulse generator used.
2. If the response speed set is low, the pulses output follows the pulses input by the manual pulse generator used.

Setting value	Response speed
≧5	4 ms (Initial value)

Setting value	Response speed
4	32 ms
3	108 ms
2	256 ms
1 or 0	500 ms

3. Bit 8 and bit 9 in D1864 (D1944, D2024): Setting the input pulses generated by the manual pulse generator specified

<table>
<tr><th>b9</th><th>b8</th><th>Input type (positive logic)</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td></td><td>Counting up/down</td></tr>
<tr><td>0</td><td>1</td><td></td><td>Pulses+Directions</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2"></td><td>A/B-phase pulses</td></tr>
<tr><td>1</td><td>1</td><td>Four times the frequency of A/B-phase pulses</td></tr>
</table>

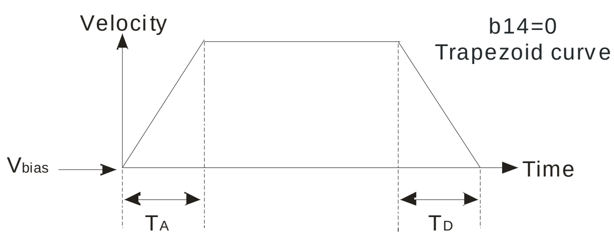
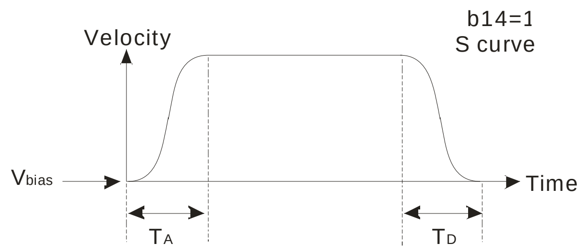
3.12.2 Introduction of Modes of Motion

1. There are ten modes of motions.
   1. Returning home
   2. JOG motion
   3. Single-speed motion
   4. Inserting single-speed motion
   5. Triggering single-speed motion by means of an external signal*
   6. Two-speed motion
   7. Inserting two-speed motion
   8. Variable motion
   9. Manual pulse generator mode
   10. Starting a cyclic/noncyclic electronic cam*
2. If more than one mode of motion is activated, they will be executed in particular order.
   1. Stopping the motion of the axis specified by software
   2. Returning home
   3. Positive JOG motion
   4. Negative JOG motion
   5. Manual pulse generator mode
   6. Variable motion
   7. Single-speed motion
   8. Inserting single-speed motion
   9. Triggering single-speed motion by means of an external signal*
   10. Two-speed motion
   11. Inserting two-speed motion
   12. Starting a cyclic/noncyclic electronic cam*

   *: Only DVP-20PM series motion controllers support this mode of motion.

   If a mode of motion is activated when another mode of motion is executed, the DVP-20PM series motion controller will continue executing the original mode.

3. There are two types of acceleration curves.

1. Trapezoid curve	2. S curve
	

3.12.3 Special Data Registers for Motion Axes

Special data registers for motion axes						Parameter	Mode of operation								
X-axis		Y-axis		Z-axis			JOG motion	Returning home	Single-speed motion	Inserting single-speed motion	Triggering single-speed motion by means of an external signal	Two-speed motion	Inserting two-speed motion	Variable motion	Manual pulse generator mode
HW	LW	HW	LW	HW	LW										
D1819	D1818	D1899	D1898	D1979	D1978	Number of pulses it takes for the motor of the axis specified to rotate once (A)	If the unit used is a motor unit, the special data registers do not need to be set. If the unit used is a mechanical unit or a compound unit, the special data registers need to be set.								
D1821	D1820	D1901	D1900	D1981	D1980	Distance generated after the motor of the axis specified rotate once (B)									
-	D1816	-	D1896	-	D1976	Setting the parameters of the axis specified	◎	◎	◎	◎	◎	◎	◎	◎	◎
D1823	D1822	D1903	D1902	D1983	D1982	Maximum speed (V_{MAX}) at which the axis specified rotates	◎	◎	◎	◎	◎	◎	◎	◎	◎
D1825	D1824	D1905	D1904	D1985	D1984	Start-up speed (V_{BIAS}) at which the axis specified rotates	◎	◎	◎	◎	◎	◎	◎	◎	◎
D1827	D1826	D1907	D1906	D1987	D1986	JOG speed (V_{JOG}) at which the axis specified rotates	◎	-	-	-	-	-	-	-	-

Special data registers for motion axes						Parameter	Mode of operation								
X-axis		Y-axis		Z-axis			JOG motion	Returning home	Single-speed motion	Inserting single-speed motion	Triggering single-speed motion by means of an external signal	Two-speed motion	Inserting two-speed motion	Variable motion	Manual pulse generator mode
HW	LW	HW	LW	HW	LW										
D1829	D1828	D1909	D1908	D1989	D1988	Speed (V_{RT}) at which the axis specified returns home	-	◎	-	-	-	-	-	-	-
D1831	D1830	D1911	D1910	D1991	D1990	Speed (V_{CR}) to which the speed of the axis specified decreases when the axis returns home	-	◎	-	-	-	-	-	-	-
-	D1832	-	D1912	-	D1992	Number of PG0 pulses for the axis specified	-	◎	-	-	-	-	-	-	-
-	D1833	-	D1913	-	D1993	Supplementary pulses for the axis specified	-	◎	-	-	-	-	-	-	-
D1835	D1834	D1915	D1914	D1995	D1994	Home position of the axis specified	-	◎	-	-	-	-	-	-	-
-	D1836	-	D1916	-	D1996	Time (T_{ACC}) it takes for the axis specified to accelerate	◎	◎	◎	◎	◎	◎	◎	◎	-
-	D1837	-	D1917	-	D1997	Time (T_{DEC}) it takes for the axis specified to decelerate	◎	◎	◎	◎	◎	◎	◎	◎	-
D1839	D1838	D1919	D1918	D1999	D1998	Target position of the axis specified (P (I))	-	-	◎	◎	◎	◎	◎	-	◎
D1841	D1840	D1921	D1920	D2001	D2000	Speed at which the axis specified rotates (V (I))	-	-	◎	◎	◎	◎	◎	◎	-
D1843	D1842	D1923	D1922	D2003	D2002	Target position of the axis specified (P (II))	-	-	-	-	-	◎	◎	-	◎
D1845	D1844	D1925	D1924	D2005	D2004	Speed at which the axis specified rotates (V (II))	-	-	-	-	-	◎	◎	-	-
-	D1846	-	D1926	-	D2006	Operation command	◎	◎	◎	◎	◎	◎	◎	◎	◎
-	D1847	-	D1927	-	D2007	Mode of operation	◎	◎	◎	◎	◎	◎	◎	◎	◎
D1849	D1848	D1929	D1928	D2009	D2008	Present command position of the axis specified (Pulse)	◎	◎	◎	◎	◎	◎	◎	◎	◎
D1851	D1850	D1931	D1930	D2011	D2010	Present command speed of the axis specified (PPS)	◎	◎	◎	◎	◎	◎	◎	◎	◎
D1853	D1852	D1933	D1932	D2013	D2012	Present command position of the axis specified (Unit)	◎	◎	◎	◎	◎	◎	◎	◎	◎
D1855	D1854	D1935	D1934	D2015	D2014	Present command speed of the axis specified (Unit)	◎	◎	◎	◎	◎	◎	◎	◎	◎
-	D1858	-	D1938	-	D2018	Electronic gear ratio of the axis specified (Numerator)	-	-	-	-	-	-	-	-	◎

Special data registers for motion axes						Parameter	Mode of operation								
X-axis		Y-axis		Z-axis			JOG motion	Returning home	Single-speed motion	Inserting single-speed motion	Triggering single-speed motion by means of an external signal	Two-speed motion	Inserting two-speed motion	Variable motion	Manual pulse generator mode
HW	LW	HW	LW	HW	LW										
-	D1859	-	D1939	-	D2019	Electronic gear ratio of the axis specified (Denominator)	-	-	-	-	-	-	-	-	◎
D1861	D1860	D1941	D1940	D2021	D2020	Frequency of pulses generated by the manual pulse generator for the axis specified	-	-	-	-	-	-	-	-	◎
D1863	D1862	D1943	D1942	D2023	D2022	Number of pulses generated by the manual pulse generator for the axis specified	-	-	-	-	-	-	-	-	◎
-	D1864	-	D1944	-	D2024	Response speed of the manual pulse generator for the axis specified	-	-	-	-	-	-	-	-	◎
D1865	-	-	-	-	-	Mode of stopping Ox0~Ox99	-	-	-	-	-	-	-	-	-
D1867	D1866	D1947	D1946	D2027	D2026	Electrical zero of the axis specified	-	-	-	-	-	-	-	-	-
D1868	-	-	-	-	-	Setting an Ox motion subroutine number	-	-	-	-	-	-	-	-	-
D1869	-	-	-	-	-	Step address in the Ox motion subroutine at which an error occurs	-	-	-	-	-	-	-	-	-

◎ indicates that the special data registers are applicable to the motion.

MEMO

4.1 Table of Basic Instructions

📖 General instructions

Instruction code	Function	Operand	Execution speed (μs)	Step	Page number
LD	Loading a Form A contact	X, Y, M, S, T, C	3.3	3	4-3
LDI	Loading a Form B contact	X, Y, M, S, T, C	3.3	3	4-3
AND	Connecting a Form A contact in series	X, Y, M, S, T, C	3.3	3	4-4
ANI	Connecting a Form B contact in series	X, Y, M, S, T, C	3.3	3	4-4
OR	Connecting a Form A contact in parallel	X, Y, M, S, T, C	3.3	3	4-5
ORI	Connecting a Form B contact in parallel	X, Y, M, S, T, C	3.3	3	4-5
ANB	Connecting circuit blocks in series	None	2.3	3	4-6
ORB	Connecting circuit blocks in parallel	None	2.3	3	4-7

📖 Output instructions

Instruction code	Function	Operand	Execution speed (μs)	Step	Page number
OUT	Driving a coil	Y, M, S	7.3	3	4-7
SET	Keeping a device ON	Y, M, S	5.6	3	4-8
RST	Resetting a contact or a register	Y, M, S, T, C, D, V, Z	6.9	3	4-8

📖 Timer and counters

API	Instruction code	Function	Operand	Execution speed (μs)	Step	Page number
96	TMR	16-bit timer	T-K or T-D	19	5	4-9
97	CNT	16-bit counter	C-K or C-D (16 bits)	16	5	4-9
97	DCNT	32-bit counter	C-K or C-D (32 bits)	16.5	6	4-10

📖 Rising-edge/Falling-edge detection instructions

API	Instruction code	Function	Operand	Execution speed (μs)	Step	Page number
90	LDP	Starting rising-edge detection	X, Y, M, S, T, C	12.3	3	4-10
91	LDF	Starting falling-edge detection	X, Y, M, S, T, C	12.3	3	4-11
92	ANDP	Connecting rising-edge detection in series	X, Y, M, S, T, C	12.3	3	4-11
93	ANDF	Connecting falling-edge detection in series	X, Y, M, S, T, C	12.3	3	4-12
94	ORP	Connecting rising-edge detection in parallel	X, Y, M, S, T, C	12.6	3	4-12
95	ORF	Connecting falling-edge detection in parallel	X, Y, M, S, T, C	12.6	3	4-13

Rising-edge/Falling-edge output instruction

API	Instruction code	Function	Operand	Execution speed (us)	Step	Page number
89	PLS	Rising-edge output	Y, M	20.7	3	4-14
99	PLF	Falling-edge output	Y, M	20.9	3	4-14

Other instructions

Instruction code	Function	Operand	Execution speed (us)	Step	Page number
P	Pointer	P0~P255	-	1	4-15

4.2 Descriptions of the Basic Instructions

Instruction code	Function	Applicable model
LD	Loading a Form A contact	20PM
		✓

Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

◆ The instruction LD applies to the Form A contact which starts from a busbar or the Form A contact which is the start of a circuit. It reserves the present contents, and stores the state which is gotten in the accumulation register.

Example

Ladder diagram:

Instruction code:		Description:
LD	**X0**	Loading the Form A contact X0
AND	X1	Connecting the Form A contact X1 in series
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
LDI	Loading a Form B contact	20PM
		✓

Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

◆ The instruction LDI applies to the Form B contact which starts from a busbar or the Form B contact which is the start of a circuit. It reserves the present contents, and stores the state which is gotten in an accumulation register.

Example

Ladder diagram:

Instruction code:		Description:
LDI	**X0**	Loading the Form B contact X0
AND	X1	Connecting the Form A contact X1 in series
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
		20PM
AND	Connecting a Form A contact in series	✓

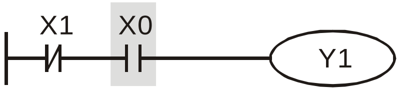
Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

- The instruction AND is used to connect a Form A contact in series. It reads the state of a contact which is connected in series, and performs the AND operation on the previous logical operation result. The final result is stored in an accumulation register.

Example

Ladder diagram:



Instruction code:		Description:
LDI	X1	**Loading the Form B contact X1**
AND	**X0**	Connecting the Form A contact X0
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
		20PM
ANI	Connecting a Form B contact in series	✓

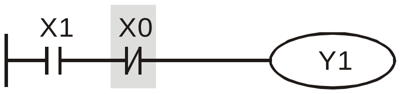
Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

- The instruction ANI is used to connect a Form B contact in series. It reads the state of a contact which is connected in series, and performs the AND operation on the previous logical operation result. The final result is stored in an accumulation register.

Example

Ladder diagram:



Instruction code:		Description:
LD	X1	Loading the Form A contact X1
ANI	**X0**	Connecting the Form B contact X0 in series
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
OR	Connecting a Form A contact in parallel	20PM
		✓

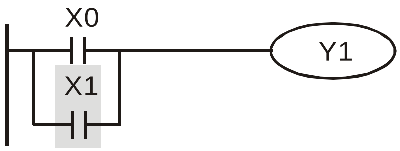
Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

◆ The instruction OR is used to connect a Form A contact in parallel. It reads the state of a contact which is connected in parallel, and performs the OR operation on the previous logical operation result. The final result is stored in an accumulation register.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
OR	**X1**	Connecting the Form A contact X1 in parallel
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
ORI	Connecting a Form B contact in parallel	20PM
		✓

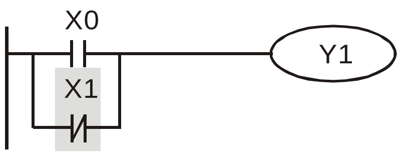
Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

◆ The instruction ORI is used to connect a Form B contact in parallel. It reads the state of a contact which is connected in parallel, and performs the OR operation on the previous logical operation result. The final result is stored in an accumulation register.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
ORI	**X1**	Connecting the Form B contact X1 in parallel
OUT	Y1	Driving the coil Y1

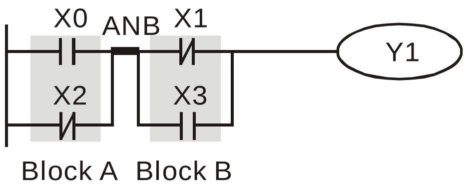
Instruction code	Function	Applicable model
		20PM
ANB	Connecting circuit blocks in series	✓
Operand	None	

Explanation

◆ The instruction ANB is used to perform the AND operation on the logical operation result reserved previously and the contents of the present accumulation register.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
ORI	X2	Connecting the Form B contact X2 in parallel
LDI	X1	Loading the Form B contact X1
OR	X3	Connecting the Form A contact X3 in parallel
ANB		Connecting the circuit blocks in series
OUT	Y1	Driving the coil Y1

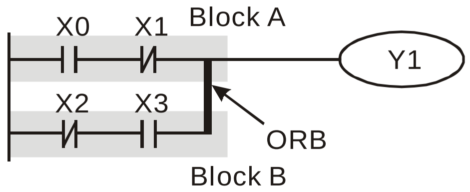
Instruction code	Function	Applicable model
		20PM
ORB	Connecting circuit blocks in parallel	✓
Operand	None	

Explanation

- The instruction ORB is used to perform the OR operation on the logical operation result reserved previously and the contents of the present accumulation register.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
ANI	X1	Connecting the Form B contact X1 in series
LDI	X2	Loading the Form B contact X2
AND	X3	Connecting the Form A contact X3 in series
ORB		Connecting the circuit blocks in parallel
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
		20PM
OUT	Driving a coil	✓

Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	-	✓	✓	✓	-	-	-

Explanation

- The logical operation result prior to the application of the instruction OUT is sent to the device specified.
- Action of a coil

Operation result	OUT		
	Coil	Contact	
		Form A contact (Normally-open contact)	Form B contact (Normally-closed contact)
False	Off	OFF	ON
True	On	ON	OFF

Example

Ladder diagram:

Instruction code:		Description:
LDI	X0	Loading the Form B contact X0
AND	X1	Connecting the Form A contact X1 in series
OUT	**Y1**	Driving the coil Y1

Instruction code	Function	Applicable model
SET	Keeping a device ON	20PM ✓

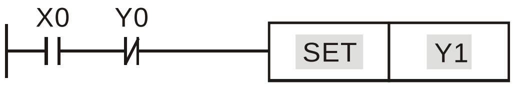
Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	-	✓	✓	✓	-	-	-

Explanation

- ◆ When the instruction SET is driven, the device specified is set to ON. Whether the instruction SET is still driven or not, the device specified remains ON. Users can set the device specified to OFF by means of the instruction RST.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
ANI	Y0	Connecting the Form B contact Y0
SET	**Y1**	Y01 remains ON.

Instruction code	Function	Applicable model
RST	Resetting a contact or a register	20PM ✓

Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	-	✓	✓	✓	✓	✓	✓

Explanation

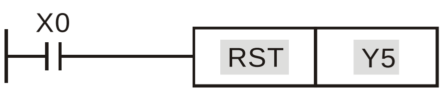
- ◆ When the instruction RST is driven, the device specified acts in the way described below.

Device	State
S, Y, M	The coil and the contact are set to OFF.
T, C	The present timer value or the present counter value becomes 0. The coil and the contact are set to OFF.
D, V, Z	The value becomes 0.

- ◆ If the instruction RST is not executed, the state of the device specified will remain unchanged.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
RST	**Y5**	Resetting Y5

Instruction code	Function	Applicable model
		20PM
TMR	16-bit timer	✓

Operand	T-K	T0~T255, K0~K32,767
	T-D	T0~T255, D0~D9,999

Explanation

- When the instruction TMR is executed, the coil specified is ON, and the timer specified begins to count. If the timer value matches the setting value (timer value≧setting value), the contact specified will act in the way described below.

NO (Normally-open) contact	OFF
NC (Normally-closed) contact	ON

Example

Ladder diagram:

Instruction code:		Description:
LD	X0	Loading the Form A contact X0
TMR	**T5 K1000**	The setting value in the timer T5 is K1000.

Additional remark

- Please refer to the specifications for the model used for more information about the timer range which can be used.

Instruction code	Function	Applicable model
		20PM
CNT	16-bit counter	✓

Operand	C-K	C0~C199, K0~K32,767
	C-D	C0~C199, D0~D9,999

Explanation

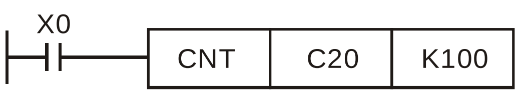
- When the counter coil specified by the instruction CNT is turned from OFF to ON, the counter value increases by 1. If the counter value matches the setting value (counter value=setting value), the contact specified will act in the way described below.

NO (Normally-open) contact	OFF
NC (Normally-closed) contact	ON

- If there are pulses sent to the counter specified by the instruction CNT after the counter value matches the setting value, the state of the contact specified and the counter value will remain unchanged. Users can reset a counter by means of the instruction RST.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
CNT	**C20 K100**	The setting value in the counter C20 is K100.

Instruction code	Function	Applicable model
		20PM
DCNT	32-bit counter	✓

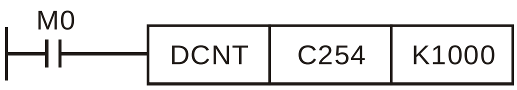
Operand		
Operand	C-K	C200, C204, C208~C255, K-2,147,483,648~K2,147,483,647
	C-D	C200, C204, C208~C255, D0~D9,999

Explanation

- DCNT is an instruction which is used to enable the 32-bit counters C200~C255.
- C221~C2255 are general up/down counters. When the counter coil specified by the instruction DCNT is turned from OFF to ON, the counter value increases or decreases by one according to the setting of M1200~M1234.

Example

Ladder diagram:



Instruction code:		Description:
LD	M0	Loading the Form A contact M0
DCNT	**C254 K1000**	The setting value in the counter C254 is K1000.

Instruction code	Function	Applicable model
		20PM
LDP	Starting rising-edge detection	✓

Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

- The usage of LDP is similar to that of LD, but the action of LDP is different from that of LD. LDP reserves the present contents, and stores the state of the rising edge-triggered contact specified to an accumulation register.

Example

Ladder diagram:

Instruction code:		Description:
LDP	X0	Starting the detection of the state of the rising edge-triggered contact X0
AND	X1	Connecting the Form A contact X1 in series
OUT	Y1	Driving the coil Y1

Additional remark

- Please refer to the specifications for the model used for more information about the operand ranges which can be used.
- If the state of a rising edge-triggered contact in a DVP-20PM series motion controller is ON before the DVP-20PM series motion controller is powered, it is TRUE after the DVP-20PM series motion controller is powered.

Instruction code	Function	Applicable model
		20PM
LDF	Starting falling-edge detection	✓

Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

◆ The usage of LDF is similar to that of LD, but the action of LDP is different from that of LD. LDF reserves the present contents, and stores the state of the falling edge-triggered contact specified to an accumulation register.

Example

Ladder diagram:

Instruction code:		Description:
LDF	**X0**	Starting the detection of the state of the falling edge-triggered contact X0
AND	X1	Connecting the Form A contact X1 in series
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
		20PM
ANDP	Connecting rising-edge detection in series	✓

Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

◆ The instruction ANDP is used to connect a rising edge-triggered contact in series.

Example

Ladder diagram:

Instruction code:		Description:
LD	X0	Loading the Form A contact X0
ANDP	**X1**	Connecting the rising edge-triggered contact X1 in series
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
		20PM
ANDF	Connecting falling-edge detection in series	✓

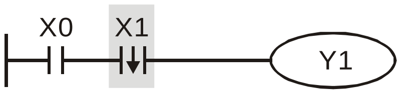
Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

◆ The instruction ANDF is used to connect a falling edge-triggered contact in series.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
ANDF	**X1**	Connecting the falling edge-triggered contact X1 in series
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
		20PM
ORP	Connecting rising-edge detection in parallel	✓

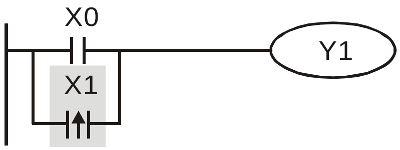
Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

◆ The instruction ORP is used to connect a rising edge-triggered contact in parallel.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
ORP	**X1**	Connecting the rising edge-triggered contact X1 in parallel
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
		20PM
ORF	Connecting falling-edge detection in parallel	✓

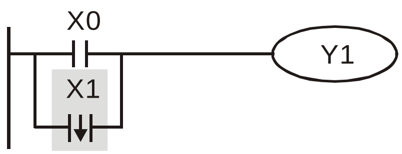
Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	✓	✓	✓	✓	✓	✓	-

Explanation

◆ The instruction ORF is used to connect a falling edge-triggered contact in parallel.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
ORF	**X1**	Connecting the falling edge-triggered contact X1
OUT	Y1	Driving the coil Y1

Instruction code	Function	Applicable model
PLS	Rising-edge output	20PM
		✓

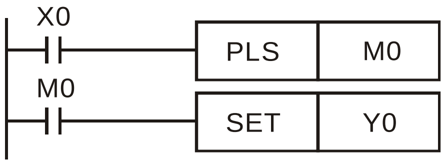
Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	-	✓	✓	-	-	-	-

Explanation

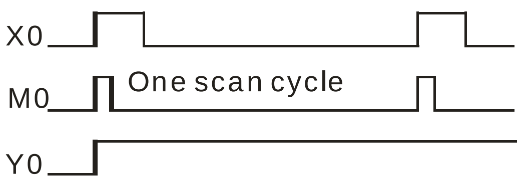
◆ PLS is a rising-edge output instruction. When X0 is turned from OFF to ON, the instruction PLS is executed. M0 sends a pulse for a scan cycle.

Example

Ladder diagram:



Timing diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
PLS	**M0**	M0 is rising edge-triggered.
LD	M0	Loading the Form A contact M0
SET	Y0	Y0 remains ON.

Instruction code	Function	Applicable model
PLF	Falling-edge output	20PM
		✓

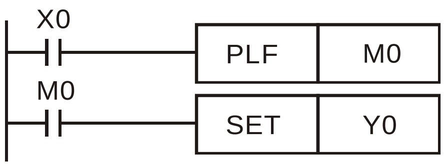
Operand	X0~X377	Y0~Y377	M0~M4,095	S0~S1,023	T0~T255	C0~C255	D0~D9,999
	-	✓	✓	-	-	-	-

Explanation

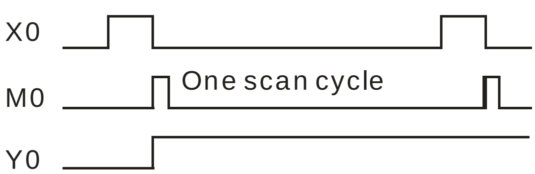
◆ PLF is a falling-edge output instruction. When X0 is turned from ON to OFF, the instruction PLF is executed. MO sends a pulse for a scan cycle.

Example

Ladder diagram:



Timing diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
PLF	**M0**	M0 is falling edge-triggered.
LD	M0	Loading the Form A contact M0
SET	Y0	Y0 remains ON.

Instruction code	Function	Applicable model
		20PM
P	Pointer	✓

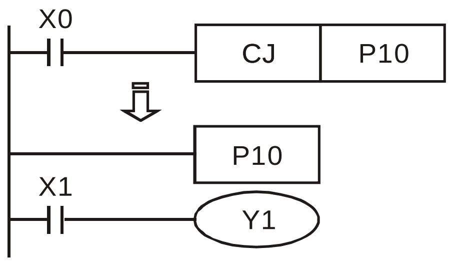
Operand	P0~P255

Explanation

A pointer can be used by API 00 CJ, API 01 CALL, API 256 CJN, and API 257 JMP. The pointers used do not have to start from P0. A pointer number can not be used repeatedly, otherwise an unexpected error will occur.

Example

Ladder diagram:



Instruction code:		Description:
LD	X0	Loading the Form A contact X0
CJ	P10	The jump instruction CJ specifies P10.
:		
P10		Pointer P10
LD	X1	Loading the Form A contact X1
OUT	Y1	Driving the coil Y1

MEMO

5.1 Table of Applied Instructions

Type	API	Instruction code 16-bit	Instruction code 32-bit	Pulse instruction	Function	Model supported 20PM	Step 16-bit	Step 32-bit	Page No.
Loop control	00	CJ	–	✓	Conditional jump	✓	3	–	5-14
	01	CALL	–	✓	Calling a subroutine	✓	3	–	5-17
	02	SRET	–	–	Indicating that a subroutine ends	✓	1	–	5-18
	07	WDT	–	✓	Watchdog timer	✓	1	–	5-20
	08	RPT	–	–	Start of a nested loop (only one loop)	✓	3	–	5-21
	09	RPE	–	–	End of a nested loop	✓	1	–	5-22
Transfer and comparison	10	CMP	DCMP	✓	Comparing values	✓	7	9	5-23
	11	ZCP	DZCP	✓	Zonal comparison Zonal comparison	✓	9	12	5-24
	12	MOV	DMOV	✓	Transferring a value	✓	5	6	5-25
	13	SMOV	–	✓	Transferring digits	✓	11	–	5-26
	14	CML	DCML	✓	Inverting bits	✓	5	9	5-29
	15	BMOV	–	✓	Transferring values	✓	7	–	5-30
	16	FMOV	DFMOV	✓	Transferring a value to several devices	✓	7	13	5-32
	17	XCH	DXCH	✓	Interchanging values	✓	5	9	5-33
	18	BCD	DBCD	✓	Converting a binary value into a binary-coded decimal value	✓	5	5	5-34
	19	BIN	DBIN	✓	Converting a binary-coded decimal value into a binary value	✓	5	5	5-35
Arithmetic	20	ADD	DADD	✓	Binary addition	✓	7	9	5-36
	21	SUB	DSUB	✓	Binary subtraction	✓	7	9	5-38
	22	MUL	DMUL	✓	Binary multiplication	✓	7	9	5-39
	23	DIV	DDIV	✓	Binary division	✓	7	9	5-40
	24	INC	DINC	✓	Adding one to a binary value	✓	3	3	5-41
	25	DEC	DDEC	✓	Subtracting one from a binary value	✓	3	3	5-42
	26	WAND	DWAND	✓	Logical AND operation	✓	7	9	5-43
	27	WOR	DWOR	✓	Logical OR operation	✓	7	9	5-44
	28	WXOR	DWXOR	✓	Logical exclusive OR operation	✓	7	9	5-45
	29	NEG	DNEG	✓	Taking the two's complement of a value	✓	3	3	5-46
Rotation and move	30	ROR	DROR	✓	Rotating bits rightwards	✓	5	9	5-48
	31	ROL	DROL	✓	Rotating bits leftwards	✓	5	9	5-49
	32	RCR	DRCR	✓	Rotating bits rightwards with a carry flag	✓	5	9	5-50
	33	RCL	DRCL	✓	Rotating bits leftwards with a carry flag	✓	5	9	5-51
	34	SFTR	–	✓	Moving the states of bit devices rightwards	✓	9	–	5-52
	35	SFTL	–	✓	Moving the states of bit devices leftwards	✓	9	–	5-53
	36	WSFR	–	✓	Moving the values in word devices rightwards	✓	9	–	5-54
	37	WSFL	–	✓	Moving the values in word devices leftwards	✓	9	–	5-56
	38	SFWR	–	✓	Moving a value and writing it into a word device	✓	7	–	5-57
	39	SFRD	–	✓	Moving a value and reading it from a word device	✓	7	–	5-58

Type	API	Instruction code		Pulse instruction	Function	Model supported	Step		Page No.
		16-bit	32-bit			20PM	16-bit	32-bit	
Data processing	**40**	ZRST	–	✓	Resetting a zone	✓	5	–	5-59
	41	DECO	–	✓	Decoder	✓	7	–	5-60
	42	ENCO	–	✓	Encoder	✓	7	–	5-62
	43	SUM	DSUM	✓	Number of bits which are ON	✓	5	9	5-64
	44	BON	DBON	✓	Checking the state of a bit	✓	7	13	5-65
	45	MEAN	DMEAN	✓	Mean	✓	7	13	5-66
	46	ANS	–	–	Driving an annunciator	✓	7	–	5-67
	47	ANR	–	✓	Resetting an annunciator	✓	1	–	5-68
	48	SQR	DSQR	✓	Square root of a binary value	✓	5	9	5-70
	49	–	DFLT	✓	Converting a binary integer into a binary floating-point value	✓	–	6	5-71
High-speed processing	**50**	REF	–	✓	Refreshing the states of I/O devices	✓	5	–	5-73
Convenience	**61**	SER	DSER	✓	Searching data	✓	9	17	5-74
	66	ALT	–	✓	Alternating between ON and OFF	✓	3	-	5-76
	67	RAMP	DRAMP	–	Ramp	✓	9	17	5-77
	69	SORT	DSORT	–	Sorting data	✓	11	21	5-79
I/O	**78**	FROM	DFROM	✓	Reading data from a control register in a special module	✓	9	12	5-81
	79	TO	DTO	✓	Writing data into a control register in a special module	✓	9	13	5-82
	87	ABS	DABS	✓	Absolute value	✓	3	5	5-85
Basic instructions	**89**	PLS	–	–	Rising-edge output	✓	3	–	4-14
	90	LDP	–	–	Starting rising-edge detection	✓	3	–	4-10
	91	LDF	–	–	Starting falling-edge detection	✓	3	–	4-11
	92	ANDP	–	–	Connecting rising-edge detection in series	✓	3	–	4-11
	93	ANDF	–	–	Connecting falling-edge detection in series	✓	3	–	4-12
	94	ORP	–	–	Connecting rising-edge detection in parallel	✓	3	–	4-12
	95	ORF	–	–	Connecting falling-edge detection in parallel	✓	3	–	4-13
	96	TMR	–	–	16-bit timer	✓	5	–	4-9
	97	CNT	DCNT	–	16-bit counter	✓	5	6	4-9
	99	PLF	–	–	Falling-edge output	✓	3	–	4-14
Communication	**100**	MODRD	–	–	Reading Modbus data	✓	7	–	5-86
	101	MODWR	–	–	Writing Modbus data	✓	7	–	5-90
Floating-point value	**110**	–	DECMP	✓	Comparing binary floating-point values	✓	7	9	5-95
	111	–	DEZCP	✓	Binary floating-point zonal comparison	✓	9	12	5-96
	112	–	DMOVR	✓	Transferring a floating-point value	✓	–	9	5-97
	116	–	DRAD	✓	Converting a degree to a radian	✓	–	6	5-98
	117	–	DDEG	✓	Converting a radian to a degree	✓	–	6	5-99
	120	–	DEADD	✓	Binary floating-point addition	✓	7	9	5-100
	121	–	DESUB	✓	Binary floating-point subtraction	✓	7	9	5-101

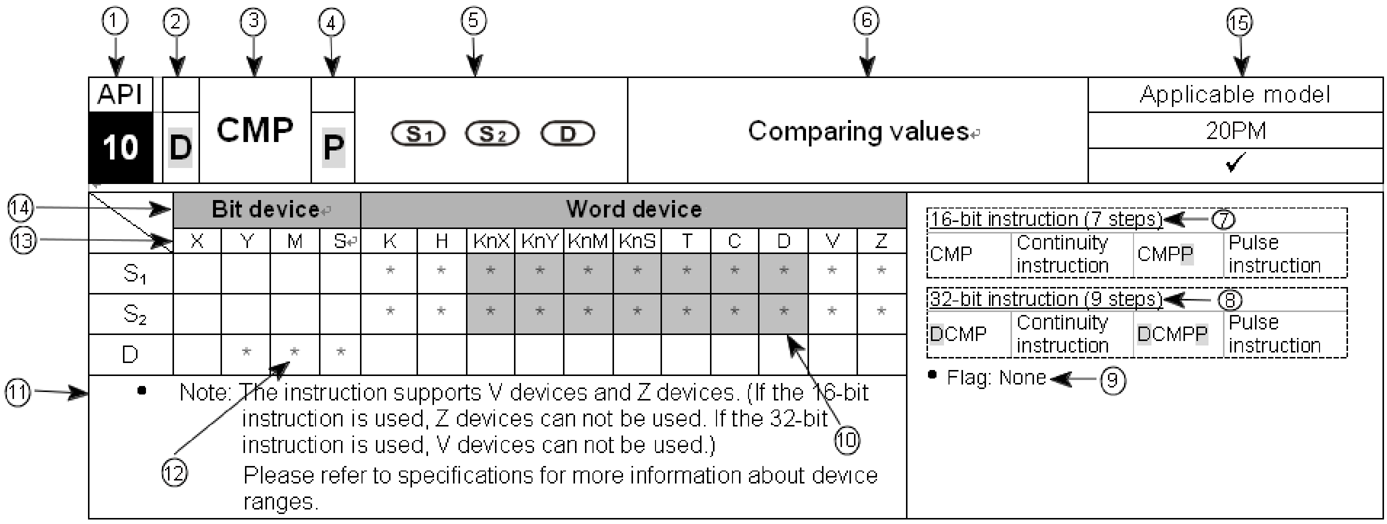
Type	API	Instruction code		Pulse instruction	Function	Model supported	Step		Page No.
		16-bit	32-bit			20PM	16-bit	32-bit	
Floating-point value	122	–	DEMUL	✓	Binary floating-point multiplication	✓	7	9	5-102
	123	–	DEDIV	✓	Binary floating-point division	✓	7	9	5-103
	124	–	DEXP	✓	Exponent of a binary floating-point value	✓	–	6	5-104
	125	–	DLN	✓	Natural logarithm of a binary floating-point value	✓	–	6	5-105
	126	–	DLOG	✓	Logarithm of a binary floating-point value	✓	–	9	5-106
	127	–	DESQR	✓	Square root of a binary floating-point value	✓	5	6	5-107
	128	–	DPOW	✓	Power of a floating-point value	✓	–	9	5-108
	129	–	DINT	✓	Converting a binary floating-point value into a binary integer	✓	–	6	5-109
	130	–	DSIN	✓	Sine of a binary floating-point value	✓	5	6	5-110
	131	–	DCOS	✓	Cosine of a binary floating-point value	✓	5	6	5-112
	132	–	DTAN	✓	Tangent of a binary floating-point value	✓	5	6	5-114
	133	–	DASIN	✓	Arcsine of a binary floating-point value	✓	–	6	5-116
	134	–	DACOS	✓	Arccosine of a binary floating-point value	✓	–	6	5-117
	135	–	DATAN	✓	Arctangent of a binary floating-point value	✓	–	6	5-118
	136	–	DSINH	✓	Hyperbolic sine of a binary floating-point value	✓	–	6	5-119
	137	–	DCOSH	✓	Hyperbolic cosine of a binary floating-point value	✓	–	6	5-120
	138	–	DTANH	✓	Hyperbolic tangent of a binary floating-point value	✓	–	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
Logical 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

Type	API	Instruction code		Pulse instruction	Function	Model supported	Step		Page No.
		16-bit	32-bit			20PM	16-bit	32-bit	
Comparison instructions	**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
Comparison instructions	**237**	AND<=	DAND<=	–	S1≦ S2	✓	5	7	5-130
	238	AND>=	DAND>=	–	S1≧ 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≠S2	✓	5	7	5-131
	245	OR<=	DOR<=	–	S1≦ S2	✓	5	7	5-131
	246	OR>=	DOR>=	–	S1≧ S2	✓	5	7	5-131
Other instructions	**147**	SWAP	DSWAP	✓	Interchanging the high byte in a device with the low byte in the device	✓	3	5	5-132
	154	RAND	DRAND	✓	Random value	✓	7	13	5-133
	202	SCAL	–	✓	Scale	✓	9	–	5-134
	203	SCLP	DSCLP	✓	Parameter scale	✓	7	13	5-136
	256	CJN	–	✓	Negated conditional jump	✓	3	–	5-140
	257	JMP	–	–	Unconditional jump	✓	3	–	5-141
	258	BRET	–	–	Returning to a busbar	✓	1	–	5-142
	259	MMOV	–	✓	Converting a 16-bit value into a 32-bit value	✓	6	–	5-143
	260	RMOV	–	✓	Converting a 32-bit value into a 16-bit value	✓	6	–	5-144

5.2 Structure of an Applied Instruction

◆ An applied instruction is composed of an instruction name and operands.
  Instruction name: An instruction name represents a function.
  Operand: An operand is the object of an operation.
  An instruction name occupie one step. The number of steps an operand occupies can be two or three, depending on the instruction used is a 16-bit instruciton or a 32-bit instruction.

◆ Descriptions of the applied instructions



① API number

② The upper cell indicates a 16-bit instruction. If the upper cell is a dotted cell, there will be no 16-bit instruction.
The lower cell indicates a 32-bit instruction. If the lower cell is a dotted cell, there is no 32-bit instruction. If there is a 32-bit instruction, **D** is displayed in the lower cell, e.g. API 10 DCMP.

③ Applied instruction name

④ If ☺ is displayed in the upper cell, a pulse instruction is generally used.
The lower cell indicates a pulse instruction. If there is a pulse instruction, **P** is displayed in the lower cell, e.g. API 12 MOVP.

⑤ Operands

⑥ Function

⑦ Number of steps occupied by a 16-bit instruction, continuity instruction name, and pulse instruction name

⑧ Number of steps occupied by a 32-bit instruction, continuity instruction name, and pulse instruction name

⑨ Flags related to an applied instruction

⑩ The devices marked with ' * ' displayed in grayscale can be modified by V devices and Z devices.

⑪ Points for attention

⑫ The devices marked with ' * ' can be used.

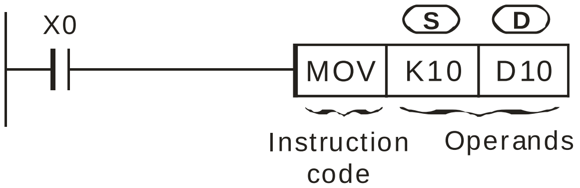
⑬ Device name

⑭ Device type

⑮ Applicable model

◆ Typing an applied instruction
Some applied instructions are composed of instruction names, e.g. BRET and SRET, but most applied instructions are composed of instruction names and operands.
The applied instructions that a DVP-20PM series motion controller can use are assigned the instruction numbers API 00~API 260. Besides, every applied instruction is assigned a mnemonic. For example,

the mnemonic of API 12 is MOV. If users want to type an instruction by means of PMSoft, they can type the mnemonic assigned to the instruction. If users want to type an instruction by means of the handheld programming panel DVPHPP03, they can type the API number assigned to the instruction. Every applied instruction specifies operands. Take the instruction MOV for instance.



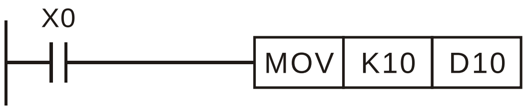
The instruction is used to move the value in the operand **S** to the operand **D**.

S	Source operand If there is more one source operand, the source operands will be represented by $\mathbf{S_1}$, $\mathbf{S_2}$, and etc.
D	Destination operand If there is more than one destination operand, the destination operands will be represented by $\mathbf{D_1}$, $\mathbf{D_2}$, and etc.
If operands are constants, they will be represented by **m**, $\mathbf{m_1}$, $\mathbf{m_2}$, **n**, $\mathbf{n_1}$, $\mathbf{n_2}$, and etc.	

◆ Length of an operand (16-bit instruction or 32-bit instruction)

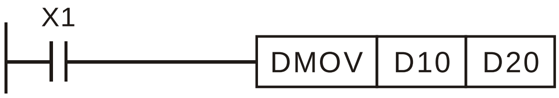
The values in operands can be grouped into 16-bit values and 32-bit values. In order to process values of difference lengths, some applied instructions are grouped into 16-bit instructions and 32-bit instructions. After "D" is added to the front of a 16-bit instruction, the instruction becomes a 32-bit instruction.

The instruction MOV is a 16-bit instruction.



When X0 is ON, K10 is moved to D10.

The instruction DMOV is a 32-bit instruction.

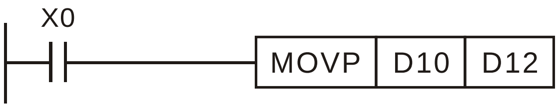


When X1 is ON, the value in (D11, D10) is moved to (D21, D20).

◆ Continuity instruction/Pulse instruction

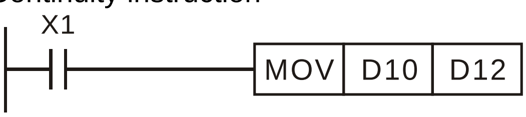
The applied instructions can be grouped into continuity instructions and pulse instructions in terms of the ways the applied instructions are executed. If an instruction in a program is not executed, the execution of the program will take less time. As a result, if there are pulse instructions in a program, the scan cycle will be shorter. If "P" is added to the back of an instruction, the instruction becomes a pulse instruction. Some instructions are mostly used as pulse instructions, e.g. INC and DEC. <u>If ☺ appears at the upper-right side of an instruction, the instruction is generally used as a pulse instruction.</u>

Pulse instruction



When X0 is turned from OFF to ON, the instruction MOVP is executed once. MOVP will not be executed again during the scan cycle, and therefore it is a pulse instruction.

Continuity instruction



Whenever X1 is ON, the instruction MOV is executed once. MOV is a continuity instruction.

When the contacts X0 and X1 are OFF, the instructions are not executed, and the values in the destation operands are not changed.

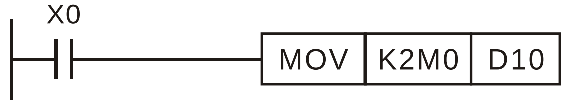
◆ Operand

1. A word device can consist of bit devices. Applied instructions can use KnX, KnY, KnM, and KnS. Values can be stored in KnX, KnY, KnM, and KnS.
2. Data registers, timers, counters, and index registers can be used as general operands.
3. A data register is a 16-bit register. If users want to use a 32-bit data register, they have to specify two consecutive data registers.
4. If a 32-bit instruction uses D0 as an operand, the 32-bit data register composed of D1 and D0 will be used. D1 occupies the high 16 bits, and D0 occupy the low 16 bits. Timers and the 16-bit

counters C0~C199 can be used in the same way.

5. If the 32-bit counters C200~C255 are used as data registers, they can be operands used by 32-bit instructions.

- Operand type
1. X devices, Y devices, M devices, and S devices can only be turned ON or OFF. They are bit devices.
2. 16-bit (or 32-bit) T devices, C device, D devices, V devices, and Z devices are word devices.
3. If Kn is added to the front of an X/Y/M/S device, a word device will be formed. For example, K2M0 represents a device composed of the eight bit devices M0~M7.



When X0 is ON, the values of M0~M7 are moved to bit 0~bit 7 in D10, and bit 8~bit 15 are set to 0.

- Values in word devices composed of bit devices

16-bit instruction	
A 16-bit value is in the range of K-32,768 to K32,767.	
Value in a word device composed of bit devices	
K1 (4 bits)	0~15
K2 (8 bits)	0~255
K3 (12 bits)	0~4,095
K4 (16 bits)	-32,768~+32,767

32-bit instruction	
A 32-bit value is in the range of K-2,147,483,648 to K2,147,483,647.	
Value in a word device composed of bit devices	
K1 (4 bits)	0~15
K2 (8 bits)	0~255
K3 (12 bits)	0~4,095
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

- General flags

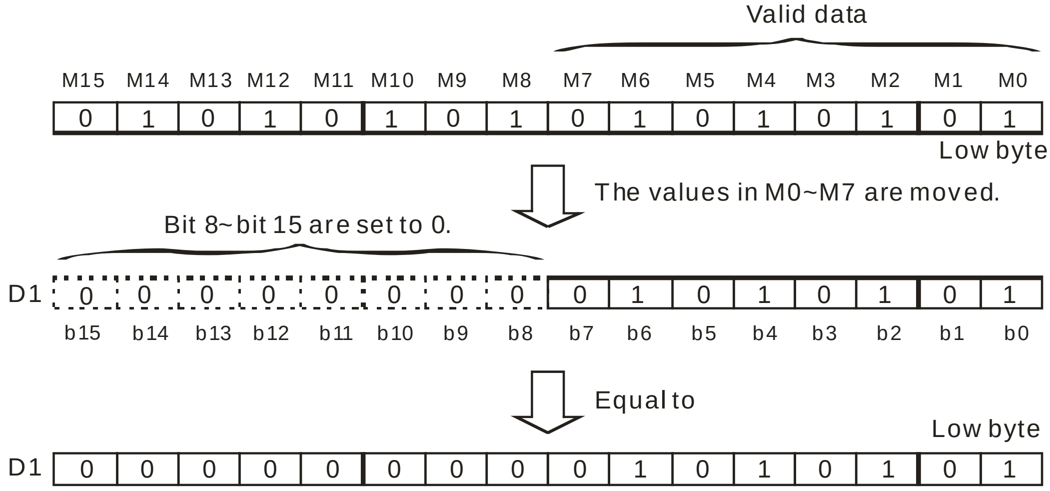
Example: M1968 is a zero flag, M1969 is a borrow flag, and M1970 is a carry flag

Every flag in a DVP-20PM series motion controller corresponds to an operation result.

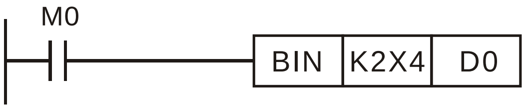
The state of a flag varies with an operation result. For example, if the instruction ADD/SUB/MUL/DIV is used in the main program O100~M102, the operation result gotten will affect the states of M1968~M1970. However, if the instruciton is not executed, the states of the flags will remain unchanged. The states of flags are related to instructions. Please refer to the explanations of instructions for more information.

5.3 Processing Values

- X devices, Y devices, M devices, and S devices can only be turned ON or OFF. They are bit devices. Values can be stored in T device, C devices, D devices, V devices, and Z devices. They are word devices. If Kn is added to the front of an X/Y/M/S device, a word device will be formed.
- If Kn is added to the front of an X/Y/M/S device, a word device will be formed. For example, K2M0 represents a device composed of the eight bit devices M0~M7.



- The value in K1M0 is moved to a 16-bit register, and bit 4~bit 15 in the register are set to 0. The value in K2M0 is moved to a 16-bit register, and bit 8~bit 15 in the register are set to 0. The value in K3M0 is moved to a 16-bit register, and bit 12~bit 15 in the register are set to 0. The value in K1M0 is moved to a 32-bit register, and bit 4~bit 31 in the register are set to 0. The value in K2M0 is moved to a 32-bit register, and bit 8~bit 31 in the register are set to 0. The value in K3M0 is moved to a 32-bit register, and bit 12~bit 31 in the register are set to 0. The value in K4M0 is moved to a 32-bit register, and bit 16~bit 31 in the register are set to 0. The value in K5M0 is moved to a 32-bit regiser, and bit 20~bit 31 in the register are set to 0. The vlaue in K6M0 is moved to a 32-bit register, and bit 24~bit 31 in the register are set to 0. The value in K7M0 is moved to a 32-bit register, and bit 28~bit 31 in the register are set to 0.
- If Kn is in the range of K1~K3 (or K4~K7), the bits which are not assigned values in the 16-bit register (the 32-bit register) to which a value is moved will be set to 0. As a result, operations will be performed on positive values if Kn is in the range of K1~K3 (or K4~K7).



☞ The binary-coded decimal value in X4~X11 is converted into a binary value, and the binary value is stored in D0.

- Users can specify bit device numbers freely. It is suggested X device numbers/Y devuce numbers should end with 0, and that M device numbers/S device numbers should start from a number which is a multiple of 8.
- Consecutive devices

Take data registers for instances. D0, D1, D2, D3, and D4 are consecutive data registers.

The consecutive word devices composed of bit devices are shown below.

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

The consecutive word devices composed of bit devices are shown above. To avoid confusion, please do not skip any word device composed of bit devices. Beisdes, if a 32-bit operation is performed on K4Y0, the high 16 bits in the 32-bit register to which the value in K4Y0 is moved will be set to 0. If a 32-bit value is required, please use K8Y0.

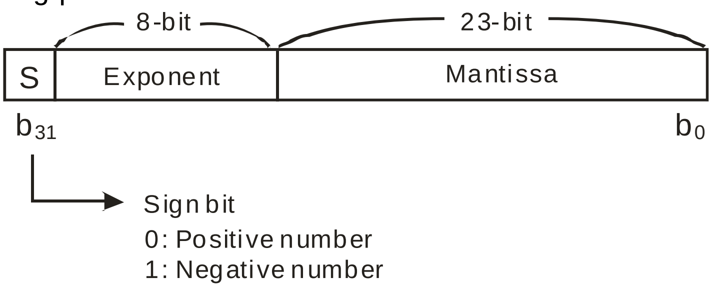
After an operation is performed, the binary integer gotten will be given priority. For example, 40÷3=13, and the remainder 1 is dropped. The integer part of the square root of an integer is retained, and the fractional part of the square root is dropped. However, if a decimal instruiction is used, a decimal will be gotten.

The applied intructions listed below are decimal instructions.

API 110 (D ECMP)	API 111 (D EZCP)	API 116 (D RAD)	API 117 (D DEG)
API 120 (D EADD)	API 121 (D ESUB)	API 122 (D EMUL)	API 123 (D EDIV)
API 124 (D EXP)	API 125 (D LN)	API 126 (D LOG)	API 127 (D ESQR)
API 128 (D POW)	API 129 (D INT)	API 130 (D SIN)	API 131 (D COS)
API 132 (D TAN)	API 133 (D ASIN)	API 134 (D ACOS)	API 135 (D ATAN)
API 136 (D SINH)	API 137 (D COSH)	API 138 (D TANH)	

Representations of binary floating-point values

The floating-point values in a DVP-20PM series motion controller are 32-bit floating-point values, and the representations of the floating-point values conform to the IEEE 754 standard.



Representation of a floating-point value:

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

A 32-bit floating-point value is in the range of $\pm 2^{-126}$ to $\pm 2^{+128}$, that is, a 32-bit floating-point value is in the range of $\pm 1.1755\times10^{-38}$ to $\pm 3.4028\times10^{+38}$.

Example 1: 23 is represented by a 32-bit floating-point value.

Step 1: Converting 23 into a binary value: 23.0=10111

Step 2: Normalizing the binary value: 10111=1.0111×24 (0111 is a mantissa, and 4 is an exponent)

Step 3: Getting the exponent which is stored

$\because$ E-B=4 →E-127=4 $\therefore$ E=131=10000011_2

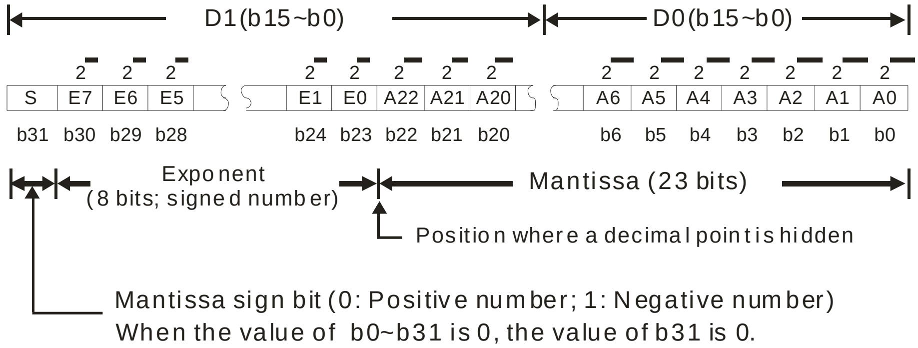
Step 4: Combining the sign bit, the exponent, and the mantissa to form a floating-point value.

0 100**0001**1 011**1000**0000**0000**0000**0000**$_2$=$41B80000_{16}$

Example 2: -23.0 is represented by a 32-bit floating-point value.

-23.0 is converted in the same way as 23.0. Users only need to change the sign bit to 1.

A DVP-20PM series motion controller uses two consecutive registers to form a 32-bit floating-point values. Take (D1, D0) in which a bianry floating-point value is stored for instance.



Decimal floating-point value

- Since binary floating-point values are not widely accepted by people, they can be converted into decimal floating-point values. However, the decimals on which operations are performed in a DVP-20PM series motion controller are still binary floating-point values.
- A decimal floating-point value is stored in two consecutive registers. The constant part is stored in the register whose device number is smaller, and the exponent part is stored in the register whose device number is bigger.

  Take (D1, D0) for instance.

  $$\text{Decimal floating-point number}=[\text{Constant D0}] * 10^{[\text{Exponent D1}]}$$

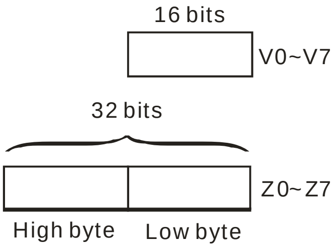
  Base: D0=±1,000~±9,999

  Exponent: D1=-41~+35

  Besides, the base 100 does not exist in D0 because 100 is represented by $1,000\times10^{-1}$. A decimal floating-point value is in the range of $\pm 1,175\times10^{-41}$ to $\pm 3,402\times10^{+35}$.
- If the instruction ADD/SUB/MUL/DIV is used in the main program O100~M102, the operation result gotten will affect the states of M1968~M1970. If a floating-point operation instruction is used, the result gotten will also affect the state of the zero flag M1968, the state of the borrow flag M1969, and the state of the carry flag M1970.
  - Zero flag: If the operation result gotten is 0, M1968 will be ON.
  - Carry flag: If the absolute value of the operaiton result gotten is greater than the maximum value allowed, M1969 will be ON.
  - Borrow flag: If the absolute value of the operation result gotten is less than the minimum value allowed, M1970 will be ON.

5.4 Using Index Registers to Modify Operands

V devices are 16-bit index registers, and Z devices are 32-bit index registers. There are 6 V devices (V0~V5), 8 Z devices (Z0~Z7) in a DVP-10PM series motion controller.



V devices are 16-bit registers. Data can be freely written into a V device, and data can be freely read from a V device. If a 32-bit value is required, please use a Z device.

Index registers can be used to modify P/I/X/Y/M/S/KnX/KnY/KnM/KnS/T/C/D devices, but they can not be used to modify index registers, constants, and Kn. For example, K4@Z0 is invalid, K4M0@Z0 is valid, and K0@Z0M0 is invalid. The devices marked with ‘ * ’ displayed in grayscale in the table in the explanation of an applied instruction can be modified by V devices and Z devices..

5.5 Instruction Index

- Arranging applied instructions in alphabetical order

Type	API	Instruction code 16-bit	Instruction code 32-bit	Pulse instruction	Function	Step 16-bit	Step 32-bit	Page No.
A	87	ABS	DABS	✓	Absolute value	3	5	5-85
	20	ADD	DADD	✓	Binary addition	7	9	5-36
	66	ALT	–	✓	Alternating between ON and OFF	3	–	5-76
	218	AND&	DAND&	–	S1&S2	5	7	5-127
	220	AND^	DAND^	–	S1^S2	5	7	5-127
	219	AND|	DAND|	–	S1|S2	5	7	5-127
	234	AND<	DAND<	–	S1 < S2	5	7	5-130
	93	ANDF	–	–	Connecting falling-edge detection in series	3	–	4-12
	92	ANDP	–	–	Connecting rising-edge detection in series	3	–	4-11
	47	ANR	–	✓	Resetting an annunciator	1	–	5-68
	46	ANS	–	–	Driving an annunciator	7	–	5-67
	237	AND<=	DAND<=	–	S1≦ S2	5	7	5-130
	236	AND<>	DAND<>	–	S1≠S2	5	7	5-130
	232	AND=	DAND=	–	S1 = S2	5	7	5-130
	233	AND>	DAND>	–	S1 > S2	5	7	5-130
	238	AND>=	DAND>=	–	S1≧ S2	5	7	5-130
	134	–	DACOS	✓	Arccosine of a binary floating-point value	–	6	5-117
	133	–	DASIN	✓	Arcsine of a binary floating-point value	–	6	5-116
	135	–	DATAN	✓	Arctangent of a binary floating-point value	–	6	5-118
B	18	BCD	DBCD	✓	Converting a binary value into a binary-coded decimal value	5	5	5-34
	19	BIN	DBIN	✓	Converting a binary-coded decimal value into a binary value	5	5	5-35
	15	BMOV	–	✓	Transferring values	7	–	5-30
	44	BON	DBON	✓	Checking the state of a bit	7	13	5-65
	258	BRET	–	–	Returning to a busbar	1	–	5-142
C	01	CALL	–	✓	Calling a subroutine	3	–	5-17
	131	–	DCOS	✓	Cosine of a binary floating-point value	5	6	5-112
	137	–	DCOSH	✓	Hyperbolic cosine of a binary floating-point value	–	6	5-120
	00	CJ	–	✓	Conditional jump	3	–	5-14
	256	CJN	–	✓	Negated conditional jump	3	–	5-140
	14	CML	DCML	✓	Inverting bits	5	9	5-29

Type	API	Instruction code		Pulse instruction	Function	Step		Page No.
		16-bit	32-bit			16-bit	32-bit	
C	10	CMP	DCMP	✓	Comparing values	7	9	5-23
	97	CNT	DCNT	–	16-bit counter	5	6	4-9
D	25	DEC	DDEC	✓	Subtracting one from a binary value	3	3	5-42
	41	DECO	–	✓	Decoder	7	–	5-60
	117	–	DDEG	✓	Converting a radian to a degree	–	6	5-99
	23	DIV	DDIV	✓	Binary division	7	9	5-40
E	42	ENCO	–	✓	Encoder	7	–	5-62
	172	–	DADDR	✓	Floating-point addition	–	13	5-122
	175	–	DDIVR	✓	Floating-point division	–	13	5-125
	120	–	DEADD	✓	Binary floating-point addition	7	9	5-100
	110	–	DECMP	✓	Comparing binary floating-point values	7	9	5-95
	123	–	DEDIV	✓	Binary floating-point division	7	9	5-103
	122	–	DEMUL	✓	Binary floating-point multiplication	7	9	5-102
	127	–	DESQR	✓	Square root of a binary floating-point value	5	6	5-107
	121	–	DESUB	✓	Binary floating-point subtraction	7	9	5-101
	124	–	DEXP	✓	Exponent of a binary floating-point value	–	6	5-104
	111	–	DEZCP	✓	Binary floating-point zonal comparison	9	12	5-96
	112	–	DMOVR	✓	Transferring a floating-point value	–	9	5-97
	174	–	DMULR	✓	Floating-point multiplication	–	13	5-124
	173	–	DSUBR	✓	Floating-point subtraction	–	13	5-123
F	49	–	DFLT	✓	Converting a binary integer into a binary floating-point value	–	6	5-71
	16	FMOV	DFMOV	✓	Transferring a value to several devices	7	13	5-32
	78	FROM	DFROM	✓	Reading data from a control register in a special module	9	12	5-81
I	24	INC	DINC	✓	Adding one to a binary value	3	3	5-41
	129	–	DINT	✓	Converting a binary floating-point value into a binary integer	–	6	5-109
J	257	JMP	–	–	Unconditional jump	3	–	5-141
L	215	LD&	DLD&	–	S1&S2	5	7	5-126
	217	LD^	DLD^	–	S1^S2	5	7	5-126
	216	LD|	DLD|	–	S1|S2	5	7	5-126
	226	LD<	DLD<	–	S1 < S2	5	7	5-129
	229	LD<=	DLD<=	–	S1≦ S2	5	7	5-129
	228	LD<>	DLD<>	–	S1≠S2	5	7	5-129
	224	LD=	DLD=	–	S1 = S2	5	7	5-129
	225	LD>	DLD>	–	S1 > S2	5	7	5-129
	230	LD>=	DLD>=	–	S1≧ S2	5	7	5-129
	125	–	DLN	✓	Natural logarithm of a binary floating-point value	–	6	5-105
	126	–	DLOG	✓	Logarithm of a binary floating-point value	–	9	5-106
	90	LDP	–	–	Starting rising-edge detection	3	–	4-10
	91	LDF	–	–	Starting falling-edge detection	3	–	4-11
M	45	MEAN	DMEAN	✓	Mean	7	13	5-66
	259	MMOV	–	✓	Converting a 16-bit value into a 32-bit value	6	–	5-143
	100	MODRD	–	–	Reading Modbus data	7	–	5-86
	101	MODWR	–	–	Writing Modbus data	7	–	5-90
	12	MOV	DMOV	✓	Transferring a value	5	6	5-25
	22	MUL	DMUL	✓	Binary multiplication	7	9	5-39
N	29	NEG	DNEG	✓	Taking the two's complement of a value	3	3	5-46
O	221	OR&	DOR&	–	S1&S2	5	7	5-128
	223	OR^	DOR^	–	S1^S2	5	7	5-128

Type	API	Instruction code		Pulse instruction	Function	Step		Page No.
		16-bit	32-bit			16-bit	32-bit	
O	**222**	OR|	DOR|	–	S1|S2	5	7	5-128
	242	OR<	DOR<	–	S1 < S2	5	7	5-131
	245	OR<=	DOR<=	–	S1≦ S2	5	7	5-131
	244	OR<>	DOR<>	–	S1 ≠ S2	5	7	5-131
	240	OR=	DOR=	–	S1 = S2	5	7	5-131
	241	OR>	DOR>	–	S1 > S2	5	7	5-131
	246	OR>=	DOR>=	–	S1≧ S2	5	7	5-131
	95	ORF	–	–	Connecting falling-edge detection in parallel	3	–	4-13
	94	ORP	–	–	Connecting rising-edge detection in parallel	3	–	4-12
P	**99**	PLF	–	–	Falling-edge output	3	–	4-14
	89	PLS	–	–	Rising-edge output	3	–	4-14
	128	–	DPOW	✓	Power of a floating-point value	–	9	5-108
R	**116**	–	DRAD	✓	Converting a degree to a radian	–	6	5-98
	67	RAMP	DRAMP	–	Ramp	9	17	5-77
	154	RAND	DRAND	✓	Random value	7	13	5-133
	33	RCL	DRCL	✓	Rotating bits leftwards with a carry flag	5	9	5-51
	32	RCR	DRCR	✓	Rotating bits rightward with a carry flag	5	9	5-50
	50	REF	–	✓	Refreshing the states of I/O devices	5	-	5-73
	260	RMOV	–	✓	Converting a 32-bit value into a 16-bit value	6	–	5-144
	31	ROL	DROL	✓	Rotating bits leftwards	5	9	5-49
	30	ROR	DROR	✓	Rotating bits rightwards	5	9	5-48
	09	RPE	–	–	End of a nested loop	1	–	5-22
	08	RPT	–	–	Start of a nested loop (only one loop)	3	–	5-21
S	**202**	SCAL	–	✓	Scale	9	–	5-134
	203	SCLP	DSCLP	✓	Parameter scale	7	13	5-136
	61	SER	DSER	✓	Searching data	9	17	5-74
	39	SFRD	–	✓	Moving a value and reading it from a word device	7	–	5-58
	35	SFTL	–	✓	Moving the states of bit devices leftwards	9	–	5-53
	34	SFTR	–	✓	Moving the states of bit devices rightwards	9	–	5-52
	38	SFWR	–	✓	Moving a value and writing it into a word device	7	–	5-57
	13	SMOV	–	✓	Transferring digits	11	–	5-26
	69	SORT	DSORT	–	Sorting data	11	21	5-79
	130	–	DSIN	✓	Sine of a binary floating-point value	5	6	5-110
	136	–	DSINH	✓	Hyperbolic sine of a binary floating-point value	–	6	5-119
	48	SQR	DSQR	✓	Square root of a binary value	5	9	5-70
	02	SRET	–	–	Indicating that a subroutine ends	1	–	5-18
	21	SUB	DSUB	✓	Binary subtraction	7	9	5-38
	43	SUM	DSUM	✓	Number of bits which are ON	5	9	5-64
	147	SWAP	DSWAP	✓	Interchanging the high byte in a device with the low byte in the device	3	5	5-132
T	**132**	–	DTAN	✓	Tangent of a binary floating-point value	5	6	5-114
	138	–	DTANH	✓	Hyperbolic tangent of a binary floating-point value	–	6	5-121
	96	TMR	–	–	16-bit timer	5	–	4-9
	79	TO	DTO	✓	Writing data into a control register in a special module	9	13	5-82
W	**26**	WAND	DWAND	✓	Logical AND operation	7	9	5-43
	07	WDT	–	✓	Watchdog timer	1	–	5-20
	27	WOR	DWOR	✓	Logical OR operation	7	9	5-44
	37	WSFL	–	✓	Moving the values in word devices leftwards	9	–	5-56
	36	WSFR	–	✓	Moving the values in word devices rightwards	9	–	5-54
	28	WXOR	DWXOR	✓	Logical exclusive OR operation	7	9	5-45

Type	API	Instruction code		Pulse instruction	Function	Step		Page No.
		16-bit	32-bit			16-bit	32-bit	
X	**17**	XCH	DXCH	✓	Interchanging values	5	9	5-33
Z	**11**	ZCP	DZCP	✓	Zonal comparison	9	12	5-24
	40	ZRST	–	✓	Resetting a zone	5	–	5-59

5.6 Descriptions of the Applied Instructions

API						Applicable model
00		CJ	P	S	Conditional jump	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z

- Note: S can be a pointer.
  S can be a pointer in the range of P0 to P255.
  A pointer can not be modified by a V device or a Z device.

16-bit instruction (3 steps)

CJ	Continuity instruction	CJP	Pulse instruction

32-bit instruction

-	-	-	-

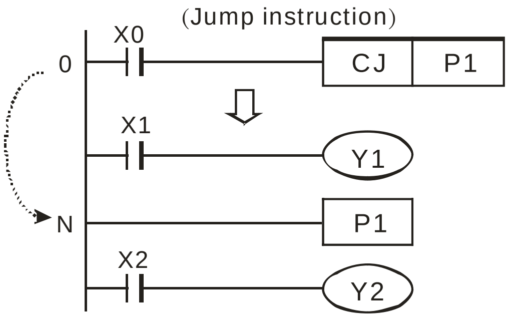
- Flag: None

Explanation

- **S**: Pointer which points to a jump destination
- If some part of the main program O100 does not need to be executed, users can use CJ or CJP to shorten the scan time. Besides, if a dual output is used, users can use CJ or CJP.
- If the program specified by a pointer is prior to the instruction CJ, a watchdog timer error will occur, and the main program will not be executed. Please use the instruction carefully.
- The instruction CJ can specify the same pointer repeatedly. The pointer specified by CJ can not be the same as the pointer specified by CALL, otherwise an error will occur.
- When the instruction CJ/CJP in a program is executed, the actions of the devices in the program are as follows.
  1. The states of the Y devices, the states of the M devices, and the states of the S devices in the program remain the same as those before the execution of the jump.
  2. The 10 millisecond timers in the program stop counting.
  3. The general counters in the program stop counting, and the general applied instructions in the program are not executed.
  4. If the instructions which are used to reset the timers in the program are driven before the jump is executed, the timers will still be reset during the execution of the jump.

Example 1

- When X0 is ON, the execution of the program jumps from address 0 to address N (P1).
- When X0 is OFF, the execution of the program starts from address 0, and the instruction CJ is not executed.



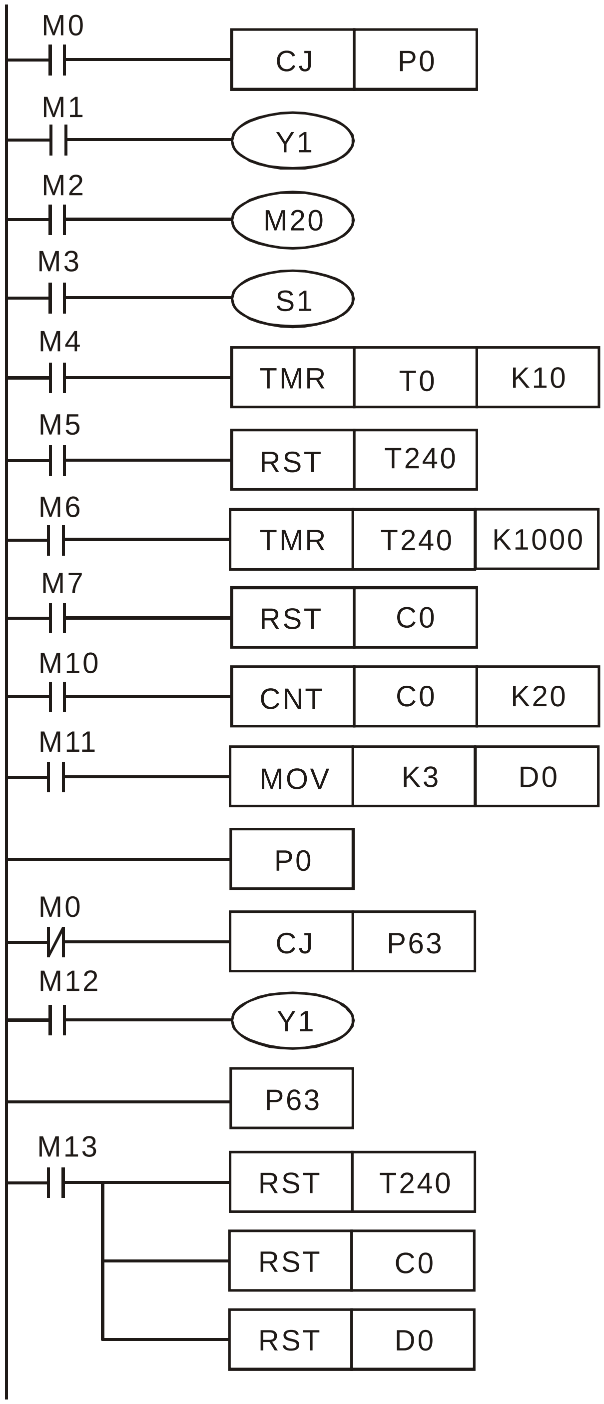
Example 2

◆ States of devices

Device	States of contacts before the execution of CJ	States of contacts during the execution of CJ	States of output coils during the execution of CJ
Y devices, M devices, S devices	M1, M2, and M3 are OFF.	M1, M2, and M3 are turned from OFF to ON.	Y1[*1], M20, and S1 are OFF.
	M1, M2, and M3 are ON.	M1, M2, and M3 are turned from ON to OFF.	Y1[*1], M20, and S1 are ON.
10 millisecond timers	M4 is OFF.	M4 is turned from OFF to ON.	The timer T0 does not count.
	M4 is ON.	M4 is turned from ON to OFF.	The timer T0 stops counting immediately. When M0 is turned from ON to OFF, the timer T0 is reset to 0.
	M6 is OFF.	M6 is turned from OFF to ON.	The timer T240 does not count.
	M6 is ON.	M6 is turned from ON to OFF.	The timer T240 stops counting immediately. When M0 is turned from ON to OFF, the timer T240 is reset to 0.
C0~C234	M7 and M10 are OFF.	M10 is ON/OFF.	The counter C0 does not count.
	M7 is OFF. M10 is ON/OFF.	M10 is ON/OFF.	C0 stops counting. After M0 is turned OFF, C0 will resume counting.
Applied instructions	M11 OFF	M11 is turned from OFF to ON.	The applied instructions are not executed.
	M11 ON	M11 is turned from ON to OFF.	The applied instructions which are skipped are not executed, but API 53~API 59 and API 157~API 159 are still executed,

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

- Y1 is a dual output. When M0 is OFF, Y1 is controlled by M1. When M0 is ON, Y1 is controlled by M12.



API		CALL		S	Calling a subroutine	Applicable model
01			P			20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z

- Note: S can be a pointer.
  S can be a pointer in the range of P0 to P255.
  A pointer can not be modified by a V device or a Z device.

16-bit instruction (3 steps)

CALL	Continuity instruction	CALLP	Pulse instruction

32-bit instruction

-	-	-	-

- Flag: None

Explanation

- **S**: Pointer which points to a subroutine
- The subroutine to which a pointer points should be written after M102, M2 and the instruction SRET.
- The pointer used by the instruction CALL can not be the same as the pointers used by the instructions CJ, CJN, and JMP.
- If only the instruction CALL is used, the same subroutine can be called repeatedly.

API	SRET		Indicating that a subroutine ends	Applicable model
02				20PM
				✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z

- Note: There is no operand.
  The instruction does not need to be driven by a contact.

16-bit instruction (1 step)

SRET	Continuity instruction	-	-

32-bit instruction

-	-	-	-

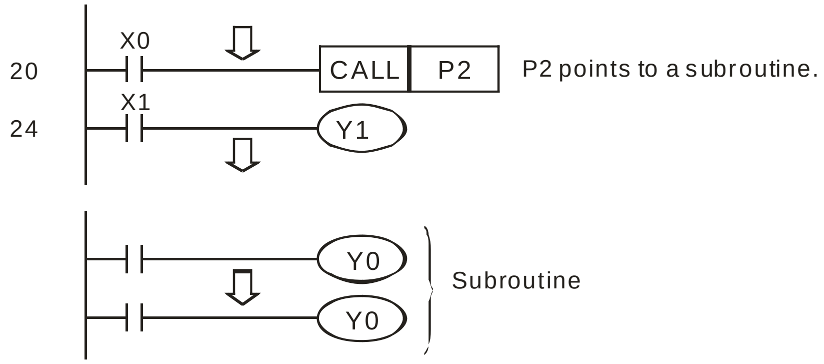
- Flag: None

Explanation

- The instruction SRET indicates that a soubroutine ends. After the execution of a subroutine in a program is complete, the instruction following CALL which calls the subroutine in the main program O100 will be executed.

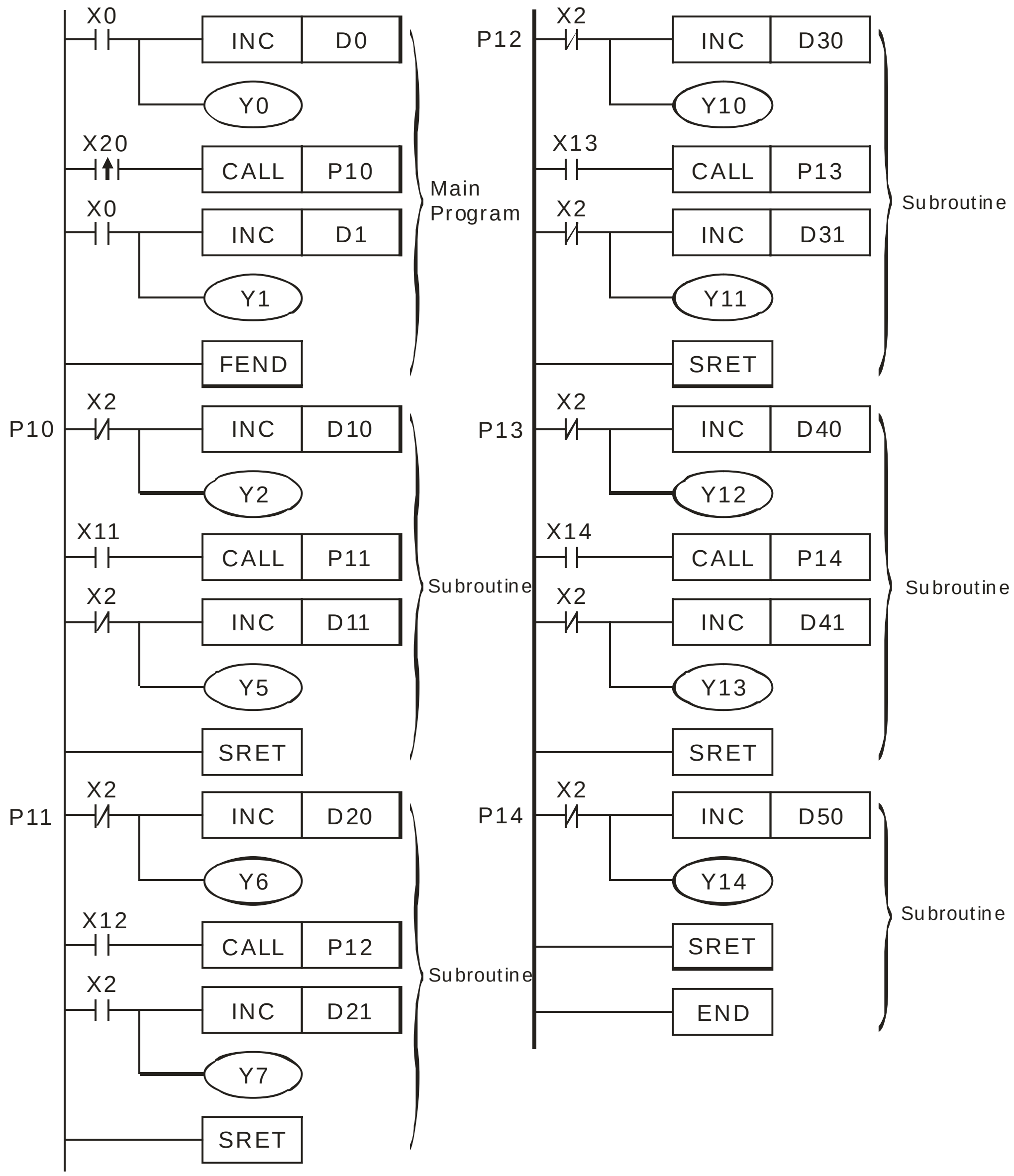
Example 1

- When X0 is ON, the instruction CALL is executed, and the execution of the program jumps to the subroutine to which P2 points. When the instruction SRET is executed, the execution of the program returns to address 24.



Example 2

- When X20 is turned from OFF to ON, the instruction CALL P10 is executed, and the execution of the program jumps to the subroutine to which P10 points.
- When X11 is ON, the instruction CALL P11 is executed, and the execution of the program jumps to the subroutine to which P11 points.
- When X12 is ON, the instruction CALL P12 is executed, and the execution of the program jumps to the subroutine to which P12 points.
- When X13 is ON, the instruction CALL P13 is executed, and the execution of the program jumps to the subroutine to which P13 points.
- When X14 is ON, the instruction CALL P14 is executed, and the execution of the program jumps to the subroutine to which P14 points. When the instruction SRET is executed, the execution of the program returns to the previous subroutine.
- When the instruction SRET in the subroutine to which P10 points is executed, the execution of the program returns to the main program.


X0
INC D0
Y0
X20
CALL P10
X0
INC D1
Y1
FEND
Main Program
P10
X2
INC D10
Y2
X11
CALL P11
X2
INC D11
Y5
SRET
Subroutine
P11
X2
INC D20
Y6
X12
CALL P12
X2
INC D21
Y7
SRET
Subroutine
P12
X2
INC D30
Y10
X13
CALL P13
X2
INC D31
Y11
SRET
Subroutine
P13
X2
INC D40
Y12
X14
CALL P14
X2
INC D41
Y13
SRET
Subroutine
P14
X2
INC D50
Y14
SRET
END
Subroutine

API				Applicable model
07	WDT		Watchdog timer	20PM
				✓

	Bit device				Word device													
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z

- Note: There is no operand.
  The instruction does not need to be driven by a contact.

16-bit instruction (1 step)

WDT	Continuity instruction	WDTP	Pulse instruction

32-bit instruction

-	-	-	-

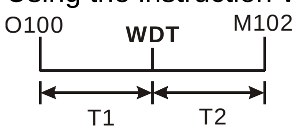
- Flag: None

Explanation

◆ The instruction WDT is used to reset the watchdog timer in a DVP-10PM series motion controller. If the scan time in a DVP-10PM series motion controller exceeds 200 milliseconds, the ERROR LED indicator of the motion controller will be ON, and users will have to disconnect the motion control module. After the users connect the motion controller again, the motion controller will judge its state according to the setting of the "STOP/RUN switch" switch. If there is no "STOP/RUN switch" switch, the motion controller will stop running automatically.

◆ The points when a watchdog timer acts are as follows.

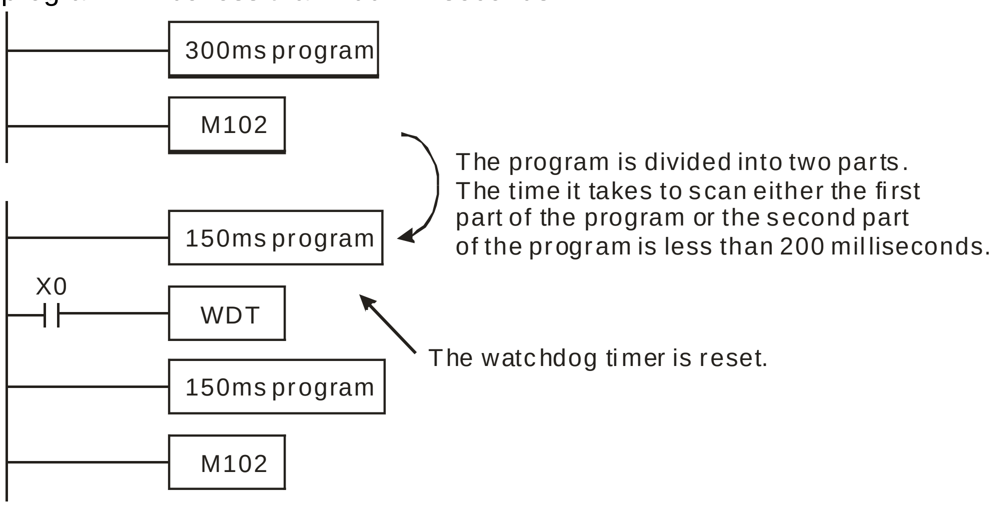
- The system is abnormal.
- The execution of a program takes much time, and therefore the scan time is greater than the setting value in D1000. There are two ways users can use to improve the situation.
  1. Using the instruction WDT



  2. Changing the value in D1000 (The default setting is 200 milliseconds.)

Example

◆ Suppose the scan time is 300 milliseconds. After the program is divided into two parts, and the instruction WDT is inserted between these two parts, the time it takes to scan either the first part of the program or the second part of the program will be less than 200 milliseconds.



Additional remark

◆ The instruction WDT is executed when a condition is met. Users can make the instruction WDT executed only in one scan cycle by writing a program. They can use the pulse instruction WDTP.

◆ The default setting of a watchdog timer is 200 milliseconds. Users can set a watchdog timer by means of D1000.

API						Applicable model
08		RPT		(S)	Start of a nested loop	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*	*	*	*	*	*	*	*	*	*	

- Note: The instruction does not need to be driven by a contact.
  The instruction supports V devices.
  Please refer to specifications for more information about device ranges.

16-bit instruction (3 steps)

RPT	Continuity instruction	-	-

32-bit instruction

-	-	-	-

- Flag: None

Explanation

- **S**: Number of times a loop is executed
- There is only one RPT-RPE loop in a program. If there is more than one RPT-RPE loop in a program, an error will occur.

API						Applicable model
09		RPE			End of a nested loop	20PM
						✓

	Bit device				Word device											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z

- Note: There is no operand.
  The instruction does not need to be driven by a contact.

16-bit instruction (1 step)			
RPE	Continuity instruction	-	-

32-bit instruction			
-	-	-	-

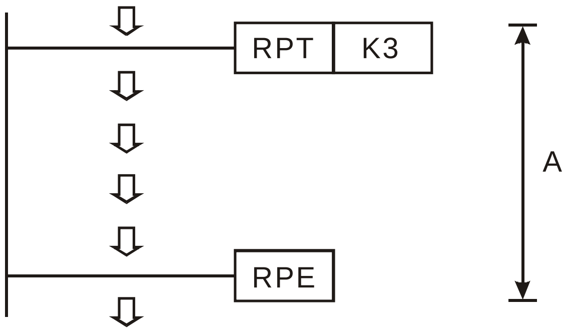
- Flag: None

Explanation

- RPT in a program specifies that the RPT-RPE loop in the program must be executed N times.
- N is in the range of K1 to K32,767. If N≦K1, N will be regarded as K1.
- Users can skip the execution of the RPT-RPE loop in a program by means of the instruction CJ.
- An error will occur if
  - the instruction RPE is before the instruction RPT.
    there is RPT, but there is no RPE.
  - the number of times RPT is used is not the same as the number of times RPE is used.
- There is only one RPT-RPE loop in a program. If there is more than one RPT-RPE loop in a program, an error will occur.

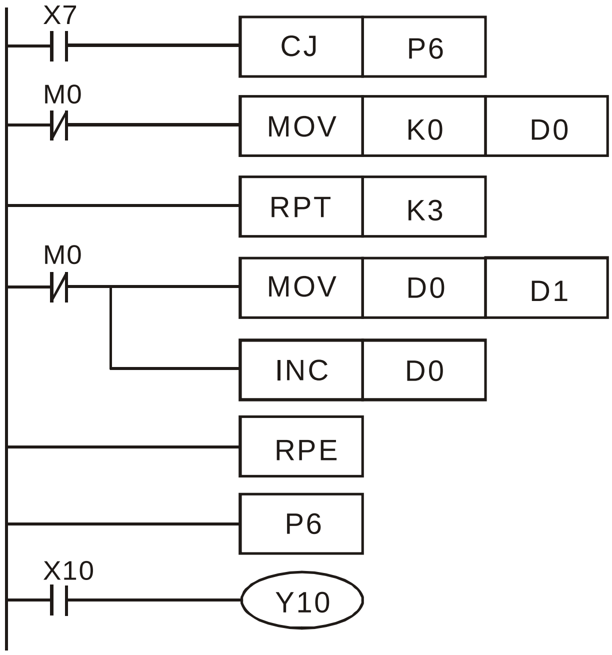
Example 1

- Part A can be executed three times by means of a RPT-RPE loop.



Example 2

- When X0.7 is OFF, the program between RPT and RPE is executed. When X0.7 is ON, the instruction CJ is executed, the subroutine to which P6 points is executed, and the program between RPT and RPE is skipped.



API						Applicable model
10	D	CMP	P	S1 S2 D	Comparing values	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*
D		*	*	*											

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

CMP	Continuity instruction	CMPP	Pulse instruction

32-bit instruction (9 steps)

DCMP	Continuity instruction	DCMPP	Pulse instruction

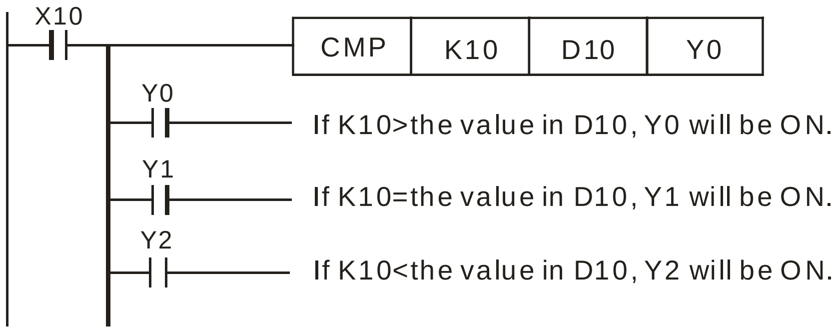
- Flag: None

Explanation

- **S_1**: Comparison value 1; **S_2**: Comparison value 2; **D**: Comparison result
- The instruction is used to compare the value in **S_1** with that in **S_2**. The comparison result is stored in **D**.
- The operand **D** occupies three consecutive devices.
- If the operand **D** is Y0, Y0, Y1, and Y2 will be occupied automatically.

Example

- When X10 is ON, the instruction CMP is executed, and Y0, Y1, or Y2 is ON. When X10 is OFF, the execution of the instruction CMP stops, and the states of Y0, Y1, and Y2 remain unchanged.
- If users want to get the result that K10≧ the value in D10, they have to connect Y0 and Y1 in series. If users want to get the result that K10≦ the value in D10, they have to connect Y1 and Y2 in series. If users want to get the result that K10≠the value in D10, they have to connect Y0, Y1, and Y2 in series.



API						
11	D	ZCP	P	S1 S2 S D	Zonal comparison	Applicable model: 20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*
S					*	*	*	*	*	*	*	*	*	*	*
D		*	*	*											

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (9 steps)
ZCP Continuity instruction ZCPP Pulse instruction

32-bit instruction (12 steps)
DZCP Continuity instruction DZCPP Pulse instruction

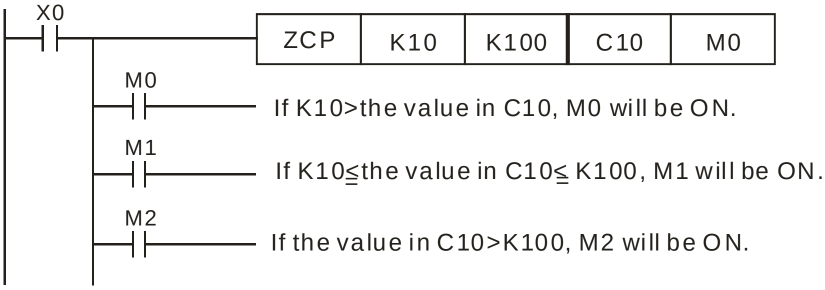
- Flag: None

Explanation

- **S_1**: Minimum value; **S_2**: Maximum value; **S**: Comparison value; **D**: Comparison result
- The instruction is used to compare the value in **S** with that in **S_1**, and compare the value in **S** with that in **S_2**. The comparison result is stored in **D**.
- The value in **S_2** must be greater than that in **S_1**.
- The operand **D** occupies three consecutive devices.

Example

- If the operand **D** is M0, M0, M1, and M2 will be occupied automatically.
- When X0 is ON, the instruction ZCP is executed, and M0, M1, or M2 is ON. When X0 is OFF, the execution of the instruction ZCP stops, and the states of M0, M1, and M2 remain unchanged.



API						Applicable model
12	D	MOV	P	S D	Transferring a value	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)
MOV Continuity instruction MOVP Pulse instruction

32-bit instruction (6 steps)
DMOV Continuity instruction DMOVP Pulse instruction

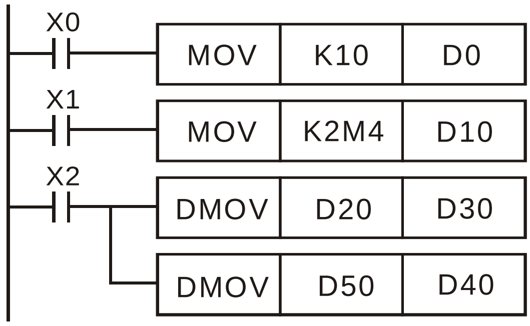
- Flag: None

Explanation

- **S**: Source; **D**: Destination
- When the instruction is executed, the value in **S** is transferred to **D**. When the instruction is not executed, the value in **D** is unchanged.
- If an operation result gotten is a 32-bit value, users can only move the operation result by means of the instruction DMOV.

Example

- If users want to move a 16-bit value, they have to use the instruction MOV.
  1. When X0 is OFF, the value in D0 is unchanged. When X0 is ON, the value K10 is transferred to the data register D0.
  2. When X1 is OFF, the value in D10 is unchanged. When X1 is ON, the value in K2M4 is transferred to the data register D10.
- If users want to move a 32-bit value, they have to use the instruction DMOV.
  When X2 is OFF, the values in (D31, D30) and (D41, D40) are unchanged. When X2 is ON, the value in (D21, D20) is transferred to (D31, D30), and the value in (D51, D50) is transferred to (D41, D40).



API			Operands	Function	Applicable model
13	SMOV	P	S m1 m2 D n	Transferring digits	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							*	*	*	*	*	*	*	*	*
m1					*	*									
m2					*	*									
D								*	*	*	*	*	*	*	*
n					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (11 steps)

SMOV Continuity instruction SMOVP Pulse instruction

32-bit instruction

– – – –

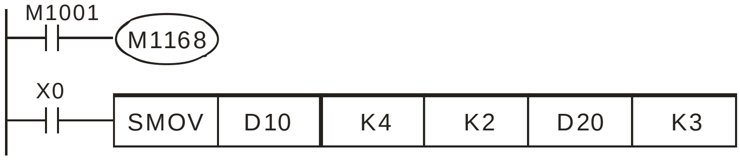
- Flag: M1168

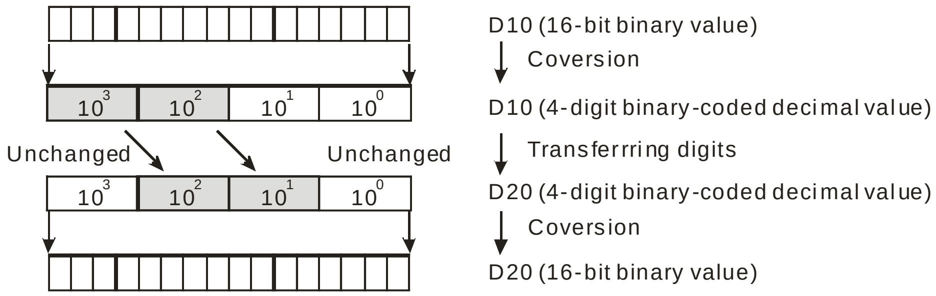
Explanation

- **S**: Data source; $\mathbf{m_1}$: Start digit which will be transferred from the source device; $\mathbf{m_2}$: Number of digits which will be transferred; **D**: Data destination; **n**: Start digit where the source data is stored in the destination device
- The value used by the instruction is a binary-coded decimal value (M1168 is OFF).
  The value used by SMOV is a binary-coded decimal value. When the instruction is executed, the $\mathbf{m_2}$ digits of the four-digit binary-code decimal value in **S** which start from the $\mathbf{m_1}^{th}$ digit of the four-digit binary-code decimal value in **S** are transferred to the $\mathbf{m_2}$ digits of the four-digit binary-code decimal value in **D** which starts from the $\mathbf{n}^{th}$ digit of the four-digit binary-code decimal value in **D**.
- The value used by the instruction is a binary value (M1168 is ON).
  When the instruction is executed, the $\mathbf{m_2}$ digits of the four-digit decimal value in **S** which start from the $\mathbf{m_1}^{th}$ digit of the four-digit decimal value in **S** are transferred to the $\mathbf{m_2}$ digits of the four-digit decimal value in **D** which starts from the $\mathbf{n}^{th}$ digit of the four-digit decimal value in **D**.
- $\mathbf{m_1}$ is in the range of 1 to 4.
- $\mathbf{m_2}$ is in the range of 1 to $\mathbf{m_1}$. (It can not be greater than $\mathbf{m_1}$.)
- **n** is in the range of $\mathbf{m_2}$ to 4. (It can not be less than $\mathbf{m_2}$.)

Example 1

- When M1168 is OFF, the value used by SMOV is a binary-coded decimal value. When X0 is ON, the two digits of the decimal value in D10 which start from the fourth digit of the decimal value (the digit in the thousands place of the decimal value) in D10 are transferred to the two digits of the decimal value in D20 which start from the third digit of the decimal value (the digit in the hundreds place of the decimal value) in D20. After the instruction is executed, the digits in the thousands place of the decimal value (10^3) and the ones place of the decimal value (10^0) in D20 will be unchanged.
- If the binary-coded decimal value used is not in the range of 0 to 9,999, an operation error will occur, the instruction will not executed, M1067 will be ON, and the error code in D1067 will be 0E18 (hexadecimal value).

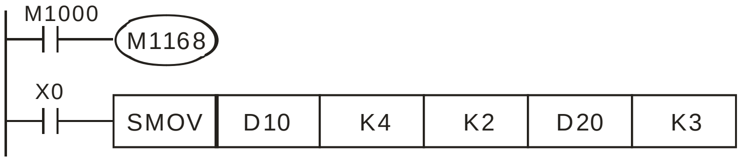


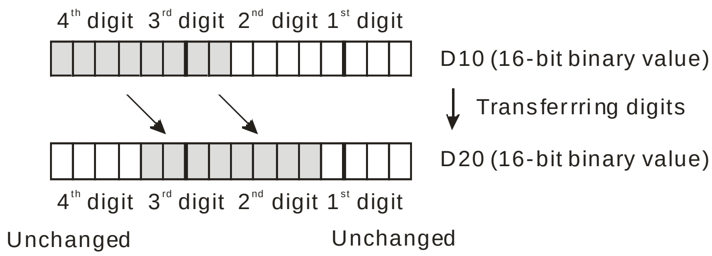


- Suppose the value in D10 is K1234, and the value in D20 is K5678. After the instruction is executed, the value in D10 will be unchanged, and the value in D20 is K5128.

Example 2

- When M1168 is ON, the value used by SMOV is a is binary value. When the instruction SMOV is executed, the binary values in D10 and D20 are not converted into the binary-coded decimal values, and evey digit which is transferred is composed of four bits.

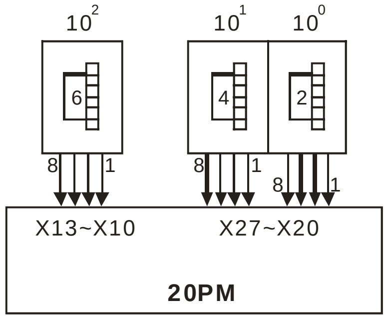


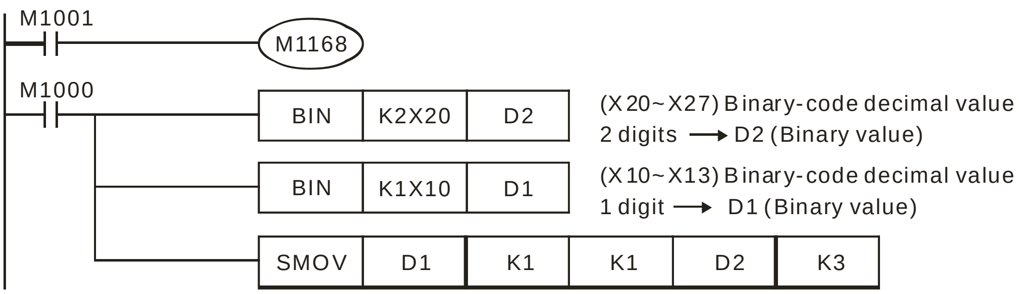


- Suppose the value in D10 is H1234, and the value in D20 is H5678. After the instruction is executed, the value in D10 will be unchanged, and the value in D20 is H5128.

Example 3

- The two digits of the value of the DIP switch on the right are transferred to the the two digits of the value in D2 which start from the second digit of the value in D2, and the one digit of the value of the DIP switch on the left is transferred to the the first digit of the value in D1. The instruction SMOV can be used to transfer the first digit of the value in D1 to the third digit of the value in D2. In other words, the two DIP switches can be combined into one DIP switch by means of the instruction SMOV.





API						Applicable model
14	D	CML	P	S D	Inverting bits	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (5 steps)

CML Continuity instruction CMLP Pulse instruction

32-bit instruction (6 steps)

DCML Continuity instruction DCMLP Pulse instruction

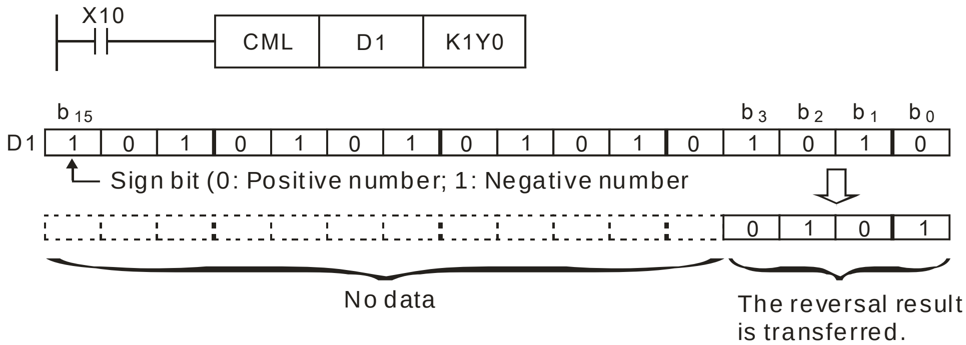
- Flag: None

Explanation

- **S**: Source; **D**: Destination
- The instruction is used to invert the bits in **S** (0→1 and 1→0), and transfer the inversion result to **D**.

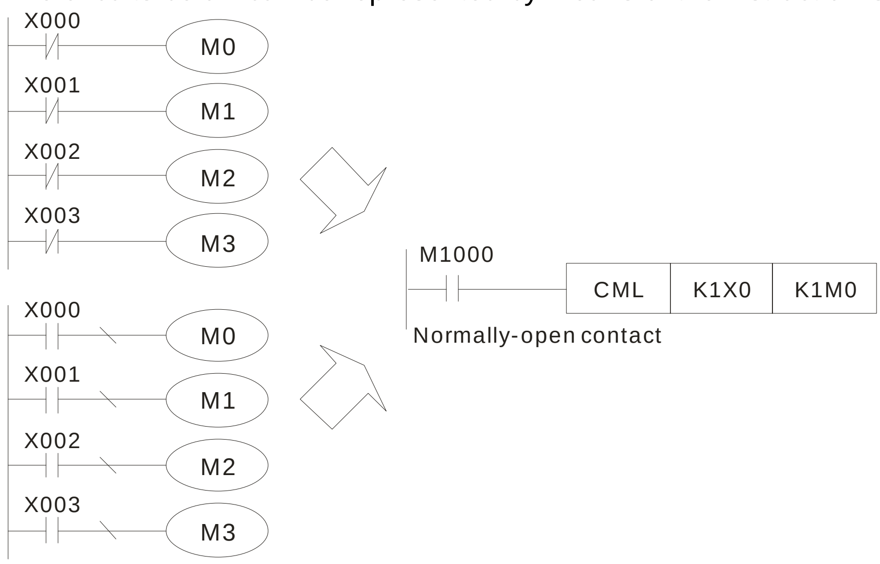
Example 1

- When X10 is ON, bit 0~bit 3 in D1 are inverted, and the inversion result is transferred to Y0~Y3.



Example 2

- The circuits below can be represented by means of the instruction CML.



API	BMOV	P	(S) (D) (n)	Transferring values	Applicable model
15					20PM
					✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							*	*	*	*	*	*	*		
D								*	*	*	*	*	*		
n					*	*					*	*	*		

- Note: Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (7 steps)

BMOV Continuity instruction BMOVP Pulse instruction

32-bit instruction

\- - - -

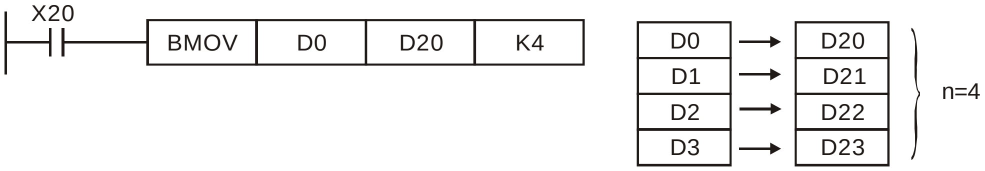
- Flag: None

Explanation

- **S**: Source; **D**: Destination; **n**: Length
- The instruction is used to transfer the values in registers to new registers. The values in the **n** registers starting from **S** are transferred to the **n** registers starting from **D**. If **n** is not in the range available, only the values in registers available will be transferred.
- **n** is in the range of 1 to 512.

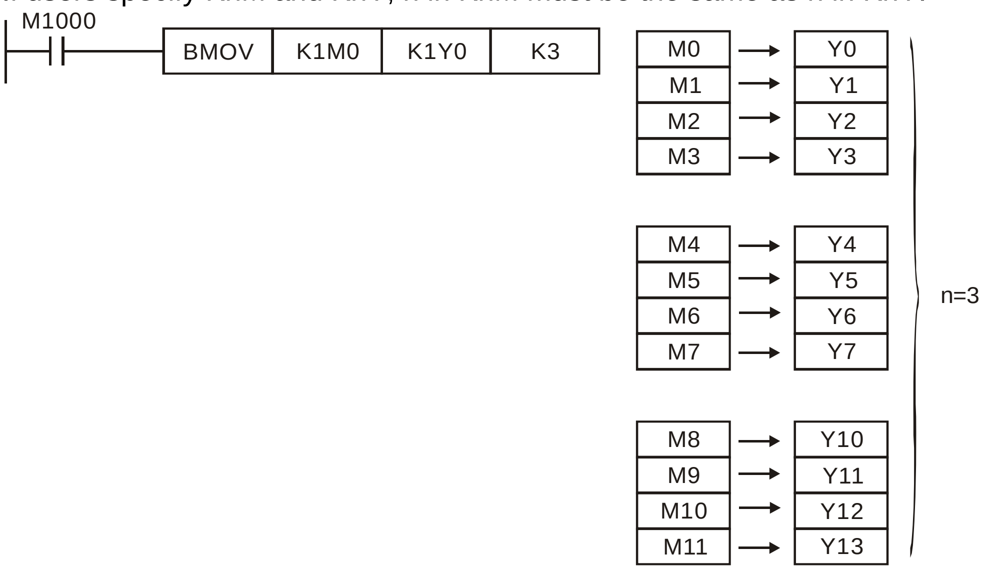
Example 1

- When X2.0 is ON, the values in D0~D3 are transferred to D20~D23.



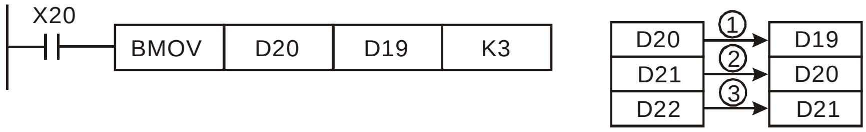
Example 2

- If users specify KnM and KnY, n in KnM must be the same as n in KnY.

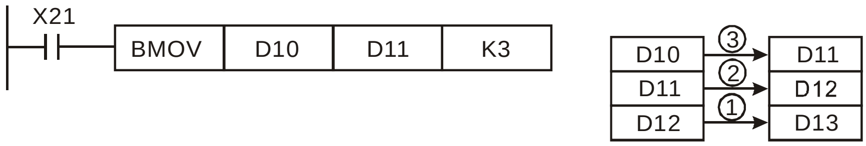


Example 3

- In order to prevent the error which results from the overlap between source devices and destination devices, the values in the source devices are transferred in the following way.
  1. The device number of **S** is greater than the device number of **D**. The values in D20~D22 are transferred in the order ①→②→③.



  2. The device number of **S** is less than the device number of **D**. The values in D10~D12 are transferred in the order ③→②→①. The values in D11~D13 are the same as the value in D10.



API					Applicable model
16		FMOV	P	(S) (D) (n)	Transferring a value to several devices	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*		
n					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (7 steps)
FMOV Continuity instruction FMOVP Pulse instruction

32-bit instruction
DFMOV Continuity instruction DFMOVP Pulse instruction

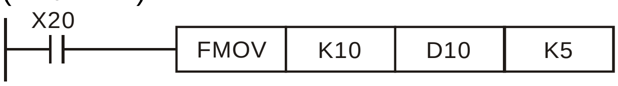
- Flag: None

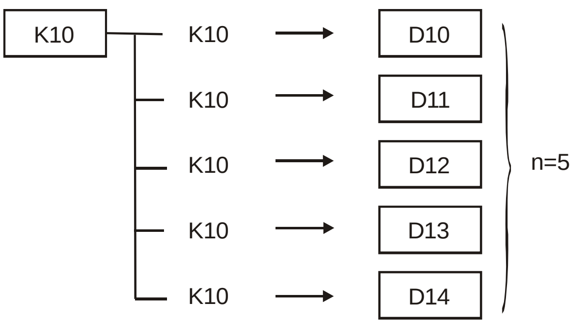
Explanation

- **S**: Source; **D**: Destination; **n**: Length
- The value in **S** is transferred to the n registers starting from **D**. If **n** is not in the range available, a value will only be transferred to registers available.
- **n** is in the range of 1 to 512.

Example

- When X20 is ON, K10 is transferred to the 5 registers starting from D10 (D10~D14).





API						Applicable model
17	D	XCH	P	D1 D2	Interchanging values	20PM ✓

	Bit device				Word device									
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D_1								*	*	*	*	*	*	*	*
D_2								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)

  Please refer to specifications for more information about device ranges.

  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (5 steps)

XCH Continuity instruction | XCHP Pulse instruction

32-bit instruction (9 steps)

DXCH Continuity instruction | DFXCHP Pulse instruction

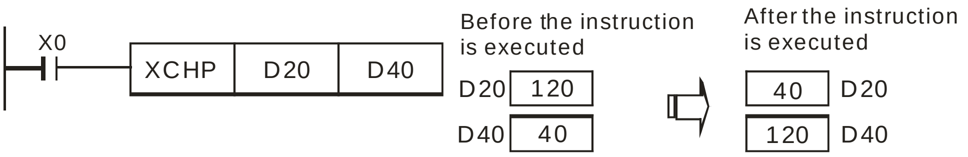
- Flag: None

Explanation

- D_1: Value which is interchanged; D_2: Value which is interchanged
- The instruction is used to interchange the value in D_1 with the value in D_2.
- It is suggested that users should use the pulse instruction XCHP.

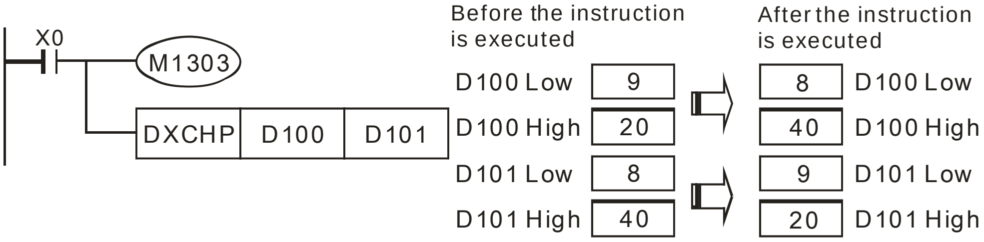
Example

- When X0 is turned from OFF to ON, the value in D20 is interchanged with the value in D40.



Additional remark

- 16-bit instruction: If D_1 is the same as D_2, and M1303 is ON, the high 8 bits are interchanged with the low 8 bits.
- 32-bit instruction: If D_1 is the same as D_2, and M1303 is ON, the high 16 bits are interchanged with the low 16 bits.
- When X0 is ON, and M1303 is ON, the high 8 bits in D100 are interchanged with the high 8 bits in D101, and the low 8 bits in D100 are interchanged with the low 8 bits in D101.



API		BCD			Converting a binary value into a binary-coded decimal value	Applicable model
18	D	BCD	P	S D		20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)

BCD	Continuity instruction	BCDP	Pulse instruction

32-bit instruction (6 steps)

DBCD	Continuity instruction	DBCDP	Pulse instruction

- Flags

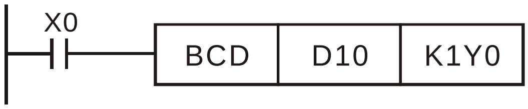
Ox	O100	
M1793	M1953	Operation error flag

Explanation

- ◆ The binary value in **S** is converted into a binary-coded decimal value, and the conversion result is transferred to **D**.
- ◆ If a binary value is converted to a binary-coded decimal value which is not in the range of 0 to 9,999, the instruction BCD will not be executed. If a binary value is converted to a binary-coded decimal value which is not in the range of 0 to 99,999,999, the instruction DBCD will not be executed.
- ◆ BCD can be used to convert the binary value in a positioning unit to a binary-coded decimal value, and transfer the conversion result to an external device, e.g. a seven-segment display.

Example

- ◆ When X0 is ON, the binary value in D10 is converted into a binary-coded decimal value, and the digit in the ones place of the conversion result is stored in K1Y0 (Y0~Y3).



If D10=001E (hexadecimal value)=0030 (decimal value), Y0~Y3=0000 (binary value).

API	W	BIN		(S) (D)	Converting a binary-coded decimal value into a binary value	Applicable model
19	D		P			20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)

BIN	Continuity instruction	BINP	Pulse instruction

32-bit instruction (6 steps)

DBIN	Continuity instruction	DBINP	Pulse instruction

- Flags

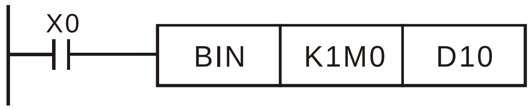
Ox	O100	
M1793	M1953	Operation error flag

Explanation

- **S**: Source; **D**: Conversion result
- The binary-coded decimal value in **S** is converted into a binary value, and the conversion result is transferred to **D**.
- The 16-bit binary-coded decimal value in **S** should be in the range of 0 to 9,999, and the 32-bit binary-coded decimal value in **S** should be in the range of 0 to 99,999,999.
- Decimal constants and hexadecimal constants are converted into binary values automatically. Users do not need to use the instruction.

Example

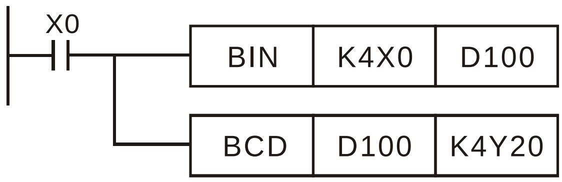
- When X0 is ON, the binary-coded decimal value in K1M0 is converted into a binary value, and the conversion result is stored in D10.



Additional remark

- Applications of the instructions BCD and BIN:

1. If a DVP-10PM series motion controller wants to read a binary-coded decimal value created by a DIP switch, users have to use the instruction BIN to convert the value into a binary value, and store the conversion result in the DVP-10PM series motion controller.
2. If users want to display a value stored in a DVP-10PM series motion controller on a seven-segment display on which binary-coded decimal values can be displayed, they have to use the instruction BCD to convert the value into a binary-coded decimal value, and transfer the conversion result to the seven-segment display.
3. When X0 is ON, the binary-coded decimal value in K4M0 is converted into a binary value, and the conversion result is stored in D100. Subsequently, the binary value in D100 is converted into a binary-coded decimal value, and the conversion result is stored in K4Y20.



API		ADD				Applicable model
20	D	ADD	P	S1 S2 D	Binary addition	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

ADD	Continuity instruction	ADDP	Pulse instruction

32-bit instruction (9 steps)

DADD	Continuity instruction	DADDP	Pulse instruction

- Flags

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

- Please refer to the additional remark below.

Explanation

- **S_1**: Augend; **S_2**: Addend; **D**: Sum
- The binary value in **S_2** is added to the binary value in **S_1**, and the sum is stored in **D**.
- The highest bit in **S_1** and the highest bit in **S_2** are sign bits. If the sign bit in a register is 0, the value in the register is a positive value. If the sign bit in a register is 1, the value in the register is a negative value.
- The flags related to 16-bit binary addition and 32-bit binary addition are listed below.

  16-bit binary addition:

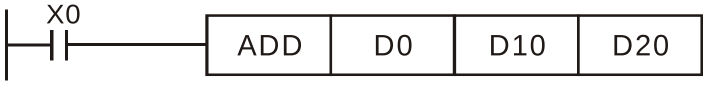
  1. If the operation result gotten is 0, a zero flag will be ON.
  2. If the operation result gotten is less than – 32,768, a borrow flag will be ON.
  3. If the operation result gotten is greater than 32,767, a carry flag will be ON.

  32-bit binary addition:

  1. If the operation result gotten is 0, a zero flag will be ON.
  2. If the operation result gotten is less than – 2,147,483,648, a borrow flag will be ON.
  3. If the operation result gotten is greater than 2,147,483,647, a carry flag will be ON.

Example 1

- 16-bit binary addition: When X0 is ON, the addend in D10 is added to the augend in D0, and the sum is stored in D20.

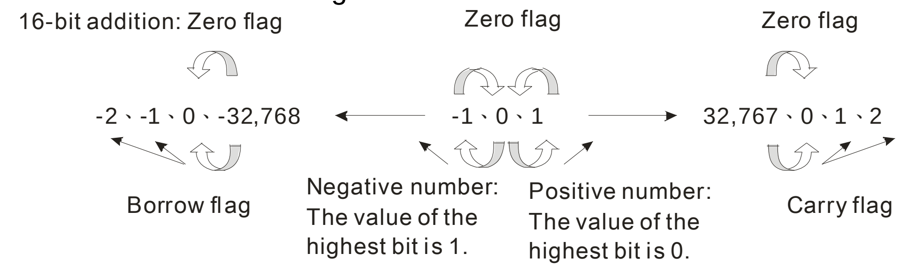


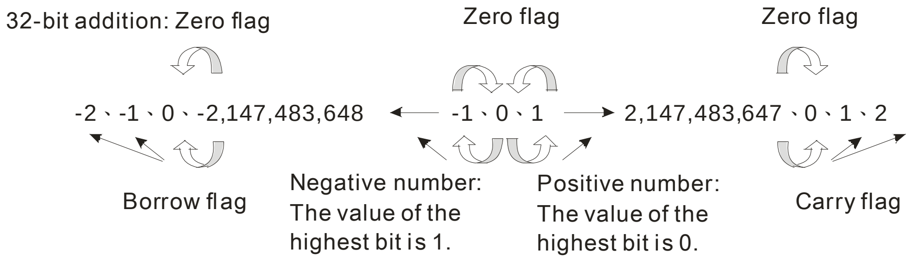
Example 2

- 32-bit binary addition: When X1 is ON, the value in (D41, D40) is added to the augend in (D31, D30), and the sum is stored in (D51, D50).

Additional remark

◆ The relations between flags and values are shown below.





API						Applicable model
						20PM
21	D	SUB	P	S1 S2 D	Binary subtraction	✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

SUB Continuity instruction SUBP Pulse instruction

32-bit instruction (9 steps)

DSUB Continuity instruction DSUBP Pulse instruction

- Flags

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

- Please refer to the additional remark below.

Explanation

- **S_1**: Minuend; **S_2**: Subtrahend; **D**: Difference
- The binary value in **S_2** is subtracted from the binary value in **S_1**, and the difference is stored in **D**.
- The highest bit in **S_1** and the highest bit in **S_2** are sign bits. If the sign bit in a register is 0, the value in the register is a positive value. If the sign bit in a register is 1, the value in the register is a negative value.
- The flags related to 16-bit binary subtraction and 32-bit binary subtraction are listed below.

  16-bit binary subtraction:

  1. If the operation result gotten is 0, a zero flag will be ON.
  2. If the operation result gotten is less than – 32,768, a borrow flag will be ON.
  3. If the operation result gotten is greater than 32,767, a carry flag will be ON.

  32-bit binary subtraction:

  1. If the operation result gotten is 0, a zero flag will be ON.
  2. If the operation result gotten is less than – 2,147,483,648, a borrow flag will be ON.
  3. If the operation result gotten is greater than 2,147,483,647, a carry flag will be ON.
- Please refer to the additional remark on the instruction ADD for more information about the relations between flags and values.

Example 1

- 16-bit binary subtraction: When X0 is ON, the subtrahend in D10 is subtracted from the minuend in D0, and the difference is stored in D20.

Example 2

- When X1 is ON, the subtrahend in (D41, D40) is subtracted from the minuend in (D31, D30), and the difference is stored in (D51, D50).

API					
22	D	MUL	P	S1 S2 D	Binary multiplication

Applicable model
20PM
✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*
D											*	*	*		

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

MUL	Continuity instruction	MULP	Pulse instruction

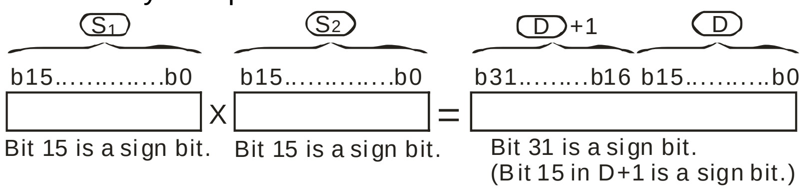
32-bit instruction (9 steps)

DMUL	Continuity instruction	DMULP	Pulse instruction

- Flag: None

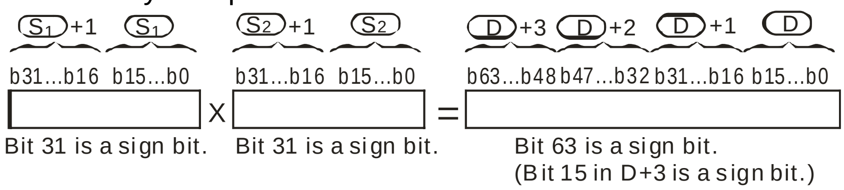
Explanation

- $\mathbf{S_1}$: Multiplicand; $\mathbf{S_2}$: Multiplier; **D**: Product
- The signed binary value in $\mathbf{S_1}$ is multiplied by the singed binary value in $\mathbf{S_2}$, and the product is stored in **D**. Users have to notice the sign bits in $\mathbf{S_1}$, $\mathbf{S_2}$, and **D** when 16-bit binary multiplication or 32-bit binary multiplication is done.
- 16-bit binary multiplication



Sign bit=0 (Positive sign); Sign bit=1 (Negative sign)

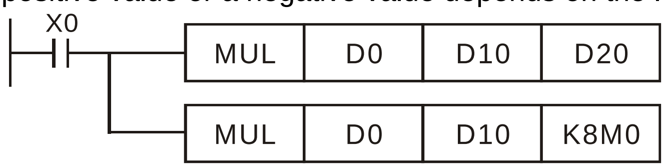
- 32-bit binary multiplication



Sign bit=0 (Positive sign); Sign bit=1 (Negative sign)

Example

- The 16-bit value in D0 is multiplied by the 16-bit value in D10, and the 32-bit product is stored in (D21, D20). The bits in D21 is the high 16 bits in (D21, D20), whereas the bits in D20 is the low 16 bits in (D21, D20). Whether the product is a positive value or a negative value depends on the leftmost bit in (D21, D20).



API						
23	D	DIV	P	S1 S2 D	Binary division	Applicable model: 20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*
D											*	*	*		

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

DIV	Continuity instruction	DIVP	Pulse instruction

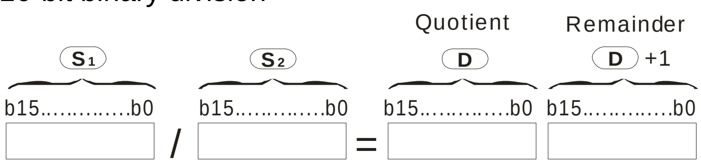
32-bit instruction (9 steps)

DDIV	Continuity instruction	DDIVP	Pulse instruction

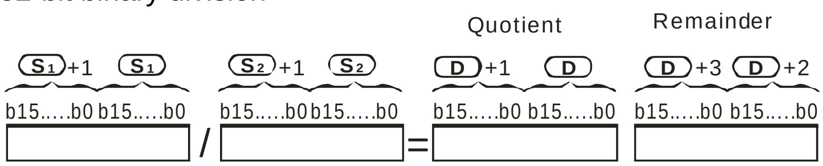
- Flag: None

Explanation

- **S_1**: Dividend; **S_2**: Divisor; **D**: Quotient and remainder
- The singed binary value in **S_1** is divided by the signed binary value in **S_2**. The quotient and the remainder are stored in **D**. Users have to notice the sign bits in **S_1**, **S_2**, and **D** when 16-bit binary division or 32-bit binary division is done.
- If the divisor in **S_2** is 0, the instruciton will not be executed.
- 16-bit binary division

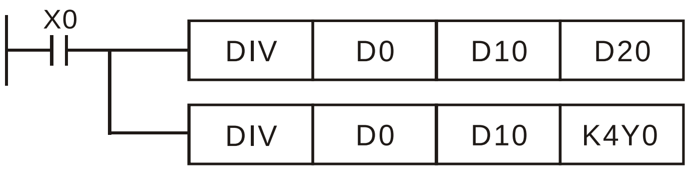


- 32-bit binary division



Example

- When X0 is ON, the dividend in D0 is divided by the divisor in D10, the quotient is stored in D20, and the remainder is stored in D21. Whether the quotient and the remainder are positive values or negative values depends on the leftmost bit in D20 and the leftmost bit in D21.



API						Applicable model
24	D	INC	P	(D)	Adding one to a binary value	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (3 steps)
INC Continuity instruction INCP Pulse instruction

32-bit instruction (3 steps)
DINC Continuity instruction DINCP Pulse instruction

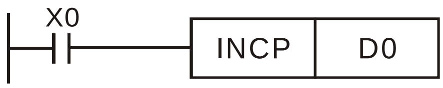
- Flag: None

Explanation

- **D**: Destination device
- If the instruction used is not a pulse instruction, the value in **D** used by the instruction increases by one whenever the instruction is executed.
- Generally, the pulse instructions INCP and DINCP are used.
- If a 16-bit operation is performed, 32,767 plus 1 equals -32,768. If a 32-bit operation is performed, 2,147,483,647 plus 1 equals -2,147,483,648.

Example

- When X0 is turned from OFF to ON, the value in D0 increases by one.



API						Applicable model
25	D	DEC	P	D	Subtracting one from a binary value	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (3 steps)

DEC	Continuity instruction	DECP	Pulse instruction

32-bit instruction (3 steps)

DDEC	Continuity instruction	DDECP	Pulse instruction

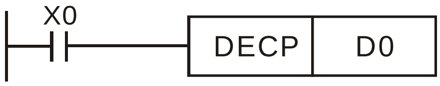
- Flag: None

Explanation

- **D**: Destination device
- If the instruction used is not a pulse instruction, the value in **D** used by the instruction decreases by one whenever the instruction is executed.
- Generally, the pulse instructions DECP and DDECP are used.
- If a 16-bit operation is performed, -32,768 minus 1 leaves 32,767. If a 32-bit operation is performed, -2,147,483,648 minus 1 leaves 2,147,483,647.

Example

- When X0 is turned from OFF to ON, the value in D0 decreases by one.



API						Applicable model
26	D	WAND	P	S1 S2 D	Logical AND operation	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

WAND Continuity instruction WANDP Pulse instruction

32-bit instruction (9 steps)

DWAND Continuity instruction DWANDP Pulse instruction

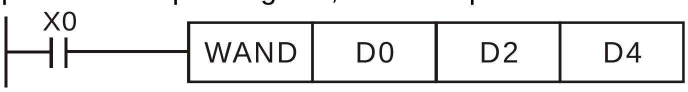
- Flag: None

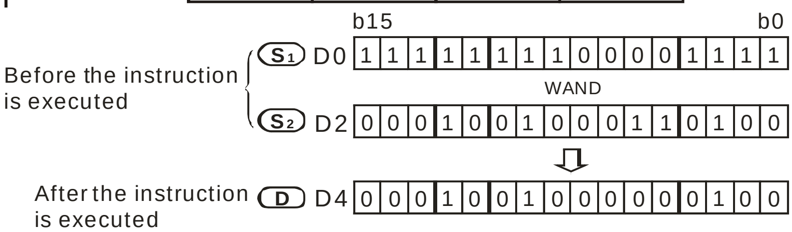
Explanation

- **S₁**: Source device 1; **S₂**: Source device 2; **D**: Operation result
- A logical AND operator takes the binary representations in **S₁** and **S₂**, and performs the logical AND operation on each pair of corresponding bits. The operation result is stored in **D**.
- The result in each position is 1 if the first bit is 1 and the second bit is 1. Otherwise, the result is 0.

Example 1

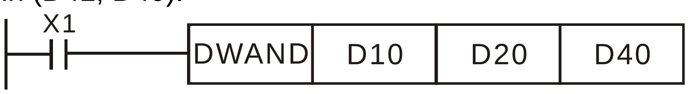
- When X0 is ON, a logical AND operator takes the values in the 16-bit device D0 and the 16-bit device D2, and performs the logical AND operation on each pair of corresponding bits, and the operation result is stored in D4.

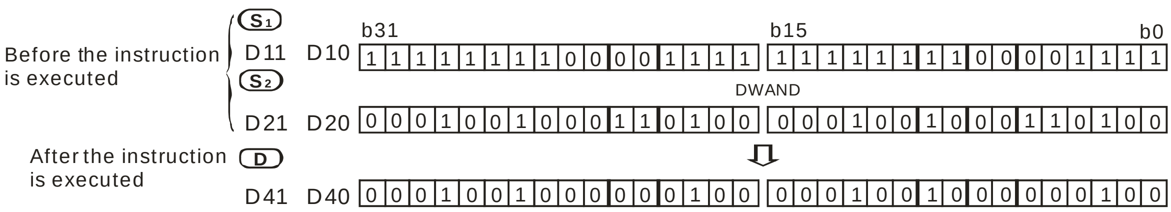




Example 2

- When X1 is ON, a logical AND operator takes the values in the 32-bit device (D11, D10) and the 32-bit device (D21, D20), and performs the logical AND operation on each pair of corresponding bits, and the operation result is stored in (D41, D40).





API						Applicable model
27	D	WOR	P	S1 S2 D	Logical OR operation	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)
WOR: Continuity instruction; WORP: Pulse instruction

32-bit instruction (9 steps)
DWOR: Continuity instruction; DWORP: Pulse instruction

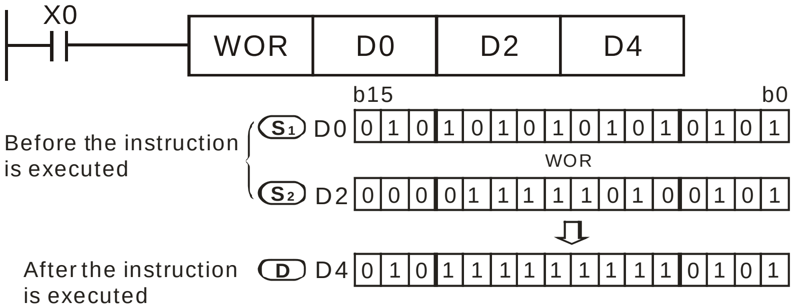
- Flag: None

Explanation

- **S_1**: Source device 1; **S_2**: Source device 2; **D**: Operation result
- A logical OR operator takes the binary representations in **S_1** and **S_2**, and performs the logical inclusive OR operation on each pair of corresponding bits. The operation result is stored in **D**.
- The result in each position is 1 if the first bit is 1, the second bit is 1, or both bits are 1. Otherwise, the result is 0.

Example 1

- When X0 is ON, a logical OR operator takes the values in the 16-bit device D0 and the 16-bit device D2, and performs the logical inclusive OR operation on each pair of corresponding bits, and the operation result is stored in D4.



Example 2

- When X1 is ON, a logical OR operator takes the values in the 32-bit device (D11, D10) and the 32-bit device (D21, D20), and performs the logical inclusive OR operation on each pair of corresponding bits, and the operation result is stored in (D41, D40).

API						Applicable model
28	D	WXOR	P	S1 S2 D	Logical exclusive OR operation	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					*	*	*	*	*	*	*	*	*	*	*
S2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

WXOR	Continuity instruction	WXORP	Pulse instruction

32-bit instruction (9 steps)

DWXOR	Continuity instruction	DWXORP	Pulse instruction

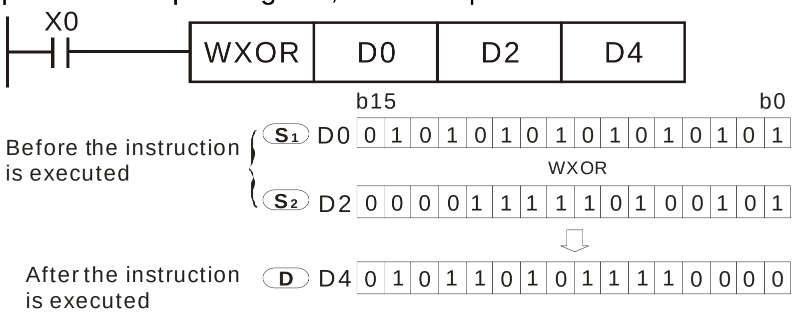
- Flag: None

Explanation

- **S1**: Source device 1; **S2**: Source device 2; **D**: Operation result
- A logical XOR operator takes the binary representations in **S1** and **S2**, and performs the logical exclusive OR operation on each pair of corresponding bits. The operation result is stored in **D**.
- The result in each position is 1 if the two bits are different, and 0 if they are the same.

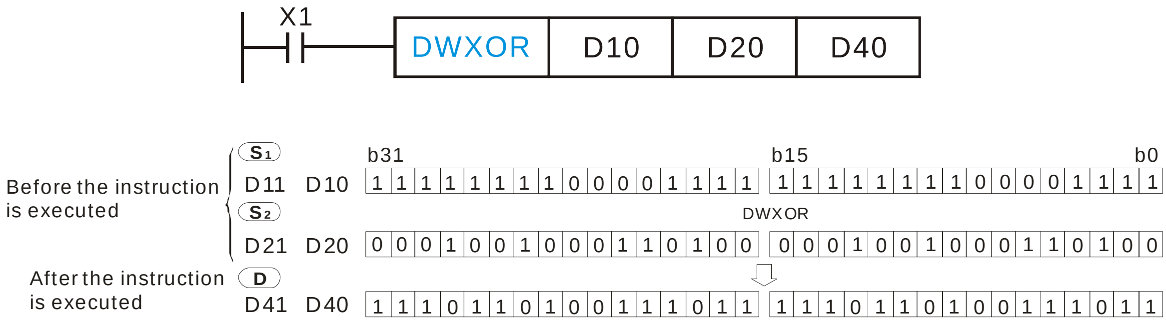
Example 1

- When X0 is ON, a logical XOR operator takes the values in the 16-bit device D0 and the 16-bit device D2, and performs the exclusive OR operation on each pair of corresponding bits, and the operation result is stored in D4.



Example 2

- When X1 is ON, a logical XOR operator takes the values in the 32-bit device (D11, D10) and the 32-bit device (D21, D20), and performs the logical exclusive OR operation on each pair of corresponding bits, and the operation result is stored in (D41, D40).



API	W	NEG		D	Taking the two's complement of a value	Applicable model
29	D		P			20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

NEG	Continuity instruction	NEGP	Pulse instruction

32-bit instruction (9 steps)

DNEG	Continuity instruction	DNEGP	Pulse instruction

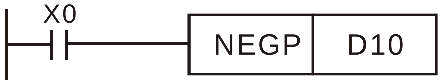
- Flag: None

Explanation

- **D**: Device in which the two's complement of the value in the device is stored
- The instructions can be used to convert a negative binary value into an absolute value.
- Generally, the pulse instructions NEGP and DNEGP are used.

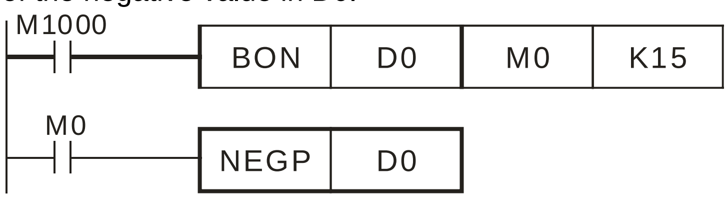
Example 1

- When X0 is turned from OFF to ON, all the bits in D0 are inverted (0 becomes 1, and 1 becomes 0), 1 is added to the result, and the final value is stored in the original register D10.



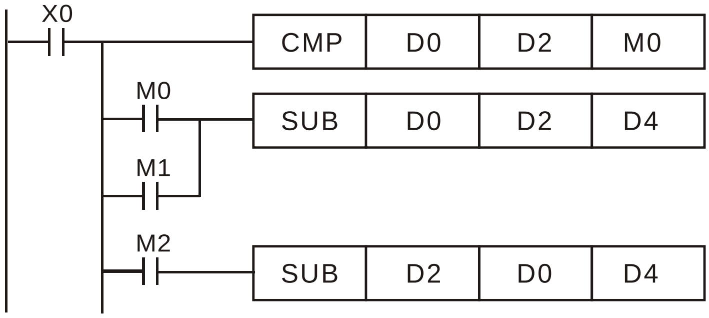
Example 2

- Getting the aboluste value of a negative value
  1. When bit 15 in D0 is 1, M0 is ON. (The value in D0 is a negative value.)
  2. When M0 is ON, the instruction NEG is used to take the two's complement of the negative value in D0.



Example 3

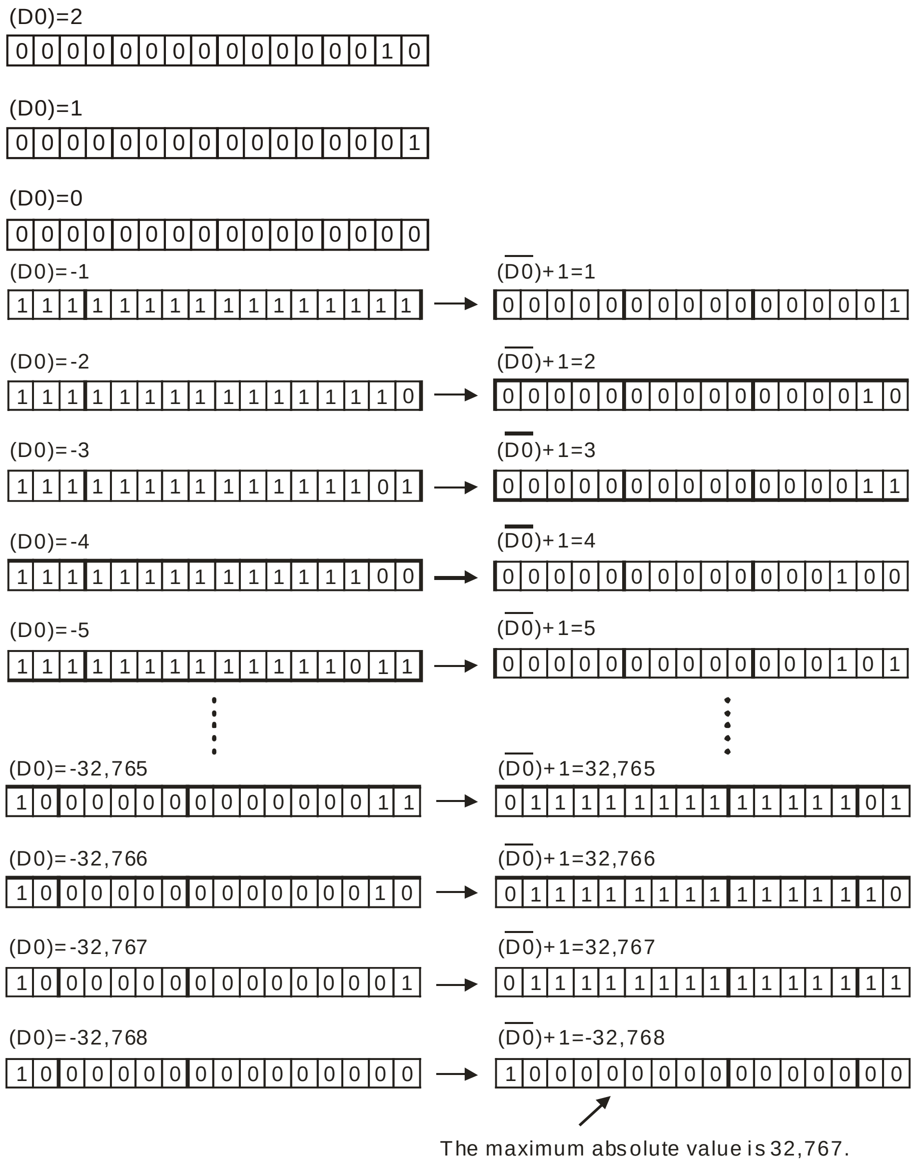
- Getting the absolute value of the difference between two values
  Suppose X0 is ON.
  1. When the value in D0 is greater than that in D2, M0 is ON.
  2. When the value in D0 is equal to that in D2, M1 is ON.
  3. When the value in D0 is less than that in D2, M2 is ON.
  4. The value in D4 is a positive value.



Additional remark

◆ The representation of a negative value and its absolute value are described below.

1. Whether the value in a register is a positive value or a negative value depends on the leftmost bit in the register. If the leftmost bit in a register is 0, the value in the register is a positive value. If the leftmost bit in a register is 1, the value in the register is a negative value.
2. The negative value in a register can be converted into its absolute value by means of the instruction NEG.



API						
30	D	ROR	P	(D) (n)	Rotating bits rightwards	Applicable model: 20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								*	*	*	*	*	*	*	*
N					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (5 steps)

ROR	Continuity instruction	RORP	Pulse instruction

32-bit instruction (9 steps)

DROR	Continuity instruction	DRORP	Pulse instruction

- Flags

Ox	O100	
M1810	M1970	Carry flag

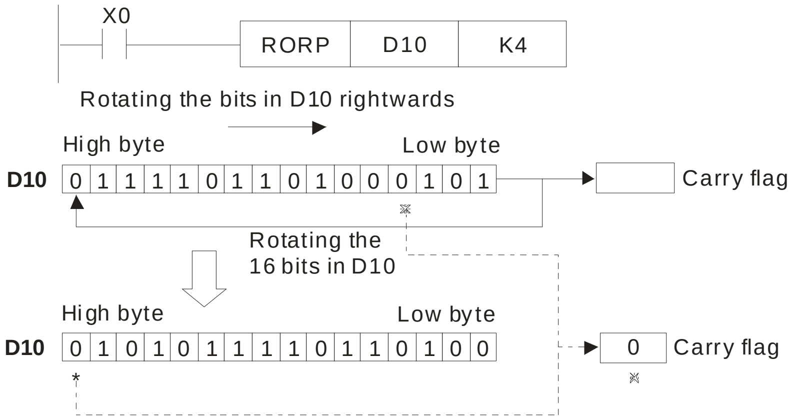
- Please refer to the additional remark below.

Explanation

- **D**: Device which is rotated; **n**: Number of bits forming a group
- The bits in **D** are divided into groups (**n** bits as a group), and these groups are rotated rightwards.
- The **n**th bit from the right is transmitted to a carry flag.
- Generally, the pulse instructions RORP and DRORP are used.
- If the operand **D** is KnY/KnM/KnS, Kn in KnY/KnM/KnS must be K4 (16 bits) or K8 (32 bits).
- 16-bit instruction: 1≤**n**≤16; 32-bit instruction: 1≤**n**≤32

Example

- When X0 is turned from OFF to ON, the bits in D10 are divided into groups (four bits as a group), and these groups are rotated rightwards. (The bit marked with ※ is transmitted to a carry flag.)



API		ROL		D n	Rotating bits leftwards	Applicable model
31	D		P			20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								*	*	*	*	*	*	*	*
n					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (5 steps)
ROL Continuity instruction | ROLP Pulse instruction

32-bit instruction (9 steps)
DROL Continuity instruction | DROLP Pulse instruction

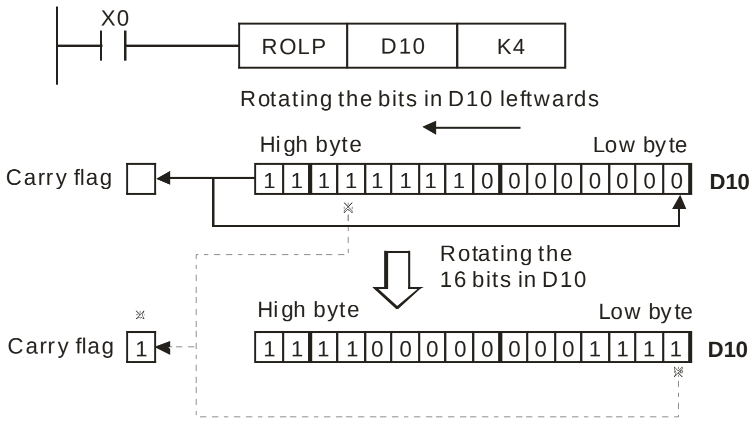
- Flags
  Ox M1810 | O100 M1970 | Carry flag
- Please refer to the additional remark below.

Explanation

- **D**: Device which is rotated; **n**: Number of bits forming a group
- The bits in **D** are divided into groups (**n** bits as a group), and these groups are rotated leftwards.
- The **n**th bit from the left is transmitted to a carry flag.
- Generally, the pulse instructions ROLP and DROLP are used.
- If the operand **D** is KY/KnM/KnS, Kn in KY/KnM/KnS must be K4 (16 bits) or K8 (32 bits).
- 16-bit instruction: 1≤**n**≤16; 32-bit instruction: 1≤**n**≤32

Example

- When X0 is turned from OFF to ON, the bits in D10 are divided into groups (four bits as a group), and these groups are rotated leftwards. (The bit marked with ※ is transmitted to a carry flag.)



API						Applicable model
32	D	RCR	P	(D) (n)	Rotating bits rightwards with a carry flag	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								*	*	*	*	*	*	*	*
n					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (5 steps)

RCR	Continuity instruction	RCRP	Pulse instruction

32-bit instruction (9 steps)

DRCR	Continuity instruction	DRCRP	Pulse instruction

- Flags

Ox	O100	
M1810	M1970	Carry flag

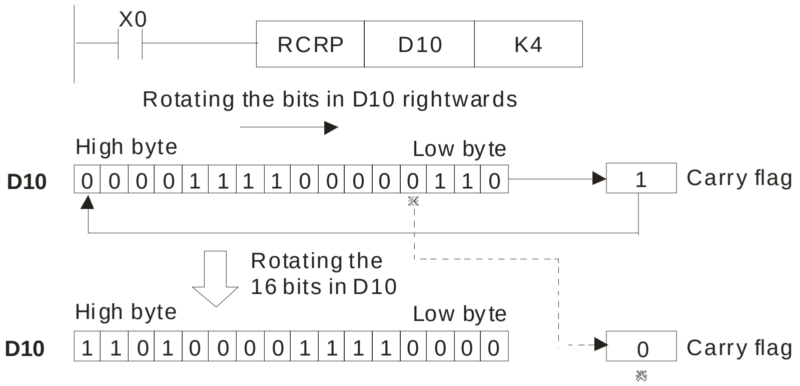
- Please refer to the additional remark below.

Explanation

- **D**: Device which is rotated; **n**: Number of bits forming a group
- The bits in **D** are divided into groups (**n** bits as a group), and these groups are rotated rightwards with a carry flag.
- The **n**th bit from the right is transmitted to a carry flag.
- Generally, the pulse instructions RCRP and DRCRP are used.
- If the operand **D** is KnY/KnM/KnS, Kn in KnY/KnM/KnS must be K4 (16 bits) or K8 (32 bits).
- 16-bit instruction: 1≤**n**≤16; 32-bit instruction: 1≤**n**≤32

Example

- When X0 is turned from OFF to ON, the bits in D10 are divided into groups (four bits as a group), and these groups are rotated rightwards with a carry flag. (The bit marked with ※ is transmitted to the carry flag.)



API						Applicable model
33	D	RCL	P	D n	Rotating bits leftwards with a carry flag	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								*	*	*	*	*	*	*	*
n					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (5 steps)

RCL Continuity instruction | RCLP Pulse instruction

32-bit instruction (9 steps)

DRCL Continuity instruction | DRCLP Pulse instruction

- Flags

Ox	O100	
M1810	M1970	Carry flag

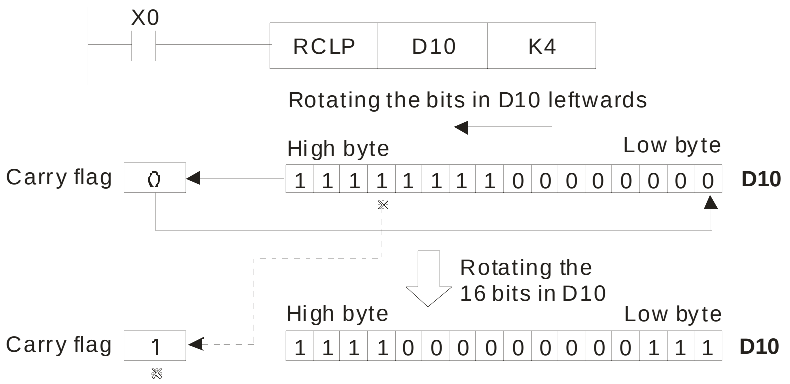
- Please refer to the additional remark below.

Explanation

- **D**: Device which is rotated。**n**: Number of bits forming a group
- The bits in **D** are divided into groups (**n** bits as a group), and these groups are rotated leftwards with a carry flag.
- The **n**th bit from the left is transmitted to a carry flag.
- Generally, the pulse instructions RCLP and DRCLP are used.
- If the operand **D** is KnY/KnM/KnS, Kn in KnY/KnM/KnS must be K4 (16 bits) or K8 (32 bits).
- 16-bit instruction: 1≤**n**≤16; 32-bit instruction: 1≤**n**≤32

Example

- When X0 is turned from OFF to ON, the bits in D10 are divided into groups (four bits as a group), and these groups are rotated leftwards with a carry flag.
  (The bit marked with ※ is transmitted to the carry flag.)



API		SFTR			Applicable model
34			P	S D n1 n2	Moving the states of bit devices rightwards	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	*	*	*	*											
D		*	*	*											
n1					*	*									
n2					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (9 steps)

SFTR	Continuity instruction	SFTRP	Pulse instruction

32-bit instruction

-	-	-	-

- Flag: None

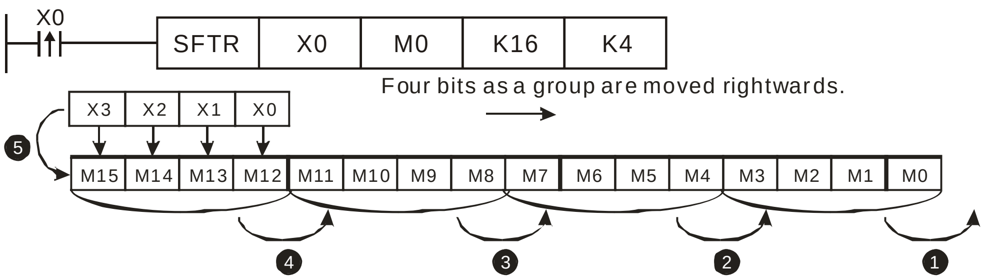
Explanation

- **S**: Initial device which is moved; **D**: Initial device which is moved; $\mathbf{n_1}$: Number of bits which are moved; $\mathbf{n_2}$: Number of bits forming a group
- The states of the $\mathbf{n_1}$ bit devices starting from **D** are divided into groups ($\mathbf{n_2}$ bits as a group), and these groups are moved rightwards. The states of the $\mathbf{n_2}$ bit devices starting from **S** are moved to the vacant devices in the devices starting from **D**.
- Generally, the pulse instruction SFTRP is used.
- $1 \le \mathbf{n_2} \le \mathbf{n_1} \le 1024$

Example

- When X0 is turned from OFF to ON, the states of the sixteen bit devices starting from M0 are divided into groups (four bits as a group), and these groups are moved rightwards.
- The states of the bit devices are moved rightwards in the order ❶~❺ during a scan cycle.

❶ M3~M0 → The states of M3~M0 are carried.
❷ M7~M4 → M3~M0
❸ M11~M8 → M7~M4
❹ M15~M12 → M11~M8
❺ X3~X0 → M15~M12



API		SFTL			Moving the states of bit devices leftwards	Applicable model
35			P	S D n1 n2		20PM ✓

	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
	Bit device				Word device										
S	*	*	*	*											
D		*	*	*											
n1					*	*									
n2					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (9 steps)
SFTL Continuity instruction SFTLP Pulse instruction

32-bit instruction
- - - -

- Flag: None

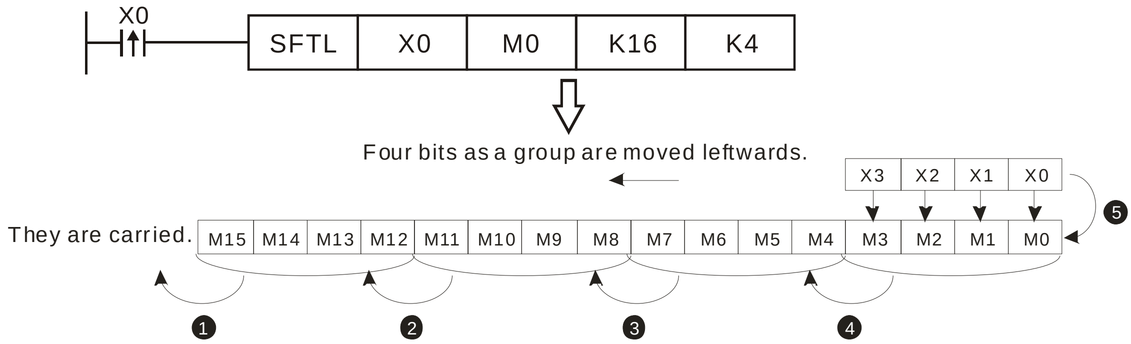
Explanation

- **S**: Initial bit device which is moved; **D**: Initial bit device which is moved; $\mathbf{n_1}$: Number of bits which are moved; $\mathbf{n_2}$: Number of bits forming a group
- The states of the $\mathbf{n_1}$ bit devices starting from **D** are divided into groups ($\mathbf{n_2}$ bits as a group), and these groups are moved leftwards. The states of the $\mathbf{n_2}$ bit devices starting from **S** are moved to the vacant devices in the devices starting from **D**.
- Generally, the pulse instruction SFTRP is used.
- $1 \leq \mathbf{n_2} \leq \mathbf{n_1} \leq 1024$

Example

- When X0 is turned from OFF to ON, the states of the sixteen bit devices starting from M0 are divided into groups (four bits as a group), and these groups are moved leftwards.
- The states of the bit devices are moved leftwards in the order ❶~❺ during a scan cycle.

❶ M15~M12 → The states of M3~M0 are carried.
❷ M11~M8 → M15~M12
❸ M7~M4 → M11~M8
❹ M3~M0 → M7~M4
❺ X3~X0 → M3~M0



API						
36		WSFR	P	S D n1 n2	Moving the values in word devices rightwards	Applicable model: 20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							*	*	*	*	*	*	*		
D								*	*	*	*	*	*		
n1					*	*									
n2					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (9 steps)
WSFR Continuity instruction | WSFRP Pulse instruction

32-bit instruction
- - - -

- Flag: None

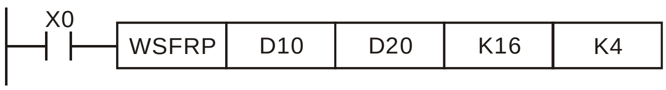
Explanation

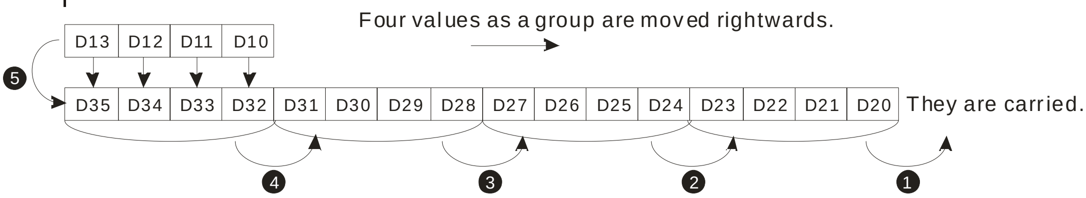
- **S**: Initial word device which is moved; **D**: Initial word device which is moved; $\mathbf{n_1}$: Number of values which are moved; $\mathbf{n_2}$: Number of values forming a group
- The values in the $\mathbf{n_1}$ word devices starting from **D** are divided into groups ($\mathbf{n_2}$ values as a group), and these groups are moved rightwards. The values in the $\mathbf{n_2}$ word devices starting from **S** are moved to the vacant word devices in the word devices starting from **D**.
- Generally, the pulse instruction WSFRP is used.
- If the operand **S** is KnX/KnY/KnM/KnS, the operand **D** can be a counter, timer, or a data register. If the operand **D** is KnY/KnM/KnS, the operand **S** can be a counter, timer, or a data register.
- If the operand **S** is KnX/KnY/KnM/KnS, and the operand **D** is KnY/KnM/KnS, Kn in KnX/KnY/KnM/KnS which is S and Kn in KnY/KnM/KnS must be the same.
- $1 \le \mathbf{n_2} \le \mathbf{n_1} \le 512$

Example 1

- When X0 is turned from OFF to ON, the values in the sixteen word devices starting from D20 are divided into groups (four values as a group), and these groups are moved rightwards.
- The values in the word devices are moved rightwards in the order ❶~❺ during a scan cycle.

❶ D23~D20 → The values in D23~D20 are carried.
❷ D27~D24 → D23~D20
❸ D31~D28 → D27~D24
❹ D35~D32 → D31~D28
❺ D13~D10 → D35~D32

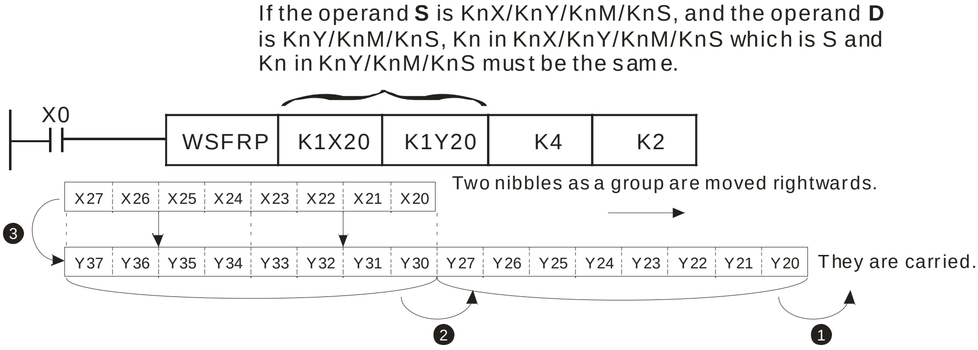




Example 2

- When X0 is turned from OFF to ON, the values in the sixteen bit devices starting from Y20 are divided into groups (eight values as a group), and these groups are moved rightwards.
- The values in the word devices are moved rightwards in the order ❶~❺ during a scan cycle.

❶ Y27~Y20 → The values in Y27~Y20 are carried.
❷ Y37~Y30 → Y27~Y20
❸ X27~X20 → Y37~Y30



API		Mnemonic		Operands	Function	Applicable model
37		WSFL	P	S D n1 n2	Moving the values in word devices leftwards	20PM ✓

	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
	Bit device				Word device										
S							*	*	*	*	*	*	*		
D								*	*	*	*	*	*		
n1					*	*									
n2					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (9 steps)

WSFL Continuity instruction WSFLP Pulse instruction

32-bit instruction

\- - - -

- Flag: None

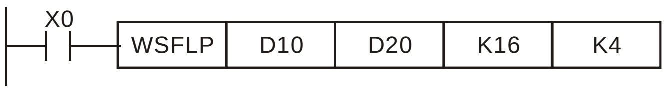
Explanation

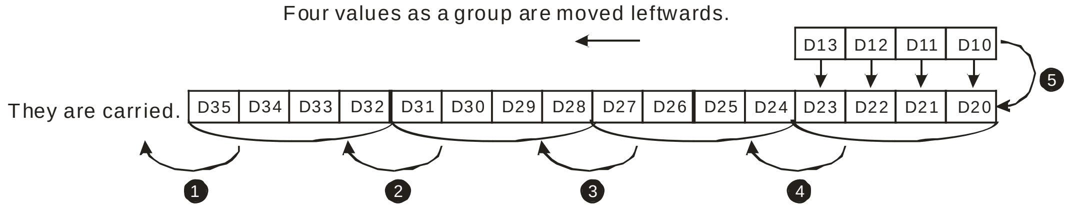
- **S**: Initial word device which is moved; **D**: Initial word device which is moved; $\mathbf{n_1}$: Number of values which are moved; $\mathbf{n_2}$: Number of values forming a group
- The values in the $\mathbf{n_1}$ word devices starting from **D** are divided into groups ($\mathbf{n_2}$ values as a group), and these groups are moved leftwards. The values in the $\mathbf{n_2}$ word devices starting from **S** are moved to the vacant word devices in the word devices starting from **D**.
- Generally, the pulse instruction WSFLP is used.
- If the operand **S** is KnX/KnY/KnM/KnS, the operand **D** can be a counter, timer, or a data register. If the operand **D** is KnY/KnM/KnS, the operand **S** can be a counter, timer, or a data register.
- If the operand S is KnX/KnY/KnM/KnS, and the operand D is KnY/KnM/KnS, Kn in KnX/KnY/KnM/KnS which is S and Kn in KnY/KnM/KnS must be the same.
- $1 \le \mathbf{n_2} \le \mathbf{n_1} \le 512$

Example

- When X0 is turned from OFF to ON, the values in the sixteen word devices starting from D20 are divided into groups (four values as a group), and these groups are moved leftwards.
- The values in the word devices are moved leftwards in the order ❶~❺ during a scan cycle.

❶ D35~D32 → The values in D35~D32 are carried.
❷ D31~D28 → D35~D32
❸ D27~D24 → D31~D28
❹ D23~D20 → D27~D24
❺ D13~D10 → D23~D20





API				Moving a value and writing it into a word device	Applicable model
38		**SFWR** P	Ⓢ Ⓓ ⓝ		20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*		
n					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (9 steps)

SFWR	Continuity instruction	SFWRP	Pulse instruction

32-bit instruction

-	-	-	-

- Flags

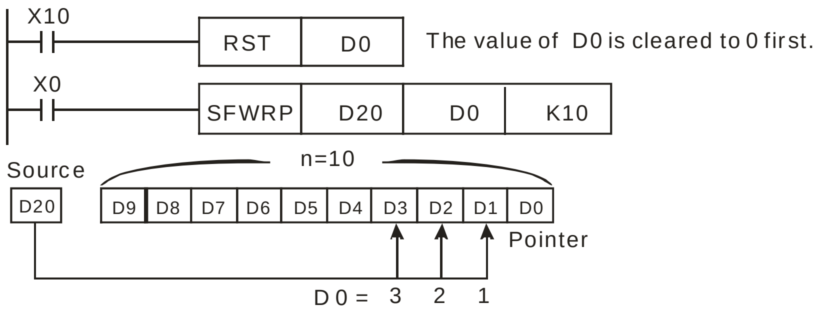
Ox	O100	
M1808	M1968	Zero flag

Explanation

- **S**: Device which is moved; **D**: Initial device; **n**: Number of devices
- The values in the **n** word devices starting from **D** are defined as first in, first out values, and **D** is taken as a pointer. When the instruction is executed, the value of the pointer **D** increases by one, and the value in **S** is written into the device to which the pointer **D** points. When the value of the pointer is greater than or equal to **n**-1, the instruction does not process the writing of the value, and a carry flag is ON.
- When the value of the pointer **D** is greater than **n**-1, the instruction does not process the writing of a value, and the carry flag M1022 is ON.
- Generally, the pulse instruction SFWRP is used.
- 2≤**n**≤512

Example

- The value of the pointer D0 is cleared to 0 first. When X0 is turned from OFF to ON, the value in D20 is written into D1, and the value of D0 becomes 1. When X0 is turned from OFF to ON again, the value in D20 is written to D2, and the value in D0 becomes 2.
- The value in D20 is moved and written into D1 in the way described below.
  ❶ The value in D20 is written into D1.
  ❷ The value of D0 becomes 1.



Additional remark

- The instruction SFWR can be used with the instruction SFRD to write a value and read values.

API						Applicable model
39		SFRD	P	S D n	Moving a value and reading it from a word device	20PM ✓

	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
	Bit device				Word device										
S					*	*		*	*	*	*	*	*		
D								*	*	*	*	*	*	*	*
n					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (9 steps)

SFRD Continuity instruction SFRDP Pulse instruction

32-bit instruction

\- - - -

- Flags

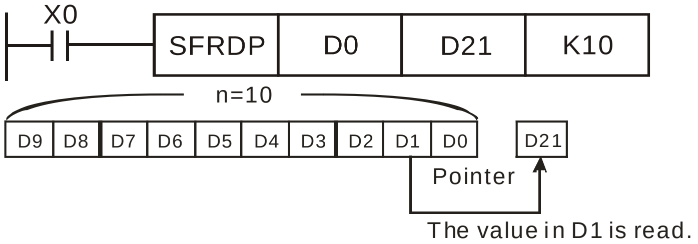
Ox	O100	
M1808	M1968	Zero flag

Explanation

- **S**: Initial device; **D**: Device into which a value is written; **n**: Number of devices
- The values in the **n** word devices starting from **S** are defined as first in, first out values, and **S** is taken as a pointer. When the instruction is executed, the value in **S** decreases by one, the value in **S**+**1** is written into **D**, the values in **S**+**n**-1~**S**+2 are moved rightwards, and the value in **S**+**n**-1 is unchanged. When the value in **S** is equal to 0, the instruction does not process the reading of the values, and a zero flag is ON.
- When the value in **S** is equal to 0, the instruction does not process the reading of the values, and the zero flag M1020 is ON.
- Generally, the pulse instruction SFRDP is used.
- 2≤**n**≤512

Example

- When X0 is turned from OFF to ON, the value in D1 is written into D21, the values in D9~D2 are moved rightwards, the value in D9 is unchanged, and the value in D0 decreases by one.
- The value in D1 is moved and written into D21 in the way described below.
  ❶ The value in D1 is written into D21.
  ❷ The values in D9~D2 are moved rightwards.
  ❸ The value in D0 decreases by one.



API		ZRST	☺			Applicable model
40			P	D1 D2	Resetting a zone	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D_1		*	*	*							*	*	*		
D_2		*	*	*							*	*	*		

- Note: Device number of $D_1 \leqq$ Device number of D_2

  The device type that D_1 specifies and the device type that D_2 specifies must be the same.

  All devices can not be modified by V devices and Z devices.

  Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)

ZRST	Continuity instruction	ZRSTP	Pulse instruction

32-bit instruction

-	-	-	-

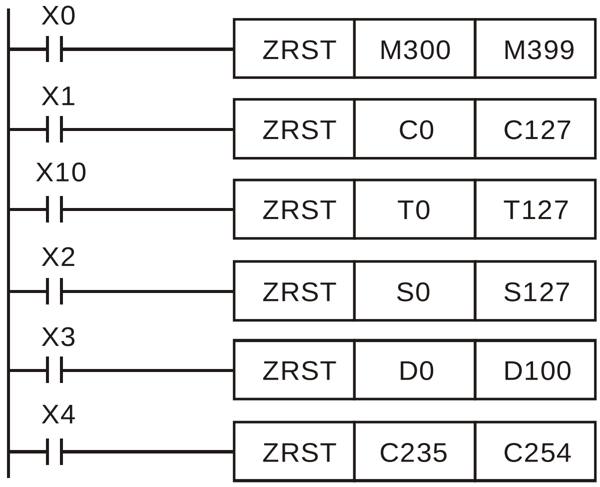
- Flag: None

Explanation

- **D_1**: Initial device which is reset; **D_2**: Final device which is reset
- The instruction ZRST can be used to reset 16-bit counter and 32-bit counters.
- If the device number of **D_1** is greater than the device number of **D_2**, only **D_2** will be reset.

Example

- When X0 is ON, the auxiliary relays M300~M399 are reset to OFF.
- When X1 is ON, the 16-bit counters C0~C127 are reset. (The values of C0~C127 are cleared to 0, and the contacts and the coils are reset to OFF.)
- When X10 is ON, the timers T0~T127 are reset. (The values of T0~T127 are cleared to 0. and the contacts and the coils are reset to OFF.)
- When X2 is ON, the stepping relays S0~S127 are reset to OFF.
- When X3 is ON, the data registers D0~D100 are reset to 0.
- When X4 is ON, the 32-bit counters C235~C254 are reset. (The values of C235~C254 are cleared to 0, and the contacts and the coils are reset to OFF.)



Additional remark

- The instruction RST can be used to reset a single device, e.g. a Y device, an M device, an S device, a T device, a C device, or a D device.

API						Applicable model
41		DECO	P	(S) (D) (n)	Decoder	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S	*	*	*	*	*	*					*	*	*	*	*
D		*	*	*							*	*	*	*	*
n					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

DECO Continuity instruction DECOP Pulse instruction

32-bit instruction

- - - -

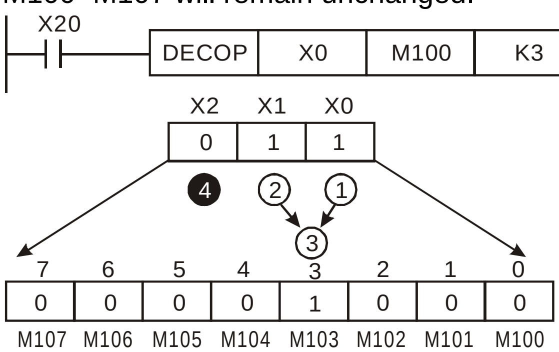
- Flag: None

Explanation

- **S**: Source device; **D**: Device in which a decoding result is stored **n**: Number of bits which are decoded
- The low **n** bits in **S** are decoded as the low 2^n bits in **D**.
- Generally, the pulse instruction DECOP is used.
- **D** is in the range of 1 to 8.

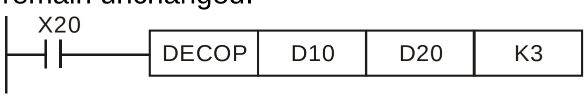
Example 1

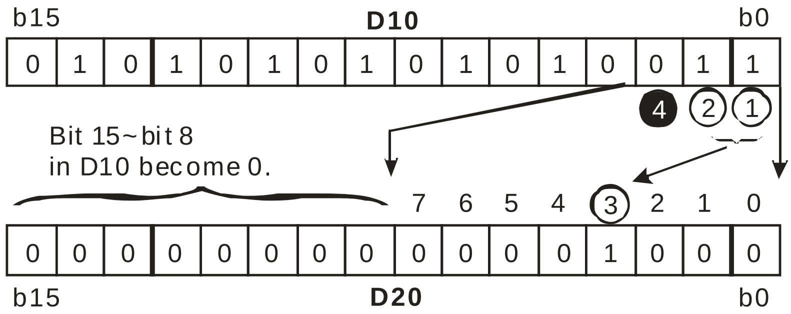
- When **D** is a bit device, **n** is in the range of 1 to 8. If **n** is 0, or greater than 8, an error will occur.
- If **n** is 8, the maximum number of bits which can be decoded is 2^8=256.
- When X20 is turned from OFF to ON, the instruction DECOP decodes X0~X2 as M100~M107.
- If the value in **S** is 3, M103 will be ON.
- After the instruciton is executed, X20 will be OFF, and the states of M100~M107 will remain unchanged.



Example 2

- When **D** is a word device, **n** is in the range of 1 to 8. If **n** is 0, or greater than 8, an error will occur.
- If **n** is 8, the maximum number of bits which can be decoded is 2^8=256.
- When X20 is turned from OFF to ON, the instruction DECOP decodes b2~b0 in D10 as b7~b0 in D20, and b15~b8 in D20 become 0.
- The low 3 bits in D10 are decoded as the low 8 bits in D20. The high 8 bits in D20 are 0.
- After the instruciton is executed, X20 will be OFF, and the value in D20 will remain unchanged.




b15
D10
b0
0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 1
4 2 1
Bit 15~bit 8
in D10 become 0.
7 6 5 4 3 2 1 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
b15
D20
b0

API	Mnemonic		Operands	Function	Applicable model
42	ENCO	P	S D n	Encoder	20PM ✓

	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
	Bit device				Word device										
S	*	*	*	*							*	*	*	*	*
D											*	*	*	*	*
n					*	*									

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

ENCO Continuity instruction | ENCOP Pulse instruction

32-bit instruction

\- \- \- \-

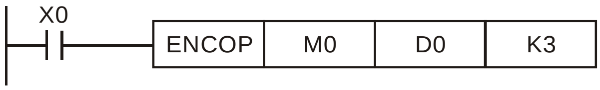
- Flag: None

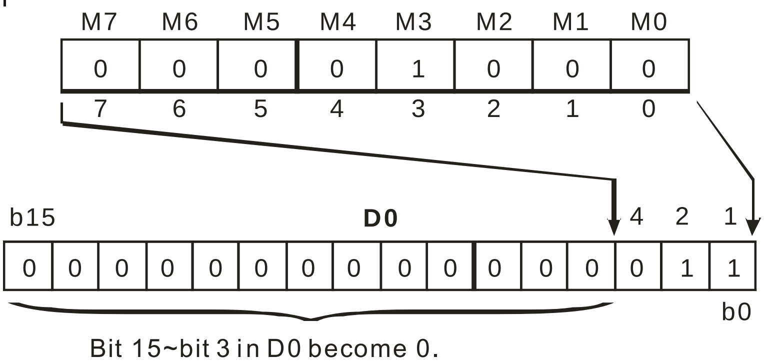
Explanation

- **S**: Source device **D**: Device in which an encoding result is stored **n**: Number of bits which are encoded
- The low 2^n bits in **S** are encoded as the low **n** bits in **D**.
- If there are many bits which are 1 in **S**, the first bit which is 1 from the left will be processed.
- Generally, the pulse instruction ENCOP is executed.
- If **S** is a bit device, **n** is in the range of 1 to 8. If **S** is a word device, **n** is in the range of 1 to 4.

Example 1

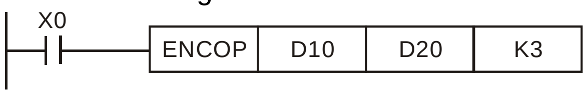
- When **S** is a bit device, **n** is in the range of 1 to 8. If **n** is 0, or greater than 8, an error will occur.
- If **n** is 8, the maximum number of bits which can be decoded is 2^8=256.
- When X0 is turned from OFF to ON, the instruction ENCOP encodes the 8 bits in M0~M7 as the low 3 bits in D0, and b15~b3 in D0 become 0.
- After the instruction ENCOP is executed, X0 will be OFF, and the data in **D** will remain unchanged.

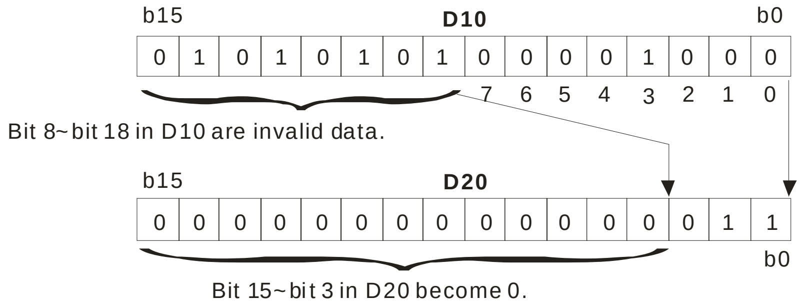




Example 2

- When **S** is a word device, **n** is in the range of 1 to 4. If **n** is 0, or larger than 4, an error will occur.
- If **n** is 4, the maximum number of bits which can be decoded is 2^4=16.
- When X0 is turned from OFF to ON, the instruction ENCOP encodes the 8 bits in D10 as the low 3 bits in D20, and b15~b3 in D20 become 0. (Bit 8~bit 15 in D10 are invalid data.)
- After the instruction ENCOP is executed, X0 will be OFF, and the data in **D** will remain unchanged.




b15
D10
b0
0 1 0 1 0 1 0 1 0 0 0 0 1 0 0 0
7 6 5 4 3 2 1 0
Bit 8~bit 18 in D10 are invalid data.
b15
D20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
b0
Bit 15~bit 3 in D20 become 0.

API						Applicable model
43	D	SUM	P	(S) (D)	Number of bits which are ON	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S							*	*	*	*	*	*	*	*	*
D											*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X device numbers/Y device numbers/M device numbers/S device numbers should start from a number which is a multiple of 16 (including 0), e.g. K1X0, K4Y20 and K4M16.

16-bit instruction (5 steps)

SUM	Continuity instruction	SUMP	Pulse instruction

32-bit instruction (9 steps)

DSUM	Continuity instruction	DSUMP	Pulse instruction

- Flags

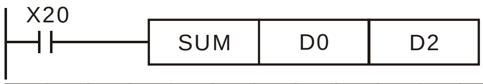
Ox	O100	
M1808	M1968	Zero flag

Explanation

- **S**: Source device; **D**: Destination device
- The number of bits which are 1 in **S** is stored in **D**.
- If the bits in **S** are 0, a zero flag will be ON.
- If the 32-bit instruction is used, **D** will occupy two registers.

Example

- When X20 is ON, the number of bits which are 1 in D0 is stored in D2.



0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0		3

D0 D2

API						Applicable model
44	D	BON	P	S D n	Checking the state of a bit	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*	*	*	*	*	*	*	*	*	*	*
D		*	*	*											
n					*	*					*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (7 steps)

BON	Continuity instruction	BON P	Pulse instruction

32-bit instruction (13 steps)

DBON	Continuity instruction	DBON P	Pulse instruction

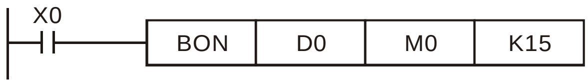
- Flag: None

Explanation

- **S**: Source device; **D**: Device in which a check result is stored; **n**: Bit whose state is judged
- The state of the **n**th bit in **S** is checked, and the result is stored in **D**.
- 16-bit instruction: **n**=0~15; 32-bit instruction: **n**=0~31

Example

- If the 15th bit in D0 is 1 when X0 is ON, M0 will be ON. If the 15th bit in D0 is 0 when X0 is ON, M0 will be OFF.
- When X0 is turned OFF, the state of M0 remains unchanged.



b15 ... b0 (D0)

b15															b0	
0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	M0=Off

D0

b15															b0	
1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	M0=On

D0

API						Applicable model
45	D	MEAN	P	S D n	Mean	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*	*
n					*	*	*	*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (7 steps)

MEAN Continuity instruction / MEANP Pulse instruction

32-bit instruction (13 steps)

DMEAN Continuity instruction / DMEANP Pulse instruction

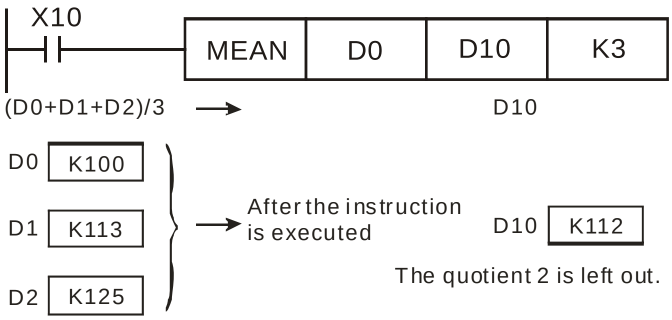
- Flag: None

Explanation

- **S**: Initial device; **D**: Device in which a mean is stored; **n**: Number of devices
- After the values in the **n** devices starting from **S** are added up, the mean of the sum is stored in **D**.
- If a remainder appears in a calculation, it will be left out.
- If **S** is not in a valid range, only the devices in the valid range will be processed.
- If **n** is not in the range of 1 to 64, an operation error will occur.
- **n**=1~64

Example

- When X10 is ON, the values in the three registers starting from D0 are added up. After the values are added up, the sum will be divided by 3. The quotient is stored in D10, and the remainder is left out.



API		ANS		Operands	Driving an annunciator	Applicable model
46			P	S m D		20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S											*				
m					*										
D				*											

- Note: All devices can not be modified by V devices and Z devices. Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

ANS Continuity instruction ANS P Pulse instruction

32-bit instruction

\- - - -

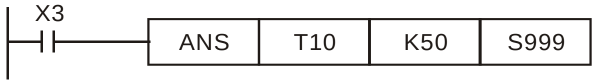
- Flag: None

Explanation

- **S**: Timer; **m**: Time; **D**: Annunciator
- The instruction ANS is used to drive an annunciator.
- **S**: T0~T183
  m: K1~K32,767 (Unit: 100 ms)
  D: S912~S1023
  See the explanation of ANR for more information.

Example

- If X3 is ON for more than 5 seconds, the annunciator S999 will be ON. Even if X3 is turned OFF, S999 will still be ON. (However, T10 will be reset to OFF, and the value of T10 will be 0.)



API				Applicable model
47	ANR	P	Resetting an annunciator	20PM
				✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z

- Note: There is no operand.
  The instruction does not need to be driven by a contact.

16-bit instruction (1 step)

ANR	Continuity instruction	ANRP	Pulse instruction

32-bit instruction

-	-	-	-

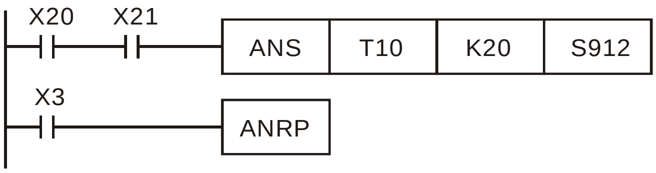
- Flag: None

Explanation

- The instruction ANR is used to reset an annunciator.
- If more than one annunciator is ON simultaneously, the annunciator whose number is smallest will be reset.
- Generally, the pulse instruction ANRP is used.

Example

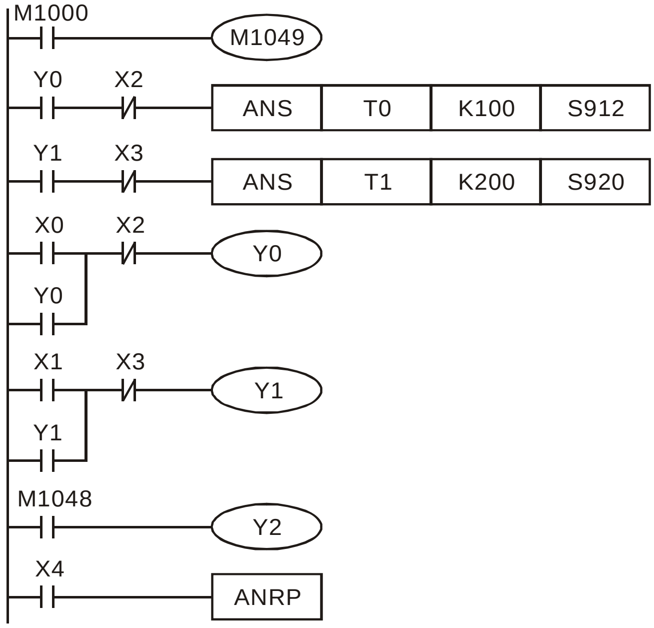
- If X20 and X21 are ON for more than 2 seconds, the annunciator S912 will be ON. If X20 and D21 are turned OFF, S912 will still be ON, T10 will be reset to OFF, and the value of T10 will be 0.
- If X20 and X21 are not ON for 2 seconds, the value of T10 will become 0.
- When X0.3 is turned from OFF to ON, the annunciator whose number is smallest in the annunciators which are driven is reset.
- When X0.3 is turned from OFF to ON again, the next annunciator whose number is smallest in the annunciators which are driven is reset.



Additional remark

◆ Application of annunciators

X0=Forward switch X1=Backward switch
X2=Front position switch X3=Back position switch
X4=Resetting button
Y0=Forward Y1=Backward
Y2=Indicator
S912=Forward annunciator S920=Backward annunciator



1. If Y0 is ON for more than 10 seconds, and X2 is OFF, S912 will be ON.
2. If Y1 is ON for more than 20 seconds, and X3 is OFF, S920 will be ON.
3. If X1 and Y1 are ON, Y1 will not be OFF until X3 is ON.
4. If an annuciator is driven, Y2 will be ON.
5. When X4 is turned from OFF to ON, the annunciator whose number is smallest in the annunciators which are driven is reset. When X4 is turned from OFF to ON again, the next annunciator whose number is smallest in the annunciators which are driven is reset.

API						Applicable model
48	D	SQR	P	S D	Square root of a binary value	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S											*				
m					*										
D				*											

- Note: All devices can not be modified by V devices and Z devices. Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

SQR	Continuity instruction	SQR P	Pulse instruction

32-bit instruction

DSQR	Continuity instruction	DSQRP	Pulse instruction

- Flags

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

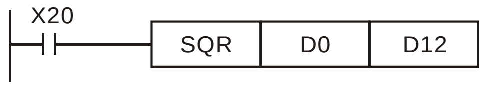
- Please refer to the explanation below.

Explanation

- **S**: Source device; **D**: Device in which a result is stored
- The square root of the value in **S** is calculated, and the result is stored in **D**.
- The value in **S** can only be a positive value. If the value in **S** is a negative value, an error will occur, and the instruction will not be executed.
- The value stored in **D** is an integer. The fractional part of a square root calculated is dropped. If the fractional part of a square root calculated is dropped, a borrow flag will be ON.
- If the value in **D** is 0, a zero flag will be ON.

Example

- When X20 is ON, the square root of the value in D0 is calculated, and the result is stored in D12.



$\sqrt{D0} \rightarrow D12$

API						Applicable model
49	D	FLT	P	S D	Converting a binary integer into a binary floating-point value	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*	*							*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DFLT and DFLTP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (6 steps)

DFLT	Continuity instruction	DFLTP	Pulse instruction

- Flags

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

- Please refer to the explanation below.

Explanation

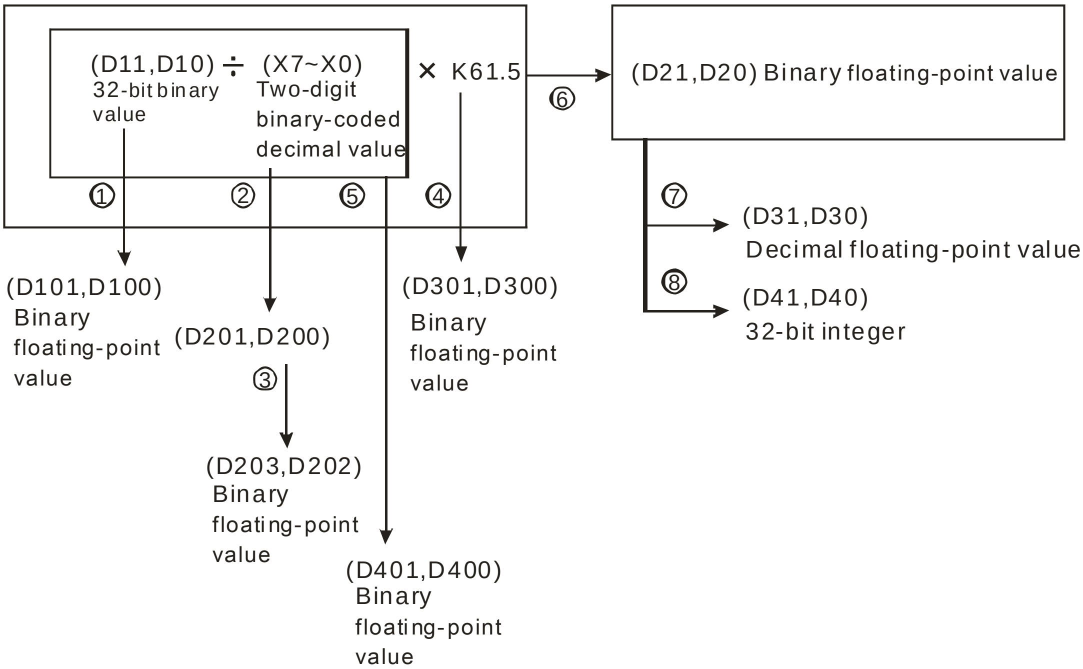
- **S**: Source device; **D**: Conversion result
- The instruction is used to convert a binary integer into a binary floating-point value.
  1. If the absolute value of the converesion result is greater than the maximum floating-point value avaliable, a carry flag will be ON.
  2. If absolute value of the conversion result is less than the mimum floating-point vlaue available, a borrow flag will be ON.
  3. If the conversion result is 0, a zero flag will be ON.

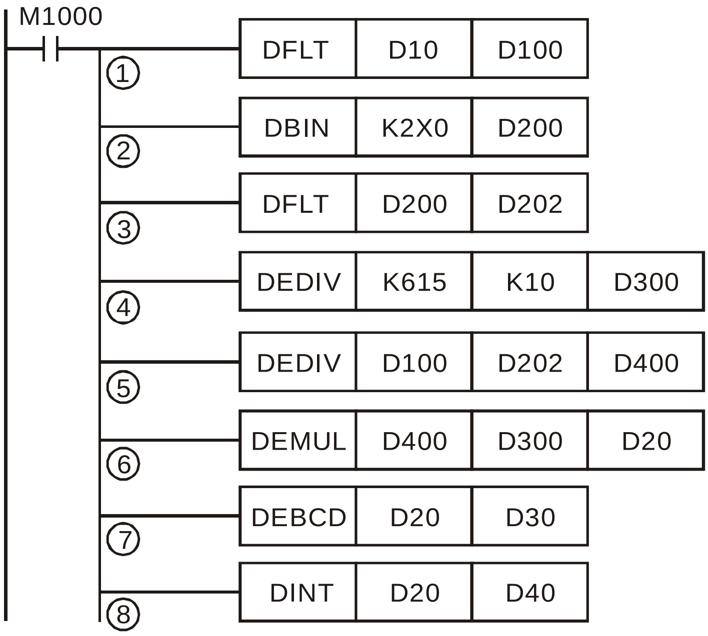
Example 1

- When X11 is ON, the binary integer in (D1, D0) is converted into a binary floating-point value, and the conversion result is stored in (D21, D20).
- Suppose the value in the 32-bit register (D1, D0) is K100,000. When X11 is ON, K100,000 is converted into the 32-bit floating-point value H4735000, and H4735000 is stored in the 32-bit register (D21, D20).

Example 2

- Users can use applied instructions to perform the following calculation.





1. The binary integer in (D11, D10) is converted into a binary floating-point value, and the conversion result is stored in (D101, D100).
2. The binary-coded decimal value in X7~X0 is converted into a binary value, and the conversion result is stored in (D201, D200).
3. The binary integer in (D201, D200) is converted into a binary floating-point value, and the conversion result is stored in (D203, D202).
4. The constant K615 is divided by the constant K10, and the quotient which is a binary floating-point value is stored in (D301, D300).
5. The binary floating-point value in (D101, D100) is divided by the binary floating-point value in (D203, D202), and the quotient which is a binary floating-point value is stored in (D401, D400).
6. The binary floating-point value in (D401, D400) is multiplied by the binary floating-point value in (D301, D300), and the product which is a binary floating-point value is stored in (D21, D20).
7. The binary floating-point value in (D21, D20) is converted into a decimal floating-point value, and the conversion result is stored in (D31, D30).
8. The binary floating-point value in (D21, D20) is converted into a binary integer, and the conversion result is stored in (D41, D40).

API						Applicable model
50		REF	P	D n	Refreshing the states of I/O devices	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D	*	*													
n					*	*									

- Note: All devices can not be modified by V devices and Z devices. Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

REF Continuity instruction REF P Pulse instruction

32-bit instruction

\- - - -

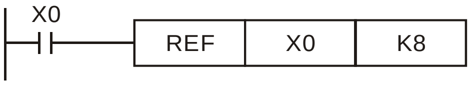
- Flag: None

Explanation

- **D**: Initial I/O device whose state is refreshed; **n**: Number of I/O devices whose states are refreshed
- The states of I/O devices are not refreshed until the instruction END is executed. When the scan of a program starts, the states of external inputs are read, and stored in the input memory. After the instruction END is executed, the contents of the output memory will be sent to output terminals. Therefore, users can use this instruction when they need the latest I/O data in an operation process.
- **D** must be an I/O device whose number ends with 0, e.g. X0, X10, Y0 or Y10. The instruction can not be used to refresh the I/O devices in a digital extension module.
- **D** must be an I/O device in a PLC.
  1. If **D** is X0 and **n** is less than or equal to 8, the states of X0~X0 will be refreshed. If **n** is greater than 8, the states of the input devices and the states of the output devices in the motion controller used will be refreshed.
  2. If **D** is Y0, and **n** is equal to 8, the states of Y0~Y7 will be refreshed. If **n** is greater than 8, the states of the input devices and the states of the output devices in the motion controller used will be refreshed.
  3. If **D** is X10 or Y10, and **n** is any number, the states of all the input devices except X0~X7, and the states of all the output devices except Y0~Y3 in the motion controller used will be refreshed.
- **n** is in the range of 4 to the number of I/O devices in the motion control module used, and is a multiple of 4.

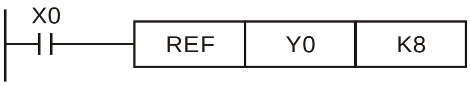
Example 1

- When X0 is ON, the DVP-20PM series motion controller reads the states of X0~X7 immediately. The input signals are refreshed without any delay.



Example 2

- When X0 is ON, the states of Y0~Y3 are sent to output terminals. The output signals are refreshed immediately without the need to wait for the execution of the instruction END.



Example 3

- When X0 is ON, the states of the input terminals starting from X10, or the states of the output terminals starting from Y10 are refreshed.

Or

API						Applicable model
61	D	SER	P	S1 S2 D n	Searching data	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1							*	*	*	*	*	*	*		
S2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*		
N					*	*							*		

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (9 steps)

SER	Continuity instruction	SERP	Pulse instruction

32-bit instruction (17 steps)

DSER	Continuity instruction	DSERP	Pulse instruction

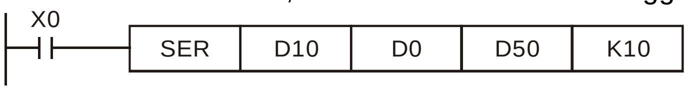
- Flag: None

Explanation

- S_1: Initial device involved in a comparison; S_2: Value which is compared; **D**: Initial device in which a comparison result is stored (5 consecutive devices are occupied.); **n**: Number of values
- S_1 is the initial register involved in a comparison, and **n** is the number of values which are compared. The values in the **n** registers starting from S_1 are compared with the value in S_2, and the comparison results are stored in the five registers starting from **D**.
- If the 32-bit instruction is used, S_1, S_2, **D**, and **n** will be 32-bit registers.
- 16-bit instruction: **n**=1~256; **n**=1~128 (32-bit instruction)

Example

- When X0 is ON, the values in D10~D19 are compared with the value in D0, and the comparison results are stored in D50~D54. If none of the values in D10~D19 are equal to the value in D0, the values in D50~D52 will be 0.
- A comparison is based on algebra (-10 < 2).
- The number of the minimum value is stored in D53, and the number of the maximum value is stored in D54. If there is more than one minimum value/maximum value, the number which is the biggest will be stored.



n

S_1	Value	Value which is compared	Number	Result
D10	88		0	
D11	100		1	Equal
D12	110	S_2 D0=K100	2	
D13	150		3	
D14	100		4	Equal
D15	300		5	
D16	100		6	Equal
D17	5		7	Minimum
D18	100		8	Equal
D19	500		9	Maximum

D	Value	Description
D50	4	Number of values which are equal to the value in D0
D51	1	Number of the first value which is equal to the value in D0
D52	8	Number of the last value which is equal to the value in D0
D53	7	Number of the minimum value
D54	9	Number of the maximum value

API		Mnemonic		Operands	Function	Applicable model
66		ALT	P	D	Alternating between ON and OFF	20PM ✓

Type / OP	Bit device X	Y	M	S	Word device K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D		*	*	*											

- Note: All devices can not be modified by V devices and Z devices. Please refer to specifications for more information about device ranges.

16-bit instruction (9 steps): ALT Continuity instruction; ALTP Pulse instruction

32-bit instruction (17 steps): - - - -

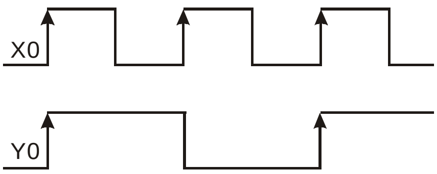
- Flag: None

Explanation

- **D**: Destination device
- When the instruction ALT is executed, the state of **D** alternates between ON and OFF.
- Generally, the pulse instruction ALTP is used.

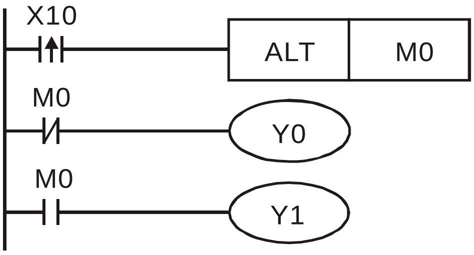
Example 1

- When X0 is turned from OFF to ON for the first time, Y0 is ON. When X0 is turned from OFF to ON for the second time, Y0 is OFF.



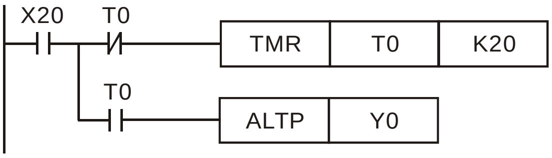
Example 2

- In the beginning, M0 is OFF, and therefore Y0 is ON, and Y1 is OFF. When X10 is turned from OFF to ON for the first time, M0 is ON. Therefore, Y1 is ON, and Y0 is OFF. When X10 is switched from OFF to ON for the second time, M0 is OFF. Therefore, Y0 is ON, and Y1 is OFF.



Example 3

- When X20 is ON, T0 generates a pulse every two seconds. The output Y0 alternates between ON and OFF according to the pulses generated by T0.



API						Applicable model
67	D	RAMP		S_1 S_2 D n	Ramp	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1													*		
S2													*		
D													*		
N													*		

- Note: All devices can not be modified by V devices and Z devices. Please refer to specifications for more information about device ranges.

16-bit instruction (9 steps)
RAMP: Continuity instruction; -

32-bit instruction (17 steps)
D RAMP: Continuity instruction; -

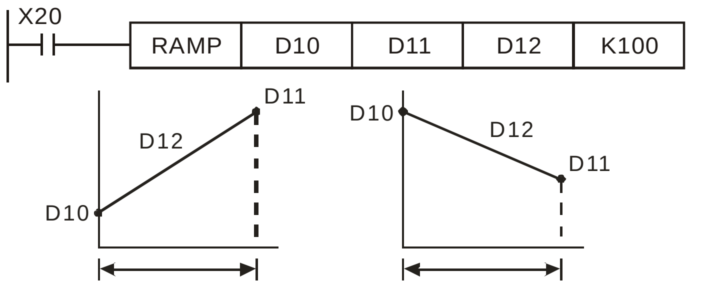
- Flags: M1026 (Please refer to the additional remark below.) M1029

Explanation

- S_1: Start a ramp; S_2: End of a ramp; **D**: Duration of a ramp (2 consecutive devices are occupied.); **n**: Number of scan cycles (**n**=1~32,767)
- The instruction is used to get a slope. Whether a slope is linear or not has an absolute relationship with scan time. When users use the instruction, they have to specify scan time in advance.
- When the contact driving the instruction RAMP is turned from OFF to ON, the value in **D** will increase from the value in S_1 to the value S_2, and the number of scan cycles is stored in **D**+1.
- If the operand **n** is a D device, the value in **n** can not be changed until the execution of the instruction stops.
- If the instruction is used with an output of analog signals, the action of cushioning a start/stop can be executed.

Example

- The start of a ramp is written into D10, and the end of the ramp is written into D11. When X20 is turned ON, the value in D12 increases from the value in D10 to the value in D11, and the number of scan cycles is stored in D13.
- After M1039 in a program is turned ON, the scan time for the program will be fixed. Users can write scan time into the special data register D1039 by means of the instruction MOV. If the scan time set is 30 milliseconds, and **n** is K100, the time it takes for the value in D12 to increase from the value in D10 to the value in D11 will be 3 seconds (30 milliseconds×100).
- If X20 is turned OFF, the execution of the instruction will stop. If X20 is turned ON again, the value in D12 will become 0, and increase again.
- If M1026 is OFF, and M1029 is ON, the value in D12 will becomes the value in D10.

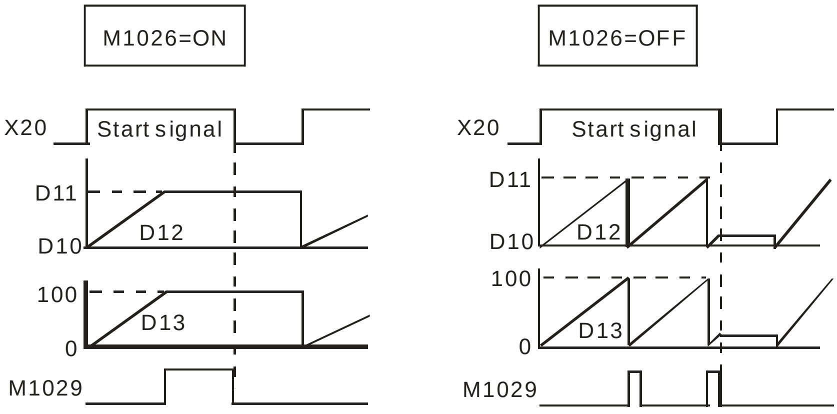


The number of scan cycles is n. D10<D11 The number of scan cycles is n. D10>D11

The number of scan cycle is stored in D13.

Additional remark

- If M1026 is turned ON/OFF, the value in D12 will change in the way described below.



API			Operands	Function	Applicable model
69	D	SORT	S m1 m2 D n	Sorting data	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S													*		
m1					*	*									
m2					*	*									
D													*		
N					*	*							*		

- Note: All devices can not be modified by V devices and Z devices.
  Please refer to specifications for more information about device ranges.

16-bit instruction (9 steps)
SORT Continuity instruction -

32-bit instruction (17 steps)
DSORT Continuity instruction -

- Flag: None

Explanation

- **S**: Initial device in which original data is stored; $\mathbf{m_1}$: Number of rows of data ($\mathbf{m_1}$ =1~32); $\mathbf{m_2}$: Number of columns of data ($\mathbf{m_2}$ =1~6); **D**: Initial device in which a sorting result is stored; **n**: Reference value (**n**=1~**m2**) (Data is sorted in algebraic order.)
- The data which is sorted is stored in the $\mathbf{m_1} \times \mathbf{m_2}$ registers starting from the register specified by **D**. If **S** and **D** specify the same register, the sorting result gotten will be the same as the original data in the register specified by **S**.
- It is better that the rightmost number of the device number of the register specified by **S** is 0.
- After the instruction is scanned $\mathbf{m_1}$ times, the sorting of data will be complete. After the sorting of data is complete, M1029 will be ON.
- The instruction can be used several times in a program, but one instruction is executed at a time.

Example

- When X0 is turned ON, the data specified is sorted in ascending order. When the sorting of the data specified is complete, M1029 is ON. When the instruction is executed, the data specified can not be changed. If users want to sort the data specified again, they can turn X0 from OFF to ON again.

1. The data which will be sorted is shown below.

m_2 columns of data

Row \ Column	1 Student number	2 Chinese	3 English	4 Math	5 Physics
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

m_1 rows of data

2. If the value in D100 is K3, users can get the sorting result shown below.

m_2 columns of data; m_1 rows of data

Row \ Column	1 Student number	2 Chinese	3 English	4 Math	5 Physics
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

3. If the value in D100 is K5, users can get the sorting result shown below.

m_2 columns of data; m_1 rows of data

Row \ Column	1 Student number	2 Chinese	3 English	4 Math	5 Physics
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

API				Operands	Function	Applicable model
78	D	FROM	P	m1 m2 D n	Reading data from a control register in a special module	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
m_1					*	*					*	*	*	*	*
m_2					*	*					*	*	*	*	*
D											*	*	*	*	*
n					*	*					*	*	*	*	*

- Note: m_1 is in the range of 0 to 255 (16-bit instruction/32-bit instruction).
  m_2 is in the range of 0 to 499 (16-bit instruction/32-bit instruction).
  n is in the range of 1 to (500-m_2) (16-bit instruction).
  n is in the range of 1~(500-m_2)/2 (32-bit instruction).
  The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)

16-bit instruction (9 steps)

FROM	Continuity instruction	FROMP	Pulse instruction

32-bit instruction (17 steps)

DFROM	Continuity instruction	DFROMP	Pulse instruction

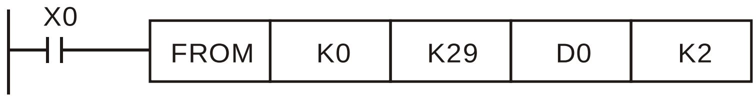
- Please refer to the additional remark below.

Explanation

- $\mathbf{m_1}$: Special module number ($\mathbf{m_1}$ is in the range of 0 to 255.); $\mathbf{m_2}$: Control register number ($\mathbf{m_2}$ is in the range of 0 to 499.); **D**: Device in which the data read will be stored; **n**: Quantity of data which will be read (16-bit instruction: 1~(500-$\mathbf{m_2}$); 32-bit instruction: 1~(500-$\mathbf{m_2}$)/2
- A DVP-10PM series motion controller can read the data in a control register in a special module by means of the instruction.
- Please refer to the additional remark on the instruction TO for more information about the numbering of special modules.

Example

- The value in CR#29 in special module 0 is read, and then stored in D0 in the motion controller used. The value in CR#30 in special module 0 is read, and then stored in D1 in the motion controller used. The two values are read at the same time.
- When X0 is ON, the instruciton is executed. When X0 is turned OFF, the instruction is not executed, and the values which are read remain unchanged.



API		TO				Applicable model
79	D		P	m1 m2 S n	Writing data into a control register in a special module	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
m_1					*	*					*	*	*	*	*
m_2					*	*					*	*	*	*	*
S					*	*					*	*	*	*	*
n					*	*					*	*	*	*	*

- Note: m_1 is in the range of 0 to 255 (16-bit instruction/32-bit instruction).
  m_2 is in the range of 0 to 499 (16-bit instruction/32-bit instruction).
  n is in the range of 1 to (500-m_2) (16-bit instruction).
  n is in the range of 1~(500-m_2)/2 (32-bit instruction).
  The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)

16-bit instruction (9 steps)

TO — Continuity instruction; TOP — Pulse instruction

32-bit instruction (17 steps)

DTO — Continuity instruction; DTOP — Pulse instruction

- Please refer to the additional remark below.

Explanation

- **m_1**: Special module number (**m_1** is in the range of 0 to 255.); **m_2**: Control register number (**m_2** is in the range of 0 to 499.); **D**: Data which will be written into a control register; **n**: Quantity of data which will be written (16-bit instruction: 1~(500-**m_2**); 32-bit instruction: 1~(500-**m_2**)/2
- A DVP-20PM series motion controller can write data into a control register in a special module by means of the instruction.

Example

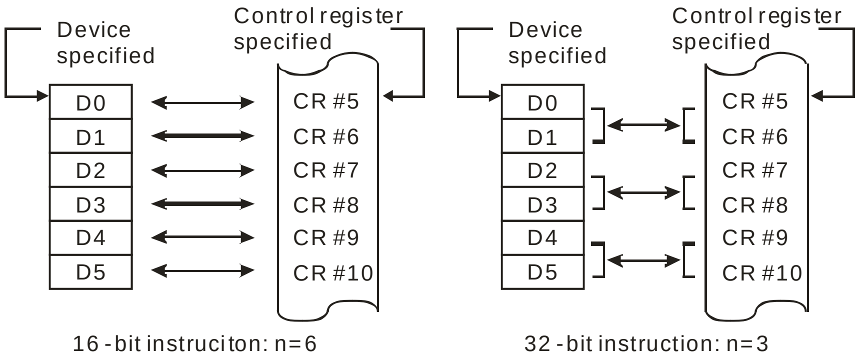
- The 32-bit instruction DTO is used. The value in (D11, D10) is written into (CR#13, CR#12) in special module 0. One value is written at a time.
- When X0 is ON, the instruction is executed. When X0 is turned OFF, the instruction is not executed, and the value which is written remains unchanged.

X0	DTO	K0	K12	D10	K1

- Regulation of operands
  1. **m_1**: **m_1** is a special module number. It is the number of a special module connected to the DVP-10PM series motion controller used.
     The number of the first special module which is connected to the DVP-10PM series motion cotroller used is 0. Eight special modules at most can be connected to the DVP-10PM series motion controller used, and they do not occupy I/O devices.
  2. **m_2**: **m_2** is a control register number. the 16-bit memories in a special modules are called control registers. Control register numbers are decimal numbers. The operation of a special module and setting values are stored in the control registers in the special module.
  3. If the instruction FROM/TO is used, one control register is taken as a unit for the reading/writing of data. If the instruction DFROM/DTO is used, two control registers are taken as a unit for the reading/writing of data.

High 16 bits	Low 16 bits	
CR #10	CR #9	◄— Control register number specified

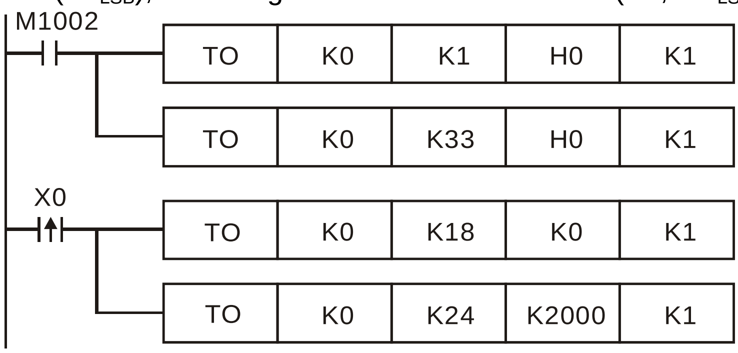
  4. **n** which is 2 in a 16-bit instruction has the same meaning as **n** which is 1 in a 32-bit instruction.



- The application of the instruciton FROM/TO is described below.

Example 1

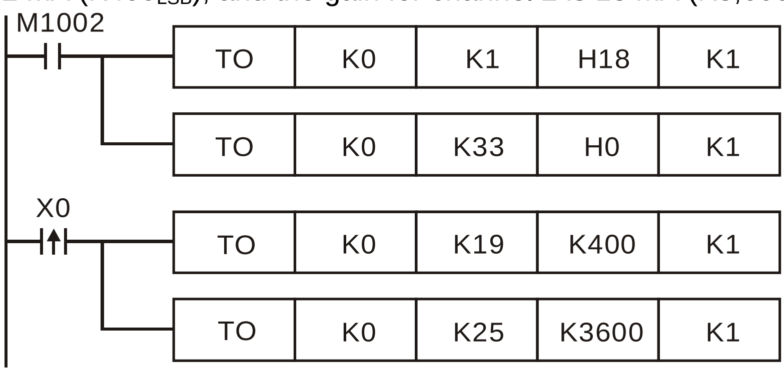
- The characteristic curve of DVP04AD-H2 is adjusted. The offset for channel 1 is 0 V (K0$_{LSB}$), and the gain for channel 1 is 2.5 V (K2,000$_{LSB}$).



1. H0 is written into CR#1 in the analog input module whose number is 0, and channel 1 is set to mode 0 (voltage input: -10 V~+10 V).
2. H0 is written into CR#33. Channel 1~channel 4 can be tuned.
3. When X0 is turned from OFF to ON, the offset K0$_{LSB}$ is writtedn into CR#18, and the gain K2,000$_{LSB}$ is written into CR#24.

Example 2

- The characteristic curve of DVP04AD-H2 is adjusted. The offset for channel 2 is 2 mA (K400$_{LSB}$), and the gain for channel 2 is 18 mA (K3,600$_{LSB}$).



1. H18 is written into CR#1 in the analog input module whose number is 0, and channel 2 is set to mode 3 (current input: -20 mA~+20 mA).
2. H0 is written into CR#33. Channel 1~channel 4 can be tuned.
3. When X0 is turned from OFF to ON, the offset K400$_{LSB}$ is writtedn into CR#19, and the gain K3,600$_{LSB}$ is written into CR#25.

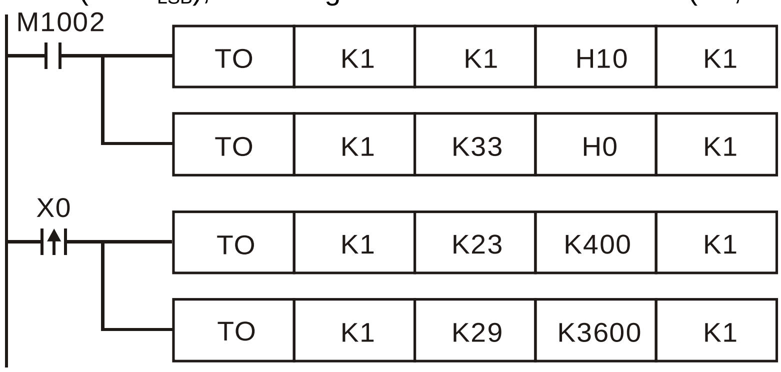
Example 3

◆ The characteristic curve of DVP02DA-H2 is adjusted. The offset for channel 2 is 0 mA (K0$_{LSB}$), and the gain for channel 2 is 10 mA (K1,000$_{LSB}$).

1. H18 is written into CR#1 in the analog output module whose number is 1, and channel 2 is set to mode 3 (current output: 0 mA~+20 mA).
2. H0 is written into CR#33. Channel 1~channel 2 can be tuned.
3. When X0 is turned from OFF to ON, the offset K0$_{LSB}$ is writtedn into CR#22, and the gain K1,000$_{LSB}$ is written into CR#28.

Example 4

◆ The characteristic curve of DVP02DA-H2 is adjusted. The offset for channel 2 is 2 mA (K400$_{LSB}$), and the gain for channel 2 is 18 mA (K3,600$_{LSB}$).



1. H10 is written into CR#1 in the analog output module whose number is 1, and channel 2 is set to mode 2 (current output: +4 mA~+20 mA).
2. H0 is written into CR#33. Channel 1~channel 2 can be tuned.
3. When X0 is turned from OFF to ON, the offset K400$_{LSB}$ is writtedn into CR#23, and the gain K K3,600$_{LSB}$ is written into CR#29.

API						Applicable model
87	D	ABS	P	D	Absolute value	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (9 steps)

ABS	Continuity instruction	ABSP	Pulse instruction

32-bit instruction (17 steps)

DABS	Continuity instruction	DABSP	Pulse instruction

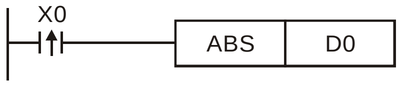
- Flag: None

Explanation

- **D**: Device whose absolute value will be gotten
- When the instruction ABS is executed, the absolute value of the value in **D** is gotten.
- Generally, the pulse instructions ABSP and DABSP are used.

Example

- When X0 is turned from OFF to ON, the absolute value of the value in D0 is gotten.



API						Applicable model
100		MODRD		S1 S2 n	Reading Modbus data	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*							*		
S_2					*	*							*		
n					*	*							*		

- Note: S_1 is in the range of K0 to K254.
  n is in the range of K1 to K6.
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

MODRD	Continuity instruction	-	-

32-bit instruction

-	-	-	-

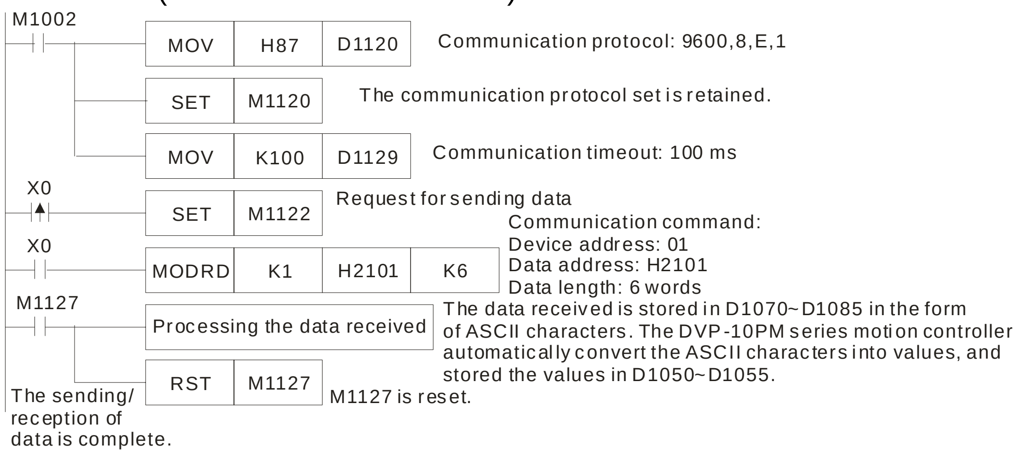
- Flags
  M1120~M1129 and M1140~M1143
  Please refer to the additional remark below.

Explanation

- **S_1**: Device address; **S_2**: Data address; **n**: Data length
- The instruction MODRD is used to drive peripheral equipment in a Modbus ACII/RUT mode. The RS-485 ports on Delta VFD series AC motor drives (except VFD-A series AC motor drives) conform to a Modbus communication format. Users can read data from a Delta AC motor drive by means of the instruction MODRD.
- **S_2** is a data address. If the data address specified is illegal, the device which is connected will respond with an error message, an error code will be stored in D1130 in the DVP-20PM series motion controller used, and M1141 will be ON.
- The data which is sent by a peripheral is stored in D1070~D1085. After a DVP-20PM series motion controller receives the data sent by a peripheral, it will automatically check whether the data received is correct. If an error occurs, M1140 will be ON.
- If an ASCII mode is used, the data sent by a peripheral will be ASCII characters, and the DVP-10PM series motion controller used will convert the data received into values, and store the values in D1050~D1055. If an RTU mode is used, D1050~D1055 will be invalid.
- If a DVP-20PM series motion controller sends correct data to a peripheral after M1140 or M1141 is turned ON, and the data with which the peripheral responds is correct, M1140 or M1141 will be reset.

Example 1

- A DVP-20PM series motion controller is connected to a VFD-B series AC motor drive (ASCII mode: M1143=OFF)



DVP-10PM series motion controller ⇨ VFD-B series AC motor drive: The DVP-10PM series motion controller sends “**01 03 2101 0006 D4**”.

VFD-B series AC motor drive ⇨ DVP-10PM series motion controller: The DVP-10PM series motion controller receives “**01 03 0C 0100 1766 0000 0000 0136 0000 3B**”.

Data transmission registers in the DVP-10PM series motion controller (message sent by the DVP-10PM series motion controller):

Register	Data		Description	
D1089 low	'0'	30 H	ADR 1	ADR (1,0): Address of the VFD-B series AC motor drive
D1089 high	'1'	31 H	ADR 0	
D1090 low	'0'	30 H	CMD 1	CMD (1,0): Command code
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	Quantity of data (count by the word)	
D1093 high	'0'	30 H		
D1094 low	'0'	30 H		
D1094 high	'6'	36 H		
D1095 low	'D'	44 H	LRC CHK 1	LRC CHK (0,1): Checksum
D1095 high	'4'	34 H	LRC CHK 0	

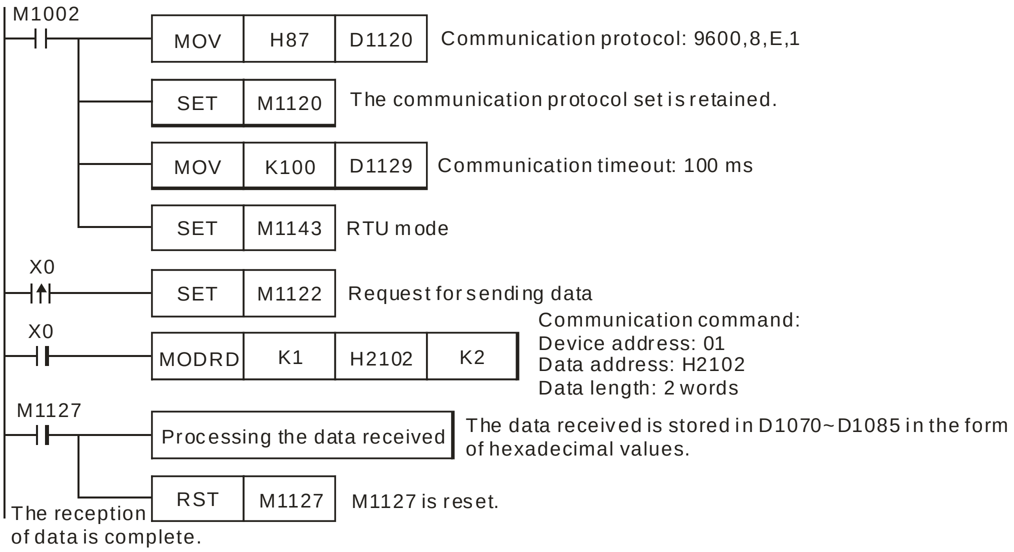
Data reception reigsters in the DVP-10PM series motion controller (message with which the VFD-B series AC motor drive responds):

Register	Data		Description	
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	Quantity of data (count by the byte)	
D1072 high	'C'	43 H		
D1073 low	'0'	30 H	Contents of the address 2101 H	The DVP-10PM series motion controller automatically converts the ASCII characters into values, and store the values in D1050. (D1050=0100 H)
D1073 high	'1'	31 H		
D1074 low	'0'	30 H		
D1074 high	'0'	30 H		
D1075 low	'1'	31 H	Contents of the address 2102 H	The DVP-10PM series motion controller automatically converts the ASCII characters into values, and store the values in D1051. (D1051=1766 H)
D1075 high	'7'	37 H		
D1076 low	'6'	36 H		
D1076 high	'6'	36 H		
D1077 low	'0'	30 H	Contents of the address 2103 H	The DVP-10PM series motion controller automatically converts the ASCII characters into values, and store the values in D1052. (D1052=0000 H)
D1077 high	'0'	30 H		
D1078 low	'0'	30 H		
D1078 high	'0'	30 H		
D1079 low	'0'	30 H	Contents of the address 2104 H	The DVP-10PM series motion controller automatically converts the ASCII characters into values, and store the values in D1053. (D1053=0000 H)
D1079 high	'0'	30 H		
D1080 low	'0'	30 H		
D1080 high	'0'	30 H		

Register	Data		Description	
D1081 low	‘0’	30 H	Contents of the address 2105 H	The DVP-10PM series motion controller automatically converts the ASCII characters into values, and store the values in D1054. (D1054=0136 H)
D1081 high	‘1’	31 H		
D1082 low	‘3’	33 H		
D1082 high	‘6’	36 H		
D1083 low	‘0’	30 H	Contents of the address 2106 H	The DVP-10PM series motion controller automatically converts the ASCII characters into values, and store the values in D1055. (D1055=0000 H)
D1083 high	‘0’	30 H		
D1084 low	‘0’	30 H		
D1084 high	‘0’	30 H		
D1085 low	‘3’	33 H	LRC CHK 1	
D1085 high	‘B’	42 H	LRC CHK 0	

Example 2

- A DVP-10PM series motion controller is connected to a VFD-B series AC motor drive (RTU mode: M1143=ON)



DVP-10PM series motion controller ⇨ VFD-B series AC motor drive: The DVP-10PM series motion controller sends “**01 03 2102 0002 6F F7**”.

VFD-B series AC motor drive ⇨ DVP-10PM series motion controller: The DVP-10PM series motion controller receives “**01 03 04 1770 0000 FE 5C**”.

Data transmission registers in the DVP-10PM series motion controller (message sent by the DVP-20PM series motion controller):

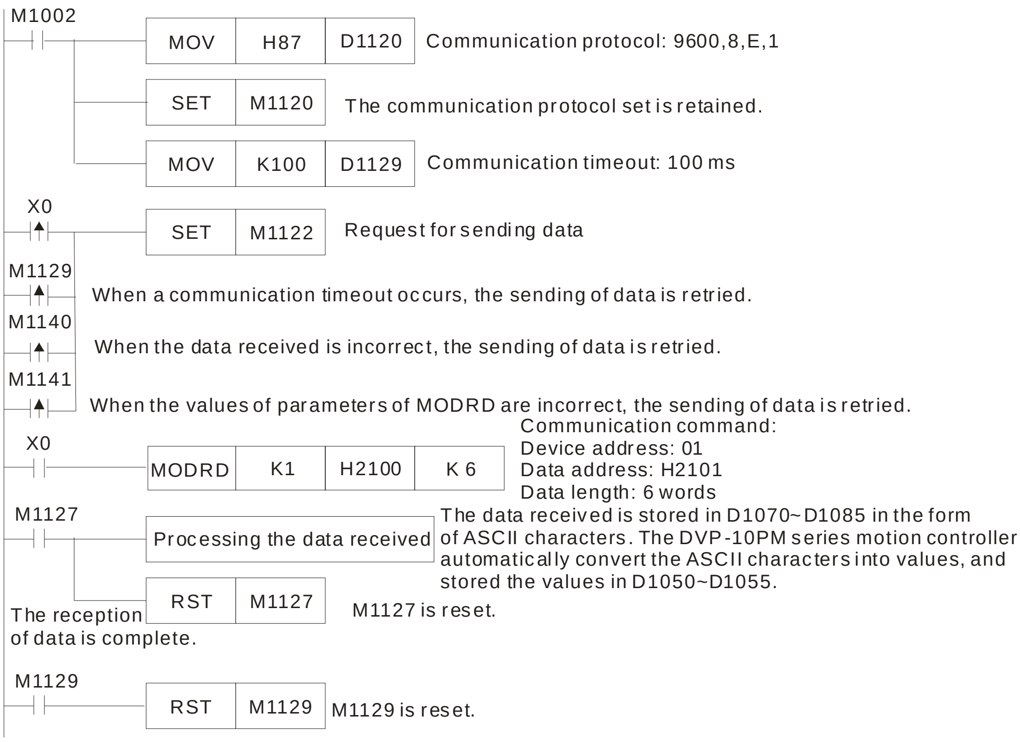
Register	Data	Description
D1089 low	01 H	Address
D1090 low	03 H	Function
D1091 low	21 H	Starting data address
D1092 low	02 H	
D1093 low	00 H	Quantity of Data (count by the word)
D1094 low	02 H	
D1095 low	6F H	CRC CHK Low
D1096 low	F7 H	CRC CHK High

Data reception registers in the DVP-10PM series motion controller (message with which the VFD-B series AC motor drive responds):

Register	Data	Description
D1070 low	01 H	Address
D1071 low	03 H	Function
D1072 low	04 H	Quantity of Data (count by the byte)
D1073 low	17 H	Contents of the address 2102 H
D1074 low	70 H	
D1075 low	00 H	Contents of the address 2103 H
D1076 low	00 H	
D1077 low	FE H	CRC CHK Low
D1078 low	5C H	CRC CHK High

Example 3

- If a communication timeout occurs, the data received is incorrect, or the values of parameters of the instruction MODRD are incorrect when a DVP-20PM series motion controller is connected to a VFD-B series AC motor drive (ASCII mode: M1143=OFF), the sending of data will be retried.
- When X0 is ON, the DVP-20PM series motion controller used reads the data in the data address H2100 in the VFD-B series AC motor drive whose device address is 01, and store the data in D1070~D1085 in the form of ASCII characters. The DVP-10PM series motion controller will automatically convert the ASCII characters in D1070~D1085 into values, and store the values in D1050~D1055.
- If a communication timeout occurs, M1129 will be ON. If M1129 is ON, M1122 will be set to ON.
- If the data received is incorrect, M1140 will be ON. If M1140 is ON, M1122 will be set to ON.
- If the values of parameters of the instruction MODWR is incorrect, M1141 will be ON. If M1141 is ON, M1122 will be set to ON.



Additional remark

- LDP/ANDP/ORP and LDF/ANDF/ORF can not precede the instruction MODRD (function code: H03), otherwise the data stored in data reception registers will be incorrect.
- The instruction can be used several times in a program, but one instruction is executed at a time.

API		Mnemonic		Operands	Function	Applicable model
101		MODWR		S1 S2 n	Writing Modbus data	20PM: ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*							*		
S_2					*	*							*		
n					*	*							*		

- Note: S_1 is in the range of K0 to K254.
  Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)

MODWR	Continuity instruction	-	-

32-bit instruction

-	-	-	-

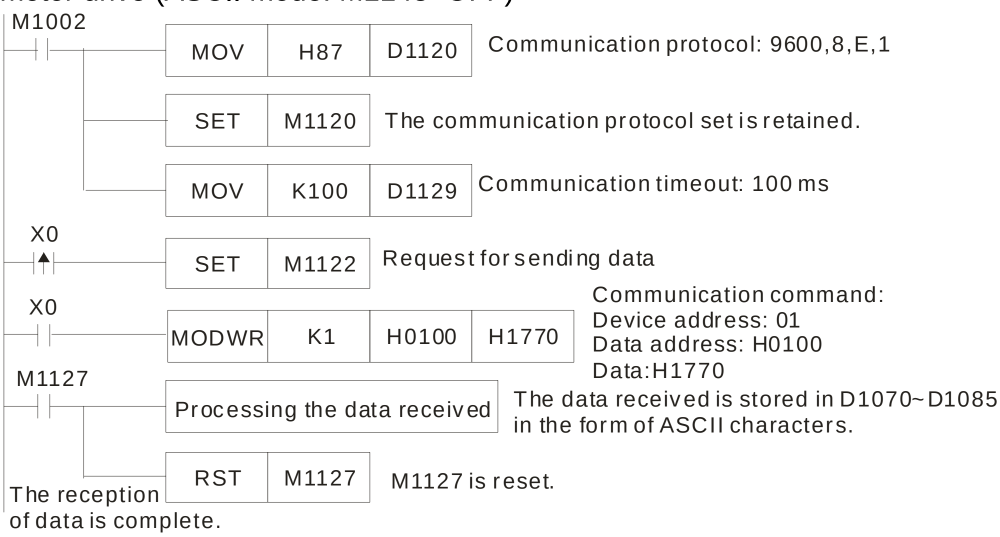
- Flags
  M1120~M1129 and M1140~M1143
  Please refer to the additional remark below.

Explanation

- **S_1**: Device address; **S_2**: Data address; **n**: Data which is written
- The instruction MODWR is used to drive peripheral equipment in a Modbus ACII/RUT mode. The RS-485 ports on Delta VFD series AC motor drives (except VFD-A series AC motor drives) conform to a Modbus communication format. Users can write data into a Delta AC motor drive by means of the instruction MODWR.
- **S_2** is a data address. If the data address specified is illegal, the device which is connected will respond with an error message, an error code will be stored in D1130 in the DVP-10PM series motion controller used, and M1141 will be ON. For example, the data address 8000H in a VFD-B series AC motor drive is illegal, and therefore M1141 is ON, and the value in D1130 is 2. Please refer to VFD-B User Manual for more information about error codes.
- The data which is sent by a peripheral is stored in D1070~D1076. After a DVP-10PM series motion controller receives the data sent by a peripheral, it will automatically check whether the data received is correct. If an error occurs, M1140 will be ON.
- If a DVP-10PM series motion controller sends correct data to a peripheral after M1140 or M1141 is turned ON, and the data with which the peripheral responds is correct, M1140 or M1141 will be reset.

Example 1

- A DVP-20PM series motion controller is connected to a VFD-B series AC motor drive (ASCII mode: M1143=OFF)



DVP-20PM series motion controller ⇨ VFD-B series AC motor drive: The DVP-20PM series motion controller sends “**01 06 0100 1770 71**”.

VFD-B series AC motor drive ⇨ DVP-20PM series motion controller: The DVP-20PM series motion controller receives “**01 06 0100 1770 71**”.

Data transmission registers in the DVP-20PM series motion controller (message sent by the DVP-20PM series motion controller):

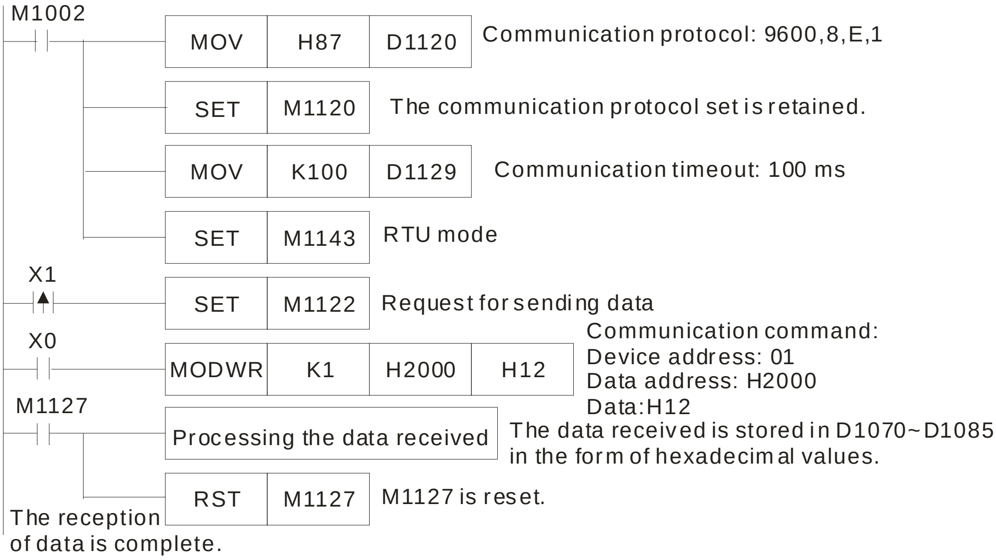
Register	Data		Description	
D1089 low	‘0’	30 H	ADR 1	ADR (1,0): Address of the VFD-B series AC motor drive
D1089 high	‘1’	31 H	ADR 0	
D1090 low	‘0’	30 H	CMD 1	CMD (1,0): Command code
D1090 high	‘6’	36 H	CMD 0	
D1091 low	‘0’	30 H	Data address	
D1091 high	‘1’	31 H		
D1092 low	‘0’	30 H		
D1092 high	‘0’	30 H		
D1093 low	‘1’	31 H	Data	
D1093 high	‘7’	37 H		
D1094 low	‘7’	37 H		
D1094 high	‘0’	30 H		
D1095 low	‘7’	37 H	LRC CHK 1	LRC CHK (0,1): Checksum
D1095 high	‘1’	31 H	LRC CHK 0	

Data reception reigsters in the DVP-10PM series motion controller (message with which the VFD-B series AC motor drive responds):

Register	Data		Description
D1070 low	‘0’	30 H	ADR 1
D1070 high	‘1’	31 H	ADR 0
D1071 low	‘0’	30 H	CMD 1
D1071 high	‘6’	36 H	CMD 0
D1072 low	‘0’	30 H	Data address
D1072 high	‘1’	31 H	
D1073 low	‘0’	30 H	
D1073 high	‘0’	30 H	
D1074 low	‘1’	31 H	Data
D1074 high	‘7’	37 H	
D1075 low	‘7’	37 H	
D1075 high	‘0’	30 H	
D1076 low	‘7’	37 H	LRC CHK 1
D1076 high	‘1’	31 H	LRC CHK 0

Example 2

- A DVP-20PM series motion controller is connected to a VFD-B series AC motor drive (RTU mode: M1143=ON)



DVP-20PM series motion controller ⇨ VFD-B series AC motor drive: The DVP-20PM series motion controller sends "**01 06 2000 0012 02 07**".

VFD-B series AC motor drive ⇨ DVP-20PM series motion controller: The DVP-20PM series motion controller receives "**01 06 2000 0012 02 07**".

Data transmission registers in the DVP-20PM series motion controller (message sent by the DVP-20PM series motion controller):

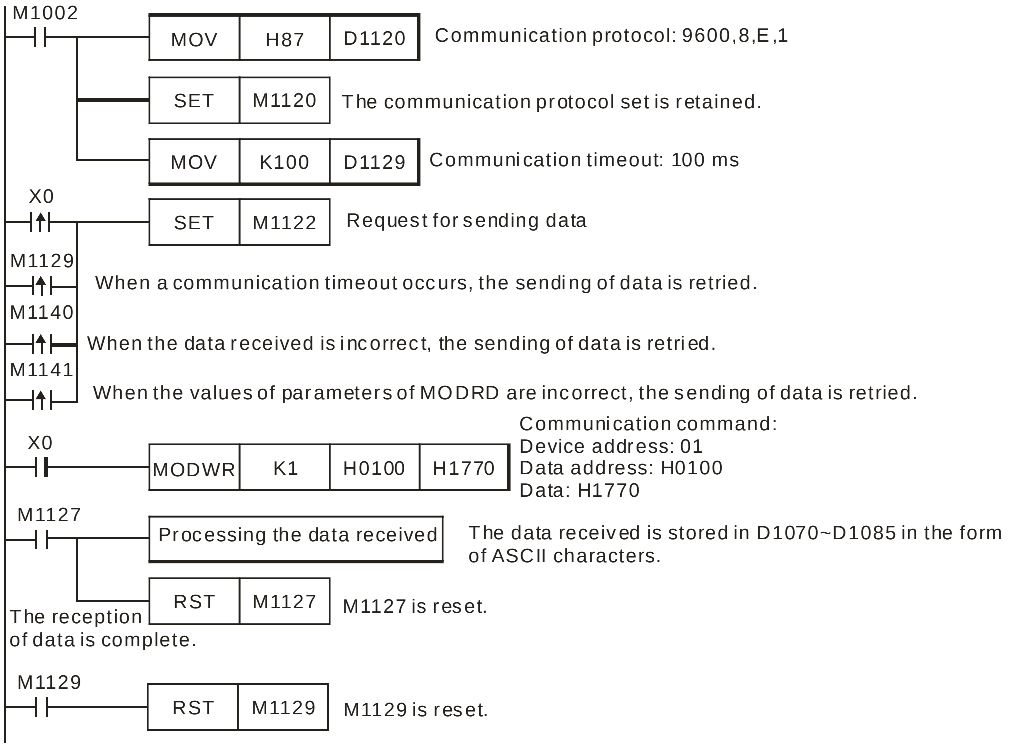
Register	Data	Description
D1089 low	01 H	Address
D1090 low	06 H	Function
D1091 low	20 H	Data address
D1092 low	00 H	
D1093 low	00 H	Data
D1094 low	12 H	
D1095 low	02 H	CRC CHK Low
D1096 low	07 H	CRC CHK High

Data reception reigsters in the DVP-10PM series motion controller (message with which the VFD-B series AC motor drive responds):

Register	Data	Description
D1070 low	01 H	Address
D1071 low	06 H	Function
D1072 low	20 H	Data address
D1073 low	00 H	
D1074 low	00 H	Data content
D1075 low	12 H	
D1076 low	02 H	CRC CHK Low
D1077 low	07 H	CRC CHK High

Example 3

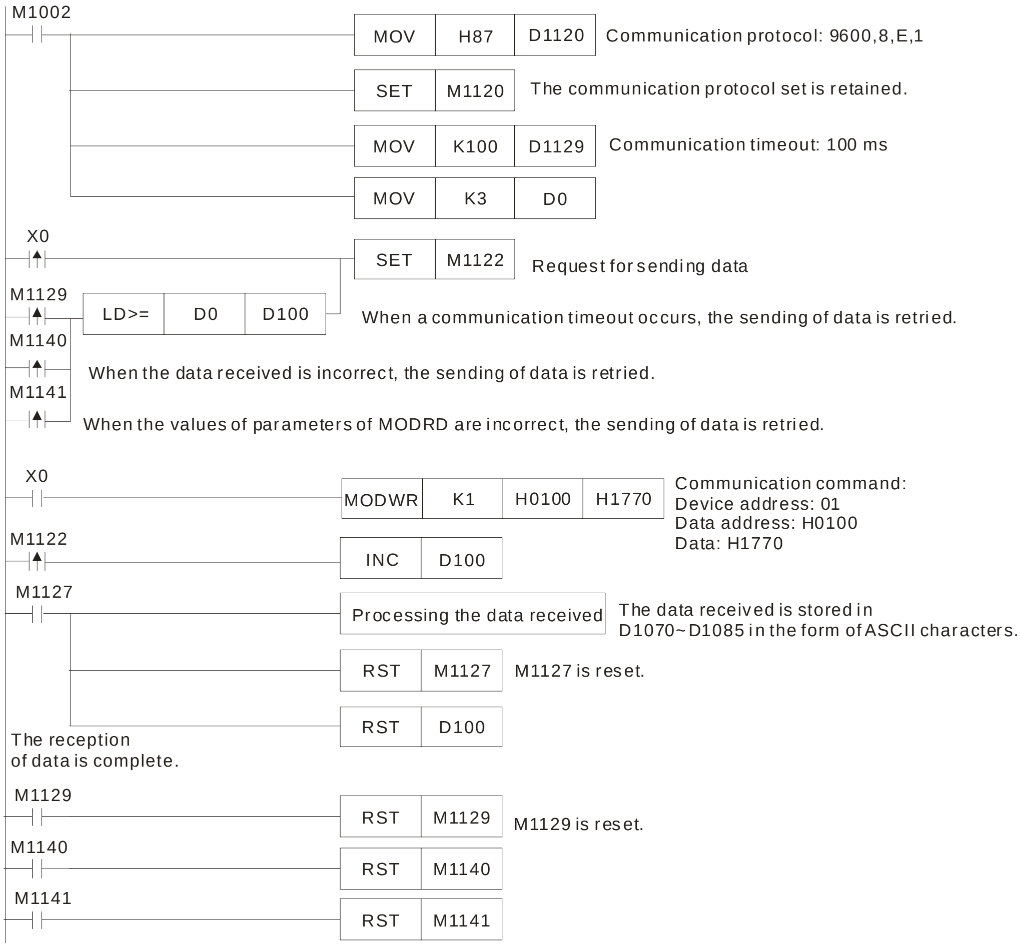
- If a communication timeout occurs, the data received is incorrect, or the values of parameters of the instruction MODRD are incorrect when a DVP-10PM series motion controller is connected to a VFD-B series AC motor drive (ASCII mode: M1143=OFF), the sending of data will be retried.
- When X0 is ON, the DVP-10PM series motion controller used write H1770 (K6000) into the data address H0100 in the VFD-B series AC motor drive whose device address is 01.
- If a communication timeout occurs, M1129 will be ON. If M1129 is ON, M1122 will be set to ON.
- If the data received is incorrect, M1140 will be ON. If M1140 is ON, M1122 will be set to ON.
- If the values of parameters of the instruction MODWR is incorrect, M1141 will be ON. If M1141 is ON, M1122 will be set to ON.



Example 4

- If a communication timeout occurs, the data received is incorrect, or the values of parameters of the instruction MODRD are incorrect when a DVP-10PM series motion controller is connected to a VFD-B series AC motor drive (ASCII mode: M1143=OFF), the sending of data will be retried. The number of times the sending of data is retired is stored in D0. The default value in D0 is 3. If communication is retried successfully, users can control the communication by means of triggering a condition.
- When X0 is ON, the DVP-10PM series motion controller used write H1770 (K6000) into the data address H0100 in the VFD-B series AC motor drive whose device address is 01.

- If a communication timeout occurs, M1129 will be ON. If M1129 is ON, M1122 will be set to ON. The number of times the sending of data is retired is stored in D0. The default value in D0 is 3.
- If the data received is incorrect, M1140 will be ON. If M1140 is ON, M1122 will be set to ON. The number of times the sending of data is retired is stored in D0. The default value in D0 is 3.
- If the values of parameters of the instruction MODWR is incorrect, M1141 will be ON. If M1141 is ON, M1122 will be set to ON. The number of times the sending of data is retired is stored in D0. The default value in D0 is 3.



Additional remark

- If LDP/ANDP/ORP or LDF/ANDF/ORF precedes the instruction MODRD (function code: H06 or H10), M1122 must be set to ON before MODRD is executed.
- The instruction can be used several times in a program, but one instruction is executed at a time.

API						Applicable model
110	D	ECMP	P	S1 S2 D	Comparing binary floating-point values	20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1					*								*		
S2					*								*		
D		*	*	*											

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DECMP and DECMPP are valid.
  The operand D occupies three consecutive devices.
  F represents a floating-point value. There is a decimal point in a floating-point value.

16-bit instruction

-	-	-	-

32-bit instruction (9 steps)

DECMP	Continuity instruction	DECMPP	Pulse instruction

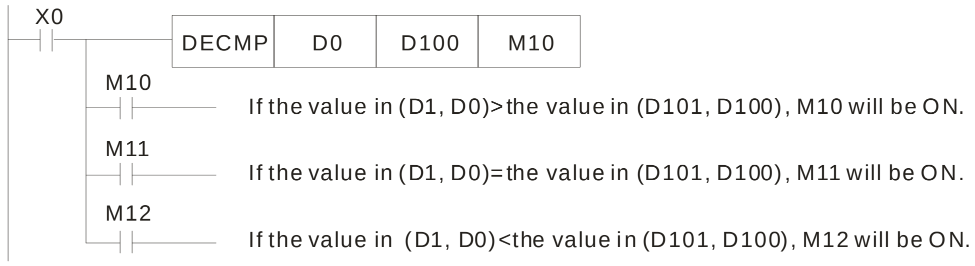
- Flag: None

Explanation

- S_1: Binary floating-point value 1; S_2: Binary floating-point value 2; **D**: Comparison result (D occupies three consecutive devices.)
- The instruction is used to compare the binary floating-point value in S_1 with that in S_2. The comparison result (> , = , or <)is stored in **D**.
- If S_1 is a floating-point value, the instruction will be used to compare the S_1 with the binary floating-point value in S_2. If S_2 is a floating-point value, the instruction will be used to compare the binary floating-point value in S_1 with S_2.

Example

- If the operand **D** is M10, M10, M11, and M12 will be occupied automatically.
- When X0 is ON, the instruction DECMP is executed, and M10, M11, or M12 is ON. When X0 is OFF, the execution of the instruction DECMP stops, and the states of M10, M11, and M12 remain unchanged.
- If users want to get the result that the value in (D1, D0)≧ the value in (D101, D100), they have to connect M10 and M11 in series. If users want to get the result that the value in (D1, D0)≦ the value in (D101, D100), they have to connect M11 and M12 in series. If users want to get the result that the value in (D1, D0)≠the value in (D101, D100), they have to connect M10, M11, and M12 in series.
- If users want to reset M10, M11, or M12, they can use the instruction RST or ZRST.



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
111	D	EZCP	P	S1 S2 S D	Binary floating-point zonal comparison	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*								*		
S_2					*								*		
S					*								*		
D		*	*	*											

- Note: The operand D occupies three consecutive devices.
  The value in S_1 is less than the value in S_2.
  Please refer to specifications for more information about device ranges.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Only the 32-bit instructions DEZCP and DEZCPP are valid.

16-bit instruction
- - - -

32-bit instruction (12 steps)
DEZCP Continuity instruction DEZCPP Pulse instruction

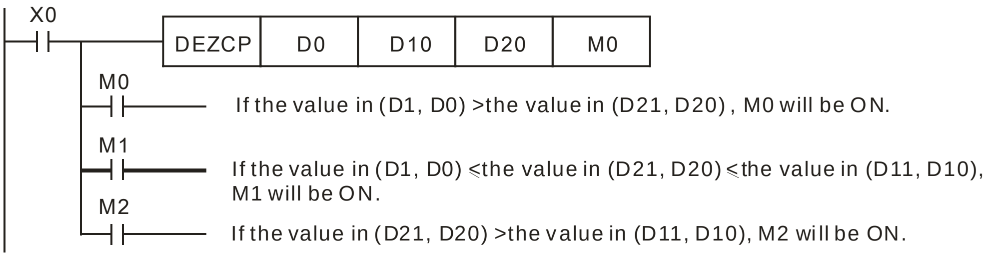
- Flag: None

Explanation

- **S_1**: Minimum binary floating-point value; **S_2**: Maximum binary floating-point value; **S**: Binary floating-point value; **D**: Comparison result (D occupies three consecutive devices.)
- The instruction is used to compare the binary floating-point value in **S** with that in **S_1**, and compare the binary floating-point value in **S** with that in **S_2**. The comparison result is stored in **D**.
- If **S_1** is a floating-point value, the instruction will be used to compare **S_1** with the binary floating-point value in **S_2**. If **S_2** is a floating-point value, the instruction will be used to compare the binary floating-point value in **S_1** with **S_2**.
- If the binary floating-point value in **S_1** is greater than that in **S_2**, the binary floating-point value in **S_1** will be taken as the maximum/minimum value during the execution of the instruction EZCP.
- If the operand **D** is M0, M0, M1, and M2 will be occupied automatically.

Example

- When X0 is ON, the instruction DEZCP is executed, and M0, M1, or M2 is ON. When X0 is OFF, the execution of the instruction DEZCP stops, and the states of M0, M1, and M2 remain unchanged.
- If users want to reset M0, M1, or M2, they can use the instruction RST or ZRST.



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
112	D	MOVR	P	S D	Transferring a floating-point value	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S															
D								*	*	*	*	*	*		

- Note: All devices can not be modified by V devices and Z devices.
  Please refer to specifications for more information about device ranges.
  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (7 steps)

-	-	-	-

32-bit instruction

DMOVR	Continuity instruction	DMOVR P	Pulse instruction

- Flag: None

Explanation

- **S**: Source; **D**: Destination
- The operand **S** can be a floating-point value.
- When the instruction is executed, the value in **S** is transferred to **D**. When the instruction is not executed, the value in **D** is unchanged.

Example

- When X0 is OFF, the value in (D11, D10) is unchanged. When X0 is ON, the value F1.2 is transferred to the data register (D11, D10).

API						Applicable model
116	D	RAD	P	S D	Converting a degree to a radian	20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Only the 32-bit instructions DRAD and DRADP are valid.

16-bit instruction

-	-	-	-

32 bit instruction (6 steps)

DRAD	Continuity instruction	DRADP	Pulse instruction

- Flags

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

- Please refer to the additional remark below.

Explanation

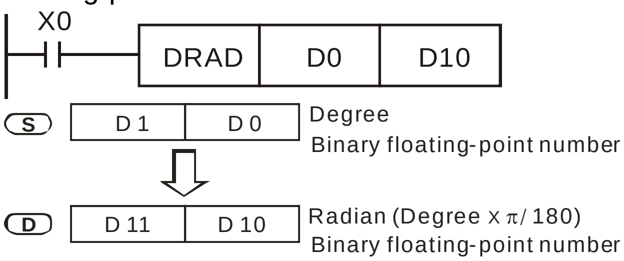
- **S**: Source (degree); **D**: Conversion result (radian)
- The equation below is used to convert a degree into a radian.

  Radian = Degree×(π /180)

- If the absolute value of a conversion result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of a conversion reuslt is less than the minimum floating-point value available, a borrow flag will be ON.
- If a converseion result is 0, a zero flag will be ON.

Example

- When X0 is ON, the degree in (D1, D0) is converted into a radian, and the conversion result is stored in (D11, D10). The radian in (D11, D10) is a binary floating-point value.



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
117	D	DEG	P	(S) (D)	Converting a radian to a degree	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Only the 32-bit instructions DDEG and DDEGP are valid.

16-bit instruction

\- - - -

32-bit instruction (6 steps)

DDEG Continuity instruction DDEGP Pulse instruction

- Flags

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

- Please refer to the additional remark below.

Explanation

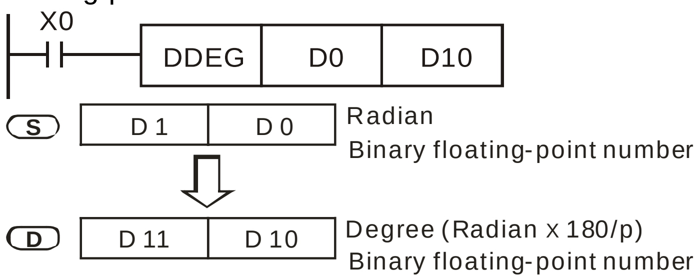
- **S**: Source (radian); **D**: Conversion result (degree)
- The equation below is used to convert a radian into a degree.

  Degree = Radian×(180/π)

- If the absolute value of a conversion result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of a conversion reuslt is less than the minimum floating-point value available, a borrow flag will be ON.
- If a conversion result is 0, a zero flag will be ON.

Example

- When X0 is ON, the radian in (D1, D0) is converted into a degree, and the conversion result is stored in (D11, D10). The degree in (D11, D10) is a binary floating-point value.



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
120	D	EADD	P	S1 S2 D	Binary floating-point addition	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*								*		
S_2					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Only the 32-bit instructions DEADD and DEADDP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (9 steps)

DEADD	Continuity instruction	DEADDP	Pulse instruction

- Flags

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

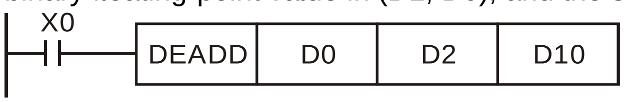
- Please refer to the additional remark below.

Explanation

- **S_1**: Augend; **S_2**: Addend; **D**: Sum
- The binary floating-point value in **S_2** is added to the binary floating-point value in **S_1**, and the sum is stored in **D**.
- If **S_1** is a floating-point value, the instruction will be used to add the binary floating-point value in **S_2** to **S_1**. If **S_2** is a floating-point value, the instruction will be used to add **S_2** to the binary floating-point value in **S_1**.
- **S_1** and **S_2** can be the same register. If the instruction DEADD is used under the circumstances, the value in the register is added to itself whenever the conditional contact is ON in a scan cycle. Generally, the pulse instruction DEADDP is used.
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON.
- If an operation result is 0, a zero flag will be ON.

Example 1

- When X0 is ON, the binary floating-point value in (D3, D2) is added to the binary floating-point value in (D1, D0), and the sum is stored in (D11, D10).



Example 2

- When X0 is ON, F1234.0 is added to the binary floating-point value in (D11, D10), and the sum is stored in (D21, D20).

Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
121	D	ESUB	P	S1 S2 D	Binary floating-point subtraction	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*								*		
S_2					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Only the 32-bit instructions DEADD and DEADDP are valid.

16-bit instruction
- - - -

32-bit instruction (9 steps)
DESUB Continuity instruction DESUBP Pulse instruction

- Flags

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

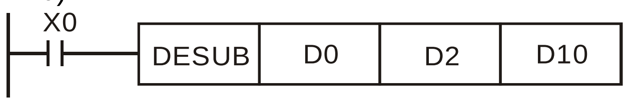
- Please refer to the additional remark below.

Explanation

- S_1: Minuend; S_2: Subtrahend; **D**: Difference
- The binary floating-point value in S_2 is subtracted from the binary floating-point value in S_1, and the difference is stored in **D**.
- If S_1 is a floating-point value, the instruction will be used to subtract the binary floating-point value in S_2 from S_1. If S_2 is a floating-point value, the instruction will be used to subtract S_2 from the binary floating-point value in S_1.
- S_1 and S_2 can be the same register. If the instruction DESUB is used under the circumstances, the value in the register is subtracted from itself whenever the conditional contact is ON in a scan cycle. Generally, the pulse instruction DESUBP is used.
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON.
- If an operation result is 0, a zero flag will be ON.

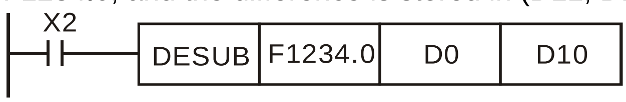
Example 1

- When X0 is ON, the binary floating-point value in (D3, D2) is subtracted from the binary floating-point value in (D1, D0), and the difference is stored in (D11, D10).



Example 2

- When X2 is ON, the binary floating-point value in (D1, D0) is subtracted from F1234.0, and the difference is stored in (D11, D10).



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
122	D	EMUL	P	(S1) (S2) (D)	Binary floating-point multiplication	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*								*		
S_2					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Onlyt the 32-bit instructions DEMUL and DEMULP are valid.

16-bit instruction

- - - -

32-bit instruction (9 steps)

DEMUL Continuity instruction DEMULP Pulse instruction

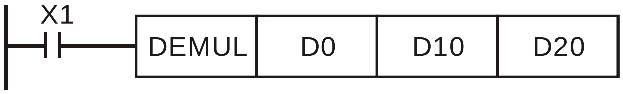
- Flags
  Ox O100
  M1810 M1970 Carry flag
- Please refer to the additional remark below.

Explanation

- **S_1**: Multiplicand; **S_2**: Multiplier; **D**: Product
- The binary floating-point value in **S_1** is multiplied by the binary floating-point value in **S_2**, and the product is stored in **D**.
- If **S_1** is a floating-point value, the instruction will be used to multiply **S_1** by the binary floating-point value in **S_2**. If **S_2** is a floating-point value, the instruction will be used to multiply the binary floating-point value in **S_1** by **S_2**.
- **S_1** and **S_2** can be the same register. If the instruction DEMUL is used under the circumstances, the value in the register is multiplied by itself whenever the conditional contact is ON in a scan cycle. Generally, the pulse instruction DEMULP is used.
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON.
- If an operation result is 0, a zero flag will be ON.

Example 1

- When X1 is ON, the binary floating-point value in (D1, D0) is multiplied by the binary floating-point value in (D11, D10), and the product is stored in (D21, D20).



Example 2

- When X2 is ON, F1234.0 is multiplied by the binary floating-point value in (D1, D0), and the product is stored in (D11, D10).

Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
123	D	EDIV	P	S1 S2 D	Binary floating-point division	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*								*		
S_2					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Only the 32-bit instructions DEDIV and DEDIVP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (9 steps)

DEDIV	Continuity instruction	DEDIVP	Pulse instruction

- Flags

Ox	O100	
M1808	M1968	Zero flag
M1809	M1969	Borrow flag
M1810	M1970	Carry flag
M1793	M1953	Operation error flag

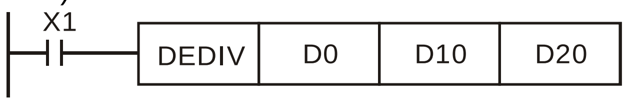
- Please refer to the additional remark below.

Explanation

- **S_1**: Dividend; **S_2**: Divisor; **D**: Quotient and remainder
- The binary floating-point value in **S_1** is divided by the binary floating-point value in **S_2**, and the quotient is stored in **D**.
- If **S_1** is a floating-point value, the instruction will be used to divide **S_1** by the binary floating-point value in **S_2**. If **S_2** is a floating-point value, the instruction will be used to divide the binary floating-point value in **S_1** by **S_2**.
- If the value in **S_2** is 0, an operation error will occur, the instruciton will not be executed, an operation error flag will be ON, and the error code H0E19 will appear.
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON.
- If an operation result is 0, a zero flag will be ON.

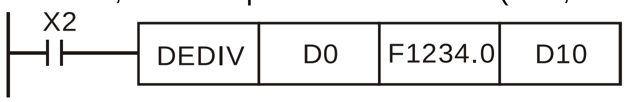
Example 1

- When X1 is ON, the binary floating-point value in (D1, D0) is divided by the binary floating-point value in (D11, D10), and the quotient is stored in (D21, D20).



Example 2

- When X2 is ON, the binary floating-point value in (D1, D0) is divided by F1234.0, and the quotient is stored in (D11, D10).



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API	D	EXP	P	S D	Exponent of a binary floating-point value	Applicable model
124	D	EXP	P	(S) (D)	Exponent of a binary floating-point value	20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Only the 32-bit instructions DEXP and DEXPP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (6 steps)

DEXP	Continuity instruction	DEXPP	Pulse instruction

- Flags

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

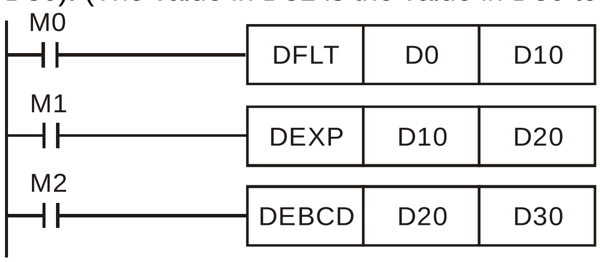
- Please refer to the additional remark below.

Explanation

- **S**: Source device; **D**: Device in which an operation result is stored
- EXP[D+1, D]=[**S**+1，**S**]. e is a base (e=2.71828), and **S** is an exponent.
- The value in **S** can be a positive value or a negative value. **D** must be a 32-bit register, and the value in **S** must be a floating-point value.
- The value in **D** is e^{S}. (e is 2.71828, and **S** represents a source value.)
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON.
- If an operation result is 0, a zero flag will be ON.

Example

- When M0 is ON, the value in (D1, D0) is converted into a binary floating-point value, and the conversion result is stored in (D11, D10).
- When M1 is ON, the exponentiation with the value in (D11, D10) as an exponent is performed. The result is a binary floating-point number, and is stored in (D21, D20).
- When M2 is ON, the binary floating-point value in (D21, D20) is converted into a decimal floating-point value, and the conversion result is stored in (D31, D30). (The value in D31 is the value in D30 to the power of 10.)



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
125	D	LN	P	S D	Natural logarithm of a binary floating-point value	20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Only the 32-bit instructions DLN and DLNP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (6 steps)

DLN	Continuity instruction	DLNP	Pulse instruction

- Flags

Ox	O100	
M1808	M1968	Zero flag
M1809	M1969	Borrow flag
M1810	M1970	Carry flag
M1793	M1953	Operation error flag

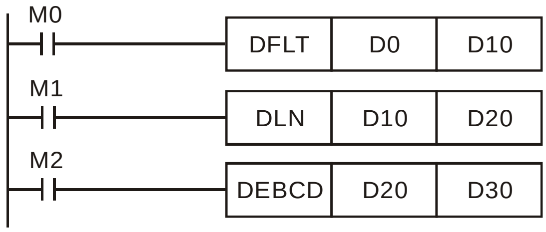
- Please refer to the additional remark below.

Explanation

- **S**: Source device; **D**: Device in which an operation result is stored
- The natural logarithm of the value in **S** is calculated.
  Ln[**S**+1, **S**]=[**D**+1, **D**]
- The value in **S** can only be a positive value. **D** must be a 32-bit register, and the value in **S** must be a floating-point value.
- If the value in **S** is not a positive value, an operation error will occur, the instruciton will not be executed, an operation error flag will be ON, and the error code H0E19 will appear.
- e^{D}=**S**.→The value in **D**=ln**S** (**S**: Source device)
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON.
- If an operation result is 0, a zero flag will be ON.

Example

- When M0 is ON, the value in (D1, D0) is converted into a binary floating-point value, and the conversion result is stored in (D11, D10).
- When M1 is ON, the natural logarithm of the floating-point value in (D11, D10) is calculated, and the operation result is stored in (D21, D20).
- When M2 is ON, the binary floating-point value in (D21, D20) is converted into a decimal floating-point value, and the conversion result is stored in (D31, D30). (The value in D31 is the value in D30 to the power of 10.)



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						
126	D	LOG	P	S1 S2 D	Logarithm of a binary floating-point value	Applicable model: 20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*								*		
S_2					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Only the 32-bit instructions DLOG and DLOGP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (9 steps)

DLOG	Continuity instruction	DLOGP	Pulse instruction

- Flags

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

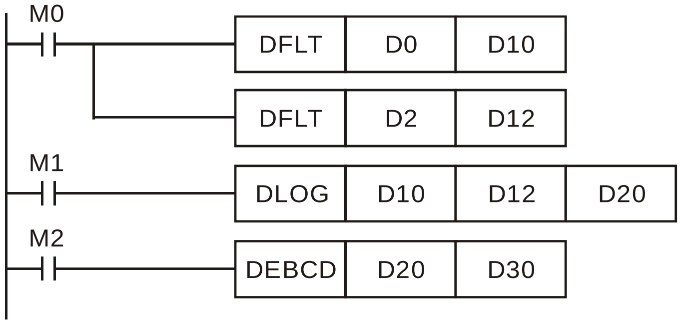
- Please refer to the additional remark below.

Explanation

- **S_1**: Device in which the base is stored; **S_2**: Source device; **D**: Device in which an operation result is stored
- The logarithm of the value in **S_2** with respect to the value in **S_1** is calculated, and the operation result is stored in **D**.
- The values in **S_1** and **S_2** can only be positive values. **D** must be a 32-bit register, and the values in **S_1** and **S_2** must be floating-point values.
- $S_1^{D}=S_2 \rightarrow D=\log_{S1}S_2$
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON.
- If an operation result is 0, a zero flag will be ON.

Example

- When M0 is ON, the values in (D1, D0) and (D3, D2) are converted into binary floating-point values, and the conversion results are stored in (D11, D10) and (D13, D12) respectively.
- When M1 is ON, the logarithm of the binary floating-point value in (D13, D12) with respect to the binary floating-point value in (D11, D10) is calculated, and the operation result is stored in (D21, D20).
- When M2 is ON, the binary floating-point value in (D21, D20) is converted into a decimal floating-point value, and the conversion result is stored in (D31, D30). (The value in D31 is the value in D30 to the power of 10.)



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API	Mnemonic		Operands	Function	Applicable model
127	D ESQR	P	S D	Square root of a binary floating-point value	20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  S is greater than or equal to 0.
  F represents a floating-point value. There is a decimal point in a floating-point value.
  Only the 32-bit instructions DESQR and DESQRP are valid.

16-bit instruction
- - - -

32-bit instruction (6 steps)
DESQR Continuity instruction DESQRP Pulse instruction

- Flags

Ox	O100	
M1808	M1968	Zero flag
M1793	M1953	Operation error flag

- Please refer to the additional remark below.

Explanation

- **S**: Source device; **D**: Device in which a result is stored
- The square root of the binary floating-point value in **S** is calculated, and the result is stored in **D**.
- If **S** is a floating-point value, the instruction will be used to calculate the floating-point value.
- If an operation result is 0, a zero flag will be ON.
- If the value in **S** is not a positive value, an operation error will occur, the instruciton will not be executed, an operation error flag will be ON, and the error code H0E19 will appear.

Example 1

- When X0 is ON, the square root of the binary floating-point value in (D1, D0) is calculated, and the result is stored in (D11, D10).

$\sqrt{(D1, D0)}$ → (D11, D10)

Binary floating-point number → Binary floating-point number

Example 2

- When X2 is ON, the square root of F1234.0 is calculated, and the result is stored in (D11, D10).

Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
128	D	POW	P	S1 S2 D	Power of a floating-point value	20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*								*		
S_2					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DPOW and DPOWP are valid.
  F represents a floating-point value. There is a decimal point in a floating-point value.

16-bit instruction
\- - - -

32-bit instruction (9 steps)
DPOW Continuity instruction DPOWP Pulse instruction

- Flags

Ox	O100	
M1808	M1968	Zero flag
M1809	M1969	Borrow flag
M1810	M1970	Carry flag
M1793	M1953	Operation error flag

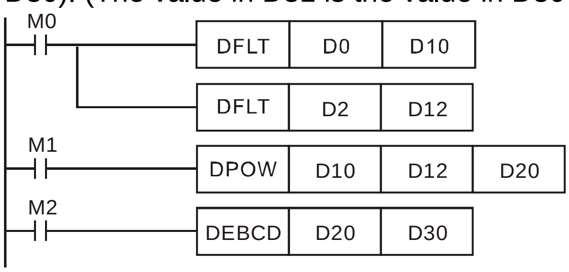
- Please refer to the additional remark below.

Explanation

- **S_1**: Device in which a base is stored; **S_2**: Device in which a power is stored; **D**: Device in which the operation result is stored
- The binary floating-point value in **S_1** is raised to the power of the value in **S_2**, and the operation result is stored in **D**.
  D=POW[**S_1**+1, **S_1**]$^{[S2+1,\ S2]}$
- The value in **S_1** can only be a positive value, whereas the value in **S_2** can be a positive value or a negative value. **D** must be a 32-bit register, and the values in **S_1** and **S_2** must be floating-point values.
- If the values in **S_1** and **S_2** are invalid, an operation error will occur, the instruciton will not be executed, an operation error flag will be ON, and the error code H0E19 will appear.
- If the absolute value of an operation result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of an operation reuslt is less than the minimum floating-point value available, a borrow flag will be ON.
- If an operation result is 0, a zero flag will be ON.

Example

- When M0 is ON, the values in (D1, D0) and (D3, D2) are converted into binary floating-point values, and the conversion results are stored in (D11, D10) and (D13, D12) respectively.
- When M1 is ON, the binary floating-point value in (D11, D10) is raised to the power of the binary floating-point value in (D13, D12), and the operation result is stored in (D21, D20).
- When M2 is ON, the binary floating-point value in (D21, D20) is converted into a decimal floating-point value, and the conversion result is stored in (D31, D30). (The value in D31 is the value in D30 to the power of 10.)



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
129	D	INT	P	S D	Converting a binary floating-point value into a binary integer	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S													*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DINT and DINTP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (5 steps)

DINT	Continuity instruction	DINTP	Pulse instruction

- Flags

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

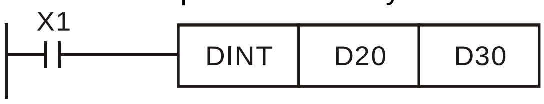
- Please refer to the explanation below.

Explanation

- **S**: Source device; **D**: Conversion result
- The binary floating-point value in **S** is converted into a binary value. The integer part of the binary value is stored in **D**, and the fractional part of the binary value is dropped.
- The instruction is the opposite of API 49 DFLT.
- If a conversion result is 0, a zero flag will be ON.
  If the fractional part of a conversion result is dropped, a borrow flag will be ON.
  If a converesion result is not in the range of -2,147,483,648 to 2,147,483,647, a carry flag will be ON.

Example

- When X1 is ON, the binary floating-point value in (D21, D20) is converted into a binary value. The integer part of the binary value is stored in (D31, D30), and the fractional part of the binary value is dropped.



API						Applicable model
130	D	SIN	P	S D	Sine of a binary floating-point value	20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: $0^{o} \leqq$ Degree $\leqq 360^{0}$

  Please refer to specifications for more information about device ranges.

  F represents a floating-point value. There is a decimal point in a floating-point value.

  Only the 32-bit instructions DSIN and DSINP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (6 steps)

DSIN	Continuity instruction	DSINP	Pulse instruction

- Flags

Ox	O100	
M1808	M1968	Zero flag
M1760	M1920	Radian/Degree flag

- Please refer to the additional remark below.

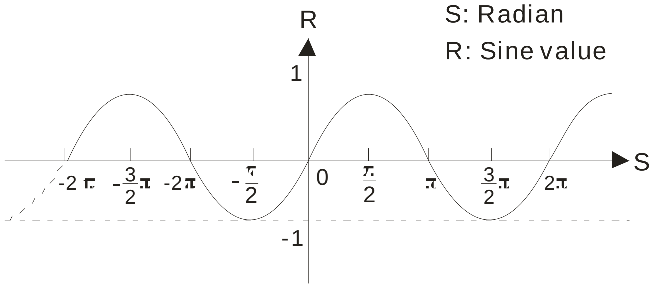
Explanation

- **S**: Source value; **D**: Sine value
- Whether the source value in **S** is a radian or a degree depends on the state of a radian/degree flag.
- If a radian/degree flag is OFF, the source value in **S** is a radian.

  Radian=Degree×π /180.
- If a radian/degree flag is ON, the source value in **S** is a degree.

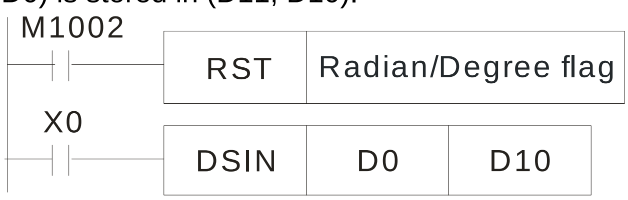
  ($0^{o} \leqq$ Degree $\leqq 360^{0}$)
- If an operation result is 0, a zero flag will be ON.
- The sine of the source value in **S** is stored in **D**.

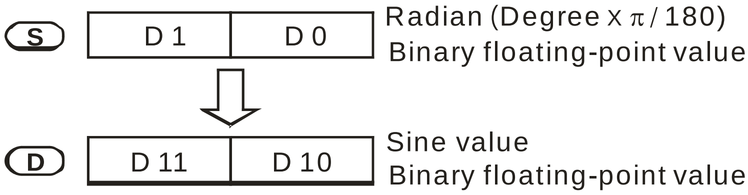
The relation between radians and sine values is shown below.



Example 1

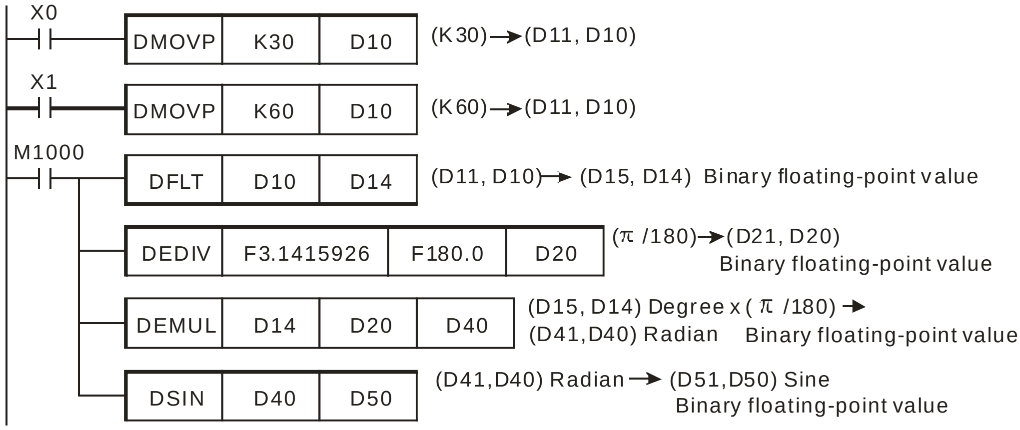
- A radian/degree flag is reset to OFF. The binary floating-point value in (D1, D0) is a radian. When X0 is ON, the sine of the binary floating-point value in (D1, D0) is stored in (D11, D10).





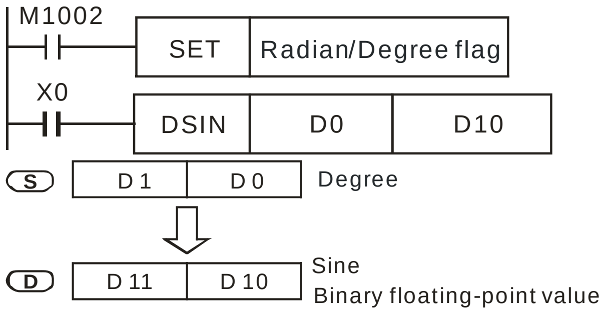
Example 2

- A radian/degree flag is OFF. A degree is set by means of X0 or X1. After the degree is converted into a radian, the sine of the radian will be calculated.



Example 3

- A radian/degree flag is set to ON. The value in (D1, D0) is a degree in the range of 0° to 360°. When X0 is ON, the sine of the value in (D1, D0) is stored in (D11, D10). The value in (D11, D10) is a binary floating-point value.



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
131	D	COS	P	S D	Cosine of a binary floating-point number	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: $0^o \leqq$ Degree $\leqq 360^0$

  Please refer to specifications for more information about device ranges.

  F represents a floating-point value. There is a decimal point in a floating-point value.

  Only the 32-bit instructions DCOS and DCOSP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (6 steps)

DCOS	Continuity instruction	DCOSP	Pulse instruction

- Flags

Ox	O100	
M1808	M1968	Zero flag
M1760	M1920	Radian/Degree flag

- Please refer to the additional remark below.

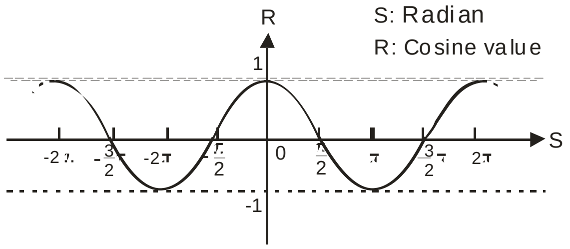
Explanation

- **S**: Source value; **D**: Cosine value
- Whether the source value in **S** is a radian or a degree depends on the state of a radian/degree flag.
- If a radian/degree flag is OFF, the source value in **S** is a radian.

  Radian=Degree×π /180.
- If a radian/degree flag is ON, the source value in **S** is a degree.

  ($0^o \leqq$ Degree $\leqq 360^0$)
- If an operation result is 0, a zero flag will be ON.
- The cosine of the source value in **S** is stored in **D**.

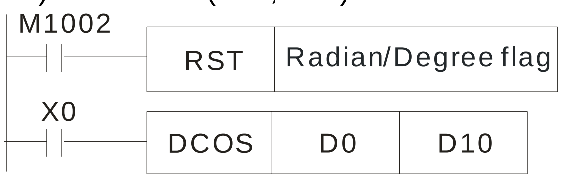
  The relation between radians and cosine values is shown below.

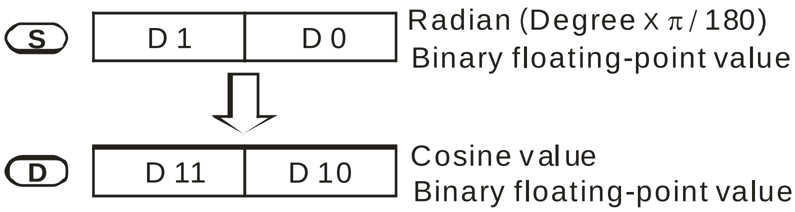


- Radian/Degree flag: If a radian/degree flag is OFF, the source value in **S** is a radian. If a radian/degree flag is ON, the source value in **S** is a degree in the range of 0° to 360°.

Example 1

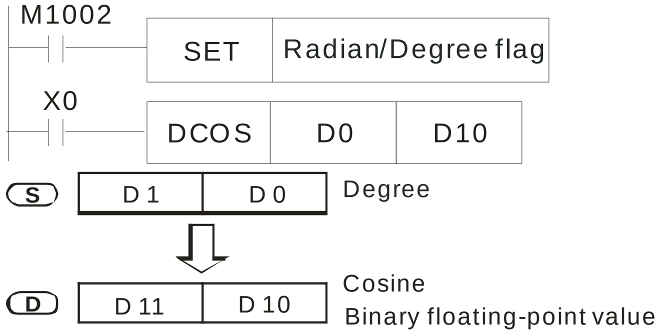
- A radian/degree flag is reset to OFF. The binary floating-point value in (D1, D0) is a radian. When X0 is ON, the cosine of the binary floating-point value in (D1, D0) is stored in (D11, D10).





Example 2

- A radian/degree flag is set to ON. The value in (D1, D0) is a degree in the range of 0° to 360°. When X0.0 is ON, the cosine of the value in (D1, D0) is stored in (D11, D10). The value in (D11, D10) is a binary floating-point value.



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API		TAN		Ⓢ Ⓓ	Tangent of a binary floating-point value	Applicable model
132	D		P			20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: $0^o \leqq$ Degree $\leqq 360^0$

  Please refer to specifications for more information about device ranges.

  F represents a floating-point value. There is a decimal point in a floating-point value.

  Only the 32-bit instructions DTAN and DTANP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (6 steps)

DTAN	Continuity instruction	DTANP	Pulse instruction

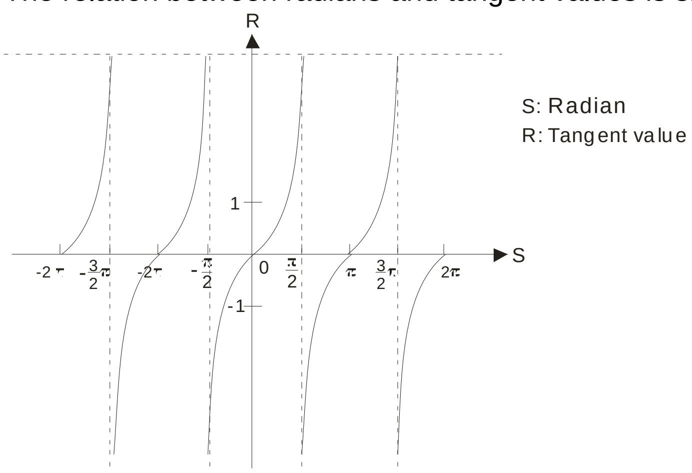
- Flags

Ox	O100	
M1808	M1968	Zero flag
M1760	M1920	Radian/Degree flag

- Please refer to the additional remark below.

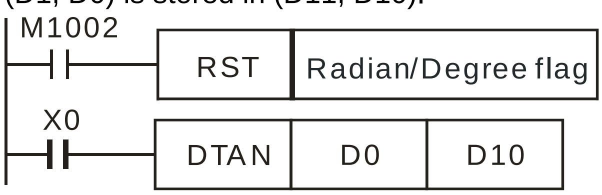
Explanation

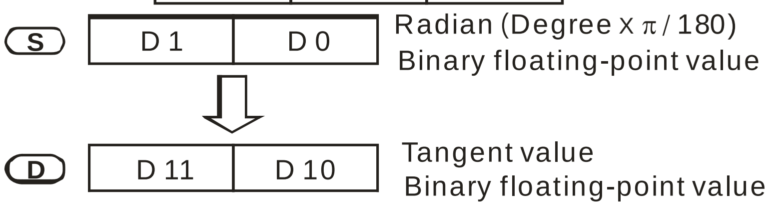
- **S**: Source value; **D**: Tangent value
- Whether the source value in **S** is a radian or a degree depends on the state of a radian/degree flag.
- If a radian/degree flag is OFF, the source value in **S** is a radian.
  Radian=Degree×π /180.
- If a radian/degree flag is ON, the source value in **S** is a degree.
  ($0^o \leqq$ Degree $\leqq 360^0$)
- If an operation result is 0, a zero flag will be ON.
- The tangent of the source value in **S** is stored in **D**.
  The relation between radians and tangent values is shown below.



Example 1

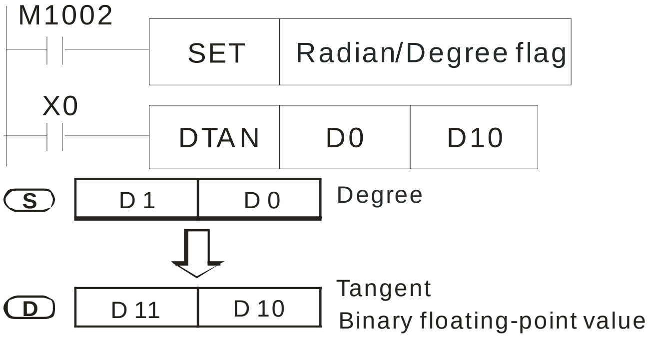
- A radian/degree flag is reset to OFF. The binary floating-point value in (D1, D0) is a radian. When X0 is ON, the tangent of the binary floating-point value in (D1, D0) is stored in (D11, D10).





Example 2

◆ A radian/degree flag is set to ON. The value in (D1, D0) is a degree in the range of 0° to 360°. When X0 is ON, the tangent of the value in (D1, D0) is stored in (D11, D10). The value in (D11, D10) is a binary floating-point value.



Additional remark

◆ Please refer to section 5.3 for more information about performing operations on floating-point values.

API						
133	D	ASIN	P	S D	Arcsine of a binary floating-point value	Applicable model: 20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DASIN and DASINP are valid.
  F represents a floating-point value. There is a decimal point in a floating-point value.

16-bit instruction
- - - -

32-bit instruction (6 steps)
DASIN Continuity instruction DASINP Pulse instruction

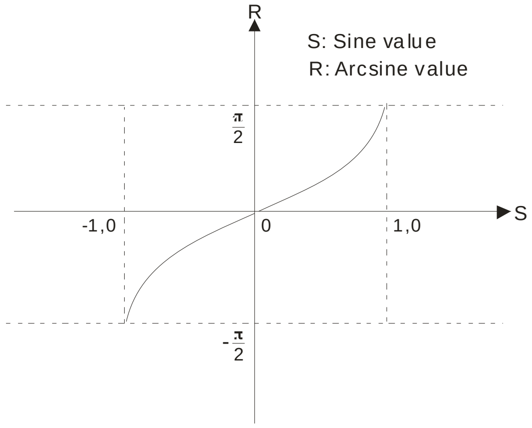
- Flags

Ox	O100	
M1808	M1968	Zero flag
M1793	M1953	Operation error flag

- Please refer to the additional remark below.

Explanation

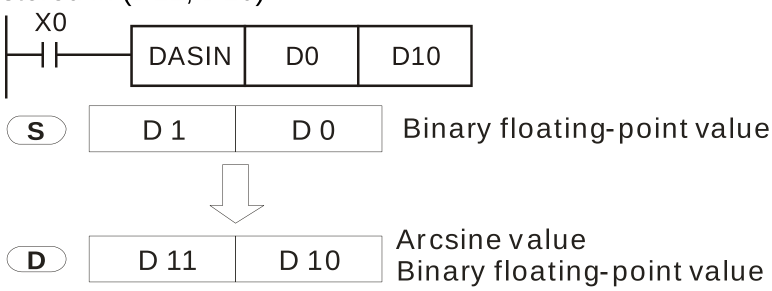
- **S**: Source value (binary floating-point value); **D**: Arcsine value
- Arcsine value=$\sin^{-1}$
  The relation between sine values and arcsine values is shown below.



- The decimal floating-point value into which the sine value in **S** is converted can only be in the range of -1.0 to +1.0. If it is not in the range, the instruction will not be executed, an operation error flag will be ON, and the error code H0E19 will appear.
- If a conversion result is 0, a zero flag will be ON.

Example

- When X0 is ON, the arcsine of the binary floating-point value in (D1, D0) is stored in (D11, D10).



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API	D	ACOS	P	S D	Arccosine of a binary floating-point value	Applicable model
134						20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DACOS and DACOSP are valid.
  F represents a floating-point value. There is a decimal point in a floating-point value.

16-bit instruction
- - - -

32-bit instruction (6 steps)
DACOS Continuity instruction DACOSP Pulse instruction

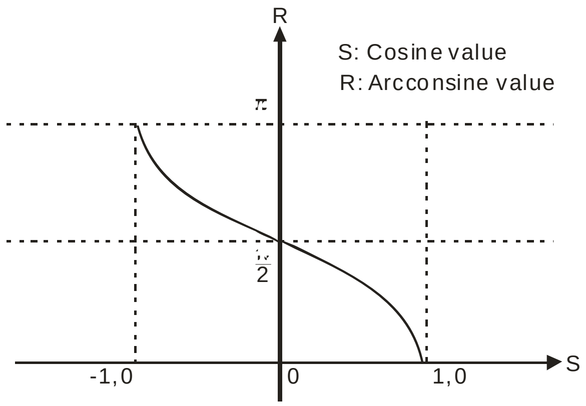
- Flags

Ox	O100	
M1808	M1968	Zero flag
M1793	M1953	Operation error flag

- Please refer to the additional remark below.

Explanation

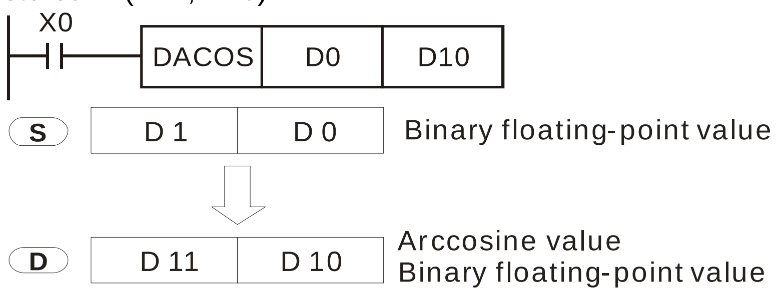
- **S**: Source value (binary floating-point value); **D**: Arccosine value
- Arccosine value=$\cos^{-1}$
  The relation between cosine values and arccosine values is shown below.



- The decimal floating-point value into which the cosine value in **S** is converted can only be in the range of -1.0 to +1.0. If it is not in the range, the instruction will not be executed, an operation error flag will be ON, and the error code H0E19 will appear.
- If a conversion result is 0, a zero flag will be ON.

Example

- When X0 is ON, the arccosine of the binary floating-point value in (D1, D0) is stored in (D11, D10).



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API		ATAN			Arctangent of a binary floating-point value	Applicable model
135	D		P	S D		20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instrucitons DATAN and DATANP are valid.
  F represents a floating-point value. There is a decimal point in a floating-point value.

16-bit instruction

-	-	-	-

32-bit instruction (6 steps)

DATAN	Continuity instruction	DATANP	Pulse instruction

- Flags

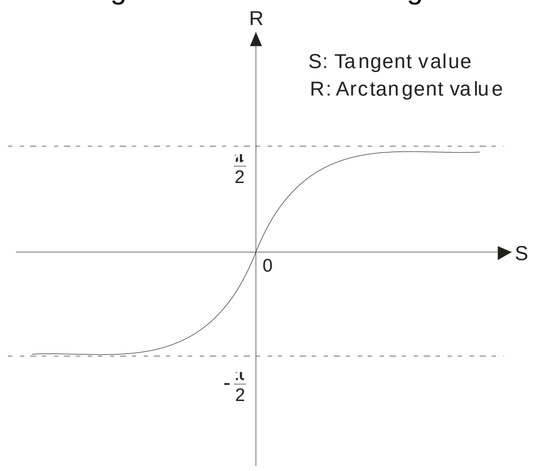
Ox	O100	
M1808	M1968	Zero flag

- Please refer to the additional remark below.

Explanation

- **S**: Source value (binary floating-point value); **D**: Arctangent value
- Arctangent value=$\tan^{-1}$

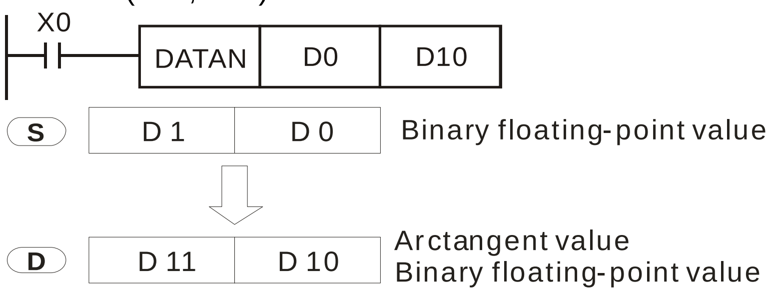
  The relation between tangent values and arctangent values is shown below.



- If a conversion result is 0, a zero flag will be ON.

Example

- When X0 is ON, the arctangent of the binary floating-point value in (D1, D0) is stored in (D11, D10).



Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API					Applicable model
136	D	SINH	P	S D	Hyperbolic sine of a binary floating-point value	20PM ✓

	Bit device				Word device											
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DSINH and DSINHP are valid.
  F represents a floating-point value. There is a decimal point in a floating-point value.

16-bit instruction
- - - -

32-bit instruction (6 steps)
DSINH Continuity instruction DSINHP Pulse instruction

- Flags

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

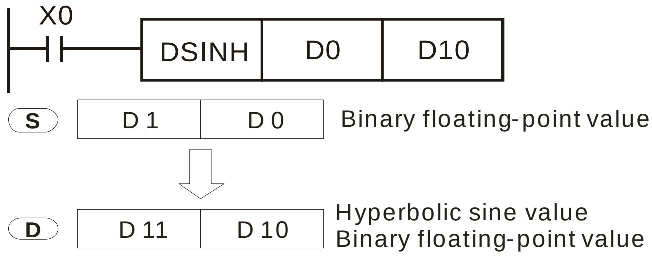
- Please refer to the additional remark below.

Explanation

- **S**: Source value (binary floating-point value); **D**: Hyperbolic sine value
- Hyperbolic sine value=(e^S-e^{-S})/2

Example

- When X0 is ON, the hyperbolic sine of the binary floating-point number in (D1, D0) is stored in (D11, D10).



Additional remark

- If the absolute value of a conversion result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of a conversion result is less than the minimum floating-point value available, a borrow flag will be ON.
- If a conversion result is 0, a zero flag will be ON.
- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						
137	D	COSH	P	(S) (D)	Hyperbolic cosine of a binary floating-point value	Applicable model: 20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Onlyt the 32-bit instructions DCOSH and DCOSHP are valid.
  F represents a floating-point value. There is a decimal point in a floating-point value.

16-bit instruction
- - - -

32-bit instruction (6 steps)
DCOSH Continuity instruction DCOSHP Pulse instruction

- Flags

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

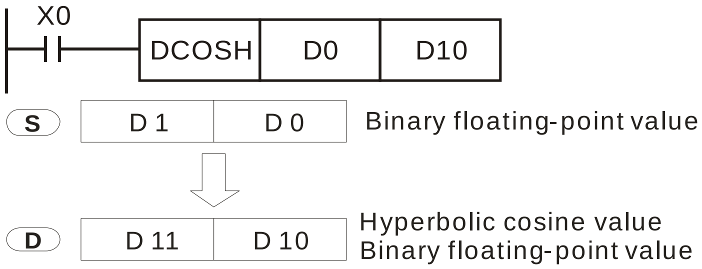
- Please refer to the additional remark below.

Explanation

- **S**: Source value (binary floating-point value); **D**: Hyperbolic cosine value
- Hyperbolic cosine value=$(e^{s}+e^{-s})/2$

Example

- When X0 is ON, the hyperbolic cosine of the binary floating-point number in (D1, D0) is stored in (D11, D10).



Additional remark

- If the absolute value of a conversion result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of a conversion result is less than the minimum floating-point value available, a borrow flag will be ON.
- If a conversion result is 0, a zero flag will be ON.
- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
138	D	TANH	P	S D	Hyperbolic tangent of a binary floating-point value	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*								*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DTANH and DTANHP are valid.
  F represents a floating-point value. There is a decimal point in a floating-point value.

16-bit instruction
- - - -

32-bit instruction (6 steps)
DTANH Continuity instruction DTANHP Pulse instruction

- Flags

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

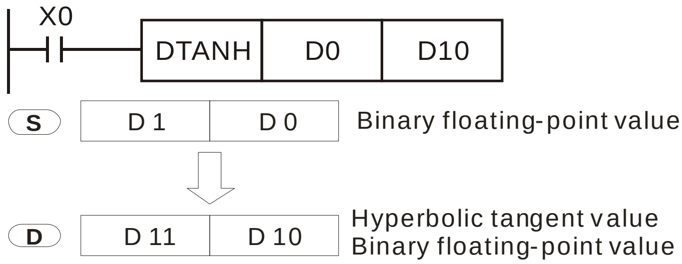
- Please refer to the additional remark below.

Explanation

- **S**: Source value (binary floating-point value); **D**: Hyperbolic tangent value
- Hyperbolic tangent value=$(e^{s}-e^{-s})/(e^{s}+e^{-s})$

Example

- When X0 is ON, the hyperbolic tangent of the binary floating-point number in (D1, D0) is stored in (D11, D10).



- If the absolute value of a conversion result is greater than the maximum floating-point value available, a carry flag will be ON.
- If the absolute value of a conversion result is less than the minimum floating-point value available, a borrow flag will be ON.
- If a conversion result is 0, a zero flag will be ON.

Additional remark

- Please refer to section 5.3 for more information about performing operations on floating-point values.

API						Applicable model
172	D	ADDR	P	S1 S2 D	Floating-point addition	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1													*		
S2													*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DADDR and DADDRP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (13 steps)

DADDR	Continuity instruction	DADDRP	Pulse instruction

- Flags

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

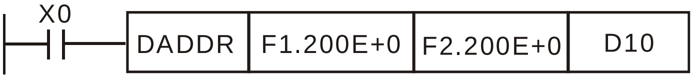
- Please refer to the additional remark below.

Explanation

- S_1: Augend; S_2: Addend; **D**: Sum
- S_1 and S_2 can be floating-point values.
- S_1 and S_2 can be floating-point values (e.g. F1.2), or data registers in which floating-point values are stored.
- If S_1 and S_2 are data registers in which floating-point values are stored, the function of API 172 DAADR is the same as the function of API 120 DEADD.
- The floating-point value in S_2 is added to the floating-point value in S_1, and the sum is stored in **D**.
- S_1 and S_2 can be the same register. If the instruction DAADR is used under the circumstances, the value in the register is added to itself whenever the conditional contact is ON in a scan cycle. Generally, the pulse instruction DADDRP is used.
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON. If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON. If an operation result is 0, a zero flag will be ON.

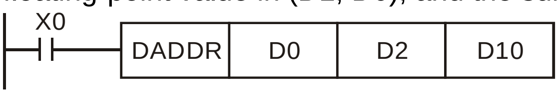
Example 1

- When X0 is ON, the floating-point value F2.200E+0 is added to the floating-point value F1.200E+0, and the sum F3.400E+0 is stored in (D11, D10). (The floating-point value F1.2 is represented by the scientific notation F1.200E+0 in a ladder diagram. The number of decimal places which are displayed can be set by means of the **View** menu in WPLSoft.)



Example 2

- When X0 is ON, the floating-point value in (D3, D2) is added to the floating-point value in (D1, D0), and the sum is stored in (D11, D10).



API						
173	D	SUBR	P	S1 S2 D	Floating-point subtraction	Applicable model: 20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1													*		
S2													*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DSUBR and DSUBRP are valid.

16-bit instruction
- - - -

32-bit instruction (13 steps)
DSUBR Continuity instruction DSUBRP Pulse instruction

- Flags

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

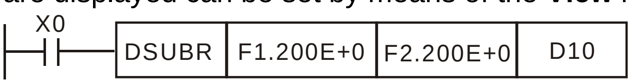
- Please refer to the additional remark below.

Explanation

- **S_1**: Minuend; **S_2**: Subtrahend; **D**: Subtrahend
- **S_1** and **S_2** can be floating-point values
- **S_1** and **S_2** can be floating-point values (e.g. F1.2), or data registers in which floating-point values are stored.
- If **S_1** and **S_2** are data registers in which floating-point values are stored, the function of API 172 DSUBR is the same as the function of API 121 DESUB.
- The floating-point value in **S_2** is subtracted from the floating-point value in **S_1**, and the difference is stored in **D**.
- **S_1** and **S_2** can be the same register. If the instruction DSUBR is used under the circumstances, the value in the register is subtracted from itself whenever the conditional contact is ON in a scan cycle. Generally, the pulse instruction DSUBRP is used.
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON. If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON. If an operation result is 0, a zero flag will be ON.

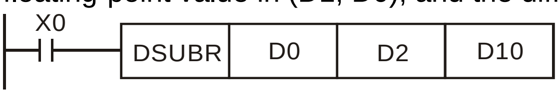
Example 1

- When X0 is ON, the floating-point value F2.200E+0 is subtracted from the floating-point value F1.200E+0, and the difference F-1.000E+0 is stored in (D11, D10). (The floating-point value F1.2 is represented by the scientific notation F1.200E+0 in a ladder diagram. The number of decimal places which are displayed can be set by means of the **View** menu in WPLSoft.)



Example 2

- When X0 is ON, the floating-point value in (D3, D2) is subtracted from the floating-point value in (D1, D0), and the difference is stored in (D11, D10).



API						Applicable model
174	D	MULR	P	S1 S2 D	Floating-point multiplication	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1													*		
S2													*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DSUBR and DSUBRP are valid.

16-bit instruction

\- \- \- \-

32-bit instruction (13 steps)

DMULR Continuity instruction DMULRP Pulse instruction

- Flags

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

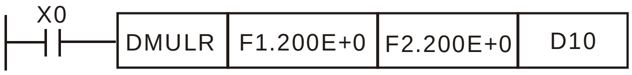
- Please refer to the additional remark below.

Explanation

- **S_1**: Multiplicand; **S_2**: Multiplier; **D**: Product
- **S_1** and **S_2** can be floating-point values.
- **S_1** and **S_2** can be floating-point values (e.g. F1.2), or data registers in which floating-point values are stored.
- If **S_1** and **S_2** are data registers in which floating-point values are stored, the function of API 172 DMULR is the same as the function of API 122 DEMUL.。
- The floating-point value in **S_1** is multiplied by the floating-point value in **S_2**, and the product is stored in **D**.
- **S_1** and **S_2** can be the same register. If the instruction DSUBR is used under the circumstances, the value in the register is multiplied by itself whenever the conditional contact is ON in a scan cycle. Generally, the pulse instruction DMULRP is used.
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON. If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON. If an operation result is 0, a zero flag will be ON.

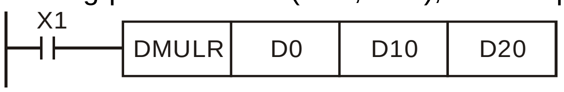
Example 1

- When X0 is ON, the floating-point value F1.200E+0 is multiplied by the floating-point value F2.200E+0, and the product F2.640E+0 is stored in (D11, D10). (The floating-point value F1.2 is represented by the scientific notation F1.200E+0 in a ladder diagram. The number of decimal places which are displayed can be set by means of the **View** menu in WPLSoft.)



Example 2

- When X1 is ON, the floating-point value in (D1, D0) is multiplied by the floating-point value in (D11, D10), and the product is stored in (D21, D20).



API						Applicable model
175	D	DIVR	P	S1 S2 D	Floating-point division	20PM ✓

	Bit device				Word device										
	X	Y	M	S	F	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S1													*		
S2													*		
D													*		

- Note: Please refer to specifications for more information about device ranges.
  Only the 32-bit instructions DSUBR and DSUBRP are valid.

16-bit instruction

-	-	-	-

32-bit instruction (13 steps)

DDIVR	Continuity instruction	DDIVP	Pulse instruction

- Flags

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

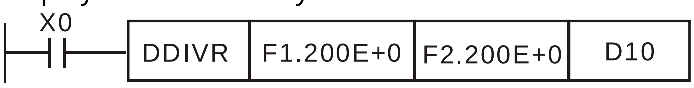
- Please refer to the additional remark below.

Explanation

- **S1**: Dividend; **S2**: Divisor; **D**: Quotient
- **S1** and **S2** can be floating-point values.
- **S1** and **S2** can be floating-point values (e.g. F1.2), or data registers in which floating-point values are stored.
- If **S1** and **S2** are data registers in which floating-point values are stored, the function of API 172 DDIVR is the same as the function of API 123 DEDIV.
- The floating-point value in **S1** is divided by the floating-point value in **S2**, and the product is stored in **D**.
- **S1** and **S2** can be the same register. If the instruction DSUBR is used under the circumstances, the value in the register is divided by itself whenever the conditional contact is ON in a scan cycle. Generally, the pulse instruction DDIVRP is used.
- If the absolute value of an oepration result is greater than the maximum floating-point value available, a carry flag will be ON. If the absolute value of an oepration reuslt is less than the minimum floating-point value available, a borrow flag will be ON. If an operation result is 0, a zero flag will be ON.

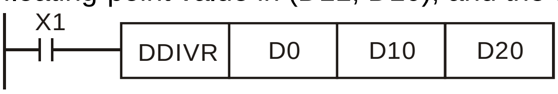
Example 1

- When X0 is ON, the floating-point value F1.200E is divided by the floating-point value F2.200E+0, and the quotient F0.545E+0 is stored in (D11, D10). (The floating-point value F1.2 is represented by the scientific notation F1.200E+0 in a ladder diagram. The number of decimal places which are displayed can be set by means of the **View** menu in WPLSoft.)



Example 2

- When X1 is ON, the floating-point value in (D1, D0) is divided by the floating-point value in (D11, D10), and the quotient is stored in (D21, D20).



API						Applicable model
215~217	D	LD#		S1 S2	Logical operation	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*

- Note: # represents &, |, or ^.
  Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)
LD# Continuity instruction - -

32-bit instruction (7 steps)
DLD# Continuity instruction - -

- Flag: None

Explanation

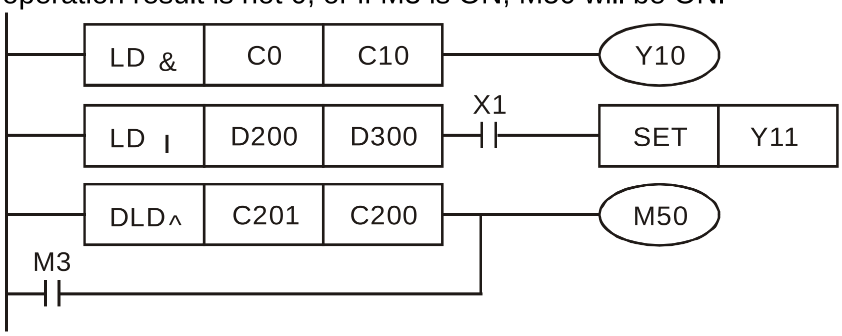
- S_1: Source device 1; S_2: Source device 2
- The instruction is used to compare the value in S_1 with that in S_2. If the comparison result is not 0, the condition of the instruction is met. If the comparison result is 0, the condition of the instruction is not met.
- The instruction LD# can be connected to a busbar directly.

API No.	16-bit instruction	32-bit instruction	ON	OFF
215	LD&	**D**LD&	S_1 & S_2 $\neq 0$	S_1 & S_2 $=0$
216	LD|	**D**LD|	S_1 | S_2 $\neq 0$	S_1 | S_2 $=0$
217	LD^	**D**LD^	S_1 ^ S_2 $\neq 0$	S_1 ^ S_2 $=0$

- &: Logical AND operation
- |: Logical OR operation
- ^: Logical exclusive OR operation
- If a 32-bit counter is used, the 32-bit insturciton DLD# must be used. If a 32-bit counter and the 16-bit instruction LD# are used, a program error will occur, and the ERROR LED indicator on the DVP-20PM series motion controller used will blink. (C200~C255 are 32-bit counters.)

Example

- A logical AND operator takes the values in C0 and C10, and performs the logical AND operation on each pair of corresponding bits. If the operation result is not 0, Y10 will be ON.
- A logical OR operator takes the values in D200 and D300, and performs the logical OR operation on each pair of corresponding bits. If the operation result is not 0 and X1 is ON, Y11 will be set to ON.
- A logical operator XOR takes the values in C201 and C200, and performs the logical exclusive OR operation on each pair of corresponding bits. If the operation result is not 0, or if M3 is ON, M50 will be ON.



API						Applicable model
218~220	D	AND#		S1 S2	Logical operation	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*

- Note: # represents &, |, or ^.
  Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)

AND #	Continuity instruction	-	-

32-bit instruction (7 steps)

DAND #	Continuity instruction	-	-

- Flag: None

Explanation

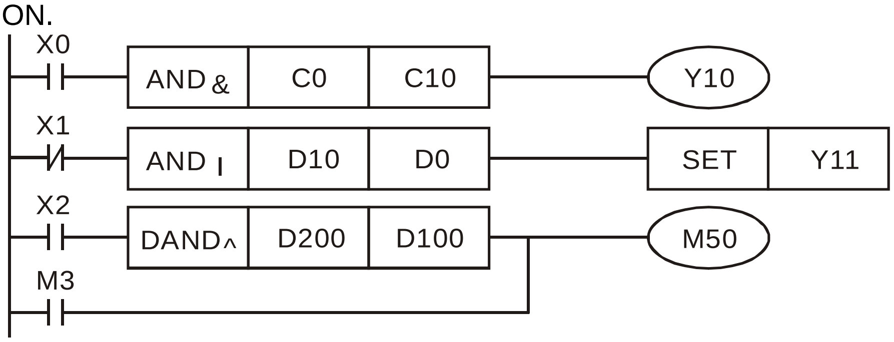
- **S_1**: Source device 1; **S_2**: Source device 2
- The instruction is used to compare the value in **S_1** with that in **S_2**. If the comparison result is not 0, the condition of the instruction is met. If the comparison result is 0, the condition of the instruction is not met.
- The instruction AND# is connected to a contact in series.

API No.	16-bit instruction	32-bit instruction	ON	OFF
218	AND&	**D**AND&	**S_1** & **S_2** ≠ 0	**S_1** & **S_2** = 0
219	AND|	**D**AND|	**S_1** | **S_2** ≠ 0	**S_1** | **S_2** = 0
220	AND^	**D**AND^	**S_1** ^ **S_2** ≠ 0	**S_1** ^ **S_2** = 0

- &: Logical AND operation
- |: Logical OR operation
- ^: Logical exclusive OR operation
- If a 32-bit counter is used, the 32-bit instruction DAND# must be used. If a 32-bit counter and the 16-bit instruction AND# are used, a program error will occur, and the ERROR LED indicator on the DVP-10PM series motion controller used will blink. (C200~C255 are 32-bit counters.)

Example

- When X0 is ON, a logical AND operator takes the values in C0 and C10, and performs the logical AND operation on each pair of corresponding bits. If the operation result is not 0, Y10 will be set to ON.
- When X1 is OFF, a logical OR operator takes the values in D10 and D0, and performs the logical OR operation on each pair of corresponding bits. If the operation result is not 0, Y1 will be set to ON.
- When X2 is ON, a logical XOR operator takes the values in (D201, D200) and (D101, D100), and performs the logical exclusive OR operation on each pair of corresponding bits. If the operation result is not 0, or if M3 is ON, M50 will be ON.



API					
221~223	D	OR#		S1 S2	Logical operation	Applicable model: 20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*

- Note: # represents &, |, or ^.
  Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)
OR# Continuity instruction - -

32-bit instruction (7 steps)
DOR# Continuity instruction - -

- Flag: None

Explanation

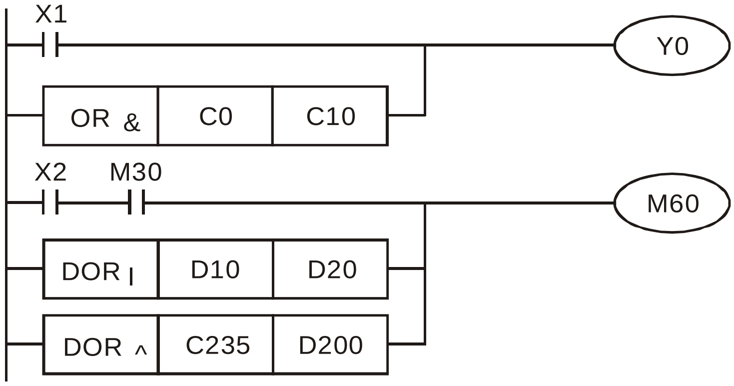
- **S_1**: Source device 1; **S_2**: Source device 2
- The instruction is used to compare the value in **S_1** with that in **S_2**. If the comparison result is not 0, the condition of the instruction is met. If the comparison result is 0, the condition of the instruction is not met.
- The instruction OR# is connected to a contact in parallel.

API No.	16-bit instruction	32-bit instruction	ON	OFF
221	OR&	**D**OR&	S_1 & S_2 ≠ 0	S_1 & S_2 = 0
222	OR|	**D**OR|	S_1 | S_2 ≠ 0	S_1 | S_2 = 0
223	OR^	**D**OR^	S_1 ^ S_2 ≠ 0	S_1 ^ S_2 = 0

- &: Logical AND operation
- |: Logical OR operation
- ^: Logical exclusive OR operation
- If a 32-bit counter is used, the 32-bit instruction DOR# must be used. If a 32-bit counter and the 16-bit instruction OR# are used, a program error will occur, and the ERROR LED indicator on the DVP-20PM series motion controller used will blink. (C200~C255 are 32-bit counters.)

Example

- When X1 is ON, Y0 is ON. Besides, when a logical AND operator performs the logical AND operation on each pair of corresponding bits in C0 and C10, and the operation result is not 0, Y0 is ON.
- When X2 and M30 are ON, M60 is ON. When a logical OR operator performs the logical OR operation on each pair of corresponding bits in the 32-bit register (D11, D10) and the 32-bit register (D21, D20), and the operation result is not 0, M60 is ON. Besides, when the logical XOR operator performs the logical exclusive OR operation on each pair of corresponding bits in the 32-bit counter C235 and the 32-bit register (D201, D200), and the operation result is not 0, M60 is ON.



API						Applicable model
224~230	D	LD※		S1 S2	Comparing values	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*

- Note: ※ represents =, >, <, <>, ≦ , or ≧ .
  Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)

LD※	Continuity instruction	-	-

32-bit instruction (7 steps)

DLD※	Continuity instruction	-	-

- Flag: None

Explanation

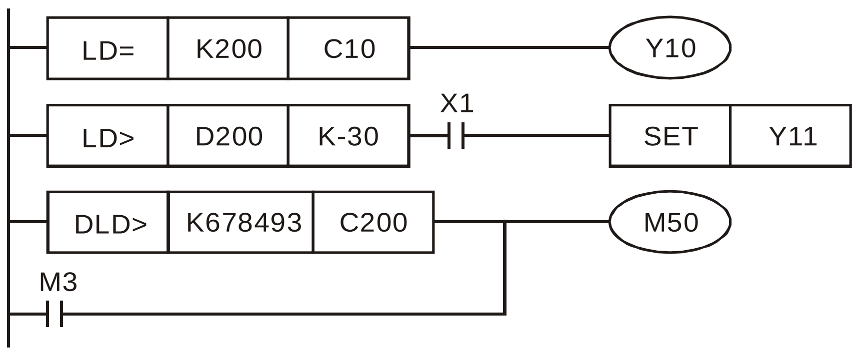
- **S_1**: Source device 1; **S_2**: Source device 2
- The instruction is used to compare the value in **S_1** with that in **S_2**. Take the instruction LD= for instance. If the comparison result is that the value in **S_1** is equal to that in **S_2**, the condition of the instruction is met. If the comparison result is that the value in **S_1** is not equal to that in **S_2**, the condition of the instruction is not met.
- The instruction LD※ can be connected to a busbar directly.

API No.	16-bit instruction	32-bit instruction	ON	OFF
224	LD =	**D**LD =	$S_1 = S_2$	$S_1 \neq S_2$
225	LD >	**D**LD >	$S_1 > S_2$	$S_1 \leqq S_2$
226	LD <	**D**LD <	$S_1 < S_2$	$S_1 \geqq S_2$
228	LD < >	**D**LD < >	$S_1 \neq S_2$	$S_1 = S_2$
229	LD < =	**D**LD < =	$S_1 \leqq S_2$	$S_1 > S_2$
230	LD > =	**D**LD > =	$S_1 \geqq S_2$	$S_1 < S_2$

- If a 32-bit counter is used, the 32-bit insturciton DLD※ must be used. If a 32-bit counter and the 16-bit instruction LD※ are used, ,a program error will occur, and the ERROR LED indicator on the DVP-20PM series motion controller used will blink. (C200~C255 are 32-bit counters.)

Example

- When the value in C10 is equal to K200, Y10 is ON.
- When the value in D200 is greater than K-30, and X1 is ON, Y11 is set to ON.
- When the value in C200 is less than K678,493, or when M3 is ON, M50 is ON.



API						
232 ~ 238	D	AND※		S1 S2	Comparing values	Applicable model: 20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*

- Note: ※ represents =, >, <, <>, ≦ , or ≧ .

  Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)

AND※ Continuity instruction - -

32-bit instruction (7 steps)

DAND※ Continuity instruction - -

- Flag: None

Explanation

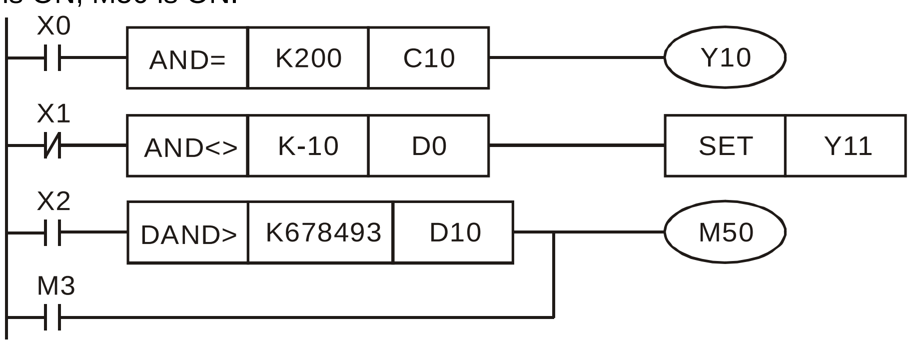
- $\mathbf{S_1}$: Source device 1; $\mathbf{S_2}$: Source device 2
- The instructions are used to compare the value in $\mathbf{S_1}$ with that in $\mathbf{S_2}$. Take the instruction AND= for instance. If the comparison result is that the value in $\mathbf{S_1}$ is equal to that in $\mathbf{S_2}$, the condition of the instruction is met. If the comparison result is that the value in $\mathbf{S_1}$ is not equal to that in $\mathbf{S_2}$, the condition of the instruction is not met.
- The instruction AND※ is connected to a contact in series.

API No.	16-bit instruction	32-bit instruction	ON	OFF
232	AND =	**D**AND =	$\mathbf{S_1 = S_2}$	$\mathbf{S_1 \neq S_2}$
233	AND >	**D**AND >	$\mathbf{S_1 > S_2}$	$\mathbf{S_1 \leqq S_2}$
234	AND <	**D**AND <	$\mathbf{S_1 < S_2}$	$\mathbf{S_1 \geqq S_2}$
236	AND < >	**D**AND < >	$\mathbf{S_1 \neq S_2}$	$\mathbf{S_1 = S_2}$
237	AND < =	**D**AND < =	$\mathbf{S_1 \leqq S_2}$	$\mathbf{S_1 > S_2}$
238	AND > =	**D**AND > =	$\mathbf{S_1 \geqq S_2}$	$\mathbf{S_1 < S_2}$

- If a 32-bit counter is used, the 32-bit insturciton DAND※ must be used. If a 32-bit counter and the 16-bit instruction AND※ are used, ,a program error will occur, and the ERROR LED indicator on the DVP-20PM series motion controller used will blink. (C200~C255 are 32-bit counters.)

Example

- When X0 is ON and the present value in C10 is equal to K200, Y10 is ON.
- When X1 is OFF and the value in D0 is not equal to K-10, Y11 is set to ON.
- When X2 is ON and the value in (D11, D10) is less than 678,493, or when M3 is ON, M50 is ON.



API					Applicable model
240~246	D	OR※	S1 S2	Comparing values	20PM ✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*

- Note: ※ represents =, >, <, <>, ≦ , or ≧ .

  Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)

OR※	Continuity instruction	-	-

32-bit instruction (7 steps)

DOR※	Continuity instruction	-	-

- Flag: None

Explanation

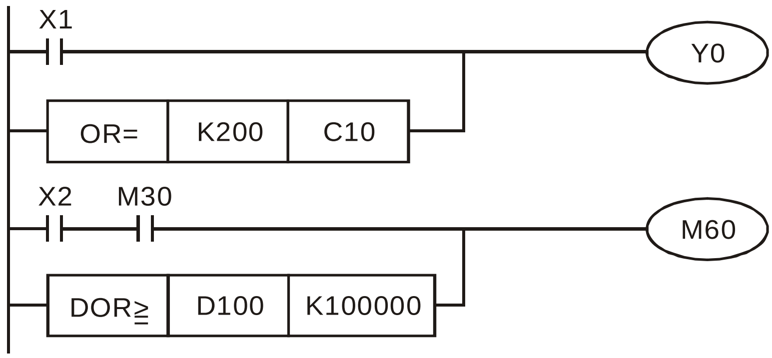
- **S_1**: Source device 1; **S_2**: Source device 2
- The instructions are used to compare the value in **S_1** with that in **S_2**. Take the instruction OR= for instance. If the comparison result is that the value in **S_1** is equal to that in **S_2**, the condition of the instruction is met. If the comparison result is that the value in **S_1** is not equal to that in **S_2**, the condition of the instruction is not met.
- The instruction OR※ is connected to a contact in parallel.

API No.	16-bit instruction	32-bit instruction	ON	OFF
240	OR =	**D**OR =	$S_1 = S_2$	$S_1 \neq S_2$
241	OR >	**D**OR >	$S_1 > S_2$	$S_1 \leqq S_2$
242	OR <	**D**OR <	$S_1 < S_2$	$S_1 \geqq S_2$
244	OR < >	**D**OR < >	$S_1 \neq S_2$	$S_1 = S_2$
245	OR < =	**D**OR < =	$S_1 \leqq S_2$	$S_1 > S_2$
246	OR > =	**D**OR > =	$S_1 \geqq S_2$	$S_1 < S_2$

- If a 32-bit counter is used, the 32-bit insturciton DOR※ must be used. If a 32-bit counter and the 16-bit instruction OR※ are used, ,a program error will occur, and the ERROR LED indicator on the DVP-20PM series motion controller used will blink. (C200~C255 are 32-bit counters.)

Example

- When X1 is ON, or when the present value in C10 is equal to K200, Y0 is ON.
- When X2 and M30 are ON, or when the value in (D101, D100) is greater than or equal to K100,000, M60 is ON.



API						
147	D	SWAP	P	(S)	Interchanging the high byte in a device with the low byte in the device	Applicable model: 20PM ✓

	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
	Bit device				Word device										
S							*	*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)

  Please refer to specifications for more information about device ranges.

  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (5 steps)

SWAP	Continuity instruction	SWAPP	Pulse instruction

32-bit instruction (7 steps)

DSWAP	Continuity instruction	DSWAPP	Pulse instruction

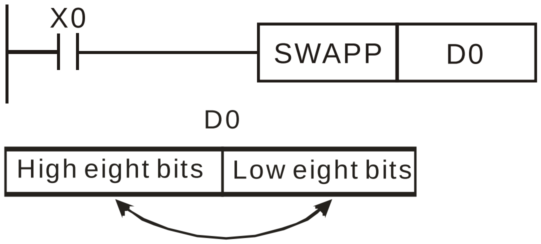
- Flag: None

Explanation

- **S**: Source device
- When the 16-bit instruction is executed, the high eight bits in **S** are interchanged with the low eight bits in **S**.
- When the 32-bit instruction is executed, the high eight bits in **S** are interchanged with the low eight bits in **S**, and the high eight bits in **S**+1 are interchanged with the low eight bits in **S**+1.
- Generally, the pulse instructions SWAPP and DSWAPP are used.

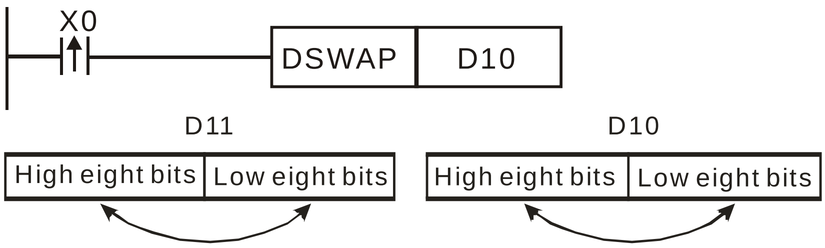
Example 1

- When X0 is ON, the high byte in D0 is interchanged with the low byte in D0.



Example 2

- When X0 is ON, the high eight bits in D11 are interchanged with the low eight bits in D11, and the high eight bits in D10 are interchanged with the low eight bits in D10.



API						Applicable model
154	D	RAND	P	S1 S2 D	Random value	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: The instruction supports V devices and Z devices. (If the 16-bit instruction is used, Z devices can not be used. If the 32-bit instruction is used, V devices can not be used.)

  Please refer to specifications for more information about device ranges.

  If KnX/KnY/KnM/KnS is used, it is suggested that X/devices/Y devices/M device numbers/S device numbers should start from a number which is a multiple of 16 in the octal numeral system or in the decimal numeral system, e.g. K1X0 (octal numeral system), K4SY20 (octal numeral system), K1M0 (decimal numeral system), and K4S16 (decimal numeral system).

16-bit instruction (5 steps)

RAND Continuity instruction RANDP Pulse instruction

32-bit instruction (7 steps)

DRAND Continuity instruction DRANDP Pulse instruction

- Flag: None

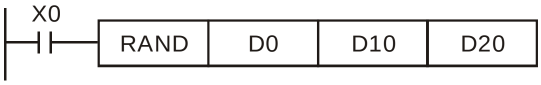
Explanation

- **S_1**: Minimum random value; **S_2**: Maximum random value; **D**: Result
- 16-bit instruction: The value in **S_1** and the value in **S_2** are in the range of K0 to K32,767.

  32-bit instruction: The value in **S_1** and the value in **S_2** are in the range of K0 to K2,147,483,647.
- The value in **S_1** must be less than the value in **S_2**. If the value in **S_1** is greater than the value in **S_2**, an operation error will occur.

Example

- When X0 is ON, the instruction RAND is used to generate a random value in the range of the value in D0 to the value in D10, and the random value is stored in D20.



API					Applicable model
202	SCAL	P	S1 S2 S3 D	Scale	20PM
					✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*							*		
S_2					*	*							*		
S_3					*	*							*		
D													*		

- Note: Please refer to specifications for more information about device ranges.

16-bit instruction (9 steps)

SCAL Continuity instruction SCALP Pulse instruction

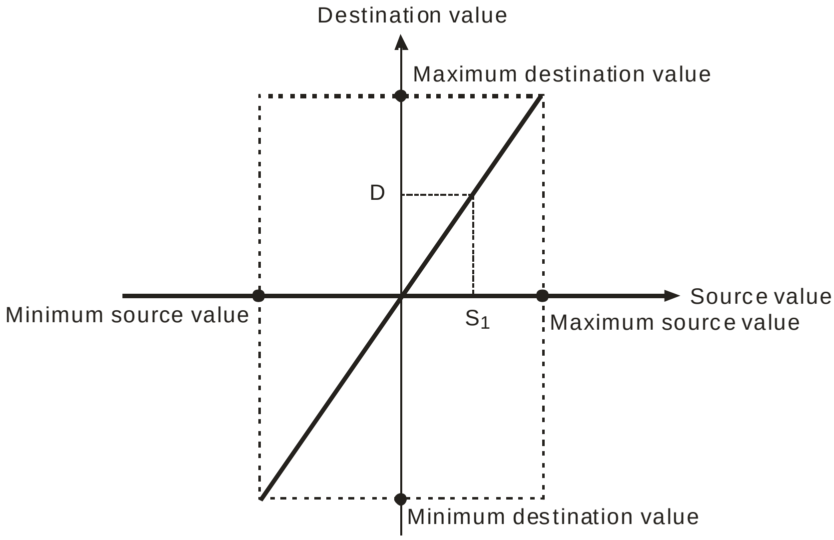
32-bit instruction (7 steps)

\- - - -

- Flag: None

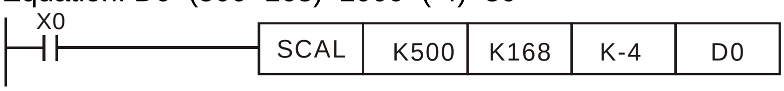
Explanation

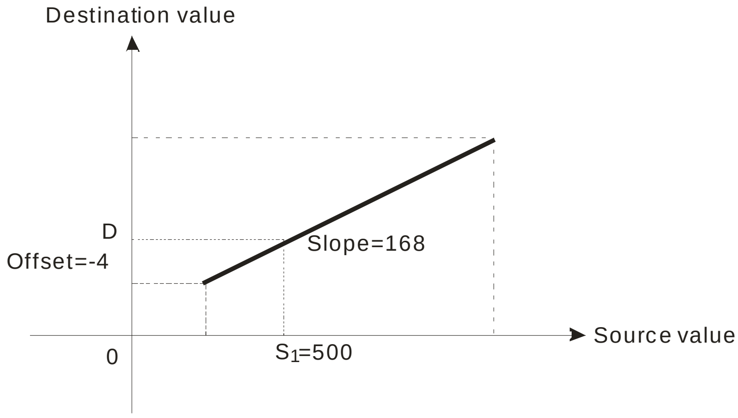
- **S_1**: Source device; **S_2**: Slope (Unit: 0.001); **S_3**: Offset; **D**: Destination device
- The values in **S_1**, **S_2**, and **S_3** must be in the range of -32767 to 32767.
- Equation: **D**=(**S_1**×**S_2**)÷1000+**S_3**
- To obtain the value in **S_2**, users have to use the slope equation below, round the result to the nearest integer, and get a 16-bit integer. To obtain the value in **S_3**, the users have to use the offset equation below, round the result to the nearest integer, and get a 16-bit integer.
- Slope equation: **S_2**=[(Maximum destination value–Minimum destination value)÷(Maximum source value–Minimum source value)]×1,000
- Offset equation: **S_3**= Minimum destination value–Minimum source value×**S_2**÷1,000
- Output curve



Example 1

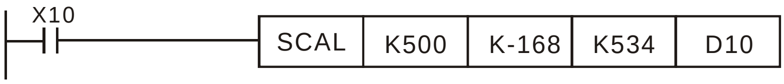
- Suppose the values in **S_1**, **S_2**, and **S_3** are 500, 168, and -4 respectively. When X0 is ON, the instruction SCAL is executed, and a scale is stored in D0.
- Equation: D0=(500×168)÷1000+(-4)=80

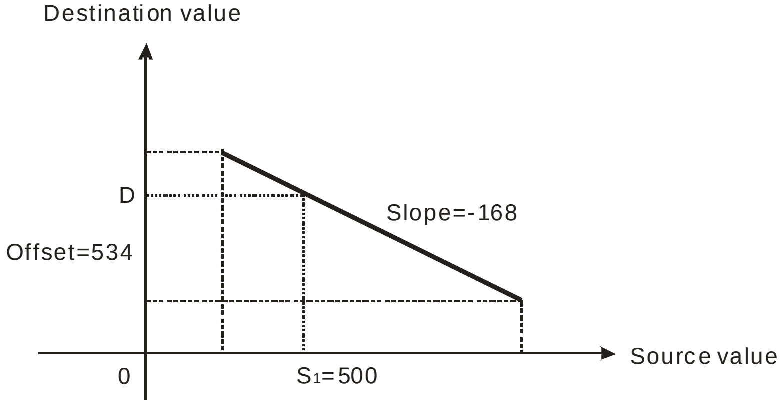




Example 2

- Suppose the values in **S_1**, **S_2**, and **S_3** are 500, -168, and 534 respectively. When X10 is ON, the instruction SCAL is executed, and a scale value is stored in D10.
- Equation: D10=(500×-168)÷1000+534=450





Additional remark

- Only when a slope and an offset are known can the instruction SCAL be used. If a slope and an offset are unknown, it is suggested that users should use the instruction SCLP.
- The value in **S_2** must be in the range of –32,768 to 32,767 (The actual value in **S_2** must be in the range of –32,768 to 32,767.) If the value in **S_2** is not in the range, please use the instruction SCLP instead.
- If users use the slop equation above, the maximum source value must be greater than the minimum source value, and the maximum destination value do not have to be greater than the minimum destination value.
- If the value in **D** is greater than 32,767, the value stored in **D** will be 32,767. If the value in **D** is less than -32,768, the value stored in **D** will be -32,768.

API						Applicable model
203	D	SCLP	P	S1 S2 D	Parameter scale	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S_1					*	*							*		
S_2					*	*							*		
S_3					*	*							*		
D													*		

- Note: Please refer to specifications for more information about device ranges.

16-bit instruction (7 steps)			
SCLP	Continuity instruction	SCLPP	Pulse instruction
32-bit instruction (13 steps)			
DSCLP	Continuity instruction	DSCLPP	Pulse instruction

- Flag: M1162

Explanation

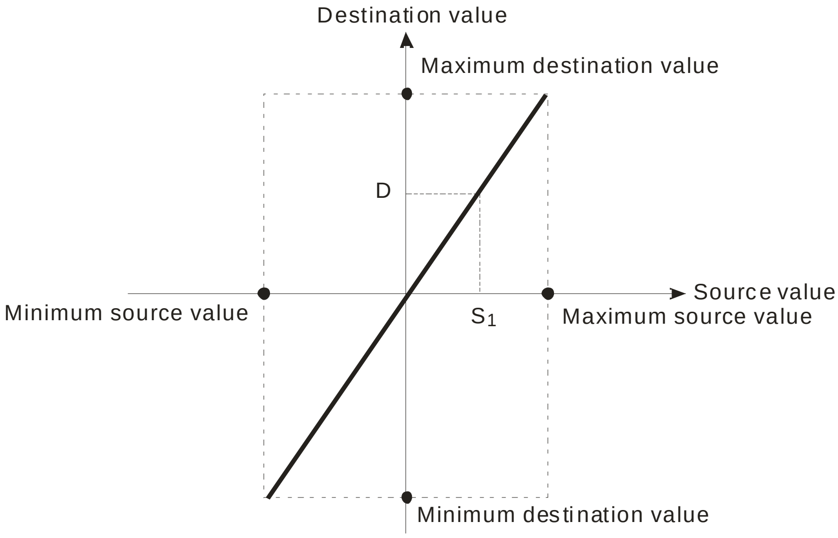
- **S_1**: Source device; **S_2**: Parameter (Unit: 0.001); **D**: Destination device
- 16-bit instruction: The setting of **S_2** is described below.

Device number	Parameter	Setting range
S_2	Maximum source value	-32768~32767
S_2+1	Minimum source value	-32768~32767
S_2+2	Maximum destination value	-32768~32767
S_2+3	Minimum destination value	-32768~32767

- If the 16-bit instruction is used, **S_2** will occupy four consecutive devices.
- 32-bit instruction: The setting of **S_2** is decribed below.

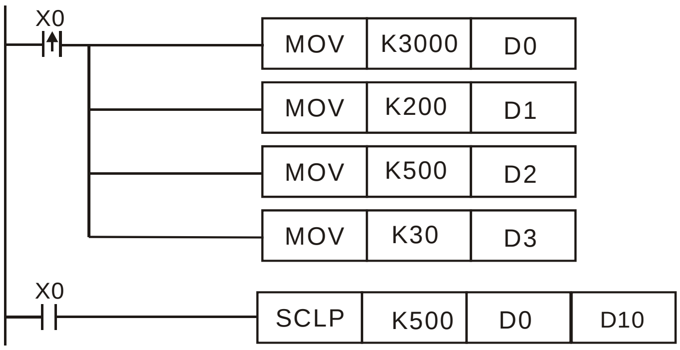
Device number	Parameter	Setting range	
		Integer	Floating-point value
S_2, **S_2**+1	Maximum source value	-2,147,483,648~ 2,147,483,647	32-bit floating-point values available
S_2+2, **S_2**+3	Minimum source value		
S_2+4, **S_2**+5	Maximum destination value		
S_2+6, **S_2**+7	Minimum destination value		

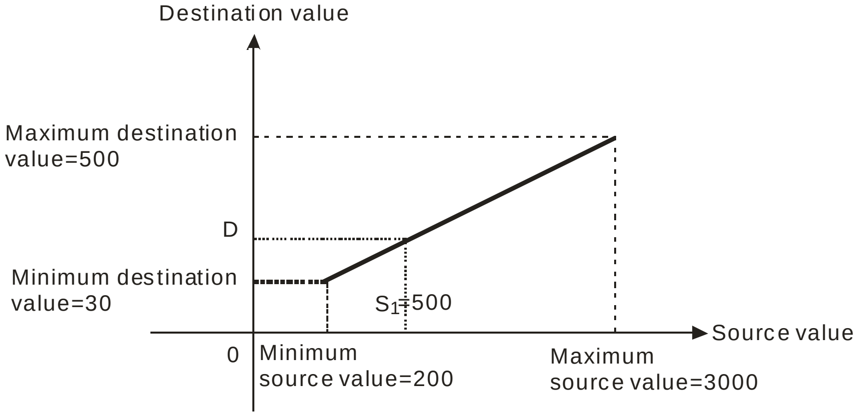
- If the 32-bit instruction is used, **S_2** will occupy eight consecutive devices.
- Equation: **D**=[(**S_1**–Minimum source value)×(Maximum destination value–Minimum destination value)]÷(Maximum source value–Minimum source value)+Minimum destination value
- Relation between the source value in **S_1** and the destination value in **D**:
  y=kx+b
  y=Destination value (**D**)
  k=Slope=(Maximum destination value–Minimum destination value)÷(Maximum source value–Minimum source value)
  x=Source value (**S_1**)
  b=Offset =Minimum destination value–Minimum source value×Slope
- After the parameters above are substituted for y, k, x, and b in the equation y=kx+b, the equation below will be obtained.
  y=kx+b=**D**=k**S_1**+b=Slope×**S_1** + Offset = Slope×**S_1** + Minimum destination value–Minimum source value×Slope=Slope×(**S_1**–Minimum source value) + Minimum destination value =(**S_1**–Minimum source value)×(Maximum destination value–Minimum destination value)÷(Maximum source value–Minimum source value) + Minimum destination value
- If the value in **S_1** is greater than the maximum source value, the value in **S_1** will be equal to the maximum source value. If the value in **S_1** is less than the minimum source value, the value in **S_1** will be equal to the minimum source value. After input values and parameters are set, an output curve will be gotten.



Example 1

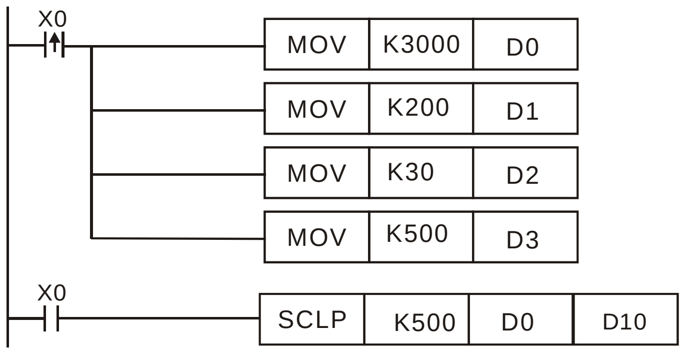
- Suppose the value in $\mathbf{S_1}$ is 500, the maximum source value in D0 is 3,000, the minimum source value in D1 is 200, the maximum destination value in D2 is 500, and the minimum destination value in D3 is 30. When X0 is ON, the instruction SCLP is executed, and a scale is stored in D10.
- Equation: D10=[(500–200)×(500–30)]÷(3,000–200)+30=80.35
  80.35 is rounded to the nearest integer, and becomes 80. 80 is stored in D10.

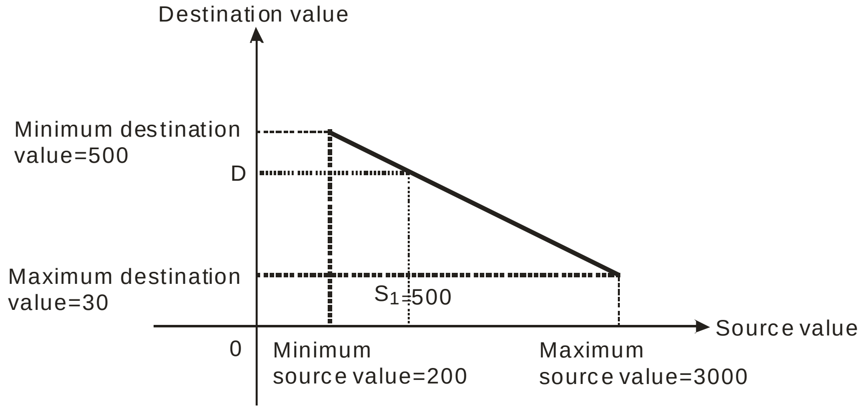




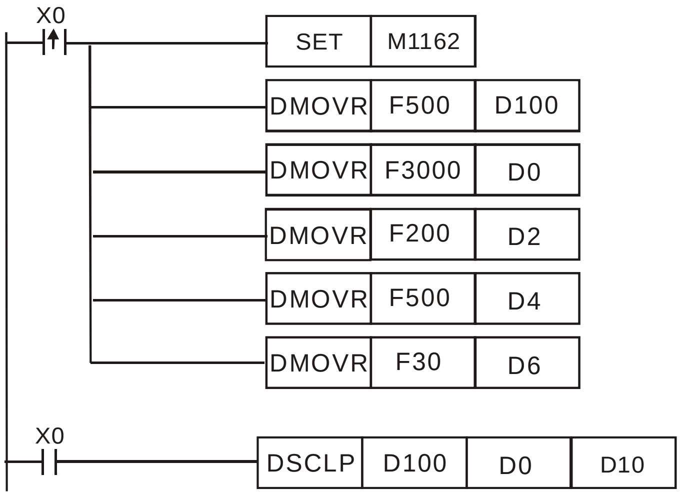
Example 2

- Suppose the value in $\mathbf{S_1}$ is 500, the maximum source value in D0 is 3,000, the minimum source value in D1 is 200, the maximum destination value in D2 is 30, and the minimum destination value in D3 is 500. When X0 is ON, the instruction SCLP is executed, and a scale is stored in D10.
- Eequation: D10=[(500–200)×(30–500)]÷(3,000–200)+500=449.64
- 449.64 is rounded to the nearest integer, and becomes 450. 450 is stored in D10.





- Suppose S_1 is D100, the value in D100 is F500, the maximum source value in D0 is F3000, the minimum source value in D2 is F200, the maximum destination value in D4 is F500, and the minimum destination value in D6 is F30. When X0 is ON, M1162 is set to ON, the instruction DSCLP is executed, and a scale is stored in D10.
- Equation: D10=[(F500–F200)×(F500–F30)]÷(F3000–F200)+F30=F80.35
  F80.35 is rounded to the nearest integer, and becomes F80. F80 is stored in D10.



Additional remark

- 16-bit instruction: The value in S_1 is in the range of the minimum source value and the maximum source value, i.e. the value in S_1 is in the range of -32,768 to 32,767. If the value in S_1 exceeds the minimum source value/the maximum source value, the minimum source value/the maximum source value will be used.
- 32-bit instruction: The integer in S_1 is in the range of the minimum source value and the maximum source value, i.e. the integer in S_1 is in the range of -2,147,483,648 to 2,147,483,647. If the integer in S_1 exceeds the minimum source value/the maximum source value, the minimum source value/the maximum source value will be used.
- 32-bit instruction: The floating-point value in S_1 is in the range of the minimum source value and the maximum source value, i.e. the floating-point value in S_1

is a 32-bit floating-point value available. If the floating-point value in **S_1** exceeds the minimum source value/the maximum source value, the minimum source value/the maximum source value will be used.

- If users use the instruction, the maximum source value must be greater than the minimum source value, and the maximum destination value does not have to be greater than the minimum destination value.

API		CJN		S	Negated conditional jump	Applicable model
256			P			20PM
						✓

	Bit device				Word device											16-bit instruction (3 steps)
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z	CJN: Continuity instruction; CJNP: Pulse instruction

- Note: The operand S can be a pointer.
  S is in the rage of P0~P255.
  A pointer can not be modified by a V device or a Z device.

32-bit instruction: - - - -

- Flag: None

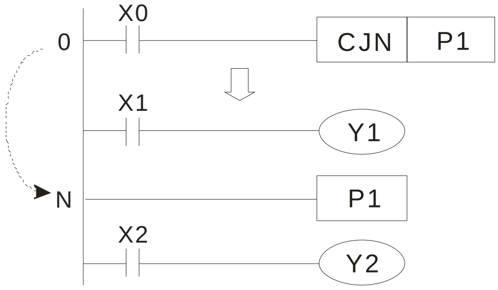
Explanation

- **S**: Pointer
- If the conditional contact connected to CJN is ON, the next address will be executed. If the conditional contact connected to CJN is not ON, the address to which **S** points will be executed.
- If some part of the main program O100 does not need to be executed, users can use CJN or CJNP to shorten the scan time. Besides, if a dual output is used, users can use CJ or CJP.
- If the program specified by a pointer is prior to the instruction CJN, a watchdog timer error will occur, and the main program O100 will not be executed. Please use the instruction carefully.
- The instruction CJN can specify the same pointer repeatedly. The pointer specified by CJN can not be the same as the pointer specified by CALL, otherwise an error will occur.
- When the instruction CJN/CJNP in a program is executed, the actions of the devices in the program are as follows.
  1. The states of the Y devices, the states of the M devices, and the states of the S devices in the program remain the same as those before the execution of the jump.
  2. The 10 millisecond timers in the program stop counting.
  3. The general counters in the program stop counting, and the general applied instructions in the program are not executed.
  4. If the instructions which are used to reset the timers in the program are driven before the jump is executed, the timers will still be reset during the execution of the jump.

Example

- When X0 is OFF, the execution of the program jumps from address 0 to address N (P1), and the addresses between address 0 and address N are skipped.
- When X0is ON, the execution of the program starts from address 0, and the instruction CJN is not executed.

(Negated conditional jump)



API						Applicable model
257		JMP		S	Unconditional jump	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z

- Note: The operand S can be a pointer.
  S is in the rage of P0~P255.
  The instruction does not need to be driven by a contact.
  A pointer can not be modified by a V device or a Z device.

16-bit instruction (3 steps)

JMP	Continuity instruction	-	-

32-bit instruction

-	-	-	-

- Flag: None

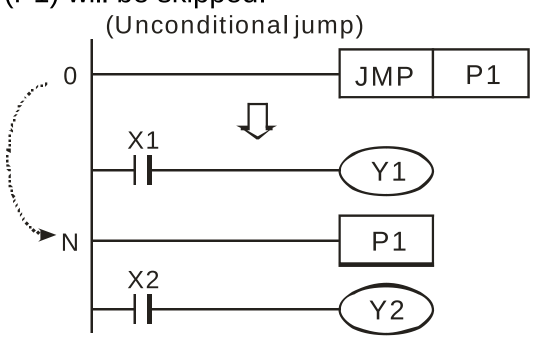
Explanation

◆ The function of JMP is similar to the function of CJ. CJ must be driven by a contact whereas JMP does not have to be driven by a contact.

◆ The pulse instruction JMPP is not supported.

Example

◆ After address 0 is scanned, address N will be executed whether there is a conditional contact before the instruction JMP (and whether the conditional contact is ON or OFF), and the addresses between address 0 and address N (P1) will be skipped.



API		Mnemonic			Function	Applicable model
258		BRET			Returning to a busbar	20PM
						✓

	Bit device				Word device											
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z

- Note: There is no operand.
  The instruction does not need to be driven by a contact.

16-bit instruction (1 step)

BRET	Continuity instruction	-	-

32-bit instruction

-	-	-	-

- Flag: None

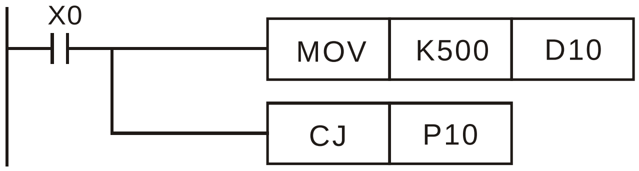
Explanation

◆ The instruction BRET does not have to be driven by a contact.

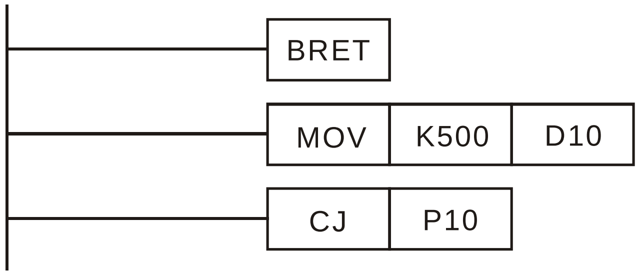
◆ After the instruction BRET is executed, the instructions which should be driven by a conditional contact will seem to be connected to a busbar, and will be executed.

Example

◆ In the general program shown below, the instructions are executed only when X0 is ON.



◆ After the instruction BRET is added, the instructions which should be driven by a contact will seem to be connected to a busbar, and will be executed.



API		MMOV			Converting a 16-bit value into a 32-bit value	Applicable model
259			P	S D		20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)

MMOV	Continuity instruction	MMOVP	Pulse instruction

32-bit instruction

-	-	-	-

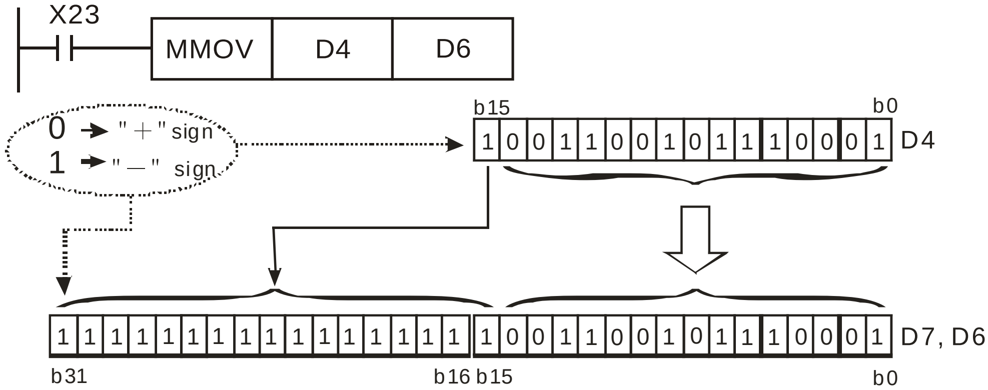
- Flag: None

Explanation

- **S**: Source device (16-bit device); **D**: Destination device (32-bit device)
- The value in the 16-bit device **S** is transferred to the 32-bit device **D**. The sign bit in **S** is duplicated, and stored in **D**.

Example

- When X23 is ON, the value in D4 is transferred to D6 and D7.



Bit 15 is D4 is transferred to bit 15~ bit 31 in (D7, D6). The value in (D7, D6) becomes a negative value. (The value in D4 is also a negative value.)

API						Applicable model
260		RMOV	P	(S) (D)	Converting a 32-bit value into a 16-bit value	20PM
						✓

	Bit device				Word device										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	V	Z
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

- Note: Please refer to specifications for more information about device ranges.

16-bit instruction (5 steps)

RMOV Continuity instruction RMOVP Pulse instruction

32-bit instruction

\- - - -

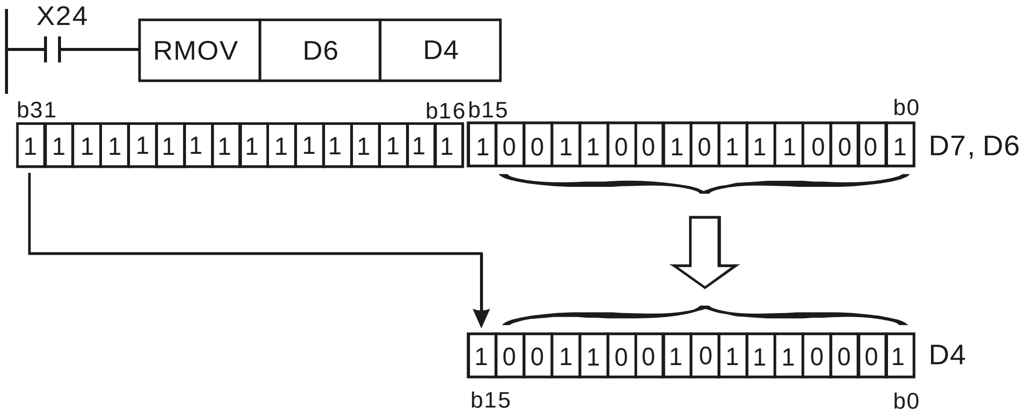
- Flag: None

Explanation

- **S**: Source device (32-bit device); **D**: Destination device (16-bit device)
- Data in the 32-bit device **S** is transferred to the 16-bit device **D**. The sign bit in **S** is retained.

Example

- When X24 is ON, data in D6 and D7 is transferred to D4.



When X24 is ON, bit 31 in D7 is transferred to bit 15 in D4, bit 0~bit 14 in D6 are transferred, and bit 15~bit 30 in D6 and D7 are not transferred.

5.7 Motion Control Function Block Table

Type	Name	Description	Page No.
Uniaxial motion control function blocks	Absolute single-speed motion	Starting absolute single-speed motion	5-152
	Relative single-speed motion	Starting relative single-speed motion	5-156
	Absolute two-speed motion	Starting absolute two-speed motion	5-160
	Relative two-speed motion	Starting relative two-speed motion	5-163
	Inserting single-speed motion	Inserting single-speed motion	5-166
	Inserting two-speed motion	Inserting two-speed motion	5-170
	JOG motion	Starting JOG motion	5-174
	Manual pulse generator mode	Enabling a manual pulse generator mode	5-177
	Electronic gear motion	Starting electronic gear motion	5-179
	Returning home	Starting motion of returning home	5-181
	Setting input sources for manual pulse generators/electronic gears	Setting input sources for manual pulse generators/electronic gears	5-183
	Stopping uniaxial motion	Stopping the motion of the axis specified	5-184
	Parameter setting 1	Setting motion parameters	5-186
	Parameter setting 2	Setting motion parameters	5-188
	Reading the present position/speed of an axis	Reading the present position/speed of an axis	5-190
	State of an axis	Reading and clearing the present erroneous state of an axis	5-192
	Setting the present position of an axis	Setting the present position of an axis	5-194
	Setting the polarities of input terminals	Setting the polarities of input terminals	5-195
	Uniaxial cyclic electronic cam motion	Uniaxial cyclic electronic cam motion	5-197
	Uniaxial noncyclic electronic cam motion	Uniaxial noncyclic electronic cam motion	5-202
	Multiaxial cyclic electronic cam motion	Multiaxial cyclic electronic cam motion	5-206
	Reading a cam point	Reading a particular point in a cam chart	5-210
	Writing a cam point	Modifying a particular point in a cam chart	5-212
	Calculating a synchronization ratio	Calculating a synchronization ratio	5-214
	Creating a cam curve	Creating a cam curve	5-217
	Updating a cam curve	Updating a cam curve	5-224
	Rotary cut	Creating a rotary cut curve, and executing electronic cam motion	5-226
	Flying shear	Creating a flying shear curve, and executing electronic cam motion	5-232
	Creating a curve for a wire winding machine	Creating a curve for a wire winding machine	5-239
Multiaxial motion control function blocks	Setting the parameters of G-code motion	Setting the parameters of G-code motion	5-243
	Executing G-code motion	Setting and executing an Ox motion subroutine	5-246
	Stopping G-code motion	Stopping the execution of an Ox motion subroutine	5-248
	Reading an M-code	Reading an M-code	5-250
Other motion control function blocks	High-speed counter	Starting a high-speed counter	5-252
	Setting high-speed comparison	Starting high-speed comparison	5-255
	Resetting high-speed comparison	Resetting high-speed comparison	5-257
	Setting high-speed capture	Starting high-speed capture	5-260
	High-speed masking	Starting high-speed masking	5-264
	Setting an interrupt	Setting the trigger for an interrupt subroutine	5-266

5.8 Introduction of the Pins in a Motion Control Function Block

5.8.1 Definitions of Input Pins/Output Pins

Common input pins and output pins in motion control function blocks are listed below. The pins listed below do not appear in a single motion control function block. For example, a motion control function block only has one input pin, that is, it has either the Execute input pin or the Enable input pin.

Input pin			
Name	**Description**	**Format**	**Setting value**
Execute	Starting the motion control function block	BOOL	True/False
Enable	Starting the motion control function block	BOOL	True/False

Output pin			
Name	**Description**	**Format**	**Setting value**
Done	The execution of the function block is complete.	BOOL	There is a transition in the Done output pin's signal from low to high when the execution of motion control function block is complete.
Valid	An output value is valid.	BOOL	There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	There is a transition in the Aborted output pin's signal from low to high when the execution of the motion control function block is interrupted by a command.
Error	An error occurs in a function block.	BOOL	There is a transition in the Error output pin's signal from low to high when an error occurs in the motion control function block.

A motion control function block has either the Execute input pin or the Enable input pin. The Execute input pin/The Enable input pin in a motion control function block is used to start the motion control function block. A motion control function block generally has the Busy output pin and the Done output pin. The Busy output pin and the Done output pin in a function block indicate the state of the motion control function block. If the execution of motion control function block is to be interrupted by another motion control function block, the Aborted output pin will be added to the motion control function block. Besides, the Error output pin in a motion control function block is used to indicate that an error occurs in the motion control function block when the motion control function block is executed.

A motion control function block has not only the Execute input pin/the Enable input pin, but also value/state input pins. The characteristics of the value/state input pins are described below.

- Use of input values
  - If the input pin that a motion control function block has is the Execute input pin, values are used when there is a transition in the Execute input pin's signal from low to high. If a new value is created, it becomes valid when the Execute input pin is triggered again.
  - If the input pin that a motion control function block has is the Enable input pin, values are used when there is a transition in the Enable input pin's signal from low to high. Compared with the Execute input pin, the Enable input pin is used more often when a value used is updated repeatedly.
- An input value exceeds a range.
  After a motion control function block is started, the input values which are not in ranges allowed will be limited, or result in an error occurring in the motion control function block. If an error occurring in a motion control function block results in an error occurring in an axis, the motion control function block is applied incorrectly. Users should prevent incorrect values from being generated in an applied program.
- Output pins are mutually exclusive.
  - If the input pin that a motion control function block has is the Execute input pin, only the Busy

output pin, the Done output pin, the Aborted output pin, or the Error output pin can be set to True. If the Execute input pin is set to True, the Busy output pin, the Done output pin, the Aborted output pin, or the Error output pin must be set to True.

- If the input pin that a motion control function block has is the Enable input pin, the Valid output pin and the Error output pin are mutually exclusive, and only the Valid output pin or the Error output pin can be set to True.

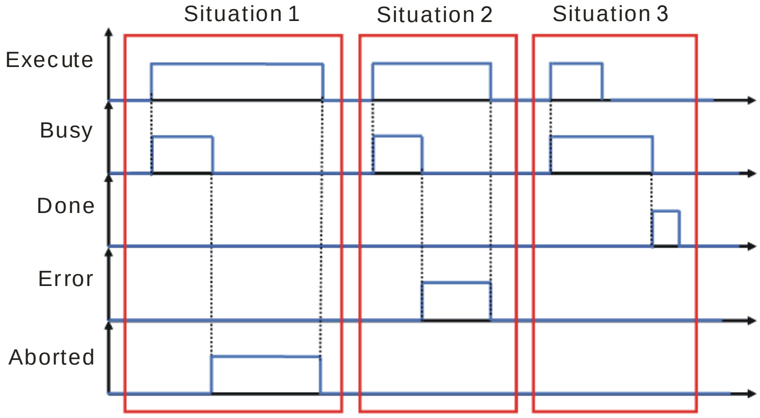
- Time when output data/states are valid
  - If the input pin that a motion control function block has is the Execute input pin, the Done output pin, the Error output pin, the Aborted output pin, and data output are reset when there is a transition in the Execute input pin's signal from high to low, but the execution of the function block does not stop when there is a transition in the Execute input pin's signal from high to low. Even if the Execute input pin in a motion control function block is reset before the execution of the motion control function block is complete, output states will still be generated and retained for one cycle. If a motion control function block is started again before the execution of the motion control function block is complete, the motion control function block will not give feedback to the Done output pin and the Aborted output pin, and an error will occur.
  - If the input pin that a motion control function block has is the Enable input pin, the Valid output pin, the Busy output pin, and the Error output pin are reset when there is a transition in the Enable input pin's signal from high to low.
- Characteristic of the Done output pin

  The Done output pin in a motion control function block will be set to True after the motion control function block is executed successfully.
- Characteristic of the Busy output pin
  - If the input pin that a motion control function block has is the Execute input pin, the motion control function block uses the Busy output pin to indicate that the execution of the motion control function block is not complete, and new output states (values) are expected to be generated. The Busy output pin is set to True when there is a transition in the Execute input pin's signal from low to high. When the Done output pin, the Aborted output pin, and the Error output pin are set to True, the Busy output pin are reset.
  - If the input pin that a motion control function block has is the Enable input pin, the motion control function block uses the Busy output pin to indicate that the execution of the motion control function block is not complete, and new output states (values) are expected to be generated. The Busy output pin in a motion control function block is set to True when there is a transition in the Enable input pin's signal from low to high, and is set to True when the motion control function block is executed. When the Busy output pin is set to True, output states (values) still change.
- Characteristic of the Aborted output pin

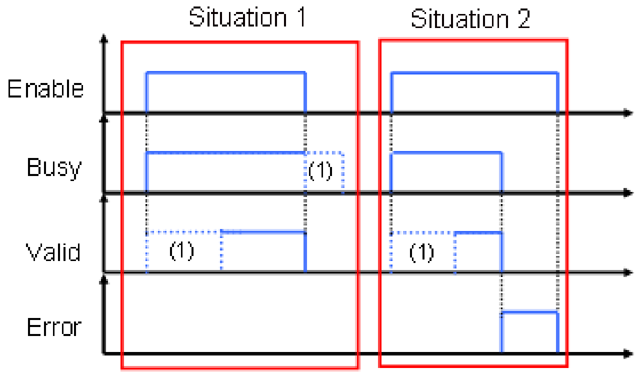
  The Aborted output pin in a motion control function block is set to True when the execution of the motion control function block is interrupted by a command.
- Relation between the Enable input pin and the Valid output pin

  If the input pin that a motion control function block has is the Enable input pin, the motion control function block uses the Busy output pin to indicate whether output data/states are valid. The Valid output pin is set to True only when the Enable input pin is set to true or output data/state are valid. If an error occurs in a motion control function block, output data/states will not be valid, and the Valid output pin will be set to False. The Valid output pin in a motion control function block will not be reset until the error occurring in the motion control function block is eliminated, and output data/states become valid.

5.8.2 Timing Diagram for Input/Output Pins



Situation 1: The execution of the motion control function block is interrupted.
Situation 2: An error occurs in the motion control function block.
Situation 3: The execution of the motion control function block is complete normally.



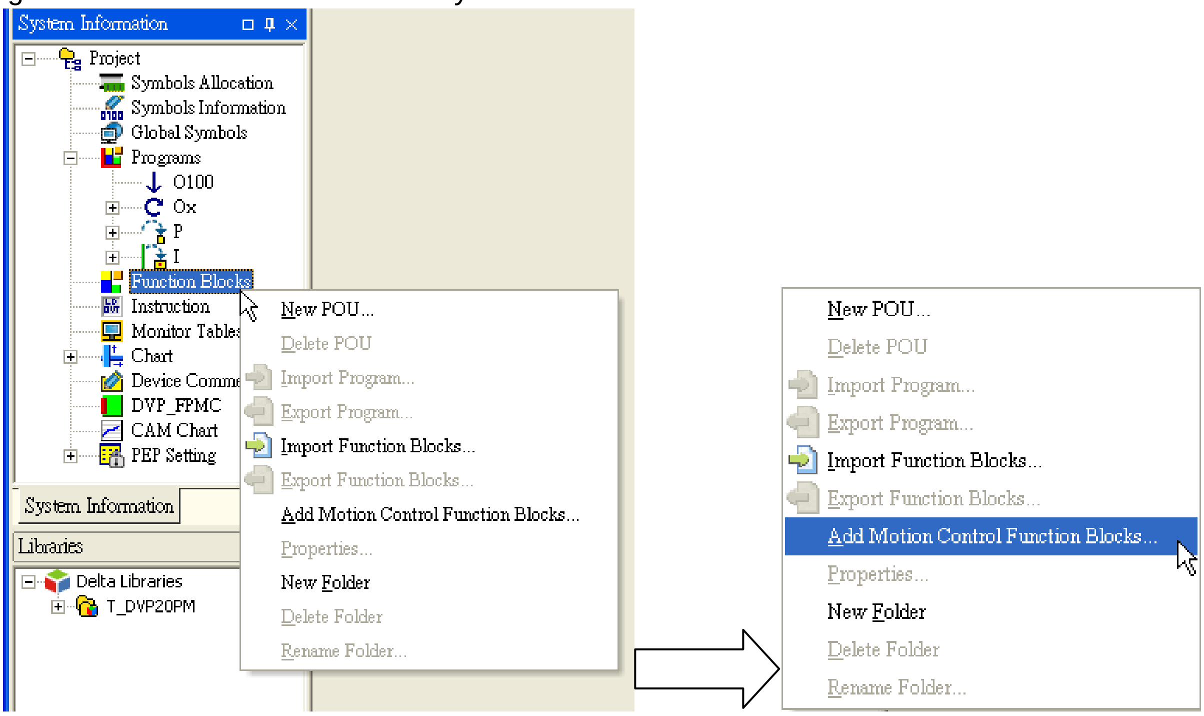
(1) It may take some time.

Situation 1: The motion control function block is executed normally.
Situation 2: An error occurs in the motion control function block.

5.8.3 Introducing the Use of PMSoft

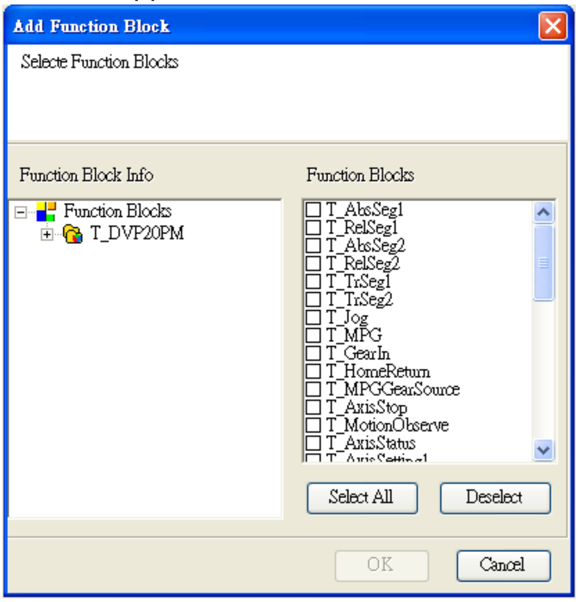
The use of the motion control function blocks in PMSoft is introduced below.

(1) Right-click **Function Blocks** in the system information area in PMSoft.



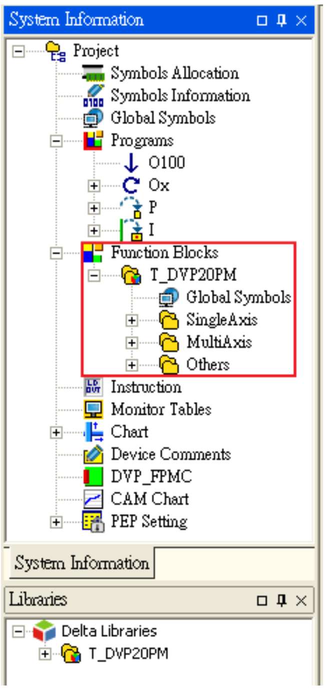
Click **Add Motion Control Function Blocks...** on the context menu.

(2) The **Add Function Block** window appears.



Users can select motion control function blocks in the **Add Function Block** window. If the users click **Select All**, all the motion control function blocks in the **Add Function Block** window will be selected. After users select motion control function blocks, they have to click **OK**.

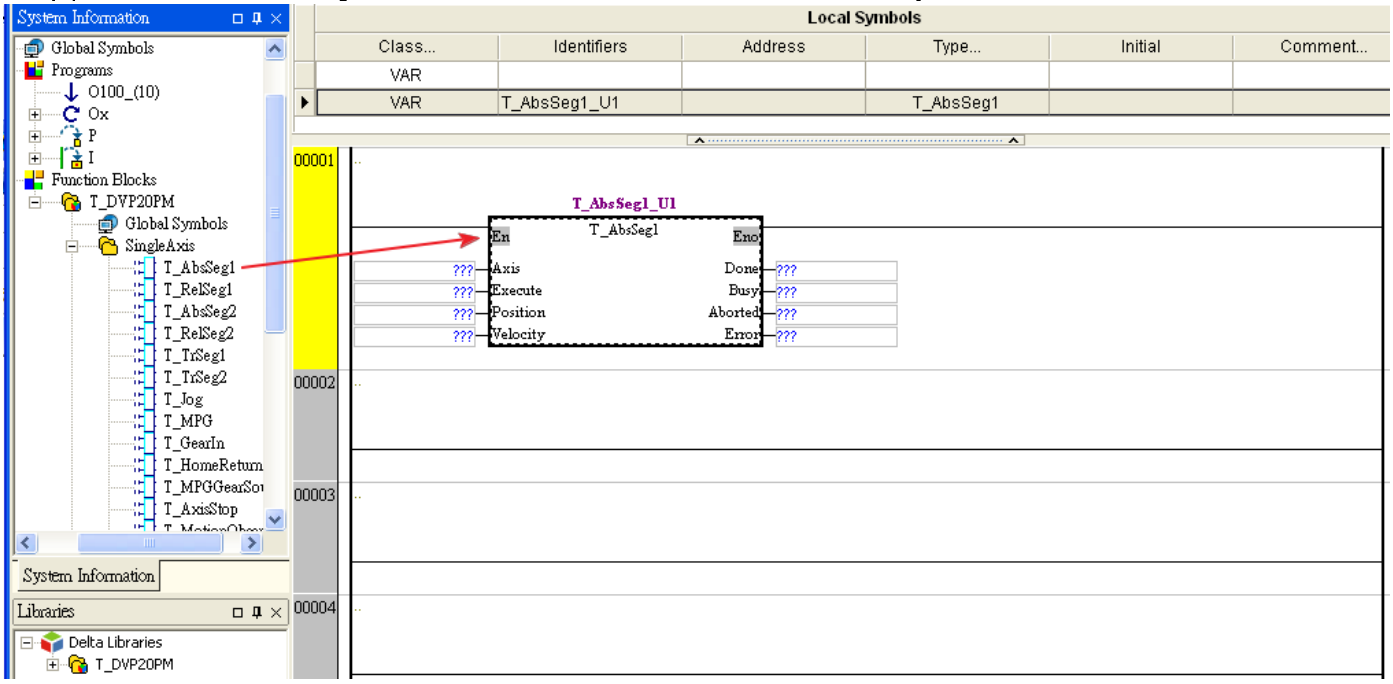
(3) After the users click **OK**, the motion control function blocks selected in the **Add Function Block** window will be automatically added to **Function Blocks** in the system information area.



- The folders added to **Function Blocks** are shown below.

- Definitions of the folders
  - SingleAxis: Uniaxial motion (Uniaxial point-to-point motion, electronic gear synchronization, and electronic cam synchronization)
  - MultiAxis: Multi-axis motion (G-code execution)
  - Others: Other functions (measuring time, high-speed comparison, high-speed capture, and setting interrupts)

(4) After the users drag motion control function blocks in folder, they can use them.



5.9 Delta-defined Parameter Table

Delta-defined parameters are for input pins in Delta motion control function blocks. Users can directly use Delta-defined parameters to operate motion control function blocks without having to know the descriptions of the input pins in the motion control function blocks. Delta-defined parameters are described below.

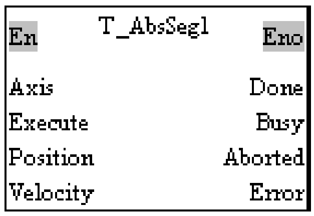
Name	Type	Value	Motion control function block	Description
TRUE	BOOL	True	All motion control function blocks	Input pin
FALSE	BOOL	False		Input pin
mcRising	BOOL	True	T_TrSeg2, T_TrSeg1, T_HomeReturn	Transition in DOG's signal from low to high
mcFalling	BOOL	False		Transition in DOG's signal from high to low
mcPositive	BOOL	True	T_HomeReturn	Returning home in the positive direction
mcNegative	BOOL	False		Returning home in the negative direction
mcSCurve	BOOL	True	T_AxisSetting2	Speed curve: S curve
mcTrapezoid	BOOL	False		Speed curve: Trapezoid curve
mcNC	BOOL	True	T_InputPolatiry	Normally-closed contact
mcNO	BOOL	False		Normally-open contact
mcUp_Up	BOOL	True	T_HTmr	A high-speed timer becomes active when its signal goes from low to high.
mcUp_Down	BOOL	False		A high-speed timer becomes active when its signal goes from high to low.
mcCmpSet	BOOL	True	T_Compare	An output is set when the condition of a comparison is met.
mcCmpRst	BOOL	False		An output is reset when the condition of a comparison is met.
mcMotor	WORD	0	T_AxisSetting2	Motor unit
mcMachine	WORD	1		Mechanical unit,
mcComp	WORD	2		Compound unit
mcUD	WORD	0	T_AxisSetting2, T_HCnt	Counting up/down
mcPD	WORD	1		Pulses+Directions
mcAB	WORD	2		A/B-phase pulses
mc4AB	WORD	3		Four times the frequency of A/B-phase pulses
IntTimer	WORD	0	T_Interrupt	An interrupt signal is triggered by a time interval.
IntX00	WORD	1		The source of an interrupt signal is X0.
IntX01	WORD	2		The source of an interrupt signal is X1.
IntX02	WORD	3	T_Interrupt	The source of an interrupt signal is X2.

Name	Type	Value	Motion control function block	Description
IntX03	WORD	4	T_Interrupt	The source of an interrupt signal is X3.
IntX04	WORD	5		The source of an interrupt signal is X4.
IntX05	WORD	6		The source of an interrupt signal is X5.
IntX06	WORD	7		The source of an interrupt signal is X6.
IntX07	WORD	8		The source of an interrupt signal is X7.
IntStart0	WORD	1		The source of an interrupt signal is Start0.
IntStop0	WORD	2		The source of an interrupt signal is Stop0.
IntStart1	WORD	3		The source of an interrupt signal is Start1.
IntStop1	WORD	4		The source of an interrupt signal is Stop1.
mcCmpAxis1	WORD	0	T_Compare	The source of a comparison is the present position of the first axis.
mcCmpAxis2	WORD	1		The source of a comparison is the present position of the second axis.
mcCmpAxis3	WORD	2		The source of a comparison is the present position of the third axis.
mcCmpAxis4	WORD	3		The source of a comparison is the present position of the fourth axis.
mcCmpC200	WORD	4		The source of a comparison is the value of C200.
mcCmpC204	WORD	5		The source of a comparison is the value of C204.
mcCmpC208	WORD	6		The source of a comparison is the value of C208.
mcCmpC212	WORD	7		The source of a comparison is the value of C212.
mcCmpCLR0	WORD	0	T_Compare	The device used for a comparison is CLR0.
mcCmpCLR1	WORD	1		The device used for a comparison is CLR1.
mcCmpY0	WORD	0		The device used for a comparison is Y0.
mcCmpY1	WORD	1		The device used for a comparison is Y1.
mcCmpY2	WORD	2		The device used for a comparison is Y2.
mcCmpY3	WORD	3		The device used for a comparison is Y3.
mcCmpRstC200	WORD	4		The device used for a comparison is C200.
mcCmpRstC204	WORD	5		The device used for a comparison is C204.
mcCmpRstC208	WORD	6		The device used for a comparison is C208.
mcCmpRstC212	WORD	7		The device used for a comparison is C212.
mcCapAxis1	WORD	0	T_Capture	The source of capture is the present position of the first axis.
mcCapAxis2	WORD	1		The source of capture is the present position of the second axis.
mcCapAxis3	WORD	2		The source of capture is the present position of the third axis.
mcCapAxis4	WORD	3		The source of capture is the present position of the fourth axis.
mcCapC200	WORD	4		The source of capture is the value of C200.
mcCapC204	WORD	5		The source of capture is the value of C204.
mcCapC208	WORD	6		The source of capture is the value of C208.
mcCapC212	WORD	7		The source of capture is the value of C212.
mcCapPG0	WORD	0	T_Capture	The source of a capture signal is PG0.
mcCapMPGB0	WORD	1		The source of a capture signal is MPGB0.
mcCapMPGA0	WORD	2		The source of a capture signal is MPGA0.
McCapLSN0	WORD	3		The source of a capture signal is LSN0.
McCapLSP0	WORD	4		The source of a capture signal is LSP0.
McCapDOG0	WORD	5		The source of a capture signal is DOG0.
mcCapStop0	WORD	6		The source of a capture signal is Stop0.
mcCapStart0	WORD	7		The source of a capture signal is Start0.
mcCapPG1	WORD	8		The source of a capture signal is PG1.
mcCapMPGB1	WORD	9		The source of a capture signal is MPGB1.
mcCapMPGA1	WORD	10		The source of a capture signal is MPGA1.
mcCapLSN1	WORD	11	T_Capture	The source of a capture signal is LSN1.

Name	Type	Value	Motion control function block	Description
mcCapLSP1	WORD	12	T_Capture	The source of a capture signal is LSP1.
McCapDOG1	WORD	13		The source of a capture signal is DOG1.
mcCapStop1	WORD	14		The source of a capture signal is Stop1.
mcCapStart1	WORD	15		The source of a capture signal is Start1.
mcX0	WORD	0		The source of a capture signal is X0.
mcX1	WORD	1		The source of a capture signal is X1.
mcX2	WORD	2		The source of a capture signal is X2.
mcX3	WORD	3		The source of a capture signal is X3.
mcX4	WORD	4		The source of a capture signal is X4.
mcX5	WORD	5		The source of a capture signal is X5.
mcX6	WORD	6		The source of a capture signal is X6.
mcX7	WORD	7		The source of a capture signal is X7.
mcX10	WORD	8		The source of a capture signal is X10.
mcX11	WORD	9		The source of a capture signal is X11.
mcX12	WORD	10		The source of a capture signal is X12.
mcX13	WORD	11		The source of a capture signal is X13.

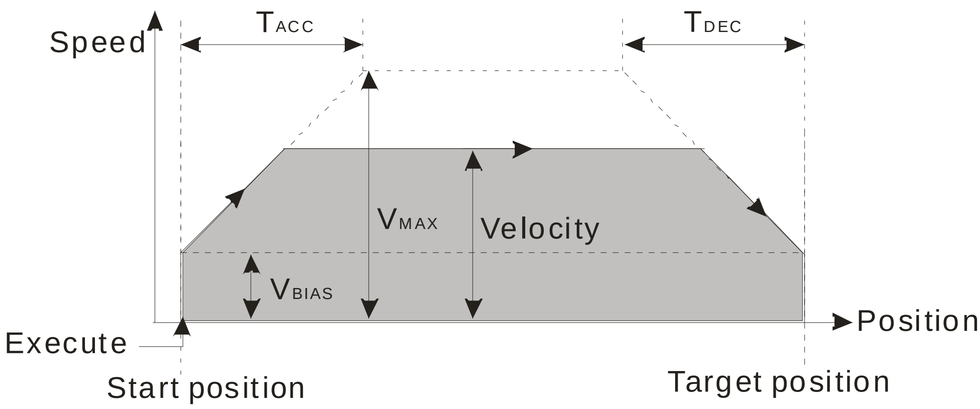
5.10 Uniaxial Motion Control Function Blocks

5.10.1 Absolute Single-speed Motion



1. Motion control function block

   The motion control function block T_AbsSeg1 is used to start absolute single-speed motion. After absolute single-speed motion is started, the speed of the absolute single-speed motion will increase from the V_{BIAS} set to the velocity set. The speed of the absolute single-speed motion will not decrease from the velocity set to the V_{BIAS} set until the present command position of the axis specified is near the target position set. Users can set the Vbias input pin, the Vmax input pin, the Tacc input pin, and the Tdec input pin in the motion control function block T_AxisSetting1. The number of pulses is a unit for the Position input pin, and the number of pulses per second is a unit for the Velocity input pin. The users can change the unit used by means of the motion control function block T_AxisSetting2.



2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
Position	Absolute position	DWORD	K-2,147,483,648~ K2,147,483,647	The value of the Position input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Velocity	Target speed	DWORD	K1~K2,147,483,647	When the motion control function block is executed, the value of the Velocity input pin is updated repeatedly.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal when motion is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. • If the Execute input pin is set to False when motion is complete, the Done output pin will be set to False in the next cycle.
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. ● If the Execute input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The axis specified is in motion before the motion control function block is executed.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

The number of pulses is a unit for the Position input pin, and the number of pulses per second is a unit for the Velocity input pin. Users can change the unit used by means of the motion control function block T_AxisSetting2.

3. Troubleshooting

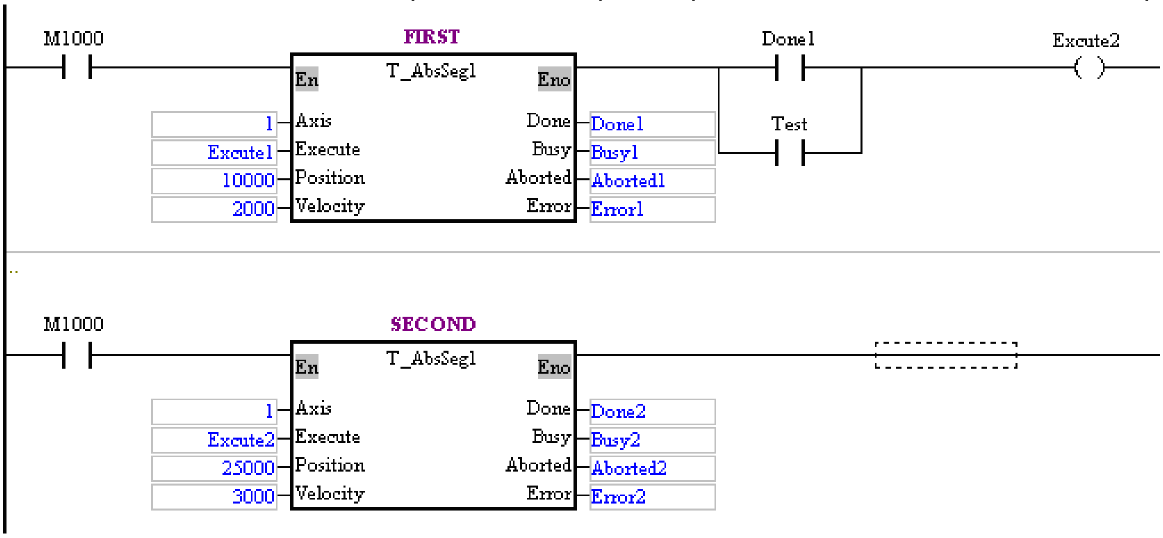
Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The motion control function block conflicts with other motion control function blocks.	Make sure that other uniaxial motion control function blocks are not started or the execution of other uniaxial motion control function blocks is complete before the motion control function block is started.

4. Example

Purposes:

- After the first single-speed motion is complete, the second single-speed motion will be executed.
- The second single-speed motion is executed before the execution of the first single-speed motion is complete.

The motion control function block named FIRST is set so that the first axis moves at a speed of 2,000 pulses per second, and moves for 10,000 pulses. The motion control function block named SECOND is set so that the first axis moves at a speed of 3,000 pulses per second, and moves for 15,000 pulses.

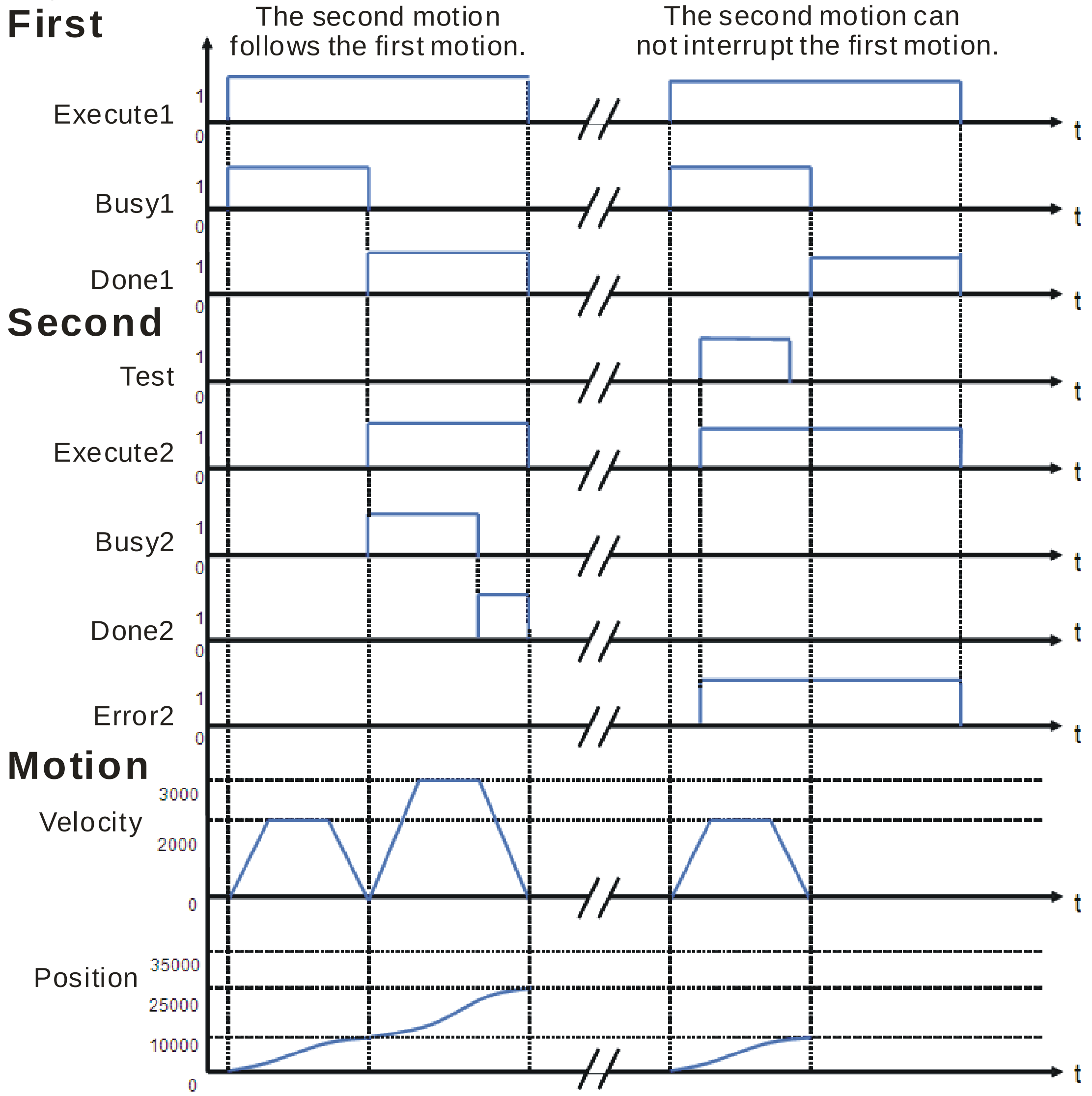


- After the first single-speed motion is complete, the second single-speed motion will be executed.

Steps:
(a) Set Execute1 to True.
(b) Wait for a transition in Done2's signal from low to high or a transition in Error2's signal from low to high.

- The second single-speed motion is executed before the execution of the first single-speed motion is complete.
  Steps:
  (a) Set Execute1 to True.
  (b) Set Test to ON when Busy1 is set to True.
  (c) Wait for a transition in Done2's signal from low to high or a transition in Error2's signal from low to high.

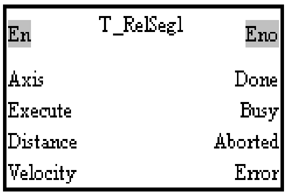
Timing diagram:



- After the first single-speed motion is complete, the second single-speed motion will be executed. After the execution of the motion control function block named FIRST is complete, the motion control function block named SECOND will be executed. The first axis moves for 25,000 pulses.
- The second single-speed motion is executed before the execution of the first single-speed motion is complete.
  When Error2 is set to True, the first axis moves for 10,000 pulses. The motion control function block named SECOND is invalid.

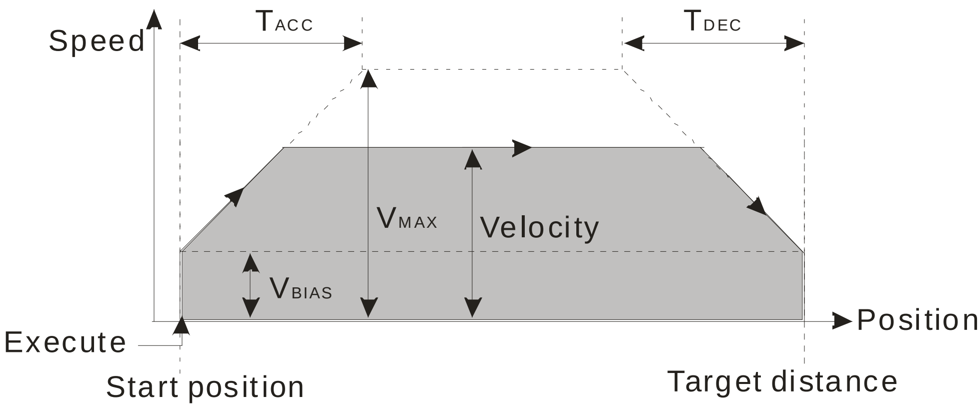
5. Modules which are supported
   The motion control function block T_AbsSeg1 supports DVP20PM00M and DVP20PM00D.

5.10.2 Relative Single-speed Motion



1. Motion control function block

   The motion control function block T_RelSeg1 is used to start relative single-speed motion. After relative single-speed motion is started, the speed of the relative single-speed motion will increase from the V_{BIAS} set to the velocity set. The speed of the relative single-speed motion will not decrease from the velocity set to the V_{BIAS} set until the distance for which the relative single-speed motion moves is the distance set. Users can set the Vbias input pin, the Vmax input pin, the Tacc input pin, and the Tdec input pin in the motion control function block T_AxisSetting1. The number of pulses is a unit for the Distance input pin, and the number of pulses per second is a unit for the Velocity input pin. Users can change the unit used by means of the motion control function block T_AxisSetting2.



2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
Distance	Relative distance	DWORD	K-2,147,483,648~ K2,147,483,647	The value of the Distance input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Velocity	Target speed	DWORD	K1~K2,147,483,647	When the motion control function block is executed, the value of the Velocity input pin is updated repeatedly.

Output pin				
Name	Function	Data type	Time when there is a transition in an output pin's signal from low to high	Time when there is a transition in an output pin's signal from high to low
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal when motion is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. • If the Execute input pin is set to False when motion is complete, the Done output pin will be set to False in the next cycle.
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. • If the Execute input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The axis specified is in motion before the motion control function block is executed.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

The number of pulses is a unit for the Distance input pin, and the number of pulses per second is a unit for the Velocity input pin. Users can change the unit used by means of the motion control function block T_AxisSetting2.

3. Troubleshooting

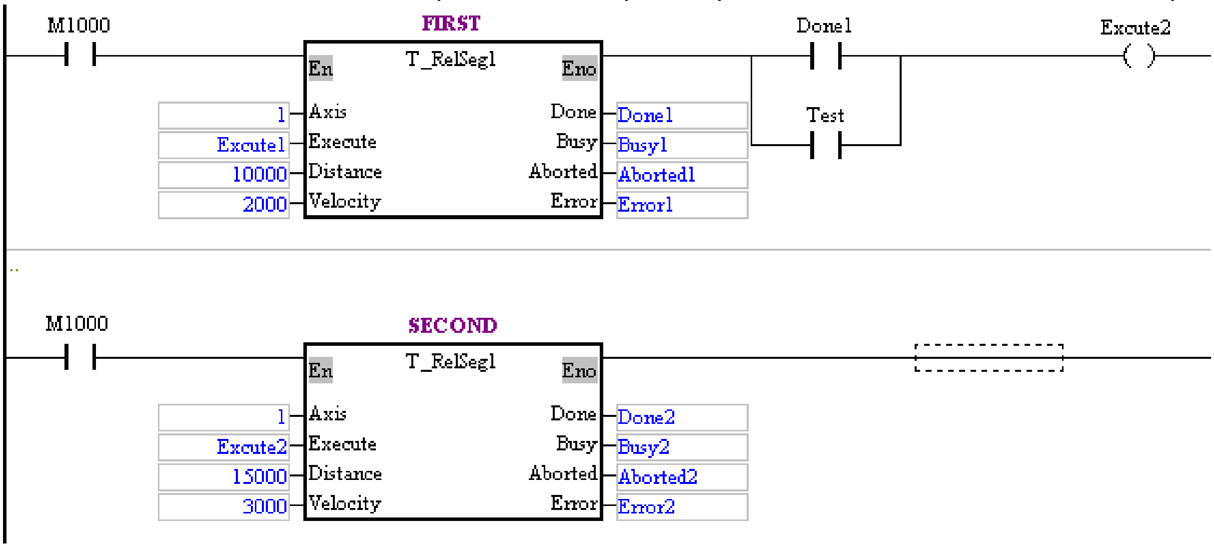
Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The motion control function block conflicts with other motion control function blocks.	Make sure that other uniaxial motion control function blocks are not started or the execution of other uniaxial motion control function blocks is complete before the motion control function block is started.

4. Example

Purposes:

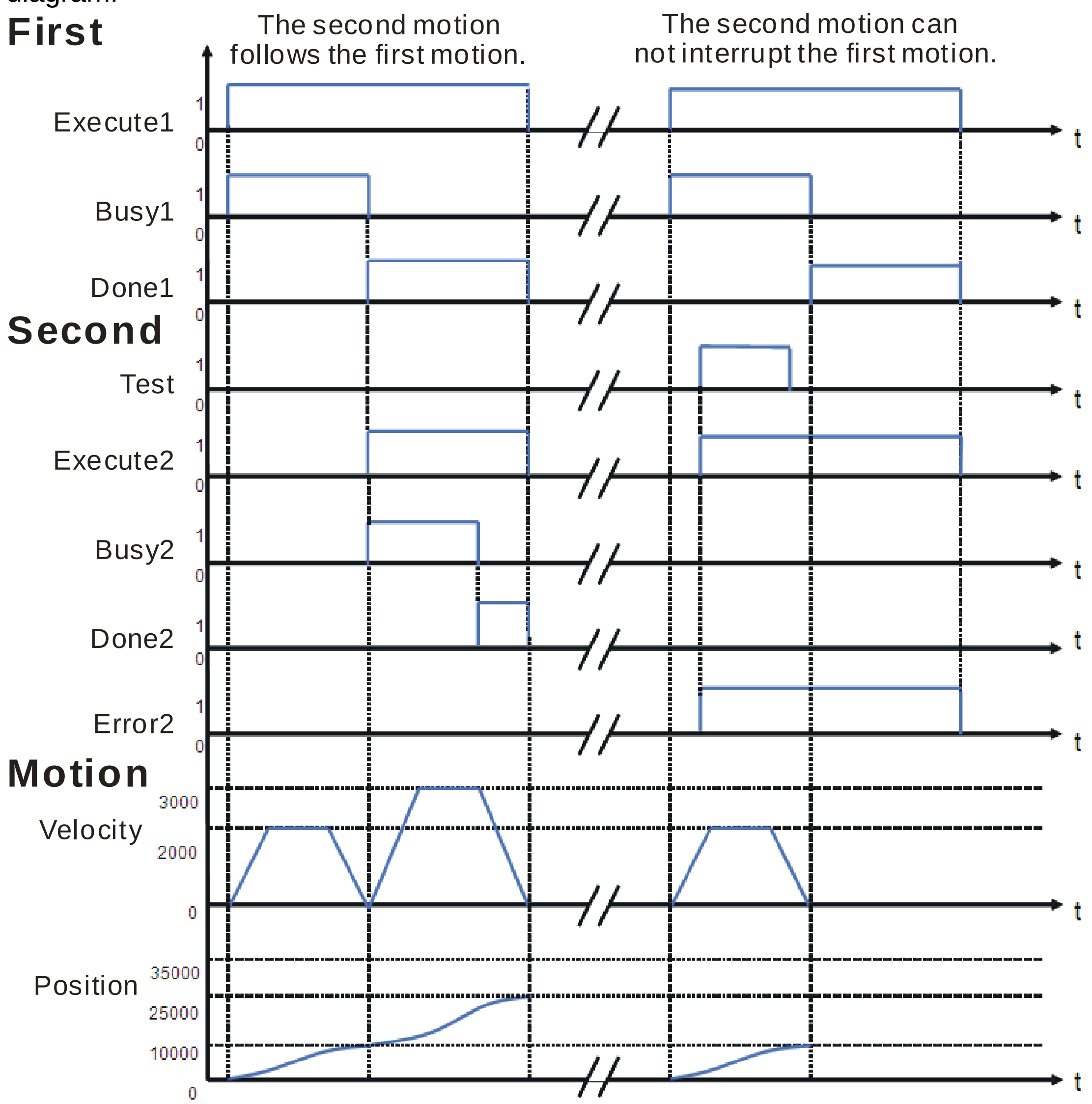
- After the first single-speed motion is complete, the second single-speed motion will be executed.
- The second single-speed motion is executed before the execution of the first single-speed motion is complete.

The motion control function block named FIRST is set so that the first axis moves at a speed of 2,000 pulses per second, and moves for 10,000 pulses. The motion control function block named SECOND is set so that the first axis moves at a speed of 3,000 pulses per second, and moves for 15,000 pulses.



- After the first single-speed motion is complete, the second single-speed motion will be executed.
  Steps:
  (a) Set Execute1 to True.
  (b) Wait for a transition in Done2's signal from low to high or a transition in Error2's signal from low to high.
- The second single-speed motion is executed before the execution of the first single-speed motion is complete.
  Steps:
  (a) Set Execute1 to True.
  (b) Set Test to ON when Busy1 is set to true.
  (c) Wait for a transition in Done2's signal from low to high or a transition in Error2's signal from low to high.

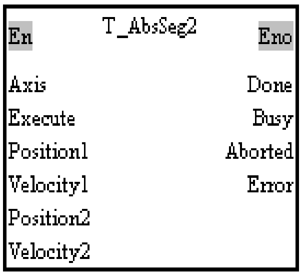
Timing diagram:



- After the first single-speed motion is complete, the second single-speed motion will be executed. When the motion control function block named FIRST is executed, the first axis moves for 10,000 pulses. After the execution of the motion control function block named FIRST is complete, the motion control function block named SECOND will be executed. When the motion control function block named SECOND is executed, the first axis moves for 15,000 pulses.
- The second single-speed motion is executed before the execution of the first single-speed motion is complete.
  When Error2 is set to True, the first axis moves for 10,000 pulses. The motion control function block named SECOND is invalid.

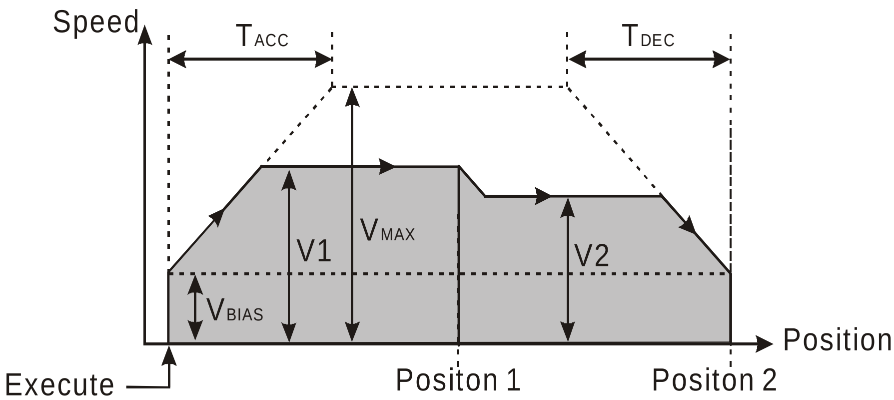
5. Modules which are supported
   The motion control function block T_RelSeg1 supports DVP20PM00M and DVP20PM00D.

5.10.3 Absolute Two-speed Motion



1. Motion control function block

   The motion control function block T_AbsSeg2 is used to start absolute two-speed motion. After absolute two-speed motion is started, the speed of the absolute two-speed motion will increase from the V_{BIAS} set to the V (I) set. The speed of the absolute two-speed motion will not increase/decrease from the V (I) set to the V (II) set until the present command position of the axis specified is near the P (I) set. The speed of the absolute two-speed motion will not decrease from the V (II) set to the V_{BIAS} set until the present command position of the axis specified is near the P (II) set. The P (I) set must be between the present command position of the axis specified and the P (II) set. Users can set the Vbias input pin, the Vmax input pin, the Tacc input pin, and the Tdec input pin in the motion control function block T_AxisSetting1. The number of pulses is a unit for the Position1 input pin/the Position2 input pin, and the number of pulses per second is a unit for the Velocity1 input pin/the Velocity2 input pin. The users can change the unit used by means of the motion control function block T_AxisSetting2.



2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
Position1	Absolute position of the first motion	DWORD	K-2,147,483,648~ K2,147,483,647	The value of the Position1 input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Velocity1	Target speed of the first motion	DWORD	K1~K2,147,483,647	The value of the Velocity1 input pin is valid when there is a transition in the Execute input pin's signal from low to high.

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Position1	Absolute position of the second motion	DWORD	K-2,147,483,648~ K2,147,483,647 (If the value of the Position1 input pin is greater than 0, the value of the Position2 input pin must be greater than or equal to the value of the Position1 input pin. If the value of the Position1 input pin is less than or equal to 0, the value of the Position2 input pin must be less than or equal to the value of the Position1 input pin.)	The value of the Position2 input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Velocity2	Target speed of the second motion	DWORD	K1~K2,147,483,647	The value of the Velocity2 input pin is valid when there is a transition in the Execute input pin's signal from low to high.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	● There is a transition in the Done output pin's signal when motion is complete.	● There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. ● If the Execute input pin is set to False when motion is complete, the Done output pin will be set to False in the next cycle.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. ● If the Execute input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The axis specified is in motion before the motion control function block is executed.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

The number of pulses is a unit for the Position1 input pin/the Position2 input pin, and the number of pulses per second is a unit for the Velocity1 input pin/the Velocity2 input pin. Users can change the unit used by means of the motion control function block T_AxisSetting2.

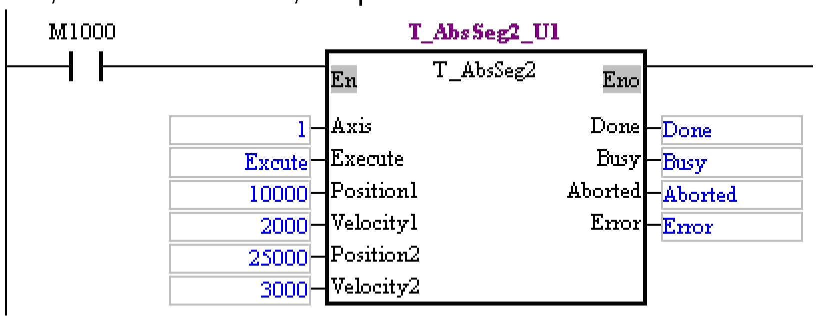
3. Troubleshooting

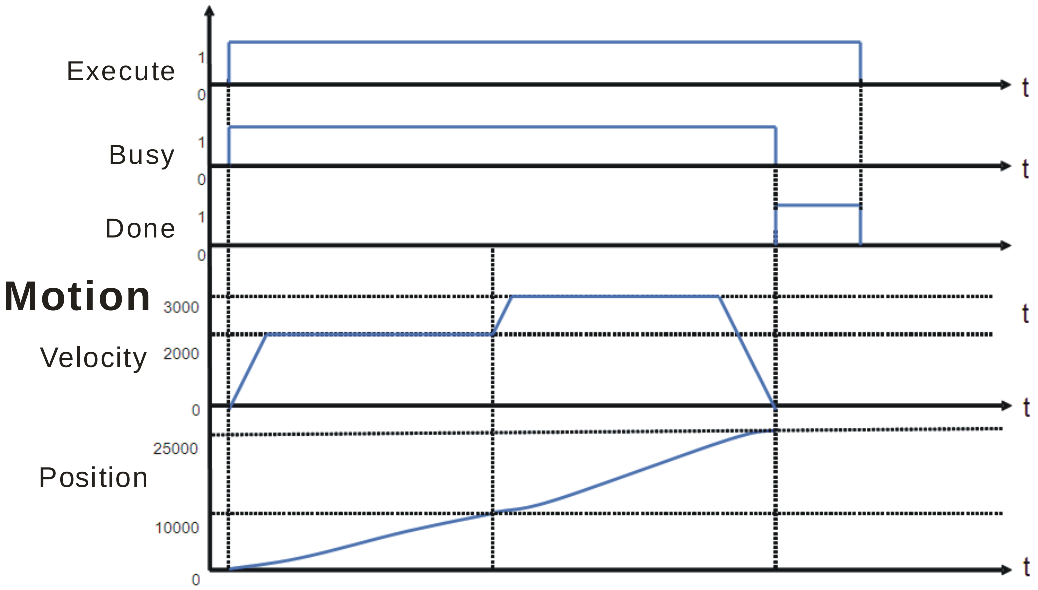
Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The motion control function block conflicts with other motion control function blocks.	Make sure that other uniaxial motion control function blocks are not started or the execution of other uniaxial motion control function blocks is complete before the motion control function block is started.

4. Example

Purposes:

- The motion control function block T_AbsSeg2 is used to start absolute two-speed motion of an axis. The first motion is set so that the first axis moves at a speed of 2,000 pulses per second, and moves for 10,000 pulses. The second motion is set so that the first axis moves at a speed of 3,000 pulses per second, and moves for 15,000 pulses.



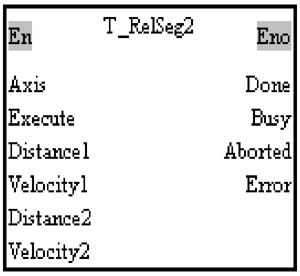


After the motion control function block is started, the first axis moves for 10,000 pulses at a speed of 2,000 pulses per second, and moves for 15,000 pulses at a speed of 3,000 pulses per second.

5. Modules which are supported

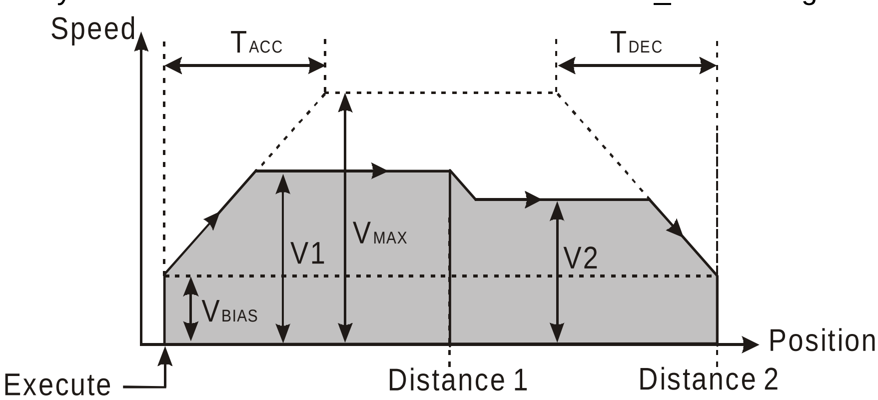
   The motion control function block T_AbsSeg2 supports DVP20PM00D and DVP20PM00M.

5.10.4 Relative Two-speed Motion



1. Motion control function block

   The motion control function block T_RelSeg2 is used to start relative two-speed motion. After relative two-speed motion is started, the speed of the relative two-speed motion will increase from the V_{BIAS} set to the V (I) set. The speed of the relative two-speed motion will not increase/decrease from the V (I) set to the V (II) set until the number of pulses output is near the value of the Distance1 input pin. The speed of the relative two-speed motion will not decrease from the V (II) set to the V_{BIAS} set until the number of pulses output is near the value of the Distance2 input pin. Users can set the Vbias input pin, the Vmax input pin, the Tacc input pin, and the Tdec input pin in the motion control function block T_AxisSetting1. The number of pulses is a unit for the Distance1 input pin/the Distance2 input pin, and the number of pulses per second is a unit for the Velocity1 input pin/the Velocity2 input pin. Users can change the unit used by means of the motion control function block T_AxisSetting2.



2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
Distance1	Relative distance for which the first motion moves	DWORD	K-2,147,483,648~ K2,147,483,647	The value of the Distance1 input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Velocity1	Target speed of the first motion	DWORD	K1~K2,147,483,647	The value of the Velocity1 input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Distance2	Relative distance for which the second motion moves	DWORD	K-2,147,483,648~ K2,147,483,647 (If the value of the Distance1 input pin is a positive value, the value of the Distance2 input pin must be a positive value. If the value of the Distance1 input pin is a negative value, the value of the Distance2 input pin must be a negative value.)	The value of the Distance2 input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Velocity2	Target speed of the second motion	DWORD	K1~K2,147,483,647	The value of the Velocity2 input pin is valid when there is a transition in the Execute input pin's signal from low to high.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	● There is a transition in the Done output pin's signal when motion is complete.	● There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. ● If the Execute input pin is set to False when motion is complete, the Done output pin will be set to False in the next cycle.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. ● If the Execute input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The axis specified is in motion before the motion control function block is executed.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

The number of pulses is a unit for the Distance1 input pin/the Distance2 input pin, and the number of pulses per second is a unit for the Velocity1 input pin/the Velocity2 input pin. Users can change the unit used by means of the motion control function block T_AxisSetting2.

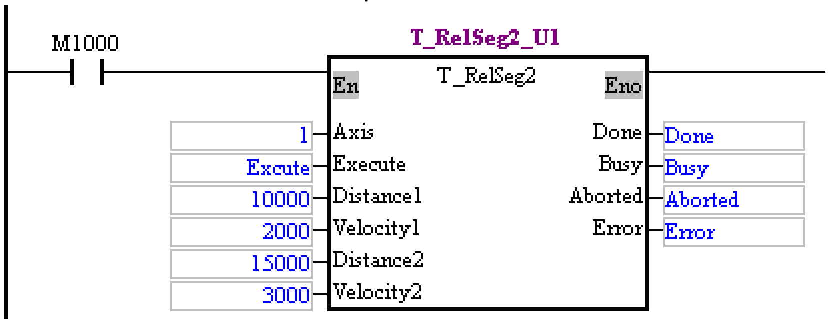
3. Troubleshooting

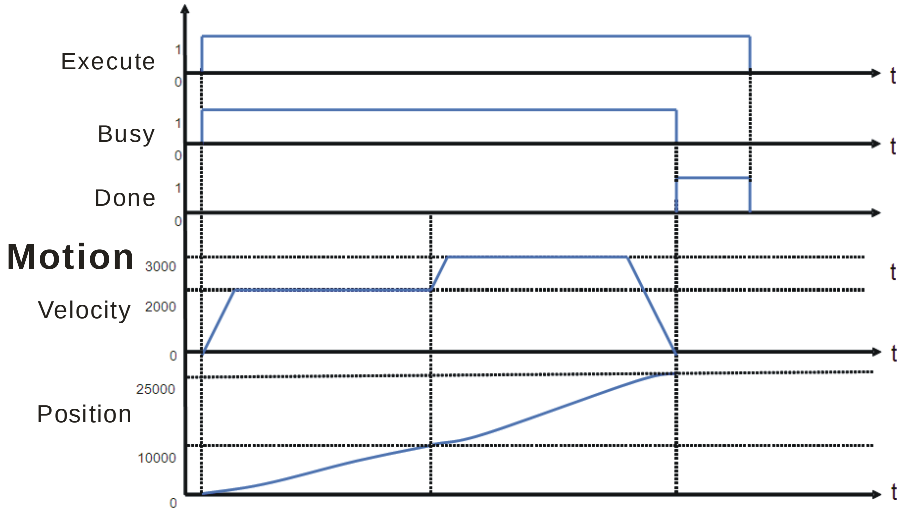
Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The motion control function block conflicts with other motion control function blocks.	Make sure that other uniaxial motion control function blocks are not started or the execution of other uniaxial motion control function blocks is complete before the motion control function block is started.

4. Example

Purpose:

- The motion control function block T_AbsSeg2 is used to start relative two-speed motion of an axis. The first motion is set so that the first axis moves at a speed of 2,000 pulses per second, and moves for 10,000 pulses. The second motion is set so that the first axis moves at a speed of 3,000 pulses per second, and moves for 15,000 pulses.



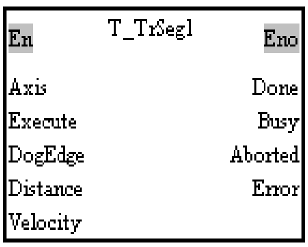


After the motion control function block is started, the first axis moves for 10,000 pulses at a speed of 2,000 pulses per second, and moves for 15,000 pulses at a speed of 3,000 pulses per second.

5. Modules which are supported

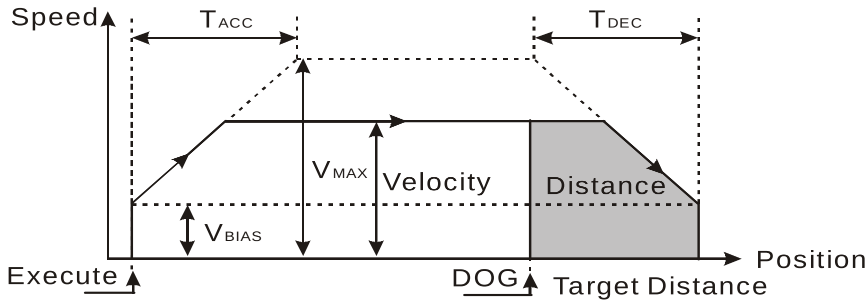
The motion control function block T_RelSeg2 supports DVP20PM00D and DVP20PM00M.

5.10.5 Inserting Single-speed Motion



1. Motion control function block

The motion control function block T_TrSeg1 is used to insert single-speed motion. The speed of motion increases from the V_{BIAS} set to the velocity set. After DOG's signal goes from low to high or from high to low, the DVP-10PM series motion controller used will continue sending pulses. The speed of the motion will not decrease from the velocity set to the V_{BIAS} set until the number of pulses output is near the value of the Distance input pin. Users can set the Vbias input pin, the Vmax input pin, the Tacc input pin, and the Tdec input pin in the motion control function block T_AxisSetting1. The number of pulses is a unit for the Distance input pin, and the number of pulses per second is a unit for the Velocity input pin. The users can change the unit used by means of the motion control function block T_AxisSetting2. If the value of the DogEdge input pin is mcRising, motion will be triggered by a transition in DOG's signal from low to high. If the value of the DogEdge input pin is mcFalling, motion will be triggered by a transition in DOG's signal from high to low.



2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
DogEdge	Transition in DOG's signal from low to high or from high to low	BOOL	mcRising (True)/ mcFalling (False)	The value of the DogEdge input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Distance	Distance for which motion moves after a transition in DOG's signal from low to high or from high to low	DWORD	K-2,147,483,648~ K2,147,483,647	The value of the Distance input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Velocity	Target speed	DWORD	K1~K2,147,483,647	The value of the Velocity input pin is valid when there is a transition in the Execute input pin's signal from low to high.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal when motion is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. • If the Execute input pin is set to False when motion is complete, the Done output pin will be set to False in the next cycle.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. ● If the Execute input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The axis specified is in motion before the motion control function block is executed.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

The number of pulses is a unit for the Distance input pin, and the number of pulses per second is a unit for the Velocity input pin. Users can change the unit used by means of the motion control function block T_AxisSetting2. If the value of the DogEdge input pin is mcRising, motion will be triggered by a transition in DOG's signal from low to high. If the value of the DogEdge input pin is mcFalling, motion will be triggered by a transition in DOG's signal from high to low.

3. Troubleshooting

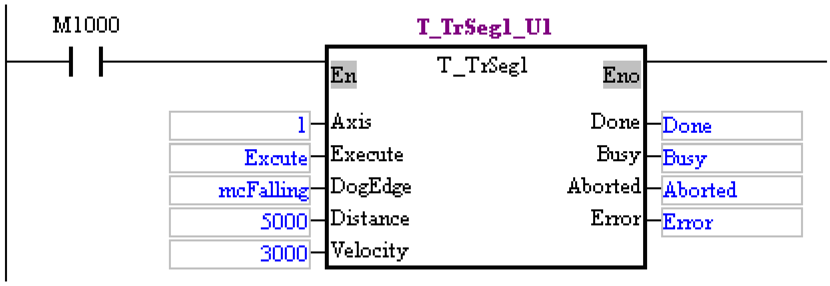
Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The motion control function block conflicts with other motion control function blocks.	Make sure that other uniaxial motion control function blocks are not started or the execution of other uniaxial motion control function blocks is complete before the motion control function block is started.

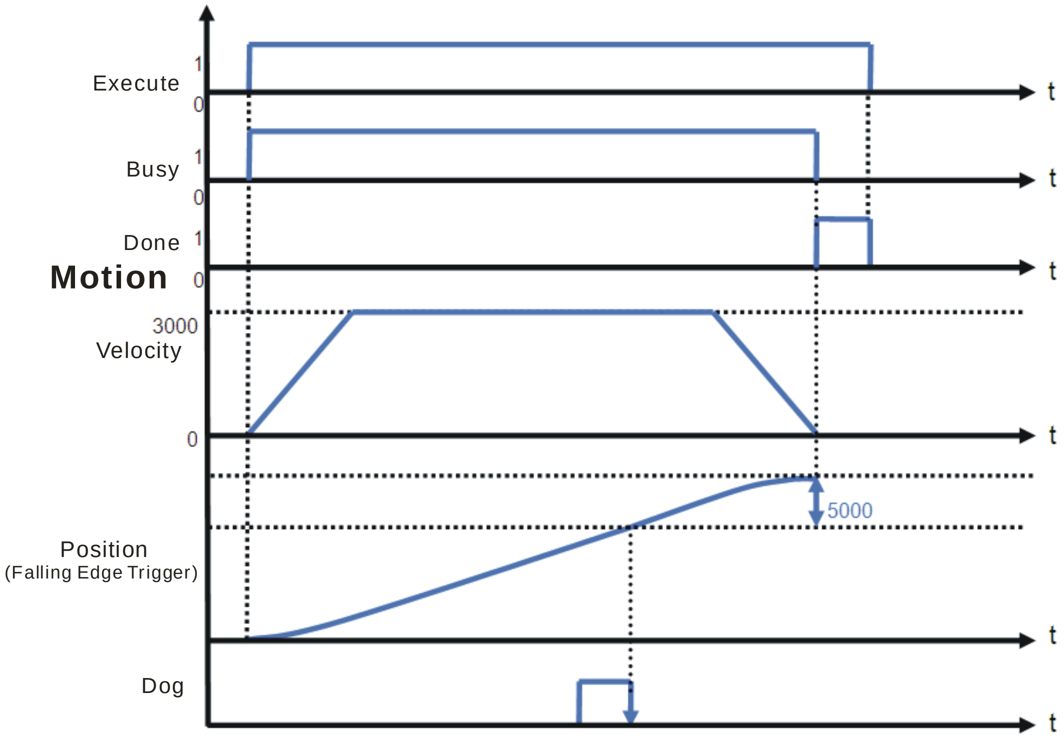
4. Examples

Example 1:

- The motion control function block T_TrSeg1 is used to insert single-speed motion which is triggered by a transition in DOG's signal from high to low.

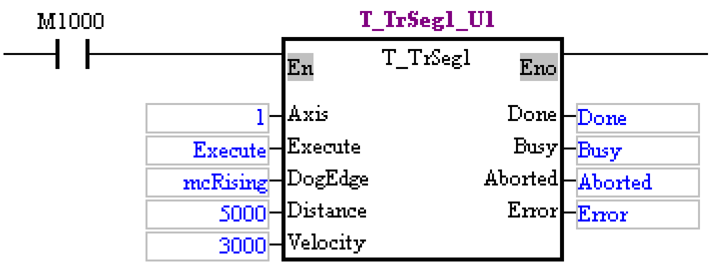
  The motion control function block named T_TrSeg1_U1 is set so that the first axis moves at a speed of 3,000 pulses per second, and will move for 5,000 pulses after a transition in DOG's signal from high to low. After the first axis moves for 5,000 pulses, Done will be set to True.

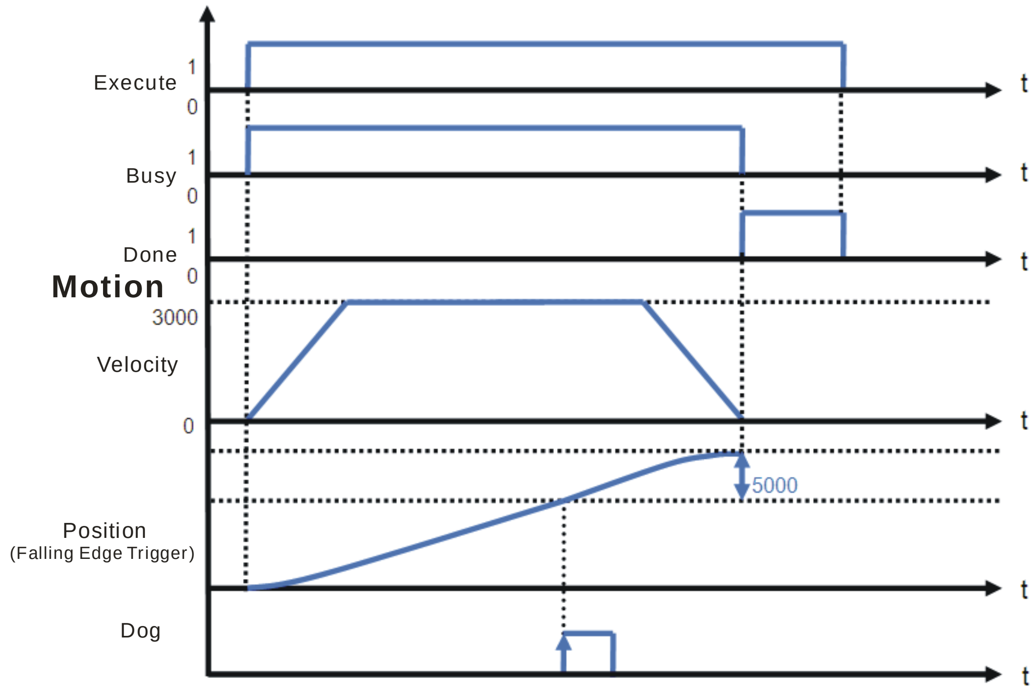




Example2:

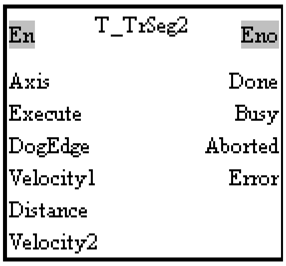
- The motion control function block T_TrSeg1 is used to insert single-speed motion which is triggered by a transition in DOG's signal from low to high.
  The motion control function block named T_TrSeg1_U1 is set so that the first axis moves at a speed of 3,000 pulses per second, and will move for 5,000 pulses after a transition in DOG's signal from low to high. After the first axis moves for 5,000 pulses, Done will be set to True.



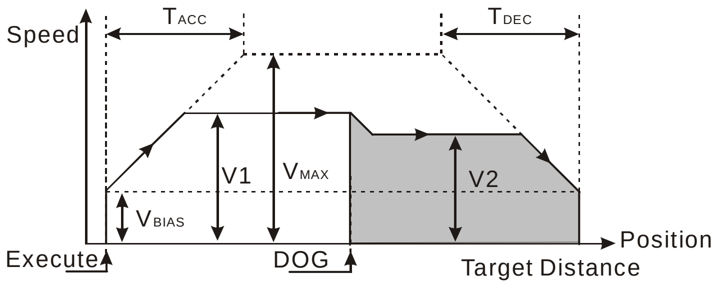


5. Modules which are supported
   The motion control function block T_TrSeg1 supports DVP20PM00D and DVP20PM00M.

5.10.6 Inserting Two-speed Motion



1. Motion control function block
   The motion control function block T_TrSeg2 is used to insert two-speed motion. The speed of motion increases from the V_{BIAS} set to the V (I) set. After DOG's signal goes from low to high or from high to low, the speed of the motion will increase/decrease from the V (I) set to the V (II) set. The motion will not stop until the number of pulses output is near the value of the Distance input pin. Users can set the Vbias input pin, the Vmax input pin, the Tacc input pin, and the Tdec input pin in the motion control function block T_AxisSetting1. The number of pulses is a unit for the Distance input pin, and the number of pulses per second is a unit for the Velocity1 input pin/the Velocity2 input pin. The users can change the unit used by means of the motion control function block T_AxisSetting2. If the value of the DogEdge input pin is mcRising, motion will be triggered by a transition in DOG's signal from low to high. If the value of the DogEdge input pin is mcFalling, motion will be triggered by a transition in DOG's signal from high to low.



2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
DogEdge	Transition in DOG's signal from low to high or from high to low	BOOL	mcRising (True)/ mcFalling (False)	The value of the DogEdge input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Velocity1	Target speed before a transition in DOG's signal from low to high or from high to low	DWORD	K1~K2,147,483,647	The value of the Velocity1 input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Distance	Distance for which motion moves after a transition in DOG's signal from low to high or from high to low	DWORD	K-2,147,483,648~ K2,147,483,647	The value of the Distance input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Velocity2	Target speed after a transition in DOG's signal from low to high or from high to low	DWORD	K1~K2,147,483,647	The value of the Velocity2 input pin is valid when there is a transition in the Execute input pin's signal from low to high.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal when motion is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. • If the Execute input pin is set to False when motion is complete, the Done output pin will be set to False in the next cycle.

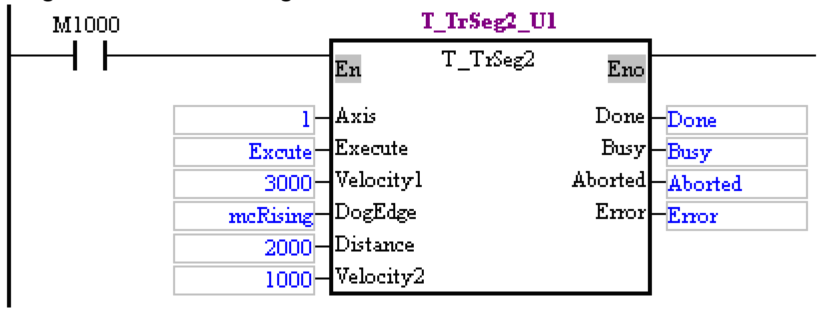
Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. • If the Execute input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The axis specified is in motion before the motion control function block is executed.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

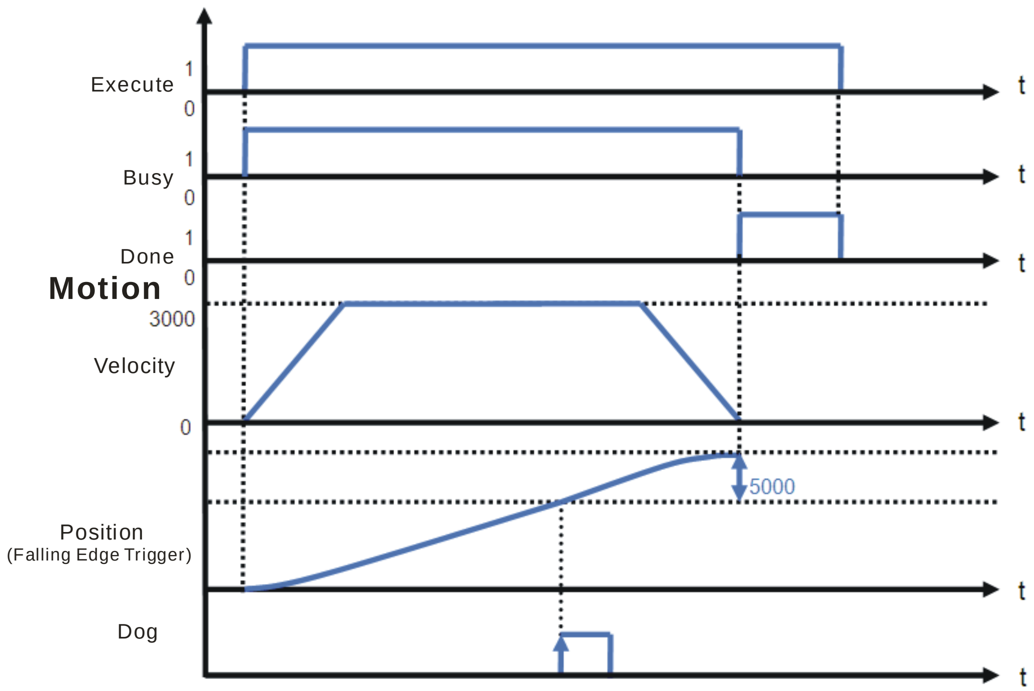
The number of pulses is a unit for the Distance input pin, and the number of pulses per second is a unit for the Velocity1 input pin/the Velocity2 input pin. Users can change the unit used by means of the motion control function block T_AxisSetting2. If the value of the DogEdge input pin is mcRising, motion will be triggered by a transition in DOG's signal from low to high. If the value of the DogEdge input pin is mcFalling, motion will be triggered by a transition in DOG's signal from high to low.

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The motion control function block conflicts with other motion control function blocks.	Make sure that other uniaxial motion control function blocks are not started or the execution of other uniaxial motion control function blocks is complete before the motion control function block is started.

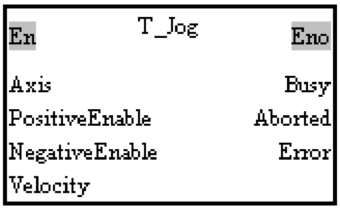
4. Example
   The motion control function block T_TrSeg2 is used to insert two-speed motion which is triggered by a transition in DOG's signal from low to high.
   The motion control function block named T_TrSeg2_U1 is set so that the first axis moves at a speed of 3,000 pulses per second, and will move for 2,000 pulses at a speed of 1,000 pulses per second after a transition in DOG's signal from low to high.



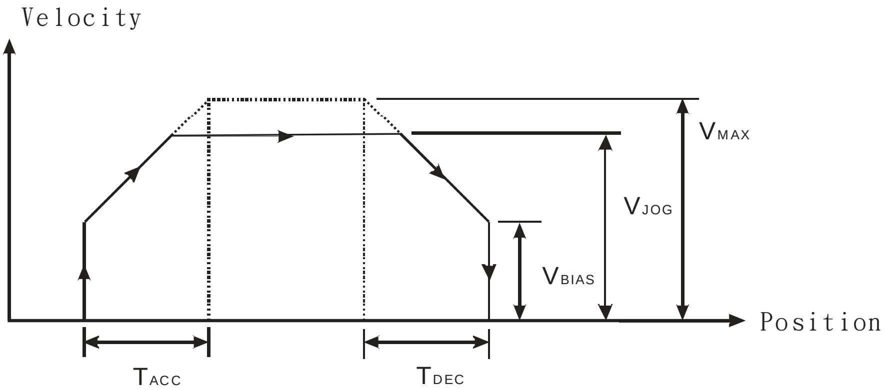


   After the first axis moves for 2,000 pulses, Done will be set to True.
5. Modules which are supported
   The motion control function block T_TrSeg2 supports supports DVP20PM00D and DVP20PM00M.

5.10.7 JOG Motion



1. Motion control function block
   The motion control function block T_Jog is used to start JOG motion. The value of the Axis input pin indicates an axis number, and the value of the Velocity input pin indicates the speed of JOG motion. If the PositiveEnable input pin is set to True, positive JOG motion will be started. If the NegativeEnable input pin is set to True, negative JOG motion will be started. The number of pulses per second is a unit for the Velocity input pin. Users can change the unit used by means of the motion control function block T_AxisSetting2.



2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin’s signal from low to high.
PositiveEnable	Enabling positive JOG motion	BOOL	True/False	● If the PositiveEnable input pin and the NegativeEnable input pin are set to True simultaneously, positive JOG motion will be enabled, and the NegativeEnable input pin will be reset to False. ● If the PositiveEnable input pin is set to True after the NegativeEnable input pin is set to True, the NegativeEnable input pin will be reset to False, the negative JOG motion will stop, and the positive JOG motion will be enabled.

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
NegativeEnable	Enabling negative JOG motion	BOOL	True/False	● If the PositiveEnable input pin and the NegativeEnable input pin are set to True simultaneously, positive JOG motion will be enabled, and the NegativeEnable input pin will be reset to False. ● If the NegativeEnable input pin is set to True after the PositiveEnable input pin is set to True, the PositiveEnable input pin will be reset to False, the positive JOG motion will stop, and the negative JOG motion will be enabled.
Velocity	Target speed	DWORD	K1~K2,147,483,647	When the motion control function block is executed, the value of the Velocity input pin is updated repeatedly.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the PositiveEnable input pin's signal from low to high or when there is a transition in the NegativeEnable input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when motion stops. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Aborted output pin's signal from high to low when there is a transition in the PositiveEnable input pin's signal from high to low or when there is a transition in the NegativeEnable input pin's signal from high to low. ● If the PositiveEnable input pin and the NegativeEnable are set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.

Output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The axis specified is in motion before the motion control function block is executed.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the PositiveEnable input pin's signal from high to low or when there is a transition in the NegativeEnable input pin's signal from high to low.

The number of pulses per second is a unit for the Velocity input pin. Users can change the unit used by means of the motion control function block T_AxisSetting2.

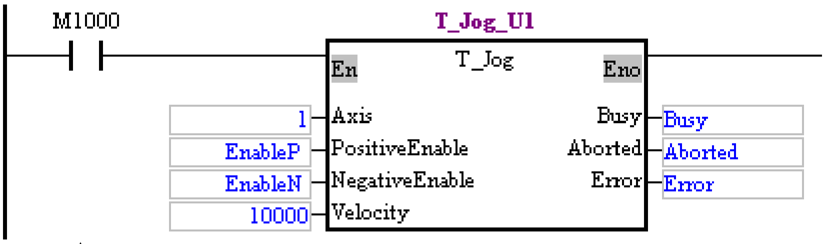
3. Troubleshooting

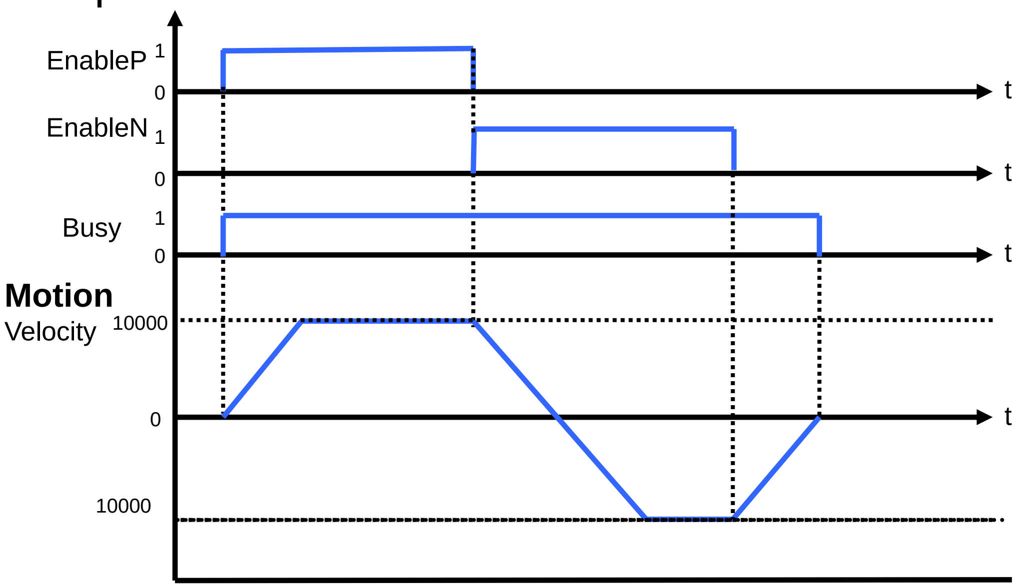
Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The motion control function block conflicts with other motion control function blocks.	Make sure that other uniaxial motion control function blocks are not started or the execution of other uniaxial motion control function blocks is complete before the motion control function block is started.

4. Example

The motion control function block T_Jog is used to start JOG motion. Positive JOG motion is enabled by EnableP, and negative JOG motion is enabled by EnableN.

The first axis moves at a speed of 10,000 pulses per second. If EnableP is set to 1, the first axis will move in the positive direction. If EnableN is set to 1, the first axis will move in the negative direction.



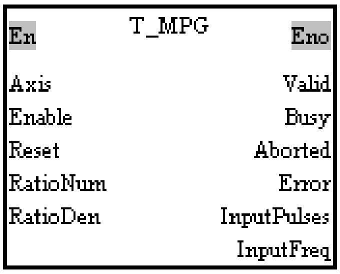


When EnableP is set to 1, the first axis moves at a speed of 10,000 pulses per second in the positive direction. When EnableN is set to 1, the first axis moves at a speed of 10,000 pulses per second in the negative direction. When EnableP and EnableN are not set to 1, the first axis stops moving.

5. Modules which are supported

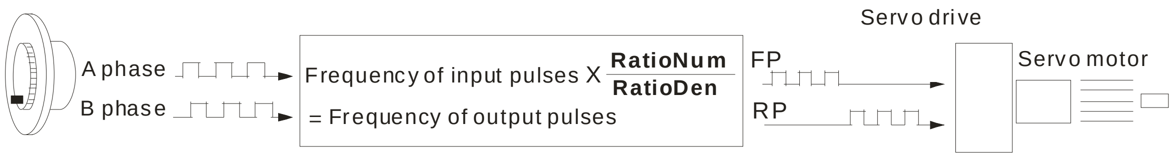
The motion control function block T_Jog supports DVP20PM00D and DVP20PM00M.

5.10.8 Manual Pulse Generator Mode

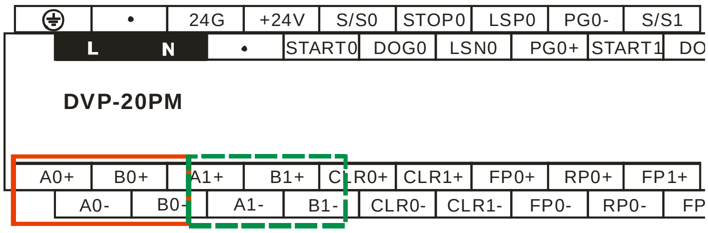


1. Motion control function block

   The motion control function block T_MPG is used to enable a manual pulse generator mode. The value of the Axis input pin indicates an axis number. The motion of the axis specified follows the operation of a manual pulse generator. The relation between the position of the axis specified and the input pulses generated by the manual pulse generator used is determined by the RatioNum input pin and the RatioDen input pin. The speed at which the manual pulse generator used responds depends on the value of the Tacc input pin and the value of the Tdec input pin. Users can set the Tacc input pin and the Tdec input pin in the motion control function block T_AxisSetting1.



   The input terminals which can be connected to manual pulse generators are shown below.



   The terminals in the solid frame are for the first axis. The terminals in the dotted frame are for the second axis and the third axis. (DVP20PM00M supports the third axis.)

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	Manual pulse generator mode	BOOL	True/False	-
Reset	Resetting the manual pulse generator used	BOOL	True/False	The value of the Reset input pin is valid when there is a transition in the Enable input pin's signal from low to high.
RatioNum	Numerator of an electronic gear ratio	DWORD	K0~K32,767	When the motion control function block is executed, the value of the RatioNum input pin is updated repeatedly.
RatioDen	Denominator of an electronic gear ratio	DWORD	K1~K32,767	When the motion control function block is executed, the value of the RatioDen input pin is updated repeatedly.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	The execution of the motion control function block is complete.	BOOL	● There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Valid output pin's signal from high to low when motion stops. ● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. ● If the Enable input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The axis specified is in motion before the motion control function block is executed.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

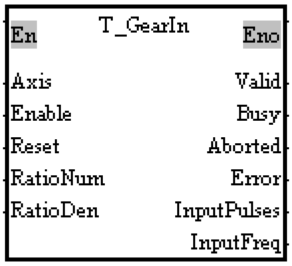
Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
InputPulses	Number of pulses generated by the manual pulse generator used	DWORD	K-2,147,483,648~ K2,147,483,647	When the motion control function block is executed, the value of the InputPulses output pin is updated repeatedly.
InputFreq	Frequency of pulses generated by the manual pulses generator used	DWORD	K0~K2,147,483,647	When the motion control function block is executed, the value of the InputFreq output pin is updated repeatedly.

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The motion control function block conflicts with other motion control function blocks.	Make sure that other uniaxial motion control function blocks are not started or the execution of other uniaxial motion control function blocks is complete before the motion control function block is started.

4. Modules which are supported
   The motion control function block T_MPG support DVP20PM00D and DVP20PM00M.

5.10.9 Electronic Gear Motion



1. Motion control function block
   The value of the RatioNum input pin is the numerator of an electronic gear ratio. The value of the RatioDen input pin is the denominator of an electronic gear ration. The Reset input pin is used to clear the number of input pulses. The speed at which the electronic gear used responds does not depend on the value of the Tacc input pin and the value of the Tdec input pin. The electronic gear used operates in accordance with the source of input. The input terminals for electronic gear motion are the same as the input terminals which can be connected to a manual pulse generator.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Slave axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Enable input pin’s signal from low to high.
Enable	Enabling electronic gear motion	BOOL	True/False	-
Reset	Resetting the InputPulses output pin	BOOL	True/False	The value of the Reset input pin is valid when there is a transition in the Enable input pin’s signal from low to high.
RatioNum	Numerator of an electronic gear ratio	DWORD	K-32,767~K32,767	When the motion control function block is executed, the value of the RatioNum input pin is updated repeatedly.
RatioDen	Denominator of an electronic gear ratio	DWORD	K1~K32,767	When the motion control function block is executed, the value of the RatioDen input pin is updated repeatedly.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	● There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Valid output pin's signal from high to low when motion stops. ● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. ● If the Enable input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The axis specified is in motion before the motion control function block is executed.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
InputPulses	Number of input pulses	DWORD	K-2,147,483,648~ K2,147,483,647	When the motion control function block is executed, the value of the InputPulses output pin is updated repeatedly.
InputFreq	Frequency of input pulses	DWORD	K 0~K 2,147,483,647	When the motion control function block is executed, the value of the InputFreq output pin is updated repeatedly.

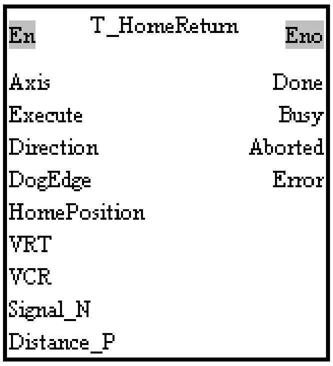
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported

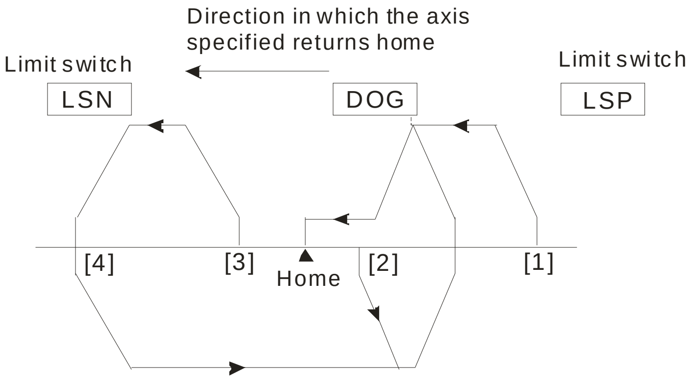
The motion control function block T_GearIn supports DVP20PM00D and DVP20PM00M.

5.10.10 Returning Home



1. Motion control function block

   The motion control function block T-HomeReturn is used to start motion of returning home. The value of the Axis input pin indicates an axis number, and the value of the Direction input pin indicates whether the axis specified returns home in the positive direction or in the negative direction. The value of the VRT input pin indicates the speed at which the axis specified returns home. The value of the DogEdge input pin indicates whether motion is triggered by a transition in DOG's signal from low to high or from high to low. The value of the VCR input pin indicates the speed to which the speed of the axis specified decreases. The value of the Signal_N input pin is the number of zero pulses. The value of the Distance_P is the number of supplementary pulses needed. After motion of returning home is complete, the value of the HomePosition input pin will be taken as the present position of the axis specified. Motion of returning home is shown below.



   Position (1): Position [1] is at the right side of the home and DOG, and DOG is OFF.
   Position (2): Position [2] is at the right side of the home, and DOG is ON.
   Position (3): Position [3] is at the left side of the home and DOG. DOG is OFF, and LSN is OFF.
   Position (4): Position [4] is at the left side of the home and DOG. DOG is OFF, and LSN is ON.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Direction	Direction in which the axis specified returns home	BOOL	mcNegative (False)/ mcPositive (True)	The value of the Direction input pin is valid when there is a transition in the Execute input pin's signal from low to high.
DogEdge	Transition in DOG's signal from low to high or from high to low	BOOL	mcFalling (False)/ mcRising (True)	The value of the DogEdge input pin is valid when there is a transition in the Execute input pin's signal from low to high.
HomePosition	Home position	DWORD	K-2,147,483,648~ K2,147,483,647	The value of the HomePosition input pin is valid when there is a transition in the Execute input pin's signal from low to high.
VRT	Speed at which the axis specified returns home	DWORD	K1~K1000000	The value of the VRT input pin is valid when there is a transition in the Execute input pin's signal from low to high.
VCR	Speed to which the speed of the axis specified decreases	DWORD	K1~VRT	The value of the VCR input pin is valid when there is a transition in the Execute input pin's signal form low to high.
Signal_N	Number of zero pulses	WORD	K0~K32,767	The value of the Signal_N input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Distance_P	Number of supplementary pulses	WORD	K-32768~K32,767	The value of the Distance_P input pin is valid when there is a transition in the Execute input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal from low to high when motion of returning home is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. • If the Execute input pin is set to False when the execution of the motion control function block is complete, the Done output pin will be set to False in the next cycle.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. ● If the Execute input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The axis specified is in motion before the motion control function block is executed.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The motion control function block conflicts with other motion control function blocks.	Make sure that other uniaxial motion control function blocks are not started or the execution of other uniaxial motion control function blocks is complete before the motion control function block is started.

4. Modules which are supported
   The motion control function block T-HomeReturn supports DVP20PM00D and DVP2OPM00M.

5.10.11 Setting Input Sources for Manual Pulse Generators/Electronic Gears

1. Motion control function block
   The motion control function block T_MPGGearSource is used to set input sources for manual pulse generators/electronic gears. If the Source input pin is set to True, the input sources for the first axis~the third axis will be A0± and B0±. If the Source input pin is set to False, the input sources for the first axis

will be A0± and B0±, and the input sources for the second axis and the third axis will be A1± and B1±.

2. Input pins/Output pins

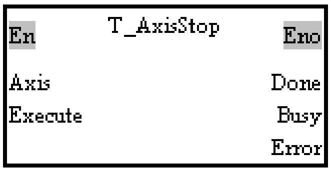
Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Enable	Enabling the setting of input sources for manual pulse generators/electronic gears	BOOL	True/False	-
Source	Input sources	BOOL	True/False	The value of the Source input pin is valid when there is a transition in the Execute input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

3. Modules which are supported

The motion control function block T_MPGGearSource supports DVP20PM00D and DVP20PM00M.

5.10.12 Stopping Uniaxial Motion



1. Motion control function block

The motion control function block T_AxisStop is used to stop the motion of the axis specified. The value of the Axis input pin indicates an axis number.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal from low to high when motion of returning home is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. • If the Execute input pin is set to False when the execution of the motion control function block is complete, the Done output pin will be set to False in the next cycle.
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The motion of the axis specified is not uniaxial motion, gear motion, or cam motion.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

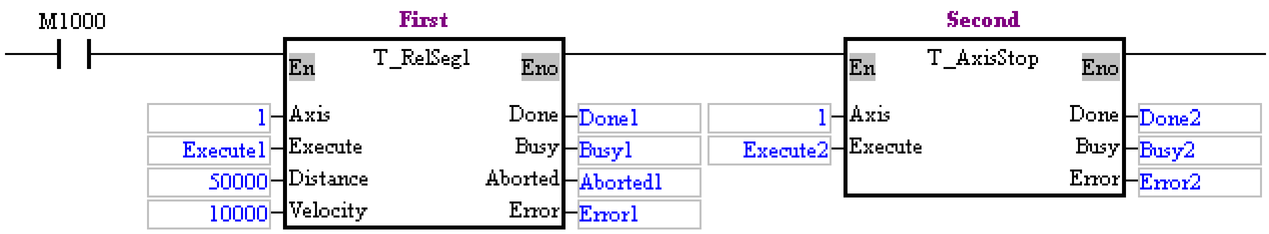
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The motion control function block conflicts with other motion control function blocks.	Make sure that other uniaxial motion control function blocks are not started or the execution of other uniaxial motion control function blocks is complete before the motion control function block is started.

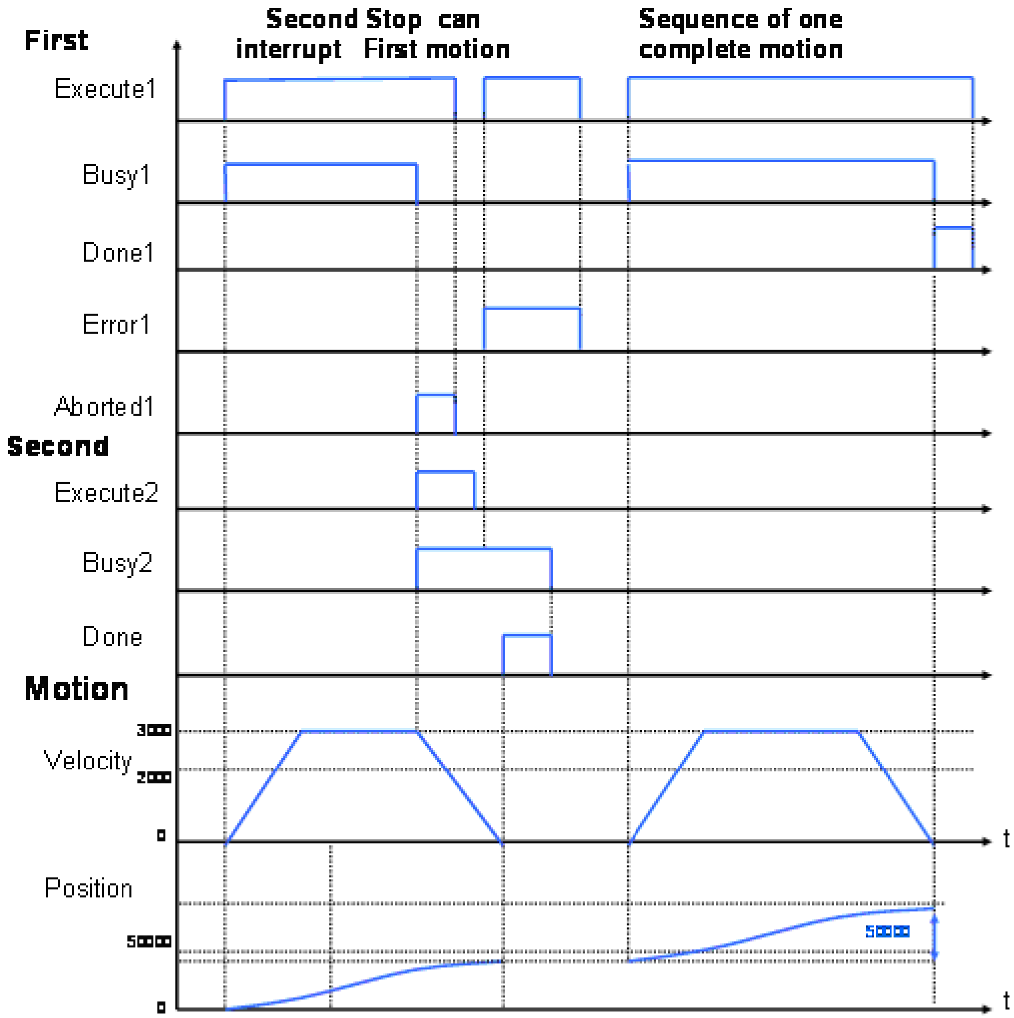
4. Example

The single-speed motion of an axis is started, and then the motion control function block T_AxisStop is used to stop the motion.

The motion control function block named First is used to start single-speed motion. It is set so that the first axis moves for 50,000 pulses at a speed of 10,000 per second. The motion control function block named Second is used to stop the motion of the first axis.



The motion control function block named First is started. Before Done 1 is set to True, Execute2 is used to start the motion control function block named Second.

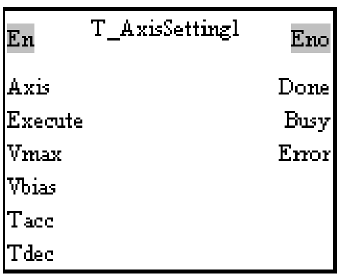


After the motion control function block named First is started, the first axis will move at a speed of 10,000 pulses per second. After the motion control function block named Second is started, Aborted1 will be set to True, Busy1 will be set to False, and the first axis will stop moving. When the motion control function block named Second is used to stop the motion of the first axis, no motion can be started. If any motion is started, an error will occur.

5. Modules which are supported

The motion control function block T_AxisStop supports DVP20PM00D and DVP20PM00D.

5.10.13 Parameter Setting I



1. Motion control function block

The motion control function block T_AxisSetting1 is used to set motion parameters. The value of the Axis input pin indicates an axis number. Users can set the maximum speed of the axis specified, the start-up speed of the axis specified, the time it takes for the start-up speed of the axis specified to increase to its maximum speed, and the time it takes for the maximum speed of the axis specified to decrease to its start-up speed. The setting of the Unit input pin in the motion control function block

T_AxisSetting2 determines the unit for the Vmax input pin and the unit for the Vbias input pin.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
Vmax	Maximum speed	DWORD	K1~K2,147,483,647	The value of the Vmax input pin is valid when there is a transition in the Execute input pin's signal from low tot high.
Vbias	Start-up speed	DWORD	K0~K2,147,483,647	The value of the Vbias input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Tacc	Acceleration time (Unit: ms)	WORD	K0~K32,767	The value of the Tacc input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Tdec	Deceleration time (Unit: ms)	WORD	K0~K32,767	The value of the Tdec input pin is valid when there is a transition in the Execute input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	● There is a transition in the Done output pin's signal from low to high when motion of returning home is complete.	● There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. ● If the Execute input pin is set to False when the execution of the motion control function block is complete, the Done output pin will be set to False in the next cycle.

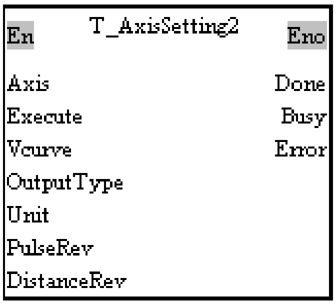
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported
   The motion control function block T_AxisSetting1 supports DVP20PM00D and DVP20PM00M.

5.10.14 Parameter Setting II



1. Motion control function block
   The motion control function block T_AxisSetting2 is used to set motion parameters. The value of the Axis input pin indicates an axis number. Users can set the velocity curve of the axis specified, an output type, and a unit. The setting of a unit requires the number of pulses it takes for a motor to rotate once and the distance for which the axis specified moves when the motor rotates once.
   There are three types of units. They are motor units, compound units, and mechanical units. The setting of a unit requires the number of pulses it takes for a motor to rotate once (the value of the PulseRev input pin) and the distance for which the axis specified moves when the motor rotates once (the value of the DistanceRev input pin). The units for positions and speeds are as shown below.

	Motor unit	Compound unit	Mechanical unit
Position	pulse	μm	μm
	pulse	mdeg	mdeg
	pulse	10^{-4} inches	10^{-4} inches
Speed	pulse/second	centimeter/minute	pulse/second
	pulse/second	10 degrees/minute	pulse/second
	pulse/second	inch/minute	pulse/second

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
Vcure	Velocity curve	BOOL	mcTrapezoid: False mcSCurve: True	The value of the Vcurve input pin is valid when there is a transition in the Execute input pin's signal from low to high.
OutputType	Output type	WORD	mcUD: 0 mcPD: 1 mcAB: 2 mc4AB: 3	The value of the OutputType input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Unit	Unit	WORD	mcMotor: 0 mcMachine: 1 mcComp: 2	The value of the Unit input pin is valid when there is a transition in the Execute input pin's signal from low to high.
PulseRev	Number of pulses it takes for a motor to rotate once	WORD	K1~K2,147,483,647	The value of the PulseRev input pin is valid when there is a transition in the Execute input pin's signal from low to high.
DistanceRev	Distance for which the axis specified moves when the motor used rotates once	WORD	K1~K2,147,483,647	The value of the DistanceRev input pin is valid when there is a transition in the Execute input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal from low to high when motion of returning home is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. • If the Execute input pin is set to False when the execution of the motion control function block is complete, the Done output pin will be set to False in the next cycle.

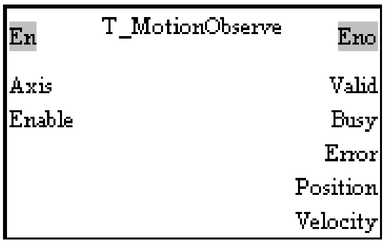
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported
   The motion control function block T_AxisSetting2 supports DVP20PM00D and DVP20PM00M.

5.10.15 Reading the Present Position/Speed of an Axis



1. Motion control function block
   The motion control function block T_MotionObserve is used to read the present position/speed of an axis. The value of the Axis input pin indicates an axis number. After the motion control function block is started, users can read the present position of the axis specified through the Position output pin, and the speed of the axis specified through the Velocity output pin.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	Manual pulse generator mode	BOOL	True/False	-

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	• There is a transition in the Valid output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. • There is a transition in the Valid output pin's signal from high to low when there is a transition in the Error input pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
Position	Present position (Pulse unit)	DWORD	K-2,147,483,648~ K2,147,483,647	When the motion control function block is executed, the value of the Position output pin is updated repeatedly.
Velocity	Present speed (Pulse unit)	DWORD	K0~K2,147,483,647	When the motion control function block is executed, the value of the Velocity output pin is updated repeatedly.

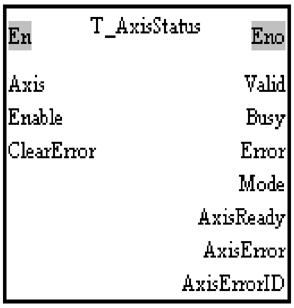
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported

   The motion control function block T_MotionObserve supports DVP20PM00D and DVP20PM00M.

5.10.16 State of an Axis



1. Motion control function block
   The motion control function block is T_AxisStatus is used to read and clear the present erroneous state of an axis. The value of the Axis input pin indicates an axis number. Users can clear the present erroneous state of the axis specified by means of the ClearError input pin. The value of the AxisErrorID output pin indicates the present erroneous state of the axis specified.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	Reading the present erroneous state of the axis specified.	BOOL	True/False	-
ClearError	The erroneous state of the axis specified is cleared when there is a transition in the ClearError input pin's signal from low to high.	BOOL	True/False	The value of the ClearError input pin is valid when the motion control function block is executed.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	● There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. ● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
Mode	Mode of motion	WORD	H0~H201 (*1)	When the motion control function block is executed, the value of the Mode output pin is updated repeatedly.
AxisReady	Ready flag for the axis specified	BOOL	True/False	When the motion control function block is executed, the value of the AxisReady output pin is updated repeatedly.
AxisError	Axis error flag	BOOL	True/False	When the motion control function block is executed, the value of the AxisError output pin is updated repeatedly.
AxisErrorID	Error code	WORD	H0002~HC4FF	When the motion control function block is executed, the value of the AxisErrorID output pin is updated repeatedly.

*1: Value of the Mode output pin

Value	Definition
H0	Idle
H100	Uniaxial motion is being stopped.
H101	Absolute single-speed motion
H102	Relative single-speed motion
H103	Absolute two-speed motion
H104	Relative two-speed motion
H105	Inserting single-speed motion
H106	Inserting two-speed motion
H107	JOG motion
H108	Manual pulse generator mode
H109	Motion of returning home
H10A	Electronic gear motion
H10B	Electronic cam motion
H10C	Uniaxial cyclic electronic cam motion
H10D	Uniaxial noncyclic electronic cam motion
H10E	Rotary cut
H10F	Flying shear
H200	G-code motion is being stopped.
H201	G-code motion

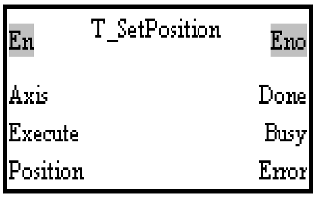
Please refer to appendix A in chapter 15 for more information about error codes.

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported
   The motion control function block is T_AxisStatus supports DVP20PM00D and DVP20PM00M.

5.10.17 Setting the Present Position of an Axis



1. Motion control function block
   The motion control function block T_SetPosition is used to set the present position of an axis. The value of the Axis input pin indicates an axis number. Users can set the present position of the axis specified by means of the Position input pin. Note: To prevent errors from occurring, please avoid using the motion control function block to set the present position of the master axis involved in cam motion or gear motion.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Axis	Motion axis number	WORD	DVP20PM00D: K1~K2 DVP20PM00M: K1~K3	The value of the Axis input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	Motion is started when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
Position	Present position of the axis specified	DWORD	K-2,147,483,648~ K2,147,483,647	The value of the Position input pin is valid when there is a transition in the Execute input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	● The writing of a position is complete.	● There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported
   The motion control function block T_SetPosition supports DVP20PM00D and DVP20PM00M.

5.10.18 Setting the Polarities of Input Terminals

1. Motion control function block
   The motion control function block T_InputPolarity is used to set the polarities of input terminals on the DVP-20PM series motion controller used. Users can set the polarities of input terminals on the DVP-20PM series motion controller used by means of input pins.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Enable	Manual pulse generator mode	BOOL	True/False	-
Pg0	Polarity	BOOL	mcNO: False mcNC: True	When the motion control function block is executed, the values of the input pins are updated repeatedly.
MPGB0	Polarity	BOOL		
MPGA0	Polarity	BOOL		
LSN0	Polarity	BOOL		
LSP0	Polarity	BOOL		
DOG0	Polarity	BOOL		

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
STOP0	Polarity	BOOL	mcNO: False mcNC: True	When the motion control function block is executed, the values of the input pins are updated repeatedly.
START0	Polarity	BOOL		
PG1	Polarity	BOOL		
MPGB1	Polarity	BOOL		
MPGA1	Polarity	BOOL		
LSN1	Polarity	BOOL		
LSP1	Polarity	BOOL		
DOG1	Polarity	BOOL		
STOP1	Polarity	BOOL		
START1	Polarity	BOOL		
PG2*	Polarity	BOOL		
LSN2*	Polarity	BOOL		
LSP2*	Polarity	BOOL		
DOG2*	Polarity	BOOL		

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	The execution of the motion control function block is complete.	BOOL	● There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
_Pg0	Polarity	BOOL	• When input pins are set to True, and the input terminals are OFF, there are transitions in these output pins' signals from low to high. • When input pins are set to False, and the input terminals are ON, there are transitions in these output pins' signals from low to high.	• When input pins are set to True, and the input terminals are ON, there are transitions in these output pins' signals from high to low. • When input pins are set to False, and the input terminals are OFF, there are transitions in these output pins' signals from high to low. • There are transitions in these output pins' signals from high to low when there is a transition in the Enable input pin's signal from high to low.
_MPGB0	Polarity	BOOL		
_MPGA0	Polarity	BOOL		
_LSN0	Polarity	BOOL		
_LSP0	Polarity	BOOL		
_DOG0	Polarity	BOOL		
_STOP0	Polarity	BOOL		
_START0	Polarity	BOOL		
_PG1	Polarity	BOOL		
_MPGB1	Polarity	BOOL		
_MPGA1	Polarity	BOOL		
_LSN1	Polarity	BOOL		
_LSP1	Polarity	BOOL		
_DOG1	Polarity	BOOL		
_STOP1	Polarity	BOOL		
_START1	Polarity	BOOL		
_PG2*	Polarity	BOOL		
_LSN2*	Polarity	BOOL		
_LSP2*	Polarity	BOOL		
_DOG2*	Polarity	BOOL		

*: The input pin is only for DVP20PM00M.

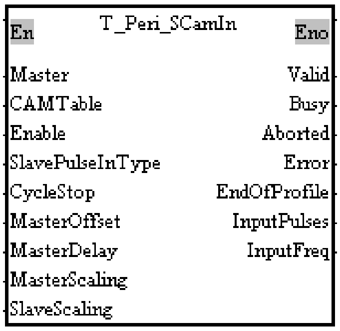
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported

   The motion control function block T_InputPolarity supports DVP20PM00D and DVP20PM00M.

5.10.19 Uniaxial Cyclic Electronic Cam Motion



1. Motion control function block

   The motion control function block T_Peri_SCamIn is used to start electronic cam motion. The value of the Master input pin indicates a master axis. The first axis is used as a slave axis. The motion of the slave axis specified follows the motion of the master axis specified. The value of the MasterOffset input pin indicates the starting angle of the master axis specified. The value of the MasterDelay input pin indicates the number of pulses the master axis specified sends before electronic cam motion is started. If the CycleStop input pin is set to True when the Enable input pin is reset, cam motion will not stop until a cycle is complete.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Master	Master axis number	WORD	K1: A0± and B0± K2: FP± for the Y-axis (No external wiring is needed.) K3: FP± and RP± for the Y-axis (No external wiring is needed.)	The value of the Master input pin is valid when there is a transition in the Enable input pin's signal from low to high.
CAMTable	Cam chart	WORD	K0~K2	The value of the CAMTable input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	Enabling electronic cam motion	BOOL	True/False	-
SlavePulseInType	Pulse type	WORD	K0: mcUD K1: mcPD K2: mcAB K3: mc4AB	The value of the SlavePulseInType input pin is valid when there is a transition in the Enable input pin's signal from low to high.
CycleStop	Stopping a whole cycle	BOOL	True/False	The value of the CycleStop input pin is valid when there is a transition in the Enable input pin's signal from high to low.
MasterOffset	Starting angle of the axis specified (Unit: Pulse)	DWORD	K0~K2,147,483,647	The value of the MasterOffset input pin is valid when there is a transition in the Enable input pin's signal from low to high.
MasterDelay	Number of pulses the master axis specified sends before electronic cam motion is started. (Unit: Pulse)	DWORD	K0~K2,147,483,647	The value of the MasterDelay input pin is valid when there is a transition in the Enable input pin's signal from low to high.
MasterScaling	Ratio which is used to reduce/enlarge the number of pulses sent by the master axis specified	FLOAT	0.~650.00 (two decimal places)	The value of the MasterScaling input pin is valid when there is a transition in the Enable input pin's signal from low to high.
SlaveScaling	Ratio which is used to reduce/enlarge the number of pulses sent by the slave axis specified	FLOAT	0.~650.00 (two decimal places)	The value of the SlaveScaling input pin is valid when there is a transition in the Enable input pin's signal from low to high.

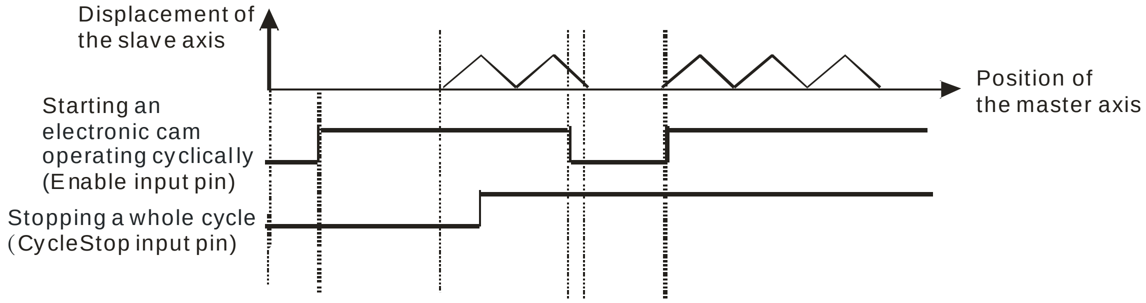
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	• There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	• There is a transition in the Valid output pin's signal from high to low when motion stops. • There is transition in the Valid output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Valid output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. • If the Enable input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The cam chart created is incorrect. • The axis specified is in motion before the motion control function block is executed.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
EndOfProfile	A cycle is complete.	BOOL	• A cam cycle is complete.	• There is a transition in the EndOfProfile output pin's signal from high to low when the next scan cycle begins.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
InputPulses	Number of input pulses	DWORD	K-2,147,483,648~ K2,147,483,647	When the motion control function block is executed, the value of the InputPulses output pin is updated repeatedly.

Value output pin				
Name	Function	Data type	Output range	Update
InputFreq	Frequency of input pulses	DWORD	K0~K2,147,483,647	When the motion control function block is executed, the value of the InputFreq output pin is updated repeatedly.

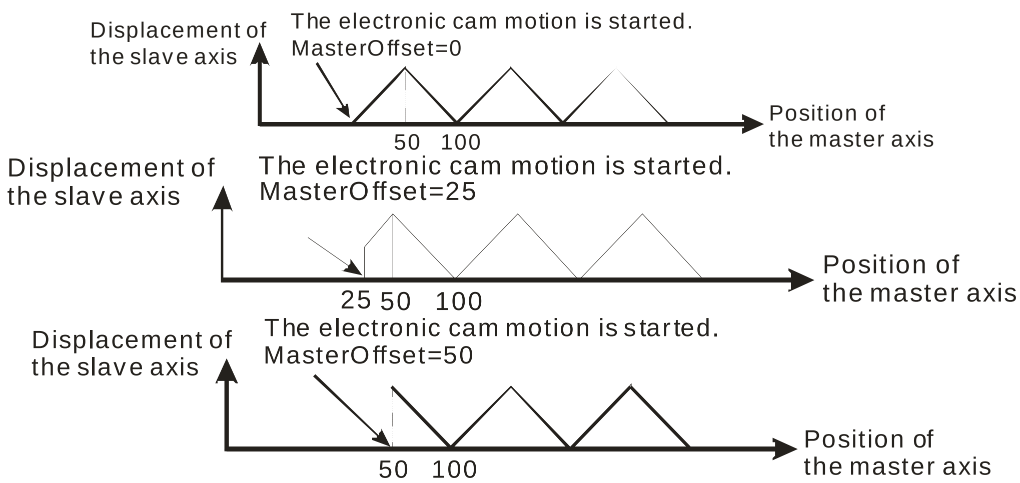
CycleStop

If the Enable input pin is set to True, the slave axis specified will mesh with the master axis specified. If the Enable input pin is set to False after the CycleStop input pin is set to True, cam motion stops when a cycle is complete.



MasterOffset

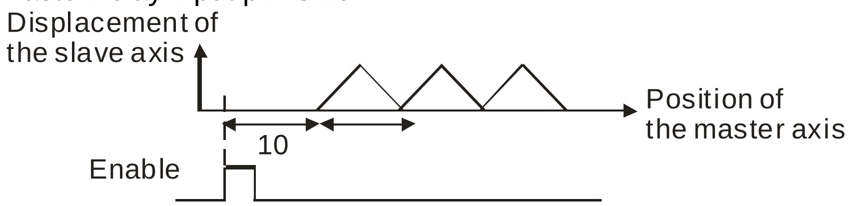
Users can set the starting angle of the master axis specified by means of the MasterOffset input pin. If the value of the MasterOffset input pin is 0, the starting angle of the master axis specified is the initial position of the master axis specified. If the value of the MasterOffset input pin is 25, the starting angle of the master axis specified is the position to which the master axis specified will move after the master axis sends 25 pulses. If the value of the MasterOffset input pin is 50, the starting angle of the master axis specified is the position to which the master axis specified will move after the master axis sends 50 pulses.



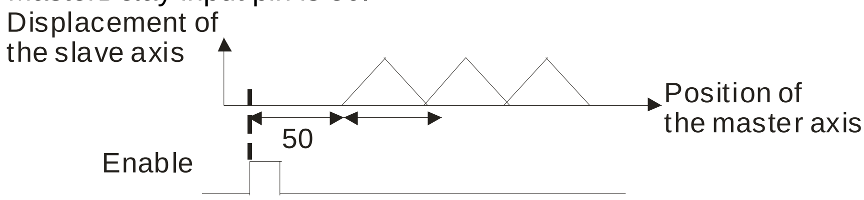
MasterDelay

A system will execute cyclic electronic cam motion after it receives a signal which enables the cyclic electronic cam motion. If users do not want to start an electronic cam immediately, and want to delay the sending of pulses by the slave axis of the electronic cam, they can set the number of pulses the master axis specified sends before the electronic cam is started by means of the MasterDelay input pin. After a system receives a signal which enables electronic cam motion, the electronic cam motion will not be started until the number of pulses the master axis specified sends is equal to the value of the MasterDelay input pin.

The value of the MasterDelay input pin is 10.

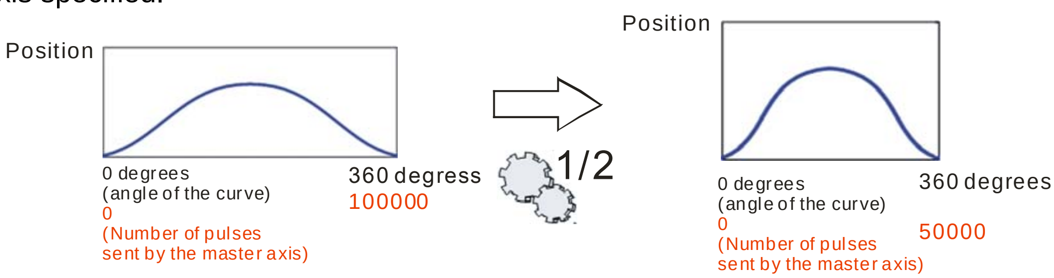


The value of the MasterDelay input pin is 50.



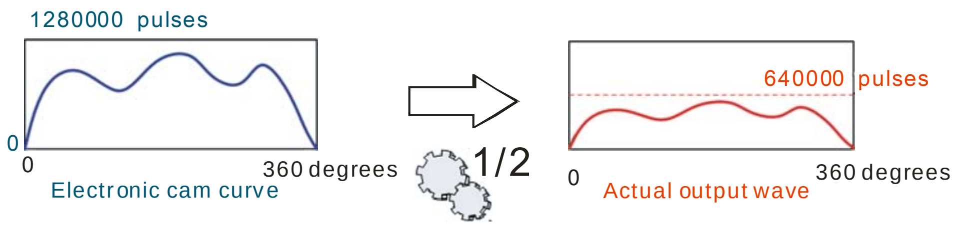
MasterScaling

Users can set a ratio which is used to reduce/enlarge the number of pulses sent by the master axis specified by means of the MasterScaling input pin. In the figure below, the value of the MasterScaling input pin is 1/2. After the slave axis specified meshes with the master axis specified, the number of pulses sent by the master axis specified will be 1/2 times the original number of pulses sent by the master axis specified.



SlaveScaling

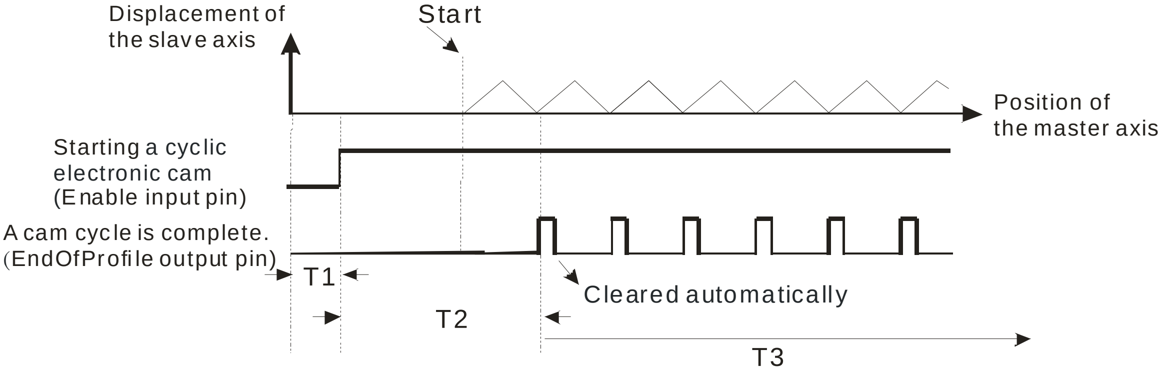
Users can set a ratio which is used to reduce/enlarge the number of pulses sent by the slave axis specified by means of the SlaveScaling input pin. In the figure below, the value of the SlaveScaling input pin is 1/2. After the slave axis specified meshes with the master axis specified, the number of pulses sent by the slave axis specified will be 1/2 times the original number of pulses sent by the slave axis specified.



EndofProfile

The steps of starting a cyclic electronic cam are as follows.

- At the time T1, the Enable input pin is set to True (a uniaxial cyclic electronic cam is started).
- After the time T2 elapses, the EndOfProfile output pin will be set to True. The value of the EndOfProfile output pin will be cleared after one scan cycle.
- During the time T3, the EndOfProfile output pin is set to True when an electronic cam cycle is complete, and the value of the EndOfProfile output pin will be cleared after one scan cycle.



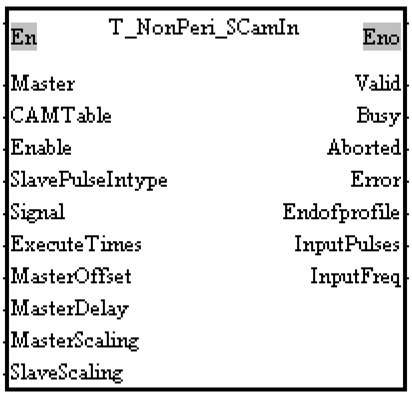
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported

The motion control function block T_Peri_SCamIn supports DVP20PM00D and DVP20PM00M.

5.10.20 Uniaxial Noncyclic Electronic Cam Motion



1. Motion control function block

The motion control function block T_NonPeri_SCamIn is used to start electronic cam motion. The value of the Master input pin indicates a master axis. The first axis is used as a slave axis. The motion of the slave axis specified follows the motion of the master axis specified. The value of the MasterOffset input pin indicates the starting angle of the master axis specified. The value of the MasterDelay input pin indicates the number of pulses the master axis specified sends before electronic cam motion is started.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Master	Master axis number	WORD	K1: A0± and B0± K2: FP± for the Y-axis (No external wiring is needed.) K3: FP± and RP± for the Y-axis (No external wiring is needed.)	The value of the Master input pin is valid when there is a transition in the Enable input pin's signal from low to high.

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
CAMTable	Cam chart	WORD	K0~K2	The value of the CAMTable input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	Enabling electronic cam motion	BOOL	True/False	-
SlavePulseInType	Pulse type	WORD	K0: mcUD K1: mcPD K2: mcAB K3: mc4AB	The value of the SlavePulseInType input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Signal	Command used to start motion	WORD	K1: Start0 K2: PG0	The value of the Signal input pin is valid when there is a transition in the Enable input pin's signal from low to high.
ExecuteTimes	Number of times noncyclic electronic cam motion is executed	WORD	K1~K32767 K32768: Same as uniaxial cyclic electronic cam motion	The value of the ExecuteTimes input pin is valid when there is a transition in the Enable input pin's signal from low to high.
MasterOffset	Starting angle of the axis specified (Unit: Pulse)	DWORD	K0~K2,147,483,647	The value of the MasterOffset input pin is valid when there is a transition in the Enable input pin's signal from low to high.
MasterDelay	Number of pulses the master axis specified sends before electronic cam motion is started. (Unit: Pulse)	DWORD	K0~K2,147,483,647	The value of the MasterDelay input pin is valid when there is a transition in the Enable input pin's signal from low to high.
MasterScaling	Ratio which is used to reduce/enlarge the number of pulses sent by the master axis specified	FLOAT	0.~650.00 (two decimal places)	The value of the MasterScaling input pin is valid when there is a transition in the Enable input pin's signal from low to high.
SlaveScaling	Ratio which is used to reduce/enlarge the number of pulses sent by the slave axis specified	FLOAT	0.~650.00 (two decimal places)	The value of the SlaveScaling input pin is valid when there is a transition in the Enable input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	• There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	• There is a transition in the Valid output pin's signal from high to low when motion stops. • There is transition in the Valid output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Valid output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. • If the Enable input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The cam chart created is incorrect. • The axis specified is in motion before the motion control function block is executed.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
EndOfProfile	A cycle is complete.	BOOL	• A cam cycle is complete	• There is a transition in the EndofProfile output pin's signal from high to low when the next cycle begins.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
InputPulses	Number of input pulses	DWORD	K-2,147,483,648~K2,147,483,647	When the motion control function block is executed, the value of the InputPulses output pin is updated repeatedly.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
InputFreq	Frequency of input pulses	DWORD	K0~K2,147,483,647	When the motion control function block is executed, the value of the InputFreq output pin is updated repeatedly.

Please refer to the description of the motion control function block T_Peri_SCamIn for more information about the MasterOffset input pin, the MasterDelay input pin, the MasterScaling input pin, the SlaveScaling input pin, and the EndOfProfile output pin.

Signal

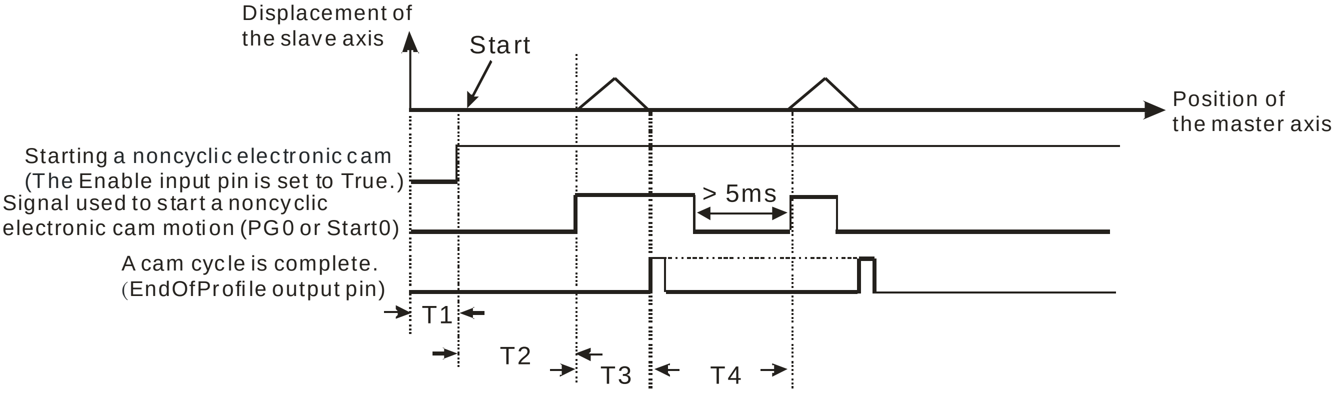
The steps of stopping noncyclic electronic cam motion are as follows.

A. At the time T1, the Enable input pin is set to True (a noncyclic electronic cam is started).

B. After the time T2 elapses, the input terminal used to start noncyclic electronic cam motion will be set to ON. (If the value of the Signal input pin is K1, Start0 will be used to start noncyclic electronic cam motion. If the value of the Signal input pin is K2, PG0 will be used to start noncyclic electronic cam motion.) The value of the EndOfProfile output pin will be cleared after one scan cycle. After the noncyclic electronic cam motion is started, the motion of the slave axis specified will follow the motion of the master axis specified in accordance with the cam chart selected.

C. After the time T3 elapses, the EndOfProfile output pin will be set to True. The value of the EndOfProfile output pin will be cleared automatically after one scan cycle.

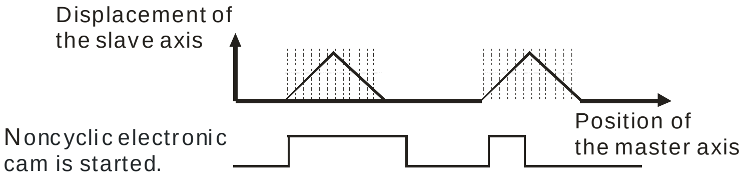
D. After the time T4 elapses, the input terminal used to start the noncyclic electronic cam motion will be set to ON. Users have to notice that more than 5 milliseconds elapses before PG0'signal or Start0's signal goes from low to high.



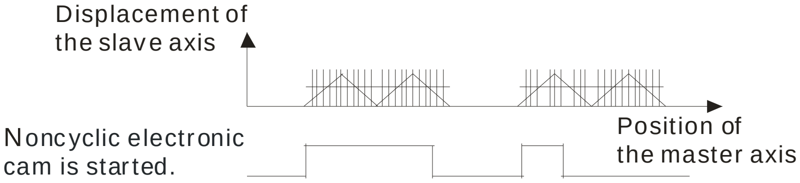
ExecuteTimes

The ExecuteTimes input pin is used to control the number of times noncyclic electronic cam motion is executed. If the value of the ExecuteTimes input pin is greater than H8000, there will be cyclic electronic cam motion. If the value of the ExecuteTimes input is 1, noncyclic electronic cam motion will be executed once. If the value of the ExecuteTimes input pin is 2, noncyclic electronic cam motion will be executed twice.

The value of the ExecuteTimes input pin is 1.



The number of the ExecuteTimes input pin is 2.

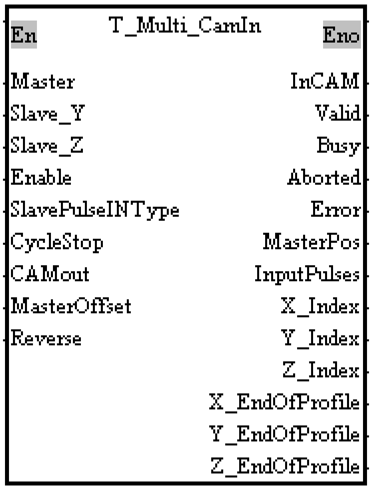


3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported
   The motion control function block T_NonPeri_SCamIn supports DVP20PM00D and DVP20PM00M.

5.10.21 Multiaxial Cyclic Electronic Cam Motion



1. Motion control function block
   The motion control function block T_Multi_CamIn is used to start electronic cam motion. The value of the Master input pin indicates a master axis. The first axis is used as a slave axis. The motion of the slave axis specified follows the motion of the master axis specified. The value of the MasterOffset input pin indicates the starting angle of the master axis specified. The value of the MasterDelay input pin indicates the number of pulses the master axis specified sends before electronic cam motion is started.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Master	Master axis number	WORD	K1: A0± and B0± K2: FP± for the Y-axis (No external wiring is needed.) K3: FP± and RP± for the Y-axis (No external wiring is needed.)	The value of the Master input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Slave_Y	Starting the Y-axis	BOOL	True/False	The value of the Slave_Y input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Slave_Z	Starting the Z-axis	BOOL	True/False	The value of the Slave_Z input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	Enabling electronic cam motion	BOOL	True/False	-
SlavePulseInType	Pulse type	WORD	K0: mcUD K1: mcPD K2: mcAB K3: mc4AB	The value of the SlavePulseInType input pin is valid when there is a transition in the Enable input pin's signal from low to high.

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
CycleStop	Stopping a whole cycle	WORD	K1: Start0 K2: PG0	The value of the CycleStop input pin is valid when there is a transition in the Enable input pin's signal from high to low.
CAMout	Not meshing with the master axis specified	BOOL	True/False	The value of the CAMout input pin is used when the motion control function block is executed.
MasterOffset	Starting angle of the axis specified (Unit: Pulse)	DWORD	K0~K2,147,483,647	The value of the MasterOffset input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Reverse	The slave axis specified rotates reversely as the master axis specified does.	BOOL	True/False	The value of the Reverse input pin is valid when there is a transition in the Enable input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
InCam	The slave axis specified meshes with the master axis specified.	BOOL	● There is a transition in the InCam output pin's signal from low to high when there is a transition in the CamOut input pin's signal from low to high.	● There is a transition in the InCam output pin's signal form high to low when there is a transition in the CamOut input pin's signal from high to high to low.
Valid	An output value is valid.	BOOL	● There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Valid output pin's signal from high to low when motion stops. ● There is transition in the Valid output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. • If the Enable input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The cam chart created is incorrect. • The axis specified is in motion before the motion control function block is executed.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
X_EndOfProfile	A cycle of the X-axis is complete.	BOOL	• A cam cycle is complete.	• There is a transition in the X_EndofProfile output pin's signal from high to low when the next scan cycle begins.
Y_EndOfProfile	A cycle of the Y-axis is complete.	BOOL	• A cam cycle is complete.	• There is a transition in the Y_EndofProfile output pin's signal from high to low when the next scan cycle begins.
Z_EndOfProfile	A cycle of the Z-axis is complete.	BOOL	• A cam cycle is complete.	• There is a transition in the Z_EndofProfile output pin's signal from high to low when the next scan cycle begins.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
X_Index	Index of an X-axis point	WORD	K1~K2047	When the motion control function block is executed, the value of the X_Index output pin is updated repeatedly.
Y_Index	Index of a Y-axis point	WORD	K1~K2047	When the motion control function block is executed, the value of the Y_Index output pin is updated repeatedly.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
Z_Index	Index of a Z-axis point	WORD	K1 ~ K2047	When the motion control function block is executed, the value of the Z_Index output pin is updated repeatedly.
InputPulses	Number of input pulses	DWORD	K-2,147,483,648~ K2,147,483,647	When the motion control function block is executed, the value of the InputPulses output pin is updated repeatedly.
InputFreq	Frequency of input pulses	DWORD	K0~K2,147,483,647	When the motion control function block is executed, the value of the InputFreq output pin is updated repeatedly.

Please refer to the description of the motion control function block T_Peri_SCamIn for more information about the MasterOffset input pin, the CycleStop input pin, and the EndOfProfile output pin.

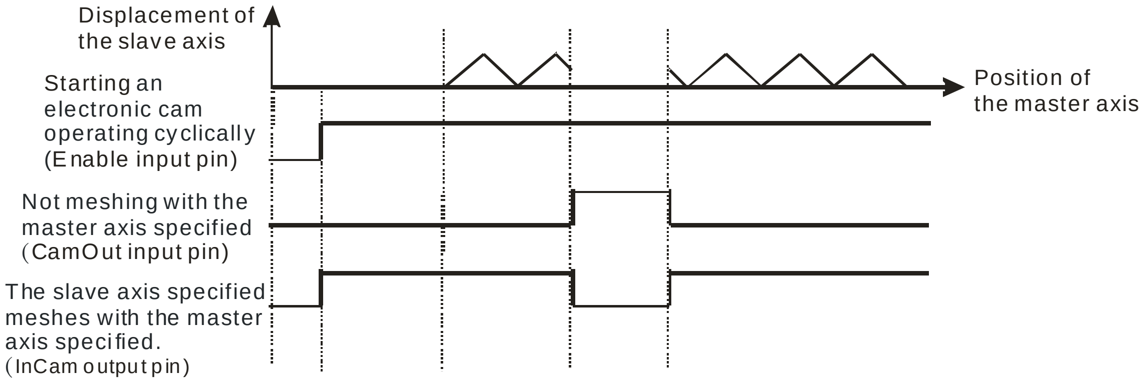
Slave_Y and Slave_Z

If a DVP-20PM series motion controller executes multi-axis electronic cam motion, they can start three axes at most. The cam chart that the X-axis executes is Cam chart 0, the cam chart that the Y-axis executes is cam chart 1, and the cam chart that the Z-axis executes is cam chart 2. The cam chart that an axis executes is fixed.

If the Slave_Y input is set to True, the motion of the Y-axis and the motion of the X-axis will follow the motion of the master axis specified. If the Slave_Z input pin is set to True, the motion of the Z-axis and the motion of the X-axis will follow the motion of the master axis specified. If the Slave_Y input pin and the Slave_Z input pin are set to False, the motion of the X-axis will follow the motion of the master axis specified.

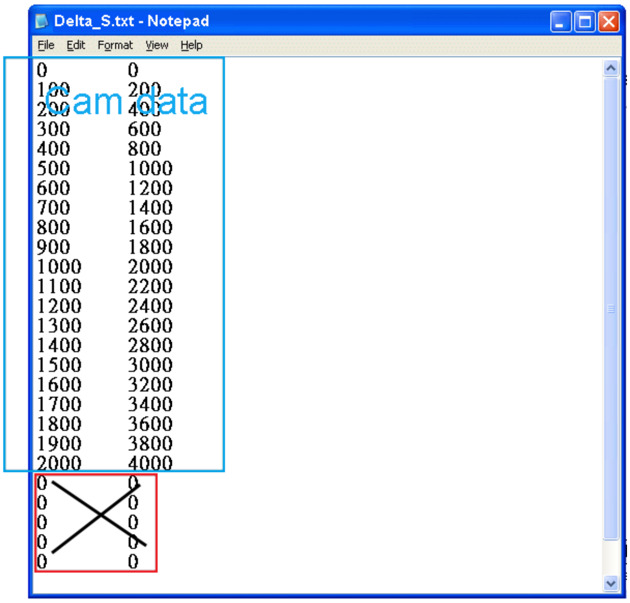
CAMout

If the CAMout input pin is set to True when the Enable input pin is True, the slave axis specified will not mesh with the master axis specified, and the InCam output pin will be set to False. If the CAMout input pin is reset to False, the slave axis specified will mesh with the master axis specified, and the InCam output pin will be set to True.



Reverse

After the Reverse input pin is set to True, the slave axis specified will rotate reversely if the master axis specified rotate reversely. Users have to notice that 0 at the bottom of a cam chart needs to be deleted.



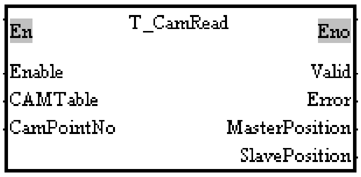
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported

   The motion control function block T_Multi_CamIn supports DVP20PM00D and DVP20PM00M.

5.10.22 Reading a Cam Point



1. Motion control function block

   The motion control function block T_CamRead is used to read a particular point in a cam chart. The value of the CAMTable input pin indicates the cam chart which is read. The value of the CamPoiontNo input pin indicates a cam point number. The value of the MasterPosition output pin indicates the position of the master axis specified, and the value of the SlavePosition output pin indicates the position of the slave axis specified.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Enable	The reading of a cam point is enabled when there is a transition in the Enable input pin's signal from low to high.	BOOL	True/False	-
CAMTable	Cam chart	WORD	K0~K2	The value of the CAMTable input pin is valid when there is a transition in the Enable input pin's signal from low to high.

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
CamPointNo	Cam point number	DWORD	K0~2047	When the motion control function block is executed, the value of the CamPointNo input pin is updated repeatedly.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	● There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. ● There is transition in the Valid output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Time when a value is valid**
MasterPosition	Position of the master axis specified	DWORD	K-2,147,483,647~ K2,147,483,647	When the motion control function block is executed, the value of the MasterPosition output pin is updated repeatedly.
SlavePosition	Position of the slave axis specified	DWORD	K-2,147,483,647~ K2,147,483,647	When the motion control function block is executed, the value of the SlavePosition output pin is updated repeatedly.

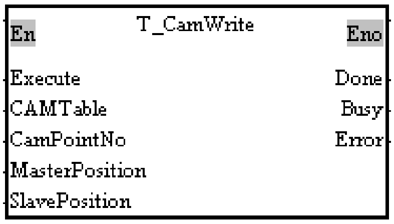
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported

The motion control function block T_CamRead supports DVP20PM00D and DVP20PM00M.

5.10.23 Writing a Cam Point



1. Motion control function block
   The motion control function block T_CamWrite is used to modify a particular point in a cam chart. The value of the CAMTable input pin indicates a cam chart. The value of the CamPoiontNo input pin indicates a cam point number. The value of the MasterPosition indicates the position of the master axis specified, and the value of the SlavePosition indicates the position of the slave axis specified.
2. Input pins/Output pins

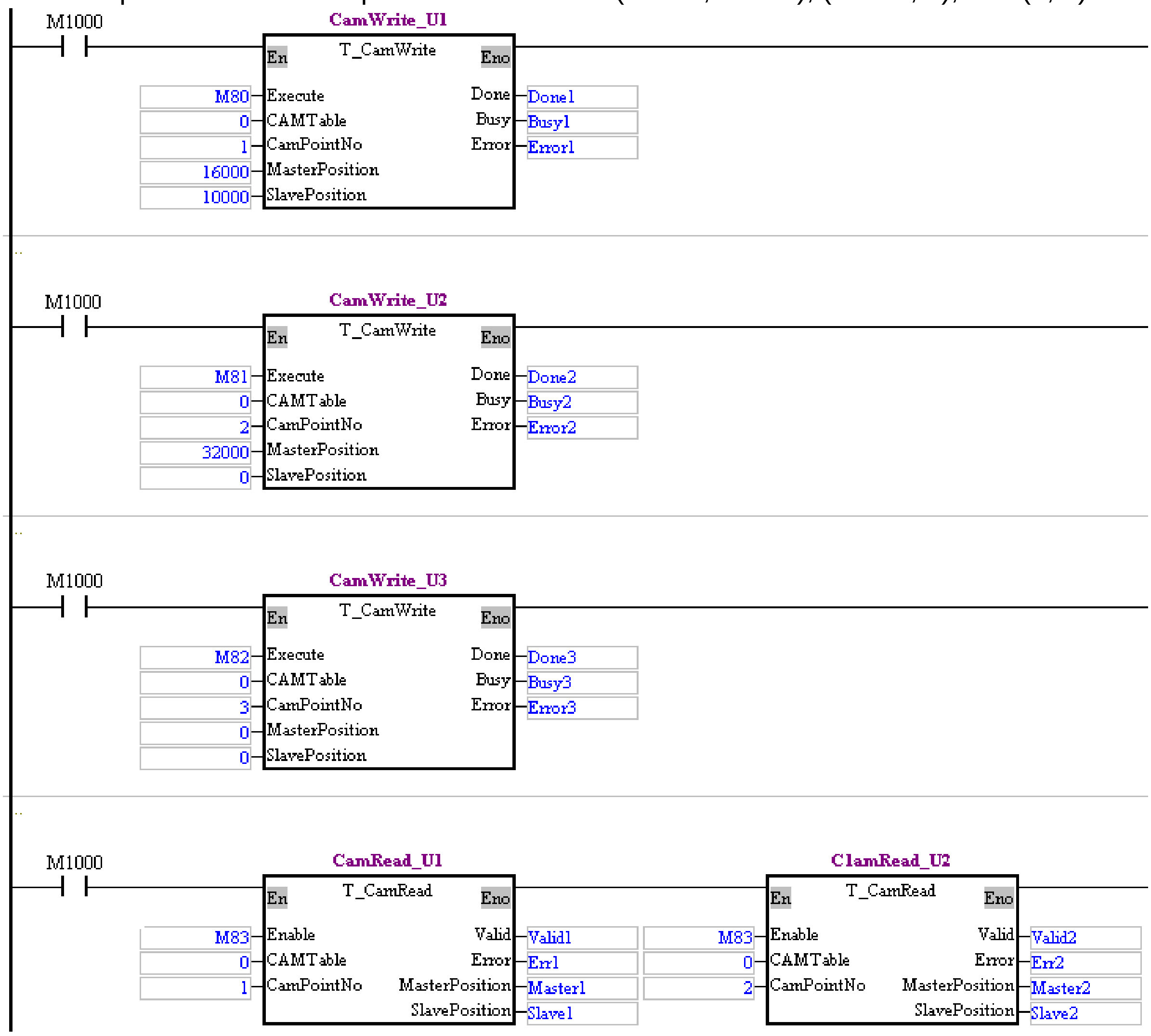
Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Execute	The writing of a cam point is enabled when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
CAMTable	Cam chart	WORD	K0~K2	The value of the CAMTable input pin is valid when there is a transition in the Execute input pin's signal from low to high.
CamPointNo	Cam point number	DWORD	K0~2047	The value of the CamPointNo input pin is valid when there is a transition in the Execute input pin's signal from low to high.
MasterPosition	Position of the master axis specified	DWORD	K-2,147,483,647~ K2,147,483,647	The value of the MasterPosition input pin is valid when there is a transition in the Execute input pin's signal from low to high.
SlavePosition	Position of the slave axis specified	DWORD	K-2,147,483,647~ K2,147,483,647	The value of the SlavePosition input pin is valid when there is a transition in the Execute input pin's signal from low to high.

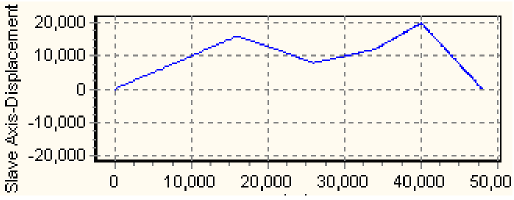
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	● There is a transition in the Done output pin's signal from low to high when the writing of a cam point is complete.	● There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The slave axis specified meshes with the master axis specified before the motion control function block is executed.	• There is transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

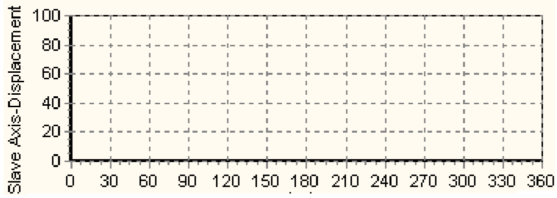
3. Example

Users can modify a particular point in cam chart 0. The slave axis specified has not meshed with the master axis specified. The three pairs of coordinates (16000, 10000), (32000, 0), and (0, 0) are written.

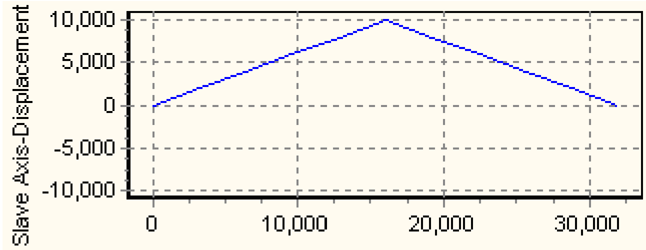




or



(a) Original cam chart



(b) Three pairs of coordinates in a cam chart

- Set M80 to True. The pair of coordinates (16000, 10000) is written into cam point number 1.
- Set M81 to True. The pair of coordinates (32000, 0) is written into cam point number 2.
- Set M82 to True. The pair of coordinates (0, 0) is written into cam point number 3.
- Set M83 to True. Cam point number 1 and cam point number 2 are read. Check whether the values read are the same as the values written into cam point number 1 and cam point number 2.
- After cam chart 0 is modified, the chart displayed will be composed of the coordinates written into cam chart 0.

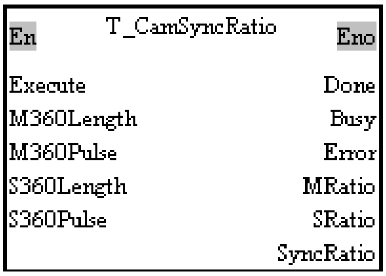
4. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

5. Modules which are supported

The motion control function block T_CamWrite supports DVP20PM00D and DVP20PM00M.

5.10.24 Calculating a Synchronization Ratio



1. Motion control function block

The motion control function block T_CamSyncRatio is used to calculate a synchronization ratio. A synchronization ration is calculated by means of the M360Length input pin, the M360Pulse input pin, the S360Length input pin, and the S360Pulse input pin. (The value of the M360Length input pin indicates physical quantity, and the value of the M360Pulse input pin indicates the number of pulses. The value of the S360Length input pin indicates physical quantity, and the value of the S360Pulse input pin indicates the number of pulses.)

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Execute	When there is a transition in the Execute input pin' signal from low to high, the calculation of a synchronization ratio is enabled.	BOOL	True/False	-
M360Length	Distance for which the master axis specified moves in a cycle	DWORD	K0~K2,147,483,647	The value of the M360Length input pin is valid when there is a transition in the Execute input pin's signal from low to high.
M360Pulse	Number of pulses for which the master axis specified moves in a cycle	DWORD	K0~K2,147,483,647	The value of the M360Pulse input pin is valid when there is a transition in the Execute input pin's signal from low to high.
S360Length	Distance for which the slave axis specified moves in a cycle	DWORD	K0~K2,147,483,647	The value of the S360Length input pin is valid when there is a transition in the Execute input pin's signal from low to high.
S360PULSE	Number of pulses for which the slave axis specified moves in a cycle	DWORD	K0~K2,147,483,647	The value of the S360Pulse input pin is valid when there is a transition in the Execute input pin's signal from low to high.

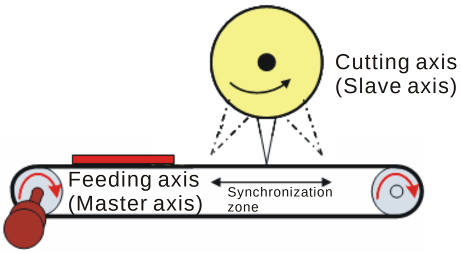
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal when motion is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high.

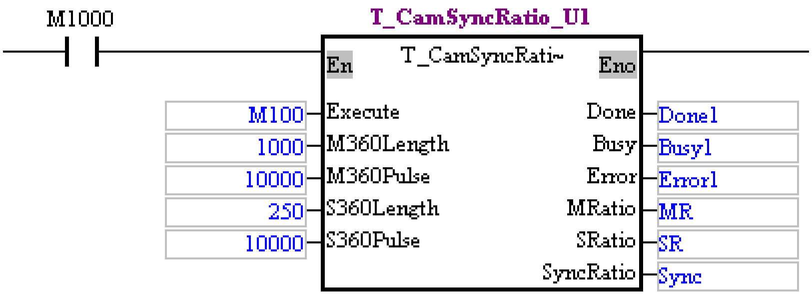
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The slave axis specified meshes with the master axis specified before the motion control function block is executed.	• There is transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Time when a value is valid**
MRatior	Ratio of the distance for which a master axis moves to the number of pulses for which the master axis moves	FLOAT	$1.1755X10^{-38}$~ $3.4028X10^{+38}$	The value of the MRatio output pin is valid when there is a transition in the Done output pin's signal from low to high.
SRatio	Ratio of the distance for which a slave axis moves to the number of pulses for which the slave axis moves	FLOAT	$1.1755X10^{-38}$~ $3.4028X10^{+38}$	The value of the SRatio output pin is valid when there is a transition in the Done output pin's signal from low to high.
SyncRatio	Synchronization ratio	FLOAT	$1.1755X10^{-38}$~ $3.4028X10^{+38}$	The value of the SyncRatio output pin is valid when there is a transition in the Done output pin's signal from low to high.

3. Example

The circumference of the mechanism connected to the slave axis used is 1000 millimeters. The slave axis used sends 10000 pulses when the mechanism connected to the slave axis used rotates once. The circumference of the mechanism connected to the master axis used is 250 millimeters. The master axis used sends 10000 pulses when the mechanism connected to the master axis used rotates once.





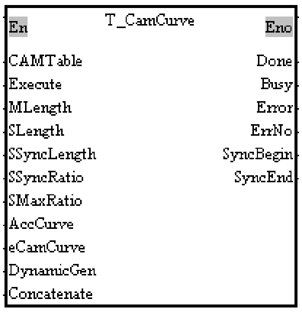
- The value of the M360Length input pin is 1000.
- The value of the M360Pulse input pin is 10000.
- The value of the S360Length input pin is 250.
- The value of the S360Pulse input pin is 10000.
- MR=10 pulses/mm
- SR=4 pulses/mm
- Sync=0.4

4. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

5. Modules which are supported
   The motion control function block T_CamSyncRatio supports DVP20PM00D and DVP20PM00M.

5.10.25 Creating a Cam Curve



1. Motion control function block
   The motion control function block T_CamCurve is used to create a cam curve. It is mainly used to create rotary cut curves and flying shear curves. The value of the CAMTable input pin indicates a cam chart. The value of the MLength input pin, the value of the SLength input pin, the value of the SSyncRatio input pin, and the value of the SMaxRatio input pin indicate the physical quantity needed to generate a cam curve. The value of the AccCurve input pin and the value of the eCamCurve determine a cam curve type.
2. Input pins/Output pins

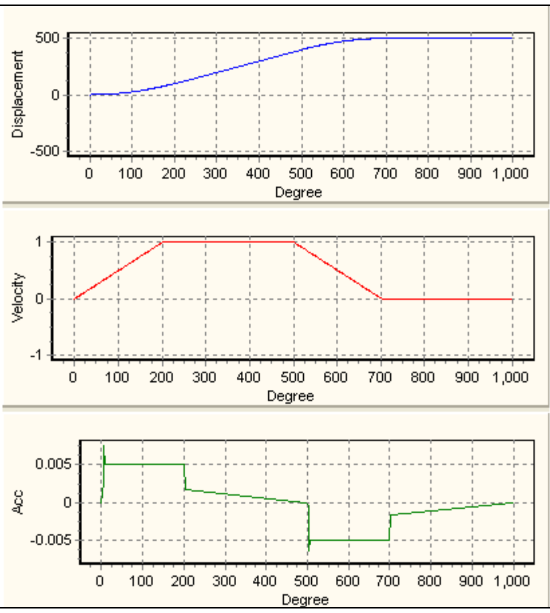
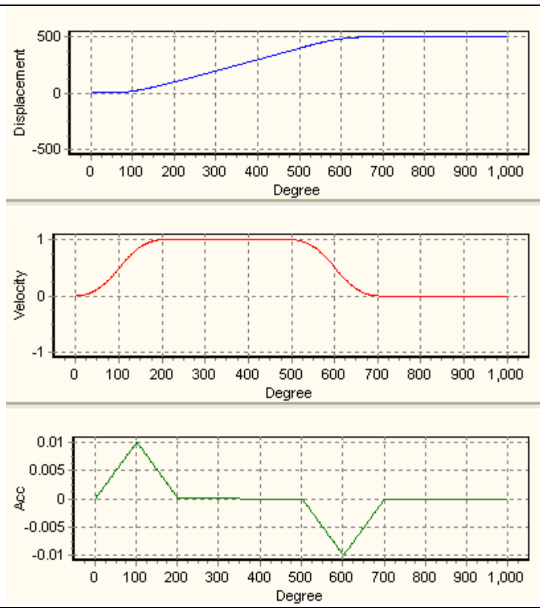
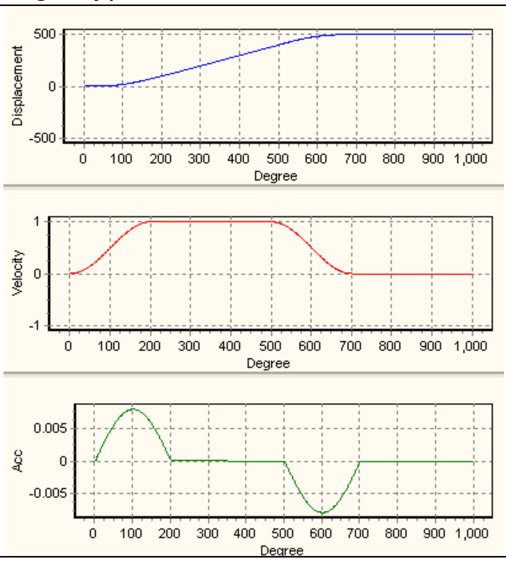
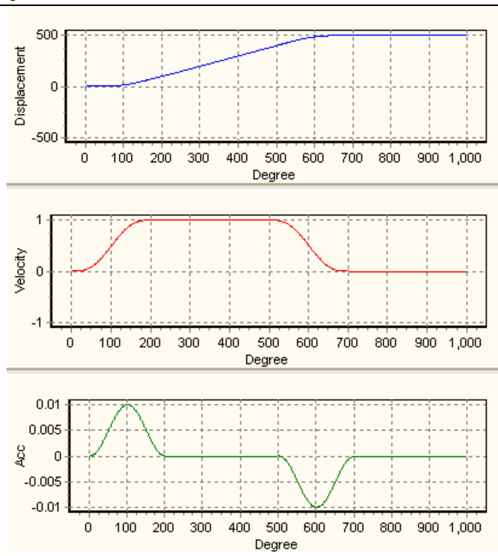
Input pin				
Name	Function	Data type	Setting value	Time when a value is valid
CAMTable	Cam chart	WORD	K0	The value of the CAMTable input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	The creation of a cam curve is enabled when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
MLength	Distance for which the master axis specified moves (Unit: Pulse)	DWORD	K1~K2,147,483,647	The value of the MLength input pin is valid when there is a transition in the Execute input pin's signal from low to high.
SLength	Distance for which the slave axis specified moves (Unit: Pulse)	DWORD	K1~K2,147,483,647	The value of the SLength input pin is valid when there is a transition in the Execute input pin's signal from low to high.
SSyncLength	Distance for which the slave axis specified is synchronized with the master axis specified (Unit: Pulse)	DWORD	K1~K2,147,483,647	The value of the SSyncLength input pin is valid when there is a transition in the Execute input pin's signal from low to high.
SSyncRatio	Synchronization ratio of the speed of the slave axis specified to the speed of the master axis specified	Float	1.1755×10^{-38}~ $3.4028 \times 10^{+38}$	The value of the SSyncRatio input pin is valid when there is a transition in the Execute input pin's signal from low to high.
SMaxRatio	Maximum ratio of the speed of the slave axis to the speed of the master axis specified	Float	1.1755×10^{-38}~ $3.4028 \times 10^{+38}$	-
AccCurve	Acceleration curve	WORD	K0~K3 (*1)	The value of the SSyncRatio input pin is valid when there is a transition in the AccCurve input pin's signal from low to high.
eCamCurve	Cam curve	WORD	K0~K5 (*2)	
DynamicGen	Changing a cam curve dynamically	BOOL	True/False	The value of the DynamicGen input pin is valid when there is a transition in the AccCurve input pin's signal from low to high.
Concatenate	Concatenation	BOOL	True/False	The value of the Concatenate input pin is valid when there is a transition in the AccCurve input pin's signal from low to high.

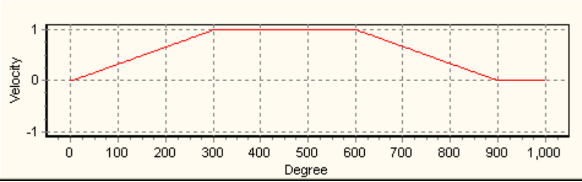
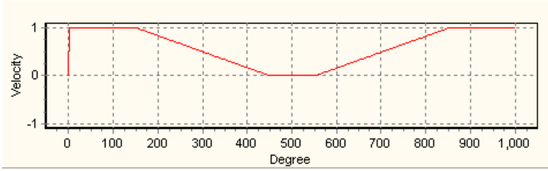
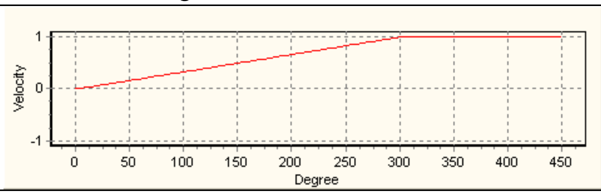
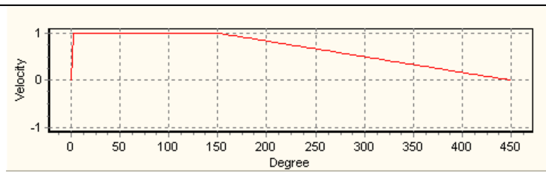
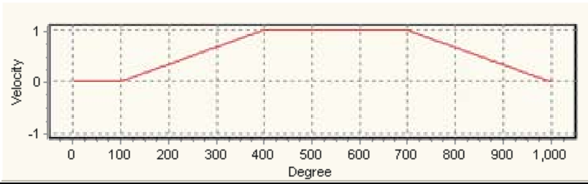
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal when motion is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. • If the Execute input pin is set to False when the execution of the motion control function block is complete, the Done output pin will be set to False in the next cycle.
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Time when a value is valid**
ErrNo	Error code	WORD	K0: The creation of a cam curve is complete. K1: The conditions set can not be used to create a cam curve.	When the motion control function block is executed, the value of the ErrNo output pin is updated repeatedly.
SyncBegin	Starting point of synchronization	DWORD	K0~K2,147,483,647	When the motion control function block is executed, the value of the SyncBegin output pin is updated repeatedly.
SyncEnd	Terminal point of synchronization	DWORD	K0~K2,147,483,647	When the motion control function block is executed, the value of the SyncEnd output pin is updated repeatedly.

*1: Value of the AccCurve input pin

<table>
<tr><td>K0: Uniform curve</td><td>K1: Uniform acceleration curve</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td>K2: SingleHypot curve</td><td>K3: Cycloid</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

*2: Value of the eCamCurve input pin

K0: leftCAM	K1: midCAMall
	
K2: midCAMbegin	K3: midCAMend
	
K5: right CAM	
	

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
ErrorNo=K1: The conditions set can not be used to create a cam curve.	● Check whether the values of the input pin are incorrect. ● Redesign a cam curve.

SSyncRatio

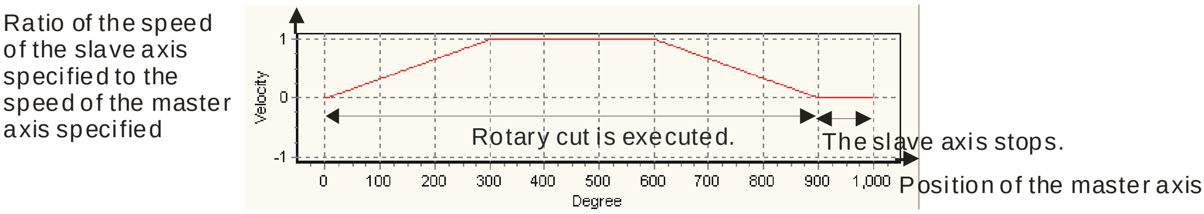
The SSyncRation input pin is the same as the CamSyncRatio output pin in the motion control function block T_CamSyncRatio. Users can write the value of the CamSyncRatio output pin in the motion control function block T_CamSyncRatio into the SSyncRation input pin.

SMaxRatio

The SMaxRation input pin is used to set the maximum ratio of the speed of the slave axis specified to the speed of the master axis specified. If there is no special requirement, the maximum ratio of the speed of the slave axis specified to the speed of the master axis specified will be 10.

eCamCurve

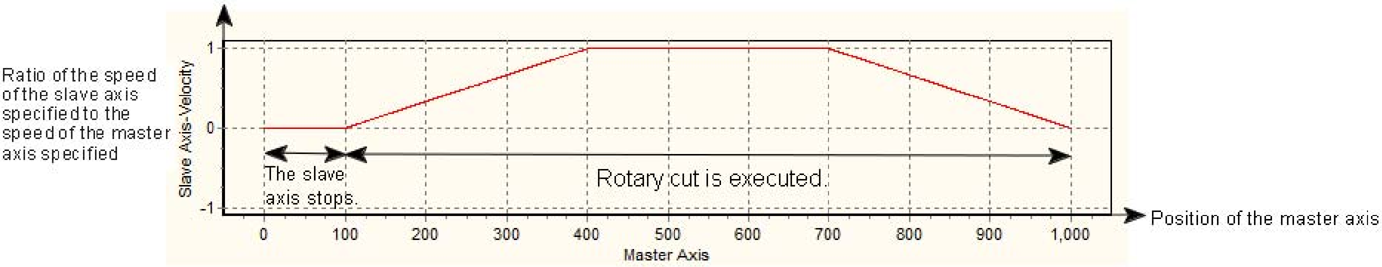
a. LeftCAM



After the slave axis specified is started, it will execute rotary cut. After the rotary cut is complete, the slave axis will stop for a while, and then continue executing the rotary cut.

CLR0 will be ON after the slave axis moves to the synchronization zone designed. After the slave axis leaves the synchronization zone designed, CLR0 will be OFF. When CLR0 is ON, its external LED indicator will not be ON. If users want to check whether CLR0 is ON, they can connect CLR0 to any general input terminal.

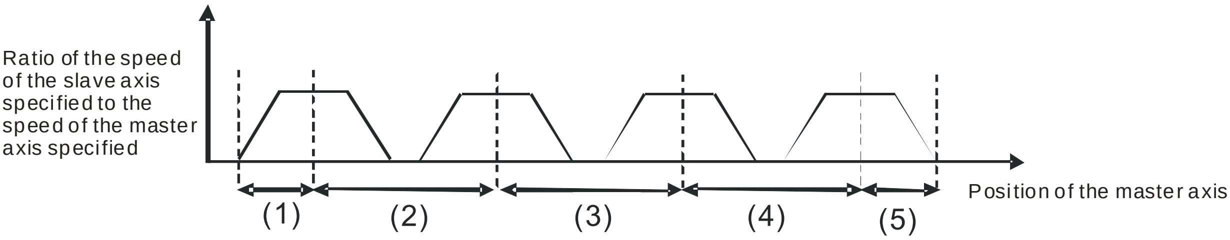
b. RightCAM



After the slave axis specified is started, it will stop for a while, and then execute rotary cut. After the rotary cut is complete, the slave axis will stop for a while, and then continue executing the rotary cut.

CLR0 will be ON after the slave axis moves to the synchronization zone designed. After the slave axis leaves the synchronization zone designed, CLR0 will be OFF. When CLR0 is ON, its external LED indicator will not be ON. If users want to check whether CLR0 is ON, they can connect CLR0 to any general input terminal.

c. midCAMbegin, midCAMall, and midCAMend: The three types of curves need to be used together.



The slave axis specified executes rotary cut four times, and then stops. There are five phases.

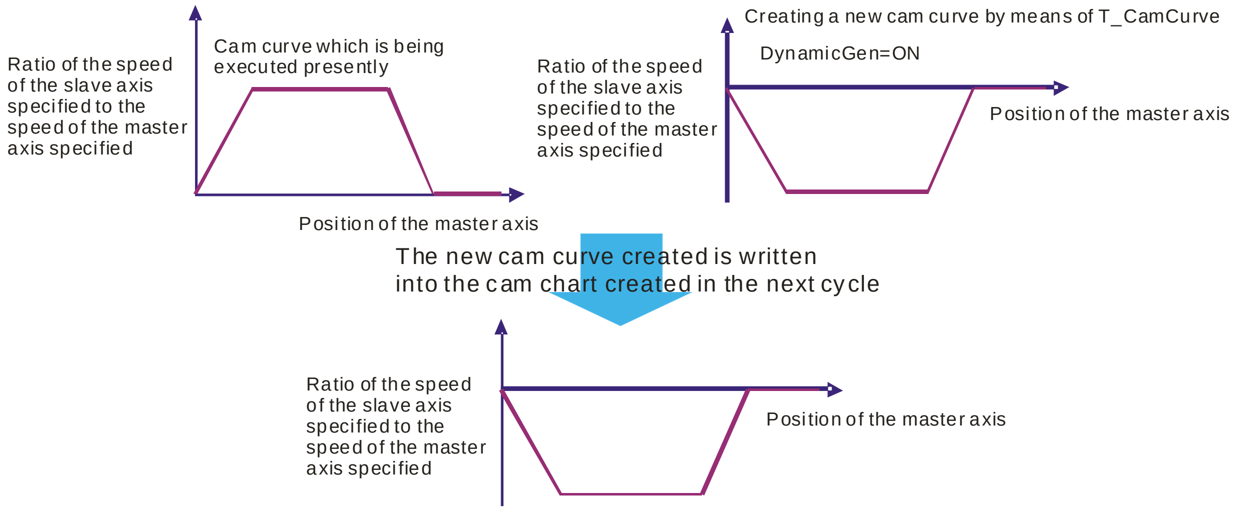
- Phase 1: The slave axis specified is at a standstill, and then begins to execute rotary cut until the first object is cut. The curve created is midCAMbegin.
- Phase 2: After the cutter used cuts the first object, it returns to its preparatory position, and then accelerates until it moves to its cutting position. The curve created is midCAMall.
- Phase 3 and phase 4: The motion in phase 2 is repeated.
- Phase 5: After an object is cut, the cutter used returns to its preparatory position. The curve created is midCAMend.

The three types of curves need to be used together, and the DynamicGen input pin needs to be set to True.

If users create a rotary cut curve in this way, CLR0 will not be ON after the slave axis specified moves to the synchronization zone designed.

DynamicGen

The DynamicGen input pin is used to modify the cam curve created after the slave axis meshes with the master axis specified. Users have to set the DynamicGen input pin to True before the Execute input is set to True. After the Execute input pin is set to True, the new cam curve created will be written into the cam chart selected in the next cycle. In the cycle following the next cycle, the slave axis specified will execute the new cam curve created.

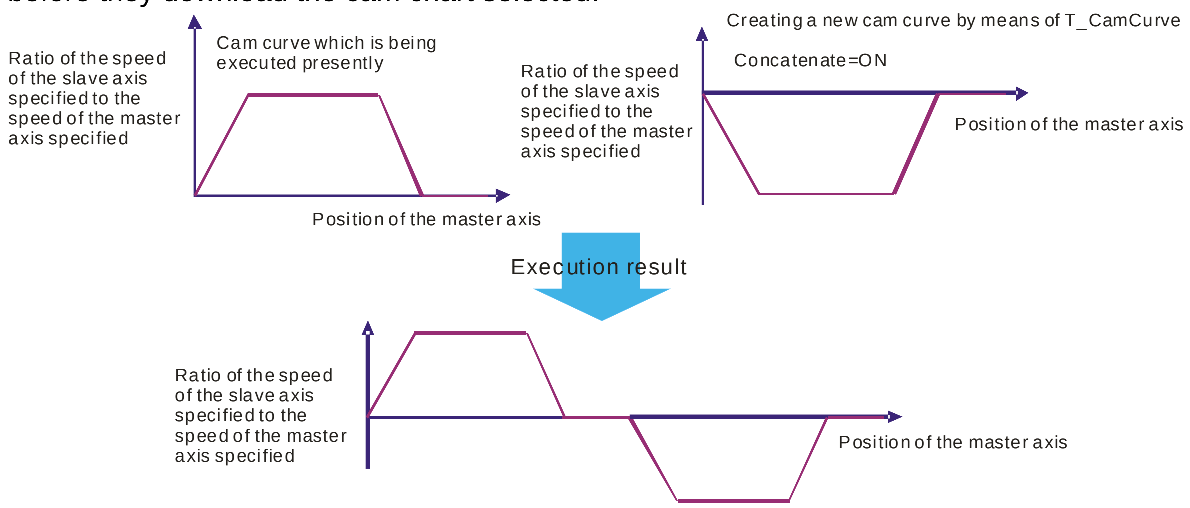


Before the slave axis specified meshes with the master axis specified, the old cam curve in the cam chart selected will be changed to the new cam curve created whether the DynamicGen input is ON or not.

Concatenate

If users want to connect the new cam curve created to the old cam curve in the cam chart selected, they have to set the Concatenate input pin to True before the Execute input pin is set to True.

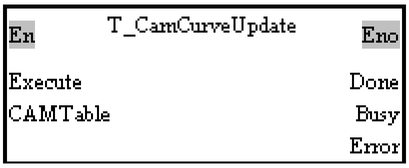
If the resolution of the original cam curve in the cam chart selected is 300, the resolution of the new cam curve must be at least 300. After the new cam curve is connected to the original cam curve, the resolution gotten will be 600. As a result, users have to set the resolution of the cam chart selected to 600 before they download the cam chart selected.



4. Modules which are supported

   The motion control function block T_CamCurve supports DVP20PM00D and DVP20PM00M.

5.10.26 Updating a Cam Curve



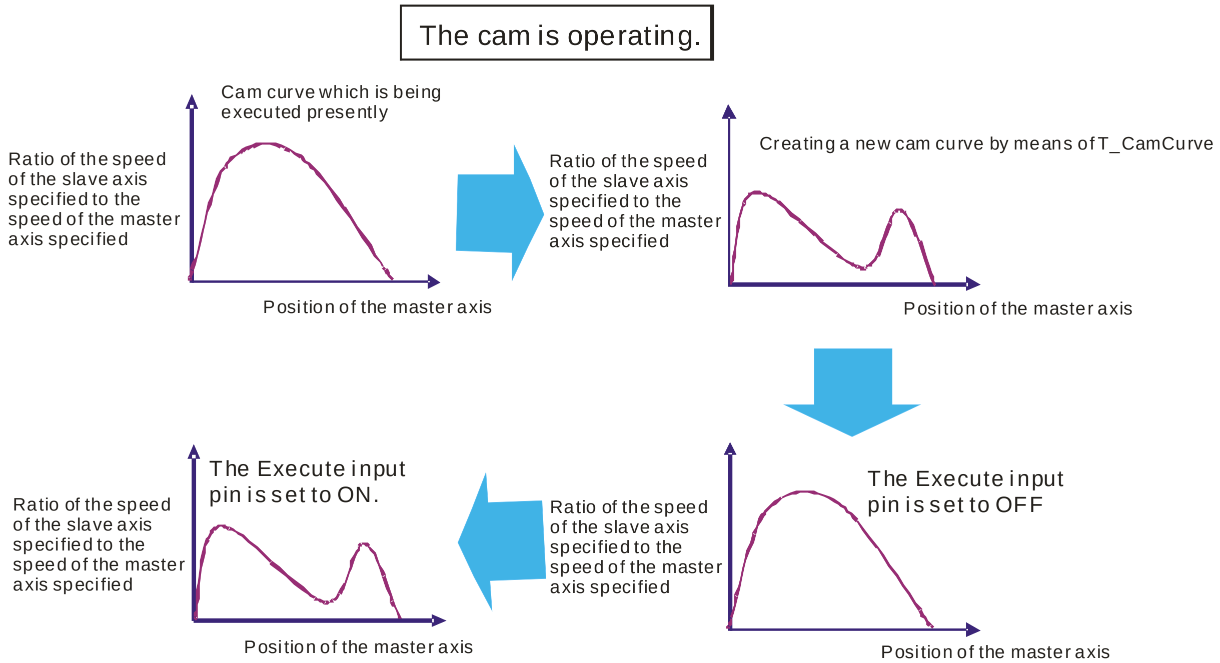
1. Motion control function block
   The motion control function block T_CamCurveUpdate is used to update a cam chart so that the cam curve in the next can cycle is the cam curve created by means of the motion control function block T_CamCurve. The value of the CAMTable input pin indicates a cam chart.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Execute	When there is a transition in the Execute input pin's signal from low to high, the update of a cam curve is enabled.	BOOL	True/False	
CAMTable	Cam chart	WORD	K0~K2	The value of the CAMTable input pin is valid when there is a transition in the Enable input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The update of a cam curve is complete.	BOOL	● There is a transition in the Done output pin's signal when motion is complete.	● There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low. ● If the Execute input pin is set to False when the execution of the motion control function block is complete, the Done output pin will be set to False in the next cycle.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect.	● There is transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

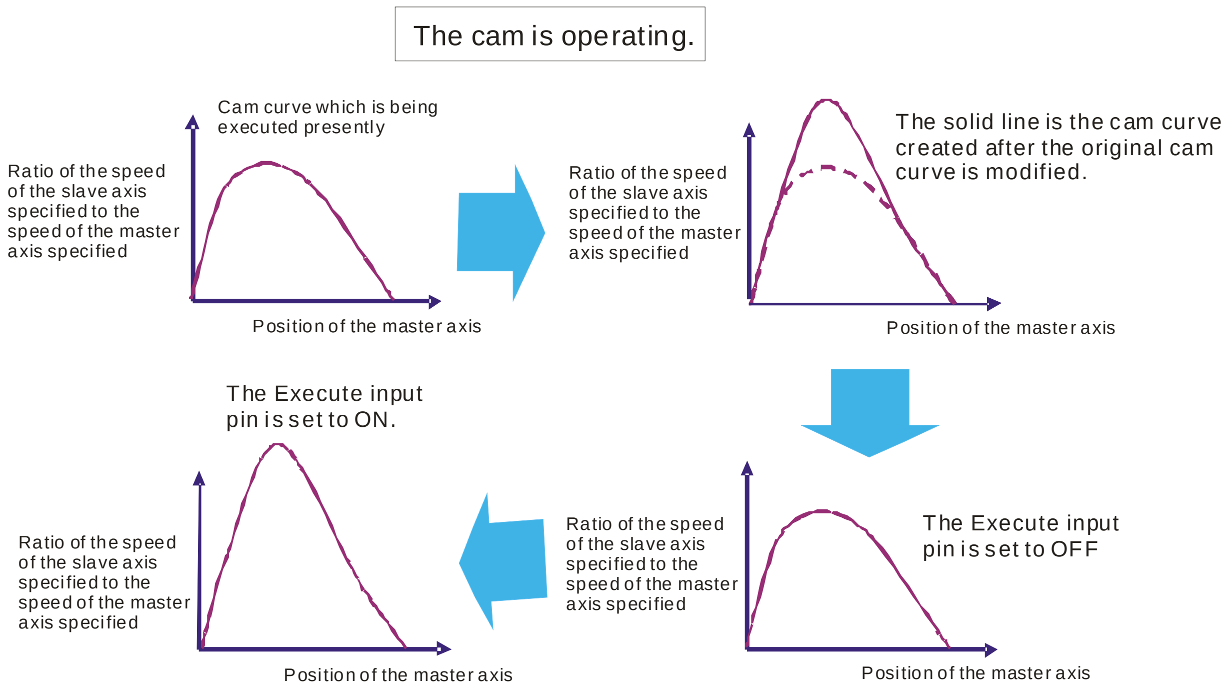
Using T_CamCurveUpdate and T_CamCurve

If users create a new cam curve by means of the motion control function block T_CamCurve when the slave axis specified meshes with the master axis specified, and want to update the cam chart selected by means of a condition instead of updating the cam chart selected immediately, they can use the motion control function block T_CamCurveUpdate.



Using T_CamCurveUpdate and T_CamWrite

If users use the motion control function block T_CamWrite to modify a point in the cam chart selected when the slave axis specified meshes with the master axis specified, the cam chart selected will not change after the point is modified. After the users update the cam chart selected by means of the motion control function block T_CamCurveUpdate, the cam chart created after the motion control function block T_CamWrite is used will be executed.



3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported

The motion control function block T_CamCurveUpdate supports DVP20PM00D and DVP20PM00M.

5.10.27 Rotary Cut

1. Motion control function block

The motion control function block T_RotaryCut is used to create a rotary cut curve, and execute electronic cam motion. Users can modify the length of a cutting by means of the MaterialSize input pin. After the length of a cutting is modified, the new cam curve created will be executed in the second or the third cam cycle. The motion control function block is only applicable to thin materials.

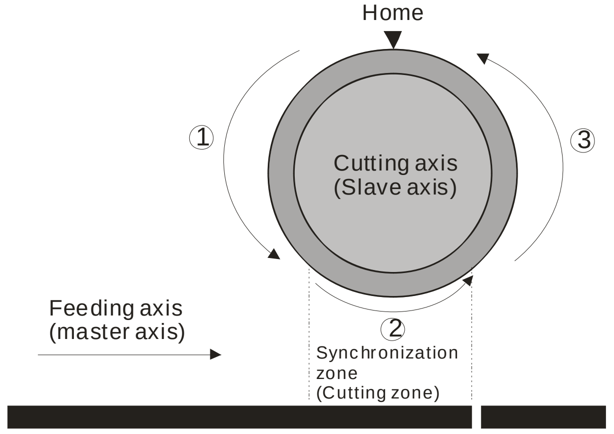
Concept

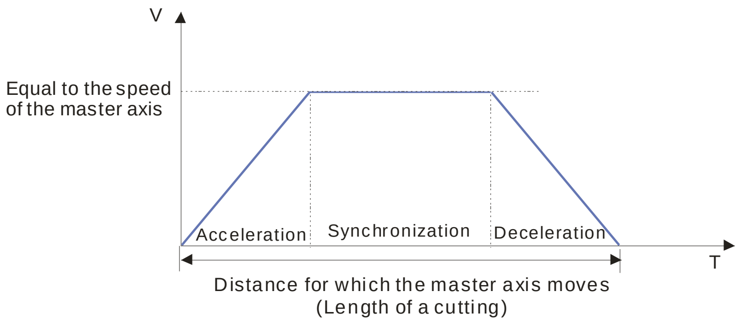
In the application of rotary cut, the cutting roller of a rotary cutter rotates in a direction. A material is cut when the blade of the **rotary** cutter comes into contact with the material. The feeding roller of the rotary cutter continuously feeds materials at a uniform speed. The relation between rotary cut and the output generated is shown below.

(1) At first, the slave axis accelerates until it moves to the synchronization zone.

(2) The speed of the slave axis is equal to the speed of the master axis in the synchronization zone, and CLR0 is ON.

(3) After the slave axis leaves the synchronization zone, it decelerates until it returns home. A cycle is complete when the slave axis is at home. Users can draw a speed relation chart.





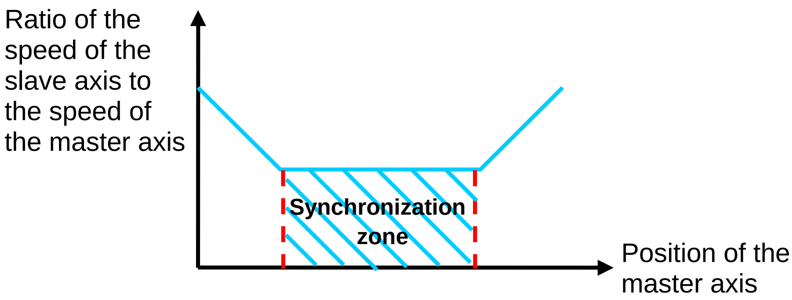
(4) During the processing of cutting materials, synchronization is an important factor. When the bladed of a rotary cutter come into contact with a material, the speed of the blade must be the same as the speed of the material. If the speed of the blade of a rotary cutter is greater than the speed of a material when it comes into contact with the material, the force which pulls the material forward will appear, and the material will be cut smoothly. If the speed of the blade of a rotary cutter is less than

the speed of a material when it comes into contact with the material, the material will be jammed.

(5) The design of a synchronization zone affects the operation of equipment. The bigger the synchronization zone is in a cycle, the less time it takes for the slave axis specified to accelerate/decelerate. If equipment needs to accelerate/decelerate in a short time, there will be a great impact on the electric machinery used and the blade used, and there will be an overcurrent passing through the servo used.

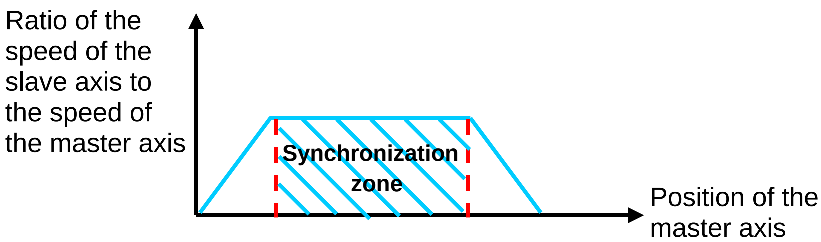
(6) Relation between the length of a cutting and the circumference of a blade: The speed of the master axis is equal to the speed of the slave axis.

◆ Length of a cutting<Length of the blade used: The speed of cutting roller used is the same as the speed of the feeding roller used in the synchronization zone designed. After the cutting roller used leaves the synchronization zone, the cutting roller will accelerate.

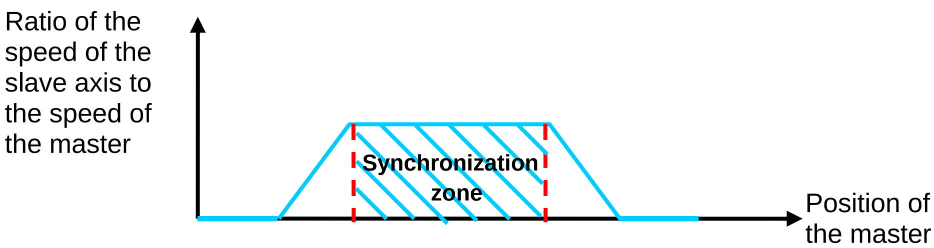


◆ Length of a cutting=Circumference of the blade used: The cutting roller used rotates at a uniform speed.

◆ One time the circumference of the blade used<Length of a cutting<Two times the circumference of the blade used: After the cutting roller used completes cutting in the synchronization zone designed, it will decelerate, and then accelerate until its speed is the same the speed of the feeding roller used.



◆ Length of a cutting>Two times the circumference of the blade used: The length of a cutting is greater than two times the circumference of the blade. (It is a common situation.) After the blade used completes cutting in a cycle, it will decelerate until it stops. After a material of a certain length is fed, the blade used will start cutting again.



2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Master	Master axis number	WORD	K1: A0± and B0± K2: FP± for the Y-axis (No external wiring is needed.) K3: FP± and RP± for the Y-axis (No external wiring is needed.)	The value of the Master input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	Enabling electronic cam motion	BOOL	True/False	-
SlavePulseInType	Pulse type	WORD	K0: mcUD K1: mcPD K2: mcAB K3: mc4AB	The value of the SlavePulseInType input pin is valid when there is a transition in the Enable input pin's signal from low to high.
CycleStop	Stopping a whole cycle	WORD	BOOL	The value of the CycleStop input pin is valid when there is a transition in the Enable input pin's signal from high to low.
MasterLength	Circumference of the master axis specified	WORD	K1~K2,147,483,647	The value of the MasterLength input pin is valid when there is a transition in the Enable input pin's signal from high to low.
MasterPulse	Number of pulses it takes for the master axis specified to rotate once (Unit: Pulse)	DWORD	K1~K2,147,483,647	The value of the MasterPulse input pin is valid when there is a transition in the Enable input pin's signal from high to low.
SlaveLength	Circumference of the slave axis specified	DWORD	K1~K2,147,483,647	The value of the SlaveLength input pin is valid when there is a transition in the Enable input pin's signal from high to low.
SlavePulse	Number of pulses it takes for the slave axis specified to rotate once (Unit: Pulse)	DWORD	K1~K2,147,483,647	The value of the SlavePulse input pin is valid when there is a transition in the Enable input pin's signal from high to low.
SynArea	Synchronization range (Unit: Pulse)	DWORD	K1~K2,147,483,647	The value of the SynArea input pin is valid when there is a transition in the Enable input pin's signal from high to low.
MaterialSize	Length of a cutting	DWORD	K1~K2,147,483,647	When the motion control function block is executed, the value of the MaterialSize input pin is updated repeatedly.
AccCurve	Acceleration curve	WORD	K0: Uniform curve K1: Uniform acceleration curve K2: SingleHypot curve K3: Cycloid	The value of the AccCurve input pin is valid when there is a transition in the Enable input pin's signal from high to low.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	• There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	• There is a transition in the Valid output pin's signal from high to low when motion stops. • There is transition in the Valid output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Valid output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. • If the Enable input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The cam chart created is incorrect. • The axis specified is in motion before the motion control function block is executed.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
EndofProfile	A cycle is complete.	BOOL	• A cam cycle is complete.	• There is a transition in the EndOfProfile output pin's signal from high to low when the next scan cycle begins.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
InputPulses	Number of input pulses	DWORD	K-2,147,483,648~ K2,147,483,647	When the motion control function block is executed, the value of the InputPulses output pin is updated repeatedly.

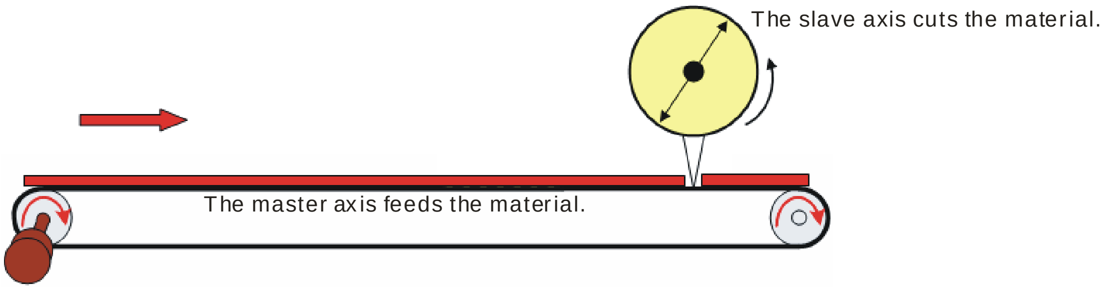
Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
InputFreq	Frequency of input pulses	DWORD	K0~K2,147,483,647	When the motion control function block is executed, the value of the InputFreq output pin is updated repeatedly.

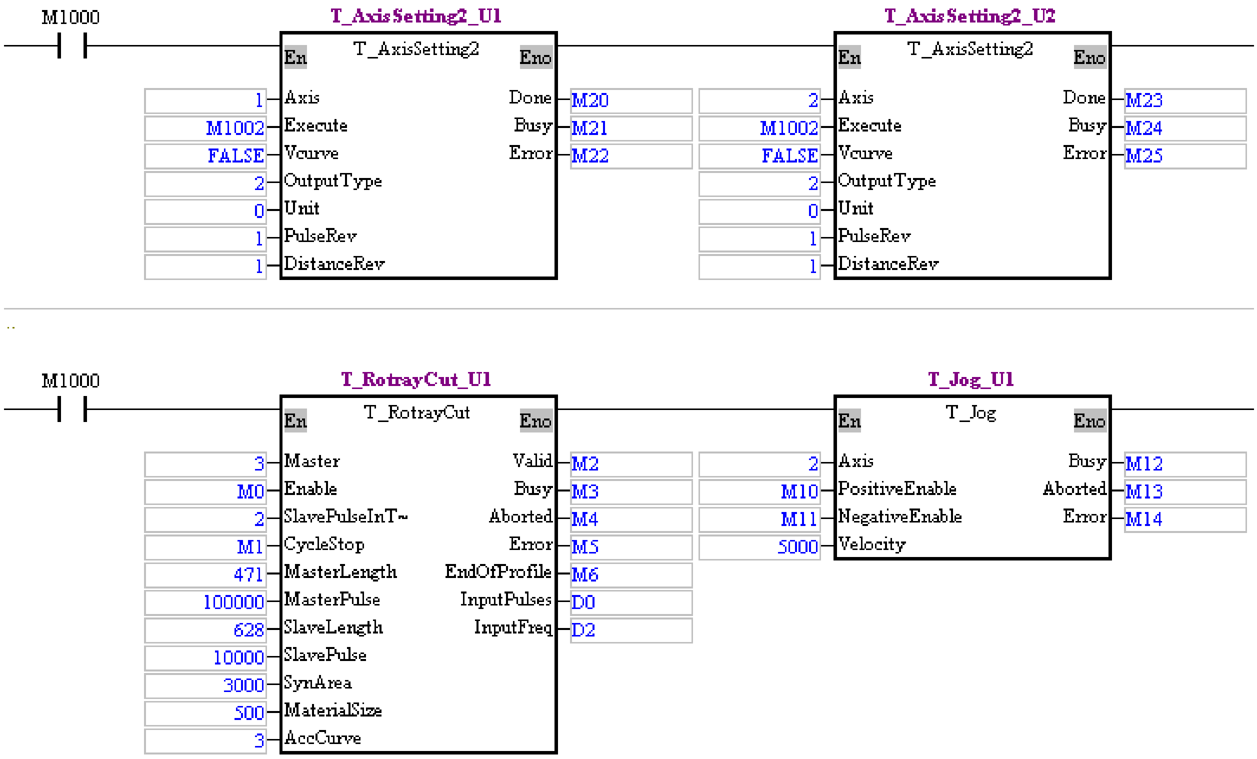
Notes:

- ◆ If the motion control function block T_RotaryCut is used to create cam chart 0 and cam chart 1, the resolution of cam chart 0 must be 300, and the resolution of cam chart 1 must be 300.
- ◆ If the slave axis specified moves to the synchronization zone designed when the motion control function block T_RotaryCut is used, CLR0 will not be ON. The motion control function block T_RotaryCut needs to be used with the motion control function block T_Compare.

3. Example

In this example, the setting of the motion control function block T_RotaryCut is described. The master axis used and the slave axis used are connected to Delta ASDA-A2 series AC servo drives. The first axis in a DVP-20PM series motion controller sends pulses to the slave axis, and the second axis in the DVP-20PM series motion controller sends pulses to the master axis.





- ◆ Descriptions of the parameters in the system used
  - ■ Pulse type: A/B
  - ■ Diameter of the master axis: 150 mm
  - ■ Gear ratio for the mechanism used to decelerate the master axis: 1:10
  - ■ Gear ration for the master axis: 128/4→The number of pulses it takes for the AC servo drive connected to the master axis to rotate once is 10,000. (Take Delta ASDA-A2 series AC servo

drive for instance.)

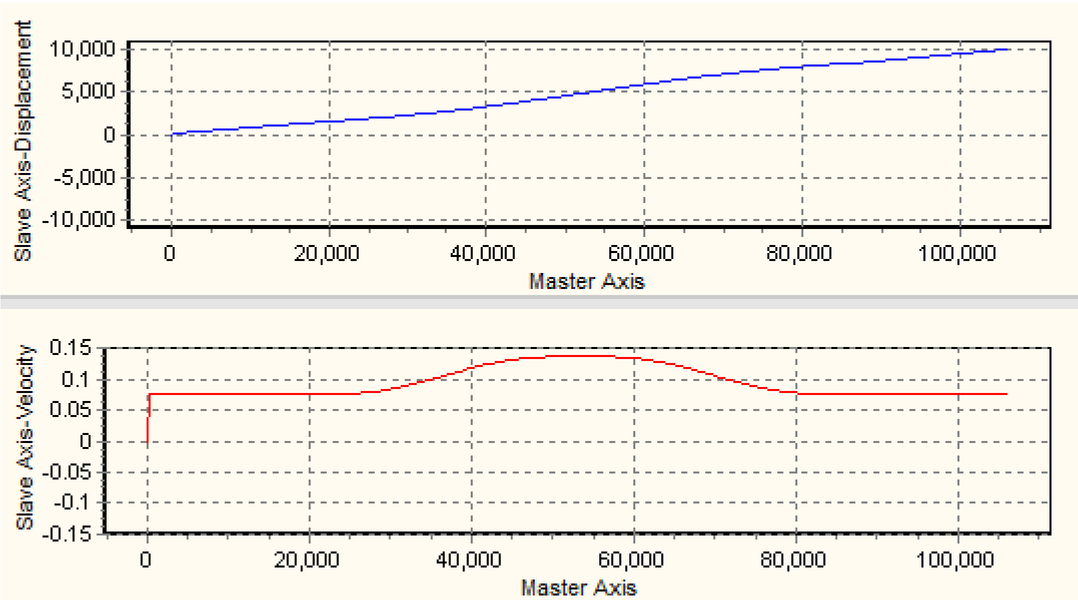
- Diameter of the slave axis: 200 mm
- Gear ratio for the slave axis: 128/4→The number of pulses it takes for the AC servo drive connected to the master axis to rotate once is 10,000. (Take Delta ASDA-A2 series AC servo drive for instance.)
- Length of a cutting: 500 mm

◆ Setting the parameters in the motion control function block T_RotaryCut

- MasterLength: 150 x π = 471→K471
- MasterPulse: Number of pulses it takes for the master axis specified to rotate once/Gear ratio for the mechanism used to decelerate the master axis specified→10000/0.1=100000→K100000
- SlaveLength: 200 x π = 628→K628
- SlavePulse: K10000
- SynArea: Approximately 1/3 times the number of pulses it takes for the slave axis specified to rotate once→K3000
- MaterialSize: K500
- AccCurve=K3 (Cycloid)

◆ Operation of the cam used

- Set the parameters in the motion control function block T_RotaryCut. After M0 is set to ON, the slave axis specified will mesh with the master axis specified.
- After M10 is set to ON, the master axis will operate, and the motion of the slave axis will follow the motion of the master axis.
- Upload the cam chart in the DVP-20PM series motion controller used.



Length of a cutting<Length of the blade used: The speed of cutting roller used is the same as the speed of the feeding roller used in the synchronization zone designed. After the cutting roller used leaves the synchronization zone, the cutting roller will accelerate.

- Checking the length of a cutting: The length of a cutting is 500 mm. Convert the length of a cutting into the number of pulses which the master axis sends in a cycle. (A motor unit is used. 500/471x100000=106157.) Check whether the length of the cam curve in the cam chart uploaded is 106175 pulses. If the length of the cam curve in the cam chart uploaded is 106175 pulses, the parameters in the motion control function block T_RotaryCut are set correctly.

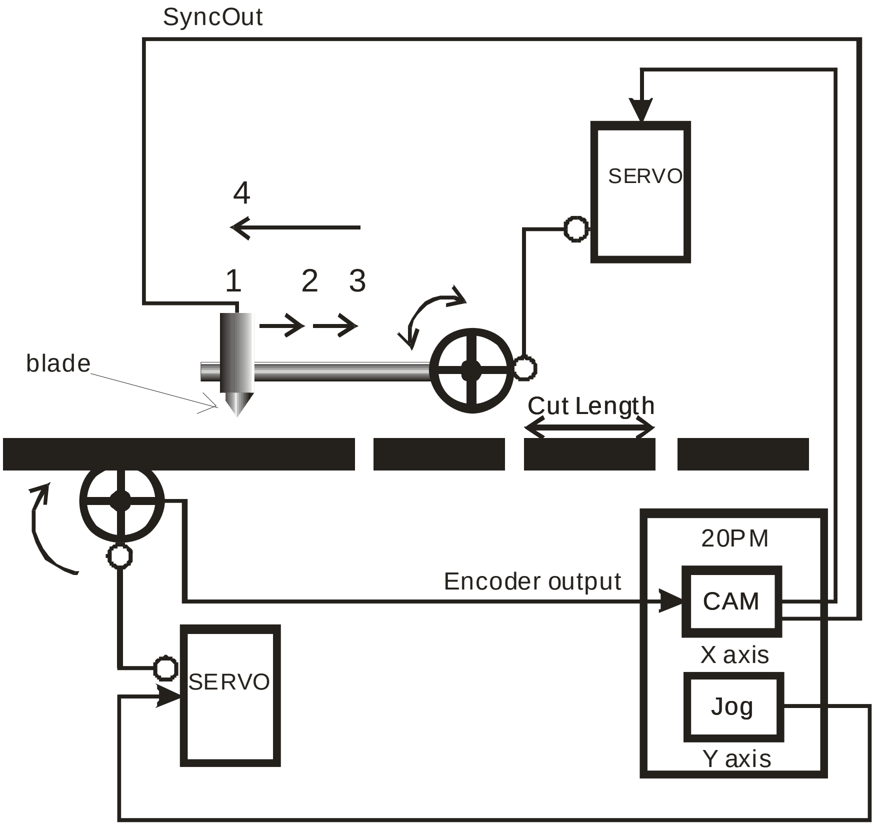
4. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
A cam curve can not be created.	Check whether the values of the parameters in the motion control block can be used to create a cam curve.

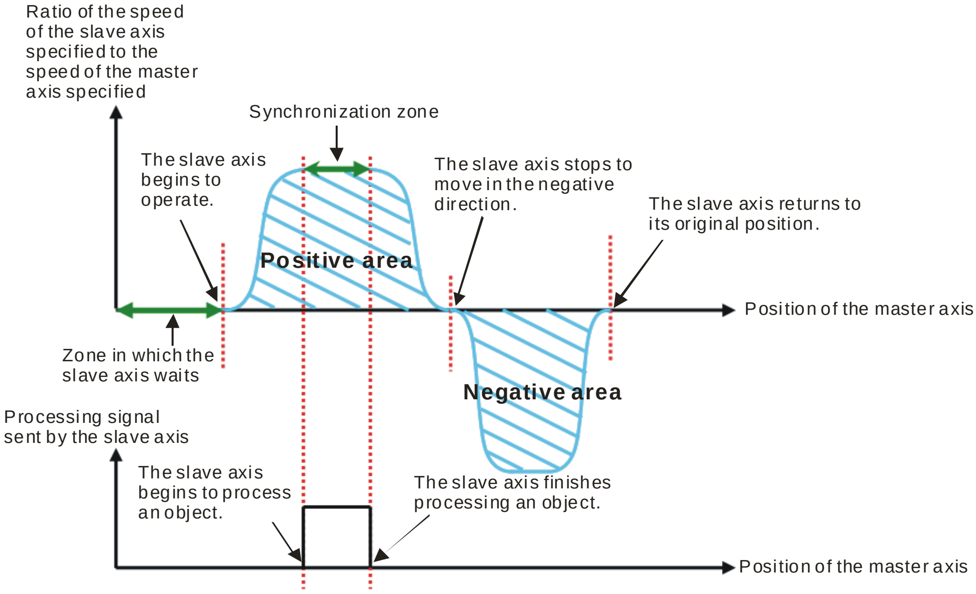
5. Modules which are supported
   The motion control function block T_RotaryCut supports DVP20PM00D and DVP20PM00M.

5.10.28 Flying Shear

1. Motion control function block
   The motion control function block T_FlyingSaw is used to create a flying shear curve, and execute electronic cam motion. Users can modify the length of a cutting by means of the MaterialSize input pin. After the length of a cutting is modified, the new cam curve created will be executed in the second or the third cam cycle.
   Concept



- Flying shear is applied to equipment which needs to move with processed objects, e.g. a tube cutter. The processing axis (slave axis) used accelerates at first so that it can catch up with a processed object. After the processing axis moves to the synchronization zone designed, the speed of the processing axis will be the same as the speed of the processed object. After the processing axis leaves the synchronization zone, it will return to its original position. The feeding axis (master axis) used always feed materials at a uniform speed.



- The movement of a flying shear is composed of catching up with an object and returning to the original position of the flying shear. The two distances for which the flying shear moves must be the same. In terms of the speed the flying shear, the positive area in which the flying shear moves must be equal to the negative area in which the flying shear moves.
- When the processing axis used process an object, the feeding axis used does not stop. The speed of the processing axis must be the same as the speed of the feeding axis. The time in which the processing axis is synchronized with the feeding axis must be long enough for the completion of the processing of an object and the movement of the object to a safe place.
- The length of the synchronization zone designed is the time it takes for the processing axis used to process an object. Besides, the design of a synchronization zone affects the operation of equipment. The bigger the synchronization zone is in a cycle, the less time it takes for the slave axis specified to accelerate/decelerate. If equipment needs to accelerate/decelerate in a short time, there will be a great impact on the electric machinery used and the blade used, and there will be an overcurrent passing through the servo used.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Master	Master axis number	WORD	K1: A0± and B0± K2: FP± for the Y-axis (No external wiring is needed.) K3: FP± and RP± for the Y-axis (No external wiring is needed.)	The value of the Master input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	Enabling electronic cam motion	BOOL	True/False	-
SlavePulseInType	Pulse type	WORD	K0: mcUD K1: mcPD K2: mcAB K3: mc4AB	The value of the SlavePulseInType input pin is valid when there is a transition in the Enable input pin's signal from low to high.

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
CycleStop	Stopping a whole cycle	WORD	BOOL	The value of the CycleStop input pin is valid when there is a transition in the Enable input pin's signal from high to low.
MasterLength	Circumference of the master axis specified (Unit: Millimeter)	WORD	K 1 ~ K 2,147,483,647	The value of the MasterLength input pin is valid when there is a transition in the Enable input pin's signal from high to low.
MasterPulse	Number of pulses it takes for the master axis specified to rotate once (Unit: Pulse)	DWORD	K1~K 2,147,483,647	The value of the MasterPulse input pin is valid when there is a transition in the Enable input pin's signal from high to low.
SlaveLength	Circumference of the slave axis specified	DWORD	K1~K 2,147,483,647	The value of the SlaveLength input pin is valid when there is a transition in the Enable input pin's signal from high to low.
SlavePulse	Number of pulses it takes for the slave axis specified to rotate once (Unit: Pulse)	FLOAT	K1~K2,147,483,647	The value of the SlavePulse input pin is valid when there is a transition in the Enable input pin's signal from high to low.
SynArea	Synchronization range (Unit: Pulse)	FLOAT	K1~K2,147,483,647	The value of the SynArea input pin is valid when there is a transition in the Enable input pin's signal from high to low.
SflySwaLength	Distance for which the slave axis specified moves forward	DWORD	K1~ K2,147,483,647	The value of the SflySwaLength input pin is valid when there is a transition in the Enable input pin's signal from high to low.
ReturnRatio	Speed ratio for the slave axis which returns to its original position (Only the ratios which are integers are supported now.)	DWORD	K1~ K2,147,483,647	The value of the ReturnRatio input pin is valid when there is a transition in the Enable input pin's signal from high to low.
MaterialSize	Length of a cutting	DWORD	K1~ K2,147,483,647	When the motion control function block is executed, the value of the MaterialSize input pin is updated repeatedly.

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
AccCurve	Acceleration curve	WORD	K0: Uniform curve K1: Uniform acceleration curve K2: SingleHypot curve K3: Cycloid	The value of the AccCurve input pin is valid when there is a transition in the Enable input pin's signal from high to low.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	• There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	• There is a transition in the Valid output pin's signal from high to low when motion stops. • There is transition in the Valid output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Valid output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low. • If the Enable input pin is set to False when the execution of the motion control function block is interrupted, the Aborted output pin will be set to False in the next cycle.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	• The execution of the motion control function block is interrupted by a command.	• There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The cam chart created is incorrect. • The axis specified is in motion before the motion control function block is executed.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
EndofProfile	A cycle is complete.	BOOL	● A cam cycle is complete.	● There is a transition in the EndOfProfile output pin's signal from high to low when the next scan cycle begins.

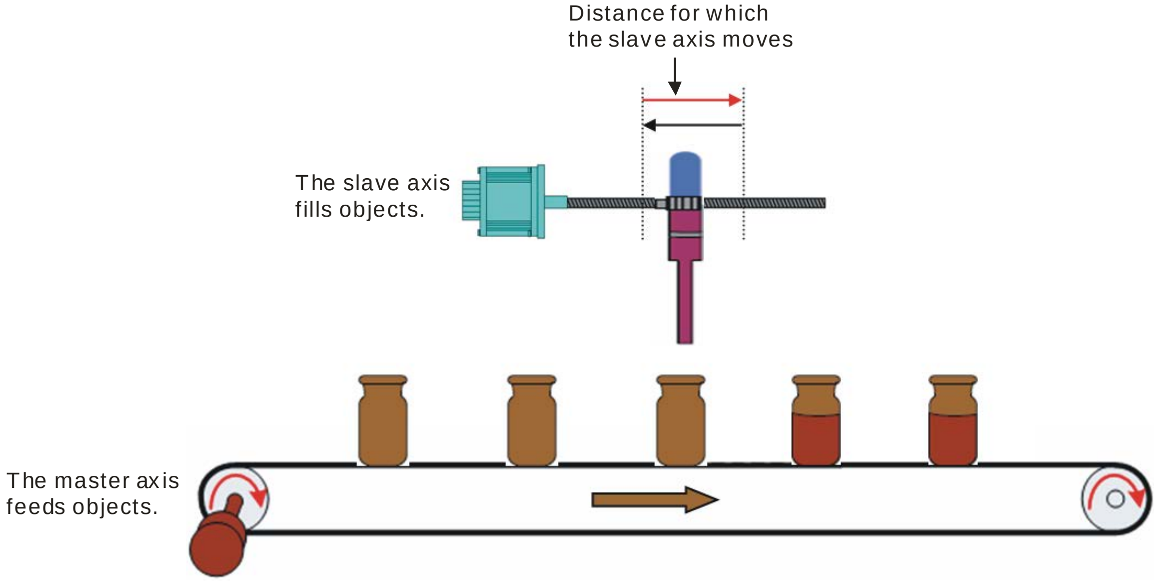
Value output pin				
Name	**Function**	**Data type**	**Output range**	**Update**
InputPulses	Number of input pulses	DWORD	K-2,147,483,648~K2,147,483,647	When the motion control function block is executed, the value of the InputPulses output pin is updated repeatedly.
InputFreq	Frequency of input pulses	DWORD	K0~K2,147,483,647	When the motion control function block is executed, the value of the InputFreq output pin is updated repeatedly.

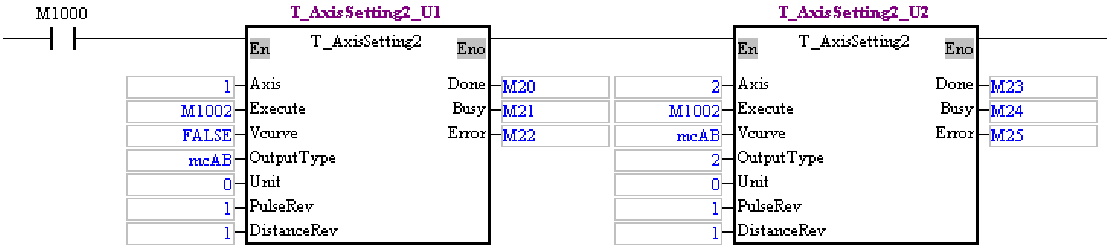
Notes:

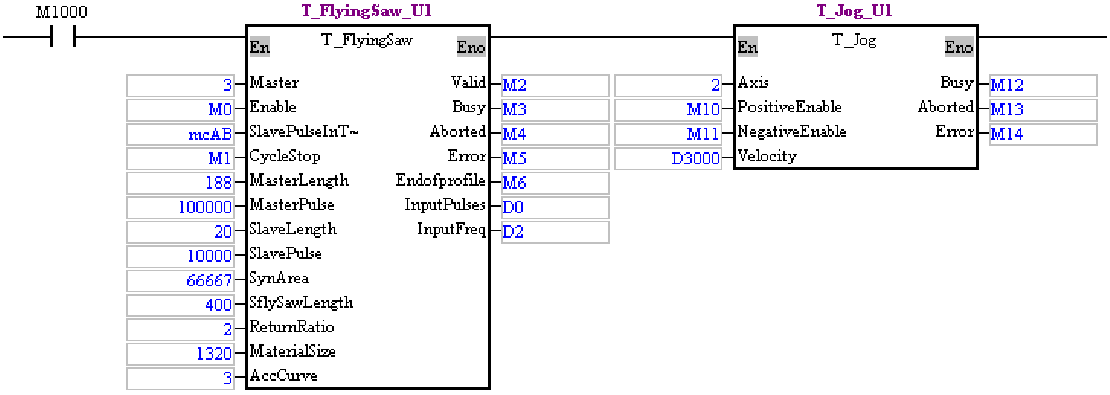
- If the motion control function block T_FlyingSaw is used to create cam chart 0 and cam chart 1, the resolution of cam chart 0 must be 600, and the resolution of cam chart 1 must be 600.
- If the slave axis specified moves to the synchronization zone designed when the motion control function block T_FlyingSaw is used, CLR0 will be ON. If the slave axis specified leaves the synchronization zone designed, CLR0 will be OFF.

3. Example

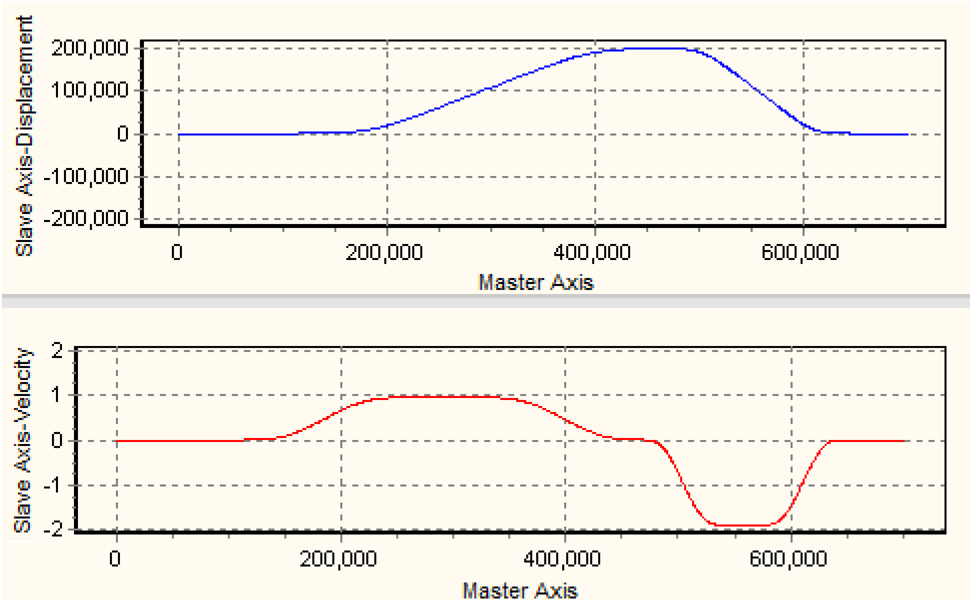
In this example, the setting of the motion control function block T_FlyingSaw is described. The master axis used and the slave axis used are connected to Delta ASDA-A2 series AC servo drives. The first axis in a DVP-20PM series motion controller sends pulses to the slave axis, and the second axis in the DVP-20PM series motion controller sends pulses to the master axis.







- Descriptions of the parameters in the system used
  - Pulse type: A/B
  - Diameter of the master axis: 60 mm
  - Gear ratio for the mechanism used to decelerate the master axis:1:10
  - Gear ration for the master axis: 128/4→128/4→The number of pulses it takes for the AC servo drive connected to the master axis to rotate once is 10,000. (Take Delta ASDA-A2 series AC servo drive for instance.)
  - Pitch: 20 mm
  - Distance for which the slave axis moves forward: 400 mm
  - Gear ratio for the slave axis: 128/4→The number of pulses it takes for the AC servo drive connected to the master axis to rotate once is 10,000. (Take Delta ASDA-A2 series AC servo drive for instance.)
  - Length of a cutting: 1320 mm
- Setting the parameters in the motion control function block T_FlyingSaw
  - MasterLength: 60 x π = 188→K188
  - MasterPulse: Number of pulses it takes for the master axis specified to rotate once/Gear ratio for the mechanism used to decelerate the master axis specified→10000/0.1=100000→K100000
  - SlaveLength: K20
  - SlavePulse: K10000
  - SynArea: Approximately 1/3 times the distance for which the slave axis specified moves forward→400/20x10000/3=66667→K66667
  - SflySwaLength: K400
  - ReturnRatio: K2
  - MaterialSize: K1320
  - AccCurve = K3 (Cycloid)
- Operation of the cam used
  - Set the parameters in the motion control function block T_FlyingSaw. After M0 is set to ON, the slave axis specified will mesh with the master axis specified.
  - After M10 is set to ON, the master axis will operate, and the motion of the slave axis will follow the motion of the master axis.
  - Upload the cam chart in the DVP-20PM series motion controller used.



- Checking the length of a cutting: The length of a cutting is 1320 mm. Convert the length of a cutting into the number of pulses which the master axis sends in a cycle. (A motor unit is used. 1320/188x100000=702127.) Check whether the length of the cam curve in the cam chart uploaded is 702127 pulses. If the length of the cam curve in the cam chart uploaded is 702127 pulses, the parameters in the motion control function block T_FlyingSaw are set correctly.

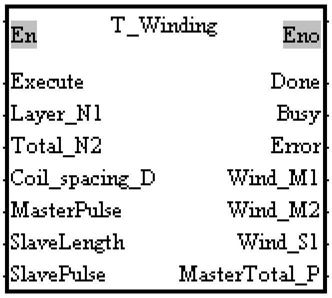
4. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
A cam curve can not be created.	Check whether the values of the parameters in the motion control block can be used to create a cam curve.

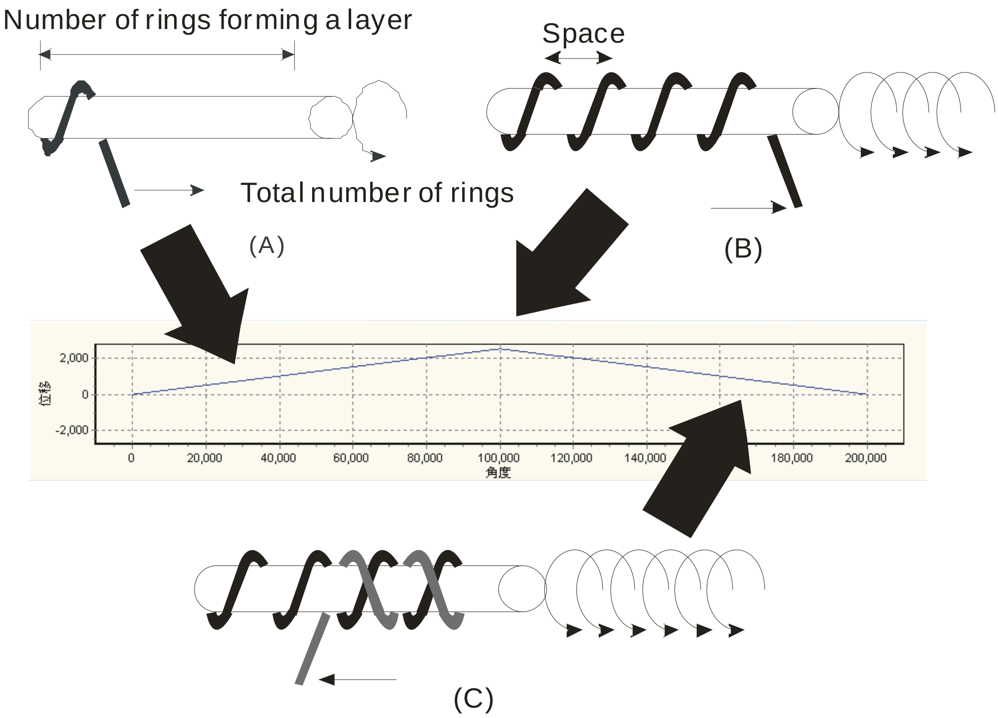
5. Modules which are supported

   The motion control function block T_FlyingSaw supports DVP20PM00D and DVP20PM00M.

5.10.29 Creating a Curve for a Wire Winding Machine



1. Motion control function block
   The motion control function block T-Winding is used to create a curve for a wire winding machine.
   Concept



The winding of a coil is shown above. The slave axis specified turns clockwise in the right direction (A). The slave axis sends pulses until the number of rings forming a layer is reached (B). The slave axis turns counterclockwise in the negative position, and sends pulses until the number of rings forming a layer is reached (C). The number of rings which forms a layer when the slave axis turns clockwise in the right direction is equal to the number of rings which form a layer when the slave axis turns counterclockwise in the negative position. The slave axis turns clockwise in the right direction, and turns counterclockwise in the negative position. It winds wires around the master axis specified. As a result, the electronic cam curve created needs to show that the slave axis moves from one end of the master axis to the other end of the master axis, and then returns. The relation between the slave axis and the master axis can be shown by only three points in the electronic cam curve. The values of the parameters in the table below can be used to create an electronic cam curve.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Execute	The creation of a cam curve is enabled when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
Layer_N1	Number of rings forming a layer	DWORD	K1~K2,147,483,647	The value of the Layer_N1 input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Total_N2	Total number of rings	DWORD	K1~K2,147,483,647	The value of the Total_N2 input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Coil_spacing_D	Space: Diameter of a wire+Interval between two wires (μm)	DWORD	K1~K2,147,483,647	The value of the Coil_spacing_D input pin is valid when there is a transition in the Execute input pin's signal from low to high.
MasterPulse	Number of pulses it takes for the master axis specified to rotate once	DWORD	K1~K2,147,483,647	The value of the MasterPulse input pin is valid when there is a transition in the Execute input pin's signal from low to high.
SlaveLength	Circumference of the slave axis specified (mm)	DWORD	K1~K2,147,483,647	The value of the SlaveLength input pin is valid when there is a transition in the Execute input pin's signal from low to high.
SlavePulse	Number of pulses it takes for the slave axis specified to rotate once	DWORD	K1~K2,147,483,647	The value of the SlavePulse input pin is valid when there is a transition in the Execute input pin's signal from low to high.

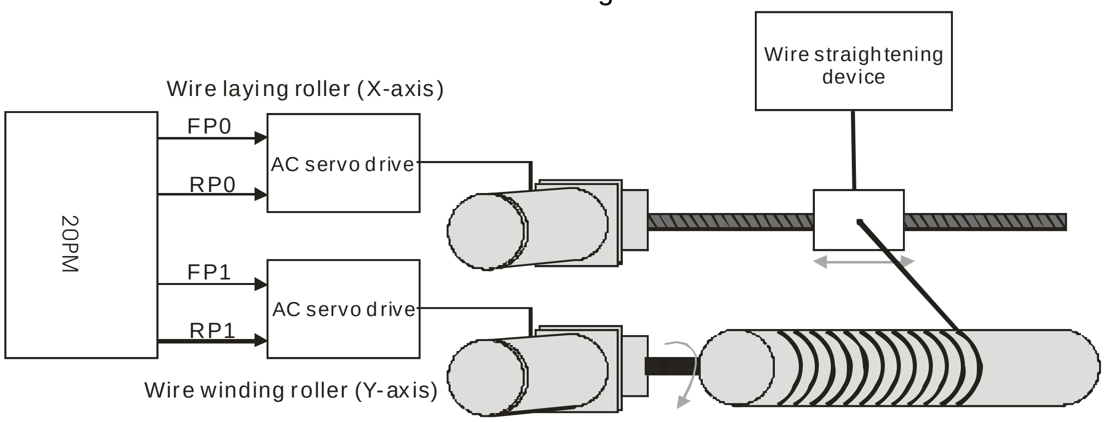
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal when motion is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

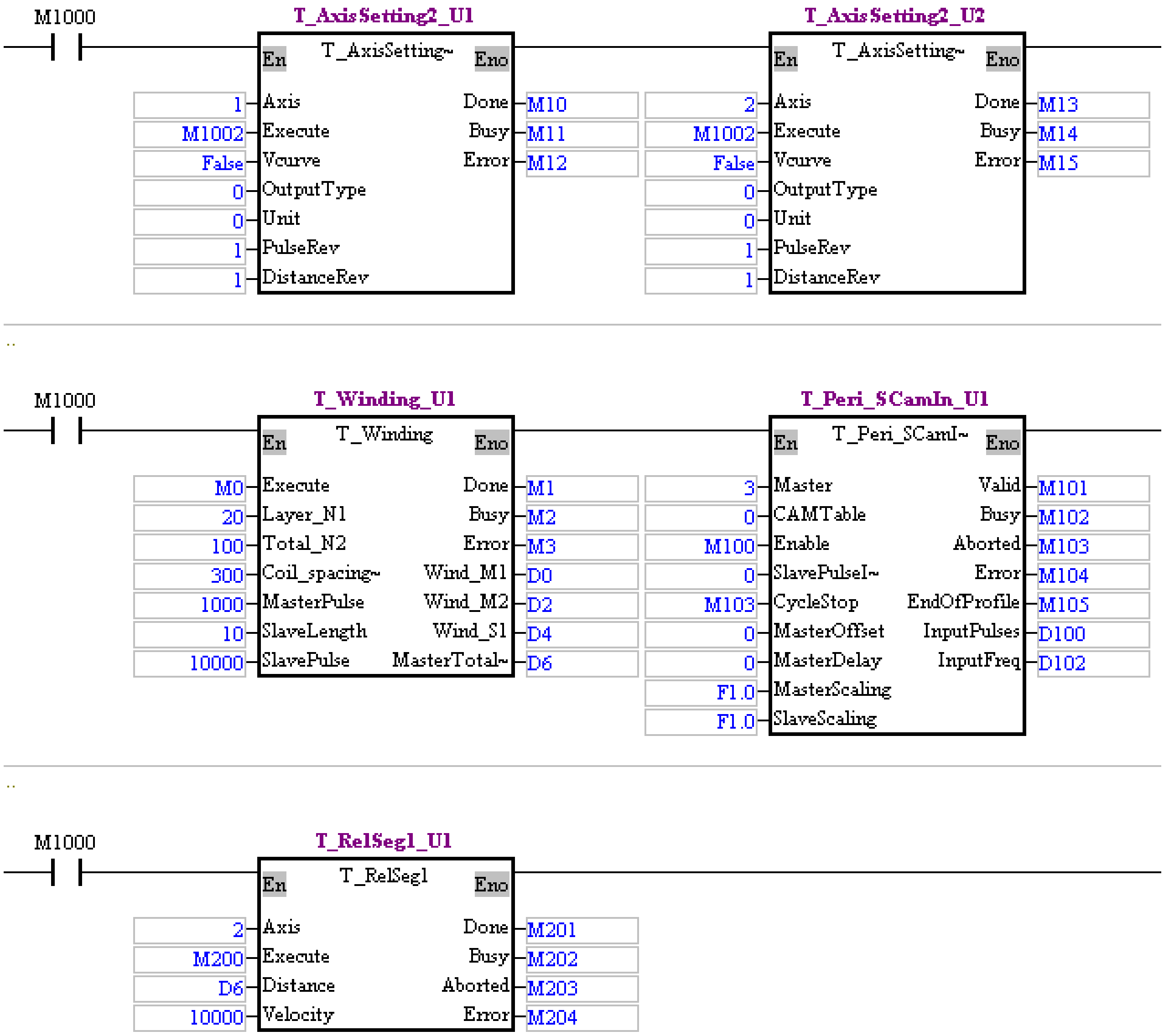
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

Note: Before the motion control function block T-Winding is used, users have to create cam chart 0, and the resolution of cam chart 0 must be 4.

3. Example

   The use of the motion control function block T-Winding to create a cam curve is described.





- Descriptions of the parameters in the system used
  - Number of rings forming a layer: 20
  - Total number of rings: 100
  - Space: 0.3 mm
  - Number of pulses it takes for the master axis specified to rotate once: 1000 pulses
  - Circumference of the slave axis specified: 10 mm
  - Number of pulses it takes for the slave axis specified to rotate once: 10,000 pulses
- Setting the parameters in the motion control function block T-Winding
  - Layer_N1: K20
  - Total_N2: K100
  - Coil_spacing_D: 0.3mm=300 um→K300
  - MasterPulse: K1,000
  - SlaveLength: K10
  - SlavePulse: K10,000

- Operation of the cam used
  - Set M0 to ON, and then upload the cam chart in the DVP-20PM series motion controller used.

  - After M100 is set to ON, the first axis will mesh with the second axis.
  - After M200 is set to ON, the master axis will operate, and the motion of the slave axis will follow the motion of the master axis. First, the master axis sends 20,000 pulses until 20 rings forming the first layer are gotten. When the master axis sends 20,000 pulses, the slave axis sends 6,000 pulses. Next, the master axis sends another 20,000 pulses until another 20 rings forming the second layer are gotten. When the master axis sends another 20,000 pulses, the slave axis sends another 6,000 pulses. The process is repeated until the total number of pulses sent by the master axis is 100,000 and the number of rings gotten is 100.

4. Troubleshooting

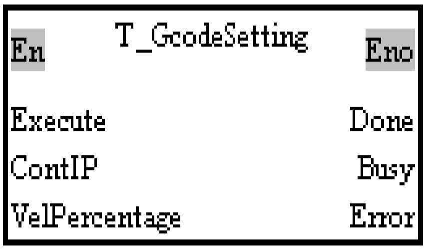
Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

5. Modules which are supported

The motion control function block T-Winding supports DVP20PM00D and DVP20PM00M.

5.11 Multiaxial Motion Control Function Blocks

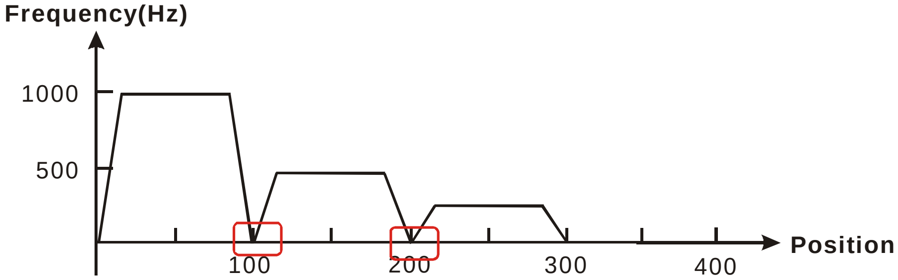
5.11.1 Setting the Parameters of G-code Motion



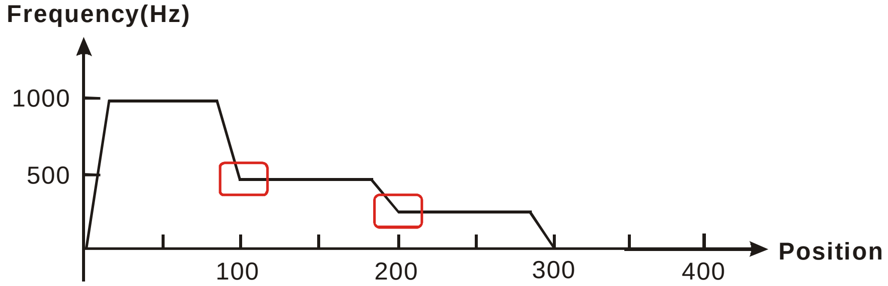
1. Motion control function block

The motion control function block T_GcodeSetting is used to set the parameters of G-code motion. The value of the ContIP input pin indicates the minimum speed to which the speed of continuous interpolation decreases. If the speed of G-code motion is less than the speed indicated by the value of the ContIP input pin, the G-code motion will move at the speed indicated by the value of the ContIP input pin. The value of the VelPercentage input pin indicates the percentage for the values of the speed parameters of G-codes.

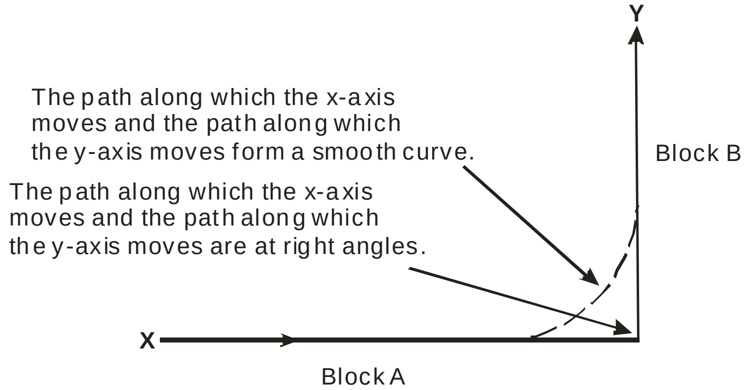
- Continuous interpolation
  - If users set the minimum speed to which the speed of continuous interpolation decreses by means of the ContIP input pin, the smaller speed will be taken as a turning point after the value of ContIP input pin is compared with the acutal speed to which the speed of continuous interpolation decreses. If the value of the ContIP input pin is greater than 0, a continuous path can be created. The maximum value of the ContIP input pin is 500000. If the value of the ContIP input pin is greater than 500000, the speed of continuous interpolation will decrese to 500 kHz.
  - If the value of the ContIP input pin is K0 (there is no continuous interpolation), the speed of motion will decrease to 0 kHz no matter what the actual deceleration is.



- The value of the ContIP input pin is K500 (there is continuous interpolation). After the value in D1796 is compared with the acutal deceleration, the smaller deceleration will be taken as a turning point. After the value of the ContIP input pin is compared with the acutal deceleration, the smaller deceleration will be taken as a turning point.



- If the ContIP input pin is not set, the path along which the X-axis moves and the path along which the Y-axis moves will be at right angles. If the ContIP input pin is set, the path along which the X-axis moves and the path along which the Y-axis moves will form a smooth curve.
  G01 X100 F1000; (Block A)
  Y100; (Block B)



Block A: Path along which the X-axis moves; Block B: Path along which the Y-axis moves

- Percentage for the values of the speed parameters of G-codes
  - Relation between the value of the VelPercentage input pin and the ratio for the values of the speed parameters of G-codes

Setting value of the VelPercentage input pin	Ratio
100	1
50	1/2
1000	10

If the value of the VelPercentage input pin is 100, the speeds of the G-codes used will be the orginial speeds. If the value of the VelPercentage input pin is 1,000, the speeds of the G-codes used will be multipled by 10. If the value of the VelPercentage input pin is 50, the speeds of the G-codes used will be half the original speed.

If the result gotten from the multiplication of the speed of a G-code by the value of the VelPercentage input pin is greater than 500000 Hz, the G-code will move the axes used at a speed of 500000 Hz.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Execute	G-code motion stops when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
ContIP	Minimum speed to which the speed of continuous interpolation decreases (Hz)	DWORD	K0~K500000	The value of the ContIP input pin is valid when there is a transition in the Execute input pin's signal from low to high.
VelPercentage	Percentage for the values of the speed parameters of G-codes	WORD	K0~K65,535	The value of the VelPercentage input pin is valid when there is a transition in the Execute input pin's signal from low to high.

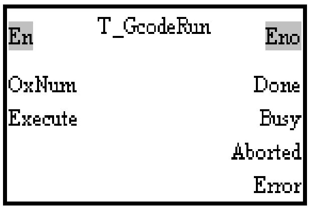
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal when motion is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect.	• There is transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Modules which are supported
   The motion control function block T_GcodeSetting supports DVP20PM00D and DVP20PM00M.

5.11.2 Executing G-code Motion



1. Motion control function block
   The motion control function block T_GcodeRun is used to set and execute an Ox motion subroutine. The value of the OxNum indicates an Ox motion subroutine number.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
OxNum	Ox motion subroutine number	WORD	OX0~OX99: 0~99 SD card: 100~199	The value of the OxNum input pin is valid when there is a transition in the Execute input pin's signal from low to high.
Execute	G-code motion stops when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	• There is a transition in the Done output pin's signal when motion is complete.	• There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Aborted output pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The slave axis specified meshes with the master axis specified before the motion control function block is executed.	● There is transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.
Aborted	The execution of the motion control function block is interrupted by a command.	BOOL	● The execution of the motion control function block is interrupted by a command.	● There is a transition in the Aborted output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

3. Troubleshooting

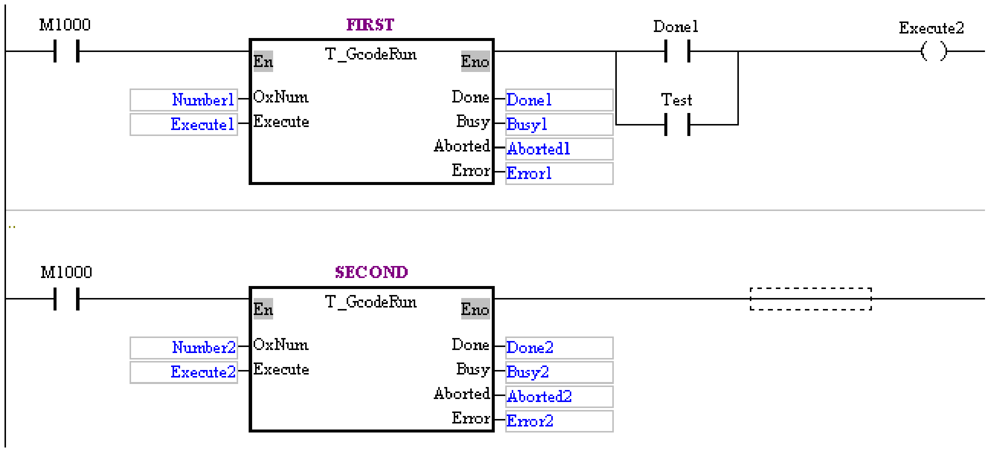
Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Example

Purposes:

- After the first G-code motion is complete, the second G-code motion will be executed.
- The second G-code motion is executed before the execution of the first G-code motion is complete.

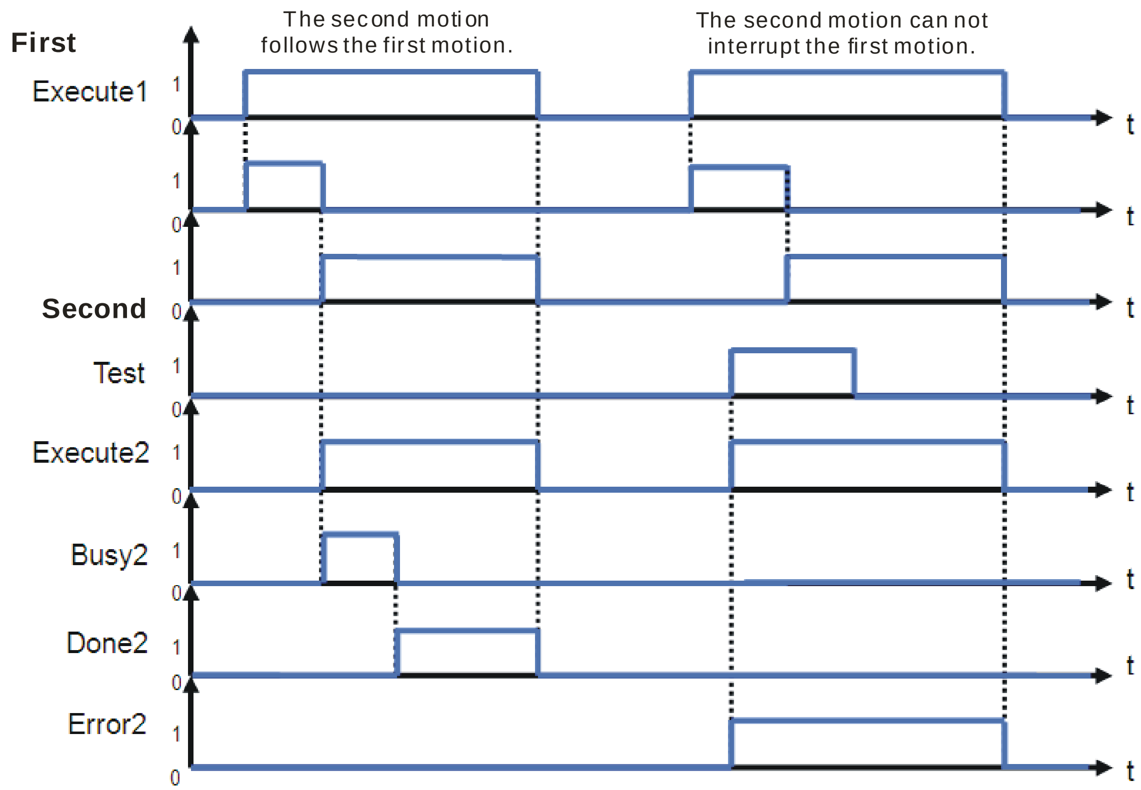
The motion control function block named FIRST and the motion control function block named SECOND are set so that two different Ox motion subroutines are executed.



- After the first G-code motion is complete, the second G-code motion will be executed.
  Steps:
  (a) Set Execute1 to True.
  (b) Wait for a transition in Done2's signal from low to high or a transition in Error2's signal from low to high.
- The second G-code motion is executed before the execution of the first G-code motion is complete.
  Steps:
  (a) Set Execute1 to True.
  (b) Set Test to ON when Busy1 is set to True.
  (c) Wait for a transition in Done2's signal from low to high or a transition in Error2's signal from low to high.

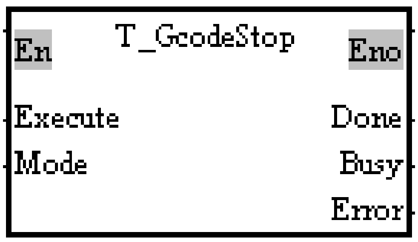
Timing diagram:

Number1 = Number2



5. Modules which are supported
   The motion control function block T_GcodeRun supports DVP20PM00D and DVP20PM00M.

5.11.3 Stopping G-code Motion



1. Motion control function block
   The motion control function block T_GcodeStop is used to stop the execution of an Ox motion subroutine.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Execute	G-code motion stops when there is a transition in the Execute input pin's signal from low to high.	BOOL	True/False	-
Mode	Mode of stopping G-code Motion	WORD	K0~K2 (*1)	The value of the Mode input pin is valid when there is a transition in the Execute input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Done	The execution of the motion control function block is complete.	BOOL	● There is a transition in the Done output pin's signal from low to high when the execution of the motion control function block is complete.	● There is a transition in the Done output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Done output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect.	● There is transition in the Error output pin's signal from high to low when there is a transition in the Execute input pin's signal from high to low.

*1: Value of the Mode input pin

Value	Definition
0	G-code motion is stopped immediately. It will be executed again next time the motion control function block T_GcodeRun is executed.
1	G-code motion is stopped immediately. The distance which remains will be completed next time the motion control function block T_GcodeRun is executed.
2	G-code motion is stopped immediately. The next G-code motion will be executed next time the motion control function block T_GcodeRun is executed.

3. Troubleshooting

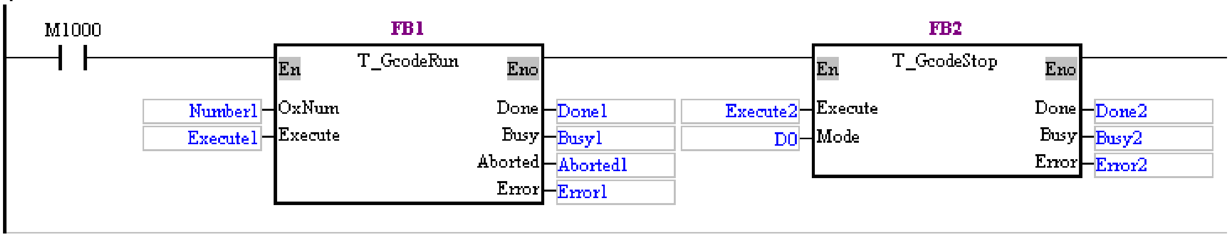
Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Example

Purpose:

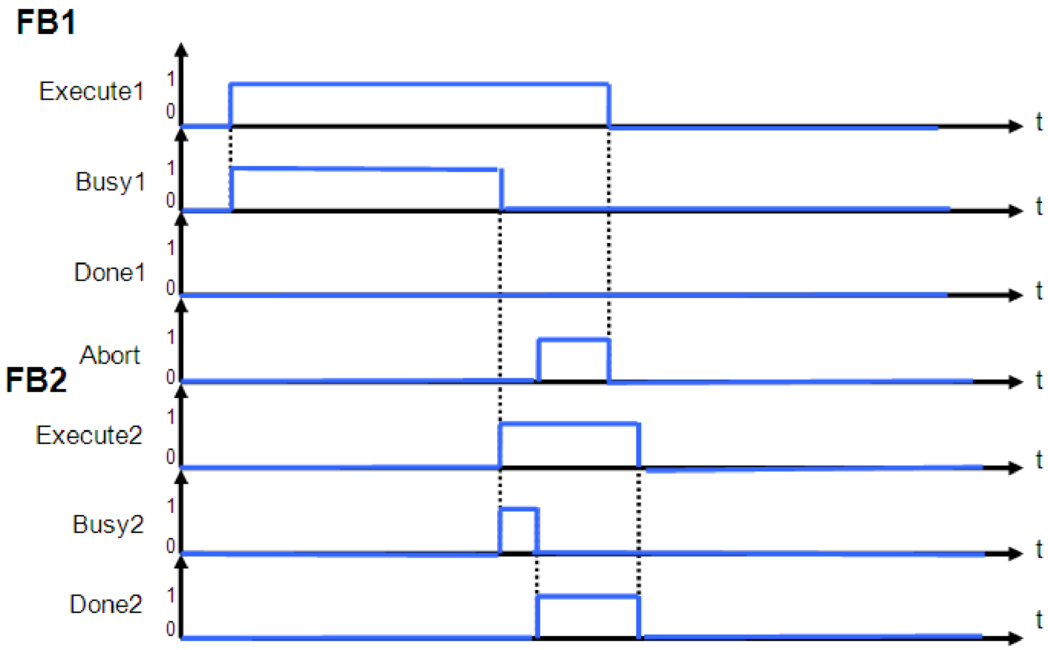
- Write K0 into D0.
- The execution of the first G-code motion is stopped before it is complete.

The motion control function block named FB1 is set so that an Ox motion subroutine is executed. The motion control function block named FB2 is set so that the execution of the Ox motion subroutines stops.



Set Execute1 to True. Execute the G-codes in the Ox motion subroutine specified. Set Execute2 to True before the execution of the G-codes in the Ox motion subroutine specified is complete. The execution of the Ox motion subroutine specified is stopped, and Aborted1 is set to True.

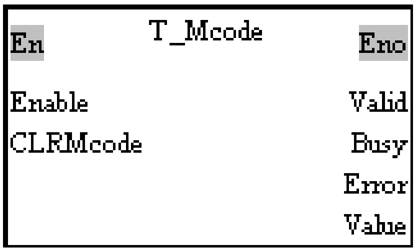
Timing diagram:



5. Modules which are supported

   The motion control function block T_GcodeStop supports DVP20PM00D and DVP20PM00M.

5.11.4 Reading an M-code



1. Motion control function block

   The motion control function block T_Mcode is used to read an M-code, and clear the M-code specified. The CLRMcode input pin is used to clear the M-code specified.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Enable	Enabling the reading of an M-code	WORD	True/False	-
CLRMcode	An M-code is cleared when there is a transition in the CLRMcode input pin's signal from low to high.	BOOL	True/False	The value of the CLRMcode input pin is valid when the motion control function block is executed.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	The Valid output pin is set to True when an M-code is executed.	BOOL	● There is a transition in the Valid output pin's signal from low to high when an M-code is executed.	● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Enable input pin's signal from low to high. ● There is a transition in the CLRMcode input pin's signal from low to high when the Valid output pin is set to True. There is a transition in the Valid output pin's signal from high to low when there is a transition in the CLRMcode input pin's signal from low to high.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Execute input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Time when a value is valid**
Value	When the Valid output pin is set to True, the value of the Value output pin indicates the M-code which is executed.	WORD	K0 ~ 4095	When the Valid output pin is set to True, the value of the Value output pin is updated repeatedly.

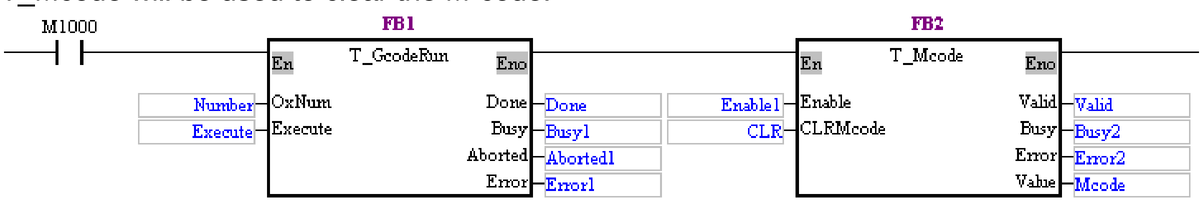
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Example

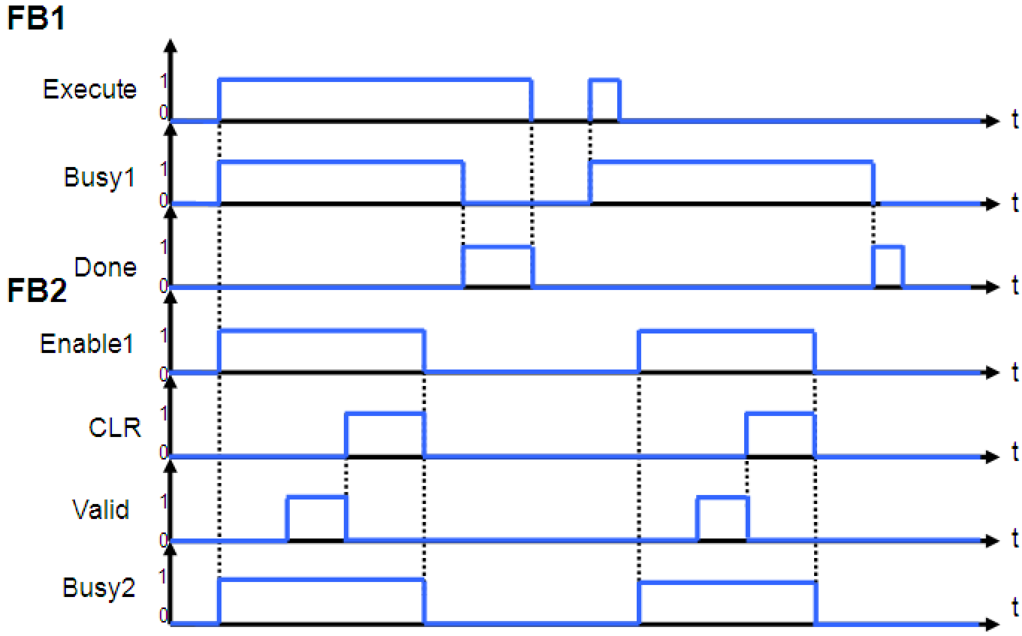
Purpose:

- When the first G-code motion is executed, the motion control function block T_Mcode is used to check the status of an M-code. If an M-code is executed, the motion control function block T_Mcode will be used to clear the M-code.



The motion control function block named FB1 is set so that an Ox motion subroutine is executed. The motion control function block named FB2 is set so that the status of an M-code is checked.

Set Execute to True. Execute the G-codes in the Ox motion subroutine specified. Set Execute1 to True before the execution of the G-codes in the Ox motion subroutine specified is complete. Check the status of the M-code which is being executed. When an M-code is executed, Valid is set to True. CLR is used to clear the M-code which is executed.

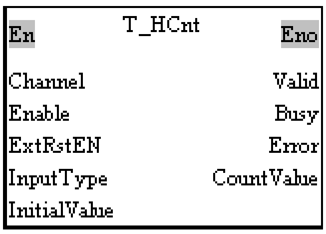


5. Modules which are supported

   The motion control function block T_Mcode supports DVP20PM00D and DVP20PM00M.

5.12 Other Motion Control Function Blocks

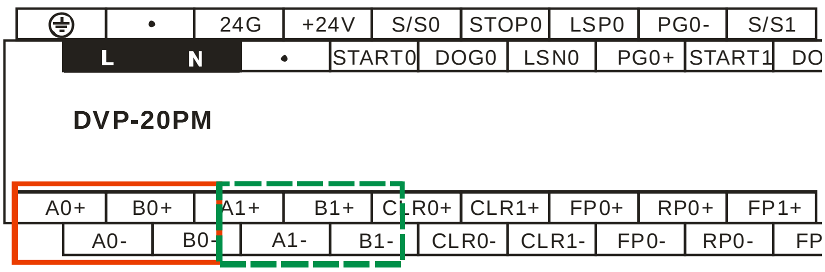
5.12.1 High-speed Counter



1. Motion control function block

   The motion control function block T_HCnt is used to start a high-speed counter. The value of the Channel input pin indicates a counter number, and the value of the InputType input pin indicates an input pulse type. The ExtRstEN input pin is used to set an external reset switch. The value of the InitialValue input pin is the initial value in the counter specified, and the value of the CountValue output pin is the value in the counter specified.

   The input terminals for the high-speed counters in a DVP-20PM series motion controller are shown below.



   The input terminals in the solid frame are for high-speed counter 0, and the input terminals in the dotted frame are for high-speed counter 1.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Channel	Counter number	WORD	0~1 (*1)	The value of the Channel input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	The motion control function block is enabled when there is a transition in the Enable input pin's signal from low to high.	BOOL	True/False	-
InputType	Input pulse type	WORD	mcUD: 0 mcPD: 1 mcAB: 2 mc4AB: 3	When the motion control function block is executed, the value of the InputType input pin is updated repeatedly.
ExtRstEN	External reset switch	BOOL	True/False (*2)	The value of the ExtRstEN input pin is valid when there is a transition in the Enable input pin's signal from low to high.
InitialValue	Initial value in the counter specified (Pulse)	DWORD	K0~2,147,483,647	The value of the InitialValue input pin is valid when there is a transition in the Enable input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	● There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The source specified has been occupied.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

Value output pin				
Name	Function	Data type	Output range	Time when a value is valid
CountValue	Value in the counter specified	DWORD	K0~2,147,483,647	When the Valid output pin is set to True, the value of the CountValue output pin is updated repeatedly.

*1: Value of the Channel input pin

Value	Definition
0	C200
1	C204

*2: External terminals for resetting the high-speed counters in a DVP-20PM series motion controller

Counter number	Reset terminal
0	PG0
1	PG1

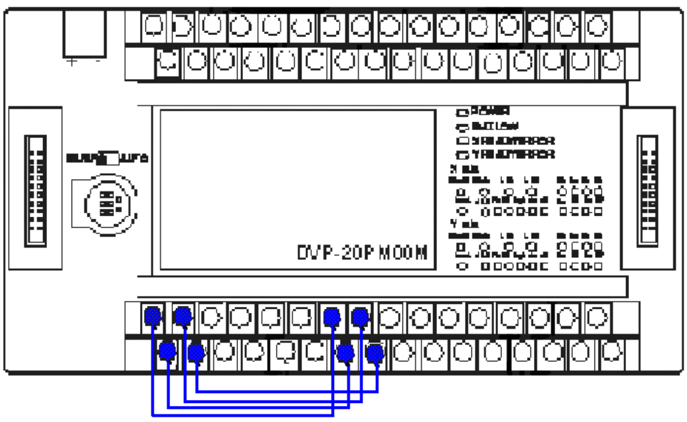
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The counter specified has been used.	Use another counter, or stop the counter which has been used.

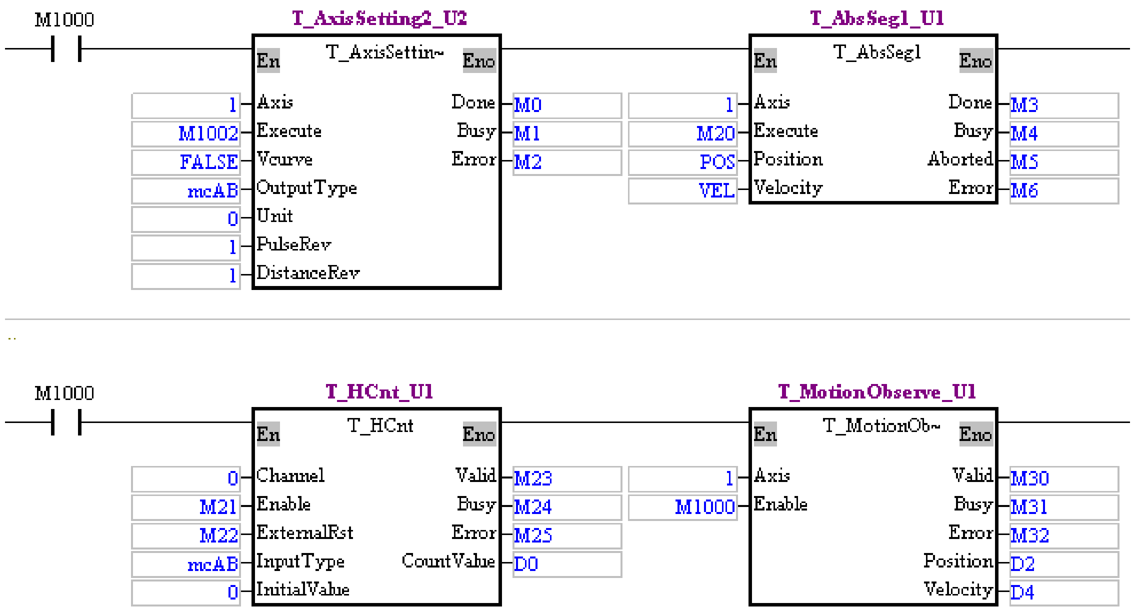
4. Example

Purpose:

- The first axis sends pulses to high-speed counter 0. Users can check whether the number of pulses output is the same as the number of pulses input. The external wiring required is shown below.



FP0 is connected to A0.
RP0 is connected to B0.

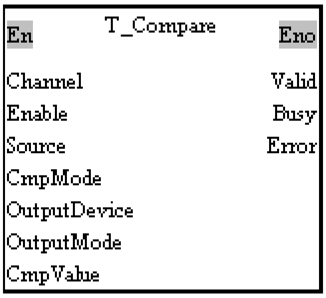


- The pulses output by the first axis are A/B-phase pulses.
- After M21 is set to ON, high-speed counter 0 will be started.

- The value of POS is 30,000, and the value of VEL is 10,000.
- After M20 is set to ON, the absolute single-speed motion set will be started.
- Compare the value in D0 (the value in high-speed counter 0) with the value in D2 when M3 is ON.

5. Modules which are supported
   The motion control function block T_HCnt supports DVP20PM00D and DVP20PM00M.

5.12.2 Setting High-speed Comparison



1. Motion control function block
   The motion control function block T_Compare is used to start high-speed comparison. The value of the Channel input pin indicates a comparator number, the value of the Source input pin indicates a source, the value of the CmpMode input pin indicates a comparison condition, and the value of the OutputDevice indicates an output device.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Channel	Comparator number	WORD	0~7	The value of the Channel input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	The motion control function block is enabled when there is a transition in the Enable input pin's signal from low to high.	BOOL	True/False	-
Source	Source	WORD	mcCmpAxis1: 0 mcCmpAxis2: 1 mcCmpAxis3: 2 mcCmpC200: 4 mcCmpC204: 5	The value of the Source input pin is valid when there is a transition in the Enable input pin's signal from low to high.
CmpMode	Comparison condition	WORD	1: = 2: ≧ 3: ≦	The value of the CmpMode input pin is valid when there is a transition in the Enable input pin's signal from low to high.
OutputDevice	Output device	WORD	mcCmpCLR0: 0 mcCmpCLR1: 1 mcCmpY2: 2 mcCmpY3: 3 mcCmpRstC200: 4 mcCmpRstC204: 5	The value of the OutputDevice input pin is valid when there is a transition in the Enable input pin's signal from low to high.
OutputMode	Output mode	BOOL	mcCmpSet: True mcCmpRst: False	The value of the OutputMode input pin is valid when there is a transition in the Enable input pin's signal from low to high.

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
CmpValue	Value with which a source is compared	DWORD	K-2,147,483,647~ K2,147,483,647	The value of the CmpValue input pin is valid when there is a transition in the Enable input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	● There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The source specified has been occupied.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

Please note that the number of high-speed comparators plus the number of high-speed capturers is 8 at most.

◆ Source
- ■ mcCmpAxis1: Present position of the first axis
- ■ mcCmpAxis2: Present position of the second axis
- ■ mcCmpAxis3: Present position of the third axis (only applicable to DVP20PM00M)
- ■ mcCmpC200: Present value in C200
- ■ mcCmpC204: Present value in C204

◆ Output device
- ■ mcCmpCLR0: CLR0
- ■ mcCmpCLR1: CLR1
- ■ mcCmpY2: Y2
- ■ mcCmpY3: Y3
- ■ mcCmpRstC200: Resetting C200
- ■ mcCmpRstC204: Resetting C204

◆ Output mode
- ■ The device specified is CLR0, CLR1, Y2, or Y3.
  McCmpSet: Enabling the output device specified
  McCmpRst: Diabling the output device specified

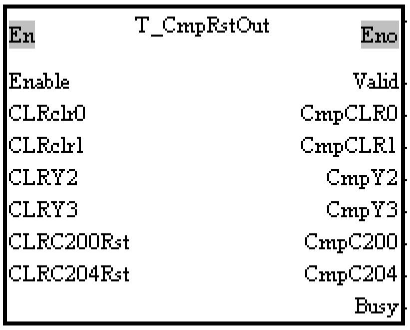
- The device specified is C200 or C204.
  McCmpSet: The value in the counter specified is cleared.
  McCmpRst: The counter specified counts.

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The comparator specified has been used.	Use another comparator.

4. Modules which are supported
   The motion control function block T_Compare supports DVP20PM00D and DVP20PM00M.

5.12.3 Resetting High-speed Comparison



1. Motion control function block
   The motion control function block T_CmpRstOut is used to reset high-speed comparison. CLRclr0, CLRclr1, CLRY2, CLRY3, CLRC200Rst, and CLRC204Rst determine the output devices which will be reset. The values of the output pins indicate whether the output devices CLR0, CLR1, Y2, Y3, C200, and C204 are enabled or disabled.
2. Input pins/Output pins

Input pin				
Name	Name	Name	Name	Name
Enable	The motion control function block is enabled when there is a transition in the Enable input pin's signal from low to high.	BOOL	True/False	-
CLRclr0	Resetting the output devices CLR0, CLR1, Y2, Y3, C200, and C204	BOOL	True/False	When the motion control function block is executed, the values of these input pins are updated repeatedly.
CLRclr1				
CLRY2				
CLRY3				
CLRC200Rst				
CLRC204Rst				

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	• There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	• There is a transition in the Valid output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	• Input values are incorrect. • The source specified has been occupied.	• There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Time when a value is valid**
CmpCLR0 CmpCLR1 CmpY2 CmpY3 CmpC200 CmpC204	States of the output devices CLR0, CLR1, Y2, Y3, C200, and C204	BOOL	True/False	When the Valid output pin is set to True, the values of these output pins are updated repeatedly.

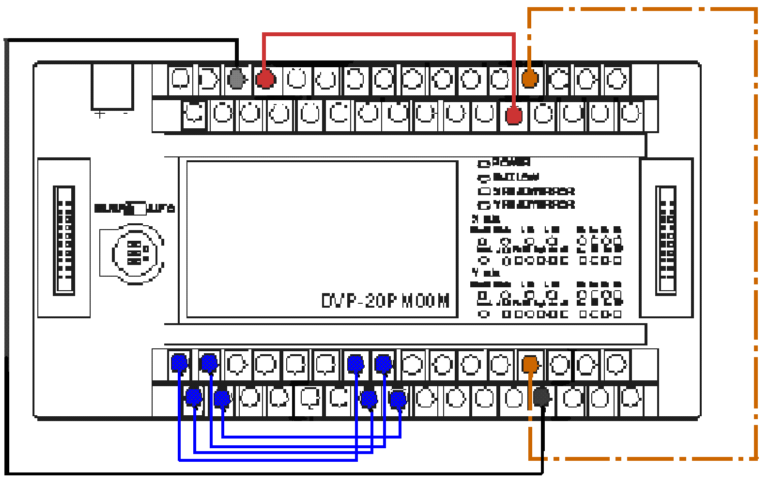
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

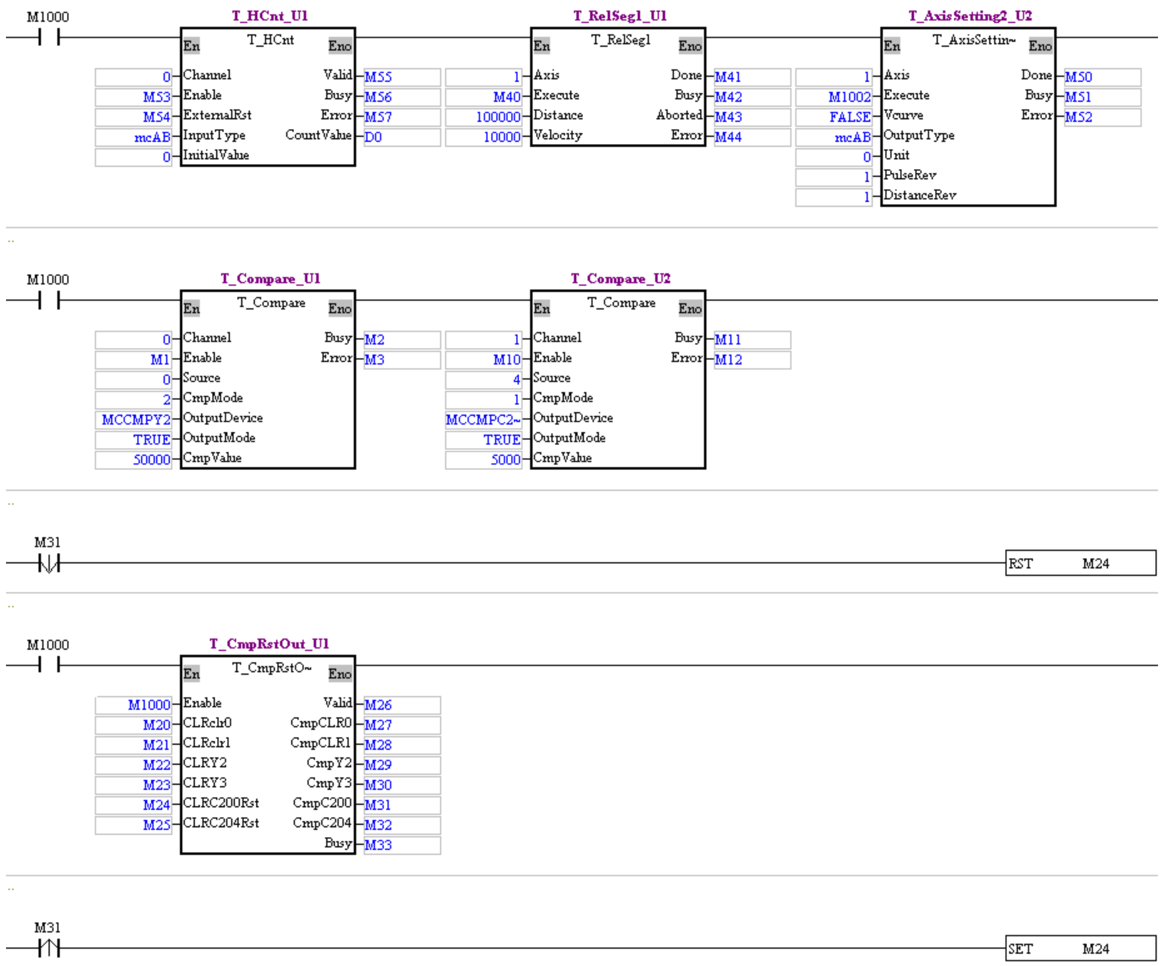
4. Example

Purpose:

- Two high-speed comparators are set. When the conditions set are met, users can check whether the output devices specified are set/reset. The external wiring required is shown below.



FP0 is connected to A0.
RP0 is connected to B0.
C2 is connected to -24G.
Y2 is connected to X0.
S/S2 is connected to +24V.



- If the program is executed, the pulses output by the first axis will be A/B-phase pulses, the motion control function block T_CmpRstOut will be started, and the states of output devices will be read.
- After M53 is set to ON, a high-speed counter will be started.
- After M1 is set to ON, high-speed comparator 0 will be started.

Setting high-speed comparator 0: If the number of pulses output by the first axis is greater than or equal to 50,000, Y2 will be set to ON.

- After M10 is set to ON, high-speed comparator 1 will be started.

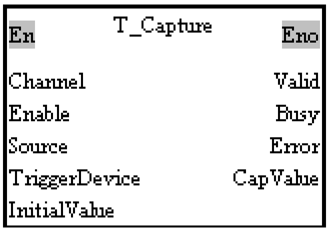
Setting high-speed comparator 1: If the value in C200 is equal to 5,000, the value in C200 will be cleared to 0.

- After M40 is set to ON, the first axis will move for 100,000 pulses.
- If the value in C200 is equal to 5,000, and M31 is ON, the comparison condition set for high-speed comparator 1 is met, and the value in C200 is cleared to 0. The value in C200 will be cleared to 0 next time the value in C200 becomes 5,000. If M31 is not reset, high-speed comparator 1 will not act next time the comparison condition set for high-speed comparator 1 is met.
- If the number of pulses output by the first axis is 100,000, the comparison condition set for high-speed comparator 0 is met, and Y2 is set to ON. When Y2 is set to ON, the users can check whether X0 is ON.
- If X0 is ON, the comparison condition set for high-speed comparator 0 is met. The users can turn X0 OFF by means of M22.

5. Modules which are supported

   The motion control function block T_CmpRstOut supports DVP20PM00D and DV-P20PM00M.

5.12.4 Setting High-speed Capture



1. Motion control function block

   The motion control function block T_Capture is used to start high-speed capture. The value of the Channel input pin indicates a capturer number. The value of the Source input pin indicates a source, the value of the TriggerDevice input pin indicates the device which triggers the capture of a value, the value of the InitialValue input pin is an initial value, and the value of the CapValue output pin is the value captured.

2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Channel	Capturer number	WORD	0~7	The value of the Channel input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	The motion control function block is enabled when there is a transition in the Enable input pin's signal from low to high.	BOOL	True/False	-
Source	Source	WORD	mcCapAxis1: 0 mcCapAxis2: 1 mcCapAxis3: 2 mcCapC200: 4 mcCapC204: 5	The value of the Source input pin is valid when there is a transition in the Enable input pin's signal from low to high.

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
TriggerDevice	Device which triggers the capture of a value	WORD	mcCapPG0: 0 mcCapMPGB0: 1 mcCapMPGA0: 2 mcCapLSN0: 3 mcCapLSP0: 4 mcCapDOG0: 5 mcCapStop0: 6 mcCapStart0: 7 mcCapPG1: 8 mcCapMPGB1: 9 mcCapMPGA1: 10 mcCapLSN1: 11 mcCapLSP1: 12 mcCapDOG1: 13 mcCapStop1: 14 mcCapStart1: 15	The value of the TriggerDevice input pin is valid when there is a transition in the Enable input pin's signal from low to high.
InitialValue	Initial value	DWORD	K-2,147,483,648~ K2,147,483,647	The value of the InitialValue input pin is valid when there is a transition in the Enable input pin's signal from low to high.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An output value is valid.	BOOL	● There is a transition in the Valid output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Valid output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The source specified has been occupied.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

Value output pin				
Name	**Function**	**Data type**	**Output range**	**Time when a value is valid**
CapValue	Value which is captured	DWORD	K-2,147,483,648~ K2,147,483,647	When the motion control function block is executed, the value of the CapValue output pin is updated repeatedly. If there is no trigger, the value captured will remain unchanged.

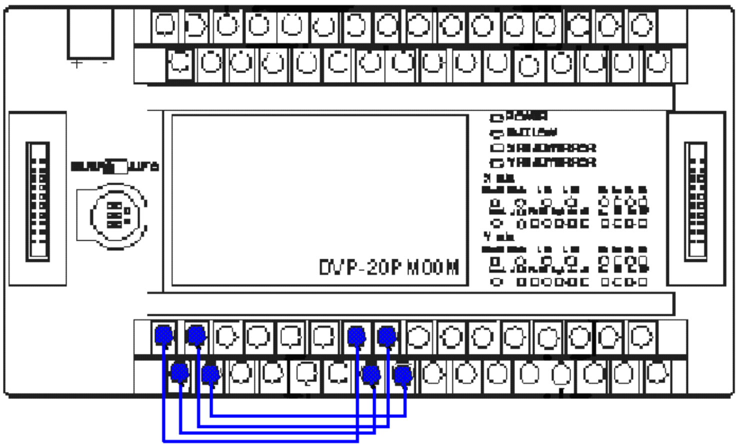
3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.
The capturer specified has been used.	Use another capturer.

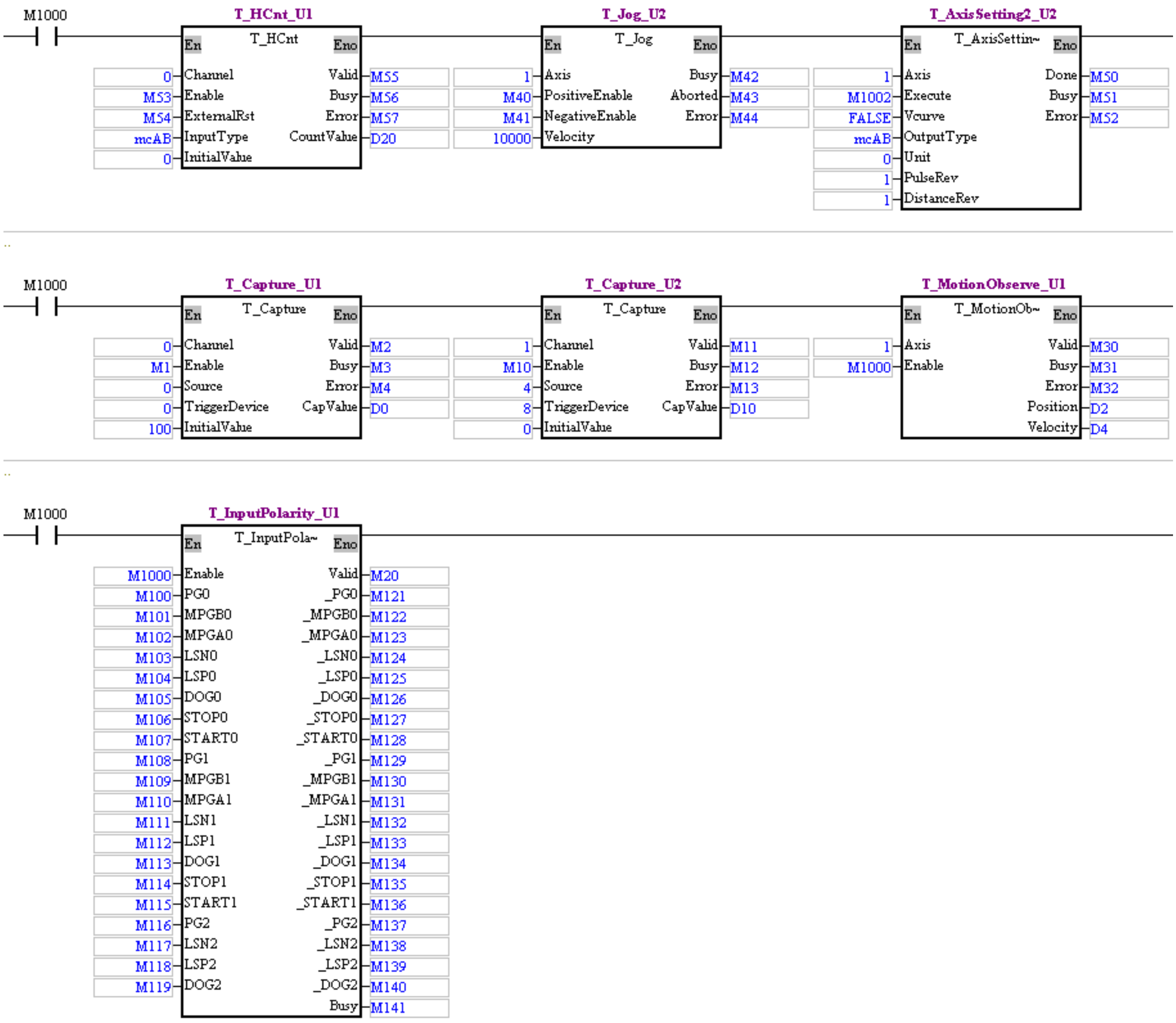
4. Example

Purpose:

Two high-speed capturers are used. If external terminals are turned ON, the present position of the first axis and the present value in C200 will be captured. The motion control function block T_InputPolarity is used to set the polarity of the external terminal which triggers the capture of the present position of the first axis, and the polarity of the external terminal which triggers the capture of the present value in C200. The external wiring required is shown below.



FP0 is connected to A0.
RP0 is connected to B0.



- The pulses output by the first axis are A/B-phase pulses. After the motion control function block T_MotionObserve is enabled, the present position of the first axis and the present speed of the first axis will be read.
- After M53 is set to ON, a high-speed counter will be started.
- After M1 is set to ON, high-speed capturer 0 will be started.

Setting high-speed capturer 0: If PG0 is turned ON, the present position of the first axis will be captured.

- After M10 is set to ON, high-speed capturer 1 will be started.

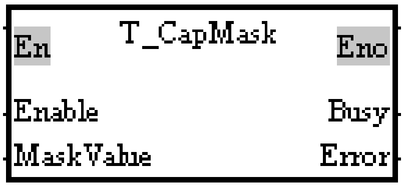
Setting high-speed capturer 1: If PG1 is turned ON, the present value in C200 will be captured.

- After M40 is set to ON, the positive JOG motion of the first axis will be started.
- If M100 is turned ON, PG0 will become a normally-closed contact, there will be a transition in PG0's signal from low to high, and the value in D0 will change.
- If M108 is turned ON, PG1 will become a normally-closed contact, there will be a transition in PG1's signal from low to high, and the value in D10 will change.

5. Modules which are supported

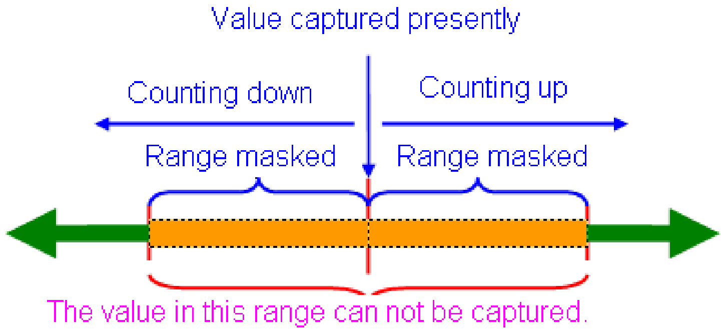
The motion control function block T_Capture supports DVP20PM00D and DVP20PM00M.

5.12.5 High-speed Masking



1. Motion control function block

   The motion control function block T_CapMask is used to start high-speed masking. The MaskValue input pin determines the range which will be masked. After high-speed masking is started, if the relative difference between the value captured this time and the value captured last time is in the range which can be masked, the signal which triggers the capture of the value this time will be disregarded.



2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
Enable	The motion control function block is enabled when there is a transition in the Enable input pin's signal from low to high.	BOOL	True/False	-
MaskValue	Range which is masked	DWORD	K0~2,147,483,647	When the motion control function block is executed, the value of the MaskValue input pin is updated repeatedly.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	• There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	• There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. • There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

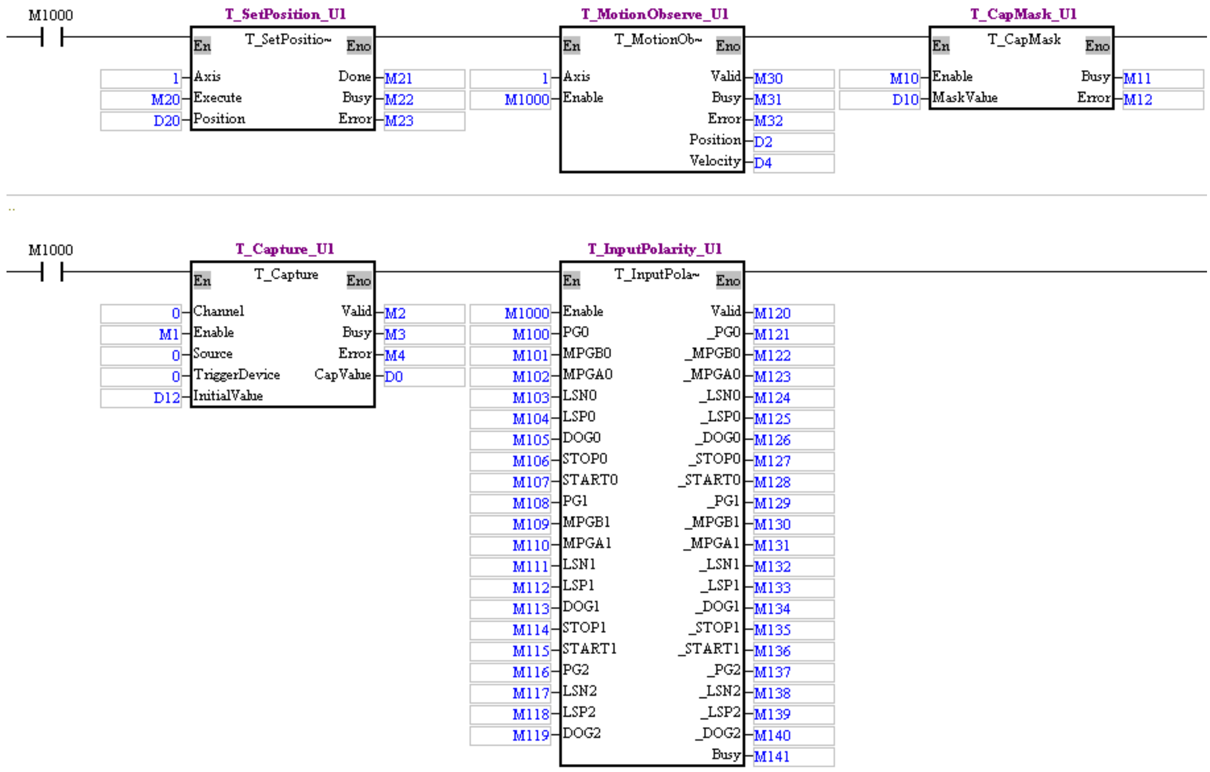
State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The source specified has been occupied.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

3. Troubleshooting

Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Example

A high-speed capturer and the motion control function block T_CapMask are used. If the present position of the first axis is in the range (the value of the CapValue output pin±the value of the MaskValue input pin) which is masked, it will not be captured after an external device is set to ON.



- After the program is executed, the present position of the first axis and the present speed of the first axis will be read.
- After the value in D20 is set to 0, and M20 is turned ON, the first axis will output 0 pulses
- Set the value in D12 to 100. After M1 is turned ON, high-speed capturer 0 will be started.

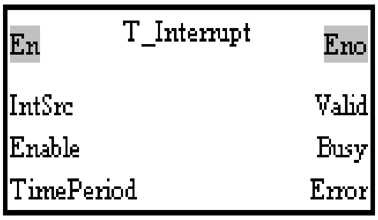
Setting high-speed capturer 0: If PG0 is turned ON, the present position of the first axis will be captured.

- Set the value in D10 to 500. After M10 is turned ON, the high-speed masking specified will be started.
- After M100 is set to ON, there will be a transition in PG0's signal from low to high, and the value of the CapValue output pin will still be 100.
- After the value in D20 is set to 500, and M20 is turned ON. The first axis will output 500 pulses.
- After M100 is set to ON, there will be a transition in PG0's signal from low to high, and the value of the CapValue output pin will still be 100.
- After the value in D20 is set to 600, and M20 is turned ON. The first axis will output 600 pulses.
- After M100 is set to ON, there will be a transition in PG0's signal from low to high, and the value of

the CapValue output pin will become 600.

5. Modules which are supported
The motion control function block T_CapMask supports DVP20PM00D and DVP20PM00M.

5.12.6 Setting an Interrupt



1. Motion control function block
The motion control function block T_Interrupt is used to set the trigger for an interrupt subroutine. The value of the IntSCR input pin indicates the trigger for an interrupt subroutine. If the interrupt set is a time interrupt, the value of the TimePeriod input pin indicates the cycle of the interrupt.
2. Input pins/Output pins

Input pin				
Name	**Function**	**Data type**	**Setting value**	**Time when a value is valid**
IntSCR	Setting an interrupt	WORD	IntTimer: 0 IntStart0: 1 IntStop0: 2 IntStart1: 3 IntStop1: 4	The value of the IntSrc input pin is valid when there is a transition in the Enable input pin's signal from low to high.
Enable	The motion control function block is enabled when there is a transition in the Enable input pin's signal from low to high.	BOOL	True/False	-
TimePeriod	Cycle of a time interrupt (Unit: ms) (Not applicable to terminal interrupts)	WORD	K1~K65,535	When the motion control function block is executed, the value of the TimePeriod input pin is updated repeatedly.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Valid	An Interrupt is enabled.	BOOL	• There is a transition in the Valid output pin's signal from low to high when an interrupt is enabled.	• There is a transition in the Valid output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

State output pin				
Name	**Function**	**Data type**	**Time when there is a transition in an output pin's signal from low to high**	**Time when there is a transition in an output pin's signal from high to low**
Busy	The motion control function block is being executed.	BOOL	● There is a transition in the Busy output pin's signal from low to high when there is a transition in the Enable input pin's signal from low to high.	● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Error output pin's signal from low to high. ● There is a transition in the Busy output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.
Error	An error occurs in the motion control function block.	BOOL	● Input values are incorrect. ● The source specified has been occupied.	● There is a transition in the Error output pin's signal from high to low when there is a transition in the Enable input pin's signal from high to low.

3. Troubleshooting

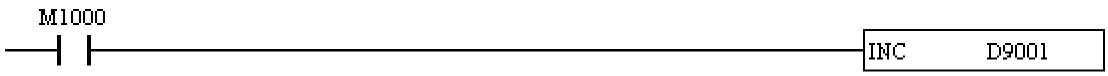
Error	Troubleshooting
The values of input pins in the motion control function block are incorrect.	Check whether the values of the input pins are in the ranges allowed.

4. Example

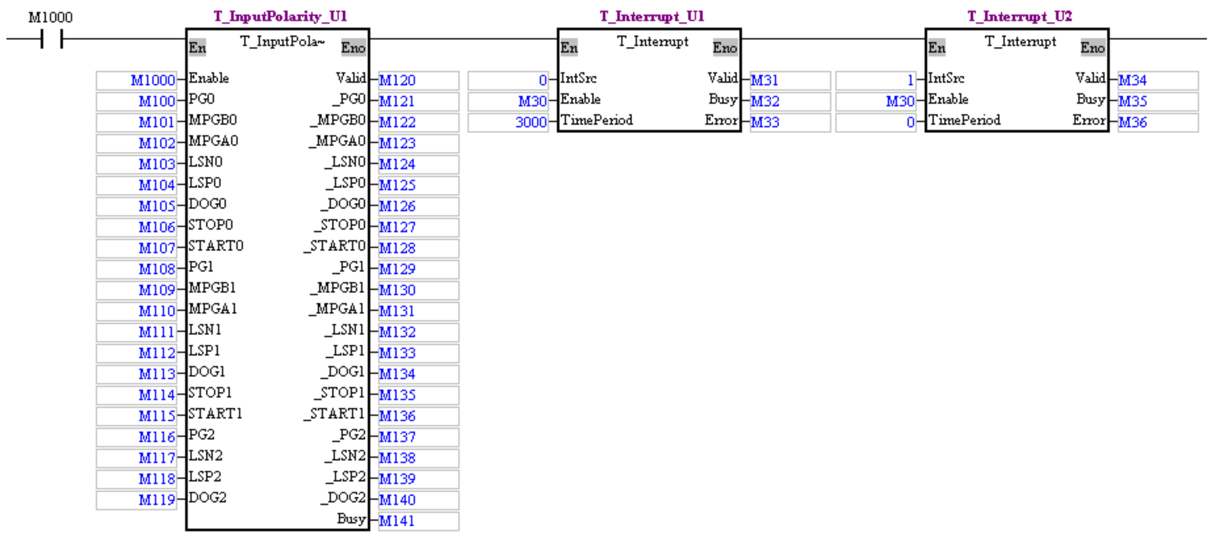
A time interrupt and an external interrupt are used. Users can use the motion control function block T_InputPolarity to simulate the state of an external terminal.

I0:

I1:



O100:



- After M1000 is set to ON, the motion control function block T_InputPolarity will be started.
- After M30 is set to ON, the time interrupt I0 and the external interrupt I1 will be started.
- After the time interrupt I0 is started, it will be executed every three seconds, and the value in D9000 will increase by one every three seconds.
- After the external interrupt I1 is started, the users can simulate the state of START0 by setting M107. If M107 is turned from OFF to ON, the value in D9001 will increase by one.

5. Modules which are supported
   The motion control function block T_Interrupt supports DVP20PM00D and DVP20PM00M.

6.1 Table of Motion Instructions and Table of G-codes

📖 Motion instructions

Type	Motion instruction number	Instruction code	Function	Applicable mode		Response time	Page number
				20D	20M		
Motion instructions	00	DRV	Rapid positioning (two axes)	✓	✓	20~25 ms	6-7
			Rapid positioning (three axes)		✓	20~25 ms	6-7
	01	LIN	Linear interpolation (two axes) (The remaining distance can be considered.)	✓	✓	20~22 ms	6-10
			Linear interpolation (three axes) (The remaining distance can be considered.)		✓	20~25 ms	6-10
	02	CW	Circular interpolation, clockwise (arc center) (The remaining distance can be considered.)	✓	✓	20~24 ms	6-13
			Helical interpolation, clockwise (arc center) (The remaining distance can be considered.)		✓	20~25 ms	6-13
	03	CCW	Circular interpolation, counterclockwise (arc center) (The remaining distance can be considered.)	✓	✓	20~24 ms	6-13
			Helical interpolation, counterclockwise (arc center) (The remaining distance can be considered.)		✓	20~25 ms	6-13
	04	CW	Circular interpolation, clockwise (radius) (The remaining distance can be considered.)	✓	✓	20~24 ms	6-19
			Helical interpolation, clockwise (radius) (The remaining distance can be considered.)		✓	20~25 ms	6-19
	05	CCW	Circular interpolation, counterclockwise (radius) (The remaining distance can be considered.)	✓	✓	20~24 ms	6-19
			Helical interpolation, counterclockwise (radius) (The remaining distance can be considered.)		✓	20~25 ms	6-19
	06	TIM	Dwell	✓	✓	-	6-23
	07	DRVZ	Returning home	✓	✓	20~25 ms	6-24
	08	SETR	Setting electrical zero	✓	✓	-	6-27
	09	DRVR	Returning to electrical zero	✓	✓	20~25 ms	6-28
	10	INTR	Two-axis single-Speed linear interpolation (The remaining distance is ignored.)	✓	✓	20~25 ms	6-29
	11	SINTR	Inserting a single speed	✓	✓	20~25 ms	6-30
	12	DINTR	Inserting two speeds	✓	✓	20~25 ms	6-32
	13	MOVC	Setting the offset of linear movement	✓	✓	-	6-34
	14	CNTC	Setting an offset for the center of an arc	✓	✓	-	6-35
	15	RADC	Setting an offset for a radius	✓	✓	-	6-36
	16	CANC	Canceling compensation	✓	✓	-	6-37
	17	ABST	Absolute programming	✓	✓	-	6-38
	18	INCT	Incremental programming	✓	✓	-	6-38
	19	SETT	Setting a present position	✓	✓	-	6-39

Additional remark: 20D=DVP20PM00D; 20M=DVP20PM00M

📖 O pointers/M-codes

Instruction code	Function	Description	Applicable model		Page number
			20D	20M	
O	Program pointer	Main program: O100 Motion subroutines: Ox0~Ox99	✓	✓	6-40
M	M-code	M0~M65535 M0~M65535 M102: End of the main program O100 M2: End of a motion subroutine	✓	✓	6-41

Additional remark: 20D=DVP20PM00D; 20M=DVP20PM00M

📖 G-codes

Type	G-code	Motion instruction code	Function	Applicable model		Page number
				20D	20M	
G-code instructions	00	DRV	Rapid positioning (two axes)	✓	✓	6-47
	00	DRV	Rapid positioning (three axes)		✓	6-45
	01	LIN	Linear interpolation (two axes) (The remaining distance can be considered.)	✓	✓	6-49
	01	LIN	Linear interpolation (three axes) (The remaining distance can be considered.)		✓	6-48
	02	CW	Circular interpolation, clockwise (arc center) (The remaining distance can be considered.)	✓	✓	6-49
	02	CW	Helical interpolation, clockwise (arc center) (The remaining distance can be considered.)		✓	6-49
	03	CCW	Circular interpolation, counterclockwise (arc center) (The remaining distance can be considered.)	✓	✓	6-49
	03	CCW	Helical interpolation, counterclockwise (arc center) (The remaining distance can be considered.)		✓	6-49
	02	CW	Circular interpolation, clockwise (radius) (The remaining distance can be considered.)	✓	✓	6-50
	02	CW	Helical interpolation, clockwise (radius) (The remaining distance can be considered.)		✓	6-50
	03	CCW	Circular interpolation, counterclockwise (radius) (The remaining distance can be considered.)	✓	✓	6-50
	03	CCW	Helical interpolation, counterclockwise (radius) (The remaining distance can be considered.)		✓	6-50
	04	TIM	Dwell	✓	✓	6-51
	17	NULL	XY plane selection		✓	6-51
	18	NULL	ZX plane selection		✓	6-51
	19	NULL	YZ plane selection		✓	6-51
	90	ABST	Absolute programming	✓	✓	6-51
	91	INCT	Incremental programming	✓	✓	6-52

Additional remark: 20D=DVP20PM00D; 20M=DVP20PM00M

6.2 Composition of a Motion Instruction/G-code

6.2.1 Motion Instruction

◆ A motion instruction is composed of an instruction name and operands.

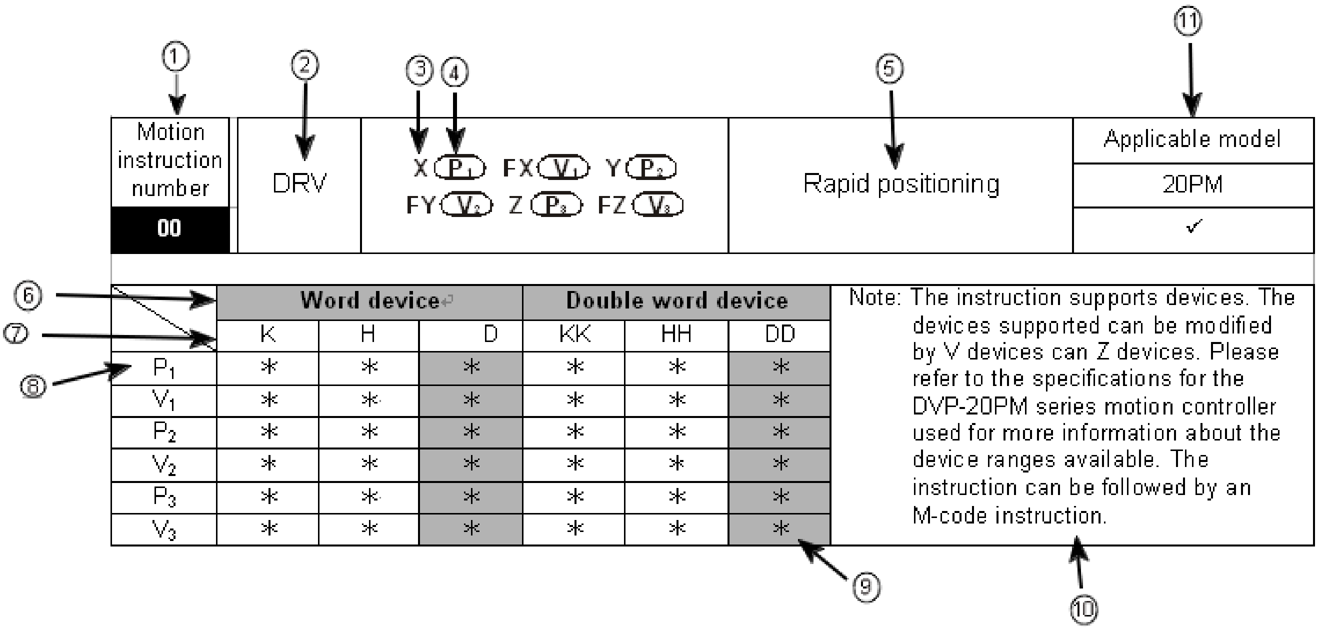
Instruction name		Function which is executed
Operand	**Indication of a function**	Parameter mark (X, Y, Z, F, FX, FY, FZ, R, I, J, K)
	Setting of a parameter	Value of a parameter

- Users must type parameter marks.
- Setting of a parameter
  1. If the value of a parameter is an arabic number or a 32-bit register, it must occupy 32 bits.
     Example: DRV X**1000** FX**1000** Y**1000** FY**DD1000**
  2. If the value of a parameter is composed of K/H/D and an arabic number, it must occupy 16 bits.
     Example: DRV X**K1000** FX**H1000** Y**K1000** FY**D1000**
  3. If the value of a parameter is an arabic number, and the value of another paramter is composed of K/H/D and an arabic number, they must occupy 16 bits and 32 bits respectively.
     Example: DRV X**1000** FX**HH1000** Y**K1000** FY**1000**

◆ Size of a motion instruction

- A motion ionstruciton in the program in DVP20PM00D occupies one step. The motion instructions DRV, LIN, CW, CCW, and SETT individaully occupy one step in the program in DVP20PM00M.
- If the value of a paramter is an arabic number, it will occupy three steps in a program.
- If the value of a parameter is composed of K/H/D/DD and an arabic number, it will occupy two steps in a program.
- If the value of a parameter is composed of KK/HH and an arabic number, it will occupy three steps in a program.

◆ Format of a motion instruction



① Motion instruction number
② Motion instruction name
③ Parameter mark
④ Value of a parameter

⑤ Description of the function of the motion instruction
⑥ Device types
⑦ Device names
⑧ The devices marked with ‘ * ’ can be used.
⑨ The devices marked with ‘ * ’ displayed in grayscale can be modified by V devices and Z devices.
⑩ Points for attention
⑪ Applicable model
The model marked with ‘✓’ supports the motion instruction

- ◆ Typing a motion instruction

Some motion instructions are composed of instruction names, e.g. DRVZ, SETR, and ABS. Most motion instructions are composed of instruction names and operands. No conditional contact precedes a motion instruction.

- ◆ Setting parameters

Before a motion instruction is executed, the maximum speed (V_{MAX}) at which the X-axis rotates, the maximum speed (V_{MAX}) at which the Y-axis rotates, and the maximum speed (V_{MAX}) at which the Z-axis rotates must be the same, the start-up speed (V_{BIAS}) at which the X-axis rotates, the start-up speed (V_{BIAS}) at which the Y-axis rotates, and the start-up speed (V_{BIAS}) at which the Z-axis rotates must be the same, the time (T_{ACC}) it takes for the X-axis to accelerate, the time (T_{ACC}) it takes for the Y-axis to accelerate, and the time (T_{ACC}) it takes for the Z-axis to accelerate must be the same, and the time (T_{DEC}) it takes for the X-axis to decelerate, the time (T_{DEC}) it takes for the Y-axis to decelerate, and the time (T_{DEC}) it takes for the Z-axis to decelerate must be the same.

Take DVP20PM00M for instance. Before DRV is executed, the maximum speed (V_{MAX}) at which the X-axis rotates, the maximum speed (V_{MAX}) at which the Y-axis rotates, and the maximum speed (V_{MAX}) at which the Z-axis rotates must be the same, the start-up speed (V_{BIAS}) at which the X-axis rotates, the start-up speed (V_{BIAS}) at which the Y-axis rotates, and the start-up speed (V_{BIAS}) at which the Z-axis rotates must be the same, the time (T_{ACC}) it takes for the X-axis to accelerate, the time (T_{ACC}) it takes for the Y-axis to accelerate, and the time (T_{ACC}) it takes for the Z-axis to accelerate must be the same, and the time (T_{DEC}) it takes for the X-axis to decelerate, the time (T_{DEC}) it takes for the Y-axis to decelerate, and the time (T_{DEC}) it takes for the Z-axis to decelerate must be the same.

	V_{MAX}	V_{BIAS}	T_{ACC}	T_{DEC}
X-axis	500,000	1,000	10	15
Y-axis	500,000	1,000	10	15
Z-axis	500,000	1,000	10	15

6.2.2 G-code Instruction

- ◆ A G-code instruction is composed of an instruction name and operands.

Instruction name		Function which is executed
Operand	Indication of a function	Parameter mark (X, Y, Z, I, J, K, R, F)
	Setting of a parameter	Value of a parameter

- Users must type parameter marks.
- Setting of a parameter: If the value of a paramter is an arabic integer or an arabic decimal, it must occupy 32 bits.

  Example 1: G00 X**100** Y**100**

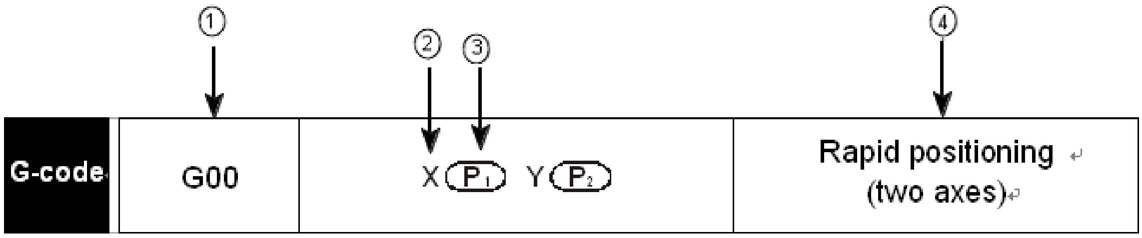
  Example 2: G00 X**100.0** Y**100.0**

- DVP20PM00M: The value of a parameter can be a 16-bit register or a 32-bit register.
  G0 XD11 YDD20 FDD25;
  XDD30 YD40;
  G1 X100.0 Y25.0 FD50;
  G02 XD60 Y50.0 ID100 JDD80;
  G03 YDD90 RD70 F300.0;
- If the value of a paramter is wrriten is a decimal, it is equal to 1,000 times the value..
  Example: G00 X**100.0** Y**100.0** is equal to G00 X**100000** Y**100000**.

◆ Size of a G-code

- A G-code in the porgram in DVP20PM00D occupies one step. G00, G01, G02, and G03 individaully occupy two steps in the program in DVP20PM00M, and the other G-code individaully occupy one step in the program in DVP20PM00M.
- If the value of a parameter is an arabic number, it will occupy three steps in a program.

◆ Format of a G-code instruction



① G-code instruction number
② Parameter mark
③ Value of a parameter
④ Description of the function of the G-code

◆ Typing a G-code instruction

Some G-code instructions are composed of instruction names, e.g. G90 and G91. Most G-code instructions are composed of instruction names and operands. No conditional contact precedes a G-code.

◆ Usage of a G-code

(a) Users can put several functions in a line.
Example: G91G01 X100.0 Y300.0 F500.0 M8 G04 X4.5;

(b) If G00, G01, G02, and G03 are in the same line, the last G-code will be executed.
Example: G02 G00 G03 G01 X100.0 Y300.0 F500.0; => G01 X100.0 Y300.0 F500.0;

(c) If G00 is used, users do not have to set the maximum speed (V_{MAX}) at which an axis rotates.
Example: G00 X100.2 Y500.0;
The maximum speeds at which the axes move are the maximum speeds set in the motion controller used.

(d) G00 and G01 can be extended to the next line.
N0000 G00 X500.0 Y125.0;
N0001 X-400.0 Y-500.0; =>G00 X-400.0 Y-500.0;
N0002 G01 X100.0 Y25.0 F200.0;
N0003 X-200.0 Y50.0; =>G01 X-200.0 Y50.0 F200.0;

(e) The speed parameter F for G01/G02/G03 can be extended to the next line.
N0000 G01 X500.0 Y125.0 F200.0;
N0001 G03 X-40.0 Y-50.0 R100.0; =>G03 X-40.0 Y-50.0 R100. F200.0;
N0002 G02 X100.0 Y25.0 I400.5 F200.0;
N0003 G01 X-200.0 Y50.0; =>G01 X-200.0 Y50.0 F200.0;

(f) G90 and G91 have high priority over the other G-codes.

G90 G01 X100.0 Y300.0 F500.0;	=>G90 G01 X100.0 Y300.0 F500.0;
G01G90 X100.0 Y300.0 F500.0;	=>G90 G01 X100.0 Y300.0 F500.0;

(g) Whether there are spaces in a program code, the program code can be identified.

G01G91X500.0 Y125.0F200.0;	=>G01 G91 X500.0 Y125.0 F200.0;

(h) Coordinates and speeds are converted into 32-bit values.

G01 X-125.5 F200.0;	=>G01 X-125500 F200000;

(i) If the value of a parameter is a decimal, it will be converted into an integer after it is multiplied by 1000.

G01 X100 Y-125.5 F200.0;	=>G01 X100 Y-125500 F200000;

(j) Ten milliseconds are a unit of measurement for dwell duration. In the example below, the system used automatically ignores 9 in P2509. (Parameter X for G04: A second is a unit of measurement for dwell duration. Parameter P for G04: A millisecond is a unit of measurement for dwell duration.)

G04 X4.5 (Dwell duration: 4.5 seconds)
G04 X5 (Dwell duration: 5 seconds)
G04 P4500 (Dwell duration: 4.5 seconds)
G04 P2509 (Dwell duration: 2.5 seconds)

(k) The G-codes not supported are ignored and not read.

G21G54G01 X-125.5 F200.0;	=>G01 X-125500 F200000;
G43G87G96 X250.5 F200.0;	=> G01 X250500 F200000;

(l) The writing of instructions conforms to the writing of general G-codes. Users can arrange the parameter marks used in any order.

G0 X4.5 Z40.0 Y30.5 F200.2;	=>G00 X4.500 Y30.500 Z40.000 F200.200
Z100.5Y400.0X300.0 ;	=>G00 X300.000 Y400.000 Z100.500
G1xd100zd300y200.45 fd400	=>G01 XD100 Y200.450 ZD300 FD400
G3 ZD100 I200.0F50.60XD300 m80	=>G03 XD300 ZD100 I200.000 F50.600 M80
G03 yD100 x9999.9Z200.0r777.7 Fd800	=>G03 X9999.900 YD100 Z200.000 R777.700 FD800

◆ Setting parameters

Before a G-code is executed, the maximum speed (V_{MAX}) at which the X-axis rotates, the maximum speed (V_{MAX}) at which the Y-axis rotates, and the maximum speed (V_{MAX}) at which the Z-axis rotates must be the same, the start-up speed (V_{BIAS}) at which the X-axis rotates, the start-up speed (V_{BIAS}) at which the Y-axis rotates, and the start-up speed (V_{BIAS}) at which the Z-axis rotates must be the same, the time (T_{ACC}) it takes for the X-axis to accelerate, the time (T_{ACC}) it takes for the Y-axis to accelerate, and the time (T_{ACC}) it takes for the Z-axis to accelerate must be the same, and the time (T_{DEC}) it takes for the X-axis to decelerate, the time (T_{DEC}) it takes for the Y-axis to decelerate, and the time (T_{DEC}) it takes for the Z-axis to decelerate must be the same.

Take DVP20PM00M for instance. Before an Ox motion subroutine (containing G-code) is executed, the maximum speed (V_{MAX}) at which the X-axis rotates, the maximum speed (V_{MAX}) at which the Y-axis rotates, and the maximum speed (V_{MAX}) at which the Z-axis rotates must be the same, the start-up speed (V_{BIAS}) at which the X-axis rotates, the start-up speed (V_{BIAS}) at which the Y-axis rotates, and the start-up speed (V_{BIAS}) at which the Z-axis rotates must be the same, the time (T_{ACC}) it takes for the X-axis to accelerate, the time (T_{ACC}) it takes for the Y-axis to accelerate, and the time (T_{ACC}) it takes for the Z-axis to accelerate must be the same, and the time (T_{DEC}) it takes for the X-axis to decelerate, the time (T_{DEC}) it takes for the Y-axis to decelerate, and the time (T_{DEC}) it takes for the Z-axis to decelerate must be the same.

	V_{MAX}	V_{BIAS}	T_{ACC}	T_{DEC}
X-axis	500,000	1,000	10	15
Y-axis	500,000	1,000	10	15
Z-axis	500,000	1,000	10	15

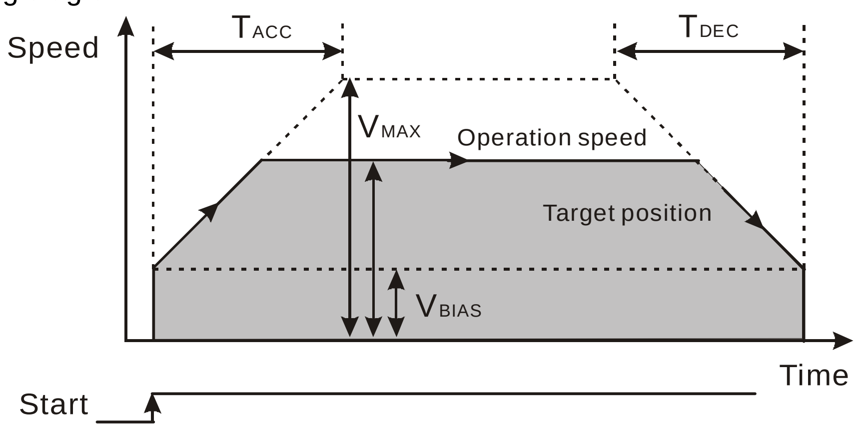
6.3 Descriptions of Motion Instructions

Motion instruction number	DRV	X P_1 FX V_1 Y P_2 FY V_2 Z P_3 FZ V_3	Rapid positioning	Applicable model
				20PM
00				✓

	Word device			Double word device			Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. Please refer to the specifications for the DVP-20PM series motion controller used for more information about the device ranges available. The instruction can be followed by an M-code instruction.
	K	H	D	KK	HH	DD	
P_1	*	*	*	*	*	*	
V_1	*	*	*	*	*	*	
P_2	*	*	*	*	*	*	
V_2	*	*	*	*	*	*	
P_3	*	*	*	*	*	*	
V_3	*	*	*	*	*	*	

Explanation

- **P_1**: Target position of the X-axis; **V_1**: Speed at which the X-axis moves; **P_2**: Target position of the Y-axis; **V_2**: Speed at which the Y-axis moves; **P_3**: Target position of the Z-axis; **V_3**: Speed at which the Z-axis moves
- Maximum **V_1**, **V_2**, **V_3** = 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 in special data registers.
- 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 users set the moving speed on an axis, they have to set the target position on the axis. However, if they set the target position, it is not a must to set the moving speed.

- DVP20OM00D supports two-axis (X-axis/Y-axis) high-speed positioning, and therefore there are 8 operand combinations for DRV.

No.	Motion instruction	Combination of operands
1		X P_1
2		X P_1 FX V_1
3		Y P_2
4	DRV	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

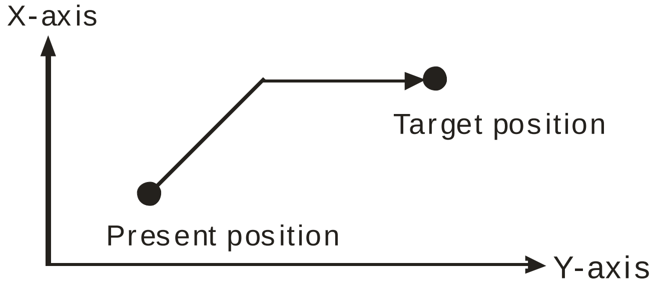
- DVP20PM00M supports 3-axis (X-axis/Y-axis/Z-axis) high-speed positioning, and therefore there are 26 operand combinations for DRV.

No.	Motion instruction	Combination of operands
1		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	DRV	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

- If users set the target position on the axis without setting the moving speed, the operation will run at V_{MAX}.

Example

- **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.
- Moving path



- Other valid combinations of operands:
  DRV XKK-345289 FXD100 YDD10Z5 FYDD102
  DRV XDD20 FXHH2345 YKK456 FYDD0 Z567 Fz234
  The above instructions are legal, and device D is the register storing the value to be set.

Additional remark

- Relevant special data registers

D1822, D1823	Maximum speed (V_{MAX}) at which the X-axis rotates
D1824, D1825	Start-up speed (V_{BIAS}) at which the X-axis rotates
D1836	Time (T_{ACC}) it takes for the X-axis to accelerate
D1837	Time (T_{DEC}) it takes for the X-axis to decelerate
D1902, D1903	Maximum speed (V_{MAX}) at which the Y-axis rotates
D1904, D1905	Start-up speed (V_{BIAS}) at which the Y-axis rotates
D1916	Time (T_{ACC}) it takes for the Y-axis to accelerate
D1917	Time (T_{DEC}) it takes for the Y-axis to decelerate
D1982, D1983	Maximum speed (V_{MAX}) at which the Z-axis rotates
D1984, D1985	Start-up speed (V_{BIAS}) at which the Z-axis rotates
D1996	Time (T_{ACC}) it takes for the Z-axis to accelerate
D1997	Time (T_{DEC}) it takes for the Z-axis to decelerate

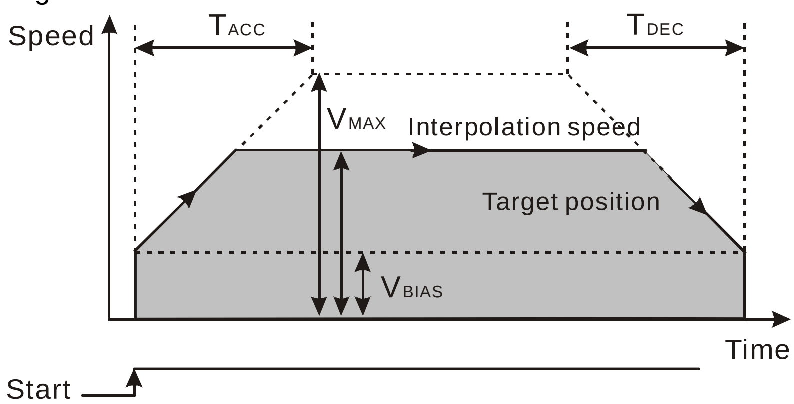
Motion instruction number	LIN	X (P1) Y (P2) Z (P3) F (V)	Linear interpolation (The remaining distance can be considered.)	Applicable model
01				20PM
				✓

	Word device			Double word device		
	K	H	D	KK	HH	DD
P_1	*	*	*	*	*	*
P_2	*	*	*	*	*	*
P_3	*	*	*	*	*	*
F	*	*	*	*	*	*

Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. Please refer to the specifications for the DVP-20PM series motion controller used for more information about the device ranges available. The instruction can be followed by an M-code instruction.

Explanation

- **P_1**: Target position of the X-axis; **P_2**: Target position of the Y-axis; **P_3**: Target position of the Z-axis; **V**: Speed of linear interpolation
- 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 in special data registers.
- Acceleration/deceleration time increases or decreases in proportional to the setting of V_{MAX}.
- Timing diagram



- The interpolation speed can be monitored by special data registers: D1850~D1851 are for the X-axis. D1930~D1931 are for the Y-axis. D2010~D2011 are for the Z-axis.
- D1865 is for setting a stop mode with the consideration on the remaining distance. (Please refer to the additional remark below 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.

- DVP20PM00 only supports two-axis (X-axis/Y-axis) synchronous interpolation positioning, and therefore there are 6 operand combinations for LIN instruction.

No.	Motion instruction	Combination of operands
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

- DVP20PM00M supports 3-axis (X, Y or Z) high-speed positioning, and therefore there are 14 operand combinations for LIN instruction.

No.	Motion instruction	Combination of operands
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

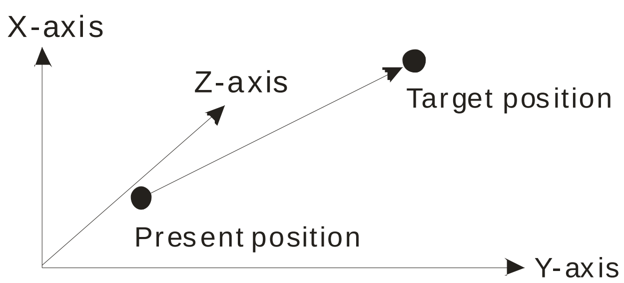
- If users set the target position on the axis without setting the moving speed, the operation will run at V_{MAX}.

Example

- **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 a speed of 40kHz.

- Moving path



Additional remark

- Relevant special data registers: Please refer to chapter 3 for more information.

D1822, D1823	Maximum speed (V_{MAX}) at which the X-axis rotates
D1824, D1825	Start-up speed (V_{BIAS}) at which the X-axis rotates
D1836	Time (T_{ACC}) it takes for the X-axis to accelerate
D1837	Time (T_{DEC}) it takes for the X-axis to decelerate
D1865	Mode of stopping Ox0~Ox99 K1: The execution of Ox0~Ox99 will resume next time Ox0~Ox99 are started. (The remaining distance is considered.) K2: The next instruction will be executed next time Ox0~Ox99 are started. Others: Ox0~Ox99 are executed again.
D1902, D1903	Maximum speed (V_{MAX}) at which the Y-axis rotates
D1904, D1905	Start-up speed (V_{BIAS}) at which the Y-axis rotates
D1916	Time (T_{ACC}) it takes for the Y-axis to accelerate
D1917	Time (T_{DEC}) it takes for the Y-axis to decelerate
D1982, D1983	Maximum speed (V_{MAX}) at which the Z-axis rotates
D1984, D1985	Start-up speed (V_{BIAS}) at which the Z-axis rotates
D1996	Time (T_{ACC}) it takes for the Z-axis to accelerate
D1997	Time (T_{DEC}) it takes for the Z-axis to decelerate

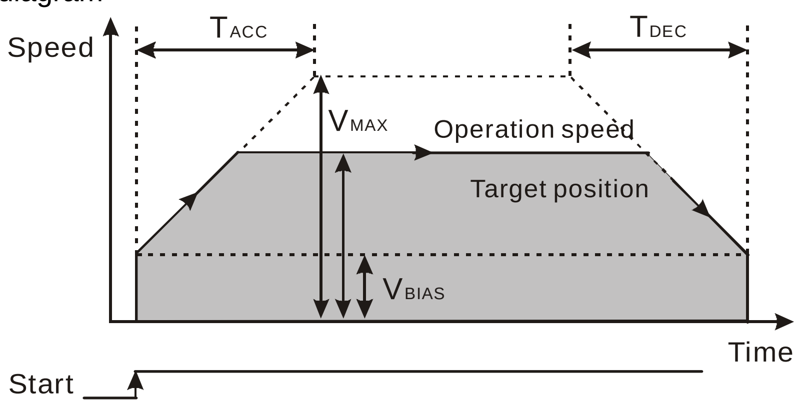
Motion instruction number	CW/CCW	X (P1) Y (P2) Z (P3) I (P4) J (P5) K (P6) F (V)	Clockwise circular/helical interpolation Counterclockwise circular/helical interpolation (arc center)	Applicable model
02 03				20PM
				✓

	Word device			Double word device		
	K	H	D	KK	HH	DD
P_1	*	*	*	*	*	*
P_2	*	*	*	*	*	*
P_3	*	*	*	*	*	*
P_4	*	*	*	*	*	*
P_5	*	*	*	*	*	*
P_6	*	*	*	*	*	*
V	*	*	*	*	*	*

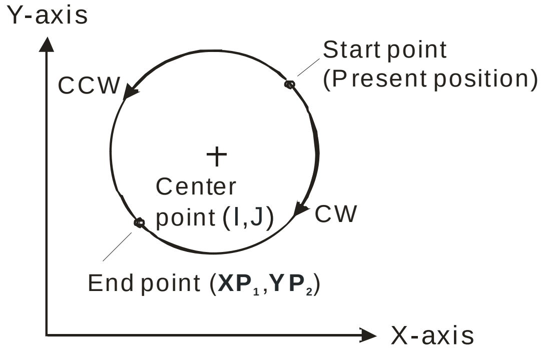
Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. The instruction can be followed by an M-code instruction.

Explanation

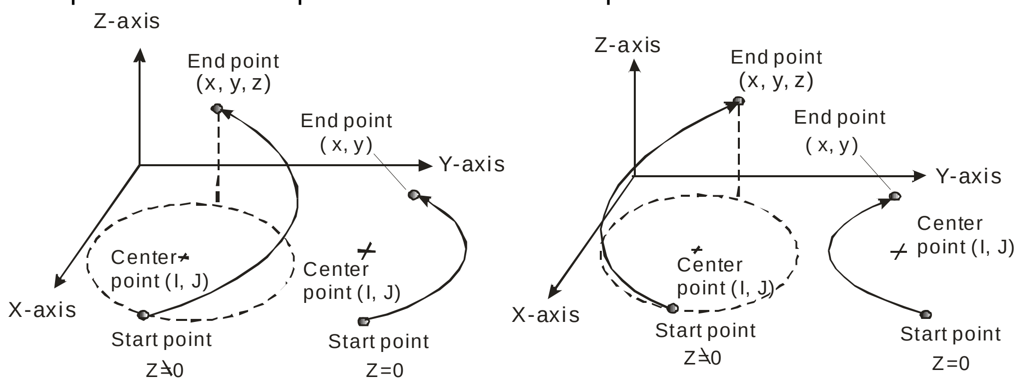
- **P_1**: Target position of the X-axis; **P_2**: Target position of the Y-axis; **P_3**: Target position of the Z-axis; **P_4**: Vector from the present position of the X-axis to an arc center; **P_5**: Vector from the present position of the Y-axis to an arc center; **P_6**: Vector from the present position of the Z-axis to an arc center; **V**: Speed of circular/helical interpolation
- **P_4**, **P_5** and **P_6**: Vectors from the present positions of an x-axis, a y-axis, and a z-axis to an arc center
- V_{MAX}=500 kHz
- Range of parameters **P_1**, **P_2**, **P_3**, **P_4**, **P_5**, **P_6**: (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 in special data registers.
- Acceleration/deceleration time increases or decreases in proportional to the setting of V_{MAX}.
- Timing diagram



- Two-axis arc interpolation: DVP20PM00D only supports circular interpolation on the XY plane. DVP20PM00M supports circular/helical interpolation on 3 planes by G-code instructions: G17 (XY plane selection), G18 (ZX plane selection) or G19 (YZ plane selection).

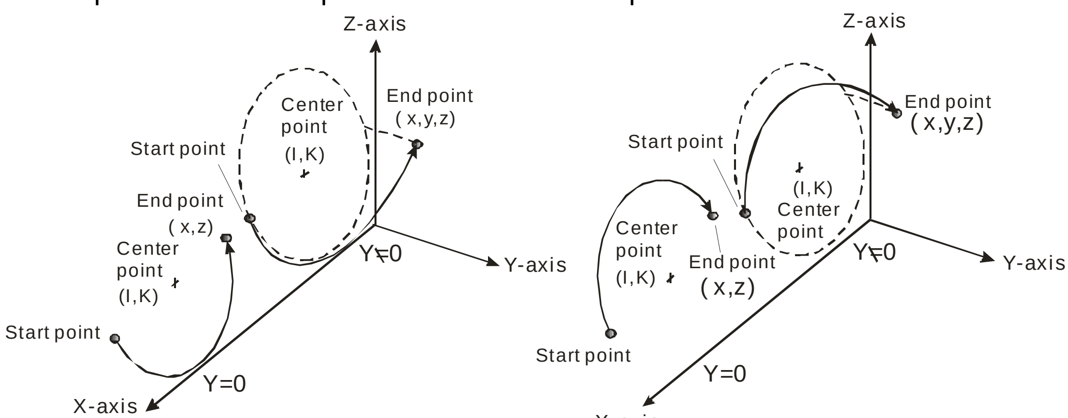


- Helical interpolation: Helical interpolation is applicable to DVP20PM00M. Three axes which are perpendicular to one another are used. They move synchronously. Helical interpolation is the extension of circular interpolation. If a helical interpolation instruction is used, and the change of height is zero, circular interpolation will be executed.
- If G17 is used, and the position of the Z-axis does not change, helical interpolation will be equivalent to circular interpolation.



CCW (Counterclockwise interpolation) CW (Clockwise interpolation)

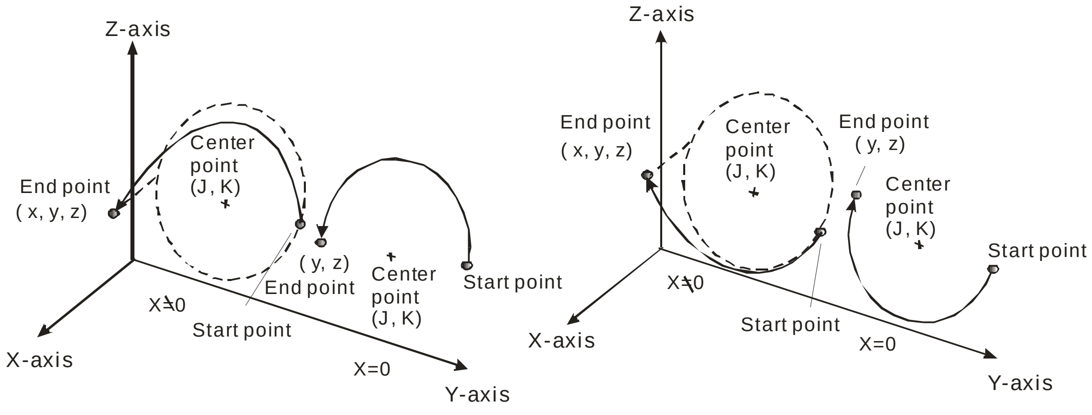
- If G18 is used, and the postion of the Y-axis does not change, helical interpolation will be equivalent to circular interpolation.



CW (Clockwise interpolation)

CCW (Counterclockwise interpolation)

- If G19 is used, and the postion of the X-axis does not change, helical interpolation will be equivalent to circular interpolation.



CCW (Counterclockwise interpolation) CW (Clockwise interpolation)

- 16-bit devices and 32-bit devices can be used together.
- Principles of writing an instruction: (1) Users have to specify a target position, and an arc center. They do not have to specify the speed of interpolation. (2) If there is no vector from the present position of an axis to its target position, users do not need to specify the target position of the axis. (3) If there is not vector form the present position of an axis to its arc center, users do not need to specify the arc center of the axis. The circular/helical instrctions listed below are available.
- DVP20PM00 only supports 2-axis (the XY plane) circular interpolation, and according to the programming rule, there are 18 operand combinations for CW/CCW.

No.	Motion instruction	Combination of operands
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

◆ DVP20PM00M supports 3-axis circular/helical interpolation, and according to the programming rule, there are 78 operand combinations for CW/CCW instruction.

No.	Motion instruction	Combination of operands	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		✓	

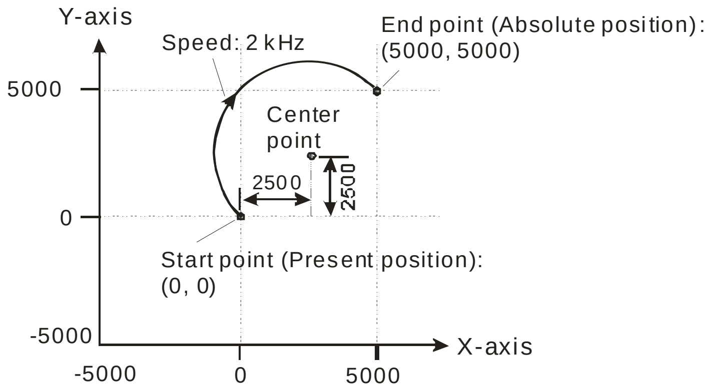
No.	Motion instruction	Combination of operands	G17	G18	G19
39	CW/CCW	X P_1 I P_3 K P_6		✓	
40		X P_1 I P_3 K P_6 F V		✓	
41		Z P_3 I P_3		✓	
42		Z P_3 I P_3 F V		✓	
43		Z P_3 K P_6		✓	✓
44		Z P_3 K P_6 F V		✓	✓
45		Z P_3 I P_3 K P_6		✓	
46		Z P_3 I P_3 K P_6 F V		✓	
47		X P_1 Z P_3 K P_6		✓	✓
48		X P_1 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			✓

No.	Motion instruction	Combination of operands	G17	G18	G19
76	CW/CCW	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			✓

- If users set the target position on the axis without setting the moving speed, the operation will run at V_{MAX}.
- The path of circular interpolation can be a 360° arc. The path of helical interpolation which is viewed from the top can be a full circle.

Example

- Example 1: Absolute coordinates are set, and CW is used. The arc start point set is (0, 0), the arc end point set is (5000, 5000), and the vector from the arc start point to the arc center point set is (2500, 2500). The output speed set is 2000 Hz.

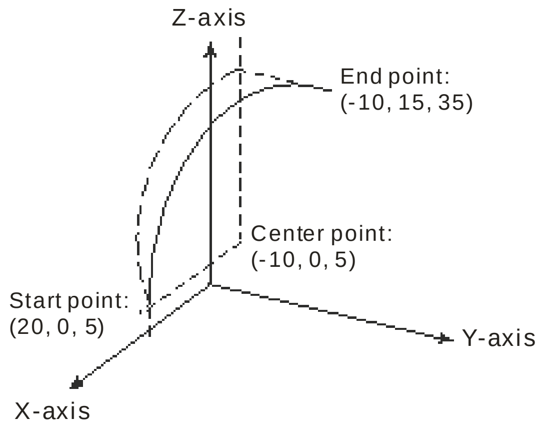


Program:

ABST;

CW XK10000 YK10000 IK2500 JK2500 FK2000;

- Absolute coordinates are set. G18 and CW are used. The arc end point set is (-10, 15, 35) and the arc cent point set is (-10, 0, 5). The output speed set is 2000 Hz.



Program:

ABS;

G18;

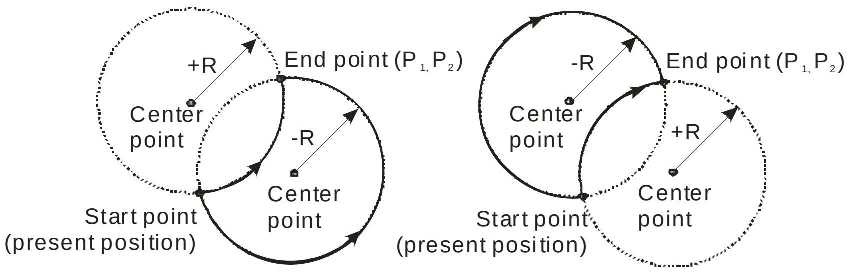
CW XK-10 YK15 ZK35 IK-30 JK0 (omissible) KK0 (omissible) FK2000;

Motion instruction number	CW/ CCW	X (P1) Y (P2) Z (P3) R (L) F (V)	Clockwise circular/helical interpolation Counterclockwise circular/helical interpolation (radius)	Applicable model
04 **05**				20PM
				✓

	Word device			Double word device			Note
	K	H	D	KK	HH	DD	Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. The instruction can be followed by an M-code instruction.
P_1	*	*	*	*	*	*	
P_2	*	*	*	*	*	*	
P_3	*	*	*	*	*	*	
L	*	*	*	*	*	*	
V	*	*	*	*	*	*	

Explanation

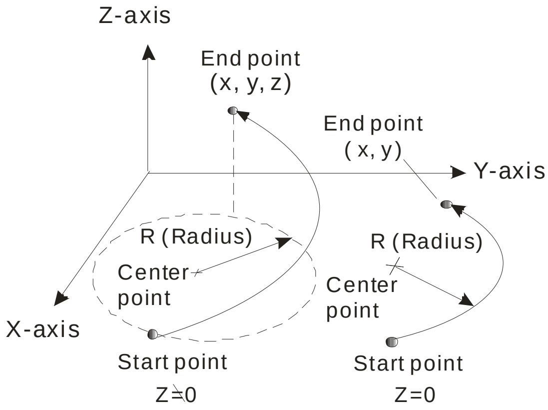
- **P_1**: Target position of the X-axis; **P_2**: target position of the Y-axis; **P_3**: target position of the Z-axis; **L**: Arc radius (If the angle subtended by an arc is less than 180°, the value of R is a positive value. If the angle subtended by an arc is greater than 180°, the value of R is a negative value.); **V**: Speed of circular/helical interpolation
- 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 in special data registers.
- Acceleration/deceleration time increases or decreases in proportional to the setting of V_{MAX}.
- 2-axis circular interpolation: DVP20PM00D only supports circular interpolation on the XY plane. DVP20PM00M supports circular/helical interpolation on 3 planes by G-code instructions: G17 (XY plane selection), G18 (ZX plane selection) or G19 (YZ plane selection).



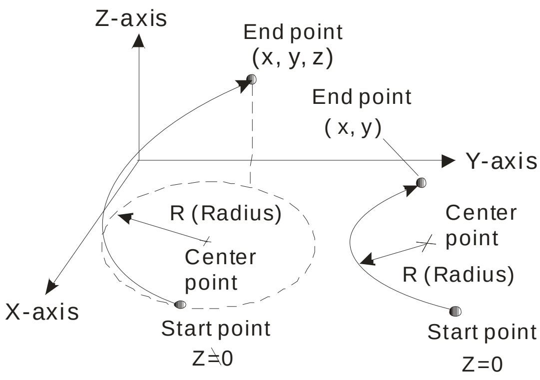
G03 (Counterclockwise interpolation) G02 (Clockwise interpolation)

- Helical interpolation: Helical interpolation is applicable to DVP20PM00M. Three axes which are perpendicular to one another are used. They move synchronously. Helical interpolation is the extension of circular interpolation. If a helical interpolation instruction is used, and the change of height is zero, circular interpolation will be executed.

- If G17 is used, and the position of the Z-axis does not change, helical interpolation will be equivalent to circular interpolation.

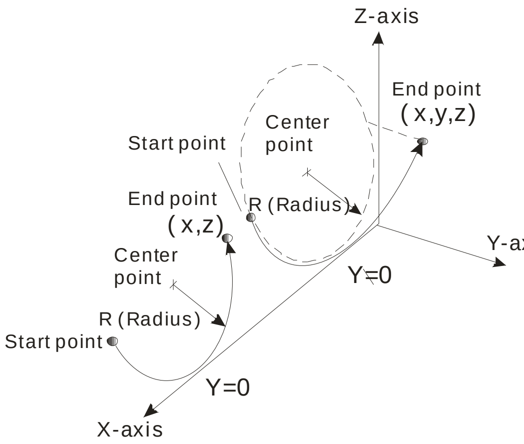


CCW (Counterclockwise interpolation)

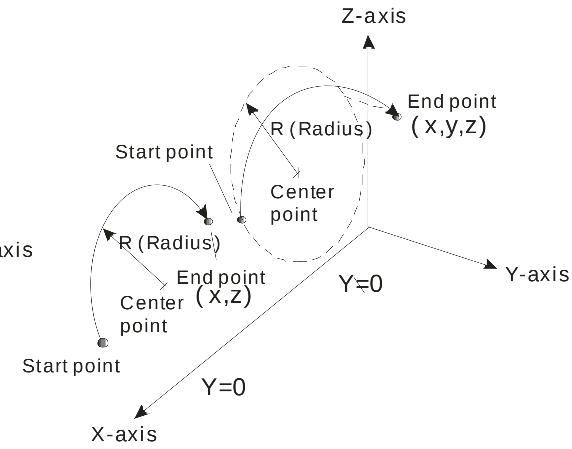


CW (Clockwise interpolation)

- If G18 is used, and the postion of the Y-axis does not change, helical interpolation will be equivalent to circular interpolation.

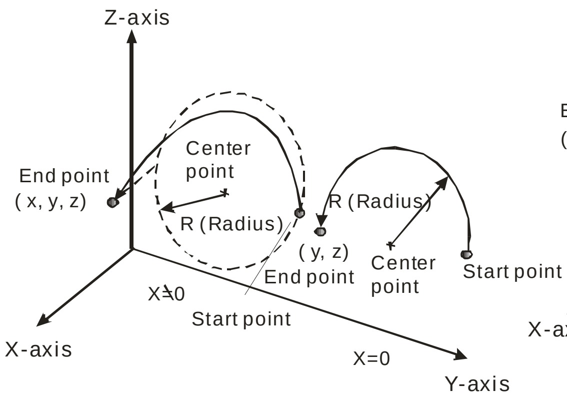


CCW (Counterclockwise interpolation)

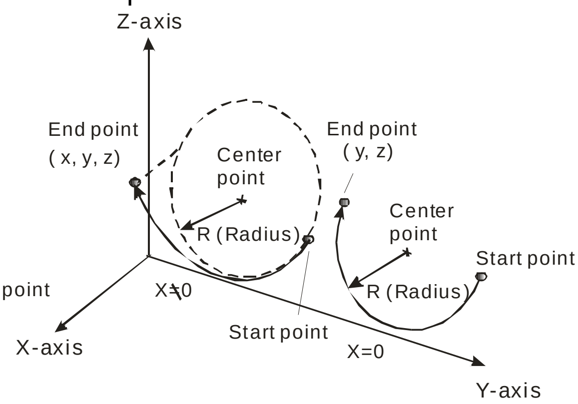


CW (Clockwise interpolation)

- If G19 is used, and the postion of the X-axis does not change, helical interpolation will be equivalent to circular interpolation.



CCW (Counterclockwise interpolation)



CW (Clockwise interpolation)

- 16-bit devices and 32-bit devices can be used together.
- Principles of writing an instruction: (1) Users have to specify a target position, and an arc center. They do not have to specify the speed of interpolation. (2) If there is no vector from the present position of an axis to its target position, users do not need to specify the target position of the axis. The circular/helical instrctions listed below are available.

- DVP20PM00D only supports 2-axis (the XY plane) circular interpolation, and according to the programming rule, there are 6 operand combinations for CW/CCW.

No.	Motion instruction	Combination of operands
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)

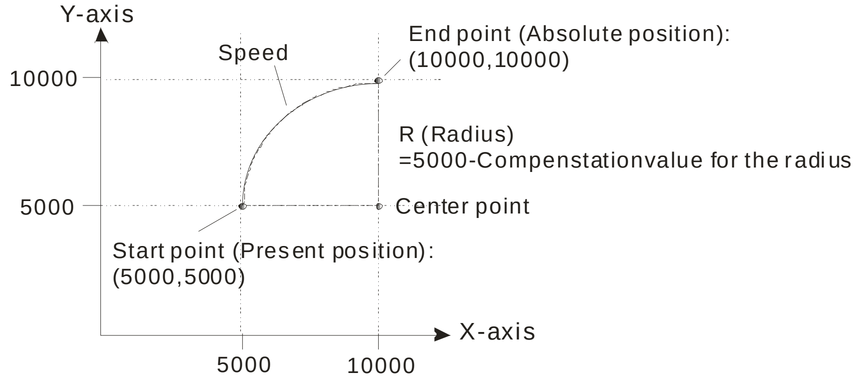
- DVP20PM00M supports 3-zxis circular/helical interpolation, and according to the programming rule, there are 14 operand combinations for CW/CCW instruction.

No.	Motion instruction	Combination of operands	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)		✓	✓

- If users set the target position on the axis without setting the moving speed, the operation will run at V_{MAX}.

Example

- Example 1: Absolute coordinates are set, and CW is used. The arc start point set is (5000, 5000), the arc end point set is (10000, 10000), and **L** is 5000. The angle subtended by the arc is less than 180°, and therefore the value of R is a positive value. The axes move at a speed of 1,000 per second.

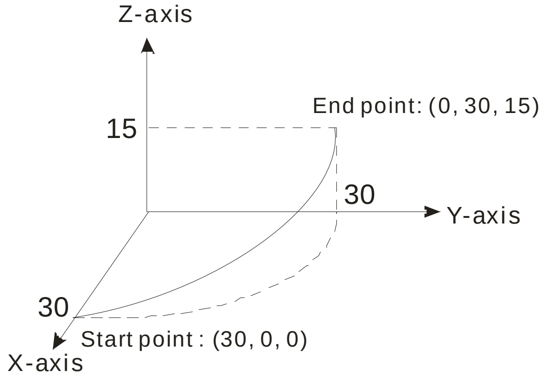


Program:

ABST;

CW XK10000 YK10000 RK5000 FK1000;

- Example 2: Absolute coordinates are set, and CCW is used. The arc start point set is (30, 0, 0), the arc end point set is (0, 30, 15), and **L** is 30.0. The angle subtended by the arc is less than 180°, and therefore the value of R is a positive value. The axes move at a speed of 100 per second.



Program:

ABST;

G17;

CCW XK0 YK30 ZK15 RK30 FK100;

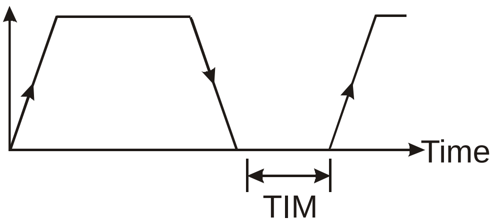
- Please refer to the explanation of RADC for more information about setting a compensation value for a radius.
- The path of circular interpolation can not be a 360° arc. The path of helical interpolation which is viewed from the top can not be a full circle.

Motion instruction number	TIM	(T)	Dwell	Applicable model
06				20PM
				✓

	Word device			Double word device			Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. Please refer to the specifications for the DVP-20PM series motion controller used for more information about the device ranges available. The instruction can be followed by an M-code instruction.
	K	H	D	KK	HH	DD	
T	*	*	*	*	*	*	

Explanation

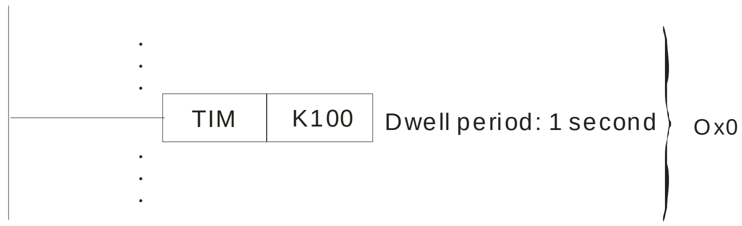
- **T**: Dwell time value (Unit: 10 milliseconds; **T**=K100: The dwell time set is 1 second.)
- A dwell period is a time interval between two instructions.



- 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

Example

- Program



Motion instruction number	DRVZ	No operand	Returning home	Applicable model
07				20PM
				✓

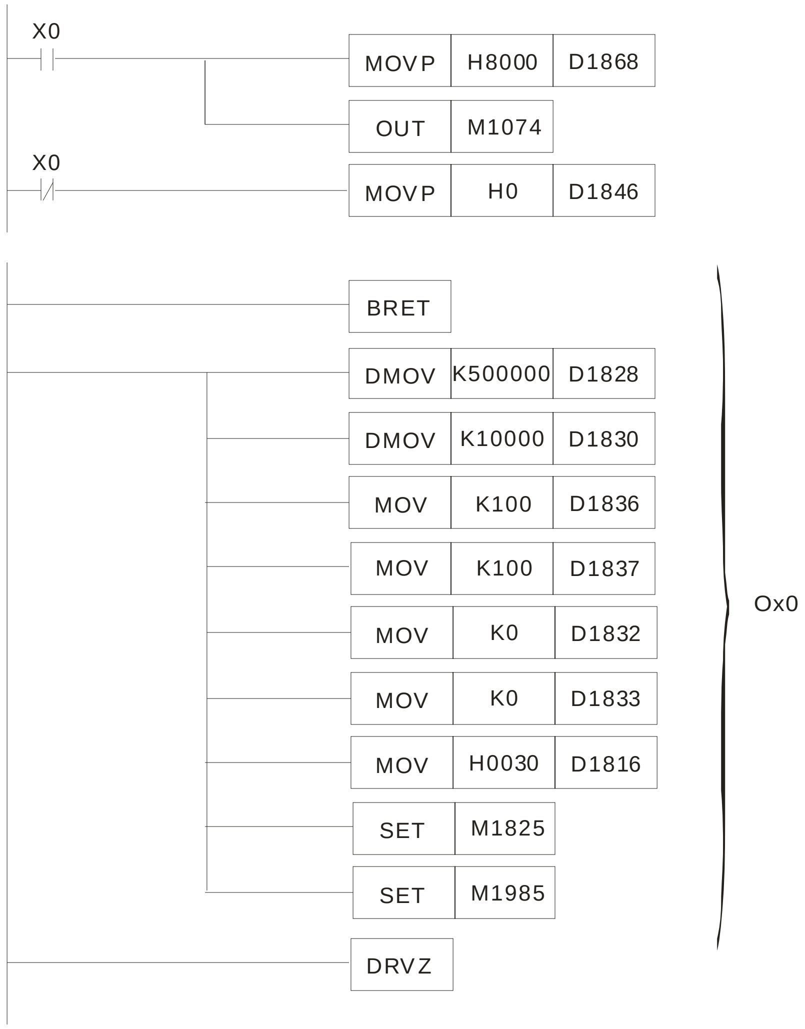
There is no operand. The instruction can be followed by an M-code instruction.

Explanation

- When DRVZ is executed, the X-axis, the Y-axis, and the Z-axis return home.
- Before enabling DRVZ, users have to first set the following parameters. (Please refer to the additional remark below for more information about special data registers.)
  1. V_{RT}: Speed (V_{RT}) at which an axis returns home
     V_{RT} cannot be modified during the execution.
     Range: 0~500 kHz
     Limitation: $V_{MAX}>V_{RT}>V_{BIAS}$
  2. Speed (V_{CR}) to which the speed of an axis decreases when the axis returns home: When a DOG signal is triggered, speed of the axis specified will decrease to the V_{CR} set. In order to accurately stop at its home, it is suggested the V_{CR} set should be a low speed.
     Range: 0~500 kHz
     Limitation: $V_{CR}<V_{RT}$
     V_{CR} cannot be modified during deceleration
  3. Time it takes for the speed of an axis to increase to the V_{RT} set.
  4. Deceleration time: Time it takes for the V_{RT} set decrease to the V_{CR} set, and the time it takes for the V_{CR} set to 0.
  5. Number of PG0 pulses for an axis: After DOG's signal is generated, the motor used will rotate for a specific number of PG0 pulses.
     Range: 0~+32,767 pulses
  6. Number of supplementary pulses for an axis: If the number of supplementary pulses for an axis is a positive number, the axis specified will move in the direction in which it returns home. If the v Number of supplementary pulses for an axis is a negative value, the axis specified will move in the direction which is opposite to the direction in which it returns home.
     Range: -32,768~32,767 pulses
  7. Make sure that the flags used to disable the X-axis and the Y-axis from retuning home are set correctly.
- Parameters below should be set in special data registers. D1816 is for the X-axis, D1896 is for the Y-axis, and D1976 is for the Z-axis.
  1. Direction in which the axis specified returns home:
     Bit 8=0: The value indicating the present command position of an axis specified decreases, and the axis returns home in the negative direction.
     Bit 8=1: The value indicating the present command position of an axis specified increases, and the axis returns home in the positive direction.
  2. Mode of returning home: Bit 9=0: Normal mode; Bit 9=1: Overwrite mode
  3. Mode of triggering the return to home :
     Bit 10=0: The return to home is triggered by a transition in DOG's signal from high to low
     Bit 10=1: The return to home is triggered by a transition in DOG's signal from low to high.
  4. If point 2 and point 3 are combined, there will be four modes of returning home. Please refer to section 3.12 for more information.

Example

◆ After X0 is ON, DRVZ in Ox0 will be executed, and the Y-axis and the Z-axis will be disabled from returning home. The X-axis accelerates to the V_{RT} (500 kHz) in 100 ms, and operates at the V_{RT} until DOG's signal goes from low to high. When DOG's signal goes from low to high, the X-axis decelerates to the V_{CR} (10 kHz) in 100 ms. The mode of returning home is a normal mode. After the X-axis rotates for a specific number of PG0 pulses, and rotate for a specific number of supplementary pulses, it will stop. In the example, the number of PG0 pulses and the number of supplementary pulses are 0. After DOG's signal goes from high to low, the X-axis will stop.



Additional remark

◆ Relevant flags: Please refer to chapter 3 for more information.

M1745	Disabling the X-axis from returning home in the Ox motion subroutine
M1825	Disabling the Y-axis from returning home in the Ox motion subroutine
M1985	Disabling the Z-axis from returning home in the Ox motion subroutine
M1074	Enabling the Ox motion subroutine specified

◆ Relevant special data registers

D1816	Setting the parameters of the X-axis
D1846	Operation command for the X-axis
D1868	Setting an Ox motion subroutine number
D1828, D1829	Speed (V_{RT}) at which the X-axis returns home
D1830, D1831	Speed (V_{CR}) to which the speed of the X-axis decreases when the X-axis returns home
D1832	Number of PG pulses for the X-axis (Generally, D1832 is used to set the umber of Z-phase signals which are sampled.)
D1833	Supplementary pulses for the X-axis (Generally, D1833 is used to set an Offset. Unit: Pulse)
D1836	Time (T_{ACC}) it takes for the X-axis to accelerate
D1837	Time (T_{DEC}) it takes for the X-axis to decelerate
D1896	Setting the parameters of the Y-axis
D1908, D1909	Speed (V_{RT}) at which the Y-axis returns home
D1910, D1911	Speed (V_{CR}) to which the speed of the Y-axis decreases when the Y-axis returns home
D1912	Number of PG pulses for the Y-axis (Generally, D1912 is used to set the umber of Z-phase signals which are sampled.)
D1913	Number of Supplementary pulses for the Y-axis (Generally, D1913 is used to set an Offset. Unit: Pulse)
D1916	Time (T_{ACC}) it takes for the Y-axis to accelerate
D1917	Time (T_{DEC}) it takes for the Y-axis to decelerate
D1976	Setting the parameters of the Z-axis
D1988, D1989	Speed (V_{RT}) at which the Z-axis returns home
D1990, D1991	Speed (V_{CR}) to which the speed of the Z-axis decreases when the Z-axis returns home
D1992	Number of PG pulses for the Z-axis (Generally, D1992 is used to set the umber of Z-phase signals which are sampled.)
D1993	Number of supplementary pulses for the Z-axis (Generally, D1993 is used to set an Offset. Unit: Pulse)
D1996	Time (T_{ACC}) it takes for the Z-axis to accelerate
D1997	Time (T_{DEC}) it takes for the Z-axis to decelerate

Motion instruction number				Applicable model
	SETR	No operand	Setting electrical zero	20PM
08				✓

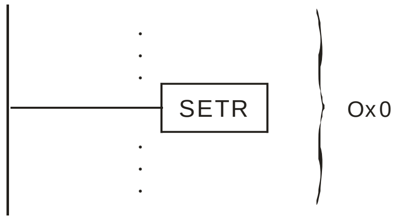
There is no operand. The instruction can be followed by an M-code instruction.

Explanation

- When SETR is executed, users can set the present position of the X-axis/Y-axis/Z-axis as electrical zero, i.e. users can move the contents of the present command position register into the register for electrical zero. DVP20PM00D only supports the setting of electrical zero of the X-axis and the setting of electrical zero of the Y-axis. DVP20PM00M supports the setting of electrical zero of the X-axis, the setting of electrical zero of the Y-axis, the setting of electrical zero of the Z-axis.

Example

- Program



Additional remark

- Relevant special data registers: Please refer to chapter 3 for more information.

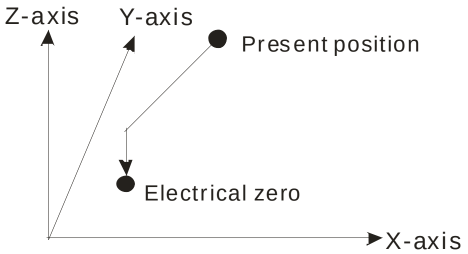
D1848, D1849	Present command position of the X-axis
D1866, D1867	Electrical zero of the X-axis
D1928, D1929	Present command position of the Y-axis
D1946, D1947	Electrical zero of the Y-axis
D2008, D2009	Present command position of the Z-axis
D2026, D2027	Electrical zero of the Z-axis

Motion instruction number	DRVR	No operand	Returning to electrical zero	Applicable model
09				20PM
				✓

There is no operand. The instruction can be followed by an M-code instruction.

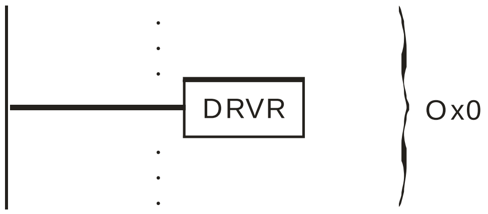
Explanation

- When DRVR is executed, the X-axis, the Y-axis, and the Z-axis return to electrical zero at the V_{MAX} (0~500 kHz) set.
- Moving path



Example

- Program



Additional remark

- Relevant special data registers: Please refer to chapter 3 for more information.

D1822, D1823	Maximum speed (V_{MAX}) at which the X-axis rotates
D1848, D1849	Present command position of the X-axis
D1866, D1867	Electrical zero of the X-axis
D1902, D1903	Maximum speed (V_{MAX}) at which the Y-axis rotates
D1928, D1929	Present command position of the Y-axis
D1946, D1947	Electrical zero of the Y-axis
D1982, D1983	Maximum speed (V_{MAX}) at which the Z-axis rotates
D2008, D2009	Present command position of the Z-axis
D2026, D2027	Electrical zero of the Z-axis

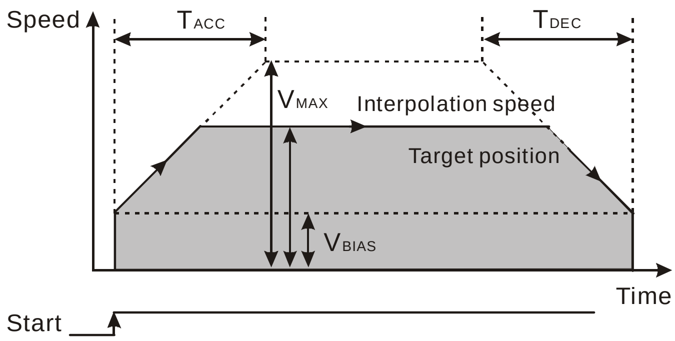
Motion instruction number	INTR	X(P1) Y(P2) F(V)	Two-axis single-Speed linear interpolation (The remaining distance is ignored.)	Applicable model
10				20PM
				✓

	Word device			Double word device		
	K	H	D	KK	HH	DD
P_1	*	*	*	*	*	*
P_2	*	*	*	*	*	*
V	*	*	*	*	*	*

Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. Please refer to the specifications for the DVP-20PM series motion controller used for more information about the device ranges available. The instruction can be followed by an M-code instruction.

Explanation

- $\mathbf{P_1}$: Target position of the X-axis; $\mathbf{P_2}$: Target position of the Y-axis; **V**: Speed of two-axis linear interpolation
- 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 in special data registers.
- Acceleration/deceleration time increases or decreases in proportional to the setting of V_{MAX}.
- Timing diagram



- The interpolation speed is monitored by special data registers: D1850~D1851 are for the X-axis. D1930~D1931 are for the Y-axis.
- The function of INTR is the same as the function of LIN. (If LIN is executed, a stop mode can be used, and the remaining distance can be considered.)
- A target position is required, but moving speed could be left out, and therefore there are 6 operand combinations for INTR.

No.	Motion instruction	Combination of operands
1	INTR	X(P1)
2		X(P1) F(V)
3		Y(P2)
4		Y(P2) F(V)
5		X(P1) Y(P2)
6		X(P1) Y(P2) F(V)

Additional remark

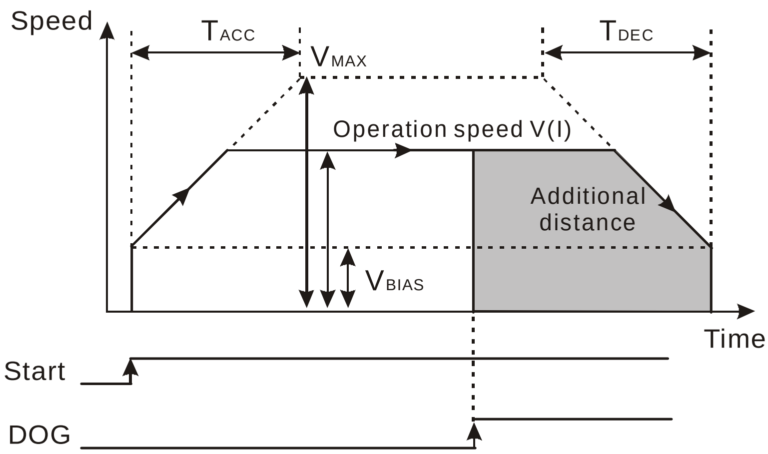
- Relevant special data register: Please refer to the addition remark on LIN for more information.

Motion instruction number	SINTR	X(P1)/Y(P2) F(V)	Inserting a single speed	Applicable model
11				20PM
				✓

	Word device			Double word device			Note
	K	H	D	KK	HH	DD	Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. Please refer to the specifications for the DVP-20PM series motion controller used for more information about the device ranges available. The instruction can be followed by an M-code instruction.
P_1	*	*	*	*	*	*	
P_2	*	*	*	*	*	*	
V	*	*	*	*	*	*	

Explanation

- **P_1**: Additional distance on the X-axis; **P_2**: Additional distance on the Y-axis; **V**: Operation speed
- Maximum **V**=V_{MAX}
- The first operand of SINTR can be either the X-axis or the 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 in special data registers.
- 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

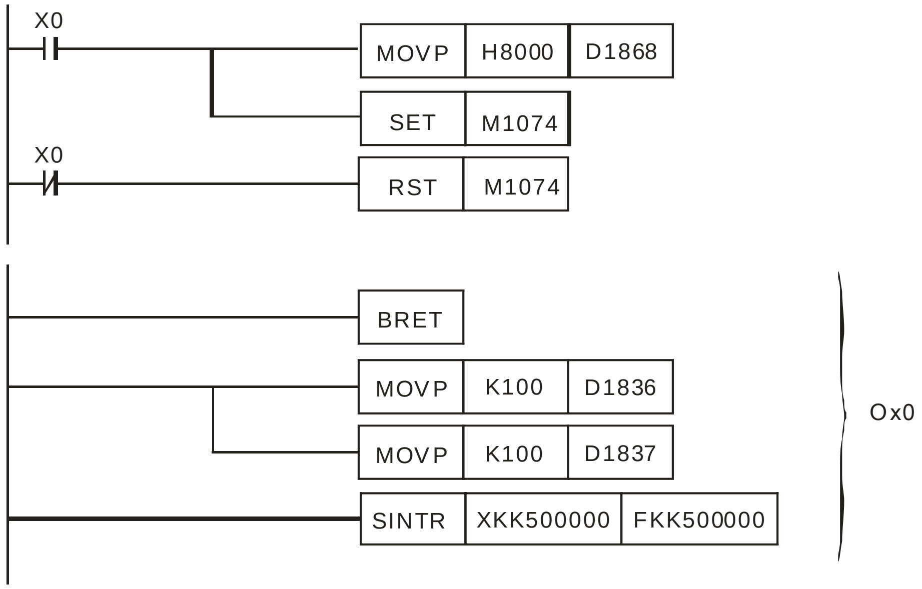


- When SINTR is enabled, the operation speed will start from the V_{BIAS} set, accelerate to the V (I) set, and operate stably. When the execution encounters DOG's signal, it will follow the additional distance set by the instruction and complete the positioning process.
- Both additional distance and moving speed are required, and therefore there are 2 operand combinations for SINTR.

No.	Motion instruction	Combination of operands
1	SINTR	X(P1) F(V)
2		Y(P2) F(V)

Example

- If X0 is ON, SINTR in Ox0 will be executed. The X-axis accelerates to 500kHz in 100 ms. When DOG's signal is triggered, SINTR will finish the single speed positioning on the X-axis with the additional 500,000 pulses set by the first operand.
- If X0 is OFF, Ox0 and SINTR will be disabled.



Additional remark

- Even if an absolute coordinate system is adopted in the program, the displacement will be regarded as additional distance after SINTR is executed.
- Relevant special data registers: Please refer to chapter 3 for more information.

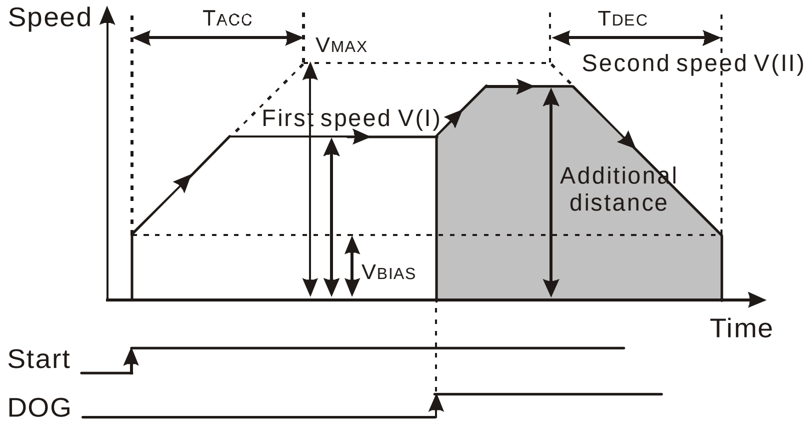
D1836	Time (T_{ACC}) it takes for the X-axis to accelerate
D1837	Time (T_{DEC}) it takes for the X-axis to decelerate
D1848, D1849	Present command position of the X-axis
D1916	Time (T_{ACC}) it takes for the Y-axis to accelerate
D1917	Time (T_{DEC}) it takes for the Y-axis to decelerate
D1928, D1929	Present command position of the Y-axis

Motion instruction number	DINTR	X P_1 / Y P_2, F V_1 F V_2	Inserting two speeds	Applicable model
12				20PM
				✓

	Word device			Double word device			
	K	H	D	KK	HH	DD	Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. Please refer to the specifications for the DVP-20PM series motion controller used for more information about the device ranges available. The instruction can be followed by an M-code instruction.
P_1	*	*	*	*	*	*	
P_2	*	*	*	*	*	*	
V_1	*	*	*	*	*	*	
V_2	*	*	*	*	*	*	

Explanation

- **P_1**: Additional distance on the X-axis; **P_2**: Additional distance on the Y-axis; **V_1**: First speed; **V_2**: Second speed
- Maximum **V_1/V_2**=V_{MAX}
- The first operand of DINTR can be either the X-axis or the Y-axis for double 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 in special data registers.
- 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

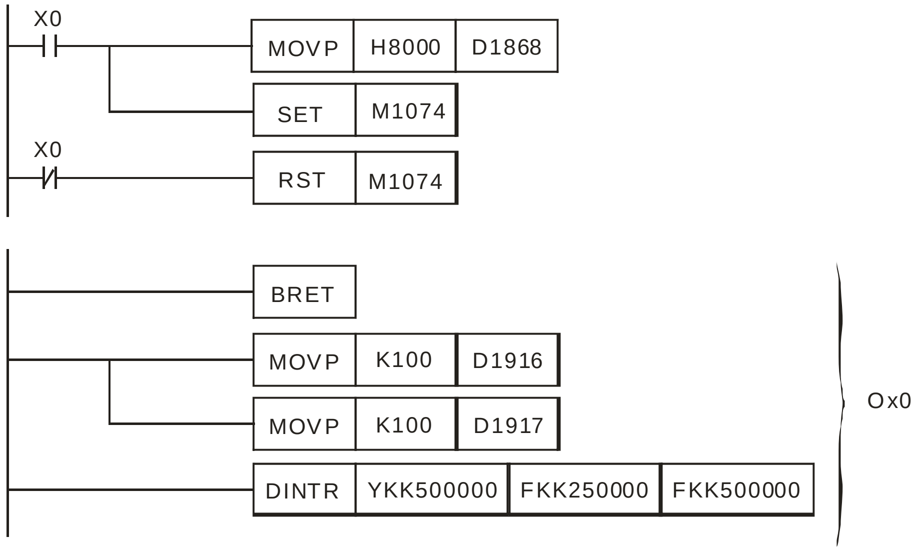


- When DINTR is enabled, the operation speed will start from the V_{BIAS} set, accelerate to the V (I) set, and operate stably. When the execution encounters DOG's signal, it will further accelerate to the V (II) set, and then finish the additional distance set by the instruction.
- Both additional distance and moving speed are required, and therefore there are 2 operand combinations for SINTR.

No.	Motion instruction	Combination of operands
1	DINTR	X P_1 F V_1 F V_2
2		Y P_2 F V_1 F V_2

Example

- If X0 is ON, DINTR in Ox0 will be executed. The Y-axis accelerates to the first speed 250 kHz in 100 ms, and operates at the speed stably. When DOG's signal is triggered, it further accelerates to the second speed 500 kHz, and finishes the double speed positioning on the Y-axis with the additional 500,000 pulses.
- If X0 is OFF, Ox0 and DINTR will be disabled.



Additional remark

- Relevant special data registers: Please refer to chapter 3 for more information.

D1836	Time (T_{ACC}) it takes for the X-axis to accelerate
D1837	Time (T_{DEC}) it takes for the X-axis to decelerate
D1848, D1849	Present command position of the X-axis
D1916	Time (T_{ACC}) it takes for the Y-axis to accelerate
D1917	Time (T_{DEC}) it takes for the Y-axis to decelerate
D1928, D1929	Present command position of the Y-axis

Motion instruction number	MOVC	X(L1) Y(L2)	Setting the offset of linear movement	Applicable model
13				20PM
				✓

	Word device			Double word device		
	K	H	D	KK	HH	DD
L_1	*	*	*	*	*	*
L_2	*	*	*	*	*	*

Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. Please refer to the specifications for the DVP-20PM series motion controller used for more information about the device ranges available. The instruction can be followed by an M-code instruction.

Explanation

- **L_1**: Offset for the target position of the X-axis; **L_2**: Offset for the target position of the Y-axis
- Users can set only the offset value of the X-axis, e.g. MOVC XDD0.
- When MOVC is executed, the offset value is automatically written into special data registers: D1708~D1709 are for the X-axis. D1724~D1725 are for the Y-axis.
- The offset of linear movement can be adopted for DRV, LIN and INTR.
- If linear movement is executed after the offset value is written into special data registers, compensation will be carried out in accordance with the offset value.
- 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.	Motion instruction	Combination of operands
1	MOVC	X(L1)
2		X(L1) Y(L2)

Additional remark

- Relevant special data registers: Please refer to chapter 3 for more information.
  D1708, D1709 Compensation value for the X-axis
  D1724, D1725 Compensation value for the Y-axis

Motion instruction number	CNTC	I (L1) J (L2)	Setting an offset for the center of an arc	Applicable model
14				20PM
				✓

	Word device			Double word device			Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. Please refer to the specifications for the DVP-20PM series motion controller used for more information about the device ranges available. The instruction can be followed by an M-code instruction.
	K	H	D	KK	HH	DD	
L_1	*	*	*	*	*	*	
L_2	*	*	*	*	*	*	

Explanation

- **L_1**: Compensation value for the center of the arc created by the X-axis; **L_2**: Compensation value for the center of the arc created by the Y-axis
- When CNTC is executed, the offset value is automatically written into special data registers: D1710~D1711 are for the X-axis. D1726~D1727 are for the Y-axis.
- The compensation value for the center of an arc can be adopted in CW and CCW.
- If circular movement is executed after the offset value is written into special data registers, compensation will be carried out in accordance with the offset value.
- 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.

Additional remark

- Relevant special data registers: Please refer to chapter 3 for more information.

D1710, D1711	Compensation value for the center of the arc created by the X-axis
D1726, D1727	Compensation value for the center of the arc created by the Y-axis

Motion instruction number	RADC	R (L)	Setting an offset for a radius	Applicable model
15				20PM
				✓

	Word device			Double word device			
	K	H	D	KK	HH	DD	Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. Please refer to the specifications for the DVP-20PM series motion controller used for more information about the device ranges available. The instruction can be followed by an M-code instruction.
L	*	*	*	*	*	*	

Explanation

- **L**: Compensation value for a radius
- When RADC is executed, the offset value is automatically written into special data registers: D1712~D1713.
- The compensation value for a radius can be adopted in CW and CCW instructions.
- If circular movement is executed after the offset value is written into special data registers, compensation will be carried out in accordance with the offset value.
- 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.

Additional remark

- Relevant special data registers: Please refer to chapter 3 for more information.
  D1712, D1713 Compensation value for the radius

Motion instruction number	CANC	No operand	Canceling compensation	Applicable model
16				20PM
				✓

There is no operand. The instruction can be followed by an M-code instruction.

Explanation

◆ When CANC is executed, all motion compensation values are cancelled, i.e. the special data registers D1708~D1709, D1724~D1725, D1710~D1711, D1726~D1727, and D1712~D1713 are cleared automatically.

Additional remark

◆ Relevant special data registers: Please refer to chapter 3 for more information.

D1708, D1709	Compensation value for the X-axis
D1724, D1725	Compensation value for the Y-axis
D1710, D1711	Compensation value for the center of the arc created by the X-axis
D1726, D1727	Compensation value for the center of the arc created by the Y-axis
D1712, D1713	Compensation value for the radius

Motion instruction number	ABST	No operand	Absolute programming	Applicable model
17				20PM
				✓

Motion instruction number	INCT	No operand	Incremental programming	Applicable model
18				20PM
				✓

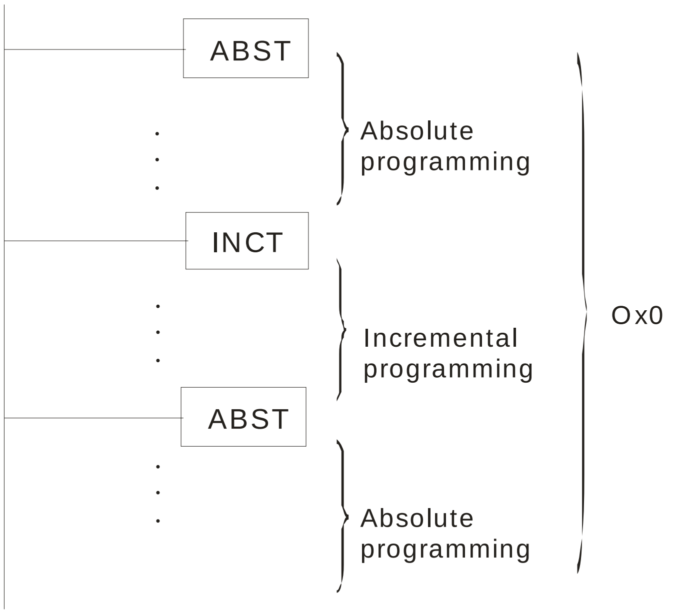
There is no operand. The instruction can be followed by an M-code instruction.

Explanation

- ABST enables an absolute coordinate system. An Absolute coordinate is a coordinate starting from 0. If the target position of an axis is greater than the present position of the axis, the motor for the axis will rotate clockwise. If the target position of an axis is less than present position of the axis, the motor for the axis will rotate counterclockwise.
- INCT enables a relative coordinate system. A relative coordinate is relative movement from the present position of an axis. If the relative coordinates used are positive values, the motor used will rotate clockwise. If the relative coordinates used are negative values, the motor used will rotate counterclockwise.
- The center point (I, J), the radius (R), and the additional distances set by SINTR and DINTR are regarded as relative increments.

Example

- If ABST and INCT are not in the program in a DVP-20PM series motion controller, the default setting for the program will be an absolute coordinate system after the STOP/RUN switch of the DVP-20PM series motion controller is turned form the “STOP” position to the “RUN” position. After INCT is executed, the motion instructions starting from the next row (e.g. DRV, LIN, CW, and CCW) will be operated in a relative coordinate system.



Motion instruction number	SETT	X P_1 Y P_2 Z P_3	Setting a present position	Applicable model
				20PM
19				✓

	Word device			Double word device			Note
	K	H	D	KK	HH	DD	
P_1	*	*	*	*	*	*	Note: The instruction supports devices. The devices supported can be modified by V devices can Z devices. Please refer to the specifications for the DVP-20PM series motion controller used for more information about the device ranges available. The instruction can be followed by an M-code instruction.
P_2	*	*	*	*	*	*	
P_3	*	*	*	*	*	*	

Explanation

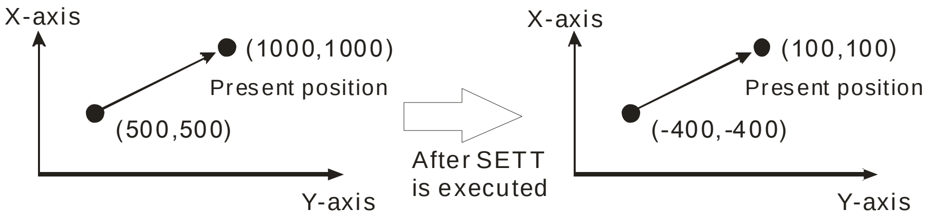
- **P_1**: Present position of the X-axis; **P_2**: Present position of the Y-axis; **P_3**: Present position of the Z-axis
- When SETT is executed, the present positions set in SETT are written automatically into present position registers: D1848~D1849 are for the X-axis. D1928~D1929 are for the Y-axis. D2008~D2009 are for the 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.
- The values in the present position registers become the values specified by SETT when SETT is executed, and therefore mechanical zero and electrical zero are changed.
- At least one present position should be specified, and therefore there are 7 operand combinations for SETT.

No.	Motion instruction	Combination of operands	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	-	✓

Example

- Program

- After SETT is executed, the present position of the X-axis and the present position of the Y-axis will be changed, and the relevant position will also be changed.



Additional remark

- Relevant special data registers: Please refer to chapter 3 for more information.
  D1848, D1849 Present command position of the X-axis
  D1928, D1929 Present command position of the Y-axis
  D2008, D2009 Present command position of the Z-axis

6.4 Descriptions of O Pointers / M-code Instructions

Instruction	Function	Applicable model
		20PM
O	Program pointer	✓

Description	Main program: O100 Motion subroutines: Ox0~Ox99

Explanation

- O100 is a main program pointer. The motion subroutines Ox0~Ox99 are activated by the main program O100. M102 indicates the end of O100.
- Ox0~Ox99 are motion subroutine pointers. Users can use them to create different motion paths. A subroutine number is stored in D1868 (CR72). If users want to store a subroutine number in D1868, they have to set bit 14 or bit 15 in D1868 to 1. After bit 12 in D1846 is set to ON or M1074 is set to ON, the execution of the Ox motion subroutine specified will start.
  Example: Two steps to activate the motion subroutine Ox99 are as follows.
  (1) Setting a number: D1868=H8063 (or H4063/HC063)
  (2) Starting the execution of Ox99: D1846=H1000 or M1074=ON
- M2: End of a motion subroutine

Example

- The main program O100 is composed of N0000~N0100, and the motion subroutine Ox50 is composed of N0102~N0304.

Line number	Program
N0000	**O100**
N0001	**LD M1000**
N0002	**MOV H8063 D1868**
N0007	**OUT M1074**
:	**:**
:	**:**
N0099	**OUT Y30**
N0100	**M102**
N0101	**NOP**
N0102	**OX50**
N0103	**DRVZ**
N0104	**ABS**
:	**:**
:	**:**
N0304	**M2**

Instruction	Function	Applicable model
		20PM
M	M-code instructions	✓

Operand	M0~M65535 M102: End of O100 M2: End of a motion subroutine

Explanation

- M-codes are used in motion subroutines, If an M-code is executed, the M-code will be stored in D1703, and M1794 will be automatically set to ON. If M1744 is ON, M1794 will be OFF, and the execution of an M-code is complete.
- A Y device can be controlled by the execution of an M-code in two ways. If the high byte in D1873 is 1, an output device can be set to ON. Users can set a Y device number in the low byte in D1873. If M1794 is ON (an M-code is executed), the Y device corresponding to the Y device number set in D1873 will be ON. If M1794 is OFF, the Y device corresponding to the Y device number set in D1873 will be OFF. Please refer to example 1 below for more information. The other way to control a Y device is using M1794 and D1703. Please refer to example 2 below for more information.
- M-codes are generally used in motion subroutines Ox0~Ox99.
- M codes can be used in two modes. They can be used in after mode or in with mode. The difference between the two modes lies in the time when an M-code is executed. Please refer to example 3 below for more information.
- After the execution of an M-code is complete, M1794 can be turned from ON to OFF in one of the two ways described blow.
  (1) Users can set M1794 to OFF, and then M1794 is reset.
  (2) Users can set M1744 to ON.

Additional remark

- M102 indicates the end of O100, and M02 indicates the end of a motion subroutine. Users should avoid using M102 and M02.

Example 1

- Display the present number of the M-code which is executed, e.g. M6, by the states of Y devices.

(1) First set the parameter in D1873

MOV H0□□□ D1873

□□ → N: Start No. of Y output

□ → 0: Disable

1: Enable

(2) Execute the M-code (M6). In this case, the DVP-20PM series motion controller used automatically writes K6 (0110) into D1703, and writes the value in D1703 into $K2Y_N$. N is the start No. of a Y device.

MOV H6 D1703

MOV D1703 $K2Y_N$

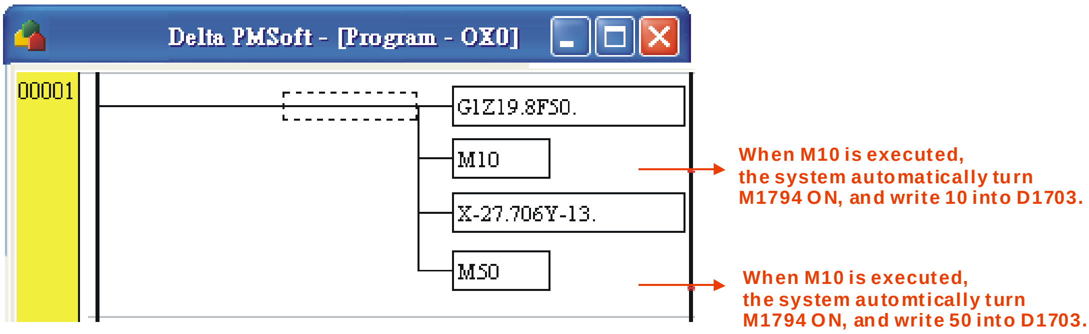
When the M-code is enabled, the instructions above run automatically. Users do not need to type instructions.

(3) The state of $K2Y_N$ varies with the setting of N in D1873.

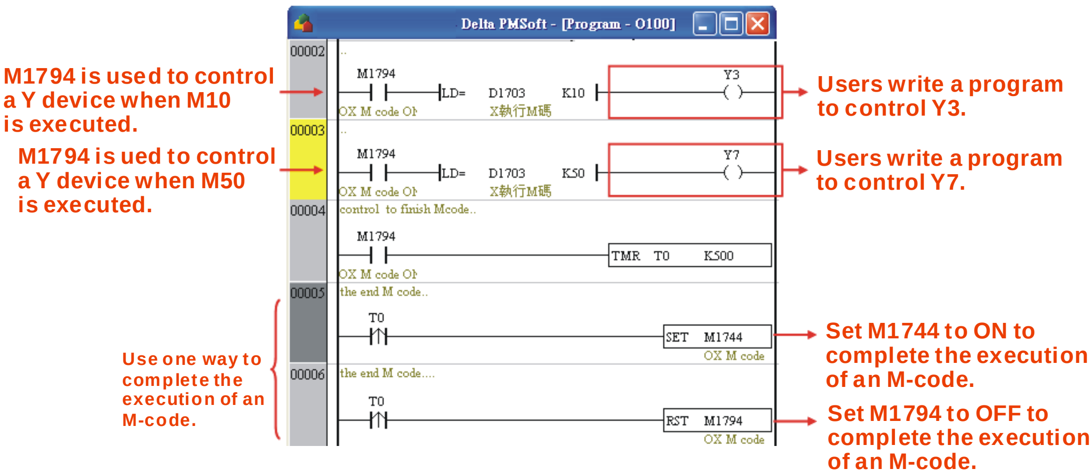
D1873	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
H00□□	The Y device specified does not act.							
H0100	0	0	0	0	0	1	1	0
H0101	0	0	0	0	1	1	0	-
H0102	0	0	0	1	1	0	-	-
H0103	0	0	1	1	0	-	-	-
⋮	⋮							

Example 2

- When an M-code in the motion program in a DVP-20PM series motion controller is executed, M1794 is ON. The M-code which is executed is stored in D1703. Users can use M1794 and D1703 to control Y devices. After an M-code in a motion subroutine is executed, M1744 can be set to ON, or M1794 can be set to OFF, otherwise the execution of the motion subroutine will stay at the M-code.



(a) Ox0



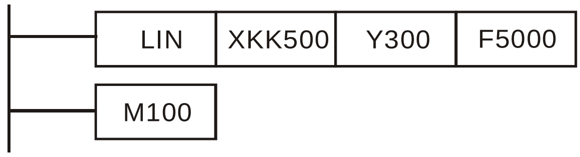
(b) O100

- The sequence control required is shown above.
- Step 1: When Ox0 is executed, M10 is executed.
  (1) M1794 is turned ON, and 10 is written into D1703.
  (2) The condition in network 00002 in O100 is met, and therefore Y3 is ON.
- Step 2: M1794 is turned ON.
  (1) To in network 00004 in O100 begins to count.
  (2) When the value in T0 is 500, T0 is ON. In order to complete the execution of M10, M1744 in network 00005 is set to ON, or M1794 in network 00006 is set to OFF. Note: Network 0005 in O100 is disabled, and therefore is not downloaded to the DVP-20PM series motion controller used. Users can use one of the two ways to complete the execution of M10.

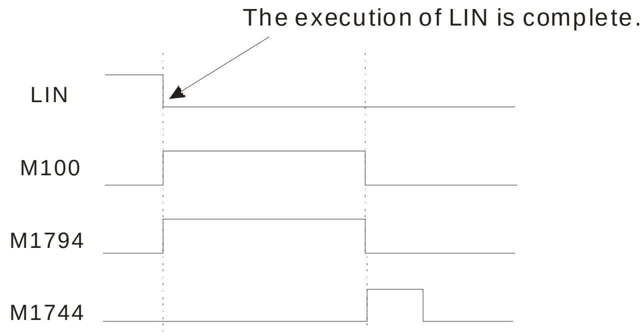
- Step 3: M50 in Ox0 is executed.
  - (1) M1794 is turned ON, and 50 is written into D1703.
  - (2) The condition in network 00003 in O100 is met, and therefore Y7 is ON.
- Step 4: M1794 is turned ON.
  - (1) T0 in network 00004 in O100 begins to count.
  - (2) When the value in T0 is 500, T0 is ON. In order to complete the execution of M50, M1744 in network 00005 is set to ON, or M1794 in network 00006 is set to OFF.
- When an M-code in a program is executed, special auxiliary relays and a special data register are used to create the functions that the program need.

Example 3

- After mode: If an M-code forms a line, it is used in after mode.

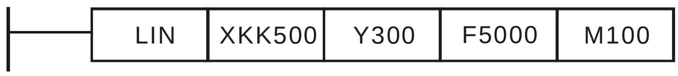


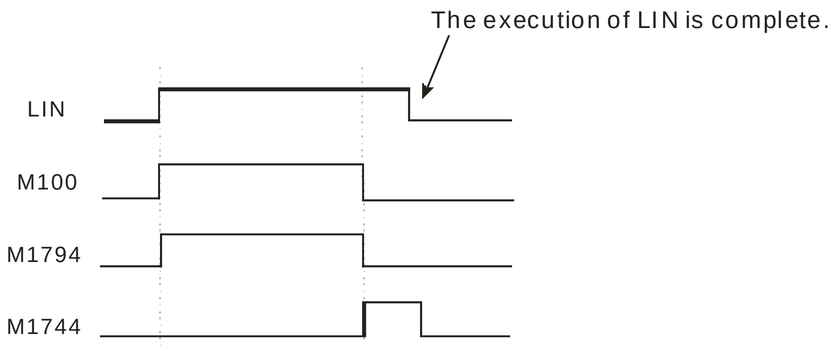
Timing diagram:



After LIN is executed, the M-code M100 will be started automatically, and M1794 will be automatically turned ON. If users want to stop M100, they have to turn M1744 ON. If they want to start the M-code again, they can create a program that starts the M-code again.

- With mode: If an M-code is in back of a motion instruction, it is used in with mode.





When the execution of LIN is triggered, the M-code M100 is started automatically, and M1794 is automatically turned ON. If users want to stop M100, they have to turn M1744 ON. If they want o start the M-code again, they have to set parameters after the execution of LIN is complete, and create a program that starts the M-code again.

Example 4

- The M-codes in N0100 and N0304 are special M-codes. The M-code in N0105 is used in with mode, and the M-code in N0250 is used in after mode.

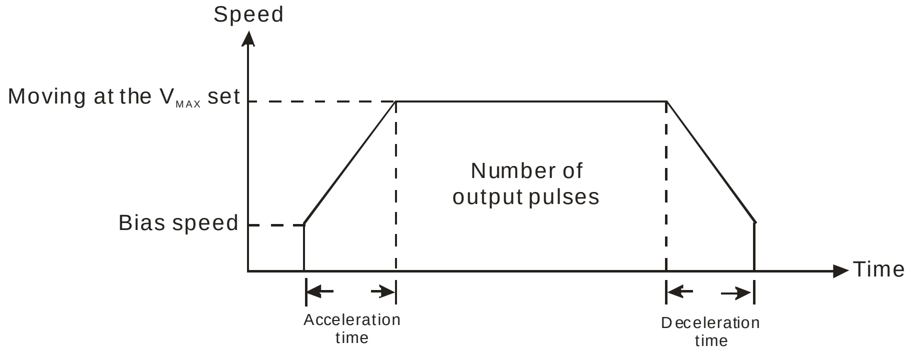
Line number	Program
N0000	**O100**
N0001	**LD M1000**
:	:
:	:
N0099	**OUT Y30**
N0100	**M102**
N0101	**NOP**
N0102	**OX50**
N0103	**DRVZ**
N0104	**ABS**
N0105	**DRV XD10 FXD12 M20**
:	:
N0250	**M08**
:	:
N0304	**M2**

6.5 Descriptions of G-code Instructions

G-code	G00	X (P1) Y (P2) Z (P3)	Rapid positioning (three axes)	Applicable model
				20PM
				✓

Explanation

- P_1: Target position of the X-axis; P_2: Target position of the Y-axis; P_3: Target position of the Z-axis
- For DVP20PM00D, the target position of the Z-axis (P_3) is controlled by the built-in third axis. Please refer to page 6-51 for more information.
- Range of parameters: -2,147,483,648~2,147,483,647 (without a decimal point); -2,147,483.648~2,147,483.647 (with a decimal point)
- The operand parameters for DVP20PM00M can also be 16-bit registers or 32-bit registers.
- Please refer to the additional remark on DRV for more information about special data registers.
- The moving speed of G00 is the V_{MAX} set.
- The setting of a position has continuity. Please refer to the addition remark below for more information.
- Acceleration/deceleration time and bias speed can be set in special data registers.
- Acceleration/deceleration time and bias speed increase or decrease in proportional to the setting of V_{MAX}.
- Timing diagram



- DVP20PM00D does not support three-axis positioning control. For third axis control, a DVP-01PU series single-axis positioning module is required. Therefore, users have to design two-axis high-speed positioning in the X-axis and the Y-axis and independent high-speed positioning for the Z-axis. For the safety of mechanical operation, if G00 is executed, Z-axis high-speed positioning will be executed first before the X-axis/Y-axis high-speed positioning. That is to say, when a DVP-20PM series motion controller executes G00 with X-Z, Y-Z, and X-Y-Z combinations, the program will automatically be divided.

  G00 ZP_3 (A)

  G00 XP_1 YP_2 (B)

  Please refer to the addition remark below for more information about (A) and (B).

Additional remark

- The setting of a position has continuity.

  Example:

  G00 X500.0 Y500.0 Z500.0;

  X1000.0 Y1000.0 Z1000.0;

  After the row with G00 is executed, the program will execute the next row. The second row in the program will reach the target position automatically by G00.

- DVP20PM00D: If G00 adopts the target position of the Z-axis (built-in third axis), the program required will be like the one shown below.

  G00 X1000 Y1000 Z100;

  The program will be compiled.

  G00 Z100; (A)

  G00 X1000 Y1000; (B)

  (A) is first executed, and at this time the operation fast moves to the position K100 on the Z-axis. (B) is next executed and the operation moves to the target position (1000, 1000) at the maximum speed.

G-code	G00	Z (P3)	Rapid positioning (third axis control)	Applicable model 20PM ✓

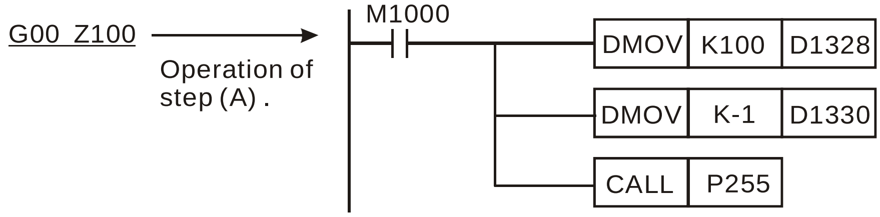
Additional remark

- DVP20PM00D does not support 3-axis positioning control. For third axis control, a DVP-01PU series single-axis positioning module is required.
- DVP20PM00D: If G00 adopts the target position of the Z-axis (built-in third axis), the program required will be like the one shown below.

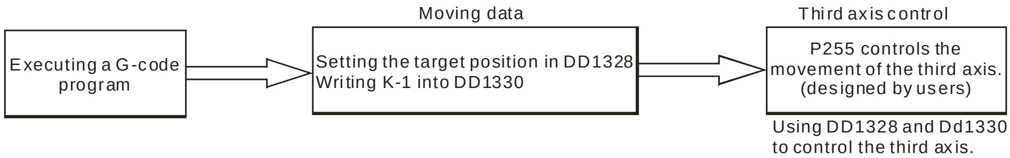
  G00 Z100; (A)

  In step (A), the DVP-20PM series motion controller used writes the target position K100 on the Z-axis into the registers D1328 and D1329. The moving speed of G00 is the maximum speed (V_{MAX}) set. Therefore, the PLC writes K-1 into D1330 and D1331 (K-1 refers to the positioning at the maximum speed). After this, the subroutine P255 is called and executed.

  Operation of step (A):



  P255 is a subroutine for controlling the third axis (e.g. a lift of a pen, clipping and releasing objects, and so on). If the target position is achieved by the control data set in DD1328 and DD1330, the operation in P255 will be executed.



  Please avoid using D1328~D1331 and P255 repeatedly when the movement of the Z-axis is controlled.

G-code	G00	X (P1) Y (P2)	Rapid positioning (two axes)	Applicable model 20PM ✓

Explanation

- The G-code is mainly for DVP20PM00D. Please refer to the explanation of DRV for more information.

G-code	G01	X(P1) Y(P2) Z(P3) F(V)	Linear interpolation (three axes) (The remaining distance can be considered.)	Applicable model 20PM
				✓

Explanation

- $\mathbf{P_1}$: Target position of the X-axis; $\mathbf{P_2}$: Target position of the Y-axis; $\mathbf{P_3}$: Target position of the Z-axis; **V**: Speed of linear interpolation
- For DVP20PM00D, the target position of the Z-axis ($\mathbf{P_3}$) is controlled by the built-in third axis. Please refer to page 6-53 for more information.
- Range of the parameters $\mathbf{P_1}$, $\mathbf{P_2}$, and $\mathbf{P_3}$: -2,147,483,648~2,147,483,647 (without a decimal point); range of the parameter **V**: 0~500,000 (without a decimal point); range of the parameters $\mathbf{P_1}$, $\mathbf{P_2}$, and $\mathbf{P_3}$: -2,147,483.648~2,147,483.647 (with a decimal point); range of the parameter **V**: 0~500.0 (with a decimal point)
- The operand parameters for DVP20PM00M can also be 16-bit registers or 32-bit registers.
- The speed of linear interpolation (**V)** has continuity. Please refer to the additional remark below for more information.
- Please refer to the explanation of LIN for more information about the positioning process of DVP20PM00M.
- DVP20PM00D does not support three-axis synchronous interpolation. For third axis control, a DVP-01PU series single-axis positioning module is required. Therefore, users have to design two-axis high-speed interpolation in the X-axis and the Y-axis and independent high-speed positioning for the Z-axis. For the safety of mechanical operation, if G01 is executed, Z-axis high-speed movement will be executed first before the X-axis/Y-axis interpolation. That is to say, when a DVP-20PM series motion controller executes G01 with X-Z, Y-Z, and X-Y-Z combinations, the program will automatically be divided.

  G01 Z$\mathbf{P_3}$ FV (A)

  G01 X $\mathbf{P_1}$ Y $\mathbf{P_2}$ FV (B)

  Please refer to page 6-53 for more information about (A) and (B).

Additional remark

- The setting of speed has continuity.

  Example:

  G01 X100 Y100 Z400 F200;

  X200 Y200;

  After the row with G01 is executed, the program will execute the next row. The second row in the program will reach the target position automatically at the speed F200 set in the first row.
- DVP20PM00D: If G01 adopts the target position of the Z-axis (built-in third axis), the program required will be like the one shown below.

  G01 X1000 Y1000 Z100 F200;

  The program will be compiled.

  G01 Z100 F200;…(A)

  G01 X1000 Y1000 F200; …(B)

  (A) is first executed, and at this time the operation fast moves to the position K100 on the Z-axis. (B) is next executed and the operation moves to the target position (1000, 1000) at the speed of K200.

G-code	G01	Z P_3 F V	Linear interpolation (three axes) (third axis control)	Applicable model: 20PM
				✓

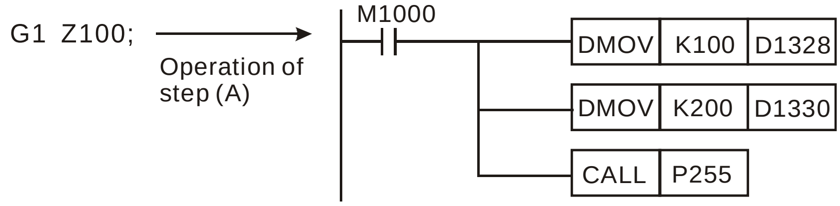
Additional remark

- DVP20PM00D does not support three-axis synchronous interpolation. For third axis control, a DVP-01PU series single-axis positioning module is required.
- DVP20PM00D: If G01 adopts the target position of the Z-axis (built-in third axis), the program required will be like the one shown below.

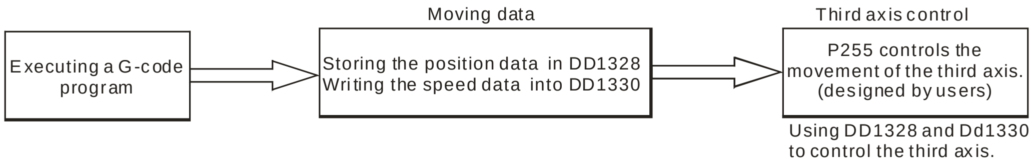
  G01 Z100 F200;…(A)

  In step (A), the DVP-20PM series motion controller used writes the target position K100 on the Z-axis into the registers D1328 and D1329. The moving speed of G01 is K200. K200 is written into D1330 and D1331. After this, the subroutine P255 is called and executed.

  Operation of step (A):



  P255 is a subroutine for controlling the third axis (e.g. a lift of a pen, clipping and releasing objects, and so on). If the target position is achieved by the control data set in DD1328 and DD1330, the operation in P255 will be executed.



  Please avoid using D1328~D1331 and P255 repeatedly when the movement of the Z-axis is controlled.

G-code	G01	X P_1 Y P_2 F V	Linear interpolation (two axes) (The remaining distance can be considered.)	Applicable model: 20PM
				✓

Explanation

- The G-code is mainly for DVP20PM00D. Please refer to the explanation of LIN for more information.

G-code	G02 G03	X(P_1) Y(P_2) Z(P_3) I(P_4) J(P_5) K(P_6) F(V)	Clockwise circular/helical interpolation Counterclockwise circular/helical interpolation (arc center)	Applicable model 20PM ✓

Explanation

- **P_1**: Target position of the X-axis; **P_2**: Target position of the Y-axis; **P_3**: Target position of the Z-axis; **P_4**: Vector from the present position of the X-axis to an arc center; **P_5**: Vector from the present position of the Y-axis to an arc center; **P_6**: Vector from the present position of the Z-axis to an arc center; **V**: Speed of circular/helical interpolation
- **P_4**, **P_5** and **P_6**: Vectors from the present positions of an x-axis, a y-axis, and a z-axis to an arc center
- Range of the parameters **P_1**, **P_2**, **P_3**, **P_4**, **P_5**, and **P_6**: -2,147,483,648~2,147,483,647 (without a decimal point); range of the parameter **V**: 0~500,000 (without a decimal point); range of the parameters **P_1**, **P_2**, **P_3**, **P_4**, **P_5**, and **P_6**: -2,147,483.648~2,147,483.647 (with a decimal point); range of the parameter **V**: 0~500.0 (with a decimal point)
- The operand parameters for DVP20PM00M can also be 16-bit registers or 32-bit registers.
- The speed of circular/helical interpolation (**V)** has continuity. Please refer to the additional remark below for more information.
- Please refer to the explanations of CW and CCW for more information about the positioning process of a DVP-20PM series motion controller.

Additional remark

- The setting of speed has continuity.
  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 is executed, the program will execute the next row. The second row in the program will reach the target position automatically at the speed F100.0 set in the first row.

G-code	G02 G03	X(P_1) Y(P_2) Z(P_3) R(L) F(V)	Clockwise circular/helical interpolation Counterclockwise circular/helical interpolation (radius)	Applicable model 20PM ✓

Explanation

- **P_1**: Target position of the X-axis; **P_2**: target position of the Y-axis; **P_3**: target position of the Z-axis; **L**: Arc radius (If the angle subtended by an arc is less than 180°, the value of R is a positive value. If the angle subtended by an arc is greater than 180°, the value of R is a negative value.); **V**: Speed of circular/helical interpolation
- Range of the parameters **P_1**, **P_2**, **P_3**, and **R**: -2,147,483,648~2,147,483,647 (without a decimal point); range of the parameter **V**: 0~500,000 (without a decimal point); range of the parameters **P_1**, **P_2**, **P_3**, and **R**: -2,147,483.648~2,147,483.647 (with a decimal point); range of the parameter **V**: 0~500.0 (with a decimal point)
- The operand parameters for DVP20PM00M can also be 16-bit registers or 32-bit registers.
- Please refer to the explanations of CW and CCW for more information about the positioning process of a DVP-20PM series motion controller.

G-code	G04	X(T)/P(T)	Dwell	Applicable model
				20PM
				✓

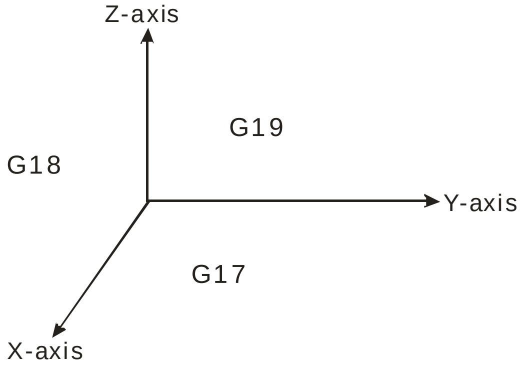
Explanation

- XT: Dwell time (Unit: 1 second)
  G4 X1: The dwell time set is one second.
  G4 X2: The dwell time set is 2.5 seconds.
- PT: Dwell time (Unit: 1 millisecond)
  G4 P100: The dwell time set is 0.1 seconds.
  G4 P4500: The dwell time set is 4.5 seconds.
  The setting value of **P** is a multiple of 10 milliseconds. If the setting value of **P** is less than 10 milliseconds, it will be regarded as 0 milliseconds. If the setting value of **P** is 23 milliseconds, it will be regarded as 20 milliseconds.
- The operand parameters for DVP20PM00M can also be 16-bit registers or 32-bit registers.
- Please refer to the explanation of TIM for more information.

G-code	**G17**	No operand	**XY plane selection**	Applicable model
	G18		**ZX plane selection**	20PM
	G19		**YZ plane selection**	✓

Explanation

- The three instructions determine the work planes for arc interpolation and helical interpolation and have no effects on linear interpolation.
- Users can switch the three planes during the execution of the instructions. If no work plane is selected, XY plane (G17) will be automatically selected as the work plane.
- Work planes:



G-code	G90	No operand	Absolute programming	Applicable model
				20PM
				✓

Explanation

- Please refer to the explanation of ABS for more information about the operation of G90.

				Applicable model
G-code	G91	No operand	Incremental programming	20PM
				✓

Explanation

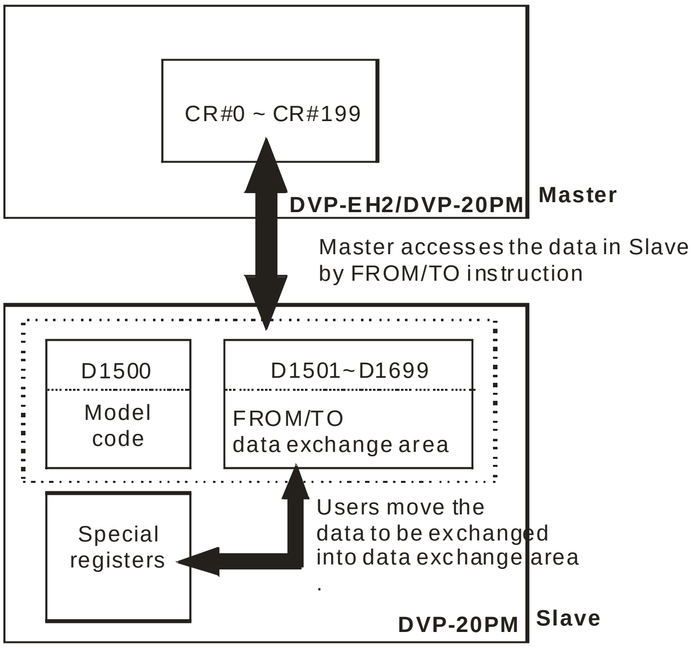
- Please refer to the explanation of INC for more information about the operation of G91.

7.1 Access between DVP-EH2, DVP-20PM (as a Master) and DVP-20PM (as a Slave)

When DVP-20PM is used as Slave, there is a data exchange area in DVP-20PM 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-20PM.

7.1.1 Structure

- ◆ DVP-EH2 and Master DVP-20PM apply FROM/TO instructions to transmit commands to Slave DVP-20PM for executing motion control subroutines.
- ◆ DVP-EH2 and Master DVP-20PM 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-20PM.



7.1.2 Example of Master-slave Data Exchange

- ◆ Set up: design the data exchange programs in Master and Slave respectively.
  - ● Slave DVP-20PM: 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-20PM 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】

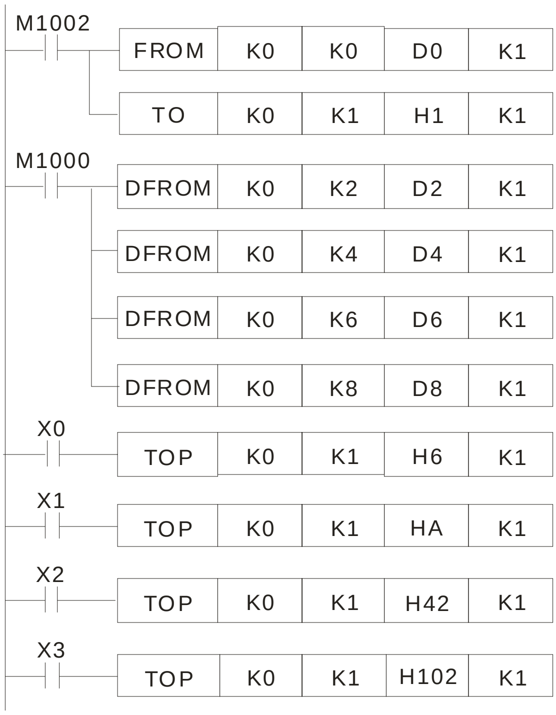
Master	Slave		Content
	Planned internally	Planned by user	
CR#0	D1500	Set up by the system	Model code of Slave DVP-20PM
CR#1	D1501	D1846	Operation commands for X axis
CR#2~3	D1502~D1503	D1848~D1849	Current position of X axis CP (Pulse)

Master	Slave		Content
	Planned internally	Planned by user	
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-20PM, 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-20PM (H'6260). Therefore, D1501~D1699 are the applicable registers.

【Program in DVP-EH2 master】

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.

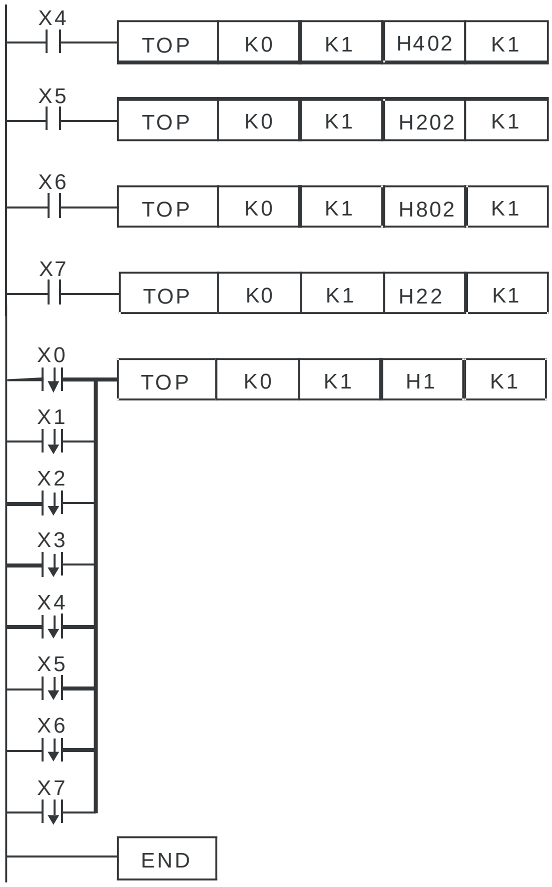
Read out CR#8 of Slave, corresponding to D1508 ~ D1509.

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.

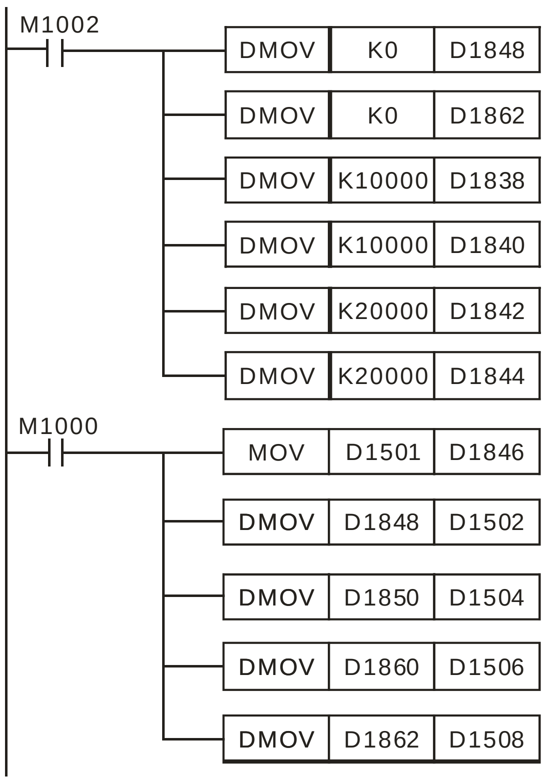
When X6 = ON, write in CR#1 of Slave, corresponding to D1501 in Slave, to enable 2-speed positioning with additional distance on X axis in Slave.

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.

Program in DVP-20PM 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

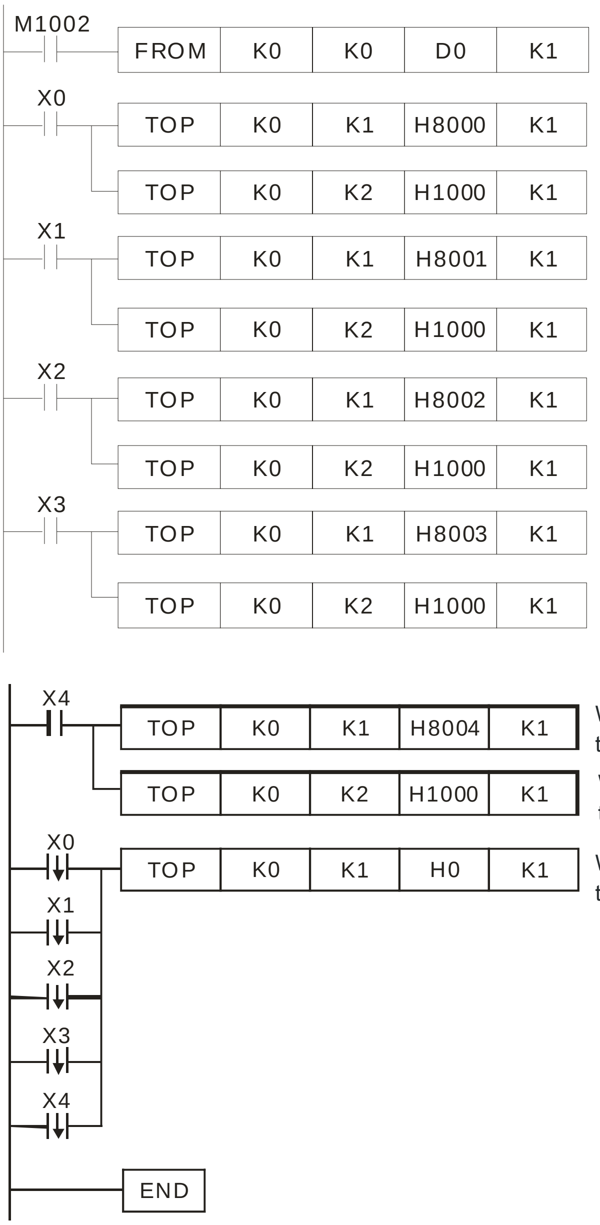
DVP-20PM and executes motion instructions in OX subroutines (see Chapter 6 for how to use motion instructions).

【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-20PM 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.
- When X0 ~ X4 = OFF, write in CR#1 of Slave, corresponding to D1501 in Slave, to disable OX subroutine in Slave.

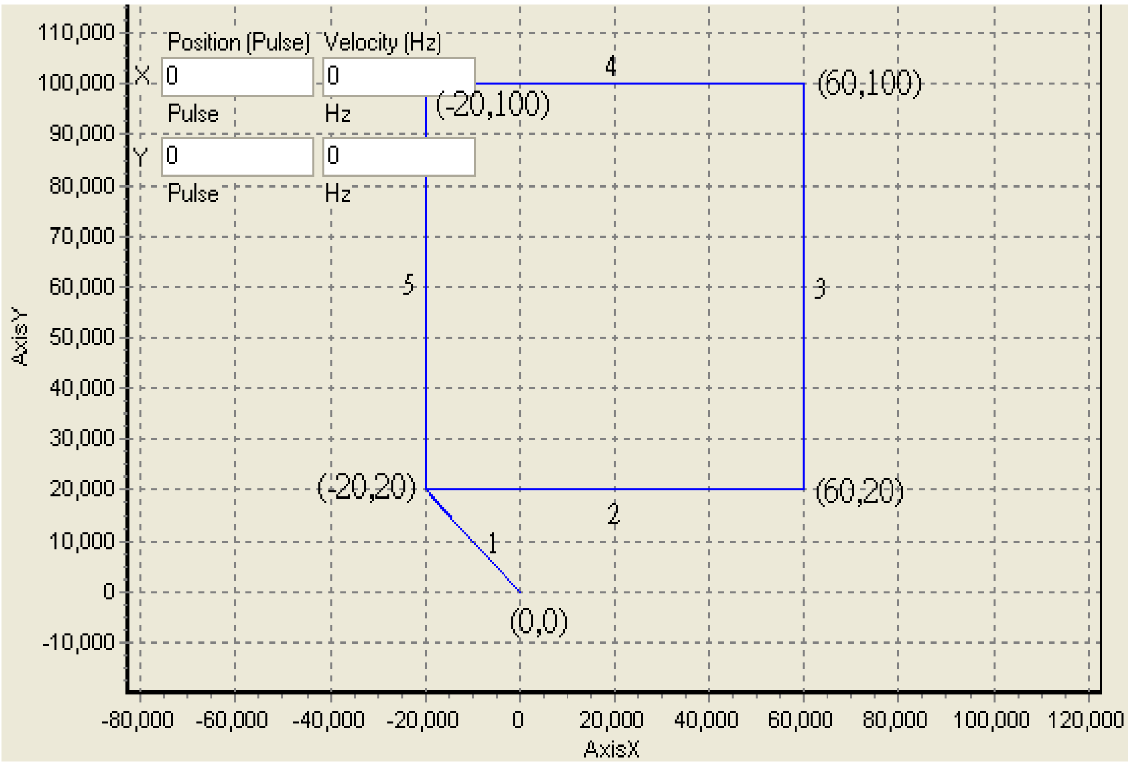
【Program in DVP-20PM slave】

Instruction mode:	Operation:
O100 LD M1002 DMOV K0 D1848 DMOV K0 D1928 LD M1000 MOV D1501 D1868 MOV D1502 D1846 M102	Place the instructions for initializing current position of X/Y axis in O100 main program. Enable O100 in Slave then call the motion subroutine.
OX00 DRV X200000 FX100000 Y200000 FY100000 M2	Place motion instruction DRV in OX00 subroutine.
OX01 LIN X100000 Y100000 F200000 M2	Place motion instruction LIN in OX01 subroutine.
OX02 CW X0 Y100000 I0 J50000 F200000 M2	Place motion instruction CW in OX02 subroutine.
OX03 CCW X0 Y100000 I0 J50000 F200000 M2	Place motion instruction CCW in OX03 subroutine.
OX04 BRET DMOV K200000 D1828 DMOV K100000 D1830 DMOV K200000 D1908 DMOV K100000 D1910 DRVZ M2	Place motion instruction DRVZ in OX04 subroutine, and set up relevant parameters for DRVZ.

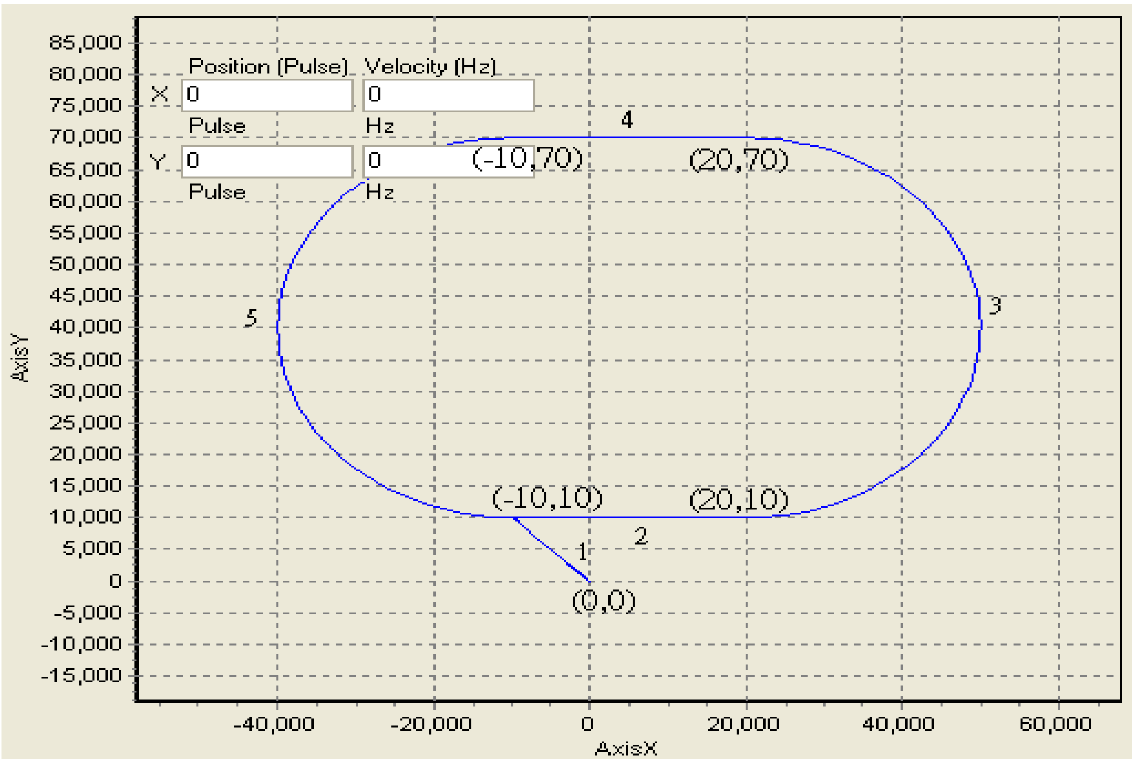
MEMO

8.1 Drawing Paths by Motion Instructions and G-codes

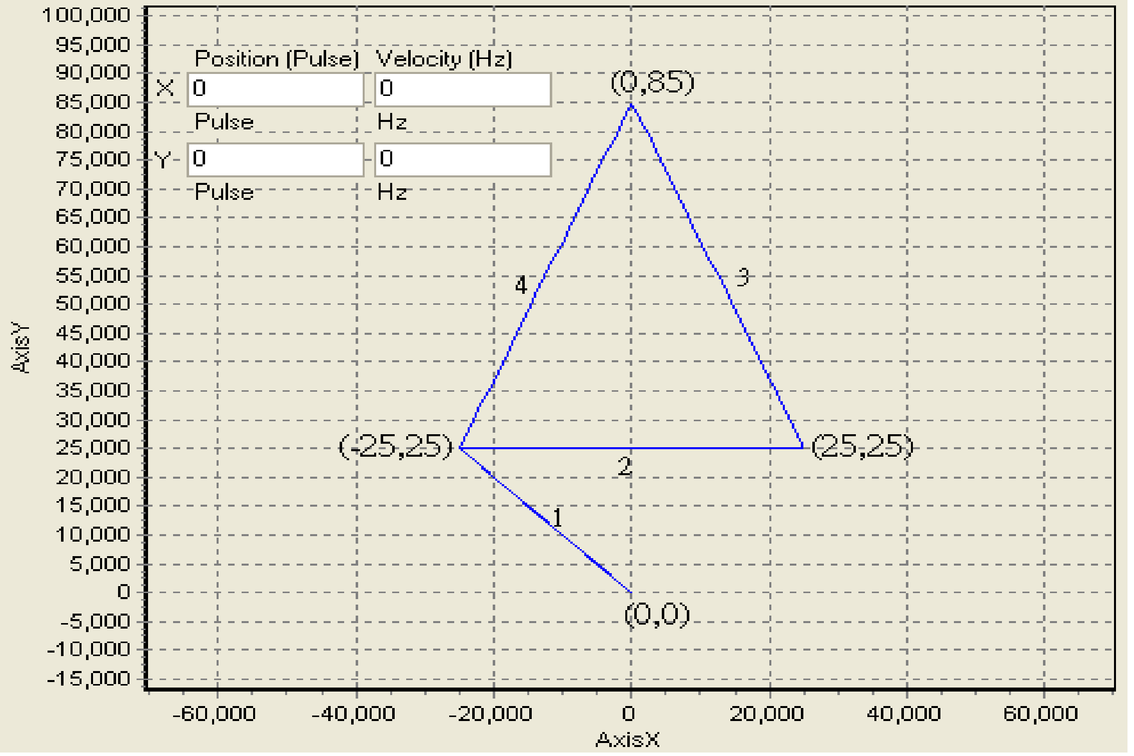
8.1.1 Paths



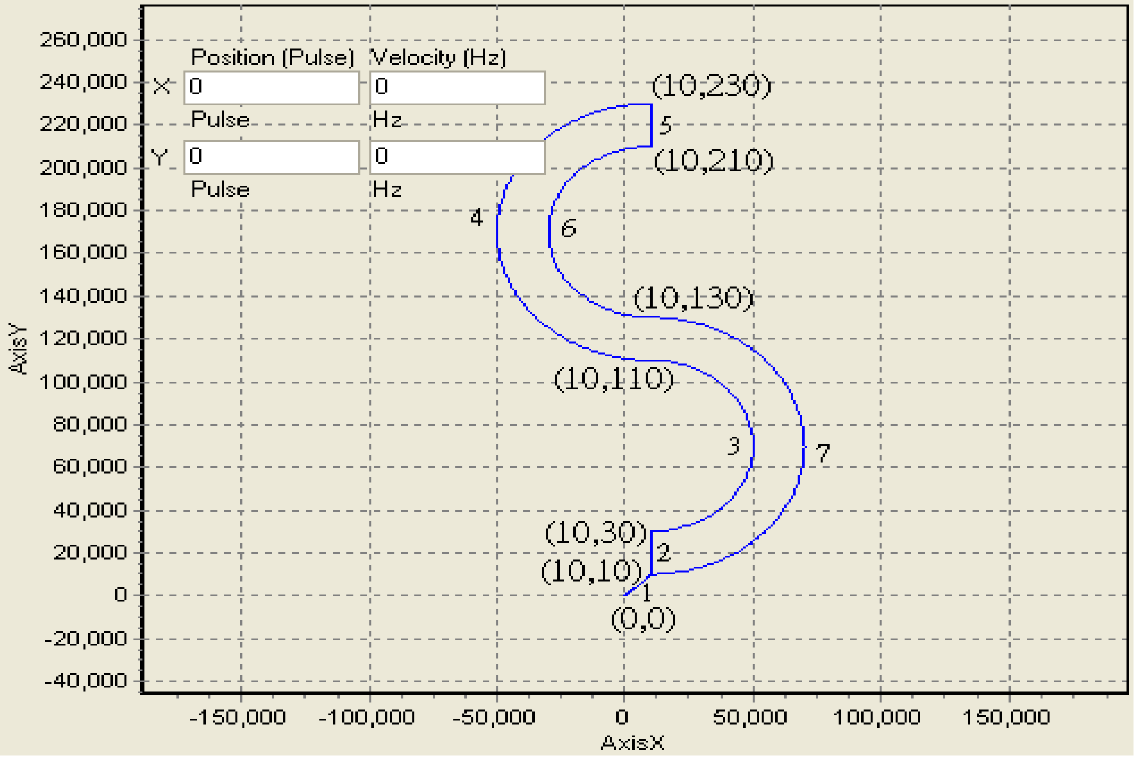
Path 1



Path 2



Path 3



Path 4

8.1.2 Steps

1. Path 1: Set up the absolute coordinates of the four points (-20, 20), (60, 20), (60, 100) and (-20, 100). Start from (0, 0).
2. Path 2: Set up the absolute coordinates of the four points (-10, 10), (20, 10), (20, 70) and (-10, 70). Start from (0, 0).
3. Path 3: Set up the absolute coordinates of the three points (-25, 25), (25, 25) and (0, 85). Start from

(0, 0).

4. Path 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

Instruction mode: Design instructions for initializing the present status of the X-axis/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 path 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				

Example of G-code instructions for path 1: Design a motion program in a pointer

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 path 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 path 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 path 3					
P0					P0 subroutine
INCT					Set up relative coordinate
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 path 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 path 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					

Example of G-code instructions for path 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. If M1072 in a DVP-20PM series motion controller is ON, the DVP-20PM series motion controller will be made to run, and the positioning instructions above will be executed.

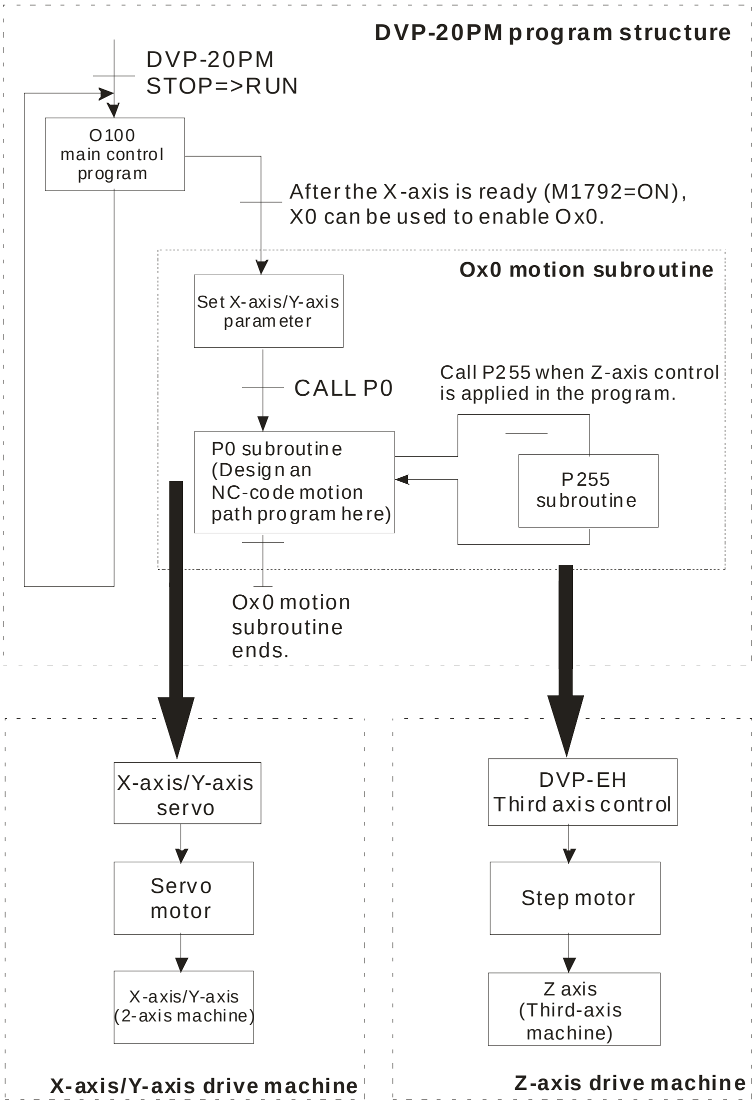
8.2 Applying Application Examples in PMSoft

Users can apply the application example “motionSample” to draw English letters, any graph or text. If the users wish to apply this function to any two-axis control equipment, they can modify the example program below for them 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-20PM for drawing, users need to convert letters or graphs into G-Codes (i.e. NC code) before designing the main control program. In addition, DVP-20PM00D only offers X-axis/Y-axis interpolation, the users have to apply additional Z-axis control for the “pen-lifting” function. In this example, DVP-EH is used to complete the third axis control. (Other controllers are also applicable).

The program structure designed is shown below.



8.2.2 Designing an Example Program

To design the example program, users can divide the program into four sections including OX0~M2, O100~M102, P255~SRET and P0~SRET explained below.

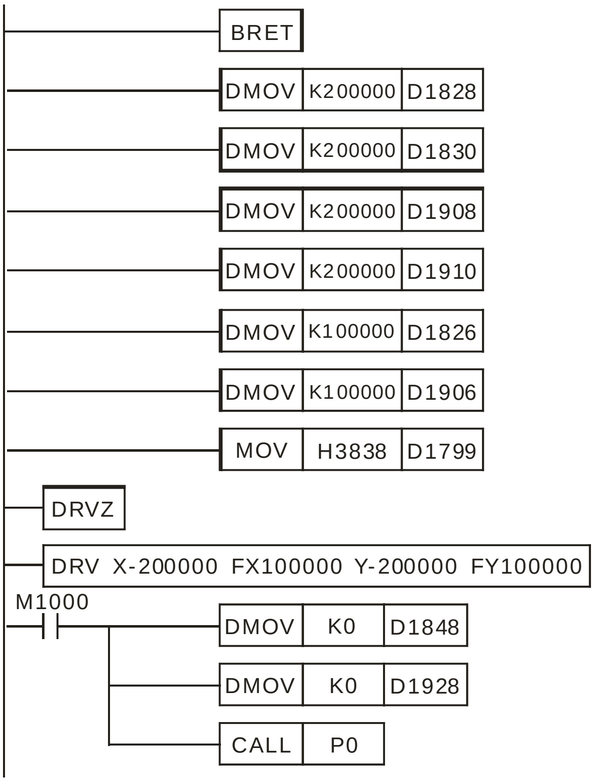
1. OX0~M2: Setting the function parameters of X-axis and Y-axis

   When DVP-20PM runs and motion subroutine (OX) is ready (M1792 = ON), set ON X0 to enable OX0 motion subroutine. OX0 will set up parameters for zero return, JOG speed, and input terminal polarity on X-axis/Y-axes.

   Next, enable zero return and move to (-200000, -200000) on X-axis/Y-axis by 100 kHz then reset the current position to 0 and call P0 subroutine. OX0 subroutine will end when the execution of P0 subroutine is completed.

   If users need to set up other control modes, please refer to explanations on special registers in Chapter 3.

   Ladder diagram: Operation:

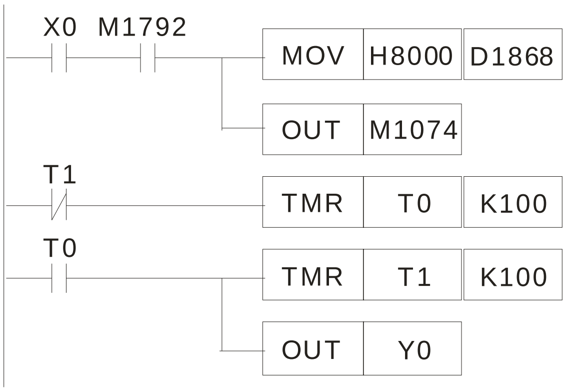


Zero return speed (VRT) of the X-axis = 200kHz
Zero return deceleration speed (VCR) of the X-axis = 200kHz
Zero return speed (VRT) of the Y-axis = 200kHz
Zero return deceleration speed (VCR) of the Y-axis = 200kHz
JOG speed of the X-axis = 100kHz
JOG speed of the Y-axis =100kHz
Set up input terminal polarity of the X-axis/Y-axis
Enable zero return on the X-axis/Y-axis
Move to (-200000,-200000) on the X-axis/Y-axis by 100kHz.
Clear the current position of the X-axis to 0.
Clear the current position of the Y-axis to 0.
Call the subroutine P0.

2. O100~M102: Main program control

   O100 main program controls whether to enable OX0 subroutine. When X0 (drive contact for enabling OX0) and M1792 (indicating ready status of OX) in the program are ON, OX0 subroutine will be enabled. You can further place other operations in the main program.

   Ladder diagram: Operation:

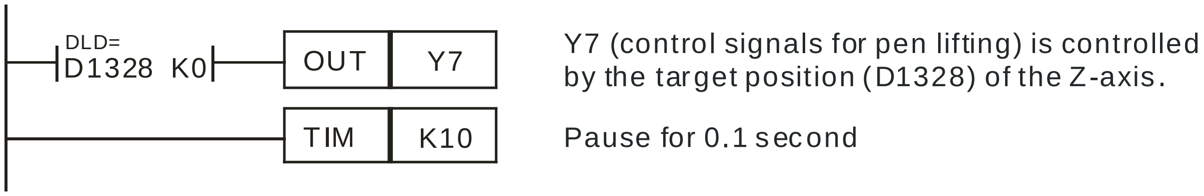


When OX is ready (M1792 = ON), OX0 motion subroutine can be enabled by X0.
Enable OX0 motion subroutine
Other operations
Other operations
Other operations

3. P255~SRET: Setting the third axis (Z-axis) control system

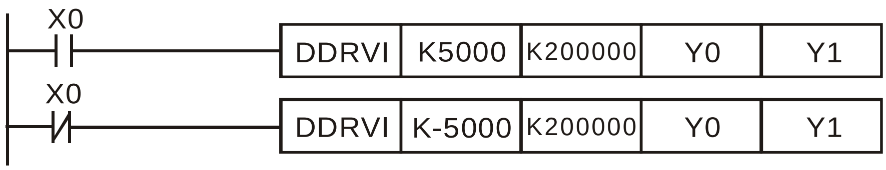
   The third axis control will activate according to the parameter setting of G00 and G01 instructions. When G00/G01 sets target position of the Z-axis (D1328), P255 will be enabled to drive Y7 (third axis control signal) and enable DVP-EH to control the lift or drop of a pen (i.e. up/down movement of the Z-axis). For more details, please refer to G00 and G01 instructions in Chapter 6.

   Ladder diagram: Operations:



   The third-axis control signal (Y7) in P255 of DVP-20PM drives the input contact X1 of DVP-EH. If X1 is ON, the step motor controlled by pulses will lift the pen through DDRVI. If X1 is OFF, the step motor will drop the pen through DDRVI.

   Program in DVP-EH:

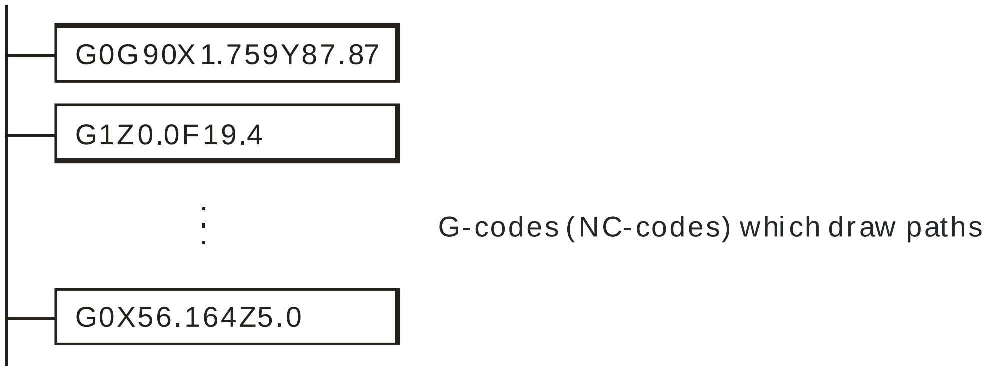


   Connect the input devices Y0 and Y1 in DVP-EH to the pulse input terminals on the step motor.

4. P0~SRET: Two-axis (X-axis/Y-axis) 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 three sections above and the drawing of letters or graphs can be performed.

   Ladder diagram: Operations:



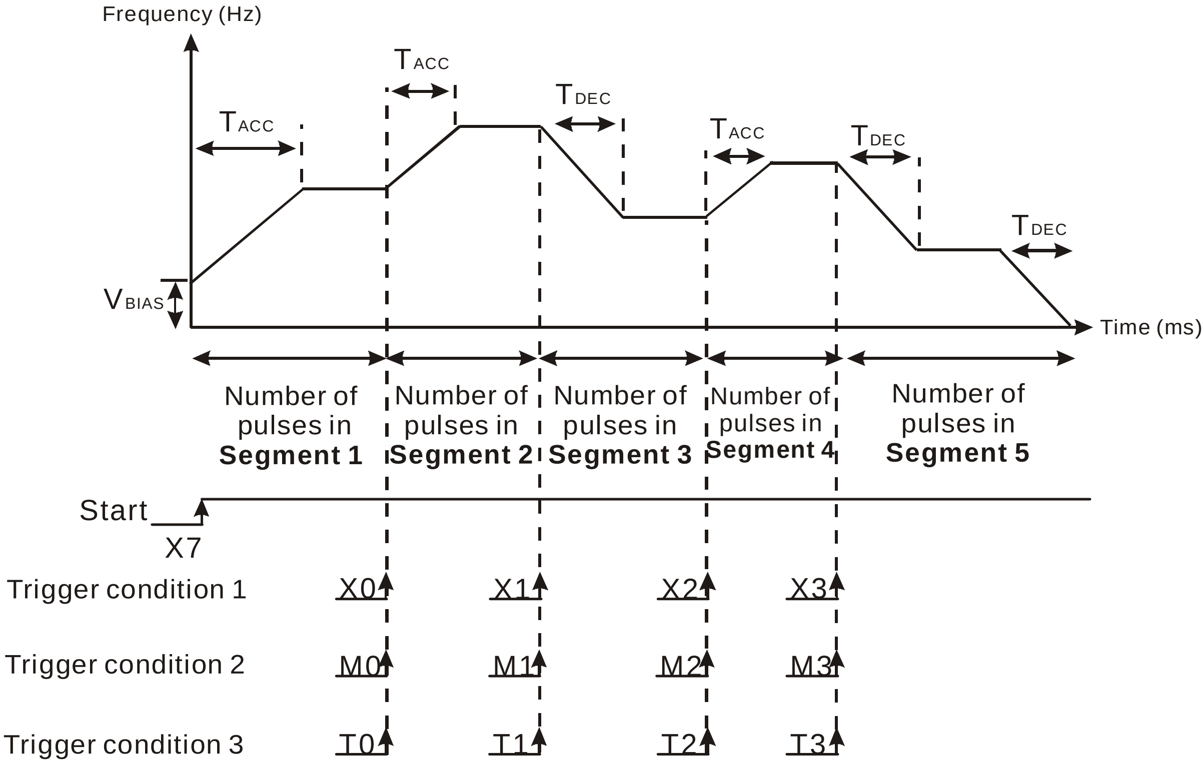
   When program section 1~program section 4 are completed, drawing of letters, graphs or any texts by DVP-20PM can be performed by DVP-20PM.

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 switch to the second speed~the fifth speed.
2. Trigger condition 2: Comparison results of the present position. M0~M3 switch to the second speed~the fifth speed.
3. Trigger condition 3: The variable speed is controlled by time. T0~T3 switch to the second speed~the fifth speed.



DD1838 (total number of output pulses)=Number of pulses in segment 1+Number of pulses in segment 2+…+Number of pulses in segment 5

8.3.2 Designing Programs

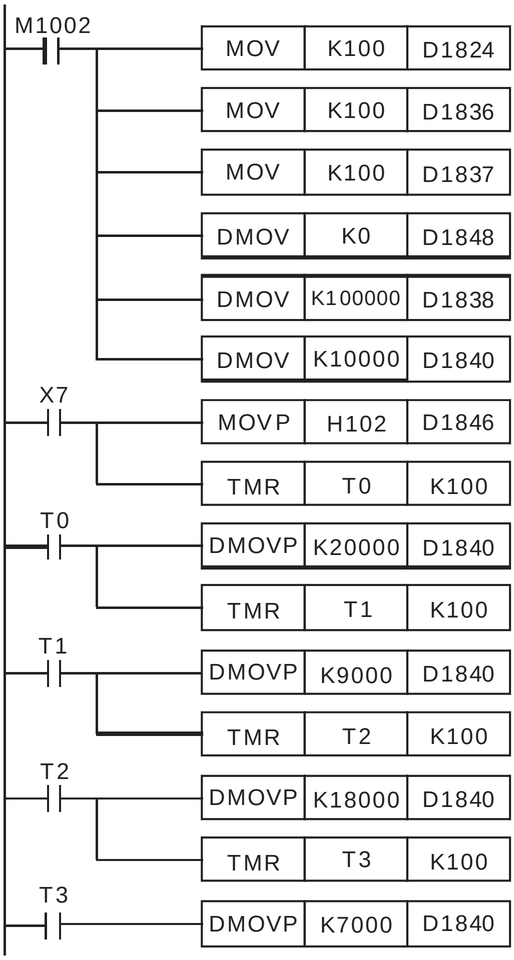
Ladder diagram of trigger condition 1:

				Operation
M1002	MOV	K100	D1824	Setting the bias speed of the X-axis (V_{BIAS})
	MOV	K100	D1836	Setting the acceleration time of the X-axis (T_{ACC})
	MOV	K100	D1837	Setting the deceleration time of the X-axis (T_{DEC})
	DMOV	K0	D1848	Clearing the present position of the X-axis to 0
	DMOV	K100000	D1838	Setting the total distance of all segments for the X-axis
	DMOV	K10000	D1840	Setting the operation speed for the first segment on the X-axis
X0	DMOVP	K20000	D1840	X0=ON: Modifying the operation speed into 20,000Hz
X1	DMOVP	K9000	D1840	X1=ON: Modifying the operation speed into 9,000Hz
X2	DMOVP	K18000	D1840	X2=ON: Modifying the operation speed into 18,000Hz
X3	DMOVP	K7000	D1840	X3=ON: Modifying the operation speed into 7,000Hz. Pulse output will be finished in this segment.
X7	DMOVP	H102	D1846	X7=ON: Enabling the single speed postioning mode of the X-axis.

Ladder diagram for trigger condition 2:

				Operation
M1002	MOV	K100	D1824	Setting the bias speed of the X-axis (V_{BIAS})
	MOV	K100	D1836	Setting the acceleration time of the X-axis (T_{ACC})
	MOV	K100	D1837	Set up the deceleration time of the X-axis (T_{DEC})
	DMOV	K0	D1848	Clearing the current position of the X-axis to 0
	DMOV	K100000	D1838	Setting the total distance of all segments for the X-axis
	DMOV	K10000	D1840	Setting the operation speed for the first segment on the X-axis
D>= D1848 K15000	SET	M0		Comparing K15000 with the present position
M0	DMOVP	K20000	D1840	M0=ON: Modifying the operation speed into 20,000Hz
D>= D1848 K30000	SET	M1		Comparing K30000 with the present position
M1	DMOVP	K9000	D1840	M1=ON: Modifying the operation speed into 9,000Hz
D>= D1848 K50000	SET	M2		Comparing K50000 with the present position
M2	DMOVP	K18000	D1840	M2=ON: Modifying the operation speed into 18,000Hz
D>= D1848 K80000	SET	M3		Comparing K80000 with the current position
M3	DMOVP	K7000	D1840	M3=ON: Modifying the operation speed into 7,000Hz. Pulse output will be finished in this segment.
X7	MOVP	H102	D1846	X7 = ON: Enabling the single speed positioning mode of the X-axis

Ladder diagram for trigger condition 3:



Operation:

Setting the bias speed of X axis (V_{BIAS})

Setting the acceleration time of X axis (T_{ACC})

Setting the deceleration time of X axis (T_{DEC})

Clearing the present position of the X-axis to 0

Setting the total distance of all segments for the X-axis

Setting the operation speed for the first segment on the X-axis

X7=ON: The single speed positioning mode on the X-axis is enabled.
T0 begins to count, and prepares to switch to the second segment.

T0=ON: The operation speed is modified into 20,000Hz.
T1 begins to count, and prepares to switch to the third segment.

T1=ON: The operation speed is modified into 9,000Hz.
T2 begins to count, and prepares to switch to the fourth segment.

T2=ON: The operation speed is modified into 18,000Hz.
T3 begins to count, and prepares to switch to the fifth segment.

T3=ON: The operation speed is modified into 7,000Hz.
Pulse output will be finished in this segment.

8.4 Connecting DVP20PM00D (as a Master) and DVP01PU-H2 (as a Slave) for Controlling the Third Axis

Operation:

1. Enable O100 and execute OX0.
2. When the execution encounters G01 Z-25000 F10000 in OX0 subroutine, the program calls 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 and subroutines:

Instruction mode: Design instructions for initializing the present status of the X-axis/Y-axis and enabling OX0 motion subroutine.

O100 main program

(O100)			O100 main program
LD	M1002		
MOV	H8000	D1868	Write in the No. (0) of OX to be enabled
SET	M1074		Enable OX motion subroutine
M102			

OX0 subroutine

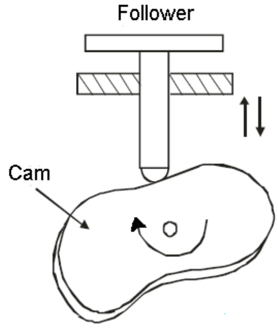
(OX0)	OX0 subroutine

G01	Z-25000	F10000			G01 third axis control
TIM	K1000				Pause for 10 seconds
G01	Z10000	F20000			G01 third axis control
M2					
P255 subroutine					
(P255)					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					

9.1 Introduction of Electronic Cams

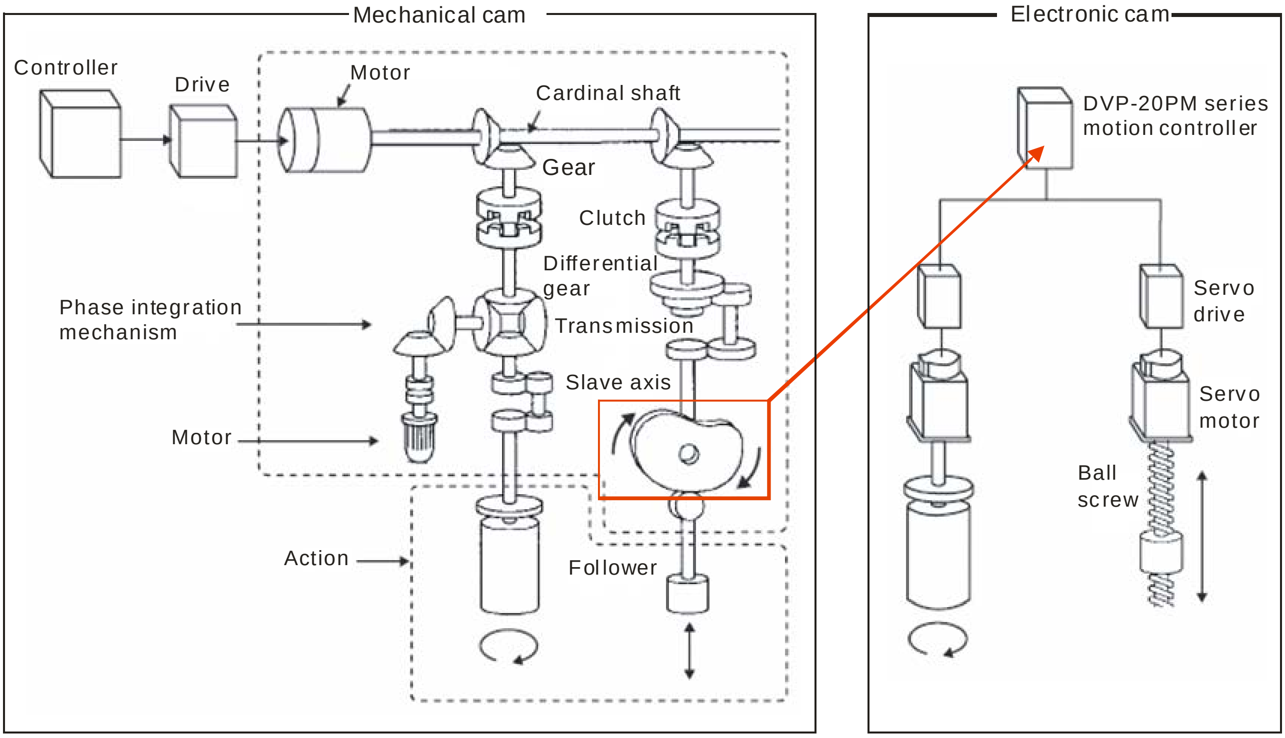
A traditional mechanical cam is composed of a cam, a follower, and a support.

- A mechanical cam is a rotating sliding piece with irregular shape. In general, it is an input object which rotates at a uniform speed. It makes a follower move regularly by coming into contact with the follower.
- A follower is a part driven by a mechanical cam. In general, it is an output object which generates motion which is not uniform, sequential, and regular motion.
- A support is a piece that which is used to support a mechanical cam and a follower.



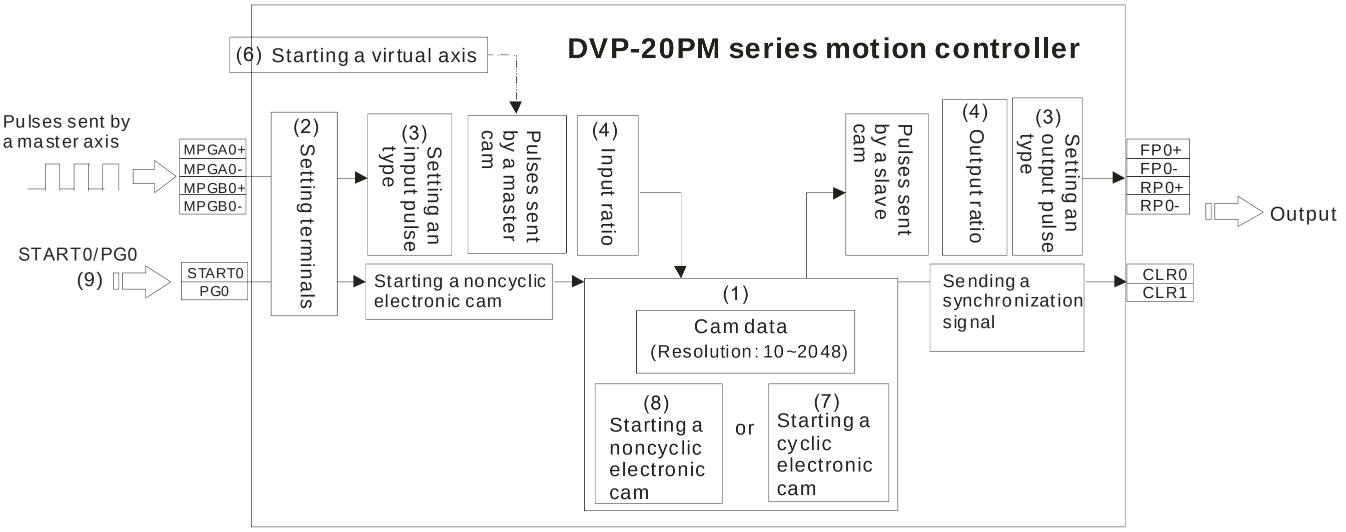
Compared with a traditional cam, an electronic cam has the following advantages.

- Friendlier user interface
- Different products require different cam curves. Users can modify the electronic cam data in an electronic cam in software. They do not need to modify a mechanism.
- High acceleration
- Smoother operation



9.2 Operation of an Electronic Cam

■ Structure



■ Steps

Step 1	Step 2	Step 3
Initial setting	Setting a master axis	Starting/Stopping an electronic cam
(1) Creating electronic cam data (2) Setting terminals (3) Setting an input/output pulse type (4) Setting an input/output ratio Setting a starting angle	Servo encode for the master axis specified Pulse signals sent by the master axis specified (6) Starting a virtual axis	(7) Starting/Stopping a cam which operates cyclically (8) Starting/Stopping a cam which does not operate cyclically

9.2.1 Initial Setting

9.2.1.1 Creating Electronic Cam Data

There are two methods of creating electronic cam data.
Method 1: Use DTO instruction to create electronic cam data directly
Method 2: Use the electronic cam chart in PMSoft to draw the cam curve
Please refer to section 9.4 for more information.

9.2.1.2 Setting Terminals

■ Input terminals

1. MPGA0/MPGB0: Pulse input terminal for Master. Max allowable frequency: 200 kHz
2. START0/PG0: Input terminal for enabling noncyclic electronic cam

■ Output terminals

1. FP/RP: Output terminal for pulse output of electronic cam. Max output frequency: 500 kHz
2. CLR0/CLR1: Output terminal for electronic cam synchronized output signal. When D1839, D1838 (PI) ≦ CP (Current Position) of Master (X axis) ≦ D1843, D1842 (PII), CLR0/CLR1 will be ON. (For special application of CLR0/CLR1, please refer to 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, users 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).

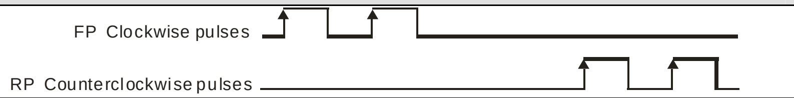
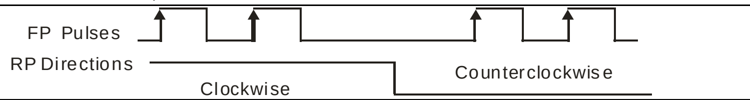
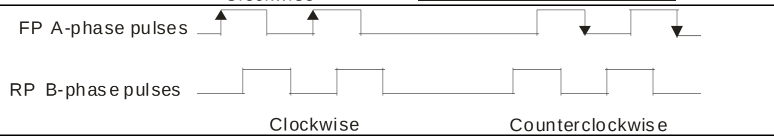
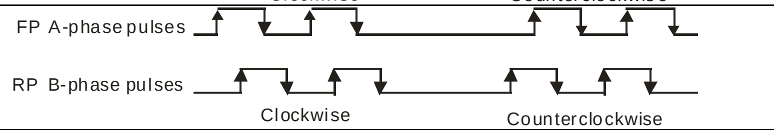
N	kHz	N	kHz
1	2656.25	11	2.593994
2	1328.125	12	1.296997
3	664.0625	13	0.648499
4	332.0313	14	0.324249
5	166.0156	15	0.162125
6	83.00781	16	0.081062
7	41.50391	17	0.040531
8	20.75195	18	0.020266
9	10.37598	19	0.010133
10	5.187988		

9.2.1.3 Setting an Input/Output Pulse Type

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.

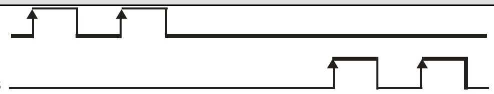
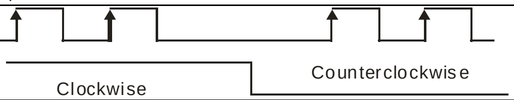
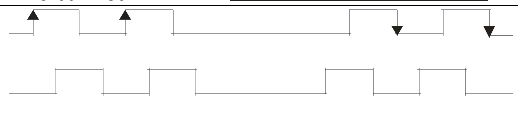
- Setting an input pulse type

MPG pulse input type setting: D1864

b9	b8	Input pulse type (positive logic)	Description
0	0		Counting up/down
0	1		Pulses+Directions
1	0		A/B-phase pulses
1	1		Four times the frequency of A/B-phase pulses

■ Setting an output pulse type

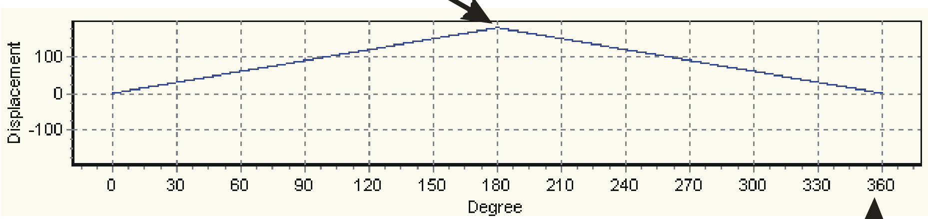
Pulse output type setting: D1816, D1896, and D1976

b5	b4	Output pulse type (positive logic)	Description
0	0		Counting up/down
0	1		Pulses+Directions
1	0		A/B-phase pulses
1	1		

9.2.1.4 Setting an Input/Output Ratio

DVP-20PM provides electronic cam data magnification function. When users input electronic cam data as the below diagram, electronic 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.

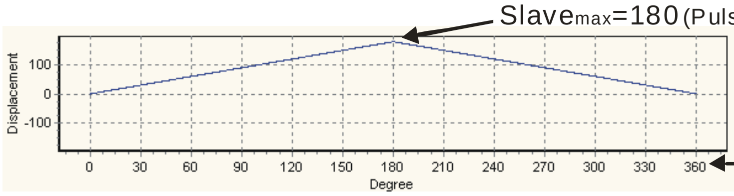
Slave$_{max}$=180(Slave unit)



Master$_{max}$=360(Master unit)

D1816		
b1	b0	Unit
0	0	Motor unit

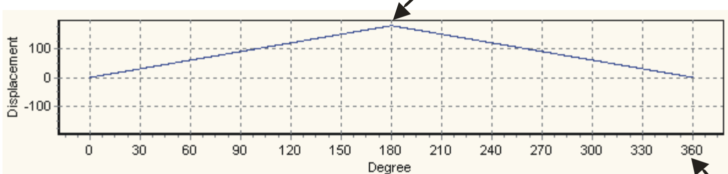
Slave$_{max}$=180 (Pulses)



Master$_{max}$=360(Pulses)

D1816		
b1	b0	Unit
0	1	Mechanical unit
1	0	Compound unit
1	1	Compound unit

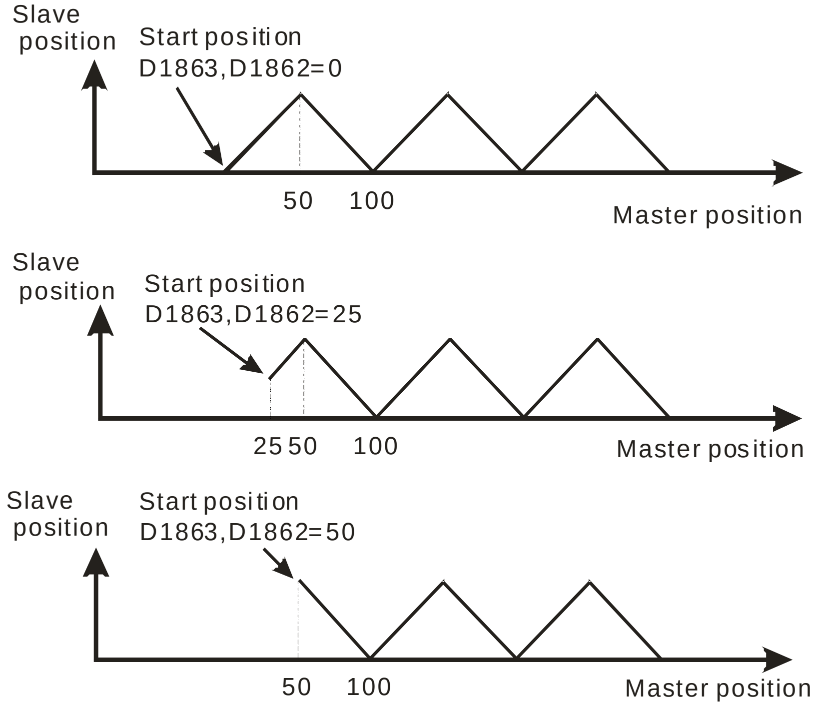
$$\frac{\text{Slave}_{max}=180 \times \text{D1819.D1818 (Pulses per revolution)}}{\text{D1821.D1820 (Displacement per revolution)}} \text{(Pulses) = Output ratio}$$



$$\frac{\text{Master}_{max}=360 \times \text{ D1858 Numerator of electronic gear ratio}}{\text{D1859 Denominator of electronic gear ratio}} \text{(Pulses) = Input ratio}$$

9.2.1.5 Setting a Starting Angle

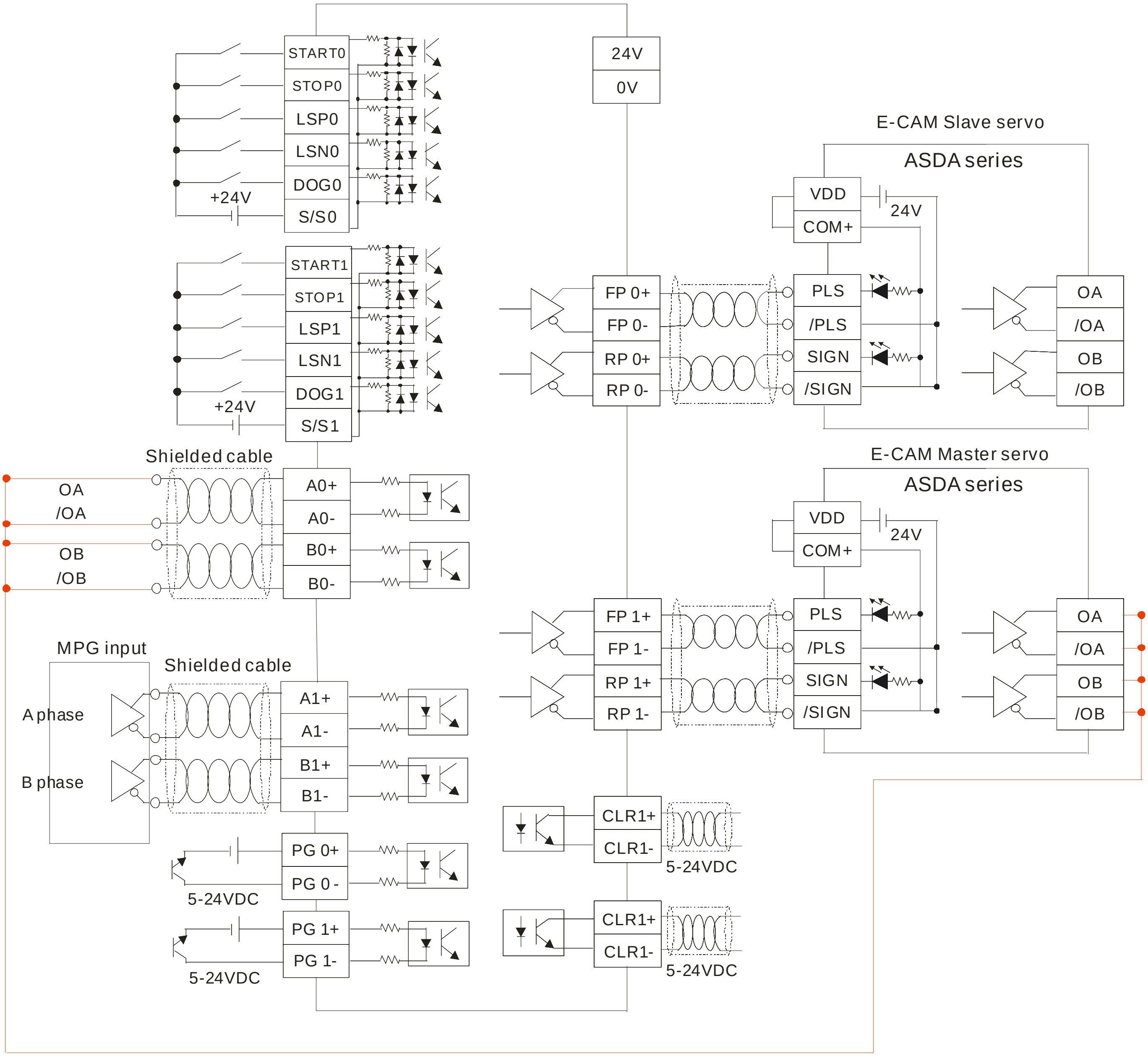
Users can set M1752=ON to enable electronic cam activation at non-zero position, and write the start position of Master in D1862, D1863.



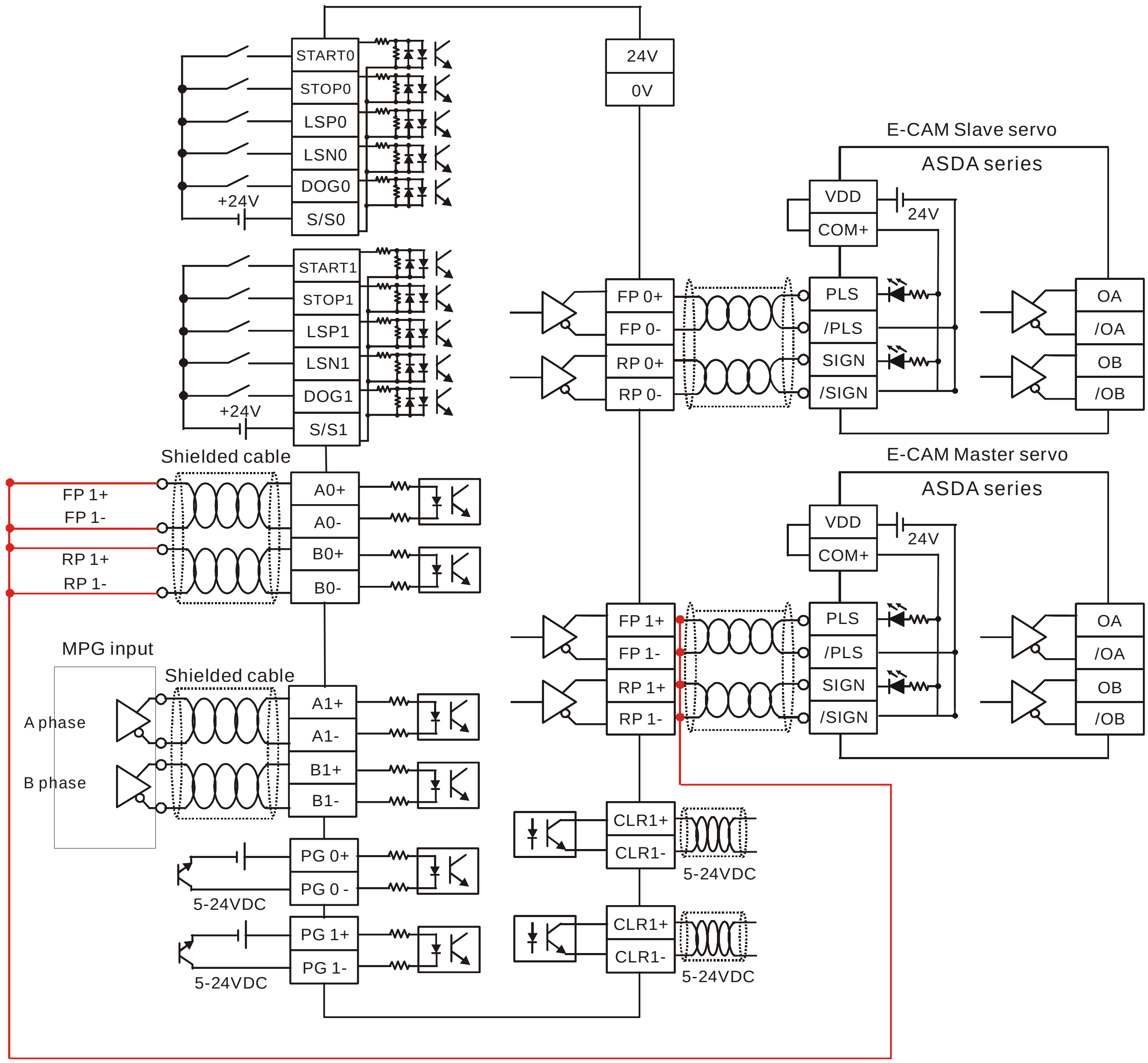
9.2.2 Setting a Master Axis

There are 3 methods to obtain Master position.

Method 1: Obtain Master position from the encoder of Master servo. Receive the signals from the encoder then convert the signals into Master position.



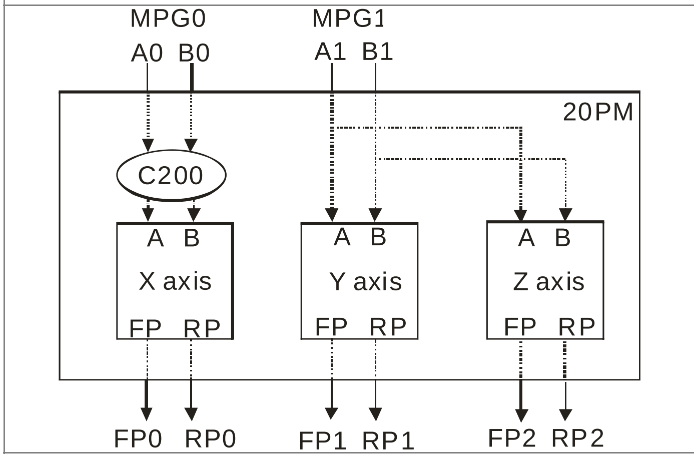
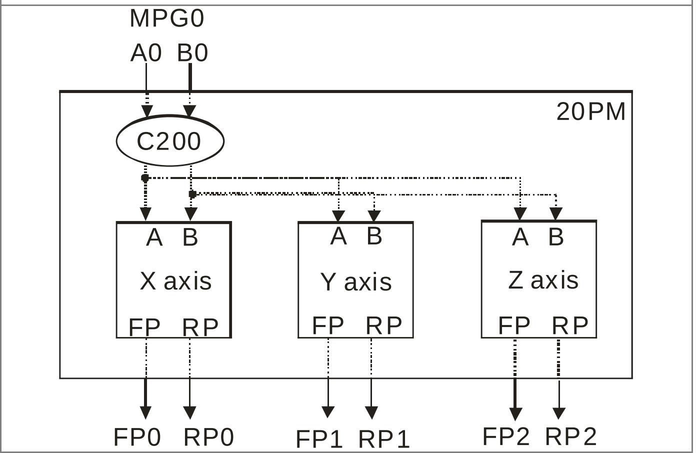
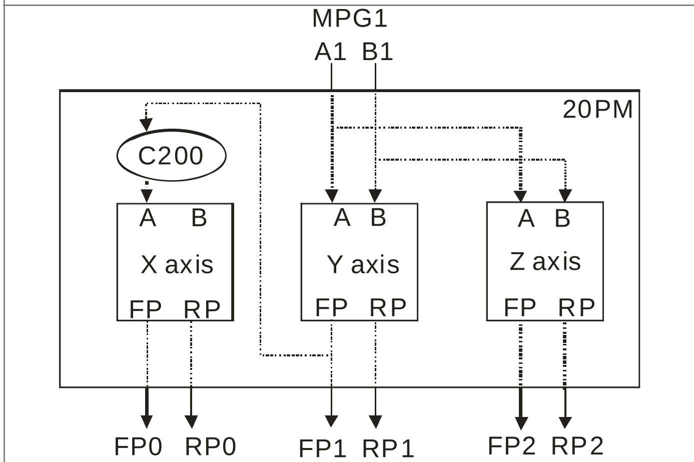
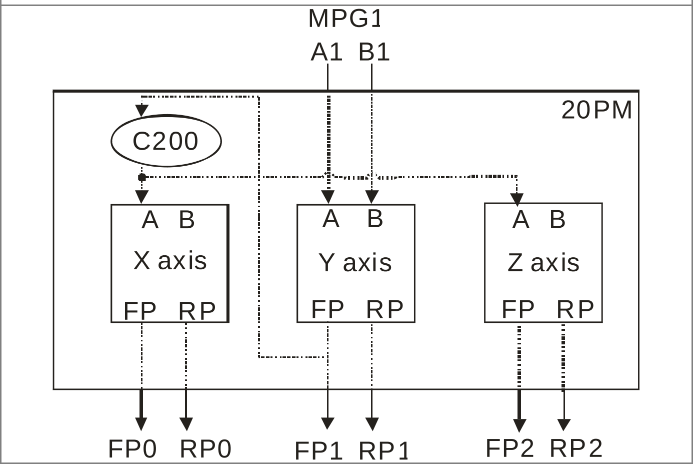
Method 2: Obtain Master position from the pulses sent by Master. Process the pulse signals sent by Master then convert the signals into Master position.



Method 3: Apply Virtual Master

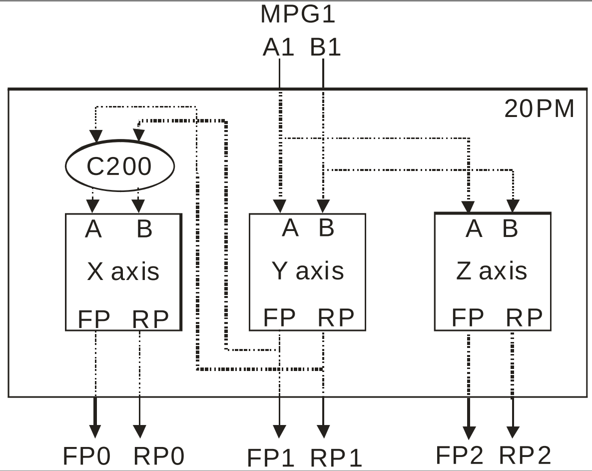
In virtual Master mode, users can catch the output signals from internal Y axis as the signal source of Master for X axis or Z axis. In this case, users don't have to apply additional wirings. To apply virtual Master mode, set ON M1909 and A0 will be internally linked with FP of Y axis; set ON M1910 and B0 will be internally linked with RP of Y axis. After the 2 flags are ready, DVP-20PM 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-20PM).

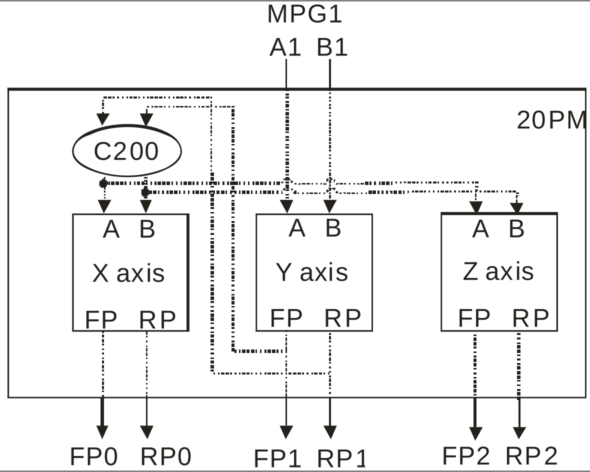
In addition, care should be taken when setting the pulse type of MPG and the pulse type of Y axis. Pulse types of the two sides should be the same. Please refer to 9.2.1.3 for the setting.

<table>
<tr><th>Single cam</th><th>Multiple cam</th></tr>
<tr><td colspan="2">■ M1909=OFF, M1910=OFF</td></tr>
<tr><td>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.</td><td>The source signal of X, Y, Z axis is from A0, B0 input of MPG0.</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">■ M1909=ON, M1910=OFF</td></tr>
<tr><td>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.</td><td>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.</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

■ M1909=ON, M1910=ON

The source signal of X axis is from FP1, RP1 (Y axis as virtual Master), and the source signal of Y, Z axis is from A1, B1 input of MPG1.	The source signal of X, Z axis is from FP1, RP1 (Y axis as virtual Master), and the source signal of Y axis is from A1, B1 input of MPG1.



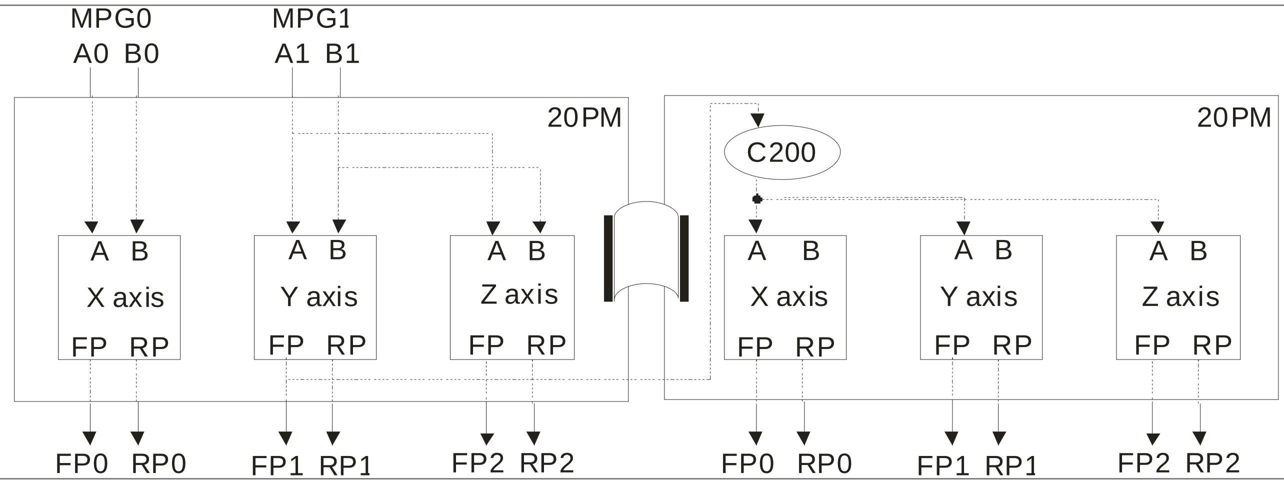


■ Setting a slave: M1909=Off, M1910=On

Take the FP signals of the master station as the sources of the master axis signals through communication. If M1910 is ON, signals will be sent to the slave station in accordance with the setting of the virtual axis in the master station.

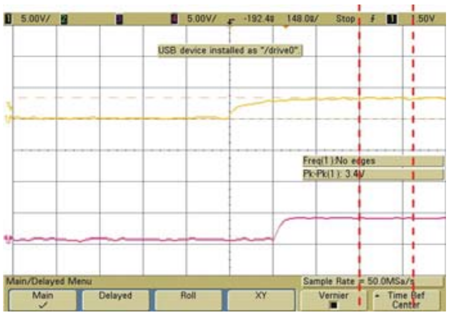
Example 1: If M1910 is ON, and M1910 is ON, FP1 and RP1 on the Master station will send signals to C200 in the slave station as the sources of the master signals.

Example 2: If M1910 is ON, and M1910 is OFF, FP1 and RP1 on the master station will send signals to C200 in the slave station as the sources of the master signals.

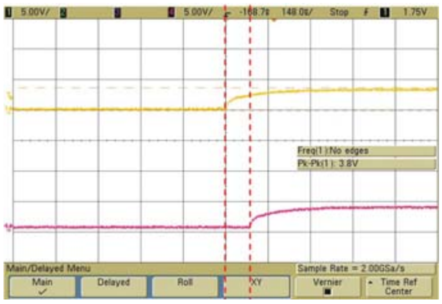


【Note】 According to the statistics below, there will be a delay of a 500 kHz pulse (2us) if 29 DVP-20PM series motion controllers are connected. Besides, the signal passing through each DVP-20PM series motion controller is not weakened.

Two DVP-20PM series motion controllers:
There is a delay of 0.148 us.



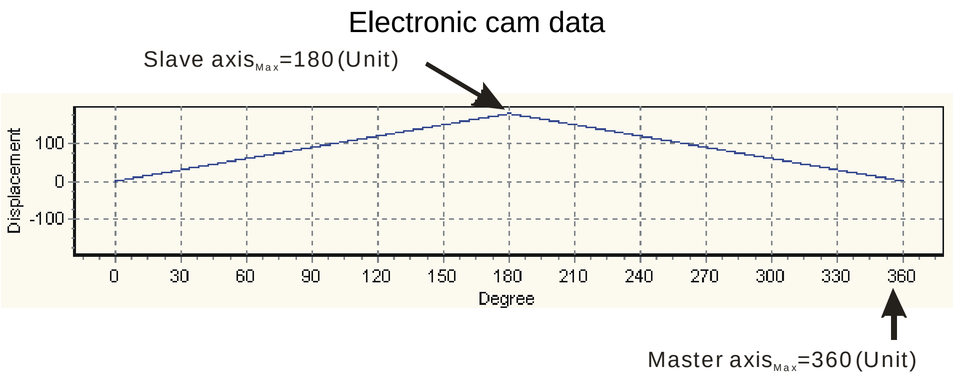
One DVP-20PM series motion controller:
There is a delay of 0.07 us.



9.2.3 Starting / Stopping an Electronic Cam

9.2.3.1 Starting / Stopping a Cyclic Electronic Cam

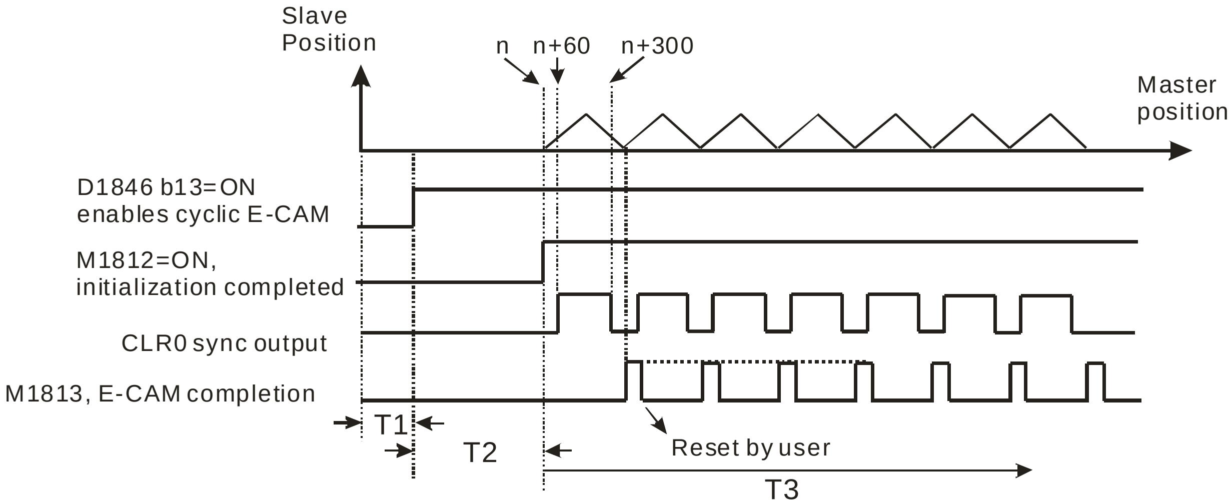
If an electronic cam operates cyclically, the slave axis of the electronic cam moves in accordance with electronic cam data when the master axis of the electronic cam moves. Electronic cam data defines only one cycle. The relation between the positions of a master axis and the positions of a slave axis is the repeated extension of electronic cam data.



- Starting a cyclic electronic cam

  The steps of starting a cyclic electronic cam are as follows.

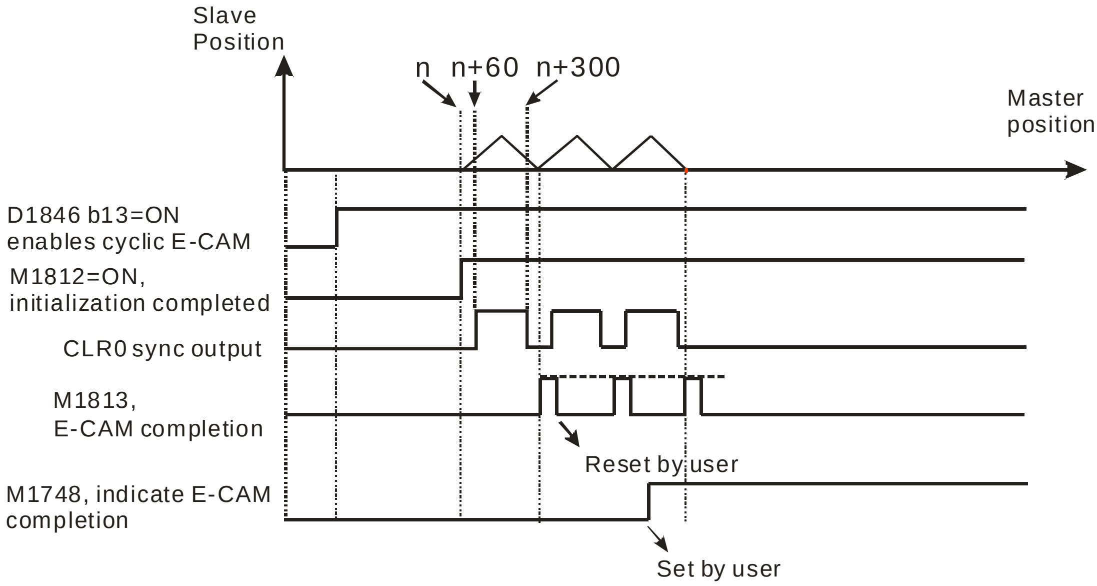
  1. In T1, D1846 bit 13=ON, cyclic electronic cam is enabled.
  2. After the initialization interval of T2, M1812=ON to indicate the completion of electronic cam initialization.
  3. In T3, cyclic operation starts after M1812=ON. Slave executes with Master constantly according to the electronic cam data.
  4. At the same time, CLR0 outputs according to the synchronized output range. When M1813=ON, an electronic cam cycle is completed. User has to reset the flag for indicating the completion of other electronic cam cycles.



- Stopping a cyclic electronic cam

  The steps of stopping a cyclic electronic cam are as follows.

  1. When D1846 bit 13 (enabling cyclic electronic cam)=OFF, the X-axis of electronic cam stops immediately.
  2. When M1748 (indicate electronic cam completion) is ON during electronic cam execution, Slave of electronic cam will stop until the current electronic cam cycle is completed.



- Example

  【Function】

  The figure shown below is electronic cam data.

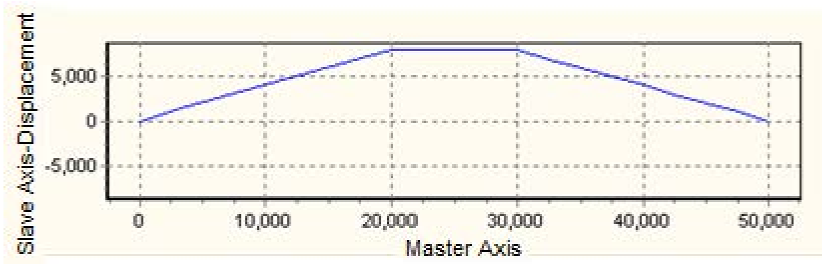
  Electronic cam cycle: Master position 0~50000

  Control unit: Motor system

  Sync output: Master position 20000~30000

  CLR0=ON: Control signal.

  Please refer to section 9.4 for more information about creating electronic cam data. Connect the Y-axis pulse output terminals (FP+, FP-, RP+, RP-) to the X-axis MPG input terminals (A0+, A0-, B0+, B0-). In this case, the input signals of the master axis are from output signals of the Y-axis.



In this example, the electronic cam data is created by PMSoft. Please refer to section 9.4 for more information about creating electronic cam data. Connect the Y-axis pulse output terminals (FP+, FP-, RP+, RP-) to the X-axis MPG input terminals (A0+, A0-, B0+, B0-). In this case, the input signals of the master axis are from output signals of the Y-axis.

【Steps】

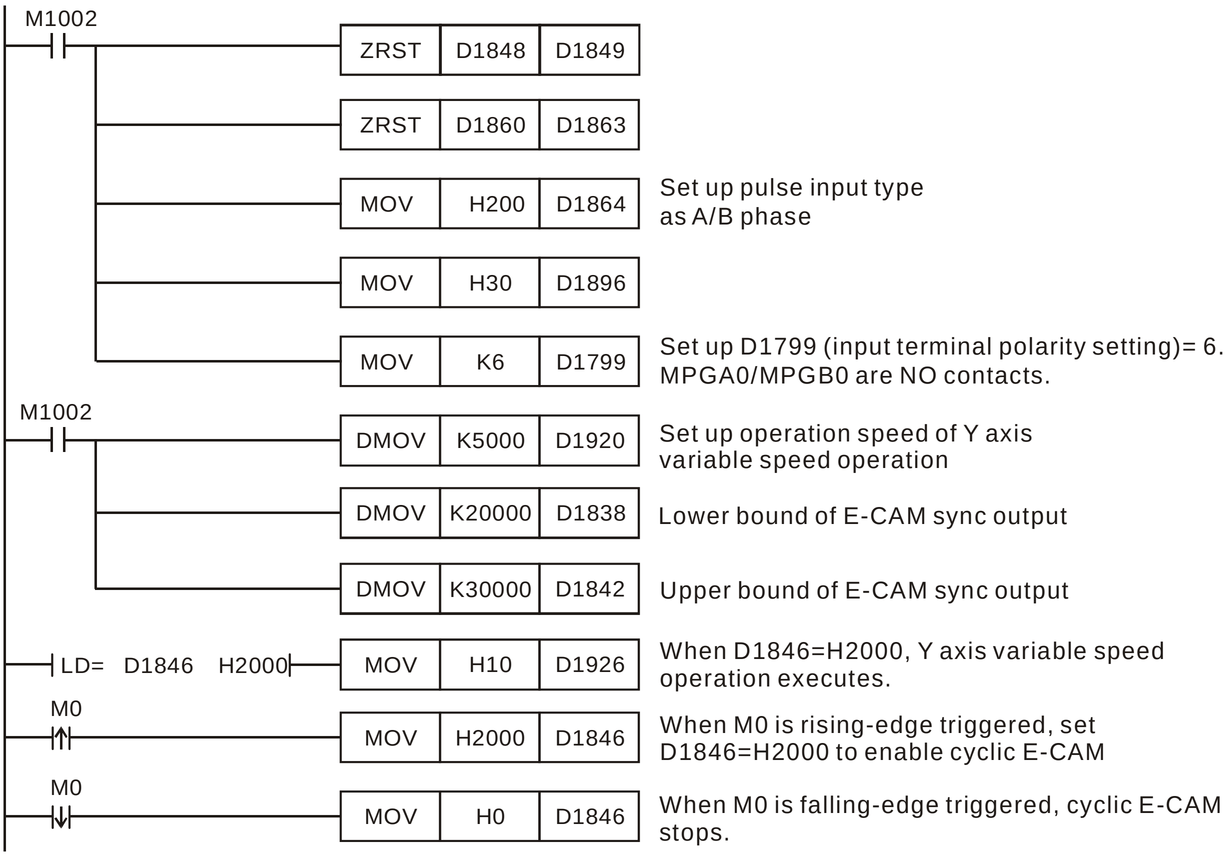
1. Set special registers. The parameters set are as follows.
   (1) Clear the content in registers D1848, D1849, D1862, D1863, D1868.
   (2) Set up input pulse type as A/B phase (D1864=H200)
   (3) Set up pulse output type of Y axis as A/B phase (H30)
   (4) Set up D1799 (input terminal polarity setting)=6. 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, users can connect CLR0- to 0V, connect CLR0+ to Xn input contact, connect S/S2 to 24V, and then monitor the ON/OFF status of Xn input.

2. Set M0 to ON to enable cyclic electronic 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 electronic cam data curve. In addition, CLR0=ON during Master position 20000~30000.
3. Reset M0.

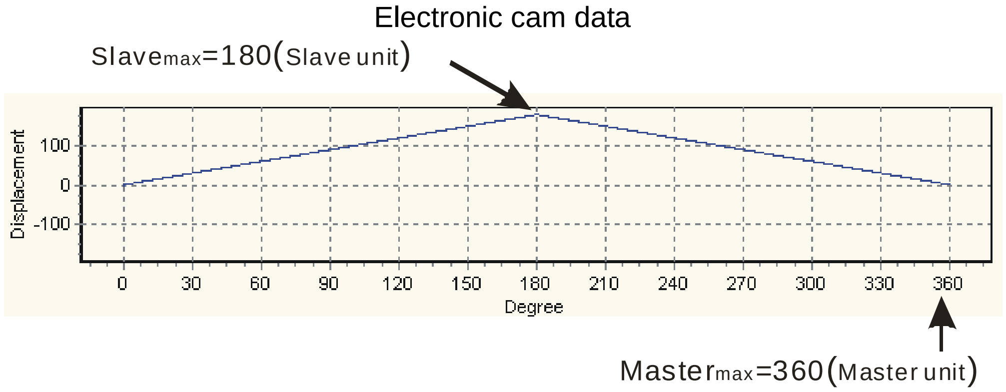
【Program】



9.2.3.2 Starting / Stopping a Noncyclic Electronic Cam

In noncyclic electronic cam, Master and Slave operate according to the user-defined electronic cam data only when electronic cam start signal (START0/PG0) is triggered. Unlike cyclic electronic cam, noncyclic electronic cam operates only one cycle for each triggered signal, i.e. electronic cam data operates only once for one triggered signal.

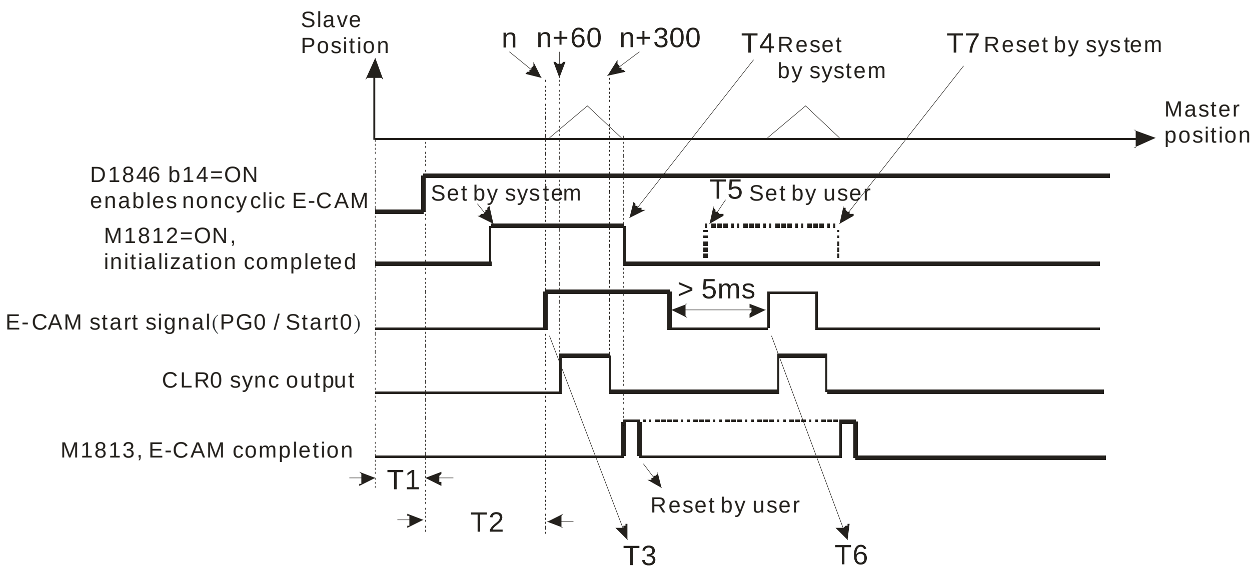
Before selecting START0 as the start signal, M1035 has to be set ON for setting STOP0/START0 as the external input. After M1746=OFF START0 will be enabled as the start signal of noncyclic electronic cam rather than PG0.



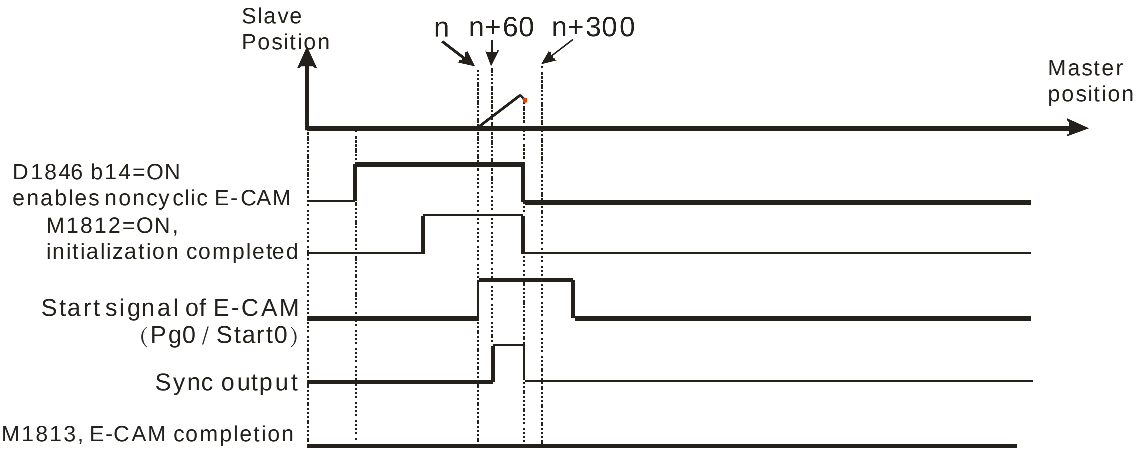
- Starting a noncyclic electronic cam

  The steps of starting a noncyclic electronic cam are as follows.

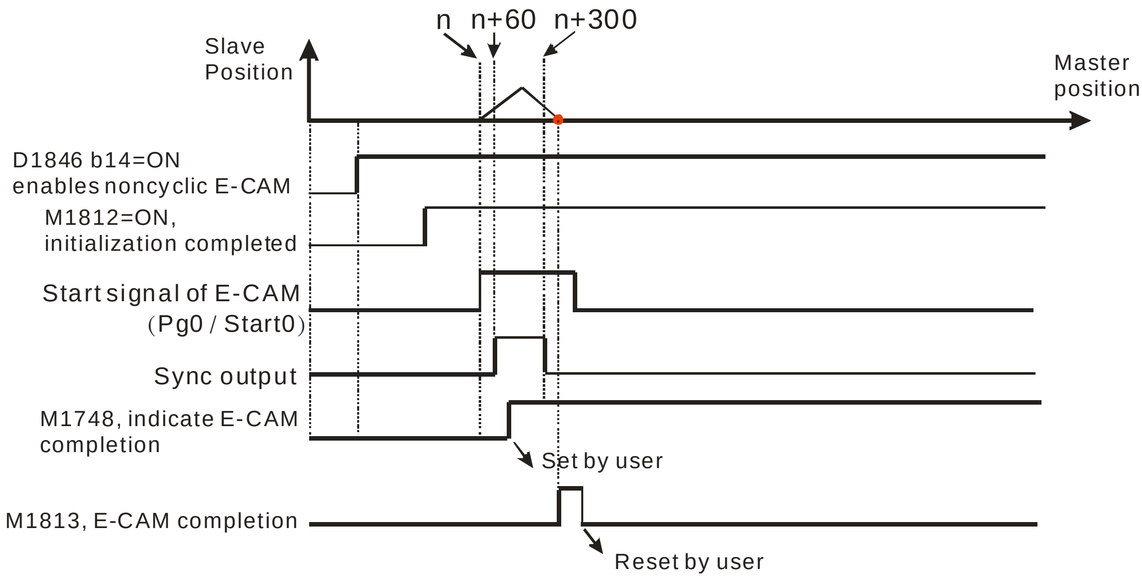
  1. In T1, D1846 bit 14=ON, noncyclic electronic cam is enabled
  2. After the initialization interval of T2, M1812=ON to indicate the completion of electronic cam initialization. At this time Slave is still not activated.
  3. In T3, noncyclic operation starts after START0/PG0 (Input terminal for enabling noncyclic electronic cam) is ON. When M1746=ON, PG0 is the start signal of noncyclic electronic cam; when M1746=OFF and M1035=ON, START0 is the start signal of noncyclic electronic cam. In this case Slave starts to operate a cycle according to the electronic cam data. ("M1035=ON" enables START0/STOP0 as the external input point).
  4. When the cycle is completed at T4, DVP-20PM will clear the state of M1812=ON. In addition, users can also confirm the electronic cam completion by ON state of M1813.
  5. In T5, users can decide whether to set ON M1812 for confirming the next electronic cam completion.
  6. In T6 and T7, actions of T3~T4 will be repeated. Please note that the interval for enabling noncyclic electronic cam should be longer than 5ms.
  7. CLR0 outputs according to the upper/lower bound of sync output during each electronic cam cycle.



- Stopping a noncyclic electronic cam
  1. When D1846 bit 14 (enabling noncyclic electronic cam)=OFF, Slave of electronic cam stops immediately.



  2. When M1748 (indicate electronic cam completion) is ON during electronic cam execution, Slave of electronic cam will stop until the current electronic cam cycle is completed.



- Example

【Function】

The figure shown below is electronic cam data.

Electronic cam cycle: Master position 0~100000

Control unit: Motor system

Every time when START0 is triggered, execute 3 noncyclic electronic cam data (D1832=2)

In this example, the electronic cam data is created by PMSoft. Please refer to section 9.4 for more information about creating electronic cam data. Connect the Y-axis pulse output terminals (FP+, FP-, RP+, RP-) to the X-axis MPG input terminals (A0+, A0-, B0+, B0-). In this case, the input signals of the master axis are from output signals of the Y-axis.

【Steps】

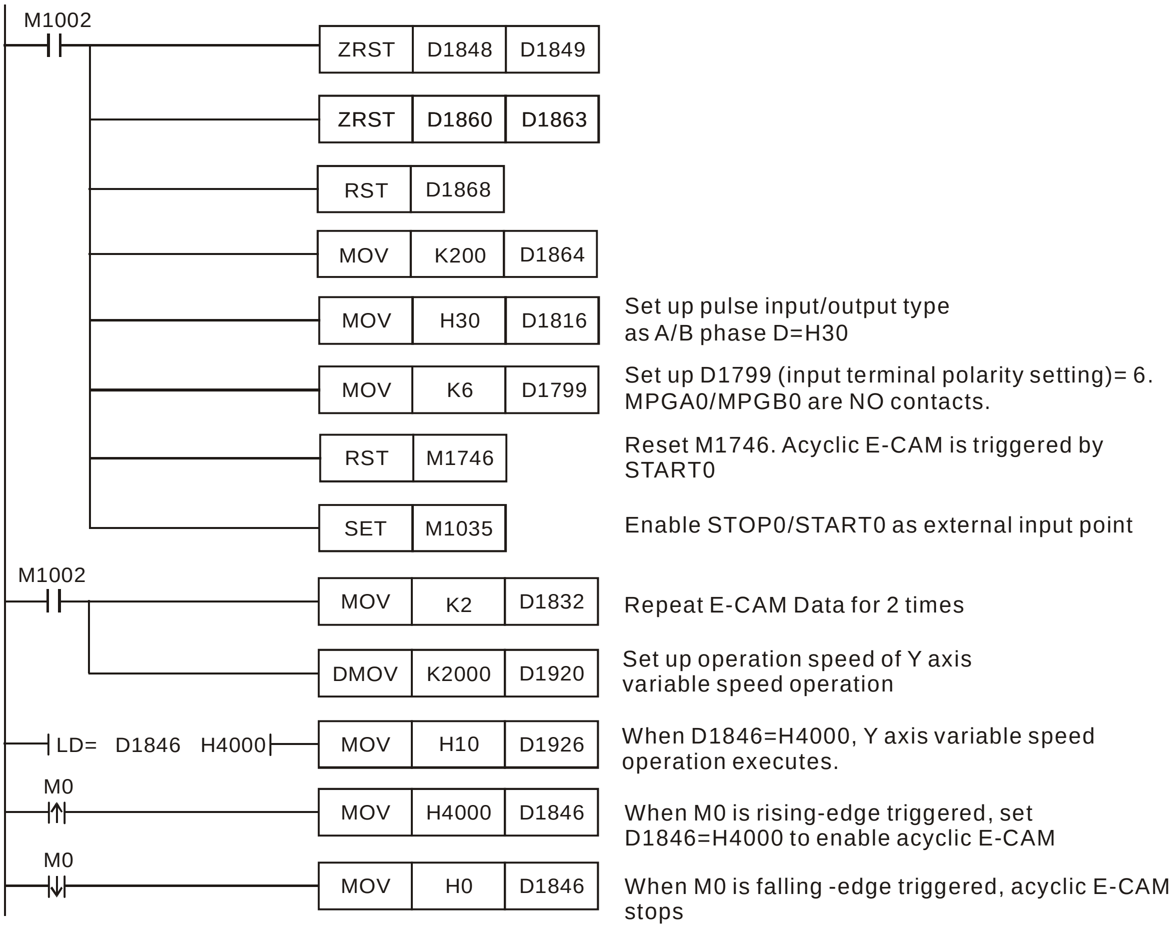
1. Set special registers. The parameters set are as follows.
   (1) Clear the content in registers D1848, D1849, D1862, D1863, D1868.
   (2) Set up input pulse type as A/B phase (D1864=H200).
   (3) Set up pulse output type of Y axis as A/B phase (D1896=H30)
   (4) Set up D1799 (input terminal polarity 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 noncyclic electronic cam.

(7) Set D1832=2 (Repeat electronic 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).

2. Set ON M0 to enable noncyclic electronic cam. When D1846=H4000, Y axis variable speed operation is enabled but X axis is not yet executed.
3. Trigger START0. X axis will be activated and executes the electronic cam data for 3 cycles.
4. Reset M0. When M0 is reset during the execution of electronic cam, X axis will be stopped.

【Program】

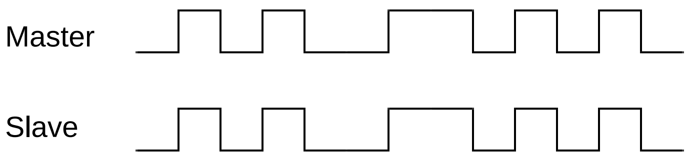


9.3 Registers for Electronic Cam Motion

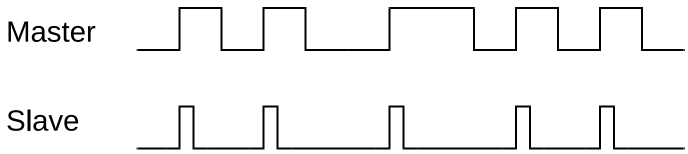
- Electronic cam max frequency

1. D1841, D1840 (VI): Electronic cam max frequency. If the registers are not set up properly, output function will not operate normally.
2. M1749: Electronic cam max frequency control

   M1749=OFF, the Slave frequency will follow the Master frequency, but the max frequency will be 300 kHz.

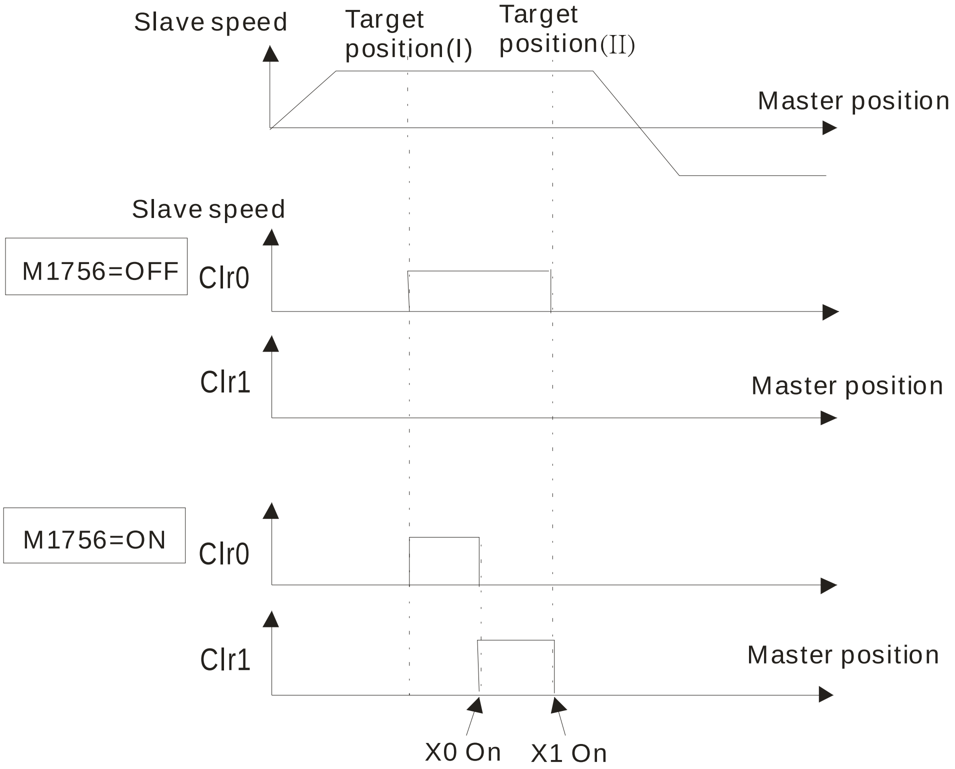


M1749=ON, the max frequency of Slave will be specified by D1841, D1840 (VI).



- Enabling cyclic electronic cam/Completion of cyclic electronic cam
  1. D1846: Enabling cyclic electronic cam. Bit 13=ON, cyclic electronic cam is enabled; bit 13=OFF, cyclic electronic cam is disabled.
  2. M1813: Completion of cyclic electronic cam. M1813 will be ON when cyclic electronic cam is completed. To restart the cyclic electronic cam, the user needs to reset this flag.
- Enabling noncyclic electronic cam

  D1846: bit 14=ON, noncyclic electronic cam enabled; bit 14=OFF, noncyclic electronic cam disabled.
- Lower/Higher bound of electronic cam synchronized output function
  1. Operation Process when M1756=ON: In flying shear/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)≦CP of Master≦D1843, 1842 (PII)”, CLR0 will be ON.



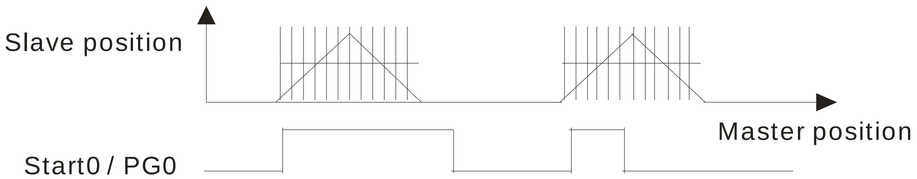
- Completion of electronic cam initialization

  When electronic cam is enabled, DVP-20PM will initialize the settings for electronic cam operation and M1812 will be ON when initialization is completed. The flag needs to be reset by the user.

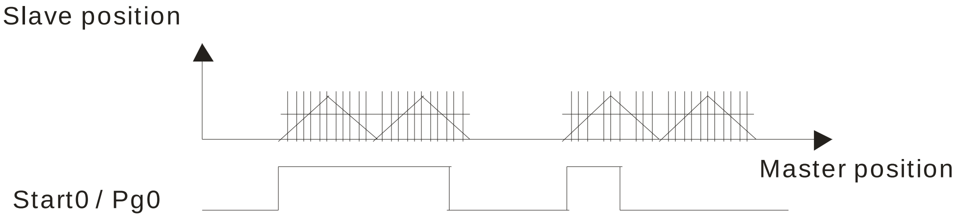
- D1832: Setting the number of times noncyclic electronic cam motion is repeated

  Users can set the number of times noncyclic electronic cam motion is repeated by means of D1832. If the value in D1832 is greater than H8000 (bit 15=1), there will be cyclic electronic cam motion. If the value in D1832 is 0, noncyclic electronic cam motion will not be repeated. If the value in D1832 is 1, noncyclic electronic cam motion will be repeated once.

  D1832=0

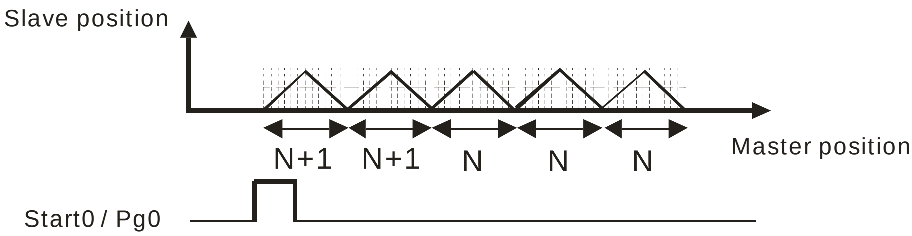


  D1832=1



- D1833: Number of remaining pulses sent by the master axis of an electronic cam

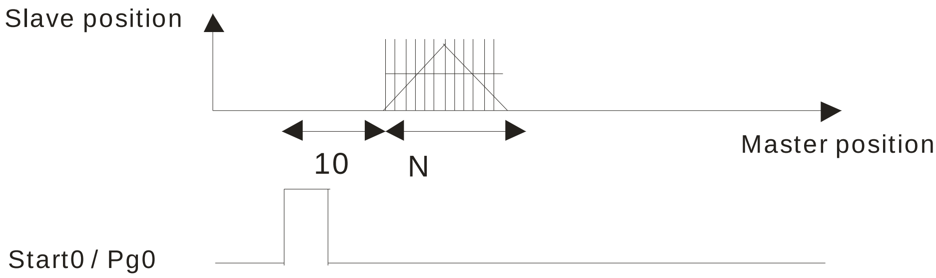
  If the number of pulses sent by the master axis of an electronic cam is not divisible by the number of pulses per ccycle, users can divide the number that is left by setting D1833. If the value in D1832 is 4, and the value in D1833 is 2, one pulse will be added to the first cycle and the second cycle. (Note: The value in D1833 can not be greater than the value in D1832.)



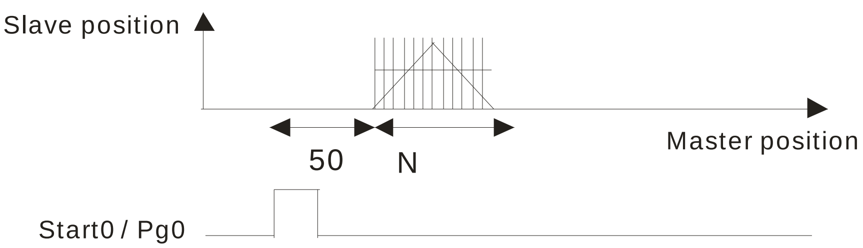
- D1834: Number of pulses sent by the master axis of the electronic cam before the electronic cam is started

  Users can delay the sending of pulses by the slave axis of a noncyclic electronic cam by setting D1834. When the number of pulses sent by the master of a noncyclic electronic cam sends is equal to the value in D1834, the slave axis of the noncyclic electronic cam sends pulses.

  D1834=10



  D1834=50



- Selecting the start signal of noncyclic electronic cam

  1. When M1746 is OFF, the start signal of noncyclic electronic cam will be START0; when M1746 is ON, PG0 will be the start signal of noncyclic electronic 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 noncyclic electronic cam rather than PG0.

- Indicating electronic cam completion

  When M1748=ON, electronic cam stops immediately when electronic cam data is completed.

- Position and frequency of Master and Slave

  Number of accumulated MPG input pulses/Master position: DD1862

  MPG input frequency/Master frequency: DD1860

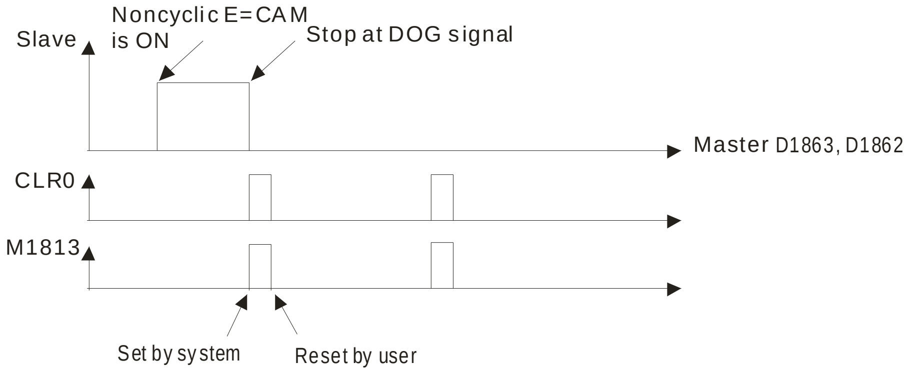
  Current position/Slave position: DD1848

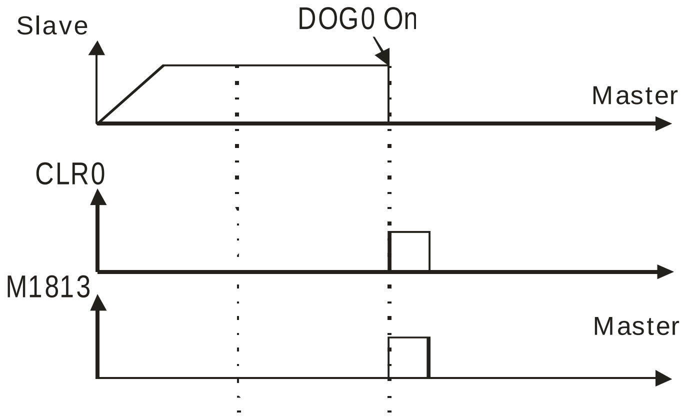
- Selecting electronic cam chart

1. Every electronic cam chart includes a set of electronic cam data. There are 3 electronic cam electronic cam charts available for 3 sets of electronic cam data.
2. Before electronic cam executes, set up D1868 to select the electronic cam chart 0~2 to be used.
3. electronic cam chart-0: D1868=0; electronic cam chart-1: D1868=H’8001; electronic cam chart-2: D1868=H’8002

- Noncyclic electronic cam immediate stop function (Stop signal: DOG)

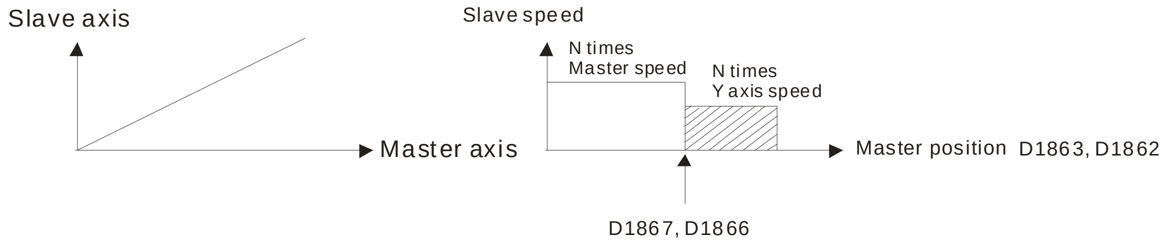
  Noncyclic electronic cam immediate stop function can be used together with M1755 for performing cutting with equal deceleration speed and correct stop position at DOG. When M1753 is ON, noncyclic electronic cam stops at DOG signal and CLR0 will be ON to clear the error of servo. Users have to reset M1813, so that CLR0 will be reset for next cutting operation.





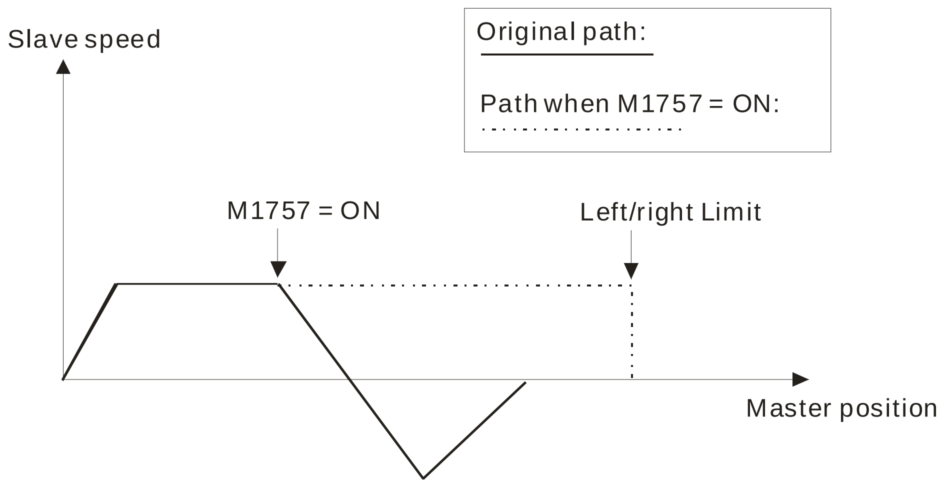
- Switching the source of Master of noncyclic electronic cam

  For noncyclic electronic 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 noncyclic electronic cam cycle is completed. (Not applicable for virtual Master mode: M1909/M1910). The function is generally used for obtaining equal deceleration speed as well as correct stop position, regardless of the error resulted from Master speed or the characteristics of photo sensor.



- Remaining current speed until meeting limit switch

  When M1757 is ON, Slave will remain current speed and continue operation until the right/left limit is reached.



9.4 Creating Electronic Cam Data

Electronic cam data defines the relation between the positions of a master axis and the positions of a slave axis.

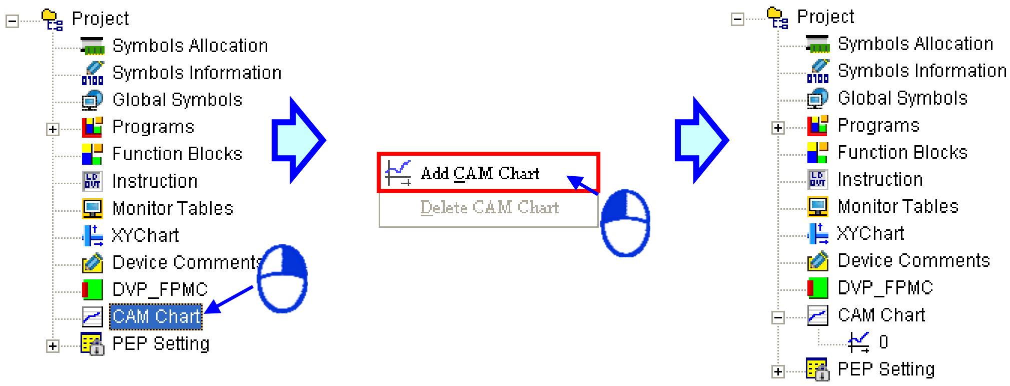
1. Before users create a cam chart in PMSoft, they have to know the relation between the positions of a master axis and the position a slave axis. There are two methods of getting the relation between the positions of a master axis and the positions of a slave axis.
   - Method 1: Function that relates the positions of a master axis to the positions of a slave axis
   - Method 2: Measuring the relation between the positions of a master axis and the positions of a slave axis at work

After electronic cam data defines the relation between the positions of a master axis and the positions of a slave axis, users can get the positions of the slave axis by means of the positions of the master axis.

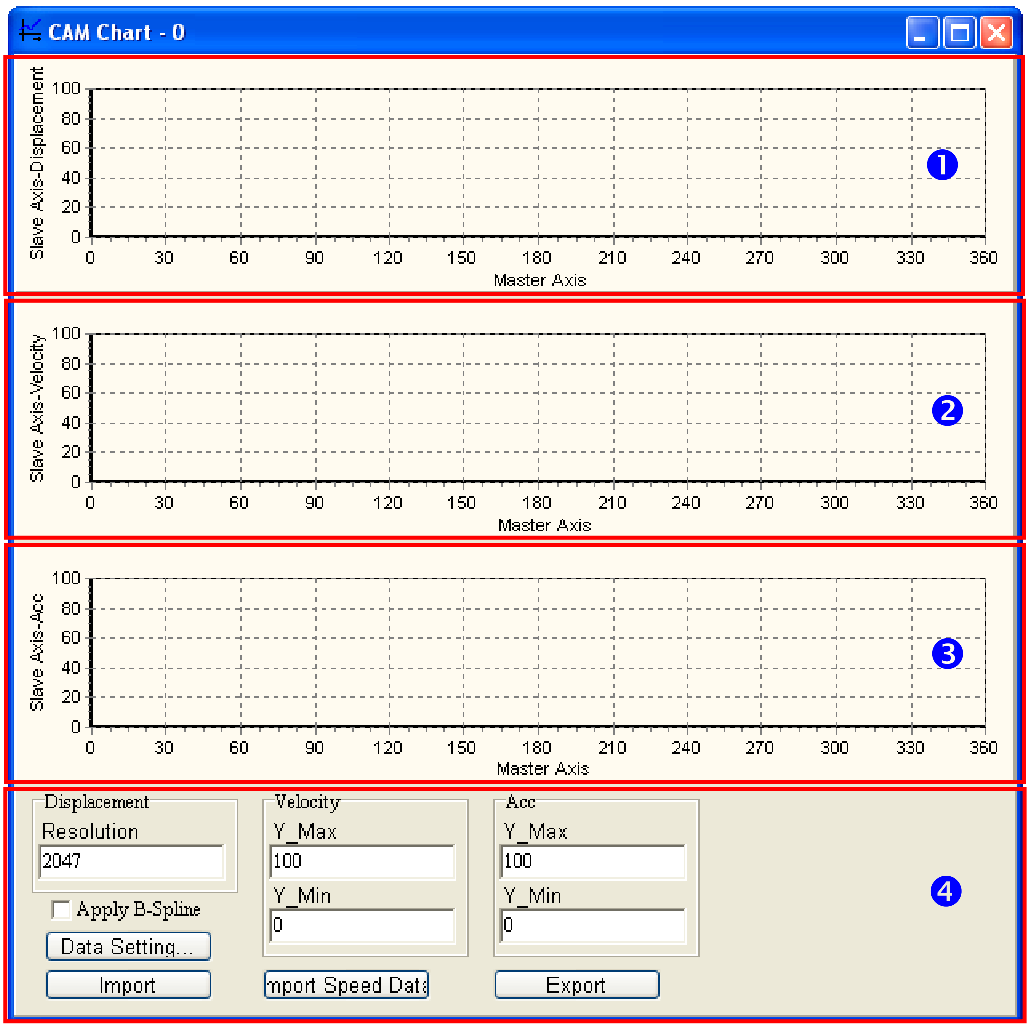
9.4.1 Creating a Cam Chart in PMSoft

9.4.1.1 Function Relating the Position of a Master Axis to the Position of a Slave Axis

After users create a project in PMSoft, right-click **CAM Chart** in the system information area, and click **Add CAM Chart** on the context menu, the **CAM Chart-0** window will appear.



The **CAM Chart-0** window is shown below.



❶ Displacement: The relation between the master axis and the slave axis is described in terms of displacement.

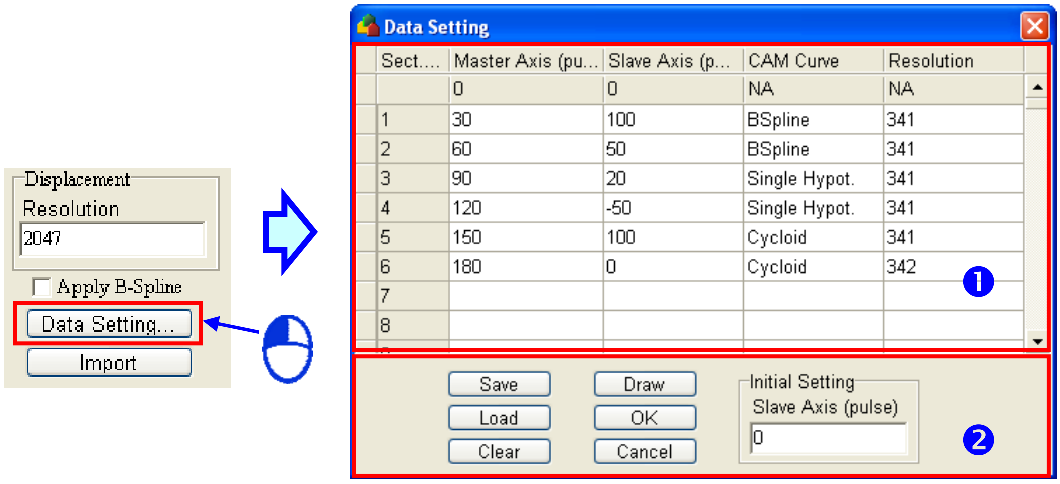
❷ Velocity: The relation between the master axis and the slave axis is described in terms of speed.

❸ Acceleration: The relation between the master axis and the slave axis is described in terms of acceleration.

❹ Data setting area:

- **Resolution**: Users can set the number of data points required in the electronic cam chart. The number of data points must be in the range of 10 to 2047.
- **Velocity**: The maximum speed of the slave axis and the minimum speed of the slave axis are shown in this section. They are calculated by the system according to the data related to displacement. Users can change the maximum speed of the slave axis and the minimum speed of the slave axis by themselves.
- **Acc**: The maximum acceleration of the slave axis and the minimum acceleration of the slave axis are shown in this section. They are calculated by the system according to the data related to displacement. Users can change the maximum acceleration of the slave axis and the minimum acceleration of the slave axis by themselves.
- **Data Setting…**: The description of the relation between the master axis and the slave axis in terms of displacement is shown in the **Data Setting** window. The displacement resolution set in the data setting area will be brought into the **Data Setting** window after the **Data Setting** window is opened. If users click the **Apply B-spline** checkbox in the data setting area, **B-spline** will be automatically selected in the **Data Setting** window.
- **Import**: Importing the description of the relation between the master axis and the slave axis in terms of displacement
- **Export**: Exporting the description of the relation between the master axis and the slave axis in terms of displacement
- **Import Speed Data**: Importing the description of the relation between the master axis and the slave axis in terms of speed

After the users click **Data Setting…** in the **CAM Chart-0** window, the **Data Setting** window will appear. The **Data Setting** window is composed of sections. The users can set a section of a cam curve in every section. A complete cam curve is composed of several sections. The users can set 360 sections at most. An electronic cam cycle is composed of the sections created by the users.



❶ Users can define the relation between the master axis and the slave axis in every section.

- **Master Axis**: Users can set the displacement of the master axis. A pulse is a unit of the measurement for displacement. The values that the users type in the **Master Axis (pulse)** column must be greater than 0, and must be in numerical order.
- **Slave Axis**: Users can set the displacement of the master axis. A pulse is a unit of the measurement for displacement. The values that the users type in the **Slave Axis (pulse)** column can be positive values or negative values.
- **CAM Curve**: The functions which can be selected are **Const Speed**, **Const Acc.**, **Single Hypot.**, **Cycloid**, and **B-Spline**. If users click the **Apply B-spline** checkbox in the **CAM Chart-0** window, **B-spline** will be automatically selected in the **Data Setting** window.
- **Resolution**: Users can set the number of data points used in a section. The number of data points must be in the range of 10 to 2047. If the users do not set resolutions for

sections, the number of data points left will be equally distributed to the sections. The users have to set resolutions according to equipment's requirements. The higher the resolutions set are, the more smoothly the equipment used operates. Besides, the size of the electronic cam data gotten is big if the resolutions set are high.

❷ After sections of a cam curve are created, users can click **Save**, **Load**, **Clear**, **Draw**, **OK**, **Cancel**, or set the initial position of the slave axis.

- **Save**: Saving the data set in sections
- **Load**: Loading the data which was saved
- **Clear**: Clearing all the data in sections
- **Draw**: Compiling the data set in sections, and drawing the electronic cam data gotten on the electronic cam chart created
- **OK**: Compiling the data set in sections, drawing the electronic cam data gotten on the electronic cam chart created, and closing the **Data Setting** window
- **Cancel**: Closing the **Data Setting** window.
- **Initial Setting**: Setting the initial position of the slave axis

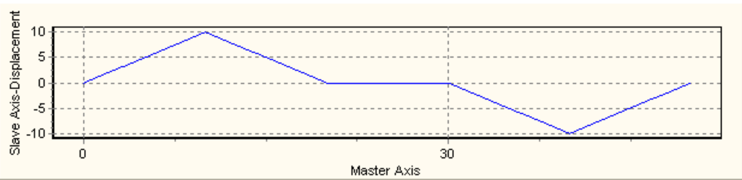
After the users click "Export" in the **CAM Chart-0** window, the displacement chart, the velocity chart, and the acceleration chart in the **CAM Chart-0** window will be saved in the CAMData folder in the folder in which PMSoft is installed. There are three files in the CAMData folder.

- <Folder in which PMSoft is installed>\CAMData\Data_S.txt: Displacement
- <Folder in which PMSoft is installed>\CAMData\Data_V.txt: Velocity
- <Folder in which PMSoft is installed>\CAMData\Data_A.txt: Acceleration

Data_S.txt, Data_V.txt, and Data_A.txt are shown below.

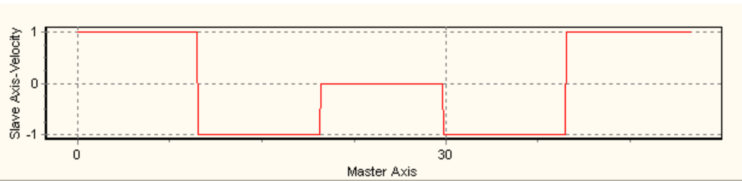
Displacement chart

Path: <Folder in which PMSoft is installed>\CAMData\Data_S.txt



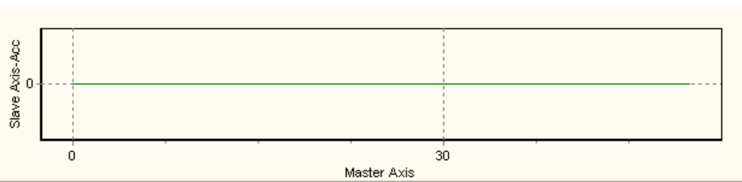
Velocity chart

Path: <Folder in which PMSoft is installed>\CAMData\Data_V.txt



Acceleration chart

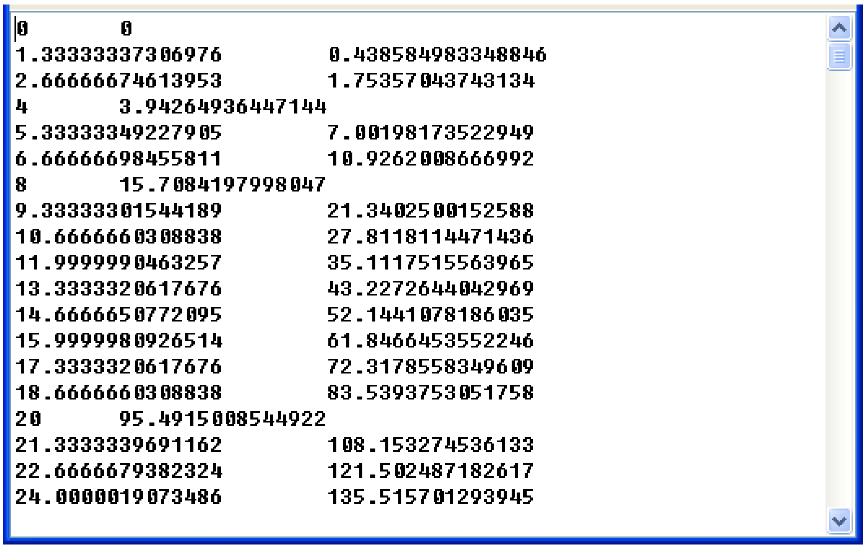
Path: <Folder in which PMSoft is installed>\CAMData\Data_A.txt



9.4.1.2 Measuring the Relation between the Position of a Master Axis and the Position of a Slave Axis at Work

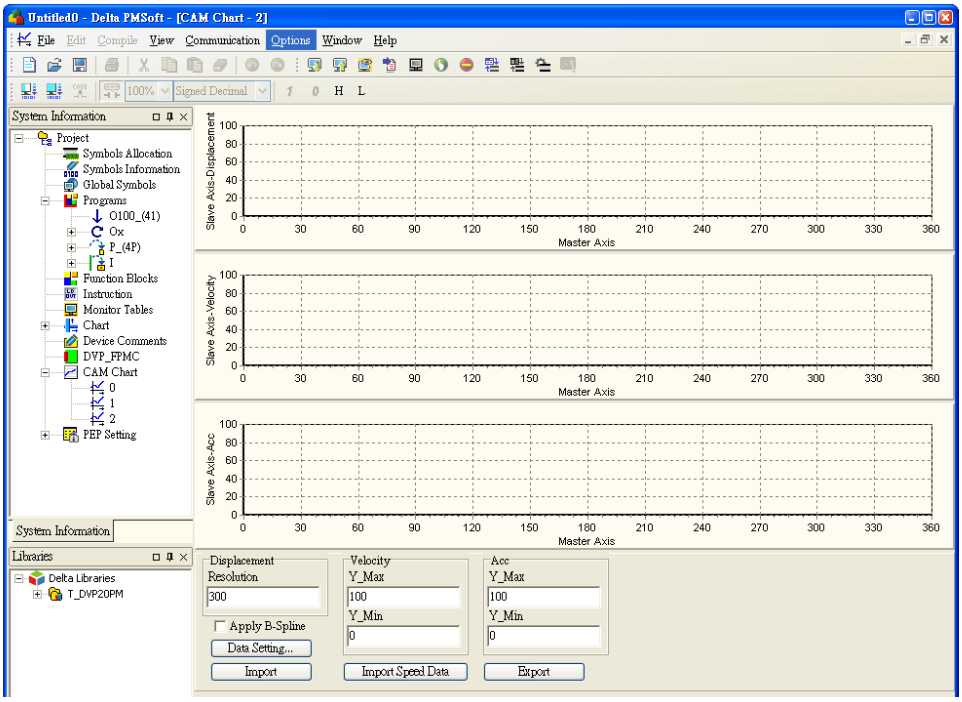
Users can store the relation between the positions of the master axis specified and the positions of the slave axis specified in a file in the CAMData folder, and then import the relation into a cam chart in PMSoft. The steps of importing the relation between the positions of the master axis specified and the positions of the slave axis specified into a can chart in PMSoft are as follows.

1. Store data about displacement in Data_S.txt in the folder in the CAMData folder.

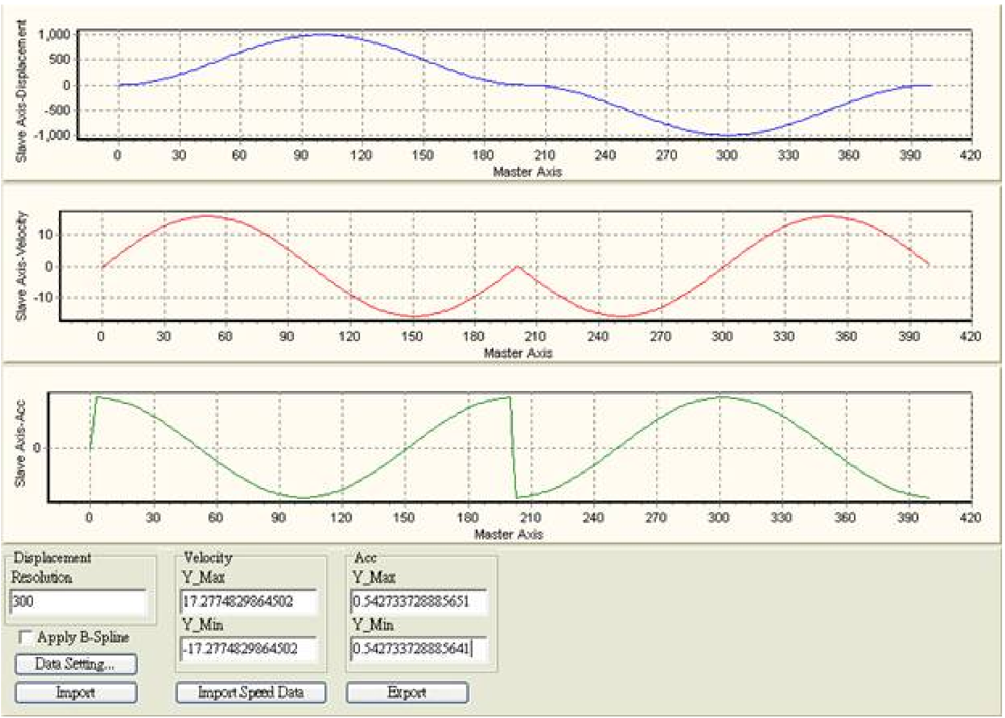


0	0
1.33333337306976	0.438584983348846
2.66666674613953	1.75357043743134
4	3.94264936447144
5.33333349227905	7.00198173522949
6.66666698455811	10.9262008666992
8	15.7084197998047
9.33333301544189	21.3402500152588
10.6666660308838	27.8118114471436
11.9999990463257	35.1117515563965
13.3333320617676	43.2272644042969
14.6666650772095	52.1441078186035
15.9999980926514	61.8466453552246
17.3333320617676	72.3178558349609
18.6666660308838	83.5393753051758
20	95.4915008544922
21.3333339691162	108.153274536133
22.6666679382324	121.502487182617
24.0000019073486	135.515701293945

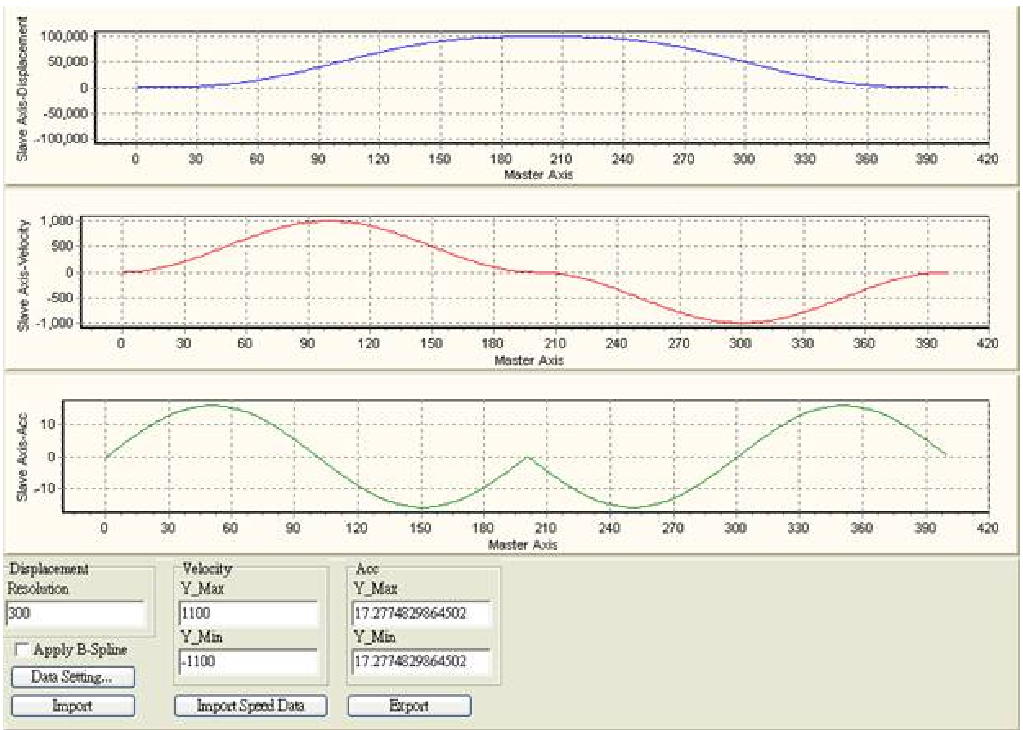
2. Open a **CAM Chart** window in PMSoft, and then type a value in the **Resolution** box.



3. After **Import** is clicked, PMSoft will read Data_S.txt and draw a displacement chart in the **CAM Chart** window, and a velocity chart and an acceleration chart will be drawn in accordance with the contents of Data_S.txt.



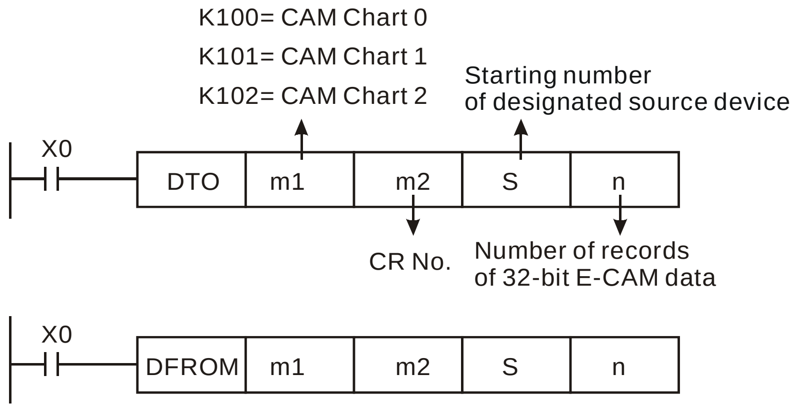
4. After **Import Speed Data** is clicked, PMSoft will read Data_S.txt and draw a velocity chart in the **CAM Chart** window, and a displacement chart and an acceleration chart will be drawn in accordance with the contents of Data_S.txt



9.4.2 Using DTO / DFROM to Create Electronic Cam Data

DVP-20PM is designed with 3 virtual modules exclusively for electronic cam 0~electronic cam 2, and the numbers of each electronic cam chart are K100, K101 and K102. Through DTO/DFROM instructions, users can set up or modify the electronic cam data in the user program. Electronic cam Data created by PMSoft has been converted into floating point format because electronic 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 electronic cam data in Monitor Table.

The format of DFROM/DTO instruction for setting up electronic cam data is as below. In electronic cam application, electronic cam 0~electronic cam 2 is regarded as #100~#102 special modules, i.e. for accessing electronic cam 0, operand $\mathbf{m_1}$ has to be set as K100. Operand $\mathbf{m_2}$ 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 electronic cam

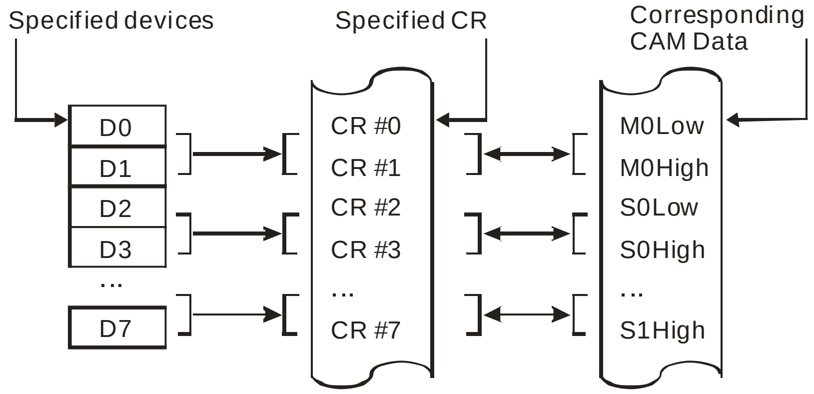
CR#	Function	Data type	Data length
0~9999	Creating/Modifying single electronic cam data	DWord	1
10000	Creating rotary cut electronic cam data	DWord	10
10001	Reading electronic cam status	DWord	7
10002	Electronic cam chart real-time change	DWord	1
10003	Smooth curve interpolation	DWord	3
10005	Reading electronic cam status in capture mode	DWord	5

■ Explanations of control registers

CR#0~9999: Creating/Modifying single electronic cam data

[Description]

DVP-20PM is built in with 3 sets of electronic cam charts: Electronic cam chart 0~electronic cam chart 2. Every electronic cam chart supports electronic cam data of 10,000 points and is set up by CR#0~9999. One set of electronic cam data consists of 2 points including Master and Slave position, and the unit of every point is Dword. For example, 2 sets of electronic 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.



M: Master position
S: Slave position
32-bit instruction, n=4 indicates 2 sets of CAM Data

CR#10000: Creating rotary cut electronic cam data

[Description]

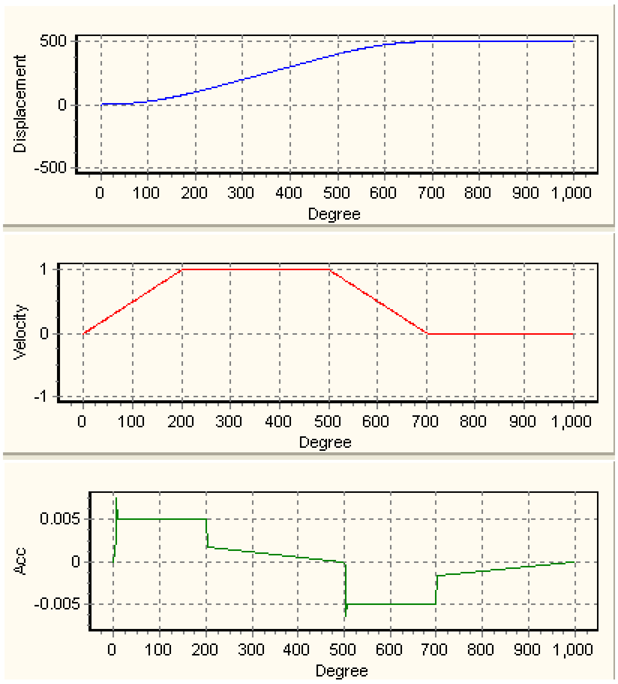
DVP-20PM provides CR#10000 exclusively for creating rotary cut electronic cam. The user can create and apply the rotary-cut electronic cam by setting up the related machinery parameters according to actual needs. Parameters for rotary-cut electronic cam are as below.

Parameter	Data	Format	Explanation
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 synchonized in sync area. Calculation method of magnification ratio: $V_1 = V_2 \Rightarrow \frac{F_1 \times 3.14 \times D_1}{R_1} = \frac{F_2 \times 3.14 \times D_2}{R_2} \Rightarrow \frac{F_2}{F_1} = \frac{R_2/D_2}{R_1/D_1}$ 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 curve (Low word)	Integer	Electronic cam acceleration setting: 0: Uniform curve 1: Uniform acceleration curve 2: SingleHypot curve 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 2: midCAMbegin 3: midCAMend (If the curve is selected, electronic cam stops automatically when completed.) 5: rightCAM: Sync area is on the right side of cam curve. Note: If 2 or 3 is selected, the length of the master axis will be 1/2 times the length of the slave axis.
			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: Changing data dynamically is not allowed. b[15]=1: Borrow the data setting of previous electronic cam.
P7	Result	Integer	Display the rotary cut electronic cam creating results (when read back). 0: OK 1: Input data is not proper.

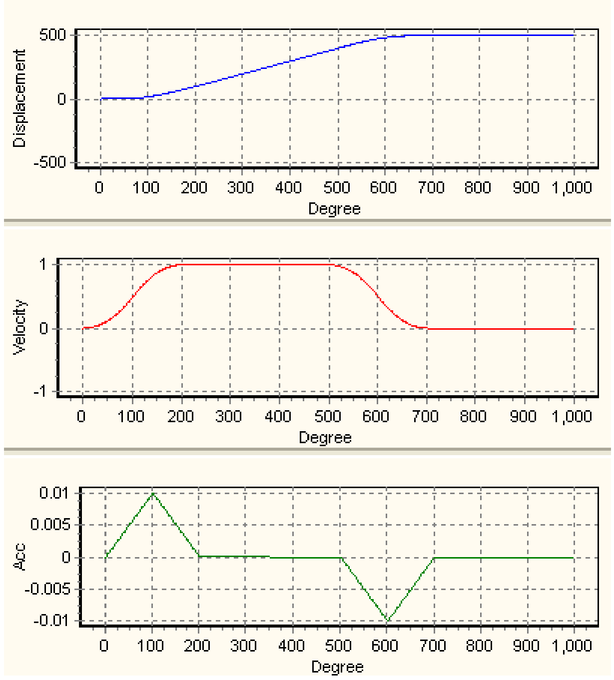
P6 sets up the rotary cut acceleration curves and the electronic cam curves, and the examples of each curve are as below.

Acceleration curves

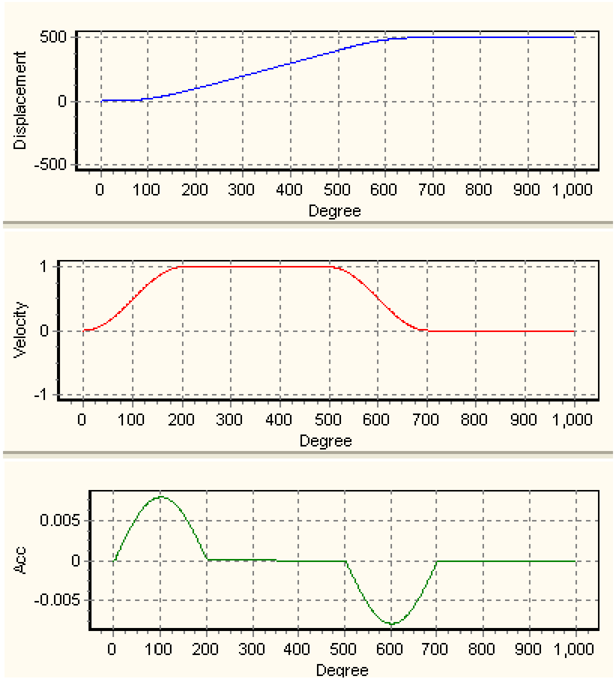
0: Uniform curve



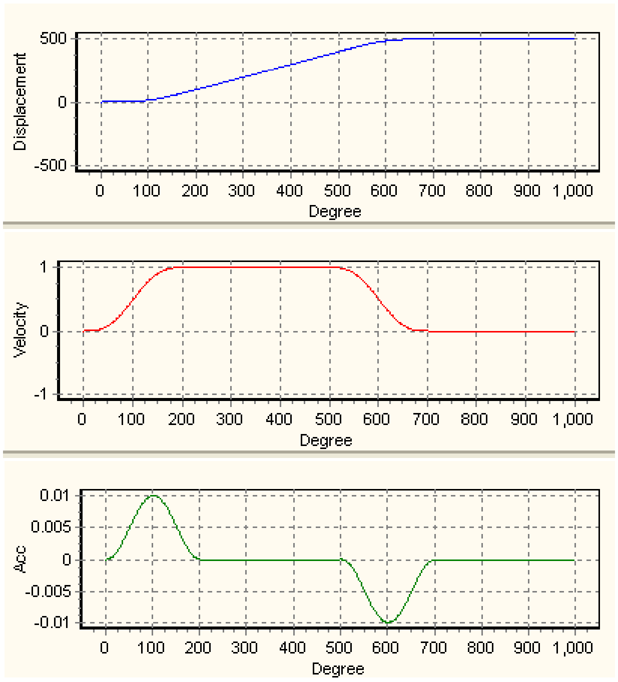
1: Uniform acceleration curve



2: SingleHypot curve

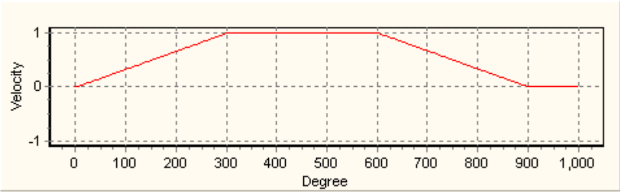


3: Cycloid

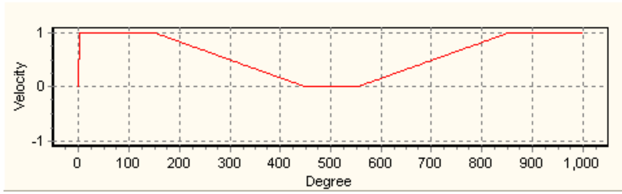


Cam curves

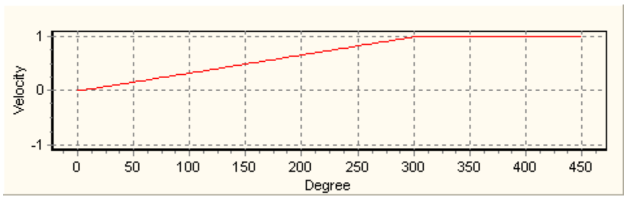
0: leftCAM



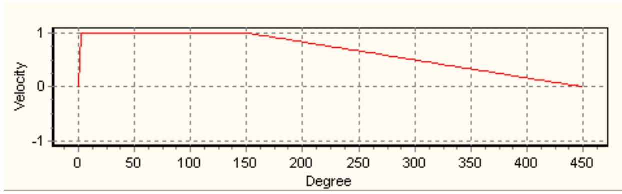
1: midCAMall



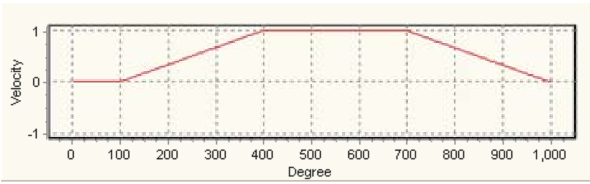
2: midCAMbegin



3: midCAMend



5: rightCAM



For detailed explanation of creating rotary cut electronic cam curves, please refer to 9.4.2.3.

The rotary cut electronic cam data can be used for general rotary cut applications. However, for thick material cutting application, the cutting angle and the cutter speed should be additionally considered on setting electronic cam. DVP-20PM provides parameters for thick material cutting application as below.

Parameter	Data	Format	Explanation
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 electronic 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 degrees
P7	Sync end angle of Slave	Integer	Set range: 180~360 degrees
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.

Parameter	Data	Format	Explanation
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 electronic cam curves, please refer to 9.4.2.4.

CR#10001: Reading electronic cam status

[Description]

The function is used to read current electronic cam status when electronic cam is executing. Only DFROM instruction is applicable for this function. The read data will be as shown below.

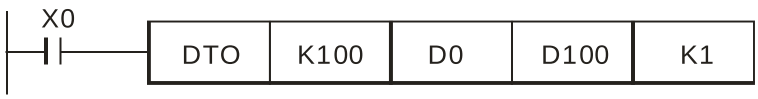
Parameter	Data	Format	Explanation
P1	Current position of Master	Integer	Reset at DOG signal
P2	Length of Master 1	Integer	DOG signal to DOG signal Single cam: Length of the master axis after the triggering of DOG
P3	Current electronic cam curve	Integer	
P4	Current position of Slave	Integer	Updated at DOG signal Single cam: Present length of the master axis after the triggering of DOG
P5	Error of Slave	Integer	Error between DOG signal and the end point of CAM cycle.
P6	Current position of Master	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 electronic 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.

CR#10002: Electronic cam chart real-time change

[Description]

There are 3 built-in electronic cam charts in PMSoft: Electronic cam chart-0~electronic cam chart-2. If users need to conduct real-time change of electronic cam chart during program execution, set up this register to designate the electronic cam chart to be enabled in the next cycle. Care should be taken on the number of electronic 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 electronic cam chart real time change. When X0=ON, set value (0, 1, 2) in D100 indicating the cam chart to be selected (electronic cam chart-0~electronic cam chart-2) will be written into DVP-20PM.



CR#10003: Smooth curve interpolation

[Description]

When the original data points are not sufficient for a smoother operation, users 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.

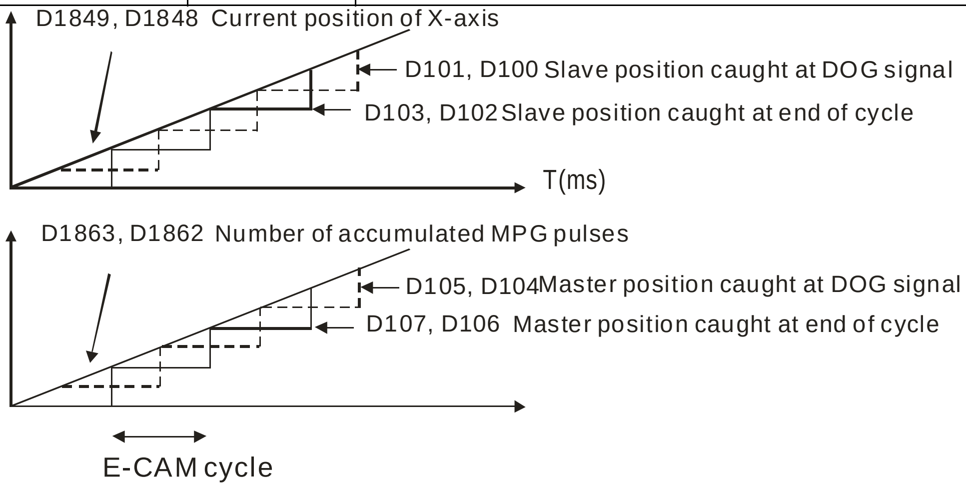
Parameter	Data	Format	Explanation
P1	Maximum length of Master	Integer	
P2	Original data points	Integer	
	Total points to be inserted	Integer	

Parameter	Data	Format	Explanation
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 type	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: Reading electronic cam status in capture mode

[Description]

Parameter	Data	Format	Explanation
P1	Slave position	Integer	Caught at DOG signal
P2	Slave position	Integer	Caught at end of cycle
P3	Master position	Integer	Caught at DOG signal
P4	Master position	Floating point	Caught at end of cycle
P5	Master position	Floating point	Caught at START0 (M1746=OFF) or PG0 (M1746=ON)



The function of reading electronic cam status in capture mode should be used with M1757 (Remain current speed), and can be applied for checking the cutter position in flying shear applications. For details please refer to 9.4.2.6.

9.4.2.1 Creating / Modifying Electronic Cam Data

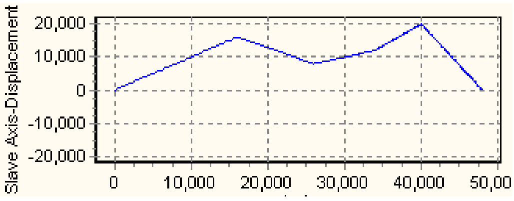
After users create electronic cam data in a cam chart in PMSoft, the cam data will be downloaded to a DVP-20PM series motion controller. If the users modify the electronic cam data in PMSoft, they have to download the new electronic cam data created to the DVP-20PM series motion controller again after they modify the electronic cam data. A DVP-20PM series motion controller allow users to modify electronic cam data. The users can modify electronic cam data when the program in the DVP-20PM series motion controller is executed.

Data length of electronic cam data can be real-time modified. When 4 words of one point (set) of electronic cam data are all 0 and identified, DVP-20PM will take it as electronic cam completion. Therefore, if users want to modify 10 points electronic cam data into 5 points, DTO instruction can be applied to write 0 into the 4 words of the sixth point.

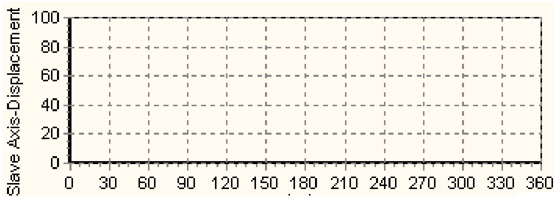
■ Example

【Function】

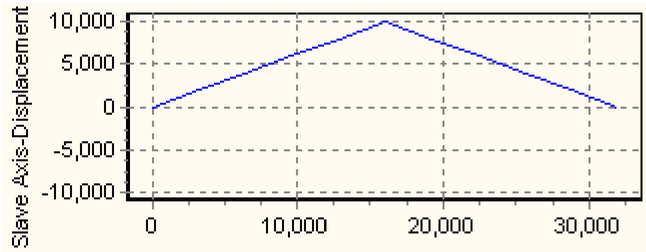
Users can modify a particular point in a cam chart. In figure (a), there is an original cam chart. There may be data or no data in the cam chart which will be modified. In the figure (b), the three pairs of coordinates (16000, 10000), (32000, 0), and (0, 0) are written.



or



(a) Original cam chart

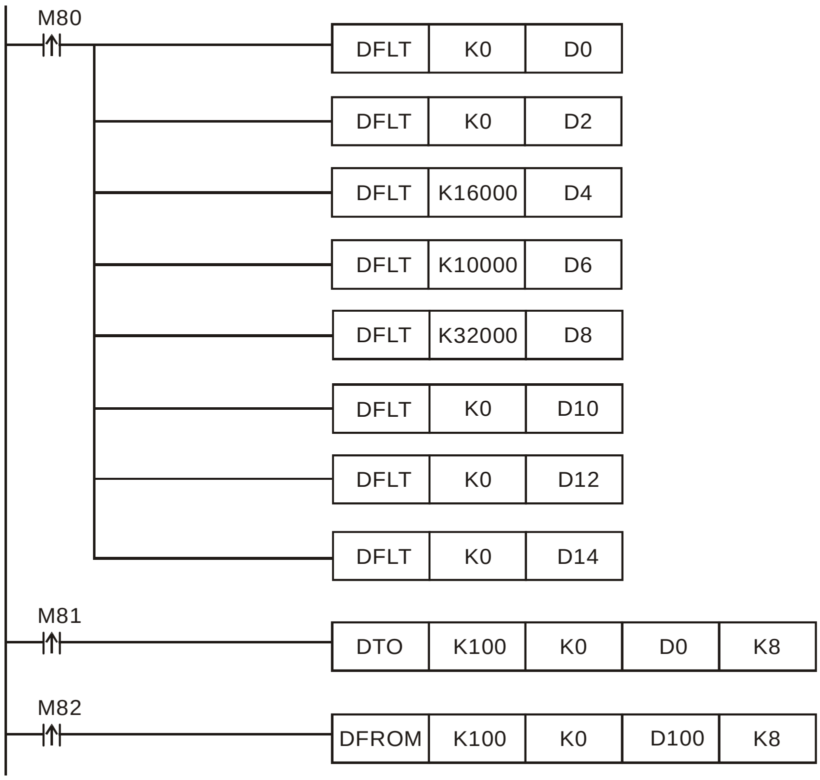


(b) Three pairs of coordinates in a cam chart

【Steps】

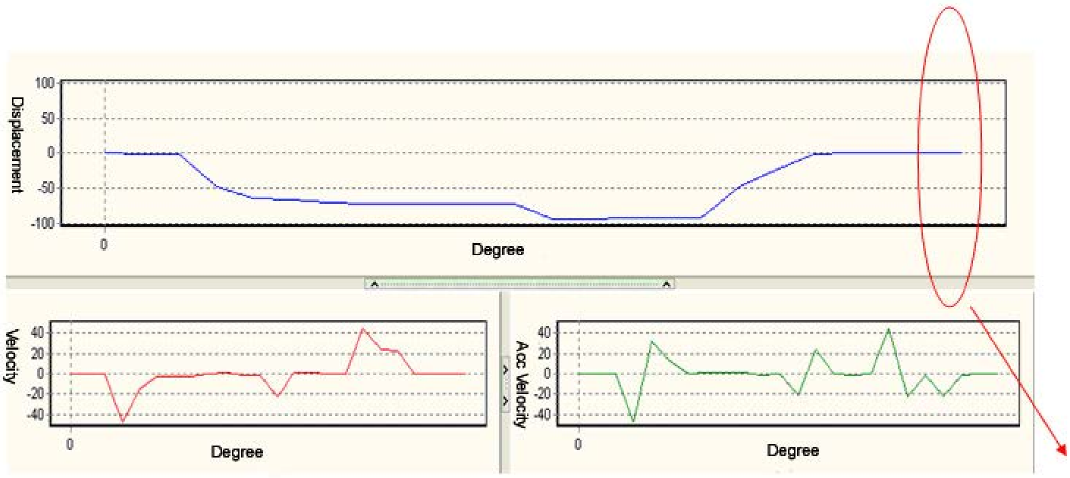
1. Set M80 to convert the 4-point electronic cam data (3-point data with an additional point 0) into binary floating point format, and store the results in D0~D15.
2. Set M81 to write the data in D0~D15 into electronic cam chart-0 (K100).
3. Set M82 to read the written data back into D100~D115 so as to confirm the data is correctly written into electronic cam chart-0.
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.

【Program】



9.4.2.2 Creating a Smooth Curve

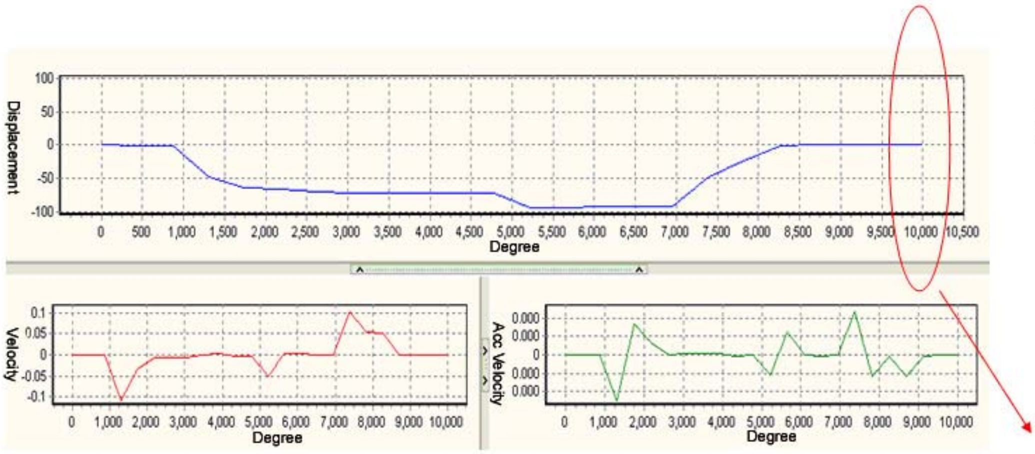
Smooth curve interpolation conducts interpolation and magnification on data points of Master, so as to smoothen the modified curve with higher resolution. The interpolation results will elevate the stability and smoothness for machine operation. In the sample below, the original data points are 24 points, 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 electronic cam curve will be drawn according to the calculated results in the same proportion of the previous electronic cam curve.



Original data points: 24

Master	Slave
0	0
1	-0.036
2	-0.288
3	-47.196
4	-62.604
5	-65.16
6	-68.332
7	-70.44
8	-70.98
9	-70.052
10	-70.756
11	-71.496
12	-93.672
13	-92.556
14	-91.224
15	-91.152
16	-91.044
17	-46.08
18	-22.86
19	-0.396
20	0.288
21	0.072
22	0
23	0.036

After the interpolation, the original points of Master 0~23 are modified as 10,000 points as below.

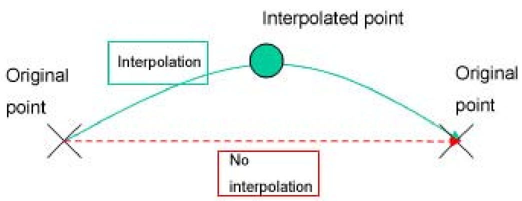


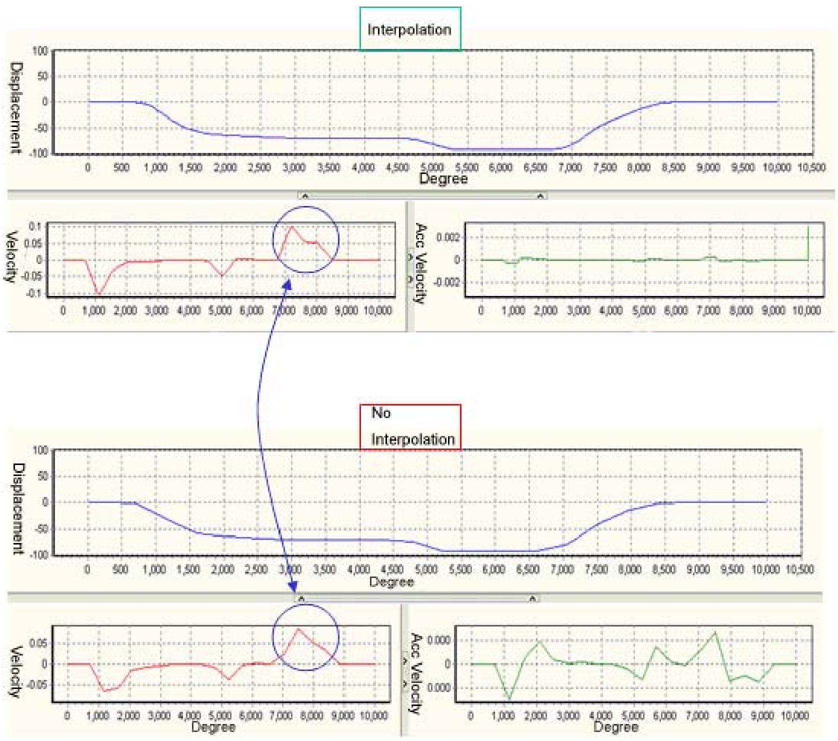
Original data points: 24

Modified data points: 10000

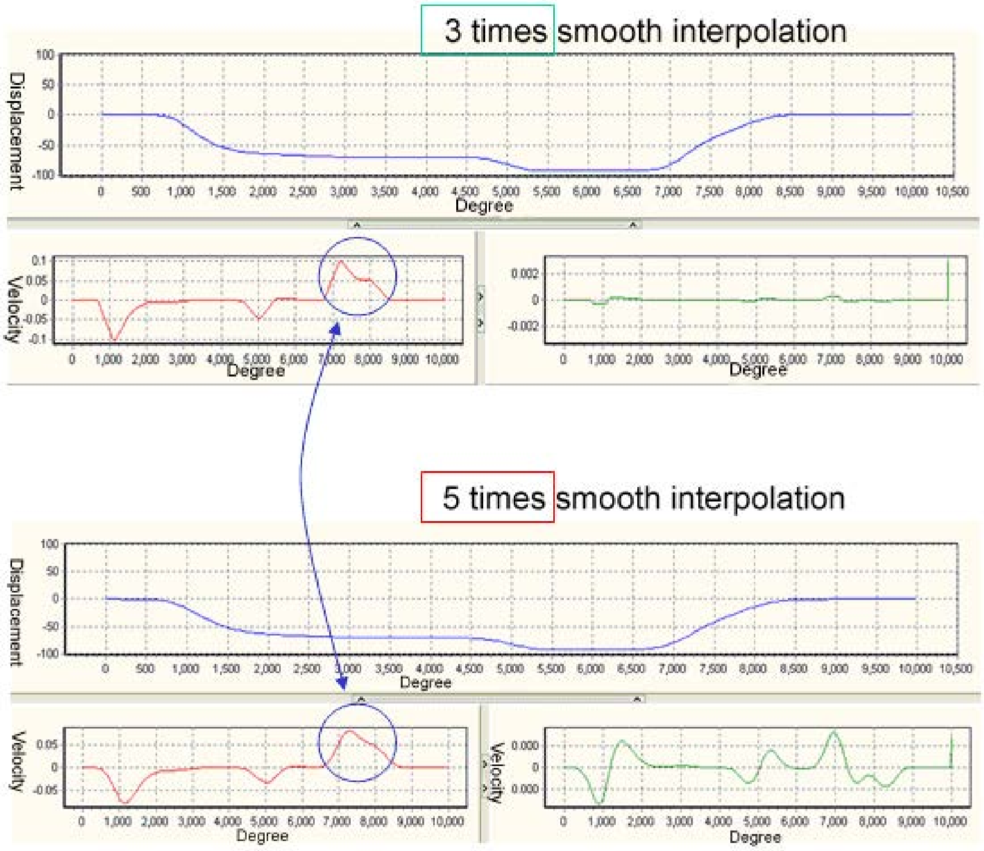
Master	Slave
0	0
434	-0.036
869	-0.288
1304	-47.196
1739	-62.604
2173	-65.16
2608	-68.332
3043	-70.44
3478	-70.98
3913	-70.052
4347	-70.756
4782	-71.496
5217	-93.672
5652	-92.556
6086	-91.224
6521	-91.152
6956	-91.044
7391	-46.08
7826	-22.86
8260	-0.396
8695	0.288
9130	0.072
9565	0
10000	0.036

The below diagram illustrates the difference between the original curve and the curve with interpolation.





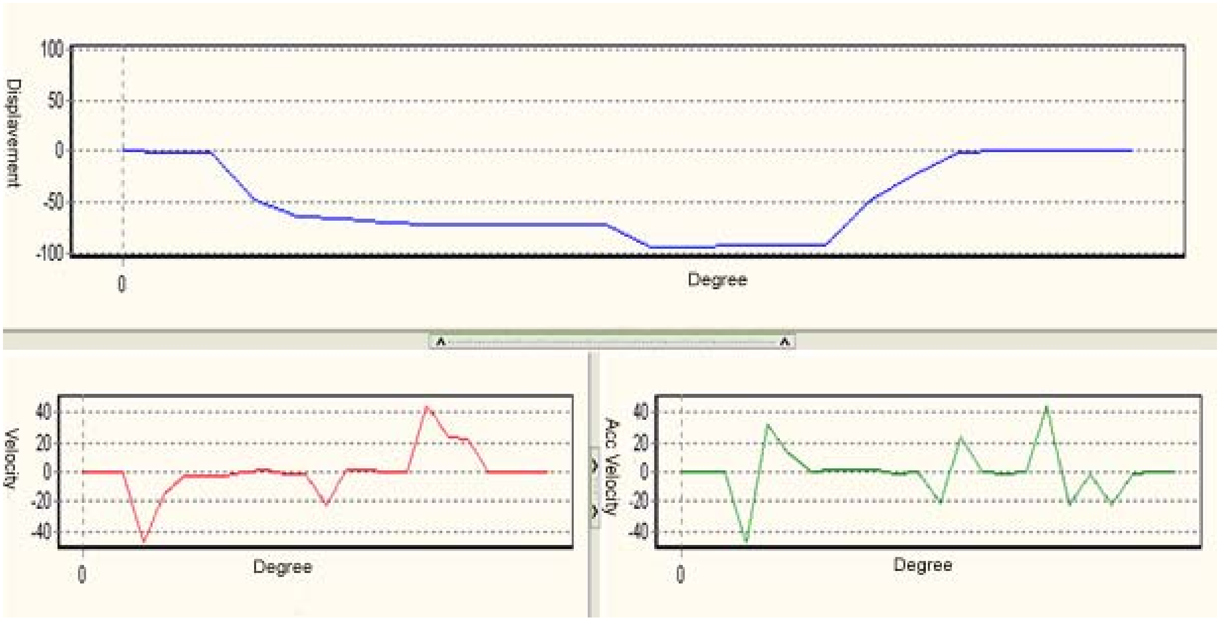
With higher smoothing coefficient, the generated curve will be smoother as well. The below diagrams show the results of 3-times and 5-times interpolation respectively.



- Example

【Function】

Conducting smooth curve interpolation on the original curve of electronic 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.

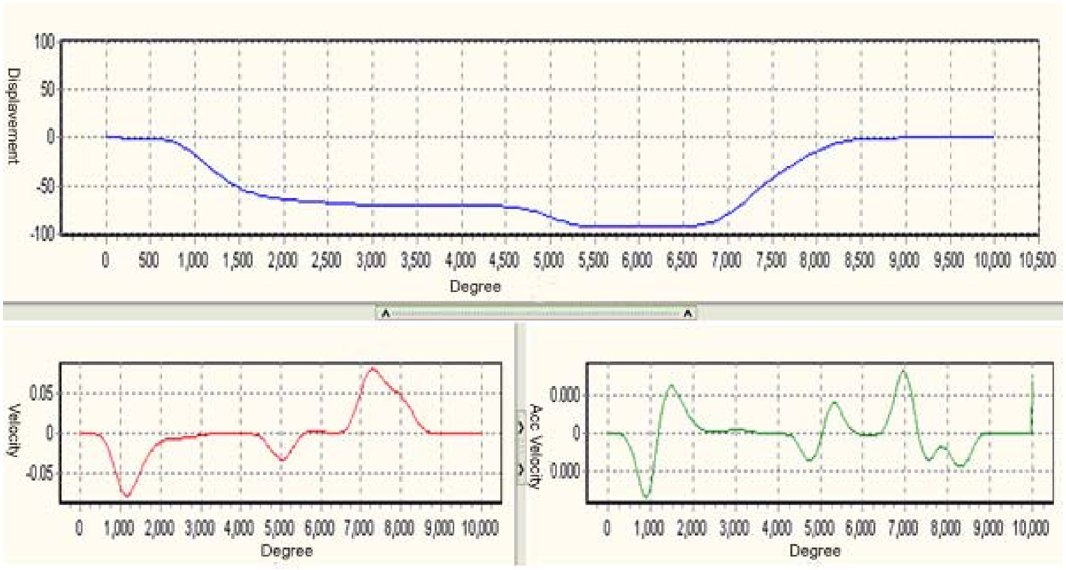


【Steps】

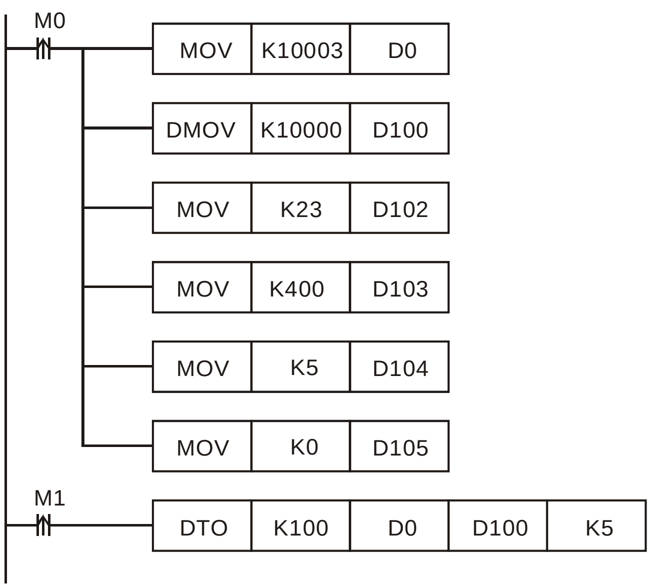
1. Create a 23-point electronic cam curve in electronic cam chart-0. Use the import function to set up the electronic cam data as below. For the set up methods please refer to section 9.4.

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

2. Set the resolution of electronic cam chart-0. The resolution should be equal or bigger than "Original data points+Total points to be inserted".
3. Set M0 to write the parameters for smooth curve interpolation into registers D100~D105.
4. Set M1 to write in the data in D100~D105 for executing smooth curve interpolation.
5. Execute the program and monitor the electronic cam curve in electronic cam chart-0. The curves after interpolation will be as below.



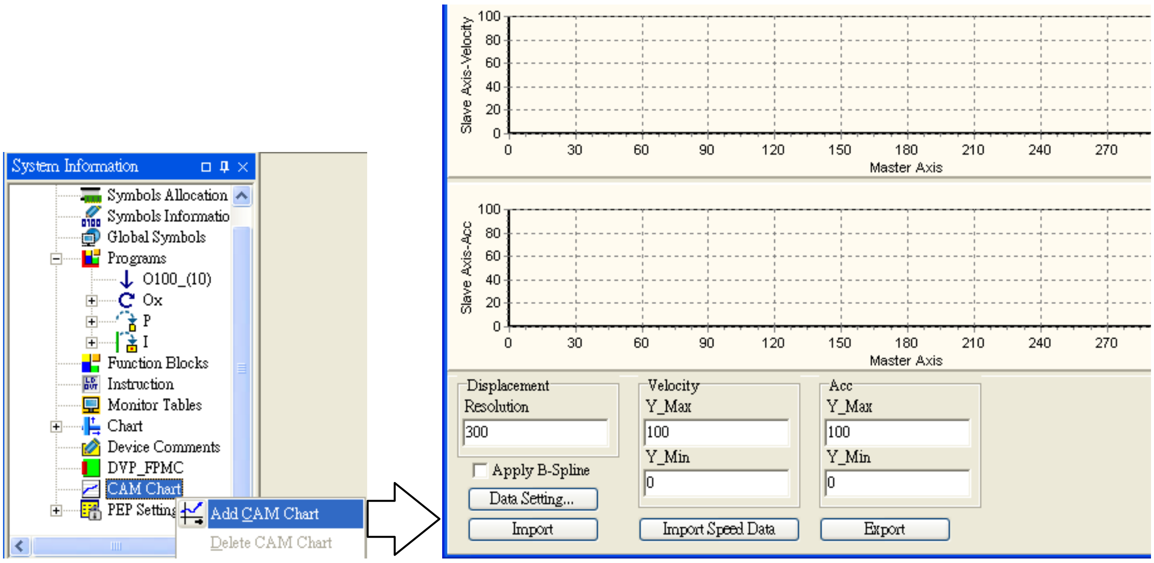
【Program】



9.4.2.3 Creating Rotary Cut Data

There are two steps of creating a rotary cut curve.

1. Creating a cam chart: Users have to create a blank cam chart in PMSoft, and then set resolution in accordance with the number of rotary cut curves. In order to create a rotary cut curve, the users need to type 300 in the **Resolution** box. Download the cam chart to a DVP-20PM series motion controller. When the DVP-20PM series motion controller operates, electronic cam data is stored in the cam chart.



2. Setting CR#10000: Users have to set the parameters related to a rotary cutter, including the distance for which the master axis specified moves, the distance for which the slave axis specified is synchronized with the master axis specified, and the synchronization ratio of the speed of the slave axis specified to the speed of the master axis specified. After CR#10000 in the special module K100 operates, a rotary cut curve will be created.

■ Example 1

【Function】

The steps of creating a rotary cut curve by means of CR#10000are described below.

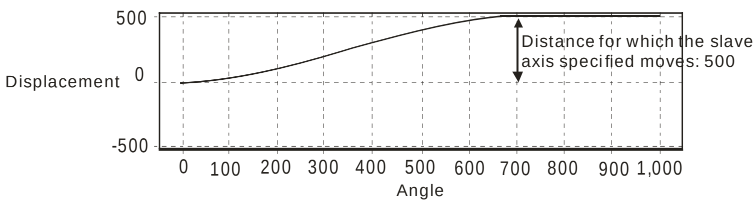
Distance for which the master axis specified moves=1000

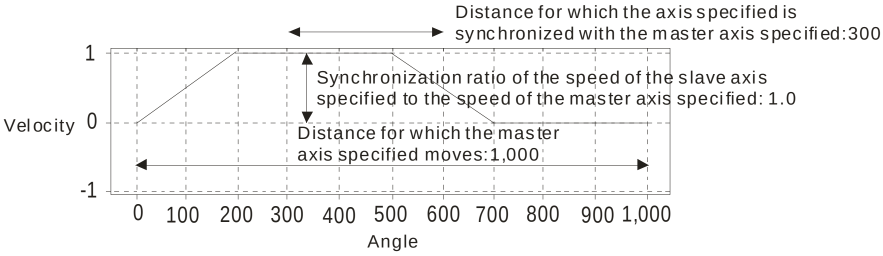
Distance for which the slave axis specified moves=500

Distance for which the slave axis specified is synchronized with the master axis specified=300

Synchronization ratio of the speed of the slave axis specified to the speed of the master axis specified=1.0

In addition, set b[14] of high word of P6=1 for electronic cam curve setting, indicating only single data real-time modification is applicable. This is to prevent rotary cut curves from being modified during program execution. The rotary cut curve created and the values of parameters are shown below.

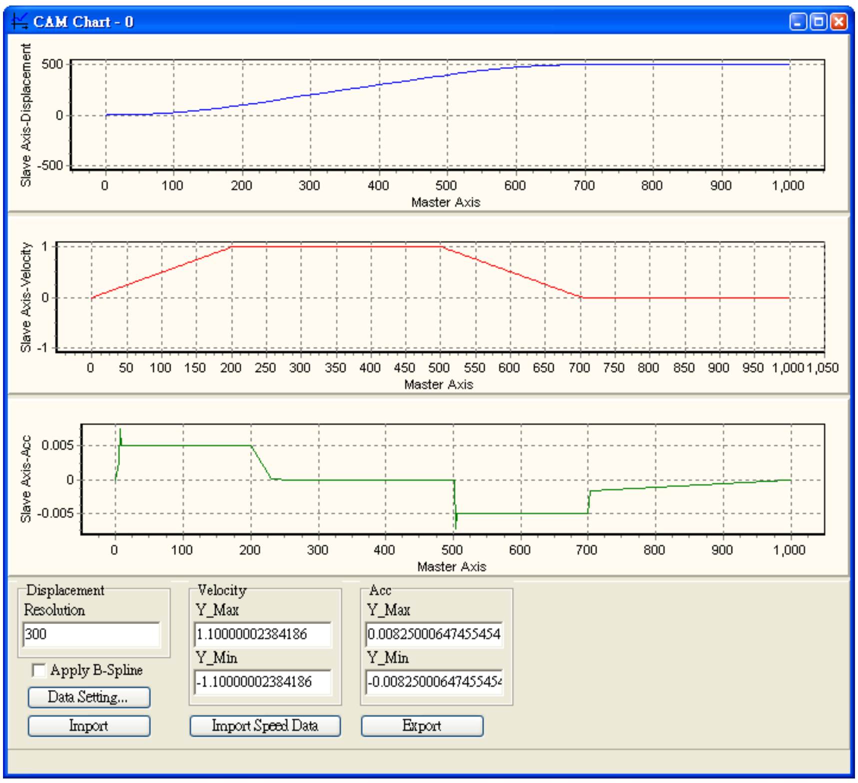




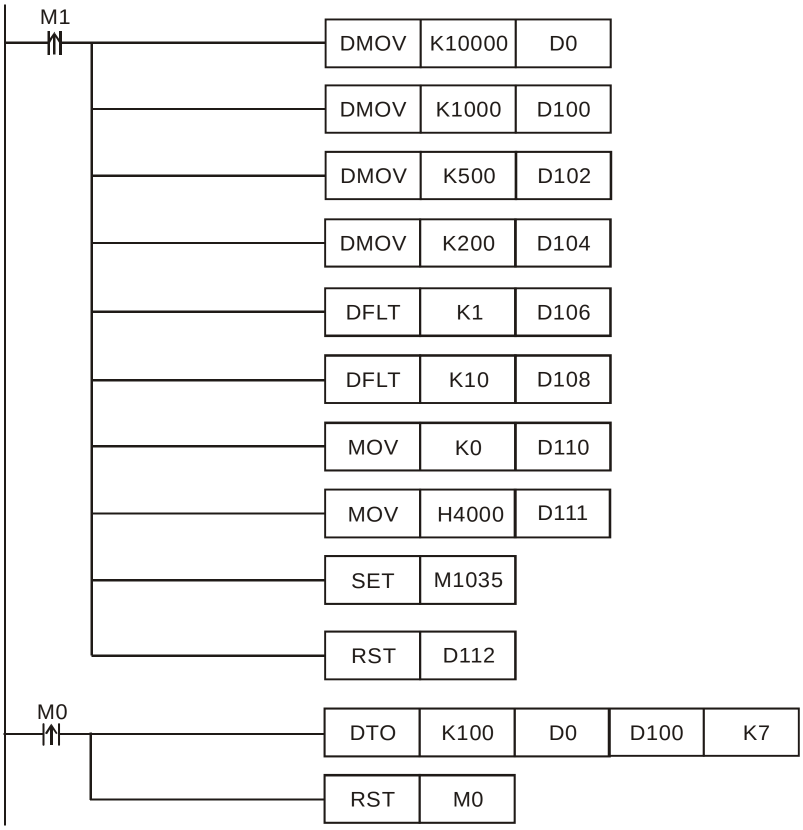
Parameter	Setting value
Distance for which the master axis specified moves	1000
Distance for which the slave axis specified moves	500
Distance for which the slave axis specified is synchronized with the master axis specified	300
Synchronization ratio of the speed of the slave axis specified to the speed of the master axis specified	1.0
Max magnification ratio	10.0
Acceleration curve	0
Cam curve	0
Cam curve setting	Only single data real-time modification is applicable
Result	0

【Steps】

1. Open a **CAM Chart** window in PMSoft, and then type 300 in the **Resolution** box.
2. Download the program created to a DVP-20PM series motion controller, and then execute the program.
3. Set M1 to write the parameters into D100~D112 and K10000 into D0.
4. Set M0 to write the data in D0~D15 into the special register K10000. After M0 is set to ON, a rotary cut curve will be created.
5. Stop the DVP-20PM series motion controller, and then upload the program in the DVP-20PM series motion controller.
6. View the first curve in the **CAM Chart-0** window. The curve is a rotary cut curve which is created automatically.



【Program】



- Example 2

【Function】

Flying shear operation can be performed by applying 2 rotary cut electronic cam curves. First, users need to create the first electronic cam curve for rotary cut. Second, design the second electronic cam curve, in which length of Slave is specified with a negative value of the previous electronic cam curve. In addition, set up the 6th data (D111 in this example) to borrow the previous electronic cam data setting and combine the 2 electronic cam curves, so that a reciprocating flying shear operation can be performed.

To create the above flying shear cam curve, 2 blank electronic cam data spaces of resolution 300 (Total 600 points) should be created before running the program. The set values of the 2 electronic cams are as below.

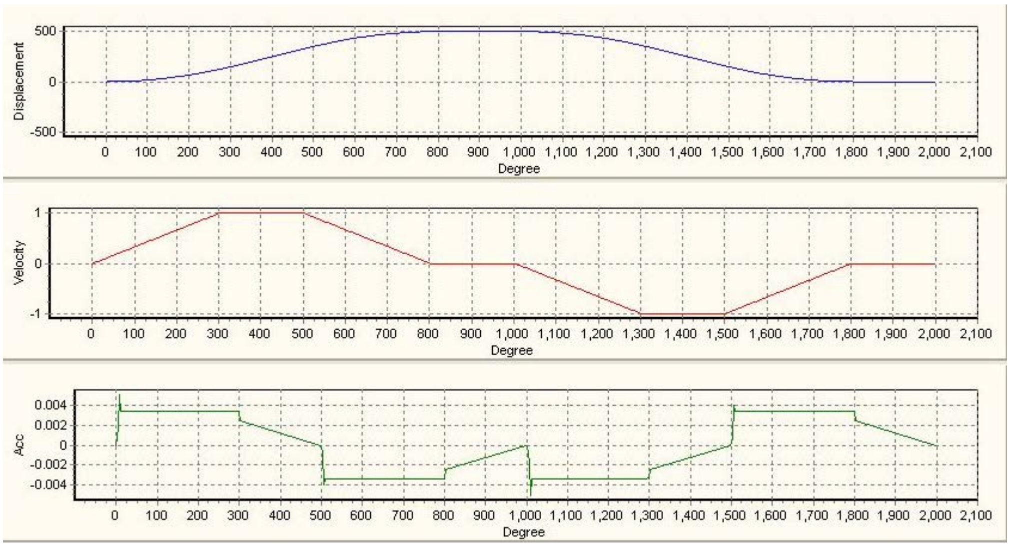
Parameter	Setting value of the first rotary cut curve	Setting value of the second rotary cut curve
Length of Master	1000	1000
Length of Slave	500	-500
Length of sync area	200	200
Magnification ratio	1.0	1.0
Max magnification ratio	10.0	10.0
Acceleration curve	0	0
Cam curve	0	0
Cam curve setting	Only single data real-time modification is applicable	1. Only single data real-time modification is applicable 2. Borrow the previous electronic cam data setting
Result	0	0

【Steps】

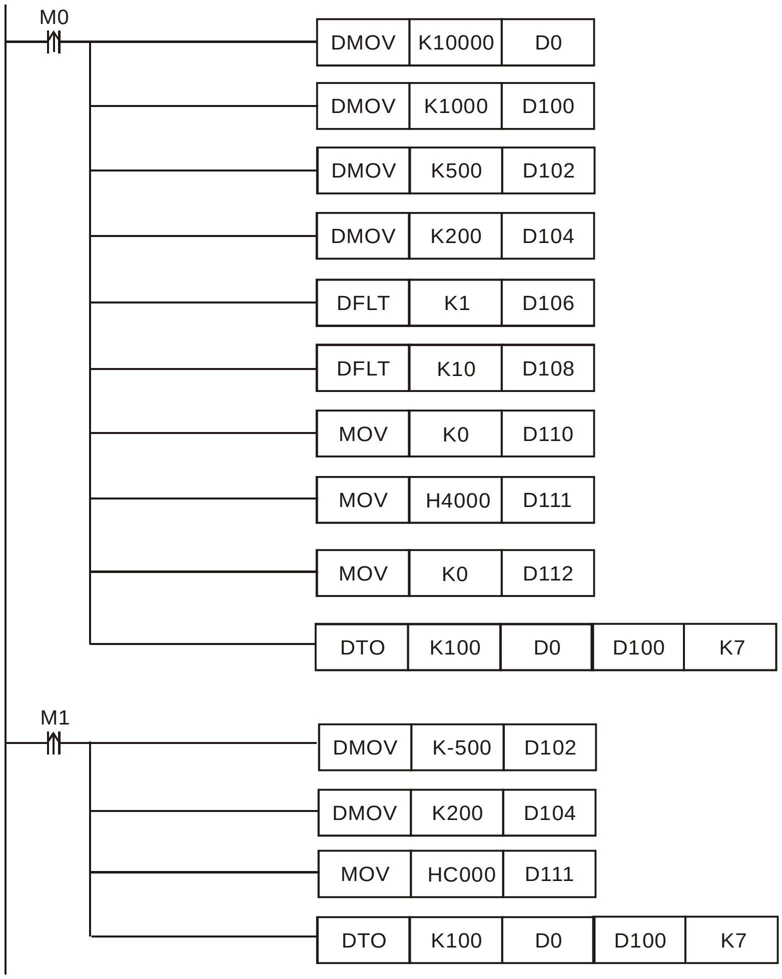
1. Open a **CAM Chart** window in PMSoft, and then type 600 in the **Resolution** box.
2. Download the program created to a DVP-20PM series motion controller, and then execute the

program.

3. Set M0 to write the parameters into D100~D112 and K10000 into D0. Write the data in D100~D112 into special register CR#10000 to generate the first rotary cut curve.
4. Set M1 to change the data in D102, D104 and D111. Write the new data into CR#10000 in the special register K100 to generate the second rotary cut curve.
5. Stop the DVP-20PM series motion controller, and then upload the program in the DVP-20PM series motion controller.
6. View the first curve in the **CAM Chart-0** window. The curve is a rotary cut curve which is created automatically.



【Program】



- Example 3

【Function】

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.

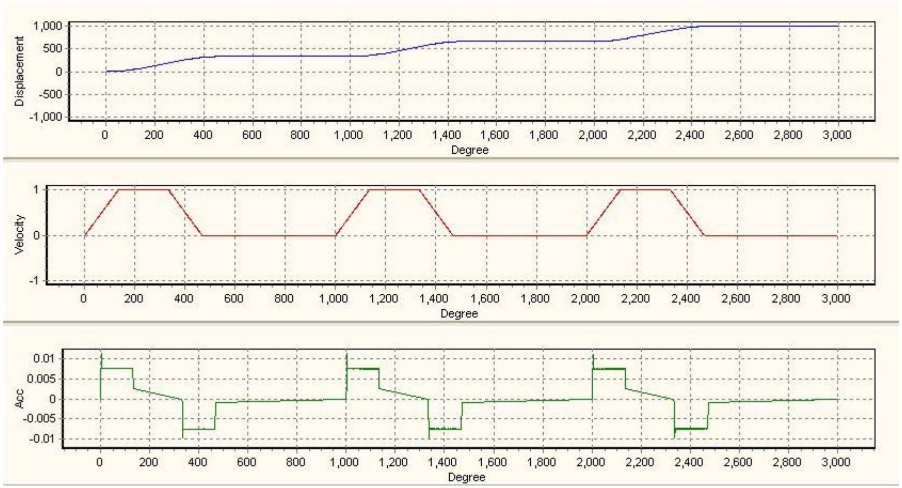
To meet this requirement, users only need to design the data setting of the first electronic cam once and set up the associated register for borrowing the data setting from the previous electronic cam curve. By this function, the 3 cam curves can be well connected and the pulses per round of Slave can be fulfilled.

The setting of parameters is shown below.

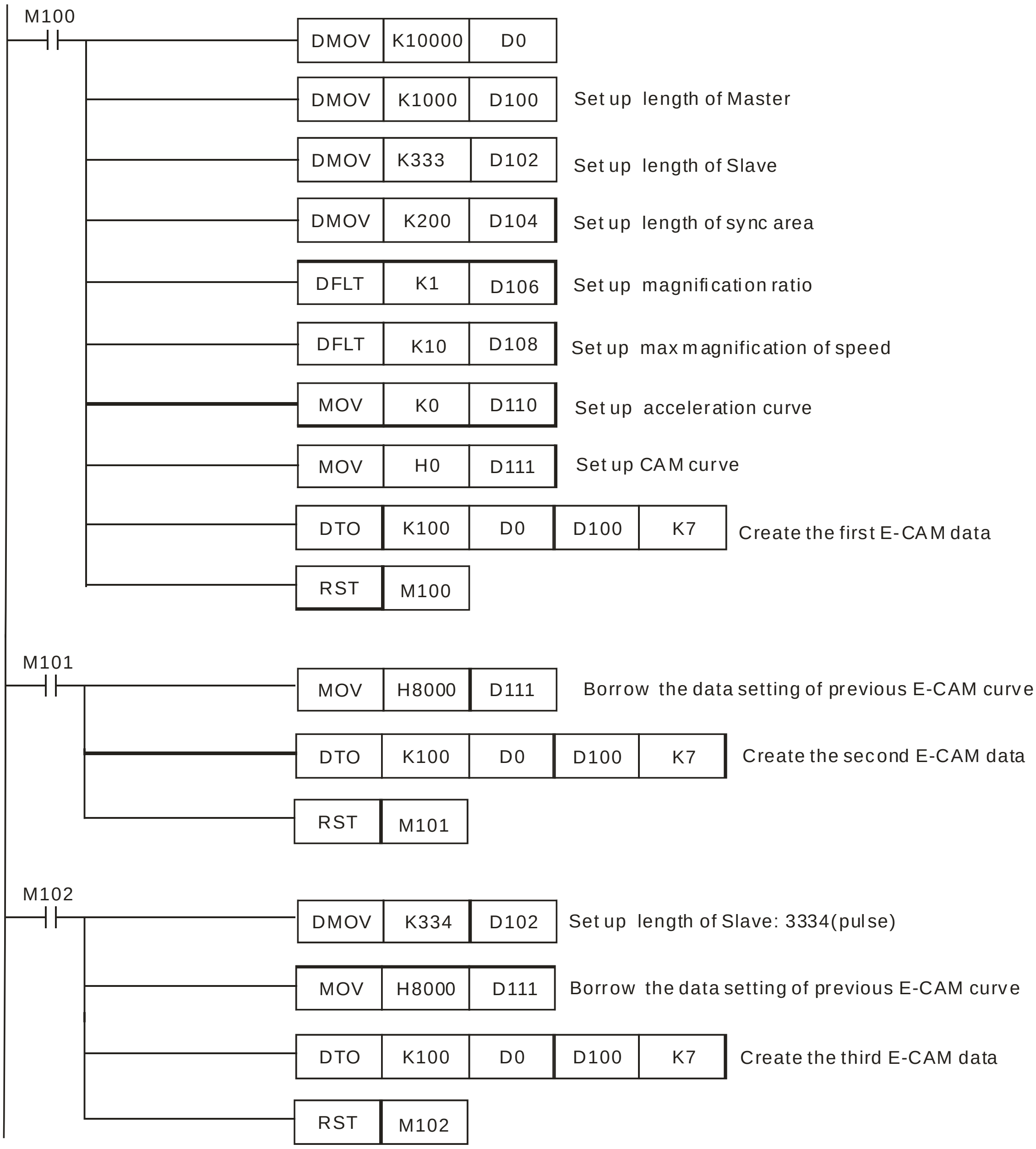
Parameter	Setting value of the first rotary cut curve	Setting value of the second rotary cut curve	Setting value of the third rotary cut curve
Length of Master	1000	1000	1000
Length of Slave	333	333	334
Length of sync area	200	200	200
Magnification ratio	1.0	1.0	1.0
Max magnification ratio	10.0	10.0	10.0
Acceleration curve	0	0	0
Cam curve	0	0	0
Cam curve setting	Only single data real-time modification is applicable	Borrow the previous electronic cam data setting	Borrow the previous electronic cam data setting
Result	0	0	0

【Steps】

1. Open a **CAM Chart** window in PMSoft, and then type 900 in the **Resolution** box.
2. Download the program created to a DVP-20PM series motion controller, and then execute the program.
3. Set M100 to write the parameters into D100~D112 and K10000 into D0. Write the data in D100~D112 into special register CR#10000 to generate the first rotary cut curve.
4. Set M101 to change the data in D111. Write the new data into CR#10000 in the special register K100 to generate the second rotary cut curve.
5. Set M102 to change the data in D102 and D111. Write the new data into CR#10000 in the special register K100 to generate the third rotary cut curve.
6. Stop the DVP-20PM series motion controller, and then upload the program in the AH500 series motion control module.
7. View the first curve in the **CAM Chart-0** window. The curve is an electronic cam curve which is created automatically.



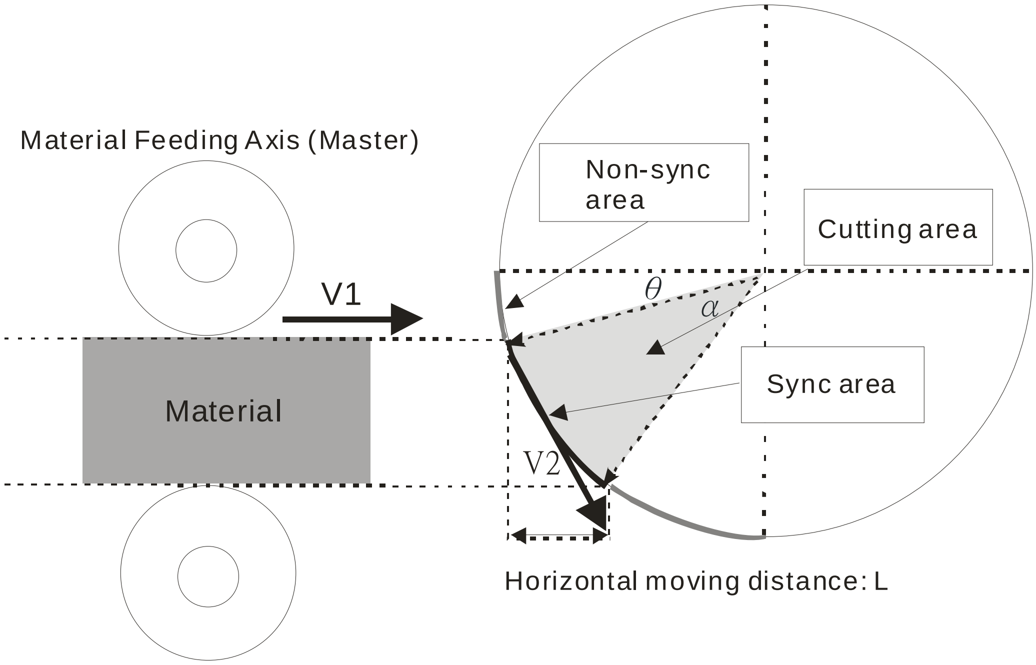
【Program】



9.4.2.4 Electronic Cam Data for Cutting Thick Material

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 electronic 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 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 syn 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.

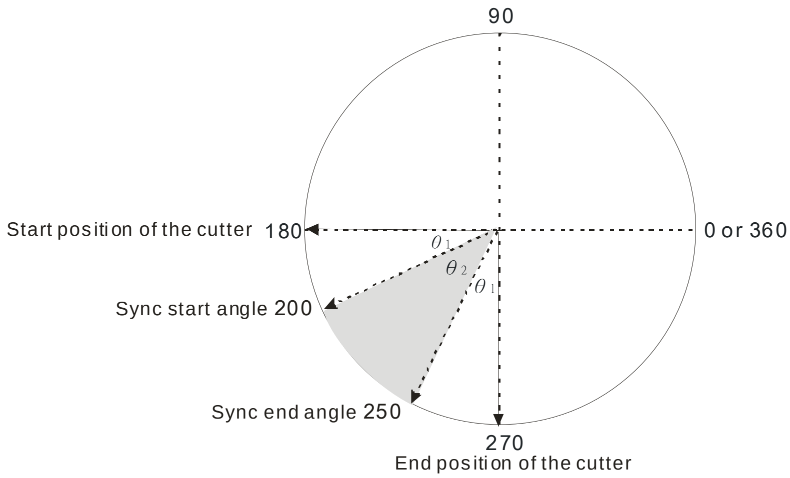
Also, care should be taken on setting the contact angle, which is related to the thickness of the material and the vertical position of the cutting device. When setting the contact angle, make sure the contact angle is a little smaller or equal to the actual contact angle otherwise an arc cutting plane 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.

- Pulses can be equally divided.

  For even number of cutters, select “Equally divide non-sync area” function by bit0 of Data 1, and DVP-20PM 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.

$$\alpha = Sync\ end\ angle\ of\ Slave - Sync\ start\ angle\ of\ Slave$$

$$\theta = \left(\frac{360}{Pulses\ per\ round}\right) \times \left(\frac{Pulses\ per\ round}{Number\ of\ cutters} - \alpha \cdot \frac{Pulses\ per\ round}{360}\right) \div 2$$

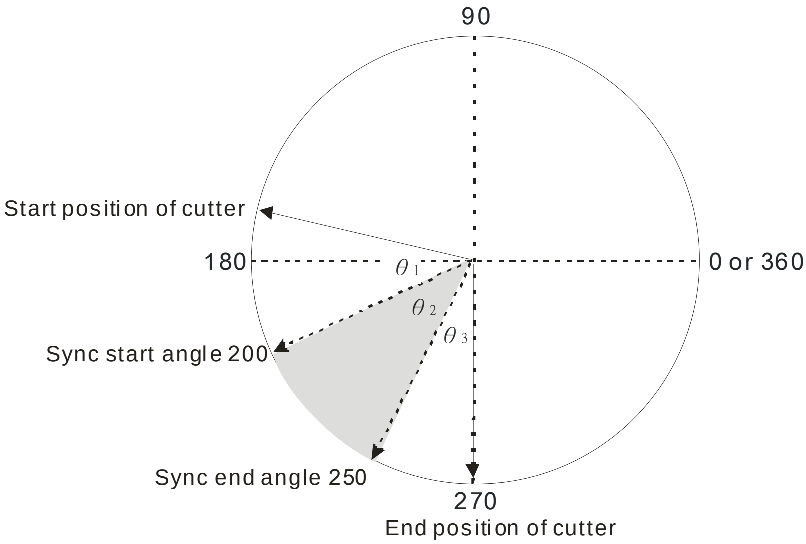


The actual adjustment steps are: 1. Selecting the sync start and sync end angle; 2. Aligning the cutter with the start position of the cutter; 3. Executing electronic cam operation; 4. Inspecting the cutting plane on material

If the cutting plane is bended, conduct fine tuning on the sync start/end angle until the cutting plane is flat and smooth.

- 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 electronic 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 electronic cam setting. When the above parameters are ready, run the thick material cutting electronic cam and conduct fine-tuning on the parameters of each cutter until the best results are obtained.



- Example

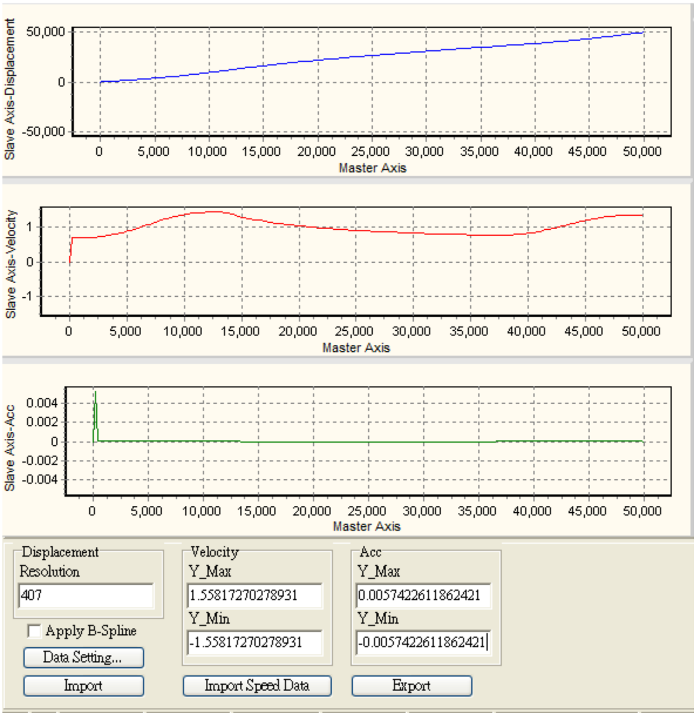
【Function】

The example is the application of using DTO instruction for creating thick material cutting electronic cam. The required parameters are listed in the table below. Users 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: θ1=210 degrees, and sync end angle of Slave: θ2=250 degrees. The cutting length can be calculated according to the parameters above: 170×314÷4=13345. Parameters for thick-material cutting are listed as below.

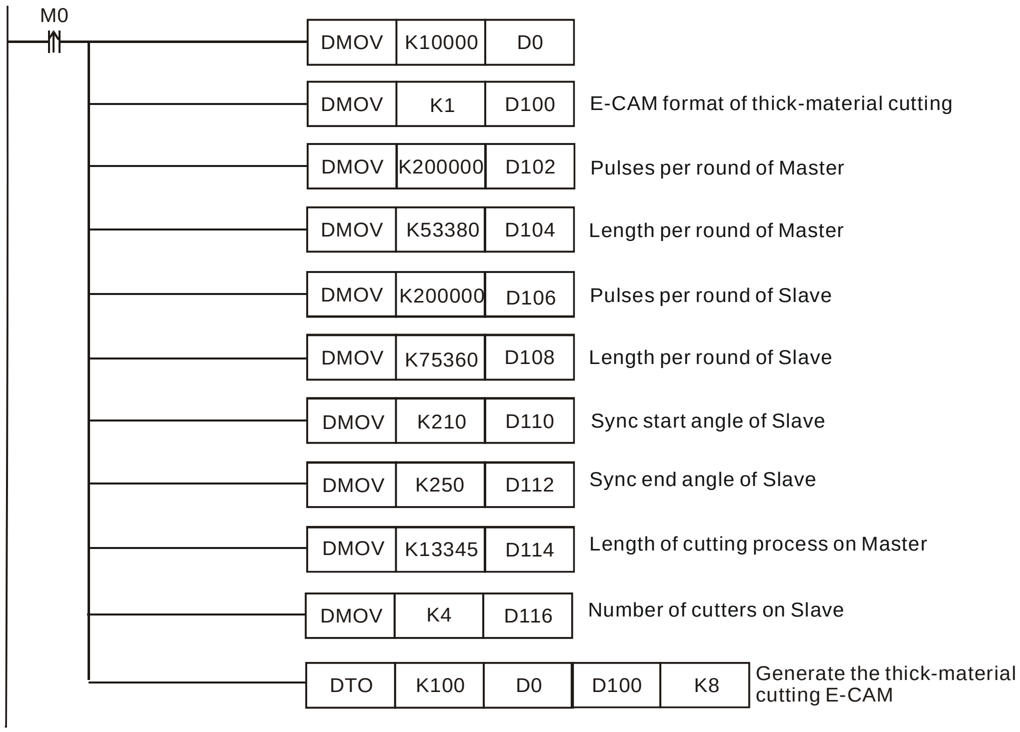
Parameter	Setting value
Electronic cam format	0
Pulses per round of Master	200000
Length per round of Master	53380
Pulses per round of Slave	200000
Length per round of Slave	75360
Sync start angle of Slave:180~360	210
Sync end angle of Slave:180~360	250
Length of cutting process on Master	13345
Number 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	-

【Steps】

1. Open a **CAM Chart** window in PMSoft, and then type 407 in the **Resolution** box.
2. Download the program created to a DVP-20PM series motion controller, and then execute the program.
3. Set M0 to write the parameters into D100~D116 and K10000 into D0. Write the data in D100~D116 into CR#10000 in the special module K100 to generate a thick material cutting electronic cam curve.
4. Stop the DVP-20PM series motion controller, and then upload the program in the DVP-20PM series motion controller.
5. View the first curve in the **CAM Chart-0** window. The curve is a thick material cutting electronic cam curve which is created automatically.

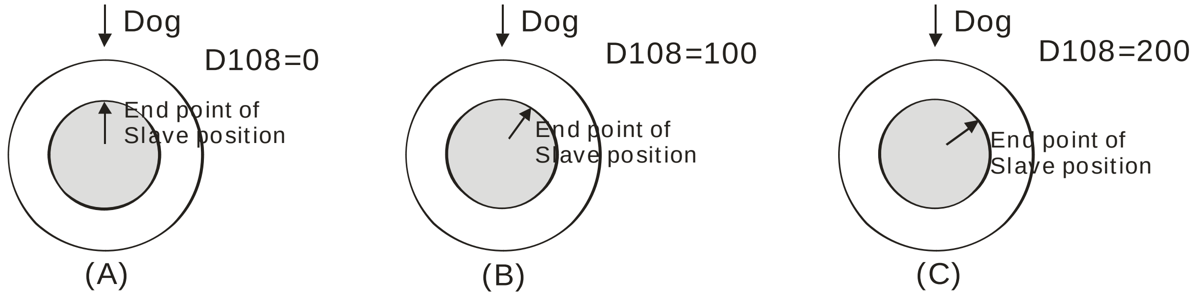


【Program】



9.4.2.5 Offset Compensation for a Slave Error

In field application, errors occur between the actual output position on device and the target output position on electronic cam curve. In addition, the error could be larger as the increase of execution times. Aiming at this problem users 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 (Figure A). Second, compare the value in backup register (D128) with D108, which has errors to the initial value (Figure B and Figure C). Third, take the difference between the 2 registers as the reference value for offset compensation.



Application method:

1. Read initial Slave error. The initial error is the value obtained by subtracting the captured value at DOG with the captured value at electronic cam completion.
2. When the Slave error increases in the next cycle, subtract the increased Slave error with the initial Slave error, and users can have the real Slave error.
3. Subtract Length per round of Slave, which is one of the electronic cam parameter, with the real Slave error and users can have the real error between the results of electronic cam output and the device output. When the real error is between "± half of Slave length", or when the real error exceeds "± half of Slave length", users conduct offset compensation for the error.
4. Compensation

$$\begin{cases} \text{Real error} > \frac{1}{2}\ \text{length of Slave} \Rightarrow \text{Offset} = (\text{Real error - length of Slave}) \times \text{Offset ratio} \\ \text{Real error} < -\frac{1}{2}\ \text{length of Slave} \Rightarrow \text{Offset} = (\text{Real error} + \text{length of Slave}) \times \text{Offset ratio} \end{cases}$$

5. In the next cycle, input the modified length of Slave (with offset) by DTO instruction and the offset compensation is completed.

■ Example

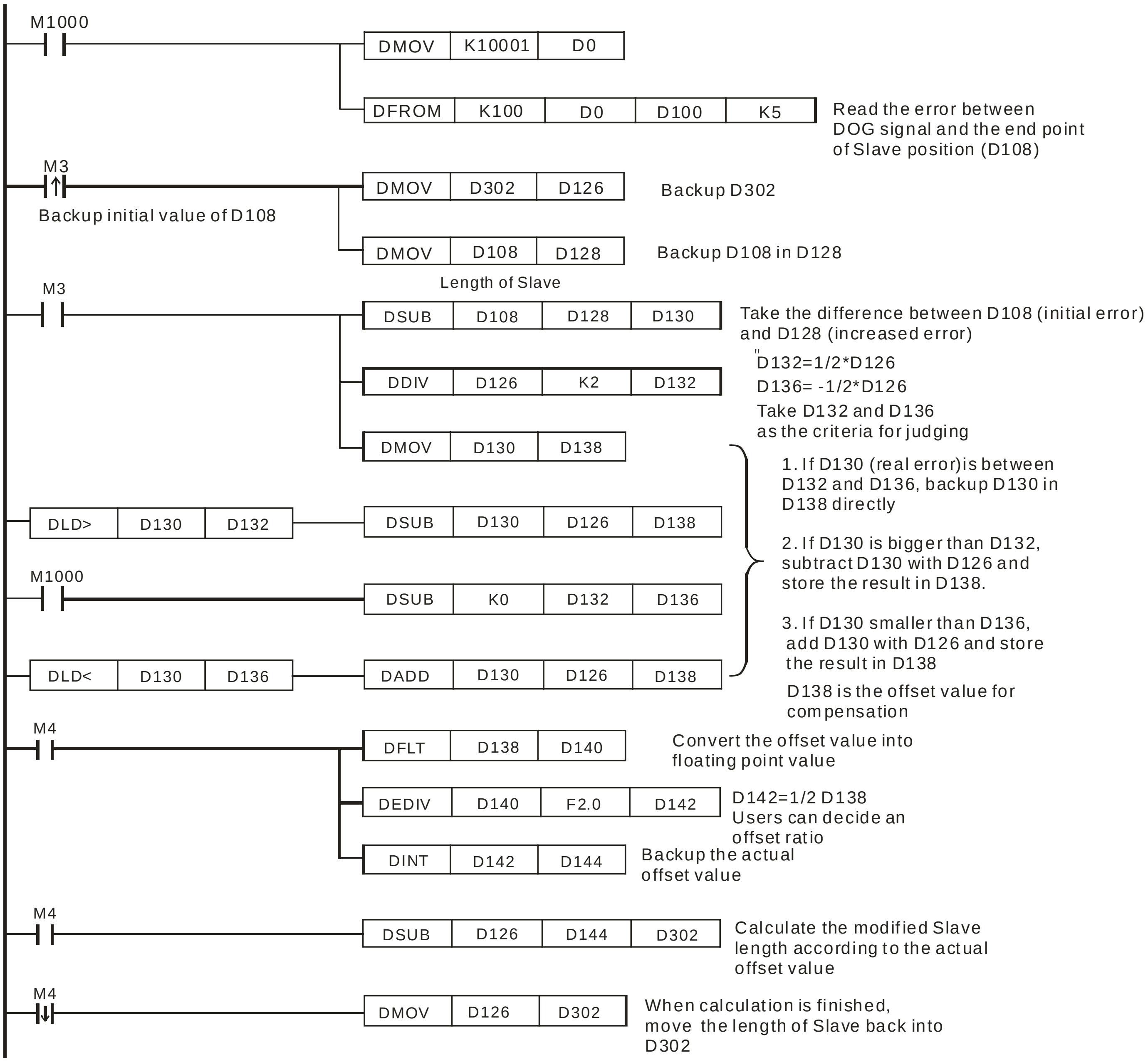
【Function】

The example is used for judging if real error between DOG switch and electronic 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.

【Steps】

1. Run the program and read the electronic cam status by DFROM instruction. Read the data in CR#10001 of special module K100 to D100~D108.
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.
3. Real error > 1/2 length of Slave→Offset=(Real error-Length of Slave)×Offset ratio
4. Real error < 1/2 length of Slave→Offset=(Real error+Length of Slave)×Offset ratio
5. Set M4 to calculate the modified length of Slave with offset compensation.
6. When M4 is reset, fill the original length of Slave back to D302.

【Program】



※ Users can modify the offset ratio (0.5*D138 in this example) according to actual needs.

※ D302 has to be applied in DTO instruction for creating rotary cut E-CAM Data with offset compensation.

9.4.2.6 Application of a Flying Shear–Checking the Position of a Cutter

In flying shear 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).

- Example

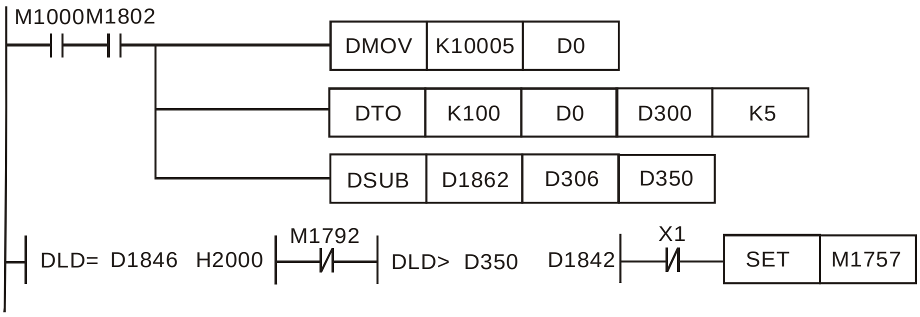
【Function】

Check whether the cutter position is correct and automatically determine the execution of M1757.

【Steps】

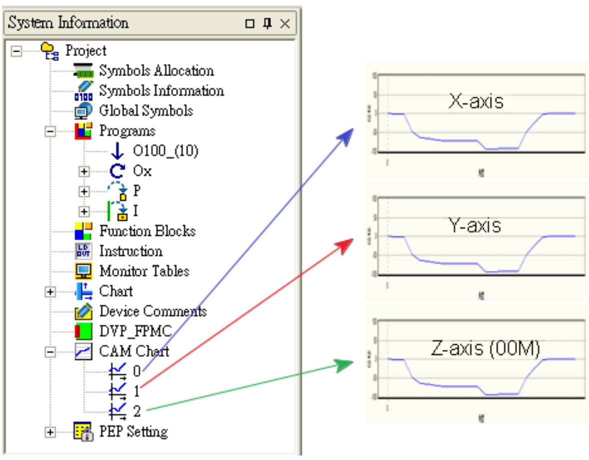
1. Run the program and read the electronic cam status in capture mode by DFROM instruction. Read the data in CR#10005 (Capture mode) of special module K100 to D300~D308 at the end of cycle.
2. Subtract the captured Master position (D306) from the number of accumulated MPG input pulses (D1862), and users can obtain the actual Master position in current cycle (D350).
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.

【Program】



9.5 Multi-axis Cam

Single-axis electronic cam can be executed in cyclic operation and noncyclic operation. Unlike single-axis electronic cam, multi-axis cam can only be executed in cyclic operation. For multi-axis cam, only one set of electronic cam charts can be applied, i.e. cam chart 0~cam chart 2 correspond to the X-axis, the Y-axis, and the Z-axis.



Same as single-axis cyclic electronic cam, multi-axis 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 to ON (D1846 =H'2000) to enable multi-axis cam.
2. Set up work modes (X-D1847 Y-D1927 Z-D2007)

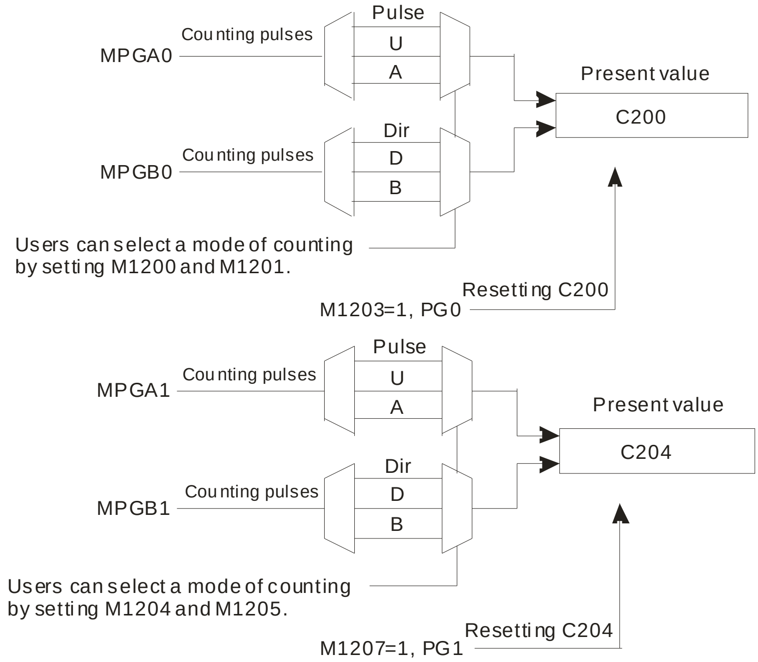
   For multi-axis cam, b12 and b11 in D1847 should be set to 01 (bit11=1). When the Y-axis and the Z-axis are applied, bit11 in D1927 and D2007 should also be set to 1.
3. C200/C204

   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.

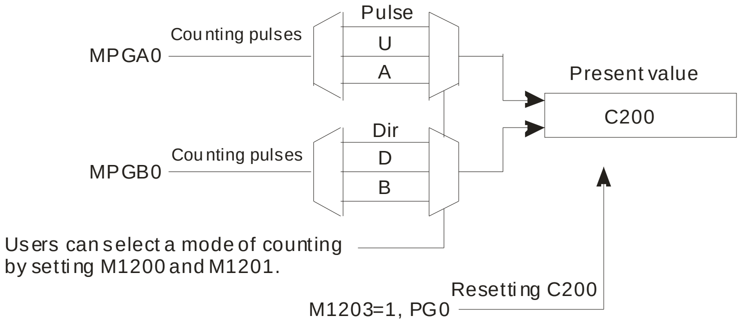
   Modes of counting for C200/C204

Device	Mode of counting for C200	Device	Mode of counting for C204
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* (One time the frequency of A/B-phase inputs)	M1204=0, M1205=1	A/B (One time the frequency of A/B-phase inputs)
M1200=1, M1201=1	4A/B (Four times the frequency of A/B-phase inputs)	M1204=1, M1205=1	4A/B (Four times the frequency of A/B-phase inputs)

M1908=Off: The input signals for C200 are controlled by A0±/B0±, and the reset signals for C200 are controlled by PG0. The input signals for C204 are controlled by A1±/B1±, and the reset signals for C204 are controlled by PG1.



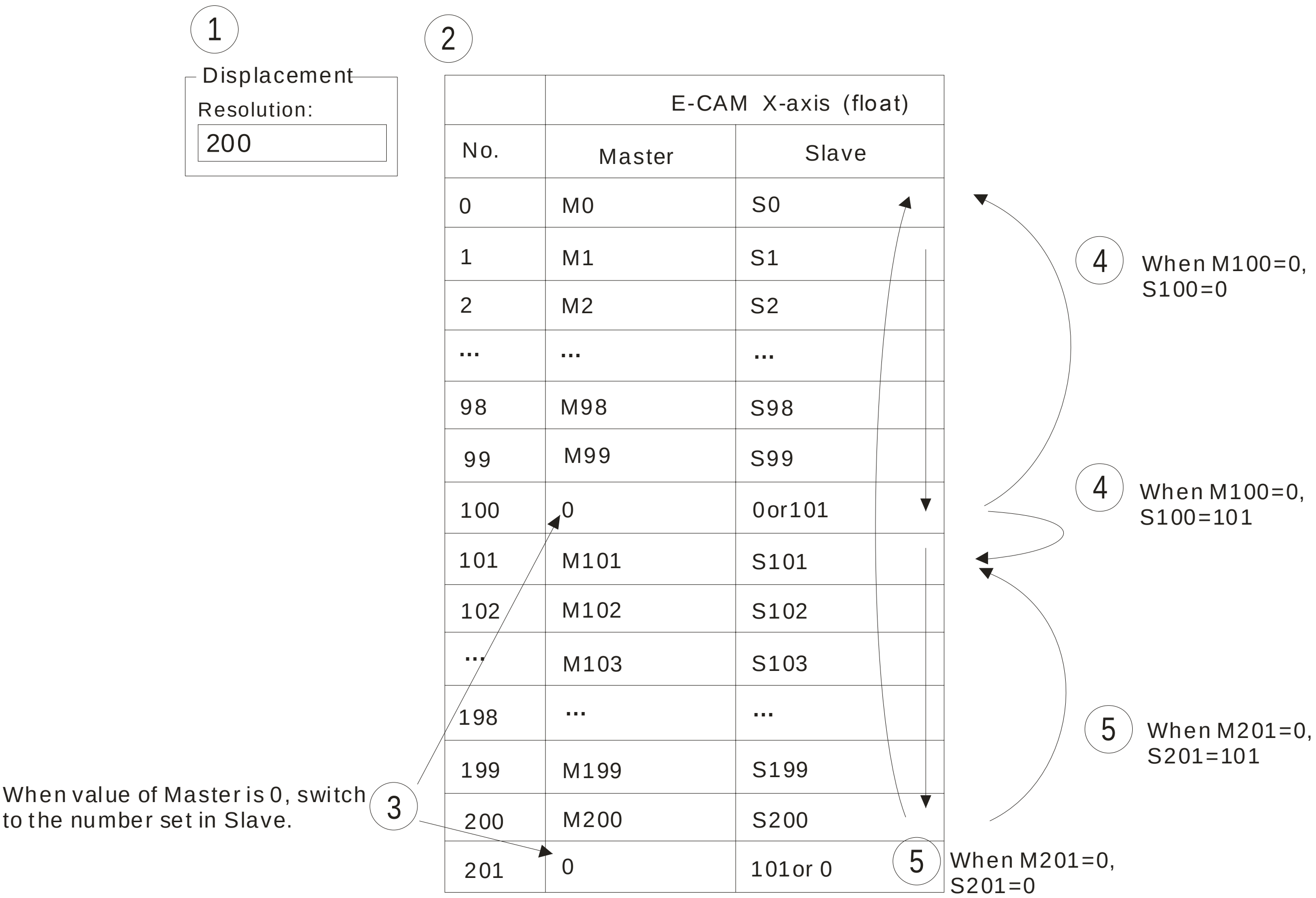
M1908=On: C200 are for the X-axis, the Y-axis, and the Z-axis. The input signals for C200 are controlled by A0±/B0±, and the reset signals for C200 are controlled by PG0.



There are 2 execution patterns of a multi-axis cam

1. Point-to-point switching pattern

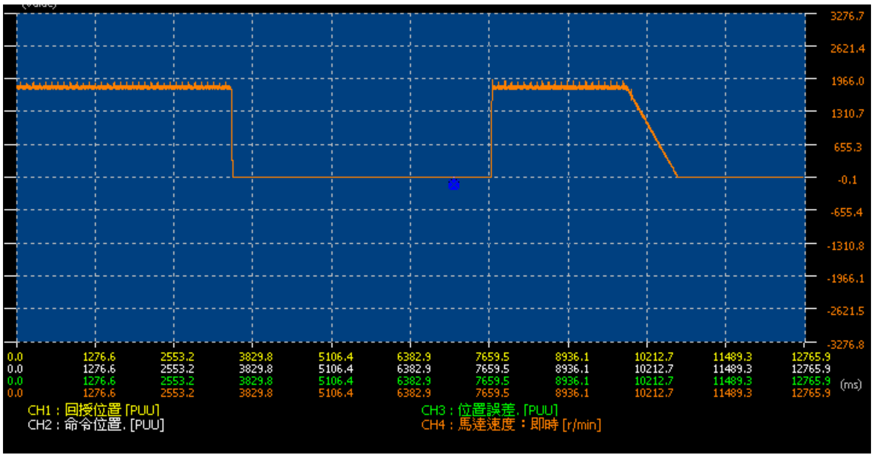
   In multi-axis cam, users can separate an M-point electronic cam data into several sections for realizing real-time cam data switching between sections. Only one section will be executed for one time and users can add new data into the next section. For switching to next section, insert an additional 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 point-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 users use an N-point electronic cam data and separate the beginning 202 points into 2 sections.



	E-CAM X-axis (float)	
No.	Master	Slave
0	M0	S0
1	M1	S1
2	M2	S2
...	...	...
98	M98	S98
99	M99	S99
100	0	0or101
101	M101	S101
102	M102	S102
...	M103	S103
198	...	...
199	M199	S199
200	M200	S200
201	0	101or 0

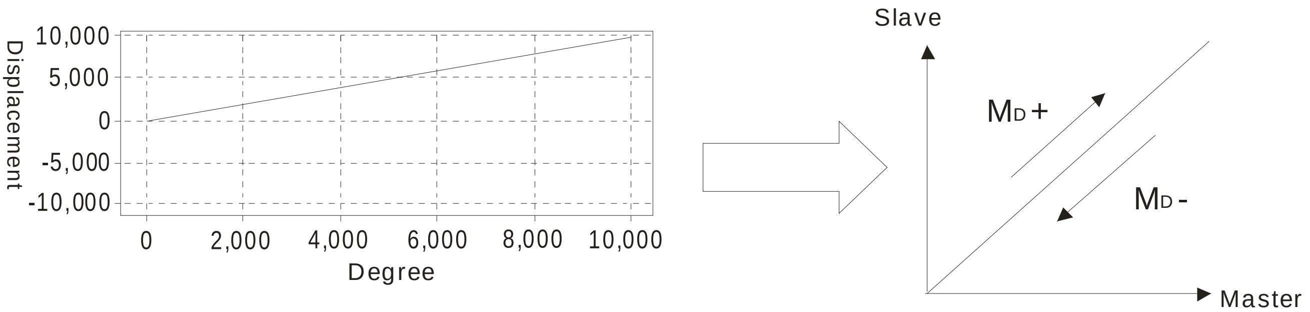
Set value of Slave in No.100 and No.201 indicates the target number for next execution. Therefore, before the switching point is reached users can insert or modify the data in the next section. However, care should be taken on setting the starting value of the second section. For second section, if the 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.

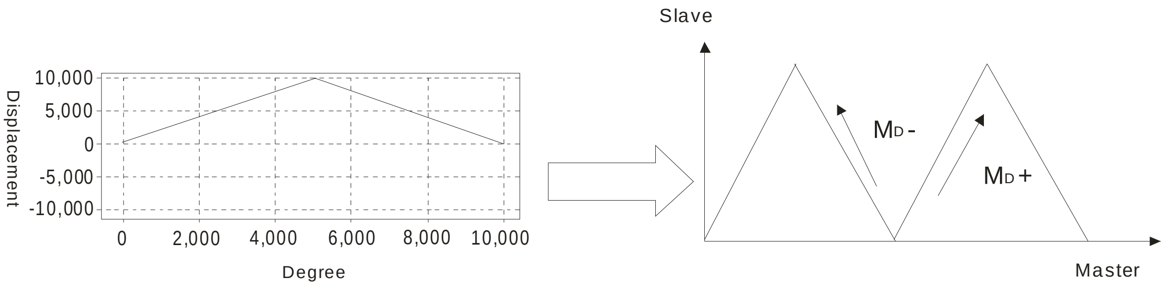
In addition, for point-to-point switching pattern, if electronic cam max frequency (DD1840) is set as 0Hz and the value of deceleration time T_{DEC} (D1837) is specified, electronic cam will decelerate to stop. The below diagram indicates the difference between normal stop operation and stop operation with deceleration.



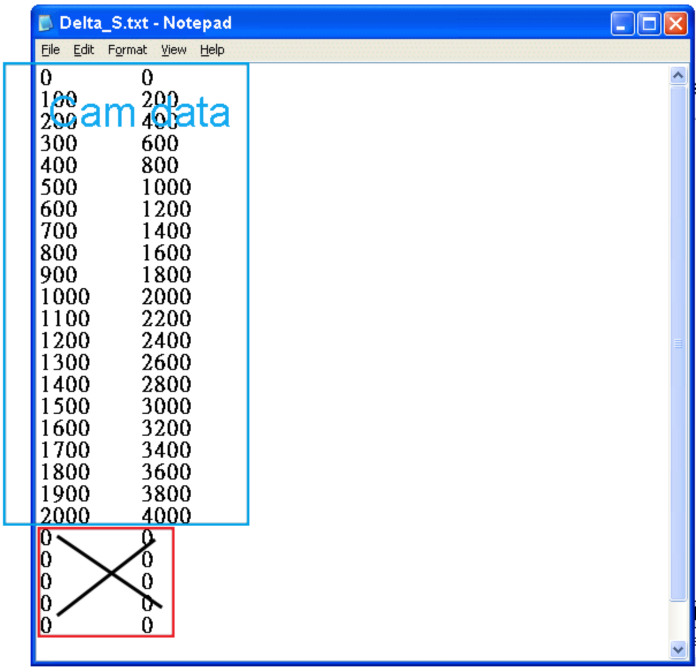
2. Forward /Reverse running Pattern

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 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 electronic cam charts.





In forward/reverse running pattern, master position 0 other than the start point should not exist.



9.6 Practical Application of an Electronic Cam

9.6.1 Application of a Winding Machine

In this application example users apply DVP-20PM for controlling the automatic high-speed 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 coils. The winding system with DVP-20PM 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 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-20PM is designed specifically for motion control applications. DVP-20PM 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 users apply the electronic cam function to solve the switching delay problem of reverse winding situation explained above.

9.6.1.1 Operation of a High-speed Winding Device

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 electronic cam operation. Copper wires are winded on the coil die through the winding flyer. Winding device is the major part of winding 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 electronic 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.

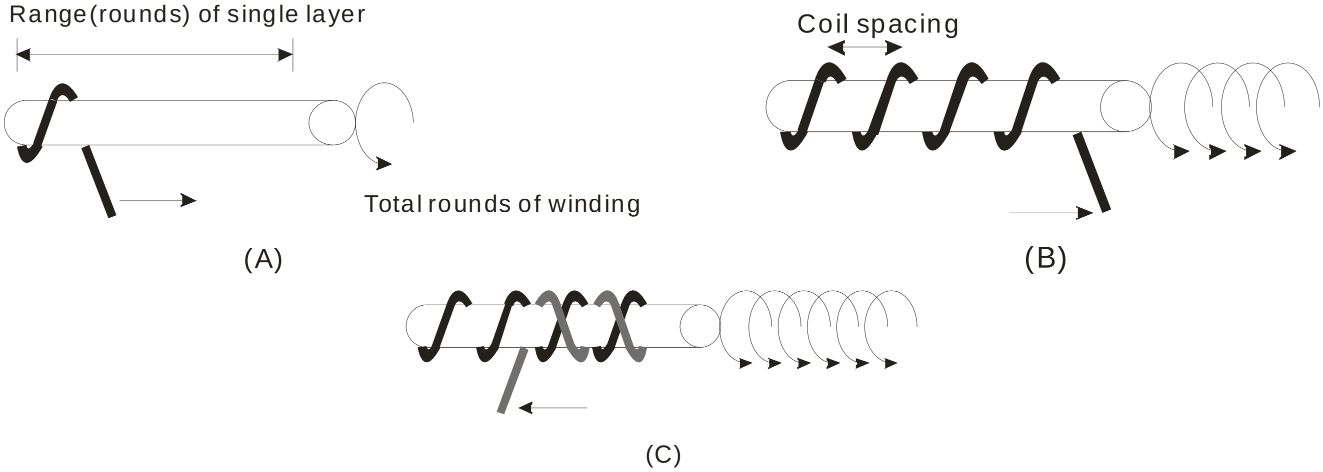
(7) Die releasing device
Die releasing device consists of index stepping motor and pneumatic die releasing device, taking the completed coil off the coil die.

(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 users 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, users can perform precise motion control between winding shaft and the coil shaft, so that the accuracy of coil winding can be obtained.

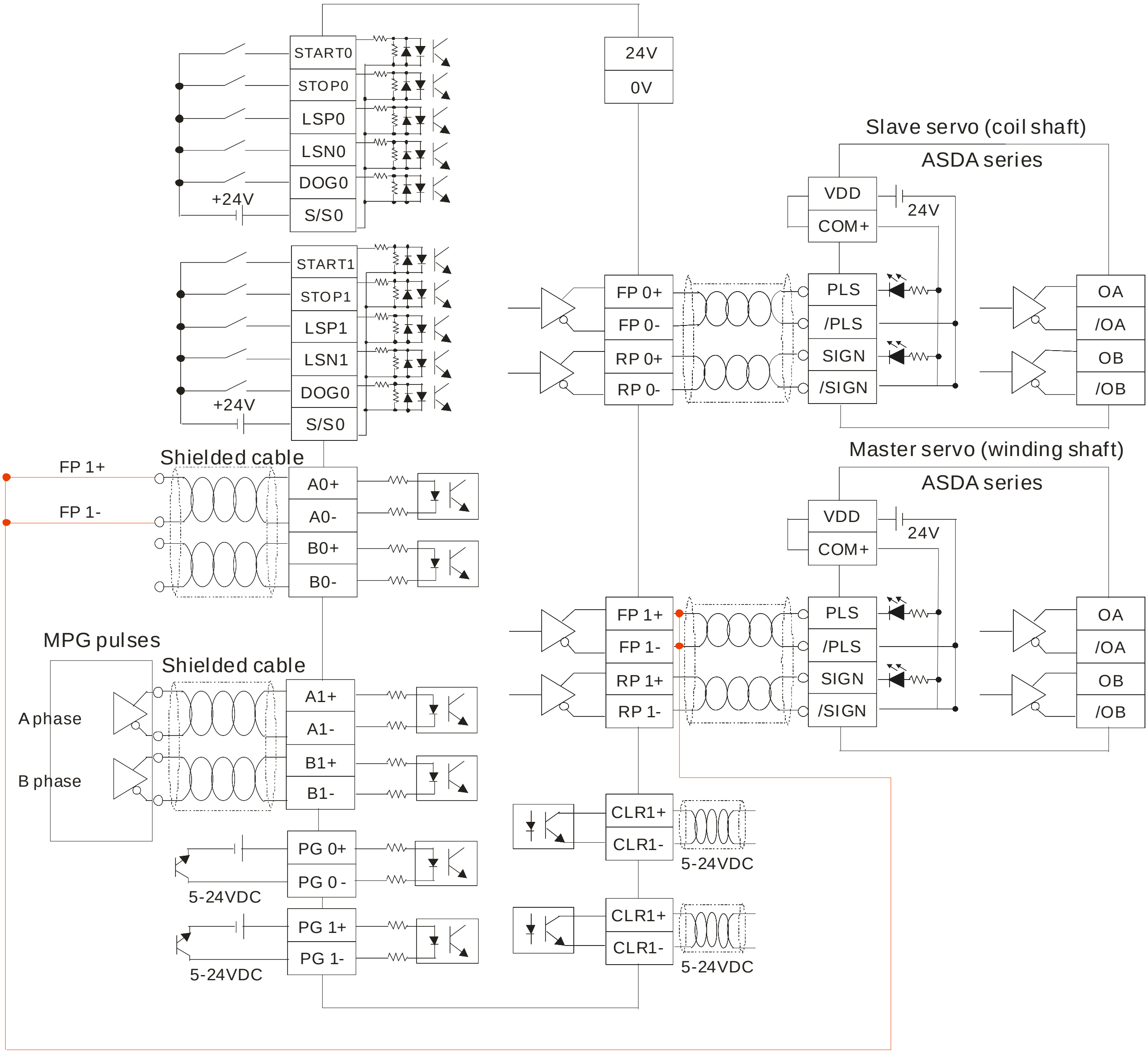
High speed winding on master shaft and slave shaft can be performed by executing single speed positioning on winding device (3), together with electronic cam operation on coil device (4). Other operations are similar to general sequential ladder control and can be executed by general functions of PLC.

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 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 electronic 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 same direction.

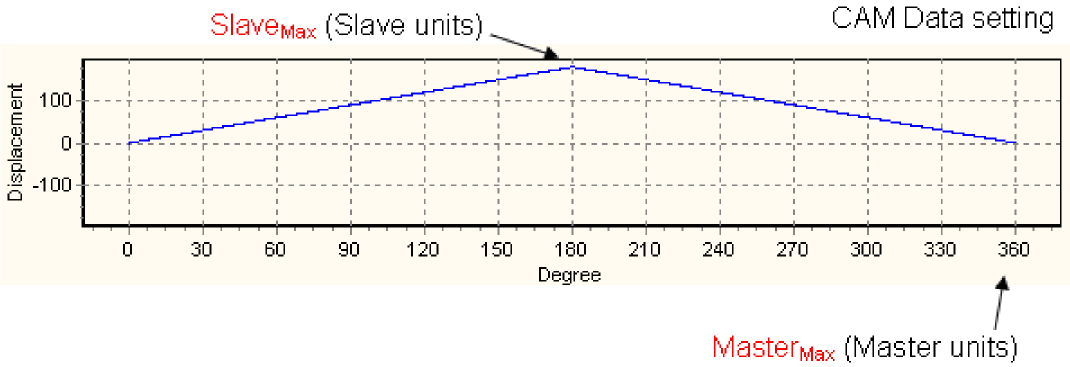
9.6.1.2 Wiring Hardware



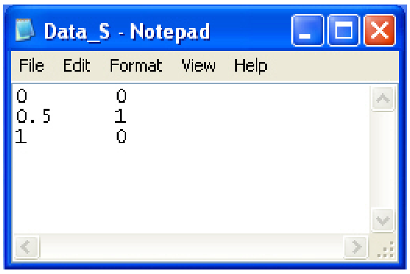
In above wiring diagram, users connect output terminal of Y axis (FP1+, FP1-) to the MPG input terminals (A0+, A0-) as the source of Master. By this wiring method, winding operation of coil shaft (Slave) will not be influenced no matter the winding direction of winding shaft (Master) is clockwise or counter-clockwise. Therefore, in this case users only connect single direction for Master.

9.6.1.3 Creating an Electronic Cam Curve

Create the electronic cam chart shown below in PMSoft.



To create electronic cam data, import the file "Data_S.txt".



Firstly in this file users set the base unit of electronic cam by specifying SlaveMax=1 and MasterMax=1. Second, select motor unit for electronic cam operation by setting up b1/b0 of D1816. Third, specify the magnification of electronic cam data according to the input parameters below. In addition, users can also select motor unit then real-time modify the SlaveMax/MasterMax value by DTO instruction. When input parameters are ready, users 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)	$=N2xB_{Master}$
	Coil shaft (Slave/X axis)	$Master_{Max}$ (pulses)	$=2xN1xB_{Master}$ "2" indicates double layer winding.
		$Slave_{max}$ (pulses)	$=N1xD/C_{Slave}$

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)	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)	$B_{Master=}3600$
		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)	$C_{Slave}=A_{Slave}/B_{Slave}$
Setting	Winding shaft (Master/Y axis)	Length of single speed positioning (pulses)	$=N2xB_{Master=}=80x3600=288000$
	Coil shaft (Slave/X axis)	$Master_{Max}$ (pulses)	$=2xN1xB_{Master}=2x10x3600=72000$ "2" indicates double layer winding.
		$Slave_{max}$ (pulses)	$=N1xD/C_{Slave}=10x0.2/(0.1/100)$ =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)	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)	$B_{Master}=3600$
		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)	$C_{Slave}=A_{Slave}/B_{Slave}$
Setting	Winding shaft (Master/Y axis)	Length of single speed positioning (pulses)	$=N2xB_{Master}=100x3600=360000$
	Coil shaft (Slave/X axis)	$Master_{Max}$ (pulses)	$=2xN1xB_{Master}=2x20x3600=144000$ "2" indicates double layer winding.
		$Slave_{max}$ (pulses)	$=N1xD/C_{Slave}=20x0.3/(0.1/100)$ =6000

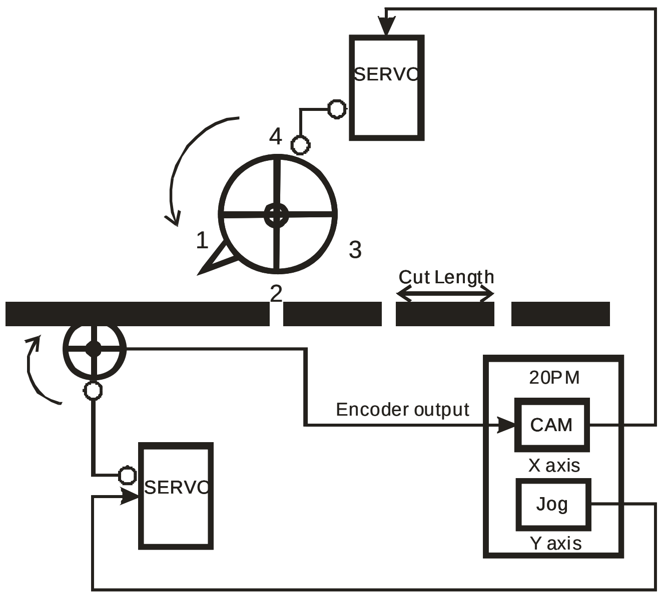
9.6.2 Application of a Rotary Cutter

In the application of cutting materials on a feeding belt, a traditional approach is that a cutting roller will rotate after a feeding roller rotates for a certain length, and the alternation of feeding materials and cutting the materials is repeated. The disadvantage of this approach is that the acceleration/deceleration needed in order for a feeding roller to rotate/stop decreases production efficiency. As a result, a new approach is that materials are fed continuously. There are two ways of cutting materials on a feeding belt. They are rotary cut and flying shear. Flying shear is reciprocating motion, while rotary cut is unidirectional motion. The cam curve for rotary cut is different from the cam curve for flying shear. In addition, thick material cutting also works in the same direction as rotary cut. The difference between think material cutting and rotary cut is the electronic cam curve variation in cutting process. Unlike the constant proportion curve of rotary cut’s sync area, the electronic cam curve of thick material cutting varies during the cutting process between the start and the end of cutting. The 3 electronic cam applications are explained as below.

9.6.2.1 Operation of a Rotary Cutter

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.

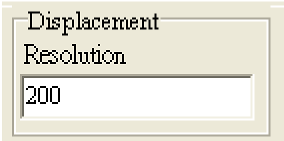
The operation and simple wiring of rotary cut is illustrated as below. Positions 1, 2, 3 and 4 respectively indicate starting position, middle position, end position and ready position of speed synchronization. When Master (Y axis) executes, Slave (X axis) accelerates from position 4 to position 1, reaching the synchronizing speed. The sync speed is maintained from position 1 to position 3. After position3, Slave decelerates and returns to position 4. The cycle repeats for continuous rotary cut operation.



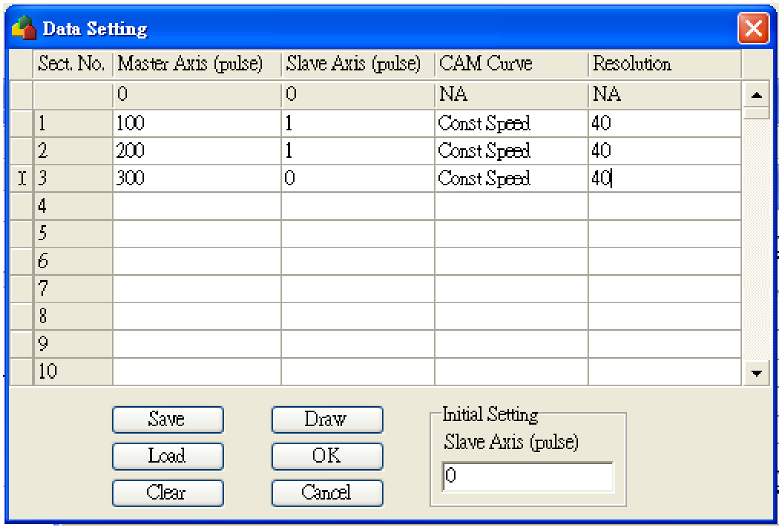
9.6.2.2 Creating an Electronic Cam Curve

In electronic cam design process, PMSoft creates electronic cam data by operating displacement data to generate the velocity data and the acc data. However, in this case users need to control the velocity between Slave and Master. To obtain the velocity, users 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.

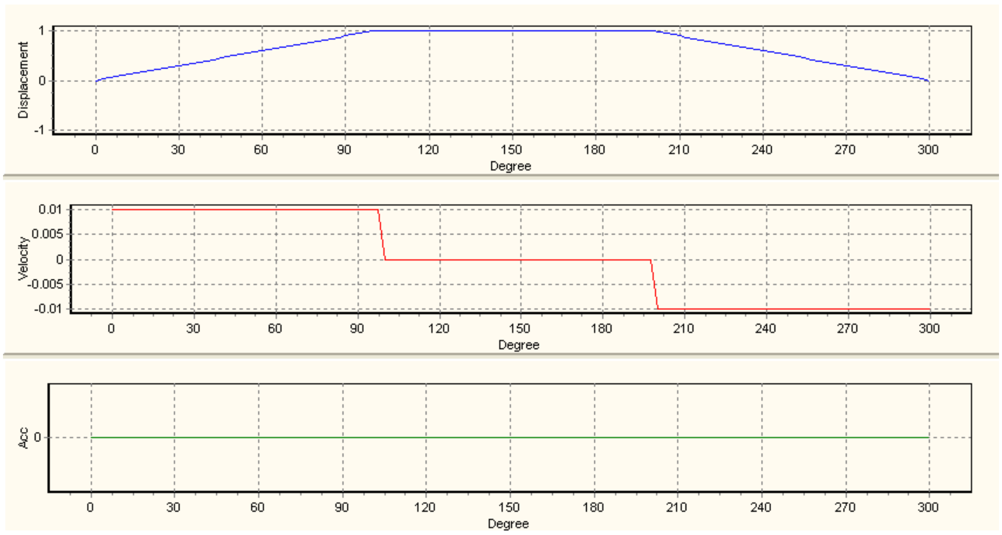
1. Set the max resolution as 200.



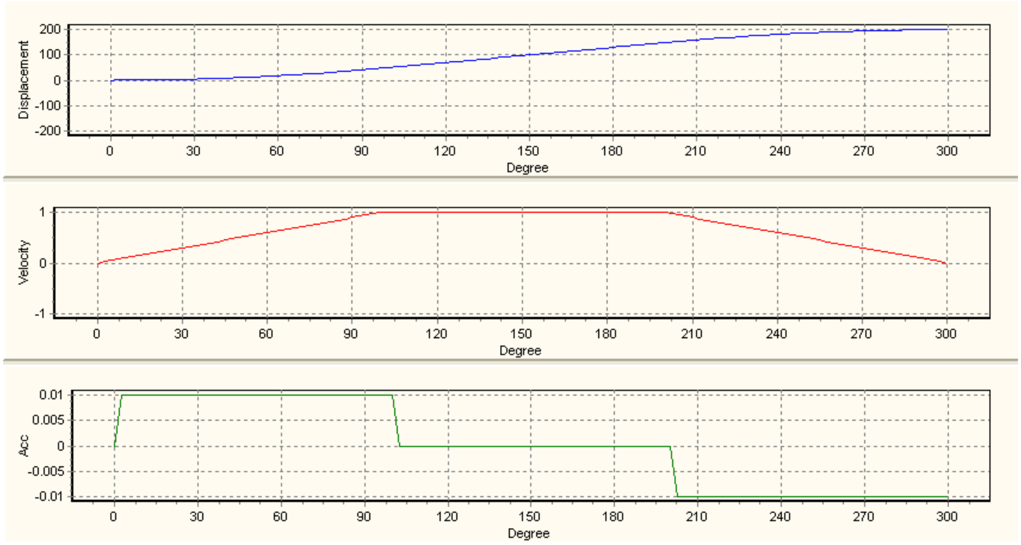
2. Input the velocity data in the displacement table.
   Position 1: 100 degrees
   Position 2: 200 degrees
   Position 3: 300 degrees



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.

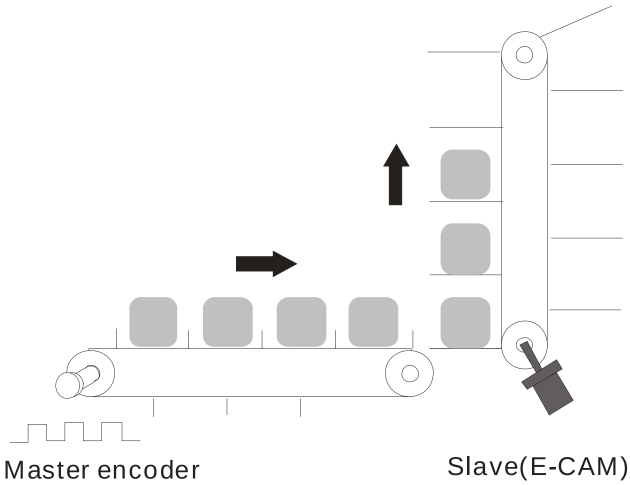


According to the above electronic cam chart, users can calculate the pulses per round of Slave (Slavemax=200) by the square measure of velocity data.

(1*100/2+1*(200-100)+1*(300-200)/2)=200

Pulses per round of Master equal to the cut length. Users can apply input /output magnification setting at 9.2.1.4 in this chapter to obtain the proper set value.

The below diagram is another application on synchronizing conveyers. The electronic cam operation of the example is similar to flying shear. 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 shear, the vertical conveyor can not keep smooth moving speed due to the acceleration and deceleration process of flying shear.

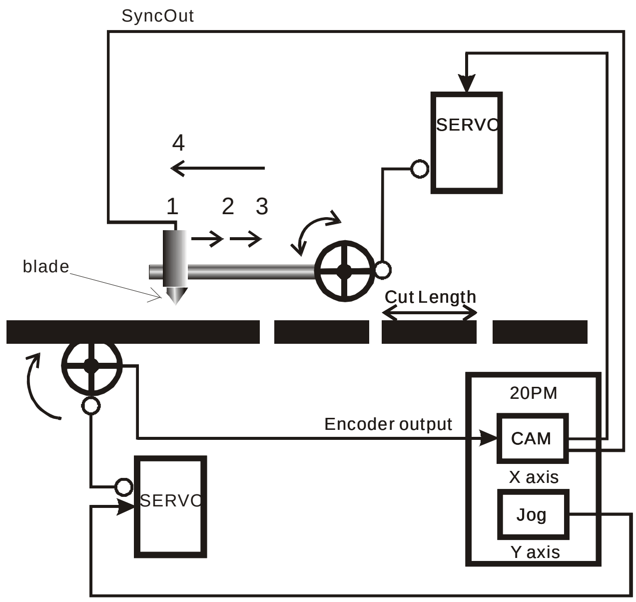


9.6.3 Application of a Flying Shear

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 (electronic 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.

9.6.3.1 Operation of a Flying Shear

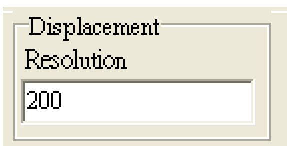
The operation and simple wiring of flying shear is illustrated as below. Positions 1, 2, 3 and 4 respectively indicate starting position, sync-start position, sync-end position and ready position. When Master (Y axis) executes, Slave (X axis) accelerates from position 1 to position 2, reaching the synchronizing speed. The sync speed is maintained from position 2 to position 3. After position3, Slave decelerates in reverse direction and returns to position 4 (Same as position 1). The cycle repeats for continuous flying shear operation.



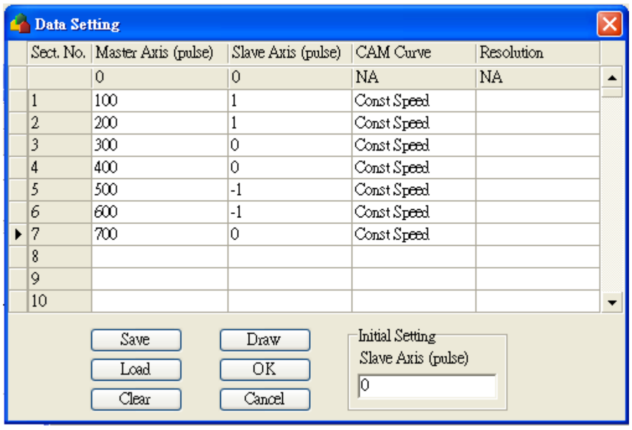
9.6.3.2 Creating an Electronic Cam Curve

In electronic cam design process, PMSoft creates electronic cam data by operating displacement data to generate the velocity data and the acc data. However, in this case users need to control the velocity between Slave and Master. To obtain the velocity, users 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.

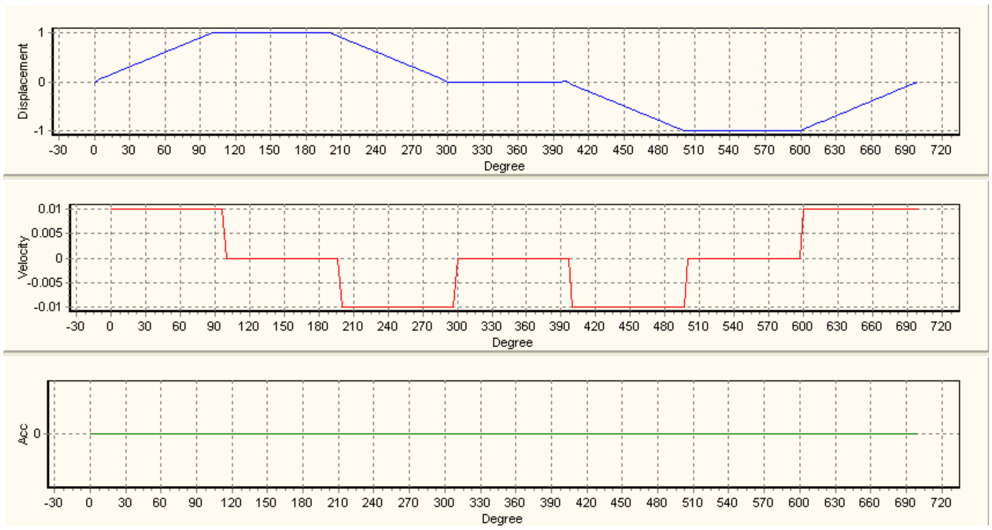


2. Input the velocity data in the displacement table.
   In the figure in section 9.6.3.1:
   Position 2: 100 degrees
   Position 3: 200 degrees
   Position 4: 700 degrees.

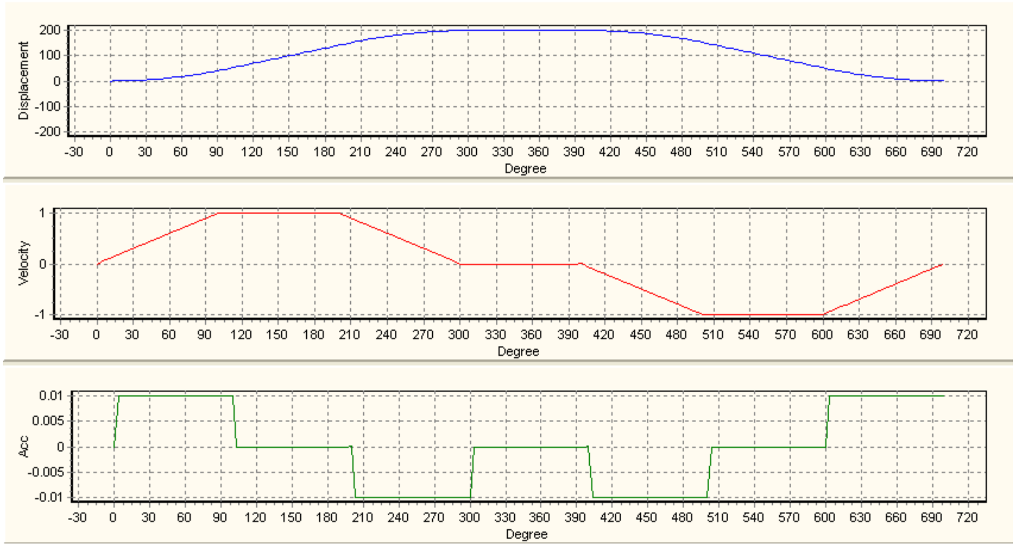


Sect. No.	Master Axis (pulse)	Slave Axis (pulse)	CAM Curve	Resolution
	0	0	NA	NA
1	100	1	Const Speed	
2	200	1	Const Speed	
3	300	0	Const Speed	
4	400	0	Const Speed	
5	500	-1	Const Speed	
6	600	-1	Const Speed	
7	700	0	Const Speed	
8				
9				
10				

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 electronic cam chart, users can calculate the pulses per round of Slave (Slavemax=200) by the square measure of velocity data.

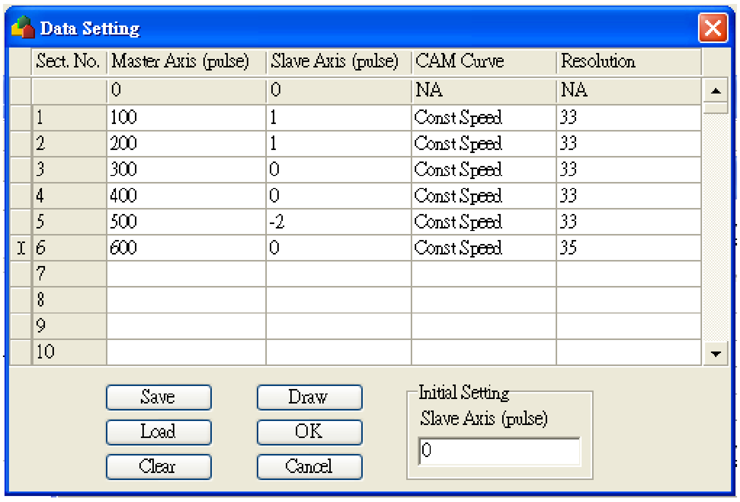
Positive square measure:

(1*100/2+1*(200-100)+1*(300-200)/2)=200

Negative square measure:

((-1)*100/2+(-1)*(200-100)+(-1)*(300-200)/2)=-200

In addition, for pulses per round of Master users can apply input /output magnification setting to obtain the proper set value. Different from the setting of rotary cut, lower/higher bound of electronic cam synchronized output section should be specified. Also, to save the time of operation cycle users can set a higher speed for the process when cutter returns to position 4 in the figure in section 9.6.3.1. See below example.



Data Setting

Sect. No.	Master Axis (pulse)	Slave Axis (pulse)	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
6	600	0	Const Speed	35
7				
8				
9				
10				

Save Load Clear Draw OK Cancel

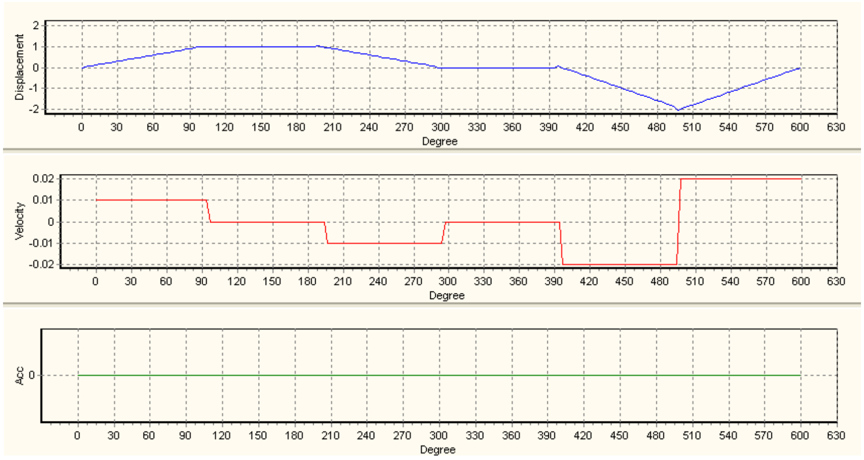
Initial Setting
Slave Axis (pulse)
0

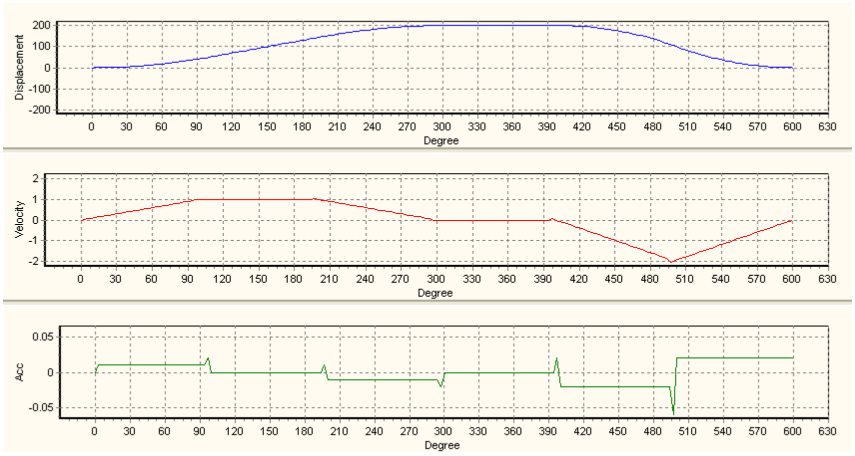
In this example, modify the angle of position 4 in the figure in section 9.6.3.1 from 700 to 600 degrees and set the stroke at section 5 as -2, indicating doubling the reverse running speed. In this case, 100 pulses (700-600) can be reduced at operation speed 100 kHz, i.e. approximately 1ms can be saved 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).

Positive square measure:

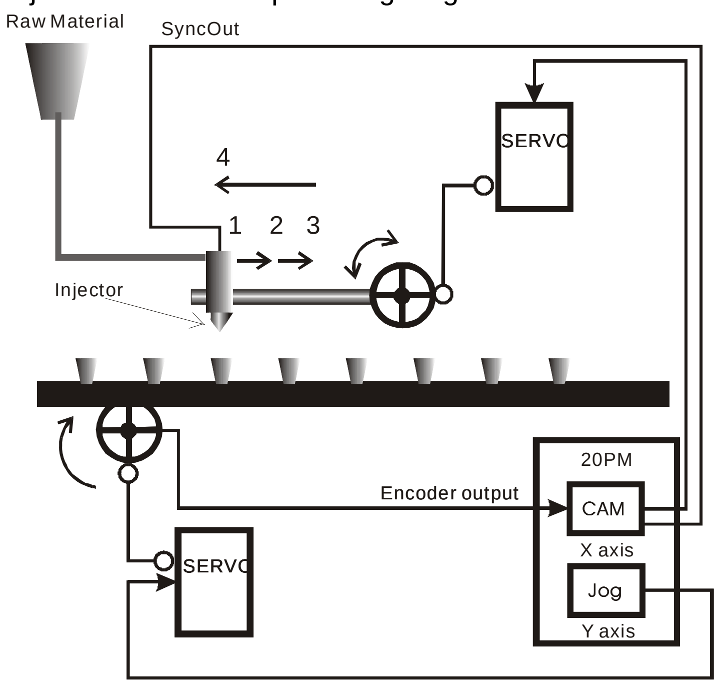
(1*100/2+1*(200-100)+1*(300-200)/2)=200

Negative square measure:
(-2*(600-400)/2)=-200



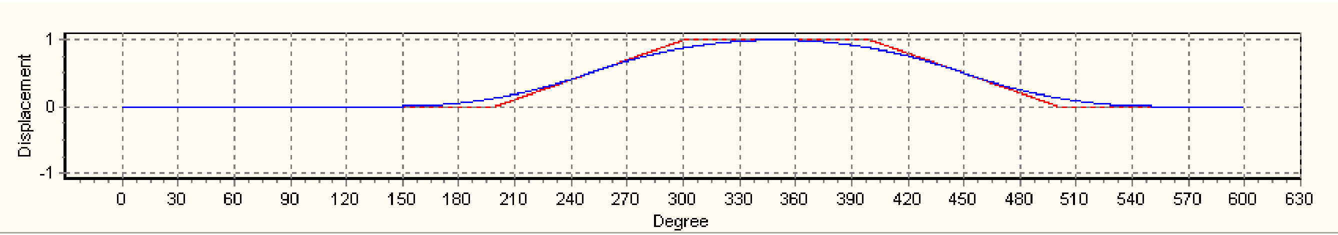


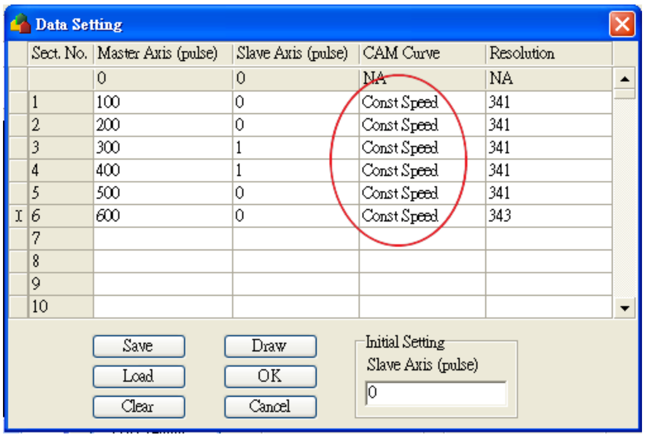
The below diagram is another flying shear application: liquid filling machine. The electronic cam operation of the example is similar to the above case of flying shear. The only difference is that the cutter is replaced by the injector. See the simple wiring diagram as below.



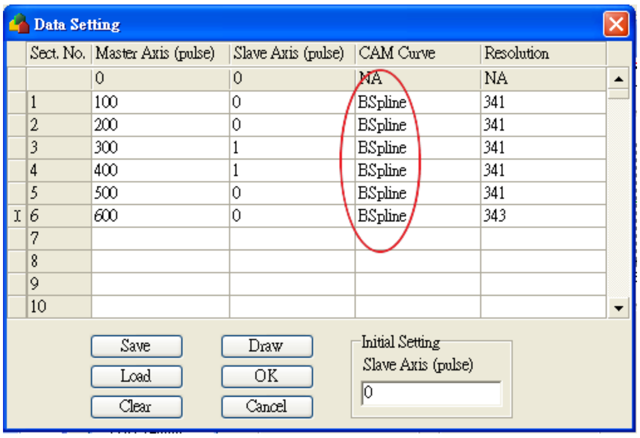
Filling machine

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, users 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 cure and acc curve. The results of the same electronic cam data with 2 different cam curves are as below. Red curve indicates Const Speed and blue curve indicates B-Spline.



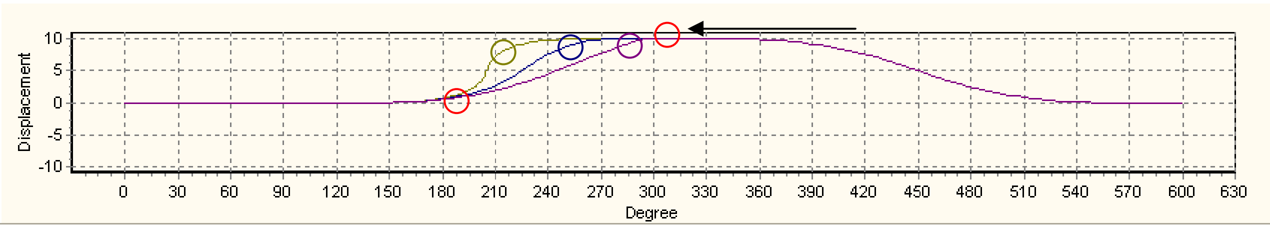


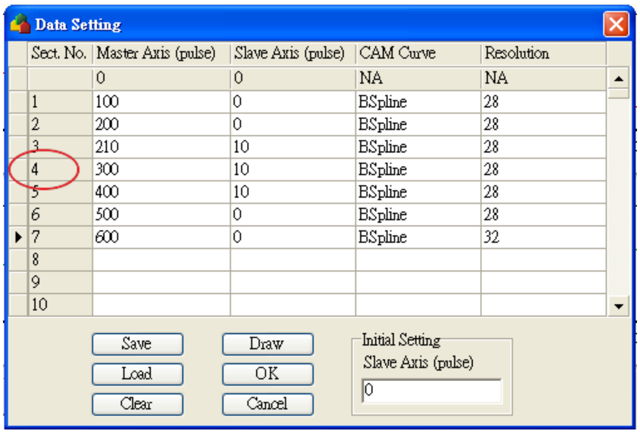
Sect. No.	Master Axis (pulse)	Slave Axis (pulse)	CAM Curve	Resolution
	0	0	NA	NA
1	100	0	Const Speed	341
2	200	0	Const Speed	341
3	300	1	Const Speed	341
4	400	1	Const Speed	341
5	500	0	Const Speed	341
6	600	0	Const Speed	343
7				
8				
9				
10				



Sect. No.	Master Axis (pulse)	Slave Axis (pulse)	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				

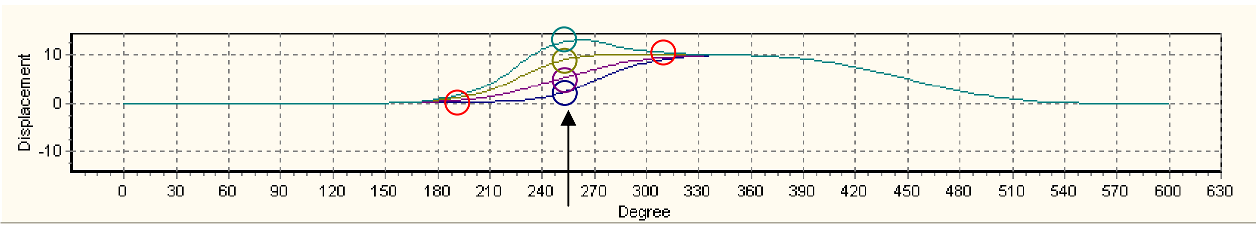
The property of B-Spline is to modify only partial curve rather than the whole curve, i.e. if additional points between 2 Master positions are inserted, only the curve in the section is modified, and the curve besides the section remains in the same shape. Here is the example of inserting 210, 250, and 290 between Master position 200 and 300. Users can observe the curve variation of each point.

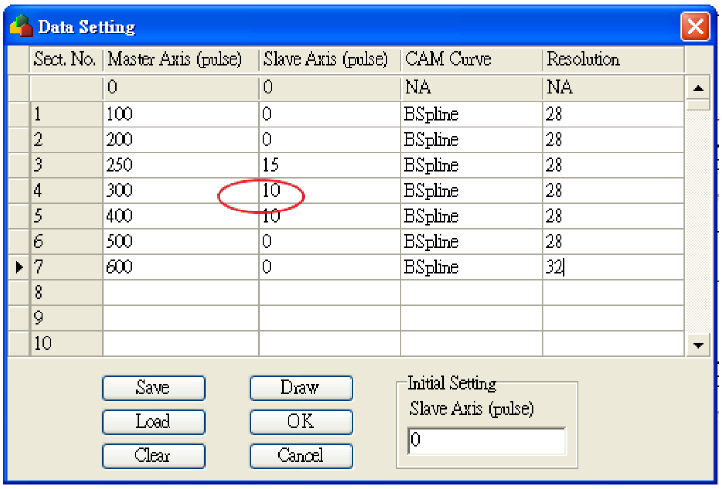




Sect. No.	Master Axis (pulse)	Slave Axis (pulse)	CAM Curve	Resolution
	0	0	NA	NA
1	100	0	BSpline	28
2	200	0	BSpline	28
3	210	10	BSpline	28
4	300	10	BSpline	28
5	400	10	BSpline	28
6	500	0	BSpline	28
7	600	0	BSpline	32
8				
9				
10				

The above example illustrates the horizontal variation of inserting different Master positions. Here is another example explaining the vertical variation of inserting different Slave positions1, 5, 10 and 15.




Data Setting
Sect. No.
Master Axis (pulse)
Slave Axis (pulse)
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
Save
Load
Clear
Draw
OK
Cancel
Initial Setting
Slave Axis (pulse)
0

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: When the source is preset, the users can select one of the 3 sources (HMI, memory card or PC) by HMI and perform the control requirement.

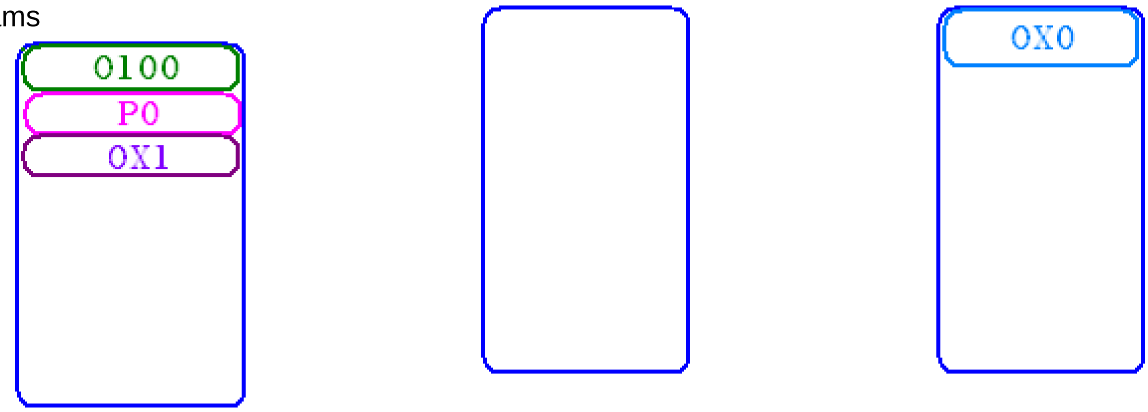
1. HMI: Editing and storing G-codes on HMI
2. Memory card: Converting the file on CAM software into G-codes, storing the G-codes in a memory card and inserting the card into the slot on a CNC machine,
3. PC (NB): Directly downloading the G-codes into a CNC machine from a PC (NB).

10.1 Downloading a Program with PEP

When programs are downloaded to DVP20PM00M, the DVP20PM00M 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 DVP20PM00M 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.

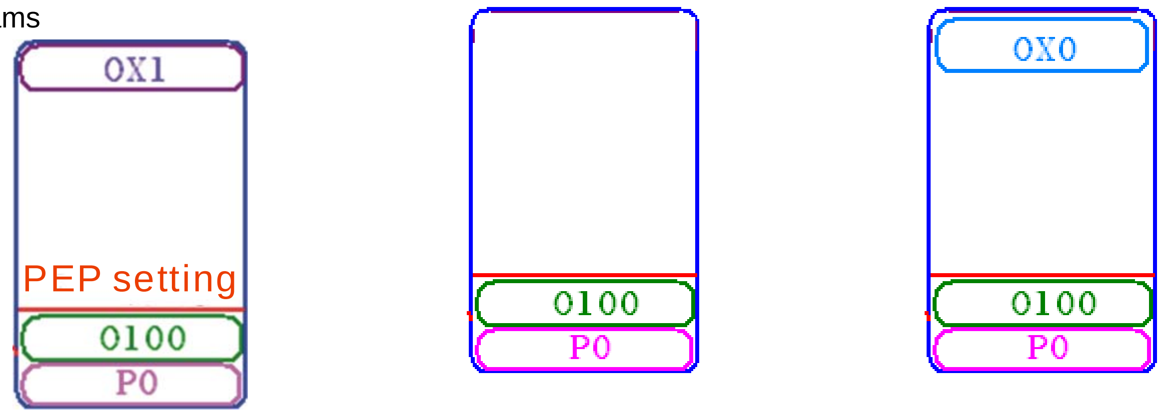
(A) No PEP protection

Original DVP20PM00M programs | Clear the programs | Download new Ox



(B) With PEP protection

Original DVP20PM00M programs | Clear the programs | Download new Ox



With PEP protection as above, constant programs O100 and P are maintained in DVP20PM00M, and Ox0 (G-codes) which needs to be constantly updated is downloaded in the general program area without PEP protection.

10.2 Methods of Downloading G-codes

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 HMI and convert G-codes through D registers.

10.2.1 PMGDL Software

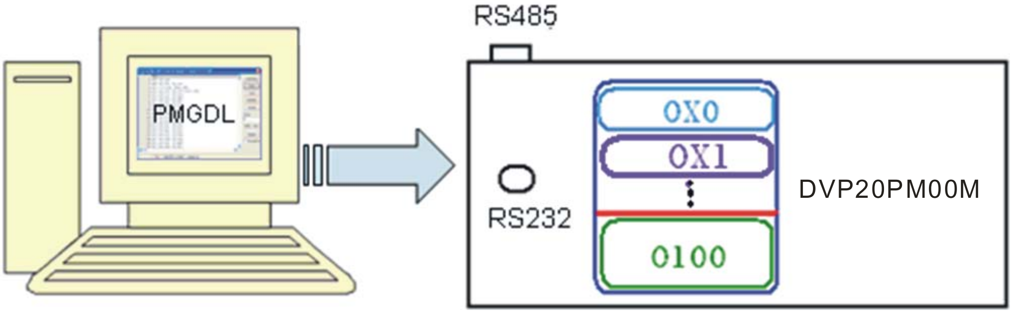
G-codes can be downloaded through PMGDL by two modes.

1. Common mode: Download G-codes to a DVP-20PM series motion controller then execute Oxn motion subroutines.
2. DNC mode: Execute Oxn subroutines while file transmission is processing.

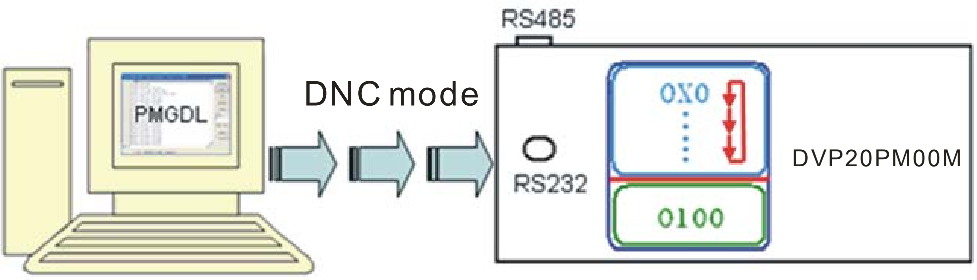
The application of Common mode and DNC mode is mainly decided according to the G-code file size. In Common mode, G-codes will be completely downloaded to PM before executing Oxn motion subroutine. In DNC mode, generally applied for huge G-code files, transmission proceeds while motion subroutine executes. In diagram (A), PMGDL in Common mode downloads all G-codes into Ox0, Ox1 in DVP20PM00M. In diagram (B), PMGDL in DNC mode transmits G-codes while motion subroutines in DVP20PM00M are 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



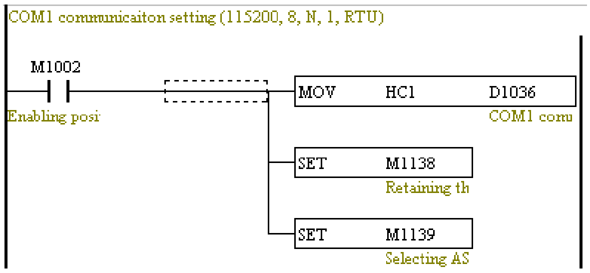
(B) DNC mode G-code download



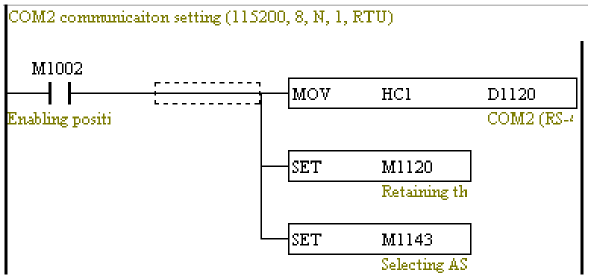
■ Common mode

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 setting up COM1

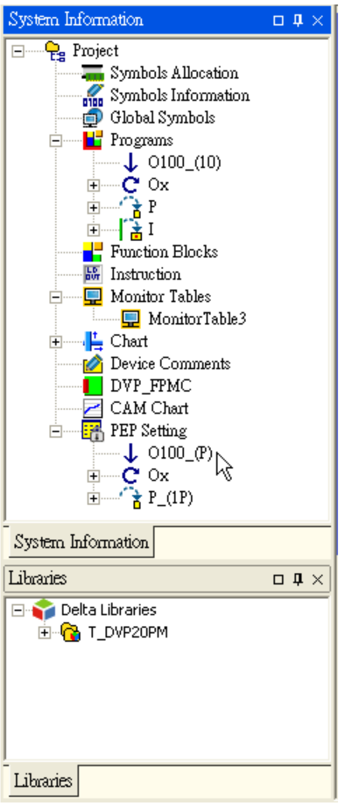


(B) Program for setting up COM2

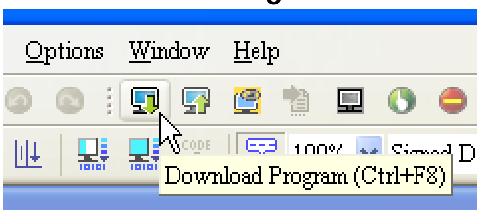


Step 2: Set up PEP settings in PMSoft to enable protection on O100 or other constant programs and download the program to DVP20PM00M as below.

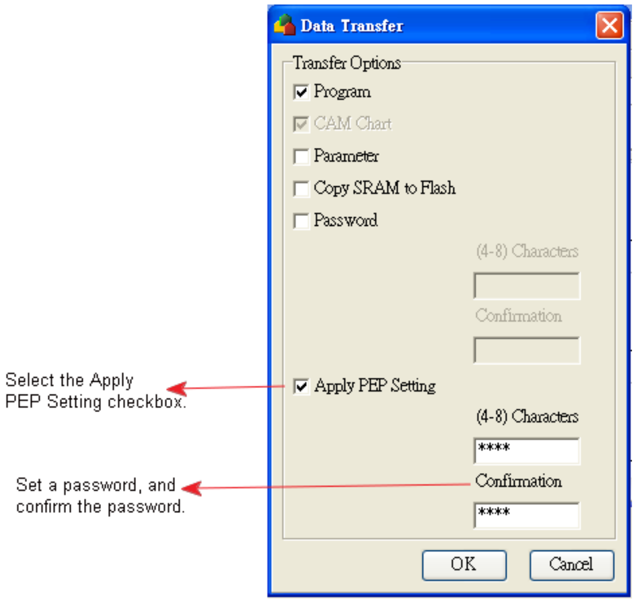
(A) Set up PEP settings on programs.



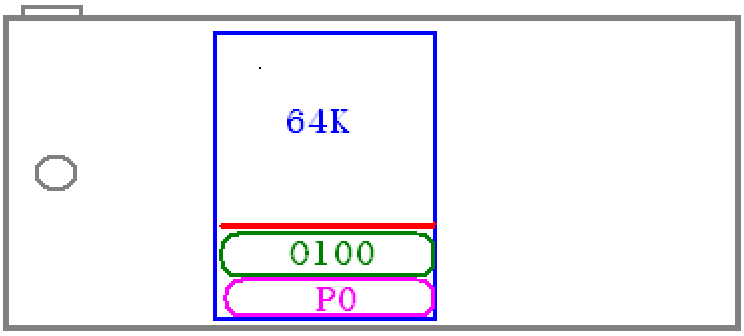
(B) Click **Download Program**.



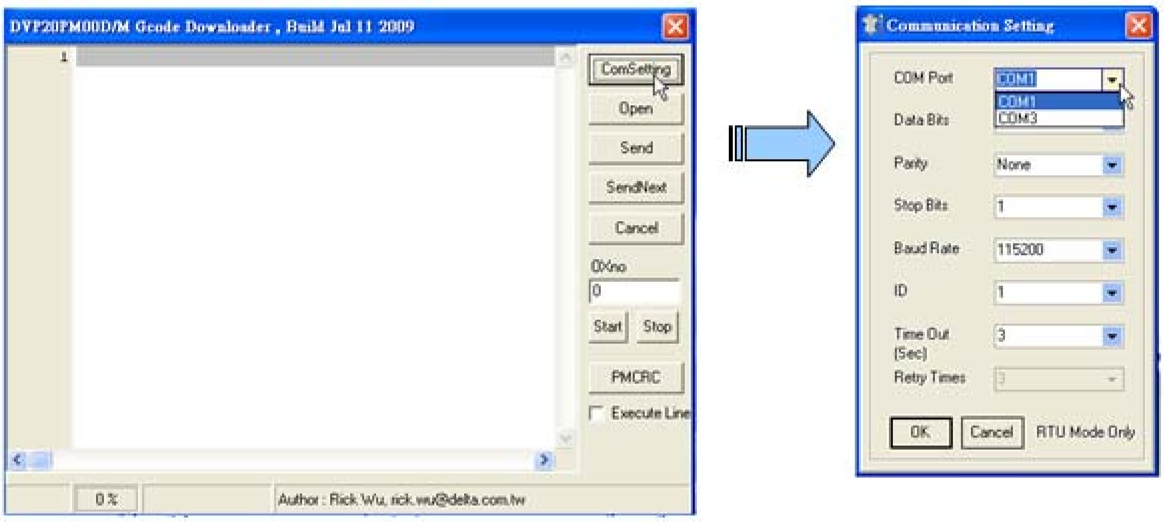
(C) Download settings.



(D) Result

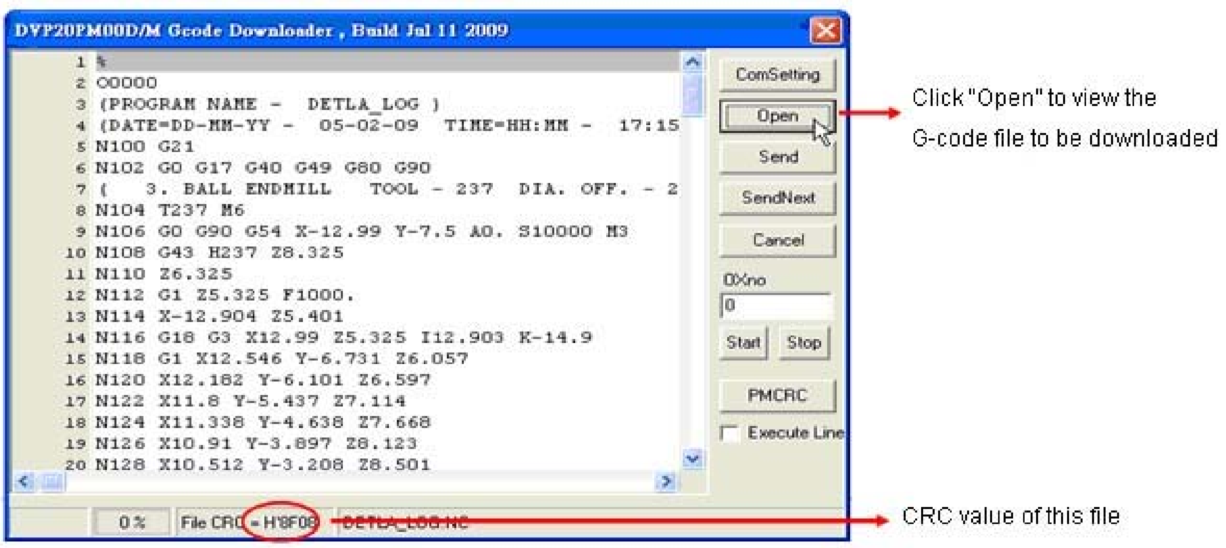


Step 3: Execute PMGDL software and set up COM Port (between PC and DVP20PM00M COM port) as below. Communication format 115200, 8, N, 1 should not be changed.

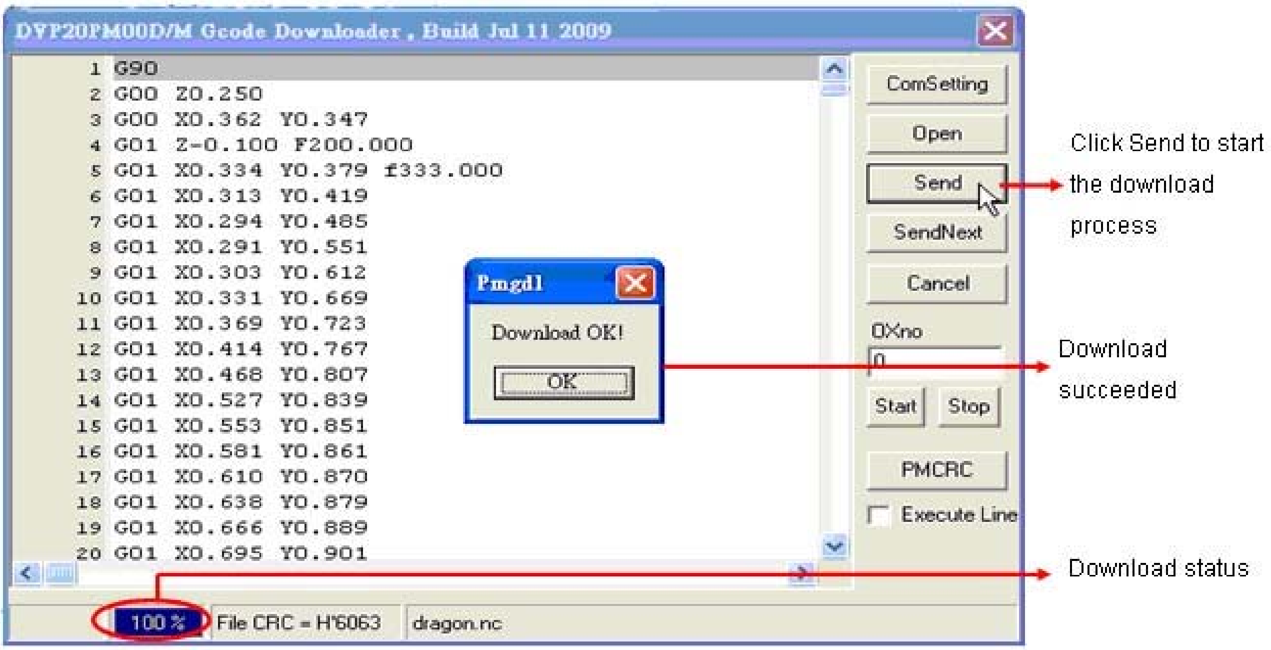


Step 4: Click "Open" to view the G-code file to be downloaded. In the bottom-left corner, File CRC will be displayed. Click "Send" to download the G-code file.

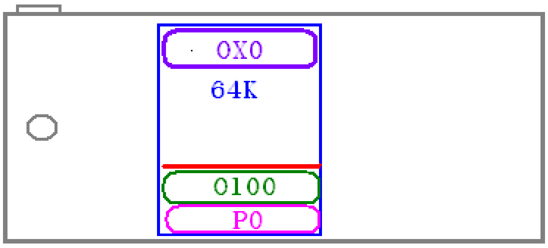
(A) Click **Open** to view the G-code file to be downloaded.



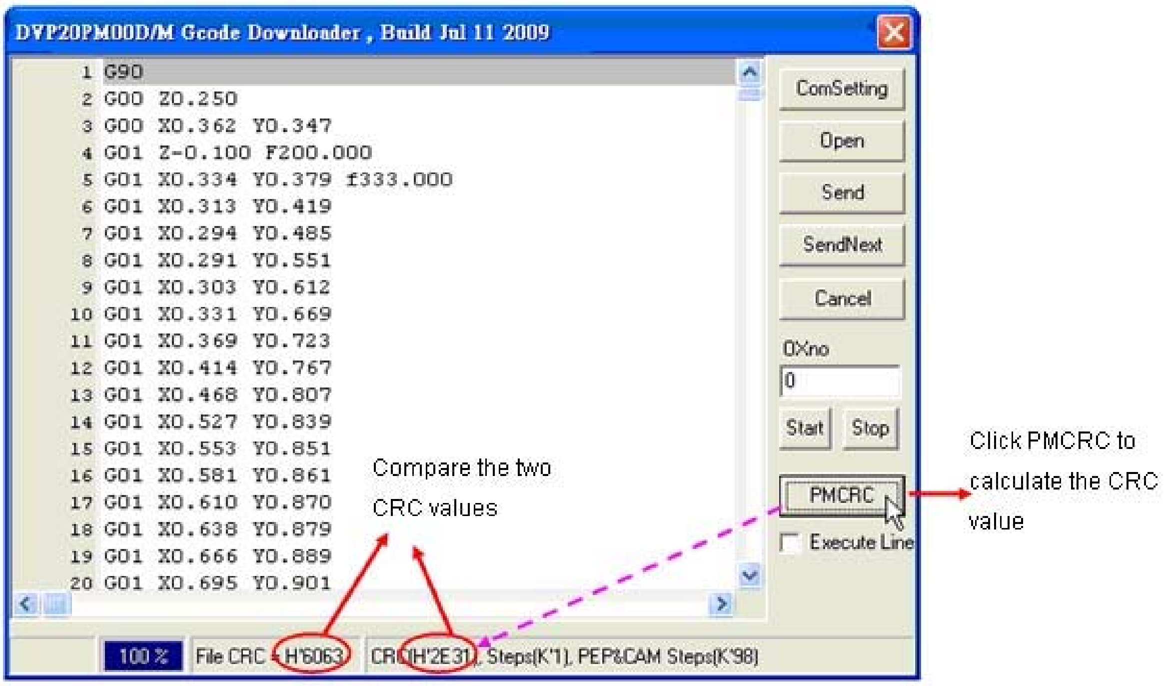
(B) Click **Send** to download the G-code file.



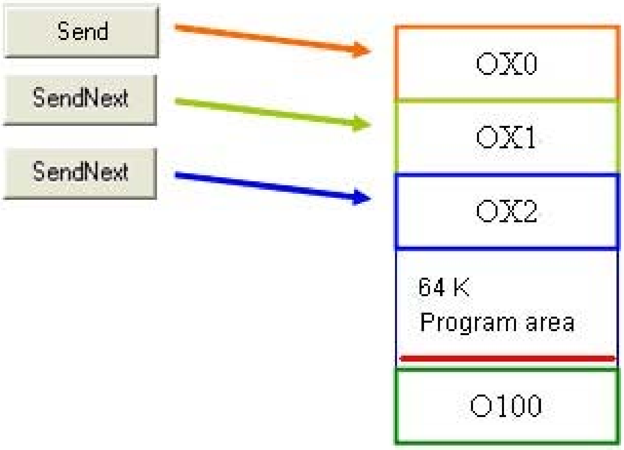
(C) Download completed.



Step 5: The users 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.



Step 6: PMGDL can download multiple G-code files to DVP20PM00M. 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.



P.S.: 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, a DVP-20PM series motion controller will automatically execute Ox0 when G-codes are downloaded to the DVP-20PM series motion controller.

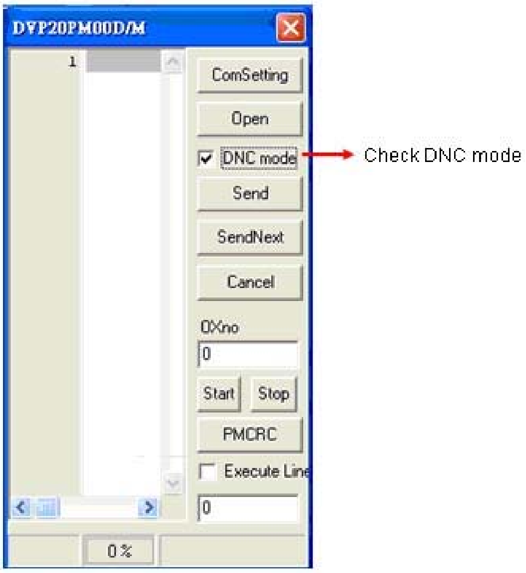
Step 1: The program downloaded to a DVP-20PM series motion controller should contain Ox0 motion subroutine, and G-codes do not need to be designed in Ox0. To create Ox0, click **OX** in the system information area in PMSoft. Double click at **0** and the **OX0** editing window will pop up automatically. After Ox0 is created, follow step 1 of Common mode to set up O100.

Step 2: Same as Step 2 of Common mode

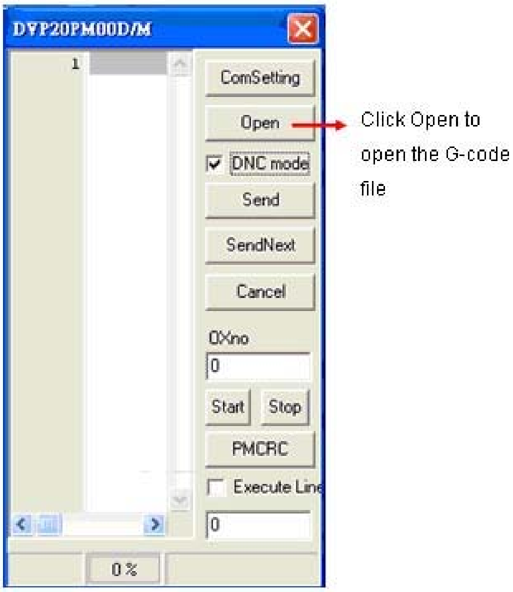
Step 3: same as Step 3 of Common mode

Step 4: (1) Select the **DNC mode** checkbox. (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.

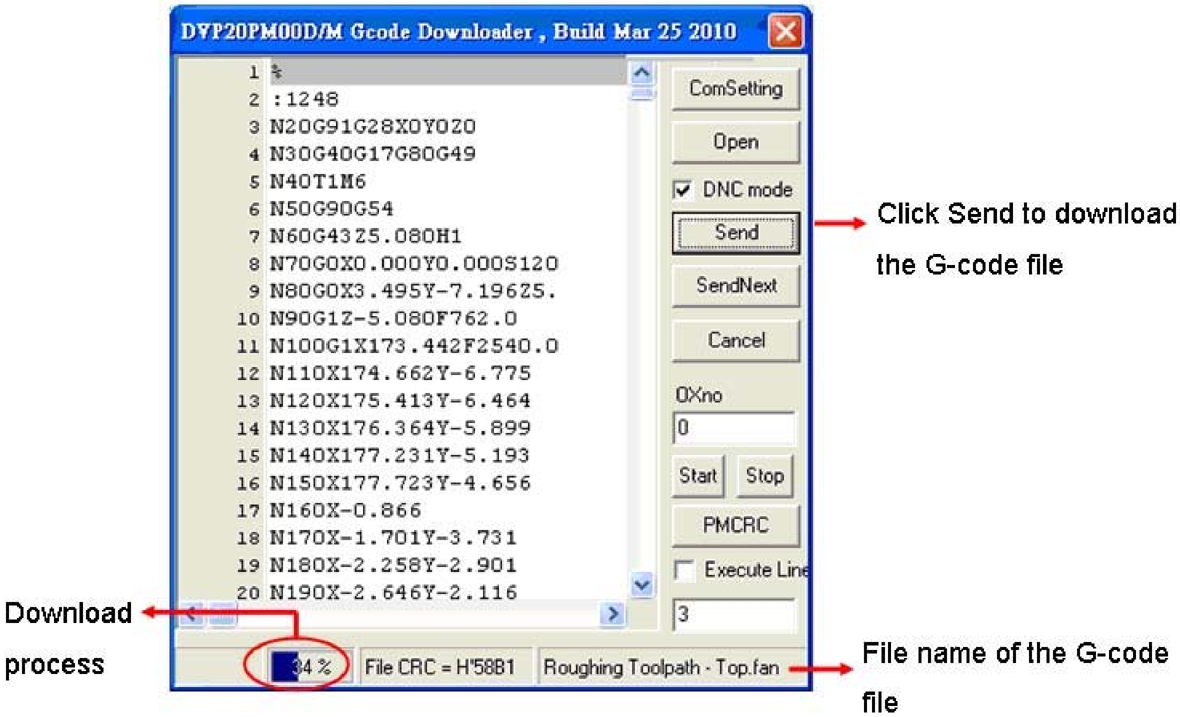
(A) Select the **DNC mode** checkbox.



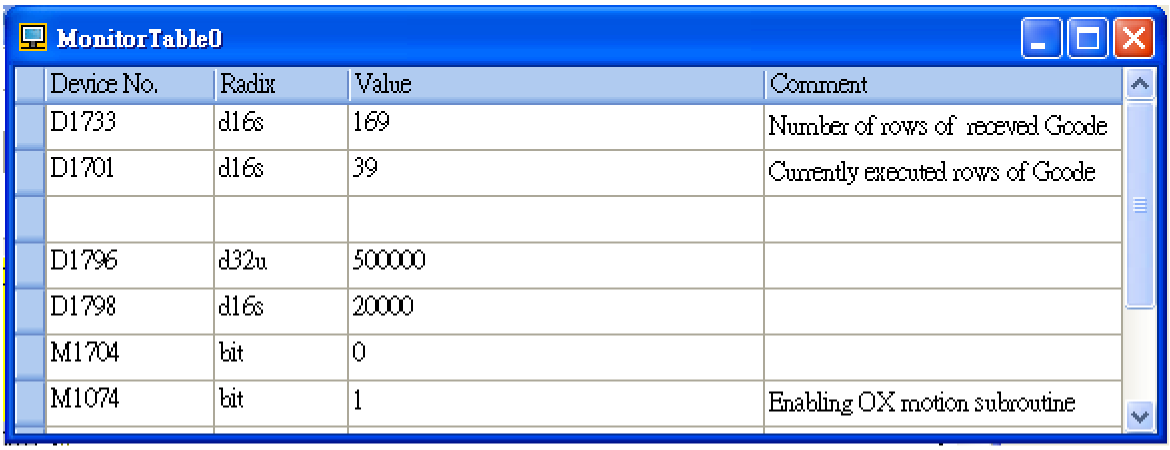
(B) Click **Open** to view the G-code file to be downloaded.



(C) Click **Send** to download the G-code file.

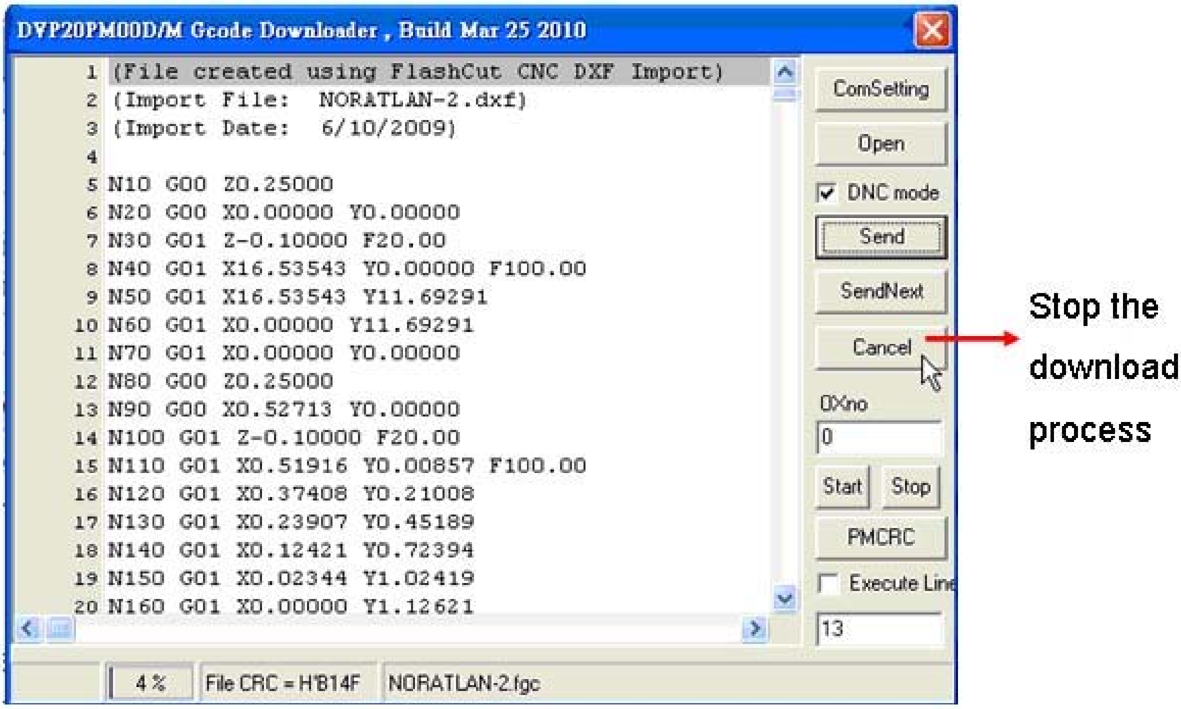


Step 5: On PMSoft, users can monitor D1733 for number of rows of received G-code on the DVP-20PM series motion controller used, and (D1702, D1701) for currently executed rows of G-code. Check if the content of the registers increases and Ox0 executes automatically as below.

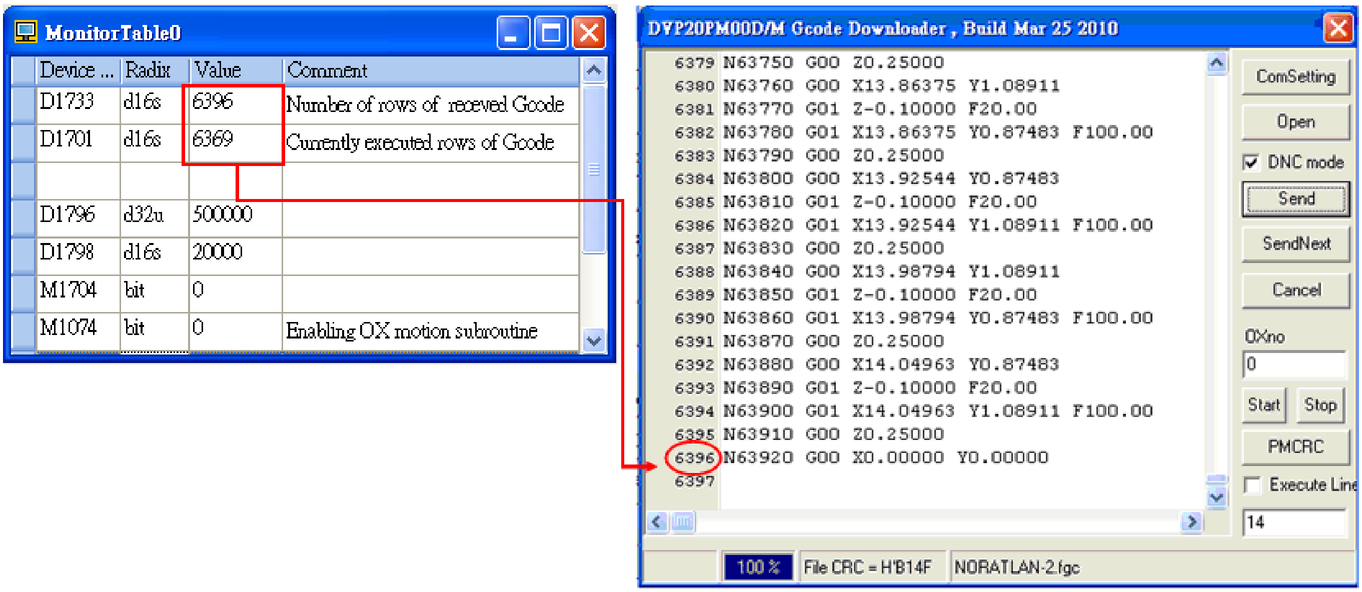

MonitorTable0

Device No.	Radix	Value	Comment
D1733	d16s	169	Number of rows of receved Gcode
D1701	d16s	39	Currently executed rows of Gcode
D1796	d32u	500000	
D1798	d16s	20000	
M1704	bit	0	
M1074	bit	1	Enabling OX motion subroutine

Step 6: If the users want to stop the operation in DNC mode, click **Cancel**. G-codes will stop when the executed rows equal to the received rows. If the users 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.



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 the users want to execute other G-codes, repeat the steps from step 4.

10.2.2 Converting the G-codes in Data Registers

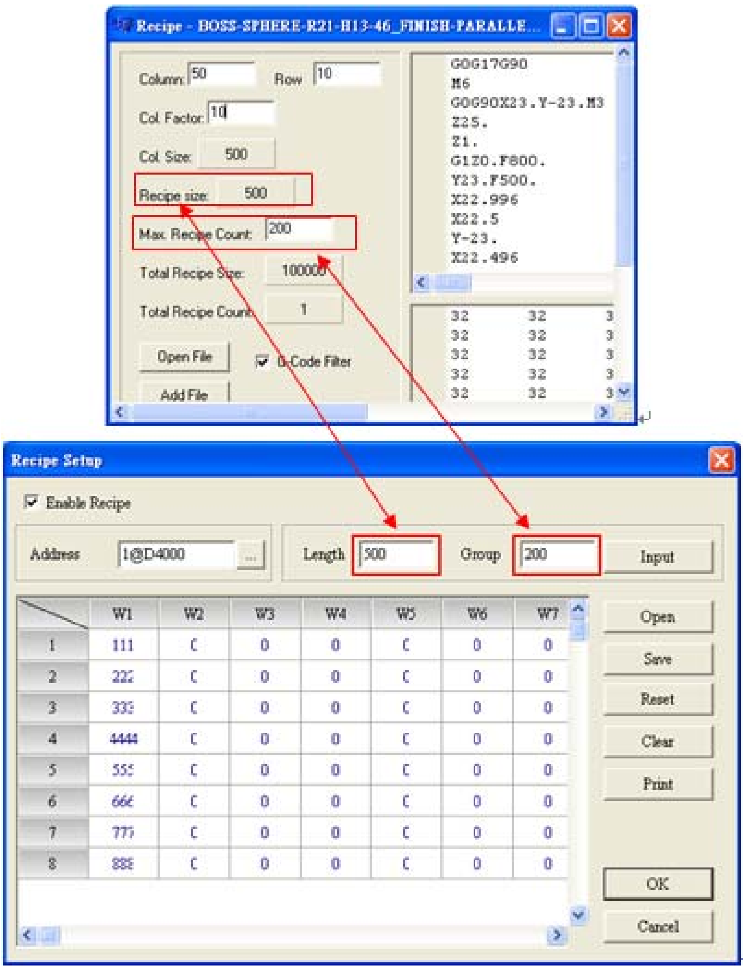
Users can also download G-codes by HMIs other than B-type. Set up the recipe on HMI and design the user program for receiving G-codes on a DVP-20PM series motion controller. When both sides are ready, the G-codes can be downloaded through HMI directly.

The G-code download program designed in a DVP-20PM series motion controller 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 a PLC according to the recipe set up in a 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, the error code 0xffff and the error row number will be displayed in designated registers.

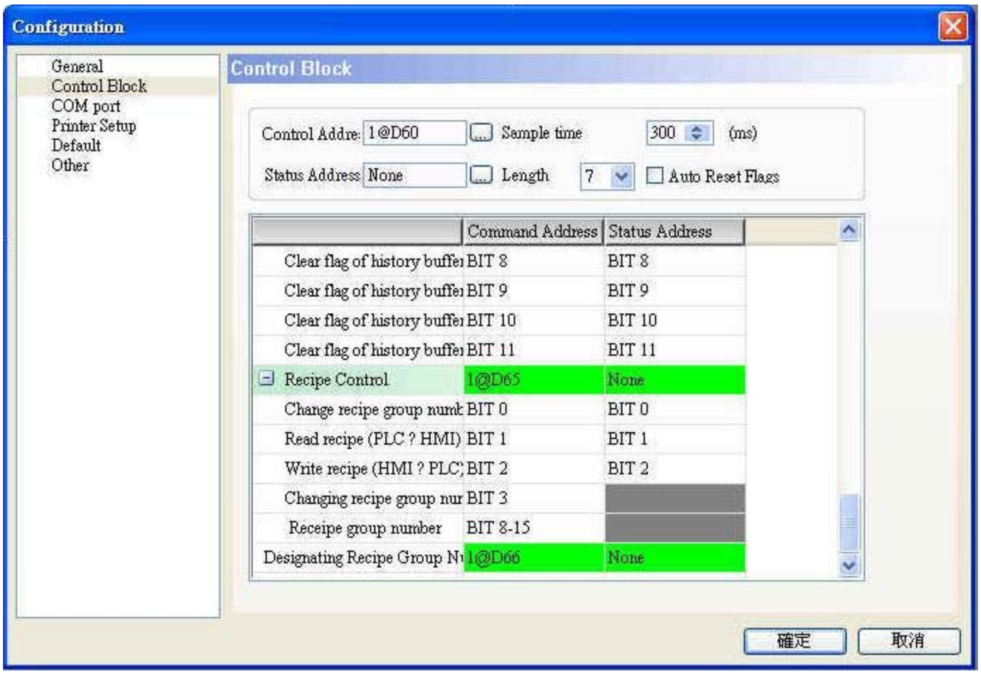
After one group are received, the DVP-20PM series motion controller used sends another value to change the group number and repeats the above process until all groups of G-code file are received, which is indicated by 20H (K32) in D3002 and D3003. When all G-codes are downloaded, the conversion results 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 the PC will not be cleared after uploading the program from the DVP-20PM series motion controller.

1. Setting up a recipe on HMI

   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 the DVP-20PM series motion controller. As the diagram below, the DVP-20PM series motion controller 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.



2. Designing a program

The special module K255 is used to converting G-codes. Users can convert G-code and monitor the result gotten by means of reading K255 and writing data to K255. Registers related to the conversion of G-codes are described below.

CR number	Function	Data type	Length
0	Initializing the conversion of G-codes	Word	1
1	Result of converting G-codes	Word	1
2	Carrying out the conversion of G-codes	Word	n

- Descriptions of control registers

CR#0: Initializing the conversion of G-codes

[Description]

Before a DVP-20PM series motion controller received G-codes form a human-machine interface or another device, they can use the control register to initialize the conversion of codes.

CR#1: Result of converting G-codes

[Description]

If no G-code needs to be sent by a human-machine interface, data will be written into the control register, and the conversion of G-codes will end. After To K255 k1 k0 k1 is used, 0xffff will appear in position 1 and the number of the line where an error occurs will appear in position 2 in the control register if the error occurs. After To K255 k1 k0 k1 is used, the values starting from the value in position 2 in CR#2 to the last G-code will be ASCII codes if the conversion of G-codes is correct.

CR#2: Carrying out the conversion of G-codes

[Description]

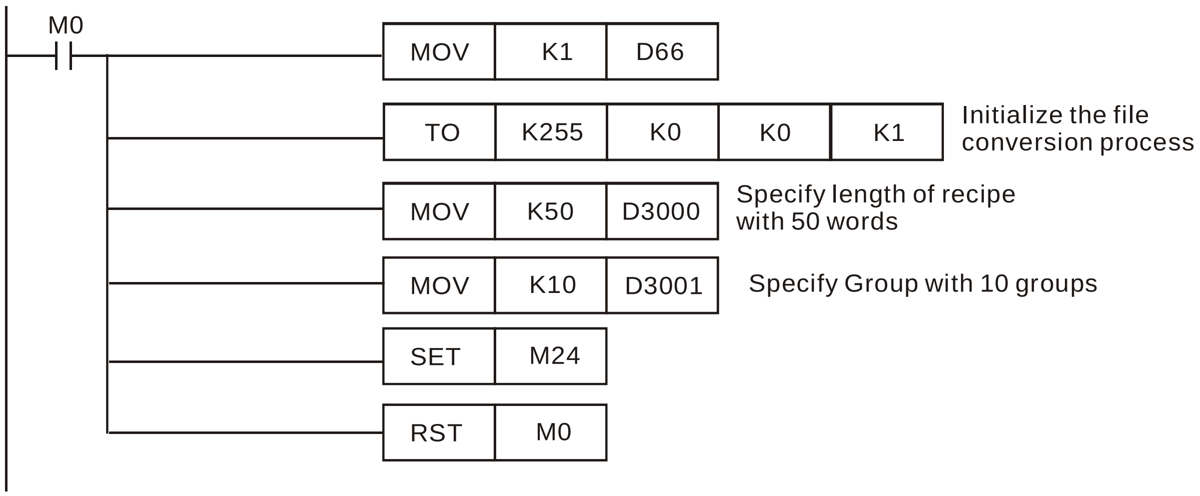
The control register is used to specify the position in which the data gotten from conversion is stored. It is described below.

Position	Data	Data type	Description
Length 1	Number of data registers for a line of G-codes	Integer	
Length 2	Number of rows of G-codes	Integer	
Length 3 ~length n	Data gotten from conversion	Integer	■ Length of the data gotten from conversion=(Number of G-codes in a line×Number of rows)+1 ■ After G-codes are converted, 0xffff will appear in position 1 and the number of the line where an error occurs will appear in position 2 if the error occurs. ■ After G-codes are converted, the values starting from the value in position 3 to the value in position n are data gotten from the conversion if the conversion of the G-codes is correct,

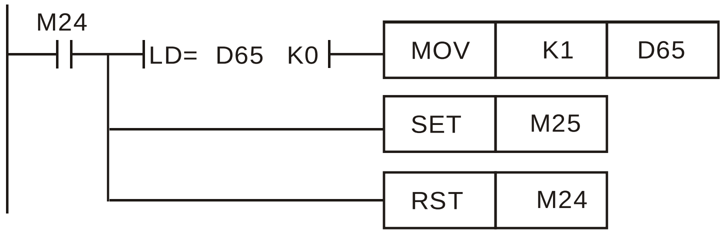
- Instructions for receiving G-codes

  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, users can check Ox0 motion subroutine for the received G-codes.

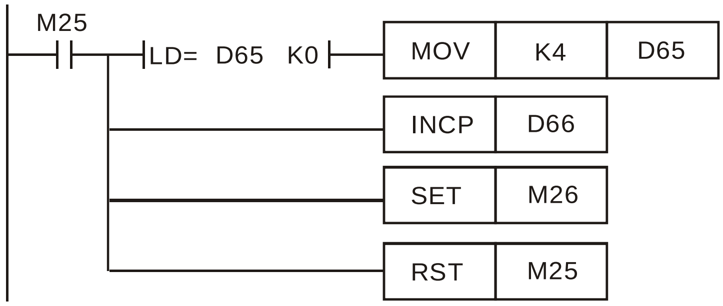
  Step1: Move K1 to D66 to designate the recipe group number. Initialize the file conversion process and setting the format of recipe according to the recipe setup in Screen Editor.



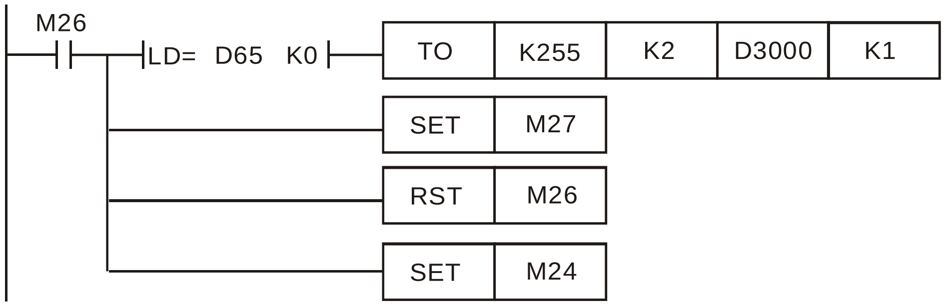
Step2: HMI reads the value in D65 as the command to control the recipe. D65 = 1 indicates changing the recipe group number.



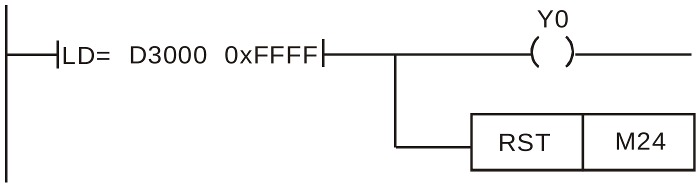
Step3: When recipe group number is changed, the value in D65 becomes 0. After this, write in the recipe command to download next group and increase the recipe group number in D66.



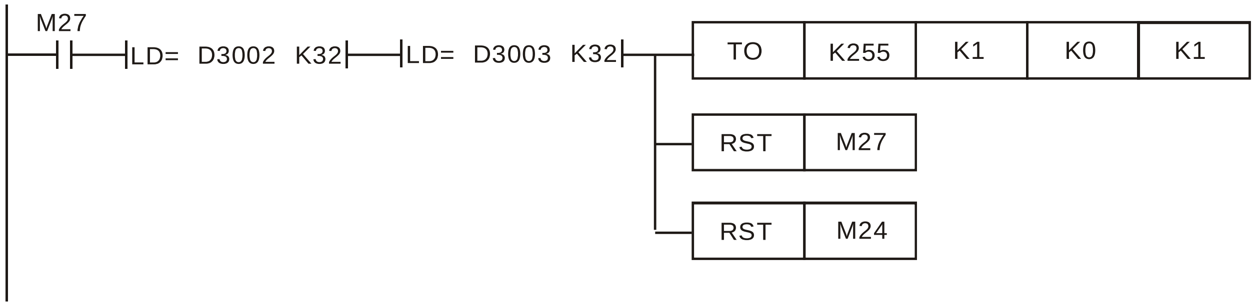
Step4: The value in D65 will be cleared as 0 every time when modified. When one group of recipe is received, conduct file conversion to convert the received data into ASCII codes and store the converted data inD3002~D3502. When the above process is completed, repeat the step to download the next group.



Step5: If errors occur in file conversion, D3000 will store the error code 0xffff. Users can stop the next download by monitoring D3000.



Step6: 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.



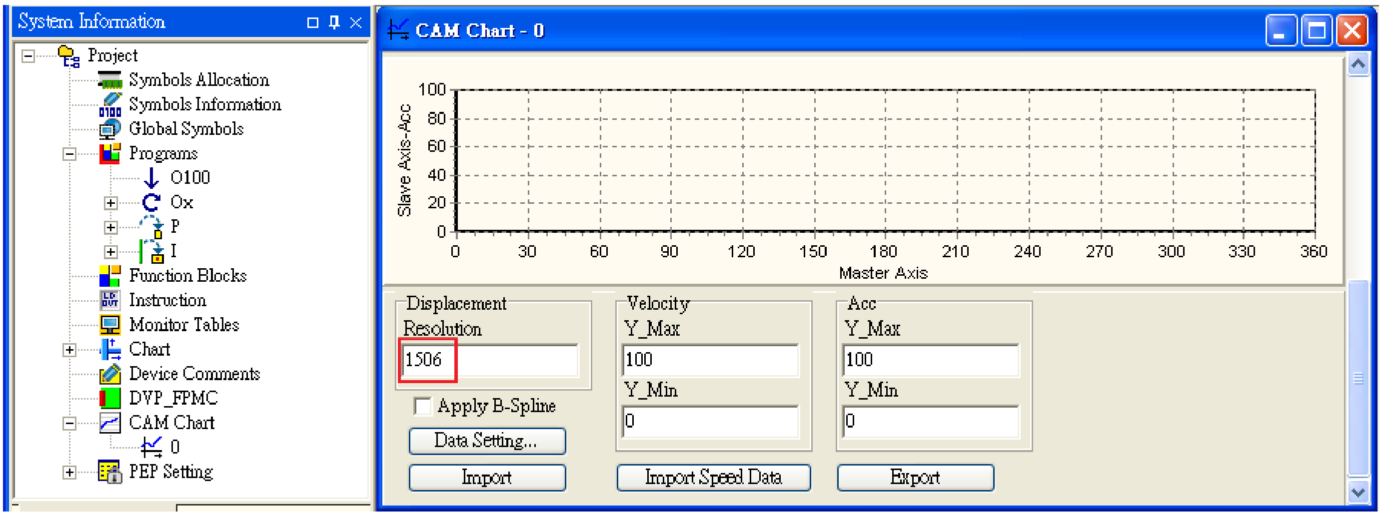
10.3 Applying G-codes to a Dispenser

For G-code application in automatic dispensers, DVP20PM00M provides a specific function which performs smooth movement with equal speed, avoiding pauses which lead to uneven dispensing results during operation.

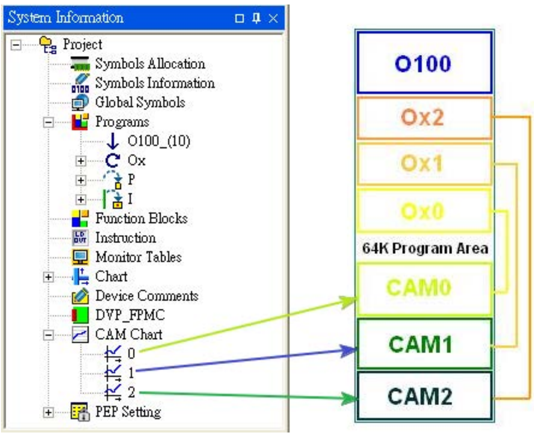
The function utilizes CAM charts for storing the G-codes/M-codes and applies E-CAM operation to execute the data in CAM charts. A DVP-20PM series motion controller is designed with three CAM charts storing 3 sets of G-code/M-code programs including Ox0, Ox1 and Ox2. Max 2048 points are allowed for each 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.

There are four operation steps. They are described below.

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. Creating 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.



3. Loading data into CAM chart: Load the G-codes/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, and then Oxn should be called.

(A) Setting up work modes of 3 axes

	X-axis	Y-axis	Z-axis
Work mode (bit 12)	D1847=H1000	D1927=H1000	D2007=H1000

(B) Calling Oxn: Call Oxn to load the G-codes/M-codes in corresponding CAM charts, without executing Oxn. When the G-codes/M-codes are loaded, M1792 will be rising-edge triggered. The below diagrams are examples demonstrating how to call Oxn subroutines.

a. Calling the Ox0 subroutine

b. Calling the Ox1 subroutine

c. Calling the Ox2 subroutine

P.S.: Do not set up M1036 (continuous interpolation) before this step, otherwise the loaded data will be G-codes/M-codes with continuous interpolation.

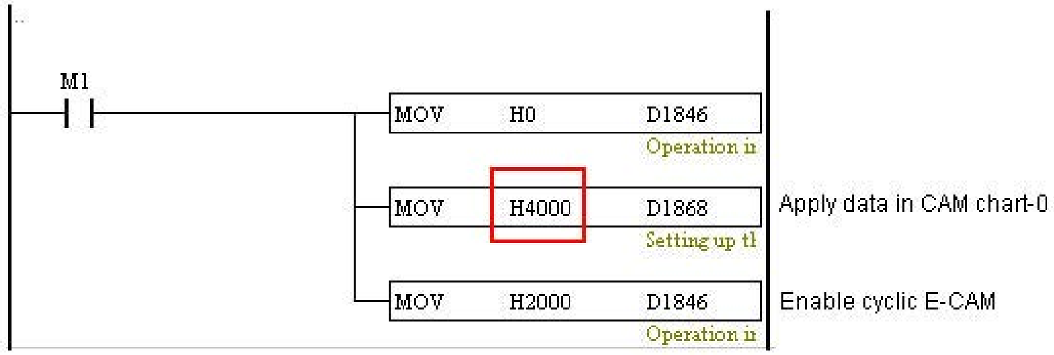
4. Enabling the dispensing operation: Execute the G-codes/M-codes in CAM chart. Work modes of 3 axes should be set up first and select the CAM chart to be executed. Set D1846=H2000 to enable cyclic E-CAM. When the operation in CAM chart is completed, M1792 will be rising-edge triggered.

(A) Setting up work modes of 3 axes: If the work modes are set up as H1000 already, skip this step.

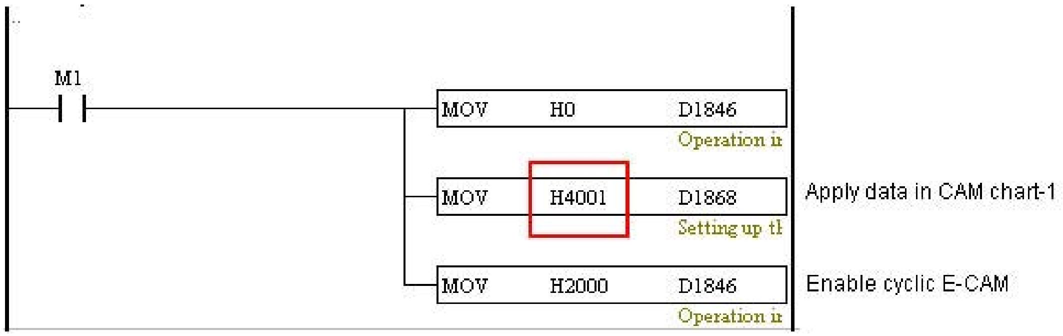
	X-axis	Y-axis	Z-axis
Work mode (bit 12)	D1847=H1000	D1927=H1000	D2007=H1000

(B) Enabling the dispensing operation by executing the Oxn subroutine

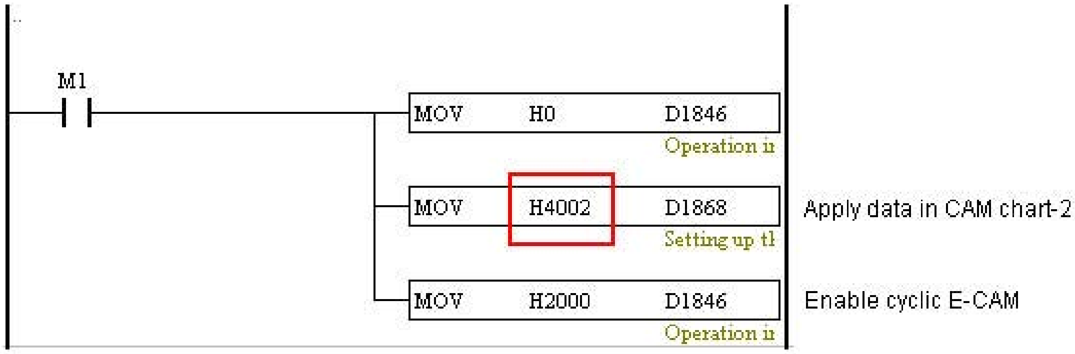
a. Calling the Ox0 subroutine



b. Calling the Ox1 subroutine

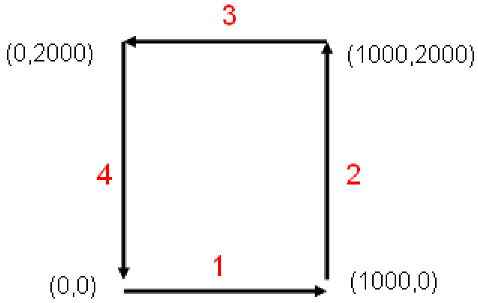


c. Calling the Ox2 subroutine

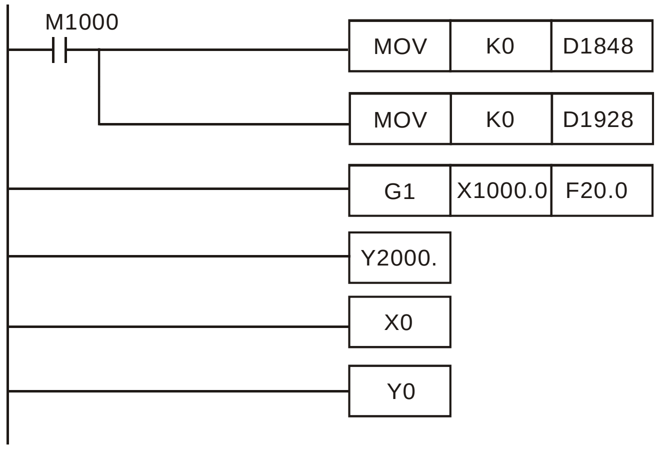


When the function is not used, make sure the work mode is set as 0 (D1847 = 0, D1927 = 0 and D2007 = 0), otherwise errors will occur if executing single speed positioning.

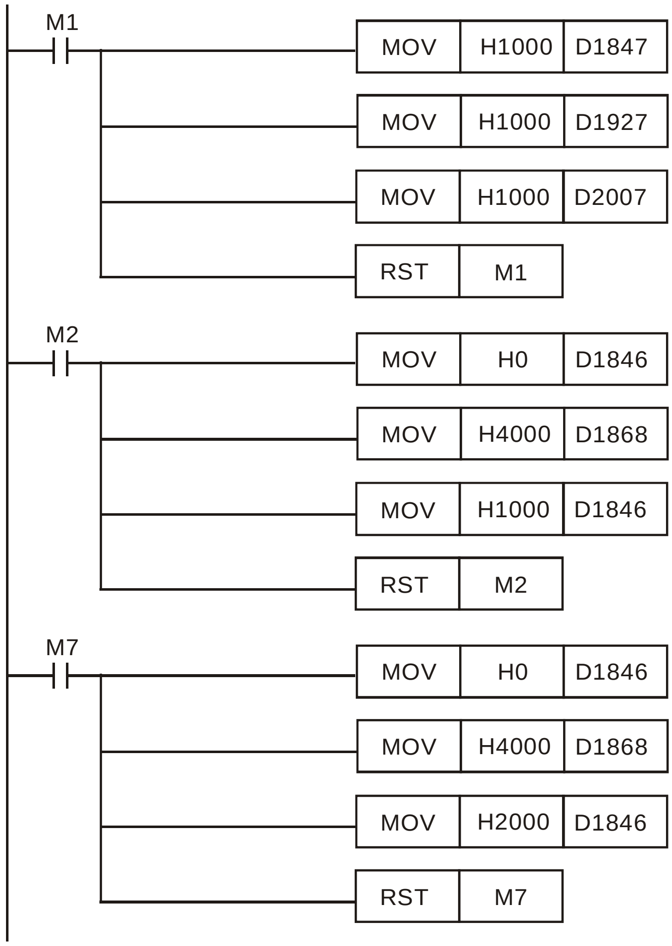
Example: Executing the motion subroutine Ox0



(A) Ox0 subroutine



(B) O100



The program in O100 is as figure (B).

Step 1: When M1 is ON, the work modes of three axes are set.

a. D1847=H1000

b. D1927=H1000

c. D2007=H1000

Step 2: When M2 is ON, the data in OX0 is loaded into CAM chart-0.

a. When the value in D1868 is H400, Ox0 is selected.

b. Load the data in OX0 into CAM chart-0.

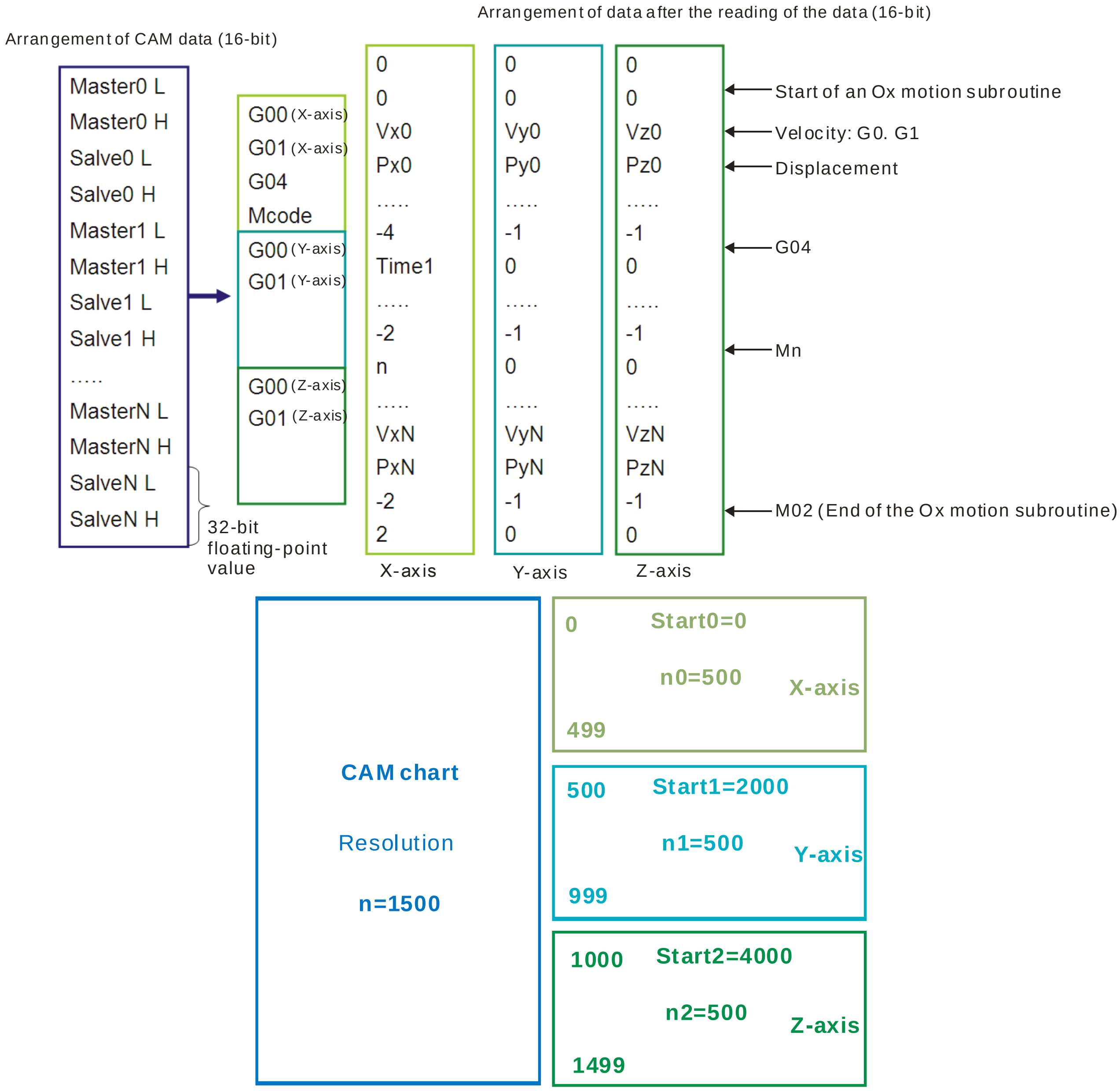
Step 3: When M7 is ON, Ox0 is executed. (The data source is in CAM chart-0)

a. When the value in D1868 is H4000, cam chart 0 is selected.

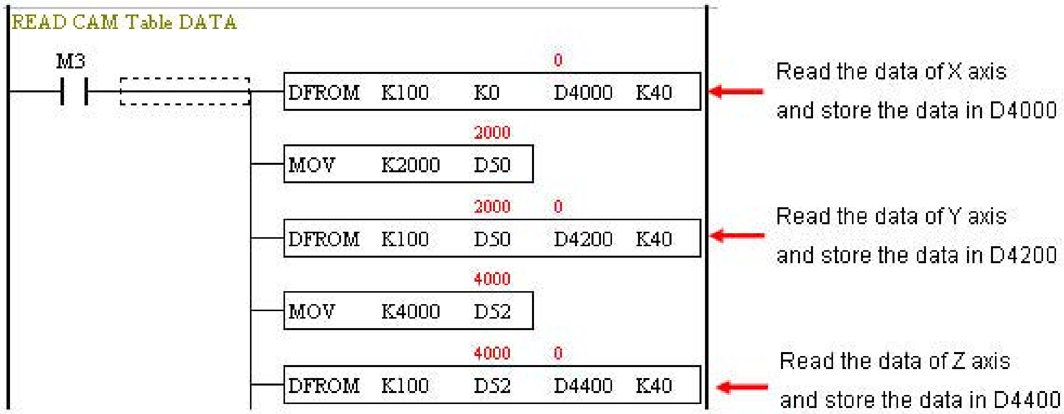
b. Enable cyclic electronic cam.

10.4 Storing G-codes / M-codes

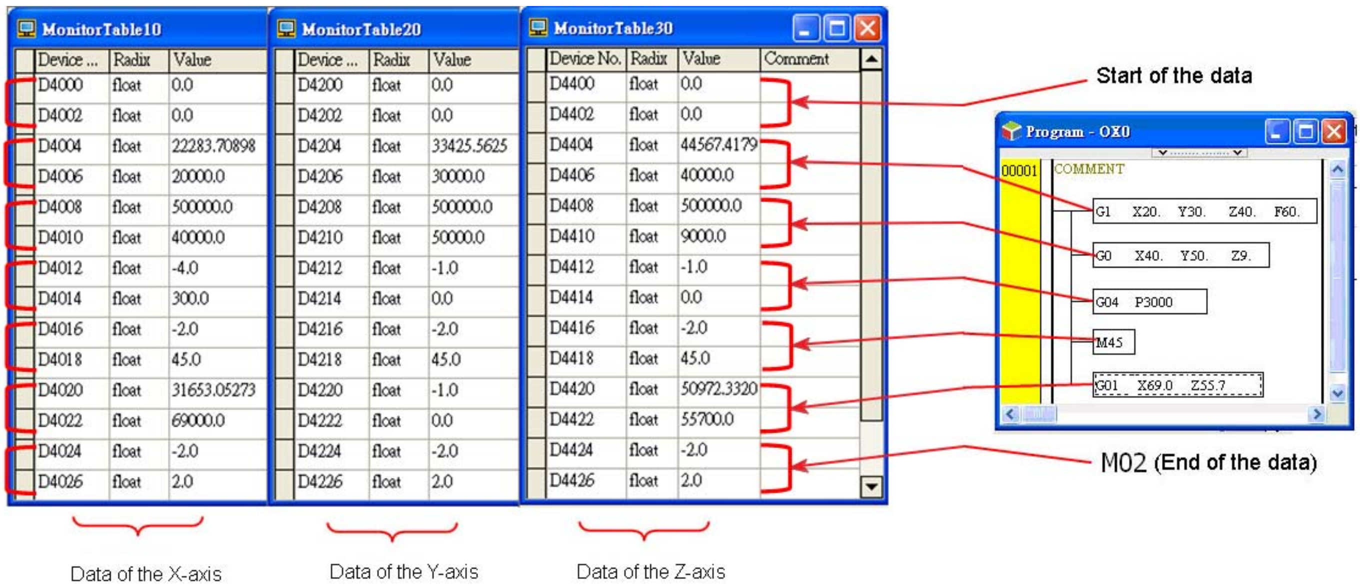
In this function, data in OXn is loaded to the corresponding CAM chart, and only G00, G01, G04 and M-codes applied in “With mode” are supported for this function. The below diagram explains the data allocation in CAM chart.



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.



P.S.: Note: In the instruction DFROM (DTO) K100 K0 D4000 K40, K100 indicates access in CAM chart-0. If K101 is set, it indicates access in CAM chart-1. If K102 is set, it indicates access in CAM chart-2.



Conclusions drawn from the data above:

1. The first data of each axis is 0.
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. Stopping 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 an M-code. The lower data is an 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. An M-code instruction should be placed in all axes.

11.1 Introduction of DVP-FPMC: CANopen Communication Card

DVP-FPMC is a CANopen communication card for a DVP-10PM series motion controller to conduct data exchange. The functions of DVP-FPMC are as follows.

- It conforms to the CANopen standard protocol DS301 v4.02.
- It supports an NMT protocol.
- It supports an SDO protocol.
- It supports the CANopen standard protocol DS402 v2.0. Four motion axes at most are supported.
- Motion axes support a profile position mode.

11.2 Specifications

- CANopen connector

Item	Specifications
Transmission method	CAN
Electrical isolation	500 V DC
Type	Removable connector (5.08 mm)
Transmission cable	2 communication cables, 1 shielded cable, and 1 ground

- Ethernet connector

Item	Specifications
Transmission method	Ethernet
Electrical isolation	500 V DC
Type	Removable connector (5.08 mm)
Transmission cable	2 communication cables, 1 shielded cable, and 1 ground

- Communication

Item	Specifications
Protocol type	PDO, SDO, SYNC (synchronous object), EMCY (emergency object), NMT, Heartbeat
Serial transmission speed	500 kbps, 1 Mbps (bits per second)
Product code	254
Equipment type	0 (Non-profile)
Company ID	477 (Delta Electronics, Inc.)

- Electrical specifications

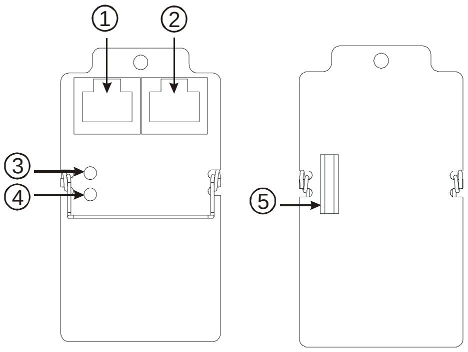
Item	Specifications
Supply voltage	24 V DC (-15~20%) (A DVP-20PM series motion controller supplies power through its internal bus.)
Electric energy consumption	1.7 W
Insulation voltage	500 V
Weight	Approximately 66 g

- Environmental specifications

Item	Specifications
Noise immunity	ESD (IEC 61131-2, IEC 61000-4-2): 8 kV air discharge, 4 kV contact discharge EFT (IEC 61131-2, IEC 61000-4-4): Power line: 2 kV; Digital I/O: 1 kV; Analog & Communication I/O: 1 kV Damped-Oscillatory Wave: Power line: 1 kV; Digital I/O: 1 kV RS (IEC 61131-2, IEC 61000-4-3): 80 MHz~1000 MHz, 1.4 GHz~2.0GHz, 10V/m
Operation/Storage	Operation: 0ºC ~ 55ºC (Temperature), 50~95% (Humidity) , pollution degree 2 Storage: -25ºC~70ºC (Temperature), 5~95% (Humidity)
Vibration/Shock resistance	International standards IEC 61131-2, IEC 68-2-6 (TEST Fc)/IEC 61131-2 & IEC 68-2-27 (TEST Ea)
Standard	IEC 61131-2

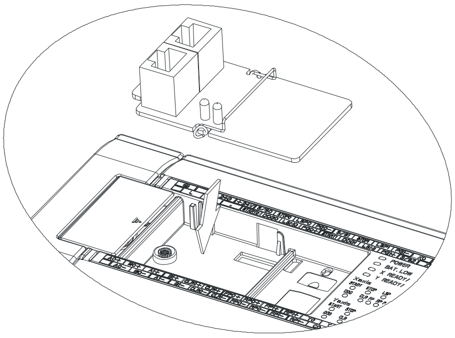
11.3 Product Profile and Installation

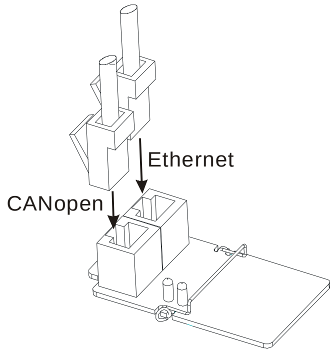
Product profile:



①	CANopen connector
②	Ethernet connector
③	CANopen indicator
④	Ethernet indicator
⑤	Communication connector

Installing DVP-FPMC on a DVP-20PM series motion controller, and connecting it to a communication cable:





11.4 Parameters for Control Registers

- Normal mode: Common parameters

CR number	Function	Attribute	Data type	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	R/W	Word	1
#056	DVP-FPMC error status	R/W	Word	1
#059	Network IP address and port setting for DVP-FPMC	R/W	Word	3
#062	Ethernet connection command and status	R/W	Word	1
#063	IP address and port setting for an Ethernet master	R/W	Word	3
#064	Length of data to be transmitted through Ethernet	R/W	Word	1
#065	Data to be transmitted through Ethernet	R/W	Word	512
#066	Length of data to be received through Ethernet	R	Word	1
#067	Data to be received through Ethernet	R	Word	512
#070	Node ID of an 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 transmission/reception register 1	R/W	Word	512
#074	SDO OD transmission/reception register 2	R/W	Word	512
#075	SDO OD transmission/reception register 3	R/W	Word	512
#076	SDO OD transmission/reception register 4	R/W	Word	512
#080	NMT command	R/W	Word	1

- A2 mode: Four-axis parameters

CR number	Function	Attribute	Data type	Length
#010	CANopen bus scan	R/W	Word	1
#020	CANopen bus communication status	R	Word	1
#040	Error status of a server	R	Word	1
#050	CANopen bus control command	R/W	Word	1
#090	Data written into a DVP-FPMC QBuffer	R/W	Word	32
#091	Address of the data written into a DVP-FPMC QBuffer	R/W	Word	32
#092	Data read from a DVP-FPMC QBuffer	R/W	Word	32
#093	Address of the data read from a DVP-FPMC QBuffer	R/W	Word	32
#**n**00	Node ID	R	Word	1
#**n**01 #**n**02	Manufacturer ID	R	Word	1
#**n**03 #**n**04	Product ID	R	Word	1
#**n**05 #**n**06	Firmware version	R	Word	1
#**n**07 #**n**08	Product type	R	Word	1
#**n**09	CANopen node communication status	R	Word	1
#**n**10	Emergency error code	R	Word	1
#**n**11 #**n**12	Manufacturer’s error code	R	Word	1
#**n**20	Servo drive status	R	Word	1
#**n**21	Present motion mode of a servo drive	R	Word	1
#**n**22 #**n**23	Servo drive position	R	Word	1
#**n**40	Node control command	R/W	Word	1
#**n**50	SDO access command and status	R/W	Word	1
#**n**51	SDO OD (object dictionary) index	R/W	Word	1
#**n**52	SDO transmission/reception register 1	R/W	Word	512
#**n**53	SDO transmission/reception register 2	R/W	Word	512
#**n**54	SDO transmission/reception register 3	R/W	Word	512
#**n**55	SDO transmission/reception register 4	R/W	Word	512
#**n**60	Servo drive control	R/W	Word	1
#**n**61	Motion mode selection	R/W	Word	1
#**n**70 #**n**71	Target position of a profile position mode	R/W	Word	1
#**n**72 #**n**73	Target speed of a profile position mode	R/W	Word	1
#**n**74 #**n**75	Acceleration time of a profile position mode (ms)	R/W	Word	1
#**n**76 #**n**77	Deceleration time of a profile position mode (ms)	R/W	Word	1
#**n**78	Profile position settings	R/W	Word	1
#**n**80	Homing method	R/W	Word	1
#**n**81 #**n**82	Home offset	R/W	Word	1
#**n**83 #**n**84	Homing speed	R/W	Word	1

CR number	Function	Attribute	Data type	Length
#**n**85	Speed at which motion homes after a transition in a DOG signal	R/W	Word	1
#**n**86				
#**n**87	Homing acceleration time	R/W	Word	1
#**n**88				
#**n**89	Enabling a homing mode	R/W	Word	1
#**n**90	Target position of an interpolation mode	R/W	Word	1
#**n**91				
#**n**92	Enabling an interpolation mode	R/W	Word	1

- CANopen common mode

CR number	Function	Attribute	Data type	Length
#500	CANopen mode switch	R/W	Word	1
#504	Enabling a heartbeat protocol	R/W	Word	1
#505	Execution status of a heartbeat protocol	R	Word	1
#506	Heartbeat statuses	R	Word	1

- Object dictionary parameters

CR number	Function	Attribute	Data type	Length
#H'1006	Synchronization cycle setting	R/W	DWord	1
#H'1017	DVP-FPMC heartbeat cycle setting	R/W	Word	1
#H'1400~#H'143F	Parameter settings for a RPDO	R/W	Word	3
#H'1600~#H'163F	Parameter settings for RPDO data mapping	R/W	DWord	4
#H'1800~#H'183F	Parameter settings for a TPDO	R/W	Word	3
#H'1A00~#H'1A3F	Parameter settings for TPDO data mapping	R/W	Dword	4
#H'2000~#H'207F	PDO data registers	R/W	Word	4

11.5 Descriptions of Control Registers

- Normal mode: Common parameters

CR#001: Firmware version of DVP-FPMC

[Description]

The firmware version of DVP-FPMC is displayed in a hex value, e.g. H'8161 indicates that the data of issuing the firmware of DVP-FPMC is "Afternoon, August 16".

CR#052: CANopen synchronous packet sending setting

[Description]

The control register has two functions.

- The low byte of CR052 sets up a CANopen synchronous function. If the value of the low byte is 1, DVP-FPMC will send out a synchronous packet. If the value is 0, the function will be disabled.
- The high byte of CR052 sets up a synchronous cycle. Setting value×5=Value in D1040. If the value of the high byte is greater than 0, the synchronization between the DVP-10PM series motion controller used and DVP-FPMC will be enabled.

Bit	Bit [15:8]	Bit [7:0]
Value	Synchronous cycle	Enabling the sending of a synchronous packet

CR#053: CANopen node ID setting

[Description]

The control register is used to set a CANopen node ID. A CANopen node ID is in the range of 5 to 127.

Default value: 127

CR#054: CANopen transmission speed setting

[Description]

The control register is used to set a CANopen transmission speed. The setting status is indicated by bit 15. If bit 15 is 1, the setting is in progress. If bit 15 is 0, the setting is completed. For example, if the CANopen transmission speed required is 1000 kb/s, users can write K1000 into CR#054.

Bit	Bit [15]	Bit [14:0]
Setting value	Setting status 0: Completed 1: In progress	1000: CANopen speed=1000 kb/s 500: CANopen speed=500 kb/s

CR#055: CANopen SDO/NMT timeout

[Description]

The control register is used to set a CANopen SDO/NMT timeout.

Unit: Millisecond

Default value: 1000

CR#056: DVP-FPMC error status

[Description]

The control register is used to display the error status of DVP-FPMC. Please refer to the table below for more information.

Error status	Value	Resolution
CANopen connection error	C1	Check the CANopen nodes of the present slaves.
Ethernet connection error	E1	Check the connection between the communication module and Ethernet.

CR#059: Network IP address and port setting for DVP-FPMC

[Description]

The control register is used to set an IP address and a port number for DVP-FPMC.

Data length: 3 words

Default IP address: 192.168.0.100

Port number: 1024

Please refer to the example below. (IP address: 192.168.0.100; Port number: 1024)

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

[Description]

The control register is used to set an Ethernet connection command, and obtain a connection status.

- H’0: Disconnected
- H’30: Connected
- H’10: Sending a connection command
- H’20: Sending a disconnection command

CR#063: IP address and port setting for an Ethernet master

[Description]

The control register is used to set an IP address and a port number for an Ethernet master.

Data length: 3 words

Please refer to the table below for more information.

Word 0		Word 1		Word 2
H-byte	L-byte	H-byte	L-byte	1024
192	168	0	100	

CR#064~CR#67: Length of data to be transmitted/received through Ethernet/Data to be transmitted/received through Ethernet

[Description]

The control register is used to set the data to be accessed through Ethernet. The maximum capacity is 1024 bytes.

- Sending data: After users write a data length data into CR#064, and data into CR#65, DVP-FPMC will automatically clear values in the two control registers to 0.
- Receiving data: Users read the contents of CR#066 first, and then read the data in CR#067.

CR#070: Node ID of an SDO server

[Description]

The control register is used to set the node ID of an SDO server. A node ID is in the range of 1 to 127.

CR#071: SDO access command and status

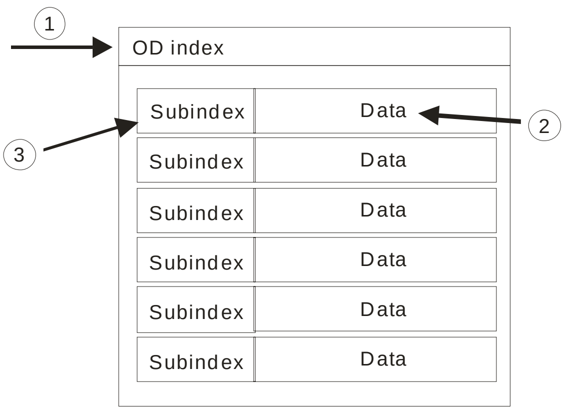
[Description]

The control register is used to set an SDO access command, and obtain a status. Please refer to the table below for more information.

Bit	Bit [15:8]	Bit [7:4]	Bit [3]	Bit [2:0]
Setting value	Subindex of a target OD index	Data length (Unit: Byte) Range: 1~8 If users want to write data, they have to specify a data length.	Error flag	Command: 0: Completed 1: Writing (including a check) 2: Reading (including a check) 3: Writing (not including a check) 4: Reading (not including a check)

Example: SDO data transmission

1. Specify the OD index of an SDO server (CR#070) in CR#072.
2. Set the data to be transmitted in CR#073~CR#076.
3. Refer to the table above. Specify a subindex in bit 15~bit 8 in CR#071, and an SDO access command.



CR#072: SDO OD index

[Description]

The control register is used to specify a target OD index.

Range: H'0000~H'FFFF.

CR#073~CR#76: SDO OD transmission/reception register 1~SDO OD transmission/reception register 4

[Description]

The data to be accessed through an SDO protocol is stored in the four control registers. The maximum capacity is 1024 bytes. If an error occurs during SDO data transmission, an error code will be stored in CR#073 and CR#074. If CR#073~CR#076 are used at a time, CR#073 functions as the LSB and CR#076 functions as the MSB.

CR#080: NMT command

[Description]

If DVP-FPMC is a master, an NMT command can be used to change a network status. Please refer to the table below for more information.

Bit	Bit [15:8]	Bit [7:0]
Setting value	Network management command 1: Enabling node communication 2: Disabling node communication 128: Switch an operation mode 129: Resetting node communication	Node ID of a slave

- Parameters for an A2 mode

  An A2 mode is one of the applications of DVP-FPMC specifically for Delta ASDA-A2 series servo drives. In an A2 mode, CANopen node ID 1~CANopen ID 4 are for ASDA-A2 series servo drives, and CR#100~CR#499 correspond to servo parameters. CR#100~CR#199 are control registers for node ID 1, CR#200~CR#299 are control registers for node ID 2, CR#300~CR#399 are control registers for node ID 3, CR#400~CR#499 are control registers for node ID 4. n in a control register number represents the digit in the hundreds place of the control register number. It is in the range of 1 to 4. Control registers for ASDA-A2 application are applicable only in an A2 mode.

CR#010: CANopen bus scan

[Description]

The control register is used to scan CANopen node ID 1~CANopen node ID 4. Bit 0~bit 3 in CR#010 correspond to node 1~node 4. If a bit is 1, its corresponding node will be scanned, and the contents of the control register will be cleared automatically. Please refer to the table below for more information.

Bit	Bit [15:4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]
Node number	Reserved	Node 4	Node 3	Node 2	Node 1

CR#020: CANopen bus communication status

[Description]

Two consecutive bits in the control register are used to display a node communication status.

00: Disconnected

01: Connected

11: Ready

Please refer to the table below for more information.

Bit	Bit [15:8]	Bit [7:6]	Bit [5:4]	Bit [3:2]	Bit [1:0]
Node number	Reserved	Node 4	Node 3	Node 2	Node 1

CR#040: Error status of a server

[Description]

The control register is used to display the error status of a servo drive. Bit 0~bit 3 in CR#010 correspond to node 1~node 4. If an error occurs, its corresponding bit will be 1. If an error reset command is executed, the contents of the register will be cleared automatically. Please refer to the table below for more information.

Bit	Bit [15:4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]
Node number	Reserved	Node 4	Node 3	Node 2	Node 1

CR#050: CANopen bus control command

[Description]

The control register is used to send bus control commands to the nodes connected successfully in a CANopen network. If the value in the control register is 1, the servos which have been connected are ON. If the value in the control registers is 128, the servos which have been connected are OFF. If the value in the control register is 129, the errors which appear are cleared. After the setting of the control register is completed, the contents of the register will be cleared automatically. Please refer to the table below for more information.

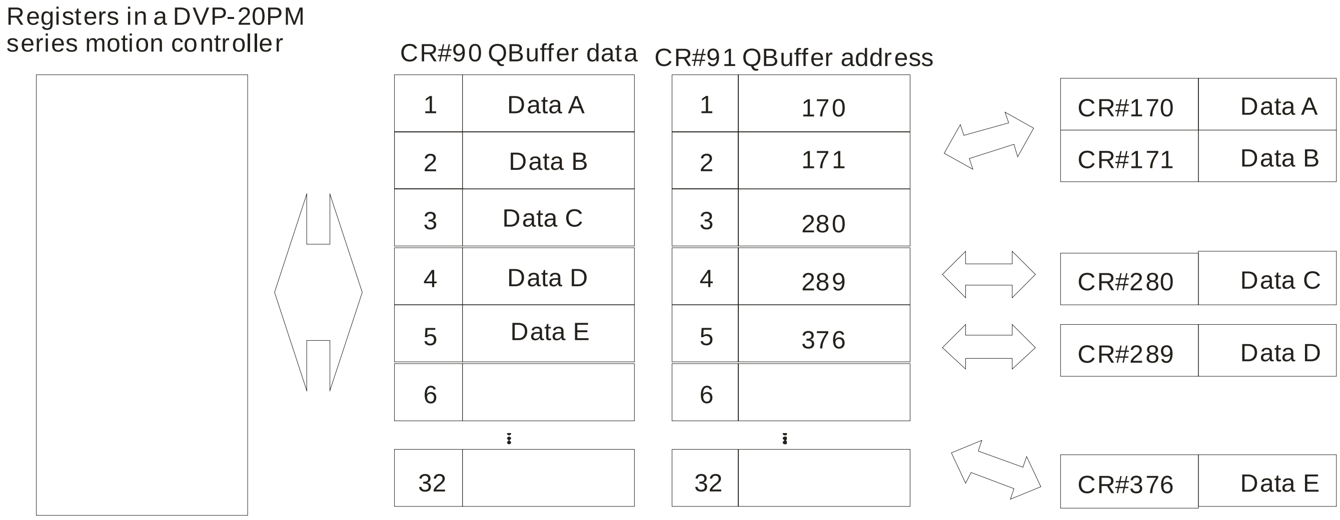
Bit	Bit [15:8]	Bit [7:0]
Value	Reserved	1: All servos are ON. 128: All servos are OFF. 129: All errors are cleared.

CR#090~CR#093: Data written into/read from a DVP-FPMC QBuffer/Address of the data written into/read from a DVP-FPMC QBuffer

[Description]

A QBuffer is designed for accessing several inconsecutive control registers by means of executing TO/FROM once. PLC scan time can be reduced by decreasing the number of times TO/FROM is executed. QBuffer access operation is described below.

For example, after the inconsecutive control register numbers CR#170, CR#171, CR#280, CR#289, and CR#376 are written into the QBuffer address CR#91, DVP-FPMC will connect the data in CR#170, CR#171, CR#280, CR#289, and CR#376 with CR#90 automatically. Users only need to access CR#90, and the data in CR#170, CR#171, CR#280, CR#289, and CR#376 can be modified.



CR#n00: Node ID

[Description]

The control register is used to display the node ID of a servo drive in a CANopen network.

Node ID=1: CR#100=1

Node ID=2: CR#200=2

Node ID=3: CR#300=3

Node ID=4: CR#400=4

CR#n01~CR#n02: Manufacturer ID

[Description]

The control registers are used to display the manufacturer ID of an ASDA-A2 series servo drive.

Data type: Double word

CR#n03~CR#n04: Product ID

[Description]

The control registers are used to display the product ID of an ASDA-A2 series servo drive.

Data type: Double word

CR#n05~CR#n06: Firmware version

[Description]

The control registers are used to display the firmware version of an ASDA-A2 series servo drive.

Data type: Double word

CR#n07~CR#n08: Product type

[Description]

The control registers are used to display the product type of an ASDA-A2 servo drive.

Data type: Double word

CR#n09: CANopen node communication status

[Description]

The control register is used to display a node communication status in a CANopen network. Please refer to the table below for more information.

Status	Value
Disconnected	H'1
Connected	H'2
Operation mode	H'5
Error	H'6
Reset	H'7

CR#n10: Emergency error code

[Description]

The control register is used to display an error code defined by a CANopen protocol when an error occurs in a certain node.

CR#n11~CR#n12: Manufacturer's error code

[Description]

The control registers are used to display an error code defined by a manufacturer when an error occurs in an ASDA-A2 series servo drive. Please refer to Delta ASDA-A2 User Manual for more information about error codes.

CR#n20: Servo drive status

[Description]

The value in the control register indicates the present status of an ASDA-A2 series servo drive. Please refer to the table below for more information.

	Status word															
	X	OM	OM	OM	X	TR	RM	X	WR	X	QS	X	FT	SO	X	RS
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- RS: The servo drive is ready. After the initialization of the servo drive is completed, the bit will be 1.
- SO: The servo drive is ON. The bit will be 1 if the servo drive is ON.
- FT: It is an error flag. If an error occurs in the servo drive, the bit will be 1.
- QS: If the bit is 1, the servo drive can be stopped urgently.
- WR: It is a warning flag. If the servo drive sends a warning message, the bit will be 1.
- RM: If the bit is 1, remote monitoring can be executed.
- TR: If the execution of a motion command is completed, the bit will be 1.
- OM [14:12]: The bits indicate the statuses of motion modes. Please see the table below for more information.

	Profile position mode	Homing mode	Interpolation mode
OM [12]	A target position has been set successfully.	A homing mode is being executed.	An interpolation mode is being executed.
OM [13]	Following error	Homing error	X

	Profile position mode	Homing mode	Interpolation mode
OM [14]	X	X	Enabling synchronization

CR#n21: Present motion mode of a servo drive

[Description]

The control register is used to display the present motion mode of a servo drive.

Value	Motion mode
0x01	Profile position mode
0x06	Homing mode
0x07	Interpolation mode

CR#n22~CR#n23: Servo drive position

[Description]

The control registers are used to display the present position of a servo drive.

Data type: Double word

CR#n40: Node control command

[Description]

The control register is used to send a node control command to the node connected. If the value in the control register is 1, the servo which has been connected is ON. If the value in the control registers is 128, the servo which has been connected is OFF. If the value in the control register is 129, the error which appears is cleared. Please refer to the table below for more information.

Bit	Bit [15:8]	Bit [7:0]
Value	Reserved	1: The servo which has been connected is ON. 128: The servo drive which has been connected is OFF. 129: The error which appears is cleared.

CR#n50: SDO access command and status

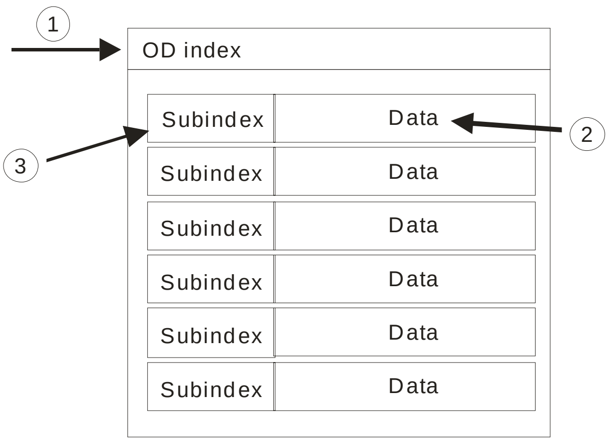
[Description]

The control register is used to set an SDO access command, and obtain a status. Please refer to the table below for more information.

Bit	Bit [15:8]	Bit [7:4]	Bit [3]	Bit [2:0]
Setting value	Subindex of a target OD index	Data length (Unit: Byte) Range: 1~8 If users want to write data, they have to specify a data length.	Error flag	Command: 0: Completed 1: Writing (including a check) 2: Reading (including a check) 3: Writing (not including a check) 4: Reading (not including a check)

Example: SDO data transmission

1. Specify the OD index of an SDO server in CR#n51.
2. Set the data to be transmitted in CR#n52~CR#n55.
3. Refer to the table above. Specify a subindex in bit 15~bit 8 in CR#n50, and an SDO access command.



CR#n51: SDO OD (object dictionary) index

[Description]

The control register is used to specify the OD index of a node.

Range: H’0000~H’FFFF.

CR#n52~CR#n55: SDO transmission/reception register 1~SDO transmission/reception register 4

[Description]

The data to be accessed through an SDO protocol is stored in the four control registers. The maximum capacity is 1024 bytes. If an error occurs during SDO data transmission, an error code will be stored in CR#n52 and CR#n53. If CR#n52~CR#n55 are used at a time, CR#n52 functions as the LSB and CR#n55 functions as the MSB.

CR#n60: Servo drive control

[Description]

The control register is used to send a control command to an ASDA-A2 series servo drive. Please refer to the table below for more information.

	Control word															
	X	X	X	X	X	X	X	X	FR	OM	OM	OM	EO	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- EO: The servo drive is enabled. The servo drive will be ON if the bit is 1.
- FR: The error occurs in the servo is cleared. The servo drive will clear the error which appears if the bit is 1.
- OM: It is used to control the function of motion modes. Please refer to the table below for more information.

	Profile position mode	Homing mode	Interpolation mode
OM [4]	Setting a new target position	Enabling a homing mode	X
OM [5]	A target position is allowed to be changed during motion.	X	X
OM [6]	Absolute/Relative positioning	X	X

Note: “X” indicates “Reserved”.

CR#n61: Motion mode selection

[Description]

The control register is used to set the motion mode of a servo drive. Please refer to the table below for more information.

Setting value	Motion mode
0x01	Profile position mode
0x06	Homing mode
0x07	Interpolation mode

CR#n70~CR#n71: Target position of a profile position mode

[Description]

The control registers are used to set the target position of a profile position mode.

Data type: Double word

CR#n72~CR#n73: Target speed of a profile position mode

[Description]

The control registers are used to set the target speed of a profile position mode.

Data type: Double word

CR#n74~CR#n75: Acceleration time of a profile position mode

[Description]

The control registers are used to set the acceleration time of a profile position mode.

Data type: Double word

CR#n76~CR#n77: Deceleration time of a profile position mode

[Description]

The control registers are used to set the deceleration time of a profile position mode.

Data type: Double word

CR#n78: Profile position settings

[Description]

The control register is used to set a profile position mode. A profile position mode can be absolute positioning or relative positioning.

- 0: Positioning is completed.
- 1: A profile position mode is absolute positioning. The value in the control register will be cleared to 0 after positioning is completed.
- 2: A profile position mode is relative positioning. The value in the control register will be cleared to 0 after positioning is completed.
- 3: A profile position mode is absolute positioning. The value in the control register will be retained after positioning is completed.

CR#n80: Homing method

[Description]

The control register is used to set a homing method.

Range: 1~35

For more information, please refer to chapter 13 in CiA DSP402 V2.0.

CR#n81~CR#n82: Home offset

[Description]

The control registers are used to set a home offset.

Range: -2,147,483,648~2,147,483,647

Data type: Double word

CR#n83~CR#n84: Homing speed

[Description]

The control registers are used to set a homing speed.

Range: 0~2,147,483,647

Data type: Double word

CR#n85~CR#n86: Speed at which motion homes after a transition in a DOG signal

[Description]

The control registers are used to set the speed at which motion homes after a transition in a DOG signal.

Range: 0~2,147,483,647

Data type: Double word

CR#n87~CR#n88: Homing acceleration time

[Description]

The control registers are used to set homing acceleration time.

Range: 0~2,147,483,647

Data type: Double word

CR#n89: Enabling a homing mode

[Description]

A homing mode will be executed if the value in the control register is 1. After homing is completed, the value in the control register will be cleared to 0 automatically.

CR#n90~CR#n91: Target position of an interpolation mode

[Description]

The control registers are used to set the target position of an interpolation mode.

Range: -2,147,483,648 ~2,147,483,647

Data type: Double word

CR#n92: Enabling an interpolation mode

[Description]

An interpolation mode will be executed if the value in the control register is 1. An interpolation mode will be disabled if the value in the control register is 0.

- CANopen common mode

CR#500: CANopen mode switch

[Description]

The control register is used to switch the CANopen mode of DVP-FPMC. If the value in the control register is 1, the CANopen mode of DVP-FPMC is an A2 mode. If the value in the control register is 2, the CANopen mode of DVP-FPMC is a normal mode.

Default value: 1

Control registers for an A2 mode will be unavailable if the CANopen mode of DVP-FPMC is a normal mode. Control registers for a normal mode will be unavailable if the CANopen mode of DVP-FPMC is an A2 mode.

CR#504: Enabling a heartbeat protocol

[Description]

If a heartbeat protocol is enabled in a common mode, the heartbeat mechanisms of node ID 1~nod ID 16 (slaves) will be enabled by a master. If the value in the control register is 0, a heartbeat protocol is disabled. If the value in the control register is 1, a heartbeat protocol is enabled.

CR#505: Execution statuses of a heartbeat protocol

[Description]

The control register is used to display the execution statuses of the heartbeat protocol executed by node ID 1~node ID 16. If the value in the control register is 0, the execution of a heartbeat protocol is completed. If the value in the control register is 1, a heartbeat protocol is being executed.

CR#506: Heartbeat statuses

[Description]

The control register is used to display the heartbeat statuses of node ID1~node ID 16. Node ID 1~node ID 16 correspond to bit 0~bit 15. If the heartbeat mechanism of a slave is enabled successfully, the bit corresponding to the salve is 1. If the heartbeat mechanism of a slave is not enabled, the bit corresponding to the salve is 0. In addition, if a slave is disconnected from a master, the bit corresponding to the slave is 0.

	Heartbeat statuses of slaves															
Node IDs	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

- Object dictionary parameters

CR#H'1006: Synchronization cycle setting

[Description]

The control register is used to set a time interval (unit) for the sending of CANopen synchronization packets in a normal mode.

Unit: μs

Data type: Double word

Default value: 5000

A synchronization cycle is measured by the millisecond now. The time less than one millisecond is ignored. It is suggested that the minimum synchronization cycle for 1 PDO~3 PDOs in a CANopen network should be 3 milliseconds, and the minimum synchronization cycle for 4 PDOs~8PDOs should be 4 milliseconds. That is to say, one millisecond will be added to a minimum synchronization cycle if four PDOs are added.

CR#H'1017: DVP-FPMC heartbeat cycle setting

[Description]

The control register is used to set a heartbeat cycle.

Unit: Millisecond

Default value: 0

If the value in the control register is 0, the heartbeat mechanism of DVP-FPMC is not enabled.

CR#H'1400~CR#H'143F: Parameter settings for a RPDO

[Description]

The control registers are used to set the parameters for RPDO in a normal mode. The capacity of the control registers is 3 words. Please refer to the table below for more information.

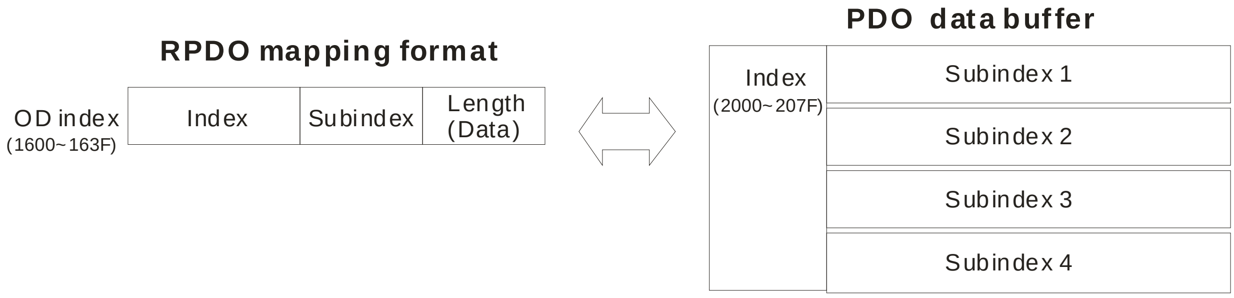
Transmission method	PDO ID	
Word 2	Word 1 (High)	Word 0 (Low)

- PDO ID: A CANopen POD ID occupies two words.
  Default values:
  CR#H'1400=H180+FPMC DVP-FPMC node ID (CR#053)
  CR# H'1401=H280+FPMC DVP-FPMC node ID (CR#053)
  CR#H'1403=H380+FPMC DVP-FPMC node ID (CR#053)
  CR#H'1404=H480+FPMC DVP-FPMC node ID (CR#053)
- Transmission method: If the value set is in the range of 1 to 240, the sending of a PDO is synchronized with the sending of a CANopen packet, and is executed every synchronization cycle. If the value set is in the range of 241 to 255, no PDO is sent.
  Default value: 241

CR#H'1600~CR#H'163F: Parameter settings for RPDO data mapping

[Description]

The control registers are used to set parameters for RPDO data mapping in a normal mode. A RPDO parameter is composed of a double word. The first word is used to set an OD Index. The high byte of the second word is used to set a subindex, and the low byte is used to set a data type. A bit is a unit for setting a data type. Please refer to the figure below for more information.



CR#H'1800~CR#H'183F: Parameter settings for a TPDO

[Description]

The control registers are used to set the parameters for TPDO in a normal mode. The capacity of the control registers is 3 words. Please refer to the table below for more information.

Transmission method	PDO ID	
Word 2	Word 1 (High)	Word 0 (Low)

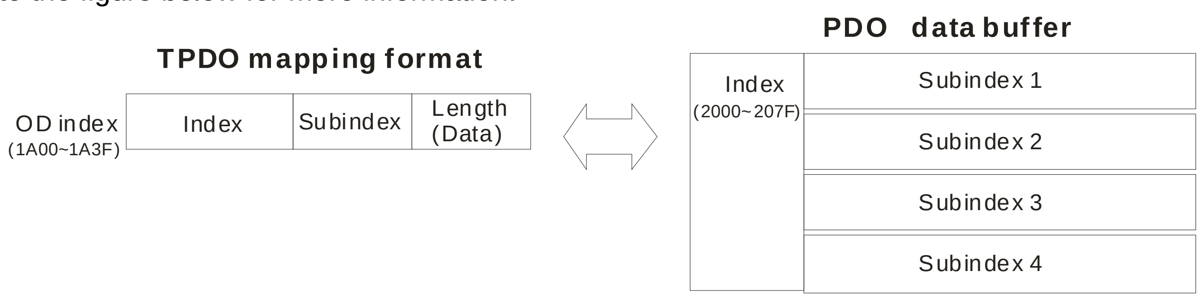
- PDO ID: A CANopen POD ID occupies two words.
  Default values:
  CR#H'1800=H200+ DVP-FPMC node ID (CR#053)
  CR# H'1801=H300+ DVP-FPMC node ID (CR#053)
  CR#H'1803=H400+ DVP-FPMC node ID (CR#053)
  CR#H'1804=H500+ DVP-FPMC node ID (CR#053)
- Transmission method: If the value set is in the range of 1 to 240, the sending of a PDO is synchronized with the sending of a CANopen packet, and is executed every synchronization cycle. If the value set is in the range of 241 to 255, no PDO is sent.
  Default value: 241

Transmission method	PDO ID	
Word 2	Word 1 (High)	Word 0 (Low)

CR#H'1A00~CR#H'1A3F: Parameter settings for TPDO data mapping

[Description]

The control registers are used to set parameters for TPDO data mapping in a normal mode. A TPDO parameter is composed of a double word. The first word is used to set an OD Index. The high byte of the second word is used to a subindex, and the low byte is used to set a data type (unit: bit). Please refer to the figure below for more information.



CR#H'2000~CR#H'207F: PDO data registers

[Description]

Data registers storing the data for PDO access. DVP-FPMC takes CR#H'2000~CR#H'207F in an OD as data registers. Every index has 4 subindices in which data can be stored. The size of a subindex is one word. If the size of the data to be accessed is bigger than one word, users have to use several subindex areas for data transmission.

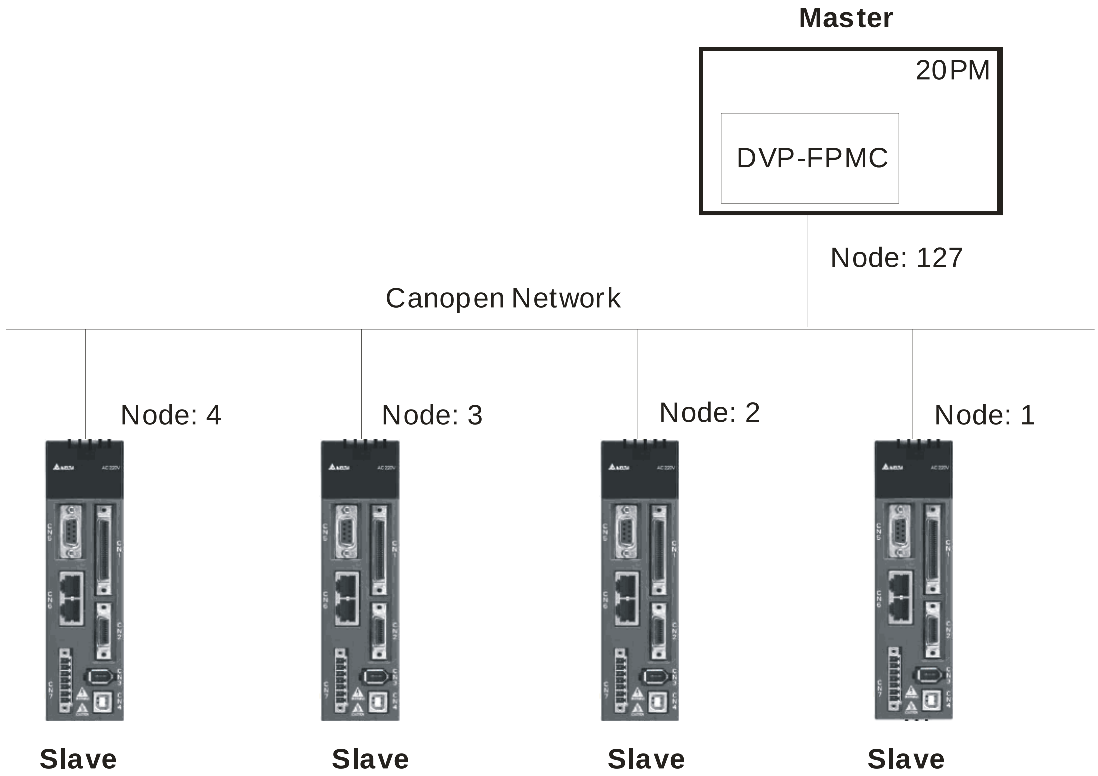
- Object dictionary for DVP-FPMC

Index (Hexadecimal value)	Object name	Number of subindices	Name	Data type	Attribute	Mapping target
1000	VAR	1	Product type	UNSIGNED32	RO	N
1006	VAR	1	Synchronization cycle	UNSIGNED32	RW	N
1018	ARRAY	5	Product information	UNSIGNED32	RO	N
1200	ARRAY	3	SDO parameter of a master	UNSIGNED32	RO	N
1280	ARRAY	4	SDO parameter of a slave	UNSIGNED32	RO	N
:	:	:	:	:	:	:
128F	ARRAY	4	SDO parameter of a slave	UNSIGNED32	RO	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
:	:	:	:	:	:	:
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 a profile position mode	UNSIGNED32	R	Y
6200	ARRAY	5	Present motion mode of a servo drive	UNSIGNED8	RW	Y
6300	ARRAY	5	Servo drive status	UNSIGNED16	RW	Y
6320	ARRAY	5	Servo drive position	UNSIGNED32	RW	Y

11.6 Setting a DVP-FPMC Mode

- A2 mode

  In an A2 mode, DVP-FPMC communicates with four Delta ASDA-A2 series servo drives through a CANopen network. During the communication, DVP-FPMC functions as a master, and the servo drives functions as slaves. The communication structure required is show below. The default node ID of DVP-FPMC is 127. The objects which are connected are node ID 1~ node ID 4. After users assign node ID 1~node ID4 to the servo drives, the servo drives can exchange data with DVP-FPMC.



In the A2 mode, there are six PDOs for the setting of servo parameters. The users can monitor the statuses of the servo drives directly by accessing control registers in a CANopen network. They do not need to set PDO parameters. Four PDOs are assigned to DVP-FPMC, and two PDOs are assigned to the servo drives. Please refer to the table below for more information.

PDO	Master (transmission)	Slave (transmission)
1	Target position of a profile position mode (CR#n70~CR#n71) Target speed of a profile position mode (CR#n72~CR#n73)	
2	Acceleration time of a profile position mode (CR#n74~CR#n75) Deceleration time of a profile position mode (CR#n76~CR#n77)	
3	Servo drive control (CR#n60)	
4	Target position of an interpolation mode (CR#n90~CR#n91)	
5		Servo drive status (CR#n20) Present motion mode of a servo drive (CR#n21)
6		Servo drive position (CR#n22~CR#n23)

Setting communication in an A2 mode:

- Setting the ASDA-A2 series servo drives

  Before creating a CANopen connection, the users have to set the servo drives to CANopen mode.

  1. Set P1-01 to H'0B. (The servo drive is set to CANopen mode.)
  2. Set P3-00. The value of P3-00 indicates a node number. It is in the range of H'01 to H'04.
  3. Set P3-01 to H'0403. The value of P3-01 indicates a baud rate. (If the high byte of the value of P3-01 is 2, the baud rate used is 500 kbps. If the high byte of the value of P3-01 is 4, the baud rate used is 1 Mbps.) The baud rates which are supported by DVP-FPMC now are 1 Mbps and 500 kbps. (Default: 1 Mbps)

- Setting DVP-FPMC

  After the setting of the CNopen parameters in the servo drives connected is completed, the users can create a CANopen network by means of DVP-FPMC.

  1. Write 1 into CR#500. DVP-FPMC is set to A2 mode.
  2. Write a node ID into CR#053. The default node ID of DVP-FPMC is 127.
  3. Write H'FFFF into CR#010. All servo drives which are connected are scanned.
  4. Read the value in CR#010 by means of the instruction FROM, and check whether the value in CR#010 is cleared to 0.
  5. Write 1 into CR#050. All servo drives which has been connected are set to ON.

■ Normal mode

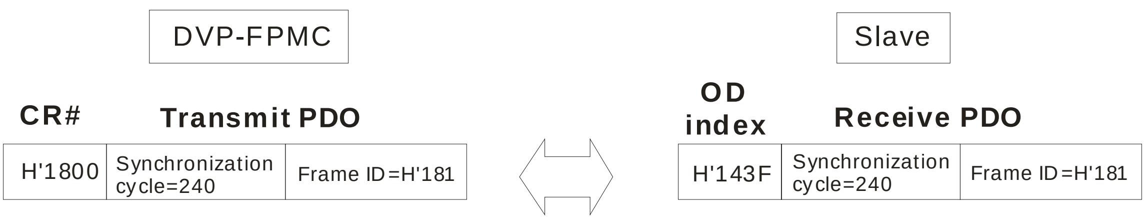
In a normal mode, users have to set the PDO parameters in DVP-FPMC and the slaves which are connected. They have to use FROM/TO to set control registers in DVP-FPMC, and use an SDO protocol to set the PDO parameters in the servo drives connected. The steps of setting the PDO parameters in DVP-FPMC are as follows.

1. Setting PDO transmission parameters

   The setting of PDO parameters includes the setting of a frame ID and the setting of a synchronization cycle. A frame ID is in the range of H'181 to H'578. Please note that the frame ID and the synchronization cycle in the PDO for a master must be the same as the frame ID and the synchronization cycle in the PDO for the slave connected. There are two kinds of PDOs: transmit and receive PDOs (TPDO and RPDO).

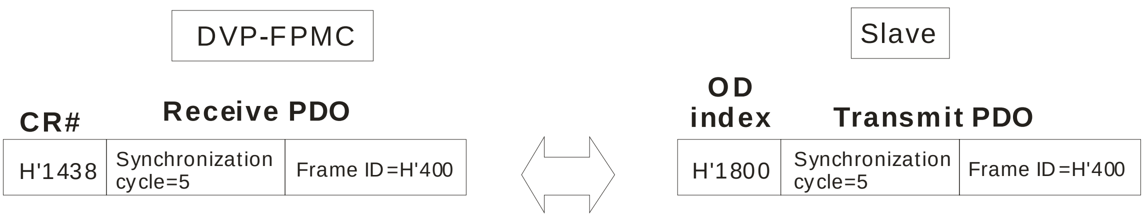
   - Setting a TPDO

     CR#H'1800~CR#H'183F in DVP-DPMC function as TPDOs. They communicate with a slave's RPDOs. For example, the OD index H'1800 (TPDO) in a master communicates with the OD index H'143F (RPDO) in a slave. The synchronization cycle set is 240, and the frame ID set is H'181.



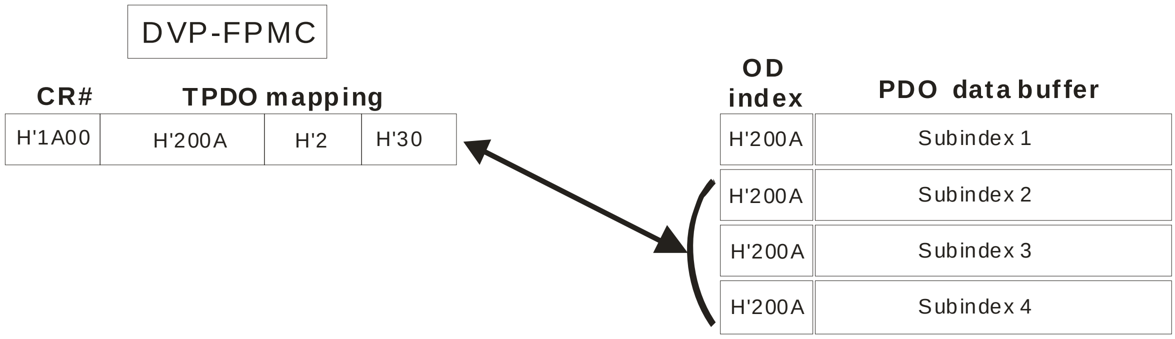
   - Setting a RPDO

     CR#H'1400~CR#H'143F in DVP-DPMC function as RPDOs. They communicate with a slave's TPDOs. For example, the OD index H'1438 (RPDO) in a master communicates with the OD index H'1800 (TPDO) in a slave. The synchronization cycle set is 5, and the frame ID set is H'400.



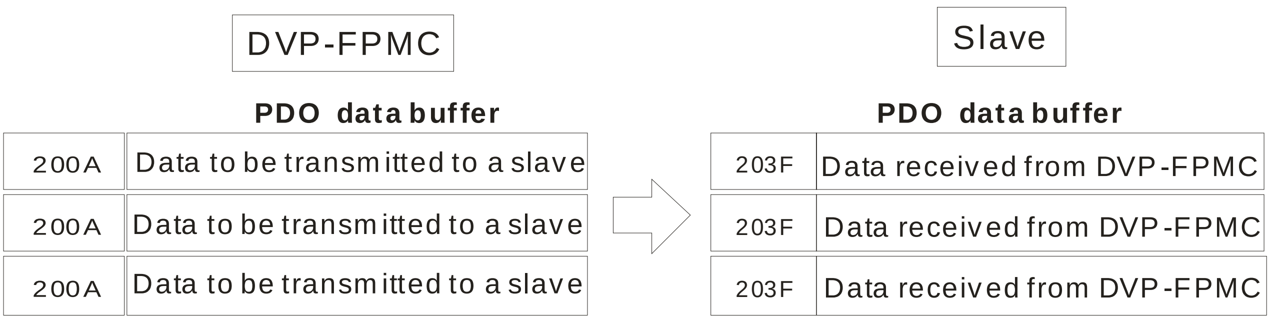
2. Setting PDO mapping parameters

   The setting of PDO mapping parameters includes the setting of a mapping target and the setting of a data length in a PDO data buffer. The maximum data length which can be set is 64 bits, that is to say, four subindices in a PDO data buffer can be occupied at a time. For example, the mapping target set is the second subindex in CR#H’200A, and the data length set is 48 bits. Subindex 2~subindex 4 in the PDO data buffer used are occupied.



3. Setting PDO data

   Write the data to be transmitted by a PDO into the OD indices set. For example, subindex 2~subindex 4 in CR#H’200A in DVP-FPMC is TPDO data, and subindex 1~subindex 3 in CR#H’203F in the slave connected is RPDO data. After communication is enabled, data will be transmitted/received every synchronization cycle.



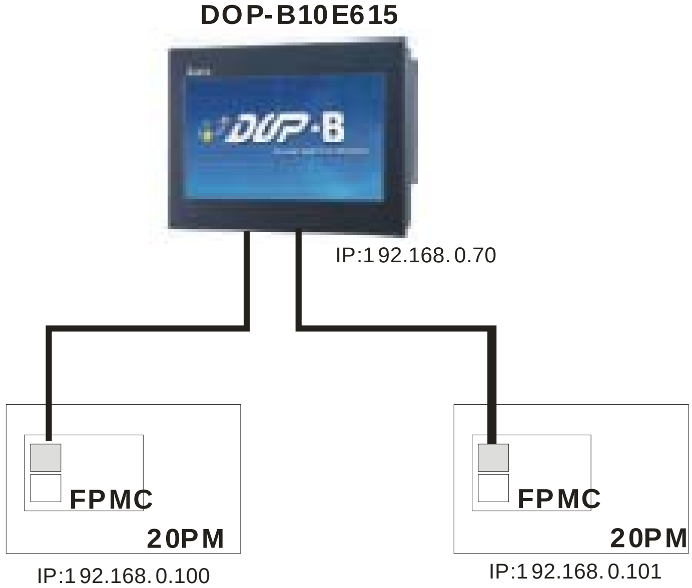
11.7 Ethernet Mode of DVP-FPMC

DVP-FPMC supports Ethernet connection. It can be connected to an Ethernet device or a PC. If DVP-FPMC is connected to a PC, PMSoft can be used to upload/download a program and monitor devices. Users only need to connect the communication port on DVP-FPMC to a communication port on equipment. Please refer to section 11.3 for more information about installing hardware. If DVP-FPMC is connected to a PC, the Ethernet LED indicator will be ON. Please check the setting of hardware or the setting of the PC connected if the Ethernet LED indicator is not ON.

11.7.1 Communication between DVP-FPMC and an HMI

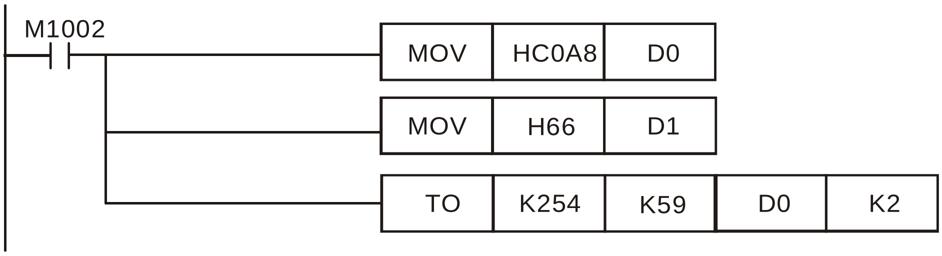
- Configuration

  In this example, two DVP-20PM series motion controllers equipped with DVP-FPMC exchange data with the HMI DOP-B10E615 through Ethernet. The hardware configuration required is shown below. The program in the HMI controls Y0~Y7 on the two DVP-20PM series motion controllers.



- Setting DVP-FPMC

  In this example, DVP-FPMC functions as a slave. The IP address of DVP-FPMC needs to be set. The IP address of the equipment to be connected and Ethernet connections do not need to be set. Users need to write the IP address of DVP-FPMC into CR#59. Please refer to 9.5 for more information about setting CR#59. Take the IP address 192.168.0.100 for instance. The program in DVP-PM is shown below.

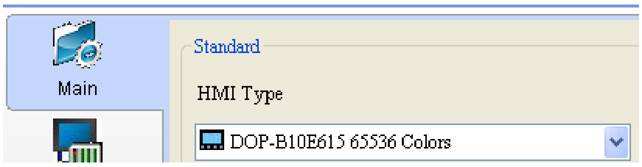


■ Setting an HMI

An HMI is used as a master. It is connected to two slaves. The IP address of the HMI and Ethernet connections need to be set. The steps of creating the connection between the HMI and DVP-FPMC are as follows.

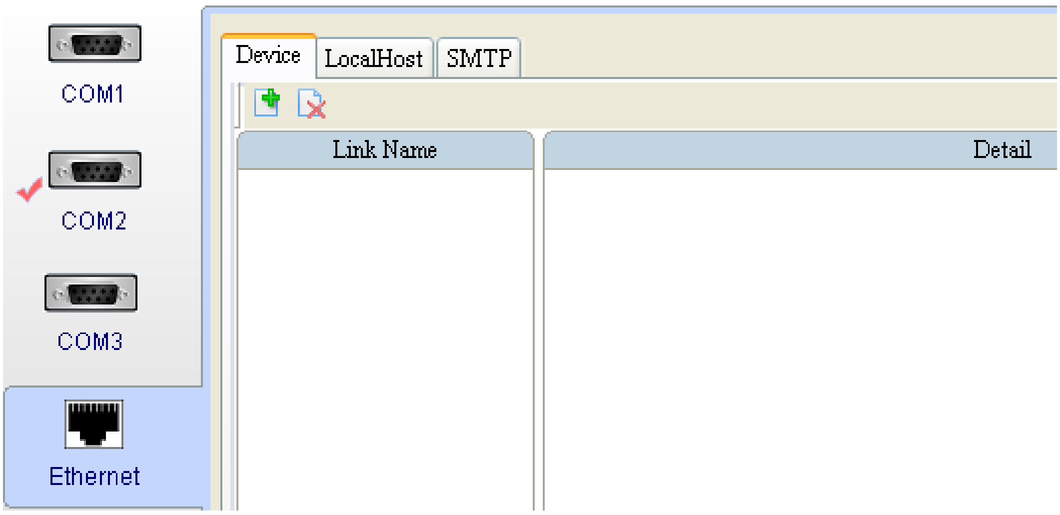
1. Click **Configuration…** on the **Option**s menu.

2. Click the **Main** tab, and then select **DOP-B10E615 65536 Colors** in the **HMI Type** drop-down list box.

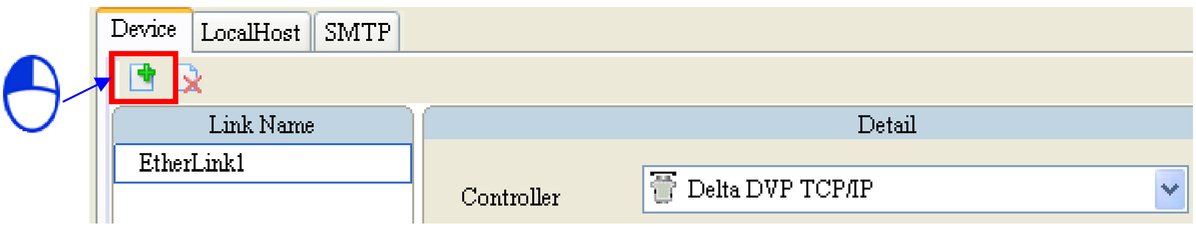


3. Click **Communication Setting** on the **Options** menu.

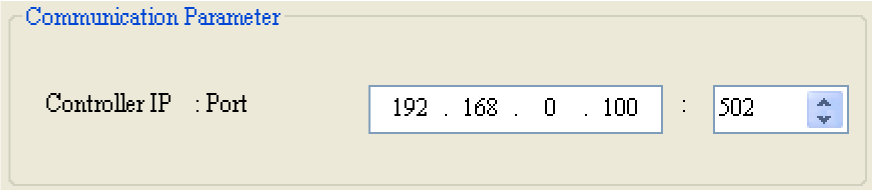
4. Click the **Ethernet** tab.



5. After users click , they have to type a link name in the **Link Name** box, and select **Delta DVP TCP/IP** in the **Controller** drop-down list box.

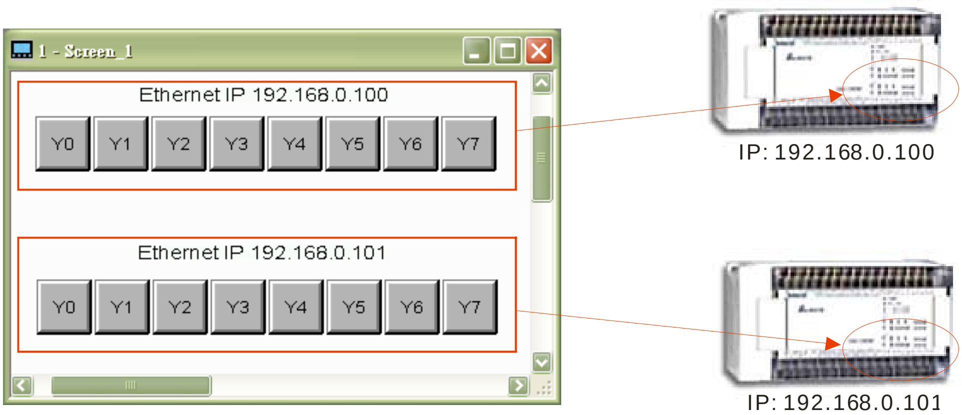


6. The users have to set the IP address of DVP- FPMC in the **Communication Parameter** section.



7. After the users select the link name created in step 5 in the **Input** window for an element, they can operate the memory defined by the element by means of Ethernet.

The HMI needs to control Y0~Y7 on two DVP-10PM series motion controllers which function as slaves. The interface required is shown below. The buttons Y0~Y7 correspond to Y0~Y7 on EtherLink1 and EtherLink2, that is to say, they correspond to Y0~Y7 on the two slaves connected. After the setting described above is completed, the HMI can connect to the two slaves by means of Ethernet.

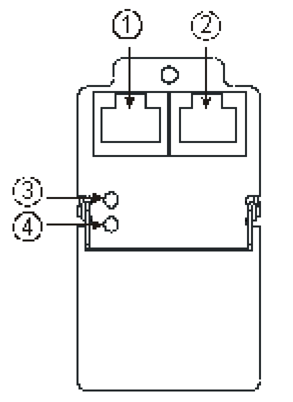


11.7.2 Communication between DVP-FPMC and PMSoft

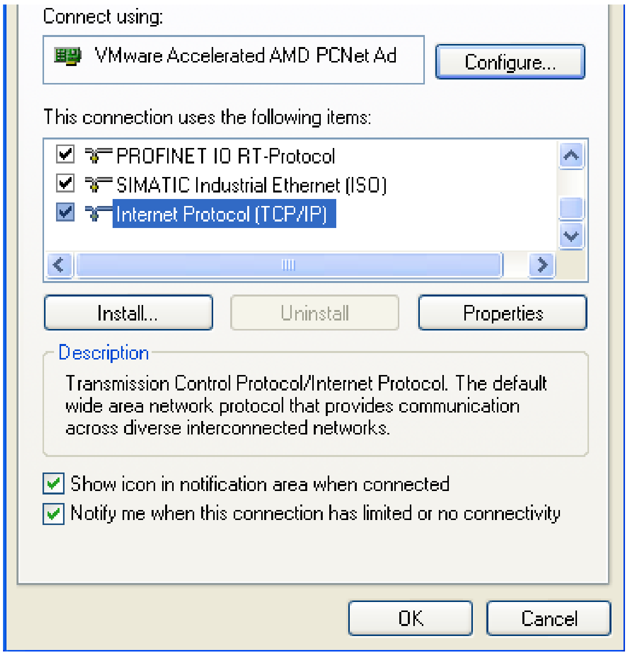
Before users create communication between DVP-FPMC and PMSoft, they have to use COMMG to create an Ethernet driver. An Ethernet driver can be used to upload the program in a DVP-10PM series motion controller, download a program into a DVP-20PM series motion controller, and monitor a DVP-20PM series motion controller.

■ Wiring hardware

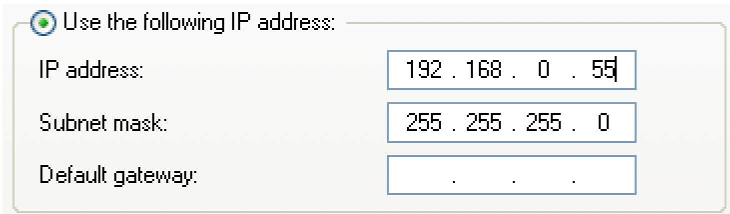
Users can connect the network port on DVP-FPMC to a network port on a PC by means of a network cable. If DVP-FPMC is connected to a PC, the Ethernet LED indicator on DVP-FPMC will be ON. Please check the setting of hardware and or the setting of the PC is the Ethernet LED indicator is not ON.

①	CANopen port	
②	Ethernet port	
③	CANopen LED indicator	
④	Ethernet LED indicator	

- Setting a PC
  1. Click **Internet Protocol (TCP/IP)** in the **Local Area Connection Properties** window, and then click **Properties**.

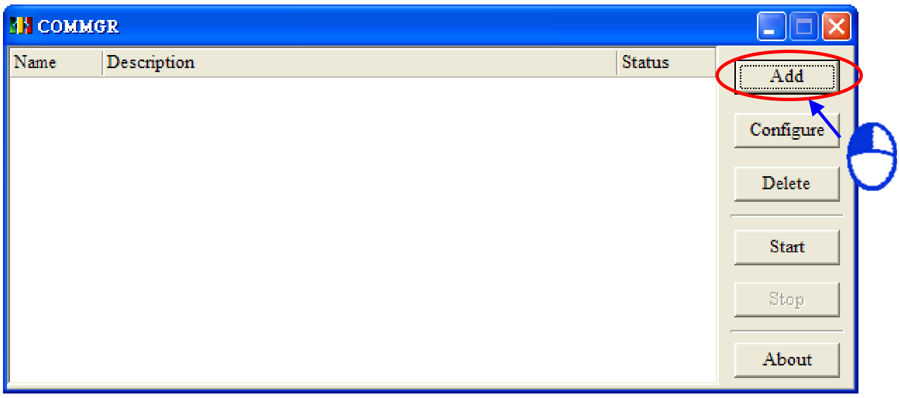


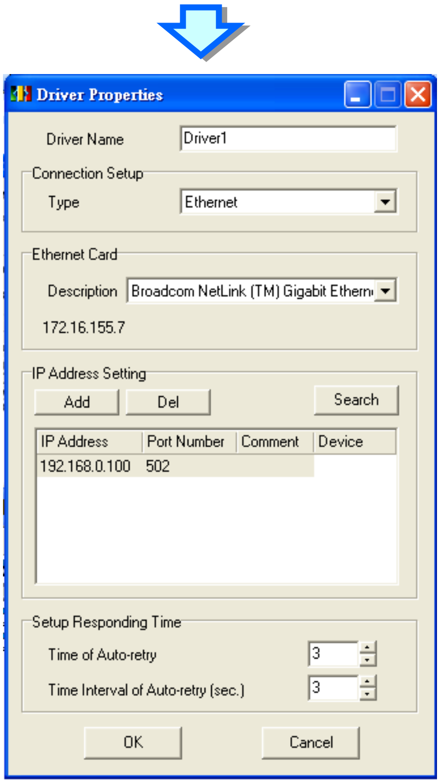
  2. Select the **Use the following IP address** option button in the **Internet Protocol (TCP/IP) Properties** window. Type 192.168.0.55 in the **IP address** box. The last number is in the range of 1 to 255, bit it can not be 100. Type 255.255.255.0 in the **Subnet mask** box, and click **OK**.



■ Setting PMSoft

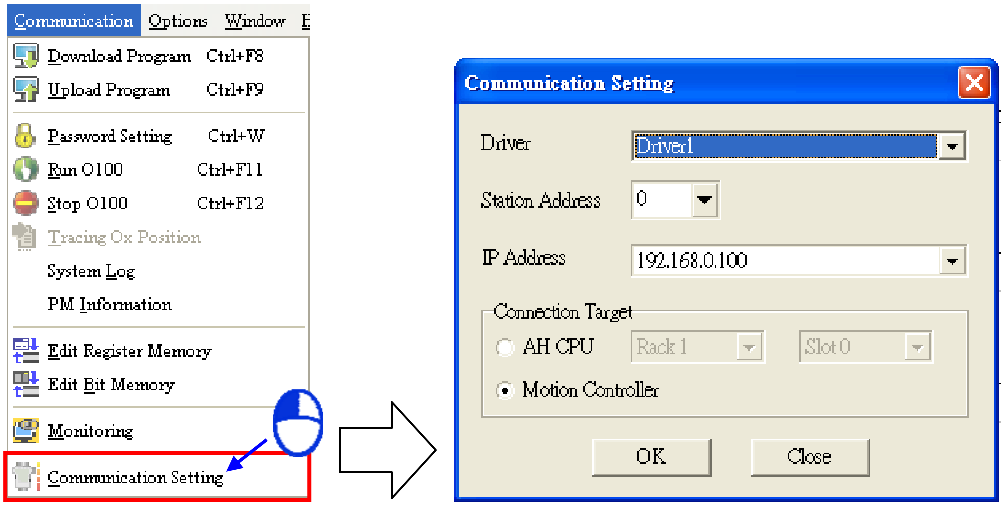
1. Click **Add** in the **COMMGR** window, and then create an Ethernet driver in the **Driver Properties** window.





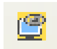
The IP Address set is 192.168.0.100, and the port number set is 502.

2. Start PMSoft, and click **Communication Setting** on the **Communication** menu. In the **Communication Setting** window, select the driver created in the first step in the **Driver** drop-down list box, and select the IP address 192.168.1.100. After OK is clicked, users can upload/download a program and monitor devices by means of Ethernet.



3. Downloading a program: If users want to download a program, they can click  on the toolbar, or click **Download Program** on the **Communication** menu. The procedure for downloading a program through Ethernet is the same as the procedure for downloading a program through a general communication port.

4. Uploading a program: If users want to upload a program, they can click  on the toolbar, or click **Upload Program** on the **Communication** menu. The procedure for uploading a program through Ethernet is the same as the procedure for uploading a program through a general communication port.

5. Monitoring a DVP-20PM series motion controller: If users want to monitor a DVP-10PM series motion controller, they can click  on the toolbar, or click **Monitoring** on the **Communication** menu. The procedure for monitoring a DVP-10PM series motion controller through Ethernet is the same as the procedure for monitoring a DVP-10PM series motion controller through a general communication port.

11.8 LED Indicators and Troubleshooting

- CANopen LED indicator

LED indicator	Description	Resolution
The green light is OFF.	A CANopen cable is not connected.	Check whether cables are connected correctly.
The green light is ON.	A CANopen cable is connected normally.	No action is required.

- Ethernet LED indicator

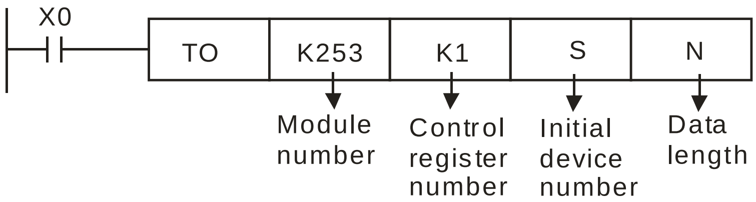
LED indicator	Description	Resolution
The green light is OFF.	DVP-FPMC is not connected to a network.	Check whether a network cable is connected correctly.
The green light is ON.	DVP-FPMC is connected to a network normally.	No action is required.
The green light blinks.	There is data exchange.	

MEMO

12.1 High-speed Comparison and High-speed Capture

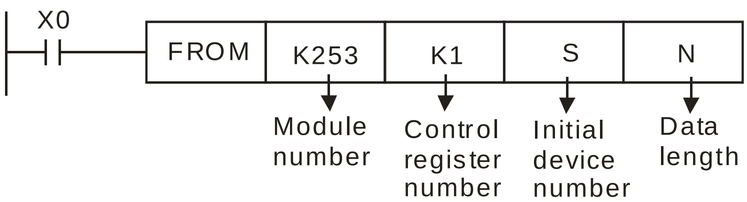
A DVP-20PM series motion controller sets and reads values by means of the instructions FROM and TO. The use of FROM/TO to set high-speed comparison and high-speed capture, and to read values is described below.

- Control



Device	Control	Resetting output devices	Setting a range which is masked
S	Initial group number n (n=0~7)	0	0
S_{+1}	0	1	2
(S_{+3}, S_{+2})	Control register whose group number is n		
(S_{+5}, S_{+4})	Data registers whose group numbers are n		
(S_{+7}, S_{+6})	Control register whose group number is n+1		
(S_{+9}, S_{+8})	Data registers whose group numbers are n+1		
:	:		
(S_{+31}, S_{+30})	Control register whose group number is n+7		
(S_{+33}, S_{+32})	Data registers whose group numbers are n+7		
N	Data length=2+m*4 m=number of groups (8 groups at most can be used.)		

- Reading



Device	Reading the values in counters	Reading the states of output devices/Enabling capture
S	Initial group number n (n=0~7)	0
S_{+1}	0	1
(S_{+3}, S_{+2})	Control register whose group number is n	States of output devices
(S_{+5}, S_{+4})	Data registers whose group numbers are n	Enabling capture (8 bits)
(S_{+7}, S_{+6})	Control register whose group number is n+1	
(S_{+9}, S_{+8})	Data registers whose group numbers are n+1	
:	:	
(S_{+31}, S_{+30})	Control register whose group number is n+7	
(S_{+33}, S_{+32})	Data registers whose group numbers are n+7	
N	Data length=2+m*4 m=number of groups (8 groups at most can be used.)	

- Control/Reading

(1) The format of a control register in a high-speed comparison mode is described below.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Item					Comparison result				Output action		Condition		Comparison source			

Item	Bit	Setting value	DVP-20PM series motion controller
Comparison source	[3-0]	0	Present position of the X-axis
		1	Present position of the Y-axis
		2	Present position of the Z-axis
		3	Value in C200
		4	Value in C204
		5	-
		6	-
		7	-
Comparison condition	[5-4]	1	Equal to (=)
		2	Greater than or equal to (≧)
		3	Less than or equal to (≦)
Output action	[7-6]	0	Set
		1	Reset
		2, 3	-
Comparison result	[11-8]	0	CLR0
		1	CLR1
		2	Y2
		3	Y3
		4	Clearing the value in C200
		5	Clearing the value in C204

(2) The format of a control register in a high-speed capture mode is described below.

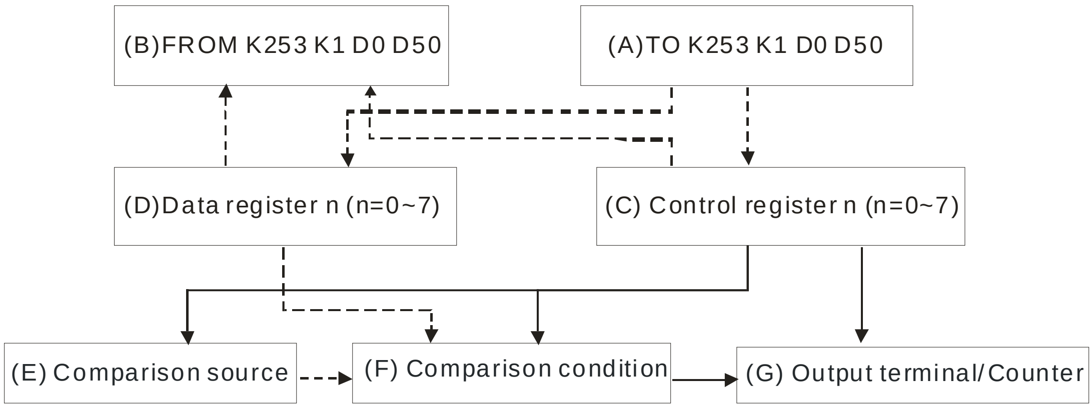
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Item	Trigger				—						Setting		Capture source			

Item	Bit	Setting value	DVP-20PM series motion controller
Capture source	[3-0]	0	Present position of the X-axis
		1	Present position of the Y-axis
		2	Present position of the Z-axis
		3	Value in C200
		4	Value in C204
		5	-
		6	-
		7	-
Setting	[5-4]	0	Capture mode
External trigger	[15-12]	0	PG0± for the X-axis
		1	B0± for the X-axis
		2	A0± for the X-axis
		3	LSN0 for the X-axis
		4	LSP0 for the X-axis
		5	DOG0 for the X-axis
		6	STOP0 for the X-axis
		7	START0 for the X-axis
		8	PG1± for the Y-axis
		9	B1±for the Y-axis
		10	A1±for the Y-axis
		11	LSN1 for the Y-axis
		12	LSP1 for the Y-axis
		13	DOG1 for the Y-axis

Item	Bit	Setting value	DVP-20PM series motion controller
External trigger	[15-12]	14	STOP1 for the Y-axis
		15	START1 for the Y-axis

12.2 High-speed Comparison

A high-speed comparison is shown below. Users use FROM/TO to read/write values so that they can compare data.



※ The dotted lines are data procedures, and the solid lines are control procedures.

Block (A): The instruction TO is used to write data into control registers (block C) and data registers (block D).

Block (B): The instruction FROM is used to read data from control registers (block C) and data registers (block D).

Block (C): User set a comparison source (block E), a comparison condition (block F), and an output terminal (block G) in a control register in accordance with the value it receives by means of TO.

Block (D): The value that users write into data registers by means of the instruction TO is compared with a comparison source (block E).

Block (E): The present positions of three axes, the values in C200 and C204 are comparison sources.

Block (F): There are three comparison conditions, they are equal to, greater than or equal to, and less than or equal to. If block D and block E meet the comparison condition set, the output terminal selected will be set to ON, the counter selected will be reset, the output terminal selected will be reset to OFF, or the counter selected will not be reset.

Block (G): If a comparison condition is met, CLR0±, CLR1±, Y2, Y3, C200, or C204 will be set or reset.

Procedure for a high-speed comparison: The instruction TO is used to write data into control registers and data registers (block A).→The comparison source set (block E) is compared with the value in data registers (block D). The comparison result meets the condition set (block F).→CLR0±, CLR1±, Y2, Y3, C200, or C204 will be set or reset (block G).

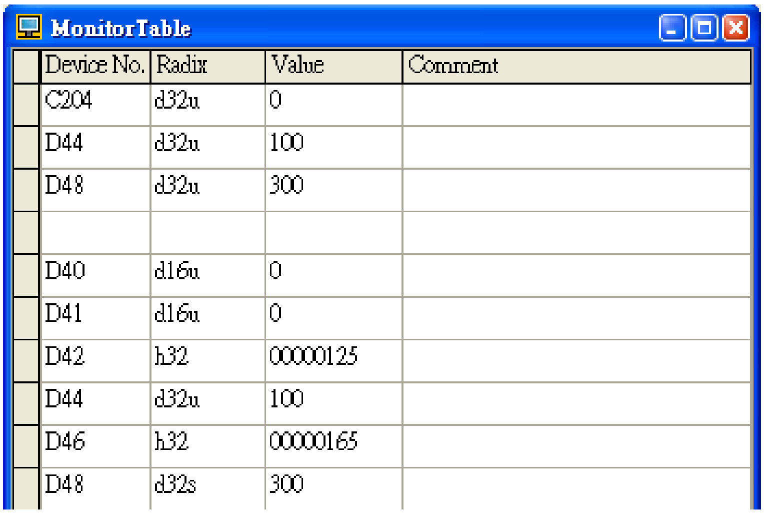
■ Example

【Description】

The high-speed counter C204 is used. If the value in C204 is greater than 100, Y1 will be set to ON. If the value in C204 is greater than 300, Y1 will be reset to OFF. Two comparators are used in a program. One comparator is used to set Y1 to ON, and the other is used to reset Y1 to OFF. When Y1 is set to ON, no LED indicator on the DVP-20PM series motion controller used will indicate that Y1 is ON, but users can know whether Y1 is ON by means of its external wiring. As a result, the terminal C1 is connected to the terminal 24G, Y1 is connected to X7, S/S2 is connected to +24V. A manual pulse generator is used, and is connected to A1± and B1±.

【Steps】

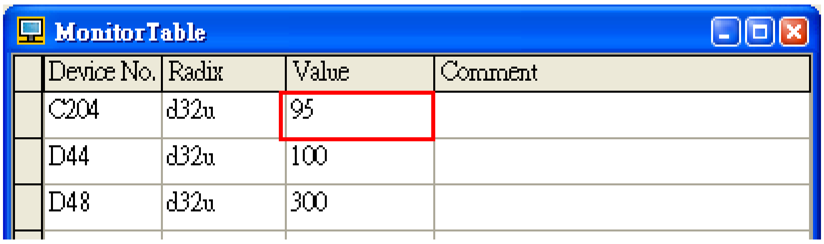
1. After O100 is started, the initial setting of two high-speed comparisons will be carried out.
   (1) D0=0→Initial group number n=0
   (2) D1=0
   (3) D20=10→Writing 10 values by means of the instruction TO (two groups of high-speed comparison values)
   (4) D60=10→Reading 10 values by means of the instruction FROM (two high-speed comparison values)
2. Two groups of high-speed comparison values are set when M1 is ON.
   (1) First group: The value in (D3, D2) is H125.→The comparison source set is C204. (The value of bit 3~bit 0 is 5.) The comparison condition set is greater than or equal to. (The value of bit 5~bit 4 is 2.) The output action selected is set. (The value of bit7~bit 6 is 0.) The terminal selected is Y1 (The value of bit11~bit 8 is 1.)
   (2) First group: The value in (D5, D4) is K100. If the value in C204 is greater or equal to K100, Y1 will be set to ON.
   (3) Second group: The value in (D7, D6) is H165.→The comparison source set is C204. (The value of bit 3~bit 0 is 5.) The comparison condition set is greater than or equal to. (The value of bit 5~bit 4 is 2.) The output action selected is reset. (The value of bit7~bit 6 is 1.) The terminal selected is Y1. (The value of bit11~bit 8 is 1.)
   (4) Second group: The value in (D9, D8) is K300. If the value in C204 is greater or equal to K300, Y1 will be reset to OFF.
3. The two high-speed comparisons are started when M2 is ON.
4. The setting of the two high-speed comparisons is read when M3 is ON.



MonitorTable

Device No.	Radix	Value	Comment
C204	d32u	0	
D44	d32u	100	
D48	d32u	300	
D40	d16u	0	
D41	d16u	0	
D42	h32	00000125	
D44	d32u	100	
D46	h32	00000165	
D48	d32s	300	

5. When M4 is ON, K1 is moved to M1204~M1207. C204 is started when M5 is set to ON. (Mode of counting: Pulse/Direction)
6. Use the manual pulse generator, and check whether C204 counts.

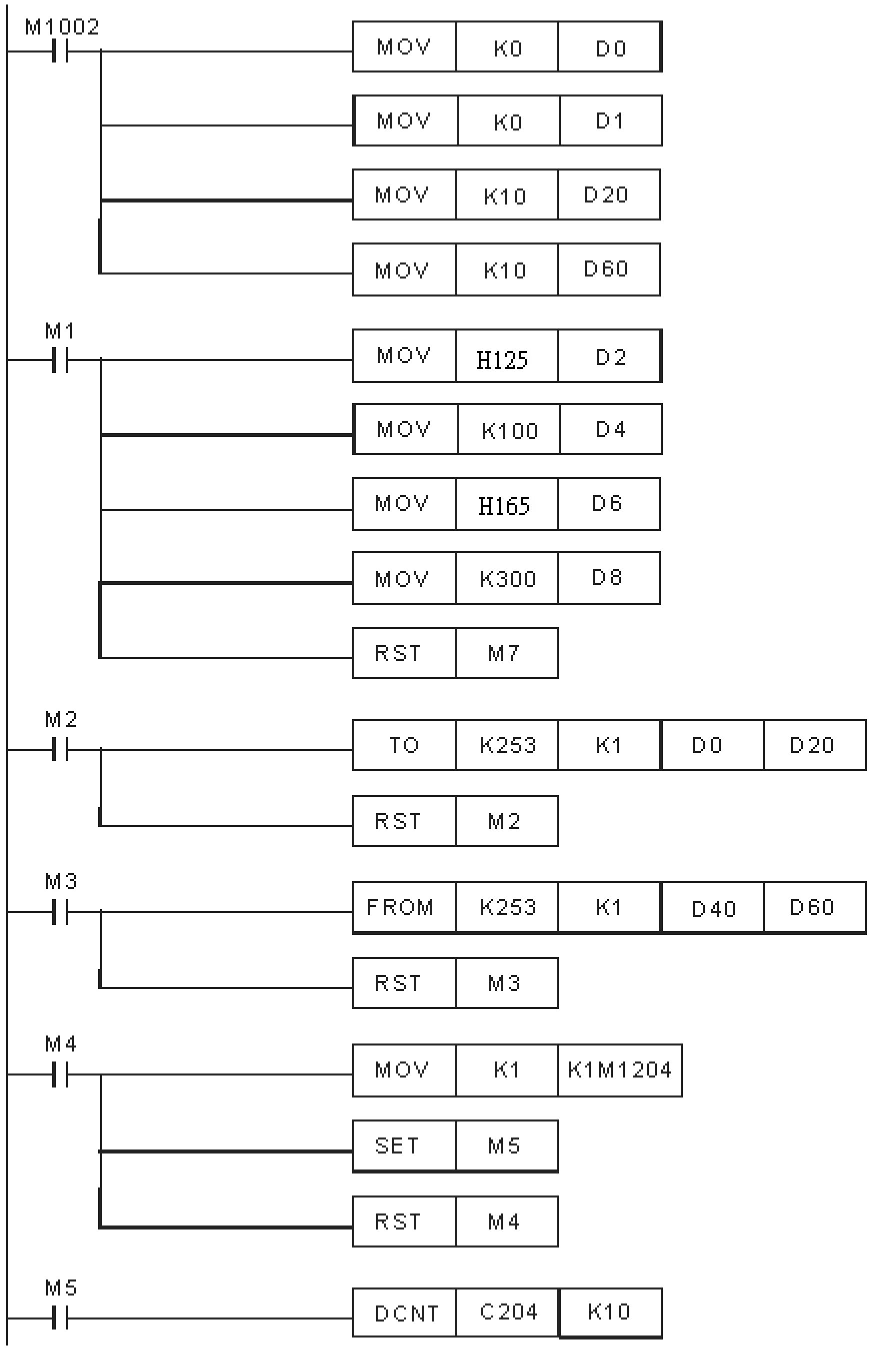


MonitorTable

Device No.	Radix	Value	Comment
C204	d32u	95	
D44	d32u	100	
D48	d32u	300	

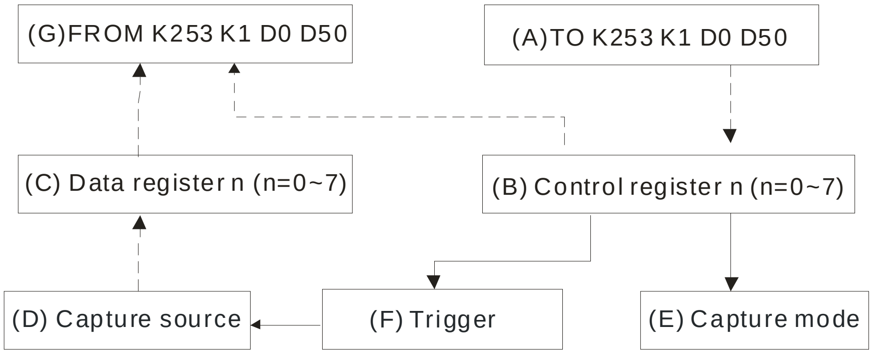
7. Use the manual pulse generator. Check whether X7 on the DVP-20PM series motion controller used is ON when the value in C204 is greater than 100. If X7 is ON, Y1 is set to ON.
8. Use the manual pulse generator. Check whether X7 on the DVP-10PM series motion controller used is OFF when the value in C204 is greater than 300. If X7 is OFF, Y1 is reset to OFF.

【Program in PMSoft】



12.3 High-speed Capture

A deviation often occurs when the present position of an axis or the value in C200/C204 is read. To prevent a deviation from occurring, users read a value immediately by setting an input terminal to ON. Capture is described below.



Block (A): The instruction TO is used to write data into control registers (block B).

Block (B): Users set a capture source (block D), set bit 5~bit 4 to 0 (block E), and set a trigger (block F) in a control register.

Block (C): The capture of a value (block D) is triggered by an input terminal, and the value captured is stored in data registers.

Block (D): The present positions of three axes, the values in C200 and C204 are capture sources.

Block (E): Capture mode

Block (F): External trigger

Block (G): The instruction FROM is used to read data from control registers (block C) and data registers (block B). The values stored in the data registers are values captured.

Procedure for a high-speed capture: The instruction TO is used to write data into control registers (block A).→An input terminal is set to ON (block F).→The present position of the X-axis/Y-axis/Z-axis, or the value in C200/C204 is captured (block D). The value captured is stored in data registers (block C).→Users read the value captured by means of the instruction FROM.

■ Example

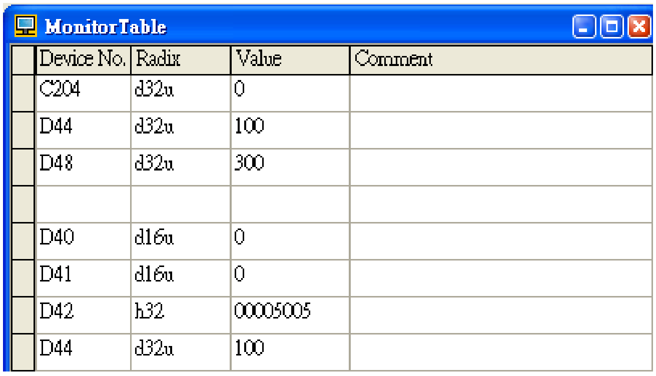
【Description】

Start the high-speed counter C204. The value in C204 is captured when X5 is set to ON. A manual pulse generator is used, and is connected to A1± and B1±.

【Steps】

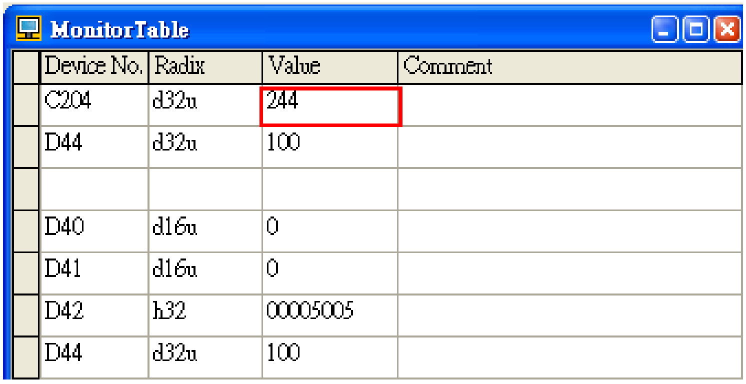
1. When M1002 in O100 is ON, the initial setting of high-speed capture is carried out.
   (1) D0=0→Initial group number n=0
   (2) D1=0
   (3) D20=6→Writing 6 values by means of the instruction TO (Only one value is captured.)
   (4) D60=10→Reading 6 values by means of the instruction FROM (Only one value is captured.)
2. When M1 is ON, the high-speed capture is set.
   (1) The value in (D3, D2) is H5005.→The capture source set is C204. (The value of bit 3~bit 0 is 5). The mode selected is a capture mode. (The value of bit 5~bit 4 is 0.) The trigger selected is X5. (The value of bit 15~bit 12 is 5.)
   (2) The value in (D5, D4) is K100. Users can set (D5, D4) by themselves.
3. The high-speed capture is started when M2 is ON.

4. The setting of the high-speed capture is read when M3 is ON.



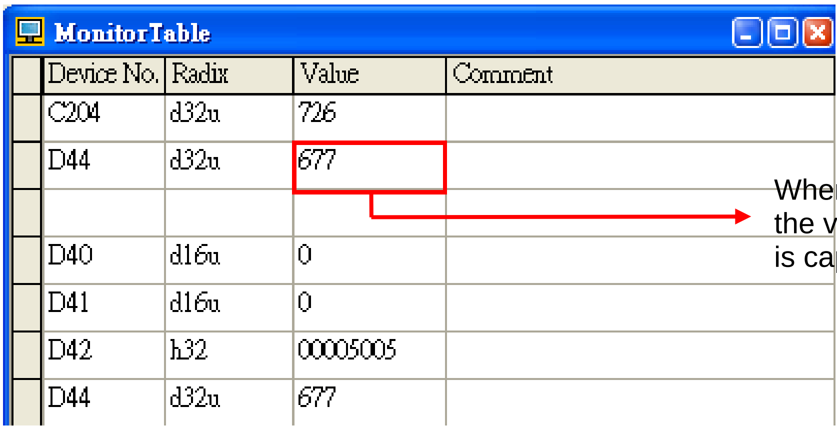
Device No.	Radix	Value	Comment
C204	d32u	0	
D44	d32u	100	
D48	d32u	300	
D40	d16u	0	
D41	d16u	0	
D42	h32	00005005	
D44	d32u	100	

5. When M4 is ON, K1 is moved to M1204~M1207. C204 is started when M5 is set to ON. (Mode of counting: Pulse/Direction)
6. Use a manual pulse generator, and check whether C204 counts.



Device No.	Radix	Value	Comment
C204	d32u	244	
D44	d32u	100	
D40	d16u	0	
D41	d16u	0	
D42	h32	00005005	
D44	d32u	100	

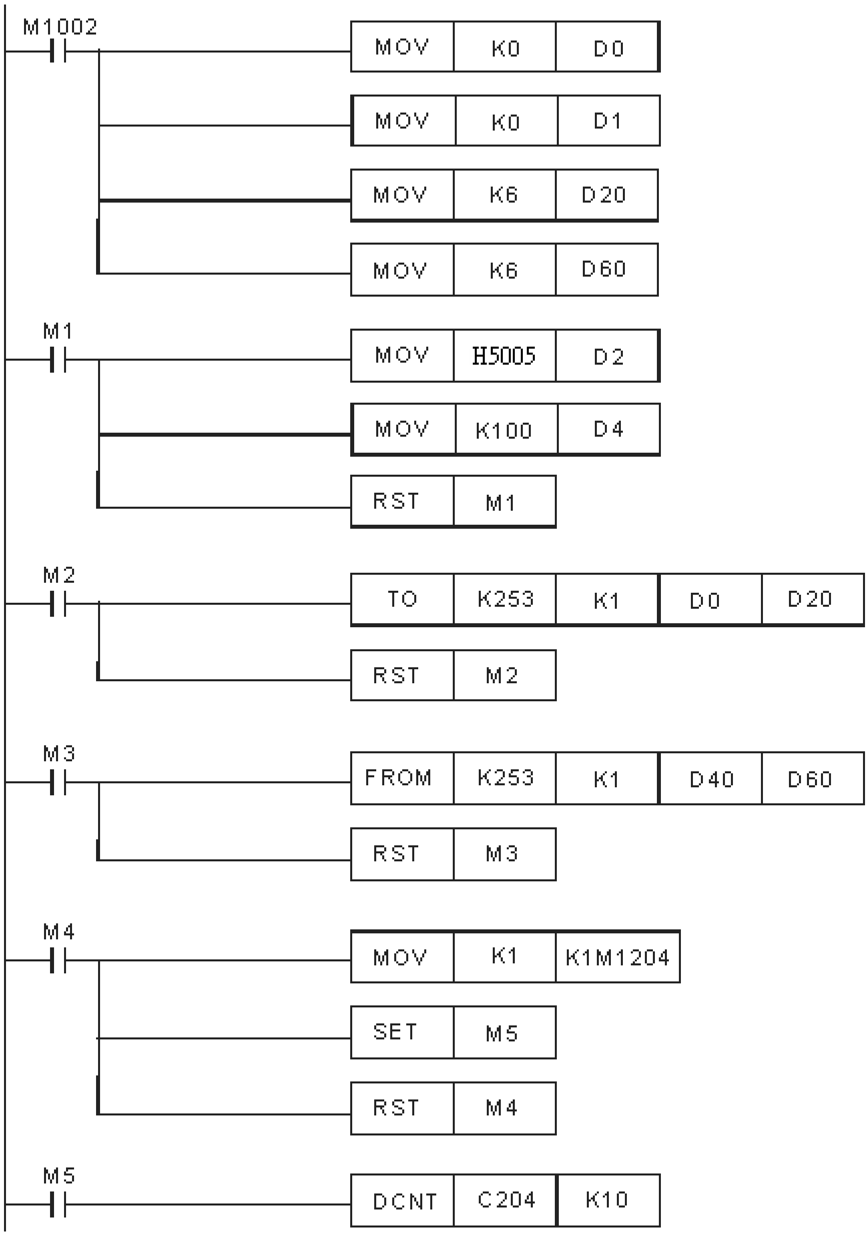
7. Use the manual pulse generator, and set X5 to ON.
8. The value captured is read when M3 is ON. When X5 is ON, the value in C204 is captured. The value captured is 677.



Device No.	Radix	Value	Comment
C204	d32u	726	
D44	d32u	677	
D40	d16u	0	
D41	d16u	0	
D42	h32	00005005	
D44	d32u	677	

When X5 is ON, the value in C204 is captured.

【Program in PMSoft】



13.1 Appendix A: Error Code Table

After a program is written into a DVP-20PM series motion controller, the ERROR LED indicator will blink and an error flag will be ON if an error occurs in O100 or an Ox motion subroutine. The reason for the error occurring in O100 or an Ox motion subroutine may be that the use of operands (devices) is incorrect, syntax is incorrect, or the setting of motion parameters is incorrect. Users can know the reasons for the errors occurring in a DVP-20PM series motion controller by means of the error codes (hexadecimal codes) stored in error registers.

- Error message table

Program block	**O100**			
Error type	**Program error**	**Motion error**		
		X-axis	**Y-axis**	**Z-axis**
Error flag	M1953	M1793	M1873	M2033
Error register	D1802	D1857	D1937	D2017
Step number	D1803	D1869		
Program block	**Ox**			
Error type	**Program error**	**Motion error**		
		X-axis	**Y-axis**	**Z-axis**
Error flag	M1793	M1793	M1873	M2033
Error register	D1857	D1857	D1937	D2017
Step number	D1869	D1869		

- Program error codes and motion error codes (hexadecimal codes)

Error code	Description	Error code	Description
0002	The subroutine used has no data.	0031	The positive-going pulses generated by motion are inhibited.
0003	CJ, CJN, and JMP have no matching pointers.	0032	The negative-going pulses generated by motion are inhibited.
0004	There is a subroutine pointer in the main program.	0033	The motor used comes into contact with the left/right limit switch set.
0005	Lack of a subroutine	0040	A device exceeds the device range available.
0006	A pointer is used repeatedly in the same program.	0041	A communication timeout occurs when MODRD/MODWR is executed.
0007	A subroutine pointer is used repeatedly.	0044	An error occurs when a device is modified by a 16-bit index register/32-bit index register.
0008	The pointer used in JMP is used repeatedly in different subroutines.	0045	The conversion into a floating-point number is incorrect.
0009	The pointer used in JMP is the same as the pointer used in CALL.	0E18	The conversion into a binary-coded decimal number is incorrect.
000A	A pointer is the same as a subroutine pointer.	0E19	Incorrect division operation (The divisor is 0.)
0011	Target position (I) is incorrect.	C401	General program error
0012	Target position (II) is incorrect.	C402	LD/LDI has been used more than nine times.
0021	Velocity (I) is incorrect.	C404	There is more than one nested program structure supported by RPT/RPE.
0022	Velocity (II) is incorrect.	C405	SRET is used between RPT and RPE.
0023	The velocity (V_{RT}) of returning home is incorrect.	C4EE	There is no M102 in the main program, or there is no M2 in a motion subroutine.
0024	The velocity (V_{CR}) to which the velocity of the axis specified decreases when the axis returns home is incorrect.	C4FF	A wrong instruction is used, or a device used exceeds the range available.
0025	The JOG speed set is incorrect.		

MEMO