Smarter. Greener. Together.

Industrial Automation Headquarters

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

Asia

Delta Electronics (Shanghai) Co., Ltd.
No.182 Minyu Rd., Pudong Shanghai, P.R.C.
Post code : 201209
TEL: 86-21-6872-3988 / FAX: 86-21-6872-3996
Customer Service: 400-820-9595

Delta Electronics (Japan), Inc.
Tokyo Office
Industrial Automation Sales Department
2-1-14 Shibadaimon, Minato-ku
Tokyo, Japan 105-0012
TEL: 81-3-5733-1155 / FAX: 81-3-5733-1255

Delta Electronics (Korea), Inc.
Seoul Office
1511, 219, Gasan Digital 1-Ro., Geumcheon-gu,
Seoul, 08501 South Korea
TEL: 82-2-515-5305 / FAX: 82-2-515-5302

Delta Energy Systems (Singapore) Pte Ltd.
4 Kaki Bukit Avenue 1, #05-04, 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

Delta Electronics (Thailand) PCL.
909 Soi 9, Moo 4, Bangpoo Industrial Estate (E.P.Z),
Pattana 1 Rd., T.Phraksa, A.Muang,
Samutprakarn 10280, Thailand
TEL: 66-2709-2800 / FAX : 662-709-2827

Delta Energy Systems (Australia) Pty Ltd.
Unit 20-21/45 Normanby Rd., Notting Hill Vic 3168, Australia
TEL: 61-3-9543-3720

Americas

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

Delta Greentech (Brasil) S/A
São Paulo Office
Rua Itapeva, 26 – 3° Andar - Bela Vista
CEP: 01332-000 – São Paulo – SP - Brasil
TEL: 55-11-3530-8642 / 55-11-3530-8640

Delta Electronics International Mexico S.A. de C.V.
Mexico Office
Vía Dr. Gustavo Baz No. 2160, Colonia La Loma,
54060 Tlalnepantla Estado de Mexico
TEL: 52-55-2628-3015 #3050/3052

EMEA

Headquarters: **Delta Electronics (Netherlands) B.V.**
Sales: Sales.IA.EMEA@deltaww.com
Marketing: Maketing.IA.EMEA@deltaww.com
Technical Support: iatechnicalsupport@deltaww.com
Customer Support: Customer-Support@deltaww.com
Service: Service.IA.emea@deltaww.com
TEL: +31(0)40 800 3800

BENELUX: **Delta Electronics (Netherlands) B.V.**
De Witbogt 20, 5652 AG Eindhoven, The Netherlands
Mail: Sales.IA.Benelux@deltaww.com
TEL: +31(0)40 800 3800

DACH: **Delta Electronics (Netherlands) B.V.**
Coesterweg 45, D-59494 Soest, Germany
Mail: Sales.IA.DACH@deltaww.com
TEL: +49(0)2921 987 0

France: **Delta Electronics (France) S.A.**
ZI du bois Challand 2, 15 rue des Pyrénées,
Lisses, 91090 Evry Cedex, France
Mail: Sales.IA.FR@deltaww.com
TEL: +33(0)1 69 77 82 60

Iberia: **Delta Electronics Solutions (Spain) S.L.U**
Ctra. De Villaverde a Vallecas, 265 1° Dcha Ed.
Hormigueras – P.I. de Vallecas 28031 Madrid
TEL: +34(0)91 223 74 20
C/Llull, 321-329 (Edifici CINC) | 22@Barcrelona, 08019 Barcelona
Mail: Sales.IA.Iberia@deltaww.com
TEL: +34 93 303 00 60

Italy: **Delta Electronics (Italy) S.r.l.**
Ufficio di Milano Via Senigallia 18/2 20161 Milano (MI)
Piazza Grazioli 18 00186 Roma Italy
Mail: Sales.IA.Italy@deltaww.com
TEL: +39 02 64672538

Russia: **Delta Energy System LLC**
Vereyskaya Plaza II, office 112 Vereyskaya str.
17 121357 Moscow Russia
Mail: Sales.IA.RU@deltaww.com
TEL: +7 495 644 3240

Turkey: **Delta Greentech Elektronik San. Ltd. Sti. (Turkey)**
Şerifali Mah. Hendem Cad. Kule Sok. No:16-A
34775 Ümraniye – İstanbul
Mail: Sales.IA.Turkey@deltaww.com
TEL: + 90 216 499 9910

GCC: **Delta Energy Systems AG (Dubai BR)**
P.O. Box 185668, Gate 7, 3rd Floor, Hamarain Centre
Dubai, United Arab Emirates
Mail: Sales.IA.MEA@deltaww.com
TEL: +971(0)4 2690148

Egypt + North Africa: **Delta Electronics**
511 Cairo Business Plaza, North 90 street,
New Cairo, Cairo, Egypt
Mail: Sales.IA.MEA@deltaww.com

*We reserve the right to change the information in this manual without prior notice.

IPC Motion Platform User Guide

IPC Motion Platform User Guide

www.deltaww.com

Smarter. Greener. Together.

Preface

Thank you for purchasing this product. This user guide provides information about the IPC Motion Platform (IMP).

This user guide includes:

- The installation and authorization of the IMP software.
- The operating instructions of the settings and IMP Quick start interface.
- The instructions of IMP Ladder Editor.
- Descriptions of the PLC instructions used by the IMP.
- Descriptions of the Motion Program Macro (MPM) used by the IMP.
- Descriptions of the parameters.
- Troubleshooting.

Product features:

The IMP (IPC Motion Platform) software design is the integration of PLC syntax, graphical HMI, and IPC floating-point computing, expanding the servo motion control on the basis of high-performance computing units. Through Delta DMCNET or EtherCAT motion communication bus, the IMP is capable of controlling up to 36 axes of servo units and conducting multi-axis coordinated motions. With the introduction of the Soft Numerical Control (SNC) motion control functions, the difficulty of developing NC series products is significantly reduced. You can develop process-related functions and programs based on the actual industry applications. Additionally, by connecting the visual system, sensor, and PC-based central control system through the Ethernet and serial communication interface, the IMP is integrated into a comprehensive industrial control network.

How to use this user guide:

You can use this user guide as a reference when installing, setting up, authenticating, and using the IMP.

Delta technical services:

Please consult your Delta equipment distributor or Delta Customer Service Center if you encounter any problems.

Safety precautions

The IMP software and industrial PC are integrated as the motion controller; therefore, for the industrial PC host with the IMP installed and other applications, the peripheral devices should comply with the specifications of each product.

Pay special attention to the following safety precautions at all times during inspection, installation, wiring, operation, maintenance, and examination of the machines.

The symbols of “DANGER”, “WARNING”, and “STOP” indicate:

DANGER	**Danger. May cause severe or fatal injuries to personnel if the instructions are not followed.**
WARNING	**Warning. May cause moderate injury to personnel, or lead to severe damage or even malfunction of the product if the instructions are not followed.**
STOP	**Absolutely prohibited activities. May cause serious damage or even malfunction of the product if the instructions are not followed.**

Operation

WARNING

- Before operating, make sure the parameter settings are set according to the application. If the parameters are not adjusted to the correct values, it may lead to malfunction of the machine or the operation may be out of control.
- Ensure you can activate the emergency stop before operating the machine.
- When applying power, make sure the motor is not rotating because of inertia of the machine or other causes.

STOP

During the operation, do not touch any rotating motor parts, or it may cause personnel injury.

DANGER

- To avoid accidents, remove all units, including the couplings and belts, during the first test run, so the machine operates without any load.
- If you fail to operate the machine properly after the servo motor connects to the machine, it may damage the machine and lead to personnel injury.
- In order to reduce the danger, make sure the servo motor can operate normally without load. Then try operating the motor with load.

Note: the content of this user guide may be revised without prior notice, please download the latest version from Delta's website (http://www.delta.com.tw/industrialautomation/).

Table of Contents

4 Ladder Editor

5 Memory Device

How to Use Programming Language?

6 Logic Editing

Appendix

1 Overview

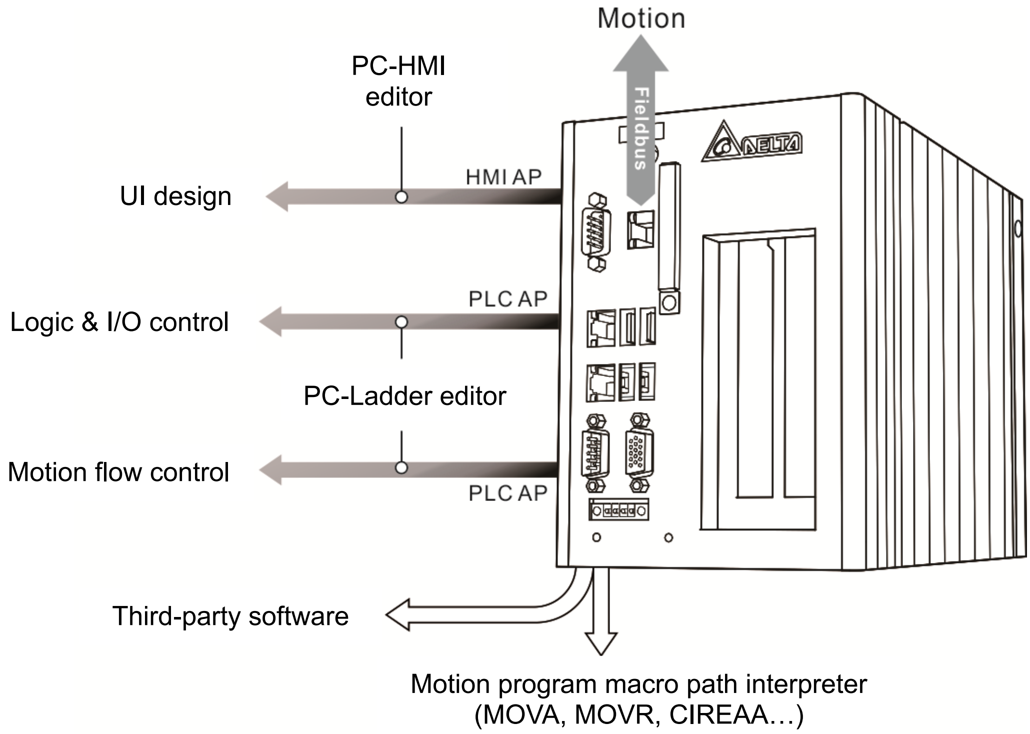
Before applying the IPC Motion Platform to the PLC, you can read this chapter to understand its operating architecture.

1

1.1 Introduction to IPC motion platform

IMP (IPC Motion Platform) is an application development platform based on high-speed motion control fieldbus. By integrating the features of Delta HMI editing software and PLC compiling software, you can simply use Delta's DOPSoft to design the software operation screen, logic control, and motion control.

You can install the IMP software kernel in Delta PAC or IPC that supports Delta motion cards, which upgrades your PAC or IPC to a high-speed communication motion controller. Unlike traditional controllers which issue servo control commands by sending pulses and analog signals, IMP uses DMCNET or EtherCAT communication fieldbus architecture to connect servo drives or remote modules in series with the configuration of PLC motion commands. For multi-axes motion applications, IMP provides various control functions, including 3-axis linear interpolation, helical and arc motion controls, for further development.



1.2 Architecture of IPC motion platform

The IMP controller software is a PC-based program with multitasking programming. It can execute the PLC and HMI to achieve high-performance integration of the operation interface and logic control, and provides optional Soft Numerical Control (SNC) function.

The IMP controller program types include PLC main program (Cyclic Task), Sub Program, and Motion Program Macro (MPM). With DMCNET or EtherCAT real-time fieldbus, IMP is built-in with up to 35 homing modes and 36-axis motion control which supports speed, torque, position, 3-axis helical and linear interpolation motion commands, incremental type and absolute type commands, and S- and T-curve functions.

By applying the built-in PLC software version and PC computing to assist the motion axis cards, IMP realizes the complex algorithm, including G-code interpreter, short path fitting, and reversal of original path. You can also use the PLC to set the user-defined M-code, T-code, and the operation logic for the machine.

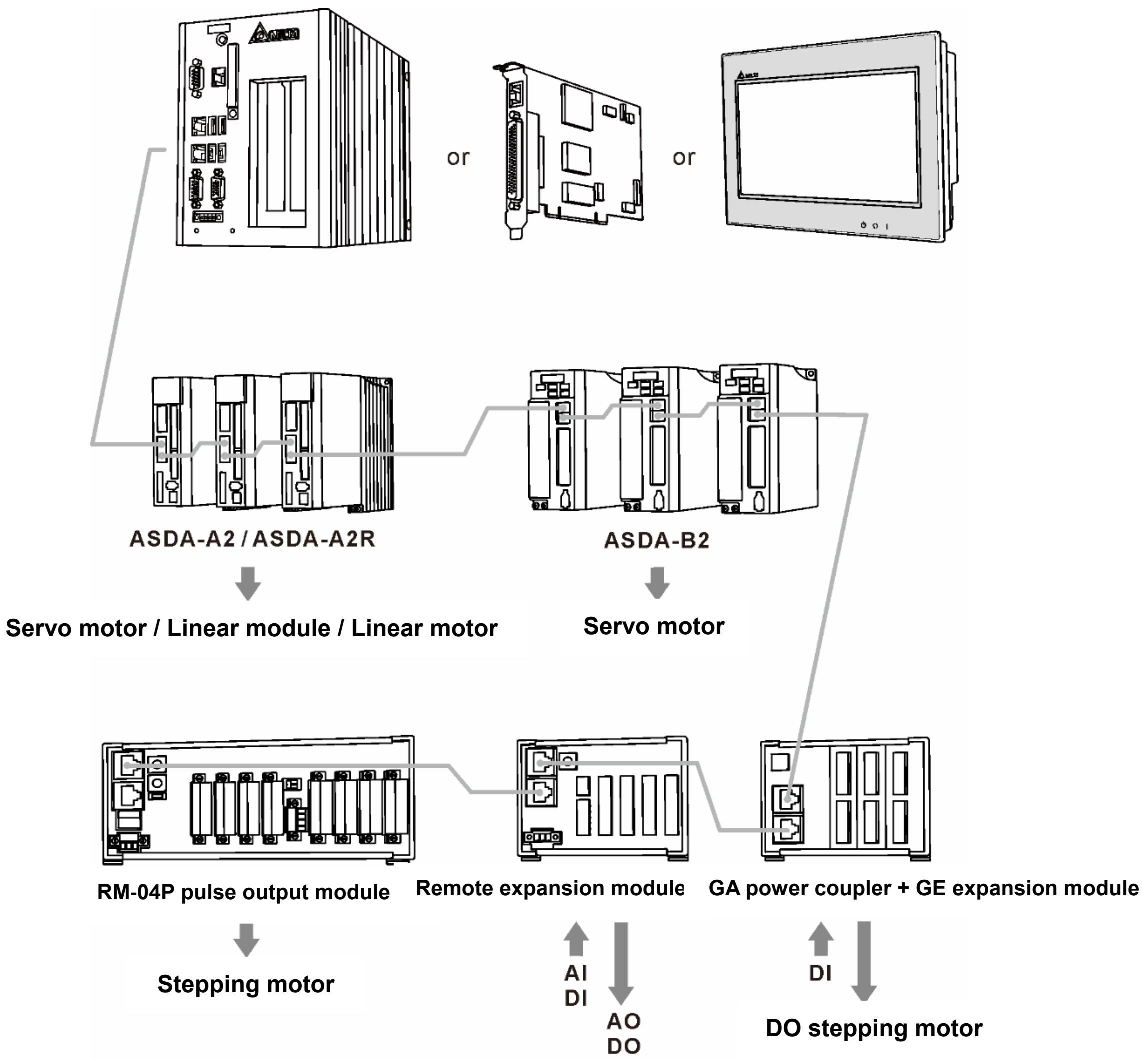


Figure 1.2.1 DMCNET architecture diagram

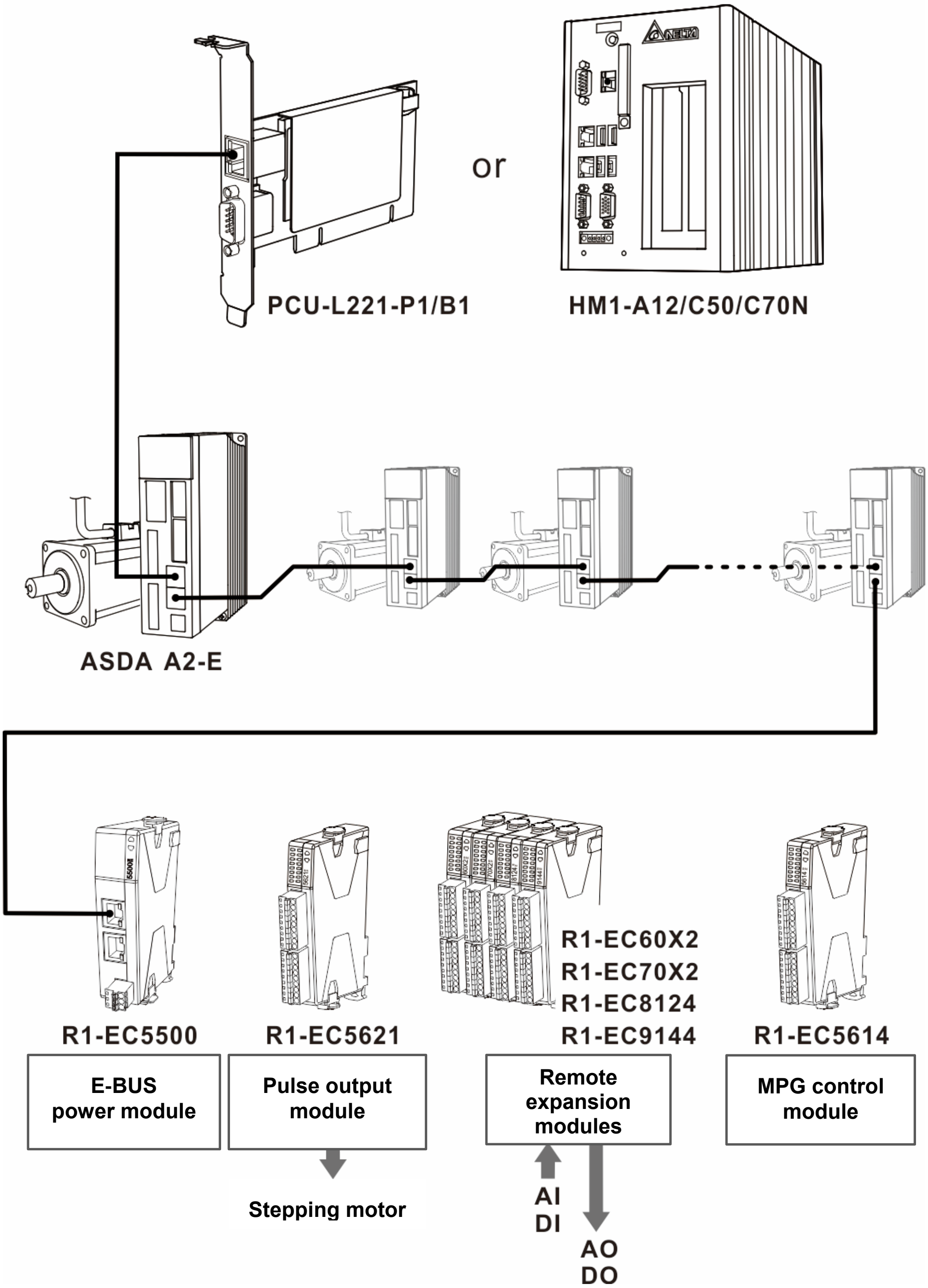


Figure 1.2.2 EtherCAT communication architecture diagram

Installation 2

This chapter provides the installation steps for the IPC Motion Platform (IMP) kernel and the instructions to acquire the software license authorization.

2

2.1 System requirements

For the best user experience, it is suggested to use this software with Delta's MH1 and MP1 series PACs. To work with other hardware, refer to the following system requirements and use with Delta's DMCNET or EtherCAT motion control cards.

System requirements	
Operation system	Windows 7 Windows 7 Embedded Windows 10
CPU	Dual core CPU 1.2 GHz [*1]
Memory	2G
Hard disk space	1GB [*2]
Display	Resolution of 1024 x 600
Human system interface	Mouse or touchscreen
System environment	.NET Framework 4.0
Others	Delta's DMCNET or EtherCAT motion control cards and PAC series products

Note:

*1: the operation of the SNC computing kernel requires CPU, so it is suggested to select a PC or an IPC equipped with three or more sets of cores. To use two sets of SNC functions, you must select models of four or more sets of cores with two sets of motion buses. If you use three sets of SNC functions, you need to select models of five or more sets of cores with three sets of motion buses...and so on.

*2: the PC or IPC for installing IMP must have the D hard disk drive.

The IMP software kernel can operate with the following DMCNET models:

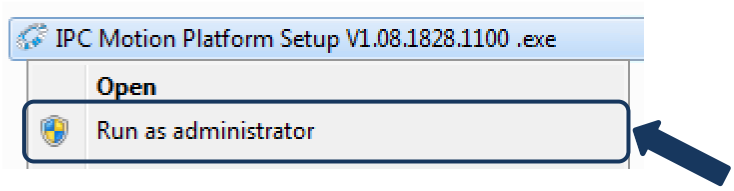
		DMCNET models						
		MH1 - C70D	MH1- C50D	MH1- A12D	MP1- A10D	PCI- DMC- B01	PCI- DMC- A02	PCI- DMC-F02
System	CPU	Intel Core i7-3612 QE Quad Core 2.1 GHz	Intel Core i5-3610 ME Dual Core 2.7 GHz	Intel Atom E3845 Quad Core 1.91 GHz	Intel Atom E3825 Dual Core 1.33 GHz	-	-	-
Operation axes		12	12	12	6	12	12	6
Maximum slaves		12	12	12	12	12	12	12
On board I/O		1/1	1/1	1/1	8/4	1/1	32/24	32/24
Compare		2	2	-	-	2	-	-
SNC		△	△	△	△	△	△	-
Remote modules	DI: RM32MN/ M64MN	△	△	△	△	△	△	△
	DO: RM32MT/ RM64MT	△	△	△	△	△	△	△
	DO / DI: RM32PT/ RIO3232RT5	△	△	△	△	△	△	△
	Stepping module: RM04PI*1	△	△	△	△	△	△	△
	AD / DA: RM04AD/ RM04DA	△	△	△	△	△	△	△
Gateway modules	Pulse module GE01PH	△	△	△	△	△	△	△
Servo motor	ASD-A2-****-F	△	△	△	△	△	△	△
	ASD-B2-****-F	△	△	△	△	△	△	△
	ASD-M-****-F	△	△	△	△	△	△	△

2

2.2 Installation steps

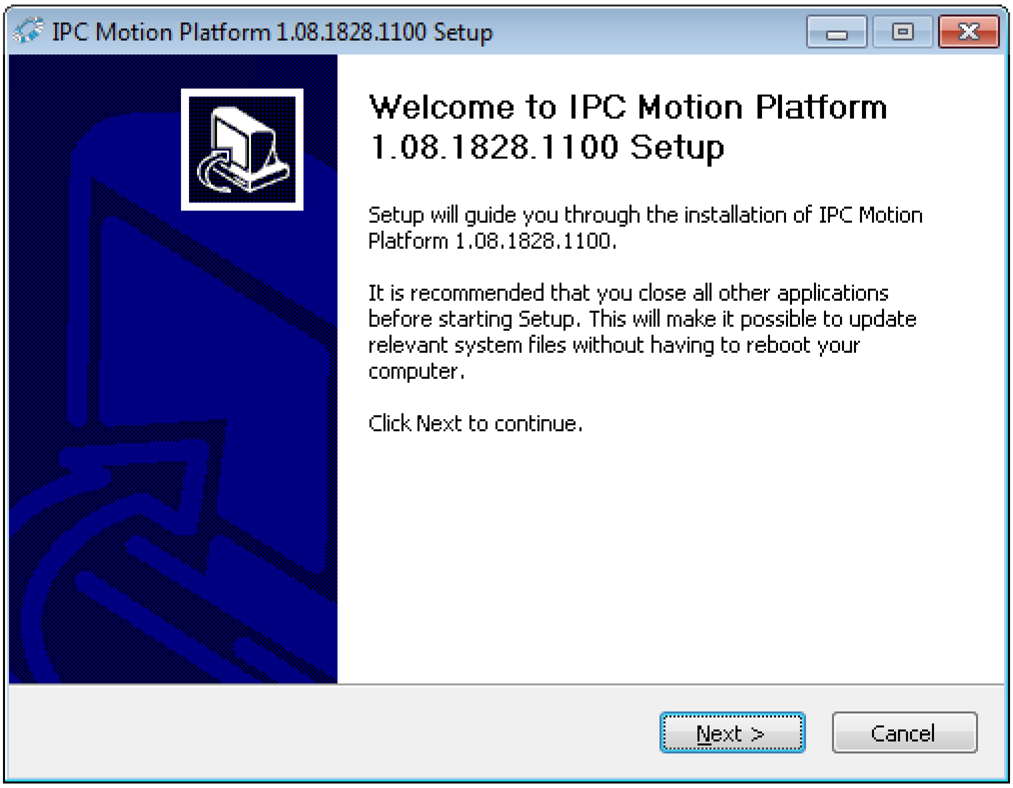
After obtaining the IMP kernel installation file, follow the steps below to install the software:

1. Execute IMP Setup.exe as the Administrator.
   a. In Windows XP, double-click IMP Setup.exe.
   b. In Windows 7 or later version, right-click IMP Setup.exe and select **Run as administrator**.



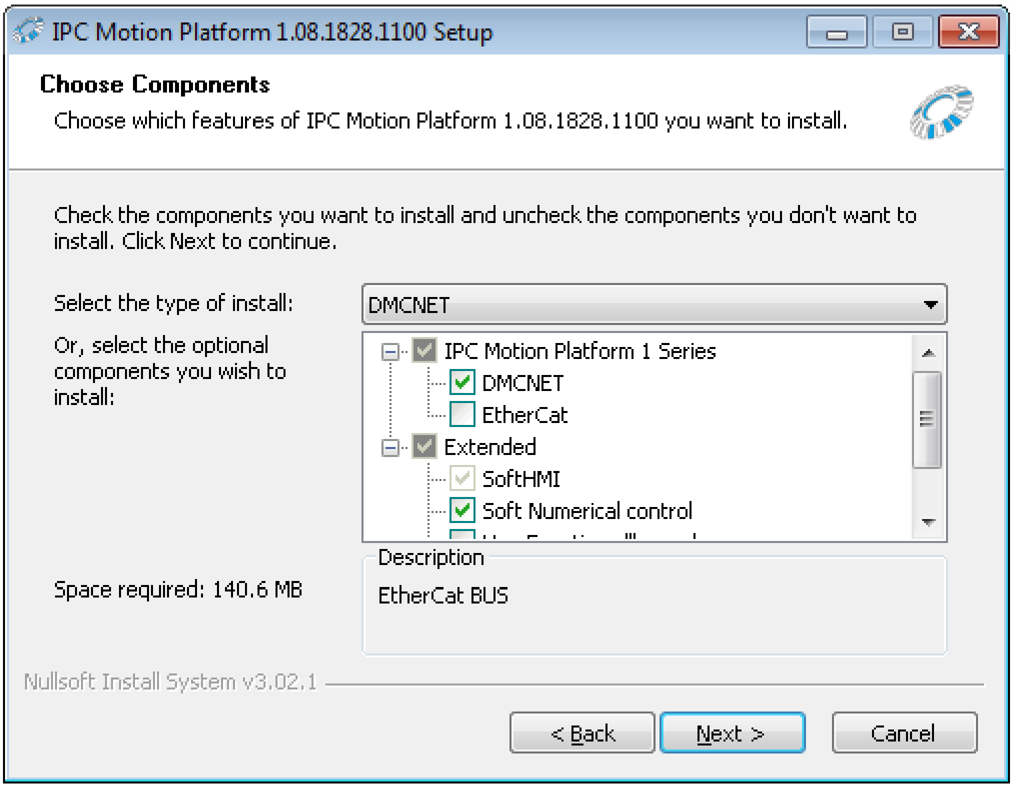
2. After selecting the installation language, click **OK**.

3. When you are on the welcome screen, click **Next**.



4. Check the components you want to install and click **Next**.

   (The kernel of the IPC Motion Platform will be installed in the following fixed directory D:\NandFlash\IPC Motion Platform.)



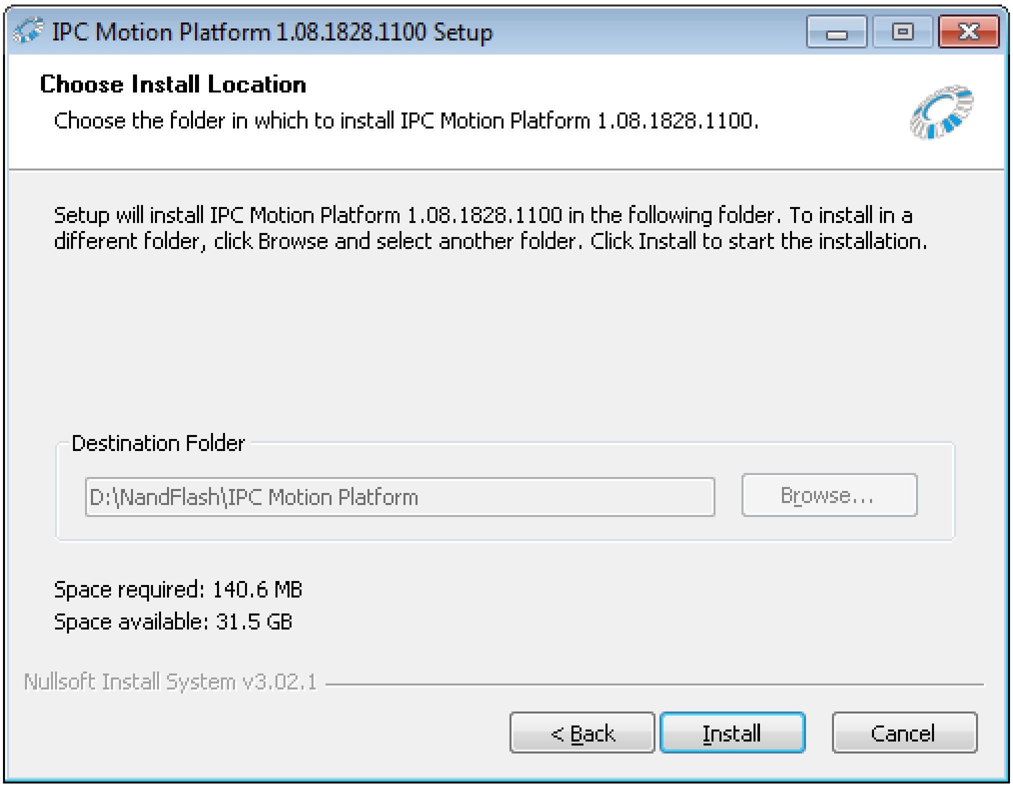
DMCNET / EtherCAT: you can only select one option according to the communication bus type of the IPC.

SoftHMI: check this item to install the HMI and PLC interpreter. The interpreter executes the interface screen and PLC logic program edited by DOPSoft.

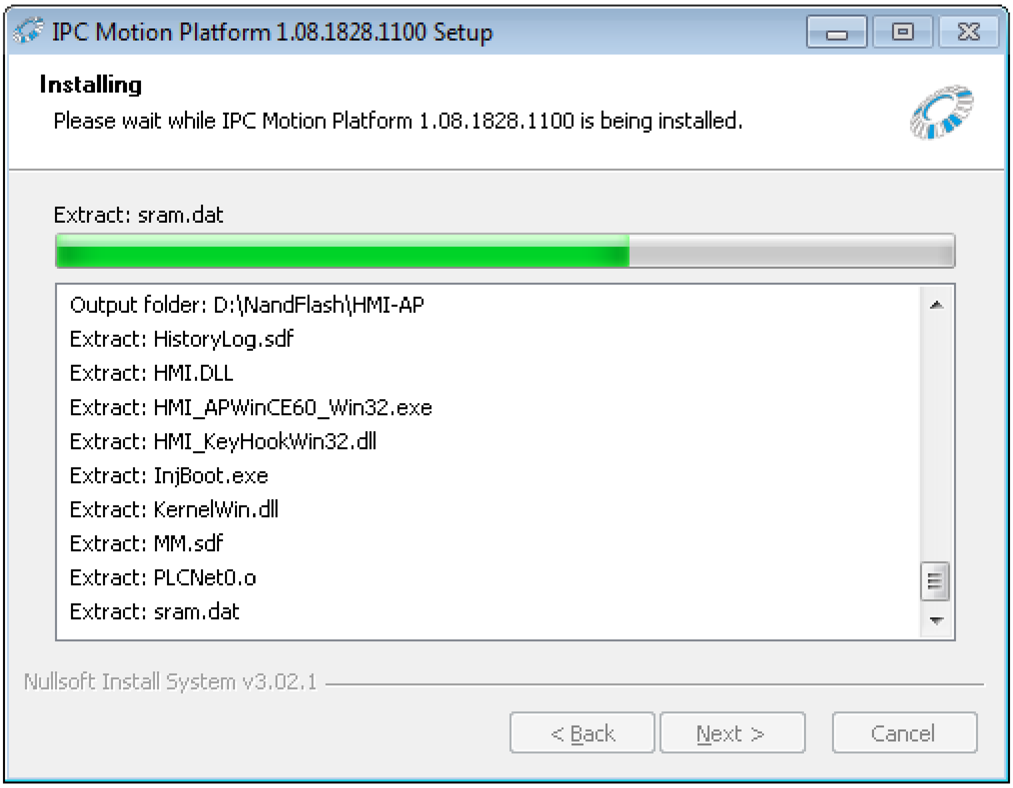
Soft Numerical control: check this item to install the G-code interpreter for value control.

2

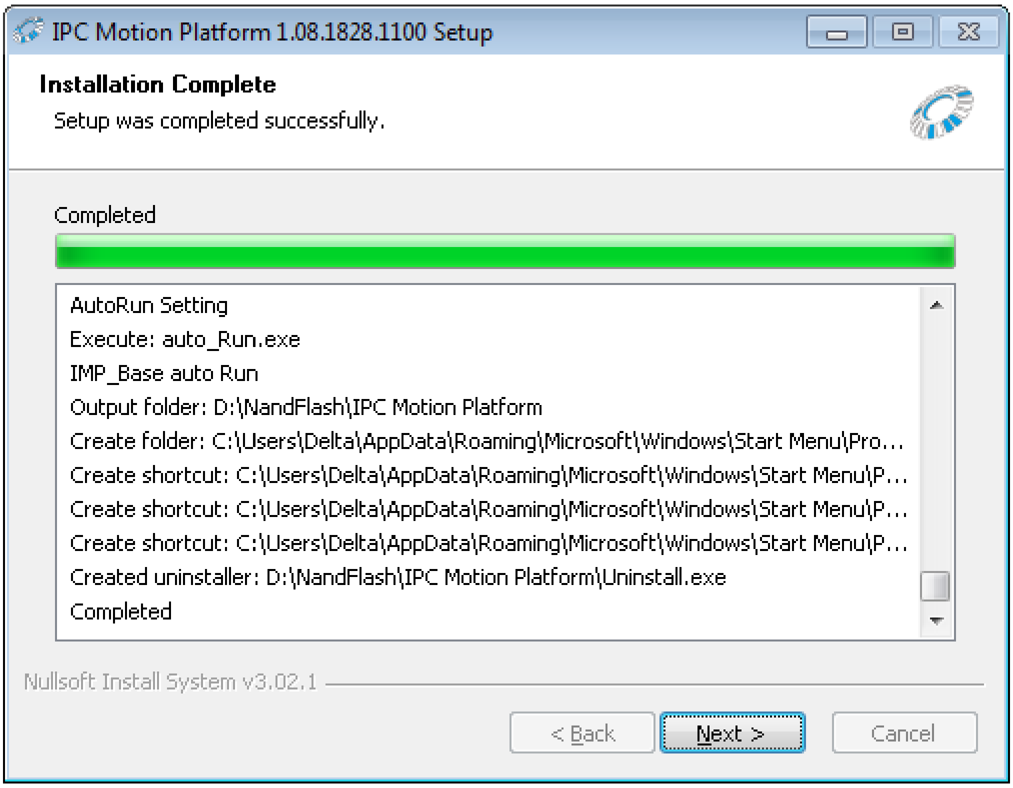
5. Select installation path.



6. Go to the IMP kernel installation screen.

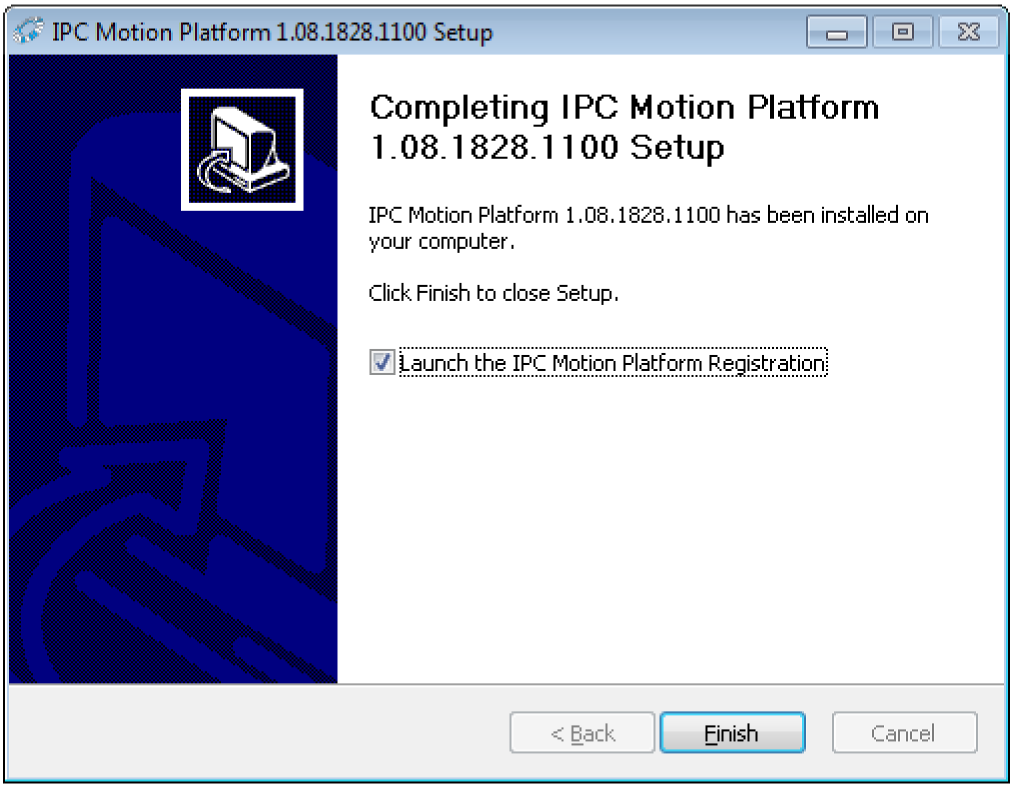


7. When the installation is completed, click **Next**.



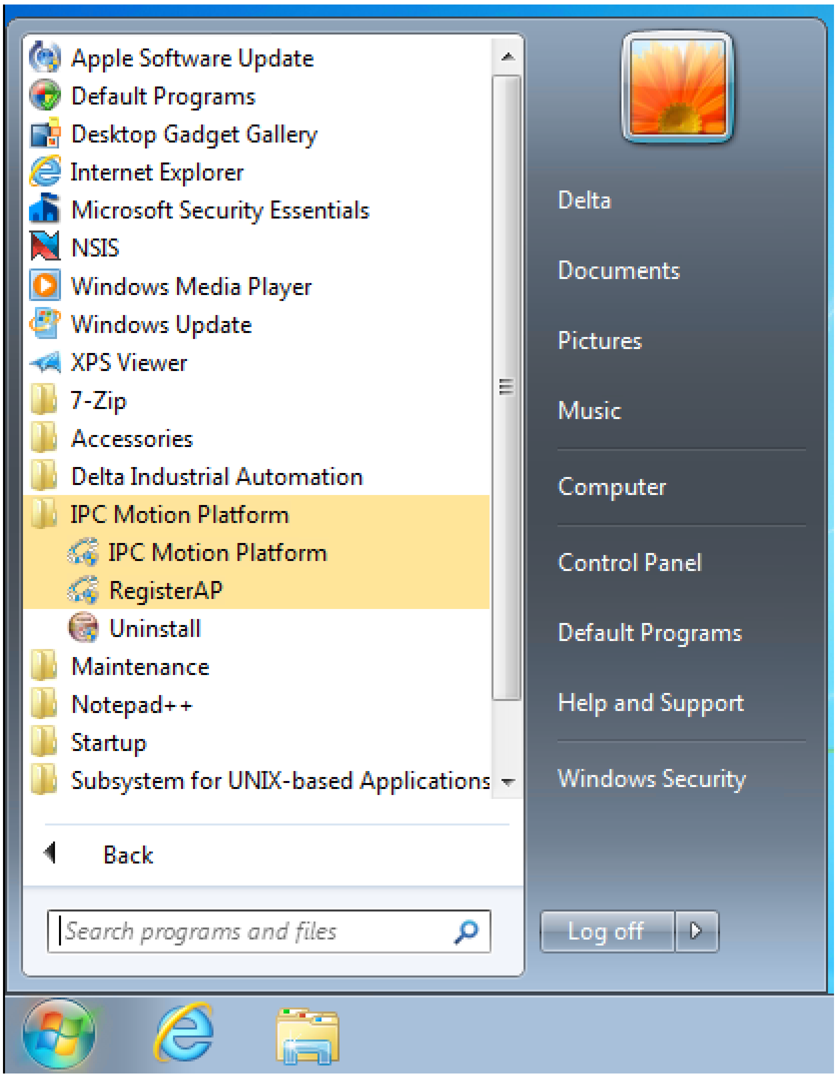
8. Click **Finish** to complete the installation.

   If you check [Launch the IPC Motion Platform Registration], the license authorization software RegisterAP launches once the installation is completed.



2

After the IMP kernel is installed, you can find the installed kernel and relevant files in the following fixed directory D:\NandFlash\IPC Motion Platform. As in the figure below, you can the find the execution shortcut in the following path: [Start] > [All programs] > [IPC Motion Platform].



2

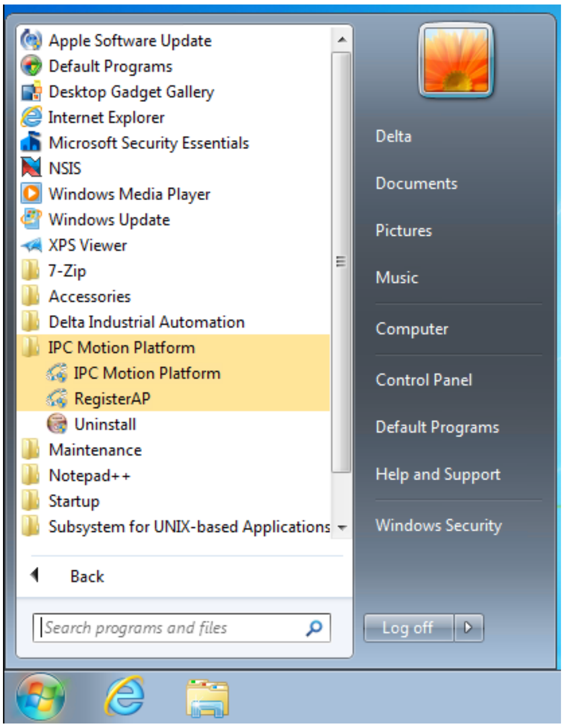
2.3 License authorization

The IMP software kernel is required to operate with Delta's motion control card and in Delta's PAC hardware environment; therefore, you need to obtain license authorization for each PAC by following the steps below. If the license authorization is completed for the IMP software contained in your purchased PAC, then you can skip this section.

Note: once the authorization is completed, do not change the hardware environment. For example, adding / removing motion control cards, motherboard, hardware, network interface, etc. may result in authorization failure and you will be unable to execute the program.

Follow the steps below to complete the license authorization:

1. Execute RegisterAP by Method A or Method B.
   In Windows XP, double-click RegisterAP.exe.
   In Windows 7 or later version, right-click RegisterAP.exe and select **Run as administrator**.

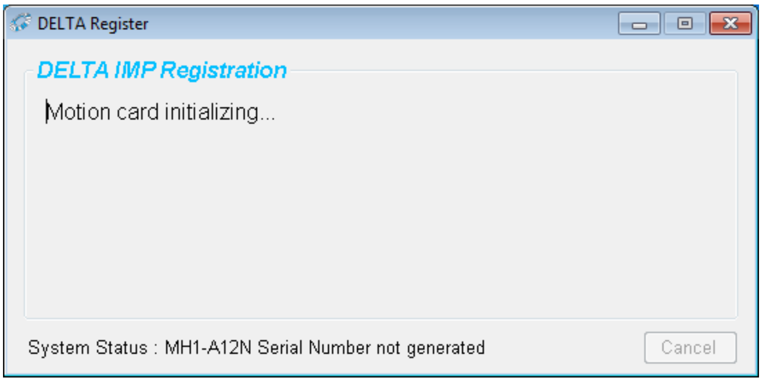
<table>
<tr><td>Method A</td><td>Go to D:\NandFlash\IPC Motion Platform\IMP base\RegisterAP, then execute  RegisterAP.exe.</td></tr>
<tr><td>Method B</td><td>Go to [Start] > [All programs] > [IPC Motion Platform] > [RegisterAP] to execute  RegisterAP.exe.

</td></tr>
</table>

2. Once you open the software, the system status is shown on the bottom of the window. If the system has not obtained the license serial number before, **Activate License** is grayed out with its function disabled.

3. Click **Generate Serial Number**.

4. Enter the product serial number of the motion control card or PAC hardware. You can find the product serial number on the sticker on the outside of the product; make sure the hardware serial number is correct. If you install multiple cards, enter the serial number of the smallest card number.

5. Wait for the registration code to generate.

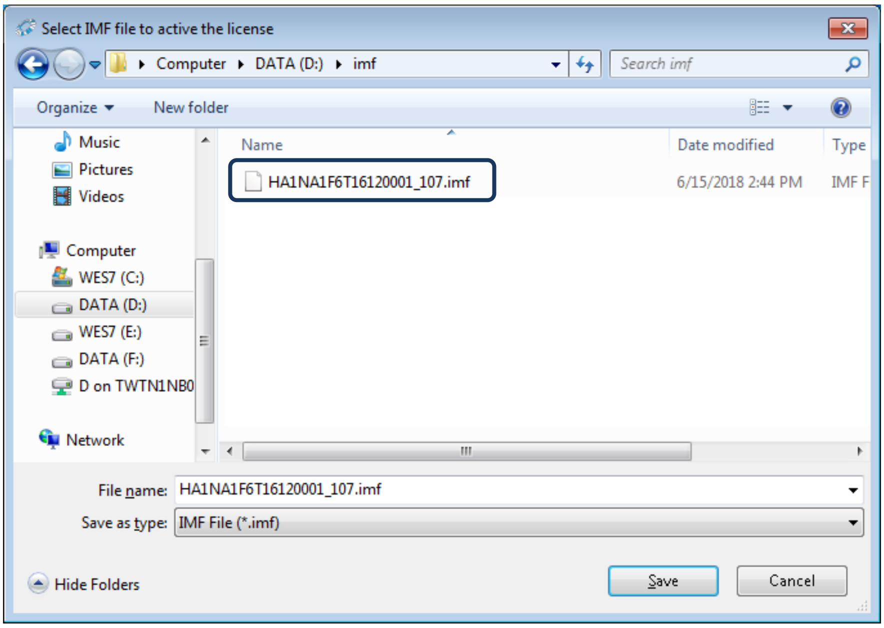


6. When you obtain the registration code, the system status will display the hardware and authorization status.

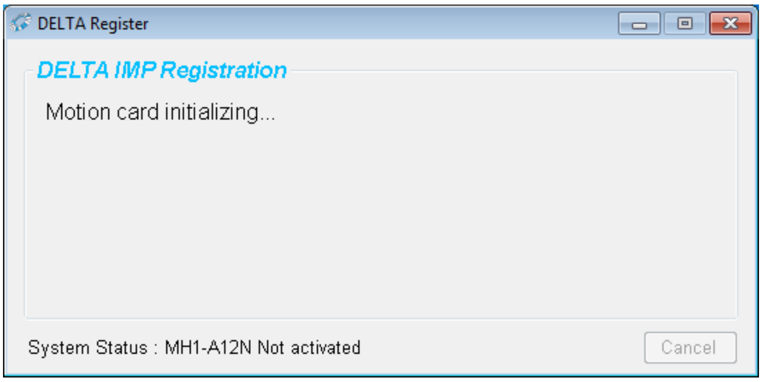
7. Save the registration code and send it to Delta to obtain the license file.
   Click **Save SN** to save this registration code to the selected file; you can save multiple registration codes in one file. Or, you can copy the registration code and send it to Delta. Delta will send the UID.imf license file to you once the registration code is received and verified. This license file can apply to one or more hardware. If the verification of the registration code has failed, Delta will notify you by email and request that you provide the correct registration code.

8. After you receive the UID.imf license file, press **Activate License** to activate the software.

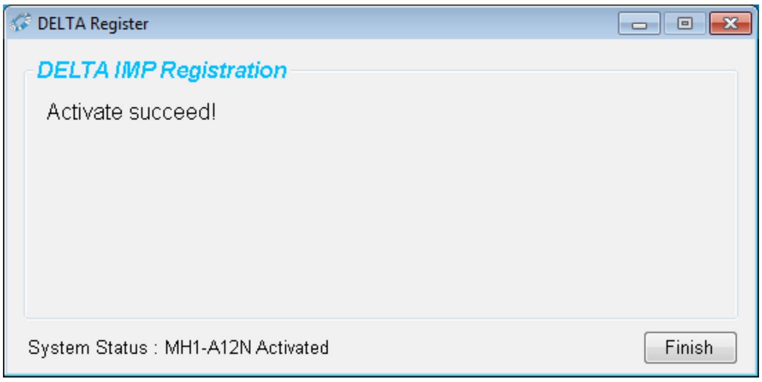
9. Then, select the .imf file when the window for selecting the license file pops up.



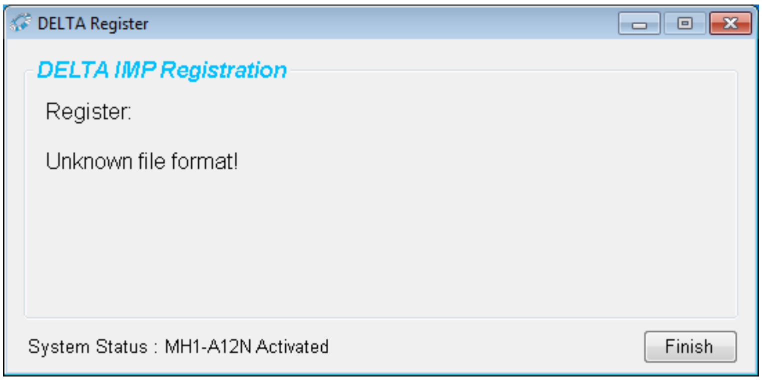
10. Wait for key authentication.



11. If the authentication is successful, the system status will display “Activated”.



If the authentication failed, the screen will display an error code. Please notify Delta of this error code.



(This page is intentionally left blank.)

IMP Quick Start

3

Before you use the IMP system for motion control, you can set the mechanism parameters and conduct motion control test runs for the machine with the Quick Start setting interface.

3

3.1 Servo drive settings

When the IMP software is activated, it automatically detects if the PC is equipped with the bus (DMCNET or EtherCAT) communication card, and check whether the bus is connected to the servo drive and remote module. If it detects the bus is not connected to the module, the HMI and PLC in the IMP system will be disabled. Check the following setup steps before operating:

1. Make sure the servo drive has no current alarms.

If Fault Reset is required, you can use the ALRM command or refer to the troubleshooting section in the servo drive user manual to clear the alarm.

2. Activate the servo drive.

To execute servo motions, the servo axis must be in the Servo On state. If not, you can use the SVON command to activate the servo axis.

3. Clear the emergency stop status of the servo drive.

- Make sure the stop command is not in execution.
- Make sure the emergency stop status of the servo drive is cleared. For more details, refer to the servo drive user manual.
- If you still cannot clear the emergency stop state, check the servo drive DI setting. For more details, refer to the servo drive user manual.

4. Others

Make sure the MPG is disabled.

Set register W19000 to 3.

3.1.1 DMCNET settings

1. Select servo drive.

Available options include Delta's ASD-A2-□□□□-F□, ASD-A2R-□□□□-F, ASD-B2-□□□□-F, and ASD-M-□□□□-F series servo drives.

2. Set the servo drive station number (P3-00).

In IMP, DMCNET fieldbus is used to connect the servo drive and remote module. There must be Station No.1 in this platform and the setting for each station number should be within the range of 1 - C.

If you apply a six-axis communication card, the servo axis station numbers should be set to 1 - 6. When the setting is complete, cycle power on the servo drive.

3. Set the servo drive control mode (P1-01).

Set the servo drive communication mode to DMCNET by setting P1-01 to 0x00b. When the setting is complete, cycle power on the servo drive.

4. Set the servo drive transmission speed (P3-01).

Set the servo drive parameter P3-01 to 3203. When the setting is complete, cycle power on the servo drive.

3.1.2 EtherCAT settings

1. Select servo drive.

Available options include Delta's ASD-A2-□□□□-E□ and ASD-A2-□□□□-E series servo drives.

2. Set the servo drive motion unit (P3-18, P3-19)

You can set the motion unit with Delta's EtherCAT A2-E series servo drive. In the IMP operation, set P3-18 to 1 and P3-19 to 20. When the setting is complete, cycle power on the servo drive.

3. Set the servo drive control mode (P1-01).

Set the servo drive communication mode to EtherCAT by setting P1-01 to 0x00C. When the setting is complete, cycle power on the servo drive.

3.2 IMP Quick Start

With this interface, you can operate the servo motions (including linear, arc, helical interpolation, etc.), DMCNET remote modules (digital, analog, dual, and pulse input / output), and set the parameters for connecting the motor and external module.

After the IMP is started, you can see the IMP icon in the lower right corner of your operating system desktop (as shown in Figure 3.2.1). Right-click the icon and you can see the following options:

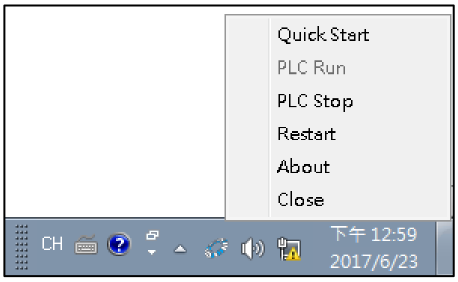


Figure 3.2.1 IMP control panel

Item	Description
Quick Start	Start IMP Quick Start and close the HMI and PLC programs.
Restart	Restart the IMP software.
PLC Run	Start the PLC and HMI programs.
PLC Stop	Close the PLC and HMI programs.
About	About IMP system version information.
Close	Close the IMP system, including the IMP, PLC, and HMI programs.

3

3.2.1 System and communication settings

System Setting

(1) IMP Auto Run

After the IMP is installed and the authentication process is completed, the IMP function will be automatically activated when booting. You can disable this setting with this option.

(2) HMI Border

After the HMI software is installed, the default setting of the HMI Border is enabled so that you can easily adjust the device. After the adjustment is completed, you can disable the HMI Border function for the HMI to appear in full screen.

(3) Language

The system supports Traditional Chinese, Simplified Chinese, and English. After you switch the languages with the interface, you need to restart the Quick Start for the language change to take effect.

(4) Export File

This function allows you to export parameter files to the specified folder or import specified parameter files. The parameter files can include the module parameters of each station, the motion macro program, and configuration of the software numerical controller.

(5) Physical Topology NodeID Check

When you enable this function, the bus connection will align with the slaves during the bus initialization phase of the IMP. If the slave sequence or the configuration of the connection is found different from the stored topology, the IMP will report the error and stop the PLC operation to avoid wrong connection of the bus resulting in machine malfunction. If this function is enabled, the topology of the current bus connection device will be stored and next time when the IMP is activated, the detection function automatically starts.

(6) Use Third Party

If you enable the function for calling third party software, the selected execution file will be called after the bus initialization is completed by the IMP, and the HMI will not start the interface operation function by the third party software.

MPM Editor

This editor allows users to edit the MPM (Motion Program Macro) to make the PLC call. The MPM is written to simplify the PLC motion flow program, and quickly modify the machine motion flow by changing the MPM. For more details on the instructions, please refer to Chapter 7 Motion Program Macro (MPM).

Modbus TCP

This function is used to set the IMP as the Ethernet slave communication, and can enable the master station function to set the slave device information connected through Modbus/TCP. You can use the table to set the Modbus format to read / write the machine communication position and the corresponding D register, then you can directly access the data in the PLC. For more details on the settings, please refer to Section 8.1 Modbus communication setting.

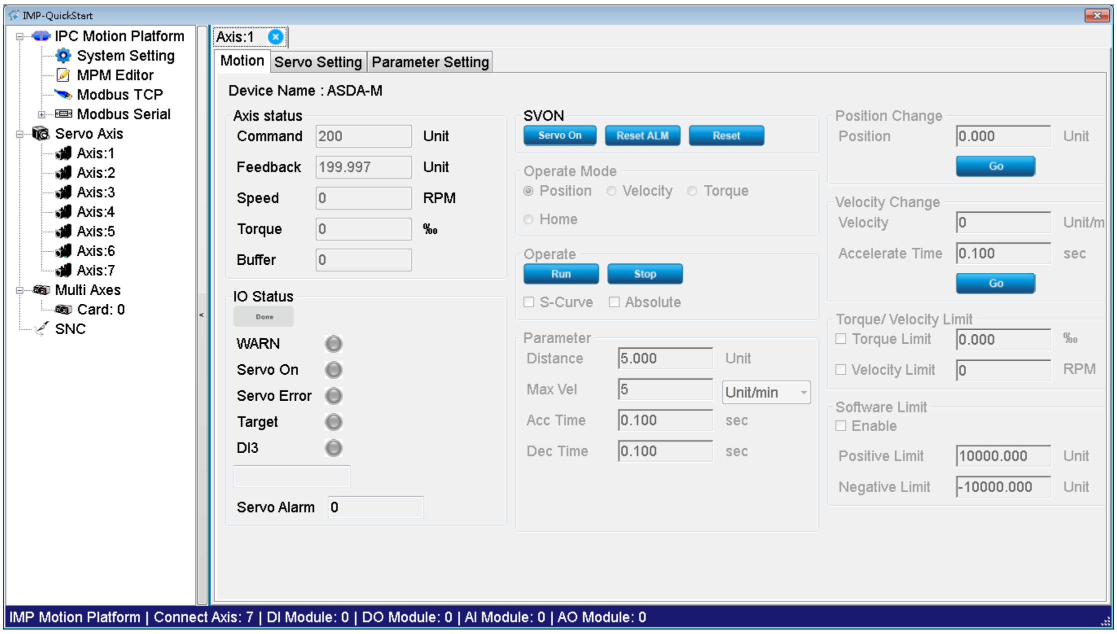
Modbus Serial

This function is used to set the IMP as the communication setting when the serial communication slave is set, and can enable the master station function to set the communication format information of the slave stations connected through RS-232/422/485 communication. You can use the table to set the Modbus format to read / write the machine communication position and the corresponding D register, then you can directly access the data in the PLC. For more details on the settings, please refer to Section 8.1 Modbus communication setting.

3.2.2 Single-axis motion

The single-axis motion control includes motion test and pulse module. You can select the servo axis displaying on the left hand side of the interface to go to each tab:

- Motion
- Servo Setting
- Parameter Setting



Motion

Modules that support single-axis motion control include servo drives and pulse modules.

The connected axis device detected by the software will be displayed on the left hand side of the window. Select a servo axis to open its corresponding motion control interface in the right hand side of the window. As shown in the figure below, you can switch between the tabs, including Motion, Servo Setting, and Parameter Setting.

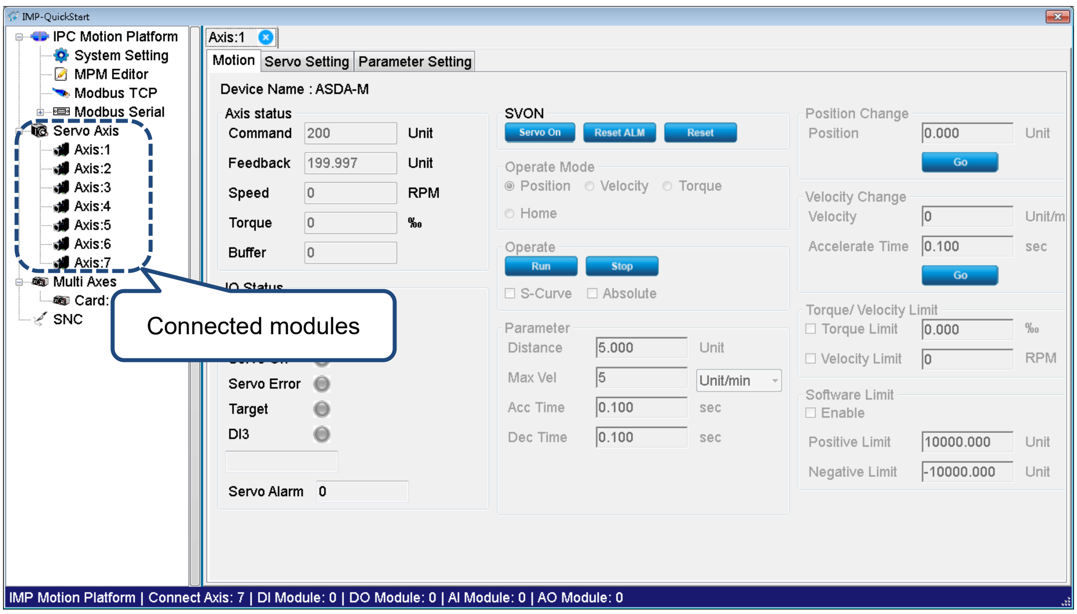


Figure 3.3.1.1 Single-axis control screen

3

The following describes the settings for Motion, Servo Setting, and Parameter Setting:

■ Motion

Item	Subitem	Description
Motion mode		Select motion mode.
	Speed unit	Select the unit for the speed motion, which includes pulse/sec, Feedrate %, and mm/min.
	Point to point	Use the point-to-point position motion mode.
	Home	Use the homing motion mode.
	Velocity	Use the speed motion mode.
	Torque	Use the torque motion mode.
Operate		Execute the motion command.
	Run	Issue the motion command according to the set motion mode and parameter settings.
	Stop	Stop the current motion.
	S-Curve	Select the acceleration / deceleration mode for the motion command. Check [S-Curve] for S-curve acceleration and uncheck [S-Curve] for linear acceleration.
	Absolute	Select absolute coordinate or relative incremental distance for the position parameter of the motion command.
Parameter		Set the related parameters for the motion command.
	Distance	Set the moving distance for the motion command (unit: mm).
	Max Vel	Set the maximum speed (normal speed) when moving.
	Acc Time	Set the acceleration time from zero speed to the set maximum speed (unit: sec).
	Dec Time	Set the deceleration time from the set maximum speed to zero speed (unit: sec).
	Homing mode	Set the homing mode with the value ranging from 0 to 35.
	Homing offset	After homing is completed, the mechanism refers to the sensor's stop position coordinates.
Axis status		The current motion state of the axis.
	Command	Display the current command position.
	Feedback	Display the current feedback position of the servo motor.
	Speed	Display the current moving speed of the servo motor.
	Torque	Display the current torque value of the servo motor.
SVON	Servo On	Enable the servo drives (SVON).
	Reset ALM	Clear the alarms for the servo drives.
	Reset	Set the current coordinate position to 0.
IO Status		Display the current status of the servo drive.
	DI3	Display the DI3 status of Delta's servo drive ASDA-A2-F.
	WARN	Display the servo drive alarm status.
	Servo On	Display the servo on status of the servo drive.
	Servo Error	Display the servo drive error status.
	Target	Display the target position arrival status of the servo drive.

Item	Description
Position Change	Change the current motion position.
Velocity Change	Change the current motion speed.
Torque / Velocity Limit	Set the maximum torque limit in speed mode; set the maximum speed limit in torque mode.
Software Limit	Set the software limit.

■ Servo Setting

When the connecting devices are Delta's ASD-A2-□□□□-E□, ASD-A2-□□□□-F□, ASD-B2-□□□□-F, and ASD-M-□□□□-F series servo drives, you can read / write the servo drive parameters through the communication bus with the Servo Setting in the single-axis control interface, as shown in the figure below. For the details of each parameter, please refer to the servo drive user manual.

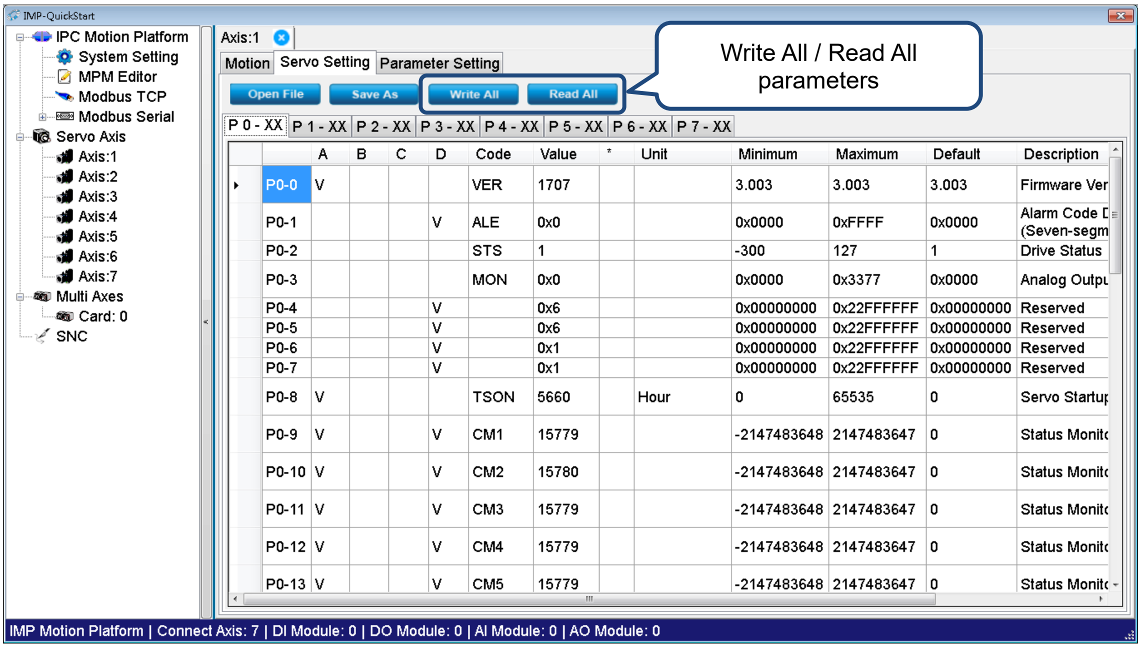


Figure 3.3.2.2 Read the servo drive parameters

■ Parameter Setting

Before using the motion axis, you must use the Parameter Setting function to input the motor resolution and the moving distance of the driving mechanism. If the pulse module is used to drive the motion axis, you also need to set the input / output pulse type and limit. After completing the setup and restarting the IMP, the PLC and SNC system can operate the servo motion with the mechanical unit. Please refer to the figure below.

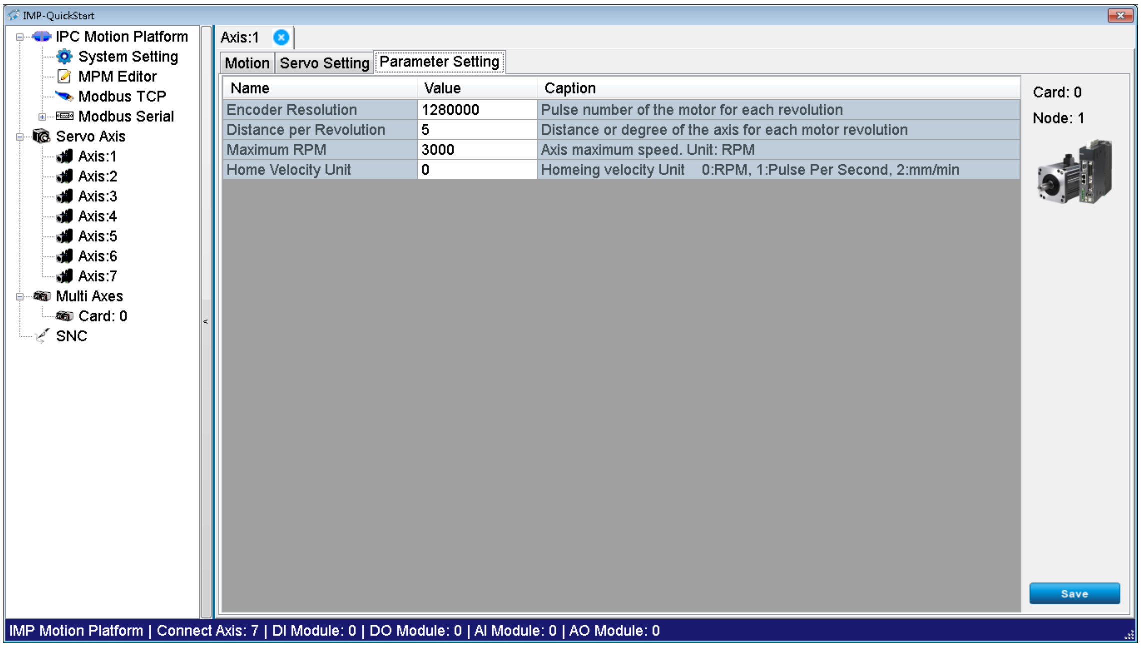


Figure 3.2.2.3 Pulse module parameter setting screen

The description of each parameter is as follows:

Name	Description	Default	Applicable device
Pulse_Per_Rev	The required pulse number per motor revolution: $f1 \times \frac{Puls\_Per\_Rev}{Dist\_per\_Rev} = f2$ $f1$: command source (mechanical unit: mm, degree) $f2$: actual output PUU	1280000 pulse/rev	Servo module: ASD-A2-□□□□-E□ ASD-A2-□□□□-F□ ASD-B2-□□□□-F ASD-M-□□□□-F Pulse module: R1-EC5621D0 ASD-DMC-RM04PI ASD-DMC-GE01PH
Dist_Per_Rev	The moving distance of the machine per motor revolution. The moving angle of the machine per motor revolution. Suggested value: 360 degrees.	10 mm/rev	
Max_RPM	Maximum speed of the motor (unit: rpm).	1000 rpm	
Home_Vel_Unit	Homing speed unit of the motor: 0: rpm/min 1: PPS 2: mm/min	0	

Name	Description	Default	Applicable device
ipulser_mode	Mode of input phase: 0: AB phase pulse 1: clockwise and counterclockwise pulse	0	Pulse module: R1-EC5621D0 ASD-DMC-RM04PI ASD-DMC-GE01PH
opulser_mode	Mode of output phase: 0: AB phase pulse 1: clockwise and counterclockwise pulse 2: pulse + symbol (b: low level) 3: pulse + symbol (a: high level)	0	
Svon_polarity	Contact type of SVON signal output: 0: normally open contact (a: high level trigger) 1: normally closed contact (b: low level trigger)	0	
PEL_ polarity	Contact type of positive limit signal input: 0: normally closed contact (b: low level trigger) 1: normally open contact (a: high level trigger)	0	
MEL_polarity	Contact type of negative limit signal input: 0: normally closed contact (b: low level trigger) 1: normally open contact (a: high level trigger)	0	
ALM_ polarity	Contact type of servo alarm signal input: 0: normally open contact (a: high level trigger) 1: normally closed contact (b: low level trigger)	0	
ORG_ polarity	Contact type of origin signal input: 0: normally open contact (a: high level trigger) 1: normally closed contact (b: low level trigger)	0	

3.2.3 Multi-axis synchronous motion

After selecting the multi-axis interpolation card number in the left window of the main screen, the multi-axis interpolation window on the right will appear (as shown in the figure below).

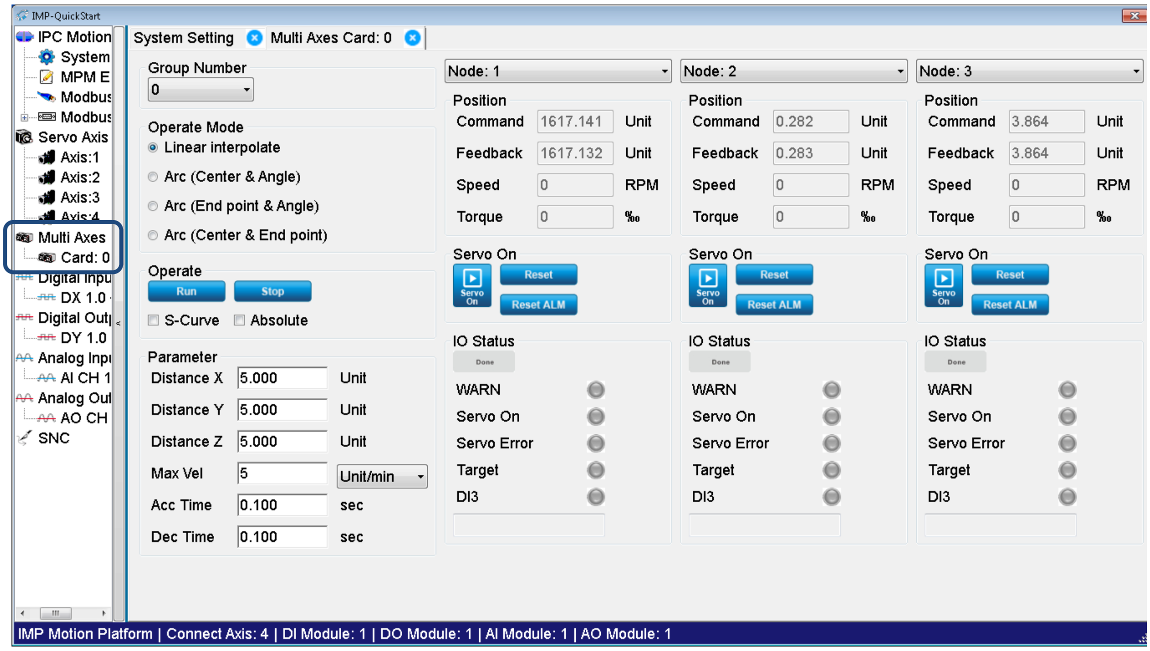


Figure 3.2.3.1 Multi-axis interpolation interface

The description of each function is as follows:

<table>
<tr><th>Item</th><th colspan="2">Description</th></tr>
<tr><td>Speed unit</td><td colspan="2">There are three types of speed units: pulse/sec, Feedrate%, and mm/min.</td></tr>
<tr><td>Group Number</td><td colspan="2">This is the interpolation group with a total of 40 groups; this group is the same as the PLC motion group.</td></tr>
<tr><td>Operate Mode</td><td colspan="2">Multi-axis synchronization:
Linear interpolation
Arc 1 (known circle center coordinates and arc angle)
Arc 2 (known endpoint coordinates and arc angle)
Arc 3 (known circle center coordinates and endpoint coordinates)</td></tr>
<tr><td rowspan="5">Operate</td><td>Subitem</td><td>Description</td></tr>
<tr><td>Run</td><td>Start running the set interpolation motion.</td></tr>
<tr><td>Stop</td><td>Stop the set interpolation motion.</td></tr>
<tr><td>S-Curve</td><td>Select the acceleration / deceleration mode for the motion command. Check [S-Curve] for S-curve acceleration and uncheck [S-Curve] for linear acceleration.</td></tr>
<tr><td>Absolute</td><td>Set the input target as relative to the current coordinates or as the system absolute coordinates.</td></tr>
</table>

Item	Subitem	Description
Parameter		Set the related parameters for the motion command.
	Distance X	Set the moving distance for the X axis motion command (unit: mm).
	Distance Y	Set the moving distance for the Y axis motion command (unit: mm).
	Distance Z	Set the moving distance for the Z axis motion command (unit: mm).
	Max Vel	Set the maximum speed (normal speed) when moving.
	Acc Time	Set the acceleration time from zero speed to the set maximum speed (unit: sec).
	Dec Time	Set the deceleration time from the set maximum speed to zero speed (unit: sec).
	Circle center X	The X axis circle center position when using Arc 1 and Arc 3 for the Operate Mode.
	Circle center Y	The Y axis circle center position when using Arc 1 and Arc 3 for the Operate Mode.
	Angle	The arc angle when using Arc 1 and Arc 2 for the Operate Mode.
	Endpoint X	The X axis target position when using Arc 2 and Arc 3 for the Operate Mode.
	Endpoint Y	The Y axis target position when using Arc 2 and Arc 3 for the Operate Mode.
	Direction	The arc moving direction when using Arc 3 for the Operate Mode.
Axis status		The current motion state of the axis.
	Position command	Display the current command position.
	Feedback	Display the current feedback position of the servo motor.
	Speed	Display the current moving speed of the servo motor.
	Torque	Display the current torque value of the servo motor.
Servo On	Servo On	Enable the servo drives (SVON).
	Reset ALM	Clear the alarms for the servo drives.
	Reset	Set the current coordinate position to 0.
IO Status		Use with the "Monitor" function of the multi-axis interpolation to display the current status of the servo drive.
	DI3	Display the DI3 status of Delta's servo drive ASDA-A2-F (only applicable to DMCNET).
	WARN	Display the servo drive alarm status.
	Servo On	Display the servo on status of the servo drive.
	Servo Error	Display the servo drive error status.
	Target	Display the target position arrival status of the servo drive.

3

3.2.4 Digital input / output

The left side of the main screen displays the DI and DO contacts that are currently operable by the system, which sequence is based on the module connection. Click on the digital input device to open the monitoring window on the right; click on the digital output device to open the manual output control window.

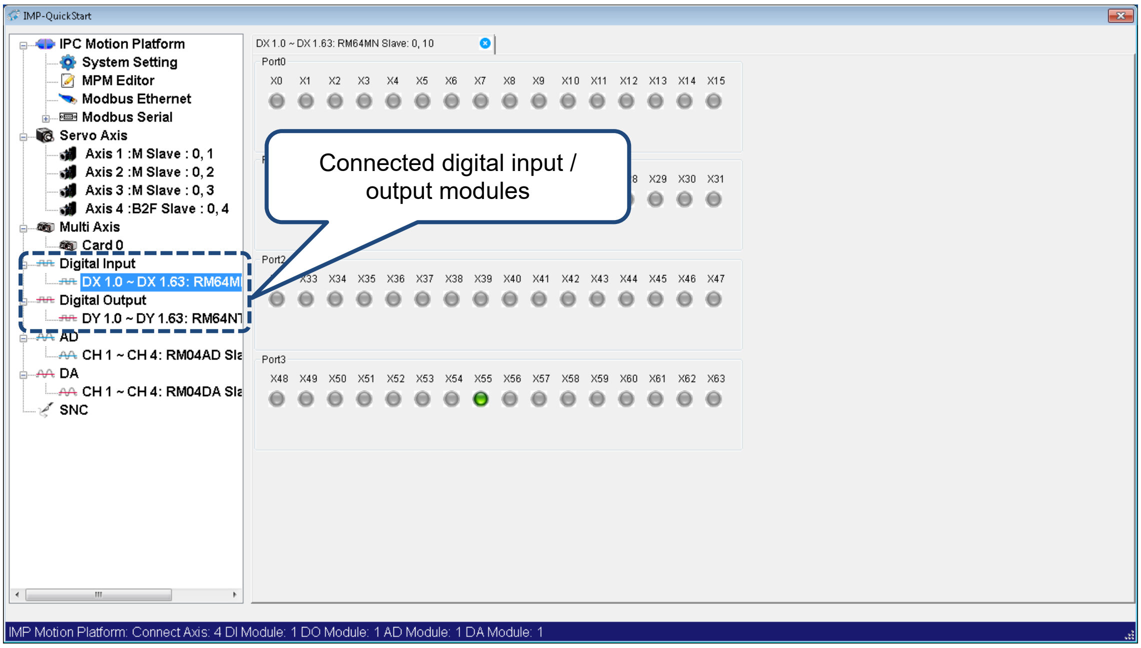


Figure 3.2.4.1 Input monitoring

3.2.5 Analog input / output

The left side of the main screen displays the analog input / output module channels that are currently operable by the system, which sequence is based on the module connection. Click on the analog input channel to open the scope window on the right; click on the analog output channel to open the manual output control window.

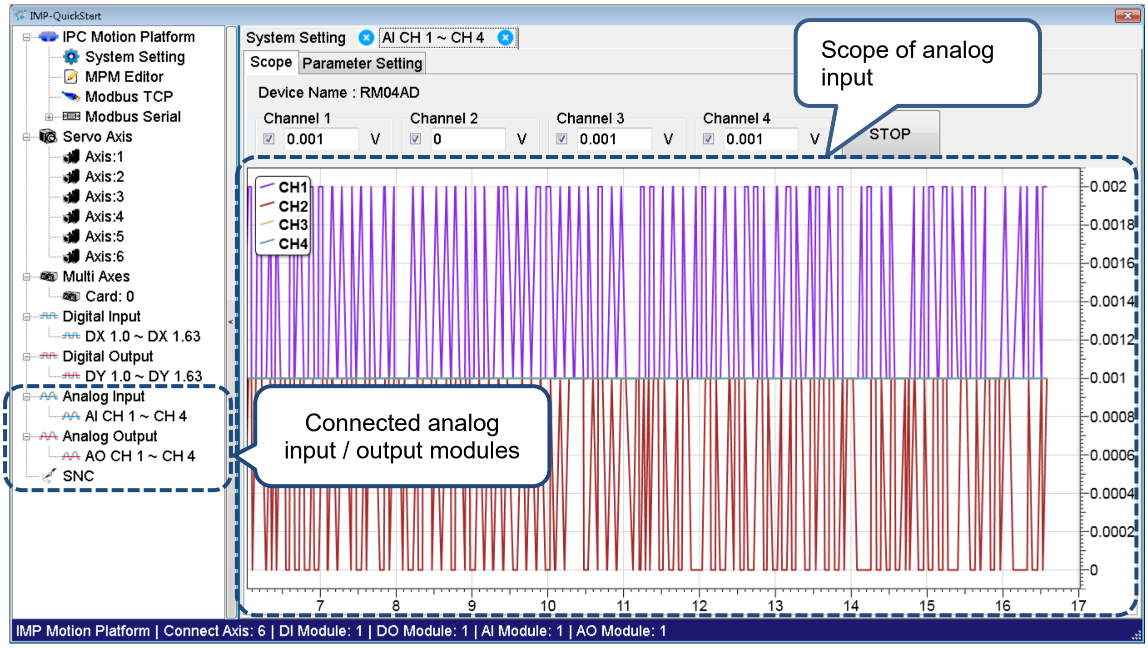


Figure 3.2.5.1 Analog input CH 1 - CH 4 view screen

3

3.2.6 Analog input

■ Parameter setting for the analog input module

When using the remote analog module ASD-DMCRM04AD / R1-EC8124D0, you can set the conversion speed, input analog signal range, and use the input signal averaging function with this interface, as shown in the figure below.

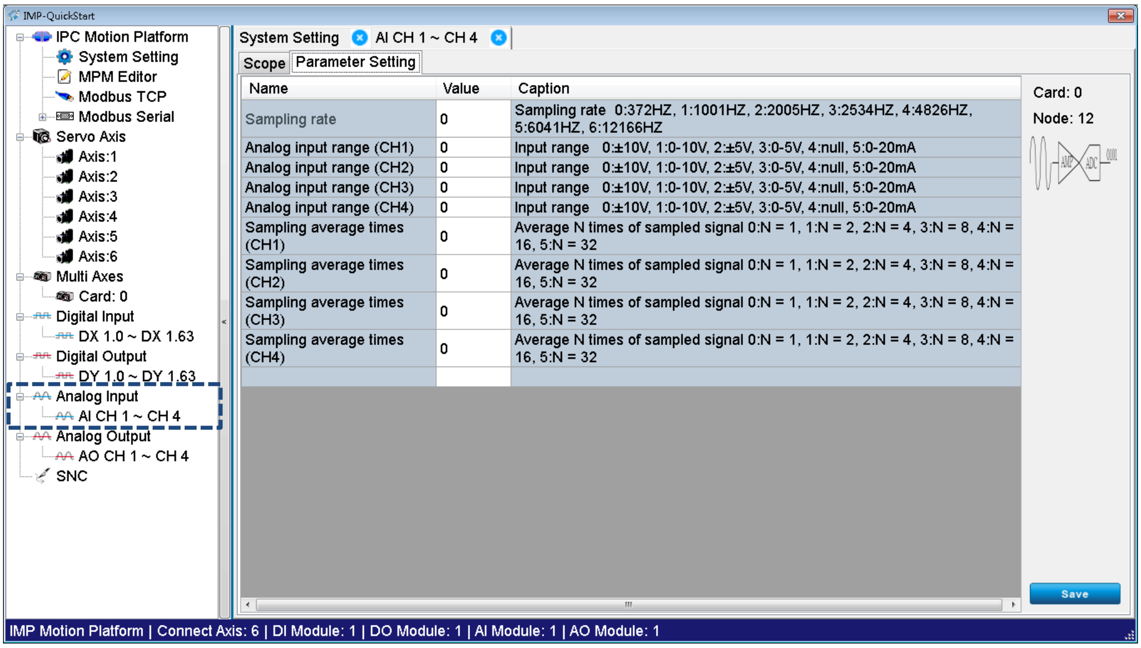


Figure 3.2.6.1 Parameter setting screen for the analog input module

The description of each parameter is as follows:

(1) Conversion_time: AD conversion speed. The parameter value is 0 - 6, and the default is 0. Refer to the table below:

Value	Conversion frequency (Hz)	-3dB gain bandwidth (Hz)	RMS noise (μv)
0	372	200	9.6
1	1001	520	15.5
2	2005	1040	22.7
3	2534	1300	26.1
4	4826	2500	39.2
5	6041	3100	46.0
6	12166	6300	120.0

(2) InRange_1,2,3,4: AD input range. The parameter value is 0 - 5, and the default is 0. Refer to the table below:

Value	Definition of input range
0	±10V
1	0V - 10V
2	±5V
3	0V - 5V
4	Reserved
5	0 - 20 mA

(3) Average_mode_1,2,3,4: AD averaging function mode. The parameter value is 0 - 5, and the default is 0. Refer to the table below:

Value	Set the number of samples for the average value
0	0
1	2
2	4
3	8
4	16
5	32

■ Parameter setting for the analog output module

When using the remote analog output module ASD-DMC-RM04DA / R1-EC9144D0, you can set four sets of analog output offset (DA offset) and the output range, as shown in the figure below.

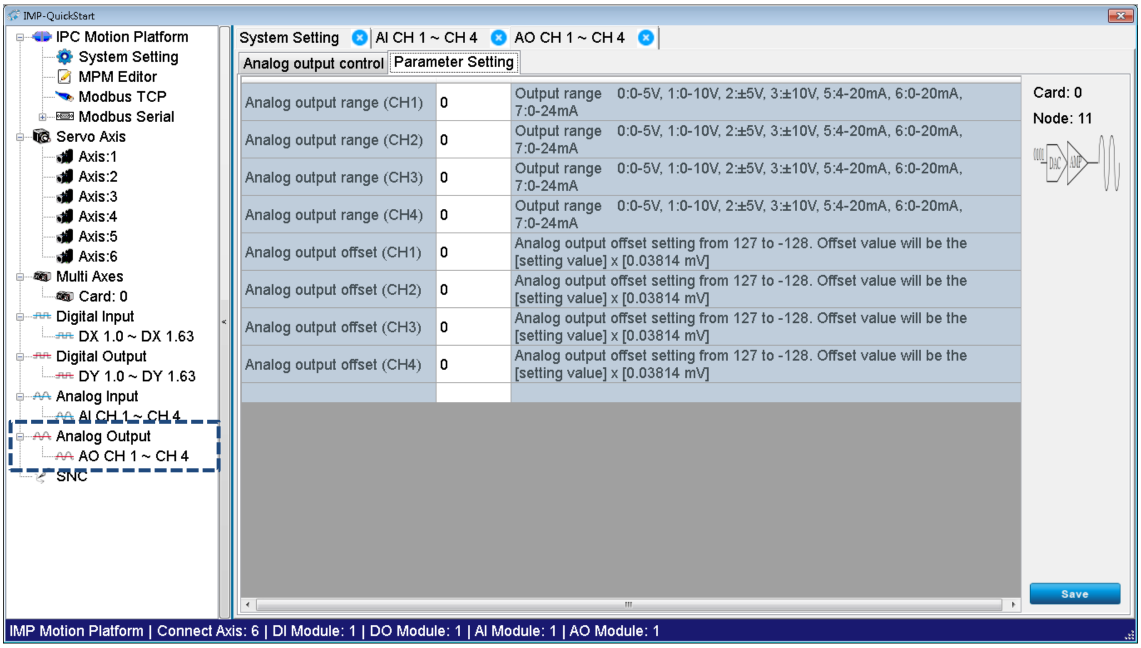


Figure 3.2.6.2 Parameter setting screen for the analog output module

3

The description of each parameter is as follows:

(1) OutRange_1,2,3,4: DA output range. The parameter value is 0 - 7, and the default is 1. Refer to the table below:

Value	Definition of output range
0	0V - 5V (default)
1	0V - 10V
2	±5V
3	±10V
4	Reserved
5	4 - 20 mA
6	0 - 20 mA
7	0 - 24 mA

(2) OutOffset_1,2,3,4: output adjustment offset (DA offset). The parameter value is -128 to +127, and the default is 0. Each adjustment offset moves approximately 0.03814 mV, as shown in the following table:

Value	Definition of adjustment offset
127	4.844 mv ≅ 127 × 0.03814
126	4.806 mv ≅ 126 × 0.03814
…	…..
1	0.038 mv ≅ 1 × 0.03814
0	No adjustment (default)
-1	-0.038 mv ≅ -1 × 0.03814
…	…
-127	-4.844 mv ≅ -127 × 0.03814
-128	-4.882 mv ≅ -128 × 0.03814

3.2.7 Software numerical control (SNC)

The SNC interface is shown as follows which is for performing software numerical control function test, parameter setting, backlash compensation, coordinate setting for G52 - G59, tool length and diameter setting, tool offset setting, and tool setter. All the settings mentioned above will be saved by the system. For the parameter function description, please refer to Chapter 9 Soft Numerical Control. (This setting is similar to HMI.)

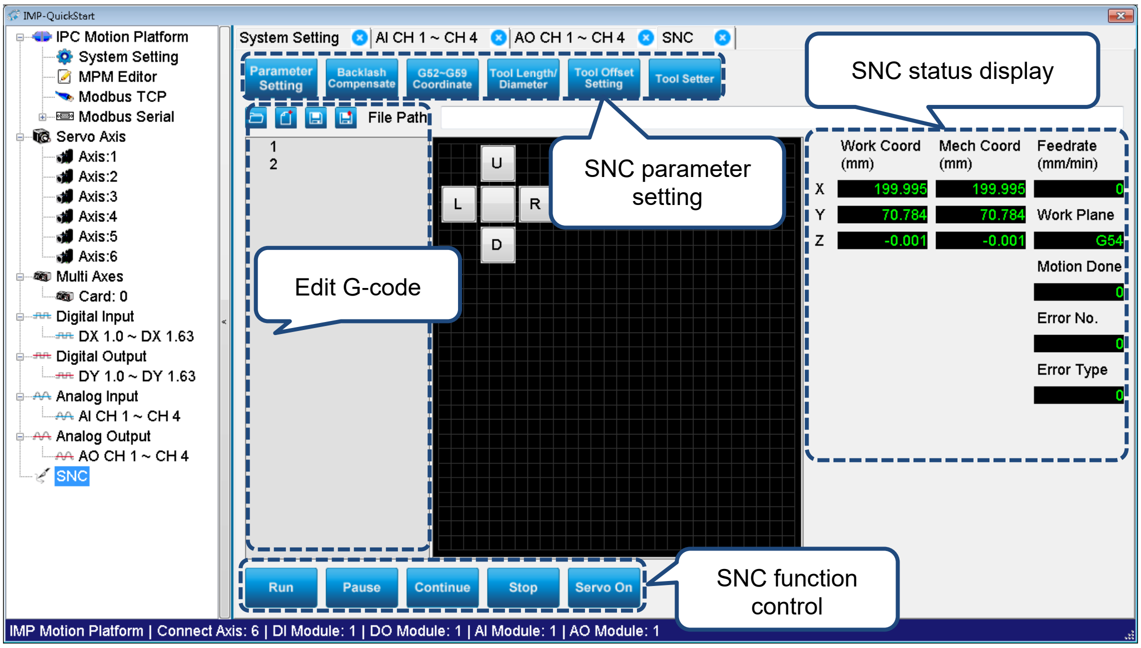


Figure 3.2.7.1 Interface of software numerical control

(This page is intentionally left blank.)

Ladder Editor 4

This chapter introduces the IMP Ladder Editor which is integrated into the DOPSoft software. For the installation of the DOPSoft software and the HMI screen editing functions, please refer to the DOPSoft User Manual.

4

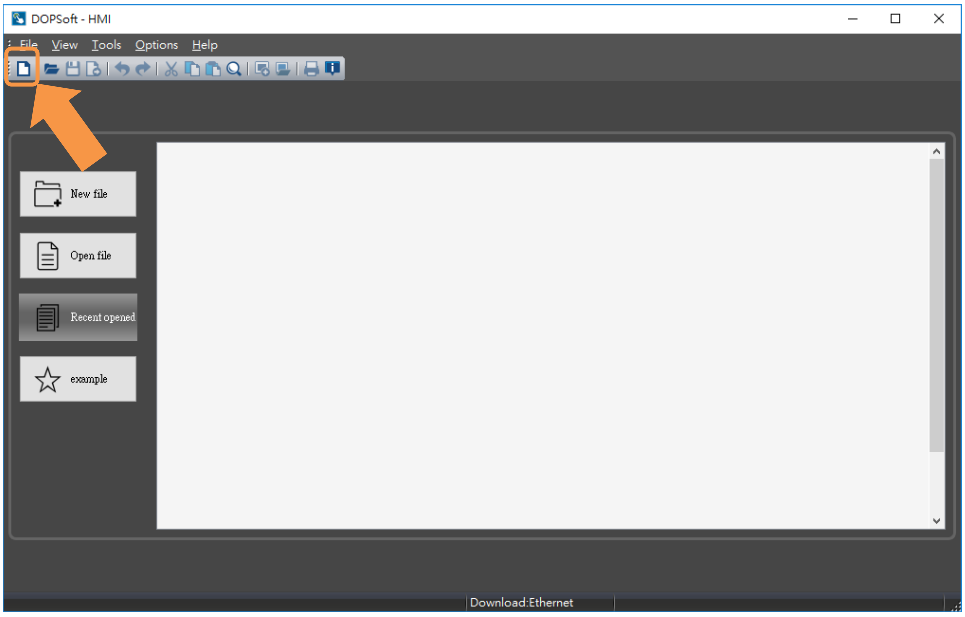
4.1 Introduction to Ladder Editor

This section explains how to start the Ladder Editor and each function on the interface.

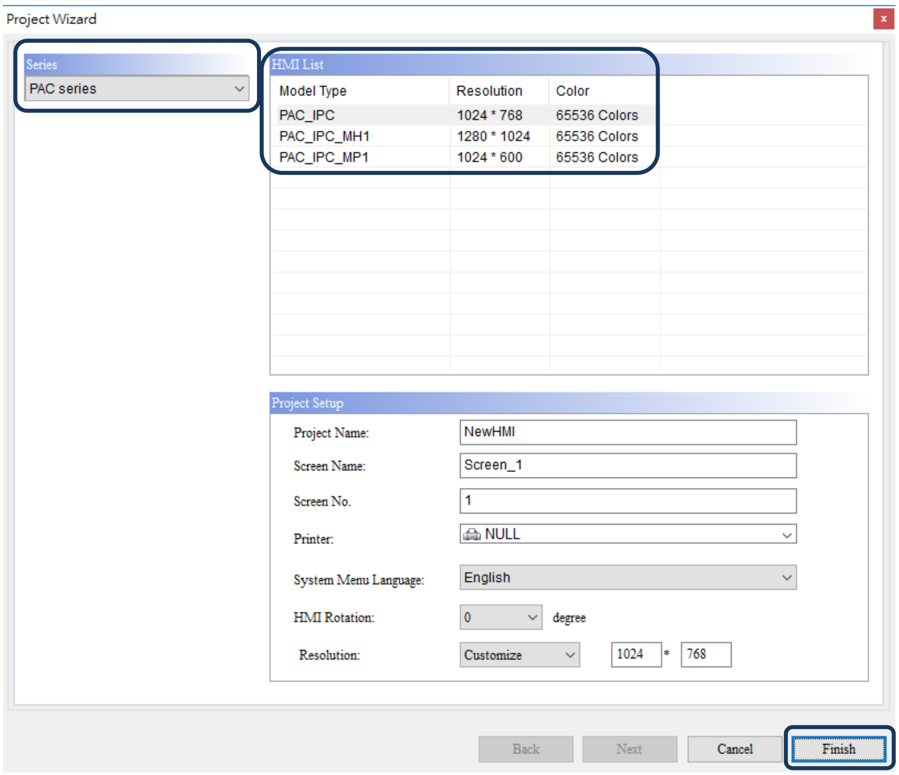
4.1.1 How to start Ladder Editor

Ladder Editor is integrated into the DOPSoft software and you can start the program by following the steps below.

1. Open DOPSoft and click .

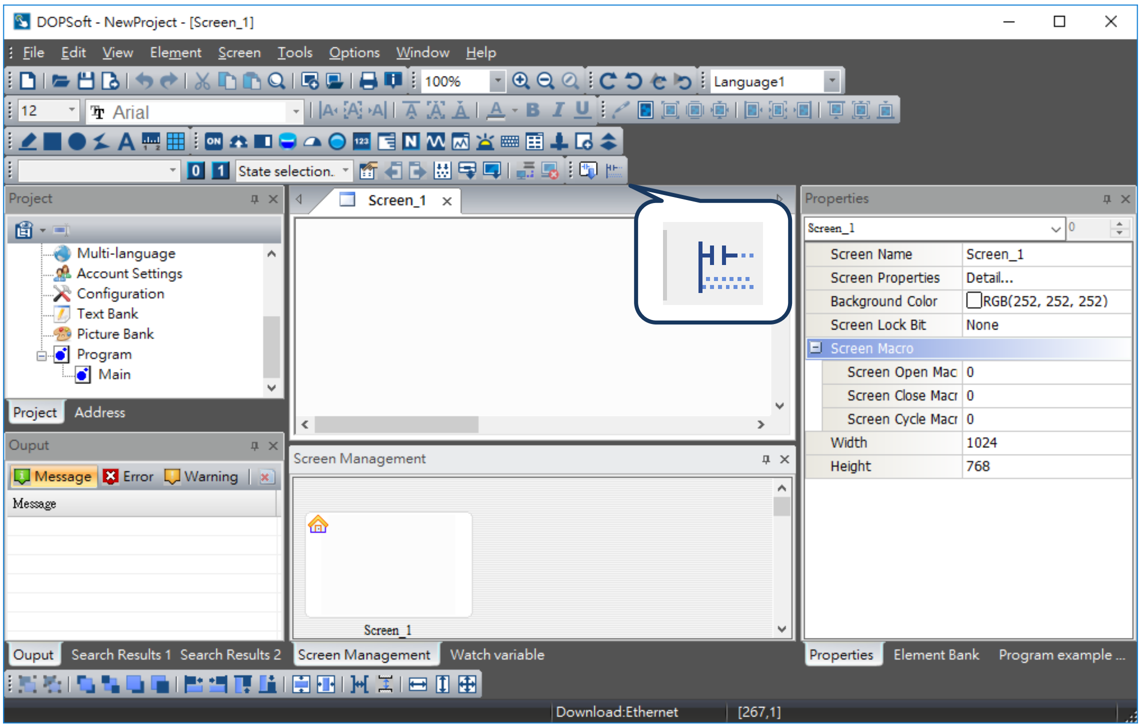


2. Select the HMI model type and click **Finish**.

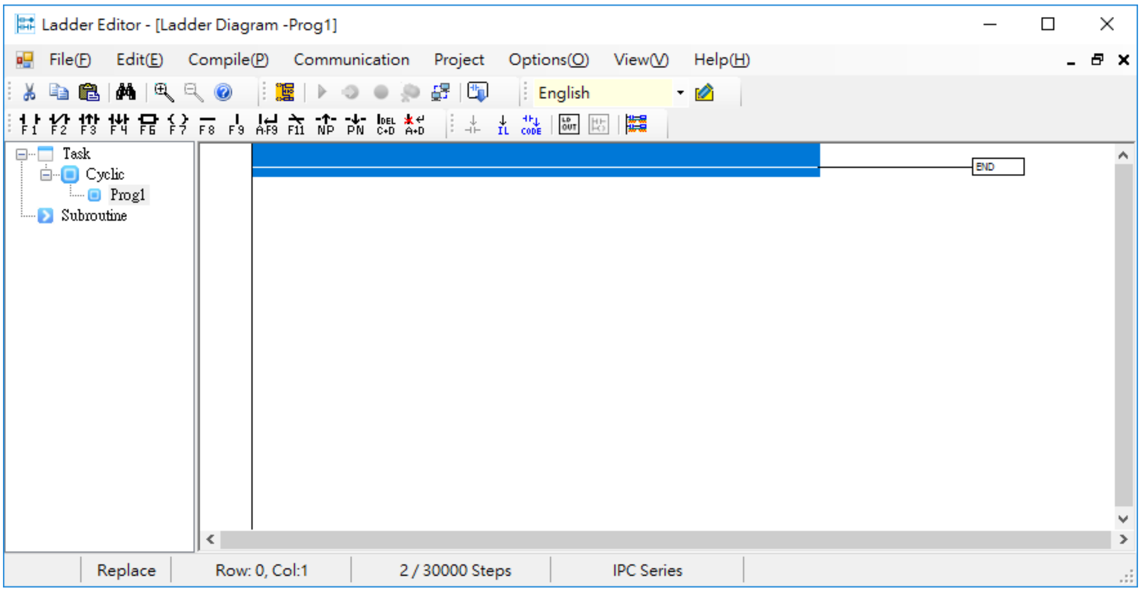


3. Open Ladder Editor.

Click  in the toolbar to start the Ladder Editor.



4. When Ladder Editor is started, the screen is shown as follows:



5. Upload / download data (connection settings)

After editing the screens and programs with DOPSoft, upload / download them to the IMP via Ethernet so they can be executed on your computer.

4

4.1.2 Program upload and download

There are two modes available based on whether IMP and DOPSoft are installed on one IPC or separately (IMP installed on IPC and DOPSoft on PC).

One PC mode: install both IPC Motion Platform and DOPSoft in an IPC. After editing the software screens and programs in DOPsoft, transfer the required files for the IPC Motion Platform to execute by the internal memory.

Multiple PC mode: install the IPC Motion Platform on an IPC and DOPSoft on a PC. After editing the software screens and programs in DOPsoft, transfer the required files for the IPC Motion Platform to execute by Ethernet.

Download all data

When downloading all data, both the screen data and recipe are downloaded to the IMP PAC. You can go to [Tools] > [Download All Data] or you can click  in the toolbar or press the keyboard shortcut **Ctrl+F8** to download all data. When downloading, make sure the PC and IMP PAC are connected and the IMP software is started. Otherwise, an error message will pop up.

Input IP address:

If you are using One PC mode, the IP address is 127.0.0.1; if using Multiple PC mode, input the IP address of the PAC.

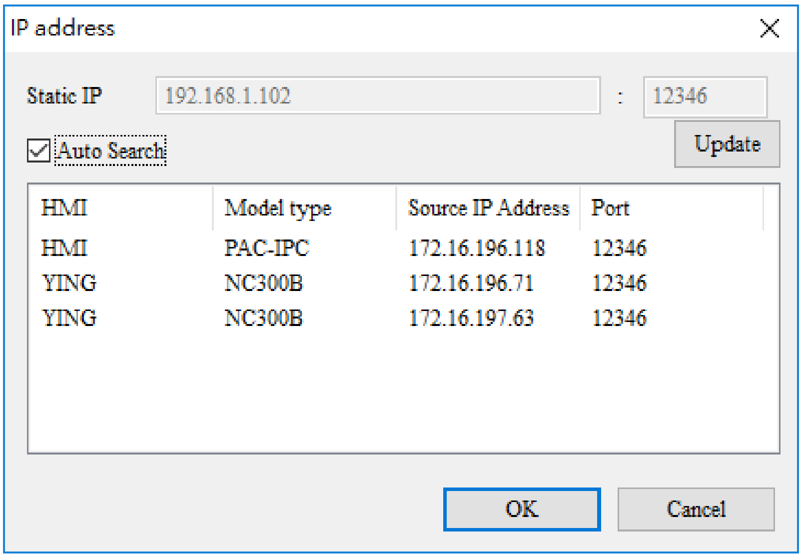


Figure 4.2.1.1 IP search / setting for downloading

Normal transmission:

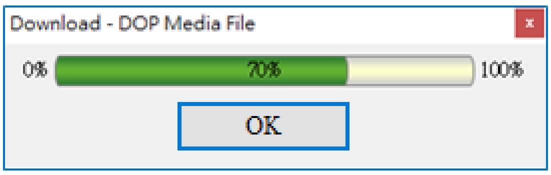
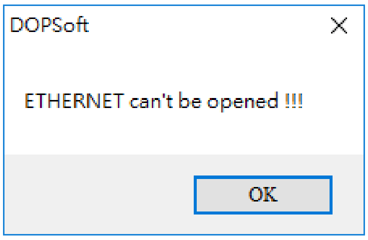


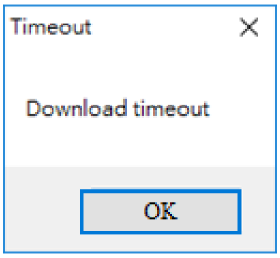
Figure 4.2.1.2 Download all data

When the transmission failed, the system will prompt the following messages based on the situation:

The following error message pops up if an error occurs when enabling the Ethernet or the connected PAC did not correctly start the IMP software.



The following error message pops up if the connection cable is removed or the communication is interrupted during the download process.



Upload all data

Before uploading all data, you will be asked to enter the password. In Figure 4.2.1.3, 12345678 is the system default password, but you can change the password by going to [Options] > [Configuration].

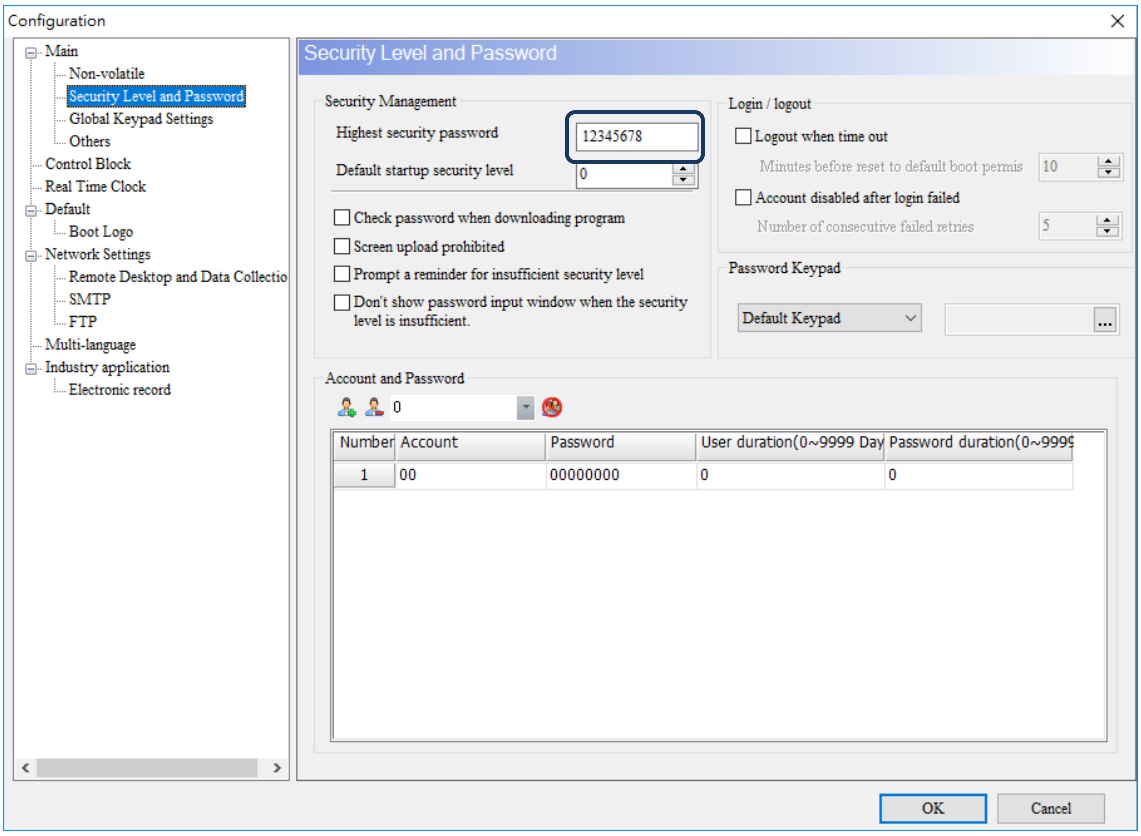


Figure 4.2.1.3 Set the security password

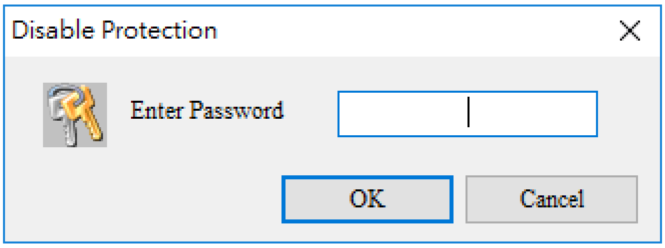


Figure 4.1.2.4 Upload all data (default: 12345678)

After entering the password, the system will upload the screen data until the progress reaches 100%. You can click **Stop** to stop the upload.

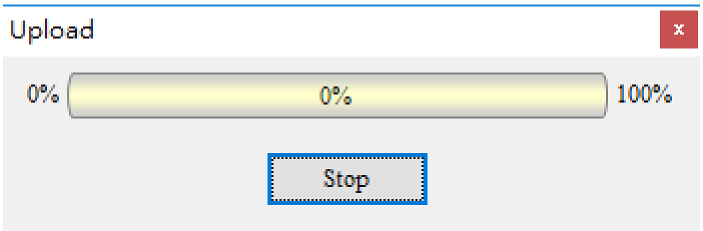


Figure 4.1.2.5 Data uploading

The software will then ask you to save the screen file for uploading, as shown in the figure below.

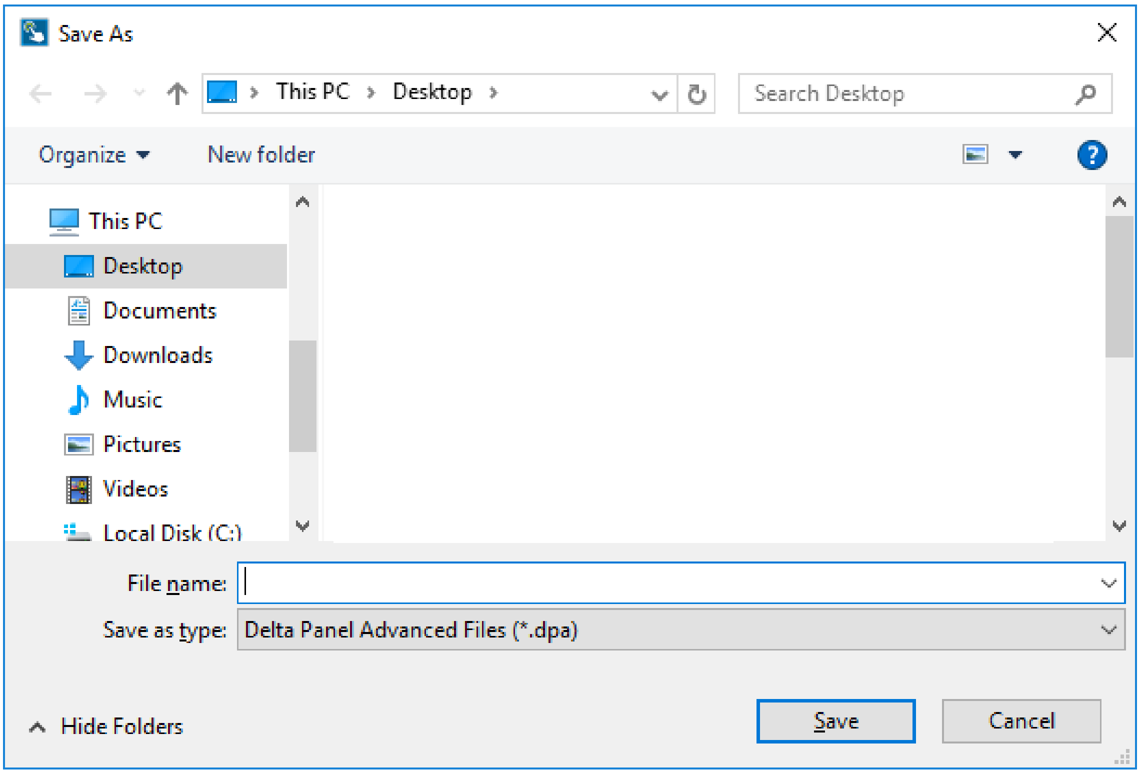


Figure 4.1.2.6 Save upload data

Other than uploading the screen data to the editing PC, you can go to [Options] > [Environment] to set whether to include picture data when uploading.

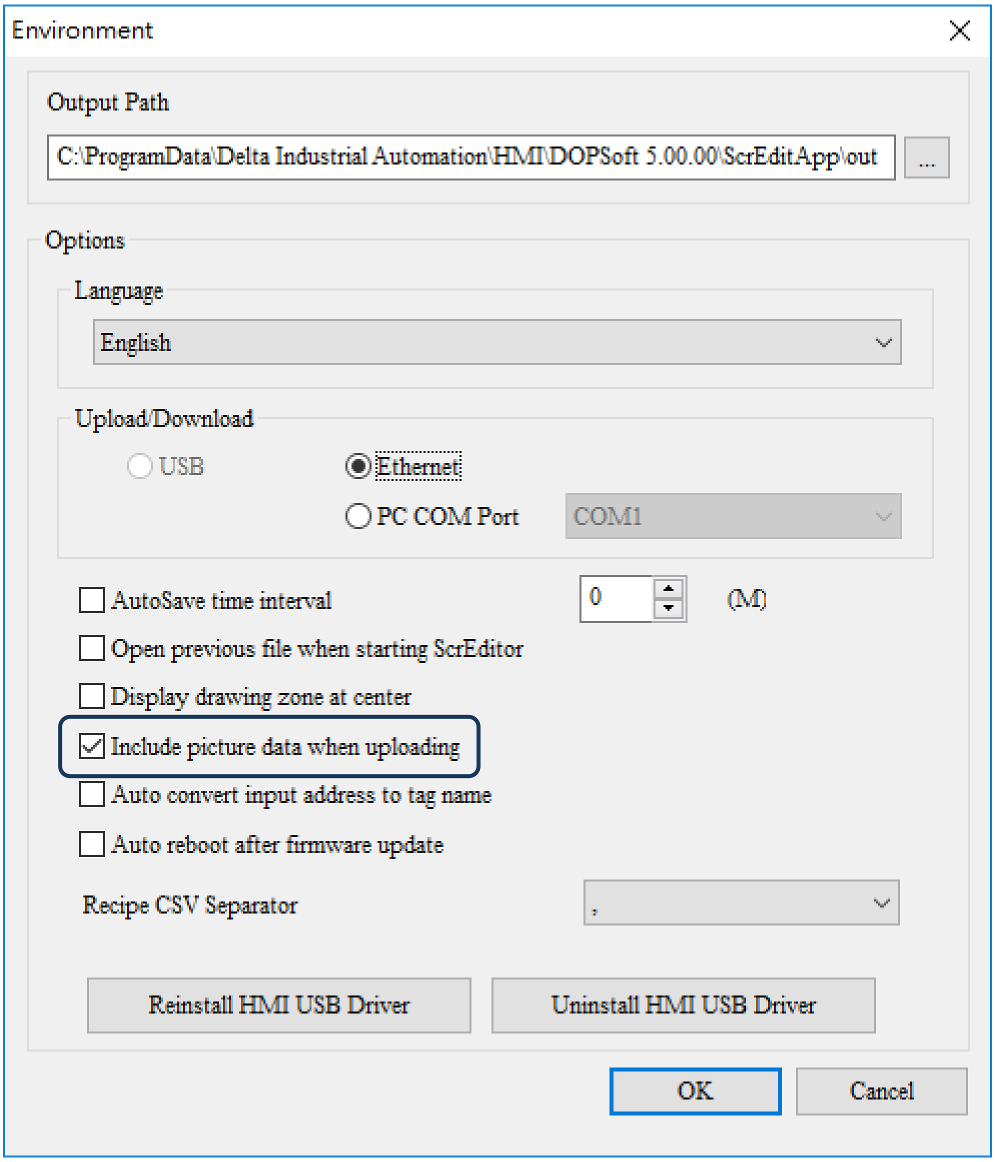


Figure 4.1.2.7 Include picture data when uploading

Download screen data

This function allows you to download only the screen data without the recipe. The steps to download screen data is the same as that of [Download All Data]. You can go to [Tools] > [Download Screen] or you can click  in the toolbar or press the keyboard shortcut **Ctrl+F9** to download the screen data.

Upload recipe

The steps to upload recipe is the same as that of [Upload All Data] and you will also need to enter the password in order to upload the recipe to the PC. The password settings are the same as those described in [Upload All Data].

4

Download recipe

To download only the recipe data, you can do it by executing [Download Recipe]. If you need to modify the recipe without editing other screen data, this function can help you save download time. When using this function, the software will prompt you to select the recipe file (.rcp) for download. Then, you can download the selected recipe file to the HMI.

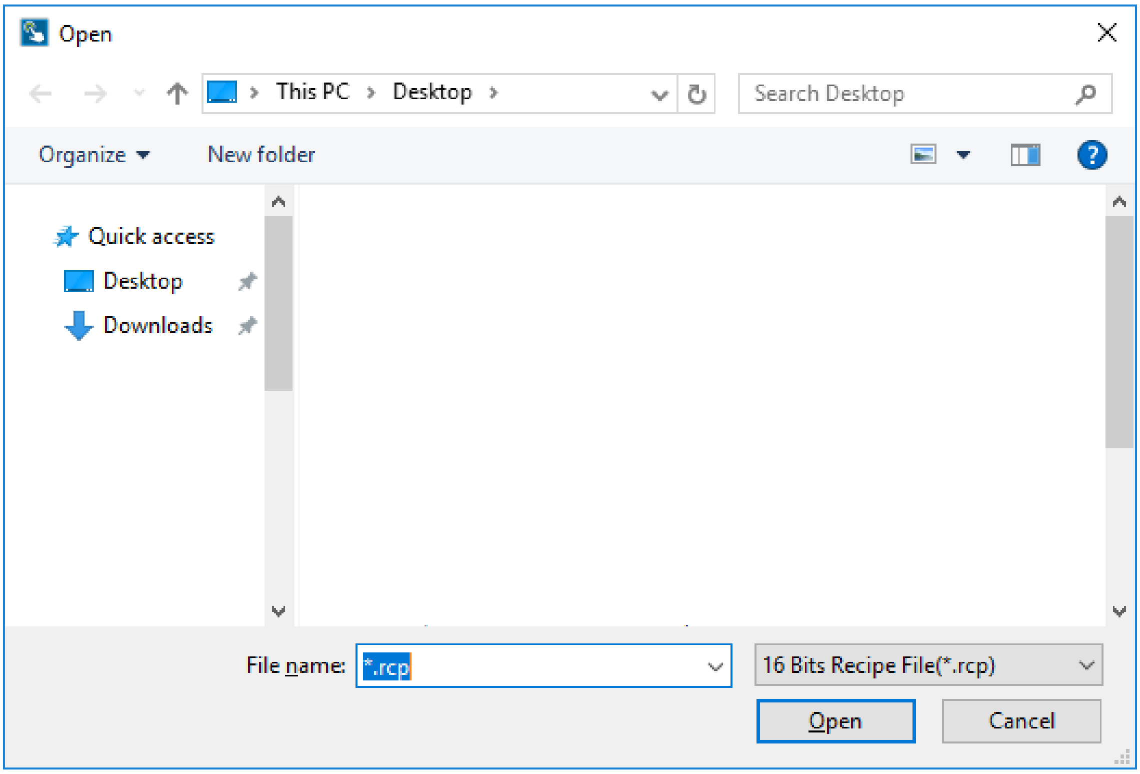


Figure 4.1.2.8 Select the recipe file for download

Download logic data

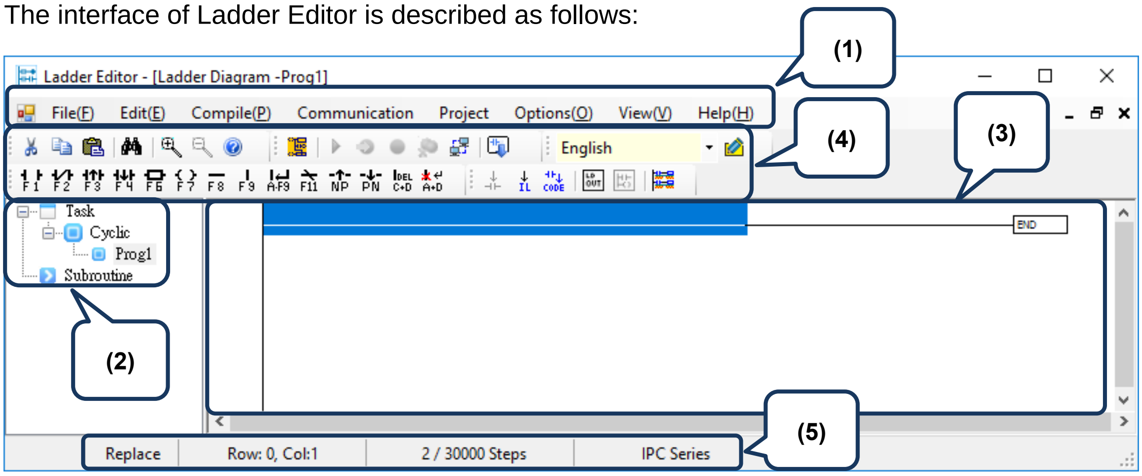
You can download the edited logic data to the IMP PAC with this function.

Upload logic data

You can upload the logic data in the IMP PAC to the PC with this function.

4.2 Create new ladder program and settings

The interface of Ladder Editor is described as follows:



(1) Toolbar: includes functions such as File, Edit, Compile, and Communication.

(2) Program tree view: displays the ladder program structure in the current project.

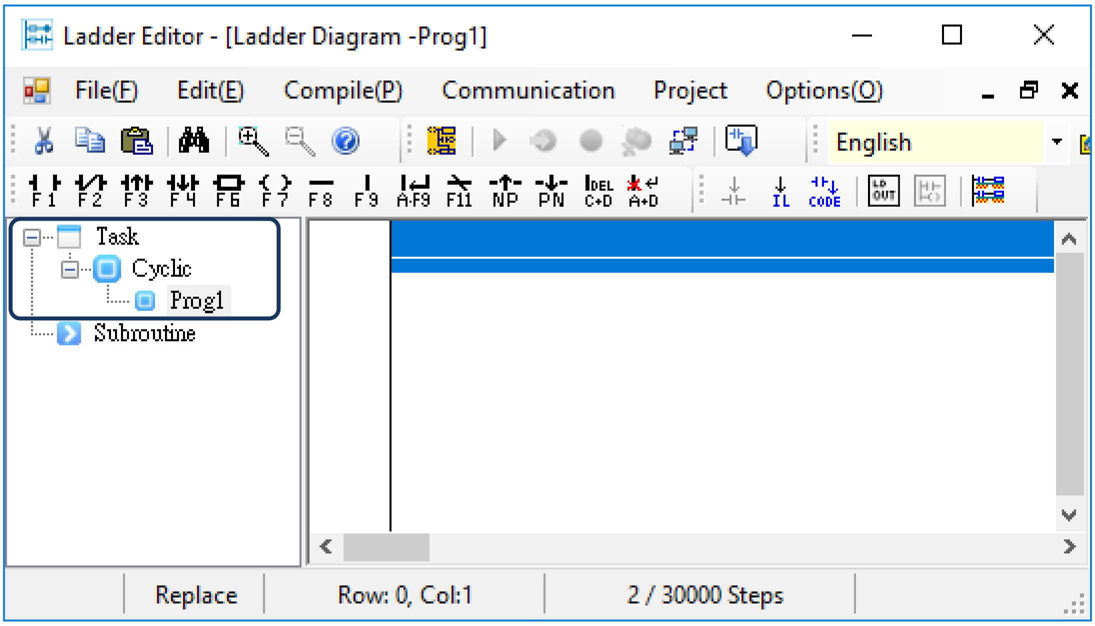
(3) Program editing section: you can edit the selected program here.

(4) Application options: includes output window, search result, and monitoring device window.

(5) Editing status: displays the current editing status which you can switch between Replace or Insert mode.

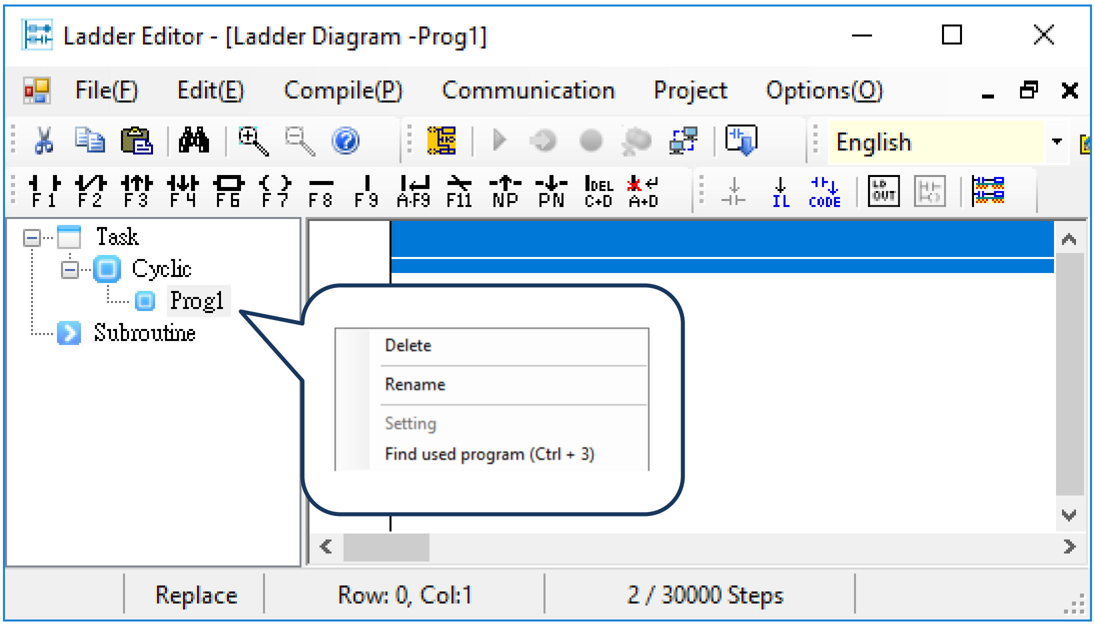
■ **Cyclic Task**

Cyclic Task is located in the program tree on the left side of the screen, as shown below:

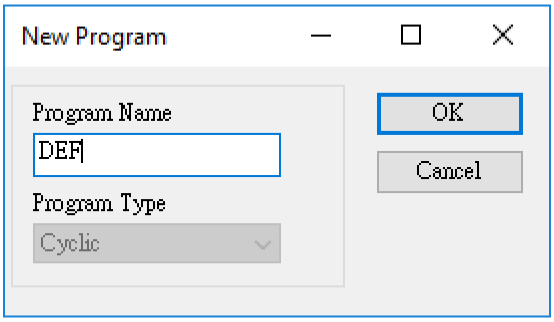


Change the program name

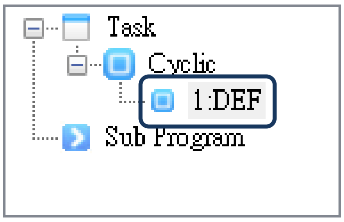
1. To change the program name, right-click the program name and click [Rename].



2. Then, the New Program window pops up for you to input the new program name.



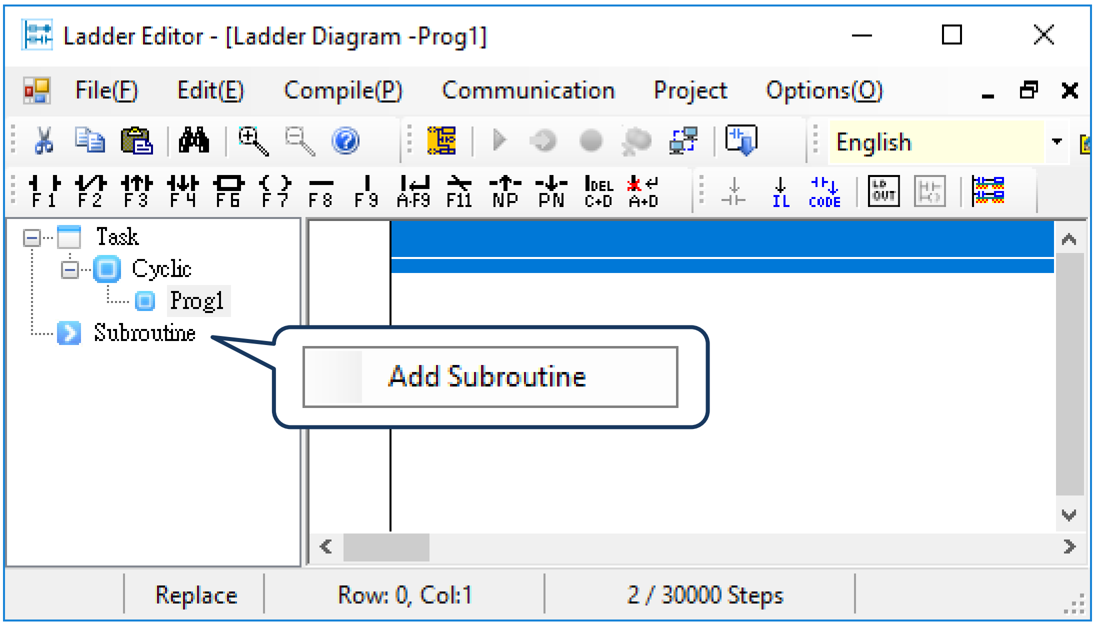
3. When done inputting, click **OK** to complete the program name change.



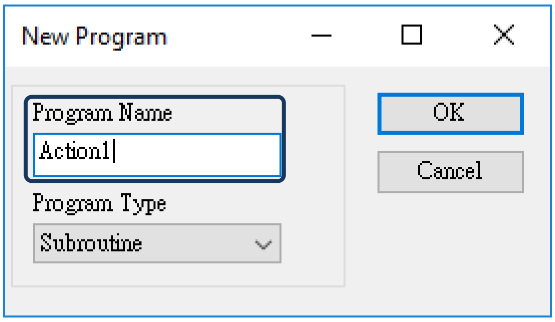
■ Subroutine

Add Subroutine

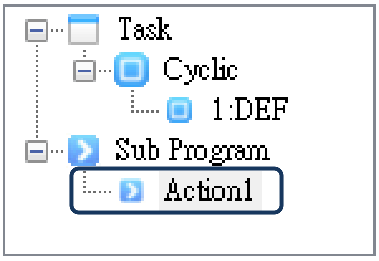
1. Right-click Subroutine and click [Add Subroutine], then the New Program window pops up.



2. Input the program name in the New program window with a maximum of 16 characters.



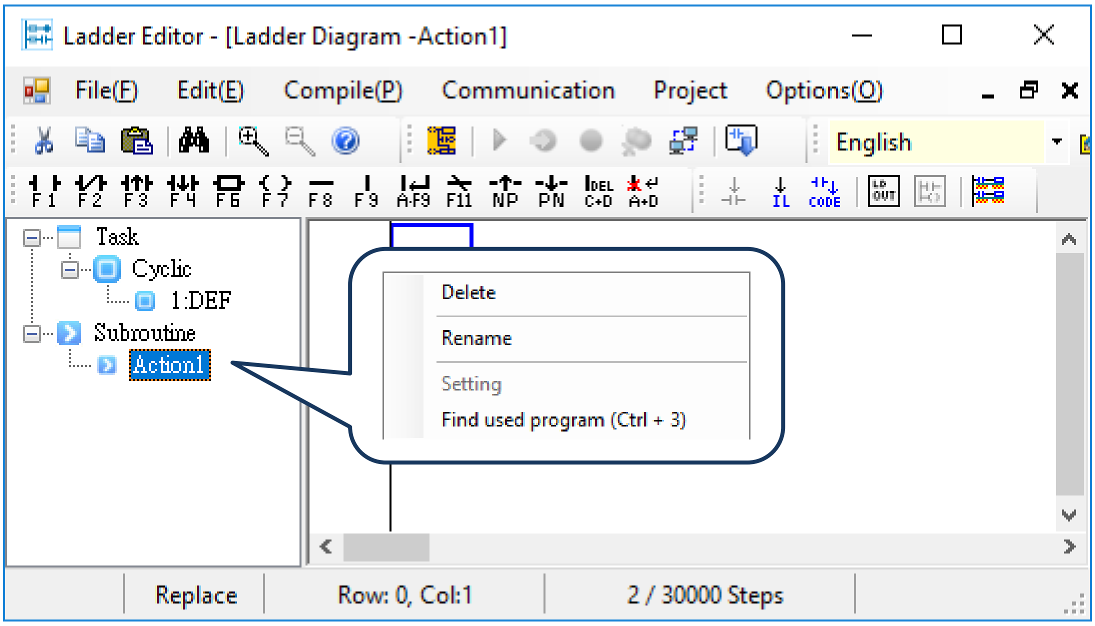
3. When done inputting, click **OK** to complete adding the subroutine.



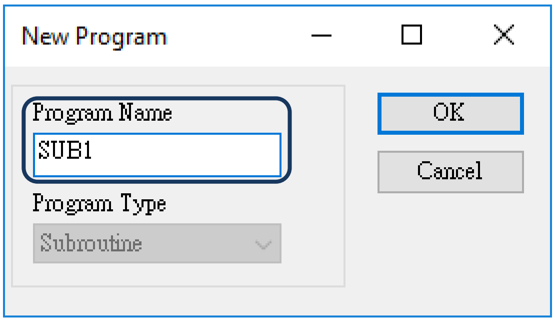
4

Change the program name

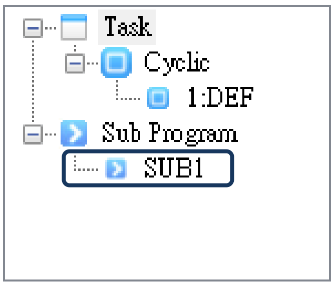
1. Right-click the program name and click [Rename], then the New Program window pops up.



2. Input the new program name in the New Program window.

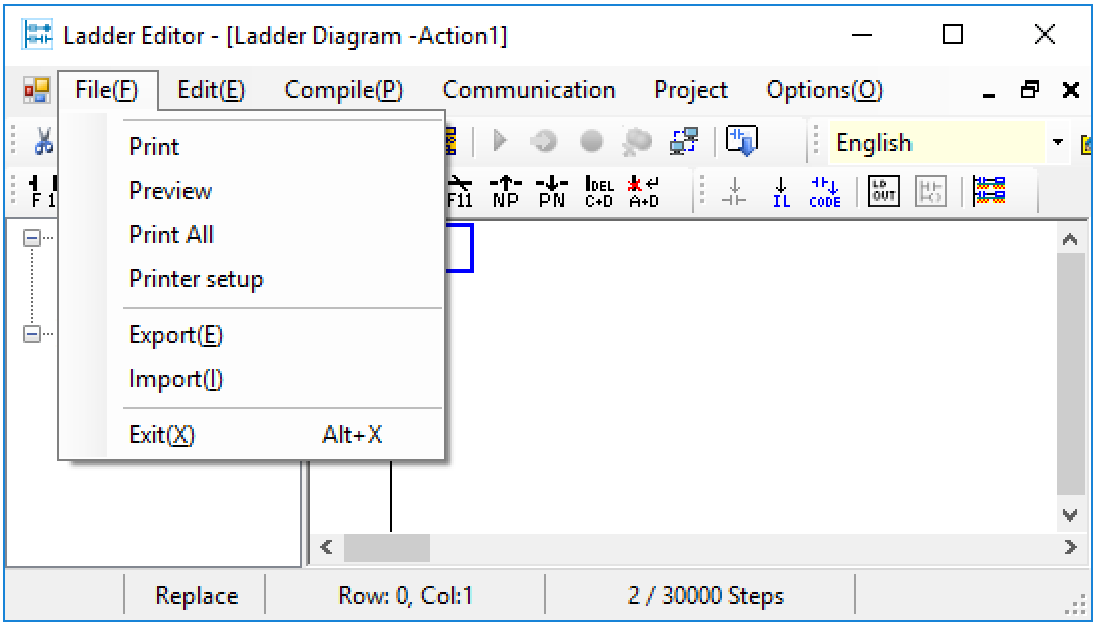


3. When done inputting, click **OK** to complete the program name change. If there is an instruction to call this subroutine in the ladder program, the called subroutine name will also change accordingly.



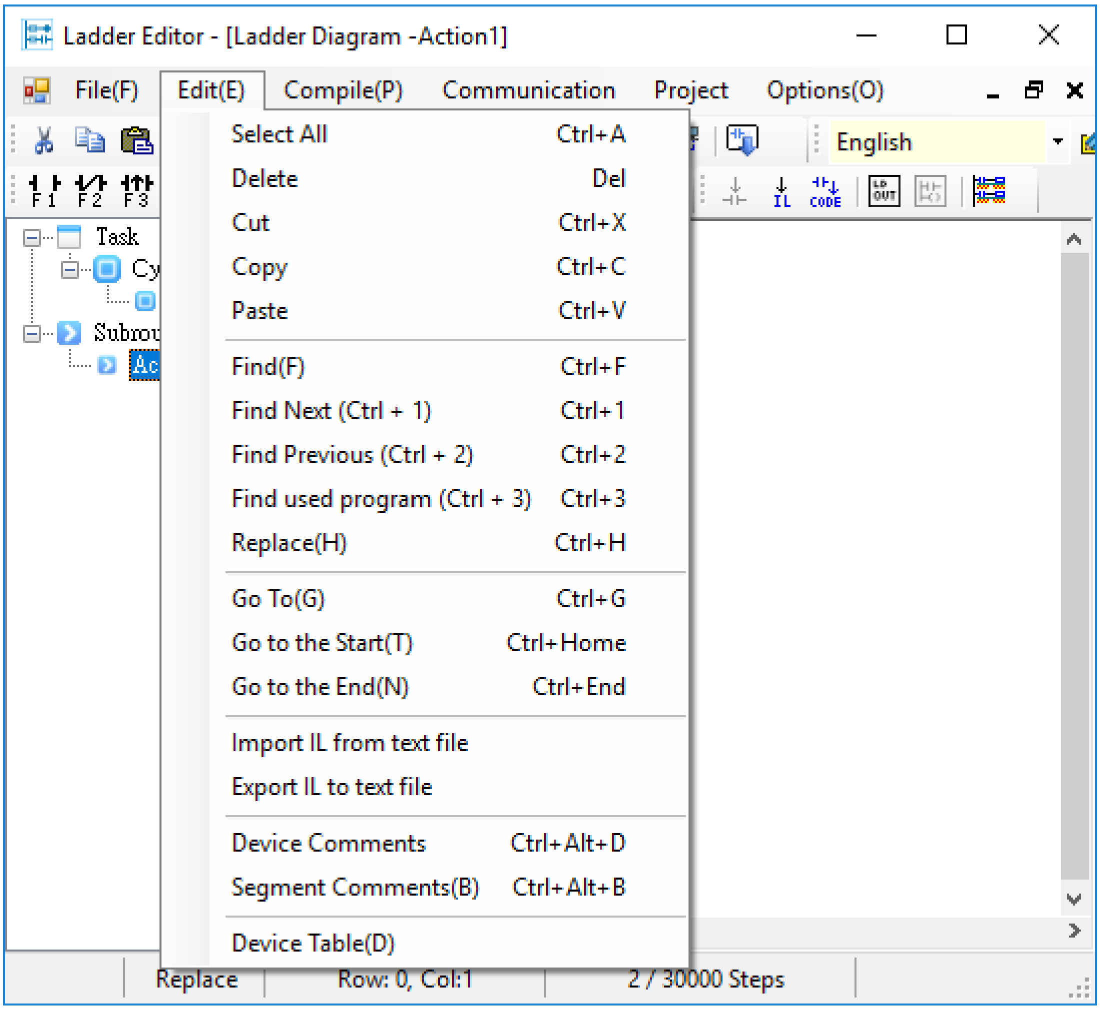
4.3 Other functions

File



Item	Description
Save	Save the current ladder program.
Print	Print the ladder program you are currently editing.
Preview	Print preview the ladder program you are currently editing.
Print All	Print all the unencrypted ladder programs.
Printer setup	Set the print format, including paper size, border, orientation, etc.
Export	Export ladder program (.cwp).
Import	Import external ladder program (.cwp).
Exit	Exit Ladder Editor.

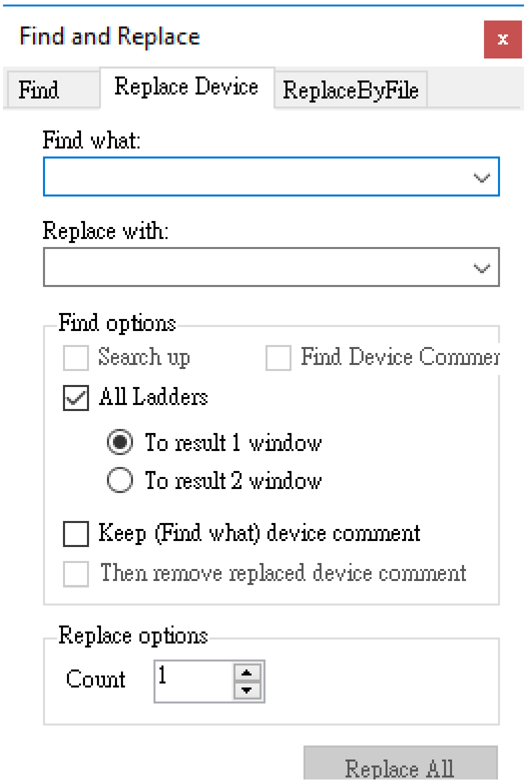
Edit



Item	Description
Select All	Select all contents in the current ladder program.
Delete	Delete the selected content.
Cut	Cut the selected content.
Copy	Copy the selected content.
Paste	Paste the selected content.
Find	Find the target in the current program or all programs.
Replace	Find the target in the current program or all programs and replace with the specified device.
Go To	Jump to the specified STEP position
Go to the Start	Jump to the STEP 0 position in the editing program.
Go to the End	Jump to the END instruction position in the editing program.
Device Comments	Edit the device comments.
Segment Comments	Edit the segment comments.
Row Comments	Edit the row comments.
Device Table	Open the Device Table window.

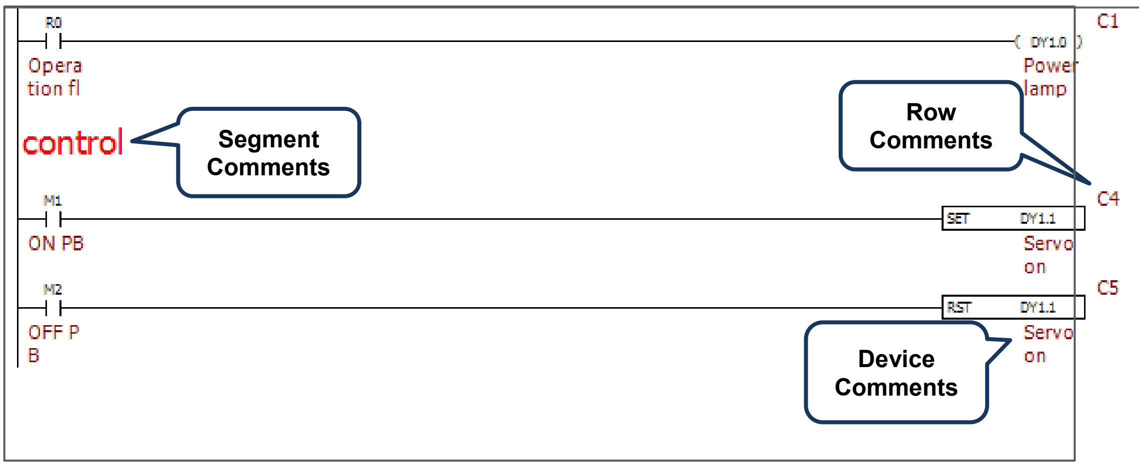
The details of each function are described below:

■ Replace

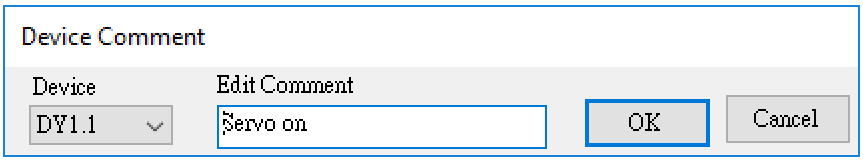


Item	Description
Find what	Input the device you are looking for.
Replace with	Input the replacing device.
All Ladders	The search range is in the current program or all programs.
To result 1 window / To result 2 window	Select to output the result to result 1 window or result 2 window.
Keep (Find what) device comment	Copy the device comment of [Replace with] to that of [Find what], and keep the device comment of [Find what].
Then remove replaced device comment	Copy the device comment of [Replace with] to that of [Find what], and remove the device comment of [Find what].
Replace options	Set the number of replaced devices.

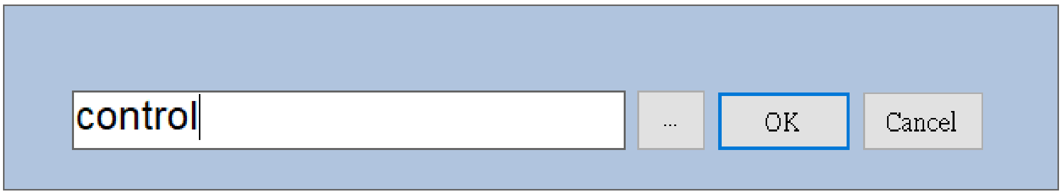
■ Device Comments / Segment Comments / Row Comments



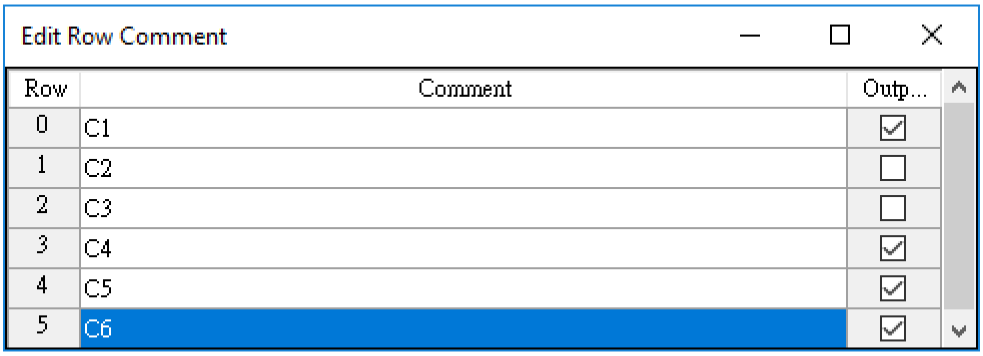
After selecting the device, click [Edit] > [Device Comments] to open the editing window.



After selecting a blank row, click [Edit] > [Segment Comments] to open the editing window.

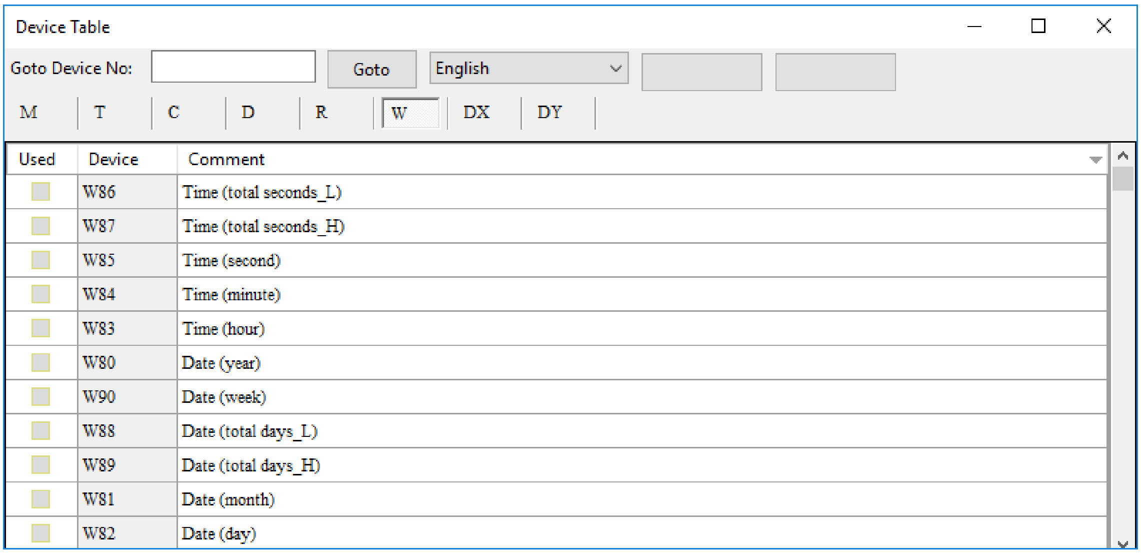


Select [Edit] > [Row Comments] to open the row editing window.



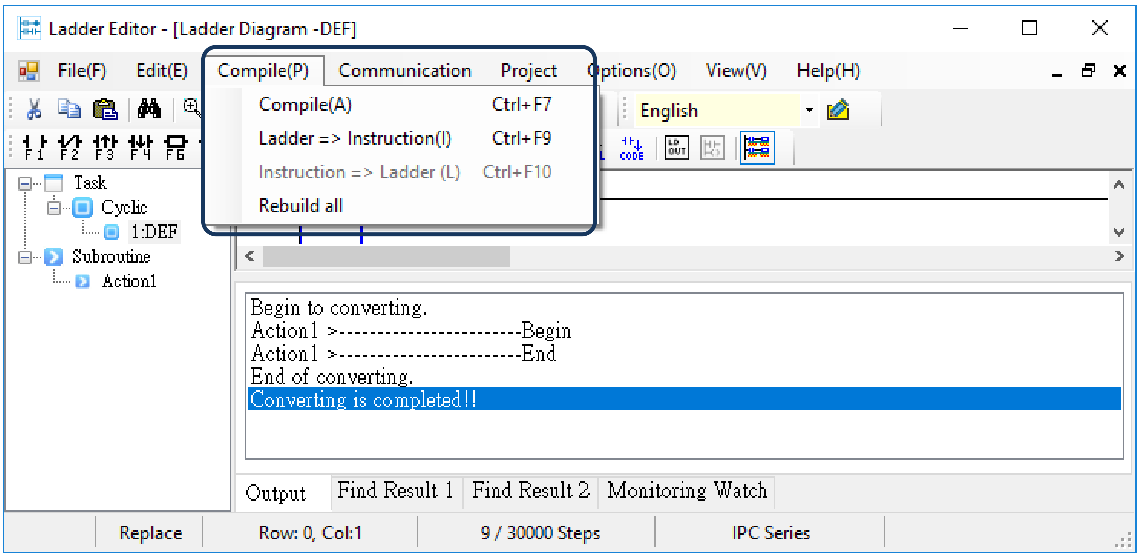
■ Device Table

This table shows all the selected devices and comments, and it also allows you to directly edit the comments.



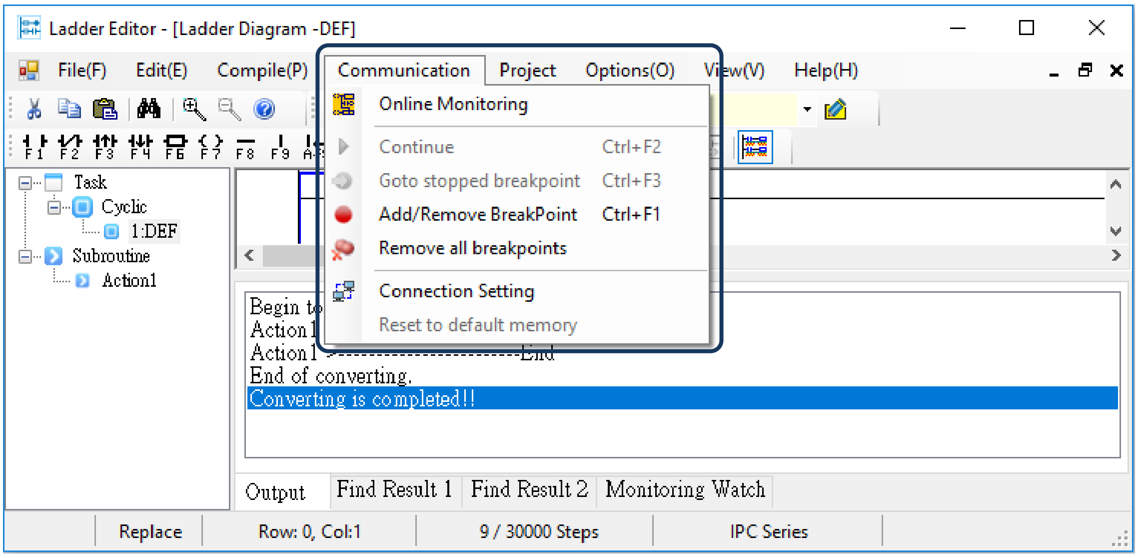
4

Compile



Item	Description
Compile	Compile all programs.
Ladder => Instruction	Compile the ladder diagrams into instructions.
Instruction => Ladder	Compile the instructions into ladder diagrams.

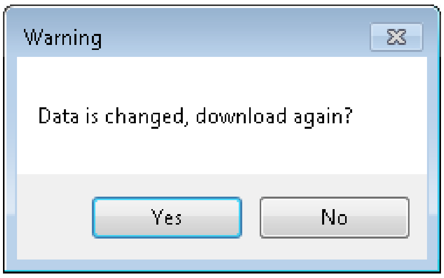
Communication



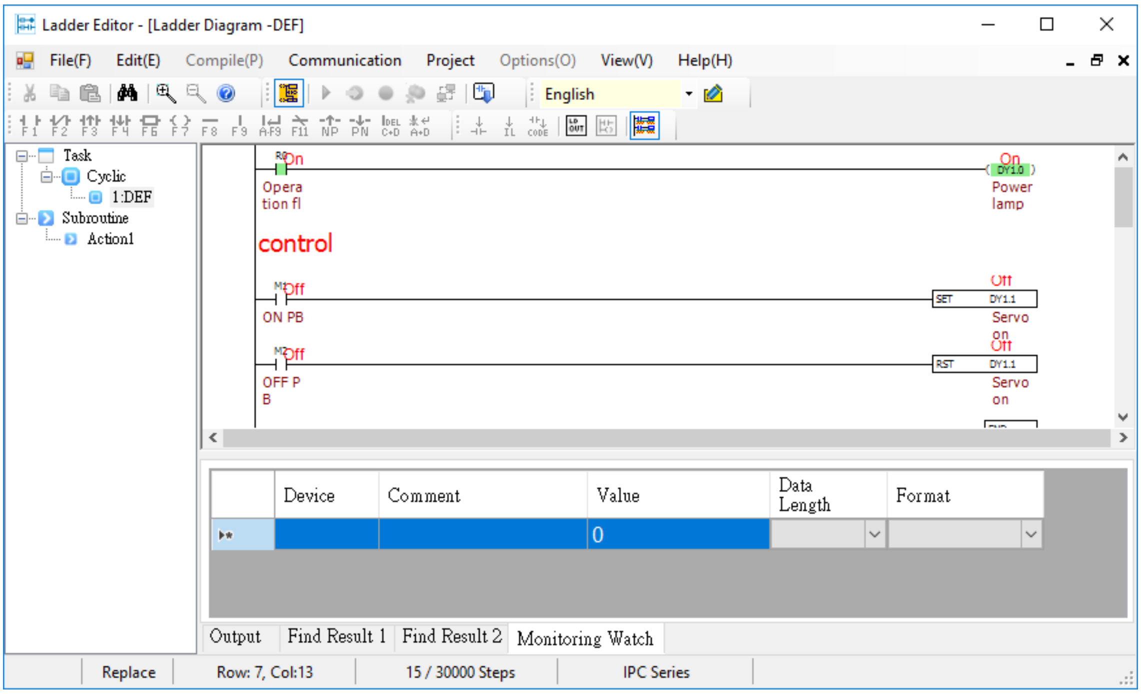
Item	Description
Online Monitoring	Online monitor the execution of the IMP / ladder programs through Ethernet.
Connection Setting	Ethernet setting for the IMP connection.
Reset to default memory	Reset the setting values of the device to the factory default.

■ Online Monitoring

Connect to the PC which has IMP installed according to the connection setting. Then, compile the ladder program before starting online monitoring. The IMP internal program and the editing software are also compared to make sure they are identical. If not, a warning message will pop up as the figure below.



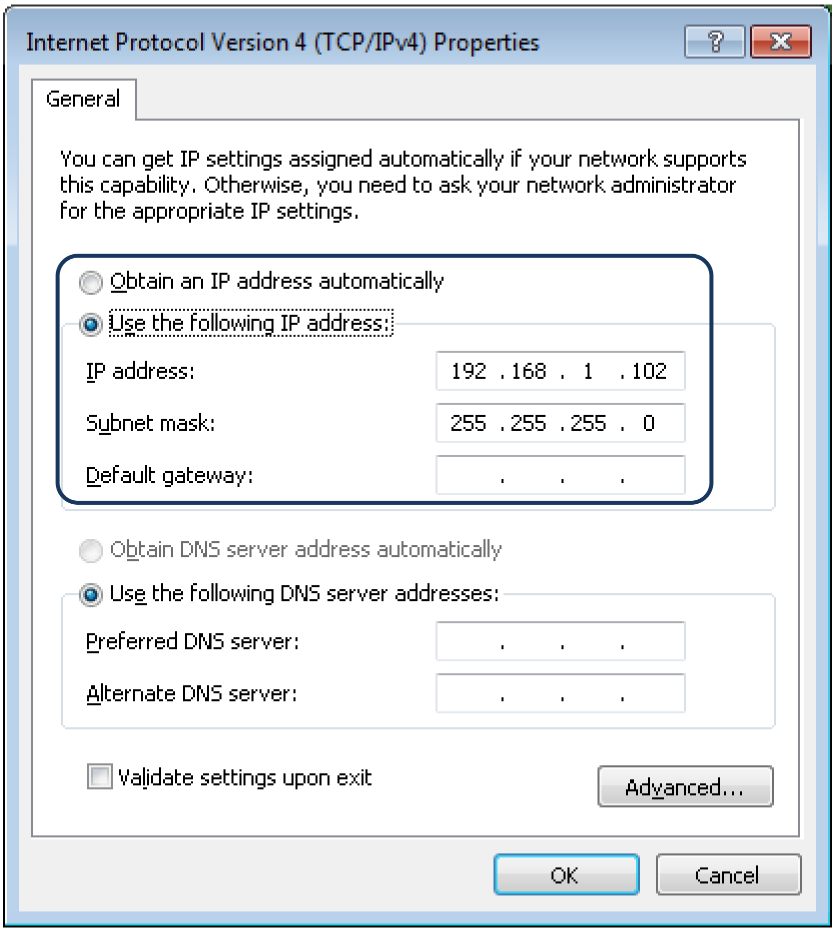
Once the connection is successful, you can start monitoring the execution status of the ladder diagram.



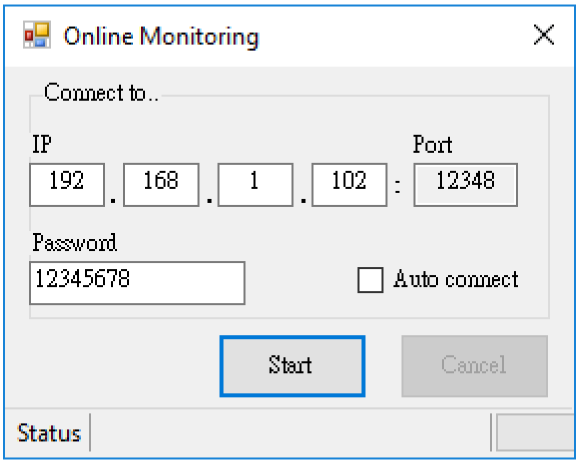
4

■ Connection Setting

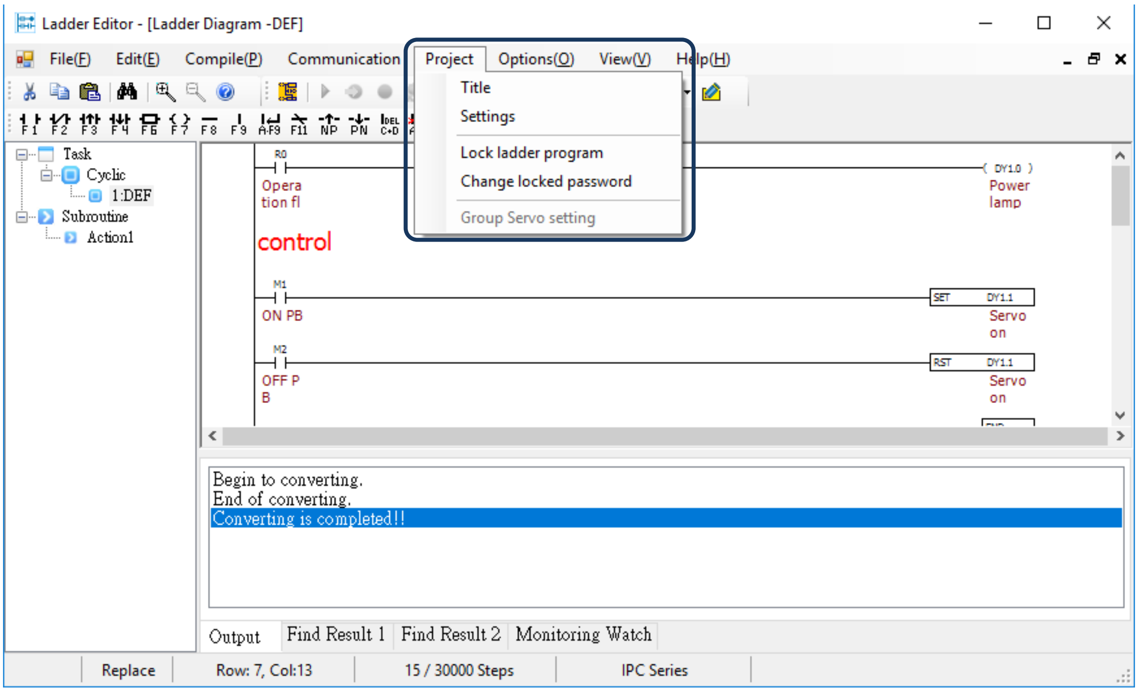
In DOPSoft, go to [Options] > [Communication Setting] to open the communication setting window. Next, enter the IP address as shown in the figure below (if the monitoring PC is in the same network), and check [Network application]. Then, download the screen to the IMP system. Please note that the IP setting of the monitoring PC must be in the same network as the IMP PC.



Enter the IP address of the IMP PC for connection, password, and port number (default: 12348).



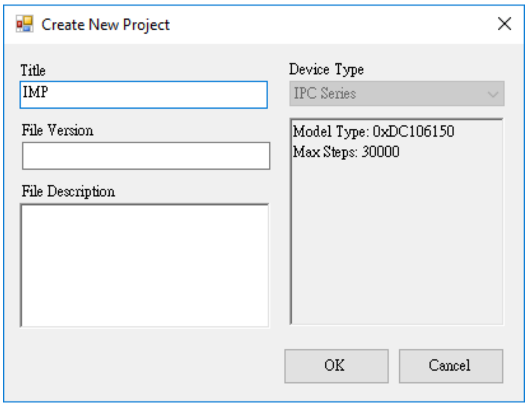
Project



Item	Description
Title	Set the project information, such as project version.
Settings	Set the auto save function to save the ladder program periodically.
Lock ladder program	Lock the selected ladder diagram, so it cannot be opened or edited.
Change locked password	Change the locked password.
Group Servo setting	Apply the servo configuration settings.

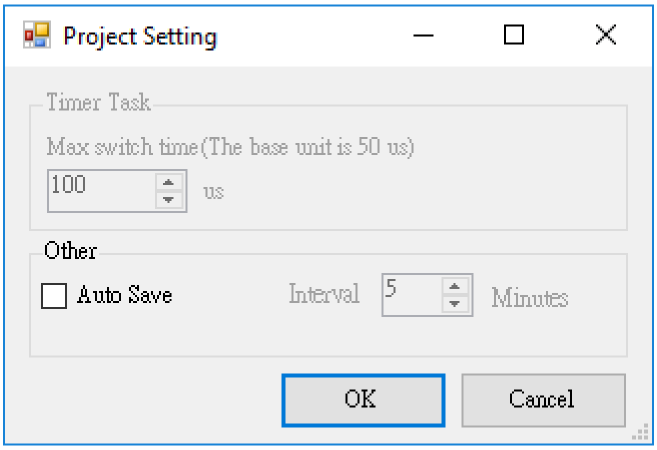
■ Title

You can input the project title, file version, and file description.



■ Settings

Check [Auto Save] and the project will automatically save the ladder program periodically.



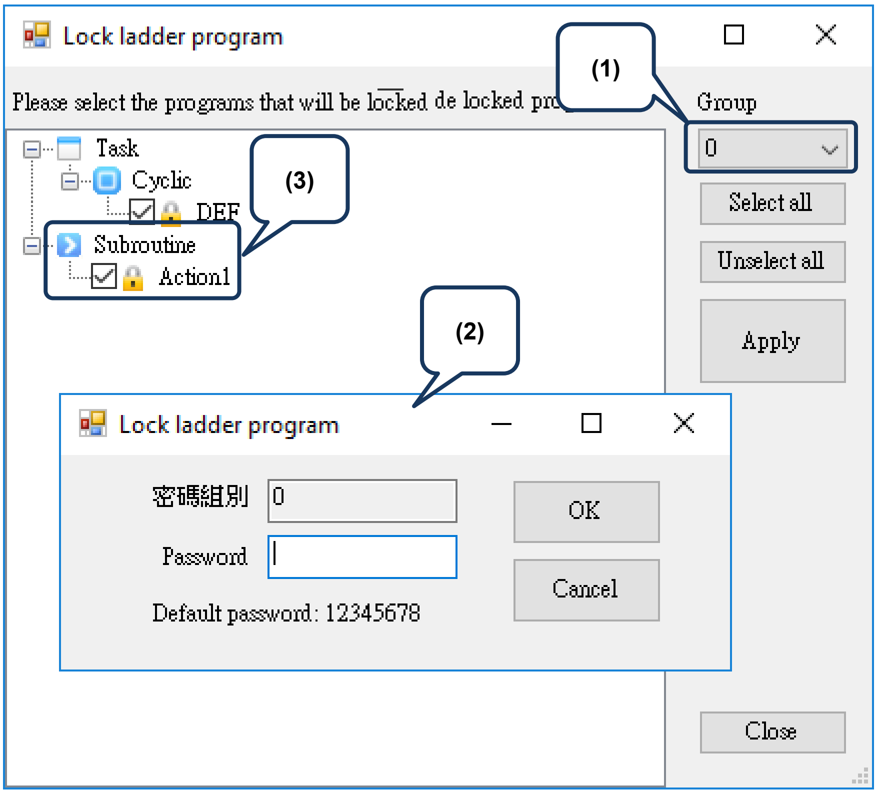
■ Lock ladder program

This function allows you to encrypt the ladder program. Follow the steps below to do so.

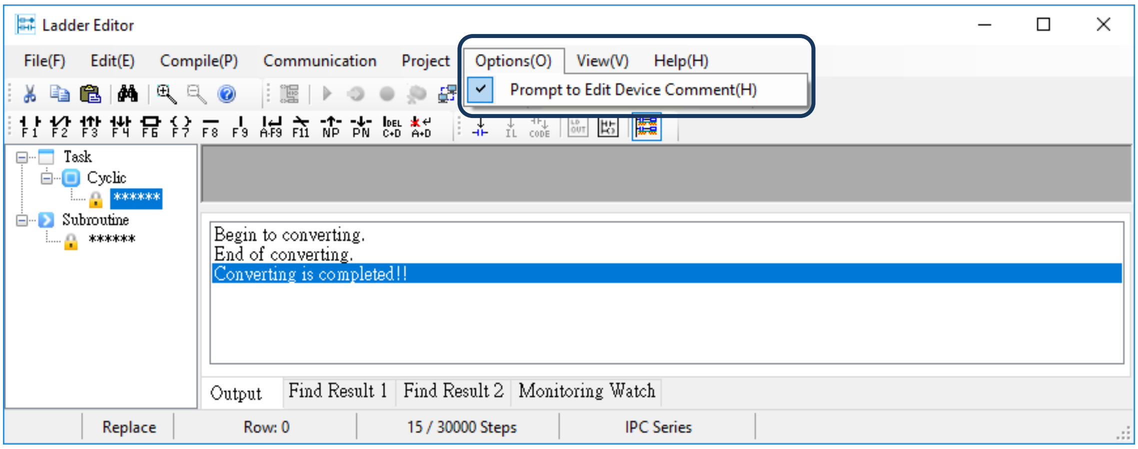
(1) Select the password group to be used.

(2) Input the password and click **OK**.

(3) Check the ladder programs for encryption and click **Apply**. Then, you will not be able to open or edit these locked ladder programs in the editing section.

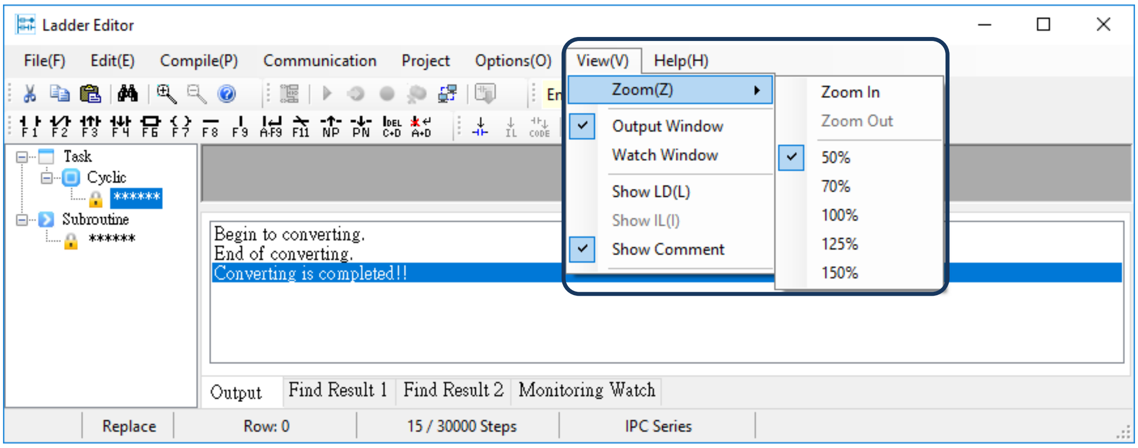


Options



Item	Description
Prompt to Edit Device Comment	Once you input an instruction, the software auto checks if the device comment exists. If there is no comment, the comment input window will automatically pop up.

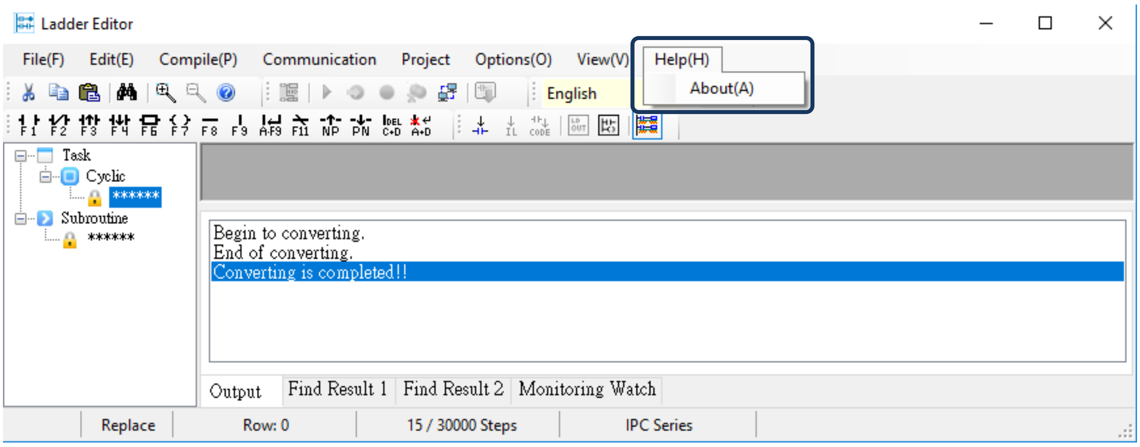
View



Item	Description
Zoom	The content in the editing window can be zoomed in and out to 50%, 70%, 100%, 125%, or 150%.
Output Window	Display the output window.
Watch Window	Display the monitor window.
Show LD	Display the ladder diagram.
Show IL	Display the instruction list.
Show Comment	Display the device comments and row comments.

4

Help



Item	Description
About	Show information about the Ladder Editor software version.

Memory Device 5

When applying relevant devices of IMP, refer to the device setting range and specifications in this chapter to ensure normal operation.

5

5.1 Device table

The following is a list of the devices used by the IMP and their corresponding setting ranges:

<table>
<tr><th>Type</th><th>Device</th><th colspan="3">Item</th><th colspan="2">Corresponding setting range</th><th>Value</th></tr>
<tr><td rowspan="11">PLC relay</td><td>DX</td><td colspan="3">Input relay</td><td>1.0 - 36.63</td><td>2304 points</td><td>0 - 1</td></tr>
<tr><td>DY</td><td colspan="3">Output relay</td><td>1.0 - 36.63</td><td>2304 points</td><td>0 - 1</td></tr>
<tr><td rowspan="3">M</td><td rowspan="3">Auxiliary relay</td><td colspan="2" rowspan="2">General</td><td>0 - 1023</td><td rowspan="3">65536 points</td><td rowspan="3">0 - 1</td></tr>
<tr><td>4096 - 65535</td></tr>
<tr><td colspan="2">Non-volatile</td><td>1024 - 4095</td></tr>
<tr><td rowspan="2">T</td><td rowspan="2">Timer</td><td colspan="2">100 ms</td><td>0 - 199, 256 - 767</td><td rowspan="2">1024 points</td><td rowspan="2">0 - 1</td></tr>
<tr><td colspan="2">10 ms</td><td>200 - 255,
768 - 1023</td></tr>
<tr><td rowspan="2">C</td><td rowspan="2">Counter</td><td colspan="2">16 bits</td><td>0 - 199</td><td rowspan="2">256 points</td><td rowspan="2">0 - 1</td></tr>
<tr><td colspan="2">32 bits</td><td>200 - 255</td></tr>
<tr><td rowspan="2">R</td><td rowspan="2">Special relay</td><td colspan="2">For PLC</td><td rowspan="2">0 - 65535</td><td rowspan="2">65536 points</td><td rowspan="2">0 - 1</td></tr>
<tr><td colspan="2">For motion mode</td></tr>
<tr><td rowspan="10">PLC register</td><td rowspan="2">T</td><td rowspan="2">Timer current value</td><td colspan="2">100 ms</td><td>0 - 199, 256 - 767</td><td rowspan="2">1024 points</td><td rowspan="2">0 - 65535</td></tr>
<tr><td colspan="2">10 ms</td><td>200 - 255,
768 - 1023</td></tr>
<tr><td rowspan="2">C</td><td rowspan="2">Counter current value</td><td colspan="2">16 bits</td><td>0 - 199</td><td rowspan="2">256 points</td><td>0 - 65535</td></tr>
<tr><td colspan="2">32 bits</td><td>200 - 255</td><td>-2147483648 to 2147,483,647</td></tr>
<tr><td rowspan="3">D</td><td rowspan="3">Data register</td><td rowspan="3">16 bits</td><td rowspan="2">General</td><td>0 - 1023</td><td rowspan="3">65536 points</td><td rowspan="3">-32768 to 32767</td></tr>
<tr><td>4096 - 65535</td></tr>
<tr><td>Non-volatile</td><td>1024 - 4095</td></tr>
<tr><td>V</td><td>Indirect register</td><td colspan="2">16 bits</td><td>0 - 127</td><td>128 points</td><td>-32768 to 32767</td></tr>
<tr><td rowspan="2">W</td><td rowspan="2">Special register</td><td rowspan="2">16 bits</td><td>For PLC</td><td rowspan="2">0 - 65535</td><td rowspan="2">65536 points</td><td rowspan="2">-32768 to 32767</td></tr>
<tr><td>For motion mode</td></tr>
<tr><td rowspan="3">HMI register</td><td>$M</td><td colspan="3">Auxiliary register</td><td>0 - 1023</td><td>1024 points</td><td>-32768 to 32767</td></tr>
<tr><td>$</td><td colspan="3">Auxiliary register</td><td>0 - 65535</td><td>65536 points</td><td>-32768 to 32767</td></tr>
<tr><td>*$</td><td colspan="3">Pointer register</td><td>0 - 65535</td><td>65536 points</td><td>-32768 to 32767</td></tr>
<tr><td>Pointer</td><td>P</td><td colspan="3">Jump pointer</td><td>0 - 255</td><td>256 points</td><td>-</td></tr>
<tr><td>Constant</td><td>K</td><td colspan="3">Decimal constant</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>Floating point</td><td>F</td><td colspan="3">Floating point</td><td>-</td><td>-</td><td>-</td></tr>
</table>

The following sections will describe the definition and setting range for each device.

5

5.1.1 Input relay (DX) / output relay (DY)

Input / output relays are numbered in decimal form. The input relay (DX) and output relay (DY) correspond to the input and output points of the DMCNET or EtherCAT module respectively. The corresponding addresses are as follows. For the operation of the local I/O, refer to Appendix A for the description of R6200 and R6300 special relays.

Device	DMC-RMxx(MN\NT\PT), HMC-RIO3232RT5, R1-EC60xxD0, R1-EC70xxD0			
	Node 1	Node 2	…	Node 36
Input (DX)	DX1.0 - DX1.63	DX2.0 - DX2.63	…	DX36.0 - DX36.63
Output (DY)	DY1.0 - DY1.63	DY2.0 - DY2.63	…	DY36.0 - DY36.63

Note:

1. DMCNET bus does not support ASD-DMC-GE16MN and ASD-DMC-GE16NT modules.
2. The EtherCAT input relay can correspond to the DI module of R1-EC6002D0, R1-EC6012D0, R1-EC6022D0, and R1-EC6032D0; and the output relay can correspond to the DO module of R1-EC7062D0.

Input relay (DX)

The input relay (DX) is connected through the communication bus to read the input signal of the DI module. In the program, there is no limit to the number of A or B contacts for each input relay. The ON / OFF state of the input relay (DX) will only follow the ON / OFF status of the external input device.

Output relay (DY)

After the PLC program operation is completed, the load of the output relay (DY) is driven through the DO module. In the program, there is no limit to the number of A or B contacts for each output relay.

5

5.1.2 Auxiliary relay (M)

Auxiliary relay is a memory shared by the motion program macro and PLC. Both auxiliary relay (M) and output relay (DY) have output coils and contact A and B, and there is no limit to the number of contacts. You can use the auxiliary relay (M) to form a control circuit, but cannot directly drive the external load. There are two types of auxiliary relays:

Type	General	Non-volatile
Number range	M0 - M1023, M4096 - M65535	M1024 - M4095
Description	When the power is cut off, all the states will be reset to off, and the states will remain off when power on again.	When the power is cut off, all the states will be maintained, and the states will remain the same when power on again.[Note]

Note: when using a non-Delta PAC product with motion card, the non-volatile data is stored in the hard disk. In the PLC program, frequent changes to the content value of the non-volatile type auxiliary relay will shorten the life of the hard disk.

5.1.3 Timer (T)

Timer (T) is numbered in decimal form and categorized into two types by unit.

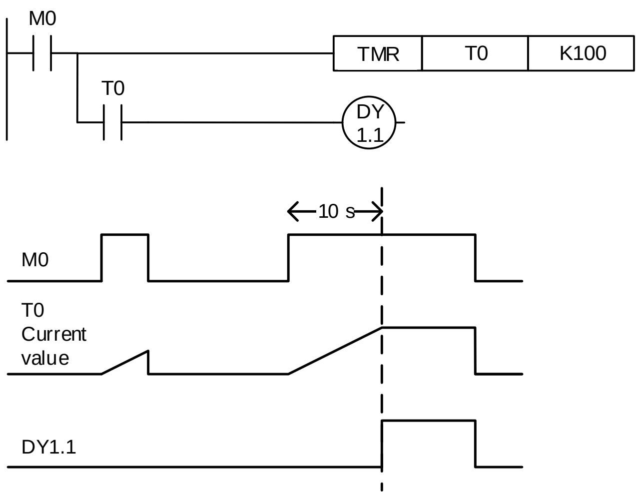
Type	100 ms for general use	10 ms for general use
Number range	0 - 199, 256 - 767	200 - 255, 768 - 1023

Note: time drift may occur due to allocation of the operating system resources.

The timer counts by 10 ms or 100 ms in progressive manner. When the current value of the timer equals the set value, the output coil is turned on. The setting value can start with K (DEC) or D (data register).

Actual set time of the timer = Time unit x Set value.

The timer starts counting when the TMR command is executed, and the output coil is turned on once the timer reaches the set value. When TMR command is stopped, the current value of the timer resets to 0 and the output coil is disconnected (as shown below).



When M0 is on, the current value of the timer T0 increases by 100 ms, and once the current value equals the set value K100 (10 seconds), the output coil T0 is on. When M0 is off or the power is off, the current value of the timer T0 resets to 0 and the output coil T0 is turned off.

5.1.4 Counter (C)

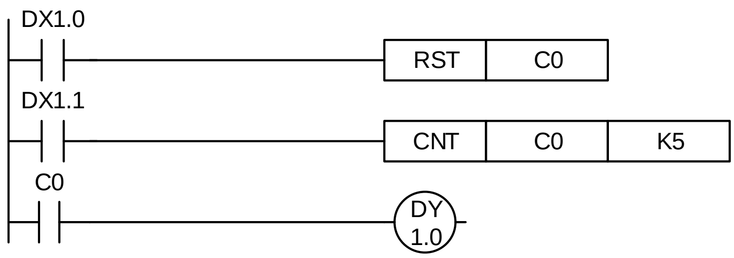
Counter (C) is numbered in decimal form and categorized into two types according to the length of the bit. The range of the numbers is as follows:

Type	16-bit for general use	32-bit for general use
Number range	0 - 199	200 - 255
Setting value	0 - 65,535	0 - 2,147,483,647
Type of the setting value	Constant K or data register D	Constant K or data register D (specify 2)
State of current value	Stop counting when reaching the set value.	
Output contact	When the counter reaches the set value, the contact is turned on and remains.	
Homing	When executing the RST command, the current value resets to 0 and the contact switches to off.	

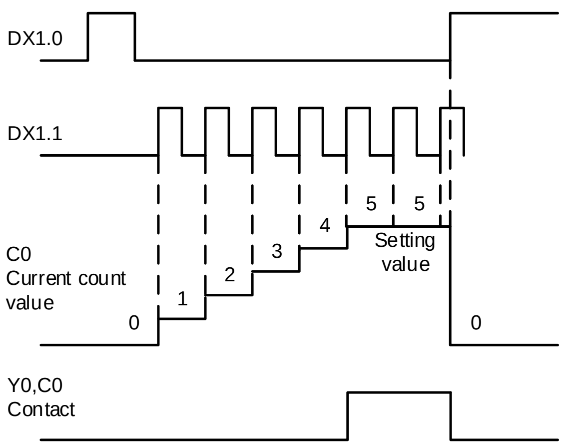
When the CNT command of the counter is in execution (triggered by the rising-edge of the signal from off to on), the counter increments by 1. If the current value of the counter equals the set value, the counter output bit is on. If the counter setting value is in DEC (start with K) and the setting value is 0 or 1, the counter output bit switches to on when the CNT command is triggered for the first time. The setting value of the counter can be set by constant K or by the value in register D.

Example:

LD	DX1.0
RST	C0
LD	DX1.1
CNT	C0 K5
LD	C0
OUT	DY1.0



When DX1.0 is on, execute the RST command, then the C0 count resets to zero and the counter C0 bit is off. When DX1.1 switches from off to on, the current value of counter C0 increments by 1. When the count of counter C0 reaches the set value K5, the counter C0 output bit is on and the current value of C0 equals the set value K5. After that, C0 no longer receives the triggering signal of DX1.1 and the current value of C0 remains at K5. See the diagram below for the example:



5.1.5 Data register (D)

Data register number

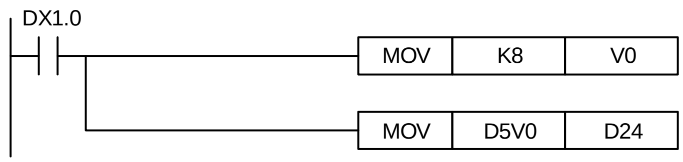
Data register (D) is the shared memory of PLC and MPM. The data length is 16 bits (-32,768 to 32,767), the highest bit is either a positive or negative sign, and the register can save the value data of -32,768 to +32,767. You can also merge two 16-bit registers into one 32-bit register (such as D+1 and D). The small number (D) is for specifying the low bit and the highest bit (D+1) is for specifying the positive or negative sign, and this register can save value data of -2,147,483,648 to +2,147,483,647. Data register (D) is categorized into two types and the range of the numbers is as follows:

Type	General	Non-volatile
Number range	D0 - D1023, D4096 - D65535	D1024 - D4095
Description	The content is cleared to 0 when power is off.	When power is off, its content is not affected; and when powering on again, the content remains unchanged.[Note]

Note: when using a non-Delta PAC product with motion card, the non-volatile data is stored in the hard disk. In the PLC program, frequent changes to the content value of the non-volatile type register will shorten the life of the hard disk.

5.1.6 Indirect register (V)

Indirect register (V) is a 16-bit register with a range of 128 points from V0 to V127. Indirect register is the same as the general data register, which is a 16-bit data register. You can read and write with the indirect register, but if you use it as a general register, it can only be used for 16-bit commands.



When DX1.0 is on, V0 = 8, D5V0 = D(5+8) = D13, the content of D13 is moved to D24.

5.1.7 HMI auxiliary register

$ register is an auxiliary register specialized for the HMI. Its data format is 16-bit and can save value data of -32,768 to +32,767. You can only access the auxiliary register through the HMI interface. You can use this auxiliary relay when writing HMI macro programs and elements. You can also set double word through the macro command (DW) or element value to combine two $ registers into a 32-bit register which can save value data of -2,147,483,648 to +2,147,483,647. Auxiliary register $M is not equipped with the non-volatile function and its data length is 16 bits, which can save value data of -32,768 to +32,767. The $ and $M auxiliary registers can only be accessed through the HMI interface.

5.1.8 Constant (K) / Floating point (F)

PLC can perform computing with two types of values. The tasks and functions of the values are described below. The computing and saving of the internal values are done in binary format. See the binary values and terminologies in the table below:

Bit	The basic unit of binary values, which is either 1 or 0.
Nibble	It consists of 4 consecutive bits (such as bit0 - bit3). And it can represent 0 - 15 in decimal form or 0 - F in hexadecimal form.
Byte	It consists of 2 consecutive nibbles which equal 8 bits (such as bit0 - bit7). And it can represent 00 - FF in hexadecimal form.
Word	It consists of 2 consecutive bytes which equal 16 bits (such as bit0 - bit15). And it can represent 4 nibbles in hexadecimal form 0000 - FFFF.
Double word	It consists of 2 consecutive words which equal 32 bits (such as bit0 - bit31). And it can represent 8 nibbles in hexadecimal form 00000000 - FFFFFFFF.

Constant K

The decimal value is usually represented by a "K" in front of the value. For example, K100 is a value of 100 in decimal form.

When bit device DX, DY, or M is used with the prefix K, the data format will become the form of nibble, byte, word, or double word. For example: K2DY1 and K4M100. K1 here represents a combination of 4 bits, and K2 - K4 represent combinations of 8, 12, and 16 bits respectively.

Floating point F

The floating point value is used as an operand in the application command, for example FADD F12.3 F0 D0 (F floating point constant).

5

5.2 System special relay

While the device functions mentioned in Section 5.1 are user-defined, the functions of the system special relay are preset by the system. The following sections will describe the system special relays (including PLC and motion status type) in detail.

5.2.1 PLC system special relay

This type of relay can be used to acquire the system status, including computing result, error monitoring, connection of peripheral devices, and triggering of physical buttons.

Type	No.	Function	Description	Property	Non-volatile
Operation flag	R0	Operation flag (Contact a)	This contact remains on while the controller is operating.	R	NO
	R1	Operation flag (Contact b)	This contact remains off while the controller is operating.	R	NO
	R4	Initial pulse	This bit is on during the first PLC cycle.	R	NO
Clock pulse	R13	0.5-second clock square pulse	When PLC is operating, this bit continues to cycle in a state of on for 0.5 second and off for 0.5 second. Note: clock drift may occur.	R	NO
	R14	1-second clock square pulse	When PLC is operating, this bit continues to cycle in a state of on for 1 second and off for 1 second. Note: clock drift may occur.	R	NO

5.2.2 Motion status special relay

This type of relay can be used to acquire the system status and relevant settings, including information about versions and controller system, error code, and peripheral devices.

■ **Motion control special relay**

Corresponding servo control address of each axis is shown in the following table (you can also refer to Appendix A):

Function	Description	Property	Bus	Axis 1	Axis 2	Axis 3	~	Axis 36
SVON control	Control servo on.	R / W	D/E	R10151	R10251	R10351	~	R13651
Software limit enabling bit	Control software limit.	R / W	D/E	R10152	R10252	R10352	~	R13652
Acceleration unit control bit	Switch acceleration time.	R / W	D/E	R10153	R10253	R10353	~	R13653
Torque limit enabling bit	Control bit.	R / W	D	R10160	R10260	R10360	~	R13660
Motion curve setting	Switch acceleration curve.	R / W	D	R10161	R10261	R10361	~	R13661
JOG direction control	Control motion direction.	R / W	D/E	R10162	R10262	R10362	~	R13662
Torque limit enabling bit in speed mode	Control bit.	R / W	D	R10163	R10263	R10363	~	R13663
Speed limit enabling bit in torque mode	Control bit.	R / W	D	R10164	R10264	R10364	~	R13664

■ **Motion status special relay**

IMP is able to control servo motions of up to 36 axes simultaneously. Corresponding servo status address of each axis is shown in the following table (you can also refer to Appendix A):

Function	Description	Property	Bus	Axis 1	Axis 2	Axis 3	~	Axis 36
Servo operation mode	Display current motion. Refer to the servo operation mode table.	R	D/E	R10100	R10200	R10300	~	R13600
		R	D/E	R10101	R10201	R10301	~	R13601
		R	D/E	R10102	R10202	R10302	~	R13602
		R	D/E	R10103	R10203	R10303	~	R13603
DI3 status (servo)	Mapping of DI3 (SLD) state.	R	D	R10104	R10204	R10304	~	R13604
Servo alarm flag	Alarm message.	R	D/E	R10105	R10205	R10305	~	R13605
SVON monitoring flag	Display motor excitation state.	R	D/E	R10108	R10208	R10308	~	R13608
Servo error flag	Display servo drive error.	R	D/E	R10109	R10209	R10309	~	R13609
Positioning complete flag	Motion command is completed.	R	D/E	R10110	R10210	R10310	~	R13610
Servo operation mode MSD0	Mode specific which displays the servo operation status.	R	D	R10112	R10212	R10312	~	R13612
Servo operation mode MSD1		R	D	R10113	R10213	R10313	~	R13613
Flag for triggering servo limit	Display that it has reached the positive limit.	R	D/E	R10114	R10214	R10314	~	R13614
	Display that it has reached the negative limit.	R	D/E	R10115	R10215	R10315	~	R13615

5

Function	Description	Property	Bus	Axis 1	Axis 2	Axis 3	~	Axis 36
Software limit trigger bit	Display that it has exceeded the software positive limit.	R	D/E	R10116	R10216	R10316	~	R13616
	Display that it has exceeded the software negative limit.	R	D/E	R10117	R10217	R10317	~	R13617
Software limit failed warning bit	Display software positive limit failed.	R	D/E	R10118	R10218	R10318	~	R13618
	Display software negative limit failed.	R	D/E	R10119	R10219	R10319	~	R13619
Ready to Switch On	-	R	E	R10120	R10220	R10320	~	R13620
Operation Enabled	-	R	E	R10121	R10221	R10321	~	R13621
Voltage Disabled	-	R	E	R10122	R10222	R10322	~	R13622
Quick Stop	-	R	E	R10123	R10223	R10323	~	R13623
Switch On Disable	-	R	E	R10124	R10223	R10324	~	R13624
Homing completed	Display the homing status after initialization.	R	D/E	R10130	R10230	R10330	~	E13630
Flag for motion in process	Display motion status.	R / W	D/E	R10165	R10265	R10365	~	R13665

Servo operation mode table:

Bit				Motion mode		
0	1	2	3	DMCNET	EtherCAT	Note
ON	OFF	OFF	OFF	Position control (PP)	Position control (PP)	In these modes, after the motion card issues the command, the servo motor will then execute the command, such as PR mode, motion with fixed speed, and motion with fixed torque.
OFF	ON	OFF	OFF	-	Speed control (VL)	
ON	ON	OFF	OFF	Speed control (PV)	Speed control (PV)	
OFF	OFF	ON	OFF	Torque control (PT)	Torque control (PT)	
ON	OFF	ON	OFF	-	-	
OFF	ON	ON	OFF	Homing	Homing	
OFF	OFF	OFF	ON	-	Position control (CSP)	Cycle synchronous operation
ON	OFF	OFF	ON	-	Speed control (CSV)	
OFF	ON	OFF	ON	-	Torque control (CST)	
ON	ON	ON	OFF	-	IP (EtherCAT)	The motion card updates the servo motor target position every communication cycle, and the servo motor follows the continuously updated target position.
ON	ON	ON	ON	IP (DMCNET)	-	

Description about the mode specific bit of the motion status:

Mode	Mode specific	
	If MSD1 displays 1	If MSD0 displays 1
Homing mode	An error has occurred when homing.	Homing can be executed.
DMCNET mode	N/A	Mode enabled.

■ **Special register for single-axis motion control**

Corresponding servo control address of each axis is shown in the following table (you can also refer to Appendix A):

Function	Description	Property	Bus	Axis 1	Axis 2	…	Axis 36
Error code of single-axis operation	Display error code of single-axis motion control: 1: servo is not on. 2: command in execution. 3: use the variable speed command in motion, but the single axis is not in motion. 4: use the variable position command in motion, but the single axis is not in motion. 5: position has exceeded the software limit. 6: reaching the software limit in motion. 20: the following master axis is not on the same bus. 21: the following master axis does not exist. 22: following the master axis, and the motion command is invalid. 23: invalid command (speed, torque, and homing) for the following master axis. 24: the following function is enabled for the following master axis.	R	D/E	W10150	W10250	…	W13650

Function	Description	Property	Bus	Axis 1	Axis 2	…	Axis 36
Single motion control code	Command code for controlling single-axis motion: 0: N/A. 1: reset to 0. 2: absolute motion. 3: relative motion. 4: JOG (abs_move). 5: JOG (Tv_move). 6: variable speed in motion. 7: variable position in motion. 8: coordinates setting. 9: speed control. 10: torque control. 13: servo alarm reset. 14: emergency stop. 15: decelerate to stop. 20: rotary table rotates one working unit in forward direction. 21: rotary table rotates one working unit in reverse direction. 22: rotary table positioning (absolute positioning, shortest path). 25: enable following control. 26: disable following control. State display code: 27: gantry slave axis status. 28: gantry reset to 0 status. 40: motion table in operation. 50: interpolation in operation. 51: group waiting. 55: MPM in operation. 60: MPG controlling. 70: SNC in operation.	R / W	D/E	W10151	W10251	…	W13651
Acceleration time	Unit: ms	R / W	D/E	W10152 W10153	W10252 W10253	… …	W13652 W13653
Deceleration time	Unit: ms	R / W	D/E	W10154 W10155	W10254 W10255	… …	W13654 W13655
Target speed of motion command	Unit: refer to the setting of motion speed unit	R / W	D/E	W10156 W10157	W10256 W10257	… …	W13656 W13657
Target coordinates of motion command	Unit: 0.001 mm (μm)	R / W	D/E	W10158 W10159	W10258 W10259	… …	W13658 W13659
Homing mode setting	Refer to Appendix B.	R / W	D/E	W10160	W10260	…	W13660
Setting of motion speed unit	Setting of motion speed unit: 0: PUU/sec 1: % 2: mm/min	R / W	D/E	W10161	W10261	…	W13661
First speed in homing mode	Default unit: rpm; you can switch the unit with Quick Start.	R / W	D/E	W10162 W10163	W10262 W10263	… …	W13662 W13663

Function	Description	Property	Bus	Axis 1	Axis 2	…	Axis 36
Second speed in homing mode	Default unit: rpm; you can switch the unit with Quick Start.	R / W	D/E	W10164	W10264	…	W13664
				W10165	W10265	…	W13665
Offset in homing mode	Unit: 0.001 mm (μm)	R / W	D/E	W10166	W10266	…	W13666
				W10167	W10267	…	W13667
Total index number of the rotary table	-	R / W	D/E	W10168	W10268	…	W13668
Target position of the rotary table	-	R / W	D/E	W10169	W10269	…	W13669
Target speed in speed mode	Unit: rpm	R / W	D/E	W10170	W10270	…	W13670
				W10171	W10271	…	W13671
Target torque in torque mode	Unit: ‰	R / W	D/E	W10172	W10272	…	W13672
				W10173	W10273	…	W13673
Torque limit in speed mode	Unit: ‰	R / W	D/E	W10174	W10274	…	W13674
				W10175	W10275	…	W13675
Speed limit in torque mode	Unit: rpm	R / W	D/E	W10176	W10276	…	W13676
				W10177	W10277	…	W13677
Written parameter of the servo	-	R / W	D/E	W10180	W10280	…	W13680
				W10181	W10281	…	W13681
Forward software limit	Unit: μm	R / W	D/E	W10182	W10282	…	W13682
				W10183	W10283	…	W13683
Reverse software limit	Unit: μm	R / W	D/E	W10184	W10284	…	W13684
				W10185	W10285	…	W13685
Servo parameter group and index value	Servo parameter group x 1000 + Servo parameter No.	R / W	D/E	W10186	W10286	…	W13686
Control code for reading / writing servo parameters	Read / write servo parameter control code: 0: N/A. 1: read servo parameters. 2: write servo parameters. 3: read servo commands. 4: set servo monitoring commands.	R / W	D/E	W10187	W10287	…	W13687
Set servo monitoring command code	-	R / W	D/E	W10188	W10288	…	W13688
The following function follows the master axis number	Software No. 1 - 36.	R / W	D/E	W10189	W10289	…	W13689

5

■ Special register for single-axis motion status

Corresponding servo state address of each axis is shown in the following table (you can also refer to Appendix A):

Function	Description	Property	Bus	Axis 1	Axis 2	…	Axis 36
Motor feedback position	Unit: μm (0.001 mm)	R	D/E	W10102	W10202	…	W13602
				W10103	W10203	…	W13603
Motion card command position	Unit: μm (0.001 mm)	R	D/E	W10104	W10204	…	W13604
				W10105	W10205	…	W13605
Target position	Unit: μm (0.001 mm)	R	D/E	W10106	W10206	…	W13606
				W10107	W10207	…	W13607
Servo DI status	-	R	D/E	W10108	W10208	…	W13608
Servo DO status	-	R	D/E	W10109	W10209	…	W13609
Current motion speed of each axis	Unit: μm/s	R	D/E	W10110	W10210	…	W13610
				W10111	W10211	…	W13611
Current output torque of motor	Unit: ‰	R	D/E	W10113	W10213	…	W13613
Motion command status	Command status code: 0: in operation. 1: positioning completed	R	D/E	W10114	W10214	…	W13614
Servo error code	Display error return value of the servo drive.	R	D/E	W10115	W10215	…	W13615
Read servo return value	-	R	D/E	W10116	W10216	…	W13616
				W10117	W10217	…	W13617
Set servo monitoring parameters	-	R	D/E	W10118	W10218	…	W13618
Current motor speed	Unit: 0.1 rpm	R	D/E	W10119	W10219	…	W13619
				W10120	W10220	…	W13620
Servo parameter reading / writing error	Read / write error codes of the servo parameters: 1: failed to read servo parameters. 2: failed to write servo parameters. 3: failed to read servo commands. 4: failed to set servo monitoring commands.	R	D/E	W10121	W10221	…	W13621

5

■ SVON

Definition

When the servo control flag (SVON) is set to On, it means the servo is on; when this flag is set to Off, it means the servo is off. Take Axis 1 as an example, if setting the relay (R10151) to On and this axis executes the Servo On operation, the corresponding monitoring flag R10108 will display "On" indicating that the servo has completed the Servo On action. Set the servo control flag (SVON) to Off, then this axis will execute the Servo Off operation, and the corresponding monitoring flag R10108 will display "Off" at the same time.

Error occurrence

The servo drive will not be able to operate normally after the flag is activated under the following circumstances:

1. Motion bus communication error.
2. Servo alarm is not cleared.

Relevant device

The corresponding PLC command of this function: SVON.

	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	…	Axis 36
SVON control	R / W	R10151	R10251	R10351	R10451	R10551	R10651	R10751	…	R13651
SVON monitoring flag	R	R10108	R10208	R10308	R10408	R10508	R10608	R10708	…	R13608

5

■ Software limit

Definition

The software limit function of the servo axis is to limit the motion range for the mechanism. When the bit for activating software limit is on, the servo axis motion will be limited to the range specified by the positive and negative software limits. Take Axis 1 as an example, if positive limit (W10182, W10183) is set to 100 and negative limit (W10184, W10185) is set to -100, and when the bit for activating software limit is on, the axis will stop and trigger the software limit flag once the servo motion exceeds the range.

Note:

1. The stop command is triggered after the software limit exceeds the range, so the stop position may exceed the setting limit.
2. The software limit function is not supported in speed mode and torque mode.

Relevant device

The corresponding PLC command of this function: SLMT, SLMTON.

	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	…	Axis 36
Activate software limit bit	R/W	R10152	R10251	R10351	R10451	R10551	R10651	R10751	…	R13651
Software limit status bit (positive)	R	R10116	R10208	R10308	R10408	R10508	R10608	R10708	…	R13608
Software limit status bit (negative)	R	R10117	R10217	R10317	R10417	R10517	R10617	R10717	…	R13617
Forward software limit	R/W	W10182	W10282	W10382	W10482	W10582	W10682	W10782	…	W13682
	R/W	W10183	W10283	W10383	W10483	W10583	W10683	W10783	…	W13683
Reverse software limit	R/W	W10184	W10284	W10384	W10484	W10584	W10684	W10784	…	W13684
	R/W	W10185	W10285	W10385	W10485	W10585	W10685	W10785	…	W13685

■ Homing mode

Definition

When homing, the servo motor runs with the first speed (HSP1). Once reaching the reference origin, the motor switches to the second speed (HSP2) to carry on the homing operation.

For description of single-axis homing mode, refer to Appendix B.

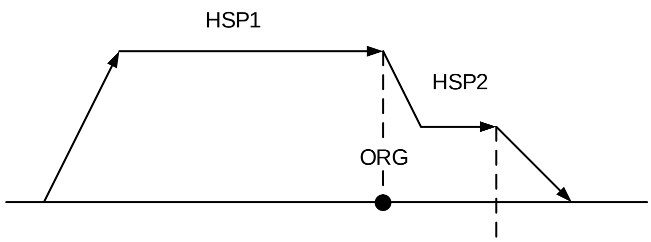


Figure 5.2.2.1 Homing speed switching

Relevant device

The corresponding PLC command of this function: HOME.

	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	…	Axis 36
Single-axis motion control code	R/W	W10151	W10251	W10351	W10451	W10551	W10651	W10751	…	W13651
Homing mode setting	R/W	W10160	W10260	W10360	W10460	W10560	W10660	W10760	…	W13660
First speed in homing mode	R/W	W10162	W10262	W10362	W10462	W10562	W10662	W10762	…	W13662
	R/W	W10163	W10263	W10363	W10463	W10563	W10663	W10763	…	W13663
Second speed in homing mode	R/W	W10164	W10264	W10364	W10464	W10564	W10664	W10764	…	W13664
	R/W	W10165	W10265	W10365	W10465	W10565	W10665	W10765	…	W13665

5

■ S-curve setting

Definition

This is the acceleration / deceleration constant setting for S-curve during motion, which you can specify for each axis.

Relevant device

The corresponding PLC command of this function: SCUR.

	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	…	Axis 36
Motion curve setting	R/W	R10161	R10261	R10361	R10461	R10561	R10661	R10761	…	R13661

■ Acceleration / deceleration time

Definition

This is to set the motor acceleration / deceleration time during motion. The acceleration time is defined as the time from zero speed to the maximum speed set in the Quick Start interface; the deceleration time is the time from the maximum speed set in the Quick Start interface to zero speed.

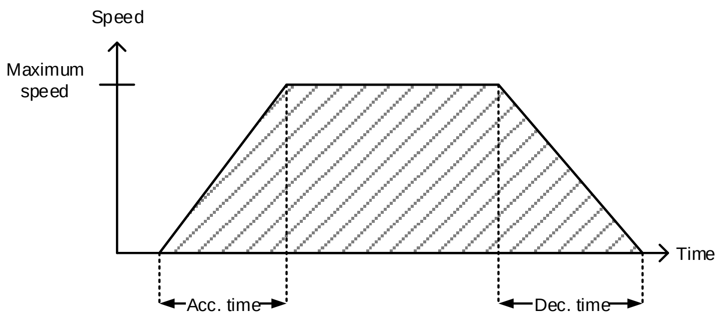


Figure 5.2.2.2 Acceleration / deceleration time setting

Relevant device

The corresponding PLC command of this function: TADC.

	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	…	Axis 36
Acceleration time	R/W	W10152	W10252	W10352	W10452	W10552	W10652	W10752	…	W13652
	R/W	W10153	W10253	W10353	W10453	W10553	W10653	W10753	…	W13653
Deceleration time	R/W	W10154	W10254	W10354	W10454	W10554	W10654	W10754	…	W13654
	R/W	W10155	W10255	W10355	W10455	W10555	W10655	W10755	…	W13655

Logic Editing

6

Before editing the PLC instructions of the IMP, you can find detailed descriptions of the instructions in this chapter.

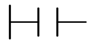
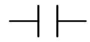
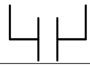
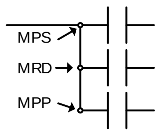
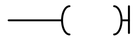
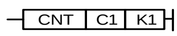
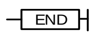
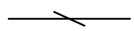
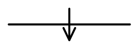
6

6.1.1 PLC instructions

6.1.1 Instruction list

The following are the instructions provided by the IMP controller.

■ List of basic instructions

Basic instruction		
Type	Function name	Symbol
Contact instruction	LD	
	LDI	
	AND	
	ANI	
	OR	
	ORI	
Combined instruction	MPS	
	MRD	
	MPP	
Output instruction	OUT	
	SET	
	RST	
Timer	TMR	
Counter	CNT	
Main program ends	END	
Subprogram ends	SRET	
Invert the operation result	INV	
Rising-edge triggered	NP	
Falling-edge triggered	PN	
No action	NOP	

■ List of application instructions

Application instruction					
Type	NO.	Instruction code		Function	Step No.
		16-bit	32-bit		
Data comparison	001	LD※	DLD※	Contact type compare	5
	002	AND※	DAND※	Contact type compare	5
	003	OR※	DOR※	Contact type compare	5
Data transmission and comparison	004	MOV	DMOV	Move data	5
	005	BMOV	-	Batch move data	11
	006	FMOV	-	Multi move data	11
Rotation	007	ROR	DROR	Rotate right	3
	008	ROL	DROL	Rotate left	3
Flow control	009	CJ	-	Conditional jump	2
	010	CALL	-	Call subroutine	2
	011	FOR	-	Nested loop starts	3
	012	NEXT	-	Nested loop ends	1
Arithmetic operation	013	ADD	DADD	BIN addition	7
	014	SUB	DSUB	BIN subtraction	7
	015	MUL	DMUL	BIN multiplication	7
	016	DIV	DDIV	BIN division	7
	017	INC	DINC	Plus one (BIN)	3
	018	DEC	DDEC	Minus one (BIN)	3
Logical operation	019	WAND	DWAND	AND operation	7
	020	WOR	DWOR	OR operation	7
	021	WXOR	DWXOR	XOR operation	7
Floating operation and conversion	022	-	FADD	Floating point number addition	7
	023	-	FSUB	Floating point number subtraction	7
	024	-	FMUL	Floating point number multiplication	7
	025	-	FDIV	Floating point number division	7
	026	-	FSIN	SIN operation in floating point number format	5
	027	-	FCOS	COS operation in floating point number format	5
	028	-	FTAN	TAN operation in floating point number format	5
	029	-	FASIN	ASIN operation in floating point number format	5
	030	-	FACOS	ACOS operation in floating point number format	5
	031	-	FATAN	ATAN operation in floating point number format	5
Data processing	032	ZRST	-	Zone reset	4
	033	DECO	-	Decoder	11
	034	ENCO	-	Encoder	11
	035	BON	DBON	Monitor bit on	5
	036	ALT	-	ON / OFF alternate	2
	037	AO	-	Analog output	5
	038	AI	-	Analog input	5

■ List of motion instructions

Motion instruction						
Type	NO.	Instruction code		Function	Step No.	
		16-bit	32-bit		16	32
Single-axis motion	050	SVON	-	Servo on	5	-
	051	UINT	-	Speed unit of single axis	5	-
	052	SCUR		Acceleration / deceleration curve setting	5	-
	053	-	TADC	Acceleration / deceleration setting	-	11
	054	-	SLMT	Software limit setting	-	11
	055	SLMTON	-	Software limit activation	5	-
	056	-	COORD	Coordinates setting	-	7
	057	-	HOME	Homing	-	11
	058	ALMR	-	Clear alarm	3	-
	059	ESTP	-	Emergency stop	3	-
	060	SDSTP	-	Decelerate to stop	3	-
	061	-	AXRPM	Read motor’s current speed	-	7
	062	AXTQR	-	Read motor’s current torque	5	-
	063	RSVP	-	Read servo parameter	-	5
	063-1	SVR		Read servo parameter return value		5
	064	WSVP	-	Write servo parameter return value	-	7
	065	SVSTS	-	Read servo DO status	5	-
	066	SVITS	-	Read servo DI status	5	-
	067	RCBL	-	Read the buffer memory usage	5	-
	068	-	RPOS	Read the actual position of the motor axis	-	7
	069	-	LPOS	Read the instruction position of the motion card	-	7
	070	-	TPOS	Read the axial target position	-	7
	071	MOTS	-	Read the status of the motion instruction	5	-
	072	ALE	-	Read the servo drive error code	5	-
	073	-	JOG	Jog	-	11
	074	-	MOVA	Absolute motion	-	11
	075	-	MOVR	Relative motion	-	11
	076	-	MOVPOS	Position change during operation	-	7
	077	-	MOVSPD	Speed change during operation	-	7
	078	-	SPD	Speed control	-	7
	079	-	TRQ	Torque control	-	7
	080	-	RSPD	Read current speed	-	7

Motion instruction						
Type	NO.	Instruction code		Function	Step No.	
		16-bit	32-bit		16	32
Multi-axis motion	100	GSET	-	Group setting	7	-
	101	GUINT	-	Group setting of speed unit	5	-
	102	GSCUR	-	Group setting of acceleration / deceleration curve	5	
	103	-	GTADC	Group setting of acceleration / deceleration time	-	11
	104	ANGLE	-	Arc angle	5	-
	105	DIR	-	Arc direction	5	-
	106	-	PITCH	Helix pitch	-	7
	107	-	DEPTH	Helix depth	-	7
	108	-	CENTER	Arc center	-	11
	109	-	ENDXY	Endpoint of arc	-	11
	110	-	MOVP	Target setting for each axis	-	13
	111	-	MOVLA	Linear motion (absolute)	-	7
	112	-	MOVLR	Linear motion (relative)	-	7
	113	-	CIRCAA	Arc absolute motion (known center coordinates and angle)	-	7
	114	-	CIRCAR	Arc relative motion (known center coordinates and angle)	-	7
	115	-	CIREAA	Arc absolute motion (known endpoint coordinates and angle)	-	7
	116	-	CIREAR	Arc relative motion (known endpoint coordinates and angle)	-	7
	117	-	CIRCEA	Arc absolute motion (known center and endpoint coordinates)	-	7
	118	-	CIRCER	Arc relative motion (known center and endpoint coordinates)	-	7
	119	-	HELIXA	Helical absolute motion	-	7
	120	-	HELIXR	Helical relative motion	-	7
	121	GESTP	-	Group emergency stop	3	-
	122	GSDSTP	-	Group deceleration to stop	3	-
Motion program macro (MPM) instruction	150	MPMST	-	MPM starts	3	-
	151	MPMSTP	-	MPM stops	3	-
	152	MPMPAU	-	MPM pauses	3	-
	153	MPMSPD	-	MPM speed changes	5	-
	154	MPMER	-	Read MPM error code	5	-
	155	MSTEP	-	Read the MPM step No.	7	-

6

6.1.2 Basic instruction

Descriptions

(1) (2) (3)

NO.	D	Instruction	P / F	S1	Function

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	●	●	●	●	●	●	-	-	-	-	-	-	● [V]	-	-	-	-	-

(4)

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	-	-

(1) D: support 32-bit type; (2) P: support rising-edge pulse type / F: support falling-edge pulse type;

(3) Operand; (4) Support indirect register.

■ LD

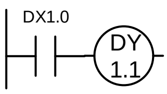
NO. -	-	LD	P / F	S1	Load contact A

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	LD	1 Step

Description: the LD instruction applies to contact A at the beginning of the bus or contact A at the beginning of a contact loop block. Its function is to save the current content and save the S1 contact status in the accumulation register.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
OUT	DY1.1	Output DY1.1 coil

6

■ LDI

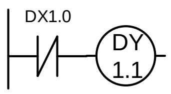
NO.	-	LDI	-	S1	Load contact B
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	LDI	1 Step

Description: the LDI instruction applies to contact B at the beginning of the bus or contact B at the beginning of a contact loop block. Its function is to save the current content and save the S1 contact status in the accumulation register.

Example: ladder diagram



Instruction code		Description
LDI	DX1.0	Load contact B of DX1.0
OUT	DY1.1	Output DY1.1 coil

■ AND

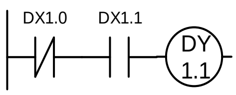
NO.	-	AND	P	S1	Serial connect contact A
-			F		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	AND	1 Step

Description: the AND instruction serial connects contact A. It reads the current status of the specified serial contacts and executes the AND operation on the acquired data together with the results from previous logic operations and saves the result in the accumulation register.

Example: ladder diagram



Instruction code		Description
LDI	DX1.0	Load contact B of DX1.0
AND	DX1.1	Serial connect contact A of DX1.1
OUT	DY1.1	Output DY1.1 coil

■ ANI

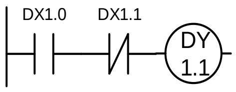
NO.	-	ANI	-	S1	Serial connect contact B
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ANI	1 Step

Description: the ANI instruction serial connects contact B. It reads the current status of the specified serial contacts and executes the AND operation on the acquired data together with the results from previous logic operations and saves the result in the accumulation register.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
ANI	DX1.1	Serial connect contact B of DX1.1
OUT	DY1.1	Output DY1.1 coil

■ OR

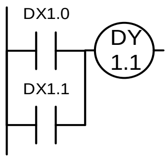
NO.	-	OR	P	S1	Parallel connect contact A
-			F		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	OR	1 Step

Description: the OR instruction parallel connects contact A. It reads the current status of the specified serial contacts and executes the OR operation on the acquired data together with the results from previous logic operations and saves the result in the accumulation register.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
OR	DX1.1	Parallel connect contact A of DX1.1
OUT	DY1.1	Output DY1.1 coil

6

■ ORI

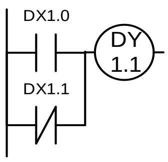
NO.	-	ORI	-	S1	Parallel connect contact B
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ORI	1 Step

Description: the ORI instruction parallel connects contact B. It reads the current status of the specified serial contacts and executes the OR operation on the acquired data together with the results from previous logic operations and saves the result in the accumulation register.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
ORI	DX1.1	Parallel connect contact B of DX1.1
OUT	DY1.1	Output DY1.1 coil

■ ANB

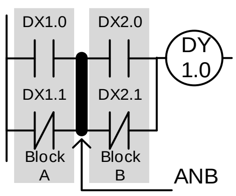
NO.	-	ANB	-	No operand	Serial connect loop block
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ANB	1 Step

Description: the ANB instruction executes the AND operation on previously saved logic operation result and current value in the accumulation register.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
ORI	DX1.1	Parallel connect contact B of DX1.1
LD	DX2.0	Load contact A of DX2.0
ORI	DX2.1	Parallel connect contact B of DX2.1
ANB		Serial connect loop blocks
OUT	DY1.0	Output DY1.0 coil

6

■ ORB

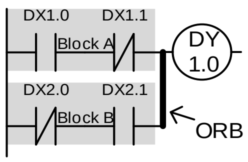
NO.	-	ORB	-	No operand	Parallel connect loop block
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ORB	1 Step

Description: the ORB instruction executes the OR operation on previously saved logic operation result and current value in the accumulation register.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
ANI	DX1.1	Serial connect contact B of DX1.1
LD	DX2.0	Load contact B of DX2.0
AND	DX2.1	Serial connect contact A of DX2.1
ORB		Parallel connect loop blocks
OUT	DY1.0	Output DY1.0 coil

■ MPS

NO.	-	MPS	-	No operand	Save in stack
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	MPS	1 Step

Description: MPS saves the current value in the accumulation register to the stack register. (Stack index increases by 1.)

■ MRD

NO.	-	MRD	-	No operand						Read stack (stack index remains)							
-			-														
	Bit device						Word device									External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	MRD	1 Step

Description: MRD retrieves the last saved logic operation result in the stack and saves it in the accumulation register. (Stack index remains unchanged.)

■ MPP

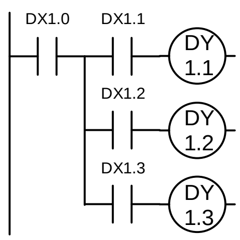
NO.	-	MPP	-	No operand						Read stack							
-			-														
	Bit device						Word device									External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	MPP	1 Step

Description: MPP retrieves the last saved logic operation result in the stack and saves it in the accumulation register. (Stack index decreases by 1.)

Note: MPS and MPP must correspond to each other, or program error will occur.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
MPS		Save in stack
AND	DX1.1	Serial connect contact A of DX1.1
OUT	DY1.1	Output DY1.1 coil
MRD		Read stack (stack index remains)
AND	DX1.2	Serial connect contact A of DX1.2
OUT	DY1.2	Output DY1.2 coil
MPP		Read stack
AND	DX1.3	Serial connect contact A of DX1.3
OUT	DY1.3	Output DY1.3 coil

■ OUT

NO.	-	OUT	-	D	Output
-			-		

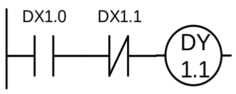
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
D	-	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	OUT	1 Step

Description: output the logic operation result to the specified bit regardless of the operation result is TRUE of FALSE.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
ANI	DX1.1	Serial connect contact B of DX1.1
OUT	DY1.1	Output DY1.1 coil

■ SET

NO.	-	SET	-	D	Action remains (ON)
-			-		

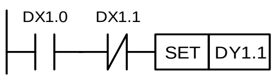
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
D	-	●	●	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	SET	1 Step

Description: when the SET instruction is executed, the specified bit is set to On and will remain on. You can use the RST instruction to set this bit to off. If the SET instruction is not executed, the status of the specified bit remains the same.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
ANI	DX1.1	Serial connect contact B of DX1.1
SET	DY1.1	DY1.1 setting (On)

■ RST

NO.	-	RST	-	D	Action clears (OFF)
-			-		

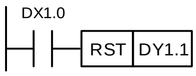
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
D	-	●	●	●	●	●	-	-	-	-	●	●	-	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	RST	1 Step

Description: when the RST instruction is executed, if the specified bit is a bit device, then the bit is cleared to FALSE; if it is a word device, the bit is cleared to 0. If the RST instruction is not executed, the status of the specified bit remains the same.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
RST	DY1.1	DY1.1 clear (Off)

■ TMR

NO.	-	TMR	-	S1, S2	Timer
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

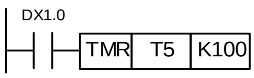
	Instruction	Step No.
32-bit	-	-
16-bit	TMR	2 Step

Description: when the TMR instruction is executed, the specified timer coil is powered and the timer starts counting. When the timer reaches the set timing (time value ≧ set value), the contact action is as follows:

NO (Normally Open) contact: close.

NC (Normally Close) contact: open.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
TMR	T5 K100	Timer T5 is set to K100

■ CNT

NO.	D	CNT	-	S1, S2	Counter
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
If the S1 operand specifies C200 - 255 as the counter, then the 32-bit DCNT instruction should be used.	32-bit	DCNT	3 Step
	16-bit	CNT	2 Step

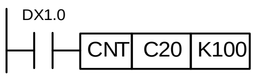
Description: When the CNT instruction changes from Off to On, the coil of the counter assigned by it switches from Off to On, leading to its counting value increasing by 1. When the counter reaches the set count (time value ≧ set value), the contact action is as follows:

NO (Normally Open) contact: close.

NC (Normally Close) contact: open.

When the count setting of S2 is reached, the counter's contacts and counting value remain the same even when more counting pulse inputs are received. You can use the RST instruction to restart counting or clear the value.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
CNT	C20 K100	Counter C20 is set to K100

■ END

NO.	-	END	-	No operand	Main program ends
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	END	1 Step

Description: the main program's last instruction must be the END instruction. PLC scans from address 0 to the END instruction, then returns to address 0 to scan again. When compiling with PLC Ladder Edit, if the PLC main program does not include the END instruction, the compiler will add it to the end of the PLC main program.

■ SRET

NO.	-	SRET	-	No operand	Subprogram ends
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	SRET	1 Step

Description: the PLC subprogram's last instruction must be the SRET instruction. PLC scans from address 0 to the SRET instruction, then ends the subprogram and returns to the main program.

■ INV

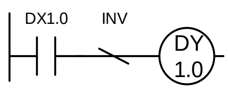
NO.	-	INV	-	No operand	Invert the operation result
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	INV	1 Step

Description: invert the logic operation result before the INV instruction and save it in the accumulation register.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
INV		Invert the operation result
OUT	DY1.0	Output DY1.0 coil

■ NP

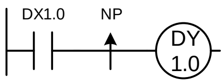
NO.	-	NP	-	No operand	Rising-edge triggered
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	NP	1 Step

Description: acquire the rising-edge status from the logic operation result before the NP instruction, then save it in the accumulation register.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
NP		Rising-edge of operation result
OUT	DY1.0	Output DY1.0 coil

■ PN

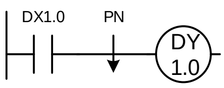
NO.	-	PN	-	No operand	Falling-edge triggered
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	PN	1 Step

Description: acquire the falling-edge status from the logic operation result before the PN instruction, then save it in the accumulation register.

Example: ladder diagram



Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
PN		Falling-edge of operation result
OUT	DY1.0	Output DY1.0 coil

6

■ NOP

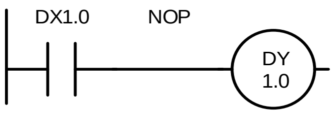
NO.	-	NOP	-	No operand	No action
-			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	NOP	1 Step

Description: the NOP instruction does not compute in the program, so the original logic operation result remains after execution. This is used when you want to delete an instruction without changing the program length, then you can replace it with the NOP instruction.

Example: ladder diagram



#NOP is not displayed in the ladder diagram

Instruction code		Description
LD	DX1.0	Load contact A of DX1.0
NOP		No action
OUT	DY1.0	Output DY1.0 coil

6

6.1.3 Application instruction

■ LD※

NO.	D	LD※	-	S1, S2	Contact type compare LD※
001			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-

Notes when applying operand: ※ can be =, >, <, <>, ≦, and ≧.

	Instruction	Step No.
32-bit	DLD※	5 Step
16-bit	LD※	5 Step

Description: S1: data source device 1; S2: data source device 2.

This instruction compares the values stored in S1 and S2. When the comparison result satisfies the condition, the contact turns on, otherwise it does not turn on.

The LD※ instruction can be directly connected to the bus.

16-bit instruction	32-bit instruction	Turn-on condition	Non-turn-on condition
LD =	DLD =	S1 = S2	S1 ≠ S2
LD >	DLD >	S1 > S2	S1 ≦ S2
LD <	DLD <	S1 < S2	S1 ≧ S2
LD < >	DLD < >	S1 ≠ S2	S1 = S2
LD < =	DLD < =	S1 ≦ S2	S1 > S2
LD > =	DLD > =	S1 ≧ S2	S1 < S2

When comparing 32-bit counters (C200 - C255) with this instruction, use the 32-bit instruction (DLD※).

■ AND※

NO.	D	AND※	-	S1, S2	Contact type compare AND※
002			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	DAND※	5 Step
	16-bit	AND※	5 Step

Description: S1: data source device 1; S2: data source device 2.

This instruction compares the values stored in S1 and S2. When the comparison result satisfies the condition, the contact turns on, otherwise it does not turn on.

AND※ is a comparison instruction that serial connects to a contact.

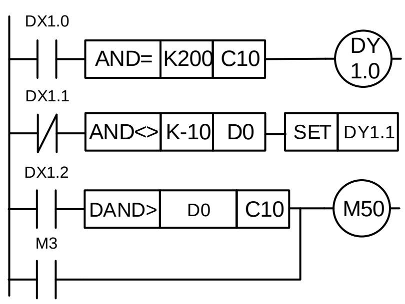
16-bit instruction	32-bit instruction	Turn-on condition	Non-turn-on condition
AND =	DAND =	S1 = S2	S1 ≠ S2
AND >	DAND >	S1 > S2	S1 ≦ S2
AND <	DAND <	S1 < S2	S1 ≧ S2
AND < >	DAND < >	S1 ≠ S2	S1 = S2
AND < =	DAND < =	S1 ≦ S2	S1 > S2
AND > =	DAND > =	S1 ≧ S2	S1 < S2

When comparing 32-bit counters (C200 - C255) with this instruction, use the 32-bit instruction (DAND※).

Example: when DX1.0 = On and the current value of C10 equals K200, then DY1.0 = On.

When DX1.1 = Off and the value of register D0 does not equal K-10, then DY1.1 = On and the status remains.

When DX1.2 = On and the value of the 32-bit register D0 (D1) is greater than C10 or M3 = On, then M50 = On.



■ OR※

NO.	D	OR※	-	S1, S2	Contact type compare OR※
003			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	DOR※	5 Step
	16-bit	OR※	5 Step

Description: S1: data source device 1; S2: data source device 2.

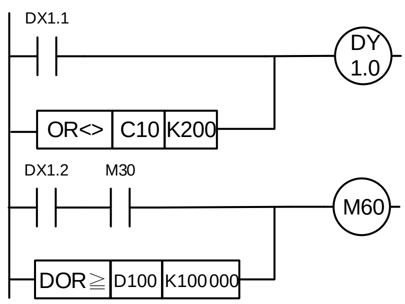
This instruction compares the values stored in S1 and S2. When the comparison result satisfies the condition, the contact turns on, otherwise it does not turn on.

OR※ is a comparison instruction that serial connects to a contact.

16-bit instruction	32-bit instruction	Turn-on condition	Non-turn-on condition
OR =	DOR =	S1 = S2	S1 ≠ S2
OR >	DOR >	S1 > S2	S1 ≦ S2
OR <	DOR <	S1 < S2	S1 ≧ S2
OR < >	DOR < >	S1 ≠ S2	S1 = S2
OR < =	DOR < =	S1 ≦ S2	S1 > S2
OR > =	DOR > =	S1 ≧ S2	S1 < S2

When comparing 32-bit counters (C200 - C255) with this instruction, use the 32-bit instruction (DOR※).

Example: when DX1.1 = On or the current value of C10 does not equal K200, then DY1.0 = On. When DX1.2 and M30 are both On or the data in the 32-bit register D100 (D101) is greater than or equals K100,000, then M60 = On.



6

■ MOV

NO.	D	MOV	-	S1, D	Move data
004			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	●	●	●	●	●	●	● [V]	●	-	● [V]	-	-
D	-	-	-	-	-	-	-	●	●	-	●	●	● [V]	●	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	DMOV	5 Step
	16-bit	MOV	5 Step

Description: S1: data source device 1; D: data transfer destination.

When executing this instruction, the data in S1 is moved to D. If you do not execute this instruction, the data in D remains the same.

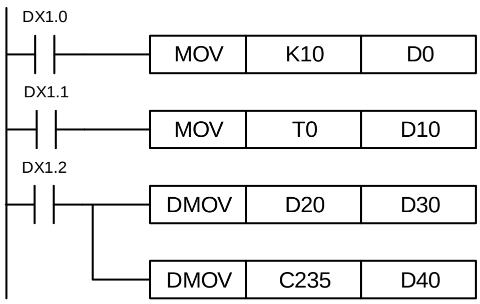
When applying 32-bit instruction (e.g. application instruction MUL), you need to use the DMOV instruction to move the operation result and the current value of the 32-bit high-speed counter.

Example: you need to use the MOV instruction to move the 16-bit data.

When DX1.0 = Off, the content of D10 remains unchanged; if DX1.0 = On, the value of K10 is sent to register D0.

When DX1.1 = Off, the content of D10 remains unchanged; if DX1.1 = On, the current value of T0 is sent to register D10.

When DX1.2 = Off, the contents of (D31, D30) and (D41, D40) remain unchanged; if DX1.2 = On, the current value of (D21, D20) is sent to register (D31, D30) and the value of C235 is sent to register (D41, D40).



6

■ BMOV

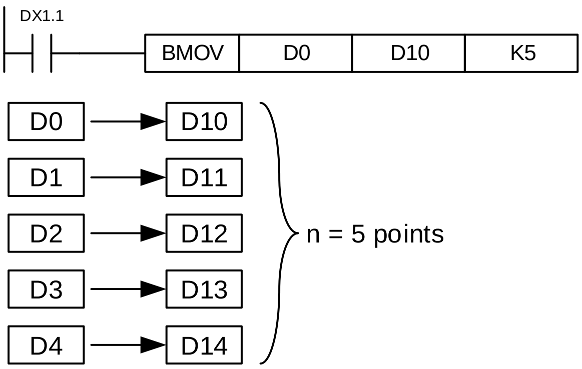
NO.	-	BMOV	-	S1, D, n	Batch move data
005			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-		-	● [V]	-	-	● [V]	-	-
n	-	-	-	-	-	-	-	-	-	●	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	BMOV	11 Step

Description: S1: start of source device; D: start of target device; n: length of transmission block. Contents from the start position (specified by S) of the device to the n^{th} register are transmitted to the device start number (specified by D) to the n^{th} register.

Example: when DX1.1 = On, the contents of D0 - D4 are transferred to the 5 consecutive registers starting from D10.



■ FMOV

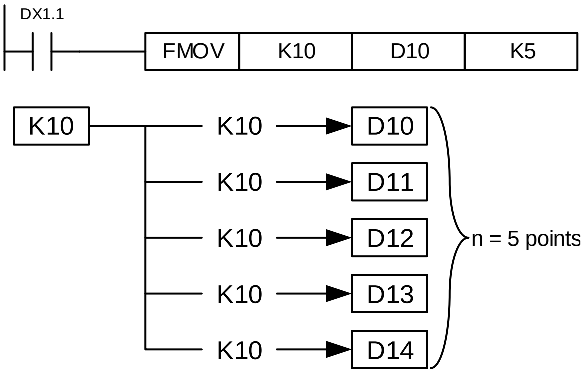
NO.	-	FMOV	-	S1, D, n	Multi move data
006			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	● [V]	-	-	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-		-	● [V]	-	-	● [V]	-	-
n	-	-	-	-	-	-	-	-	-	●	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	FMOV	11 Step

Description: S1: start of source device; D: start of target device; n: length of transmission block. Content of S is transmitted to the n^{th} register starting from the device start number specified D. If the number of points specified by n exceeds the used range of the device, only the valid range is transmitted.

Example: when DX1.1 = On, the content of K10 is transferred to the 5 consecutive registers starting from D10.



6

■ ROR

NO.	D	ROR	-	D, n	Rotate right
007			-		

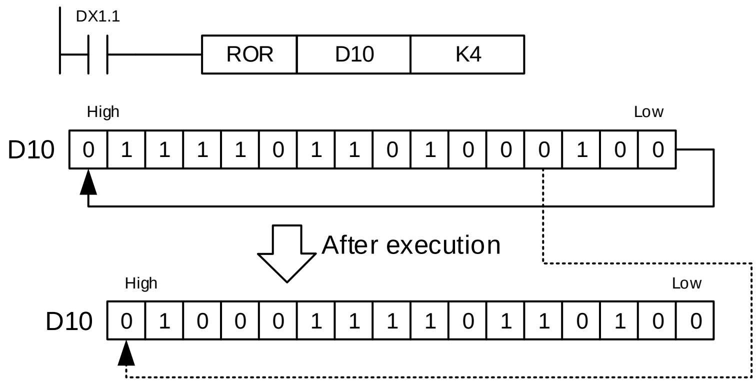
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-
n	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
Range: n = K1 - K16 (16-bit); n = K1 - K32 (32-bit)	32-bit	DROR	3 Step
	16-bit	ROR	3 Step

Description: D: device for rotation; n: number of bits in one rotation.

Rotate the device content specified by D one time to the right by n bits.

Example: when DX1.1 changes from off to on, the 16 bits in D10 right rotates in unit of 4 bits as shown in the figure below.



■ ROL

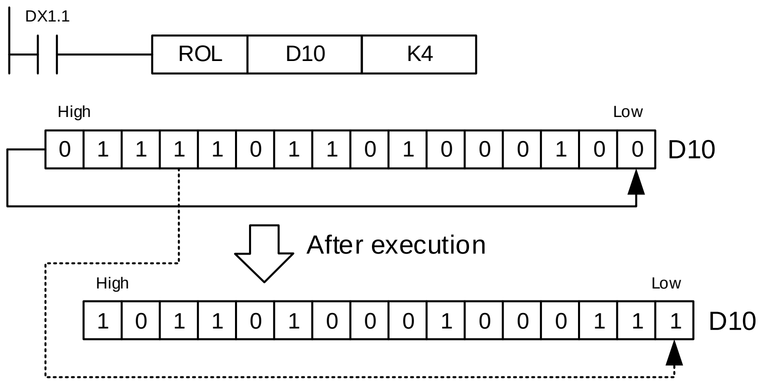
NO.	D	ROL	-	D, n							Rotate left						
008			-														
	Bit device						Word device									External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-
n	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-

Notes when applying operand: Range: n = K1 - K16 (16-bit); n = K1 - K32 (32-bit)		Instruction	Step No.
	32-bit	DROL	3 Step
	16-bit	ROL	3 Step

Description: D: device for rotation; n: number of bits in one rotation.

Rotate the device content specified by D one time to the left by n bits.

Example: when DX1.1 changes from off to on, the 16 bits in D10 left rotates in unit of 4 bits as shown in the figure below.



6

■ CJ

NO.	-	CJ	-	S	Jump
009			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand: the S operand can specify P0 - P255.		Instruction	Step No.
	32-bit	-	-
	16-bit	CJ	2 Step

Description: S: target of conditional jump.

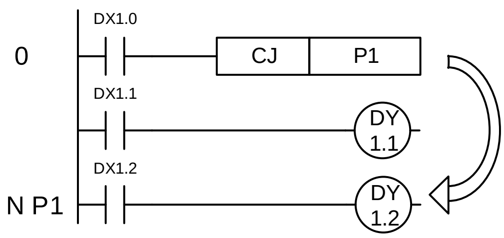
When you do not want to execute a section of the PLC program, you can apply the CJ instruction to shorten the PLC running time or use the dual output of the PLC program. If the program location specified by pointer P appears before the CJ instruction, the PLC may not be able to complete the program scan. However, the CJ instruction can repeatedly specify the same pointer P.

Device actions when executing the jump instruction:

1. States of devices Y, M and S remain the same as before executing the jump instruction.
2. When executing timing, the 10-ms and 100-ms timers will pause.
3. Timers T192 - T199 for executing the subprogram keeps on timing and the output contacts operate normally.
4. Counter stops counting.
5. If the clear instruction of the timer is executed before jumping, then the device is still in the clear status during the jump execution.
6. The application instruction will not be executed.

Example: when DX1.0 = On, the program automatically jumps from Address 0 to Address N (the specified label P1) and continues its execution by skipping all the addresses in between.

When DX1.0 = Off, the program executes starting from Address 0 and continues in sequence, and the CJ instruction is not executed.



6

■ CALL

NO.	-	CALL	-	S									Call subroutine					
010			-															
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand: the S operand is used with the Ladder Editor subroutine name.		Instruction	Step No.
	32-bit	-	-
	16-bit	CALL	2 Step

Description: S: name of the subroutine to be called. The subroutine should be created before being called.

The CALL instruction can be used to call the same subroutine for unlimited times.

The subroutine can also apply this instruction to call other subroutines for up to eight layers including the original subroutine.

■ FOR

NO.	-	FOR	-	S									Nested loop starts					
010			-															
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	FOR	3 Step

Description: S: the number of times the loop is to be executed.

■ NEXT

NO.	-	NEXT	-	-	Nested loop ends
012			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand: no operand is required.		Instruction	Step No.
	32-bit	-	-
	16-bit	NEXT	1 Step

Description: the FOR instruction specifies the FOR to NEXT loop to execute for N times.

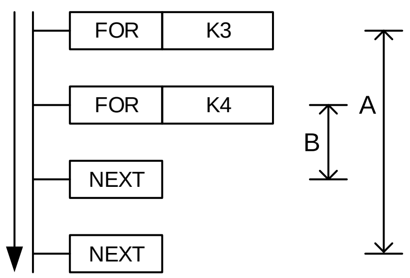
After exiting the FOR to NEXT loop, the program continues running.

The valid range is N = K1 to K32,767. If the specified number of times is N ≦ K1, the specified number of times is regarded as K1.

In the following conditions, error may occur:

1. The NEXT instruction precedes the FOR instruction.
2. The FOR instruction is executed without the NEXT instruction.
3. The END and SRET instructions are followed by the NEXT instruction.
4. The number of FOR and NEXT instructions is different.
5. The FOR to NEXT loop can nest for up to 5 layers. If the nesting layers exceed the limit, grammar error may occur.
6. The CJ instruction cannot be used between FOR and NEXT

Example: after executing the A program for 3 times, continue to execute the NEXT instruction. For each execution of the A program, the B program will execute four times, so the B program runs for 12 times (3 × 4 = 12) in total.



6

■ ADD

NO.	D	ADD	-	S1, S2, D	BIN addition
013			-		

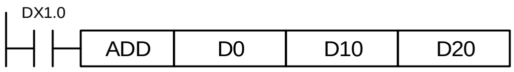
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	●	-	● [V]	-	-
S2							-	-	-	●	●	●	● [V]	●	-	● [V]		-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	●*1	-	-	-	-

Notes when applying operand: *1: only supports 16-bit instructions.		Instruction	Step No.
	32-bit	DADD	7 Step
	16-bit	ADD	7 Step

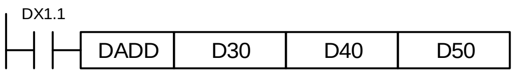
Description: S1: summand; S2: addend; D: sum.

Add the values in data sources S1 and S2 in BIN format and save the result in D. The first bit of each data indicates it is positive (0) or negative (1). This enables algebraic addition operations, such as 3 + (-9) = -6.

Example 1: 16-bit BIN addition: when DX1.0 = On, add summand D0 and addend D10 and save the result in D20.



Example 2: 32-bit BIN addition: when DX1.1 = On, add summand (D31, D30) and addend (D41, D40) and save the result in (D51, D50). (D30, D40, and D50 are the lower 16-bit data; D31, D41, and D51 are the upper 16-bit data.)



6

■ SUB

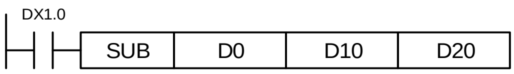
NO.	D	SUB	-	S1, S2, D	BIN subtraction
014			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	●	-	● [V]	-	-
							-	-	-	●	●	●	● [V]	●	-	● [V]		-
	-	-	-	-	-	-	-	-	-	-	-	-	●	●*1	-	-	-	-

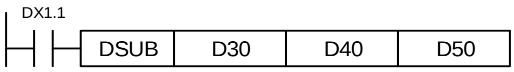
Notes when applying operand: *1: only supports 16-bit instructions.		Instruction	Step No.
	32-bit	DSUB	7 Step
	16-bit	SUB	7 Step

Description: S1: minuend; S2: subtrahend; D: difference.
Subtract the value in data source S2 from the value in data source S1 in BIN format and save the result in D. The first bit of each data indicates it is positive (0) or negative (1). This enables algebraic subtraction operations, such as 3 + (-9) = -6.

Example 1: 16-bit BIN subtraction: when DX1.0 = On, subtract the value of D10 from D0 and save the result in D20.



Example 2: 32-bit BIN subtraction: when DX1.1 = On, subtract the value of (D41, D40) from (D31, D30) and save the result in (D51, D50). (D30, D40, and D50 are the lower 16-bit data; D31, D41, and D51 are the upper 16-bit data.)



6

■ MUL

NO. 015	D	MUL	- / -	S1, S2, D	BIN multiplication

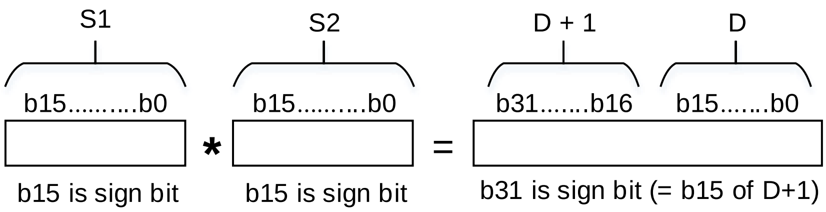
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	●	-	● [V]	-	-
S2							-	-	-	●	●	●	● [V]	●	-	● [V]		-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	●*1	-	-	-	-

Notes when applying operand:		Instruction	Step No.
16-bit instruction D operand occupies 2 consecutive points. 32-bit instruction D operand occupies 4 consecutive points. *1: only supports 16-bit instructions.	32-bit	DMUL	7 Step
	16-bit	MUL	7 Step

Description: S1: multiplicand; S2: multiplier; D: product.

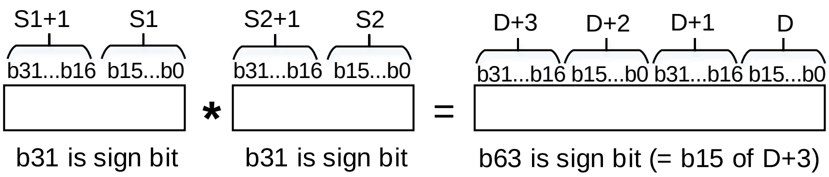
Multiply values in data source S1 by S2 in signed binary format and save the result in D. When applying 16-bit and 32-bit operations, please note the difference in the position of the sign bit for S1, S2, and D.

16-bit BIN multiplication operation:

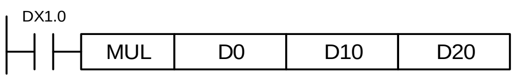


Sign bit = 0 indicates positive number; sign bit = 1 indicates negative number.

32-bit BIN multiplication operation:



Example: multiply 16-bit D0 by 16-bit D10 and the result is the product of 32-bit. The upper 16-bit is saved in D21, the lower 16-bit is saved in D20, and on / off of the leftmost bit indicates the positive and negative result.



6

■ DIV

NO.	D	DIV	-	S1, S2, D	BIN division
016			-		

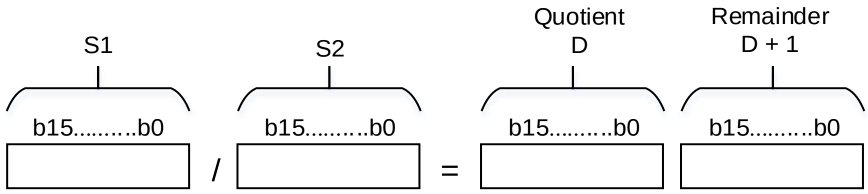
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	●	-	● [V]	-	-
S2							-	-	-	●	●	●	● [V]	●	-	● [V]		-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	●*1	-	-	-	-

Notes when applying operand:		Instruction	Step No.
16-bit instruction D operand occupies 2 consecutive points. 32-bit instruction D operand occupies 4 consecutive points. *1: only supports 16-bit instructions.	32-bit	DDIV	7 Step
	16-bit	DIV	7 Step

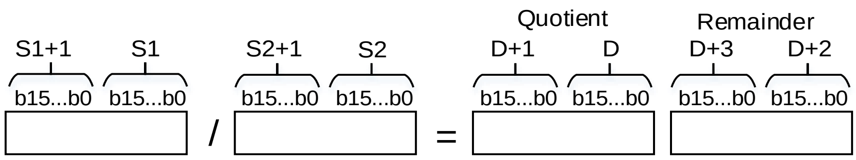
Description: S1: dividend; S2: divisor; D: quotient and remainder.

Value in data source S1 is divided by S2 in signed binary format, and the quotient and remainder will be saved in D. When applying 16-bit and 32-bit operations, please note the difference in the position of the sign bit for S1, S2, and D.

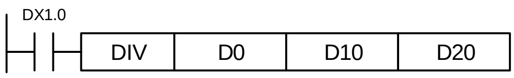
When the divisor is 0, the instruction is not executed:



32-bit BIN division operation:



Example: 16-bit D0 is divided by 16-bit D10 and the result is the quotient of 32-bit. The upper 16-bit is saved in D21, the lower 16-bit is saved in D20, and on / off of the leftmost bit indicates the positive and negative result.



■ INC

NO.	D	INC	-	D	Plus one (BIN)
017			-		

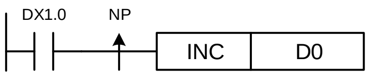
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
D	-	-	-	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-

Notes when applying operand: 32-bit instruction D operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	DINC	3 Step
	16-bit	INC	3 Step

Description: D: target device.

When executing this instruction, the value in the specified device D increments by 1 for each program scan cycle. In the case of 16-bit operations, 32,767 plus 1 becomes -32,768; for 32-bit operations, 2,147,483,647 plus 1 becomes -2,147,483,648.

Example: when DX1.0 is off then on, the value in D0 automatically increments by 1.



■ DEC

NO.	D	DEC	-	D	Minus one (BIN)
018			-		

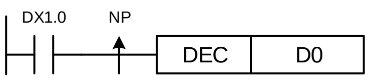
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
D	-	-	-	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-

Notes when applying operand: 32-bit instruction D operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	DDEC	3 Step
	16-bit	DEC	3 Step

Description: D: target device.

When executing this instruction, the value in the specified device D decrements by 1 for each program scan cycle. In the case of 16-bit operations, -32,768 minus 1 becomes 32,767; for 32-bit operations, -2,147,483,648 minus 1 becomes 2,147,483,647.

Example: when DX1.0 is off then on, the value in D0 automatically decrements by 1.



6

■ WAND

NO.	D	WAND	-	S1, S2, D	AND operation
019			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

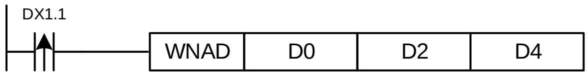
Notes when applying operand:		Instruction	Step No.
	32-bit	DWAND	7 Step
	16-bit	WAND	7 Step

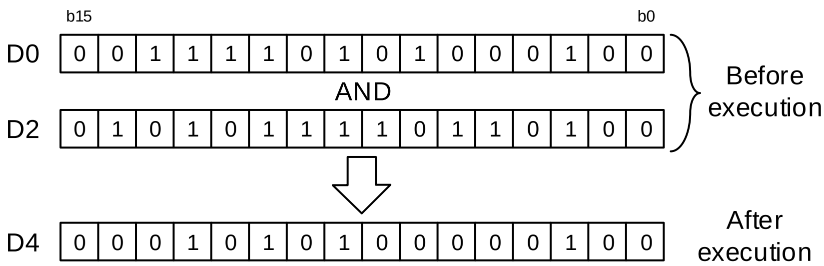
Description: S1: data source device 1; S2: data source device 2; D: operation result.

Execute AND operation on data sources S1 and S2 and save the result in D.

The AND operation rule is when any value is 0, the result is 0.

Example: when DX1.1 = On, execute AND operation on 16-bit D0 and D2, and save the result in D4.





6

■ WOR

NO.	D	WOR	-	S1, S2, D	OR operation
020			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

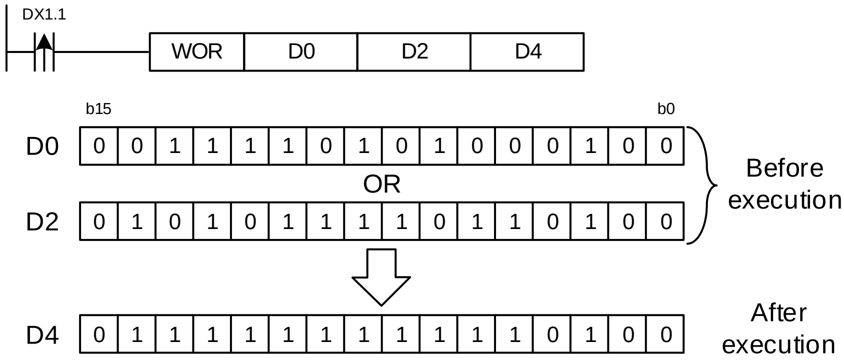
Notes when applying operand:		Instruction	Step No.
	32-bit	DWOR	7 Step
	16-bit	WOR	7 Step

Description: S1: data source device 1; S2: data source device 2; D: operation result.

Execute OR operation on data sources S1 and S2 and save the result in D.

The OR operation rule is when any value is 1, the result is 1.

Example: when DX1.1 = On, execute OR operation on 16-bit D0 and D2, and save the result in D4.



6

■ WXOR

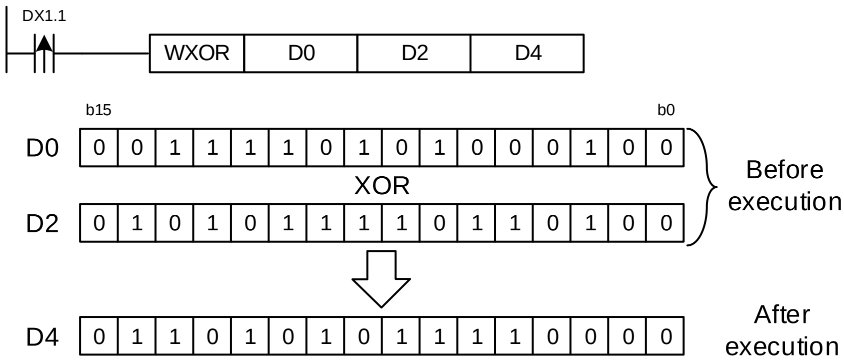
NO.	D	WXOR	-	S1, S2, D	XOR operation
021			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	DWXOR	7 Step
	16-bit	WXOR	7 Step

Description: S1: data source device 1; S2: data source device 2; D: operation result.
Execute XOR operation on data sources S1 and S2 and save the result in D.
The XOR operation rules are if both values are the same, the result is 0; if not, the result is 1.

Example: when DX1.1 = On, execute XOR operation on 16-bit D0 and D2, and save the result in D4.



■ FADD

NO.	-	FADD	-	S1, S2, D	Floating point number addition
022			-		

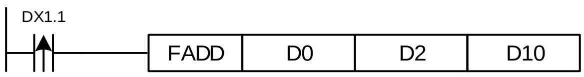
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	FADD	7 Step
	16-bit	-	-

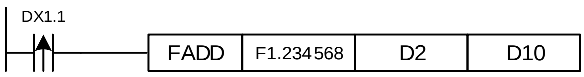
Description: S1: summand; S2: addend; D: sum.

Add the value in the register specified by S2 to the value in the register specified by S1, and save the sum in the register specified by D. All operations are executed in the floating point number format.

Example 1: when DX1.1 = On, add floating point numbers (D3, D2) to floating point numbers (D1, D0), and save the result in (D11, D10).



Example 2: when DX1.1 = On, add floating point numbers (D3, D2) to F1.234568, and save the result in (D11, D10).



6

■ FSUB

NO.	-	FSUB	-	S1, S2, D	Floating point number subtraction
023			-		

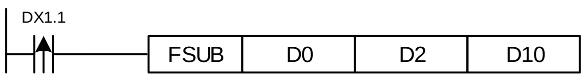
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	FSUB	7 Step
	16-bit	-	-

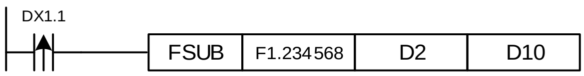
Description: S1: minuend; S2: subtrahend; D: difference.

Subtract the value in the register specified by S2 from the value in the register specified by S1, and save the difference in the register specified by D. All operations are executed in the floating point number format.

Example 1: when DX1.1 = On, subtract floating point numbers (D3, D2) from floating point numbers (D1, D0), and save the result in (D11, D10).



Example 2: when DX1.1 = On, subtract floating point numbers (D3, D2) from F1.234568 and save the result in (D11, D10).



■ FMUL

NO.	-	FMUL	-	S1, S2, D	Floating point number multiplication
024			-		

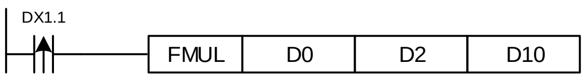
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	FMUL	7 Step
	16-bit	-	-

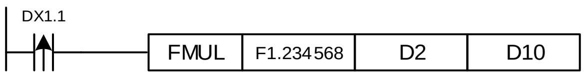
Description: S1: multiplicand; S2: multiplier; D: product.

Multiply the value in the register specified by S1 by the value in the register specified by S2, and save the product in the register specified by D. All operations are executed in the floating point number format.

Example 1: when DX1.1 = On, multiply floating point numbers (D1, D0) by floating point numbers (D3, D2), and save the result in (D11, D10).



Example 2: when DX1.1 = On, multiply F1.234568 by floating point numbers (D3, D2) and save the result in (D11, D10).



■ FDIV

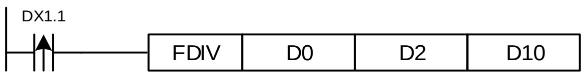
NO. 025	D	FDIV	- / -	S1, S2, D	Floating point number division													
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	FDIV	7 Step
	16-bit	-	-

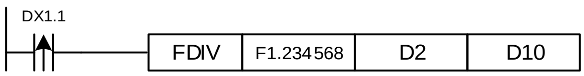
Description: S1: dividend; S2: divisor; D: quotient.

Divide the value in the register specified by S1 by the value in the register specified by S2, and save the quotient in the register specified by D. All operations are executed in the floating point number format. If the value of the divisor S2 is 0, the instruction is not executed due to arithmetic operation error.

Example 1: when DX1.1 = On, divide floating point numbers (D1, D0) by floating point numbers (D3, D2), and save the result in (D11, D10).



Example 2: when DX1.1 = On, divide F1.234568 by floating point numbers (D3, D2) and save the result in (D11, D10).



■ FSIN

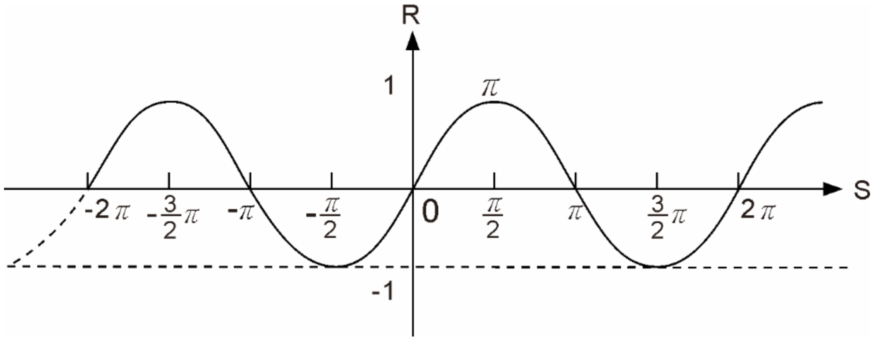
NO. 026	-	FSIN	- / -	S1, D	SIN operation in floating point number format

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
S operand occupies 2 consecutive points and F device is available.	32-bit	FSIN	5 Step
D operand occupies 2 consecutive points.	16-bit	-	-

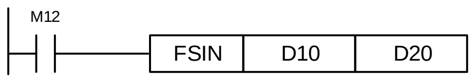
Description: S1: specified source value (floating point number); D: result acquired from SIN value (floating point number).

Obtain SIN value from the source radian specified by S and save the value in the register specified by D. The figure below shows the relation between the radian and result:



S: radian data; R: result (SIN value)

Example: when M12 = On, obtain SIN value from the radian of (D11, D10) and save it in (D21, D20), which is in floating point number format.



6

■ FCOS

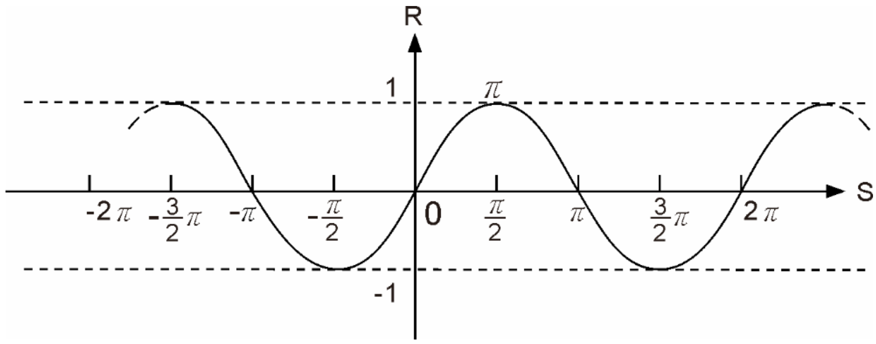
NO.	-	FCOS	-	S1, D	COS operation in floating point number format
027			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
S operand occupies 2 consecutive points. D operand occupies 2 consecutive points.	32-bit	FCOS	5 Step
	16-bit	-	-

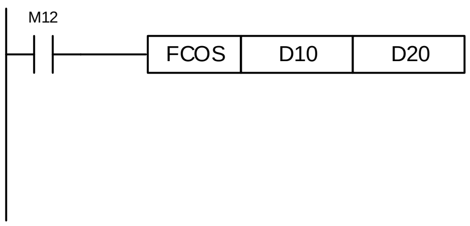
Description: S1: specified source value (floating point number); D: result acquired from COS value (floating point number).

Obtain COS value from the source radian specified by S and save the value in the register specified by D. The figure below shows the relation between the radian and result:



S: radian data; R: result (COS value)

Example: when M12 = On, obtain COS value from the radian of (D11, D10) and save it in (D21, D20), which is in floating point number format.



6

■ FTAN

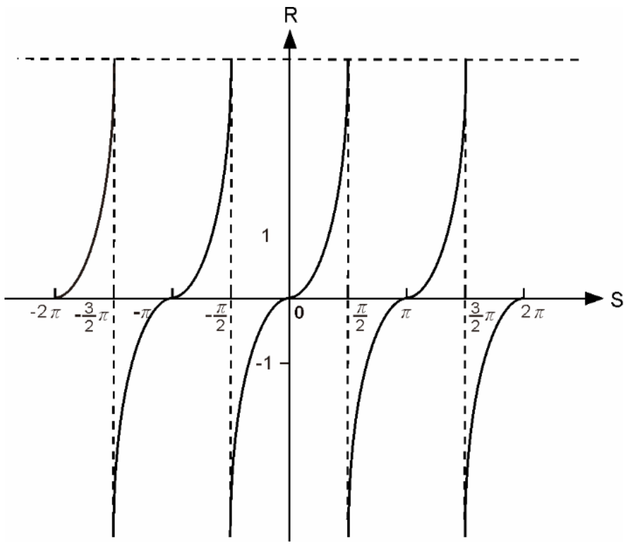
NO.	-	FTAN	-	S1, D	TAN operation in floating point number format
028			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand: S operand occupies 2 consecutive points. D operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	FTAN	5 Step
	16-bit	-	-

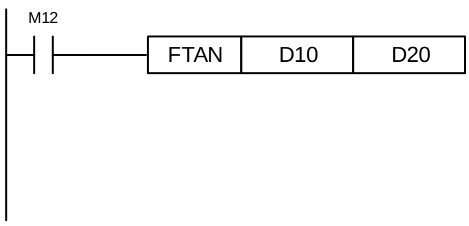
Description: S1: specified source value (floating point number); D: result acquired from TAN value (floating point number).

Obtain TAN value from the source radian specified by S and save the value in the register specified by D. The figure below shows the relation between the radian and result:



S: radian data; R: result (TAN value)

Example: when M12 = On, obtain TAN value from the radian of (D11, D10) and save it in (D21, D20), which is in floating point number format.



■ FASIN

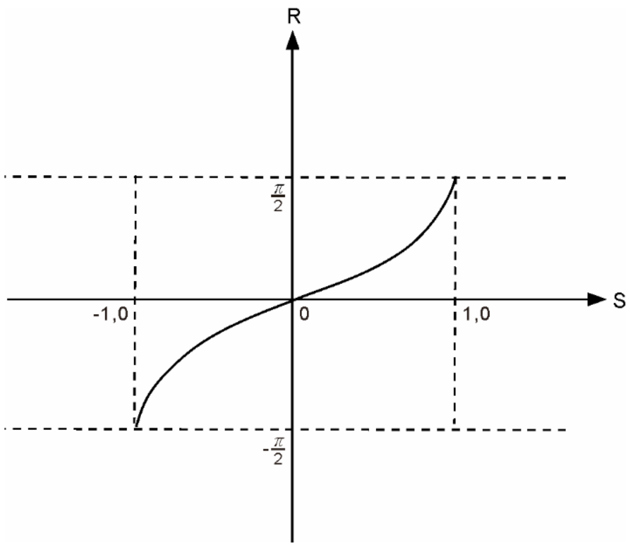
NO.	-	FASIN	-	S1, D	ASIN operation in floating point number format
029			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
S operand occupies 2 consecutive points.	32-bit	FASIN	5 Step
D operand occupies 2 consecutive points.	16-bit	-	-

Description: S: source of specified sine value (floating point number); D: radian result acquired from ASIN value (floating point number).

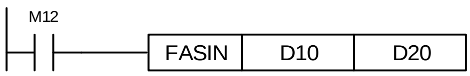
ASIN value = $\sin^{-1}$. The figure below shows the relation between the input data and result:



S: input data (sine); R: result of ASIN value (radian)

The sine value specified by S operand must be between -1.0 and +1.0. If the value is not within this range, this instruction is not executed.

Example: when M12 = On, obtain ASIN value from (D11, D10) and save it in (D21, D20), which is in floating point number format.



6

■ FACOS

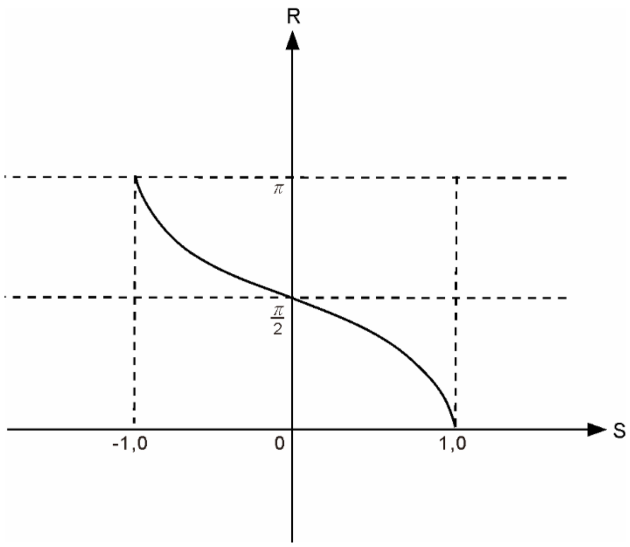
NO.	-	FACOS	-	S1, D	ACOS operation in floating point number format
030			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
S operand occupies 2 consecutive points and F device is available.	32-bit	FACOS	5 Step
D operand occupies 2 consecutive points.	16-bit	-	-

Description: S: source of specified cosine value (floating point number); D: radian result acquired from ACOS value (floating point number).

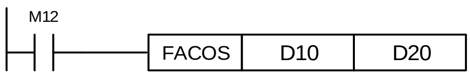
ACOS value = $\cos^{-1}$. The figure below shows the relation between the input data and result:



S: input data (cosine); R: result of ACOS value (radian)

The cosine value specified by S operand must be between -1.0 and +1.0. If the value is not within this range, this instruction is not executed.

Example: when M12 = On, obtain ACOS value from (D11, D10) and save it in (D21, D20), which is in floating point number format.



■ FATAN

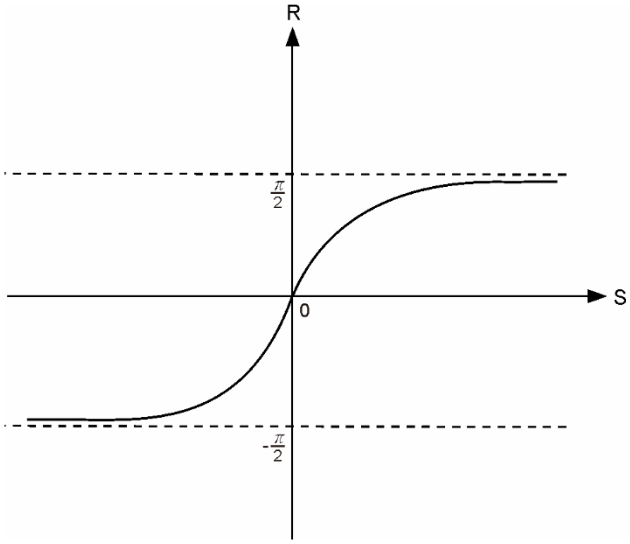
NO.	-	FATAN	-	S1, D	ATAN operation in floating point number format
031			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:		Instruction	Step No.
S operand occupies 2 consecutive points and F device is available.	32-bit	FATAN	5 Step
D operand occupies 2 consecutive points.	16-bit	-	-

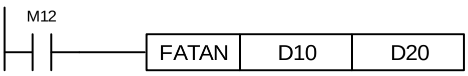
Description: S: source of specified tangent value (floating point number); D: radian result acquired from ATAN value (floating point number).

ATAN value = $\tan^{-1}$. The figure below shows the relation between the input data and result:



S: input data (tangent); R: result of ATAN value (radian)

Example: when M12 = On, obtain ATAN value from (D11, D10) and save it in (D21, D20), which is in floating point number format.



6

■ ZRST

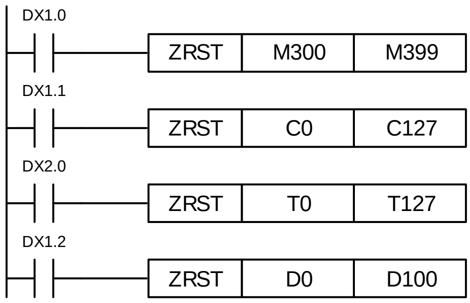
NO.	-	ZRST	-	D1, D2	Zone reset
032			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
D1	-	●	●	●	●	-	-				●	●	●	-	-	-	-	-
D2	-	●	●	●	●	-	-				●	●	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ZRST	4 Step

Description: D1: start device of zone reset; D2: end device of zone reset.
ZRST command supports 16-bit and 32-bit counters at the same time. When D1 operand ID is smaller than D2 operand ID, only the operand specified by D2 is reset.

Example: when DX1.0 = On, auxiliary relays M300 - M399 are reset to Off.
When DX1.1 = On, 16-bit counters C0 - C127 are reset. (Set the value to 0 and reset contacts and coils to Off.)
When DX2.0 = On, counters T0 - T127 are reset. (Set the value to 0 and reset contacts and coils to Off.)
When DX1.2 = On, data registers D0 - D100 are reset to 0.



6

■ DECO

NO. 033	-	DECO	- / -	S1, D, n	Decoder

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S	●	●	●	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-
D1	-	●	●	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-
D2	-	-	-	-	-	-	-	-	-	●	-	●	-	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	DECO	5 Step

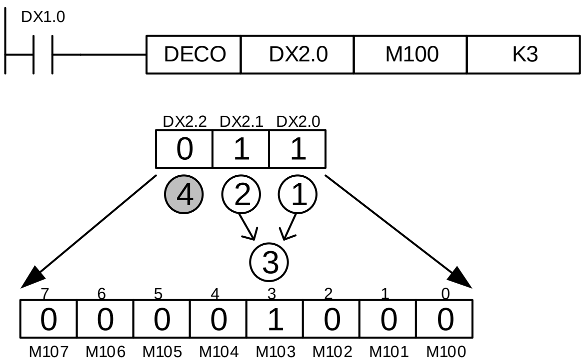
Description: S: source device for decoding; D: device for saving decoded value;

n: decoding bit length.

Use the lower n bit in source device S to decode and save the result with the 2 n bit length in D.

Example: when D is a bit device, n = 1 - 8. If n = 0 or n > 8, an error occurs.

If n = 8, this instruction can decode up to 2^8 = 256 points. When decoding, make sure the same range of storage device is not used repeatedly. When DX1.0 = On, the DECO instruction decodes values in DX2.0 - DX2.2 to M100 - M107. When the data source is 1 (bit 1 is on) + 2 (bit 2 is on) = 3, the 4^{th} bit (M103) from M100 is set to 1. When the DECO instruction is executed and DX1.0 is Off, the status of the data that has been decoded is unchanged.



6

■ ENCO

NO.	-	ENCO	-	S1, D, n	Encoder
034			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S	●	●	●	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-
D1	-	●	-	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-
D2	-	-	-	-	-	-	-	-	-	●	-	●	-	-	-	-	-	-

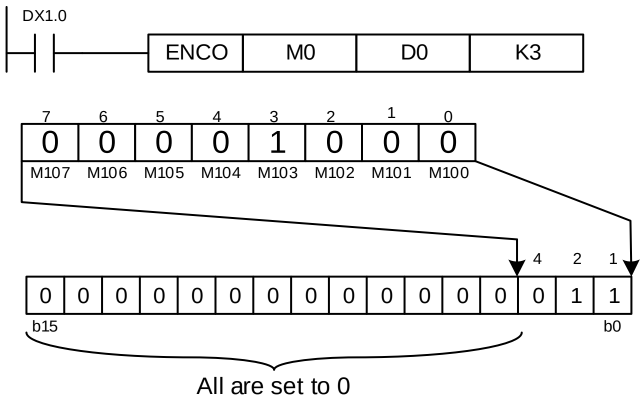
Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ENCO	5 Step

Description: S: source device for encoding; D: device for saving encoded value;

n: encoding bit length.

Use the data of the lower 2 n bit length in source device S to encode and save the result in D. If the data source S has multiple bits that are 1, the instruction will only process the first 1 bit from high bit to low bit.

Example: when S is a bit device, n = 1 - 8. If n = 0 or n > 8, an error occurs. If n = 8, this instruction can encode 2^8 = 256 points. When DX1.0 = On, the ENCO instruction encodes data M0 - M7 and saves it in lower 3 bits (b2 - b0) of D0. Bits (b15 - b3) that are not used in D0 are set to 0. When the ENCO instruction is executed and DX1.0 is Off, the data in device D is unchanged.



■ BON

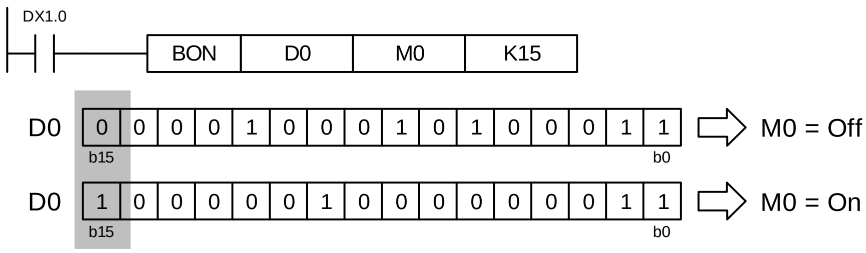
NO.	D	BON	-	S1, D, n	Monitor bit on
035			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S	-	-	-	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-
D	-	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
n	-	-	-	-	-	-	-	-	-	●	-		-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	DBON	5 Step
	16-bit	BON	5 Step

Description: S: source device; D: device for saving value; n: monitoring bit.

Example: when DX1.0 = On, if the value of the 15th bit in D0 is 1, then M0 = On; if the value is 0, then M0 = Off. If DX1.0 is Off, M0 remains unchanged.



■ ALT

NO. 036	-	ALT	- / -	D	ON / OFF alternate

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
D	-	●	●	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-

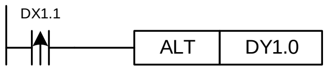
Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ALT	2 Step

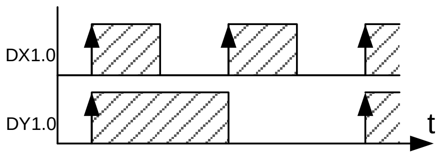
Description: D: target device.

When executing the ALT instruction, D alternates between On and Off.

Example: if DY1.0 is Off, when DX1.1 changes from Off to On for the first time, DY1.0 = On.

When D1.X0 changes from Off to On for the second time, DY1.0 = Off.





■ AO

NO. 037	-	AO	- / -	S1, S2	Analog output

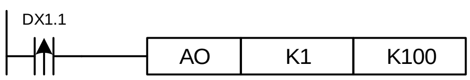
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	AO	5 Step

Description: S1: channel selection (K0 - K31); S2: output voltage.

You can apply the AO instruction to output voltage with the analog output module.

Example: when DX1.0 = On, the output voltage of the CH1 analog output module is 100 (unit varies according to different modes).



6

■ AI

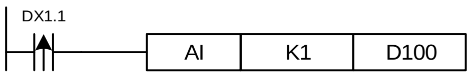
NO.	-	AI	-	S1, S2	Analog input
038			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	AI	5 Step

Description: S1: channel selection (K0 - K31); S2: input voltage.

Example: when DX1.0 = On, D100 displays the input value of the CH1 analog input module.



6.1.4 Single-axis motion instruction

PLC issues motion control instructions with the special register W and special relay R. The motion kernel can perceive the change of the PLC motion instruction by monitoring the register, and execute the corresponding motion control through bus communication (DMCNET or EtherCAT). The IMP editor will program the commonly used motion control commands into motion instructions, so when the PLC issues a motion instruction, the system automatically fills the parameters into the corresponding register position.

■ SVON

NO.	-	SVON	-	S1, S2	Servo on
050			-		

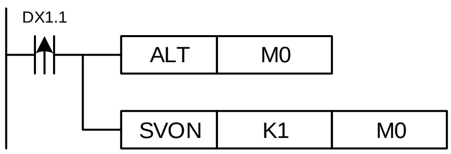
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	SVON	5 Step

Description: S1: servo axis No. (K1 - K36); S2: control flag of Servo ON / OFF.

When the SVON instruction is executed, the servo axis of node number S1 switches on / off according to the S2 state.

Example: when DX1.1 = On for the first time, the first servo axis is enabled; when DX1.1 = On for the second time, the first servo axis is disabled.



6

■ UNIT

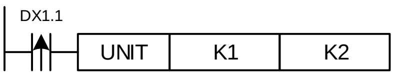
NO. 051	-	UNIT	- / -	S1, S2	Speed unit of single axis

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	UINT	5 Step

Description: S1: servo axis No. (K1 - K36); S2: unit setting (0: PUU/s; 1: %; 2: mm/min).
When applying the single-axis motion, you can use the UNIT instruction to select the speed unit.

Example: when DX1.1 = On, the speed unit of the first axis is mm.



■ SCUR

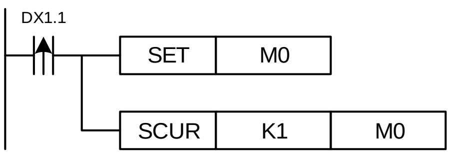
NO. 050	-	SCUR	- / -	S1, S2	Acceleration / deceleration curve setting

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	SCUR	5 Step

Description: S1: servo axis No. (K1 - K36); S2: Acceleration / deceleration curve type (On: S-Curve, Off: T-Curve).
You can use the SCUR instruction to change the acceleration / deceleration curve type.

Example: when DX1.1 = On, the acceleration / deceleration curve of the first servo axis is set to S-Curve.



6

■ TADC

NO. 053	-	TADC	- / -	S1, S2, S3	Acceleration / deceleration setting

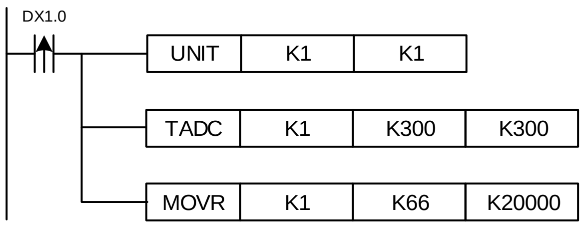
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

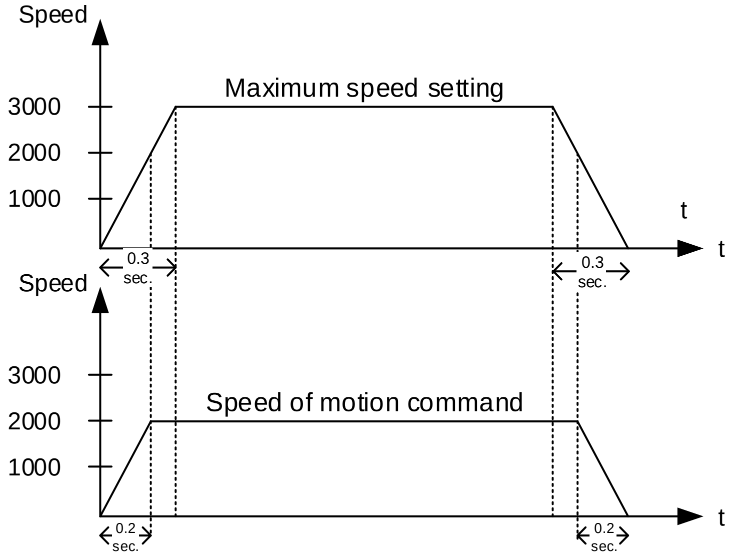
Notes when applying operand:		Instruction	Step No.
S2 operand occupies 2 consecutive points.	32-bit	TADC	11 Step
S3 operand occupies 2 consecutive points.	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: acceleration (unit: 1 ms); S3: deceleration (unit: 1 ms).

When the TADC instruction is executed, the set acceleration time is the time from zero speed to the set maximum speed and the set deceleration time is the time from the set maximum speed to zero speed. (You can set the maximum speed on the Quick Start interface.)

Example: assuming that the maximum speed of Axis 1 is set to 3000 rpm on the Quick Start interface, when DX1.0 = On, set the Axis 1 motion unit as percentage, acceleration / deceleration time as 0.3 seconds, and the axis will run 20 mm at the maximum speed of 2,000 rpm.





6

■ SLMT

NO. 054	-	SLMT	- / -	S1, S2, S3	Software limit setting

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
S2 operand occupies 2 consecutive points. S3 operand occupies 2 consecutive points.	32-bit	SLMT	11 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: positive software limit position (unit: 0.001 mm); S3: negative software limit position (unit: 0.001 mm).

Example: when the SLMT instruction is executed, the positive limit position is set to S2 for the S1 servo axis and the negative limit position is S3.

Note: this instruction is used with the SLMTON instruction.

■ SLMTON

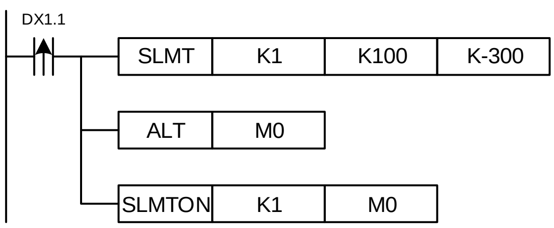
NO. 055	-	SLMTON	- / -	S1, S2	Software limit activation

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	SLMTON	5 Step

Description: S1: servo axis No. (K1 - K36); S2: off - disable software limit / on - enable software limit.

Example: when DX1.1 is on for the first time, the software lower limit of the first servo axis is set to 0.1 mm, the upper limit is set to -0.3 mm, and the function of software limit is enabled. When DX1.1 is on for the second time, the software limit is disabled.



■ COORD

NO. 056	-	COORD	- / -	S1, S2	Coordinates setting

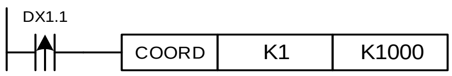
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand occupies 2 consecutive points.

	Instruction	Step No.
32-bit	-	-
16-bit	COORD	7 Step

Description: S1: servo axis No. (K1 - K36); S2: the set position (unit: 0.001 mm).
When the COORD instruction is executed, the specified servo axis coordinate is changed to S2.

Example: when DX1.1 = On, the current position of the first servo axis changes to 1000 (unit: 0.001 mm).



■ HOME

NO. 057	-	HOME	- / -	S1, S2, S3	Homing

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

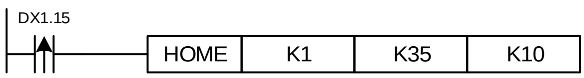
Notes when applying operand:
S3 operand occupies 2 consecutive points.

	Instruction	Step No.
32-bit	HOME	11 Step
16-bit	-	-

Description: S1: servo axis No. (K1 - K36): S2: homing mode; S3: offset value (unit: 0.001 mm).
When the HOME instruction is executed, the specified servo axis performs homing with the specified homing method. Then, set this origin coordinate to S3 using the offset value specified by S3.

Note: when applying homing mode, the switches of positive / negative limits and homing signal must be connected to the servo drive.

Example: when DX1.15 = On, the first servo axis finds the origin point with Homing mode 35 and sets this position as 0.010 mm.



6

■ ALMR

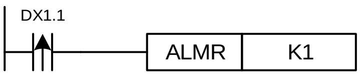
NO. 058	-	ALMR	- / -	S1					Clear alarm									
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ALMR	3 Step

Description: S1: servo axis No. (K1 - K36).

When servo alarm occurs, you can execute the ALMR instruction to clear the alarm.

Example: when DX1.1 = On, the alarm of the first servo axis will be cleared.



■ ESTP

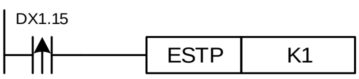
NO. 059	-	ESTP	- / -	S1					Emergency stop									
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ESTP	3 Step

Description: S1: servo axis No. (K1 - K36).

When the ESTP command is executed, the specified servo immediately decelerates at the maximum deceleration until the servo stops.

Example: when DX1.15 = On, the first servo axis immediately decelerates to stop at the maximum deceleration.



To avoid danger, it is suggested that the emergency stop function is used with the following functions:

1. The emergency stop signal triggers the servo DI emergency stop.
2. The emergency stop signal cuts off the servo power circuit.
3. Install safety circuits according to the actual equipment requirements.

■ SDSTP

NO.	-	SDSTP	-	S1	Decelerate to stop
060			-		

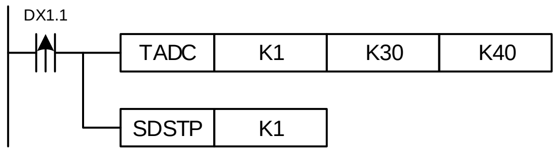
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	SDSTP	3 Step

Description: S1: servo axis No. (K1 - K36).

When the SDSTP instruction is executed, the servo axis of node number S1 decelerates to stop according to the deceleration setting.

Example: when DX1.1 = On, the first servo axis decelerates to stop in 0.04 seconds.



■ AXRPM

NO.	-	AXRPM	-	S1, D	Read motor’s current speed
061			-		

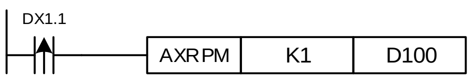
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand: D operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	AXRPM	7 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36).

You can obtain the current output speed of the servo motor (unit: 0.01 rpm) with the AXRPM instruction.

Example: when DX1.1 = On, D100 and D101 will display the current speed of the first servo axis.



6

■ AXTQR

NO.	-	AXTQR	-	S1, D	Read motor’s current torque
062			-		

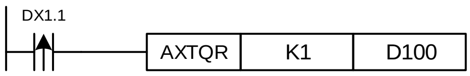
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	AXTQR	5 Step

Description: S1: servo axis No. (K1 - K36); D: torque of servo axis (‰).

You can obtain the current output torque of the servo axis (unit: 0.1%) with the AXTQR instruction.

Example: when DX1.1 = On, D100 displays the current torque of the first servo axis.



6

■ RSVP

NO.					
063	-	RSVP	- -	S1, S2	Read servo parameter

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2										●			●					

Notes when applying operand: D operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	RSVP	7 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: servo parameter group x 1000 + servo parameter No.

NO.					
063-1	-	SVR	- -	S1, S2	Read servo parameter return value

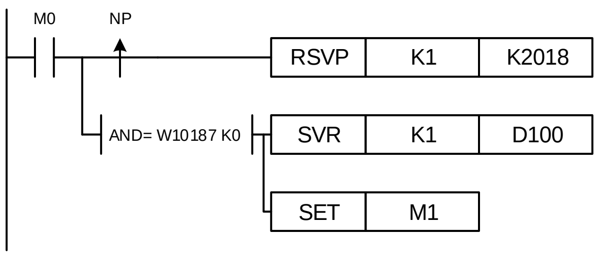
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2										●			●					

Notes when applying operand: D operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	SVR	7 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: read the servo parameter storage location.

You can read the servo parameters with the RSVP instruction and obtain the return value with the SVR instruction. The execution of the servo parameters is not instantaneous, so issue the RSVP, WSVP, and SVR instructions after making sure the execution is finished. This restriction does not include instructions for other servos.

Example: when M0 = On, read parameter P2-18 of servo axis 1 and write this value to D100 and D101. Then, set M1 to On.



6

■ WSVP

NO. 064	-	WSVP	- / -	S1, S2, D	Write servo parameter

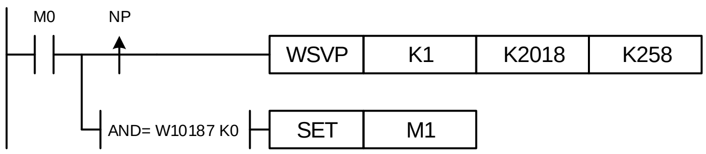
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2										●			●					
D	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: D operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	WSVP	7 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: servo parameter group x 1000 + servo parameter No.; D: write parameter.

You can write the servo parameters with the WSVP instruction. The execution of the servo parameters is not instantaneous, so issue the RSVP and WSVP instructions after making sure the execution is finished. This restriction does not include instructions for other axes.

Example: when M0 = On, write 258 (0x0102) to parameter P2-18 of servo axis 1, then set M1 to On.



To avoid danger, it is suggested to execute the write-in parameter function when servo off.

6

■ SVSTS

NO.					
065	-	SVSTS	- -	S1, D	Read servo DO status

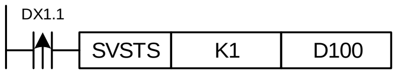
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	SVSTS	5 Step

Description: S1: servo axis No. (K1 - K36); D: DO status of servo drive.

You can obtain the DO status of the servo drive with the SVSTS instruction.

Example: when DX1.1 = On, the DO status of the first servo drive is saved in D100.



■ SVITS

NO.					
066	-	SVITS	- -	S1, D	Read servo DI status

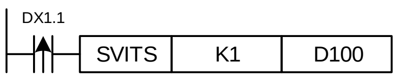
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	SVITS	5 Step

Description: S1: servo axis No. (K1 - K36); D: DI status of servo drive.

You can obtain the DI status of the servo drive with the SVITS instruction.

Example: when DX1.1 = On, the DI status of the first servo drive is saved in D100.



■ RCBL

NO.	-	RCBL	-	S1, D	Read the buffer memory usage
067			-		

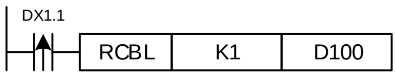
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	RCBL	5 Step

Description: S1: servo axis No. (K1 - K36); D: buffer memory usage.

You can use the RCBL instruction to read the motion command buffer status.

Example: when DX1.1 = On, save the number of motion commands in the buffer of the first axis to D100.



■ RPOS

NO.	-	RPOS	-	S1, S2	Read the actual position of the motor axis
068			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

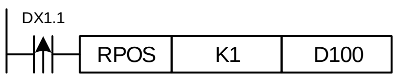
Notes when applying operand: D operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	RPOS	7 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: actual position of servo axis (unit: 0.001 mm).

You can obtain the actual position of the motor's driving mechanism with the RPOS instruction.

Note: the servo motor sends the feedback coordinates through the encoder, which is called the actual position.

Example: when DX1.1 = On, the feedback position of the first servo motor is saved to (D101, D100).



6

■ LPOS

NO.					
069	-	LPOS	- / -	S1, S2	Read the axial instruction position

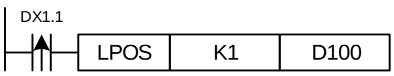
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
D operand occupies 2 consecutive points.	32-bit	LPOS	7 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: instruction position of servo axis (unit: 0.001 mm).
You can obtain the instruction position issued by the current motion command with the LPOS instruction.

Note: when you issue any motion commands, the IMP control system divides the path from the current position to the target position into several nodes. Then, the command is sent every 1 ms, and this current command is called the instruction position.

Example: when DX1.1 = On, the instruction position of the first servo axis is saved to (D101, D100).



■ TPOS

NO.					
070	-	TPOS	- / -	S1, S2	Read the axial target position

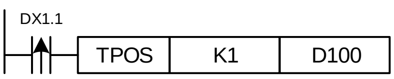
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
D operand occupies 2 consecutive points.	32-bit	TPOS	7 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: target position of servo axis (unit: 0.001 mm).
You can obtain the target position of the executed command with the TPOS instruction.

Note: the final position specified by the motion command is called the target position.

Example: when DX1.1 = 0, the command target of the first servo axis is saved to (D101, D100).



■ MOTS

NO.	-	MOTS	-	S1, D	Read the status of the motion instruction
071			-		

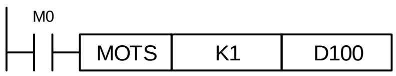
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	MOTS	5 Step

Description: S1: servo axis No. (K1 - K36); D: status of motion command.

You can read the motion status of the servo axis with the MOTS instruction.

Example: when DX1.1 = On, D100 displays the motion status of the first servo axis.



■ ALE

NO.	-	ALE	-	S1, D	Read error code
072			-		

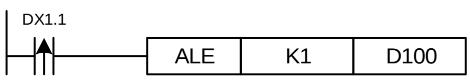
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ALE	5 Step

Description: S1: servo axis No. (K1 - K36); D: error code of servo axis.

You can obtain the error code of the servo drive with the ALE instruction.

Example: when DX1.1 = On, D100 displays the error code of the first servo axis.



■ JOG

NO.	-	JOG	-	S1, S2, S3	Jog
073			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

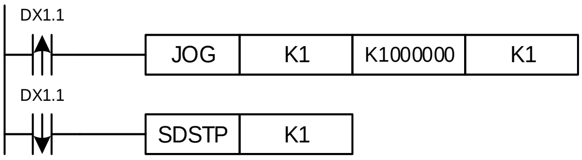
Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	JOG	11 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: jog speed (default unit: PUU/s; can be changed by the UNIT instruction); S3: operation direction.

When the JOG instruction is executed, the servo motor specified by S1 accelerates at the acceleration limit, and once the speed reaches the specified speed of S2, it runs at a constant speed. After the JOG instruction is finished, the servo motor decelerates according to the deceleration limit until it comes to a complete stop.

The minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when DX1.1 = On, the first servo axis runs at the speed of 1,000,000 PUU/s in reverse direction.



6

■ MOVA

NO.	-	MOVA	-	S1, S2, S3	Absolute motion
074			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

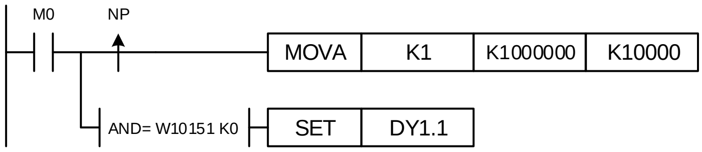
Notes when applying operand:		Instruction	Step No.
S2 operand occupies 2 consecutive points.	32-bit	MOVA	11 Step
S3 operand occupies 2 consecutive points.	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: motion speed (default unit: PUU/s; can be changed by the UNIT instruction); S3: target position.

When the MOVA instruction is executed, the servo motor of node number S1 runs at the speed set by S2 and stops when it reaches the coordinates specified by S3.

The minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when M0 = On, the first servo axis runs at the speed of 1,000,000 PUU/s to the position of 10 mm, and DY1.1 is on when it reaches the position.



■ MOVR

NO.	-	MOVR	-	S1, S2, S3	Relative motion
075			-		

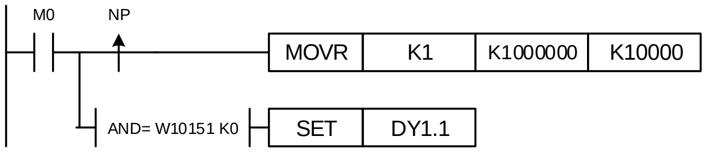
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
S2 operand occupies 2 consecutive points. S3 operand occupies 2 consecutive points.	32-bit	MOVR	11 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: motion speed (default unit: PUU/s; can be changed by the UNIT instruction); S3: motion target (unit: 0.001 mm).

When the MOVR instruction is executed, the servo motor of node number S1 runs at the speed set by S2 and stops when it reaches the coordinates of [current position + S3]. The minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when M0 = On, the first servo axis runs at the speed of 1,000,000 PUU/s to the position of [current position + 10 mm], and DY1.1 is on when it reaches the position.



■ MOVPOS

NO.					
NO. 076	-	MOVPOS	- / -	S1, S2	Position change during operation

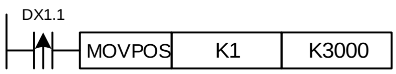
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand occupies 2 consecutive points.

	Instruction	Step No.
32-bit	MOVPOS	7 Step
16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: motion position (unit: 0.001 mm).
When the MOVPOS instruction is executed, the servo axis of node number S1 updates its target position to the one set by S2. This instruction can only be applied during MOVA motion.

Example: when DX1.1 = On, the target position of the first servo axis is changed to 3.000 mm.



■ MOVSPD

NO.					
NO. 077	-	MOVSPD	- / -	S1, S2	Speed change during operation

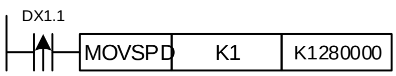
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand occupies 2 consecutive points.

	Instruction	Step No.
32-bit	MOVSPD	7 Step
16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: motion speed (default unit: PUU/s; can be changed by the UNIT instruction).
When the MOVSPD instruction is executed, the servo motor of node number S1 changes to the speed set by S2. This instruction can only be applied during MOVA and MOVR motions, and the minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when DX1.1 = On, the speed of the first servo axis is changed to 1,280,000 PUU/s.



■ SPD

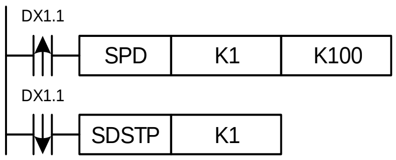
NO.	-	SPD	-	S1, S2	Speed control
078			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	SPD	7 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: motion speed (unit: 0.1 rpm).
When the SPD instruction is executed, the servo motor of node number S1 accelerates according to the acceleration setting, and once the speed reaches the specified speed of S2, it runs at a constant speed.

Example: when DX1.1 = On, the first servo axis runs at the speed of 10.0 rpm in forward direction; when DX1.1 = Off, the first servo axis decelerates to stop.



■ TRQ

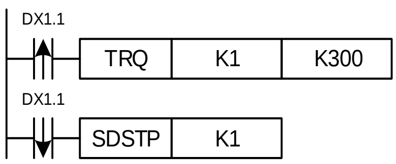
NO.	-	TRQ	-	S1, S2	Torque control
079			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	TRQ	7 Step
	16-bit	-	-

Description: S1: servo axis No. (K1 - K36); S2: output target torque (±1000)(‰).
When the TQR instruction is executed, the servo axis of node number S1 outputs the torque specified by S2.

Example: when DX1.1 = On, the first servo axis runs at the rated torque of 300‰ in forward direction.



6

■ RSPD

NO.	-	RSPD	-	S1, D	Read current speed
080			-		

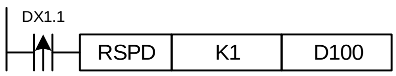
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

Notes when applying operand:
D operand occupies 2 consecutive points.

	Instruction	Step No.
32-bit	RSPD	7 Step
16-bit	-	-

Description: S1: servo axis No. (K1 - K36); D: servo axis speed (unit: 0.001 mm/min).
You can obtain the speed of the motor's driving mechanism with the RSPD instruction.

Example: when DX1.1 = On, the speed of the driving mechanism for the first servo axis is saved to (D101, D100).



6.1.5 Interpolation motion instruction

The IMP system supports multi-axis synchronous motion, which can achieve linear interpolation of any three axes, helical interpolation of any two or three axes, and continuous interpolation on the same motion card.

Group setting GSET ⇨ Parameter setting ⇨ Target setting of each axis MOVP ⇨ Motion triggered

Parameter setting table for group motion:

	GUNIT	GSCUR	GTADC	ANGLE	DIR	PITCH	DEPTH	CENTER	ENDXY
MOVLA	△	△	△	-	-	-	-	-	-
MOVLR	△	△	△	-	-	-	-	-	-
CIRCAA	△	△	△	●	△	-	-	●	
CIRCAR	△	△	△	●	△	-	-	●	
CIREAA	△	△	△	●	△	-	-	-	●
CIREAR	△	△	△	●	△	-	-	-	●
CIRCEA	△	△	△	-	△	-	-	●	●
CIRCER	△	△	△	-	△	-	-	●	●
HELIXA	△	△	△	-	●	●	●	●	-
HELIXR	△	△	△	-	●	●	●	●	-

●: required / △: optional / -: invalid

■ GSET

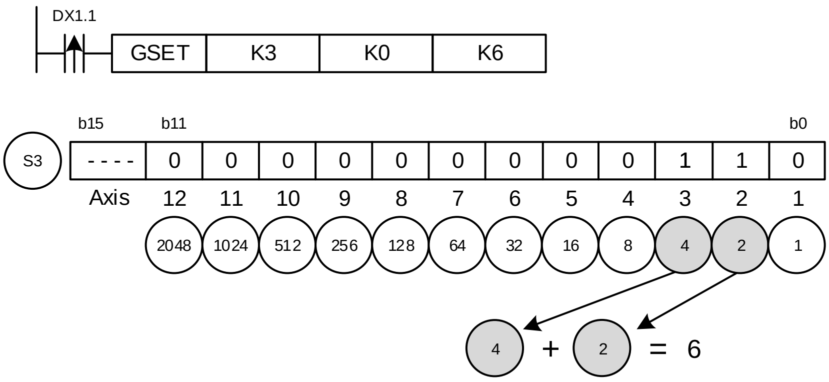
NO.	-	GSET	-	S1, S2, S3	Group setting
100			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	GSET	7 Step

Description: S1: group No. (K1 - K40); S2: card No.; S3: setting of the applied axis (bit).
Before applying the interpolation function, you must assign the servo axes as a group with the GSET instruction.

Example: when DX1.1 = On, the second and third axes of card No. 0 are assigned as motion group 3.



6

■ GUNIT

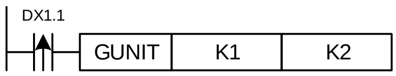
NO.	-	GUNIT	-	S1, S2	Group setting of speed unit
101			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	GUINT	5 Step

Description: S1: group No. (K1 - K40); S2: unit setting (0: PUU/s; 1: %; 2: mm/min).
When applying the group interpolation function, you can use the GUNIT instruction to select the speed unit.

Example: when DX1.1 = On, the unit of the first motion group is set to mm/min.



■ GSCUR

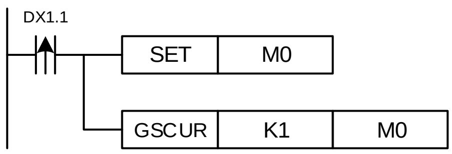
NO.	-	GSCUR	-	S1, S2	Group setting of acceleration / deceleration curve
102			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	GSCUR	5 Step

Description: S1: group No. (K1 - K40); S2: control flag of acceleration / deceleration curve (Off: T-curve; On: S-curve).
When applying the group interpolation function, you can use the GSCUR instruction to select the acceleration / deceleration smooth curve.

Example: when DX1.1 = On, the acceleration / deceleration curve of the first motion group is set to S-curve.



6

■ GTADC

NO.	-	GTADC	-	S1, S2, S3	Group setting of acceleration / deceleration time
103			-		

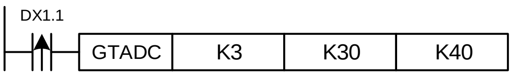
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points. S3 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	GTADC	11 Step
	16-bit	-	-

Description: S1: group No. (K1 - K40); S2: acceleration time (unit: ms); S3: deceleration time (unit: ms).

When applying the group interpolation function, you can use the GTADC instruction to set the acceleration / deceleration time.

Example: when DX1.1 = On, the acceleration and deceleration times of the third motion group are set to 0.03 seconds and 0.04 seconds, respectively.



■ ANGLE

NO.	-	ANGLE	-	S1, S2	Arc angle
104			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

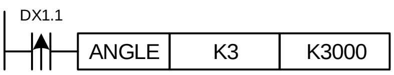
Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	ANGLE	5 Step

Description: S1: group No. (K1 - K40); S2: arc angle (unit: 0.1 degree).

When applying the arc motion instruction, you can use the ANGLE instruction to set the angle of the arc.

Note: this instruction is used with the arc interpolation instruction.

Example: when DX1.1 = On, the arc angle of the third motion group is set to 300 degrees.



6

■ DIR

NO.	-	DIR	-	S1, S2	Arc direction
105			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

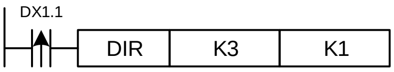
Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	DIR	5 Step

Description: S1: group No. (K1 - K40); S2: arc direction (0: clockwise; 1: counterclockwise).

When applying the arc motion instruction, you can use the DIR instruction to set the direction of the arc.

Note: this instruction is used with the arc interpolation instruction.

Example: when DX1.1 = On, the arc direction of the third motion group is set to counterclockwise.



■ PITCH

NO.	-	PITCH	-	S1, S2	Helix pitch
106			-		

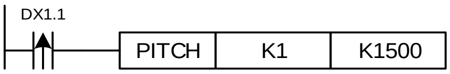
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	PITCH	7 Step
	16-bit	-	-

Description: S1: group No. (K1 - K40); S2: helix pitch (unit: 0.001 mm).

When applying the helical motion instruction, you can use the PITCH instruction to set the pitch of the helix.

Example: when DX1.1 = On, the helix pitch of the third motion group is set to 1.5 mm.



6

■ DEPTH

NO.	-	DEPTH	-	S1, S2	Helix depth
107			-		

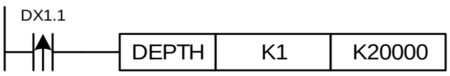
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	DEPTH	7 Step
	16-bit	-	-

Description: S1: group No. (K1 - K40); S2: helix depth (unit: 0.001 mm).

When applying the helical motion instruction, you can use the DEPTH instruction to set the total depth of the helix.

Example: when DX1.1 = On, the helix depth of the third motion group is set to 20 mm.



■ CENTER

NO.	-	CENTER	-	S1, S2, S3	Arc center
108			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points. S3 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	CENTER	11 Step
	16-bit	-	-

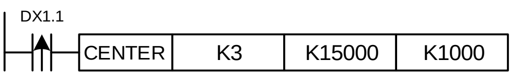
Description: S1: group No. (K1 - K40); S2: X-coordinate of arc center (unit: 0.001 mm);

S3: Y-coordinate of arc center (unit: 0.001 mm).

When applying the arc motion instruction, you can use the CENTER instruction to set the circle center of the arc.

Note: this instruction is used with the arc interpolation instruction.

Example: when DX1.1 = On, the arc center of the third motion group is set to (150.00, 10.00).



■ ENDXY

NO.	-	ENDXY	-	S1, S2, S3	Endpoint of arc
109			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points. S3 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	ENDXY	11 Step
	16-bit	-	-

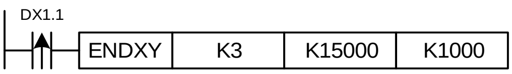
Description: S1: group No. (K1 - K40); S2: X-coordinate of arc endpoint (unit: 0.001 mm);

S3: Y-coordinate of arc endpoint (unit: 0.001 mm).

When applying the arc motion instruction, you can use the ENDXY instruction to set the endpoint of the arc.

Note: this instruction is used with the arc interpolation instruction.

Example: when DX1.1 = On, the arc endpoint of the third motion group is set to (15.000 mm, 1.000 mm).



■ MOVP

NO.	-	MOVP	-	S1, S2, S3	Target setting for each axis
110			-		

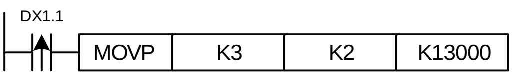
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S3 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	MOVP	13 Step
	16-bit	-	-

Description: S1: group No. (K1 - K40); S2: target axis No. (K1 - K12); S3: target position (unit: 0.001 mm).

When applying the multi-axis synchronous function, you can use the MOVP instruction to set the target position of each axis.

Example: when DX1.1 = On, the target position of Axis 2 in the third motion group is set to 13 mm.



■ MOVLA

NO.	-	MOVLA	-	S1, S2	Linear motion (absolute)
111			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand occupies 2 consecutive points.

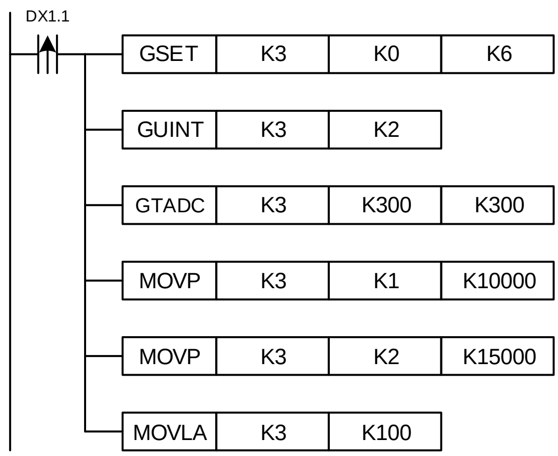
	Instruction	Step No.
32-bit	MOVLA	7 Step
16-bit	-	-

Description: S1: group No. (K1 - K40); S2: target speed (default unit: PUU/s, can be changed by the GUNIT instruction).

To use the multi-axis linear absolute motion, you can use the MOVLA instruction to trigger this function. The minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when DX1.1 = On:

1. The third motion group is set to use motion card No. 0 and the servo drives of node numbers 2 and 3.
2. The speed unit of the third motion group is set to mm/min.
3. The acceleration and deceleration times of the third motion group are set to 0.3 seconds and 0.3 seconds, respectively.
4. The target position of Axis 1 in the third motion group is set to 10.000 mm.
5. The target position of Axis 2 in the third motion group is set to 15.000 mm.
6. Trigger the linear interpolation of two-axis (absolute) in the third motion group at the speed of 100/min.



■ MOVLR

NO.	-	MOVLR	-	S1, S2	Linear motion (relative)
112			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand occupies 2 consecutive points.

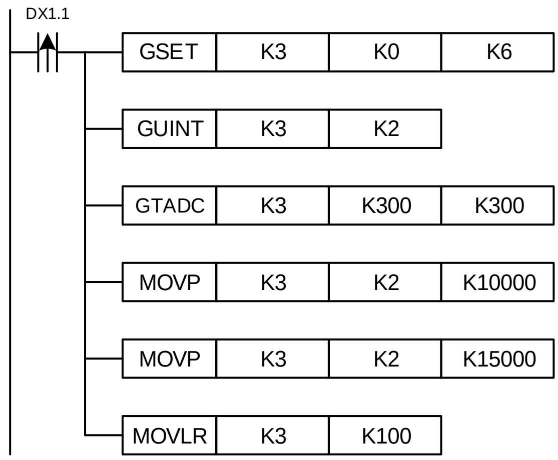
	Instruction	Step No.
32-bit	MOVLR	7 Step
16-bit	-	-

Description: S1: group No. (K1 - K40); S2: target speed (default unit: PUU/s, can be changed by the GUNIT instruction).

To use the multi-axis linear relative motion, you can use the MOVLR instruction to trigger this function. The minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when DX1.1 = On:

1. The third motion group is set to use motion card No. 0 and the servo drives of node numbers 2 and 3.
2. The speed unit of the third motion group is set to mm/min.
3. The acceleration and deceleration times of the third motion group are set to 0.3 seconds and 0.3 seconds, respectively.
4. Set Axis 1 in the third motion group to increment by 10.000 mm.
5. Set Axis 1 in the third motion group to increment by 15.000 mm.
6. Trigger the linear interpolation of two-axis (relative) in the third motion group at the speed of 100/min.



6

■ CIRCAA

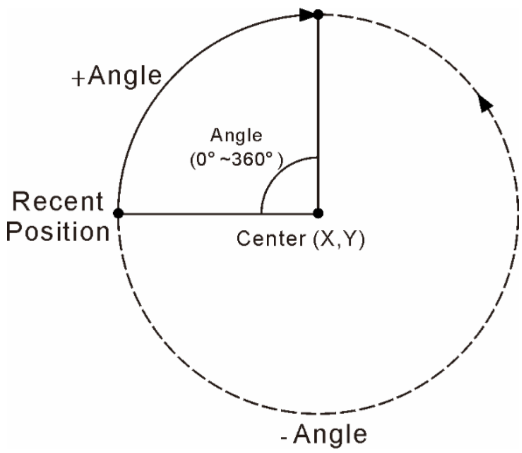
NO.					
113	-	CIRCAA	- / -	S1, S2	Arc absolute motion (known center coordinates and angle)

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	CIRCAA	7 Step
	16-bit	-	-

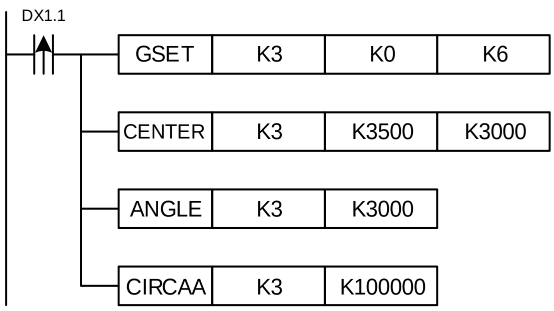
Description: S1: group No. (K1 - K40); S2: target speed (default unit: PUU/s, can be changed by the GUNIT instruction).

To use the multi-axis arc absolute motion, with the known arc center coordinates and angle, you can use the CIRCAA instruction to trigger this function. The minimum speed limit of this instruction is 5 PUU/s or 1%.



Example: when DX1.1 = On:

1. The third motion group is set to use motion card No. 0 and the servo drives of node numbers 2 and 3.
2. The arc center position of the third motion group is set to (3.500, 3.000) mm.
3. The arc angle of the third motion group is set to 300 degrees.
4. Trigger the two-axis arc absolute motion of the third motion group.



■ CIRCAR

NO.	-	CIRCAR	-	S1, S2	Arc relative motion (known center coordinates and angle)
114			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

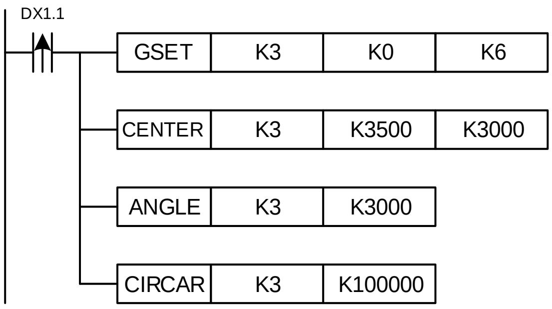
Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	CIRCAR	7 Step
	16-bit	-	-

Description: S1: group No. (K1 - K40); S2: target speed (default unit: PUU/s, can be changed by the GUNIT instruction).

To use the multi-axis arc relative motion, with the known arc center coordinates and angle, you can use the CIRCAR instruction to trigger this function. The minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when DX1.1 = On:

1. The third motion group is set to use motion card No. 0 and the servo drives of node numbers 2 and 3.
2. The circle center position of the third motion group is set to (Current position X + 3.500, Current position Y + 3.000) mm.
3. The arc angle of the third motion group is set to 300 degrees.
4. Trigger the two-axis arc relative motion of the third motion group.



■ CIREAA

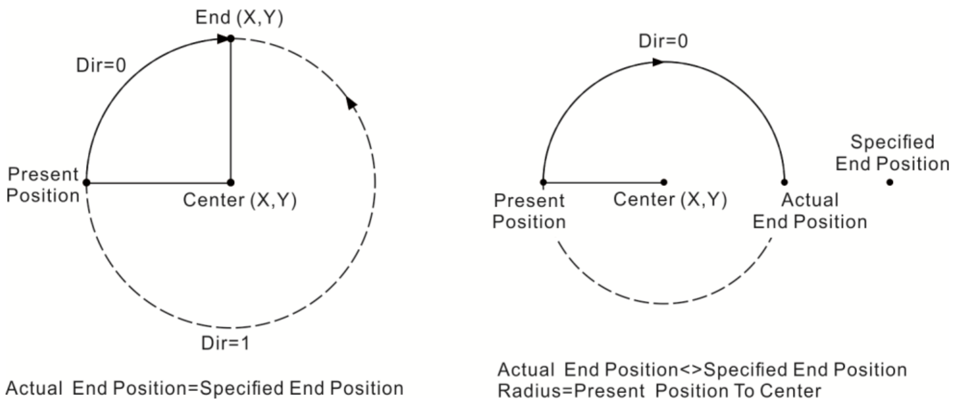
NO.	-	CIREAA	-	S1, S2	Arc absolute motion (known endpoint coordinates and angle)
115			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	CIREAA	7 Step
	16-bit	-	-

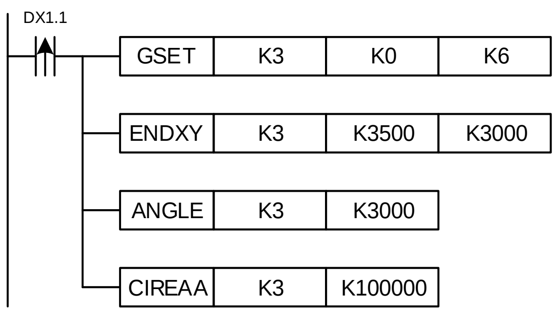
Description: S1: group No. (K1 - K40); S2: target speed (default unit: PUU/s, can be changed by the GUNIT instruction).

To use the multi-axis arc absolute motion, with the known endpoint coordinates and arc angle, you can use the CIREAA instruction to trigger this function. The minimum speed limit of this instruction is 5 PUU/s or 1%.



Example: when DX1.1 = On:

1. The third motion group is set to use motion card No. 0 and the servo drives of node numbers 2 and 3.
2. The endpoint position of the third motion group is set to (3.500, 3.000) mm.
3. The arc angle of the third motion group is set to 300 degrees.
4. Trigger the two-axis arc absolute motion of the third motion group.



6

■ CIREAR

NO.	-	CIREAR	-	S1, S2	Arc relative motion (known endpoint coordinates and angle)
116			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

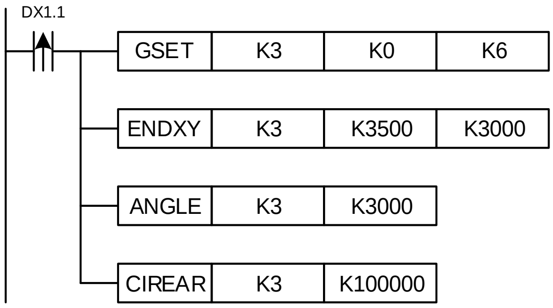
Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	CIREAR	7 Step
	16-bit	-	-

Description: S1: group No. (K1 - K40); S2: target speed (default unit: PUU/s, can be changed by the GUNIT instruction).

To use the multi-axis arc relative motion, with the known endpoint coordinates and arc angle, you can use the CIREAR instruction to trigger this function. The minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when DX1.1 = On:

1. The third motion group is set to use motion card No. 0 and the servo drives of node numbers 2 and 3.
2. The endpoint position of the third motion group is set to (Current position X + 3.500, Current position Y + 3.000) mm.
3. The arc angle of the third motion group is set to 300 degrees.
4. Trigger the two-axis arc relative motion of the third motion group.



■ CIRCEA

NO.	-	CIRCEA	-	S1, S2	Arc absolute motion (known center and endpoint coordinates)
117			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

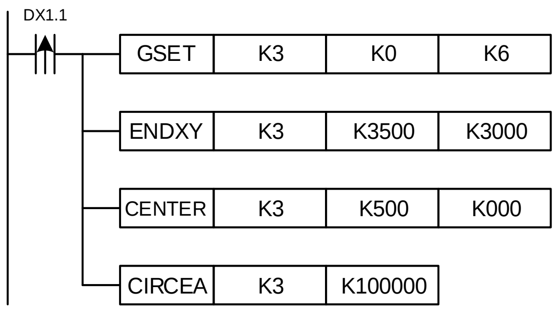
Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	CIRCEA	7 Step
	16-bit	-	-

Description: S1: group No. (K1 - K40); S2: target speed (default unit: PUU/s, can be changed by the GUNIT instruction).

To use the multi-axis arc absolute motion, with the known arc center and endpoint coordinates, you can use the CIRCEA instruction to trigger this function. The minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when DX1.1 = On:

1. The third motion group is set to use motion card No. 0 and the servo drives of node numbers 2 and 3.
2. The endpoint position of the third motion group is set to (3.500, 3.000) mm.
3. The arc center position of the third motion group is set to (0.500, 0.000) mm.
4. Trigger the two-axis arc absolute motion of the third motion group.



■ CIRCER

NO.					
118	-	CIRCER	- / -	S1, S2	Arc relative motion (known center and endpoint coordinates)

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	CIRCER	7 Step
	16-bit	-	-

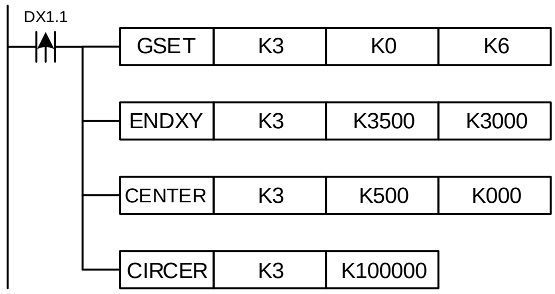
Description: S1: group No. (K1 - K40); S2: target speed (default unit: PUU/s, can be changed by the GUNIT instruction).

To use the multi-axis arc relative motion, with the known circle center and endpoint coordinates, you can use the CIRCER instruction to trigger this function.

The minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when DX1.1 = On:

1. The third motion group is set to use motion card No. 0 and the servo drives of node numbers 2 and 3.
2. The endpoint position of the third motion group is set to (Current position X + 3.500, Current position Y + 3.000) mm.
3. The arc center position of the third motion group is set to (Current position X + 0.500, Current position Y + 0.000) mm.
4. Trigger the two-axis arc relative motion of the third motion group.



■ HELIXA

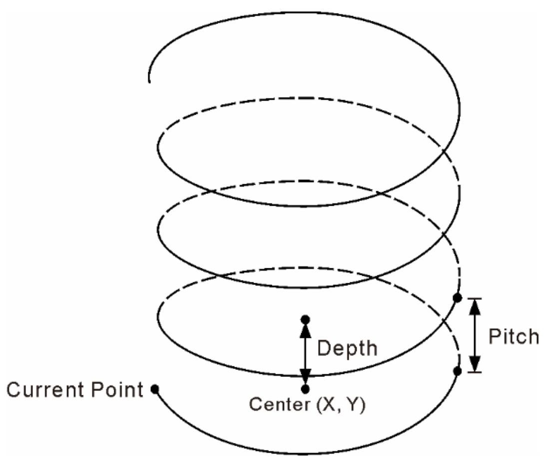
NO.	-	HELIXA	-	S1, S2	Helical absolute motion
119			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	HELIXA	7 Step
	16-bit	-	-

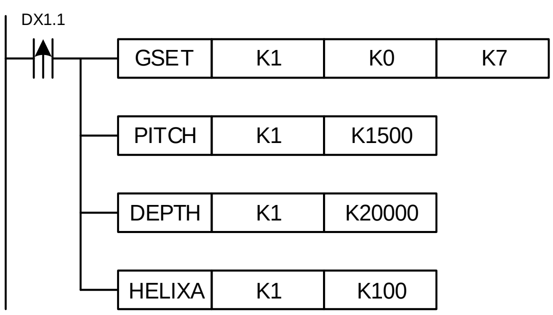
Description: S1: group No. (K1 - K40); S2: target speed (default unit: PUU/s, can be changed by the GUNIT instruction).

To use the three-axis helical interpolation motion for moving to the absolute height position, you can use the HELIXA instruction to trigger this function. The minimum speed limit of this instruction is 5 PUU/s or 1%.



Example: when DX1.1 = On:

1. The third motion group is set to use motion card No. 0 and the servo drives of node numbers 1, 2, and 3.
2. The third motion group is set to rise the height of 1.5 mm per rotation.
3. The target position of upward rotation for the third motion group is set to 20.000 mm.
4. The third motion group is triggered to execute three-axis helical motion (absolute position) at the speed of 100 PUU.



6

■ HELIXR

NO.	-	HELIXR	-	S1, S2	Helical relative motion
120			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand occupies 2 consecutive points.		Instruction	Step No.
	32-bit	HELIXR	7 Step
	16-bit	-	-

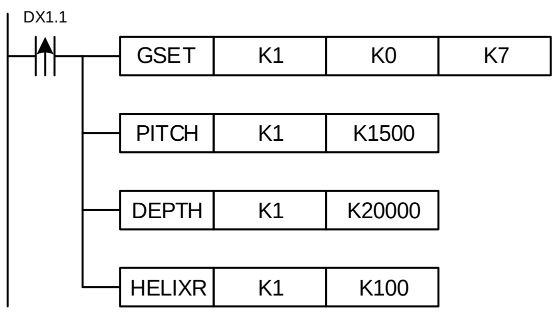
Description: S1: group No. (K1 - K40); S2: target speed (default unit: PUU/s, can be changed by the GUNIT instruction).

To use the three-axis helical interpolation motion for moving the height relative to the current position, you can use the HELIXR instruction to trigger this function.

The minimum speed limit of this instruction is 5 PUU/s or 1%.

Example: when DX1.1 = On:

1. The third motion group is set to use motion card No. 0 and the servo drives of node numbers 2 and 3.
2. The third motion group is set to rise the height of 1.5 mm per rotation.
3. The target position of upward rotation for the third motion group is set to current height + 20.000 mm.
4. The third motion group is triggered to execute three-axis helical motion (relative position) at the speed of 100 PUU.



6

■ GESTP

NO.	-	GESTP	-	S1	Group emergency stop
121			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	GESTP	3 Step

Description: S1: group No. (K1 - K40).

When the GESTP instruction is executed, servo motor in the S1 group decelerates to stop at the maximum deceleration speed.

Example: when DX1.1 = On, all servo axes in the third motion group will decelerate to stop at the maximum deceleration speed.

■ GSDSTP

NO.	-	GSDSTP	-	S1	Group deceleration to stop
122			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

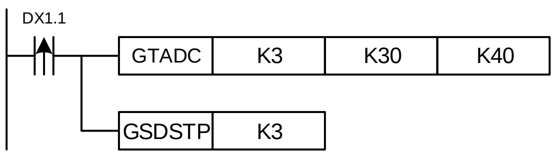
Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	GSDSTP	3 Step

Description: S1: group No. (K1 - K40).

When the GSDSTP instruction is executed, the servo motor in the S1 group decelerates to stop according to the deceleration time setting (GTADC).

Example: when DX1.1 = On, all servo axes in the third motion group will decelerate to stop according to the deceleration time setting of 0.4 seconds.



6.1.6 Motion program macro (MPM) control instruction

■ MPMST

NO. 150	-	MPMST	- / -	S1									MPM starts					
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	MPMST	3 Step

Description: S1: group No. (K0 - K99).

Enable MPM to start running the motion program macro.

Example: when DX1.1 = On, execute the motion program macro (MPM) for the first group.

■ MPMSTP

NO. 151	-	MPMSTP	- / -	S1									MPM stops					
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

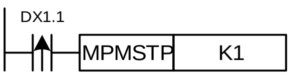
Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	MPMSTP	3 Step

Description: S1: group No. (K0 - K99).

Disable MPM to stop running the motion program macro and stop all servo motions controlled by this MPM.

Example: when DX1.1 = On, stop the motion program macro (MPM) for the first group.



■ MPMPAU

NO.	-	MPMPAU	-	S1	MPM pauses
152			-		

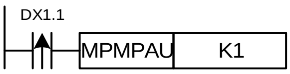
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	MPMPAU	3 Step

Description: S1: group No. (K0 - K99).

This instruction pauses the motion program macro, and the process resumes when the MPMST instruction is issued.

Example: when DX1.1 = On, pause the motion program macro (MPM) for the first group.



■ MPMSPD

NO.	-	MPMSPD	-	S1, S2	MPM speed changes
153			-		

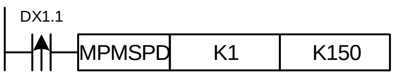
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	MPMSPD	5 Step

Description: S1: group No. (K0 - K99); S2: speed change percentage (valid range is 0 - 1000%).

To change the MPM running speed, its range should be between 0 - 1000% of the original speed.

Example: when DX1.1 = On, the speed of the motion program macro (MPM) for the first group is changed to 150%.



■ MPMER

NO.	-	MPMER	-	S1, D	Read MPM error code
154			-		

	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

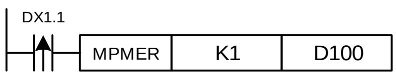
Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	MPMER	5 Step

Description: S1: group No. (K0 - K99).

The following is a list of MPM error codes.

Error code	Description
1	The motion program macro (MPM) failed to start, and the servo axis is in use.
2	The group axis is not enabled (SVON).
3	An error occurred in the group axis.
4	The group axis triggered the positive limit.
5	The group axis triggered the negative limit.
6	The instruction exceeded the software positive limit.
7	The instruction exceeded the software negative limit.
8	When MPM is executing the homing instruction, the execution is interrupted.
9	The loaded external file does not exist.
10	MPM programming syntax error
11	Loading the external file has failed.

Example: when DX1.1 = On, save the obtained error code of the motion program macro (MPM) for the first group in D100.



6

■ MSTEP

NO.	-	MSTEP	-	S1, D1, D2	Read the MPM step No.
155			-		

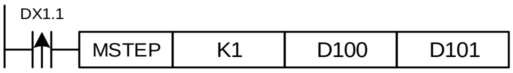
	Bit device						Word device										External device	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	Bit	Character
S1										●			●					
D1	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-
D2	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
	32-bit	-	-
	16-bit	MSTEP	7 Step

Description: S1: group No. (K0 - K99); D1: total step No.; D2: executed step No.

Read the step No. that has been executed and the executable step No. of MPM.

Example: when DX1.1 = On, save the total step No. of the motion program macro (MPM) for the first group in D100 and the executed step No. in D101.



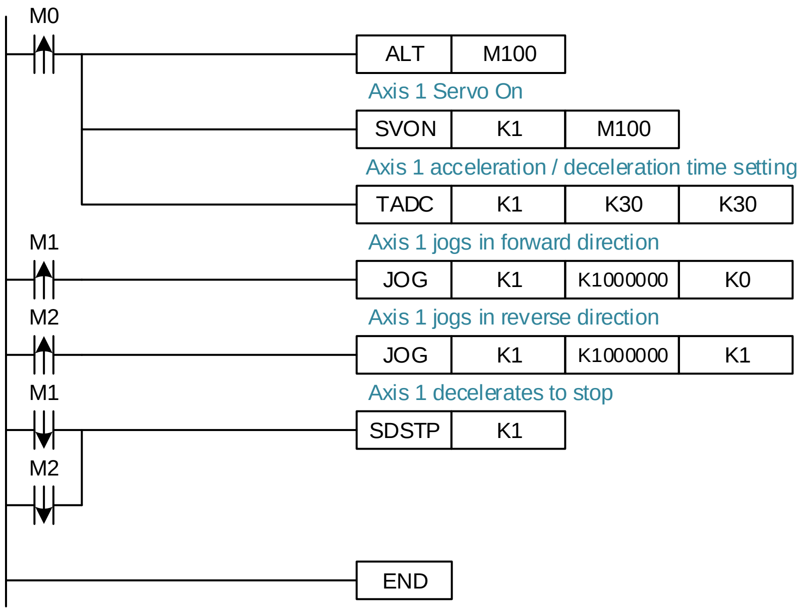
6.1.7 Program example

■ Jog

Description: the servo motor runs in jog mode. Press the jog (+) key when the servo axis is on, and the motor runs in the forward direction; press the jog (-) key, and the motor runs in the reverse direction.

Parameter definition: M0 is the bit for triggering servo on / off; M1 is the bit for controlling the jog operation in forward direction; M2 is the bit for controlling the jog operation in reverse direction.

Example:



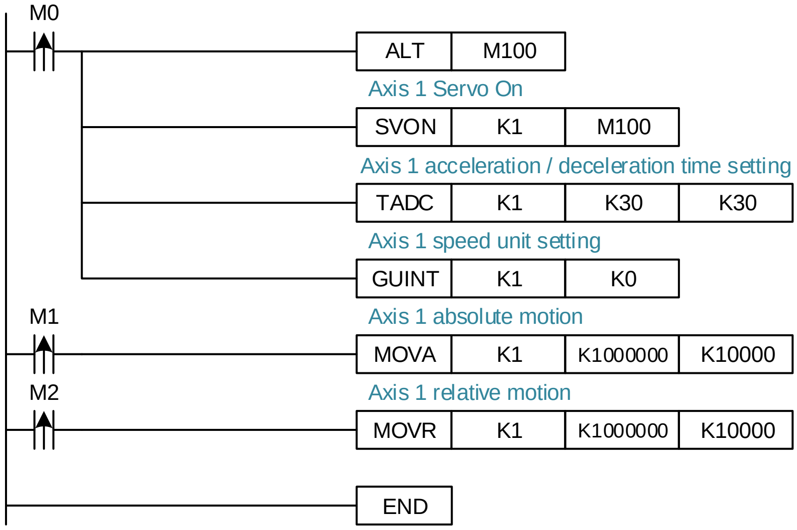
6

■ Single axis motion

Description: this example is point-to-point motion control. Use M0 to control servo on / off and set the required parameters including acceleration / deceleration time and the single axis speed unit. When the servo is on, trigger M1 to issue the target position and trigger the absolute position motion. Then, trigger M2 to issue the target position for each axis and trigger the relative position motion.

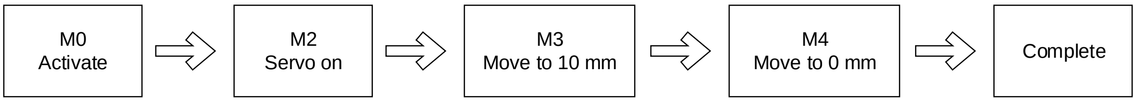
Parameter definition: M0 is the bit for triggering servo on / off; M1 is the bit for controlling the absolute position; M2 is the bit for controlling the relative position.

Example:



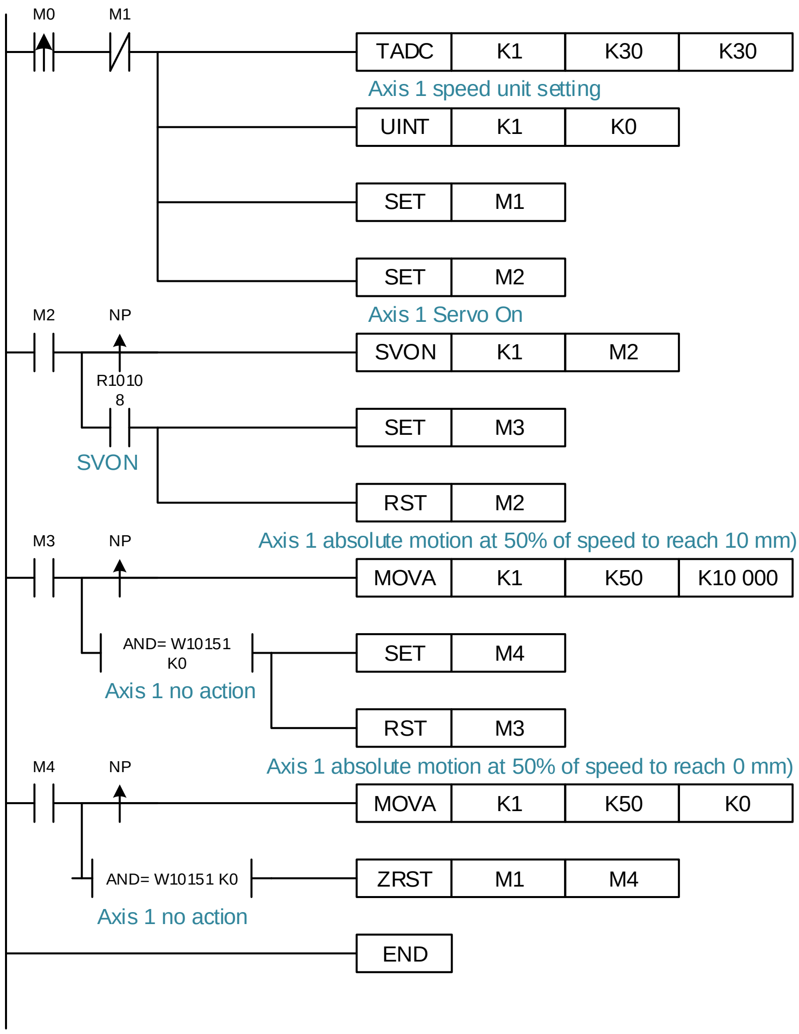
- Single axis point-to-point motion

Description: the sequence of the servo motor point-to-point motion example is as follows:



Parameter definition: M0: bit for triggering the motion; M1: operation flag.

Example:

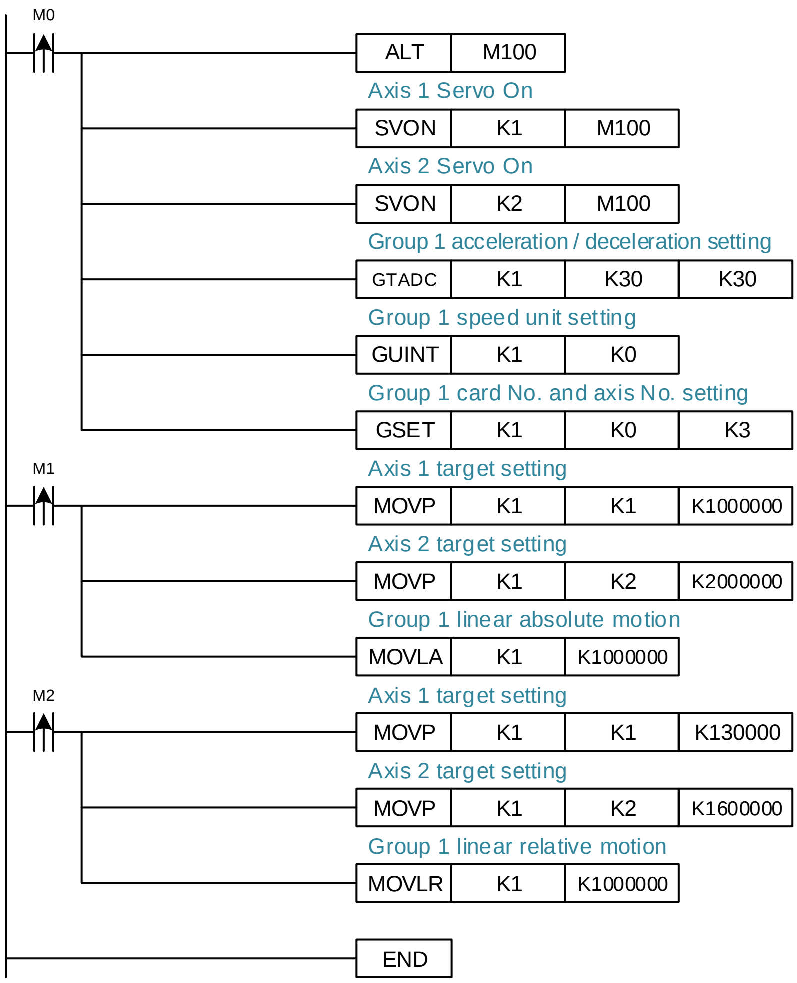


■ Two axes linear interpolation motion

Description: this example is the application of linear interpolation for two axes. Use M0 to control servo on / off and set the required parameters for the interpolation function, including acceleration / deceleration time, group speed unit, and group setting. When the servo is on, trigger M1 to issue the target position of each axis and trigger the absolute linear interpolation motion. Then, trigger M2 to issue the target position for each axis and trigger the relative linear interpolation motion.

Parameter definition: M0 is the bit for triggering servo on / off; M1 is the bit for controlling the group motion to the absolute position by linear interpolation; M2 is the bit for controlling the group motion to the relative distance by linear interpolation.

Example:



■ MPG

Description: in this example, the node number of ASD-DMC-RM64MN that connects to the MPG is set to 9 and the connected DMCNET motion card No. is 0. Set the first IMP axis as the X axis controlled by the MPG and the third IMP axis as the Y axis controlled by MPG with both using quadruple frequency as the input signal. This function can be used to individually control X axis or Y axis of the MPG.

Parameter definition: M0 is the bit to enable / disable the MPG.

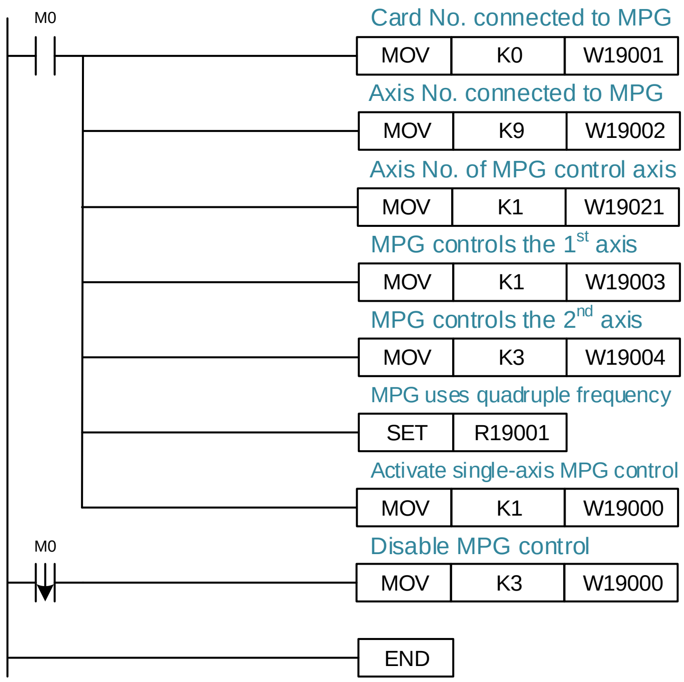
Note: this function needs to be used with specific modules (DMCNET: ASD-DMC-RM64MN; EtherCAT: R1-EC5614D0), and must share the same bus as the control servo axis. If the set axis and motion card are not on the same bus, the error flag R19010 will be on when the MPG is enabled.

The following is a list of MPG related special registers and relay related positions.

Position	Function	Description	Property	Bus
W19000	MPG control code	By switching the register value, you can trigger the MPG functions below:[*1] 0: none 1: enable MPG 2: MPG simulation 3: disable MPG	R/W	D/E
W19001	Card number connected to MPG	1. The card number of the I/O module that the MPG uses to connect to the motion card. 2. This register setting is available only when the MPG is disabled.	R/W	D/E
W19002	Node number connected to MPG	1. The node number set by the MPG using the I/O module. 2. This register setting is available only when the MPG is disabled.	R/W	D/E
W19003	Node number of the 1st servo controlled by the MPG	1. MPG can specify four controlling nodes. The connected I/O module must share the same bus as the control servo; if not, the error flag R19010 will be on. 2. If you do not specify the MPG servo node number, the MPG control is set as disabled by default. 3. This register setting is available only when the MPG is disabled.	R/W	D/E
W19004	Node number of the 2nd servo controlled by the MPG		R/W	D/E
W19005	Node number of the 3rd servo controlled by the MPG		R/W	D/E
W19006	Node number of the 4th servo controlled by the MPG		R/W	D/E
W19007	Node number of the 5th servo controlled by the MPG		R/W	E
W19008	Node number of the 6th servo controlled by the MPG		R/W	E
W19010	MPG running speed	Running speed of the MPG (unit: %).	R	D/E
W19012	MPG current position	Number of the MPG counter (PUU)	R	D/E
W19013			R	D/E
W19020	MPG error code	This is the returned value when an MPG error occurs. This value is 0 when there are no errors.	R	D/E
W19021	Number of axis controlled by the MPG	DMCNET: 1 - 4; EtherCAT: 1 - 6	R/W	D/W
R19001	MPG quadruple frequency control bit	The pulse output per MPG rotating scale. On: output 1 pulse per four MPG rotating scales. Off: output 1 pulse per MPG rotating scale.	R/W	D/E

Position	Function	Description	Property	Bus
R19010	Error flag of MPG control setting	The MPG connected module and the controlled servo axis are not on the same bus.	R	D/E

Example:



Note:

1. When the MPG function is enabled, Pin 9 (P2 X00) and Pin 8 (P2 X01) cannot be used when P3H/P3L (Group 3 GPIO) on the ASD-DMC-RM64MN is set to the MPG mode.
2. The default value of the built-in DMC card number in the PAC is 0; the default value of the built-in EtherCAT card number is 16.

6

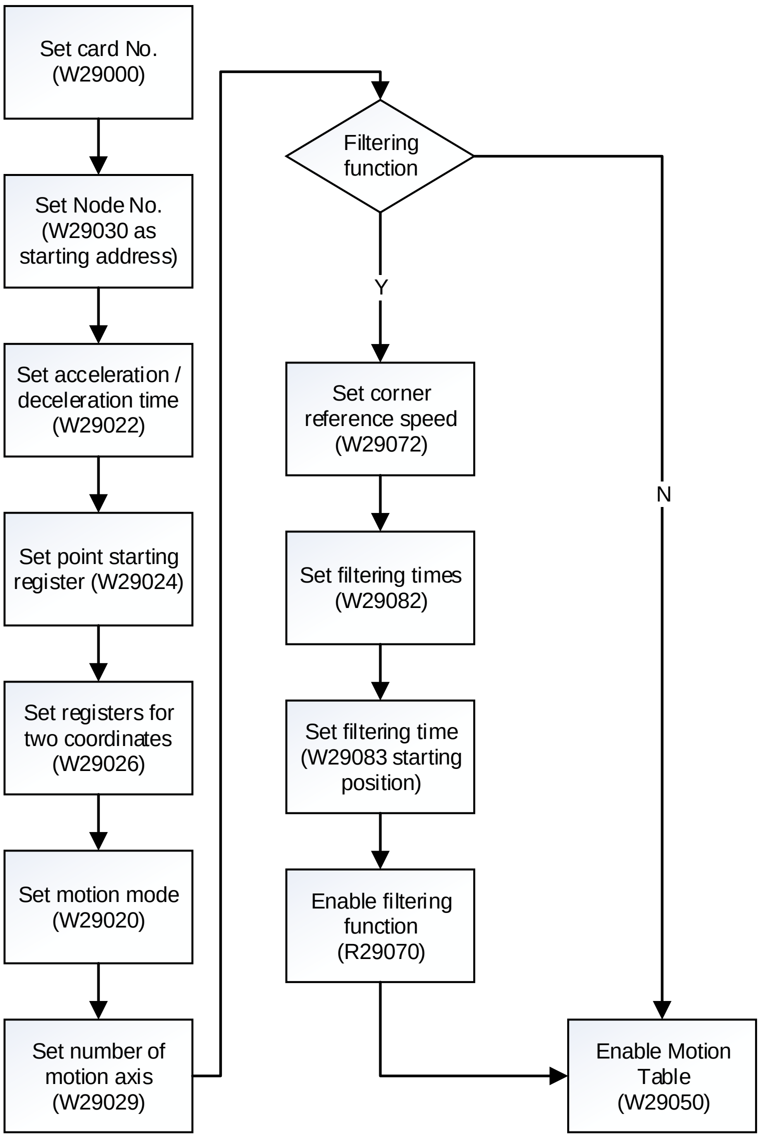
6.1.8 Motion table

■ Features

This function enables the multi-axis and multi-point interpolation motion. The velocity look-ahead function is provided to smooth the speed connection between the points, and the motion process can be operated through the I/O module. It is suitable for conditions such as continuous track processing, processing environment similar to NC path, continuous corner speed (allowing path distortion), etc.

■ Instructions

Motion table supports 6-axis differential motion. Advanced communication motion card can support two sets of motion table motions at the same time. Refer to the following figure for the setting process:



Note: motion card models supported by motion table: PCI-DMC-B01, PCI-DMC-A02, PCI-DMC-B02, PCI-L221-B1, and PCI-L221-P1.

6

- Table parameters

This table describes the parameters related to motion table. The first group of motion table in the motion card is represented by Table1-1 and the second group is represented by Table1-2.

Function	Property	Bus	Table1-1	Table1-2	Description
Motion card number	R/W	D/E	W29000		Motion Table 1-1 and 1-2 card numbers.
Line number in execution	R	D/E	W29010	W29110	Display the current motion position information.
Current motion linear speed	R	D/E	W29012	W29112	Unit: mm/min
	R	D/E	W29013	W29113	
Status	R	D/E	W29015	W29115	0: stop; 1: pause; 2: in operation.
Status error code	R	D/E	W29016	W29116	1: the specified axis is in use. 2: the group axis is not enabled. 3: an error occurred in the command axis. 4: speed setting error. 5: the motion card number designated by the command does not exist.
Digital output node number	R/W	D/E	W29018	W29118	Set the linked output module information in motion.
Digital output port	R/W	D/E	W29020	W29120	
Digital output start bit	R/W	D/E	W29021	W29121	
Total number of points	R/W	D/E	W29022	W29122	Set the read format of the motion points.
Number of starting register D	R/W	D/E	W29024	W29124	
Register point offset	R/W	D/E	W29026	W29126	
Speed operation mode	R/W	D/E	W29028	W29128	0: fixed speed mode. 1: single speed definition mode.
Number of axis in use	R/W	D/E	W29029	W29129	Set the number of axes used for motion.
Node number of the 1st axis	R/W	D/E	W29030	W29130	Set the actual node number corresponding to the motion.
Node number of the 2nd axis	R/W	D/E	W29031	W29131	
Node number of the 3rd axis	R/W	D/E	W29032	W29132	
Node number of the 4th axis	R/W	D/E	W29033	W29133	
Node number of the 5th axis	R/W	D/E	W29034	W29134	
Node number of the 6th axis	R/W	D/E	W29035	W29135	
Control code	R/W	D/E	W29050	W29150	0: no command 1: start processing 2: pause processing 3: continue processing 4: stop processing
Operation speed	R/W	D/E	W29052	W29152	Must set as the speed fixed mode (W29028 is set to 0) for the motion speed control to take effect.
	R/W	D/E	W29053	W29153	
Acceleration time	R/W	D/E	W29054	W29154	TAcc (unit: ms)
	R/W	D/E	W29055	W29155	

Function	Property	Bus	Table1-1	Table1-2	Description
Speed change percentage	R/W	D/E	W29062	W29162	Set the feed rate percentage.
Accumulated length	R/W	D/E	W29070	W29170	Speed continuous control is enabled (R29170 is on) for the functions to take effect.
	R/W	D/E	W29071	W29171	
Corner reference speed	R/W	D/E	W29072	W29172	
	R/W	D/E	W29073	W29173	
Reference length	R/W	D/E	W29074	W29174	
	R/W	D/E	W29075	W29175	
Reference angle	R/W	D/E	W29076	W29176	
	R/W	D/E	W29077	W29177	
Reference speed	R/W	D/E	W29078	W29178	
	R/W	D/E	W29079	W29179	
Reference radius	R/W	D/E	W29080	W29180	
	R/W	D/E	W29081	W29181	
I/O control switch	R/W	D/E	R29018	R29118	ON: enable
Single step mode	R/W	D/E	R29050	R29150	ON: enable single step mode.
Single step triggering	R/W	D/E	R29051	R29151	ON: enable single step triggering.
Speed change control switch	R/W	D/E	R29062	R29162	ON: enable.

■ Filter function

To maintain the corner speed of the path in the actual processing motion, extreme acceleration / deceleration may occur at the turning point of the path. At this time, the machine may generate uneven running speed and vibration. When the filtering function is enabled, smooth path can be achieved and the speed change can be effectively smoothed by sacrificing part of the track precision. Thus, the filtering function needs to be tested with the on-site requirements and machine for the best result. The parameters in this section can also work on Table1-1 and 1-2 at the same time.

Function	Property	Bus	Table1	Description
Corner speed control	R/W	D/E	R29070	ON: enable.
AMF filtering times	R/W	D/E	W29056	The suggested setting value for the filtering times (AMFNum: 0 - 2) is 2. The greater the value, the smoother the speed will be, but the path error of the corner will be larger.
Filtering times	R/W	D/E	W29082	0: disable; 1: 1 time; 2: 2 times.
Node 1_filterTime	R/W	D/E	W29083	Filter time (unit: ms)
Node 2_filterTime	R/W	D/E	W29084	
Node 3_filterTime	R/W	D/E	W29085	
Node 4_filterTime	R/W	D/E	W29086	
Node 5_filterTime	R/W	D/E	W29087	
Node 6_filterTime	R/W	D/E	W29088	
Node 7_filterTime	R/W	D/E	W29089	
Node 8_filterTime	R/W	D/E	W29090	
Node 9_filterTime	R/W	D/E	W29091	
Node A_filterTime	R/W	D/E	W29092	
Node B_filterTime	R/W	D/E	W29093	
Node C_filterTime	R/W	D/E	W29094	

Motion Program Macro (MPM)

7

Motion Program Macro (MPM) is a macro language for IMP motion control. You can create the motion path with MPM and simplify the PLC motion control program.

7.1 List and overview of instructions

This section introduces the macros for IMP motion control. IMP can edit 100 sets of motion program macros. Each set can store 200 lines of instructions and support multiple control commands, including motion application instructions of servo motion control and related application instructions. You can find more details in Sections 7.1.1 and 7.1.2.

No.	Code	Function	No.	Code	Function
1	SETM	Set the auxiliary relay	23	GTADC	Group setting of acceleration / deceleration time
2	RSTM	Reset the auxiliary relay	24	COORD	Single-axis coordinate setting
3	CALLM	Call the auxiliary relay	25	SPD	Single-axis motion in speed mode
4	DELAY	Delay time (unit: ms)	26	TRQ	Single-axis motion in torque mode
5	ADD	Addition	27	SDSTP	Single-axis deceleration to stop
6	SUB	Subtraction	28	ESTP	Single-axis emergency stop
7	MUL	Multiplication	29	GSDSTP	Group deceleration to stop
8	DIV	Division	30	GESTP	Group emergency stop
9	MOV	Move data	31	HOME	Homing
10	FOR	Start of the FOR loop	32	MOVA	Single-axis in absolute motion
11	NEXT	End of the FOR loop	33	MOVR	Single-axis in relative motion
12	IF (bit)	Compare the bit content	34	MOVLA	Linear interpolation in absolute motion
13	IF (word)	Compare the word content	35	MOVLR	Linear interpolation in relative motion
14	ELSE	Else	36	CIRCAA	Arc absolute motion (arc center, angle)
15	ENDIF	End of comparison	37	CIRCAR	Arc relative motion (arc center, angle)
16	DO	Start of DO…LOOP	38	CIREAA	Arc absolute motion (end, angle)
17	LOOP (bit)	End of DO…LOOP (bit)	39	CIREAR	Arc relative motion (end, angle)
18	LOOP (word)	End of DO…LOOP (word)	40	CIRCEA	Arc absolute motion (center, arc end)
19	WHILE (bit)	Start of the WHILE loop (bit)	41	CIRCER	Arc relative motion (center, arc end)
20	WHILE (word)	Start of the WHILE loop (word)	42	HELIXA	Three-axis helical interpolation in absolute motion
21	ENDWHILE	End of the WHILE loop	43	HELIXR	Three-axis helical interpolation in relative motion
22	GUNIT	Motion speed unit setting	44	TAPPING	Tapping

No.	Code	Function	No.	Code	Function
100	FSIN	Sine	106	FLT	Integer->Floating point
101	FCOS	Cosine	107	INT	Floating point->Integer
102	FTAN	Tangent	108	WARA	Write servo parameter
103	FASIN	Arcsine	109	RPARA	Read servo parameter
104	FACOS	Arccosine	110	COORDROTATE	Rotate coordinates
105	FATAN	Arctangent	-	-	-

7

7.1.1 Application instruction

This section will detail each Motion Program Macro (MPM) instruction. The application instructions include flow control and the comparison function of the four arithmetic operation. The following describes the meaning of each field in the instruction table.

How to read the table?

(1) (2) (3) (4)

NO.	-	Instruction	D1			Function	
1							
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	xis No.	Operator
D1	●	-	-	-	-	-	-
Notes when applying operand:						Instruction	
					32-bit	-	
					16-bit	SETM	

(1) D: supports 32-bit instructions.

(2) Operand.

(3) M: auxiliary relay (refer to more details in Chapter 5).

(4) K: constant; decimal value starts with the character K. (This setting value can be an integer or a decimal.) For example, K100 indicates this value is 100 in decimal form; K10.35 indicates this value is 10.35 in decimal form. However, when applying PLC instructions, constants (K) can only be integers; numbers with decimal points are not allowed.
D: data register (refer to more details in Chapter 5).
E: pointer register (reserved).

7

Application instruction

■ **SETM**

NO. 1	-	SETM	D1		Set the auxiliary relay		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
D1	●	-	-	-	-	-	-

Notes when applying operand:

	Instruction
32-bit	-
16-bit	SETM

Description: D1: device position; set operand D1 to ON.

Example: set relay M1000 to ON.

Instruction code	Description
SETM,M1000	Set relay M1000 to ON.

■ **RSTM**

NO. 2	-	RSTM	D1		Reset the auxiliary relay		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
D1	●	-	-	-	-	-	-

Notes when applying operand:

	Instruction
32-bit	-
16-bit	RSTM

Description: D1: device position; set operand D1 to OFF.

Example: reset relay M1000 to OFF.

Instruction code	Description
RSTM,M1000	Reset relay M1000 to OFF.

7

CALLM

NO. 3	-	CALLM	D1		Call the auxiliary relay	
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
D1	●	-	-	-	-	-	-

Notes when applying operand:

	Instruction
32-bit	-
16-bit	CALLM

Description: D1: device position; you can set the specified auxiliary relay to ON with this instruction and wait for the status of the auxiliary relay to be cleared (the auxiliary relay can be cleared by the HMI, PLC, or other MPM) to execute the next instruction. This command can be used with the PLC. The MPM issues the CALLM instruction and when the PLC reads that the corresponding auxiliary relay is ON, the specified action (i.e. function execution or logic operation) is executed. Then the status of this auxiliary relay is cleared and the next instruction is automatically executed.

Example: set relay M1000 to ON. Then, after M1000 is reset to OFF, set M20 to ON.

Instruction code	Description
CALLM,M1000	Set relay M1000 to ON and the relay waits to be cleared.
SETM,20	Set relay M20 to ON.

DELAY

NO. 4	-	DELAY	S1		Delay time		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-

Notes when applying operand:

	Instruction
32-bit	-
16-bit	DELAY

Description: S1: delay time (unit: ms). Execute the next instruction after the delay time (S1).

Example: set the relay M10 to ON. Then, wait for 10 seconds to reset M10 to OFF.

Instruction code	Description
SETM,10	Set relay M10 to ON and the relay waits to be cleared.
DELAY,10000	Delay for 10000 ms.
RSTM,10	Reset relay M10 to OFF.

■ ADD

NO. 5	D	ADD	S1, S2, D1	Addition			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
S2	-	-	●	●	-	-	-
D1	-	-	-	●	-	-	-

Notes when applying operand:		Instruction
For 16-bit instructions, D1 operand occupies 2 consecutive points.	32-bit	DADD
For 32-bit instructions, D1 operand occupies 4 consecutive points.	16-bit	ADD

Description: S1: summand; S2: addend; D1: sum. Add the values of registers S1 and S2 and store the sum in register D1.

Example: set registers D10 and D11 to 15 and 13 respectively. Add the values of D10 and D11, then store the sum in register D20. After the execution, the value of register D20 is 28.

Instruction code	Description
MOV,15,D10	Set register D10 to 15.
MOV,13,D11	Set register D11 to 13.
ADD,D10,D11,D20	Add the values of registers D10 and D11 and store the sum in register D20 (sum = 28).

■ SUB

NO. 6	D	SUB	S1, S2, D1	Subtraction			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
S2	-	-	●	●	-	-	-
D1	-	-	-	●	-	-	-

Notes when applying operand:		Instruction
For 16-bit instructions, D1 operand occupies 2 consecutive points.	32-bit	DSUB
For 32-bit instructions, D1 operand occupies 4 consecutive points.	16-bit	SUB

Description: S1: minuend; S2: subtrahend; D1: difference. Subtract the value of S2 from S1 and store the difference in register D1.

Example: set registers D10 and D11 to 15 and 13 respectively. Subtract the value of D11 from D10 and store the difference in register D20. After the execution, the value of register D20 is 2.

Instruction code	Description
MOV,15,D10	Set register D10 to 15.
MOV,13,D11	Set register D11 to 13.
SUB,D10,D11,D20	Subtract the value of D11 from D10 and store the difference in register D20 (difference = 2).

■ MUL

NO. 7	D	MUL	S1, S2, D1		Multiplication		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
S2	-	-	●	●	-	-	-
D1	-	-	-	●	-	-	-

Notes when applying operand:		Instruction
For 16-bit instructions, D1 operand occupies 2 consecutive points. For 32-bit instructions, D1 operand occupies 4 consecutive points.	32-bit	DMUL
	16-bit	MUL

Description: S1: multiplicand; S2: multiplier; D1: product. Multiply the value of S1 by S2 and store the product in register D.

Example: set registers D10 and D11 to 15 and 13 respectively. Multiply the value of D10 by D11 and store the product in registers D20 and D21. After the execution, the read values of registers D20 and D21 will be 195.

Instruction code	Description
MOV,15,D10	Set register D10 to 15.
MOV,13,D11	Set register D11 to 13.
MUL,D10,D11,D20	Multiply the value of D10 by D11 and store the product in registers D20 and D21.

■ DIV

NO. 8	D	DIV	S1, S2, D1		Division		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
S2	-	-	●	●	-	-	-
D1	-	-	-	●	-	-	-

Notes when applying operand:		Instruction
For 16-bit instructions, D1 operand occupies 2 consecutive points. For 32-bit instructions, D1 operand occupies 4 consecutive points.	32-bit	DDIV
	16-bit	DIV

Description: S1: dividend; S2: divisor; D1: quotient and remainder. Divide the value of S1 by S2 and store the quotient in register D1 and remainder in register D1+1.

Example: set registers D10 and D11 to 15 and 13 respectively. Divide the value of D10 by D11 and store the quotient in register D20. After the execution, the values of D20 is 1 and D21 is 2.

Instruction code	Description
MOV,15,D10	Set register D10 to 15.
MOV,13,D11	Set register D11 to 13.
DIV,D10,D11,D20	Divide the value of D10 by D11 and store the quotient in D20 and remainder in D21.

■ MOV

NO.	D	MOV		S1, D1	Move data		
9							
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	-	●	-	-	-
Notes when applying operand:						Instruction	
					32-bit	DMOV	
					16-bit	MOV	

Description: S1: source; D1: target. Copy the data in source S1 to the target operand D1 and the data in source S1 remains unchanged.

Example: set the initial value of register D10 to 15 and move this value to D13. Then, the value of D13 will be 15.

Instruction code	Description
MOV,15,D10	Set register D10 to 15.
MOV,D10,D13	Copy the value of register D10 to register D13 and the value of D13 will be 15.

■ FOR

NO.	-	FOR		S1	Start of the FOR loop		
10							
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
Notes when applying operand:						Instruction	
					32-bit	-	
					16-bit	FOR	

Description: S1: number of the loop. Repeatedly execute the instructions between FOR and NEXT for S1 time(s), then jump to NEXT to execute the next instruction.

■ NEXT

NO.	-	NEXT	-	End of the FOR loop			
11							
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
	-						

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	NEXT

Description: the NEXT instruction must be used with the FOR instruction. In the following conditions, error may occur:

(1) The NEXT instruction precedes the FOR instruction.

(2) The FOR instruction is executed without the NEXT instruction.

(3) The number of FOR and NEXT instructions is different.

The FOR to NEXT loop can nest for up to 10 layers. If the nesting layers exceed the limit, error may occur and the program cannot execute normally.

Example: execute the loop for 50 times and the value of register D11 will be 50.

Instruction code	Description
MOV,0,D11	Reset the value of register D11 to 0.
FOR,50	Start of the FOR loop.
ADD,D11,1,D11	The value of register D11 increases by 1.
NEXT	End of the FOR loop

■ IF (bit)

NO.	-	IF	S1==S2	Compare the bit content			
12							
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	●	-	-	-	-	-	-
S2	-	-	-	-	-	-	●

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	IF

Description: S1: device position; S2: condition (ON or OFF).

If relay S1 has fulfilled the condition of S2, execute the next instruction; otherwise, jump to the ELSE instruction. If there is no corresponding ELSE instruction, jump to the ENDIF instruction.

■ IF (word)

NO. 13	-	IF	S1, S2, S3		Compare the word content		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
S2	-	-	-	-	-	-	●
S3	-	-	●	●	-	-	-
Notes when applying operand:						Instruction	
					32-bit	-	
					16-bit	IF	

Description: S1: device position; S2: operator (==, <>, <=, >=, <, >); S3: device position.

If S2 is ==, when register S1 fulfills the condition of S3, the next instruction will be executed.

If S2 is <=, when the value of register S1 is smaller than or equal to the condition of S3, the next instruction will be executed.

If S2 is >=, when the value of register S1 is greater than or equal to the condition of S3, the next instruction will be executed.

If S2 is <, when the value of register S1 is smaller than the condition of S3, the next instruction will be executed.

If S2 is >, when the value of register S1 is greater than the condition of S3, the next instruction will be executed.

If the condition is not fulfilled, jump to the ELSE instruction; if there is no corresponding ELSE instruction, jump to the ENDIF instruction.

■ ELSE

NO. 14	-	ELSE	-		Else		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
	-						
Notes when applying operand:						Instruction	
					32-bit	-	
					16-bit	ELSE	

Description: when the IF statement is false, you can use the ELSE instruction to execute other instructions. However, both IF and ENDIF instructions must be applied in this circumstance.

7

■ ENDIF

NO. 15	-	ENDIF	-		End of comparison	
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
	-						

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	ENDIF

Description: this is applied with the IF and ELSE instructions.

Example: if the value of register D10 is 10, set relay M0 to ON; if not, reset relay M0 to OFF.

Instruction code	Description
IF,D10,==,K10	If the value of register D10 is 10, execute the next instruction; otherwise, jump to the ELSE instruction.
SETM,0	Set relay M0 to ON.
ELSE	Else.
RSTM,0	Reset relay M0 to OFF.
ENDIF	End the IF instruction.

■ DO

NO. 16	-	DO	-		Start of DO…LOOP		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
	-						

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	DO

Description: this instruction must be applied with the LOOP instruction and is inserted before LOOP.

■ LOOP (bit)

NO. 17	-	LOOP	S1==S2		End of DO…LOOP (bit)		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	●	-	-	-	-	-	-
S2	-	-	-	-	-	-	●

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	LOOP

Description: S1: device position; S2: condition (ON or OFF).

If relay S1 has fulfilled the condition of S2, execute the corresponding DO instruction; otherwise, execute the next instruction.

■ LOOP (word)

NO. 18	-	LOOP	S1, S2, S3		End of DO…LOOP (word)		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
S2	-	-	-	-	-	-	●
S3	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	LOOP

Description: S1: condition 1; S2: operator (==, <>, <=, >=, <, >); S3: condition 2.

If S2 is ==, when register S1 fulfills the condition of S3, the corresponding DO instruction will be executed.

If S2 is <=, when the value of register S1 is smaller than or equal to the condition of S3, the corresponding DO instruction will be executed.

If S2 is >=, when the value of register S1 is greater than or equal to the condition of S3, the corresponding DO instruction will be executed.

If S2 is <, when the value of register S1 is smaller than the condition of S3, the corresponding DO instruction will be executed.

If S2 is >, when the value of register S1 is greater than the condition of S3, the corresponding DO instruction will be executed.

If the statement is false, the next instruction is executed.

Example: if the value of register D10 plus 1 is smaller than 10, carry on executing the loop until the value is greater than 10, then the value of register D10 is cleared to 0.

Instruction code	Description
DO	Start of DO…LOOP
ADD,D10,K1,D10	The value of register D10 increases by 1.
LOOP,D10,<,K10	If the value of register D10 is smaller than 10, carry on executing the loop; if greater than 10, exit the loop.
MOV,K0,D10	Reset the value of register D10 to 0.

■ **WHILE (bit)**

NO. 19	-	WHILE	S1==S2	Start of the WHILE loop (bit)		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	-	-	-	-
S2	-	-	-	-	-	-	●

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	WHILE

Description: S1: device position; S2: condition (ON or OFF).

If relay S1 has fulfilled the condition of S2, execute the next instruction; otherwise, execute the corresponding ENDWHILE instruction.

■ **WHILE (word)**

NO. 20	-	WHILE	S1, S2, S3	Start of the WHILE loop (word)			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
S2	-	-	-	-	-	-	●
S3	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	WHILE

Description: S1: condition 1; S2: operator (==, <>, <=, >=, <, >); S3: condition 2.

If S2 is ==, when register S1 fulfills the condition of S3, the next instruction will be executed.

If S2 is <=, when the value of register S1 is smaller than or equal to the condition of S3, the next instruction will be executed.

If S2 is >=, when the value of register S1 is greater than or equal to the condition of S3, the next instruction will be executed.

If S2 is <, when the value of register S1 is smaller than the condition of S3, the next instruction will be executed.

If S2 is >, when the value of register S1 is greater than the condition of S3, the next instruction will be executed.

If the statement is false, the corresponding ENDWHILE instruction will be executed.

■ ENDWHILE

NO. 21	-	ENDWHILE	-		End of the WHILE loop		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
	-						

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	ENDWHILE

Description: this instruction must be applied with the WHILE instruction.

Example: if the value of register D10 is smaller than or equal to 100, execute the instructions in the loop. Then, the value of register D10 will increase by 1 each time the loop is executed until the value is over 100. The loop ends when the register value is over 100 and the final value of register D10 will be 101.

Instruction code	Description
MOV,0,D10	Set register D10 to 0.
WHILE,D10,<=,100	If the value of register D10 is smaller than or equal to 100, execute the content of the loop.
ADD,D10,1,D10	The value of register D10 increases by 1.
ENDEHILE	End of the loop.

■ FSIN

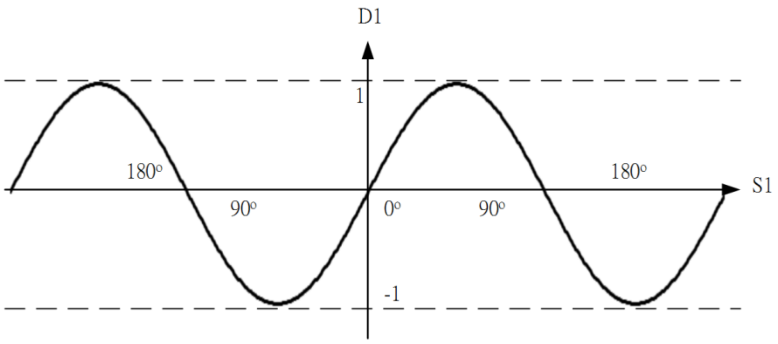
NO. 21	-	FSIN	S1 D1		Sine		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	●	●	-	-	-

Notes when applying operand:
S1 operand occupies 2 consecutive points.
D1 operand occupies 2 consecutive points.

	Instruction
32-bit	FSIN
16-bit	-

Description: S1: specified source value (floating point number); D1: result acquired from SIN value (floating point number).

Obtain SIN value from the source angle specified by S1 and save the value in the register specified by D1. The figure below shows the relation between the angle and result:



S1: angle data; R: result (SIN value)

Example: set the values of registers D11 and D10 to 45.000 (floating point format), perform SIN operation on the contents of register D10, and store the result in register D12. After the execution, the values of registers D13 and D12 are 0.707 (floating point format).

Instruction code	Description
FLT,45,D10	Set registers D11 and D10 to 45 (floating point format).
FSIN,D10,D12	Perform SIN operation on the contents of D11 and D10, and store the result of 0.707 in D13 and D12.

■ FCOS

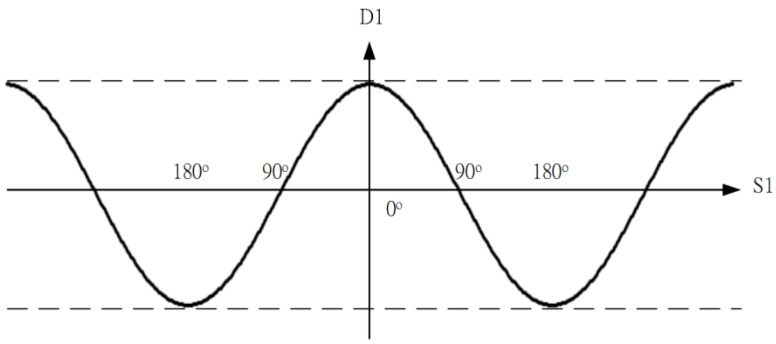
NO. 21	-	FCOS	S1 D1		Cosine		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
S1 operand occupies 2 consecutive points.	32-bit	FCOS
D1 operand occupies 2 consecutive points.	16-bit	-

Description: S1: specified source value (floating point number); D1: result acquired from COS value (floating point number).

Obtain COS value from the source angle specified by S1 and save the value in the register specified by D1.

The figure below shows the relation between the angle and result:



Example: set the values of registers D11 and D10 to 45.000 (floating point format), perform COS operation on the contents of register D10, and store the result in register D12. After the execution, the values of registers D13 and D12 are 0.707 (floating point format).

Instruction code	Description
FLT,45,D10	Set registers D11 and D10 to 45 (floating point format).
FCOS,D10,D12	Perform COS operation on the contents of D11 and D10, and store the result of 0.707 in D13 and D12.

■ **FTAN**

NO.	-	FTAN	S1 D1	Tangent
21				

	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	●	●	-	-	-

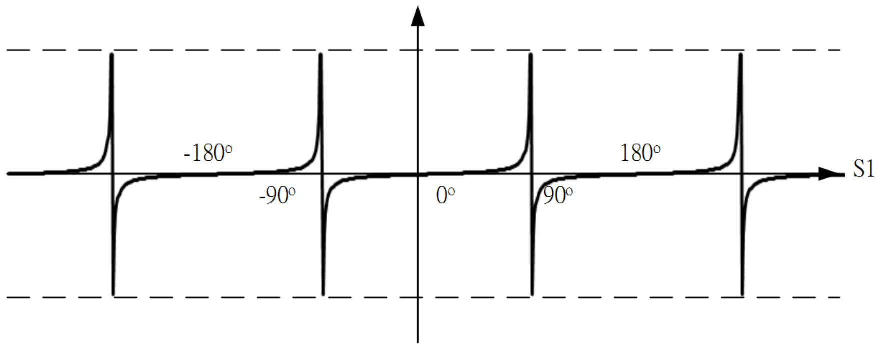
Notes when applying operand:
S1 operand occupies 2 consecutive points.
D1 operand occupies 2 consecutive points.

	Instruction
32-bit	FTAN
16-bit	-

Description: S1: specified source value (floating point number); D1: result acquired from TAN value (floating point number).

Obtain TAN value from the source angle specified by S1 and save the value in the register specified by D1.

The figure below shows the relation between the angle and result:



Example: set the values of registers D11 and D10 to 45.000 (floating point format), perform TAN operation on the contents of register D10, and store the result in register D12. After the execution, the values of registers D13 and D12 are 1.0 (floating point format).

Instruction code	Description
FLT,45,D10	Set registers D11 and D10 to 45 (floating point format).
FTAN,D10,D12	Perform TAN operation on the contents of D11 and D10, and store the result of 1.0 in D13 and D12.

■ FASIN

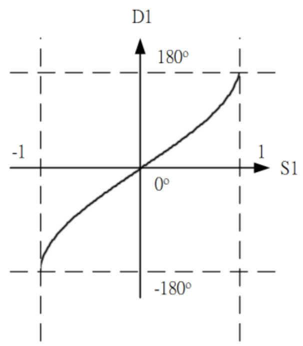
NO. 100	-	FASIN	S1 D1		Arcsine		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
S1 operand occupies 2 consecutive points.	32-bit	FASIN
D1 operand occupies 2 consecutive points.	16-bit	-

Description: S1: source of specified sine value (floating point number); D1: radian result acquired from ASIN value (floating point number).

Obtain Sin^{-1} value from the source specified by S1 and save the value in the register specified by D1.

The figure below shows the relation between the angle and result:



Example: perform SIN^{-1} operation on the value of 0.5 and store the result in register D12. After the execution, the values of registers D13 and D12 are 30.0 (floating point format).

Instruction code	Description
FASIN,0.5,D12	Perform SIN^{-1} operation on 0.5 and store the result of 30.0 in D13 and D12.

■ FACOS

NO. 101	-	FACOS	S1 D1	Arccosine

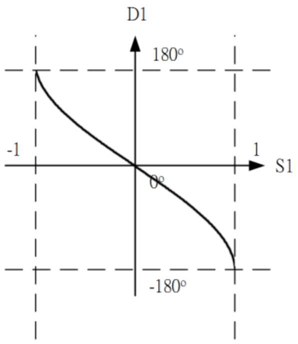
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
S1 operand occupies 2 consecutive points.	32-bit	FACOS
D1 operand occupies 2 consecutive points.	16-bit	-

Description: S: source of specified cosine value (floating point number); D1: radian result acquired from ACOS value (floating point number).

Obtain Cos^{-1} value from the source specified by S1 and save the value in the register specified by D1.

The figure below shows the relation between the angle and result:



Example: perform COS^{-1} operation on the value of 0.5 and store the result in register D12. After the execution, the values of registers D13 and D12 are 60.0 (floating point format).

Instruction code	Description
FACOS,0.5,D12	Perform COS^{-1} operation on 0.5 and store the result of 60.0 in D13 and D12.

■ FATAN

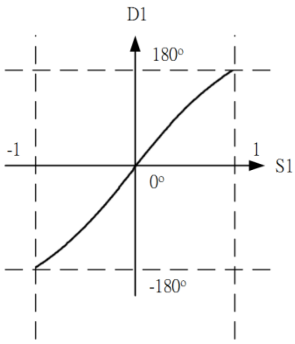
NO. 102	-	FATAN	S1 D1		Arctangent		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
S1 operand occupies 2 consecutive points. D1 operand occupies 2 consecutive points.	32-bit	FATAN
	16-bit	-

Description: S1: source of specified tangent value (floating point number); D1: radian result acquired from ATAN value (floating point number).

Obtain Tan^{-1} value from the source specified by S1 and save the value in the register specified by D1.

The figure below shows the relation between the angle and result:



Example: perform TAN^{-1} operation on the value of 0.5 and store the result in register D12. After the execution, the values of registers D13 and D12 are 26.57 (floating point format).

Instruction code	Description
FATAN,0.5,D12	Perform TAN^{-1} operation on 0.5 and store the result of 26.57 in D13 and D12.

■ FLT

NO. 103	D	FLT		S1 D1		Convert to decimal value in binary format	
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
32-bit instruction: S1 operand occupies 2 consecutive points.	32-bit	DFLT
16-bit and 32-bit instructions: D1 operand occupies 2 consecutive points.	16-bit	FLT

Description: S1: specified source value (integer); D1: result (floating point number).

Example: set D10 to the constant value of 45, convert the content of register D10 to floating point format, and store it in registers D13 and D12. After the execution, the values of registers D13 and D12 are 45.0 (floating point format).

Instruction code	Description
MOV,45,D10	Set register D10 to 45 (constant).
FLT,D10,D12	Read the value of register D10, convert it to floating point format, and store the result of 45.0 (floating point format) in D13 and D12.

■ INT

NO. 104	D	INT		S1 D1		Convert decimal value in binary format to integer	
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
16-bit and 32-bit instructions: S1 operand occupies 2 consecutive points.	32-bit	DINT
32-bit instruction: D1 operand occupies 2 consecutive points.	16-bit	INT

Description: S1: specified source value (floating point number); D1: result (integer).

Example: convert the value of 1.23 to constant format and store it in register D12. After the execution, the value of register D12 is 1 (floating point format).

Instruction code	Description
INT,1.23,D12	Convert the value of 1.23 to constant format and store it in register D12. After the execution, the value of register D12 is 1 (constant format).

■ **WPARA**

NO. 105	-	WPARA	S1 S2 S3 S4	Write servo parameter			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	WPARA

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis; S2: parameter group; S3: parameter offset value; S4: write value.

Example: write the value of 100 to P5-03 of the Z setting axis.

Instruction code	Description
WPARA,X,5,3,100	Write 100 to servo P5-03.

■ **RPARA**

NO. 106	-	RPARA	S1 S2 S3 D1	Read servo parameter			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
D1	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	RPARA

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis; S2: parameter group; S3: parameter offset value; D1: read value.

Example: after reading the P5-03 parameters of the X-axis servo, store the values in register D10.

Instruction code	Description
RPARA,X,5,3,100	Read servo P5-03 to D10.

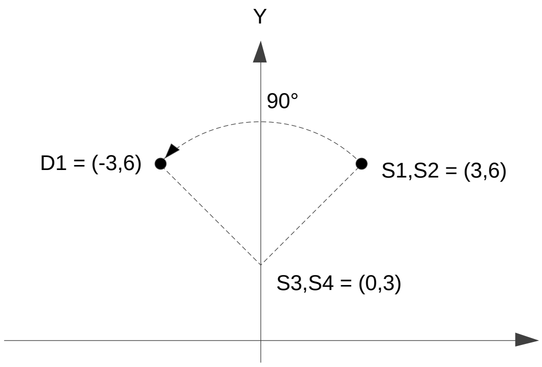
7

■ CoordRotate

NO. 107	-	CoordRotate	S1 S2 S3 S4 S5 D1		Rotate coordinates		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●	-	-	-
S2	-	-	●	●	-	-	-
S3	-	-	●	●	-	-	-
S4	-	-	●	●	-	-	-
S5	-	-	●	●	-	-	-
D1	-	-	-	●	-	-	-

Notes when applying operand:		Instruction
S1, S2, S3, S4, and S5 operands each occupy 2 consecutive floating point numbers.	32-bit	CoordRotate
D1 operand occupies 4 consecutive floating point numbers.	16-bit	-

Description: S1: X-axis coordinate; S2: Y-axis coordinate; S3: X-axis rotation center; S4: Y-axis rotation center; D5: rotation angle; D1 target value (D1+1, D1: X-axis coordinate after rotation; D1+3, D1+2: Y-axis coordinate after rotation).



Example: rotate the coordinate (3, 6) by 90 degrees based on (0, 3) as the center and store the result (-3, 6) in the four registers starting with D1.

Instruction code	Description
CoordRotate,0,3,3,6,90,D1	Store the coordinate rotation result in D1.

7.1.2 Motion application instruction

This section will detail the motion application instructions of each MPM. The motion application instructions are servo motion control related functions, including fixed torque, single-axis motion, and multi-axis interpolation instructions. In the case where the servo motions do not conflict with each other, the IMP can run the MPMs in a time-division multiplexing manner. The following describes the meaning of each field in the instruction table.

How to read the table?

(1) (2)

NO. / 1	-	Instruction	D1	Function

	Bit device		(3) Word device			(4) Code and symbol	
	M	-	K	D	E	xis No.	Operator
D1	●	-	-	-	-	-	-

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	SETM

(1) D: supports 32-bit instructions.

(2) Operand.

(3) M: auxiliary relay (refer to more details in Chapter 5).

(4) K: constant; decimal value starts with the character K. (This setting value can be an integer or a decimal.) For example, K100 indicates this value is 100 in decimal form; K10.35 indicates this value is 10.35 in decimal form. However, when applying PLC instructions, constants (K) can only be integers; numbers with decimal points are not allowed.
D: data register (refer to more details in Chapter 5).
E: pointer register (reserved).

7

■ GUNIT

NO. 22	-	GUNIT	S1	Setting of motion speed unit			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	-	-	-	-
Notes when applying operand:						Instruction	
					32-bit	-	
					16-bit	GUNIT	

Description: S1: motion speed unit (0: puu/s; 1: percentage; 2: mm). Set the motion speed unit for the MPM group and execute the next instruction. The motion speed unit set by this instruction is only valid when this MPM executes the motion instruction. Please note that this instruction differs from the defined range of the GUNIT and UNIT instructions of the PLC.

Example: set the motion speed unit of the MPM to puu/s and move to the absolute position 10 mm of the X-axis with the speed of 1000000 puu/s.

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
MOVA,X,1000000,10	Move to the absolute position of 10 mm with the speed of 1000000 puu/s.

■ GTADC

NO. 23	-	GTADC	S1, S2	Group setting of acceleration / deceleration time			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	●	●	-	-	-
Notes when applying operand: If S1 and S2 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the GTADC instruction reads 35.997.						Instruction	
					32-bit	GTADC	
					16-bit	-	

Description: S1: acceleration time (unit: s), the duration to accelerate to the maximum speed; S2: deceleration time (unit: s), the duration to decelerate from the maximum speed to stop. Set the acceleration / deceleration times for the MPM group and execute the next instruction. The acceleration / deceleration times set by this instruction is only valid when this MPM executes the interpolation motion instruction. Please note that this instruction differs from the defined range of the GTADC and TADC instructions of the PLC.

Note: for details about setting the acceleration / deceleration times, refer to Chapter 6.

Example: set the motion speed unit of the MPM to puu/s. The servo motor accelerates to 1000000 puu/s, then it starts to decelerate before reaching the absolute position 10 mm of the X-axis. (The motor accelerates to the system set maximum speed in 0.3 seconds and decelerates from the system set maximum speed to stop in 0.4 seconds.)

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
GTADC,0.3,0.4	Set the group acceleration and deceleration times to 0.3 seconds and 0.4 seconds respectively.
MOVA,X,1000000,10	Move to the absolute position of 10 mm with the speed of 1000000 puu/s.

■ COORD

NO. 24	-	COORD		S1, S2		Single-axis coordinate setting	
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
If S2 uses register D, the last three digits are decimals. For example, if the value of D100 is K35997, the COORD instruction reads 35.997.	32-bit	COORD
	16-bit	-

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis; S2: coordinates (unit: mm). Change the servo axis coordinate specified by S1 into the coordinate specified by S2, and then execute the next instruction.

Example: set the motion speed unit of the MPM to puu/s, move to the absolute position 10 mm of the X-axis with the speed of 1000000 puu/s, then change the target position to 20 mm.

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
MOVA,X,1000000,10	Move to the absolute position of 10 mm with the speed of 1000000 puu/s.
COORD,X,20	Set the target position of X-axis to 20 mm.

7

SPD

NO.	-	SPD	S1, S2	Single-axis motion in speed mode		
25						
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
If S2 uses register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the SPD instruction reads 35.997.	32-bit	SPD
	16-bit	-

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis; S2: motion speed (unit: rpm). Trigger Axis S1 to operate at the speed specified by S2 and execute the next instruction. This instruction automatically stops when the execution of MPM group is completed.

Example: set the motion speed unit for MPM to puu/s. Axis X runs at the fixed speed of 100 rpm for 10 seconds, then this MPM is completed and this axis stops running.

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
SPD,X,100	Axis X runs at the fixed speed of 100 rpm.
DELAY,10000	Delay for 10000 ms (10 s).

TRQ

NO.	-	TRQ	S1, S2	Single-axis motion in torque mode		
26						
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	TRQ

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis; S2: torque (unit: ‰). Trigger Axis S1 to operate at the torque specified by S2 and execute the next instruction. This instruction automatically stops when the execution of MPM group is completed.

Example: set the motion speed unit for MPM to puu/s. Axis X runs with the servo motor's maximum torque of 150‰ for 10 seconds, then this MPM is completed and this axis stops running.

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
TRQ,X,150	Axis X runs with the servo motor's maximum torque of 150‰.
DELAY,10000	Delay for 10000 ms.

■ SDSTP

NO.	-	SDSTP	S1		Single-axis deceleration to stop		
27							
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
Notes when applying operand:						Instruction	
					32-bit	-	
					16-bit	SDSTP	

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis. The servo axis specified by S1 decelerates at the deceleration speed set by the GTADC instruction, and executes the next instruction.

Example: set the motion speed unit for MPM to puu/s. Axis X runs at the fixed speed of 100 rpm for 10 seconds, and decelerates to stop in 0.4 seconds. Then, it moves to the the absolute position 10 mm.

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
GTADC,0.3,0.4	Set the group acceleration and deceleration times to 0.3 seconds and 0.4 seconds respectively.
SPD,X,100	Axis X runs at the fixed speed of 100 rpm.
DELAY,10000	Delay for 10000 ms.
SDSTP,X	Axis X decelerates to stop in 0.4 seconds.
MOVA,X,100000,10	Axis X moves to the absolute position of 10 mm with the speed of 100000 puu/s.

■ ESTP

NO.	-	ESTP	S1		Single-axis emergency stop		
28							
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
Notes when applying operand:						Instruction	
					32-bit	-	
					16-bit	ESTP	

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis. The servo axis specified by S1 decelerates to stop at the maximum deceleration speed, and executes the next instruction.

7

Example: set the motion speed unit for MPM to puu/s. Axis X runs at the fixed speed of 100 rpm for 10 seconds, and decelerates to stop at the maximum deceleration speed. Then, it moves to the the absolute position 10 mm.

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
GTADC,0.3,0.4	Set the group acceleration and deceleration times to 0.3 seconds and 0.4 seconds respectively.
SPD,X,100	Axis X runs at the fixed speed of 100 rpm.
DELAY,10000	Delay for 10000 ms.
ESTP,X	Axis X decelerates to stop at the maximum deceleration speed.
MOVA,X,100000,10	Axis X moves to the absolute position of 10 mm with the speed of 100000 puu/s.

■ **GSDSTP**

NO.	-	GSDSTP		-		Group deceleration to stop	
29							
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
	-						

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	GSDSTP

Description: the servo axes specified by MPM decelerate to stop within the deceleration time set by the GTADC instruction, and execute the next instruction.

Example: set the motion speed unit for MPM to puu/s. Axis X and Axis Y run at the fixed speed of 100 rpm for 10 seconds, and the servo axes specified by MPM decelerate to stop in 0.4 seconds.

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
GTADC,0.3,0.4	Set the group acceleration and deceleration times to 0.3 seconds and 0.4 seconds respectively.
SPD,X,100,Y,100	Axis X and Axis Y run at the fixed speed of 100 rpm.
DELAY,10000	Delay for 10000 ms.
GSDSTP	All axes specified by MPM decelerate to stop in 0.4 seconds.

■ GESTP

NO.	-	GESTP	-	Group emergency stop		
30						
	Bit device		Word device		Code and symbol	
	M	-	K	D	E	Axis No.	Operator
	-						

Notes when applying operand:		Instruction
	32-bit	-
	16-bit	GESTP

Description: the servo axes specified by MPM decelerate to stop at the maximum deceleration speed, and execute the next instruction.

Example: set the motion speed unit for MPM to puu/s. Axis X and Axis Y run at the fixed speed of 100 rpm for 10 seconds, and the servo axes specified by MPM decelerate to stop at the maximum deceleration speed.

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
GTADC,0.3,0.4	Set the group acceleration and deceleration times to 0.3 seconds and 0.4 seconds respectively.
SPD,X,100,Y,100	Axis X and Axis Y run at the fixed speed of 100 rpm.
DELAY,10000	Delay for 10000 ms.
GESTP	All axes specified by MPM decelerate to stop at the maximum deceleration speed.

7

■ **HOME**

NO. 31	-	HOME	S1, S2, S3, S4, S5		Homing			
	Bit device		Word device			Code and symbol		
	M	-	K	D	E	Axis No.	Operator	
S1	-	-	-	-	-	●	-	
S2	-	-	●	●	-	-	-	
S3	-	-	●	●(DW)	-	-	-	
S4	-	-	●	●(DW)	-	-	-	
S5	-	-	●	●(DW)	-	-	-	

Notes when applying operand:
If S3, S4, and S5 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the HOME instruction reads 35.997.

	Instruction
32-bit	HOME
16-bit	-

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis; S2: homing mode (for details about the homing instruction, refer to Chapter 6); S3: first homing speed (unit: rpm); S4: second homing speed (unit: rpm); S5: offset value (unit: mm). When executing the HOME instruction, the assigned servo axis S1 looks for the origin with the homing method specified by S2. And changes the servo axis current coordinates to the coordinates specified by S5, and then executes the next instruction.

Note: changing the speed or executing the pause instruction during the homing process will cause this MPM to end unexpectedly.

Example: Axis X executes the homing process with mode 34, and when completed, it moves to the absolute position 20 mm at 50% of the maximum speed.

Note: Homing mode 34 is to look for the encoder Z phase in reverse direction.

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
HOME,X,34,100,200,20	Axis X executes the homing process with mode 34, and the first speed is 100 rpm and the second speed is 200 rpm. After homing is completed, set this position as the absolute position 20 mm.
MOVA,X,50,20	Axis X moves to the absolute position 20 mm at 50% of the maximum speed.

■ MOVA

NO. 32	-	MOVA	S1, S2, S3	Single-axis in absolute motion			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	●	●(DW)	-	-	-
S3	-	-	●	●(DW)	-	-	-

Notes when applying operand:		Instruction
If S2 and S3 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the MOVA instruction reads 35.997.	32-bit	MOVA
	16-bit	-

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis; S2: motion speed (unit: default is puu/s which is the same as the PLC setting); S3: target position (unit: mm). When executing the MOVA instruction, the servo motor specified by S1 runs at the speed set by S2, and stops when reaching the coordinates specified by S3, then executes the next instruction.

Example: set the motion speed unit to percentage. Axis X and Axis Y start moving simultaneously, and when reaching the absolute positions 200 mm and 300 mm respectively, Axis X returns to the absolute position 0 mm.

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
MOVA,X,50,200,Y,60,300	Axis X moves to the absolute position 200 mm at 50% of the maximum speed and Axis Y moves to the absolute position 300 mm at 60% of the maximum speed.
MOVA,X,50,0	Axis X moves to the absolute position 0 mm at 50% of the maximum speed.

■ MOVR

NO.							
33	-	MOVR	S1, S2, S3		Single-axis in relative motion		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	●	●(DW)	-	-	-
S3	-	-	●	●(DW)	-	-	-

Notes when applying operand:
If S2 and S3 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the MOVR instruction reads 35.997.

	Instruction
32-bit	MOVR
16-bit	-

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis; S2: motion speed (unit: default is puu/s which is the same as the PLC setting); S3: incremental distance (unit: mm). When executing the MOVR instruction, the servo axis specified by S1 runs at the speed set by S2, and stops when reaching the coordinates of (current position + S3), then executes the next instruction.

Example: set the motion speed unit to percentage. Axis X and Axis Y start moving simultaneously, and reach the positions of (Axis X current position + 200 mm) and (Axis Y current position + 300 mm) respectively.

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
MOVR,X,50,200,Y,60,300	Axis X moves 200 mm at 50% of the maximum speed and Axis Y moves 300 mm at 60% of the maximum speed.

■ MOVLA

NO.							
34	-	MOVLA	S1, S2, S3		Linear interpolation in absolute motion		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●(DW)	-	-	-
S2	-	-	-	-	-	●	-
S3	-	-	●	●(DW)	-	-	-

Notes when applying operand:
If S1 and S3 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the MOVLA instruction reads 35.997.

	Instruction
32-bit	MOVLA
16-bit	-

Description: S1: maximum speed (unit: default is puu/s; works with the GUNIT instruction); S2: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis; S3: target position (unit: mm). The servo axis specified by S2 moves to the target position S3 at the maximum speed set by S1.

Example: set the motion speed unit to percentage. Axis X and Axis Y conduct interpolation at 50% of the maximum speed, and reach the absolute positions of 200 mm and 300 mm respectively at the same time. Then, this MPM ends.

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
MOVLA,50,X,200,Y,300	Axis X and Axis Y conduct interpolation at 50% of the maximum speed, and reach the absolute positions of 200 mm and 300 mm respectively at the same time.

- **MOVLR**

NO.	-	MOVLR	S1, S2, S3		Linear interpolation in relative motion		
35							
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●(DW)	-	-	-
S2	-	-	-	-	-	●	-
S3	-	-	●	●(DW)	-	-	-

Notes when applying operand: If S1 and S3 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the MOVLR instruction reads 35.997.		Instruction
	32-bit	MOVLR
	16-bit	-

Description: S1: maximum speed (unit: default is puu/s; you can use the GUNIT instruction to select the speed unit); S2: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis; S3: incremental distance (unit: mm). The servo axis specified by S2 moves to the target position of (current position + S3) at the maximum speed set by S1.

Example: set the motion speed unit to percentage. Axis X and Axis Y conduct interpolation at 50% of the maximum speed, and reach the positions of (Axis X current position + 200 mm) and (Axis Y current position + 300 mm) respectively at the same time. Then, this MPM ends.

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
MOVLR,50,X,200,Y,300	Axis X and Axis Y conduct interpolation at 50% of the maximum speed, and move 200 mm and 300 mm respectively at the same time.

7

■ CIRCAA

NO. 36	-	CIRCAA	S1, S2, S3, S4, S5, S6	Arc absolute motion (center, angle)

	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	-	-	-	●	-
S3	-	-	●	●(DW)	-	-	-
S4	-	-	●	●(DW)	-	-	-
S5	-	-	●	●	-	-	-
S6	-	-	●	●(DW)	-	-	-

Notes when applying operand:		Instruction
If S3, S4, and S6 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the CIRCAA instruction reads 35.997.	32-bit	CIRCAA
	16-bit	-

Description: S1: X: 1^{st} axis, Y: 2^{nd} axis, Z: 3^{rd} axis, U: 4^{th} axis, V: 5^{th} axis, W: 6^{th} axis.

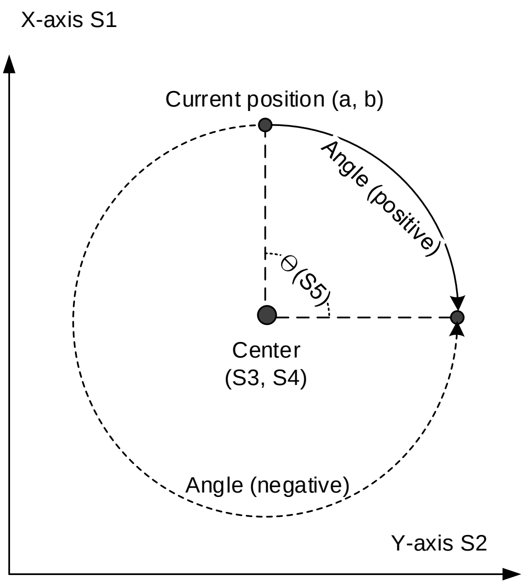
S2: X: 1^{st} axis, Y: 2^{nd} axis, Z: 3^{rd} axis, U: 4^{th} axis, V: 5^{th} axis, W: 6^{th} axis.

S3: X-coordinate of the center (unit: mm).

S4: Y-coordinate of the center (unit: mm).

S5: angle (unit: degree).

S6: maximum speed (unit: default is puu/s; you can use the GUNIT instruction to select the speed unit).



7

■ CIRCAR

NO. 37	-	CIRCAR	S1, S2, S3, S4, S5, S6	Arc relative motion (center, angle)			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	-	-	-	●	-
S3	-	-	●	●(DW)	-	-	-
S4	-	-	●	●(DW)	-	-	-
S5	-	-	●	●	-	-	-
S6	-	-	●	●(DW)	-	-	-

Notes when applying operand:		Instruction
If S3, S4, and S6 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the CIRCAR instruction reads 35.997.	32-bit	CIRCAR
	16-bit	-

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis.

S2: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis.

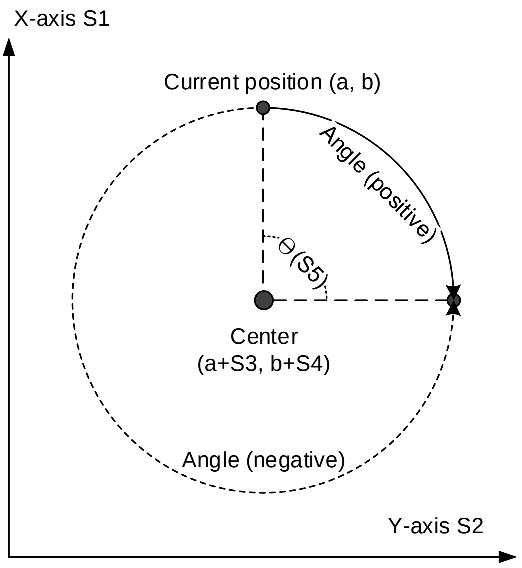
S3: relative distance to the center (X-axis, unit: mm).

S4: relative distance to the center (Y-axis, unit: mm).

S5: angle (unit: degree).

S6: maximum speed (unit: default is puu/s; you can use the GUNIT instruction to select the speed unit).

S1 is the X servo axis and S2 is the Y servo axis. The X-coordinate of the center is its current coordinate + S3 and the Y-coordinate is its current coordinate + S4. The angle between the current position and the arc end is S5. Axis X and Axis Y move at the maximum speed set by S6.



■ CIREAA

NO. 38	-	CIREAA	S1, S2, S3, S4, S5, S6		Arc absolute motion (end, angle)		
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	-	-	-	●	-
S3	-	-	●	●(DW)	-	-	-
S4	-	-	●	●(DW)	-	-	-
S5	-	-	●	●	-	-	-
S6	-	-	●	●(DW)	-	-	-

Notes when applying operand:		Instruction
If S3, S4, and S6 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the CIREAA instruction reads 35.997.	32-bit	CIREAA
	16-bit	-

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis.

S2: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis.

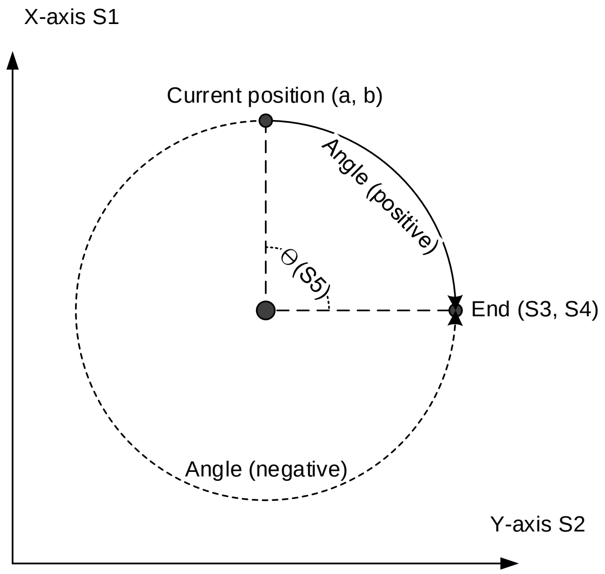
S3: X-coordinate of the arc end (unit: mm).

S4: Y-coordinate of the arc end (unit: mm).

S5: angle (unit: degree).

S6: maximum speed (unit: default is puu/s; you can use the GUNIT instruction to select the speed unit).

S1 is the X servo axis and S2 is the Y servo axis. The X-coordinate of the arc end is S3 and the Y-coordinate is S4. The angle between the current position and the arc end is S5. Axis X and Axis Y move at the maximum speed set by S6.



7

■ CIREAR

<table>
<tr><td>NO.
39</td><td>-</td><td>CIREAR</td><td colspan="2">S1, S2, S3, S4, S5, S6</td><td colspan="3">Arc relative motion (end, angle)</td></tr>
<tr><td rowspan="2"></td><td colspan="2">Bit device</td><td colspan="3">Word device</td><td colspan="2">Code and symbol</td></tr>
<tr><td>M</td><td>-</td><td>K</td><td>D</td><td>E</td><td>Axis No.</td><td>Operator</td></tr>
<tr><td>S1</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>●</td><td>-</td></tr>
<tr><td>S2</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>●</td><td>-</td></tr>
<tr><td>S3</td><td>-</td><td>-</td><td>●</td><td>●(DW)</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>S4</td><td>-</td><td>-</td><td>●</td><td>●(DW)</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>S5</td><td>-</td><td>-</td><td>●</td><td>●</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>S6</td><td>-</td><td>-</td><td>●</td><td>●(DW)</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="5" rowspan="3">Notes when applying operand:
If S3, S4, and S6 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the CIREAR instruction reads 35.997.</td><td></td><td colspan="2">Instruction</td></tr>
<tr><td>32-bit</td><td colspan="2">CIREAR</td></tr>
<tr><td>16-bit</td><td colspan="2">-</td></tr>
</table>

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis.

S2: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis.

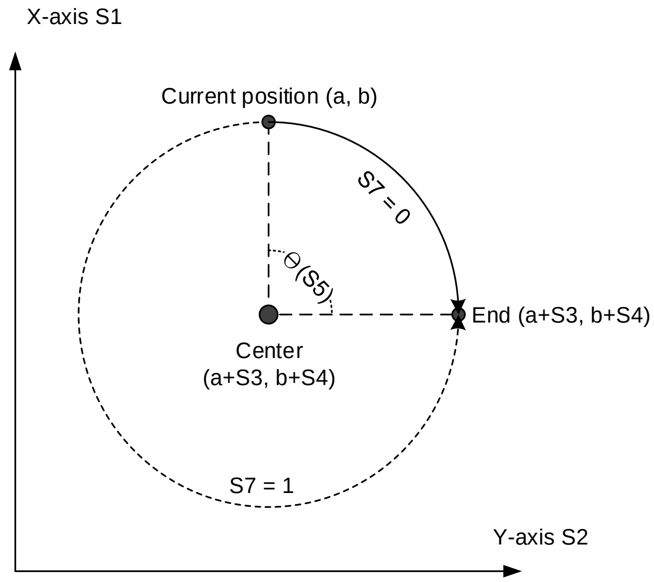
S3: relative distance to the arc end (X-axis, unit: mm).

S4: relative distance to the arc end (Y-axis, unit: mm).

S5: angle (unit: degree).

S6: maximum speed (unit: default is puu/s; you can use the GUNIT instruction to select the speed unit).

S1 is the X servo axis and S2 is the Y servo axis. The X-coordinate of the arc end is its current coordinate + S3 and the Y-coordinate is its current coordinate + S4. The angle between the current position and the arc end is S5. Axis X and Axis Y move at the maximum speed set by S6.



7

■ **CIRCEA**

NO. 40	-	CIRCEA	S1, S2, S3, S4, S5, S6, S7, S8	Arc absolute motion (center, arc end)			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	-	-	-	●	-
S3	-	-	●	●(DW)	-	-	-
S4	-	-	●	●(DW)	-	-	-
S5	-	-	●	●(DW)	-	-	-
S6	-	-	●	●(DW)	-	-	-
S7	-	-	●	●	-	-	-
S8	-	-	●	●(DW)	-	-	-

Notes when applying operand:		Instruction
If S3, S4, S5, S6, and S8 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the CIRCEA instruction reads 35.997.	32-bit	CIRCEA
	16-bit	-

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis.

S2: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis.

S3: X-coordinate of the center (unit: mm).

S4: Y-coordinate of the center (unit: mm).

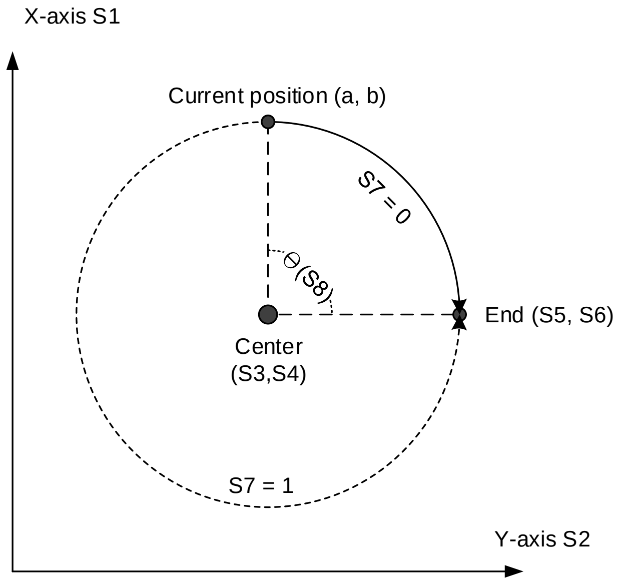
S5: X-coordinate of the arc end (unit: mm).

S6: Y-coordinate of the arc end (unit: mm).

S7: direction (0: CW; 1: CCW).

S8: maximum speed (unit: default is puu/s; you can use the GUNIT instruction to select the speed unit).

S1 is the X servo axis and S2 is the Y servo axis. The X-coordinate of the center is S3 and the Y-coordinate is S4. The X-coordinate of the arc end is S5 and the Y-coordinate is S6. Axis X and Axis Y move at the maximum speed set by S8.



Example: Axis X and Axis Y move to the position (-15, 15), make a 1/4 arc motion around the center at (0, 0) in clockwise direction, and stop at the position (15,15). Then, this MPM ends.

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
MOVA,X,50,-15,Y,50,15	Axis X and Axis Y move to the position (-15, 15).
CIRCEA,X,Y,0,0,15,15,0,50	Move in circular motion.

■ **CIRCER**

NO.	-	CIRCER	S1, S2, S3, S4, S5, S6, S7, S8		Arc relative motion (center, arc end)		
41							
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	-	-	-	●	-
S3	-	-	●	●(DW)	-	-	-
S4	-	-	●	●(DW)	-	-	-
S5	-	-	●	●(DW)	-	-	-
S6	-	-	●	●(DW)	-	-	-
S7	-	-	●	●	-	-	-
S8	-	-	●	●(DW)	-	-	-

Notes when applying operand:		Instruction
If S3, S4, S5, S6, and S8 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the CIRCER instruction reads 35.997.	32-bit	CIRCER
	16-bit	-

Description: S1: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis.

S2: X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis.

S3: relative distance to the center (X-axis, unit: mm).

S4: relative distance to the center (Y-axis, unit: mm).

S5: relative distance to the arc end (X-axis, unit: mm).

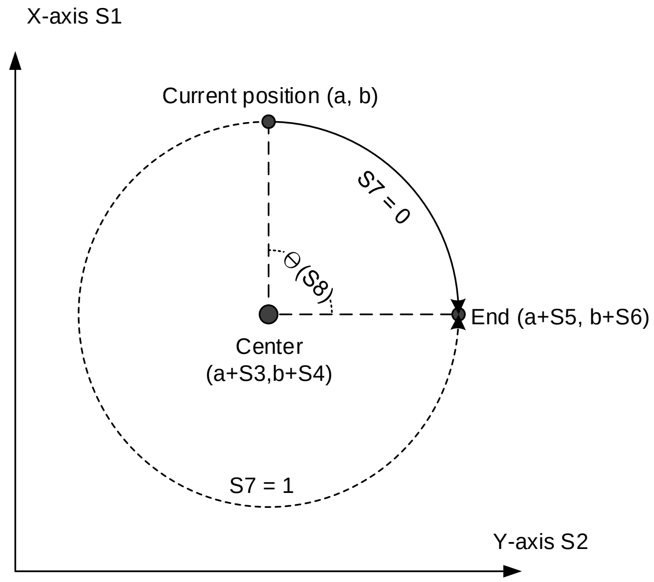
S6: relative distance to the arc end (Y-axis, unit: mm).

S7: direction (0: CW; 1: CCW).

S8: maximum speed (unit: default is puu/s; you can use the GUNIT instruction to select the speed unit).

S1 is the X servo axis and S2 is the Y servo axis. The X-coordinate of the center is its current coordinate + S3 and the Y-coordinate is its current coordinate + S4.

The X-coordinate of the arc end is its current coordinate + S5 and the Y-coordinate is its current coordinate + S6. Axis X and Axis Y move at the maximum speed set by S8.



Example: Axis X and Axis Y move to the position (-15, 15), make a 1/4 arc motion around the center at (0, 0) in clockwise direction, and stop at the position (15, 15). Then, this MPM ends.

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
MOVA,X,50,-15,Y,50,15	Axis X and Axis Y move to the position (-15, 15).
CIRCER,X,Y,15,-15,30,0,0,50	Move in circular motion.

■ HELIXA

NO.	-	HELIXA	S1, S2, S3, S4, S5, S6, S7, S8, S9	Three-axis helical interpolation in absolute motion
42				

	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	-	-	-	●	-
S3	-	-	-	-	-	●	-
S4	-	-	●	●(DW)	-	-	-
S5	-	-	●	●(DW)	-	-	-
S6	-	-	●	●(DW)	-	-	-
S7	-	-	●	●(DW)	-	-	-
S8	-	-	●	●	-	-	-
S9	-	-	●	●(DW)	-	-	-

Notes when applying operand:		Instruction
If S4, S5, S6, S7, and S9 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the HELIXA instruction reads 35.997.	32-bit	HELIXA
	16-bit	-

Description: S1: specify Axis 1 (X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis).

S2: specify Axis 2 (X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis).

S3: specify Axis 3 (X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis).

S4: X-coordinate of the center.

S5: Y-coordinate of the center.

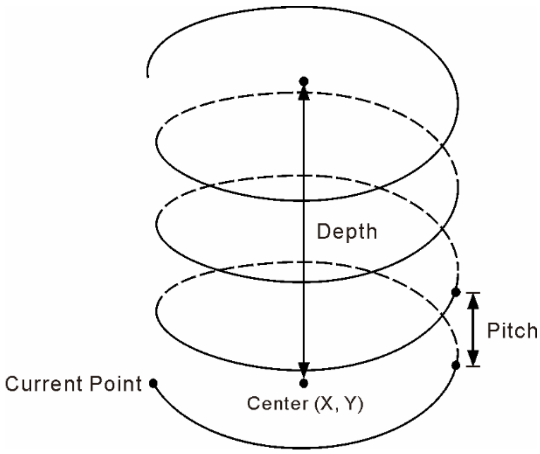
S6: helix depth: the overall rising height.

S7: helix pitch: the height between two turns of arc.

S8: direction (0: CW; 1: CCW).

S9: maximum speed (unit: default is puu/s; you can use the GUNIT instruction to select the speed unit).

S1 is the X-direction servo axis, S2 is the Y-direction servo axis, and S3 is the Z-direction servo axis. The X-coordinate of the helix center is S4 and the Y-coordinate is S5. The overall helix depth is S6, the helix pitch is S7, and the direction is S8. The servo axes specified by S1 and S2 apply S9 as the linear speed to calculate the speed of each axis.



Note: the speed of S1 and S2 are calculated according to the maximum speed (S9), and the speed of S3 is calculated according to the values of the helix depth (S6) and the helix pitch (S7).

The speed calculation formula of the vertical axis S3:

$$\frac{\text{S7 Pitch} \ * \text{S9 Maxve}}{2\pi \times \text{Radius}}$$

Example: Axis X and Axis Y move to the position (-15, 15), make an arc motion around the center at (0, 0) in clockwise direction.

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
MOVA,X,50,-15,Y,50,15	Axis X and Axis Y move to the position (-15, 15).
HELIXA,X,Y,Z,0,0,100,10,0,50	Move in helical circular motion.

The two axes elevate by 10 mm every turn and stop moving when reaching 100 mm on Z-axis. Then, this MPM ends.

■ HELIXR

NO. 43	-	HELIXR	S1, S2, S3, S4, S5, S6, S7, S8, S9	Three-axis helical interpolation in relative motion			
	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	-	-	-	●	-
S3	-	-	-	-	-	●	-
S4	-	-	●	●(DW)	-	-	-
S5	-	-	●	●(DW)	-	-	-
S6	-	-	●	●(DW)	-	-	-
S7	-	-	●	●(DW)	-	-	-
S8	-	-	●	●	-	-	-
S9	-	-	●	●(DW)	-	-	-

Notes when applying operand:		Instruction
If S4, S5, S6, S7, and S9 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the HELIXR instruction reads 35.997.	32-bit	HELIXR
	16-bit	-

Description: S1: specify Axis 1 (X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis).

S2: specify Axis 2 (X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis).

S3: specify Axis 3 (X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis, W: 6th axis).

S4: relative distance to the center (X-axis).

S5: relative distance to the center (Y-axis).

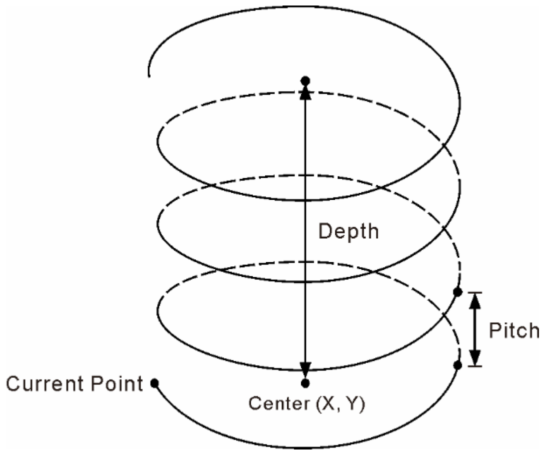
S6: helix depth: the overall rising height.

S7: helix pitch: the height between two turns of arc.

S8: direction (0: CW; 1: CCW).

S9: maximum speed (unit: default is puu/s; you can use the GUNIT instruction to select the speed unit).

S1 is the X-direction servo axis, S2 is the Y-direction servo axis, and S3 is the Z-direction servo axis. The X-coordinate of the helix center is its current coordinate + S4 and the Y-coordinate is its current coordinate + S5. The overall helix depth is S6, the helix pitch is S7, and the direction is S8. The servo axes specified by S1 and S2 apply S9 as the linear speed to calculate the speed of each axis.



Note: the speed of S1 and S2 are calculated according to the maximum speed (S9), and the speed of S3 is calculated according to the values of the helix depth (S6) and the helix pitch (S7).

The speed calculation formula of the vertical axis S3:

$$\frac{S7\ \ \text{Pitch}\ \ * S9\ \ \text{Maxve}}{2\pi \times \text{Radius}}$$

Example: Axis X and Axis Y move to the position (-15, 15), make an arc motion around the center at (0, 0) in clockwise direction. The two axes elevate by 10 mm every turn and stop moving when reaching 100 mm on Z-axis. Then, this MPM ends.

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
MOVA,X,50,-15,Y,50,15	Axis X and Axis Y move to the position (-15, 15).
HELIXA,X,Y,Z,15,-15,100,10,0,50	Move in helical circular motion.

■ TAPPING

NO.	-	TAPPING	S1, S2, S3, S4, S5, S6, S7, S8	Tapping				
44								

	Bit device		Word device			Code and symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	-	-	-	●	-
S3	-	-	●	●(DW)	-	-	-
S4	-	-	●	●(DW)	-	-	-
S5	-	-	●	●	-	-	-
S6	-	-	●	●	-	-	-
S7	-	-	●	●	-	-	-
S8	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
If S3 and S4 use register D, the last three digits are decimals. For example, if the value of register D100 is K35997, the TAPPING instruction reads 35.997.	32-bit	TAPPING
	16-bit	-

Description: S1: rotation axis (X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis 5, W: 6th axis).

S2: feeding axis (X: 1st axis, Y: 2nd axis, Z: 3rd axis, U: 4th axis, V: 5th axis 5, W: 6th axis).

S3: tapping depth (unit: mm).

S4: tapping pitch (unit: mm).

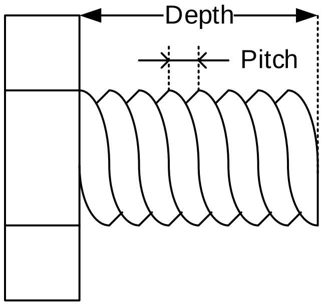
S5: tapping speed (unit: rpm).

S6: retraction speed (unit: rpm).

S7: retraction delay (unit: s).

S8: clockwise / counterclockwise (0: CW; 1: CCW).

This instruction is for tapping only. The feeding axis S2 executes feeding according to the tapping speed (S5) and tapping pitch (S4), and stops when reaching the set value of S3. After the duration set by S7, it moves back to the origin at the retraction speed set by S6.



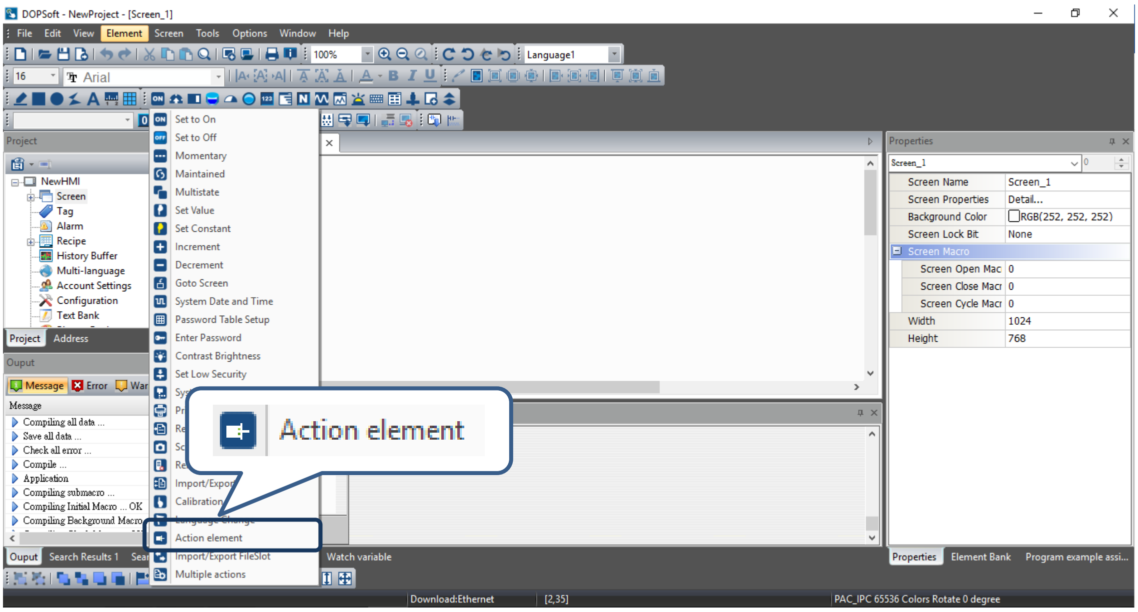
Example: the tapping speed of rotation axis X is 100 rpm, and at each turning cycle of axis X, the corresponding axis Y feeds 7 mm (pitch). The total feeding distance of axis Y is 70 mm. When the tapping finishes, axis X and axis Y decelerate to stop and delay for 65 ms, then the rotation axis X carries out the retraction at 160 rpm in reverse direction.

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
GTADC,0.3,0.3	Set the acceleration and deceleration time.
TAPPING,X,Y,70,7,100,160,65,0	Tapping procedure.
SETM,0	Set the auxiliary relay M0 to ON.

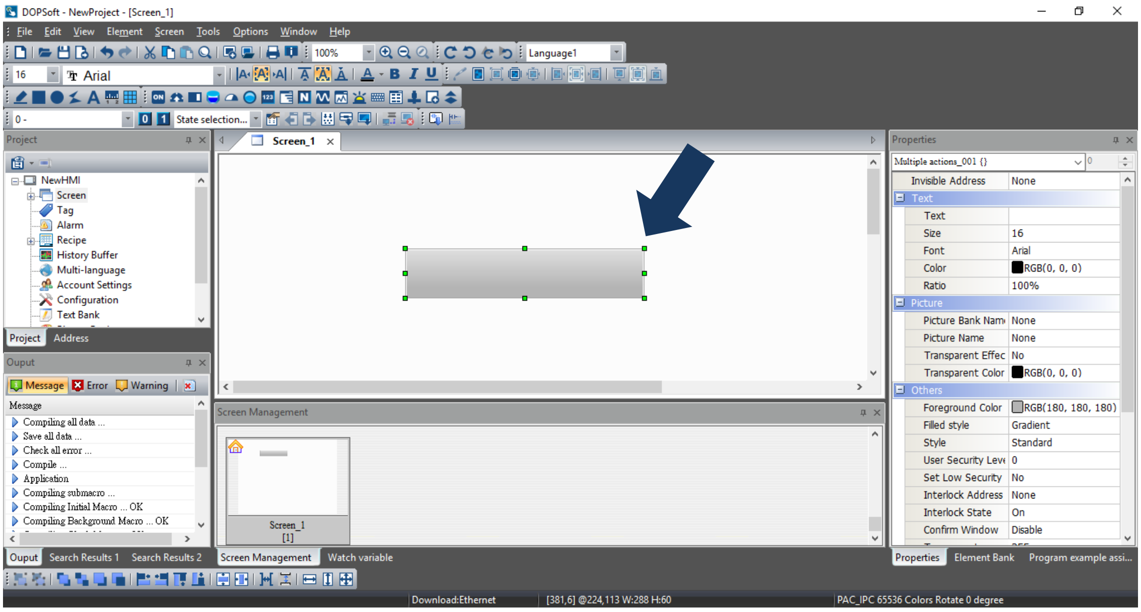
7.2 Motion Program Macro (MPM) editor

■ Use the SOFTHMI software to call the MPM editor

1. Create an Action element on the editing screen of DOPSoft.

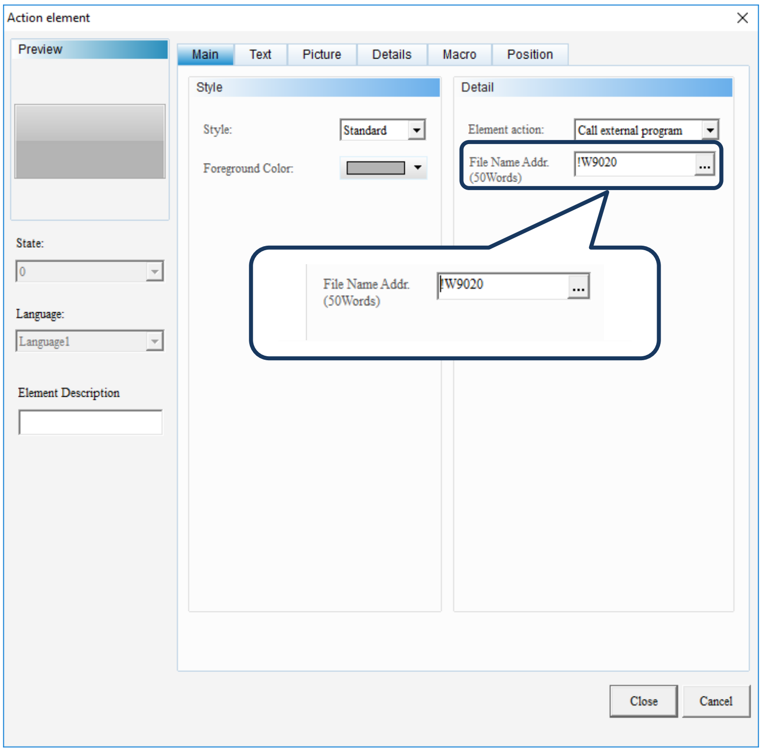


2. Draw an element.

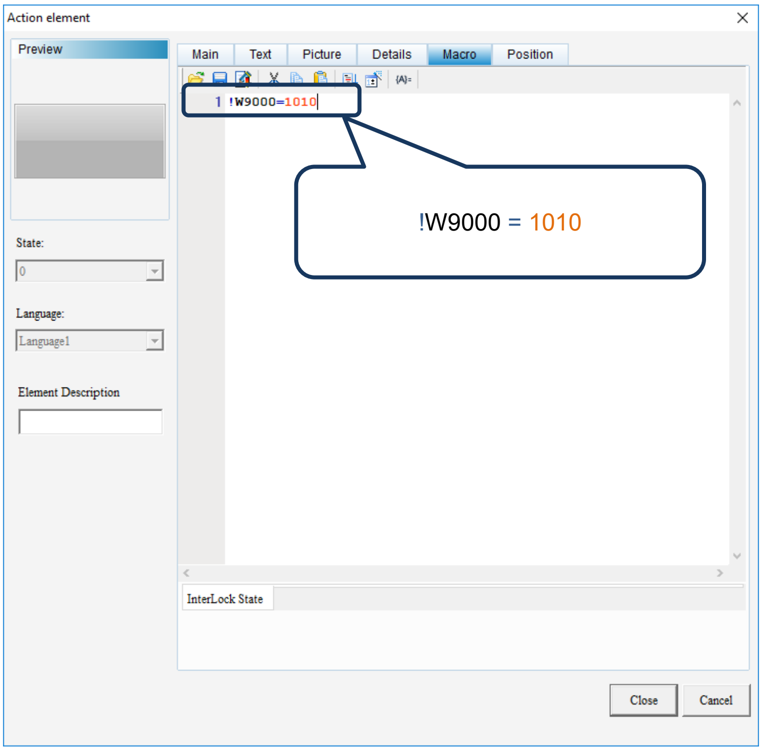


7

3. Double-click the element and enter “!W9020” to File Name Addr.

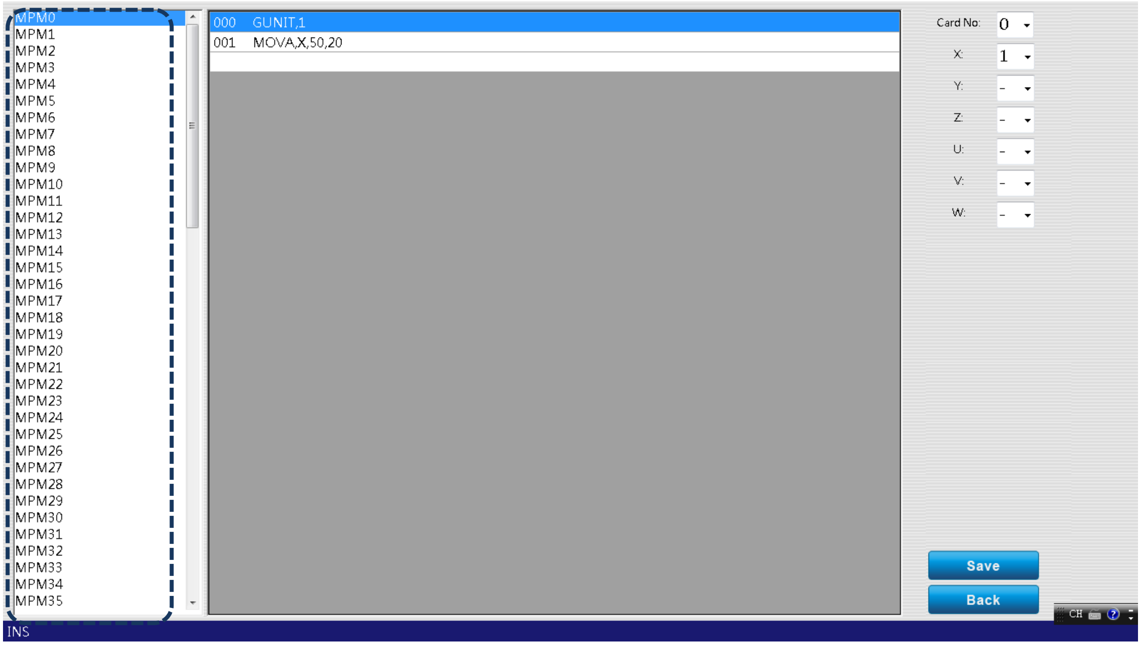


4. Click Macro and enter “!W9000=1010”. Then, click **Close**.

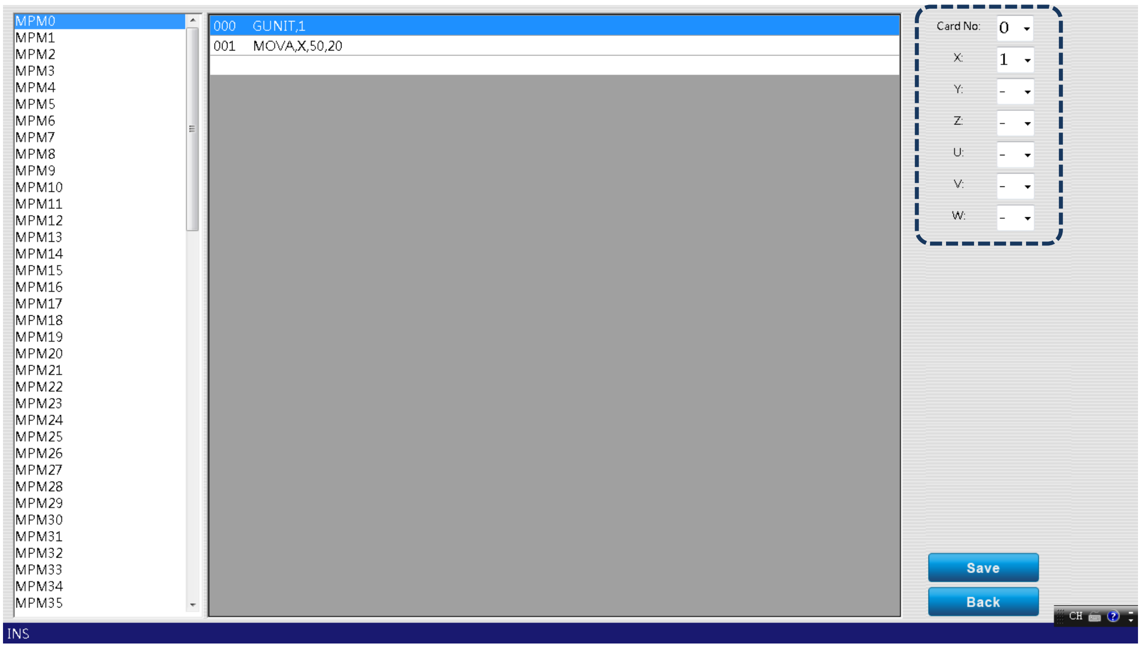


■ Operating instructions for the MPM editor

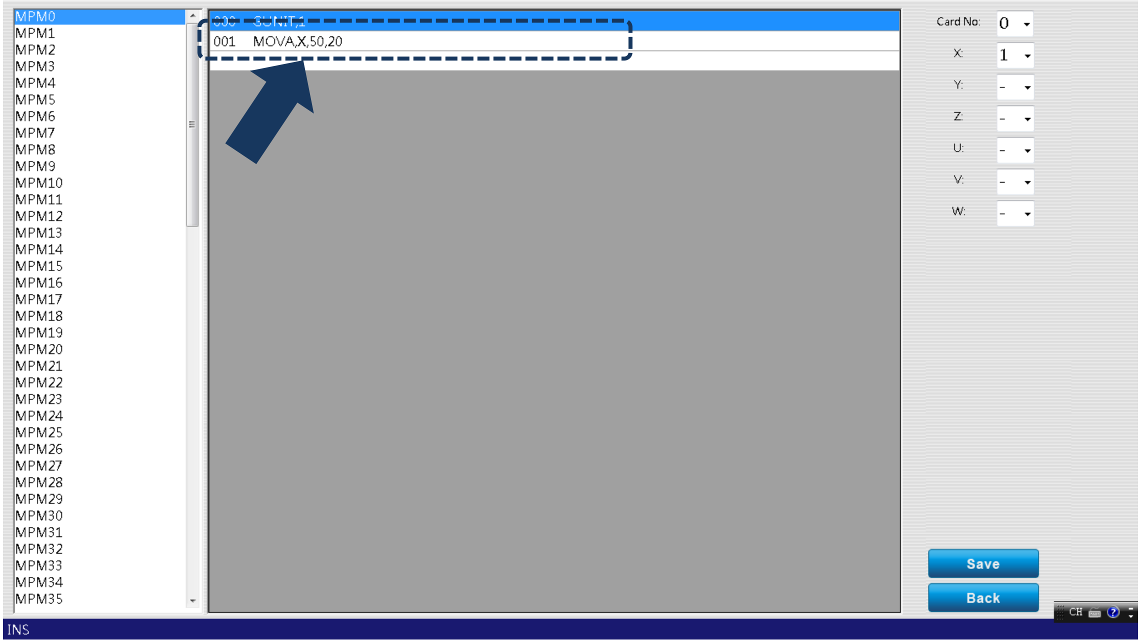
1. Select the MPM No. (valid range is 0 to 99) you want to edit, then you can open the editing function and save the MPM.



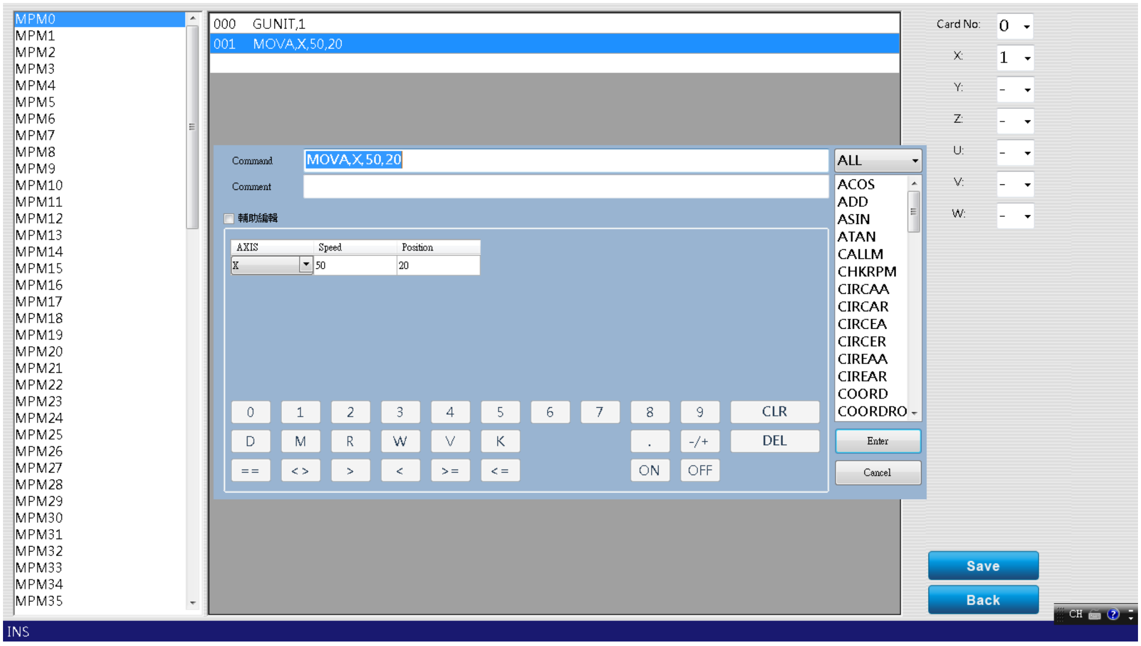
2. Select the card number and axis. The range of the card number is between 0 and F; the available axes are X, Y, Z, U, V and W, which you can set the corresponding node number.



3. In the middle of the screen is the instruction display section. Select the instruction for editing and double-click the left mouse button or press **Enter** to go to the instruction editing window.



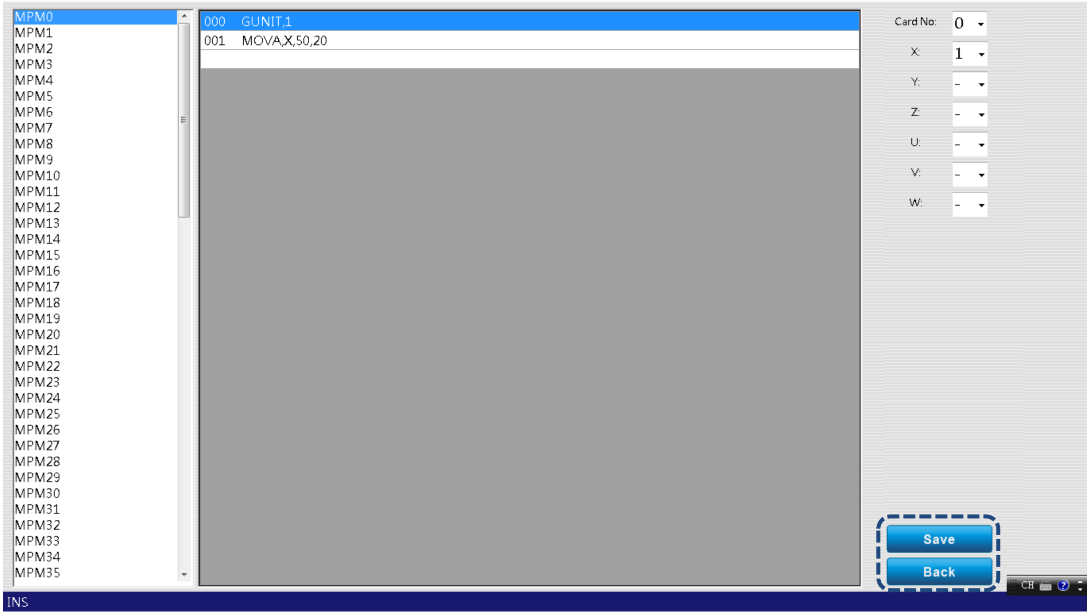
4. You can directly type in the instruction or select the instruction from the drop-down list on the right. It is suggested that you first select the instruction type.



5. The [Assistant editor] function will display different parameter items according to different instructions to guide the user. You can apply functions like copy, paste, insert, and delete when editing a line of instruction.

6. After you are done editing the parameters, click **Enter** to complete the editing of the instruction line.

7. Click **Save** to save the macros when you are done editing. Then, click **Back** to exit the MPM editor.



Communication 8

You can find the information about the communication setting and related definition in this chapter before applying the IMP communication function.

8

8.1 Modbus communication setting

Go to [Quick start] > [IPC Motion Platform] to enter the Modbus TCP / Modbus Serial setting page. On the setting page, you can view or change the IMP communication parameters, including the slave station number and TCP port. IMP can also be regarded as the master station and by setting the automatic communication list, the IMP system can automatically exchange data with devices D and M through the communication interface.

The communication setting interface is divided into: (1) tree view of communication setting and (2) setup section. Click [IPC Motion Platform] in the tree view to display the IMP communication configuration settings. See the detailed information below:

Name	Description	Default
IMP Slave station	Set the station number for the communication slave.	1
TCP Port	Set as the port used by the Modbus / TCP slave station.	502
TCP port Amount	Set the port number used by the Modbus / TCP master station. This amount is distinguished by the IP address of the connected device with one channel for one IP address.	0
COM port Amount	Set the COM port number used by the master station of the Modbus serial communication. This amount is distinguished by the communication port (COM) provided by the host with one channel for one port.	0

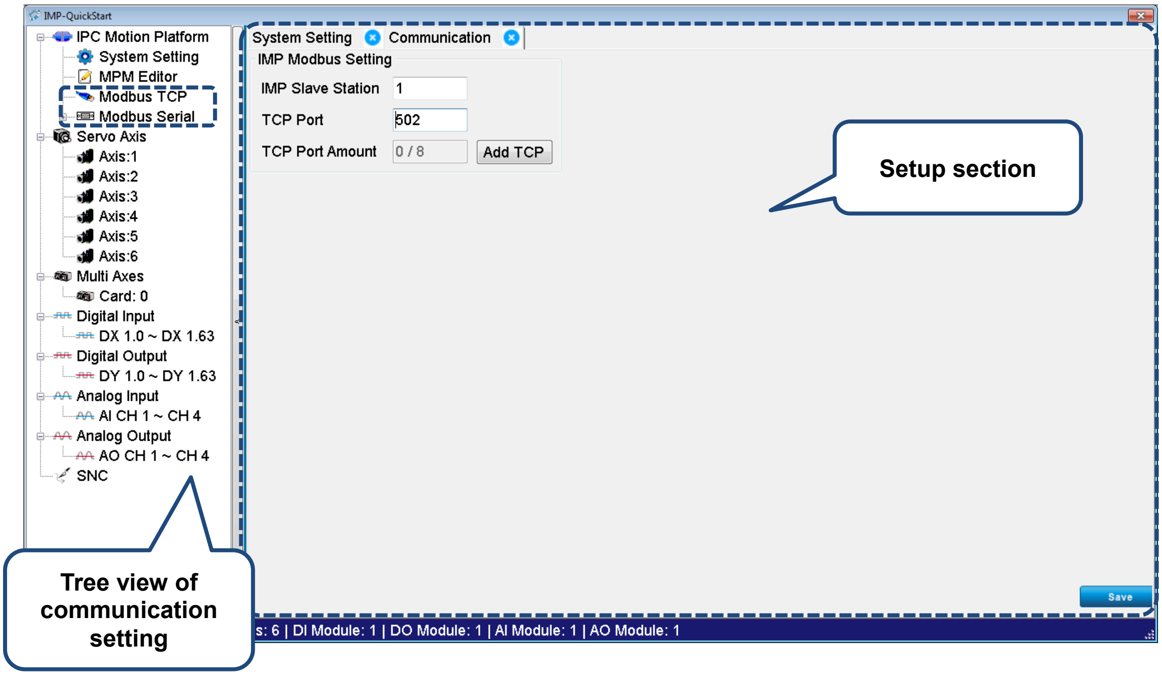


Figure 8.1.1 IMP communication interface

8.1.1 Ethernet communication setting

■ Modbus / TCP port setting

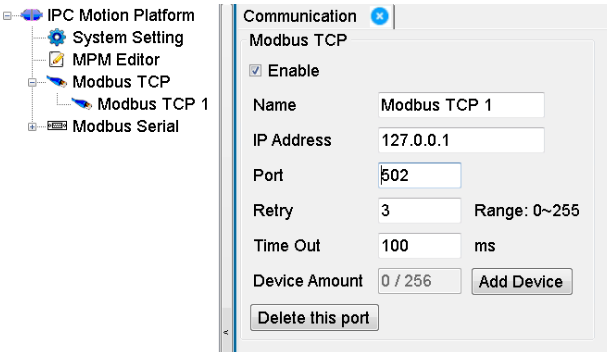


Figure 8.1.1.1 Modbus / TCP port setting

Ethernet channel setting is for creating the Modbus / TCP connection. By clicking on the Ethernet channel name (e.g. TCP / IP1) from the tree view of communication setting on the left, the Ethernet channel setup section appears on the right. See the detailed descriptions below:

Name	Description
Enable	Check [Enable] to enable the Ethernet channel.
Name	Set the port name for the Ethernet connection.
IP Address	Set the IP address of the connected equipment.
Port	Set the network communication port for the connected equipment.
Retry	Set the retry times when data transmission fails. The range is between 0 and 225 times.
Time Out	Set the communication timeout (ms). (The set value must be at least 10 ms.)
Device Amount	Display the connected device amount of the Ethernet channel. (Add Device: add new communication device.)
Delete this port	Delete the Ethernet channel.

■ Modbus / TCP connection device setting

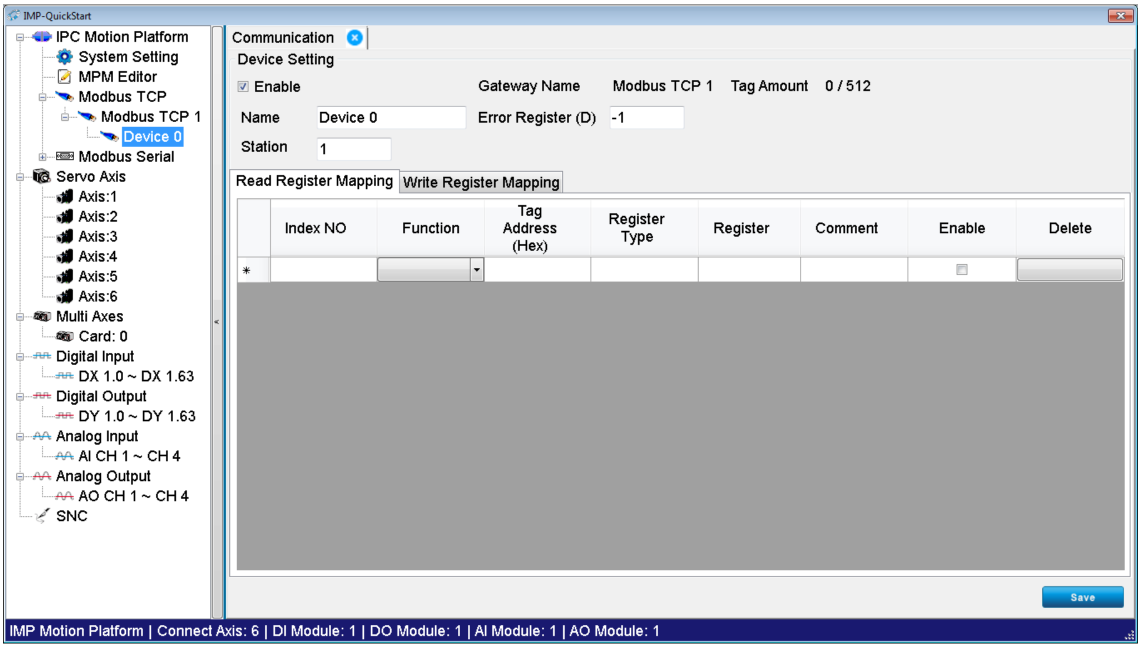


Figure 8.1.1.2 Modbus / TCP connection device setting

Ethernet connection device setting is used when IMP reads and writes the device data via communication. Through the setting of the command mapping table, the communication command will be automatically generated during operation. And the communication data will be mapped to the internal memory device of the IMP PLC. By clicking on the Ethernet device name (e.g. Device) from the tree view of communication setting on the left, the Ethernet device setup section appears on the right. See the detailed descriptions below:

Name	Description
Enable	Check [Enable] to enable the communication device connected to the Ethernet.
Name	Set the device name.
Station	Set the device station number with the range between 0 and 225.
Gateway name	Display the name of the currently used connection channel.
Error Register	Set the D register address for saving the communication error code. When set to -1, no error code will be shown*1.
Tag Amount	Display the number of currently used communication addresses.
Read Register Mapping	Mapping table for reading the device*2.
Write Register Mapping	Mapping table for writing to the device*2.

Note:

1. For details about the error code table, refer to Section 8.1.4.
2. For settings of the device mapping table, refer to Section 8.1.3.

8

8.1.2 Serial communication setting

■ Serial communication port setting

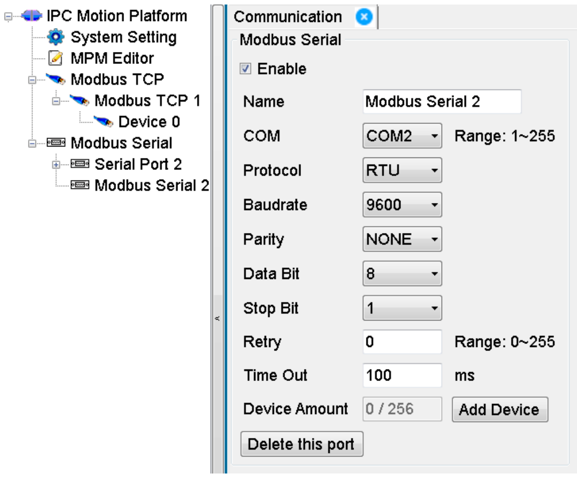


Figure 8.1.2.1 Modbus / Serial port setting

Serial communication port setting is for creating the Modbus / Serial connection. By clicking on the serial port name (e.g. Serial Port 1) from the tree view of communication setting on the left, the setup section appears on the right. See the detailed descriptions below:

Name	Description
Enable	Check [Enable] to enable the serial connection port.
Name	Set the connection port name.
COM	Set the COM port number used by the connection channel.
Protocol	Set the communication protocol format: ASCII or RTU.
Baud rate	Set the serial communication baud rate: 4800, 9600, 19200, 38400, 57600, or 115200.
Parity	Set the communication parity check mechanism: None, Odd, or Even.
Data Bit	Set the length of the communication data. The standard length of each set of data bit is 7 or 8.
Stop Bit	Set the length of the stop bit: 1 or 2.
Retry	Set the retry times when data transmission fails. The range is between 0 and 225 times.
Time Out	Set the communication timeout (ms). (The set value must be at least 10 ms.)
Device Amount	Display the connected device amount of the serial communication port. (Add Device: add new communication device.)
Delete this port	Delete the serial communication port.

- Modbus / Serial connection device setting

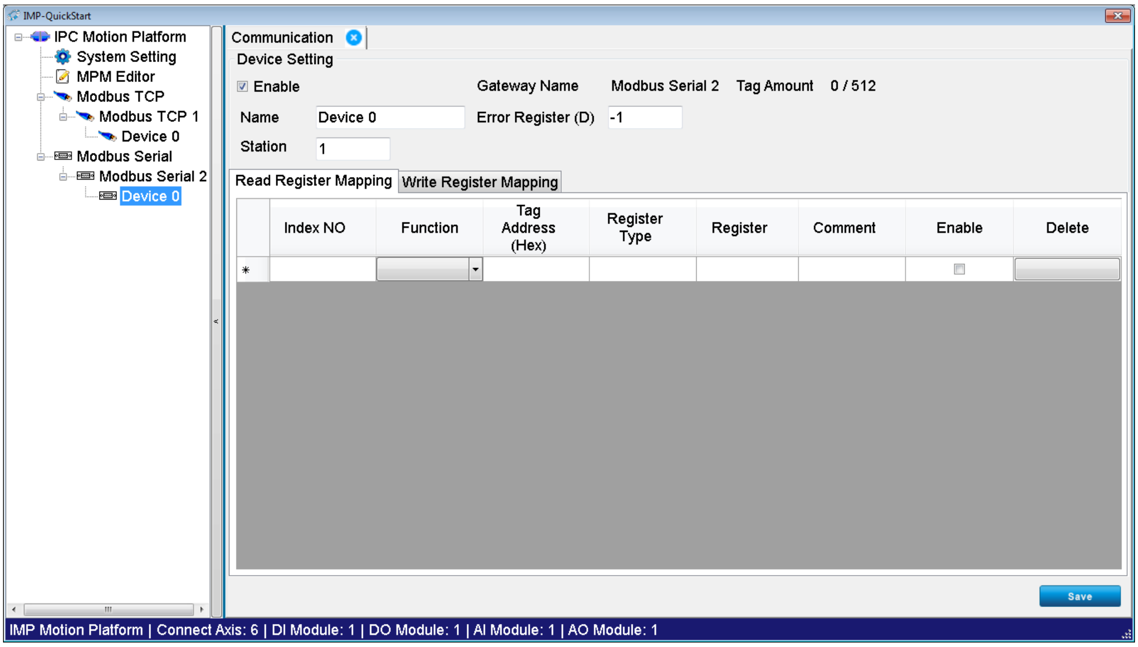


Figure 8.1.1.2 Modbus / Serial connection device setting

Serial connection device setting is used when IMP reads and writes the instruction list of the communication device. Through the list, IMP automatically generates communication instructions during operation and the communication data will be mapped to the internal memory device of the IMP PLC. By clicking on the serial device name (e.g. Device) from the tree view of communication setting on the left, the setup section appears on the right. See the detailed descriptions below:

Name	Description
Enable	Check [Enable] to enable the communication connection of the connected device.
Name	Set the name of the connection device.
Station	Set the station number that connects to the communication device.
Gateway name	Display the gateway name used by the current connection device.
Error Register	Set the D register address for saving the communication error code. When set to -1, no error code will be shown[*1].
Tag Amount	Display the number of currently used communication addresses.
Read Register Mapping	Mapping table for reading the device[*2].
Write Register Mapping	Mapping table for writing to the device[*2].

Note:

1. For details about the error code table, refer to Section 8.1.4.
2. For settings of the device mapping table, refer to Section 8.1.3.

8.1.3 Communication instruction setting

■ Read Register Mapping table

Read Register Mapping | Write Register Mapping

	Index	Function	Tag Address (Hex)	Type	D Register	Comments	Enable	Delete
	1	RW	0010	D	100	speed h	☑	Delete
	2	RW	0011	D	101	speed l	☑	Delete
	3	RW	0012	D	102	Timer 1	☑	Delete
	4	RW	0013	D	103	Timer 2	☑	Delete
	5	RB	0014	M	200	start	☑	Delete
	6	RB	0015	M	201	stop	☑	Delete
	7	RB	0016	M	202	pause	☑	Delete
	8	RB	0017	M	203	men/auto	☑	Delete
*							☐	

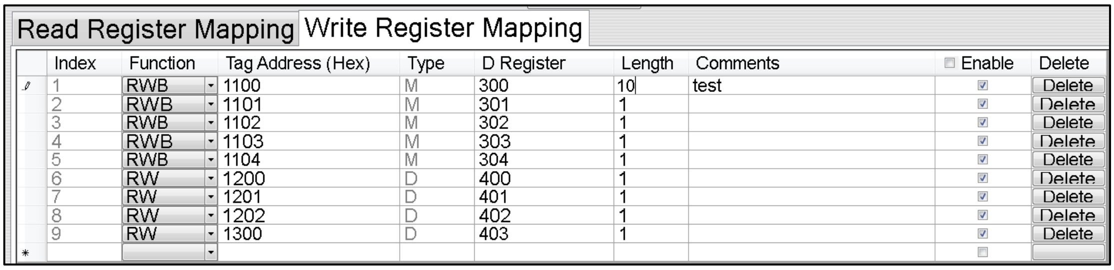
Figure 8.1.3.1 Read device mapping table of Modbus

Through the Read Register Mapping table setting, IMP will continue to issue the Modbus read command during the execution process, and store the returned value in the corresponding memory devices (D, M). If the communication addresses are consecutive or the interval between each address is less than 100, those addresses will be automatically read in batch (0X10) to optimize the communication.

Name	Description
Index	Serial number of the communication instruction.
Function	Modbus function code: RW(0x03), R(0x04), RWB(0x01), and RB(0x02).
Tag Address	Set the communication address of the read data, which is displayed in hexadecimal format, for example: $FF1A_{16}$.
Type	Display the device type for data storage. D: data register. M: auxiliary relay.
Register	Set the device address for data storage.
Comments	Add comments in this column.
Enable	Check [Enable] to enable the communication instruction.
Delete	Delete the communication instruction.

■ Write Register Mapping table

Modbus parameter setting page



Read Register Mapping | Write Register Mapping

Index	Function	Tag Address (Hex)	Type	D Register	Length	Comments	Enable	Delete
1	RWB	1100	M	300	10	test	☑	Delete
2	RWB	1101	M	301	1		☑	Delete
3	RWB	1102	M	302	1		☑	Delete
4	RWB	1103	M	303	1		☑	Delete
5	RWB	1104	M	304	1		☑	Delete
6	RW	1200	D	400	1		☑	Delete
7	RW	1201	D	401	1		☑	Delete
8	RW	1202	D	402	1		☑	Delete
9	RW	1300	D	403	1		☑	Delete

Figure 8.1.3.2 Write device mapping table of Modbus

Name	Description
Index	Serial number of the communication instruction.
Function	Modbus function code: RW(0x06) and RWB(0x05).
Tag Address	Set the communication address to write data, which is displayed in hexadecimal format, for example: $FF1A_{16}$.
Type	Display the device type for data storage. D: data register. M: auxiliary relay.
Register	Set the device address of the data source.
Length	Set the communication length. The default is 1 (unit: word).
Comments	Add comments in this column.
Enable	Check [Enable] to enable the communication instruction.
Delete	Delete the communication instruction.

8.1.4 Communication error code

Code	Description
01	Wrong function code. The communication function code is not supported.
02	Wrong communication address. Accessing illegal communication address.
03	Communication data error.
04	Slave station error. Unknown error occurred.
06	Slave station is busy. The instruction is not completed.
101	Failed to enable COM / TCP connection.
102	COM port setting exceeded the range.
103	COM port is not enabled.
104	Modbus function code error.
105	The length of the read data exceeds the maximum limit. The maximum length is 100 words or 200 bits.
106	Slave station number setting error. The valid range is between 1 and 255.
107	Address of accessing device exceeds the range. The valid range is between 0 and 65535.
108	Serial communication timeout.
109	Communication check error (RTU CRC).
110	Communication check error (ASCII LRC).
111	Connection port initialization failed.
112	Connection to the Modbus master station failed.
113	TCP communication transmission failed.
114	Modbus / TCP communication timeout.
116	TCP port creation error.
120	The length of the written data exceeds the maximum limit. The maximum length is 100 words or 200 bits.
121	The length of the read data exceeds the maximum limit.
202	Undefined communication instruction is used.
203	Transmitted a single communication instruction using a COM port that is set to off.
204	Transmitted a single communication instruction using a COM port that has not been set up.
205	COM port failed to enable when transmitting a single communication instruction.

8.2 Modbus communication address

IMP supports Ethernet, RS-485, RS-422, and RS-232 communication protocols. The memory address range supported by the Modbus Server, the correspondence with the Modbus communication address, and the function codes supported by the respective addresses are shown in the table below.

- Modbus communication protocol

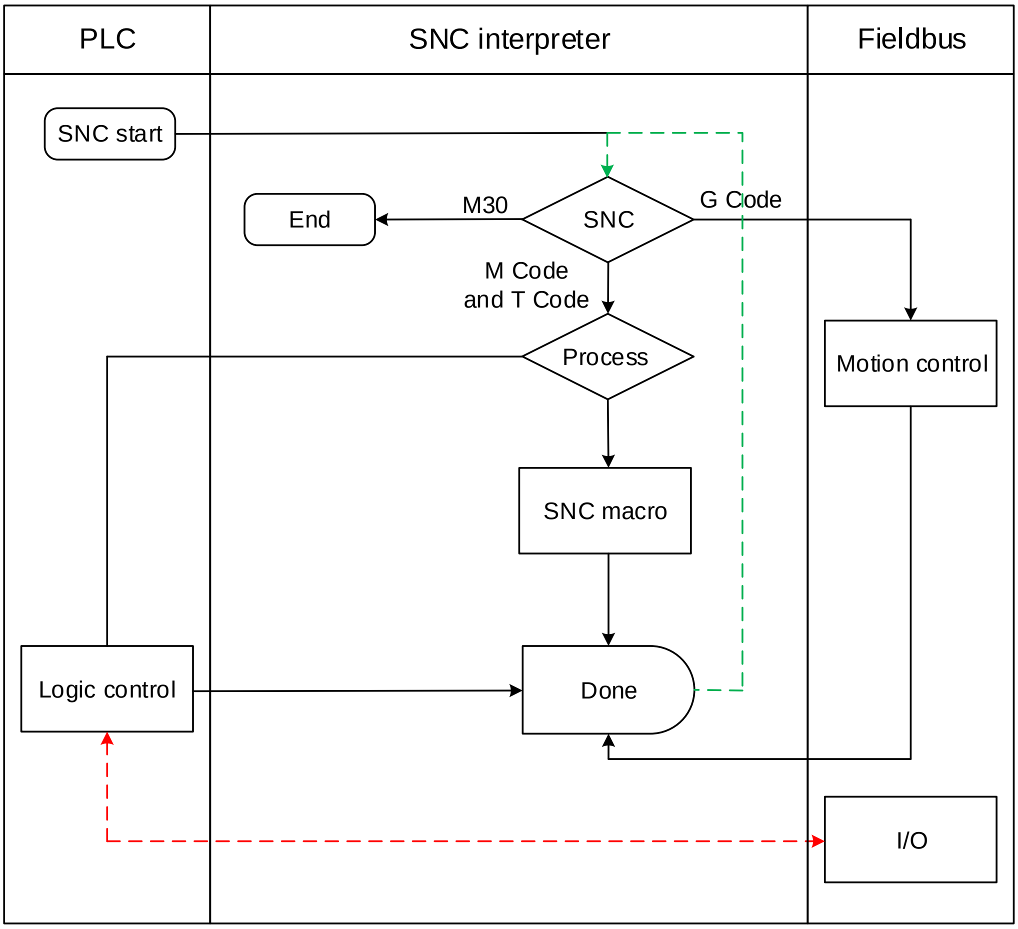
Device	Range	Type	Modbus communication address (Hex)	Modbus / TCP function code
M	M0 - M19999	Bit	0000 - 4E1F	01, 05, 0F
DX	DX1.0 - DX36.63	Bit	D000 - D8FF	02
DY	DY1.0 - DY36.63	Bit	E000 - E8FF	01, 05, 0F
T	T0 - T256	Bit	F000 - F0FF	01, 05, 0F
	T0 - T256	Word	F000 - F0FF	03, 04, 06, 10, 17
C	C0 - C255	Bit	F800 - F8FF	01, 05, 0F
	C0 - C199	Word	F800 - F8C7	03, 04, 06, 10, 17
	C200 - C255	DWord	F8C8 - F937	03, 04, 06, 10, 17
D	D0 - D59999	Word	0000 - EA5F	03, 04, 06, 10, 17

Software Numerical Control (Optional) 9

Soft Numerical Control (SNC) is an interpreter built on the basis of a communication control system. It mainly assists the motion control core to execute route calculation with the computer core computing, such as G-code interpretation, short path fitting, original path reversing, etc. In addition to G-code, M-code and T-code are also provided for the programmers of Programmable Logic Control (PLC) to adjust the equipment.

9.1 SNC framework

IMP is integrated with Soft Numerical Control (SNC). When SNC is started by the PLC, the interpreter is automatically loaded with the G-code file for route setting, and in accordance with the motion path in the file, it will issue motion commands through the fieldbus. If the given command is recognized as M-code or T-code, the interpreter will distinguish where the command belongs by reference to the SNC parameters. Assuming that the PLC receives the control, PLC will read the status of the memory device and execute the corresponding actions (e.g. feeding or retrieving the cutting tool, on / off of the cutting fluid). Once the actions are completed, the control will be returned to the SNC interpreter; when it is determined that there is no need to transfer the control, the SNC interpreter will automatically execute the command actions, such as program pause and program end.



9

9.2 Parameter descriptions

This section describes the parameter settings for the various functions of the SNC interpreter. You can modify the parameters with the IMP Quick Start interface (refer to Section 3.2.7).

9.2.1 Tool

Tool information

Set the tool number and type for the SNC.

Parameter	Description	Default
Spindle_Current_T	Set the tool number of the current spindle. This parameter will be modified synchronously when the spindle tool exchange is completed during the SNC operation. Parameter range: 1 - 100.	1
SpindleToolNo1	Set the first tool number of the spindle. When set to 0, it indicates that the spindle tool is not in use. Parameter range: 0 - 100.	1
SpindleToolCnt	Set the total number of spindle tools. When set to 0, it indicates that the spindle tool is not in use. Parameter range: 0 - 100.	100
VerticalDrillNo1	Set the first tool number for vertical drilling. When set to 0, it indicates that the vertical drilling tool is not in use. Parameter range: 0 - 100.	0
VerticalDrillCnt	Set the total number of vertical drills. When set to 0, it indicates that the vertical drill is not in use. Parameter range: 0 - 100.	0
HorizontalDrillNo1	Set the first tool number for horizontal drilling. When set to 0, it indicates that the horizontal drilling tool is not in use. Parameter range: 0 - 100.	0
HorizontalDrillCnt	Set the total number of horizontal drills. When set to 0, it indicates that the horizontal drill is not in use. Parameter range: 0 - 100.	0
SawNo1	Set the first blade number. When set to 0, it indicates that the blade tool is not in use. Parameter range: 0 - 100.	0
SaeCnt	Set the number of blades. When set to 0, it indicates that the blade is not in use. Parameter range: 0 - 100.	0

Tool length and tool radius

For tool length and tool radius compensation, you can enter the parameter setting interface with Quick Start to set the compensation value of each tool from T1 to T100, and use with G-codes of G41 - G43 to enable the tool length / tool radius compensation function.

Parameter	Description	Default
SNC T_n Length	Set the tool length for each tool. (Unit: mm)	0
SNC T_n Radius	Set the tool radius for each tool. (Unit: mm)	0

Tool offset

For tool offset compensation, you can enter the parameter setting interface through Quick Start to set the offset compensation value for each tool.

Note: tool offset compensation is not valid in the G53 machine coordinate system.

Parameter	Description	Default
T_n offset X	Set the X-axis offset length of each tool. (Unit: mm)	0
T_n offset Y	Set the Y-axis offset length of each tool. (Unit: mm)	0
T_n offset Z	Set the Z-axis offset length of each tool. (Unit: mm)	0

Tool setter

The following tool length measurement function is performed by the servo drive, which is executed when the tool touches the tool setter and triggers the servo drive to capture the exact length of the current tool. You can set the tool setter parameters through the Quick Start interface. For the more function descriptions, refer to Section 9.4.2. (Only ASD - **** - A2-F series servo drives support the tool length measurement function.)

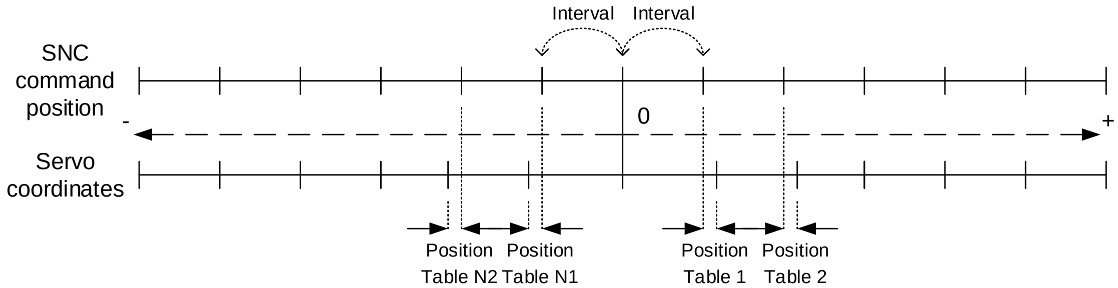
Parameter	Description	Default
ToolGauge_Interval	Set the distance between the tool setter plane and the working plane. (Unit: mm)	0
ToolGauge_SensorType	Set the contact type for the automatic tool setter. Settings: 0: contact A (NO); 1: contact B (NC).	0
ToolGauge_X	Set the absolute coordinate (X-axis) of the starting point for the tool length measurement function. (Unit: mm) The machine will move to this X coordinate before the tool measurement starts.	0
ToolGauge_Y	Set the absolute coordinate (Y-axis) of the starting point for the tool length measurement function. (Unit: mm) The machine will move to this Y coordinate before the tool measurement starts.	0
ToolGauge_Z	Set the absolute coordinate (Z-axis) of the starting point for the tool length measurement function. (Unit: mm) The machine will move to this Z coordinate before the tool measurement starts.	0
ToolGauge_1Down_Speed	Set the first descent speed of the Z axis during the tool measurement process. (Unit: mm/min) > 0: Z forward movement measurement. < 0: Z reverse movement measurement.	0
ToolGauge_2Down_Speed	Set the second descent speed of the Z axis during the tool measurement process. (Unit: mm/min) > 0: Z forward movement measurement. < 0: Z reverse movement measurement.	0
ToolGauge_Up_Speed	Set the ascent speed of the Z axis during the tool measurement process. (Unit: mm/min) > 0: Z forward movement measurement. < 0: Z reverse movement measurement.	0

9

9.2.2 Linear error compensation

Pitch error compensation

You can set the pitch error compensation of each axis to achieve better machining precision. The SNC compensates for the line within each effective distance based on the setting of each defined error distance. You can set the pitch error compensation parameters through the Quick Start interface. As shown in the figure below, the pitch error compensation can compensate 100 points in positive and negative values. Although G-code has already issued the positioning command, the machining precision is not accurate due to the pitch error. In this case, you can measure the actual mechanism position and command error of each distance with the proper measurement and use the pitch compensation to improve the machining precision.



Parameter	Description	Default
SNC_PEF_AxisX_Enable(200001) - SNC_PEF_AxisW_Enable(200009)	Function: enable the pitch error compensation function for each axis. Setting value: 0: disable the pitch error compensation function. 1: enable the pitch error compensation function.	0
SNC_PEF_AxisX_Interval(200011) - SNC_PEF_AxisW_Interval(200019)	Function: set the point-to-point distance of the pitch error compensation for each axis. (Unit: mm)	0
SNC_PEF_Position_Table_N100 - SNC_PEF_Position_Table_N1	Function: set the compensation value of each point for the negative coordinate pitch error. You can set 100 points for each axis. (Unit: mm)	0
SNC_PEF_Position_Table_1 - SNC_PEF_Position_Table_100	Function: set the compensation value of each point for the positive coordinate pitch error. You can set 100 points for each axis. (Unit: mm)	0

Backlash error compensation

You can set the reverse backlash compensation of each axis to achieve better machining precision. SNC compensates according to the change of the motion direction. You can set the backlash error compensation parameters through the Quick Start interface.

Parameter	Description	Default
SNC_AxisX_Backlash_Enable(601) - SNC_AxisZ_Backlash_Enable(609)	Function: enable the backlash error compensation function for each axis. Setting value: 0: disable the backlash error compensation function. 1: enable the backlash error compensation function.	0
SNC_AxisX_Backlash_Value(611) - SNC_AxisZ_Backlash_Value(619)	Function: set the backlash error compensation value for each axis. (Unit: mm)	0
SNC_AxisX_Backlash_Dir(621) - SNC_AxisZ_Backlash_Dir(629)	Function: set the backlash error compensation direction. Setting value: 1: reverse compensation. -1: forward compensation.	1
SNC_AxisX_Backlash_Speed(631) - SNC_AxisZ_Backlash_Speed(639)	Function: set the backlash error compensation speed. (Unit: mm/min)	0
SNC_AxisX_Backlash_Acc(641) - SNC_AxisZ_Backlash_Acc(649)	Function: set the backlash error compensation acceleration speed. (Unit: mm/s^2)	0

9.2.3 SNC related settings

Hardware configuration

9

The following parameters describe the number of motion axes used by the SNC and the settings of the servo drive card number / station number for each axis. All of the following parameters must match the servo settings. For example, if there is a three-axis motion to be controlled by the SNC, you must set SNC_Axes to 3.

Parameter	Description	Default
Groupnum	Function: total number of groups used by the SNC. Setting value: 1 - 4.	1
SNC_Axes(1000)	Function: set the total number of axes used by the SNC. Setting value: 0 - 9	0
SNC_Card_NO(1011)	Function: set the motion card number used by the SNC. DMCNET bus setting value: 0 - 15. EtherCAT bus setting value: 0 - 16.	0
SNC_AxisX_Node(1021) - SNC_AxisW_Node(1029)	Function: set the servo drive station number used for each axis of the SNC. Use the DMCNET bus setting value: 0, -1: disabled; 1 - 12: corresponding station number. Use the EtherCAT bus setting value: -1: disabled; 0 - 100: corresponding station number.	-1

9.2.3.1 Look ahead

The look ahead function is a very important function in the NC application. Its settings are more complex, mainly providing SNC to control the motion path and working speed. In general, the SNC will maintain the set feed rate in each line of G-code instructions. To execute this function, the SNC will mix the end and start speeds between G-codes. Increasing the speed mixing ratio will result in a smoother path and speed, but the precision of each turn will be compromised. On the other hand, reducing the speed mixing ratio will improve path precision, but the significant acceleration changes will cause machine vibration. Therefore, different types of machines are suitable for different look ahead settings. You can set the parameters by referring to this section.

- Model description

Calculate the maximum speed limit of the corner when executing the arc command.

$$\text{Maximum corner speed} = \sqrt{\frac{R}{R_{ref}}} \times ArcSpeed_{ref}$$

R_{ref} = SNC_Look_Ahead_Mode1_Arc_Radius (852).

$ArcSpeed_{ref}$ = SNC_Look_Ahead_Mode1_Arc_Speed (853).

Example:

When R_{ref} and $ArcSpeed_{ref}$ use the system default values of 2 and 600, the NC code execution program calculates as follows:

G90

G00 X0 Y0 F200

G01 X20 Y30 F200

G01 X50 Y60

G02 X70 Y40 R20 F200 ➔ (Radius R = 20)

G01 X40 Y10

G02 X20 Y30 R20 ➔ (Radius R = 20)

When the radius is 20, the maximum feed rate limit is $\sqrt{\frac{20}{2}} \times 600 \cong 1897.3666$. The set feed rates of G02 and G03 will be limited by 1897.366 mm/sec, as the outer trajectory shown in Figure 9.2.3.1; the inner trajectory is the motion trajectory that is not limited by the maximum corner speed.

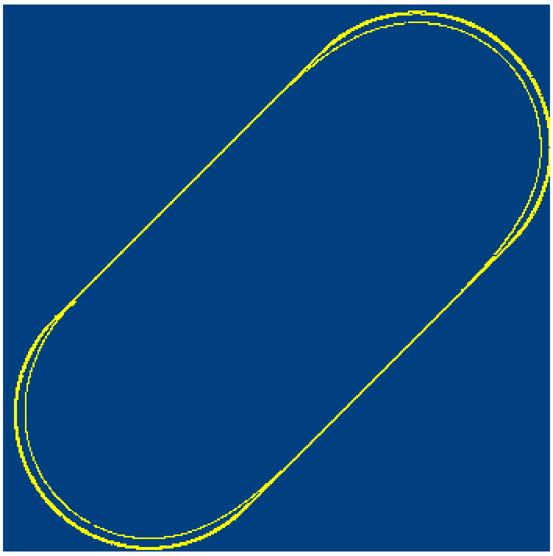


Figure 9.2.3.1

9

Corner speed limit:

SNC uses this calculated value to reduce the machine vibration at the turning point between two motion commands, including the motion commands of G01, G02, and G03.

$$\text{Reference corner feed rate} = \frac{Speed_{ref}}{2\sin(\frac{Theta}{2})}$$

$Speed_{ref}$ = SNC_Look_Ahead_Mode1_Turn_On_Speed. (854)

Example:

When $Speed_{ref}$ uses the system default parameter, the NC code execution program is as follows.

G90

G01 X0 Y40 F50000

G01 X40 Y40 ➔ (The path is at an angle of 90 degrees)

G01 X40 Y0 ➔ (The path is at an angle of 90 degrees)

G01 X0 Y0 ➔ (The path is at an angle of 90 degrees)

Maximum feed rate limit: $\frac{2000}{2\times\sin(\frac{90}{2})} \cong 1414.214$

The maximum speed between two instructions is limited to 1414.214 mm/sec (as the outer trajectory in Figure 9.2.3.2). When the set value of $Speed_{ref}$ is bigger, the maximum speed between the two instructions is not limited, which will cause the trajectory to deviate (as the inner trajectory in Figure 9.2.3.2).

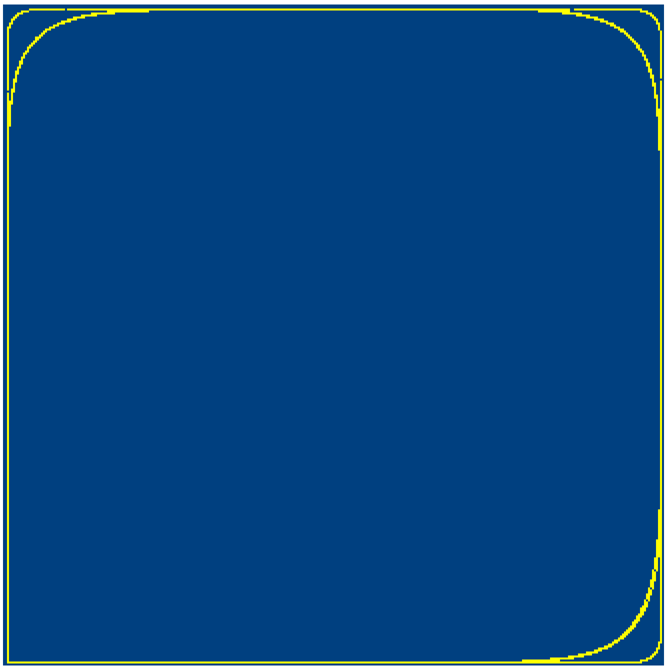


Figure 9.2.3.2

Figure 9.2.3.3 displays the path smoothing function provided by the SNC. By setting the filter parameters of SNC_Look_Ahead_Mode1_Motion_Scurve_Time (855), SNC_Look_Ahead_Mode1_DDA_Filter_Time (856), and SNC_Look_Ahead_Mode1_DDA_Scurve_Time (857), the path will become smoother.

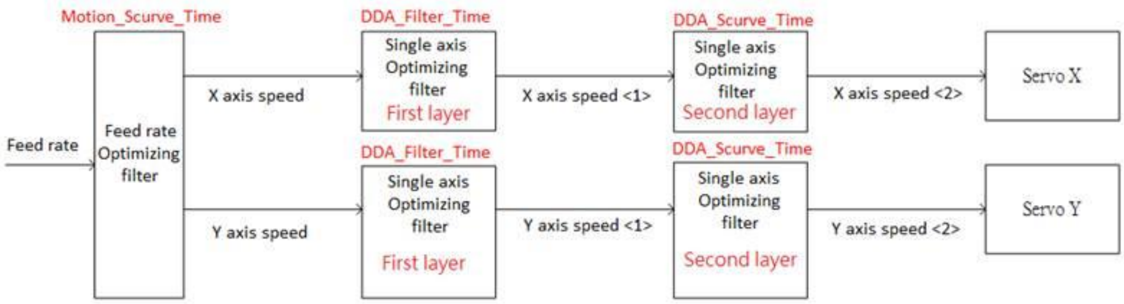


Figure 9.2.3.3

1. SNC_Look_Ahead_Mode1_Motion_Scurve_Time (855)

This parameter sets the path smoothing function to milliseconds as the unit. The path is smoothed by interpolation and the longer the interpolator processing time, the smoother the working path will be.

Example:

When SNC_Look_Ahead_Mode1_Motion_Scurve_Time (855) is set to the default value of 0.1, based on the communication period of 1 ms, SNC will average the last 99 path instructions with the current instruction. As shown in Figure 9.2.3.4, the path smoothing function is disabled for the blue curve, parameter 855 is set to 0; the red curve has parameter 855 set to 0.1 second. You can see the red curve is smoother than the blue curve and the overall acceleration time is extended by 100 ms.

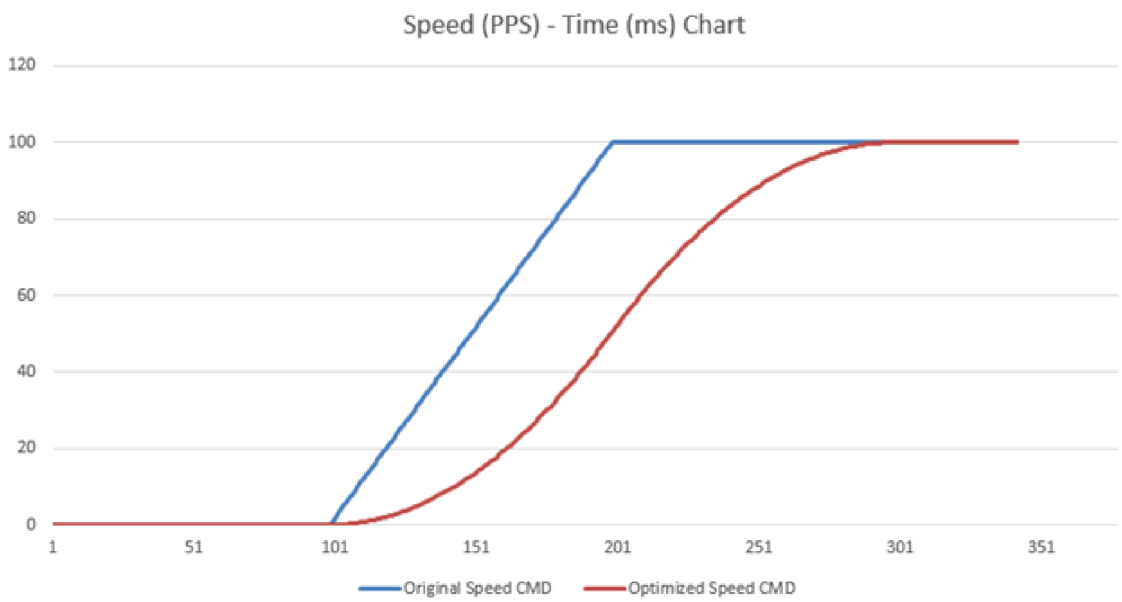


Figure 9.2.3.4

2. SNC_Look_Ahead_Mode1_DDA_Filter_Time (856)

After the SNC motion path is calculated by SNC_Look_Ahead_Mode1_Motion_Scurve_Time (855), the combined motion trajectory is separated into motion commands for each axis. This parameter is set to the smoothing time, which after the separation, the motion commands of each axis are averaged within the set time to achieve the smoothing speed.

3. SNC_Look_Ahead_Mode1_DDA_Scurve_Time (857)

After the SNC smoothes the motion path of each axis by SNC_Look_Ahead_Mode1_DDA_Filter_Time (856), the motion speed of each axis is smoothed for a second time by this filtering time.

Note:

You can try using the IMP default parameters of <855>, <856>, and <857>. But if you want a smoother path, you can gradually increase these parameter values; otherwise, reducing these parameter values can achieve a more precise motion path.

Since the use of the filter parameters will increase the acceleration and deceleration times, you can calculate the acceleration time from zero speed to the constant speed of each axis. Refer to the following example:

When the NC code defines the motion feed rate as 5000 mm/s (F5000) and the SNC maximum acceleration time parameter SNC_AxisX_PermitMaxACC (991) is set to 8333.333 mm/s^2:

$$v_1 = v_0 + a \times t$$

$$v_1 = 50000 \quad mm/min \cong 833.333\, mm/s$$

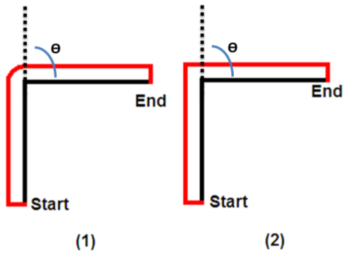
$$v_0 = 0$$

$$\mathrm{Acc} = 8333.333 \quad mm/s^2$$

Acceleration time $\mathrm{t} = 100\mathrm{ms} = 0.1s$

According to the default parameter values of <855>, <856>, and <857>, the total acceleration time of the X axis from zero speed to constant speed F5000 is ➔ 100 ms + <855> + <856> + <867> = 100 ms + 100 ms + 200 ms + 100 ms = 0.5 ms.

■ Error control

Parameter	Description	Default
SNC_Tolerance(860)	Set the tolerance for continuous cutting. (Unit: mm)	0.01
SNC_Circle_Tolerance(867)	Function: when G02 and G03 arc cutting are set, G-code is allowed to input the deviation of the center and the actual operation. (Unit: mm)	0.001
SNC_Kerf_Permit_Angle(886)	Function: set the determining angle for using arc or linear interpolation when the correction tool radius encounters cornering. (Unit: degree)  1. When the angle between the two lines (θ) > SNC_Kerf_Permit_Angle, use the path behavior in Figure (1). 2. When the angle between the two lines (θ) < SNC_Kerf_Permit_Angle, use the path behavior in Figure (2).	179

■ Speed smoothing

To achieve the best performance of each machine and produce high quality processed products, SNC must automatically optimize the working path, speed, acceleration, deceleration, etc.

The following parameters can assist you in adjusting the SNC for optimizing the process.

Parameter	Description	Default
SNC_Look_Ahead_Mode1_Arc_Radius (852)	The reference radius of the arc instruction (G02 or G03). (Unit: mm)	2
SNC_Look_Ahead_Mode1_Arc_Speed (853)	The reference speed limit of the arc instruction (G02 or G03).	600
SNC_Look_Ahead_Mode1_Turn_On_Speed (854)	The reference corner speed limit at the corner between two motion commands (G01 and G02 / G03).	1000
SNC_Look_Ahead_Mode1_Motion_Scurve_Time (855)	The S-curve control time of the motion command acceleration / deceleration time. (Unit: second) G00 does not support this S-curve feature.	0.1
SNC_Look_Ahead_Mode1_DDA_Filter_Time (856)	The filtering time for the first smoothing control of the working path. (Unit: second)	0.05
SNC_Look_Ahead_Mode1_DDA_Scurve_Time (857)	The filtering time for the second smoothing control of the working path. (Unit: second)	0.03

9

9.2.3.2 Speed setting

■ Rapid positioning

Parameter	Description	Default
SNC_Feed_Rate_G00(1171)	The SNC_G00_Use_Non_Line(870) parameter must be set to 1 for the setting of the G00 default feed rate to be valid.	6000
SNC_Tacc_G00(1172)	The SNC_G00_Use_Non_Line(870) parameter must be set to 1 for the setting of the G00 acceleration to be valid. (Unit: mm/s^2)	500
SNC_Tdec_G00(1173)	The SNC_G00_Use_Non_Line(870) parameter must be set to 1 for the setting of the G00 deceleration to be valid. (Unit: mm/s^2)	500

■ Processing speed

Parameter	Description	Default
SNC_Feed_Rate_G01_ Default(1183)	Set the default feed rate of G01. (Unit: mm/min)	6000
SNC_Feed_Rate_G01(1174)	Set the feed rate upper limit of G01. (Unit: mm/min)	6000
SNC_Feed_Rate_G01(1175)	Set the combined feed acceleration upper limit of G01. (Unit: mm/min)	500
SNC_Feed_Rate_G01(1176)	Set the combined feed deceleration upper limit of G01. (Unit: mm/min)	500
SNC_Feed_Rate_Circle _Default(1184)	Set the default feed rates of G02 and G03. (Unit: mm/min)	6000
SNC_Feed_Rate_Circle(1177)	Set the feed rate upper limits of G02 and G03. (Unit: mm/min)	6000
SNC_Tacc_Circle(1178)	Set the combined feed acceleration upper limits of G02 and G03. (Unit: mm/min)	500
SNC_Tdec_Circle (1179)	Set the combined feed deceleration upper limits of G02 and G03. (Unit: mm/min)	500
SNC_AxisX_PermitMaxACC (991) - SNC_AxisW_PermitMaxACC (999)	Set the acceleration and deceleration of each axis. (Unit: mm/s^2)	500

9.2.3.3 Speed limit

Shared parameters

Parameter	Description	Default
SNC_Use_LimitSpeed(980)	Speed limit function. 0: disable (default). 1: enable; use the set value of the SNC_Axis_LimitSpeed parameter. 2: enable; G00 uses the set value of the SNC_Axis_G00_LimitSpeed parameter, and G01, G02, and G03 use the set value of the SNC_Axis_LimitSpeed parameter.	0
SNC_AxisX_LimitSpeed(971) - SNC_AxisW_LimitSpeed(979)	Function: set the processing speed limit for each axis. (Unit: mm/min) When SNC_Use_LimitSpeed is set to 1, this parameter limits the maximum speed of each axis. When SNC_Use_LimitSpeed is set to 2, this parameter limits all the speed motions other than G00.	0
SNC_AxisX_G00_LimitSpeed (671) - SNC_AxisW_G00_LimitSpeed (679)	Function: set the G00 rapid feed rate limit for each axis. (Unit: mm/min) This parameter is valid when SNC_Use_LimitSpeed is set to 2.	0

9.2.3.4 Special function

Parameter	Description	Default
SNC_User_Scan_Mcode(704)	Function: set to check the macros (M, T, S, and H) during NC code pre-scanning before the SNC starts. 0: disable (default). 1: enable.	0
SNC_Keep_Sharp_Variables (705)	Set to save #variable contents of the subprogram. 0: #variable contents will be cleared when SNC is executed (default). 1: save.	1
SNC_Different_Work_Plane (708)	Set to allow the main program and subprogram to use different working planes. 0: disable (default). 1: enable.	1
SNC_Allow_kerf_With_M_ Code(709)	Allow executing M-code during tool radius compensation, but not calling external macros. 0: disable (default). 1: enable.	1
SNC_Alwasy_Check_Axis_ Alarm(810)	Set to check for errors occurred in the software limit, hardware limit, and servo drive even when SNC is not processing, and generate SNC error messages. 0: disable (default). 1: enable.	0
SNC_Reverse(811)	Set the path reverse function to be enabled when using the MPG reversing and path reversing functions. This function will lower the PC performance and occupy memory space, so it is recommended to turn it off when not in use. 0: disable (default). 1: enable.	0
SNC_Work_Plane(1500)	Set the default coordinates of the working plane. Range: 54 - 59, 59.1 - 59.9.	54
SNC_Code_Work_Plane_ Macro1(717)	Set the default coordinates of the working plane for the macro. Range: 54 - 59, 59.1 - 59.9.	54

9.2.3.5 System record

This parameter is used for testing. Please turn it off when not testing to lower system load.

Parameter	Description	Default
SNC_Dump_Log(821)	Record the motion command to C:\ motion_record.txt. This feature is used by developers to adapt the SNC motion status. 0: disable (default). 1: enable.	0
SNC_Dump_Log_Macro(830)	Record the M-code operation by the SNC. This feature is used by developers to adapt the SNC motion status. 0: disable (default). 1: enable.	0
SNC_Record_Enable(823)	Record the position coordinates of each communication cycle during the SNC operation in C:\ dda_cmd_record_Look_Ahead_1_6.txt. This feature is used by developers to adapt the SNC motion status. 0: disable (default). 1: enable.	0

9.3 SNC interpreter

9.3.1 G-code supporting table

Code	Description	Support	Code	Description	Support
G00	Linear rapid positioning	☐	G50.1	Cancel mirror image	☐
G01	Cutting feed	☐	G51.1	Enable mirror image	☐
G02	Clockwise arc cutting	☐	G52	Local coordinate setting (offset)	☐
G03	Counterclockwise arc cutting	☐	G53	Machine coordinate positioning	☐
G04	Pause	☐	G54 - G59	Workpiece coordinate system setting	☐
G09	Exact positioning	☐	G61	Exact positioning mode	☐
G17	X-Y plane selection	☐	G64	General cutting mode	☐
G18	Z-X plane selection	☐	G65	Single macro program call	☐
G19	Y-Z plane selection	☐	G66	Macro program call	☐
G20	Apply inch as the unit		G67	Disable macro mode	☐
G21	Apply mm as the unit		G68	Coordinate rotation	☐
G28	Return through reference point	☐	G69	Cancel coordinate rotation	☐
G29	Return from reference point	☐	G73	High speed peck drilling cycle	☐
G30	Return to any reference point	☐	G80	Cancel cycle	☐
G40	Cancel tool radius compensation	☐	G81	Drilling cycle	☐
G41	Tool radius compensation left [*1]	☐	G82	Pause drilling cycle at the bottom of the hole	☐
G42	Tool radius compensation right [*1]	☐	G83	Peck drilling cycle	☐
G43	Tool length compensation (+)[*2]	☐	G85	Boring cycle	☐
G44	Tool length compensation (-)[*2]	☐	G89	Pause boring cycle at the bottom of the hole	☐
G49	Disable tool length compensation	☐	G90	Absolute designation	☐
G50	Disable scale function	☐	G91	Incremental designation	☐
G51	Enable scale function	☐	-	-	-

☐: standard support.

9.3.2 M-code and T-code

The function description of the defined M-code interpreted by SNC is detailed as below:

Code	Description
M02	Function: stop the system. Description: end the program and stop the system.
M30	Function: stop the system and the program indicator returns to the starting position. Description: end the program and stop the system.
M98	Function I: subprogram control. For fixed path processing or commonly used functions, when O0000.NC - O9999.NC files are put in the system folder D:\NandFlash\IPC Motion Platform\IMP base\SNC_Macro, M98 command can be used to call the files. Description: the programming format of M98: M98 P__ L__. P: the file code to be called (if you input P0000, O0000.NC will be called). L: the number of times to execute the subprogram. Function II: call the subprogram in the NC code. Description: the programming format of M98: M98 H__ L__. H: the subprogram code to be called (if you input H0000, N0000 will be called). L: the number of times to execute the subprogram. NC code example: G54//Set the workpiece coordinate system to G54 G00 X0 Y0//Rapid positioning M98 H50 L2//Execute the N000 subprogram twice M30//Program ends N50//Subprogram code G01 X10//Linear interpolation Y20 X20 Y0 M99//Subprogram ends
M99	Function I: cycling. Description: when the G-code main program encounters M99, it returns to the first line of the main program. Function II: end the subprogram. Description: when using M99 in the subprogram, the subprogram will end and return to the main program.

Custom M-code and T-code

When running into undefined M-code and T-code, SNC will enable the corresponding R relay (listed in the table below), and release control. Once PLC completes the corresponding motion and finishes clearing the flag, SNC will retrieve the control and continue interpreting G-code. The spindle tool number is set in the tool setting parameter (Section 9.4.4). If T-code is not the spindle tool number, the system will compensate the offset between the tool position and the spindle.

M-code and T-code command flag

	SNC1 memory location	SNC2 memory location	SNC3 memory location	SNC4 memory location
T1 - T100	R32001 - R32100	R34001 - R34100	R36001 - R36100	R38001 - R38100
M0 - M999	R31000 - R31999	R33000 - R33999	R35000 - R35999	R37000 - R37999

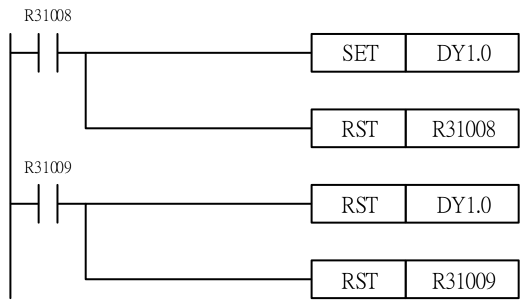
Program example:

This example defines M08 as enabling cutting oil, M09 as disabling cutting oil, and DY1.0 as the cutting oil output pump. In the G54 workpiece plane, enable the cutting oil to cut a square with a length and width of 20 mm.

NC code:

```
        G54
        G00 Z0
        G00 X20 Y20
        M08//Enable cutting oil
        G01 Z-20
        G01 X0
        G01 Y0
        G01 X20
        G01 Y20
        G00 Z0
        M09//Disable cutting oil
M30//Main program ends
```

PLC program:



9.3.3 Definitions of SNC variables

List of variables

Number	Name	Description
#0 - #25	Local variable	When executing G-code or calling an external macro, this variable section exists independently in each program and does not affect each other.
#26 - #1800 #1913 - #49999	SNC global variable	When executing G-code or calling an external macro, this variable section is interlinked until the memory is cleared at the end of the execution of the main G-code. [Note]
#1801 - #1832	PLC bit output	R32301 > #1801 (1-bit *32).
#1833 - #1848	PLC word output	W32301 > #1833 (16-bit *16).
#1864 - #1895	PLC bit input	R32334 < #1864 (1-bit *32).
#1896 - #1912	PLC word input	W32317 < #1896 (16-bit *16).
#2000 - #3999	System status area	SNC operation related message area.
@0 - 49999	Non-volatile type global variable	When executing G-code or calling an external macro, you can read / write this variable section until the memory data is cleared when the IMP is off.

Note: when the SNC_Keep_Sharp_Variables(705) parameter is set to 1, the variable # memory will be saved until the IMP is off.

Argument and local variable

In addition to G, L, N, O, and P, other variable codes can be used as designated arguments for data transfer of the local variable when using G65 or G66 macro subprogram calls.

Variable code	Local variable	Variable code	Local variable	Variable code	Local variable
A	#0	I	#3	V	#21
B	#1	J	#4	W	#22
C	#2	K	#5	X	#23
D	#6	M	#12	Y	#24
E	#7	Q	#16	Z	#25
F	#8	R	#17	-	-
H	#10	U	#20	-	-

System variable

G65 and G66 call the subprogram, and bring in the argument.

No.	Description	Property
#0 - #25	Local variable	R/W

Use T-code to call the submacro.

No.	Description	Property
#2021	H-code	R
#2023	T-code	R
#2024	S-code	R
#2304	Tool number on spindle 1	R
#2500	Retrieve tool number on magazine 1	R

SNC operation status area.

No.	Description	Property
#3000	Current workpiece coordinate plane	R
#3001	X-axis workpiece coordinate	R
#3002	Y-axis workpiece coordinate	R
#3003	Z-axis workpiece coordinate	R
#3004	A-axis workpiece coordinate	R
#3005	B-axis workpiece coordinate	R
#3006	C-axis workpiece coordinate	R
#3007	U-axis workpiece coordinate	R
#3008	V-axis workpiece coordinate	R
#3009	W-axis workpiece coordinate	R
#3011	X-axis machine coordinate	R
#3012	Y-axis machine coordinate	R
#3013	Z-axis machine coordinate	R
#3014	A-axis machine coordinate	R
#3015	B-axis machine coordinate	R
#3016	C-axis machine coordinate	R
#3017	U-axis machine coordinate	R
#3018	V-axis machine coordinate	R
#3019	W-axis machine coordinate	R
#3021	X-axis relative coordinate	R
#3022	Y-axis relative coordinate	R
#3023	Z-axis relative coordinate	R
#3024	A-axis relative coordinate	R
#3025	B-axis relative coordinate	R
#3026	C-axis relative coordinate	R
#3027	U-axis relative coordinate	R
#3028	V-axis relative coordinate	R
#3029	W-axis relative coordinate	R
#3031	X-axis remaining distance	R
#3032	Y-axis remaining distance	R
#3033	Z-axis remaining distance	R
#3034	A-axis remaining distance	R
#3035	B-axis remaining distance	R
#3036	C-axis remaining distance	R
#3037	U-axis remaining distance	R
#3038	V-axis remaining distance	R
#3039	W-axis remaining distance	R

Macro interface output / input

By using the variable numbers #1801 - #1911, you can know the interface information in the program, and read or write the MLC signal status. The value can be Bit or Word. When the signal type is Bit, the variable value is only 1 or 0; when the signal type is Word, the variable value can be any value.

■ PLC bit input, write to SNC signal status (PLC > NC).

Name	Property	SNC	SNC 2	SNC 3	SNC4
Write to #1801	R/W	R32301	R34301	R36301	R38301
Write to #1802	R/W	R32302	R34302	R36302	R38302
Write to #1803	R/W	R32303	R34303	R36303	R38303
Write to #1804	R/W	R32304	R34304	R36304	R38304
Write to #1805	R/W	R32305	R34305	R36305	R38305
Write to #1806	R/W	R32306	R34306	R36306	R38306
Write to #1807	R/W	R32307	R34307	R36307	R38307
Write to #1808	R/W	R32308	R34308	R36308	R38308
Write to #1809	R/W	R32309	R34309	R36309	R38309
Write to #1810	R/W	R32310	R34310	R36310	R38310
Write to #1811	R/W	R32311	R34311	R36311	R38311
Write to #1812	R/W	R32312	R34312	R36312	R38312
Write to #1813	R/W	R32313	R34313	R36313	R38313
Write to #1814	R/W	R32314	R34314	R36314	R38314
Write to #1815	R/W	R32315	R34315	R36315	R38315
Write to #1816	R/W	R32316	R34316	R36316	R38316
Write to #1817	R/W	R32317	R34317	R36317	R38317
Write to #1818	R/W	R32318	R34318	R36318	R38318
Write to #1819	R/W	R32319	R34319	R36319	R38319
Write to #1820	R/W	R32320	R34320	R36320	R38320
Write to #1821	R/W	R32321	R34321	R36321	R38321
Write to #1822	R/W	R32322	R34322	R36322	R38322
Write to #1823	R/W	R32323	R34323	R36323	R38323
Write to #1824	R/W	R32324	R34324	R36324	R38324
Write to #1825	R/W	R32325	R34325	R36325	R38325
Write to #1826	R/W	R32326	R34326	R36326	R38326
Write to #1827	R/W	R32327	R34327	R36327	R38327
Write to #1828	R/W	R32328	R34328	R36328	R38328
Write to #1829	R/W	R32329	R34329	R36329	R38329
Write to #1830	R/W	R32330	R34330	R36330	R38330
Write to #1831	R/W	R32331	R34331	R36331	R38331
Write to #1832	R/W	R32332	R34332	R36332	R38332

■ PLC bit output, read SNC signal status (NC > PLC).

Name	Property	SNC	SNC 2	SNC 3	SNC4
Read #1864	R	R32334	R34334	R36334	R38334
Read #1865	R	R32335	R34335	R36335	R38335
Read #1866	R	R32336	R34336	R36336	R38336
Read #1867	R	R32337	R34337	R36337	R38337
Read #1868	R	R32338	R34338	R36338	R38338
Read #1869	R	R32339	R34339	R36339	R38339
Read #1870	R	R32340	R34340	R36340	R38340
Read #1871	R	R32341	R34341	R36341	R38341
Read #1872	R	R32342	R34342	R36342	R38342
Read #1873	R	R32343	R34343	R36343	R38343
Read #1841	R	R32344	R34344	R36344	R38344
Read #1875	R	R32345	R34345	R36345	R38345
Read #1876	R	R32346	R34346	R36346	R38346
Read #1877	R	R32347	R34347	R36347	R38347
Read #1878	R	R32348	R34348	R36348	R38348
Read #1879	R	R32349	R34349	R36349	R38349
Read #1880	R	R32350	R34350	R36350	R38350
Read #1881	R	R32351	R34351	R36351	R38351
Read #1882	R	R32352	R34352	R36352	R38352
Read #1883	R	R32353	R34353	R36353	R38353
Read #1884	R	R32354	R34354	R36354	R38354
Read #1885	R	R32355	R34355	R36355	R38355
Read #1886	R	R32356	R34356	R36356	R38356
Read #1887	R	R32357	R34357	R36357	R38357
Read #1888	R	R32358	R34358	R36358	R38358
Read #1889	R	R32359	R34359	R36359	R38359
Read #1890	R	R32360	R34360	R36360	R38360
Read #1891	R	R32361	R34361	R36361	R38361
Read #1892	R	R32362	R34362	R36362	R38362
Read #1893	R	R32363	R34363	R36363	R38363
Read #1894	R	R32364	R34364	R36364	R38364
Read #1895	R	R32365	R34365	R36365	R38365

■ PLC word input, write to SNC signal status (PLC > NC).

Name	Property	SNC	SNC 2	SNC 3	SNC4
Write to #1833	R/W	W32301	W34301	W36301	W38301
Write to #1834	R/W	W32302	W34302	W36302	W38302
Write to #1835	R/W	W32303	W34303	W36303	W38303
Write to #1836	R/W	W32304	W34304	W36304	W38304
Write to #1837	R/W	W32305	W34305	W36305	W38305
Write to #1838	R/W	W32306	W34306	W36306	W38306
Write to #1839	R/W	W32307	W34307	W36307	W38307
Write to #1840	R/W	W32308	W34308	W36308	W38308
Write to #1841	R/W	W32309	W34309	W36309	W38309
Write to #1842	R/W	W32310	W34310	W36310	W38310
Write to #1843	R/W	W32311	W34311	W36311	W38311
Write to #1844	R/W	W32312	W34312	W36312	W38312
Write to #1845	R/W	W32313	W34313	W36313	W38313
Write to #1846	R/W	W32314	W34314	W36314	W38314
Write to #1847	R/W	W32315	W34315	W36315	W38315
Write to #1848	R/W	W32316	W34316	W36316	W38316

■ PLC word output, read SNC signal status (NC > PLC).

Name	Property	SNC	SNC 2	SNC 3	SNC4
Read #1896	R	W32317	W34317	W36317	W38317
Read #1897	R	W32318	W34318	W36318	W38318
Read #1898	R	W32319	W34319	W36319	W38319
Read #1899	R	W32320	W34320	W36320	W38320
Read #1900	R	W32321	W34321	W36321	W38321
Read #1901	R	W32322	W34322	W36322	W38322
Read #1902	R	W32323	W34323	W36323	W38323
Read #1903	R	W32324	W34324	W36324	W38324
Read #1904	R	W32325	W34325	W36325	W38325
Read #1905	R	W32326	W34326	W36326	W38326
Read #1906	R	W32327	W34327	W36327	W38327
Read #1907	R	W32328	W34328	W36328	W38328
Read #1908	R	W32329	W34329	W36329	W38329
Read #1909	R	W32330	W34330	W36330	W38330
Read #1910	R	W32331	W34331	W36331	W38331
Read #1911	R	W32332	W34332	W36332	W38332

9.3.4 Macro syntax

NC supports # variables and expressions. This section provides details of the expressions and statements.

- Variable operation

Symbol	Usage	Definition
( )	#i = ABS(#k)	Brackets
-	#i = ACOS(#k)	Negative sign
+	#i = ASIN(#k)	Addition sign
-	#i = ATAN(#k)	Subtraction sign
*	#i = COS(#k)	Multiple sign
/	#i = SIN(#k)	Division sign

- Statement

Symbol	Usage	Definition
>,<,>=,<=	If (#k_1 > #k_2)	Comparison
=	If (#k_1 = #k_2	Equal to
<>	If (#k_1 <> #k_2)	Not equal to
NOT	if(#k_1=1and (not(#k_2=1)))then	Complement
AND	if(#k_1=1and#k_2=1)then	And
XOR	if(#k_1=1xor#k_2=1)then	Exclusive or
OR	if(#k_1=1or#k_2=1)then	Or

- Arithmetic command

Symbol	Usage	Definition
ABS	#i = ABS(#k)	Absolute value
ACOS	#i = ACOS(#k)	Arccosine
ASIN	#i = ASIN(#k)	Arcsine
ATAN	#i = ATAN(#k)	Arctangent
COS	#i = COS(#k)	Cosine
SIN	#i = SIN(#k)	Sine
TAN	#i = SIN(#k)	Tangent
SQRT	#i = SQRT(#k)	Square root value
ROUND	#i = ROUND(#k)	Rounding number

- Flow control

Conditional statement

When IF [statement] is fulfilled, the program flow executes the program from GOTO to program line number N. When IF [statement] is not fulfilled, the program flow will execute the next single block of the statement. Refer to the following description:

Example:

```
#1 = 1
#2 = 2
G90 G00 X0 Y0 Z0
IF (#1 > #2) THEN
     X100
ELSEIF (#1 = #2) THEN
     Y100
END_IF
```

Loop example

```
G90 G00 X0
#1 = 0
WHILE(#1 < 100)DO
     X#1
     #1 = #1 + 10
END_WHILE      // Program end position X90
```

Jump example

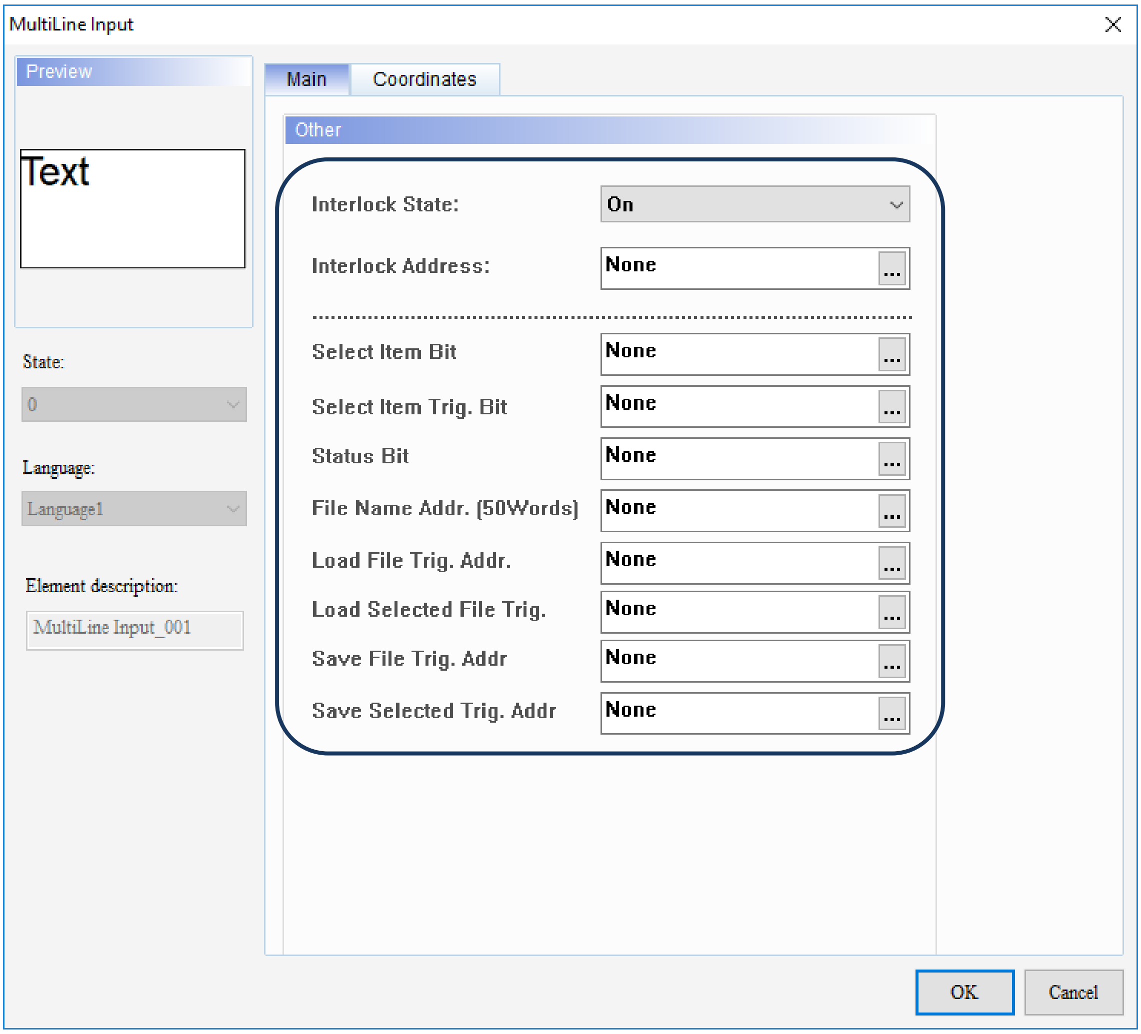
```
G90 G00 X0 Y0 Z0
GOTO1
X100
N1
Y100       // Program end position X0 Y100
```

9.4 Descriptions of SNC related functions

9.4.1 Accessing G-code

Descriptions of the G-code editing interface functions

IMP uses DOPSoft multi-line editor as the editing interface for G-code, providing interface designers with greater flexibility.



Select Item Bit: set the bit of the line to be selected.

Select Item Trig. Bit: set the trigger enable bit of the selected line.

Status Bit: show the returned value of the file opening status. For more details about the multi-line input operation status, refer to the following table.

Return value	Function description
1	Element in execution.
2	Cancel execution.
3	Execution completed.
4	Execution failed.
5	Failed to open file.
6	Failed to save file.
7	Successfully opened file.
8	Successfully saved file.

File Name Addr. (50 Words): record the storage address of current files. W31100 is the storage address for SNC1 path.

Load File Trig. Addr.: set the trigger address to read the file from the specified path.

Load Selected File Trig.: set the trigger address to open old files.

Save File Trig. Addr.: set the trigger address to save the file.

Save Selected Trig. Addr.: set the trigger address to create a new file.

Parameters for accessing G-code

Use SNC Group 1 as an example: (Refer to Appendix A for register address.)

W31100: SNC reads the file path of the G-code, and occupies 50 addresses and 100 bytes consecutively.

9.4.2 Automatic tool setting

Function description

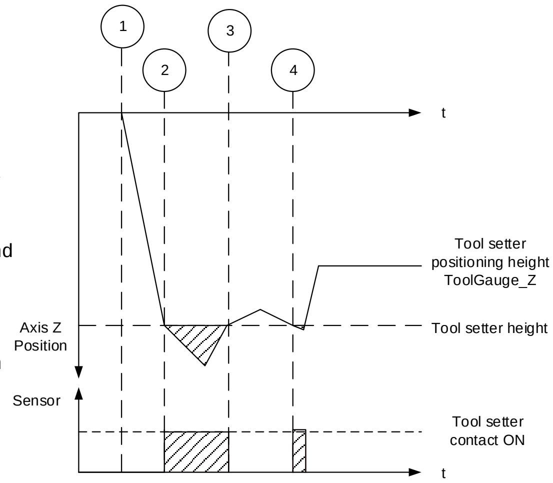
To achieve higher precision, you can execute the automatic tool setting function with the built-in PR mode in the servo drive to obtain the position where the tool touches the tool setter.

1: when the tool setting motion starts, the Z axis is positioned according to the parameter of ToolGauge_Z, then the X and Y axis positioning are completed. The servo is switched to the PR mode and moves down according to the set speed of ToolGauge_1Down_Speed.

2: when the tool first touches the tool setter, Axis Z decelerates and moves up.

3: when the tool leaves the tool setter, Axis Z decelerates and moves down at the second speed set by the user.

4: when the tool touches the tool setter the second time, the touch point is recorded. Axis Z then decelerates and moves up to the position set by ToolGauge_Z.



Note: PR mode is applicable to ASD-****-A2-F models.

Parameters

Use SNC Group 1 as an example: (Refer to Appendix A for register address.)

■ W31000

Function: activate SNC control code.

Setting value:

14: start tool setting program.

15: stop tool setting program.

Note: for more details about the tool setting parameters, refer to Section 9.2.1.

9.4.3 Single step mode

Function description

You can select G-code single step mode or continuous execution mode before starting the SNC. Single step mode executes G-code in the unit of lines, meaning a single line will be executed when the rising-edge is triggered each time.

Parameters

Use SNC Group 1 as an example: (Refer to Appendix A for register address.)

- R32981: enable SNC single step mode (only applicable before the SNC starts).
- R32982: trigger SNC single step execution.

9.4.4 Spindle control

Function description

The SNC spindle operation can be adjusted with the spindle speed (S) parameters. When the SNC interpreter executes M-code containing S-function parameters, SNC will save the S-function parameter values in a special register for the PLC to perform spindle speed adjustment.

Parameters

Use SNC Group 1 as an example: (Refer to Appendix A for register address.)

- W31020: after the SNC reads the S-function parameters, it will save the values in the register address.

9.4.5 Manual feed rate adjustment

Function description

You can manually control the SNC feed rate through the user interface or external switch.
When this function is enabled, the speed calculation formula is shown as below:
G-code original speed x SNC feed rate (0 - 100%) = SNC actual execution speed.

Parameters

Use SNC Group 1 as an example: (Refer to Appendix A for register address.)

- W19001: set the motion card No. of the ASD-DMC-RM64MN module that is connected to the MPG.
- W19002: set the station No. of the ASD-DMC-RM64MN module that is connected to the MPG.
- W19000: to use the MPG simulation mode, set the value to 2.
- R19001: set the scale value of each MPG rotating block and the output pulse rate (set value: ON: quadruple frequency; OFF: single frequency).
- R32997: enable the feed rate of SNC MPG control.
- W32480: set the feed rate of SNC manual control.

If R32997 is OFF, the feed rate is automatically set to 100% when the SNC is disabled.

9.4.6 MPG simulation mode

Function description

You can use the MPG rotation speed to simulate the SNC feed rate. The speed calculation formula is shown as below:

G-code original speed x MPG rotation percentage (0 - 100%) = SNC actual execution speed.

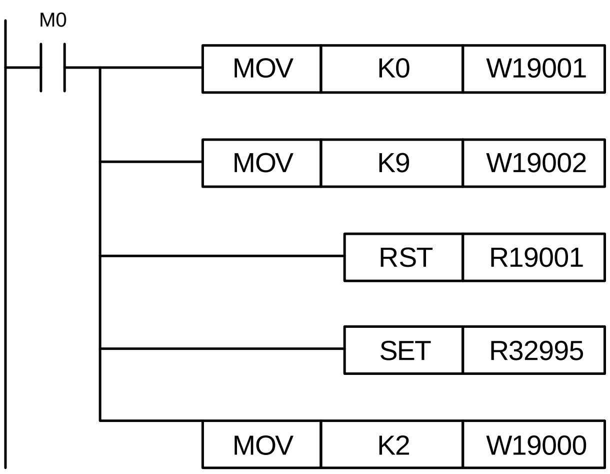
Parameters

Use SNC Group 1 as an example: (Refer to Appendix A for register address.)

- W19001: set the motion card No. of the ASD-DMC-RM64MN/R1-EC5614 module that is connected to the MPG.
  0 - 15: motion card No.
  16: PAC uses RTX as the EtherCAT master station.
- W19002: set the station No. of the ASD-DMC-RM64MN/R1-EC5614 module that is connected to the MPG.
  0 - 99: module station number (DMCNET supports 1 - 12, EtherCAT motion card supports 0 - 31, PAC uses RTX as the EtherCAT master station and supports 0 - 99).
  101: connect the MPG module with the motion card / PAC I/O.
- W19000: to use the MPG simulation mode, set the value to 2.
- R19001: set the scale value of each MPG rotating block and the output pulse rate (set value: ON: quadruple frequency; OFF: single frequency).
- R32995: enable the function for using the MPG to simulate the SNC1 feed rate.
  If this function is disabled, the feed rate W32480 is automatically set to 0.
- W32951: set the MPG simulation reverse speed (mm/min) which must be set before the SNC starts to be valid. The reverse path will ignore the feed rate set by G-code. To apply this function, you need to enable the reverse function.

Example

The following example uses ASD-DMC-RM64MC to connect the MPG module with the card No. as 0 and RM64MN station No. as 9.



9.4.7 External macro

Function description

Apart from using G-code to describe the motion path, the SNC also uses M-code for mechanical motion controlling and T-code for tool changing. In the IMP system, after G-code is interpreted, it is output directly through the fieldbus and executed by the motion unit. If it is M-code or T-code, the control will be transferred to the PLC which will determine the execution actions of the M-code or T-code with the ladder diagrams, and these actions include enabling or disabling I/O or using the external macro functions. External macro can be used to describe certain mechanical functions, including tool changing and origin motion.

Parameters

Use SNC Group 1 as an example: (Refer to Appendix A for register address.)

- W31015

Function: SNC calls the macro file control code.

Setting value:

0: no command or end the execution of the action.

1: call D:\NandFlash\IPC Motion Platform\IMP base\SNC_Macro\Oxxxx.nc macro.

2: call D:\NandFlash\IPC Motion Platform\IMP base\SNC_Macro\Txxxx.nc macro.

99: return value, file not found.

- W31016

Function: SNC calls the macro file No.

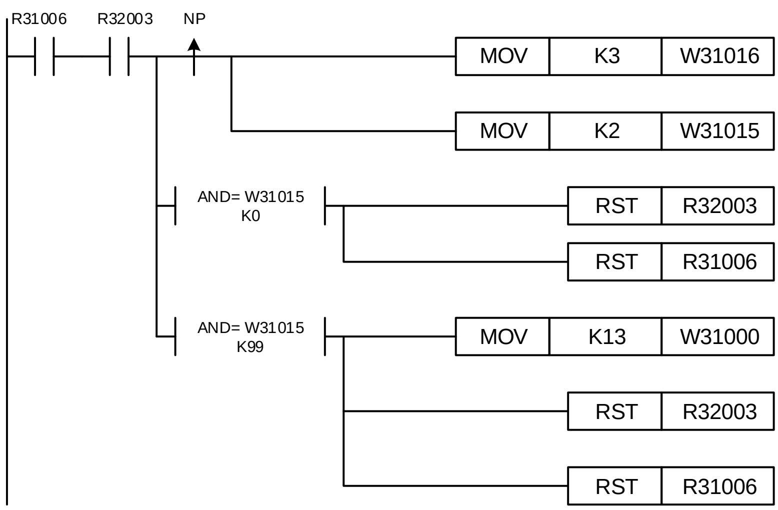
Range: 0000 - 9999.

Example

This example demonstrates M6 as the spindle changing to T3 as the spindle tool. When M6T3 is encountered in the NC code, the IMP SNC system sets R31006 and R32003 to ON, and calls the T0003.nc external macro through the PLC to perform motion.

NC code: PLC ladder diagram:

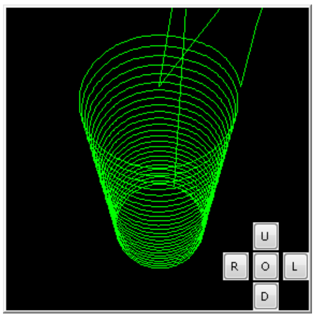
G54
G00 Z0
G00 X20 Y20
M6T3//Tool changing
G01 Z-20
G01 X0
G01 Y0
G01 X20
G01 Y20
G00 Z0
M30//Main program ends



9.4.8 G-code preview

Function description

You can use the G-code preview function to generate a path trajectory graphic.



Parameters

Use SNC Group 1 as an example: (Refer to Appendix A for register address.)

W32994: simulate the window width (pixels).

W32995: simulate the window height (pixels).

W32996: G-code in operation (0: G00, 1: G01, 2: G02, 3: G03).

W32997: simulation window displays the X coordinates (pixels).

W32998: simulation window displays the Y coordinates (pixels).

W32999: simulation window control (1: hide simulation window, 2: display simulation window, 3: clear window content, 4: close simulation window, 10: open simulation window).

9

9.5 List of SNC error codes

9.5.1 File and data error

Number	Name	Description
1	ERR_FILE_NOT_EXIST	The file does not exist when reading the G-code path.
2	ERR_NO_DATA	In the opened G-code file, the content has no processing string or content (the total number of lines is zero).
3	ERR_DATA_NOT_COMPLETE	The second parameter of API SNC_set_process_data is zero.
4	ERR_START_OVER	The starting line exceeds the total number of G-code lines.
5	ERR_DMC_01_DLL_Not_Full_Version	The SNC and DMCNET DLL versions are incompatible.
6	ERR_CUTTING_LINE_TOO_SHORT	Error / warning occurs when the cutting line is too short.
7	ERR_GOTO_LINE_WRONG	When using the goto command, there is no corresponding label.
8	ERR_GOTO_LINE_REDEFINED	When using the goto command, the label is duplicated.
9	ERR_GROUP_OUT_OF_RNG	The set value of the SNC_Append_Group parameter is out of range.
10	ERR_GROUP_SAME_GROUP	The set value of the SNC_Append_Group parameter is duplicated.
11	ERR_GROUP_NO_APPEND	The SNC group No. is not specified.
13	ERR_CREATE_THREAD_FAIL	SNC failed to create a thread.

9.5.2 Duplicate definition of the G-code (in the same line of G-code)

Number	Name	Description
101	ERR_GCODE_MULTIPLE_A_WORDS_ON_ONE_LINE	Duplicate definition of variable A.
102	ERR_GCODE_MULTIPLE_B_WORDS_ON_ONE_LINE	Duplicate definition of variable B.
103	ERR_GCODE_MULTIPLE_C_WORDS_ON_ONE_LINE	Duplicate definition of variable C.
104	ERR_GCODE_MULTIPLE_D_WORDS_ON_ONE_LINE	Duplicate definition of variable D.
105	ERR_GCODE_MULTIPLE_E_WORDS_ON_ONE_LINE	Duplicate definition of variable E.
106	ERR_GCODE_MULTIPLE_F_WORDS_ON_ONE_LINE	Duplicate definition of variable F.
107	ERR_GCODE_MULTIPLE_H_WORDS_ON_ONE_LINE	Duplicate definition of variable H.
108	ERR_GCODE_MULTIPLE_I_WORDS_ON_ONE_LINE	Duplicate definition of variable I.
109	ERR_GCODE_MULTIPLE_J_WORDS_ON_ONE_LINE	Duplicate definition of variable J.
110	ERR_GCODE_MULTIPLE_K_WORDS_ON_ONE_LINE	Duplicate definition of variable K.
111	ERR_GCODE_MULTIPLE_L_WORDS_ON_ONE_LINE	Duplicate definition of variable L.
112	ERR_GCODE_MULTIPLE_M_WORDS_ON_ONE_LINE	Duplicate definition of variable M.
113	ERR_GCODE_MULTIPLE_P_WORDS_ON_ONE_LINE	Duplicate definition of variable P.
114	ERR_GCODE_MULTIPLE_Q_WORDS_ON_ONE_LINE	Duplicate definition of variable Q.
115	ERR_GCODE_MULTIPLE_R_WORDS_ON_ONE_LINE	Duplicate definition of variable R.
116	ERR_GCODE_MULTIPLE_S_WORDS_ON_ONE_LINE	Duplicate definition of variable S.
117	ERR_GCODE_MULTIPLE_T_WORDS_ON_ONE_LINE	Duplicate definition of variable T.
118	ERR_GCODE_MULTIPLE_U_WORDS_ON_ONE_LINE	Duplicate definition of variable U.
119	ERR_GCODE_MULTIPLE_V_WORDS_ON_ONE_LINE	Duplicate definition of variable V.
120	ERR_GCODE_MULTIPLE_W_WORDS_ON_ONE_LINE	Duplicate definition of variable W.
121	ERR_GCODE_MULTIPLE_X_WORDS_ON_ONE_LINE	Duplicate definition of variable X.
122	ERR_GCODE_MULTIPLE_Y_WORDS_ON_ONE_LINE	Duplicate definition of variable Y.
123	ERR_GCODE_MULTIPLE_Z_WORDS_ON_ONE_LINE	Duplicate definition of variable Z.

9

9.5.3 Variable of G-code is a negative number

Number	Name	Description
201	ERR_GCODE_NEGATIVE_D_WORD	Variable D is a negative number.
202	ERR_GCODE_NEGATIVE_F_WORD	Variable F is a negative number.
203	ERR_GCODE_NEGATIVE_G_WORD	Variable G is a negative number.
204	ERR_GCODE_NEGATIVE_H_WORD	Variable H is a negative number.
205	ERR_GCODE_NEGATIVE_L_WORD	Variable L is a negative number.
206	ERR_GCODE_NEGATIVE_M_WORD	Variable M is a negative number.
207	ERR_GCODE_NEGATIVE_P_WORD	Variable P is a negative number.

9.5.4 Undefined G-code character / function

Number	Name	Description
301	ERR_GCODE_BAD_CHARACTER	Variable range is not within A to Z.
302	ERR_GCODE_UNKNOWN_CHARACTER	Unknown variable.
303	ERR_GCODE_UNKNOWN_G_CODE	Unknown G-code.
304	ERR_GCODE_UNKNOWN_WORD_STARTING_ WITH_A	Unknown function starting with A.
305	ERR_GCODE_UNKNOWN_WORD_STARTING_ WITH_C	Unknown function starting with C.
306	ERR_GCODE_UNKNOWN_WORD_STARTING_ WITH_E	Unknown function starting with E.
307	ERR_GCODE_UNKNOWN_WORD_STARTING_ WITH_F	Unknown function starting with F.
308	ERR_GCODE_UNKNOWN_WORD_STARTING_ WITH_L	Unknown function starting with L.
309	ERR_GCODE_UNKNOWN_WORD_STARTING_ WITH_P	Unknown function starting with P.
310	ERR_GCODE_UNKNOWN_WORD_STARTING_ WITH_R	Unknown function starting with R.
311	ERR_GCODE_UNKNOWN_WORD_STARTING_ WITH_S	Unknown function starting with S.
312	ERR_GCODE_UNKNOWN_WORD_STARTING_ WITH_T	Unknown function starting with T.
313	ERR_GCODE_UNKNOWN_OPERATION	The expression contains undefined operators.
314	ERR_GCODE_BUG_UNKNOWN_OPERATION	Undefined operating mode.
315	ERR_GCODE_UNKNOWN_OPERATION_NAME_ STARTING_WITH_A	Unknown operator starting with A.
316	ERR_GCODE_UNKNOWN_OPERATION_NAME_ STARTING_WITH_E	Unknown operator starting with E.
317	ERR_GCODE_UNKNOWN_OPERATION_NAME_ STARTING_WITH_G	Unknown operator starting with G.
318	ERR_GCODE_UNKNOWN_OPERATION_NAME_ STARTING_WITH_L	Unknown operator starting with L.
319	ERR_GCODE_UNKNOWN_OPERATION_NAME_ STARTING_WITH_M	Unknown operator starting with M.
320	ERR_GCODE_UNKNOWN_OPERATION_NAME_ STARTING_WITH_N	Unknown operator starting with N.
321	ERR_GCODE_UNKNOWN_OPERATION_NAME_ STARTING_WITH_O	Unknown operator starting with O.
322	ERR_GCODE_UNKNOWN_OPERATION_NAME_ STARTING_WITH_X	Unknown operator starting with X.

Number	Name	Description
330	ERR_GCODE_UNKNOWN_WORD_WHERE_UNARY_OPERATION_COULD_BE	Undefined function is used.
331	WARNING_GCODE_G10_UNKNOWN_TYPE	G10 not supported.

9.5.6 G-code setting exceeds the range

Number	Name	Description
403	ERR_GCODE_M_CODE_TOO_BIG	M-code exceeds the range; allowable range: 0 - 999.
405	ERR_GCODE_PARAMETER_NUMBER_OUT_OF_RANGE	Access variable # exceeds the range.
406	ERR_GCODE_H_WORD_EMPTY	1. Tool No. is not set when executing tool length compensation. 2. Tool No. is not set when executing tool radius compensation.
408	ERR_GCODE_GLOBAL_PARAMETER_NUMBER_OUT_OF_RANGE	Access variable @ exceeds the range.

9

9.5.7 Other errors

Number	Name	Description
501	ERR_GCODE_NEGATIVE_OR_ZERO_Q_VALUE	The set value of the feed cutting depth Q cannot be smaller than or equal 0.
502	ERR_GCODE_NEGATIVE_SPINDLE_SPEED	The set value of speed S cannot be smaller than 0.
503	ERR_GCODE_NEGATIVE_TOOL_ID	The set value of tool T cannot be smaller than 0.
504	ERR_GCODE_TWO_G_CODES_USED_FROM_ SAME_MODAL_GROUP	Repeatedly set functional group within one G-code.
511	ERR_GCODE_G51_AXES_NOT_EQUAL_TWO	Scaling value must set with two axes.
512	ERR_GCODE_G51_X_SCALE_VALUE_ZERO	Scaling point is set, but the scaling value is not specified.
513	ERR_GCODE_G51_Y_SCALE_VALUE_ZERO	
514	ERR_GCODE_G51_Z_SCALE_VALUE_ZERO	
516	ERR_GCODE_G51_1_AXES_NOT_ASSIGN	Mirror axis is not specified.
522	ERR_GCODE_G68_ROTATE_ANGLE_NOT_ ASSING	Rotation angle is not specified.
530	ERR_Cycle_Repet_Cnt_Negative	The repeating count of the drilling cycle is negative.
701	ERR_SNC_INITIAL_FAILED	SNC initialization failed.
702	ERR_CANT_SET_WHEN_PROCESSING	Unable to set parameters during processing.
703	ERR_AXIS_OUT_OF_RNG	Axis number exceeds the range; allowable range: 1 - 9.
704	ERR_AXIS_REDEFINE	Axis number is repeatedly used.
706	ERR_AXES_ZERO	Total number of axis in use is zero.
707	ERR_AXES_OUT_OF_RNG	The number of axis used by G-code is larger than the set value of SNC_Axes.
708	ERR_MACRO_MODE_OUT_OF_RNG	Incorrect setting of macro mode.
709	ERR_CALLBACK_NULL	CALLBACK function is not specified.
711	ERR_GEAR_ZERO	Gear ratio is zero.
713	ERR_G00_SPD_ZERO	G00 feed speed is zero.
715	ERR_WRONG_PLANE	Incorrect setting of working plane.
723	ERR_MACRO_OVER_RNG	Layer of subprogram exceeds four layers.
727	ERR_G02_G03_PARAM	Incorrect setting of G02 and G03 parameters.
728	ERR_G02_G03_PLANE	Incorrect working plane of G02 and G03 (G17 - G19).
729	ERR_G02_G03_CALC	Unable to calculate the coordinates of G02 and G03.
730	ERR_G02_G03_AXES_OVER	The number of axis used by G02 and G03 is over 3 axes.
731	ERR_PROCESSING_IS_RUNNING	Unable to use the processing function during processing.
732	ERR_TOOL_MAX_OVER_RNG	SNC_Tool_Max parameter setting exceeds the range; allowable range: 1 - 100.
733	ERR_CUTTER_COMPENSATION_ARC_PLANE_ NOT_SUPPORT	Tool radius compensation only supports the X-Y plane.

Number	Name	Description
734	ERR_CUTTER_COMPENSATION_CANT_USE_HELI	Tool radius compensation and helical interpolation cannot be used at the same time.
735	ERR_CUTTER_COMPENSATION_CALC	Unable to calculate the path of tool radius compensation.
737	ERR_CUTTER_FIRST_MOTION_ARC	The first motion of tool radius compensation cannot be G02 / G03.
738	ERR_CUTTER_NOT_FINISH	The macro function cannot be executed before the tool radius compensation is completed.
741	ERR_MEMORY_ALLOC_FAIL	Memory allocation is in error.
742	ERR_USER_CALLBACK_NULL	Used the user macro function, but the CALLBACK function is not specified.
761	ERR_ISO_CYCLE_MODE_OUT_OF_RNG	Incorrect setting of drilling mode.
762	ERR_CALLBACK_ISO_CYCLE_NULL	The CALLBACK function is not set for the drilling mode.
763	ERR_ISO_CYCLE_NOT_SUPPORT	Drilling cycle mode is not supported.
802	ERR_SETTING_GEAR	Incorrect setting of gear ratio.
803	ERR_SETTING_AXIS	Incorrect setting of SNC axis number.
805	ERR_TOOL_RADIUS_INCORRECT	Incorrect tool radius value (too big or too small).
806	ERR_SETTING_TOOL_MAX_ZERO	Used the SNC_Check_Tool_No parameter, but the set value of SNC_Tool_Max is zero.
807	ERR_SETTING_DIRECT	Incorrect direction setting (the setting value can only be -1 or 1).
809	ERR_SETTING_UNIT	Incorrect unit setting.
810	ERR_TOOL_PARTS_OVER_RNG	T-code group number in the same line exceeds the range (use '/' to separate).
901	ERR_DEVICE_04PI_MODE1	Device cannot be 04PI Mode1.
902	ERR_DEVICE_RM_MODULE	Device cannot be RM module.
903	ERR_DEVICE_NO_DEVICE	Cannot find the device with the station No.
904	ERR_DEVICE_UNKNOWN	Unknown device.
911	ERR_API_ERRNO	Bottom layer API returns error; read parameter SNC_API_ErrNo.
1001	ERR_GROUP_INIT_FIRST	Group number is not set.
1002	ERR_GRUOP_OVER_RANGE	Operation group exceeds the set range.
1004	ERR_GRUOP_CARD_TYPE	Wrong card type.
2000	ERR_TRIGGER_SOFT_LIMIT The V1.08 version added error codes of 2001 - 2019 indicating the axis number that triggered the software limit.	Software limit is triggered. 20X0: Axis X - W software positive limit is triggered. 20X1: Axis X - W software negative limit is triggered.
9999	ERR_SECURITY_FAILED	Security authentication failed.

9

9.5.8 SNC activation error code

SNC1	SNC2	SNC3	SNC4	Description	Troubleshooting
30001	30101	30201	30301	X axis is in motion and the SNC failed to start.	SNC cannot be executed simultaneously with the PLC motion command. Make sure that the SNC sets each servo axis to the standby mode before starting.
30002	30102	30202	30302	Y axis is in motion and the SNC failed to start.	
30003	30103	30203	30303	Z axis is in motion and the SNC failed to start.	
30004	30104	30204	30304	A axis is in motion and the SNC failed to start.	
30005	30105	30205	30305	B axis is in motion and the SNC failed to start.	
30006	30106	30206	30306	C axis is in motion and the SNC failed to start.	
30007	30107	30207	30307	U axis is in motion and the SNC failed to start.	
30008	30108	30208	30308	V axis is in motion and the SNC failed to start.	
30009	30109	30209	30309	W axis is in motion and the SNC failed to start.	
30051	30151	30251	30351	X axis is not enabled (Servo off).	You can enable the servo drive by using the SVON instruction. If you cannot enable the servo drive due to servo error, refer to the servo drive user manual for troubleshooting.
30052	30152	30252	30352	Y axis is not enabled (Servo off).	
30053	30153	30253	30353	Z axis is not enabled (Servo off).	
30054	30154	30254	30354	A axis is not enabled (Servo off).	
30055	30155	30255	30355	B axis is not enabled (Servo off).	
30056	30156	30256	30356	C axis is not enabled (Servo off).	
30057	30157	30257	30357	U axis is not enabled (Servo off).	
30058	30158	30258	30358	V axis is not enabled (Servo off).	
30059	30159	30259	30359	W axis is not enabled (Servo off).	
30061	30161	30261	30361	The R relay corresponding to M-code is not correctly executed, so the activation of the SNC is prohibited.	Check the cause for the M-code / T-code not being able to execute and correct the PLC program.
30062	30162	30262	30362	The R relay corresponding to T-code is not correctly executed, so the activation of the SNC is prohibited.	
30063	30163	30263	30363	Undefined range of M-code is used.	Modify the M-code / T-code set number.
30064	30164	30264	30364	Undefined range of T-code is used.	
30070	30170	30270	30370	Servo drive motion mode error.	-
30080	30180	30280	30380	Multi-axis synchronous is executing the homing process.	-

List of Special Register A

When editing the PLC instructions of the IMP, you can find the definitions of each register in this chapter.

A

A.1 Troubleshooting

Error occurrence	Causes	Corrective actions
After starting the IMP software, it displays "Register failed!”	IMP software is not certified.	Obtain the license authorization file and register with the RegisterAP software; refer to Section 2.3 for detailed information.
Input / output signal error.	Connection of the terminal block is loose or has poor contact.	Check if the wiring or terminal is loose.
Pop-up window displays “Init fail, please rstart software!”	The motion card and driver are not properly installed.	Check if the Delta motion card and driver are installed correctly.
Pop-up window displays “Card NO:X NO slave found”.	Communication protocol or node number setting error.	In DMCNET fieldbus, node 1 must exist. Check if the setting for the communication protocol and node number is correct. (Refer to Section 3.1 for more details.)
	Poor communication.	The maximum connection distance is 30 meters for DMCNET and please use shielded-twisted pair cables.
DOPSoft is unable to download the PLC program and a pop-up window displays “ETHERNET can’t opened”.	The HMI software of the IMP is not activated.	Start the IMP software and make sure that the HMI software has been started normally.
	Ethernet is not connected.	Check if the Ethernet is connected.
	The firewall of the IPC for installing the IMP software is not disabled.	Set the IMP software as a firewall exception or disable the firewall.
	The editing computer and IMP platform are not on the same local area network.	Change both IP addresses to the same subnetwork.
A warning message of “DB!” pops up on the HMI software.	SSCERuntime_x86-ENU is not installed.	Install SSCERuntime_x86-ENU.

A.2 List of special registers (W, R) in the IMP system

■ Operation status

Function	Property	Bus	No.	Description
Operation flag (Contact a)	R	D/E	R0	Operation flag of the IMP software.
Operation flag (Contact b)	R	D/E	R1	Operation flag of the IMP software (Contact b).
Reserved	-	D/E	R2	Unexpected conditions may occur when operating such relays.
Reserved	-	D/E	R3	Unexpected conditions may occur when operating such relays.
Initial pulse	R	D/E	R4	This bit is on during the first PLC cycle.
Reserved	-	D/E	R5	Unexpected conditions may occur when operating such relays.
HMI minimized flag	R	D/E	R6	The HMI software window is minimized.
Clock pulse 0.5 / 0.5 seconds	R	D/E	R13	When the PLC is operating, this bit is on for 0.5 seconds and off for 0.5 seconds. Clock drift may occur.
Clock pulse 1 second / 1 second	R	D/E	R14	When the PLC is operating, this bit is on for 1 seconds and off for 1 second. Clock drift may occur.

■ Perpetual calendar

Function	Property	Bus	No.	Description
Date (year)	R	D/E	W80	Read the system time when using the IMP software.
Date (month)	R	D/E	W81	
Date (day)	R	D/E	W82	
Time (hour)	R	D/E	W83	
Time (minute)	R	D/E	W84	
Time (second)	R	D/E	W85	
Time (total seconds)	R	D/E	W86	Starts counting from 00:00:00.
	R	D/E	W87	
Date (total days)	R	D/E	W88	Starts counting from 1/1/1980.
	R	D/E	W89	
Date (week)	R	D/E	W90	-

A

- Motion card information

Function	Property	Bus	Card 1	Card 2	Card 3	Description
Motion card number	R	D/E	W6000	W6500	W7000	Display the number of the motion card.
Motion card version	R	D/E	W6001	W6501	W7001	Display the firmware version of the motion card.
	R	D/E	W6002	W6502	W7002	
Number of the motion card transmission error	R	D/E	W6003	W6503	W7003	Display the accumulated number of times of the motion card communication error.
Number of the motion card receiving error	R	D/E	W6004	W6504	W7004	
	R	D/E	W6005	W6505	W7005	

- DMCNET connected device type

Name	Property	Bus	Card 1	Card 2	Card 3	Description
Device type of node 1	R	D	W6010	W6510	W7010	Read the connected device type. See below for the code and corresponding model: 0x07020192 : ASD-A3-F 0X04020192: ASD-A2-F 0X04020192: ASD-A2-R 0X06020192: ASD-M 0X04020192: ASD-A2-S 0X04020192: ASD-B2-F 0X04020192: ASD-DMC-RM32NT 0X04020192: ASD-DMC-RM64NT 0X04020192: ASD-DMC-RM32MN 0X04020192: ASD-DMC-RM64MN 0X04020192: ASD-DMC-RM32PT 0X14100191: ASD-DMC-RM04PiM2 0X04020192: ASD-DMC-RM04AD 0X04020192: ASD-DMC-RM04DA 0X04020192: ASD-DMC-GE01PH 0X04020192: HMC-RIO3232RT5
	R	D	W6011	W6511	W7011	
Device type of node 2	R	D	W6012	W6512	W7012	
	R	D	W6013	W6513	W7013	
Device type of node 3	R	D	W6014	W6514	W7014	
	R	D	W6015	W6515	W7015	
Device type of node 4	R	D	W6016	W6516	W7016	
	R	D	W6017	W6517	W7017	
Device type of node 5	R	D	W6018	W6518	W7018	
	R	D	W6019	W6519	W7019	
Device type of node 6	R	D	W6020	W6520	W7020	
	R	D	W6021	W6521	W7021	
Device type of node 7	R	D	W6022	W6522	W7022	
	R	D	W6023	W6523	W7023	
Device type of node 8	R	D	W6024	W6524	W7024	
	R	D	W6025	W6525	W7025	
Device type of node 9	R	D	W6026	W6526	W7026	
	R	D	W6027	W6527	W7027	
Device type of node 10	R	D	W6028	W6528	W7028	
	R	D	W6029	W6529	W7029	
Device type of node 11	R	D	W6030	W6530	W7030	
	R	D	W6031	W6531	W7031	
Device type of node 12	R	D	W6032	W6532	W7032	
	R	D	W6033	W6533	W7033	

A

■ EtherCAT connected device type

Name	Property	Bus	Card 1	Card 2	Card 3	Description
Device type of node 0	R	E	W6010	W6510	W7010	Read the connected device type. See below for the code and corresponding model: 0x10305070: ASD-A2-E 0x5500: R1-EC5500 0x5614: R1-EC5614 0x5621: R1-EC5621 0x6002: R1-EC6002 0x6022: R1-EC6022 0x7062: R1-EC7062 0x8124: R1-EC8124 0x9144: R1-EC9144 0x9621: R1-EC9621
			W6011	W6511	W7011	
≀			≀	≀	≀	
Device type of node 99			W6208	W6708	W7208	
			W6209	W6709	W7209	

■ DMCNET connected device version

Name	Property	Bus	Card 1	Card 2	Card 3	Description
Firmware version of node 1	R	D	W6082	W6582	W7082	Display the firmware version of the connected device. Refer to the user manual of each device for further information.
	R	D	W6083	W6583	W7083	
Firmware version of node 2	R	D	W6084	W6584	W7084	
	R	D	W6085	W6585	W7085	
Firmware version of node 3	R	D	W6086	W6586	W7086	
	R	D	W6087	W6587	W7087	
Firmware version of node 4	R	D	W6088	W6588	W7088	
	R	D	W6089	W6589	W7089	
Firmware version of node 5	R	D	W6090	W6590	W7090	
	R	D	W6091	W6591	W7091	
Firmware version of node 6	R	D	W6092	W6592	W7092	
	R	D	W6093	W6593	W7093	
Firmware version of node 7	R	D	W6094	W6594	W7094	
	R	D	W6095	W6595	W7095	
Firmware version of node 8	R	D	W6096	W6596	W7096	
	R	D	W6097	W6597	W7097	
Firmware version of node 9	R	D	W6098	W6598	W7098	
	R	D	W6099	W6599	W7099	
Firmware version of node 10	R	D	W6100	W6600	W7100	
	R	D	W6101	W6601	W7101	
Firmware version of node 11	R	D	W6102	W6602	W7102	
	R	D	W6103	W6603	W7103	
Firmware version of node 12	R	D	W6104	W6604	W7104	
	R	D	W6105	W6605	W7105	

A

- DMCNET communication error

Name	Property	Bus	Card 1	Card 2	Card 3	Description
Communication error times of node 1	R	D	W6154	W6654	W7154	Display the accumulated number of communication errors between the master station and slave station.
Communication error times of node 2	R	D	W6155	W6655	W7155	
Communication error times of node 3	R	D	W6156	W6656	W7156	
Communication error times of node 4	R	D	W6157	W6657	W7157	
Communication error times of node 5	R	D	W6158	W6658	W7158	
Communication error times of node 6	R	D	W6159	W6659	W7159	
Communication error times of node 7	R	D	W6160	W6660	W7160	
Communication error times of node 8	R	D	W6161	W6661	W7161	
Communication error times of node 9	R	D	W6162	W6662	W7162	
Communication error times of node 10	R	D	W6163	W6663	W7163	
Communication error times of node 11	R	D	W6164	W6664	W7164	
Communication error times of node 12	R	D	W6165	W6665	W7165	

A

■ Read the status of the motion card input contact

Name	Property	Bus	Card 1	Card 2	Card 3	Description
Motion card input 0	R	D	R6200	R6700	R7200	Read the DMCNET input contacts. Supported motion cards: ASD-DMC-A02 ASD-DMC-F02 Supported PAC: MP1-A12D-15
Motion card input 1	R	D	R6201	R6701	R7201	
Motion card input 2	R	D	R6202	R6702	R7202	
Motion card input 3	R	D	R6203	R6703	R7203	
Motion card input 4	R	D	R6204	R6704	R7204	
Motion card input 5	R	D	R6205	R6705	R7205	
Motion card input 6	R	D	R6206	R6706	R7206	
Motion card input 7	R	D	R6207	R6707	R7207	
Motion card input 8	R	D	R6208	R6708	R7208	
Motion card input 9	R	D	R6209	R6709	R7209	
Motion card input 10	R	D	R6210	R6710	R7210	
Motion card input 11	R	D	R6211	R6711	R7211	
Motion card input 12	R	D	R6212	R6712	R7212	
Motion card input 13	R	D	R6213	R6713	R7213	
Motion card input 14	R	D	R6214	R6714	R7214	
Motion card input 15	R	D	R6215	R6715	R7215	
Motion card input 16	R	D	R6216	R6716	R7216	
Motion card input 17	R	D	R6217	R6717	R7217	
Motion card input 18	R	D	R6218	R6718	R7218	
Motion card input 19	R	D	R6219	R6719	R7219	
Motion card input 20	R	D	R6220	R6720	R7220	
Motion card input 21	R	D	R6221	R6721	R7221	
Motion card input 22	R	D	R6222	R6722	R7222	
Motion card input 23	R	D	R6223	R6723	R7223	
Motion card input 24	R	D	R6224	R6724	R7224	
Motion card input 25	R	D	R6225	R6725	R7225	
Motion card input 26	R	D	R6226	R6726	R7226	
Motion card input 27	R	D	R6227	R6727	R7227	
Motion card input 28	R	D	R6228	R6728	R7228	
Motion card input 29	R	D	R6229	R6729	R7229	
Motion card input 30	R	D	R6230	R6730	R7230	
Motion card input 31	R	D	R6231	R6731	R7231	

A

■ Motion card output contact control

Name	Property	Bus	Card 1	Card 2	Card 3	Description
Motion card output 0	R/W	D	R6300	R6800	R7300	Set the DMCNET output contacts. Supported motion cards: ASD-DMC-A02 ASD-DMC-F02 Supported PAC: MP1-A12D-15
Motion card output 1	R/W	D	R6301	R6801	R7301	
Motion card output 2	R/W	D	R6302	R6802	R7302	
Motion card output 3	R/W	D	R6303	R6803	R7303	
Motion card output 4	R/W	D	R6304	R6804	R7304	
Motion card output 5	R/W	D	R6305	R6805	R7305	
Motion card output 6	R/W	D	R6306	R6806	R7306	
Motion card output 7	R/W	D	R6307	R6807	R7307	
Motion card output 8	R/W	D	R6308	R6808	R7308	
Motion card output 9	R/W	D	R6309	R6809	R7309	
Motion card output 10	R/W	D	R6310	R6810	R7310	
Motion card output 11	R/W	D	R6311	R6811	R7311	
Motion card output 12	R/W	D	R6312	R6812	R7312	
Motion card output 13	R/W	D	R6313	R6813	R7313	
Motion card output 14	R/W	D	R6314	R6814	R7314	
Motion card output 15	R/W	D	R6315	R6815	R7315	
Motion card output 16	R/W	D	R6316	R6816	R7316	
Motion card output 17	R/W	D	R6317	R6817	R7317	
Motion card output 18	R/W	D	R6318	R6818	R7318	
Motion card output 19	R/W	D	R6319	R6819	R7319	
Motion card output 20	R/W	D	R6320	R6820	R7320	
Motion card output 21	R/W	D	R6321	R6821	R7321	
Motion card output 22	R/W	D	R6322	R6822	R7322	
Motion card output 23	R/W	D	R6323	R6823	R7323	
Motion card output 24	R/W	D	R6324	R6824	R7324	
Motion card output 25	R/W	D	R6325	R6825	R7325	
Motion card output 26	R/W	D	R6326	R6826	R7326	
Motion card output 27	R/W	D	R6327	R6827	R7327	
Motion card output 28	R/W	D	R6328	R6828	R7328	
Motion card output 29	R/W	D	R6329	R6829	R7329	
Motion card output 30	R/W	D	R6330	R6830	R7330	
Motion card output 31	R/W	D	R6331	R6831	R7331	

■ Module number

Function	Property	Bus	No.	Description
Total card number	R	D/E	W8000	Based on the number of devices that the IMP is connected to, PLC can check if all slave devices are well connected.
Total number of servo axis	R	D/E	W8001	
Reserved	R	D/E	W8002	
Node number of digital input module	R	D/E	W8003	
Node number of digital output module	R	D/E	W8004	
Channel number of analog input	R	D/E	W8005	
Channel number of analog output	R	D/E	W8006	

■ System identification code

Function	Property	Bus	No.	Description
Device identification code	R	D/E	W8010	This identification code can be used to protect the PLC program. Verifying the identification code in the PLC program can prevent the PAC program from being copied. This unique identification code is generated by the hardware, and if you change the network card, motherboard, CPU, hard disk, or motion card, the identification code may also change.
	R	D/E	W8011	
	R	D/E	W8012	
	R	D/E	W8013	

■ User interface

Function	Property	Bus	No.	Description
User interface activation code (User_interface)	R/W	D/E	W9000	Before calling the user interface, you should call the following settings or editing software with this parameter setting: (Refer to W9020 for the user interface call path.) 1010: MPM editor. 3000: SNC parameter setting. 3010: SNC backlash compensation setting. 3020: coordinates setting for G52 - G59. 3030: tool length and tool radius setting. 3040: tool offset setting (spindle and line boring machine). 3050: setting of tool setter.
Execution path of user interface (50 registers are applied)	R	D/E	W9020	By using the DOPSoft action element and setting its execution path, you can activate the user interface. (Work with W9000.)

■ System control

Function	Property	Bus	No.	Description
System operation control area	R/W	D/E	W9200	After this register is set, you need to trigger R9200 for it to take effect. 0: none. 1: disable the IMP software and operation system. 2: stop the PLC operation. 3: restart IMP. 4: turn off IMP.
Activate system operation	R/W	D/E	R9200	Activate the system operation functions.

■ Analog input

<table>
<tr><th>Function</th><th>Property</th><th>Bus</th><th>No.</th><th>Description</th></tr>
<tr><td>Analog input value (CH1)</td><td>R</td><td>D/E</td><td>W9800</td><td rowspan="32">Set the input range of the AD module according to the Quick Start interface. See below for the corresponding resolution:
Voltage mode (0 - 5V): 0 - 5000 (unit: mV)
Voltage mode (0 - 10V): 0 - 10000 (unit: mV)
Voltage mode (-5 to 5V): -5000 to 5000 (unit: mV)
Voltage mode (-10 to 10V): -10000 to 10000 (unit: mV)
Current mode (0 - 20 mA): 0 - 20000 (unit: 0.001 mA)</td></tr>
<tr><td>Analog input value (CH2)</td><td>R</td><td>D/E</td><td>W9801</td></tr>
<tr><td>Analog input value (CH3)</td><td>R</td><td>D/E</td><td>W9802</td></tr>
<tr><td>Analog input value (CH4)</td><td>R</td><td>D/E</td><td>W9803</td></tr>
<tr><td>Analog input value (CH5)</td><td>R</td><td>D/E</td><td>W9804</td></tr>
<tr><td>Analog input value (CH6)</td><td>R</td><td>D/E</td><td>W9805</td></tr>
<tr><td>Analog input value (CH7)</td><td>R</td><td>D/E</td><td>W9806</td></tr>
<tr><td>Analog input value (CH8)</td><td>R</td><td>D/E</td><td>W9807</td></tr>
<tr><td>Analog input value (CH9)</td><td>R</td><td>D/E</td><td>W9808</td></tr>
<tr><td>Analog input value (CH10)</td><td>R</td><td>D/E</td><td>W9809</td></tr>
<tr><td>Analog input value (CH11)</td><td>R</td><td>D/E</td><td>W9810</td></tr>
<tr><td>Analog input value (CH12)</td><td>R</td><td>D/E</td><td>W9811</td></tr>
<tr><td>Analog input value (CH13)</td><td>R</td><td>D/E</td><td>W9812</td></tr>
<tr><td>Analog input value (CH14)</td><td>R</td><td>D/E</td><td>W9813</td></tr>
<tr><td>Analog input value (CH15)</td><td>R</td><td>D/E</td><td>W9814</td></tr>
<tr><td>Analog input value (CH16)</td><td>R</td><td>D/E</td><td>W9815</td></tr>
<tr><td>Analog input value (CH17)</td><td>R</td><td>D/E</td><td>W9816</td></tr>
<tr><td>Analog input value (CH18)</td><td>R</td><td>D/E</td><td>W9817</td></tr>
<tr><td>Analog input value (CH19)</td><td>R</td><td>D/E</td><td>W9818</td></tr>
<tr><td>Analog input value (CH20)</td><td>R</td><td>D/E</td><td>W9819</td></tr>
<tr><td>Analog input value (CH21)</td><td>R</td><td>D/E</td><td>W9820</td></tr>
<tr><td>Analog input value (CH22)</td><td>R</td><td>D/E</td><td>W9821</td></tr>
<tr><td>Analog input value (CH23)</td><td>R</td><td>D/E</td><td>W9822</td></tr>
<tr><td>Analog input value (CH24)</td><td>R</td><td>D/E</td><td>W9823</td></tr>
<tr><td>Analog input value (CH25)</td><td>R</td><td>D/E</td><td>W9824</td></tr>
<tr><td>Analog input value (CH26)</td><td>R</td><td>D/E</td><td>W9825</td></tr>
<tr><td>Analog input value (CH27)</td><td>R</td><td>D/E</td><td>W9826</td></tr>
<tr><td>Analog input value (CH28)</td><td>R</td><td>D/E</td><td>W9827</td></tr>
<tr><td>Analog input value (CH29)</td><td>R</td><td>D/E</td><td>W9828</td></tr>
<tr><td>Analog input value (CH30)</td><td>R</td><td>D/E</td><td>W9829</td></tr>
<tr><td>Analog input value (CH31)</td><td>R</td><td>D/E</td><td>W9830</td></tr>
<tr><td>Analog input value (CH32)</td><td>R</td><td>D/E</td><td>W9831</td></tr>
</table>

■ Analog output

Function	Property	Bus	No.	Description
Analog output value (CH1)	R/W	D/E	W9900	Set the output range of the DA module according to the Quick Start interface. See below for the corresponding resolution: Voltage mode (0 - 5V): 0 - 5000 (unit: mV) Voltage mode (0 - 10V): 0 - 10000 (unit: mV) Voltage mode (-5 to 5V): -5000 to 5000 (unit: mV) Voltage mode (-10 to 10V): -10000 to 10000 (unit: mV) Current mode (4 - 20 mA): 4000 - 20000 (unit: 0.001 mA) Current mode (0 - 20 mA): 0 - 20000 (unit: 0.001 mA) Current mode (0 - 24 mA): 0 - 24000 (unit: 0.001 mA)
Analog output value (CH2)	R/W	D/E	W9901	
Analog output value (CH3)	R/W	D/E	W9902	
Analog output value (CH4)	R/W	D/E	W9903	
Analog output value (CH5)	R/W	D/E	W9904	
Analog output value (CH6)	R/W	D/E	W9905	
Analog output value (CH7)	R/W	D/E	W9906	
Analog output value (CH8)	R/W	D/E	W9907	
Analog output value (CH9)	R/W	D/E	W9908	
Analog output value (CH10)	R/W	D/E	W9909	
Analog output value (CH11)	R/W	D/E	W9910	
Analog output value (CH12)	R/W	D/E	W9911	
Analog output value (CH13)	R/W	D/E	W9912	
Analog output value (CH14)	R/W	D/E	W9913	
Analog output value (CH15)	R/W	D/E	W9914	
Analog output value (CH16)	R/W	D/E	W9915	
Analog output value (CH17)	R/W	D/E	W9916	
Analog output value (CH18)	R/W	D/E	W9917	
Analog output value (CH19)	R/W	D/E	W9918	
Analog output value (CH20)	R/W	D/E	W9919	
Analog output value (CH21)	R/W	D/E	W9920	
Analog output value (CH22)	R/W	D/E	W9921	
Analog output value (CH23)	R/W	D/E	W9922	
Analog output value (CH24)	R/W	D/E	W9923	
Analog output value (CH25)	R/W	D/E	W9924	
Analog output value (CH26)	R/W	D/E	W9925	
Analog output value (CH27)	R/W	D/E	W9926	
Analog output value (CH28)	R/W	D/E	W9927	
Analog output value (CH29)	R/W	D/E	W9928	
Analog output value (CH30)	R/W	D/E	W9929	
Analog output value (CH31)	R/W	D/E	W9930	
Analog output value (CH32)	R/W	D/E	W9931	

A.3 List of special registers (W, R) for single-axis motion

■ Special registers (W) for single-axis motion status

Function	Property	Bus	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	~	Axis 36
Motor feedback position	R	D/E	W10102	W10202	W10302	W10402	W10502	~	W13602
	R	D/E	W10103	W10203	W10303	W10403	W10503	~	W13603
Motion card command position	R	D/E	W10104	W10204	W10304	W10404	W10504	~	W13604
	R	D/E	W10105	W10205	W10305	W10405	W10505	~	W13605
Target position	R	D/E	W10106	W10206	W10306	W10406	W10506	~	W13606
	R	D/E	W10107	W10207	W10307	W10407	W10507	~	W13607
Servo drive DI status	R	D/E	W10108	W10208	W10308	W10408	W10508	~	W13608
Servo drive DO status	R	D/E	W10109	W10209	W10309	W10409	W10509	~	W13609
Current motion speed of each axis	R	D/E	W10110	W10210	W10310	W10410	W10510	~	W13610
	R	D/E	W10111	W10211	W10311	W10411	W10511	~	W13611
Current output torque of the motor	R	D/E	W10113	W10213	W10313	W10413	W10513	~	W13613
Command status	R	D/E	W10114	W10214	W10314	W10414	W10514	~	W13614
Servo error code	R	D/E	W10115	W10215	W10315	W10415	W10515	~	W13615
Read servo return value	R	D/E	W10116	W10216	W10316	W10416	W10516	~	W10616
	R	D/E	W10117	W10217	W10317	W10417	W10517	~	W10167
Current motor speed (rpm)	R	D/E	W10119	W10219	W10319	W10419	W10519	~	W13619
	R	D/E	W10120	W10220	W10320	W10420	W10520	~	W13620
Error code of single-axis operation	R	D/E	W10150	W10250	W10350	W10450	W10550	~	W13650

■ Special register (W) for single-axis motion control

Function	Property	Bus	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	~	Axis 36
Single-axis motion control code	R/W	D/E	W10151	W10251	W10351	W10451	W10551	~	W13651
Acceleration time	R/W	D/E	W10152	W10252	W10352	W10452	W10552	~	W13652
	R/W	D/E	W10153	W10253	W10353	W10453	W10553	~	W13653
Deceleration time	R/W	D/E	W10154	W10254	W10354	W10454	W10554	~	W13654
	R/W	D/E	W10155	W10255	W10355	W10455	W10555	~	W13655
Target speed of motion command	R/W	D/E	W10156	W10256	W10356	W10456	W10556	~	W13656
	R/W	D/E	W10157	W10257	W10357	W10457	W10557	~	W13657
Target coordinates of motion command	R/W	D/E	W10158	W10258	W10358	W10458	W10558	~	W13658
	R/W	D/E	W10159	W10259	W10359	W10459	W10559	~	W13659
Homing mode setting	R/W	D/E	W10160	W10260	W10360	W10460	W10560	~	W13660
Setting of motion speed unit	R/W	D/E	W10161	W10261	W10361	W10461	W10561	~	W13661
First speed in homing mode	R/W	D/E	W10162	W10262	W10362	W10462	W10562	~	W13662
	R/W	D/E	W10163	W10263	W10363	W10463	W10563	~	W13663
Second speed in homing mode	R/W	D/E	W10164	W10264	W10364	W10464	W10564	~	W13664
	R/W	D/E	W10165	W10265	W10365	W10465	W10565	~	W13665
Offset in homing mode	R/W	D/E	W10166	W10266	W10366	W10466	W10566	~	W13666
	R/W	D/E	W10167	W10267	W10367	W10467	W10567	~	W13667

Function	Property	Bus	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	~	Axis 36
Target speed in speed mode	R/W	D/E	W10170	W10270	W10370	W10470	W10570	~	W13670
	R/W	D/E	W10171	W10271	W10371	W10471	W10571	~	W13671
Target torque in torque mode	R/W	D/E	W10172	W10272	W10372	W10472	W10572	~	W13672
	R/W	D/E	W10173	W10273	W10373	W10473	W10573	~	W13673
Torque limit in speed mode	R/W	D/E	W10174	W10274	W10374	W10474	W10574	~	W13674
	R/W	D/E	W10175	W10275	W10375	W10475	W10575	~	W13675
Speed limit in torque mode	R/W	D/E	W10176	W10276	W10376	W10476	W10576	~	W13676
	R/W	D/E	W10177	W10277	W10377	W10477	W10577	~	W13677
Maximum torque limit	R/W	D/E	W10178	W10278	W10378	W10478	W10578	~	W13678
	R/W	D/E	W10179	W10279	W10379	W10479	W10579	~	W13679

■ Single-axis special relay

Function	Property	Bus	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	~	Axis 36
Servo operation mode	R	D/E	R10100	R10200	R10300	R10400	R10500	~	R13600
	R	D/E	R10101	R10201	R10301	R10401	R10501	~	R13601
	R	D/E	R10102	R10202	R10302	R10402	R10502	~	R13602
	R	D/E	R10103	R10203	R10303	R10403	R10503	~	R13603
Servo DI3 status	R	D/E	R10104	R10204	R10304	R10404	R10504	~	R13604
Servo alarm flag	R	D/E	R10105	R10205	R10305	R10405	R10505	~	R13605
SVON monitoring flag	R	D/E	R10108	R10208	R10308	R10408	R10508	~	R13608
Servo error flag	R	D/E	R10109	R10209	R10309	R10409	R10509	~	R13609
Positioning complete flag	R	D/E	R10110	R10210	R10310	R10410	R10510	~	R13610
Servo operation mode (mode specific)	R	D/E	R10112	R10212	R10312	R10412	R10512	~	R13612
	R	D/E	R10113	R10213	R10313	R10413	R10513	~	R13613
Ready to Switch On	R	E	R10120	R10220	R10320	R10420	R10520	~	R13620
Operation Enabled	R	E	R10121	R10220	R10321	R10421	R10521	~	R13621
Voltage Disabled	R	E	R10122	R10222	R10322	R10422	R10522	~	R13622
Quick Stop	R	E	R10123	R10223	R10323	R10423	R10523	~	R13623
Switch On Disable	R	E	R10124	R10214	R10324	R10424	R10524	~	R13624
Homing completed	R	D/E	R10130	R10230	R10330	R13430	R13530	~	R13630
SVON control	R/W	D/E	R10151	R10251	R10351	R10451	R10551	~	R13651
Software limit enabling bit	R/W	D/E	R10152	R10252	R10352	R10452	R10552	~	R13652
Motion curve setting	R/W	D/E	R10161	R10261	R10361	R10461	R10561	~	R13661
JOG direction control	R/W	D/E	R10162	R10262	R10362	R10462	R10562	~	R13662
Torque limit enabling bit in speed mode	R/W	D/E	R10163	R10263	R10363	R10463	R10563	~	R13663
Speed limit enabling bit in torque mode	R/W	D/E	R10164	R10264	R10364	R10464	R10564	~	R13664

A

■ Read / write servo parameters of single-axis

Name	Property	Bus	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	~	Axis 36
Read servo parameter return value	R	D/E	W10116	W10216	W10316	W10416	W10516	~	W13616
	R	D/E	W10117	W10217	W10317	W10417	W10517	~	W13617
Servo parameter reading / writing error	R	D/E	W10121	W10221	W10321	W10421	W10521	~	W13621
Servo user monitor the return value	R	D/E	W10123	W10223	W10323	W10423	W10523	~	W13623
	R	D/E	W10124	W10224	W10324	W10424	W10524	!	W13624
Written parameter of the servo	R/W	D/E	W10180	W10280	W10380	W10480	W10580	~	W13680
	R/W	D/E	W10181	W10281	W10381	W10481	W10581	~	W13681
Servo parameter group and index value	R/W	D/E	W10186	W10286	W10386	W10486	W10586	~	W13686
Control code for reading / writing servo parameters	R/W	D/E	W10187	W10287	W10387	W10487	W10587	~	W13687
Set servo monitoring parameters	R/W	D/E	W10188	W10288	W10388	W10488	W10588	~	W13688

■ Software limit of single-axis

Name	Property	Bus	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	~	Axis 36
Forward software limit	R/W	D/E	W10182	W10282	W10382	W10482	W10582	~	W13682
	R/W	D/E	W10183	W10283	W10383	W10483	W10583	~	W13683
Reverse software limit	R/W	D/E	W10184	W10284	W10384	W10484	W10584	~	W13684
	R/W	D/E	W10185	W10285	W10385	W10485	W10585	~	W13685
Flag for triggering servo limit (positive)	R	D/E	R10114	R10214	R10314	R10414	R10514	~	R13614
Flag for triggering servo limit (negative)	R	D/E	R10115	R10215	R10315	R10415	R10515	~	R13615
Flag for triggering software limit (positive)	R/W	D/E	R10116	R10216	R10316	R10416	R10516	~	R13616
Flag for triggering software limit (negative)	R/W	D/E	R10117	R10217	R10317	R10417	R10517	~	R13617
Software positive limit invalid	R/W	D/E	R10118	R10218	R20318	R10418	R10518	~	R13618
Software negative limit invalid	R/W	D/E	R10119	R10219	R20319	R10419	R10519	~	R13619

A.4 List of special registers (W, R) for servo group

Function	Property	Bus	Group 1	Group 2	Group 3	Group 4	Group 5	~	Group 40
Group motion control code	R/W	D/E	W20000	W20100	W20200	W20300	W20400	~	W23900
Speed unit of group motion	R/W	D/E	W20001	W20101	W20201	W20301	W20401	~	W23901
Card number used by group motion	R/W	D/E	W20002	W20102	W20202	W20302	W20402	~	W23902
Selected axis in the group (bit)	R/W	D/E	W20003	W20103	W20203	W20303	W20403	~	W23903
	R/W	E	W20004	W20104	W20204	W20304	W20404	~	W23904
Change the speed during operation	R/W	D/E	W20012	W20112	W20212	W20312	W20412	~	W23912
	R/W	D/E	W20013	W20113	W20213	W20313	W20413	~	W23913
Maximum speed of group motion	R/W	D/E	W20014	W20114	W20214	W20314	W20414	~	W23914
	R/W	D/E	W20015	W20115	W20215	W20315	W20415	~	W23915
Acceleration time of group motion	R/W	D/E	W20016	W20116	W20216	W20316	W20416	~	W23916
	R/W	D/E	W20017	W20117	W20217	W20317	W20417	~	W23917
Deceleration time of group motion	R/W	D/E	W20018	W20118	W20218	W20318	W20418	~	W23918
	R/W	D/E	W20019	W20119	W20219	W20319	W20419	~	W23919
Arc angle of group motion	R/W	D/E	W20020	W20120	W20220	W20320	W20420	~	W23920
Direction of group motion	R/W	D/E	W20021	W20121	W20221	W20321	W20421	~	W23921
X-coordinate of circle center	R/W	D/E	W20022	W20122	W20222	W20322	W20422	~	W23922
	R/W	D/E	W20023	W20123	W20223	W20323	W20423	~	W23923
Y-coordinate of circle center	R/W	D/E	W20024	W20124	W20224	W20324	W20424	~	W23924
	R/W	D/E	W20025	W20125	W20225	W20325	W20425	~	W23925
X-coordinate of arc end point	R/W	D/E	W20026	W20126	W20226	W20326	W20426	~	W23926
	R/W	D/E	W20027	W20127	W20227	W20327	W20427	~	W23927
Y-coordinate of arc end point	R/W	D/E	W20028	W20128	W20228	W20328	W20428	~	W23928
	R/W	D/E	W20029	W20129	W20229	W20329	W20429	~	W23929
Helix depth of the three axes	R/W	D/E	W20030	W20130	W20230	W20330	W20430	~	W23930
	R/W	D/E	W20031	W20131	W20231	W20331	W20431	~	W23931
Helix pitch of the three axes	R/W	D/E	W20032	W20132	W20232	W20332	W20432	~	W23932
	R/W	D/E	W20033	W20133	W20233	W20333	W20433	~	W23933
Spindle tapping speed	R/W	D/E	W20051	W20151	W20251	W20351	W20451	~	W23951
Spindle retrieving speed	R/W	D/E	W20052	W20152	W20252	W20352	W20452	~	W23952
Tapping pitch distance	R/W	D/E	W20053	W20153	W20253	W20353	W20453	~	W23953
Delay time after tapping is completed	R/W	D/E	W20054	W20154	W20254	W20354	W20454	~	W23954
Tapping depth	R/W	D/E	W20056	W20156	W20256	W20356	W20456	~	W23956
	R/W	D/E	W20057	W20157	W20257	W20357	W20457	~	W23957
Target value of the 1st axis	R/W	D/E	W20070	W20170	W20270	W20370	W20470	~	W23970
	R/W	D/E	W20071	W20171	W20271	W20371	W20471	~	W23971
Target value of the 2nd axis	R/W	D/E	W20072	W20172	W20272	W20372	W20472	~	W23972
	R/W	D/E	W20073	W20173	W20273	W20373	W20473	~	W23973
Target value of the 3rd axis	R/W	D/E	W20074	W20174	W20274	W20374	W20474	~	W23974
	R/W	D/E	W20075	W20175	W20275	W20375	W20475	~	W23975

A

Function	Property	Bus	Group 1	Group 2	Group 3	Group 4	Group 5	~	Group 40
Target value of the 4th axis	R/W	D/E	W20076	W20176	W20276	W20376	W20476	~	W23976
	R/W	D/E	W20077	W20177	W20277	W20377	W20477	~	W23977
Target value of the 5th axis	R/W	D/E	W20078	W20178	W20278	W20378	W20478	~	W23978
	R/W	D/E	W20079	W20179	W20279	W20379	W20479	~	W23979
Target value of the 6th axis	R/W	D/E	W20080	W20180	W20280	W20380	W20480	~	W23980
	R/W	D/E	W20081	W20181	W20281	W20381	W20481	~	W23981
Target value of the 7th axis	R/W	D/E	W20082	W20182	W20282	W20382	W20482	~	W23982
	R/W	D/E	W20083	W20183	W20283	W20383	W20483	~	W23983
Target value of the 8th axis	R/W	D/E	W20084	W20184	W20284	W20384	W20484	~	W23984
	R/W	D/E	W20085	W20185	W20285	W20385	W20485	~	W23985
Target value of the 9th axis	R/W	D/E	W20086	W20186	W20286	W20386	W20486	~	W23986
	R/W	D/E	W20087	W20187	W20287	W20387	W20487	~	W23987
Target value of the 10th axis	R/W	D/E	W20088	W20188	W20288	W20388	W20488	~	W23988
	R/W	D/E	W20089	W20189	W20289	W20389	W20489	~	W23989
Target value of the 11th axis	R/W	D/E	W20090	W20190	W20290	W20390	W20490	~	W23990
	R/W	D/E	W20091	W20191	W20291	W20391	W20491	~	W23991
Target value of the 12th axis	R/W	D/E	W20092	W20192	W20292	W20392	W20492	~	W23992
	R/W	D/E	W20093	W20193	W20293	W20393	W20493	~	W23993
Interpolation error code	R	D/E	W20095	W20195	W20295	W20395	W20495	~	W23995
Group motion in process	R	D/E	R20000	R20100	R20200	R20300	R20400	~	R23900
Acceleration curve of group motion	R/W	D/E	R20010	R20110	R20210	R20310	R20410	~	R23910

A.5 List of special registers (W, R) for Motion table

Function	Property	Bus	Table1-1	Table1-2	Table1-1	Table1-2	Table1-1	Table1-2
Motion card number	R/W	D/E	W29000		W29200		W29400	
Line number in execution	R	D/E	W29010	W29110	W29210	W29310	W29410	W29510
Current motion linear speed	R	D/E	W29012	W29112	W29212	W29312	W29412	W29512
	R	D/E	W29013	W29113	W29213	W29313	W29413	W29513
Status	R	D/E	W29015	W29115	W29215	W29315	W29415	W29515
Status error code	R	D/E	W29016	W29116	W29216	W29316	W29416	W29516
I/O output node number	R/W	D/E	W29018	W29118	W29218	W29318	W29418	W29518
/O output port	R/W	D/E	W29020	W29120	W29220	W29320	W29420	W29520
I/O output start bit	R/W	D/E	W29021	W29121	W29221	W29321	W29421	W29521
Total number of points	R/W	D/E	W29022	W29122	W29222	W29322	W29422	W29522
Number of starting register D	R/W	D/E	W29024	W29124	W29224	W29324	W29424	W29524
Register point offset	R/W	D/E	W29026	W29126	W29226	W29326	W29426	W29526
Speed operation mode	R/W	D/E	W29028	W29128	W29228	W29328	W29428	W29528
Number of axis in use	R/W	D/E	W29029	W29129	W29229	W29329	W29429	W29529
Node number of the 1st axis	R/W	D/E	W29030	W29130	W29230	W29330	W29430	W29530
Node number of the 2nd axis	R/W	D/E	W29031	W29131	W29231	W29331	W29431	W29531
Node number of the 3rd axis	R/W	D/E	W29032	W29132	W29232	W29332	W29432	W29532
Node number of the 4th axis	R/W	D/E	W29033	W29133	W29233	W29333	W29433	W29533
Node number of the 5th axis	R/W	D/E	W29034	W29134	W29234	W29334	W29434	W29534
Node number of the 6th axis	R/W	D/E	W29035	W29135	W29235	W29335	W29435	W29535
Control code	R/W	D/E	W29050	W29150	W29250	W29350	W29450	W29550
Operation speed	R/W	D/E	W29052	W29152	W29252	W29352	W29452	W29552
	R/W	D/E	W29053	W29153	W29253	W29353	W29453	W29553
Acceleration time	R/W	D/E	W29054	W29154	W29254	W29354	W29454	W29554
	R/W	D/E	W29055	W29155	W29255	W29355	W29455	W29555
Speed change percentage	R/W	D/E	W29062	W29162	W29262	W29362	W29462	W29562
Accumulated length	R/W	D/E	W29070	W29170	W29270	W29370	W29470	W29570
	R/W	D/E	W29071	W29171	W29271	W29371	W29471	W29571
Corner reference speed	R/W	D/E	W29072	W29172	W29272	W29372	W29472	W29572
	R/W	D/E	W29073	W29173	W29273	W29373	W29473	W29573
Reference length	R/W	D/E	W29074	W29174	W29274	W29374	W29474	W29574
	R/W	D/E	W29075	W29175	W29275	W29375	W29475	W29575
Reference angle	R/W	D/E	W29076	W29176	W29276	W29376	W29476	W29576
	R/W	D/E	W29077	W29177	W29277	W29377	W29477	W29577
Reference speed	R/W	D/E	W29078	W29178	W29278	W29378	W29478	W29578
	R/W	D/E	W29079	W29179	W29279	W29379	W29479	W29579
Reference radius	R/W	D/E	W29080	W29180	W29280	W29380	W29480	W29580
	R/W	D/E	W29081	W29181	W29281	W29381	W29481	W29581
I/O control switch	R/W	D/E	R29018	R29118	R29218	R29318	R29418	R29518
Single step mode	R/W	D/E	R29050	R29150	R29250	R29350	R29450	R29550

Function	Property	Bus	Table1-1	Table1-2	Table1-1	Table1-2	Table1-1	Table1-2
Single step triggering	R/W	D/E	R29051	R29151	R29251	R29351	R29451	R29551
Speed change control switch	R/W	D/E	R29062	R29162	R29262	R29362	R29462	R29562

A

A.6 List of filtering special registers (W, R) for Motion table

Function	Property	Bus	Table1	Table2	Table3
Corner speed control	R/W	D/E	R29070	R29270	R29470
AMF filtering times	R/W	D	W29056	W29256	W29456
Filtering times	R/W	D	W29082	W29282	W29482
Node 1_filterTime	R/W	D	W29083	W29283	W29483
Node 2_filterTime	R/W	D	W29084	W29284	W29484
Node 3_filterTime	R/W	D	W29085	W29285	W29485
Node 4_filterTime	R/W	D	W29086	W29286	W29486
Node 5_filterTime	R/W	D	W29087	W29287	W29487
Node 6_filterTime	R/W	D	W29088	W29288	W29488
Node 7_filterTime	R/W	D	W29089	W29289	W29489
Node 8_filterTime	R/W	D	W29090	W29290	W29490
Node 9_filterTime	R/W	D	W29091	W29291	W29491
Node 10_filterTime	R/W	D	W29092	W29292	W29492
Node 11_filterTime	R/W	D	W29093	W29293	W29493
Node 12_filterTime	R/W	D	W29094	W29294	W29494

A.7 List of special registers (W, R) for Motion Program Macro (MPM)

Function	Property	Bus	MPM 1	MPM 2	MPM 3	MPM 4	MPM 5	~	MPM 100
Command control code	R/W	D/E	W30000	W30010	W30020	W30030	W30040	~	W30990
Total line number	R/W	D/E	W30001	W30011	W30021	W30031	W30041	~	W30991
Line number in execution	R/W	D/E	W30002	W30012	W30022	W30032	W30042	~	W30992
Feed rate percentage	R/W	D/E	W30003	W30013	W30023	W30033	W30043	~	W30993
Error position	R	D/E	W30007	W30017	W30027	W30037	W30047	~	W30997
Syntax error code	R	D/E	W30008	W30018	W30028	W30038	W30048	~	W30998
Execute error code	R	D/E	W30009	W30019	W30029	W30039	W30049		W30999
Stepping flag	R/W	D/E	R30000	R30010	R30020	R30030	R30040		R30990
Stepping activation	R/W	D/E	R30001	R30011	R30021	R30031	R30041		R30991

A.8 List of special registers (W, R) for SNC

■ SNC system control (W)

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Command control code	R/W	D/E	W31000	W33000	W35000	W37000	0: no command 8: execute reversing 9: stop reversing, continue machining 10: start machining 11: pause machining 12: resume machining (only valid when machining was paused) 13: stop machining 14: start the procedure for tool length measurement* 15: stop the procedure for tool length measurement*
No. of machining platform to be processed	R/W	D/E	W31002	W33002	W35002	W37002	No. of machining platform that is waiting to be processed of the SNC.
No. of machining platform in process	R	D/E	W31003	W33003	W35003	W37003	No. of machining platform that is in processing of the SNC.
Ignore NC code	R/W	D/E	W31004	W33004	W35004	W37004	Bit1: ignore /0 Bit2: ignore /1 Bit3: ignore /2 Bit4: ignore /3 Bit5: ignore /4 Bit6: ignore /5 Bit7: ignore /6 Bit8: ignore /7 Bit9: ignore /8 Bit10: ignore /9
Current tool No. of the spindle	R	D/E	W31005	W33005	W35005	W37005	Tool number of the SNC spindle: After the tool change is completed, set the relays R32998, 34998, 36998, and 38998 to on. Then the system will automatically write the tool No. for the tool change based on the T-code number.
Change tool and set the tool No. manually	R/W	D/E	W31006	W33006	W35006	W37006	When tool change is completed in manual mode: SNC1: trigger R32999 bit and W31006 will be written to W31005. SNC2: trigger R34999 bit and W31006 will be written to W33005. SNC3: trigger R36999 bit and W31006 will be written to W35005. SNC4: trigger R38999 bit and W31006 will be written to W37005.

A

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Skip activation	R/W	D/E	W31007	W33007	W35007	W37007	When the SNC receives the activation command, if this parameter is not set to 0 and the corresponding jump bit is on, then the G-code before the line set in this parameter will not be executed. 0: disable (default). 1: activate line number.
	R/W	D/E	W31008	W33008	W35008	W37008	
Error code	R	D/E	W31009	W33009	W35009	W37009	SNC error code: refer to Section 9.5 for detailed descriptions of the error codes.
Error type	R	D/E	W31010	W33010	W35010	W37010	SNC error type: refer to Section 9.5 for detailed descriptions of the error codes.
Operation status	R	D/E	W31011	W33011	W35011	W37011	SNC operation status: 0: stop 1: pause 2: running
File path (128 registers are applied)	R/W	D/E	W31100	W33100	W35100	W37100	G-code file path that will be executed by the SNC. (Apply a total of 128 registers within the range of W31100 - W31227.)

Note: Z-axis servo drive supports ASD-A2-****-F and ASD-A2-****-F for using the tool length automatic measurement function.

■ SNC system control (R)

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Disabling bit for G-code pre-detection function	R/W	D/E	R32950	R34950	R36950	R38950	Check for NC code error before starting the SNC (pre-detection function). On: disable pre-detection function.
X-axis motion limitation	R/W	D/E	R32971	R34971	R36971	R38971	Off: disable the axial motion limit (default). On: limit the axial motion.
Y-axis motion limitation	R/W	D/E	R32972	R34972	R36972	R38972	
Z-axis motion limitation	R/W	D/E	R32973	R34973	R36973	R38973	
A-axis motion limitation	R/W	D/E	R32974	R34974	R36974	R38974	
B-axis motion limitation	R/W	D/E	R32975	R34975	R36975	R38975	
C-axis motion limitation	R/W	D/E	R32976	R34976	R36976	R38976	
U-axis motion limitation	R/W	D/E	R32977	R34977	R36977	R38977	
V-axis motion limitation	R/W	D/E	R32978	R34978	R36978	R38978	
W-axis motion limitation	R/W	D/E	R32979	R34979	R36979	R38979	
Enabling bit for single-step execution	R/W	D/E	R32981	R34981	R36981	R38981	Set this function before the SNC starts to operate. Off: execute G-code continuously (default). On: the system executes one single line of G-code at a time. When it pauses, you need to trigger the single-step execution bit again to execute another line.
Activation bit for single-step execution	R/W	D/E	R32982	R34982	R36982	R38982	
Completion bit for spindle tool change	R/W	D/E	R32998	R34998	R36998	R38998	The SNC tool change function is controlled by the PLC. When the PLC completes the tool change function, the spindle tool change completion bit is triggered, and the SNC will continue to replace the current tool number and complete the set path.

■ M-code, T-code, and external macro

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Control code	R/W	D/E	W31015	W33015	W35015	W37015	Write the file number that is ready for calling. Then, write the control code according to the types of O, T macros, and the SNC starts to execute the external macro procedure. If the execution failed, the control code automatically changes to 99. Control command: 0: none. 1: call ""O"" macro. 2: call ""T"" macro. Error return: 98: skipping lines to call external macro is not supported. 99: no file is found.
File number	R/W	D/E	W31016	W33016	W35016	W37016	
M-code 00	R/W	D/E	R31000	R33000	R35000	R37000	When the SNC encounters M-code and T-code when running G-code, the corresponding R relay will be set, and the SNC path will pause to wait for the PLC to process. When the PLC completes processing M-code and T-code, the corresponding R relay must be cleared, and the function flow must be added after the SNC is completed. (Some M-codes are set by default, refer to Section 9.3.2 for more details.)
≀			≀	≀	≀	≀	
M-code 999			R31999	R33999	R35999	R37999	
T-code 01	R/W	D/E	R32001	R34001	R36001	R38001	
≀			≀	≀	≀	≀	
T-code 100			R32100	R34100	R36100	R38100	

■ External macro register (W)

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Write to #1833	R/W	D/E	W32301	W34301	W36301	W38301	
Write to #1834	R/W	D/E	W32302	W34302	W36302	W38302	
Write to #1835	R/W	D/E	W32303	W34303	W36303	W38303	
Write to #1836	R/W	D/E	W32304	W34304	W36304	W38304	
Write to #1837	R/W	D/E	W32305	W34305	W36305	W38305	
Write to #1838	R/W	D/E	W32306	W34306	W36306	W38306	
Write to #1839	R/W	D/E	W32307	W34307	W36307	W38307	
Write to #1840	R/W	D/E	W32308	W34308	W36308	W38308	Write the value of the PLC special register to the external macro register.
Write to #1841	R/W	D/E	W32309	W34309	W36309	W38309	
Write to #1842	R/W	D/E	W32310	W34310	W36310	W38310	
Write to #1843	R/W	D/E	W32311	W34311	W36311	W38311	
Write to #1844	R/W	D/E	W32312	W34312	W36312	W38312	
Write to #1845	R/W	D/E	W32313	W34313	W36313	W38313	
Write to #1846	R/W	D/E	W32314	W34314	W36314	W38314	
Write to #1847	R/W	D/E	W32315	W34315	W36315	W38315	
Write to #1848	R/W	D/E	W32316	W34316	W36316	W38316	

Name	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Read #1896	R	D/E	W32317	W34317	W36317	W38317	Read the value of the external macro register through the PLC special register.
Read #1897	R	D/E	W32318	W34318	W36318	W38318	
Read #1898	R	D/E	W32319	W34319	W36319	W38319	
Read #1899	R	D/E	W32320	W34320	W36320	W38320	
Read #1900	R	D/E	W32321	W34321	W36321	W38321	
Read #1901	R	D/E	W32322	W34322	W36322	W38322	
Read #1902	R	D/E	W32323	W34323	W36323	W38323	
Read #1903	R	D/E	W32324	W34324	W36324	W38324	
Read #1904	R	D/E	W32325	W34325	W36325	W38325	
Read #1905	R	D/E	W32326	W34326	W36326	W38326	
Read #1906	R	D/E	W32327	W34327	W36327	W38327	
Read #1907	R	D/E	W32328	W34328	W36328	W38328	
Read #1908	R	D/E	W32329	W34329	W36329	W38329	
Read #1909	R	D/E	W32330	W34330	W36330	W38330	
Read #1910	R	D/E	W32331	W34331	W36331	W38331	
Read #1911	R	D/E	W32332	W34332	W36332	W38332	

■ External macro relay (R)

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Write to #1801	R/W	D/E	R32301	R34301	R36301	R38301	Write the value of the PLC special relay to the external macro relay.
Write to #1802	R/W	D/E	R32302	R34302	R36302	R38302	
Write to #1803	R/W	D/E	R32303	R34303	R36303	R38303	
Write to #1804	R/W	D/E	R32304	R34304	R36304	R38304	
Write to #1805	R/W	D/E	R32305	R34305	R36305	R38305	
Write to #1806	R/W	D/E	R32306	R34306	R36306	R38306	
Write to #1807	R/W	D/E	R32307	R34307	R36307	R38307	
Write to #1808	R/W	D/E	R32308	R34308	R36308	R38308	
Write to #1809	R/W	D/E	R32309	R34309	R36309	R38309	
Write to #1810	R/W	D/E	R32310	R34310	R36310	R38310	
Write to #1811	R/W	D/E	R32311	R34311	R36311	R38311	
Write to #1812	R/W	D/E	R32312	R34312	R36312	R38312	
Write to #1813	R/W	D/E	R32313	R34313	R36313	R38313	
Write to #1814	R/W	D/E	R32314	R34314	R36314	R38314	
Write to #1815	R/W	D/E	R32315	R34315	R36315	R38315	
Write to #1816	R/W	D/E	R32316	R34316	R36316	R38316	
Write to #1817	R/W	D/E	R32317	R34317	R36317	R38317	
Write to #1818	R/W	D/E	R32318	R34318	R36318	R38318	
Write to #1819	R/W	D/E	R32319	R34319	R36319	R38319	
Write to #1820	R/W	D/E	R32320	R34320	R36320	R38320	
Write to #1821	R/W	D/E	R32321	R34321	R36321	R38321	

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Write to #1822	R/W	D/E	R32322	R34322	R36322	R38322	
Write to #1823	R/W	D/E	R32323	R34323	R36323	R38323	
Write to #1824	R/W	D/E	R32324	R34324	R36324	R38324	
Write to #1825	R/W	D/E	R32325	R34325	R36325	R38325	
Write to #1826	R/W	D/E	R32326	R34326	R36326	R38326	
Write to #1827	R/W	D/E	R32327	R34327	R36327	R38327	
Write to #1828	R/W	D/E	R32328	R34328	R36328	R38328	
Write to #1829	R/W	D/E	R32329	R34329	R36329	R38329	
Write to #1830	R/W	D/E	R32330	R34330	R36330	R38330	
Write to #1831	R/W	D/E	R32331	R34331	R36331	R38331	
Write to #1832	R/W	D/E	R32332	R34332	R36332	R38332	
Read #1864	R	D/E	R32334	R34334	R36334	R38334	
Read #1865	R	D/E	R32335	R34335	R36335	R38335	
Read #1866	R	D/E	R32336	R34336	R36336	R38336	
Read #1867	R	D/E	R32337	R34337	R36337	R38337	
Read #1868	R	D/E	R32338	R34338	R36338	R38338	
Read #1869	R	D/E	R32339	R34339	R36339	R38339	
Read #1870	R	D/E	R32340	R34340	R36340	R38340	
Read #1871	R	D/E	R32341	R34341	R36341	R38341	
Read #1872	R	D/E	R32342	R34342	R36342	R38342	
Read #1873	R	D/E	R32343	R34343	R36343	R38343	
Read #1841	R	D/E	R32344	R34344	R36344	R38344	
Read #1875	R	D/E	R32345	R34345	R36345	R38345	
Read #1876	R	D/E	R32346	R34346	R36346	R38346	
Read #1877	R	D/E	R32347	R34347	R36347	R38347	
Read #1878	R	D/E	R32348	R34348	R36348	R38348	Read the value of the external macro relay through the PLC special relay.
Read #1879	R	D/E	R32349	R34349	R36349	R38349	
Read #1880	R	D/E	R32350	R34350	R36350	R38350	
Read #1881	R	D/E	R32351	R34351	R36351	R38351	
Read #1882	R	D/E	R32352	R34352	R36352	R38352	
Read #1883	R	D/E	R32353	R34353	R36353	R38353	
Read #1884	R	D/E	R32354	R34354	R36354	R38354	
Read #1885	R	D/E	R32355	R34355	R36355	R38355	
Read #1886	R	D/E	R32356	R34356	R36356	R38356	
Read #1887	R	D/E	R32357	R34357	R36357	R38357	
Read #1888	R	D/E	R32358	R34358	R36358	R38358	
Read #1889	R	D/E	R32359	R34359	R36359	R38359	
Read #1890	R	D/E	R32360	R34360	R36360	R38360	
Read #1891	R	D/E	R32361	R34361	R36361	R38361	
Read #1892	R	D/E	R32362	R34362	R36362	R38362	
Read #1893	R	D/E	R32363	R34363	R36363	R38363	

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Read #1894	R	D/E	R32364	R34364	R36364	R38364	
Read #1895	R	D/E	R32365	R34365	R36365	R38365	

A

- Spindle speed control

Name	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Spindle speed control	R	D/E	W31020	W33020	W35020	W37020	When the SNC interpreter reads the S parameter, the number after the S parameter will be saved to the register.

- Read tool information

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Control code	R/W	D/E	W31029	W33029	W35029	W37029	Read tool information. Control command: 0: none. 1: read tool information. Error return: 99: reading failed.
Target tool number	R/W	D/E	W31030	W33030	W35030	W37030	Read the target tool number (1 - 100) in advance.
Read tool length	R	D/E	W31031	W33031	W35031	W37031	Read tool length.
	R	D/E	W31032	W33032	W35032	W37032	
Read tool radius	R	D/E	W31033	W33033	W35033	W37033	Read tool radius.
	R	D/E	W31034	W33034	W35034	W37034	
Read the tool offset value of X-coordinate	R	D/E	W31035	W33035	W35035	W37035	Read the tool offset value of X-coordinate.
	R	D/E	W31036	W33036	W35036	W37036	
Read the tool offset value of Y-coordinate	R	D/E	W31037	W33037	W35037	W37037	Read the tool offset value of Y-coordinate.
	R	D/E	W31038	W33038	W35038	W37038	
Read the tool offset value of Z-coordinate	R	D/E	W31039	W33039	W35039	W37039	Read the tool offset value of Z-coordinate.
	R	D/E	W31040	W33040	W35040	W37040	
Triggering bit for tool length record	R	D/E	R32940	R34940	R36940	R38940	When the differential signal of this bit switching from off to on is generated, the system will use the current Z axis machine coordinate record as the tool length information for the current tool number.

A

■ Error type

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Total error number	R	D/E	W32106	W34106	W36106	W38106	Total error number of G-code
Total warning number	R	D/E	W32108	W34108	W36108	W38108	Total warning number of G-code
Axial error alarm	R	D/E	W32161	W34161	W36161	W38161	Axial error alarm.
G-code error number	R	D/E	W32162	W34162	W36162	W38162	Refer to Section 9.5 for detailed descriptions of the error codes.
Software limit trigger	R	D/E	W32163	W34163	W36163	W38163	Software limit is triggered.
Wrong API value	R	D/E	W32164	W34164	W36164	W38164	Wrong returned value of underlying API. Refer to Section 9.5 for detailed descriptions of the error codes.
Setting error	R	D/E	W32165	W34165	W36165	W38165	Display the setting error code. Refer to Section 9.5 for detailed descriptions of the error codes.
Device error	R	D/E	W32166	W34166	W36166	W38166	Display the device error code. Refer to Section 9.5 for detailed descriptions of the error codes.
System error code	R	D/E	W32170	W34170	W36170	W38170	Display the system error code. Refer to Section 9.5 for detailed descriptions of the error codes.
Tool error	R	D/E	W32171	W34171	W36171	W38171	Display the tool error code. Refer to Section 9.5 for detailed descriptions of the error codes.
Wrong line number of G-code	R	D/E	W32178	W34178	W36178	W38178	Display the wrong line number of G-code.
	R	D/E	W32179	W34179	W36179	W38179	

■ G-code interpreter

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Interpreted line number	R	D/E	W32180	W34180	W36180	W38180	The line number that has been interpreted by the G-code interpreter.
	R	D/E	W32181	W34181	W36181	W38181	
Executed line number	R	D/E	W32187	W34187	W36187	W38187	The line number that has been executed by G-code.
	R	D/E	W32188	W34188	W36188	W38188	
Total line number	R	D/E	W32193	W34193	W36193	W38193	Total line number of G-code.
	R	D/E	W32194	W34194	W36194	W38194	
Estimate total time spent	R	D/E	W32230	W34230	W36230	W38230	Estimate the total time spent for executing G-code.
Execution time	R	D/E	W32231	W34231	W36231	W38231	The time the G-code has been executed.
Remaining time	R	D/E	W32233	W34233	W36233	W38233	Estimate the remaining time for executing G-code.
Current execution rate (%)	R	D/E	W32236	W34236	W36236	W38236	The progress (0 - 100%) of G-code execution.

■ Speed control (W)

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Maximum speed limit of G00	R/W	D/E	W32478	W34478	W36478	W38478	Maximum speed percentage of G00 should work with R32996.
Current feed rate (%)	R	D/E	W32479	W34479	W36479	W38479	Current feed rate of the SNC operation.
Target feed rate (%)	R/W	D/E	W32480	W34480	W36480	W38480	Write the target feed rate percentage to the SNC. SNC 1: work with R32997 SNC 2: work with R34997 SNC 3: work with R36997 SNC 4: work with R38997
Current feeding speed (mm/min)	R	D/E	W32774	W34774	W36774	W38774	Read the current feeding speed of the SNC.
	R	D/E	W32775	W34775	W36775	W38775	
Reverse path reference speed (mm/min)	R/W	D/E	W32951	W34951	W36951	W38951	The speed setting when the SNC executes the reverse path. The F feed rate of G-code is invalid during the reverse path.

A

■ Speed control (R)

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Enabling bit for MPG simulation	R/W	D/E	R32995	R34995	R36995	R38995	When MPG simulation is disabled, the group feed rate of SNC will be automatically set to 0.
Enabling bit for G00 feed rate setting	R/W	D/E	R32996	R34996	R36996	R38996	On: enable G00 feed rate adjustment. You can adjust the feed rate with the registers below: SNC 1: W31026 SNC 2: W33026 SNC 3: W35026 SNC 4: W37026
Enabling bit for SNC feed rate setting	R/W	D/E	R32997	R34997	R36997	R38997	On: you can adjust the SNC feed rate with the registers below: SNC1: W32480 SNC1: W34480 SNC1: W36480 SNC1: W38480

■ Coordinates (W)

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
X-axis feedback position	R	D/E	W32520	W34520	W36520	W38520	The encoder feedback position of each axis (machine coordinate). (Feedback / Gear)
	R	D/E	W32521	W34521	W36521	W38521	
Y-axis feedback position	R	D/E	W32522	W34522	W36522	W38522	
	R	D/E	W32523	W34523	W36523	W38523	
Z-axis feedback position	R	D/E	W32524	W34524	W36524	W38524	
	R	D/E	W32525	W34525	W36525	W38525	
A-axis feedback position	R	D/E	W32526	W34526	W36526	W38526	
	R	D/E	W32527	W34527	W36527	W38527	
B-axis feedback position	R	D/E	W32528	W34528	W36528	W38528	
	R	D/E	W32529	W34529	W36529	W38529	
C-axis feedback position	R	D/E	W32530	W34530	W36530	W38530	
	R	D/E	W32531	W34531	W36531	W38531	
U-axis feedback position	R	D/E	W32532	W34532	W36532	W38532	
	R	D/E	W32533	W34533	W36533	W38533	
V-axis feedback position	R	D/E	W32534	W34534	W36534	W38534	
	R	D/E	W32535	W34535	W36535	W38535	
W-axis feedback position	R	D/E	W32536	W34536	W36536	W38536	
	R	D/E	W32537	W34537	W36537	W38537	
X-axis command position	R	D/E	W32538	W34538	W36538	W38538	Command position of each axis (machine coordinate). (Command / Gear)
	R	D/E	W32539	W34539	W36539	W38539	
Y-axis command position	R	D/E	W32540	W34540	W36540	W38540	
	R	D/E	W32541	W34541	W36541	W38541	

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Z-axis command position	R	D/E	W32542	W34542	W36542	W38542	
	R	D/E	W32543	W34543	W36543	W38543	
A-axis command position	R	D/E	W32544	W34544	W36544	W38544	
	R	D/E	W32545	W34545	W36545	W38545	
B-axis command position	R	D/E	W32546	W34546	W36546	W38546	
	R	D/E	W32547	W34547	W36547	W38547	
C-axis command position	R	D/E	W32548	W34548	W36548	W38548	
	R	D/E	W32549	W34549	W36549	W38549	
U-axis command position	R	D/E	W32550	W34550	W36550	W38550	
	R	D/E	W32551	W34551	W36551	W38551	
V-axis command position	R	D/E	W32552	W34552	W36552	W38552	
	R	D/E	W32553	W34553	W36553	W38553	
W-axis command position	R	D/E	W32554	W34554	W36554	W38554	
	R	D/E	W32555	W34555	W36555	W38555	
X-axis workpiece coordinate	R	D/E	W32556	W34556	W36556	W38556	Workpiece coordinate of each axis. (Workpiece coordinate = Machine coordinate - Set coordinate of working plane)
	R	D/E	W32557	W34557	W36557	W38557	
Y-axis workpiece coordinate	R	D/E	W32558	W34558	W36558	W38558	
	R	D/E	W32559	W34559	W36559	W38559	
Z-axis workpiece coordinate	R	D/E	W32560	W34560	W36560	W38560	
	R	D/E	W32561	W34561	W36561	W38561	
A-axis workpiece coordinate	R	D/E	W32562	W34562	W36562	W38562	
	R	D/E	W32563	W34563	W36563	W38563	
B-axis workpiece coordinate	R	D/E	W32564	W34564	W36564	W38564	
	R	D/E	W32565	W34565	W36565	W38565	
C-axis workpiece coordinate	R	D/E	W32566	W34566	W36566	W38566	
	R	D/E	W32567	W34567	W36567	W38567	
U-axis workpiece coordinate	R	D/E	W32568	W34568	W36568	W38568	
	R	D/E	W32569	W34569	W36569	W38569	
V-axis workpiece coordinate	R	D/E	W32570	W34570	W36570	W38570	
	R	D/E	W32571	W34571	W36571	W38571	
W-axis workpiece coordinate	R	D/E	W32572	W34572	W36572	W38572	
	R	D/E	W32573	W34573	W36573	W38573	
X-axis remaining distance	R	D/E	W32574	W34574	W36574	W38574	Remaining distance from the target position. (Remaining distance = Target coordinate - Machine coordinate)
	R	D/E	W32575	W34575	W36575	W38575	
Y-axis remaining distance	R	D/E	W32576	W34576	W36576	W38576	
	R	D/E	W32577	W34577	W36577	W38577	
Z-axis remaining distance	R	D/E	W32578	W34578	W36578	W38578	
	R	D/E	W32579	W34579	W36579	W38579	
A-axis remaining distance	R	D/E	W32580	W34580	W36580	W38580	
	R	D/E	W32581	W34581	W36581	W38581	
B-axis remaining distance	R	D/E	W32582	W34582	W36582	W38582	
	R	D/E	W32583	W34583	W36583	W38583	

A

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
C-axis remaining distance	R	D/E	W32584	W34584	W36584	W38584	
	R	D/E	W32585	W34585	W36585	W38585	
U-axis remaining distance	R	D/E	W32586	W34586	W36586	W38586	
	R	D/E	W32587	W34587	W36587	W38587	
V-axis remaining distance	R	D/E	W32588	W34588	W36588	W38588	
	R	D/E	W32589	W34589	W36589	W38589	
W-axis remaining distance	R	D/E	W32590	W34590	W36590	W38590	
	R	D/E	W32591	W34591	W36591	W38591	
Starting point of relative coordinate on X-axis	R	D/E	W32718	W34718	W36718	W38718	Starting point of relative coordinate. (The end position of the last G-code.)
	R	D/E	W32719	W34719	W36719	W38719	
Starting point of relative coordinate on Y-axis	R	D/E	W32720	W34720	W36720	W38720	
	R	D/E	W32721	W34721	W36721	W38721	
Starting point of relative coordinate on Z-axis	R	D/E	W32722	W34722	W36722	W38722	
	R	D/E	W32723	W34723	W36723	W38723	
Starting point of relative coordinate on A-axis	R	D/E	W32724	W34724	W36724	W38724	
	R	D/E	W32725	W34725	W36725	W38725	
Starting point of relative coordinate on B-axis	R	D/E	W32726	W34726	W36726	W38726	
	R	D/E	W32727	W34727	W36727	W38727	
Starting point of relative coordinate on C-axis	R	D/E	W32728	W34728	W36728	W38728	
	R	D/E	W32729	W34729	W36729	W38729	
Starting point of relative coordinate on U-axis	R	D/E	W32730	W34730	W36730	W38730	
	R	D/E	W32731	W34731	W36731	W38731	
Starting point of relative coordinate on V-axis	R	D/E	W32732	W34732	W36732	W38732	
	R	D/E	W32733	W34733	W36733	W38733	
Starting point of relative coordinate on W-axis	R	D/E	W32734	W34734	W36734	W38734	
	R	D/E	W32735	W34735	W36735	W38735	
Relative coordinate value of X-axis	R	D/E	W32736	W34736	W36736	W38736	Relative coordinate value of each axis. (Relative coordinate = Machine coordinate - Starting point of relative coordinate)
	R	D/E	W32737	W34737	W36737	W38737	
Relative coordinate value of Y-axis	R	D/E	W32738	W34738	W36738	W38738	
	R	D/E	W32739	W34739	W36739	W38739	
Relative coordinate value of Z-axis	R	D/E	W32740	W34740	W36740	W38740	
	R	D/E	W32741	W34741	W36741	W38741	
Relative coordinate value of A-axis	R	D/E	W32742	W34742	W36742	W38742	
	R	D/E	W32743	W34743	W36743	W38743	
Relative coordinate value of B-axis	R	D/E	W32744	W34744	W36744	W38744	
	R	D/E	W32745	W34745	W36745	W38745	
Relative coordinate value of C-axis	R	D/E	W32746	W34746	W36746	W38746	
	R	D/E	W32747	W34747	W36747	W38747	
Relative coordinate value of U-axis	R	D/E	W32748	W34748	W36748	W38748	
	R	D/E	W32749	W34749	W36749	W38749	
Relative coordinate value of V-axis	R	D/E	W32750	W34750	W36750	W38750	
	R	D/E	W32751	W34751	W36751	W38751	

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Relative coordinate value of W-axis	R	D/E	W32752	W34752	W36752	W38752	
	R	D/E	W32753	W34753	W36753	W38753	
Current coordinate system	R	D/E	W32754	W34754	W36754	W38754	The currently applied workpiece coordinate system (G54 - G59).

■ Coordinates (R)

Function	Property	Bus	SNC	SNC 2	SNC 3	SNC 4	Description
Starting point of relative coordinate on X-axis	R/W	D/E	R32986	R34986	R36986	R38986	This special relay detects the rising-edge pulse and sets the current position as the starting point of the relative coordinate, then this relay is set to off.
Starting point of relative coordinate on Y-axis	R/W	D/E	R32987	R34987	R36987	R38987	
Starting point of relative coordinate on Z-axis	R/W	D/E	R32988	R34988	R36988	R38988	
Starting point of relative coordinate on A-axis	R/W	D/E	R32989	R34989	R36989	R38989	
Starting point of relative coordinate on B-axis	R/W	D/E	R32990	R34990	R36990	R38990	
Starting point of relative coordinate on C-axis	R/W	D/E	R32991	R34991	R36991	R38991	
Starting point of relative coordinate on U-axis	R/W	D/E	R32992	R34992	R36992	R38992	
Starting point of relative coordinate on V-axis	R/W	D/E	R32993	R34993	R36993	R38993	
Starting point of relative coordinate on W-axis	R/W	D/E	R32994	R34994	R36994	R38994	

(This page is intentionally left blank.)

Homing Mode B

This chapter helps with understanding the definitions of different homing modes.

B

B.1 List of homing modes

Mode	Definition of homing origin	Processing of the limit signals
1	After reaching the negative limit, the first Z pulse when moving in the forward direction.	Touching the positive limit is invalid.
2	After reaching the positive limit, the first Z pulse when moving in the reverse direction	Touching the negative limit is invalid.
3	Start in the forward direction and look for the first Z pulse after leaving the home switch in reverse direction.	Touching the limit signal in the same direction is regarded as error.
4	Start in the forward direction and look for the first Z pulse after reaching the home switch in forward direction.	
5	Start in the reverse direction and look for the first Z pulse after leaving the home switch in forward direction.	
6	Start in the reverse direction and look for the first Z pulse after reaching the home switch in reverse direction.	
7	Start in the forward direction and look for the first Z pulse after leaving the home switch in reverse direction.	Run in the reverse direction after reaching the limit in the same direction.
8	Start in the forward direction and look for the first Z pulse after reaching the home switch in forward direction.	
9	Start in the forward direction and look for the first Z pulse after reaching the home switch in reverse direction.	
10	Start in the forward direction and look for the first Z pulse after leaving the home switch in forward direction.	
11	Start in the reverse direction and look for the first Z pulse after leaving the home switch in forward direction.	
12	Start in the reverse direction and look for the first Z pulse after reaching the home switch in reverse direction.	
13	Start in the reverse direction and look for the first Z pulse after reaching the home switch in forward direction.	
14	Start in the reverse direction and look for the first Z pulse after leaving the home switch in reverse direction.	
15	Reserved	-
16	Reserved	-
17	The pulse reaching the negative limit when running in the reverse direction.	Touching the positive limit is invalid.
18	The pulse reaching the positive limit when running in the positive direction.	Touching the negative limit is invalid.
19	Start in the forward direction and look for the pulse after leaving the home switch in reverse direction.	Touching the limit signal in the same direction is regarded as error.
20	Same as 19.	
21	Start in the reverse direction and look for the pulse after leaving the home switch in forward direction.	
22	Same as 21.	
23	Start in the forward direction and look for the pulse after leaving the home switch in reverse direction.	Run in the reverse direction after reaching the limit in the same direction.
24	Same as 23.	
25	Start in the forward direction and look for the pulse after leaving the home switch in forward direction.	
26	Same as 25.	
27	Start in the reverse direction and look for the pulse after leaving the home switch in forward direction.	
28	Same as 27.	
29	Start in the reverse direction and look for the pulse after leaving the home switch in reverse direction.	
30	Same as 29.	

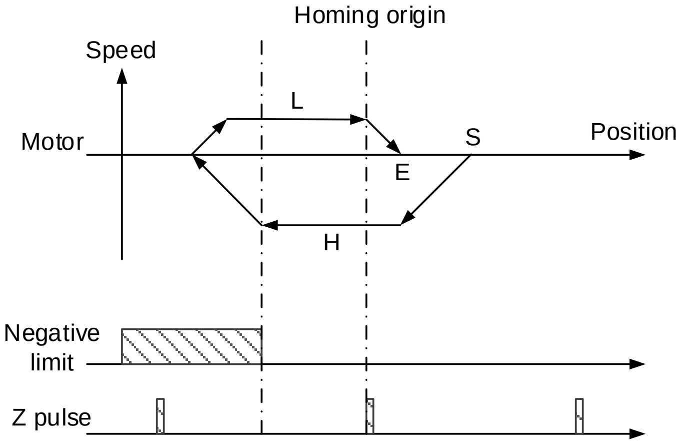
Mode	Definition of homing origin	Processing of the limit signals
31	Reserved	-
32	Reserved	-
33	The first Z pulse when moving in the reverse direction.	Touching the positive or negative limit is regarded as error.
34	The first Z pulse when moving in the forward direction.	
35	Set the current position as the new homing origin.	-

B

B.2 Description of homing modes

Mode 1

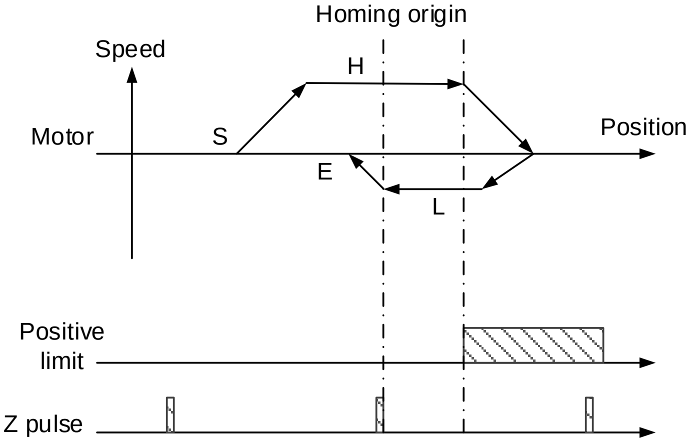
The motor runs in the reverse direction at high speed until it reaches the negative limit switch, then it decelerates and leaves the negative limit in the forward direction at low speed. The motor looks for the first Z pulse of the encoder, which is regarded as the new homing origin.



H: high speed (1^{st} speed)
L: low speed (2^{nd} speed)
S: starting point
E: end point
Z pulse: zero point of each encoder cycle

Mode 2

The motor runs in the forward direction at high speed until it reaches the positive limit switch, then it decelerates and leaves the positive limit in the reverse direction at low speed. The motor looks for the first Z pulse of the encoder, which is regarded as the new homing origin.

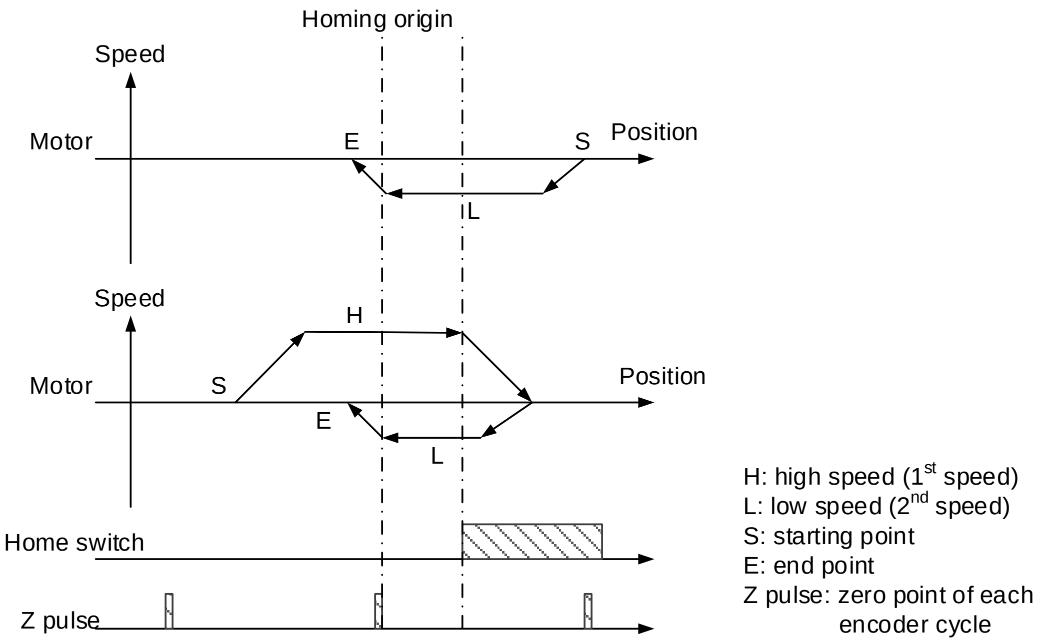


H: high speed (1^{st} speed)
L: low speed (2^{nd} speed)
S: starting point
E: end point
Z pulse: zero point of each encoder cycle

B

Mode 3

- Home switch is off: the motor runs in the forward direction at high speed until it reaches the home switch, then it decelerates and leaves the home switch in the reverse direction at low speed. The motor looks for the first Z pulse of the encoder, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the reverse direction at high speed until it leaves the home switch, then it looks for the first Z pulse of the encoder at low speed, which is regarded as the new homing origin.

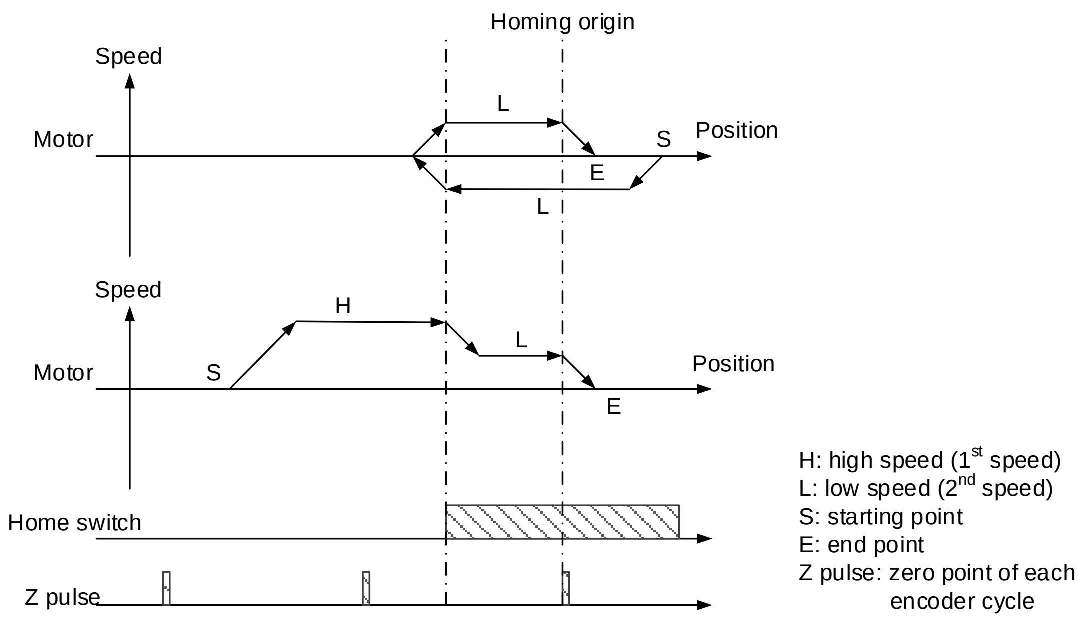


B

Mode 4

Mode 4 is similar to Mode 3 but with different moving directions after receiving the signal changes of the home switch.

- Home switch is off: the motor runs in the forward direction at high speed until it reaches the home switch, then it looks for the first Z pulse of the encoder at low speed, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the reverse direction at high speed until it leaves the home switch, then it decelerates and moves in the forward direction at low speed to reach the home switch again. The motor looks for the first Z pulse of the encoder, which is regarded as the new homing origin.

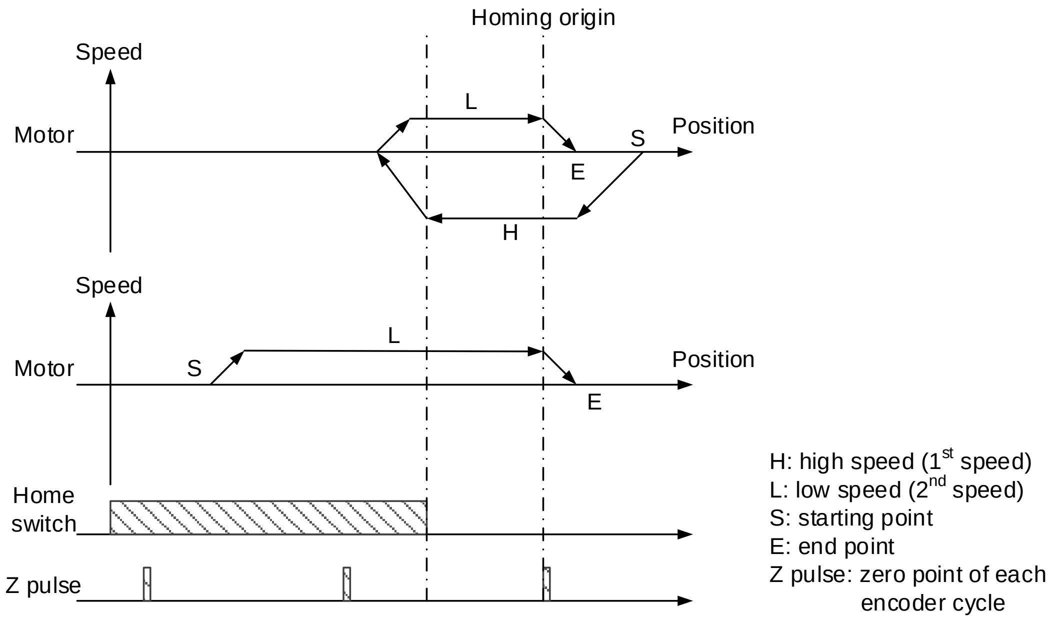


B

Mode 5

Mode 5 is similar to Mode 3 but with different initial moving directions.

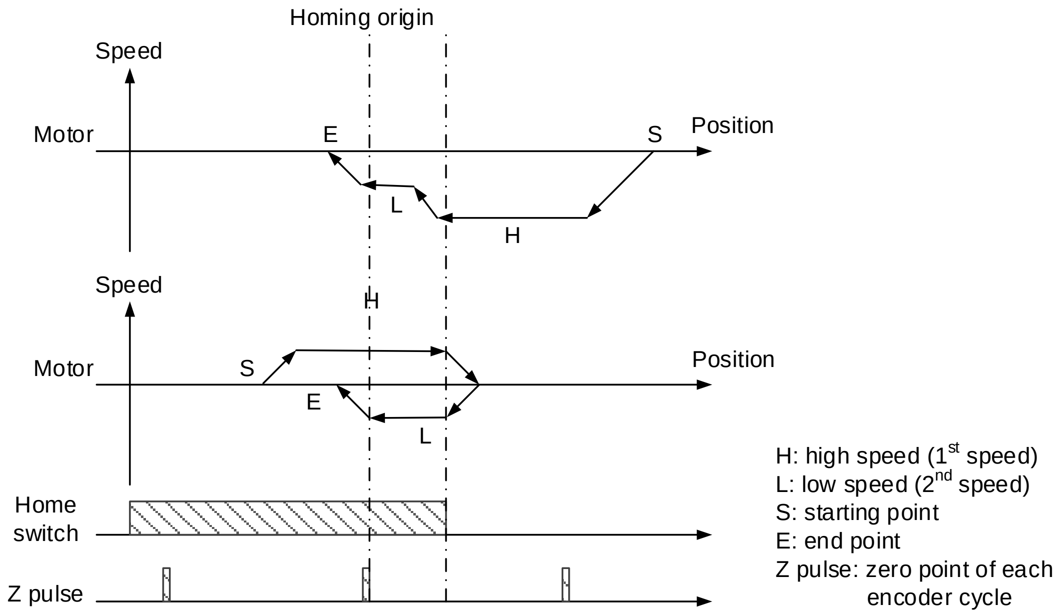
- Home switch is off: the motor runs in the reverse direction at high speed until it reaches the home switch, then it decelerates and leaves the home switch in the forward direction at low speed. The motor looks for the first Z pulse of the encoder, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the forward direction at low speed until it leaves the home switch, then it looks for the first Z pulse of the encoder at low speed, which is regarded as the new homing origin.



Mode 6

Mode 6 is similar to Mode 4 but with different initial moving directions.

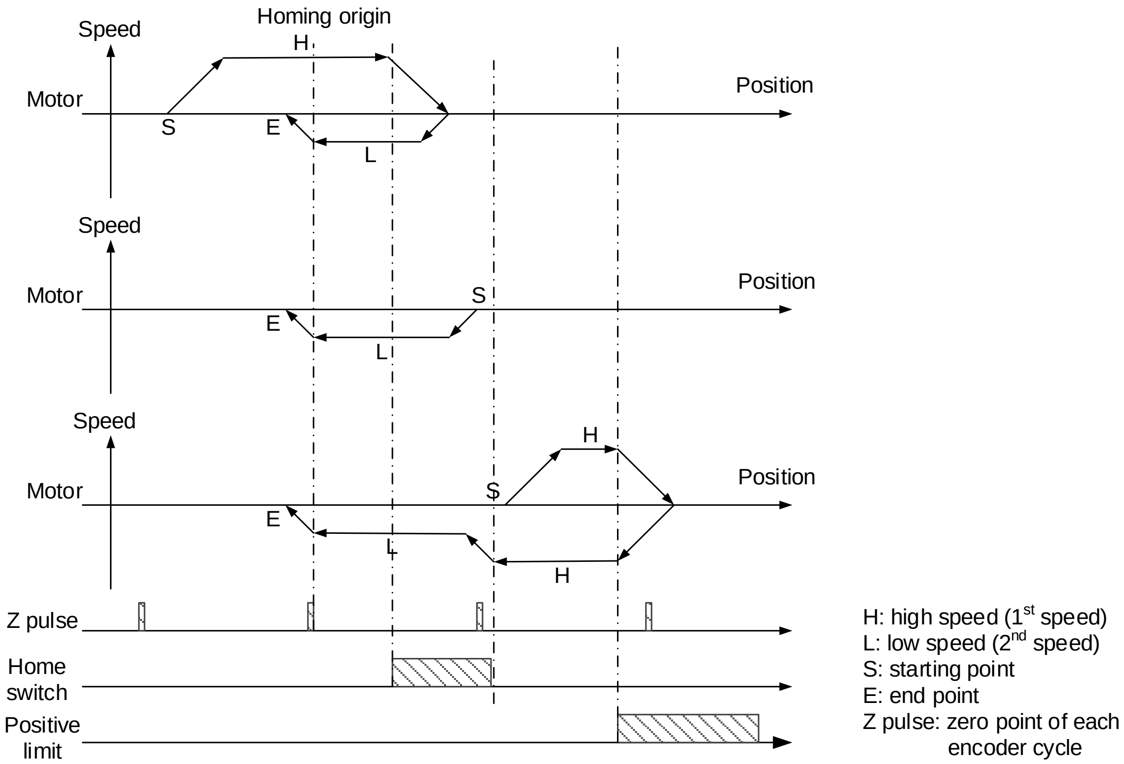
- Home switch is off: the motor runs in the reverse direction at high speed until it reaches the home switch, then it looks for the first Z pulse of the encoder at low speed, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the forward direction at high speed until it leaves the home switch, then it decelerates and moves in the reverse direction at low speed to reach the home switch again. The motor looks for the first Z pulse of the encoder, which is regarded as the new homing origin.



B

Mode 7

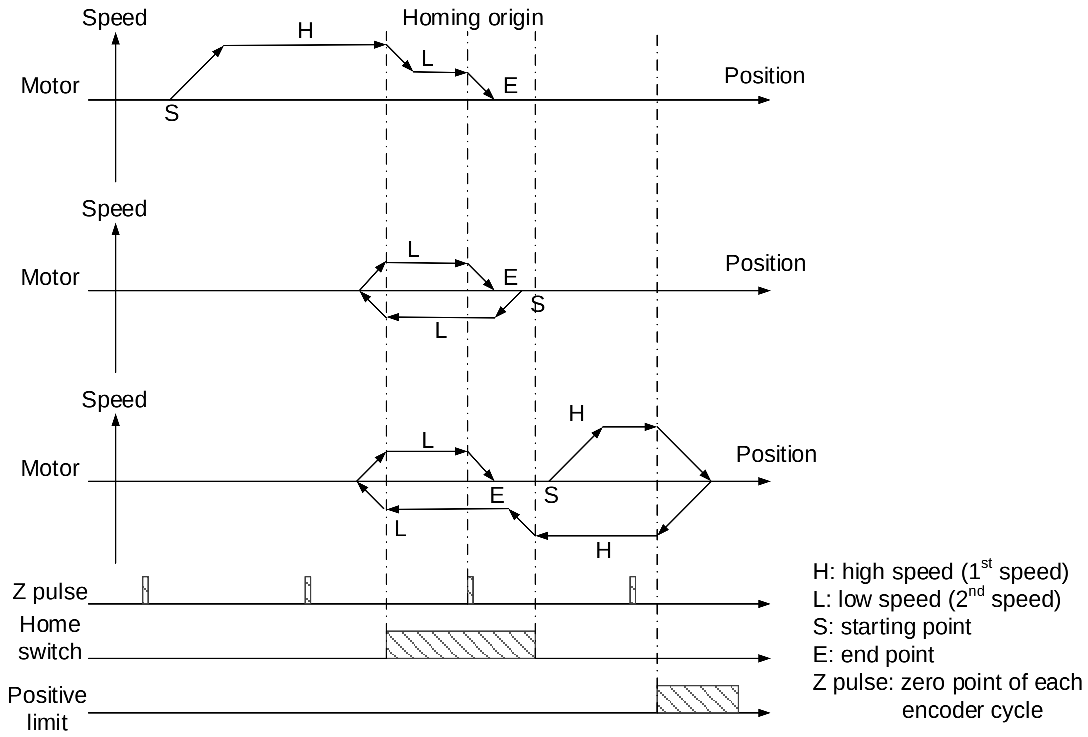
- Home switch is off: the motor runs in the forward direction at high speed until it reaches the home switch, then it decelerates and runs in the reverse direction at low speed until leaving the home switch. The motor then looks for the first Z pulse of the encoder at low speed, which is regarded as the new homing origin. If the motor reaches the positive limit before triggering the home switch, it will then move in the reverse direction until reaching the home switch. The motor switches to low speed and when it leaves the home switch, it looks for the first Z pulse of the encoder, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the reverse direction at low speed until it leaves the home switch, then it looks for the first Z pulse of the encoder at low speed, which is regarded as the new homing origin.



B

Mode 8

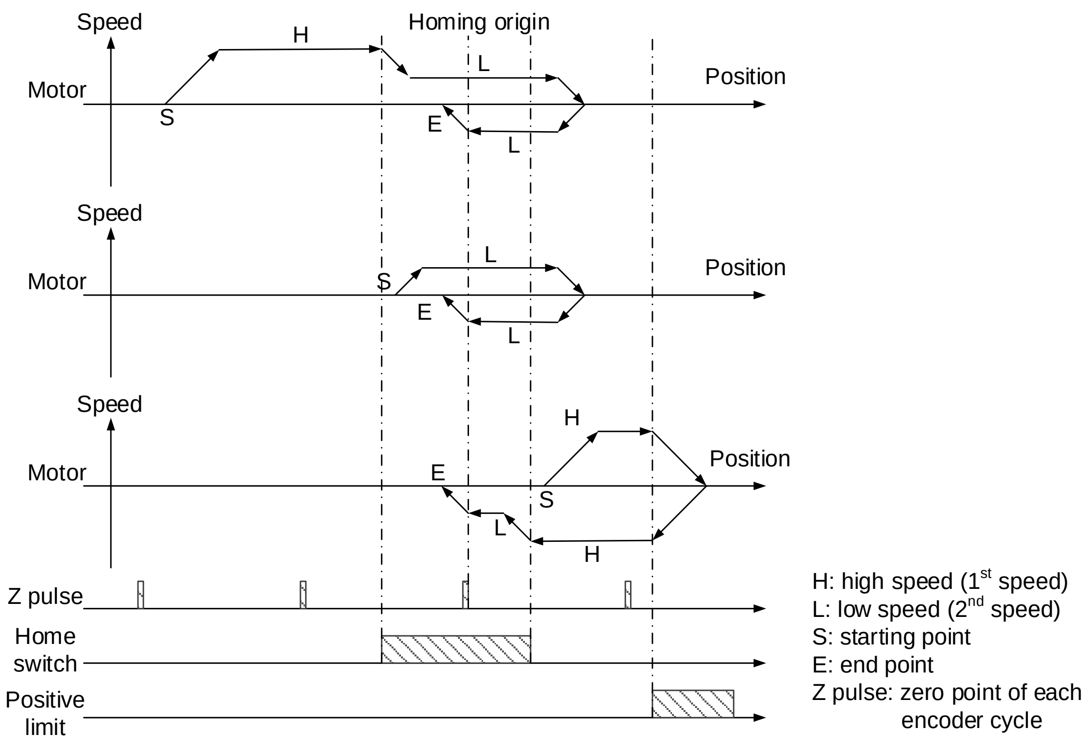
- Home switch is off: the motor runs in the forward direction at high speed until it reaches the home switch, then it switches to low speed to look for the first Z pulse of the encoder, which is regarded as the new homing origin. If the motor reaches the positive limit before triggering the home switch, it will then move in the reverse direction until reaching the home switch, and switch to low speed. After the motor leaves the home switch, it will move in the forward direction at low speed and look for the first Z pulse of the encoder after reaching the home switch, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the reverse direction at low speed until it leaves the home switch, then it decelerates and moves in the forward direction at low speed to reach the home switch again. The motor looks for the first Z pulse of the encoder, which is regarded as the new homing origin.



B

Mode 9

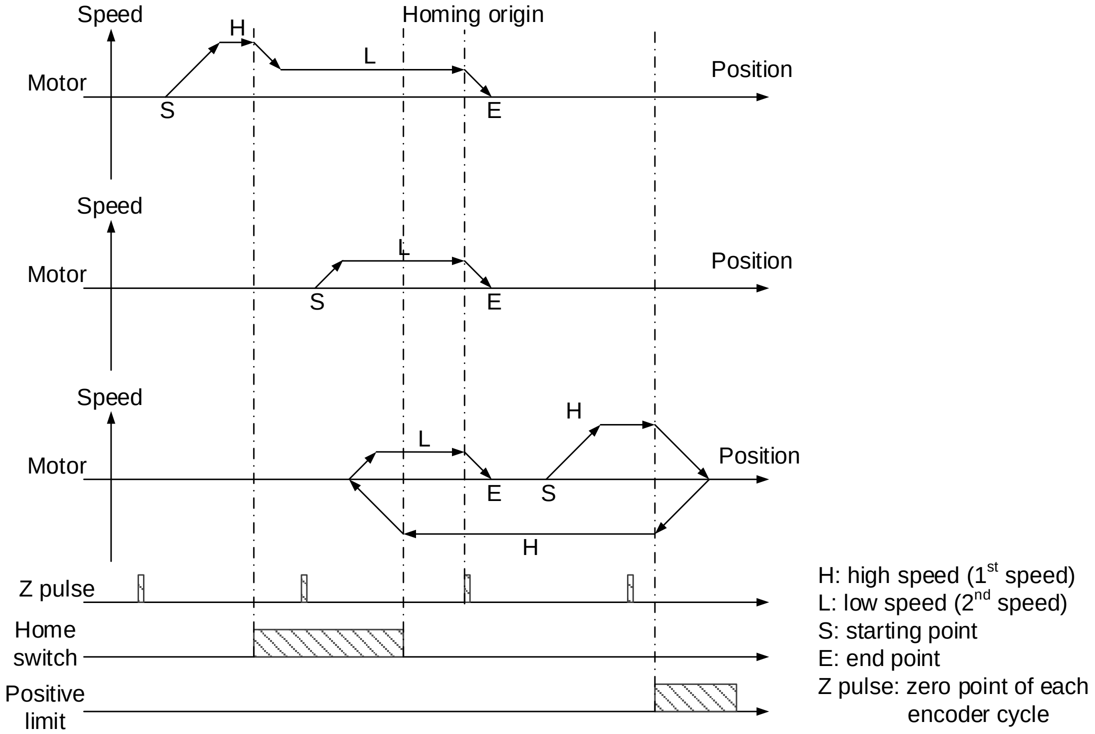
- Home switch is off: the motor runs in the forward direction at high speed until it reaches the home switch, then it switches to low speed and waits until leaving the home switch to look for the first Z pulse of the encoder after reaching the home switch in reverse direction at low speed, which is regarded as the new homing origin. If the motor reaches the positive limit before triggering the home switch, it will then move in the reverse direction until receiving the rising-edge pulse of the home switch, and it will move at low speed to look for the first Z pulse of the encoder after reaching the home switch, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the forward direction at low speed until it leaves the home switch, then it decelerates and moves in the reverse direction at low speed to look for the first Z pulse of the encoder after reaching the home switch, which is regarded as the new homing origin.



B

Mode 10

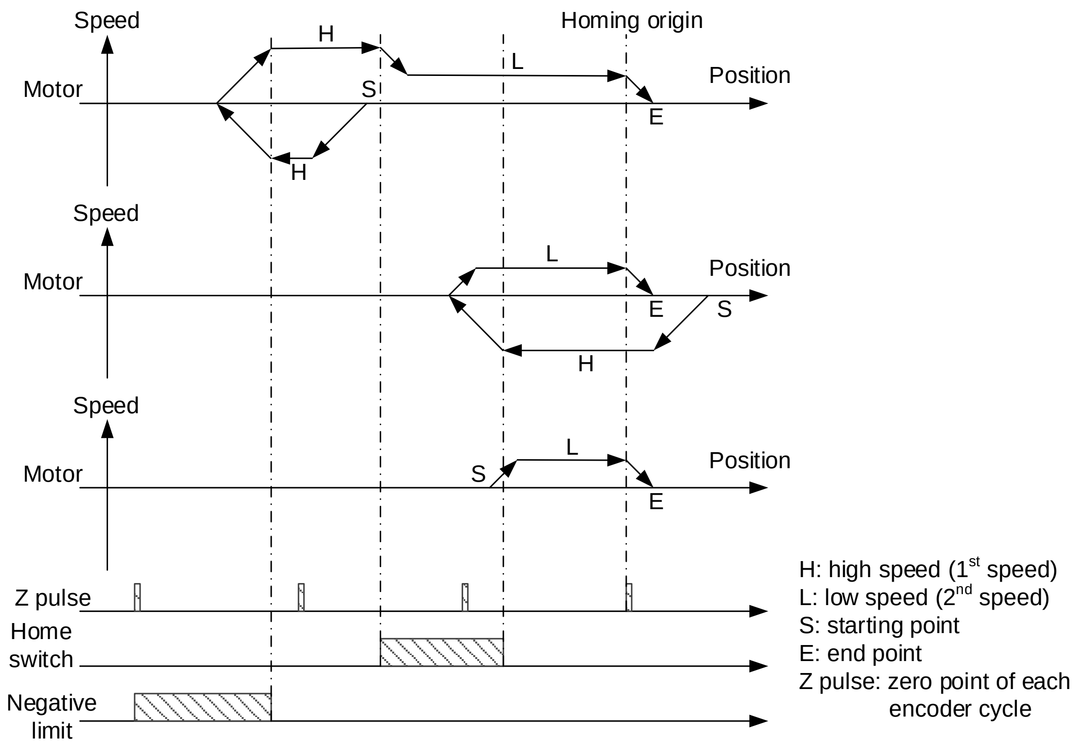
- Home switch is off: the motor runs in the forward direction at high speed until it reaches the home switch, then it switches to low speed and waits until leaving the home switch to look for the first Z pulse of the encoder, which is regarded as the new homing origin. If the motor reaches the positive limit before triggering the home switch, it will then move in the reverse direction until receiving the home switch signal, and switch to low speed to look for the first Z pulse of the encoder after reaching the home switch in forward direction, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the forward direction at low speed until it leaves the home switch, then it looks for the first Z pulse of the encoder, which is regarded as the new homing origin.



B

Mode 11

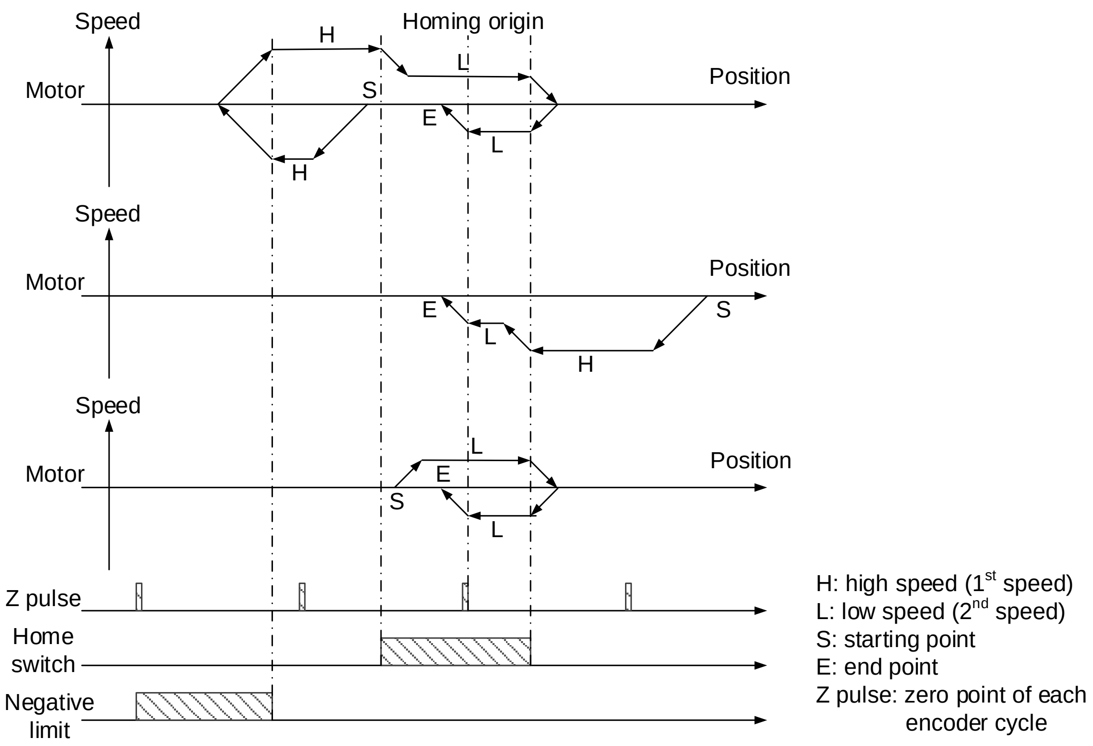
- Home switch is off: the motor runs in the reverse direction at high speed until it reaches the home switch, then it switches to forward direction at low speed until leaving the home switch to look for the first Z pulse of the encoder, which is regarded as the new homing origin. If the motor reaches the negative limit before triggering the home switch, it will then move in the reverse direction until reaching the home switch. The motor switches to low speed and when it leaves the home switch, it looks for the first Z pulse of the encoder, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the forward direction at low speed until it leaves the home switch, then it looks for the first Z pulse of the encoder, which is regarded as the new homing origin.



B

Mode 12

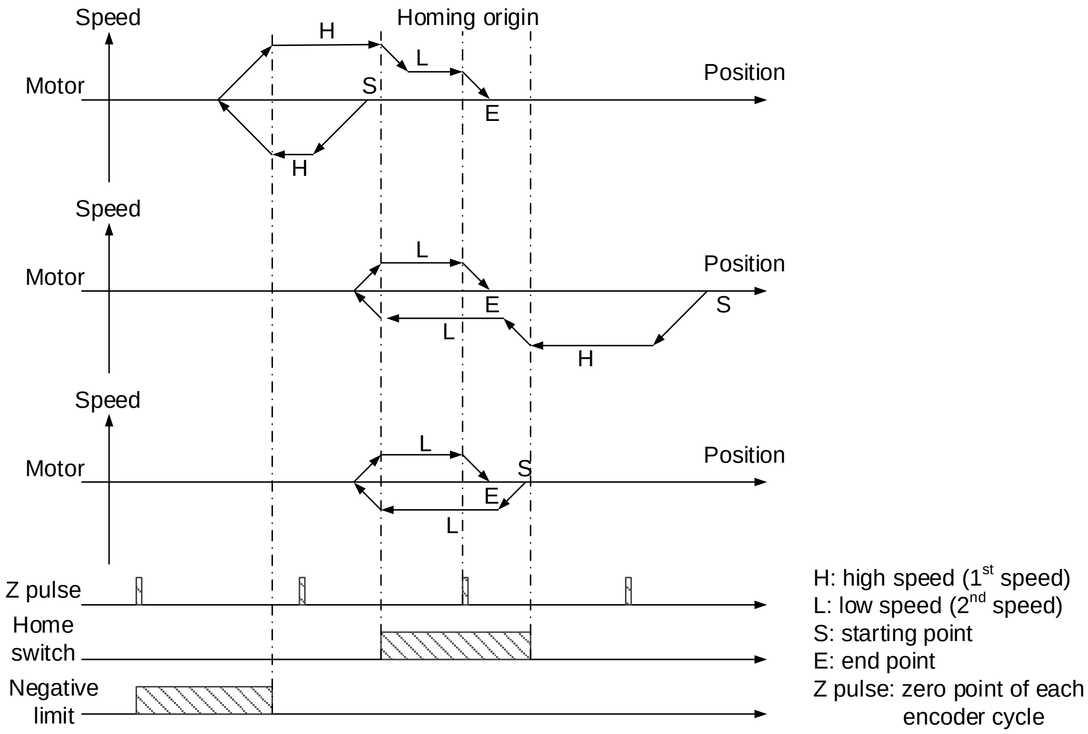
- Home switch is off: the motor runs in the reverse direction at high speed until it reaches the home switch, then it switches to low speed and looks for the first Z pulse of the encoder, which is regarded as the new homing origin. If the motor reaches the negative limit before triggering the home switch, it will then move in the reverse direction until reaching the home switch. The motor switches to low speed and when it leaves the home switch, it will move in the reverse direction again to reach the home switch and look for the first Z pulse of the encoder, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the forward direction at low speed until it leaves the home switch, then it moves in the reverse direction at low speed to reach the home switch and look for the first Z pulse of the encoder, which is regarded as the new homing origin.



B

Mode 13

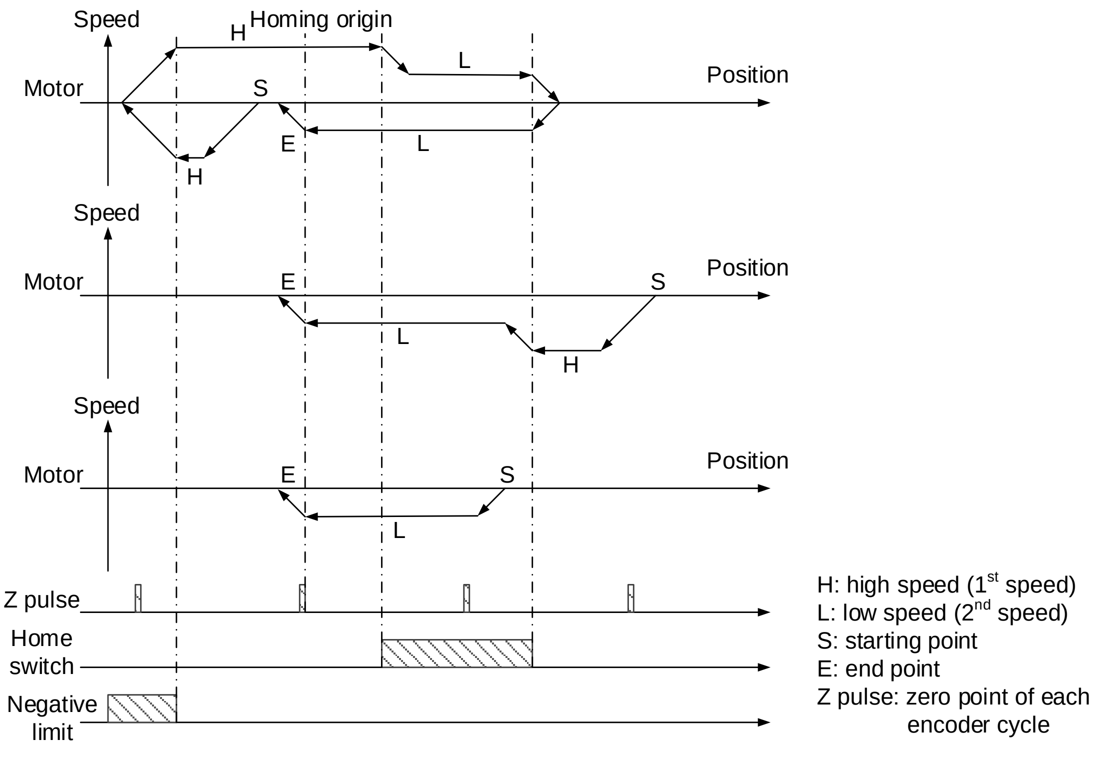
- Home switch is off: the motor runs in the reverse direction at high speed until it reaches the home switch, then it switches to low speed until leaving the home switch. The motor moves in forward direction at low speed to look for the first Z pulse of the encoder after reaching the home switch, which is regarded as the new homing origin. If the motor reaches the negative limit before triggering the home switch, it will then move in the reverse direction until reaching the home switch. The motor switches to low speed and looks for the first Z pulse of the encoder, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the reverse direction at low speed until it leaves the home switch, then it moves in the forward direction at low speed to reach the home switch and look for the first Z pulse of the encoder, which is regarded as the new homing origin.



B

Mode 14

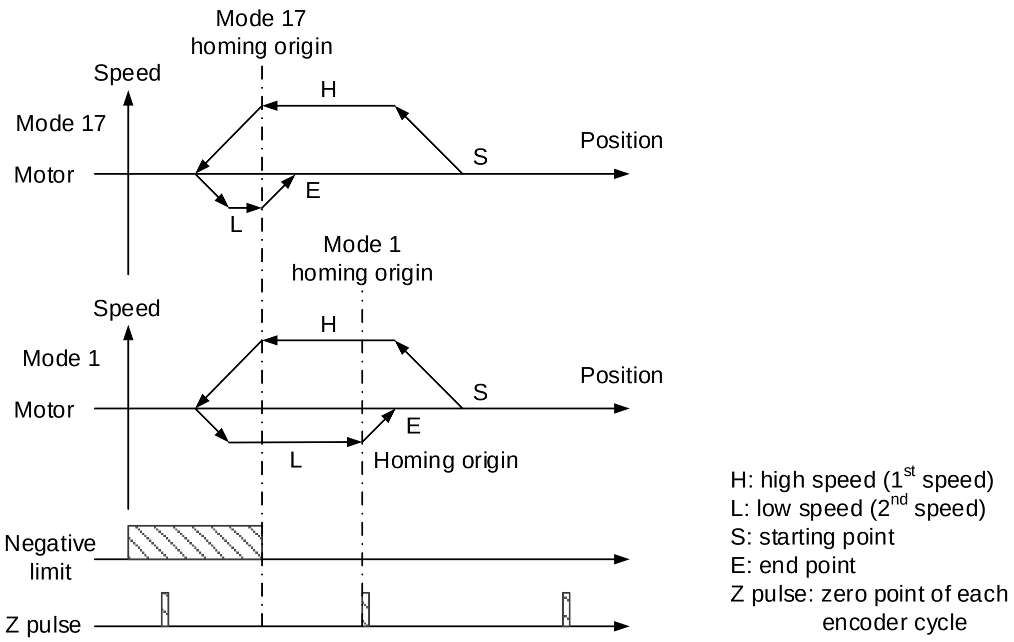
- Home switch is off: the motor runs in the reverse direction at high speed until it reaches the home switch, then it switches to low speed to look for the first Z pulse of the encoder after leaving the home switch, which is regarded as the new homing origin. If the motor reaches the negative limit before triggering the home switch, it will then move in the forward direction to reach the home switch. The motor switches to low speed and when it leaves the home switch, the motor moves in reverse direction at low speed to look for the first Z pulse of the encoder after reaching the home switch, which is regarded as the new homing origin.
- Home switch is on: the motor runs in the reverse direction at low speed until it leaves the home switch, then it looks for the first Z pulse of the encoder, which is regarded as the new homing origin.



B

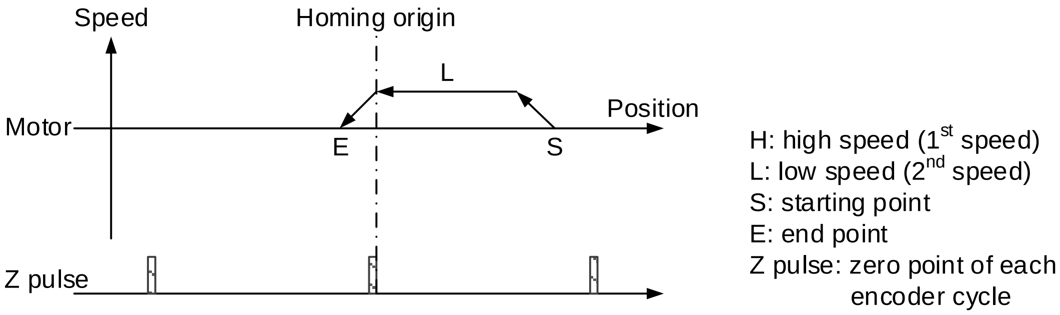
Modes 17 - 30

Modes 17 - 30 are similar to Modes 1 - 14 respectively with the following differences: for Modes 1 - 14, after receiving signals of the home switch or limits, the motor looks for the Z pulse and regards it as the new homing origin; whereas for Modes 17 - 30, the motor regards the switching signals of the home switch or limits as the new homing origin. Refer to the figure below for the differences between Mode 1 and Mode 17.



Mode 33

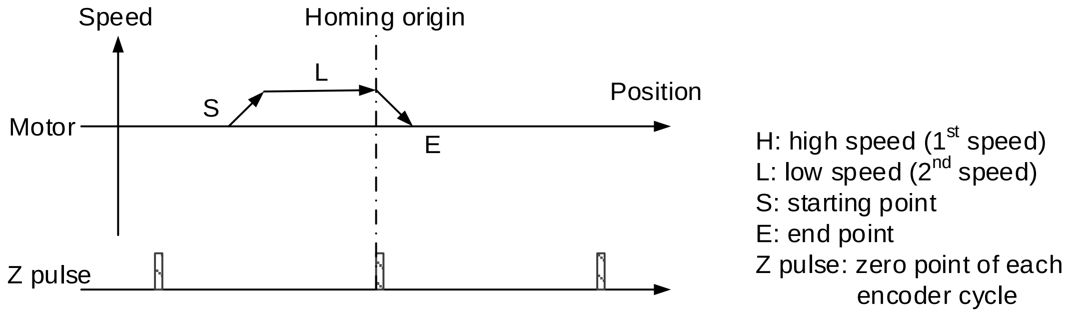
The motor runs in the reverse direction to look for the first Z pulse of the encoder, which is regarded as the new homing origin.



B

Mode 34

The motor runs in the forward direction to look for the first Z pulse of the encoder, which is regarded as the new homing origin.



Mode 35

Set the current position as the new homing origin.

(This page is intentionally left blank.)

Revision History

Release date	Version	Chapter	Revision contents
June, 2016	V1.0 (First edition)	-	-
December, 2018	V2.0 (Second edition)	1	Add description for the EtherCAT bus which is now supported by the software.
		2.1	Modify computer performance requirements. Delete supported product list.
		2.2	Remove .NET Framework from the installation process. Add installation options for DMCNET / EtherCAT.
		2.3	Add note for license authorization.
		3.1	Add setting description for EtherCAT bus with A2-E.
		3.2	Add description for the control panel. Update UI image and corresponding text content.
		5.1	Add corresponding registers of each bus type. Add notes about using the non-volatile function on Delta PAC and PC.
		5.2	Add description for the corresponding registers of the EtherCAT bus.
		6.1	Correct analog input / output instruction code error. Add the RSVP instruction. Correct the descriptions for some instructions. Add Section 6.1.8 Motion table.
		7.1	Correct the operator errors for the IF, LOOP, and WHILE instructions.
		7.2	Update UI image and corresponding text content.
		8.1	Modify the descriptions for Time Out and COM port. Update UI image and corresponding text content.
		8.2	Add the ranges for DX and DY. Delete Modbus function code 17.
		9.1	Add description for the EtherCAT bus which is now supported.
		9.2	Modify the parameters and corresponding instructions according to the new UI.
		9.3	Correct the conversion of processing units for G20 and G21. Correct the description for G89. Add Section 9.3.3 Definitions of SNC variables. Add Section 9.3.4 Macro syntax.
		9.4	Correct description of the MPG example in Section 9.4.6. Correct the example of the SNC calling external macro in Section 9.4.7. Add G-code simulation example in Section 9.4.8.
		9.5	Add error code description. Add Section 9.5.8 SNC activation error code.

Release date	Version	Chapter	Revision contents
		Appendix A	Add the applicable bus types. Add description for the corresponding registers of EtherCAT. Add corresponding registers of the DMCNET motion card local I/O. Add system identification code. Add description for corresponding registers of the Motion table. Correct description for the corresponding registers of SNC.
		Appendix B	Correct the list of homing methods.

For more information about the IPC Motion Platform User Guide, refer to:

(1) ASDA-A2 Series User Manual

(2) ASDA-B2 Series User Manual

(3) DMCNET Remote Module User Manual

(4) DMCNET Gateway Module User Manual