Smarter. Greener. Together.

Industrial Automation Headquarters

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

Asia

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

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

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

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

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

Delta Electronics (India) Pvt. Ltd.
Plot No 43 Sector 35, HSIIDC
Gurgaon, PIN 122001, Haryana, India
TEL : 91-124-4874900 / FAX : 91-124-4874945

Americas

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

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

Europe

Delta Electronics (Netherlands) B.V.
Eindhoven Office
De Witbogt 20, 5652 AG Eindhoven, The Netherlands
TEL : +31 (0)40-8003800 / FAX : +31 (0)40-8003898

IPC Motion Platform User Guide

IPC Motion Platform User Guide

www.deltaww.com

Smarter. Greener. Together.

Preface

Thank you for choosing this product. This manual provides information related to IPC Motion Platform (hereafter referred to as IMP).

This manual includes:

- The installation and validation of IMP software
- The operating instructions of the settings and IMP Quick Start
- The instructions of IMP Ladder Editor
- Descriptions of PLC instructions used by IMP
- Descriptions of the Motion Program Macro (MPM) used by IMP
- Parameters descriptions
- Troubleshooting

Product features:

IMP (IPC Motion Platform) software design is the integration of PLC syntax, graphical HMI, and IPC floating-point operation, expanding the servo motion control on the basis of high-performance operation units. Through DELTA DMCNET communication fieldbus, IMP is capable of controlling up to 36 axes of servo units, and conducting multi-axis synergic motion. With the introducing of Soft Numerical Control (SNC) motion control functions, the difficulty of developing NC series product is significantly reduced. In accordance to the actual industry applications, user can develop functions and programs related to engineering process. Additionally, by connecting visual system, sensor, and PC-based central control system through Ethernet and serial communication interface, IMP integrated a comprehensive industry control network.

How to use this manual:

This manual can be used as reference while applying IMP, which contains the information related to the product installation, setting, validation, as well as instructions of how to use this product.

DELTA technical services:

Please consult the distributors or DELTA customer service center if any problem occurs.

Safety Precautions

IMP software and industrial PC are integrated as the motion controller; therefore, for the industrial PC host and other applications, the relative hardware devices should be in compliance with the specifications of each product.

Pay special attention to the following safety precautions anytime during inspection, installation, wiring, operation and examination.

The symbols of "DANGER", "WARNING" and "STOP" represent:

DANGER	**It indicates the potential hazards. It is possible to cause severe injury or fatal harm if not follow the instructions.**
WARNING	**It indicates the potential hazards. It is possible to cause minor injury or lead to serious damage of the product or even malfunction if not follow the instructions.**
STOP	**It indicates the absolute prohibited activity. It is possible to damage the product or cannot be used due to malfunction if not follow the instructions.**

Operation

WARNING

- Before operation, please adjust the parameter setting value according to the user needs. If it is not adjusted to the correct setting value, it is possible to lead to malfunction of the machine or the operation might become out of control.
- Before the machine starts to operate, please ensure that the emergency stop can be activated anytime.
- When power on, please make sure the motor shaft stands still and will not operate because of mechanical inertia or other causes.

STOP

During the operation, it is prohibited to touch any rotating motor parts. Or it is possible to cause personnel injury.

DANGER

- In order to prevent any accident, please separate the couplings and belts of the machine and operate the motor without load for its initial trial run.
- If users fail to operate the machine properly after the servo motor connects to the equipment, it would cause the damage of the equipment and lead to personnel injury.

Note: If there is any difference of each version, please download the latest version at DELTA's website (http://www.delta.com.tw/industrialautomation/).

Table of Contents

Before Operation

1 Overview

1.1 Introduction to IPC Motion Platform 1-2
1.2 Structure of IPC Motion Platform 1-3

2 Installation

2.1 System requirements 2-2
2.2 Installation 2-4
2.3 License Authorization 2-8

Operation and Settings

3 IMP Quick Start

3.1 Servo drive settings 3-2
3.2 Start IMP Quick Start 3-3
3.2.1 Setting 3-4
3.2.2 Mechanical setting 3-5
3.2.3 Parameter Import / Export 3-12
3.2.4 Auto start IMP after PC starts 3-12
3.3 Trials and operation 3-13
3.3.1 Single-axis control 3-13
3.3.2 Multi-axes interpolation 3-16
3.3.3 Digital input / output 3-17
3.3.4 Analog Input / Output 3-19
3.3.5 Software numerical control (Option) 3-20

4 Ladder Editor

4.1 Introduction to Ladder Editor 4-2
4.1.1 How to start Ladder Editor 4-2
4.1.2 Program upload and download 4-5

4.2 Create new ladder program and settings 4-10
4.3 Other functions 4-14

5 Memory Device

5.1 Device table 5-2
5.1.1 DMCNET input relay(DX) / DMCNET output relay (DY) 5-3
5.1.2 Auxiliary relay (M) 5-4
5.1.3 Timer (T) 5-4
5.1.4 Counter (C) 5-5
5.1.5 Data register (D) 5-6
5.1.6 Indirect register (V) 5-6
5.1.7 HMI auxiliary register 5-6
5.1.8 Constant (K) / Float point (F) 5-7
5.2 System special relay 5-8
5.2.1 PLC special relay 5-8
5.2.2 Motion status special relay 5-9

How to use programming language?

6 Logic Editing

6.1 PLC Instructions 6-2
6.1.1 Instruction list 6-2
6.1.2 Basic Instruction 6-7
6.1.3 Application Instruction 6-19
6.1.4 Single-axis motion instruction 6-54
6.1.5 Interpolation Motion Instruction 6-74
6.1.6 Motion Program Macro (MPM) Control Instruction 6-92
6.1.7 Example 6-96

7 Motion Program Macro (MPM)

7.1 Instruction list and overview 7-2
7.1.1 Application instruction 7-3
7.1.2 Motion application instructions 7-15
7.2 Motion Program Macro (MPM) Editor 7-37

Parameters and Settings

8 Communication

8.1 Modbus Communication Setting ········ 8-2
8.1.1 Ethernet Communication Setting ········ 8-3
8.1.2 Serial Communication Setting ········ 8-5
8.1.3 Setting of Communication Instruction ········ 8-7
8.1.4 Communication Error Code ········ 8-9
8.2 IPC Motion Platform Memory Address ········ 8-10

9 Soft Numerical Control, SNC (Optional)

9.1 SNC Framework ········ 9-2
9.2 Parameter Descriptions ········ 9-3
9.2.1 SNC Special Parameters ········ 9-3
9.2.2 Motion Parameters ········ 9-5
9.2.3 Speed Control ········ 9-7
9.2.4 Hardware Information ········ 9-9
9.2.5 Tool Setting Parameters ········ 9-10
9.2.6 Backlash Compensation ········ 9-11
9.2.7 Reverse Backlash Compensation ········ 9-11
9.3 SNC Interpreter ········ 9-12
9.3.1 G Code Supporting Table ········ 9-12
9.3.2 M Code and T Code ········ 9-14
9.4 Descriptions of SNC Functions ········ 9-15
9.4.1 Accessing G Code ········ 9-15
9.4.2 External Macro ········ 9-16
9.4.3 Automatic Tool Setting ········ 9-17
9.4.4 Single Step Mode ········ 9-18
9.4.5 Spindle Control ········ 9-18
9.4.6 Manual Feed Rate Adjustment ········ 9-18
9.4.7 MPG Simulation Mode ········ 9-19
9.5 Description of SNC Error Codes ········ 9-19

Appendix

A List of Special Register

Troubleshooting ······ A-2
List of special register (W, R) in IMP system ······ A-3
Special register (W, R) of single-axis motion ······ A-10
Special Register (W, R) of Servo Group ······ A-13
Special Register (W, R) of Motion Program Macro (MPM) ······ A-15
Special Register (W, R) of SNC ······ A-15

B Homing Mode

List of homing methods ······ B-2
Description of homing methods ······ B-4

1 Overview

Before applying IPC Motion Platform PLC, you may read this chapter to understand its operation structure.

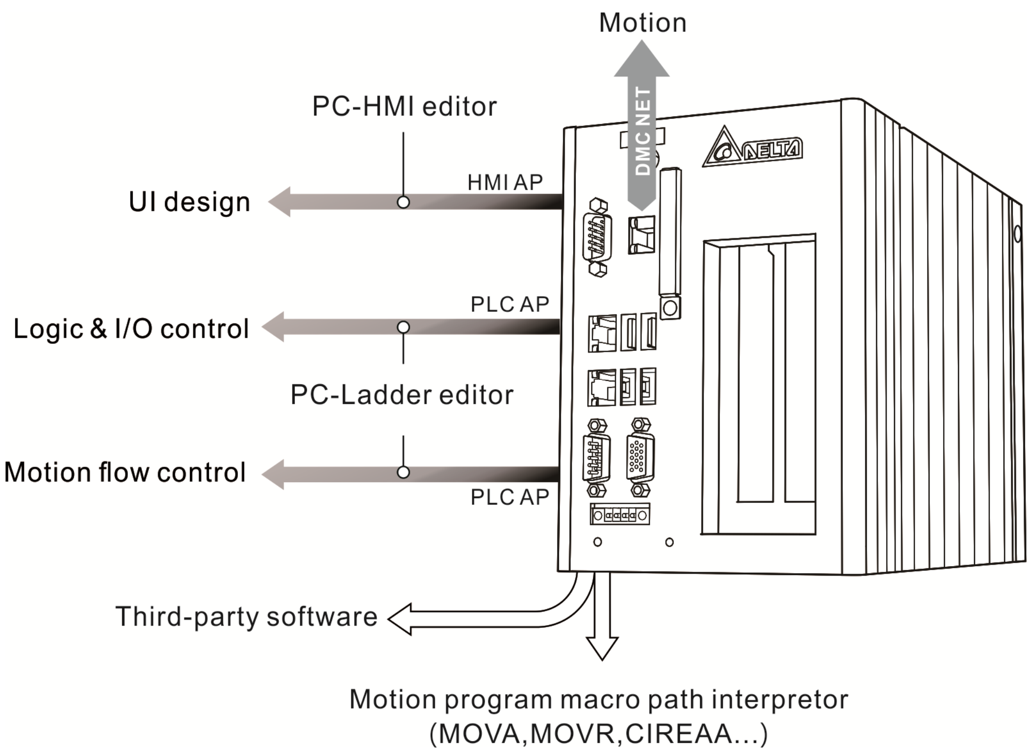
1.1 Introduction to IPC Motion Platform ········· 1-2
1.2 Structure of IPC Motion Platform ········· 1-3

1

1.1 Introduction to IPC motion platform

IMP (IPC Motion Platform) is an application platform that is established in the framework of fieldbus for high-speed motion control. With the integration of HMI editing software and PLC program editor, designing operation screen, logic control and motion control can easily be done via DOPSoft 3.0.

The kernel of IMP software can be installed in Delta PAC and IPC that supports Delta motion cards. This kernel will transform your PAC or IPC into a high-speed communication type motion controller. Unlike traditional controllers, which issue commands by means of sending pulses and analog signals to control the servo drive, IMP applies DMCNET communication fieldbus to serial connect servo drives and remote modules with 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 Structure of IPC motion platform

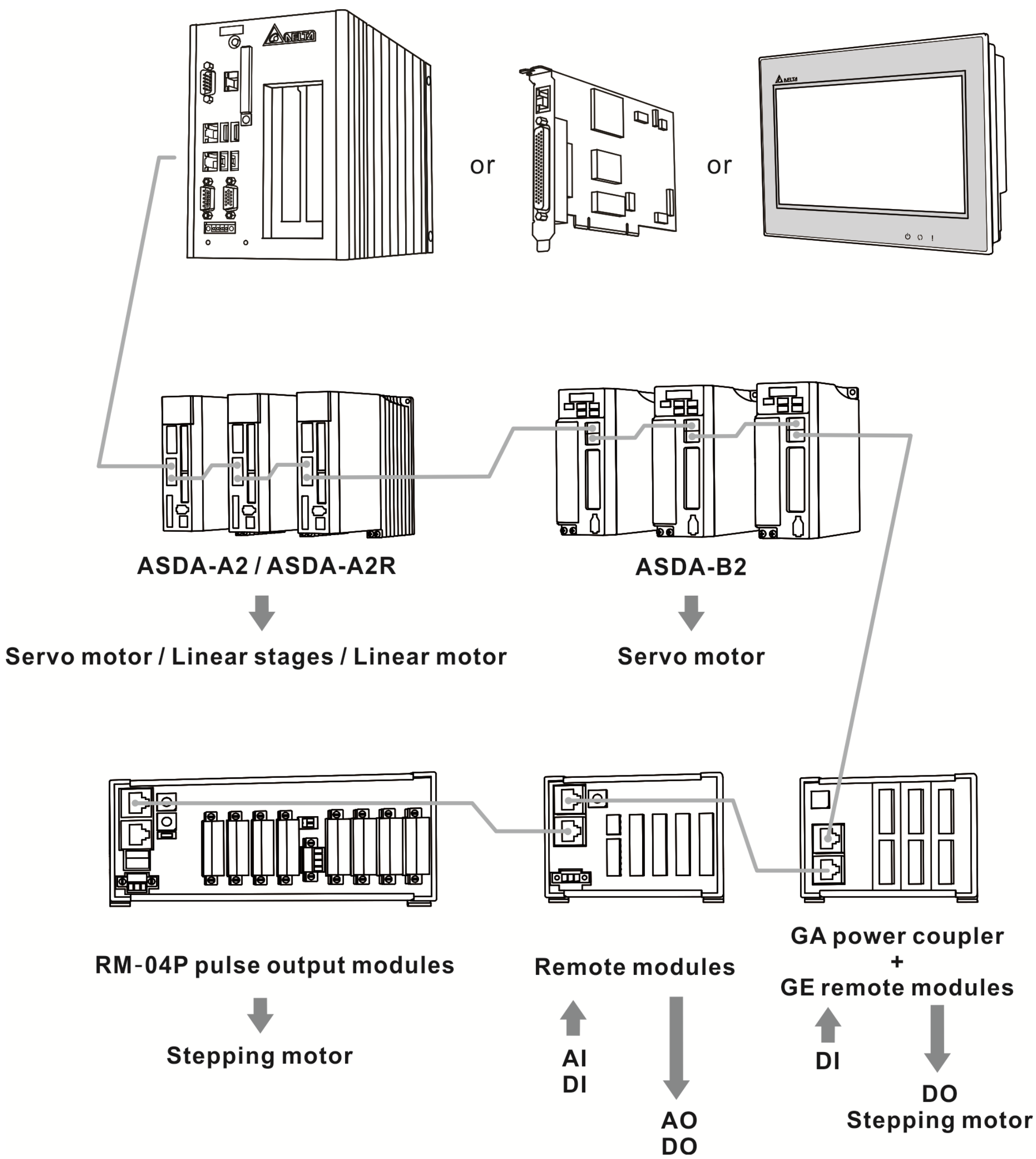
1

IMP is a PC-based program and is able to multitask. It can execute programs of PLC and HMI respectively, integrating the operation interface and logic control, and providing optional Soft Numerocal Control (SNC) functions.

The program types in IMP controller include Cyclic Task, Subroutine, and Motion Program Macro (MPM). With DMCNET real-time fieldbus, IMP has the built-in 35 homing modes and can exercise 36-axis motion control*, which supports speed, torque, position, 3-axis helical and linear interpolation commands, increamental type, absolute type command, and S- and T-curve functions.

By applying the built-in PLC software version and PC computing to assist DMCNET motion cards, IMP realizes the complex algorithm, including G-code interpretor, look ahead function, and reverse along original path. And users can set M code, T code and the operation logic for the machine.

Note: IMP supports max. 3 motion control cards with up to 12 nodes/axes for each.



(This page is intentionally left blank.)

Installation

2

You will find the steps to install IPC Motion Platform kernel and the method to acquire the license authorization in this chapter.

2.1 System requirements 2-2
2.2 Installation 2-4
2.3 License Authorization 2-8

2

2.1 System requirements

In order to have the best user experience, using MH1 and MP1 series PAC manufactured by Delta is suggested. This system is required to work with Delta DMCNET motion control cards. For other hardware information, please refer to the following requirement.

Requirements	
Operation system	Windows 7 Windows 7 Embedded Windows XP [*1] Windows XP Embedded[*1]
CPU	Dual core CPU 1.2Ghz [*2]
Memory	2G
Hard disk drive	Hard disk drive[*3]of 1GB
Display	Resolution of 1024 x 600
Human-system interface	Mouse or touch screen
System environment	.NET Framework 3.5 SP1
Others	It is required to work with Delta PAC products and motion control cards such as PCI-DMC-A02, PCI-DMC-B01, PCI-DMC-B02, MH1-C70D, MH1-C50D, MH1-A12D, MH1-S30D, and MP1-A10D.

Note:

*1: For Windows XP, Windows XP Embedded operation system, please install .NET Framework 3.5 SP1. (You may download it at: https://www.microsoft.com/en-US/download/details.aspx?id=22)

*2: As the operation of computing kernel of SNC requires CPU, please select a multi-core PC or IPC if using the SNC function. To use two sets of SNC function, you must select models of 4-core above with 2 DMCNET channels. To use thee sets of SNC, it requires models of 5-core above with 3 DMCNET channels and so on.

*3: The PC or IPC for installing IMP must have (D) Hard disk drive.

2

To operate IMP kernel, it requires DMCNET equipment shown as follows:

		Model					
		MH1- C70D	MH1-C50D	MH1-A12D	MP1-A10D	PCI-DMC-B01	PCI-DMC-A02
System	CPU	Intel Core i7-3612 QE Quad Core 2.1GHz	Intel Core i5-3610 ME Dual Core 2.7GHz	Intel Atom E3845 Quad Core 1.91GHz	Intel Atom E3825 Dual Core 1.33GHz	-	-
Operation axes		12	12	12	6	12	12
Max. Slaves		12	12	12	12	12	12
On board I/O		1/1	1/1	1/1	8/4	1/1	32/24
Compare		2	2	-	-	2	-
SNC		△	△	△	△	△	△
Remote type modules	DI: RM32MN/ M64MN	△	△	△	△	△	△
	DO: RM32MT/ RM64MT	△	△	△	△	△	△
	DO/DI: RM32PT/ RIO3232RT5	△	△	△	△	△	△
	Pulse type: RM04PI*1	△	△	△	△	△	△
	AD/DA: RM04AD/ RM04DA	△	△	△	△	△	△
Gateway type modules	Pulse type: GE01PH	△	△	△	△	△	△
	DI: GE16MN	-	-	-	-	-	-
	DO: GE16MT	-	-	-	-	-	-
Servo systems	ASD-A2-****-F	△	△	△	△	△	△
	ASD-B2-****-F	△	△	△	△	△	△
	ASD-M-****-F	△	△	△	△	△	△

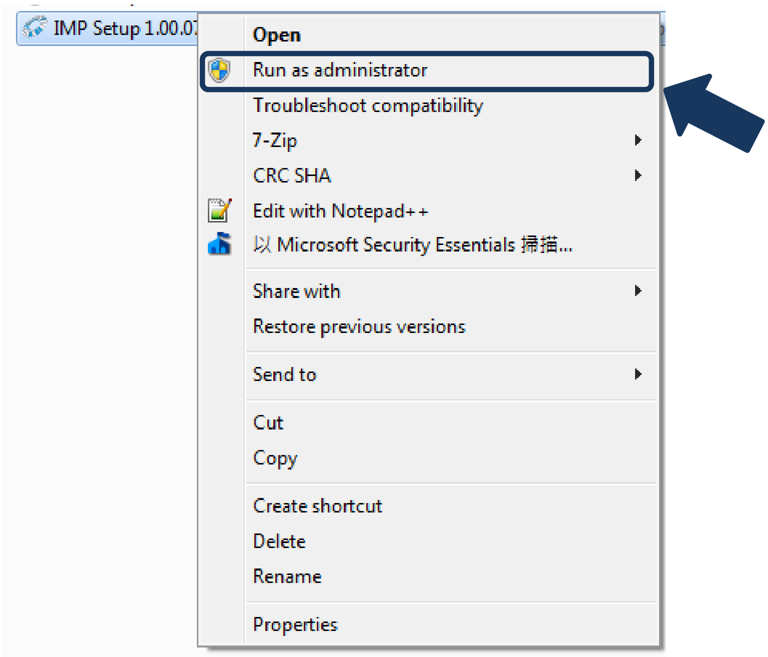
2

2.2 Installation

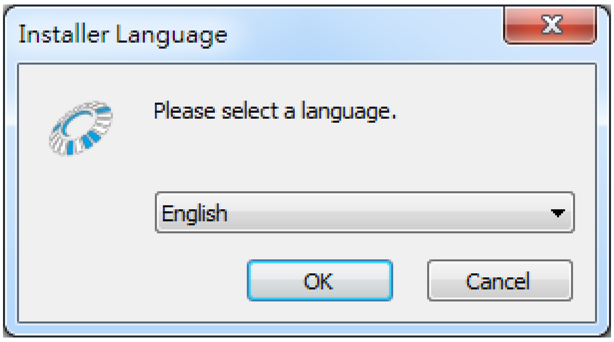
Upon obtaining the installation file of IMP kernel, please follow the steps to install the software.

1. For Windows XP, Windows XP Embedded operation system, please install “.net framework 3.5 SP1”.

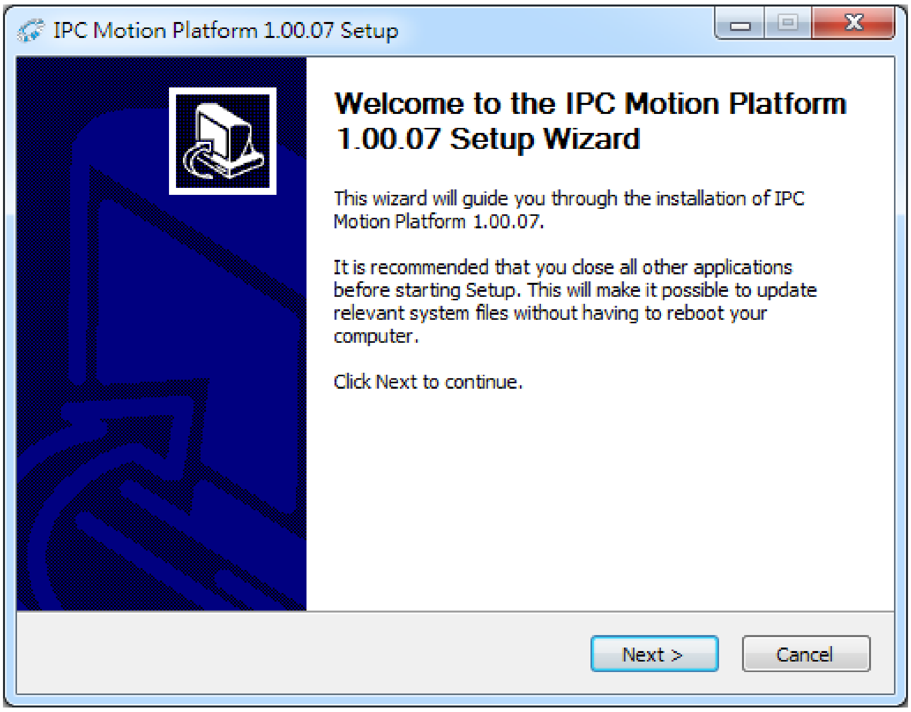
2. Launch “IMP Setup.exe” file as Administrator.
   a. For Windows XP, double click on “IMP Setup.exe”.
   b. For Windows 7 or higher, right click on “IMP Setup.exe” and click on **Run as Administrator.**



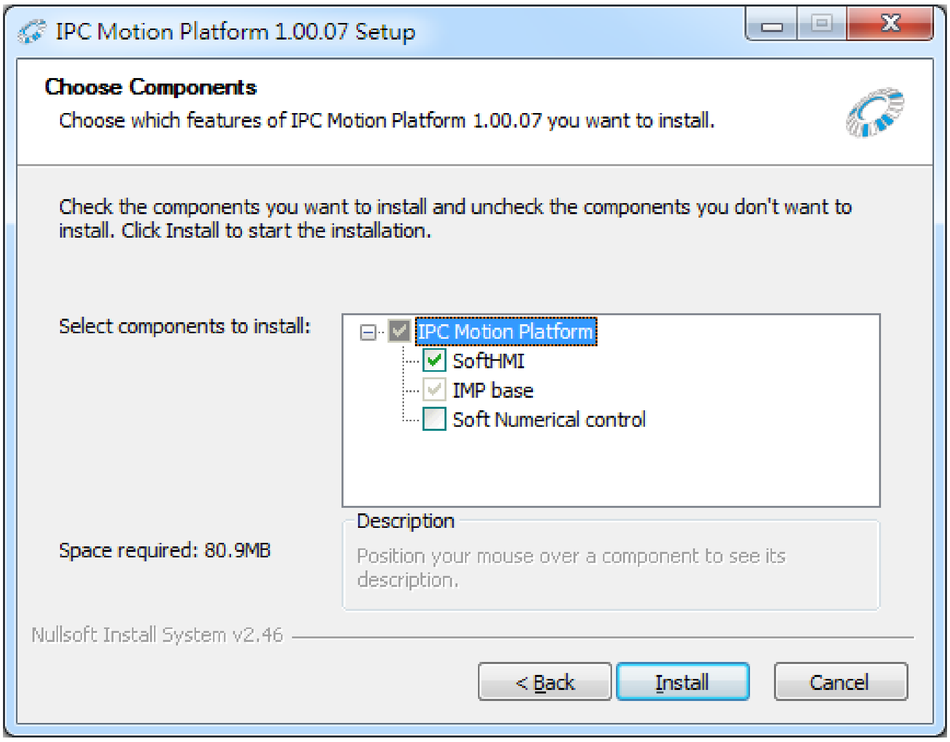
3. Select the installer language and click **OK**.



4. You will be directed to the welcome page. Now, click on **Next**.



5. Select the components to be installed and click on **Install** to start the installation. (The kernel of IPC Motion Platform will be installed in the fixed directory [D:\NandFlash\IPC Motion Platform])

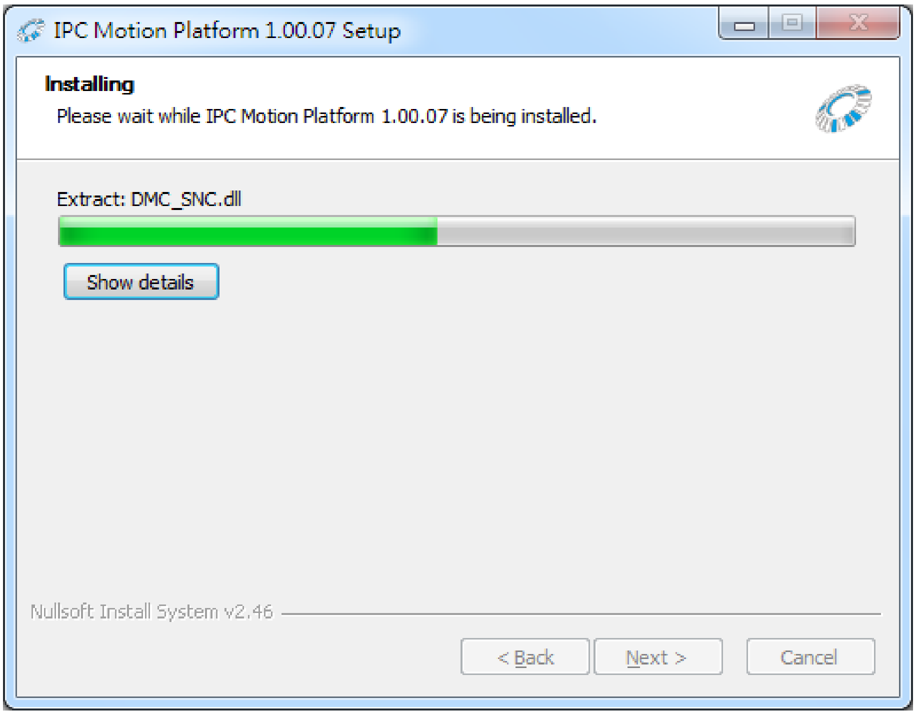


SoftHMI checkbox: If this item is selected, HMI and PLC interpreter will be installed. The interpreter will execute the interface screen and PLC program edited by DOPSOFT 3.00.01.

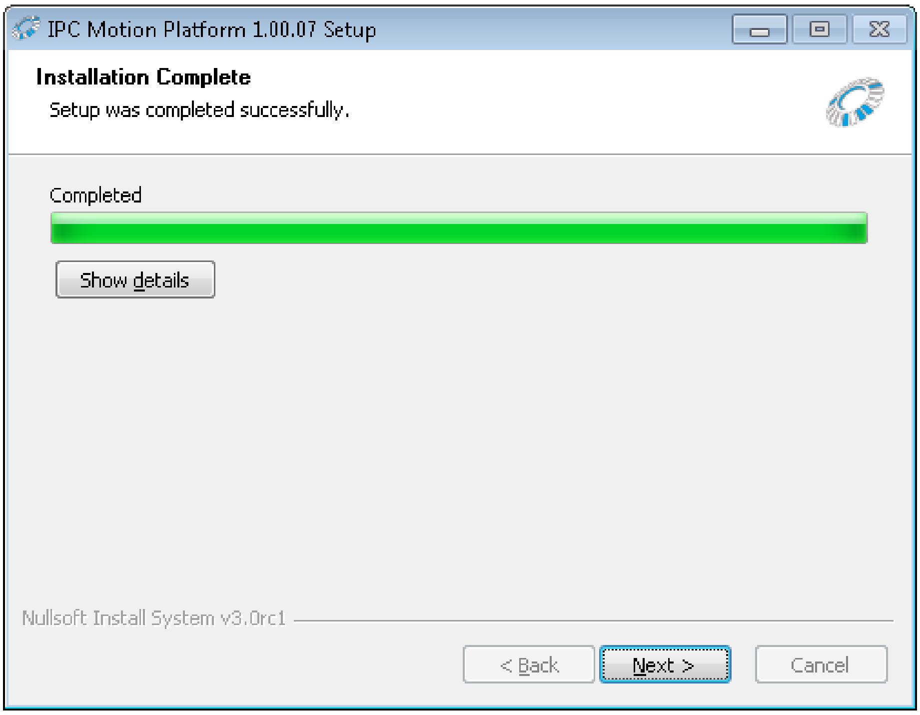
Soft Numerical control checkbox: If this item is checked, G code interpreter for value control will be installed. Please note that this will increase the computing time for IMP, check this item only when necessary.

2

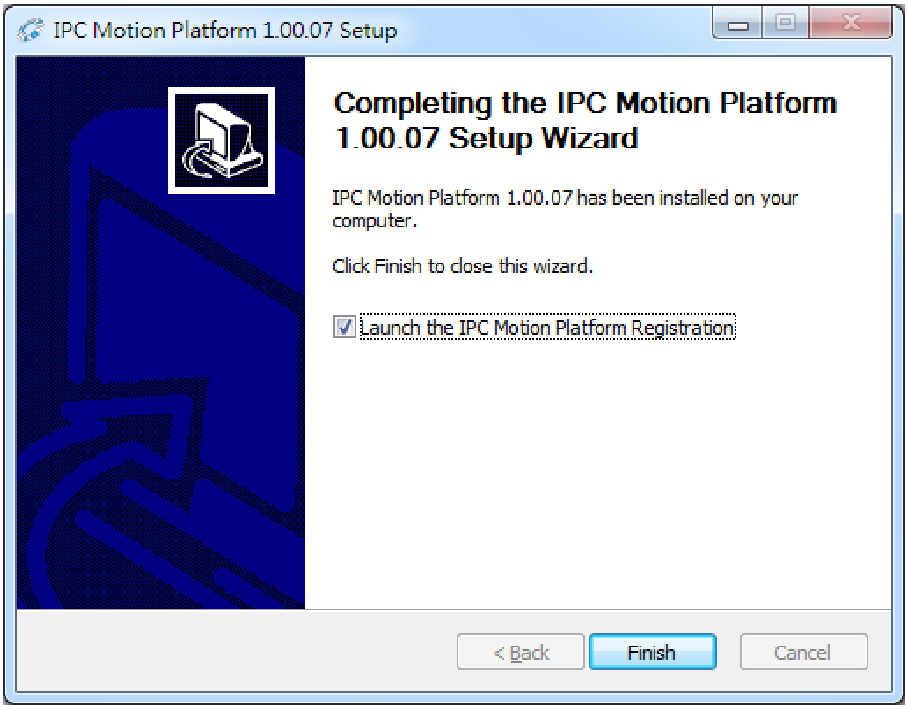
6. Then, the waiting screen for installing IPC Motion Platform will be shown.



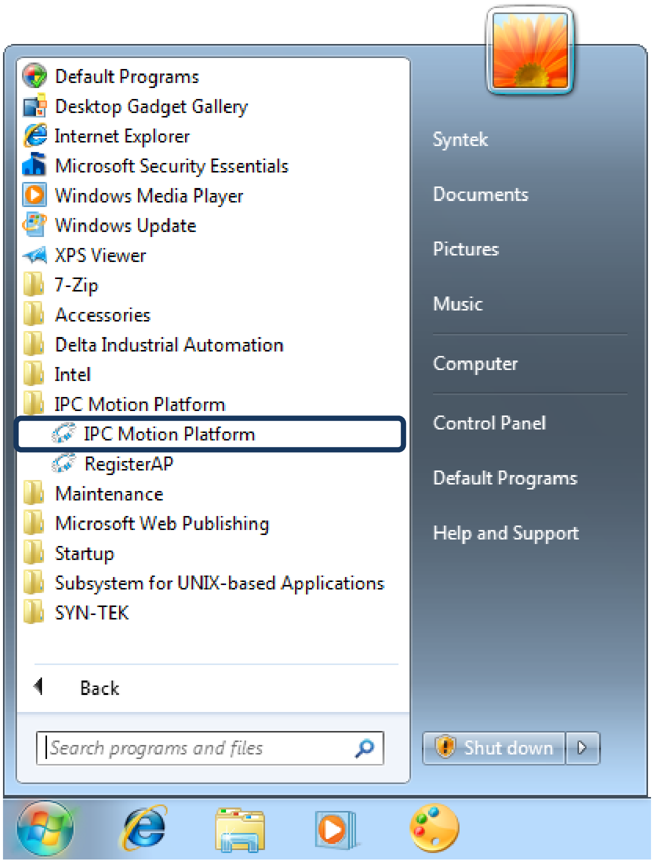
7. Installation completed. Click on **Next**.



8. Click on **Finish** to complete the installation.
   If **Launch the IPC Motion Platform Registration** is checked, the license authorization software RegisterAP will be launched once the installation is completed.



After the IMP kernel is installed, you may find it and its relevant files in the fixed directory [D:\NandFlash\IPC Motion Platform]. And you will find the shortcut for the executable file by going to [Start] > [All programs] > [IPC Motion Platform]. See the figure below.



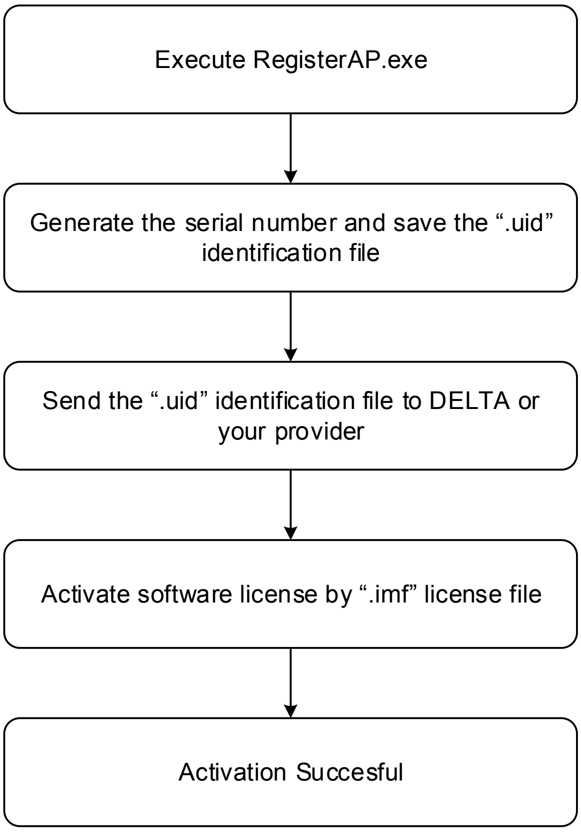
2

2.3 License Authorization

IMP professional has to be run by DMCNET motion control card and in DELTA PAC hardware environment; therefore, you need to get license authorization for each PAC. The procedure is shown in the following flow chart.

Note: Once the authorization is completed, DO NOT change the hardware environment. Action such as using additional DMCNET motion cards may result in authorization failure and program cannot be executed.

Flow chart of license authorization:

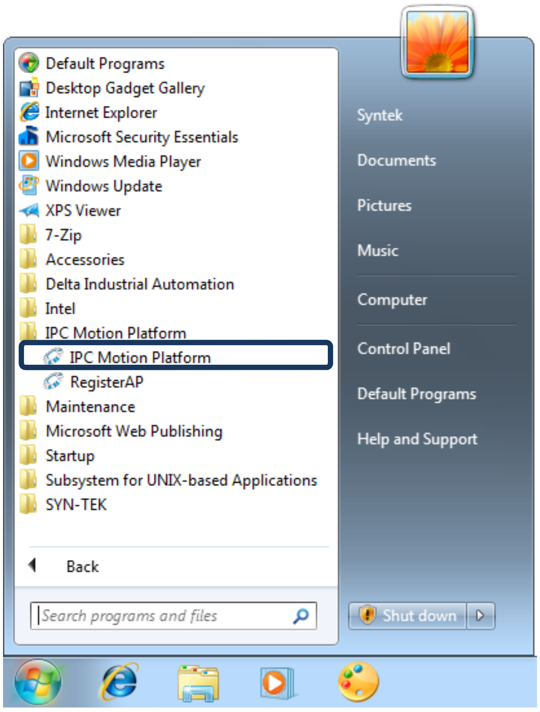


Please follow the instructions to get your own license:

1. Launch the RegisterAP application either by Method A or Method B.

   For Windows XP, double click on "RegisterAP.exe" file;
   For Windows 7 or higher, right click on "RegisterAP.exe" file and click on **Run as Administrator.**

Method A	Go to [D:\NandFlash\IPC Motion Platform\IMP base\RegisterAP]. Then, launch RegisterAP.exe as administrator.
Method B	Go to [Start] > [All programs] > [IPC Motion Platform] > [RegisterAP]. Then, launch RegisterAP.exe

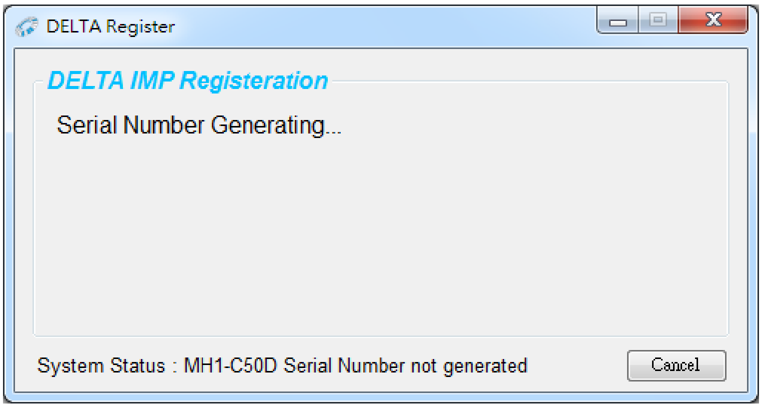


2. Once the software is started, the system status will be shown on the bottom of the window. The **Activate License** button will be disabled if the serial number is not generated.

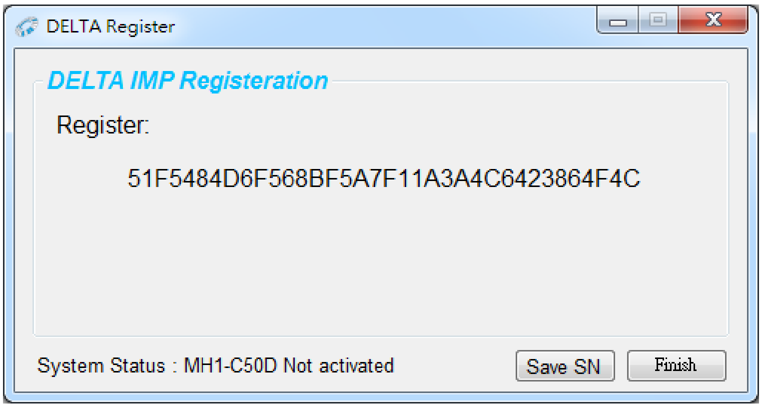
2

3. Press **Generate Serial Number**.

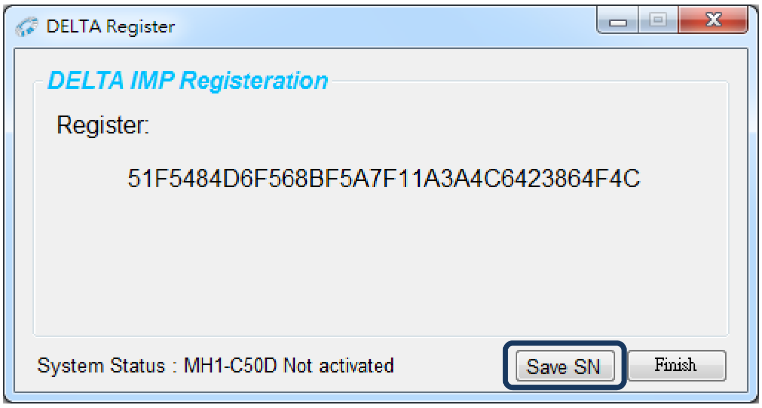
4. Wait for serial number generation.



5. When you get the register number successfully, the hardware and activation status will be shown in system status bar.

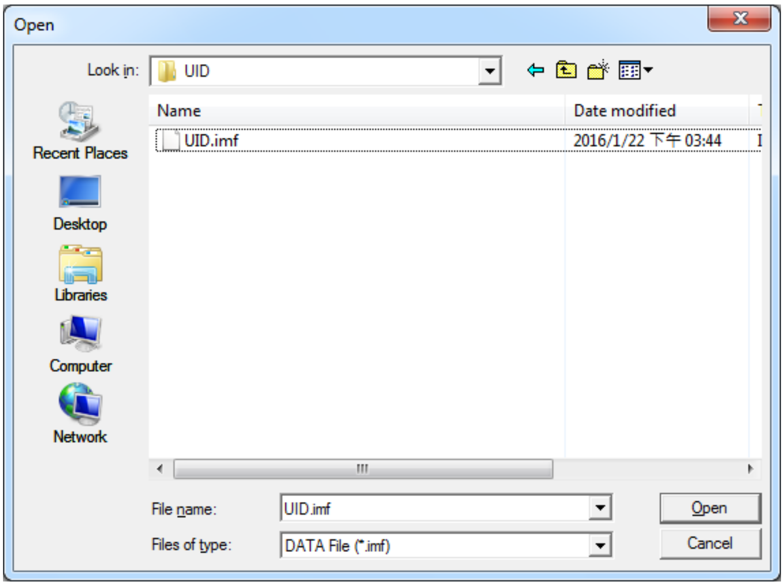


6. Save the register number and sent it to DELTA.
   Click on **Save SN** and you can save this number to the selected file. (Multiple serial numbers can be saved in one file.) Or, you can directly copy the register number and send it. Delta will verify the number and send a UID.imf license file to you. This file can be applied to more than one pieces of hardware. If verification is failed, Delta will be sending you an email and ask you to provide the correct register number.



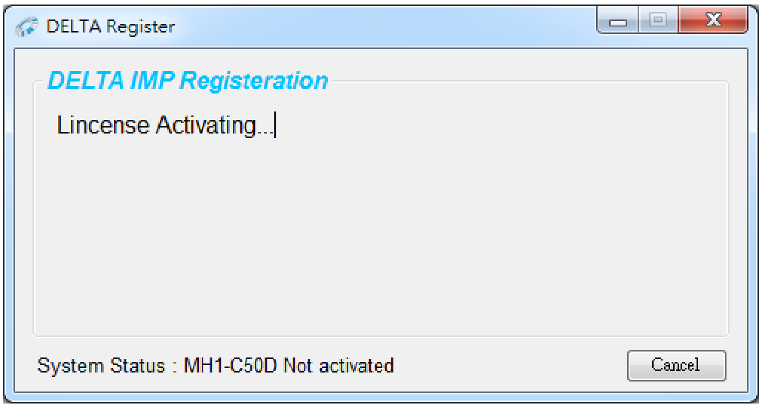
7. Once you get the ".imf" license file, you can start to activate the software by pressing **Activate License**.

8. Then, a window will pop out to ask you to select the "imf" license file sent from Delta.

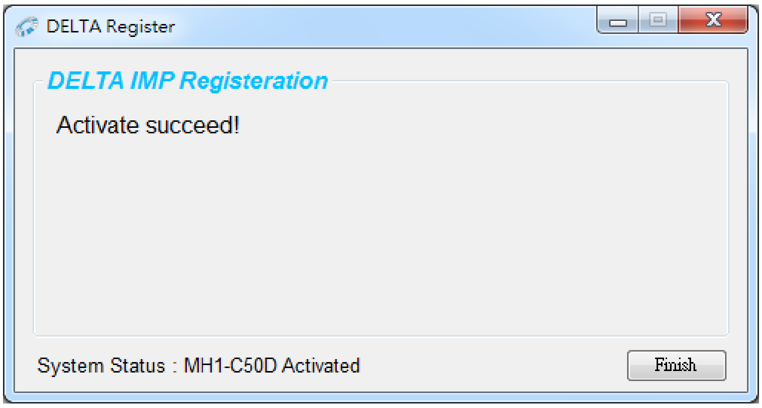


2

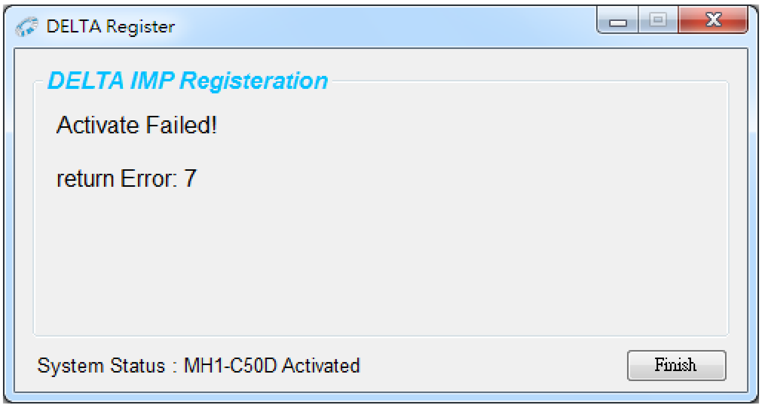
9. Wait for license activation.



10. If activation is successful, the system status will show “Activated”.



If activation is failed, the screen will show an error code. Please inform DELTA of this error code.



IMP Quick Start

3

Before using IMP system to do motion controls, you may set the mechanism parameters and conduct trials for machine's motion control via Quick Start setting interface.

3.1 Servo drive settings 3-2
3.2 Start IMP Quick Start 3-3
3.2.1 Setting 3-4
3.2.2 Mechanical setting 3-5
3.2.3 Parameter Import / Export 3-12
3.2.4 Auto start IMP after PC starts 3-12
3.3 Trials and operation 3-13
3.3.1 Single-axis control 3-13
3.3.2 Multi-axes interpolation 3-16
3.3.3 Digital input / output 3-17
3.3.4 Analog Input / Output 3-19
3.3.5 Software numerical control (Option) 3-20

3.1 Servo drive settings

IMP software will check if any DMCNET motion card is installed in the PC and then check if the card is connected to the servo drive and remote I/O module. If no remote I/O module is detected, HMI and PLC in IMP will be disabled. Please follow the steps below to the setup.

- **Select the servo drive**

ASD-A2-****-F, ASD-B2-****-F, or ASD-M-****-F series servo drives manufactured by Delta

- **Set the station (node) number (P3-00)**

In IMP, DMCNET field bus is used to connect servo drives and remote modules. There must be station No.1 and each station/node's setting value should be within 1 ~ C.

If PAC such as model MP1-A10D or MH1-S30D is applied, the servo station number should be set to 1~ 6.

- **Set control mode (P1-01)**

Set the control mode to DMCNET by setting P1-01 to 0x00b.

- **Set the transmission speed (P3-01)**

Set servo parameter P3-01 to 3203.

- **Make sure no alarm occurs**

Make sure no alarm occurs. If Alarm Reset is required, you may use ALRM command or refer to the troubleshooting chapter of the servo drive user manual to clear the alarm.

- **Switch to "Servo On"**

To execute servo commands, the servo drive has to be in Servo On state. If not, you may execute SVON command to switch to Servo On.

- **Ensure the Emergency stop is released**

1. Make sure no stop command is in execution.
2. Make sure the emergency stop is released. Please refer to servo drive user manuals for detailed description.
3. If the servo drive is still in emergency stop state and cannot be cleared, check the DI setting, which detail can be found in servo drive manuals.

- **Others**

Make sure the servo drive is not performing MPG functions. Please ensure MPG is disabled. Set register W19000 to 3.

3.2 Start IMP Quick Start

With this interface, users may exercise servo motion controls (functions such as liner, arc, and helical interpolation), DMCNET remote module controls, including digital, analog, pulse input/output modules and set parameters to connect the motor and external modules.

After IMP is started, you will see an IMP icon at the right-down corner (See Figure 3.2.1). Right-click on it and you will see the following options:

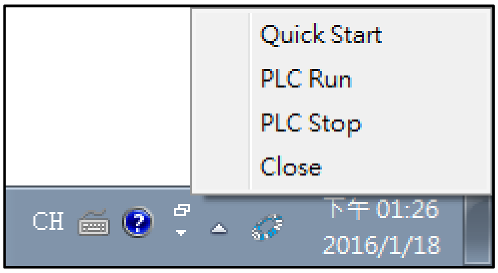


Figure 3.2.1 IMP control panel

Item	Description
IMP Quick Start	Start the IMP Quick Start and stop HMI and PLC programs.
PLC Run	Run PLC and HMI programs.
PLC Stop	Stop PLC and HMI programs.
Close	Close the IMP system, including IMP, PLC and HMI programs.

3

After started, the main screen of IMP Quick Start will pop out (See the figure below). Its items are presented based their functions.

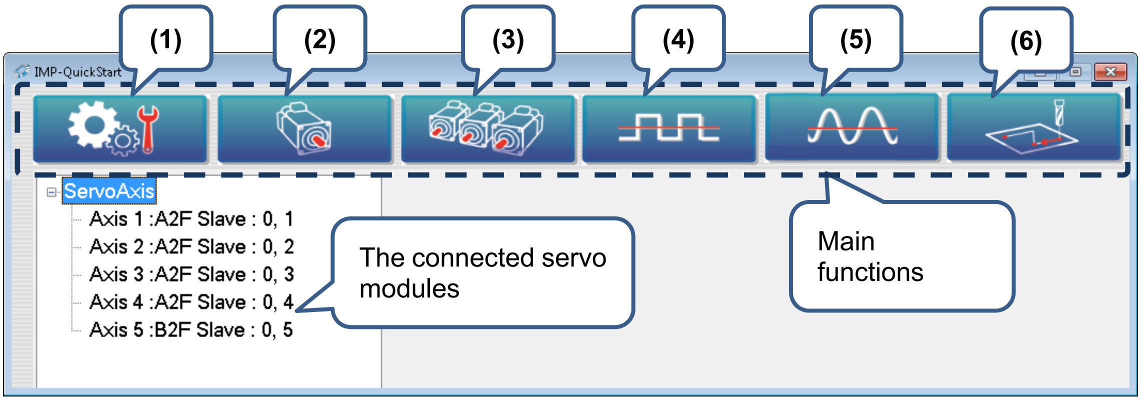


Figure 3.2.2 Main screen of Quick Start

(1) Setting: For setting parameter values of each module.

(2) Single-Axis: For single-axis motion and monitoring each axis.

(3) Multi-Axis: For interpolation and monitoring each servo axis.

(4) DIO: For monitoring the digital input/output points and force output status.

(5) AD/DA: For monitoring values of analog output/input and force output status.

(6) Soft Numerical Control (This function requires installing SNC interpreter): For SNC motion test and monitoring.

3.2.1 Setting

The function of **Setting** is described as follows:

- Parameter Setting (Modules that connected to each node via DMCNET and parameter setting)
- Communication Setting (Refer to chapter 8)
- MPM Edit
- Parameter Export/Import
- Cancel IMP Auto Run
- HMI Border
- Language

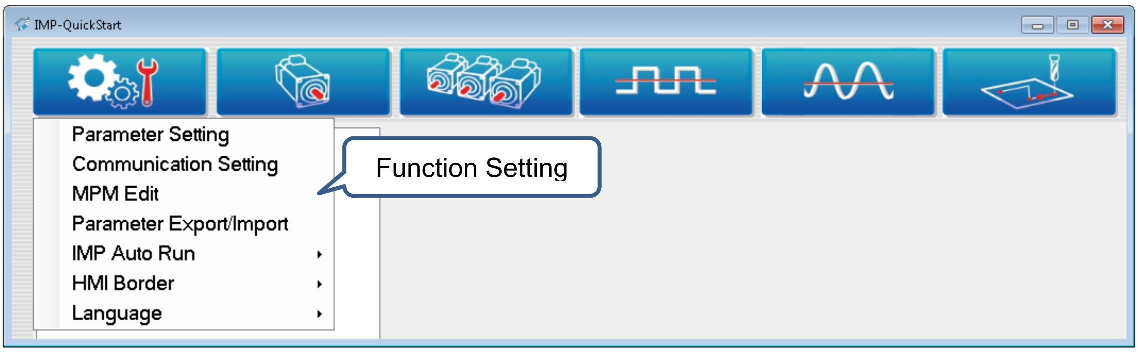


Figure 3.2.2.1 Function list

3.2.2 Parameter setting

You can start viewing and editing the mechanical parameters of IMP by clicking on **Parameter Setting** in IMP Quick Start. See the figure below.

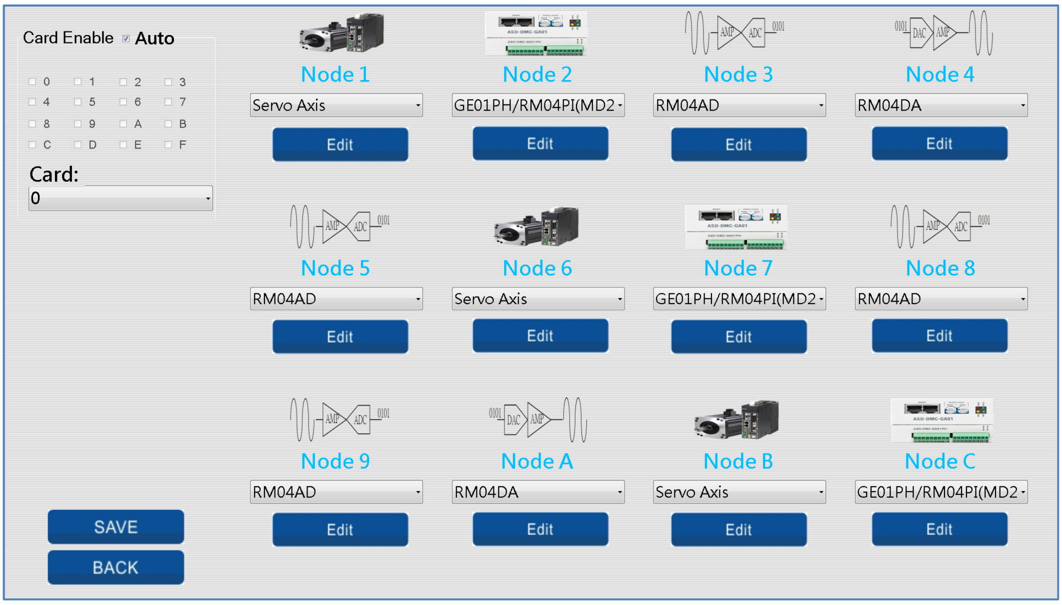


Figure 3.2.2.1 Parameter setting screen

■ Operation

There are two modes for card selection, **Auto** and **Manual** in the setting page.

If it is in Auto mode, you can view and edit all 16 motion cards (0 ~ F) as shown in the figure below.

(1) Check the **Auto** checkbox.

(2) Select the card you are going to edit from the combo box.

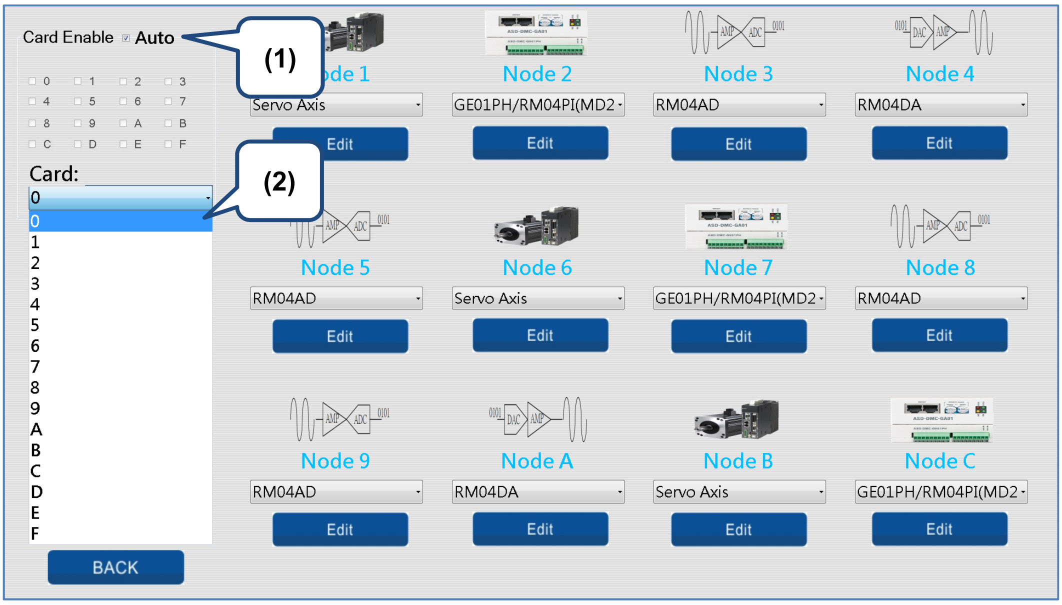


Figure 3.2.2.2 Card selection in Auto mode

If it is in **Manual** mode, only the selected cards will show in the combo box for viewing and editing. See the figure below.

(1) Uncheck the **Auto** checkbox**.**

(2) Select the cards (Max. 3). The combo box will show the card numbers you selected for viewing and editing.

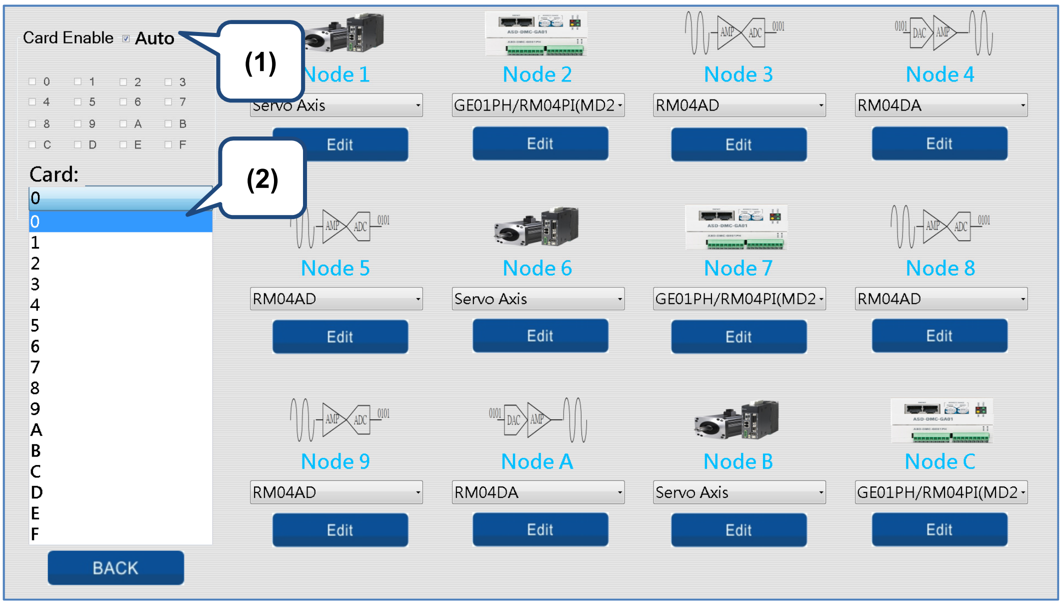


Figure 3.2.2.3 Card selection in Manual mode

This setting screen allows you to set mechanical parameters for each module. It is categorized into four module types by different applications:

1. Servo Axis
2. Ge01PH / Rm04PiM2
3. Rm04AD
4. Rm04DA

Please follow the steps shown below to set module type parameters for each node.

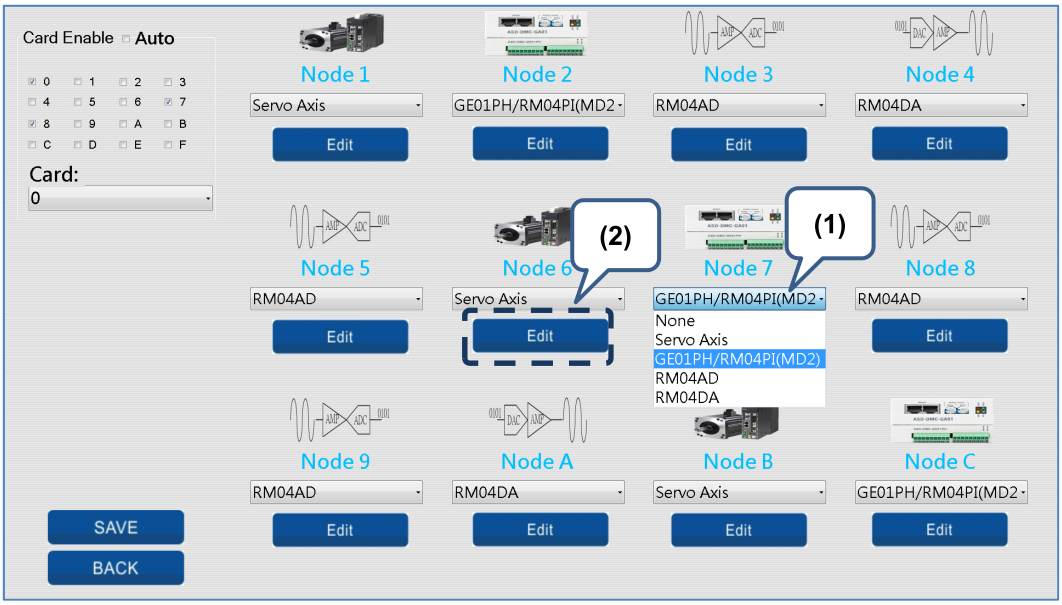


Figure 3.2.2.4 Setting screen of module type for each node

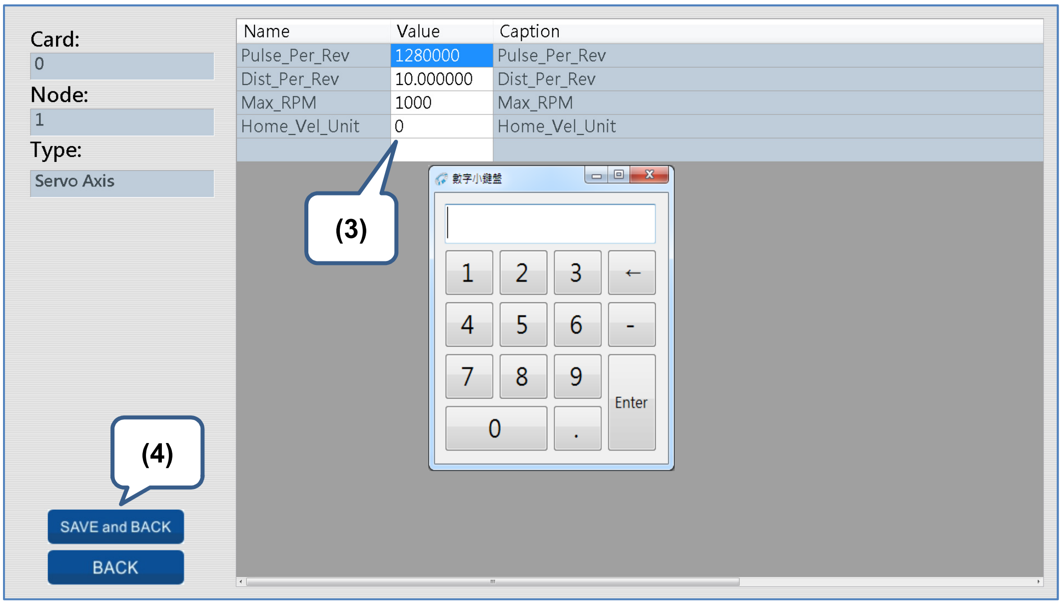


Figure 3.2.2.5 Parameter setting (for modules) screen

(1) Select the node you are going to set up and choose the module type from the combo box.

(2) Click on the node button to enter the parameter setting page.

(3) After entering the setting page, you can click on the value directly to edit it or input the value by the screen keyboard. See Figure 3.2.2.5. To edit the parameter, click on the numbers and press **Enter**. (It is the figure for demonstration; the actual content is subject to change according to the selected module types. Details will be illustrated in the following sections.)

(4) Click on **SAVE and BACK** to save the parameter values and exit the setting page. Or, click on **Exit** to discard the setting and exit.

(5) After finishing setting the motion cards and module nodes, click on **SAVE ALL** to save all parameter values. Then, click **Exit** to leave the page.

3

■ Parameter setting of servo axis

Servo axis includes ball screws, the rotary table, and racks. See the figure below for parameter settings:

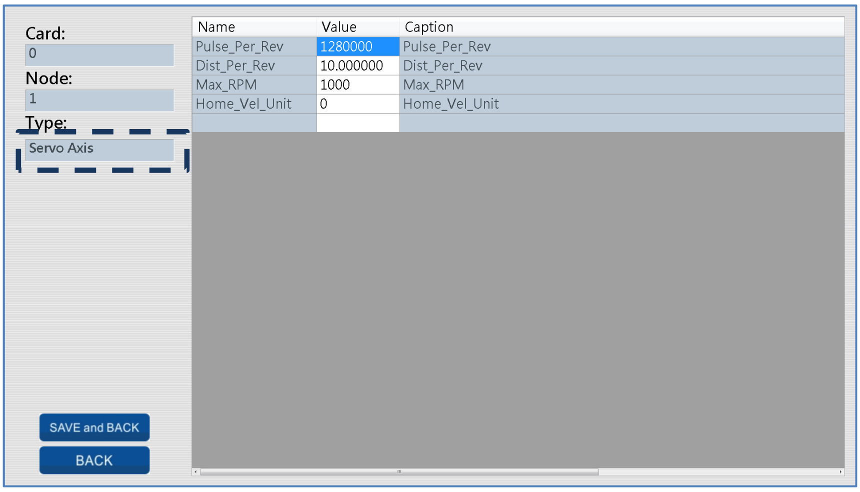


Figure 3.2.2.6 Parameter setting page of servo axis

Name	Description	Default
Pulse_Per_Rev	The required pulse number per motor revolution	1280000 PUU/rev
Dist_Per_Rev	The moving distance per motor revolution	10 mm/rev
Max_RPM	Maximum speed of motor (Unit: RPM)	1000 RPM
Home_Vel_Unit	Homing speed unit of motor	0: PUU/rev

3

- Parameter setting of remote module Ge01PH / Rm04PiM2

  When applying remote module Ge01PH / Rm04PiM2 to connect servo axis and stepping motors, you can set the resolution of output pulse and the pulse type and polarity signal type via this setting page. See the figure below.

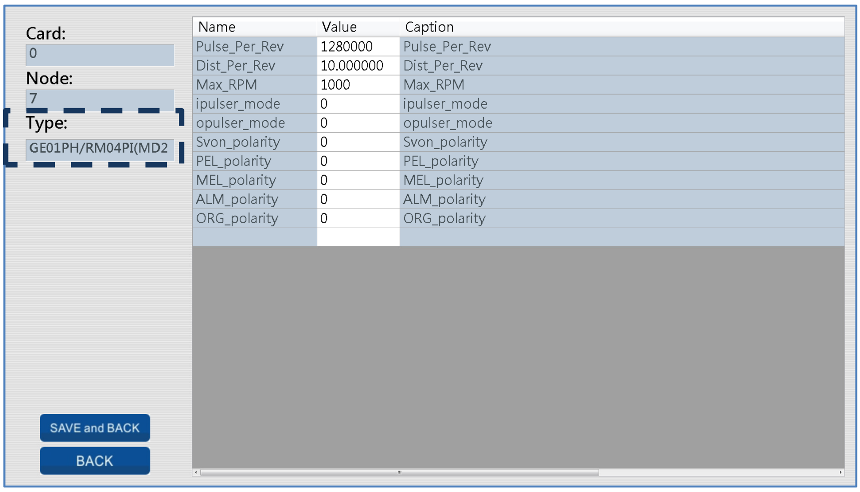


Figure 3.2.2.7 Parameter setting page for pulse type module

Parameter setting is explained in the following table:

Name	Description	Value
Pulse_Per_Rev	The required pulse number per motor revolution	1280000 pulse/rev
Dist_Per_Rev	The moving distance per motor revolution	10 mm/rev
	The rotating degree per motor revolution	360°/rev
Max_RPM	Maximum motor speed (Unit: rpm)	1000 mm/minute
ipulser_mode	Mode of input phase: 0: AB phase 1: CW and CCW pulse phase	0
opulser_mode	Mode of output phase: 0: AB pulse phase 1: CW and CCW pulse phase 2: Pulse + symbol (b, low-level) 3: Pulse + symbol (a, high-level)	0
Svon_polarity	Contact type of SVON signal output: 0: Normal open contact a (high-level triggered) 1: Normal close contact b (low-level triggered)	0
PEL_ polarity	Contact type of positive limit signal input: 0: Normal close contact b (low-level triggered) 1: Normal open contact a (high-level triggered)	0
MEL_polarity	Contact type of negative limit signal input: 0: Normal close contact b (low-level triggered) 1: Normal open contact a (high-level triggered)	0
ALM_ polarity	Contact type of servo alarm signal input: 0: Normal open contact a (high-level triggered) 1: Normal close contact b (low-level triggered)	-
ORG_ polarity	Contact type of origin signal input: 0: Normal open contact a (high-level triggered) 1: Normal close contact b (low-level triggered)	-

- Mechanical parameter setting for remote module RM04AD
  When applying remote module RM04AD, you can set the speed, range of analog input signal, and use the function of averaging the input signal values. See the figure below.

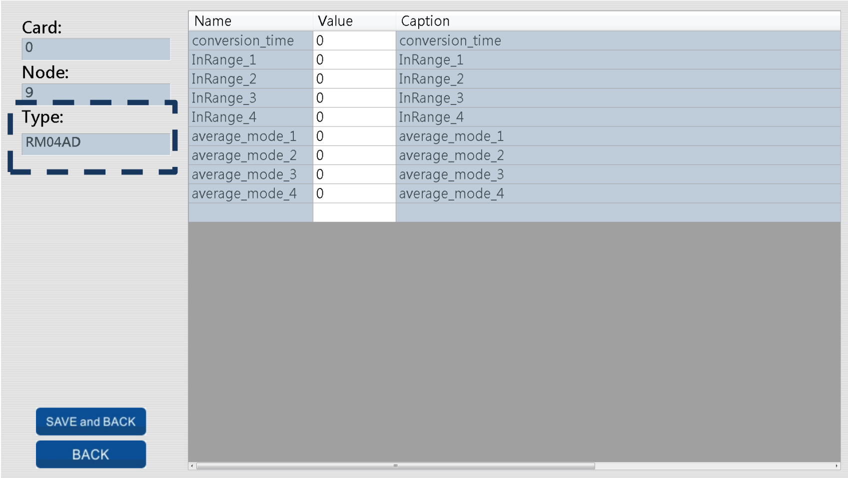


Figure 3.2.2.8 Parameter setting for analog input

Parameter descriptions are shown as follows.

(1) conversion_time: AD conversion speed. The parameter value is 0 ~ 6; default is 0. See table below.

Value	Output data rate (Hz)	-3dB Frequency (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; Parameter value: 0 ~ 5; Default: 0. See table below.

Value	Valid input range
0	±10 V
1	0 V ~ 10 V
2	±5 V
3	0 V ~ 5 V
4	Reserved
5	0 ~ 20 mA

(3) average_mode_1,2,3,4: AD average mode; Parameter value: 0 ~ 5; Default:0. See table below:

Value	Sampling number of the average setting
0	0
1	2
2	4
3	8

Value	Sampling number of the average setting
4	16
5	32

■ Mechanical parameter setting for remote module RM04DA

Module RM04DA can set four sets of analog output offset (DA offset) and output range. See the following figure.

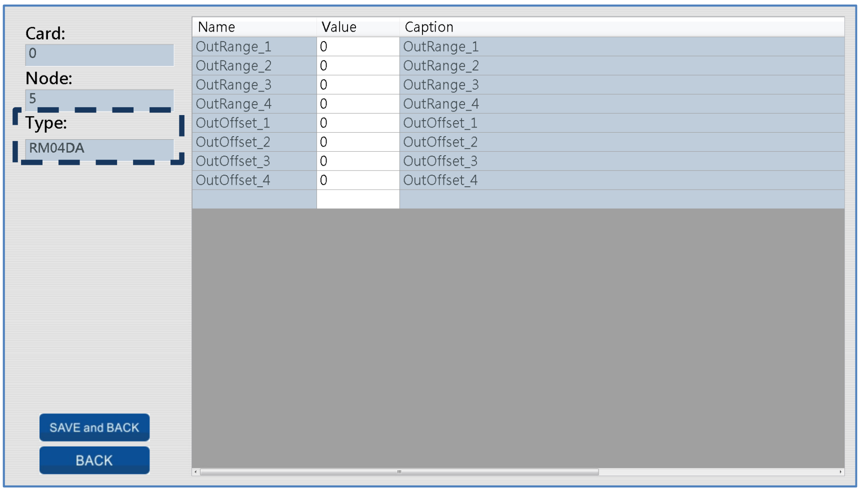


Figure 3.2.2.9 Parameter setting page for analog output module

Parameter setting is as follows:

(1) OutRange_1,2,3,4: DA output range. The setting value is 0~7; default is 0. See the table below:

Value	Valid input range
0	0 V ~ 5 V (default)
1	0 V ~ 10 V
2	±5 V
3	±10 V
4	Reserved
5	4 ~ 20 mA
6	0 ~ 20 mA
7	0 ~ 24 mA

(2) OutOffset_1,2,3,4: Output calibration offset (DA offset); Setting value: -128 ~ +127; Default: 0. The offset is around 0.03814 mV each time of calibration. See the table below:

Value	Definition of calibration offset adjustment
127	4.844 mv ≅ 127 × 0.03814
126	4.806 mv ≅ 126 × 0.03814
…	…..
1	0.038 mv ≅ 1 × 0.03814
0	none (default)
-1	-0.038 mv ≅ -1 × 0.03814

3

Value	Definition of calibration offset adjustment
…	…
-127	-4.844 mv ≅ -127 × 0.03814
-128	-4.882 mv ≅ -128 × 0.03814

3.2.3 Parameter Import / Export

This function allows you to export parameter files to the specified folder and import the parameter files. Parameter files can include parameters of each node, MPM, configuration of SNC. See the figure below.

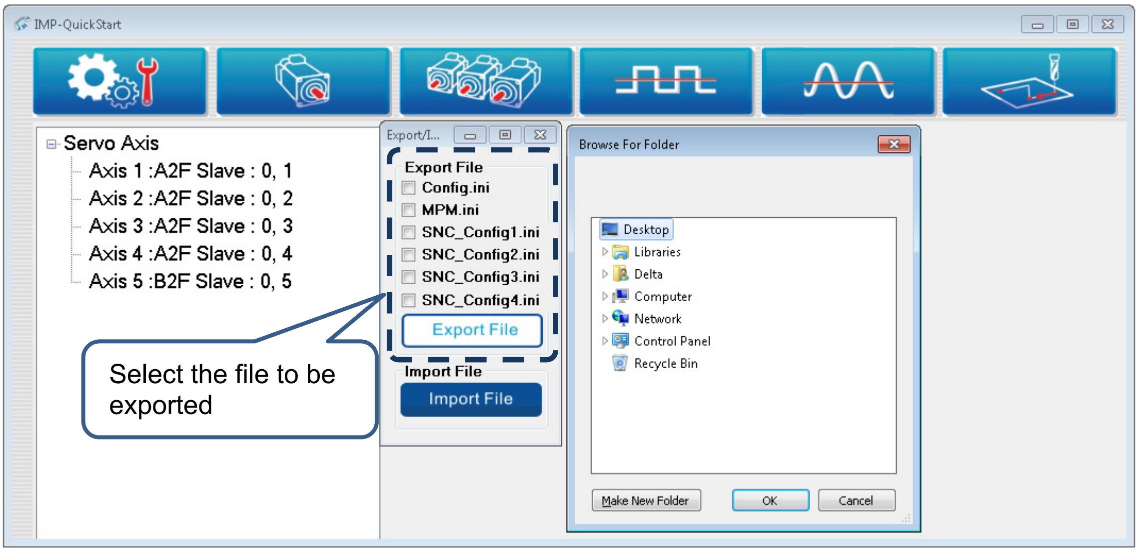


Figure 3.2.3.1 Screen of parameter export

3.2.4 Auto start IMP after PC starts

To run IPC Motion Platform as soon as PC is started, please go to [Setting] and check [Enable IMP AUTO Run]. (Then, the item will display [Cancel IMP AUTO Run])

3.3 Trials and operation

3.3.1 Single-axis control

3

Modules that support single-axis control include servo drives, GE01PH, and RM04PI/Mode2. The connected modules will be displayed on the left hand side of the screen. Double click on the servo axes, the dialogue box will show the corresponding interface for motion control. See figure below.

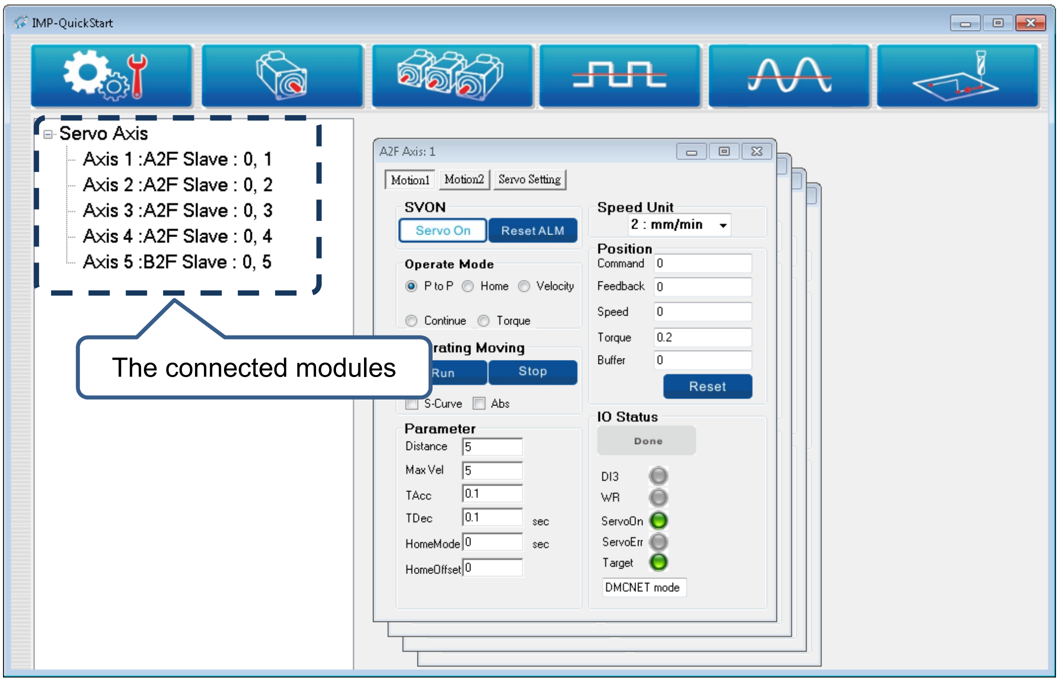


Figure 3.3.1.1 Screen of single-axis control

To enter the interface of single-axis control and start the operation, the servo drive has to be in Servo On state. See the figure below.

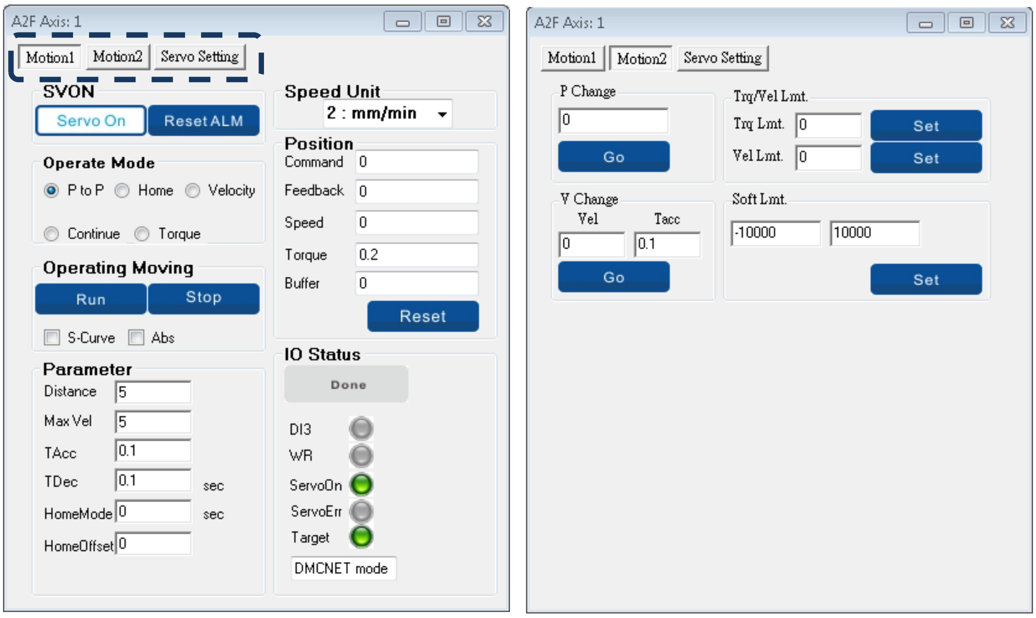


Figure 3.3.1.3 Interface of single-axis control

Settings of Motion1, Motion2, and Driver Setting (Read/Write servo parameters) are described as follows:

■ Motion1

Item	Description
SVON	It includes Servo On and Alarm reset (Reset ALM).
Speed Unit	There are three units available: pulse/sec, Feedrate %, and mm/min
Operate Mode	It includes P to P, Homing, Velocity, Continue, and Torque mode.
Operating Moving	It includes Run (start running), Stop (Stop running), S-curve, ABS (Absolute/Relative position).
Parameter	It is for setting distance, start velocity (Start Vel), Max. velocity (Max Vel), Acceleration time (TAcc), Deceleration time (TDec), Home Mode and Home Offset.
Position	It includes resetting command (Reset). This section also displays position command, feedback position, motor speed, torque, and motion command that has not been executed (Buffer).
IO Status	It displays the current motion status, such as state of DI3, power on (PWRON), servo error (ServoErr), and target reached (Target).

■ Motion2

Item	Description
P Change	Change target position during motion.
V Change	Change current moving velocity during motion.
Trq/Vel Lmt.	For setting the max. torque in speed mode and max. velocity in torque mode.
Soft LMT.	Set the software limit.

■ Servo Setting (Read/Write servo parameters)

Through the servo setting interface of single-axis control, users can write the parameters of A2F, B2F, and M series servo drives via DMCNET

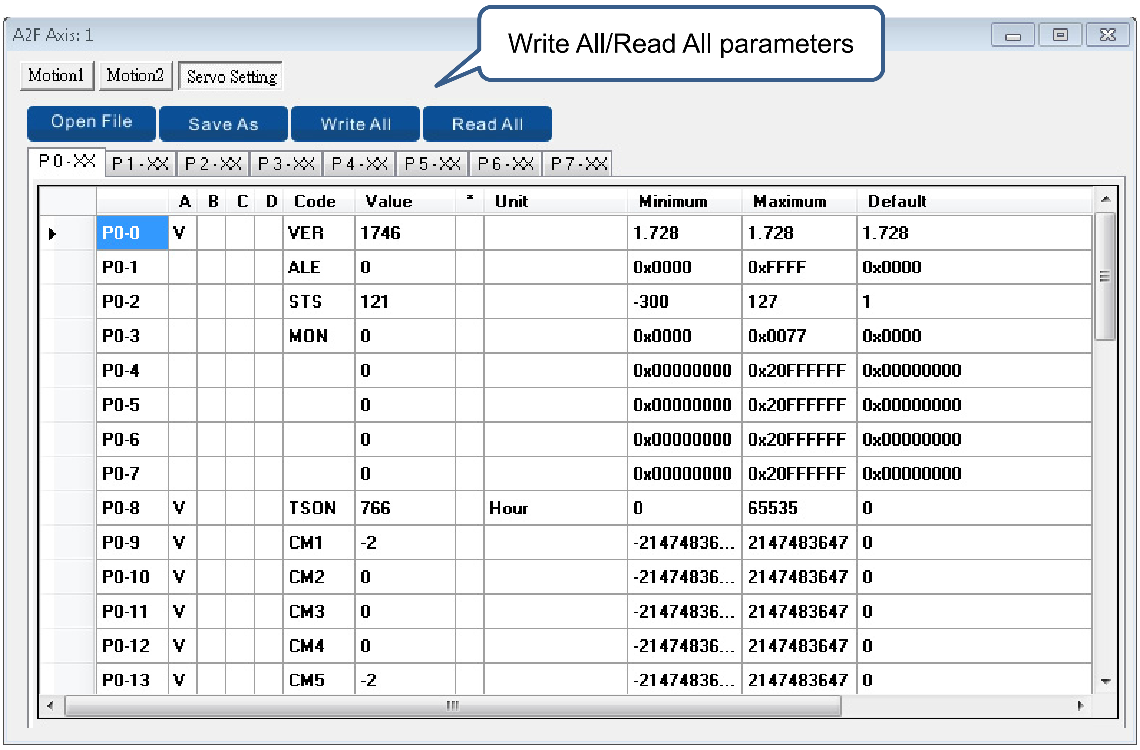


Figure 3.3.1.5 Read parameters of servo drive

3.3.2 Multi-axis interpolation

Click on multi-axis button in the Main screen. Then, select the motion card number to be used and the interface of multi-axis interpolation will pop out. See the figure below.

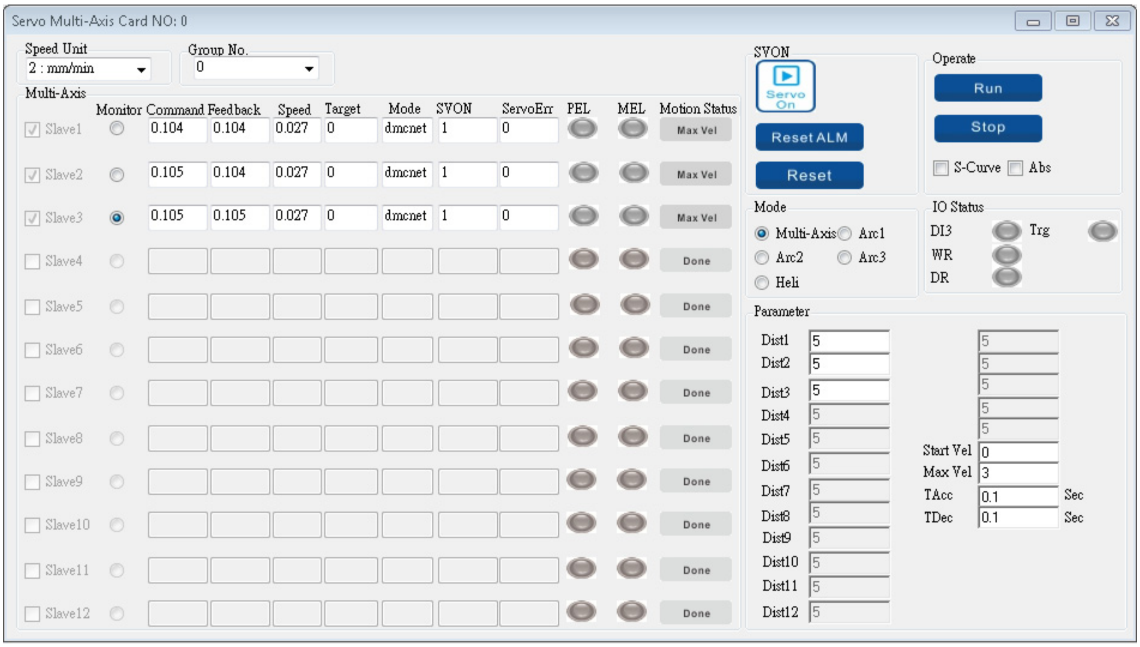


Figure 3.3.2.1 Multi-axis interpolation

Description of each function is as follows:

Item	Sub-item	Description
Speed Unit		Motion speed units include: pulse/sec, Feedrate% and mm/min.
Group No.		40 interpolation groups are available for selection. They are the same as the PLC motion group.
Multi-Axis		Select the axis for multiple-axis interpolation, and these axes are regarded as a group and the state of the following sub items will be shown at the same time.
	Sub-item	Description
	Command	Display the command end position.
	Feedback	Display motor’s feedback position.
	Speed	Display the current moving speed of servo motor.
	Target	Display that positioning of the servo is completed.
	Mode	Servo operation mode: DMCNET, position, velocity, torque, and homing mode.
	PWRON	Display servo status (SVON).
	ServoErr	Display servo error code.
	PEL	The servo motion exceeds the positive limit.
	MEL	The servo motion exceeds the negative limit.
	Motion status	Display servo motion status.
SVON	Sub item	Description
	Servo On	Click this button to switch the servo drive to “Servo On” state.
	Reset Alm	Click this button to reset the alarm.
	Reset	Click this button to restore the default setting of multi-axis interpolation

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Mode</td><td>Multi-axis mode: Linear interpolation
Arc 1 mode: with the known coordinate of arc’s circle center and angle)
Arc 2 mode: with the known coordinate of end point and arc angle)
Arc 3 mode: with the known coordinate of circle center and end point)
Heli mode: Three-axis helical motion</td></tr>
<tr><td>Operate</td><td><table>
<tr><th>Sub-item</th><th>Description</th></tr>
<tr><td>Run</td><td>Start interpolating</td></tr>
<tr><td>Stop</td><td>Stop interpolating</td></tr>
<tr><td>S-curve</td><td>Set the S-curve or T-curve.</td></tr>
<tr><td>Abs</td><td>Check this box to input absolute coordinates; Uncheck it to input relative coordinate.</td></tr>
</table></td></tr>
<tr><td>IO status</td><td>This works with the Monitor function in multi-axis interpolation. It displays the state of DI3, servo warning (WR), data error (DR), and triggering bit (Trg).</td></tr>
<tr><td>Parameter</td><td>Set motion parameters such as distance, starting velocity, max. speed, acceleration time, and deceleration time. Parameters will be shown based on the selected motion mode and axial number.</td></tr>
</table>

3

3.3.3 Digital input / output

Modules that support DI/DO include RM64MN, RM64NT, RM32MN, RM32NT, RM32PT, and RIO3232RT5. The left side of the screen will show the DI and DO modules that have been connected and their I/O point range sorting is based on module type, card No., and node No.. Click on it and the monitor and force input screen will pop out.

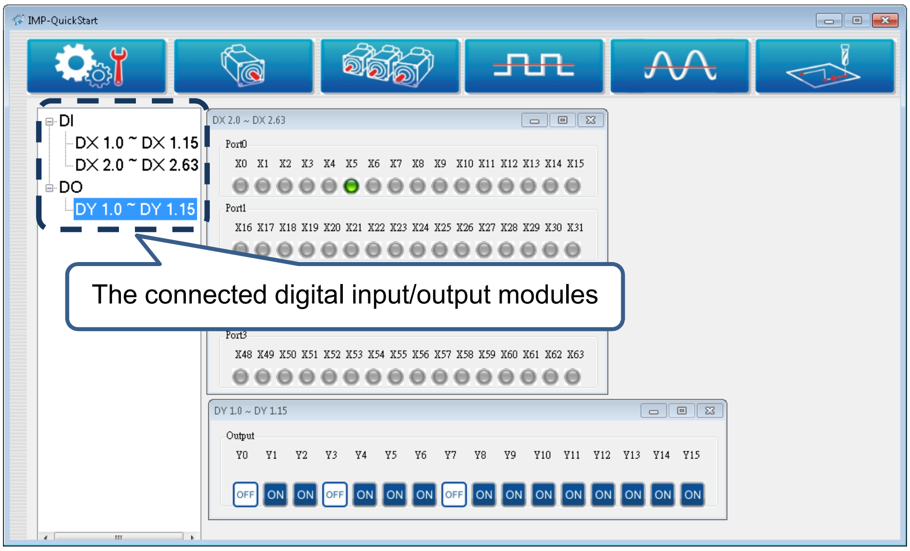


Figure 3.3.3.1 Input monitoring

Example 1:

See the above Figure. Connect module RM32PT and RM64MN in sequence.

a. RM32PT has 16 input points and 16 output points. The DI display are DX1.0 ~ DX1.15, which are for the second node of the DI module. The DO display are DY1.0 ~ DY1.15, which are for the first node of DO module.

b. RM64MN has 64 points. The display of input module is DX2.0 ~ DX2.63, which are for the first node of the DI module.

Example 2:

See the figure below. Connect module RM32PT and RM64NT in sequence.

a. RM32PT has 16 input points and 16 output points. The display are DX1.0 ~ DX1.15, which are for the first node of the DI module. The DO display are DY2.0 ~ DY2.15, which are for the first node of the output module.

b. The display of input module is DY2.0 ~ DX2.63, which are for the second node of the DI module.

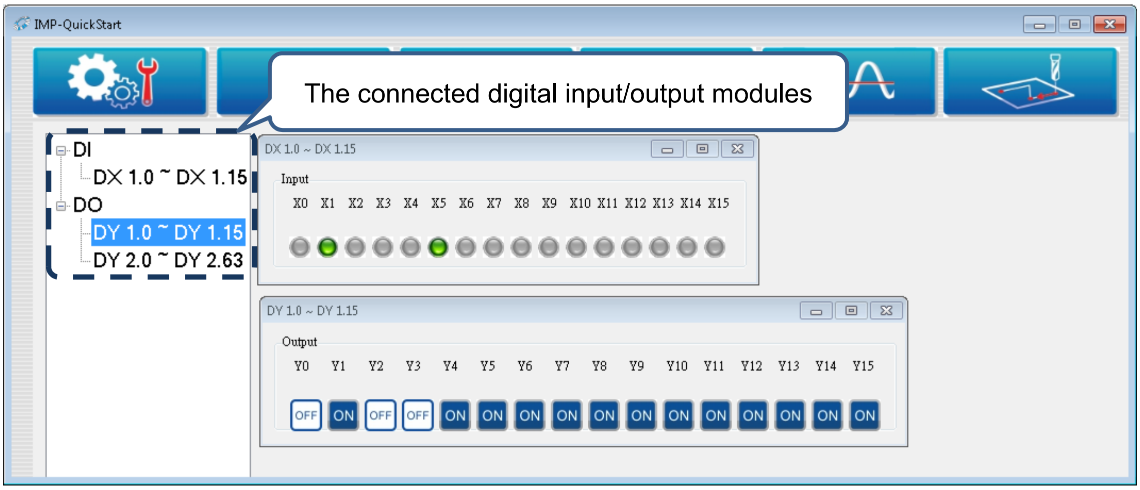


Figure 3.3.3.2 Output monitoring

3.3.4 Analog Input / Output

The modules that support analog input and output include RM04AD and RM04DA. In the left side of the screen, it shows the analog input/output modules that are connected and display the channel range in accordance with its type, card number, and station number in sequence.

3

Example 1:
As shown in the following figure, the system is connected with two RM04AD and one RM04DA module. The two RM04AD are 4-channel analog input modules, which are displayed as CH1 ~ CH4 and CH5 ~ CH8 respectively. It means these digital input modules can use the analog input channel 1 to 8 of the PLC. The RM04DA is a 4-channel analog output module, which is displayed as CH1 ~ CH4. It means this digital output module can use the analog output channel 1 to 4 of the PLC.

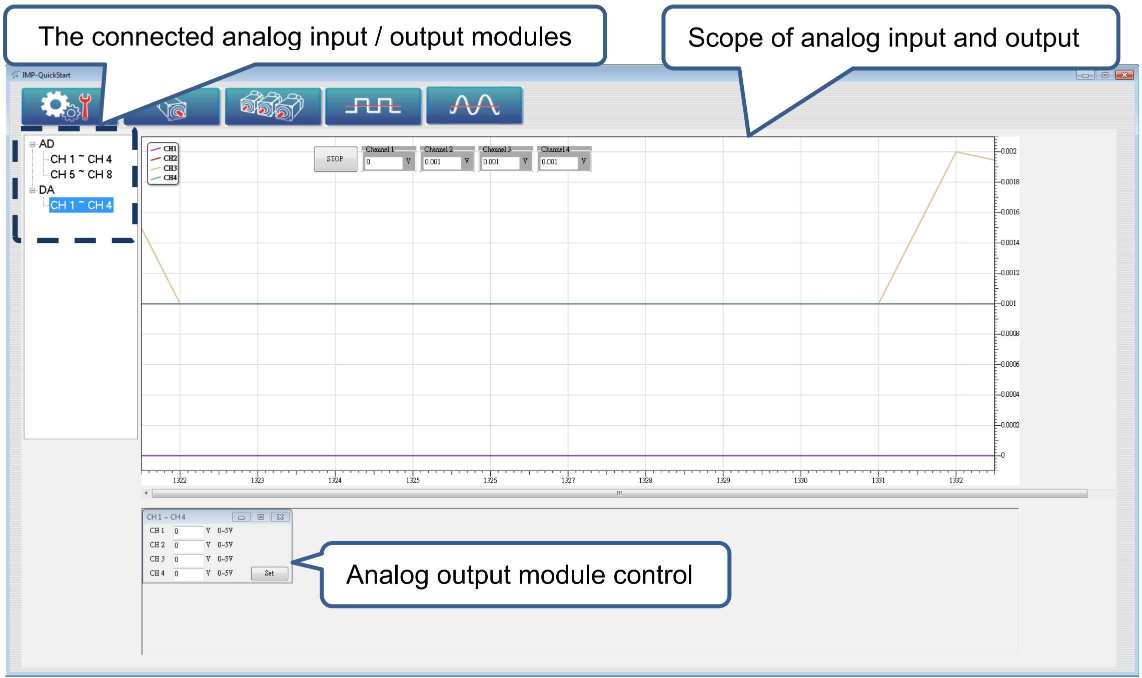


Figure 3.3.4.1 CH1 ~ CH 4

3

3.3.5 Software numerical control (Option)

The SNC interface is shown as follows. It is for testing functions of numerical control, parameter setup, backlash compensation, coordinate setting for G52 ~ G59, cutter length and radius setting, tool offset setting, and tool length setter. All these settings mentioned above will be saved by the system. (Its setting is similar to HMI)

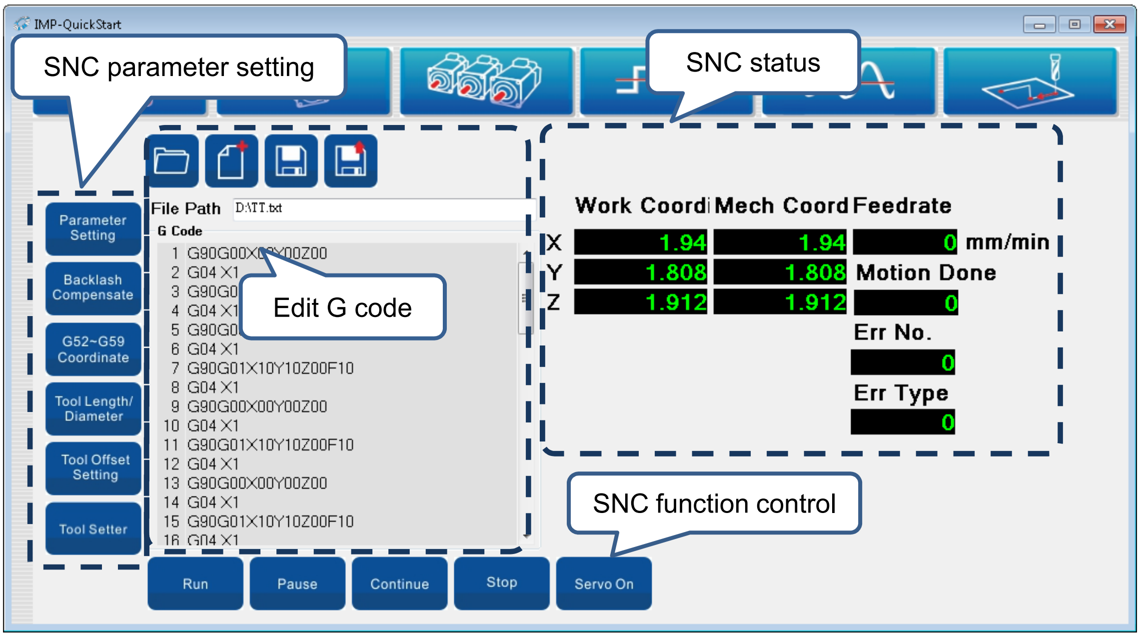


Figure 3.3.5.1 Interface of Software numerical control

Ladder Editor 4

This chapter will introduce IMP Ladder Editor that is integrated in DOPSoft 3.00.01. Please refer to DOPSoft user manual for insallation and HMI screen editing functions.

4.1 Introduction to Ladder Editor ···· 4-2
4.1.1 How to start Ladder Editor ···· 4-2
4.1.2 Program upload and download ···· 4-5
4.2 Create new ladder program and settings ···· 4-10
4.3 Other functions ···· 4-14

4

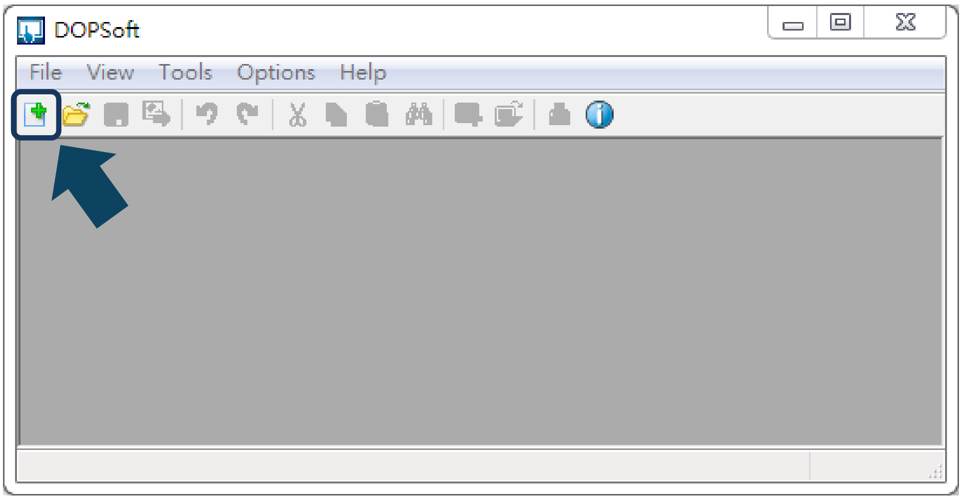
4.1 Introduction to Ladder Editor

This chapter will tell you how to start the Ladder Editor and function of each part of the interface.

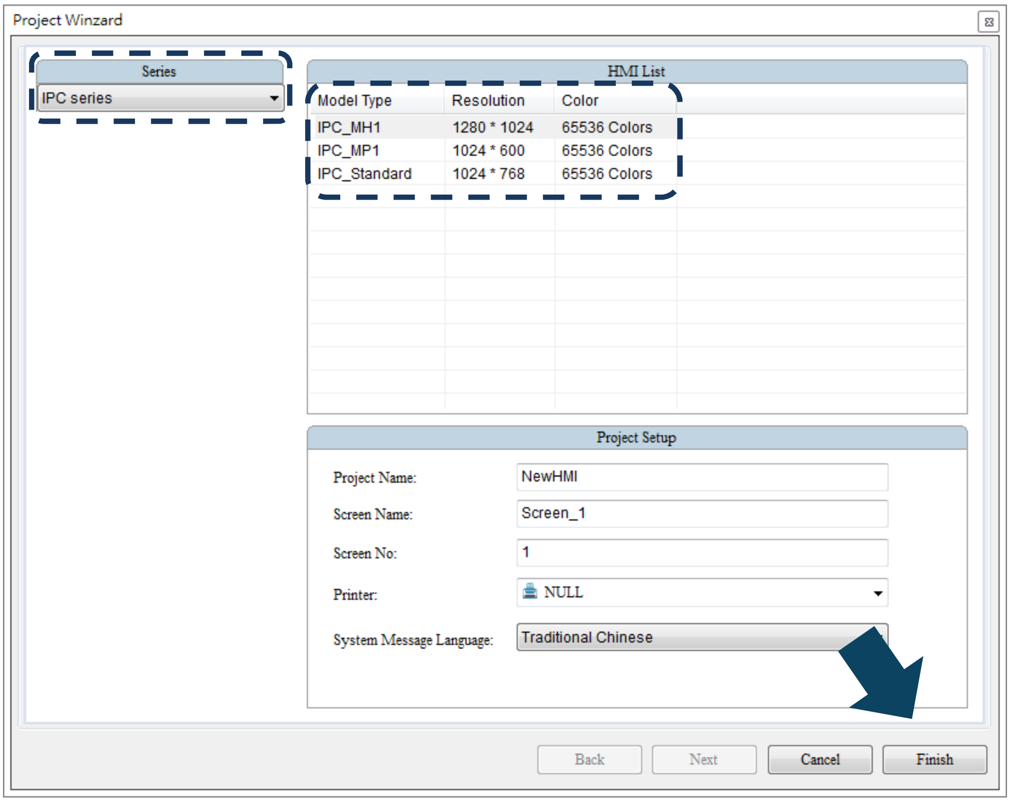
4.1.1 How to start Ladder Editor

Ladder Editor has been integrated in DOPSOFT3.00.01. You may start it by following the steps below.

1. Open DOPSoft and click on **New**.

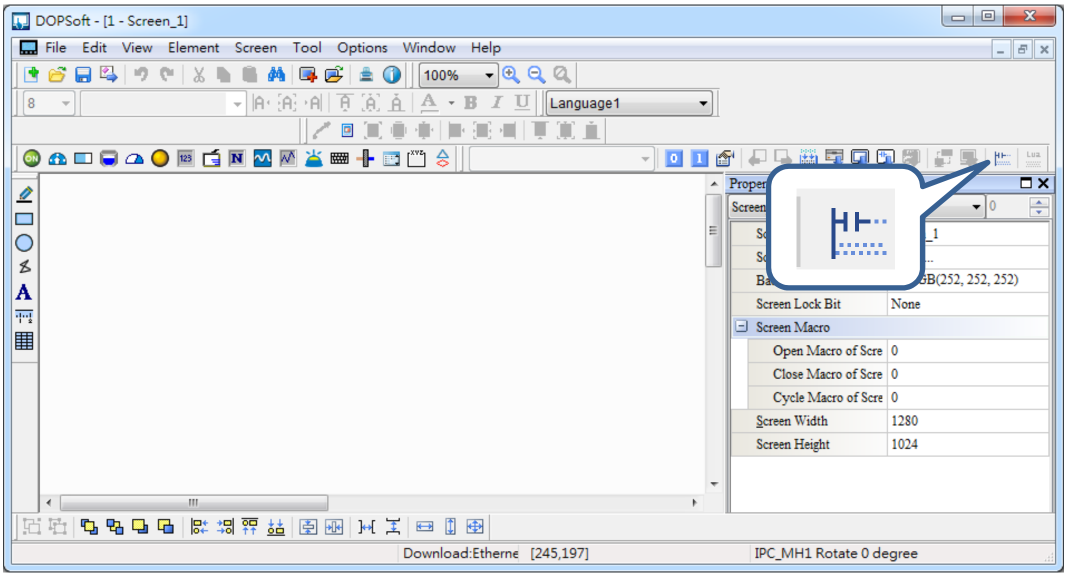


2. Select HMI model type and click on **Finish**.

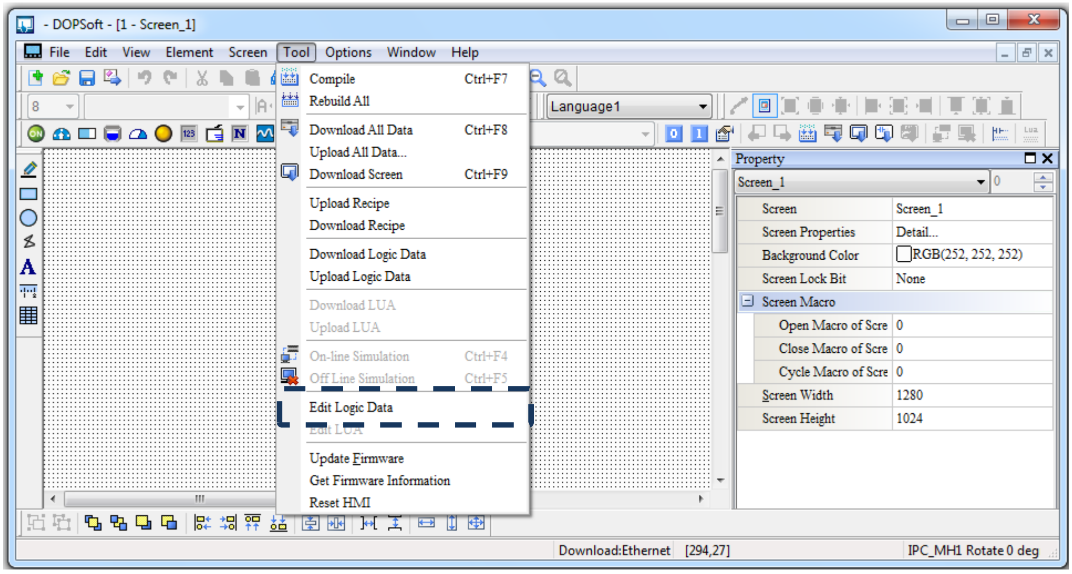


3. Open Ladder Editor.

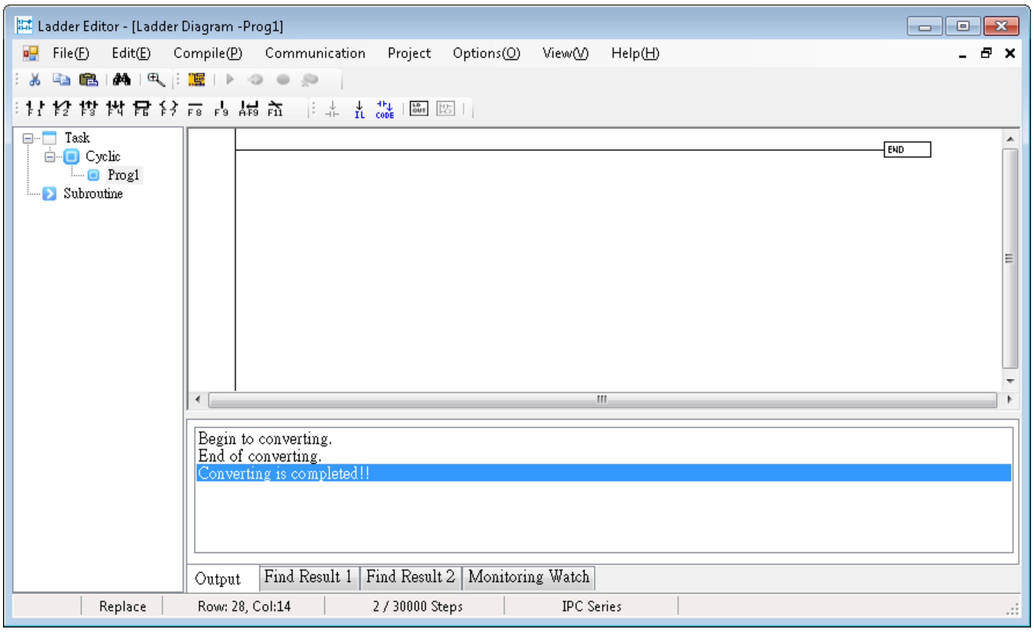
Click [Edit Logic Data] in the tool bar and start Ladder Editor.



Or, you can also go to [Tool] > [Edit Logic Data] to start it.



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



5. Upload/download data (connection settings)

After editing the screens and programs by DOPSOFT3.00.01, upload/download them to IMP via Ethernet so they can be executed by your PC.

4.1.2 Program upload and download

There are two modes available based on whether DOPSoft3.00.01 is installed in one IPC or separately, IPC/PC.

Mode 1: Install both IPC Motion Platform and DOPSOFT3.00.01 in an IPC. And you can edit the software screens and programs via DOPsoft, then trammit them via internal memory sharing.

Mode 2: Install IPC Motion Platform in the IPC and install DOPSOFT3.00.01 in your own PC. After editing the screens and programs with DOPsoft, transmit them via Ethernet for PAC to execute these file by IPC Motion Platform.

Download all data

When downloading all data, both the screen data and recipe will be downloaded to the PAC that has executed IMP. To download all data, you can go to [Tool] > [Dlownload all data]. Or, click on the  icon in the toolbar or press **Ctrl+F8**. Make sure the PC and IMP PAC are connected and IMP software is enabled. Otherwise, error messages will pop out.

Input IP address:

To download data in Mode 1, the IP address is 127.0.0.1; In Mode 2, input the IP address of the PAC.

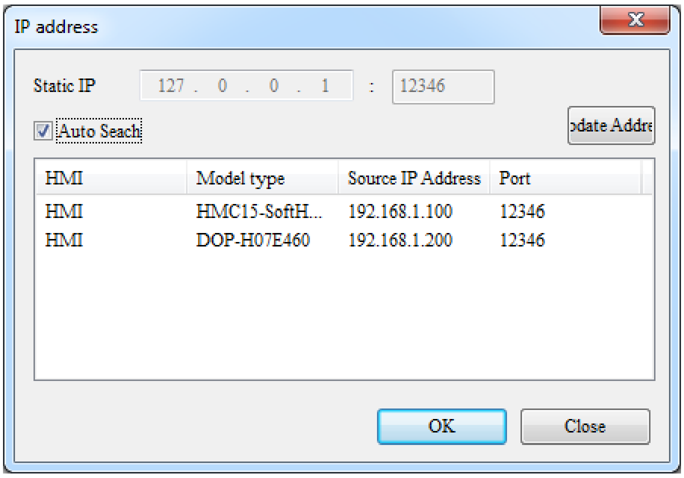


Figure 4.2.1.1 IP search/setting and download

Normal transmission:

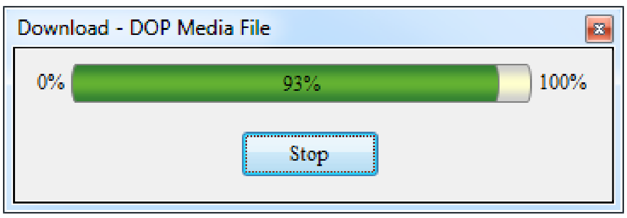
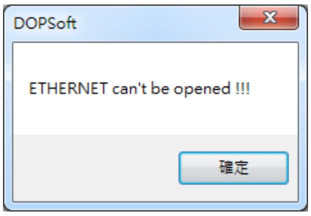


Figure 4.2.1.2 Dowload all data

When transmission is failed, the system will prompt the following message in different conditions.

An error acours when connecting via Ethernet or IMP is not correctly started by PAC:



Connecting cable is removed or transmission is cut off in the download process:

Upload all data

When uploading all data, you will be asked to enter the password (Figure 4.2.1.3). Please enter the system default password [12345678] here. To change the password, you can go to [Option] > [Configuration].

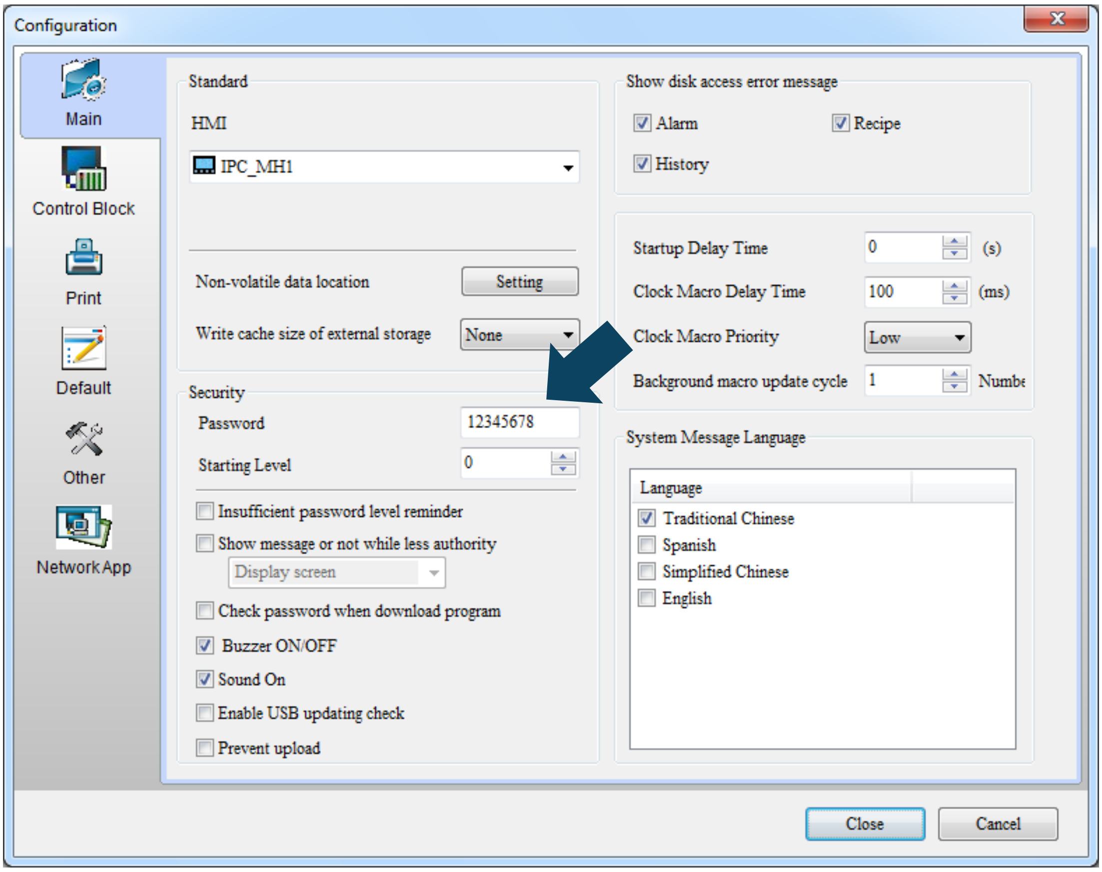


Figure 4.2.1.3 Set up security password

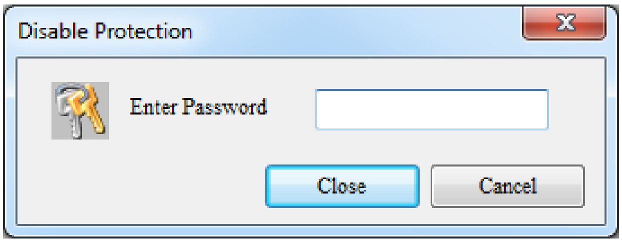


Figure 4.1.2.4 Upload all data (default: 12345678)

After entering the password, the system will upload the screen data and stops until completed (100%). To stop uploading, press **Stop**.

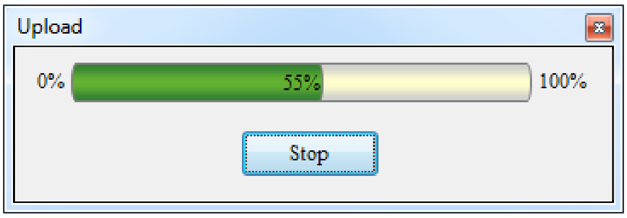


Figure 4.1.2.5 Uploading

Then, you will be asked to save the screen file. See figure below.

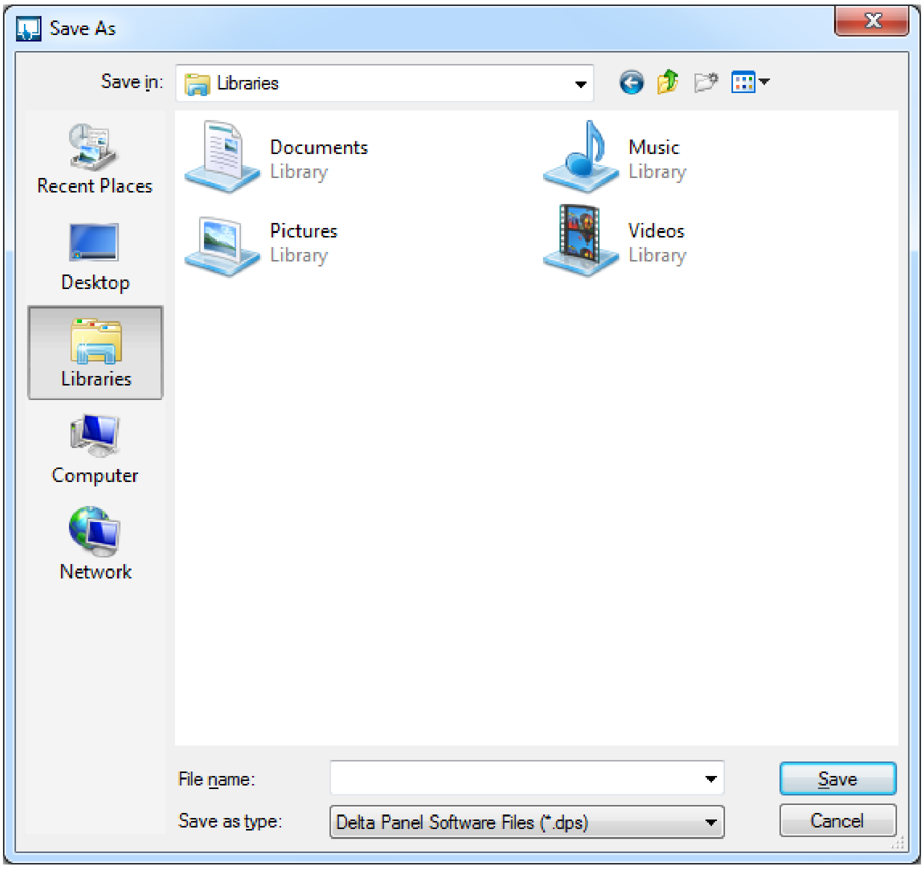


Figure 4.1.2.6 Upload data and Save as…

Apart from uploading the screen data to the editing PC, you can go to [Options] > [&Emvironment] to set whether to include picture data when uploading.

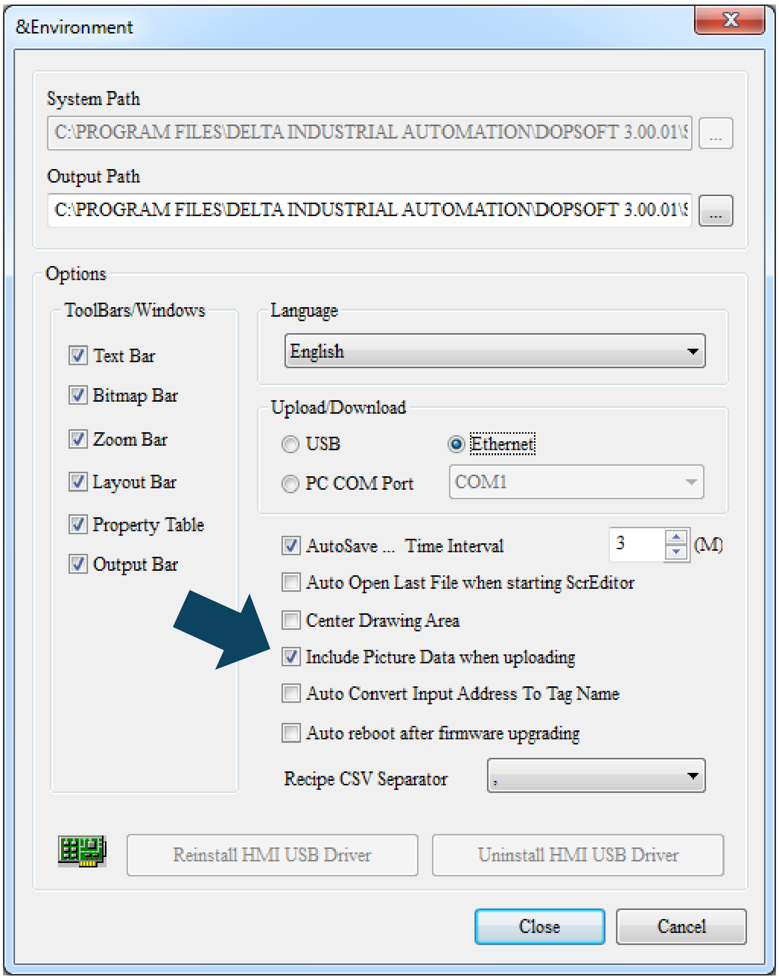


Figure 4.1.2.7 Include picture data when uploading

Download screen data

This function allows you to download screen data without the recipe. The steps to carry out his function is the same as that illustrated in [Download all data]. You can do it by going to [Tool] > [Download screen data] or click on the icon in the toolbar, or press **Ctrl+ F9.**

Upload recipe

The steps to execute upload recipe is identical to the steps to upload all data. You will have to enter the password in order to upload the recipe to PC. And its steps to set password are also the same as described in [Upload all data] section.

4

Download recipe

To download the recipe data only, users can do it by executing **Download recipe**. If you need to modify the recipe without editing other screen data, this function will help you save the download time. When using this function, you will be asked to select the recipe file to be downloaded (.rcp). Then, it will be downloaded to HMI.

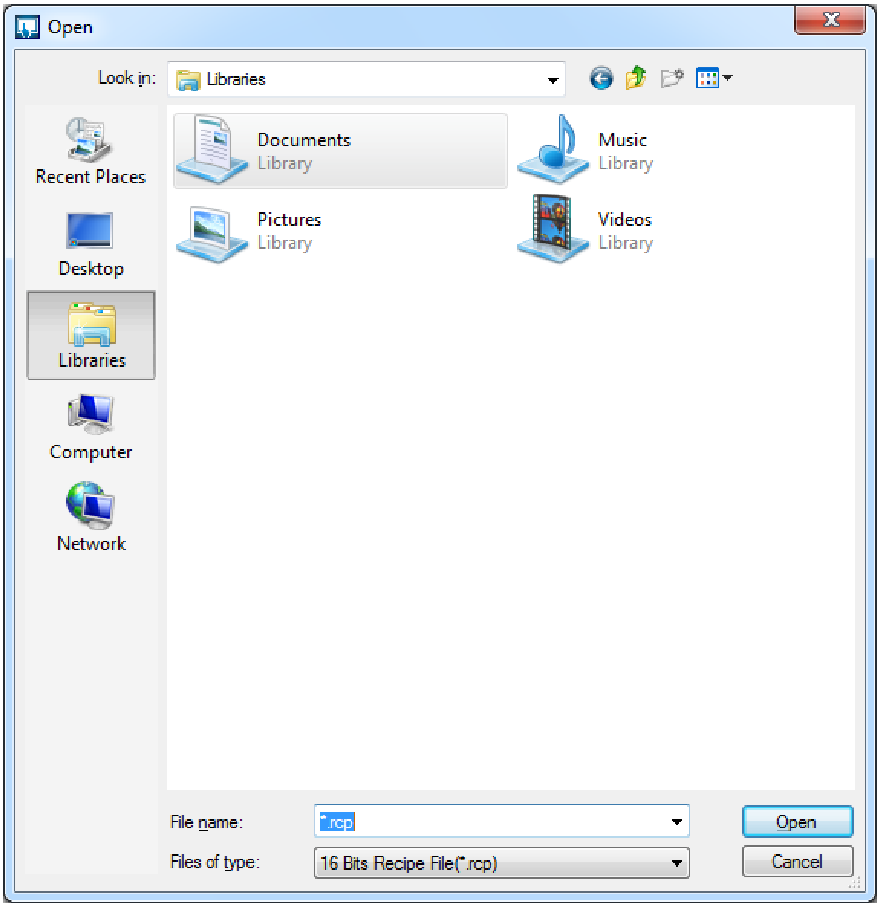


Figure 4.1.2.8 Select the recipe file to be downloaded

Download Logic Data

This function enables you to download the edited logic data to IMP PAC.

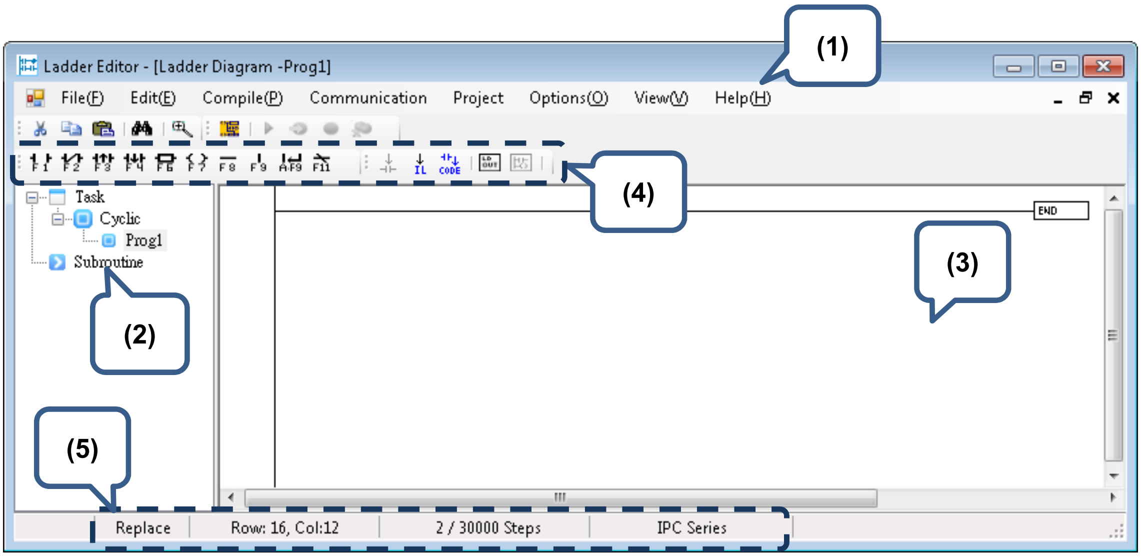
Upload Logic Data

This function enables you to upload logic data from IMP PAC to PC.

4

4.2 Create new ladder program and settings

The interface of Ladder Editor is described as follows:



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

(2) PLC project tree view: It shows the ladder program structure in the current PLC project.

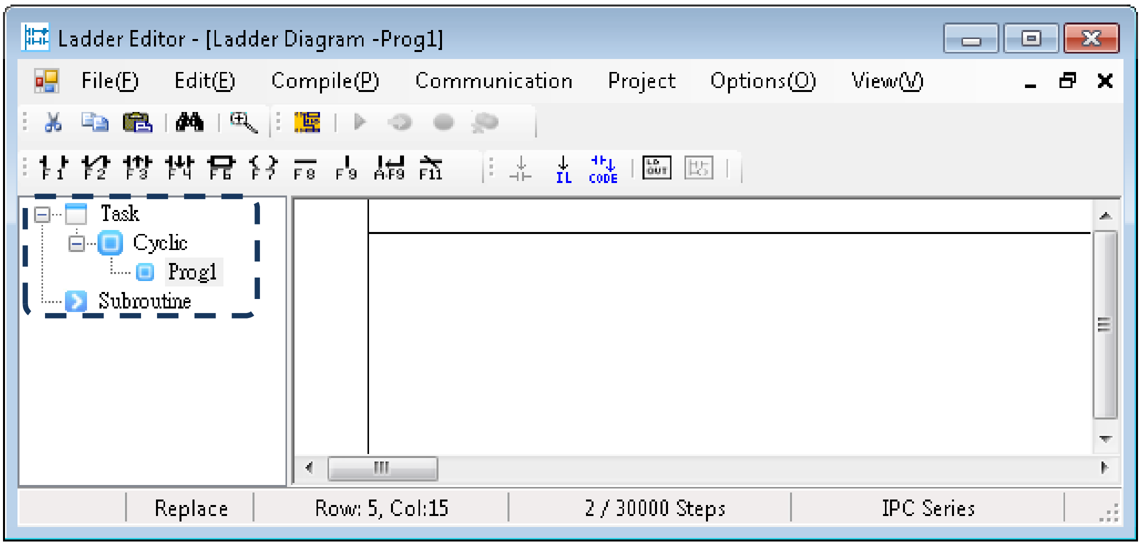
(3) Program editing section: It allows users to edit the PLC program.

(4) Application options: It allows you to select from output window, find window, and monitoring device window.

(5) Editing status: It displays current editing mode, [Replace] or [Insert].

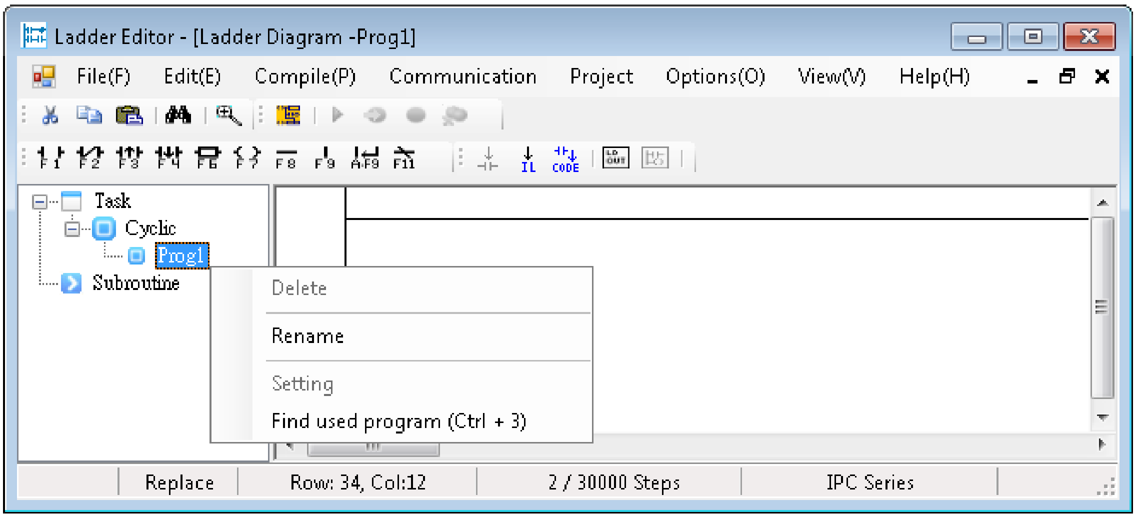
■ Cyclic Task

Cyclic Task can be found in the PLC project tree view on the left hand side of the screen. See the figure below.



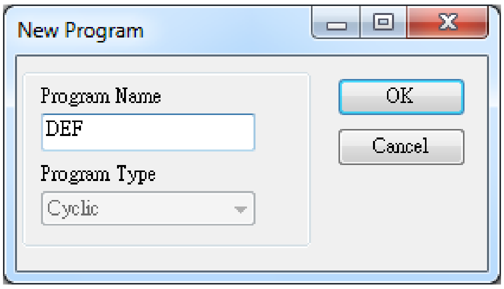
Change the program name

1. To change the program name, move the cursor to the program name, right-click the mouse and then click on [Rename].

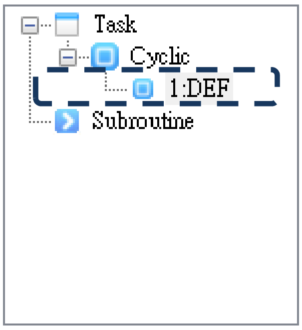


4

2. Then, the screen of [New Program] will pop out. Input the new program name in the field. Please click on **OK**.



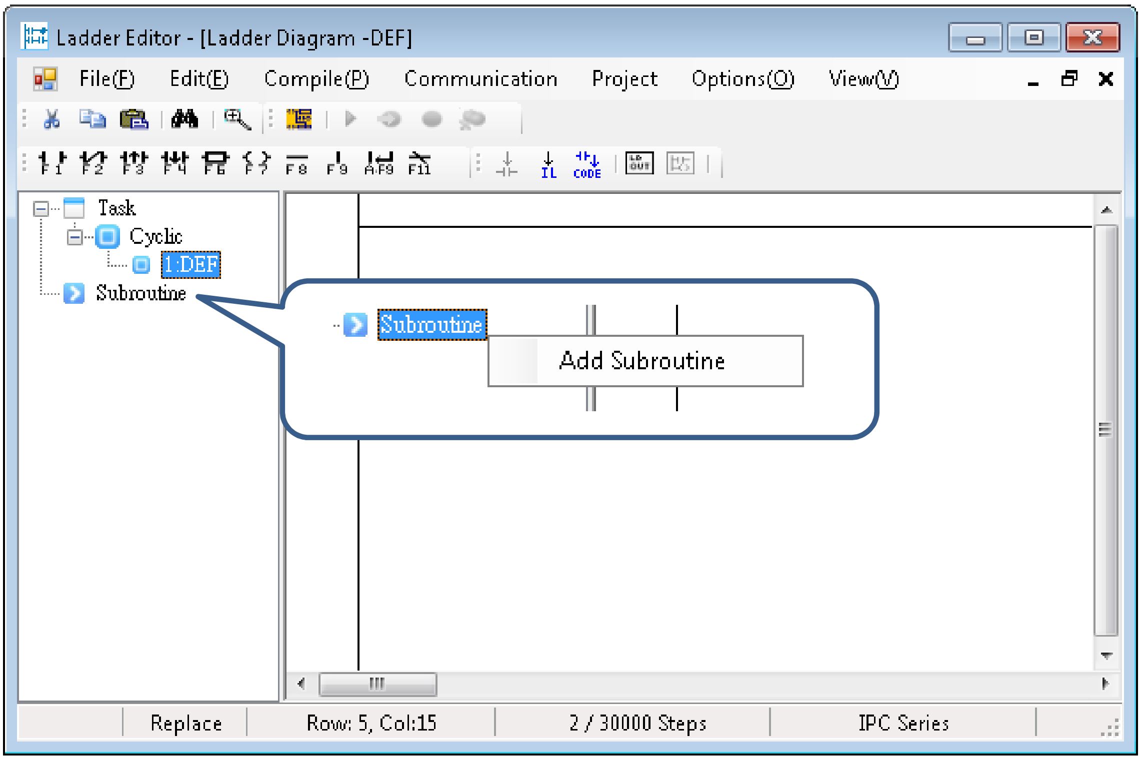
3. The name is changed successfully.



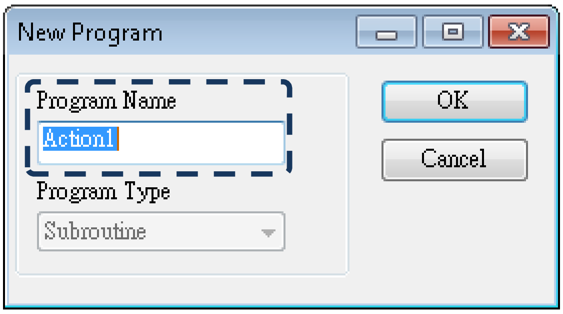
■ **Subroutine**

Add Subroutine

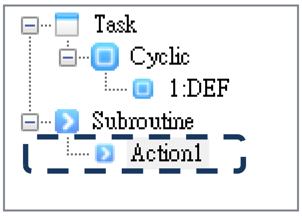
1. Right-click the mouse and click [Add Subroutine]. Then, the [New Program] window will pop out.



2. Input the program name in the [New program] window with the maximum character of 16. Please click on **OK**.

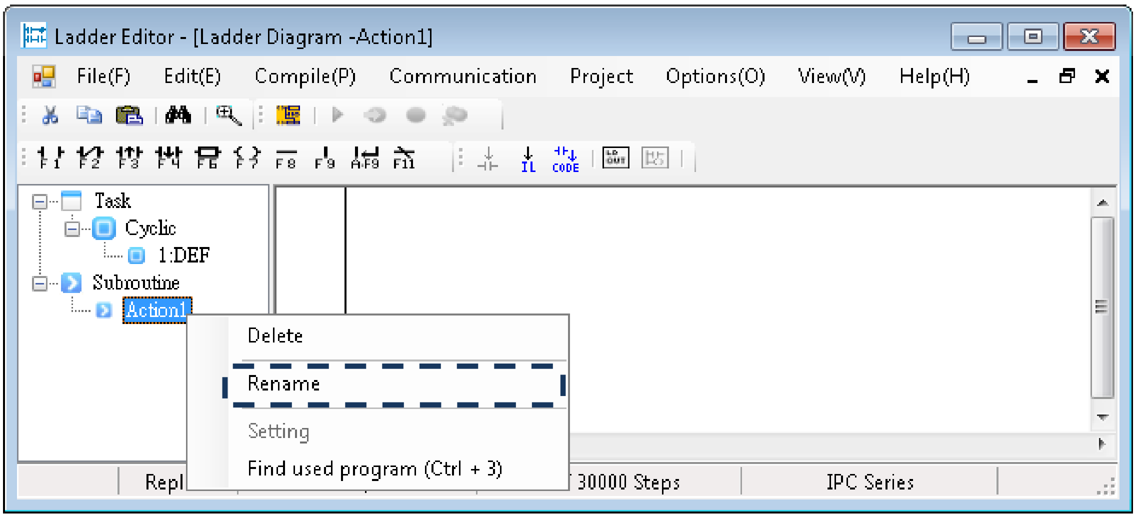


3. The name will be shown in the PLC project tree view.

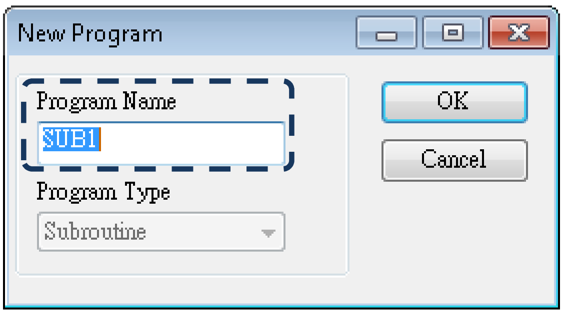


Change the program name

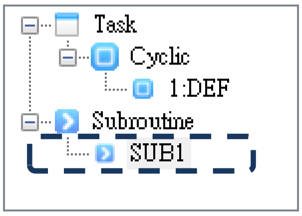
1. Right-click the mouse and click [Rename]. Then, the [New Program] window will pop out.



2. Input the new program name in the [New Program] window. Then, click on **OK**.

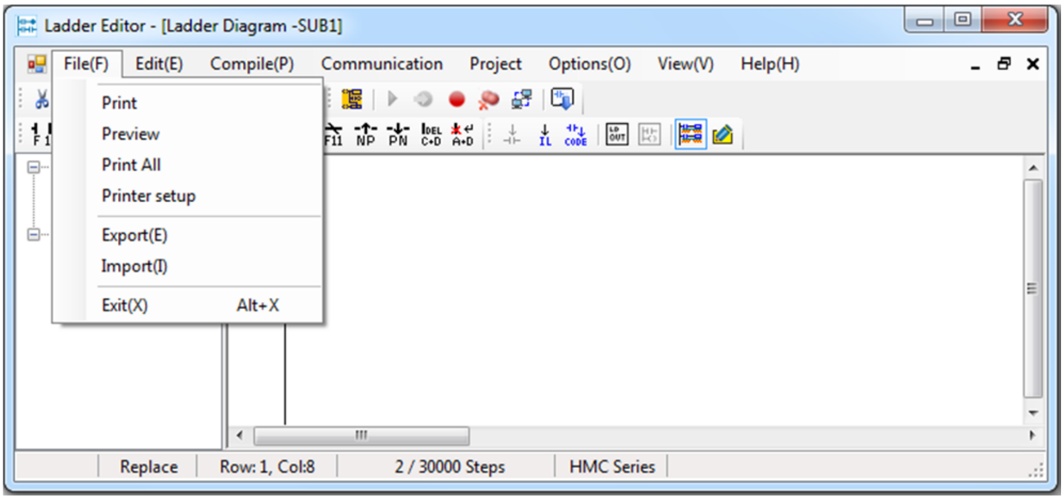


3. The subroutine name will be changed. And the subroutine name being called in the ladder program will also be changed in the meantime.



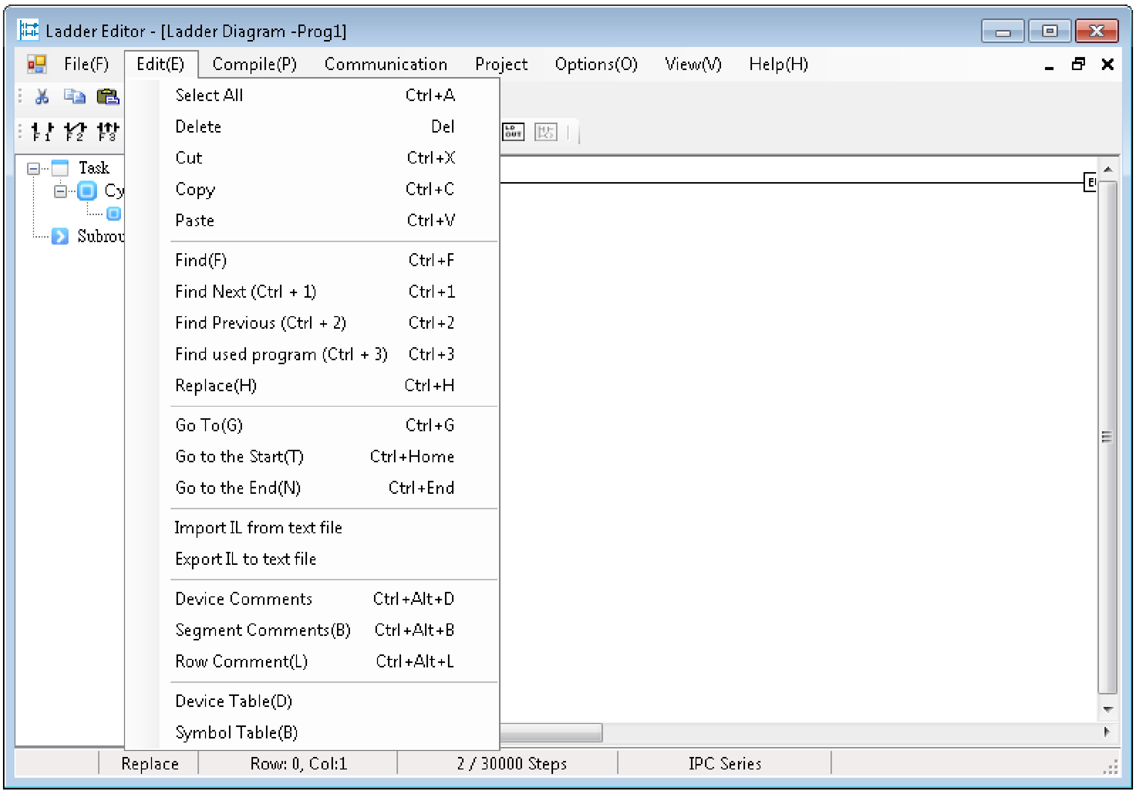
4.3 Other functions

File



Item	Description
Print	Print the ladder program you are currently editing.
Preview	Printing preveiw the ladder program you are currently editing.
Print All	Print all the unencrypted ladder programs.
Printer setup	Set printing page layout, including size, margins, and orientations.
Export(E)	Export ladder program (.cwp)
Import(I)	Import external ladder program (.cwp)
Exit(X)	Exit Ladder Editor

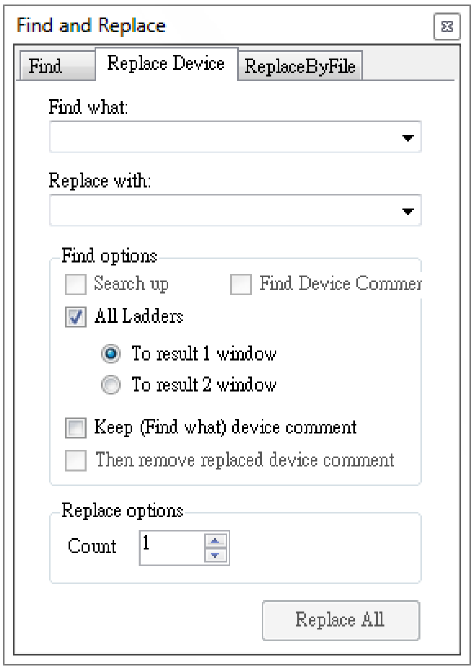
Edit



Item	Description
Select All	Select all contents in current ladder program.
Delete	Delete the selected content.
Cut	Cut the selected content.
Copy	Copy the selected content.
Paste	Paste the selected content.
Find(F)	Find the target in current program or all programs.
Replace(H)	Find the device in current/all programs and replace with the specified device.
Go To(G)	Go to STEP command.
Go to the Start(T)	Go to STEP 0 command of the program.
Go to the End(N)	Go to the END command of the program.
Device Comments	Edit device comments.
Segment Comments(B)	Edit segment comments.
Row Comment(L)	Edit row comments.
Device Table(D)	Open the device table.
Symbol Table(B)	Open the symbol table.

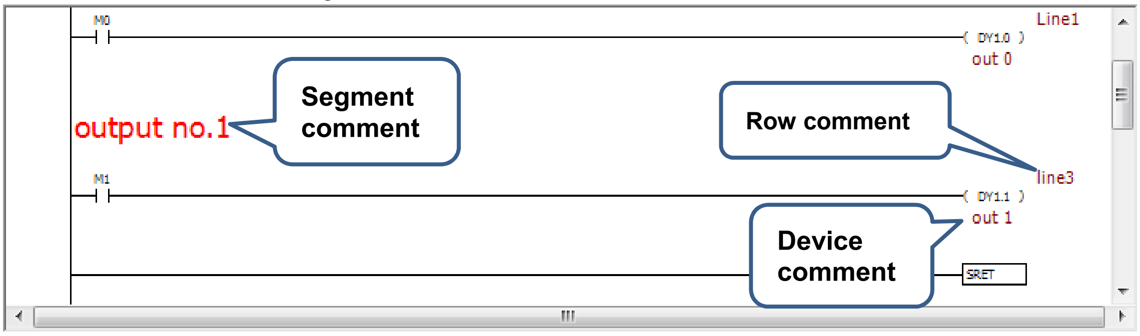
The detailed description will be elaborated as in the following paragraph.

- Replace

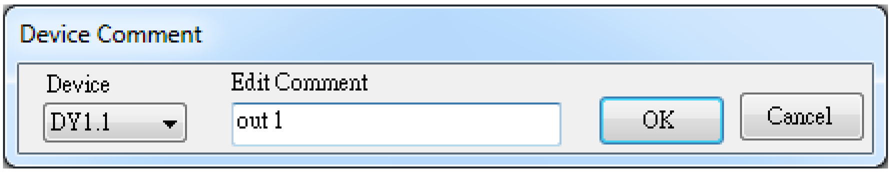


Item	Description
Find what	Input the device you are looking for.
Replace with	Input the replacing device.
All Ladders	Output the result to [Find result 1] or [Find result 2].
Keep (Find what) device comment	When coping the device comment of “Replace with” to that of “Find what”, keep the device comment of “Find what”.
Then remove replaced device comment	When coping the device comment of “Replace with” to that of “Find what”, remove the device comment of “Find what”.
Replcae options	The number of the device(s) to be replaced.

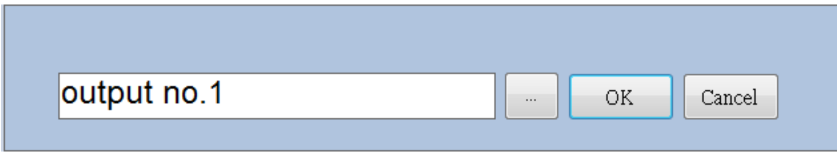
- Device comment \ Segment comment \ Row comment



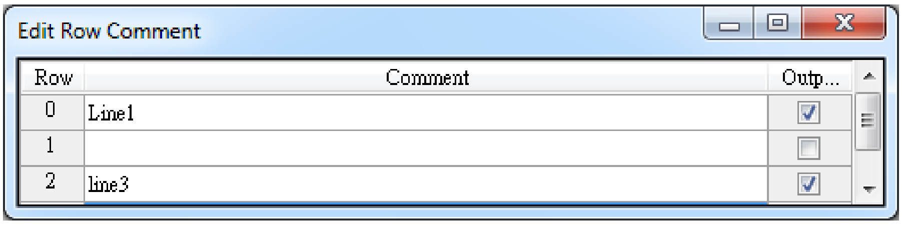
Select the device and click [Edit] > [Device Comment] to open the eiditng window.



Select a blank row and click [Edit] > [Segment comments] to open the editing window.

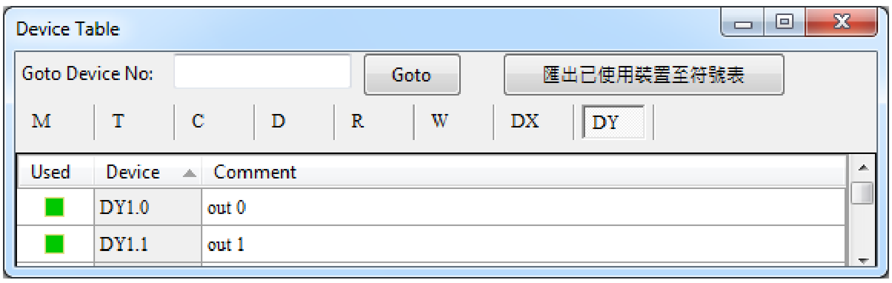


Select [Edit] > [Row Comment] to open the row editing window.

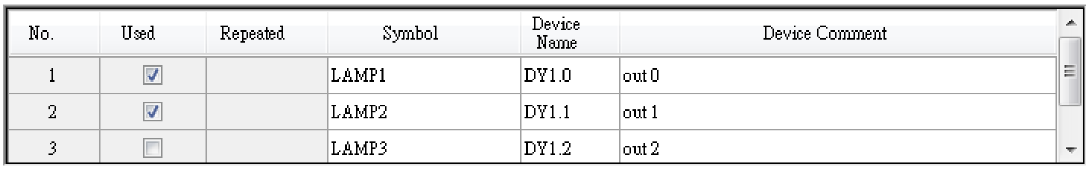


- Device Table

This table shows all devices and their comments and allows you to edit comments directly.



- Symbol Table

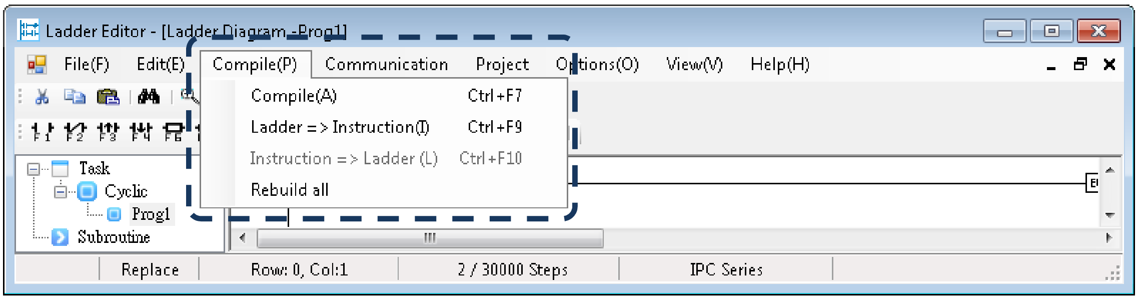


No.	Used	Repeated	Symbol	Device Name	Device Comment
1	☑		LAMP1	DY1.0	out 0
2	☑		LAMP2	DY1.1	out 1
3	☐		LAMP3	DY1.2	out 2

Further detail about each column is described as follows:

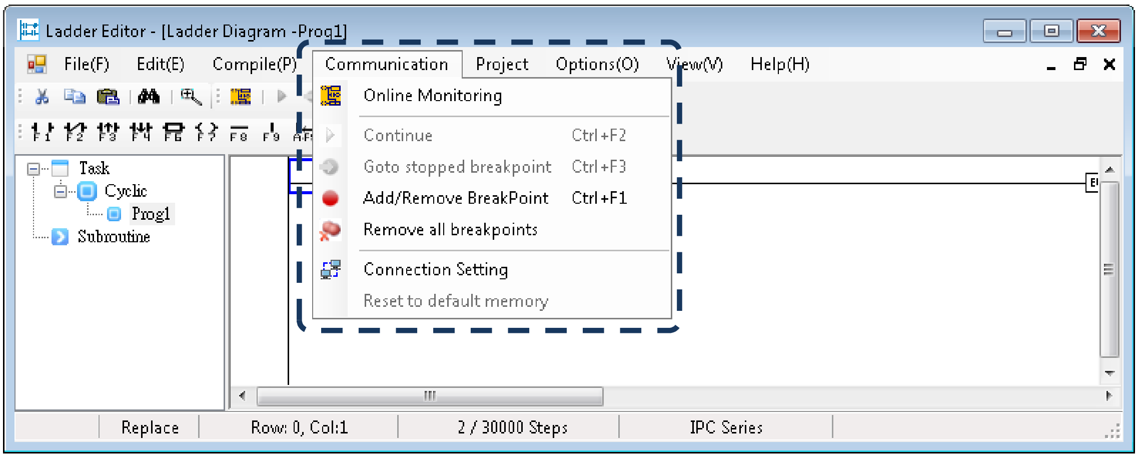
Item	Description
Used	If this item is checked, it means this device is used in the program.
Repeated	Repeated symbol: The same symbol is used by different devices. Repeated device: The same device has used more than one symbol.
Symbol	The symbol used by the devices; Device names can be replaced by symbols in the program.
Device Name	Select the device that uses symbols.

Compile



Item	Description
Compile(A)	Compile all programs.
Ladder => Instruction(I)	Compile the ladder diagrams into instructions.
Instruction => Ladder(L)	Compile the instructions into ladder diagrams.
Rebuild all	Re-compiling all programs.

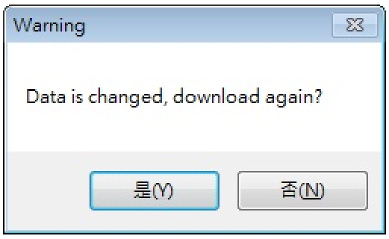
Communication



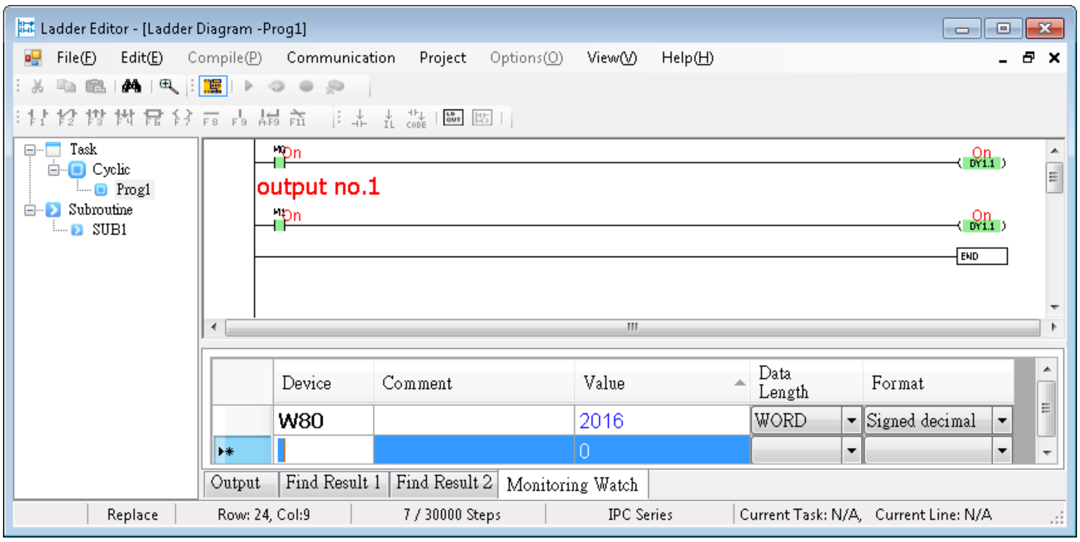
Item	Description
Online Monitoring	Online monitor the execution of IMP/Ladder programs via Ethernet.
Connection Setting	Ethernet setting for IMP connection
Reset to default memory	Reset the setting values of the device to the default.

■ Online Monitoring

Connect to the PC which has installed IMP according to the connection setting. Then, compile the ladder program before starting online monitoring. Please note that the internal program of IMP and the editing software have to be identical. If not, a warning message will pop out. See figure below.

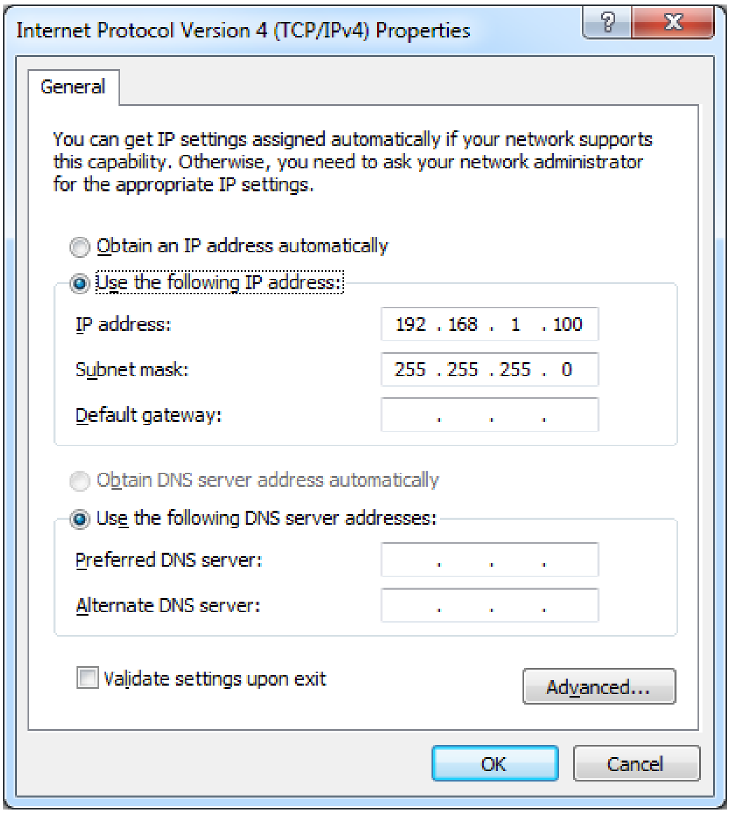


Once the connetion is successful, you can start monitoring the exeucuting status of the ladder program.

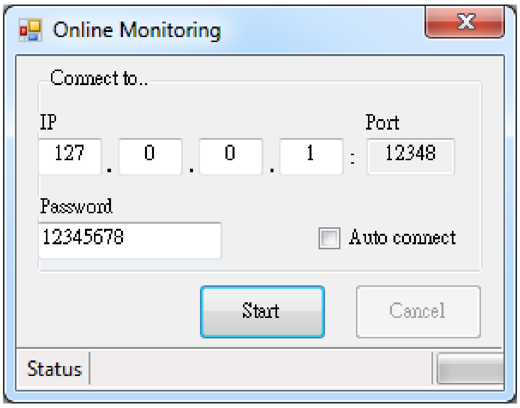


■ Connection settings

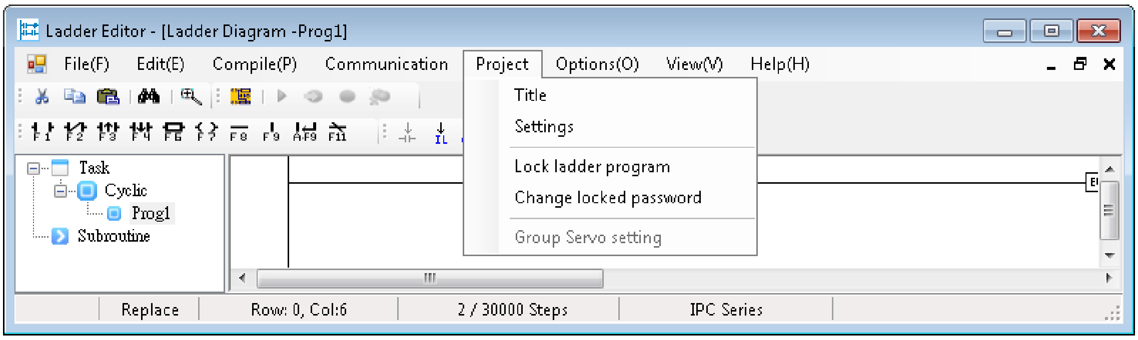
Go to [Option] > [Communication Setting] in DOPSoft. Enter the IP address of PAC or PC as shown in the figure below (If the monitoring PC is in the same network.) Please note that the PC with DOPSoft and PAC with IMP installed have to be in the same network.



Enter the IP address of PC (with installation of IMP) to be connected. And enter password (default: 12345678) and port number (default: 12348)



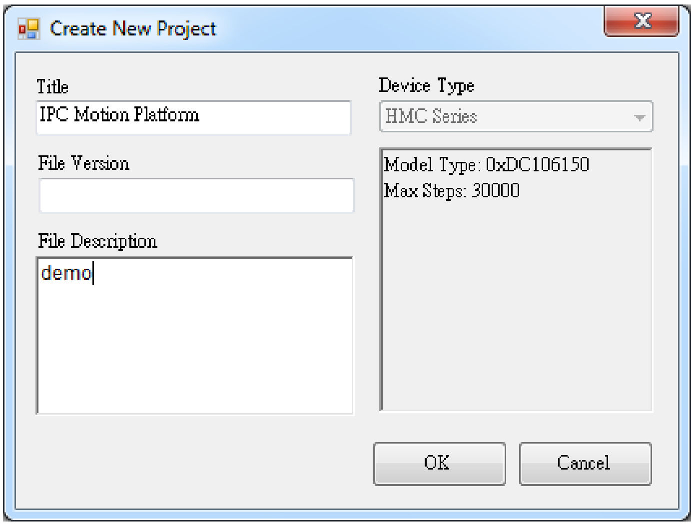
Project



Item	Description
Title	It is for setting the information such as project version.
Settings	It is for setting the auto save function, which automatically saves the ladder diagram periodically.
Lock ladder program	Lock the selected ladder diagram and it can neither be opened nor edited.
Change locked password	Change the locked password.
Group Servo setting	Apply the servo configuration.

■ Title

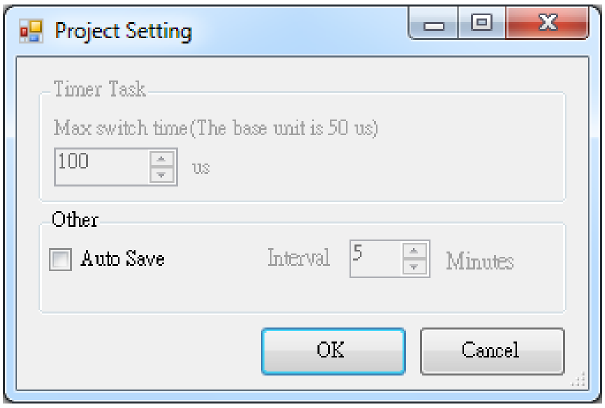
You can input the project title, file version and file description in the **Create New Project** window.



4

■ Settings

Go to [Project] > [Project Settings]. Check **Auto Save** and the ladder programs in the project will be saved automatically and periodically.



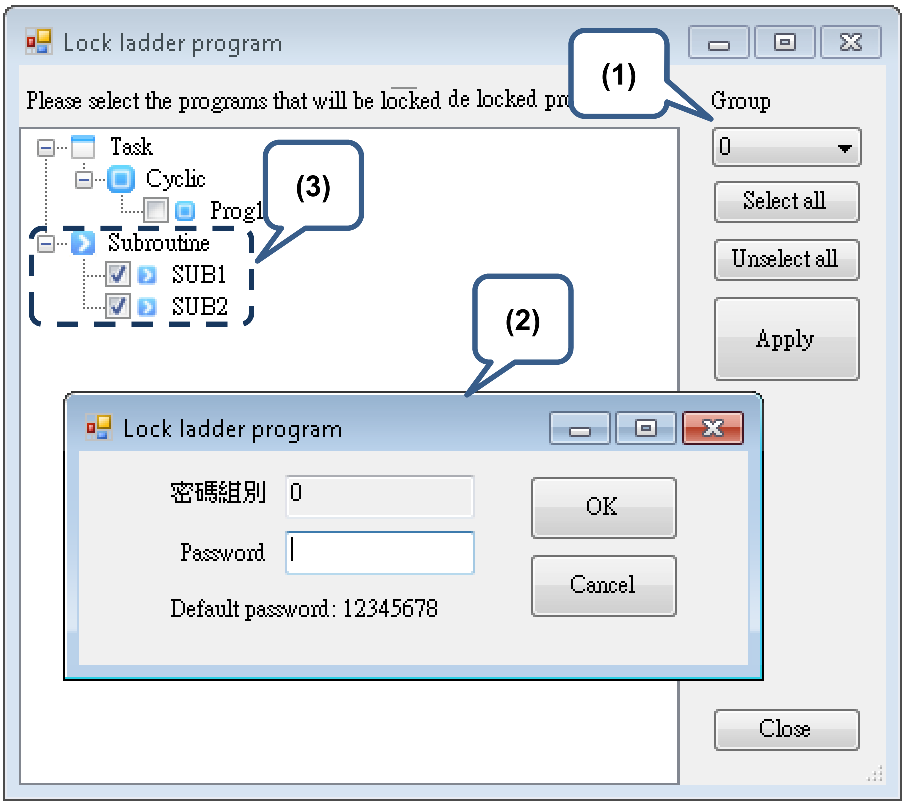
■ Lock Ladder Program

This function allows you to encypt the ladder program. Follow the steps below to complete the encryption.

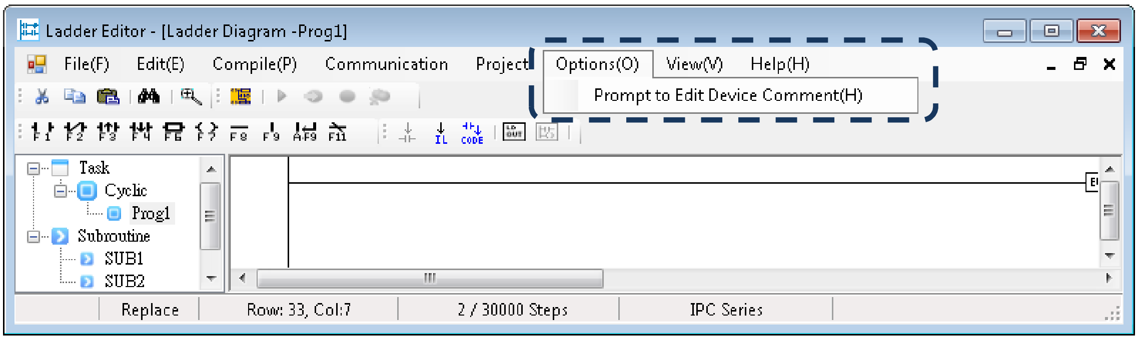
(1) Select the password group to be used.

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

(3) Select the ladder diagram to be encrypted and click **OK**. Then, you will not be able to open or edit these ladder diagrams in the editing section.

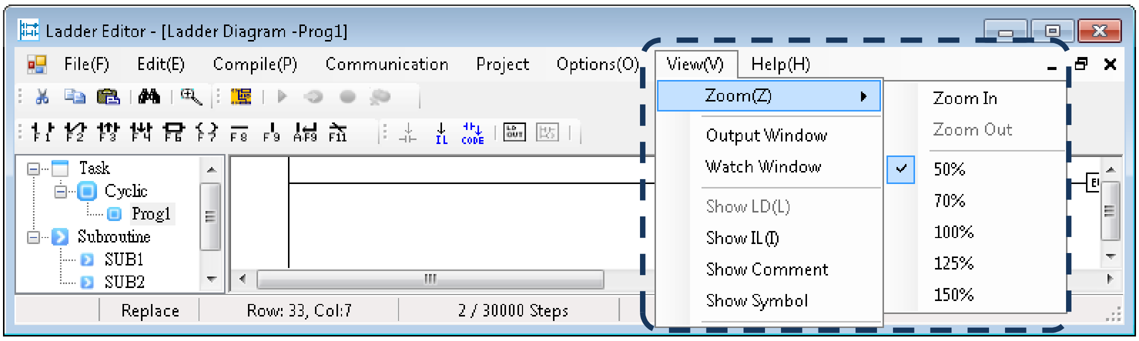


Options



Item	Description
Prompt to Edit Device Comment(H)	Auto check if the device comment exists. If there is no comment, the comment input window will pop out.

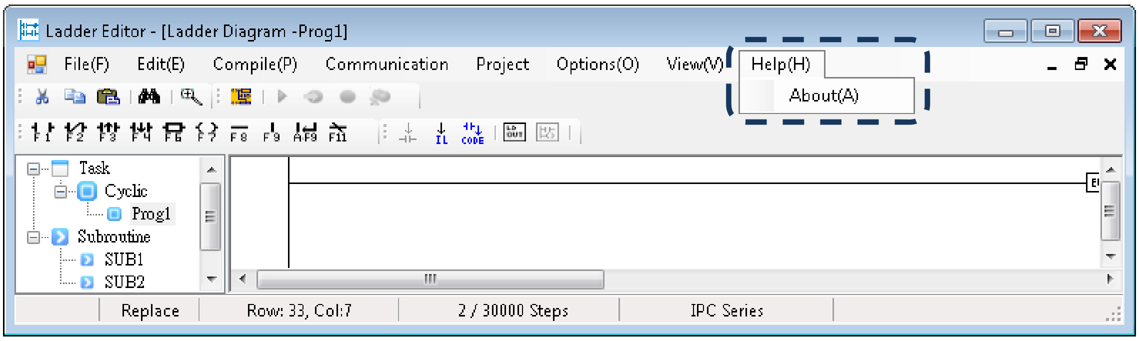
View



Item	Description
Zoom(Z)	The content in the editing window can be zoomed in and out to 50%, 70%, 100%, 125%, or 150%.
Output Window	Show the output window.
Watch Window	Show the watch window.
Show LD(L)	Show ladder diagram.
Show IL(I)	Show instruction list.
Show Comment	Show device comments and row comments.
Show Symbol	Show symbols or devices.

4

Help



Item	Description
About(A)	Show information about the verison of Ladder Editor.

Memory Device 5

When applying the relevant devices of IMP, please refer to the corresponding setting range and specifications to ensure normal operation.

5.1 Device table ········· 5-2
5.1.1 DMCNET input relay(DX) / DMCNET output relay (DY) ········· 5-3
5.1.2 Auxiliary relay (M) ········· 5-4
5.1.3 Timer (T) ········· 5-4
5.1.4 Counter (C) ········· 5-5
5.1.5 Data register (D) ········· 5-6
5.1.6 Indirect register (V) ········· 5-6
5.1.7 HMI auxiliary register ········· 5-6
5.1.8 Constant (K) / Float point (F) ········· 5-7
5.2 System special relay ········· 5-8
5.2.1 PLC special relay ········· 5-8
5.2.2 Motion status special relay ········· 5-9

5

5.1 Device table

The table below shows the corresponding range of IMP devices:

<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 Replay</td><td>DX</td><td colspan="3">DMCNET 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">DMCNET 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 rowspan="2" colspan="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">-</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-bit</td><td>0 ~ 199</td><td rowspan="2">256 points</td><td rowspan="2">-</td></tr>
<tr><td colspan="2">32-bit</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>-</td><td rowspan="2">65536 points</td><td rowspan="2">-</td></tr>
<tr><td colspan="2">For motion mode</td><td>-</td></tr>
<tr><td rowspan="10">PLC Register</td><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 ~ 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</td><td colspan="2">16-bit</td><td>0 ~ 199</td><td rowspan="2">256 points</td><td>0 ~ 65535</td></tr>
<tr><td colspan="2">32-bit</td><td>200 ~ 255</td><td>-2147483648
~2147,483,647</td></tr>
<tr><td rowspan="3">D</td><td rowspan="3">Data register</td><td rowspan="3">16-bit</td><td rowspan="2">General</td><td>0 ~ 1023</td><td rowspan="3">65536 points</td><td rowspan="3">-32768 ~ 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-bit</td><td>0 ~ 127</td><td>128 points</td><td>-32768 ~ 32767</td></tr>
<tr><td rowspan="2">W</td><td rowspan="2">Special register</td><td rowspan="2">16-bit</td><td>For PLC</td><td>-</td><td rowspan="2">65536 points</td><td rowspan="2">-32768 ~ 32767</td></tr>
<tr><td>For motion mode</td><td>-</td></tr>
<tr><td rowspan="3">HMI Register</td><td>$M</td><td colspan="3">Auxiliary register</td><td>0 ~ 1023</td><td>1024 points</td><td>-32768 ~ 32767</td></tr>
<tr><td>$</td><td colspan="3">Auxiliary register</td><td>0 ~ 65535</td><td>65536 points</td><td>-32768 ~ 32767</td></tr>
<tr><td>*$</td><td colspan="3">Pointer register</td><td>0 ~ 65535</td><td>65536 points</td><td>-32768 ~ 32767</td></tr>
<tr><td>Pointer</td><td>P</td><td colspan="3">Jump point</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>Float point</td><td>F</td><td colspan="3">Float point</td><td>-</td><td>-</td><td>-</td></tr>
</table>

The following sections will describe the device definition and their corresponding setting range.

5.1.1 DMCNET input relay (DX) / DMCNET output relay (DY)

5

DMCNET input / output relays are numbered in decimal form. DMCNET input relay (DX) and DMCENT output relay (DY) correspond to the output and input points of DMCNET RM(MN\NT\PT)[*1] or HMC-RIO3232RT5 module, respectively. The corresponding address is shown as follows:

Device	DMC-RMxx(MN\NT\PT) & HMC-RIO3232RT5			
	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 ASD-DMC-GE16MN and ASD-DMC-GE16NT modules are not supported.

DMCNET Input Relay (DX)

DMCNET input relay (DX) connects via the fieldbus and receives signals from input modules. In the program, number of contact A and B is not limited on each input relay. The ON/OFF state of DMCNET input relay (DX) is only determined by the external device.

DMCNET Output Relay (DY)

After one PLC program cycle completed, the computing result will be output to the DO module (DY). In the program, number of contact A and B is not limited.

5

5.1.2 Auxiliary relay (M)

Auxiliary relay is a shared memory of motion program macro (MPM) and PLC. Both auxiliary relay (M) and output relay (DY) have output coils and contact A and B. And number of these contacts is not limited. Users can use the auxiliary relay (M) to form a control circuit, but its computing result will not be output directly. There are two types of auxiliary relay:

Type	General	Volatile
Number range	M0 ~ M1023; M4096 ~ M65535	M1024 ~ M4095
Description	When power is cut off, the status will be OFF; when power on again, the status remains OFF.	The status remains the same before power is cut off.

5.1.3 Timer (T)

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

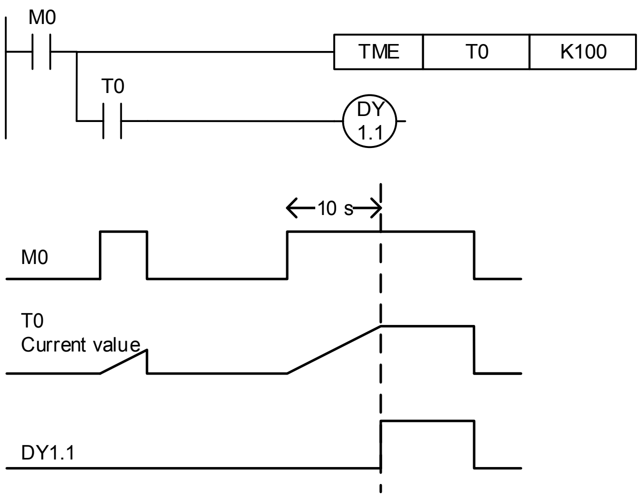
Type	100 ms / General	10 ms / General
Number range	0 ~ 199; 256 ~ 767	200 ~ 255; 768 ~ 1023

Note: Clock drift may occur in this occasion.

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

The actual setting time = Time unit x Setting value

When TMR command is executed, timer starts. Once it reaches the setting value, the output bit is ON. When TMR command is stopped, the current value resets to 0 and the output bit is OFF.



When M0 is ON, the timer T0's current time increases with 100 ms as a unit. When the current value equals the setting value K100 (10 sec), the output bit T0 is ON.
When M0 is OFF or power is cut off, timer T0 will resets to 0 and the output bit T0 becomes OFF.

5

5.1.4 Counter (C)

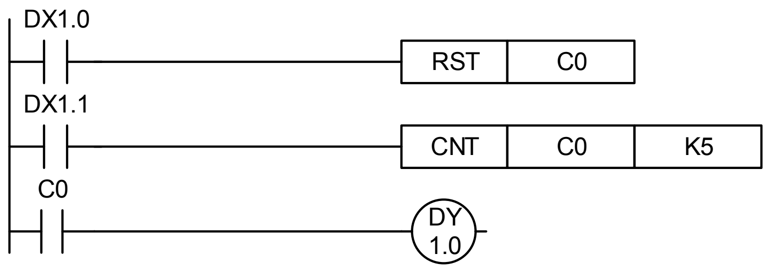
The counter is numbered in decimal form and can be categorized into two types. See the following number range.

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 setting value	
Output contact	When counter reaches the setting value, the bit is ON and being held.	
Homing	When executing RST command, the current value resets to 0 and the bit switches to OFF.	

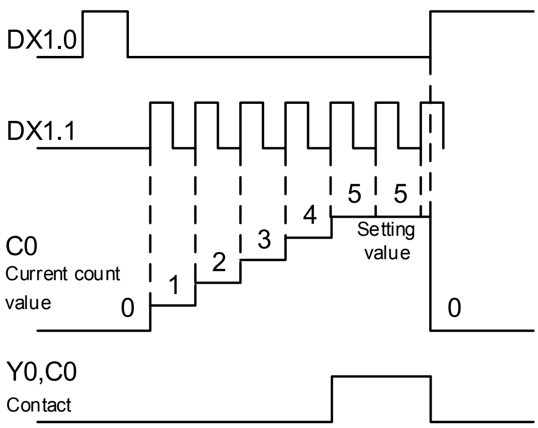
When CNT command of the counter is being executed (signal turns from OFF to ON), the count increases by 1. If the current value equals the setting value, the counter output bit is ON. If counter setting value is in DEC (start with K) and the setting value is 0 or 1, the counter output bit becomes ON at the first time CNT command is triggered.) The counter setting value can be set by constant K or by the value of register D.

Example:

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



When DX1.0 is ON and RST command is executed, count of C0 resets to 0 and counter C0 output bit is OFF. When DX1.1 switches from OFF to ON state, C0's current value increases by 1. When counter C0 reaches the setting value K5, the C0 output bit is ON and C0's current value equals the setting value (i.e. K5). Then, C0 will not receive the triggering signal of DX1.1 and current value of C0 remains K5. See the sequence diagram for example below.



5.1.5 Data register (D)

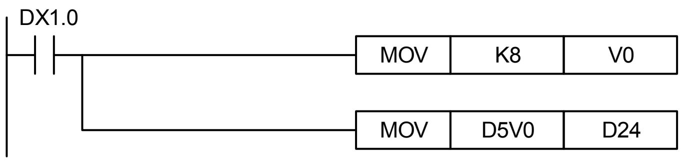
Data register number

Data register (D) is the shared memory of PLC and MPM. Its data size is 16-bit (-32,768 ~ 32,767), the highest bit is either a positive or negative sign. The register can save data of -32,768 ~ +32,767. Uses can join two 16-bit registers into one 32-bit register. (Example **D+1**,**D**. The smaller number (**D**) is for specifying Low word; High word (**D+1**) is for specifying positive or negative sign. And this register saves data of -2,147,483,648 ~ +2,147,483,647. It can be categorized into two types; the number range is shown as follows:

Type	General	Non-volatile
Number range	D0 ~ D1023, D4096 ~ D65535	D1024 ~ D4095
Description	The content will be cleared to 0 when power is cut off.	The content remains unchanged after power is on again.

5.1.6 Indirect register (V)

Indirect register (V) is a 16-bit register, which corresponding points are from V0 ~ V127, 128 points in total. Indirect and general data register are both 16-bit data registers, which allows users to read and write. For general purpose, they can only be used for 16-bit commands.



When DX1.0 is ON, V0 = 8; D5V0 = D(5+8) = D13.
Now, content of D13 will be moved to D24.

5.1.7 HMI auxiliary register

$ register is the auxiliary register special designed for HMI. Its data format is 16-bit and able to save -32,768 ~ +32,767 data. Users can only access this auxiliary register via HMI. This auxiliary relay can be used to edit HMI macro programs and elements. It can also set double word via macro command (DW). Combine the two$ registers into a 32-bit register and it can save -2,147,483,648 ~ +2,147,483,647 data. This register $M does not have the non-volatile function and its data size is 16-bit, which can save -32,768 ~ +32,767 data. $ and $M auxiliary registers can only be accessed via HMI interface.

5.1.8 Constant (K) / Float point (F)

PLC can do the computing by constant (K) and float point (F). Details are described as follows.

Constant K

Usually, a decimal value will start with a “K”. Take K100 for example, this value is 100 in decimal form.

When bit device DX, DY, or M is with the prefix K, their data format will become Nibble, Byte, Word, or Double word.

Example: K2DY1 and K4M100. K1 here represents a combination of 4 bits. K2 ~ K4 represent combinations of 8, 12, and 16 bits.

Float point F

Float point value is used as an operand.

Example: FADD F12.3 F0 D0 (F float constant)

The computing and saving of the internal values are done in binary format. See the binary values and terminology in the table below.

Bit	The value’s basic unit in binary form. It is either 1 or 0.
Nibble	It is composed by 4 consecutive bits. (e.g., bit0 ~ bit3) It can represent 0 ~ 15 in decimal form or 0 ~ F in hexadecimal form.
Byte	It is composed by two bytes, which equals 8 bits (e.g., bit0～bit7). And it also represents 00 ~ FF of hexadecimal form.
Word	It is composed by two words, which is 16- bit (e.g. bit0~bit15) And it represents 4 nibbles in Hex, 0000 ~ FFFF.
Double word	It is composed by two consecutive double words, which equal 32 bits. (e.g., bit0 ~ bit31) And it can represent 8 double words in hexadecimal, 00000000 ~ FFFFFFFF.

5.2 System special relay

While the device functions mentioned in section 5.1 are user-defined, the functions of system special relay have been set by the system. In the following sections, details about system special relay will be elaborated (including PLC and motion status type).

5.2.1 PLC special relay

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

Type	No.	Function	Description	Property	Non-volatile
Operation flag	R0	Operation flag (Contact a)	When controller is operating, this bit remains ON.	R	NO
	R1	Operation flag (Contact b)	When controller is operating, this bit remains OFF.	R	NO
	R4	Initial pulse	This bit is ON in the first PLC cycle.	R	NO
Clock pulse	R13	0.5-second clock pulse	When PLC is operating, this bit is ON for 0.5 sec and OFF for 0.5 sec. Note: Clock drift may occur in this occasion.	R	NO
	R14	1-second clock pulse	When PLC is operating, this bit is ON for 1 min and OFF for another 1 min. Note: Clock drift may occur in this occasion.	R	NO

5.2.2 Motion status special relay

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

■ Motion control special relay

Corresponding DMCNET address of each axis is shown in the following table. (You can also refer to Appendix A.)

Function	Description	Property	Axis 1	Axis 2	Axis 3	~	Axis 36
SVON control	Servo-on control	R / W	R10151	R10251	R10351	~	R13651
Software limit enabling bit	Software limit control	R / W	R10152	R10252	R10352	~	R13652
Motion curve setting	Acceleration curve switchover	R / W	R10161	R10261	R10361	~	R13661
Jog direction control	Motion direction control	R / W	R10162	R10262	R10362	~	R13662
Torque limit enabling bit in speed mode	Control bit	R / W	R10163	R10263	R10363	~	R13663
Speed limit enabling bit in torque mode	Control bit	R / W	R10164	R10264	R10364	~	R13664
Flag of motion in process	Display motion status	R / W	R10165	R10265	R10365	~	R13665

■ Motion status special relay

IMP is able to control 36 axes of servo motions via DMCNET simultaneously. The corresponding address of each axis is shown in the table. (You can also refer to Appendix A.)

Function	Description	Property	Axis 1	Axis 2	Axis 3	~	Axis 36
Servo operation mode	Display current motion. Please refer to Servo operation mode table.	Read Only	R10100	R10200	R10300	~	R13600
			R10101	R10201	R10301	~	R13601
			R10102	R10202	R10302	~	R13602
			R10103	R10203	R10303	~	R13603
DI3 status (Servo)	DI3 (SLD) state mapping		R10104	R10204	R10304	~	R13604
Servo alarm flag	Alarm message		R10105	R10205	R10305	~	R13605
SVON monitoring flag	Display motor excitement state.		R10108	R10208	R10308	~	R13608
Servo error flag	Display servo error.		R10109	R10209	R10309	~	R13609
Positioning complete flag	Motion command is completed.		R10110	R10210	R10310	~	R13610
Servo control mode	Mode Specific, which displays servo status		R10112	R10212	R10312	~	R13612
			R10113	R10213	R10313	~	R13613
Flag for triggering servo limit	Display that it has reached the positive limit.		R10114	R10214	R10314	~	R13614
	Display that it has reached the negative limit.		R10115	R10215	R10315	~	R13615
Software limit control bit	Display that it has exceeded the software positive limit.		R10116	R10216	R10316	~	R13616
	Display that it has exceeded the software negative limit.		R10117	R10217	R10317	~	R13617

5

Servo operation mode table:

bit	0	1	2	3	Mode	Remarks
ON / OFF	ON	OFF	OFF	OFF	Position mode	In these modes, after the motion card issues the command, the servo drive will then execute the command, such as motion in PR mode, motion with fixed speed and fixed torque.
ON / OFF	ON	ON	OFF	OFF	Speed mode	
ON / OFF	OFF	OFF	ON	OFF	Torque mode	
ON / OFF	OFF	ON	ON	OFF	Homing mode	
ON / OFF	ON	ON	ON	ON	DMCNET mode	DMCNET mode: the motion card updates servo motor's target position every 1 ms. And servo motor continuously update the new target position.

Description about Mode Specific bit status:

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

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

The DMCNET corresponding address for each axis is as follows:

Function	Description	Property	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 is being executed 3: Command of change speed during motion, but the single axis is not in motion. 4: Command of change position during motion, but the single axis is not in motion. 5: Position has exceeded the software limit. 6: Reaching the software limit during motion. 20: The following master axis is not using the same fieldbus. 21: The following master axis does not exist.	R	W10150	W10250	…	W13650

Function	Description	Property	Axis 1	Axis 2	…	Axis 36
Single motion control code	The command code of 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: Change speed during motion 7: Change position during motion 8: Coordinates setting 9: Speed control 10: Torque control 13: Servo alarm reset 14: Emergency stop 15: Decelerates 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) State display code: 50: Interpolation in operation 51: Group awaits 55: MCM in operation 60: MPG in operation 70: SNC in operation	R / W	W10151	W10251	…	W13651
Acceleration time	Unit: µs	R / W	W10152	W10252	…	W13652
	-	R / W	W10153	W10253	…	W13653
Deceleration time	Unit: µs	R / W	W10154	W10254	…	W13654
	-	R / W	W10155	W10255	…	W13655
Target speed of motion command	Unit: refer to the setting of motion speed unit	R / W	W10156	W10256	…	W13656
	-	R / W	W10157	W10257	…	W13657
Target coordinates of motion command	Unit: µm	R / W	W10158	W10258	…	W13658
	-	R / W	W10159	W10259	…	W13659
Homing mode setting	Please refer to Appendix B	R / W	W10160	W10260	…	W13660
Setting of motion speed unit	Setting of motion speed unit: 0: PUU/sec 1: % 2: mm/min	R / W	W10161	W10261	…	W13661
1st speed in homing mode	Unit: rpm	R / W	W10162	W10262	…	W13662
	-	R / W	W10163	W10263	…	W13663
2nd speed in homing mode	Unit: rpm	R / W	W10164	W10264	…	W13664
	-	R / W	W10165	W10265	…	W13665
Offset in homing mode	Unit: µm	R / W	W10166	W10266	…	W13666
	-	R / W	W10167	W10267	…	W13667
Total index number of rotary table	-	R / W	W10168	W10268	…	W13668
Target position of rotary table	-	R / W	W10169	W10269	…	W13669
Target speed in speed	Unit: rpm	R / W	W10170	W10270	…	W13670

5

Function	Description	Property	Axis 1	Axis 2	…	Axis 36
mode	-	R / W	W10171	W10271	…	W13671
Target torque in torque mode	Unit: ‰	R / W	W10172	W10272	…	W13672
	-	R / W	W10173	W10273	…	W13673
Torque limit in speed mode	Unit: ‰	R / W	W10174	W10274	…	W13674
	-	R / W	W10175	W10275	…	W13675
Speed limit in torque mode	Unit: rpm	R / W	W10176	W10276	…	W13676
	-	R / W	W10177	W10277	…	W13677
Written value of the servo	-	R / W	W10180	W10280	…	W13680
	-	R / W	W10181	W10281	…	W13681
Forward software limit	Unit: μm	R / W	W10182	W10282	…	W13682
	-	R / W	W10183	W10283	…	W13683
Reverse software limit	Unit: μm	R / W	W10184	W10284	…	W13684
	-	R / W	W10185	W10285	…	W13685
DMC_Servo parameter group & index value	Servo parameter group x 1000 + Servo parameter No.	R / W	W10186	W10286	…	W13686
Read/Write servo parameter control code	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	W10187	W10287	…	W13687

5

■ **Special register of Single-axis motion status**

DMCNET corresponding address is shown as follows:

Function	Description	Property	Axis 1	Axis 2	…	Axis 36
Motor feedback position	Unit: μm	Read Only	W10102	W10202	…	W13602
	-		W10103	W10203	…	W13603
Command position	Unit: μm		W10104	W10204	…	W13604
	-		W10105	W10205	…	W13605
Target position	Unit: μm		W10106	W10206	…	W13606
	-		W10107	W10207	…	W13607
Servo DI status	-		W10108	W10208	…	W13608
Servo DO status	-		W10109	W10209	…	W13609
Current motion speed of each axis	Unit: μm		W10110	W10210	…	W13610
	-		W10111	W10211	…	W13611
Current output torque of motor	Unit: ‰		W10113	W10213	…	W13613
Command status	Command status code: 0: Not completed 1: completed		W10114	W10214	…	W13614
Servo error code	Display servo error code		W10115	W10215	…	W13615
Read servo return value	-		W10116	W10216	…	W13616
	-		W10117	W10217	…	W13617
Set servo monitoring parameters	-		W10118	W10218	…	W13618
Current motor speed	Unit: rpm		W10119	W10219	…	W13619
	-		W10120	W10220	…	W13620
Servo parameter reading/writing error	Read/Write error code of servo parameter: 1: Fail to read servo parameters. 2: Fail to write servo parameters. 3: Fail to read servo commands. 4: Fail to set servo monitoring commands		W10121	W10221	…	W13621

5

SVON

Definition

When servo control flag (SVON) is set to ON, it means the servo is ON; and when this flag is set to OFF, it means the servo is OFF. Take Axis 1 for example, if setting the relay (R10151) to ON and the servo axis is in Servo ON state, the corresponding flag R10108 will display “ON”. While this axis is in Servo OFF state, the corresponding monitoring flag R10108 will display “OFF”.

Error occurrence

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

1. DMCNET connection 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 of servo axis is to limit the mechanism motion. When the **bit for activating software limit** is ON, the servo motion will be limited to the range specified by the positive and negative values. Take Axis 1 for example, if positive limit (W10182, W10183) is 100 and negative limit (W10184, W10185) is -100, and when this bit is ON, the axis will stop and trigger the software limit flag once the servo motion exceeds the range.

Note:
1. When the software limit exceeds the range, the stop command will be triggered. And the stopping position might 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
Bit for activating software limit	R / W	R10152	R10251	R10351	R10451	R10551	R10651	R10751	…	R13651
Bit for controlling software limit (Positive)	R	R10116	R10208	R10308	R10408	R10508	R10608	R10708	…	R13608
Bit for controlling software limit (Negative)	R	R10117	R10217	R10317	R10417	R10517	R10617	R10717	…	R13617
Software forward limit	R / W	W10182	W10282	W10382	W10482	W10582	W10682	W10782	…	W13682
	R / W	W10183	W10283	W10383	W10483	W10583	W10683	W10783	…	W13683
Software reverse 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, servo motor runs with the 1st speed (HSP1).But once reaching the reference origin, the motor will run with the 2nd speed (HSP2) to carry on.

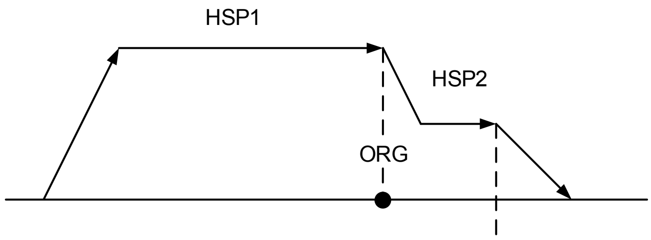


Figure 5.2.2.1 Homing speed switchover

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
1st speed in homing mode	R / W	W10162	W10262	W10362	W10462	W10562	W10662	W10762	…	W13662
	R / W	W10163	W10263	W10363	W10463	W10563	W10663	W10763	…	W13663
2nd 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 can be specified for each axis respectively.

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

It is to set the acceleration and deceleration time during motion. The acceleration time is defined by the time of the highest speed set via the Quick Start interface. In addition, the deceleration time is the time between the highest speed and motor stop set via the Quick Start interface.

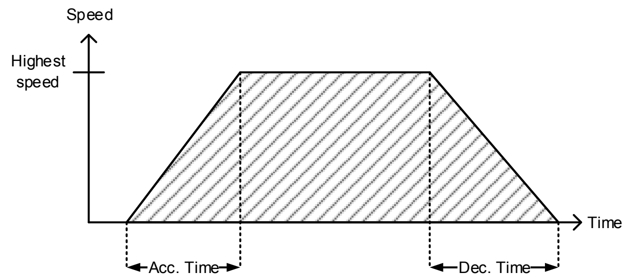


Figure 5.2.2.2 Acceleration / Deceleration time setting

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

(This page is intentionally left blank.)

Logic Editing 6

Before editing the PLC instruction, you can find detail descriptions of the instructions in this chapter.

6.1 PLC Instructions ··· 6-2
6.1.1 Instruction list ··· 6-2
6.1.2 Basic Instruction ··· 6-7
6.1.3 Application Instruction ··· 6-19
6.1.4 Single-axis motion instruction ··· 6-54
6.1.5 Interpolation Motion Instruction ··· 6-74
6.1.6 Motion Program Macro (MPM) Control Instruction ··· 6-92
6.1.7 Example ··· 6-96

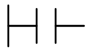
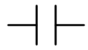
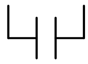
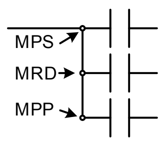
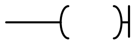
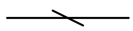
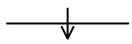
6

6.1 PLC Instructions

6.1.1 Instruction list

Followings are the instruction provided by IMP controller.

- Basic Instruction

Basic Instruction		
Type	Funciton Name	Symbol
Contact Instruction	LD	
	LDI	
	AND	
	ANI	
	OR	
	ORI	
Combined instruction	MPS	
	MRD	
	MPP	
Output Instruction	OUT	
	SET	
	RST	
Timer	TMR	
Counter	CNT	
Cyclic Task Ends	END	
Sub Program 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.	Function Name		Function	Step No.
		16-bit	32-bit		
Data compare	001	LD※	DLD※	Contact type compare	5
	002	AND※	DAND※	Contact type compare	5
	003	OR※	DOR※	Contact type compare	5
Data transmission and comparion	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 subroutines	2
	011	FOR	-	Nest loops start	3
	012	NEXT	-	Nest loops end	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
Logic 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	Foating 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 process	032	ZRST	-	Zone reset	4
	033	DECO	-	Decoder	11
	034	ENCO	-	Encoder	11
	035	BON	DBON	Monitor specified bit status	5

6

Application Instruction					
Type	NO.	Function Name		Function	Step No.
		16-bit	32-bit		
	036	ALT	-	ON/OFF alternate	2
	037	AO	-	Analog output	5
	038	AI	-	Analog input	5

- List of motion instructions

Motion Instruction						
Type	NO.	Function Name		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	-	COODR	Coordinates setting	-	7
	057	-	HOME	Homing	-	11
	058	ALRM	-	Alarm clearance	3	-
	059	ESTP	-	Emergency stop	3	-
	060	SDSTP	-	Decelerate to stop	3	-
	061	-	AXRPM	Access motor’s current speed	-	7
	062	AXTQR	-	Access motor’s current torque	5	-
	063	RSVR	-	Access servo parameter	-	5
	064	WSVP	-	Write-in servo parameter	-	7
	065	SVSTS	-	Access DO status	5	-
	066	SVITS	-	Access DI status	5	-
	067	RCBL	-	Access the buffer usage	5	-
	068	-	RPOS	Access the actual position of motor	-	7
	069	-	LPOS	Access the axial instruction position	-	7
	070	-	TPOS	Access the axial taret position	-	7
	071	MOTS	-	Access the status of motion instruction	5	-
	072	ALE	-	Access servo drive error code(s)	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

Motion Instruction						
Type	NO.	Function Name		Function	Step No.	
		16-bit	32-bit		16	32
	078	-	SPD	Speed control	-	7
	079	-	TRQ	Torque control	-	7
	080	-	RSPD	Access the current speed	-	7
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’s circle center	-	11
	109	-	ENDXY	End point of arc	-	11
	110	-	MOVP	Target position setting of each axis	-	13
	111	-	MOVLA	Linear motion (absolute)	-	7
	112	-	MOVLR	Linear motion (relative)	-	7
	113	-	CIRCAA	Arc motion (with the known absolute coordinate of circle center and angle)	-	7
	114	-	CIRCAR	Arc motion (with the known relative coordinate of circle center and angle)	-	7
	115	-	CIREAA	Arc motion (with the known absolute coordinate of arc end point and angle)	-	7
	116	-	CIREAR	Arc motion (with the known relative coordinate of end point and angle)	-	7
	117	-	CIRCEA	Arc motion (with the known absolute coordinate of arc’s circle center and end point)	-	7
	118	-	CIRCER	Arc motion (with the known relative coordinate of arc’s circle center and end point)	-	7
	119	-	HELIXA	Helical motion (absolute)	-	7
	120	-	HELIXR	Helical motion (relative)	-	7
	121	GESTP	-	Emergency stop for group	3	-
	122	GSDSTP	-	Decelerate to stop for group	3	-
Motion program macro (MPM) instruction	150	MPMST	-	MPM starts	3	-
	151	MPMSTP	-	MPM stops	3	-
	152	MPMPAU	-	MPM pauses	3	-
	153	MPMSPD	-	MPM feedrate overwrite	5	-
	154	MPMER	-	Access MPM error code	5	-

6

Motion Instruction						
Type	NO.	Function Name		Function	Step No.	
		16-bit	32-bit		16	32
	155	MSTEP	-	Access the step No. of MPM	7	-

6.1.2 Basic Instruction

Descriptions

(1) (2) (3)

NO.	D	Instruction	P / F	S1	Function

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	●	●	●	●	●	●	-	-	-	-	-	-	● [V]	-	-	-	-	-

(4)

Notes when applying operand:

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

(1) D: It supports 32-bit instructions; (2) P: It supports rising-edge triggered type / F: It supports falling-edge triggered type; (3) Operand; (4) Index register

■ LD

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

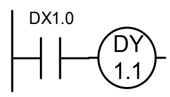
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

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

Description: The LD instruction applies to the starting A contact of a left bus bar or a starting A contact in loop block. It saves the current value and stores the acquired S1 contact status in a cumulative register.

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0's A contact
OUT	DY1.1	Output to DY1.1

■ LDI

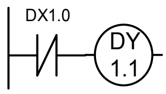
NO.	-	LDI	-	S1	Load B contact
-			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

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

Description: The LDI instruction applies to the starting B contact of a left bus bar or a starting B contact in the loop block. It saves the current value and stores the acquired S1 contact status in a cumulative register.

Example: Ladder diagram:



Instruction code		Description
LDI	DX1.0	Load to DX1.0's B contact
OUT	DY1.1	Output to DY1.1

■ AND

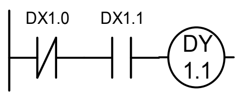
NO.	-	AND	P	S1	Serial connect A contact
-			F		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

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

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

Example: Ladder diagram:



Instruction code		Description
LDI	DX1.0	Load to DX1.0's B contact
AND	DY1.1	Serial connect to DX1.1's A contact
OUT	DY1.1	Output to DY1.1

■ ANI

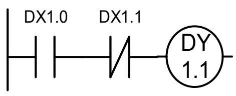
NO.	-	ANI	-	S1	Serial connect B contact
-			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

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

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

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0’s A contact
ANI	DX1.1	Serial connect to DX1.1’s B contact
OUT	DY1.1	Output to DY1.1

■ OR

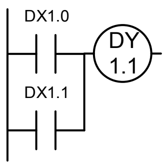
NO.	-	OR	P	S1	Parallel connect A contact
-			F		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

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

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

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0’s A contact
OR	DX1.1	Parallel connect to DX1.1’s A contact
OUT	DY1.1	Output to DY1.1

■ ORI

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

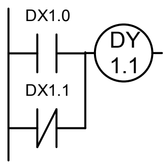
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

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

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

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0's A contact
ORI	DX1.1	Parallel connect to DX1.1's B contact
OUT	DY1.1	Output to DY1.1

■ ANB

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

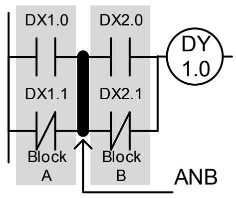
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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 a cumulative register.

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0's A contact
ORI	DX1.1	Parallel connect to DX1.1's B contact
LD	DX2.0	Load to DX2.0's A contact
ORI	DX2.1	Parallel connect to DX1.1's B contact
ANB		Serial connect two blocks
OUT	DY1.0	Output to DY1.0

■ ORB

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

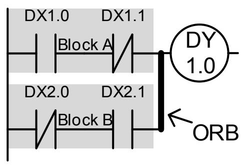
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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 the current value in a cumulative register.

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0's A contact
ANI	DX1.1	Serial connect to DX1.1's B contact
LD	DX2.0	Load to DX2.0's B contact
AND	DX2.1	Parallel connect to DX2.1's A contact
ORB		Parallel connect two blocks
OUT	DY1.0	Output to DY1.0

■ MPS

NO.	-	MPS	-	No operand	Saves it in stack
-			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word

Notes when applying operand:

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

Description: It saves the current value contained in the cumulative register in a stack. (Stack index increases by 1)

■ MRD

NO.	-	MRD	-	No operand	Read stack (Stack index remain intact)
-			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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

Description: It saves the logic operation result that previously saved in a stack in cumulative register. (Stack index remains intact)

■ MPP

NO.	-	MPP	-	No operand	Read stack
-			-		

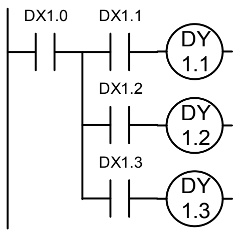
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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

Description: It retrieves the last saved logic operation result and saves it in a cumulative register. (Stack index decreases by 1)

Note: MPS instruction has to be worked with MPP. Otherwise, error occurs.

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0's A contact
MPS		Saves it in stack
AND	DX1.1	Serial connect to DX1.1's A contact
OUT	DY1.1	Output to DY1.1
MRD		Read stack (Stack index remains intact)
AND	DX1.2	Serial connect to DX1.2's A contact
OUT	DY1.2	Output to DY1.2
MPP		Read stack
AND	DX1.3	Serial connect to DX1.3's A contact
OUT	DY1.3	Output to DY1.3

■ OUT

NO.	-	OUT	-	D	Output
-			-		

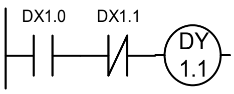
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
D	-	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-

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 to DX1.0’s A contact
ANI	DX1.1	Serial connect to DX1.1’s B contact
OUT	DY1.1	Output to DY1.1

■ SET

NO.	-	SET	-	D	Fix actions (ON)
-			-		

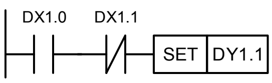
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
D	-	●	●	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

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

Description: When SET instruction is executed, the specified bit will be on. The RST instruction can be used to set this bit to off. If the SET instruction is not executed, status of the specified bit remains the same.

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0’s A contact
ANI	DX1.1	Serial connect to DX1.1’s B contact
OUT	DY1.1	DY1.1 bit on

RST

NO.	-	RST	-	D	Bit off
-			-		

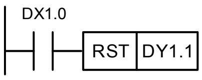
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
D	-	●	●	●	●	●	-	-	-	-	●	●	-	-	-	-	-	-

Notes when applying operand:

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

Description: When RST instruction is executed, the specified bit will be off. If the specified device is word, then it will be cleared to 0. If the RST instruction is not executed, status of the specified device remains the same.

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0's A contact
RST	DY1.1	DY1.1 bit off

TMR

NO.	-	TMR	-	S1, S2	Timer
-			-		

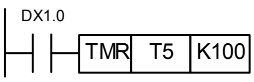
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

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

Description: After a TMR instruction is executed, the specified timer switches to On and starts timing. When setup time is reached (present value ≧ setup value):
NO (Normally Open) contact: closes;
NC (Normally Close) contact: opens.

Example: Ladder diagram:



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

■ CNT

NO.	D	CNT	-	S1, S2	Counter
-			-		

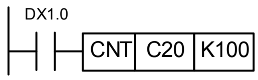
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
When S1 specifies C200~255 as the counter, then it should apply 32-bit DCNET instruction.

	Instruction	Step No.
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 setup count is reached (present value ≧ setup value):
NO (Normally Open) contact: closes;
NC (Normally Close) contact: opens.
When the count settings of S2 is reached, the counter's contacts and counting values remain intact even when more counting pulse inputs are received. An RST instruction is required to restart counting or clear the value.

Example: Ladder diagram:



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

■ END

NO.	-	END	-	No operand	Cyclic task ends
-			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

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

Description: The cyclic task's last instruction has to be an END instruction. PLC scans from address 0 to END instruction. Then, return to address 0 to scan again. When applying PLC Ladder Edit, END instruction will be added automatically.

■ SRET

NO.	-	SRET	-	No operand	Sub program ends
-			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

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

Description: The last instruction in PLC subroutine has to be the SRET instruction. In subroutine, PLC will scan from address 0 to SRET instruction. Then, subroutine ends and the cyclic task continue execution.

■ INV

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

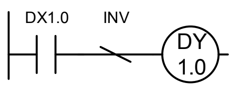
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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 saves it in a cumulative register.

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0's A contact
INV		Inverse the operation result
OUT	DY1.0	Output to DY1.0

6

■ NP

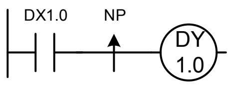
NO.	-	NP	-	No operand	Rising edge triggered
-			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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 which is before NP instruction, then store it in accumulative register.

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0's A contact
NP		Rising edge triggered
OUT	DY1.0	Output to DY1.0

■ PN

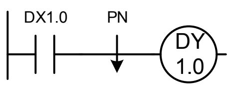
NO.	-	PN	-	No operand	Falling edge triggered
-			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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 which is before PN instruction, then store it in the accumulative register.

Example: Ladder diagram:



Instruction code		Description
LD	DX1.0	Load to DX1.0's A contact
PN		Falling edge triggered
OUT	DY1.0	Output to DY1.0

■ NOP

NO.	-	NOP	-	No operand	No action
-			-		

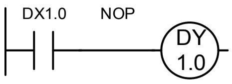
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

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

Description: The NOP instruction does not compute at all. After its execution, the logic operation result remains. If users desire to delete a statement in a program and keep the program size intact, then it can be replaced with an NOP instruction.

Example: Ladder diagram:



#NOP will not be in ladder diagram.

Instruction code		Description
LD	DX1.0	Load to DX1.0's A contact
NOP		No action
OUT	DY1.0	Output to DY1.0

6.1.3 Application Instruction

■ LD※

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

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-

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

	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 values stored in S1 and S2. When the operation result satisfies the condition, the contact turns on otherwise it does not turn on.
The LD※ instruction may connect to a bus bar directily.

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

Note: It has to use the 32-bit instruction (DLD※) to compare the 32-bit counter (C200~C255).

■ AND※

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

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-

Notes when applying operand:

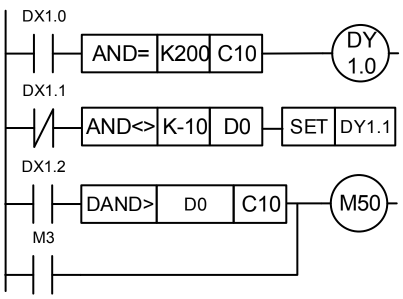
	Instruction	Step No.
32-bit	DAND※	5 Step
16-bit	AND※	5 Step

Description: S1: Data source device 1; S2: Data source device 2.
This instruction compares values stored in S1 and S2. When the comparing result satisfies the condition, the contact turns on.
The AND※ is a compare instruction serial connects to a contact.

16-bit instruction	32-bit instruction	Turn-on condition	Not 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

Note: It has to use the 32-bit instruction (DAND※) to compare the 32-bit counter (C200~C255).

Example: When DX1.0 = On and the 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 status remains.
When DX1.2 = On and value of 32-bit register D0 (D11) is less than 999,999 or M3 = On, then M50 = On.



■ OR※

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

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	-	-	● [V]	-	-

Notes when applying operand:

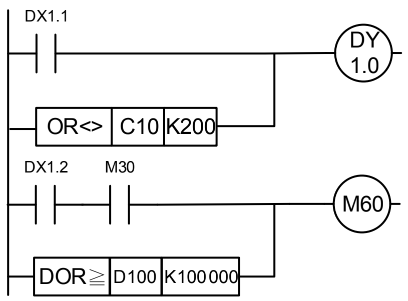
	Instruction	Step No.
32-bit	DOR※	5 Step
16-bit	OR※	5 Step

Description: S1: Data source device 1; S2: Data source device 2.
When the comparing result satisfies the condition, the contact turns on.
The OR※ is a compare instruction parallel 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

Note: It has to use the 32-bit instruction (DOR※) to compare the 32-bit counter (C200 ~ C255).

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 32-bit register D100 (D101) is greater or equals K100,000, then M60 = On.



6

■ MOV

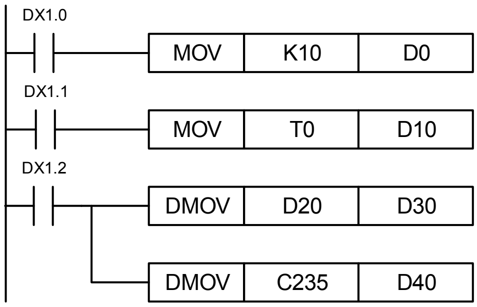
NO.	D	MOV	-	S1、D	Move data
004			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
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: Destination of data to be moved to.
When executing this instruction, data in S1 will be moved to D. If not, data in D remains the same.
When applying 32-bit instruction (e.g. application instruction MUL), users need to use the DMOV instruction to move the operation result and the current value of 32-bit high-speed counter.

Example: Users need to move 16-bit data with the MOV instruction.
When DX1.0 = Off, contents of D10 remain intact; if DX1.0 = On, K10 will be sent to register D0.
When DX1.1 = Off, contents of D10 remain intact; if DX1.1 = On, T0's current value will be sent to register D10.
When DX1.2 = Off, content of (D31, D30) and (D41, D40) remain intact; If DX1.2 = On, the current value of (D21, D20) will be sent to register (D31, D30) and the value of C235 will be sent to register (D41, D40).



6

■ BMOV

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

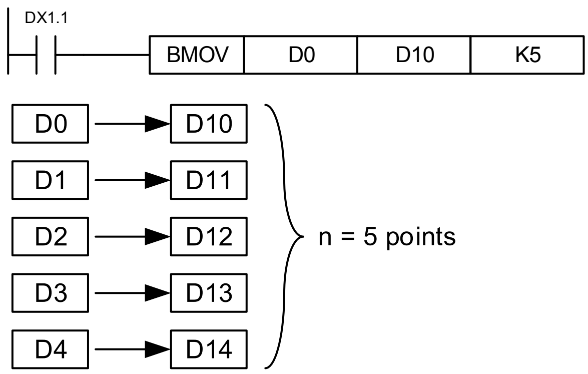
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-		-	● [V]	-	-	● [V]	-	-
n	-	-	-	-	-	-	-	-	-	●	-	-	● [V]	-	-	● [V]	-	-

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. Content of the n^{th} register starting from the device specified in S is sent to the one specified in D.

Example: When DX1.1 = On, content of D0 ~ D4 will be sent to the five registers starting from D10, respectively.



■ FMOV

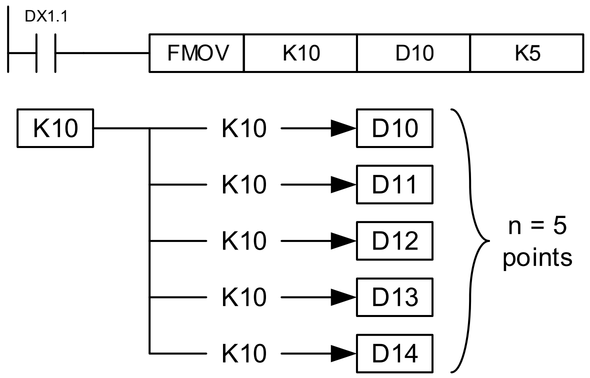
NO.	-	FMOV	-	S1、D、n	Multi move data
006			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
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 the n^{th} register starting from the device specified in S is sent to the one specified in D. However, if the point number specified by n exceeds the used range, only the one within the range will be sent.

Example: When DX1.1 = On, content of K10 will be sent to the five consecutive registers starting from D10.



6

■ ROR

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

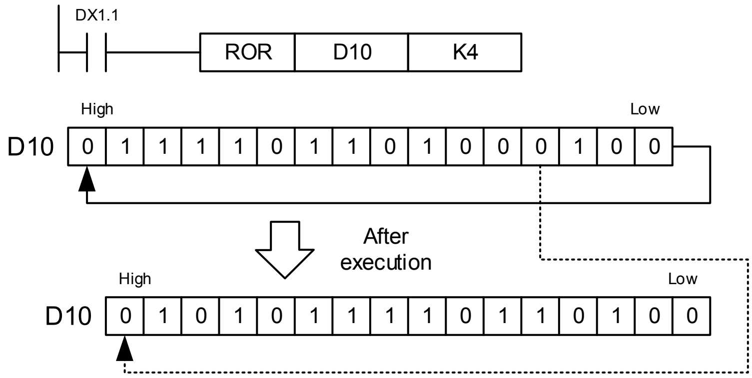
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-
n	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-

Notes when applying operand:
Range: n = K1 ~ K16 (16-bit);
n = K1 ~ K32 (32-bit)

	Instruction	Step No.
32-bit	DROR	3 Step
16-bit	ROR	3 Step

Description: D: Device to be rotated; n: Number of bits to be rotated for a time.
Right rotate n bits of digit contained in the device specified by D for one time.

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



6

■ ROL

NO.	D	ROL	-	D、n	Rotate left
008			-		

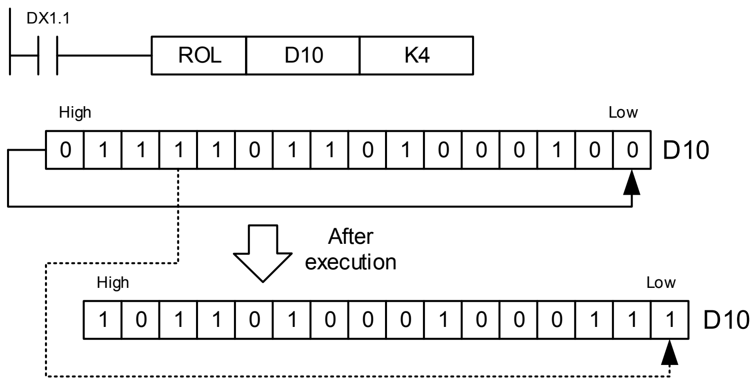
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-
n	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-

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 to be rotated; n: Number of bits to be rotated for a time.
Left rotate n bits of digit contained in the device specified by D for one time.

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



■ CJ

NO.	-	CJ	-	S	Jump
009			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes on the use of operands:
The S operand can assign index P0 ~ P255

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

Description: S: Target to be jumped to (conditional)

Users can use the CJ instruction to skip a section of statements in PLC program to reduce scan time. The CJ instruction can also be applied when two OUT instructions specify the same device.

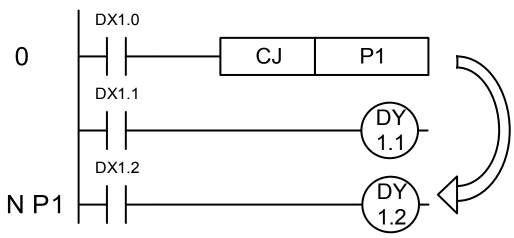
If the program specified by pointer P appears before the CJ instruction, PLC might not be able to complete the scan. However, the CJ instruction can repeatedly specify same pointer P.

Device actions when executing the jump instruction:

1. Status of device Y, M and S remain intact before executing the jump instruction.
2. The 10 ms and 100 ms of timer stops timing.
3. Timer T192 ~ T199 used by subroutine keeps on timing and the output contact functions normally.
4. Counter stops counting.
5. If the clear instruction of timer is executed before jumping, then the device will be in clear status when executing the jump instruction.
6. Application instruction will not be executed.

Example: When DX1.0 = On, the program jumps from address 0 to N (the specified label P1) for execution and ignore all statements in between.

When DX1.0 = Off, the program executes from address 0 downward in sequence as the common ones and ignores the CJ instruction.



6

■ CALL

NO.	-	CALL	-	S	Call subrutine													
010			-															
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes on the use of operands: The S operand is the name of subroutine in Ladder Editor.		Instruction	Step No.
	32-bit	-	-
	16-bit	CALL	2 Step

Description: S: Name of subroutine to be called. The subroutine should be created before being called.
The CALL instruction can be used to call the same subroutine as many times as desired. The subroutine also can apply this instruction to call another eight layers of subroutines, including the original one.

■ FOR

NO.	-	FOR	-	S	Nest loops start													
010			-															
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes on the use of operands:		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	-	-	Nest loops end													
012			-															
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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 ~ NEXT loop to execute for N times. After exiting the FOR ~ NEXT loop, the program continues running.

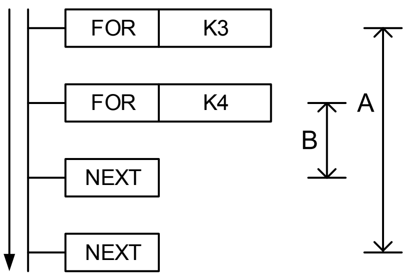
The valid range is specified as N = K1 ~ K32,767. If the specified execution time is N ≦ K1, the time will be K1.

The CJ instruction can be used to exit the FOR ~ NEXT loop.

Followings are the possible errors:

1. The NEXT instruction precedes the FOR instruction.
2. The FOR instruction is executed without the NEXT instruction.
3. END and SRET instructions follow by the NEXT instruction.
4. FOR and NEXT instructions are not in pair.
5. The FOR ~ NEXT loop can nest for up to five layers. If the nesting number exceeds the limit, grammar error might occur.

Example: Program A continues running the NEXT instruction after being repeated for three times. When program A is executed for one time, program B is executed for four times. That is, program B runs for 12 times (3 x 4 = 12) in total.



■ ADD

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

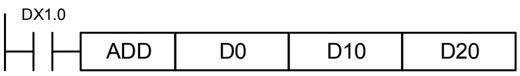
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	●	-	● [V]	-	-
S2							-	-	-	●	●	●	● [V]	●	-	● [V]		-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	●*1	-	-	-	-

Notes when applying operand:
*1: It only supports 16-bit instruction.

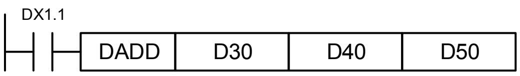
	Instruction	Step No.
32-bit	DADD	7 Step
16-bit	ADD	7 Step

Description: S1: Summand; S2: Addend; D: Sum.
Add values in data sources S1 and S2 in BIN format and save the result in device D. The very first bit of each data represents it’s 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 low 16-bit data; D31, D41 and D51 are the higher one.)



6

■ SUB

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

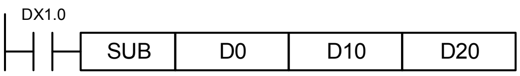
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	●	●	●	● [V]	●	-	● [V]	-	-
							-	-	-	●	●	●	● [V]	●	-	● [V]		-
	-	-	-	-	-	-	-	-	-	-	-	-	●	●*1	-	-	-	-

Notes when applying operand:
*1:It only supports 16-bit instruction.

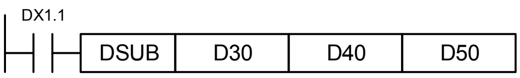
	Instruction	Step No.
32-bit	DSUB	7 Step
16-bit	SUB	7 Step

Description: S1: Minuend; S2: Subrahend; D: Difference.
Subtract values in data sources S1 and S2 in BIN format and save the result in device D. The very first bit of each data represents it's positive sign (0) or negative sign (1). This enables algebraic addition 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 low 16-bit data; D31, D41 and D51 are the higher one.)



6

■ MUL

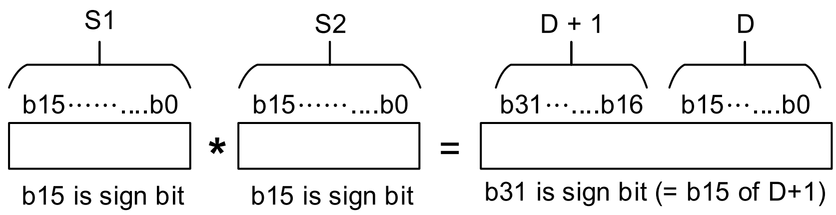
NO.	D	MUL	-	S1, S2, D	BIN multiplication
015			-		

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

Notes when applying operand:
16-bit instruction D operand takes consecutive 2 devices
32-bit instruction D operand takes consecutive 4 devices
*1: It only supports 16-bit instruction.

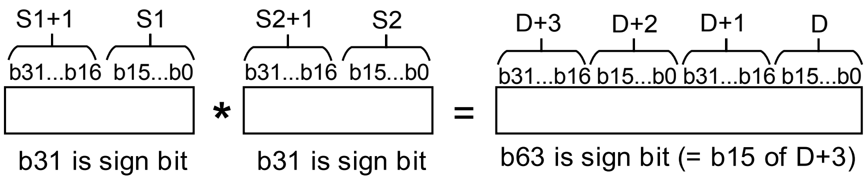
	Instruction	Step No.
32-bit	DMUL	7 Step
16-bit	MUL	7 Step

Description: S1: Multiplicand; S2: Multiplier; D: Product.
Multiply values in data source S1 and S2 in signed binary and save it in device D. When applying 16-bit and 32-bit operation, please pay attention to the sign bit of data in S1, S2 and D.
16-bit BIN multiplication operation:

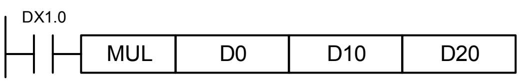


Sign bit = 0 indicataes positive number; sign bit = 1 indicates negative number.

32-bit BIN multiplication operation:



Example: Multiply 16-bit register D0 and D10 and save the outcome in 32-bit register. The higher 16-bit is saved in D21 and the lower 16-bit is saved in D20. On/off of the very left bit indicates the positive and negative status of the result.



6

■ DIV

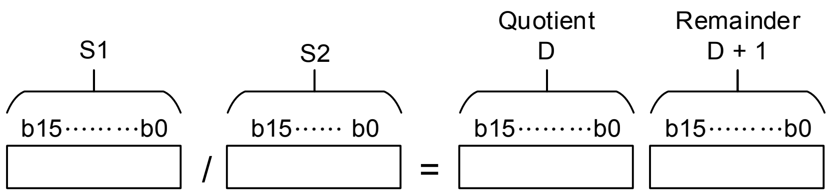
NO.	D	DIV	-	S1, S2, D	BIN division
016			-		

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

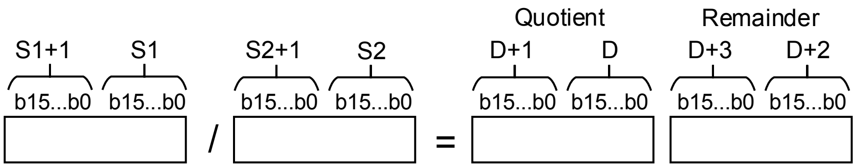
Notes when applying operand:
16-bit instruction D operand takes consecutive 2 devices
32-bit instruction D operand takes consecutive 4 devices
*1:It only supports 16-bit instruction.

	Instruction	Step No.
32-bit	DDIV	7 Step
16-bit	DIV	7 Step

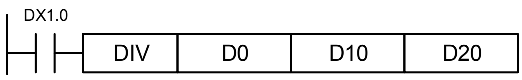
Description: S1: Dividend; S2: Divisor; D: Quotient and remainder.
Data source S1 is divided by S2 in binary format (signed) and the quotient and remainder will be saved in D. When applying 16-bit and 32-bit operation, please pay attention to the sign bit of data in S1, S2 and D.
When divisor is 0, it will not execute the instruction:



32-bit BIN division operation:



Example: Devide 16-bit register D0 and D10 and the result will be a 16-bit quotient and a 16-bit remainder. Save the quotient in D20 and remainder in D21. On/off of the very left bit indicates the positive and negative status of the result.



■ INC

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

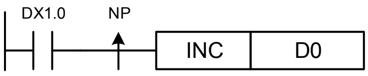
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
D	-	-	-	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-

Notes when applying operand:
32-bit instruction D operand takes consecutive 2 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 will plus 1 every time it is scanned by the program. For 16-bit operation, the sum of 32,767 and 1 is -32,768; For 32-bit operation, the sum of 2,147,483,647 and 1 is -2,147,483,648.

Example: When DX1.0 is Off then On, value in D0 will increase by 1 automatically.



■ DEC

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

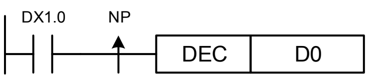
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
D	-	-	-	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-

Notes when applying operand:
32-bit instruction D operand takes consecutive 2 points

	Instruction	Step No.
32-bit	DDEC	3 Step
16-bit	CED	3 Step

Description: D: target device.
When executing this instruction, the value in the specified device D will be minus 1 every time it is scanned by the program. For 16-bit operation, the sum of -32,768 minus 1 is 32,767; For 32-bit operation, the sum of -2,147,483,648 minus 1 is 2,147,483,647.

Example: When DX1.0 is Off then On, value in D0 will decrease by 1 automatically.



■ WAND

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

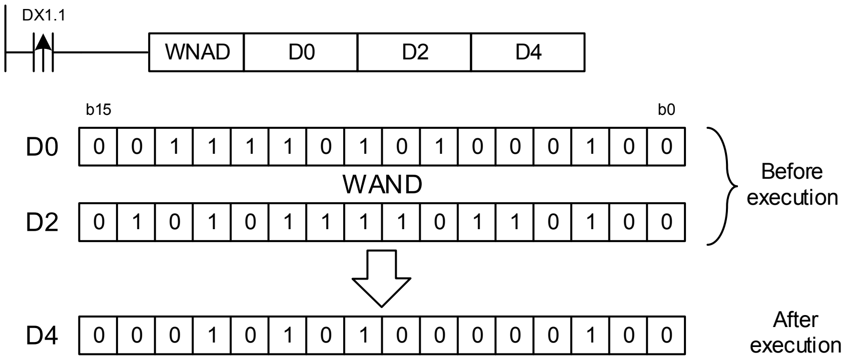
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

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.
Do AND operation on data sources S1 and S2 and save its result in D. Any value in AND operation shows 0, the result is 0.

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



6

■ WOR

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

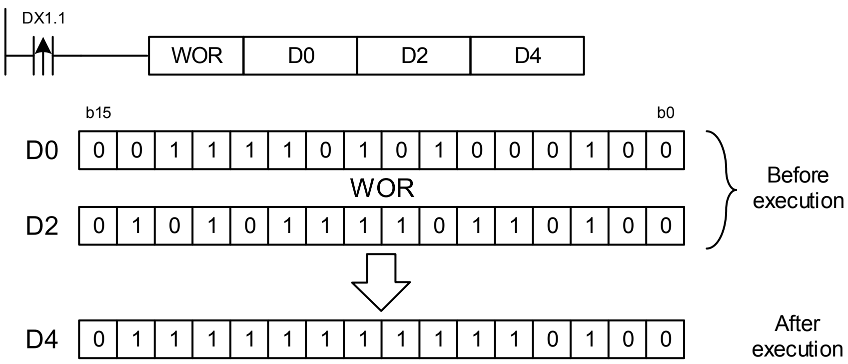
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

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.
Do OR operation on data sources S1 and S2 and save its result in D. Any value in OR operation is 1, the result is 1.

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



6

■ WXOR

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

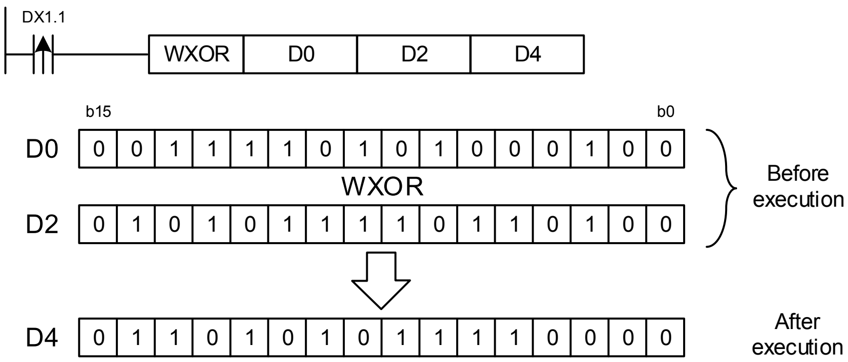
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	●	● [V]	●	-	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

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.
Do XOR operation on data sources S1 and S2 and save its outcome in D. In XOR operation, if both values are the same, the result will be 0, if not, the result will be 1.

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



■ FADD

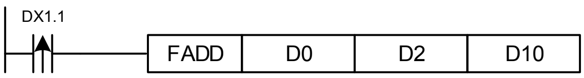
NO.	-	FADD	-	S1, S2, D	Floating point number addition
022			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

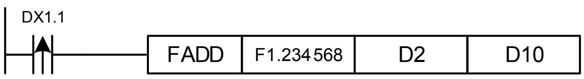
Notes when applying operand:		Instruction	Step No.
	32-bit	FADD	7 Step
	16-bit	-	-

Description: S1: Summand; S2: Addend; D: Sum.
Add the value in register specified by S1 and S2 and save the sum in register specified by D. All operations are executed in floating point number format.

Example 1: When DX1.1 = On, save the sum of two floating point numbers (D1, D0) and two floating point numbers (D3, D2) in (D11, D10).



Example 2: When DX1.1 = On, save the sum of F1.234568 and floating points (D3, D2) in (D11, D10).



■ FSUB

NO.	-	FSUB	-	S1, S2, D	Binary floating point number subtarction
023			-		

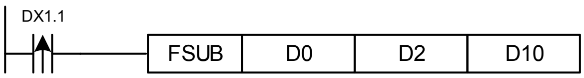
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:

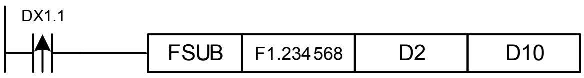
	Instruction	Step No.
32-bit	FSUB	7 Step
16-bit	-	-

Description: S1: Minuend; S2: Subtrahend; D: Difference.
Subtarct the value in register specified by S1 by the value in register specified by S2 and save the outcome in register specified by D. All operations are executed in floating point number format.

Example 1: When DX1.1 = On, subtract floating point numbers (D3, D2) from (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).



6

■ FMUL

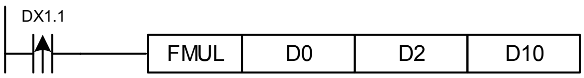
NO.	-	FMUL	-	S1, S2, D	Floating point number multiplication
024			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

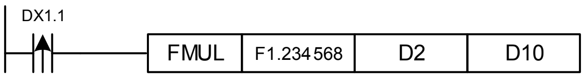
Notes when applying operand:		Instruction	Step No.
	32-bit	FMUL	7 Step
	16-bit	-	-

Description: S1: Multiplier; S2: Multiplier; D: Product
Multiply the value contained in register specified by S1 and S2 and save the product in register specified by D. All operations are executed in floating point number format.

Example 1: When DX1.1 = On, save the product of the floating point numbers (D1, D0) multiply floating point numbers (D3, D2) in (D11, D10).



Example 2: When DX1.1 = On, save the product of F1.234568 mutiply floating point numbers (D3, D2) in (D11, D10).



6

■ FDIV

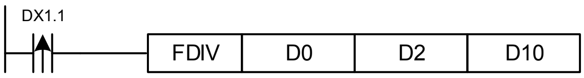
NO.	D	FDIV	-	S1, S2, D							Floating point number division							
025			-															
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
S2	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-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

Notes when applying operand:

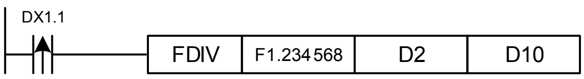
	Instruction	Step No.
32-bit	FDIV	7 Step
16-bit	-	-

Description: S1: Dividend; S2: Divisor; D: Quotient.
Devide the value in the register specified by S1 and S2 and save the quotient in register specified by D. All operations are executed in floating point number format.If the value in S2 is 0, then the instruction is ignored with the error message of “computing error”.

Example 1: When DX1.1 = On, save the remainder of the floating point numbers (D1, D0) divided by floating point numbers (D3, D2) in (D11, D10).



Example 2: When DX1.1 = On, divide F1.234568 by floating point number (D3, D2). Then, save the outcome in (D11, D10).



6

■ FSIN

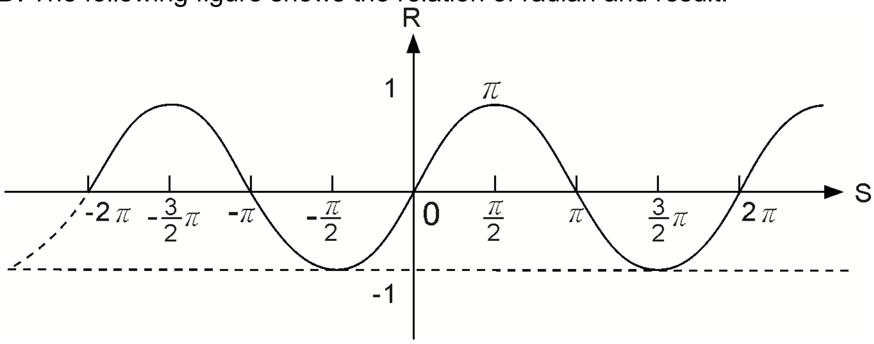
NO.	-	FSIN	-	S1, D	SIN operation in floating point number format
026			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand: S operand takes 2 consecutive devices and in floating point number format D operand takes 2 consecutive devices		Instruction	Step No.
	32-bit	FSIN	5 Step
	16-bit	-	-

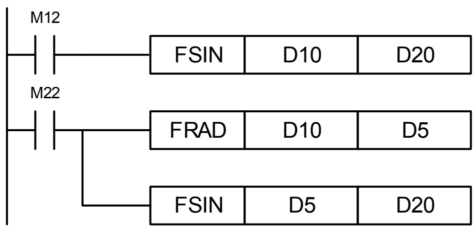
Description: S1: Specified source value (floating point number); D: Result acquired from SIN value (floating point number).

Acquire SIN value from the radian specified by S and save in the register specified by D. The following figure shows the relation of radian and result:



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

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



6

■ FCOS

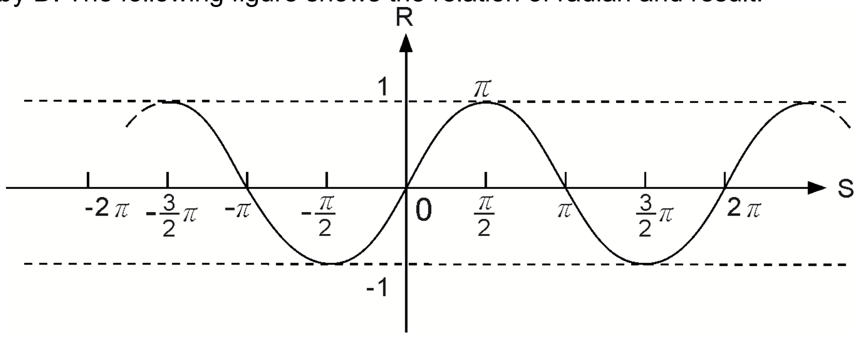
NO.	-	FCOS	-	S1, D	COS operation in floating point number format
027			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:
S operand takes 2 consecutive devices
D operand takes 2 consecutive devices

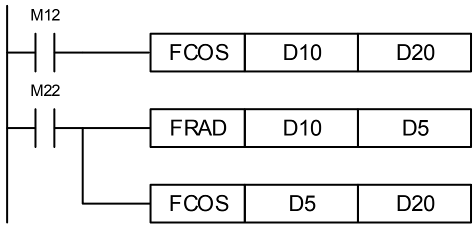
	Instruction	Step No.
32-bit	FCOS	5 Step
16-bit	-	-

Description: S1: Specified source value (floating point number); D: Acquire COS value (floating point number)
Acquire COS value from the radian specified by S and save in the register specified by D. The following figure shows the relation of radian and result:



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

Example: When M12 = On, acquire COS value from RAD value of (D11, D10) and save in (D21, D20), which is in floating point number foramt.



6

■ FTAN

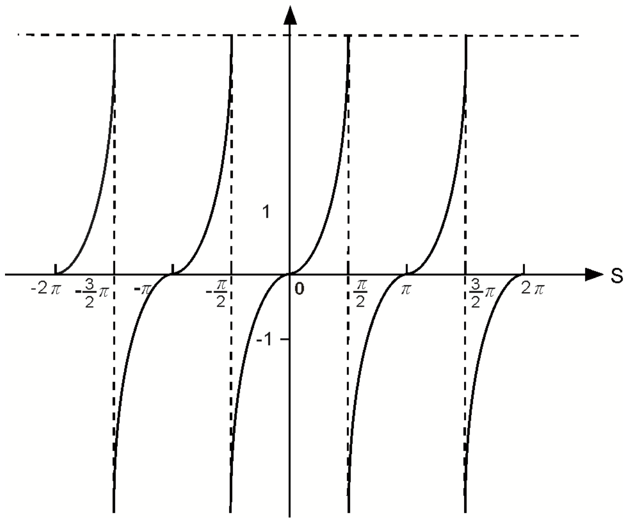
NO.	-	FTAN	-	S1, D	TAN operation in floating point number format
028			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:
S operand takes 2 consecutive devices
D operand takes 2 consecutive devices

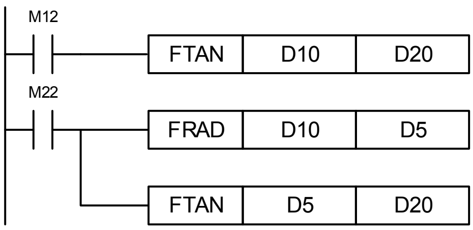
	Instruction	Step No.
32-bit	FTAN	5 Step
16-bit	-	-

Description: S1: Specified source value (floating point number); D: Acquire TAN value (floating point number)
Acquire TAN value from the radian specified by S and save in the register specified by D. The following figure shows the relation of radian and result:



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

Example: When M12 = On, acquire TAN value from FAD value of (D11, D10) and save in (D21, D20), which is in floating point number foramt.



■ FASIN

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

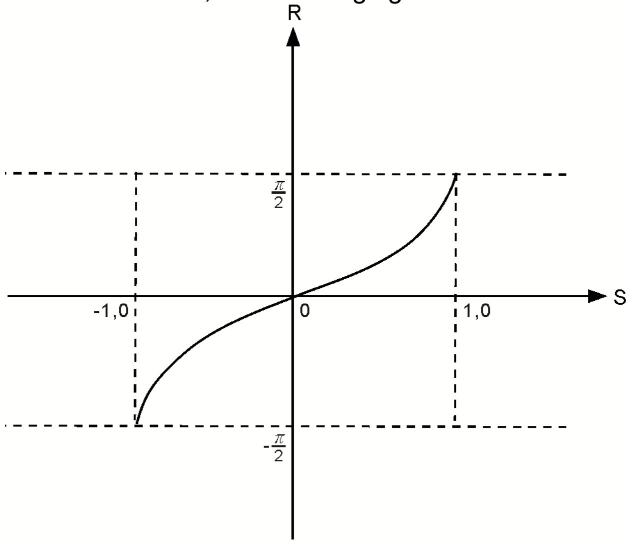
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand:
S operand takes 2 consecutive devices
D operand takes 2 consecutive devices

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

Description: S: Source of specified sine value (floating point number); D: Reault of acquired radian (floating point number).

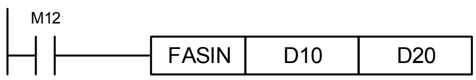
ASIN value = $\sin^{-1}$; the following figure shows the relation of input data andresult.



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

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

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



■ FACOS

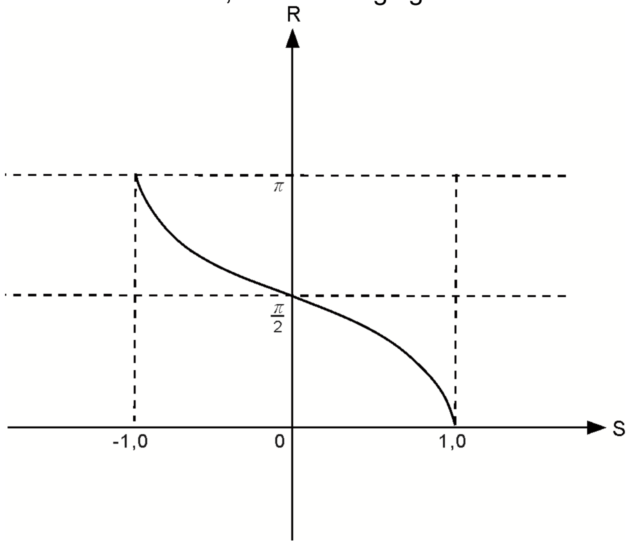
NO.	-	FACOS	-	S1, D	ACOS operation in floating point number format
030			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand: S operand takes 2 consecutive devices and in floating point number format D operand takes 2 consecutive devices		Instruction	Step No.
	32-bit	FACOS	5 Step
	16-bit	-	-

Description: S: Source of specified cosine value (floating point number); D: Result of acquired ACOS value (floating point number)

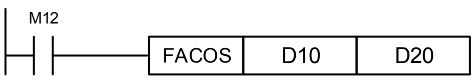
ACOS value = $\cos^{-1}$; the following figure shows the relation of input data and result.



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

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

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



■ FATAN

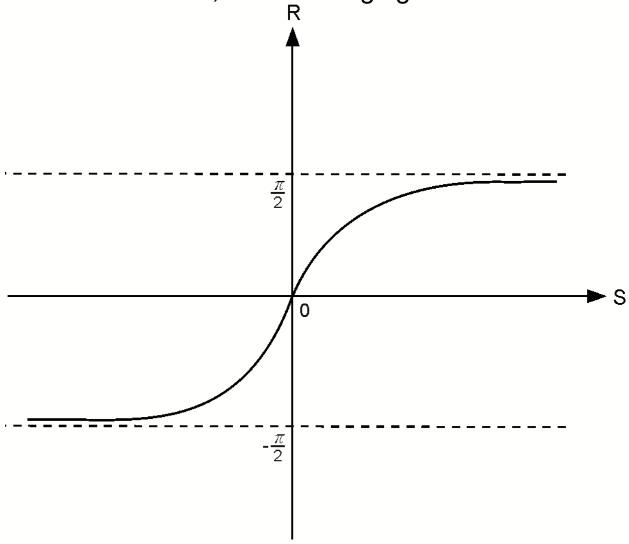
NO.	-	FATAN	-	S1, D	ATAN operation in floating point number format
031			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	●	● [V]	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	● [V]	-	-	● [V]	-	-

Notes when applying operand: S operand takes 2 consecutive devices and in floating point number format D operand takes 2 consecutive devices		Instruction	Step No.
	32-bit	FATAN	5 Step
	16-bit	-	-

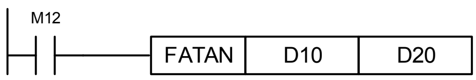
Description: S: Source of specified tangent value (floating point number); D: Radian result of acquired ATAN value (floating point number)

ATAN value = $\tan^{-1}$; the following figure shows the relation of input data and result.



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

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



6

■ ZRST

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

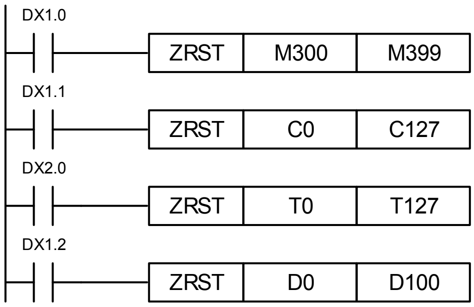
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
D1	-	●	●	●	●	-	-				●	●	●	-	-	-	-	-
D2	-	●	●	●	●	-	-				●	●	●	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	ZRST	4 Step

Description: D1: Start device of zone reset; D2: Ending 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 device 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 all reset (Set the value to 0 and reset contacts and coils to Off).
When DX2.0 = On, counters T0 ~ T127 are all reset (Set the value to 0 and reset contacts and coils to Off).
When DX1.2 = On, data registers D0 ~ D100 are all reset to 0.



■ DECO

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

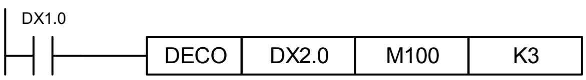
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S	●	●	●	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-
D1	-	●	●	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-
D2	-	-	-	-	-	-	-	-	-	●	-	●	-	-	-	-	-	-

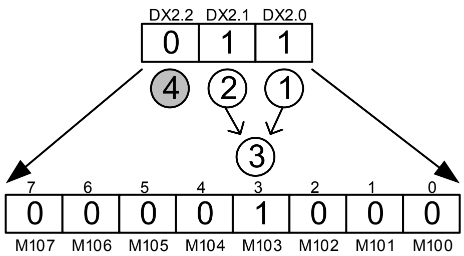
Notes when applying operand:

	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 device D.

Example: When D is a bit device, n = 1 ~ 8. If n = 0 or n > 8, error occurs.
When n = 8, this instruction can decode up to 2^8 = 256 points. (After decoding, please make sure the same range of storage device is not used by different devices.); When DX1.0 = On, DECO instruction decodes values in DX2.0 ~ DX2.0 to M100 ~ M107. When the data source is 1 (bit 1 is on) + 2 (bit 2 is on) = 3, the 3rd bit (M103) from M100 is set to 1. When the DECO instruction is complete and DX1.0 is Off, status that has been decoded remains.





6

■ ENCO

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

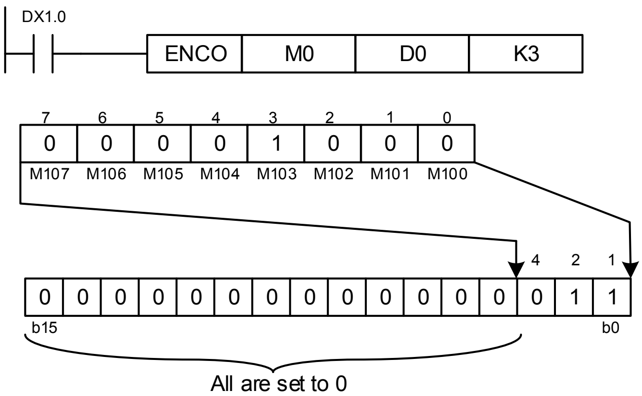
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S	●	●	●	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-
D1	-	●	-	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-
D2	-	-	-	-	-	-	-	-	-	●	-	●	-	-	-	-	-	-

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 lower 2^n bit length in source device S to encode and save the result in device D. If there is more than 1 bit in device S is 1, it will only process the lowest bit 1.

Example: When S is a bit device, n = 1 ~ 8. If n = 0 or n > 8, error occurs. When n = 8, this instruction can encode up to 2^8 = 256 points. When DX1.0 = On, ENCO instruction will encode data (M0 ~ M7) and save in lower 3-bit (b2 ~ b0) of D0. Bits that are not used in D0 (b15 ~ b3) will be set to 0. When ENCO instruction is complete and DX1.0 is Off, status in device D remains.



■ BON

NO.	D	BON	-	S1, D, n	Monitor specified bit status
035			-		

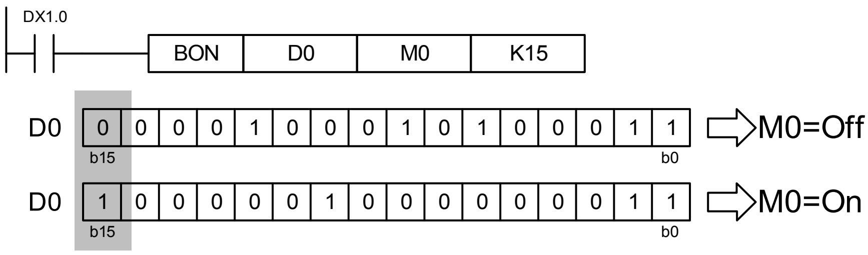
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S	-	-	-	-	-	-	-	-	-	-	●	●	● [V]	-	-	-	-	-
D	-	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
n	-	-	-	-	-	-	-	-	-	●	-		-	-	-	-	-	-

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

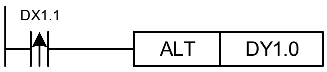
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
D	-	●	●	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-

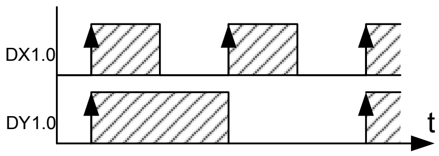
Notes when applying operand:

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

Description: D: Target device.
When executing ALT instruction, D alternates On between 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

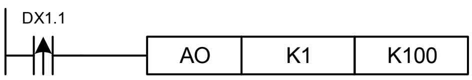
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

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

Description: S1: Channel selection (K0 ~ K31); S2: Output voltage
The AO instruction can be applied to output the voltage via analog output module.

Example: When DX1.0 = On, the output voltage of CH1 analog output module is 100 (Unit is changed in accordance with different control mode.)



6

■ AI

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

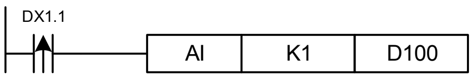
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	F	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

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 will display the analog input value of CH1



6.1.4 Single-axis motion instruction

PLC issues motion control instruction via special register W and special relay R. Motion kerne can percieve the change of PLC instruction by monitoring the register. Then, the system controls the motion through DMCNET communication. IMP editor will program the commonly used motion control to motion instruction. When PLC issues the motion instruction, the parameters will be set to the corresponding register position.

■ SVON

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

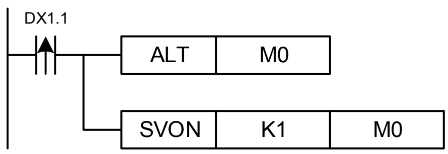
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

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

Description: S1: No. of servo axis (K0 ~ K31); S2: Control flag of Servo ON/OFF. When SVON instruction is being executed, servo drive of node number S1 will be on/off in accordance with S2 status.

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.



■ UNIT

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

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

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

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

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

■ SCUR

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

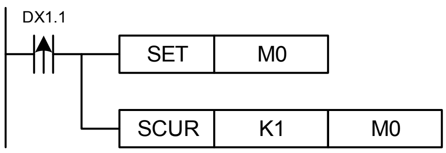
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

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

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

Example: When DX1.1 = On, the first axis applies S-curve acceleration/deceleration.



■ TADC

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

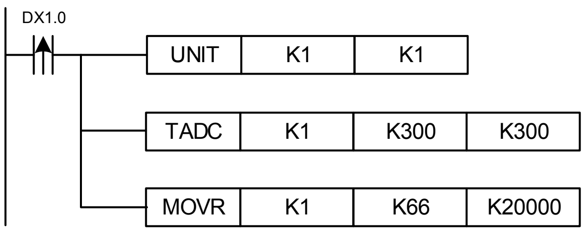
	Bit Device						Word Device												External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

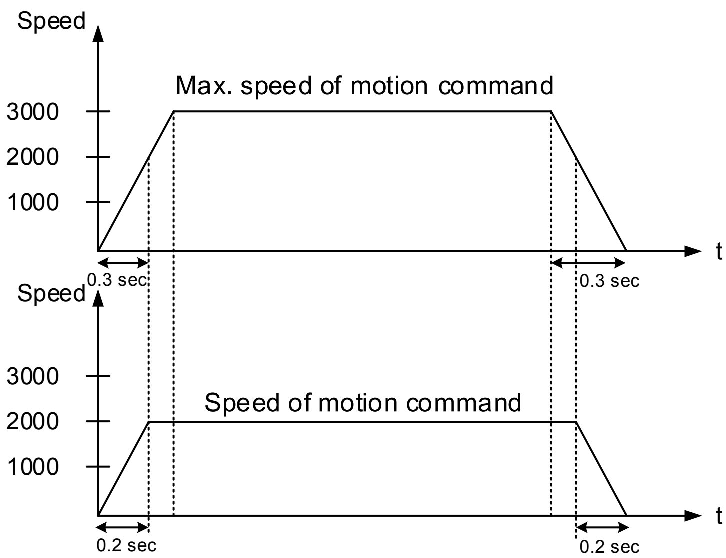
Notes when applying operand:
S2 operand takes 2 consecutive devices
S3 operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: Acceleration (Unit: 1 ms); S3: Deceleration (Unit: 1 ms)
When GTADC instruction is executed, the acceleration time is the time between motor stop and the highest speed. And the deceleration time is the time between the highest speed and motor stop (it can be set via Quick start interface).

Example: Assuming that the max. speed of Axis 1 is set to 3000 rpm via Quick start interface, when DX1.0 = On, its motion unit will be percentage, acceleration and deceleration time is 0.3 seconds and the axis will run 20 mm with the speed of 2000 rpm.





■ SLMT

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

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices
S3 operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: Positive software limit (unit: 0.001 mm); S3: Negative software limit (unit: 0.001 mm).

Example: When the SDSTP instruction is executed, servo drive's positive limit position of station number S1 will be S2 and its negative limit position will be S3. Note: This instruction shall work with the SLMTON instruction.

■ SLMTON

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

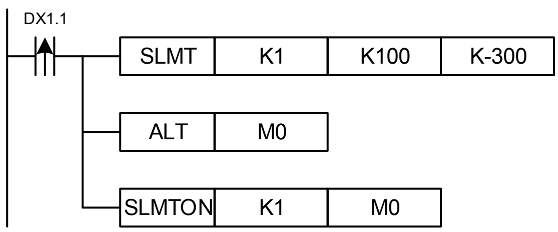
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes when applying operand:

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

Description: S1: No. of servo axis (K1 ~ K36); S2: off-Disable software limit / on-Enable software limit.

Example: When DX1.1 is on for the first time, reverse software limit of the first servo axis is set to 0.1 mm, forward software limit is set as -0.3 mm and the function of software limit will be enabled. When DX1.1 is on for the second time, the software limit will be disabled.



■ COORD

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

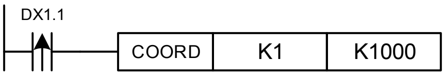
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices

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

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

Example: When DX1.1 = On, current position of the first axis will be changed to 1000 (unit: 0.001 mm).



■ HOME

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

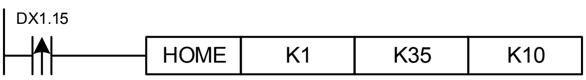
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S3 operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36): S2: Homing mode; S3: Offset distance (unit: 0.001 mm).
When the HOME instruction is executed, the specified servo axis will do homing by the specified homing method. Then, use the offset value set by S3 and set S3 as the origin coordinate.
Note: When applying homing mode, switch of positive/negative limit and homing signal shall be connected to the servo drive.

Example: When DX1.15 = On, the first servo axis should search the homing point with homing mode 35 and set the position as 0.010 mm.



6

■ ALMR

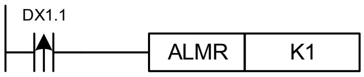
NO.	-	ALMR	-	S1	Alarm clearance
058			-		

	Bit Device						Word Device												External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

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

Description: S1: No. of servo axis (K1 ~ K36).
When servo alarm occurs, the ALRM instruction can be used to clear the alarm.

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



■ ESTP

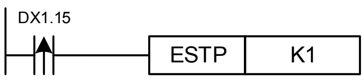
NO.	-	ESTP	-	S1	Emergency stop
059			-		

	Bit Device						Word Device												External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

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

Description: S1: No. of servo axis (K1 ~ K36).
When the ESTP instruction is executed, the specified servo will decelerate from the highest speed to stop.

Example: When DX1.15 = On, the fisrt servo axis will immediately decelerate to stop.



For avoiding dangers, emergency stop shall have the following functions:

1. The emergency stop signal shall trigger servo DI of emergency stop.
2. The emergency stop signal should cut off the servo power circuit.
3. Please install the safety circuit according to the actual requirement.

6

■ SDSTP

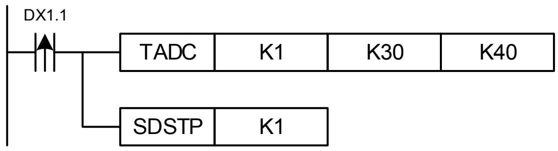
NO.	-	SDSTP	-	S1	Decelerate to stop
060			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

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

Description: S1: No. of servo axis (K1 ~ K36).
When the SDSTP instruction is executed, servo drive of node number S1 will decelerate to stop.

Example: When DX1.1 = On, the first servo axis will decelerate to stop in 0.04 seconds.



■ AXRPM

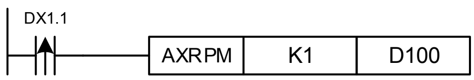
NO.	-	AXRPM	-	S1, D	Access motor’s current speed
061			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

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

Description: S1: No. of servo axis (K1 ~ K36).
Users can acquire motor’s current speed via the AXRPM instruction.

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



■ AXTQR

NO.					
062	-	AXTQR	- / -	S1, D	Access motor's current torque

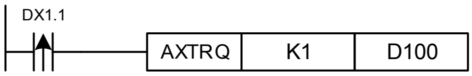
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:

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

Description: S1: No. of servo axis (K1 ~ K36); D: Torque of servo axis (‰).
Users can acquire the motor's current output torque via the AXTQR instruction.

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



■ RSVP

NO.					
063	-	RSVP	- / -	S1, S2	Access servo parameters

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2										●			●					

Notes when applying operand:
D operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: Servo parameters groups x 1000 + parameter No.

■ SVR

NO.	-	SVR	-	S1, S2	Access the returned value from servo
063-1			-		

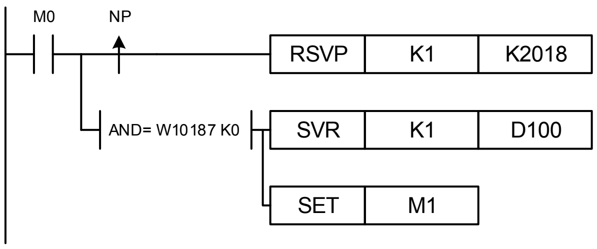
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2										●			●					

Notes when applying operand:
D operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: Accessing the storage position of servo parameters.
Users can access servo parameters via the RSVP instruction and acquire the returned value via the SVR instruction. The servo parameter will not be changed promtly. Thus, issue RSVP, WSVP and SVR instructions after the parameter setting is complete.

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



■ WSVP

NO.	-	WSVP	-	S1、S2、D	Write-in servo parameters
064			-		

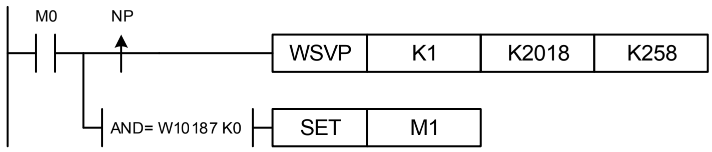
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2										●			●					
D	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S3 operand takes 2 consecutive devices		Instruction	Step No.
	32-bit	WSVR	7 Step
	16-bit	-	-

Description: S1: No. of servo axis (K1 ~ K36); S2: Servo parameters group x 1000 + parameter No.; D: Write in parameters

Users can write servo parameters via the WSVP instruction. The servo parameter will not be changed promtly. Thus, issue RSVP and WSVP instructions after the parameter setting is complete. This condition can only be applied for one axis.

Example: When M0 = On, P2-18 of servo axis 1 is set to 258 (0x0102). Then, M1 is on.



It is suggested to apply write-in parameters function when servo off.

■ SVSTS

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

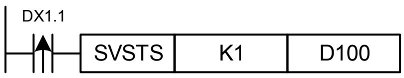
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
	Bit Device						Word Device										External D.	
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:

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

Description: S1: No. of servo drive (K1 ~ K36); D: DO status of servo axis.
Users can acquire digital output status of the servo drive via the SVITS instruction.

Example: When DX1.1 = On, the digital output status of the first servo drive will be saved in D100.



■ SVITS

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

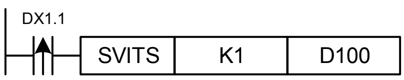
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
	Bit Device						Word Device										External D.	
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:

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

Description: S1: No. of servo axis (K1 ~ K36); D: DI status of servo axis
Users can acquire digital input status of the servo drive via the SVITS instruction.

Example: When DX1.1 = 0, the digital input status of the first servo drive will be saved in D100.



6

■ RCBL

NO.	-	RCBL	-	S1, D	Access the buffer usage
067			-		

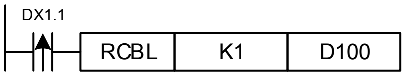
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:

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

Description: S1: No. of servo axis (K1 ~ K36); D: Buffer usage
The RCBL instruction can be used to access current buffer status.

Example: When DX1.1 = On, save the buffer usage in D100.



■ RPOS

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

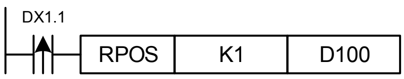
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:
D operand takes consecutive 2 points

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

Description: S1: No. of servo axis (K1 ~ K36); S2: Actual position of servo axis (unit: 0.001 mm).
Users can acquire the motor's actual position via the RPOS instruction.
Note: Through the encoder, servo motor sends the feedback coordinates to the servo drive. And that coordinate is called the actual position.

Example: When DX1.1 = On, the first motor's feedback position will be saved to (D101, D100).



6

■ LPOS

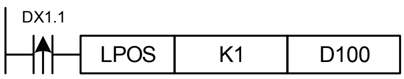
NO.	-	LPOS	-	S1, S2	Access the axial instruction position
069			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:		Instruction	Step No.
D operand takes 2 consecutive devices	32-bit	LOPOS	7 Step
	16-bit	-	-

Description: S1: No. of servo axis (K1 ~ K36); S2: Instruction position issued by motion control card (unit: 0.001 mm).
When issuing any motion instructions, DMCNET master will divide the path into several nodes and issue several instructions to reach the target position. And when the instruction is sent every 1 ms, the LPOS instruction can be used to acquire the node position of current instruction.

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



■ TPOS

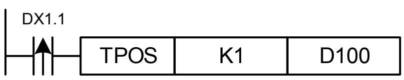
NO.	-	TPOS	-	S1, S2	Access the axial target position
070			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

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

Description: S1: No. of servo axis (K1 ~ K36); S2: Target position of servo axis (unit: 0.001 mm).
The TPOS instruction can be used to acquire the target position
Note: The final position specified by the motion instruction is called target position.

Example: When DX1.1 = 0, target instruction of the first servo axis will be saved in (D101, D100).



6

■ MOTS

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

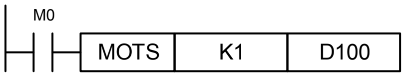
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:

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

Description: S1: No. of servo axis (K1 ~ K36); D: Status of motion instruction
The MOTS instruction can be used to acquire the status of servo axis.

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



■ ALE

NO.	-	ALE	-	S1, D	Access servo drive error code(s)
072			-		

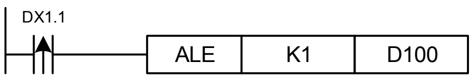
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-		-	-	●	-	-	-	-	-

Notes when applying operand:

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

Description: S1: No. of servo axis (K1 ~ K36); D: Error code of servo axis
The ALE instruction can be used to acquire the error code of servo drive.

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



■ JOG

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

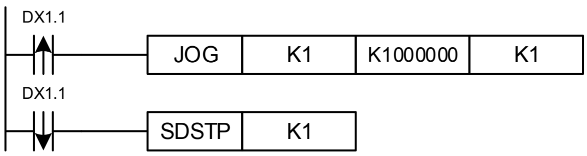
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: Jog speed (default unit: PUU/s; it can be changed by the instruction); S3: Operation direction
When the JOG instruction is executed, the motor specified by S1 instruction will accelerate then runs at the constant speed when it reaches the speed specified by S2. After the JOG instruction is complete, the motor starts to decelerate to stop.

Example: When DX1.1 = On, the first servo axis will run with the speed of 1000000 PUU/s in reverse direction.



■ MOVA

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

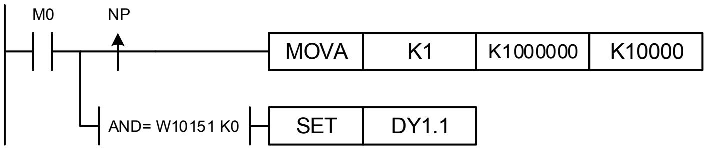
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices
S3 operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: motion speed (default unit: PUU/s, it can be changed by the instruction); S3: Target position.
When the MOVA instruction is executed, the motor of node number S1 runs at the speed set by S2. Then, it stops when reaching the position specifiied by S3.

Example: When M0 = On, the first motor runs with the speed of 1000000 PUU/s to the position of 10 mm. When the first motor reaches the position, DY1.1 is On.



6

■ MOVR

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

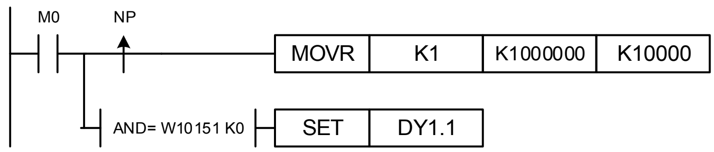
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices
S3 operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: motion speed (default unit: PUU/s, it can be changed by the instruction); S3: Motion target.
When the MOVR instruction is executed, the motor with node number S1 runs at the speed set by S2 and stops when reaching the position (current position + S3).

Example: When M0 = On, the first motor runs with the speed of 1000000 PUU/s to the position (current position + 10 mm). Then DY1.1 is On.



■ MOVPOS

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

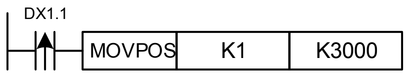
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: Motion position.
When the MOVPOS instruction is executed, servo axis of node number S1 will update its target position to the one set by S2; This instruction can only be applied in absolute motion.

Example: When DX1.1 = On, the target position of the first servo axis will be changed to 3.000 mm.



■ MOVSPD

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

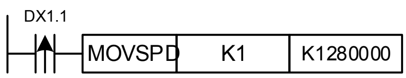
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: motion speed (default unit: PUU/s, it can be changed by the instruction.)
When the MOVSPD instruction is executed, servo axis of node number S1 will change to the speed set by S2. This instruction can be applied in single-axis motion (absolute) only.

Example: When DX1.1 = O n, speed of the first servo motor will be changed to 1280000 PUU/s.



■ SPD

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

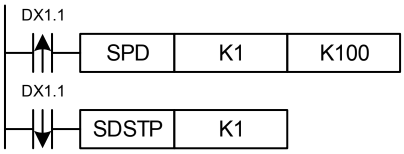
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
	Bit Device						Word Device										External D.	
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: Motion speed (default unit: rmp)
When the SPD instruction is executed, servo axis of node number S1 will accelerate first. Then, it runs at the constant speed when reaching the speed set by S2.

Example: When DX1.1 = On, the first servo motor runs with the speed of 100 rpm in forward direction; When DX1.1 = Off, the first servo motor will decelerate to stop.



■ TRQ

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

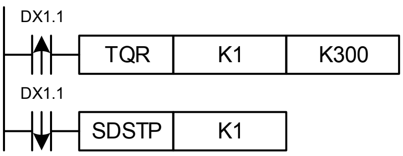
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
	Bit Device						Word Device										External D.	
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); S2: Output target torque (±1000) (‰).
When the TQR instruction is executed, servo axis of node number S1 outputs the torque that specified by S2.

Example: When DX1.1 = on, the first servo motor runs with the rated torque of 300 ‰ in forward direction.



■ RSPD

NO.	-	RSPD	-	S1, D	Access the current speed
080			-		

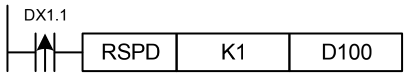
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

Notes when applying operand:
D operand takes 2 consecutive devices

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

Description: S1: No. of servo axis (K1 ~ K36); D: Servo axis speed (unit: 0.001 mm/min). RSPD instruction can be used to acquire the motor speed in linear motion.

Example: When DX1.1 = On, speed of the first servo axis in linear motion will be saved to (D101, D100).



6.1.5 Interpolation Motion Instruction

IMP system supports multi-axis synchronous motion. Functions like linear interpolation of any two or three axes, helical interpolation of any three axes or continuous interpolation can be applied in the same motion card.

Group setting GSET ⇨ Parameter setting ⇨ Target setting of each axis MOVP ⇨ Motion triggered

Parameters setting table of grouping motion:

	GUNIT	GSCUR	GTADC	ANGLE	DIR	PITCH	DEPTH	CENTER	ENDXY
MOVLA	△	△	△	-	-	-	-	-	-
MOVLR	△	△	△	-	-	-	-	-	-
CIRCAA	△	△	△	●	△	-	-	●	
CIRCAR	△	△	△	●	△	-	-	●	
CIREAA	△	△	△	●	△	-	-	-	●
CIREAR	△	△	△	●	△	-	-	-	●
CIRCEA	△	△	△	-	△	-	-	●	●
CIRCER	△	△	△	-	△	-	-	●	●
HELIXA	△	△	△	-	●	●	●	●	-
HELIXR	△	△	△	-	●	●	●	●	-

●: Compulsory / △: Option / -: N/A

■ GSET

NO.	-	GSET	-	S1, S2, S3	Group setting
100			-		

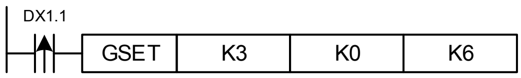
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

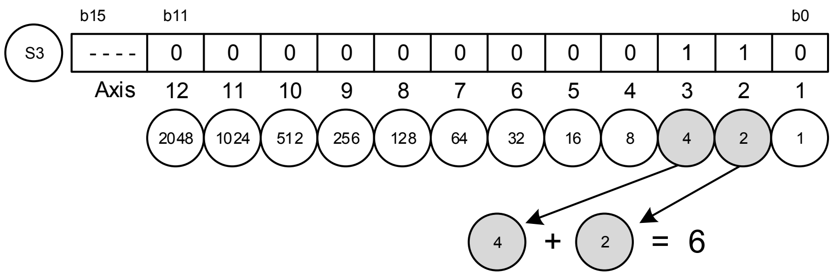
Notes when applying operand:

	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, the GSET instruction has to be used to assign the applied axes as a group.

Example: When DX1.1 = On, the second and third axis of card no. 0 will be assign as motion group 3.





6

■ GUNIT

NO.	-	GUNIT	-	S1, S2	Group setting of speed unit
101			-		

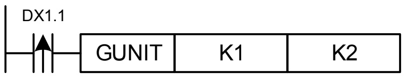
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

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 group interpolation function, the GUNIT instruction can be used to select the speed unit.

Example: When DX1.1 = On, unit of the first motion group is mm/min.



■ GSCUR

NO.	-	GSCUR	-	S1, S2	Group setting of acceleration/deceleration curve
102			-		

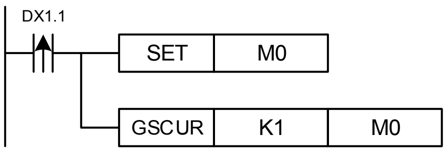
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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 group interpolation function, the GSCUR instruction can be used to select acceleration/deceleration smoothing curve.

Example: When DX1.1 = On, the motion group 1 will be set as S-curve acceleration/deceleration.



■ GTADC

NO.	-	GTADC	-	S1, S2, S3	Group setting of acceleration/deceleration time
103			-		

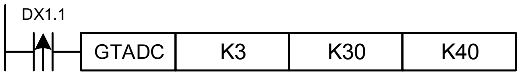
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes consecutive 2 points
S3 operand takes consecutive 2 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 group interpolation, the GTADC instruction can be used to set the acceleration/deceleration time.

Example: When DX1.1 = On, the acceleration and deceleration time of motion group 3 is 0.03 seconds and 0.04 seconds, respectively.



■ ANGLE

NO.	-	ANGLE	-	S1, S2	Arc angle
104			-		

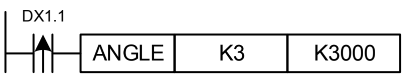
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

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 users wish to apply the arc motion instruction, the ALGLE instruction can be used to set the included angle of arc.
Note: This instruction shall work with the arc interpolation instruction.

Example: When DX1.1 = On, the arc angle of motion group 3 will be 300 degrees.



6

■ DIR

NO.	-	DIR	-	S1, S2	Arc direction
105			-		

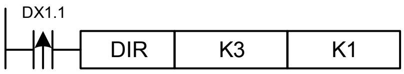
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

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 users wish to use arc motion instruction, the DIR instruction can be used to set the arc direction.
Note: This instruction shall work with the arc interpolation instruction.

Example: When DX1.1 = On, the arc direction of motion group 3 will be clockwise.



■ PITCH

NO.	-	PITCH	-	S1, S2	Helix pitch
106			-		

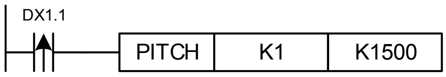
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: Group No. (K1 ~ K40); S2: Helix pitch (unit: 0.001 mm).
When users wish to apply helical motion instruction, the PITCH instruction can be used to set the helical pitch.

Example: When DX1.1 = On, the helix pitch of motion group 3 will be 1.5 mm.



■ DEPTH

NO.	-	DEPTH	-	S1, S2	Helix depth
107			-		

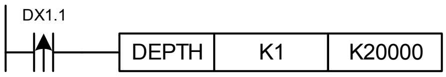
	Bit Device						Word Device									External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: Group No. (K1 ~ K40); S2: Helix depth (unit: 0.001 mm).
When users wish to apply helical motion instruction, the DEPTH instruction can be used to set the total helix depth.

Example: When DX1.1 = On, the helix depth of motion group 3 will be 20 mm.



■ CENTER

NO.	-	CENTER	-	S1, S2, S3	Arc’s circle center
108			-		

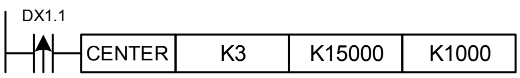
	Bit Device						Word Device									External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices
S3 operand takes 2 consecutive devices

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

Description: S1: Group No. (K1 ~ K40); S2: Arc’s circle center on X-axis (unit: 0.001 mm); S3: Arc’s circle center on Y-axis (unit: 0.001 mm).
When users wish to apply arc motion instruction, the CENTER instruction can be used to set the circle center.
Note: This instruction shall work with the arc interpolation instruction.

Example: When DX1.1 = On, the arc’s circle center of motion group 3 will be (150.00, 10.00).



■ ENDXY

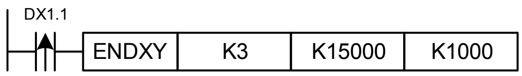
NO.	-	ENDXY	-	S1, S2, S3							End point of arc							
109			-															
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices
S3 operand takes 2 consecutive devices

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

Description: S1: Group No. (K1 ~ K40); S2: End point of arc on X-axis (unit: 0.001 mm); S3: End point of arc on Y-axis (unit: 0.001 mm).
When users wish to apply arc motion instruction, the ENDXY instruction can be used to set the end point of arc.
Note: This instruction shall work with arc interpolation instruction.

Example: When DX1.1 = On, the end point of arc of motion group 3 will be (150.000, 1.000).



■ MOVP

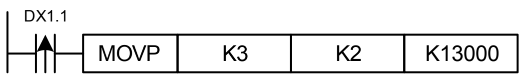
NO.	-	MOVP	-	S1, S2, S3							Target position setting of each axis							
110			-															
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S3	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S3 operand takes 2 consecutive devices

	Instruction	Step No.
32-bit	MOVP	13 Step
16-bit	-	-

Description: S1: Group No. (K1 ~ K40); S2: No. of target axis (K1 ~ K12); S3: Target position (unit: 0.001 mm).
When users wish to apply multi-axis synchronous function, the MOVP instruction can be used to set the target position of each axis.

Example: When DX1.1 = On, the target position of axis 2 in motion group 3 is 13 mm.



6

■ MOVLA

NO.	-	MOVLA	-	S1, S2	Linear motion (absolute)
111			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

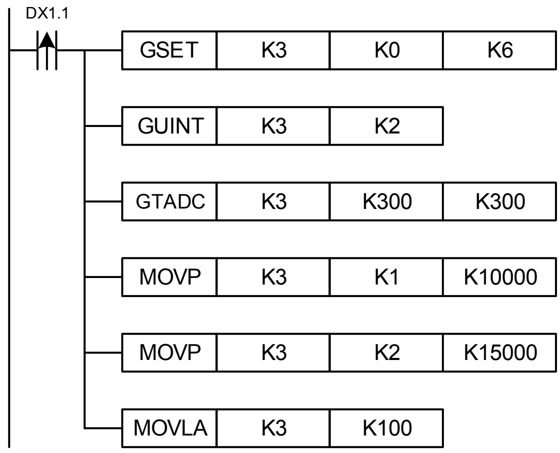
Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: Group No. (K1 ~ K40); S2: Target speed (default unit: PUU/s, it can be set by GUNIT instruction).
To move to absolute position by multi-axis linear motion, the MOVLA instruction can be used to activate the function.

Example: When DX1.1 = On:

1. Motion card no. 0 and servo drives of node number 2 and 3 will be grouped into motion group 3.
2. Speed unit of motion group 3 will be mm/min.
3. The acceleration and deceleration time of motion group 3 will be 0.3 and 0.03 seconds respectively.
4. Target position of axis 1 will be 10.000 mm.
5. Target position of axis 2 will be 15.000 mm.
6. Motion group 3 is triggered to execute linear interpolation of two-axis (absolute position).



■ MOVLR

NO.	-	MOVLR	-	S1, S2	Linear motion (relative)
112			-		

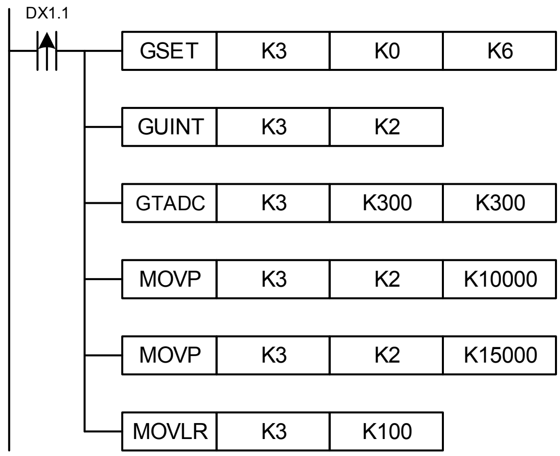
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand takes 2 consecutive devices		Instruction	Step No.
	32-bit	MOVLA	7 Step
	16-bit	-	-

Description: S1: Group No. (K1 ~ K40); S2: Target speed (default unit: PUU/s, it can be set by GUNIT instruction.)
To move to the relative position by multi-axis linear motion, the MOVLR instruction can be used to activate the function.

Example: When DX1.1 = On:

1. Motion card no. 0 and servo drives of node number 2 and 3 will be grouped into motion group 3.
2. Speed unit of motion group 3 will be mm/min.
3. The acceleration and deceleration time of motion group 3 will be 0.3 and 0.03 seconds respectively.
4. The incremental position of axis 1 will be 10.000 mm.
5. The incremental position of axis 2 will be 15.000 mm.
6. Motion group 3 is triggered to execute linear interpolation of two-axis (relative position).



■ CIRCAA

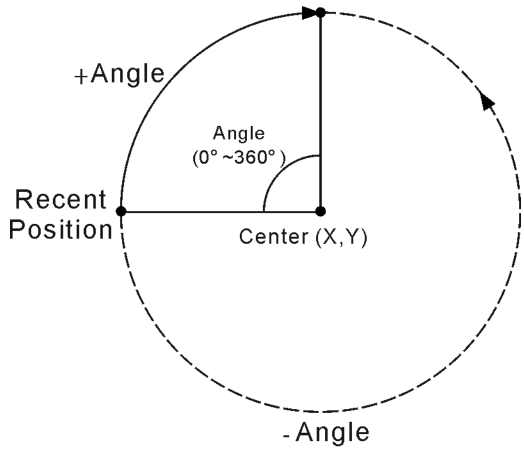
NO.	-	CIRCAA	-	S1, S2	Arc motion (with the known absolute coordinate of circle center and angle).
113			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:
S2 operand takes 2 consecutive devices

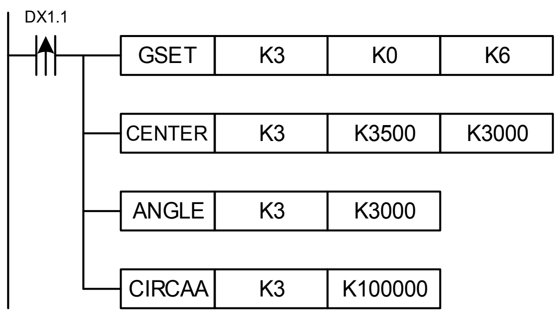
	Instruction	Step No.
32-bit	CIRCAA	7 Step
16-bit	-	-

Description: S1: Group No. (K1 ~ K40); S2: Target speed (default unit: PUU/s, it can be set by GUNIT instruction).
When users wish to apply arc interpolation of two-axis, with the known absolute coordinate of circle center and included angle of arc, the CIRCAA instruction can be used to activate this function.



Example: When DX1.1 = On:

1. Motion card no. 0 and servo drives of node number 2 and 3 will be grouped into motion group 3.
2. The circle center position of motion group 3 is (3.500, 3.000) mm.
3. The arc angle of motion group 3 is 300 degrees.
4. Motion group 3 is triggered to execute arc interpolation of two-axis (absolute position).



■ CIRCAR

NO.	-	CIRCAR	-	S1, S2	Arc motion (with the known relative coordinate of circle center and angle)
114			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

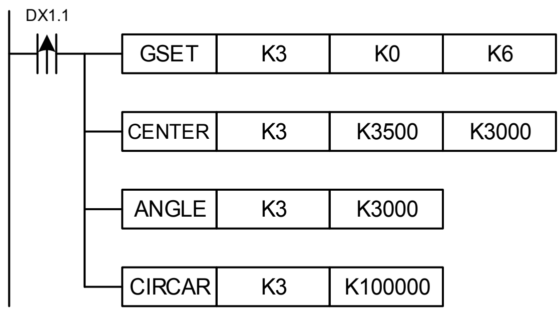
Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: Group No. (K1 ~ K40); S2: Target speed (default unit: PUU/s, it can be set by Gunit instruction).
When users wish to apply arc interpolation of two-axis, with the known distance between the arc's circle center and current point and the included angle, the CIRCAR instruction can be used to activate the function.

Example: When DX1.1 = On:

1. Motion card no. 0 and servo drives of node number 2 and 3 will be grouped into motion group 3.
2. The circle center position of motion group 3 is (current position X + 3.500, current position Y + 3.000) mm.
3. The arc angle of motion group 3 is 300 degrees.
4. Motion group 3 is triggered to execute arc interpolation of two-axis (relative position).



■ CIREAA

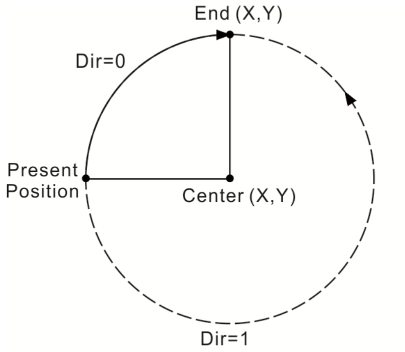
NO.	-	CIREAA	-	S1, S2	Arc motion (with the known absolute coordinate of arc end point and angle)
115			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

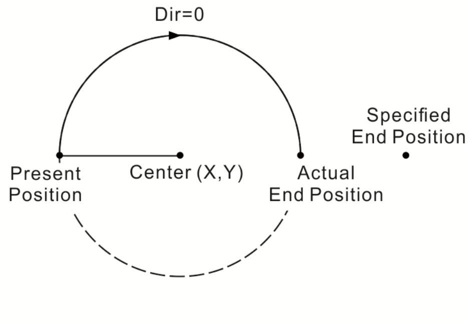
Notes when applying operand: S2 operand takes 2 consecutive devices		Instruction	Step No.
	32-bit	CIREAA	7 Step
	16-bit	-	-

Description: S1: Group No. (K1 ~ K40); S2: Target speed (default unit: PUU/s, it can be set by GUNIT instruction).

To move to the absolute position by arc interpolation of two-axis, with the known coordinate of end point and included angle of arc, the CIREAA instruction can be used to activate this function.



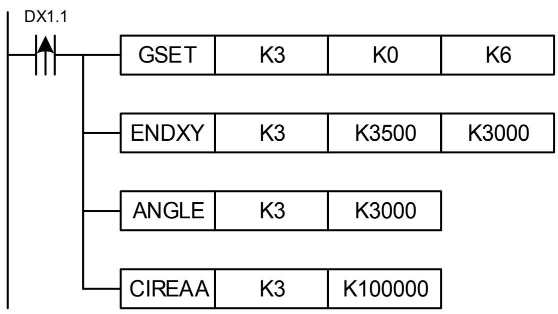
Actual End Position=Specified End Position



Actual End Position<>Specified End Position
Radius=Present Position To Center

Example: When DX1.1 = On:

1. Motion card no. 0 and servo drives of node number 2 and 3 will be grouped into motion group 3.
2. The end point of motion group 3 is (3.500, 3.000) mm.
3. The arc angle of motion group 3 is 300 degrees.
4. Motion group 3 is triggered to execute arc interpolation of two-axis (absolute position).



■ CIREAR

NO.	-	CIREAR	-	S1, S2	Arc motion (with the known relative coordinate of end point and angle)
115			-		

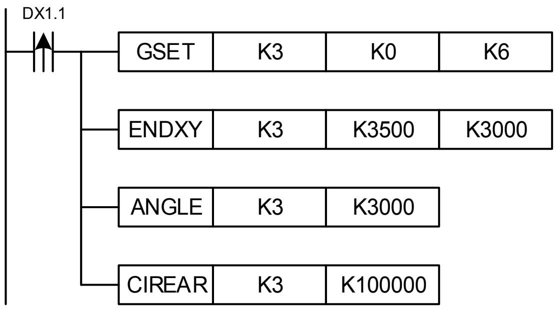
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand takes 2 consecutive devices		Instruction	Step No.
	32-bit	CIREAR	7 Step
	16-bit	-	-

Description: S1: Group No. (K1 ~ K40); S2: Target speed (default unit: PUU/s, it can be set by GUNIT instruction).
To move to the relative position by arc interpolation of two-axis, with the known coordinate of end point and the included angle of arc, the CIREAR instruction can be used to activate the function.

Example: When DX1.1 = On:

1. Motion card no. 0 and servo drives of node number 2 and 3 will be grouped into motion group 3.
2. The end point of motion group 3 is (current position X + 3.500, current position Y + 3.000) mm.
3. The arc angle of motion group 3 is 300 degrees.
4. Motion group 3 is triggered to execute arc interpolation of two-axis (relative position).



■ CIRCEA

NO.	-	CIRCEA	-	S1, S2	Arc motion (with the known absolute coordinate of arc's circle center and end point)
116			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

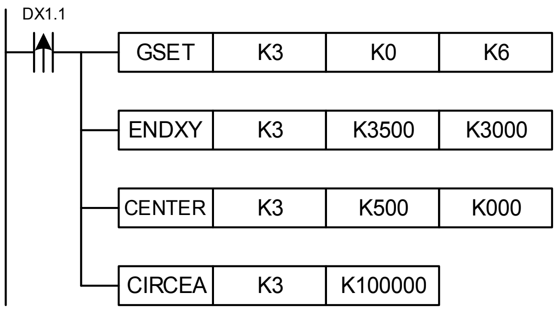
Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: Group No. (K1 ~ K40); S2: Target speed (default unit: PUU/s, it can be set by GUNIT instruction).
When users wish to apply arc interpolation of two-axis, with the known absolute position of arc's circle center and end point, the CIRCEA instruction can be used to activate this function.

Example: When DX1.1 = On:

1. Motion card no. 0 and servo drives of node number 2 and 3 will be grouped into motion group 3.
2. The end point of motion group 3 is (3.500, 3.000) mm.
3. The arc's circle center of motion group 3 is (0.500, 0.000) mm.
4. Motion group 3 is triggered to execute arc interpolation of two-axis (absolute position).



■ CIRCER

NO.	-	CIRCER	-	S1, S2	Arc motion (with the known relative coordinate of arc' circle center and end point)
117			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

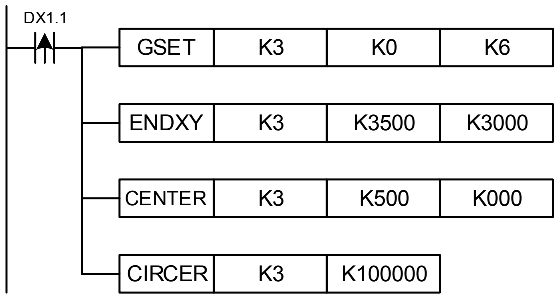
Notes when applying operand:
S2 operand takes 2 consecutive devices

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

Description: S1: Group No. (K1 ~ K40); S2: Target speed (default unit: PUU/s, it can be set by GUNIT instruction).
To move to the relative position by arc interpolation of two-axis, with the known coordinate of arc's circle center and end point, the CIRCER instruction can be used to trigger this function.

Example: When DX1.1 = On:

1. Motion card no. 0 and servo drives of node number 2 and 3 will be grouped into motion group 3.
2. The end point of motion group 3 is (current position X + 3.500, current position Y + 3.000) mm.
3. The arc' s circle center of motion group 3 is (current position X + 0.500, current position Y + 0.000) mm.
4. Motion group 3 is triggered to execute arc interpolation of two-axis (relative position).



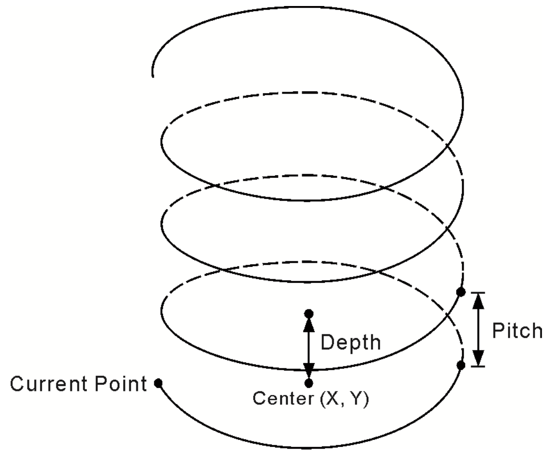
■ HELIXA

NO.	-	HELIXA	-	S1, S2	Helical motion (absolute)
118			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

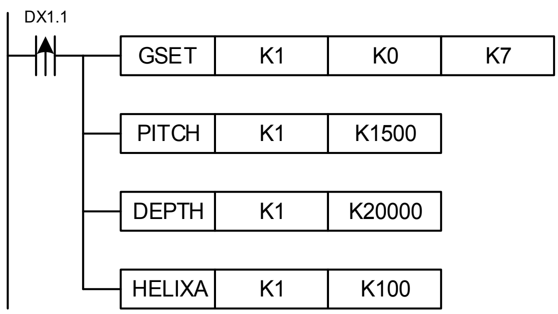
Notes when applying operand: S2 operand takes 2 consecutive devices		Instruction	Step No.
	32-bit	HELIXA	7 Step
	16-bit	-	-

Description: S1: Group No. (K1 ~ K40); S2: Target speed (default unit: PUU/s, it can be set by GUNIT instruction).
To move to the absolute position by helical interpolation of three-axis, the HELIXA instruction can be used to activate this function.



Example: When DX1.1 = On:

1. Motion card no. 0 and servo drives of node number 2 and 3 will be grouped in motion group 3.
2. Motion group 3 will upwardly rotate 1.5 mm for one rotation.
3. The target position of upward rotation for motion group 3 is 20.000 mm.
4. Motion group 3 will be triggered for executing three-axis helical motion (absolute position) with the speed of 100 PUU.



■ HELIXR

NO.	-	HELIXR	-	S1, S2	Helical motion (relative)
119			-		

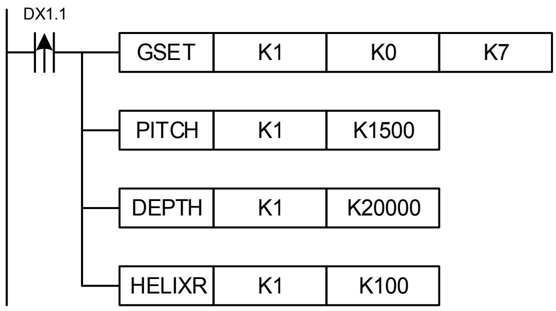
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand: S2 operand takes 2 consecutive devices		Instruction	Step No.
	32-bit	HELIXE	7 Step
	16-bit	-	-

Description: S1: Group No. (K1 ~ K40); S2: Target speed (default unit: PUU/s, it can be set by GUNIT instruction).
To move to the relative position by helical interpolation of three-axis, the HELIXR instruction can be used to activate this function.

Example: When DX1.1 = On:

1. Motion card no. 0 and servo drives of node number 2 and 3 will be grouped into motion group 3.
2. Motion group 3 will upwardly rotate 1.5 mm for one rotation.
3. The target distance of upward rotation for motion group 3 is current position + 20.000 mm.
4. Motion group 3 will be triggered to execute three-axis helical motion (relative position) with the speed of 100 PUU.



6

■ GESTP

NO. 120	-	GESTP	- / -	S1	Group emergency stop

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	GESTP	3 Step

Description: S1: Group No. (K1 ~ K40).
When GESTP instruction is executed, servo motor in S1 group will decelerate to stop with the max. deceleration speed.

Example: When DX1.1 = On, all servo axes in motion group 3 will decelerate to stop with the max. deceleration speed.

■ GSDSTP

NO. 121	-	GSDSTP	- / -	S1	Decelerate to stop for group

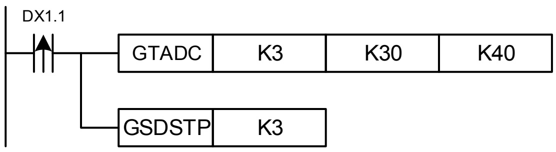
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

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, servo motor in S1 group will decelerate to stop according to the deceleration time setting.

Example: When DX1.1 = On, all servo axes in motion group 3 will decelerate to stop within the setting time 0.4 seconds.



6

6.1.6 Motion Program Macro (MPM) Control Instruction

■ MPMST

NO.	-	MPMST	-	S1	MPM starts
150			-		

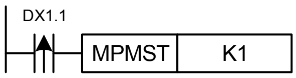
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	MPMST	3 Step

Description: S1: Group No. (K1 ~ K99).
When MPM starts, this instruction is to start the MPM prgroam.

Example: When DX1.1 = On, group 1 that executes MPM starts to run.



■ MPMSTP

NO.	-	MPMSTP	-	S1	MPM stops
151			-		

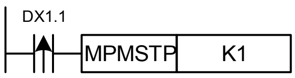
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	MPMSTP	3 Step

Description: S1: Group No. (K0 ~ K99).
MPM program stops, this instruction is to stop the MPM. When it is applied, all servo axes that controller by MPM will be stopped.

Example: When DX1.1 = On, motion group 1 that executes MPM stops running.



6

MPMPAU

NO.	-	MPMPAU	-	S1	MPM pauses
152			-		

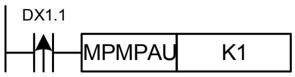
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

	Instruction	Step No.
32-bit	-	-
16-bit	MPMPAU	3 Step

Description: S1: Group No. (K0 ~ K99).
This instruction is to pause running of the MPM. And the MPM resumes when MPMST is issued.

Example: When DX1.1 = On, group 1 that executes MPM pauses.



MPMSPD

NO.	-	MPMSPD	-	S1, S2	MPM feedrate overwrite
153			-		

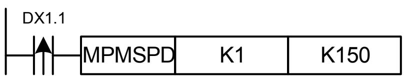
	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
S2	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-

Notes when applying operand:

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

Description: S1: Group No. (K0 ~ K99); S2: MPM speed change (0 ~ 1000 %)
To change the MPM running speed, its range should be between 0 ~ 1000%.

Example: When DX1.1 = On, the speed of MPM group 1 will be changed to 150%.



■ MPMER

NO.	-	MPMER	-	S1、D	Access MPM error code
154			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
S1	-	-	-	-	-	-	-	-	-	●	-	-	●	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-

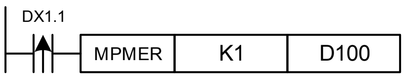
Notes when applying operand:

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

Description: S1: Group No. (K0 ~ K99).
Access the error code of MPM:

Erro code	Description
1	The servo axis is used, MPMST (MPM starts) instruction failed.
2	The group axis is not in Servo On state
3	Error occurs in group axis
4	The group axis reaches the positive limit.
5	The group axis reaches the negative limit.
6	The instruction position exceeds the range of positive limit.
7	The instruction position exceeds the range of negative limit.
8	When MPM executes homing instruction, the execution is interrupted.
10	MPM programming syntax error

Example: When DX1.1 = On, the error code will be saved in D100.



6

■ MSTEP

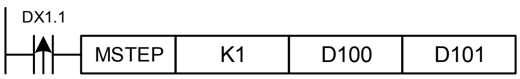
NO.	-	MSTEP	-	S1, D1, D2	Access the step No. of MPM
155			-		

	Bit Device						Word Device										External D.	
	DX	DY	M	T	C	R	KnDX	KnDY	KnM	K	T	C	D	V	Z	W	bit	word
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: Step No. that has been exectued. Access the executable step No. and the step No. that has been exectued of MPM.

Example: When DX1.1 = On, the total step No. of MPM group 1 will be saved in D100 and the step No. that has been executed will be saved in D101.



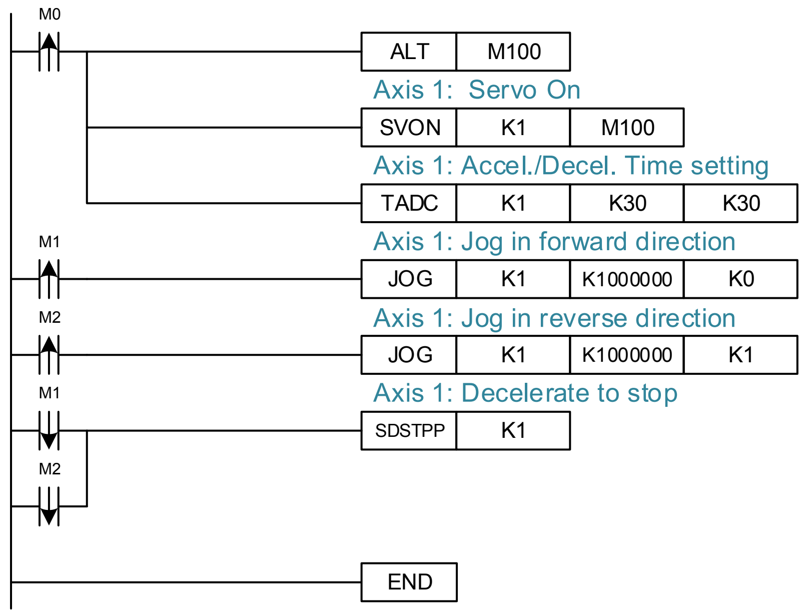
6.1.7 Example

- Jog

Description: Servo motor runs in jog mode. Press the jog (+) key when servo is on, and the motor runs in forward direction. On the other hand, press the jog (-) key, and the motor will run in the opposite direction.

Parameter definition: M0 is the bit of triggering servo on/off, M1 is the bit of controlling jog operation in forward direction and M2 is the bit of controlling jog operation in reverse direction.

Example:

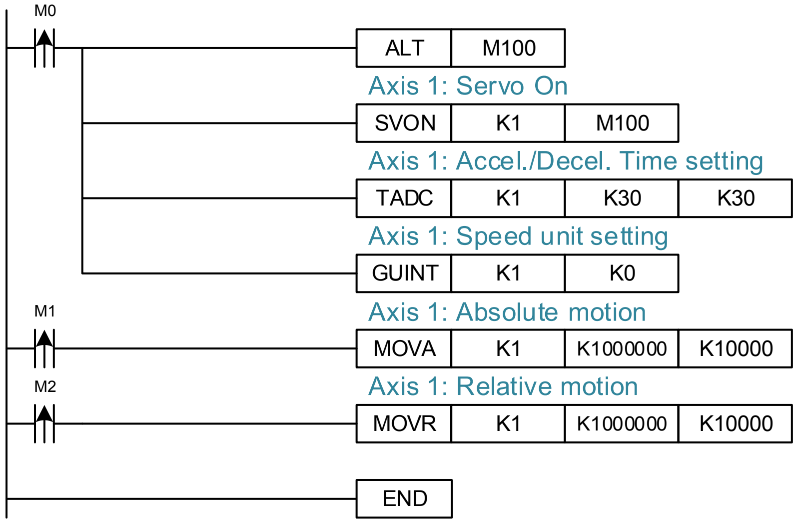


■ Single-axis motion

Description: This example is point-to-point control. M0 is used to control servo on/off and set parameters including acceleration/deceleration time as well as the speed unit of single axis. When servo is on, trigger M1 to set the absolute movement for each axis and move each axis to the target position. Then, trigger M2 to set the relative position for each axis and move each axis to the target position.

Parameter definition: M0 is the bit of triggering servo on/off, M1 is the bit of triggering point-to-point motion to absolute position and M2 is the bit of triggering point-to-point motion to relative position.

Example:



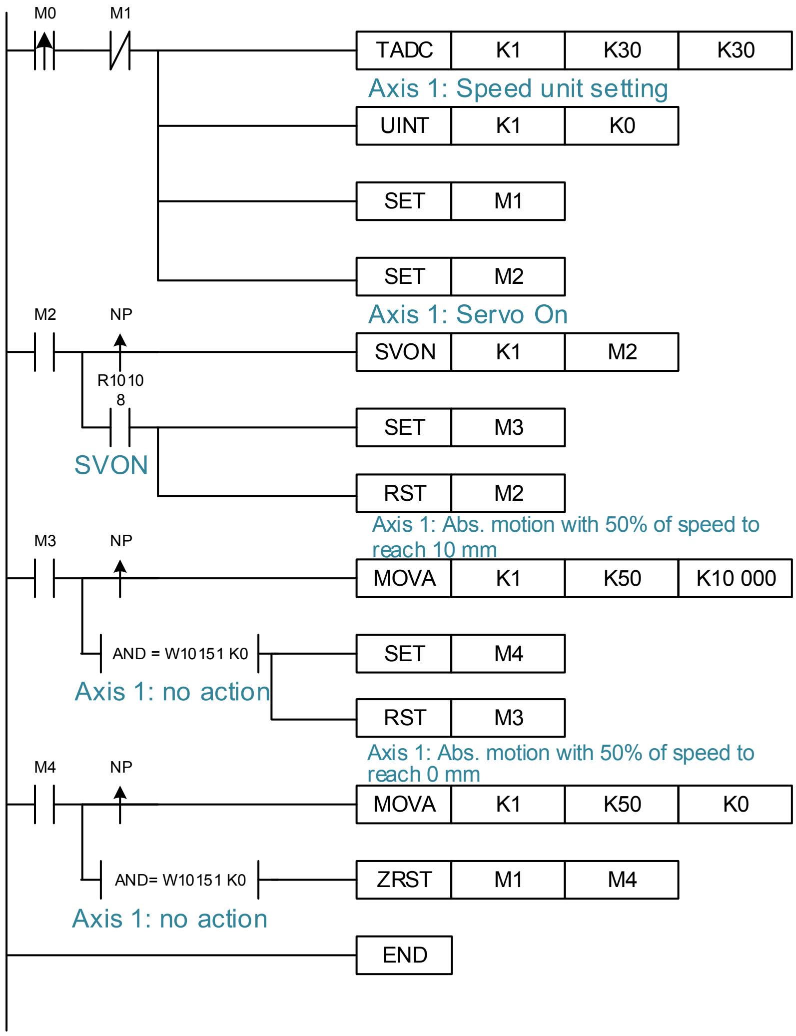
■ Point-to-point motion of single axis

Description: Here is the example of point-to-point motion. See the sequence below:

M0 Activate ⇨ M2 Servo On ⇨ M3 Move to 10 mm ⇨ M4 Move to 0 mm ⇨ Complete

Parameter definition: M0 is the bit to activate the motion; M1 is the flag and will be on when the motion is activated.

Example:

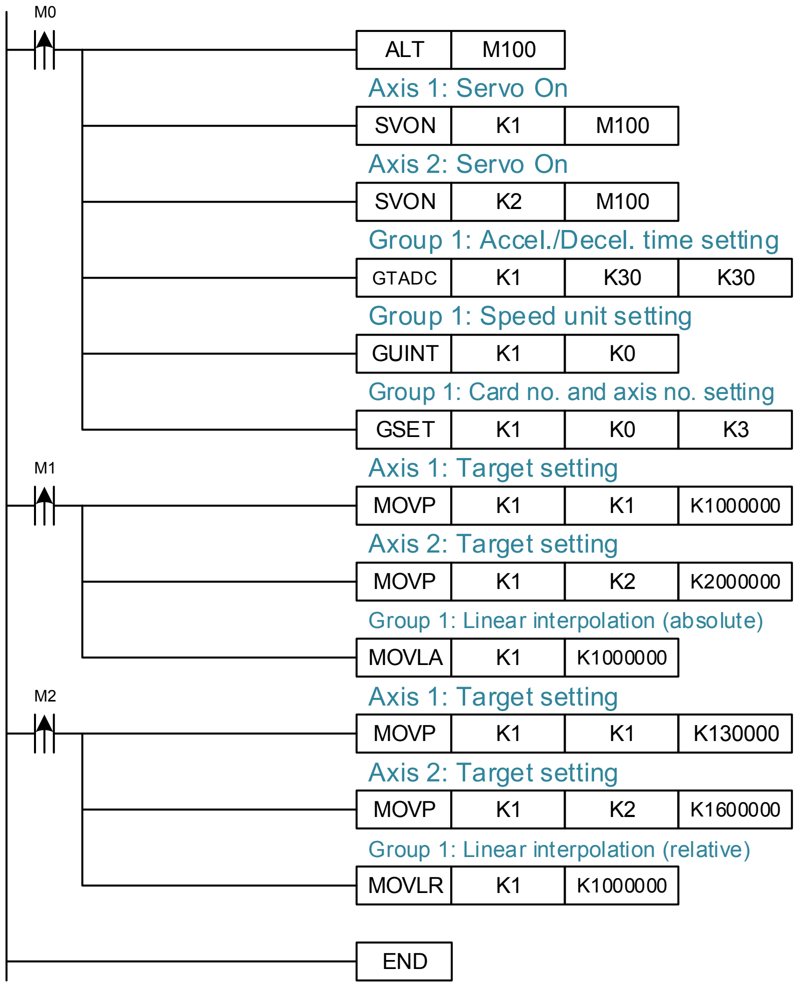


- Two-axis linear interpolation

Description: This example is the application of linear interpolation of two axes. M0 can be used to control servo on/off and set the parameters for interpolation, including acceleration/deceleration time, group speed unit and the servo axis in one group. M1 is used to set the target position of each axis and move each axis to the absolute position by linear interpolation. M2 is used to set the target position of each axis and move each axis to the relative position by linear interpolation.

Parameter definition: M0 is the bit of triggering servo on/off. M1 is the bit for group moving to the absolute position by linear interpolation. M is the bit for group moving to the relative position by linear interpolation.

Example:



■ MPG

Description: In this example, node number of ASD-DMC-RM64MN that connects to MPG is set to 9. The connected DMCNET motion card no. is 0. And the first IMP axis is the axis X controlled by MPG, the third IMP axis is the axis Y controlled by MPC and both use quadruple frequency as the input signal. This function can be used to individually control axis X or Y of MPG.

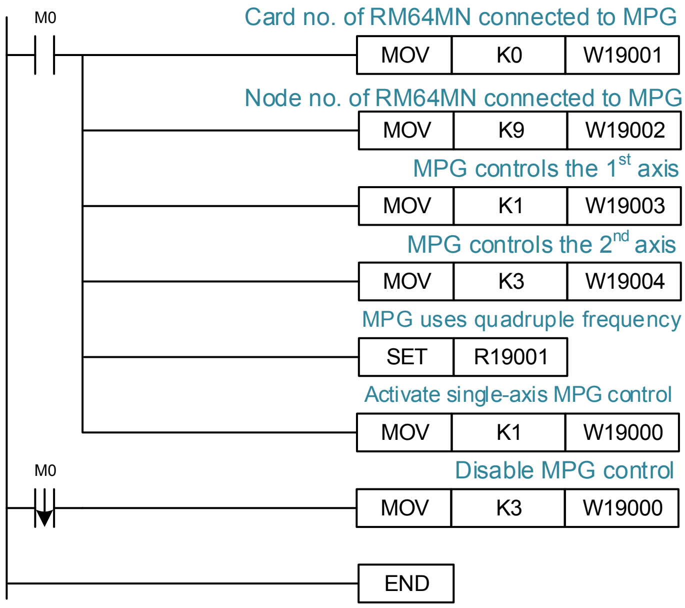
Parameter definition: M0 is the bit to enable/disable MPG.

Note: This function should work with RM64MN module. It also has to share the same DMCNET BUS as the controlled servo axis. If not, the error flag R19010 will be on after MPG is enabled.

Following is the list of MPG related special register and relay.

Position	Function	Description	Property
W19000	MPG control code	By switching the value, you can trigger the MPG functions below: 0: none 1: MPG enable 2: MPG simulation 3: MPG disable	R/W
W19001	Card number connected to MPG	1. Card number of ASD-DMC-RM64MN module that connected to MPG. 2. This setting is available only when MPG is disabled.	R/W
W19002	Node number connected to MPG	1. It is the node number of ASD-DMC-RM64MN module that connected to MPG. 2. This setting is available only when MPG is disabled.	R/W
W19003	The node number of first servo axis controlled by MPG	1. MPG can specify four controlling nodes. The connected module (ASD-DMC-RM64MN) should share the same DMCNET communication as the specified servo axis. Otherwise, flag of R19010 will be on. 2. If you do not specify the node number, MPG will be disabled. 3. This setting is available only when MPG is disabled.	R/W
W19004	The node number of second servo axis controlled by MPG		R/W
W19005	The node number of third servo axis controlled by MPG		R/W
W19006	The node number of fourth servo axis controlled by MPG		R/W
W19010	MPG running speed	MPG running speed (unit: %)	R
W19020	MPG error code	It is the returned value when error occurs. The value will be 0 when no error occurs.	R
R19001	Enable/disable quadruple frequency input of MPG	The proportion of MPG scale and output pulse On: MPG outputs 1 pulse for four scales. Off: MPG output 1 pulse for 1 scale	R/W
R19010	Error flag of MPG setting	The connected module (ASD-DMC-RM64MN) does not share the same DMCNET communication as the controlled servo axis.	R

Example:



Note: The default value of the built-in DMC card number is 0.

(This page is intentionally left blank.)

7 Motion Program Macro (MPM)

Motion Program Macro (MPM) is a macro language for IMP motion control. You can create the motion path via MPM and simplify the PLC program for motion control.

7.1 Instruction list and overview ······ 7-2

7.1.1 Application instruction ······ 7-3

7.1.2 Motion application instructions ······ 7-15

7.2 Motion Program Macro (MPM) Editor ······ 7-37

7

7.1 Instruction list and overview

This section mainly introduces the macros for IMP motion control. IMP supports 100 editable MPMs and each macro can store up to 200 instructions. There are 44 types of instructions available, including servo motion type and relevant application type. You can find more detail in section 7.1.1 and 7.1.2.

No.	Code	Function	No.	Code	Function
1	SETM	Setting the auxiliary relay	23	GTADC	Group setting of acceleration / deceleration time
2	RSTM	Resetting the auxiliary relay	24	COORD	Single-axis coordinate setting
3	CALLM	Calling 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)	To see if the bit status is On/OFF	34	MOVLA	Linear interpolation in absolute motion
13	IF(word)	Compare the word(s) content	35	MOVLR	Linear interpolation in relative motion
14	ELSE	Else	36	CIRCAA	Absolute arc motion (Arc center; angle)
15	ENDIF	End of the IF statement	37	CIRCAR	Relative arc motion (Arc center; angle)
16	DO	Start of the **DO…LOOP**	38	CIREAA	Absolute arc motion (End; angle)
17	LOOP(bit)	End of the **DO…LOOP** (bit)	39	CIREAR	Relative arc motion (End; angle)
18	LOOP(word)	End of the **DO…LOOP** (word)	40	CIRCEA	Absolute arc motion (Arc center; end)
19	WHILE(bit)	Start of the WHILE loop (bit)	41	CIRCER	Relative arc motion (Arc center; 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 interpolation in relative motion
22	GUNIT	Setting the motion speed unit	44	TAPPING	Tapping

7.1.1 Application instruction

This section will detail each MPM instruction. The application instructions include flow control, the four fundamental operations of arithmetic and data comparison.

7

How to read the table?

(1) (2)

NO.	-	Instruction	D1	Function
1				

	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	

(1) D: It supports 32-bit instructions;

(2) Operand;

(3) M: Auxiliary relay (More detail can be found in Chapter 5 <Memory Device>.)

(4) K: Constant; decimal value is started with character K. (This setting value can be an integer or a decimal.)

For example, K100 represents that this value is 100 in decimal form. K10.35 stands for the value 10.35 in decimal form. Please note that when applying PLC instructions, constants (K) can only be integers; numbers with decimal points are not allowed.

D: Data register (More detail can be found in Chapter 5 < Memory Device >.)

7

Application Instructions

■ SETM

NO. 1	-	SETM	D1			Setting 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; set operand D1 to ON.

Example: Set relay M1000 to ON.

Instruction code	Description
SETM,M1000	Set M1000 relay to ON.

■ RSTM

NO. 2	-	RSTM	D1			Resetting 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; set operand D1 to OFF.

Example: Reset relay M1000 to OFF.

Instruction code	Description
RSTM,M1000	Reset relay M1000 to OFF.

CALLM

NO. 3	-	CALLM	D1			Calling 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; this instruction can set the specified auxiliary relay to ON; after the state of the auxiliary relay is reset (this can be done by HMI, PLC and other MPM), the next instruction will be executed. This instruction can work with PLC and is issued via MPM. When PLC accesses the corresponding auxiliary relay which state is ON, the specified action will be done (such as performing functions and logic operation). Then, execute the next MPM instruction automatically.

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 reset to OFF.
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 (S1).

Example: Set the M10 relay to ON. Then, wait for 10 seconds and reset M10 to OFF.

Instruction code	Description
SETM,10	Set relay M10 to ON and the relay waits to be reset to OFF.
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:

For 16-bit instructions, D1 operand takes 2 consecutive devices.
For 32-bit instructions, D1 operand takes 4 consecutive devices.

	Instruction
32-bit	DADD
16-bit	ADD

Description: S1: summand; S2: addend; D1: sum. Add the values of register 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. The value of register D20 will be 28 after execution.

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 register 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:

For 16-bit instructions, D1 operand takes 2 consecutive devices.
For 32-bit instructions, D1 operand takes 4 consecutive devices.

	Instruction
32-bit	DSUB
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. The value of register D20 will be 2 after the operation.

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:

For 16-bit instructions, D1 operand takes 2 consecutive devices.
For 32-bit instructions, D1 operand takes 4 consecutive devices.

	Instruction
32-bit	DMUL
16-bit	MUL

Instruction: S1: multiplicand; S2: multiplier; D1: product. Multiply S2 by S1 and store the product in register D.

Example: Set registers D10 and D11 to 15 and 13 respectively. Multiply the value of D11 by D10 and store the product in registers D20 and D21. After the operation, the read value of register 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 D11 by D10 and store the product in register 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:

For 16-bit instructions, D1 operand takes 2 consecutive devices.
For 32-bit instructions, D1 operand takes 4 consecutive devices.

	Instruction
32-bit	DDIV
16-bit	DIV

Instruction: S1: dividend; S2: divisor; D1: quotient and remainder. S1 divided by S2. The quotient is stored in register D1 and remainder is stored in register D1+1.

Example: Set registers D10 and D11 to 15 and 13 respectively. D10 divided by D11. Store the quotient in register D20. After the execution, D20's value will be 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	Value of D10 divided by D11. Store the quotient in D20 and remainder in D21.

7

■ 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 source S1 to the target operand D1 and the source S1's data remains unchanged.

Example: Set the initial value of D10 to 15 and move register D10's value to D13. Then, 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, S2, S3	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) and jump to NEXT to execute the next instruction.

7

■ 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 has to be used with the FOR instruction. Please follow the rules below or an error will occur:
(1) The FOR instruction has to be issued before NEXT.
(2) The FOR instruction has to be executed with NEXT.
(3) One FOR instruction for one NEXT instruction. They have to be applied in pairs. The FOR~ NEXT loop can nest up to 10 layers. When exceeding 10, an error occurs and the program cannot be executed normally.

Example: Execute the loop for 50 times and then the value of register D11 will be 50.

Instruction code	Description
MOV,0,D11	Reset 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		To see of the bit status is ON/OFF		
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; 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 ENDIF instruction.

■ **IF(word)**

NO. 13	-	IF	S1, S2, S3		Compare the word(s) 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; S2: operator (“==”, “<=”, “>=”, “<”, and “>”); S3: device

If S2 is ==, register S1 fulfills the condition of S3 and the next instruction will be executed.

If S2 is <=, register S1’s value is smaller than or equal to S3’s value and the next instruction will be executed.

If S2 is >=, register S1’s value is greater than or equal to S3’s value and the next instruction will be executed.

If S2 is <, register S1’s value is smaller than S3’s value and the next instruction will be executed.

If S2 is >, register S1’s value is greater than S3’s value and the next instruction will be executed.

If the condition is not fulfilled, it jumps to the ELSE instruction. If there is no corresponding ELSE instruction, jumps to 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, the ELSE instruction can be used to execute other instructions (Please note that both IF and ENDIF instructions have to be applied in this circumstance.)

■ ENDIF

NO.	-	ENDIF	-	End of comparison		
15						
	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 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 M10 to OFF.
ENDIF	End of the IF instruction

■ DO

NO.	-	DO	-	Start of the **DO....LOOP**		
16						
	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 has to be applied with the LOOP instruction and is inserted before LOOP.

■ LOOP(bit)

NO.	-	LOOP	S1==S2	End of the **DO...LOOP** (bit)
17				

	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; S2: condition (ON or OFF)
If relay S1 fufills the condition of S2, execute the corresponding DO instruction; otherwise, execute the next instruction.

■ LOOP (word)

NO.	-	LOOP	S1, S2, S3	End of the **DO...LOOP** (bit)
18				

	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 (“==”, “<=”, “>=”, “<”, and “>”); S3: condition 2.
If S2 is ==, register S1 fulfills the condition of S3 and the corresponding DO instruction will be executed.
If S2 is <=, register S1’s value is smaller than or equal to S3’s value and the corresponding DO instruction will be executed.
If S2 is >=, register S1’s value is greater than or equal to S3’s value and the corresponding DO instruction will be executed.
If S2 is <, register S1’s value is smaller than S3’s value and the corresponding DO instruction will be executed.
If S2 is >, register S1’s value is greater than S3’s value and the corresponding DO instruction will be executed.
If the statement is false, the next instruction is executed.

Example: If register D10’s value plus 1 and the sum is smaller than 10, the loop will continue to be executed. Until the value is greater than 10, register D10’s value is cleared to 0.

Instruction code	Description
DO	Start of the **DO....LOOP**
ADD,D10,K1,D10	Value of register D10 plus 1.
LOOP,D10,<,K10	If value of register D10 is smaller than 10, carry on executing the loop. If greater than 10, exit the loop.
MOV,K0,D10	Reset register D10 to 0.

WHILE (bit)

NO.	-	WHILE		S1==S2		Start of the WHILE loop (bit)	
19							
	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; S2: condition (ON or OFF)
If relay S1 fulfills the condition of S2, execute the next instruction; otherwise, execute the corresponding ENDWHILE instruction.

WHILE (word)

NO.	-	WHILE		S1, S2, S3		Start of the WHILE loop (word)	
20							
	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 (“==”, “<=”, “>=”, “<”, and “>”); S3: condition 2.
If S2 is ==, register S1 fulfills the condition of S3 and the next instruction will be executed.
If S2 is <=, register S1’s value is smaller than or equal to S3’s value and the next instruction will be executed.
If S2 is >=, register S1’s value is greater than or equal to S3’s value and the next instruction will be executed.
If S2 is <, register S1’s value is smaller than S3’s value and the next instruction will be executed.
If S2 is >, register S1’s value is greater than S3’s value and the next instruction will be executed.
If the statement is false, the ENDWHILE instruction will be executed.

■ ENDWHILE

NO.	-	ENDWHILE	-	End of the WHILE loop
21				

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 has to be applied with the WHILE instruction.

Example: If value of register D10 is smaller than or equal to 100, execute the instructions in the loop. Then, value of register D10 will increase by 1 every time the loop is executed and stop increasing when the value is over 100. The loop stops when register value is over 100 and D10’s value stops at 101.

Instruction code	Description
MOV,0,D10	Set register D10 to 0.
WHILE,D10,<=,100	If value of register D10 is smaller than or equal to 100, execute the content of the loop.
ADD,D10,1,D10	Value of register D10 plus 1.
ENDEHILE	End of the loop

7

7.1.2 Motion application instructions

This section will detail the application instructions of MPM. These instructions are for servo motion controls, including operation with fixed torque, single-axis motion, and multi-axis interpolation. The table below tells you how to read the data from the instruction table.

How to read the table?

(1) (2)

NO. / 1	-	Instruction	D1	Function

(3)

	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

(1) D: It supports 32-bit instructions;

(2) Operand;

(3) M: Auxiliary relay (Details can be found in chapter 5 <Memory Device>.)

(4) K: Constant; decimal value is started with character K. (The setting value here can be a integer or a decimal.)
Take K100 for example, K stands for decimal form and the value is 100; K10.35 represents that the value is 10.35 in decimal form. When using PLC instructions, constant (K) can only be integers.
D: data register (Details can be found in Chapter 5 <Memory Device>.)

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 speed unit of the MPM group and execute the next instruction. The speed unit set by this instruction is only valid when MPM executes the motion command. Please note that this instruction's valid range is different from GUNIT and UNIT of PLC.

Example: Set the speed unit of MPM to puu/s. Move to the X-axis absolute position 10 mm at 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 10 mm at 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 or S1 applies register D, the last three digits will be decimals. For example, if value of register D100 is K35997 and when GTADC instruction is executed, the read value will be 35.997.

	Instruction
32-bit	GTADC
16-bit	-

Description: S1: acceleration time (unit: s), the time it takes when accelerating to the max. speed; S2: deceleration time (unit: s), the time is takes when decelerating from the max. speed to stop. Set the acceleration/deceleration time of the MPM group and execute the next instruction. The acceleration/deceleration time set by this instruction is only valid when the interpolation instruction is executed. Please note that the valid range of this instruction is different from GTADC and TADC instruction of PLC. Note: Please refer to Chapter 6 <Logic Editing> for descriptions about setting acceleration/deceleration time.

Example: Set the speed unit of MPM to puu/s. The servo motor accelerates to 1000000 puu/s. Then, it starts to decelerate before reaching the X-axis' absolute position 10 mm. (Acceleration: motor reaches the max. speed set by the system within 0.3 seconds; Deceleration: motor decelerates from the max. speed set by the system to stop within 0.4 seconds.)

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
GTADC,0.3,0.4	Set the acceleration and deceleration time of the group to 0.3 seconds and 0.4 seconds respectively.
MOVA,X,1000000,10	Move to the absolute position 10 mm with the speed of 1000000 puu/s.

■ COORD

NO.	-	COORD	S1, S2	Single-axis coordinate setting			
24							
	Bit Device		Word Device			Code and Symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	●	●	-	-	-

Notes when applying operand:		Instruction
If S2 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the read value will be 35.997.	32-bit	COORD
	16-bit	-

Description: S1: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 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 speed unit of MPM to puu/s, move to the absolute position 10 mm of the X-axis at 1000000 puu/s, and change the target position into 20 mm.

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s
MOVA,X,1000000,10	Move to the absolute position 10 mm at 1000000 puu/s.
COORD,X,20	Set the target position of X-axis to 20 mm.

7

SPD

NO. 25	-	SPD	S1, S2	Single-axis motion in speed mode		
	Bit Device		Word Device			Code and Symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	●	●	-	-	-

Notes when applying operand:

If S2 applies register D, the last three digits are decimals. For instance, if value of register D100 is K35997, the read value will be 35.997.

	Instruction
32-bit	SPD
16-bit	-

Description: S1: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis; S2: motion speed (unit: rpm). Trigger axis S1 to operate at speed specified by S2 and execute the next instruction. This instruction automatically stops when execution of MPM group is completed.

Example: Set the speed unit of MPM to puu/s. The axis X runs at 100 rpm for 10 seconds. Then, the MPM is completed and this axis stops running.

Instruction code	Description
GUNIT,0	Set motion speed unit to puu/s.
SPD,X,100	Motor runs at fixed speed 100 rpm.
DELAY,10000	Delay for 10000 ms (10 s).

TRQ

NO. 26	-	TRQ	S1, S2	Single-axis motion in torque mode.		
	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: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis; S2: torque (unit: ‰) Trigger axis S1 to run with the torque specified by S2 and then execute the next instruction. This instruction automatically stops when execution of the MPM group is completed.

Example: Set the speed unit of MPM to puu/s. The axis X runs with the max. torque 150‰ for 10 seconds. Then, this MPM ends and the axis stops running.

Instruction code	Description
GUNIT,0	Set motion speed unit to puu/s.
TRQ,X,150	The servo motor runs with the max. torque 150‰.
DELAY,10000	Delay for 10000 ms.

7

■ SDSTP

NO. 27	-	SDSTP	S1	Single-axis deceleration to stop			
	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: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis. The servo axis specified by S1 decelerates at the speed set by GTADC instruction, and executes the next instruction.

Example: Set the motion speed unit of MPM to puu/s. The 1st axis of the group (axis X) runs at fixed speed 100 rpm for 10 seconds, and decelerates to stop within 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 time to 0.3 seconds and 0.4 seconds respectively.
SPD,X,100	The axis X runs at fixed speed100 rpm.
DELAY,10000	Delay for 10000 ms.
SDSTP,X	The axis X decelerates to stop within 0.4 seconds.
MOVA,X,100000,10	Move to the absolute position 10 mm at 100000 puu/s.

■ ESTP

NO. 28	-	ESTP	S1	Single-axis emergency stop			
	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: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis. The servo axis specified by S1 decelerates to stop at the max. speed, and executes the next instruction.

Instruction code	Description
GUNIT,0	Set the motion speed unit to puu/s.
GTADC,0.3,0.4	Set the group acceleration and deceleration time to 0.3 seconds and 0.4 seconds respectively.

7

Example: Set the motion speed unit of MPM to puu/s. The 1st axis of the group (axis X) runs at fixed speed 100 rpm for 10 seconds, and decelerates to stop at the max. speed. Then, it moves to the the absolute position 10 mm.

SPD,X,100	The axis X runs at fixed speed100 rpm.
DELAY,10000	Delay for 10000 ms.
ESTP,X	The axis X decelerates to stop at the max. speed.
MOVA,X,100000,10	Move to the absolute position 10 mm at 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: All servo axes specified by MPM decelerate to stop within the time set by the GTADC instruction, and execute the next instruction.

Example: Set the motion speed unit of MPM to puu/s. The axis X and Y, specified by MPM, run at fixed speed 100 rpm for 10 seconds, and decelerate to stop within 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 time to 0.3 seconds and 0.4 seconds respectively.
SPD,X,100,Y,100	The axis X and Y run at fixed speed100 rpm.
DELAY,10000	Delay for 10000 ms.
GSDSTP	All axes specified by MPM decelerate to stop within 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: All servo axes specified by MPM decelerate to stop at the max. speed, and execute the next instruction.

Example: Set the speed unit of MPM to puu/s. The axis X and Y, specified by MPM, run at fixed speed 100 rpm for 10 seconds, and decelerate to stop at the max. 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 time to 0.3 seconds and 0.4 seconds respectively.
SPD,X,100,Y,100	The axis X and Y run at fixed speed100 rpm.
DELAY,10000	Delay for 10000 ms.
GESTP	All axes set by MPM decelerate to stop at the max. speed.

7

HOME

NO.	-	HOME	S1, S2, S3, S4, S5	Homing
31				

	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:		Instruction
If S3, S4, or S5 applies register D, the last three digits are decimals.	32-bit	HOME
For instance, if value of D100 is K35997, the value read by HOME instruction will be 35.997.	16-bit	-

Description: S1: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis; S2: Homing methods (please refer to Chapter 6 <Logic Editing> for descriptions about homing instructions); S3: the first homing speed (unit: rpm); S4: the second homing speed (unit: rpm); S5: offset (unit: mm). When executing HOME instruction, the assigned servo axis S1 looks for the origin with the homing method specified by S2. And change the servo axis coordinates into the coordinates specified by S5, and then executes the next instruction. Note: During the homing process, changing speed or executing Stop instruction will cause MPM to end unexpectedly.

Example: The 1st axis of the group (axis X) executes the homing process with method 34, and when completed, it moves to the absolute position 20 mm at 50% of the max. speed.

Note: Homing method 34 is to look for encoder Z pulse in reverse direction.

Instruction code	Description
GUNIT,0	Set the motion speed unit to percentage.
HOME,X,34,100,200,20	The axis X executes the homing process with method 34. The first speed is 100 rpm; the second speed is 200 rpm. After completed the homing, set the new home origin as absolute position 20 mm.
MOVA,X,50,20	The axis X moves to the absolute position 20 mm at 50% of the max. speed.

■ MOVA

NO.	-	MOVA	S1, S2, S3	Single-axis in absolute motion		
32						
	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 or S3 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by MOVA instruction is 35.997.	32-bit	MOVA
	16-bit	-

Description: S1: X: the 1^{st} axis, Y: the 2^{nd} axis, Z: the 3^{rd} axis, U: the 4^{th} axis, V: the 5^{th} axis, W: the 6^{th} axis; S2: speed (default unit: puu/s, same as the PLC setting); S3: target position (unit: mm). When executing 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, the next instruction is executed.

Example: Set the motion speed unit to percentage. The axis X and axis Y start moving simutaneourly, and when reaching the absolute position 200 mm and 300 mm respectively, the axis X returns to the absolute position 0 mm.

Instruction code	Description
GUNIT,0	Set the motion speed unit to percentage.
MOVA,X,50,200,Y,60,300	The axis X moves to the absolute position 200 mm at 50% of the max. speed, and the axis Y moves to the absolute position 300 mm at 60% of the max. speed.
MOVA,X,50,20	The axis X moves to the absolute position 0 mm at 50% of the max. speed.

■ MOVR

NO.	-	MOVR	S1, S2, S3	Single-axis in relative motion		
33						
	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 or S3 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by MOVR instruction is 35.997.	32-bit	MOVR
	16-bit	-

Description: S1: X: the 1^{st} axis, Y: the 2^{nd} axis, Z: the 3^{rd} axis, U: the 4^{th} axis, V: the 5^{th} axis, W: the 6^{th} axis; S2: speed (default unit: puu/s, same as the PLC setting); S3: incremental distance (unit: mm). When executing MOVR instruction, the servo axis specified by S1 runs at the speed set by S2, and stops when reaching the coordinates of (current point + S3).

7

Example: Set the motion speed unit to percentage. Starting at the same time, the axis X moves to the target position (its current point + 200 mm, and the axis Y moves to the target position (its current point + 300 mm).

Instruction code	Description
GUNIT,1	Set the motion speed unit to percentage.
MOVR,X,50,200,Y,60,300	The axis X moves to the absolute position 200 mm at 50% of the max. speed, and the axis Y moves to the absolute position 300mm at 60% of the max. speed.

■ MOVLA

NO.	-	MOVLA	S1, S2, S3		Linear interpolation in absolute motion		
34							
	Bit Device		Word Device			Code and Symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	●	●(DW)	-	-	-
S2	-	-	-	-	-	●	-
S3	-	-	●	●(DW)	-	-	-

Notes when applying operand:		Instruction
If S1 or S3 applies register D, the last three digits are decimals; For instance, if value of D100 is K35997, the value read by MOVLA instruction is 35.997.	32-bit	MOVLA
	16-bit	-

Description: S1: the max. speed (default unit: puu/s; works with GUNIT instruction); S2: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis; S3: target position (unit: mm). The servo axis specified by S2 moves to the target position S3 at the max. speed set by S1.

Example: Set the motion speed unit to percentage. The axis X and axis Y conduct two-axis interpolation at 50% of the max. speed, and respectively reach the absolute position 200 mm and 300 mm at the same time. Then, this MPM ends.

Instruction code	Description
GUNIT,0	Set the motion speed unit to percentage.
MOVLA,50,X,200,Y,300	The axis X and axis Y conduct interpolation at 50% of the max. speed, and respectively reach the absolute position 200 mm and 300 mm at the same time .

7

■ MOVLR

NO. 35	-	MOVLR	S1, S2, S3	Linear interpolation in relative 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 or S3 applies register D, the last three digits are decimals; For instance, if value of D100 is K35997, the value read by MOVLR instruction is 35.997.

	Instruction
32-bit	MOVLR
16-bit	-

Description: S1: the max. speed (default unit: puu/s; the unit can be adjusted with GUNIT instruction); S2: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis; S3: incremental distance (unit: mm). The servo axis specified by S2 moves to the target position (current point + S3) at the max. speed set by S1.

Example: Set the motion speed unit to percentage. The axis X and axis Y conduct two-axis interpolation at 50% of the max. speed, and simultaneously, the axis X reaches the target position (its current point + 200 mm), and the axis Y reaches the target position (its current point + 300 mm). Afterwards, this MPM ends.

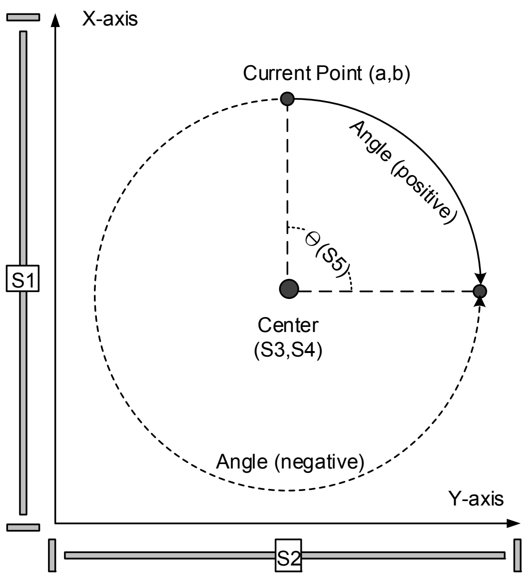
Instruction code	Description
GUNIT,0	Set the motion speed unit to percentage
MOVLR,50,X,200,Y,300	The axis X and axis Y conduct interpolation at 50% of the max. speed, and respectively reach the absolute position 200 mm and 300 mm at the same time.

■ CIRCAA

NO. 36	-	CIRCAA	S1, S2, S3, S4, S5, S6	Absolute arc 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, or S6 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by CIRCAA is 35.997.	32-bit	CIRCAA
	16-bit	-

Description: S1: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;
S2: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;
S3: the X coordinate of the center (unit: mm);
S4: the Y coordinate of the center (unit: mm);
S5: angle (unit: degree);
S6: the max. speed (default unit: puu/s; the unit can be changed by GUNIT instruction).



■ CIRCAR

NO.	-	CIRCAR	S1, S2, S3, S4, S5, S6	Relative arc motion (center, angle)
37				

	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, or S6 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by CIRCAR instruction is 35.997.	32-bit	CIRCAR
	16-bit	-

Description: S1: X: the 1^{st} axis, Y: the 2^{nd} axis, Z: the 3^{rd} axis, U: the 4^{th} axis, V: the 5^{th} axis, W: the 6^{th} axis;

S2: X: the 1^{st} axis, Y: the 2^{nd} axis, Z: the 3^{rd} axis, U: the 4^{th} axis, V: the 5^{th} axis, W: the 6^{th} axis;

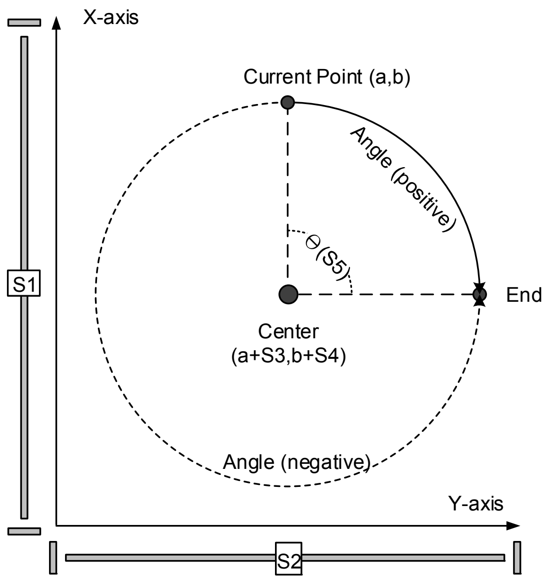
S3: relative distance to the center (X-axis, unit: mm);

S4: relative distance to the center (Y-axis, unit: mm);

S5: angle (unit: degree);

S6: the max. speed (default unit: puu/s; the unit can be changed by GUNIT instruction).

S1 specifies the servo axis moving along X-axis, and S2 specifies the servo axis moving along Y-axis. The center’s X coordinate is its current point + S3, and Y coordinate is its current point + S4. The included angle between the current point and the arc end is S5. The servo axes, which are specified by S1 and S2 respectively, move at the max. speed set by S6.



■ CIREAA

NO.				
38	-	CIREAA	S1, S2, S3, S4, S5, S6	Absolute arc 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, or S6 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by CIREAA instruction is 35.997.	32-bit	CIREAA
	16-bit	-

Description: S1: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;

S2: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;

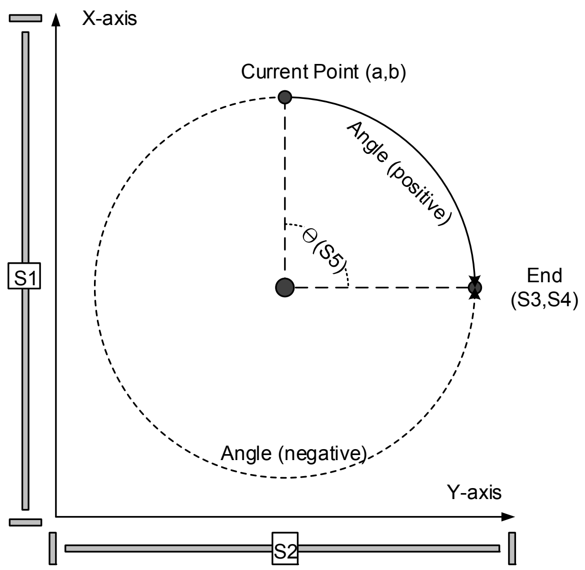
S3: the X coordinate of the arc end (unit: mm);

S4: the Y coordinate of the arc end (unit: mm);

S5: angle (unit: degree);

S6: the max. speed (default unit: puu/s; the unit can be changed by GUNIT instruction).

S1 specifies the servo axis moving along X-axis, and S2 specifies the servo axis moving along Y-axis. The X coordinate of the arc end is S3, Y coordinate is S4. The included angle between the current point and the arc end is S5. The servo axes, which are specified by S1 and S2 respectively, move at the max. speed set by S6.



7

■ CIREAR

NO. 39	-	CIREAR	S1, S2, S3, S4, S5, S6	Relative arc 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, or S6 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by CIREAR instruction is 35.997.	32-bit	CIREAR
	16-bit	-

Description: S1: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;

S2: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;

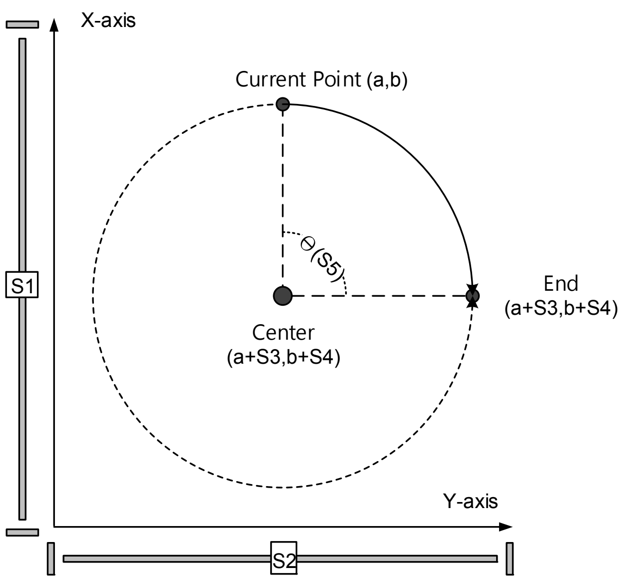
S3: relative distance to the arc end (X-axis, unit: mm);

S4: relative distance to the arc end (Y-axis, unit: mm);

S5: angle (unit: degree);

S6: the max. speed (default unit: puu/s; the unit can be changed by GUNIT instruction).

S1 specifies the servo axis moving along X-axis, and S2 specifies the servo axis moving along Y-axis. The X coordinate of the arc end is its current point + S3, Y coordinate is its current point + S4. The included angle between the current point and the arc end is S5. The servo axes, which are specified by S1 and S2 respectively, move at the max. speed set by S6.



■ CIRCEA

NO.	-	CIRCEA	S1, S2, S3, S4, S5, S6, S7, S8	Absolute arc motion (center, end)
40				

	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, or S8 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by CIRCEA instruction is 35.997.	32-bit	CIRCEA
	16-bit	-

Description: S1: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;

S2: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;

S3: the X coordinate of the center (unit: mm);

S4: the Y coordinate of the center (unit: mm);

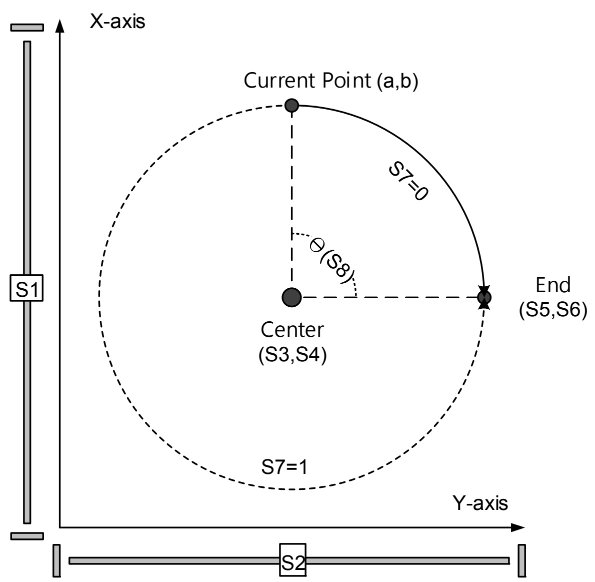
S5: the X coordinate of the arc end (unit: mm);

S6: the Y coordinate of the arc end (unit: mm);

S7: direction (0: CW; 1: CCW);

S8: the max. speed (default unit: puu/s; the unit can be changed by GUNIT instruction).

S1 specifies the servo axis moving along X-axis, and S2 specifies the servo axis moving along Y-axis. The X coordinate of the center is S3 and Y coordinate is S4. The X coordinate of the arc end is S5 and Y coordinate is S6. The servo axes, which are specified by S1 and S2 respectively, move at the max. speed set by S8.



Example: The 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,0	Set the motion speed unit to percentage
MOVA,X,50,-15,Y,50,15	The 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. 41	-	CIRCER	S1, S2, S3, S4, S5, S6, S7, S8	Relative arc motion (center, 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:

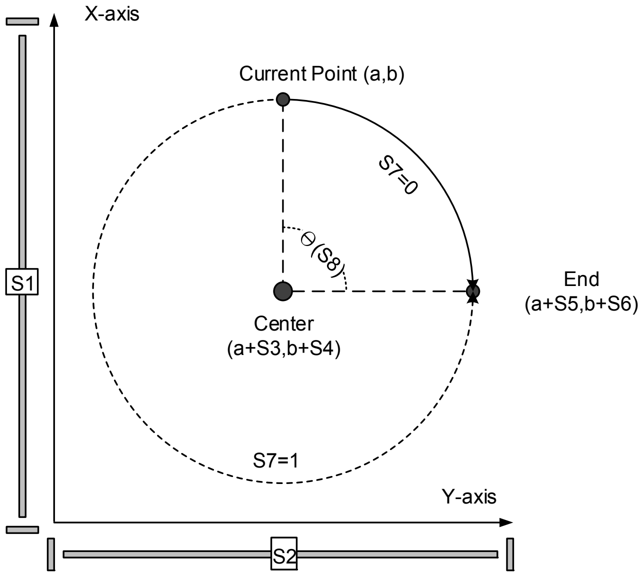
If S3, S4, S5, S6, or S8 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by CIRCER instruction is 35.997.

	Instruction
32-bit	CIRCER
16-bit	-

Description: S1: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;
S2: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 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: the max. speed (default unit: puu/s; the unit can be changed by GUNIT instruction).

7

S1 specifies the servo axis moving along X-axis, and S2 specifies the servo axis moving along Y-axis. The X coordinate of the center is its current point + S3, and Y coordinate is its current point + S4. The X coordinate of the arc end is its current point + S5, and Y coordinate is its current point + S6. The servo axes, which are specified by S1 and S2 respectively, move at the max. speed set by S8.



Example: The 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, stop at the position (15,15), and then the MPM ends.

Instruction code	Description
GUNIT,0	Set the speed unit to percentage
MOVA,X,50,-15,Y,50,15	The 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:

If S4, S5, S6, S7, or S9 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by HELIXA instruction is 35.997.

	Instruction
32-bit	HELIXA
16-bit	-

Description: S1: X: the 1^{st} axis, Y: the 2^{nd} axis, Z: the 3^{rd} axis, U: the 4^{th} axis, V: the 5^{th} axis, W: the 6^{th} axis;

S2: X: the 1^{st} axis, Y: the 2^{nd} axis, Z: the 3^{rd} axis, U: the 4^{th} axis, V: the 5^{th} axis, W: the 6^{th} axis;

S3: X: the 1^{st} axis, Y: the 2^{nd} axis, Z: the 3^{rd} axis, U: the 4^{th} axis, V: the 5^{th} axis, W: the 6^{th} axis;

S4: the X coordinate of the helix center (unit: mm);

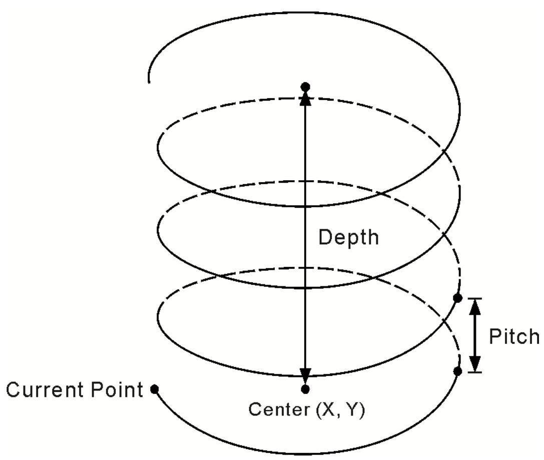
S5: the Y coordinate of the helix center (unit: mm);

S6: Helix depth: the overall rising height (unit: mm);

S7: Helix pitch: the distance between two tuns of arc (unit: mm);

S8: direction (0:CW; 1:CCW);

S9: the max. speed (default unit: puu/s; the unit can be changed by GUNIT instruction); S1 specifies the servo axis moving along X-axis, S2 specifies the servo axis moving along Y-axis, and S3 specifies the servo axis moving along Z-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, which are specified by S1 and S2 respectively, apply S9 as the linear speed to calculate the speed of each axes.



7

Note: the speed of S1 and S2 are calculated according to the max. speed (S9), but 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: The axis X and axis Y move to the position (-15, 15), and they make 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, MPM ends.

Instruction code	Description
GUNIT,0	Set the motion speed unit to percentage
MOVA,X,50,-15,Y,50,15	The axis X and axis Y move to the position of (-15, 15).
HELIXA,X,Y,Z,0,0,100,10,0,50	Move in circular motion

■ **HELIXR**

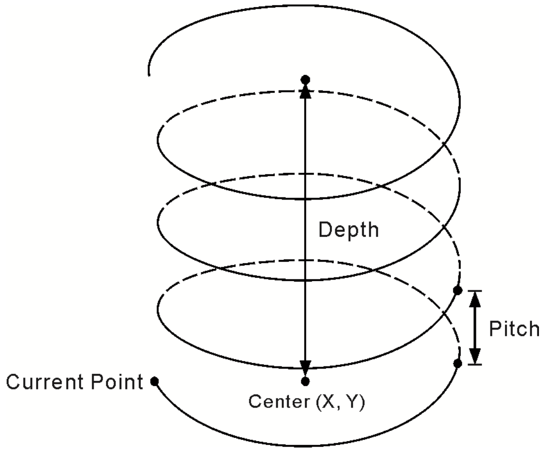
NO.	-	HELIXR	S1, S2, S3, S4, S5, S6, S7, S8, S9	Three-axis interpolation in relative motion			
43							
	Bit Device		Word Device			Code and Symbol	
	M	-	K	D	E	Axis No.	Operator
S1	-	-	-	-	-	●	-
S2	-	-	-	-	-	●	-
S3	-	-	-	-	-	●	-
S4	-	-	●	●(DW)	-	-	-
S5	-	-	●	●(DW)	-	-	-
S6	-	-	●	●(DW)	-	-	-
S7	-	-	●	●(DW)	-	-	-
S8	-	-	●	●	-	-	-
S9	-	-	●	●(DW)	-	-	-

Notes when applying operand:		Instruction
If S4, S5, S6, S7, or S9 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by HELIXR is 35.997.	32-bit	HELIXR
	16-bit	-

Description: S1: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;
S2: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;
S3: X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;
S4: relative distance to the helix center (X-axis) (unit: mm);
S5: relative distance to the helix center (Y-axis) (unit: mm);
S6: Helix depth: the overall rising height (unit: mm);
S7: Helix pitch: the distance between two tuns of arc (unit: mm);
S8: direction (0: CW; 1: CCW) S9: the max. speed (default unit: puu/s; the unit can be changed by GUNIT instruction);
S1 specifies the servo axis moving along X-axis, S2 specifies the servo axis moving along Y-axis, and S3 specifies the servo axis moving along Z-axis. The X coordinate of the helix center is its current point + S4, the Y coordinate is its current point + S5. The overall helix depth is S6, the helix pitch is S7, and the direction is S8. The servo axes, which are specified by S1 and S2 respectively, apply S9 as the linear speed to

7

calculate the speed of each axes.



Note: the speed of S1 and S2 are calculated according to the max. speed (S9), whereas 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: The axis X and axis Y move to the position (-15, 15), and move in arc around the center at (0,0) in clockwise direction. And they elevate by 10 mm every turn and stop moving when reaching 100 mm on Z-axis. Then, this MPM ends.

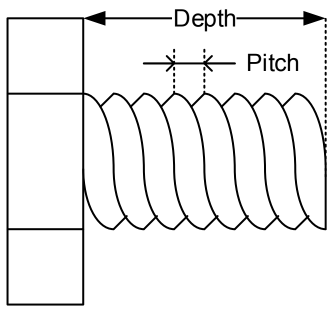
Instruction code	Description
GUNIT,0	Set the speed unit to percentage
MOVA,X,50,-15,Y,50,15	The axis X and axis Y move to the position of (-15, 15).
HELIXA,X,Y,Z,15,-15,100,10,0,50	Move in circular motion

■ **TAPPING**

NO. 44	-	TAPPING	S1, S2, S3, S4, S5, S6, S7, S8	Tapping			
	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 or S4 applies register D, the last three digits are decimals. For instance, if value of D100 is K35997, the value read by TAPPING instruction is 35.997.	32-bit	TAPPING
	16-bit	-

Description: S1: Rotation axis X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis;
S2: Feeding axis X: the 1st axis, Y: the 2nd axis, Z: the 3rd axis, U: the 4th axis, V: the 5th axis, W: the 6th axis; S3: tapping depth (unit: mm);
S4: tapping pitch (unit: mm);
S5: tapping speed (unit: rpm);
S6: extracting speed (unit: rpm);
S7: extracting delay (unit: s);
S8: clockwise/ counterclockwise (0: CW; 1: CCW);
This instruction is for TAPPING only. The feeding axis S2 executes the feeding according to the tapping speed S5 and pitch S4, and stops when reaching the set value of S3. After staying for the time set by S7, it moves back to the origin at the extracting speed set by S6.



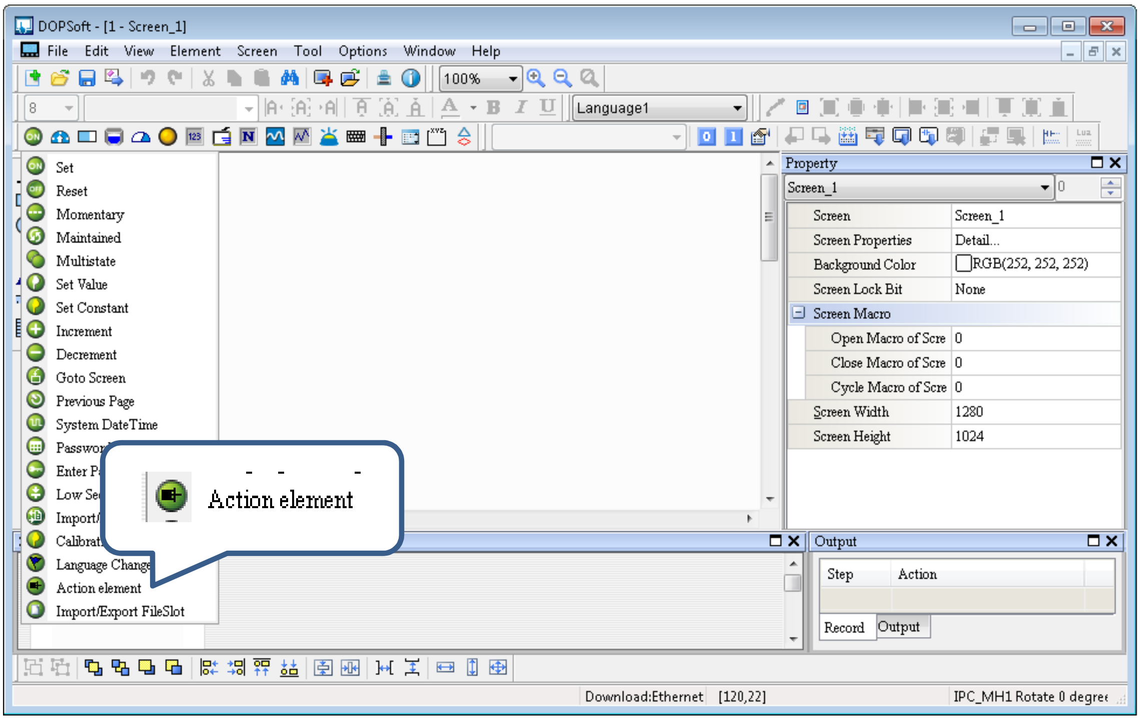
Example: The tapping speed of rotation axis X is 100 rpm, and at each turning cycle of the axis X, the corresponding axis Y feeds 7 mm (pitch). The total feeding distance of axis Y is 70 mm. When the tapping finishes, the axis X and Y decelerate to stop, delay for 65 ms, and then carry out the extraction at 160 rpm in reverse direction.

Instruction code	Description
GUNIT,0	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 program
SETM,0	Set relay M0 to ON.

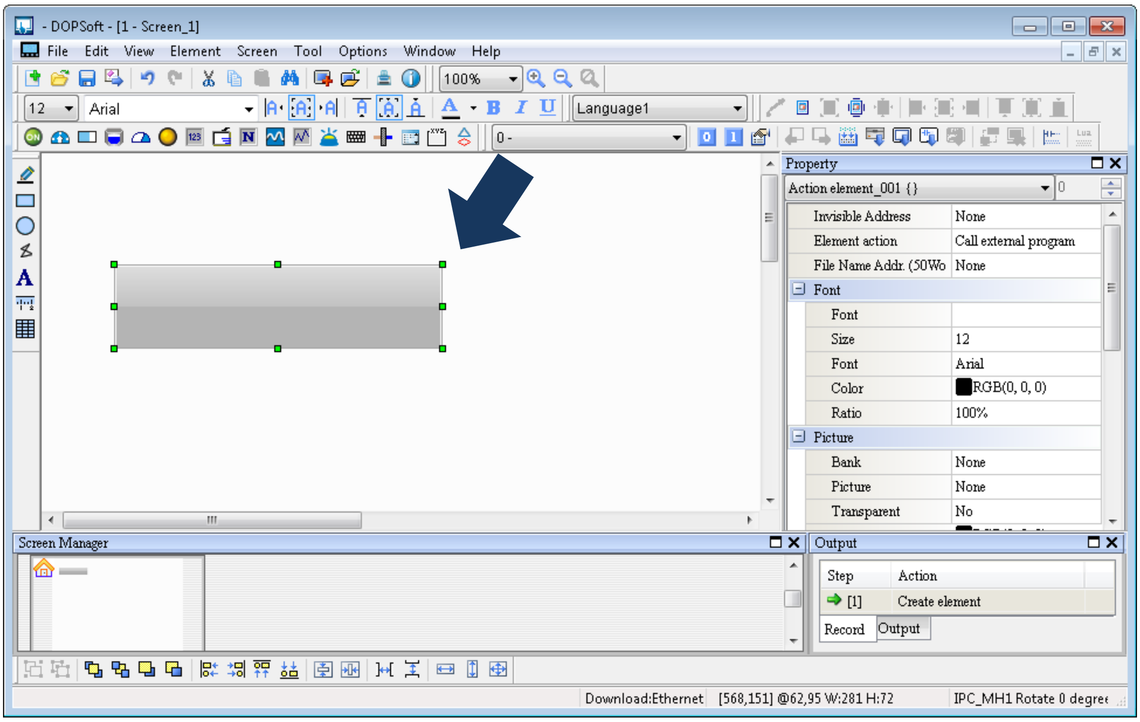
7.2 Motion Program Macro (MPM) Editor

■ **Use SOFTHMI software to call MPM editor**

1. Add Action element in the editing screen of DOPSOFT3.00.01.

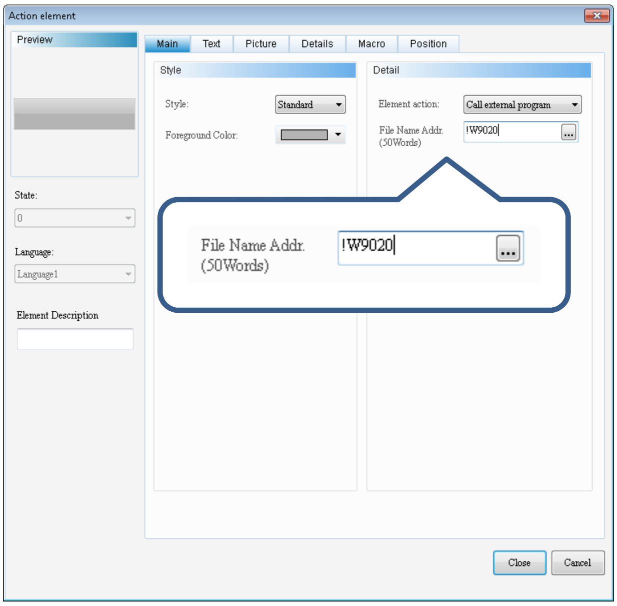


2. Draw an element.

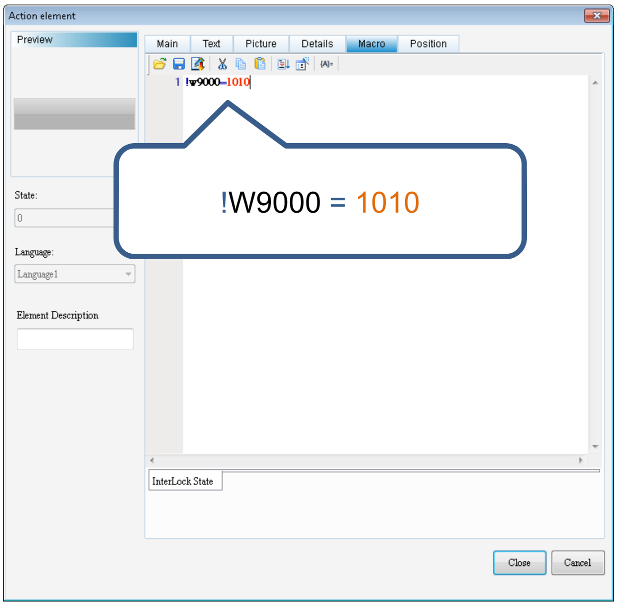


7

3. Double click on the element and enter “!W9020” for **File Name Address**.

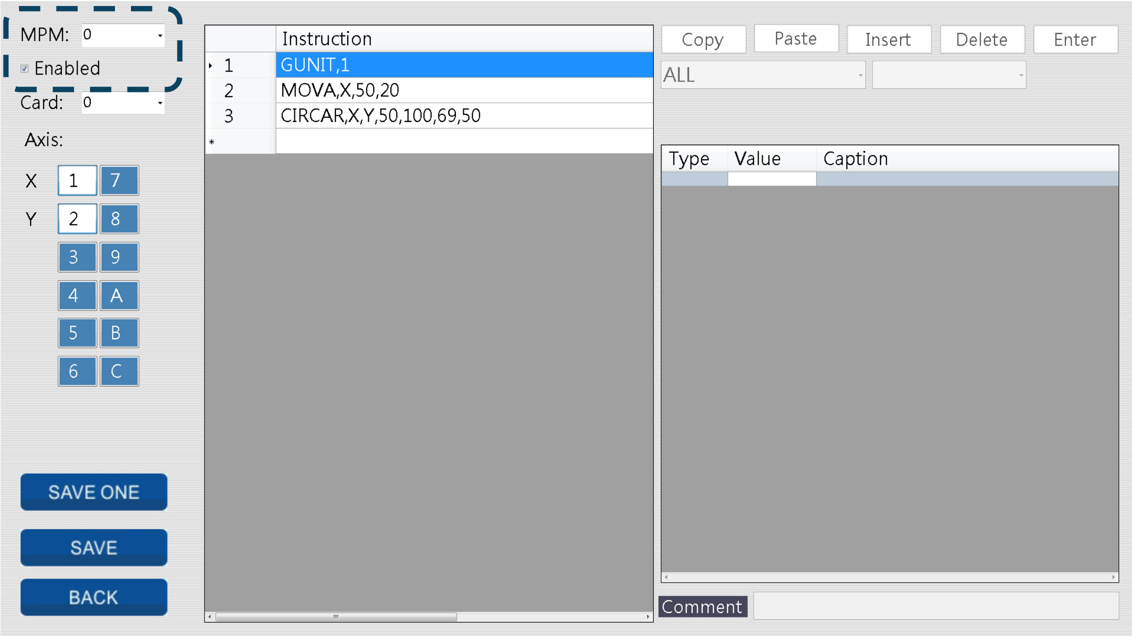


4. Click on **Macro** and enter “!W9000=1010”. Then, click on **Close**.

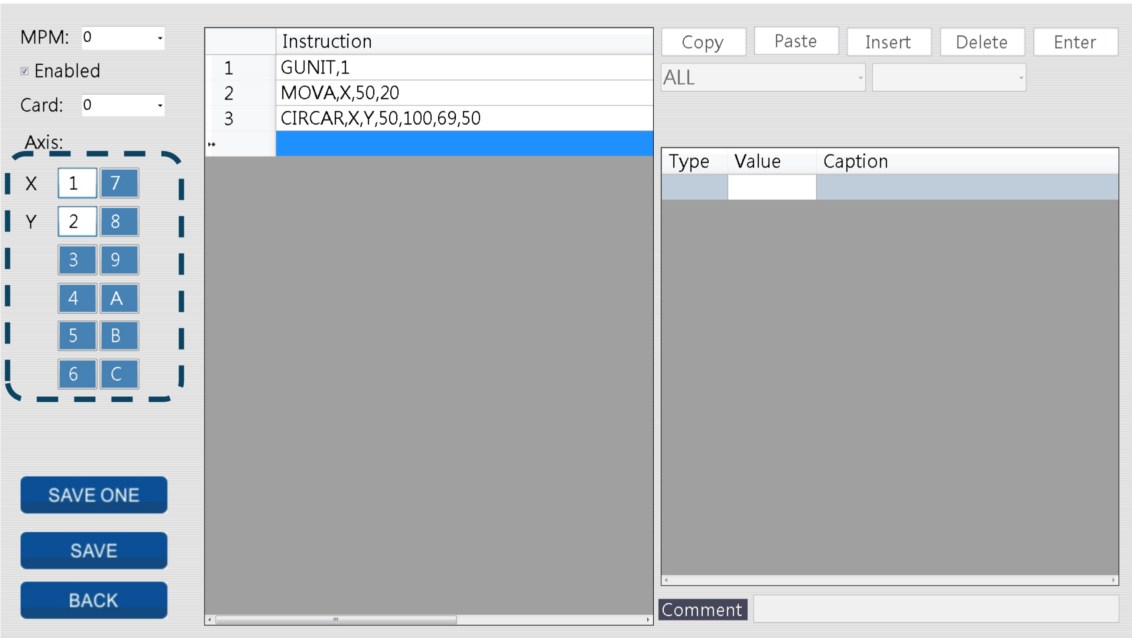


■ Descriptions for MPM editor operation

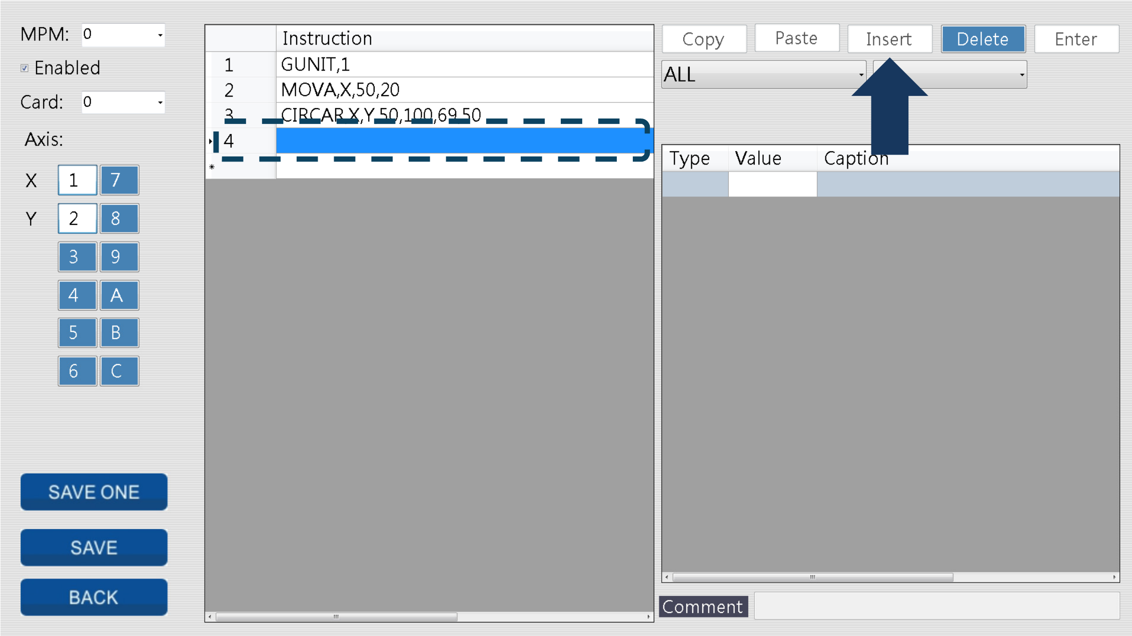
1. Select the MPM no. to be edited and check **Enabled** to enable the function of editing and saving MPM. The valid range of MPM no. is 0 to 99.



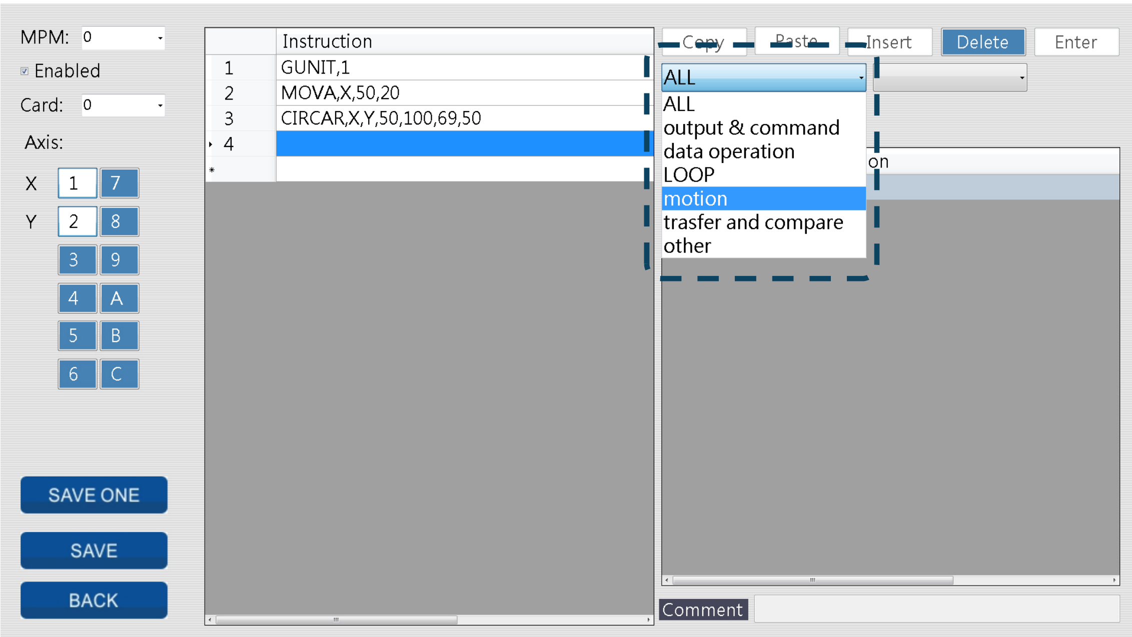
2. Select the card number and axis number. Range of the card number is between 0 and F. As for axis number, users can select 6 axes at most, which are X, Y, Z, U, V and W.



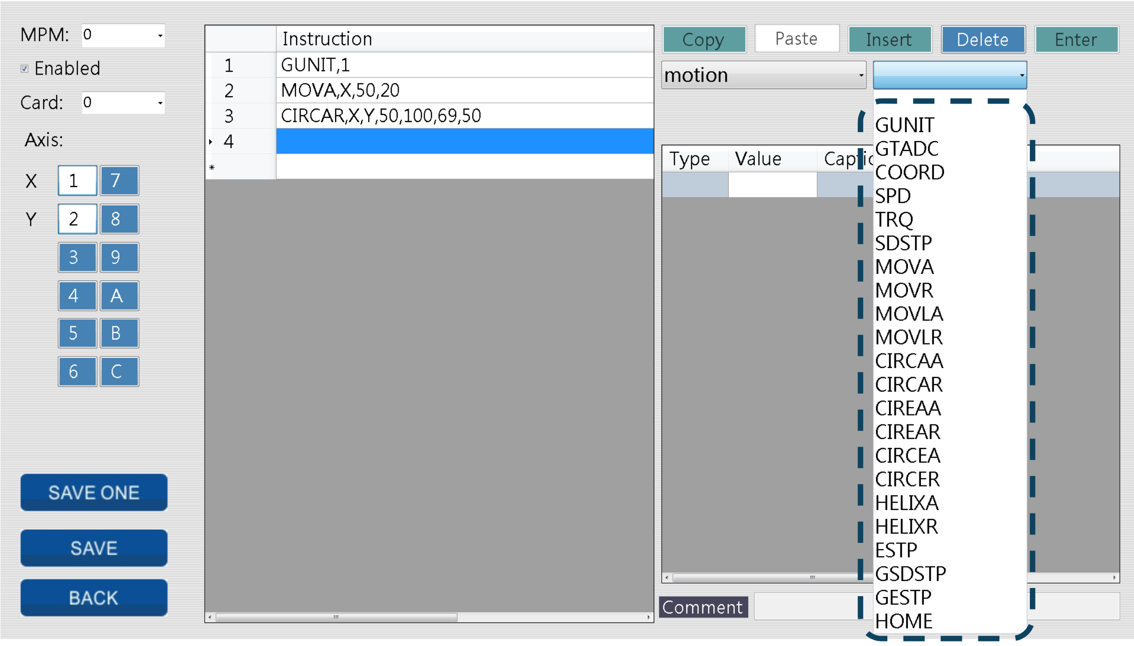
3. In the middle of the screen is the instruction table. Select the instruction to be edited. Users also can click **Insert** to add one blank line and edit the content on the right hand side. See the figures below.



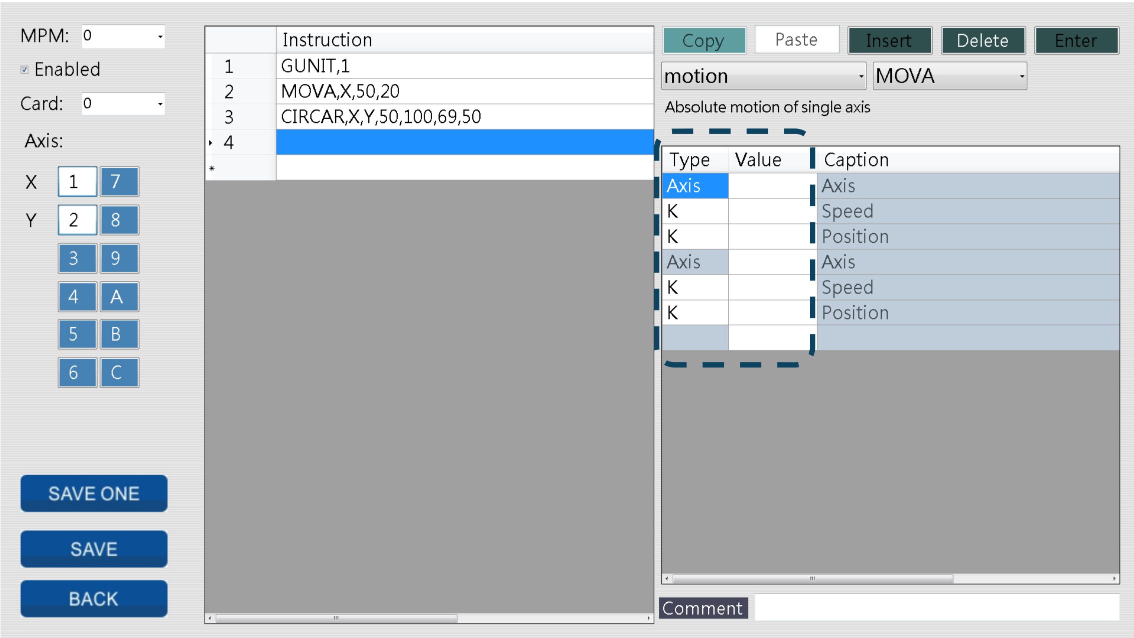
4. Select the instruction to be added from the drop-down list. It is suggested to select the instruction type first.



5. Then, select the instruction form the list.



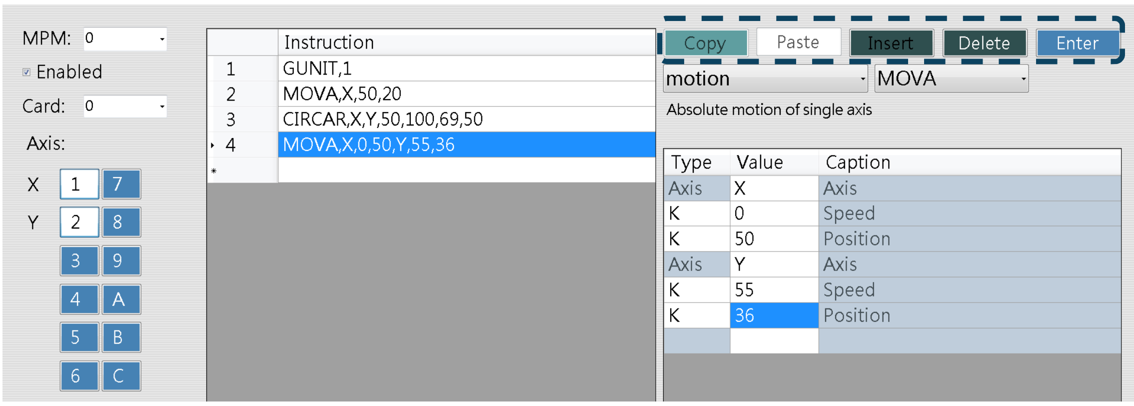
6. The screen will display parameters according to the instruction you selected.



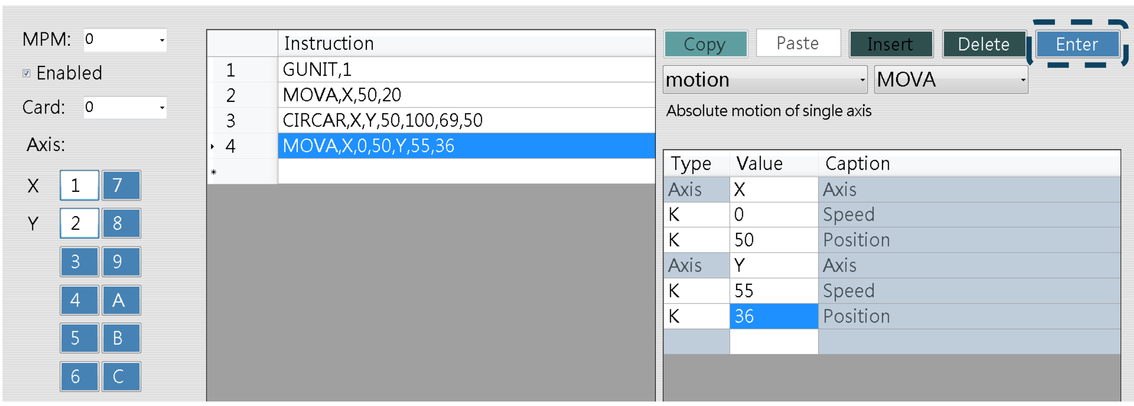
7

7. When you select the parameters type, enter the parameter value via on-screen keyboard. The pop-out on-screen keyboard is different in accordance with the parameters type you chose.

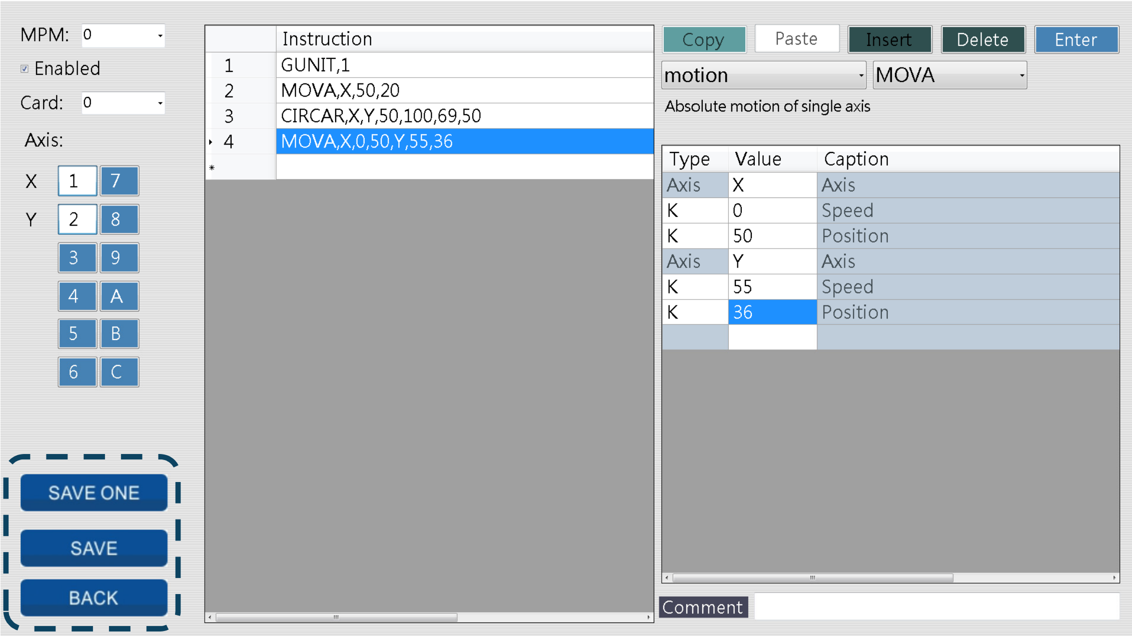
8. Functions like **Copy**, **Paste**, **Insert** and **Delete** can be applied when editing parameters.



9. Then, click **Enter** to complete one line of instruction editing.



10. To save the setting for one macro, click **SAVE ONE**. To save the macros that have been edited, click **SAVE**. Then, you can click **BACK** to exit MPM editor.



(This page is intentionally left blank.)

Communication 8

You can find the information about communication setting and the address related definition in this chapter before applying IMP communication.

8.1 Modbus Communication Setting 8-2

8.1.1 Ethernet Communication Setting 8-3

8.1.2 Serial Communication Setting 8-5

8.1.3 Setting of Communication Instruction 8-7

8.1.4 Communication Error Code 8-9

8.2 IPC Motion Platform Memory Address 8-10

8

8.1 Modbus Communication Setting

Go to [Quick start] > [Setting] > [Communication Setting]. In communication setting page, users can view or change the setting of IMP communication parameters, including the slave station number and TCP port. IMP can also be regarded as the master station. Through the setting of communication interface, device D and M can exchange the data with the external device.

The communication setting interface can be divided into two parts, (1) Tree view of communication setting and (2) Setting section. Click on **IPC Motion Platform** from the tree view, the configuration setting of IMP communication will show on the right hand side. See the detailed information below.

Name	Description	Default
IMP Slave Station	It sets the slave station number of communication.	1
TCP Port	It is the port used by Modbus/TCP slave station.	502
TCP Port Amount	It sets the port number used by Modbus/TCP master station.	0
COM Port Amount	It sets the COM port number used by the master station of Modbus serial communication.	0

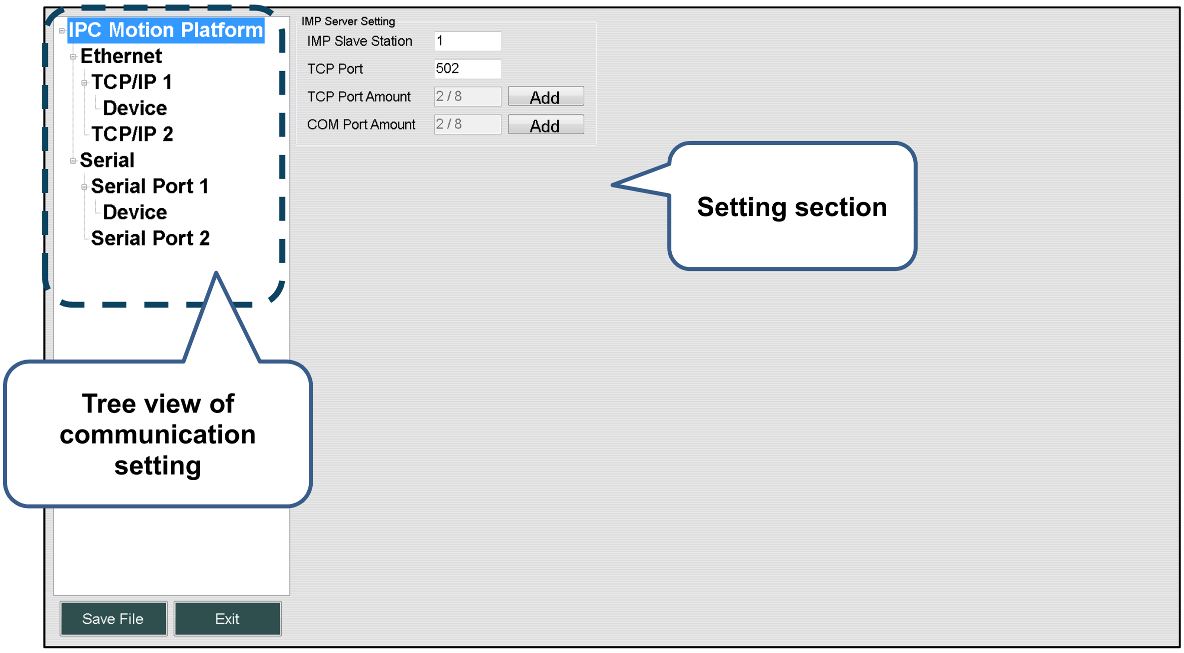


Figure 8.1.1 IMP Communication Interface

8

8.1.1 Ethernet Communication Setting

■ Modbus/TCP port setting

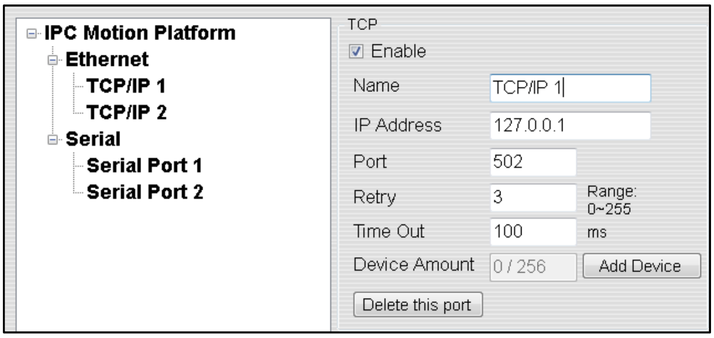


Figure 8.1.1.1 Modbus/TCP port setting

Ethernet channel setting is for creating Modbus/TCP connection. By clicking on the channel name (e.g. TCP/IP1) from the tree view of communication setting on the left, the Ethernet channel setting section appears on the right hand side. See the detailed description below.

Name	Description
Enable	Check this box to enable Ethernet channel.
Name	It sets the port name for Ethernet connection.
IP Address	It sets the IP address of the connected equipment.
Port	It sets the port that connects the Network.
Retry	It sets the retry times when data transmission failed. The range is between 0 and 225.
Time Out	It sets the time for communication timeout. Its unit is millisecond (ms).
Device Amount	It displays the device amount that using Ethernet port. (Add Device: To add new communication device.)
Delete this port	Delete the Ethernet port.

- Modbus/TCP connection device setting

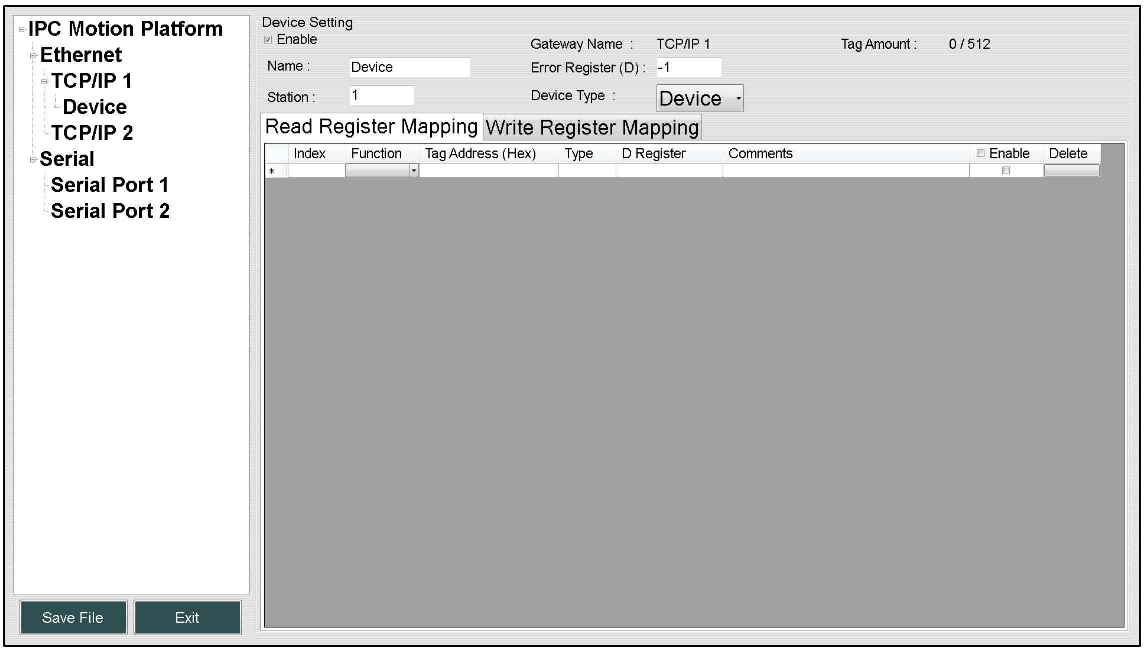


Figure 8.1.1.2 Modbus/TCP connection device setting

Ethernet connection device setting is used when IMP reads and writes the device data via communication. Through the setting of command mapping table, the communication command will be automatically generated during operation. And the communication data will be mapped to the internal memory of IMP PLC. Users can find the device name (e.g. Device) from the tree view of communication setting. And the setting section will appear on the right. See the detailed information below.

Name	Description
Enable	Check this box to enable the communication device that connected to the Ethernet.
Name	It sets the device name.
Station	It sets the device station number, range from 0 to 255.
Gateway name	It displays the gateway name that currently used for connection.
Error Register	It sets the address that saves the communication error code. When it is set to -1, no error code will be shown on the list[*1].
Device Type	It sets the device type.
Tag Amount	It displays the number of address that currently used.
Read Register Mapping	Mapping table for reading the data from the slave station [*2]
Write Register Mapping	Mapping table for writing the data to the slave station [*2]

Note:
1. Please refer to section 8.1.4 for further information of error code table.
2. Please refer to section 8.1.3 for further information of mapping setting table.

8.1.2 Serial Communication Setting

- Serial port setting

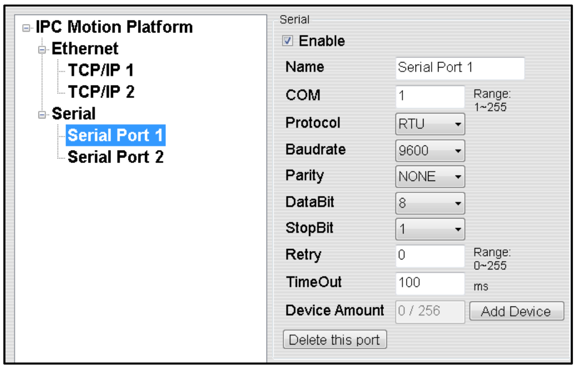


Figure 8.1.2.1 Modbus/Serial port setting

Users have to complete the setting of serial port for Modbus/Serial connection. By clicking on the serial port name from the tree view (e.g. Serial Port 1), its setting section will appear on the right hand side. See the detailed information below.

Name	Description
Enable	Check this box to enable the connection of serial port.
Name	It is the port name that can be set by users.
Com	It is the Com Port number for setting the connection.
Protocol	It sets the format of communication protocol, such as ASCII and RTU.
Baud rate	It sets the communication baud rate. Users can select 4800, 9600, 19200, 38400, 57600 and 115200.
Parity	It sets the parity check, None, Odd or Even.
Data Bit	It sets the data bit length. The standard length of each set of data bit is 7 or 8.
Stop Bit	It sets the length of the stop bit, 1 or 2.
Retry	It sets the retry times when data transmission failed. The range is between 0 and 225.
Time Out	It sets the time for communication timeout. Its unit is millisecond (ms).
Device Amount	It displays the device amount of serial communication port (Add Device: Add new communication device)
Delete this port	It deletes this port.

8

■ Modbus / Serial connection device setting

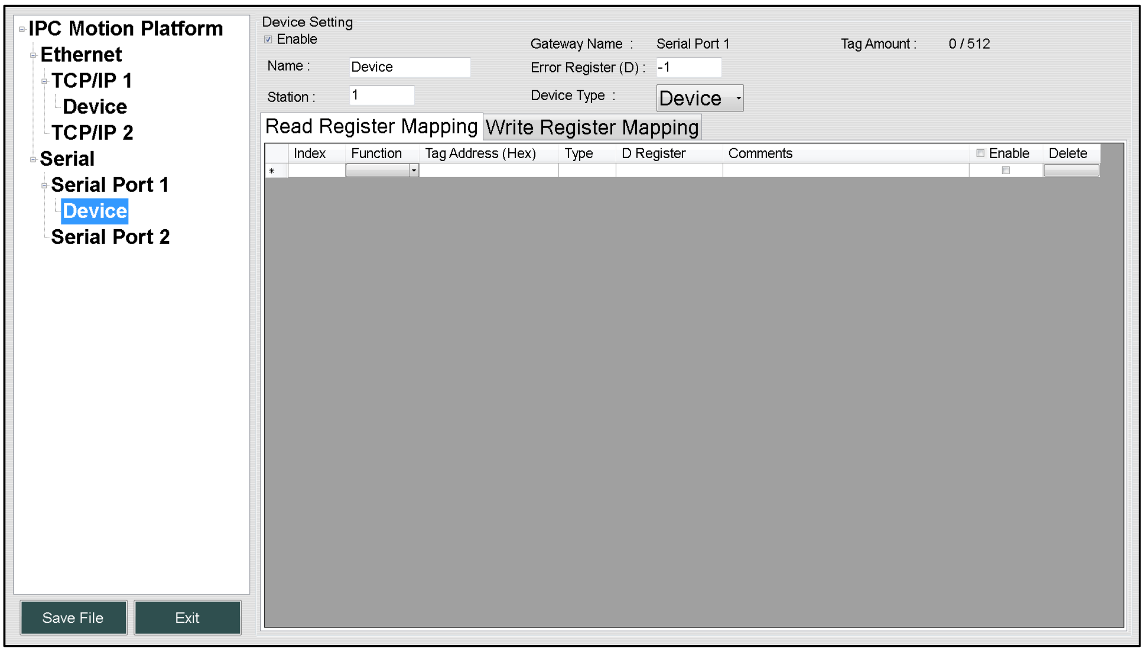


Figure 8.1.1.2 Modbus/TCP connection device setting

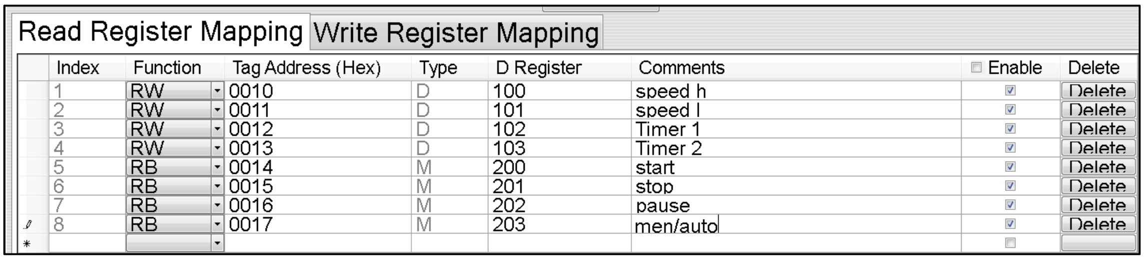
Serial connection device is applied to access and write the instruction list of IMP. Through the list, IMP automatically generates communication instructions during operation and the data will be saved in the internal memory device of PLC. Users can view the device name (e.g. Device) from the tree view and the setting section on the right hand side. See the detailed information below.

Name	Description
Enable	Check this box to enable the connection of the device.
Name	Users can define the device name.
Station	It sets the station number that connects to communication device.
Gateway name	It displays the gateway name that currently used.
Error Register	It sets the address that saves the communication error code. When it is set to -1, no error code will be shown on the list*1.
Device Type	It sets the device type.
Tag Amount	It displays the number of address that currently used.
Read Register Mapping	Mapping table for reading the data from the slave station *2
Write Register Mapping	Mapping table for writing the data to the slave station *2

Note:
1. Please refer to section 8.1.4 for further information of error code table.
2. Please refer to section 8.1.3 for further information of mapping setting table.

8.1.3 Setting of Communication Instruction

- Read register mapping


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

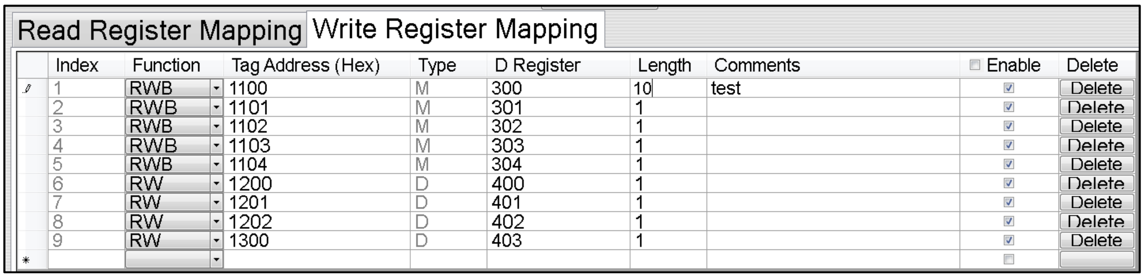
Figure 8.1.3.1 Read register mapping table of Modbus

Through the setting of Read Register Mapping Table, IMP keeps sending Modbus accessing command during operation. And the returned value will be saved in the corresponding memory device (D or M). If the addresses are consecutive or the interval of each address is less than 100, those addresses will be read in batch (batch read instruction) to optimize the communication.

Name	Description
Index	It is the serial number of communication instruction.
Function	Modbus function code: RW(0x03), R(0x04), RWB(0x01) and RB(0x02)
Tag Address	It sets the communication address of data accessing and is displayed in hexadecimal format, such as $FF1A_{16}$.
Type	It displays the device type of data saving. D: Data register M: Auxiliary relay
Register	It sets the device address that saves the data.
Comments	It is the section for comments.
Enable	Check this box to enable the communication instruction.
Delete	Delete this communication instruction.

- Write register mapping

  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 8.1.3.1 Write register mapping table of Modbus

Name	Description
Index	It is the serial number of communication instruction.
Function	Modbus function code: RW(0x06) and RWB(0x05)
Tag Address	It sets the communication address of data writing and is displayed in hexadecimal format, such as $FF1A_{16}$.
Type	It displays the device type of data saving. D: Data register M: Auxiliary relay
Register	It sets the device address of data source.
Length	It sets the communication length. The default value is 1. (unit: word)
Comments	It is the section for users to define.
Enable	Check this box to enable the communication instruction.
Delete	Delete this communication instruction.

8.1.4 Communication Error Code

Code	Description
01	Wrong function code. The communication function code is not supported.
02	Illegal address. The address is not allowed by the system.
03	Communication data error
04	Slave station error. Unknown error occurred.
06	Slave station is busy. The instruction is not completed.
101	Fail to enable COM / TCP connection
102	Com port setting exceeds the range
103	Com port is not enabled
104	Modbus function code error
105	Length of the reading data exceeds the range. The max. length should be 100 words or 200 bits.
106	Slave station number setting error. The range should be 1 ~ 255.
107	Address of accessing device exceeds the range. The range should be 0 ~ 65535.
108	Serial communication timeout
109	Checksum error (RTU CRC)
110	Checksum error (ASCII CRC)
111	Connection port initialization failed
112	Fail to connect to Modbus master station
113	TCP transmission failed
114	Modbus / TCP communication timeout
116	TCP port creation error
120	Length of the writing data exceeds the range. The max. length should be 100 words or 200 bits

8.2 IPC Motion Platform Memory Address

The communication protocols supported by IMP include Ethernet, RS-485, RS-422 and RS-232. Please refer to the following table for the range of memory address supported by Modbus Server, Modbus address and the function code supported by each device.

- Modbus communication protocol

Device	Range	Type	Modbus Address (Hex)	Modbus/TCP Function code
M	M0 ~ M19999	Bit	0000 ~ 4E1F	01, 05, 0F
DX	DX1.0 ~ DX12.63	Bit	D000 ~D2FF	02
DY	DY1.0 ~ DY12.63	Bit	E000 ~ E2FF	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

Soft Numerical Control, SNC (Optional)

9

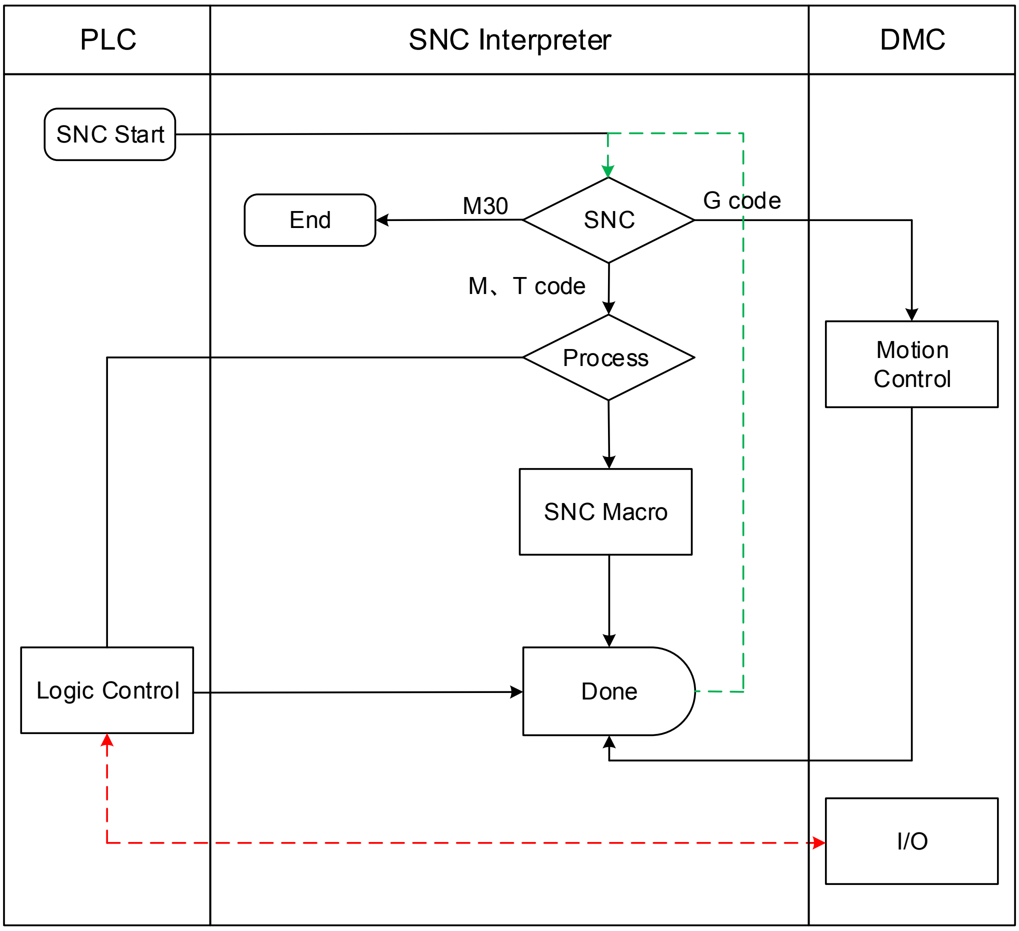
Soft Numerical Control (SNC) is the interpreter built on DMCNET motion card. It mainly assists DMCNET to execute route calculation via PC core computation, such as G Code interpretation, short path fitting and original path reversing. Apart from G Code, M Code and T Code are also provided to the programmer of Programmable Logic Control (PLC) for user-defined custom functions.

9.1 SNC Framework ········· 9-2
9.2 Parameter Descriptions ········· 9-3
9.2.1 SNC Special Parameters ········· 9-3
9.2.2 Motion Parameters ········· 9-5
9.2.3 Speed Control ········· 9-7
9.2.4 Hardware Information ········· 9-9
9.2.5 Tool Setting Parameters ········· 9-10
9.2.6 Backlash Compensation ········· 9-11
9.2.7 Reverse Backlash Compensation ········· 9-11
9.3 SNC Interpreter ········· 9-12
9.3.1 G Code Supporting Table ········· 9-12
9.3.2 M Code and T Code ········· 9-14
9.4 Descriptions of SNC Functions ········· 9-15
9.4.1 Accessing G Code ········· 9-15
9.4.2 External Macro ········· 9-16
9.4.3 Automatic Tool Setting ········· 9-17
9.4.4 Single Step Mode ········· 9-18
9.4.5 Spindle Control ········· 9-18
9.4.6 Manual Feed Rate Adjustment ········· 9-18
9.4.7 MPG Simulation Mode ········· 9-19
9.5 Description of SNC Error Codes ········· 9-19

9

9.1 SNC Framework

IMP is integrated with Soft Numerical Control (SNC). When SNC is started by PLC, the interpreter is automatically loaded with G Code file for route setting, and in accordance with the movement path from the file, it will give motion commands through DMCNET fieldbus. If the given command is recognized as M Code or T Code, the interpreter will distinguish where the command belongs by reference to SNC parameters. Assuming that PLC receives the control, PLC will read the memory integrated device status and execute relative actions (e.g. feeding or retrieving the cutting tool, on/off switch of cutting fluid interrupter). Once the actions are completed, the control will be returned to the interpreter; when the control is judged as not need changing, the interpreter will automatically execute command actions, such as program pause and program end.



9.2 Parameter Descriptions

This section particularly describes the parameter settings for the various functions of SNC interpreter.

9.2.1 SNC Special Parameters

The table below describes the settings of SNC system:

Name	Description
SNC_Card_Type(700)	Function: Type of the card in use. Set value: 0: DMC_B01.
SNC_Machine_Type(702)	Function: Subroutine type. Set value: 0.
SNC_Keep_Sharp_Variables (705)	Function: Whether to save all # variables of the subroutine. Set value: 0: All # variables will be cleared when SNC is executed; 1: Save.
SNC_User_Tool_Length_ Compenstation_Type(802)	Function: Select mode of acquiring tool length compensation. Set value: 0: Acquire tool length compensation directly from software parameters (2001 ~ 2100); 1: Acquire tool length compensation from macro.
SNC_User_Tool_Cutter_ Compensation_Type(803)	Function: Select mode of acquiring tool cutter radius compensation. Set value: 0: Acquire tool cutter radius compensation directly from software parameters (2201 ~ 2300); 1: Acquire tool cutter radius compensation from macro.
SNC_Check_Tool_No(805)	Function: Ensure the tool to use is on the tool list. Set value: 0: Disable (Default); 1: Enable.
SNC_Scan(807)	Function: G code pre-scanning function. Set value: 0: Disable; 1: Enable (Default).
SNC_Alwasy_Check_Axis_ Alarm(810)	Function: Check errors occurred in software limit, hardware limit and servo drive even when SNC is not processing, and generate SNC error messages. Set value: 0: Disable (Default); 1: Enable.
SNC_Ignore_NC_FeedRate (859)	Function: Ignore the feed rate setting in NC code when SNC is processing. Set value: 0: Disable (Default); 1: Enable.

Name	Description
SNC_Tool_Max(1890)	Function: Working with SNC_Check_Tool_No parameter, set the tool's maximum usage restriction.
SNC_Different_Work_Plane (708)	Function: Allow the main program and subroutine to use different working planes. Set value: 0: Disable (Default); 1: Enable.
SNC_Allow_kerf_With_M_Code(709)	Function: Allow inserting M Code in the process of tool cutter radius compensation process. Set value: 0: Disable (Default); 1: Enable.
SNC_Symbol_Enable(711)	Function: Enable symbol function, which allows to process from the code line after the symbol, working with below parameters: SNC_Symbol_User_Define_Character and SNC_Symbol_User_Define_Number Set value: 0: Disable (Default); 1: Enable. E.g.: The execution of G Code starts processing from the code line after "$2", of which the parameter setting is shown as below: SNC_Symbol_Enable = 1 SNC_Symbol_User_Define_Character = $ SNC_Symbol_User_Define_Number = 2
SNC_Code_Work_Plan_Macro1(717)	Function: The default working plane of the subroutine will only be available when the parameter of SNC_Different_Work_Plane (708) was enabled. Set value: 54 ~ 59, 59.1 ~ 59.9 (suggested values).
SNC_Work_Plane(1500)	Function: Set the coordinates of the working plane (default: 54) Set value: 54 ~ 59, 59.1 ~ 59.9 (suggested values).
SNC_Tool_Multi_Set_On_One_Line(1891)	Function: Allow setting multiple T Codes in the same line. Set value: 0: Disable (Default); 1: Enable.
SNC_User_Scan_Mcode (704)	Function: Whether to call macro (M, T, S, H) during pre-scanning. Set value: 0: Disable (Default); 1: Enable.
SNC_Dump_Log(821)	Function: Record motion commands. Set value: 0: Disable (Default); 1: Enable.
SNC_Dump_Log_Macro (830)	Function: Record M Code actions. Set value: 0: Disable (Default); 1: Enable.

Name	Description
SNC_Record_Enable(823)	Function: Record dda. Set value: 0: Disable (Default); 1: Enable.
NC_Use_Soft_Limit(1150)	Function: Function of software limit. Set value: 0: Disable (Default); 1: Enable, G Code will check whether the motion exceeds the software limit during pre-scanning; 2: Enable, check whether the motion exceeds the software limit only when processing.

9.2.2 Motion Parameters

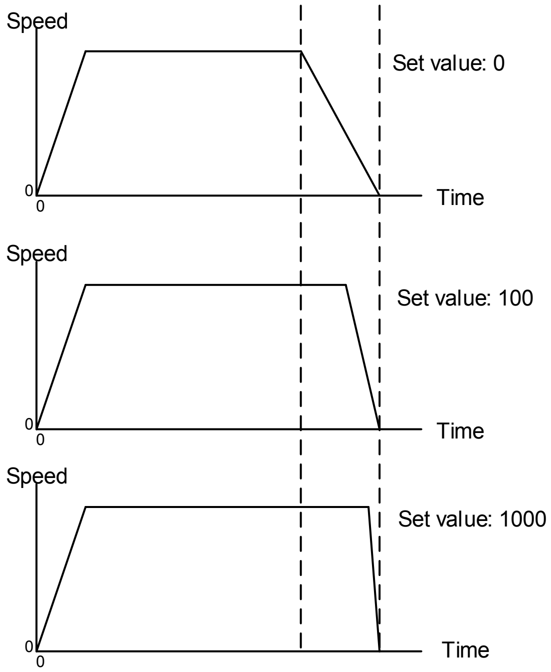
To set SNC motion path rules, please refer to the following table.

Name	Description
SNC_Ignore_NC_FeedRate (859)	Function: Ignore the feed rate setting in NC Code in general processing. Set value: 0: Not to ignore (Default); 1: Ignore.
SNC_Tolerance(860)	Function: Set the tolerance of continuous cutting. (Unit: mm)
SNC_Circle_Tolerance(867)	Function: Set the tolerance of arc cutting. (Unit: mm)
SNC_Kerf_Permit_Angle (886)	Function: Determine the correction angle of tool radius. When encountering lead angle, user can determine whether to use arc or linear line.
SNC_Corner_Control(887)	Function: Enable the function of corner rounding. Set value: 0: Disable (Default); 1: Enable.
SNC_Corner_Angle(888)	Function: Set the angle of corner rounding.
SNC_Corner_Speed(889)	Function: Set the speed of corner rounding.
SNC_Use_Look_Ahead (890)	Function: Enable the Look Ahead function to optimize the motion path. Set value: 0: Disable (Default); 1: Enable.
SNC_Fix_Slope(891)	Function: Set the elements to control acceleration and deceleration. Set value: 0: time; 1: acceleration.
SNC_Curve(892)	Function: Not available. Default: T-curve.
SNC_Scale_Rate(893)	Function: Adjust the scaling. Set value: 0: 0.001 (Default); 1: 0.00001

Name	Description
SNC_Short_Line_Warning_Or_Error(894)	Function: Set the processing mechanism when the point to point distance is too short during pre-scanning. Set value: 0: warning (Default); 1: error.
SNC_Time_Wait_For_Motion_Done(899)	Function When the motion command is sent, the waiting time of the motor before moving to the target position. (Unit: second) Set value: 0.001 (Default)
SNC_Tdec_Sd_Stop(900)	Function: Set Sd_Stop parameter of deceleration. Set value: Unit: time, if SNC_Fix_Slope is set to 0; Unit: acceleration, if SNC_Fix_Slope is set to 1.
SNC_G00_Use_Non_Line(870)	Function: Whether to use interpolation while setting G00. Set value: 0: Disable (Default); 1: Enable.
SNC_Feed_Rate_G00_AxisX(871)~SNC_Feed_Rate_G00_AxisW(879)	Function: Set G00 default feed rate when not using interpolation.

9.2.3 Speed Control

To set the mechanism's maximum acceleration/deceleration and the default feed rate, please refer to the descriptions below:

Name	Description
SNC_PermitMaxDec (989)	Function: Set the maximum deceleration (unit: mm/s²). It affects the Look Ahead multiple elements function from deceleration to stop. Smaller value indicates lower variation tolerance of the mechanism, and the speed changes will thus become steady, whereas the speed changes will be rapid. Set value: 
SNC_Str_Vel(990)	Function: Set the initial speed. (Unit: mm/min, same as feed rate)
SNC_AxisX_PermitMaxACC (991)~SNC_AxisW_PermitMaxACC(999)	Function: Set maximum acceleration for each axis. (Unit: mm/s²)
SNC_Feed_Rate_Percent (1170)	Function: Set the feed rate percentage. The program stops when the set value is 0.
SNC_Feed_Rate_G00(1171)	Function: Set the default feed rate of G00.
SNC_Tacc_G00(1172)	Function: Set the acceleration of G00. Set value: Unit: second, if SNC_Fix_Slope is set to 0; Unit: mm/s², if SNC_Fix_Slope is set to 1.
SNC_Tdec_G00(1173)	Function: Set the deceleration of G00. Set value: Unit: second, if SNC_Fix_Slope is set to 0; Unit: mm/s², if SNC_Fix_Slope is set to 1.

9

Name	Description
SNC_Feed_Rate_G01 (1174)	Function: Set the feed rate upper limit of G01.
SNC_Tacc_G01(1175)	Function: Set the acceleration of G01. Set value: Unit: second, if SNC_Fix_Slope is set to 0; Unit: mm/s², if SNC_Fix_Slope is set to 1.
SNC_Tdec_G01(1176)	Function: Set the deceleration of G01. Set value: Unit: second, if SNC_Fix_Slope is set to 0; Unit: mm/s², if SNC_Fix_Slope is set to 1.
SNC_Feed_Rate_Circle (1177)	Function: Set the feed rate upper limit of G02, G03.
SNC _Tacc_Circle(1178)	Function: Set the acceleration of G02, G03. Set value: Unit: second, if SNC_Fix_Slope is set to 0; Unit: mm/s², if SNC_Fix_Slope is set to 1.
SNC_Tdec_Circle(1179)	Function: Set the deceleration of G02, G03. Set value: Unit: second, if SNC_Fix_Slope is set to 0; Unit: mm/s², if SNC_Fix_Slope is set to 1.
SNC_Feed_Rate_G01_ Default(1183)	Function: Set the default feed rate of G0. If not set, SNC will apply the set value of SNC_Feed_Rate_G01.
SNC_Feed_Rate_Circle _Default(1184)	Function: Set the default feed rate of G02, G03. If not set, SNC will apply the set value of SNC_Feed_Rate_Circle.
SNC_Feed_Rate_Rate (1199)	Function: Set the feed rate resolution. Set value: 0.01: The unit of SNC_Feed_Rate_Percent is percentage. 0.001: The unit of SNC_Feed_Rate_Percent is permillage.
SNC_Feed_Rate_Override_ Change_Timer(1197)	Function: Set acceleration/deceleration time while adjusting the feed rate. Set value: Range: 0.1 ~ 10 seconds.
SNC_AxisX_Tacc_ G00(1841)~SNC_AxisW_ Tacc_G00(1849)	Function: Set acceleration of each axis at G00 command. Set value: Unit: second, if SNC_Fix_Slope is set to 0; Unit: mm/s², if SNC_Fix_Slope is set to 1.
NC_AxisX_Tdec_ G00(1851)~SNC_AxisW_ Tdec_G00(1859)	Function: Set deceleration of each axis at G00 command. Set value: Unit: second, if SNC_Fix_Slope is set to 0; Unit: mm/s², if SNC_Fix_Slope is set to 1.

Name	Description	
SNC_AxisX_LimitSpeed(971)~ SNC_AxisW_LimitSpeed(980)	Function:	Set the speed limit of each axis. (Unit: mm/min, same as feed rate)
SNC_Use_LimitSpeed(980)	Function:	Speed limit.
	Set value:	0: Disable (Default); 1: Enable, apply the set value of SNC_Axis_LimitSpeed; 2: Enable, only G00 applies the set value of SNC_Axis_G00_LimitSpeed, and other G codes (G01, G02 and G03) apply the set value of SNC_Axis_LimitSpeed.
SNC_NO_Triangle_G00 (1840)	Function:	If the speed curve of G00 is triangle-shaped (rapid acceleration/deceleration may cause trembling), the maximum speed curve will change to trapezoid-shaped by reference to the settings of SNC_NO_Triangle_Scale(1850).
	Set value:	0: Disable (Default); 1: Enable.
SNC_NO_Triangle_Scale (1850)	Function:	While SNC_NO_Triangle_G00 (1840) is enabled, set the percentage of the maximum speed to change the shape of the speed curve from triangle to trapezoid.
	Set value:	Range: 0.0 ~ 1.0; Default: 0.9.

9.2.4 Hardware Information

Following table describes the corresponding settings between the axis number used by SNC and the physical servo motor.

Name	Description	
SNC_Axes(1000)	Function:	Set the axis amount used by SNC.
SNC_AxisX_Axis(1001)~ SNC_AxisW_Axis(1009)	Function:	Set the No. of each axis.
SNC_Card_NO(1011)	Function:	Set the No. of the motion card used by SNC.
SNC_AxisX_Node(1021)~ SNC_AxisW_Node(1029)	Function:	Set the station No. for the servo drives used by each SNC axis.
SNC_AxisX_Dir(1141)~SNC_AxisW_Dir(1149)	Function:	Set directions for each SNC axis.
	Set value:	1: Move to the same direction as commanded by G Code; -1: Move to the opposite direction as commanded by G Code.

9.2.5 Tool Setting Parameters

Following table describes the settings for the SNC automatic tool setting function and the spindle tool No. recording function:

(Only ASD -****-A2-F series servo drive supports automatic tool setting function)

Name	Description
Spindle_Current_T	Function: Set the tool No. of the current spindle.
ToolGauge_Interal	Function: Set the distance between the work plane and the surface of the tool setter. (Unit: mm)
ToolGauge_SensorType	Function: Set the type of the electrical contact used by the tool setter. Set value: 0: Contact b (NC); 1: Contact a (NO)
ToolGauge_X	Function: Set the absolute coordinate (X-axis) of the starting point of the automatic tool setting program.
ToolGauge_Y	Function: Set the absolute coordinate (Y-axis) of the starting point of the automatic tool setting program.
ToolGauge_Z	Function: Set the absolute coordinate (Z-axis) of the starting point of the automatic tool setting program.
ToolGauge_1Down_Speed	Function: Set the first speed of Axis Z moving down in the automatic tool setting program. (Unit: mm/min)
ToolGauge_2Down_Speed	Function: Set the second speed of Axis Z moving down in the automatic tool setting program. (Unit: mm/min)
ToolGauge_Up_Speed	Function: Set the speed of Axis Z moving up in the automatic tool setting program. (Unit: mm/min)
SpindleToolNo1	Function: Set the initial tool No. of the spindle.
SpindleToolCnt	Function: Set the tool amount of the spindle.

9.2.6 Backlash Compensation

Following settings for the backlash compensation of each axis helps to achieve better precision in processing.

Name	Description
SNC_PEF_AxisX_Enable(200001)~SNC_PEF_AxisW_Enable(200009)	Function: Enable the backlash compensation function of each axis.
SNC_PEF_AxisX_Interval(200011)~SNC_PEF_AxisW_Interval(200019)	Function: Set the point to point distance of the backlash compensation of each axis. (Unit: mm)
SNC_PEF_Position_Table_1~SNC_PEF_Position_Table_100	Function: Set the backlash compensation value of each axis. The value of each axis can be set to 100 points respectively.

9.2.7 Reverse Backlash Compensation

Following settings for reverse backlash compensation of each axis helps to achieve better precision in processing.

Name	Description
SNC_AxisX_Backlash_Enable(601)~SNC_AxisW_Backlash_Enable(609)	Function: Enable the reverse backlash compensation function of each axis.
SNC_AxisX_Backlash_Value(611)~SNC_AxisW_Backlash_Value(619)	Function: Set the backlash reverse compensation value of each axis.
SNC_AxisX_Backlash_Dir(621)~SNC_AxisW_Backlash_Dir(629)	Function: Set the direction of reverse backlash compensation. Set value: 1: No compensation when moving forward; -1: No compensation when moving backward.
SNC_AxisX_Backlash_Speed(631)~SNC_AxisW_Backlash_Speed(639)	Function: Set the speed of reverse backlash compensation.
SNC_AxisX_Backlash_Acc(641)~SNC_AxisW_Backlash_Acc(649)	Function: Set the acceleration speed of reverse backlash compensation.

9

Name	Description
SNC_AxisX_Backlash_ Ignore_Range_Pulse(661)~ SNC_AxisW_Backlash_ Ignore_Range_Pulse(669)	Function: Ignore the pulse if it is too small during backlash compensation, so as to prevent incorrect compensation caused by small errors of the motion card and NC coordinate transformation.

9.3 SNC Interpreter

9.3.1 G Code Supporting Table

G Code	Description	Support	G Code	Description	Support
G00	Rapid linear positioning	□	G61	Exact stop positioning mode	□
G01	Cutting feed	□	G64	General cutting mode	□
G02	Clockwise arc cutting	□	G65	Simple macro calling	□
G03	Counterclockwise arc cutting	□	G66	Macro command calling	□
G04	Pause command	□	G67	Cancel macro mode	□
G09	Exact stop positioning	□	G68	Coordinate system rotation	□
G17	X-Y plane selection	□	G69	Cancel coordinate system rotation	□
G18	Z-X plane selection	□	G70	Set the unit to “inch”	□
G19	Y-Z plane selection	□	G71	Set the unit to “mm”	□
G28	Go to reference point	□	G73	High speed perk drilling cycle	□
G29	Go from reference point to the original position	□	G74	Left hand tapping cycle	△
G30	Go to any reference point	□	G76	Fine boring cycle	△
G40	Cancel tool radius compensation	□	G80	Cycle cancellation	□
G41	Tool radius left compensation[*1]	□	G81	Drilling cycle	□
G42	Tool radius right compensation[*1]	□	G82	Drilling cycle with dwell at hole bottom	□
G43	Tool length compensation (+)[*2]	□	G83	Perk drilling cycle	□
G44	Tool length compensation (-)[*2]	□	G84	Tapping cycle	△
G49	Cancel Tool length compensation	□	G85	Boring cycle	□
G50	Cancel scale function	□	G86	High speed drilling cycle	△

G Code	Description	Support	G Code	Description	Support
G51	Enable scale function	□	G87	Fine back boring cycle	△
G50.1	Cancel mirror image function	□	G88	Semi-automatic fine boring cycle	△
G51.1	Enable mirror image function	□	G89	Boring cycle with dwell at hole bottom	□
G52	Set local coordinate system (offset)	□	G90	Absolute moving	□
G53	Machine coordinate setting	□	G91	Incremental moving	□
G54~ G59	Set working coordinate system	□	-	-	-

□：Standard supports

△：Enable PLC editing

Note:

1. If the value of SNC_User_Tool_Cutter_Compensation_Type is set to 1, the value of the tool radius can only be confirmed after tool gauging. As predicting the compensation value is not available, G41 and G42 will not be executed during pre-scanning.
2. If the parameter value of SNC_User_Tool_Length_Compensation_Type is set to 1, the value of the tool length can only be confirmed after tool gauging. As predicting the compensation value is not available, G43 and G44 will not be executed during pre-scanning.

9

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
M30	Function: End of system. Description: Stop program and end system.
M98	Function: Subroutine 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: Calling file code (if input P0000, O0000.NC will be called); L: The times of running subroutine.
M99	Function I: Cycling. Description: When encounter M99, G Code will return to the first line of the cyclic task for execution. Function II: Subroutine Ending. Description: When using M99, the subroutine will end and return to the cyclic task.

Custom M Code and T Code

When run 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 number is set in tool setting parameters (see section 9.2.5). If T code assigns other tool number except the number for spindle, the system will automatically compensate the offset distance between the assigned tool position and the spindle's position.

M Code and T Code command flag

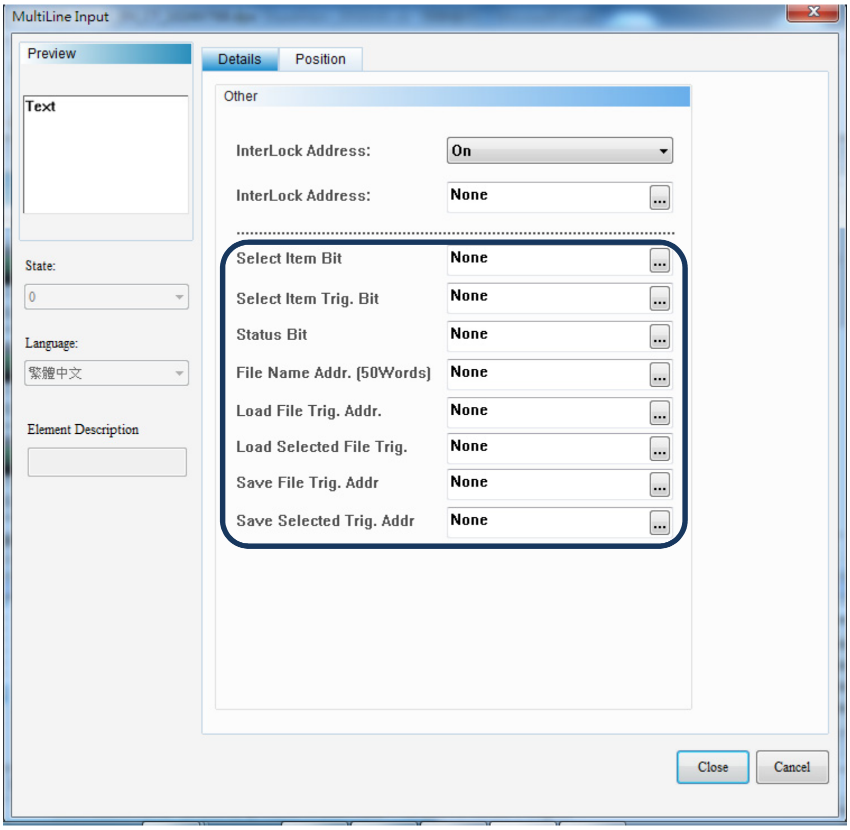
	SNC 1 Memory Location	SNC 2 Memory Location	SNC 3 Memory Location	SNC 4 Memory Location
T1 ~ T100	R32001 ~ R32100	R34001 ~ R34100	R36001 ~ R36100	R38001 ~ R38100
M0 ~ M999	R31000 ~ R31999	R33000 ~ R33999	R35000 ~ R35999	R37000 ~ R37999

9.4 Descriptions of SNC Functions

9.4.1 Accessing G Code

Introduction to the functions of G Code editing interface

IMP uses DopSoft 3.00.01 Multiline Editor as the editing interface of 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: It shows the returned value of the file opening status. More details are given in the table below describing the operation status of multiline input.

Return value	Description of the functions	Return value	Description of the functions
1	Element in process	5	Opening file failure
2	Execution canceled	6	Saving file failure
3	Execution completed	7	File successfully opened
4	Execution failure	8	File successfully saved

File Name Addr. (50 Words):	To record the storage address of current files. W31100 is the storage address of SNC 1 path.
Load File Trig. Addr.	Set the trigger address to load 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.
Create File Trig. Addr.	Set the trigger address to create new file.

Parameters for accessing and saving G Code

Using SNC group as an example: (Refer to Appendix A for register address)

- W31100: SNC accesses the file path of G Code, sequentially occupies 50 addresses, which is 100 bytes.

9.4.2 External Macro

Function description

Apart from using G Code to describe motion path, SNC also uses M Code for mechanical motion controlling and T Code for tool changing. In IMP system, after being interpreted, G Code is output directly through fieldbus, and executed by motion unit. If it is M Code or T Code, the control will be transferred to PLC, which will determine the execution actions of the M Code or T Code via ladder diagrams, and these actions include enabling or disabling I/O and the function of using external macro. External macro can be used to describe certain mechanical functions, including tool changing and origin moving.

Parameters

Using SNC group as an example: (Refer to Appendix A for register address)

- W31015

Function: SNC calling the macro file control code.
Set value: 0: No action; 1: Call Macro O; 2: Call Macro T.

Return value: 99: File not found.

- W31016

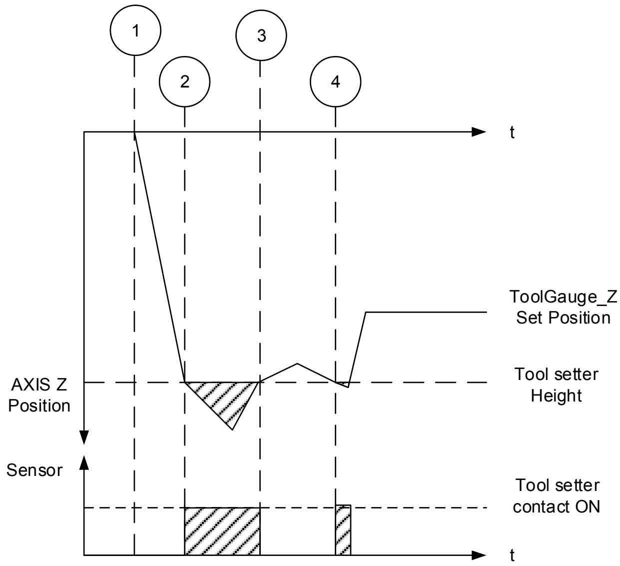
Function: SNC calling the macro file No.
Range: 0000~9999

9.4.3 Automatic Tool Setting

Description of the function

To achieve higher precision, IMP's tool setting function allows executing automatic tool setting through the PR mode built in the servo drive for obtaining the position where the tool reaches the tool setter.

1: When the tool setting motions start, Axis Z moves down at the first speed set by the user.
2: When the tool first touches the tool setter, Axis Z decelerates and then 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: The mode mentioned above applies to ASD-****-A2-F models.

Parameters

Using SNC group as an example: (Refer to Appendix A for register address)

- W31000

Function: Activate SNC control code.

Set value:

14: Start tool setting program.

15: Stop tool setting program.

Note: Please refer to section 9.2.5 for more detailed information of tool setting parameters.

9.4.4 Single Step Mode

Description of the function

This function allows choosing G Code single step mode and sequential execution mode before starting SNC. Single step mode executes G Code in the unit of lines: A single line will be executed when the rising-edge trigger is activated every time.

Parameters

Using SNC group as an example: (Refer to Appendix A for register address)

- R32981: Enable SNC single step execution mode (only applicable before starting SNC).
- R32982: Trigger SNC single step execution.

9.4.5 Spindle Control

Description of the function

The SNC spindle operation can be adjusted through S-function parameters. When SNC interpreter executes M Code containing S-function parameters, SNC will save the S-function parameter values in special register for PLC to perform spindle speed adjustment.

Parameters

Using SNC group as an example: (Refer to Appendix A for register address)

- W31020: After obtaining S-function parameters, SNC will save the values in the register address.

9.4.6 Manual Feed Rate Adjustment

Description of the function

The user can manually control SNC feed rate through user interface or external switch, and when such 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

Using SNC group 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 MPG.
- W19002: Set the station No. of the ASD-DMC-RM64MN module that is connected to MPG.
- W19000: To use MPG simulation mode, please 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 manually control.

If R32997 is OFF, the feed rate will be automatically set to 100% when SNC is shut down.

9.4.7 MPG Simulation Mode

Description of the function

The user can use MPG rotation speed to simulate SNC feed rate, and the speed calculation formula is shown as below:

G Code original speed x MPG rotation percentage (0 ~ 100%) = SNC actual execution speed.

Parameters

Using SNC group 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 MPG.
- W19002: Set the station No. of the ASD-DMC-RM64MN module that is connected to MPG.
- W19000: To use the MPG simulation mode, please 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 of using MPG to simulate SNC feed rate, and if this function is disable, the feed rate W32480 will be automatically set to 0.

9.5 Description of SNC Error Codes

- Data and information error

No.	Name	Description
1	ERR_FILE_NOT_EXIST	File does not exist.
2	ERR_NO_DATA	1. No string is found. 2. No object to process (the total line number is zero).
3	ERR_DATA_NOT_COMPLETE	The second parameter of API SNC_set_process_data is zero.
4	ERR_START_OVER	The start line exceeds the total line number.
5	ERR_DMC_01_DLL_Not_Full_Version	DDL version is wrong.
6	ERR_CUTTING_LINE_TOO_SHORT	Error / warning occurs when the cutting line is too short.
7	ERR_GOTO_LINE_WRONG	Use goto command but cannot find the corresponding tag.
8	ERR_GOTO_LINE_REDEFINED	Use goto command but the tag is duplicated.

No.	Name	Description
9	ERR_GROUP_OUT_OF_RNG	The set value of SNC_Append_Group exceeds the range.
10	ERR_GROUP_SAME_GROUP	The set value of SNC_Append_Group is duplicated.
11	ERR_GROUP_NO_APPEND	SNC group No. is not specified.
13	ERR_CREATE_THREAD_FAIL	Fail to create thread.

- Duplicate definition of G Code (in the same line of G Code).

No.	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.

No.	Name	Description
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.

- The variable of G Code is a negative number.

No.	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

■ The undefined G Code character / function

No.	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	Unknown operator
314	ERR_GCODE_BUG_UNKNOWN_OPERATION	Unknown operation
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

No.	Name	Description
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
330	ERR_GCODE_UNKNOWN_WORD_WHERE_UNARY_OPERATION_COULD_BE	Unknown function
331	WARNING_GCODE_G10_UNKNOWN_TYPE	Not supporting G10

■ G Code setting exceeds the range.

No.	Name	Description
403	ERR_GCODE_M_CODE_TOO_BIG	M code exceeds the range; Valid 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

■ Other errors

No.	Name	Description
501	ERR_GCODE_NEGATIVE_OR_ZERO_Q_VALUE	The value of the feed cutting depth Q cannot be smaller than or equals 0.
502	ERR_GCODE_NEGATIVE_SPINDLE_SPEED	The value of the speed S cannot be smaller than 0.
503	ERR_GCODE_NEGATIVE_TOOL_ID	The value of the 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 has to 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 repeated count of the peck drilling cycle is negative.
701	ERR_SNC_INITIAL_FAILED	SNC initialization failure.
702	ERR_CANT_SET_WHEN_PROCESSING	Parameter cannot be accessed during processing.
703	ERR_AXIS_OUT_OF_RNG	Axis number exceeds the range; valid range: 1 ~ 9.
704	ERR_AXIS_REDEFINE	Axis number is repeatedly used.
706	ERR_AXES_ZERO	Total axis amount in use is zero.
707	ERR_AXES_OUT_OF_RNG	The axis amount 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	E-gear ratio is zero.
713	ERR_G00_SPD_ZERO	G00 feed speed is zero.
715	ERR_WRONG_PLANE	Incorrect setting of work plane.

No.	Name	Description
723	ERR_MACRO_OVER_RNG	Layer of subroutine exceeds four layers.
727	ERR_G02_G03_PARAM	Incorrect setting of G02 and G03 parameters.
728	ERR_G02_G03_PLANE	G02 and G03 use the wrong work plane (G17~G19).
729	ERR_G02_G03_CALC	G02 and G03 cannot calculate coordinate.
730	ERR_G02_G03_AXES_OVER	Axis amount used by G02 and G03 is over 3 axes.
731	ERR_PROCESSING_IS_RUNNING	Processing function is not applicable when the system is in process.
732	ERR_TOOL_MAX_OVER_RNG	SNC_Tool_Max parameter setting exceeds the range; valid range: 1~100.
733	ERR_CUTTER_COMPENSATION_ARC_PLANE_NOT_SUPPORT	Tool radius compensation only supports X-Y plane.
734	ERR_CUTTER_COMPENSATION_CANT_USE_HELI	Tool radius compensation and helical interpolation cannot be used in the same time.
735	ERR_CUTTER_COMPENSATION_CALC	The path of tool radius compensation cannot be calculated.
737	ERR_CUTTER_FIRST_MOTION_ARC	The first motion of tool radius compensation cannot be G02 or G03.
738	ERR_CUTTER_NOT_FINISH	Before the tool radius compensation completes, macro function cannot be executed.
741	ERR_MEMORY_ALLOC_FAIL	SNC memory allocating failure.
742	ERR_USER_CALLBACK_NULL	Use 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 in drilling mode.
763	ERR_ISO_CYCLE_NOT_SUPPORT	Drilling cycle mode is not supported.
802	ERR_SETTING_GEAR	Incorrect setting of gear ratio

9

No.	Name	Description
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	Use SNC_Check_Tool_No parameter, but the set value of SNC_Tool_Max is zero.
807	ERR_SETTING_DIRECT	Wrong direction (-1, 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	This station No. is unable to find the device.
904	ERR_DEVICE_UNKNOWN	Unknown device
911	ERR_API_ERRNO	Bottom layer API returns error; please access parameter SNC_API_ErrNo.
1001	ERR_GROUP_INIT_FIRST	Group number is not set.
1002	ERR_GRUOP_OVER_RANGE	Operation group exceeds the setting range.
1004	ERR_GRUOP_CARD_TYPE	Wrong card type
2000	ERR_TRIGGER_SOFT_LIMIT	Software limit is triggered.
9999	ERR_SECURITY_FAILED	Security authentication failed.

List of Special Register A

When editing PLC instructions of IMP, you can find definitions of each register in this chapter.

Troubleshooting ··········· A-2
List of special register (W, R) in IMP system ··········· A-3
Special register (W, R) of single-axis motion ··········· A-10
Special Register (W, R) of Servo Group ··········· A-13
Special Register (W, R) of Motion Program Macro (MPM) ··········· A-15
Special Register (W, R) of SNC ··········· A-15

A

Troubleshooting

Error occurrence	Causes	Corrective actions
IMP software cannot be activated.	IMP software is not certified.	You should get license authorization and register through RegisterAP. (Please refer to Section 2.3 for detailed information.)
Input/output signal error	Connection of the terminal socket is loosed or has poor contact	Check if the wiring or terminal is loosed.
A window showed “Init fail, please rstart software!” pops out.	The DMCNET card and driver were not properly installed.	Check if the DMC card and driver are installed correctly.
A window showed “Card NO:X NO slave found” pops out.	Communication protocol or node number setting error	In DMCNET field bus, node 1 has to be existed. Please check if the setting of communication protocol and node number is correct. (Refer to Section 3.1 for detailed information).
	Poor communication	The max. connection distance is 30 meters for DMCNET. Please select the shielded-twisted pair cable.
DOPsoft is unable to download PLC program. A window showed “ETHERNET can’t opened” pops out.	HMI software of IMP is not activated.	Open IMP software to make sure HMI software can be opened.
	Ethernet is not connected.	Check if the Ethernet is connected.
	Industrial PC firewall for installing IMP software is enabled.	Put IMP software in firewall exception list or disable the firewall.
	PC that editing the program should share the same subnetwork with IMP software.	Change both IP addresses to the same subnetwork.
A warning message of “DB!” pops out by HMI software.	“SSCERuntime_x86-ENU” is not installed.	Please install “SSCERuntime_x86-ENU”.

List of special register (W, R) in IMP system

■ Operation status

Function	Property	No.	Description
Operation flag (Contact a)	R	R0	Operation flag of IMP software
Operation flag (Contact b)	R	R1	Operation flag of IMP software (Contact b)
Reserved	-	R2	Reserved
Reserved	-	R3	Reserved
Initial pulse	R	R4	This bit is On in the first PLC cycle.
Reserved	-	R5	Reserved
Minimization flag of HMI	R	R6	HMI is operated in minimization state.
Clock pulse 0.5 / 0.5-second	R	R13	When PLC is operating, this bit is ON for 0.5 sec and OFF for 0.5 sec. Clock drift may occur in this occasion.
Clock pulse 1 / 1-second	R	R14	When PLC is operating, this bit is ON for 1 sec and OFF for 1 sec. Clock drift may occur in this occasion.

■ Perpetual Calendar

Function	Property	No.	Description
Date (year)	R	W80	Accessing the system time when using IMP software.
Date (month)	R	W81	
Date (day)	R	W82	
Time (hour)	R	W83	
Time (minute)	R	W84	
Time (second)	R	W85	
Time (total seconds)	R	W86	It starts to count from 00:00:00.
	R	W87	
Date (total days)	R	W88	It starts to count from 1/1/1980.
	R	W89	
Date (week)	R	W90	

A

■ Motion card information

Function	Property	Card 1	Card 2	Card 3	Description
Motion card number	R	W6000	W6500	W7000	Display the number of DMCNET motion card
Motion card version	R	W6001	W6501	W7001	Display the firmware version of DMCNET motion card
	R	W6002	W6502	W7002	
Times of DMCNET transmission error	R	W6003	W6503	W7003	Display the accumulative number of DMCNET communication error
Times of DMCNET receiving error	R	W6004	W6504	W7004	
	R	W6005	W6505	W7005	

■ The connected device type

Function	Property	Card 1	Card 2	Card 3	Description
Device type of node 1	R	W6010	W6510	W7010	Read the connected device type. See below for the code and corresponding model: 0X04020192: ASD-A2-F 0X08020192: ASD-A2-R 0X06020192: ASD-M 0X09020192: ASD-A2-S 0X05020192: ASD-B2-F 0X4120191: ASD-DMC-RM32NT 0X8120191: ASD-DMC-RM64NT 0X4110191: ASD-DMC-RM32MN 0X8110191: ASD-DMC-RM64MN 0X4130191: ASD-DMC-RM32PT 0X14100191: ASD-DMC-RM04PiM2 0X8140191: ASD-DMC-RM04AD 0X8180191: ASD-DMC-RM04DA 0X11210191: ASD-DMC-GE01PH 0X8330191: HMC-RIO3232RT5
	R	W6011	W6511	W7011	
Device type of node 2	R	W6012	W6512	W7012	
	R	W6013	W6513	W7013	
Device type of node 3	R	W6014	W6514	W7014	
	R	W6015	W6515	W7015	
Device type of node 4	R	W6016	W6516	W7016	
	R	W6017	W6517	W7017	
Device type of node 5	R	W6018	W6518	W7018	
	R	W6019	W6519	W7019	
Device type of node 6	R	W6020	W6520	W7020	
	R	W6021	W6521	W7021	
Device type of node 7	R	W6022	W6522	W7022	
	R	W6023	W6523	W7023	
Device type of node 8	R	W6024	W6524	W7024	
	R	W6025	W6525	W7025	
Device type of node 9	R	W6026	W6526	W7026	
	R	W6027	W6527	W7027	
Device type of node 10	R	W6028	W6528	W7028	
	R	W6029	W6529	W7029	
Device type of node 11	R	W6030	W6530	W7030	
	R	W6031	W6531	W7031	
Device type of node 12	R	W6032	W6532	W7032	

Function	Property	Card 1	Card 2	Card 3	Description
	R	W6033	W6533	W7033	

A

- The connected device version

Function	Property	Card 1	Card 2	Card 3	Description
Firmware version of node 1	R	W6082	W6582	W7082	Display the firmware version of the connected device. Please refer to the user manual of each device for further information.
	R	W6083	W6583	W7083	
Firmware version of node 2	R	W6084	W6584	W7084	
	R	W6085	W6585	W7085	
Firmware version of node 3	R	W6086	W6586	W7086	
	R	W6087	W6587	W7087	
Firmware version of node 4	R	W6088	W6588	W7088	
	R	W6089	W6589	W7089	
Firmware version of node 5	R	W6090	W6590	W7090	
	R	W6091	W6591	W7091	
Firmware version of node 6	R	W6092	W6592	W7092	
	R	W6093	W6593	W7093	
Firmware version of node 7	R	W6094	W6594	W7094	
	R	W6095	W6595	W7095	
Firmware version of node 8	R	W6096	W6596	W7096	
	R	W6097	W6597	W7097	
Firmware version of node 9	R	W6098	W6598	W7098	
	R	W6099	W6599	W7099	
Firmware version of node 10	R	W6100	W6600	W7100	
	R	W6101	W6601	W7101	
Firmware version of node 11	R	W6102	W6602	W7102	
	R	W6103	W6603	W7103	
Firmware version of node 12	R	W6104	W6604	W7104	
	R	W6105	W6605	W7105	

A

■ Communication error

Function	Property	Card 1	Card 2	Card 3	Description
Communication error times of node 1	R	W6154	W6654	W7154	Display the accumulative number of communication error between master station and slave station.
Communication error times of node 2	R	W6155	W6655	W7155	
Communication error times of node 3	R	W6156	W6656	W7156	
Communication error times of node 4	R	W6157	W6657	W7157	
Communication error times of node 5	R	W6158	W6658	W7158	
Communication error times of node 6	R	W6159	W6659	W7159	
Communication error times of node 7	R	W6160	W6660	W7160	
Communication error times of node 8	R	W6161	W6661	W7161	
Communication error times of node 9	R	W6162	W6662	W7162	
Communication error times of node 10	R	W6163	W6663	W7163	
Communication error times of node 11	R	W6164	W6664	W7164	
Communication error times of node 12	R	W6165	W6665	W7165	

■ Module number

Function	Property	No.	Description
Total card number	R	W8000	Based on the device amount that IMP connects, PLC can check if all devices are well connected.
Total number of servo axis	R	W8001	
Reserved	R	W8002	
Node number of digital input module	R	W8003	
Node number of digital output module	R	W8004	
Channel number of analog input	R	W8005	
Channel number of analog output	R	W8006	

■ User interface

Function	Property	No.	Description
User interface activation code (User_interface)	R/W	W9000	Before calling the user interface, users should complete the parameter setting as below. (Refer to W9020 for the call path.) 1000: Parameter setting of DMC slave station 1010: MPM editor 1020: Communication setting 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	W9020	Through the action element in DOPSoft 3.00.XX version, users can set the execution path for activating the user interface. (Work with W9000)

■ System control

Function	Property	No.	Description
System operation area	R/W	W9200	You have to trigger R9200 to activate this function after the register is set. Bit 0: none Bit 1: Disable IMP software and operation system Bit 2: Stop PLC operation
System function activation	R/W	R9200	Activate the system functions

■ Analog input

Function	Property	No.	Description
Analog input value (CH1)	R	W9800	Set the channel input range of AD module in accordance with Quick Start interface setting. See below for the corresponding resolution. Voltage mode (0 ~ 5 V): 0 ~ 5000 (unit: mV) Voltage mode(0 ~ 10 V): 0 ~ 10000 (unit: mV) Voltage mode (-5 ~ 5 V): -5000 ~ 5000 (unit: mV) Voltage mode (-10 ~ 10 V): -10000 ~ 10000 (unit: mV) Current mode (0 ~ 20 mA): 0 ~ 20000 (unit: 0.001 mA)
Analog input value (CH2)	R	W9801	
Analog input value (CH3)	R	W9802	
Analog input value (CH4)	R	W9803	
Analog input value (CH5)	R	W9804	
Analog input value (CH6)	R	W9805	
Analog input value (CH7)	R	W9806	
Analog input value (CH8)	R	W9807	
Analog input value (CH9)	R	W9808	
Analog input value (CH10)	R	W9809	
Analog input value (CH11)	R	W9810	
Analog input value (CH12)	R	W9811	
Analog input value (CH13)	R	W9812	
Analog input value (CH14)	R	W9813	
Analog input value (CH15)	R	W9814	
Analog input value (CH16)	R	W9815	
Analog input value (CH17)	R	W9816	
Analog input value (CH18)	R	W9817	
Analog input value (CH19)	R	W9818	
Analog input value (CH20)	R	W9819	
Analog input value (CH21)	R	W9820	
Analog input value (CH22)	R	W9821	
Analog input value (CH23)	R	W9822	
Analog input value (CH24)	R	W9823	
Analog input value (CH25)	R	W9824	
Analog input value (CH26)	R	W9825	

Function	Property	No.	Description
Analog input value (CH27)	R	W9826	
Analog input value (CH28)	R	W9827	
Analog input value (CH29)	R	W9828	
Analog input value (CH30)	R	W9829	
Analog input value (CH31)	R	W9830	
Analog input value (CH32)	R	W9831	

- Analog output

Function	Property	No.	Description
Analog output value (CH1)	R/W	W9900	Set the channel output range of DA module in accordance with Quick Start interface setting. See below for the corresponding resolution. Voltage mode (0 ~ 5 V): 0 ~ 5000 (unit: mV) Voltage mode(0 ~ 10 V): 0 ~ 10000 (unit: mV) Voltage mode (-5 ~ 5 V): -5000 ~ 5000 (unit: mV) Voltage mode (-10 ~ 10 V): -10000 ~ 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 ~ 24mA): 0 ~ 24000 (unit: 0.001 mA)
Analog output value (CH2)	R/W	W9901	
Analog output value (CH3)	R/W	W9902	
Analog output value (CH4)	R/W	W9903	
Analog output value (CH5)	R/W	W9904	
Analog output value (CH6)	R/W	W9905	
Analog output value (CH7)	R/W	W9906	
Analog output value (CH8)	R/W	W9907	
Analog output value (CH9)	R/W	W9908	
Analog output value (CH10)	R/W	W9909	
Analog output value (CH11)	R/W	W9910	
Analog output value (CH12)	R/W	W9911	
Analog output value (CH13)	R/W	W9912	
Analog output value (CH14)	R/W	W9913	
Analog output value (CH15)	R/W	W9914	
Analog output value (CH16)	R/W	W9915	
Analog output value (CH17)	R/W	W9916	
Analog output value (CH18)	R/W	W9917	
Analog output value (CH19)	R/W	W9918	
Analog output value (CH20)	R/W	W9919	
Analog output value (CH21)	R/W	W9920	
Analog output value (CH22)	R/W	W9921	
Analog output value (CH23)	R/W	W9922	
Analog output value (CH24)	R/W	W9923	

Function	Property	No.	Description
Analog output value (CH25)	R/W	W9924	
Analog output value (CH26)	R/W	W9925	
Analog output value (CH27)	R/W	W9926	
Analog output value (CH28)	R/W	W9927	
Analog output value (CH29)	R/W	W9928	
Analog output value (CH30)	R/W	W9929	
Analog output value (CH31)	R/W	W9930	
Analog output value (CH32)	R/W	W9931	

A

Special register (W, R) of single-axis motion

■ Special register (W) of single-axis motion status

Function	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	~	Axis 36
Motor feedback position	R	W10102	W10202	W10302	W10402	W10502	W10602	~	W13602
	R	W10103	W10203	W10303	W10403	W10503	W10603	~	W13603
Command position	R	W10104	W10204	W10304	W10404	W10504	W10604	~	W13604
	R	W10105	W10205	W10305	W10405	W10505	W10605	~	W13605
Target position	R	W10106	W10206	W10306	W10406	W10506	W10606	~	W13606
	R	W10107	W10207	W10307	W10407	W10507	W10607	~	W13607
Servo DI status	R	W10108	W10208	W10308	W10408	W10508	W10608	~	W13608
Servo DO status	R	W10109	W10209	W10309	W10409	W10509	W10609	~	W13609
Current motion speed of each axis	R	W10110	W10210	W10310	W10410	W10510	W10610	~	W13610
	R	W10111	W10211	W10311	W10411	W10511	W10611	~	W13611
Current output torque of motor	R	W10113	W10213	W10313	W10413	W10513	W10613	~	W13613
Command status	R	W10114	W10214	W10314	W10414	W10514	W10614	~	W13614
Servo error code	R	W10115	W10215	W10315	W10415	W10515	W10615	~	W13615
Current motor speed (RPM)	R	W10119	W10219	W10319	W10419	W10519	W10619	~	W13619
	R	W10120	W10220	W10320	W10420	W10520	W10620	~	W13620
Single-axis motion error code	R	W10150	W10250	W10350	W10450	W10550	W10650	~	W13650

■ Special register (W) of single-axis motion control

Function	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	~	Axis 36
Single motion control code	R/W	W10151	W10251	W10351	W10451	W10551	W10651	~	W13651
Acceleration time	R/W	W10152	W10252	W10352	W10452	W10552	W10652	~	W13652
	R/W	W10153	W10253	W10353	W10453	W10553	W10653	~	W13653
Deceleration time	R/W	W10154	W10254	W10354	W10454	W10554	W10654	~	W13654
	R/W	W10155	W10255	W10355	W10455	W10555	W10655	~	W13655
Target speed of motion command	R/W	W10156	W10256	W10356	W10456	W10556	W10656	~	W13656
	R/W	W10157	W10257	W10357	W10457	W10557	W10657	~	W13657
Target coordinates of motion command	R/W	W10158	W10258	W10358	W10458	W10558	W10658	~	W13658
	R/W	W10159	W10259	W10359	W10459	W10559	W10659	~	W13659
Homing mode setting	R/W	W10160	W10260	W10360	W10460	W10560	W10660	~	W13660
Setting of motion speed unit	R/W	W10161	W10261	W10361	W10461	W10561	W10661	~	W13661
1st speed in homing mode	R/W	W10162	W10262	W10362	W10462	W10562	W10662	~	W13662

Function	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	~	Axis 36
	R/W	W10163	W10263	W10363	W10463	W10563	W10663	~	W13663
2^{nd} speed in homing mode	R/W	W10164	W10264	W10364	W10464	W10564	W10664	~	W13664
	R/W	W10165	W10265	W10365	W10465	W10565	W10665	~	W13665
Offset in homing mode	R/W	W10166	W10266	W10366	W10466	W10566	W10666	~	W13666
	R/W	W10167	W10267	W10367	W10467	W10567	W10667	~	W13667
Target speed	R/W	W10170	W10270	W10370	W10470	W10570	W10670	~	W13670
	R/W	W10171	W10271	W10371	W10471	W10571	W10671	~	W13671
Target torque	R/W	W10172	W10272	W10372	W10472	W10572	W10672	~	W13672
	R/W	W10173	W10273	W10373	W10473	W10573	W10673	~	W13673
Torque limit in speed mode	R/W	W10174	W10274	W10374	W10474	W10574	W10674	~	W13674
	R/W	W10175	W10275	W10375	W10475	W10575	W10675	~	W13675
Speed limit in torque mode	R/W	W10176	W10276	W10376	W10476	W10576	W10676	~	W13676
	R/W	W10177	W10277	W10377	W10477	W10577	W10677	~	W13677

■ Single-axis special register

Function	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	~	Axis 36
Servo operation mode	R	R10100	R10200	R10300	R10400	R10500	R10600	~	R13600
	R	R10101	R10201	R10301	R10401	R10501	R10601	~	R13601
	R	R10102	R10202	R10302	R10402	R10502	R10602	~	R13602
	R	R10103	R10203	R10303	R10403	R10503	R10603	~	R13603
DI3 status (servo)	R	R10104	R10204	R10304	R10404	R10504	R10604	~	R13604
Servo alarm flag	R	R10105	R10205	R10305	R10405	R10505	R10605	~	R13605
SVON monitoring flag	R	R10108	R10208	R10308	R10408	R10508	R10608	~	R13608
Servo error flag	R	R10109	R10209	R10309	R10409	R10509	R10609	~	R13609
Positioning complete flag	R	R10110	R10210	R10310	R10410	R10510	R10610	~	R13610
Servo operation mode (Mode Specific)	R	R10112	R10212	R10312	R10412	R10512	R10612	~	R13612
	R	R10113	R10213	R10313	R10413	R10513	R10613	~	R13613
SVON control	R/W	R10151	R10251	R10351	R10451	R10551	R10651	~	R13651
Software limit enabling bit	R/W	R10152	R10252	R10352	R10452	R10552	R10652	~	R13652
Motion curve setting	R/W	R10161	R10261	R10361	R10461	R10561	R10661	~	R13661
JOG direction control	R/W	R10162	R10262	R10362	R10462	R10562	R10662	~	R13662
Enabling bit for torque limit in speed mode	R/W	R10163	R10263	R10363	R10463	R10563	R10663	~	R13663
Enabling bit for	R/W	R10164	R10264	R10364	R10464	R10564	R10664	~	R13664

A

Function	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	~	Axis 36
speed limit in torque mode									
Motion in process flag	R	R10165	R10265	R10365	R10465	R10565	R10665	~	R13665
Motion command ready but not being executed	R	R10166	R10266	R10366	R10466	R10566	R10666	~	R13666

■ Read/Write parameters of single-axis servo

Function	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	~	Axis 36
Read servo return value	R	W10116	W10216	W10316	W10416	W10516	W10616	~	W13616
	R	W10117	W10217	W10317	W10417	W10517	W10617	~	W13617
Servo parameter reading/writing error	R	W10121	W10221	W10321	W10421	W10521	W10621	~	W13621
Written value of the servo	R/W	W10180	W10280	W10380	W10480	W10580	W10680	~	W13680
	R/W	W10181	W10281	W10381	W10481	W10581	W10681	~	W13681
Servo parameter group and index value	R/W	W10186	W10286	W10386	W10486	W10586	W10686	~	W13686
Control code for reading/writing servo parameters	R/W	W10187	W10287	W10387	W10487	W10587	W10687	~	W13687
Set servo monitoring parameters	R/W	W10188	W10288	W10388	W10488	W10588	W10688	~	W13688

■ Single axis software limit

Function	Property	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	~	Axis 36
Software forward limit	R/W	W10182	W10282	W10382	W10482	W10582	W10682	~	W13682
	R/W	W10183	W10283	W10383	W10483	W10583	W10683	~	W13683
Software reverse limit	R/W	W10184	W10284	W10384	W10484	W10584	W10684	~	W13684
	R/W	W10185	W10285	W10385	W10485	W10585	W10685	~	W13685
Bit for triggering servo limit (Positive)	R	R10114	R10214	R10314	R10414	R10514	R10614	~	R13614
Bit for triggering servo limit (Negative)	R	R10115	R10215	R10315	R10415	R10515	R10615	~	R13615
Bit for triggering software limit (Positive)	R/W	R10116	R10216	R10316	R10416	R10516	R10616	~	R13616
Bit for triggering software limit (Negative)	R/W	R10117	R10217	R10317	R10417	R10517	R10617	~	R13617

Special Register (W, R) of Servo Group

Function	Property	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	~	Group 40
Group motion control code	R/W	W20000	W20100	W20200	W20300	W20400	W20500	~	W23900
Speed unit of group motion	R/W	W20001	W20101	W20201	W20301	W20401	W20501	~	W23901
Card number used by group motion	R/W	W20002	W20102	W20202	W20302	W20402	W20502	~	W23902
Selected axes in the group (bit)	R/W	W20003	W20103	W20203	W20303	W20403	W20503	~	W23903
Change speed during motion	R/W	W20012	W20112	W20212	W20312	W20412	W20512	~	W23912
	R/W	W20013	W20113	W20213	W20313	W20413	W20513	~	W23913
Max. speed of group motion	R/W	W20014	W20114	W20214	W20314	W20414	W20514	~	W23914
	R/W	W20015	W20115	W20215	W20315	W20415	W20515	~	W23915
Acceleration time of group motion	R/W	W20016	W20116	W20216	W20316	W20416	W20516	~	W23916
	R/W	W20017	W20117	W20217	W20317	W20417	W20517	~	W23917
Deceleration time of group motion	R/W	W20018	W20118	W20218	W20318	W20418	W20518	~	W23918
	R/W	W20019	W20119	W20219	W20319	W20419	W20519	~	W23919
Arc angle of group motion	R/W	W20020	W20120	W20220	W20320	W20420	W20520	~	W23920
Direction of motion group	R/W	W20021	W20121	W20221	W20321	W20421	W20521	~	W23921
X-coordinate of arc center	R/W	W20022	W20122	W20222	W20322	W20422	W20522	~	W23922
	R/W	W20023	W20123	W20223	W20323	W20423	W20523	~	W23923
Y-coordinate of arc center	R/W	W20024	W20124	W20224	W20324	W20424	W20524	~	W23924
	R/W	W20025	W20125	W20225	W20325	W20425	W20525	~	W23925
X-coordinate of arc end point	R/W	W20026	W20126	W20226	W20326	W20426	W20526	~	W23926
	R/W	W20027	W20127	W20227	W20327	W20427	W20527	~	W23927
Y- coordinate of arc end point	R/W	W20028	W20128	W20228	W20328	W20428	W20528	~	W23928
	R/W	W20029	W20129	W20229	W20329	W20429	W20529	~	W23929
Total spiral pitch of the three axes	R/W	W20030	W20130	W20230	W20330	W20430	W20530	~	W23930
	R/W	W20031	W20131	W20231	W20331	W20431	W20531	~	W23931
Spiral pitch of the three axes	R/W	W20032	W20132	W20232	W20332	W20432	W20532	~	W23932
	R/W	W20033	W20133	W20233	W20333	W20433	W20533	~	W23933
Spindle tapping speed	R/W	W20051	W20151	W20251	W20351	W20451	W20551	~	W23951
Spindle retrieving speed	R/W	W20052	W20152	W20252	W20352	W20452	W20552	~	W23952
Tapping pitch	R/W	W20053	W20153	W20253	W20353	W20453	W20553	~	W23953
Delay time after tapping completed	R/W	W20054	W20154	W20254	W20354	W20454	W20554	~	W23954
Tapping depth	R/W	W20056	W20156	W20256	W20356	W20456	W20556	~	W23956

A

Function	Property	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	~	Group 40
	R/W	W20057	W20157	W20257	W20357	W20457	W20557	~	W23957
Target value of the 1st axis	R/W	W20070	W20170	W20270	W20370	W20470	W20570	~	W23970
	R/W	W20071	W20171	W20271	W20371	W20471	W20571	~	W23971
Target value of the 2nd axis	R/W	W20072	W20172	W20272	W20372	W20472	W20572	~	W23972
	R/W	W20073	W20173	W20273	W20373	W20473	W20573	~	W23973
Target value of the 3rd axis	R/W	W20074	W20174	W20274	W20374	W20474	W20574	~	W23974
	R/W	W20075	W20175	W20275	W20375	W20475	W20575	~	W23975
Target value of the 4th axis	R/W	W20076	W20176	W20276	W20376	W20476	W20576	~	W23976
	R/W	W20077	W20177	W20277	W20377	W20477	W20577	~	W23977
Target value of the 5th axis	R/W	W20078	W20178	W20278	W20378	W20478	W20578	~	W23978
	R/W	W20079	W20179	W20279	W20379	W20479	W20579	~	W23979
Target value of the 6th axis	R/W	W20080	W20180	W20280	W20380	W20480	W20580	~	W23980
	R/W	W20081	W20181	W20281	W20381	W20481	W20581	~	W23981
Target value of the 7th axis	R/W	W20082	W20182	W20282	W20382	W20482	W20582	~	W23982
	R/W	W20083	W20183	W20283	W20383	W20483	W20583	~	W23983
Target value of the 8th axis	R/W	W20084	W20184	W20284	W20384	W20484	W20584	~	W23984
	R/W	W20085	W20185	W20285	W20385	W20485	W20585	~	W23985
Target value of the 9th axis	R/W	W20086	W20186	W20286	W20386	W20486	W20586	~	W23986
	R/W	W20087	W20187	W20287	W20387	W20487	W20587	~	W23987
Target value of the 10th axis	R/W	W20088	W20188	W20288	W20388	W20488	W20588	~	W23988
	R/W	W20089	W20189	W20289	W20389	W20489	W20589	~	W23989
Target value of the 11th axis	R/W	W20090	W20190	W20290	W20390	W20490	W20590	~	W23990
	R/W	W20091	W20191	W20291	W20391	W20491	W20591	~	W23991
Target value of the 12th axis	R/W	W20092	W20192	W20292	W20392	W20492	W20592	~	W23992
	R/W	W20093	W20193	W20293	W20393	W20493	W20593	~	W23993
Interpolation error coder	R	W20095	W20195	W20295	W20395	W20495	W20595	~	W23995
Group motion in process	R	R20000	R20100	R20200	R20300	R20400	R20500	~	R23900
Acceleration curve of group motion	R/W	R20010	R20110	R20210	R20310	R20410	R20510	~	R23910

A

Special Register (W, R) of Motion Program Macro (MPM)

Function	Property	MPM 1	MPM 2	MPM 3	MPM 4	MPM 5	MPM 6	~	MPM 100
Command control code	R/W	W30000	W30010	W30020	W30030	W30040	W30050	~	W30990
Total line number	R/W	W30001	W30011	W30021	W30031	W30041	W30051	~	W30991
The line number that currently being executed	R/W	W30002	W30012	W30022	W30032	W30042	W30052	~	W30992
Feedrate	R/W	W30003	W30013	W30023	W30033	W30043	W30053	~	W30993
Specify the wrong line	R	W30007	W30017	W30027	W30037	W30047	W30057	~	W30997
Syntax error code	R	W30008	W30018	W30028	W30038	W30048	W30058	~	W30998
Execution error code	R	W30009	W30019	W30029	W30039	W30049	W30059		W30999
Stepping flag	R/W	R30000	R30010	R30020	R30030	R30040	R30050		R30990
Stepping activate	R/W	R30001	R30011	R30021	R30031	R30041	R30051		R30991

Special Register (W, R) of SNC

■ SNC system control(W)

Function	Property	SNC	SNC 2	SNC 3	SNC4	Description
Command control code	R/W	W31000	W33000	W35000	W37000	0: No Command 10: Start machining 11: Pause machining 12: Resume machining (The function is valid when machining is paused before) 13: Stop machining 14: Start the procedure of tool length measurement 15: Stop the procedure of tool length measurement
Reload system parameters	R/W	W31001	W33001	W35001	W37001	It may require restarting IMP software to validate the parameters. 0: No Command 1: Reload SNC Config parameters
No. of machining platform to be processed	R/W	W31002	W33002	W35002	W37002	No. of machining platform that is waiting to be processed of SNC.
No. of machining platform being processed	R	W31003	W33003	W35003	W37003	No. of machining platform that is being processed of SNC
Spindle number	R	W31005	W33005	W35005	W37005	Tool No. of SNC spindle: After tool change is complete,set the relays, R32998, 34998, 36998, 38998, to On. And they will inform the system that tool change is complete. Then, the system will automatically set the tool no. for tool change in accordance with the T code.
Change tool and set tool no. in manual	R/W	W31006	W33006	W35006	W37006	When tool change is complete in manual mode:

A

Function	Property	SNC	SNC 2	SNC 3	SNC4	Description
mode						SNC1: Trigger R32999 bit and W31006 will be set to W31005 SNC2: Trigger R34999 bit and W31006 will be set to W33005 SNC3: Trigger R36999 bit and W31006 will be set to W35005 SNC4: Trigger R38999 bit and W31006 will be set to W37005
Jump activate	R/W	W31007	W33007	W35007	W37007	When SNC receives the activation command, if the parameter is not set to 0 and tis corresponded jump bit is on, then the G code runs before the set line will not be executed. 0: Function disable (default) 1~: Activated line number This function should work with the following bits: SNC1: R32983 SNC2: R34983 SNC3: R36983 SNC4: R38983
	R/W	W31008	W33008	W35008	W37008	
Error code	R	W31009	W33009	W35009	W37009	SNC error code Please refer to section 9.5 for detailed descriptions.
Error type	R	W31010	W33010	W35010	W37010	SNC error type: Please refer to section 9.5 for detailed descriptions.
Operation status	R	W31011	W33011	W35011	W37011	SNC operation state 0: Stop 1: Pause 2: Running
File path (128 registers are applied)	R/W	W31100	W33100	W35100	W37100	It is the G code file path that will be executed by SNC (128 registers are applied)

A

■ SNC system control(R)

Function	Property	SNC	SNC 2	SNC 3	SNC4	Description
Movement limitation of X-axis	R/W	R32971	R34971	R36971	R38971	Off: Cut off the axial movement limit (default) On: Limit the axial movement
Movement limitation of Y-axis	R/W	R32972	R34972	R36972	R38972	-
Movement limitation of Z-axis	R/W	R32973	R34973	R36973	R38973	-
Movement limitation of A-axis	R/W	R32974	R34974	R36974	R38974	-
Movement limitation of B-axis	R/W	R32975	R34975	R36975	R38975	-
Movement limitation of C-axis	R/W	R32976	R34976	R36976	R38976	-
Movement limitation of U-axis	R/W	R32977	R34977	R36977	R38977	-
Movement limitation of V-axis	R/W	R32978	R34978	R36978	R38978	-
Movement limitation of W-axis	R/W	R32979	R34979	R36979	R38979	-
Single-step enabling bit	R/W	R32981	R34981	R36981	R38981	This function is valid only before SNC operation. Off: G code is executed continuously (default). On: The system only executes one single line of G code for one time. When it pauses, users should trigger the single-step bit again for another line.
Single-step execution bit	R/W	R32982	R34982	R36982	R38982	-
Jump enabling bit	R/W	R32983	R34983	R36983	R38983	This function is valid only before SNC operation and should work with the following conditions: SNC1 should work with W31007 SNC2 should work with W33007 SNC3 should work with W35007 SNC4 should work with W37007

A

■ M, T code and external macro

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Control code	R/W	W31015	W33015	W35015	W37015	Set the file no. that is ready to call. Then, set the control code according to the type of O, T macro and SNC starts to execute external macro procedure. If the execution is failed, the control code will be changed to 99 automatically. 0: none 1: Call ""O"” macro 2: Call ""T"” macro ----------------------------------- 99: No file is found.
File number	R/W	W31016	W33016	W35016	W37016	
M code (00)	R/W	R31000	R33000	R35000	R37000	-
~		~	~	~	~	
M code (999)	R/W	R31999	R33999	R35999	R37999	
T code (01)	R/W	R32001	R34001	R36001	R38001	-
~		~	~	~	~	
T code (100)	R/W	R32100	R34100	R36100	R38100	

■ Register (W) of external macro

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Write to #1833	R/W	W32301	W34301	W36301	W38301	Write the value of PLC special register to the register of external macro
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	

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Read #1896	R	W32317	W34317	W36317	W38317	Read the value of the register of external macro via PLC special register
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	

- Relay (R) of external macro

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Write to #1801	R/W	R32301	R34301	R36301	R38301	Write the value of PLC special register to the register of external macro
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	

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
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	
Write to #1833	R/W	R32333	R34333	R36333	R38333	
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 the value of the register of external macro via PLC special register
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	

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
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	

- Spindle speed control

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Spindle speed control	R	W31020	W33020	W35020	W37020	When SNC interpreter reads the S code, the number after S code will be saved to the register.

- Read tool information

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Control code	-	W31029	W33029	W35029	W37029	Read tool information 0: none 1: Read tool information 99: Accessing failed
Target tool number	R/W	W31030	W33030	W35030	W37030	Read the target tool number (1~100) in advance
Read tool length	R	W31031	W33031	W35031	W37031	Read tool length
	R	W31032	W33032	W35032	W37032	
Read tool radius	R	W31033	W33033	W35033	W37033	Read tool radius
	R	W31034	W33034	W35034	W37034	
Read the tool offset value of X-axis	R	W31035	W33035	W35035	W37035	Read the tool offset value of X-axis

A

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
	R	W31036	W33036	W35036	W37036	
Read the tool offset value of Y-axis	R	W31037	W33037	W35037	W37037	Read the tool offset value of Y-axis
	R	W31038	W33038	W35038	W37038	
Read the tool offset value of Z-axis	R	W31039	W33039	W35039	W37039	Read the tool offset value of Z-axis
	R	W31040	W33040	W35040	W37040	

■ Error type

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Total error number	R	W32106	W34106	W36106	W38106	Total error number of G code
Total warning number	R	W32108	W34108	W36108	W38108	Total warning number of G code
Axial error alarm	R	W32161	W34161	W36161	W38161	Axial error alarm
G code error number	R	W32162	W34162	W36162	W38162	Please refer to section 9.5 for detailed descriptions of error code.
Software limit trigger	R	W32163	W34163	W36163	W38163	It displays the state when triggering software limit.
API error	R	W32164	W34164	W36164	W38164	Wrong returned value of API. Please refer to section 9.5 for detailed descriptions of error code.
Setting error	R	W32165	W34165	W36165	W38165	It displays the setting error code. Please refer to section 9.5 for detailed descriptions of error code.
Device error	R	W32166	W34166	W36166	W38166	It displays the device error code. Please refer to section 9.5 for detailed descriptions of error code.
System error code	R	W32170	W34170	W36170	W38170	It displays the system error code. Please refer to section 9.5 for detailed descriptions of error code.
Tool error	R	W32171	W34171	W36171	W38171	It displays tool error code. Please refer to section 9.5 for detailed descriptions of error code.
Wrong line number of G code	R	W32178	W34178	W36178	W38178	It displays the wrong line of G code.
	R	W32179	W34179	W36179	W38179	

■ G code interpreter

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Interpreting line number	R	W32180	W34180	W36180	W38180	The line number that has been interpreted by G code interpreter
	R	W32181	W34181	W36181	W38181	
The executed line number	R	W32187	W34187	W36187	W38187	The line number that has been executed.
	R	W32188	W34188	W36188	W38188	
Total line number	R	W32193	W34193	W36193	W38193	Total line number of G code
	R	W32194	W34194	W36194	W38194	
Estimate total timespan	R	W32230	W34230	W36230	W38230	Estimate the total time consumed when executing G code
Execution time	R	W32231	W34231	W36231	W38231	The execution time when execution G code
Remaining time	R	W32233	W34233	W36233	W38233	Estimate the remaining time when executing G code
Current execution rate (%)	R	W32236	W34236	W36236	W38236	The processing percentage (0 ~ 100) when executing G code. (The progress of executing G code in percentage. (0 ~ 100))

■ Speed control (W)

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Max. speed limit of G00	R/W	W32478	W34478	W36478	W38478	Function of max. speed (percentage) of G00 should work with R32996.
Current feedrate (%)	R	W32479	W34479	W36479	W38479	Current feedrate of SNC operation
Target feedrate (%)	R/W	W32480	W34480	W36480	W38480	Write the target feedrate (percentage) to 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	W32774	W34774	W36774	W38774	Read the current feeding speed of SNC
	R	W32775	W34775	W36775	W38775	

■ Speed control (R)

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Enabling bit of MPG simulation	R/W	R32995	R34995	R36995	R38995	When MPG simulation is disabled, the group feedrate of SNC will be set to 0 automatically.
Enabling bit of G00 feedrate setting	R/W	R32996	R34996	R36996	R38996	On: G00 feedrate adjustment enabled. Users can adjust the feedrate through the registers(W) below: SNC 1: W31026 SNC 2: W33026 SNC 3: W35026 SNC 4: W37026
Enabling bit of SNC feedrate setting	R/W	R32997	R34997	R36997	R38997	On: Adjust SNC feedrate through the register (W) below: SNC1: W32479 SNC1: W34479 SNC1: W36479 SNC1: W38479

A

■ Coordinates(W)

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Feedback position of mechanical coordinate on X-axis	R	W32520	W34520	W36520	W38520	The encoder's feedback position of each axis in mechanical coordinate system (Feedback / Gear)
	R	W32521	W34521	W36521	W38521	
Feedback position of mechanical coordinate on Y-axis	R	W32522	W34522	W36522	W38522	
	R	W32523	W34523	W36523	W38523	
Feedback position of mechanical coordinate on Z-axis	R	W32524	W34524	W36524	W38524	
	R	W32525	W34525	W36525	W38525	
Feedback position of mechanical coordinate on A-axis	R	W32526	W34526	W36526	W38526	
	R	W32527	W34527	W36527	W38527	
Feedback position of mechanical coordinate on B-axis	R	W32528	W34528	W36528	W38528	
	R	W32529	W34529	W36529	W38529	
Feedback position of mechanical coordinate on C-axis	R	W32530	W34530	W36530	W38530	
	R	W32531	W34531	W36531	W38531	
Feedback position of mechanical coordinate on U-axis	R	W32532	W34532	W36532	W38532	
	R	W32533	W34533	W36533	W38533	
Feedback position of mechanical coordinate on V-axis	R	W32534	W34534	W36534	W38534	
	R	W32535	W34535	W36535	W38535	
Feedback position of mechanical coordinate on W-axis	R	W32536	W34536	W36536	W38536	
	R	W32537	W34537	W36537	W38537	
Command position of mechanical coordinate on X-axis	R	W32538	W34538	W36538	W38538	Command position of each axis in mechanical coordinate system (Command / Gear)
	R	W32539	W34539	W36539	W38539	
Command position of mechanical coordinate on Y-axis	R	W32540	W34540	W36540	W38540	
	R	W32541	W34541	W36541	W38541	
Command position of mechanical coordinate on Z-axis	R	W32542	W34542	W36542	W38542	
	R	W32543	W34543	W36543	W38543	
Command position of mechanical coordinate on A-axis	R	W32544	W34544	W36544	W38544	
	R	W32545	W34545	W36545	W38545	
Command position of mechanical coordinate on B-axis	R	W32546	W34546	W36546	W38546	
	R	W32547	W34547	W36547	W38547	
Command position of mechanical coordinate on C-axis	R	W32548	W34548	W36548	W38548	
	R	W32549	W34549	W36549	W38549	
Command position of mechanical coordinate on U-axis	R	W32550	W34550	W36550	W38550	
	R	W32551	W34551	W36551	W38551	
Command position of mechanical coordinate	R	W32552	W34552	W36552	W38552	

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
on V-axis	R	W32553	W34553	W36553	W38553	
Command position of mechanical coordinate on W-axis	R	W32554	W34554	W36554	W38554	
	R	W32555	W34555	W36555	W38555	
Working coordinate of X-axis	R	W32556	W34556	W36556	W38556	Working coordinate of each axis (Working coordinate = mechanical coordinate - the set coordinate of working plane)
	R	W32557	W34557	W36557	W38557	
Working coordinate of Y-axis	R	W32558	W34558	W36558	W38558	
	R	W32559	W34559	W36559	W38559	
Working coordinate of Z-axis	R	W32560	W34560	W36560	W38560	
	R	W32561	W34561	W36561	W38561	
Working coordinate of A-axis	R	W32562	W34562	W36562	W38562	
	R	W32563	W34563	W36563	W38563	
Working coordinate of B-axis	R	W32564	W34564	W36564	W38564	
	R	W32565	W34565	W36565	W38565	
Working coordinate of C-axis	R	W32566	W34566	W36566	W38566	
	R	W32567	W34567	W36567	W38567	
Working coordinate of U-axis	R	W32568	W34568	W36568	W38568	
	R	W32569	W34569	W36569	W38569	
Working coordinate of V-axis	R	W32570	W34570	W36570	W38570	
	R	W32571	W34571	W36571	W38571	
Working coordinate of W-axis	R	W32572	W34572	W36572	W38572	
	R	W32573	W34573	W36573	W38573	
Remaining distance of X-axis	R	W32574	W34574	W36574	W38574	Remaining distance from the target position (Remaining distance = target coordinate - mechanical coordinate)
	R	W32575	W34575	W36575	W38575	
Remaining distance of Y-axis	R	W32576	W34576	W36576	W38576	
	R	W32577	W34577	W36577	W38577	
Remaining distance of Z-axis	R	W32578	W34578	W36578	W38578	
	R	W32579	W34579	W36579	W38579	
Remaining distance of A-axis	R	W32580	W34580	W36580	W38580	
	R	W32581	W34581	W36581	W38581	
Remaining distance of B-axis	R	W32582	W34582	W36582	W38582	
	R	W32583	W34583	W36583	W38583	
Remaining distance of C-axis	R	W32584	W34584	W36584	W38584	
	R	W32585	W34585	W36585	W38585	

A

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Remaining distance of U-axis	R	W32586	W34586	W36586	W38586	
	R	W32587	W34587	W36587	W38587	
Remaining distance of V-axis	R	W32588	W34588	W36588	W38588	
	R	W32589	W34589	W36589	W38589	
Remaining distance of W-axis	R	W32590	W34590	W36590	W38590	
	R	W32591	W34591	W36591	W38591	
Start point of relative coordinate on X-axis	R	W32718	W34718	W36718	W38718	Start point of relative coordinates (The end position of the last G code)
	R	W32719	W34719	W36719	W38719	
Start point of relative coordinate on Y-axis	R	W32720	W34720	W36720	W38720	
	R	W32721	W34721	W36721	W38721	
Start point of relative coordinate on Z-axis	R	W32722	W34722	W36722	W38722	
	R	W32723	W34723	W36723	W38723	
Start point of relative coordinate on A-axis	R	W32724	W34724	W36724	W38724	
	R	W32725	W34725	W36725	W38725	
Start point of relative coordinate on B-axis	R	W32726	W34726	W36726	W38726	
	R	W32727	W34727	W36727	W38727	
Start point of relative coordinate on C-axis	R	W32728	W34728	W36728	W38728	
	R	W32729	W34729	W36729	W38729	
Start point of relative coordinate on U-axis	R	W32730	W34730	W36730	W38730	
	R	W32731	W34731	W36731	W38731	
Start point of relative coordinate on V-axis	R	W32732	W34732	W36732	W38732	
	R	W32733	W34733	W36733	W38733	
Start point of relative coordinate on W-axis	R	W32734	W34734	W36734	W38734	
	R	W32735	W34735	W36735	W38735	
Relative coordinate value of X-axis	R	W32736	W34736	W36736	W38736	Relative coordinate value of X-axis (Relative coordinate value = mechanical coordinate - start point of relative coordinate)
	R	W32737	W34737	W36737	W38737	
Relative coordinate value of Y-axis	R	W32738	W34738	W36738	W38738	
	R	W32739	W34739	W36739	W38739	
Relative coordinate value of Z-axis	R	W32740	W34740	W36740	W38740	
	R	W32741	W34741	W36741	W38741	
Relative coordinate value of A-axis	R	W32742	W34742	W36742	W38742	
	R	W32743	W34743	W36743	W38743	
Relative coordinate value of B-axis	R	W32744	W34744	W36744	W38744	

A

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
	R	W32745	W34745	W36745	W38745	
Relative coordinate value of C-axis	R	W32746	W34746	W36746	W38746	
	R	W32747	W34747	W36747	W38747	
Relative coordinate value of U-axis	R	W32748	W34748	W36748	W38748	
	R	W32749	W34749	W36749	W38749	
Relative coordinate value of V-axis	R	W32750	W34750	W36750	W38750	
	R	W32751	W34751	W36751	W38751	
Relative coordinate value of W-axis	R	W32752	W34752	W36752	W38752	
	R	W32753	W34753	W36753	W38753	
Current coordinate	R	W32754	W34754	W36754	W38754	The coordinate (G53 ~ G59) that currently applies by SNC.

■ Coordinates(R)

Function	Property	SNC	SNC 2	SNC 3	SNC 4	Description
Start point of relative coordinate on X-axis	R/W	R32986	R34986	R36986	R38986	The special register detects the rising-edge triggered signal and set current position as the start point of relative coordinate. Then the relay is set to off.
Start point of relative coordinate on Y-axis	R/W	R32987	R34987	R36987	R38987	
Start point of relative coordinate on Z-axis	R/W	R32988	R34988	R36988	R38988	
Start point of relative coordinate on A-axis	R/W	R32989	R34989	R36989	R38989	
Start point of relative coordinate on B-axis	R/W	R32990	R34990	R36990	R38990	
Start point of relative coordinate on C-axis	R/W	R32991	R34991	R36991	R38991	
Start point of relative coordinate on U-axis	R/W	R32992	R34992	R36992	R38992	
Start point of relative coordinate on V-axis	R/W	R32993	R34993	R36993	R38993	
Start point of relative coordinate on W-axis	R/W	R32994	R34994	R36994	R38994	

(This page is intentionally left blank.)

Homing Mode

B

This chapter helps with understanding the definitions of different homing methods.

List of homing methods B-2
Description of homing methods B-4

B

List of homing methods

Method	Defining homing origin	Status of limit switch
1	The first Z pulse the motor found when it runs in forward direction after reaching the negative limit	The servo drive will show error if the positive limit is reached.
2	The first Z pulse the motor found when it runs in reverse direction after reaching the positive limit	The servo drive will show error if the negative limit is reached.
3	The first Z pulse the motor found after it leaves the home switch in reverse direction	The servo drive will show error if the positive or negative limit is reached.
4	The first Z pulse the motor found after it reaches the home switch in forward direction	
5	The first Z pulse the motor found after it reaches the home switch in reverse direction	
6	The first Z pulse the motor found after it leaves the home switch in forward direction	
7	The first Z pulse the motor found after it leaves the home switch in reverse direction	If the positive limit is reached, the motor runs in opposite direction and keeps looking for the homing origin; if the negative limit is reached, the servo drive will show error.
8	The first Z pulse the motor found after it reaches the home switch in forward direction	
9	The first Z pulse the motor found after it reaches the home switch in reverse direction	
10	The first Z pulse the motor found after it leaves the home switch in forward direction	
11	The first Z pulse the motor found after it leaves the home switch in forward direction	If the negative limit is triggered, the motor runs in opposite direction and keeps looking for the homing origin; if the positive limit is reached, the servo drive will show error.
12	The first Z pulse the motor found after it reaches the home switch in reverse direction	
13	The first Z pulse the motor found after it reaches the home switch in forward direction	
14	The first Z pulse the motor found after it leaves the home switch in reverse direction	
15	Reserved	-
16	Reserved	-
17	The signal the motor found when it reaches the negative limit.	The servo drive will show error if the positive limit is reached.
18	The signal the motor found when it reaches the positive limit.	The servo drive will show error if the negative limit is reached.
19	The signal the motor found when it leaves the home switch in reverse direction	The servo drive will show error if the positive or negative limit is

B

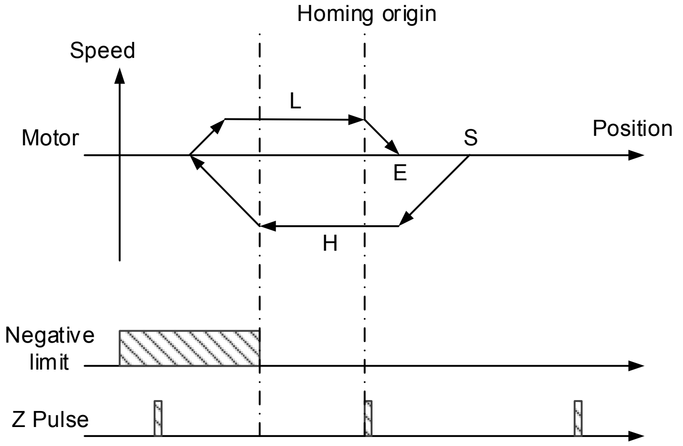
Method	Defining homing origin	Status of limit switch
20	The signal the motor found when it reaches the home switch in forward direction	reached.
21	The signal the motor found when it reaches the home switch in reverse direction	
22	The signal the motor found when it leaves the home switch in forward direction	
23	The signal the motor found when it leaves the home switch in reverse direction	If positive limit is reached, the motor runs in opposite direction and keeps looking for the homing origin; if negative limit is reached, the servo drive will show error.
24	The signal the motor found when it reaches the home switch in forward direction	
25	The signal the motor found when it reaches the home switch in reverse direction	
26	The signal the motor found when it leaves the home switch in forward direction	
27	The signal the motor found when it leaves the home switch in forward direction	If the negative limit is triggered, the motor runs in opposite direction and keeps looking for the homing origin; if positive limit is reached, the servo drive will show error.
28	The signal the motor found when it reaches the home switch in reverse direction	
29	The signal the motor found when it reaches the home switch in forward direction	
30	The signal the motor found when it leaves the home switch in reverse direction	
31	Reserved	-
32	Reserved	-
33	The first Z pulse the motor found when it runs in reverse direction	The servo drive will show error if the positive or negative limit is reached.
34	The first Z pulse the motor found when runs in forward direction	
35	The motor regards the current position as the new homing origin.	

B

Description of homing methods

1. **Method 1**

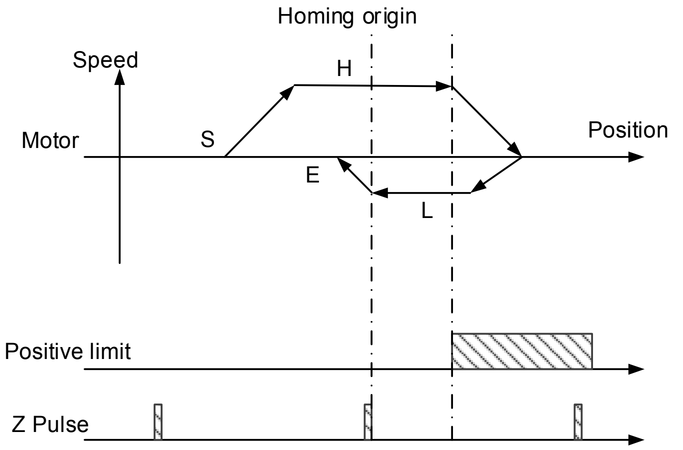
The motor runs in reverse direction at high speed until it reaches the negative limit. Then, it decelerates in forward direction. When the motor leaves the negative limit at low speed, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.



H: High speed (1st speed)
L: Low speed (2nd speed)
S: Starting point
E: Ending point
Z Pulse: Zero point of each cycle of the encoder

2. **Method 2**

The motor runs in forward direction at high speed until it reaches the positive limit. Then, it decelerates in reverse direction. When the motor leaves the positive limit at low speed, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.

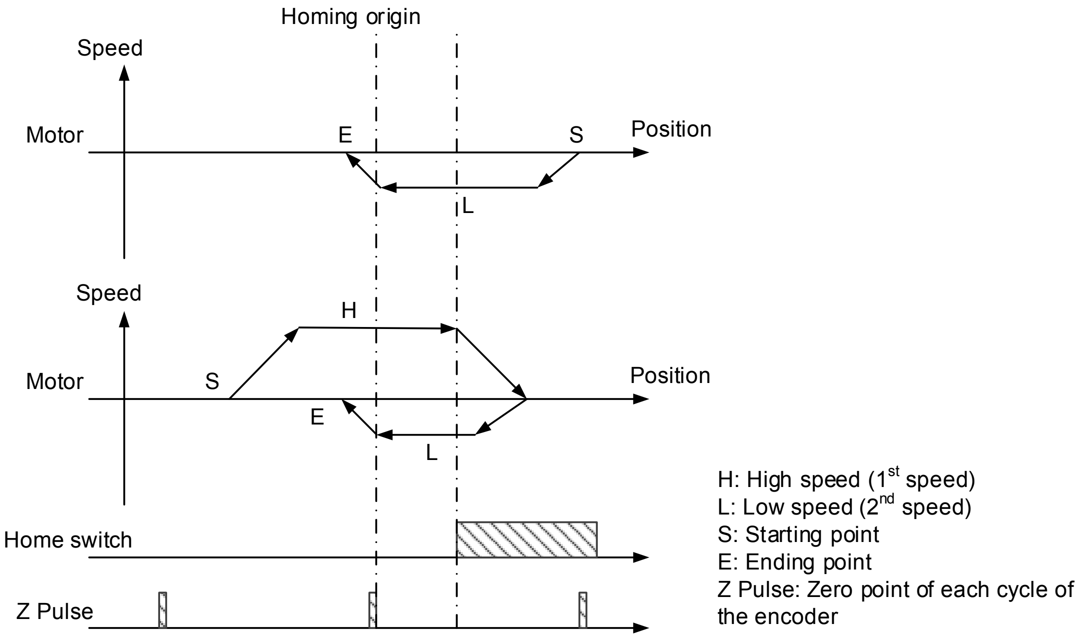


H: High speed (1st speed)
L: Low speed (2nd speed)
S: Starting point
E: Ending point
Z Pulse: Zero point of each cycle of the encoder

B

3. **Method 3**

- Home switch ON: The motor runs in reverse direction at low speed until it leaves the home switch. Then, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch OFF: The motor runs in forward direction at high speed until it reaches the home switch. Then, it decelerates in reverse direction. When the motor leaves the home switch at low speed, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.

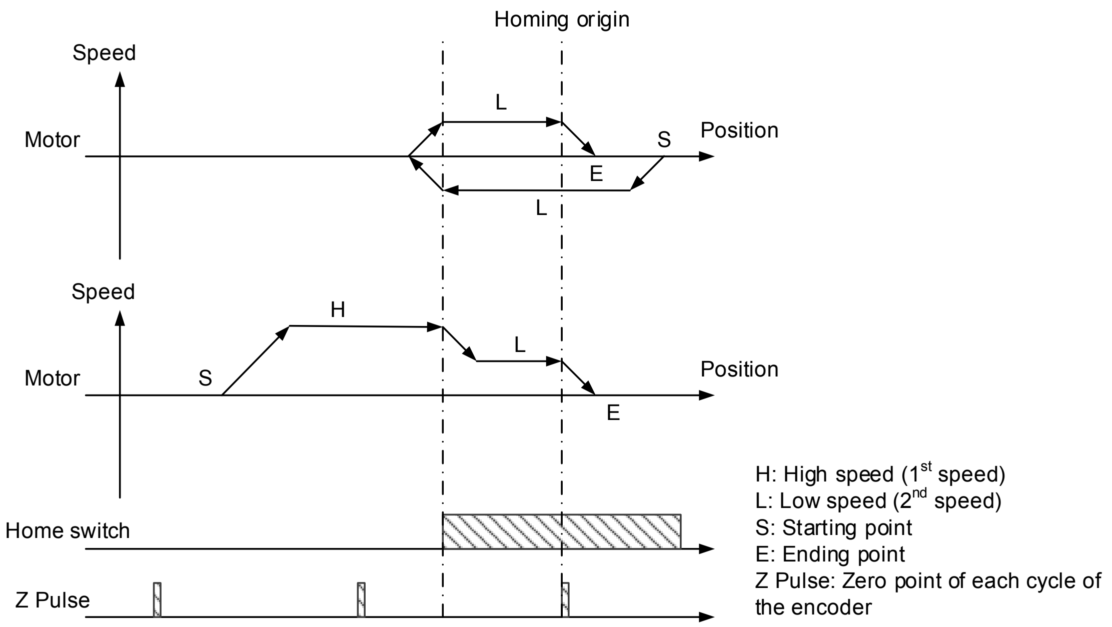


B

4. **Method 4**

Method 4 is similar to Method 3 but with different moving directions after receiving the signal changes of the home switch.

- Home switch ON: The motor runs in reverse direction at low speed until it leaves the home switch. Then, the motor reaches the home switch again in forward direction. When the motor reaches the home switch again, it will regard the first Z pulse as the new homing origin.
- Home switch OFF: The motor runs in forward direction at high speed until it reaches the home switch. Then, the motor decelerates and runs at low speed and regards the first Z pulse it looked for as the new homing origin.

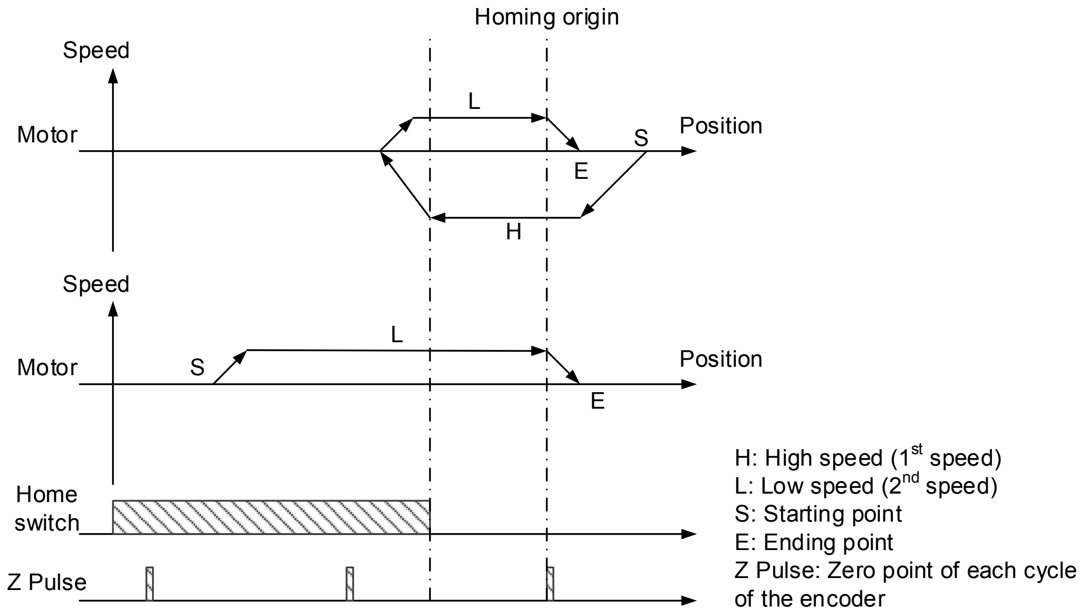


B

5. Method 5

Method 5 is similar to similar to method 3 but with different initial moving directions.

- Home switch OFF: The motor runs in reverse direction at high speed until it reaches the home switch. Then, the motor decelerates in forward direction. When the motor leaves the switch at low speed and looks for the first Z pulse, it regards the first Z pulse as the new homing origin.
- Home switch ON: The motor runs in forward direction at low speed until it leaves the home switch. Then, it looks for the first Z pulse and regards the first Z pulse as the new homing origin.

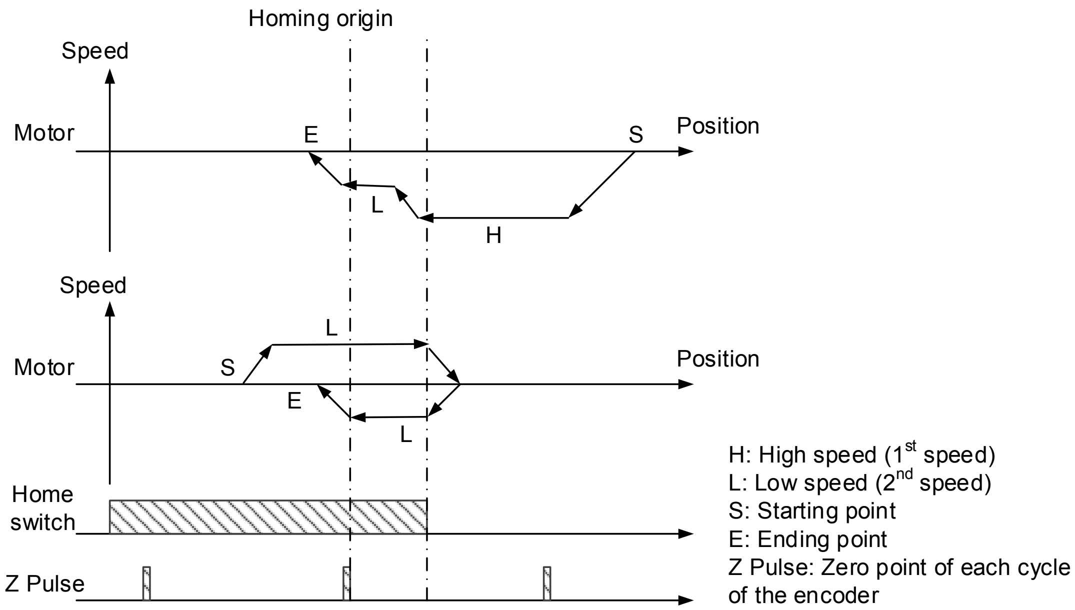


B

6. **Method 6**

Method 6 is similar to method 4 but with different initial moving directions.

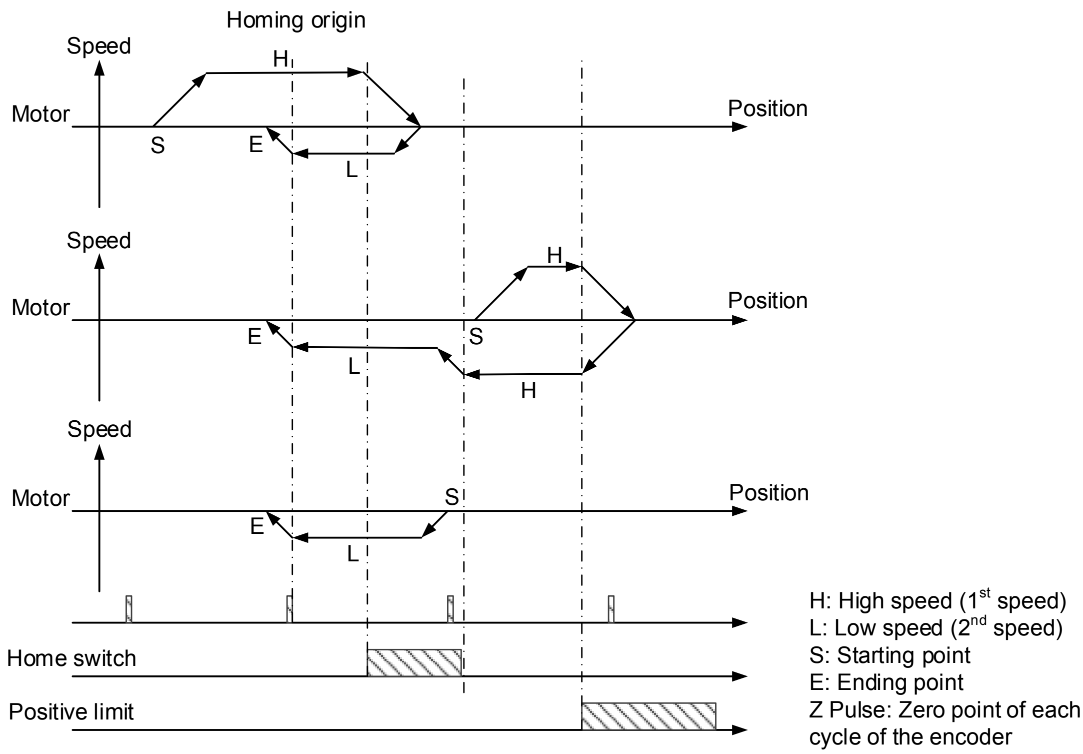
- Home switch OFF: The motor runs in reverse direction at high speed until it reaches the home switch. Then, the motor runs at low speed, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch ON: The motor runs in forward direction at low speed until it leaves the home switch. Then, the motor runs in reverse direction. When the motor reaches the home switch again, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.



B

7. **Method 7**

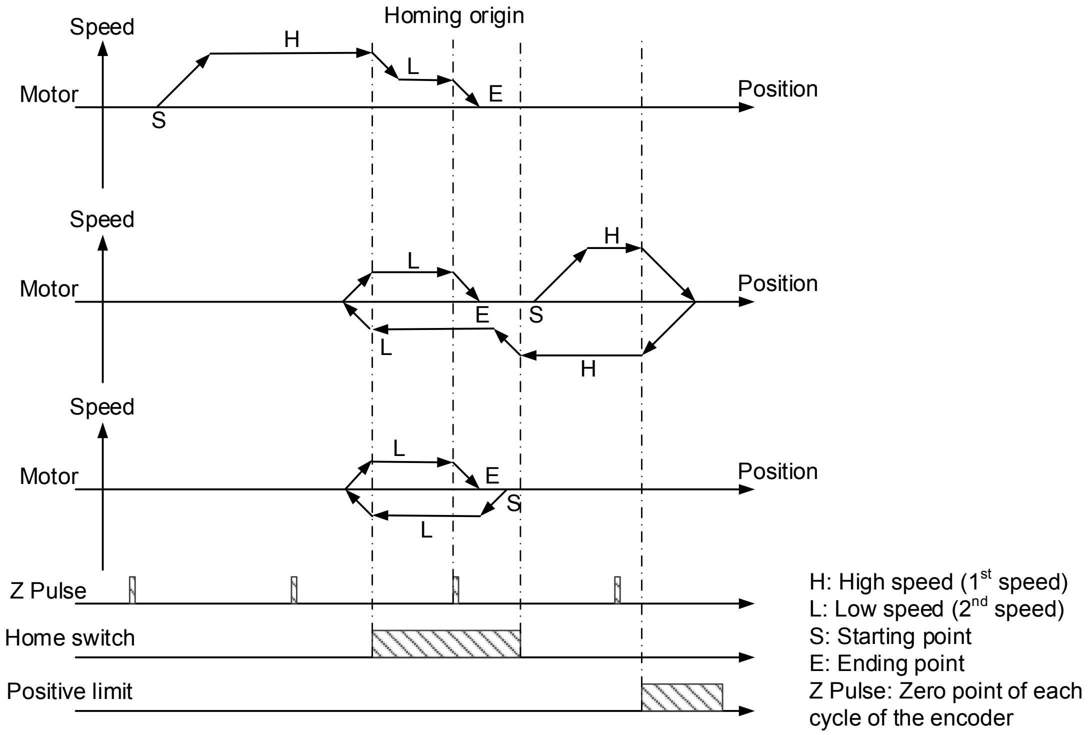
- Home switch OFF: The motor runs in forward direction at high speed until it reaches the home switch. Then, it decelerates in reverse direction. When the motor leaves the home switch at low speed, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch OFF: The motor runs in forward direction at high speed. When the motor triggers the positive limit before reaching the home switch, it runs in reverse direction until reaching the home switch. Then, the motor decelerates to low speed. When the motor leaves the home switch, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch ON: The motor runs in reverse direction at low speed until it leaves the home switch. Then, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.



B

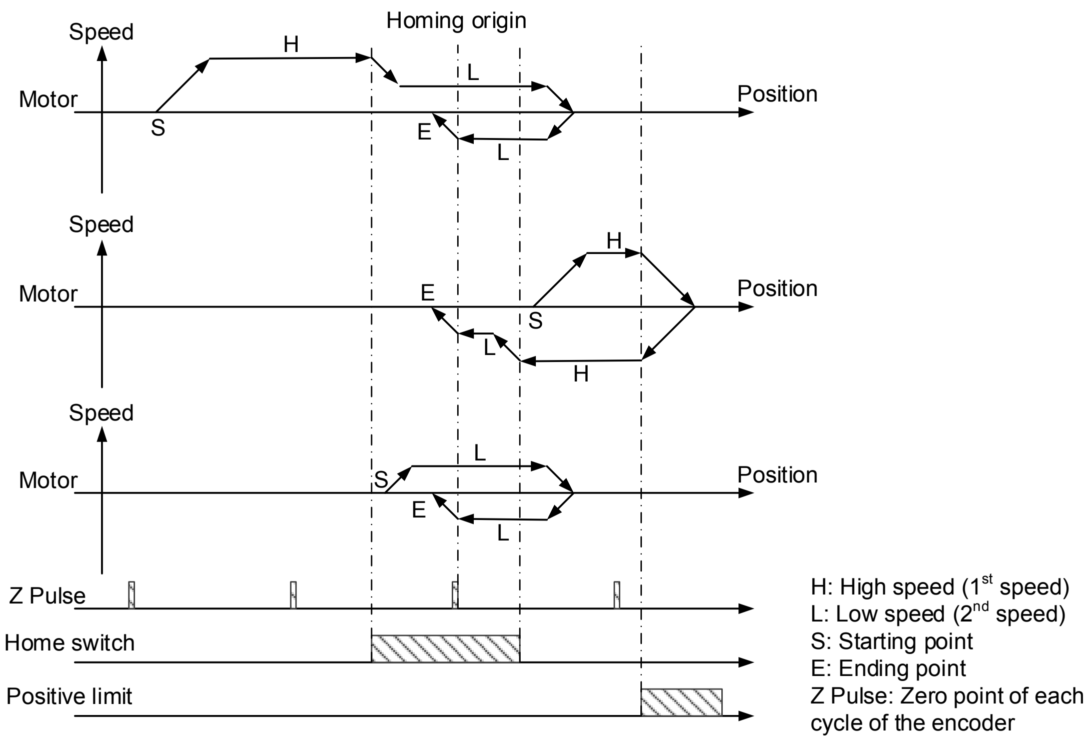
8. **Method 8**

- Home switch OFF: The motor runs in forward direction at high speed until it reaches the home switch. Then, the motor runs at low speed, starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch OFF: The motor runs in forward direction at high speed. When the motor triggers the positive limit before reaching the home switch, it runs in reverse direction until reaching the home switch. Then, the motor decelerates and leaves the home switch at low speed. Afterwards, the motor runs in forward direction. When the home switch is reached again, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch ON: The motor runs in reverse direction at low speed until it leaves the home switch. Then, it runs in forward direction. When the home switch is reached again, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.



9. **Method 9**

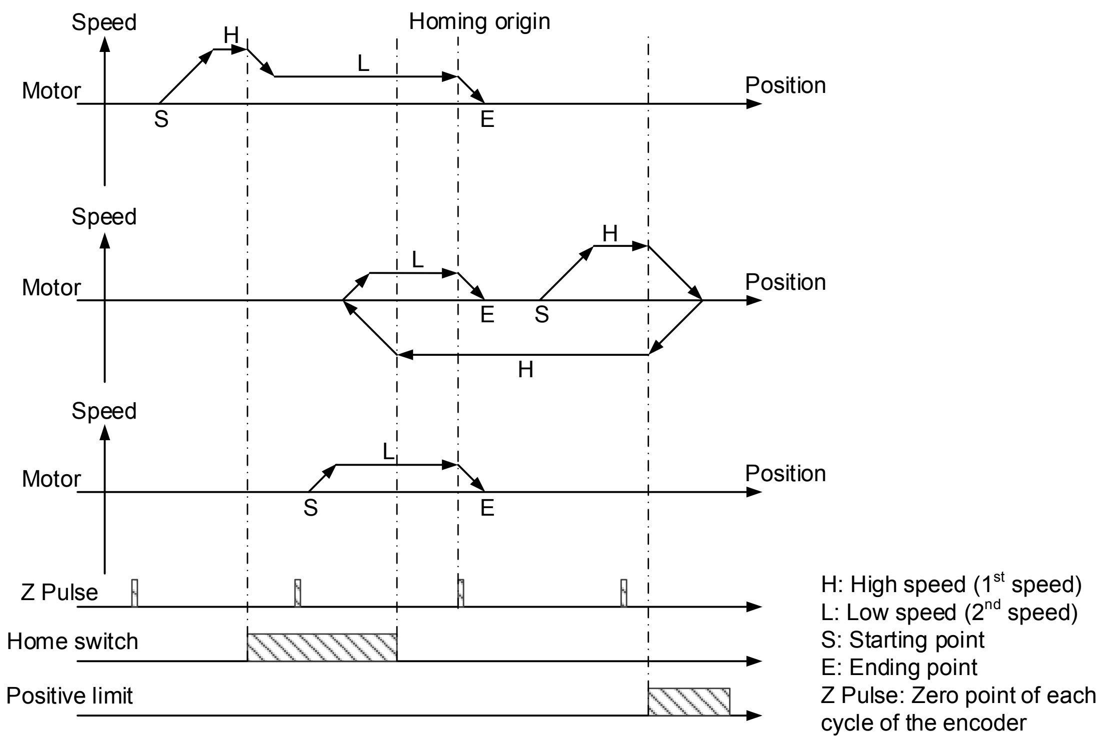
- Home switch OFF: The motor runs in forward direction at high speed until it reaches the home switch. Then, the motor decelerates and leaves the home switch at low speed. Afterwards, the motor runs in reverse direction. When the motor reaches the home switch again, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch OFF: The motor runs in forward direction at high speed. When the motor triggers the positive limit before reaching the home switch, it runs in reverse direction until reaching the home switch. After the motor runs at low speed, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch ON: The motor runs in forward direction at low speed until it leaves the home switch. Then, it runs in reverse direction. When the home switch is reached again, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.



B

10. **Method 10**

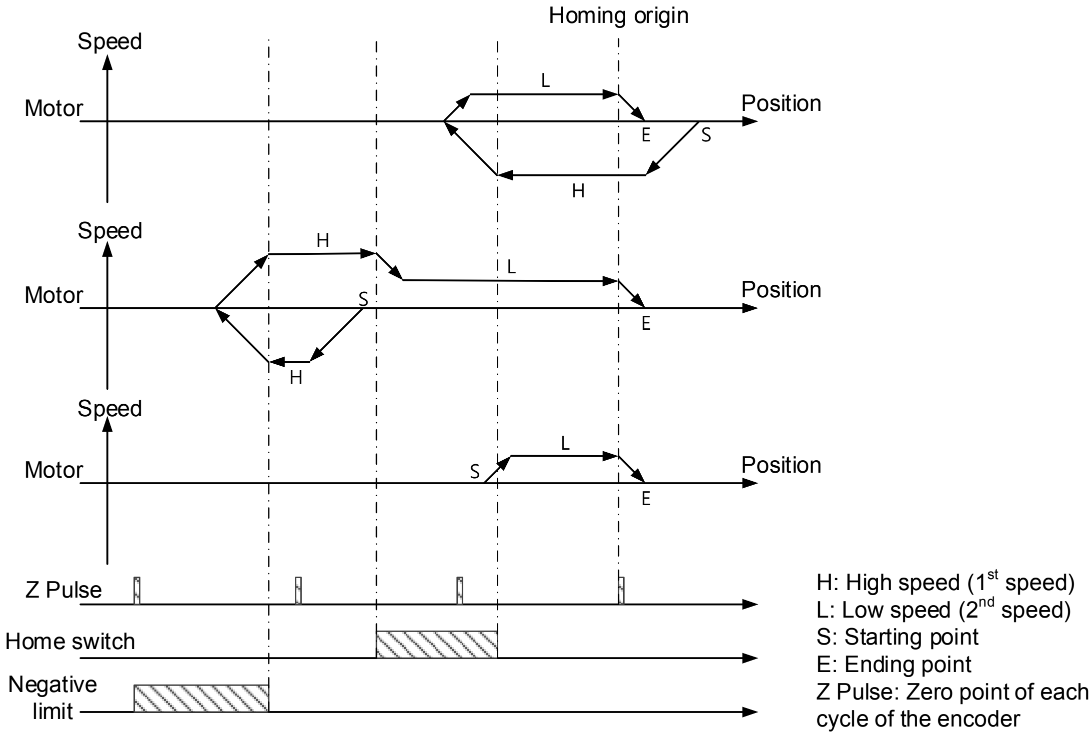
- Home switch OFF: The motor runs in forward direction at high speed until it reaches the home switch. Then, it runs at low speed. When the motor leaves the home switch, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch OFF: The motor runs in forward direction at high speed. When the motor triggers the positive limit before reaching the home switch, it runs in reverse direction until reaching the home switch. Then, the motor decelerates in forward direction. When the motor leaves the home switch at low speed, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch ON: The motor runs in forward direction at low speed until it leaves the home switch. Then, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.



B

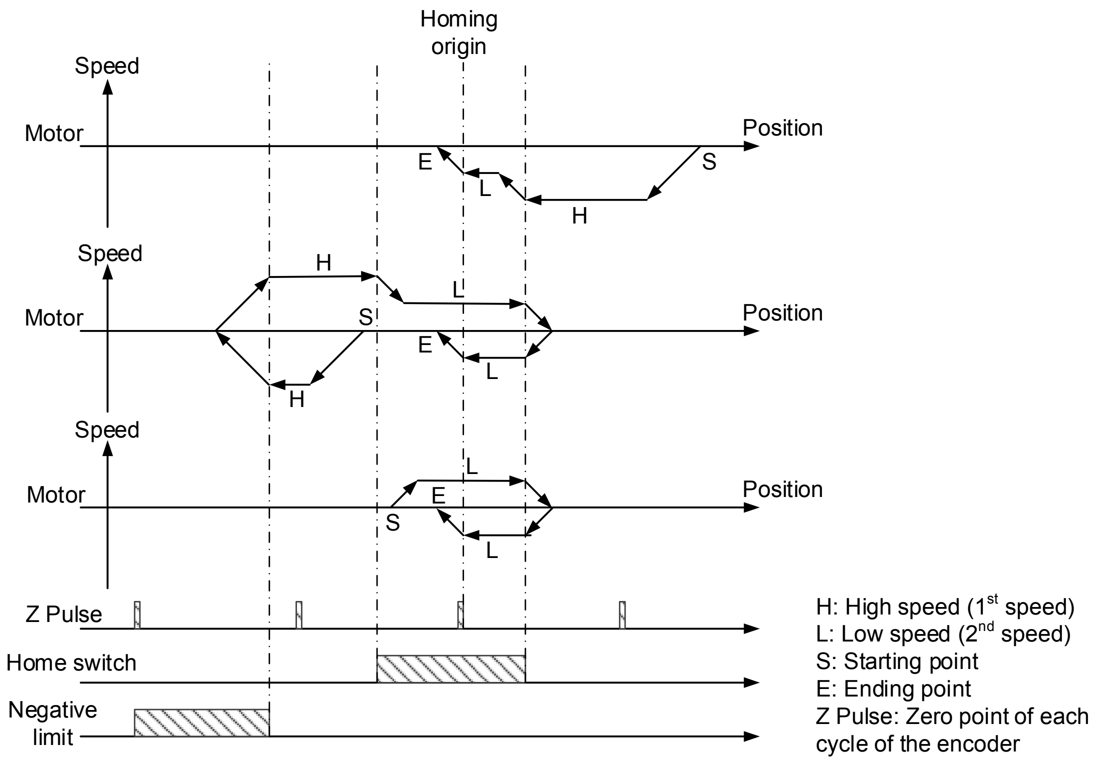
11. Method 11

- Home switch OFF: The motor runs in reverse direction at high speed until it reaches the home switch. Then, the motor decelerates in forward direction. When the motor leaves the switch at low speed, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch OFF: The motor runs in reverse direction at high speed. When the motor triggers the negative limit before reaching the home switch, it runs in forward direction until reaching the home switch. Then, the motor decelerates. When the motor leaves the home switch, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch ON: The motor runs in forward direction at low speed until it leaves the home switch. Then, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.



12. Method 12

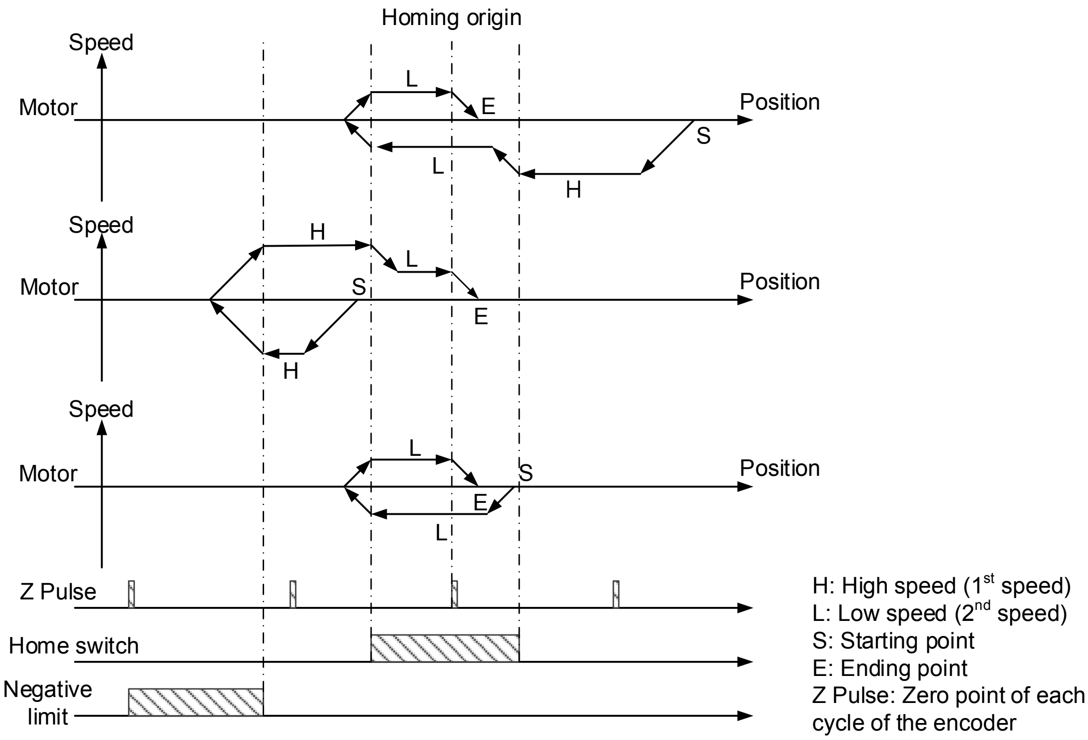
- Home switch OFF: The motor runs in reverse direction at high speed until it reaches the home switch. Then, the motor runs at low speed, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch OFF: The motor runs in reverse direction at high speed. When the motor triggers the negative limit before reaching the home switch, it runs in forward direction until reaching the home switch. Then, the motor decelerates and leaves the home switch at low speed. Afterwards, it runs in reverse direction. When the motor reaches the home switch again, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch ON: The motor runs in forward direction at low speed until it leaves the home switch. Then, the motor runs in reverse direction. When the home switch is reached again, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.



B

13. **Method 13**

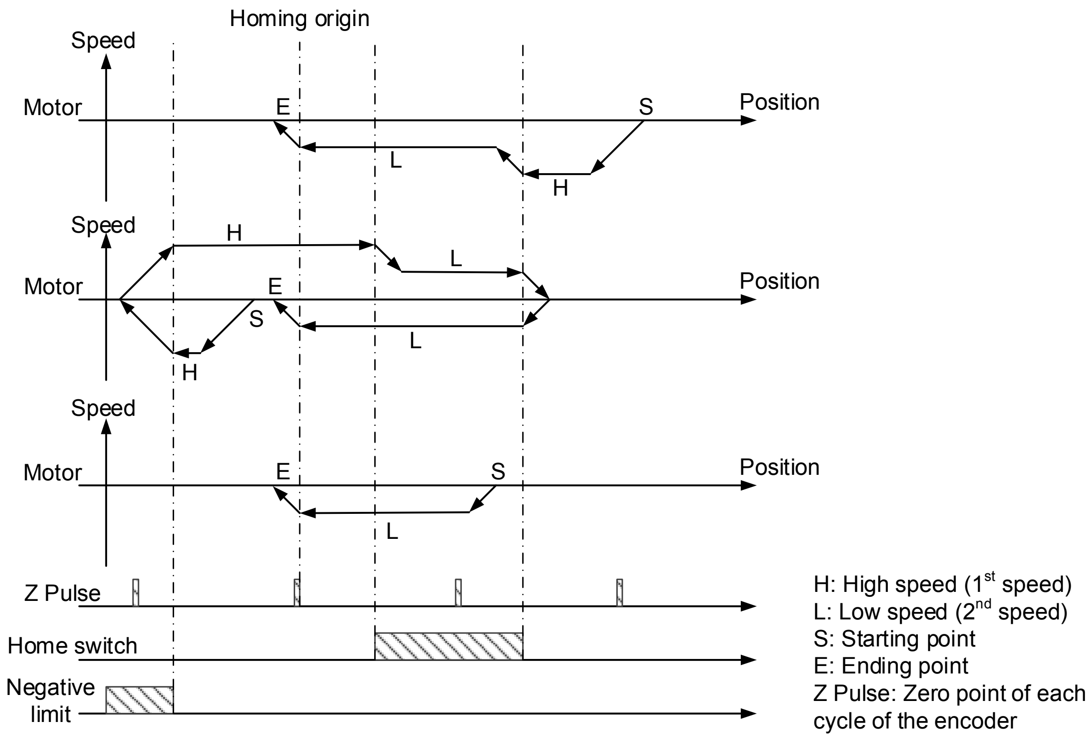
- Home switch OFF: The motor runs in reverse direction at high speed until it reaches the home switch. Then, the motor decelerates and leaves the switch at low speed. Afterwards, the motor runs in forward direction. When it reaches the home switch again, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch OFF: The motor runs in reverse direction at high speed. When the motor triggers the negative limit before reaching the home switch, the motor runs in forward direction until reaching the home switch. When the motor runs at low speed, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch ON: The motor runs in reverse direction at low speed until it leaves the home switch. Then, the motor runs in forward direction. When it reaching the home switch again, the motor starts to look for the first Z pulse and regard the first Z pulse as the new homing origin.



B

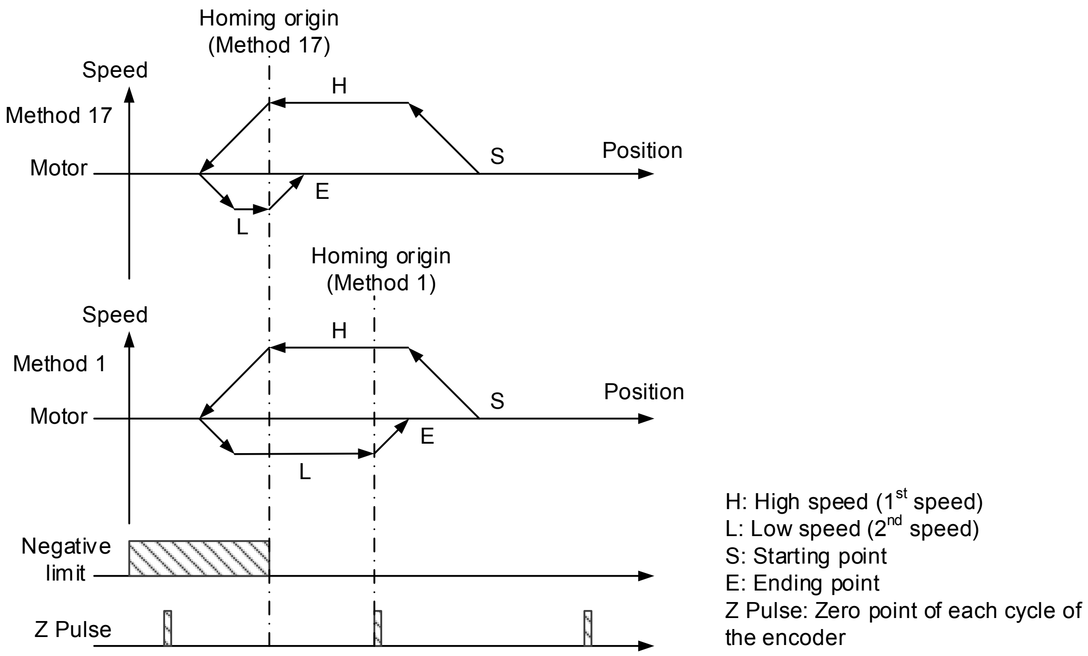
14. **Method 14**

- Home switch OFF: The motor runs in reverse direction at high speed until it reaches the home switch. Then, the motor decelerates to low speed. When the motor leaves the home switch, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch OFF: The motor runs in reverse direction at high speed. When the motor triggers the negative limit before reaching the home switch, it runs in forward direction until reaching the home switch. Then, the motor decelerates and leaves the home switch at low speed. Afterwards, the motor runs in reverse direction and reaches the home switch again. When the motor leaves the home switch again, it starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.
- Home switch ON: The motor runs in reverse direction at low speed until it leaves the home switch. Then, the motor starts to look for the first Z pulse and regards the first Z pulse as the new homing origin.



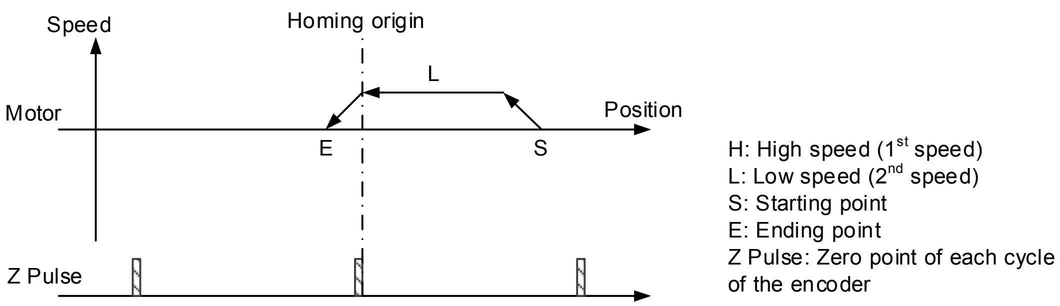
15. Method 17 ~ 30

Method 17 ~ 30 are similar to method 1 ~ 14 with following differences: In method 1 ~ 14, after receiving signals of the limits or home switch, the motor looks for Z pulse and regards the Z pulse as the new homing origin, whereas in method 17 ~ 30, the motor regards the signals as the new homing origin. Please refer to the figure below for the differences between method 1 and method 17.



16. Method 33

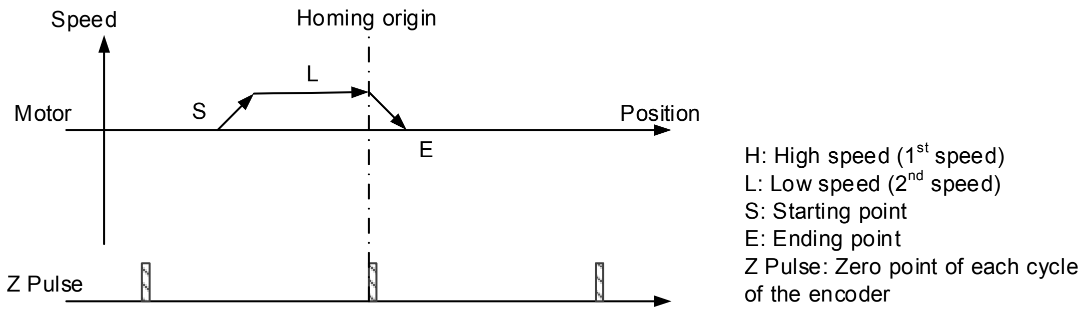
The motor runs in reverse direction looking for the first Z pulse and regards the first Z pulse as the new homing origin.



B

17. Method 34

The motor runs in forward direction looking for the first Z pulse and regards the first Z pulse as the new homing origin.



18. Method 35

The motor regards the current position as the new homing origin.

Revision History

Date of Release	Version	Revision
June, 2016	V1.0 (First version)	-
-	-	-
-	-	-

For more information about IPC Motion Platform User Guide, please refer to

(1) ASDA-A2 Series User Manual (Released on July, 14, 2015)

(2) ASDA-B2 Series User Manual (Released on November, 25, 2014)

(3) DMCNET Remote Module User Manual (Released on December, 27, 2013)

(4) DMCNET Gateway Remote Module Manual (Released on December, 13, 2013)

(This page is intentionally left blank.)